appviewx



ADC+ Guides

Version: 2025.0.0.0

Copyright AppViewX, Inc.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

This document may not be copied, disclosed, transferred, or modified without the prior written consent of AppViewX, Inc. While all content is believed to be correct at the time of publication, it is provided as general-purpose information. The content is subject to change without notice and is provided “as is” and with no expressed or implied warranties whatsoever, including, but not limited to, a warranty for accuracy made by AppViewX. The software described in this document is provided under written license only, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. Unauthorized use of software or its documentation can result in civil damages and criminal prosecution.

Trademarks

The trademarks, logos, and service marks displayed in this manual are the property of AppViewX or other third parties. Users are not permitted to use these marks without the prior written consent of AppViewX or such third party which may own the mark.

Contact Information

AppViewX, Inc.

222 Broadway, FL 19

New York, NY 10038

Email: info@appviewx.com

Web: www.appviewx.com

Contents

Preface 15

Chapter 1. ADC+ User Guide 17

Introduction 17

About AppViewX 17

Overview of ADC/Load Balancer 18

Need for a Centralised ADC Service Lifecycle Management Console/Interface, and More 19

What is AppViewX ADC+ 20

ADC+ Lifecycle Management 20

Getting Started 24

Accessing ADC+ 25

Demo Mode 26

Enabling Demo Mode 27

Understanding Features in Demo Mode 27

Exiting Demo Mode 32

Insights 32

Overview Tab 32

Compliance Tab 35

CVE Tab 37

Network Map 38

Viewing Network Map 39

Viewing the Network Routes for IP Addresses 39

Traffic Management 40

Dashboard Overview 40

Default Dashboard 51

Custom Dashboard 57

Launch Automation from Application-Centric View through Rules 109

App Search 109

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Asset Management 134
Asset Management Overview 134
Device Inventory 141
Onboard Device 153
Device Group 259
Config Management 261
Backup Configuration 261
Restore Configuration 270
Compare Configuration 277
Automation 282
Workflow Catalog 282
Workflow Request 295
Alerts & Logs 296
Alerts 296
Logs 313
Account 316
Role-Based Access Control 317
Accessing the Quick Config Option 317
Authentication 318
User 319
User Group 320
Role 322
Resources 323
Studio 333
Rules 335
Reports 337
Settings 339
Device Settings 339
Object Settings 343

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

iHealth Report....346
Statistics Settings....347
Glossary....349
Chapter 2. ADC+ Admin Guide....350
Getting Started....350
Inventory Actions....352
Before You Begin....352
Importing Devices....352
Managing Devices....353
Exporting Device Details....354
Deleting ADC Device(s)....354
Subscribing and Unsubscribing to Syslog....355
Config Fetch....356
Generating and Downloading an iHealth Report....356
Selecting Inventory Columns....357
Pagination....358
Device Group....359
Device Group Addition....359
Device Group Modification....360
Device Group Deletion....360
Assign a Device to Group(s)....361
Assign/unassign devices....361
Unassign a Device from Group(s)....362
Backup and restore....363
Backup and Restore....363
Configuring Device Backup....363
Restore....366
Configuring Backup Settings....370
Archive Setting Customization....371

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Compare Backup..... 372
Role Based Access Control..... 376
Before You Begin..... 376
Assign Devices to Resource..... 376
Assign Objects to Resource..... 378
Control Center..... 381
Searching for Objects..... 381
Control Center Search Result Views..... 388
View the Object Details..... 395
Bookmarks..... 397
Filter ADC Search Results..... 400
Export Search Results..... 400
Orphan Objects..... 401
Right click actions..... 402
Configuring Dashboards..... 409
About Dashboard..... 410
Create a Dashboard/Widget..... 415
Add Widgets..... 417
Dashboard Actions..... 438
Alert Management..... 443
System Settings..... 448
Device Settings..... 448
Configuration for Device Settings..... 449
Object Settings..... 452
Object Settings Configuration..... 452
Configuring F5 iHealth Report Settings..... 454
Configuring Statistics Collection..... 455
Configuration Updates through Syslog..... 457
Schedulers Used by ADC+..... 463

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Chapter 3. ADC+ API Guide..... 465

Understanding the AppViewX ADC+ API..... 466

Authentication using a Service Account..... 470

Retrieve session ID using login API..... 470

Using Session ID for further API calls..... 475

Authentication using a User Account..... 479

Retrieve Access Token using get-service-token API..... 480

Using Access Token in the header for further API calls..... 484

Object..... 488

Search for Object..... 488

Get Object Hierarchy..... 496

Composite Keys for F5 WideIp..... 504

Get Object Configuration..... 505

Action - Enable an Object..... 510

Action - Disable an Object..... 515

Action - Forcedown Object..... 520

Action - Forcedown Active Object..... 525

Action - Graceful Disable Object..... 530

Action - Enable Persistence for Object..... 534

Action - Disable Persistence for Object..... 539

Action - Clear Persistence Record..... 543

Action – InService (Activate) Object..... 548

Action - OutOfService (Suspend) Object 553

Action - Set Service Down Value..... 557

Action - Enable Proxy for Object..... 563

Action - Disable Proxy for Object..... 567

Get Objects in Constant Status..... 572

Get Objects in Constant State..... 578

Get Object Status Trend..... 584

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

State Status Drift Response....589
Action - Set Loadbalancing Method....595
Action - Set Connection Limit....601
Action - Set Ratio....606
Action - ARP Enable/Disable....612
Action - Set Weight....617
Action - Set CNAME....622
Get Transaction Status....627
Search - LTM and GTM Pool....632
Application View Widget - Action by Group....638
Update Traffic Grid Widget....644
Get Class Attribute Associations for a Class Management Widget....649
Update Class Attribute for a Class Management Widget....655
Device....666
Generate Device Backup....666
Get Backup Details for a Device....672
Device Backup....678
External Device Backup Download....682
Backup Archive....686
Device Restore....691
Device Restore Status....695
Search for Device Groups....701
Add Device....706
Delete Device....779
Search for Device....783
Sync Configuration....795
Manage Device....799
Unmanage a Device....804
Dashboard....808

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Create an Application Widget....809
Get Class Management Widget Details....819
Delete a Widget....827
Chapter 4. ADC+ Automation Workflow Guide....832
Introduction....832
Overview....832
Application Delivery Automation....832
Self-service and Automate Application Delivery services....832
Accessing ADC Predefined Workflow....833
Viewing Workflows (Application Provisioning Catalog) 834
F5 BIG-IP LTM....834
Create F5 LTM VIP – iRule....834
Create F5 LTM VIP with DNS....837
Create F5 LTM VIP with Service Now....843
Create F5 LTM VIP Advanced....850
Create F5 LTM VIP with Partition....857
Delete F5 LTM VIP - Basic....864
Delete F5 LTM VIP - Advanced....867
Disable Unused F5 LTM VIP 871
Disable and Delete Unused F5 LTM VIP....873
Get Unused LTM VIP and Delete....877
Get Unused LTM VIP and Notify....878
Disable and Delete Unused VIP-State/Status....879
Disable and Delete Unused VIP-State/Status/Statistics....883
Disable and Delete Unused VIP-Statistics....887
Notify Unused VIP-StateStatusStatistics....891
Delete F5 LTM VIP with Service Now....894
Manage F5 LTM Pool Members....897
Manage F5 LTM Pool members with Service Now....900

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Modify F5 LTM VIP Advanced..... 903
Create F5 LTM Multiple VIP.....906
Create F5 LTM Monitors - Multiple Devices.....912
Create F5 LTM Monitors..... 916
F5 BIG-IP GTM.....920
Create F5 GTM WideIP Basic.....920
Create F5 GTM WideIP Advanced.....929
Create F5 GTM WideIP with Topology..... 938
Create F5 GTM WideIP with Topology and Patition..... 946
Create F5 GTM WideIP with Service Now.....954
Modify F5 GTM WideIP Advanced.....961
Delete F5 GTM WideIP Advanced..... 963
Delete F5 GTM WideIP with DNS.....966
Disable and Delete Unused WIP-State|Status..... 968
Disable and Delete Unused URL-State|Status.....971
F5 XC Distributed Cloud.....973
Create F5 XC HTTP Load Balancer - Basic..... 973
Create F5 XC TCP Load Balancer - Basic..... 975
Delete F5 XC HTTP Load Balancer - Basic.....978
Delete F5 XC TCP Load Balancer - Basic.....979
F5 BIG-IP AS3..... 980
DNS.....980
LTM..... 987
F5 BIG IP ASM.....994
Web App Vulnerability Remediation..... 994
F5 BIG-IP..... 998
Infoblox IPAM.....999
Create Infoblox DNS Records – Basic.....999
Modify Infoblox DNS Records - Basic.....1002

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Delete Infoblox DNS Records - Basic..... 1005
Modify Infoblox DNS Record – Advanced..... 1008
Delete Infoblox DNS Records – Advanced..... 1011
Bluecat IPAM..... 1013
Create BlueCat DNS Records..... 1013
Modify BlueCat DNS Records..... 1017
Delete BlueCat DNS Record..... 1020
F5 BIG-IP System..... 1024
Automate F5 BIG-IP Upgrades..... 1024
Compliance - CVE Reporting and Golden Config Compliance..... 1035
F5 BIG-IP ZTP..... 1053
HA..... 1053
Standalone..... 1062
Migration..... 1079
Netscaler to F5 BIG-IP..... 1079
GITOps..... 1081
GitHub - Create New Project..... 1081
GitLab - Create New Project..... 1082
GITOps - Create or Modify GTM Application Services..... 1083
GITOps - Create or Modify LTM Application Services..... 1086
GITOps - Delete LTM or GTM Application Services..... 1089
GitOps - Pull Config and Execute AS3..... 1092
Nginx Plus..... 1095
Create NginxPlus LB Server..... 1095
Create NginxPlus LB Server with Service Now..... 1099
Delete NginxPlus LB Server..... 1103
Delete NginxPlus LB Server with Service Now..... 1104
Modify NginxPlus LB Server..... 1106
Modify NginxPlus LB Server with Service Now..... 1107

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Citrix SLB ... 1109
Create Citrix SLB Virtual Server ... 1109
Delete Citrix SLB Virtual Server ... 1112
Modify Citrix SLB Virtual Server ... 1113
Citrix GSLB ... 1115
Create Citrix GSLB Virtual Server ... 1115
Delete Citrix GSLB Virtual Server ... 1118
Modify Citrix GSLB Virtual Server ... 1120
Citrix Context Switch ... 1121
Create Citrix CS Virtual Server ... 1122
Delete Citrix CS Virtual Server ... 1125
Citrix NetScaler System ... 1127
Citrix NetScaler Golden Config Compliance ... 1127
RSeries ... 1133
RSeries Device Restore ... 1133
RSeries Device Backup ... 1134
Reports ... 1135
ADC Device Status Report ... 1135
ADC Device Status Report ... 1136
Sample Workflow Creation ... 1136
Sample Workflow Creation ... 1136
Designing F5 BIG-IP Automation Workflow ... 1136
Designing Infoblox IPAM Automation Workflow ... 1149
Designing BlueCat IPAM Automation Workflow ... 1160
Chapter 5. ADC+ Troubleshooting Guide ... 1171
Overview ... 1171
Asset Management Issues ... 1171
Device addition in Unresolved/Failed State ... 1171
A10 Device ... 1173

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Citrix Device....1173
Akamai device....1174
Amazon ELB....1175
WAF Devices....1176
Device Inventory Action Issues....1184
Issue in device import....1184
Issue in Unmanaging a Device....1186
Issue in Fetch Config....1187
Device Backup and Restore Issues....1188
Issue in schedule/trigger device backup....1188
Unable to View the Old Backup Files....1189
Issue in Email Notification for Configured for Backups....1190
Issue in Restoring a Device....1191
Dashboard Inventory Action Issues....1192
Issues in Dashboard Delete....1192
Issues in Dashboard Import....1193
Issues in Dashboard Export....1198
Issues in Dashboard Settings....1200
ADC Dashboard Specific Issues....1203
Issue in Class Management - Search....1203
Issue in Class Management - View Action....1203
Issue in Class Management - Modify Action....1204
Issue in Class Management - Rollback Action....1206
Issue in Application Widget Actions....1206
Application Widget Settings....1208
Script Execution Widget Actions....1209
Control Center Search and Action Issues....1210
Issue while performing actions on an object....1210
Issues in Control Center Search....1212

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

iHealth Report Generation Issues.....1217
Unable to Generate iHealth Report.....1217
Issues in iHealth Report Generation.....1218

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Preface

Revision History

Revision	Description	Date
1.0	Initial release of document for Release 2025.0.0.	November 2025

About the Documentation

Welcome to the ADC guide, which explains the capabilities of AppViewX's multi-vendor Application Delivery Controller (ADC) platform and provides step-by-step instructions to manage, automate, orchestrate, and monitor the application delivery services.

Audience

This document is intended for:

- Application teams
- Network Operations (NetOps)
- NetDevOps
- Traffic Management
- Automation and DevOps

Third-Party Software Acknowledgments

This section serves as a placeholder to document the third-party components referenced in this guide, along with their associated trademark information.

For example,

- This document includes software details developed by VMware, Inc. (www.vmware.com).

Text Conventions

This section defines the Notice icon and text convention used in this guide.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Notice Icons

Convention	Description
Note	Indicates readers to take note. Notes contain helpful suggestions or references to material not covered in the document.
Tip	Indicates readers that they can save time by performing the action described in the paragraph affixed to this icon.
Warning	Indicates readers that they can save time by performing the action described in the paragraph affixed to this icon.
Best Practice	Alerts readers to a recommended use or implementation.
	Indicates that the feature or functionality available only on AppViewX On-Prem.
	Indicates that the feature or functionality available only on AppViewX Software as a Service (SaaS).

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Chapter 1: ADC+ User Guide

Introduction

About AppViewX

The AppViewX Platform is the industry's first product category that offers Multi-vendor load balancer-centric Management, Automate and Orchestrate best-in-class and open-source application delivery services, and enables Self Servicing capabilities. It supports both traditional and non-traditional ADCs from leading vendors:

- A10 Networks
- Akamai Technologies
- Amazon Web Services (Elastic Load Balancers)
- Brocade
- Cisco
- Citrix
- F5 Networks
- HA Proxy
- NGINX
- Radware
- AVI

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Overview of ADC/Load Balancer

Application Delivery Controllers (ADC)/Load balancer distributes network traffic across servers. It acts as a reverse proxy to increase the reliability of the applications by distributing incoming traffic from clients across multiple server resources.

Three core functions of load balancers are:

- Efficiently distribute Network Load or Client's Request across servers.
- Sending requests to the server available, enabling high application availability.
- Increase and decrease the number of servers based on traffic.

Types of ADC/Load Balancer

On a high level, there are 3 types of Load Balancer based on appliance:

Hardware-based Load Balancer

A Hardware-based load balancer typically has a dedicated hardware appliance with relevant features. These are expensive to purchase and maintain, but gives users the full control and comes in different hardware configuration based on traffic requirement.

Popular hardware-based load balancer:

- F5
- Citrix Netscaler
- A10
- AVI
- Nginx Plus

Cloud-Based Load Balancer

A Cloud-Based Load Balancer is available on the cloud with full features, without the headache of purchasing and maintaining the costly hardware. It gives the user the option of "Pay as you use".

Popular cloud-based load balancer:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Amazon Web Services (AWS)
- Google Cloud (GC)
- Microsoft Azure

Software-based Load Balancer

A software-based load balancer id is available as a code. It can be installed on the server to turn them into load balancers. It also gives you the flexibility of choosing your own platforms like Windows, Linux, or Docker.

Popular software-based load balancer:

- HAProxy
- Nginx
- AVI

Need for a Centralised ADC Service Lifecycle Management Console/ Interface, and More

The Growth of IT Infrastructure is directly proportional to the growth of Business. As the IT Infrastructure grows it eventually includes the need for load balancers from many different vendors. The decision to include multiple vendors might be the relic of time, cost, and requirement of a specific application. Managing and maintaining it requires lots of resources and a comprehensive long-term plan.

The advantage of having diverse load balancer vendors is, your entire infrastructure is not vulnerable to specific network attacks or viruses. The disadvantage is managing different vendor products requires complex and costly support. Most of the time various product services come with different management tools, making IT infrastructure management complicated and unorganized.

AppViewX is Set to change all that, It is a modular, low-code software application that enables the automation and orchestration of network infrastructure using an intuitive, context-aware, visual workflow. It is closed-loop and state-aware, capable of verifying that intent has been achieved and providing actionable insights and automated remediation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What is AppViewX ADC+

AppViewX enables NetOps & SecOps teams to support ever-increasing business demands with greater agility. It offers state-of-the-art management capabilities that map to the needs of application owners, network engineers, and network operations.

ADC is an intuitive platform for Management, Automation, Orchestration, and Observability for application delivery services across Hybrid multi-cloud environments. With ADC+, you can gain application-centric insights, visibility into application services, and simplify application delivery infrastructure operations. It offers a single pane of glass/Interface and a holistic view of your application delivery across hybrid multi-cloud environments.

With ADC+, you can:

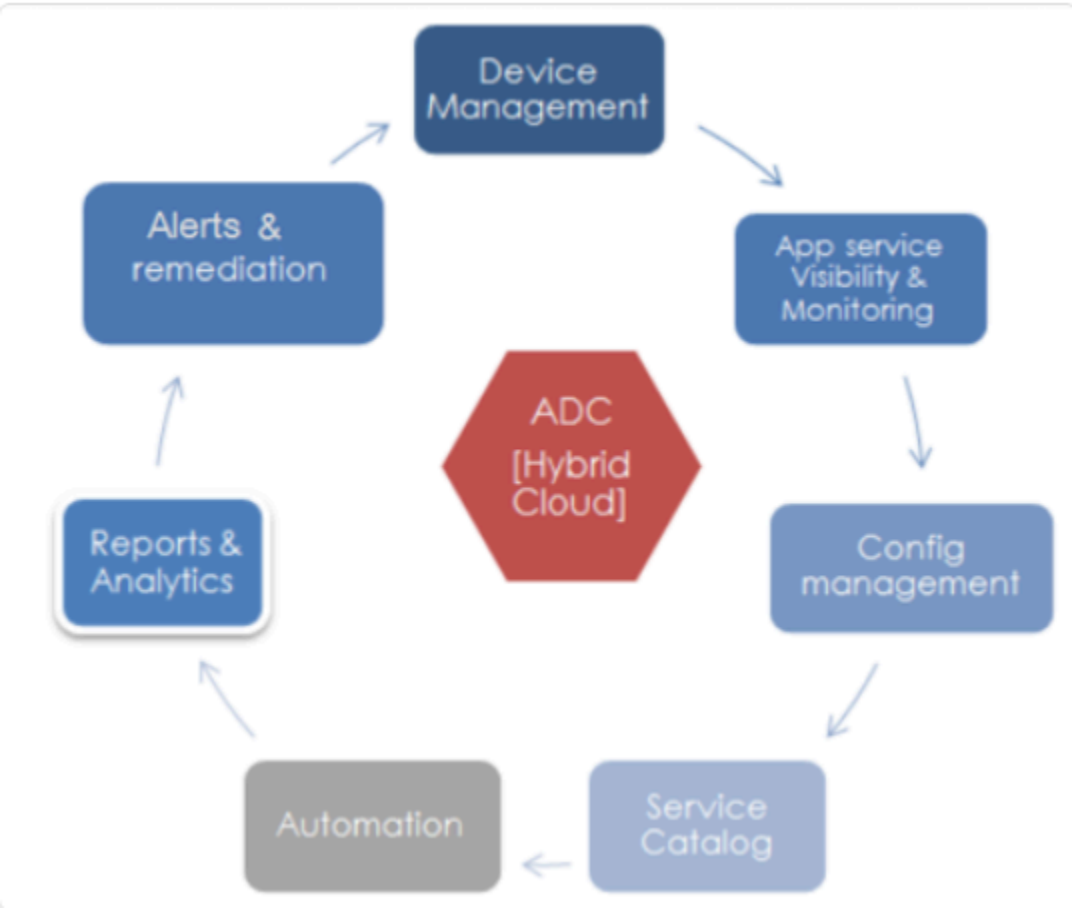
- **Device Management:** Onboard supported ADC+ vendor devices (Hardware, Cloud, and Software) into the AppViewX inventory using the IP Address/FQDN.
- **App Service Visibility and Monitoring:** Gain app-centric visibility for your ADC+ infrastructure in a single window.
- **Configuration Management:** Manage your network configuration by automating backups, tracking and reporting network changes, and ensuring compliance with defined policies.
- **Service Catalog:** Access an intuitive self-service catalog for service offerings tailored to Line of Business (LOB), employees, and partners via customized pages.
- **Automation:** Abstract application infrastructure and business applications to automate changes on your ADC infrastructure.

ADC+ Lifecycle Management

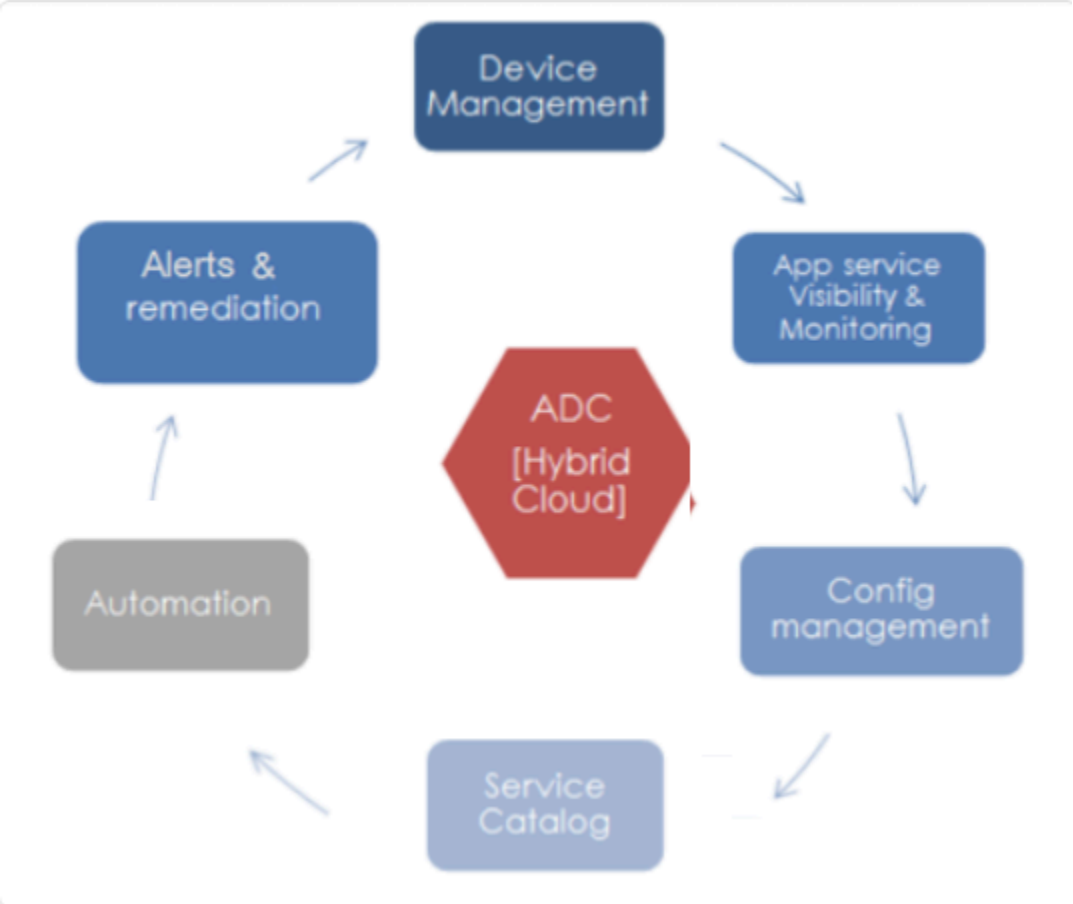
The ADC+ lifecycle defines the processes by which a Device or an Application is Discovered, Managed, Monitored, Automated, and Remediated. The multi-vendor (Cloud, On-prem) and intent-based load balancer application lifecycle management solution continuously aligns itself to required service levels, security, and compliance policies.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Device Management

- AppViewX's centralized multi-vendor Inventory Management allows you to onboard the supported ADC vendor devices (Hardware, Cloud, and Software) into the AppViewX inventory using the IP Address/ FQDN.
- AppViewX will initiate communication using the provided credentials and Discover the Applications/ Objects along with their configuration that is hosted on the devices. The Discovered Applications can be accessed within the product.
- Access to each device and its application objects can be controlled using access control policies. The AppViewX Platform applies granular access control to application objects, certificates, and configuration templates.
- The solution makes it easy to control access and delegate tasks and integrates with external directory service systems such as AD, RADIUS, TACACS, and LDAP. Roles can be easily created, changed, or discontinued per the needs of the organization.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To know more, refer to the Asset Management section.

App Service Visibility and Monitoring

- ADC provides App-centric visibility for your ADC infrastructure in a single window.
- The Search engine allows you to look for any application and recreates the topological view starting from the Global load balancer to the end server. This provides the network team visibility into the infrastructure to troubleshoot application-related issues faster.
- Get better insights into your Application health, state, status, utilization, and performance through pre-built dashboards.
  - Monitor and Manage Application Traffic through Custom Dashboards and Widgets that allow Traffic routing
  -  Monitor live traffic
  - Distributes traffic ratio across data centers
- Enables self-servicing and launch automation from application-centric views.

To know more, refer to the Dashboard section and App Search section.

Configuration Management

- Manage your network configuration by automating the backups, track and report network changes and ensure they are compliant with the defined policies.
- Audit the configuration drift and prevent unauthorized or unwanted changes by comparing the configuration across devices visually and restore the changes to adhere to the standards.

To know more, refer to the Configuration Management section.

Service Catalog

- ADC offers an intuitive self-service catalog to access service offerings for Line of Business (LOB), employees, and partners via customized pages.
- ADC ships out-of-the-box self-service catalog pages by persona for NetOps, Application Teams, SecOps based on the permissions assigned.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To know more, refer to Service Catalog User Guide.

Automation

- AppViewX's Application Delivery Automation solution abstracts application infrastructure and business applications to automate changes on your ADC infrastructure.
- The Visual Workflow tool provides an intuitive system for designing self-serviceable, event-driven and intelligent automation workflows. It leverages best-in-class automation and orchestration methodologies to ensure faster application deployments in brownfield and greenfield environments.
- It offers prepackaged automation workflows to provision application delivery and Infra and error-free application deployment and orchestration.

To know more, refer to the Automation section.

Alert and Remediate

- Track all the activities that take place within AppViewX or any external entity that is connected to the AppViewX system in the form of logs and alerts.
- Periodically monitor your Application Services or Devices to get notified and automate remediation in the event of any critical changes.

To know more, refer to the Alerts & Logs section.

Getting Started

Know your Deployments for ADC+

We support multiple deployment options to cater to various customer needs and infrastructure preferences. Our solutions can be deployed in On-Premises environments, Managed Kubernetes platforms such as EKS, AKS, GKE and OpenShift and as a Software as a Service (SaaS).

SaaS Deployment (Highly Secure and Hassle-Free)	
Our Software as a Service (SaaS) offering is designed for organizations that prioritize security, simplicity, and efficiency. In this deployment mode, we manage all aspects of application hosting, maintenance, and scaling, providing a worry-free experience for our customers. Our SaaS platform is built with cutting-edge security measures, including robust encryption, multi-factor authentication, and continuous monitoring to ensure your data and operations are protected at all times.	

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Choosing SaaS not only reduces the burden on your IT teams but also ensures that you benefit from the latest updates, features, and security enhancements without any additional effort. This option is ideal for organizations of all sizes, particularly those looking to quickly access our services with the assurance of enterprise-grade security and compliance. For additional information about saas deployment, click here.

On-Premises Deployment

On-Premises deployment enables organizations to install and operate our applications on their own infrastructure. This approach offers the highest level of control and customization, making it particularly suitable for organizations with strict security, compliance, or performance needs. It is best suited for enterprises with dedicated IT resources and the expertise to manage complex infrastructure. For additional information about on-premises deployment, click here.



Managed Kubernetes Deployment (AKS, EKS, and GKE)

We offer support for deployment on Managed Kubernetes services such as Amazon EKS, Microsoft AKS, and Google GKE. Managed Kubernetes platforms simplify the management of Kubernetes clusters by handling tasks like cluster provisioning, maintenance, and scaling. This deployment option is suitable for organizations looking for a balance between control and operational simplicity, and for those who prefer leveraging cloud providers' infrastructure and expertise. For additional information about managed kubernetes deployment, click here.

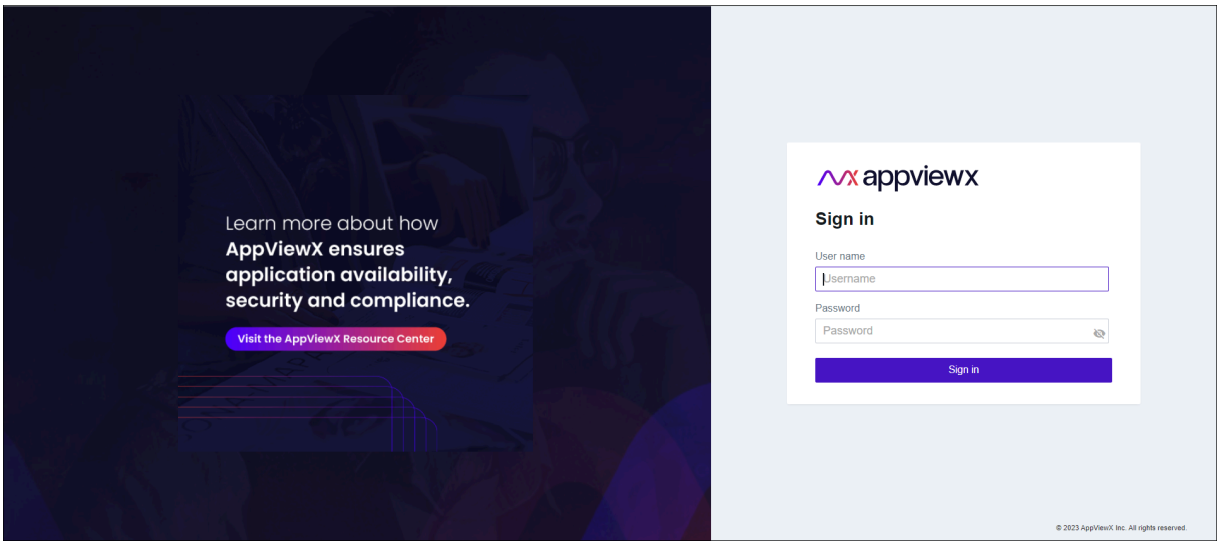


Accessing ADC+

Steps to access the ADC+ are:

1. Log into AppViewX with a valid credential (URL provided by AppViewX).
   The AppViewX Landing page appears.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



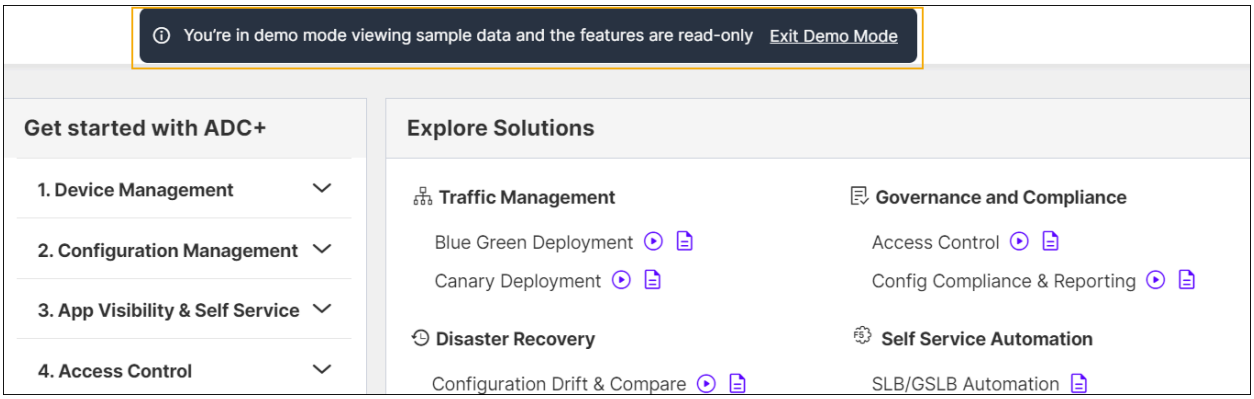
2. Click the menu button located in the upper-left corner of the screen.
3. Click **ADC+** on the left navigation bar.

Demo Mode

Demo Mode is a state in which the ADC system operates to showcase its capabilities and features in a controlled environment. In Demo Mode, you can explore all ADC features with sample data, but you will not be able to perform any actions.

The Demo Mode feature option can be found within the left navigation panel. You have the flexibility to enable or disable this feature according to your requirements. Availability of other ADC features in Demo Mode might vary depending on the permissions set by the Access Control Framework (ACF).

To access the Demo Mode, Navigate to **Menu** > **ADC+** and, then click the **DEMO MODE** toggle button. While in Demo Mode, you will notice that the Demo Mode option in the left navigation panel is highlighted. Additionally, there will be a banner displaying content indicating that you are currently in Demo Mode.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Enabling Demo Mode

To enable the Demo Mode:

1. Go to **Menu** > **ADC+**.
2. Click the Demo Mode toggle button.

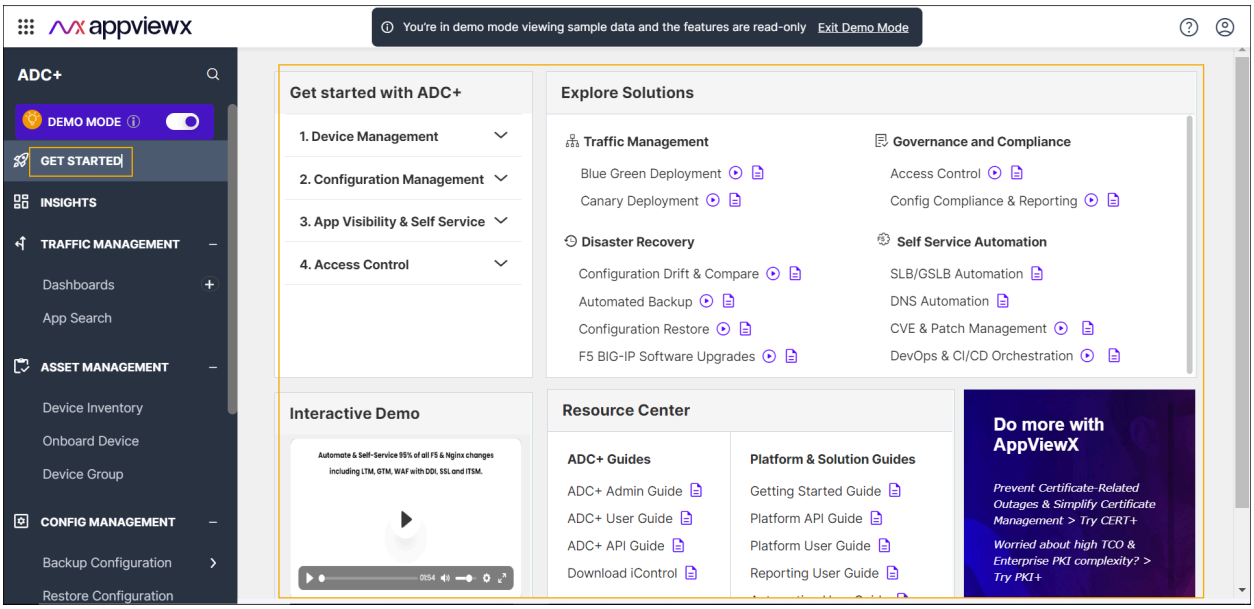
   The Demo Mode option in the left navigation panel is highlighted and a notification message displayed on the banner, indicating that the Demo Mode has been enabled.

Understanding Features in Demo Mode

Though you can not perform any actions within ADC+ while in Demo mode, you can still experience how the output appears for each action performed on the device/object.

Get Started

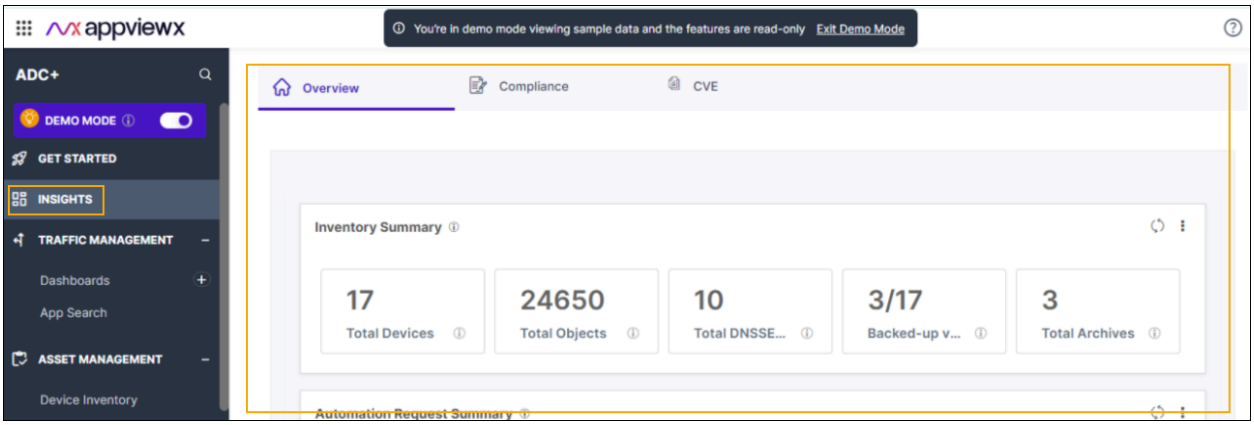
Getting Started page provides quick links to explore ADC+ features, solutions provided by ADC+, documentation for assistance, and links to other AppViewX offers. You can simply click on the links to be redirected to specific features, guided videos, or help documents.



Insights

Similar to the behavior observed in the ADC+'s live data, the Insights menu demonstrates comprehensive analytics using the sample data.

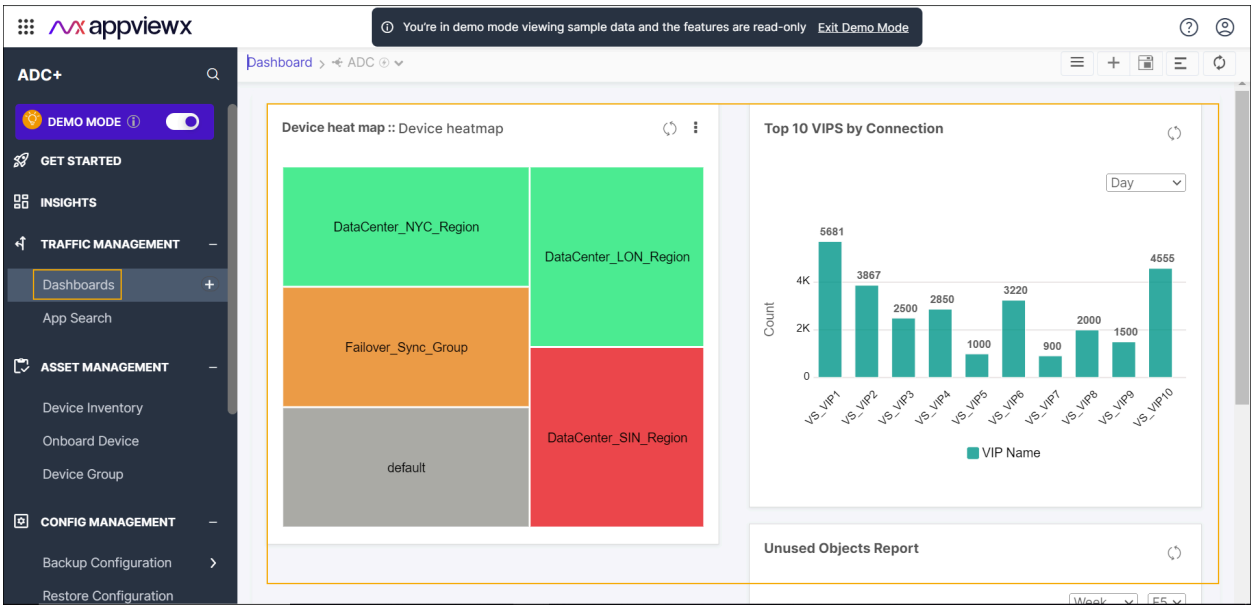
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Traffic Management

Dashboard

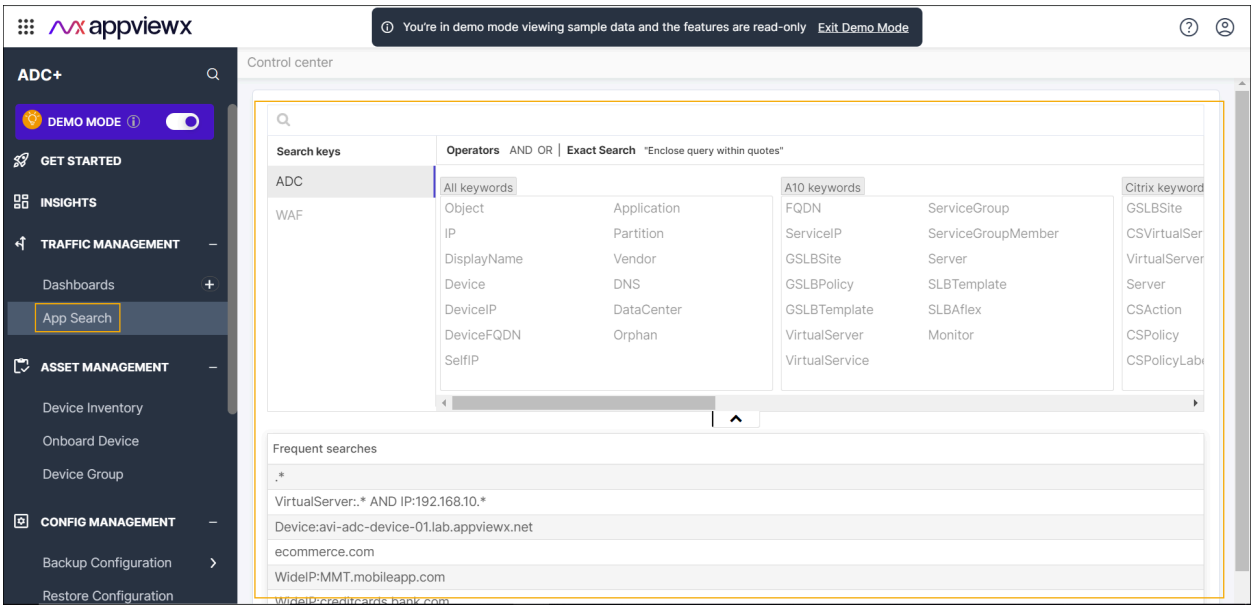
In the Dashboard menu, configured applications and their respective objects are presented through dashboard widgets. In Demo Mode, you can only showcase the data populated within the dashboard widgets and view all applications along with their associated objects.



App Search

In the App Search menu, you can access and view predefined keywords, vendor-specific keywords, and frequently searched terms. Frequent searches allow you to locate all entities, configurations, or objects related to ADC devices. This search results matching applications in your infrastructure.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Upon reaching the search results page, you have the option to,

- filter applications by status. These statuses include All, Active, and Standby.
- view the parent object and its details.
- switch between the Application view and infrastructure view.

Asset Management

Device Inventory

Within the Device Inventory page, you have the ability to interact with device information in a seamless manner. By clicking on any device link located in the **Name** column, you can instantly access comprehensive device details. Furthermore, you can also click on the status link found within the **Status** column to promptly view the device's status log.

Onboard Device

On the **Device Details** tab, you can only view the screen intended for onboarding a device associated with the select vendor. It is important to note that onboarding devices in the Demo Mode is not supported through this tab.

Device Group

The Device Group page provides access only to view the device groups along with their respective descriptions.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

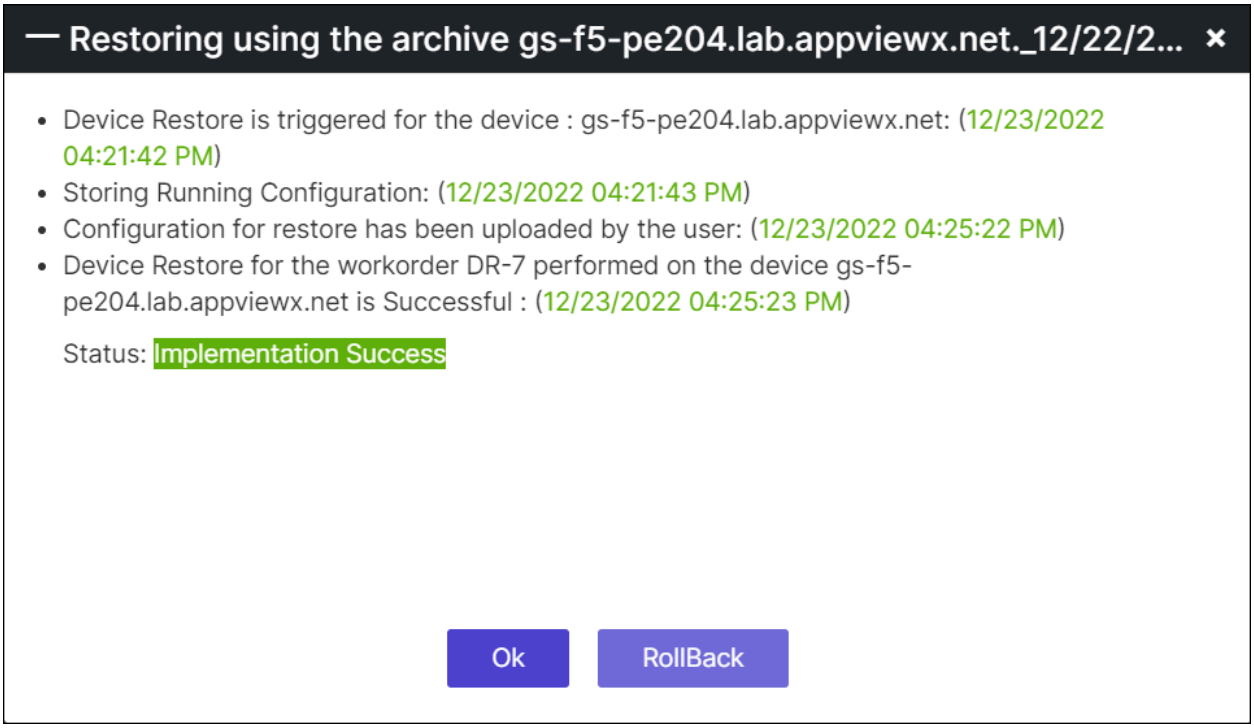
Config Management

Backup Configuration

- Within the Manage Backup page, you have access only to view the device backups that have been created.
- Within the Create Backup page, you have access to view the create backup screen. It is important to note that creating device backup in the Demo Mode is not supported.

Restore Configuration

Within the Restore Configuration page, you have the access to select a device or object, proceed with the restore functionality, and subsequently review the logs related to the restoration process. However, it is important to note that the capability to perform a rollback action is not supported in this context.



Compare Configuration

Within the Compare Configuration page, you have the access to select devices or objects, compare the configurations, and verify the archive details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Automation

- **Workflow Catalog**
  - Within the **View/Run** page, you can only view the available list of automation workflows organized into their respective categories. It is important to note that you will not be able to execute any of the workflows in the Demo Mode.
  - Within the Scheduled Jobs page, you can only view the list of scheduled requests.
- **Workflow Request**
  - On the Workflow Request page, you have the capability to view all triggered automation workflows categorized by their respective statuses. By clicking on a specific request number, you can access a detailed view of the stages comprising the executed workflow. This functionality enables you to gain insights into the progress and outcomes of each workflow instance.

Alerts and Logs

Within the Alerts inventory and Logs inventory page, you have the ability to only view the alerts and logs that have been triggered for the associated devices respectively. These pages provide a comprehensive overview of the notifications and recorded events related to your devices.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Exiting Demo Mode

To exit from the Demo Mode:

1. Go to **Menu** > **ADC+**.
2. Click the Demo Mode toggle button.
   The highlight indication on the Demo Mode option in the left navigation panel disappears and the notification message displayed on the banner also disappears.

Insights

The INSIGHTS menu provides comprehensive information and analytics regarding the overall device/object/application summary, compliance, and security of applications/devices delivered through the ADC infrastructure. The INSIGHTS menu populates with relevant data, once the devices have been successfully onboarded and the applications have been discovered. Each reports within the INSIGHTS menu refreshes every 2 hours at regular intervals. To access the INSIGHTS page, navigate to **menu** > **ADC** > **INSIGHTS**. These insights are organized into the following tabs:

- Overview
- Compliance

  On **Compliance Report & Remediations** section, you can

  - search for or apply filters to the devices based on Device type, Policy, or Device to refine your results.
  - initiate the remediation process for the non-compliant devices by clicking the **Remediate** button.
  - download the list of devices in the .csv format by clicking the **Download CSV** button.
- **CVE** - The interactive chart displays a summary of Common Vulnerabilities and Exposures (CVE) as well as an overall risk summary. To view the most recent data in the CVE Summary, click the refresh button. For additional details about the grid, click on the information button.

Overview Tab

The high-level summary provides an overview of device inventory, automation requests, device/object status, and more. To access additional information about the grid, simply click on the information button.

Inventory Summary

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The Inventory Summary report presents a concise overview of the devices, applications, backups, and F5 DNS secondary keys managed within AppViewX. Click on any widget to access detailed information specific to that particular summary on the respective widgets popup. Within the popup window, you can view all the entries and download the content in the desired format if needed. The Widgets within the Inventory Summary report are:

Widgets in the Inventory Summary

Widgets Name	Description
Total Devices	The count of devices that are managed within AppViewX.
Total Objects	The count of applications that are managed within AppViewX.
Backed-up vs Overall Devices	The count of devices with backed up configurations vs devices without backed up configurations.
Total Archives	The count of all archived devices.

Automation Request Summary

Automation Request Summary presents an overview of the status and trend of automation workflow requests. Click on any widget to access detailed information specific to that particular summary on the respective widgets popup. Within the popup window, you have the ability to view all entries, switch between **Request Details** and **By Workflow** views, and download the content in the desired format if needed. The Widgets within the Automation Request Summary report are:

Widgets in the Automation Request Summary

Widgets Name	Description
All Request	The count of workflow automation requests with all the status (open, closed, failed).
Open Request	The count of open workflow automation requests.
Close Request	The count of close workflow automation requests.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Widgets in the Automation Request Summary (continued)

Widgets Name	Description
Failed Request	The count of failed workflow automation requests.

Device Snapshot

The Device Snapshot provides a visual representation of the Device Inventory in the form of a graphical display. click on the corresponding area, and a popup window will appear with the relevant details. Within the popup window, you have the ability to device details by switching between **Status**, **Vendor**, and **Device Details** views, and download the content in the desired format if needed. The Device Snapshot visual representation includes areas that display the total number of devices and the breakdown of devices based on different categories or criteria.

Object Snapshot

The Object Snapshot provides a visual representation of the Application Inventory in the form of a graphical display. Click on the corresponding area, and a popup window will appear with the relevant details. Within the popup window, you have the ability to object details by switching between **Vendor**, and **Object Details** views, and download the content in the desired format if needed. The Object Snapshot visual representation includes areas that display the total number of object and the breakdown of objects based on object types.

Device CVE

The Device CVE snapshot provides a visual representation of the summary of the discovered CVEs in the form of a graphical display. Click on the corresponding area, and a popup window will appear with the relevant details. Within the popup window, you have the ability to view the list of discovered CVEs and download the content in the desired format if needed. The Device CVE visual representation includes areas that display the total number of discovered CVEs and the breakdown of CVEs based on severity.

Supported Vendor: F5

Device Compliance

The Device Compliance snapshot provides a visual representation of the summary of the compliant and non-compliant in the form of a graphical display. Click on the corresponding area, and a popup window will appear with the relevant details. Within the popup window, you have the ability to view

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

the list of devices pertaining to the selected area and download the content in the desired format if needed. The Device Compliance visual representation includes areas that display the total number of devices categorized by their compliance status. It also provides a breakdown of devices based on their compliance status, showing how many are compliant, non-compliant, or pending.

Supported Vendor: F5 and Citrix

Unused VIPs and WIPs

The Unused VIPs and WIPs block provides a visual representation of the unused VIPS and WIPs. Click on the corresponding block, and a popup window will appear with the relevant details. Within the popup window, you have the ability to view the list of unused VIPs/WIPs pertaining to the selected block and download the content in the desired format if needed. The Unused VIPs and WIPs visual representation presents a graphical display illustrating the count of the unused VIPs and WIPs for easy understanding and analysis.

Supported Vendor: F5

KSK Expiration

The KSK Expiration snapshot provides a visual representation of the summary of the keys with their expiration dates in the form of a graphical display. This snapshot helps for a quick overview and analysis of key expiration information. Click on the corresponding area, and a popup window will appear with the relevant details. Within the popup window, you have the ability to view the list of keys pertaining to the selected area and download the content in the desired format if needed. The KSK Expiration visual representation includes areas that display the total number of keys and provides a breakdown based on their expiration details. This breakdown helps to understand the distribution of keys according to their expiration dates, allowing for better visibility and analysis.

Supported Vendor: F5

Compliance Tab

Ensuring compliance of F5 BIG-IP device configurations with organizational standards is crucial. AppViewX offers a comprehensive solution that encompasses compliance enforcement, reporting, and closed-loop remediation to achieve this objective. In general, you have the flexibility to utilize pre-built golden configurations provided by AppViewX or define your own compliance policies. By conducting compliance checks, you can easily identify non-compliant configurations and initiate automated remediation with a single click.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

AppViewX offers a policy inventory that centralizes the management of all compliance policies. Within this inventory, you will find a collection of pre-built policies as well as custom policies created by users. You have the flexibility to select and run any of these policies against your F5 devices. The rules within each policy represent the specific parameters and their expected values configured for compliance. They help ensure that your F5 devices adhere to the desired configurations and standards defined by the policies.

Pre-built Compliance Policy

The pre-built compliance policy ensures that the basic system configurations of F5 are inherently compliant. For example, Banner, Device group auto sync , DNS Server, NTP Server, Gateway IP, Timezone, remote syslog servers, etc. Since this is a default policy, all device commands and remediation commands are handled in the background. You will only see the parameter and its expected default value, which can be modified as needed. For more details on the Automation Workflow, see Golden Config Compliance.

Custom Compliance Policy

You also have the ability to define custom compliance policies. These policies can encompass various aspects such as ensuring standard profile configurations, monitoring configurations, password policies, and more for LTM. When it comes to custom policies, it is necessary to formulate the validation command, condition, and remediation command for each rule. This information proves valuable when conducting the compliance check and ensuring adherence to the defined policies.

Device Validation Against Policy

After defining policies successfully, you can view the compliance status of devices, indicating the number of rules that are compliant versus non-compliant. Non-compliance signifies that a rule is not configured on the device as per the expected value. You have the option to switch to other custom policies and check their respective compliance status. This report offers an overall view of the compliance status of your ADC infrastructure in accordance with your organization's standards.

Use Case

If you want to receive a comprehensive report that includes detailed information about each compliance parameter, along with remediation recommendations, you can opt for the report presents the information in a tabular format, covering all the relevant details.

Solution: AppViewX offers a closed-loop remediation mechanism to ensure device compliance. By reviewing the report, you can identify values that require remediation. With a single click, the device configuration can be modified to meet compliance standards. Additionally, you have the option to

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

incorporate ServiceNow approvals to review and approve changes, as well as set specific change windows as needed.

Device Compliance By Policy

Within this section, you can assess the compliance of devices according to the selected policy and/or device. You have the option to view a graphical representation of the compliance status and also download it in the .png image format if desired. Click on the desired area to access the compliance details within the popup window. On the popup window, you have the ability to sort the list by clicking on each column. This allows you to arrange the data in ascending or descending order based on the selected column.

Compliance Report & Remediations

Within this section, you can evaluate the compliance of devices based on various criteria such as device type, policy, device selection, and search parameters. The devices are presented in a tabular format, allowing you to review their compliance status. If needed, you can download the device list in the .csv format for further analysis. By clicking the **Remediate** button, you can initiate the remediation process for non-compliant devices. This action redirects you to the automation workflow, where you can continue with the necessary steps to address the compliance issues and resolve them accordingly.

CVE Tab

CVEs serve as the primary source for vulnerability management by identifying potential vulnerabilities. As a standard practice, F5 discloses security vulnerabilities and exposures related to their products to assess the impact on devices. AppViewX, on the other hand, collects details of these Common CVE on a daily basis and maintains a CVE database within its platform. Currently, the source of information for AppViewX is the CVE Mitre site, where publicly disclosed information about security vulnerabilities and exposures can be obtained.

All the collected CVEs and their impact on F5 versions are cross-verified against the F5 devices managed in the inventory. AppViewX then proceeds to publish the vulnerabilities in the form of reports, providing valuable insights on the affected devices. The CVE summary report provides an overview of the number of CVEs detected across all F5 devices, categorizing them by severity (such as critical, high, medium, and low). Based on the severity of these CVEs, the overall risk summary of the ADC infrastructure is calculated. In this specific data set, the risk rating falls within grade D, as indicated here.

Each grade, ranging from A to D, in AppViewX has a predefined definition. The legend provided explains the logic behind these grades. For example, if there are 5 or fewer High CVEs, it falls under Grade A,

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

which represents the safest grade. With 10 High CVEs, the grade would be B, while 30 would fall under Grade C. Anything beyond that would be classified as Grade D. It's important to note that this logic can be customized according to specific requirements. This report provides valuable insights into the security posture of your infrastructure.

You will receive a comprehensive report that includes detailed information about each CVE, along with remediation recommendations. This report ensures that you have a clear understanding of the vulnerabilities and offers guidance on how to address them effectively. The report provides information on each CVE, including the CVE name, its score, the affected version, and the recommended version with the available fix. Let's consider one specific CVE as an example. This CVE addresses the xxx issue, and F5 recommends upgrading to version 15.x to resolve it. AppViewX offers a closed-loop remediation option to seamlessly perform this upgrade on the F5 devices in a quick and simplified manner. After the upgrade, the device's version will be updated from 14 to 15, ensuring that the device is free of vulnerabilities.

CVE Summary

Within this section, you can view the complete list of CVEs that impact the infrastructure. These CVEs are displayed and categorized by severity for a clear overview. By clicking on the corresponding area, a popup window will provide you with detailed information related to the CVEs. Within the popup window, you can view the list of discovered CVEs and download the information in the preferred format if required. The Device CVE visual representation includes sections that display the total count of discovered CVEs and further breakdowns based on their severity levels. This breakdown helps in understanding the distribution and severity of the CVEs present in the devices.

Overall Risk Summary

Within this section, you can view the risk summary of the devices based on the severity of CVE's in the form of graphs.

Supported vendor: F5

Recommendations & Remediations

Within this section, you can view report containing recommendations and remedial actions.

Network Map

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The Network Topology refers to the physical or logical arrangement of devices and connections in a network. The primary objective is to create a visual representation of the network topology, specifically focusing on managed firewalls, Application Delivery Controllers (ADCs), Layer 2 (L2) devices, Layer 3 (L3) devices, and routes. This graphical representation serves as a visual map of how these network components are interconnected, providing a clear and informative overview of the network's structure and configuration.

Benefits of Network Map

- **Network Topology Discovery**: The solution automatically identifies and maps various network elements within an organization's infrastructure. This includes critical components like routers, switches, firewalls, load balancers, and other network devices.
- **Application Connectivity Mapping**: It goes beyond hardware and also maps out the application connectivity requirements within the network. This means it can discover which applications and services are active in the environment and determine how they communicate with each other.
- **Impact Assessment**: This capability is crucial for assessing the potential consequences of making changes to firewall rules or modifying the network. It allows organizations to predict how such changes might affect the overall network and application connectivity.
- **Dependency Analysis**: The solution delves into the dependencies between various network components and applications. It helps organizations understand the intricate interrelationships between devices and services. This analysis is highly valuable for identifying potential bottlenecks, single points of failure, and security risks within the network.

This solution helps to network administrators and IT professionals, providing a comprehensive and visual understanding of their network infrastructure, application interactions, and potential vulnerabilities. It aids in making informed decisions, optimizing network performance, and enhancing security by uncovering hidden risks and dependencies within the network.

Viewing Network Map

To view a network map,

1. Go to **Menu** > **ADC+** > **Network Map**.
   The **Network Map** page is displayed.
2. (Optional) Use the mouse scroll button or zoom options to adjust the size of the topology view according to your preference.

Viewing the Network Routes for IP Addresses

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To view the network routes,

1. Go to **Menu** > **ADC+** > **Network Map**.
   The **Network Map** page is displayed.
2. Click the IP drop-down menu.
3. Enter the Source IP and Destination IP in its respective fields.
4. Click **Find Path**.
   Displays the different networks routes between the source and destination IPs.

> **Note:**
> To discard the search, click **Cancel**.

Traffic Management

Dashboard Overview

AppViewX Platform helps Enterprise IT manage, automate, and orchestrate application delivery services providing an application-centric view into the state of the application delivery infrastructure running in multi-cloud environments. Application, network and security engineers may self-service and initiate automation workflows in a single click that deliver compliance and true business agility. This gives application and operation teams a topology view of the application delivery infrastructure with better insights into application health, state, status, utilization, and performance.

The Dashboard module enables the user to manage, monitor, and interpret all the configured applications and their objects. It uses customizable widgets for managing ADC objects. The dashboard comes with a variety of pre-made dashboard widgets (components) that you can use to build a specific dashboard view that meets the different needs of your business.

It has been categorized into Default Dashboard and Custom Dashboard.

Change the Settings for a Dashboard

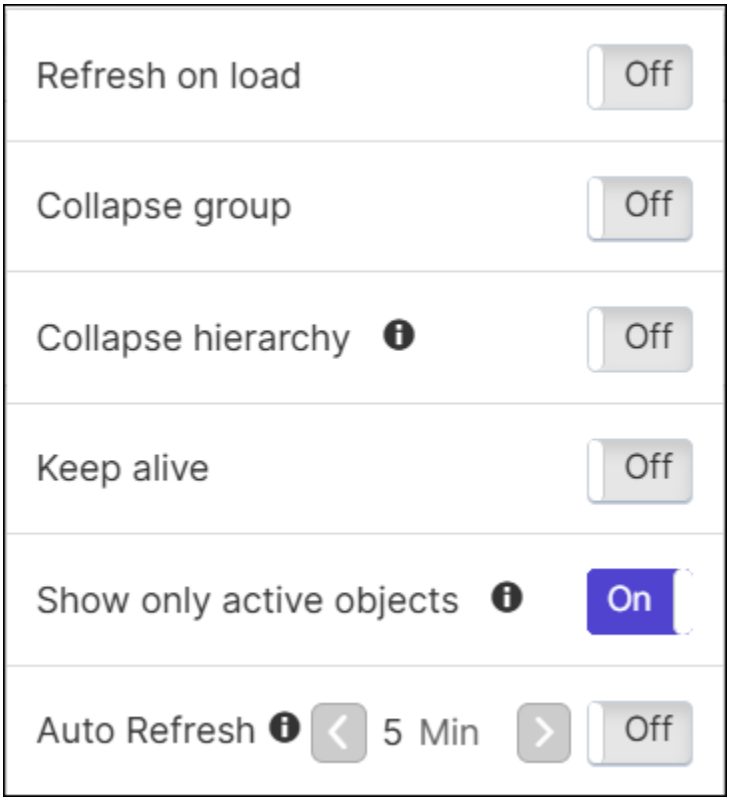
- AppViewX recommends to enable Syslog based configuration update so that any changes on the end devices with regards to state/status will be reflected in AppViewX on a near real-time basis ensuring a seamless experience.
- The ‘Last Refresh time’ on the widget displays the last time the widget is updated with the latest State/ Status information.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Click on the ‘Refresh’ button on the widget to perform on-demand State/Status information. Each Object will contain a last refresh time on the tooltip, you can also trigger a refresh for a particular object from the tooltip.
- Enable ‘Auto refresh’ from the ⚙ (**Settings**) of the dashboard page with a particular interval (1 - 5 mins) which will automatically refresh the state/status information of the objects.
- Enable ‘Refresh on load’ from the ⚙ (**Settings**) of the dashboard page to get the latest state/status information on loading the dashboard.
- ‘Keep Alive’ option on the dashboard **Settings**⚙ if enabled will ensure the session is not timeout by refreshing the dashboard every 15 mins.

To change the settings for a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**..
2. If you have more than one dashboard, in the dashboard list, click the name of the one whose settings you want to view and/or update.
3. In the Command bar for the dashboard, click the ⚙ (**Settings**) button.
   A dropdown menu appears, listing the settings you can change:



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Refresh on load** - If this setting is enabled, each time the dashboard loads, it displays the latest data in the system.
- **Collapse group** - If this setting is enabled, each time the dashboard loads, all groups within widgets on the dashboard display in a collapsed state by default.
- **Keep alive** - If this setting is enabled, the dashboard expiry function is disabled, allowing users to monitor all objects on the dashboard without interruptions caused by session timeouts.
- **Show only active objects** - If this setting is enabled, the objects of an active device are displayed in run time. This functionality is applicable only for Application View widgets.
- **Auto Refresh** - If this setting is enabled and refresh time is configured, the Dashboard data will be automatically refreshed at the specified time interval. The Auto Refresh option is enabled for users who have the Auto Refresh ACF permission. Users with this permission can set the auto-refresh time interval when creating an Application Widget dashboard or from the Settings > Auto Refresh option. Additionally, the admin user can disable the Auto Refresh option on demand for all users. To enable or disable the Auto Refresh option, see Platform User

4. If the setting is currently disabled, a 

(**Disabled**) button appears.
5. If the setting is currently enabled, an 

(**Enabled**) button appears.
6. To change a setting, click the respective button and the alternative setting will be displayed.

Create a Dashboard

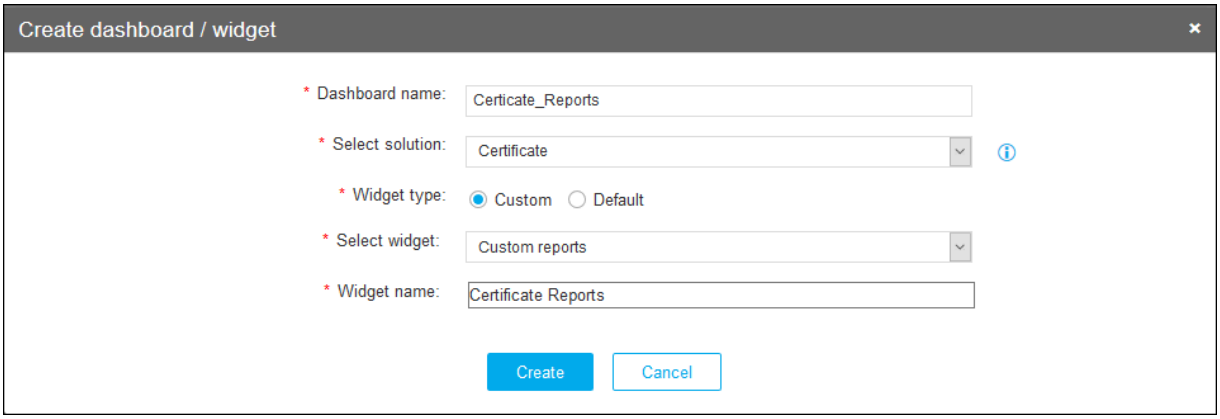
The custom dashboard allows you to manage, monitor, and interpret all the configured applications and their objects. It provides customizable widgets to get an overview of all the ADC Applications within the AppViewX platform and manage/monitor traffic from a single screen. Customize your Application dashboard with predefined and custom widgets as per the business need. You can create dashboard using any of the following method:

To create a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If this is the first dashboard you are creating in the system, click the **Create Dashboard** button that appears in the center of the screen. If you have already created at least one dashboard, click  (**Create**) icon on the top-right or click  (**Create**) icon.
3. On the **Create dashboard / widget** pop-up window, enter a name for the new dashboard.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The name must not contain spaces in it.



Create Dashboard/Widget - Field and Description Table

Field	Description
***Dashboard name**	Enter the name of the dashboard.
***Select solution**	Select a solution from the **Select Solution** dropdown to which you want the widget to be created: ADC, Firewall, Certificate, SSH, or WAF. Select the solution from the dropdown to which you want the corresponding widgets to be managed: Certificate.
***Widget type**	Select the **Widget Type** as follows: • **Custom** - Choose any one of the below options from the **Select Widget** dropdown: • **Application view** – Allows you to group the service objects of a single application. The widget displays the health of these objects and the number of current connections that the services are receiving. • **Traffic statistics** – Displays a chart showing live and historic performance statistics for individual device objects. • **Script execution** – Saves script files on a local machine and provides easy access

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	to maintain and execute script commands from within the widget. •  **Traffic grid** – Allows you to monitor and control the Traffic Percentage of the Applications across data centers. The status, state, and statistics for applications can be viewed through this widget. •  **Class management** – Allows you to view and modify the classes associated with iRules. •  **HeatMap** – Allows you to view statistics for managed, failed, and unresolved devices or device groups. • **Default** - Choose the default widgets you want to manage/monitor in the custom dashboard. Select the default widgets you want to manage/ monitor in the custom dashboard.
*__Select widget__	• For **Custom** widget type, Choose any one of the below options from the **Select Widget** dropdown: • **Application view** – Allows you to group the service objects of a single application. The widget displays the health of these objects and the number of current connections that the services are receiving. •  **Traffic statistics** – Displays a chart showing live and historic performance statistics for individual device objects. •  **Script execution** – Saves script files on a local machine and provides easy access

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	to maintain and execute script commands from within the widget. •  **Traffic grid** – Allows you to monitor and control the Traffic Percentage of the Applications across data centers. The status, state, and statistics for applications can be viewed through this widget. •  **Class management** – Allows you to view and modify the classes associated with iRules. •  **HeatMap** – Allows you to view statistics for managed, failed, and unresolved devices or device groups. • For **Default** widget type, choose the default widgets you want to manage/monitor in the custom dashboard. Select the default widgets you want to manage/monitor in the custom dashboard.
***Widget name**	Enter a name for the new widget that you will be creating on the dashboard.
**: Mandatory fields*	

4. Click **Create**.
   You are redirected to the widget configuration screen, which varies according to the widget selected.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- The Settings screen for the new dashboard/widget appears. The contents of the Settings screen vary depending on the type of widget you are adding to your new dashboard.
- All the default dashboards and custom dashboards created are listed in the dashboard inventory.
- If the Auto Sort Dashboards Alphabetically is enabled the dashboards are listed in alphabetical order, if disabled dashboards are listed based on the created order.

Delete a Dashboard

To delete a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one you want to delete.
3. When the dashboard opens, click the  (**Delete**) button in the Command bar at the top of the screen.
   A warning screen appears, indicating that deleting a dashboard remove all widgets associated with it.
4. Click **Yes** to continue.

Export a Dashboard

To export a dashboard from AppViewX:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. Go to the top-level of the Dashboard module by clicking the Dashboard link in the breadcrumb field or click the  (**Dashboard inventory**) button in the Command bar.
3. In the dashboard table, select the checkbox beside the dashboard you want to export.
4. Click the  (**Export**) button in the Command bar at the top of the screen.
5. On the **Export Dashboard** screen that pops up, select one of the following radio buttons:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Report (.csv)** - Select this option if you want to view the dashboard as a CSV file and do not plan to import it into another build of the AppViewX platform.
- **Import dashboard (.json)** - Select this option if you are downloading the dashboard with the intention of importing it into another build of the AppViewX platform.

6. Click **Export** to export the dashboard.

Import a Dashboard

To import a dashboard, the contents must be zipped and must contain single checksum .text files and/or .json files. This functionality is applicable only for the Dashboard that contains the Application View widget.

To import a dashboard into AppViewX:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. Go to the top-level of the Dashboard module by clicking the Dashboard link in the breadcrumb field or click the (**Dashboard inventory)** button in the Command bar..
3. Click the (**Import**) button in the Command bar at the top of the screen. The Import screen appears.
4. Click the **Browse** button and then locate and select the zip file you are importing.
5. Click **Upload**.
   The import function then checks to make sure the imported file and filename are valid.



6. After the zip file is successfully uploaded, select the checkbox in the column beside the dashboard name.

> **Note:**
> If the dashboard has the same name as an existing dashboard, use the Dashboard name field on the Import screen to enter a new name for the dashboard you are importing.

7. Click **Submit**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

8. On the Confirmation screen that appears, click **Proceed**.
9. Click the Dashboard link in the breadcrumb field to return to the top-level **Dashboard** screen.
   The new dashboard appears in the list.

Rename a Dashboard

To rename a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one you want to rename.
3. When the dashboard opens, click the  (**Rename dashboard**) button in the Command bar at the top of the screen.
4. On the **Rename dashboard** screen that pops up, enter a new name for the dashboard.
5. Click **Update**.

Save the Dashboard

To save the changes:

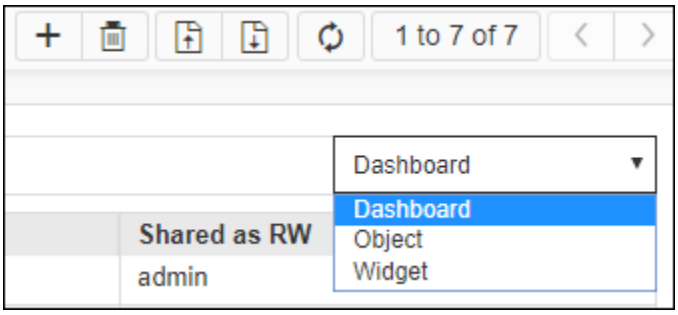
1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. After organizing widget by dragging and dropping them within the dashboard, click the  (**Save Dashboard**) button in the Command bar.
   A pop-up message appears at the top of the dashboard, "**Dashboard saved successfully**."

Search for a Dashboard, Object, or Widget

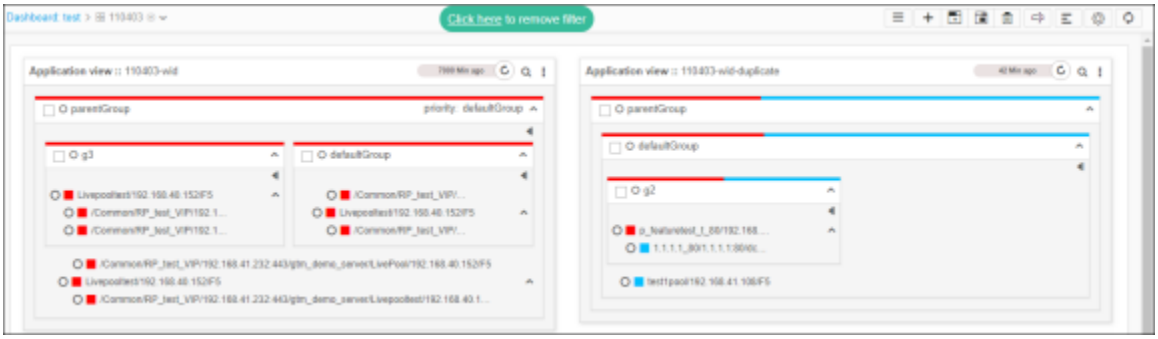
To search for a dashboard, object, or widget from the **Dashboard** screen:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. Click **Dashboard** in the breadcrumbs trails at the top of the screen to go to the top-level **Dashboard** screen or click the  **menu** button in the Command bar.
3. On the **Dashboard** screen that appears, enter the word or phrase you want to search by in the **Search** field.
4. Click the dropdown list at the right of the **Search** field and select the one you want to search for: **Dashboard**, **Object**, or **Widget**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



5. Press Enter key on your keyboard to run the search.
   The search results are displayed in a table below the search field.
6. Click one of the **Dashboard name** links in the search results field to open a filtered view of the corresponding dashboard, which shows only the widget or object within a widget that you searched for. In the image below, an Object search was conducted using the search phrase "testvserver.".



Share a Dashboard

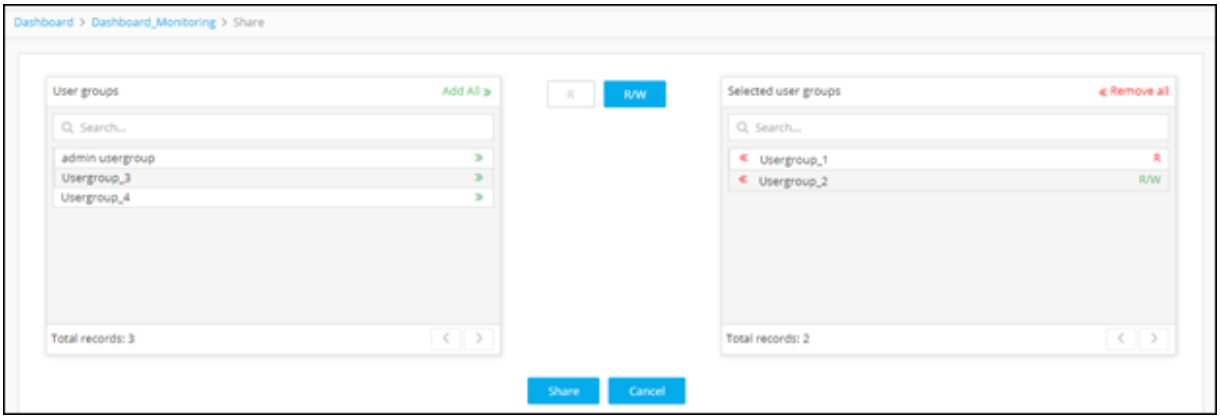
Depending on your permissions or access rights, you might have the ability to share a dashboard.

To share a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard table, click the name of the one you want to share.
3. When the dashboard opens, click the  (**Share**) button in the Command bar at the top of the screen.
   The **Share** screen appears.
4. Begin the assignment process by clicking the  (**Read-Only**) button. Each subsequent role you assign will only have read-only permissions within the dashboard.
5. In the **Roles** field, click the » (**Assign item**) icon beside each role you want to share the dashboard with on a read-only basis.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

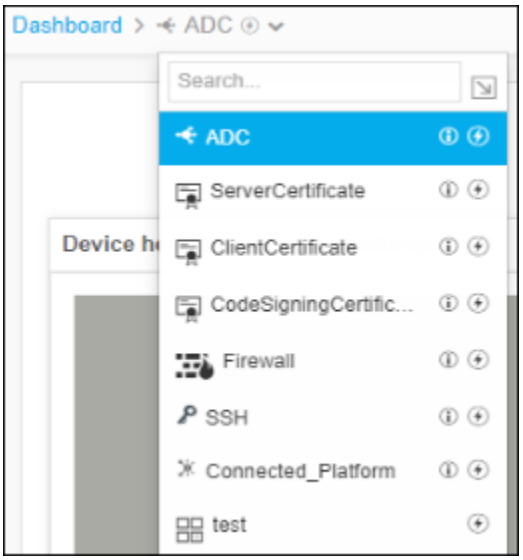
6. Click the R/W (**Read-Write**) button. Each subsequent role you assign will have read/write permissions within the dashboard.
7. In the **Roles** field, click the » (**Assign item**) icon beside each role you want to share the dashboard with on a read/write basis.
8. Click **Share** to finish.



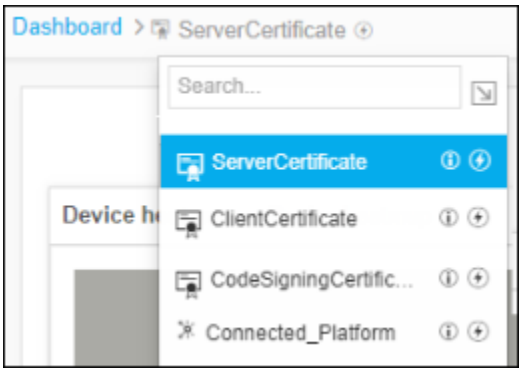
Switch Between Dashboards

To switch from one dashboard to another:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. Move your cursor to the breadcrumbs field of the current dashboard.
3. Click the current dashboard name.
4. From the dropdown list located next to the dashboard name in the breadcrumb, select the name of the dashboard you wish to switch to.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



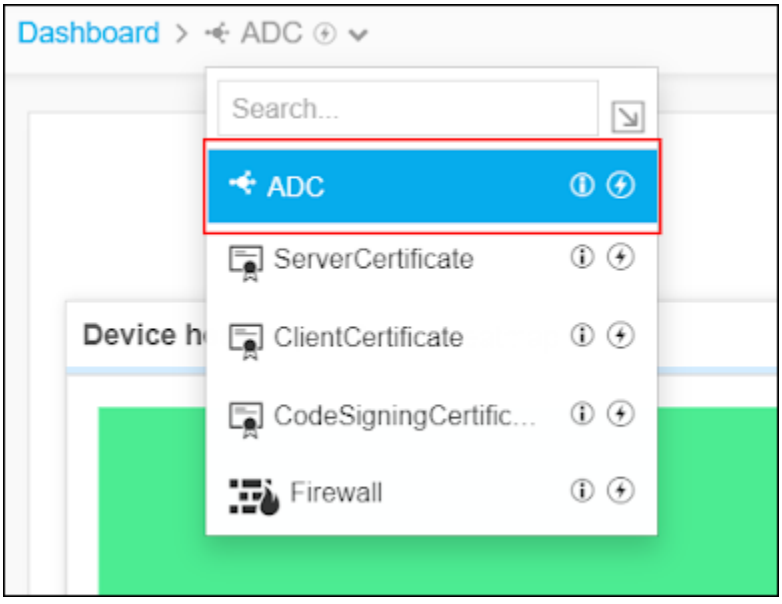
Note:

You can also click the (**Dashboard inventory**) button in the Command Bar and click the name of the dashboard you want to switch to.

Default Dashboard

Users can get access to a pre-built Insights dashboard to monitor key application services KPI's/metrics.

Displays the pre-defined widgets to provide the usage reports and up-to-the-minute reports containing statuses and statistics for devices and certificates managed within the AppViewX platform.



The purpose of the key widgets is to:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Monitor Device and Application health
- Monitor Top Applications Serving Maximum Traffic
- Monitor Unused Objects to Optimize LB Config
- Monitor App-centric Reports

AppViewX ships default dashboard called ADC available in dashboard inventory for monitoring the load balancing application managed in AppViewX.

This default dashboard is the landing dashboard screen by default. You can change the landing dashboard screen by clicking the  icon beside its name.

Monitor Device and Application health on a Pre-Built Dashboard. Get better insights into the Device and Application health based on the statistical data configured for a specific interval.

To view the reports related to ADC, go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**. A pre-built ADC Dashboard appears by default.

The following reports are segregated and displayed as widgets on the ADC screen:

- Monitor Device and Application health
- Monitor Top Applications Serving Maximum Traffic
- Monitor Unused Objects to Optimize LB Config
- Monitor App-centric Reports

Note:
The default widgets can be viewed, aligned, and/or refreshed and not editable.

Monitor App-centric

Applications that contain traffic, bandwidth, status, success, and object count statistics are shown in the App-centric report. You can get to the App-centric report through either of the following paths:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- By entering the application name in the search bar of the ADC dashboard page. The App- centric report appears when you click one of the search results.
- By clicking any individual component (bar, pie wedge, grid square) in one of the following reports: Application heatmap, Top 10 applications by connections, and Top 25 applications by connections.

The App-centric report contains the following statistical widgets:

- **DNS Success Rate** - Displays statistics on the DNS success rate of applications based on the load balancing method.
- **Server Status Report** - Displays the status of all LTM pools.
- **Application Object Count** - Provides a pie chart showing the status of each object in the application.
- **Application Bandwidth Report** - Displays the overall bandwidth consumed by the application on an odometer.
- **Traffic Statistics Summary** - Displays the Virtual Server (VIP) level traffic statistics of a particular application for the period you choose. It provides the total traffic/connection served by the application.
- **VIP Level Traffic Statistics** - Displays the individual Virtual Server (VIP) level traffic details for the period (day/week/month/quarter) you choose. It is used to determine the highest traffic served per Virtual Server (VIP) and all the traffic/connections served by the Virtual Server (VIP) in that application.

Monitor Device and Application Health

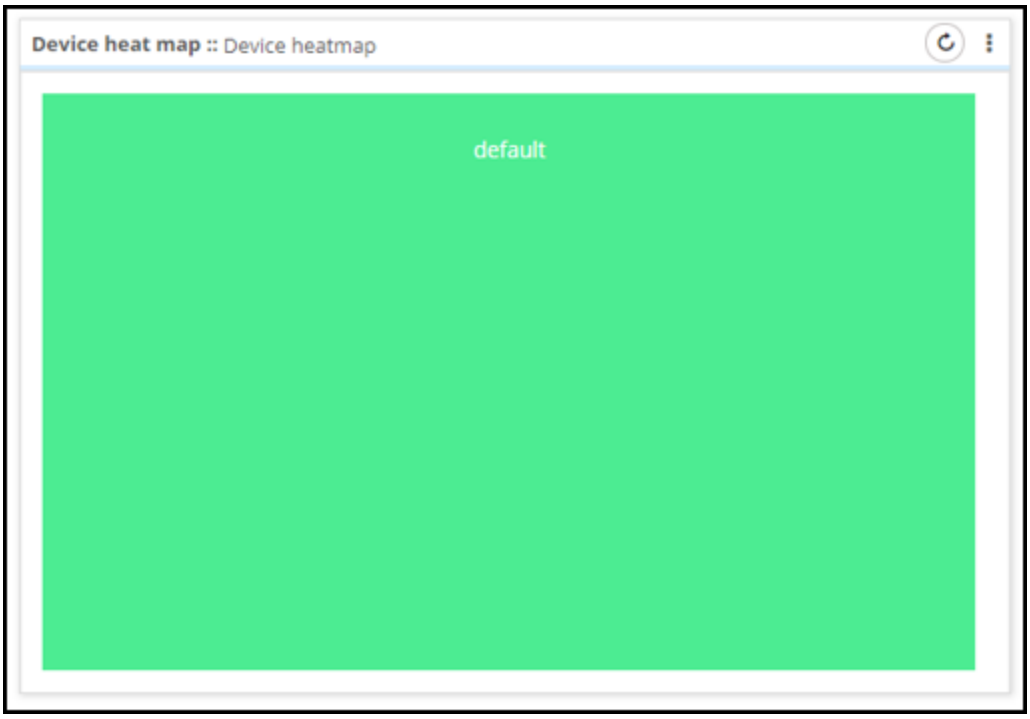
The following heat map reports are used to monitor device and application Health:

Device Heat Map

The device heat map is a graphical representation of the health of individual devices. The heatmap widget within the ADC dashboard displays each device group as a separate color block. When you hover your cursor over any device group block, a screen will pop up showing the number of Critical, Warning, Safe, and Not reachable devices available within the device group.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- If a device does not belong to any of the device groups in AppViewX, it appears in the default group block.
- When you click a device group block, the screen refreshes and displays all of the ADC devices available within the group as individual blocks.
- When you click on any device block, a screen appears displaying the memory, CPU, and bandwidth utilization, device info, logs, reports, and alarms for the respective device.

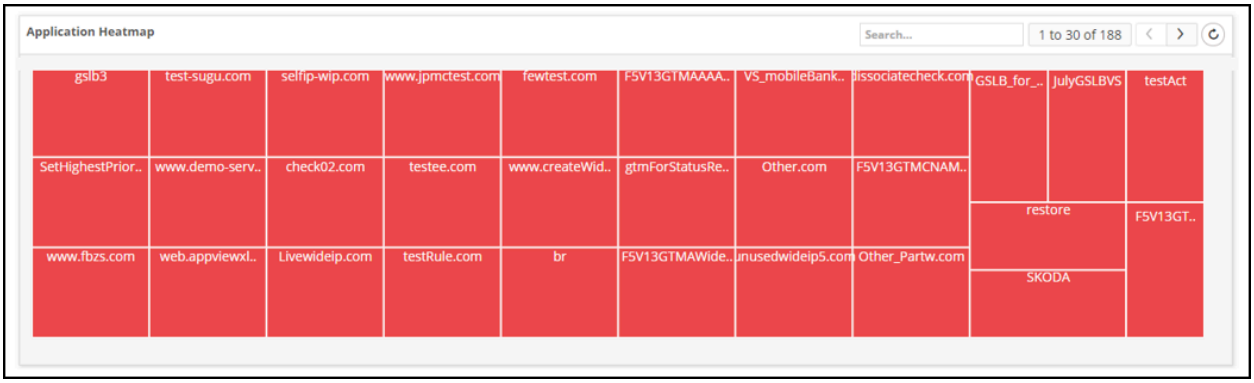


Application Heatmap

The application heatmap is a graphical representation of the status of an individual application, with each application displayed as a separate color block.

- When you click on any of the blocks, the respective App-centric report is displayed.
- When you hover your cursor over any of the blocks, the Application/Device name, State, Status, Success rate, and Object count are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Monitor Top Applications Serving Maximum Traffic

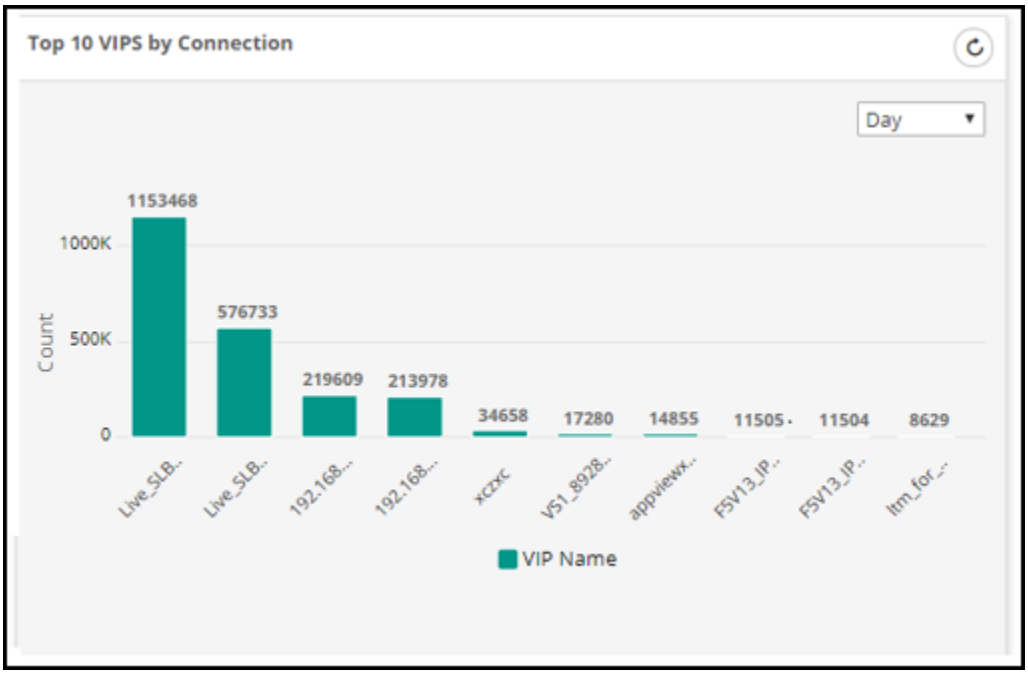
The top applications and their connections are monitored by the following reports:

Top 10 VIPS by Connection

A bar chart that shows the top 10 VIPs that currently has a maximum number of connections for a period (day/week/month/3 months) you choose.

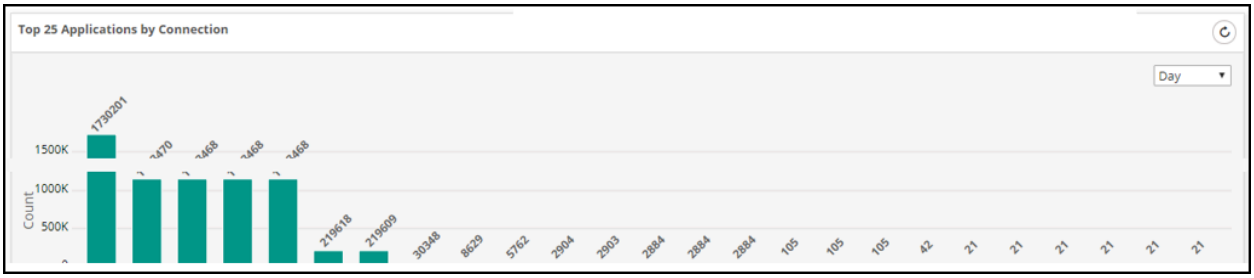
- When you click on any of the bars, the app search view of the ADC object is displayed.
- Application Heatmap - A graphical representation of the status of an individual application, with each application displayed as a separate color block.
- When you click on any of the blocks, the respective App-centric report is displayed.
- When you hover your cursor over any of the blocks, the Application/Device name, State,Status, Success rate, and Object count are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Top 25 Applications by Connection

A bar chart that shows the top 25 applications that currently have maximum connections for a period (day/ week/month/3 months) you choose. Each bar in the chart represents an application. When you click on any of the bars, the respective Appcentric report is displayed.



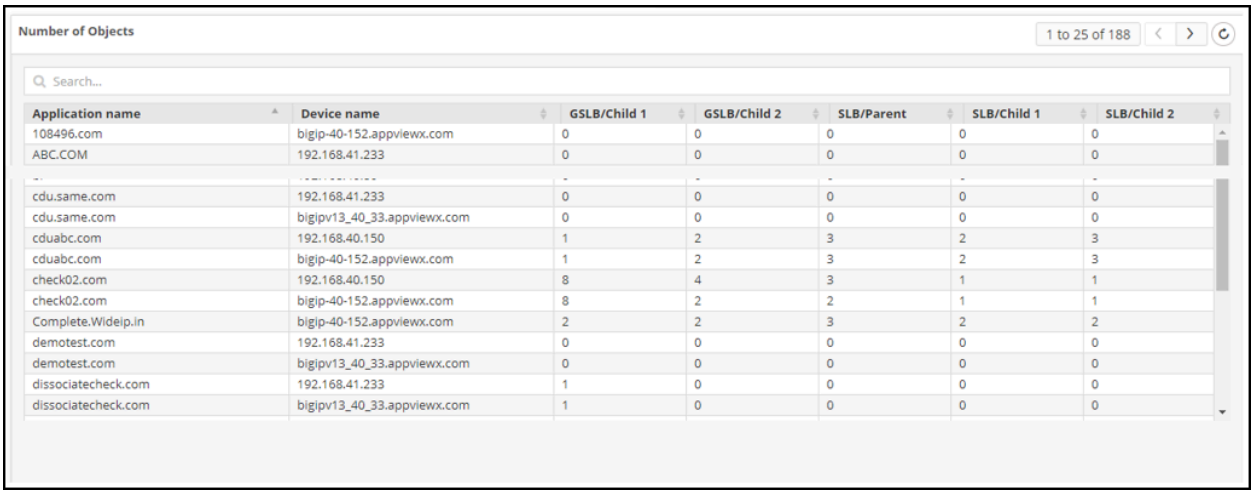
Monitor Unused Objects to Optimize LB Config

The following reports are used to monitor the unused objects in order to optimize LB configuration:

Number of Objects

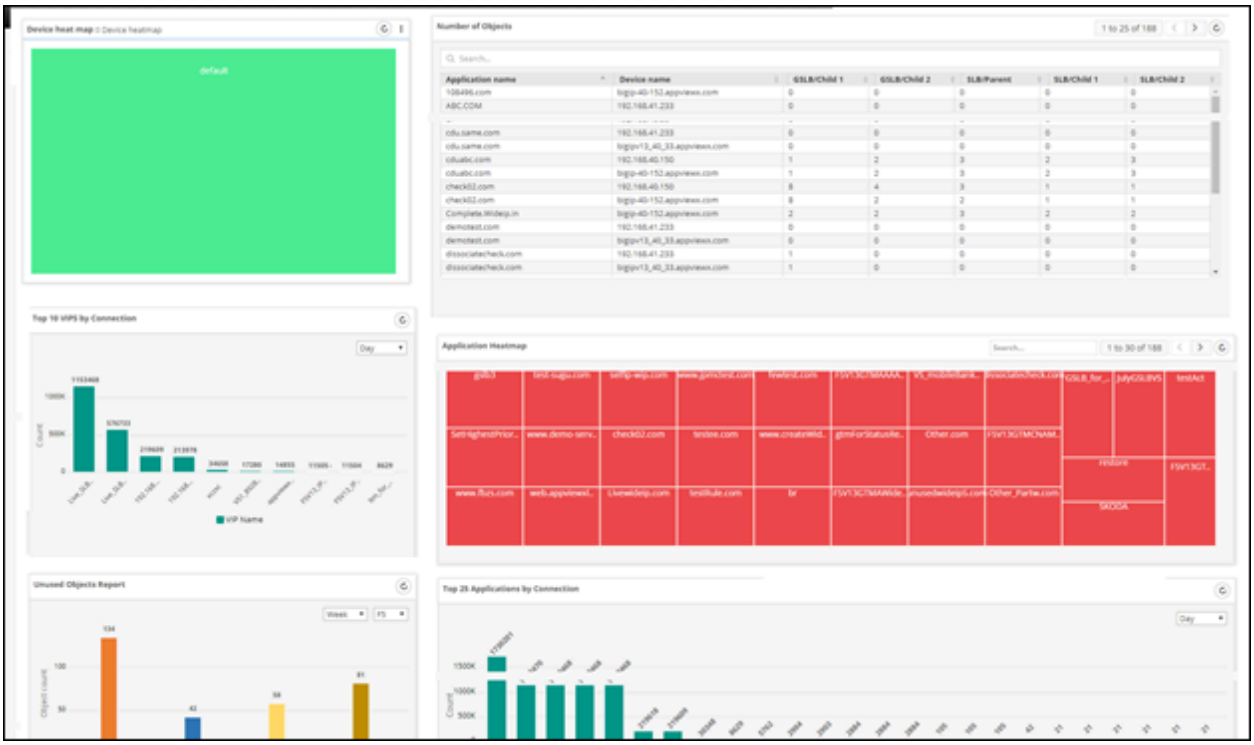
It displays the total number of objects available in each application. It provides the number of children available for each object based on their hierarchy. When you click on any application name, the topological view of the object is displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Unused Objects Report

A bar chart that shows the unused objects across all the active F5, Citrix, and A10 devices for a period (Week/Month/Quarter) you choose. This report helps users to scavenge unnecessary configurations of the device and manage configurations on the devices.



Custom Dashboard

The Netops users can design custom dashboard/app widgets in order to provide an overview of all the ADC devices and their objects within the AppViewX platform. A user can manage both the custom

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

widgets and the default widgets that belong to multiple solutions (such as ADC, Certificate, Firewall, WAF, SSH, Visual Workflow, and AppVision) in the Dashboard that are created.

Manage and Monitor Application Traffic on Custom Dashboards and Widgets

The custom dashboard allows you to manage, monitor, and interpret all the configured applications and their objects. It provides customizable widgets to get an overview of all the ADC Applications within the AppViewX platform and manage/monitor traffic from a single screen. Customize your Application dashboard with predefined and custom widgets as per the business need. It is highly flexible and enables you to create your own dashboards and share with selected users.

Application View Widget

An Application View widget allows users to group the service objects of a single application. The widget allows the user to manage and monitor the applications by displaying the health of their objects and the number of current connections that the services are receiving.

You can configure Application view widget on your dashboard by using any of the following method:

Configure an Application View Widget

Configure an Application View Widget Using Predefined Layout

Using Predefined Layout, you can configure application view widget by performing the following quick steps:

1. choose the widget Layout/Application grouping format.
2. configure Application patterns
3. tag Widgets to a Dashboard

Gain instant visibility into your Applications across Devices, Data centers, Sync groups etc., in less than 30 seconds. With frequent change in addition and removal of applications, AppViewX ensures that the Dashboards are dynamic in nature and updates any change in the Application configuration instantly.

With AppViewX’s granular and renewed RBAC, this becomes even more efficient as you can simply tag the rbac resources to the Dashboards. Application dashboards get dynamically created and updated exclusively for the users sharing the tagged resources.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

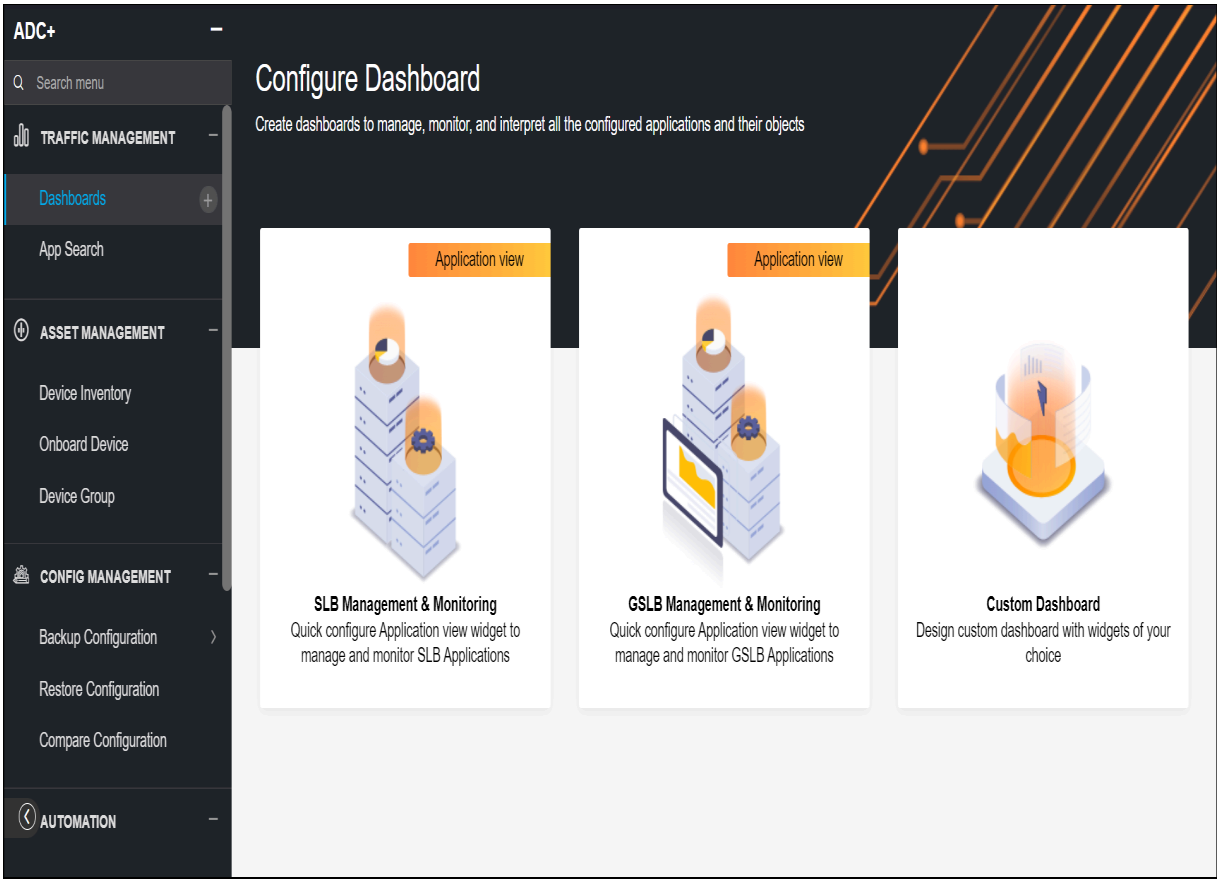
Note:
Even custom flavours can be created from this page using custom layout.

To configure an Application view widget using predefined layout:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. In the left navigation pane, click the add icon located beside **Dashboards**.
   The **Configure Dashboard** page appears. In this page, the layouts are categorized as follows:

   • SLB Management & Monitoring

   • GSLB Management & Monitoring



3. Hover the mouse over the desired layout category.
4. Click the **Use** button.
   The configuration page opens.
5. Select the field information in the **Application Layout** section:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Application Layout Section - Field and Description Table

Field	Description
Object Type*	Object type of the application. Select the kind of object you want to add. The options that appear in this field vary based on the selected option in the previous screen.
Select Object Grouping Format*	Choose how the applications should be grouped within the widget. The above image depicts how the applications will be grouped within the widget. The options are as follows: • **Data Center View** - select this option to group the applications based on the value given while adding the device. • **Device View** - select this option to group the applications based on device name.
**: Mandatory fields*	

6. Enter or select the field information in the **Add Objects** section.

Add Objects Section - Field and Description Table

Field	Description
Name	Name of the widget.
Add Resource (or) Add Objects	**Add Resource** - Click this button to configure the widget with a resource, which is already configured within AppViewX in **Accounts** > **Resource** page. By doing this, any updates made to the resources will automatically be seen in the widget as well. **Add Objects** - Click this button to add objects manually or use specific regex patterns so that any new applications matching the given regex will automatically be added to the widget.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	Note: Only 1000 objects can be included in each widget.
Add Widget	Click this button to configure one more widget with the same layout selected. Maximum of three widgets can be created for a layout.

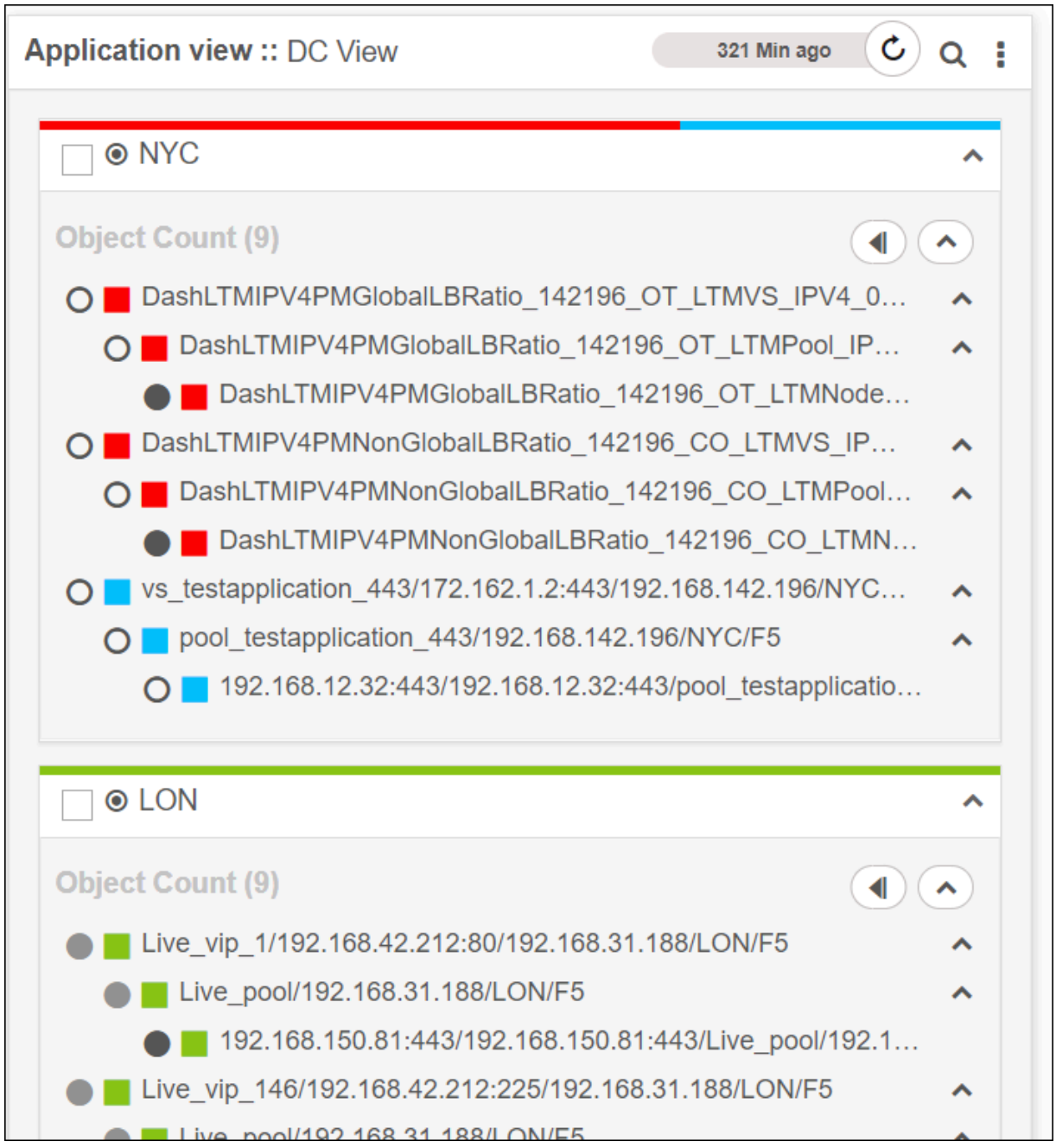
7. Select the field information in the **Dashboard Allocation** section.

Dashboard Allocation Section - Field and Description Table

Field	Description
Dashboard*	Select the **New/Existing** option to either create a new dashboard or select an existing dashboard into which newly created widget(s) to be added. **Note:** When you select **New** dashboard option, you need to provide name for the Dashboard.

8. Click **Create**.
   The widgets are created and you will be landed in the dashboard. You can see the widgets created with the input applications grouped based on the layout selected.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

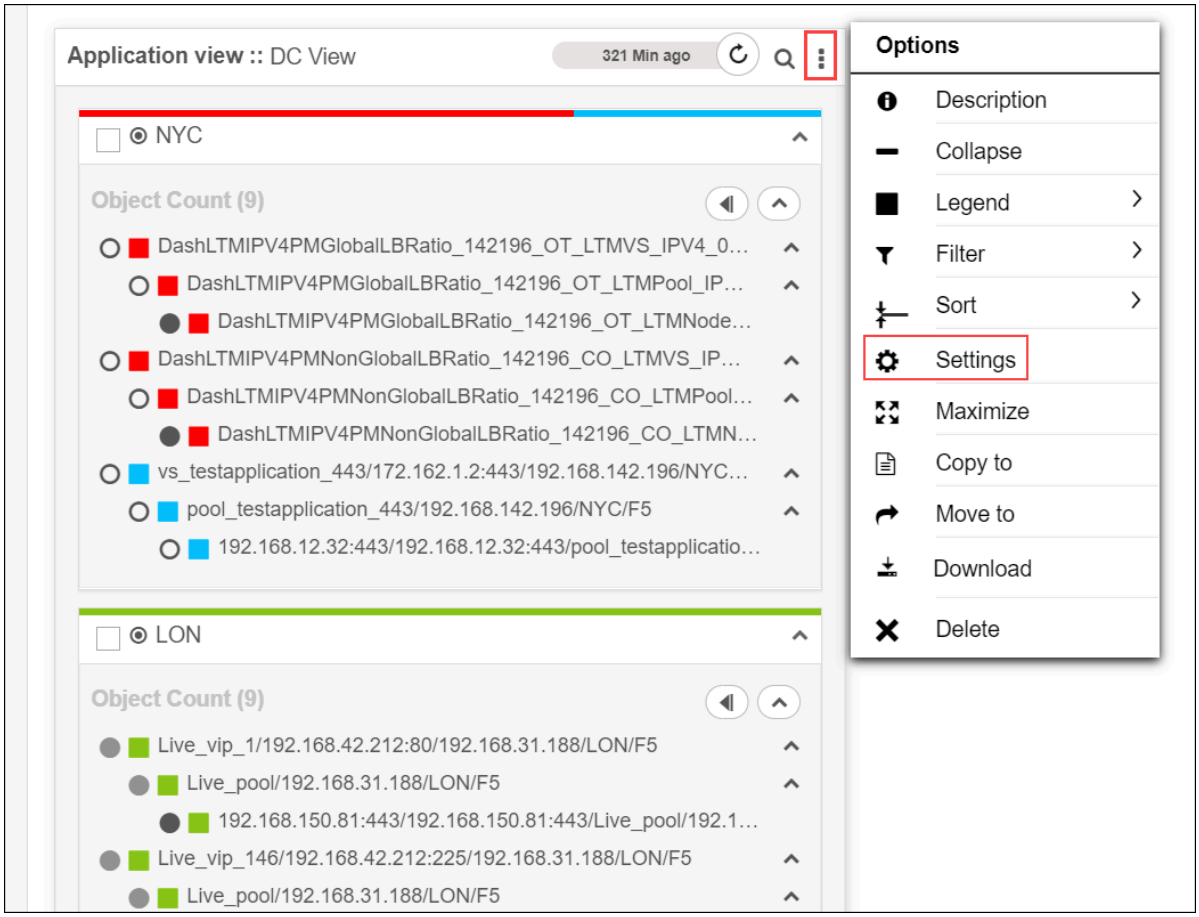


9. To reconfigure the applications or to change the layout,

Note:
For a widget configured with layout, you cannot add/delete objects or perform any group level actions from the below screen as the groups are updated dynamically based on layout selected.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Select **Settings** from the dashboard options.



- To change the layout,

  a. Click the edit icon located next to the layout name.

     The configuration page opens.

  b. Change the layout in the configuration page, and then click the **Update** button.

- To remove the applied layout, click the delete icon locate next to the layout name, and then click **Proceed** in the confirmation pop-up.

> **Note:**
> Removing the layout converts the widget to a custom layout and the associated widget group will no longer update dynamically.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Configure an Application View Widget Using a Custom Method

AppViewX offers Standard predefined templates for GSLB and SLB Applications management that allows you to create Instant Application Dashboards. By using the templates, the grouping of objects will be handled dynamically upon scaling of applications. This reduces the widget configuring time. Along with this the maintenance of the widget is also automated as the widget will be grouped automatically based on the layout selected. Follow three simple steps to create an App Dashboard quickly for your Application teams that enables seamless self servicing.

To configure an Application view widget using standard method:

1. If you are creating an Application view widget as part of creating a new dashboard, complete the steps in the Create a Dashboard topic, then jump to Step 5 below. If you are creating an Application view widget for an existing dashboard, click the  (**Add widget**) button in the Command bar of the dashboard.
2. On the **Create** widget screen that pops up, select the solution ADC from the dropdown list.
3. Select the **Custom** radio button as the type of widget.
4. Select the **Application view** as the widget you will be creating.
5. Enter a unique name for the widget.
6. Click **Create**.
   The **Settings** screen for the Application view widget opens, displaying an empty DefaultGroup where the user can add the objects.

   Note:
   For example, In order to Monitor SLB objects of specific applications, start creating groups in the name of the Application and add SLB objects into required groups.

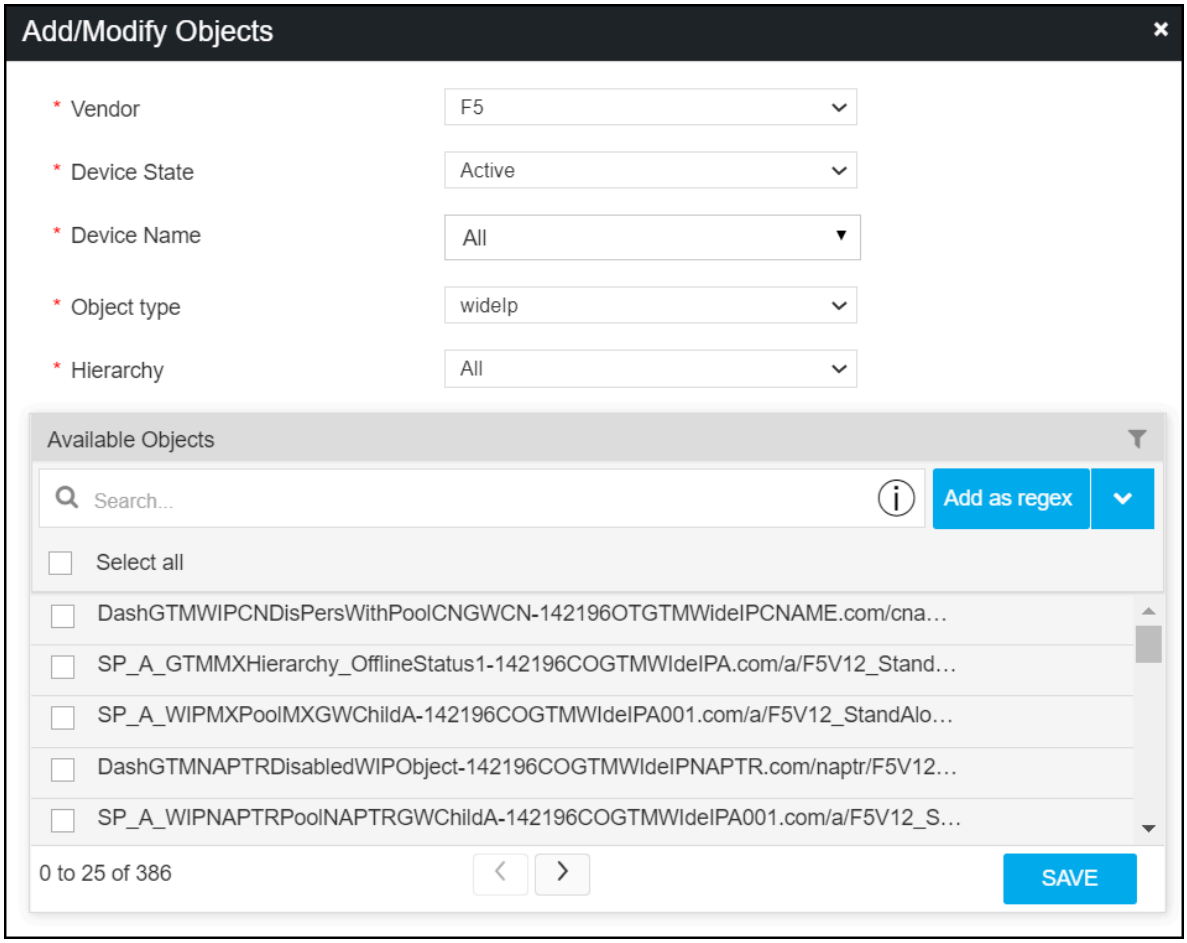
7. Modify the widget name, group name, and add a description by clicking the corresponding fields. It will then become a text-entry field that allows you to make changes.
8. Add objects to the group for the widget one of the following ways:

   - manually
   - using regex

9. Add objects manually to the group for the widget as follows:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. Click the **Add/Modify Objects** button to add the objects at the group level.



Field	Description
***Vendor**	Select the vendor from the **Vendor** dropdown list.
***Device State**	Select whether you want to include devices in the widget that have a status of Active, Standby, or All, where "All" means devices with a status of either Active or Standby.
***Device Name**	Select the device name whose objects you want to add to in the widget.
***Object type**	Select the kind of object you are adding. The options that appear in this field vary depending on the vendor you selected.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Hierarchy**	Select the required level of hierarchy that should be displayed in the dashboard. The options that appear in this field vary depending on the vendor and the object type you selected. The objects based on your selection will be displayed in the Available Objects table at the bottom of the screen. For example, A WideIP object will have the following hierarchical types, • **All** - WideIP along with all hierarchical objects like Pools and Pool members. • **All Pools** - WideIP along with the Pools. • **All Pool Members** - WideIP along with the Pool Members. • **WideipOnly** - WideIP only without any hierarchy. • **None** - No objects will be explicitly listed on the widget. Instead, only the state/status of the objects will be represented on the widget for monitoring.
**: Mandatory fields*	

b. Select the checkbox beside the object name and click **SAVE** to add the objects to the group.

10. Add objects via regex to the group for the widget as follows:

    a. Click the **Add/Modify Objects** button to add the objects at the group level. Select the object filter details such as **Vendor**, **Device State**, **Device Name**, **Object type**, **Hierarchy**.

    The objects that match the filter details are displayed.

    b. Add regex in the Search field, and then press Enter.

    The objects that match regex are displayed.

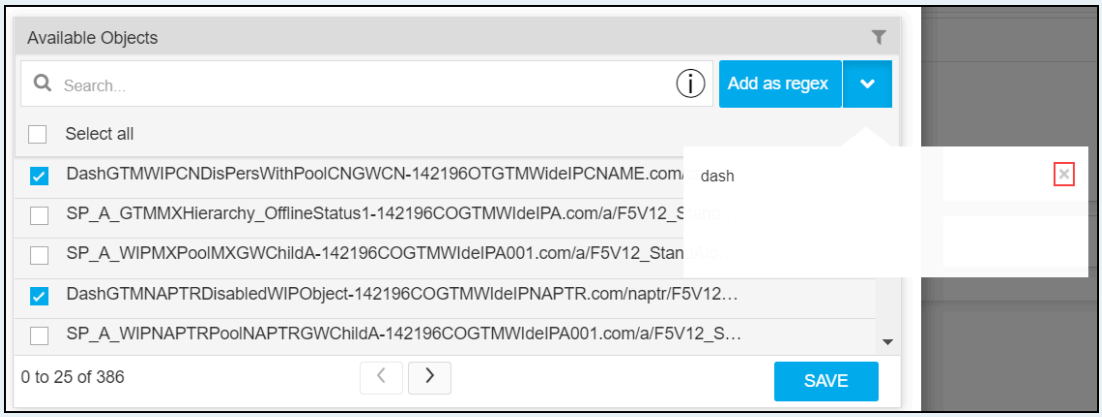
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

c. Click **Add as regex**.

The displayed objects are added to the group, the regex is added to the list, and it can be used whenever it is needed.

Note:

- You can add any number of regex. To view the list of regex, click the **Add as regex** dropdown option.
- The regex(s) is specific to a widget, dashboard, group, and filters.
- To delete a regex from the list, click the close button against the regex in the list.

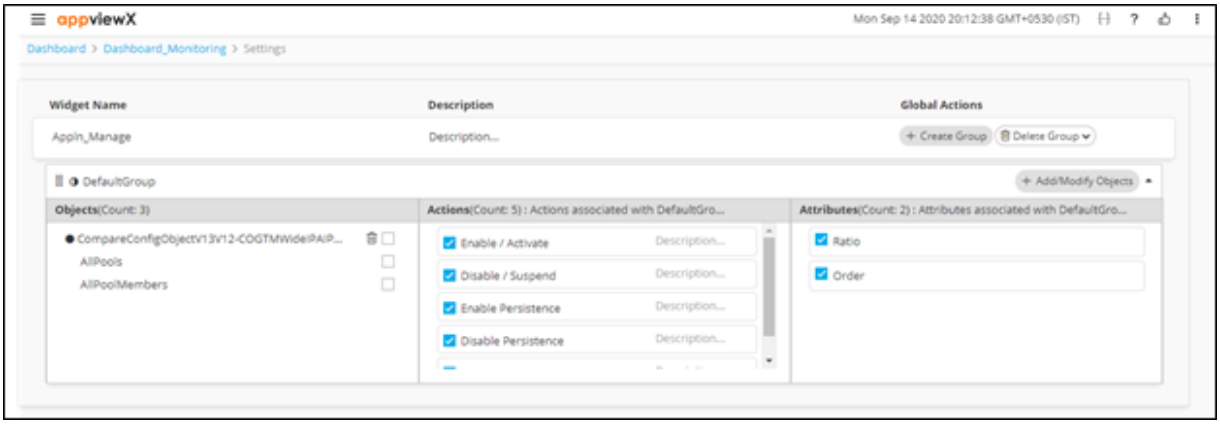


- To view sample regex, click the information button.
- When new objects are added and matched with the regex the objects are added to the widget automatically.

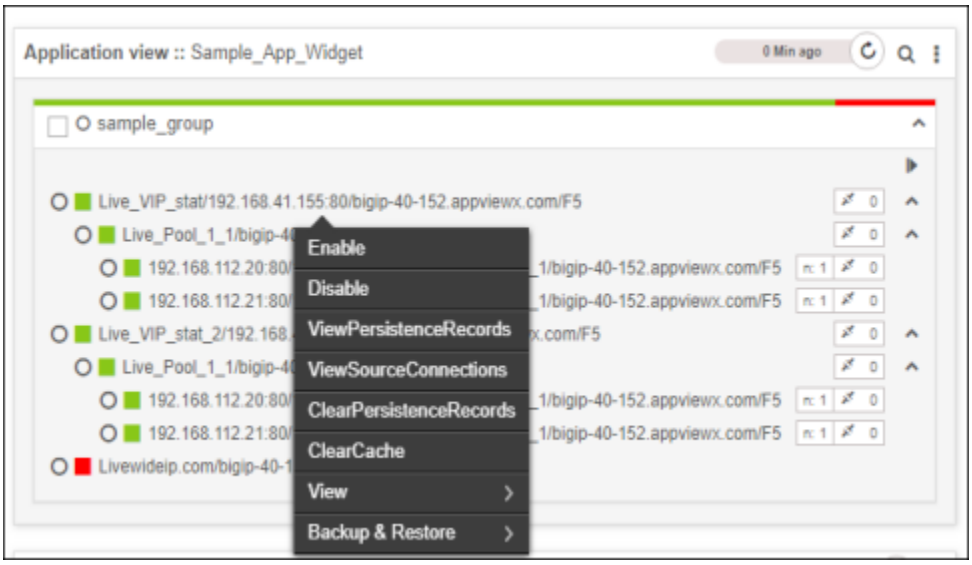
11. By default, access to the actions and attributes applies to all objects within that group. You can customize these permissions manually by selecting or deselecting permissions for individual objects.
12. For example, you might grant group members actions (such as Enable, Disable, ForceDown, and so on) and attributes (such as Ratio, Order, Weight, Connections, and Priority) permissions to one object in a group, but only Enable, Disable, and Ratio permissions to another object in the group and no permissions to objects in a sub-group of the group.
13. When the widget appears on the dashboard, the full list of actions you created can be accessed by right-clicking any object within the widget.
14. Click on the or (**Show/Hide Attributes**) button to view or hide the full list of attributes you created for a group.
15. When you have finished, click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The Application traffic can be managed and monitored.



16. The dashboard screen reappears, displaying the widget you just created.



17. To create a new group,

    a. Click the **Create Group** button under Global Actions.

    b. On the pop-up that appears, enter a group name to help the users identify it.

    c. Select the parent that you want to associate with the group from the dropdown list.

    d. Click **Save**.

    e. Repeat steps 7-15 to include the group in the dashboard.

18. To delete a group, complete the following steps:

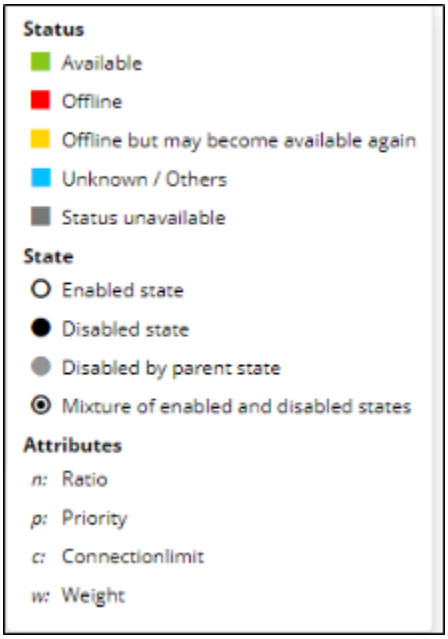
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. Click the **Delete Group** dropdown menu under Global Actions.

b. Select the groups that you want to remove from the widget and click **Delete**.

Monitor and Manage Application Traffic

The widget with the configured Application along with its hierarchy will be rendered onto your dashboard and start monitoring the Applications State/Status. The following Legend helps to identify the state/status definition, click the ⋮ (**Options**) button and select legend in the Command bar at the top of the widget.



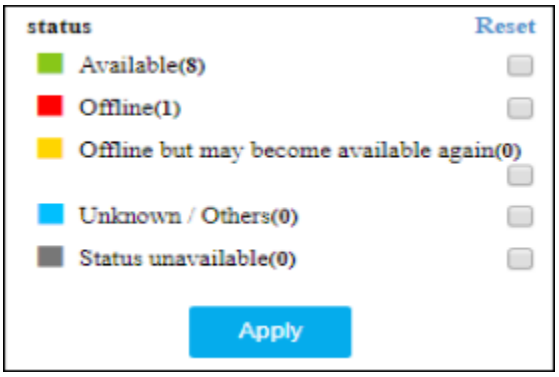
The applications can be filtered and sorted as follows:

Filter the Applications

To filter the applications based on the status:

1. Click the ⋮ (**Options**) button and select the ▼ (**Filter**) icon in the Command bar at the top of the widget.
2. In the **Status** field that appears, select the checkboxes beside the Status info that you would like to filter.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



The applications that match the filtered status appear on the screen.

Sort the Applications

To sort the applications in ascending/descending order:

1. Click the (**Options**) button and select the (**Sort**) icon > Ascending/ Descending in the Command bar at the top of the widget.
   The objects are arranged and displayed in the corresponding order.
2. Click the **Save Dashboard** icon to retain the sorting.

Monitor Near Real-Time State/Status

Monitor the near real-time state/status of your applications from the Application view widgets ensuring a seamless experience.

Note:
AppViewX recommends to enable Syslog-based configuration updates for near real-time monitoring. Refer to Platform Guide for enabling Syslog configuration.

You can enable the below options in the Dashboard settings for near real-time monitoring. For Dashboard setting, see Dashboard Setting section.

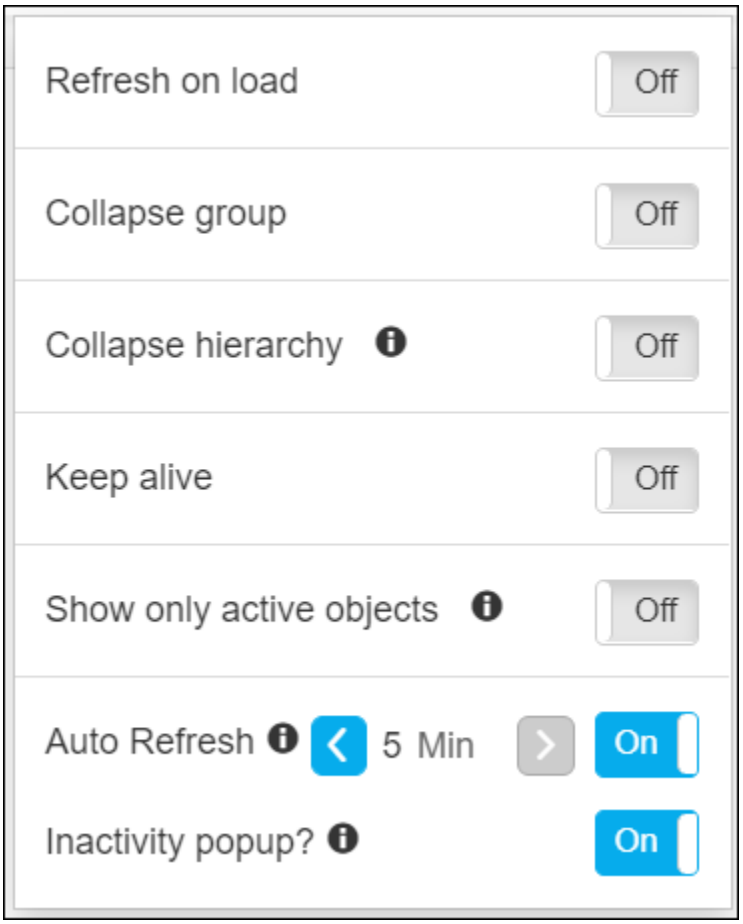
- **Refresh on load** - Enables Refresh on load from the (**Settings**) of the dashboard page to get the latest state/status information on loading the dashboard.
- **Last Refresh time** - Displays the last time the widget is updated with the latest State/Status information.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Refresh** - Click the **Refresh** button on the widget to perform on-demand State/Status information. Each Object contains a last refresh time on the tooltip, you can also trigger a refresh for a particular object from the tooltip.
- **Collapse group** - If this setting is enabled, each time the dashboard loads, all groups within widgets on the dashboard display in a collapsed state by default.
- **Collapse hierarchy** - If this setting is enabled, each time the dashboard loads, all object hierarchy within widgets on the dashboard display in a collapsed state by default.
- **Keep Alive** - This option is available in the dashboard **settings** . If this option is enabled, it ensures the session is not timeout by refreshing the dashboard every 15 mins.
- **Show only active objects** - If this option is enabled, only the active objects are displayed within the widget.
- **Auto Refresh** - Enables Auto-refresh from the (**Settings**) of the dashboard page with a particular interval (1 - 5 mins), which automatically refreshes the state/status information of the objects.
- **Inactivity popup** - When the **Auto Refresh** enabled the Inactivity popup option will also be enabled by default. Enabling the **Auto Refresh** and **Inactivity popup** options will notify the user about the inactivity (no mouse or keyboard events) of the screen through Confirmation popup, if there is no activity in the screen for last 15 minutes. In the Confirmation popup, user can choose **Yes** or **No** option. When user selects,
  - **Yes** - the **Auto Refresh** and **Inactivity popup** options will be disabled immediately and will result reduced the device calls.
  - **No** - the Confirmation popup will be closed without any change in the auto refresh and the popup will be notified after 15 mins. If the **Do not ask me again** checkbox is selected and the **No** button clicked, the Confirmation popup will not be notified until user enables and there will no change in the auto refresh.

> **Note:**
> After 15 minutes, in case user don't choose any option in the Confirmation popup, the popup will stay for 1 more minutes for user to choose option in Confirmation popup. Then the timer will be shown for 15 seconds. After that the Auto Refresh option will be temporarily disabled for this session due to inactivity.

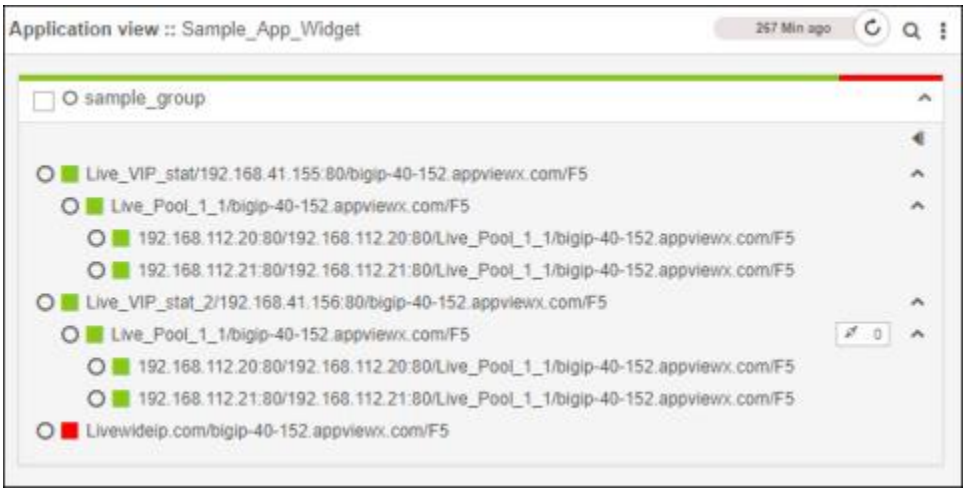
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Example, in the following sample application widget,

- all of the objects in the widget show that they are in an Enabled state-they all have hollow circles beside their names
- one of the InternalDMZ-Members objects is offline, so it shows a red square beside its name, indicating its Offline status.
- Three of the other objects display gray squares, indicating their Unlicensed/None/Failure States status.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- The temperature bar, which is the colored bar at the top of each group or object name, displays the overall status of all components within the widget. In the example above, note that the virtual server group shows a solid green temperature bar because all components under it are Available, whereas the Members group shows a mostly green bar transitioning to red, to indicate that some of the components within it are Unavailable. Hover your cursor over the color in the temperature bar to see the number of components that have the corresponding status.

Search for an Object in an Application View Widget

To search for an object in an Application view widget:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**..
2. If you have more than one dashboard, in the dashboard list, click the name of the one containing the widget you want to search.
3. In the widget Command bar, click the (**Search object**) icon.
4. In the search field that pops up, begin entering the name or number of the object you want to find.
5. As you enter text in the search field, the contents of the widget update dynamically to display only those objects that match the characters or numbers entered so far. In the example below, the only

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

objects remaining in the widget are those that contain the letters entered into the search field: best. All other objects have been filtered out.



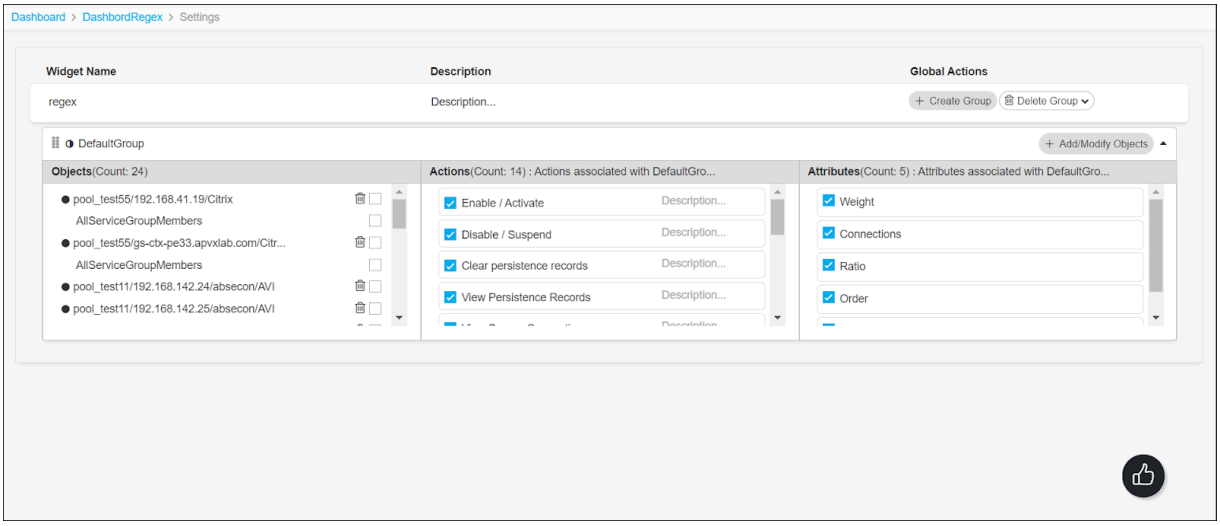
Modify an Application View Widget

The Application view widget contains the objects under the group. By default, the DefaultGroup is available. Either you can add objects to the DefaultGroup or create a new group of objects. The details of the Application view widget can be modified as needed.

To modify an Application view widget:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one containing the widget whose contents you want to make changes to.
3. Click the ⋮ (**Options**) button and select the ⚙ (**Settings**) icon in the Command bar at the top of the widget.

   The Settings screen appears:



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. You can modify the **Application view** widget as follows:

Application View Widget - Field and Description Table

Field	Description
Widget Name	Update the name of the widget.
Description	Update the description of the widget.
Global Actions	There are two Global Actions. Click, •  to create a new Application view widget group. •  to select the desired group from the drop-down list and delete.

Add/Modify Objects	• Click  to add and modify an object.
	• In the **Add/Modify Objects** popup, provide the object details, such as vendor, Device State, Device Nam, etc., search for objects, and/or filter the objects to get the desired objects.
	• Select or unselect the objects by clicking the objects checkbox.
	• Alternatively, you can add objects to the group by creating regex for the object details to get the desired objects. To add objects using regex,
	  • After providing the object details in the **Add/Modify Objects** popup, add the desired regex in the search field.
	  • Click the **Add as regex** button.
	  The objects that match the regex are added to the group and the regex is added to the regex list.
	  Note:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• You can add any number of regex. • To view the regex list, click **Add as regex** dropdown option. • The regex(s) is specific to a widget, dashboard, group, and filters. • To view sample regex, click the information button. • When new objects are added and matched with the regex the objects are added to the widget automatically. • Click **Save** to add the objects to the group.
Objects section	To delete an object, click against a particular object and then click **Yes** in the confirmation modal.
Actions section	Enable or disable action(s) for the objects by selecting the checkbox of the respective action. The enabled actions will be updated in the action list. By default, access to the actions applies to all objects within that group.
Attributes section	Enable or disable attribute(s) for the objects by selecting the checkbox of the respective attribute. By default, access to the attributes applies to all objects within that group.

5. The modified details will be updated in the Application view widget.

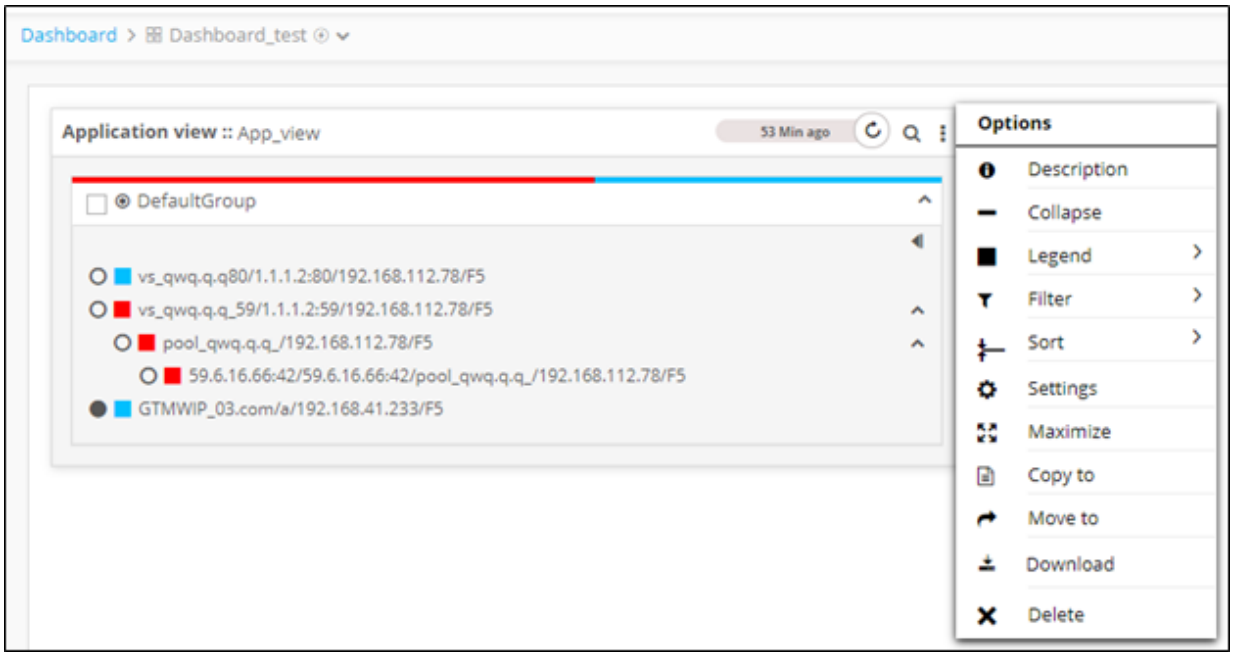
One-Touch Action Provisioning and Self Servicing

- The app-centric dashboard/widget allows you to self-service, monitor, and troubleshoot application issues in a single pane.
- One-touch/click application provisioning and Traffic management actions through a simplified RBAC.

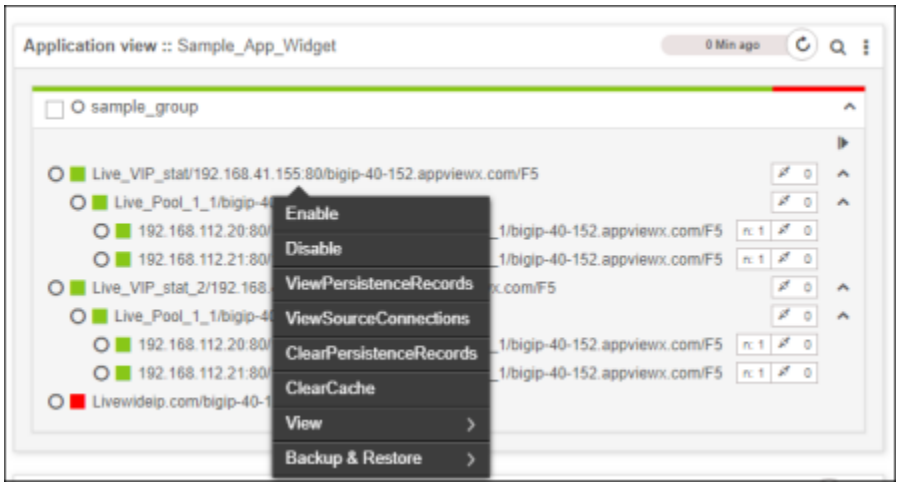
The configured objects along with their hierarchy are rendered in the form of a widget.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Users can expand and collapse the group by using the Expand or Collapse options from the (**Options**) button.



- Execute Server Rotation or Route/Swing traffic between your environments and perform various other operations on a single click.



The following actions can be performed (**Individual or bulk**) on an Application View widget by right-clicking the Group/Individual object and accessing the Actions menu:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Enable** - Enable an object
- **Disable** - Disable an object and terminate all active connections
- **Graceful disable** (AVI devices only) - Disable an object only when all the currently active client connections are closed by either the server or the client
- **Drain** (NGINX only) - To put the server into the draining mode
- **Backup & Restore**
  - **Backup device** - Create a backup of the device associated with the object. For more details refer to the Create a Device Backup Group section of this guide.
  - **Restore object** - Restore object configuration to a previous state. For more details refer to the Restore an Object section of this guide.
  - **Compare** - Compare current and/or archived configurations of similar objects or across devices to ensure compliance. For more details, refer to the Compare ADC Objects section of this guide.
- **View**
  - **View graph** - View the timeline statistics of an object. For more details, refer to the View Timeline Statistics for an Object section of this guide.
  - **View config** - View the current configuration of all levels of the device object. For more details, refer to the View Configuration Details section of this guide.
  - **View log/history** - View the log history of the object.
  - **View alerts** - View any alerts related to the object.
- **Clear persistence records** (F5 devices and Citrix SLB Virtual Server only) - Clear the persistence records for Virtual Server (VIP) and pools
- **View persistence records** (F5 devices only and Citrix SLB Virtual Server) - View the persistence records for Virtual Server (VIP) and pools
- **Advanced**

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Enable/Disable persistence** (F5 devices only) - Turn on or off the tracking and storing of session data, which is used to ensure that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions.
- **Enable/Disable all** (F5 devices only) - Turn on or off the object available across all the pools.
- **Forcedown all** (F5 devices only) - Force shutdown of the object available across all the pools.
- **Enable/Disable Proxy** (Cloudflare devices only) - Turn on or off the routing of traffic through an intermediary server. When enabled, the proxy hides the origin server for enhanced security and performance, and provides features like caching and traffic filtering. When disabled, traffic is sent directly to the origin server without these added protections or optimizations.

- **View source connections** (F5 devices only) - View the source connection IPs for Virtual Server (VIP) and pools
- **Forcedown** (F5 devices only) - Force shutdown of the object
- **FD clear active connections** (F5 devices only) - Clear active connections to the object
- **View Topology** - Display the topology view of the selected object in a new window

The following are the additional attributes that can be monitored on the widget,

- **Ratio** - To view the ratio of the pool member from widget level
- **Weight** - To view the real server weight for the applicable objects from widget level

Actions on HA Devices and Device Failover

AppViewX recommends to enable Syslog-based configuration updates to reflect the device flip over on a near real-time basis.

To enable the Syslog based configuration updates:

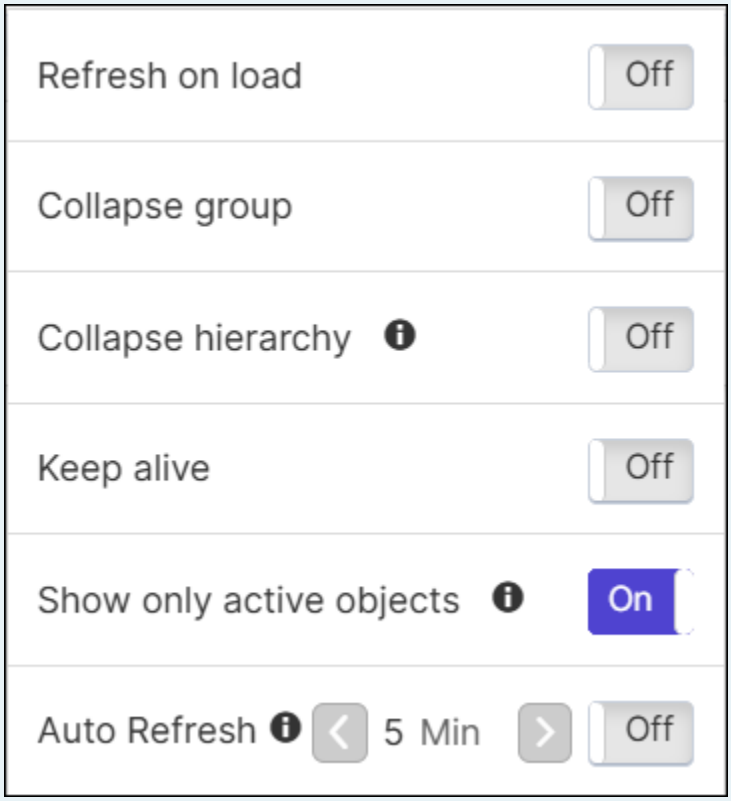
- Enable **Show only active objects** option from the **Dashboard settings**.

  It ensures the widget is always updated with the latest active device objects whenever a device failover is identified.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

By default, this option is enabled.



• To disable this option, click the ⚙ (**Settings**) icon on the dashboard page.

Note:

In addition to this, any actions that are executed on an application hosted on a HA device will always be reflected on the current active device to ensure the active devices are up to date and no changes are overwritten during device synchronization. AppViewX also ensures the Active and Standby devices are synchronized at a regular interval.

Following are other actions that can be performed by clicking the **Options** on the widget:

• Maximize

• Minimize

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Copy to - a copy of the widget can be placed from one dashboard to another dashboard by selecting the (**Copy to**) icon under (**Options**).
- Move to - any widget can be moved from one dashboard to another dashboard by selecting the (**Move to**) icon under (**Options**).
- Download - contents of an Application view widget can be downloaded in .csv format to your local system by selecting the (Download) icon under (**Options**).
- Delete - a group, object, or action can be deleted from an Application view widget by selecting (**Options**) > (**Settings**) > **Groups/Objects/Actions**, and then click **Delete**.

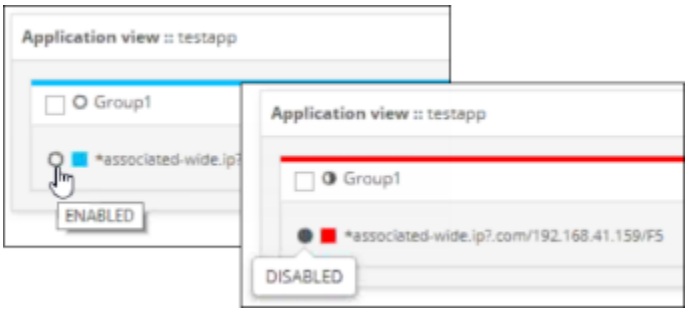
Note:
Every Group must have at least one action associated with it, so if there is only one action listed, the Delete button will not be enabled.

Enable or Disable Objects (appl.services) Displayed in a Widget

Application teams can quickly perform provisioning, server rotation as part of a standard change window by enabling or disabling the objects.

Note:
This process can be integrated into ITSM change management.

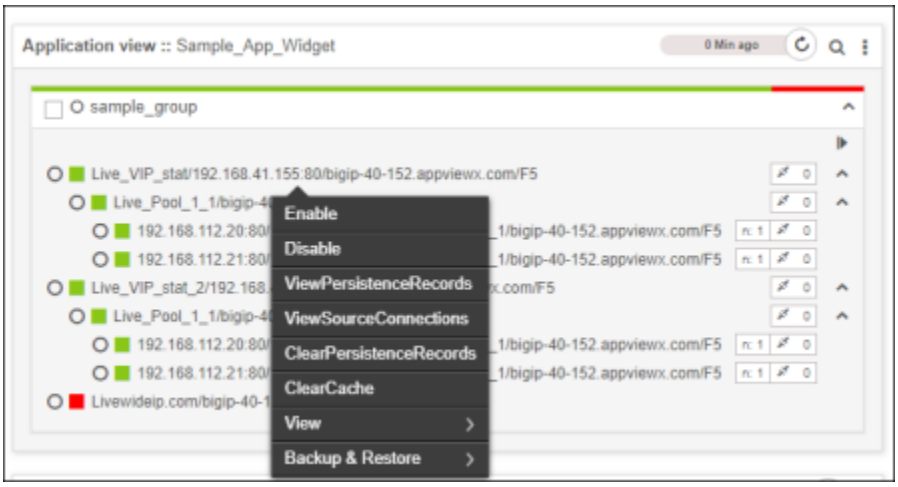
Objects that are enabled within a widget contain an open circle beside their names and display the word ENABLED when you hover your cursor over the circle. Objects that are disabled contain a filled black circle beside their names and display the word DISABLED when you hover over them.



To enable or disable objects displayed in a widget:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If the specific dashboard you want to access is not displayed on the screen, move your cursor to the breadcrumbs field of the current dashboard.
3. Click the current dashboard name.
4. In the dropdown list that appears, click the name of the dashboard you want to switch to.
5. When the dashboard opens, locate the widget that contains the object whose status you want to change.
6. Right-click on the object and select **Enable** or **Disable** from the dropdown menu that appears.



7. On the **Confirmation** screen that pops up, enter comments relating to the enable or disable action, then click **Yes**.
8. Click the ↻ (**Refresh**) icon in the widget Command bar to see the updated view of the widget.

Perform Bulk Actions on Objects in a Widget

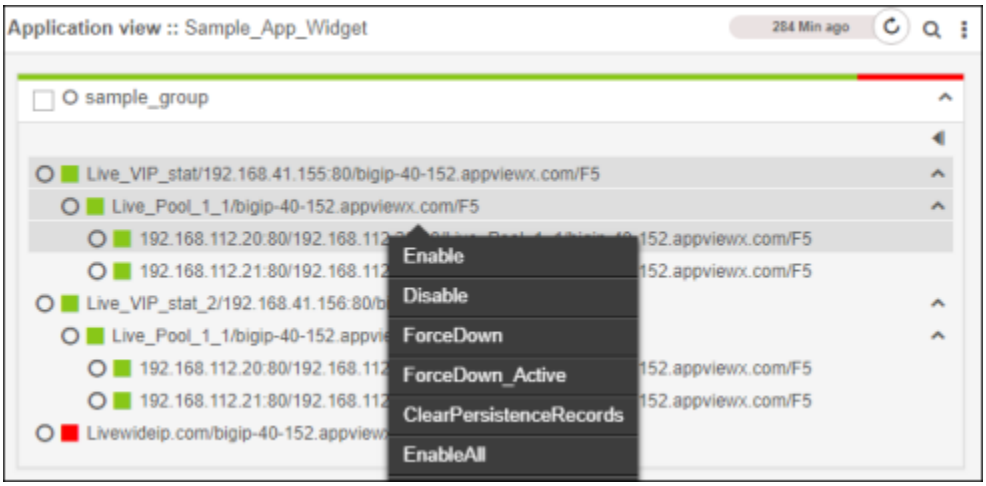
The Dashboard module allows you to perform the same action on multiple objects within the same widget simultaneously.

To perform bulk actions on objects in a widget:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If the specific dashboard you want to access is not displayed on the screen, move your cursor to the breadcrumbs field of the current dashboard.
3. Click the current dashboard name.
4. In the dropdown list that appears, click the name of the dashboard you want to switch to.
5. When the dashboard opens, locate the widget containing the objects that you want to perform bulk actions on.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

6. Click on each object within the widget that you want to perform a bulk action on. If you want all objects in the group to be included, select the check beside the group name.
7. Right-click anywhere on the widget and select the action you want to perform from the dropdown menu.



8. On the **Confirmation** screen that pops up, enter comments related to the action you are performing, then click **Yes**.
9. Click the (**Refresh**) icon in the widget Command bar to see the updated view of the widget.

Force LTM Servers Offline Within a Widget

During server maintenance, it is sometimes necessary to force a Local Traffic Management (LTM) server within a widget. When this is done, the server is still able to serve existing active connections, but no new connections are processed.

To force an LTM server offline within a widget:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If the specific dashboard you want to access is not displayed on the screen, move your cursor to the breadcrumbs field of the current dashboard.
3. Click the current dashboard name.
4. In the dropdown list that appears, click the name of the dashboard you want to switch to.
5. When the dashboard opens, locate the widget that contains the LTM server you want to force offline.
6. Click the LTM server name in the widget, then right-click on the server name, and then select **Forcedown** from the dropdown menu that appears.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



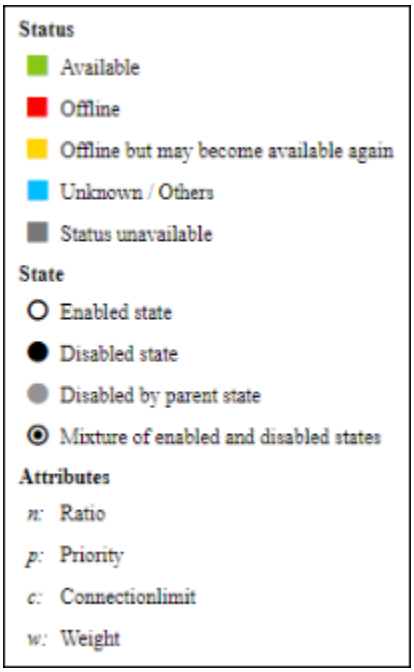
7. On the **Confirmation** screen that pops up, enter comments relating to the force-down action, then click **Yes**.
8. Click the (**Refresh**) icon in the widget Command bar to see the updated view of the widget.

View the Different Statuses and States for a Widget

To see a list of all of the statuses an Application view or Traffic grid widget can have:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. Select a dashboard that has an Application view or Traffic grid widget.
3. Click the (**Options**) button and select (**Legend**) icon in the Command bar at the top of the widget. A legend appears, listing each of the possible widget statuses and states. For Application view widgets, there are five possible statuses and four possible states, as shown below.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



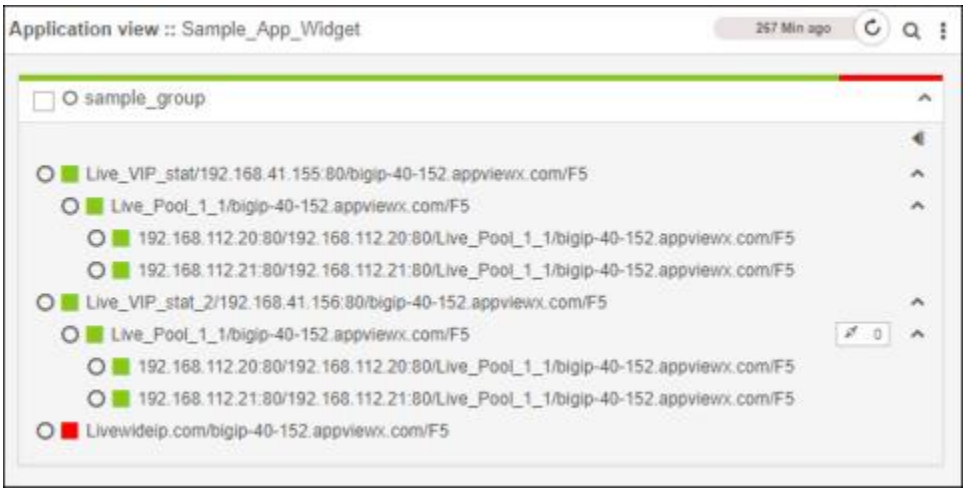
4. The list for Traffic grid widgets is larger, with a total of 9 possible statuses and 6 possible states, as shown below:



5. Using this legend, you can tell at a glance the current status and state of each component of the widget. In the Application view widget example below, all of the objects in the widget show that they are in an Enabled state-they all have hollow circles beside their names-but one of the InternalDMZ-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Members objects is offline, so it shows a red square beside its name, indicating its Offline status. Three of the other objects display gray squares, indicating their Unlicensed/None/Failure States status.



6. The temperature bar, which is the colored bar at the top of each group or object name, displays the overall status of all components within the widget. In the example above, note that the virtual server group shows a solid green temperature bar because all components under it are Available, whereas the Members group shows a mostly green bar transitioning to red, to indicate that some of the components within it are Unavailable. Hover your cursor over color in the temperature bar to see the number of components that have the corresponding status.

Traffic Statistics Widget - Application Traffic Monitoring

With the Traffic statistics widget, you can monitor the live and historic statistics of devices/ objects including the number of requests being received, the load on each of the load balancers, and peak request times. The widget helps to perform an in-depth analysis of your network traffic. It will benefit traffic monitoring to track down network problems and use the information to guarantee bandwidth and quality of service for business applications.

You can perform the following actions:

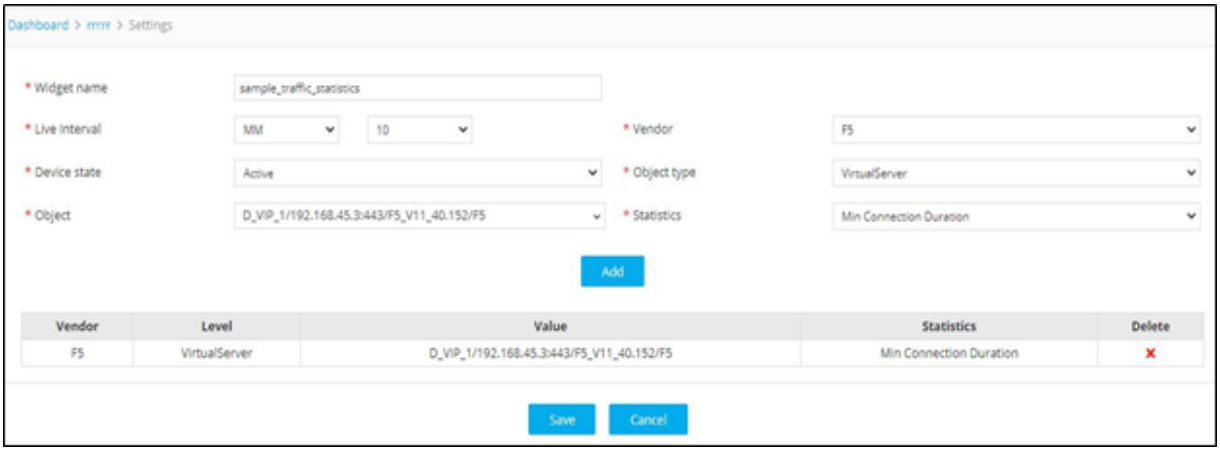
- Configure traffic statistics widget
- Monitor live and historic traffic flow

Configure Traffic Statistics Widget

To configure a Traffic statistics widget on your dashboard:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. If you are creating a Traffic statistics widget as part of creating a new dashboard, complete the steps in the Create a Dashboard topic above, then jump to Step 5 below. If you are creating a Traffic statistics widget for an existing dashboard, click the  (**Add widget**) button in the Command bar of the dashboard.
2. On the **Create widget** screen that pops up, select **Traffic statistics** as the widget type.
3. Enter a name for the widget.
4. Click **Create**.
5. On the **Settings** screen that appears, enter the time interval in minutes and seconds for collecting statistics from the devices.



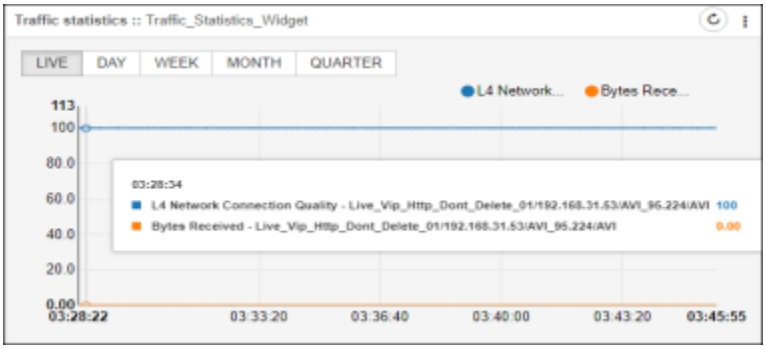
Statistics Tab - Field and Description Table

Field	Description
***Live Interval**	Set how often you want to collect the live performance statistics of an object. To configure the minimum Auto Refresh time interval, see Statistics.
Vendor	Select the vendor whose devices you want to collect statistics for A10, AVI, Citrix, or F5.
***Device state**	Select whether you want to include devices in the widget that have a status of Active, Standby, or All, where "All" means devices with a status of either Active or Standby.
***Object type**	Select the type of object you are adding. The available options in this field depend on the selected vendor.

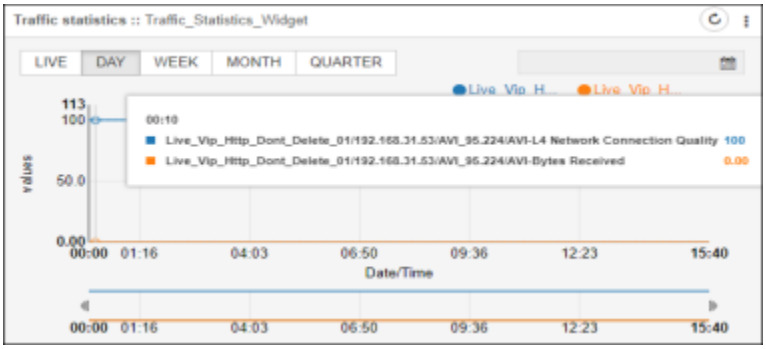
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Object**	Select an object you want to gather statistics for. The displayed list of objects varies depending on the selected object type.
***Statistics**	Enter the kind of statistics you want to gather. The list of statistics varies depending on the object you selected in Step 9 and can include both live and historical data within the same widget.
**: Mandatory fields*	

6. Click **Add** to add the object to the widget.
7. Repeat steps 5-6 for each vendor, device, and object you want to include in the widget.
8. When you are done adding objects, click **Save** to create the widget.
9. The dashboard screen reappears, displaying the widget (Live performance statistics) you just created.



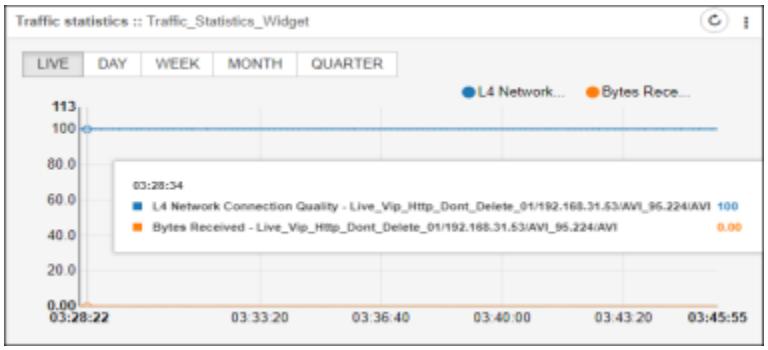
10. The historic performance statistics will be collected and displayed based on the time interval you configured in the **Settings** > **ADC** > **Statistics**. For detailed information, refer to the Statistics Settings section in the ADC Settings topic of this guide.
11. Click on Day/Week/Month/Quarter tab within the widget to view the historic performance statistics.



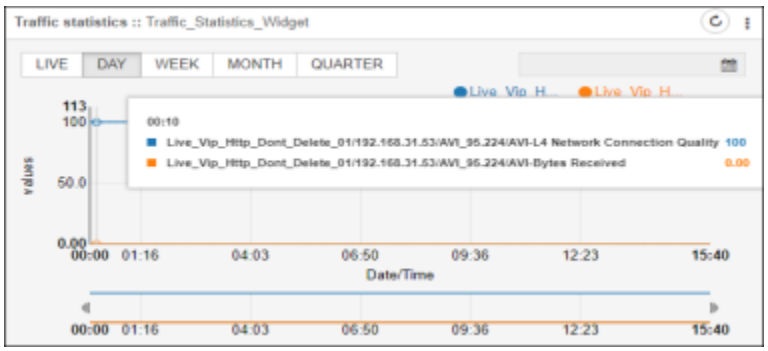
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Monitor Live and Historic Traffic Flow

- The line graph represents the Live, Day, Week, Month, and Quarter data.
- The Live tab will collect the real-time statistics from the device and plot as per the interval.
- Refresh on the widget to get the latest data on-demand basis.



- The historic statistics interval should have been configured globally in the Settings menu. To configure the statistics interval, see the Settings section.
- The historical data will be collected on a regular interval, aggregated, and plotted in the form of line graphs under respective tabs.
- Click on Day/Week/Month/Quarter tab within the widget to view the historic performance statistics.
- You also have the provision to pick a date or select a range using the sliding bar and narrow it down to a particular timeline.
- Hover over the graph to get the statistics value of a particular object.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Traffic Grid Widget

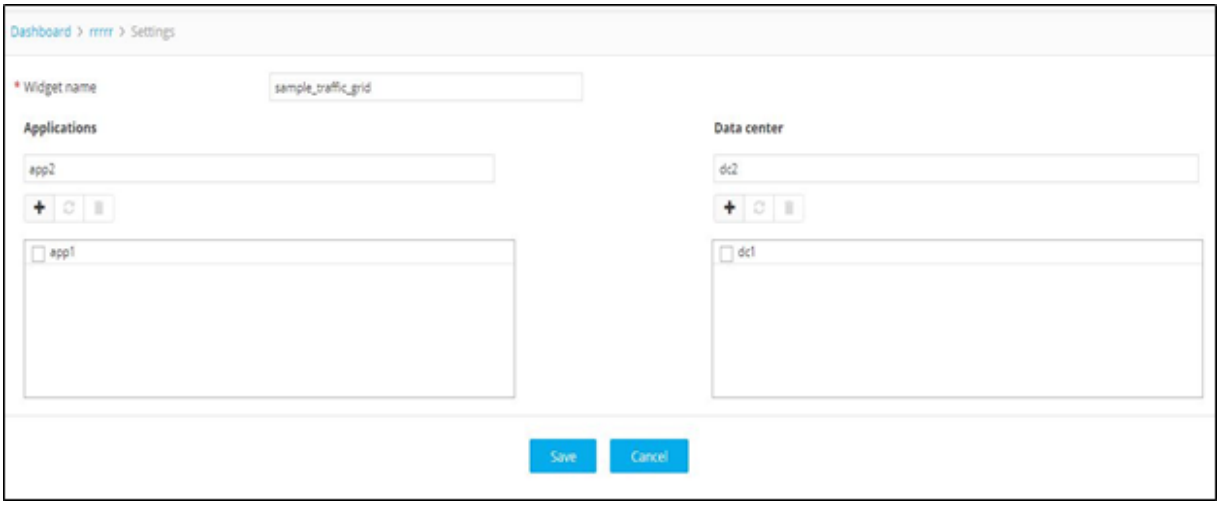
With this widget, you can monitor and control the amount of application traffic that flows across the various data centers hosting the application. Rules created within the widget define what percentage of traffic goes to which data center, making it possible to maximize dispersed resources from a single location.

You can perform the following actions:

Configure a Traffic Grid Widget

To configure a Traffic grid widget on your dashboard:

1. If you are creating a Traffic grid widget as part of creating a new dashboard, complete the steps in the Create a Dashboard topic above, then jump to Step 5 below. If you are creating a Traffic grid widget for an existing dashboard, click the + (**Add widget**) button in the Command bar of the dashboard.
2. On the **Create widget** screen that pops up, select the **Traffic grid** as the widget type.
3. Enter a name for the widget.
4. Click **Create**.
5. On the **Settings** screen, enter the name of the application you want to monitor traffic for.
6. Click the + (**Add**) button to add the application to the list.
7. Enter the names of the data center you want to configure traffic flows for.
8. Click the + (**Add**) button to add the data center to the list.
9. Repeat steps 5-8 for each additional application and/or data center you want to include in the widget.
10. Click **Save**.

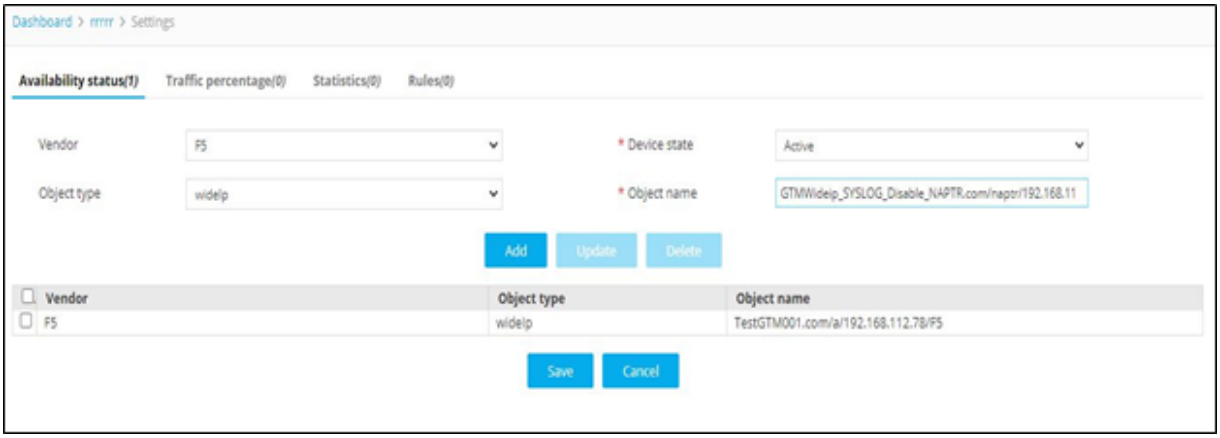


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The widget then appears as a grid on the dashboard. Note that the applications and data centers display NA instead of statistics because they have not yet been configured.



11. Click the ⚙ (**Settings**) icon inside each Application/Datacenter cell of the grid to configure it.
12. The **Settings** screen appears, with the **Availability status** tab displayed by default.



Availability status Tab - Field and Description Table

Field	Description
Vendor	Select the vendor whose object you want to monitor availability in each data center.
***Device state**	Select whether you want to monitor the traffic for a device that has a status of Active, Standby, or All, where "All" means devices with a status of either Active or Standby.
***Object type**	Select the type of object that you want to add. The available options in this field depend on the selected vendor.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Object Name	start typing to see a list of named objects whose traffic you can monitor. When you see the one you want, move your cursor over it. and then click on it.
**: Mandatory fields*	

13. Click **Add**.
14. Repeat steps 12-13 for any other objects you want to monitor.
15. After adding all the object types you want to monitor for availability, click **Save**.
16. Click the **Traffic Percentage** tab.



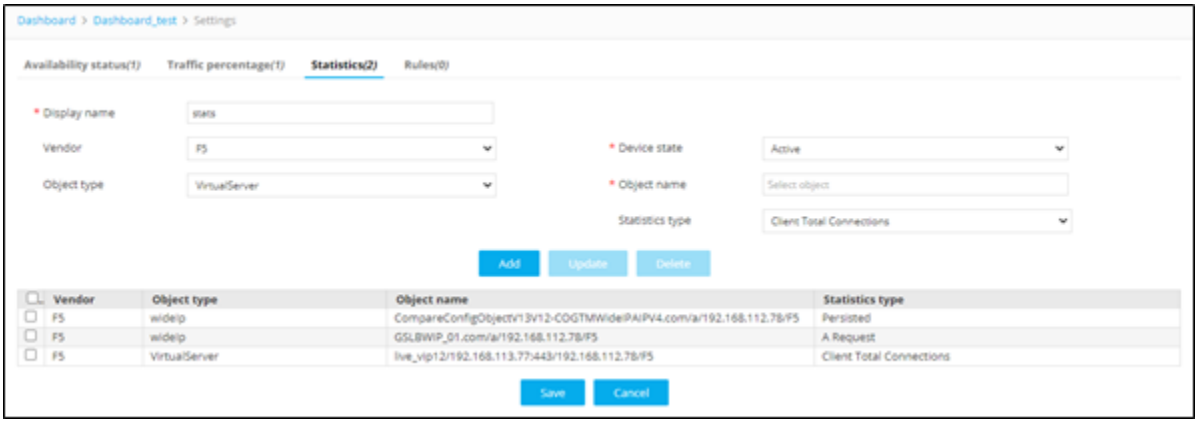
Traffic percentage Tab - Field and Description Table

Field	Description
Vendor	Select the vendor whose object you want to monitor availability in each data center.
***Device state**	Select whether you want to monitor the traffic for a device that has a status of Active, Standby, or All, where "All" means devices with a status of either Active or Standby.
***Object type**	Select the object that you want to add. The options are wideIp Pool, wideIp PoolMember, or ItmPoolMember.
Object Name	Start typing to see a list of named objects whose traffic you can monitor. When you see the one you want, move your cursor over it. and then click on it.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
**: Mandatory fields*	

17. Click **Add**.
18. Repeat steps 16 -17 for any other objects you want to monitor.
19. After adding all the object types you want to monitor for availability, click **Save**.
20. Click the **Statistics** tab.



Statistics Tab - Field and Description Table

Field	Description
***Display name**	Enter the text you want to pop up when a user hovers over the (**Pie**) icon within a table cell of the widget. Note that only 10 characters can be entered in the field, so the name must be brief, but descriptive. 
Vendor	Select the vendor whose object you want to generate statistics across all data centers.
***Device state**	Select whether you want to generate statistics for a device that has a status of Active, Standby, or All.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Object type**	Select the kind of object you are adding: wideIp PoolMember or ItmPoolMember.
Object Name	Start typing to see a list of named objects whose statistics you can generate. When you see the one you want, move your cursor over it and click it.
**: Mandatory fields*	

21. Click **Add**.
22. Repeat steps 20 -21 for any other objects you want to generate statistics for.
23. Click **Save**.
24. Click the **Rules** tab if you want to create rules that govern traffic percentages for the widget.

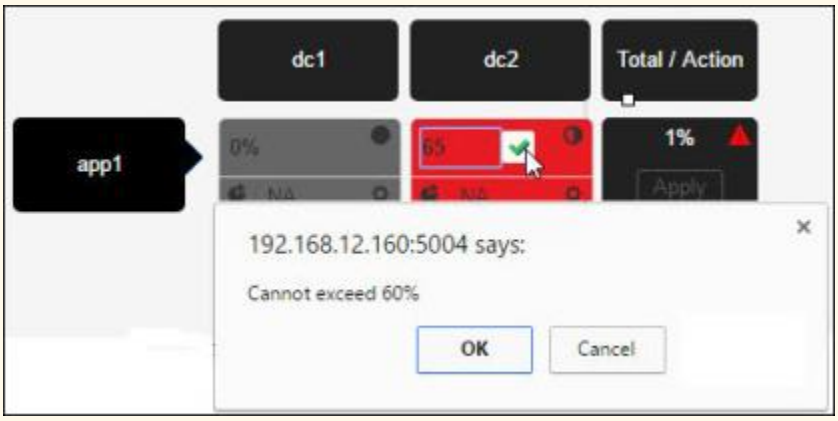


25. In the **Rule type** field, select one of the following options:

Warning:

Warning - A warning rule causes a popup text box to appear if the user violates the restrictions defined by the rule, but the user can still complete the action. For example, if you create a rule that says a data center cannot receive more than 60% of the traffic on a device and a user tries to set the percentage for that data center to 65%, a warning pops up on the screen when the user clicks the (Enter) icon in the table cell, as shown below. Despite the warning, the value 65 remains in the cell after the user closes the popup screen.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- **Restriction** - A restriction rule causes a popup text box to appear if the user violates the restrictions defined by the rule. In this case, the user is unable to complete the action. For example, if you create a rule that says a data center cannot receive more than 60% of the traffic on a device and a user tries to set the percentage for that data center to 65%, a restriction screen pops up on the screen when the user clicks the (Enter) icon in the table cell. When the user clicks OK to dismiss the popup, the percentage in the cell automatically reverts to its original value, erasing the user's entry of 65%.

- **Action** - An action rule causes an event that you define to happen when the traffic percentage for a data center reaches a level or range that you define. The two main event types are Enable and Disable, so when the level you set is reached, an object or set of objects you selected is automatically enabled or disabled.

26. After creating rules for the widget, click **Save**.
    For instructions on how to change the percentage values within the Traffic grid widget cells, refer to Change the Percentage Values Within a Traffic Grid Widget.

Monitor and Distribute Traffic Across Datacenter

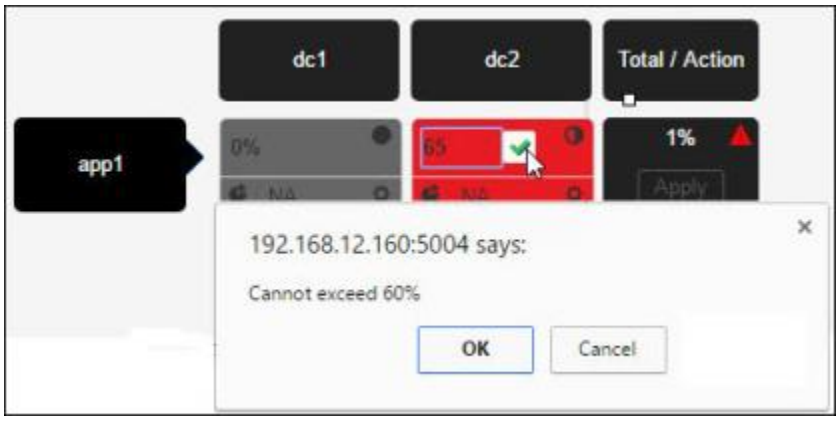
- The availability status (Up/Down/Offline etc.,) of the Application will be represented in appropriate color code along with the statistics value that is configured.
- The current Traffic percentage can be reviewed and distributed to other data centers accordingly. Distribute the traffic that equals to 100% and click Apply.
- Applying the changes will modify the ratio of the members on the end device.
- If the traffic percentage applied is not successful, the user will be indicated with a red triangle alert icon against the applications.
- During Maintenance, if a particular data center should not be receiving any traffic, it can be made 0 and the rest of the data centers can distribute the load.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- The traffic distributions are always reflected on current active devices.



- If a Rule is configured, before executing the traffic percentage changes, the rule condition will be validated.
  - In case of a Warning rule, the end-user will be prompted with a warning message to not proceed. Users can choose to Terminate or proceed with the changes.
  - In case of a Restriction rule, the end-user will not be able to proceed with the changes on exceeding the threshold limit.
  - When an Action rule is configured and if the traffic percentage meets the condition, the configured objects will be automatically Enabled/Disabled accordingly.



Change the Percentage Values Within a Traffic Grid Widget

To change the traffic percentages listed for each data center in a Traffic Grid widget:

1. Open the dashboard containing the Traffic Grid widget you want to update.
2. Click the current percentage value for the first data center.
3. The cell then becomes a text-entry field, as shown in the bottom-right cell in the image below.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter the new percentage in the text field.
5. Click the (**Enter**) icon to save your changes.
6. Repeat the process for each of the other data centers, ensuring that the total equals 100%.
   For an explanation of the different colors and symbols displayed in the cells, refer to View the Different Statuses and States for a Widget for a Widget.

Script Execution (SE) Widget

A Script execution widget saves script files on a local machine and provides easy access to maintain and execute script commands from within the widget. The script files (shell or python) are pre-written files existing in the AppViewX server. It allows the user to enable/disable actions that are performed through the action widgets and also, to write custom scripts that can be remotely run on the ADC devices.

You can perform the following actions on the widget:

- Configure a script execution widget
- Execution of script

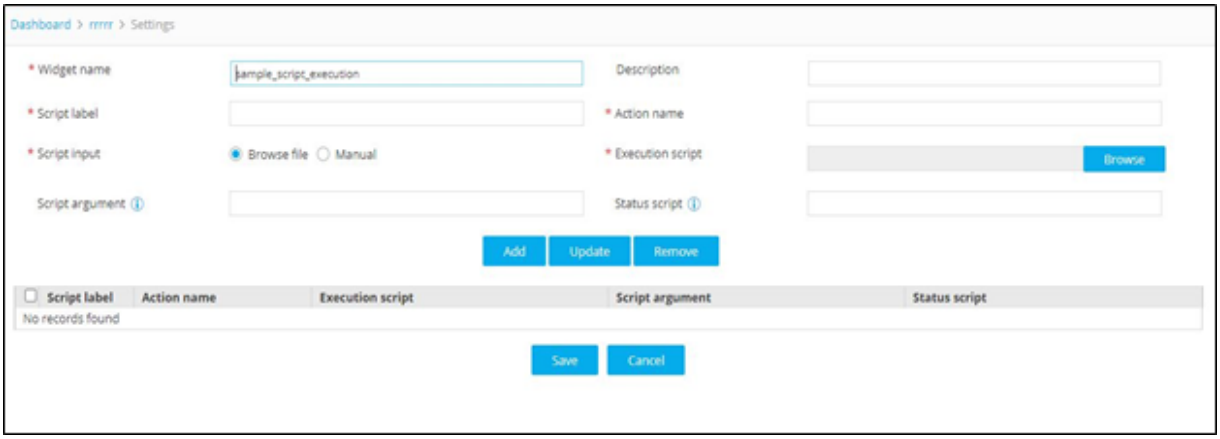
Configure a Script Execution Widget

To configure a Script execution widget on your dashboard:

1. If you are creating a Script execution widget as part of creating a new dashboard, complete the steps in the Create a Dashboard topic above, then jump to Step 5 below. If you are creating a Script execution widget for an existing dashboard, click the (**Add widget**) button in the Command bar of the dashboard.
2. On the **Create widget** screen that pops up, select **Script execution** as the widget type.
3. Enter a name for the widget.
4. Click **Create**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

On the **Settings** screen that appears,



Create Script Execution Widget - Field and Description Table

Field	Description
***Script label**	Enter a label for the script bar you will be adding to the widget. The label should help identify what sort of script you are adding so that it can be readily identified when viewed through the widget.
***Script input**	Select the **Browse file** or **Manual** radio button based on how you want to upload the scripts.
Script argument	In the **Script argument** field, enter any information for the script with a comma as delimiters. Upon script execution, the mentioned arguments are passed to the script provided that the script is capable of receiving arguments (through variables).
Description	Enter a description of what the script does so that it can be readily identified when viewed through the widget.
***Action Name**	Enter the action that you want to perform using the script execution widget.
***Execution script**	In this field, do the following:

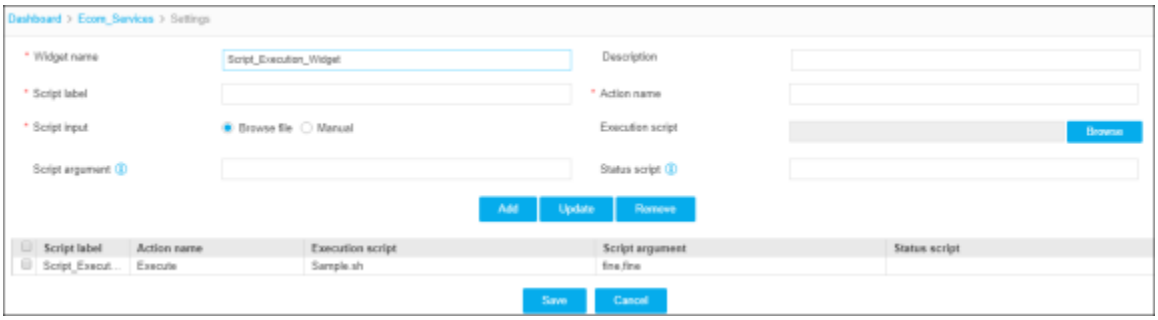
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description

	• If the *Browse file* radio button is selected in Step 8, click the **Browse** button and navigate to the file you want to add, then click **Open**. Below are the designated executable locations to be used for shell and Python files:
	  • For Shell files, #! /bin/sh.
	  • For Python files, #! /appviewx/dependencies/appviewx_addons/Python/bin/python.
	• If *Manual* is selected in Step 8, enter the full path of the script. These scripts will need manual intervention to execute.

Status script	Enter the full path of the script to update the color and status message of the bar. The status and result of the script are updated simply by loading or refreshing the widget.
**: Mandatory fields*	

5. Click **Add** to add the details in the table at the bottom of the screen.



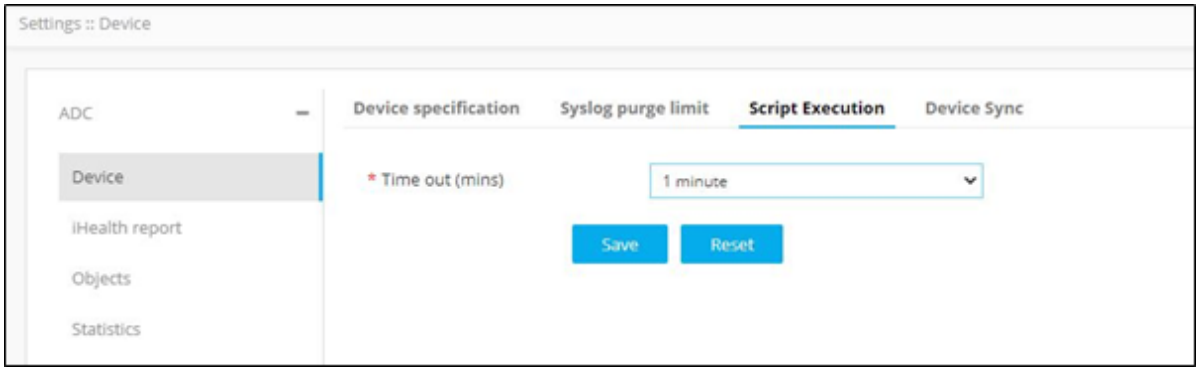
6. Repeat steps for each additional script you want to add.
7. Select the checkbox beside the script label column and then, click Save.
8. Configure script timeout to avoid prolonged script execution. To configure the script timeout,

   • Go to ☰ **Menu** > **Settings** > **ADC** > **Device** > **Script Execution** > **Time out**.

   • Mention the time out from 60 seconds to 5 minutes. The default timeout is 60 seconds (1 min).

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
For more detail, see Script Execution in the Device Settings.

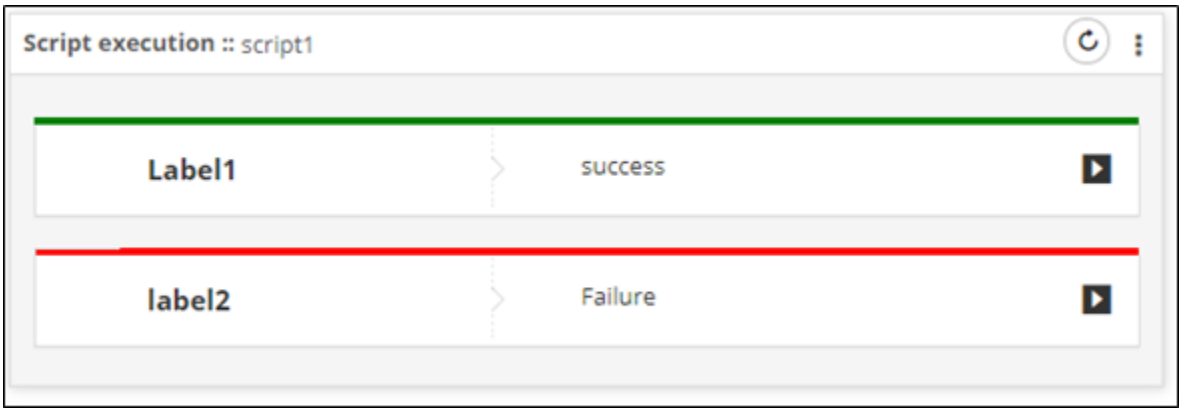


Execution of Scripts

On the successful configuration of widget for script execution, the scripts can be executed on a need basis.

To execute the script:

1. Click (**Play**) on the widget.



2. The color and message on the bar is determined by the following return message of the status script:

echo Color:Green,Response:"Pool failover_pool_web is up"
Color: Green - to update the color of the bar to green
Response:"Pool failover_pool_web is up" - to show the string inside "" as status

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Color: Red - to update the color of the bar to red

3. The status script will update or run on every refresh of the widget, to get the latest status.

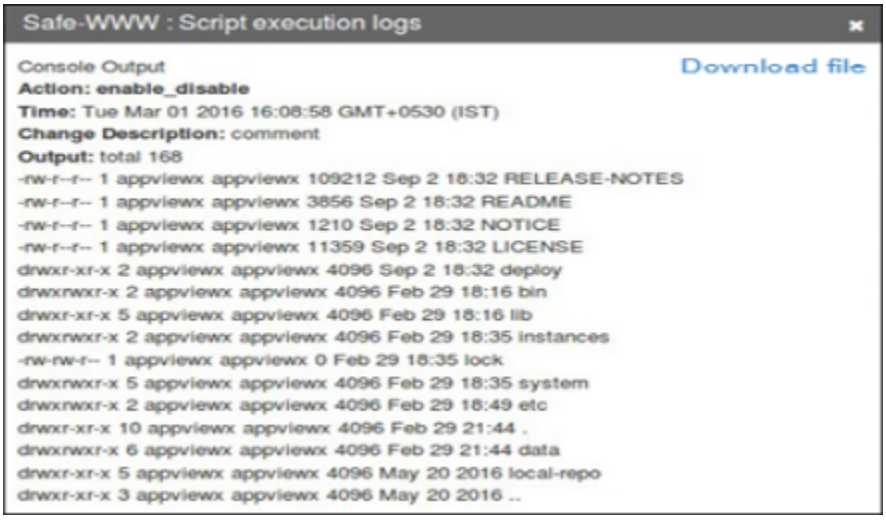
> **Note:**
> Frequent use cases like Device Memory check, CPU check, etc., can be executed quickly on demand, which will provide the run time data from the device.

4. To run the execution script, click the (**Play**) icon and select one of the following:

   • **Action name**: To internally trigger the execution from the AppViewX server CLI.

   • **View Logs**: To check the previous logs of the script that are executed.

5. When it finishes, a popup screen appears listing the script execution logs, as shown in the following example:



6. Click **Download file** if you want to download an output file.
7. The file is downloaded to your computer.
8. Navigate to the location where you want the file to go, then click **Save**.

Class Management Widget

A Class management widget allows you to view and modify the classes associated with iRules for devices. By grouping objects into classes and assigning actions to the classes, users are able to configure ADC devices within AppViewX. There is no need to access the ADC devices directly, which eliminates the chance of crashing a box with syntactically incorrect commands.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

iRule is a flexible feature that lets the user manage or customize their application traffic. Classes are usually defined within an iRule. String, Address, Integer are different kinds of classes. The following section explains data group/classes and their types.

Data Group/Classes

A data group is a simple group of related elements like a set of IP addresses of a class.

Data groups can be one of the following types:

- **Address**: An address data group consists of a collection of IP addresses.
- **String**: A string address group consists of a group of strings, such as *.jpg.
- **Integer**: An integer data group consists of a group of integers.
- **External File**: You have the option to store data groups in an external file. The benefit of storing data groups in an external file is the data does not need to be sorted by the system when it is loaded because it is stored outside of the **bigip.conf** file. The elements in the external data group file should be stored in comma-separated lists with a newline character after each entry. Refer to the *BIG-IP Configuration Guide* for more details about Data Groups.

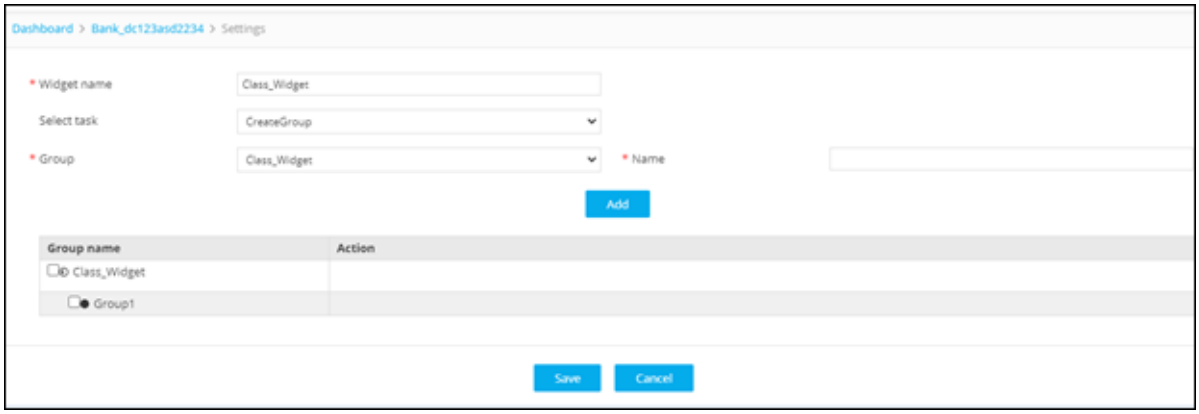
Configure a Class Management Widget

To configure a Class management widget on your dashboard:

1. If you are creating a Class management widget as part of creating a new dashboard, complete the steps in the Create a Dashboard topic above, then jump to Step 5 below. If you are creating a Class management widget for an existing dashboard, click the  (**Add widget**) button in the Command bar of the dashboard.
2. On the **Create widget** screen that pops up, select **Class management** as the widget type.
3. Enter a unique name for the widget.
4. Click **Create**.
5. On the **Settings** screen, configure Actions to any of the following as per the need:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **View class** - to configure view class, see Configure View Class section.
- **Modify class** - to configure modify class, see Configure Modify Class section.



Configure View Class (Internal/External)

View class action retrieves the configured class attributes from the device and allows to view the latest configuration.

To create a view class:

1. On the **Settings** screen, configure Actions for the view class.

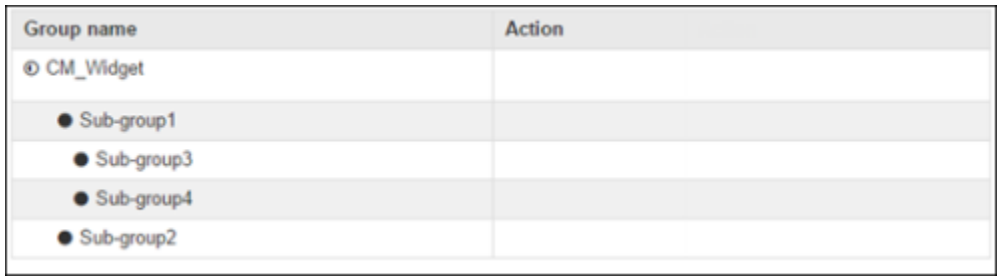
Create Group - Field and Description Table

Field	Description
Select task	Keep the field set to **CreateGroup**. The Group field automatically uses the widget name as the default group.
***Group**	Enter a name for the group.
**: Mandatory fields*	

2. Click **Add** to add the group to the widget.
3. (Optional) If you want to create a sub-group, enter another group name in the **Name** field and click **Add**.
   The new group is added under the original one.
4. (Optional) If you want to create a sub-group of a sub-group, select the sub-group in the Group dropdown list, then enter the sub-sub-group name in the **Name** field before clicking **Add**.

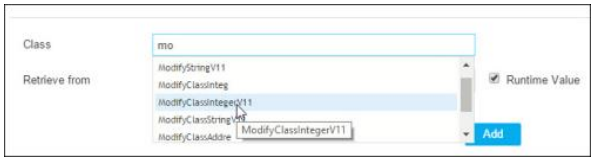
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. In the example below, CM_Widget is the group, Sub-group1 and Sub-group2 are created as sub-groups of CM_Widget, and Sub-group3 and Sub-group4 are created as sub-groups of Sub-group1.



6. After creating groups and sub-groups, select or enter the details.

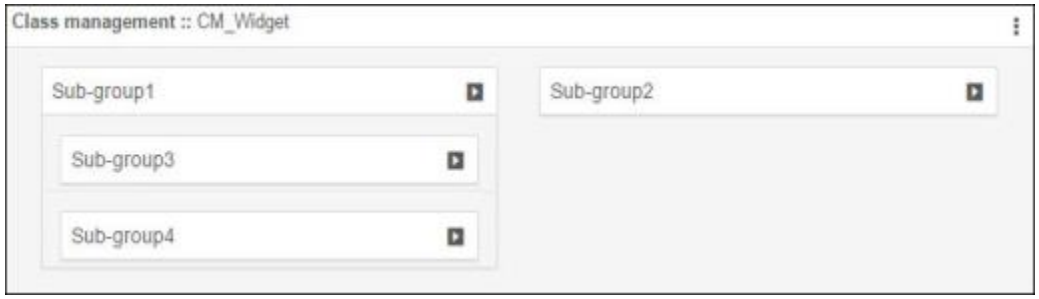
Create Actions - Field and Description Table

Field	Description
Select task	Select **CreateAction** dropdown list.
***Actions**	select **View Class**.
***Group**	select the group you are creating the action for.
***Name**	Enter the action a name that clearly identifies what action the user can take.
***Class**	Start typing the name of the class that you are creating a view or modify action for. As you start to type, all classes that match the characters you have entered so far appear in a list. Select the class you want to add.  **Note:** For F5 devices, the classes are controlled by the permission provided in the Resource section. Based on the permission given for the classes, they appear in the **Class** field dropdown list for the selection. If no permission has been given for the classes, they do not

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	appear in the **Class** field dropdown list. In case the permission has been removed for the classes while you manage a class in the dashboard, the "Permission Denied" notification will be displayed.
**: Mandatory fields*	

7. The **Retrieve from** field populates automatically with the location of the class that you selected.
8. Enabling the **Run time value** option, allows the end-user/consumer to change the class name during the execution.
9. Click **Add** to add the class to the group.
   The group and all sub-groups you created appear in a hierarchical structure within the widget.
10. Click the ▶ (**Actions**) icon to access the list of available actions for the corresponding group or sub-group.



Note:
If needed modify the configured Group/Action.

Configure Modify Class (Internal/External)

Modifying class action allows you to view and change the details of a class.

To create a modify class:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. On the **Settings** screen, create group for the modify class.

Create Group - Field and Description Table

Field	Description
Select task	Keep the field set to **CreateGroup**. The Group field automatically uses the widget name as the default group.
***Group**	Enter a name for the group.
**: Mandatory fields*	

2. Click **Add** to add the group to the widget.
3. (Optional) If you want to create a sub-group, enter another group name in the **Name** field and click **Add**.
   The new group is added under the original one.

Heatmap Widget

A Heatmap widget allows you to view statistics for managed, failed, and unresolved devices or device groups. Widget will collect the statistical data for Inprogress, queued, failed, and managed devices as long as the objects are available in the database. The heatmap reports are not viewed for unmanaged devices.

In the Heatmap widget, ADC device groups appear as color-coded blocks, with the colors representing the following:

- **Green** - All devices in the group are healthy
- **Red** - At least one device in the group is in a critical state
- **Gray** - Statistics have not been collected for the device group.
- **Orange** - One or more devices in the group have reached a warning limit

Copy a Widget to Another Dashboard

To copy a widget to another dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one containing the widget that you want to copy.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Click the  (**Options**) button and select the  (**Copy to**) icon in the Command bar at the top of the widget.
4. On the Copy widget screen that pops up, select the dashboard you want to copy the widget.
5. Click **Copy to Finish**.

Move a Widget to Another Dashboard

To move a widget to another dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one containing the widget that you want to move.
3. Click the  (**Options**) button and select the  (**Move to**) icon in the Command bar at the top of the widget.
4. On the Move widget screen that pops up, select the dashboard you want to move the widget to.
5. Click **Move to Finish**.

Delete a Widget

Deleting a widget removes it from the current dashboard, but does not remove it from any other dashboards it has been copied to.

To delete a widget:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one containing the widget you want to delete.
3. Click the  (**Options**) button and select the (**Delete**) icon in the Command bar of the widget.
4. On the confirmation screen that pops up, click **Yes**.
5. The dashboard refreshes and no longer displays the widget.

Custom Widget

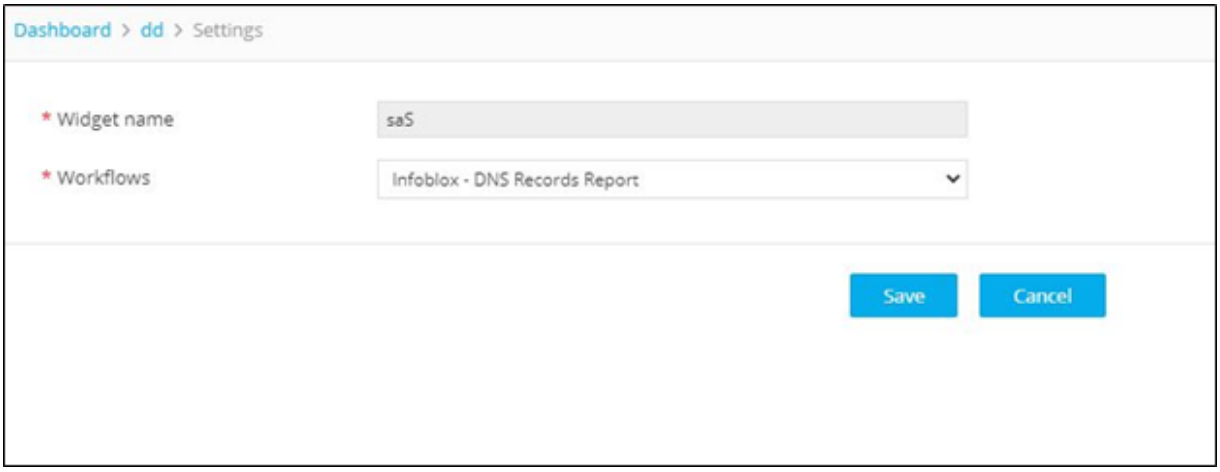
Custom reports are used for monitoring the usage of the device and application metrics, which are configured by existing workflows.

Configure a Custom Widget Using Workflow

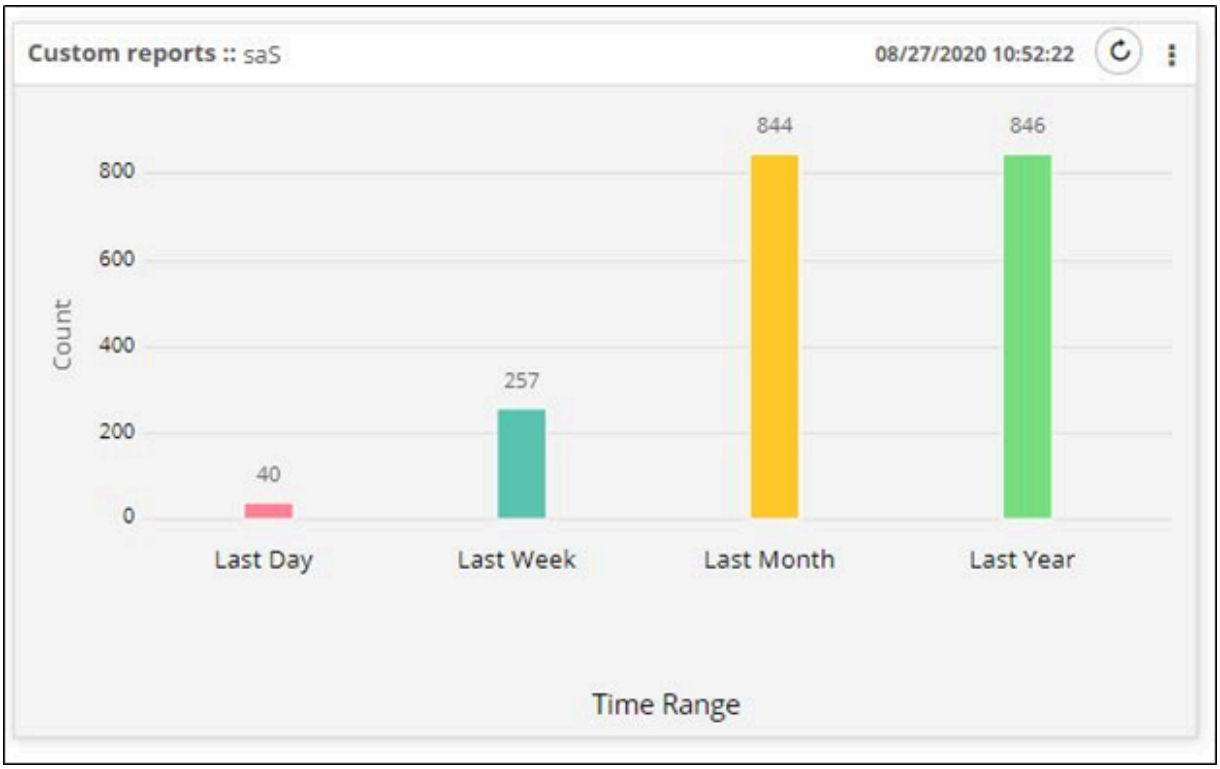
To configure a customreports widget on your dashboard:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. In the **Dashboard**, click the  (**Add widget**) button in the Command bar of the dashboard.
2. On the **Create** widget screen that pops up, select **Custom reports** as the widget type.
3. Enter a name for the widget, and then click **Create**.
   Custom reports configuration screen appears.



4. In the **Workflows** field, select the workflow that you want to monitor the usage of the device and application metrics.
5. When you are done configuring the Custom reports settings, click **Save**.
   he custom reports widget is displayed on the dashboard, with configured workflow reports.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Launch Automation from Application-Centric View through Rules

- Enforce business process and launch automation from App-centric views through the Rule configuration.
- Actions that are triggered from App-centric views will look for Rules configured and execute an appropriate workflow in the background.
- Upon executing any action, a request ID will be generated and a stage view will be displayed representing the action status.
- The request ID could be used to track progress at any point in time.
- Sample use cases
  - Service NOW integration for any action triggered from AppViewX.
  - Prevent users from taking all servers out of rotation as this might lead to an application-level outage.
  - Email notification for any Server Rotation action performed by the team member.

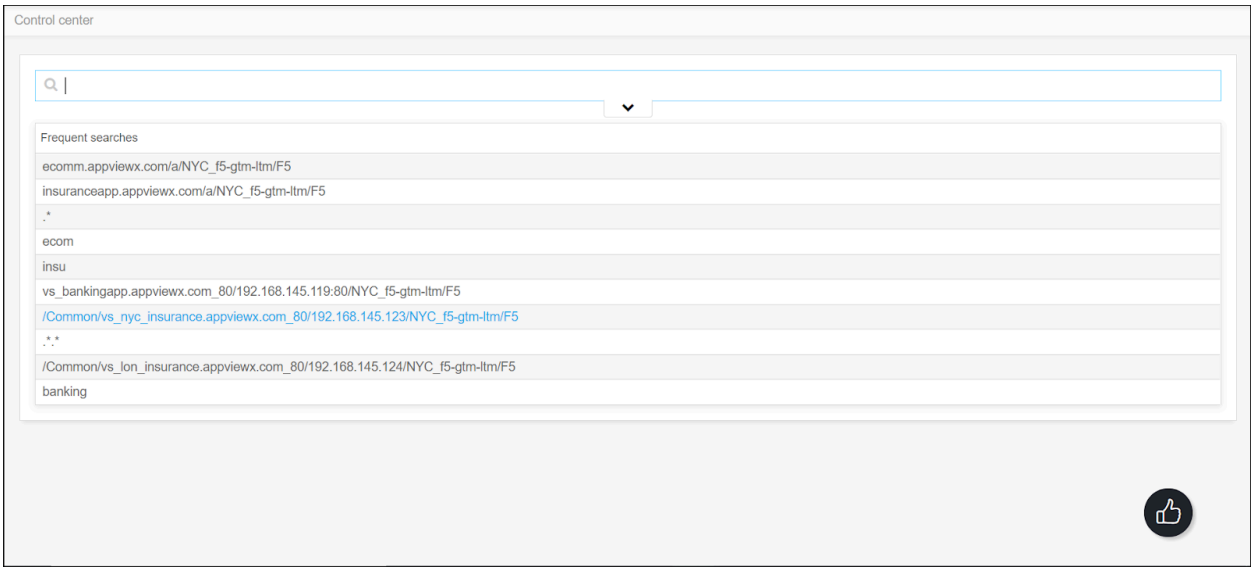


App Search

App Search - Overview

The App Search module displays a centralized object repository from the Control Center that allows you to search for and then monitor and manage all the entities, configurations, or objects of the ADC devices. Control Center provides a holistic NOC overview for app teams and network teams to quickly search and monitor application services based on RBAC in a visual intuitive topology view.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



In App Search module, the Control Center acts as a search engine inside the AppViewX to search and monitor any Application in your infrastructure. From the app search, any application-related details can be searched/found the complete infrastructure details and services supporting the application. The Control Center within the App Search module also provides better insights into the configuration, state, and performance of the application and helps to troubleshoot application outages more effectively.

To use the Control Center within the App Search module, the following prerequisites must be met:

- Each device you want to control must have been a managed entity in AppViewX.
- A role and resource must have been assigned with the appropriate device/object level that you want to control. However, for the orphan and secondary objects, global access must have been provided.

Run a Search

To run a search within the Control Center:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**..
2. In the App Search module, the Control Center search screen is displayed by default. The ways to search the ADC devices are:

Search Using Free Text Entries

This is the most common type of search. You can enter text in the search field and press Enter on your keyboard. The following search features and functionalities are supported:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Case insensitive keyword and string matching. For example, the search result for F5 would be **f5_86.appviewx.com**.
- Exact match search strings through the use of double quotation marks around search terms. For example, "**gs-f5-pe21.lab.appviewx.net**".
- Entering only the application name in the search field is enough to find all the hierarchical details related to it.
- Boolean AND and OR operators:
  - **AND Operator** - search results contain both terms that existed in the search query. For example, **gs-f5-pe21.lab.appviewx.net AND default**.
  - **OR Operator** - search results contain one or both terms that existed in the search query. For example, **gs-f5-pe21.lab.appviewx.net OR default**.
  - **AND** and **OR** - When a Boolean AND operator and an OR operator exist in the same search string, the AND operator is executed first by default. For example, **gs-f5-pe21.lab.appviewx.net AND gs-f5-pe51.lab.appviewx.net OR default.**

If parentheses appear in a Boolean query, the query components within the parentheses are executed first, followed by the query components outside the parentheses. For example, **(gs-f5-pe21.lab.appviewx.net AND default) OR 3.4.5.6**.

For example, typing only the application name in the search bar is enough to find all the hierarchical details related to it:



Search Using Frequent Search Links

The App Search screen displays a list of frequent searches immediately below the search field. Click any of the items in the Frequent searches list to search the AppViewX platform for that word, phrase, or character set.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

In the ADC search results screen, click the vertical ellipse button next to the search bar and select **Frequent Searches** to view a list of frequent searches.



Search Using Predefined Search Keys

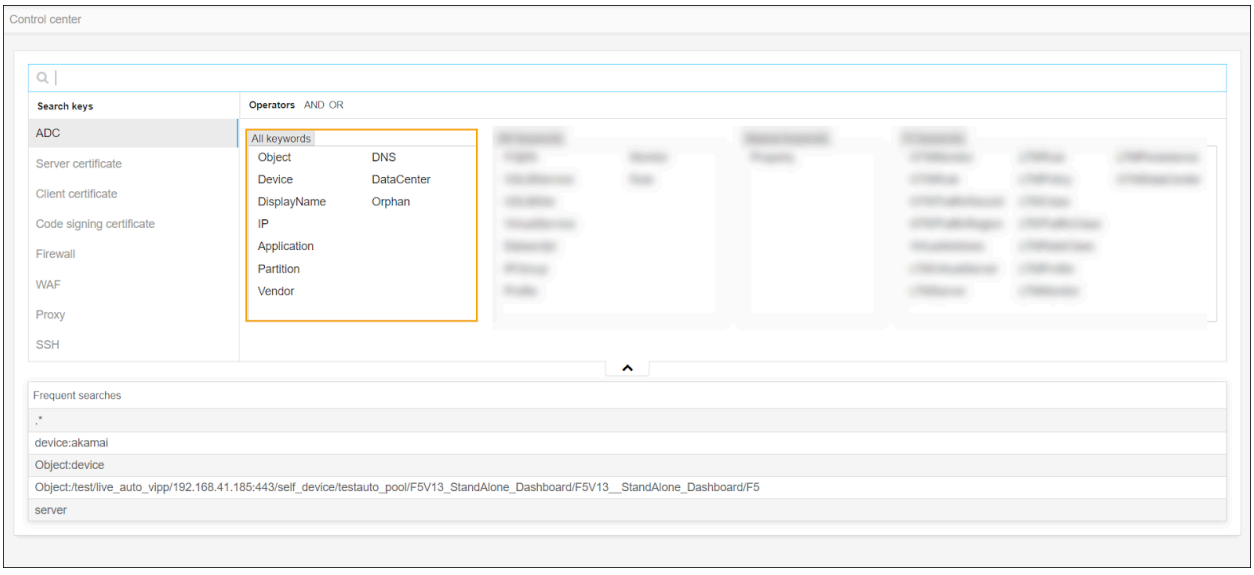
AppViewX provides a set of predefined keys specific to each ADC vendor to help the search operation. The types of predefined keywords are:

- All keywords
- Vendor-specific keywords
- Logical operator with keywords

The predefined search keywords can be used to create highly targeted search queries. To create a search query, see create a search query.

All Keywords

These keywords are generic to all ADC vendors and help searching objects irrespective of vendors.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

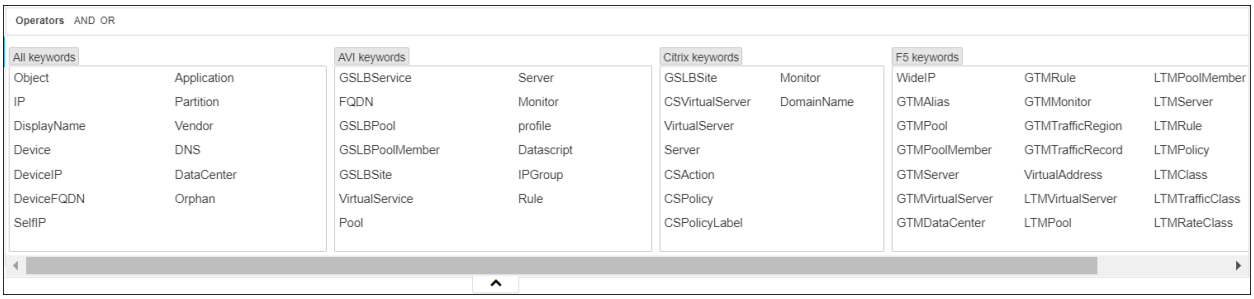
Sample Search Keywords

Common Keywords	Description
Object:	Search for objects using the name.
Device:	Search for objects from a particular device managed in the ADC inventory.
Displayname:	Search for objects using the Displayname. Displayname represents the naming format of your objects that needs to be followed throughout the application. Refer <displayname section>
IP:	Search for objects using the IP address. Eg: Objects like Virtual Servers, Pool members that belong to a particular IP can be searched.
Application:	Search for objects associated with an Application name. AppViewX tags all the hierarchical objects of an Application using the WideIP name as a tag during every config fetch.
Partition:	Search for objects from a device partition or tenant or context.
Vendor:	Search for objects from a particular vendor (F5, AVI, etc.,)
DNS:	Search for objects using their DNS name. AppViewX performs a scheduled (twice a day) reverse lookup for the objects that contain an IP address in it and persists the DNS name of the objects.
Datacenter:	Search for objects from a particular Datacenter. Datacenter name information should have been provided at the time of device addition.
Orphan:	Search for objects that do not have Parent association. Objects like GTM/LTM pools that are not associated with WideIP/Virtual Server will be considered as Orphans.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendor-Specific Keywords

ADC vendor-specific keywords are available to search for objects from a particular ADC vendor. The keywords are grouped as per the vendors managed in the inventory.



Sample Search Queries

Vendor-Specific Keywords	Description
vendor:Citrix AND CSVirtualServer:.*	The list of all the CS Virtual Server applications from Citrix devices managed in AppViewX.
vendor:Citrix AND CSVirtualServer:Test.*	The list of all the CS Virtual Server applications from Citrix devices containing Test in it.
vendor:AVI AND FQDN:.*	The list of all the applications with FQDN from all the AVI devices managed into AppViewX.
vendor:AVI AND FQDN:Test.*	The list of applications with FQDN containing “Test” in it, from all the AVI devices managed into AppViewX.

Logical Operator with Keywords

The common keywords and Vendor specific keywords can be combined with any Logical operator.

Sample Keywords

Keywords	Description
virtual server:* AND device: F5_v11	The list of virtual servers available in the F5_v11 device.
device: F5_v11 AND virtual server:* AND IP:192.168.96.101	The list of virtual servers in the F5_v11 device that uses the IP address 192.168.96.101 is identified.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Create a Search Query

To create a search query:

1. On the **Search** screen, click the  (**Expand**) tab at the bottom of the search field.
2. The field expands to display a list of search keys available for the ADC.
3. For each metadata type, you add to the search query, enter a value or partial value to search for. If you enter no text after the search key, the search engine automatically searches for "all values."
4. (Optional) Click the AND and OR operators at the top of the Search keys field to create Boolean searches.
5. When you are finished creating the search query, click inside the search field, then click Enter on your keyboard to run the search.

Search Using Regular Expression (Regex)

You can enter regex (such as * or .*) or the object/certificate name followed by the regex (.*) in the search field and click Enter on your keyboard. The search results will be displayed as follows:

- .* - All the device objects/certificates along with their hierarchy are displayed in their respective tabs.
- (searchstring).*- This RegEx results in the objects, whose name starts with the search string..
- The search results that are matched with the text entered will be highlighted.

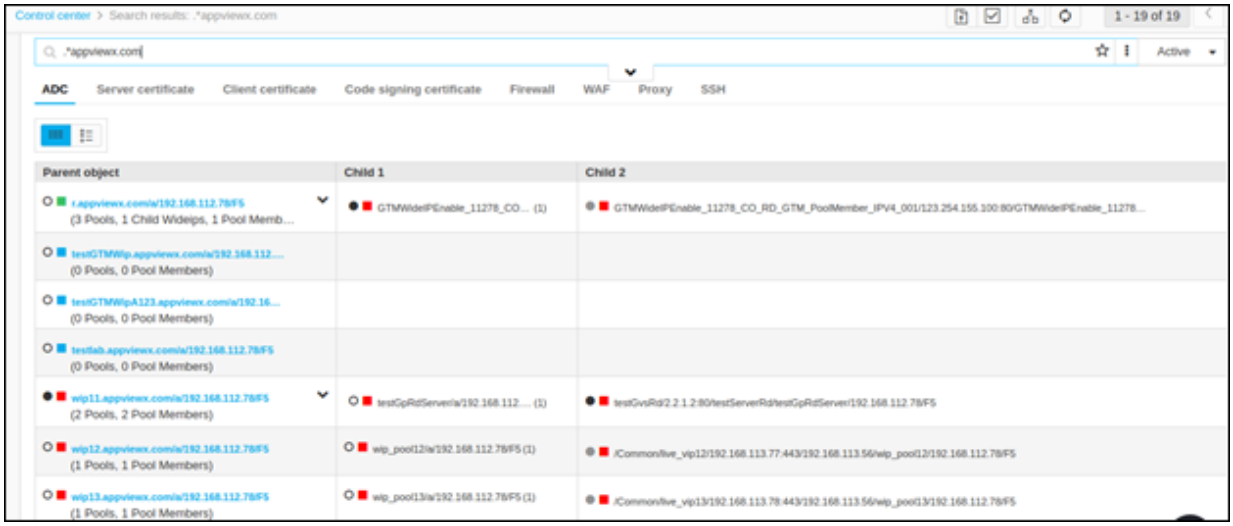
Examples

- For the RegEx **gs-f5-pe.***, the search results that are matched with the text entered will be highlighted:



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

• If all the subdomains of a DNS are needed to be searched, then .*DNSName shows all the subdomain:



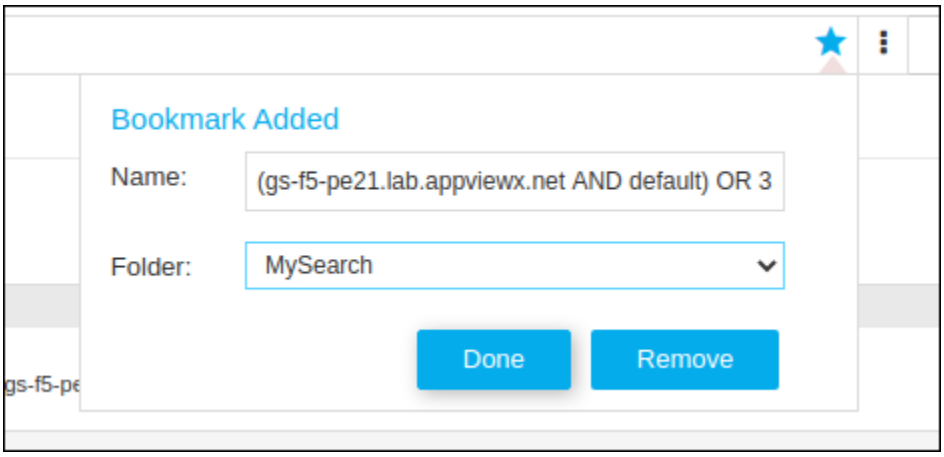
Create a Bookmark

If any search strings or search queries are repeatedly used, it can be bookmarked for future access.

To create a bookmark for the ADC frequent search items,

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**..
   The **Control Center** search screen appears.
2. Run a search.
3. On the search results screen, click the  button next to the search bar.
4. Click the  button to create a bookmark folder.
5. Click the  button on the search bar. A pop-up message will be displayed at the top of the screen, **Bookmark(s) created successfully**.
6. On the **Bookmark Added** pop-up screen, enter a name for the bookmark to help the users identify it.

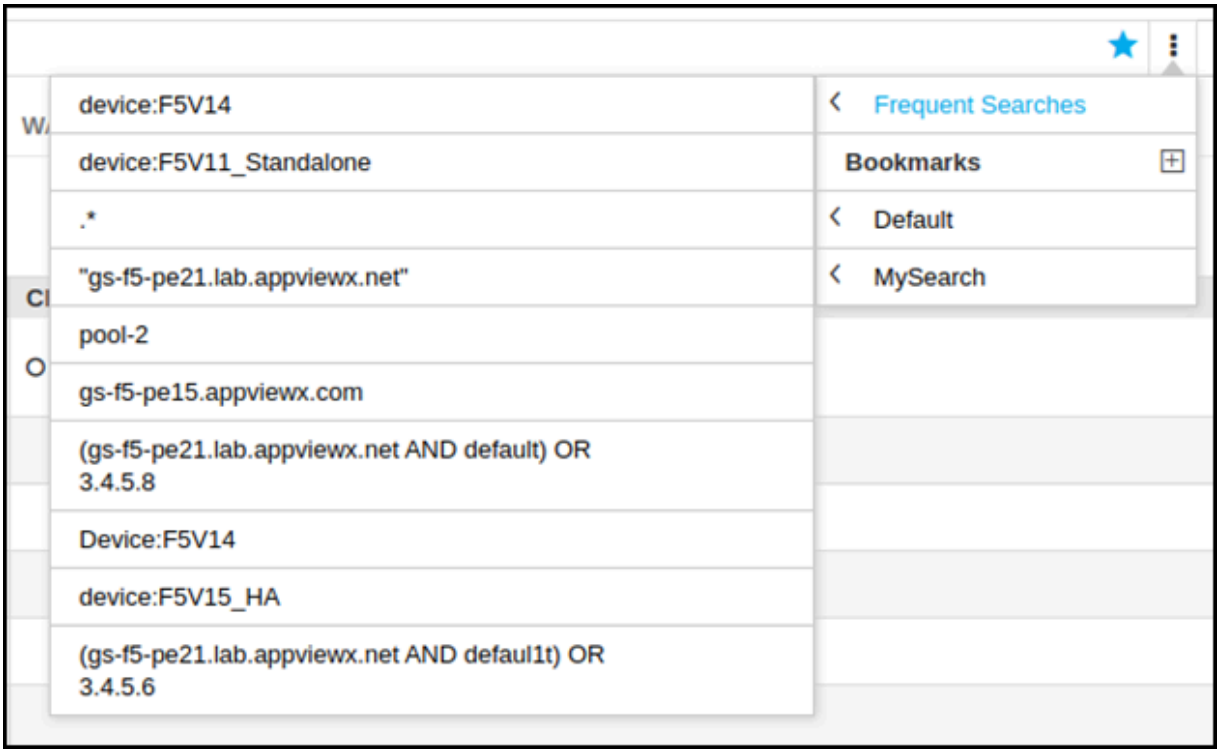
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. Select the folder to which you want to add this bookmark from the dropdown list and click **Done**.

> **Note:**
> If you want to delete the bookmark, click the Remove button. A pop-up message will be displayed at the top of the screen, Bookmark(s) deleted successfully.

Queries that are frequently used are added to the frequent searches by default and the top 10 queries can be accessed in a single click as below:



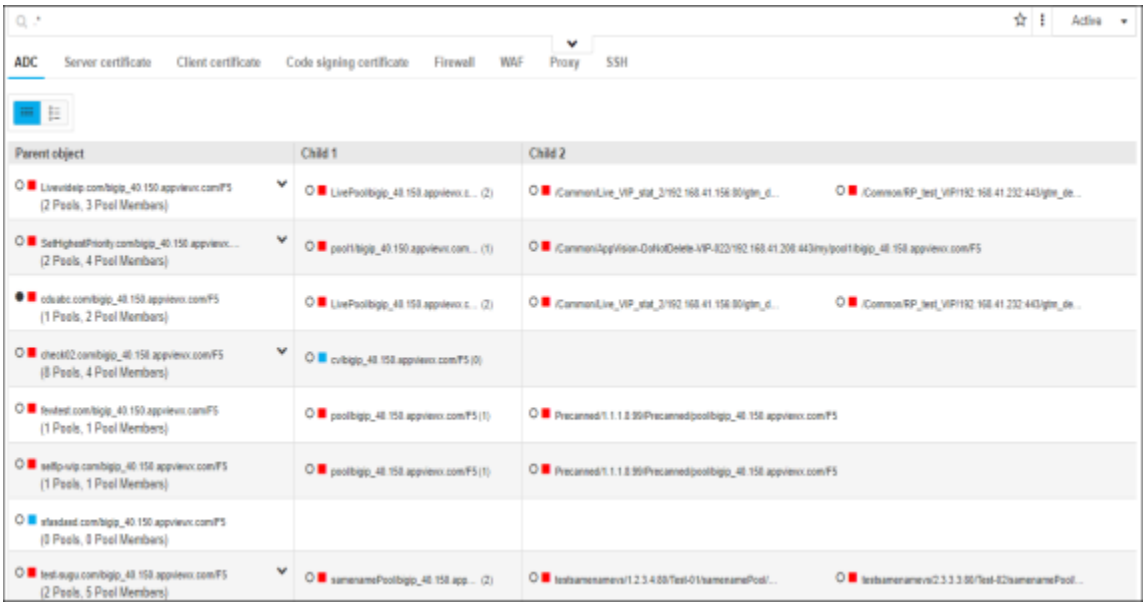
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

View Basic Details of ADC Search Results

In the App Search module, the Control Center is designed to display the ADC search results in both the Application and Infrastructure View. Although each view displays the search results in a different manner, the same search query is used. In order to view the search results, ensure that you have been assigned to a role that has access to Application View and/or Infrastructure View along with the set of actions you want to perform on the objects.

Application View

This is the default search result landing page. The parent objects are listed in the left-most column and related first-level children and second-level children appearing in subsequent columns.



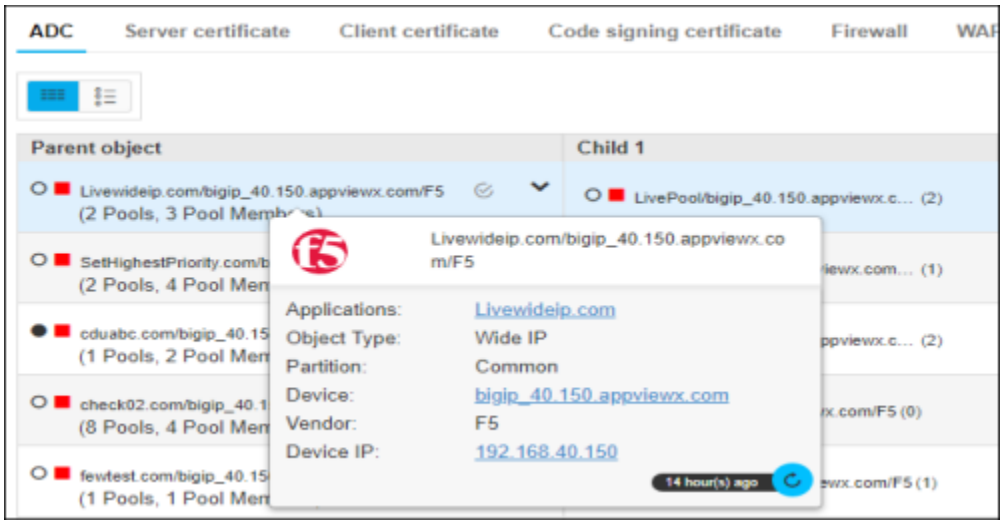
To view the basic details of search results within the Control Center,

1. Run a search.
   By default, search results are displayed in the Application view.
2. Each object has a series of symbols that appear before its name. You can hover over the icon to identify the State/Status of the objects.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Circles (State)**
  - ● A solid black circle indicates that the object is disabled.
  - ○ A hollow circle indicates that it is enabled.
  - ● A solid grey circle indicates that the object is parent disabled or none when the object state is unknown.
- **Squares (Status)**
  - ■ A green square indicates that the object is available and enabled or healthy (Cloudflare only).
  - ■ A blue square indicates that the object is unknown and disabled/draining (Nginx only).
  - ■ A red square indicates that the object is offline and enabled/down (Nginx only) or Critical (Cloudflare only).
  - ■ An orange square indicates that the object is offline/mixed (Nginx) but might become available again and enabled or Degraded (Cloudflare only).
  - ■ A solid grey square indicates that status of the object is unknown or not monitored / health check is in progress (Cloudflare only).
- Get additional information about the object on hovering. An actionable tooltip will be available that represents the object's details, last refresh time, etc.
- In the search results field, hover your cursor over the result whose basic details you want to view.
- A popup box appears, listing the details for the object.

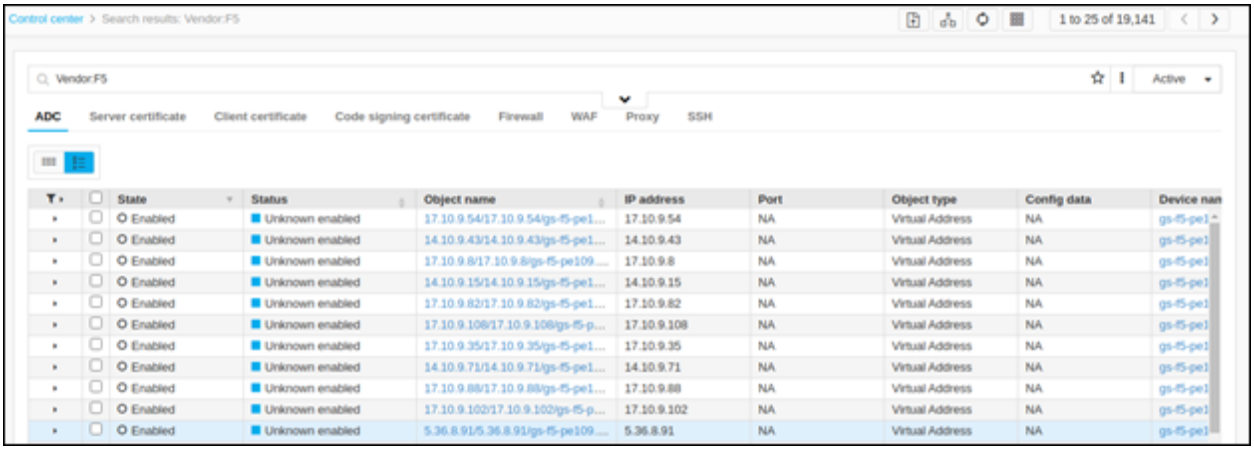
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- The DNS name of an object (applicable only for F5 device objects) will be displayed in the pop-up box only if DNS is configured during AppViewX installation.

Infrastructure View

Only the ADC objects corresponding to your search criteria are displayed and not its hierarchy (such as a parent, child 1, and child 2). By clicking the  (**Infrastructure view**) button to switch from the Application view.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

View the Object Details

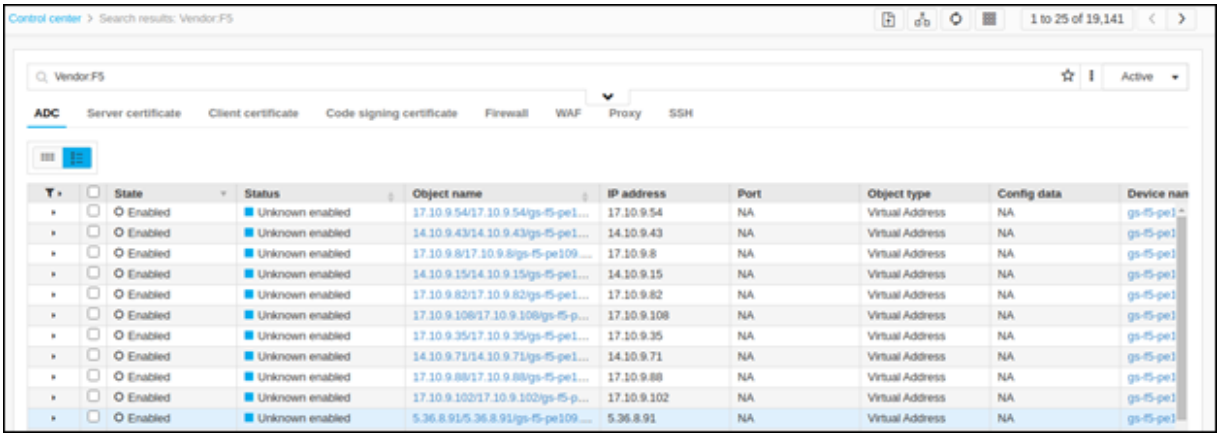
1. The object details are represented in the form of a tabular with additional information as follows:

Object Details - Column and Description Table

Column name	Description	Default?
State	State of the objects.	Yes
Status	Status of the objects.	Yes
Object name	Based on the display name configured.	Yes (mandatory)
IP address	The IP address of the objects.	Yes
Port	Port of the IP address.	Yes
Object type	Type of object.	Yes
Config data	Configuration of the objects is displayed in a Pop upon clicking the link.	Yes
Device name	The device name of the object. On clicking, the user is redirected to the device addition page.	Yes
Vendor	Vendor of the object.	Yes
Device IP/FQDN	IP or FQDN of the device. The user is redirected to the device login on clicking.	No
Connection count	Display the live connection count of the objects if applicable.	No
Partition	Partition/tenant/context of the objects if applicable	No
DNS name	DNS resolution of the objects if applicable.	No

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

2. In the search results field, hover your cursor over the result whose basic details you want to view.



Filter the Objects

To filter the objects in the Infrastructure view:

1. Run a search.
2. Click the filter icon located at the first cell of the table.
3. The filter options are enabled in the header of the table.
4. Select and/or enter the criteria to filter the objects.
   The objects that match the filter criteria are displayed.

Note:

- To remove the filter option, click again the filter icon.
- To reset the filter criteria that have been applied, select and/or remove the text that has been applied for filtering the objects.

View Additional Details of Search Results

There are two ways to view additional details about search results and the method varies depending on the type of object you are viewing:

- **Topology View**: Clicking the object takes you to topological screens that provide much more information about the corresponding object than is displayed on the search results screen.
- **ADC Topology Actions**: ADC primary and secondary object search results can be right-clicked to view the list of actions available for them.

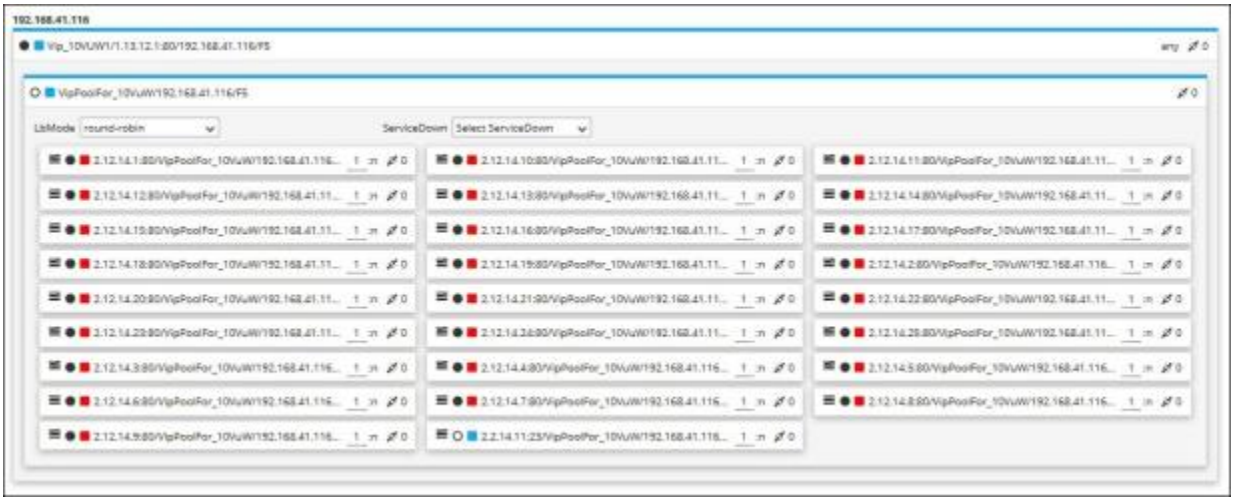
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Topology View

When you click a search result for a primary ADC object either from the **Application** or **Infrastructure** view, a topology view opens, providing a detailed, hierarchical map of the structure of the ADC. The following are the sample ADC topologies within the Control Center.

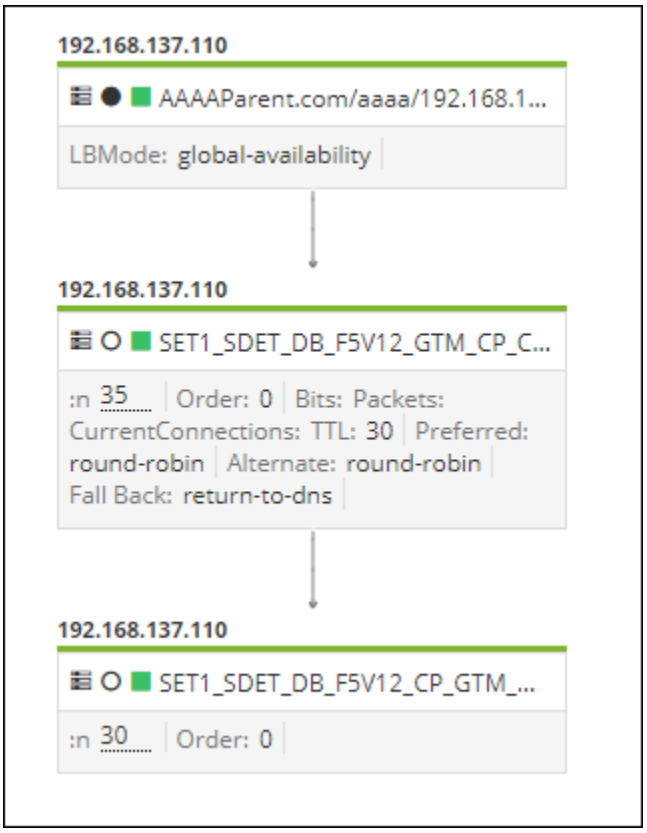
For example, read the topology view: GTM > LTM > Pool > Pool mem...

Virtual Server (VIP)/SLB Topology

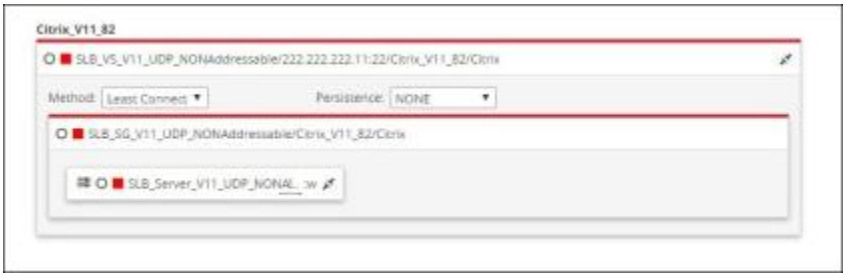


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Wide IP/GSLB Topology

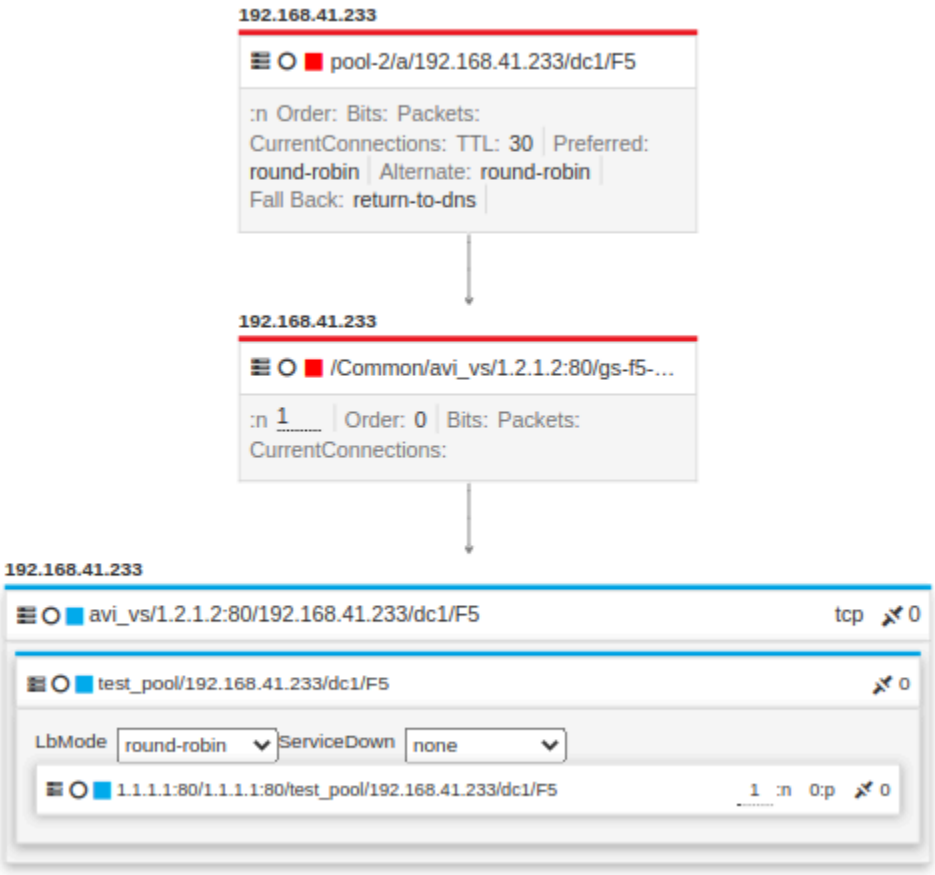


Server Load Balancing (SLB) Virtual Server Topology



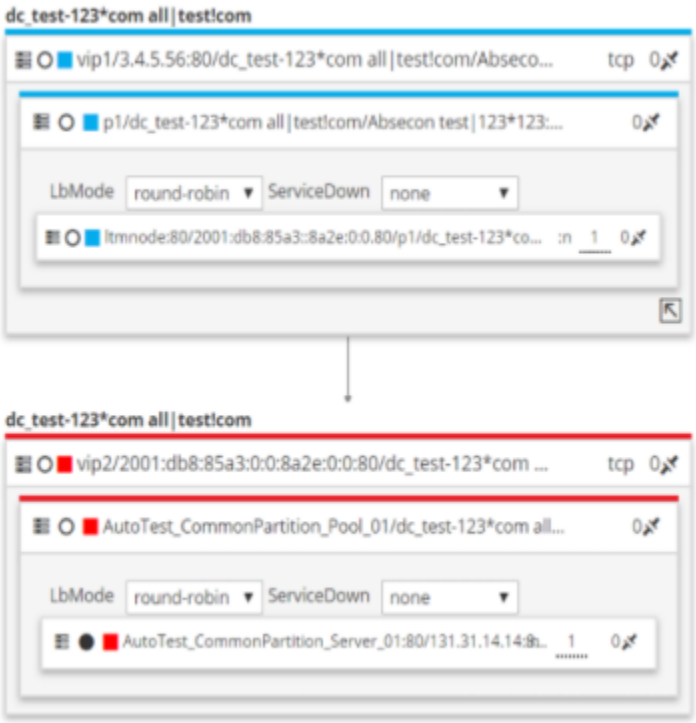
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Virtual Server (VIP) under Wide IP Topology



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Virtual Server IP (VIP) under VIP Topology



Filter ADC Search Results

To filter ADC search results within the Control Center:

1. Run a search.
2. On the search results screen, click the **Active** drop-down option.



3. Select one of the three filter options in the dropdown list that appears:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **All** - View all search results, regardless of status.
- **Active** - View search results for objects with a status of Active.
- **Stand by** - View search results for objects with a status of Standby.

4. The screen then refreshes and filters the results on both the Application view and Infrastructure view based on the filter you selected.

Export Search Results

When you search for the objects in the Control Center, the objects that match the search keyword are displayed in the Application view. The resulting objects can be exported. You can export the selected objects or all the resulting objects.

To export objects search results within the Control Center:

1. Run a search.

   By default, the search results are displayed in the  (**Application view**) page.
2. Click the checkboxes beside the object name to select the object details that you want to export from the grid or select all of the ADC objects by clicking the  (**Select all**) button in the Command bar.
3. All of the search results on the screen display with a blue background to indicate that they have been selected.
4. Click the  (**Export**) button in the Command bar.

   The `.csv` file is downloaded to your computer.
5. If you want to export the search results displayed on the infrastructure view,  (**Infrastructure view**) button to switch from the Application view search result page and do the following:
   - Select the checkboxes beside the first column of the grid to select the object details that you want to export from the grid. You can select all of the objects from the search results screen by selecting the first checkbox in the grid.
   - Click the  (**Export**) button in the Command bar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- On the **Export** screen that pops up, select either the **All columns** or the **Displayed columns** radio button based on what you want to export from the grid.

  The `.csv` file is downloaded to your computer.

Access the Actions Menu for Objects on the ADC Search Results and Topology Screens

To access the actions menus for objects that appear on the ADC search results and Topology screens:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
   The Control Center search screen appears.
2. Search for the ADC object whose actions menu you want to access.
3. In the search results field, you can do the following:

   - **Application View** - Right-click the object or one of its first or second-level children to view the list of actions you can perform.

   - You can multi-select or select all of the objects from the grid either by clicking the  (**Select**) button beside the object or the  (**Select all**) button in the Command bar.

   - **Infrastructure View** - Select the checkboxes beside the first column of the grid and then, click Actions to view the list of actions you can perform.

   - You can select all of the objects from the inventory by selecting the first checkbox in the grid.

   - Click the object's name in the search results field either from the Application view or Infrastructure view. The topology screen corresponding to the object you selected appears, right-click any of the components in the topology to view the list of actions you can perform.

     - The actions that appear in the list vary depending on the type of object you right-click as well as whether it is a parent or child, but the most common actions are listed below. To initiate any of the actions, scroll down in the list and click the corresponding action button.

Actions Menu for Objects

Actions	Description
Enable	Enable an object.
Disable	Disable an object and terminate all active connections.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Actions	Description
Graceful disable (AVI devices only)	Disable an object only when all the currently active client connections are closed by either the server or the client.
Backup & Restore	• **Backup device** - Create a backup of the device associated with the object. For more details refer to the Create a Device Backup Group section of this guide. • **Restore object** - Restore object configuration to a previous state. For more details refer to the Restore an Object section of this guide. • **Compare** - To compare the objects with their associated devices. For more details, refer to the Compare ADC Objects section of this guide.
View	• **View graph** - View the timeline statistics of an object. For more details, refer to the View Timeline Statistics for an Object section of this guide. • **View config** - View the current configuration of all levels of the device object. For more details, refer to the View Configuration details section of this guide. • **View log/history** - View the log history of the object. • **View alerts** - View any alerts related to the object.
Compare config	Compare current and/or archived configurations of similar objects or the same object over time.
Clear persistence records (F5 devices only)	Clear the persistence records for VIP and pools.
View persistence records (F5 devices only)	View the persistence records for VIP and pools.
Advanced	• **Enable/Disable persistence** (F5 devices only) - Turn on or off the tracking and storing of session data, which is used to ensure that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Actions	Description
	• **Enable/Disable all** (F5 devices only) - Turn on or off the object available across all the pools. • **Forcedown all** (F5 devices only) - Force shutdown of the object available across all the pools.
View source connections (F5 devices only)	View the source connection IPs for VIP and pools.
Forcedown (F5 devices only)	Force shutdown of the object.
FD clear active connections (F5 devices only)	Clear active connections to the object.
Open in a new window	To display the topology view of the selected object in a new window.
Additional Actions on the Topology View	• **Ratio** - To modify the ratio of the pool member. • **Address Resolution Protocol (ARP)** - To enable/disable the ARP for the virtual address. • **Load Balancing (LB) mode** - To modify the load balancing method for the pools and service groups. • **Service down** - To control the connection management behavior of the pool. The following are the service-down settings that you can perform on the pool: Reselect, Reject, Drop, and None. • **Weight** - To modify the real server weight.
Drain	Put the server into the draining mode.

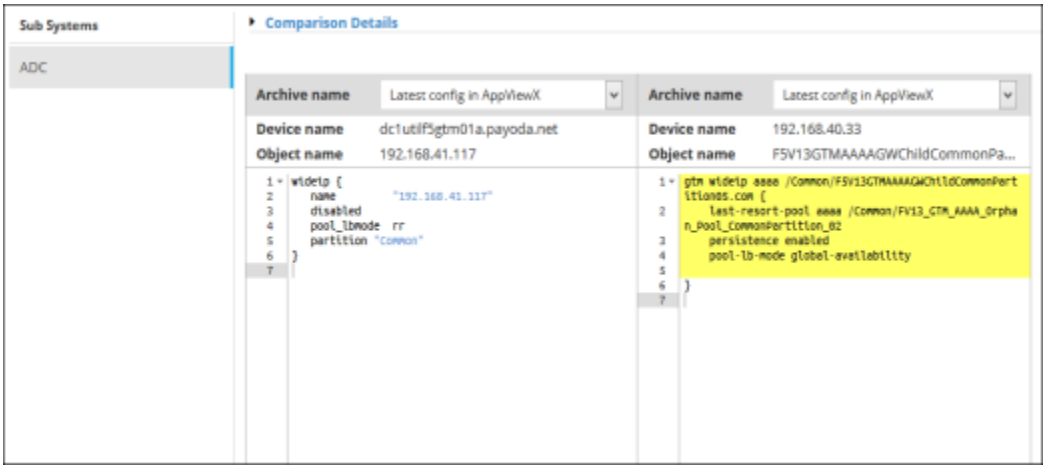
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Compare ADC Objects

Comparisons are only possible for F5 ADC objects. Comparisons are also possible for multiple objects. You can select up to 5 devices to compare with the original device you selected.

To compare ADC devices from the Control Center results within the App Search:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
   The Control Center search screen appears.
2. Run a search.
   By default, the search results are displayed in the (**Application view**) page.
3. Click the (**Select**) button beside each object in the list that you want to use in the comparison.
4. Right-click the selected objects and click **Compare config** from the dropdown list that appears.
5. If you want to compare configurations for the search results displayed on the infrastructure view, click the (**Infrastructure view**) button to switch from the Application view search result page.
6. Select the checkboxes beside the first column of each object you want to use in the comparison and then, navigate to **Actions** > **Compare Config**.
7. The screen refreshes and displays the archived configurations side-by-side:



Filter the Information Displayed in an ADC Topology

To filter the information displayed in an ADC topology:

1. Open the ADC topology that you want to filter.
2. In the Command bar, click the (**Filter**) button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. In the dropdown list that appears, deselect the type of information you want to filter out of the topology. Note that the abbreviation LB in the "LB configuration" filter refers to load balancing.



4. The topology updates and no longer displays the information you filtered out.

> **Note:**
> To turn off filters, repeat the steps above, but select, rather than deselect, the type of information you want to view.

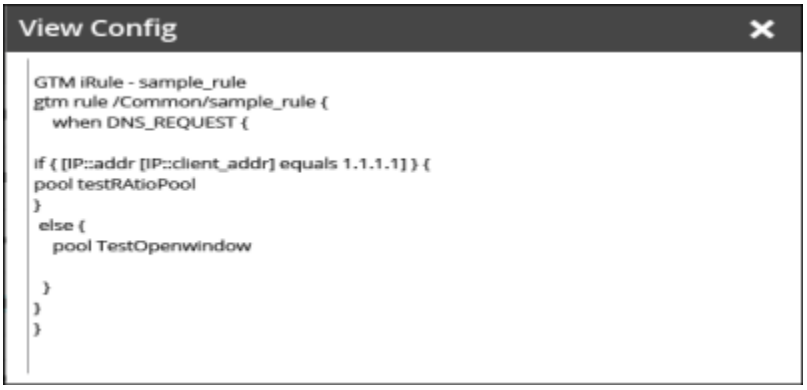
View Configuration Details

To view the configuration details of the ADC device objects:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**..
   The Control Center search screen appears.
2. Run a search.
   By default, the search results are displayed in the  (**Application view**) page.
3. Right-click the device name and select **View** > **View config** from the dropdown list that appears.
4. If you want to view the statistics for the search results displayed on the infrastructure view, click the  (**Infrastructure view**) button to switch from the Application view search result page.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. Select the checkboxes beside the first column of each object details and then, navigate to **Actions** > **View** > **View config**. A popup screen appears, displaying the object's configuration as shown in the image below:



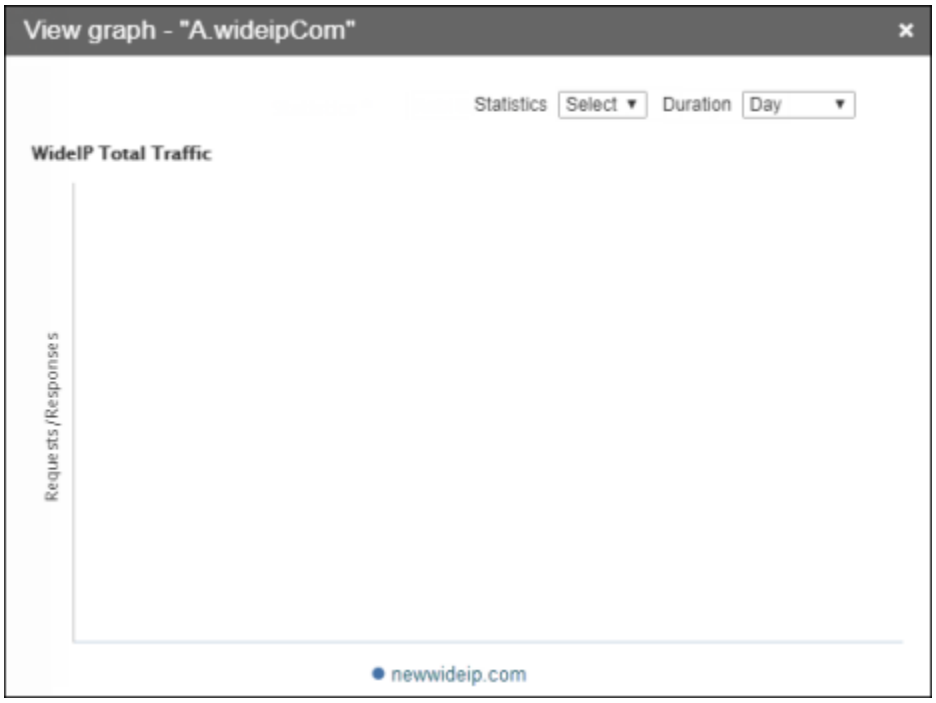
View Timeline Statistics for an Object

To view a customizable graph that displays timeline statistics for an F5, Citrix or A10 object visible through the App Search module:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
   The Control Center search screen appears.
2. Run a search.
3. By default, the search results are displayed in the  (**Application view**) page. Right-click the device name and select **View** > **View graph** from the dropdown list that appears.
4. If you want to view the statistics for the search results displayed on the infrastructure view, click the  (**Infrastructure view**) button to switch from the Application view search result page.
5. Select the checkboxes beside the first column of each object details and then, navigate to **Actions** > **View** > **View graph**. A statistics chart appears, displaying the following two fields at the top:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Statistics** - The entries in this list vary depending on the device you selected. As you select different entries, the graph below updates to display the statistics related to the item you chose.
- **Interval** - The following time intervals can be selected: **Day**, **Week**, **Month**, or **3 Months**. As you select different intervals, the graph below updates to display the statistics for the corresponding time frame.



Asset Management

Asset Management Overview

AppViewX's ADC solution provides a single-pane-of-glass view of complex ADC infrastructure, supporting physical and virtual devices from the industry's leading ADC providers. It provides application-centric, customizable reports to help monitor application utilization and enable efficient capacity planning on the ADCs. It gives application and operation teams a topology view of the application delivery infrastructure for better visibility of their application. AppViewX can also project better insights into application health, state, status, utilization, and performance of applications from the managed network assets.

The ADC devices are managed/monitored under the following tabs within the Inventory:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Device
- Groups
- Back-up & Restore

Supported ADC Vendors and Prerequisites

To manage the Application Delivery Controller (ADC) devices on AppViewX, there are certain prerequisites. In the following tables, you can find the prerequisites for the supported vendors.

Prerequisites for Managing A10, AVI, Akamai, and Array

Vendors Supported	A10*	AVI***	Akamai**	Array
Version Supported	v2.7.x , v4.0.1, v4.1.x, v5.1.x, and v6.x	16.2.x, 16.3.x, 17.1.x,17.2.x,18.2.x, 20.1.x, 22, 30, 31.x		Rel.APV.10.4.2.93
IP Address/FQDN	IP address and FQDN	IP address and FQDN **Note:** Device should not be added with Cluster IP		IP address and FQDN
User Privilege	CLI/WEB/xapi access and Read/ Write access is required	System Admin Role for all tenants. **Note:** SuperUser Access is required only to verify	Client token, Client Secret, Access Token, Host URL	api access

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendors Supported	A10*	AVI***	Akamai**	Array
		terminal check, not mandatory.		
Enable Password	Required	Not applicable	Not applicable	NA
License Check	No	No	Not applicable	No
Services and Ports for AppViewX Communication	22, 80, and 443	22, 443, 5514 (To receive syslog on AppViewX from AVI)	443 Outbound access from AppViewX is needed to manage the device and Perform Akamai Actions	A port needs to be specified for API access. **Note:** Any port no. can be used.
Internet Access/ Proxy if Required	Not required	Not required	Not required	NA
Location from which Certificates are discovered if managed	No specific location	No specific location (we use API's to get certificate details)	Not applicable	NA

Prerequisites for Managing Amazon ELB, BigIQ, Cisco, and Citrix

Vendors Supported	Amazon ELB**	BigIQ**	Cisco ACE**	Citrix*
Version Supported	Only Classic and Application ELB is supported	5.x (5.2/5.3), 7, 8	5	v9.x, v10.x , V11.x, v12.x, v13.x
IP Address/FQDN	Not applicable	IP address	IP address	IP address and FQDN
User Privilege	Access Key, Secret Key, and	{ Role : Admin , tmsh access	Username / Password,	Username / Password,

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendors Supported	Amazon ELB**	BigIQ**	Cisco ACE**	Citrix*
	Region information is required.	is required } Username / Password, Credential List AppViewx/ CyberArk/Thycotic	Credential List AppViewx/ CyberArc	Credential List AppViewx/CyberArk Role:(Superuser access)/ BeyondTrust
Enable Password	Not applicable	Not applicable	Not applicable	Not applicable
License Check	Not applicable	Yes	No	No
Services and Ports for AppViewX Communication	Port 443, Outbound Internet access is required to manage the ELB in AWS	Port 22 SSH	Outbound Access from AppViewX is needed for SSH service and HTTPS service	Outbound Access from AppViewX for SSH and HTTPS service (Default/ Custom Port Information) is required Outbound access to port 5514 is required for Citrix Device to forward the syslog to AppViewX
Internet Access/ Proxy if Required	Required	Not applicable	Yes, if the needs be accessed with Public IP address.	Yes, if the needs be accessed with Public IP address.
Location from which Certificates are discovered if managed	Not applicable (Certificates that are assigned to the listeners will be discovered by AppViewX)	Certificate Discovery is not supported	Certificate Discovery is not supported	Certificates are parsed only from the /flash/nsconfig/ ssl/ directory

Prerequisites for Managing F5, HAproxy, NginxPlux, and Infoblox LBDN

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendors Supported	F5***	F5XC***	HAProxy*	NginxPlux***	Infoblox LBDN**
Version Supported	v10.x , v11.x , v12.x , v13.x, v14.x, v15.x, v16.x, v17	NA	Open source version: 1.4.x, 1.5.x, 1.6.x, 2.x (Community), 2.x (Enterprise)	R11, 18, 19, 20, 21, 22, 23, 24, 25,26, 27, 28, 29, 30, 31	v8.2 and wapi v2.7.1
IP Address/ FQDN	IP address or FQDN	FQDN	IP address	IP address	IP address or FQDN
User Privilege	{ Role : Admin , bash access is required } Username / Password, Credential List AppViewx/ CyberArk/ BeyondTrust	admin	Username / Password, Credential List AppViewx/ CyberArc/ Thycotic	Access to Nginx installed configuration location and access to check syntax of niginx.conf and restart nginx service to apply the changes	Superuser or access to LBDN objects that needs to be managed from AppViewX
Enable Password	Not applicable	NA	Not applicable	Not applicable	Not applicable
License Check	Yes	No	Yes	Yes	Not applicable
Services and Ports for AppViewX Communication	22 and 443	NA	Port 22 SSH	22 and any custom port for stats	443 or custom port to access API calls
Internet Access/Proxy if Required	Not applicable	Internet access required	Not applicable	Not applicable	Not applicable
Location from which Certificates are discovered if managed	/config/ filestore/files_d/ <partition>_d/ certificate_d/	NA	/config/filestore/ fil es_d/_d / certificate_d/ /etc/ httpd/conf/ssl.	/etc/nginx/ conf.d/ or any custom location	Not applicable

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendors Supported	F5***	F5XC***	HAProxy*	NginxPlux***	Infoblox LBDN**
	/etc/httpd/conf/ssl.crt/ /etc/pki/tls/certs/ V10 /config/ssl/ssl.crt/ /etc/httpd/conf/ssl.crt/ /etc/pki/tls/certs		crt/ /etc/pki/tls/certs/ V10 /config/ssl/ssl.crt/ /etc/httpd/conf/ssl.crt/ /etc/pki/tls/certs/		

Prerequisites for Managing Cloudflare

Vendors Supported	Cloudflare**
Version Supported	v4
IP Address/FQDN	NA
User Privilege	Api Token **Note:** When creating a Cloudflare API token for managing load balancing, the following minimum permissions should be granted:Permissions: 1. Zone-level Permissions: Load Balancing: Read/Edit This permission allows the token to manage and modify load balancer configurations within specified zones.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Prerequisites for Managing Cloudflare (continued)

Vendors Supported	Cloudflare**
	2. Account-level Permissions: Load Balancing Pools and Monitors: Read/ Edit This permission enables the token to manage and modify load balancing pools and health monitors within specified accounts. Resources: **Account Resource**: Include the specific account(s) that should be accessible using this token. **Zone Resource**: Include all zones where load balancing management is required.
Enable Password	NA
License Check	NA
Services and Ports for AppViewX Communication	Port 443, Outbound Internet access is required to manage the cloudflare
Internet Access/Proxy if Required	Required
Location from which Certificates are discovered if managed	NA

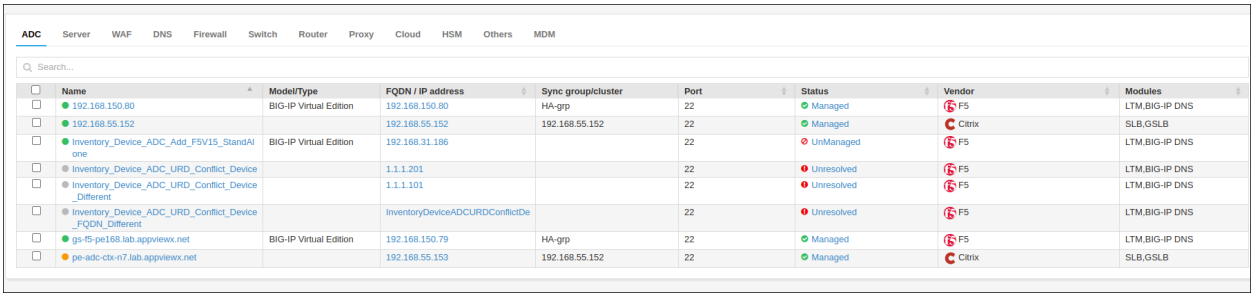
Note:

- * - The vendors are supported for certificate management and automation only.
- ** - The vendors are supported only for automation.
- *** - The vendors are supported for device/application end-to-end management, certification management, and automation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Inventory

Device inventory provides a birds-eye view of all the ADC devices onboarded into AppViewX. It is the single console where all the information about ADC devices such as number of objects, current status, Group Name, and version number will be displayed.



The ADC devices can be onboarded and managed from the Device Inventory. The complete details of the devices are available in the device Inventory for easy access and monitoring. Device Inventory gives you detailed information as described in the following table:

Device Inventory - Column and Description Table

Column Name	Description
Checkbox	Allows selection of individual devices for actions.
Name	The unique name assigned to each device.
Model/Type	The specific model or type designation of the device. Supported only for F5 vendor.
FQDN/IP Address	The FQDN or IP address of the device.
Sync groupcluster	Indicates the synchronization group or cluster to which the device belongs.
Port	The port number used for communication with the device.
Vendor	The vendor of the device.
Modules	Lists any modules installed on the device.
Data center	Specifies the data center location where the device is hosted.
Object count	Total number of objects managed or monitored by the device. For F5 vendor support (LTM Module):

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Inventory - Column and Description Table (continued)

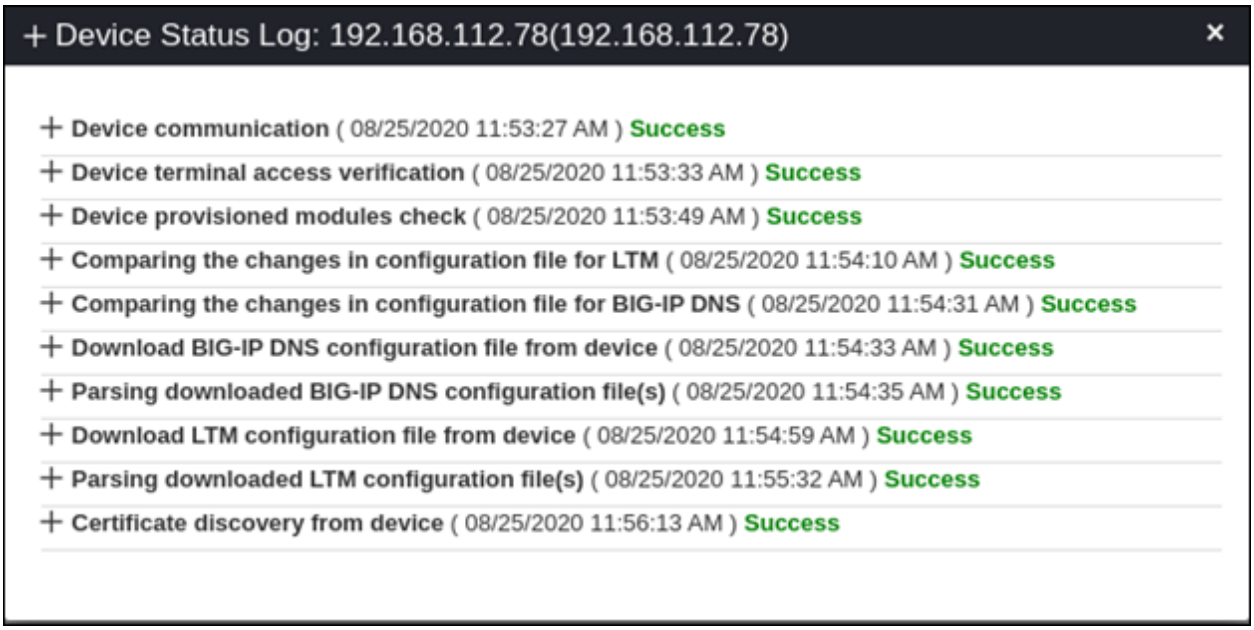
Column Name	Description
	For non-ADC customers, when parsing is skipped based on a database flag, the object counts will display a managed status. Even if there are issues with managing the device, its status will still appear as Managed. You can verify the details of any failures in the audit logs.
Version	Current software version running on the device.
Status	Current operational status of the device (e.g., Managed, Unmanaged).
Report	Access to reports related to device performance or issues.
Services	Lists the services provided by the device.

Once the ADC device details are onboarded, devices must successfully go through the following setup process to be Managed into AppViewX,

- AppViewX tries to establish a connection with the device with the provided username and password. The communication protocol varies based on the ADC device providers.
- If the connection gets established, the required access is validated. For example, If any F5 ADC device gets added, Terminal access verification happens from AppViewX.
- The requested module is validated in the ADC Device. For example, if an ADC device is added as GSLB, then the appropriate provision is validated.
- The version of ADC device is fetched and validated for the supported version.
- The configuration is fetched from the device through REST or any other supported protocol by the ADC device provider. The configurations are made available in AppViewX.
- If the certificate discovery is enabled, then appropriate certificates from the ADC device are discovered.
- If secondary device detection is enabled, then failover/standby device information is fetched.

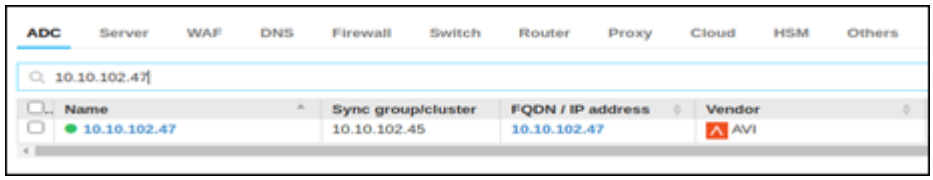
The progress of these steps can be viewed in the inventory by clicking the status column of the device. By clicking the (+) button, detailed information will be available for each step.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

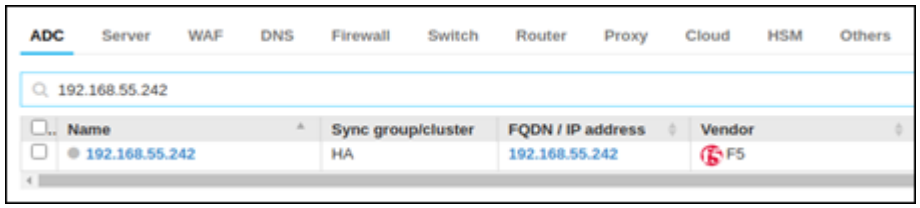


Device State

High Availability of the applications can be ensured by configuring ADC devices in Active-Standby mode. This type of configuration will be planned for disaster recovery. In some cases, the devices in the group will be in active-active mode for handling the traffic efficiently. In AppViewX, the current mode of the device is called state and it will be displayed along with the device name in the inventory page. The device will be in any one of the following states,

State	Color Code	Description
Active	Green Circle	The device is currently processing the traffic of the application in the Active-Standby setup. 
StandBy	Orange Circle	The device will be ideal and ready to take over whenever the failover occurs in the Setup. 

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

State	Color Code	Description
Offline	Grey Circle	The device is taken down for maintenance and in this state, the traffic will not be passed through the device. 

To know the current state, mouse over the color icon available before the device name.

This state will get updated in a particular time interval during configuration fetch from the device.

For F5 and Citrix, if Syslog is enabled, it will get a near real-time update whenever there is a flip in the device.

Device Status

On adding the ADC device into AppViewX, the status defines the current stage of device configuration parsing in AppViewX. This data will be available as a separate column in the device inventory page. The device will be in any of the following statuses in the inventory and the detailed progress can be viewed by clicking the status image,

Status	Symbol	Description
Queued	Queued	The device details are added in AppViewX and waiting for configuration fetch to start from the ADC device.
Inprogress	Inprogress	It states that the communication to the device is successful and configuration fetch from the device is yet to be completed.
Managed	Managed	All the configurations from the added device are pulled, parsed, and available in AppViewX. This stage states that the device is successfully onboarded into AppViewX.
Unresolved	Unresolved	AppViewX could not connect to the device due to communication or credentials issues.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status	Symbol	Description
Failed	Failed	AppViewX established a connection with the device but the device configuration parsing was failed due to some exceptions. The detailed reason for failure can be known by clicking the Failed icon itself.
Unmanaged	UnManaged	If any maintenance activity needs to perform on a device, then the device can be moved to this stage in AppViewX by clicking the () button on the top right corner of the inventory page. In this stage, AppViewX will not trigger any communication to the device. Whenever required, the device can be managed again by clicking ().

Import Devices

Device import provides a hassle-free experience in onboarding multiple ADC devices into AppViewX in one single step. For onboarding multiple devices, the details should be filled in the excel sheet in the predefined format and can be uploaded to AppViewX and from there AppViewX will dynamically onboard all the devices available in the sheet.

To import devices using a `.csv` file:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**..
   By default, the **ADC** tab opens.
2. Click the  **Import** button in the Command bar.
3. On the **Import** screen that appears, navigate to the location of the import file, then select it.
4. (Optional) Enable the **Host** toggle button to import the host devices.
5. Click **Import** to add the devices and their details to the Inventory.

> **Note:**
> When the file is uploaded with improper structure or incorrect data, the import process will terminate with the errors highlighted.

Search for ADC Devices

Devices that are added to AppViewX can be found in the inventory by searching it by name in the search bar.

To search for objects:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Enter the search keyword in the **Search** field.
   The devices that match the search keyword are displayed.
3. The search term can be any of the device details like name, IP, provider, version, status of the device, sync group name, etc.
4. You can also search based on sync group or custom group name to see the devices available in a particular group.

Deleting ADC Device(s)

To delete ADC device(s):

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Select the desired ADC device(s) to delete.
3. Click the  (**Delete**) button in the Command bar.
   The **Delete** confirmation modal appears.
4. Click **Yes** to delete the selected ADC device(s).

> **Note:**
> To discard the deletion, click **No**. Note: The device details and configurations will be permanently deleted from AppViewX. Note: If the deleted device is onboarded again, it will be considered as onboarding a new device.

Manage and Unmanage Devices

To manage or unmanage devices:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. If the device you want to manage or unmanage is not listed on the screen, run a search to locate it.

> **Note:**
> If you try to manage a device that is already in managed state or unmanage a device that is already in unmanaged state, an error message appears at the top of the screen.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

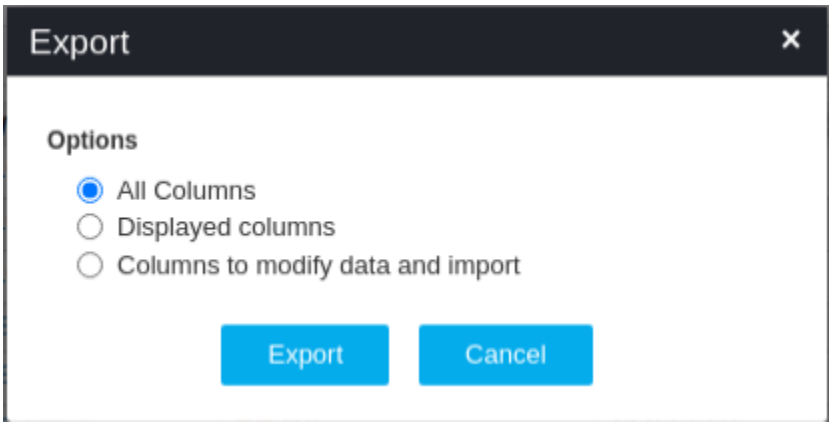
3. Click the checkbox beside the device name.
4. To start managing the device, click the (**Manage**) button in the Command bar at the top of the screen. To stop managing a device, click the **Unmanage** button.

Export Device Details

The device details, which are available in the Device Inventory page can be exported into an Excel file.

To export the details of one or more devices:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. If the device you want to export is not listed on the screen, run a search to locate it.
3. Click the checkbox beside the device name. If you are exporting details of multiple devices of the same kind, select the checkboxes for each one.
4. Click the (**Export**) button in the Command bar at the upper right of the screen.
5. On the **Export** pop-up screen that appears, select the type of information you want to export:



- **All Columns** - Select this option if you want to export all information about the device.
- **Displayed columns** - Select this option if you want to export only the information that is visible on the Device screen. This is useful if you need to compare values or settings for different devices and do not have any need to see the less important data.
- **Columns to modify data and import** - Select this option if you are exporting device details to make modifications and then re-import the data into the Device Inventory.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

6. On the screen that opens, select the location where you want the device details file to go, then click **Save**.
7. The details are then downloaded as an Excel (.xls) file.

Enable or Disable LB Sync

To enable or disable LB sync:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. If the device you want to enable or disable LB sync is not listed on the screen, run a search to locate it.

> **Note:**
> If you try to enable LB sync on a device that is already in LB sync enabled or disable LB sync on a device that is already in LB sync disabled, an error message appears at the top of the screen.

3. Click the checkbox beside the device name.
4. To enable LB sync for the device, click the **Actions** in the Command bar at the top of the screen, and then select **Enable LB sync**. To disable LB sync for the device, click the **Actions** in the Command bar at the top of the screen, and then select **Disable LB sync**.

Manually Fetch the Configuration for a Device

If the latest configuration in the device needs to be pulled into AppViewX, those devices can be selected and fetch config can be triggered. AppViewX will communicate with the device and pull the latest configuration available in the device and persist in AppViewX.

To manually get the configuration for a device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. If the device is not listed on the screen, run a search to locate it.
3. Click the checkbox beside the device name. If you want to fetch configurations for multiple devices of the same type, select their checkboxes, too.
4. Click the  (**Fetch Config**) button in the Command bar.
5. A notification appears at the top of the screen stating, "**Fetch config has been triggered for the device(s).**".

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Subscribe and Unsubscribe to Syslog

Subscribing to Syslog for a device enables receiving near real-time configuration updates. The Subscribing Syslog option is supported only for devices that are in the Managed and Failed status.

Supported Vendor: F5

Supports near real time updates for state and status of the objects.

Prerequisites:

- During installation, enable the flag for Syslog receiver. For setting the flag, see AppViewX Cloud Connector guide.
- Ensure Syslog reception is enabled in Platform. See Platform User Guide for enabling the Syslog reception.
- Ensure the device is managed.

To Subscribe/unsubscribe Syslog for a device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Select the checkbox of a preferred device.
3. Select the **Actions** dropdown menu from the menu bar.
4. Select **Subscribe Syslog** from the dropdown list.
   The process of subscribing to Syslog may take some time, and once completed, the status in the Syslog column is updated to **Subscribed** along with a timestamp.

> **Note:**
> To cancel the subscription to Syslog, choose the **Unsubscribe Syslog** option from **Actions** dropdown menu. The unubscription option is supported for the devices that are in the Managed, Failed, or Unresolved status.

Generate and Download an iHealth Report

BIG -IP iHealth is a diagnostic tool developed by F5 to manage local traffic manager (LTM) and global traffic manager (GTM) devices. The iHealth report provides tailored diagnostic information that gives

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

you valuable, actionable insight into the efficiency of the hardware and software running in your BIG-IP system.

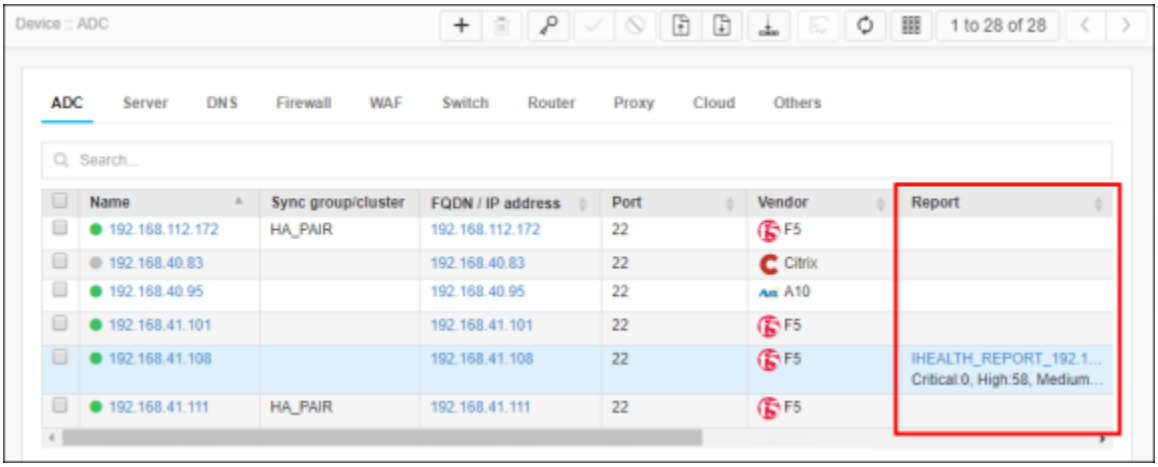
- iHealth reports can be generated at the time you want to view or schedule it in advance through the Workflow.
- To generate an iHealth report, ensure that the proxy is configured in the Settings module. For detailed information, refer to the Proxy Settings section of Platform User guide.

To generate and download an iHealth report:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Select the checkbox beside the ADC device for which you want to generate an iHealth report.
3. Click the (**Generate iHealth Report**) button in the Command bar.

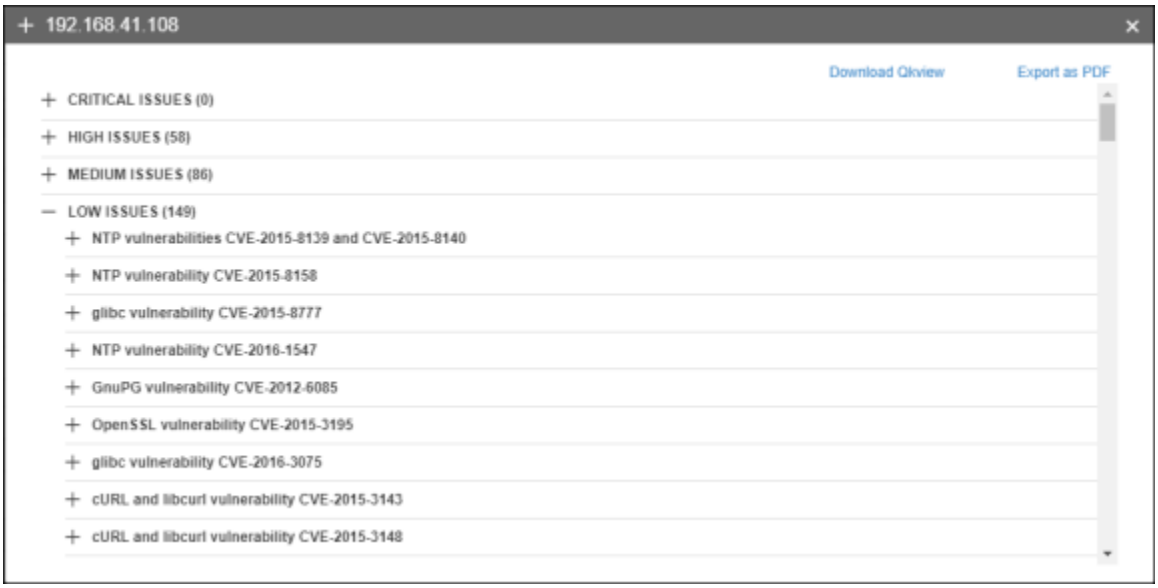
> **Note:**
> You might need to scroll to the right to see the Report column.

4. On the **iHealth Report Generate** screen that pops up, enter the case number for the report, then click **Generate**.
5. When the report is generated, it appears as a link in the **Report** column on the Device: ADC screen.
6. Click the **IHEALTH_REPORT** link.

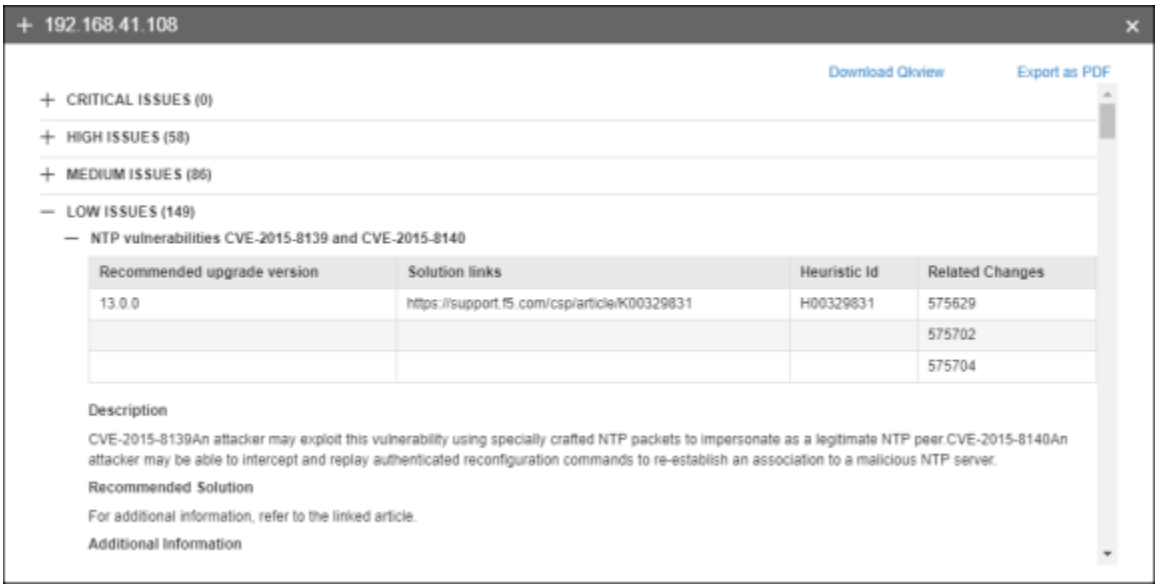


7. The iHealth report screen that pops up lists all of the current issues with the device, classified by their severity: critical, high, medium, or low.
8. Click any of the + (**Expand**) icons beside a severity level to view the issues within the corresponding category.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



9. Within each severity level, click the name of a specific issue to view complete details, a recommended solution, and additional information about it.



10. (Optional) Download the entire iHealth report as a Qkview file by clicking the **Download Qkview** link or as a PDF file by clicking the **Export as PDF** link, both of which appear in the top right corner of the screen.

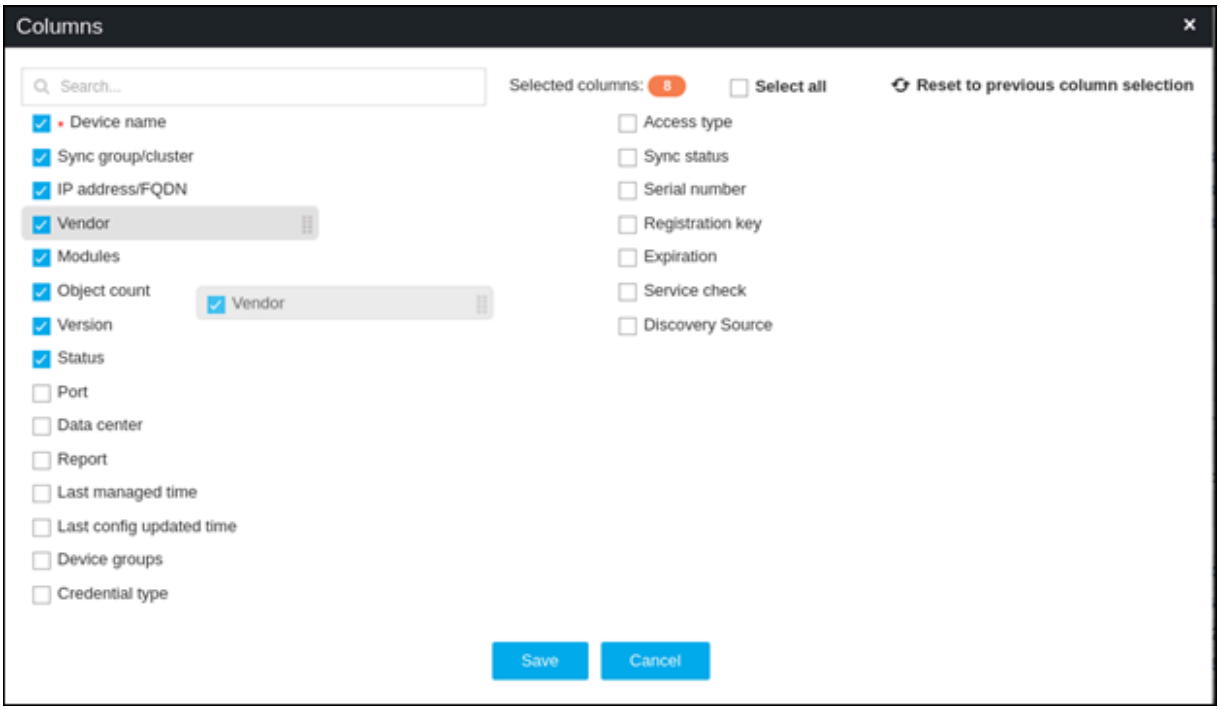
Customizing Columns

The columns in the Device Inventory page are highly customizable as per the user's convenience. Any column can be hidden, added, or alter the order of the columns.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To customize columns:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the  (**Columns**) button in the Command bar.
   The **Columns** pop-up opens.



3. In the Columns pop-up, you can modify the columns by doing any of the following or in the combination:

   a. Select or deselect the desired columns to be displayed in the **ADC** tab.

   b. Alter the order of the column by dragging the column name to the desired order.

   c. Select all the available columns by clicking the **Select all** checkbox.

   d. Reset to the previous column selection by clicking the "**Reset to previous column selection**".

4. Click **Save**.

> **Note:**
> To discard the changes, click **Cancel**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Configuration Sync between AppViewX and Device

The device configuration will be fetched into AppViewX periodically. By default, this process happens automatically at 12:00 AM everyday. The time and days can be configured as per the custom demand. For more detail, see Settings.

Onboard Device

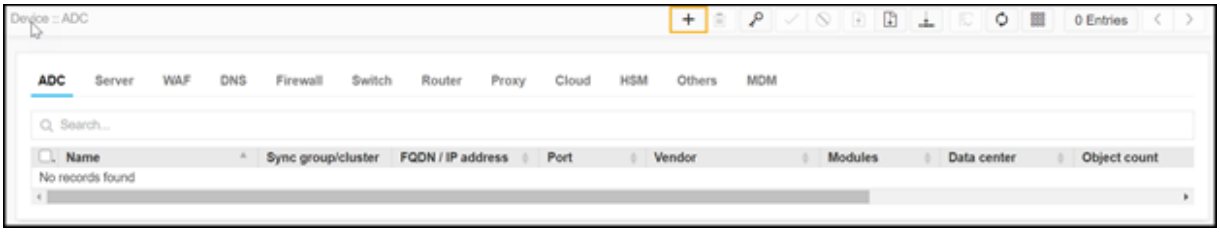
Discover/Onboard an ADC Device

Onboard the supported ADC vendor devices (Hardware, Cloud, and Software) into the AppViewX inventory using the IP Address/FQDN. AppViewX will initiate the communication using the provided credentials and Discover the Applications/Objects along with their configuration that are hosted on the devices. The Discovered Applications can be accessed within the product.

For vendor-specific device addition, refer to Discover/Onboard ADC Device for Vendor.

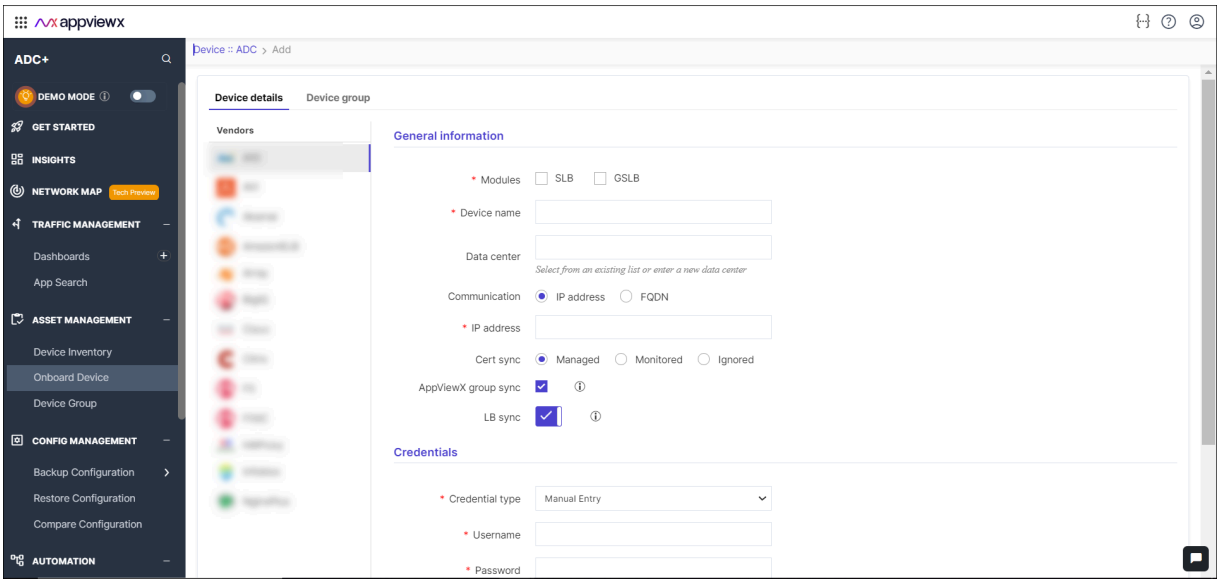
To onboard a device into Device Inventory:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. In the **ADC** tab, click add + button located upper right corner.



   The **Add** page appears.
3. Enter or select the field information in the **General Information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



General information Section - Field and Description Table

Name	Type	Description	Validation
Module	Check box	SLB / GSLB Module.	NA
Device name	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Text	Datacenter name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
Communication	Radio button	Devices can be accessed using an IP address or FQDN.	NA
IP Address	Text	The IPv4 address of the device.	The IP address should be in the right format.
FQDN	Text	The FQDN of the device.	the FQDN should be in a valid format.
Cert Sync	Radio button	Managed:	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
		The certificates of the device can be managed. Monitored: The certificates of the device can be monitored. Ignored: The certificate sync can be ignored.	
AppViewX Group Sync	Check box	This should be enabled if the user wants to sync the devices within the device group.	NA
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

4. Enter or select the field information in the **Credentials** section:

Credentials - Field and Description Table

Field	Description

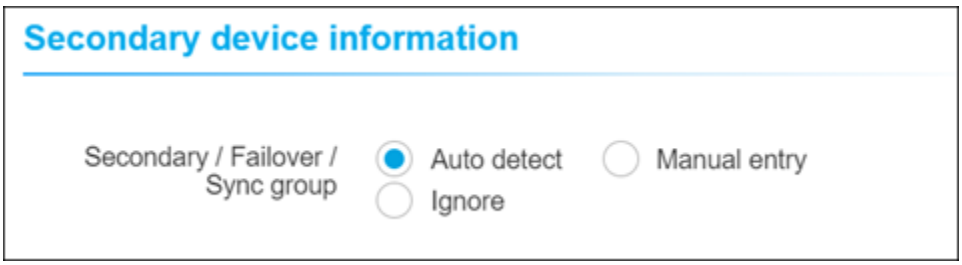
***Credential type**	Credentials can be manually provided or stored as a one-time entry onto the credential library and referred at the time of device addition. Select one of the following credential types from the drop-down list:
	• **Manual Entry** - The user name and password of the device need to be entered with device details. By default, the Manual Entry option is selected.
	• **AppViewX Credential List** - The user name and password can be added to the List and that entry can be referred to during device addition. The credential lists are integrated within AppViewX application for the secured authentication. The available credential lists are:
	  • Credential List - AppViewX
	  • Credential List - CyberArk

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• Credential List - Thycotic • Credential List - HashiCorp To create a credential list, see Creating Credential List in the *Platform User Guide*.
***Username**	Username for the ADC device when you select the **Manual Entry** credential type.
***Password**	Valid password for the ADC device when you select the **Manual Entry** credential type. **Note:** Use strong passwords for secure device communication. Your Passwords can be of any length with a combination of alpha-numerical, symbols, and special characters.
Enable password	For the **AppViewX Credential List** credential type, select your credential list with a CYBERARK username and app ID from the dropdown list.
**: Mandatory fields*	

Note:
The asterisk (*) symbol indicates mandatory fields.

5. Enter or select the field information in the **Secondary device information** section as follows:



- **Auto-Detect** - This option will automatically detect the corresponding secondary devices and add it as a new entry into AppViewX inventory using the Primary device's credential.
- **Manual Entry** - This selection will enable you to manually add Secondary devices with a Sync-group name entered for reference. This name will be used to identify the pairs in the inventory. Follow similar steps.
- **Ignore** - This option can be enabled if you need to ignore the detection of the secondary device associated with the current device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- In a typical network configuration, Application traffic will be handled by multiple ADC devices for ensuring the high availability of the application. This distribution of ADC servers will be meaningful during any disaster recovery and avoid a single point of failure. To achieve this, multiple ADC devices will be configured in Active/Standby or in failover groups. In this grouping, one ADC device will be serving the traffic and the rest of the devices in the group will act as a backup in case of a failure. The configuration will be in sync between these devices. At the same time, devices in a sync group can be in active-active mode also. You can manage one or more such Secondary devices (Failover/Standby devices) in inventory.
- By clicking the **Add** button, multiple devices can be added as secondary devices and all the devices will be available in the grid.
- By managing the primary and secondary devices in AppViewX during the device flips, traffic routing and management can be seamlessly handled in AppViewX.

6. Click **Save** to add an ADC device.

Note:

- To discard the changes, click **Cancel**.
- Repeat the same steps to add more ADC device(s).

Discover/Onboard a DNS Vendor Device

Onboard the supported DNS vendor devices to the AppViewX inventory using the IP Address/FQDN. AppViewX will initiate the communication using the provided credentials and subsequently manage them within the DNS Inventory. This functionality is particularly useful for DNS automation use cases.

Supported DNS Vendors: BIND, Infoblox, and QIP

To onboard a DNS vendor device into Device Inventory:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **DNS** tab, click add + button located upper right corner.
   The **Add** page appears.
3. Select a preferred vendor from the **Vendors** list on the left side.
4. Enter or select the field information in the **Add** page on the various sections.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. Click **Save** to add a DNS device.

> **Note:**
> • To discard the onboarding of DNS device, click **Cancel**.
> • Repeat the same steps to add more DNS device(s).

Discover/Onboard a WAF Vendor Device

Onboard the supported WAF vendor devices to the AppViewX inventory using the IP Address/FQDN. AppViewX will initiate the communication using the provided credentials and subsequently manage them within the WAF Inventory. This functionality is particularly useful for WAF automation use cases.

The WAF vendor devices can only be managed through Device Inventory, and WAF enablement has been restricted in the Control Center from AppViewX v24.1.0.0 Release.

On-Prem:

• If an ADC+ license is present, parsing of objects will occur.
• If there is no ADC+ license, parsing of objects will be skipped.

SaaS:

• Parsing of objects will be skipped regardless of the license status.

Supported DNS Vendors: Barracuda, F5, Imperva, and Cloudflare

Barracuda

Before You Begin

Prerequisites

Prerequisites for Managing Barracuda devices on AppViewX are:

• **IP Address/FQDN**: IP Address/FQDN
• **User Privilege**: Username and password, API access
• **Credential List**: AppViewX, CyberArk, HashiCorp, Thycotic, and BeyondTrust
• **Enable Password**: No
• **License Check**: Yes

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Services and Ports for AppViewX Communication**: A port is required to configure access to APIs.

> **Note:**
> Any port can be configured as needed.

- **Internet Access/Proxy if Required**: Not required
- **Location from which Certificates are discovered if managed**: Not applicable

Adding Barracuda Device

1. Go to (**Menu**) > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **WAF** tab.
3. Click the **(Add)** icon.
   The **Device details** page is displayed.
4. Select **Barracuda** from the Vendors list.
5. In the **General Information** section, select/enter the details as follows.

Fields	Description
Platform	Select the platform from the dropdown field • Azure • AWS • On-Prem
***Device name**	Enter a unique name for the device to be onborded.
Communication	Select from *IP address* and *FQDN*.
***IP Address**	This field is displayed if **Communication = IP Address** Enter the IP address of the device to be onboarded.
***FQDN**	This field is displayed if **Communication = FQDN** Enter the fully qualified domain name of the device to be onboarded.
***Rest port**	Enter the rest port number.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
Cert Sync	Choose from any of the following: • *Managed* - AppViewX performs the config fetch operations and the certificates are discovered and managed in the inventory. CLM actions (push & bind, rollback etc.) can be performed on them. • *Monitored* - AppViewX performs the config fetch operations and the certificates are downloaded in the inventory in the read-only state. CLM actions cannot be performed on them. • *Ignored* - AppViewX only performs the config fetch operations for the devices. There is no certificate discovery performed.
**: Mandatory fields*	

6. In the **Credentials** section, select/enter the details as indicated below. The credentials entered in this section are used to authenticate the session between the AppViewX node and the WAF device.

Fields	Description
***Credential Type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.
***Access type**	Select the method to access the Barracuda. API is the default value.
***Username**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the username is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated username for Barracuda.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
	If the Credential Type is selected as **Credential List - Appviewx**, the password is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated password to access Barracuda.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk** The dropdown contains a list of credentials stored in AppViewX.
**: Mandatory fields*	

7. In the **Secondary device information** section, select/enter the details as follows.

Fields	Description
Secondary / Alternate device	Select either *Auto-detect* or *Manual entry*.
***Device name**	Enter a unique name for the secondary device to be onborded.
Communication	Select from *IP address* and *FQDN*.
***IP Address**	This field is displayed if **Communication = IP Address** Enter the IP address of the secondary device to be onboarded.
***FQDN**	This field is displayed if **Communication = FQDN** Enter the fully qualified domain name of the secondary device to be onboarded.
***Rest port**	Enter the rest port number of the secondary device.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.
**: Mandatory fields*	

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

In the **Secondary device credentials** section (displayed only if **Secondary / Alternate device** = **Manual entry** select/enter the details as follows.

Fields	Description
***Use same credentials**	The *Enable* checkbox is used to fix the credentials type to either *Credential List - Appviewx* or *Credential List - CyberArk* based on its selection.
***Credential type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.
***Access type**	Select the method to access the secondary device. API is the default value.
***Username**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the username is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated username for the secondary device.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the password is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated password to access the secondary device.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk**

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
	The dropdown contains a list of credentials stored in AppViewX.
**: Mandatory fields*	

8. (*Optional step*) Click **Add**. This button is displayed only if **Secondary device information** is **Manual entry**. More than one devices can be configured. (use the **Update** and **Delete** buttons to make changes or remove the devices added.)
9. Click **Save**.
   The device is onboarded successfully.

What's Next

- Troubleshoot Onboarding Issues

Cloudflare

Before You Begin

Prerequisites

Prerequisites for Managing Cloudflare devices on AppViewX are:

- **IP Address/FQDN**: Account name
- **User Privilege**: NA
- **Credential List**: AppViewX, CyberArk, HashiCorp, Thycotic, and BeyondTrust
- **Enable Password**: No
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: Not applicable
- **Internet Access/Proxy if Required**: Not required
- **Location from which Certificates are discovered if managed**: Not applicable

Adding Cloudflare Device

1. Go to (**Menu**) > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **WAF** tab.
3. Click the **(Add)** icon.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Device details** page is displayed.

4. Select **Cloudflare** from the Vendors list.
5. In the **General Information** section, select/enter the details as follows.

Fields	Description
***Account name**	A user defined unique name to identify the server. No special characters other than '-' , '_' , '.' , '*' , '|' , '!' and spaces are not allowed.
Data center	Enter the data center routing. It holds all the SSL related information that is to be retrieved from the Cloudflare device.
Cert Sync	Choose from any of the following: • *Managed* - AppViewX performs the config fetch operations and the certificates are discovered and managed in the inventory. CLM actions (push & bind, rollback etc.) can be performed on them. • *Monitored* - AppViewX performs the config fetch operations and the certificates are downloaded in the inventory in the read-only state. CLM actions cannot be performed on them. • *Ignored* - AppViewX only performs the config fetch operations for the devices. There is no certificate discovery performed.
**: Mandatory fields*	

6. In the **Credentials** section, select/enter the details as indicated below. The credentials entered in this section are used to authenticate the session between the AppViewX node and the WAF device.

Fields	Description
***Credential Type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.
***Access type**	Select the method to access the Cloudflare.
***Token based Authentication**	Select the checkbox to enable the token based authentication.
API Token	Enter the API token if the token based authentication is enabled.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***Username**	This field will be visible if the Credential Type is selected as **Manual entry and token based authentication is disabled** . Enter the designated username for Cloudflare.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry and token based authentication is disabled** . Enter the designated password to access Cloudflare.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk** The dropdown contains a list of credentials stored in AppViewX.
**: Mandatory fields*	

7. Click **Save**.
   The device is onboarded successfully.

F5

Before You Begin

Prerequisites

Prerequisites for Managing F5 devices on AppViewX are:

- **IP Address/FQDN**: IP Address/FQDN
- **User Privilege**: Username and password, and API access
- **Credential List**: AppViewX, CyberArk, HashiCorp, Thycotic, and BeyondTrust
- **Enable Password**: Not required
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: Not required
- **Internet Access/Proxy if Required**: Not required
- **Location from which Certificates are discovered if managed**: Not applicable

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Adding F5 Device

1. Go to (**Menu**) > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **WAF** tab.
3. Click the **(Add)** icon.
   The **Device details** page is displayed.
4. Select **f5** from the Vendors list.
5. In the **General Information** section, select/enter the details as follows.

Fields	Description
Platform	Select the platform from the dropdown field. ASM is the default value.
***Device name**	Enter a unique name for the device to be onborded.
Communication	Select from *IP address* and *FQDN*.
***IP Address**	This field is displayed if **Communication = IP Address** Enter the IP address of the device to be onboarded.
***FQDN**	This field is displayed if **Communication = FQDN** Enter the fully qualified domain name of the device to be onboarded.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.
Cert Sync	Choose from any of the following: • *Managed* - AppViewX performs the config fetch operations and the certificates are discovered and managed in the inventory. CLM actions (push & bind, rollback etc.) can be performed on them. • *Monitored* - AppViewX performs the config fetch operations and the certificates are downloaded in the inventory in the read-only state. CLM actions cannot be performed on them. • *Ignored* - AppViewX only performs the config fetch operations for the devices. There is no certificate discovery performed.
**: Mandatory fields*	

6. In the **Credentials** section, select/enter the details as indicated below. The credentials entered in this section are used to authenticate the session between the AppViewX node and the WAF device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***Credential Type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.
***Access type**	Select the method to access the F5 device. API is the default value.
***Username**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the username is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated username for Barracuda.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the password is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated password to access Barracuda.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk** The dropdown contains a list of credentials stored in AppViewX.
***Token based authentication**	Select the checkbox to enable the token based authentication.
**: Mandatory fields*	

7. In the **Secondary device information** section, select/enter the details as follows.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
Secondary / Alternate device	Select either *Auto-detect* or *Manual entry*.
***Device name**	Enter a unique name for the secondary device to be onborded.
Communication	Select from *IP address* and *FQDN*.
***IP Address**	This field is displayed if **Communication = IP Address** Enter the IP address of the secondary device to be onboarded.
***FQDN**	This field is displayed if **Communication = FQDN** Enter the fully qualified domain name of the secondary device to be onboarded.
***Rest port**	Enter the rest port number of the secondary device.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.
**: Mandatory fields*	

In the **Secondary device credentials** section (displayed only if **Secondary / Alternate device** = **Manual entry** select/enter the details as follows.

Fields	Description
***Use same credentials**	The *Enable* checkbox is used to fix the credentials type to either *Credential List - Appviewx* or *Credential List - CyberArk* based on its selection.
***Credential type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.
***Access type**	Select the method to access the secondary device. API is the default value.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***Username**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the username is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated username for the secondary device.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the password is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated password to access the secondary device.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk** The dropdown contains a list of credentials stored in AppViewX.
**: Mandatory fields*	

8. (*Optional step*) Click **Add**. This button is displayed only if **Secondary device information** is **Manual entry**. More than one devices can be configured. (use the **Update** and **Delete** buttons to make changes or remove the devices added.)
9. Click **Save**.
   The device is onboarded successfully.

What's Next

- Troubleshoot Onboarding Issues

Imperva Securesphere

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before You Begin

Prerequisites

Prerequisites for Managing Imperva Securesphere devices on AppViewX are:

- **IP Address/FQDN**: IP Address/FQDN
- **User Privilege**: Username and password, and API access
- **Credential List**: AppViewX, CyberArk, HashiCorp, Thycotic, and BeyondTrust
- **Enable Password**: Not required
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: A port is required to configure access to APIs.

> **Note:**
> Any port can be configured as needed.

- **Internet Access/Proxy if Required**: Depends on the use case; not mandatory
- **Location from which Certificates are discovered if managed**: Not applicable

Addding Imperva Securesphere Device

Prerequisites:

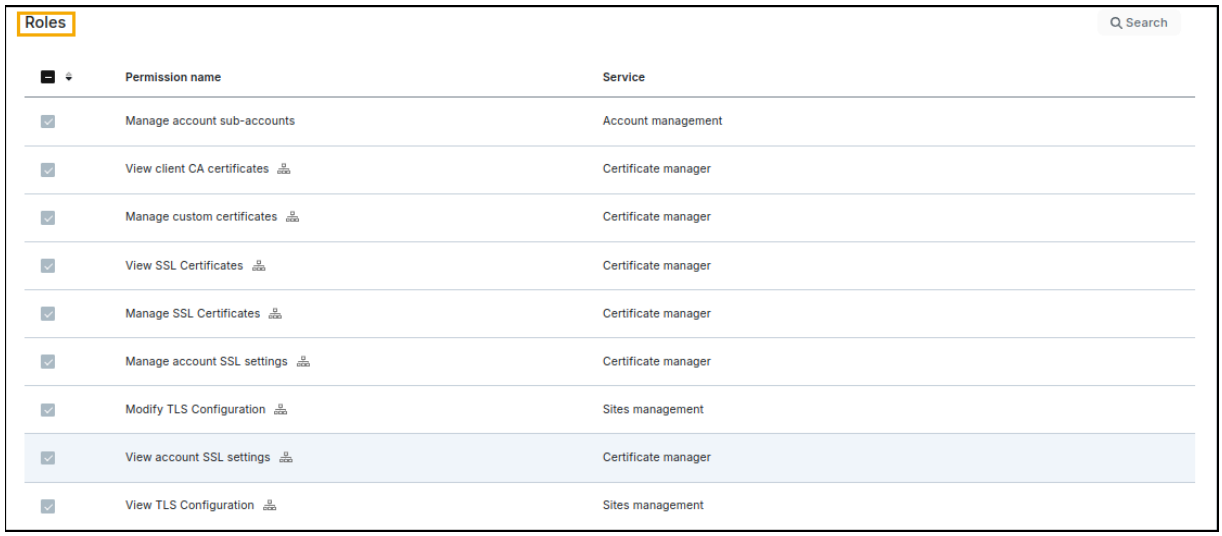
This prerequisite applies only to **Imperva SaaS**. To efficiently perform Certificate Lifecycle Management (CLM) operations, you must configure user account roles and permissions within Imperva SaaS. Administrator or Service account access to the Imperva SaaS portal to create a new User and role.

Define and manage roles to assign the appropriate permissions for Imperva SaaS configuration sync, endpoint enrollment, push, and binding from AppViewX to Imperva SaaS.

Access the Roles page to create and manage roles. This page is available to the account administrator or users with the 'Manage user roles' permission.

1. On the sidebar, click **User Management** > **Roles** > **New Role**.
2. Create a new role with the below permissions.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Permissions:

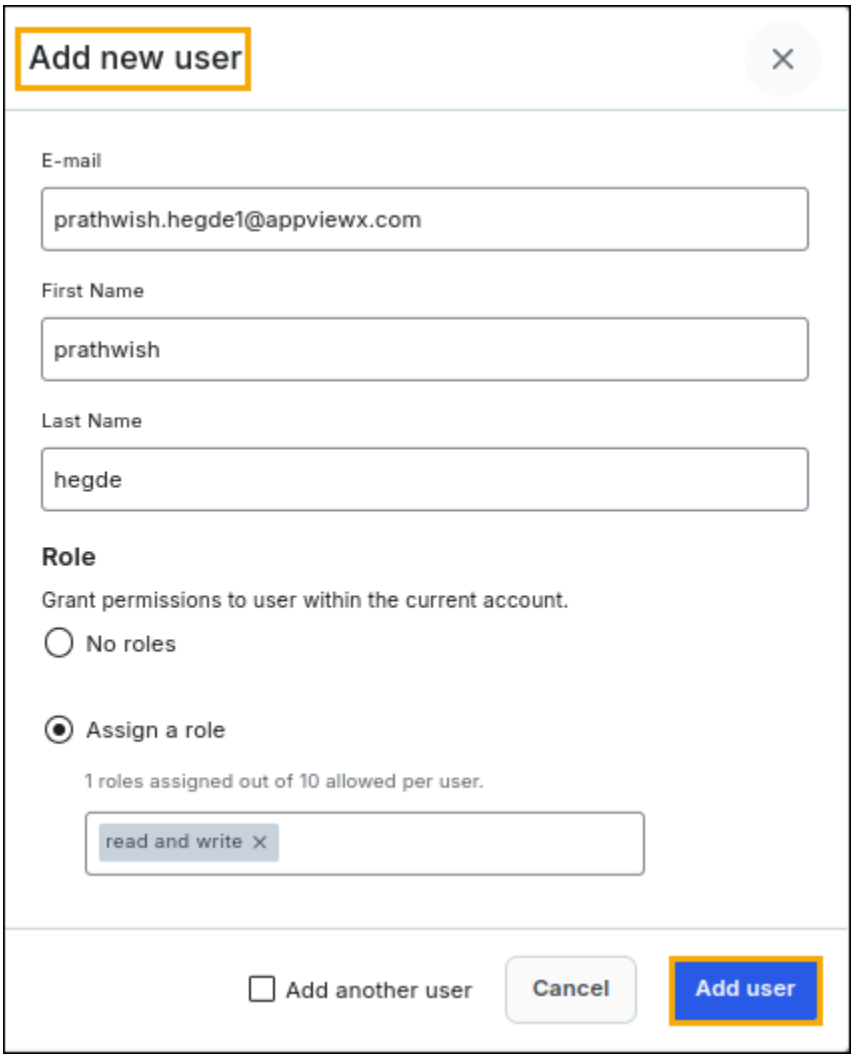
- View client CA certificates
- Manage custom certificates
- Manage account SSL settings
- View account SSL settings
- View TLS Configuration
- View SSL Certificates
- Manage SSL Certificates
- Modify TLS Configuration
- Manage account sub-accounts

API Key Management

To use the API, the client must be authenticated by Imperva.

1. On the sidebar, click **User Management** > **Users**.
2. Click **Add New user** to create a new user, and then enter the details as required.

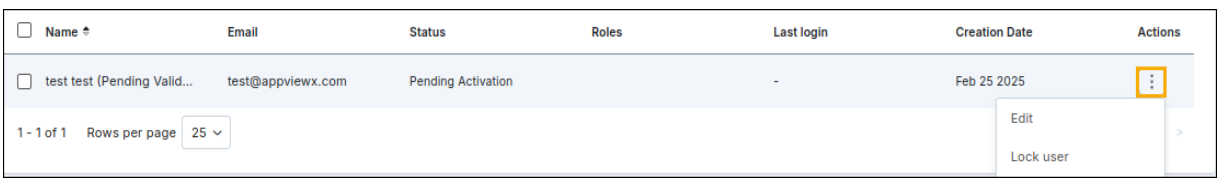
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



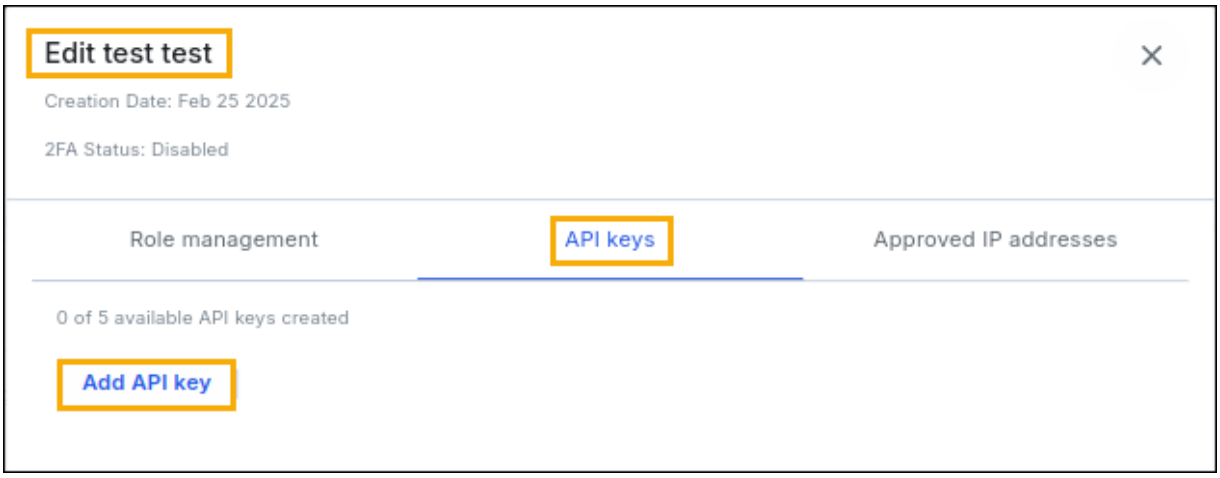
3. Click the **Add User** button.

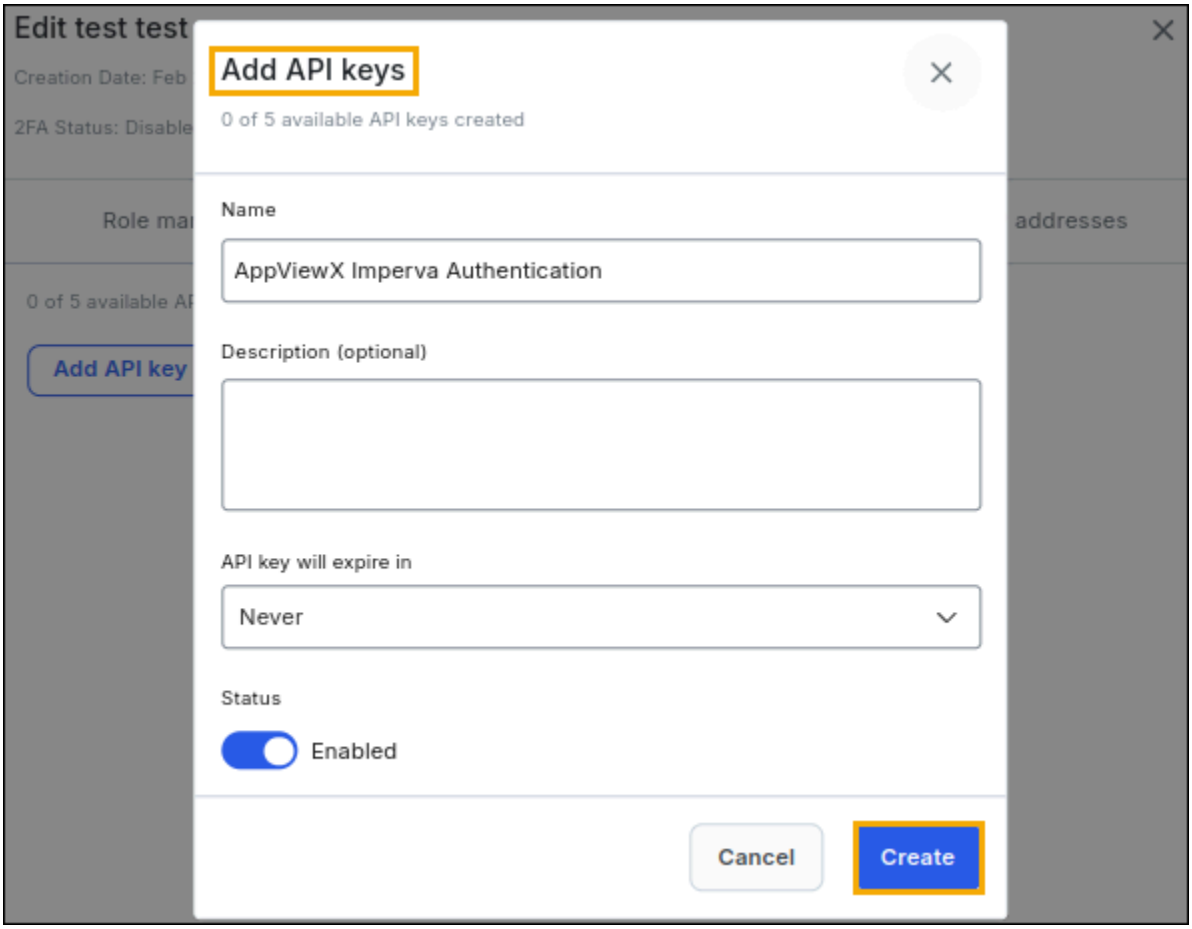
> **Note:**
> When a new user is added to an account, a verification email is sent to their registered address. The user must click the link in the email to verify their address and set a login password.

4. Once the User account is created, select 'Edit,' navigate to the 'API Keys' category, and generate an API ID and API Key.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.





5. Click **Create**. It creates an API ID and API Key.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
Copy the API key now, as it will no longer be accessible once you close this window.

1. Go to (**Menu**) > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **WAF** tab.
3. Click the **(Add)** icon.
   The **Device details** page is displayed.
4. Select **Imperva Securesphere** from the Vendors list.
5. In the **General Information** section, select/enter the details as follows.

Note:
This field is displayed if **General Information** = **AWS**

Fields	Description
Platform	Select the platform from the dropdown field. The available platforms are: • **AWS** • **SaaS**
***Device name**	Enter a unique name for the device to be onboarded.
Communication	This field is displayed if **Platform = AWS**Select *IP address* or *FQDN*.
***IP Address**	This field is displayed if **Platform = AWS** and **Communication = IP Address**. Enter the IP address of the device to be onboarded.
***FQDN**	This field is displayed if **Communication = FQDN**. Enter the fully qualified domain name of the device to be onboarded.
***Rest port**	Enter the rest port number.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.
Proxy Required	Enable this field if the device communication needs to happen via a proxy. The proxy details configured in general settings will be used for communication.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
Cert Sync	Choose from any of the following: • *Managed* - AppViewX performs the config fetch operations and the certificates are discovered and managed in the inventory. CLM actions (push & bind, rollback etc.) can be performed on them. • *Monitored* - AppViewX performs the config fetch operations and the certificates are downloaded in the inventory in the read-only state. CLM actions cannot be performed on them. • *Ignored* - AppViewX only performs the config fetch operations for the devices. There is no certificate discovery performed.
**: Mandatory fields*	

Note:
This field is displayed if **General Information** = **SaaS**

Limitation: Imperva SaaS does not offer an API for exporting certificate content and only provides certificate metadata. Consequently, certificates from Imperva SaaS cannot be directly discovered. However, if a certificate is already present in the AppViewX inventory, a connector will be created using the serial number from the metadata.

Note:
AppViewX supports certificate enrollment and deployment to the primary Imperva SaaS account and its associated sub-accounts.

Fields	Description
Platform	Select the platform from the dropdown field. The available platforms are, • **AWS** • **SaaS** **Note:** By default, **AWS** platform is selected.
***Device name**	Enter a unique name for the device to be onborded.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***FQDN**	This field is displayed if **Communication = FQDN** Enter the fully qualified domain name of the device to be onboarded.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.
Proxy Required	Enable this field if the device communication needs to happen via a proxy. The proxy details configured in general settings will be used for communication.
Cert Sync	Choose from any of the following: • *Managed* - AppViewX performs the config fetch operations and the certificates are discovered and managed in the inventory. CLM actions (push & bind, rollback etc.) can be performed on them. • *Monitored* - AppViewX performs the config fetch operations and the certificates are downloaded in the inventory in the read-only state. CLM actions cannot be performed on them. • *Ignored* - AppViewX only performs the config fetch operations for the devices. There is no certificate discovery performed.
**: Mandatory fields*	

6. In the **Credentials** section, select/enter the details as indicated below. The credentials entered in this section are used to authenticate the session between the AppViewX node and the WAF device.

Note:
This field is displayed if **General Information** = **AWS**

Fields	Description
***Credential Type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***Access type**	Select the method to access the Imperva SecureSphere. API is the default value.
***Username**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the username is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated username for Imperva Sourcesphere.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the password is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated password to access Barracuda.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk** The dropdown contains a list of credentials stored in AppViewX.
**: Mandatory fields*	

Note:
This field is displayed if **General Information** = **SaaS**

Fields	Description
***Credential Type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box.
***Access type**	Select the method to access the Imperva SecureSphere. API is the default value.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***API ID**	Enter the valid API ID.
***API Key**	Enter the valid API ID.
**: Mandatory fields*	

7. In the **Secondary device information** section, select/enter the details as follows.

> **Note:**
> This field is displayed if **General Information** = **AWS**

Fields	Description
Secondary / Failover / Sync group	Select either *Ignore* or *Manual entry*.
***Device name**	Enter a unique name for the secondary device to be onborded.
Communication	Select from *IP address* and *FQDN*.
***IP Address**	This field is displayed if **Communication = IP Address** Enter the IP address of the secondary device to be onboarded.
***FQDN**	This field is displayed if **Communication = FQDN** Enter the fully qualified domain name of the secondary device to be onboarded.
***Rest port**	Enter the rest port number of the secondary device.
Data center	Enter the data center name. It holds all the SSL related information that is to be retrieved from the server.
**: Mandatory fields*	

In the **Secondary device credentials** section (displayed only if **Secondary / Alternate device** = **Manual entry** select/enter the details as follows.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fields	Description
***Use same credentials**	The *Enable* checkbox is used to fix the credentials type to either *Credential List - Appviewx* or *Credential List - CyberArk* based on its selection.
***Credential type**	Select the credential type from the dropdown. • *Manual entry* (default) - to directly add the credentials in the username and password text box. • *Credential List - Appviewx* - to use any credentials from the saved AppViewX Credential List. • *Credential List - CyberArk* - to use any credentials from the saved AppViewX CyberArk List.
***Access type**	Select the method to access the secondary device. API is the default value.
***Username**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the username is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated username for the secondary device.
***Password**	This field will be visible if the Credential Type is selected as **Manual entry**. If the Credential Type is selected as **Credential List - Appviewx**, the password is entered by default and field is read only. If the Credential Type is selected as **Credential List - Cyerark**, this field is not displayed. Enter the designated password to access the secondary device.
***Credential list**	This field will be visible only if the Credential type is selected as **Credential List - AppViewX** or **Credential List - CyberArk** The dropdown contains a list of credentials stored in AppViewX.
**: Mandatory fields*	

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

8. (*Optional step*) Click **Add**. This button is displayed only if **Secondary device information** is **Manual entry**. More than one devices can be configured. (use the **Update** and **Delete** buttons to make changes or remove the devices added.)
9. Click **Save**.
   The device is onboarded successfully.

What's Next

- Troubleshoot Onboarding Issues

Discover/Onboard an Other Network Device

Onboard the supported DNS vendor devices to the AppViewX inventory using the IP Address/FQDN. AppViewX will initiate the communication using the provided credentials and subsequently manage them within the DNS Inventory. This functionality is particularly useful for DNS automation use cases.

Onboard any supported network devices to the AppViewX inventory using their IP Address or FQDN. AppViewX will initiate communication with the devices using the provided credentials and manage them within the Others Inventory. This functionality is applicable for automating tasks involving various types of network devices.

To onboard a DNS vendor device into Device Inventory:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **Others** tab, click add + button located upper right corner.
   The **Add** page appears.
3. Enter or select the field information in the **Add** page on the various sections.
4. Click **Save** to add an other network device.

> **Note:**
> • To discard the onboarding of other network device, click the **Cancel** button.
> • Repeat the same steps to add more other network device(s).

Vendor Specific Discover/Onboard ADC Device

A10

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before You Begin

Prerequisites

Prerequisites for Managing A10 devices on AppViewX are:

- **Version Supported**: v2.7.x , v4.0.1, v4.1.x, v5.1.x , and v6.x
- **IP Address/FQDN**: IP address and FQDN
- **User Privilege**: CLI/WEB/xapi access and Read/Write access is required
- **Enable Password**: Required
- **License Check**: No
- **Services and Ports for AppViewX Communication**: 22, 80, and 443
- **Internet Access/Proxy if Required**: Not required
- **Location from which Certificates are discovered if managed**: No specific location

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

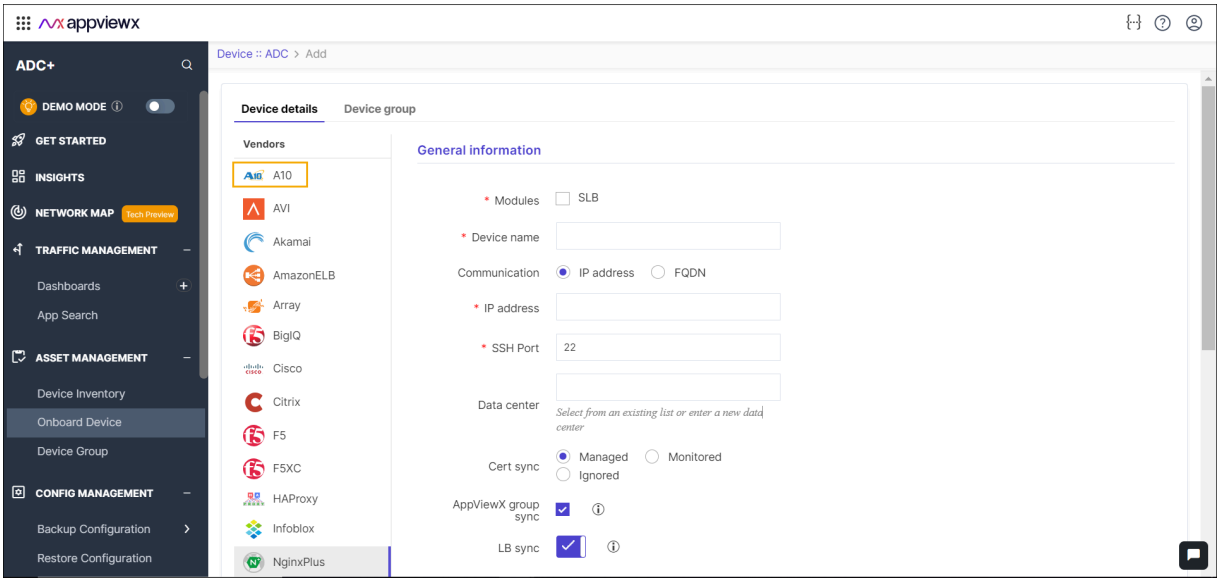
To associate ACF permissions with a role, see Platform User Guide.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Adding A10 Device

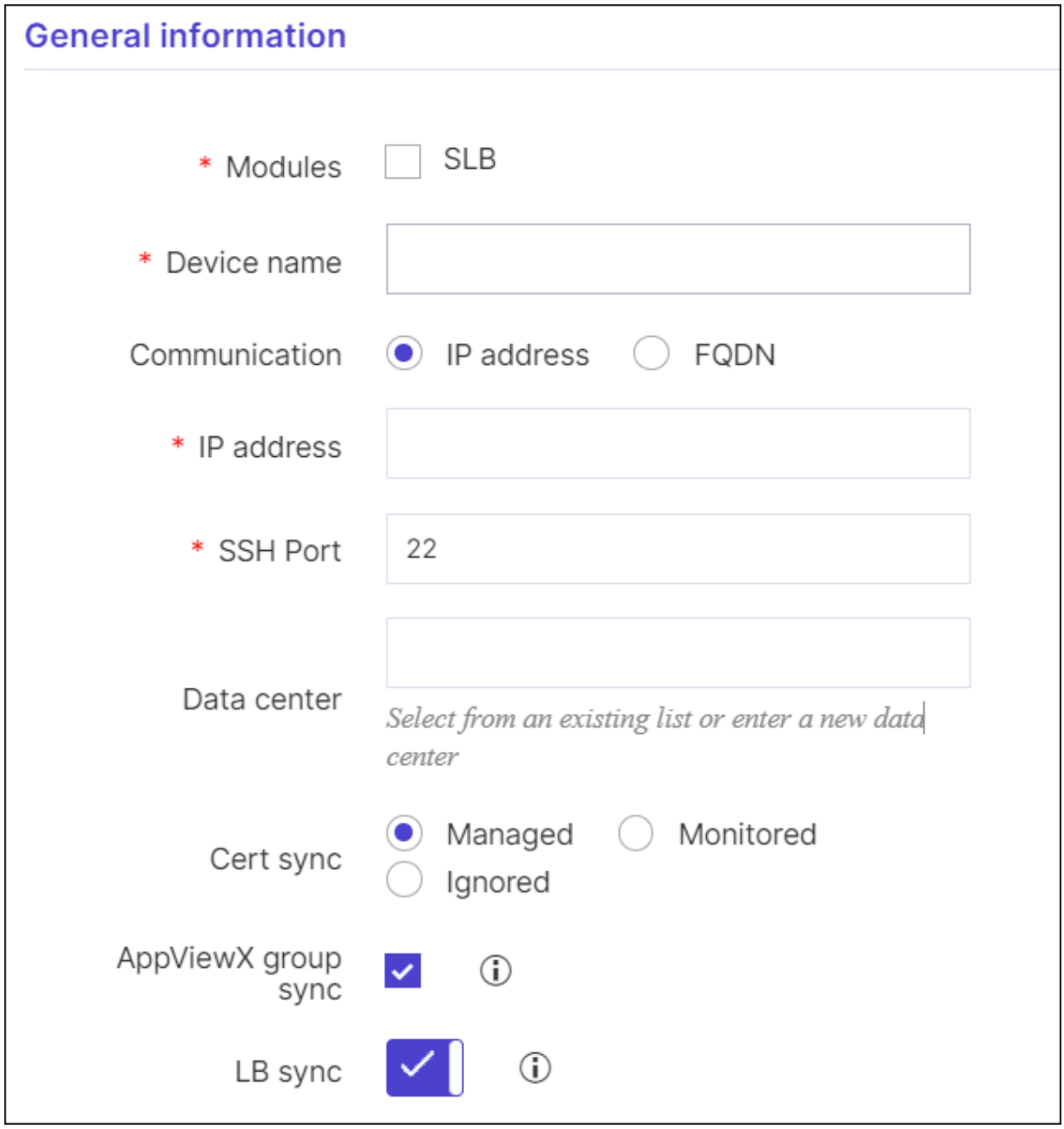
To add an A10 device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **A10** from the left sidebar.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



General information Section - Field and Description Table

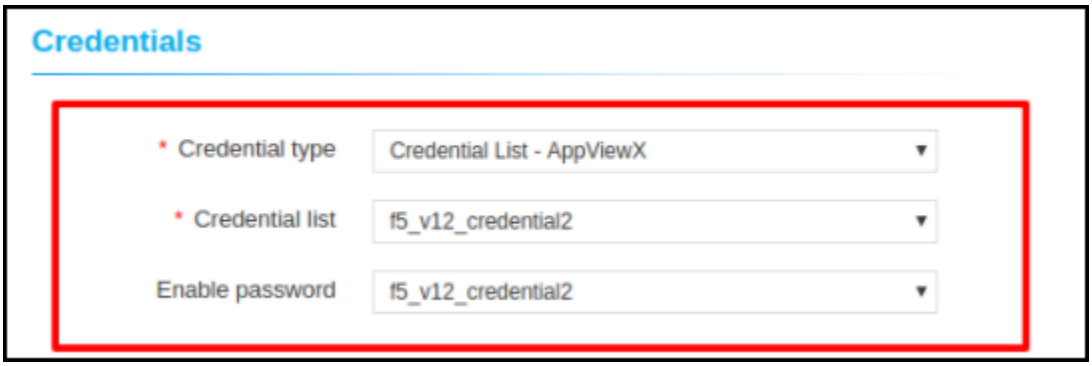
Name	Description
Module	SLB / GSLB Module.
Device name	Unique name of the device to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

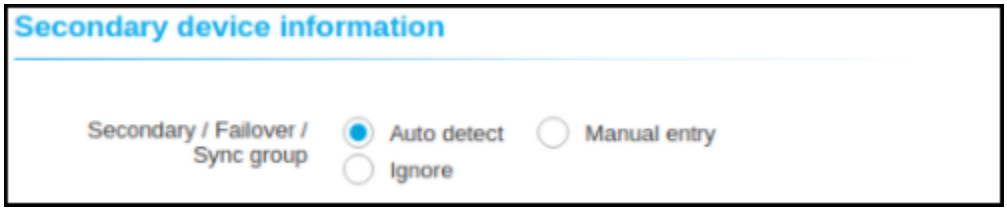
Name	Description
	Note: Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Datacenter name where the device is configured. The default value is Absecon. **Note:** Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
Communication	Devices can be accessed using an IP address or FQDN.
IP Address	The IPv4 address of the device. The IP address should be in the right format.
FQDN	The FQDN of the device. The FQDN should be in a valid format.
Cert Sync	Managed: The certificates of the device can be managed. Monitored: The certificates of the device can be monitored. Ignored: The certificate sync can be ignored.
AppViewX Group Sync	This should be enabled if the user wants to sync the devices within the device group.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
**: Mandatory fields*	

5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credential List**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



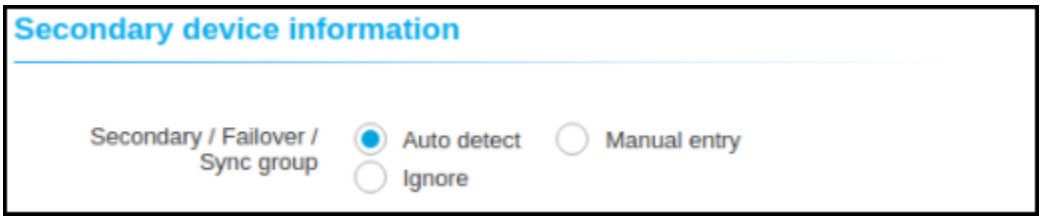
Credentials Section - Field and Description Table

Name	Description
***Credential type**	**Manual entry:** The user should enter the username and password.  **Credential List:** The user can select the credential details which are already stored in the credential inventory page. 
***Username**	If manual entry is selected, the user name should be entered by the user.
***Password**	If manual entry is selected, the password should be entered by the user.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
Enable password	Enable password of the A10 device.
**: Mandatory fields*	

6. Enter or select the field information in the **Secondary device information** section.



7. The following table provides the field description for adding ADC device details in the **Secondary device information** section:

Secondary device information Section - Field and Description Table

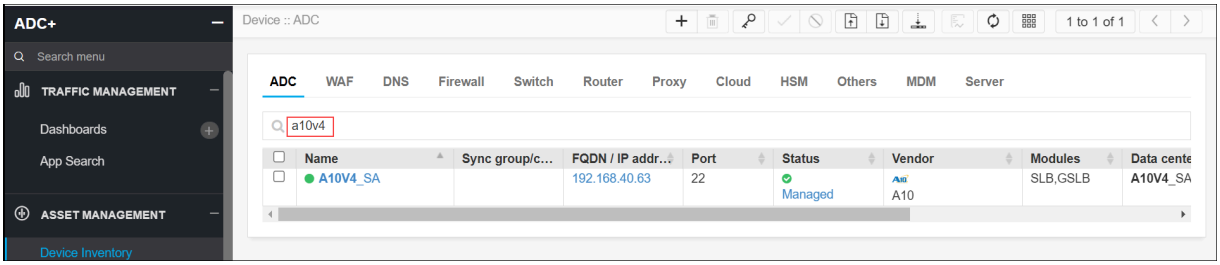
Name	Description
Secondary device information	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually. **Ignore**: The user can use this option to ignore the auto-detection of the peer devices.

8. Click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Validating the A10 Device Addition

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

AVI

Before You Begin

Prerequisites

Prerequisites for Managing AVI devices on AppViewX are:

- **Version Supported**: 16.2.x, 16.3.x, 17.1.x,17.2.x,18.2.x, 20.1.x, 22, 30, 31.x
- **IP Address/FQDN**: IP address and FQDN

> **Note:**
> Device should not be added with Cluster IP

- **User Privilege**: System Admin Role for all tenants.

> **Note:**
> SuperUser Access is required only to verify terminal check, not mandatory.

- **Enable Password**: Not applicable
- **License Check**: No
- **Services and Ports for AppViewX Communication**: 22, 443, 5514 (To receive syslog on AppViewX from AVI)

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Internet Access/Proxy if Required**: Not required
- **Location from which Certificates are discovered if managed**: No specific location (we use API's to get certificate details)

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

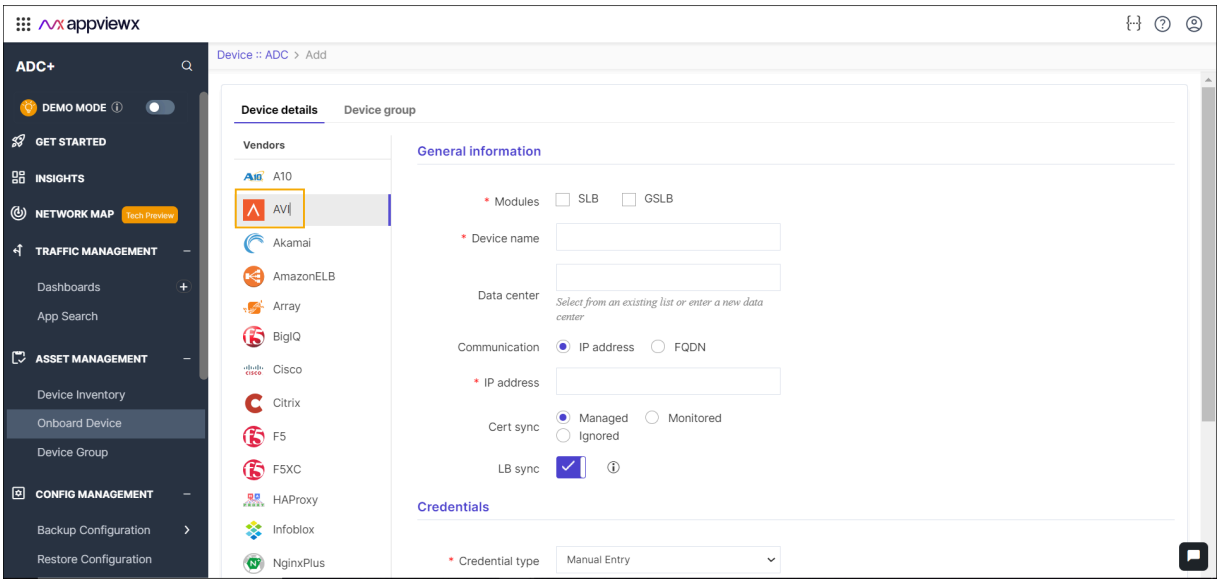
To associate ACF permissions with a role, see Platform User Guide.

Adding AVI Device

To add an AVI device:

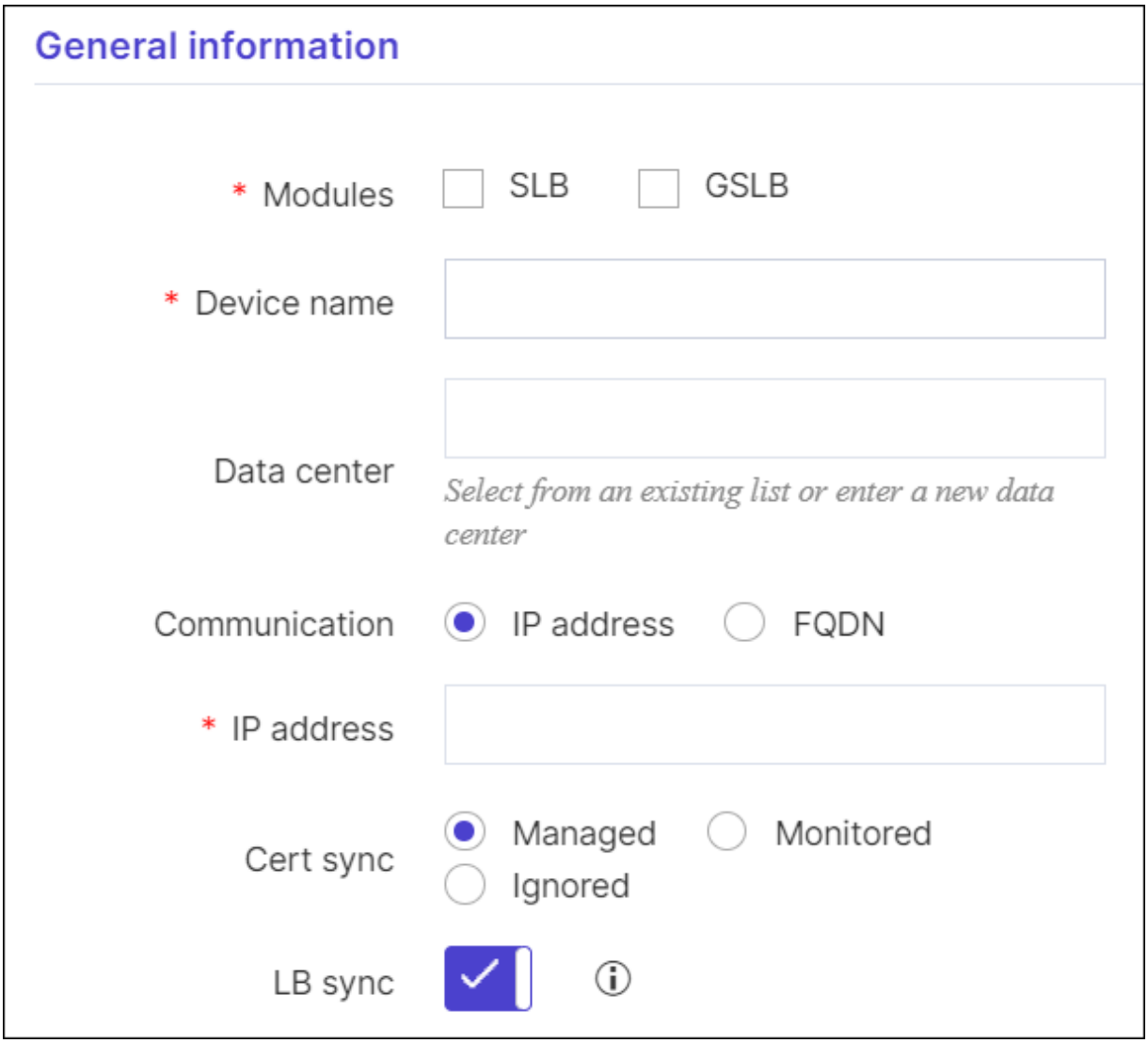
1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **AVI** from the left sidebar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



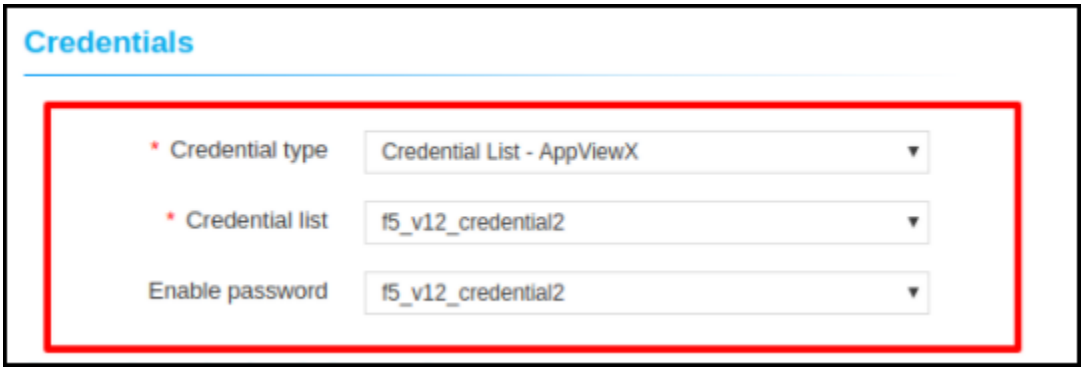
General information Section - Field and Description Table

Name	Description
***Module**	SLB / GSLB Module.
***Device name**	Unique name of the device to be added. **Note:** Device name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Datacenter name where the device is configured. The default value is Absecon.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

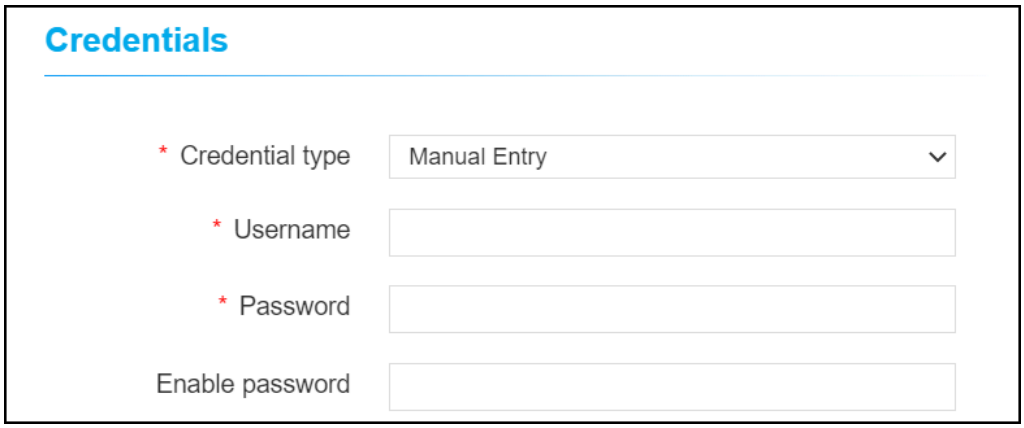
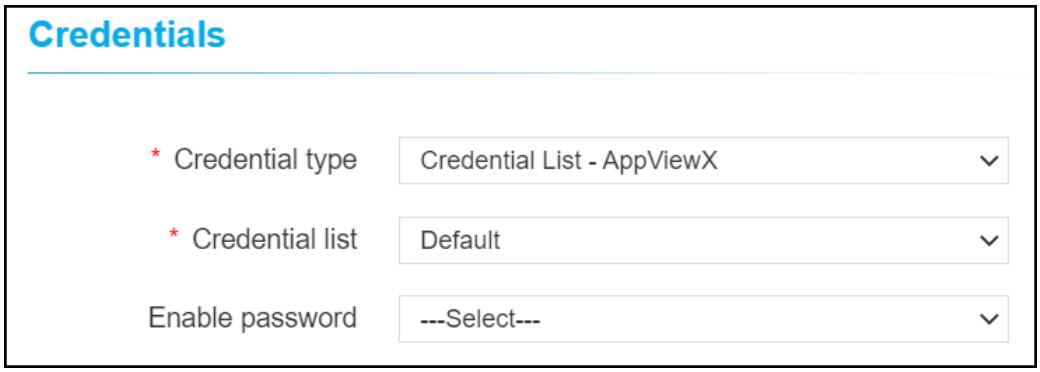
Name	Description
	Note: Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
Communication	Devices can be accessed using an IP address or FQDN.
***IP Address**	The IPv4 address of the device. **Note:** The IP address should be in the right format.
FQDN	The FQDN of the device. **Note:** The FQDN should be in a valid format.
Cert Sync	**Managed**: The certificates of the device can be managed. **Monitored**: The certificates of the device can be discovered and can only be monitored. **Ignored**: The certificate sync can be ignored.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
**: Mandatory fields*	

5. Enter or select the field information in the **Credentials** section.



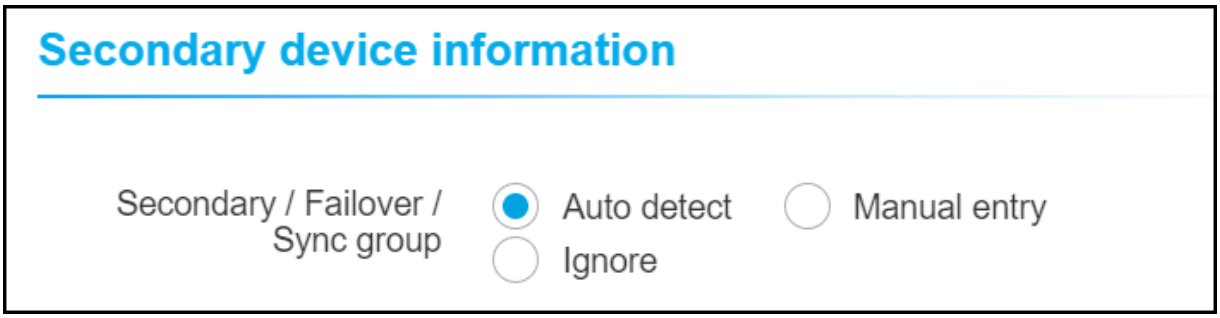
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Credentials Section - Field and Description Table

Name	Description
***Credential type**	**Manual entry**: The user should enter the username and password.  **Credential List**: The user can select the credential details which are already stored in the credential inventory page. 
***Username**	If manual entry is selected, the user name should be entered by the user.
***Password**	If manual entry is selected, the password should be entered by the user.
Enable password	Enable password of the AVI device.
**: Mandatory fields*	

6. Enter or select the field information in the **Secondary device information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Secondary device information Section - Field and Description Table

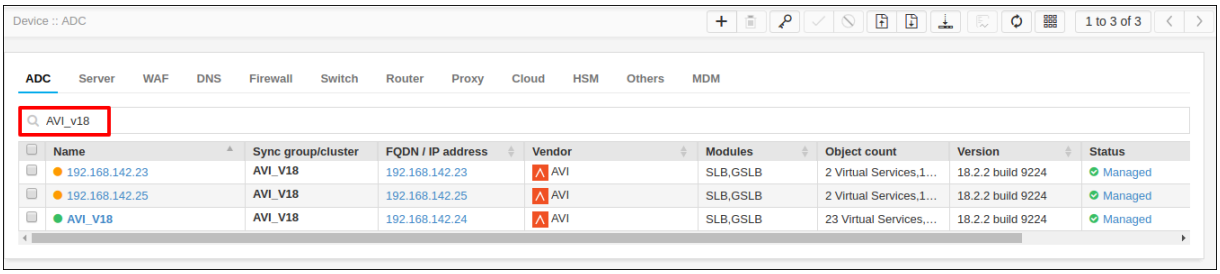
Name	Description
Secondary device information	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually. **Ignore**: The user can use this option to ignore the auto-detection of the peer devices.

7. Click **Save**.

Validating the AVI Device Addition

After adding the device, you can validate the device by searching the device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Check the Logs for transaction ID and details

Akamai

Before You Begin

Prerequisites

Prerequisites for Managing Akamai devices on AppViewX are:

- **User Privilege**: Client token, Client Secret, Access Token, Host URL
- **Enable Password**: Not applicable
- **License Check**: Not applicable
- **Services and Ports for AppViewX Communication**: 443 Outbound access from AppViewX is needed to manage the device and Perform Akamai Actions
- **Internet Access/Proxy if Required**: Not required
- **Location from which Certificates are discovered if managed**: Not applicable

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

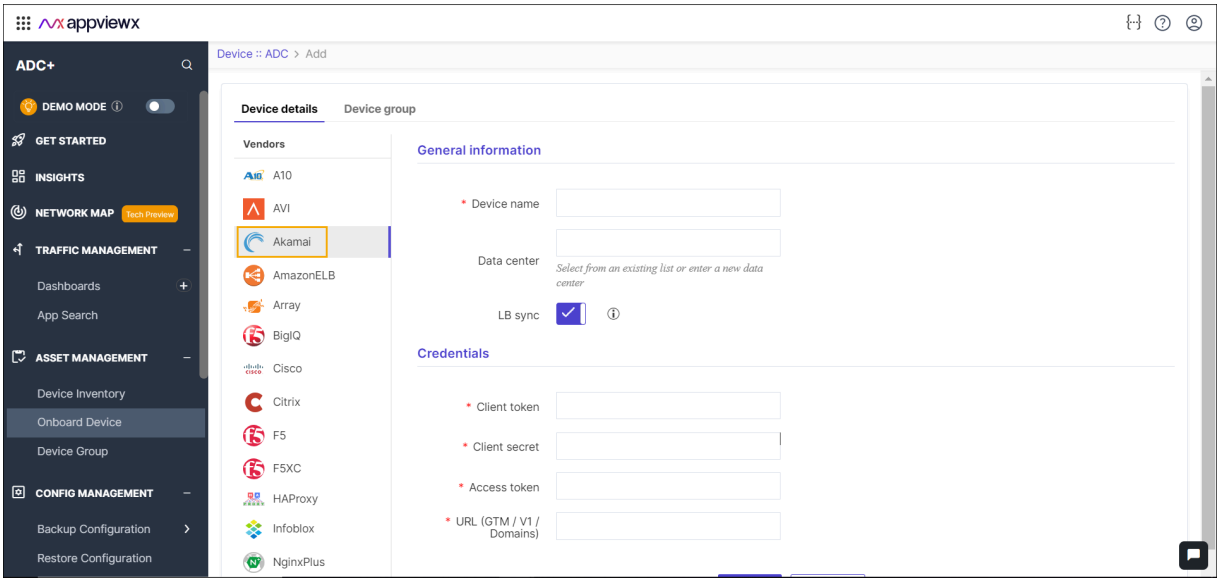
- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

To associate ACF permissions with a role, see Platform User Guide.

Adding Akamai Device

To add an Akamai device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **Akamai** from the left sidebar.



4. Enter the required details for the device addition.

Device Addition - Field and Description Table

Name	Description
***Device name**	Unique name of the device to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	Note: Device name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Datacenter name where the device is configured. The default value is Absecon. **Note:** Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
***Client token**	Client token of the Akamai device.
***Client secret**	Client secret of the Akamai device.
***Access token**	Access token of the device.
***URL**	Valid URL of the Akamai device.
**: Mandatory fields*	

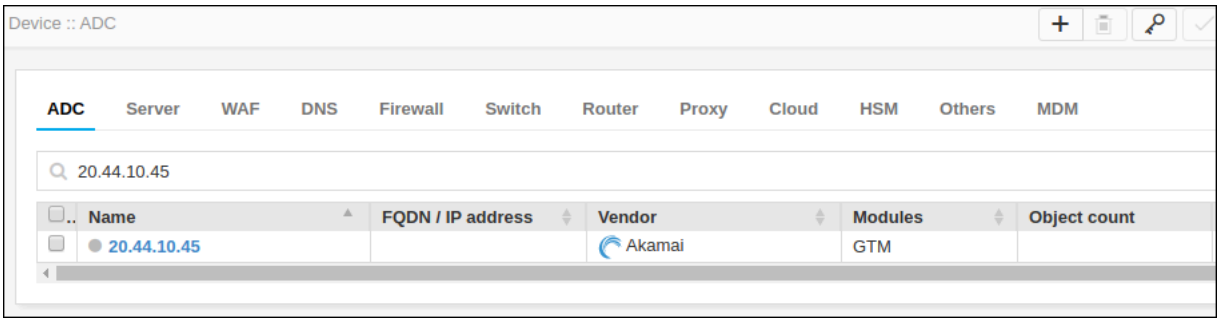
5. Click **Save**.

Validating the Akamai Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

Amazon ELB

Before You Begin

Prerequisites

Prerequisites for Managing Amazon ELB devices on AppViewX are:

- **Version Supported**: Only Classic and Application ELB is supported
- **IP Address/FQDN**: Not applicable
- **User Privilege**: Access Key, Secret Key, and Region information is required.
- **Enable Password**: Not applicable
- **License Check**: Not applicable
- **Services and Ports for AppViewX Communication**: Port 443, Outbound Internet access is required to manage the ELB in AWS
- **Internet Access/Proxy if Required**: Required
- **Location from which Certificates are discovered if managed**: Not applicable (Certificates that are assigned to the listeners will be discovered by AppViewX)

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Browsers	Version
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

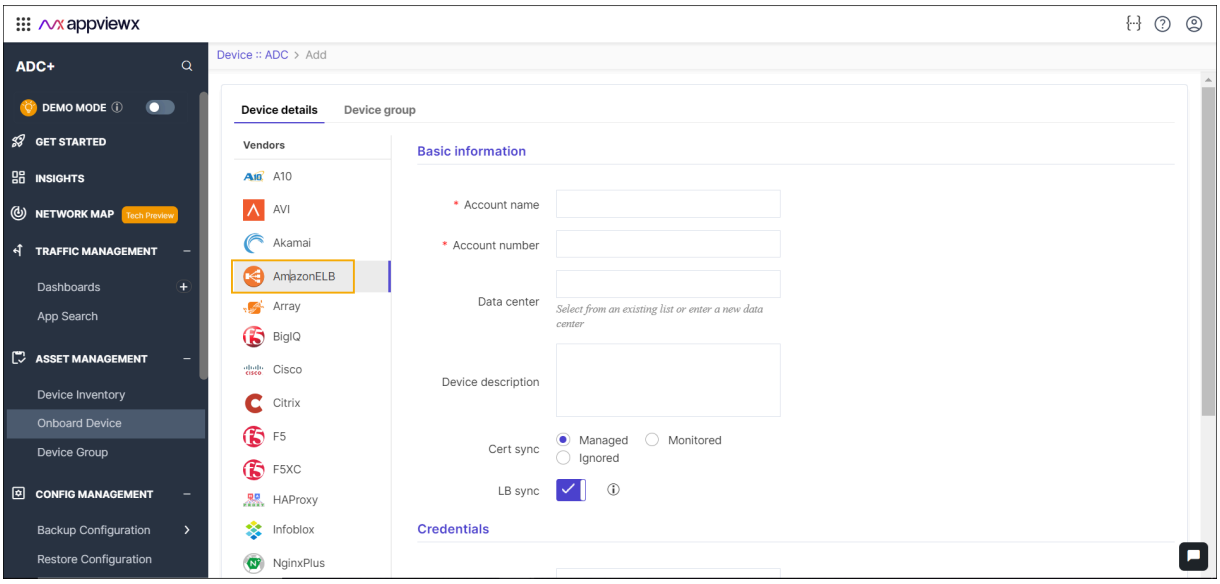
To associate ACF permissions with a role, see Platform User Guide.

Adding Amazon ELB

To add an Amazon ELB:

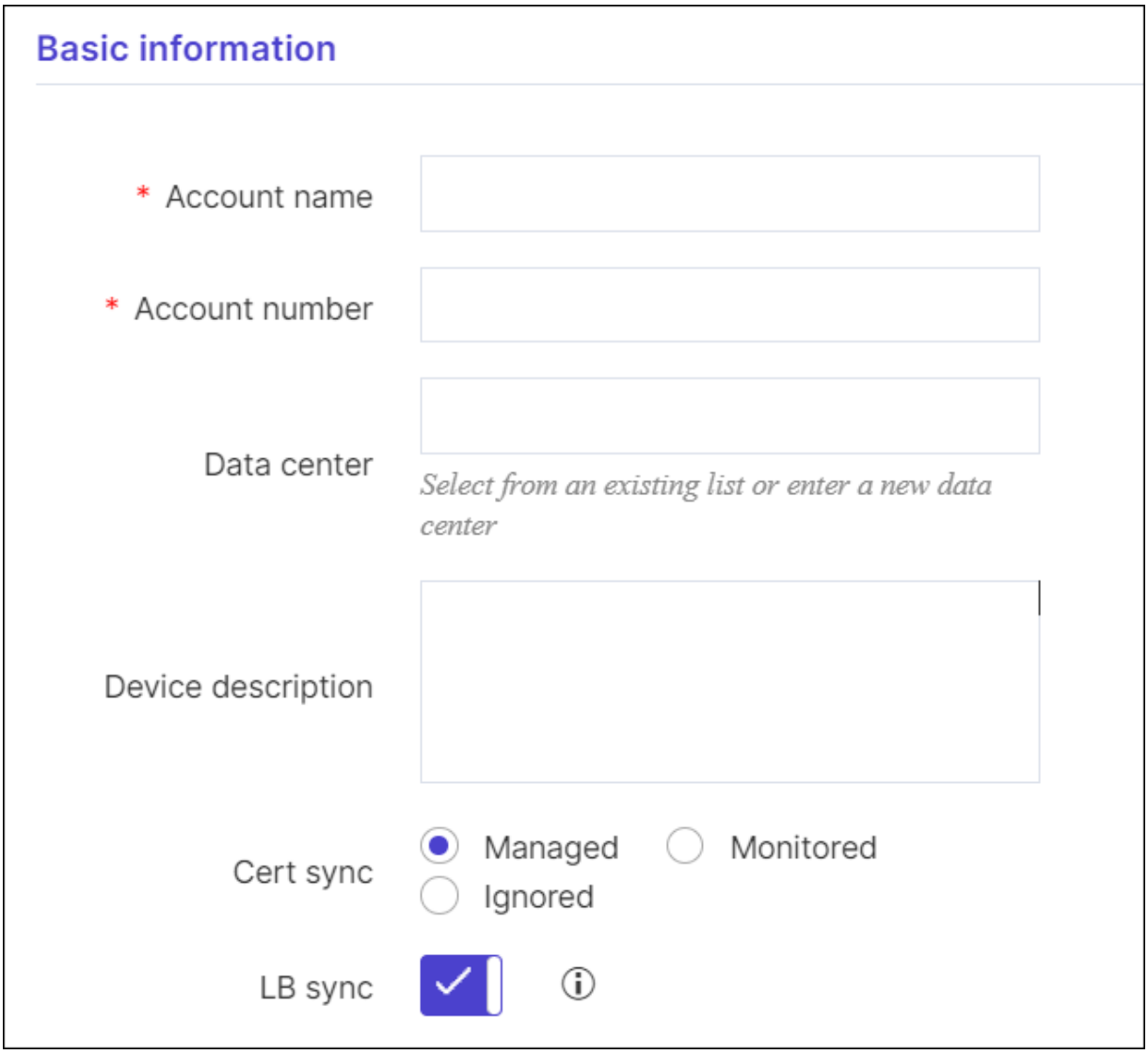
1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select Amazon ELB from the left sidebar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **Basic information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



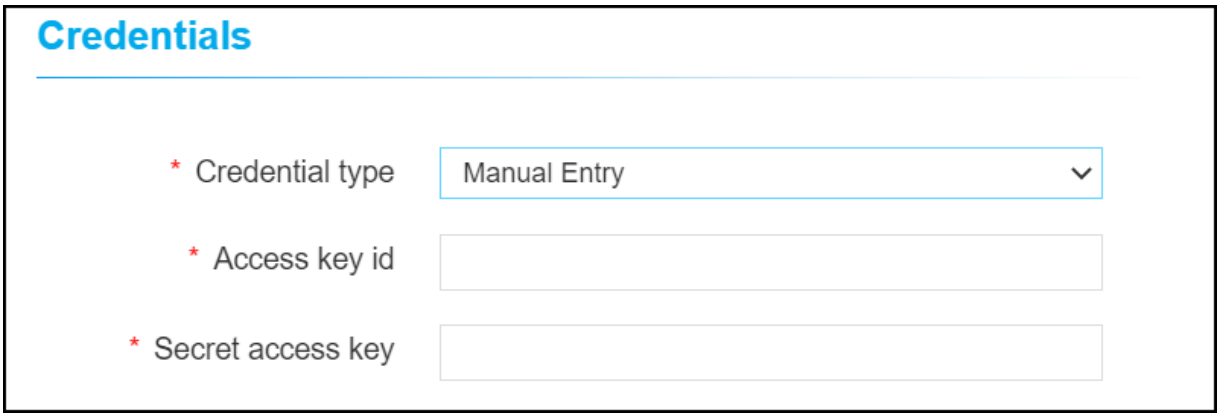
Basic information Section - Field and Description Table

Name	Description
***Account name**	The account name of the AmazonELB device.
***Account number**	Account number of the AmazonEB device. **Note:** Only numbers are allowed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
Device description	Description about the AmazonELB account.
Cert Sync	**Managed**: The certificates of the device can be managed. **Monitored**: The certificates of the device can be discovered and can only be monitored. **Ignored**: The certificate sync can be ignored.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
**: Mandatory fields*	

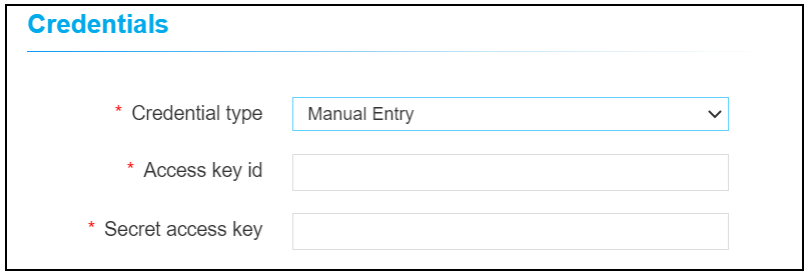
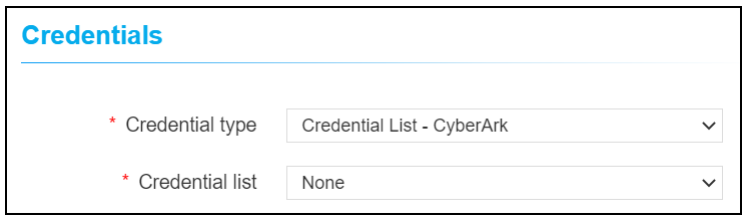
5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.



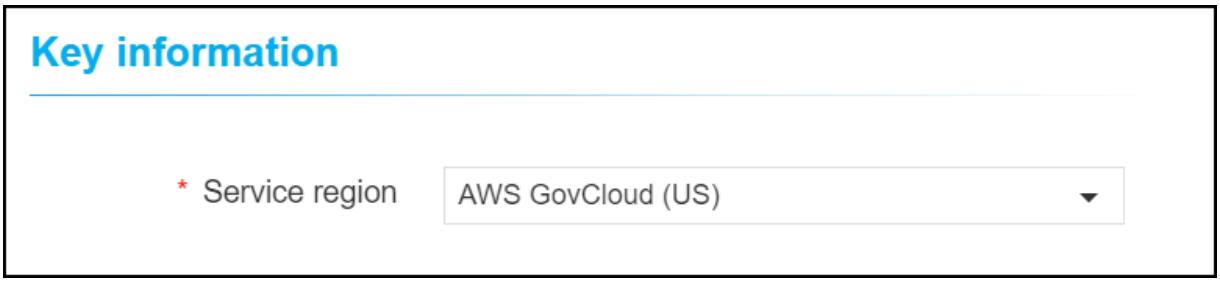
Credentials Section - Field and Description Table

Name	Description
***Credential type**	**Manual entry**: The user should enter the Access key ID and Secret access key.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	 **Credential List**: The user can select the credential details which are already stored in the credential inventory page. 
***Access key id**	Access key id of the AmazonELB device.
***Secret access key**	Secret access key of the AmazonELB device.
**: Mandatory fields*	

6. Enter or select the field information in the **Key information** section.



Key information Section - Field and Description Table

Name	Description
***Service region**	Service region of the AmazonELB device.
**: Mandatory fields*	

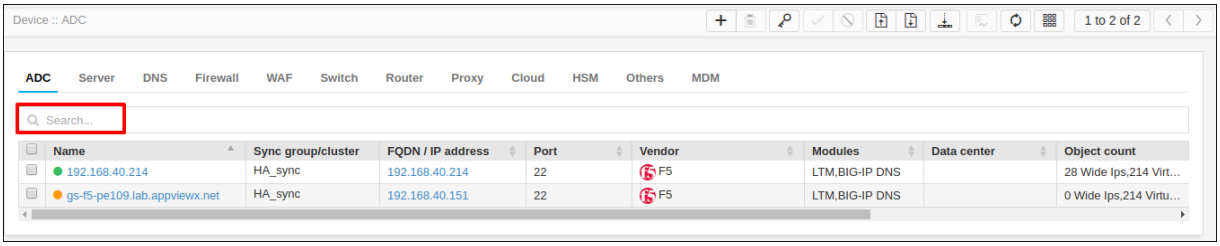
7. Click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Validating the Amazon ELB Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

Array

Before You Begin

Prerequisites

Prerequisites for Managing Array devices on AppViewX are:

- **Version Supported**: Rel.APV.10.4.2.93
- **IP Address/FQDN**: IP address and FQDN
- **User Privilege**: api access
- **Enable Password**: NA
- **License Check**: No
- **Services and Ports for AppViewX Communication**: A port needs to be specified for API access.

> **Note:**
> Any port no. can be used.

- **Internet Access/Proxy if Required**: NA
- **Location from which Certificates are discovered if managed**: NA

Web Browser Requirement

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

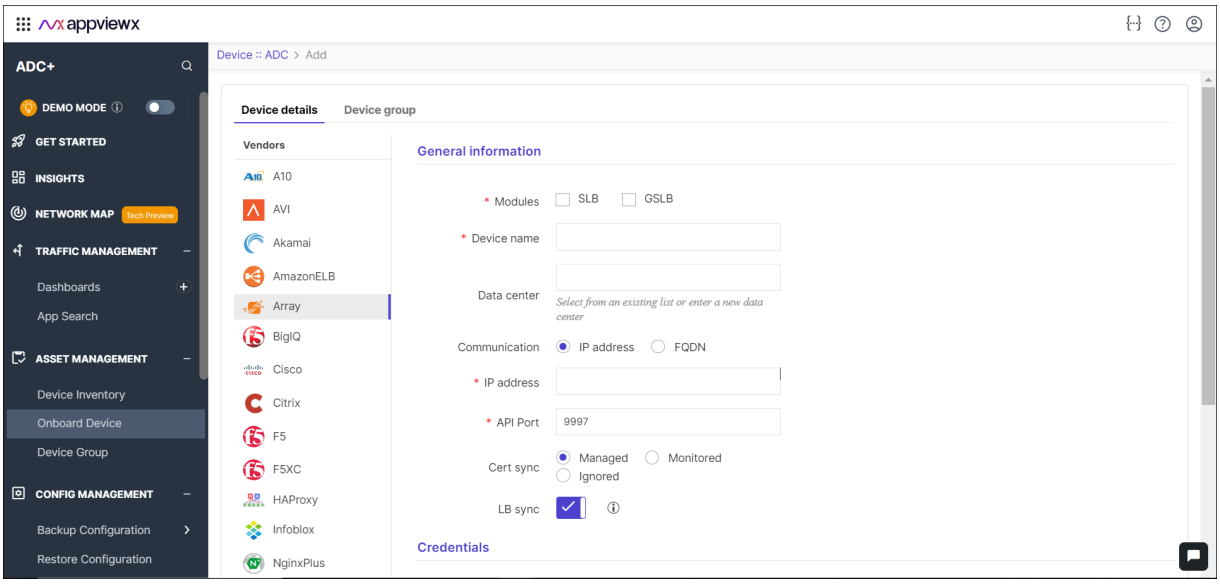
To associate ACF permissions with a role, see Platform User Guide.

Adding Array Device

To add an Array device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **Array** from the left sidebar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **General information** section.

General information Section - Field and Description Table

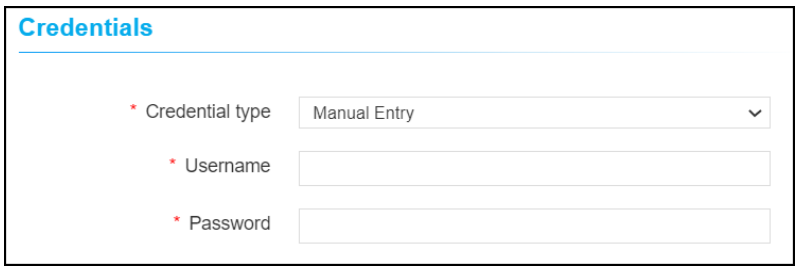
Name	Description
***Modules**	Select modules checkbox. The values are • SLB • GSLB
***Device name**	Unique name of the device to be added. **Note:** Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Enter the Datacenter name where the device is configured. The default value is Absecon. **Note:** Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
Communication	Select the communication method for accessing devices, either through an IP address or FQDN.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

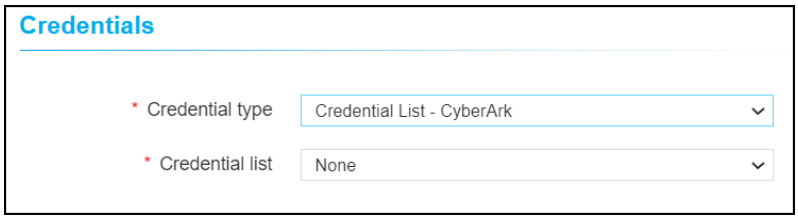
Name	Description
***IP Address**	Enter the IPv4 address of the device. **Note:** The IP address must be in the correct format.
***API Port**	Enter the communication port of the device. **Note:** Only the numbers are allowed.
Cert Sync	Select the certificate sync option: • **Managed**: The certificates of the device can be managed. • **Monitored**: The certificates of the device can be monitored. • **Ignored**: The certificate sync can be ignored.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
**: Mandatory fields*	

5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.

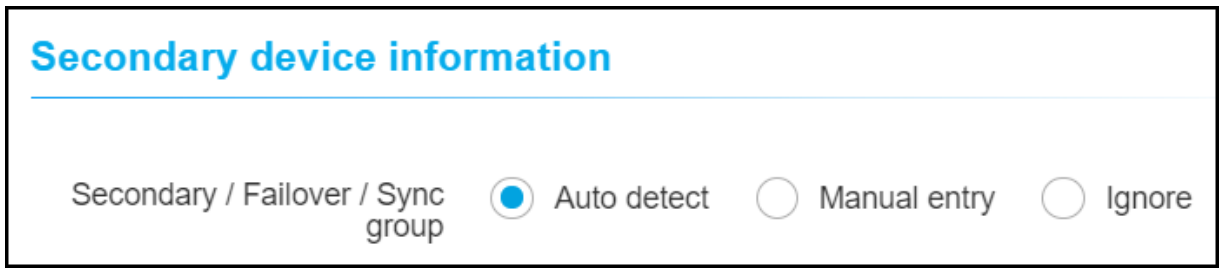
Credentials Section - Field and Description Table

Name	Description
***Credential type**	**Manual entry:** The user should enter the username and password. 

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	Credential List: The user can select the credential details which are already stored in the credential inventory page. 
***Username**	Enter the user name of the device.
***Password**	Enter the password of the device.
**: Mandatory fields*	

6. Enter or select the field information in the **Secondary device information** section.



Secondary device information Section - Field and Description Table

Name	Description
***Secondary / Failover / Sync group**	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually. **Ignore**: The user can use this option to ignore the auto-detection of the peer devices.
**: Mandatory fields*	

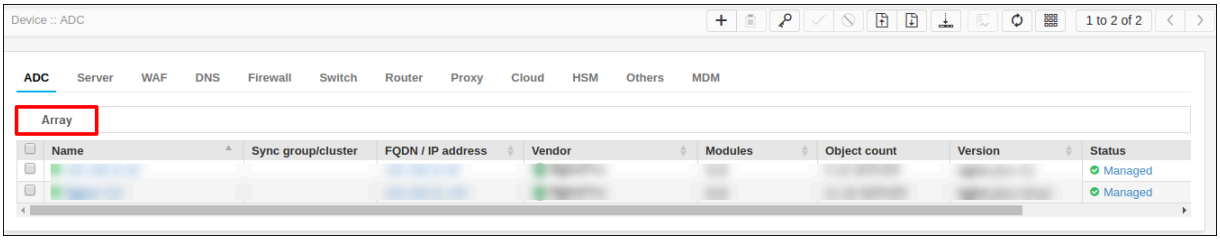
7. Click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Validating the Array Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

BigIQ

Before You Begin

Prerequisites

Prerequisites for Managing BigIQ devices on AppViewX are:

- **Version Supported**: 5.x (5.2/5.3), 7, 8
- **IP Address/FQDN**: IP address
- **User Privilege**: { Role : Admin , tmsh access is required } Username /Password, Credential List AppViewx/CyberArk/Thycotic
- **Enable Password**: Not applicable
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: Port 22 SSH
- **Internet Access/Proxy if Required**: Not applicable
- **Location from which Certificates are discovered if managed**: Certificate Discovery is not supported

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Browsers	Version
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

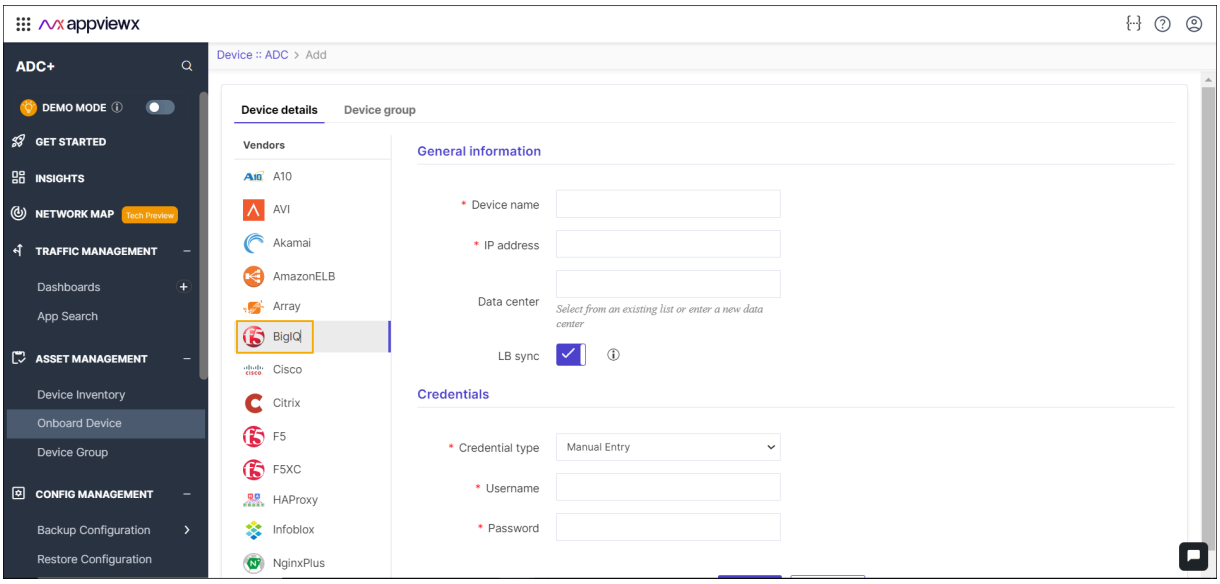
To associate ACF permissions with a role, see Platform User Guide.

Adding BigIQ Device

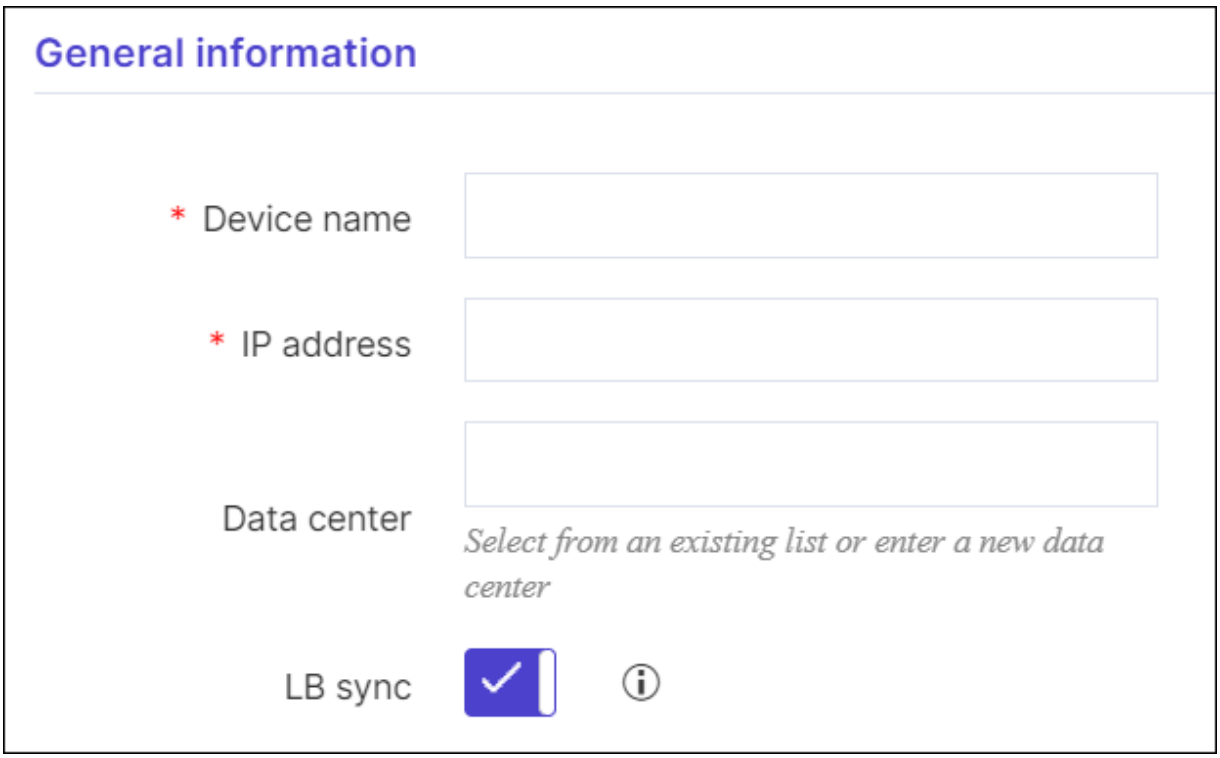
To add a BigIQ device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **BigIQ** from the left sidebar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **General information** section.

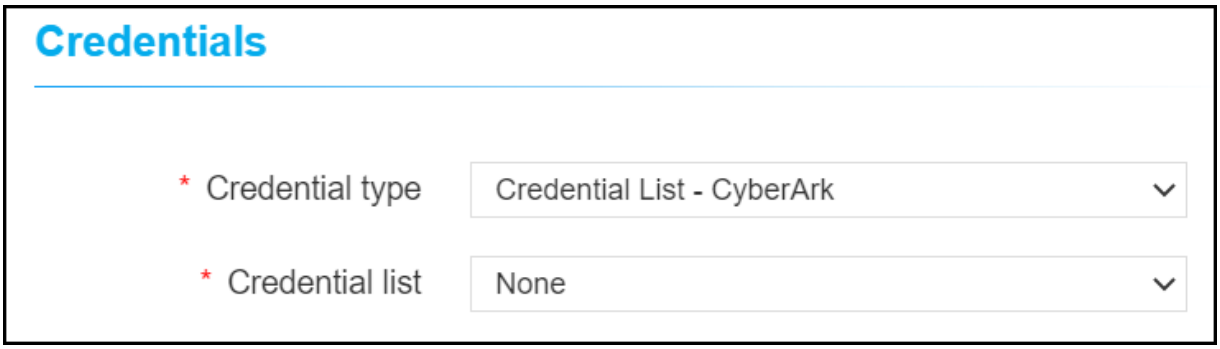


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

General information Section - Field and Description Table

Name	Type	Description	Validation
***Device name**	Text	Unique name of the device to be added.	Device name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
***IP Address**	Text	IP Address of the BigIQ device.	IP address should be in IPv4 format.
Data center	Text	Datacenter name where the device is configured. Default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

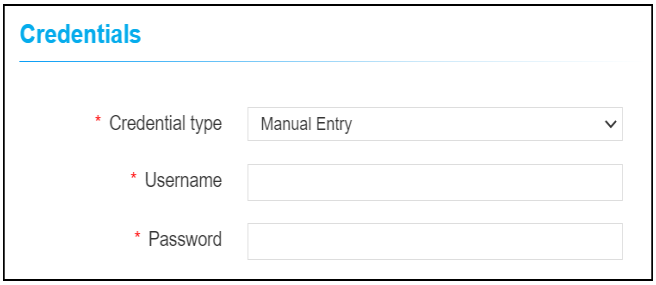
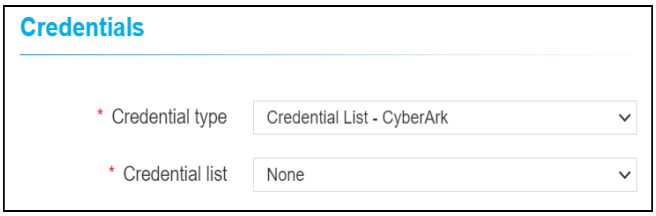
5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.



Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

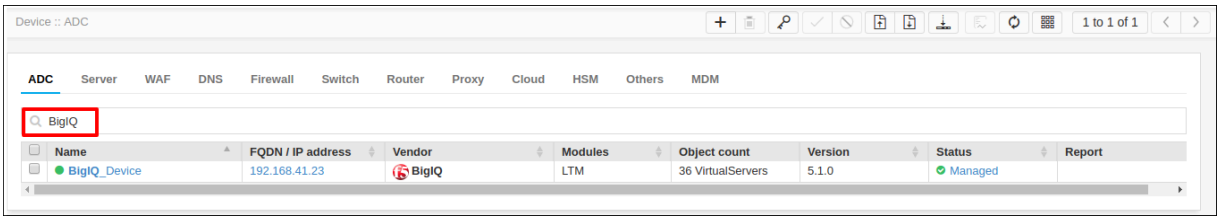
Name	Type	Description	Validation
		 **Credential List**: The user can select the credential details which are already stored in the credential inventory page. 	
***Username**	Text	Username of the BigIQ device.	NA
***Password**	Text	Password of the BigIQ device.	NA
**: Mandatory fields*			

6. Click **Save**.

Validating the BigIQ Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Cisco

Before You Begin

Prerequisites

Prerequisites for Managing Cisco devices on AppViewX are:

- **Version Supported**: 5
- **IP Address/FQDN**: IP address
- **User Privilege**: Username /Password, Credential List AppViewx/CyberArc
- **Enable Password**: Not applicable
- **License Check**: No
- **Services and Ports for AppViewX Communication**: Outbound Access from AppViewX is needed for SSH service and HTTPS service
- **Internet Access/Proxy if Required**: Yes, if the needs be accessed with Public IP address.
- **Location from which Certificates are discovered if managed**: Certificate Discovery is not supported

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

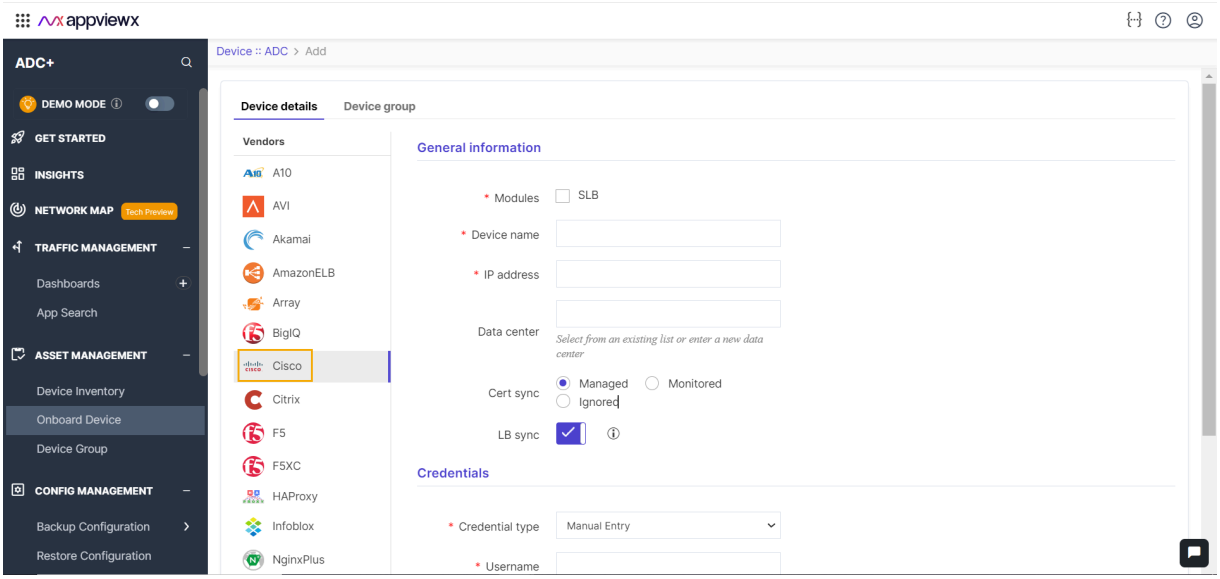
To associate ACF permissions with a role, see Platform User Guide.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Adding Cisco Device

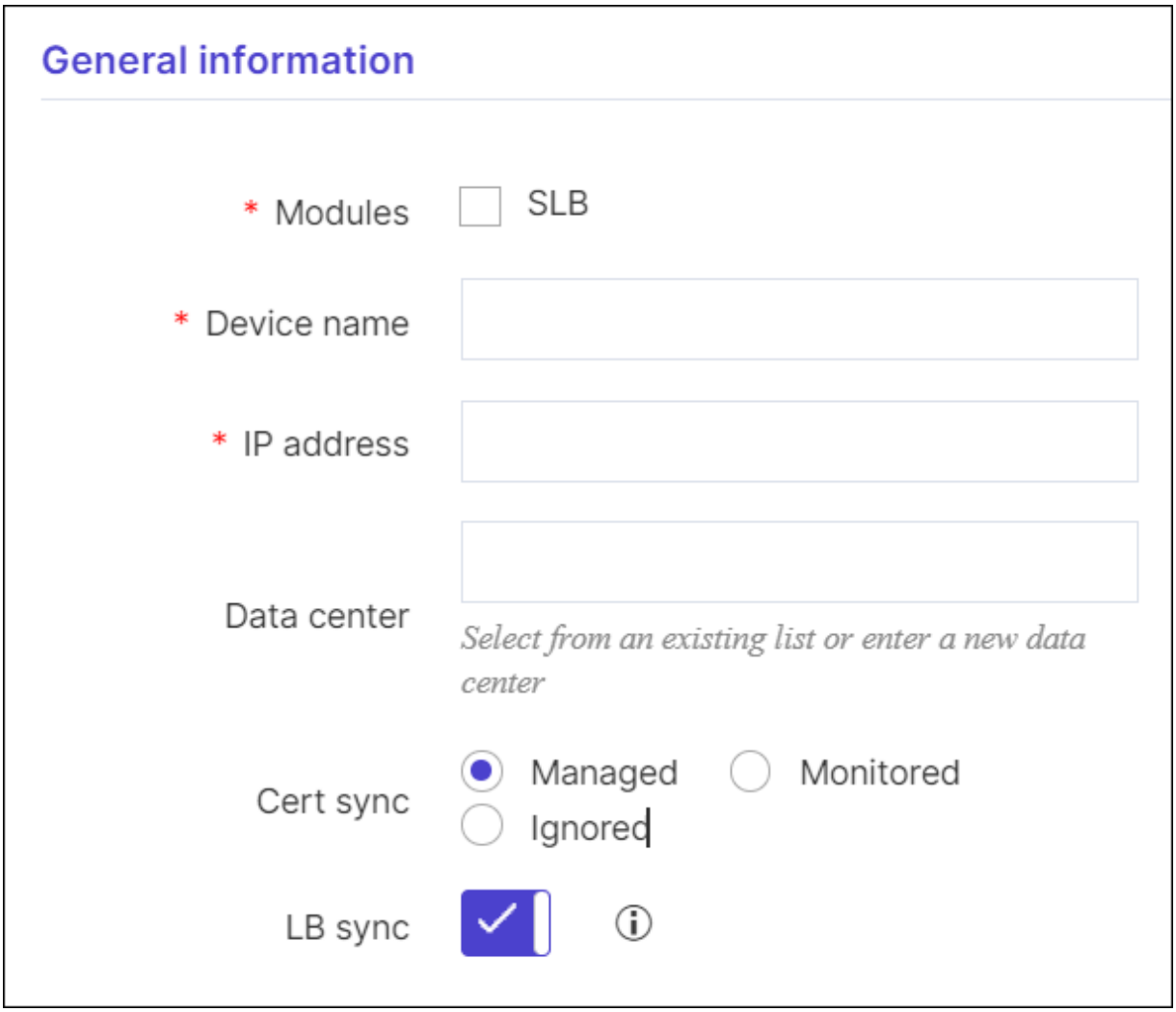
To add a Cisco device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **Cisco** from the left sidebar.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



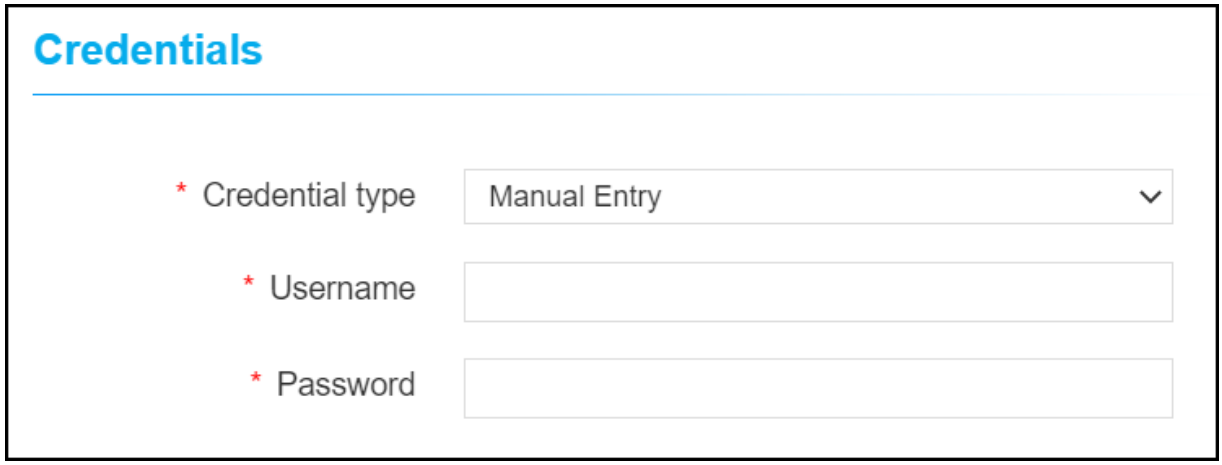
General information Section - Field and Description Table

Name	Type	Description	Validation
***Module**	Check box	SLB Module.	NA
***Device name**	Text	Unique name of the device to be added.	Device name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
***IP Address**	Text	IP Address of the BigIQ device.	IP address should be in IPv4 format.
Data center	Text	Datacenter name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
			characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
***Cert Sync**	Radio button	**Managed**: The certificates of the device can be managed. **Monitored**: The certificates of the device can be discovered and can only be monitored. **Ignored**: The certificate sync can be ignored.	NA
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

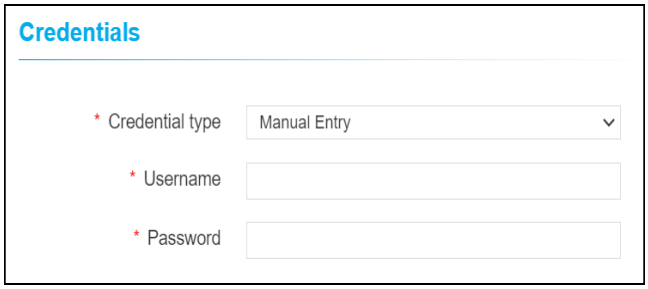
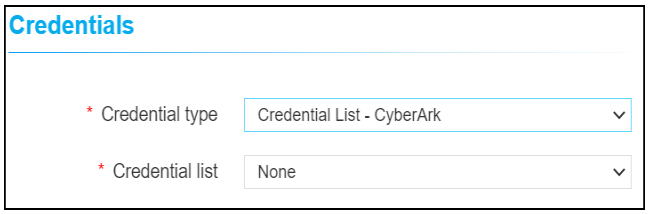
5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.



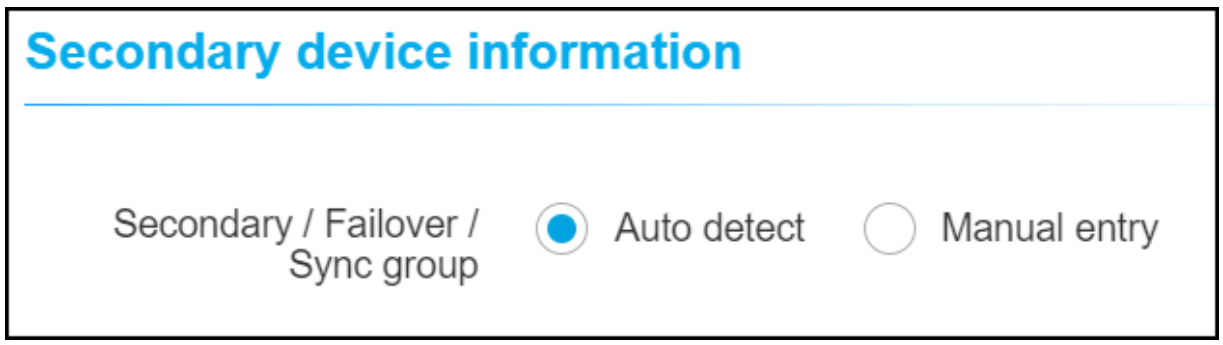
Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
		 **Credential List**: The user can select the credential details which are already stored in the credential inventory page. 	
***Username**	Text	Username of the Cisco device.	NA
***Password**	Text	The password of the Cisco device.	NA
**: Mandatory fields*			

6. Enter or select the field information in the **Secondary device information** section.



Secondary device information Section - Field and Description Table

Name	Type	Description	Validation
Secondary device information	Radio button	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

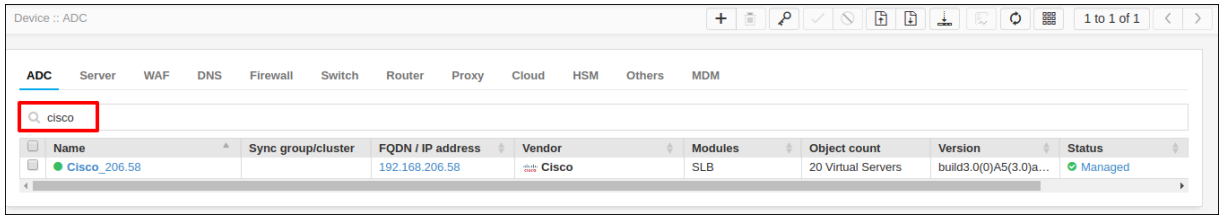
Name	Type	Description	Validation
		Manual entry: The user can use this option to add the peer devices manually.	

7. Click **Save**.

Validating the Cisco Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

Citrix

Before You Begin

Prerequisites

Prerequisites for Managing Citrix devices on AppViewX are:

- **Version Supported**: v9.x, v10.x , V11.x, v12.x, v13.x
- **IP Address/FQDN**: IP address and FQDN
- **User Privilege**: Username /Password, Credential List AppViewx/CyberAr k Role:(Superuser access)/ BeyondTrust
- **Enable Password**: Not applicable
- **License Check**: No
- **Services and Ports for AppViewX Communication**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

  - Outbound Access from AppViewX for SSH and HTTPS service (Default/Custom Port Information) is required
  - Outbound access to port 5514 is required for Citrix Device to forward the Syslog to AppViewX
- **Internet Access/Proxy if Required**: Yes, if the needs be accessed with Public IP address.
- **Location from which Certificates are discovered if managed**: Certificates are parsed only from the /flash/nsconfig/ssl/ directory

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

To associate ACF permissions with a role, see Platform User Guide.

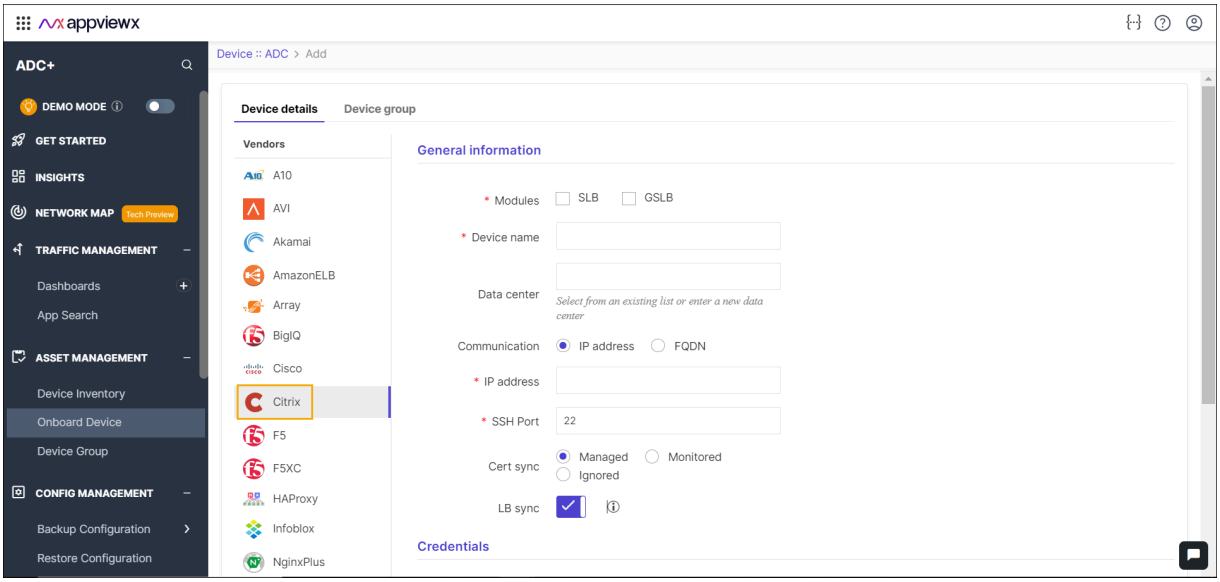
Adding Citrix Device

To add a Citrix device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.

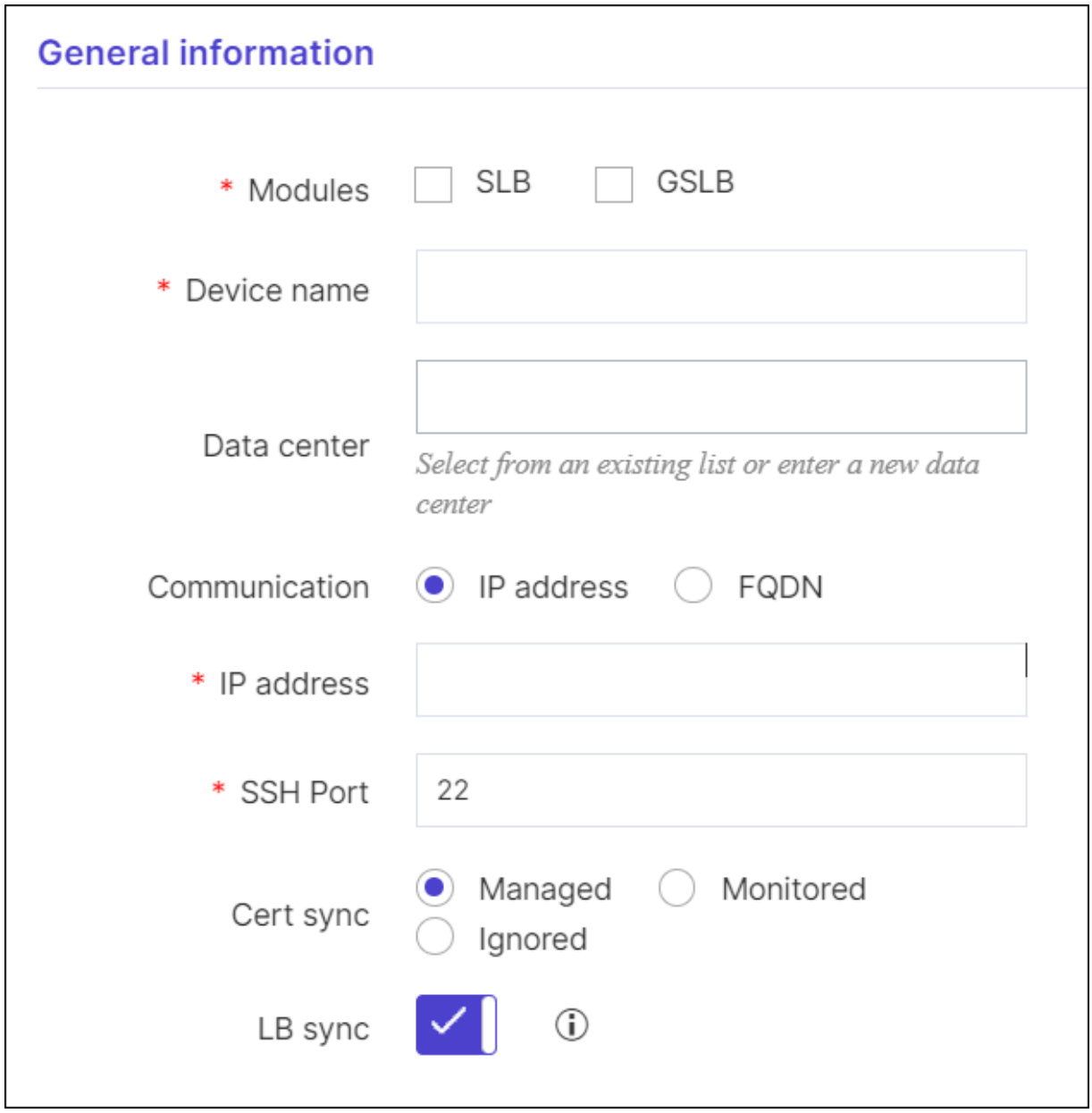
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. In the **Device details** page, select **Citrix** from the left sidebar.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



General information Section - Field and Description Table

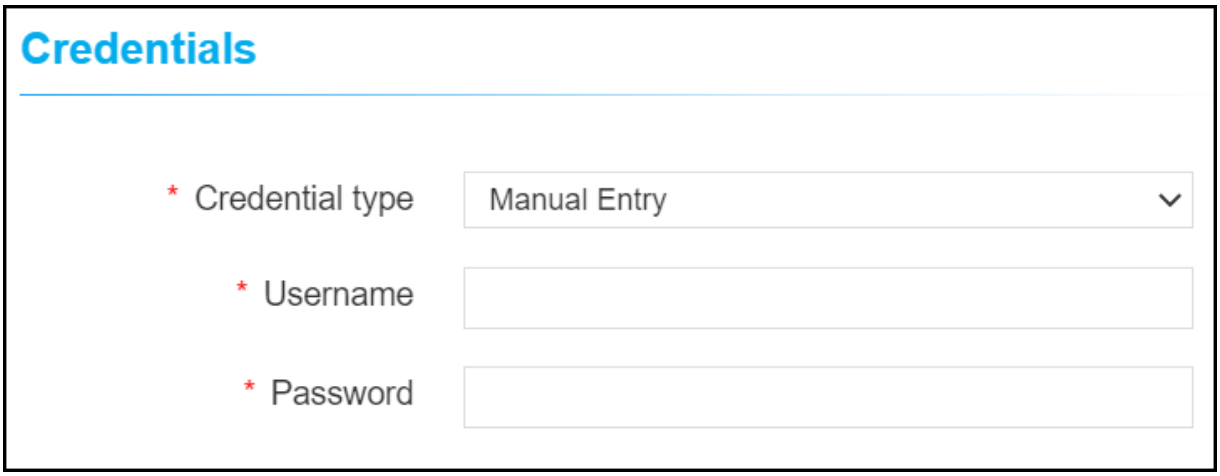
Name	Type	Description	Validation
***Module**	Check box	SLB / GSLB Module.	NA
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

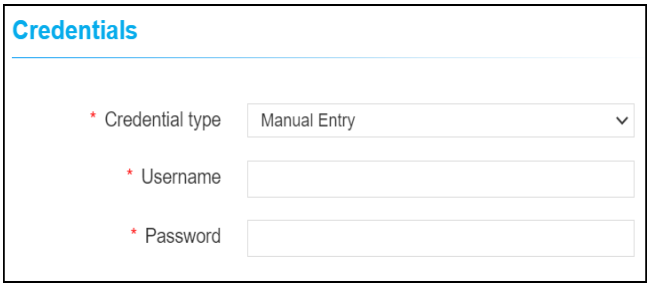
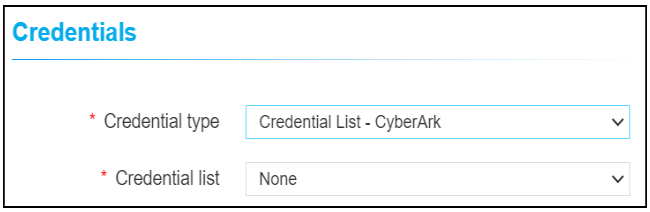
Name	Type	Description	Validation
Data center	Text	Datacenter name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
***Communication**	Radio button	Devices can be accessed using an IP address or FQDN.	NA
***IP Address**	Text	The IPv4 address of the device.	The IP address should be in the right format.
***FQDN**	Text	The FQDN of the device.	the FQDN should be in a valid format.
***SSH Port**	Text	Communication port of the device.	Numbers only.
***Cert Sync**	Radio button	**Managed**: The certificates of the device can be managed. **Monitored**: The certificates of the device can be monitored. **Ignored**: The certificate sync can be ignored.	NA
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

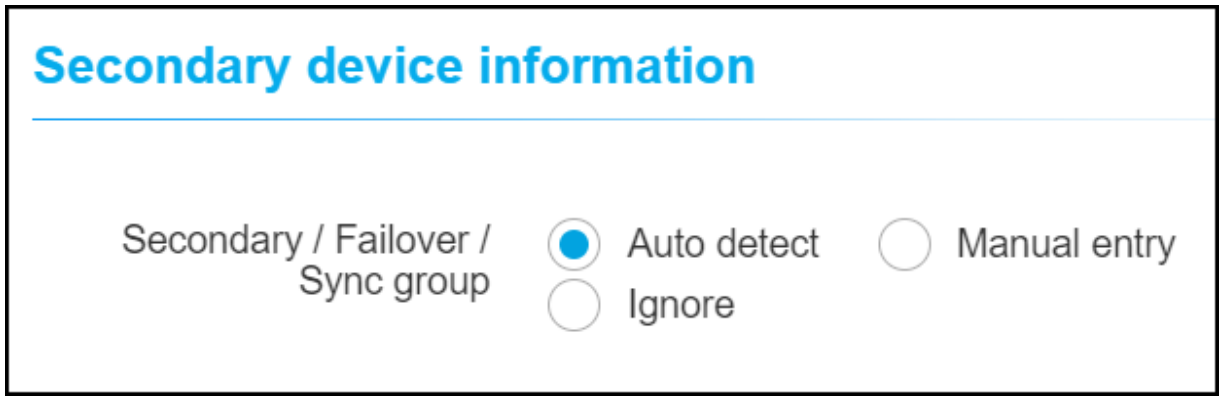


Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password.  **Credential List**: The user can select the credential details which are already stored in the credential inventory page. 	NA
***Username**	Text	The user name of the device.	NA
***Password**	Text	The password of the device.	NA
**: Mandatory fields*			

6. Enter or select the field information in the **Secondary device information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Secondary device information Section - Field and Description Table

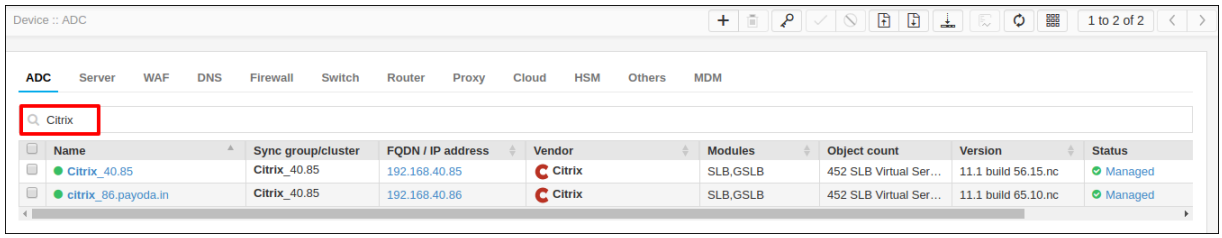
Name	Type	Description	Validation
Secondary device information	Radio button	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually. **Ignore**: The user can use this option to ignore the auto-detection of the peer devices.	NA

7. Click **Save**.

Validating the Citrix Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Adding Cloudflare Device

Before You Begin

Prerequisites

Prerequisites for Managing Cloudflare devices on AppViewX are:

- **Version Supported**: v4
- **IP Address/FQDN**: NA
- **User Privilege**: Api Token

> **Note:**
>
> When creating a Cloudflare API token for managing load balancing, the following minimum permissions should be granted:Permissions:
>
> 1. **Zone-level Permissions**: Load Balancing: Read/Edit
>   This permission allows the token to manage and modify load balancer configurations within specified zones.
> 2. **Account-level Permissions**: Load Balancing Pools and Monitors: Read/Edit
>   This permission enables the token to manage and modify load balancing pools and health monitors within specified accounts.
>
> Resources:
>
> **Account Resource**: Include the specific account(s) that should be accessible using this token.
>
> **Zone Resource**: Include all zones where load balancing management is required.

- **Enable Password**: NA
- **License Check**: No
- **Services and Ports for AppViewX Communication**: Port 443, Outbound Internet access is required to manage the cloudflare
- **Internet Access/Proxy if Required**: Required
- **Location from which Certificates are discovered if managed**: NA

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Browsers	Version
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

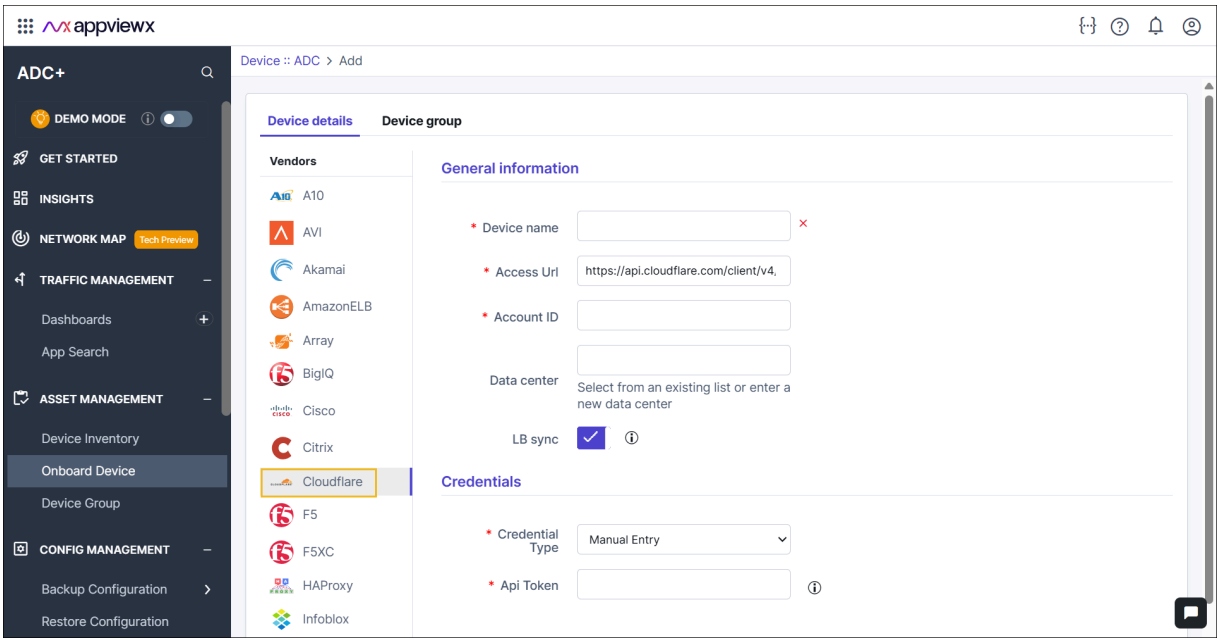
To associate ACF permissions with a role, see Platform User Guide.

Adding Cloudflare Device

To add a Cloudflare device:

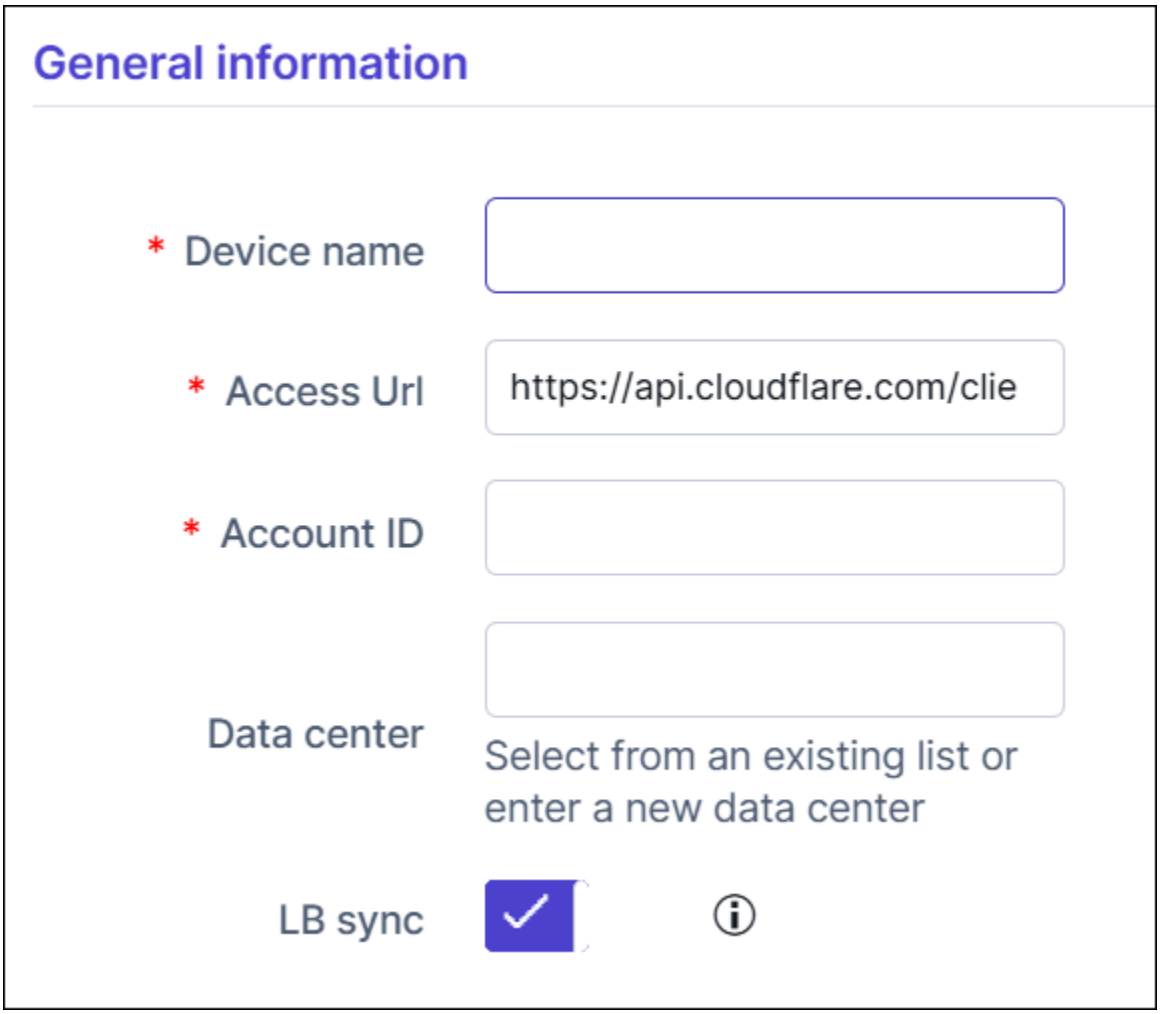
1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **Cloudflare** from the left sidebar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



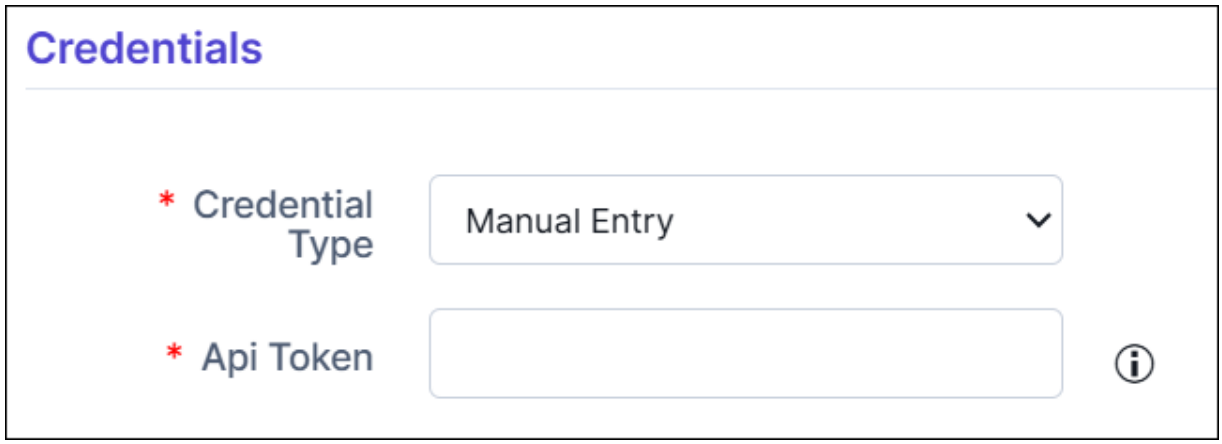
General information Section - Field and Description Table

Name	Description
***Device name**	Unique name of the device to be added. **Note:** Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Access Url	Access URL of Cloudflare.
***Account Id**	Account ID of Cloudflare.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
Data center	Datacenter name where the device is configured. The default value is Absecon. **Note:** Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
**: Mandatory fields*	

5. Enter or select the field information in the **Credentials** section.



Credentials Section - Field and Description Table

Name	Description
***Credential type**	**Manual entry**: The user should enter the API token.
***Api Token**	API token. Required token permissions are: • Zone Level: Load Balancing • Account Level: Load Balancing Pool and Monitors
**: Mandatory fields*	

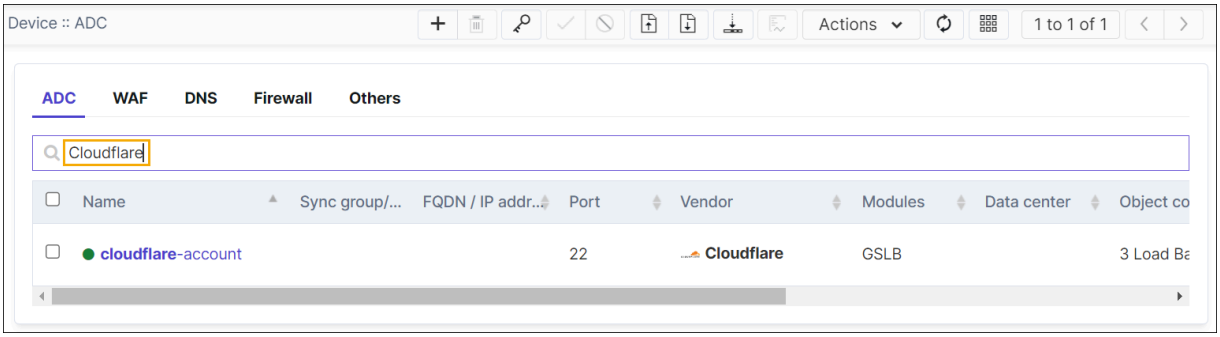
6. Click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Validating the Cloudflare Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

F5

Before You Begin

Prerequisites

Prerequisites for Managing F5 devices on AppViewX are:

- **Version Supported**: v10.x , v11.x , v12.x , v13.x, v14.x, v15.x, v16.x, v17
- **IP Address/FQDN**: IP address or FQDN
- **User Privilege**: { Role : Admin , bash access is required } Username /Password, Credential List AppViewx/CyberArk/BeyondTrust
- **Enable Password**: Not applicable
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: 22 and 443
- **Internet Access/Proxy if Required**: Not applicable
- **Location from which Certificates are discovered if managed**:

  /config/filestore/files_d/<partition>_d/certificate_d/

  /etc/httpd/conf/ssl.crt/

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

/etc/pki/tls/certs/

V10

/config/ssl/ssl.crt/

/etc/httpd/conf/ssl.crt/

/etc/pki/tls/certs

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

To associate ACF permissions with a role, see Platform User Guide.

Adding F5 Device

To add a F5 device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. In the **Device details** page, click the **F5** icon.
4. Enter or select the field information in the **General information** section.

   a. For the device type **Device/Tenant**:

General information Section for "Device/Tenant" Device Type - Field and Description Table

Name	Type	Description	Validation
Device Type	Drop-down	When you select Device Type as Device/ Tenant, you will need to provide the relevant device information in the following fields.	NA
***Module**	Check box	LTM / BIG-IP DNS Module.	NA
VCMP Guest	Check box	To add a device as a vcmp guest, this checkbox should be checked.	NA
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Text	Data center name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
***Communication**	Radio button	Devices can be accessed using an IP address or FQDN.	NA
***IP Address**	Text	The ipv4 address of the device.	The IP address should be a valid IPv4 format.
***FQDN**	Text	The FQDN of the device.	the FQDN should be in a valid format.
***SSH Port**	Text	Communication port of the device.	Numbers only.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
***Cert Sync**	Radio button	**Managed**: The certificates of the device can be managed. **Monitored**: The certificates of the device can be monitored. **Ignored**: The certificate sync can be ignored.	NA
DNSSEC sync	Radio button	By default, the Ignored option is selected. If your ADC acts as an authoritative DNS server, then select the Managed option to manage the DNSSEC Zones and Keys (KSK and ZSK).	
AppViewX Group Sync	Check box	This should be enabled if the user wants to sync the devices within the device group.	NA
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

b. For the device type **Host**:

General information Section for "Host" Device Type - Field and Description Table

Name	Type	Description	Validation
Device Type	Drop-down	When you select Device Type as Host, you will need to provide the relevant device information in the following fields.	NA
Host Type	Radio button	Select any of the following host types:	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
		• vCMP Host - Enables the running of multiple virtual instances of BIG-IP software on a single physical device • rSeried Host - A host type representing the F5 rSeries hardware platform, designed for high-performance application delivery and security. • Velos Partition - A logical partition on the F5 VELOS platform, allowing for the segmentation of resources and management of multiple tenants within a single VELOS chassis.	
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Text	Data center name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
***Communication**	Radio button	Devices can be accessed using an IP address or FQDN.	NA
***IP Address**	Text	The ipv4 address of the device.	The IP address should be a valid IPv4 format.
***FQDN**	Text	The FQDN of the device.	the FQDN should be in a valid format.
***SSH Port**	Text	Communication port of the device.	Numbers only.
**: Mandatory fields*			

c. For the device type **Controller**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

General information Section for "Controller" Device Type - Field and Description Table

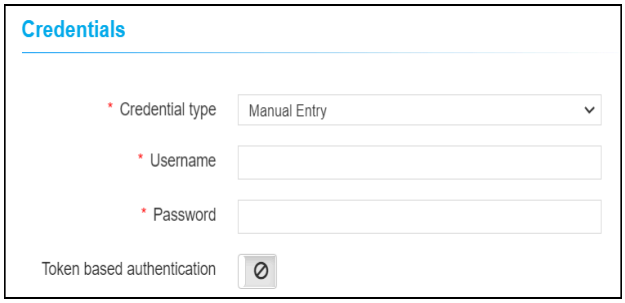
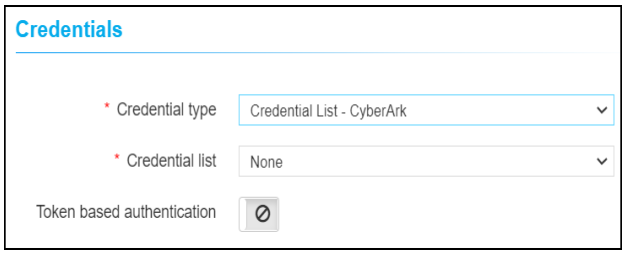
Name	Type	Description	Validation
Device Type	Drop-down	When you select Device Type as Controller, you will need to provide the relevant device information in the following fields.	NA
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Data center	Text	Data center name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
***Communication**	Radio button	Devices can be accessed using an IP address or FQDN.	NA
***IP Address**	Text	The ipv4 address of the device.	The IP address should be a valid IPv4 format.
***FQDN**	Text	The FQDN of the device.	the FQDN should be in a valid format.
***SSH Port**	Text	Communication port of the device.	Numbers only.
**: Mandatory fields*			

5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.

Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
		 **Credential List**: The user can select the credential details which are already stored in the credential inventory page. For more details on secure authentication, refer to Platform User Guide. 	
***Username**	Text	The user name of the device.	NA
***Password**	Text	The password of the device.	NA
Token based authentication	Toggle button	To access the device REST API using token.	NA
**: Mandatory fields*			

6. Enter or select the field information in the **Secondary device information** section.

Secondary device information Section - Field and Description Table

Name	Type	Description	Validation
Secondary device information	Radio button	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

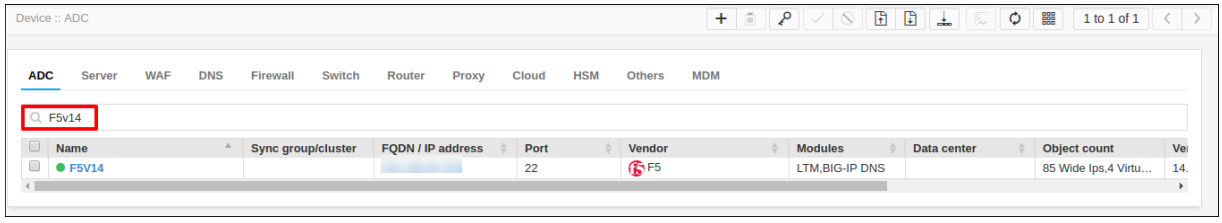
Name	Type	Description	Validation
		Ignore: The user can use this option to ignore the auto-detection of the peer devices.	

7. Click **Save**.

Validating F5 Device Addition

After adding the device, you can validate the device by searching the device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. From the device inventory page, search for the added F5 device name.



What's Next

- Check the Logs for transaction ID and details

F5XC

Before You Begin

Prerequisites

Prerequisites for Managing F5XC devices on AppViewX are:

- **Version Supported**: NA
- **IP Address/FQDN**: FQDN
- **User Privilege**: admin
- **Enable Password**: NA
- **License Check**: No
- **Services and Ports for AppViewX Communication**: NA
- **Internet Access/Proxy if Required**: Internet access required
- **Location from which Certificates are discovered if managed**: NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

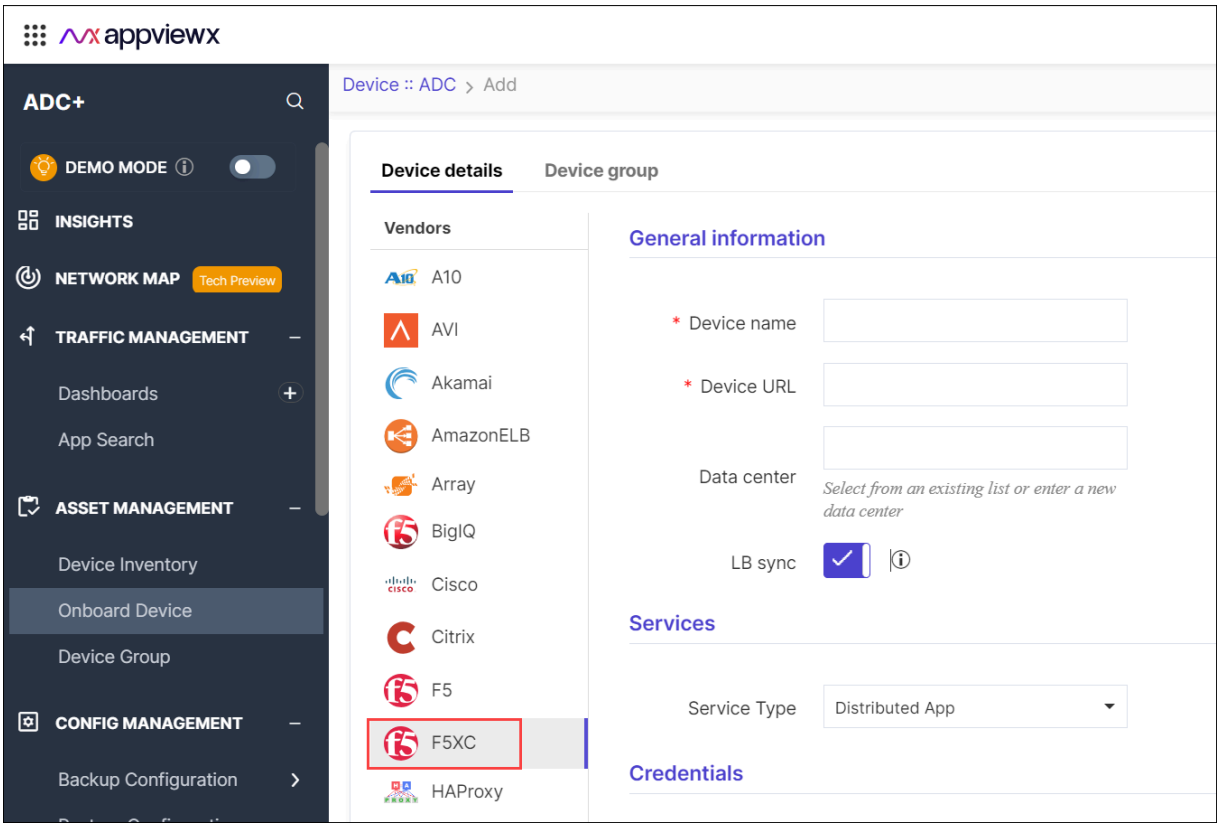
To associate ACF permissions with a role, see Platform User Guide.

Adding F5XC Device

To add a F5XC device:

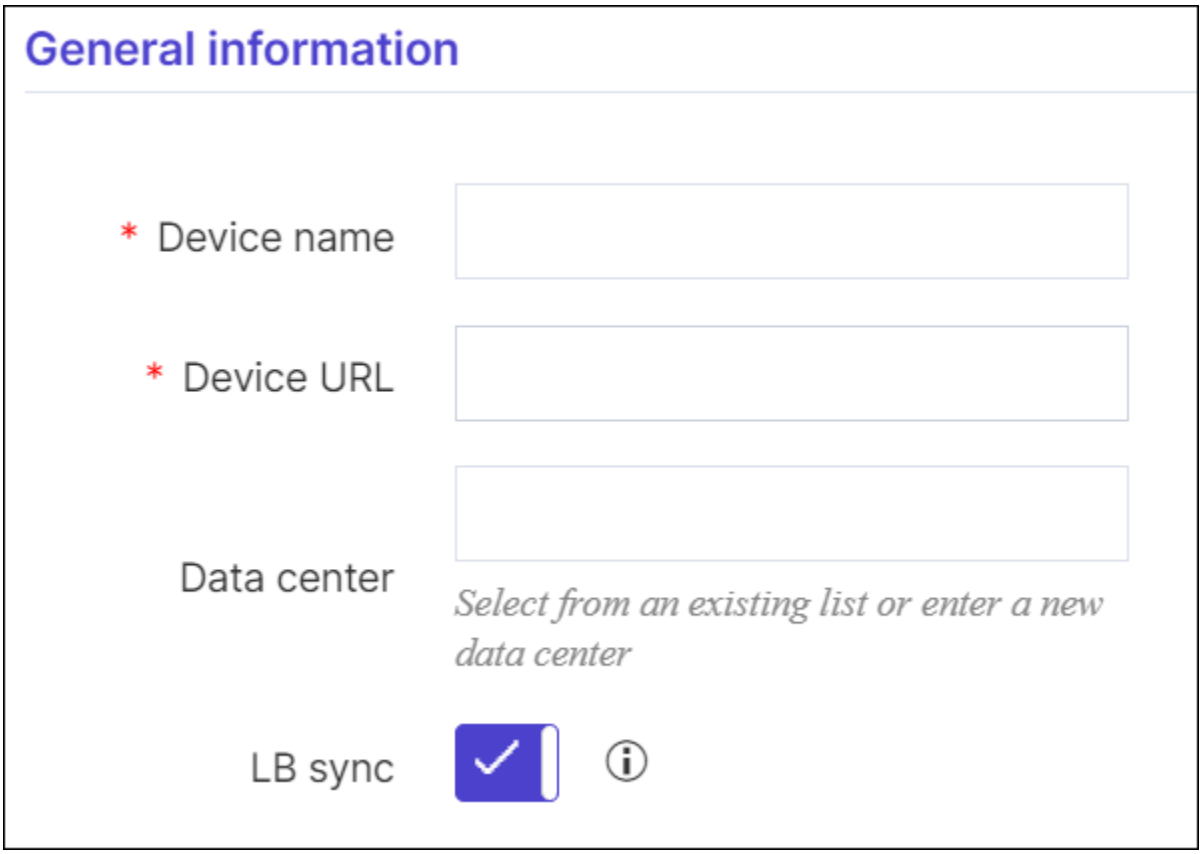
1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, click on the **F5XC** icon.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

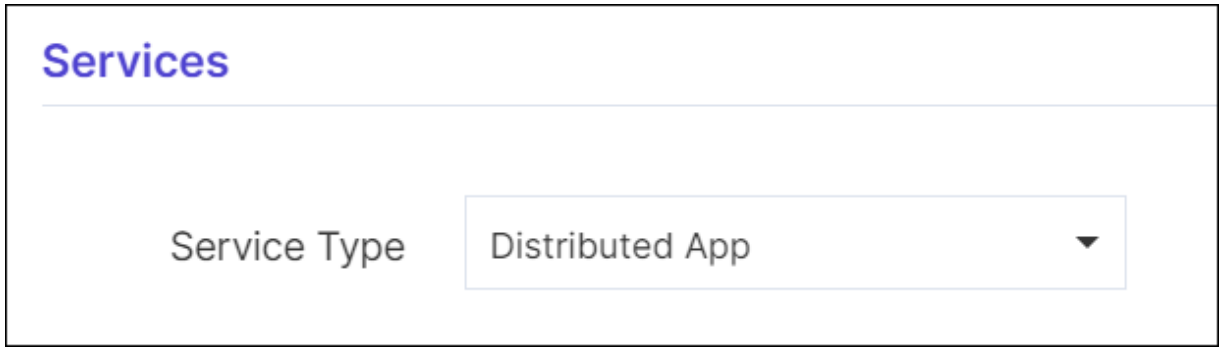


General information Section - Field and Description Table

Name	Description
***Device name**	Enter a unique name of the device to be added. Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
***Device URL**	Enter URL to access the device.
Data center	Select a data center name where the device is configured from the list or enter a new datacenter name. The default value is Absecon. Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
LB Sync	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.
**: Mandatory fields*	

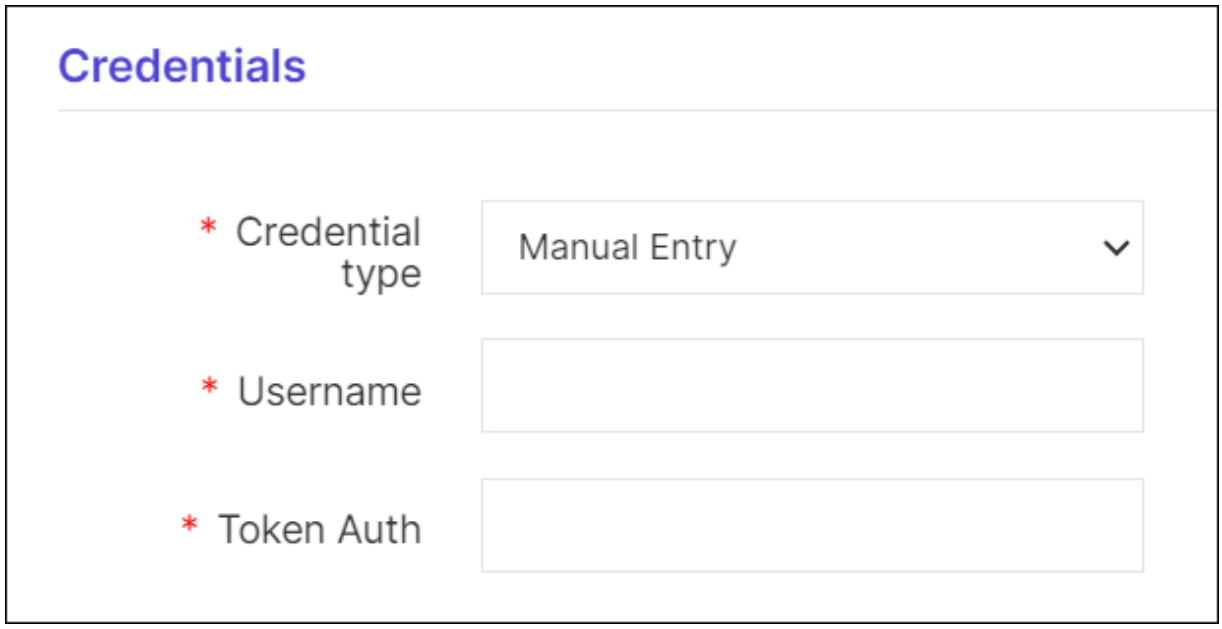
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. In the **Services** section, select service types from the dropdown list.



Note:
More than one service types can be selected.

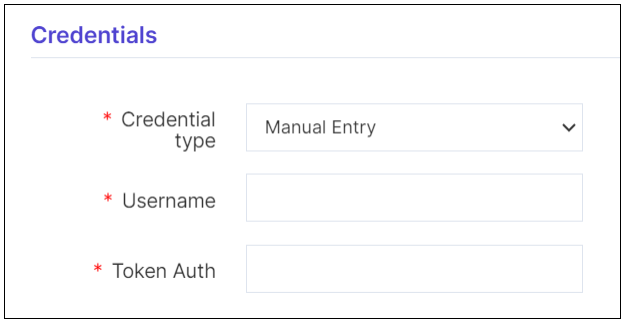
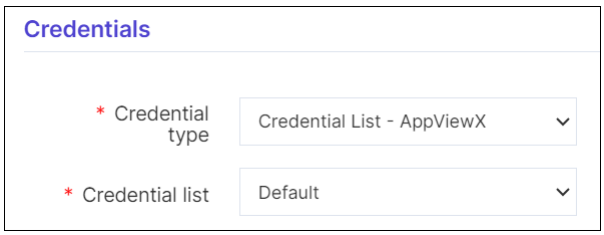
6. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.



Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

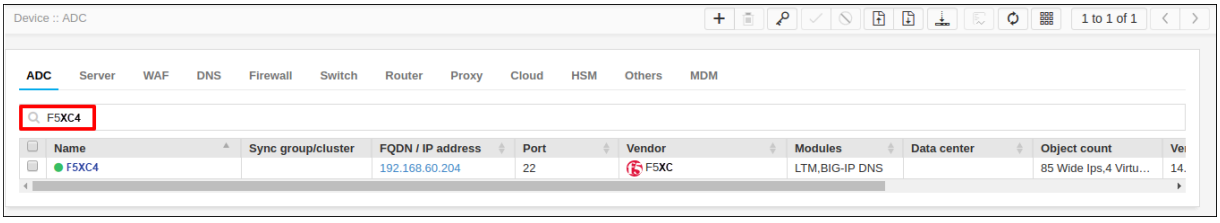
Name	Type	Description	Validation
		 **Credential List**: The user can select the credential details which are already stored in the credential inventory page. For more details on secure authentication, refer to Platform User Guide. 	
***Username**	Text	The user name of the device.	NA
***Token Auth**	Text	Authentication token to access the device REST API using token.	NA
**: Mandatory fields*			

7. Click **Save**.

Validating F5XC Device Addition

After adding the device, you can validate the device by searching the device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. From the device inventory page, search for the added F5XC device name.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Check the Logs for transaction ID and details

HAProxy

Before You Begin

Prerequisites

Prerequisites for Managing HAProxy devices on AppViewX are:

- **Version Supported**: Open source version: 1.4.x, 1.5.x, 1.6.x, 2.x (Community), 2.x (Enterprise)
- **IP Address/FQDN**: IP address
- **User Privilege**: Username /Password, Credential List AppViewx/CyberArc/Thycotic
- **Enable Password**: Not applicable
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: Port 22 SSH
- **Internet Access/Proxy if Required**: Not applicable
- **Location from which Certificates are discovered if managed**:

  /config/filestore/fil es_d/_d /certificate_d/ /etc/httpd/conf/ssl. crt/ /etc/pki/tls/certs/

  V10

  /config/ssl/ssl.crt/ /etc/httpd/conf/ssl. crt/ /etc/pki/tls/certs/

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

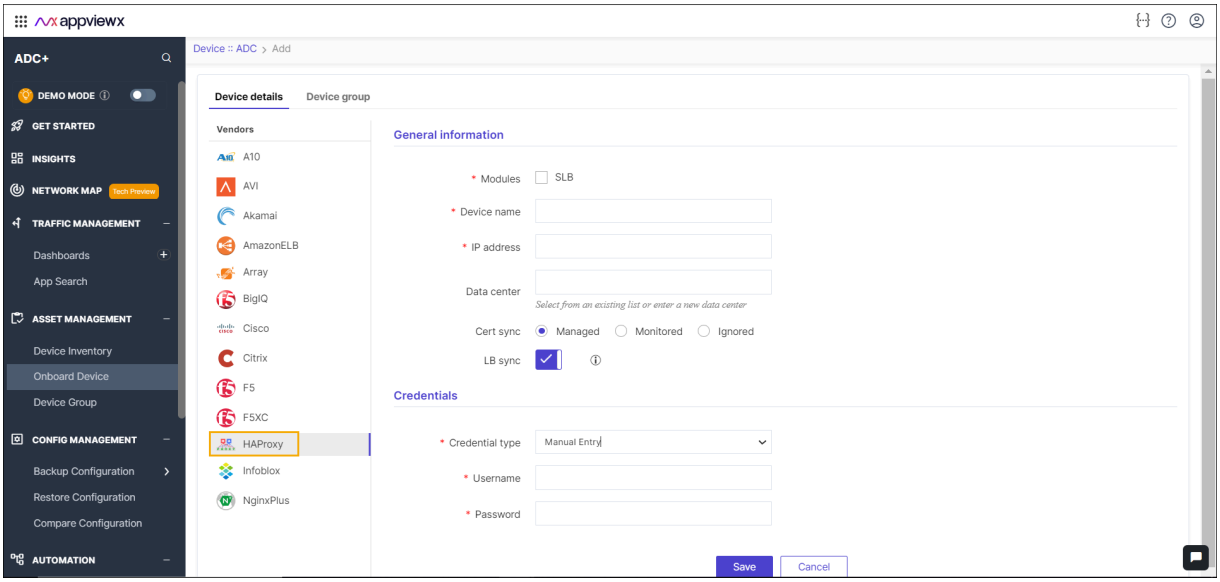
- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

To associate ACF permissions with a role, see Platform User Guide.

Adding HAProxy Device

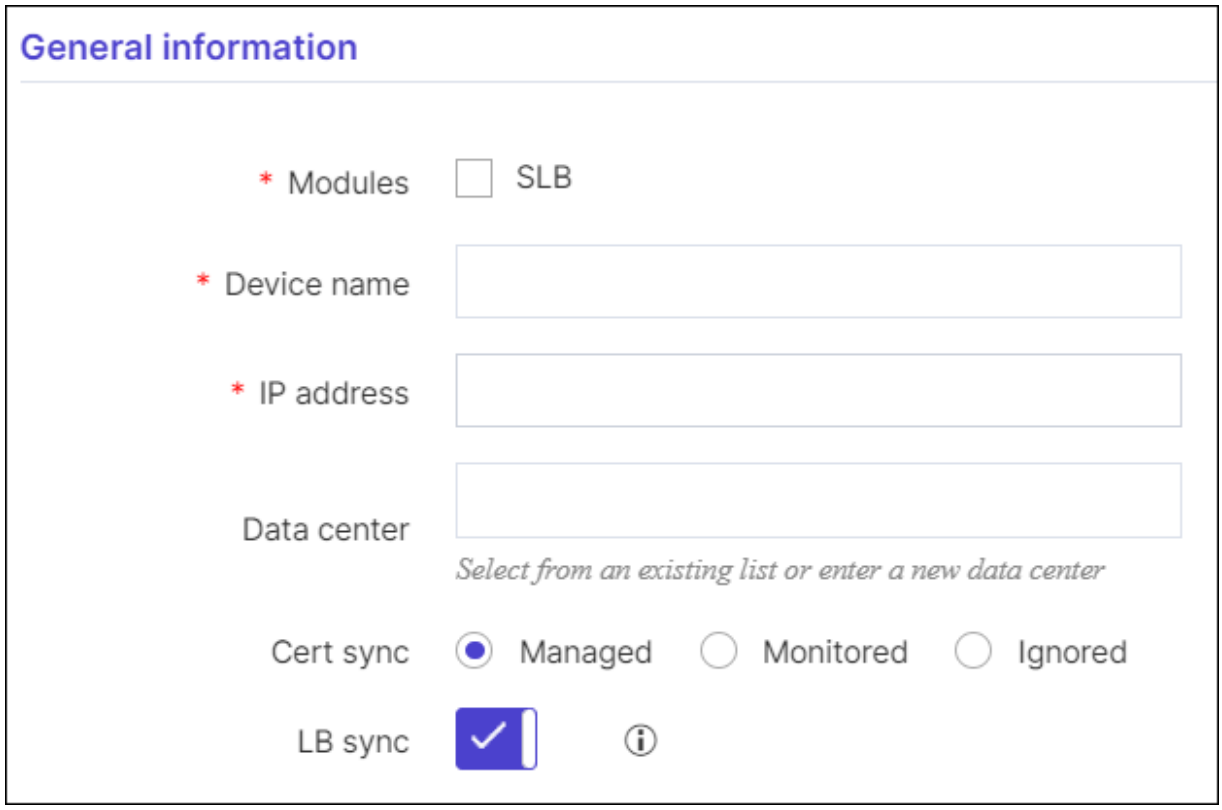
To add a HAProxy device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **HAProxy** from the left sidebar.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



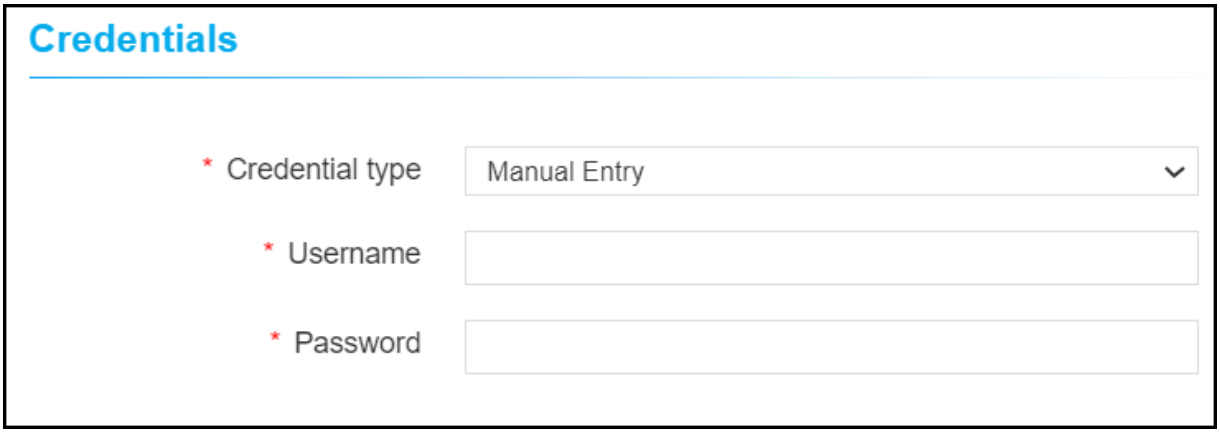
General information Section - Field and Description Table

Name	Type	Description	Validation
***Modules**	Checkbox	Select a module.	NA
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
***IP Address**	Text	The ipv4 address of the device.	The IP address should be in the right format.
Data center	Text	Datacenter name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
***Cert Sync**	Radio button	**Managed:** The certificates of the device can be managed.	NA

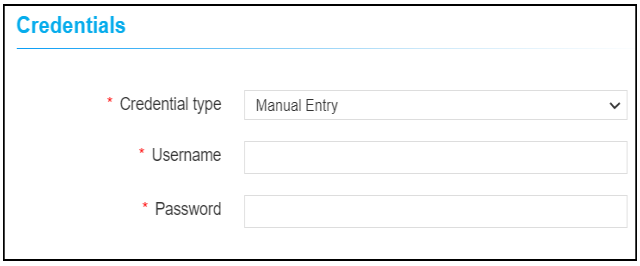
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description	Validation
		Monitored: The certificates of the device can be monitored. **Ignored:** The certificate sync can be ignored.	
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

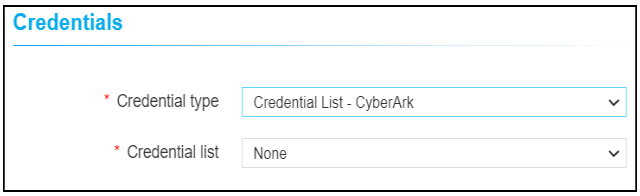
5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.



Credentials Section Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password. 	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

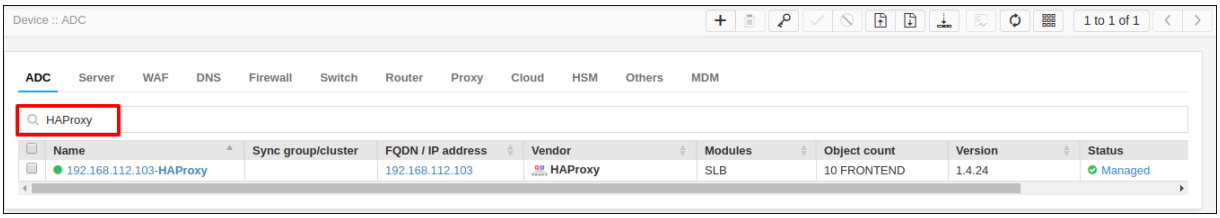
Name	Type	Description	Validation
		Credential List: The user can select the credential details which are already stored in the credential inventory page. 	
***Username**	Text	The user name of the device.	NA
***Password**	Text	The password of the device.	NA
**: Mandatory fields*			

6. Click **Save**.

Validating the HAProxy Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

InfoBlox

Before You Begin

Prerequisites

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Prerequisites for Managing Infoblox devices on AppViewX are:

- **Version Supported**: v8.2 and wapi v2.7.1
- **IP Address/FQDN**: IP address or FQDN
- **User Privilege**: Superuser or access to LBDN objects that needs to be managed from AppViewX
- **Enable Password**: Not applicable
- **License Check**: Not applicable
- **Services and Ports for AppViewX Communication**: 443 or custom port to access API calls
- **Internet Access/Proxy if Required**: Not applicable
- **Location from which Certificates are discovered if managed**: Not applicable

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

To associate ACF permissions with a role, see Platform User Guide.

Adding InfoBlox Device

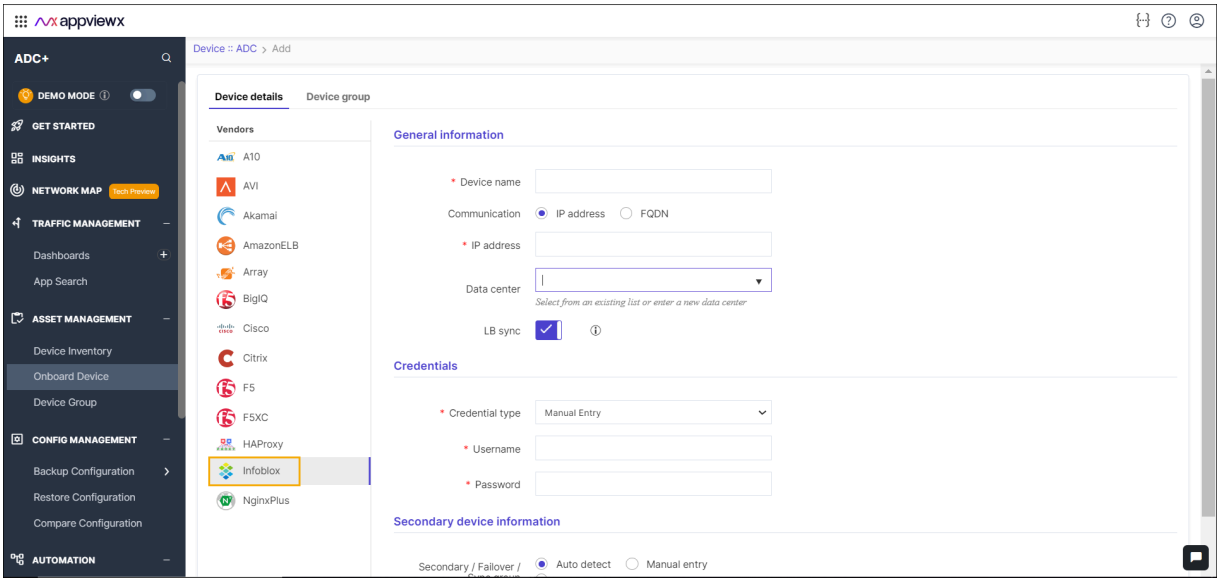
To add an InfoBlox device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:

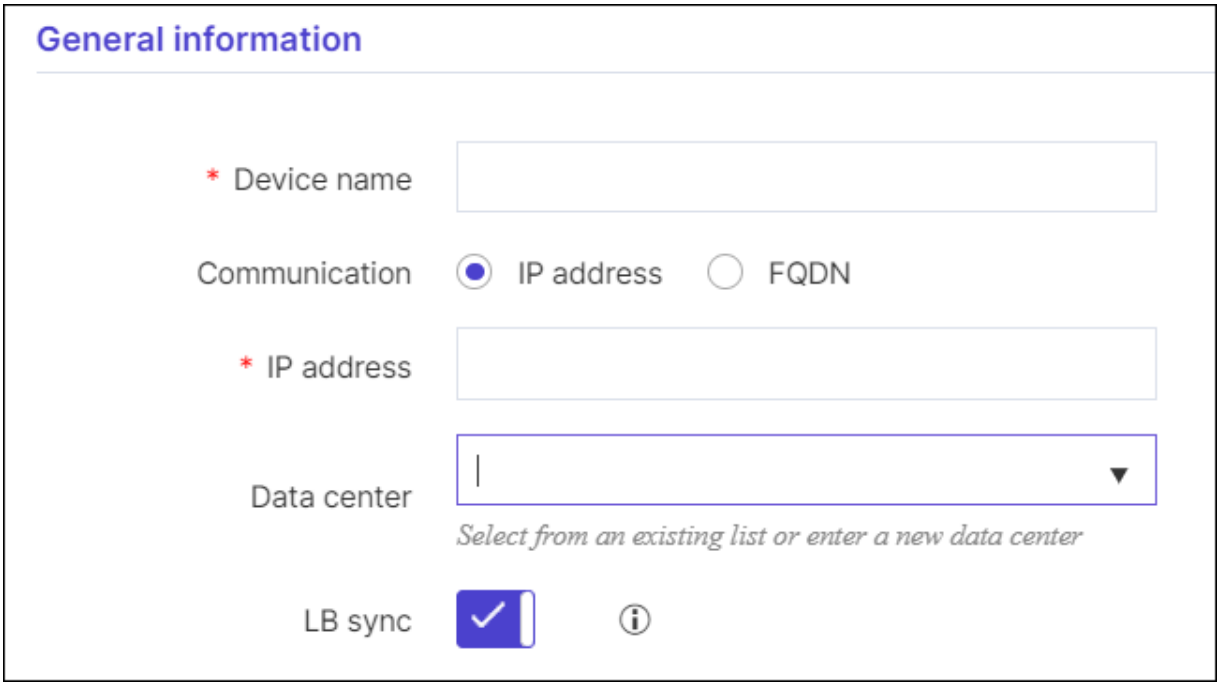
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
- Click **Onboard Device** in the left navigation panel.

3. In the **Device details** page, select **Infoblox** from the left sidebar.



4. Enter or select the field information in the **General information** section.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

General information Section - Field and Description Table

Name	Type	Description	Validation
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
***Communication**	Radio button	Devices can be accessed using an IP address or FQDN.	NA
***IP Address**	Text	The IPv4 address of the device.	The IP address should be in the right format.
***FQDN**	Text	The FQDN of the device.	The FQDN should be in a valid format.
Data center	Text	Datacenter name where the device is configured. The default value is Absecon.	Data center names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
LB Sync	Checkbox	Enable this toggle button to discover and manage the load balancer configurations.	
**: Mandatory fields*			

5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.

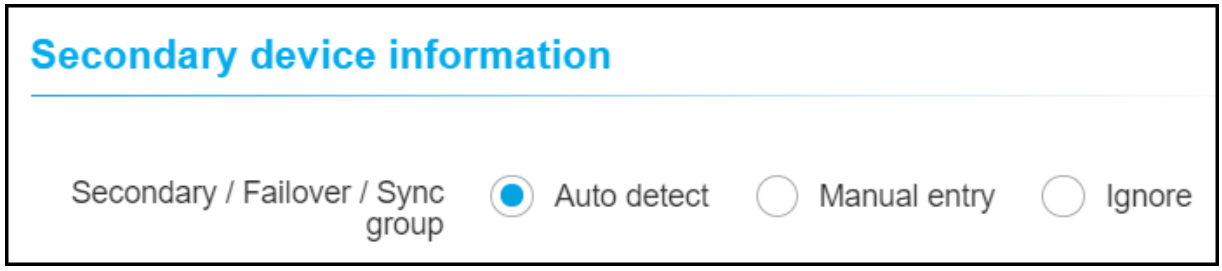


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry**: The user should enter the username and password. **Credential List**: The user can select the credential details which are already stored in the credential inventory page.	NA
***Username**	Text	The user name of the device.	NA
***Password**	Text	The password of the device.	NA
**: Mandatory fields*			

6. Enter or select the field information in the **Secondary device information** section.



Secondary device information Section - Field and Description Table

Name	Type	Description	Validation
Secondary / Failover / Sync group	Radio button	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually. **Ignore**: The user can use this option to ignore the auto-detection of the peer devices.	NA

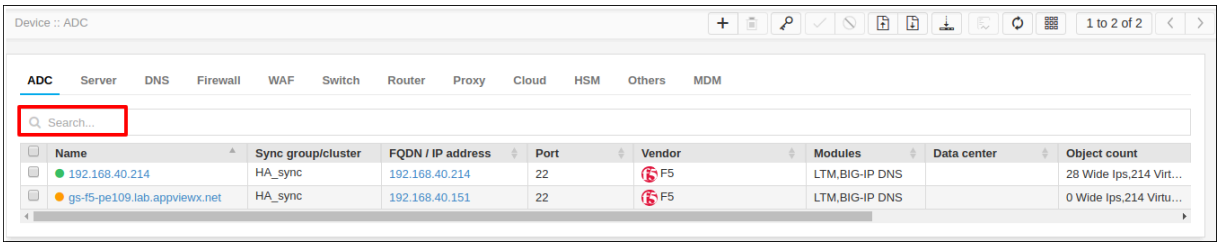
7. Click **Save**.

Validating the Infoblox Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

NginxPlus

Before You Begin

Prerequisites

Prerequisites for Managing Nginx devices on AppViewX are:

- **Version Supported**: R11, 18, 19, 20, 21, 22, 23, 24, 25,26, 27, 28, 29, 30, 31
- **IP Address/FQDN**: IP address
- **User Privilege**: Access to Nginx installed configuration location and access to check syntax of niginx.conf and restart nginx service to apply the changes
- **Enable Password**: Not applicable
- **License Check**: Yes
- **Services and Ports for AppViewX Communication**: 22 and any custom port for stats
- **Internet Access/Proxy if Required**: Not applicable
- **Location from which Certificates are discovered if managed**:

  /etc/nginx/conf.d/ or any custom location

Web Browser Requirement

Browsers	Version
Internet Explorer	v11.0.9600.18817 or later
Firefox	v74.0.1 (64-bit) or later

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Browsers	Version
Google Chrome	117.0.5938.134 (Official Build) (64-bit) or later

ACF Permission

Ensure the following ACF permission is available to add and manage devices:

- ADC > Inventory > Add / Modify
- ADC > Device group > Add / Modify

You can check this under any of the following menus:

- **Old menu**: menu > Accounts > Role
- **New menu**: menu > Platform > IDENTITY > Role

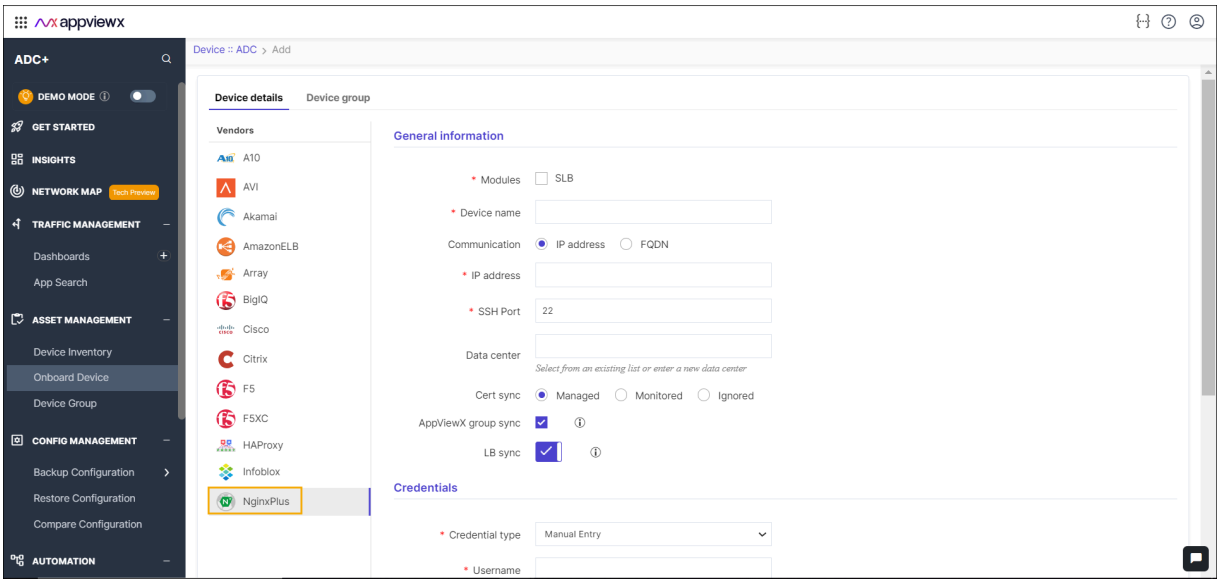
To associate ACF permissions with a role, see Platform User Guide.

Adding NginxPlus Device

To add an NginxPlus device:

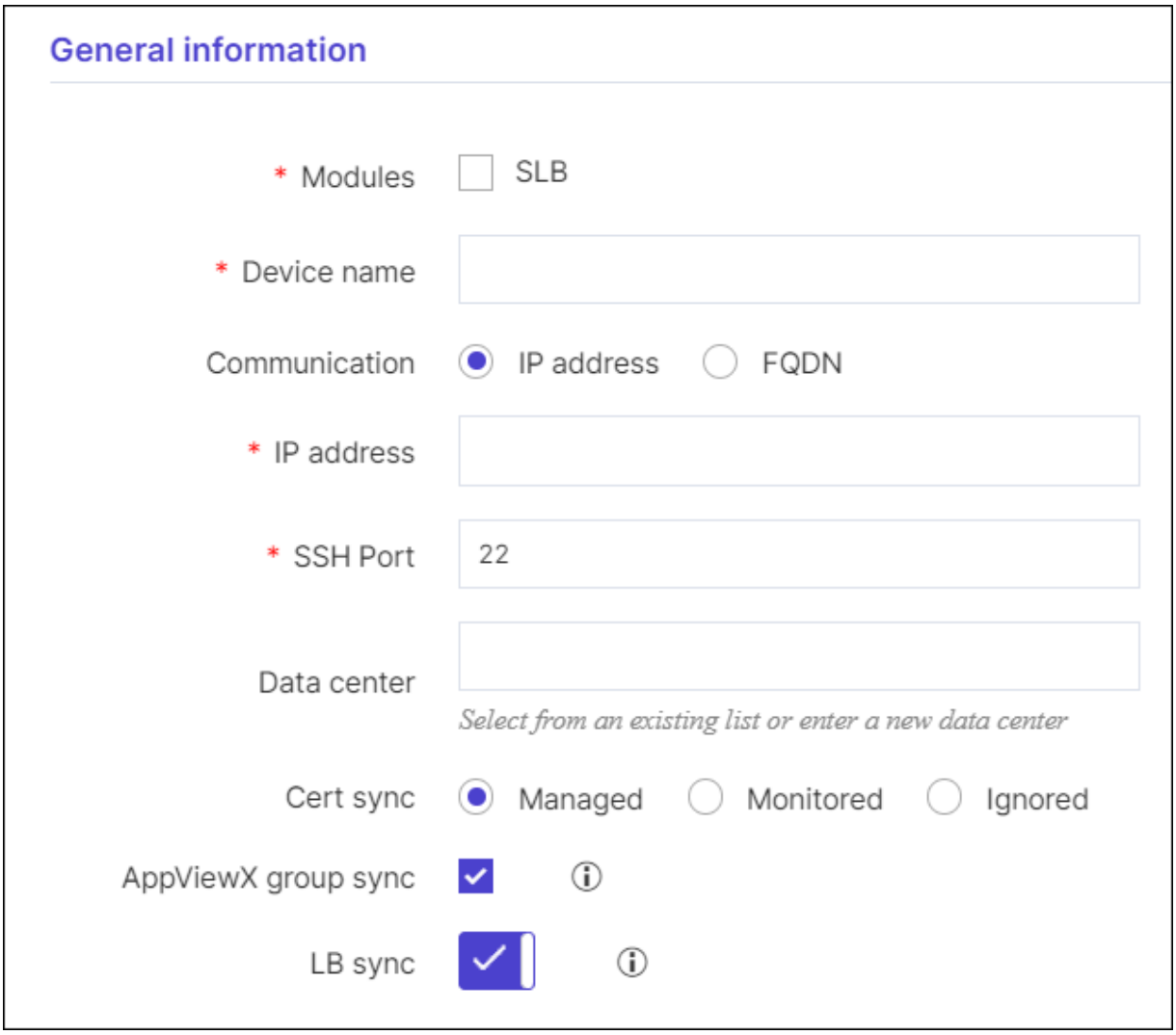
1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT**.
2. Perform one of the followings:
   - Click **Device Inventory**, and then select **Add (+)** icon to navigate to the Device details page.
   - Click **Onboard Device** in the left navigation panel.
3. In the **Device details** page, select **NginxPlus** from the left sidebar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Enter or select the field information in the **General information** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



General information Section - Field and Description Table

Name	Type	Description	Validation
***Modules**	Check box	SLB	NA
***Device name**	Text	Unique name of the device to be added.	Device names can only contain alphanumeric characters, '-' , '_' , '.' , '*' , '|' , '!' and spaces.
Communication	Radio button	Devices can be accessed using an IP address or FQDN.	No

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

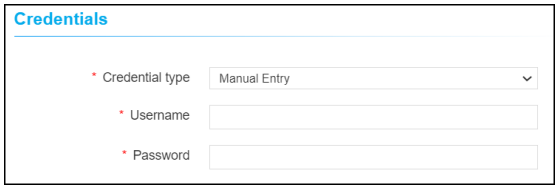
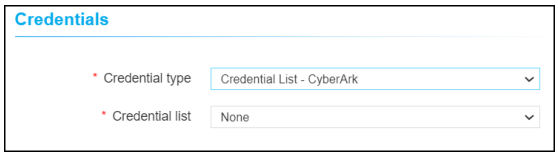
Name	Type	Description	Validation
***IP Address**	Text	The IPv4 address of the device.	The IP address should be in the right format.
***SSH Port**	Text	Communication port of the device.	Numbers only.
Data center	Text	Datacenter name where the device is configured. The default value is Absecon.	Data center name can only contain alphanumeric characters, '-' , '_' , '.' , '*' , ':' , '|' and spaces.
Cert Sync	Radio button	**Managed**: The certificates of the device can be managed. **Monitored**: The certificates of the device can be monitored. **Ignored**: The certificate sync can be ignored.	
LB Sync	Checkbox	By default, this option is enabled to discover and manage the load balancer configurations. If not required, you can disable this option.	
**: Mandatory fields*			

5. Enter or select the field information in the **Credentials** section. You can select **Manual Entry** or **Credentials List**.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Credentials Section - Field and Description Table

Name	Type	Description	Validation
***Credential type**	Dropdown	**Manual entry:** The user should enter the username and password.  **Credential List:** The user can select the credential details which are already stored in the credential inventory page. 	NA
***Username**	Text	The user name of the device.	NA
***Password**	Text	The password of the device.	NA
**: Mandatory fields*			

6. Enter or select the field information in the **Authenticaion details** section.

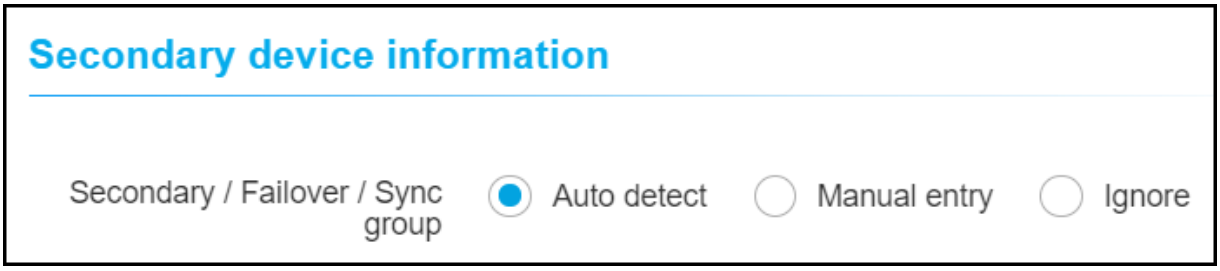


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Authenticaion details Section - - Field and Description Table

Name	Type	Description	Validation
Sudo Auth	Checkbox	Select the **Sudo Auth** checkbox in the to enable the Sudo authentication for a non-root user credentials.	
API Auth	Dropdown	This option allows the basic or token based authentication for API enabled devices. The options are: • **Basic Auth** - select this option to configure HTTP authentication with password. You can choose device credentials or manual entry. • **Token Auth** - select this option to enter the JSON Web tokens for token based authentication.	

7. Enter or select the field information in the **Secondary device information** section.



Secondary device information Section - Field and Description Table

Name	Type	Mandatory	Description	Validation
Secondary / Failover / Sync group	Radio button	Yes	**Auto detect**: The user should select this option to auto-detect and add the peer devices in the inventory page. **Manual entry**: The user can use this option to add the peer devices manually.	NA

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

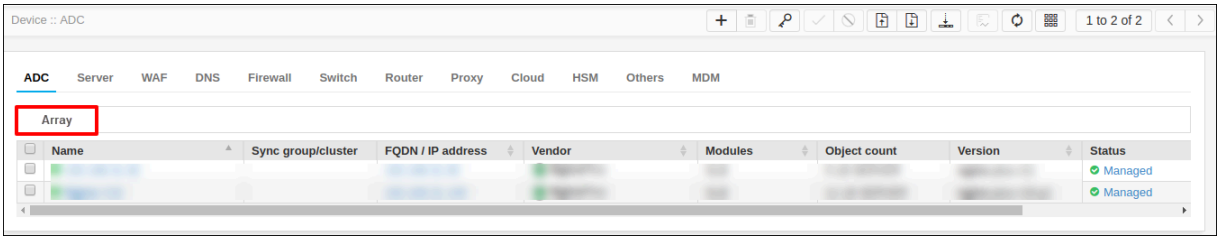
Name	Type	Mandatory	Description	Validation
			Ignore: The user can use this option to ignore the auto-detection of the peer devices.	

8. Click **Save**.

Validating the NginxPlus Device Addition

After adding the device, you can validate the device by searching device in the device inventory.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Search the device name and validate whether the device is added successfully.



What's Next

- Check the Logs for transaction ID and details

Device Group

Group Overview

Device group functionality enables categorizing the ADC devices as per custom demands. For example, ADC devices can be grouped based on business units, data centers, global locations, users, or custom needs. This will be helpful in accessing devices of similar types. These custom device groups are efficiently used while scheduling group-based backups and configuring a Heatmap Widgets for monitoring health and triggering automation changes.

Add an ADC Group

To add an ADC group to AppViewX:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **Asset Management** > **Device Group**.
   The **Group** screen opens.
2. In the **ADC** tab, click the (**Add**) button in the Command bar.
3. On the **Add** screen, enter the name of the new group. (Recommended) Enter a description of the group to help users identify it.
4. In the Device selection field, click the (**Assign item**) icon beside each device you want to include in the group.
5. When you have finished assigning devices to the group, click **Save** to add them to the system.

> **Note:**
> Rather than adding devices manually, you can click the Add search string link and create a search string that automatically assigns all existing devices that match the filter criteria to the ADC group. The benefit of using a search string rather than selecting devices manually is that the search string continues to work in the background, auto-assigning all new devices to the ADC group if the devices match the search criteria you set up.

Modify an ADC Group

To modify an ADC group to AppViewX:

1. Go to **Menu** > **ADC+** > **Asset Management** > **Device Group**.
   The **Group** screen opens.
2. If the ADC group whose details you want to modify is not displayed on the screen, use the search field to locate it.
3. In the **ADC** tab, select the checkbox beside the name of the ADC group.
4. Click the **(Modify)** icon in the Command bar.
5. On the Modify screen that appears, make whatever changes you want to the content.
6. Click **Update** to save your changes.

Delete an ADC Group

To delete an ADC group to AppViewX:

1. Go to **Menu** > **ADC+** > **Asset Management** > **Device Group**.
   The Group screen opens.
2. Select the checkbox beside the group you want to delete.
3. Click the (**Delete**) button in the Command bar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. On the confirmation screen that pops up asking you if you are sure you want to proceed, click **Yes**.
5. The group is then removed from the AppViewX system.

Config Management

Backup Configuration

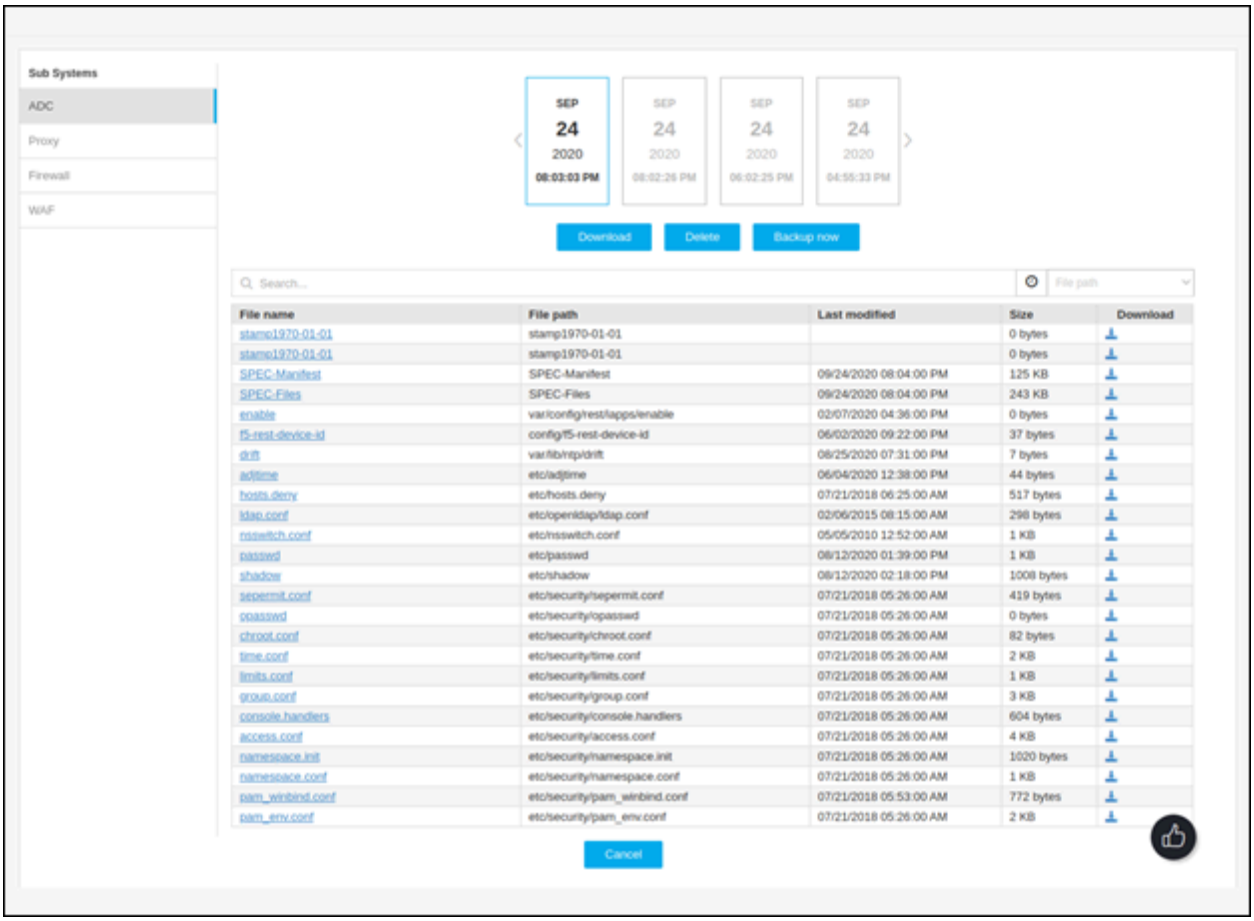
Manage the complete backup of your devices stored in a central repository in AppViewX. Define your own groups as per the business unit or data center and scheduled backups as needed.

The Backup groups are listed with the devices and their respective archives (with timestamp). Click on the device name to get into the details - A detailed Archive management screen appears that allows you to do the following:

- Click on the Archive to view the file/config within the archive.
- Download the complete archive or individual files.
- Delete the selected archives
- Generate Instant backup

To improve system efficiency and offload backup operations from core subsystems, the `avx_device_backup_bot` pod can be enabled to handle all ADC device backup requests. See AppViewX Setup guide for enabling and disabling Device Backup pod.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Backup Configuration Management

Application Delivery Controller solution enables periodical backups to be done on a manual basis or with a scheduling system, which ensures that the entire config archives of all devices are added and managed in a central repository with AppViewX. Along with the ability to perform the backup, you can also restore the archive back to the device either complete archive of device or object level (Virtual Server, Wideip, Monitor, policy, etc.) based on the requirement (for example, restore only one virtual server that was misconfigured during modifications).

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Benefits of Configuring Backup

- Ability to compare between two archives to see exactly what changes have taken place between the time frame of archive 1 (vs) archive 2.
- Essential details can be viewed and downloaded as pdf such as the files within the backup folder bigip, conf, etc. for your reference
- If the backup is scheduled on a regular basis, whatever misconfigurations or misbehavior of network outage or power failure on the device can be recovered and restored for valid configuration data.
- Manage your network configuration by automating the backups, track and report network changes and ensure they are compliant with the defined policies.
- Audit the configuration drift and prevent unauthorized or unwanted changes by comparing the configuration across devices visually and restore the changes to adhere to the standards.

Create a Device Backup Group

A device backup group is a container used to store all of the backups and restore records for a particular group within the AppViewX system. For RSeries host device backup, see the RSeries Device Backup workflow.

To create a device backup group:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Create Backup**. The **Create** screen opens.

   Note:
   Rather than adding devices manually, you can click the Add search string link and create a search string that automatically assigns all existing devices that match the filter criteria to the backup group. The benefit of using a search string rather than selecting devices manually is that the search string continues to work in the background, auto-assigning all new devices to the backup group if the devices match the search criteria you set up.

2. Enter a name for the backup group.
3. (Recommended) Enter a description of the group that makes it easy for users to determine what sort of device backups are found within the group.
4. Select the **Device** or **Device Group** radio button based on how you want to define the device backup.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. In the **Available** devices field, click the  (**Assign item**) icon beside each device whose backups and restores you want to include in the group.
6. In the **scheduling** field, select either the **Scheduler** radio button and then set the frequency, starting date, and time for the backups, or select the **Generate** now radio button to start the backup as soon as you click Save.
7. In the **Email configuration** field, enter the email addresses, separated by commas, of all users who should be sent a copy of the backup.
8. (Recommended) Enter a short, clear description in the **Subject** field so that it will help the recipients understand why they are receiving the email: for example, "Weekly backup of ADC devices."
9. Click **Save**.

> **Note:**
> You can customize the archive count for storing the daily, weekly, monthly, and yearly backups individually. Using this feature, you can maintain scheduled archives without it being overwritten by instant backups.

Enable External Storage for Backup

Prerequisites:

• Ensure the target server is added and managed under **Device Inventory > Others** tab.

To enable external server for bakup:

1. Go to **Menu** > **ADC+** > **SETTINGS** >Module Settings > Device.
2. Click the **Backup** tab.
3. From the **Server Name** dropdown, select the desired server.

> **Note:**
> Only servers listed under the **Others** inventory are displayed here.

4. Enter the **File Storage Path** where the backup should be pushed on the external server.
5. Click **Save**.

Once the external backup configuration is enabled, backups will be stored both in **AppViewX** and on the configured **external server**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
All backup operations are recorded in the **Logs** for auditing and tracking purposes.

What's Next

- Create a Device Backup Group

Delete a Device Backup Group

To remove a device backup group from the AppViewX system:

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Backup** tab opens.
2. Click the (**Delete**) icon against the device backup group that you want to delete.
   The **Confirmation** pop-up opens.
3. In the **Confirmation** pop-up, you have an option to enable Retain the generated backups. This option is enabled by default.

   Note:
   The retained backups will be moved to the **Default group** for later references.

4. Click **Yes** to confirm that you want to delete the device backup group.

   Note:
   To discard the deletion, click **No**.

Edit the Details of a Backup Group

To edit the details of a device backup group:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Backup** tab opens.
2. In the list of backup groups on the screen, click the (**Edit**) icon for the device backup group you want to modify.
   The **Modify** screen opens.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. In the **Modify** screen, update description, add/remove devices, change scheduler interval, and Email options.
4. Click **Save**.

Initiate Instant Device Backup

A device backup group is a container used to store all of the backups and restore records for a particular group within the AppViewX system.

To create an instant device backup group:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Backup** tab opens.
2. If the **Backup** tab is not displayed by default, click to open it.
3. In the list of backup groups on the screen, click the (**Expand**) icon for the device you want to back up immediately.
4. Click the **Device name** link in the table that appears.
5. On the **Archive details** screen that appears, click **Backup Now**.

Schedule a Device Backup

To schedule a device backup:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Backup** tab opens.
2. Click the **Create** button in the Command bar.
3. On the **Create** screen that opens, enter a name for the backup you are about to schedule.
4. (Optional) Enter a description of the backup.
5. In the **Available devices** field, click the **Assign** icon beside each device you want to add to include in the backup.
6. Select the **Scheduler** radio button.
7. Depending on how often you want to schedule a backup, select the Daily, Weekly, Monthly, or Yearly radio button.
8. Type the starting date of the backup or click the (**Starting date**) icon and select the date from the popup screen that appears.
9. Type the starting time for the backup or click the (**Time**) icon and select a start time using the sliding bars for Hour and Minute. When you are finished, click Done to close the popup screen.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. (Optional) If you want certain users to be notified of the backup generation status (success or failure) via an email, enter their email addresses separated by commas in the To field of the Email configuration section, then enter a subject that will help the recipients understand why they are receiving the email: for example, "Weekly backup of ADC devices."

    Note:
    Email notifications will be sent for each device within the backup group. The successful backup notification will have the archive attachment in .ucs format via email.

11. Click **Save** to finish scheduling the backup.

    Note:
    - Rather than adding components manually, you can click the Add search string link and create a search string that automatically assigns all existing devices that match the search criteria. The benefit of using a search string rather than selecting devices manually is that the search string continues to work in the background, auto-assigning all new devices to the backup if the devices match the search criteria you set up.
    - The minute bar has only two settings: to the left, representing the top of the hour (:00) and to the right, representing half-past the hour (:30).

View the Backup Schedule for a Device

To view the backup schedule for a device:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Backup** tab opens.
2. In the **Backup** tab, Hover your mouse over the Calendar icon of the device whose backup schedule you want to view.
3. A popup box appears, listing the frequency, start date, and start time of backups for the selected device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
If a device has no backups scheduled, no calendar icon appears in its row in the device table, as is the case for the a10 device that appears in the image above.

Delete the Backup and Restore History for a Device

To delete the Backup and Restore history:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Restore** tab opens.
2. In the list of backup groups on the screen, click the (**Expand**) icon for the device backup group whose backup and restore history you want to delete.
3. In the table that appears, click the Device Name link for the device whose history you want to delete.
4. On the **Archive details** screen that appears, click the **Delete** button.
5. On the popup screen that appears, click **Yes** to confirm that you want to delete the backup and restore history for the device group.

Download the Backup and Restore History for a Device

To download the Backup and Restore history:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Restore** tab opens.
2. In the list of backup groups on the screen, click the (**Expand**) icon for the device backup group that contains the device whose history you want to download.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. In the table that appears, click the Device name link for the device whose history you want to download. The Archive details screen appears, listing all of the back-ups and restore events for the device.
4. Perform one or both of the following download actions:

   - Scroll through the date range at the top of the screen, click on a specific date, then click the Download button to download all backup and restore events that occurred on that date for the device.
   - Use the search field and calendar filter to locate specific backups or restore events based on filename, file path, or date range, then click the (**Download**) icon for each result you want to download.

View the Backup and Restore History for a Device

To view the Backup and Restore history:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Restore** tab opens.
2. In the list of backup groups on the screen, click the (**Expand**) icon for the device backup group that contains the device whose history you want to view.
3. In the table that appears, click the Device name link for the device whose history you want to view.
4. The Archive details screen appears, listing all of the back-ups and restore events for the device.
5. (Optional) Use the search field and calendar filter to locate specific backup or restore events based on filename, file path, or date range.

Edit the Settings of the Backup Screen

The device backups are referred to as Archives. Customize the Archives limits per device (maximum of 80 archives per device can be configured) through the customization option. Upon crossing the limit, the oldest archive will be overwritten automatically. You can also distribute the archive count against Daily, Weekly, Monthly, and Yearly flags so that scheduled archives are not overwritten soon. Save the changes upon configuration.

To edit the settings of the Backup screen, complete the following steps:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   The **Restore** tab opens.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

2. In the Sub Systems column on the left, click ADC (default tab).
3. Click the (**Settings**) button in the Command bar.
4. On the Settings popup screen that appears, select the number of archives you want to keep for each device: you can choose to keep anywhere from 1 to 80.
5. Select the Display warning messages when overwriting the archives checkbox if you want users to be warned whenever they are about to overwrite an archive.
6. Click **Submit** to save the settings.

Manage Host Backup

To manage host backups:

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
   Byfault, the **Backup** tab opens.
2. In the Sub Systems column on the left, click ADC (default tab).
3. Click the **Host Backup** tab.
   The host backups are displayed.
4. Click the download or delete icon to download or remove the backups as required.

Restore Configuration

AppViewX allows you to restore configuration backups on ADC devices. It supports,

- device-level to overwrite the complete device config
-  object-level restoration to restore a particular Application that is failing or detected with an unwanted change.

Restore and Rollback a Device

Restore the configuration of the ADC device from current config to a preferred config. AppViewX enables to compare the configuration before proceeding with the restoration. During a restore, AppViewX will take a backup of the current configuration that is to be used during rollback.

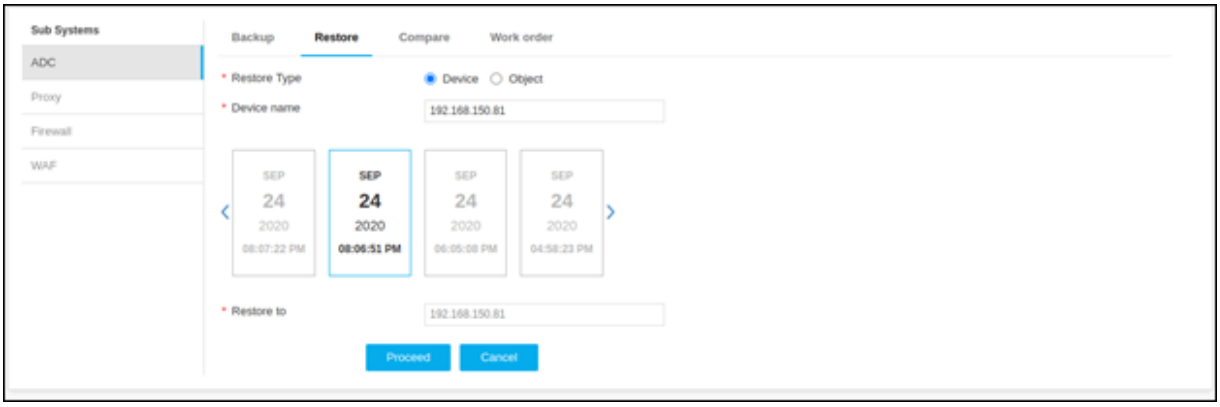
On proceeding with restore, a Work order will be generated in AppViewX to track progress. On a successful restore, required configurations will be updated in the device. In case of failure, an automatic rollback will be initiated by AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

For RSeries host device restore, see the RSeries Device Restore workflow.

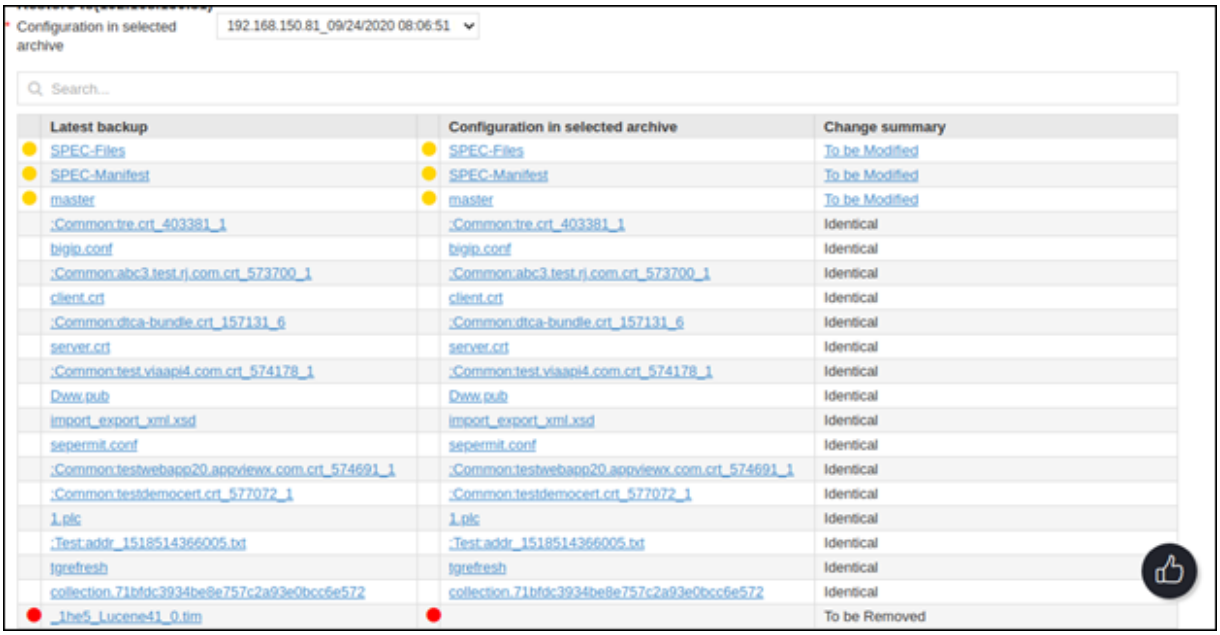
To restore a device:

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Restore Configuration**.
   The **Restore** tab opens.
2. Select the **Restore Type**: Device.
3. Select a **Backup Location**: Local.
4. Enter the **Device name**.
5. In the date range field that appears below the **Device name** field, click the date you want to restore the device or object to.
6. Leave the default value in the **Restore to** field.



7. Click **Proceed**.
   The screen refreshes and displays a Configuration in the latest archive field, which should show the backup you selected. The table below shows all of the files contained in the backup and displays yellow circles beside each file that has been modified since the backup was taken and red circles beside each file that has been removed since the backup date.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



8. At the bottom of the screen, enter a reason for restoring to the backup.
9. Click **Restore**.
10. Validate the changes against the latest backup and the configuration that is going to be restored. Enter a reason for restore and proceed.
11. AppViewX will proceed with restoring the entire configuration file or the particular configuration of the objects in a step-by-step manner.
12. A **Work Order ID** will be generated to track the progress at anytime. (Track the IDs in the Work order tab)
13. During restoration, if AppViewX identifies any issues, an automatic configuration Rollback will be triggered, which can be tracked in the work order.
14. A manual Rollback can also be initiated on a need basis.

Restore and Rollback a Device Using External Storage

Restore the configuration of the ADC device from current config to a preferred config using the configuration stored in the external storage. AppViewX enables to compare the configuration before proceeding with the restoration. During a restore, AppViewX will take a backup of the current configuration that is to be used during rollback.

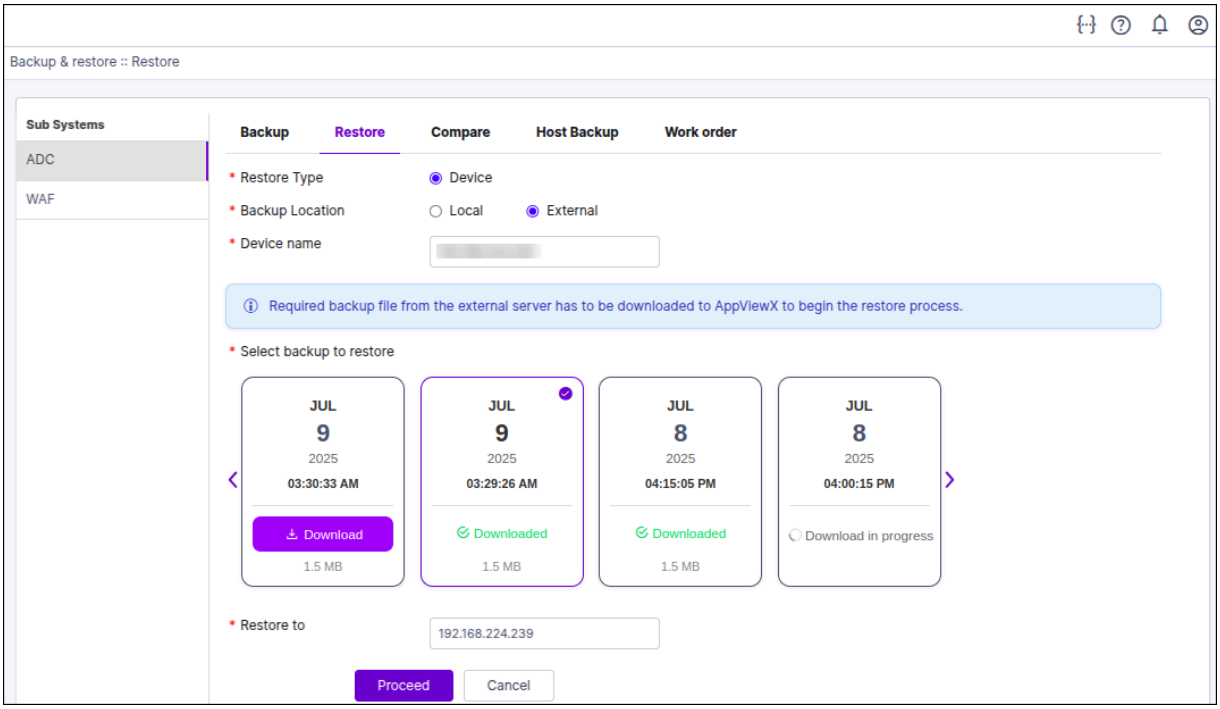
On proceeding with restore, a Work order will be generated in AppViewX to track progress. On a successful restore, required configurations will be updated in the device. In case of failure, an automatic rollback will be initiated by AppViewX.

For RSeries host device restore, see the RSeries Device Restore workflow.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

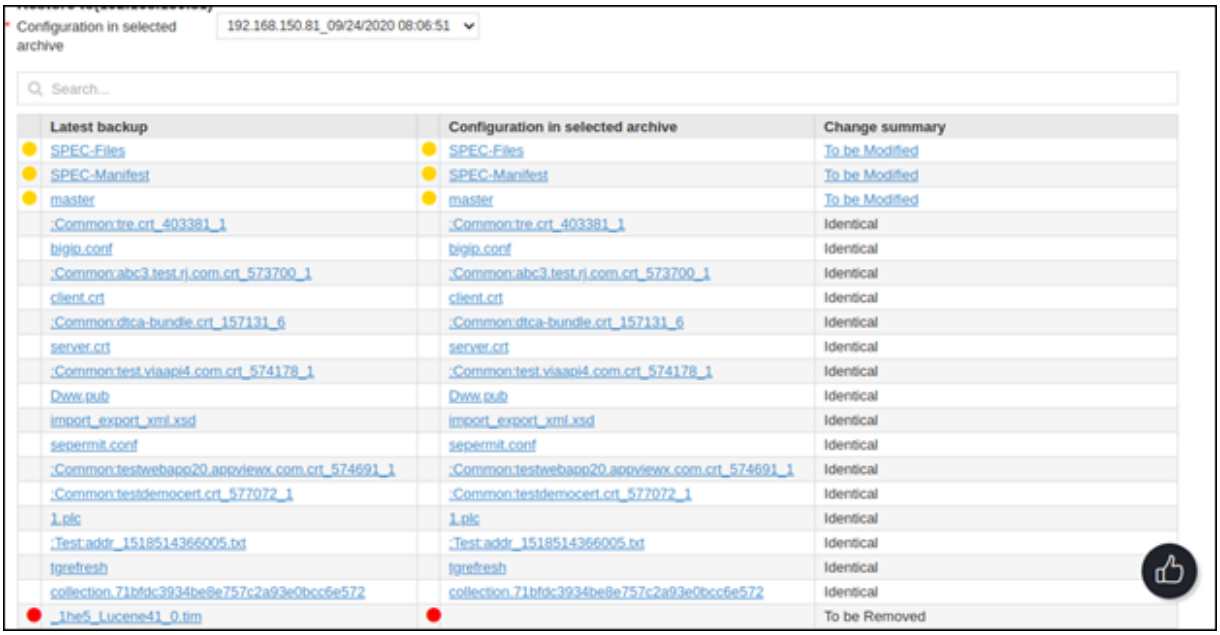
To restore a device:

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Restore Configuration**.
   The **Restore** tab opens.
2. Select the **Restore Type**: Device.
3. Select a **Backup Location**: External.
4. Enter the **Device name**.
5. In the date range field that appears below the **Select backup to restore** field, click the date you want to restore the device.
6. Click Download of the selected date to download the backup file from the external storage.
7. Leave the default value in the **Restore to** field.



8. Click **Proceed**.
   The screen refreshes and displays a Configuration in the latest archive field, which should show the backup you selected. The table below shows all of the files contained in the backup and displays yellow circles beside each file that has been modified since the backup was taken and red circles beside each file that has been removed since the backup date.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



9. At the bottom of the screen, enter a reason for restoring to the backup.
10. Click **Restore**.
11. Validate the changes against the latest backup and the configuration that is going to be restored. Enter a reason for restore and proceed.
12. AppViewX will proceed with restoring the entire configuration file or the particular configuration of the objects in a step-by-step manner.
13. A **Work Order ID** will be generated to track the progress at anytime. (Track the IDs in the Work order tab)
14. During restoration, if AppViewX identifies any issues, an automatic configuration Rollback will be triggered, which can be tracked in the work order.
15. A manual Rollback can also be initiated on a need basis.

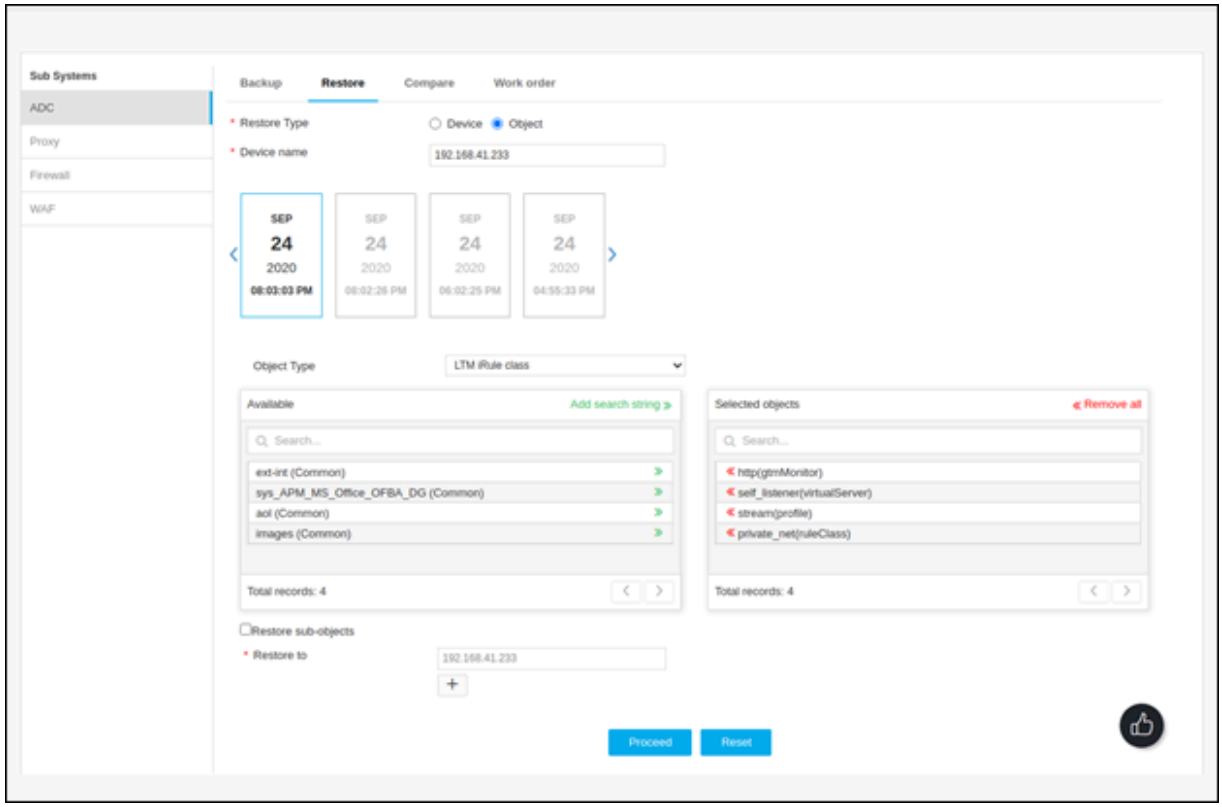
Restore and Rollback an Object

Restore one or more Application services configurations from the current config to a preferred config and apply it on multiple devices. AppViewX enables to compare the configuration before proceeding with the restoration. During the restoration, the complete hierarchy of the selected object(s) will be restored in a step-by-step manner. AppViewX will consider the recent device backup as the source to roll back the configurations. On successful restore, configurations will be updated in the device. In case of failure, an automatic rollback will be initiated by AppViewX.

To restore an object:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Restore Configuration**.
   The **Restore** tab opens.
2. Select the **Restore Type**: Object.
3. Select the Source device, Archive to be used for restore.
4. Select the objects that need to be restored.
5. Select the destination devices (it could be one or more devices), and then click **Proceed**.



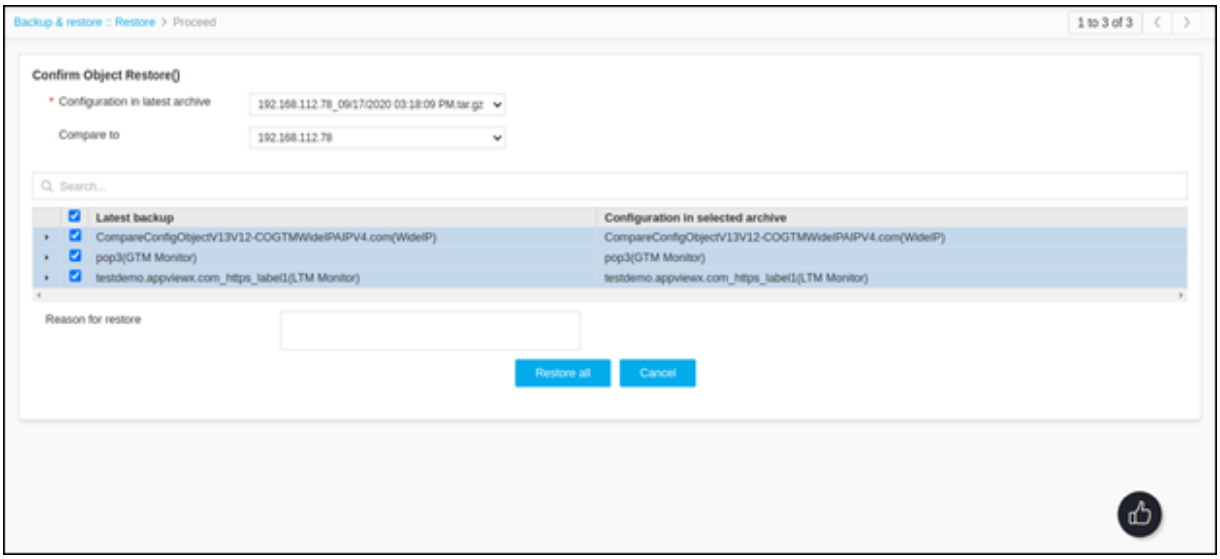
6. The screen refreshes and displays a table that compares the latest device backup and the selected archive configuration with change highlighted. Each object can be expanded and viewed:

   • Yellow - Modified line

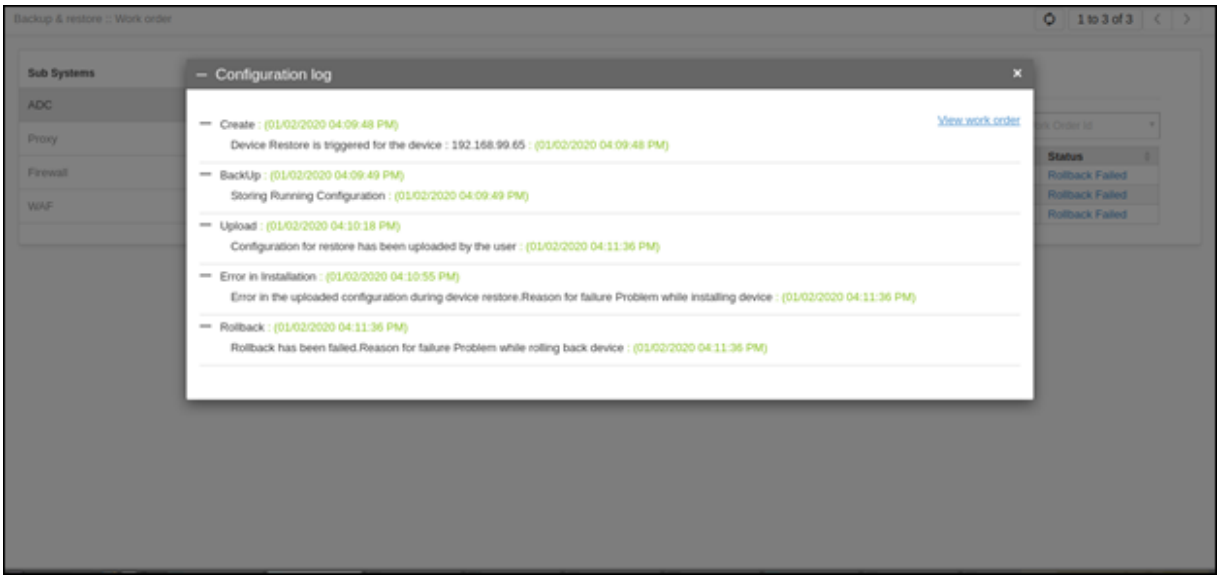
   • Green - Added line

   • Reg - Deleted line

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. Validate the changes against the latest backup and the configuration that is going to be restored. Enter a reason for restore and proceed.
8. AppViewX will proceed with restoring the particular configuration of the objects selected in a step-by-step manner.
9. A **Work Order ID** will be generated to track the progress anytime. (Track the IDs in the Work order tab)
10. During restoration, if AppViewX identifies any issues with respect to even one of the objects, an automatic configuration Rollback will be triggered, which can be tracked in the work order.
    A manual Rollback can also be initiated on a need basis.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Compare Configuration

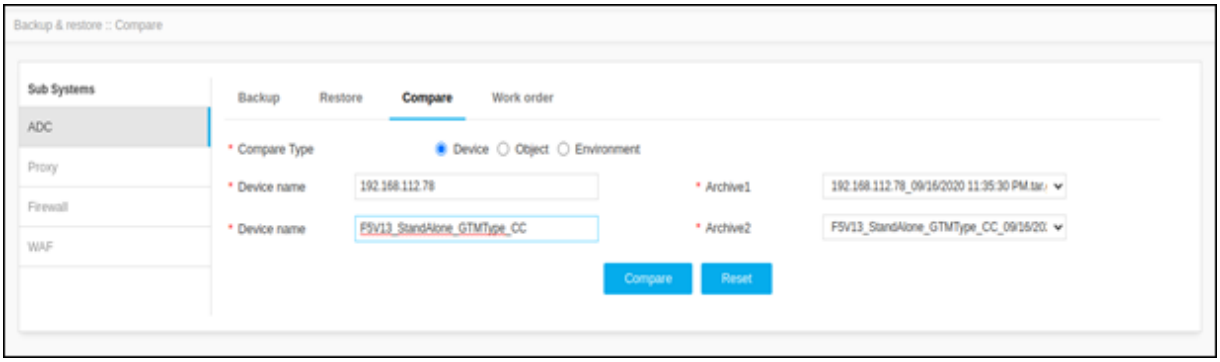
AppViewX platform enables you to compare the complete configuration of load balancers within or across multiple devices through previous backups. It indicates the files that are modified and also provides an option to drill down into the actual configuration change. Ensure the changes are validated and generate reports out of it.

AppViewX allows the configuration to be compared for the entire Device, Specific object within the device, or a custom Environment as per the need.

Compare Device Backups

To compare backups for the same or different devices:

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Compare Configuration**.
   The **Compare** tab opens.
2. On the **Compare** screen that opens, select the **Compare Type**: Device, Object, or Environment.
3. Select the Device names for which the configuration needs to be compared. The devices that have valid backups generated will be listed here.
4. In the **Device name** field, enter the name of the first device whose backup you want to compare.
5. In the **Archive 1** field, select the first backup in the comparison.
6. In the second **Device name** field, enter the name of the second device in the comparison. If you want to compare backups from the same device, enter the same name you entered in Step 4.
7. In the **Archive 2** field, select the second backup in the comparison.
8. Click **Compare**.



9. A table appears at the bottom of the screen, showing all of the files contained in the two backups you selected. The files are compared globally and a change summary is listed,

   • **Yellow** - Denotes the modifications

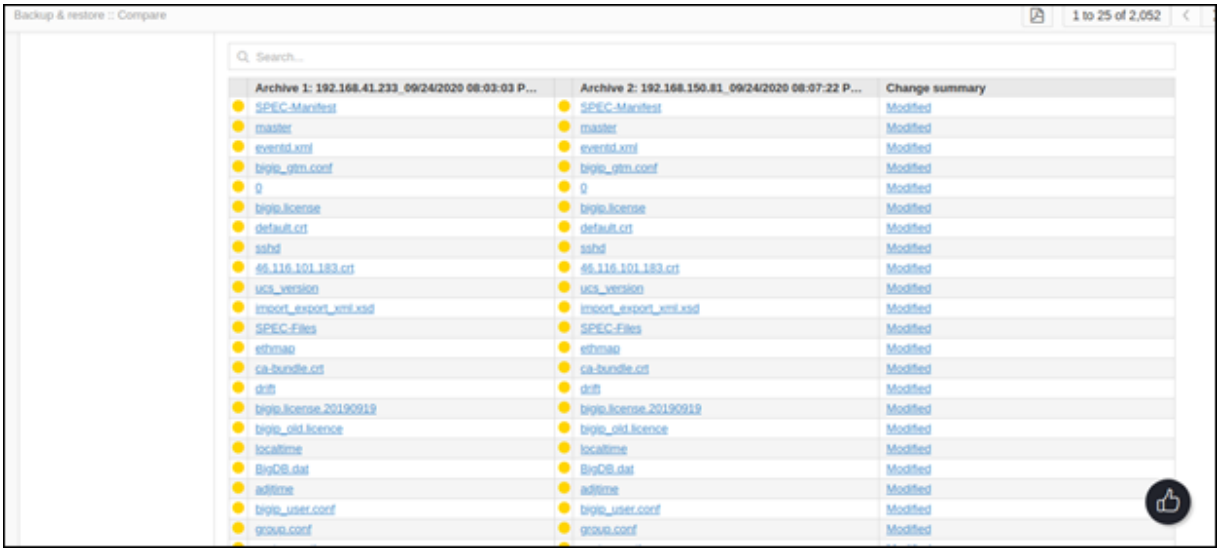
   • **Green** - Denotes the new additions

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Red** - Denotes the deletions

> **Note:**
> Click on the **Modified summary** for a line-by-line comparison of a particular file that is labeled as modified. This will help us to which line of the configuration is modified exactly thus helps to troubleshoot changes faster.

10. Click on the **File name** to get the configuration details.
11. Change **Archive name** to compare against a different configuration backup.
12. Get the high-level device file summary and it can be exported using the Export as PDF option.



Compare Multiple Configurations of an Object

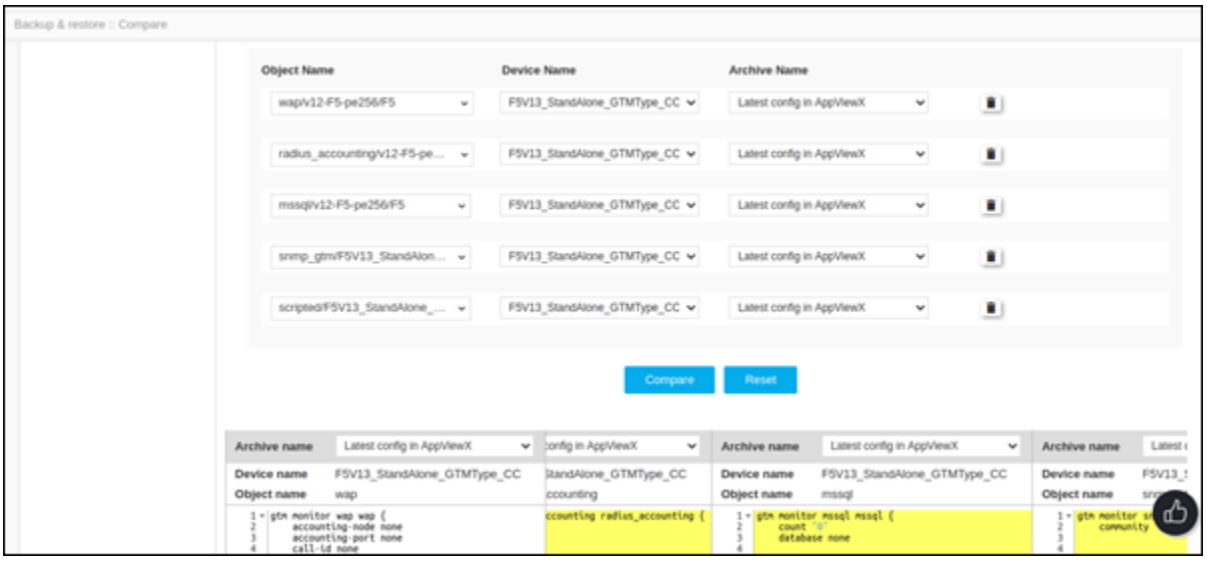
Compare the object-level configuration of load balancer devices using previous backups or the latest configuration in AppViewX. Ensure a particular object (Say iRule) config is compliant across all the devices in a business unit. You can also track the exact configuration changes on your application by comparing them against all the recent archives.

To compare multiple configurations of an object:

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Compare Configuration**.
   The **Compare** tab opens.
2. Select the vendor whose objects you want to compare.

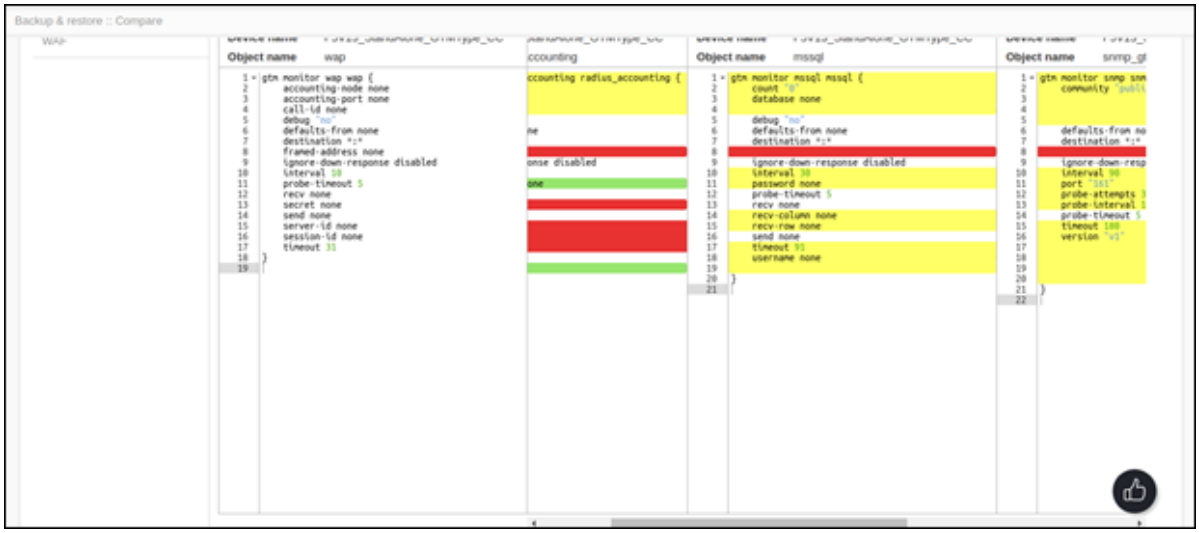
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Select the object type you want to compare. The possible object types are WideIp, GTM Pool, GTM Server, GTM Monitor, GTM iRule, VirtualServer, LTM Pool, LTM Monitor, Profile, LTM iRule, LTM iRule Class, LTM Policy, LTM Persistence, Traffic Record, Traffic Region.
4. Click **Object Name** dropdown list and select the name of the object whose configurations you want to compare.
5. In the **List of Devices** field that appears, click the first **Device Name** dropdown list and select the first device you want to use in the comparison.
6. If you want to compare configurations of the same device over time, choose the device you selected in Step 8.
7. In the **Archive Name** field, select the first configuration you want to use in the comparison.
8. In the second **Device Name** dropdown list, select the second device you want to use in the comparison.
9. In the second **Archive Name** field, select the configuration you want to compare the first configuration to.
10. Click the **Add** button to add more configurations of an object.
11. Click **Compare**.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- The screen refreshes and displays the two archived configurations side-by-side.
- A maximum of five archives can be compared.



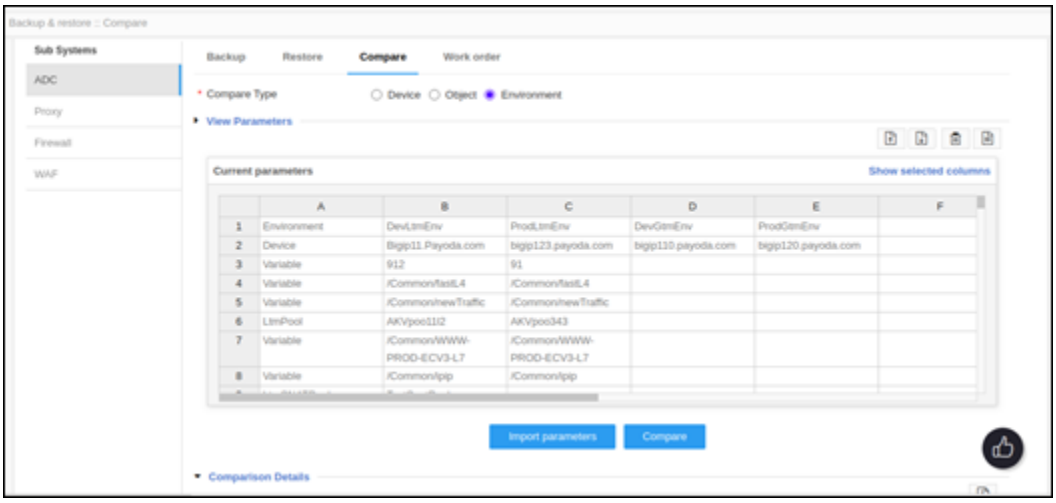
Compare Configurations of Custom Environments

AppViewX allows you to compare the custom environment configuration. A custom environment could be specific to a business unit comprising specific devices and objects. You can import the custom environment details (devices and objects) into AppViewX in the form of an Excel file and compare the source and destination environment configuration of F5.

To compare configurations of custom environments:

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Compare Configuration**.
   The **Compare** tab opens.
2. In the **Compare Type** field, select the Environment radio button.
3. Click the icon beside **View Parameters** section to expand it.
   The **Current Parameters** table is displayed.
4. In the table, you can export, import, or delete parameters; download the sample using the buttons on the top-left corner. You can also right-click inside the table to delete a row; select, delete or unselect a column.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



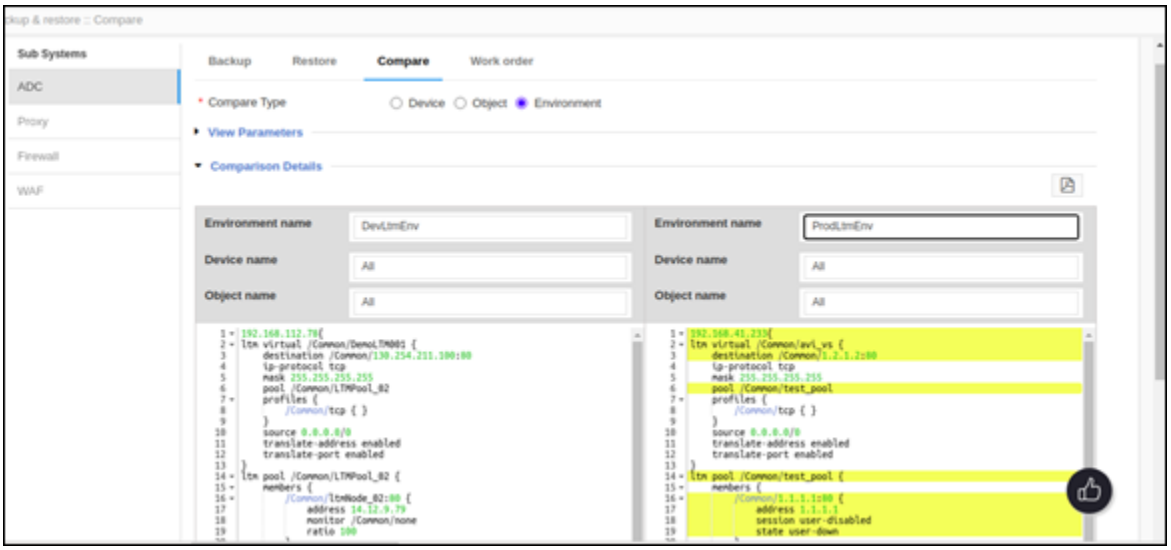
5. Click **Import Parameters**.
6. Click **Compare** to compare the source and destination environments.
7. Select the environment names, device names, and object names to be compared from the respective dropdown lists in source and destination environments. A line-by-line comparison will be displayed below each environment with the following color coding:

   • **Green** - Denotes the new additions

   • **Yellow** - Denotes the modifications

   • **Red** - Denotes the deletions

8. If required, you can click the (**Export to PDF**) button to download the comparison report to your computer.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Enforce Golden Config Compliance

A “golden” configuration is a configuration version that can be used as a standard configuration for error-free and rapid deployment. Automation workflows help you create golden configurations adhering to organizational policies.

For more details, see the Enforce Golden Config Compliance in the Workflow section.

Automation

Workflow Catalog

The AppViewX Platform helps Enterprise IT manage, automate, and orchestrate application delivery services.

Key Advantages:

- Abstract hardware, software, open-source, and cloud solutions from end-users requesting application delivery services
- Orchestrate how GSLB, DNS, load-balancing, firewall, WAF, certificate, network, ITSM, and notification services are leveraged in the multi-cloud infrastructure.
- Integrate industry-leading ITSM and notification services into automation workflows.

- Expose network infrastructure services to DevOps tools like Ansible, Terraform, Chef, and Puppet.
- Build repeatable and compliant infrastructure with workflows that map to your business process.
- Use the business process to enforce how compliance is achieved in multi-cloud data centers.
- Automate how unused resources are returned back to the application delivery infrastructure.

- Enable end-users to self-service and launch automation from application-centric views.

- Enable common load-balancer and DNS service requests to be self-serviced by application and operation teams.
- Minimize the operational expense of maintaining automation workflows when changing vendors or versions.

Automate and Self-Serve ADC deployments with Pre-packaged Workflows

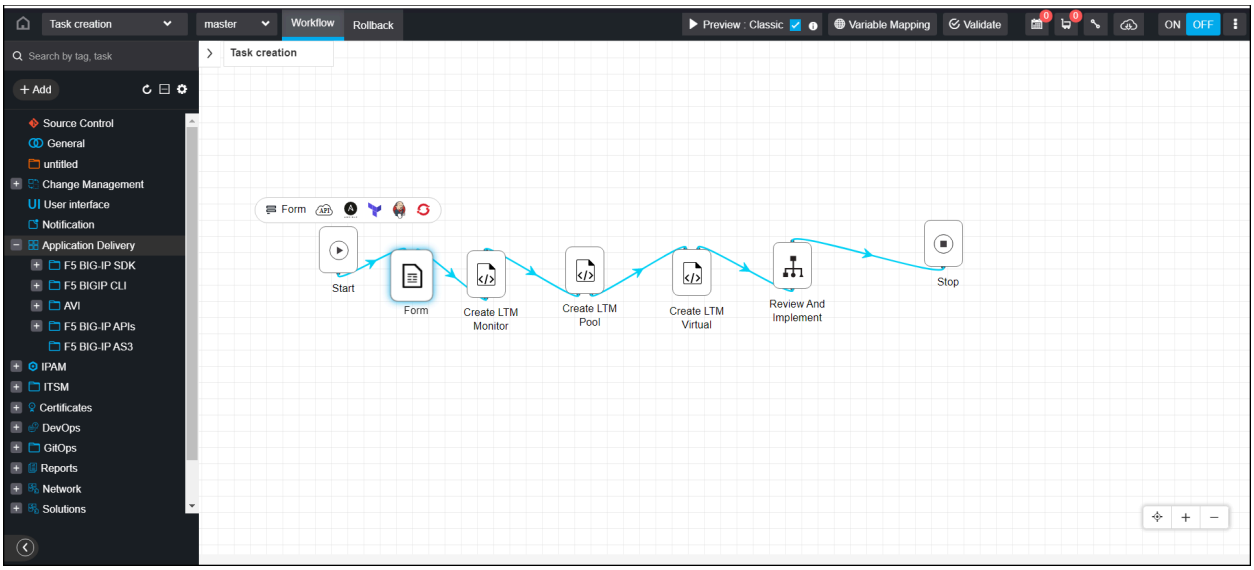
AppViewX is a true multi-vendor platform for managing the ever-evolving complex enterprise infrastructure. As your application delivery infrastructure becomes progressively more complex,

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

enterprises must turn to network infrastructure management and automation solutions to ensure successful deployments. AppViewX offers a suite of tools for organizations striving to become true digital enterprises.

The AppViewX platform will help you automate third-party, best-of-breed, and open-source network services like those provided by application delivery controllers, security devices, certificate authorities, DNS servers, routers/switches, and more.

The AppViewX platform integrates with other best-in-class IT service management solutions to govern and record service requests in the larger automated workflow. It can be leveraged to help users move faster, eliminate errors, and reduce costs, making it the best choice for enterprises working to adapt to new technologies, processes, and application delivery expectations.



Workflow Catalog Page

AppViewX offers Visibility on the One-touch Automation solution offerings to address network and application service requests. The offerings are available in the Request catalog page. Click **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**. The automation offerings (workflows) are controlled by AppViewX's modern RBAC control feature, thus providing individuals groups to maintain and access their own workflows.

Multi-vendor ADC Application Deployment

As the application deployment in enterprises is complex, time-consuming, and cumbersome, AppViewX's automation solution helps in speeding up the process to bridge the isolation across multiple teams that are involved in application deployment. AppViewX also integrates with leading technology and best-in-

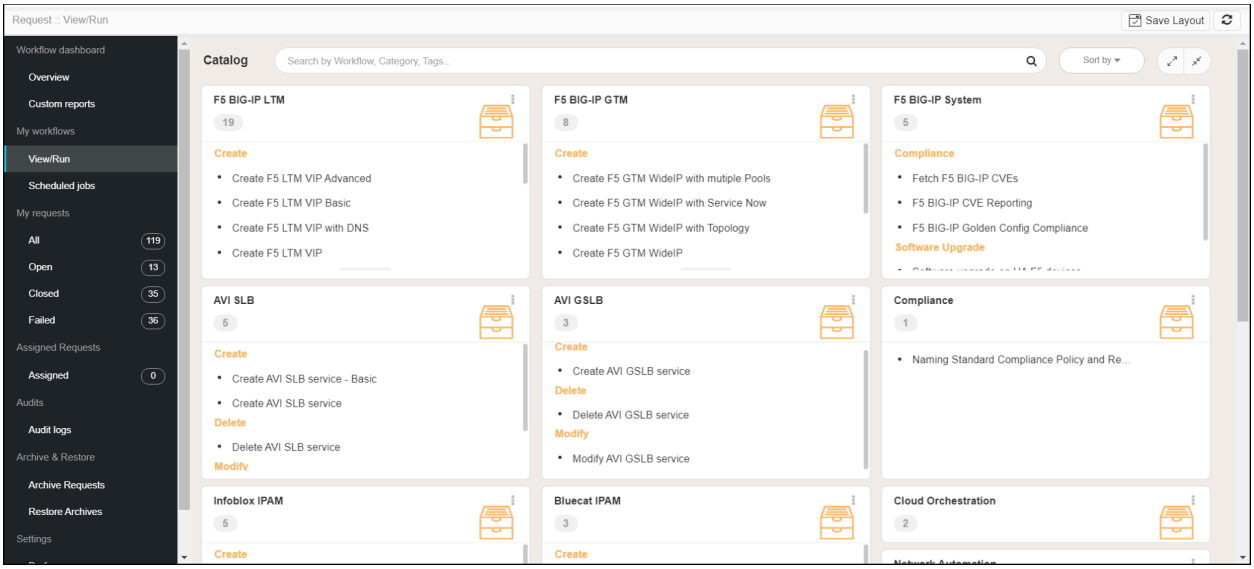
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

class IT service management solutions to govern and record service requests in the larger automated workflow. It can be leveraged to help users move faster, eliminate errors and reduce costs.

The prebuilt automation workflows deliver the following when it comes to Application deployments,

- Creation/Modification/Deletion of WideIPs and Virtual Servers on AppViewX managed ADC devices through simple, self-service forms.
- Integration with DDI (Bluecat, InfoBlox), and ITSM (ServiceNow) enabling a consistent change management control and governance.
- Integration with AS3 (declarative approach) for faster Application deployment by enabling Creation/ Modification/Deletion of BIG IP Application Services.

To access the prepackaged workflows, click **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run** The workflows are grouped as per the vendors, select the required flow and invoke the required flow to deploy/manage an application.



Refer to Visual Workflow Guide for creating workflows

https://pages.appviewx.com/rs/249-TWN-899/images/Guide_App_Provisioning_Automation%20Workflow_AppViewX.pdf.

Process Flow for Create Virtual Server (sample)

The Create Virtual Server workflow creates a virtual server and associates it with profiles, monitors, pool, and pool members in F5 LTM using Infoblox and ServiceNow integration. It uses a simple, self-service based approach to gather application-provisioning requirements and generate vendor-specific

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

configurations or REST APIs. This self-service workflow filters F5 ADC devices based on the user's access permissions, defined by Role-Based Access Control (RBAC). The platform integrates with IPAM systems like Infoblox, which allows users to reserve a free IP address from the available address pools and create DNS binding for the new virtual server in Infoblox.

The workflow also includes an option to create or bind existing profiles and monitors to the virtual server and allows users to create change request tickets in ITSM systems ¬like ServiceNow for approvals and tracking. The service request change ID is associated with the work order and is updated based on the implementation status.

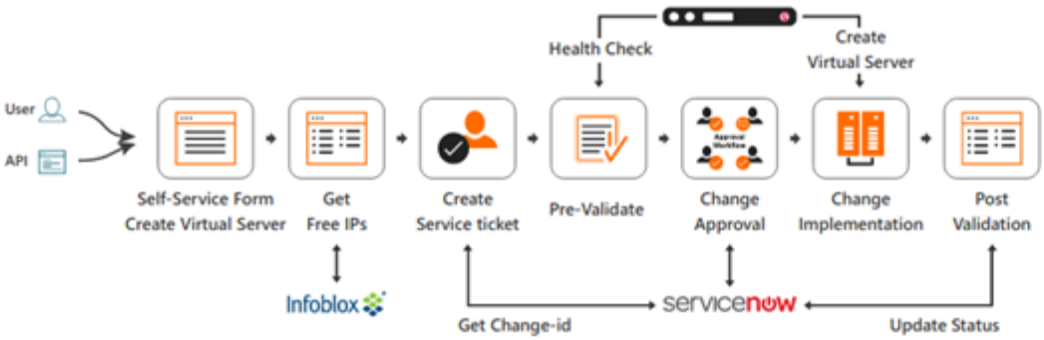


Figure 2: Automate Application Provisioning on ADCs

Refer to ADC+ Automation Guide for more detail.

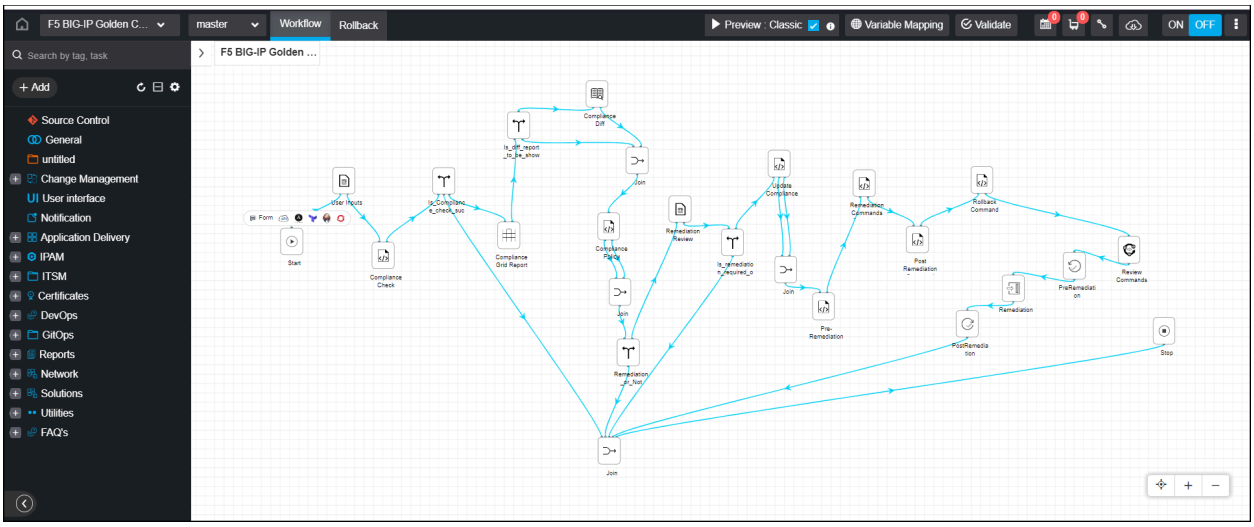
Enforce Golden Config Compliance and Migrate Configurations

In this digital world, organizations depend on the continuous availability of applications and networks for business success. The advent of digitalization has changed the way the network is being managed. In heterogeneous networks, network leaders face various challenges in properly managing configurations, carrying out changes, ensuring compliance to regulations, and minimizing network downtime triggered by human errors.

AppViewX offers a Prepackaged 'Golden Configuration Compliance on F5 BIG-IP' workflow to run a compliance check across all the managed F5 BIG-IP devices. The workflow is designed to validate the system-wide configurations on F5 Devices for configuration standardization or compliance purposes. Thus maintaining the Integrity and security of an enterprise-level infrastructure.

Click **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run** > **F5 BIG-IP System** > **Compliance**. Invoke the workflow 'F5 BIG-IP Golden Config Compliance'.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- Create golden configurations adhering to your organizational policies and store different file variants in the AppViewX collection module.
- Execute the workflow by selecting the Golden configlet file (Choose an existing file or create new) and the devices on which it needs to be applied. AppViewX validates the system-wide configurations on F5 Devices for configuration standardization or compliance purposes.
- AppViewX generates a standard config and pushes it to the end devices based on approvals.

Refer to ADC+ Automation Workflow Guide for more details.

Compare Configurations in workflows

AppViewX provides a flexible component (Diff checker) to validate the configuration changes over time. As avoiding configuration drift is a major challenge, AppViewX's visual comparison tool that could be integrated into the workflows helps to overcome this scenario and ensures devices are compliant. Access the component by navigating to **Menu** > **Studio** > **Workflow** > **Create** > **Diff checker component**. Drag and Drop the component into the required workflow wherever comparison is needed.

It allows static and dynamic comparison of configurations across two to five devices and objects. Diff View provides a side-by-side comparison of configuration files in a single view. And, it also highlights the configurations that are added, modified, or deleted in different color codes.

With Diff Checker, the AppViewX platform helps to perform compliance configuration checks across devices and allows the configurations to adhere to standards.

- Validate configuration changes between pre-validation and post-validation stages
- Validate for configuration changes between implementation and rollback stages

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Validate for configuration changes between multiple peer reviews within an automation process
- Validate if device configurations are compliant with the standard golden configurations

Optimize Load Balancer with Unused Object Decommissioning

Manage unused configurations during hardware replacement or version upgrades to optimize your load balancers. Set of Virtual Servers/Objects that do not handle traffic for a certain period of time (three months or more) are often referred to as unused objects. Unintentionally maintaining these inactive objects leaves many unwanted IP ports open and their IPs unused and also incur overhead costs for the enterprises. With AppViewX's Application Delivery Automation solution, decommissioning virtual servers can be done in a standardized and automated way.

Automate the Decommissioning of Virtual Servers

Invoke AppViewX's Prepackaged workflow solution or customize it accordingly to automate the decommissioning of the virtual IPs identified either individually or in bulk. The unused or free IPs are then moved to IPAM systems.

- AppViewX generates the list of unused objects, Disable and Decommission them automatically based on approvals resulting in,
  - More efficient allocation of IP resources. Unused IPs can be used elsewhere
  - Reduced service costs
  - The elimination of unnecessary management and monitoring overhead

Click **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**. Search for the workflow Get Unused LTM VIP and Delete, Get Unused LTM VIP and Notify, Disable F5 LTM VIP, or Disable and Delete F5 LTM VIP.



Refer to ADC+ Automation Workflow Guide for more Details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

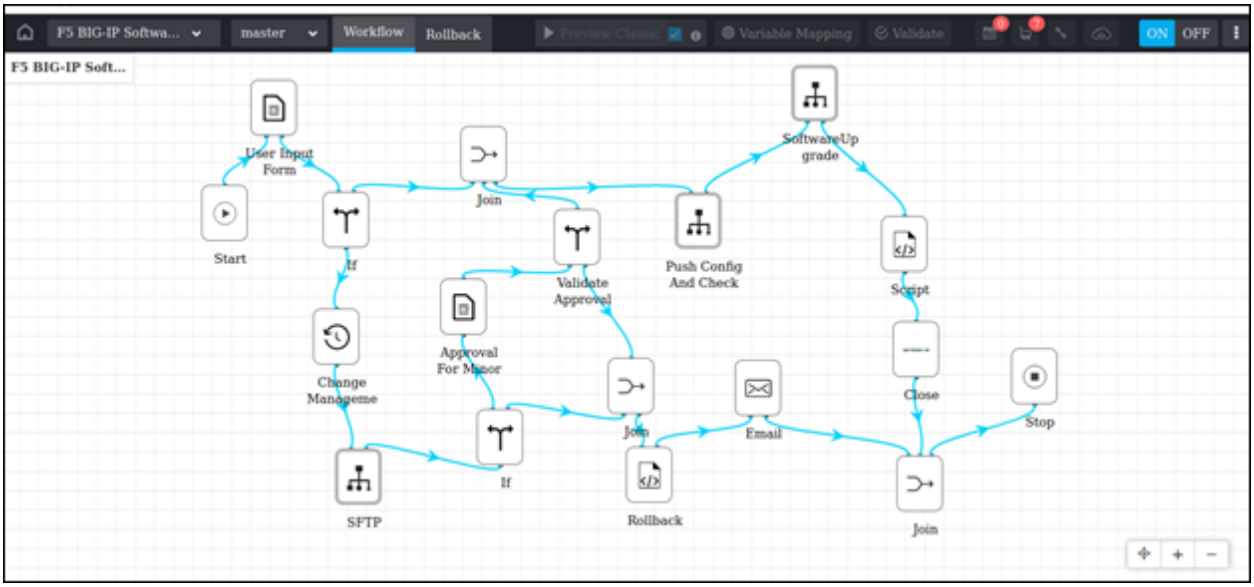
Upgrade of Load Balancers

AppViewX simplifies patch upgrades and version upgrades to the latest F5 version with minimum application downtime. Using AppViewX, you can migrate configurations across different platforms, such as BIG-IP virtual editions and BIG-IP hardware, including the new iSeries platform. Migration through AppViewX ensures clean installation and eliminates errors.

Upgrading F5 BIG-IP devices to their latest versions is a complex, multi-step process involving pre-migration validation checks, UCS back-ups, storing and managing UCS files, updating configurations, and post-migration validation checks. With the phase-out of F5 BIG-IP v10, AppViewX is offering a way to seamlessly transition to F5's latest platforms. By enabling end-to-end automation, organizations can now take advantage of the new features software enhancements can bring, without undergoing a tedious, error-prone migration process.

Click **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run** > **F5 BIG-IP System** > **System**. Invoke the workflows **Software upgrade on HA BIG-IP devices** or **Software upgrade** on Standalone BIG-IP devices.

The workflow will do necessary Pre checks, Transfer the required software image to the end device, Install the Image, Reboot & Reactivate the license and perform the post validation checks to ensure device configuration is not disturbed. If required, you can customize the steps with a simple drag and drop option and create your own custom flow or add additional approvals.



Refer to Automation Guide for more details.

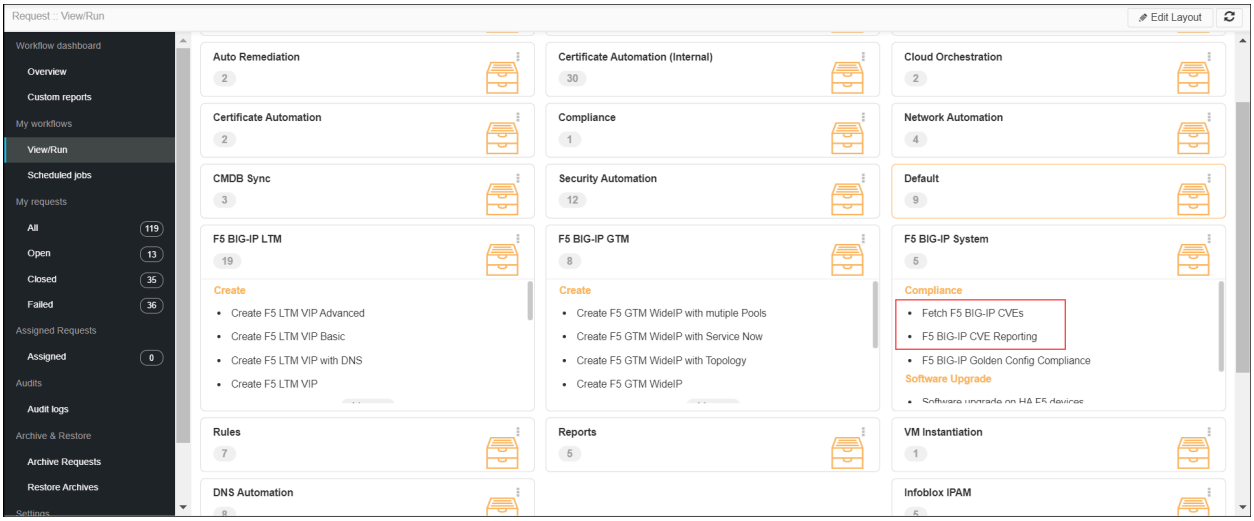
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

CVE Reporting

The vulnerabilities are informed in the form of reports known as Common Vulnerabilities and Exposures (CVE). At AppViewX, we have designed industry-leading methods where the entire process of validating the CVEs and upgrading an ADC to patch the vulnerabilities is automated from end to end.

You can manage and automate your ADC upgrades with simple task flows which can also be customized as per your requirements. The methods employed by AppViewX are verified, reviewed, and are updated every time we come across something better. Our product allows you to get timely updates from official sources on the latest vulnerabilities that can be obtained whenever you want them. We deliver better than the age-old method of manually checking for vulnerabilities for every single device that you use with just one workflow that brings all that information right to you with ease. To keep track of insecure devices, the discovered vulnerabilities are validated against the ADCs handled in AppViewX, and timely/intuitive reports with severity and suggestions are provided. Post this, upgrades are made to the recommended path version using AppViewX's seamless upgrade flows to resolve the discovered vulnerabilities.

To access the prepackaged workflows, click **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run** The workflows are grouped as per the vendors, select the required flow and invoke the required flow to deploy/manage an application.



Orchestrate Changes in App Delivery and Enforce Deployment Standards

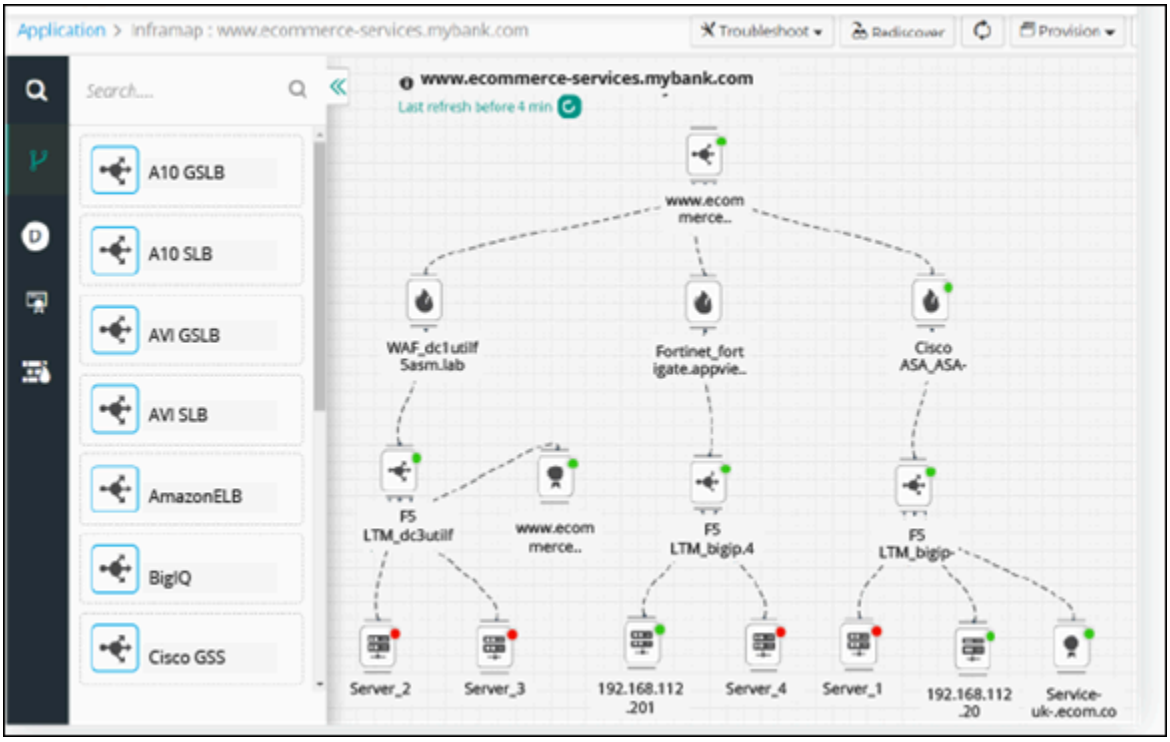
Orchestrate Application Delivery in Multi-Cloud Environments

AppViewX accurately abstracts hardware, software, open-source, and cloud solutions from end-users requesting application delivery services and can be used by business processes to enforce how compliance is achieved in multi-cloud data centers. It can be efficiently used to orchestrate how GSLB,

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

DNS, load-balancing, firewall, WAF, certificate, network, ITSM, and notification services are leveraged in the multi-cloud infrastructure. Through the pre-packaged solutions, enforce deployment standards defined by an organization and orchestrate changes without compromising on security.

Build your own end-to-end orchestration services through the pre-defined solution components available in the Workflow module. Click **Menu** > **Studio** > **Workflow** > **Create Flow**. Drag and drop the components as needed to build your own service.



Refer to ADC+ Automation Guide for more detail.

Orchestrate Application Delivery with DevOps Integration Tools

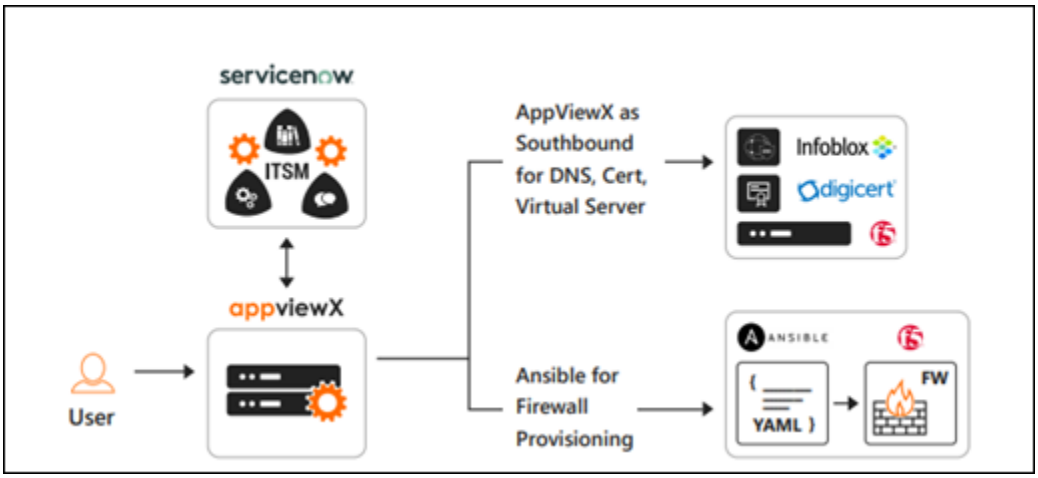
AppViewX exposes the network infrastructure services to DevOps tools like Ansible, Terraform, Chef, and Puppet. It integrates with the tools to build repeatable and compliant infrastructure with workflows that map to your business process. Enable end-users to self-service and launch automation from application-centric views and Automate the ticketing and governance of DevOps service requests with ITSM tools.

Advantages of using DevOps tools with AppViewX:

- Highly scalable and adaptive
- Zero learning curve
- Define once, deploy everywhere

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- DevSecOps-friendly
- Application-centric visibility
- Context-aware troubleshooting and auto-remediation
- Centralized management and extensive integrations
- Comprehensive network and application security management



For more information, click here.

Connecting AppViewX with Ansible

AppViewX can be run on the same control node where Ansible is installed. AppViewX can function as both Northbound and Southbound for Ansible.

AppViewX as the automation controller; Ansible as Southbound:

- You can import Ansible playbooks into AppViewX and call the appropriate ones to construct workflows in the Visual Workflow builder. You can also view and edit the playbook contents from inside AppViewX.

Examples:

- Application provisioning across F5 BIG-IP LTM, AFM using Ansible playbook.
- Instantiate a new server on AWS using AppViewX and Ansible

AppViewX as the orchestration controller; Ansible as Northbound:

- You can download an Ansible module file and a sample YAML file for the required AppViewX workflow on demand. Upload the Ansible module files onto the Ansible Server, provide input using the sample YAML file and execute the module file (playbook) on the Ansible server.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Examples:

- Application provisioning across F5 BIG-IP LTM and AFM
- Automating server instantiation and pool member addition using Ansible

Refer to Automation Guide for more detail.

Orchestrate with Ansible Integration

AppViewX integration with Ansible playbooks makes end-to-end network automation more intuitive, flexible, and user-driven.

- Integrated automation approach to aligning and achieving business goals
- Implement automation with AppViewX and Ansible for continuous delivery

Ansible - Ansible can be used to automate specific instances in the network, like configuring switches ad-hoc and opening firewall ports. Ansible's CLI-based automation has the capability to drill deep into the network, and since it's agentless, can be used on any device from any vendor.

AppViewX - AppViewX is an end-to-end low-code network orchestration and security solution. It handles high-level network operations like real-time monitoring and reporting, key/certificate provisioning and management, context-aware troubleshooting and auto-remediation, and self-servicing capabilities. While Ansible (Core) is CLI-based, AppViewX is GUI-based. It uses REST API to connect with third-party tools and cloud platforms

Self-Servicing through Custom Workflows

Increase the operational efficiencies of your organizations by enabling self-servicing through AppViewX's custom workflow capabilities. The workflow-centric automation leverages various components of the network infrastructure and helps in defining a logical flow of activities or tasks needed to deliver a service from start to finish.

The workflow studio is a repository that contains all the pre-packaged and custom workflow solutions. Manage your workflows from this studio and Create, Delete, Edit, Enable, Disable, or Clone the workflows as needed.

Design Custom Workflows

Automate your network infrastructure business process and enable Self Servicing for Application teams through the creation of custom workflows.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

AppViewX's Visual Workflow tool provides an intuitive system for designing self-serviceable, event-driven, intelligent, and automation workflows. It enables operational agility and service efficiency in the various stages of network and application service delivery. It helps to accelerate time-to-value by providing everything needed for the complete automation of application delivery network infrastructure.

Click **Menu** > **Studio** > **Workflow**. Start to design your custom workflows. Following are some of the capabilities available in the workflow studio.

- Design workflows using the library of tasks supporting multi-vendor ecosystem on F5, AVI, A10, Citrix - Solutions, and Libraries
- Intelligent work order execution - execute tasks sequentially or in parallel
- Build custom Email tasks in each stage with required parameters
- Validate configuration changes using Pre and Post validation checks
- Configure Reviews/Approvals at each stage to execute a change
- Plug and Play integrations into ITSM systems - Service NOW, Jira, etc.,
- Plug and Play integrations into DDI systems - Infoblox, Bluecat, QIP, etc.,
- Design Rollback workflows in case of execution failure or on a need basis.

Key Advantage:

- Build custom, event-driven automation using pre-built tasks and workflows
- Integrate with ITSM tools for ticketing and governance or send an email and Slack messages
- Expose a catalog of service requests for network security management
- Enable self-service to automation workflows with user-friendly forms

Self Service App Catalog

AppViewX provides a one-stop portal with all the service offerings in the form of low code pages to enable self-servicing for the end-users (like App owners, Network Engineers, CA, Admin, etc.) in a simplified and intuitive manner. There are Out of the Box catalogs and as well option to customize catalogs as per the organization standards. Access the opinion through **Menu** > **Studio** > **Pages**.

Benefits

- An intuitive self-service portal to access all service offerings for employees, and partners via Single Pane of Glass.
- Customer Experience – Users only see services that are available to them, described in business terms they understand.
- Build your own App catalog/portal.

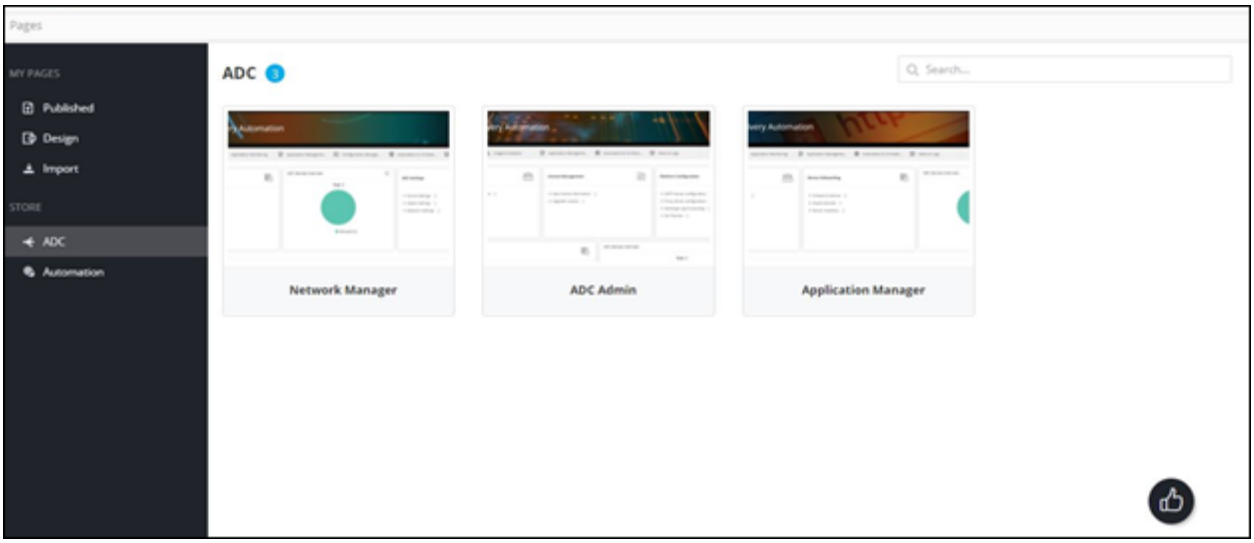
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Configurable Catalog page that allows for personalized task views.
- Low Code designer portal – Low Code and configurable toolsets, allowing users to brand Catalog solution so it feels familiar and intuitive for the end-users.

Self Service Catalog for ADC Roles

AppViewX provides out of the box self-service catalog page for the ADC Roles based on the permissions assigned. Refer Role for more details about the out of the box roles.

Access the catalog through, **Menu** > **Studio** > **Pages** > **ADC store**.



The catalogs comprise all following relevant functions and the life cycle of Application delivery presented in the form of a journey thus enabling a seamless self-servicing experience to the end-users. The catalogs could be cloned to achieve any sort of customization.

Design Your Own App Catalog

On clicking design, you will have options to design a custom app catalog. Define a name and description for the catalog. Drag and Drop some of the following components to build a catalog,

- Custom Branding
- Drag and Drop Reports/Workflows
- Enable Quick links to the solutions
- Configure External links
- Embed Page/Logo/Widget
- Design custom section using HTML editor palette
- Tab menu component for Horizontal, Vertical tab menus

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Publish and Share the Catalogs to End-users

Assign/Share the custom catalog page/workspace to specific user groups, users. The users with the shared pages can start accessing the functions available in the pages based on their RBAC.

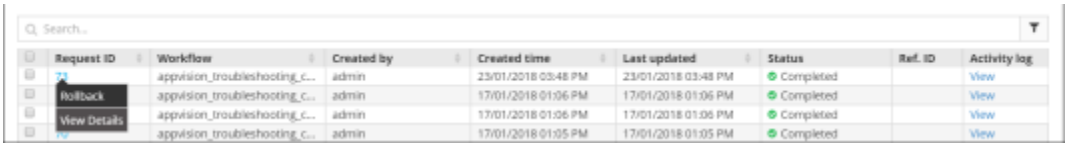
The catalog page could also be set as a landing/home page to enable a simplified journey to your end-users.

Refer to Service Catalog Guide for more detail.

Workflow Request

You can group your workflows accordingly and create a workflow catalog. The list of grouped workflows in the catalog will be displayed in this section.

- The **Workflow Request** section displays the total number of **All**, **Open**, **Closed**, and **Failed** requests, clicking upon these tabs displays the corresponding workflow details in a table.
- You can also perform the following tasks:
  - Right-click the **Request ID** and select one of the following options from the pop up that appears:
    - **Rollback** - Triggers an alternative rollback workflow that you have added or imported for the workflow.
    - **View details** - Displays all the tooltips configured in the Global variables.

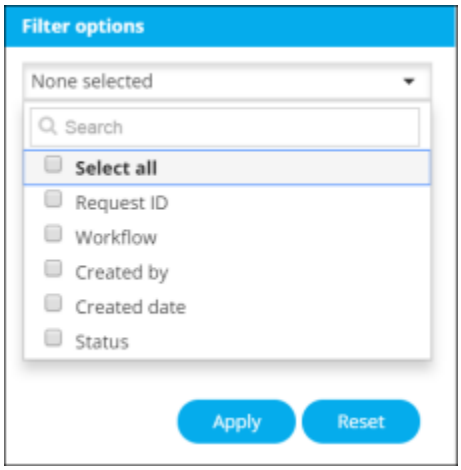


  - From the **Request ID** column, you can click on the request for which you want to view the details. On the tabs in this screen, you can perform the following tasks:

    For the workflow(s) containing grid component(s) and script(s) to generate data, you can click the (**Download**) button to save the data to PC in XLS or CSV file format.
  - Search for a request using the **Search** field.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Click the (**Filter**) button to select the options you want to use to sort the requests.



Alerts & Logs

Alerts

Alerts are a construct intended to inform administrators of significant events or complications that occur within the application. AppViewX provisions Proactive Application Service Monitoring & Alerting for quick remediation.

Alerts could be raised based on various metrics. In AppViewX, alerts are Device and Application based and each has its severity level. Severity levels are as follows:

- **Critical** - For issues that are causing disastrous results or impacts on functionality. These are top priorities and must be resolved immediately.
- **Fatal** - For issues that can cause disastrous results or impacts on functionality. These are a major priority and should be resolved soon.
- **Major** - For issues that are important and require a resolution, but that is not the highest priority.
- **Minor** - For issues that are of low priority and need a resolution.
- **Notification** - For issues that are not alerts or warnings, but which must eventually be addressed.

AppViewX receives alerts only for devices that have been subscribed to AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Types of Alert

The types of thee alerts are as follows:

- **Application Alert** - ADC specific Application actions executed from AppViewX (Control Center or Dashboards) will be raised as an alert captured in the alerts inventory. These alerts are focused on the behavior and performance of applications being delivered through the ADC.

  ADC Application alerts include:

Event Type	Description	Scenario	Message
Statistics	The statistics data collection to monitor the historic statistics for object level.	device statistics failure	Statistics collection for the device <"device name"> configured in alerts/alert <"alert names"> failed due to <"error reason">.
Restore/Rollback	Restore or rollback of an object configuration to a previous state.	Object Restore	Successful : Object <object name> restore action successful for the workorder <workOrderId>
Dashboard import	Status of the Dashboard import.	Dashboard import	<ObjectsName> association to <dashboardName> failed, as they are not associated to <deviceName>
Any action	Any action performed at the application level. The actions include: • Enable • Disable	GTM sync setting (gtm flush)	Action Enable executed successfully on object <ObjectDisplayname> by the user admin. GTM sync setting is enabled and hence the action is executed on the Master device <device name>.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Event Type	Description	Scenario	Message
			The actual triggered Peer device is <device name>

For configuring Application alert notification, see Configuring an Application Alert Notification.

- **Device Alert** - ADC specific Device actions executed from AppViewX (Control Center or Dashboards) will be raised as an alert captured in the alerts inventory. The administrator receives notifications for events performed at the device level.

  ADC Device alerts include:

Event Type	Description	Scenario	Message
Status change	Device status changes between any of the following: • Managed • Unmanaged • Failed • Unresolved Other status changes of the device are also captured under this event type. For example, modification of the Device status from x to y.	Device status change (Managed ->Unmanaged, Failed, Unresolved)	Device status modified from Manged to "Unmanaged/ Failed / Unresolved" on device:<device name>
Auto-detect	Automatic detection of the secondary Device status.	Device auto-detect failure	Secondary device auto detect failed for the device <device name> due to <error message>

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Event Type	Description	Scenario	Message
State change	Success/failure message for the Device state.	Device state flip	Device flip action was sucessfull on device < device name>. Modified properties{} via syslog
Device import	Success/failure message for the Device import.	Device import failure	Device Import from the file <fileName> failed due to <error message/ invalid column headers>
DNS_lookup	A scheduled reverse lookup for the objects that contain an IP address in it and persists the DNS name of the objects.	DNS lookup	DNS Reverse lookup failed for device <deviceName> due to <reason>
Any Action	Any action performed at device level. The actions include:	SSL Handshake failure (key exchange)	Communication failed for device <deviceName> due to <SSL handshake error>
Backup	Creation of the device backup.	Device backup success	Backup/Scheduled Backup has been generated successfully for the device <deviceName>
Restore/Rollback	Device-level to overwrite the complete device config.	Device Rollback	Successful : Device <device name> rollback action successful for the workorder <workOrderId>
Ihealth	iHealth report generation for the device.	Ihealth generation success	IHealth Report for <deviceName> is generated successfully

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Event Type	Description	Scenario	Message
Statistics	The statistics data collection to monitor the historic statistics for devices.	Device statistcs failure	Action <Updating live statistics> on <device Name> failed due to error <errorMessage>

For configuring Device alert notification, see Configure a Device Alert Notification.

- **Threshold Alert** - notification or warning generated by an ADC when a statistic threshold of application/ device. threshold or limit is exceeded. These alerts are designed to inform administrators about certain conditions that may require attention or action. This alert helps the administrator to take proactive measures to monitor and prevent outages or application issues. The administrator receives alert notifications based on the configured settings in the Threshold alert settings.

  Threshold alerts include:

Event Type	Description	Scenario	Message
Statistics	The conditions that will generate an alert notification.	Threshold Statistics collection	Threshold level reached for the object <object name>. <Statistics name – current value> is <operator> the threshold <configured value> AND < Statistics name – current value > is <operator > the threshold <configured value>.

For configuring Threshold alert notification, see Configuring Threshold Alert Notification.

- **Syslog Alert** - a notification or warning generated by an ADC that is sent as a log message to a Syslog server. These alerts are designed to provide information about specific events or conditions related to the ADC's operation, performance, or security. Any changes occurring in the device can be detected and patterned through Syslog alerts.

  Some examples of Syslog alerts include:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Syslog based updates for Object state change eg: Enabled, Disabled etc.,
  - Real time updates are supported only for F5. The other vendor object state will be updated at an interval of 15 mins.
- Syslog based updates for Object status change eg: Available, Offline etc.,
  - Real time updates are supported only for F5. The other vendor object status will be updated at an interval of 15 mins.
- Syslog based updates for Device failover state within 5 to 10 mins (Active, Standby) -
  - Applicable only for F5 and Citrix devices.
- Syslog based updates for Creation/Modification/Deletion of Primary objects eg: WideIP, Virtual Server, etc., and Secondary objects eg: Monitor, Profile, etc., at an interval of 15 mins in light weight config sync (Grok filters not restricted to particular object type but configurable)
  - When there are configuration updates on the objects like a new member added/removed/modified, it will be automatically reflected in AppViewX database with respective object mapping.
  - If those hierarchy objects are configured in ACL/Dashboard, the permission and actions will be automatically assigned respectively.

For configuring Syslog alert notification, see Configuring Syslog Alert Notification.

Alerts Inventory

Alert inventory lists ADC alerts that are available within AppViewX. It provides an overview of all the different types of alerts that can be generated or triggered, along with their corresponding details and configurations. The alert inventory helps in effectively managing and monitoring various events, incidents, or issues by providing a centralized reference for understanding and handling alerts. It allows you to have a clear understanding of the alerts they can expect and facilitates efficient alert management and response.

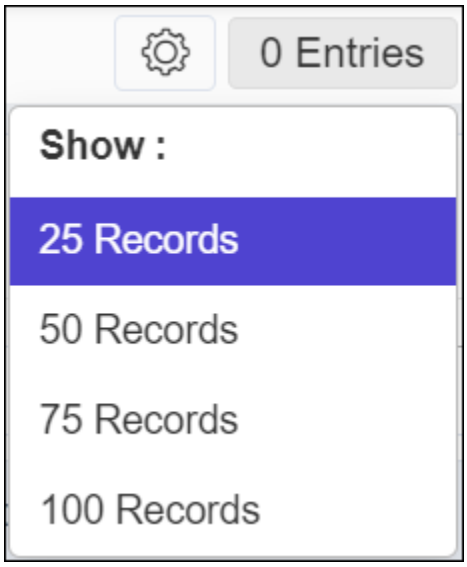
For the ADC Alert Inventory, go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts** > **ADC** tab.

On the Alert page, you can:

- Search for alerts.

  set the rows displayed on the page by hovering the mouse over the **Entries**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- Filter the alerts based on the date range by clicking  and/or column name by clicking .

The Alert inventory page consists the following columns:

Column Name	Description
Timestamp	Date and time when the alert generation.
ID	Unique identification of the alert.
Event type	Type of event for which the alert notification is configured refers to the specific category that triggers the alert.
Severity	The severity assigned to the alert. The severity levels are: • Critical • Fatal • Major • Minor • Notification
Category	Category of the alert notification. The categories are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Column Name	Description
	• Application • Device • Threshold • Syslog
Devices	IP address of the device for which the alert notification has been configured.
Applications	Name of the application.
Purpose/Usage	The message for which the alert notification is created for.
Alert detail	The message for the alert notification typically includes essential information about the triggered event. It provides enough details to help administrators quickly understand the nature of the alert. Sample message: `Device status modified from x to y on device <device name>`

Email/SNMP Alert Notifications

AppViewX supports notifications via Email and SNMP based on the configuration to improve the chances of remedial action.

To get notified via Email, **SMTP** must be configured. To configure For SMTP configuration, refer to Platform User Guide. You can provide multiple Email addresses with Subject lines to receive the alert.



To get notified via SNMP (Simple Network Management Protocol), following information are required:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Destination IP and Port
- SNMP version (v1 or v2)
- Community String - The string is similar to a user ID or password

Configure an Alert from AppViewX

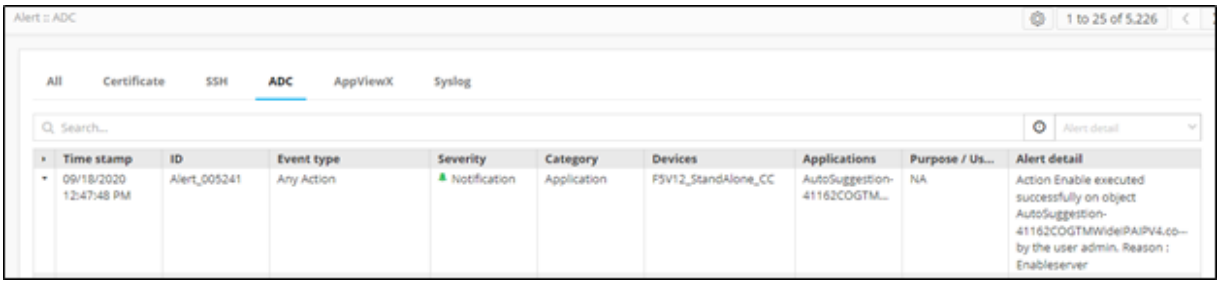
Configuring an Application Alert Notification

When ADC specific Application actions are executed from AppViewX, whether through Control Center or Dashboards, any resulting events or changes will be raised as alerts and captured in the alerts inventory. These alerts serve as notifications of the actions taken and their outcomes, allowing administrators to monitor and respond to the Application's behavior effectively.

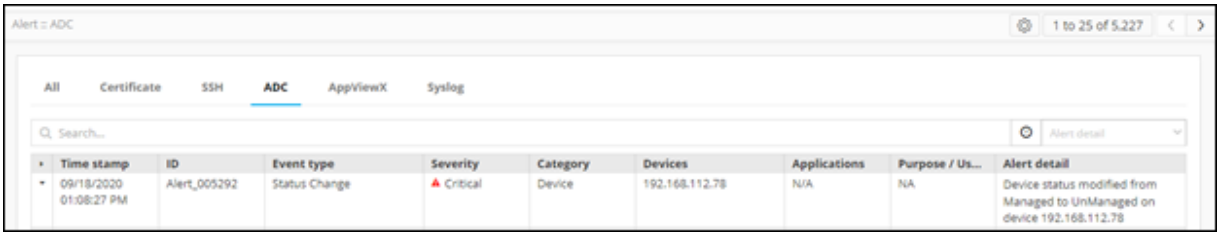
To get the application alert notification, administrator has to configure as follows:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts** > **ADC** tab.
   All the successful and failure events are captured in the alerts inventory along with the user comments. This enabled an Admin to track end-user actions.
2.
3. Every alert contains the following information:
   - Alert ID
   - Alert Event type
   - Severity
   - Category
   - Devices
   - Applications
   - Alert Detail
4. Search for any alert by the Application, Device name, or message etc.
   All successful actions are tagged as Notification and Failure as Critical alerts respectively.
5. Event type helps to identify the type of event - An action event, restore event, or backup event etc.,
   An example of Object Enable Alert:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



An example of Device status change alert (Managed to Unmananged):



Configure a Device Alert Notification

When ADC specific device actions are executed from AppViewX, whether through Control Center or Dashboards, any resulting events or changes will be raised as alerts and captured in the alerts inventory. These alerts serve as notifications of the actions taken and their outcomes, allowing administrators to monitor and respond to the Device's behavior effectively.

To get the device alert notification, administrator has to configure as follows:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts** > **ADC** tab.
   All the successful and failure events are captured in the alerts inventory along with the user comments. This enabled an Admin to track end-user actions.
2. Every alert contains the following information:

   • Alert ID

   • Alert Event type

   • Severity

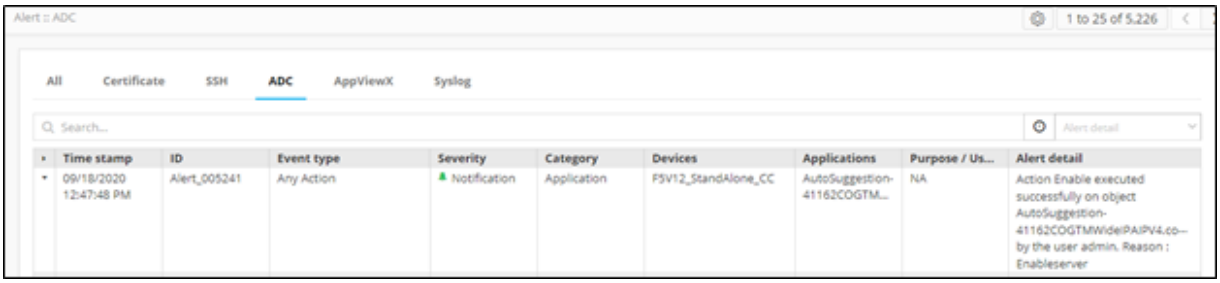
   • Category

   • Devices

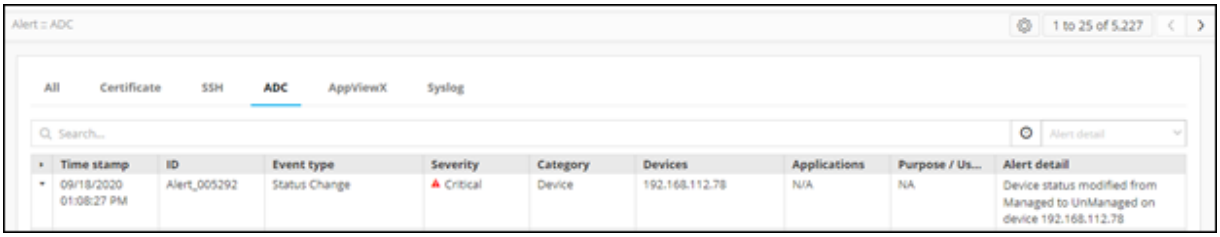
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Applications
- Alert Detail

3. Search for any alert by the Device name or message etc.
   All Successful actions are tagged as Notification and Failure as Critical alerts respectively.
4. Event type helps to identify the type of event - An action event or Restore event or Backup event etc.
   An example of Object Enable Alert:



   An example of Device status change alert (Managed to Unmananged):



Configuring Threshold Alert Notification

AppViewX allows near real-time monitoring of Critical Applications/Devices that breaches statistical thresholds like CPU, Memory, Number of requests, etc. Customize the threshold limits as per your organization, monitor Apps on real-time, and get notified via Email or SNMP.

To configure a Threshold alert notification,

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
2. Click (**Settings**) icon, and then ADC.
3. Enter an **Alert Name**, **Message**, and **Category** as Threshold.
4. Configure the critical Devices/Applications that need to be monitored.
5. Set an **Alert monitoring interval** from 10 secs to 60 secs that fetches the run time data from the device to validate against the threshold.
6. Set a Cool off period like 10, 20, or 30 mins. The cool-off period prevents the user from getting bombarded with notification alerts for the same event.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. Configure the Statistics that need to be monitored and set a value ie >=,<=,<,>,==.
8. Create an Alert with multiple conditions with Logic AND and OR by clicking .
9. Define a Severity for your event based on the threshold conditions (Critical, Fatal, Major or Minor).
10. Configure the notification type as Email or SNMP.
11. Configure multiple Alerts as needed and Add them to the Grid. The configured Alerts could be modified or deleted anytime by selecting the Alert from the grid.
12. Following are some of the alerts that can be configured:

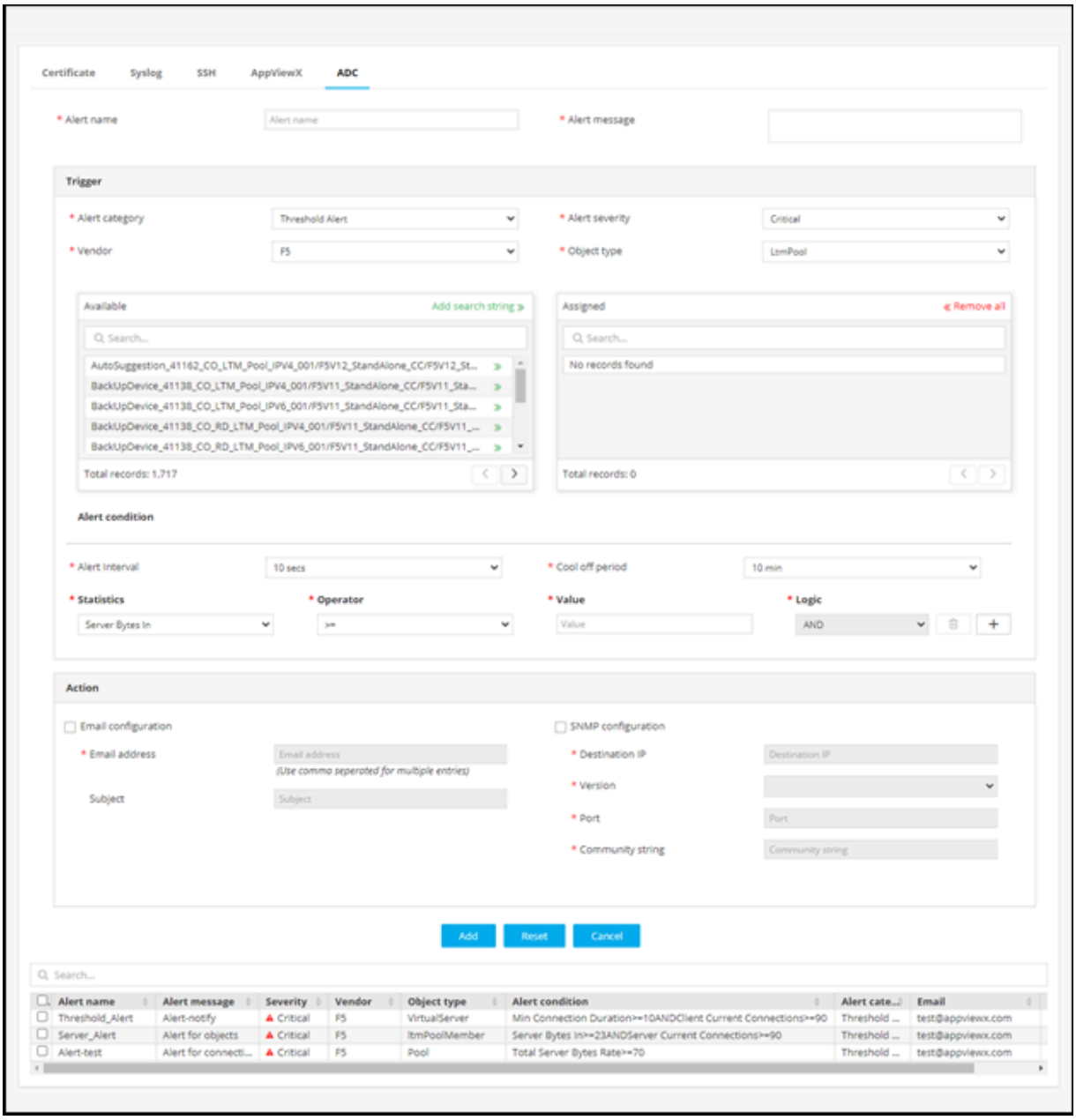
    • A Virtual server’s Current connection exceeded 100

    • A critical Device’s CPU breaches 60

    • A server’s Number of Requests is lesser than expected.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

13. The configured events will be notified to the end user via Email or SNMP and as well captured in the Alerts inventory for tracking.



Configuring Syslog Alert Notification

AppViewX subscribes to all device-level logs, where it acts as a Syslog listener. The logs of any devices added in AppViewX can be viewed as Syslog by navigating to **Logging** > **Syslog**. However, devices tend to generate huge amounts of data, a Syslog alert is a convenient way to get notified about

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

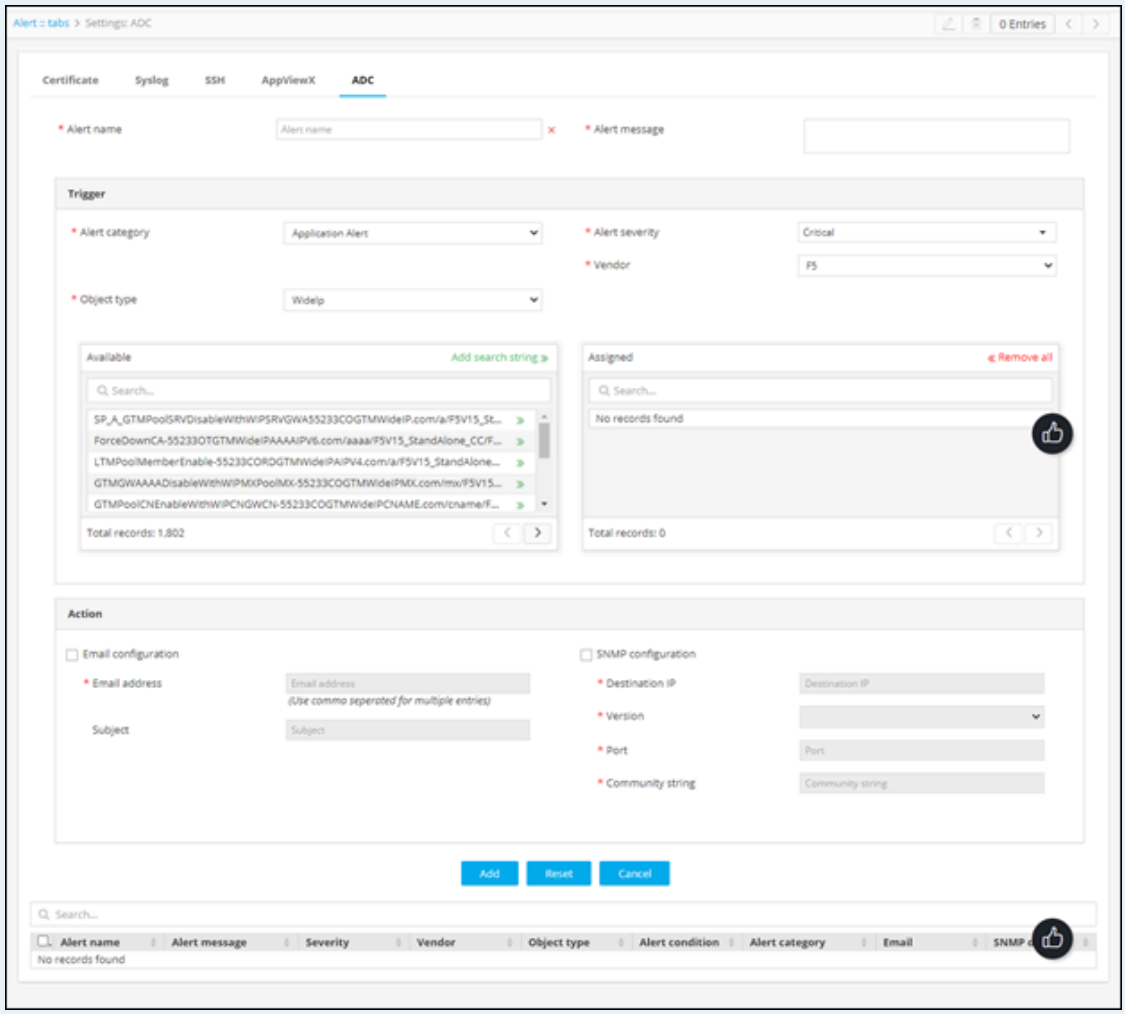
a specific event that is of importance to you. It also allows for a closed-loop remediation by associating workflows.

To configure a Syslog alert notification:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
2. Click  (**Settings**) icon, and then ADC.
3. Enter an Alert name and Alert message.
4. Select the **Alert Category** from the drop-down list. The options are **Application Alert** and **Device Alert**.
5. Select the **Alert Severity** from the drop-down list.
6. Configure the critical **Device/Applications** that need to be monitored.

> **Note:**
> Rather than adding devices manually, you can click the Add search string link and create a search string that automatically assigns all existing objects or devices that match the filter criteria. The benefit of using a search string rather than selecting devices manually is that the search string continues to work in the background and auto-assign all new devices if the devices match the search criteria you set up.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. Add the Pattern/Regex that needs to be monitored on the Syslog received. Multiple strings can be provided with comma-separated, which will be considered as Boolean AND operator.
8. Following are some of the alerts that can be configured,

   - **Sample syslog** - <133>Sep 19 04:24:38 bigip-40-152 notice mcpd[6046]: 01070417:5: AUDIT - user admin - transaction #84153993-4 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name \"/Common/testVs\" virtual_server_profile_profile_name \"/Common/tcp\" virtual_server_profile_profile_type 5 virtual_server_profile_profile_context 0 } } [Status=Command OK]\n

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- For instance, if the Syslog alert is configured for the object and Regex pattern is given as “create” Whenever an object is created and a Syslog is received for that object as above. An alert will be raised for the same and notified to the user.
- You can also pass certain metadata from the alert to the workflow. In the Metadata section, enter a key and its associated value in the respective fields. This is the additional information that will be used by the workflow that is going to be associated with.

9. Associate any out of the box or custom workflow that needs to be executed on the occurrence of a configured Syslog event.
10. Configure multiple Alerts as needed and Add it to the Grid. The configured Alerts could be modified or deleted anytime by selecting the Alert from the grid.

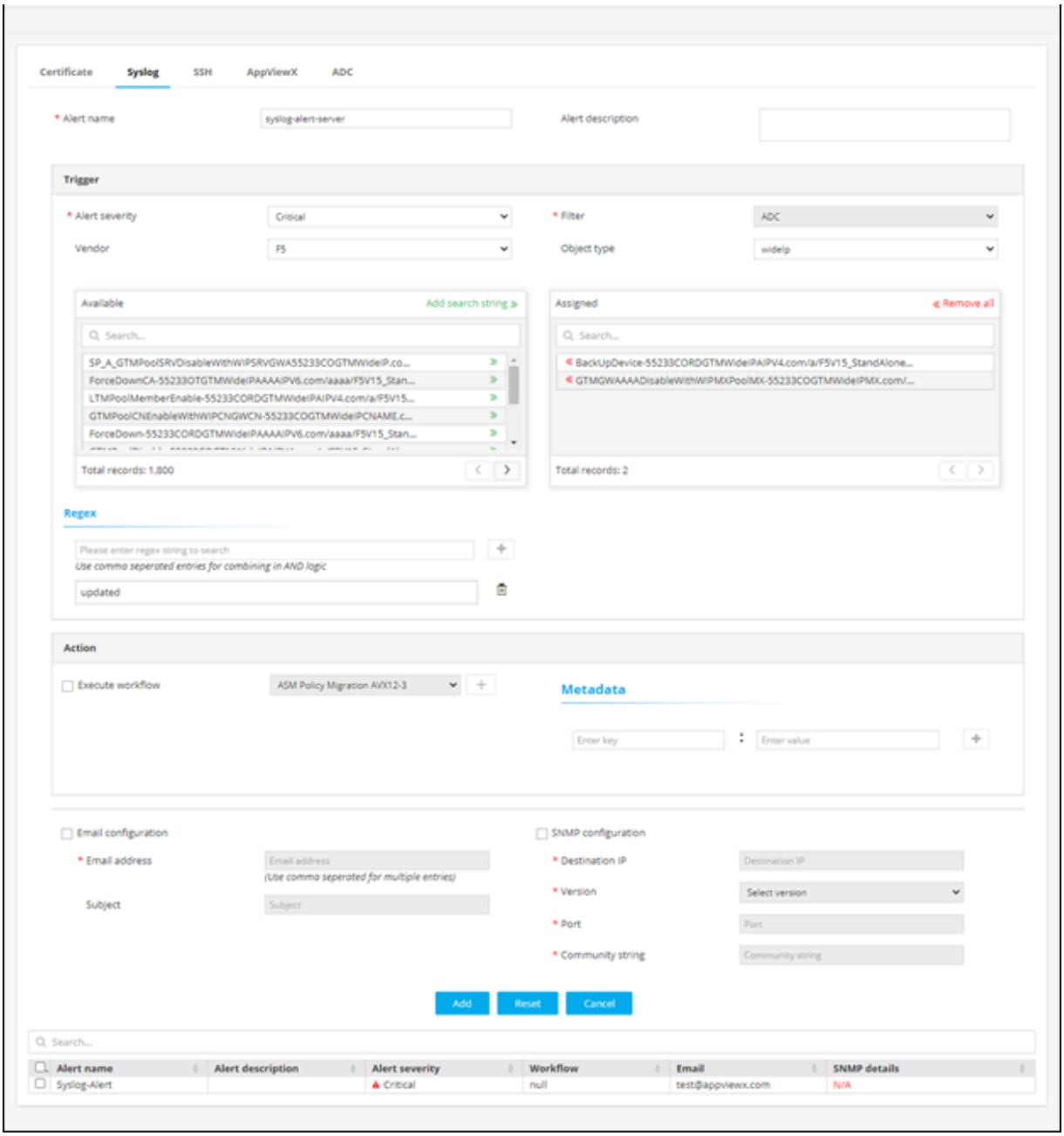
Monitor Critical Applications/Devices and Receive Notification

Alert settings can be configured to trigger Email/SNMP alerts when any action is executed on configured Application(s) or Device(s).

To configure Application/Device alerts:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
2. Click  (**Settings**) icon, and then ADC.
3. Enter an **Alert name** and **Alert message**.
4. Select the **Alert Category** from the drop-down list. The options are **Application Alert** and **Device Alert**.
5. Select the **Alert Severity** from the drop-down list.
6. Configure the critical Devices/Applications that need to be monitored.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. In the **Action** section, select the checkbox of Email or SNMP.
8. (Optional) Configure multiple Alerts as needed and add it to the grid. The configured alerts can be modified or deleted anytime by selecting the **Alert** from the grid.
9. Following are some of the alerts that are notified upon configuration:

   • A critical Application is Disabled or Forced down (Event - Any Action)

   • A critical Device is Restored to a different configuration (Event - Restore)

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Device Flipped from Active to Standby (Event - State change)
- A critical Device is Failed during a config fetch (Event - Status change)
- Device Backup failed (Event - Backup)

Logs

Logs keep track of all activities that take place within AppViewX or any external entity that is connected to the AppViewX system. The Logging functionality in AppViewX tracks user activities and creates the device and object-level event logs.

The Logging screen allows you to view logs, which are grouped into the following categories:

- **All** - All category logs are displayed in this section. The logs listed are based on the latest user actions.
- **Audit** - All user-related changes
- **Self-audit** - Logged in user changes
- **ADC** - ADC specific logs
- **AppViewX** - AppViewX infrastructure level logs
-  **Syslog** - End device level logs

 **Syslogs**

- AppViewX subscribes to all device-level logs, where it acts as a Syslog listener. AppViewX registers for Syslog notifications with the devices managed in AppViewX or with external servers (like Kafka) and receives notifications when a change is identified on the device. Based on the notifications, the AppViewX system shall automatically update the configurations based on a configured interval ensuring device changes are up-to-date.
- Syslogs received from the devices and the changes that are processed in AppViewX based on the Syslogs will be captured in the logging module.
- Following are the events handled by AppViewX:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Syslog-based updates for Object state change eg: Enabled, Disabled, etc.,
- Syslog-based updates for Object status change eg: Available, Offline, etc.,
- Syslog-based updates for Device failover state eg: Active, Standby, etc.,
- Syslog-based updates for Creation/Modification/Deletion of Primary objects eg: WideIP, Virtual Server, etc., and Secondary objects eg: Monitor, Profile, etc., at an interval of 15 mins.
  - When there are configuration updates on the objects like a new member added/removed/modified, it will be automatically reflected in the AppViewX database with respective object mapping.
  - If those hierarchy objects are configured in ACL/Dashboard, the permission and actions will be automatically assigned respectively.

- Refer to Install, Upgrade, and Maintenance Guide for more details and Syslog configuration in AppViewX.
- AppViewX can either handle logs directly from the device (Logstash IP registration directly on the device) or through external servers. If the user has enabled Kafka properties, the Logstash registration will be automatically removed from the device.
- AppViewX integrates with external log servers like KAFKA when the Syslogs are not allowed to be received from load balancers directly.
- Kafka is a cluster that runs on one or more servers used to store real-time logs for a device. Ensure that the external log server details are configured in AppViewX to fetch the device logs.
- Refer to Platform User Guide for more details and Kafka configuration in AppViewX.

Within the Logging screen, you can perform any of the following actions:

- View details of a log
- Configure logging for ADC object types
- Export logs

View Details of a Log

To view log details:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ALERTS & LOGS** > **Logs**.
   The **Logging** screen opens. All the logs are displayed by default.
2. Click the tab that corresponds to the type of log you want to view.
3. On the log screen, view all details for a log by hovering your cursor over each column of data or click the (**Expand**) icon for the log you want to view. The table row expands to display all details for the log.

> **Note:**
> Search for any logs from the logging list based on User, Device name, log message, object details, source IP, AppViewX node, Method of login also search based on a particular column.

Schedule Log Reports

In addition to providing scheduled logging activities, AppViewX allows you to set up logging tasks for specific ADC object types and have the log results emailed automatically to designated recipients.

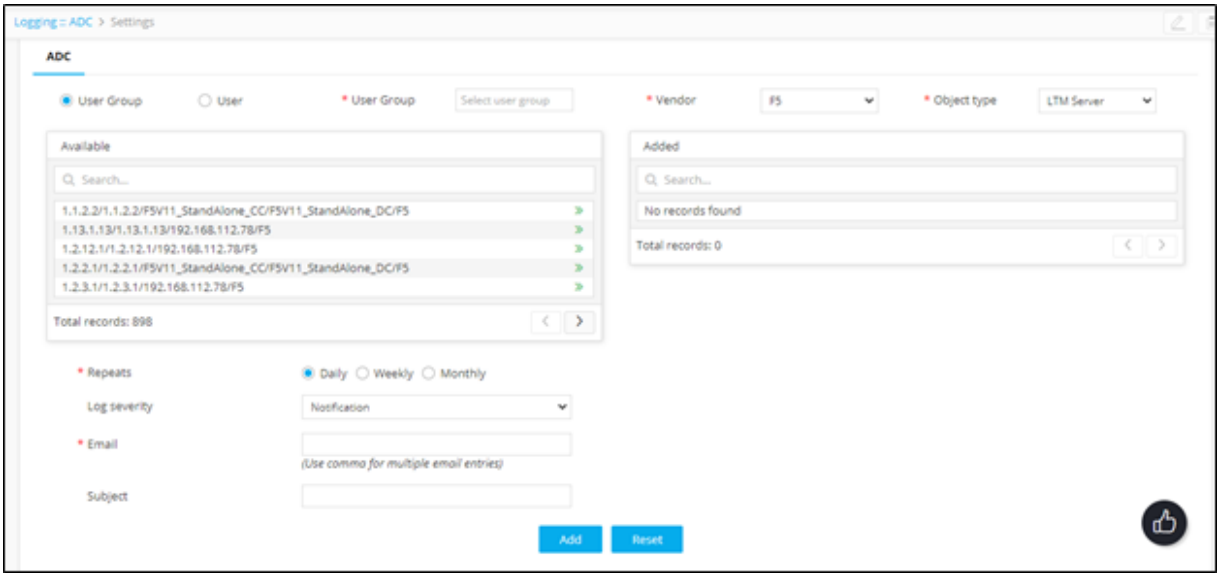
To schedule the log reports:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Logs**.
   The **Logging** screen opens.
2. Click the (**Settings**) button on the top.
3. At the top of the **Settings** screen, select the **User Group** or **User** radio button, and then enter the name of the user or group that the log report relates to.
4. In the **Vendor** field, select the vendor whose object or objects you want to generate a log report for.
   In the **Object Type** field, select the object that you want to generate a log report for.
5. In the **Available** field, click the » (Assign item) icon beside each object you want to include in the log report.
6. In the **Repeats** field, select how often you want the log report to be generated: Daily, Weekly, or Monthly.
   From the **Log severity** dropdown list, select one of the following:

   - Critical
   - Fatal
   - Major
   - Minor
   - Notification

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. This is to let the user know the severity of the log reports that have been sent.
8. In the **Email** field, enter the email addresses of users receiving the log report.
9. (Recommended) In the **Subject** field, enter a short, clear description of the log report so that the email recipients can tell at a glance what the message contains.
10. Click **Add** to add the log report generation task to the table at the bottom of the screen.



Forward a Log

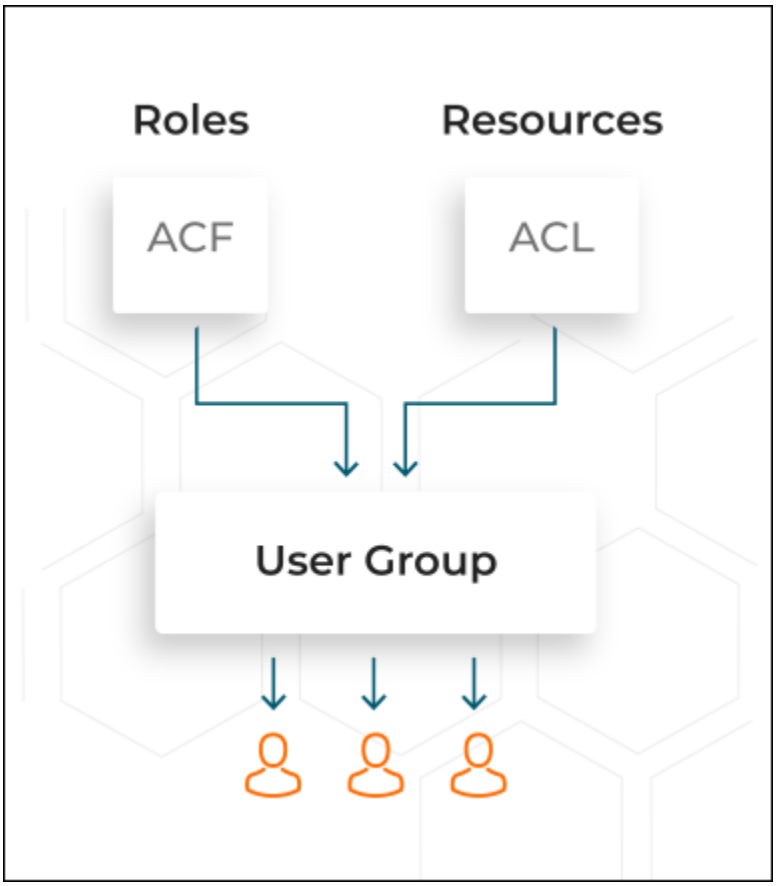
Log forwarding implies forwarding log data from our AppViewX log servers to external hosts like (splunk, elastic) to backup up the data or to process, reporting, monitoring in an external system. The user has to define which type of log data needs to be forwarded and in which format. Audit, AppViewX, and ADC logs can be forwarded. Supported protocol - TCP, UDP.

For more details and Kafka configuration in AppViewX, see Platform User Guide.

Account

AppViewX offers comprehensive support for Role and Resource-Based Access Control (RBAC). It allows you to integrate with the existing identity stores such as Active Directory (AD) and Lightweight Directory Access Protocol (LDAP) to enforce authorization policies. Roles and Resources can be customized to suit any organizational structure and user requirements.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Role-Based Access Control

Role-Based Access Control (RBAC) is key to eliminating working in silos. AppViewX advanced ADC automation and orchestration platform customizes roles at a very granular level, assigns authorized functions to a specific role, and specifies device and application access to users.

Once you have deployed the ADC in your infrastructure, you have to onboard users with different levels of permissions to control the operations. ADC helps you with a completely customizable Account module to create users, roles, user groups, and resources. Also, it provides the default recommended roles and permissions as per industry standards that can be cloned and customized as required.

Accessing the Quick Config Option

To configure RBAC using Quick Config option,

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Click **Menu** > **Settings** > **General** > **Authentication** > **Quick Config** option.
- The Authentication stage opens in a wizard flow with the LDAP sub-tab displayed by default. On the same screen as part of the wizard flow, user groups, roles, and resources stages are displayed at the top. Click on the respective stage for configuration.

Ways to Access Quick Config Wizard Flow

- Click **Menu** > **Account** > **User group** > **Quick Config** option.

(or)

- Click **Account** > **Role** > **Quick Config** option

(or)

- Click **Account** > **Resource** > **Quick Config** option.

The Authentication stage opens in a wizard flow with the LDAP sub-tab displayed by default. On the same screen as part of the wizard flow, user groups, roles, and resources stages are displayed at the top. Click on the respective stage for configuration.

For detailed instructions to perform the above-mentioned actions, see Platform User Guide.

To configure role-based access control, the user should have ACF permissions. With ACF permission, you can:

- Assign devices to resource
- Assign objects to resource

Authentication

The **Settings** > **General** tab enables you to configure and manage authentication for

1. **Lightweight Directory Access Protocol (LDAP)**: LDAP stands for Lightweight Directory Access Protocol. As the name suggests, it is a lightweight client-server protocol for accessing directory services, specifically X. 500-based directory services. LDAP runs over TCP/IP or other connection-oriented transfer services.
2. **LDAPS (LDAP over SSL)**: Also known as LDAP over TLS and LDAP over SSL, LDAPS allows for the encryption of LDAP data (which includes user credentials) in transit when a directory bind is being established, thereby protecting against credential theft.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. **TACACS**: TACAS is a CISCO proprietary system utilizing a transmission control protocol (TCP). In TACAS, the entire authentication packet is fully encrypted and allows us to set up our own CISCO independent servers and databases. The AppViewX system allows you to add more than one Terminal Access Controller Access-Control System (TACACS) server for authentication.
4. **RADIUS**: RADIUS, which stands for "Remote Authentication Dial-In User Service", is a network protocol that controls user network access via authentication and accounting. The AppViewX system allows you to add more than one Remote Authentication Dial-In User Service (RADIUS) server for authentication.
5. **SAML**: SAML stands for Security Assertion Markup Language. It is an XML-based open-standard for transferring identity data between two parties: an identity provider (IdP) and a service provider (SP). The AppViewX system allows you to add more than one Security Assertion Markup Language (SAML) server for authentication.

For detailed configuration, refer to Platform User Guide.

On the successful configuration of Authentication protocol in AppViewX, start importing user groups from AD or create manual user groups.

User

A user is an individual who has access to AppViewX using a unique username and password maintained internally or by an external enterprise server such as Active Directories (AD).

Note:

- Administrators can define how users should be authenticated to AppViewX. User authentication can either be an internal authentication or external authentication via LDAP, RADIUS, TACACS, and Single Sign-on.
- To create user accounts, you must be assigned the Administrator role.
- You must add a user to the user group as the roles and resources cannot be directly associated with the user.

In the User page, the following actions can be performed:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Create a User**: Create new users and associate them with user groups.
- **Modify a User**: Modify the existing user.
- **Delete a User**: Delete the existing user.
- **Enable a User**: Enable an existing user to activate the user.
- **Disable a User**: Disable the existing user to temporarily deactivate the login.
- **Import User**: Import the list of existing users. if needed the sample template file can be downloaded, which is in .csv format.

For detailed instructions to perform the above-mentioned actions, see Platform Userguide.

User Group

A user group is a group of individuals that have access to the same roles and resources. User Groups can be created manually or synced from an active directory or can be bulk uploaded using a spreadsheet.

When you associate a role and resource with a user group, the users within that user group are granted all of the role's and resource's corresponding privileges and permissions.

You can associate roles and resources only with user groups

In the User group page, the following actions can be performed:

- Create a User Group: Create a user group with basic information and associate Roles and Resources to it.
- Modify a User Group: Modify a user group by adding, removing, or modifying the Roles and Resources on demand. The associated users of the user group will be reflected with the changes made to the user group.
- Delete a User Group: Select the required user group and delete the same. You cannot delete a user group that has active users in it. Users belonging to the deleted user group cannot log into AppViewX.
- Clone a User Group: The clone user group option allows you to create an exact copy of an existing user group with a different name. You can modify the roles and resources association while cloning a user group.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Enable a User Group: Enable the required user group.
- Disable a User Group: Disable the required user group temporarily. You cannot disable a user group that has active users in it and users who are associated with a disabled user group cannot log in to AppViewX.

For detailed instructions to perform the above-mentioned actions, see Platform Userguide.

AppViewX encourages users to use the “quick config “option made available to import user groups seamlessly from AD. Following are the Action that can be performed on the User Group:

- Add New User GroupAdd New User Group by Sync Groups from LDAP option:

  1. Go to **Menu** > **Account** > **User Group** > **Quick Config** option.

     Authentication stage part of RBAC Configuration wizard flow displayed by default.

  2. Go to the **User group** stage as part of the RBAC Configuration wizard flow.
  3. Click the **Add new group** button.
  4. Select ‘Sync groups from LDAP’ option.
  5. Click **Proceed**.

- Add New User Group by TACACS/RADIUS/SAML/AppViewX option:

  1. Go to **Menu** > **Account** > **User Group** > **Quick Config** option.

     Authentication stage part of RBAC Configuration wizard flow displayed by default.

  2. Navigate to the **User group** stage as part of the RBAC Configuration wizard flow.
  3. Click the **Add new group** button.
  4. Select ‘TACACS/RADIUS/SAML/AppViewX’ option.

- Add **New User Group** by clicking the **Bulk Import** option:

1. Go to **Menu** > **Account** > **User Group** > **Quick Config** option.

   Authentication stage part of RBAC Configuration wizard flow displayed by default.

2. Go to the **User group** stage as part of the RBAC Configuration wizard flow.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Click **Add new group**.

4. Select the **Bulk Import** option

Role

Each role assigns a specific set of permissions relating to the modules that can be accessed and the tasks that can be performed in each AppViewX module. The roles can be assigned only to a user group. The user groups that are assigned with a role will automatically inherit all the associated permissions. User groups can be assigned with more than a role.

The Roles management of Inventory comprises some of the Out of the Box (OOB) roles available for ADC. The OOB roles can be cloned, enabled, and disabled. It can not be updated or deleted. Administrators can also create custom roles. Custom roles can be updated, deleted, enabled, and disabled. Users can either use OOB roles (if match their needs) or custom roles to map to user groups. The available roles are:

- **Application Manager ADC**: The application manager will have complete/partial (complete access will not be given to Shared devices) insight on all the application-related devices. The application manager gets “write” access to all the objects of application objects.
- **Auditor ADC**: Auditor access is predominantly for archived logs and the backed-up data along with the current reports and log data which would be used for all kinds of activity tracing.
- **DevOps ADC**: The user can automate, execute workflows, perform plugin management and upgrade.
- **Executive Director ADC**: Executive access is more towards business-level operational smoothness. The success rate of the changes and the monitoring data for the critical devices in report form ensure adequate details to have respective vendors communicate more data-oriented. This role would also help in understanding failure patterns.
- **Network Manager**: Access to complete infrastructure as the responsibility of access provisioning to users and the limited device access provisioning is also handled by this user only. It would require almost admin level of access on the infrastructure, would only not include core system-level access.
- **Traffic Manager**: The traffic manager can manipulate the state and status of any object/device to which he has been given access, there are possibilities that users may not have access to all the objects in the device as most devices are based on a shared network infrastructure basis.

In the **Roles** page, the following actions can be performed:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Create a Custom Role**: Create a custom role. Note that to assign functions at a more granular level, click the expand icon beside a function checkbox and then select individual sub-options within that function. In the image below, for example, you can select ADC, which automatically assigns all six sub-options and the sub-sub-options within them, or you can expand the ADC function and select only the sub-options or sub-sub-options you want to assign.
- **Modify a Role**: Modify a role in AppViewX.

> **Note:**
> The Out of the box role functions can not be edited. Only custom roles functions can be edited.

- **Delete a Role**: To delete a role from AppViewX.

> **Note:**
> You can not delete a role that has active users in it. Also Out of the box (OOB) roles cannot be deleted. Only custom roles can be deleted.

- **Clone a role**: The Clone a role option allows you to create an exact copy of an existing role with a different name. The user can modify the permissions and tasks that can be performed while cloning a role.
- **Enable a Role**: Enable a role in AppViewX.
- **Disable a Role**: Disable a role in AppViewX. Note that You cannot disable a role that has active users in it and The users associated with a disabled role through a user group will not be allowed to log in to AppViewX.
- **Role Mapping to User groups**: To map a role to user groups.

For detailed instructions to perform above-mentioned actions, see Platform Userguide.

Resources

All the devices and objects that are configured within AppViewX are termed as Resources. The resource allows you to specify access at a granular level across all the devices and modules of AppViewX listed in this section, where the permission definitions are independent of each other. The resources can be assigned only to a User group. The users that are assigned to the user groups will automatically inherit the permissions associated with that resource. User groups can be assigned with more than one resource.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

In the **Resources** page, the following actions can be performed:

- **Create Resources**: Create Resources with ADC devices and objects in R or RW permissions
- **Modify Resources**: Modify Resources by adding, removing, or modifying permissions on devices and objects.
- **Delete Resources**: Delete Resources. Users associated with the deleted resources will not be able to access the Application.

AppViewX can be configured to not only provide permission to specific devices in the inventory but also to specific objects (Virtual server, Pool, Servers) within the device. This helps you to create custom user roles according to their need limiting the access by making sure the unauthorized access to critical objects is limited.

AppViewX gives you the flexibility to create users with various permission levels to access devices and objects. One is **Read only (R)** where users are allowed to access devices or objects to only view the configuration without permission to modify it. The second level of permission is **Read & Write (RW)**, where users have the access to both views and modify the device or object configuration.

AppViewX encourages users to use the **quick config** option made available to streamline the user creation process and help users identify the true potential of RBAC.

Using the **Quick config** option, you can perform the following actions:

- Dynamic rule-based resource tagging of newly discovered ADC objects, Certificates based on Query using object/Certificate fields available within AppViewX.
- Dynamic rule-based resource tagging of newly discovered ADC objects, Certificates using a script to tag based on data available with external tools (SNOW, Other CMDB, etc.).
- Rule templates are pre-shipped to ease the rule creation to dynamically tag resources
- Dynamically created resources can be assigned to user groups dynamically by mapping the respective rule to the required user groups as part of the Rules in Use inventory in the wizard flow.
- Manage the order of execution for the RBAC rules.
- Manage short circuit options to dynamically tag ADC objects.

For assigning the devices or objects to resource, refer to the following sections:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

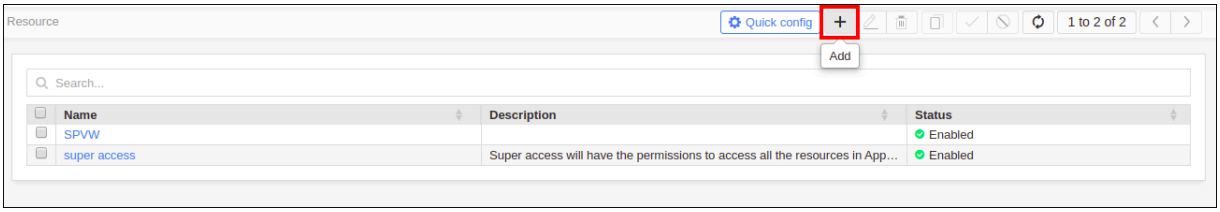
- Assign Devices to Resource
- Assign Objects to Resource

You can also check the following video link for more details:

https://www.appviewx.com/use-rbac-eliminate-working-silos/

Assign Devices to Resource

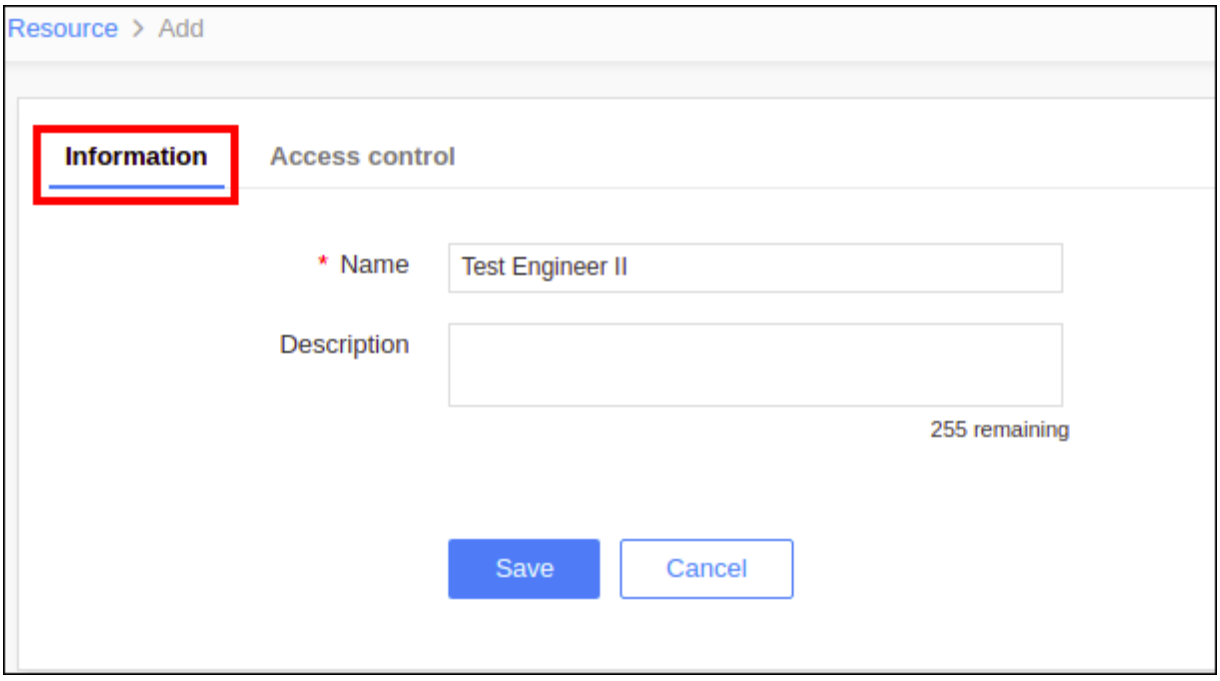
1. Go to **Menu** > **Account** > **Resource**.
2. Click on the  **(Add)** icon to navigate to the resource addition page.



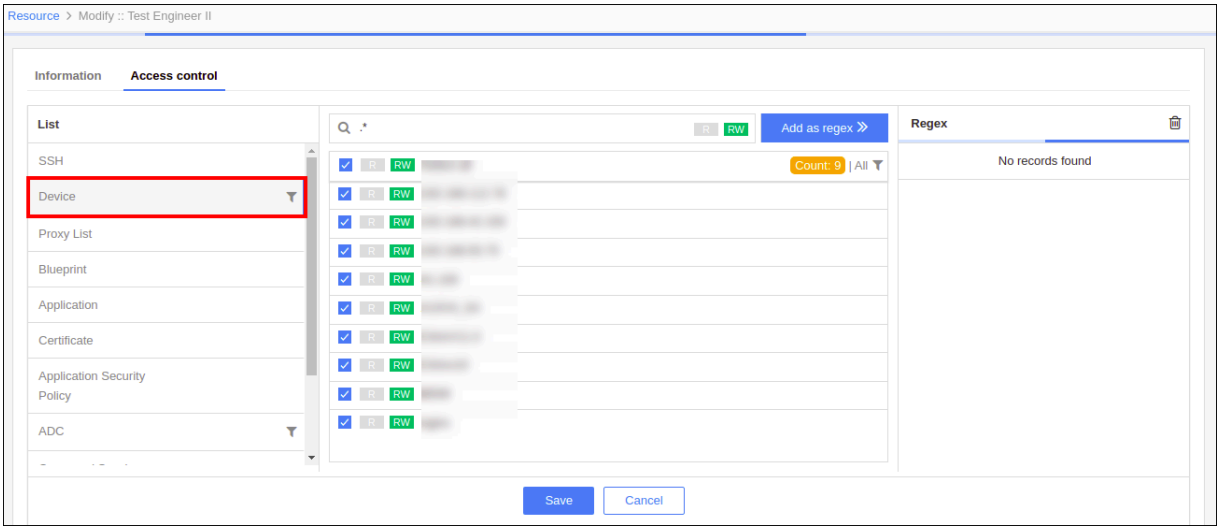
3. From the **Resource Add** page, configure the group details as given below and click **Save**.

Field	Type	Description	Validation
***Name**	Text	Name of the resource to be added.	Resource name cannot be empty.
Description	Text	Description of the resource.	A maximum of 255 alphanumeric characters and space are allowed.
**: Mandatory fields*			

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

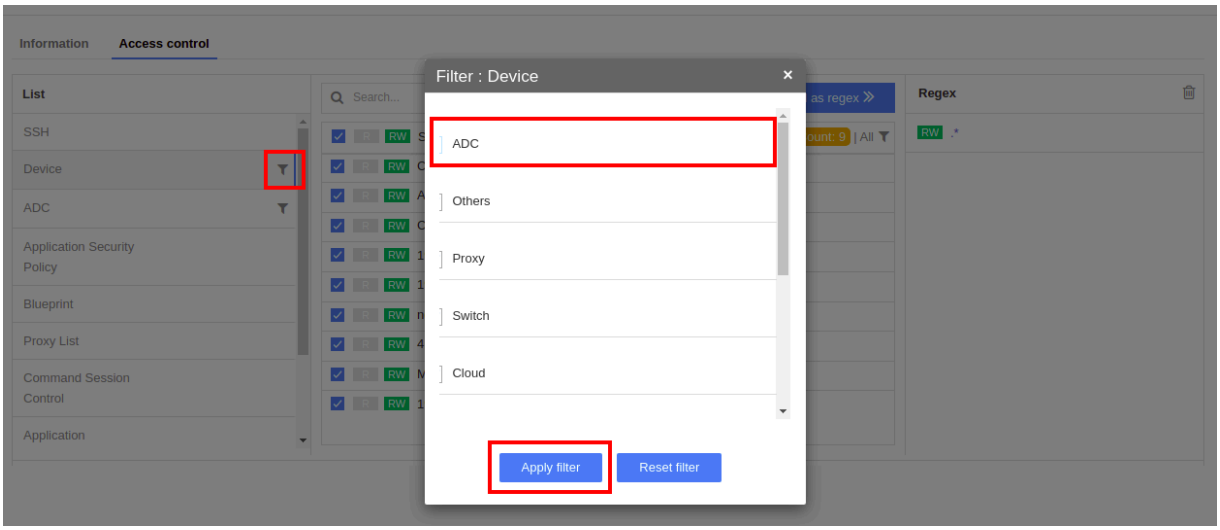


4. Click the **Access control** tab, select **Device** from the list in the left panel.

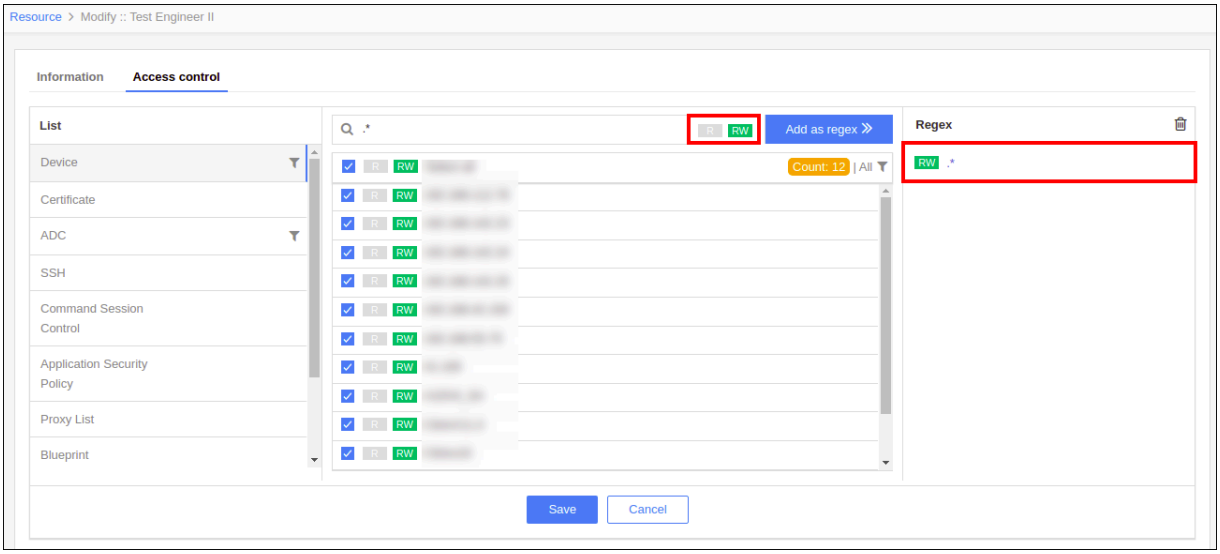


5. Select the filter icon, select ADC in the filter list and then click **Apply Filter**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Search for the regex of the device name(s), provide R/RW permission, and click Add as Regex to assign the device(s) to the resource.

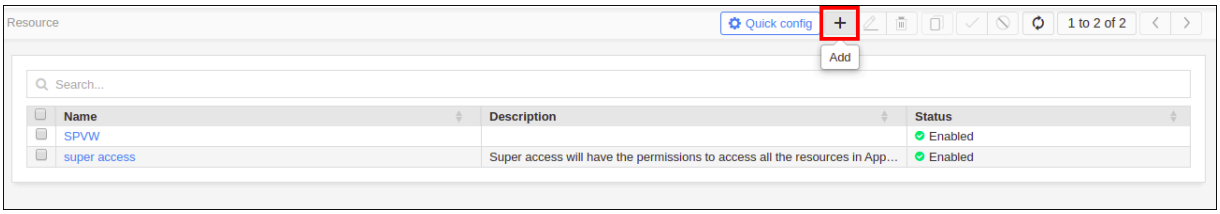


7. Click **Save**.

Assign Objects to Resource

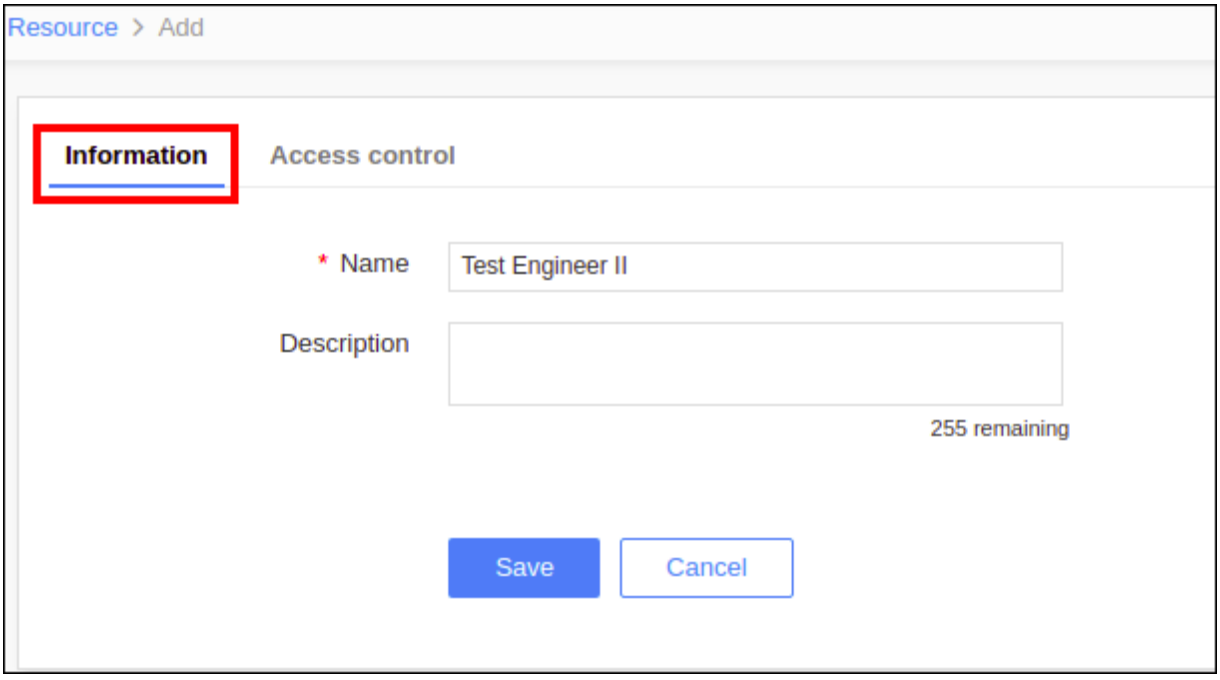
1. Go to **Menu** > **Account** > **Resource**.
2. Click on the  **(Add)** icon to navigate to the resource addition page.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



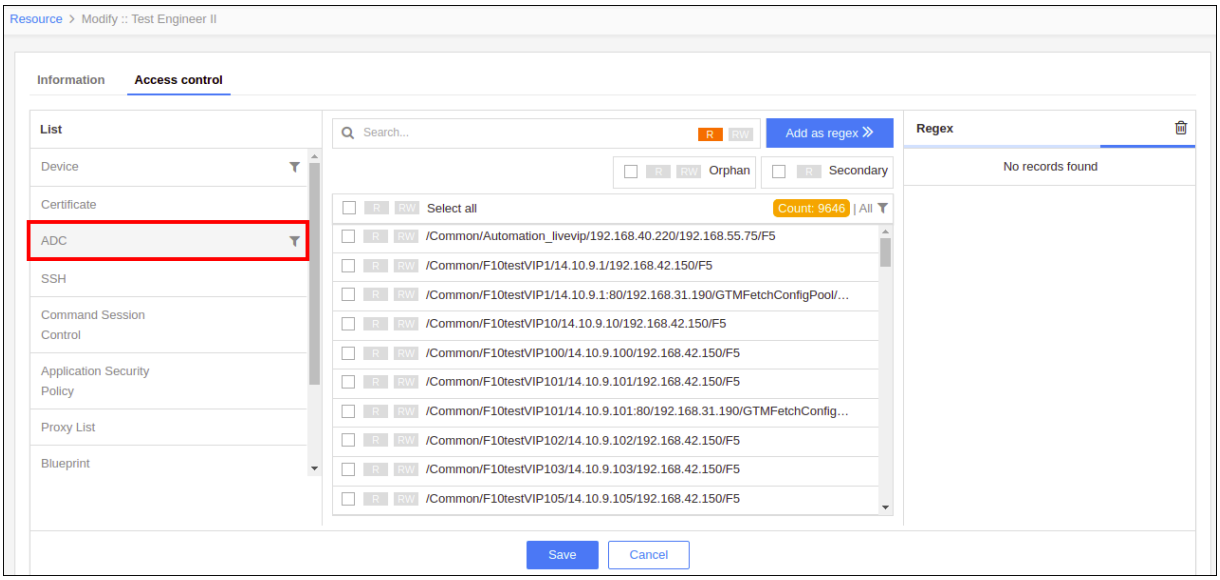
3. From the Resource Add page, configure the group details as given below and click **Save**.

Field	Type	Description	Validation
***Name**	Text	Name of the resource to be added.	Resource name cannot be empty.
Description	Text	Description of the resource.	A maximum of 255 alphanumeric characters and space are allowed.
**: Mandatory fields*			

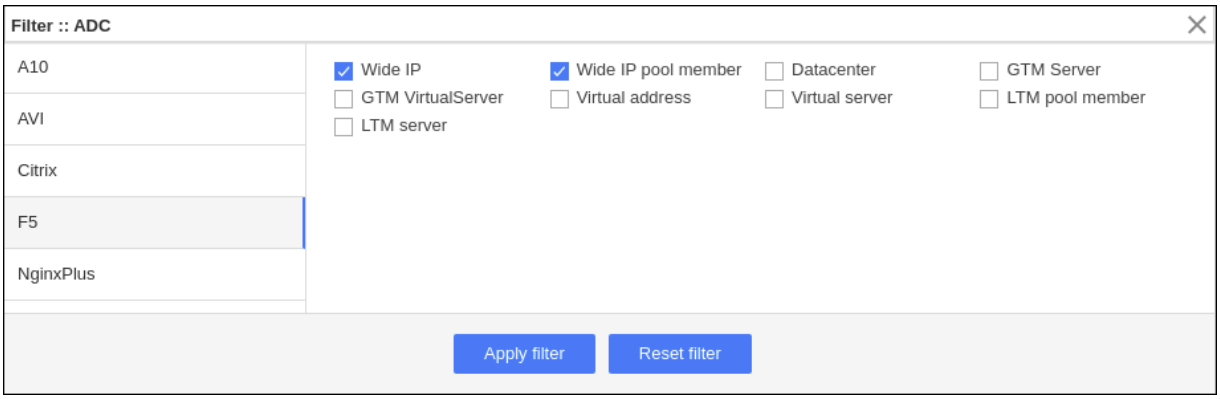


4. From the Resource Access control page, select ADC from the list in the left panel.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



5. Select the filter icon, select the ADC vendor in the filter list and the object type(s), and then click **Apply Filter**.



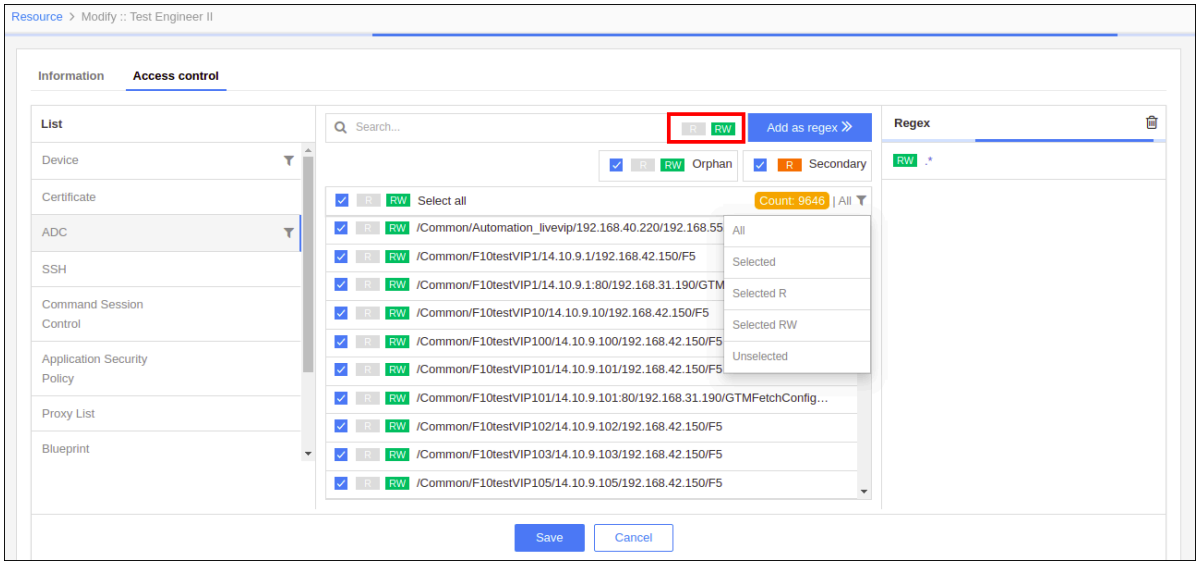
Note:

For F5 devices, the LTM Class object type is also displayed and by selecting this object type, the permission for class objects can be managed. The applied permission for the class objects from here will reflect in Dashboard, Control Center, and Restore/Compare.

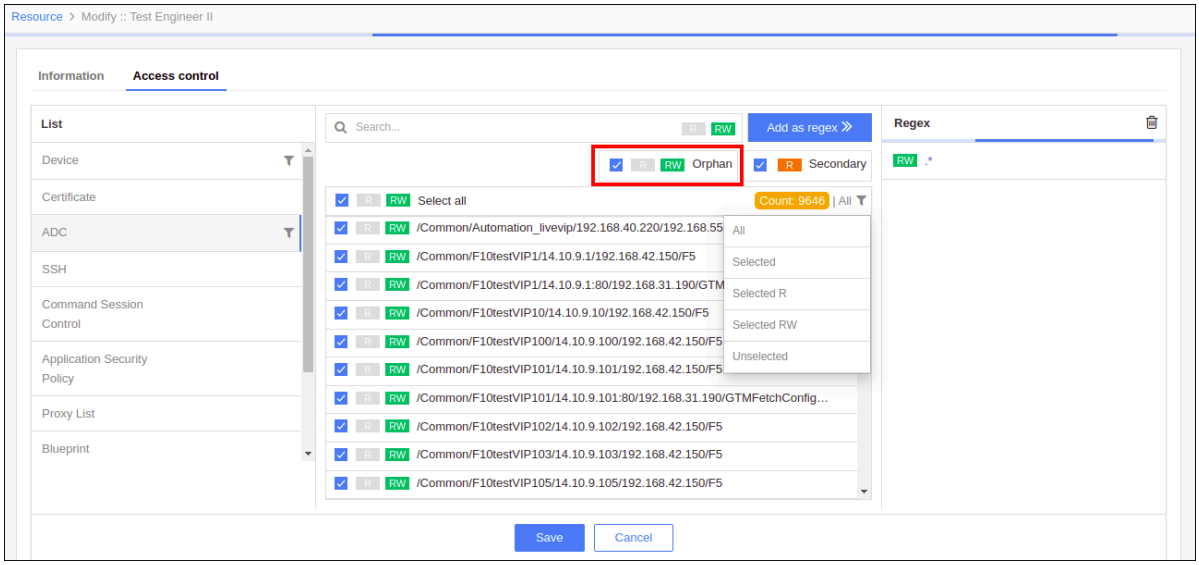
6. Search for the regex of the object display name(s).

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Select R/RW permission for the object(s).

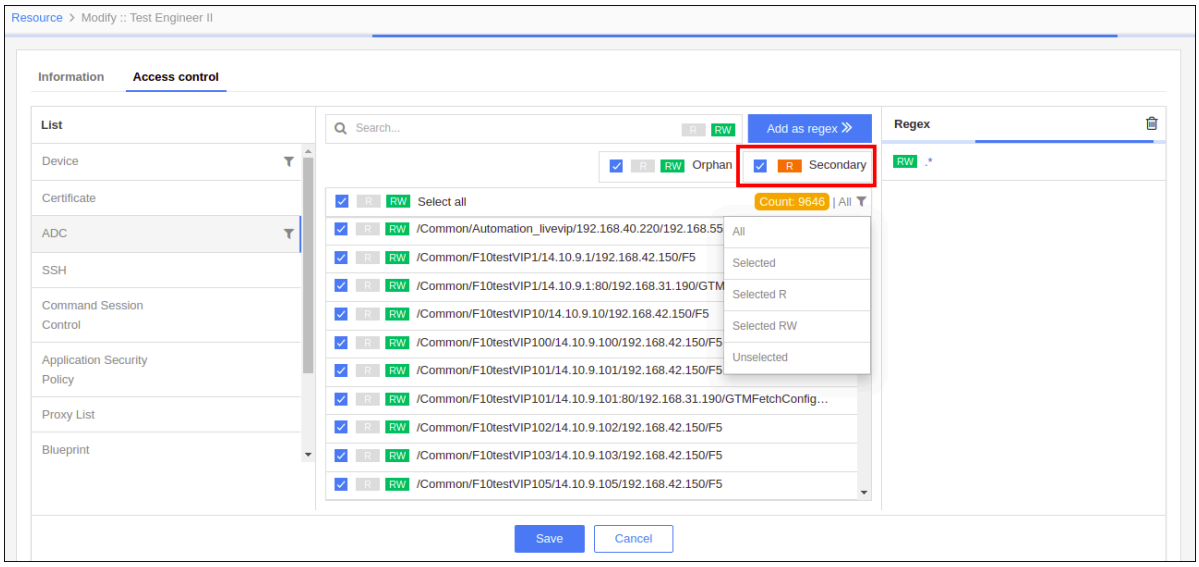


- Select R/RW permission for the orphan object(s)(optional).

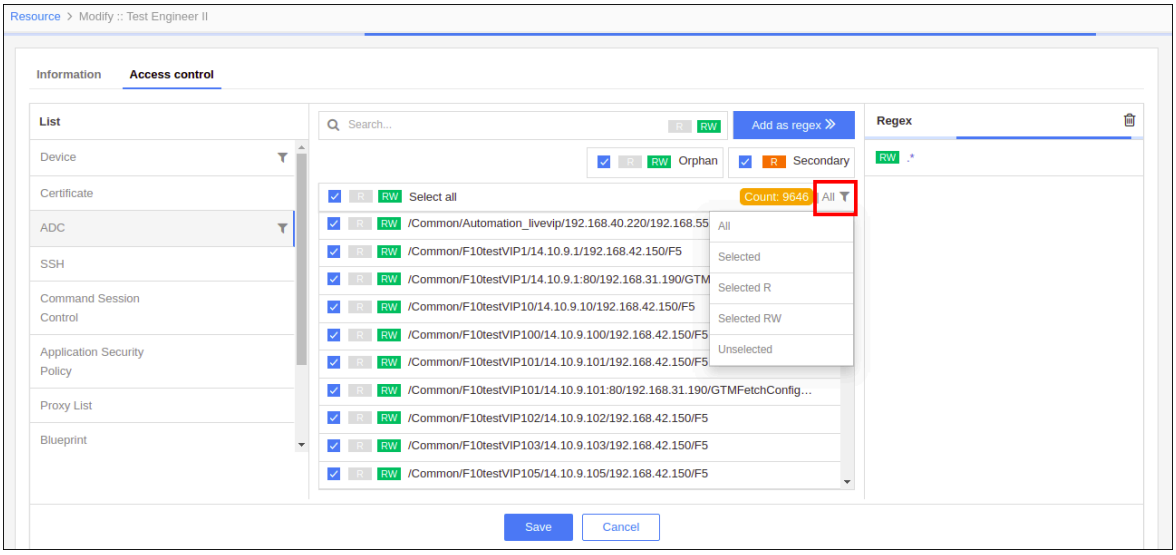


- Select R permission for the secondary object(s)(optional).

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

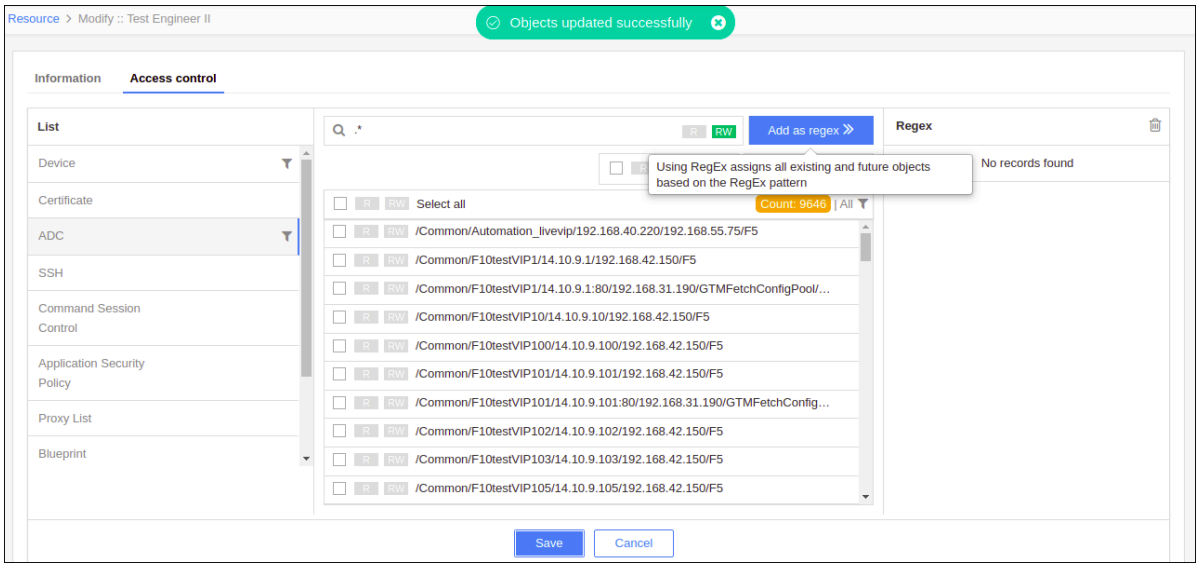


- Click the ALL filter icon to filter the objects based on the given permission.
  - Click All to get all the objects.
  - Click Selected to get the objects with any permission.
  - Click Selected R to get the objects with Read permission.
  - Click Selected RW to get the objects with Read/Write permission.
  - Click Unselected to get the objects without any permission.

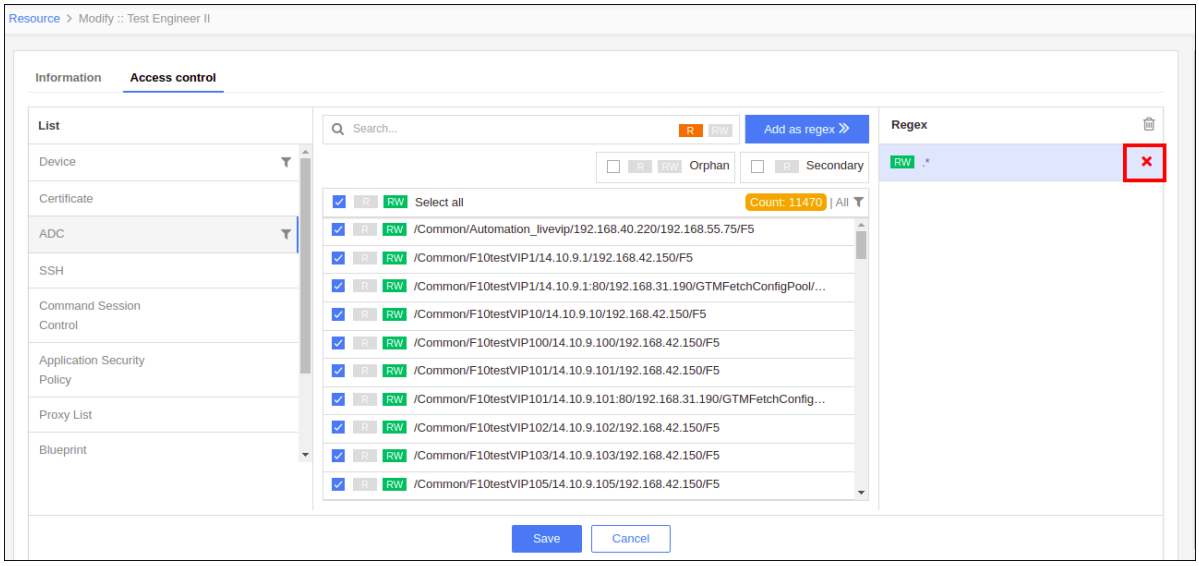


- Click Add as Regex to assign the objects(s) to the resource.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

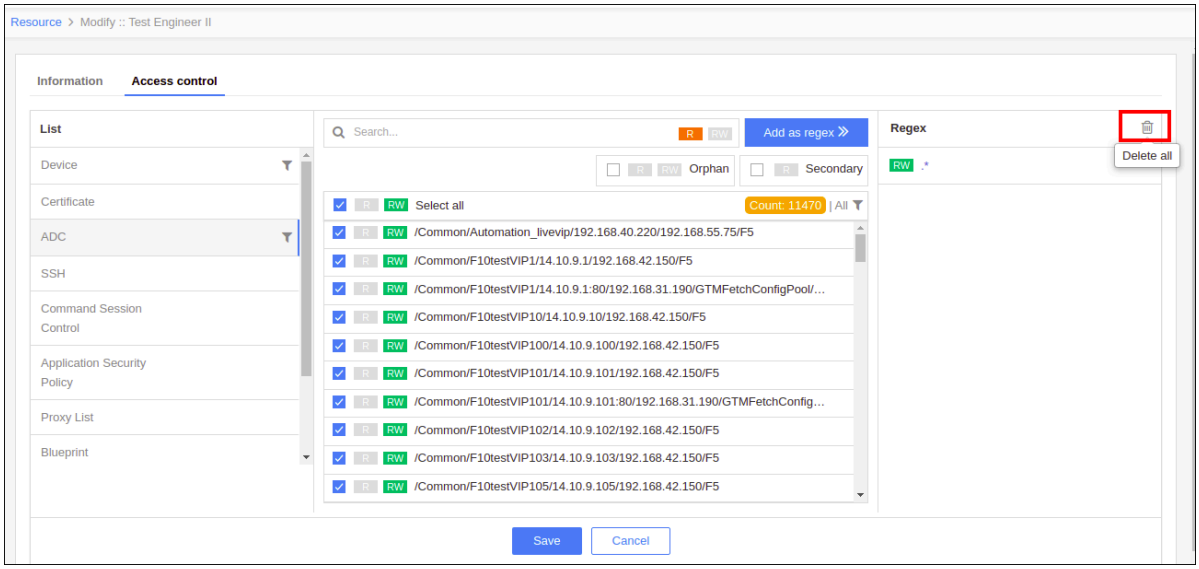


- Click the cross icon to delete any regex from the resource.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Click the delete icon to delete all the regex from the resource.



7. Click **Save**.

Studio

This chapter covers the major features and functionality of the AppViewX Studio module, which is a platform that can be used for the following purposes:

- Building custom workflows and tasks to make network operations agile
- Performing decision-based automation to achieve single or zero-touch provisioning
- Automating tasks using a third-party integration
- mention the OOB workflows for ADC that are available.

The Studio module is divided into three sub-systems, each of which has its own set of tasks you can perform:

- **Workflow** - It comprises of the following:
  - **Design** a custom workflow using a drag and drop UI or Import an existing workflow.
  - **Variables & Hooks** - Execute custom REST or script codes on top of the existing code using the **Hooks** section. You can define **Magicvariables** (static or dynamic), which are similar to the global variables. The only difference is that the global variables can be used only within one workflow,

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

whereas magic variables can be used across workflows. The hooks can also be mapped with the magic variables.

- **General** - Add/manage Python codes and regular expressions using the **Helper script** and **Regex Library sections** respectively. You can access the AppViewX **GitHub** repository containing various reusable workflows.

For more detail, refer to Visual Workflow Guide (link).

- **Reports** - It comprises of the following tabs:
  - **My reports** contain all the reports created by you or all the reports that have been created, depending on the permissions you have been assigned. It allows you to create, clone, or delete report(s).
  - **Store** allows you to view or clone 31 (6 samples and 25 certificates) pre-built reports.

For more detail, refer to Reports Guide.

- **Rules** - Allows you to create an action (force down, enable, and disable) related rule(s) to trigger workflows for the ADC objects.

For more detail, refer to Visual Workflow Guide (link).

- **Pages** - AppViewX provides a one-stop portal with all the service offerings in the form of low code pages to enable self-servicing for the end-users (like App owners, Network Engineers, CA, Admin, etc.) in a simplified and intuitive manner. There are Out of the Box catalogs and as well option to customize catalogs as per the organization standards. Access the opinion through **Menu** > **Studio** > **Pages**.
  - **Benefits**

- An intuitive self-service portal to access all service offerings for employees, and partners via Single Pane of Glass.
- Customer Experience – Users only see services that are available to them, described in business terms they understand.

- Build your own App catalog/portal.

- Configurable Catalog page that allows for personalized task views
- Low Code designer portal – Low Code and configurable toolsets, allowing users to brand Catalog solution so it feels familiar and intuitive for the end-users.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

For more detail, refer to Service Catalog User Guide.

Rules

Rules Inventory helps to define custom business rule(s) for specific actions and workflow(s) can be invoked for closed-loop automation.

Users can customize any action as per the business use case using workflow and associate it with the actions. The workflow will be triggered for a Rule-based on the criteria configured.

Key Features:

- A common Rule Inventory integrating key platforms (i.e., Control Center, Device Management, Alerts, Insight, etc.).
- Provision to define custom business rule(s) for module-specific actions.
- Provision to define the trigger, rule criteria for module-specific actions.
- Provision to define one or many workflow actions against a rule.
- Provision to execute the specific action defined based on the pre-defined rule.
- Provision to trigger workflow requests/token ID, and indicate the relevant status.
- Provision for a common notification center for actioning workflow requests.

Why Rules, Policy in AppViewX?

A policy is a condition that when met causes the system to perform a specific set of actions.

Policies can be used for event-driven automation and to automate network change and

configuration tasks such as:

- A comprehensive policy/rule inventory to define, manage policies, and triggers.
- Web interface to configure business data objects without changing code.
- Eliminates hard coding of complex business logic; enables faster policy-driven GTM solutions.
- Define internal rules across modules – Certificate, Alert, Control Center, Dashboard, Device Management, Reports, etc.
- Perform event-driven orchestration based on internal and external event(s).

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Provision to map Business processes to specific actions across AppViewX.
- Detecting and collecting configuration changes made by external users.
- Notifying users and systems when a configuration change is detected.
- Auto-remediating critical compliance violations.
- A rule can be a server-side, client-side logic that runs when any CRUD operation is performed.

The Rules sub-system within the Studio module allows you to perform the following tasks:

- Create a rule
- Clone a rule
- Delete a rule

Clone a Rule

To clone a rule:

1. Go to **Menu** > **Studio** > **Rules**.
2. The **Rules** screen displays a list of all the rules.
3. Select the rule you want to clone.
4. From the **Actions** dropdown, click **Clone**.
5. In the screen that appears, enter a **Name** for the cloned rule.
6. Click **Enable** to clone the rule in an enabled state. Alternatively, you can click **Save As Draft** to save the cloned rule to the AppViewX system and enable it later on.

Create a Rule

To create a rule using the Rules sub-system in the Studio module:

1. Go to **Menu** > **Studio** > **Rules**.
   A list of rules is displayed.
2. Click **Create Rule**.
3. In the **Rule Name** box, enter a name for the rule.
4. (Optional) In the **Description** box, enter additional information about the rule.
5. (Optional) Select the location/entity where the rule must run and the action upon which the rule must be triggered.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

6. (Optional) Select the specific action on a page where the rule has to be executed. The action could be Enable or Disable.
7. Associate a workflow against the Rule.
   AppViewX provides default Action execution workflows that integrate with Service NOW for change management. You can associate a default workflow or custom workflows.
8. (Optional) Define the **Rule criteria**.

> **Note:**
> The default workflows are available in the Studio >Workflows> Rules category. AppViewX allows to clone and customize the default workflows.

Delete a Rule

To delete one or more rule(s):

1. Go to **Menu** > **Studio** > **Rules**.
2. The **Rules** screen displays a list of all the rules.
3. Select the rule(s) you want to delete.
4. From the **Actions** dropdown, click **Delete**.
5. In the **Confirmation** dialog box that appears, click **Yes**.

Reports

The Reports sub-system within the Studio module allows you to perform the following tasks:

> **Note:**
> For more details about the report, refer to the Reporting User Guide.

Clone a Report

To clone a report:

1. Go to **Menu** > **Studio** > **Reports**.
   The Reports screen appears.
2. In the **My Reports** tab, all the reports are displayed based on the permissions that you have been assigned. Select the report you want to clone.
3. From the **Actions** dropdown, click **Clone**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. In the **Clone report** screen that appears, enter a **Name** for the cloned report.
5. Click **Save**.

Create a Report

To create a report using the Reports sub-system in the Studio module:

1. Go to **Menu** > **Studio** > **Reports**.
2. The **Reports** screen appears. In the **My Reports** tab, all the reports are displayed based on the permissions that you have been assigned.
3. Click **Create New Report**.
4. In the **BASIC INFO** screen of the report creation wizard,
   - Enter a **Name** for your report.
   - (Optional) In the **Description** field, type additional information related to the report.
   - Click **Yes** if you want to control access to the report using different permissions for different users.
   - Click **Save** to resume the chart creation later or click **Next** to proceed with the chart creation.
5. In the **DATA SOURCE** screen,
   - From the **Select data source** dropdown, choose one of the following sources to query data, and build the report:
     - **Query builder** - Allows you to select one of the existing queries or create a new query.
     - **Hooks** - Allows you to select one of the pre-built OOB hooks or create a new hook (Script or REST).
   - Click **Save** resume the chart creation later or click Next to proceed with the chart creation.
6. In the **CHART CONFIGURATION** screen,
   - Click on one of the following reports to **Select chart type - PIE**, **DONUT**, **BAR**, **STACKED BAR**, **GRID**, **LINE**, and **METRIC**.
   - The fields that appear vary for each chart type, at a minimum, fill in all the mandatory fields.
   - Click **Save** to resume the chart creation later or click **Next** to proceed with the chart creation.
7. In the **CHART DRILLDOWN** screen, select the checkbox to view more specific layers of the data or information being analyzed.
8. Select one of the following drill-down types for the chart:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Set redirect URL** to configure the URL to any page to which the redirection from the chart must happen.
- **Grid** to associate the chart to a hook and a workflow.

9. Click **Save & Enable** to save the report to the AppViewX system.
10. The report is added to the **My reports** tab, it can be enabled or disabled using the toggle button in the **Status** column.
11. After the report is created, you can perform one of the following actions on the left-hand pane:
    - **Pin** the report to a new or an existing dashboard.
    - Set the **Interval** during which the chart data must be collected. This can be customized to happen once or recursively.
    - **Share** the report with the various recipients at a specific point of time once or repeatedly.
    - Enable or disable the report using the toggle button.

Delete a Report

To delete one or more reports:

1. Go to **Menu** > **Studio** > **Reports**.
   The **Reports** screen appears.

   In the **My Reports** tab, all the reports are displayed based on the permissions that you have been assigned.
2. Select the report(s) you want to delete.
3. From the **Actions** dropdown, click **Delete**.
4. In the **Confirm delete** dialog box that appears, click **Yes**.

Settings

AppViewX's ADC Settings helps manage the Device and Object specifications related to Configuration synchronization, Report, and Statistics.

Device Settings

The device settings can be configured in the following tabs:

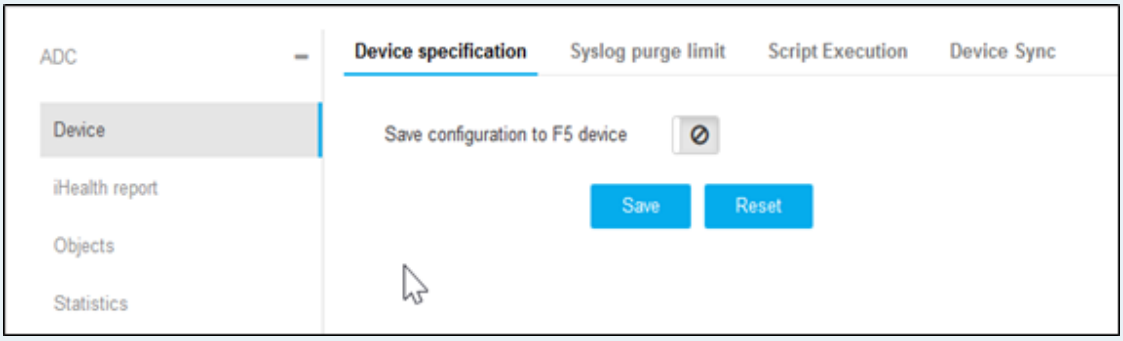
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Specification Tab

Save the changes of device specification configuration on the F5 device after config fetch. This enables all the changes to be reflected on the F5 file.

To configure a device specification:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Device Specification** tab.
2. Click the toggle icon to enable **Save Configuration to the device**.

> **Note:**
> In this example, F5 device specification is shown.
>
> 
>

3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

Syslog Purge Limit

Syslog purge limit enables to limit of the number of Syslog persisted against a device. The oldest logs will be automatically deleted on exceeding the limit.

To configure the Syslog purge limit:

1. Go to **Menu** >**ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Syslog purge limit** tab.
2. On the **Syslog purge limit** tab, specify the syslog limit per device. By default, the limit is set to 1000 and can be extended till 5000.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



3. Click **Save**.

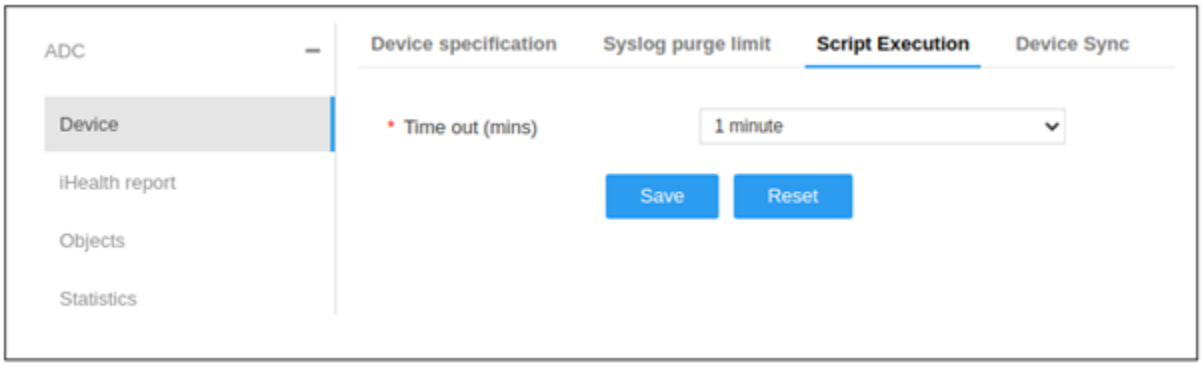
> **Note:**
> To reset the changes, click **Reset**.

Script Execution

Script Execution setting allows you to specify the time-out limit of script execution. If the script execution exceeds the specified time, AppViewX will automatically terminate the script.

To configure script execution:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Script Execution** tab.
2. On the **Script Execution** tab, specify the timeout limit.



3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Sync

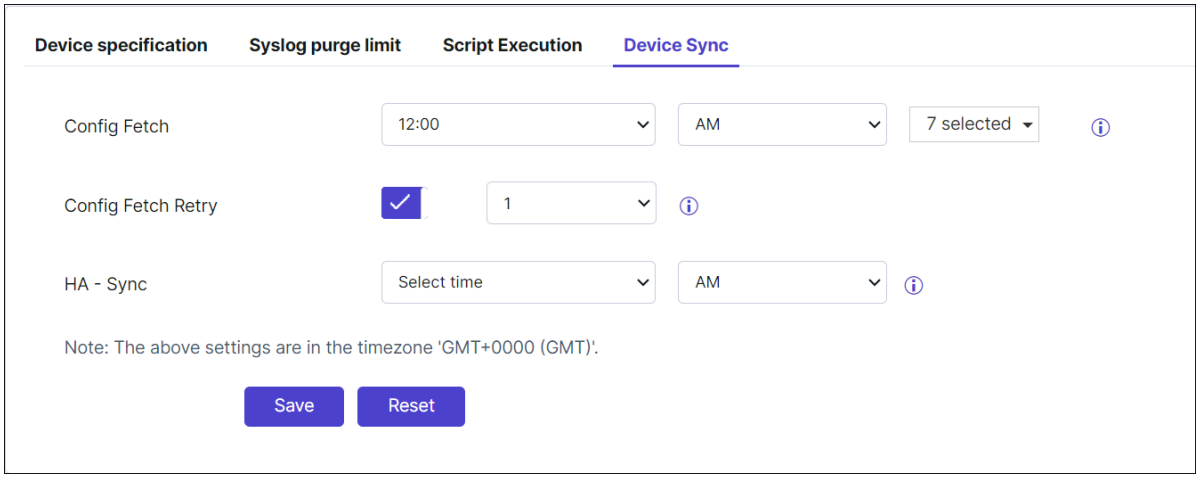
The device Sync feature allows to customize the configuration sync operations.

To customize the configuration sync operations:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **Device** > **Device Sync** tab.
2. On the **Device Sync** tab, specify the following:

- **Config Fetch** - Allows to schedule the config fetch process at a specific time for selected days. Ensures device changes are reflected in AppViewX.
- **Config Fetch Retry** - Enable this option and select the maximum number of retries for devices that move to a failed or unresolved state during the midnight sync. The retry will be started based on the config fetch settings. By default, the config fetch duration is set to 4 hours and the frequency is set to 15 minutes. You can configure the duration and frequency of the retry attempts in AppViewX. The parameters are:
  - **CONFIG_FETCH_RETRY_DURATION** - Specifies the duration for retrying config fetches, defined in hours.
  - **CONFIG_FETCH_RETRY_FREQUENCY** - Specifies the frequency of retry attempts, defined in minutes.
- **HA - Sync** - Ensures the HA devices are synchronized in specified intervals.

> **Note:**
> To set device sync, the AppViewX Group Sync flag must be enabled while adding a device to run HA sync as per configured interval.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

Statistics

The Statistics feature allows to set the minimum auto refresh time interval.

To configure the minimum auto refresh time interval:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **Device** > **Statistics** tab.
2. On the **Statistics** tab, select the desired minimum auto refresh time interval from the **Minimum Interval** drop-down menu.
3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

Object Settings

The object settings can be configured under the following tabs:

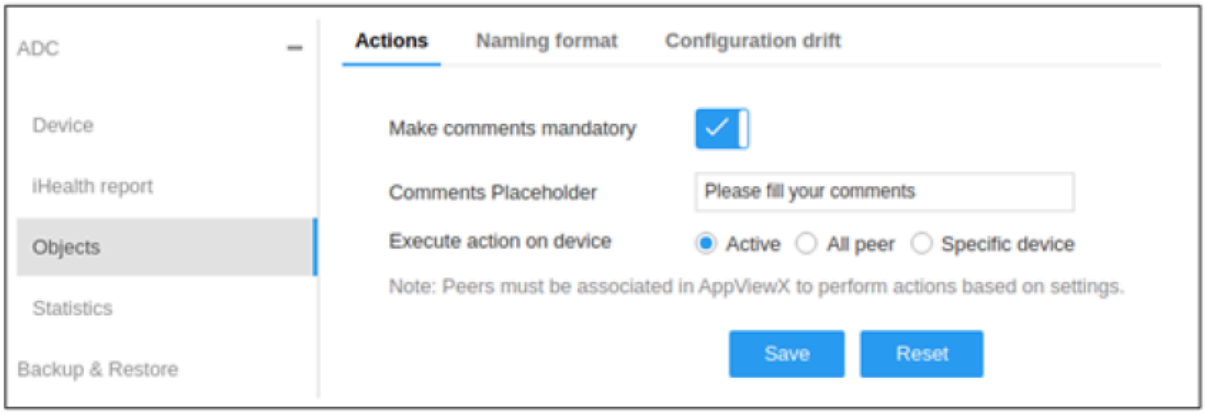
Actions Tab

On the Actions tab, you can customize the actions execution as needed

To customize the actions:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Objects**.
   By default, the **Actions** tab appears.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



2. Enable the **Make Comments Mandatory** toggle button to set the comments option mandatory while executing an action from the Control Center or dashboard. This ensures end users are mandated to mention the reason for their actions.
3. Enter comments in the **Comments Placeholder**, such as SNOW ID, Enter Application impacted, etc. to impose a standard to be followed upon action execution.
4. Select **Active**, **All Peer**, or **Specific Device** radio buttons for **Execute Action on Device**. By default, AppViewX actions are triggered on current Active devices. This can be modified as needed.
5. Ensure Peers are associated in AppViewX for seamless execution.
6. Click **Save**.

> **Note:**
> To discard the changes, click **Reset**.

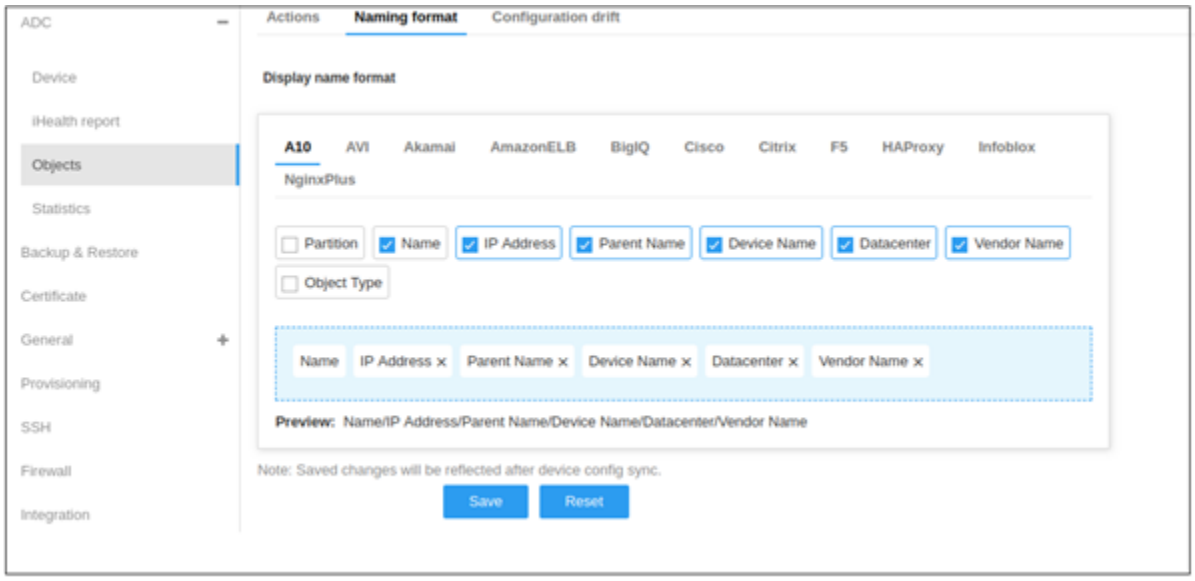
Naming Format Tab

On the Naming Format tab, you can customize the display name format of your objects that needs to be followed throughout the application.

To customize the naming format:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Objects**.
2. Click the **Naming Format** tab.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



3. On the **Display Name Format** page, choose the vendor to be modified and select the respective Property, reorder the fields as needed.
4. Click **Save**.

> **Note:**
> To discard the changes, click **Reset**.

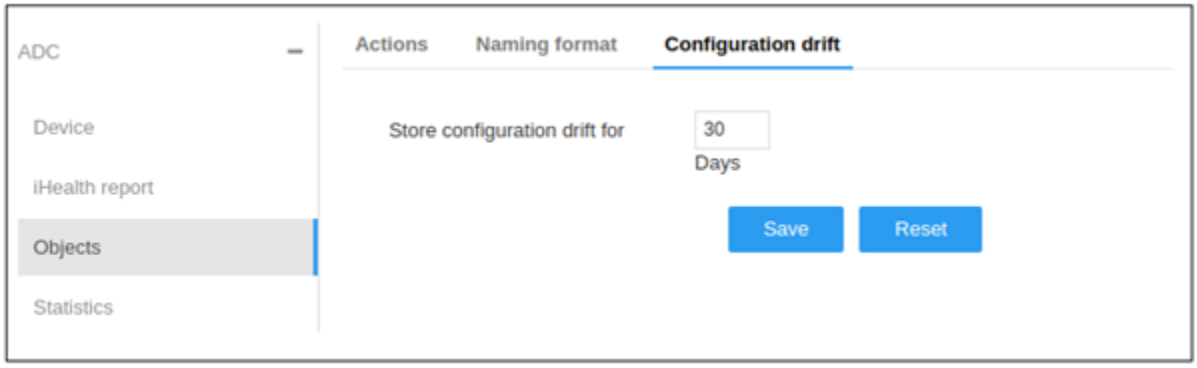
Configuration Drift Tab

AppViewX identifies the object configuration changes (as part of config fetch, Syslog notification, etc.,) and stores the drift to generate reports, governance, and rollbacks. You can customize the number of days the drift must be persisted.

To customize the number of days for the configuration drift:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Objects**.
2. Click the **Configuration Drift** tab.



3. Enter the store configuration drift value up till 90 days in the **Store Configuration Drift** for the field.
4. Click **Save**.

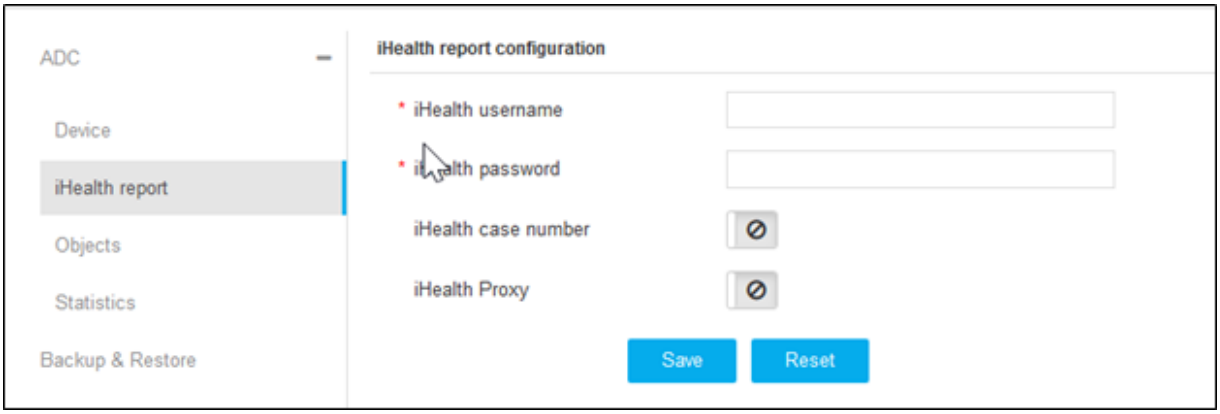
> **Note:**
> To discard the changes, click Reset.

iHealth Report

Configure health portal information for a vendor to generate iHealth reports of devices.

To configure the iHealth report:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **iHealth Report**.



2. On the **iHealth Report Configuration** page, enter the **iHealth username** and **password**.
3. Enable the **iHealth case number** and **iHealth Proxy** toggle buttons to upload the generated qkview file against the mentioned case number for future reference.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click **Save**.

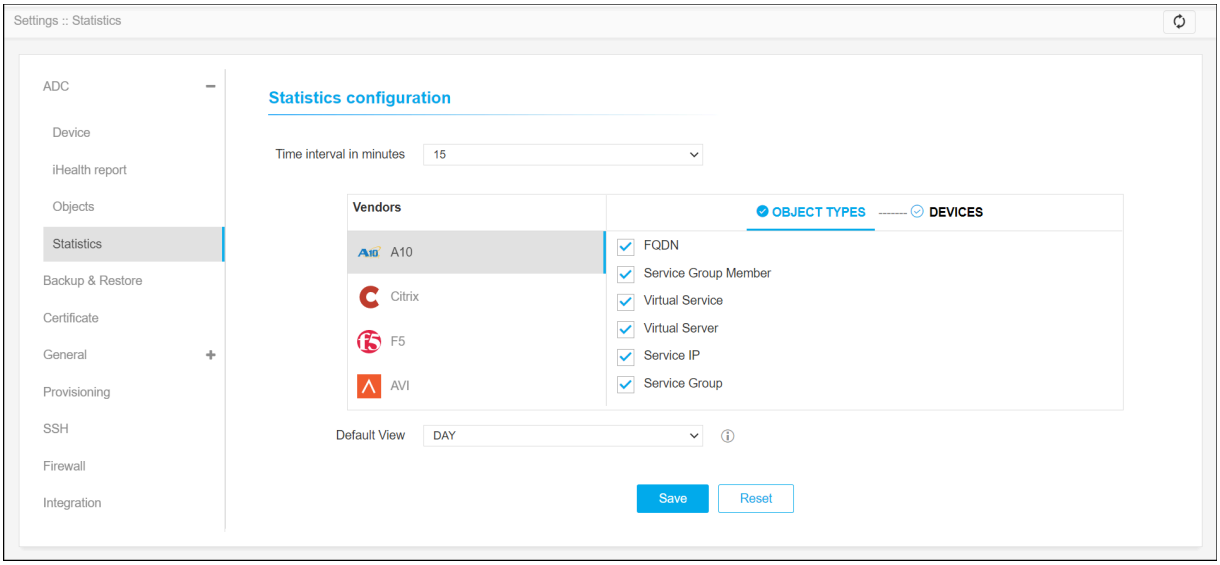
> **Note:**
> To discard the changes, click **Reset**.

Statistics Settings

Configures the statistics data collection to monitor the historic statistics with respect to the vendors and object types. Statistics plugin has to be deployed in the node for collecting statistics data.

To configure statistics:

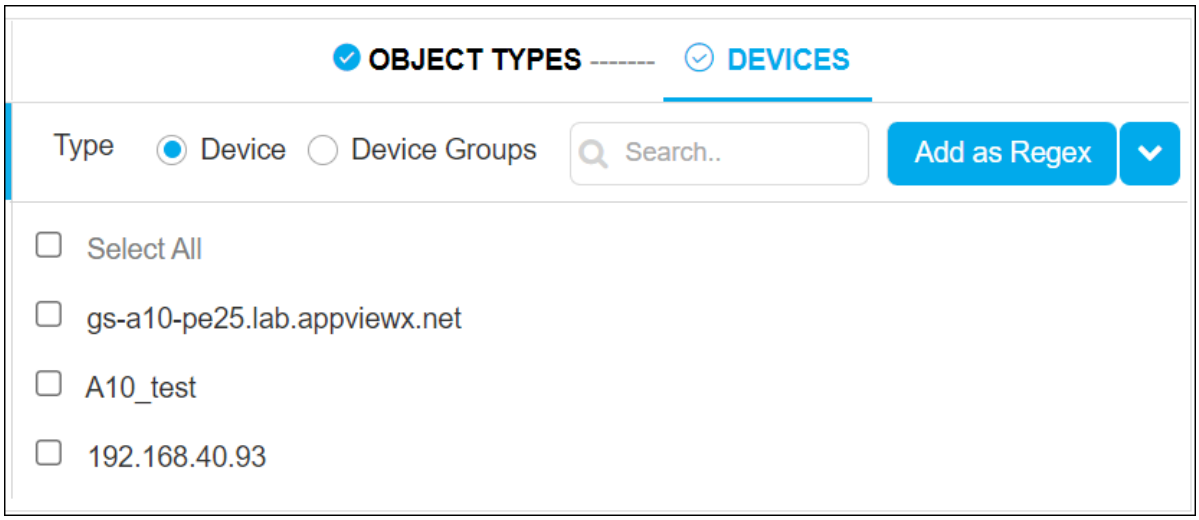
1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Statistics**.



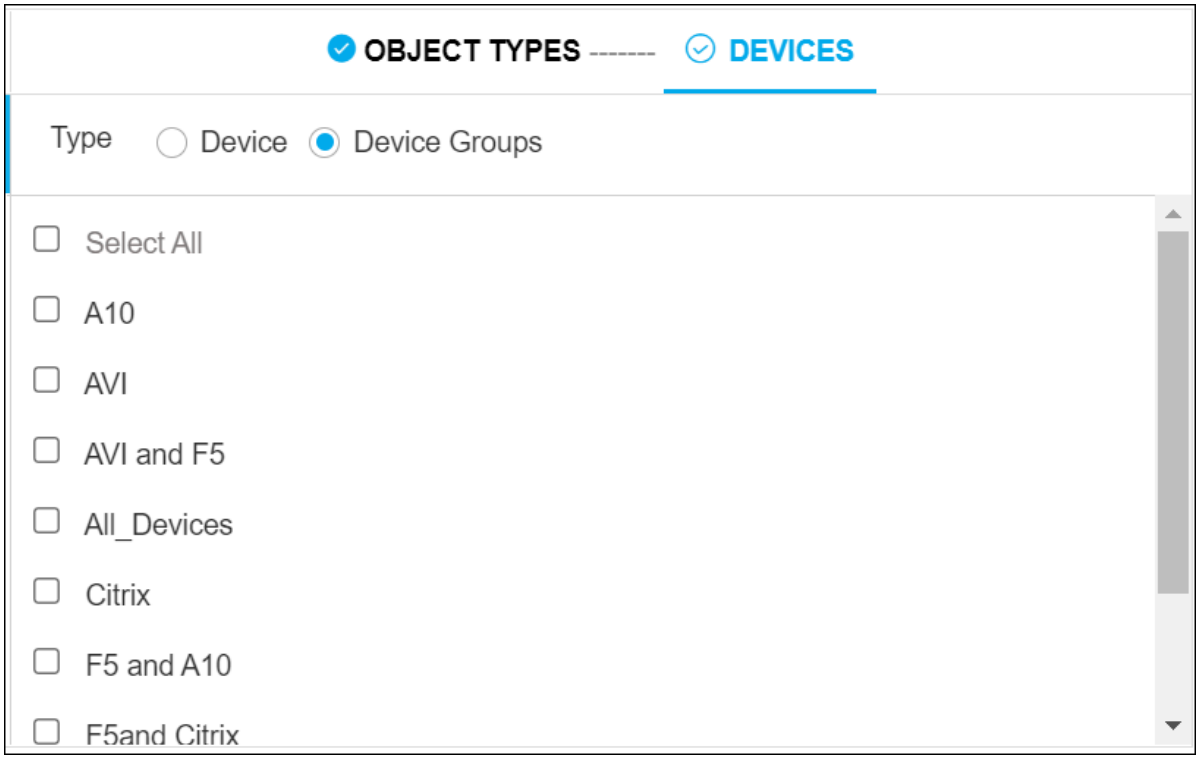
2. On the **Statistics Configuration** details page, choose the time from the **Time Interval in Minutes** drop-down.
3. Select the vendor, and then choose the object types and devices from the list for which statistics must be fetched.
4. For the device selection, click the **DEVICES** tab, and then,
   - Select the **Type** of device selection as **Device** or **Device Groups**.
   - When you select the **Device** radio button, you get the option to select the devices from the list either by searching or using regex.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
If Regex is added for filtering the device, the statistics will be configured for the newly added devices as well, which match the regex.



- When you select the **Device Groups** radio button, you get the option to select the device groups from the list.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. Select the default view for the Traffic statistics widget from the **Default View** dropdown. The possible options for default view are:

   • Day (default)

   • Live

   • Week

   • Month

   • Quater

6. Click **Save**.
   AppViewX collects the stats, aggregates them, and provides out-of-the-box reports in the form of dashboards.

   Note:
   To discard the changes, click **Reset**.

Glossary

This table describes the common terms used in this guide.

Terms	Definition
LDAP	LDAP stands for Lightweight Directory Access Protocol. As the name suggests, it is a lightweight client-server protocol for accessing directory services, specifically X. 500-based directory services.
Object	Objects are traffic management entities configured in the devices that are responsible for load balancing the traffic. Eg: A virtual server is a traffic-management object on the BIG-IP system that is represented by an IP address and a service. For a given network, there are usually multiple objects functioning. Some object types - WideIP, Virtual Server, Pool, Pool Member, Irule, Monitor, Profile, Policy, etc.,
OOB Workflow	A workflow defines a logical flow of activities or tasks from a Start event to an End event to execute a specific service. AppViewX ships predefined Out of the Box (OOB) workflows that allow application teams to take ownership and self-service their applications.
Resource	Resources are the devices and objects that are configured within AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Chapter 2: ADC+ Admin Guide

Use this guide to set up and manage ADC+ administrative functionality. Also learn about the new features and capabilities that make it easier for you to configure and administer your ADC+ deployment. The documents helps to configure below functionalities:

- AppViewX provides App centric visibility of your ADC infrastructure in a single window.
- Onboard the supported ADC vendor devices (Hardware, Cloud and Software) into the AppViewX inventory using the IP Address/FQDN.
- Helps to configure credentials and Discover the Applications/Objects along with its configuration that are hosted on the devices. The Discovered Applications can be accessed within the product.
- Steps to configure the access control policies to access devices and its application objects. The AppViewX Platform applies granular access control to application objects, certificates and configuration templates.
- Explains possible Search engine patterns that allows to look for any application and also visualize the topological view starting from the Global load balancer to the end server. This provides the network teams visibility into the infrastructure to troubleshoot application-related issues faster.
- Steps to configure dashboard to monitor Application health, state, status, utilization and performance through pre built dashboards. Monitor and Manage Application Traffic through Custom Dashboards and Widgets that allows Traffic routing, Monitor live traffic and Distributes traffic ratio across data centers.
- Steps to manage network configuration by automating the backups, track and report network changes and ensure they are compliant with the defined policies.
- Helps to understand/configure the Settings to periodically monitor your Application Services or Devices to get notified of any critical changes via alerting module.

Getting Started

Overview

Use this guide to set up and manage ADC+ administrative functionality. Also learn about the new features and capabilities that make it easier for you to configure and administer your ADC+ deployment. Setup of the AppViewX Cloud Connector is necessary to enable the connectivity between AppViewX Cloud and the Enterprise Network. For AppViewX SaaS deployment, refer to AppViewX Cloud Connector User Guide. The documents helps to configure below functionalities:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- AppViewX provides App centric visibility of your ADC infrastructure in a single window.
- Onboard the supported ADC vendor devices (Hardware, Cloud and Software) into the AppViewX inventory using the IP Address/FQDN.
- Helps to configure credentials and Discover the Applications/Objects along with its configuration that are hosted on the devices. The Discovered Applications can be accessed within the product.
- Steps to configure the access control policies to access devices and its application objects. The AppViewX Platform applies granular access control to application objects, certificates and configuration templates.
- Explains possible Search engine patterns that allows to look for any application and also visualize the topological view starting from the Global load balancer to the end server. This provides the network teams visibility into the infrastructure to troubleshoot application-related issues faster.
- Steps to configure dashboard to monitor Application health, state, status, utilization and performance through pre built dashboards. Monitor and Manage Application Traffic through Custom Dashboards and Widgets that allows Traffic routing, Monitor live traffic and Distributes traffic ratio across data centers.
- Steps to manage network configuration by automating the backups, track and report network changes and ensure they are compliant with the defined policies.
- Helps to understand/configure the Settings to periodically monitor your Application Services or Devices to get notified of any critical changes via alerting module.

Supported Web Browsers

Supported Web Browsers

Browser	Version	Notes
Firefox	Till latest (Version 84.0.4147.135)	
Chrome	Till latest (Version 80.0)	
IE	Limited support in 9, Full support from 10+	No support for IE9 post AppViewX Version 11.0
Safari	Till latest (Windows - Version 5.1.7, macOS - Version 13.1.2)	From AppViewX Version 11.1
Opera	Till latest (Version 70)	From AppViewX Version 11.1

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

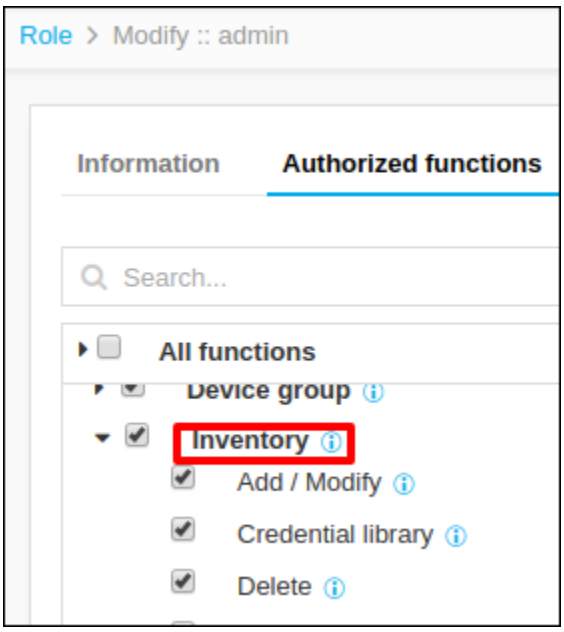
Supported Device and OS

Device	OS	Resolution
Desktop	Windows	1024 X 768 onwards, 1366x768, 1920x1080, Higher
Desktop	Linux	1024 X 768 onwards, 1366x768, 1920x1080, Higher
Desktop	Mac	1024 X 768 onwards, 1366x768, 1920x1080, Higher
iPad	iOS	1024 X 768

Inventory Actions

Before You Begin

To do any actions in the ADC-inventory, the user should have ACF permissions to access the inventory page and its various actions as mentioned in the screenshot.



Importing Devices

Device import provides a hassle-free experience in onboarding multiple ADC devices into AppViewX in one single step. For onboarding multiple devices, the details should be filled in the excel sheet in the

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

predefined format and can be uploaded to AppViewX and from there AppViewX will dynamically onboard all the devices available in the sheet.

To import devices using a `.csv` file:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Click the **Import** icon in the Command bar.
3. On the Import screen that appears, navigate to the location of the import file, then select it.
4. Click **Import** to add the devices and their details to the Inventory.

> **Note:**
> When the file is uploaded with improper structure or incorrect data, the import process will terminate with the errors highlighted.

Related information
Managing Devices

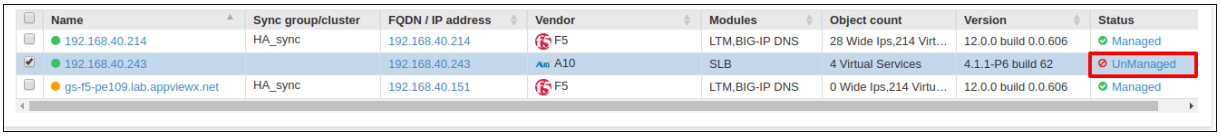
Managing Devices

To manage or unmanage devices:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. If the device you want to manage or unmanage is not listed on the screen, run a search to locate it.

> **Note:**
> If you manage a device that is already in managed state or unmanage a device that is already in unmanaged state, an error message appears at the top of the screen.

3. Click the checkbox beside the device name.
4. To start managing the device, click the (**Manage**) icon in the Command bar at the top of the screen.
5. To stop managing a device, click the (**Unmanage**) icon.
   The selected devices will be moved to **Unmanaged** status.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Related information
Exporting Device Details
Deleting ADC Device(s)

Exporting Device Details

The device details, which are available in the Device Inventory page can be exported into an Excel file.

To export the details of one or more devices:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. If the device you want to export is not listed on the screen, run a search to locate it.
3. Click the checkbox beside the device name. If you are exporting details of multiple devices of the same kind, select the checkboxes for each one.
4. Click the **Export** icon in the Command bar at the upper right of the screen.
5. On the **Export** pop-up screen that appears, select the type of information you want to export:
   - **All Columns** - Select this option if you want to export all information about the device.
   - **Displayed columns** - Select this option if you want to export only the information that is visible on the Device screen. This is useful if you need to compare values or settings for different devices and do not have any need to see the less important data.
   - **Columns to modify data and import** - Select this option if you are exporting device details to make modifications and then re-import the data into the Device Inventory.
6. On the screen that opens, select the location where you want the device details file to go, then click **Save**.
   The details are then downloaded as an Excel (**.xls**) file.

Related information
Deleting ADC Device(s)

Deleting ADC Device(s)

To delete ADC device(s):

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Select the desired ADC device(s) to delete.
3. Click the  (**Delete**) icon in the Command bar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Delete** confirmation modal appears.

4. Click **Yes** to delete the selected ADC device(s).

> **Note:**
> To discard the deletion, click **No**. The device details and configurations will be permanently deleted from AppViewX. If the deleted device is onboarded again, it will be considered as onboarding a new device.

Subscribing and Unsubscribing to Syslog

Subscribing to Syslog for a device enables receiving near real-time configuration updates. The Subscribing Syslog option is supported only for devices that are in the Managed and Failed status.

Supported Vendor: F5

Supports near real time updates for state and status of the objects.

Prerequisites:

- During installation, enable the flag for Syslog receiver. For setting the flag, see AppViewX Cloud Connector User Guide.
- Ensure Syslog reception is enabled in Platform. See Platform user guide for enabling the Syslog reception.
- Ensure the device is managed.

To Subscribe/unsubscribe Syslog for a device:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Select the checkbox of a preferred device.
3. Select the **Actions** dropdown menu from the menu bar.
4. Select **Subscribe Syslog** from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The process of subscribing to Syslog may take some time, and once completed, the status in the Syslog column is updated to **Subscribed** along with a timestamp.

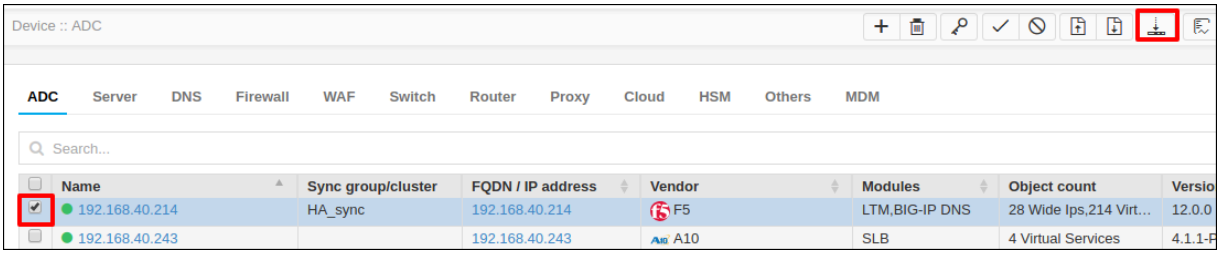
> **Note:**
> To cancel the subscription to Syslog, choose the **Unsubscribe Syslog** option from **Actions** dropdown menu. The unubscription option is supported for the devices that are in the Managed, Failed, or Unresolved status.

Config Fetch

This action is being used to fetch and update the device configuration in AppViewX.

1. Select **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2.
3.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Select the devices from the **Device Inventory** page.



3. Click the **Config fetch** icon.
   Once the config fetch is triggered, the device will be moved to **Queued** status and then **InProgress** status. Eventually, the device status will be updated to **Managed** once the device configuration is downloaded and parsed successfully in the AppViewX.

Generating and Downloading an iHealth Report

BIG -IP iHealth is a diagnostic tool developed by F5 to manage local traffic manager (LTM) and global traffic manager (GTM) devices. The iHealth report provides tailored diagnostic information that gives you valuable, actionable insight into the efficiency of the hardware and software running in your BIG-IP system.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- iHealth reports can be generated at the time you want to view or schedule it in advance through the Workflow.
- To generate an iHealth report, ensure that the proxy is configured in the Settings module. For detailed information, refer to the Proxy Settings section of Platform User guide.

To generate and download an iHealth report,

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
   By default, the **ADC** tab opens.
2. Select the checkbox beside the ADC device for which you want to generate an iHealth report.
3. Click the **Generate iHealth Report** button () in the Command bar.

   Note:
   You might need to scroll to the right to see the Report column.

4. On the iHealth Report Generate screen that pops up, enter the case number for the report, then click Generate.
   When the report is generated, it appears as a link in the **Report** column on the **Device: ADC** screen.
5. Click the **IHEALTH_REPORT** link.
   The iHealth report screen that pops up lists all of the current issues with the device, classified by their severity: critical, high, medium, or low.
6. Click any of the + (**Expand**) icons beside a severity level to view the issues within the corresponding category.
7. Within each severity level, click the name of a specific issue to view complete details, a recommended solution, and additional information about it.
8. (Optional) Download the entire iHealth report as a Qkview file by clicking the **Download Qkview** link or as a PDF file by clicking the **Export as PDF** link, both of which appear in the top right corner of the screen.

Related information
Selecting Inventory Columns
Pagination

Selecting Inventory Columns

The user can select or unselect the list of columns to be displayed in the inventory page using this action.

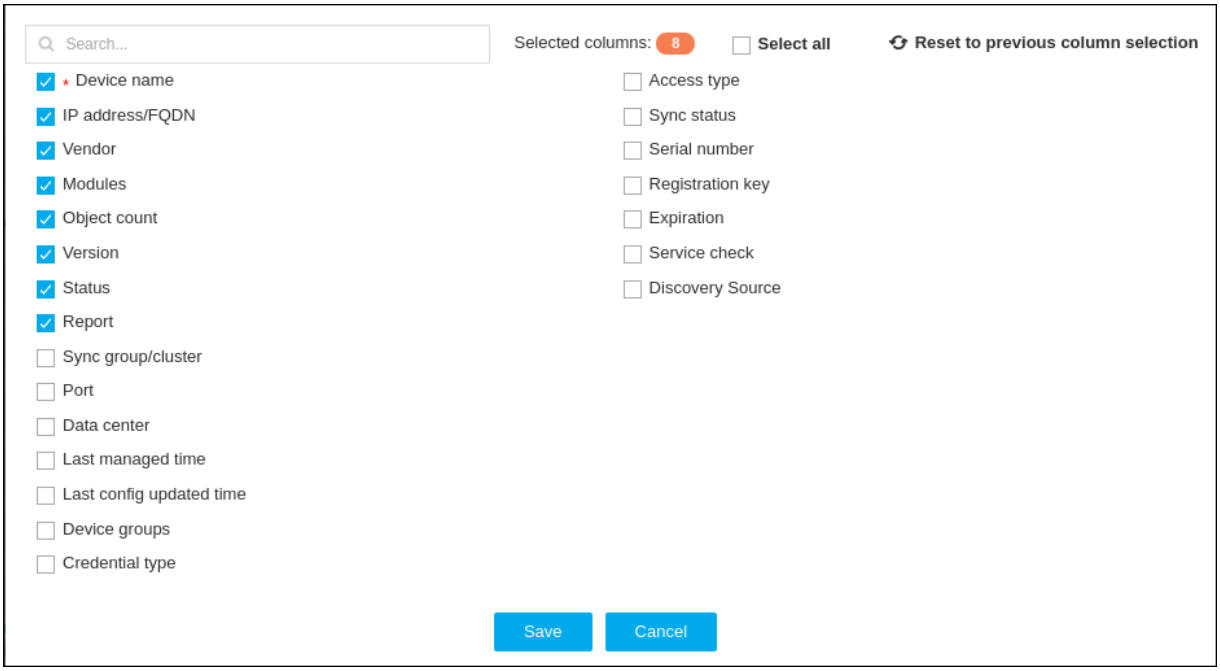
1. Select **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3.

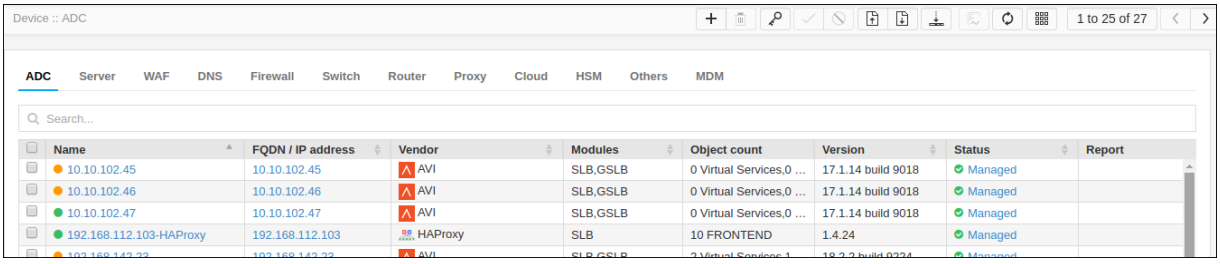
4.

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**.
2. Click on the **Columns** icon.
3. In the popup window, select or unselect the columns to be displayed in the inventory page.



4. Click **Save**.

   Once the user selects the columns to be displayed in the inventory page and saves the data, the device inventory page will be reloaded again with the selected columns.



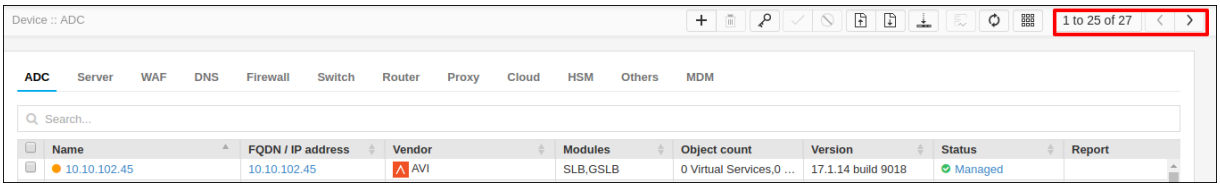
Related information
Pagination

Pagination

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Pagination is being used to display many devices into the discrete pages. So The user can configure the number of device details to be displayed in a page and he can navigate to the previous and the next page using the previous and the next icon.

1. Select Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Inventory**..
2. Hover the mouse pointer over on the device count and select the number of records to be displayed in the device inventory page.



3. Select the Previous or Next icon to move to the previous or next page.

Related information
Generating and Downloading an iHealth Report
Selecting Inventory Columns

Device Group

Device Group Addition

To add an ADC group to AppViewX:

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Group**.
   The **Group** screen opens.
2. In the **ADC** tab, click the + (**Add**) icon in the Command bar.
3. On the **Add** screen, enter the name of the new group. (Recommended) Enter a description of the group to help users identify it.
4. In the **Device selection** field, click the » (**Assign item**) icon beside each device you want to include in the group.
   When you have finished assigning devices to the group, click **Save** to add them to the system.

> **Note:**
> Rather than adding devices manually, you can click the Add search string link and create a search string that automatically assigns all existing devices that match the filter criteria to the

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

ADC group. The benefit of using a search string rather than selecting devices manually is that the search string continues to work in the background, auto-assigning all new devices to the ADC group if the devices match the search criteria you set up.

Related information
Device Group Modification
Device Group Deletion

Device Group Modification

To modify an ADC group to AppViewX:

1. Click the **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Group**.
   The **Group** screen opens.
2. If the ADC group whose details you want to modify is not displayed on the screen, use the search field to locate it.
3. In the **ADC** tab, select the checkbox beside the name of the ADC group.
4. Click the (**Modify**) icon in the Command bar.
5. On the Modify screen that appears, make whatever changes you want to the content.
6. Click **Update** to save your changes.

Related information
Device Group Deletion

Device Group Deletion

To delete an ADC group to AppViewX:

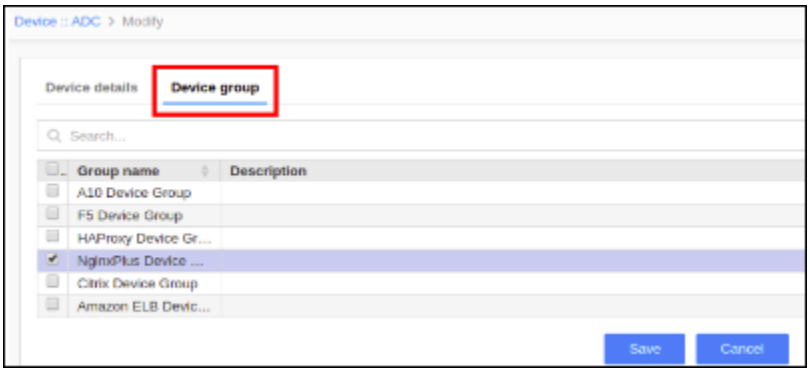
1. Click the **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Group**.
   The Group screen opens.
2. Select the checkbox beside the group you want to delete.
3. Click the (**Delete**) icon in the Command bar.
4. On the confirmation screen that pops up asking you if you are sure you want to proceed, click **Yes**.
5. The group is then removed from the AppViewX system.

Related information
Device Group Addition

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Assign a Device to Group(s)

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Group**.
   The **Group** screen opens.
2. Search for the device to be added to device group(s) and select the respective device and navigate to the **Device group** page.
3. Select the device group(s).



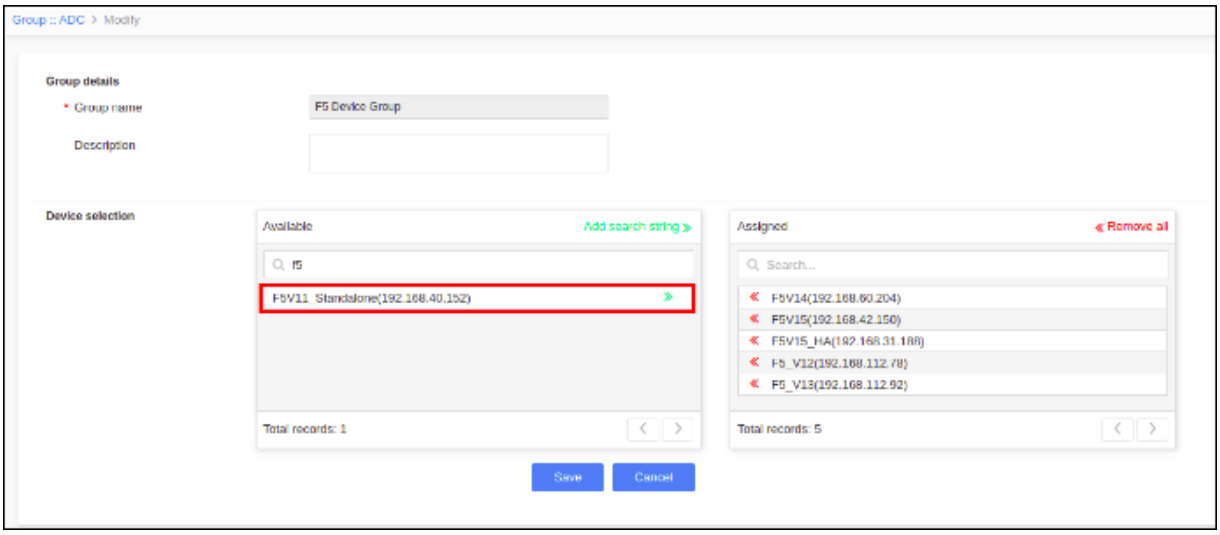
4. Click **Save**.

Related information
Assign/unassign devices
Unassign a Device from Group(s)

Assign/unassign devices

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Group**.
   The **Group** screen opens.
2. From the **Group** page, select the device group to be modified and click the **(Modify)** icon to navigate to the Group Modification page.
3. To assign devices in the group, search for the device and select the devices under the **Available** list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. To unassign devices in the group, search for the device and select the devices under the **Assigned** list.
5. Click **Save**.

Related information
Assign a Device to Group(s)
Unassign a Device from Group(s)

Unassign a Device from Group(s)

1. Go to **Menu** > **ADC+** > **ASSET MANAGEMENT** > **Device Group**.
   The **Group** screen opens.
2. Search for the device to be removed from device group(s) and select the respective device and navigate to the **Device group** page.
3. Unselect the device group(s).



4. Click **Save**.

Related information
Assign a Device to Group(s)
Assign/unassign devices

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Backup and restore

Backup and Restore

Overview

Configuring devices for Backup and Restore required the users to have added the devices whose backup is to be taken in the Inventory.

User should also have the ACF permission for Backup and restore.

Vendors Supported in AppViewX for BackUp

Vendors	BackUp	Restore
F5	Device	Device & Object
Citrix	Device	Device & Object
A10	Device	Device & Object
Avi	Device	Device

Related information
Configuring Device Backup
Restore
Configuring Backup Settings
Archive Setting Customization
Compare Backup

Configuring Device Backup

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
2. Click the **+** (**Add**) icon to navigate to the BackUp configuration page.
3. Select individual devices or device groups for BackUp.
4. Provide the required details:

Device Backup - Field and Description Table

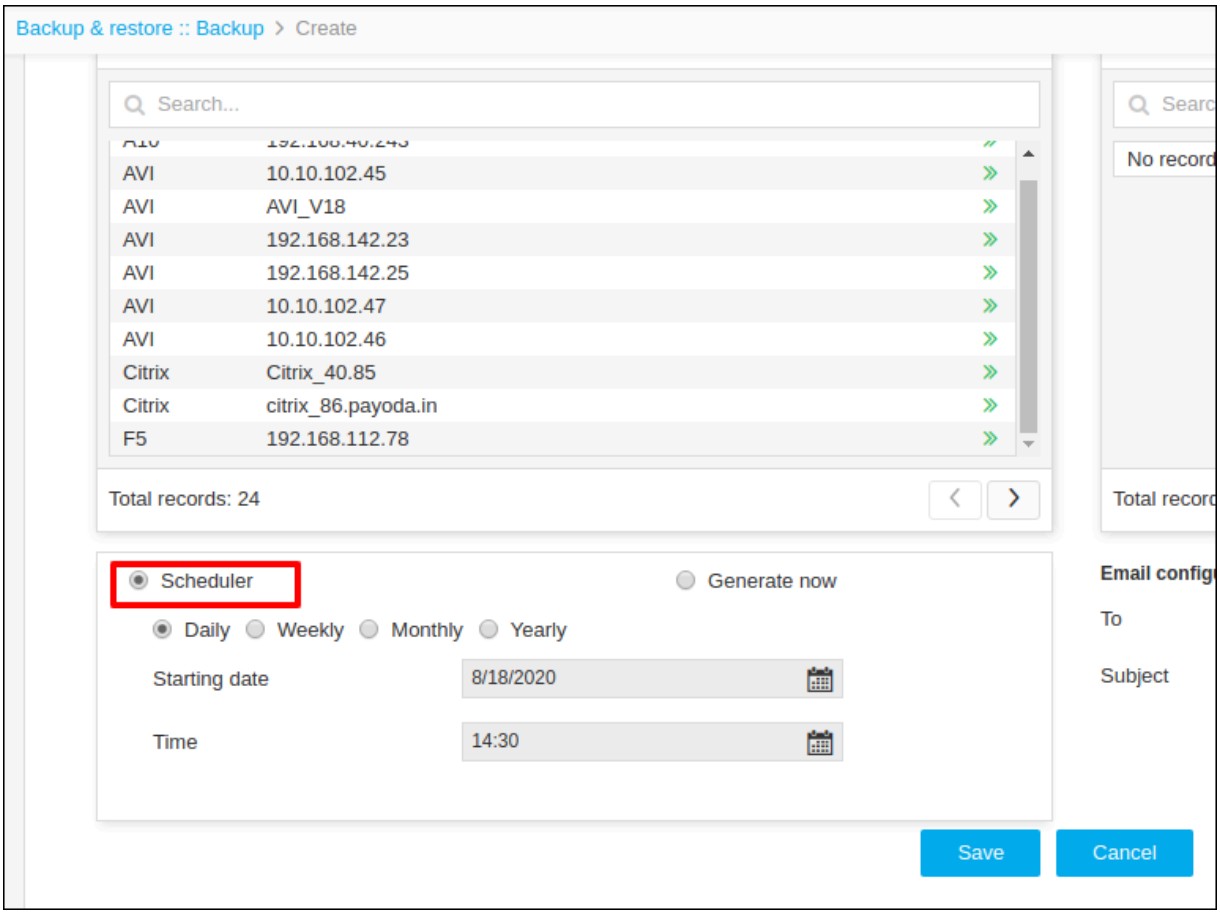
Name	Description
***Device Type**	Type of device to be selected (ADC,Proxy,WAF,Firewall)

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
***BackUp Name**	BackUp name accepts only alphanumeric characters, '-' , '_' , '.'. Also should not start or end with a special character.
Description	Type the appropriate descriptive content.
***Configure**	Select the Device or Device Group radio button.
***Device or Device Group selection**	Search and select device or device group.
***Scheduler/Generate now**	Select the Scheduler or Generate now radio button.
Email configuration to	Type the email addrese(es).
Email configuration subject	Type the subject line to be included in the email.
**: Mandatory fields*	

5. Schedule the BackUp by clicking **Scheduler** or **Generate now**.
6. Select the scheduler on Daily/Weekly/Monthly or Yearly basis and provide the date and time appropriately.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. Click **Save**.
8. To view BackUps generated in a particular group, click the **+** (**Add**) icon on the tab that has the backup group name.
9. To edit the already created BackUp group, click the  (**Edit**) icon on the tab mentioned with the backup group name.

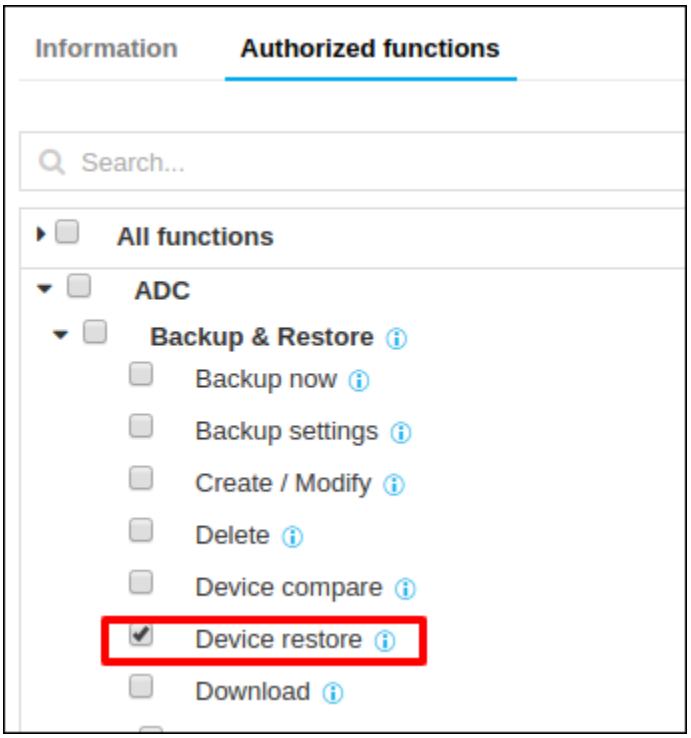
Related information
Backup and Restore
Restore
Configuring Backup Settings
Archive Setting Customization
Compare Backup

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Restore

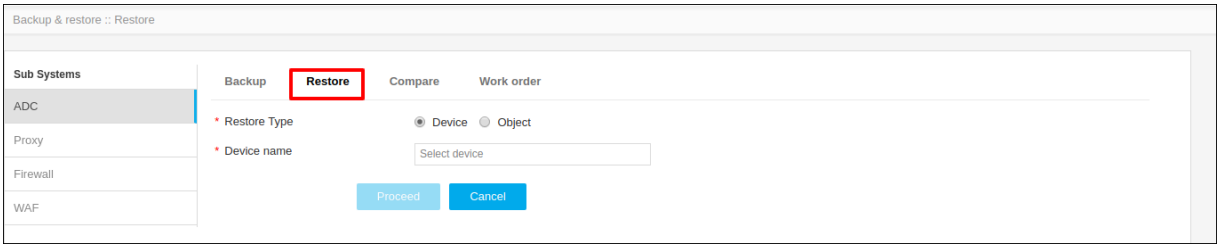
Restore from Backup

Users can restore the device/objects added in AppViewX from the Backup generated. The users should have ACF permission for Restore and ACL permission on those devices for which restore has to be performed.



Restore Device/Object with the Backup Generated

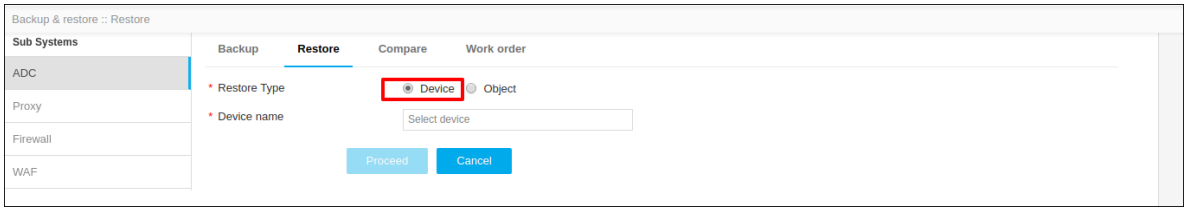
1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Restore Configuration**.
2. Click the **Restore** tab to navigate to the **Restore** page.



3. Click **Device** or **Object** that needs to be restored.

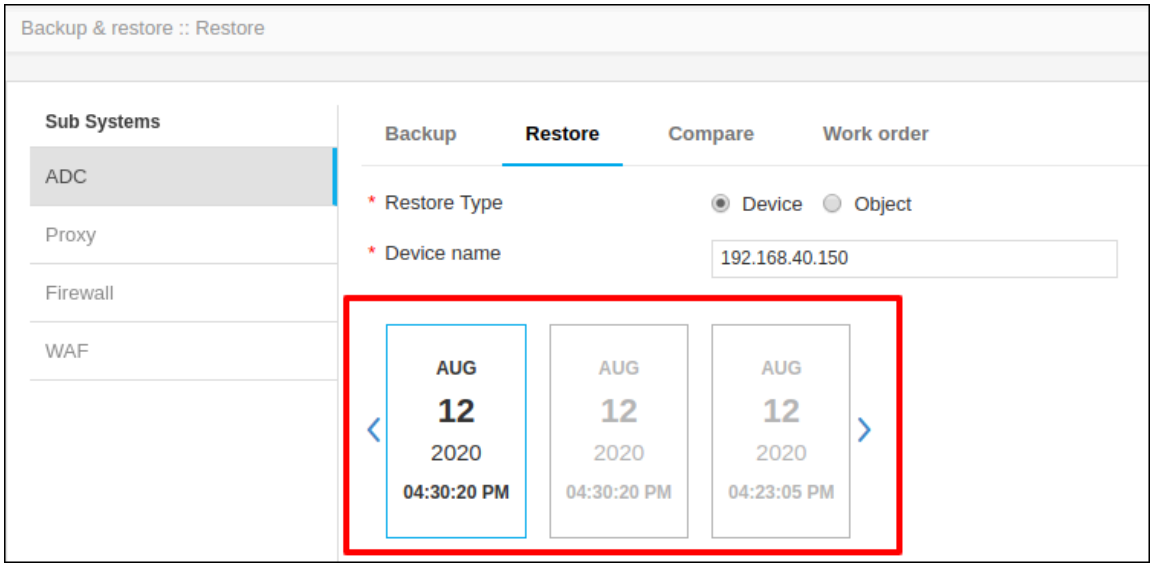
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. **Device Restore:**



i. Provide the device name whose backup is to be restored in the **Device Name** text box.

All the Backups generated for this particular device will be listed below.

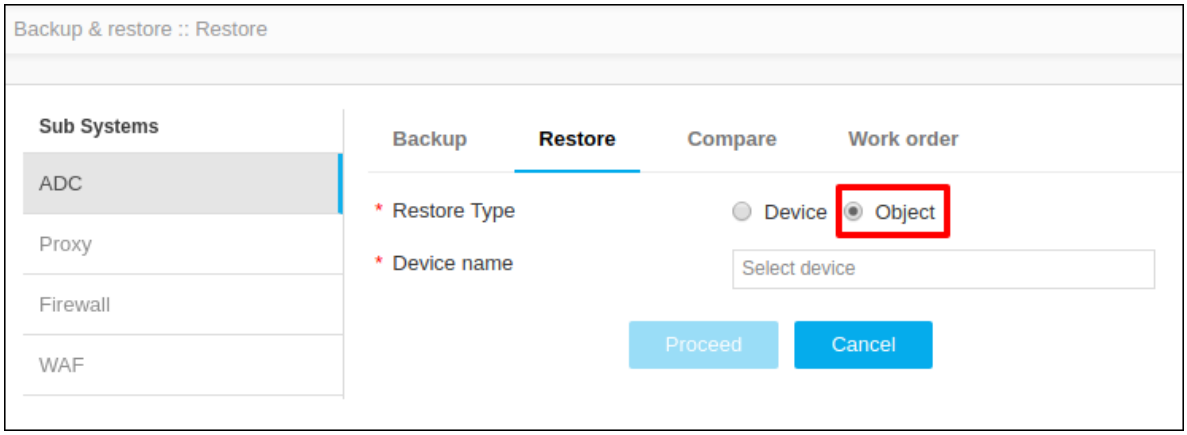


ii. Select which backup to be used and provide the device name to which the restore needs to be performed.

iii. Click **Proceed** to navigate to the next page, where the Reason for restore is to be provided.

iv. Click **Restore**.

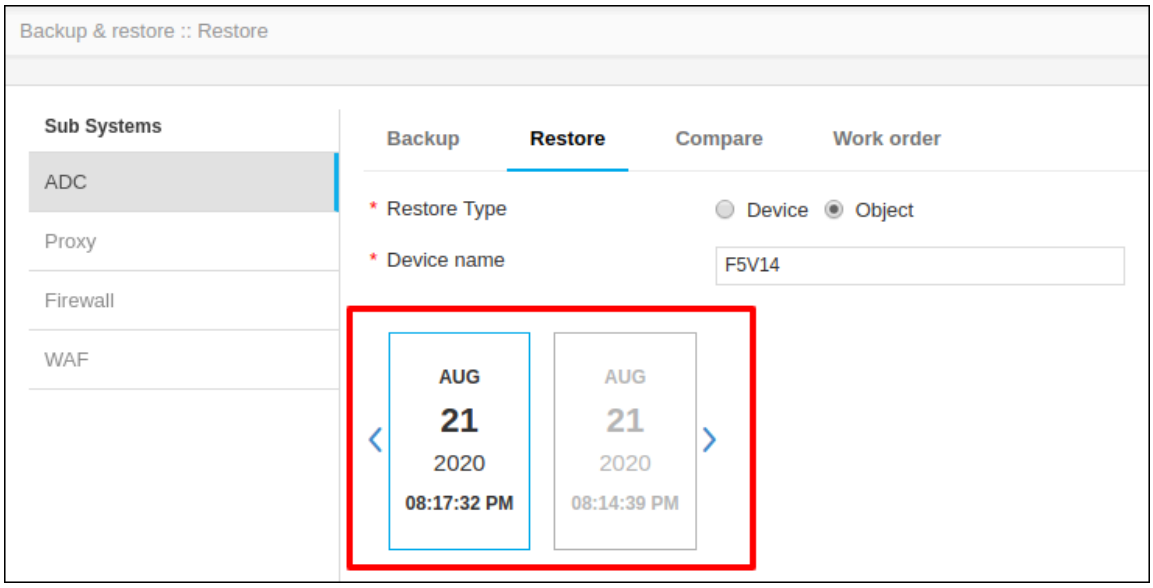
b. **Object Restore:**



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

i. Provide the device name whose object is to be taken for restore in the **Device Name** text box.

All the Backups generated for this particular device will be listed below.



ii. Select which backup to be used for restoring the object.

iii. Select which type of object is to be restored by clicking on the drop down menu for **Object Type**.

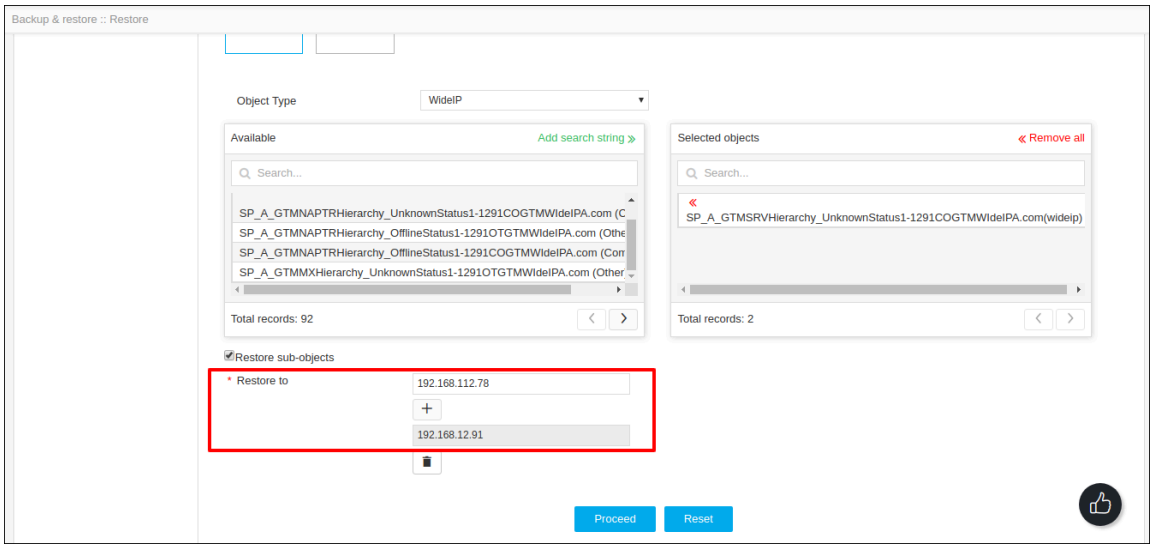
iv. All the objects pertaining to the selected object type will be listed below. Users can select which all objects to be restored by moving the objects from the available object list to selected object list.

v. If the restore needs to be performed on sub-objects as well, then the user can check the checkbox provided to **Restore sub-objects**.

vi. Provide the device name to which the back up objects need to be restored.

vii. Users can restore an object to a single/many devices. Restoring more than one device can be done by clicking **+** (**Add**) sign and providing the device name.

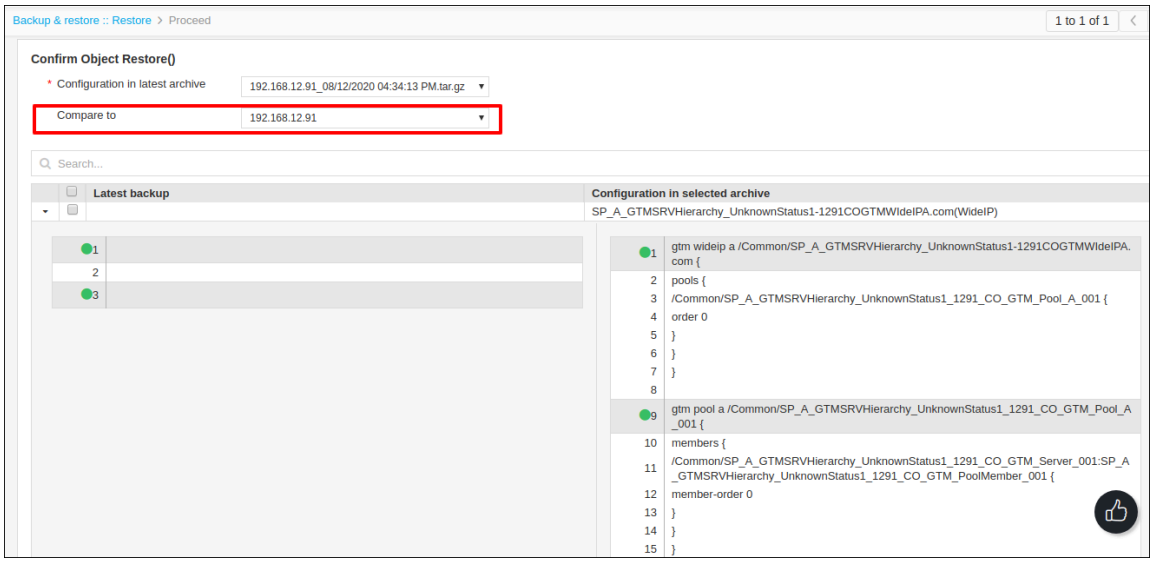
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



viii. Click proceed to navigate to the next page where Confirmation to Object Restore is requested.

ix. Check or compare the configuration of objects from the back up and the current object and cross verify the same.

x. Provide the reason for restore, and then click **Restore all**.



The restore success message is displayed on the screen once restore is successful.

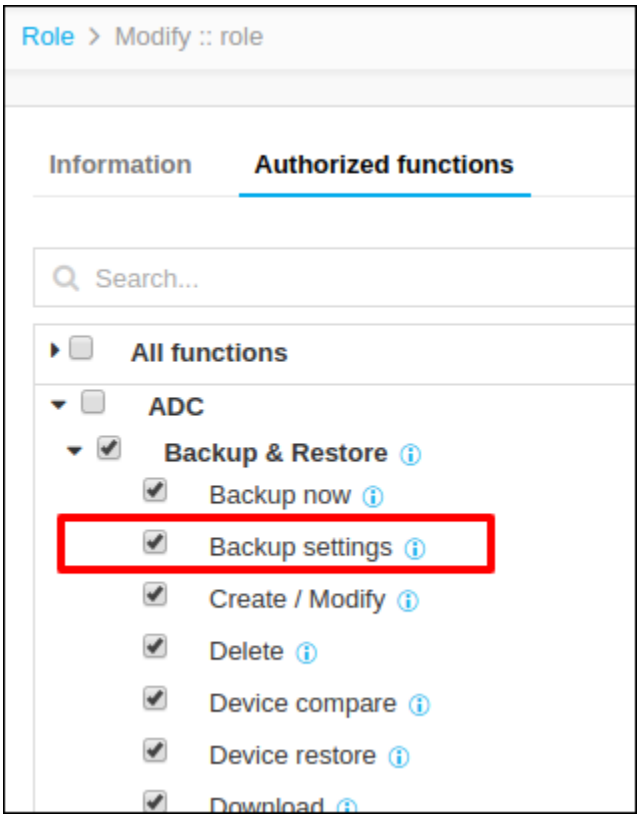
Related information
Backup and Restore
Configuring Device Backup
Configuring Backup Settings
Archive Setting Customization
Compare Backup

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Configuring Backup Settings

Configuring Backup Settings

To configure Backup & Restore settings, the user should have ACF permissions to Backup settings.

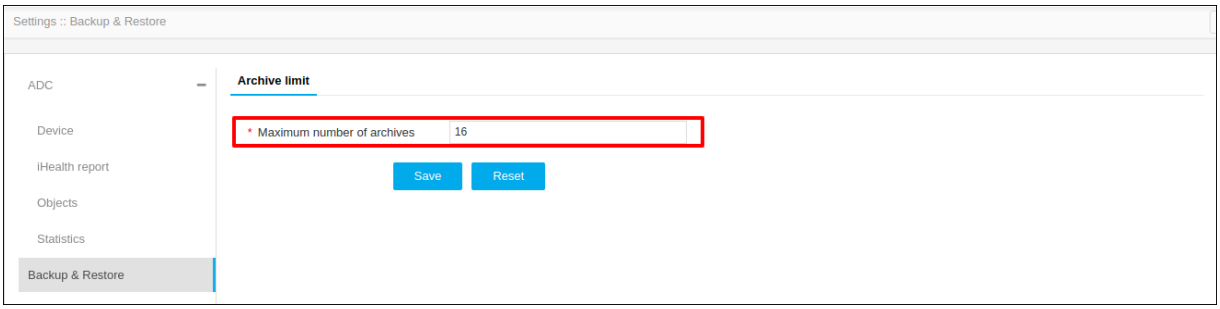


Configuring the Maximum Number of Archives Saved

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **Device** > **Backup & Restore**.
2. Enter the maximum number of archives to be stored in AppViewX in Maximum number of archives test box.

   The default value i 16.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



3. Click **Save**.

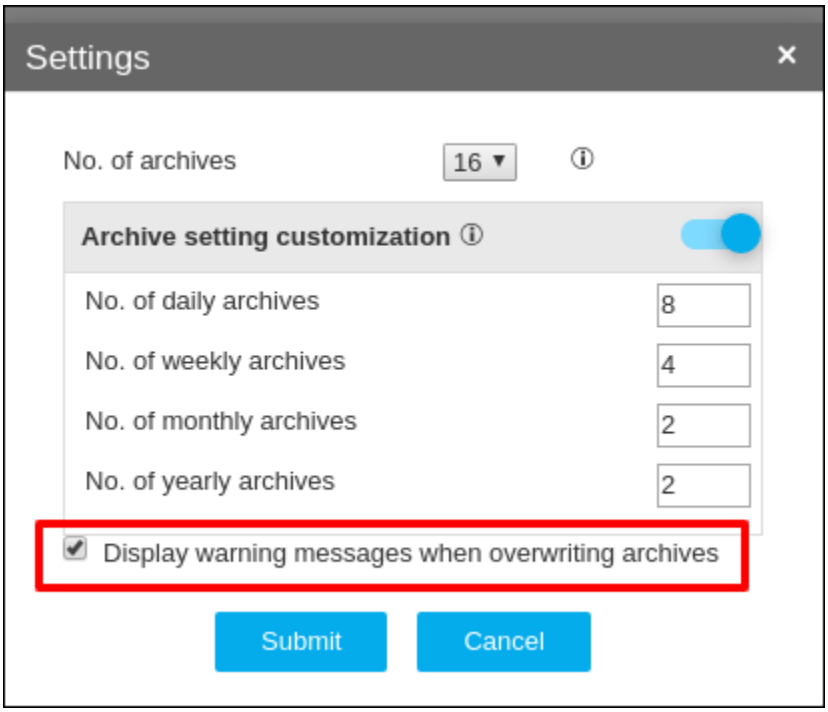
Related information
Backup and Restore
Configuring Device Backup
Restore
Archive Setting Customization
Compare Backup

Archive Setting Customization

Users can customize the number of archives stored for devices based on daily/weekly/monthly/yearly.

1. Go to **Menu** > **CONFIG MANAGEMENT** > **Backup Configuration** > **Manage Backups**.
2. Click the ⚙**Settings** icon on the top right corner.
3. Enable the **Archive setting customization** button.
4. Provide the number of archives to be stored for devices on a daily/weekly/monthly/yearly basis.
5. Enable the radio button to display the warning message when archives are overridden.
   Enabling this will show a pop up if the number of archives configured by the user causes any removal/deletion of archives.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Click **Submit**.

Related information
Backup and Restore
Configuring Device Backup
Restore
Configuring Backup Settings
Compare Backup

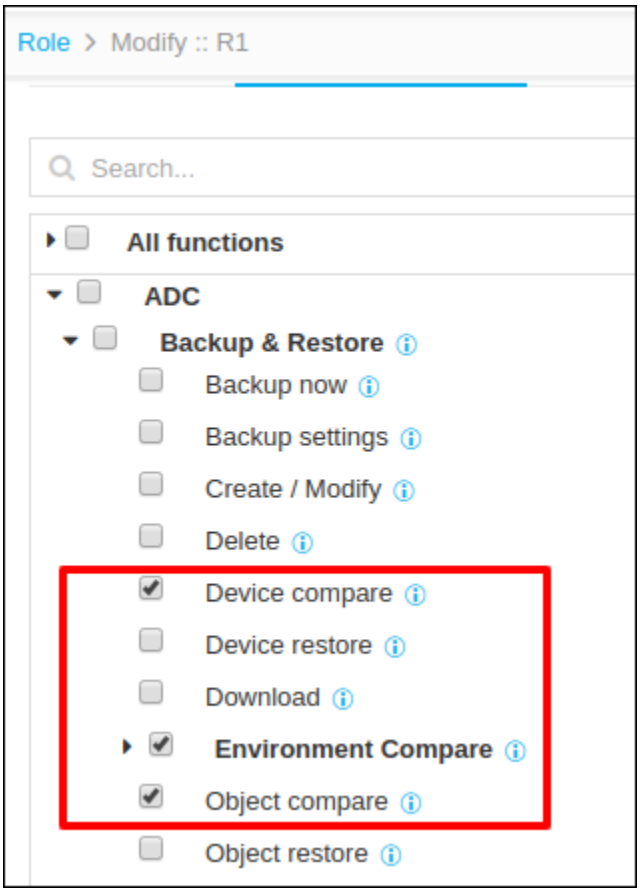
Compare Backup

Overview

Users can compare between two archives generated. They can compare between both the devices as well as objects.

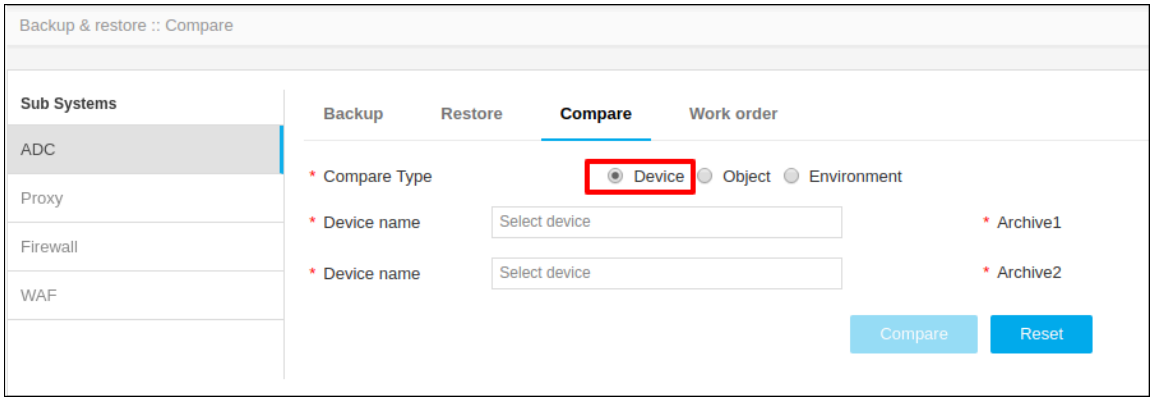
To compare the backup, the user should have ACF permissions to Device compare, Object compare and environment compare.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



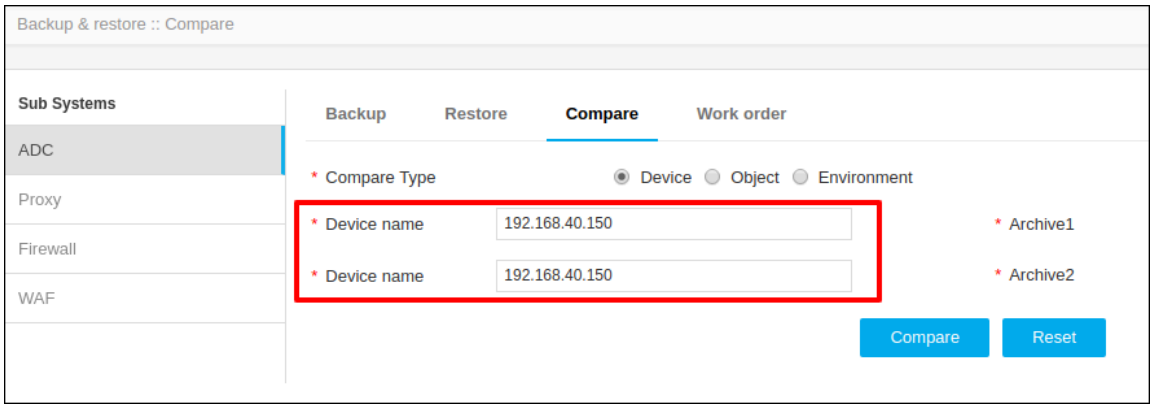
Comparing Between Two Backup Generated

1. Go to **Menu** > **ADC+** > **CONFIG MANAGEMENT** > **Compare Configuration**.
2. Click on the **Compare** tab to navigate to the Restore page.
   a. **Device compare:**
      i. Select the **Device** radio button in **Compare Type**.



      ii. Provide the device names across which the backup is to be compared.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

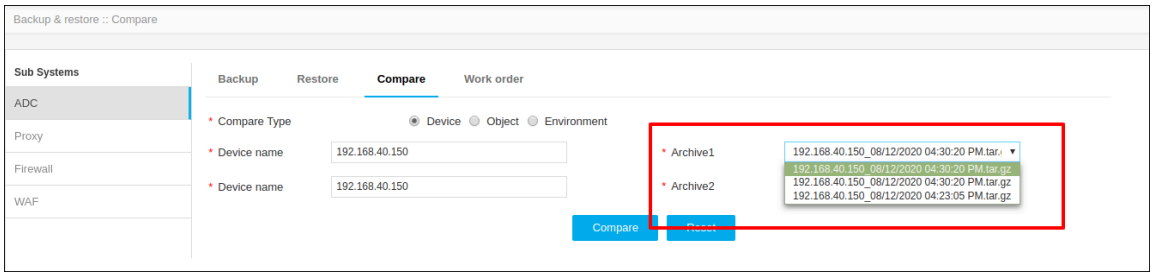


Note:

The backup compared either between the same devices or different devices as well.

iii. After the device is selected, the backups generated for that particular device will be brought up in the archive list.

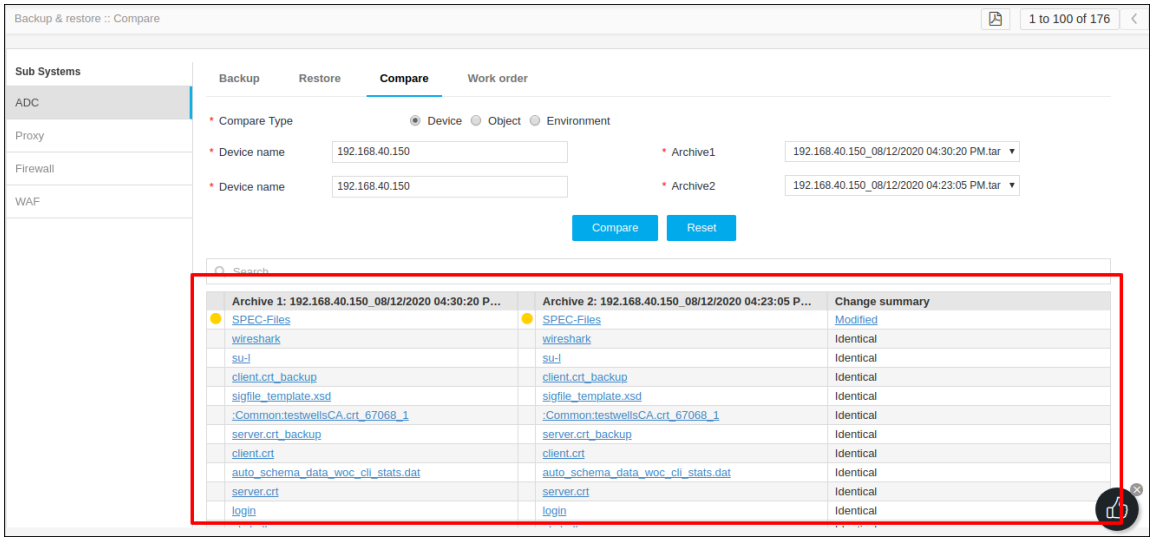
iv. Select which archive is to be compared from the drop down list.



v. Click **Compare**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The two archives along with the change summary will be displayed in the same page below.



b. **Object compare:**

i. Select the **Object** radio button in **Compare type**.

ii. Select the **Vendor** and **Object type** to be compared from the drop down menu.

iii. Select the object name to be compared from the drop down menu of **Object Name**. This will automatically populate the device name tab corresponding to this object.

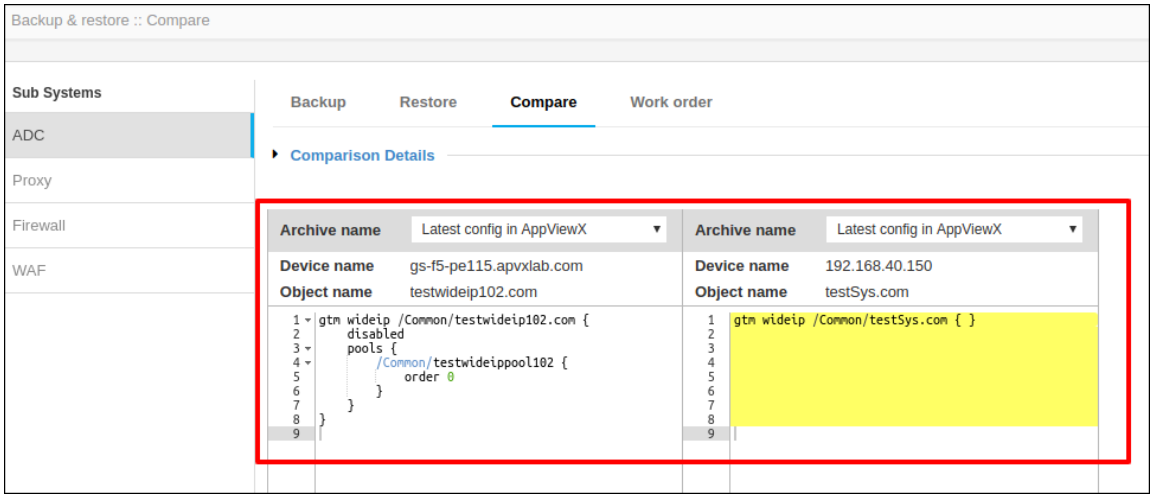
iv. From the drop-down menu of Archive names, select which archive to be compared. Users can either select the latest archive generated or select any of the archives generated earlier for the object.

v. Similarly users can add other objects to be compared.

vi. Users can add more than two objects to be compared by clicking the **+** (**Add**) icon near the Archive names. Users can add maximum of five objects to compare.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

vii. Click **Compare**.



Role Based Access Control

Before You Begin

To configure role based access control, the user should have ACF permissions.

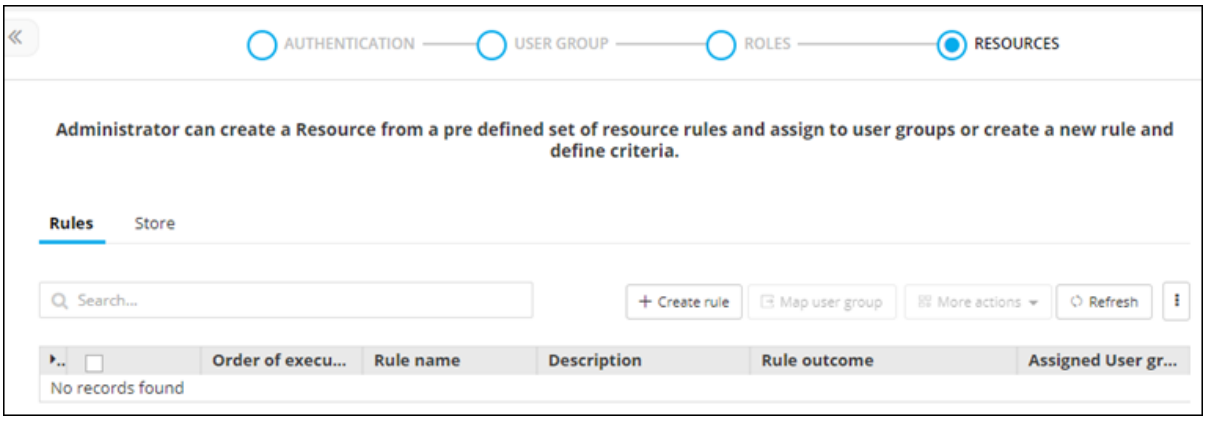
Assign Devices to Resource

1. In the new menu, go to **Menu** > **Platform** > **ACCESS MANAGEMENT** > **Quick Config**.

> **Note:**
> In the old menu, go to **Menu** > **Settings** > **General** > **Authentication**, and then click **Quick Config**

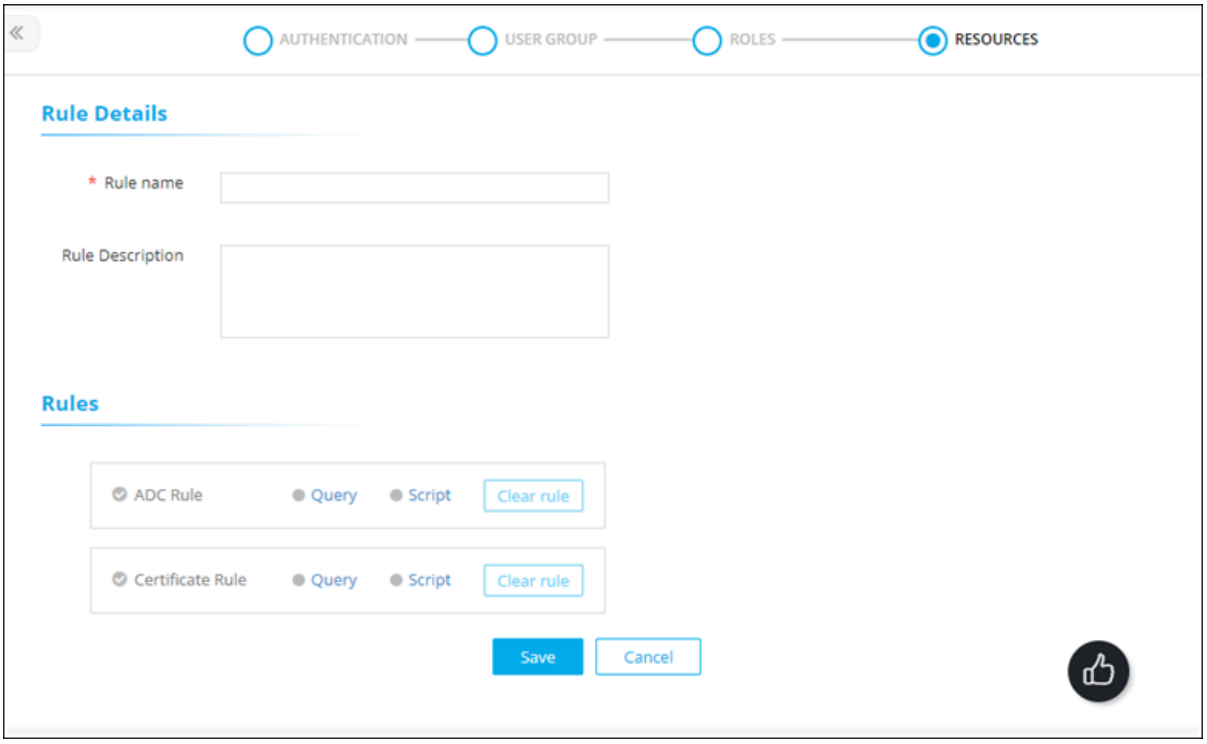
2. On the **RBAC Journey :: Authentication** page, select **RESOURCES** radio button within wizard flow to add device to the resource.
3. In the **Rules** tab, click + Create rule.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4.

The **Rules :: Create** screen is displayed.



5. In the **Rule Details** section, enter the following details:

Rule Details - Field and Descriptions Table

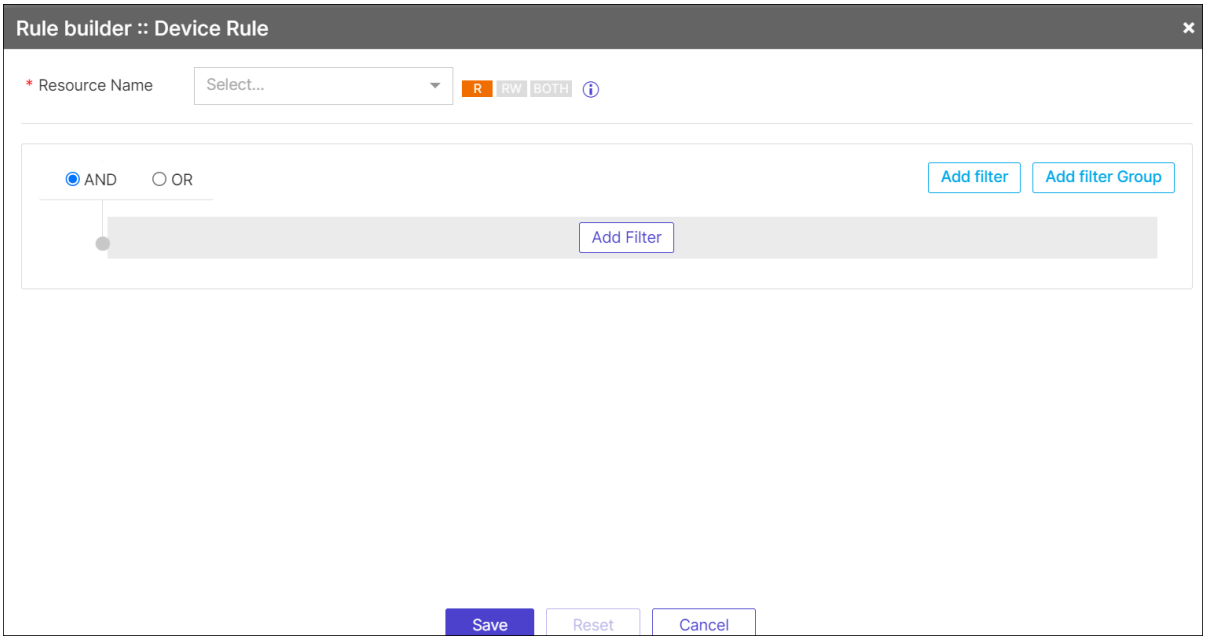
Field	Description
***Rule Name**	Resource name.
Rule Description	Brief description of the resource/granular level accesses associated with the resource.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description

Note:
**: Mandatory fields*

6. To configure a rule to dynamically tag ADC objects using Query, select **Query** in the **Rules** section, for the **Device Rule**.
   The **Rule builder :: Device Rule** action pane is displayed.



7. Select the Resource Name from the drop-down list, R/RW permission, and then AND or OR condition for monitoring the dynamic tagging of ADC devices based on rule condition.
8. Click **Add Filter**.
9. Select the field information from the drop-down list.
10. (Optional) Add filter group by clicking **Add filter Group** and then select the field information from the drop-down list.
11. Click **Save**.

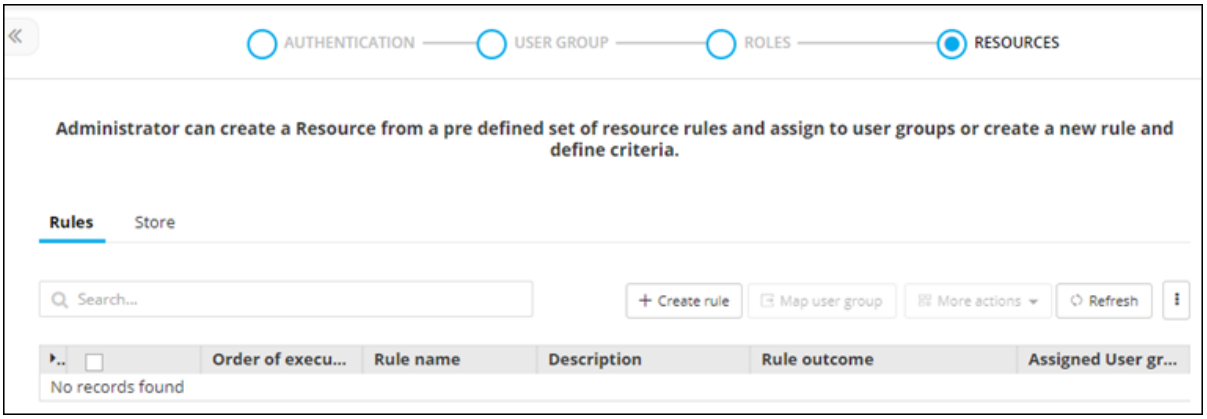
Assign Objects to Resource

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

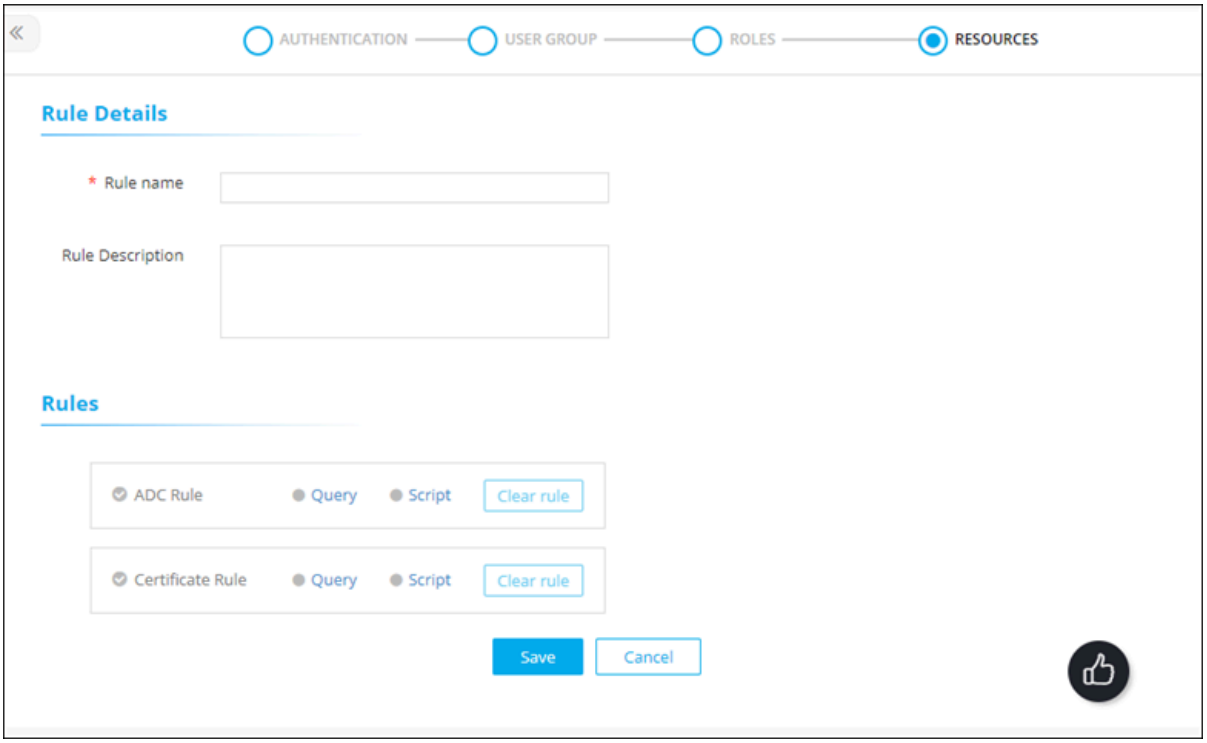
1. In the new menu, go to **Menu** > **Platform** > **ACCESS MANAGEMENT** > **Quick Config**.

> **Note:**
> In the old menu, go to **Menu** > **Settings** > **General** > **Authentication**, and then click **Quick Config**

2. On the **RBAC Journey :: Authentication** page, select **RESOURCES** radio button within wizard flow to add device to the resource.
3. In the **Rules** tab, click + Create rule.



The **Rules :: Create** screen is displayed.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

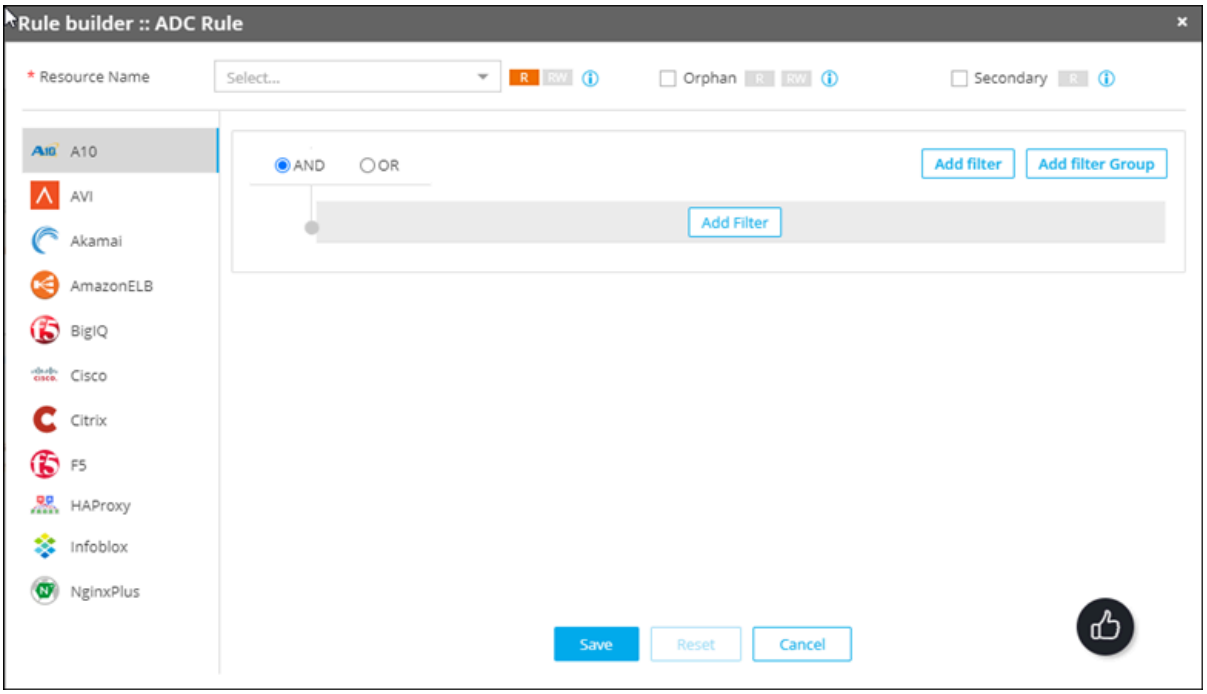
4. In the **Rule Details** section, enter the following details:

Rule Details - Field and Descriptions Table

Field	Description
***Rule Name**	Resource name.
Rule Description	Brief description of the resource/granular level accesses associated with the resource.

Note:
**: Mandatory fields*

5. To configure a rule to dynamically tag ADC objects using Query, select **Query** in the **Rules** section, for the **ADC Rule**, .
6. The **Rule builder :: ADC Rule** action page is displayed.



7. On the **Rule builder :: ADC Rule** action pane, click **Add Filter**.
8. Select the field, condition and enter the value to be monitored for dynamic tagging of ADC objects based on rule condition.
9. Click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Control Center

Control center module is a centralized object repository which allows the user to manage and monitor the objects. To manage and monitor the objects AppViewX provides some basic action in right click of object in control center. The objects will be categorized into three types based on which the right click actions may vary.

- **Primary objects** - Objects that are used in load balancing. These objects have access permission at individual level. For example in F5 objects like wide ip, pool, pool member, server are considered as primary objects.
- **Orphan objects** - Objects without parents will be considered as orphan objects. For orphan objects AppViewX allows the user to provide access in global level not at individual level. For example, an F5 pool that is not associated with wide ip will be considered an orphan pool.
- **Secondary objects** - Objects that are associated with primary objects but which do not directly participate in load balancing are called secondary objects. AppViewX provides global level access permissions for secondary objects. For example monitors, profiles, etc., Where monitors are used to monitor the health of the servers.

Related information
Searching for Objects

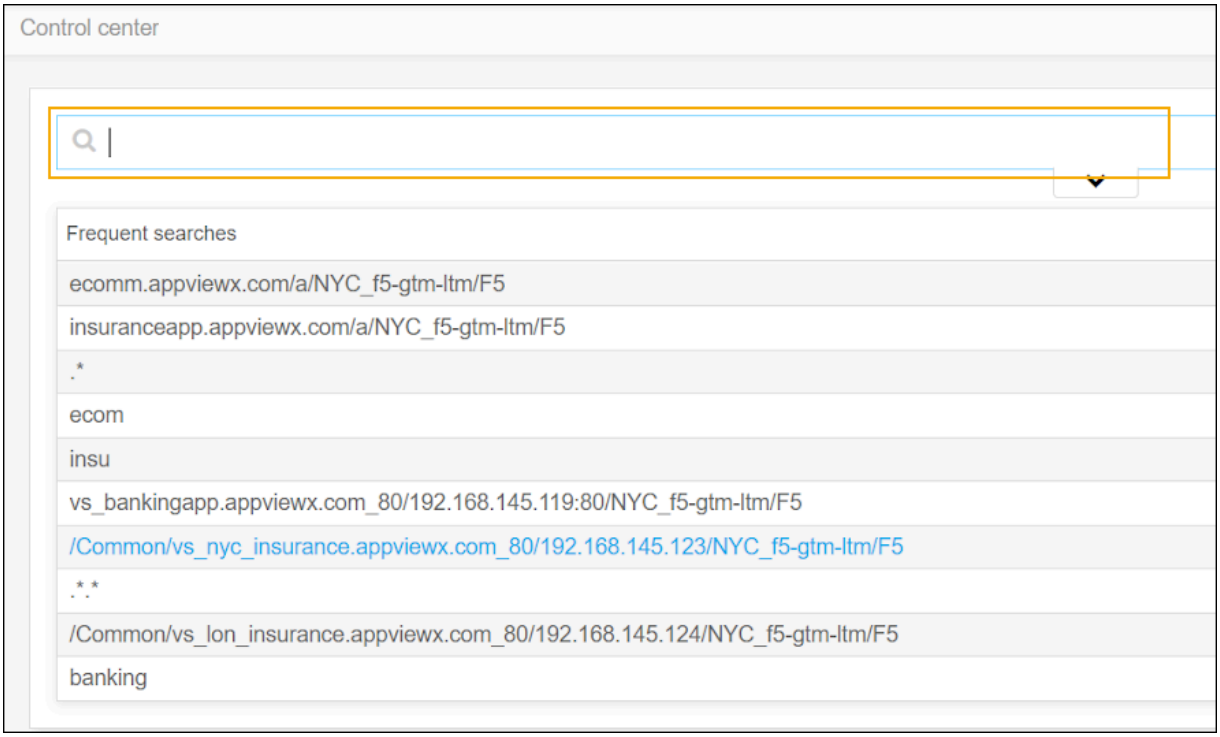
Searching for Objects

AppViewX allows users to search for objects in the control center module. The search results that are matched with the text entered will be highlighted.

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**..
   When the user clicks on Control Center, it defaults to the search screen.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

2. Enter a search query in the search bar and click the enter button on the keyboard.



Before you begin

- Devices that the user wants to control should be added to AppViewX.
- Users must have the ACF permissions to access the Control Center. Users should also have the object level access (ACL) in the resources.
- To perform actions on objects users should have proper ACF permission and ACL permission.

Search Using Free Text Entries

This is the most common type of search. You can enter text in the search field and press Enter on your keyboard.

Related information
Search Using Frequent Search Links
Search Using Predefined Search Keys

Partial Text Search

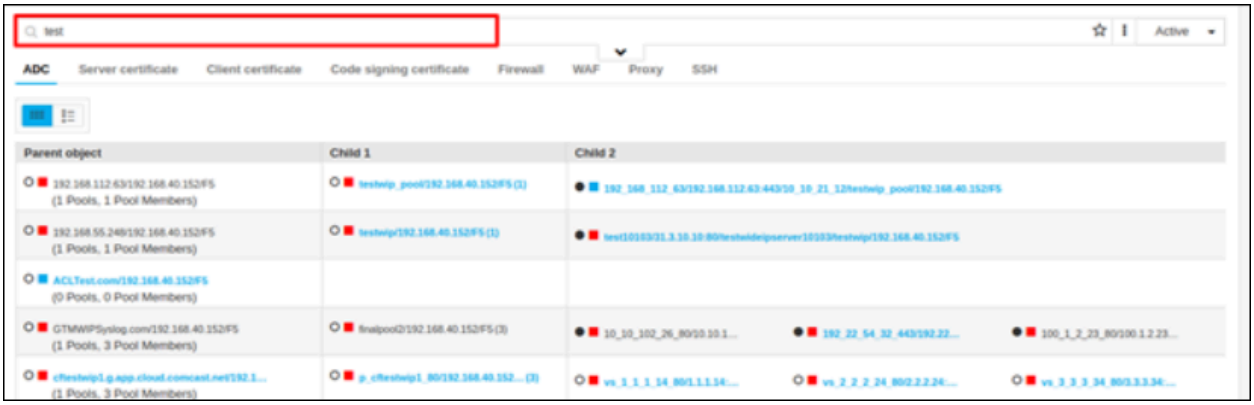
In a partial text search, users can directly input plain text, regardless of the case, into the search box. When a user enters plain text, AppViewX will retrieve all objects that contain the entered text.

For example, search for text "test".

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The corresponding result could be:

- test
- testwideip.com
- wideip.test.com
- pool_member-Test
- Test_virtual_server



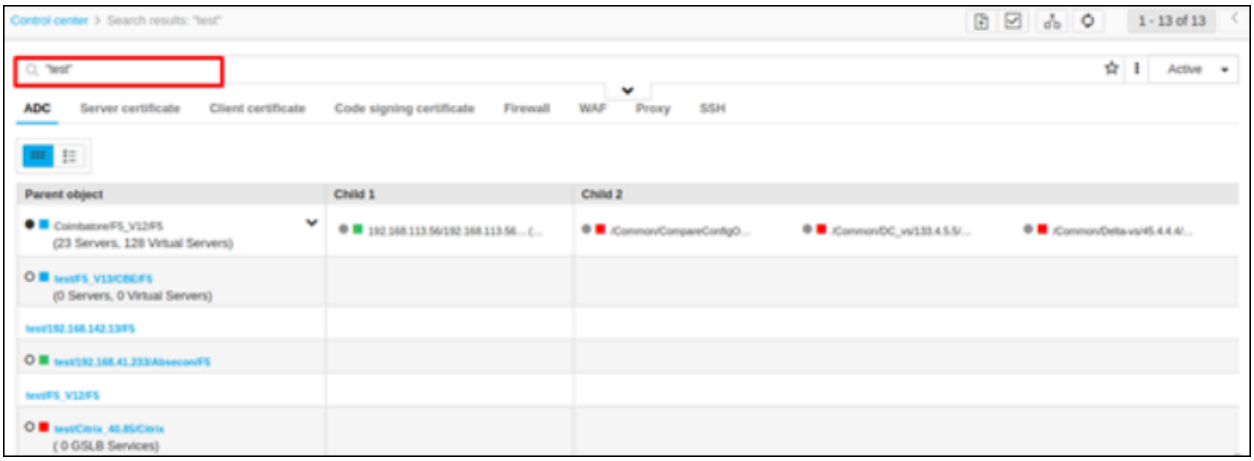
Exact Search

To perform an exact match, users should enclose the search strings within double quotation marks (""). When users conduct this type of search, AppViewX will retrieve objects that precisely match the given string, considering the case of the letters.

For example, search for text "test".

The corresponding result could be: test

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Search Using Regular Expression

Using .* AppViewX allows the user to do a regex search.

Sample Regex - Table

Regex	Description
*	All the objects will be displayed.
<String>.*	Object names starting with the entered string will be fetched and displayed.
.*<String>	Object names ending with the entered string will be fetched and displayed.
<String>.*<String>	In the result object names start and end with the entered string and in the middle part it may contain anything.

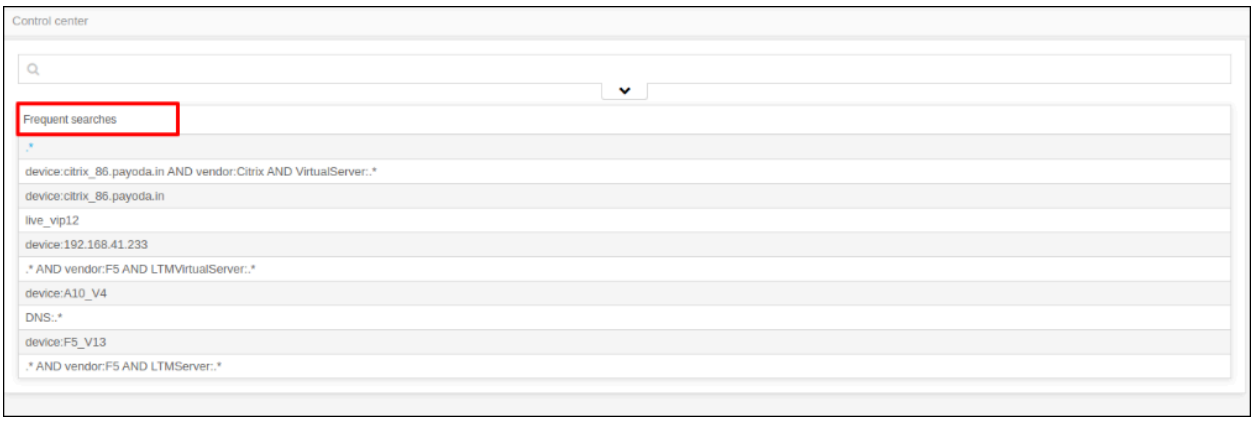


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Search Using Frequent Search Links

The App Search screen displays a list of frequent searches immediately below the search field. Click any of the items in the Frequent searches list to search the AppViewX platform for that word, phrase, or character set.

In the ADC search results screen, click the vertical ellipse button next to the search bar and select **Frequent Searches** to view a list of frequent searches.



Related information
Search Using Free Text Entries
Search Using Predefined Search Keys

Search Using Predefined Search Keys

AppViewX provides a set of predefined keys specific to each ADC vendor to help the search operation. The types of predefined keywords are:

- All keywords
- Vendor-specific keywords
- Logical operator with keywords

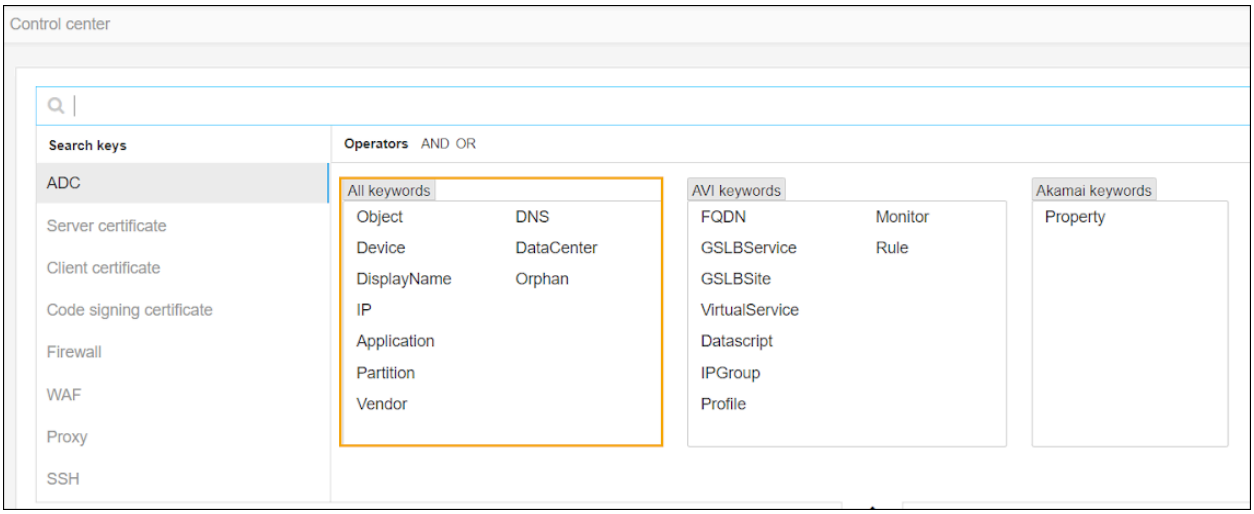
The predefined search keywords can be used to create highly targeted search queries. To create a search query, see create a search query.

Related information
Search Using Free Text Entries
Search Using Frequent Search Links

All Keywords

These keywords are generic to all ADC vendors and help searching objects irrespective of vendors.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Sample Search Keywords

Common Keywords - Table

Common Keywords	Description
Object	Search for objects using the name.
Device	Search for objects from a particular device managed in the ADC inventory.
Displayname	Search for objects using the Displayname. Displayname represents the naming format of your objects that needs to be followed throughout the application. Refer <displayname section>
IP	Search for objects using the IP address. Eg: Objects like Virtual Servers, Pool members that belong to a particular IP can be searched.
Application	Search for objects associated with an Application name. AppViewX tags all the hierarchical objects of an Application using the WideIP name as a tag during every config fetch.
Partition	Search for objects from a device partition or tenant or context.
Vendor	Search for objects from a particular vendor (F5, AVI, etc.,)

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Common Keywords - Table (continued)

Common Keywords	Description
DNS	Search for objects using their DNS name. AppViewX performs a scheduled (twice a day) reverse lookup for the objects that contain an IP address in it and persists the DNS name of the objects.
Datacenter	Search for objects from a particular Datacenter. Datacenter name information should have been provided at the time of device addition.
Orphan	Search for objects that do not have Parent association. Objects like GTM/LTM pools that are not associated with WideIP/Virtual Server will be considered as Orphans.

Logical Operator with Keywords

The common keywords and Vendor specific keywords can be combined with any Logical operator.

Users also have the option to add multiple search conditions using boolean operators. AppViewX supports both the AND and OR boolean operators.

- AND search results will contain all the terms or meet all the conditions of the search query. For example, a search for ‘test AND server’ retrieves all objects whose display name contains both test and server.
- OR search results will contain any one/all the terms or meet any one/all the conditions of the search query. For example, a search for ‘test OR server’ returns all objects whose display name contains either test,server or both test and server.
- If both AND and OR operators exist in the same search query, by default AND operator will be executed first.
- When a search query includes parentheses, the components within the parentheses will be executed first, followed by the components outside the parentheses.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Sample Keywords

Sample Keywords - Table

Keywords	Description
virtual server:* AND device: F5_v11	The list of virtual servers available in the F5_v11 device.
device: F5_v11 AND virtual server:* AND IP:192.168.96.101	The list of virtual servers in the F5_v11 device that uses the IP address 192.168.96.101 is identified.

Create a Search Query

To create a search query:

1. On the **Search** screen, click the  (**Expand**) tab at the bottom of the search field.
2. The field expands to display a list of search keys available for the ADC.
3. For each metadata type, you add to the search query, enter a value or partial value to search for. If you enter no text after the search key, the search engine automatically searches for "all values."
4. (Optional) Click the AND and OR operators at the top of the Search keys field to create Boolean searches.
5. When you are finished creating the search query, click inside the search field, then click Enter on your keyboard to run the search.

Control Center Search Result Views

When the user does a search AppViewX gives the result in a different view. Although each view displays the search results in a different manner, the search query used is the same.

1. Select **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
   When the user clicks on **App Search**, it defaults to the search screen.
2. Build a search query and click the enter button on the keyboard.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

AppViewX supports 3 types of views in the control center.

- Grid view
- Infrastructure view
- Topology view

Grid View

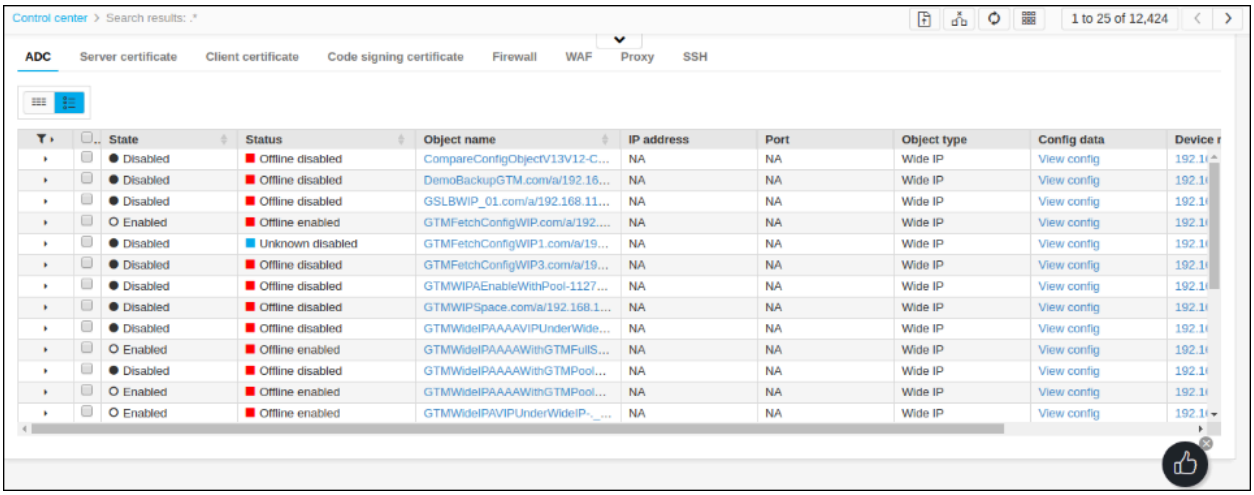
This serves as the default search landing page, showcasing fundamental details of the object alongside the corresponding hierarchy. When the search results encompass more than one record, they will consistently be presented in the grid view. In this view, the results are organized in a table format, with parent objects featured in the left column, and associated first-level and second-level children displayed in subsequent columns.



Infrastructure View

Within the infrastructure view, only the ADC objects matching your search criteria will be presented, without displaying their hierarchy (e.g., parent, child1, child2). The search results will automatically appear in the grid view. To go to the infrastructure view, click on the Infrastructure view button from the grid view.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Users can customize the displayed columns as per the requirement in infrastructure view. To customize the displayed columns, follow the below steps:

1. Click on the columns button.
2. Select/unselect the columns by checking/unchecking them as per the requirement.
3. Drag and drop the column names to change the order of the displayed columns.
4. If the user wants to reset the changes click on Reset to the previous column selection.
5. Click **Save**.

Infrastructure view provides a few more options to the user to fine tune the results.

- **Filters:** When the user clicks on the Filter button in the first column, AppViewX displays the filters for all the columns. Users can search/ filter the objects as per their requirements using these filters.
- **Change the display order (Sorting):** AppViewX default displays the result in a predefined order. To change the display order of the records, click on the Change order button near the column header. Users can switch to ascending and descending order using this button.

Topology View

This view is specifically relevant to primary objects. AppViewX offers a comprehensive, hierarchical structure map of the object, including properties such as ratio, order, load balancing method, and more. In the topology view, users can perform actions like updating the ratio and load balancing methods.

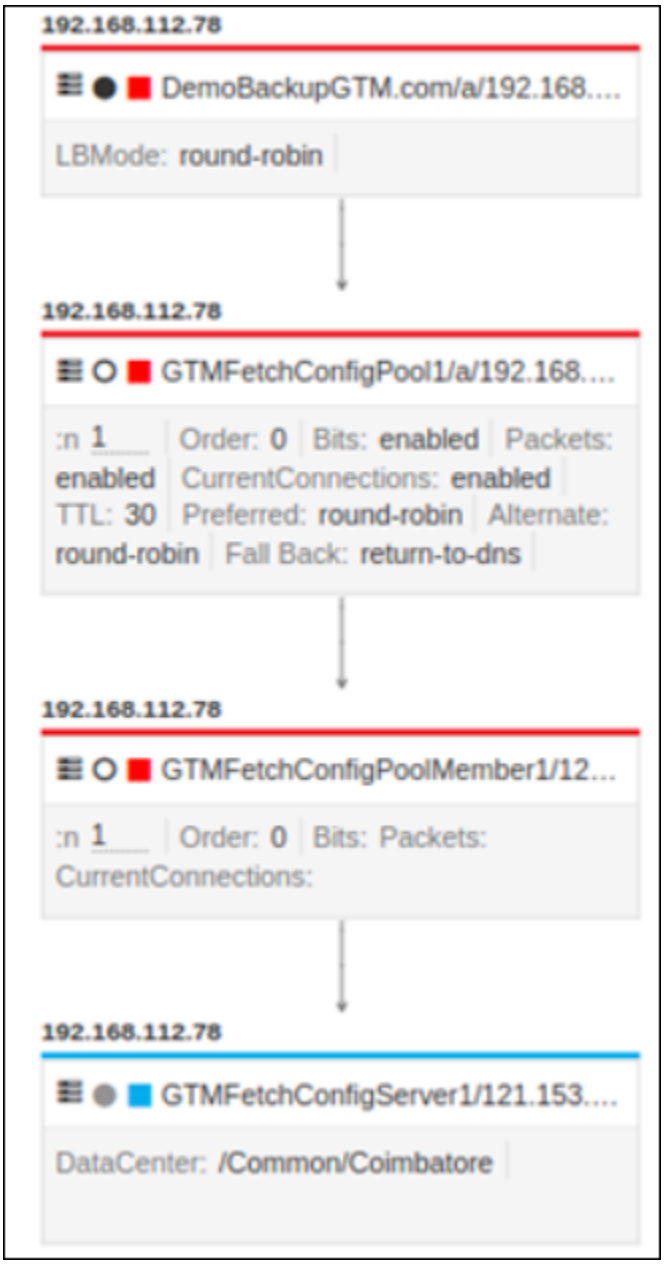
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- When the search result contains only one record then it will default render in the topology view.
- When the users click on the primary object's name from the grid view or infrastructure view.

The following are the sample ADC topologies supported in the control center.

Wide IP Topology

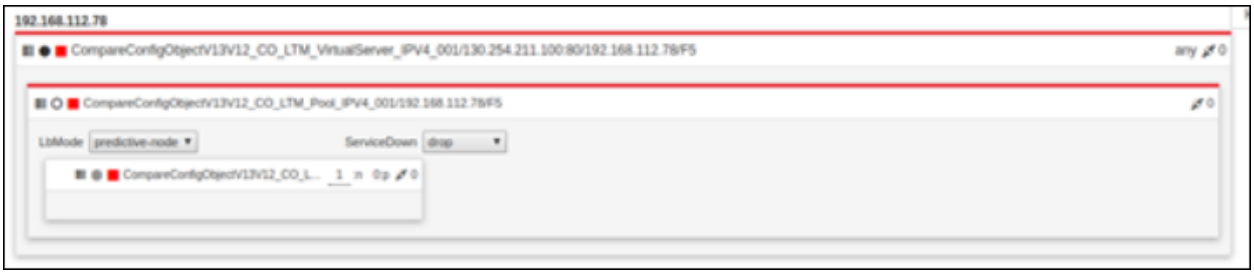
This is the topology structure defined for Global Traffic Manager (GTM) or Global Server Load Balancing (GSLB) objects in AppViewX.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

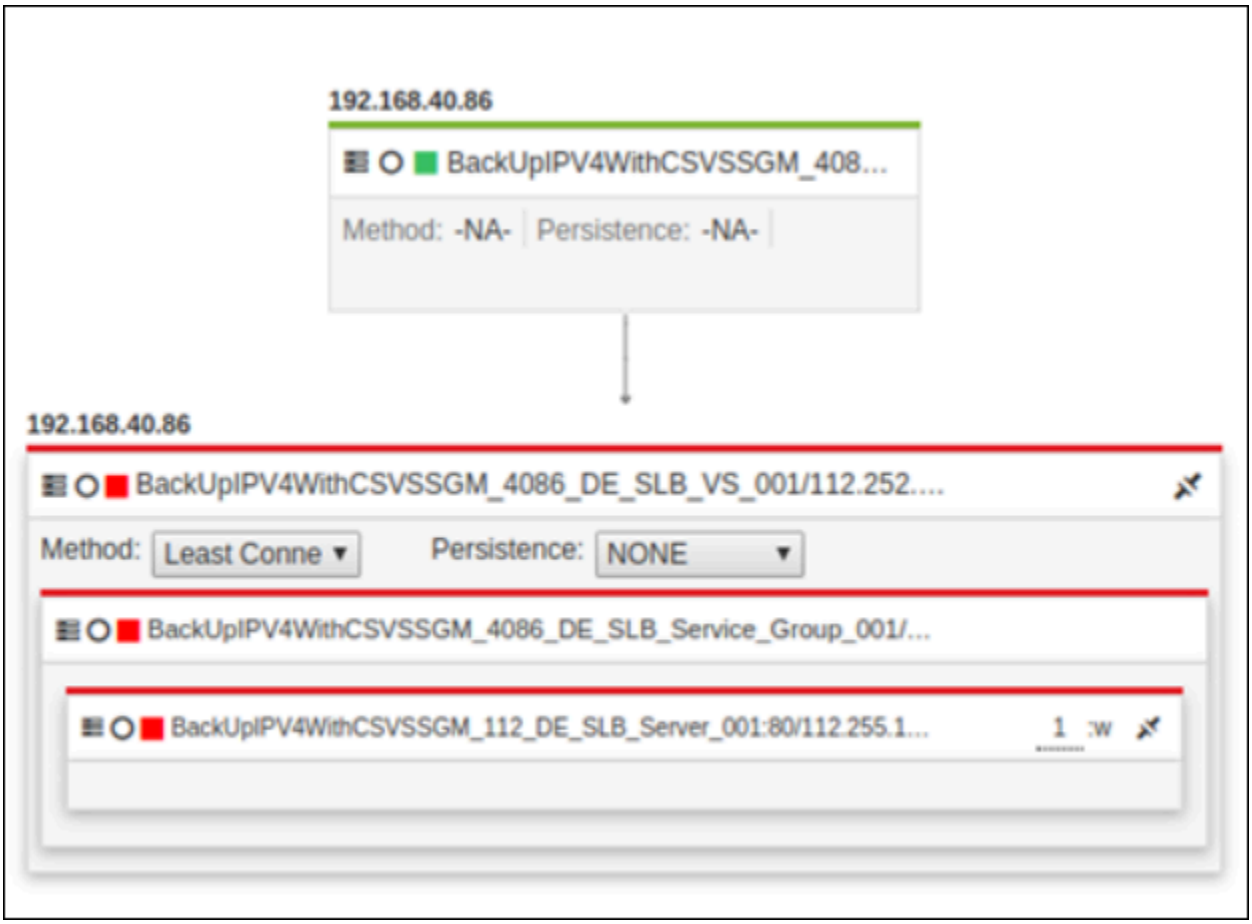
Virtual Server (VIP) Topology

This is the topology structure defined for Local Traffic Manager (LTM) or Server Load Balancing (SLB) objects in AppViewX.



Content Switching (CS) Virtual Server Topology

This is the topology structure of content switching virtual server objects. This object is specific to the Citrix vendor.

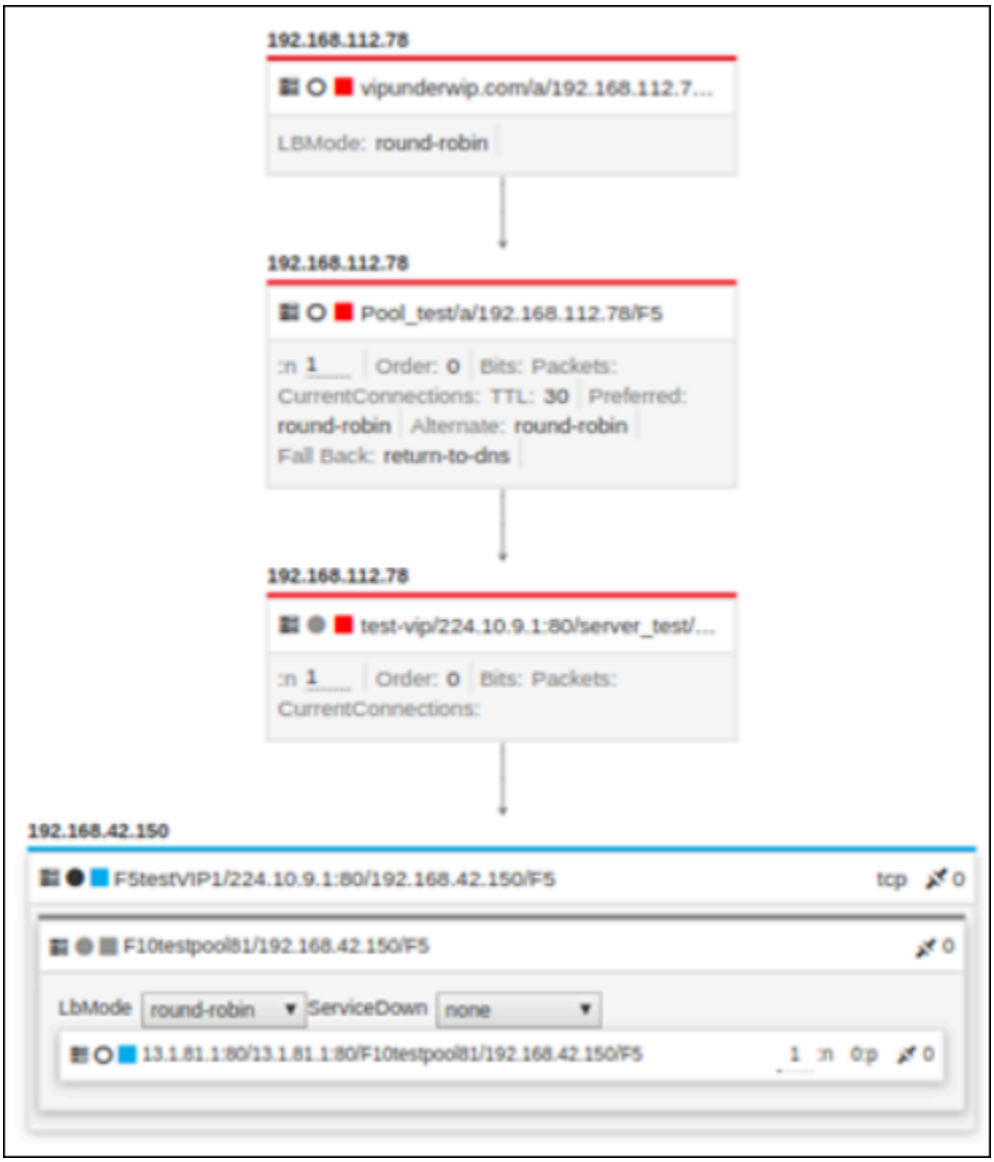


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

VIP Under Wide IP Topology

This is the topology structure displayed when an LTM/ SLB object is mapped under the GTM/GSLB.

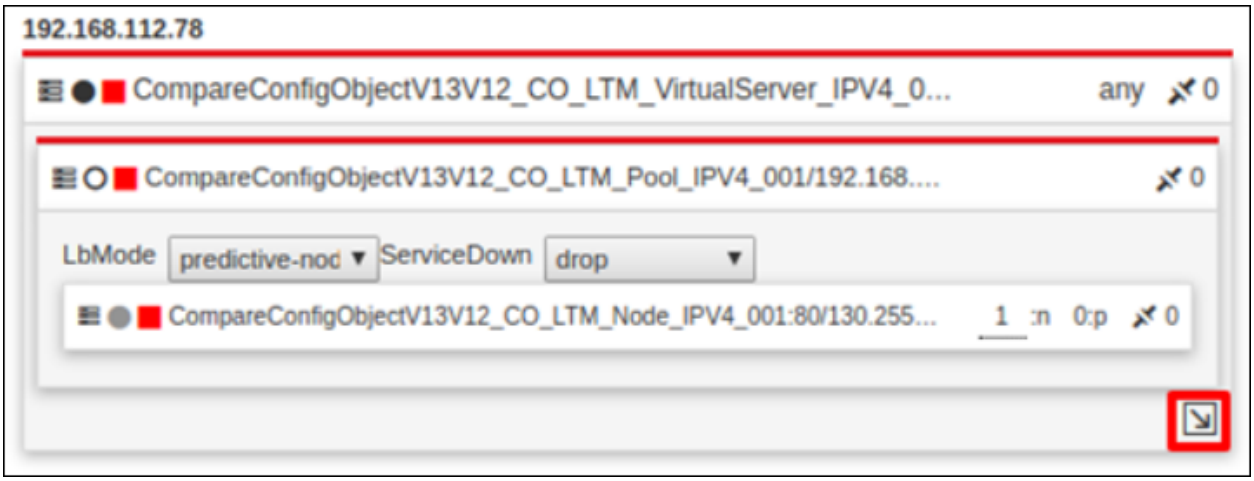
Note:
The wide IP and the VIP may belong to the same /different devices of different vendors.



VIP Under VIP Topology

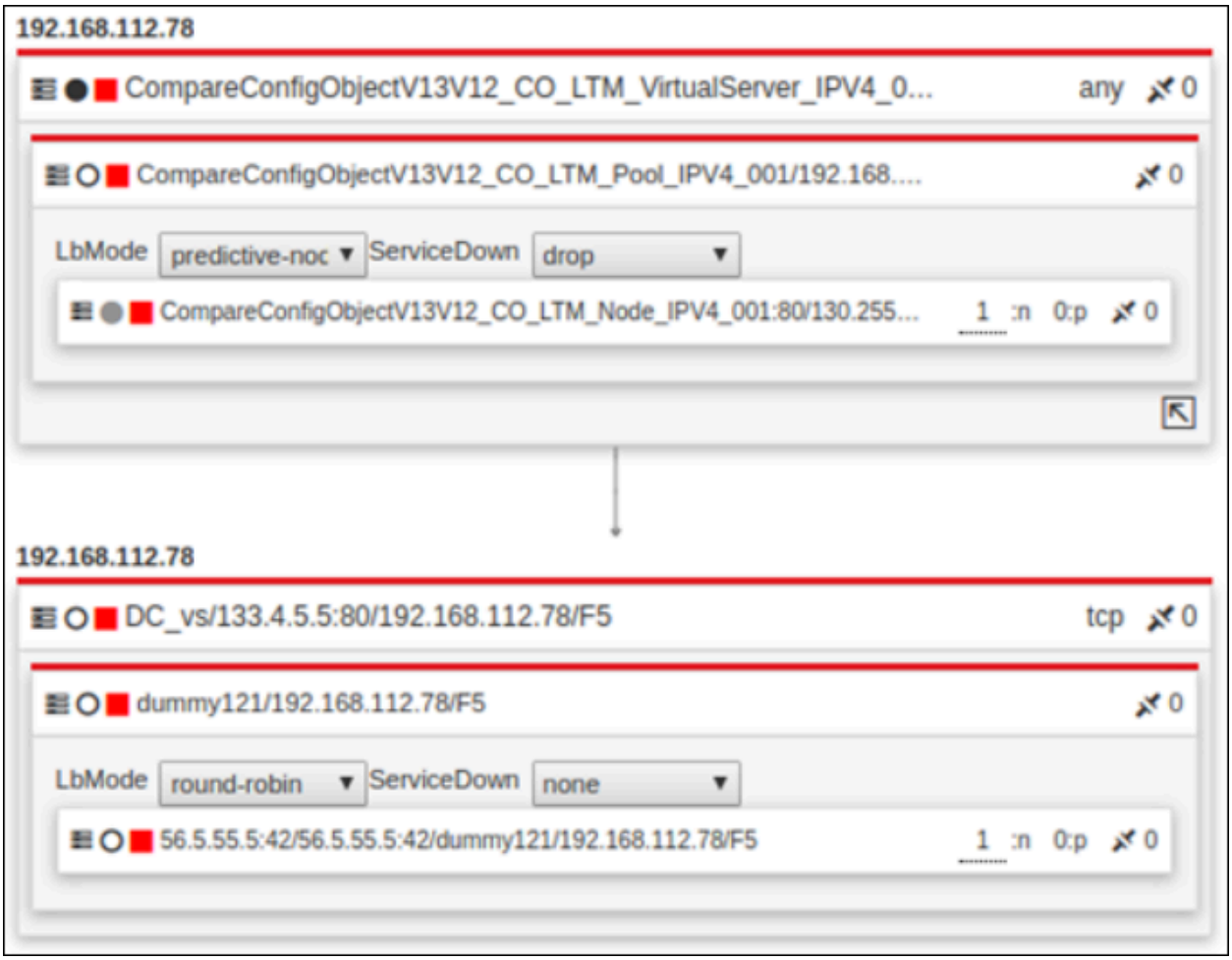
This is the topology structure displayed when an LTM/ SLB object is mapped under another LTM/ SLB object. Click on the down arrow to Expand button to get the VIP under VIP topology.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:

The VIPs may belong to the same /different devices of different vendors.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

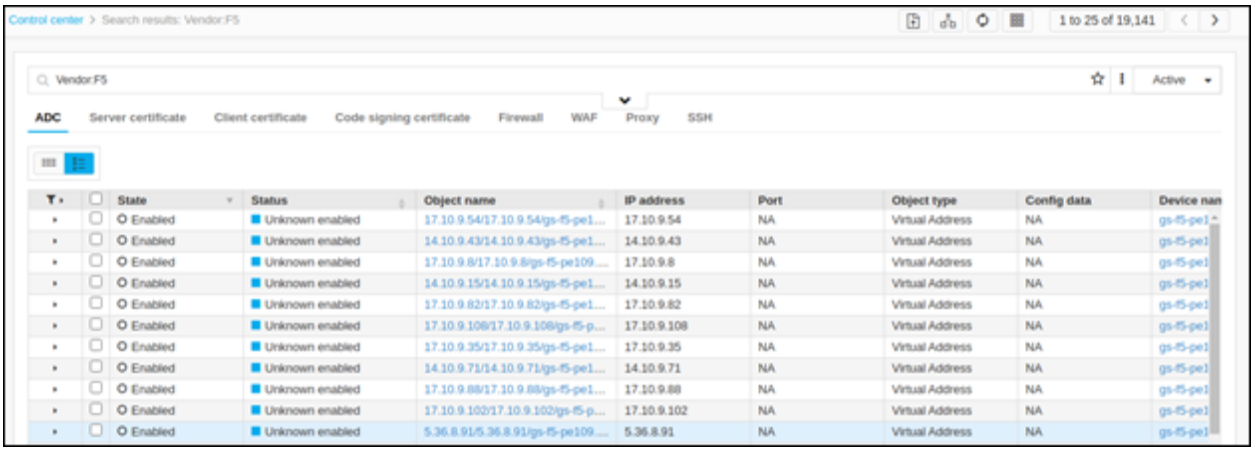
View the Object Details

The object details are represented in the form of a tabular with additional information as follows:

Column name	Description	Default View Columns
State	State of the objects.	Yes
Status	Status of the objects.	Yes
Object name	Based on the display name configured.	Yes (mandatory)
IP address	The IP address of the objects.	Yes
Port	Port of the IP address.	Yes
Object type	Type of object.	Yes
Config data	Configuration of the objects is displayed in a Pop upon clicking the link.	Yes
Device name	The device name of the object. On clicking, the user is redirected to the device addition page.	Yes
Vendor	Vendor of the object.	Yes
Device IP/FQDN	IP or FQDN of the device. The user is redirected to the device login on clicking.	No
Connection count	Display the live connection count of the objects if applicable.	No
Partition	Partition/tenant/context of the objects if applicable	No
DNS name	DNS resolution of the objects if applicable.	No

Hover your cursor over the result in the search results field to view the basic details of the selected item.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Related information
Search Using Free Text Entries
Search Using Frequent Search Links
Search Using Predefined Search Keys
Filter the Objects
About Infrastructure

Filter the Objects

To filter the objects in the Infrastructure view,

1. Run a search.
2. Click the filter icon located at the first cell of the table.

   The filter options are enabled in the header of the table.
3. Select and/or enter the criteria to filter the objects.

   The objects that match the filter criteria are displayed.

Note:

- To remove the filter option, click again the filter icon.
- To reset the filter criteria that have been applied, select and/or remove the text that has been applied for filtering the objects.

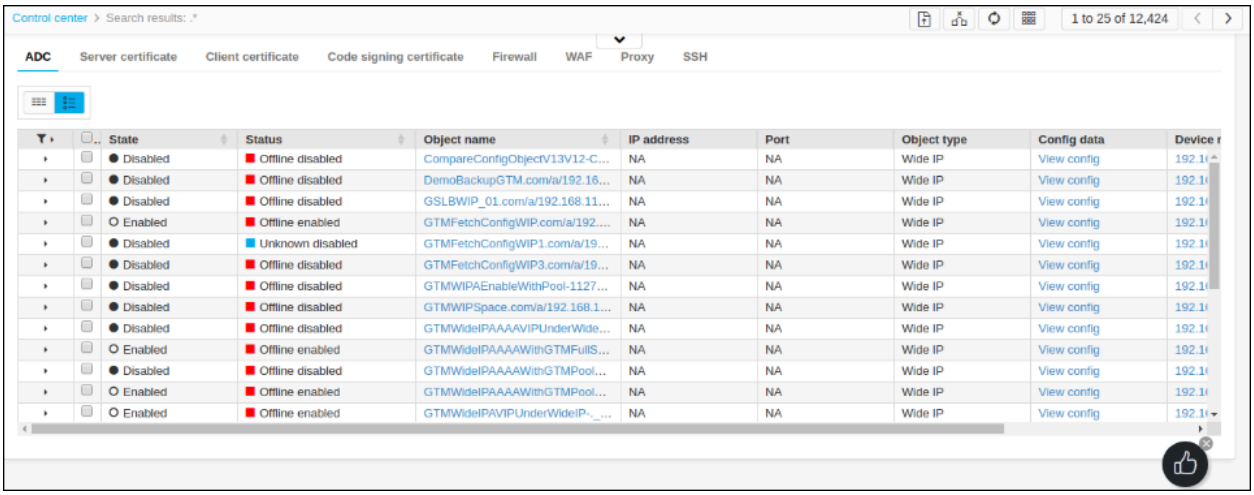
Related information
Search Using Free Text Entries
Search Using Frequent Search Links
Search Using Predefined Search Keys
About Infrastructure

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

About Infrastructure

Infrastructure View

Only the ADC objects corresponding to your search criteria are displayed and not its hierarchy (such as a parent, child 1, and child 2). By clicking the  (**Infrastructure view**) icon to switch from the Application view.



Related information
Search Using Free Text Entries
Search Using Frequent Search Links
Search Using Predefined Search Keys
Filter the Objects

Bookmarks

AppViewX allows the user to bookmark the frequently used search items.

Create a Bookmark

If any search strings or search queries are repeatedly used, it can be bookmarked for future access.

To create a bookmark for the ADC frequent search items,

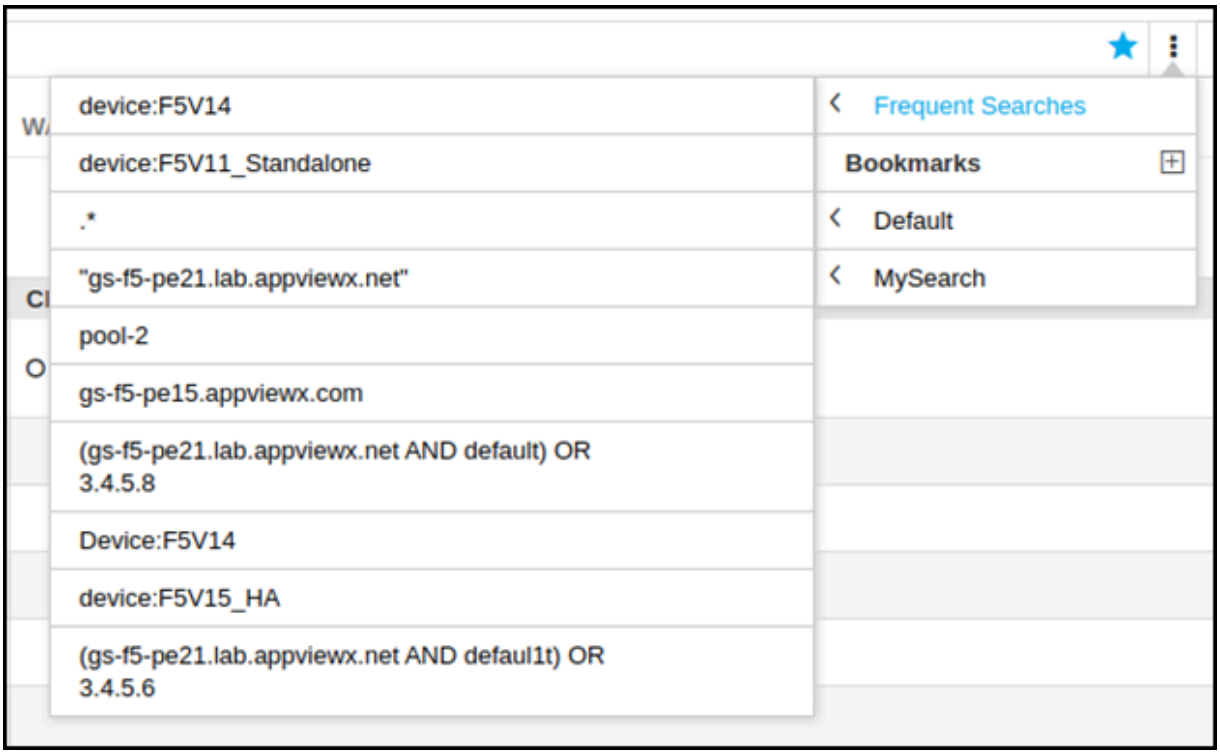
1. Click **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**. The Control Center search screen appears.
2. Run a search.
3. On the search results screen, click the  icon next to the search bar.
4. Click the  (Add) icon to create a bookmark folder.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. Click the (Star) icon on the search bar. A pop-up message will be displayed at the top of the screen, **Bookmark(s) created successfully**.
6. On the **Bookmark Added** pop-up screen, enter a name for the bookmark to help the users identify it.
7. Select the folder to which you want to add this bookmark from the dropdown list and click **Done**.

> **Note:**
> If you want to delete the bookmark, click the Remove button. A pop-up message will be displayed at the top of the screen, Bookmark(s) deleted successfully.

Queries that are frequently used are added to the frequent searches by default and the top 10 queries can be accessed in a single click as below:



Delete Bookmark

Follow the below steps to delete a bookmark.

1. Log in to the AppViewX application with valid credentials.
2. Select **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
3. Run a search.
4. In the grid / infrastructure view click on **More Options** button (3 vertical dots) next to the search bar.

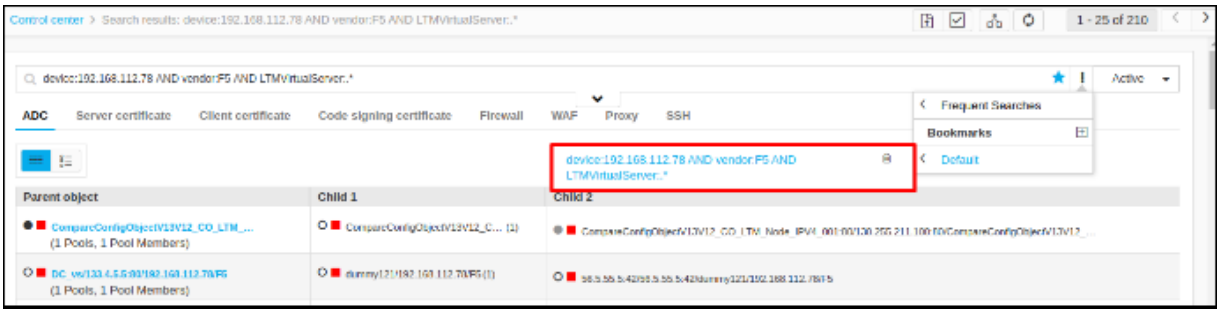
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



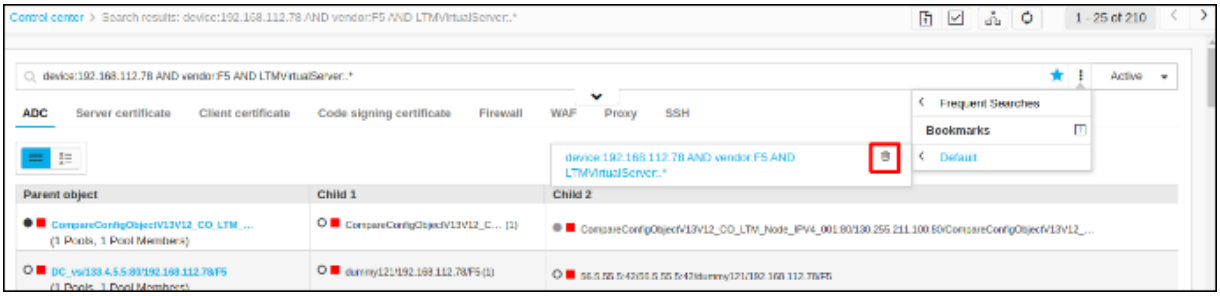
5. Mouse hover the folder name under the bookmarks menu.



All the bookmarks under the folder will be displayed.



6. Mouse hover on the bookmark name and the delete button will be displayed.



7. Click **Delete**.

Related information
Search Using Free Text Entries
Search Using Frequent Search Links
Search Using Predefined Search Keys

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Filter the Objects
About Infrastructure

Filter ADC Search Results

To filter ADC search results within the Control Center:

1. Run a search.
2. On the search results screen, click the **Active** drop-down option.



3. Select one of the three filter options in the dropdown list that appears:
   - **All** - View all search results, regardless of status.
   - **Active** - View search results for objects with a status of Active.
   - **Stand by** - View search results for objects with a status of Standby.
4. The screen then refreshes and filters the results on both the Application view and Infrastructure view based on the filter you selected.

Related information
Search Using Free Text Entries
Search Using Frequent Search Links
Search Using Predefined Search Keys
Filter the Objects
About Infrastructure

Export Search Results

When you search for the objects in the Control Center, the objects that match the search keyword are displayed in the Application view. The resulting objects can be exported. You can export the selected objects or all the resulting objects.

To export objects search results within the Control Center:

1. Run a search.

   By default, the search results are displayed in the  (**Application view**) page.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

2. Click the checkboxes beside the object name to select the object details that you want to export from the grid or select all of the ADC objects by clicking the  (**Select all**) icon in the Command bar.
3. All of the search results on the screen display with a blue background to indicate that they have been selected.
4. Click the  (**Export**) icon in the Command bar.
   The .csv file is downloaded to your computer.
5. If you want to export the search results displayed on the infrastructure view,  (**Infrastructure view**) icon to switch from the Application view search result page and do the following:
   - Select the checkboxes beside the first column of the grid to select the object details that you want to export from the grid. You can select all of the objects from the search results screen by selecting the first checkbox in the grid.
   - Click the  (**Export**) icon in the Command bar.
   - On the **Export** screen that pops up, select either the **All columns** or the **Displayed columns** radio button based on what you want to export from the grid.

     The .csv file is downloaded to your computer.

Related information
Search Using Free Text Entries
Search Using Frequent Search Links
Search Using Predefined Search Keys
Filter the Objects
About Infrastructure

Orphan Objects

Objects without parents will be considered as orphan objects. For example, an F5 pool that is not associated with wide ip will be considered an orphan pool.

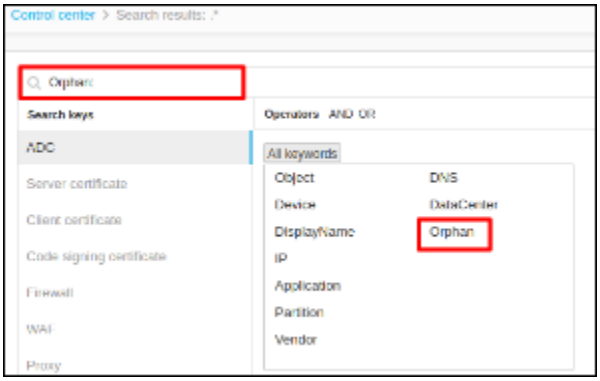
There are two ways to view the orphan objects in the Control Center.

Using Orphan Keyword

1. Log in to the AppViewX application with valid credentials.
2. Select **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
3. Expand the keywords tab.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click on **Orphan** keyword from the all keywords section and click enter.



Using Orphan Objects Button

1. Log in to the AppViewX application with valid credentials.
2. Select **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
3. Run a search.
4. From the grid / infrastructure view click the **Orphan Objects** icon.



5. In the grid view, the parent column will always be empty for the orphan object.

Right click actions

AppViewX provides some right click actions for objects in all the views. Users can do right click action on individual / multiple objects. The actions that appear on the list may vary depending on the object type and the vendor.

1. Log in to the AppViewX application with valid credentials.
2. Select **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **App Search**.
3. Run a search.

Executing From Application View

- Users can do right click actions on individual object level in grid view. To do an action on an individual object right click on the object.
- Users can do right click actions on multiple objects level by selecting the objects using the Select button near the object. After selecting the objects do a right click. When the user clicks on a particular action the actions will be applied only for the applicable actions from the selected objects.

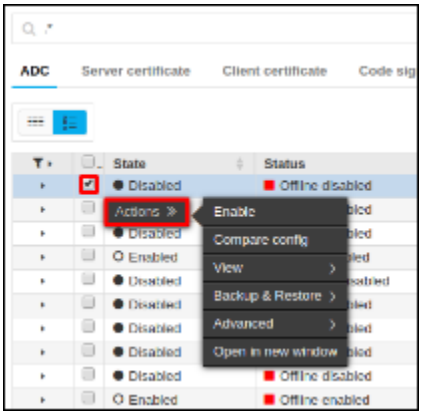
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Users can do right click actions on all the objects in the page. When the user clicks on the Select All button in the top right corner and then right click on the objects. When the user clicks on a particular action the actions will be applied only for the applicable objects from the selected objects.



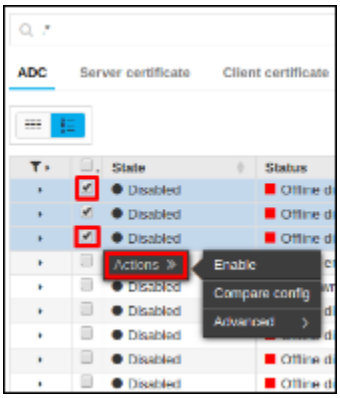
Executing From Infrastructure View

- Users can do right click actions on individual object level in grid view. To do an action on an individual object, select a single object and hover on **Actions**.

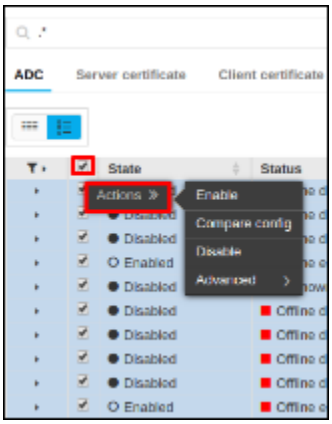


- Users can do right click actions on multiple objects level by selecting the checkboxes of the object. After selecting the objects hover on **Actions**. When the user clicks on a particular action the actions will be applied only for the applicable actions from the selected objects.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



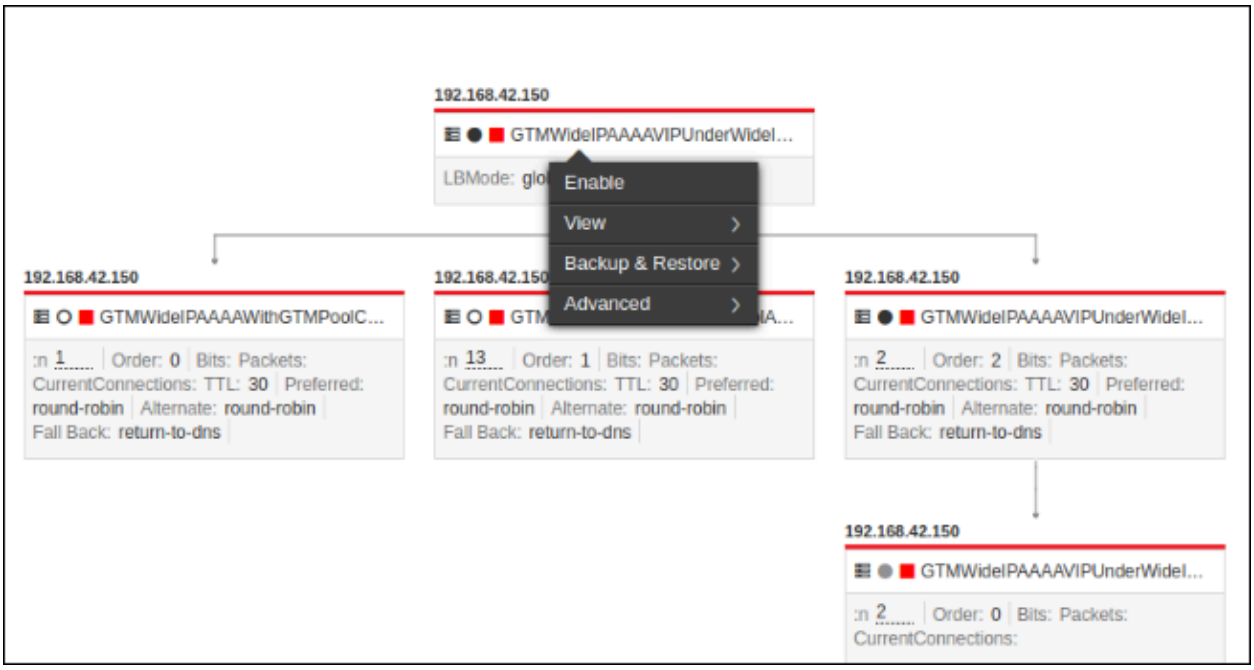
- Users can do right click actions on all the objects in the page. Users can select all the objects by selecting the **Select all** checkbox (first checkbox in the grid). When the user clicks on a particular action the actions will be applied only for the applicable objects from the selected objects.



Executing From Topology View

From topology view users can do right click actions on individual objects. To do an action on an individual object right click on the object and click the required action.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Below are the right click actions supported in AppViewX. Basically they fall under two categories.

- **Write actions** : These actions will change the configuration of the object in the device and the respective changes will get updated in AppViewX also. Whenever the user triggers the below actions user will get a **Comments box** for most of the actions, where the entered comments will be maintained in auditlog. To perform the action users should enter a valid comment in that and then click **Yes.**
  - Enable
  - Disable
  - Force down
  - FD clear active connections
  - Enable all
  - Disable All
  - Forcedown all
  - Enable persistence
  - Disable persistence
  - Restore object
  - Clear persistence
- **Read actions** : These are the view actions. This will not modify any configuration.
  - Compare config
  - View source connections
  - View persistence
  - Backup device

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- View graph
- View config
- View log
- View Alerts

Related information
Actions

Actions

Actions - Table

Action	Description
Enable	Enables the Object.
Disable	Disables the Object.
Force Down (Specific for F5)	Force down the object.
FD Clear Active Connections (Specific for F5)	Force down the object and terminate all the current connections.
Graceful disable (Specific for AVI)	Disable the object only when all the current active connections are closed either by server or client.
Compare Config	Compare current and /or archived configurations of objects with the same name.
View Source Connections (Specific for F5)	View the active connections for VIP and LTM Pool.
View Persistence (Specific for F5)	View the persistence for VIP and LTM Pool.
Clear source Connections (Specific for F5)	Clear the persistence for VIP and LTM Pool.
Open in New Window (Grid / Infrastructre View)	Displays the topology view of the selected object in the new window.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

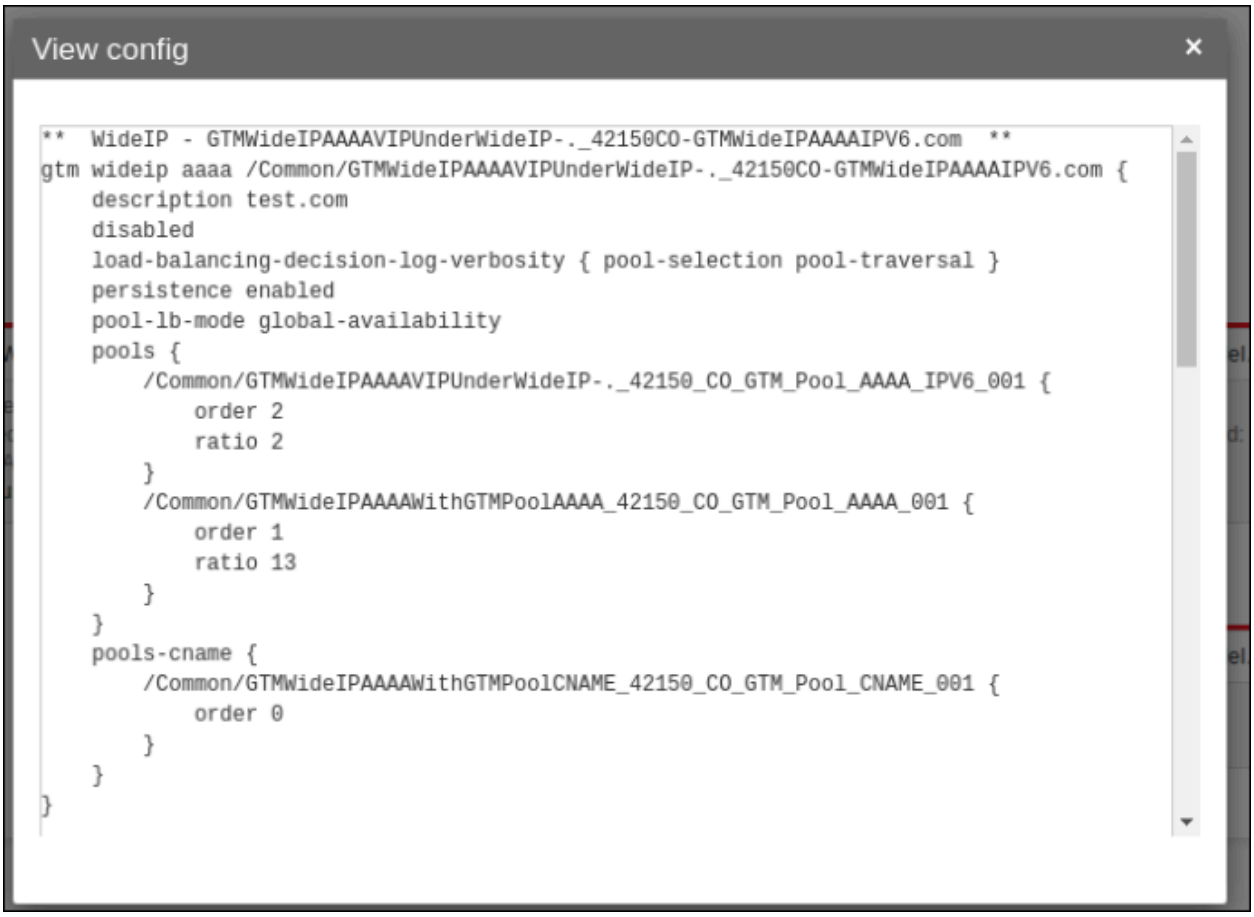
Actions - Table (continued)

Action		Description
Backup & Restore	Backup Device	Create the backup of the device associated with the object. Backup will be available in the Default backup group.
	Restore Object	Restore the object configuration to a previous state.
Advanced (Specific for F5)	Enable All	Enable all the LTM pool members with the same ip port on the same device.
	Disable All	Disable all the LTM pool members with the same ip port on the same device.
	Force Down All	Force down all the LTM pool members with the same ip port on the same device.
	Enable Persistence	Enable or Disable the tracking and storing of session data, which is used to ensure that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions.
	Disable Persistence	
View	View Config	View the configuration of the object and its child objects.
	View Graph	View the timeline statistics for the object.
	View Log	View the log history specific to this object.
	View Alerts	View the alerts related to this object.

View Config

Shows the configuration of the selected object and its child.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



View Graph

Shows the timeline statistics for the selected object. Graph will be available only if the statistics is configured in the Settings. For more details on settings refer Configuring statistics collection.

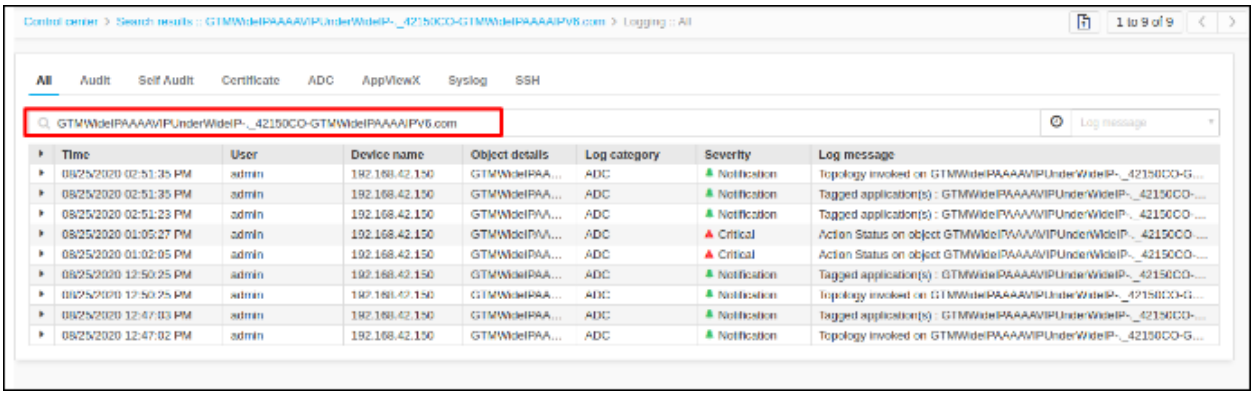
When the user clicks on the view graph pop-up will be displayed.

- **Statistics** - The entries in this list vary depending on the object. when the user selects the different stat parameters the corresponding graph will be updated.
- **Interval** - The following time intervals can be selected: **Day**, **Week**, **Month**, or **3 Months**. When the user selects different intervals the graph displayed will be updated to the selected time frame.

View Log

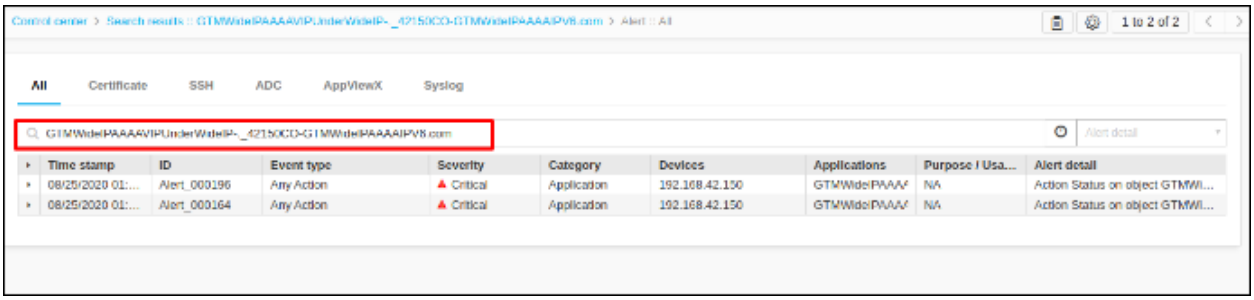
When the user applies the view log action on particular objects, the audit logs related to that particular object alone will be filtered and displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



View Alert

When the user applies the view alert action on particular objects, the alerts related to that particular object alone will be filtered and displayed.



Related information
Right click actions

Configuring Dashboards

Dashboards help to organize various resources at a single place and monitor them in a single window. AppViewX supports four different custom widgets for managing ADC objects. They are:

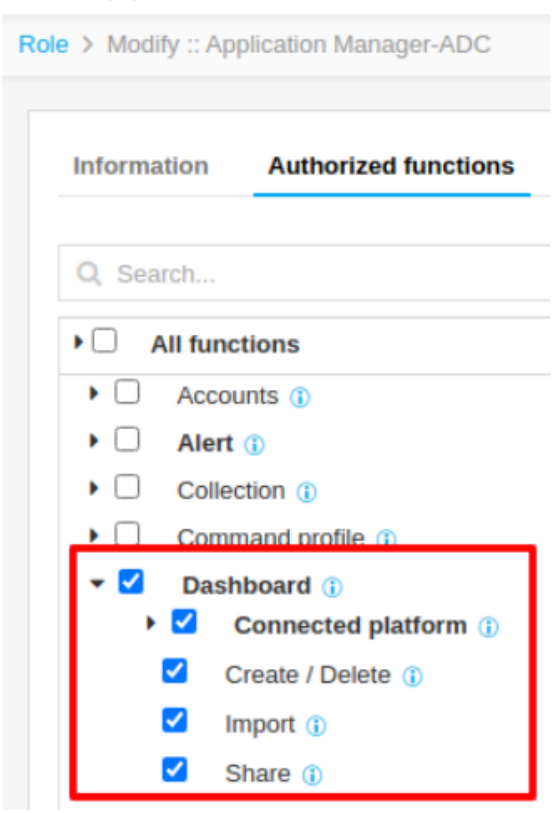
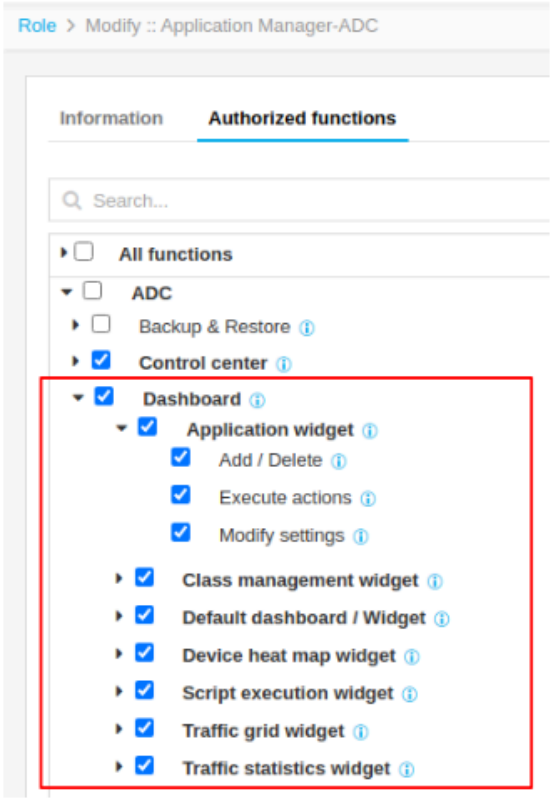
- Application view Widget
- Class Management Widget
- Traffic Statistics Widget
- Traffic Grid Widget
- Script Execution Widget

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

About Dashboard

Before you begin

- To configure dashboards and widgets, the user should have ACF permissions.

Access for Dashboard:	Access for ADC Widgets:
	

- For adding objects in the widgets, the device should be Managed in the inventory and the user should have object level access (ACL) in the resources. For performing actions on the object, RW permission should be provided.

Default Dashboard

The Default ADC Dashboard is pre-configured and has a default set of widgets. The widgets are Device heat map, Top 10 VIPS by Connection, Application Heatmap, Number of Objects, Top 25 Applications by Connection and Unused Objects Report.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To view the reports related to ADC,

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.

   A pre-built ADC Dashboard appears by default.

2. From **Dashboard Inventory** click on ADC dashboard.



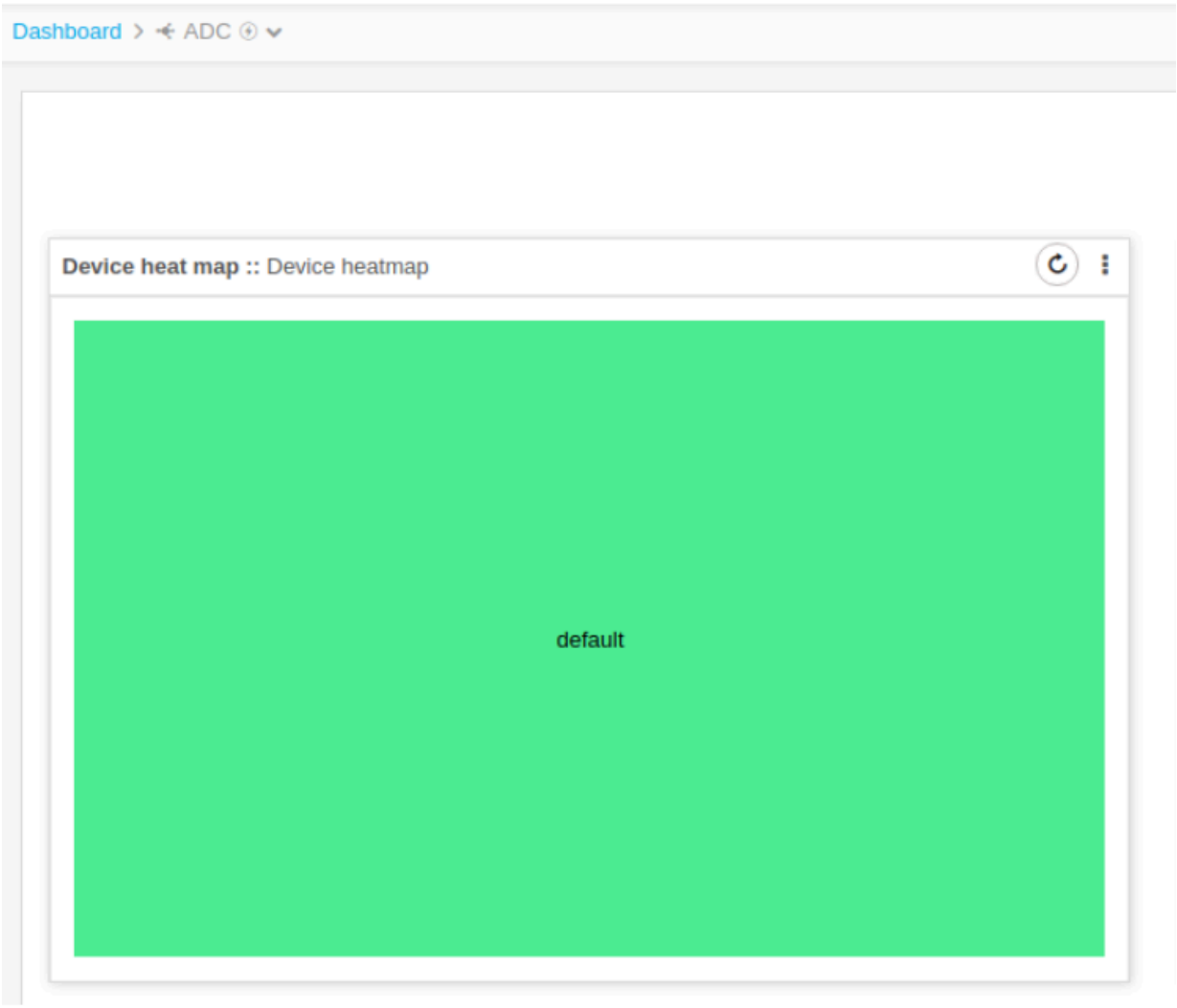
Note:
The default widgets can be viewed, aligned, and/or refreshed and not editable.

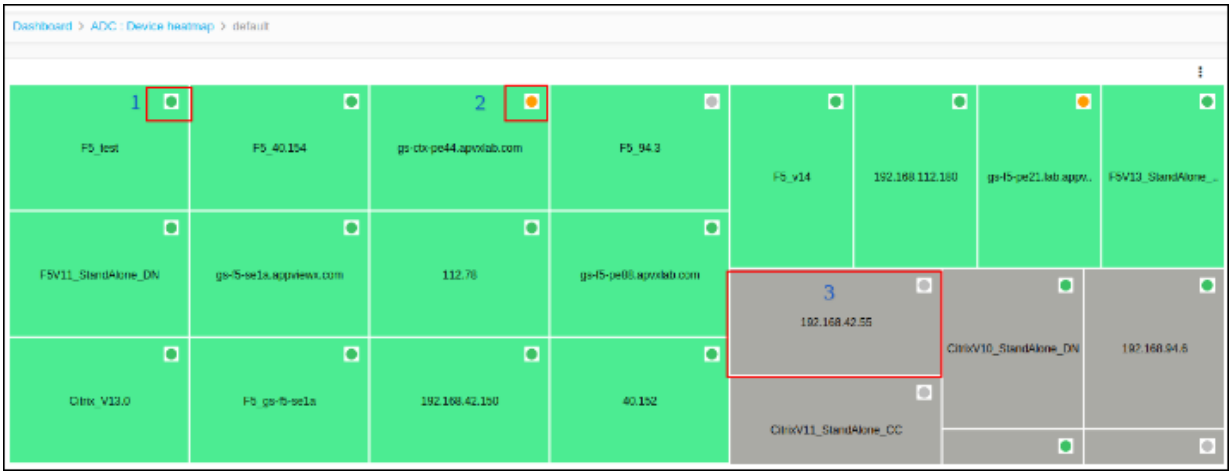
Widgets in Default Dashboard

- **Device heat map**

  Device heat map widget shows the CPU utilization, Bandwidth Utilization and Memory Utilization of each device managed in Device Inventory.

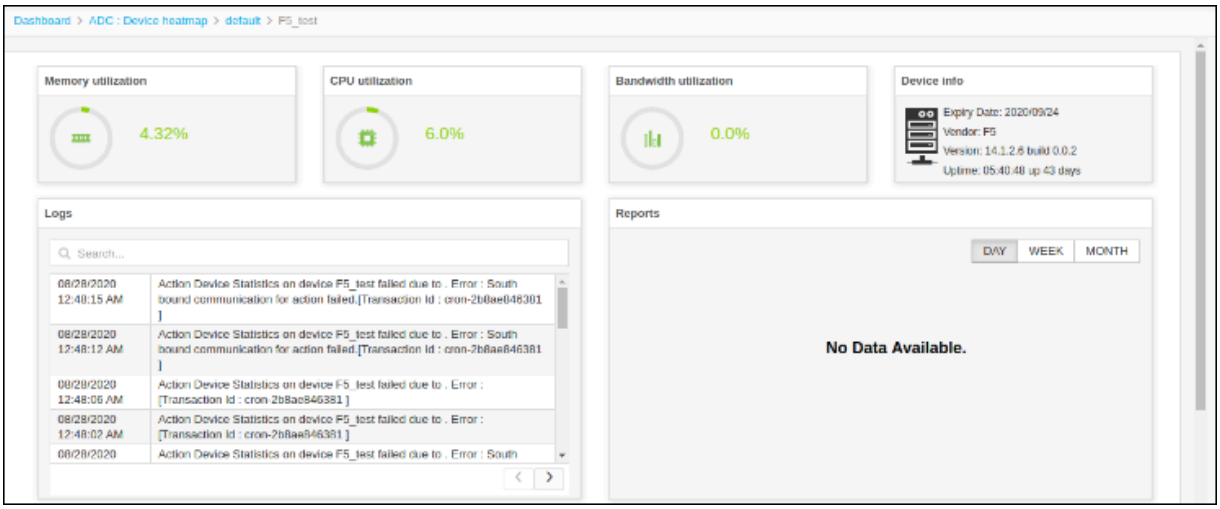
Copyright © 2025 AppViewX, Inc. All Rights Reserved.





Copyright © 2025 AppViewX, Inc. All Rights Reserved.

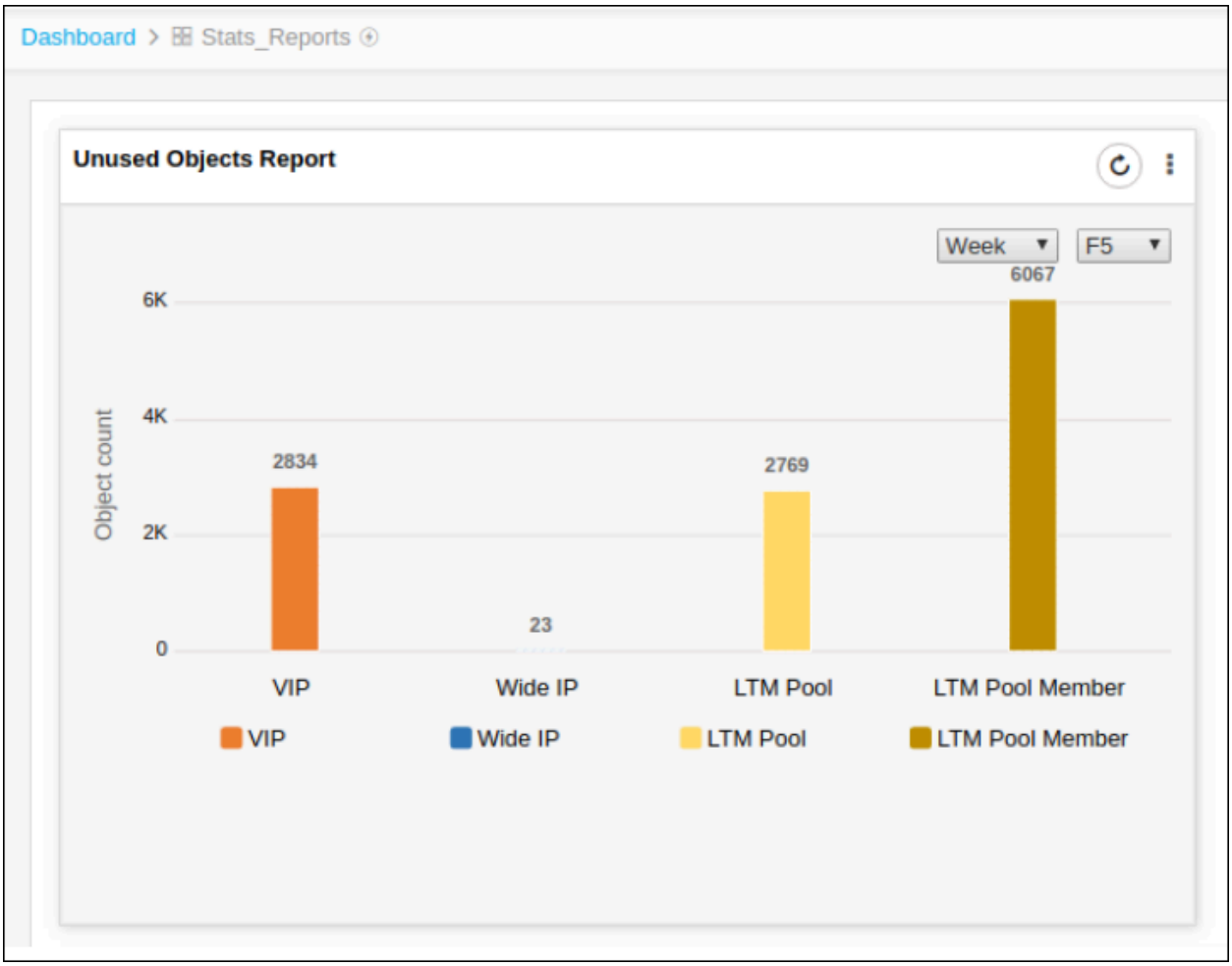
1. Active Device
2. Standby Device
3. Unreachable devices are shown as Grey.
4. Click on the Device and to get detailed information about CPU utilization.



Unused Object Report

Unused Object Report shows the number of unused objects over a period of time. Supported vendors are F5, A10, and Citrix.

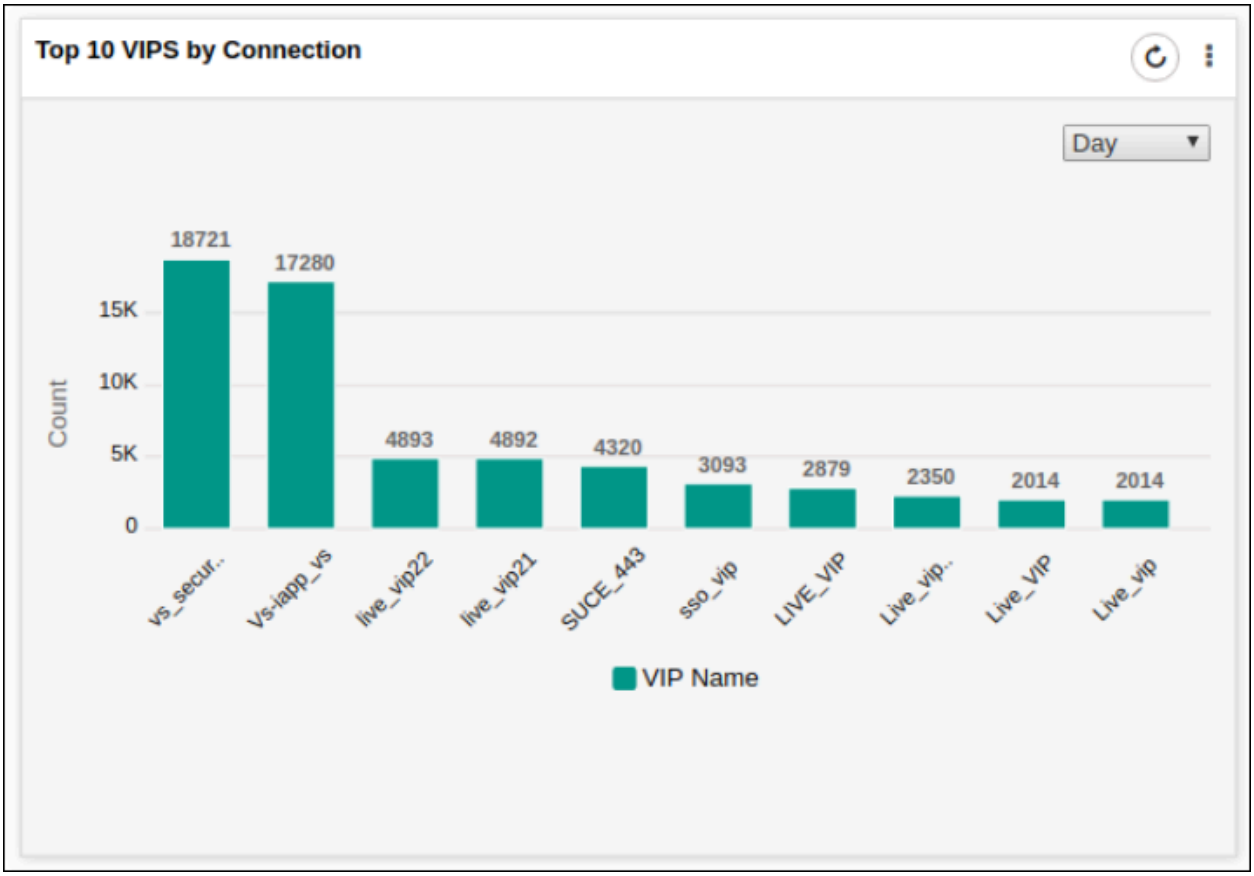
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Top 10 VIPS by Connection

This widget shows the details about the top 10 VIPs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Related information
Create a Dashboard/Widget
Add Widgets

Create a Dashboard/Widget

The custom dashboard allows you to manage, monitor, and interpret all the configured applications and their objects. It provides customizable widgets to get an overview of all the ADC Applications within the AppViewX platform and manage/monitor traffic from a single screen. Customize your Application dashboard with predefined and custom widgets as per the business need. You can create dashboard using any of the following method:

To create a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If this is the first dashboard you are creating in the system, click the Create Dashboard button that appears in the center of the screen. If you have already created at least one dashboard, click + (**Create**) icon on the top-right or click + (**Create**) icon.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. On the Create Dashboard/Widget pop-up window, enter a name for the new dashboard.
   The name must not contain spaces in it.
4. Select a solution from the **Select Solution** dropdown to which you want the widget to be created: ADC, Firewall, Certificate, SSH or WAF. Select the solution from the dropdown to which you want the corresponding widgets to be managed: Certificate.
5. Select the **Widget Type** as Custom or Default.
6. If the Custom radio button is selected in Step 5, then choose any one of the below options from the Select Widget dropdown:
   - **Application view** – Allows you to group the service objects of a single application. The widget displays the health of these objects and the number of current connections that the services are receiving.
   - **Traffic statistics** – Displays a chart showing live and historic performance statistics for individual device objects.
   - **Script execution** – Saves script files on a local machine and provides easy access to maintain and execute script commands from within the widget.
   - **Traffic grid** – Allows you to monitor and control the Traffic Percentage of the Applications across data centers. The status, state, and statistics for applications can be viewed through this widget.
   - **Class management** – Allows you to view and modify the classes associated with iRules.
   - **HeatMap** – Allows you to view statistics for managed, failed, and unresolved devices or device groups.
7. If the default radio button is selected in Step 5, choose the default widgets you want to manage/monitor in the custom dashboard. Select the default widgets you want to manage/monitor in the custom dashboard.
8. Enter a name for the new widget that you will be creating on the dashboard.
9. Click **Create**.
   You are redirected to the widget configuration screen, which varies according to the widget selected.

Note:
- The Settings screen for the new dashboard/widget appears. The contents of the Settings screen vary depending on the type of widget you are adding to your new dashboard.
- All the default dashboards and custom dashboards created are listed in the dashboard inventory.
- If the Auto Sort Dashboards Alphabetically is enabled the dashboards are listed in alphabetical order, if disabled dashboards are listed based on the created order.

Related information
Add Widgets

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Add Widgets

Widget can be created in a new Dashboard or existing Dashboard**.**

Add Widget in New Dashboard

1. Log in to the AppViewX application with valid credentials.
2. Select **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards** > Click the **+** (**Add**) icon.

Add Widget in Existing Dashboard

1. Log in to the AppViewX application with valid credentials.
2. Select **Menu** > **Dashboard**.
3. Search for the Dashboard from dashboard inventory and Click on the dashboard name add a new widget.



4. Click the Add Widget icon.



5. Select Widget Type and provide Widget name.

   Based on the selected widget type, the corresponding widget settings page is loaded.

Related information
Create a Dashboard/Widget
Dashboard Actions

Application View Widget

Using Application View Widget, the ADC Primary objects like WideIP, GTM Pool, Virtual servers, LTM Pool and nodes can be monitored for state and status change and Actions can be performed. The actions for each object is predefined and can be selected accordingly.

To Create a Group

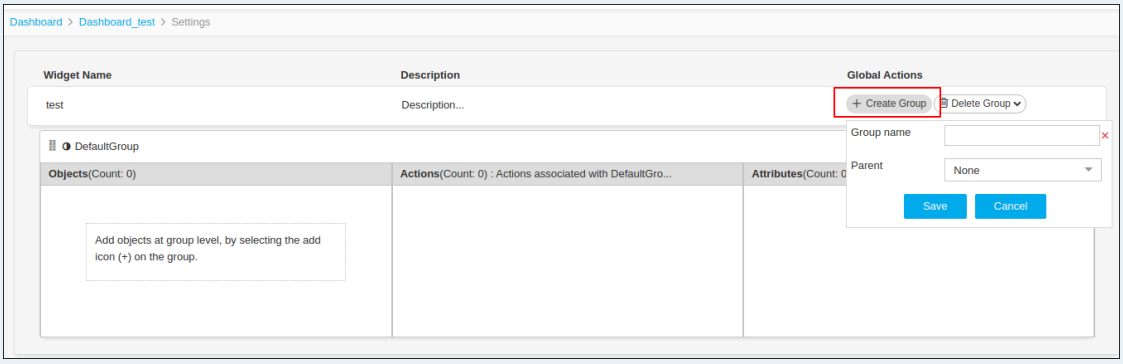
1. Click **Create Group**.
2. Provide a user defined group name.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Parent group can also be selected from the list of available groups added in the widget.

Note:

Maximum group hierarchy allowed is four.

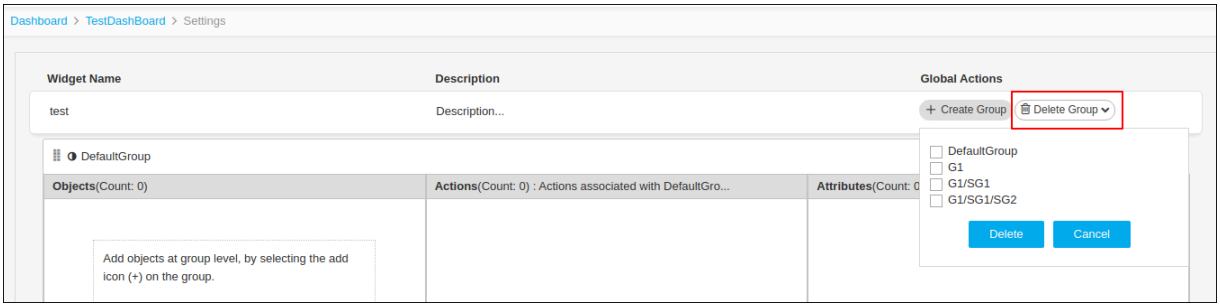


To Delete a Group

1. Click **Delete Group** and select the groups to be deleted.

CAUTION:

The groups will be deleted with all subgroups and it's configured objects.

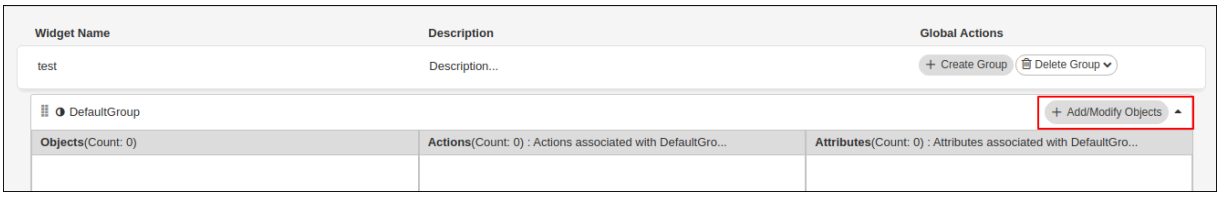


2. Click yes on the confirmation to proceed the deletion.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To Add/Modify Objects in a Group

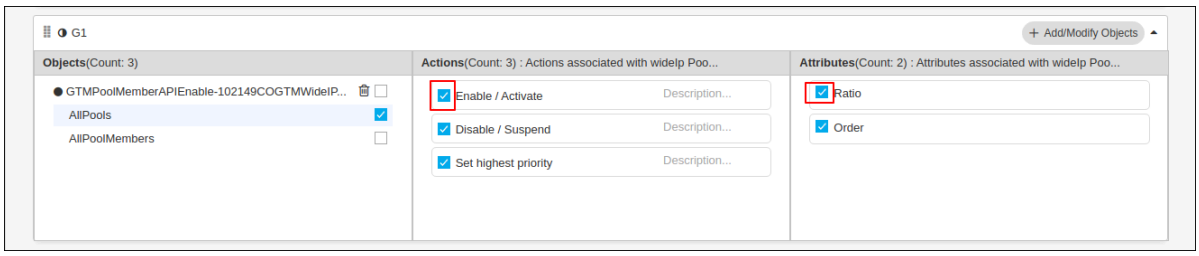
1. Click **Add/Modify** objects on the Default/Created Group



Add/Modify Objects - Field and Description Table

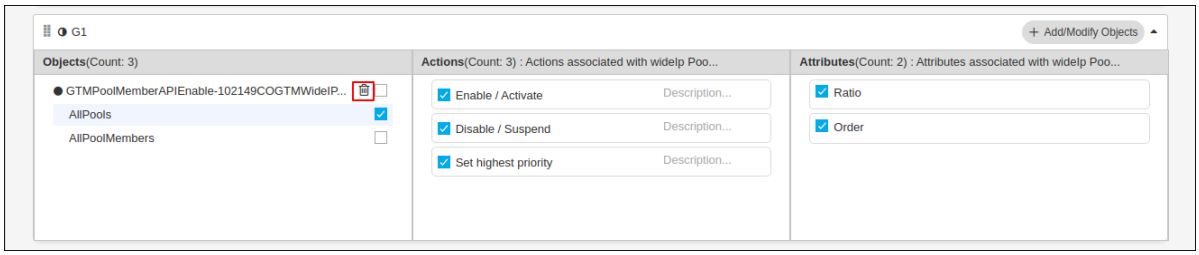
Field	Description
***Vendor**	Vendors of Managed devices from the inventory are listed in Alphabetical order.
***Device State**	Available options: **Active** : Active devices are listed. **Standby** : Standby devices are listed. **All** : All Managed devices are listed.
***Device Name**	Based on vendor and device state, available devices are listed.
***Object Type**	Based on the vendor, all supported object types are listed.
***Hierarchy**	Based on the Object Type selected the hierarchy is listed.
**: Mandatory fields*	

- Based on the objects added, default actions are selected. Users can deselect the actions



- To delete added objects in a group, click on the delete symbol next to the object.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



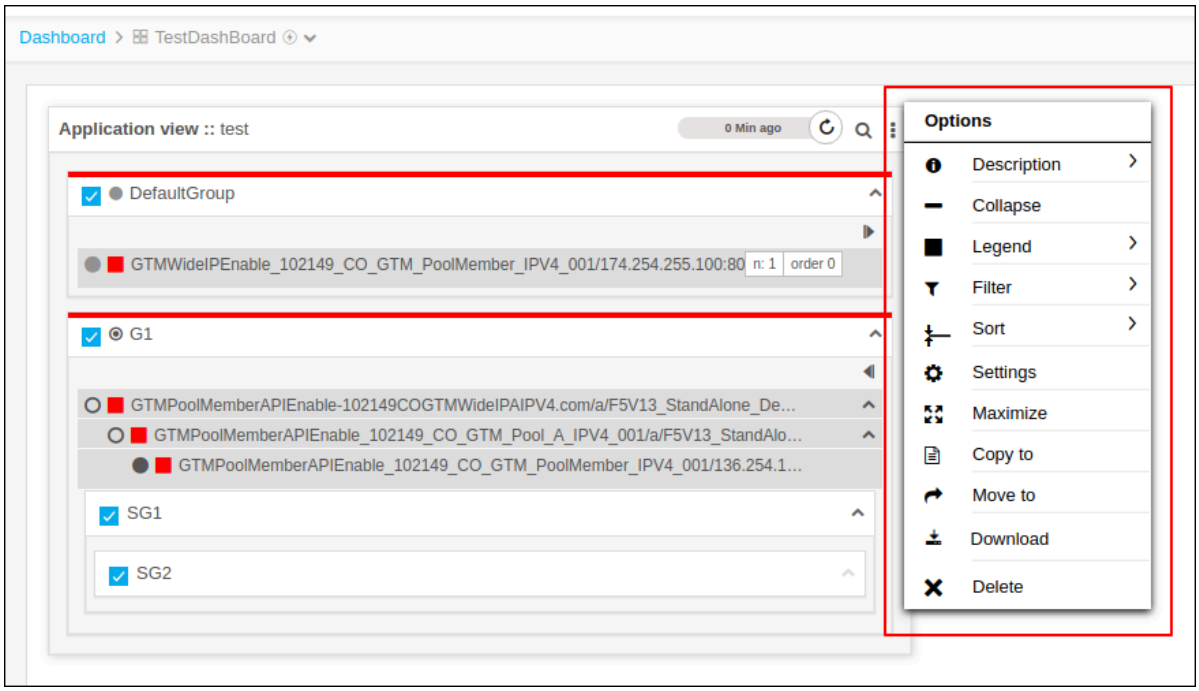
- After completing the configuration, a Test Widget will be displayed. The test widget includes the following:



- Widget Name
- Last refresh time, when the widget is refreshed,the current state and status are fetched from the device and updated in the widget
- To search for an object added in the widget
- Settings options
- Collapse/Expand the Group
- To display the attributes of the objects

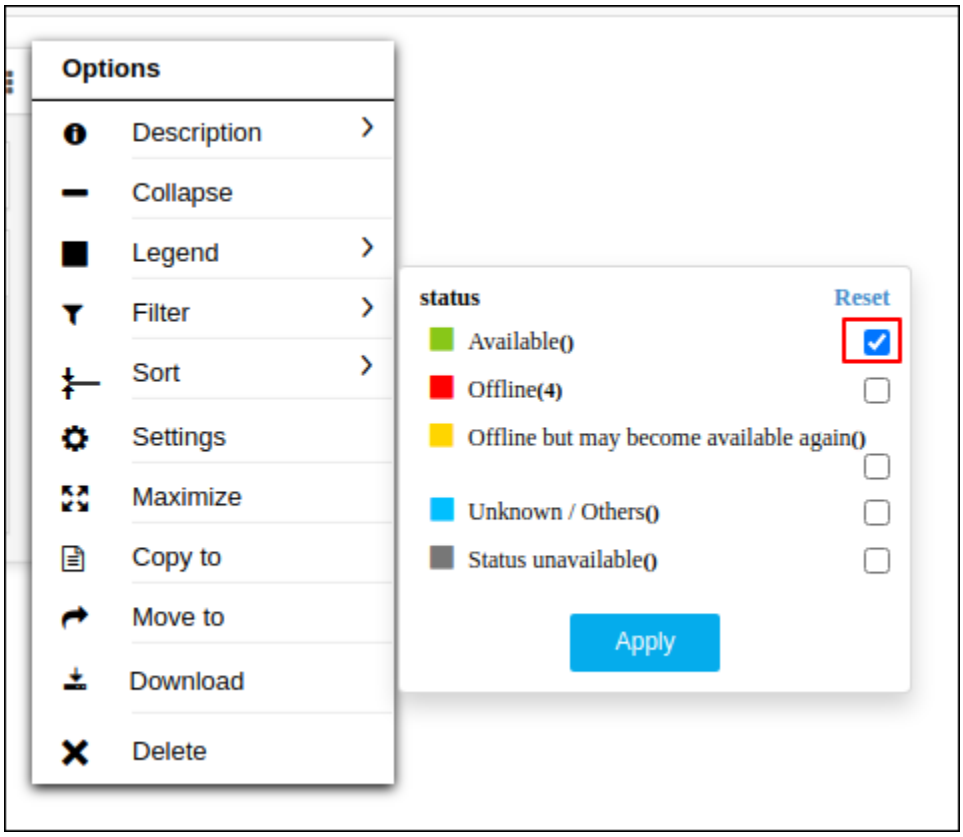
Widget Settings:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- Description - Description provided for the widget while creating it.
- Collapse/Expand the Widget
- Legend - Gives information about representations used in Widget.
- Filter - Filter the objects to be shown based on status and click **Apply**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- Sort - To Sort the objects in Ascending/Descending order
- Settings - Settings page of the widget
- Maximize - To Maximize the widget
- Copy to - The widget is copied to another Dashboard.
- Move to - The widget is moved to another Dashboard.
- Download - Download the widget as csv file
- Delete - to delete the widget

2. Based on the above filters, available objects are listed. Specific objects can also be searched and added.
3. Select the objects to be added and click **ADD**.

Actions on Objects/Groups

Right click on Object/Groups and actions are displayed based on the object type and ACF permissions assigned to the user.

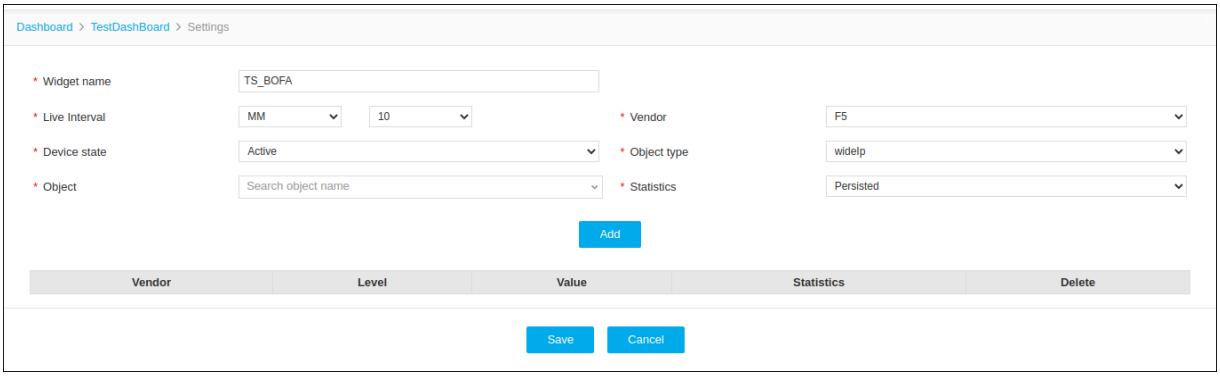
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Related information
Traffic Statistics Widget
Script Execution Widget
Traffic Grid Widget
Class Management Widget

Traffic Statistics Widget

Traffic statistics widget is used to monitor live/historic statistics of objects.

1. Add objects



Add objects - Field and Decription Table

Name	Description
***Widget Name**	User defined Name for the Widget No special characters are allowed, except '_' , '.' , '-' , '|' , ':'. The name should not start with a special character.
***Live Interval**	Set time interval for live statistics collection.
***Vendor**	Statistics supported Vendors are listed
***Device State**	Available options: **Active** : Active devices are listed. **Standby** : Standby devices are listed. **All** : All Managed devices are listed.

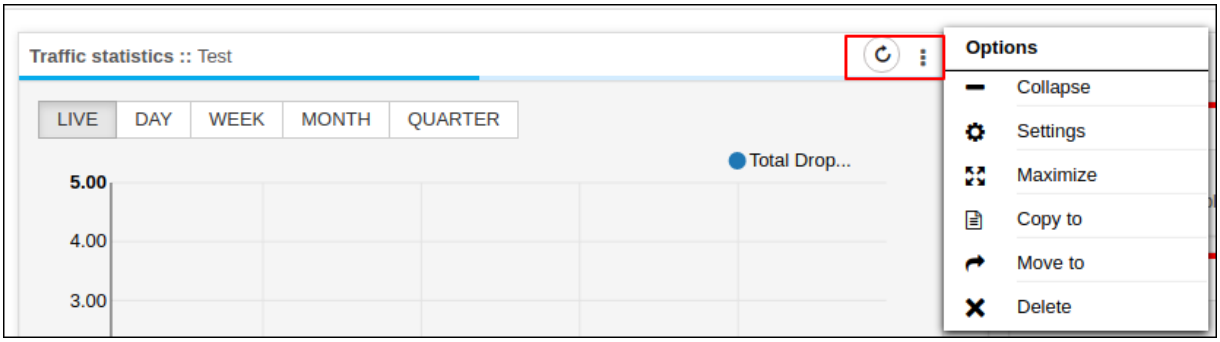
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
***Object Type**	Based on the vendor, all supported object types are listed.
***Object**	Based on vendor and object type filter available objects are listed.
***Statistics**	Based on vendor and object type filter supported statistics parameters are listed.
Alias Name	Custom name for the object name.
**: Mandatory fields*	

2. Provide the required information and click **Add**.
3. Click **Save**.

   By default Live stats are collected. If Insight is enabled then historic statistics are shown.

   Widget Setting:



- Refresh - Refreshing the widget will get live stats for the configured objects.
- Collapse/Expand - Collapse and expand the widget screen
- Settings - Configure the settings
- Maximize - To maximize the widget size
- Copy to - The widget is copied to another dashboard
- Move to - The widget is moved to another dashboard.
- Delete - Delete the widget

Related information
Application View Widget
Script Execution Widget
Traffic Grid Widget
Class Management Widget

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

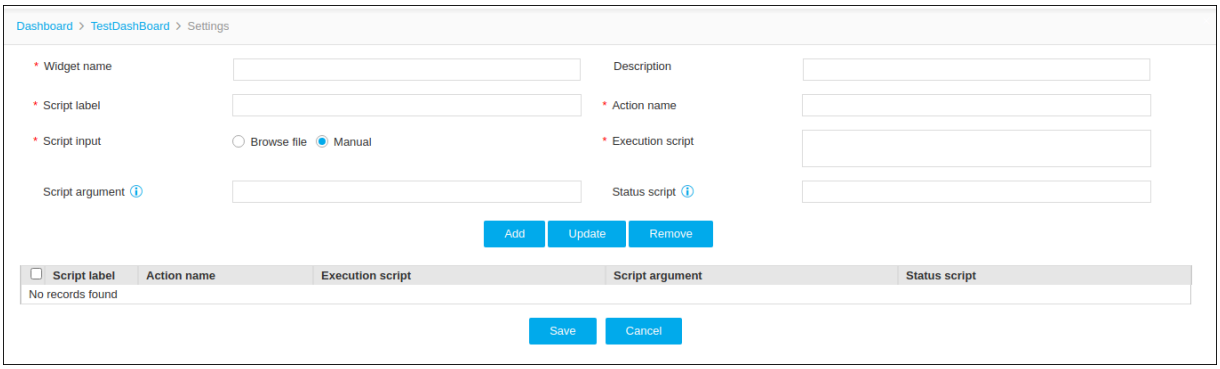
Script Execution Widget

Using Script Execution Widget, scripts which can be executed within 5 minutes can be configured and executed.

Before you begin: Set Timeout in system Settings. Refer to Configuring Script Execution Settings.

Configuring Script Execution Widget

1. Add scripts.



2. Enter or select the script details.

Add Scripts - Field and Description Table

Name	Description
***Widget Name**	User defined Name for the Widget **Note:** No special characters are allowed, except '_' , '.' , '-' , '|' , ':'. The name should not start with a special character.
Description	Description for the Widget
***Script label**	Enter script label.
***Action Name**	Enter the action name.
***Script Input**	**Browse file:**

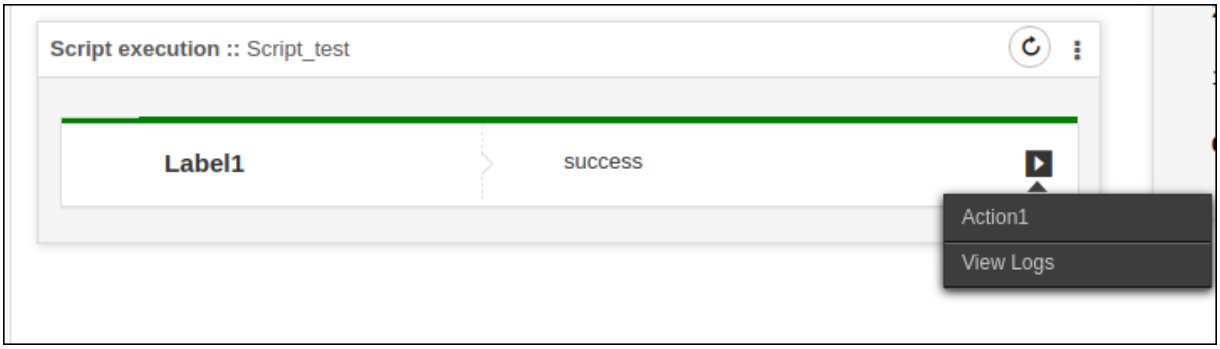
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	Script file can be uploaded. **Manual:** Script can be typed in the text box.
***Execution Script**	1.File can be uploaded. or 2. Manual script input can be added.
Script argument	Arguments for the script can be mentioned with space as delimiters.
Status script	The full path of the scripts in AppViewX servers to update the color and status message of the bar. This functionality is used to get the status/result of a script.

3. After adding all information, click **Add** and **Save**.

Execution of Scripts

1. Click the **Play** icon, and then click the **Action Name** configured.



2. To verify the output logs, click the **View Logs** option.

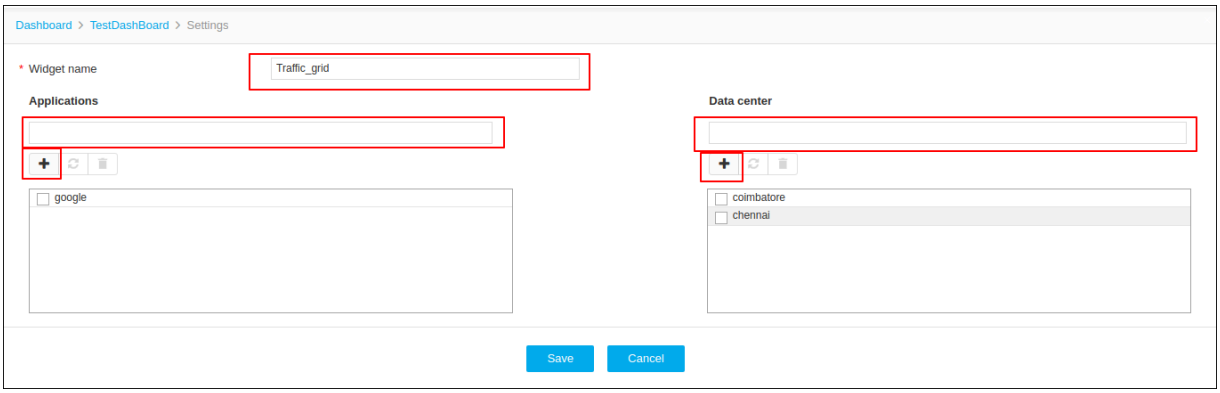
Related information
Application View Widget
Traffic Statistics Widget
Traffic Grid Widget
Class Management Widget

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Traffic Grid Widget

Configuration Steps:

1. Enter the widget name.

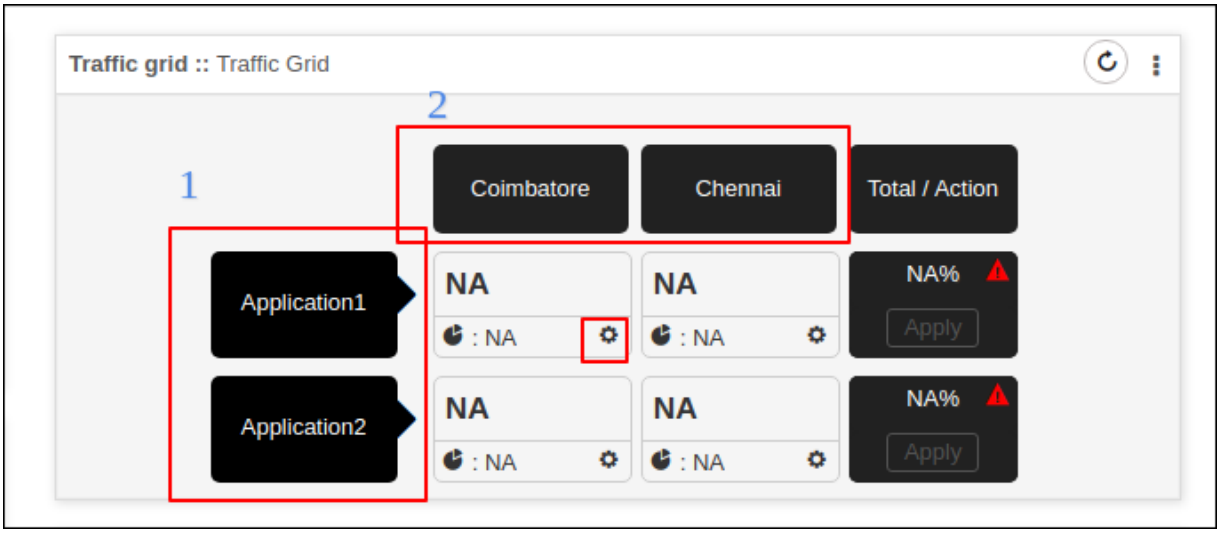


2. Add **Application Name** and **Data Centers** and click the **+** (**Add**) icon.

> **Note:**
> As required, rearrange the application and data center lists using a drag-and-drop method.

3. Click **Save**.

   Sample Widget



- 1. Applications Configured
- 2. Data Centers added
- 3. Click on **Settings** icon to configure objects and rules

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Configure Availability Status

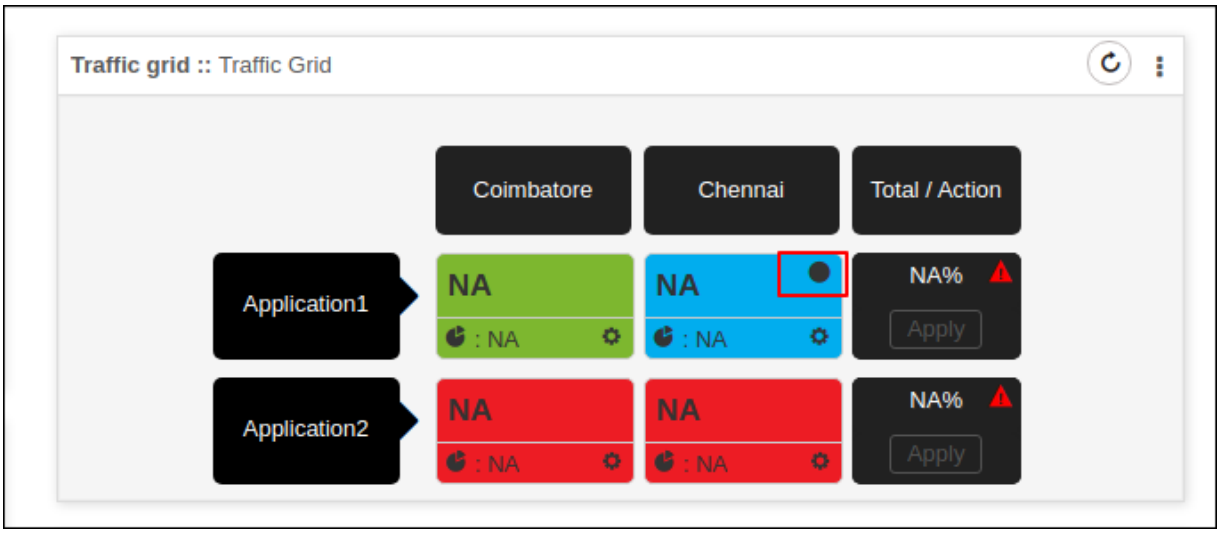
1. Provide required information

Availability Status - Field and Description Table

Field	Description
***Vendor**	Select vendor
***Device State**	Available options: **Active** : Active devices are listed. **Standby** : Standby devices are listed. **All** : All Managed devices are listed.
***Object type**	All Primary ADC objects are listed.
***Object Name**	Object names can be searched and selected.
**: Mandatory fields*	

2. Click **Add**, and then **Save**.
3. Added objects can also be updated by selecting the object in the list and **Update**
4. Added Objects can be deleted by selecting the object in the list and **Delete**

   After configuring the objects the color representation changes according to status of the objects and State of the objects are also represented at top right corner of every application.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

2. Configure Traffic Percentage

By adding objects in traffic percentage, users can route the traffic by changing the traffic percentage.

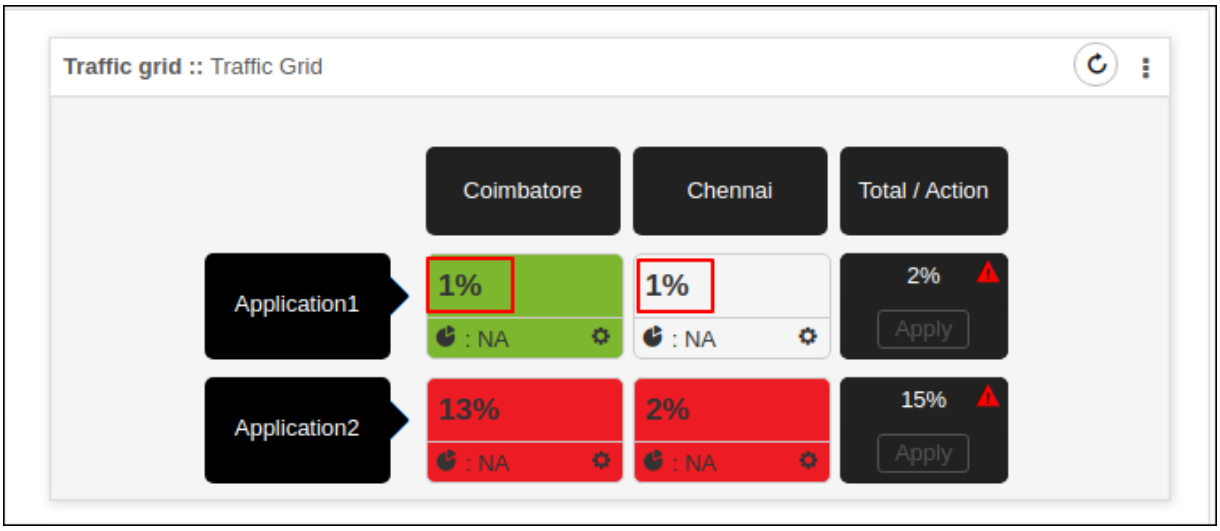
1. Provide required information

Traffic Percentage - Field and Description Table

Field	Description
***Vendor**	Select vendor
***Device State**	Available options: **Active** : Active devices are listed. **Standby** : Standby devices are listed. **All** : All Managed devices are listed.
***Object type**	Only traffic controllable objects like pool members are listed
***Object Name**	Object names can be searched and selected.
**: Mandatory fields*	

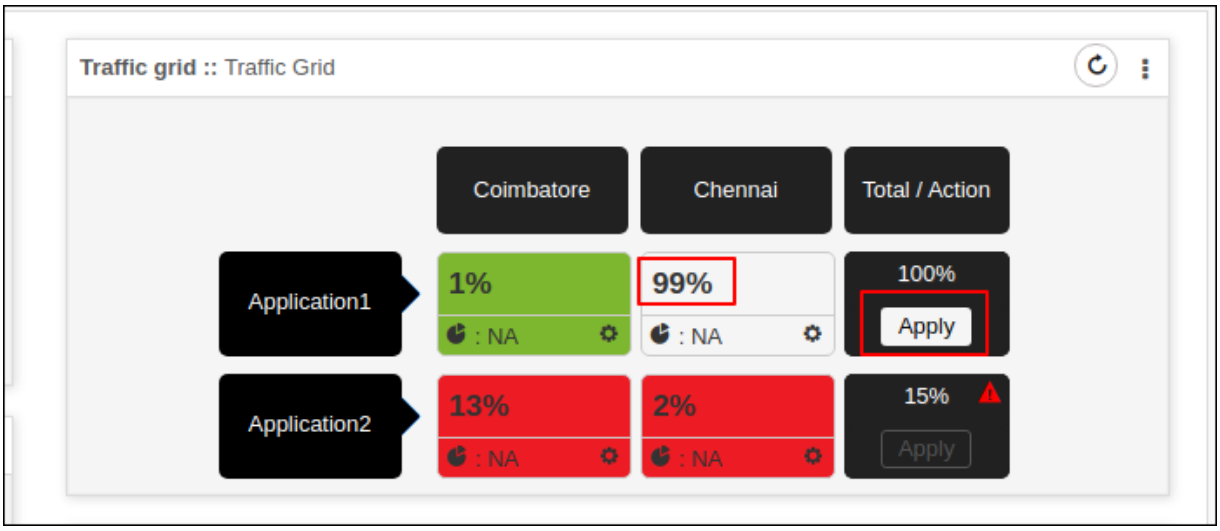
2. Select the objects, and then click **Add**.

   After adding the traffic percentage value is shown on the widget.



3. Modify the traffic percentage values and apply them to the objects by clicking **Apply**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Modify the traffic percentage and click Apply to execute action on the objects added.

> **Note:**
> The sum of all traffic percentages must be 100.

3. Configure the Statistics

By adding objects in the statistics tab, stats for added objects can be monitored.

Select/enter required information.

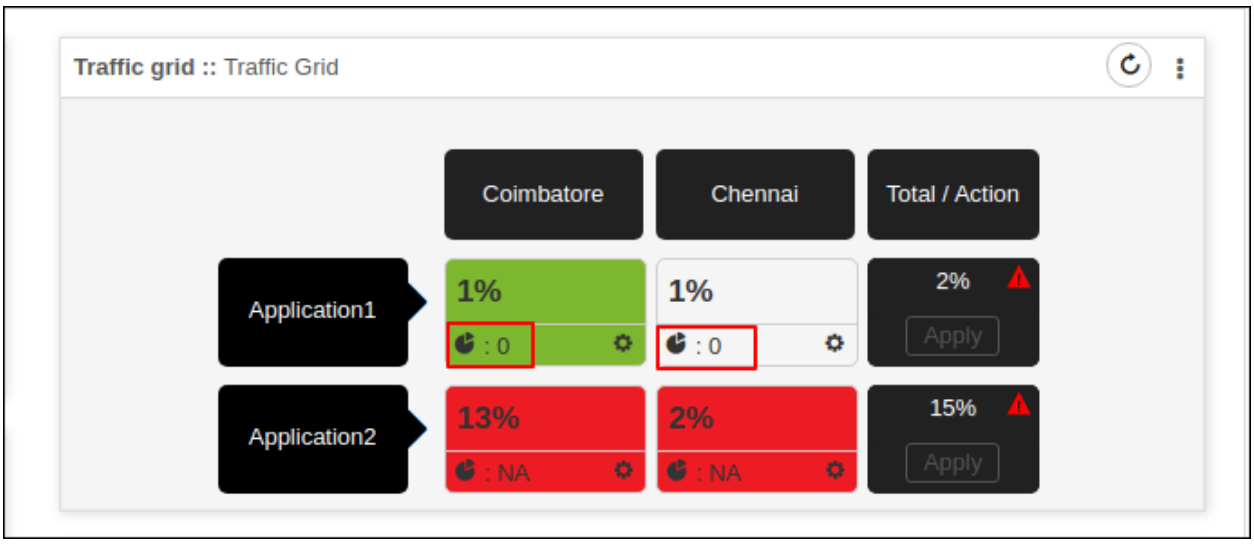
Statistics - Field and Description Table

Name	Type	Description
***Display Name**	Text Box	Enter user defined name for stats to be displayed. **Note:** Only 10 characters are allowed.
***Vendor**	Drop Down	Select a vendor from the drop-down list.
***Device State**	Drop Down	Select a device state from the drop-down list. Available options are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Type	Description
		• **Active**: Active devices are listed. • **Standby**: Standby devices are listed. • **All**: All Managed devices are listed.
***Object type**	Drop Down	All Primary ADC objects are listed. Select an object ype from the drop-down list.
***Object Name**	Drop Down	Object names can be searched and selected. Select an object name from the drop-down list.
***Statistics Type**	Drop Down	Statistics parameters for selected object type are listed. Select a type of statistics from the drop-down list.
**: Mandatory fields*		

Stats collected for the configured objects are displayed in the widget



4. Add Rules For Action

1. **Configure Warning for an action**

   a. Select **Rule type** as **Warning**.

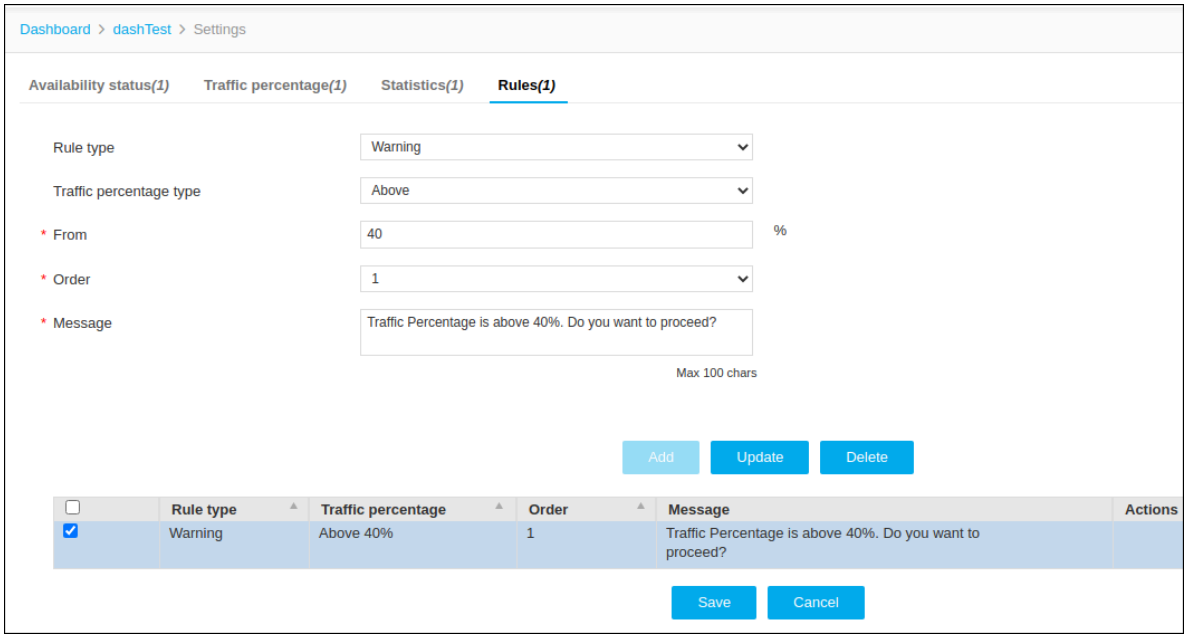
   b. Set the condition to be checked.

   c. Set the order of execution. Maximum 10 rules can be configured.

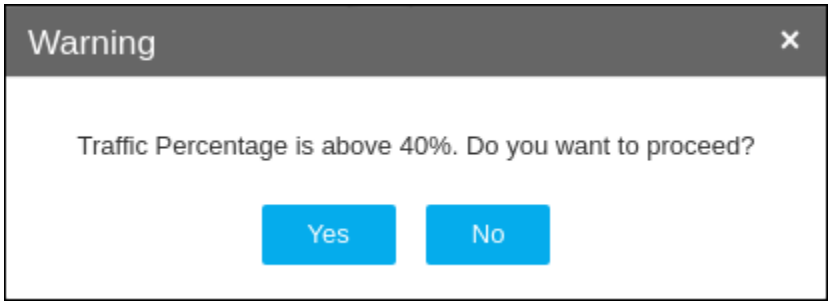
   d. Provide a warning message to be displayed when the condition becomes true.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

e. Click **Add** and **Save**.



On performing action , the warning message is displayed when the condition is true.



2. Configure Restriction for an action.

   a. Select **Rule type** as **Restriction**.

   b. Set the condition to be checked.

   c. Set the order of execution. Maximum 10 rules can be configured.

   d. Provide a Restriction message to be displayed when the condition becomes true.

   e. Click **Add** and **Save**.

   Adding restrictions will not allow the user to perform the action.

3. Configure action when certain condition met.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. Select **Rule type** as **Action**.

b. Set the condition to be checked.

c. Configure Enable/Disable action on objects.

d. Select the **Action Type** and the object on which action has to be executed.

e. Click **Add** and **Save**.

When the condition becomes true, the configured action is executed on the object.

Related information
Application View Widget
Traffic Statistics Widget
Script Execution Widget
Class Management Widget

Class Management Widget

Class management widget is used to execute actions on the Data group of F5 devices. The data group values can be Viewed and Modified.

Configuration

Enter/select the required information.

Class Management Widget Configuration - Field and Description Table

Name	Description
***Widget Name**	Enter the user defined Name for the Widget **Note:** No special characters are allowed, except '_' , '.' , '-' , '|' , ':'. The name should not start with a special character.
***Select task**	Select a task from the drop-down list. • **Create Group**: Create a new Group.Default Group would be the Widget name. • **Modify Group**: Modify the groups added. • **Create Action**: Create View/Modify action.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• **Modify Action**: • Modify the configured actions.
***Group**	List of groups added in the Widget. Select a group from the drop-down list.
***Name**	Enter the user defined name for Groups/Actions.
**: Mandatory fields*	

Create Group

1. Select task as **Create Group**.
2. Provide a **Group name**.
3. Click **Add**.

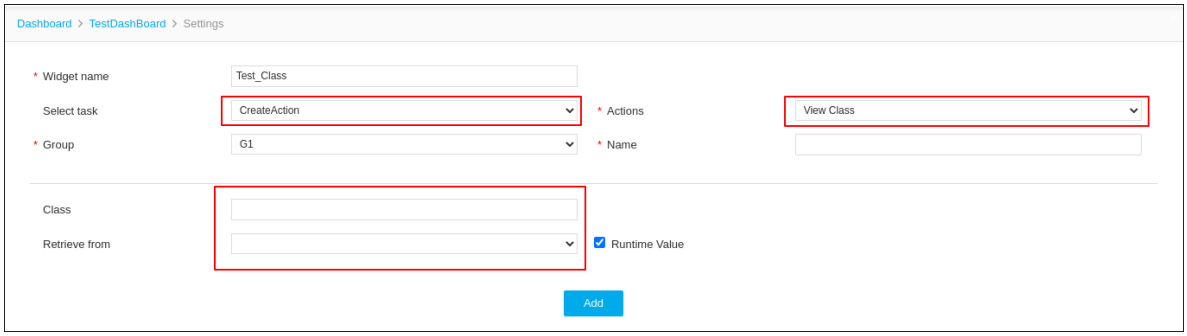
Create Action

1. **View Class**

   a. Select task as **Create Action**.

   b. Select **View Class** in **Actions**.

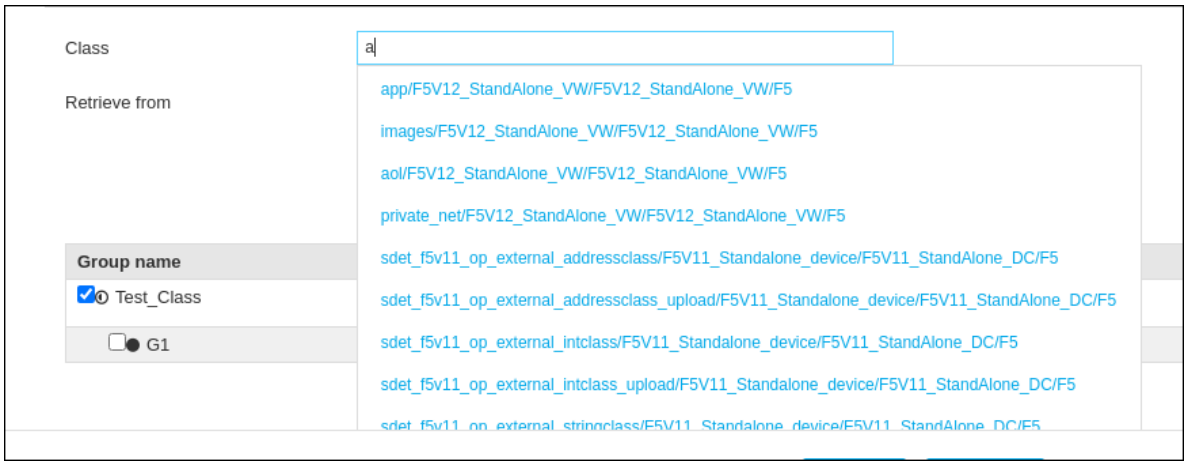
   c. Select the group and provide **Action Name**.



   d. Search the data group name in **Field Class**.

   The list of devices are displayed in **Retrieve From**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



e. Select the data group, and then click **Add** and **Save**.

2. **Modify Class**

a. Select task as **Create Action**

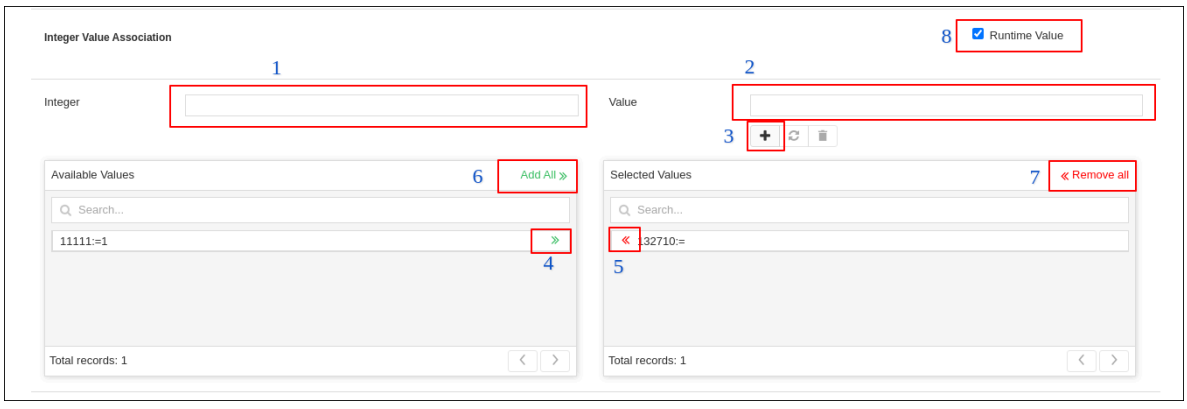
b. Select **Modify Class** in **Actions**.

c. Select the group and provide **Action name**.

d. Search the data group name in **Field Class**.

The list of devices are displayed in **Retrieve From**.

e. Add Attributes to be added in the datagroup.



f. Add **Key** in the **Integer** field.

g. Add value in the **Value** field.

h. Click on the **Add** icon.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The added pair is moved to Available Values.

i. In **Selected Values**, attributes available in the device are listed.

j. Use **Add**/**Remove** and **Add All**/**Remove All** arrows to modify the attributes.

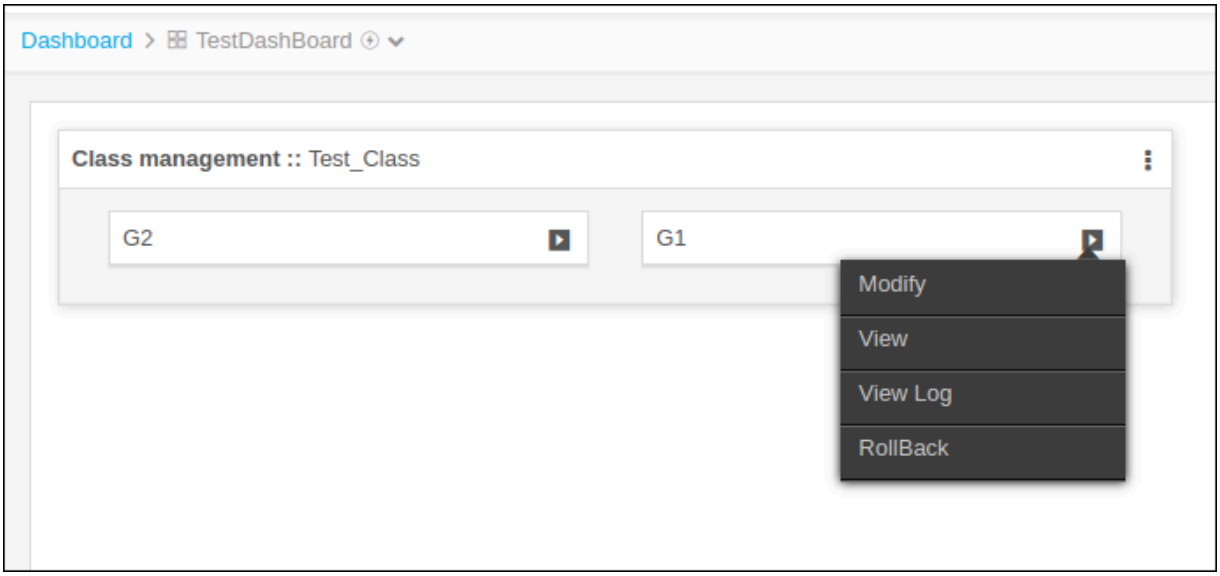
k. Selecting runtime value allows to modify the attributes at the time of execution.

l. Select the devices on which the action has to be executed.

m. Click **Add** and **Save**.

Execution of Configured Actions

1. Click the Play icon on the Group.

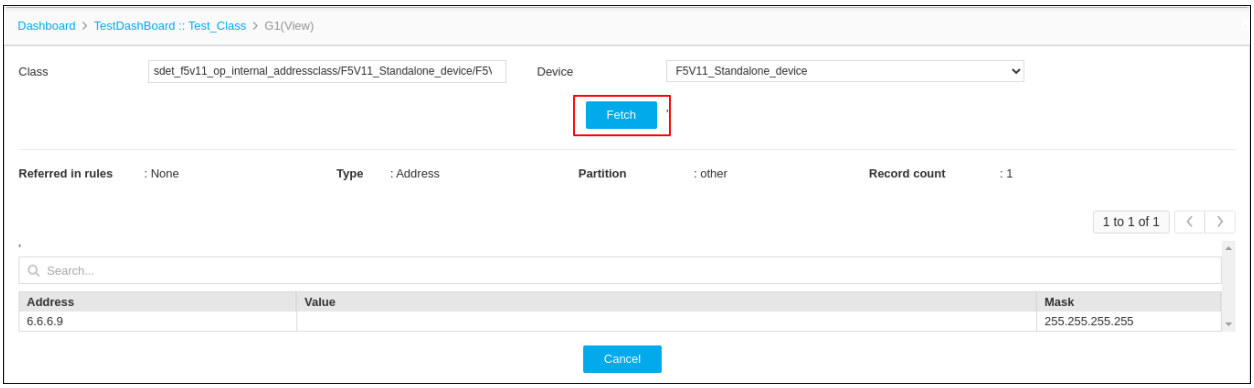


2. Click the action name to be executed.

View Action Execution

View Action will get the data group attributes from the device. Click on **Fetch** to get current values from the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

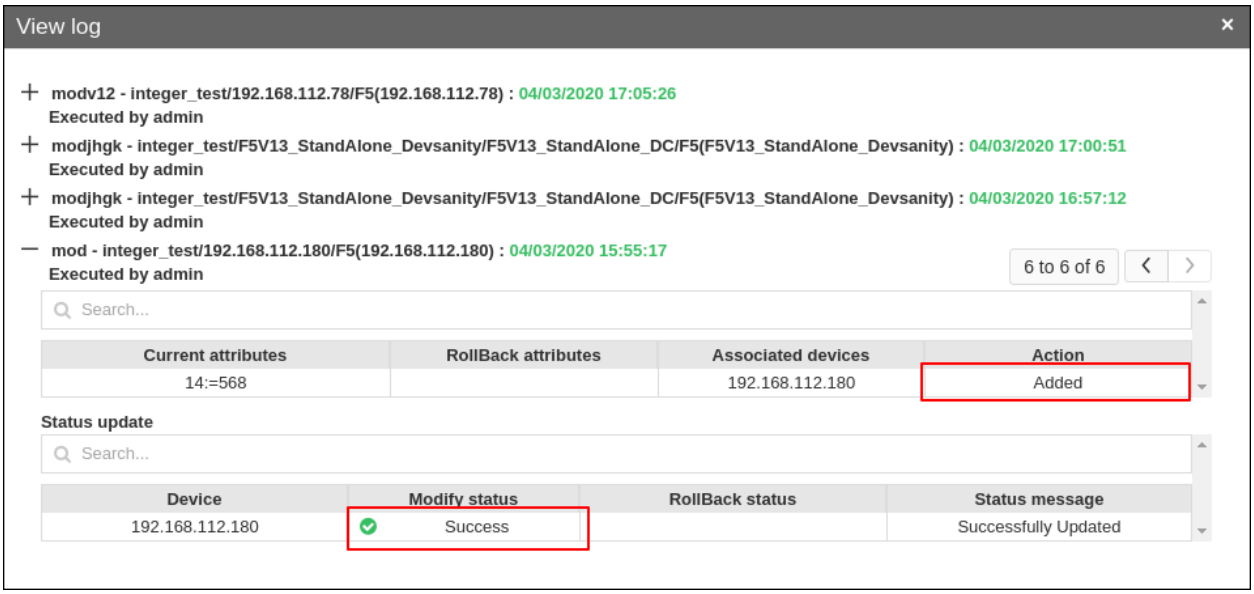


Modify Action Execution

1. Add/Modify the attributes and push it to Device Values.
2. Push the devices on which the action has to be executed.
3. Click **Execute**.

View Logs

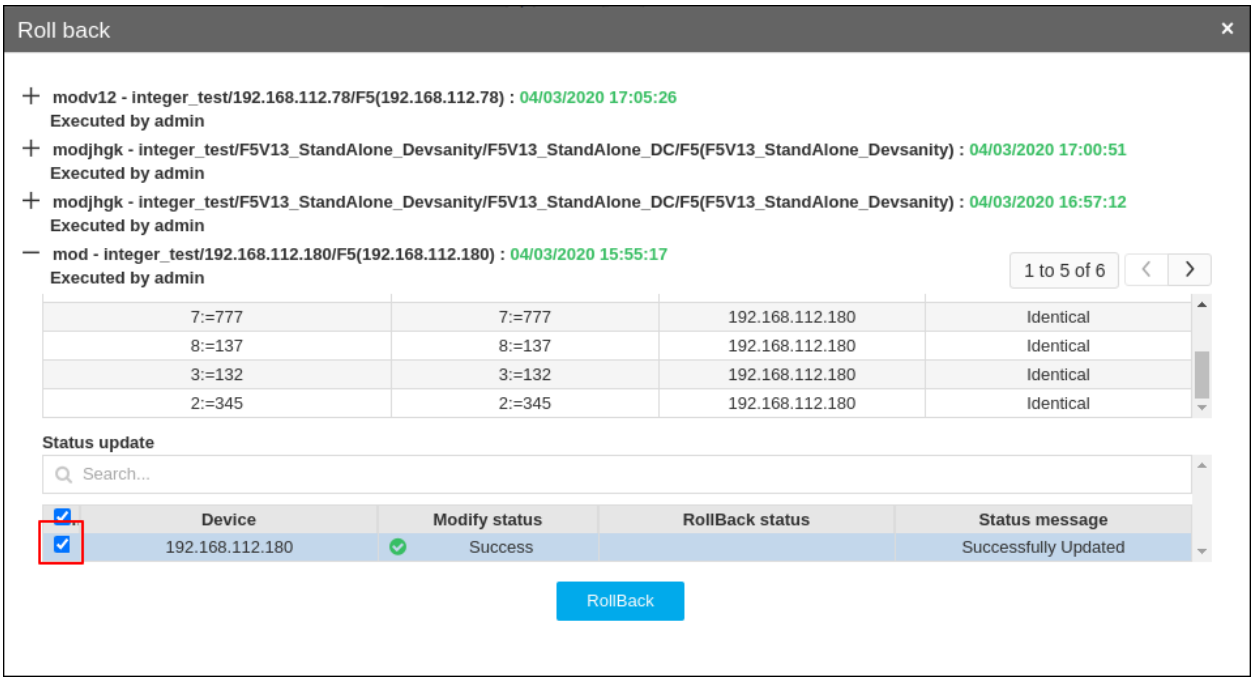
For a particular group each Action Execution logs are maintained. Newly added attributes are easily identified. Action status is also updated. For any failure in action, the failure logs are updated in status message.



Rollback

The last action executed can be rolled back. Select the action, and then click **Rollback**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Related information
Application View Widget
Traffic Statistics Widget
Script Execution Widget
Traffic Grid Widget

Dashboard Actions

This section elaborates on dashboard actions and their corresponding functionalities.

Sharing a Dashboard

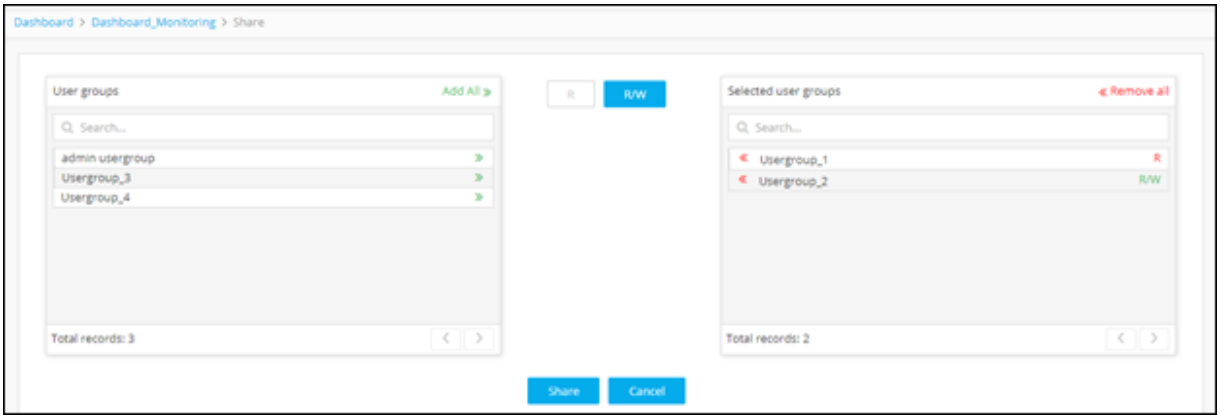
Based on your permissions, you may be able to share a dashboard.

To share a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard table, click the name of the one you want to share.
3. When the dashboard opens, click the  (**Share**) button in the Command bar at the top of the screen. The Share screen appears.
4. Begin the assignment process by clicking the  (**Read-Only**) button. Each role you assign next will have read-only permissions within the dashboard.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. In the Roles field, click the » (**Assign item**) icon beside each role you want to share the dashboard with on a read-only basis.
6. Click the R/W (**Read-Write**) button. Each role you assign next will have read/write permissions within the dashboard.
7. In the Roles field, click the » (**Assign item**) icon beside each role you want to share the dashboard with on a read/write basis.
8. Click the **Share** button to finish.



Deleting a Dashboard

To delete a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one you want to delete.
3. When the dashboard opens, click the (**Delete**) button in the Command bar at the top of the screen.
   A screen pops up, warning you that deleting a dashboard also deletes all widgets on the dashboard.
4. Click **Yes** to continue.

Saving a Widget

By default last created widgets will be displayed first in dashboards. This option can be used when the user wants to organize the widgets in the customized order.

1. Log in to the AppViewX application with valid credentials.
2. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Go to any user defined dashboard.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Users can change the order of the widgets by drag and drop.
5. Click **Save Widgets.**



Renaming a Dashboard

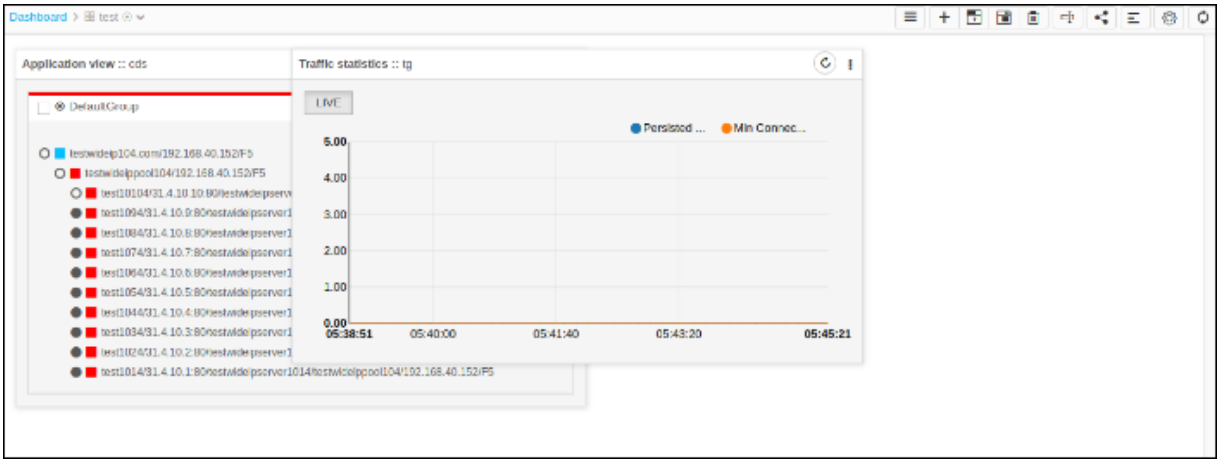
To rename a dashboard:

1. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
2. If you have more than one dashboard, in the dashboard list, click the name of the one you want to rename.
3. When the dashboard opens, click the  (**Rename dashboard**) button in the Command bar at the top of the screen.
4. On the Rename dashboard screen that pops up, enter a new name for the dashboard.
5. Click **Update** to finish.

Align Widget

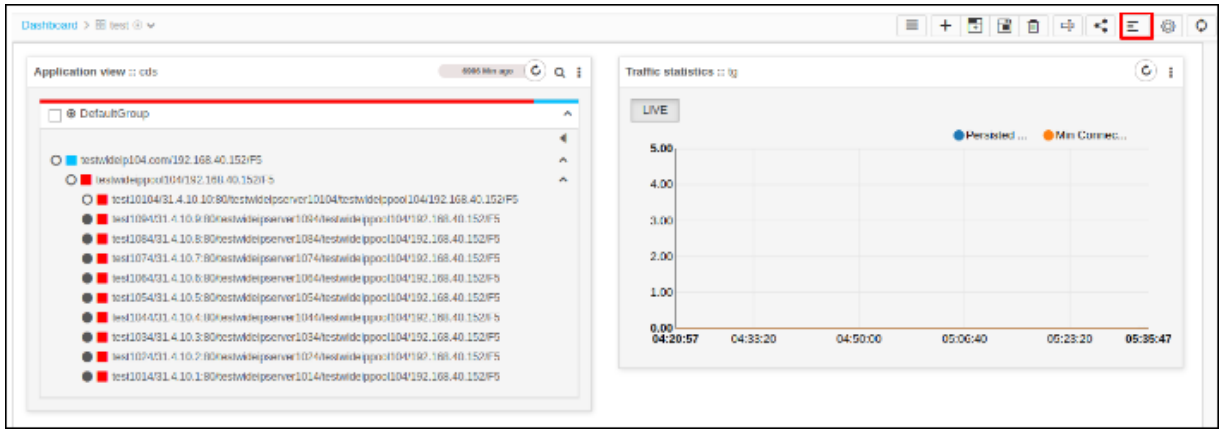
AppViewX gives an option to align the widgets if it gets misplaced.

1. Log in to the AppViewX application with valid credentials.
2. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
   Consider the widget in the dashboard is misplaced.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

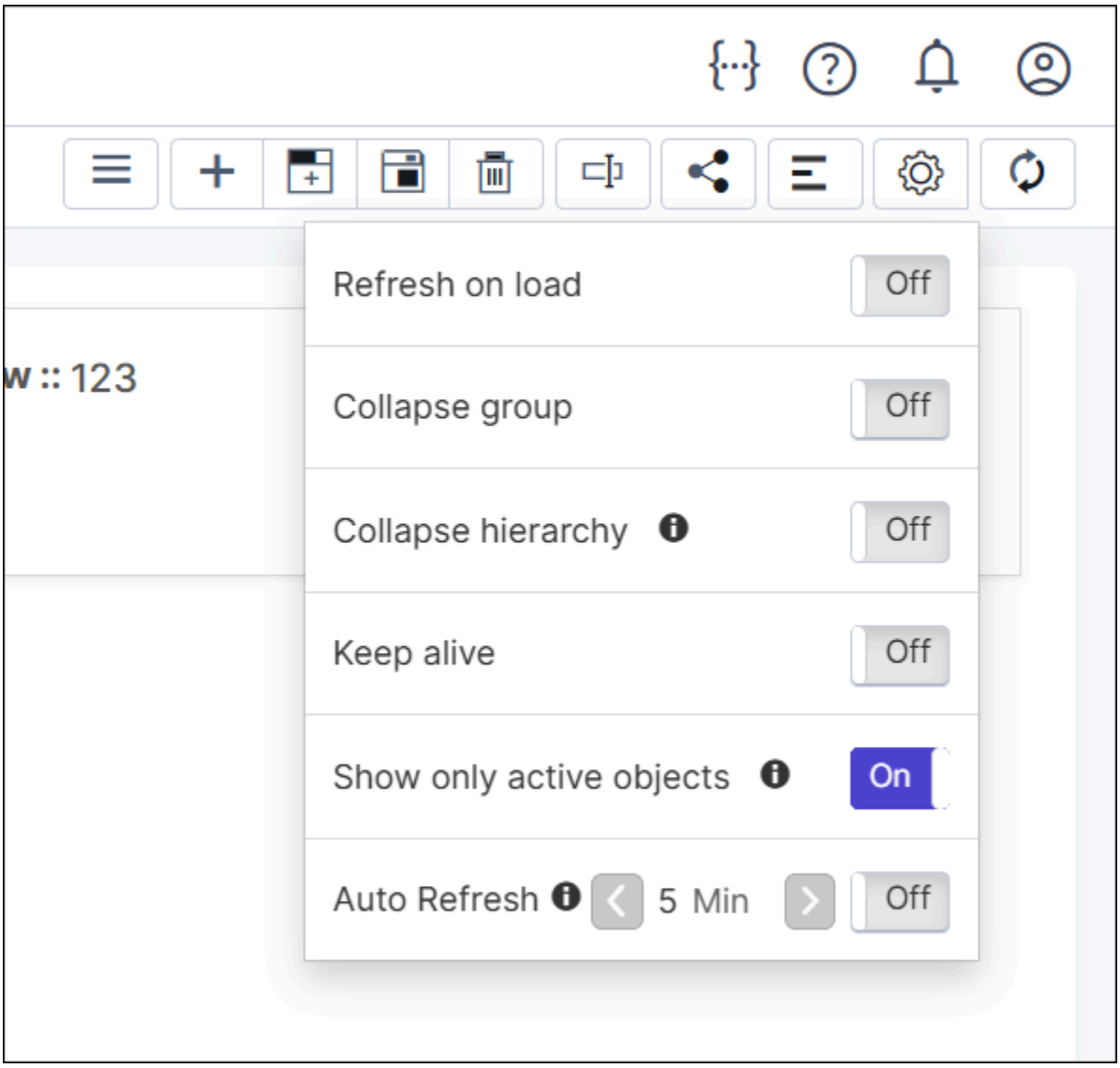
3. To align dashboards in their place click the **Align** icon.



Settings

1. Log in to the AppViewX application with valid credentials.
2. Go to **Menu** > **ADC+** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Click the **Settings** icon.
4. Change the setting by clicking the corresponding slider.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



- **Refresh on load** - If this setting is enabled, each time the dashboard loads, it will trigger the refresh call automatically to load the latest data.
- **Collapse group** - If this setting is enabled, all the groups inside the widgets on the dashboard will display in collapsed format by default.
- **Keep alive** - If this setting is enabled, the auto session expiry function will be disabled. Using this user can monitor the objects without session timeout interruptions.
- **Show only active objects**- If this setting is enabled, the objects of an active device alone be displayed from the configured objects. It is applicable only for **Application view widget**.
- **Auto Refresh** - If this setting is enabled and refresh time is configured, the Dashboard data will be automatically refreshed at the specified time interval. The Auto Refresh option is enabled for users who have the Auto Refresh ACF permission. Users with this permission can set the auto-refresh time

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

interval when creating an Application Widget dashboard or from the Settings > Auto Refresh option. Additionally, the admin user can disable the Auto Refresh option on demand for all users. To enable or disable the Auto Refresh option, see Platform User

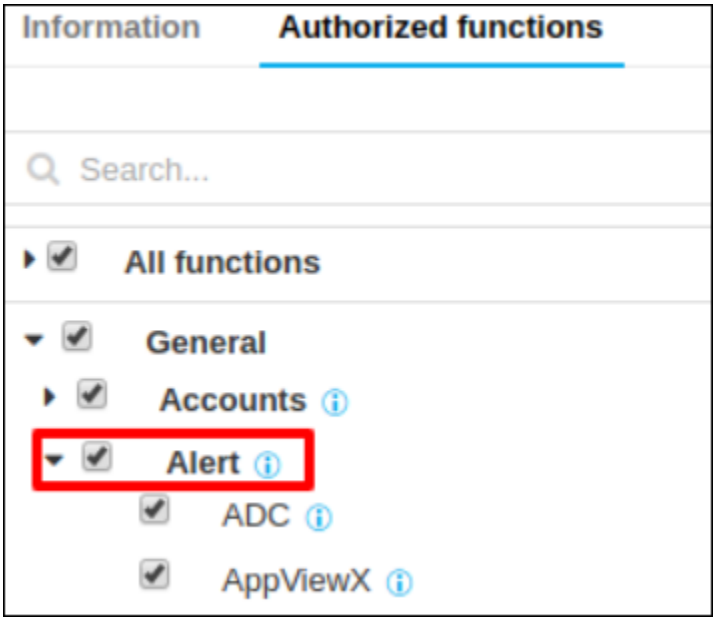
Configure Auto Refresh Setting

To enable or disable the Auto Refresh option, see Platform User guide.

Related information
Alert Management
Add Widgets
Create a Dashboard/Widget

Alert Management

To view/configure alerts, the users should have the ACF permissions.



Create an ADC Alert

To create an ADC alert:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
2. Click (**Settings**) icon on the top.
3. On the **Settings** screen that opens, click the ADC tab if it is not already open.
4. In the **Alert name** box, enter a name for the alert.
5. In the **Alert message** field, enter the message that users will receive for the alert.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

6. In the **Trigger** region, in the **Alert category** field, you can choose from **Threshold Alert**, **Application Alert**, and **Device Alert**.

> **Note:**
> Rather than adding objects manually, you can click the Add search string link and create a search string that automatically assigns all existing objects that match the filter criteria to the alert. The benefit of using a search string rather than selecting objects manually is that the search string continues to work in the background, auto-assigning all new objects to the alert if the objects match the search criteria you set up.

7. From the **Alert severity** dropdown list, select one of the following options:

   - **Critical** - For issues that are causing disastrous results or impacts on functionality. These are top priorities and must be resolved immediately.
   - **Fatal** - For issues that can cause disastrous results or impacts on functionality. These are a major priority and should be resolved soon.
   - **Major** - For issues that are important and require a resolution, but that is not the highest priority.
   - **Minor** - For issues that are of low priority and need a resolution.
   - **Notification** - For issues that are not alerts or warnings, but which must eventually be addressed.

8. In the **Vendor** field, select from the vendor whose device or devices you want to set an alert for.
9. In the **Object type** field, select the vendor object that you want to set an alert for.
   The contents of this field vary depending on the vendor you selected in the previous step.
10. In the **Available** field, click the » (**Assign**) icon beside each object/device you want to add to the alert.
    The following Alert conditions are applicable only for the Threshold alert.

> **Note:**
> To add another condition to the alert, click the (**Add**) button, then in the **Logic** field select **AND** or **OR** to define the relationship between the first condition and the second. AND relationships require both conditions to be met for an alert to be sent, OR relationships require that only one condition be met for an alert to be sent. Only based on the above user-defined conditions, threshold alerts will be raised in AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. In the **Alert interval** field, select how often you want the system to check for breaches of the threshold levels that you are about to define.
b. Checks can be set to occur every 10, 20, 30, 40, 50, or 60 seconds.In the Cool off the period field, select how much time the system should wait before sending another alert about a continuing threshold breach: 10, 20, or 30 minutes.
c. In the **Statistics** field, define the conditions that will generate an alert by selecting values in the **Statistics**, **Operator**, and **Value fields**.

11. To send an email alert, **SMTP** must be configured. Refer to the Configure SMTP for Email Alerting topic for details on how to do this. When you have finished, complete the following steps to use email as an alert method:
    a. Select the **Email Configuration** checkbox.
    b. In the **Email Address** field, enter email addresses to send the alert. Use commas to separate the addresses.
    c. In the **Subject** field, leave the default text or enter the text that briefly describes the kind of alert the user is receiving in their Inbox.
12. To use the **Simple Network Management Protocol (SNMP)** to send the alert, complete the following steps:

    • Enter the **Destination IP** for the alert.

    • Select the **Version** of SNMP you want to use: V1 or V2.

    • Enter the port of the alert that should be used for the alert.

    • Enter the **Community String** for the alert. The string is similar to a user ID or password and allows users to access the requested information on the device.

13. Click **Add** to save the alert to the AppViewX system.



Note:
For the Application and Device alert, when any user executes changes on the configured application/device(s), AppViewX sends a notification based on the appropriate actions associated with the alert (Email/SNMP). Only the changes that are executed via AppViewX will be tracked and notified.

Create a Syslog Alert

AppViewX subscribes to all device-level logs, where it acts as a syslog listener. The logs of any

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

devices added in AppViewX can be viewed as syslog by navigating to Logging->Syslog. However, devices tend to generate huge amounts of data, a Syslog alert is a convenient way to get notified about a specific event that is of importance to you. It also allows for a closed loop remediation by associating workflows.

To create a syslog alert:

1. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
2. Click on  (**Settings**) icon, and then **Syslog** tab.
3. Provide an **Alert Name** and **Message**.
4. Mention the **Severity**, it could be one or multiple.
5. Configure the critical Device/Applications that need to be monitored.
6. Add the Pattern/Regex that needs to be monitored on the Syslog received. Multiple strings can be provided with comma-separated, which will be considered as Boolean AND operator.
7. Following are some of the alerts that can be configured,

   - Sample syslog - <133>Sep 19 04:24:38 bigip-40-152 notice mcpd[6046]: 01070417:5: AUDIT - user admin - transaction #84153993-4 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name \"/Common/testVs\" virtual_server_profile_profile_name \"/Common/tcp\" virtual_server_profile_profile_type 5 virtual_server_profile_profile_context 0 } } [Status=Command OK]\n

   - For instance, if the Syslog alert is configured for the object and the Regex pattern is given as “create” Whenever an object is created and a Syslog is received for that object as above. An alert will be raised for the same and notified to the user.

8. You can also pass certain metadata from the alert to the workflow. In the Metadata section, enter a key and its associated value in the respective fields. This is the additional information that will be used by the workflow that is going to be associated with.
9. Associate any out of the box or custom workflow that needs to be executed on the occurrence of a configured Syslog event.
10. Configure multiple Alerts as needed and Add it to the Grid. The configured Alerts could be modified or deleted anytime by selecting the Alert from the grid.

Change alert settings

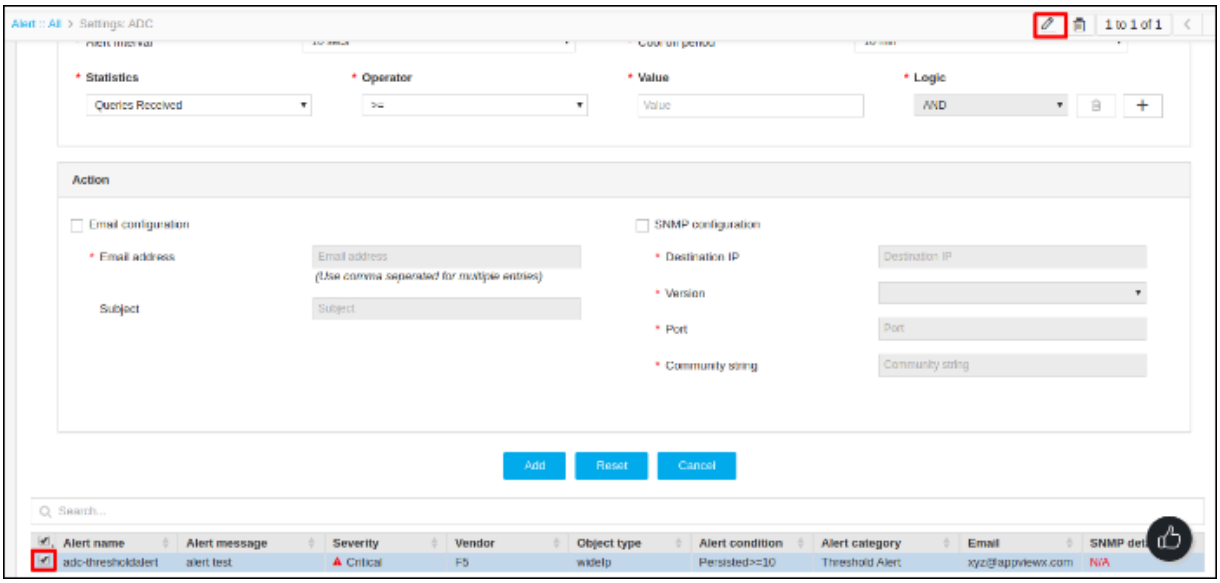
Fields in alert settings may differ based on the alert type, but the process for modifying the alert settings is identical. To modify the alert settings use the below steps:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

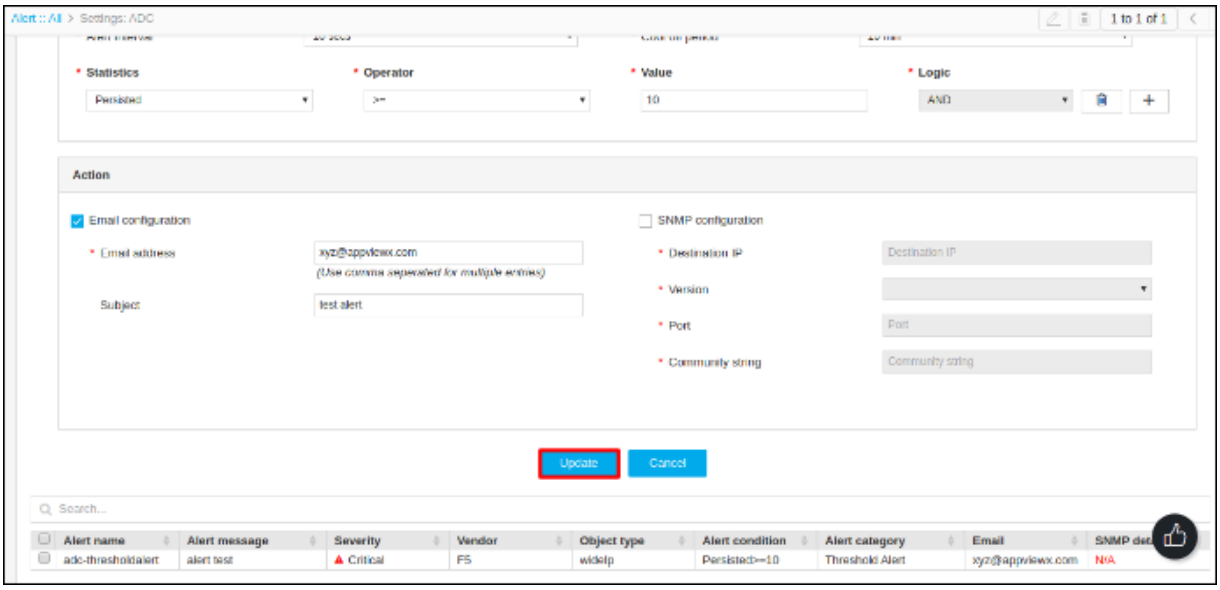
1. Log in to the AppViewX application with valid credentials.
2. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
3. Click on the **Settings** button.



4. To modify the alert settings of a particular alert type click on the corresponding tab. For example ADC or Syslog.
5. Select an alert from the table and the modify button in the top right cornea will be enabled.
6. Click on the **Modify** button and make the required changes.



7. Click **Update**.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

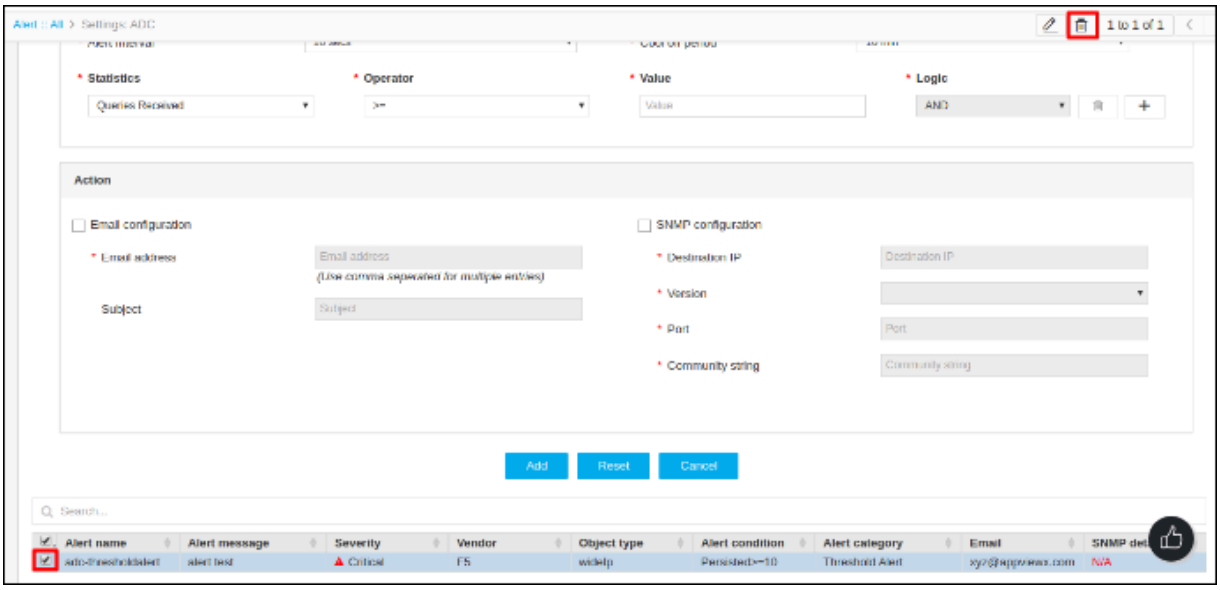
Deleting alerts

Fields in alert settings may differ based on the alert type, but the process for deleting the alert settings is identical. To delete the alert settings use the below steps:

1. Log in to the AppViewX application with valid credentials.
2. Go to **Menu** > **ADC+** > **ALERTS & LOGS** > **Alerts**.
3. Click on the **Settings** button.



4. To delete the alert settings of a particular alert type click on the corresponding tab. For example ADC or Syslog.
5. Select an alert from the table and the delete button in the top right corner will be enabled.
6. Click **Delete**.



Related information
Create a Dashboard/Widget
Add Widgets
Dashboard Actions

System Settings

Device Settings

The device settings can be configured in the following tabs:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Device specification
- Syslog purge limit
- Script Execution
- Device Sync

Related information
Configuration for Device Settings
Object Settings
Object Settings Configuration

Configuration for Device Settings

Configuring Device Specification

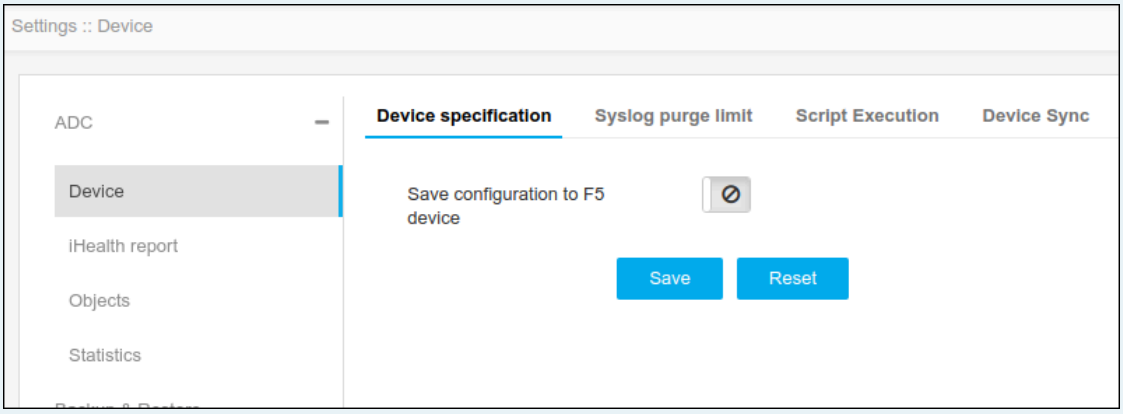
Save the changes of device specification configuration on the F5 device after config fetch. This enables all the changes to be reflected on the F5 file.

To configure a device specification,

1. Click **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Device Specification** tab.
2. Click the toggle icon to enable Save Configuration to the device.

Note:

In this example, F5 device specification is shown.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Click **Save**.

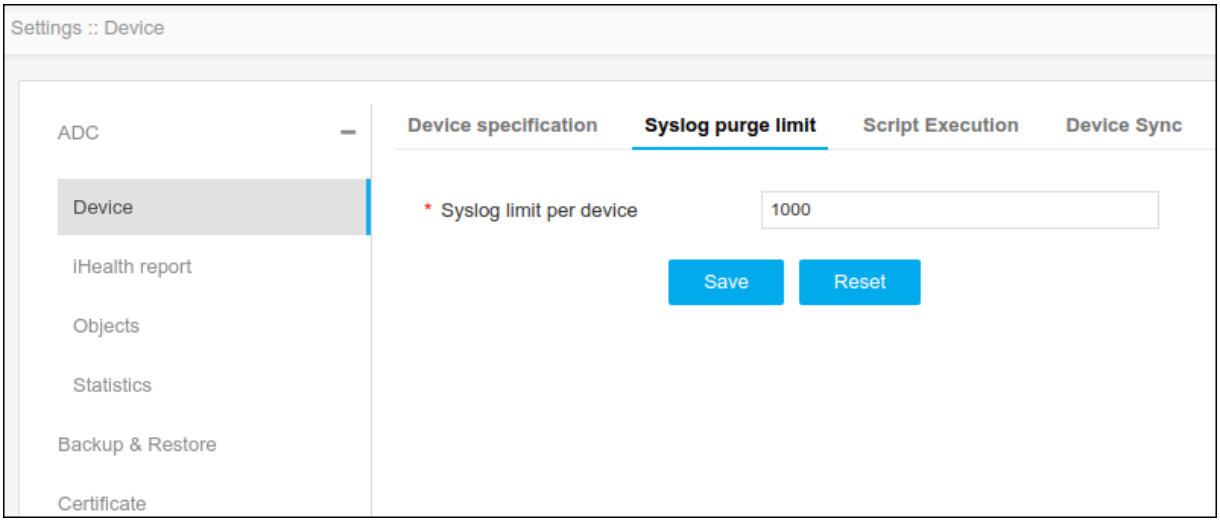
> **Note:**
> To reset the changes, click **Reset**.

Configuring Syslog Purge Limit

Syslog purge limit enables to limit of the number of Syslog persisted against a device. The oldest logs will be automatically deleted on exceeding the limit.

To configure the Syslog purge limit,

1. Click **Menu** >**ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Syslog purge limit** tab.
2. On the Syslog purge limit tab, specify the syslog limit per device.



By default, the limit is set to 1000 and can be extended till 5000.

3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

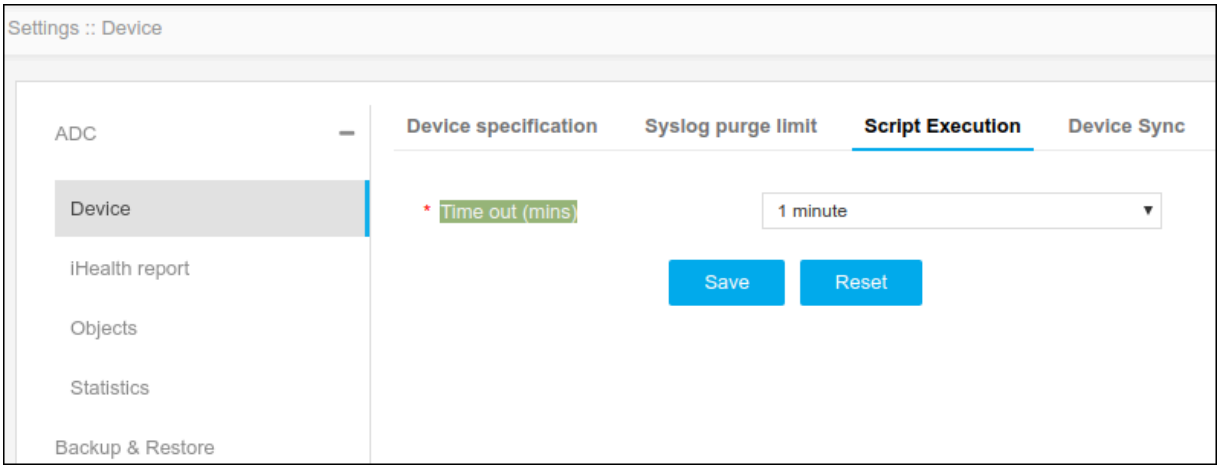
Configuring Script Execution

Script Execution setting allows you to specify the time-out limit of script execution. If the script execution exceeds the specified time, AppViewX will automatically terminate the script.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To configure script execution,

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Script Execution** tab.
2. On the **Script Execution** tab, specify the timeout limit.



3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

Configuring Device Sync

The device Sync feature allows to customize the configuration sync operations.

To customize the configuration sync operations:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Device** > **Device Sync** tab.
2. On the **Device Sync** tab, specify the following:

- **Config fetch** - Allows to schedule the config fetch process at a specific time for selected days. Ensures device changes are reflected in AppViewX.
- **HA - Sync** - Ensures the HA devices are synchronized in specified intervals.

> **Note:**
> To set device sync, the AppViewX Group Sync flag must be enabled while adding a device to run HA sync as per configured interval.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

3. Click **Save**.

> **Note:**
> To reset the changes, click **Reset**.

Related information
Device Settings
Object Settings
Object Settings Configuration

Object Settings

The object settings can be configured under the following tabs:

- Actions
- Naming Format
- Configuration drift

Related information
Device Settings
Configuration for Device Settings
Object Settings Configuration

Object Settings Configuration

Setting Up Object Configuration

On the **Actions** tab, you can customize the actions execution as needed

To customize the actions:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Objects**.
   By default, the **Actions** tab appears.

2. Enable the **Make Comments Mandatory** toggle button to set the comments option mandatory while executing an action from the Control Center or dashboard.
   This ensures end users are mandated to mention the reason for their actions.
3. Enter comments in the **Comments Placeholder**, such as SNOW ID, Enter Application impacted, etc. to impose a standard to be followed upon action execution.
4. Select **Active**, **All Peer**, or **Specific Device** radio buttons for **Execute Action on Device**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

By default, AppViewX actions are triggered on current Active devices. This can be modified as needed.

5. Ensure peers are associated in AppViewX for seamless execution.
6. Click **Save**.

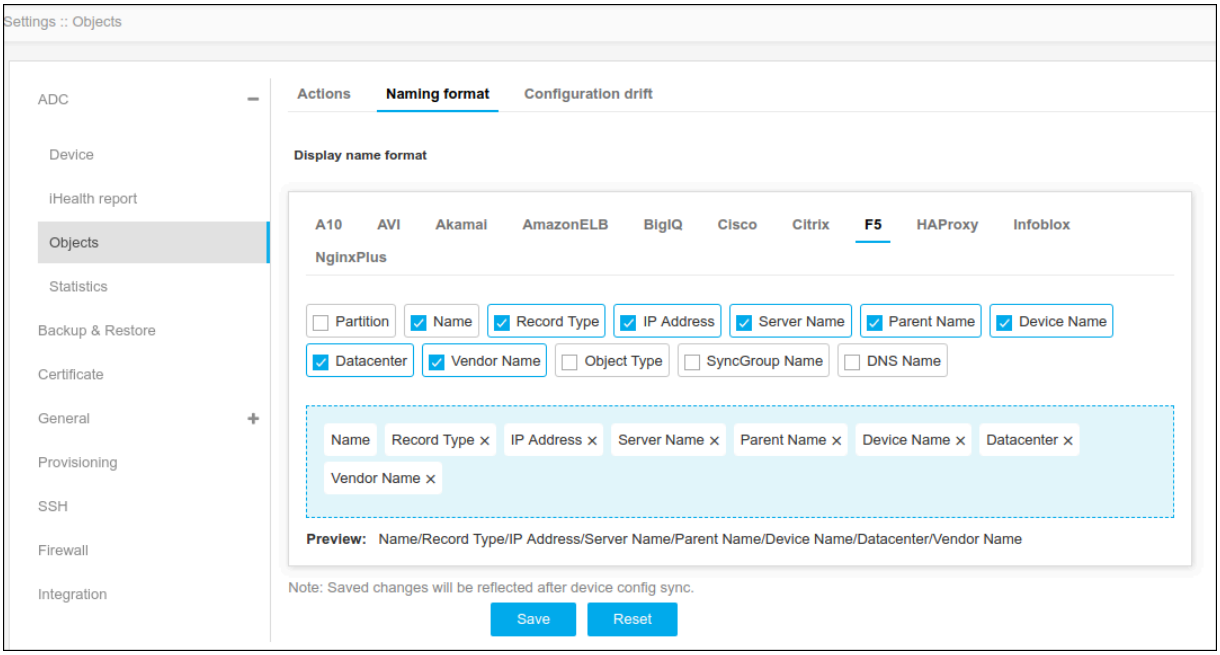
> **Note:**
> To discard the changes, click **Reset**.

Configuring Object Naming Format

On the Naming Format tab, you can customize the display name format of your objects that needs to be followed throughout the application.

To customize the naming format:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Objects**.
2. Click the **Naming Format** tab.



3. On the **Display Name Format** page, choose the vendor to be modified and select the respective **Property**, reorder the fields as needed.
4. Click **Save**.

> **Note:**
> To discard the changes, click **Reset**.

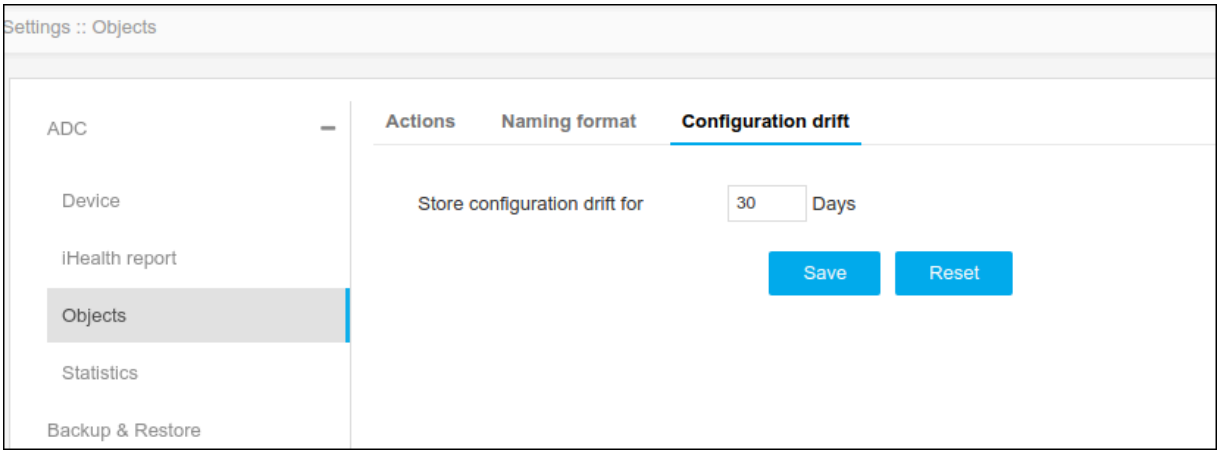
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Configuration of the Drift Storage Limit

AppViewX identifies the object configuration changes (as part of config fetch, Syslog notification, etc.,) and stores the drift to generate reports, governance, and rollbacks. You can customize the number of days the drift must be persisted.

To customize the number of days for the configuration drift:

1. Go to **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Objects**.
2. Click the **Configuration Drift** tab.



3. Enter the store configuration drift value up till 90 days in the **Store Configuration Drift** for the field.
4. Click **Save**.

> **Note:**
> To discard the changes, click **Reset**.

Related information
Device Settings
Configuration for Device Settings
Object Settings

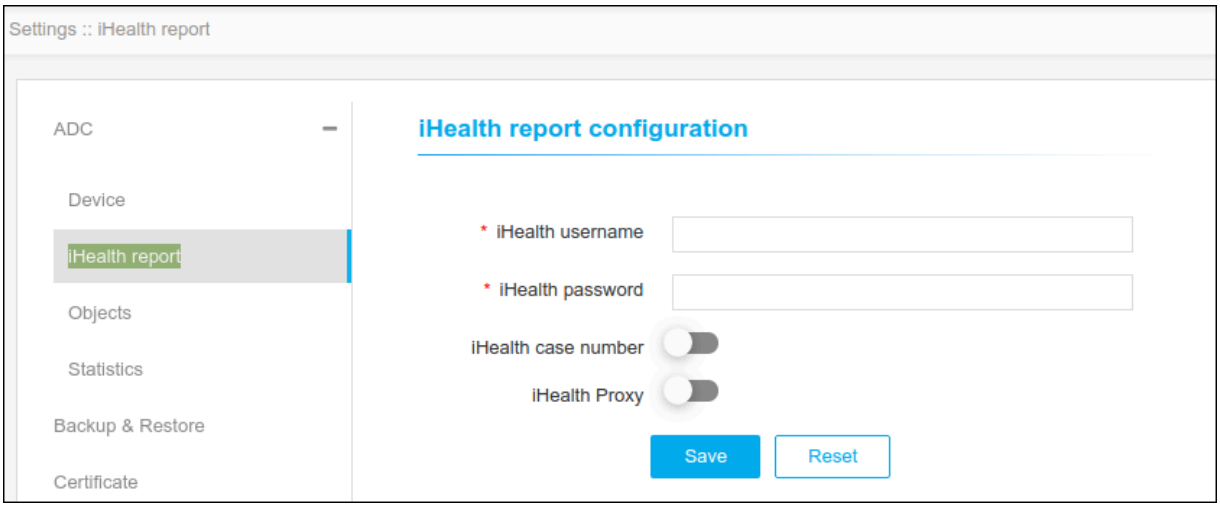
Configuring F5 iHealth Report Settings

Users can configure the F5 iHealth related setting on this page. iHealth is the feature in F5 used to diagnose health of the F5 device. In AppViewX, we collect the qkview file from the F5 device, push to the iHealth site and generate an iHealth report and that will be shown in the device inventory page under the iHealth report column. On this page, users can configure the username and password to be used for logging into iHealth site. In addition to that, users can also enable/disable whether proxy is to be used for iHealth login and whether a case number is mandatory for iHealth report

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To configure the iHealth report,

1. Click **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **iHealth Report**.



> **Note:**
> The iHealth settings configuration for the username and password must be submitted or passed along with the ClientID and ClientSecret according to the new iHealth2 login APIs.

2. On the **iHealth Report Configuration** page, enter the **iHealth username** and **password**.
3. Enable the **iHealth case number** and **iHealth Proxy** toggle buttons to upload the generated qkview file against the mentioned case number for future reference.
4. Click **Save**.

> **Note:**
> To discard the changes, click **Reset**.

Configuring Statistics Collection

In the statistics module, AppViewX collects the statistics of the LTM and GTM objects. It used to monitor the Load and performance of LTM and GTM Objects and devices. Users configure types of objects that need to collect statistics and time intervals at which statistics need to collect.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Configures the statistics data collection to monitor the historic statistics with respect to the vendors and object types.
- Statistics plugin has to be deployed in the node for collecting statistics data.
- AppViewX will collect the stats, aggregate them and provide out of the box reports in the form of dashboards.

To configure statistics,

1. Click **Menu** > **ADC+** > **SETTINGS** > **Module Settings** > **ADC** > **Statistics**.
2. On the **Statistics Configuration** details page, choose the time from the **Time Interval in Minutes** drop-down.
3. Select the vendor, and then choose the object types and devices from the list for which statistics must be fetched.
4. For the device selection, click the **DEVICES** tab, and then,
   - Select the **Type** of device selection as **Device** or **Device Groups**.
   - When you select the **Device** radio button, you get the option to select the devices from the list either by searching or using regex.

     Note:
     If Regex is added for filtering the device, the statistics will be configured for the newly added devices as well, which match the regex.

   - When you select the **Device Groups** radio button, you get the option to select the device groups from the list.
5. Select the default view for the Traffic statistics widget from the **Default View** dropdown. The possible options for default view are:
   - Day (default)
   - Live
   - Week
   - Month
   - Quater
6. Click **Save**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

AppViewX collects the stats, aggregates them, and provides out-of-the-box reports in the form of dashboards.

Note:
To discard the changes, click **Reset**.

Configuration Updates through Syslog

AppViewX functions as a Syslog listener, subscribing to device-level logs. It establishes Syslog notifications with the devices managed within its system, allowing it to detect any changes occurring on these devices. These notifications trigger automatic configuration updates in AppViewX at specified intervals, ensuring that device changes are promptly reflected.

The received Syslog from the devices and the subsequent changes processed by AppViewX in response to these Syslog are meticulously recorded within the logging module.

Syslog Configuration in AppViewX

To successfully subscribe the device to Syslog, it is important to follow the configuration steps in the specified sequence below:

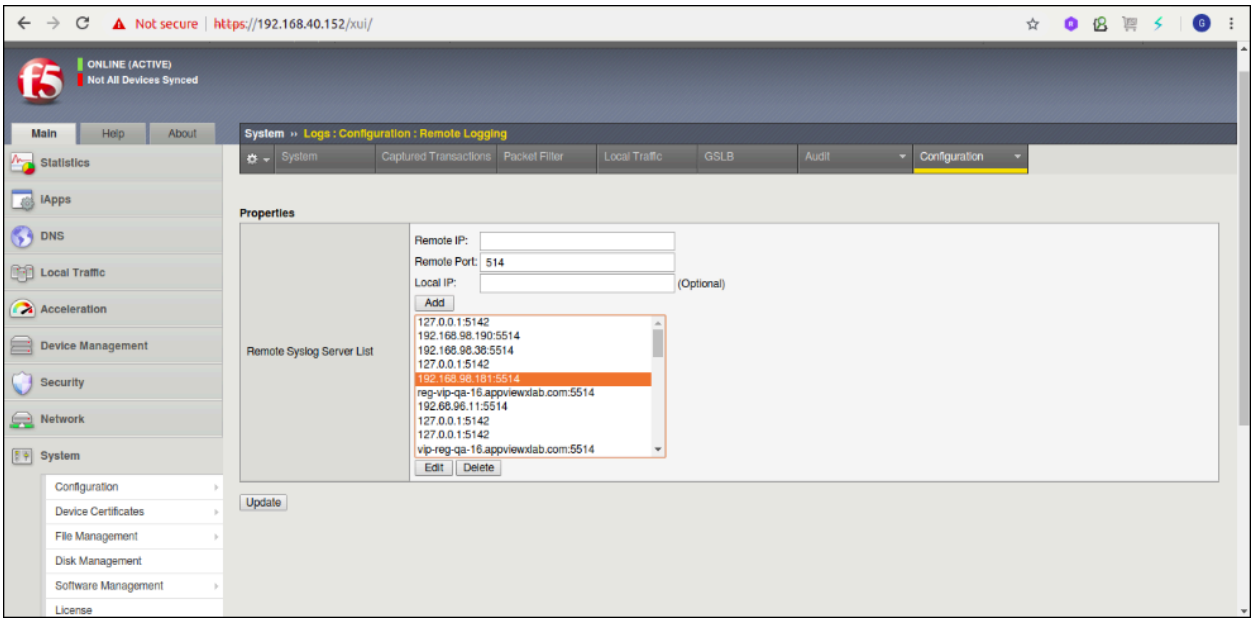
1. Enable the Syslog receiver flag during the cloud connector installation. For instructions on how to set the flag, see AppViewX Cloud Connector guide.
2. Enable **avx_platform_syslog** plugins.
3. Set up the IP address or port for Syslog reception within the Platform. For instructions on how to configure the Syslog reception, see Platform user guide.
4. In case you have an older version of AppViewX and want to make use of Syslog capabilities for ADC SaaS, you must manually activate the Syslog flag by setting SYSLOG_ENABLED=`true` in the path ccpath/deps/properties.

Device Syslog Subscription Example (F5 Sample)

Devices will be subscribed for Syslog only when Syslog flag enabled in settings and is set to true, and designated hosts and port are provided. Syslog subscription for a device is initiated when you select the device and click subscribe in device inventory page. The IP address and port which is Cloud connector IP or VIP IP that is configured in the Platform settings can be subscribed on the device. This can be

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

verified manually in F5 device by navigating to **Main** > **System** > **Logs** > **Remote Logging** window > **Configuration** tab.



In addition to the above mentioned F5 configuration, execute the following commands on each F5 device to enable F5 to transmit the complete FQDN in logs. AppViewX is capable of processing only those Syslogs that contain the entire FQDN of the F5 device, as Host Names could be identical across multiple load balancers.

```
tmsh modify sys syslog include "options {use_fqdn(yes); keep_hostname(no); };"

tmsh save sys config

tmsh restart sys service syslog-ng
```

Vendors Supported

Device Version/Vendor	Object Types
F5 v12 - 17	LTM objects: LTM VirtualServer,LTM Pool,LTM PoolMember,Virtual Address,LTM Node and secondary objects. GTM objects: GTM WideIP,GTM Pool,GTM Server,GTM Datacenter,GTM Virtual Server and secondary objects. State and Status

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Syslog Processing

A new Node.js component has been introduced within the cloud connector to handle Syslog processing. This component initiates its processing by loading predefined information from a configmap.

The preset information is composed of two key components:

1. **mid-server-syslog-grok-patterns-config**: This segment comprises the grok patterns that correspond to the log messages sent from the device to the Syslog component.
2. **mid-server-syslog-grok-filter-config**: The second part contains metadata information. This information outlines the grok patterns that will be applied to the received log messages, specifying the details that should be extracted from these messages. Furthermore, it defines the content that should be forwarded to the subsystem responsible for parsing the message.

Once the preset information has been integrated into the Syslog component, the component will initiate the receiver function within the Kubernetes pod, operating on port 5514. Externally, incoming Syslog messages will be directed to the designated host on port 30025. Following receipt on port 30025, the message will then be forwarded to port 5514 for further processing.

Logs originating from the device will undergo filtration according to the grok filters established within the "mid-server-syslog-grok-filter-config" configmap in the Kubernetes environment. Grok serves as a method to compare a line with a regular expression, map specific parts of the line into metadata fields, and push the matched Syslog message along with metadata field information to avx-platform Syslog pod in Compute cluster for subsystem based parsing.

Based on the filters applied, messages or events are directed to the avx-platform-syslog component. In response to this, log messages are stored within the logging collection. Concurrently, event details are sent to the avx-subsystem-adc. This enables the parsing of real-time updates pertaining to the state and status of an object.

Syslogs can be retained for up to 5000 entries per device. When this limit is exceeded, the logs will be automatically cleared. You can adjust this configuration within the settings.

Scenarios

The following scenarios have been taken into consideration:

- Syslog based updates for Object state change For example: Enabled, Disabled etc.,
    - Real time updates are supported only for F5.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Syslog based updates for Object status change For example: Available, Offline etc.,
  - Real time updates are supported only for F5.

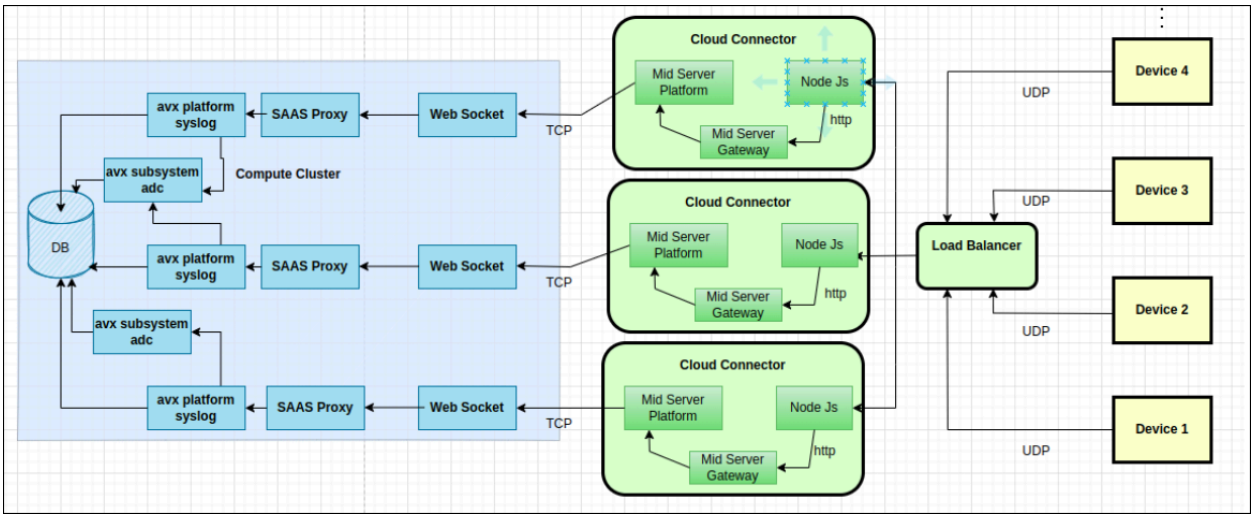
Real-time State/Status Processing for F5

The node-js syslog component filters the state/status Syslog of F5 and forwards the message to the Syslog_Receiver which subsequently transmits them to subsystem adc plugin along with the Syslog details. Within the subsystem_adc plugin, the message undergoes parsing, leading to updates in the state/status of the respective object within the 'me_adc' and 'ro' collections. After the update, Syslog_Receiver adds the Syslog message with a ‘Picked’ status in logging collection. Grok filters are restricted to only the managed object types in AppViewX, which is configurable. The changes will be reflected in the control center/dashboard or other modules wherever the objects are configured which will enable near real time monitoring.

Database Operations for Processing a Syslog Message

- Retrieve device information using the FQDN.
- Update the state of the object in the 'me_adc' collection.
- Update the state of the object in the 'ro' collection.
- Include the Syslog message with a selected status in the Logging collection.

Technical Flow Diagram



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Syslog Based Alerting

As devices often produce substantial data volumes, employing a Syslog alert offers a user-friendly approach to receive notifications regarding critical Syslog information. Users can set up Syslog alerts for specific events or Syslog patterns that hold significance. In the event of a triggered Syslog alert, actions such as sending an email or initiating a workflow can be activated for prompt response.

For Configuring Syslog Alert notification, see Configuring Syslog Alert Notification.

Performance Metrics

Vendors Tested: F5

Load:

1. 300 device and 33lk objects in RO collection
2. 2700 valid Syslog message in 10 seconds send from cc to compute cluster

Observation:

1. No Pod restart observed cloud connecter and Compute cluster
2. Observed CPU utilization 60 to 65% in Database

Recommendation:

1. 1.5 GB of memory is required for Syslogs pod in cloud connector.
2. One Syslog pod in cc, able to process approximately 270 logs per second.

Limitations

- The state status drift functionality will be non-functional as the session ID is not being provided by the node-js component.
- On the Syslog alert configuration, users can only select a maximum of 10 devices or objects from the list of available devices or objects.

Sample Logs

StateChange:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
<133>Sep 19 04:29:30 bigip-40-152 notice mcpd[6046]: 01070417:5: AUDIT - user admin - transaction #84197177-2 - object 0 - modify { virtual_server
{ virtual_server_name \"/Common/testVs\" virtual_server_enabled 0 virtual_server_update_status 1 } } [Status=Command OK]\n
```

StatusChange:

```
<133>Sep 19 04:30:59 bigip-40-152 notice mcpd[6046]: 01071682:5: SNMP_TRAP: Virtual /Common/testVs has become unavailable\n


<133>Sep 19 04:30:41 bigip-40-152 notice mcpd[6046]: 0107183a:5: Virtual Address /Common/1.2.3.16 general status changed from RED to YELLOW.\n
```

Troubleshooting Steps

1. To verify the reception of UDP packets, run the command `tcpdump -ni any port 30025` within the cloud connector.
2. To ensure that the Device VIP is receiving packets, run the command `tcpdump -ni any host VIP_IP_address port 5514` on the F5 device.
3. For confirming if logs are passing through the filters, check the logs within the Syslog pod in the cloud connector environment.

Frequently Asked Questions (FAQ)

1. **How many Syslogs are being received & processed per second in AppViewX?**

   The performance testing of the Syslog components in our local testing environment has confirmed their capability to handle 300 logs per second.

2. **How the processing of Syslog and TRAP events in AppViewX is implemented for actions such as enabling, disabling, and performing forcedowns in F5 devices?**

   Traps are not supported in AppViewX. However, the mentioned actions triggers modifications within the device, subsequently triggering a Syslog event. The processing procedures remain consistent with those outlined in #3.

3. **How AppViewX manages the processing of Syslog and TRAP events for state and status changes in F5 objects?**

   Syslog messages received from the device are grouped into batches and subsequently relayed to the subsystem plugins. Within these Syslog messages, we extract key information including the hostname, object details, and its state and status information. Utilizing the object information obtained, we locate the corresponding entry in our database where a change in state or status occurred. This information is then updated in the database accordingly.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. **Under what circumstances does the state and status of an object not get updated, even after the Syslog message has been parsed?**

   If the hostname is not configured for a device, even when logs are received, it becomes impossible to identify the device where these changes are occurring. Consequently, the state and status of the object won't be updated within AppViewX.

Schedulers Used by ADC+

By deployment, the scheduler below will be activated for ADC specific functionality monitoring purposes. On demand these schedulers can be modified with the help of a support team

Scheduler Name	Description	Frequency
AdcBackUpDeviceJob	BackUpDeviceJob for Adc	Every 30 Minutes
AdcDownloadConfigJob	DownloadConfigJob for Adc	Every day at 00:00:00 AM
AdcDeviceSyncJob	Device sync Job for Adc	Every 15 Minutes
AdcDNSReverseLookupJob	Dns reverse lookup Job for Adc	Every day 6AM & 6PM
SyslogPushJob	Syslog Push for Subsystems	Every 5 Minutes
SyslogCDUJob	Syslog CDU Trigger	Every 1 Minutes
StatAggregationJob	Statistics Aggregation Job.	Every 6 Hrs Start from 00:00:00 Hrs
PurgeRawStatisticsJob	Purge Raw Statistics Job	Every day at 1 AM
PurgeAggregateStatisticsJob	Purge Aggregate Statistics Job	At 01:00:00am, on the 1st day, every month starting in January
UnusedObjectsReportAggregationJob	Unused Objects Report Aggregation Job	Every day at 00:30:00 AM
ADCDashboardStatisticsCollect	Statistics collection for Heatmap and traffic statistics widget	Every 5 seconds
AdcConfigFetchValidatorJob	Validate ConfigJob for Adc in given Interval	Every 15 Minutes

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Scheduler Name	Description	Frequency
ADCConfigDriftPurgeJob	Purge job for config drift logs	Every day at 00:00:00 AM
ADCVipTrafficAggregator	Aggregator job for calculating total traffic served	Every 1 hour
ADCVipTrafficAggregator	Aggregator job for calculating total traffic served	Every day at 00:00:00 AM
ADCApplicationTrafficHourAggregator	Aggregator job for calculating total traffic served by an application	Every 1 hour
ADCApplicationTrafficDayAggregator	Aggregator job for calculating total traffic served by an application	Every day at 00:00:00 AM
AVRLogsProcessingJob	Aggregator job for calculating top talkers to VIP	Every 15 Minutes
PurgeAvrLogsJob	Purge job for top talkers to VIP	At 01:00:00am, on the 1st day, every month starting in January
AdcDeviceHASyncConfigurableJob	Device ha sync configurable Job for Adc	Every day at 10:00:00 PM
AdcBackUpDeviceMonitorJob	Monitor BackUpDeviceJob Scheduler for Adc	Every day at 11:45:00 PM
ADCStateStatusDriftPersist	Persist adc object state status drift	Every day at 03:00:00 AM
ADCStateStatusDriftPurge	Purge adc object state status drift	Every Sunday at 03:00:00 PM

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Chapter 3: ADC+ API Guide

This guide provides information on the APIs to use for managing, automating, orchestrating, and observability for application delivery services.

API Reference

This applies only to AppViewX customers. To try the AppViewX exposed APIs on Swagger, follow these steps:

1. From the top right corner of your screen, click {...} **(API Reference)** icon.

   > **Note:**
   > The **API Reference** icon is displayed for the Admin user and users with ACF permission enabled for API Reference.

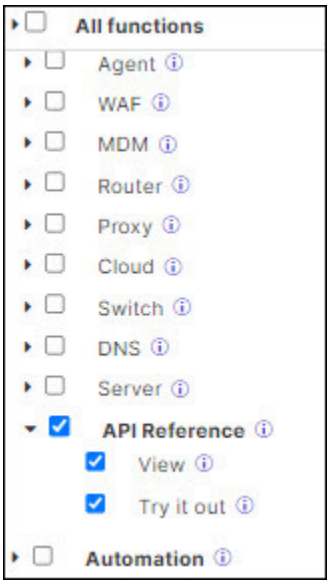
   You will be redirected to **AppViewX API Reference** page.
2. Click the required API to **Try it out**.

Enable ACF Permission for API Reference

To enable the ACF permission for non-admin roles to access the API reference, follow these steps:

1. Go to **Platform** module **IDENTITY** > **Role** > **Authorized functions**.
2. In the **Authorized functions** section, expand the **Platform**, by clicking the ▸ icon.
3. To assign **API Reference**, select the checkbox beside **API Reference**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Click **Save**.

Best Practices for Working with the AppViewX API

- **Use appropriate HTTP methods.**

  Ensure that the correct HTTP method is used for each operation (e.g., GET for retrieval, POST for creation).

- **Handle errors gracefully.**

  Implement proper error handling in your application to manage API responses.

- **Use secure storage.**

  Store access tokens securely and avoid hardcoding them in your application code.

- **Implement pagination.**

  For endpoints that return large datasets, implement pagination using limit and offset parameters.

Understanding the AppViewX ADC+ API

The AppViewX API is a programmatic way to get data in and out of the AppViewX subsystems. With access to RESTful AppViewX APIs, you can leverage the raw potential of AppViewX. It provides a powerful way to channel the data into native business applications. This document comprises of module-wise APIs used in AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

RESTful HTTPS Requests

Type	Description
GET	GET requests, retrieve resource representation/information only and not to modify it.
POST	POST APIs create new subordinate resources. For example, a file is subordinate to a directory containing it or a row is subordinate to a database table. In terms of REST, POST methods are used to create a new resource into the collection of resources.
PUT	PUT APIs are used to update existing resources (if a resource does not exist then API may decide whether to create a new resource or not).
DELETE	DELETE APIs are used to delete resources (identified by the Request-URI).

Requests

Each endpoint URL is built in the same way by the following structure:

```
http://<IP/HostName/TenantName>:<GWPORT>/avxapi/<Endpoint>?<gwsource>
```

The following section provides an explanation of each component of the sample URL structure used in API requests. For quick reference, this section is referenced in all the API topics as **Reference** in this guide.

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Request Structure

All endpoints accept a request structure that should consist of JSON formatted data. To ensure the request is accepted, set the header **Content-Type: application/json**.

The following example shows a request to add a resource:

```
{
"payload": {
"name": "resource_1",
"description": "This is a sample resource."
}
}
```

Response Structure

The Content-Type of the response is typically determined by the Content-Type header, and for most endpoints, it will be application/json. All requests that reach the server, regardless of the response code, will retrieve a response body. A successful request will contain a body with the requested information, for example:

```
https://appviewxapi.com/avxapi/resource?gwsource=external
```

Returns the following JSON structure that a resource is added:

```
{
"response": "Resource added successfully",
"message": "Resource added successfully",
"appStatusCode": null,
"tags": null,
"headers": null
}
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Description of Server Responses

HTTP Code	Response Message
200 OK	The request was successful (some API calls may return 201 or 202 instead).
400 Bad Request	The request is not understood or required parameters are missing.
401 Unauthorized	Authentication failed or the user doesn't have permissions for the requested operation.
409 Forbidden	Access denied.
404 Not Found	Resource not found.
429 Too many requests	The number of requests to the service has crossed the threshold.
503 Service unavailable	The client cannot communicate with the service.
504 Gateway timeout	The given request has exceeded the expected time.

URI Scheme

- **Host :** {url}
- **BasePath :** /avxapi
- **Schemes :** HTTPS
- **URL :** https://{url}/avxapi

Types of Accounts in AppViewX

There are two types of accounts in AppViewX:

- **User Accounts**: These are used by actual users.
- **Service Accounts**: These are used by system services such as web servers, automation tools, and so on.

AppViewX recommends using a Service Account for accessing APIs from automation tools. Service Accounts are supported with oAuth standard for a more secure and standard way of accessing APIs.



Note:
AppViewX supports both User Account and Service Account for accessing APIs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Authentication using a Service Account

For accessing APIs, you can login via two types of accounts:

- User account

  A **user account** represents an individual person interacting with the application or the system. User accounts are used for accessing the system on behalf of a user.

  For accessing APIs with a user account, you need to get the session ID by providing a username and password in the login API. This session ID can then be used for accessing other APIs.

  > **Note:**
  > You can also use the username and password in all API calls instead of the sessionId. However, this is not recommended.

- Service account

  A **service account** represents a non-human entity such as an application or a service. It is used for automated processes or system-to-system interactions without human intervention.

  For accessing APIs with a service account, you need to get the Access Token by providing Client ID and Client Secret in get-service-token API. This Access Token can then be used for accessing other APIs.

  > **Note:**
  > Access Token Validity is 30 minutes by default and it can be configured in **Settings** > **Authentication** > **oAuth Settings**.

For accessing APIs with a service account, the subsequent sections will help you with instructions to:

Retrieve session ID using login API

This API used to retrieve the session ID using the login API for secure authentication and access to system resources.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

- Make sure you have valid login credentials (Username and Password) for accessing the system.
- You cannot use OAuth credentials (Client ID and Client Secret) for login.
- To access the APIs using the service token, use the API with the Service Account.

Request Structure

Endpoint	`/login`
Type	`POST`
Sample URL	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/login?&gwsource=external` To understand the elements of the sample URL, click here.
Headers	
Content-Type	application/json
Request timeout period	15 minutes

Understanding the sample URL

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Input Parameters

	Description
username *Header*	(Mandatory) Use login name of the user. **Type:** String **Example:** "admin"
password *Header*	(Mandatory) Password for the username. **Type:** String **Example:** "AppViewX@123"
otp *Header*	(Mandatory only if MFA is enabled) If MFA is enabled, enter the OTP received on your registered email ID in the header. Multifactor authentication (MFA) is a security mechanism that requires users to provide two or more verification factors to gain access to a resource If MFA is enabled, and you try to login with only the username and password, you will get the following error upon execution of the API: **MFA is enabled. We have sent an OTP to your email ID: aaa********r@appviewx.com**. In this case, ensure that the OTP is included in the header and try logging in again. **Type:** String **Example:** "OTP : **609700**"
Content-Type *Header*	(Mandatory) The parameter should be set to `application/json` to specify the nature of the data in the payload. **Type:** String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameters (continued)

	Description
	Example: "application/json"
gwsource *Query*	(Mandatory) Source from which the request is triggered. The values can be: • web • external **Type:** String

Response Structure

- **Status Code:** 200 Ok
- **Message:** Login Successful
- **Headers:**
  - **Content-Type:** application/json

Response Parameters

Name	Description
response	The response contains the attributes needed to retrieve the session ID.
message	Success message or failure description in case of error.
appStatusCode	Application specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Name	Description
status	Indicates the overall status of the response. The values can be: • SUCCESS • FAILURE
appStatusCode	An application-specific status code, if applicable.
statusDescription	Description of the status, if available.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
sessionId	Unique identifier for the session.
lockDownPeriod	Number of login attempts remaining.
termsAccepted	
passwordExpiryMsg	
emailId	

Status Codes

HTTP Code	appStatusCode	Response Message
200 OK	NA	Login successful
400 Bad request	ACCT_AUTH_001	Username or password cannot be null or empty.
401 Unauthorized	ACC_AUTH_022	Login failed. Invalid credentials.
401 Unauthorized	ACC_AUTH_006	Login failed. Invalid credentials.

Sample Request/Response

Use Case

Log in to the application with a username and password.

Sample Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/login?&gwsource=external
```

Request Payload

```
{}
```

Sample Response


```
{
   "response": {
      "status": "SUCCESS",
      "appStatusCode": null,
      "statusDescription": null,
      "sessionId": "avx~~c73a4f56-f4ab-4cdf-aadf-6d90bf406077",
      "authCode": null,
      "lockDownPeriod": 15,
      "emailId": null,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
        "termsAccepted": true,
        "passwordExpiryMsg": ""
    },
    "message": "Login successful.",
    "appStatusCode": null,
    "tags": null,
    "headers": null
}
```


What can you do next?

After the sessionID ris etrieved using the login API, you can now use the session ID for API calls.

Using Session ID for further API calls

The sessionID retrieved using the login API can be used in the header for making further API calls.

In this section, as an example, we are using the session ID with the API call for manaing a device.

Before you begin

- Session ID is obtained from the login API.
- Ensure that the session ID is valid and has not expired.

Request Structure

Endpoint:	/adc-device-manage
Type:	PUT
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-manage?gwsource=external` To understand the elements of the sample URL, click here.
Headers	
Content-Type:	application/json

Understanding the sample URL

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Mandatory	Description	Field Type	Constraints
deviceName *String[]*	Yes	(Mandatory) Name of the device(s) to manage. **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.		

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	State change to manage the device has been initiated.	-
400 Bad Request	ADC_INV_0234	Device name is mandatory.Value cannot be null or empty.	Check and ensure a valid value is provided in field deviceName.
400 Bad Request	AVX-VLDTN-001	Invalid payload data.	Check and ensure a valid value is provided in field payload.
404 Not Found	ADC_INV_0010	Device not found.	Any of the given devices is not found in the database. Ensure if all the requested devices are added in inventory.
403 Forbidden	ADC_INV_0008	Selected device(s) are not in un-managed state. Operation cannot be performed.	All the requested devices are not in unmanaged state. Requested devices should be in unmanaged state to perform manage action.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Change the state of device ADC_DEVICE_1 from unmanaged to managed.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-manage
```

Sample Request

```
{
  "payload": {
    "deviceName": [
      "ADC_DEVICE_1"
    ]
  }
}
```

Sample Response

```
{
  "response": "State change to manage the device has been initiated",
  "message": "State change to manage the device has been initiated",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Authentication using a User Account

For accessing APIs, you can login via two types of accounts:

- User account

  A **user account** represents an individual person interacting with the application or the system. User accounts are used for accessing the system on behalf of a user.

  For accessing APIs with a user account, you need to get the session ID by providing a username and password in the login API. This session ID can then be used for accessing other APIs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

> **Note:**
> You can also use the username and password in all API calls instead of the sessionId. However, this is not recommended.

- Service account

  A **service account** represents a non-human entity such as an application or a service. It is used for automated processes or system-to-system interactions without human intervention.

  For accessing APIs with a service account, you need to get the Access Token by providing Client ID and Client Secret in get-service-token API. This Access Token can then be used for accessing other APIs.

> **Note:**
> Access Token Validity is 30 minutes by default and it can be configured in **Settings** > **Authentication** > **oAuth Settings**.

For accessing APIs with a user account, the subsequent sections will help you with instructions to:

Retrieve Access Token using get-service-token API

The API provides a streamlined process for retrieving service tokens related to account management tasks.

Before you begin

- Make sure you have valid login credentials for accessing the system.

Request Structure

Endpoint:	`/acctmgmt-get-service-token`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/acctmgmt-get-service-token?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Authentication:	Yes
Request timeout period	15 minutes

Understanding the sample URL

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Input Parameters

	Description
Authorization *Header*	(Mandatory) Please form a string in this format <Client ID>:<Client Secret> and do base64 encoding. Then prepend a key 'Basic' before the encoded value. Final value should be "Basic <EncodedValue>". **Type:** *String* **Example:** "admin"

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameters (continued)

	Description
Content-Type *Header*	(Mandatory) The parameter should be set to `application/json` to specify the nature of the data in the payload. **Type:** *String* **Example:** "application/json"
grant_type *Payload*	(Mandatory) Payload Type should be "Form". The value of the param should be "Client_Credentials". **Type:** *Text*

Response Structure

- **Status Code:** 200 Ok
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Response Parameters

Name	Description
response	The response contains the attributes needed to retrieve the access token.
message	Success message or failure description in case of error.
appStatusCode	Application specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Code	appStatusCode	Response Message
200 OK	NA	Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Code	appStatusCode	Response Message
400 Bad request	ACCT_SA_003	Service account is invalid/not found::[Service account not found in the database]
400 Bad request	OAUTH_CLNT_015	Client Password is incorrect::[Invalid Client credential]
400 Bad request	ACCT_SA_001	Invalid Request::[Invalid client Id or secret]
500 Internal Server Error	avx-common-011	Error while processing.

Sample Request/Response

Use Case

Retrieve Access Token.

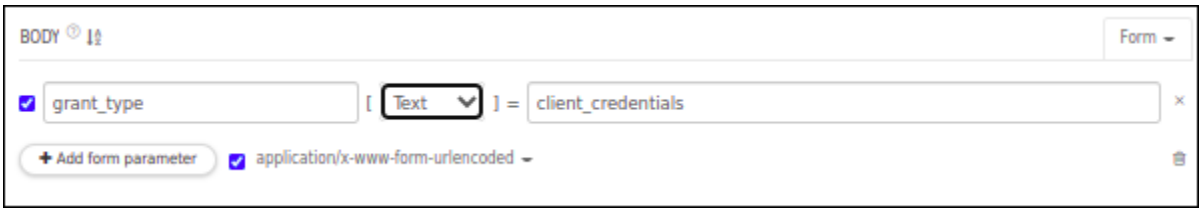
Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/acctmgmt-get-service-token?gwsource=external
```

Content-Type: application/x-www-form-urlencoded

Authorization: Basic
NTIxYzNhZDItZWE0ZS00NDdiLWE1MWItOTYyMWJiN2VhMTI2OjU1QVUjTk84JSpaaGd2TmZhWVVtdHZYMGRrRWhvZ

Request Payload



Sample Response


```
{
"response":
 "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJwbGF0Zm9ybSIsImF1ZCI6ImF2eCIsImNsaWVudElkIjoiOTcwNzRlNDEtOGFmOS00NTZkLTlhNjQtZjBj
 NGJiOTA4MDQ4IiwiaXNzIjoiYXZ4IiwiZXhwIjoxNjUwMzY5MzY3LCJncmFudCB0eXBlIjoiY2xpZW50X2NyZWRlbnRpYWxzIn0.HZnkuUEjXIeqJWqpqiNWFHqIDI7
 GYf4cWx6VwbjGD_0",
"message": null,
"appStatusCode": null,
"tags":{},
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
"headers": null
}
```

Using Access Token in the header for further API calls

The access token retrieved using the get-service-token API can be used in the header for making further API calls.

In this section, as an example, we are using the access token with the API call for managing a device.

Before you begin

- Access Token is obtained from the get-service-token API.
- Ensure that the Access Token is valid and has not expired.

Request Structure

Endpoint:	`/adc-device-manage`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-manage?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Understanding the sample URL

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Input Parameters

Name	Description
Token *Header*	(Mandatory) Use token retrieved from login API. **Type:** *String* **Example:** eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJwbGF0Zm9ybSIsImF1ZCI6ImF2eCIsImNsaWVudElkIjoiOTcwNzRlNDEtOGFmOS00NTZkLTlhNjQtZjBjNGJiOTA4MDQ4IiwiaXNzIjoiYXZ4IiwiZXhwIjoxNjUwMzY5MzY3LCJncmFudCB0eXBlIjoiY2xpZW50X2NyZWRlbnRpYWxzIn0.HZnkuUEjXIeqJWqpqiNWFHqIDI7GYf4cWx 6VwbjGD_0
gwsource *Query*	(Mandatory) Source from which the request is triggered. The values can be: • web • external **Type:** *String*
Payload *String*	(Mandatory) Input data for request body in application/json format. For payload details, see Payload section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Mandatory	Description	Field Type	Constraints
deviceName *String[]*	Yes	(Mandatory) Name of the device(s) to manage. **Constraints**: It can contain only alphanumeric characters, ‘-’ , ’_’ , ‘.’ , ‘*’ , ‘|’, ‘!’.		

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	State change to manage the device has been initiated.	-
400 Bad Request	ADC_INV_0234	Device name is mandatory.Value cannot be null or empty.	Check and ensure a valid value is provided in field deviceName.
400 Bad Request	AVX-VLDTN-001	Invalid payload data.	Check and ensure a valid value is provided in field payload.
404 Not Found	ADC_INV_0010	Device not found.	Any of the given devices is not found in the database. Ensure if all the requested devices are added in inventory.
403 Forbidden	ADC_INV_0008	Selected device(s) are not in un-managed state. Operation cannot be performed.	All the requested devices are not in unmanaged state. Requested devices should be in unmanaged state to perform manage action.

Sample Request/Response

Use Case

Change the state of device ADC_DEVICE_1 from unmanaged to managed.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-manage`

Sample Request

```
{
  "payload": {
    "deviceName": [
      "ADC_DEVICE_1"
    ]
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }
}
```

Sample Response

```
{
  "response": "State change to manage the device has been initiated",
  "message": "State change to manage the device has been initiated",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Object

The objects of the devices can be grouped related to the application to understand the hierarchy and traffic flow of the application. By using the object APIs, you can search, view, manage all the entities, configurations, or objects.

Search for Object

Search for objects that are already discovered in AppViewX to monitor or to perform actions on it. This API will return results based on the resources assigned to the user.

Before you begin

Before attempting to search for objects with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Infrastructure view**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-objects-search`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-objects-search?gwsource=external` To understand the elements of the sample URL, click here.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Mandatory	Description	Field Type	Constraints
input *Key value pair*	Yes	(Mandatory) Input filter key and value to perform the search. Multiple key and value pairs can also be mentioned.		Possible values: Input
filter*Key value pair*	No	(Optional) Filter params to be applied on response such as start,limit and sort.	Key value pair	Possible value: Filter

Response Structure

200 OK, returns string of type application/json with the following body params:

Name	Description
response *String*	Contains the requested objects config data.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Response

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
objects *Object[]*	List of objects from the database, that matches the input query.
totalRecords *Integer*	Total no of records matching the search query.
obtainedRecords *Integer*	Total no of records returned considering the filter query.
obtainedRecordRange.start *Integer*	Start index for the range of records returned considering the filter query.
obtainedRecordRange.end *Integer*	End index for the range of records returned considering the filter query.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Matching results found for the given input.	-
400 Bad request	ADC_OBJECT_SEARCH_001	Invalid search input.	Check and ensure if a valid value is given in the request payload field - input.
400 Bad Request	ADC_OBJECT_SEARCH_003	Invalid object type.	Given object type value under request payload field ‘objectType’ is not supported.
400 Not Found	ADC_OBJECT_SEARCH_005	Device not found.	Check if the device name given in the payload request field - deviceName is valid and is available in inventory.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
404 Not Found	ADC_OBJECT_SEARCH_004	No matching records found.	No results were found for the given search query. Refine the search query.

Input Keys

The following are the most commonly used keys to search for an object managed in AppViewX. Apart from these, there are more keys that are available in the database parsed against each object, that can be used as a search param.

Name	Description
deviceName *String*	(Optional) Name of the device added in inventory. It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.
objectType *String*	(Optional) ObjectType to query. Possible values of objectTypes can be found here.
vendor *String*	(Optional) Vendor of the device. Eg.: F5, AVI, etc.
destinationIp *String*	(Optional) Field denoting the IP of the virtual server object in F5. The IP address must be in IPv4 format.
destinationPort *String*	(Optional) Field denoting the port of virtual server object in F5. It must be between 1 and 65535.
partition *String*	(Optional) Partition of the object. It must contain alphanumeric values, '-' ,'.', '_' and space.
name	(Optional) Name of the object.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
ip *String*	(Optional) IP value of the object. The IP address must be in IPV4 format.

Filter Keys

Name	Description
start *Integer*	(Optional) Start index of the response to the display. The default value is taken as 0.
max *Integer*	(Optional) Number of entries from a start index to display. The default value is taken as 99. Max = 100
sortColumn *String*	(Optional) Column on which sort is to be applied. Possible Values are "displayName", "name", and "sortLevel".
sortOrder *String*	(Optional) Sorting order. Possible values: asc,desc.

Sample Request/Response

Use Case

View all the Wide IPs in the system with the name testGw.com.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-objects-search?gwsource=external`

Sample Request

```
{
  "payload": {
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "input": {
    "objectType": "WideIp",
    "name": "testGw.com"
  },
  "filter": {
    "start": 1,
    "max": 100,
    "sortColumn": "displayName",
    "sortOrder": "asc"
  }
 }
}
```


Sample Response


```
{
 "response": {
  "objects": [
   {
    "alias": "",
    "code": "gw",
    "color": "AVAILABILITY_STATUS_BLUE",
    "configData": "gtm wideip /Common/testGw.com {\n    pools {\n        /Common/asdf {\n            order 0\n            ratio 4\n        }\n    }\n}\n",
    "dashboardObjectName": "testGw.com(192.168.40.150)",
    "description": "",
    "deviceId": "5f36bed67620c40e97e30348",
    "displayName": "testGw.com/192.168.40.150/F5",
    "gp": [
     "gp:@5f36bed67620c40e97e30348:@asdf:@Common"
    ],
    "ipv6NoErrorNegTtl": "",
    "ipv6NoErrorResponse": "disabled",
    "last-resort-pool": "none",
    "lbmode": "round-robin",
    "name": "testGw.com",
    "partition": "Common",
    "persistCidrIpv4": "",
    "persistCidrIpv6": "",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
      "persistence": "DISABLED_PERSISTENCE",
      "persistenceTtl": "",
      "poolPropertyMap": {

      },
      "status": "ENABLED",
      "statusCode": "UNKNOWN ENABLED",
      "vendor": "F5",
      "resource_id": "gw:@5f36bed67620c40e97e30348:@testGw.com:@Common"
    }
  ],
  "totalRecords": 1,
  "obtainedRecords": 1,
  "obtainedRecordRange": {
    "start": 1,
    "end": 1
  }
 },
 "message": "Matching results found for the given input.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Object Hierarchy
- Get Object Configuration

Get Object Hierarchy

Search for objects that are already discovered in AppViewX to monitor or to perform actions on it. This API will return results based on the resources assigned to the user.

Before you begin

Before attempting to get an object hierarchy with this API, users must ensure any one of the ACF permission is available from any of the following:

- **ADC** > **ControlCenter** > **Infrastructure view**
- **ADC** > **ControlCenter** > **Application view**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-object-hierarchy`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-object-hierarchy?gwsource=external`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

	To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Body	

Payload

Payload Details

Name	Description
object Key-value pair	(Mandatory) The composite key values that have to be given to identify an object. See the Composite Keys for F5 WideIp Composite key section for details.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the requested objects config data.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	Topology successfully retrieved for the given object.	-
400 Bad Request	ADC_CC_0514	Invalid object details	Check and ensure if a valid value is given in payload request field - ‘object’.
400 Bad Request	ADC_DASH_0528	Invalid device name	Check and ensure if payload request field ‘object.deviceName’ is given.
400 Bad Request	ADC_DASH_0526	Invalid Vendor	The vendor is not supported. Check for possible valid vendors under field ‘tags.validVendors’ in the response.
400 Bad Request	ADC_DASH_0512	Exception: Invalid Object type	Check if a valid non-null/non-empty value is given in the payload request field ‘object.objectType’. If a valid value is given, the given object type might not be supported. Check for possible valid object types under field ‘tags’ in the response.
404 Not Found	ADC_INV_0010	Device not found.	Device not found in database. Check if the device name given under field deviceName of the object property is available in the inventory.
404 Not Found	ADC_INV_2003	No Object Found for the request	Check the composite keys and values given under the payload field ‘object’.
403 Forbidden	ADC_CC_0006	Permission is not available for given objects.	Check if the requested object has either Read/Read Write permission.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Retrieving hierarchy of the wide IP object "www.appviewx.com" from partition "Common" on device 192.168.112.78.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-object-hierarchy?gwsource=external

Sample Request


```
{
  "payload": {
    "object": {
      "deviceName": "192.168.xxx.xx",
      "objectType": "wideIp",
      "partition": "Common",
      "objectName": "www.appviewx.com",
      "recordType": "a"
    }
  }
}
```


Sample Response


```
{
  "response": {
    "_id": "gw:@5ea9637de0dc314883cbd04f:@www.appviewx.com:@Common:@a",
    "alias": "",
    "code": "gw",
    "color": "AVAILABILITY_STATUS_RED",
    "configData": "gtm wideip a /Common/www.appviewx.com {\n    pools {\n        /Common/gtm_pool_ipv6 {\n            order 0\n            ratio 99\n        }\n    }\n}\n",
    "dashboardObjectName": "www.appviewx.com(192.168.xxx.xx)",
    "description": "",
    "deviceId": "5ea9637de0dc314883cbd04f",
    "displayName": "www.appviewx.com/a/192.168.xxx.xx",
    "gp": [
      "gp:@5ea9637de0dc314883cbd04f:@gtm_pool_ipv6:@Common:@a"
    ],
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
"gr": [
],
"ipv6NoErrorNegTtl": "",
"ipv6NoErrorResponse": "disabled",
"last-resort-pool": "none",
"lbmode": "round-robin",
"name": "www.appviewx.com",
"partition": "Common",
"persistCidrIpv4": "",
"persistCidrIpv6": "",
"persistence": "DISABLED_PERSISTENCE",
"persistenceTtl": "",
"poolPropertyMap": {
},
"recordType": "a",
"status": "ENABLED",
"statusCode": "OFFLINE ENABLED",
"vendor": "F5",
"lastStatusSyncTime": 1597830719934,
"children": [
 {
  "_id": "gp:@5ea9637de0dc314883cbd04f:@gtm_pool_ipv6:@Common:@a",
  "alternate": "round-robin",
  "avaiablityValue": "",
  "avaliablityRequirement": "all",
  "cname": "",
  "code": "gp",
  "color": "AVAILABILITY_STATUS_RED",
  "configData": "gtm pool a /Common/gtm_pool_ipv6 { }\n",
  "dashboardObjectName": "gtm_pool_ipv6(192.168.xxx.xx)",
  "deviceId": "5ea9637de0dc314883cbd04f",
  "deviceName": "192.168.xxx.xx",
  "displayName": "gtm_pool_ipv6/a/192.168.xxx.xx",
  "dynamicRatio": "disabled",
  "fallback": "return-to-dns",
  "fallbackIp": "0.0.0.0",
  "gm": [
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
],
"gpm": [
],
"gtr": [
],
"gw": [
  "gw:@5ea9637de0dc314883cbd04f:@www.appviewx.com:@Common:@a"
],
"gwChild": [
],
"limitConnections": "",
"limitbps": "",
"limitpps": "",
"manualResume": "disabled",
"maxAddrReturned": "1",
"monitors": [

],
"name": "gtm_pool_ipv6",
"parentName": [
  "www.appviewx.com"
],
"partition": "Common",
"preferred": "round-robin",
"qualityofService": {
  "qos-lcs": "30",
  "qos-hit-ratio": "5",
  "qos-hops": "0",
  "qos-packet-rate": "1",
  "qos-topology": "0",
  "qos-vs-score": "0",
  "qos-kilobytes-second": "3",
  "qos-vs-capacity": "0",
  "qos-rtt": "50"
},
"ratioMap": [
 {
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
            "partition": "Common",
            "recordType": "a",
            "name": "www.appviewx.com",
            "_id": "gw:@5ea9637de0dc314883cbd04f:@www.appviewx.com:@Common:@a",
            "order": "0",
            "ratio": "99"
          }
        ],
        "recordType": "a",
        "status": "ENABLED",
        "statusCode": "OFFLINE ENABLED",
        "ttl": "30",
        "vendor": "F5",
        "verifyMemberAvailability": "enabled",
        "lastStatusSyncTime": 1597830719934,
        "ratio": "99",
        "order": "0",
        "children": [


        ]
      }
    ]
  },
  "message": "Topology successfully retrieved for the given object.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **HostName**: A human-readable label assigned to a device (host) on a network

  The hostname will be included in the endpoint URL for an on-prem deployment.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Object Configuration
- Composite Keys for F5 WideIp

Composite Keys for F5 WideIp

Name	Description
deviceName	(Mandatory) Name of the device added in inventory. It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'. **Type**: String
objectType	(Mandatory) ObjectType to query. It accepts alphanumeric characters and spaces. **Type**: String
partition	(Mandatory) Partition of the wide IP object. It accepts alphanumeric characters, '-' , '.' , '_' and space. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
objectName	(Mandatory) Name of the wide IP object. **Type**: String
recordType	(Mandatory) Record type of the Wideip object (Needed only for F5 versions above v12). Possible values: A, AAAA, CNAME, etc. **Type**: String

Get Object Configuration

This API retrieves the configuration of an object as stored in the AppViewX inventory. It can retrieve the latest configuration and also the configuration is known at a given time if any.

Note:
The configuration is not fetched from the device in real-time but from the AppViewX inventory.

Before You Begin

Before attempting to get an object configuration with this API, users must ensure any of the ACF permission is available from the following:

- **ADC** > **Control Center**, **ADC** > **Dashboard** > A**pplication Widget**
- **ADC** > **Orphan objects**

This can be checked under **Accounts** > **Role** > **Authorized Functions**

Request Structure

Endpoint:	`/adc-object-configuration`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-object-configuration?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
objectId *String*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.
timeStamp *Long*	(Optional) To get the known configuration at the given time, mention the time stamp as well. If not mentioned, the latest known configuration will be returned.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the requested objects config data.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	Configuration found for the given object.	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_INV_1554	Config data not found for given object details	Check if given timeStamp and objectId is valid.
403 Forbidden	ADC_INV_1558	Cannot access resource, permission not available.	Check if Read/Read Write permission is available for the requested object.
404 Not Found	ADC_INV_2003	No Object Found for the request	Check the resource id given under field ‘objectId’ is valid.

Sample Request/Response

Use Case

Get the known object configuration available at the time 1549362738957(time in milliseconds) for the GTM WideIP object with resource id - **gw:@5f36bed67620c40e97e30348:@testGw.com:@Commo**

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-object-configuration?gwsource=external

Sample Request

```
{
  "payload": {
    "objectId": "gw:@5f36bed67620c40e97e30348:@testGw.com:@Common",
    "timeStamp": 1549362738957
  }
}
```

Sample Response

```
{
  "response": "**  WideIP - testGw.com  **\ngtm wideip /Common/testGw.com {\n    pools {\n        /Common/asdf {\n
order 0\n            ratio 4\n        }\n    }\n}\n\n**  GTM Pool - asdf  **\ngtm pool /Common/asdf {\n
disabled\n    members {\n        /Common/server123:gvs1 {\n            order 0\n        }\n    }\n}\n\n**  GTM
Server - server123  **\ngtm server /Common/server123 {\n    addresses {\n        146.125.206.132 {\n
device-name server123\n        }\n    }\n    datacenter /Common/cbe\n    monitor /Common/bigip \n    virtual-servers
{\n        gvs1 {\n            destination 146.125.206.133:80\n        }\n        gvs2 {\n            destination
146.125.206.133:443\n        }\n        gvss {\n            destination 146.125.206.133:443\n        }\n        vs1
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

{\n destination 2.2.3.4:50\n }\n }\n}\n\n** GTM Monitor - bigip **\ngtm monitor bigip bigip

{\n aggregate-dynamic-ratios \"none\"\n defaults-from none\n destination *:*\n ignore-down-response

disabled\n interval 30\n timeout 90\n}\n\n",

"message": "Configuration found for the given object.",

"appStatusCode": null,

"tags": null,

"headers": null

}

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Action - Enable an Object
- Action - Disable an Object.

Action - Enable an Object

This API triggers the action to enable one or more ADC objects to which the action applies. The enable action is supported on the following object types:

- **F5** - WideIP, WideIP pool, WideIP pool member, VIP, VIP pool member, LTM node, GTM server, virtual address
- **A10** - FQDN, ServiceIP, SLB_VSV, SLB Server, SLB VS
-  **AVI** - Virtual service
-  **Cisco** - GSLB Answer, SLB virtual server, SLB real server
- **Citrix** - GSLB virtual server, GSLB services, SLB virtual servers, SLB service group, SLB service
-  **Akamai** - Datacenter

Before you begin

Before attempting to initiate enable action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable/Disable/ForceDown**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-enable`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-enable?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Content-Type:	application/json

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	Mandatory param 'objectIds' is missing or empty	Check if a non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Trigger enables action on objects with resource id gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-enable?gwsource=external`

Sample Request

```
{
  "payload": {
    "objectIds": [
      "gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common",
      "gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common"
    ]
  }
}
```

Sample Response

```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 2 objects : gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,
gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **HostName**: A human-readable label assigned to a device (host) on a network

  The hostname will be included in the endpoint URL for an on-prem deployment.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Disable an Object.
- Action - Forcedown Object

Action - Disable an Object

This API initiates action Disable on one or more ADC objects for which the action applies to. Disable action disables the object and terminate all active connections. Disable action is supported on the following object types:

- **F5** - WideIP, WideIP pool, WideIP pool member, VIP, VIP pool member, LTM node, GTM server, virtual address
- **A10** - Fqdn, ServiceIP, SLB_VSV, SLB Server, SLB VS
-  **AVI** - Virtual Service
-  **Cisco** - GSLB Answer, SLB virtual server, SLB real server
- **Citrix** - GSLB virtual server, GSLB services, SLB virtual servers, SLB service group, SLB service
-  **Akamai** - Datacenter

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to initiate enable action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable/Disable/ForceDown**

This can be checked under **Accounts** > **Role** > **Authorized Functions.**

Request Structure

Endpoint:	/adc-device-object-action-disable
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-disable?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response	Success message or failure description in case of error.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission

Sample Request/Response

Use Case

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Trigger disable action on objects with resource id gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-disable?gwsource=external
```

Sample Request


```
{
  "payload": {
    "objectIds": [
      "gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common",
      "gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common"
    ]
  }
}
```


Sample Response


```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 2 objects : gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,
gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Disable an Object.
- Action - Forcedown Object

Action - Forcedown Object

This API initiates the forcedown action on one or more ADC objects to which the action applies. Forcedown action is supported on the following object types:

- **F5** - LTM pool member and LTM Server

Before you begin

Before attempting to initiate forcedown action with this API, users must ensure the ACF permission is available

- **ADC Accounts** > **Role** > **Authorized Functions** > **ControlCenter** > **Actions** > **Enable/Disable/ ForceDown**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-forcedown`
Type:	`PUT`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-forcedown?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Mandatory	Description	Field Type	Constraints
objectIds *String[]*	Yes	Resource ID of the objects. Use the search API to search/get the resource ID.		-

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Sample Request/Response

Use Case

Trigger disable action on objects with resource id lpm:@5f369cba0ca9846e86a13e82:@appviewx_pool:@Common::1.1.1.10:12:@Common

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-forcedown?gwsource=external

Sample Request

```
{
  "payload": {
    "objectIds": [
      "lpm:@5f369cba0ca9846e86a13e82:@appviewx_pool:@Common::1.1.1.10:12:@Common"
    ]
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }
}
```

Sample Response

```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 objects :
lpm:@5f369cba0ca9846e86a13e82:@appviewx_pool:@Common::1.1.1.10:12:@Common",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Action - Forcedown Active Object.

Action - Forcedown Active Object

This API initiates the forcedown active action on one or more ADC objects to which the action applies. Forcedown active action is supported on the following object types:

- **F5** - LTM pool member
- **A10** - Virtual Service, Service Group Member
- **Citrix** - SLB Virtual Server

Before you begin

Before attempting to initiate enable action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **ForceDown active**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-forcedown_active`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-forcedown_active?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Sample Request/Response

Use Case

Trigger disable action on objects with resource id lpm:@5f369cba0ca9846e86a13e82:@appviewx_pool:@Common::1.1.1.10:12:@Common

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-forcedown_active?gwsource=external

Sample Request


```
{
  "payload": {
    "objectIds": [
      "lpm:@5f369cba0ca9846e86a13e82:@appviewx_pool:@Common::1.1.1.10:12:@Common"
    ]
  }
}
```


Sample Response


```
{
  "response": "Action triggered successfully.",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 "message": "Action triggered for the following 1 objects :
lpm:@5f369cba0ca9846e86a13e82:@appviewx_pool:@Common::1.1.1.10:12:@Common",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Forcedown Active Object.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Action - Graceful Disable Object

This API initiates the graceful disable action on one or more ADC objects to which the action applies. Graceful disable is used to disable an object only when all the currently active client connections are closed by either the server or the client. Graceful disable action is supported on the following object types:

- **AVI** - AVI Server

Before you begin

Before attempting to initiate graceful disable action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable/Disable/ForceDown**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-gracefuldisable`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-gracefuldisable?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Trigger graceful disable action on objects with resource id - aviserver:@5f369d260ca9846e86a13f30:@Pool_Other:@0.0.0.0:80:@aviserver:@Other:@0.0.0.0:@Default-Cloud

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-gracefuldisable?gwsource=external

Sample Request


```
{
  "payload": {
    "objectIds": [  "aviserver:@5f369d260ca9846e86a13f30:@Pool_Other:@0.0.0.0:80:@aviserver:@Other:@0.0.0.0:@Default-Cloud"
    ]
  }
}
```


Sample Response


```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 objects :
aviserver:@5f369d260ca9846e86a13f30:@Pool_Other:@0.0.0.0:80:@aviserver:@Other:@0.0.0.0:@Default-Cloud",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **HostName**: A human-readable label assigned to a device (host) on a network

  The hostname will be included in the endpoint URL for an on-prem deployment.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Enable Persistence for Object

Action - Enable Persistence for Object

This API initiates the action to enable persistence on one or more ADC objects to which the action applies. Enabling persistence allows for the tracking and storing of session data, ensuring that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions. The enable persistence action is supported on the following object types:

- **F5** – WideIp

Before you begin

Before attempting to initiate enable persistence action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable / Disable persistence**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Request Structure

Endpoint:	`/adc-device-object-action-enablepersistence`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-enablepersistence?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource	(Mandatory) Specifies the source from which the request is triggered. The options are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Query	• web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Sample Request/Response

Use Case

Trigger enable persistence action on objects with resource id gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@

Request URL

```
https://<Tenant_name/Host_name>:<portno>/avxapi/adc-device-object-action-enablepersistence?gwsource=external
```

Sample Request

```
{
  "payload": {
    "objectIds": [
      "gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common",
      "gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common"
    ]
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }
}
```

Sample Response

```
{
 "response": "Action triggered successfully.",
 "message": "Action triggered for the following 2 objects :
gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Action - Enable Persistence for Object

Action - Disable Persistence for Object

This API initiates action Disable persistence on one or more ADC objects for which the action applies to. Disable persistence is used to turn off the tracking and storing of session data, which is used to ensure that client requests are directed to the same pool member throughout the life of a session or during subsequent sessions. Enable persistence action is supported on the following object types:

- **F5** – WideIp

Before you begin

Before attempting to initiate enable persistence action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable / Disable persistence**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-disablepersistence`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-disablepersistence?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Sample Request/Response

Use Case

Trigger disable persistence action on objects with resource id gw:@<objectid>:@192.168.112.63:@Common,gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common

Note:
<objectid> represents your object ID.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-disablepersistence?gwsource=external

Sample Request

```
{
"payload": {
"objectIds": [
"gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common",
"gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common"
]
}
}
```

Sample Response

```
{
"response": "Action triggered successfully.",
"message": "Action triggered for the following 2 objects :
gw:@5f369cba0ca9846e86a13e82:@192.168.112.63:@Common,gw:@5f369cba0ca9846e86a13e82:@192.168.55.248:@Common",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
"appStatusCode": null,
"tags": null,
"headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Clear Persistence Record

Action - Clear Persistence Record

This API initiates the action to blear persistence records on one or more ADC objects for which the action applies. Clear persistence action is supported on the following object types:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **F5** – Virtual Server, LTM pool

Before you begin

Before attempting to initiate Clear persistence records action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Clear persistence connections**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-clearpersistencerecords
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-clearpersistencerecords?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if a non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Trigger clear persistence record action on objects with resource id - vs:@5f369cc20ca9846e86a13e99:@ACLTest:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-clearpersistencerecords`

Sample Request

```
{
  "payload": {
    "objectIds": [
      "vs:@5f369cc20ca9846e86a13e99:@ACLTest:@Common"
    ]
  }
}
```

Sample Response

```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 objects : vs:@5f369cc20ca9846e86a13e99:@ACLTest:@Common",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The hostname will be included in the endpoint URL for an on-prem deployment.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action – InService (Activate) Object

Action – InService (Activate) Object

This API initiates the action to InService (Activate) on one or more ADC objects for which the action applies. InService(Activate) action is supported on the following object type:

- **Cisco** – GSLB Answer, SLB virtual server, SLB real server

Before you begin

Before attempting to initiate InService(Activate) action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable / Disable persistence**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-activate`
Type:	`PUT`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-activate?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if a non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Sample Request/Response

Use Case

Trigger activate action on objects with resource id - slbrs:@5f369cb20ca9846e86a13e72:@ipv6_server-farm_82::DemoRealServers:@Admin

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-activate

Sample Request

```
{
  "payload": {
    "objectIds": [
      "slbrs:@5f369cb20ca9846e86a13e72:@ipv6_server-farm_82::DemoRealServers:@Admin"
    ]
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }
}
```

Sample Response

```
{
 "response": "Action triggered successfully.",
 "message": "Action triggered for the following 1 objects : slbrs:@5f369cb20ca9846e86a13e72:@ipv6_server-
farm_82::DemoRealServers:@Admin",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Action - OutOfService (Suspend) Object

Action - OutOfService (Suspend) Object

This API initiates the action to out of service (suspend) on one or more ADC objects for which the action applies. The suspend action is supported on the following object type:

- **Cisco** – GSLB Answer, SLB virtual server, and SLB real server

Before you begin

Before attempting to initiate suspend action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable / Disable persistence**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-suspend
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-suspend?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String **Constraints**: The session ID is used when username and password are not provided.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> objects : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check if non-null/non-empty value is given in the payload field - “objectIds”.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_CC_0501	No Objects found for performing the given action.	Check and ensure if the requested objectId is available and not deleted.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Sample Request/Response

Use Case

Trigger suspend action on objects with resource id - slbrs:@5f369cb20ca9846e86a13e72:@ipv6_server-farm_82::DemoRealServers:@Admin

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-suspend

Sample Request

```
{
  "payload": {
    "objectIds": [
      "slbrs:@5f369cb20ca9846e86a13e72:@ipv6_server-farm_82::DemoRealServers:@Admin"
    ]
  }
}
```

Sample Response

```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 objects : slbrs:@5f369cb20ca9846e86a13e72:@ipv6_server-
farm_82::DemoRealServers:@Admin",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Set CNAME

Action - Set Service Down Value

This API initiates the action to set service down on one or more ADC objects for which the action applies. The action service down is supported on the following object type:

- **F5** - LTM pool

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to initiate set service down action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Service down**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-servicedown
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-servicedown?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description.
objects *Key value pair[]*	(Mandatory) Object information containing resource ID and parent resource ID.
serviceDown *Strin*	(Mandatory) Service down value to set. Possible service down values - none,reset,reselect,drop. **Constraints:** It accepts only alphanumeric characters, '-' , '.', '_', '(', ')'.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_CC_05014	Invalid input. Please provide 'serviceDown' to perform action!	Check and ensure if a non-null/non-empty value is given in payload field - ‘serviceDown’.
400 Bad Request	ADC_OBJ_ACTION_001	Object id not found.	Check and ensure if a non-null/non-empty value is given in payload field - ‘objects._id’.
400 Bad Request	ADC_OBJ_ACTION_002	parent object id not found.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects.parent_id’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s).	-

Sample Request/Response

Use Case

Set service down value to reselect on object with resource id - lp:@5f4393cd80d9c82e890183ce:@tets.avx.com_pool:@Common with parent object - vs:@5f4393cd80d9c82e890183ce:@DC_vs:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-servicedown`

Sample Request

```
{
  "payload": {
    "objects": [
      {
        "_id": "lp:@5f4393cd80d9c82e890183ce:@tets.avx.com_pool:@Common",
        "parent_id": "vs:@5f4393cd80d9c82e890183ce:@DC_vs:@Common"
      }
    ],
    "serviceDown": "reselect"
  }
}
```

Sample Response

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
{
 "response": "Action triggered successfully.",
 "message": "Action triggered for the following 1 object(s) : lp:@5f4393cd80d9c82e890183ce:@tets.avx.com_pool:@Common",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Objects in Constant State

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Action - Enable Proxy for Object

This API initiates action Enable proxy on one or more ADC objects for which the action applies to. Enable Proxy feature routes traffic through an intermediary server, hiding the origin server for enhanced security and performance. It helps protect the server from direct exposure, optimize traffic, and provide additional features like caching and filtering. The enable proxy action is supported on the following object types:

- **Cloudflare** – LoadBalancer

Before you begin

Before attempting to initiate set service down action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Enable / Disable Proxy**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-enableproxy`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-enableproxy?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_001	Object id not found.	Check and ensure if a non-null/non-empty value is given in payload field - ‘objects._id’.
400 Bad Request	ADC_OBJ_ACTION_002	parent object id not found.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects.parent_id’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s).	-

Sample Request/Response

Use Case

Trigger enable proxy action on objects with resource idcflb:@6706c22622d1165f74002dde:@test.avxddidemo.net:@cb1a7eabcda964347d225fae0a3448e9

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-enableproxy`

Sample Request

```
{
   "payload": {
      "objectIds": [
         "cflb:@6706c22622d1165f74002dde:@test.avxddidemo.net:@cb1a7eabcda964347d225fae0a3448e9"
      ]
   }
}
```

Sample Response

```
{
   "response": "Action triggered successfully.",
   "message": "Action triggered for the following 1 objects :
 cflb:@6706c22622d1165f74002dde:@test.avxddidemo.net:@cb1a7eabcda964347d225fae0a3448e9",
   "appStatusCode": null,
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Objects in Constant State

Action - Disable Proxy for Object

This API initiates action Disable proxy on one or more ADC objects for which the action applies to. Disable Proxy feature stops routing traffic through the intermediary server, allowing direct access to the origin server. The disable proxy action is supported on the following object types:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

• **Cloudflare** – LoadBalancer

Before you begin

Before attempting to initiate set service down action with this API, users must ensure the ACF permission is available:

• **ADC** > **ControlCenter** > **Actions** > **Enable / Disable Proxy**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-disableproxy
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-disableproxy?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response	Success message or failure description in case of error.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_OBJ_ACTION_001	Object id not found.	Check and ensure if a non-null/non-empty value is given in payload field - ‘objects._id’.
400 Bad Request	ADC_OBJ_ACTION_002	parent object id not found.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects.parent_id’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects’.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s).	-

Sample Request/Response

Use Case

Trigger disable proxy action on objects with resource idcflb:@6706c22622d1165f74002dde:@test.avxddidemo.net:@cb1a7eabcda964347d225fae0a3448e9

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-disableproxy

Sample Request

```
{
   "payload": {
      "objectIds": [
         "cflb:@6706c22622d1165f74002dde:@test.avxddidemo.net:@cb1a7eabcda964347d225fae0a3448e9"
      ]
   }
}
```

Sample Response

```
{
   "response": "Action triggered successfully.",
   "message": "Action triggered for the following 1 objects :
 cflb:@6706c22622d1165f74002dde:@test.avxddidemo.net:@cb1a7eabcda964347d225fae0a3448e9",
   "appStatusCode": null,
   "tags": null,
   "headers": null
}
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Objects in Constant State

Get Objects in Constant Status

This API fetches the ADC objects that remain in a constant status for Last N days. In the response lastFlipDate fields gives the details of last state flip occurred date from which object state of the object was not flipped.

Supported Vendor: F5

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to get objects in the constant state with this API, users must ensure any of the ACF permission from is available:

- **ADC** > **Control Center**
- **ADC** > **Dashboard** > **Application Widget**
- **ADC** > **Orphan objects**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-constant-object-status-fetch
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-constant-object-status-fetch?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectType *String[]*	(Mandatory) Object Types. Should be left empty if object type is not needed to include in query. **Possible values**: wideIp, wideIp Pool, wideIp PoolMember, childWideIp, server,VirtualServer, Pool, ltmPoolMember, VirtualAddress, and LTM Server.
deviceName *String[]*	(Optional) Name of the device(s). **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.
status *String*	(Mandatory) Status of the object. **Possible Values**: "OFFLINE", "UNKNOWN", and "AVAILABLE" Intially, the status is in "OFFLINE. It changes to after sometimes.
lastNDays	(Mandatory) Last 'N' no. of days for which an object remained in constant state.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
Integer	
vendor *String*	(Optional) Vendor of the devce. For example, F5, NginxPlus, etc.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the response for objects in requested status for lastNDays requested. See State Status Drift response for more details.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	-	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
400 Bad Request	ADC_DASH_1604	Invalid Input: Mandatory fields 'status' is missing	Check and ensure if a non-null/non-empty value is given in payload field - ‘status’.
400 Bad Request	ADC_DASH_1605	Invalid Input: Mandatory field 'lastNDays' is missing	Check and ensure if a non-null and a valid integer value is given in the field - ‘lastNDays’.
400 Bad Request	ADC_DASH_0002	Invalid Input	Check the type of the request fields - objectType,deviceName and lastNDays. Expected field types: • lastNDays - Integer • deviceName - String[] • objectType - String[]
404 Not Found	ADC_DASH_1608	No objects found for the given input.	• Check if the requested objects have Read or ReadWrite permission. • Refine the search query.

Sample Request/Response

Use Case

Get LTM pool member objects which remained in status - “OFFLINE” for the last 2 days.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-constant-object-status-fetch`

Sample Request

```
{
  "payload": {
    "objectType": [
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "ltmPoolMember"
   ],
   "status": "OFFLINE",
   "lastNDays": 2
  }
}
```


Sample Response


```
{
 "response": [
  {
   "objectId": "lpm:@5f4e3423fcbb9809756c6b82:@testweb:@Common::test:443:@Common",
   "objectName": "test:443",
   "objectType": "LTM Pool Member",
   "deviceId": "5f4e3423fcbb9809756c6b82",
   "deviceName": "192.168.41.233",
   "vendor": "F5",
   "state": null,
   "status": "OFFLINE",
   "displayName": "test:443/one.appviewx.com:443/testweb/192.168.41.233/F5",
   "effectiveDays": 0,
   "lastFlipDate": "2020-09-01"
  }
 ],
 "message": null,
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Object Status Trend

Get Objects in Constant State

This API fetches the ADC objects that remain in a constant state for Last N days. In the response lastFlipDate fields gives the details of the last state flip occurred date from which object state of the object was not flipped.

Supported Vendor: F5

Before you begin

Before attempting to get objects in the constant state with this API, users must ensure any of the ACF permission from is available:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **ADC** > **Control Center**, **ADC** > **Dashboard** > **Application Widget**
- **ADC** > **Orphan objects**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-constant-object-state-fetch
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-constant-object-state-fetch?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type	(Mandatory) Specifies the format of the data in the payload. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectType *String[]*	(Mandatory) Object Types. Should be left empty if object type is not needed to include in query. **Possible values**: wideIp, wideIp Pool, wideIp PoolMember, childWideIp, server,VirtualServer, Pool, ltmPoolMember, VirtualAddress, and LTM Server.
deviceName *String[]*	(Mandatory) Name of the device(s). **Constraints**: It can contain only alphanumeric characters, ‘-’ , ’_’ , ‘.’ , ‘*’ , ‘|’, ‘!’.
state *String*	(Mandatory) State of the object. **Possible Values**: ‘ENABLED’, ‘DISABLED’, and ‘DISABLED_BY_PARENT’.
lastNDays *Integer*	(Mandatory) Last ‘N’ no. of days for which an object remained in constant state.
vendor *String*	(Optional) Vendor of the devce. For example, F5, NginxPlus, etc.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the response for objects in requested state for lastNDays requested. See State Status Drift response for more details.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	-	-
400 Bad Request	ADC_DASH_1607	Invalid Input: Mandatory fields 'state' is missing.	Check and ensure if a non-null/non-empty value is given in payload field - ‘state’.
400 Bad Request	ADC_DASH_1605	Invalid Input: Mandatory field 'lastNDays' is missing.	Check and ensure if a non-null and a valid integer value is given in field - ‘lastNDays’
400 Bad Request	ADC_DASH_0002	Invalid Input	Check the type of the request fields - objectType,deviceName and lastNDays. Expected field types:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
			• lastNDays - Integer • deviceName - String[] • objectType - String[]
404 Not Found	ADC_DASH_1608	No objects found for the given input.	• Check if the requested objects have Read or ReadWrite permission. • Refine the search query.

Sample Request/Response

Use Case

Get LTM pool member objects which remained in state - “ENABLED” for the last 2 days.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-constant-object-state-fetch`

Sample Request

```
{
 "payload": {
  "objectType": [
   "ltmPoolMember"
  ],
  "state": "ENABLED",
  "lastNDays": 2
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }


}
```

Sample Response


```
{
 "response": [
  {
   "objectId": "lpm:@5f4e2ffafcbb9809756c6b4d:@perf56testpool77:@Common::15.1.77.3:80:@Common",
   "objectName": "15.1.77.3:80",
   "objectType": "LTM Pool Member",
   "deviceId": "5f4e2ffafcbb9809756c6b4d",
   "deviceName": "192.168.40.214",
   "vendor": "F5",
   "state": "ENABLED",
   "status": null,
   "displayName": "15.1.77.3:80/15.1.77.3:80/perf56testpool77/192.168.40.214/F5",
   "effectiveDays": 0,
   "lastFlipDate": "2020-09-01"
  }
 ],
 "message": null,
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The hostname will be included in the endpoint URL for an on-prem deployment.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Objects in Constant Status

Get Object Status Trend

This API fetches the status flip history of the given object. It will respond with the last N status flip history of an object.By default N is configured as 100 based on the usage and N can be configured.

Supported Vendor: F5

Before you begin

Before attempting to get an object status trend with this API, users must ensure any of the ACF permission from is available:

- **ADC** > **Control Center**,
- **ADC** > **Dashboard** > **Application Widget**
- **ADC** > **Orphan objects**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-object-status-trend-fetch`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-object-status-trend-fetch?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Type: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectId *String*	(Mandatory) Resource ID of the object. Use the search API to search/get the resource ID.
lastNDays *Integer*	(Mandatory) Last 'N' no. of days for which an object remained in constant state.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the response for requested objects last 'N' flip history. See State Status Drift response for more details.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	-	-
400 Bad Request	ADC_DASH_1606	Invalid Input: Mandatory fields 'objectId' is missing	Check and ensure if a non-null/non-empty value is given in the query param field - ‘objectId’.
400 Bad Request	ADC_DASH_0035	Invalid Input	Check the type of the query param field - lastNCounts. Expected type is Integer.
404 Not Found	ADC_DASH_1608	No objects found for the given input	• Check if the requested objects have Read or ReadWrite permission. • Refine the search query.

Sample Request/Response

Use Case

Get last 5 status flip trend for objectId - lpm:@5f4e3091a7452e427bcede52:@newltmpool:@Common::12.9.1.0:21:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-object-status-trend-fetch`

Sample Request

```
NA
```

Sample Response

```
{
  "response": [
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  {
    "objectId": "lpm:@5f4e3091a7452e427bcede52:@newltmpool:@Common::12.9.1.0:21:@Common",
    "objectName": "12.9.1.0:21",
    "objectType": "LTM Pool Member",
    "deviceId": "5f4e3091a7452e427bcede52",
    "deviceName": "gs-f5-pe109.lab.appviewx.net",
    "vendor": "F5",
    "state": null,
    "status": "OFFLINE",
    "displayName": "12.9.1.0:21/12.9.1.0:21/newltmpool/gs-f5-pe109.lab.appviewx.net/F5",
    "effectiveDays": 0,
    "lastFlipDate": "2020-09-01"
  }
 ],
 "message": null,
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Set Loadbalancing Method

State Status Drift Response

Name	Description
objectName *String*	Name of the object.
objectId *String*	Resource Id of the object in AppViewX.
objectType *String*	Object type as mentioned in AppViewX.
deviceId *String*	Resource ID of the device to which the object belongs.
deviceName *String*	Name of the device to which the object belongs.
vendor *String*	Vendor of the device to which the object belongs.
displayName *String*	DisplayName given for the object in AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
status *String*	Status of the object.
state *String*	State of the object.
effectiveDays *Integer*	No of days the object remained in a state/status. Effective days will be 0 for the latest record (i.e) the active record.
lastFlipDate *String*	Date on which the state/status of the object flipped to its current state/status. Date Format:YYYY-MM-DD.

Possible LoadBalancing Modes

Vendor list with LoadBalancing Modes

Vendor	ObjectType	Possible LB Modes
F5	LTM Pool	'round-robin','ratio-member','least-connections-member','observed-member','predictive-member','ratio-node','least-connections-node','fastest-node','observed-node','predictive-node','dynamic-ratio-node','fastest-app-response','least-sessions','dynamic-ratio-member','weighted-least-connections-member','weighted-least-connections-node','ratio-session','ratio-least-connections-member','ratio-least-connections-node'
Citrix	GSLB Virtual Server	'Round Robin','Least Connection','Least Response Time','Source IP Hash','Least Bandwidth','Least Packets','Custom Load','Dynamic Method(RTT)','Static Proximity'
Citrix	SLB Virtual Server	'Round Robin','Least Connection','Least Response Time','URL HASH','Domain Hash','Destination IP Hash','Source IP Hash','Least Bandwidth','Least Packets','Source IP Destination IP Hash','LRTM','Source IP Source Port Hash','Call ID Hash','Custom Load','Token','Least Request'

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendor list with LoadBalancing Modes (continued)

Vendor	ObjectType	Possible LB Modes
A10	FQDN	'forward response','forward query','forward both','drop','reject'
A10	SLB Service Group	'round-robin','least-connection','service-least-connection','weighted-rr','weighted-least-connection','service-weighted-least-connection','fastest-response','least-request','round-robin-strict','stateless-src-ip-hash', 'stateless-dst-ip-hash','stateless-src-dst-ip-hash','stateless-per-pkt-round-robin','stateless-src-ip-only-hash','src-ip-only-hash','src-ip-hash','dst-ip-only-hash','dst-ip-hash'
AVI	SLB Pool	'LB_ALGORITHM_CONSISTENT_HASH','LB_ALGORITHM_FASTEST_RESPO
Infoblox	LBDN	'GLOBAL_AVAILABILITY','RATIO','ROUND_ROBIN','TOPOLOGY'
Infoblox	Pool	'ALL_AVAILABLE','GLOBAL_AVAILABILITY','RATIO','ROUND_ROBIN','TOPOL
Cloudflare	LoadBalancer	off,geo,random,dynamic_latency,proximity,least_outstanding_requests,least_co
Cloudflare	Pool	random, hash, least_outstanding_requests (edited)

Input Keys For Device

Name	Description
name *String*	(Optional) Name of the device. **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.
ip *String*	(Optional) IP of the device.
vendor *String*	(Optional) Vendor of the device. For example: F5, AVI,etc.
version	(Optional) Version of the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	**Constraints**: It only accepts alphanumeric characters, ‘-’, ‘:’, ‘.’, ‘_’ and space.
dataCenter *String*	(Optional) Data center of the device.
status *String*	(Optional) Status of the device in inventory. **Possible values**: Managed","UnManaged", "Unresolved", "Failed","Inprogress", and "Queued".
haActive *String*	(Optional) HA state of the device. Active/StandBy.
syncFailoverGroupName *String*	(Optional) Failover group name of the device.

Supported Object Type/Object Code

Vendor codes and Description

Vendor	Object Code	Object Type	Description
F5	vs	Virtual Server	LTM Virtual server
F5	lpm	Pool Member	LTM pool member
F5	lm	LTM Monitor	LTM Monitor
F5	lpr	Profile	Profile
F5	lr	LTM iRule	LTM i-rule
F5	ltmpersistence	LTM Persistence	LTM Persistence
F5	gw	WideIP	GTM Wide IP
F5	gpm	Pool Member	GTM pool member
F5	gp	GTM Pool	GTM pool

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendor codes and Description (continued)

Vendor	Object Code	Object Type	Description
F5	gm	GTM Monitor	GTM monitor
F5	gs	GTM Server	GTM Server
F5	gr	GTM iRule	GTM i-rule
Citrix	citrixvs	SLB Virtual Server	SLB virtual server
Citrix	citrixsv	SLB Service	SLB Service
Citrix	citrixslbsg	SLB Service Group	Service Group
Citrix	citrixslbsgm	SLB Service Group Member	Service Group Member
Citrix	citrixcsvs	CS Virtual Server	CS Virtual Server
Citrix	citrixcspolicylabel	CS Policy Label	CS Policy Label
Citrix	citrixcspolicy	CS Policy	CS Policy
Citrix	citrixcsaction	CS Action	CS Action
Citrix	citrixmonitor	Monitor	Monitor
Citrix	citrixserver	Server	Server
Citrix	citrixgslbvs	GSLB Virtual Server	GSLB Virtual Server
Citrix	citrixgslbsv	GSLB Service	GSLB Service
Citrix	citrixgslbsg	GSLB Service Group	GSLB Service Group
Citrix	citrixgslbsgm	GSLB Service Group Member	GSLB Service Group Member
Citrix	citrixgslbsite	GSLB Site	GSLB Site
Akamai	akagp	akagp	akagp
Akamai	akadc	akadc	akadc
Akamai	akags	akags	akags
A10	a10vs	Virtual Server	Virtual Server
A10	a10sg	Service Group	Service Group

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendor codes and Description (continued)

Vendor	Object Code	Object Type	Description
A10	a10sr	Service Group Member	Service Group Member
A10	a10hm	SLB Health Monitor	SLB Health Monitor
A10	a10fx	aFlex	aFlex
A10	a10cl	Class	Class
A10	a10tm	Template	Template
A10	a10node	Server	Server
A10	a10snatpool	SnatPool	A10 SNAT pool
A10	a10zn	FQDN	FQDN
A10	a10sip	Service IP	Service IP
A10	a10hm	GSLB Health Monitor	GSLB Health Monitor
A10	a10p	Policy	Policy
AVI	avivsv	Virtual Service	Virtual Service
AVI	avipool	Pool	Pool
AVI	aviserver	Server	Server
AVI	aviAppProfile	Application Profile	Application Profile
AVI	aviNwProfile	Network Profile	Network Profile
AVI	aviAnlProfile	Analytical Profile	Analytical Profile
AVI	aviSslProfile	Ssl Profile	Ssl Profile
AVI	aviAppPerProfile	Application Persistence Profile	Application Persistence Profile
AVI	aviPkiProfile	Pki Profile	Pki Profile
AVI	aviMonitor	Monitor	Monitor
AVI	aviIpGroup	Ip Group	Ip Group
AVI	aviNsRule	Network Security Rule	Network Security Rule
AVI	aviHttpSecRule	Http Security Rule	Http Security Rule
AVI	aviHttpReqRule	Http Request Rule	Http Request Rule

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Vendor codes and Description (continued)

Vendor	Object Code	Object Type	Description
AVI	aviHttpResRule	Http Response Rule	Http Response Rule
AVI	aviDataScript	Data Script	Data Script
Cisco	slbvs	Virtual Server	Virtual Server
Cisco	slbsf	Server Farm	Server Farm
Cisco	Slbrs	Real Server	Real Server
NginxPlus	nserver	LB Server	LB Server
NginxPlus	nupstream	Upstream	Upstream
NginxPlus	nbackendserver	Upstream Server	Upstream Server
Cloudflare	cflb	LoadBalancer	LoadBalancer
Cloudflare	cfpool	Pool	Pool
Cloudflare	cfep	Endpoint	Endpoint (Origin Server)

Action - Set Loadbalancing Method

This API is used to initiate action Set load balancing method on one or more ADC objects for which the action applies to. Suspend action is supported on the following object types:

- **F5** – LTM pools
- **Citrix** – GSLB Virtual Server and SLB Virtual Server
- **A10** - FQDN and SLB Service Group
- **AVI** - SLB Pool
- Infoblox - LBDN and Pool
- **Cloudflare** - LoadBalancer and Pool

Before you begin

Before attempting to initiate suspend action with this API, users must ensure the ACF permission is available:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **ADC** > **ControlCenter** > **Actions** > **LB mode**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-lbmethod
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-lbmethod?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type	(Mandatory) Specifies the format of the data in the payload. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objects *Key value pair[]*	(Mandatory) Object information containing resource id and parent resource id. For object details, click here.
lbMode *String*	(Mandatory) Load balancing mode to set. Possibles values for lbMode can be found here.

Object Details

Name	Description
_id *String*	Resource ID of the object. **Note:** Use the search API to search/get the resource ID.
parent_id	Resource ID of the parent object.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Object Details (continued)

Name	Description
String	**Note:** Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_CC_05014	Invalid input. Please provide 'lbMode' to perform action!	Check and ensure if a non-null/non-empty value is given in the payload field - ‘lbMode’.
400 Bad Request	ADC_OBJ_ACTION_001	Object id not found.	Check and ensure if a non-null/non-empty value is given in paylo.ad field - ‘objects._id’.
400 Bad Request	ADC_OBJ_ACTION_002	parent object id not found.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects.parent_id’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check and ensure if a non-null/non-empty value is given in the payload field - ‘objects’.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and chan.ge the request objectId.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s)	-

Sample Request/Response

Use Case

Trigger set load balancing mode action on object with resource id - lp:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-lbmethod`

Sample Request

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
{
 "payload": {
  "objects": [
   {
    "_id": "lp:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common",
    "parent_id": "vs:@5f369cc10ca9846e86a13e95:@CVS:@Common"
   }
  ],
  "lbMode": "round-robin"
 }
}
```


Sample Response


```
{
 "response": "Action triggered successfully.",
 "message": "Action triggered for the following 1 object(s) : lp:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Set Connection Limit

Action - Set Connection Limit

This API initiates action Set connection limit on one or more ADC objects for which the action applies to. Set connection limit action is supported on the following object type:

- **F5** - LTM servers

Before you begin

Before attempting to initiate a set connection limit action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-connectionlimit`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-connectionlimit?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Mandatory	Description	Field Type	Constraints
objectIds *String[]*	Yes	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.		-
connectionLimit *String*	Yes	(Mandatory) Connection limit to set. **Constraints**: Allows only numerical values. For example, 4.	String	

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_CC_05011	Invalid input. Please provide 'connectionLimit' to perform action!	Check and ensure if a non-null/non-empty value is given in the payload field - ‘‘connectionLimit’’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action	• Check and ensure if a non-null/non-empty value is given in the payload field - ‘objectIds’. • Check if the requested object is available and not deleted.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s)	-

Sample Request/Response

Use Case

Trigger set connection limit action on object with resource id - ln:@5f369cba0ca9846e86a13e82:@1.1.1.1:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-connectionlimit`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request


```
{
  "payload": {
    "objectIds": [
      "ln:@5f369cba0ca9846e86a13e82:@1.1.1.1:@Common"
    ],
    "connectionLimit": "4"
  }
}
```


Sample Response


```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 object(s) : ln:@5f369cba0ca9846e86a13e82:@1.1.1.1:@Common",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - Set Ratio

Action - Set Ratio

This API initiates action Set ratio on one or more ADC objects for which the action applies to. Set ratio action is supported on the following object types:

- **F5** – GTM pool, GTM pool member, LTM pool member
- **AVI** - GSLB Pool Member, Server
- **Akamai** - Datacenter

Before you begin

Before attempting to initiate set ratio action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Ratio**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-object-action-ratio`
Type:	`PUT`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-ratio?gwsource=external` To understand the elements of the sample URL, click here.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
objects *Key value pair[]*	(Mandatory) Object information containing resource ID and parent resource ID. For object details, click here.
ratio	(Mandatory) Ratio to set. **Constraints**: Allows only numerical values.

Object Details

Name	Description
_id *String*	Resource ID of the objects. **Note:** Use the search API to search/get the resource ID.
parent_id *String*	Resource ID of the parent objects. **Note:** Use the search API to search/get the resource ID.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response	Success message or failure description in case of error.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_CC_05012	Invalid input. Please provide 'ratio' to perform action!	Check and ensure if a non-null/non-empty value is given in payload field - ‘ratio’.
400 Bad Request	ADC_OBJ_ACTION_001	Object id not found.	Check and en..sure if a non-null/non-empty value is given in payload field - ‘objects._id’
400 Bad Request	ADC_OBJ_ACTION_002	parent object id not found.	Check and ensure if a non-null/non-empty value is given in payload field - ‘objects.parent_id’
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	Check and ensure if a non-null/non-empty value is given in payload field - ‘objects’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s)	-

Sample Request/Response

Use Case

Set a ratio of 1 on object with resource id - lpm:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common::ms_serv:443:@Common with parent object - lp:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-ratio`

Sample Request

```
{
  "payload": {
    "objects": [
      {
        "_id": "lpm:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common::ms_serv:443:@Common",
        "parent_id": "lp:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common"
      }
    ],
    "ratio": "1"
  }
}
```

Sample Response

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
{
 "response": "Action triggered successfully.",
 "message": "Action triggered for the following 1 object(s) : lpm:@5f369cc10ca9846e86a13e95:@ms_pool_1:@Common::ms_serv:443:@Common",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Action - ARP Enable/Disable

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Action - ARP Enable/Disable

This API initiates action ARP enable/disable on one or more ADC objects for which the action applies to. Set ARP action is supported on the following object types:

- **F5** - Virtual address
- **A10** - Virtual server

Before you begin

Before attempting to initiate a set ARP action with this API, users must ensure the ACF permission is available:

- **ADC** > **Control Center** > **Actions** > **ARP**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-arp
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-arp?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Note: Use the search API to search/get the resource ID.
arp *String*	Action. Possible values are ‘ENABLE and DISABLE’ .

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_CC_05010	Invalid input. Please provide 'arp' to perform action!	Check and ensure if a non-null/non-empty value is given in the payload field - ‘arp’’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	• Check and ensure if a non-null/non-empty value is given in the payload field - '`objectIds`'. • Check if the requested object is available and not deleted.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request `objectId`.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read-Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s)	-

Sample Request/Response

Use Case

Trigger ARP disable action on objects with resource id - va:@5f4393db80d9c82e890183ef:@5.36.8.91:@Common

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-arp`

Sample Request

```
{
  "payload": {
    "objectIds": [
      "va:@5f4393db80d9c82e890183ef:@5.36.8.91:@Common"
    ],
    "arp": "DISABLED"
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }
}
```

Sample Response

```
{
 "response": "Action triggered successfully.",
 "message": "Action triggered for the following 1 object(s) : va:@5f4393db80d9c82e890183ef:@5.36.8.91:@Common",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Action - Set Weight

Action - Set Weight

This API initiates action Set weight on one or more ADC objects for which the action applies to. Set weight action is supported on the following object types:

- Citrix - SLB service group member, SLB service, Service
- A10 - Service group member, Server
- Cloudflare - Endpoint

Before you begin

Before attempting to initiate a set weight action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **Ratio**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-weight
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-weight?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.
weight *String*	(Mandatory) Weight to set. **Constraints**: Allows only numerical values.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message along with the objectIds or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation.
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_CC_0506	Invalid input. Please provide 'weight' to perform action!	Check and ensure if a non-null/non-empty value is given in the payload field - ‘weight’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	• Check and ensure if a non-null/non-empty value is given in the payload field - ‘objectIds’. • Check if the requested object is available and not deleted.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId.
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s)	-

Sample Request/Response

Use Case

Set weight of 2 on objects with resource id - citrixsv:@5f4393c580d9c82e890183c1:@mytest::test:80

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-weight`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request


```
{
  "payload": {
    "objectIds": [
      "citrixsv:@5f4393c580d9c82e890183c1:@mytest::test:80"
    ],
    "weight": "2"
  }
}
```


Sample Response


```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 object(s) : citrixsv:@5f4393c580d9c82e890183c1:@mytest::test:80",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete a Widget

Action - Set CNAME

This API initiates the action to set CNAME on one or more ADC objects for which the action applies. The set CNAME action is supported on the following object type:

- **Akamai** - Property

Before you begin

Before attempting to initiate a set CNAME action with this API, users must ensure the ACF permission is available:

- **ADC** > **ControlCenter** > **Actions** > **CNAME**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-object-action-cname
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-cname?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
objectIds *String[]*	(Mandatory) Resource ID of the objects. Use the search API to search/get the resource ID.
cname *String*	(Mandatory) CNAME to set.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Action triggered for the following <n> object(s) : <resourceIds>	-
400 Bad Request	ADC_CC_05013	Invalid input. Please provide 'cname' to perform action!	Check and ensure. if a non-null/non-empty value is given in payload field - ‘weight’.
400 Bad Request	ADC_OBJ_ACTION_003	No object(s) not found to perform action.	• Check and ensure if a non-null/non-empty value is given in payload field - ‘objectIds’ • Check if the requested object is available and not deleted.
400 Bad Request	ADC_OBJ_ACTION_004	Action is invalid for the given objects.	Check for supported objects for the action and change the request objectId
403 Forbidden	ADC_DASH_0029	No given object(s) have Write permission.	Check and ensure if the requested object has Read Write permission.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s)	-

Sample Request/Response

Use Case

Set cname value ‘www.test.com’ on objects with resource id - akagp:@5c59407cdd342b6c9431f609:@support.avxlab.net.akadns.net

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-object-action-cname`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request


```
{
  "payload": {
    "objectIds": [
      "akagp:@5c59407cdd342b6c9431f609:@support.avxlab.net.akadns.net"
    ],
    "cname": "www.test.com"
  }
}
```


Sample Response


```
{
  "response": "Action triggered successfully.",
  "message": "Action triggered for the following 1 object(s) : akagp:@5c59407cdd342b6c9431f609:@support.avxlab.net.akadns.net",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete a Widget

Get Transaction Status

This API fetches the status of the actions for objects involved in a transaction. This support is available for all object-level actions like Enable, Disable, etc. Transaction ID received from the response of these APIs can be sent as input to get the status.

Request Structure

Endpoint:	`/adc-get-transaction-status`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-get-transaction-status?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String **Constraints**: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Header	**Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
transactionId *String*	(Mandatory) Transaction ID of action triggered. It can be found in the response of the action APIs under the tags field in the name ‘transactionId’.
Status *String*	(Optional) Transaction Status to be searched.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
Objects *Array*	(Optional) Transaction status of objects to be searched.
deviceNames *Array*	(Optional) Transaction status of the device to be searched.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the requested pool objects and pool members.
message *String*	Failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Code

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Ok	-	Refer to sample response	

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	AVX_GW_009	Resource not found, reason - Invalid apiid	
400	avx-common-028	Invalid/ Incorrect payload	

Sample Request/Response

Use Case

To fetch the action status of the object/device for the transaction.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-get-transaction-status`

Sample Request

```
{
  "payload": {
    "transactionId": "external-17631fj9h1",
    "objects": [{
        "objectType": "ltmPoolMember",
        "deviceName": "gs-f5-pe55.apvxlab.com",
        "objectName": "livepool",
       "partition": "Common",
       "recordType": "Common",
       "vendor": "F5"
    }, {
       "objectType": "ltmPoolMember",
       "deviceName": "192.168.31.188",
      "objectName": "livepool",
       "partition": "Common",
      "recordType": "Common",
       "vendor": "F5"
      }]
  }
}
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Response


```
{
   "response": {
      "transactionId": "external-17631fj9h1",
       "action": "Enable",
       "user": "admin",
       "comments": "action",
       "deviceObjects": [{
            "code": "go",
            "partition": "Common",
            "displayName": "livpool/a/gs-f5-pe55.apvxlab.com/F5",
            "vendor": "F5",
             "logMessage": "Action Disable executed successfully on object livpool/a/gs-f5-pe55.apvxlab.com/F5 by the user admin Reason: perform disable. The action is executed on a Standby device as an Active peer is not identified in AppViewX",
             "objectName": "livpool",
             "objectId": "gp:@5fa5305656aaf54f67a001f7:@livpool:@Common:@a",
  "status": "success"
          }, {
            "code": "go",
            "partition": "Common",
            "displayName": "livepool/a/192.168.31.188/F5",
             "vendor": "F5",
             "logMessage": "Action Disable executed successfully on object livepool/a/192.168.31.188/F5 by the user admin Reason : perform disable.The action is executed on a Standby device as an Active peer is not identified in AppViewX",
             "objectName": "livepool",
              "objectId":
"gp:@5fa5305656aaf54f67a002ca:@livpool:@Common:@a",
              "status": "success"
             } ],
   "timeStamp": 1606722095762
   }
 }
```


Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Search - LTM and GTM Pool

Search - LTM and GTM Pool

This API searches for LTM and GTM pools and associated pool members. LTM and GTM Pool Search action is supported on the following object types:

- F5 - LTM pool
- F5 - GTM pool

Before You Begin

Before attempting to search for the pool(s) and associated pool members with this API, users must ensure any of the ACF permission is available:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **ADC** > **Control Center**
- **ADC** > **Dashboard** > **Application Widget**
- **ADC** > **Orphan objects**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-controlcenter-pool-object-search
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-controlcenter-pool-object-search ?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String
	Constraints: The password, along with the username, is used when the session ID is not provided.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
deviceName *String*	(Mandatory) Device name of the pool to be searched for.
objectName *String*	(Mandatory) Object name to be searched.
objectType *String*	(Mandatory) GTM pool and LTM pool have to be searched. **Constraints:** Value of param should be TM pool or WideIP Pool or both.
partition *String*	(Optional) Partition in which the pool lies.
recordType *String*	(Optional) Pool record type.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
orphan *Boolean*	(Optional) Flag to get orphan objects.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the requested pool objects and pool members.
message *String*	Failure description in case of error
appStatusCode *String*	Application-specific status code for the response. It will be non-null for failure response.
tags *String*	More info in case of a failed response.

Response

Name	Description
result *Object*	Contains result object

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Object

Name	Description
Object	List of objects from the database that match the input query.
Object[]	

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Ok	-	Refer to sample response	
404 Not Found	AVX_GW_009	Resource not found, reason - Invalid apiid	
400	avx-common-028	Invalid/ Incorrect payload	

Sample Request/Response

Use Case

Search pool object with mandatory and optional parameters.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-controlcenter-pool-object-search`

Sample Request

```
{
  "payload": {
    "objects": [
      {
        "objectType": "ltm Pool",
        "deviceName": "192.168.42.150",
        "objectName": "pool_demo36_80",
        "orphan" : true
      }
    ]
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  }
}
```

Sample Response

```
{
  "response": {
    "result": {
      "objects": [
        {
          "id": "lp:@60518caecc48a15dda15a673:@pool_demo36_80:@Common",
          "name": "pool_demo36_80",
          "deviceName": "192.168.42.150",
          "objectType": "ltm Pool",
          "code": "lp",
          "displayName": "pool_demo36_80/192.168.42.150/F5",
          "orphan": true,
          "recordType": null,
          "status": "ENABLED",
          "statusCode": "UNKNOWN ENABLED",
          "childs": [
            {
"id":"lpm:@6051da15a73:@pool_demo36_80:@Common::192.168.40.31:80:@Common",
              "name": "192.168.40.31:80",
              "code": "lpm",
               "displayName": "192.168.40.31:80/192.168.40.31:80/pool_demo36_80/192.168.42.150/F5",
              "partition": "Common",
              "ip": "192.168.40.31",
              "port": "80",
              "status": "DISABLED",
              "statusCode": "UNKNOWN DISABLED",
              "memberName": "/Common/192.168.40.31",
              "parentName": "pool_demo36_80",
              "parentPartion": "Common"
            }
    ]
      }
  },
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "message": null,
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Object Configuration

Application View Widget - Action by Group

This API performs action based on widget/group names.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to delete a widget with this API, user must ensure any one of the below ACF permissions is available,

- **ADC** > **Dashboard** > **Application widget** > **Add / Delete**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-dashboard-applicationview-action-by-group
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-dashboard-applicationview-action-by-group?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
dashboardName *Payload*	(Mandatory) Name of the dashboard. **Constraints**: Allowed special characters : '_', '-', ':', '.', '|' (Should not be empty and not start with special characters). **Type**: String
widgetName *Payload*	(Mandatory) Name of the Application View widget. **Constraints**: Allowed special characters : ‘-’ ,’_’ , ‘.’ , ‘:’ , ’|’ **Type**: String
groupNames *Payload*	Name of the group. **Constraints**: • If the **groupNames** parameter is not provided, the action (e.g., enable, disable) will be applied to the entire widget, including all groups associated with it. • If **groupNames** are specified, the action will be applied only to the objects within those groups. • To reference a subgroup, use the format `<parent-group>/<child-group>`. For example: `VIP/PoolMember`.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Type: Object
action *Payload*	(Mandatory) Name of the group. **Possible Values:** Enable, Disable **Type**: String
comment *Payload*	(Optional) Additional notes or remarks related to the query. Used for tracking or clarification purposes. **Type**: String

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Widget deleted successfully.	-
400 Bad Request	ADC_DASH_8791	Dashboard name is mandatory.	Check and ensure if a valid value is given in the query param field - dashboardName.
400 Bad Request	ADC_DASH_8792	Widget name is mandatory.	Check and ensure if a valid value is given in query param field - widgetName
404 Not Found	ADC_DASH_0045	Dashboard not available for given name.	Requested dashboard not available in database. Check if the dashboard is deleted.
404 Not Found	ADC_DASH_3010	Widget not found.	Check if the widget is deleted or if the widget is available in a different dashboard.
403 Forbidden	ADC_DASH_0048	Permission not available for dashboard.	Check if the user has Read Write access for the requested dashboard.

Sample Request/Response

Use Case

Group the application view widget named **CutoverDashboard-Small** under the group **VIP/Pools** within the dashboard.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-dashboard-applicationview-action-by-group?gwsource=external

Sample Request

```
{
"dashboardName": "CutoverDashboard-Small",
"widgetType": "ApplicationView",
"widgetName": "CutoverWidget-Small-1",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
"groupNames": [ "VIP/Pools" ],
"action": "Enable",
"comment": " Enabling widget"
}
```

Sample Response

```
{
   "response": "success",
   "message": null,
   "appStatusCode": null,
   "tags": {
      "transactionId": "test-123"
   },
   "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete Device

Update Traffic Grid Widget

This API updates a traffic grid widget within the dashboard.

Before you begin

Before attempting to delete a widget with this API, user must ensure any one of the below ACF permissions is available,

- **ADC+** > **Traffic Management** > **Dashboard** > **Traffic grid widget** > **Monitor traffic percentage**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-dashboard-widget-trafficgrid-action-update
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-dashboard-widget-trafficgrid-action-update?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type	(Mandatory) Specifies the format of the data in the payload.
Header	**Type**: String
	Constraints: The value of this param must be ‘application/json’.
gwsource	(Mandatory) Specifies the source from which the request is triggered. The options are:
Query	• web • external
	Type: String
dashboardName	(Mandatory) Name of the dashboard.
Payload	**Constraints**: Allowed special characters : '_', '-', ':', '.', '|' (Should not be empty and not start with special characters).
	Type: String
widgetName	(Mandatory) Name of the traffic grid widget.
Payload	**Constraints**: Allowed special characters : ‘-’ ,’_’ , ‘.’ , ‘:’ , ’|’
	Type: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
actionDetails *Payload*	Refer to Action Details.
comment *Payload*	(Optional) Additional notes or remarks related to the query. Used for tracking or clarification purposes. **Type**: String

actionDetails

Name	Description
application *Query*	(Mandatory) Name of the application. **Type**: String
dcRatioMap *Query*	(Mandatory) Defines the distribution ratio for data center resources or workloads across multiple data centers. **Type**: Object

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Widget updated successfully
message *String*	Success message or failure description in case of error.
appStatusCode	Application specific status code for the response. Will be non-null for failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
tags	More info in case of failure response.
String	

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Widget deleted successfully.	-
400 Bad Request	ADC_DASH_8791	Dashboard name is mandatory.	Check and ensure if a valid value is given in the query param field - dashboardName.
400 Bad Request	ADC_DASH_8792	Widget name is mandatory.	Check and ensure if a valid value is given in query param field - widgetName
404 Not Found	ADC_DASH_0045	Dashboard not available for given name.	Requested dashboard not available in database. Check if the dashboard is deleted.
404 Not Found	ADC_DASH_3010	Widget not found.	Check if the widget is deleted or if the widget is available in a different dashboard.
403 Forbidden	ADC_DASH_0048	Permission not available for dashboard.	Check if the user has Read Write access for the requested dashboard.

Sample Request/Response

Use Case

Update the application "TestApp" across three data centers (avx, cbe, che) in the dashboard "CutoverDashboard".

Request URL

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-dashboard-widget-trafficgrid-action-update?gwsource=external

Sample Request

```
{
    "dashboardName": "CutoverDashboard",
    "widgetName": "CutoverWideget-2",
    "comment": "update",
    "actionDetails": [
        {
            "application": "TestApp",
            "dcRatioMap": {
                "avx": 30,
                "cbe": 30,
                "che": 40
            }
        }
    ]
}
```

Sample Response

```
{
    "response": "success",
    "message": null,
    "appStatusCode": null,
    "tags": null,
    "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **HostName**: A human-readable label assigned to a device (host) on a network

  The hostname will be included in the endpoint URL for an on-prem deployment.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete Device

Get Class Attribute Associations for a Class Management Widget

This API fetches associated attributes for a class management widget within the dashboard.

Before you begin

Before attempting to delete a widget with this API, user must ensure any one of the below ACF permissions is available,

- **General** > **Dashboard** > **Create / Delete**
- **ADC** > **Dashboard** > **Class management widget** > **Add / Delete**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-dashboard-widget-classmanagement-get-stringvalue-association-rest`
Type:	`PUT`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-dashboard-widget-classmanagement-get-stringvalue-association-rest?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
dashboardName *Payload*	(Mandatory) Name of the dashboard. **Constraints**: Allowed special characters : '_', '-', ':', '.', '|' (Should not be empty and not start with special characters). **Type**: String
widgetName *Payload*	(Mandatory) Name of the class management widget. **Constraints**: Allowed special characters : '-' ,'_' , '.' , ':' , '|' **Type**: String
className *Payload*	(Mandatory) Name of a class, including device path and object class. **Type**: String
deviceName *Payload*	(Mandatory) Name of the device associated with the class. **Type**: String
widgetId *Payload*	(Mandatory) Unique identifier of the widget. **Type**: String
actionName *Payload*	(Mandatory) Action to be performed. **Possible Value:** Edit **Type**: String
groupName *Payload*	(Mandatory) Name of the widget group in the dashboard. **Type**: String
actionMode *Payload*	(Mandatory) Modifying class action allows you to view and change the details of a class. **Type**: String
objectId	(Mandatory) Unique object ID associated with the class instance.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Payload	**Type**: String

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Widget deleted successfully.	-
400 Bad Request	ADC_DASH_8791	Dashboard name is mandatory.	Check and ensure if a valid value is given in the query param field - dashboardName.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
400 Bad Request	ADC_DASH_8792	Widget name is mandatory.	Check and ensure if a valid value is given in query param field - widgetName
404 Not Found	ADC_DASH_0045	Dashboard not available for given name.	Requested dashboard not available in database. Check if the dashboard is deleted.
404 Not Found	ADC_DASH_3010	Widget not found.	Check if the widget is deleted or if the widget is available in a different dashboard.
403 Forbidden	ADC_DASH_0048	Permission not available for dashboard.	Check if the user has Read Write access for the requested dashboard.

Sample Request/Response

Use Case

fetch and modify the class attributes associated with a specific F5 device (pe-qa-f5-n8.lab.appviewx.net) under a defined class (aol/pe-qa-f5-n8.lab.appviewx.net/F5.

).

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/`adc-dashboard-widget-classmanagement-get-stringvalue-association-rest?gwsource=external

Sample Request

```
className=aol/pe-qa-f5-n8.lab.appviewx.net/F5
deviceName=pe-qa-f5-n8.lab.appviewx.net
widgetId=6854f7f6df937f36ecc8a3d7
actionName=Edit
groupName=Test
actionMode=ModifyClass
objectId=lc:@68553483df937f36ecc8ac47:@aol:@Common
```

Sample Response

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
{
    "response": {
        "selectedClassAttributes": [
            {
                "Mask": "255.255.255.252",
                "Type": "Network",
                "String": "192.168.70.0",
                "Value": "",
                "keyData": "192.168.70.0/30:="
            }
        ],
        "partition": "Common",
        "ruleList": "",
        "availableDevices": [
            {
                "name": "192.168.55.132",
                "id": "6835ae789d3ae02294dcfb2b"
            }
        ],
        "iTotalDisplayRecords": 17,
        "availableClassAttributes": [
            {
                "keyData": "10.10.10.0/24:=",
                "String": "10.10.10.0/24",
                "Value": "",
                "type": "Address"
            }
        ],
        "type": "Address",
        "selectedDevices": [
            {
                "name": "pe-qa-f5-n8.lab.appviewx.net",
                "id": "68553483df937f36ecc8ac47"
            }
        ]
    },
    "message": null,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete Device

Update Class Attribute for a Class Management Widget

This API modifies the class attributes for a class maangement widget within dashboard.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to delete a widget with this API, user must ensure any one of the below ACF permissions is available,

- **ADC** > **Dashboard** > **Class management widget** > **Add / Delete**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-dashboard-widget-class-management-modifyAction
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-dashboard-widget-class-management-modifyAction?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
widgetId *Payload*	(Mandatory) Unique identifier of the widget. **Type**: String
groupName *Payload*	(Mandatory) Name of the logical group associated with the widget. **Type**: String
classType *Payload*	(Mandatory) Type of class being updated. **Type**: String
className *Payload*	(Mandatory) Name of the class. **Type**: String
objectId *Payload*	(Mandatory) Unique identifier of the object. **Type**: String
logTimeStamp *Payload*	(Mandatory) Log Timestamp. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
deviceName *Payload*	(Mandatory) device where the class is applied. **Type**: String
actionMode *Payload*	(Mandatory) Modifying class action allows you to view and change the details of a class. **Type**: String
actionName *Payload*	(Mandatory) Action to be performed. **Possible Value:** Edit **Type**: String
defaultClassName *Payload*	(Mandatory) Name of the default class. **Type**: String
selDeviceName *Payload*	(Mandatory) Name of the selected device. **Type**: String
availClassAttrList *Payload*	(Mandatory) List of available class attributes for validation or selection. **Type**: String
path *Payload*	(Optional) Path details. **Type**: String
deviceList *Payload*	(Mandatory) Devices involved in the modification operation. **Type**: Object
classAttrList *Payload*	(Mandatory) List of address objects (with mask and type) to be added or updated. **Type**: Object
rollBack	(Mandatory) Determine if rollback should be allowed on failure.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Payload	**Type**: Boolean

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Widget deleted successfully.	-
400 Bad Request	ADC_DASH_8791	Dashboard name is mandatory.	Check and ensure if a valid value is given in the query param field - dashboardName.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
400 Bad Request	ADC_DASH_8792	Widget name is mandatory.	Check and ensure if a valid value is given in query param field - widgetName
404 Not Found	ADC_DASH_0045	Dashboard not available for given name.	Requested dashboard not available in database. Check if the dashboard is deleted.
404 Not Found	ADC_DASH_3010	Widget not found.	Check if the widget is deleted or if the widget is available in a different dashboard.
403 Forbidden	ADC_DASH_0048	Permission not available for dashboard.	Check if the user has Read Write access for the requested dashboard.

Sample Request/Response

Use Case

Update the class management widget "CutoverWideget-2" in the "CutoverDashboard" dashboard to perform a ModifyClass action on the "F5" class for the device "pe-qa-f5-n8.lab.appviewx.net".

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/`adc-dashboard-widget-class-management-modifyAction?gwsource=external

Sample Request

```
{
  "widgetId": "6854f7f6df937f36ecc8a3d7",
  "groupName": "Test",
  "classType": "Address",
  "className": "aol/pe-qa-f5-n8.lab.appviewx.net/F5",
  "objectId": "lc:@68553483df937f36ecc8ac47:@aol:@Common",
  "logTimeStamp": 1750856426000,
  "deviceName": "pe-qa-f5-n8.lab.appviewx.net",
  "actionMode": "ModifyClass",
  "actionName": "Edit",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
"defaultClassName": "aol/pe-qa-f5-n8.lab.appviewx.net/F5",
"selDeviceName": "pe-qa-f5-n8.lab.appviewx.net",
"availClassAttrList": [
    {
        "keyData": "192.168.70.0/30:=",
        "String": "192.168.70.0/30",
        "Value": "",
        "type": "Address"
    },
    {
        "keyData": "192.168.70.0/32:=",
        "String": "192.168.70.0/32",
        "Value": "",
        "type": "Address"
    },
    {
        "keyData": "192.168.1.0/30:=",
        "String": "192.168.1.0/30",
        "Value": "",
        "type": "Address"
    },
    {
        "keyData": "10.10.10.0/24:=",
        "String": "10.10.10.0/24",
        "Value": "",
        "type": "Address"
    },
    {
        "keyData": "10.10.10.0/30:=",
        "String": "10.10.10.0/30",
        "Value": "",
        "type": "Address"
    }
],
"path": "",
"deviceList": [
    {
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
        "name": "pe-qa-f5-n8.lab.appviewx.net",
        "_id": "68553483df937f36ecc8ac47"
    }
],
"classAttrList": [
    {
        "Mask": "255.255.255.252",
        "Type": "Network",
        "String": "1.2.3.4",
        "Value": "",
        "keyData": "1.2.3.4/30:="
    },
    {
        "Mask": "255.255.224.0",
        "Type": "Network",
        "String": "64.12.96.0",
        "Value": "",
        "keyData": "64.12.96.0/19:="
    },
    {
        "Mask": "255.255.255.252",
        "Type": "Network",
        "String": "192.168.70.0",
        "Value": "",
        "keyData": "192.168.70.0/30:="
    },
    {
        "Mask": "255.255.240.0",
        "Type": "Network",
        "String": "195.93.16.0",
        "Value": "",
        "keyData": "195.93.16.0/20:="
    },
    {
        "Mask": "255.255.252.0",
        "Type": "Network",
        "String": "195.93.48.0",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "Value": "",
    "keyData": "195.93.48.0/22:="
  },
  {
    "Mask": "255.255.224.0",
    "Type": "Network",
    "String": "195.93.64.0",
    "Value": "",
    "keyData": "195.93.64.0/19:="
  },
  {
    "Mask": "255.255.224.0",
    "Type": "Network",
    "String": "195.93.96.0",
    "Value": "",
    "keyData": "195.93.96.0/19:="
  },
  {
    "Mask": "255.255.252.0",
    "Type": "Network",
    "String": "198.81.0.0",
    "Value": "",
    "keyData": "198.81.0.0/22:="
  },
  {
    "Mask": "255.255.254.0",
    "Type": "Network",
    "String": "198.81.8.0",
    "Value": "",
    "keyData": "198.81.8.0/23:="
  },
  {
    "Mask": "255.255.240.0",
    "Type": "Network",
    "String": "198.81.16.0",
    "Value": "",
    "keyData": "198.81.16.0/20:="
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
},
{
    "Mask": "255.255.255.128",
    "Type": "Network",
    "String": "202.67.65.128",
    "Value": "",
    "keyData": "202.67.65.128/25:="
},
{
    "Mask": "255.255.240.0",
    "Type": "Network",
    "String": "205.188.112.0",
    "Value": "",
    "keyData": "205.188.112.0/20:="
},
{
    "Mask": "255.255.255.252",
    "Type": "Network",
    "String": "205.188.146.144",
    "Value": "",
    "keyData": "205.188.146.144/30:="
},
{
    "Mask": "255.255.240.0",
    "Type": "Network",
    "String": "205.188.192.0",
    "Value": "",
    "keyData": "205.188.192.0/20:="
},
{
    "Mask": "255.255.254.0",
    "Type": "Network",
    "String": "205.188.208.0",
    "Value": "",
    "keyData": "205.188.208.0/23:="
},
{
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
            "Mask": "255.255.248.0",
            "Type": "Network",
            "String": "207.200.112.0",
            "Value": "",
            "keyData": "207.200.112.0/21:="
        },
        {
            "keyData": "10.10.10.0/30:=",
            "String": "10.10.10.0/30",
            "Value": "",
            "type": "Address"
        }
    ],
    "rollBack": false
}
```


Sample Response


```
{
    "response": {
        "dbUpdateResult": true,
        "failedDevices": [],
        "DBLoggingResult": "Logging Succeded",
        "succeedDevices": [
            "pe-qa-f5-n8.lab.appviewx.net"
        ]
    },
    "message": null,
    "appStatusCode": null,
    "tags": null,
    "headers": null
}
```


Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete Device

Device

The Devices can be added into AppViewX and monitored. By using the APIs, the devices can be added, searched, and/or managed within AppViewX.

Generate Device Backup

This API triggers backup for a device. The generated backups are stored in the device inventory under the mentioned backup group name. If a backup group doesn’t exist, a new backup group will be created.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Backup now**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-backup
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-backup?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
deviceName *String*	(Mandatory) Name of the device added in inventory. **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.
backupName *String*	(Mandatory) The name of the backup group. A backup group will be created in this name if it isn't available. If the backup group is already existing in the system then backup will be triggered only if one of the following conditions are met. • The existing backup group is of device type. In this case, the device will be added to the backup group if it isn't available. • The existing backup group is of device group type and any of the configured device groups contain the requested device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	Constraints: It can contain only Alphanumeric characters , ‘-’, ‘_’, ‘.’ ,’*’ , ‘:’ , ‘|’ , and space.
passphrase *String*	(Optional) PassPhrase key to encrypt the backup. **Constraints**: Passphrase is supported only for vendor F5.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device backup triggered successfully.	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_INV_1117	Invalid backup name. Backup name accepts only '_','-','.' as special characters and should not start or end with special characters.	Check if the value given under field ‘backupName’ is valid. Backup name accepts only '_','-','.' as special characters and should not start or end with special characters.
400 Bad Request	ADC_INV_1116	Backup name cannot be null or empty.	Check if a non-null/non-empty value is given under the payload field - ‘backupName’.
400 Bad Request	ADC_INV_1092	Device Name cannot be null or empty.	Check if a non-null/non-empty value is given under payload field - ‘deviceName’.
400 Bad Request	ERR_OBJECT_023	Respective vendor's device is not supported. Action cannot be performed.	Request only devices of vendor - A10,AVI,Citrix,F5.
400 Bad Request	ERR_OBJECT_023	Respective vendor's device is not supported. Action cannot be performed.	Request only devices of vendor - AVI,F5.
400 Bad Request	ADC_INV_1115	Mismatch in backup type, cannot add device to existing group as it already has a devicegroup configured.	• Change the request backupName for which device is configured and not device group. • Give a new backupName to be created and the requested device will be added in that backup group.
403 Forbidden	ADC_INV_1098	Cannot access resource, permission not available.	Check if Read/Read Write permission is available for the requested device.
403 Forbidden	ADC_INV_1077	Action cannot be performed on unmanaged device object(s).	Check if the requested device is in unmanaged state. If required, the device should be moved to managed state for performing backup.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is available in inventory.

Sample Request/Response

Use Case

Get the list of backups available for the device testdevice.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-backup`

Sample Request

```
{
  "payload": {
    "backupName": "test",
    "deviceName": "testDevice"
  }
}
```

Sample Response

```
{
  "response": "Backup triggered for device testDevice",
  "message": "Device backup triggered successfully.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Backup Details for a Device

Get Backup Details for a Device

This API retrieves a list of device backups available in Device Inventory against any particular ADC device.

Before you begin

Before attempting to get back up details with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Object compare**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Request Structure

Endpoint:	/adc-device-archives
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-archives?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource	(Mandatory) Specifies the source from which the request is triggered. The options are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Query	• web • external **Type**: String
deviceName *Query*	(Mandatory) Name of the device. Constraints: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the response params for the get backup details request. For more details, see response.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Response

Name	Description
<deviceName> *Key value pair*	Device name given in request will be the key and the value is an object that contains the backup details. For more details, see Backup File Detail.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Backup File Details

Name	Description
filename *String*	Device name given in request will be the key and the value is an object that contains the backup details.
metadata *Key value pair*	Metadata field denoting the backup details. For more details, see Backup File Metadata.
chunkSize *Integer*	Size of the backup.
length *Integer*	Length of the backup chunk stored in the database.
md5 *String*	MD5 sum of the device backup.
_id *String*	Resource id denoting the backup file metadata stored in the database.

Backup File Metadata

Name	Description
timeStamp *String*	Time Stamp denoting the time on which the backup file is added to the database.
groupName *String[]*	Group settings name in which the backup resides.
isEncrypted *Boolean*	Boolean value denoting whether the device backup is encrypted or not.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
deviceId *String*	Resource ID of the device for which backup is retrieved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	Archive details retrieved successfully.	-
400 Bad Request	ADC_INV_1084	Device name is mandatory.	Check and ensure if a non-null/non-empty value is given in query param field - ‘deviceName’.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Get the list of archives for the device with name testDevice.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-archives`

Sample Request

```
NA
```

Sample Response


```
{
  "response": {
    "testDevice": [
      {
        "name": null,
        "chunkSize": 261120,
        "length": 1192434,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
      "md5": "f2663fdf65d3db4644bd074fd6306de1",
      "filename": "testDevice_1597903384000.tar.gz",
      "metadata": {
        "timeStamp": "1597903439000",
        "mode": "daily",
        "groupName": [
          "test"
        ],
        "isEncrypted": "false",
        "deviceId": "5f36bed67620c40e97e30348"
      },
      "_id": "5f3e124f25ac8a2582b08e57"
    }
  ]
 },
 "message": "Archive details retrieved successfully.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Device Backup

This API triggers the backup for the device.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Backup now**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/adc-device-backup`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-backup?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
backupName *String*	(Mandatory) Name of the backup.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
deviceName *String*	(Optional) Name of the device.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Accepted	-	null	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Sample Request/Response

Use Case

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Trigger the device backup with the device name and backup name.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-backup
```

Sample Request

```
{
   "backupName": "backup_node22_summary",
   "deviceName": "pe-adc-qa-f5-n4.lab.appviewx.net"
 }
```

Sample Response

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Reference

External Device Backup Download

This API triggers the backup download for external device.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Backup now**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-inventory-external-device-backup-download
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-external-device-backup-download?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
externalBackupArchiveId *String*	(Mandatory) External device backup archive ID. **Constraints**: It can contain only alphanumeric.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Accepted	-	null	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Sample Request/Response

Use Case

Fetch the device backup details.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-external-device-backup-download`

Sample Request

```
{
}
```

Sample Response

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
{
    "response": {
        "name": null,
        "chunkSize": 0,
        "length": 21641098,
        "filename": "192.168.150.82_1757663737000.tar.gz",
        "vendor": "F5",
        "metadata": {
            "timeStamp": "1757663752036",
            "downloadStatus": "IN_PROGRESS",
            "deviceId": "683583fa9d3ae02294dcd655"
        },
        "_id": "68c3d208c4eb7e56da0b9a08"
    },
    "message": null,
    "appStatusCode": null,
    "tags": null,
    "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Device Backup

Backup Archive

This API fetches the list of backup archive details for the device.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Backup now**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-archives
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-archives?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
backupName *String*	(Mandatory) Name of the backup.
archiveObject *Object*	(Mandatory) Details of the archive object. See archiveObject.

archiveObject

restoreArchiveFileId *String*	(Mandatory) ID of the restore archive file.
rollBackArchiveFileId *String*	(Mandatory) ID of the roll back archive file.
reasonForRestore *String*	(Mandatory) Restore reason. It is a free-form field.
isDeviceRestore *Boolean*	(Mandatory) Device restore status. Possible values: true, false
restoreArchiveFileName *String*	(Mandatory) Name of the restore archive file.
rollBackArchiveFileName *String*	(Mandatory) Name of the roll back archive file.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Accepted	-	null	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Sample Request/Response

Use Case

Fetch the backup archive by providing the device name and archive details..

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-archives`

Sample Request

```
{
    "deviceName": "192.168.147.21",
    "archiveObject": {
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "restoreArchiveFileId": "68c152afc4c3ed31ccacbb68",
    "rollBackArchiveFileId": "68c152afc4c3ed31ccacbb68",
    "reasonForRestore": "",
    "isDeviceRestore": "true",
    "restoreArchiveFileName": "192.168.147.21_1757500079000.tar.gz",
    "rollBackArchiveFileName": "192.168.147.21_1757500074000.tar.gz"
}
```


Sample Response

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

What's Next

- Search for Device Groups

Device Restore

This API triggers device restore.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Restore**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-inventory-device-restore
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-device-restore?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
backupName *String*	(Mandatory) Name of the backup.
archiveObject *Object*	(Mandatory) Details of the archive object. See archiveObject.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

archiveObject

restoreArchiveFileId *String*	(Mandatory) ID of the restore archive file.
rollBackArchiveFileId *String*	(Mandatory) ID of the roll back archive file.
reasonForRestore *String*	(Mandatory) Restore reason. It is a free-form field.
isDeviceRestore *Boolean*	(Mandatory) Device restore status. Possible values: true, false
restoreArchiveFileName *String*	(Mandatory) Name of the restore archive file.
rollBackArchiveFileName *String*	(Mandatory) Name of the roll back archive file.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode	Application-specific status code for the response. Will be non-null for failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Accepted	-	null	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Sample Request/Response

Use Case

Trigger device restore.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-device-restore`

Sample Request

```
{
   "payload": {
      "deviceName": "192.168.147.21",
      "archiveObject": {
         "restoreArchiveFileId": "68c152afc4c3ed31ccacbb68",
         "rollBackArchiveFileId": "68c152afc4c3ed31ccacbb68",
         "reasonForRestore": "",
         "isDeviceRestore": "true",
         "restoreArchiveFileName": "192.168.147.21_1757500079000.tar.gz",
         "rollBackArchiveFileName": "192.168.147.21_1757500074000.tar.gz"
      },
      "source": "backupRestore_deviceRestore"
   }
}
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Response

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Device Restore Status

Device Restore Status

This API fetches the status of the device restore.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Restore**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-inventory-get-device-restore-status
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-get-device-restore-status?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
backupName *String*	(Mandatory) Name of the backup.
deviceName *String*	(Mandatory) Name of the device.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Accepted	-	null	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Sample Request/Response

Use Case

Get the status of device restore.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-get-device-restore-status`

Sample Request

```
{
}
```

Sample Response

```
{
   "response": [
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
{
    "id": "68791c3fb83ca219e42fd69c",
    "logs": [
        {
            "name": "Create",
            "time": 1752767551339,
            "messages": [
                {
                    "time": 1752767551000,
                    "message": "Device Restore is triggered for the device : 192.168.150.82"
                }
            ],
            "workOrderId": 1,
            "mongoId": "68791c3fb83ca219e42fd69c"
        },
        {
            "name": "BackUp",
            "time": 1752767551554,
            "messages": [
                {
                    "time": 1752767551000,
                    "message": "Storing Running Configuration"
                }
            ],
            "mongoId": "68791c3fb83ca219e42fd69c"
        },
        {
            "name": "Upload",
            "time": 1752767631419,
            "messages": [
                {
                    "time": 1752767738000,
                    "message": "Configuration for restore has been uploaded by the user"
                }
            ],
            "mongoId": "68791c3fb83ca219e42fd69c"
        },
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
            {
                "name": "Install",
                "time": 1752767738619,
                "messages": [
                    {
                        "time": 1752767738000,
                        "message": "Device Restore for the workorder DR-1 performed on the device 192.168.150.82 is Successful."
                    }
                ],
                "mongoId": "68791c3fb83ca219e42fd69c"
            }
        ],
        "deviceName": "192.168.150.82",
        "status": "Request Closed",
        "deviceId": "683583fa9d3ae02294dcd655"
    }
  ],
  "message": null,
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Device Restore

Search for Device Groups

This API fetches the device groups details based on the device name.

Before you begin

Before attempting to initiate device backup with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Backup & Restore** > **Backup now**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-inventory-get-all-group-settings/backUpRestore
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-get-all-group-settings/backUpRestore?gwsource=external To understand the elements of the sample URL, click here.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
SearchParam *String*	(Optional) Filter for narrowing down results **Constraints**: It can contain only alphanumeric characters, ‘-’ , ’_’ , ‘.’ , ‘*’ , ‘|’, ‘!’.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application-specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 Accepted	-	null	-
400 Bad Request	ADC_INV_2006	Invalid input.	Check if a non-null/non-empty value is given under field ‘payload’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Get the device groups details based on the device name.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-inventory-get-all-group-settings/backUpRestore

Sample Request


```
{

}
```


Sample Response


```
{
    "response": {
        "backupnowPermission": true,
        "allBackUpSettings": [
            {
                "periodicSettings": {
                },
                "doClusterBackup": false,
                "category": "ADC",
                "description": "",
                "searchStrings": [
                ],
                "listOfSelectedDevices": [

                    {
                        "vendor": "F5",
                        "name": "BVT_ControlCenter_ADC_F5V13",
                        "_id": "6887a3f8c6f406522b24212f",
                        "category": "ADC"
                    }
                ],
                "isGenerateNow": true,
                "type": "DEVICE",
                "isEmailConfigured": false,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
            "listOfStatusForEmail": [
                "Success",
                "Failure"
            ],
            "emailSummary": false,
            "settingsType": "backUpRestore",
            "isGenerateNowClicked": false,
            "name": "Default_Group",
            "triggered": false,
            "_id": "5ec7b2f9632af7d292580cab"
        }
    ],
    "deletePermission": true,
    "archiveSettings": {
        "_id": "backUpRestore",
        "noOfArchives": 3,
        "isWarningEnabled": false,
        "isObjectBackupEnabled": true,
        "archivesSplitUp": {
            "daily": 0,
            "weekly": 3,
            "monthly": 0,
            "yearly": 0
        },
        "isAdvancedSettingsEnabled": true
    },
    "maxNoOfArchives": 1,
    "groupsProgressStatus": [
    ]
  },
  "message": null,
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Backup Details for a Device

Add Device

The API adds an ADC device to the Device Inventory.

Before you begin

Before attempting to add a device with this API, users must ensure ACF permission is available:

- **ADC** > **Device group** > **Add / Modify**

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-add
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	Contains all the params to be sent in the request body for the post request. See the following sections for each vendors: • Add an Array Device • Add a F5XC Device • F5 Payload • Citrix Payload • A10 payload • AVI Payload • Cisco Payload • AmazonELB Payload • Infoblox Payload • BigIQ Payload • HAProxy Payload • NginxPlus Payload

Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Vendor-Specific Information

Add an Array Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor	(Mandatory) Vendor of the device - Array.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is 'Credential List - AppViewX or 'Credential List - CyberArk'. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	(Optional) Modules with which the device is to be added. The possible values are: • LTM • BIG-IP DNS
isControllerDevice *Boolean*	(Optional) Boolean flag to denote if the device to be added is a controller. **Constraints**: The value must be Yes, if the device to be added is Velos Controller.
sshPort *String*	(Optional) Device communication port. The default value is 22. It must be between 1 and 65535.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	The default value is Ignored.
isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether AppViewX group is required. Only for applicable vendors. The default value is False.
isAutoDetectTenants *Boolean*	(Optional) Applicable only for modules - rSeries Host. Boolean value to denote whether tenant devices should be automatically discovered and added to AppViewX. The default value is True.
tokenBasedAuthentication *Boolean*	(Optional) Token Based authentication. The possible values are True/False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
tenantProperties.credentialType *String*	(Optional) Type of credential to be associated for the tenant device. Applicable only for device with module rSeries Host. The possible values are: • Credential List - AppViewX • Credential List - CyberArk • Credential List - Thycotic • Credential List - HashiCorp
tenantProperties.credentialName *String*	(Optional) Name of the credential to be associated with the tenant device. Applicable only for device with module rSeries Host

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	Constraints: • Required only if the credential type is 'Credential List - AppViewX or 'Credential List - CyberArk'. • Credentials of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' and space only. • Length 2 - 64 characters.
tenantProperties.module *String[]*	(Optional) Modules with which the tenant device is to be added. Applicable only for devices with module rSeries Host. The possible values are: • LTM • BIG-IP DNS
tenantProperties.sshPort *String*	(Optional) Device communication port for the tenant device. Applicable only for devices with module rSeries Host. The default value is 22.
tenantProperties.certSync *String*	(Optional) Certificate sync mode for the tenant device. Applicable only for devices with module rSeries Host. The possible values are Managed, Monitored, and Ignored The default value is Ignored.
tenantProperties.isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether the AppViewX group is required for the tenant device. Only for applicable vendors. Applicable only for devices with module rSeries Host. The default value is False.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add an Array device of name testArrayDevice,with IP address 192.168.xxx.xx with both SLB and GSLB modules with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/avxapi/adc-device-add`

Sample Request

```
{
  "payload": {
    "name": "testArrayDevice",
    "communicationAddressType": "ip",
    "deviceCommunicationPort": {
          "apiPort": "9997"
      },

    "ip": "192.168.147.23",
    "vendor": "Array",
    "module": [
      "SLB",
      "GSLB"
    ],
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Manage Device

Add a F5XC Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - F5XC.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • FQDN
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType	(Mandatory) Type of credential to be associated for the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or ’Credential List - CyberArk’. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	(Optional) Modules with which the device is to be added. The possible values are: • LTM • BIG-IP DNS

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
Service *String[]*	Service with which the device to be added. The possible values are: • Distributed App • DNS Management
account *Boolean*	Device to be added - account or services. The possible values are: • true • false
isHostDevice *Boolean*	(Optional) The type of device to be added: either the host or tenant. The possible values are: • true • false The value must be true if the device to be added is vCMP Host or rSeries Host.
hostType *String*	(Optional) Type of the host device to be added. **Constraints**: Mandatory if ‘isHostDevice’ is true. The possible Values are: • vCMP Host • rSeries Host • Velos Partition
isControllerDevice *Boolean*	(Optional) Boolean flag to denote if the device to be added is a controller. **Constraints**: The value must be Yes, if the device to be added is Velos Controller.
sshPort *String*	(Optional) Device communication port. The default value is 22.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	It must be between 1 and 65535.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether AppViewX group is required. Only for applicable vendors. The default value is False.
isAutoDetectTenants *Boolean*	Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
isAutoDetectTenants *Boolean*	(Optional) Applicable only for modules - rSeries Host. Boolean value to denote whether tenant devices should be automatically discovered and added to AppViewX. The default value is True.
tokenBasedAuthentication *Boolean*	(Optional) Token Based authentication. The possible values are True/False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
tenantProperties.credentialType *String*	(Optional) Type of credential to be associated for the tenant device. Applicable only for device with module rSeries Host. The possible values are: • Credential List - AppViewX • Credential List - CyberArk • Credential List - Thycotic • Credential List - HashiCorp
tenantProperties.credentialName *String*	(Optional) Name of the credential to be associated with the tenant device. Applicable only for device with module rSeries Host **Constraints**: • Required only if the credential type is 'Credential List - AppViewX or 'Credential List - CyberArk'. • Credentials of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' and space only. • Length 2 - 64 characters.
tenantProperties.module *String[]*	(Optional) Modules with which the tenant device is to be added. Applicable only for devices with module rSeries Host. The possible values are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• LTM • BIG-IP DNS
tenantProperties.sshPort *String*	(Optional) Device communication port for the tenant device. Applicable only for devices with module rSeries Host. The default value is 22.
tenantProperties.certSync *String*	(Optional) Certificate sync mode for the tenant device. Applicable only for devices with module rSeries Host. The possible values are Managed, Monitored, and Ignored The default value is Ignored.
tenantProperties.isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether the AppViewX group is required for the tenant device. Only for applicable vendors. Applicable only for devices with module rSeries Host. The default value is False.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode	Application specific status code for the response. Will be non-null for failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Add an F5XC device of name testF5XCDevice,with device URL https://appviewx.console.ves.volterra.io with both Distributed App and DNS management Service with username and Auth Token.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add
```

Sample Request

```
{
  "payload": {
    "name": "testF5XCDevice",
    "communicationAddressType": "fqdn",



    "fqdn": "appviewx.console.ves.volterra.io",
    "vendor": "F5XC",
    "service": [
      "Distributed App",
      "DNS management"
    ],
    "account": true,
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```

Sample Response

```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

What's Next

- Manage Device

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Add a F5 Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - F5.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Credential List - CyberArk • Credential List - BeyondTrust
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or ’Credential List - CyberArk’. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	(Mandatory) Modules with which the device is to be added. The possible values are: • LTM • BIG-IP DNS
isHostDevice *Boolean*	(Optional) Device to be added - host or tenant. The possible values are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• true • false **Constraints**: The value must be true if the device to be added is vCMP Host or rSeries Host.
hostType *String*	(Optional) Type of the host device to be added. **Constraints**: Mandatory if ‘isHostDevice’ is true Possible Values: vCMP Host,rSeries Host.
isHostDevice *Boolean*	(Optional) The type of device to be added: either the host or tenant.
hostType *String*	(Optional) Type of the host device to be added. **Constraints**: Mandatory if ‘isHostDevice’ is true. The possible Values are: • vCMP Host • rSeries Host • Velos Partition
sshPort *String*	(Optional) Device communication port. The default value is 22.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether AppViewX group is required. Only for applicable vendors.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	The default value is False.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
isAutoDetectTenants *Boolean*	(Optional) Applicable only for module - rSeries Host. Boolean value to denote whether tenant devices should be automatically discovered and added to AppViewX. The default value is True.
tokenBasedAuthentication *Boolean*	(Optional) Token Based authentication. The possible values are True/False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
tenantProperties.credentialType *String*	(Mandatory) Type of credential to be associated for the tenant device. Applicable only for device with module rSeries Host. The possible values are: • Credential List - AppViewX • Credential List - CyberArk • Credential List - Thycotic • Credential List - HashiCorp

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
tenantProperties.credentialName *String*	(Mandatory) Name of the credential to be associated with the tenant device. Applicable only for device with module rSeries Host. **Constraints**: • Required only if credential type is 'Credential List - AppViewX or 'Credential List - CyberArk'. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' and space only. • Length 2 - 64 characters.
tenantProperties.module *String[]*	(Mandatory) Modules with which the tenant device is to be added. Applicable only for devices with module rSeries Host. The possible values are: • LTM • BIG-IP DNS
tenantProperties.sshPort *String*	(Optional) Device communication port for the tenant device. Applicable only for devices with module rSeries Host. The default value is 22.
tenantProperties.certSync *String*	(Optional) Certificate sync mode for the tenant device. Applicable only for devices with module rSeries Host. Possible values are Managed, Monitored, Ignored The default value is Ignored.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
tenantProperties.isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether the AppViewX group is required for the tenant device. Only for applicable vendors. Applicable only for devices with module rSeries Host. The default value is False.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add an F5 device of name testF5Device,with IP address 192.168.xxx.xx with both LTM and BIG-IP DNS modules with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request

```
{
  "payload": {
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "name": "testF5Device",
    "communicationAddressType": "ip",
    "ip": "192.168.112.93",
    "vendor": "F5",
    "module": [
      "LTM",
      "BIG-IP DNS"
    ],
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }

}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Use Case

Add an F5 vcmp host device of name vcmpHost1,with IP address 192.168.xxx.xx with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request


```
{
  "payload": {
    "name": "vcmpHost1",
    "communicationAddressType": "ip",
    "ip": "192.168.112.93",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "vendor": "F5",
    "isHostDevice": true,
    "hostType": "vCMP Host",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Use Case

Add an F5 rSeries Host device of name testF5Device,with IP address 192.168.xxx.xx and auto detect all the tenants with LTM module.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request


```
{
  "payload": {
    "name": "testF5Device",
    "communicationAddressType": "ip",
    "ip": "192.168.2.10",
    "isAutoDetectTenants": true,
    "vendor": "F5",
    "isHostDevice": true,
    "hostType": "rSeries Host",
    "credentialType": "Manual Entry",
    "userName": "admin",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "password": "QVZYIUY1VEVTVA==",
    "tenantProperties": {
      "credentialType": "Credential List - AppViewX",
      "module": [
        "LTM"
      ],
      "credentialName": "avx"
    }
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Use Case

Add an F5 Velos Controller device of name testControllerDevice,with IP address 192.168.xxx.xx.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request


```
{
  "payload": {
    "name": "testControllerDevice",
    "communicationAddressType": "ip",
    "ip": "192.168.2.10",
    "vendor": "F5",
    "isControllerDevice": true,
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "QVZYIUY1VEVTVA=="
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 }
}
```

Sample Response

```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Use Case

Add an F5 Velos Partition device of name testPartitionDevice,with IP address 192.168.xxx.xx and auto detect all the tenants with the LTM module.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request

```
{
  "payload": {
    "name": "testPartitionDevice",
    "communicationAddressType": "ip",
    "ip": "192.168.2.10",
    "isAutoDetectTenants": true,
    "vendor": "F5",
    "isHostDevice": true,
    "hostType": "Velos Partition",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "QVZYIUY1VEVTVA==",
    "tenantProperties": {
      "credentialType": "Credential List - AppViewX",
      "module": [
        "LTM"
      ],
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "credentialName": "avx"
   }
  }
}
```

Sample Response

```
{
 "response": "Device is added to the inventory.",
 "message": "Device has been successfully added and configuration parse is triggered.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```

What's Next

- Manage Device
- Search for Device

Add a Citrix Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - Citrix.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk • Credential List - BeyondTrust
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or ’Credential List - CyberArk’. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module	(Mandatory) Modules with which the device is to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String[]	The possible values are: • SLB • GSLB
sshPort *String*	(Optional) Device communication port. The default value is 22. It must be between 1 and 65535.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
dataCenter *String*	(Optional) Data center with which the device is to be added.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response	Success message or failure description in case of error.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add a Citrix device of name testF5Device,with IP address 192.168.xxx.xx with both SLB and GSLB modules, with username and password.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add

Sample Request

```
{
  "payload": {
    "name": "testCitrixDevice",
    "communicationAddressType": "ip",
    "ip": "192.168.112.86",
    "vendor": "Citrix",
    "module": [
      "SLB",
      "GSLB"
    ],
    "certSync": "Monitored",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```

Sample Response

```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


What's Next

- Manage Device

Add a Cloudflare Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
accessUrl	(Mandatory) Access URL of the Cloudflare.
accountNumber	(Mandatory) Account number.
vendor *String*	(Mandatory) Vendor of the device - Cloudflare.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible value is: • Manual Entry
module *String[]*	(Mandatory) Modules with which the device is to be added. The possible value is: • GSLB
dataCenter	(Optional) Data center with which the device is to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add a Cloudflare device of name cloudflare4,with access URL https://api.cloudflare.com/client/v4/ with account number b15f0ecdf78c8b0d27923d2db551844f.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request

```
{
   "payload": {
      "name": "cloudflare4",
      "accessUrl": "https://api.cloudflare.com/client/v4/",
      "accountNumber": "b15f0ecdf78c8b0d27923d2db551844f",
      "vendor": "Cloudflare",
      "module": [
         "GSLB"
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    ],
    "credentialType": "Manual Entry",
    "password": "U0xJR1hPajFpcHlYNWFPeEdHX1ZnNGhfeGVZenlvRjhIbVlMdUw3Mg=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


What's Next

- Manage Device

Add an A10 Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - Citix.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or ’Credential List - CyberArk’. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	(Mandatory) Modules with which the device is to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
sshPort *String*	(Optional) Device communication port. The default value is 22. It must be between 1 and 65535.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether AppViewX group is required. Only for applicable vendors. The default value is False.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add an A10 device of name testA10Device,with IP address 192.168.xx.xxxx with both SLB and GSLB modules, with username and password.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add

Sample Request

```
{
  "payload": {
    "name": "testA10Device",
    "communicationAddressType": "ip",
    "ip": "192.168.xx.xxx",
    "vendor": "A10",
    "module": [
      "SLB",
      "GSLB"
    ],
    "certSync": "Monitored",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YTEw",
    "enablePassword": "YTEw"
  }
}
```

Sample Response

```
{
  "response": "Device is added to the inventory.",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 "message": "Device has been successfully added and configuration parse is triggered.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


What's Next

- Manage Device

Add an AVI Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - AVI.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType	(Mandatory) Type of credential to be associated for the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or ’Credential List - CyberArk’. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	(Mandatory) Modules with which the device is to be added. The possible values are: • SLB • GSLB
certSync *String*	(Optional) Certificate sync mode. The possible values are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Managed • Monitored • Ignored The default value is Ignored.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add an AVI device of name testAVIDevice,with IP address 192.168.xx.xx with both SLB and GSLB modules, with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request


```
{
  "payload": {
    "name": "testAVIDevice",
    "communicationAddressType": "ip",
    "ip": "192.168.xx.xx",
    "vendor": "AVI",
    "module": [
      "SLB",
      "GSLB"
    ],
    "certSync": "Monitored",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


What's Next

- Manage Device

Add a Cisco Device

Payload

Name	Description
name	(Mandatory) Name of the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	**Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - Cisco.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible value is IP.
ip *String*	(Mandatory) IP of the device. **Constraints**: The IP address must be in IPV4 format.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Required only if credential type is 'Credential List - AppViewX or 'Credential List - CyberArk'. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	(Mandatory) Modules with which the device is to be added. The possible value is SLB.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add a Cisco device of name testCiscoDevice,with IP address 192.168.41.26 with SLB module, with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request

```
{
  "payload": {
    "name": "testCiscoDevice",
    "communicationAddressType": "IP",
    "ip": "192.168.41.26",
    "vendor": "Cisco",
    "module": [
      "SLB"
    ],
    "certSync": "Monitored",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


What's Next

- Manage Device

Add an Amazon ELB Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN
ip *String*	(Mandatory) IP of the device. **Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Required only if communicationAddressType is IP. • The IP address must be in IPV4 format.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or ’Credential List - CyberArk’. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module	(Mandatory) Modules with which the device is to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String[]	
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
sshPort *String*	(Optional) Device communication port. The default value is 22. It must be between 1 and 65535.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
isAppViewXSyncRequired *Boolean*	(Optional) Boolean value to denote whether AppViewX group is required. Only for applicable vendors. The default value is False.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
dataCenter *String*	(Optional) Data center with which the device is to be added.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

NA

Sample Request

NA

Sample Response

NA

What's Next

- Manage Device

Add an Infoblox Device

Payload

Name	Description
name	(Mandatory) Name of the device.
String	**Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - Infoblox.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values are: • IP • FQDN
ip *String*	(Mandatory) IP of the device. **Constraints**: • Required only if communicationAddressType is IP. • The IP address must be in IPV4 format.
fqdn *String*	(Mandatory) FQDN of the device. **Constraints**: Required only if communicationAddressType is FQDN.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is 'Credential List - AppViewX. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
isAutoDetectSecondary *Boolean*	(Optional) Boolean value to denote whether secondary devices should be automatically discovered and added to AppViewX. The default value is False.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Add an Infoblox device of name testInfobloxDevice,with IP address 192.168.41.30 with username and password.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add

Sample Request


```
{
  "payload": {
    "name": "testInfobloxDevice",
    "communicationAddressType": "IP",
    "ip": "192.168.41.30",
    "vendor": "Infoblox",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


What's Next

- Manage Device

Add a BigIQ Device

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - BigIQ.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible values is IP.
ip *String*	(Mandatory) IP of the device. **Constraints**: Required only if communicationAddressType is IP.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName	(Mandatory) Name of the credential to be associated with the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	**Constraints**: • Required only if credential type is 'Credential List - AppViewX or Credential List - CyberArk. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags	More info in case of failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add a BigIQ device of name testBigIQDevice,with IP address 192.168.41.28 with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request


```
{
  "payload": {
    "name": "testBigIQDevice",
    "communicationAddressType": "IP",
    "ip": "192.168.41.29",
    "vendor": "BigIQ",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


What's Next

- Manage Device

Add a HAProxy Device

Payload

Name	Description
name	(Mandatory) Name of the device.
String	**Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - HAProxy.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible value is IP.
ip *String*	(Mandatory) IP of the device. **Constraints**: The IP address must be in IPV4 format.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk • Credential List -BeyondTrust
userName *String*	(Mandatory) User name of the device. **Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Required only if credential type is 'Credential List - AppViewX or Credential List - CyberArk. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	Modules with which the device is to be added. The possible value is SLB.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response	Success message or failure description in case of error.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add a HAProxy device of name testHAProxyDevice,with IP address 192.168.41.27 with SLB module, with username and password.

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add

Sample Request


```
{
  "payload": {
    "name": "testHAProxyDevice",
    "communicationAddressType": "IP",
    "ip": "192.168.41.27",
    "vendor": "HAProxy",
    "module": [
      "SLB"
    ],
    "certSync": "Monitored",
    "credentialType": "Manual Entry",
    "userName": "admin",
    "password": "YXBwdGFnQDEyMw=="
  }
}
```


Sample Response


```
{
  "response": "Device is added to the inventory.",
  "message": "Device has been successfully added and configuration parse is triggered.",
  "appStatusCode": null,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "tags": null,
  "headers": null
}
```

What's Next

- Manage Device

Add an NginxPlus Device

Payload

Name	Description
name *String*	(Mandatory) Name of the device. **Constraints**: • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
vendor *String*	(Mandatory) Vendor of the device - NginxPlus.
communicationAddressType *String*	(Mandatory) Communication type to the device. The possible value is IP.
ip *String*	(Mandatory) IP of the device. **Constraints**: The IP address must be in IPV4 format.
credentialType *String*	(Mandatory) Type of credential to be associated for the device. The possible values are: • Manual Entry • Credential List - AppViewX • Credential List - CyberArk
userName	(Mandatory) User name of the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	**Constraints**: Required only if credential Type is Manual entry. It will not accept ‘,”,<,>.
password *String*	(Mandatory) Encoded value of the password to authenticate with the device. **Constraints**: Required only if credential Type is Manual entry.
credentialName *String*	(Mandatory) Name of the credential to be associated with the device. **Constraints**: • Required only if credential type is ’Credential List - AppViewX or Credential List - CyberArk. • Credential of the requested name should be added in AppViewX already. • Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.
module *String[]*	Modules with which the device is to be added. The possible value is SLB.
certSync *String*	(Optional) Certificate sync mode. The possible values are: • Managed • Monitored • Ignored The default value is Ignored.
dataCenter *String*	(Optional) Data center with which the device is to be added. **Constraints**:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
	• Accepted special characters: '-' , '_' , '.' , '*' , '|' , '!' and space only. • Length 2 - 64 characters.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Device has been successfully added and configuration parse is triggered	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_INV_0004	Validation error	One or more fields are invalid. Reason for validation failure and possible remediation will be available in ‘tags’ field of response.
409 Conflict	ADC_INV_0001	Device(s) Already Exists	Device already exists in AppViewX. Change the name of the device and IP/FQDN.
409 Conflict	ADC_INV_0006	Device Name Already Exists.	Change the name of the device and retry adding.
409 Conflict	ADC_INV_0007	IP Already Exists	Change the IP of the device and retry adding.
409 Conflict	ADC_INV_0027	FQDN already exists	Change the FQDN field of the device and retry adding.
404 Not Found	ADC_INV_1076	Device not available for the given details.	Check if the requested device is added in the inventory.

Sample Request/Response

Use Case

Add an NginxPlus device of name testNginxPlusDevice,with IP address 192.168.41.28 with SLB module, with username and password.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-add`

Sample Request

```
{
  "payload": {
    "name": "testNginxPlusDevice",
    "communicationAddressType": "IP",
    "ip": "192.168.41.28",
    "vendor": "NginxPlus",
    "module": [
      "SLB"
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  ],
  "certSync": "Monitored",
  "credentialType": "Manual Entry",
  "userName": "admin",
  "password": "YXBwdGFnQDEyMw=="
 }
}
```


Sample Response


```
{
 "response": "Device is added to the inventory.",
 "message": "Device has been successfully added and configuration parse is triggered.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


What's Next

- Manage Device

Delete Device

This API deletes any device that is available in inventory.

Before you begin

Before attempting to delete devices with this API, users must ensure the ACF permission is available:

- **ADC** > **Inventory** > **Delete**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device
Type:	DELETE
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device?gwsource=external

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

	To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
deviceName	(Mandatory) Name of the device to be deleted.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Query	**Constraints**: It can contain only alphanumeric characters, ‘-’ , ’_’ , ‘.’ , ‘*’ , ‘|’, ‘!’.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	Device(s) deleted succesfully.	-
400 Bad Request	ADC_INV_1562	Invalid Device Name. Device	Device name is not given/empty value is given. Give a valid device name in query param field - ‘deviceName’

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
		name cannot be null or empty.	
400 Bad Request	ADC_INV_1561	All the given devices are invalid.	All the given device(s) are invalid. Check if the device is already added in the ADC inventory, Check if the requesting user has RW permission for the device.
404 Not Found	ADC_INV_1559	All the given devices are not found.	Check if the device is already added in the ADC inventory.
403 Forbidden	ADC_INV_1560	All the given devices doesn't have sufficient permission.	Check if the requesting user has RW permission for the device. This can be verified by navigating to Account -> Resource page and checking under Device tab for the user’s resource

Sample Request/Response

Use Case

Delete two devices, 192.168.40.150 and test available in the ADC inventory.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device`

Sample Request

```
NA
```

Sample Response


```
    {
 "appStatusCode": null,
 "headers": null,
 "message": "Device(s) deleted succesfully.",
 "response": "Device(s) deleted succesfully.",
 "tags": {
  "Deleted Device(s)": [
   "Inventory_Device_ADC_Delete_F5V13_StandAlone"
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  ],
  "Invalid Device(s)": []
 }
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Add Device

Search for Device

This API searches for ADC devices that are available in the inventory for any user.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to search devices with this API, users must have the ACF permission

- **ADC** > **Inventory** > **View**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-search
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-search?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	Contains all the params to be sent in the request body for the post request. See Payload details. **Type**: Key value pair

Payload

Name	Description
input *Key value pair*	(Mandatory) Key value pair to search devices. The possible value is Input.
filter *Key value pair*	(Optional) Filter params to be applied on response such as start,limit and sort. The possible value is Filter.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	Matching results found for the given input.	-
200 OK	-	No matching results found.	-
400 Bad Request	ADC_DEVICE_SEARCH_001	Invalid search input.	• Check and ensure if non-null/non-empty value is given in field - ‘payload’. • Check and ensure if non-null/non-empty value is given under payload field - ‘input’.

Sample Request/Response

Use Case

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Get the details of the class management widget with name classWgt under the dashboard test.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-search`

Sample Request


```
{
  "payload": {
    "input": {
      "vendor": "AVI",
      "haActive": "Active"
    }
  }
}
```


Sample Response


```
{
  "response": {
    "devices": [
      {
        "name": "192.168.40.161",
        "ip": "192.168.40.161",
        "fqdn": null,
        "deviceType": [
          "SLB",
          "GSLB"
        ],
        "module": [
          "SLB",
          "GSLB"
        ],
        "dataCenter": "CBE",
        "vendor": "AVI",
        "version": "v12",
        "detailedVersion": "12.0 build 56.20.nc",
        "subsystem": "LoadBalancer",
        "category": "ADC",
        "partition": [
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "default",
    "Other",
    "partition1@sted",
    "_partition-spl.=@sted",
    "API_SB_Test"
  ],
  "keywords": [
    "192.168.40.161",
    "192.168.40.161",
    "SLB",
    "GSLB",
    "CBE",
    "AVI",
    "v12",
    "12.0 build 56.20.nc",
    "Manual Entry",
    "Managed",
    "Active",
    "HE2H81UJ47",
    "22",
    "default",
    "Other",
    "partition1@sted",
    "_partition-spl.=@sted",
    "API_SB_Test",
    "API"
  ],
  "access": [
    {
      "accessType": "API",
      "userName": "nsroot",
      "password": "******",
      "credentialStoreId": null,
      "community": null,
      "snmpVersion": null,
      "key": "xxxxxx",
      "secondaryPassword": null,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "secondaryPassKey": null,
    "accessToken": null,
    "url": null
  }
 ],
 "enablePassword": null,
 "enablePasswordKey": null,
 "sameCredential": false,
 "credentialType": "Manual Entry",
 "credentialName": "",
 "description": null,
 "status": "Managed",
 "objectSummary": [
  {
   "name": null,
   "object": "SLB Virtual Servers",
   "count": 138,
   "_id": null
  },
  {
   "name": null,
   "object": "CS Virtual Servers",
   "count": 50,
   "_id": null
  },
  {
   "name": null,
   "object": "GSLB Virtual Servers",
   "count": 153,
   "_id": null
  }
 ],
 "deviceStatusList": [
  {
   "title": "Device communication",
   "message": "Primary Device communication successful",
   "status": "Success",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
        "timeStamp": 1597862177000
      },
      {
        "title": "Device provisioned modules check",
        "message": "Selected device type matches with the Provisioned Modules.",
        "status": "Success",
        "timeStamp": 1597862178000
      },
      {
        "title": "Downloading configuration file",
        "message": "Downloaded configuration file",
        "status": "Success",
        "timeStamp": 1597862224000
      },
      {
        "title": "Comparing the changes in configuration file",
        "message": "Configuration files changed.",
        "status": "Success",
        "timeStamp": 1597862228000
      },
      {
        "title": "Parsing downloaded SLB configuration file(s)",
        "message": "Objects added to AppViewX - Virtual Server:138, Service Group:97, Service Group Member:97, Service:36, CS VirtualServer:50,
Server:329",
        "status": "Success",
        "timeStamp": 1597862228000
      },
      {
        "title": "Parsing downloaded GSLB configuration file(s)",
        "message": "Objects added to AppViewX - GSLB Virtual Server153, Service:44, Server:329",
        "status": "Success",
        "timeStamp": 1597862231000
      },
      {
        "title": "Subscription for syslog notifications from device",
        "message": "Registered with device for syslog subscription",
        "status": "Success",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "timeStamp": 1597862249000
  }
 ],
 "registrationKey": null,
 "serialNumber": "HE2H81UJ47",
 "modelNumber": "1000",
 "expiryDate": null,
 "serviceCheck": null,
 "manageCertificate": "Managed",
 "csrfToken": null,
 "enableEvent": false,
 "deviceCommunicationPort": {
   "sshPort": 22,
   "httpsPort": 443
 },
 "lastConnectedIP": null,
 "lastRebootTime": "4:27AM  up 23 days",
 "haMode": null,
 "haActive": "Active",
 "highAvailability": [

 ],
 "lastConnectedHAList": null,
 "synchronizationGroups": null,
 "synchronizationGroupsMode": null,
 "syncGroup": null,
 "syncStatus": null,
 "trafficGroups": null,
 "syncFailoverGroupName": "",
 "syncOnlyGroups": null,
 "autoSyncEnabled": false,
 "autoSyncNeeded": false,
 "primaryDeviceId": null,
 "childDevicesInfo": null,
 "parentDeviceId": null,
 "parentDeviceName": null,
 "deviceBackupSettings": {
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "name": null,
    "mode": null,
    "dayNumber": null,
    "dateNumber": null,
    "weekNumber": null,
    "month": null,
    "frequency": null,
    "startingFrom": null,
    "archiveCount": 5,
    "isDate": null,
    "isConfigBackUpEnabled": null,
    "configSettingsBackUpType": null,
    "configBackUpTime": null,
    "_id": null
  },
  "deviceGroups": [

  ],
  "attributes": {
    "syslogHost": "192.168.142.156:5514"
  },
  "generalAlert": null,
  "thresholdAlert": null,
  "deviceUrl": null,
  "agentUrl": null,
  "devIndex": 0,
  "platform": null,
  "userAction": null,
  "model": null,
  "sessionId": null,
  "tmosVersion": null,
  "createdBy": "admin",
  "noDetect": false,
  "createdDateTime": 1597816172139,
  "lastModifiedBy": "system",
  "lastModifiedDateTime": 1597862279214,
  "associatedDevices": null,
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "deviceSyncStatus": null,
    "hostName": "AVI_12.0_standalone",
    "previousDeviceStatus": "Managed",
    "deviceCommunication": "Success",
    "localLBConfigUpdated": null,
    "localLBConfigDownload": "Success",
    "localLBConfigParsing": "Success",
    "globalLBConfigUpdated": null,
    "globalLBConfigDownload": "Success",
    "globalLBConfigParsing": "Success",
    "lastConfigUpdatedTime": 1597862279168,
    "communicationAddressType": "ip",
    "lastPostConfigTriggeredTime": 1597862279476,
    "regions": null,
    "accountNumber": null,
    "discoverySource": null,
    "deviceUpgraded": false,
    "postConfigFlag": false,
    "authToken": null,
    "tokenBasedAuthentication": false,
    "lastConfigFetchTriggeredTime": 1597816251693,
    "managementIp": null,
    "lastManagedTime": 1597862279168,
    "_id": "5f3cbd6cecd19733ebccc6fb",
    "vLans": null,
    "iHealthReport": null,
    "isPrimary": true,
    "isAutoDetect": true,
    "isSecondaryCommunication": false,
    "isHostedOnHypervisor": false
  }
],
"totalRecords": 1,
"obtainedRecords": 1,
"obtainedRecordRange": {
  "start": 0,
  "end": 0
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  }
 },
 "message": "Matching results found for the given input.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Manage Device
- Unmanage a Device

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Delete a Widget
- Delete Device

Sync Configuration

This API triggers configuration sync for a device available in Device Inventory.

Before you begin

Before attempting to initiate configuration sync with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Inventory** > **Fetch config**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-config-sync
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-config-sync?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String **Constraints**: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Description
deviceName *String*	(Mandatory) Name of the device. **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message with device name or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	Config sync triggered succesfully.	-
400 Bad Request	ADC_INV_1559	Invalid Device Name. Device name cannot be null or empty.	• Check and ensure if non-null/ non-empty value is given in field - ‘payload’. • Check and ensure if non-null/non-empty value is given under payload field - ‘deviceName’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	AVX-VLDTN-001	Invalid payload data.	Requested value in the deviceName field does not match the accepted pattern.
404 Not Found	ADC_INV_0010	Device not found.	Check if the requested device is added in the inventory.
403 Forbidden	ADC_INV_0005	Config fetch not allowed for Unmanaged/ Queued devices and Device(s) on which other operations are already in progress.	-
403 Forbidden	ADC_INV_1558	Cannot access resource, permission not available.	Check if Read/Read Write permission is available for the requested device.

Sample Request/Response

Use Case

Trigger configuration sync for the device - testDevice.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-config-sync`

Sample Request

```
{
  "payload": {
    "deviceName": "192.168.42.150"
  }
}
```

Sample Response

```
{
  "response": "Config sync triggered for given device - 192.168.42.150",
  "message": "Config sync triggered successfully.",
  "appStatusCode": null,
  "tags": null,
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
  "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Get Object Configuration

Manage Device

This API changes the unmanaged state to managed state. Before the state of the device changed to managed state, the device configuration sync process was triggered.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Before you begin

Before attempting to manage devices with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Inventory** > **Manage / Unmanage**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-manage
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-manage?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String
	Constraints: The session ID is used when username and password are not provided.
username	(Mandatory) Specifies the login username used to access the AppViewX.
Header	**Type**: String
	Constraints: The username, along with the password, is used when the session ID is not provided.
password	(Mandatory) Specifies the login password used to access the AppViewX.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be 'application/json'.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Payload

Name	Mandatory	Description	Field Type	Constraints
deviceName *String[]*	Yes	(Mandatory) Name of the device(s) to manage. **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.		

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
202 Accepted	-	State change to manage the device has been initiated.	-
400 Bad Request	ADC_INV_0234	Device name is mandatory.Value cannot be null or empty.	Check and ensure a valid value is provided in field deviceName.
400 Bad Request	AVX-VLDTN-001	Invalid payload data.	Check and ensure a valid value is provided in field payload.
404 Not Found	ADC_INV_0010	Device not found.	Any of the given devices is not found in the database. Ensure if all the requested devices are added in inventory.
403 Forbidden	ADC_INV_0008	Selected device(s) are not in un-managed state. Operation cannot be performed.	All the requested devices are not in unmanaged state. Requested devices should be in unmanaged state to perform manage action.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Sample Request/Response

Use Case

Change the state of device ADC_DEVICE_1 from unmanaged to managed.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-manage`

Sample Request

```
{
  "payload": {
    "deviceName": [
      "ADC_DEVICE_1"
    ]
  }
}
```

Sample Response

```
{
  "response": "State change to manage the device has been initiated",
  "message": "State change to manage the device has been initiated",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

  The tenant name will be included in the endpoint URL for a SaaS deployment.

- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Unmanage a Device

Unmanage a Device

This API changes the state of ADC devices from managed state to unmanaged state. No action can be performed on the device(s), which are in the unmanaged state and its objects/applications.

Before you begin

Before attempting to manage devices with this API, users must ensure any one of the ACF permission is available:

- **ADC** > **Inventory** > **Manage / Unmanage**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/adc-device-unmanage
Type:	PUT
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-unmanage?gwsource=external To understand the elements of the sample URL, click here.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
deviceName *String[]*	(Mandatory) Name of the device(s) to manage. **Constraints**: It can contain only alphanumeric characters, '-' , '_' , '.' , '*' , '|', '!'.

Response Structure

- **Status Code:** 202 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
202 Accepted	-	State change to unmanage the device has been initiated.	-

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
400 Bad Request	ADC_INV_0234	Device name is mandatory.Value cannot be null or empty.	Check and ensure a valid value is provided in field deviceName.
400 Bad Request	AVX-VLDTN-001	Invalid payload data.	Check and ensure a valid value is provided in field payload.
404 Not Found	ADC_INV_0010	Device not found.	Any of the given devices are not found in the database. Ensure if all the requested devices are added in inventory
403 Forbidden	ADC_INV_0009	Selected device(s) are in in-progress/un-managed state. Operation cannot be performed.	All the requested devices are in unmanaged/inprogress state. Requested devices should not be in Inprogress/ unmanaged state.

Sample Request/Response

Use Case

Change the state of device ADC_DEVICE_1 from managed to unmanaged.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/adc-device-unmanage
```

Sample Request

```
{
  "payload": {
    "deviceName": [
      "ADC_DEVICE_1"
    ]
  }
}
```

Sample Response

```
{
  "response": "State change to unmanage the device has been initiated.",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 "message": "State change to unmanage the device has been initiated.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Manage Device

Dashboard

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

By using the following APIs, you can manage, monitor, and interpret all the configured applications and their objects.

Create an Application Widget

This API creates an Application widget and configures objects.

Before you begin

Before attempting to create an application widget with this API, users must ensure ACF permission is available:

- **ADC** > **Dashboard Application widget** > **Add /Delete**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/dashboard-widget-applicationview`
Type:	`POST`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/dashboard-widget-applicationview?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
dashboardName *Query*	(Mandatory) Name of the dashboard in which a widget is to be created. If the given dashboard is not available a new dashboard will be created in the given name. **Constraints**: Accepts only '_', '-', ':', '.', '|'(Should not be empty and not start with special characters). **Type**: String
payload *Body*	(Mandatory) Contains all the params to be sent in the request body for the post request. See Payload section for more details.

The input supports up to three hierarchical levels of groups. group, subGroup1, subGroup2, subGroup3 are static keys.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Payload

Name	Description
name *String*	(Mandatory) Name of the widget. **Constraints**: Accepts only '_', '-', ':', '.', '|'(Should not be empty and not start with special characters).
group *Key value pair[]*	(Optional) Contains attributes denoting the group to be created. For more details, see Group.

Widget Group

Name	Description
name *String*	(Mandatory) Name of the group. **Constraints:** Accepts only '_', '-', ':', '.', '|'(Should not be empty and not start with special characters).
action *Key value pair[]*	(Optional) Contains a list of attributes denoting the actions to be added in the group. For more details, see Action.
objects *Key value pair[]*	(Optional) Contains a list of attributes related to the object to be added in the group such as hierarchy, vendor, objectType. For more details, see ObjectInfo.
subGroup1 *Key value pair[]*	(Optional) Subgroups to be added within this group. Only three levels of groups are possible. Groups can be mentioned with static keys group, subGroup1, subGroup2 within one another. For more details, see Group.

ObjectInfo

Name	Description
vendor	(Optional) Device vendor.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
objectType *String*	(Optional) Object Type as mentioned in AppViewX.
hierarchy *String*	(Optional) Hierarchy to display.
selectedList *Key value pair[]*	(Optional) List of objects along with actions to be associated with the object. For more details, see Object.

Object

Name	Description
resourceId *String*	(Optional) Resource id of the object to be added in the widget group. **Note:** Use the search API to search/get the resource ID.
actions *String[]*	(Optional) List of actions to be associated with this object. Action name should be the custom action name defined under ‘action.name’ property of the group.

Action

Name	Description
name *String*	(Optional) Name to be given for the particular action in this group.
type *String*	(Optional) Type of action. Limited to the actions that are supported in AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-		-
400 Bad Request	DASH_WID_APPVIEW_0001	Mandatory field 'dashboardName' is missing	Check and ensure if a non-null/ non-empty value is given in query param field - ‘dashboardName’
400 Bad Request	ADC_DASH_0511	Exception : Invalid Widget name	Check and ensure if a valid value for widget name is given in payload field - ‘name’. Field accepts only '_', '-', ':', '.', '|'(Should

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
			not be empty and not start with special characters).
400 Bad Request	ADC_DASH_0517	Exception : Invalid Action	Check the action name given in the group field - ‘action.type’. Limited to actions supported in AppViewX
400 Bad Request	ADC_DASH_0518	Exception : Invalid hierarchy	Check the value given under the object field - ‘hierarchy’ is supported. Valid hierarchies for requested vendor and objectType can be found in tags field of response also
400 Bad Request	ADC_DASH_0014	Invalid dashboard name.	Change the value given in the query param field - ‘dashboardName’ and retry. dashboardName field accepts only '_', '-', ':', '.', '|'(Should not be empty and not start with special characters).
400 Bad Request	ADC_DASH_0516	Exception : Duplicate Action	Check if any of the action is already configured in the group or parentGroup. Group name in which duplicate is found will be available in tags field of response
403 Forbidden	ADC_DASH_0005	Permission not available for dashboard.	Check if the logged in user has access to the requested dashboard
409 Conflict	ADC_DASH_0010	Widget name already exists.	Change the widget name in the request and retry

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
404 Not Found	DASH_WID_APPVIEW_0003	Object not found	Check if requested objects are available in AppViewX. Missing objects resourceIds will be available under tags field of response.
500 Internal Server Error	ADC_DASH_0507	Exception occured while adding Application widget	• Check if given value under object field ‘vendor’ is support by AppViewX • Check if given value under object field ‘objectType’ is support by AppViewX
409 Conflict	ADC_DASH_0607	Object Count of the Widget exceeds threshold limit {thresholdLimit}. Currently added Object count : {CurrentCountInRequest}	Remove input objects so the input count should be within allowed threshold limit(1000)

Sample Request/Response

Use Case

- Create an application widget with a name appwidget and with a group group_1.
- group_1 to have two actions configured – enable and disable.
- group_1 to have two virtual servers – ‘vs:@5f4e2ffafcbb9809756c6b4d:@ACLTest:@Common’ and ‘vs:@5f4e2ffafcbb9809756c6b4d:@ACLTest1:@Common’ configured in it.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Both the virtual servers to display all the hierarchical children in the group (hierarchy: all) ‘vs:@5f4e2ffafcbb9809756c6b4d:@ACLTest:@Common’ and ‘vs:@5f4e2ffafcbb9809756c6b4d:@ACLTest1:@Common’ to have actions - enable(en) and disable(dis) configured. group_1 to have child group ‘grp_sub1’ inside it .
- ‘grp_sub1’ to have two wide IP inside – ‘gw:@5f4e3423fcbb9809756c6b82:@test.com:@Common:@a’ and ‘gw:@5f4e3423fcbb9809756c6b82:@AppViewXSyslog.com:@Common:@a’.
- Both the wide IPs to display all the hierarchal children in the group (hierarchy: all)
- Both the wide IPs to be configured with actions enable persistence(enPer) and disable persistence(disPer).
- ‘grp_sub1’ to be configured with actions enable persistence(enPer) and disable persistence(disPer).

Request URL

https://<IP/HostName/TenantName>:<GWPORT>/avxapi/dashboard-widget-applicationview

Sample Request

```
{
  "payload": {
    "name": "appwidget",
    "group": [
      {
        "name": "group_1",
        "action": [
          {
            "name": "en",
            "type": "Enable"
          },
          {
            "name": "dis",
            "type": "Disable"
          }
        ],
        "objects": [
          {
            "vendor": "F5",
            "objectType": "VirtualServer",
```

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
    "hierarchy": "All",
    "selectedList": [
     {
       "resourceId": "vs:@5f4e2ffafcbb9809756c6b4d:@ACLTest:@Common",
       "actions": [
         "en",
         "dis"
       ]
     },
     {
       "resourceId": "vs:@5f4e2ffafcbb9809756c6b4d:@ACLTest1:@Common",
       "actions": [
         "en",
         "dis"
       ]
     }
    ]
  }
 ],
 "subGroup1": [
  {
    "name": "grp_sub1",
    "action": [
     {
       "name": "enPer",
       "type": "EnablePersistence"
     },
     {
       "name": "disPer",
       "type": "DisablePersistence"
     }
    ],
    "objects": [
     {
       "vendor": "F5",
       "objectType": "wideIp",
       "hierarchy": "All",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
            "selectedList": [
              {
                "resourceId": "gw:@5f4e3423fcbb9809756c6b82:@gowthamTest.com:@Common:@a",
                "actions": [
                  "enPer",
                  "disPer"
                ]
              },
              {
                "resourceId": "gw:@5f4e3423fcbb9809756c6b82:@AppViewXSyslog.com:@Common:@a",
                "actions": [
                  "enPer",
                  "disPer"
                ]
              }
            ]
          }
        ]
      }
    ]
  }
 ]
}
```


Sample Response


```
{
  "response": "Application widget is created under the given dashboard.",
  "message": "Application widget is created under the given dashboard.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```


Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete a Widget

Get Class Management Widget Details

This API gets the details for a given class management widget.

Before you begin

Before attempting to get class management widget with this API, user must ensure any one of the ACF permission is available:

- **ADC** > **Dashboard** > C**lass management widget** > **Add / Delete**
- **ADC** > **Dashboard** > **Class management widget** > **Execute actions**
- **ADC** > **Dashboard** > **Class management widget** > **Modify settings**

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	/dashboard-widget
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/dashboard-widget?gwsource=external To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId *Header*	(Mandatory) Specifies the session ID of the user session received after login. **Type**: String **Constraints**: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
dashboardName *Query*	(Mandatory) Name of the dashboard. **Constraints**: Allowed special characters : '_', '-', ':', '.', '|' (Should not be empty and not start with special characters). **Type**: String
widgetName *Query*	(Mandatory) Name of the class management widget. **Constraints**: Allowed special characters : ‘-’ ,’_’ , ‘.’ , ‘:’ , ’|’ **Type**: String

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Contains the requested class management widget details.
message *String*	Success message or failure description in case of error.
appStatusCode	Application specific status code for the response. Will be non-null for failure response.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Name	Description
String	
tags	More info in case of failure response.
String	

Status Codes

HTTP Status Code	appStatusCode	Message	Possible remediation
200 OK	-	Widget details retrieved successfully.	-
400 Bad Request	ADC_DASH_0601	Dashboard name is mandatory	Check and ensure if a valid value is given in query param field - dashboardName
400 Bad Request	ADC_DASH_0602	Widget name is mandatory	Check and ensure if a valid value is given in the query param field - widgetName.
400 Bad Request	ADC_DASH_0014	Invalid dashboard name.	Requested dashboard name is invalid. Dashboard name accepts only '_', '-', ':', '.', '|'(Should not be empty and not start with special characters).
400 Bad Request	ADC_DASH_0008	Widget name accepts only - , _ , . , : , |.	Requested widget name is invalid.Widget name accepts only - , _ , . , : , |.
400 Bad Request	ADC_DASH_0009	Widget name cannot exceed 50 characters.	Requested widget name is invalid. Check the widget name length.
404 Not Found	ADC_DASH_0001	Dashboard not available for given name.	Requested dashboard is not found in the database. Check if the dashboard is deleted.
404 Not Found	ADC_DASH_2507	Class Management Exception : Widget detail not available in db for given name.	Check if the widget is deleted or if the widget is available in a different dashboard.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible remediation
403 Forbidden	ADC_DASH_0005	Permission not available for dashboard.	Check if the user has Read or Read/Write access for the requested dashboard.

Sample Request/Response

Use Case

Get the details of the class management widget with name classWgt under the dashboard test.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/dashboard-widget`

Sample Request

```
NA
```

Sample Response


```
{
  "response": {
    "name": "classWgt",
    "subsystem": "adc",
    "userDefined": true,
    "applicableFor": null,
    "disabled": false,
    "size": {
      "width": 634,
      "height": 69
    },
    "keywords": [

    ],
    "componentName": "classWgt",
    "description": null,
    "addedGroups": [
      "classWgt/g1"
    ],
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
"removedGroups": [


],
"addedActions": [
  {
    "groupName": "g1",
    "action": "view"
  }
],
"removedActions": [


],
"settings": {
  "actionlist": {
    "g1": {
      "view": {
        "action": "ViewClass",
        "hostProfile": "",
        "actionGroupName": "g1",
        "className": "test_data_group/gs-f5-pe115.apvxlab.com/F5",
        "classId": "lc:@5f36c1567620c40e97e30350:@test_data_group:@Common",
        "deviceId": "gs-f5-pe115.apvxlab.com(5f36c1567620c40e97e30350)",
        "stringValueAssociationCheck": false,
        "selectedStringValueObject": null,
        "availableStringValueObject": null,
        "deviceAssociationCheck": false,
        "selectedDeviceObject": null,
        "availableDeviceObject": null,
        "defaultExecuted": false,
        "msg": "",
        "isRuntimeCheck": true
      }
    }
  },
  "settings": [
    {
      "name": "classWgt",
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
      "parent": "",
      "hierarchy": "",
      "level": "group",
      "nodeLevel": 0,
      "actions": [

      ],
      "children": [

      ],
      "actionStatus": null,
      "classActionName": null,
      "id": "classWgt"
    },
    {
      "name": "g1",
      "parent": "classWgt",
      "hierarchy": "",
      "level": "group",
      "nodeLevel": 1,
      "actions": [
        "view"
      ],
      "children": [



      ],
      "actionStatus": null,
      "classActionName": null,
      "id": "g1"
    }
  ]
},
"classLog": null,
"viaImport": false,
"StdCustomFlag": "ClassManagement",
"_id": "5f3f97be25ac8a2582b0b2f2"
```


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

```
 },
 "message": "Widget details retrieved successfully.",
 "appStatusCode": null,
 "tags": null,
 "headers": null
}
```


Reference

Understanding the sample URL:

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete a Widget

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Delete a Widget

This API deletes any widget that is created by the AppViewX user.

Before you begin

Before attempting to delete a widget with this API, user must ensure any one of the below ACF permissions is available,

- **General** > **Dashboard** > **Create / Delete**
- **ADC** > **Dashboard** > **Application widget** > **Add / Delete**
- **ADC** > **Dashboard** > **Class management widget** > **Add / Delete**
- **ADC** > **Dashboard** > **Device heat map widget** > **Add / Delete**
- **ADC** > **Dashboard** > **Script execution widget** > **Add / Delete**
- **ADC** > **Dashboard** > **Traffic grid widget** > **Add / Delete**
- **ADC** > **Dashboard** > **Traffic statistics widget** > **Add / Delete**

This can be checked under **Accounts** > **Role** > **Authorized Functions**.

Request Structure

Endpoint:	`/dashboard-widget`
Type:	`DELETE`
Sample URL:	`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/dashboard-widget?gwsource=external` To understand the elements of the sample URL, click here.
Headers:	
Content-Type:	application/json

Input Parameter

Name	Description
sessionId	(Mandatory) Specifies the session ID of the user session received after login.
Header	**Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Input Parameter (continued)

Name	Description
	Constraints: The session ID is used when username and password are not provided.
username *Header*	(Mandatory) Specifies the login username used to access the AppViewX. **Type**: String **Constraints**: The username, along with the password, is used when the session ID is not provided.
password *Header*	(Mandatory) Specifies the login password used to access the AppViewX. **Type**: String **Constraints**: The password, along with the username, is used when the session ID is not provided.
Content-Type *Header*	(Mandatory) Specifies the format of the data in the payload. **Type**: String **Constraints**: The value of this param must be ‘application/json’.
gwsource *Query*	(Mandatory) Specifies the source from which the request is triggered. The options are: • web • external **Type**: String
dashboardName *Query*	(Mandatory) Name of the dashboard. **Constraints**: Allowed special characters : '_', '-', ':', '.', '|' (Should not be empty and not start with special characters). **Type**: String
widgetName *Query*	(Mandatory) Name of the class management widget. **Constraints**: Allowed special characters : ‘-’ ,’_’ , ‘.’ , ‘:’ , ’|’ **Type**: String

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Response Structure

- **Status Code:** 200 OK
- **Message:** Successful
- **Headers:**
  - **Content-Type:** application/json

Name	Description
response *String*	Success message or failure description in case of error.
message *String*	Success message or failure description in case of error.
appStatusCode *String*	Application specific status code for the response. Will be non-null for failure response.
tags *String*	More info in case of failure response.

Status Codes

HTTP Status Code	appStatusCode	Message	Possible Remediation
200 OK	-	Widget deleted successfully.	-
400 Bad Request	ADC_DASH_8791	Dashboard name is mandatory.	Check and ensure if a valid value is given in the query param field - dashboardName.
400 Bad Request	ADC_DASH_8792	Widget name is mandatory.	Check and ensure if a valid value is given in query param field - widgetName
404 Not Found	ADC_DASH_0045	Dashboard not available for given name.	Requested dashboard not available in database. Check if the dashboard is deleted.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

HTTP Status Code	appStatusCode	Message	Possible Remediation
404 Not Found	ADC_DASH_3010	Widget not found.	Check if the widget is deleted or if the widget is available in a different dashboard.
403 Forbidden	ADC_DASH_0048	Permission not available for dashboard.	Check if the user has Read Write access for the requested dashboard.

Sample Request/Response

Use Case

Delete a widget named widget_1 from the dashboard dashboard_1.

Request URL

`https://<IP/HostName/TenantName>:<GWPORT>/avxapi/`dashboard-widget?gwsource=external

Sample Request

```
NA
```

Sample Response

```
{
  "response": "Widget deleted successfully.",
  "message": "Widget deleted successfully.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Reference

Understanding the sample URL:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **IP/HostName/TenantName**: Replace with the actual IP address, hostname, or tenant name based on the specific configuration in AppViewX.
  - **IP**: A unique identifier assigned to each device connected to a computer network that uses the Internet Protocol for communication

    The IP address will be included in the endpoint URL for an on-prem deployment.
  - **HostName**: A human-readable label assigned to a device (host) on a network

    The hostname will be included in the endpoint URL for an on-prem deployment.
  - **TenantName**: An identifier label for a tenant given to indicate which tenant's data the API request will access/modify

    The tenant name will be included in the endpoint URL for a SaaS deployment.
- **GWPORT**: AppViewX gateway port

  A gateway port refers to a network port through which data is sent and received to communicate with a gateway in an on-prem deployment.

  Example: **31443**
- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **execute-hook**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- Delete Device

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Chapter 4: ADC+ Automation Workflow Guide

Introduction

Overview

This guide describes AppViewX's ADC predefined workflows to automate and self-service deployments and helps you to integrate with third-party, best-of-breed, and open-source network services like those provided by application delivery controllers, security devices, certificate authorities, DNS servers, routers/switches, and more.

Application Delivery Automation

As your application delivery infrastructure becomes progressively more complex, enterprises must turn to network infrastructure management and automation solutions to ensure successful deployments.

ADC offers predefined workflows to automate and self-service deployments and helps you to integrate with third-party, best-of-breed, and open-source network services like those provided by application delivery controllers, security devices, certificate authorities, DNS servers, routers/switches, and more.

The workflows can integrate with other best-in-class IT service management solutions to govern and record service requests in the larger automated workflow. It can be leveraged to help users move faster, eliminate errors, and reduce costs.

Self-service and Automate Application Delivery services

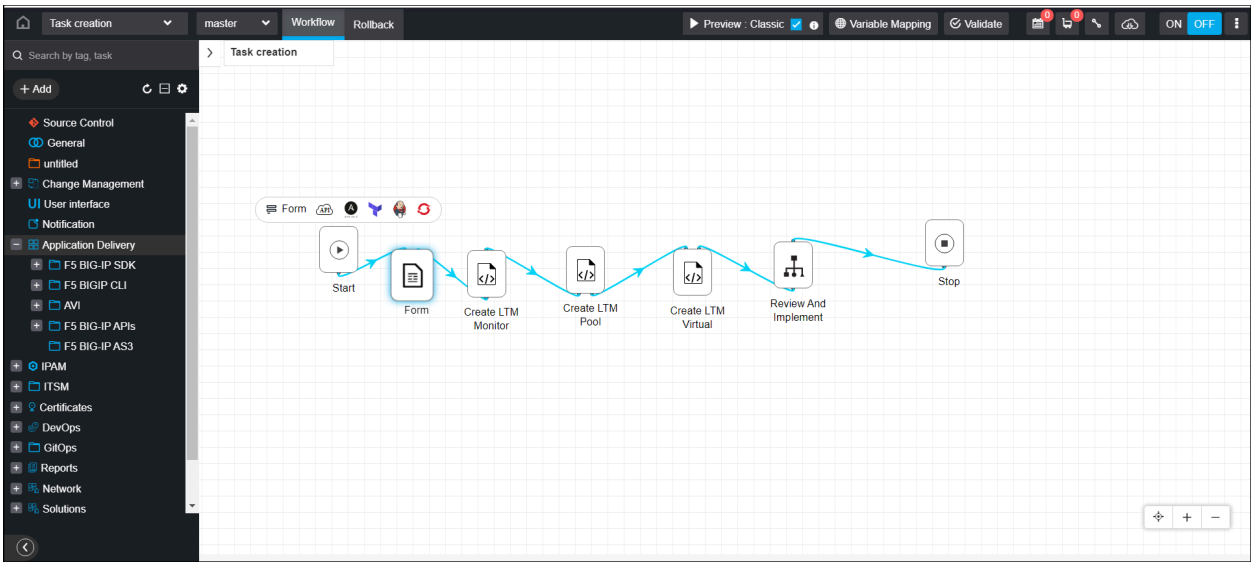
AppViewX is a true multi-vendor platform for managing the ever-evolving complex enterprise infrastructure. As your application delivery infrastructure becomes progressively more complex, enterprises must turn to network infrastructure management and automation solutions to ensure successful deployments. AppViewX offers a suite of tools for organizations striving to become true digital enterprises.

The AppViewX platform will help you automate third-party, best-of-breed, and open-source network services like those provided by application delivery controllers, security devices, certificate authorities, DNS servers, routers/switches, and more.

The AppViewX platform integrates with other best-in-class IT service management solutions to govern and record service requests in the larger automated workflow. It can be leveraged to help users to move

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

faster, eliminate errors, and reduce costs, making it the best choice for enterprises working to adapt to new technologies, processes, and application delivery expectations.



Accessing ADC Predefined Workflow

AppViewX offers Visibility into the One-touch Automation solution offerings to address network and application service requests. Automation solution offerings are available in the **View/Run Workflows** catalog page.

To access ADC+ Predefined workflow,

1. Log into the AppViewX application with valid credentials.
2. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**.
   The ADC+ predefined workflows are listed. The automation offerings (workflows) are controlled by AppViewX's modern RBAC control feature, thus providing individuals groups to maintain and access their own workflows.

   On the **View/Run Workflows** page, you can:

   • View the workflows grouped under the categories.

   • Edit the workflow by selecting the **Edit** option from ⋮.
   • Edit the layout.

   • Refresh the **View/Run Workflows** by clicking the ↻.
   • Sort the workflows by ascending, descending, or user preference order.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Run the workflows.
- Schedule to run a workflow.

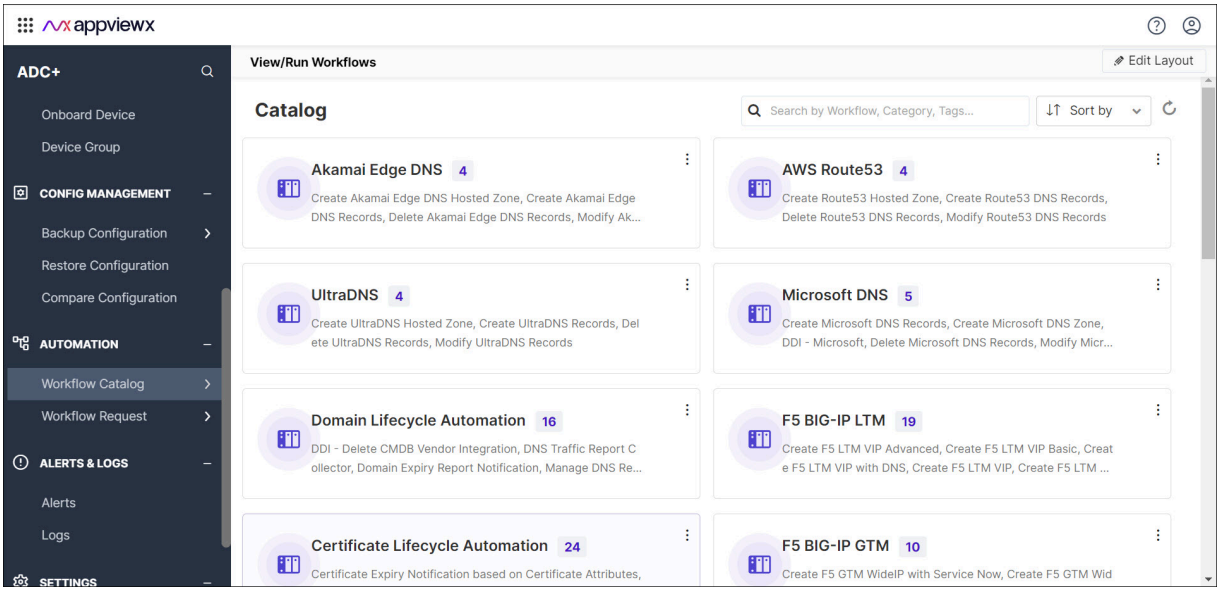
Viewing Workflows (Application Provisioning Catalog)

All the ADC Predefined workflows can be viewed in the Workflow Catalog page.

To view the workflows,

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**.
   The workflows are displayed within their respective categories.
2. Select a workflow category of your choice.
   A popup window opens, displaying the workflows organized within the selected category.

> **Note:**
> You can scroll down to see all the workflows within this category. Additionally, you can use the search function to locate specific workflows within this window.



F5 BIG-IP LTM

Create F5 LTM VIP – iRule

This workflow generates the F5 LTM iRule.

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM VIP – Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Create F5 LTM VIP iRule :: FormBuilder** page is displayed.
5. Enter/select **Create LTM IRule** details.

Create LTM IRule Section - Field and Description Table

Field	Description
***Datacenter**	Displays the datacenter list for devices created in the Device Inventory. Select the relevant datacenter from the drop-down menu to associate with the device for this request. If a device has been created in the Device Inventory without a specified datacenter, select **None** as the datacenter.
***Device Name**	Displays a list of devices linked to the selected datacenter. If the datacenter is set as **None**, it lists devices created without specific datacenter details. Select the desired device from the available drop-down options.
***IRule Name**	Enter the name assigned to an iRule in F5 Networks' BIG-IP LTM.
***IRule Content**	Enter the content or script written within an iRule.

6. Click **Submit**.
   The **Confirmation** dialog box is displayed.

> **Note:**
> - If you want to save this form and edit it later, click **Save Draft**, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click **Cancel**.

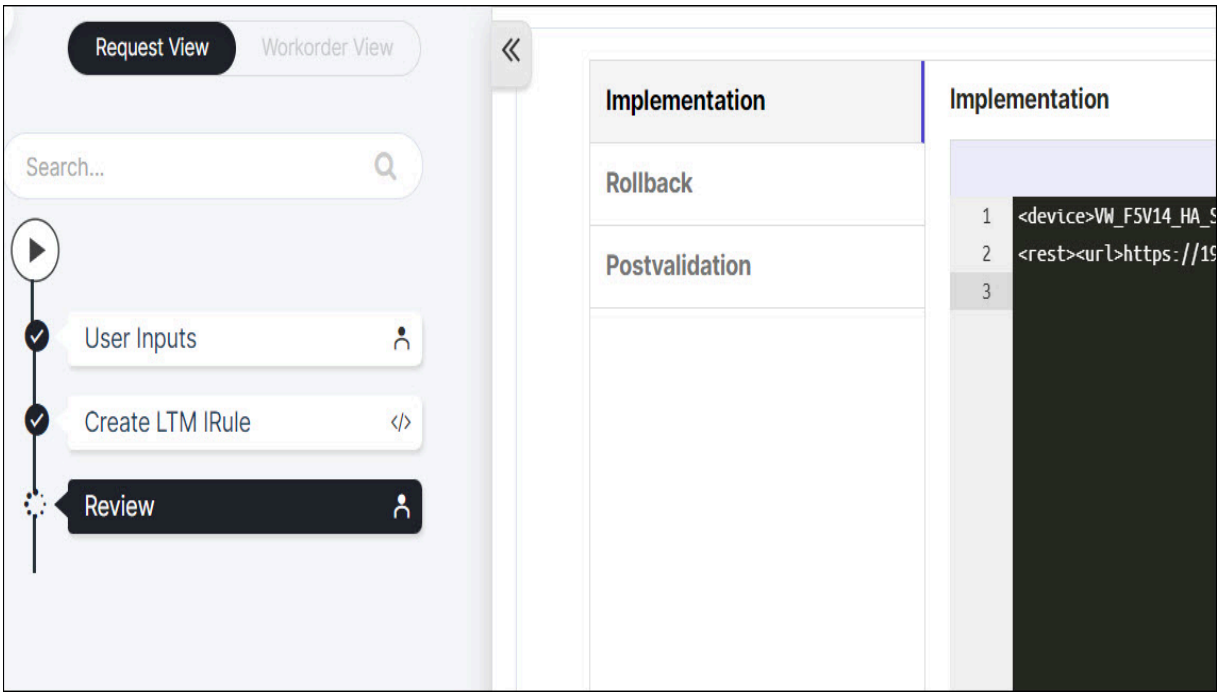
7. Click **OK** to submit the form.
   The validation starts automatically and reaches the **Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



9. After the review, click **Submit**.
   The Confirmation dialog box is displayed.

Note:
To stop running the workflow creation, click **Cancel**.

10. Click **OK** to continue the workflow creation.

Note:
It takes a while to complete the request.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The workflow is created.

> **Note:**
> The stages of the request are shown on the left of the screen. To view a specific stage of the request, click on the corresponding stage.

Create F5 LTM VIP with DNS

This workflow creates App Owner Centric VIP and integrates with DNS - Infoblox/Bluecat for A record creation.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover over the **Create F5 LTM VIP with DNS** workflow.
4. Click the Run 

button.
   The **Request > Create F5 LTM VIP with DNS :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be added. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter/select the **DNS Record Details**.

DNS Record Details Section - Field and Description Table

Field	Description
Do you want to create A record?	The default option is **No**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	To create A record in DNS select **Yes** and enter the **Vendor**, **DNS Device**, **Network View**, and **Subnet** details.

7. Enter/select the **Virtual Details**.

Virtual Details Section - Field and Description Table

Field	Description
FQDN/Application Name	Enter the FQDN or application name of the virtual server.

Do you need to Enable SSL Certificates?	Select any of the following options to enable SSL certificates. The default option is No-InSecure Site.
	• **No-InSecure Site**
	• **Yes** - Only the servers will have SSL certificates
	  • **No** – The only Client will have an SSL Certificate
	  • **Yes** - Both Client and Servers will have SSL certificates

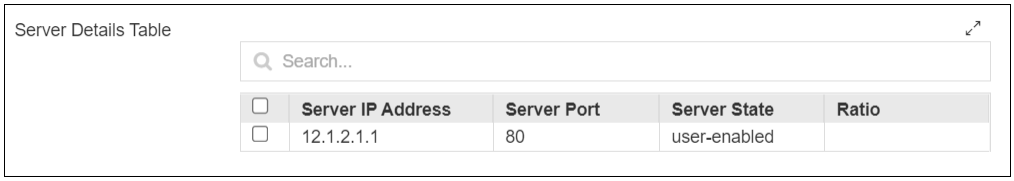
IP Choice	This option is applicable when a DNS record is to be created. Select any of the following IP choices: • **Manual** - allows to give the IP manually. • **Next Available IP from IPAM** - allows to fetch the next available IP from IPAM. To get the IP, click the Fetch Next Available IP and Reserve button.
Destination Address	Enter the destination IP address information for the virtual server.
Service Port	Enter a service port.

8. Enter/select the **Pool and Monitor Details**.

Field descriptions for the Pool and Monitor Details section

Field	Description
Load Balancing Method	The load balancing method is used to select a pool in this WideIP. The default is round-robin. The methods are: • round-robin - the system selects the pools sequentially. • least-connection-node - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • least-connection-member - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • ratio-member - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.
Address	Enter the IP address of the pool.
Service Port	Enter a service port.
State	The current state of the pool members. The states are: • **user-enabled** - when you select this option, the system sends traffic to this pool member regardless of the pool member's state. • **user-disabled** - when this option is selected, the pool member can handle only persistent or active connections.
Pool Members	Enter the IP address of the pool member. And then click the Add  button. Any number of pool numbers can be added to the pool. After adding the pool, you can manage them. 
Monitor / Health Check Type	Select the health monitors that are available to add for the pool: • **HTTP** • **TCP**

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Request Method and URI	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. The default setting is GET /\r\n, which retrieves a default HTML file for a website. To retrieve a specific page from a website, specify a fully qualified path name.
Expected Response	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

9. Enter/select the **Snat Pool Details**.

Snat Pool Details Section - Field and Description Table

Field	Description
Snat Choice	Select the SNAT choice for any connections using this pool. The options are: • **AutoMap** -This option allows you to select a translation address from the available self-IP address. • **SNAT** - This option allows you to select a floating self IP as a translation address. When this option is selected, the Snat Pool field appears. Select the Snat pool from the drop-down option:

10. Enter/select the **Persistence Details**.

Persistence Details Section - Field Description Table

Field	Description
Persistence	This option allows you to use a pre-configured object that automatically enables persistence when you assign the profile to a virtual server. • **None** • **Create New** - create a new persistence by providing persistence details: • **Existing** - select the existing persistence details:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

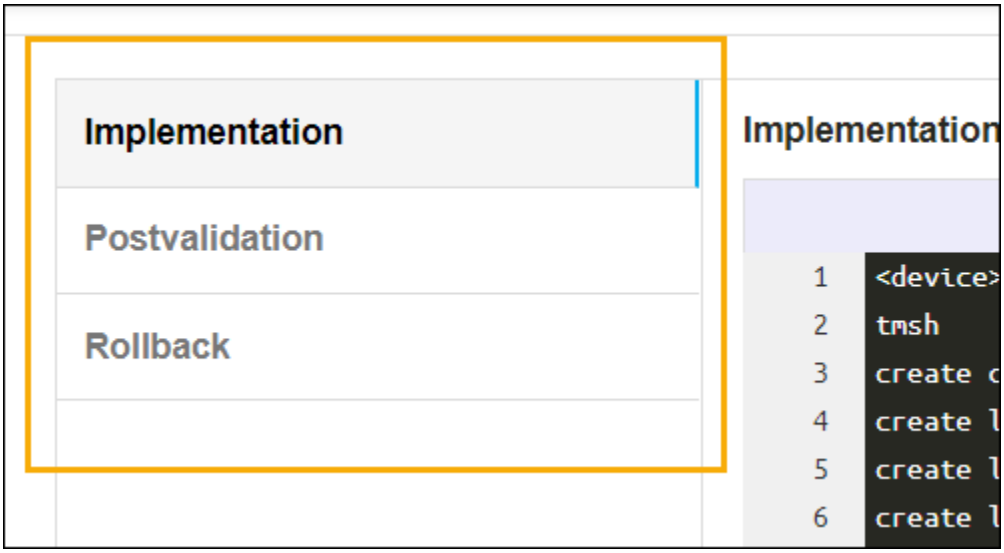
> **Note:**
> If you need to cancel the request, make sure that you unreserve the IP.
> To unreserve the IP, click **Un-reserve IP**.

11. Click **Submit**.
    The Confirmation popup opens.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

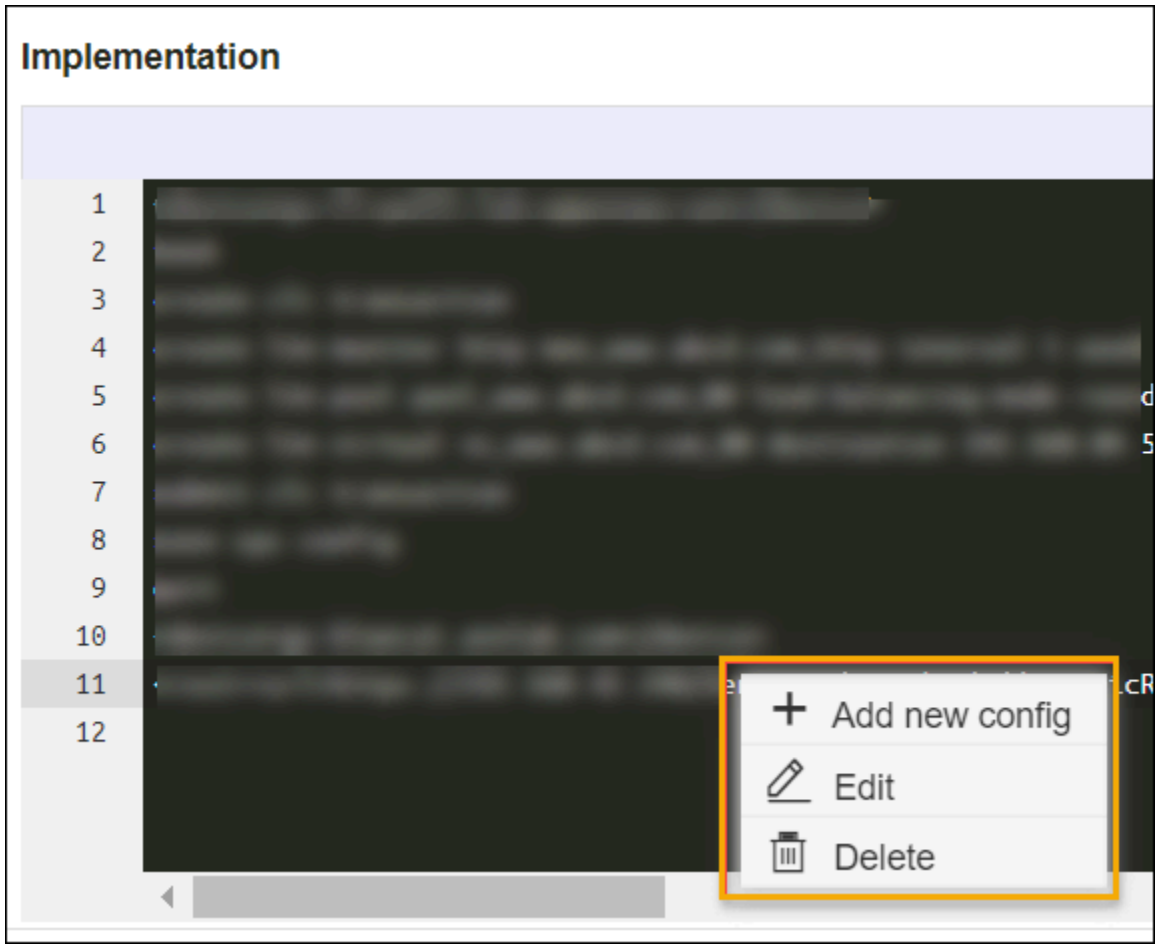
    To submit the form, click **OK**.

    The pre-validation starts automatically and reaches the **Review** stage.
13. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



14. (Optional) If you need to change any data at this stage, you can update by clicking the right mouse button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



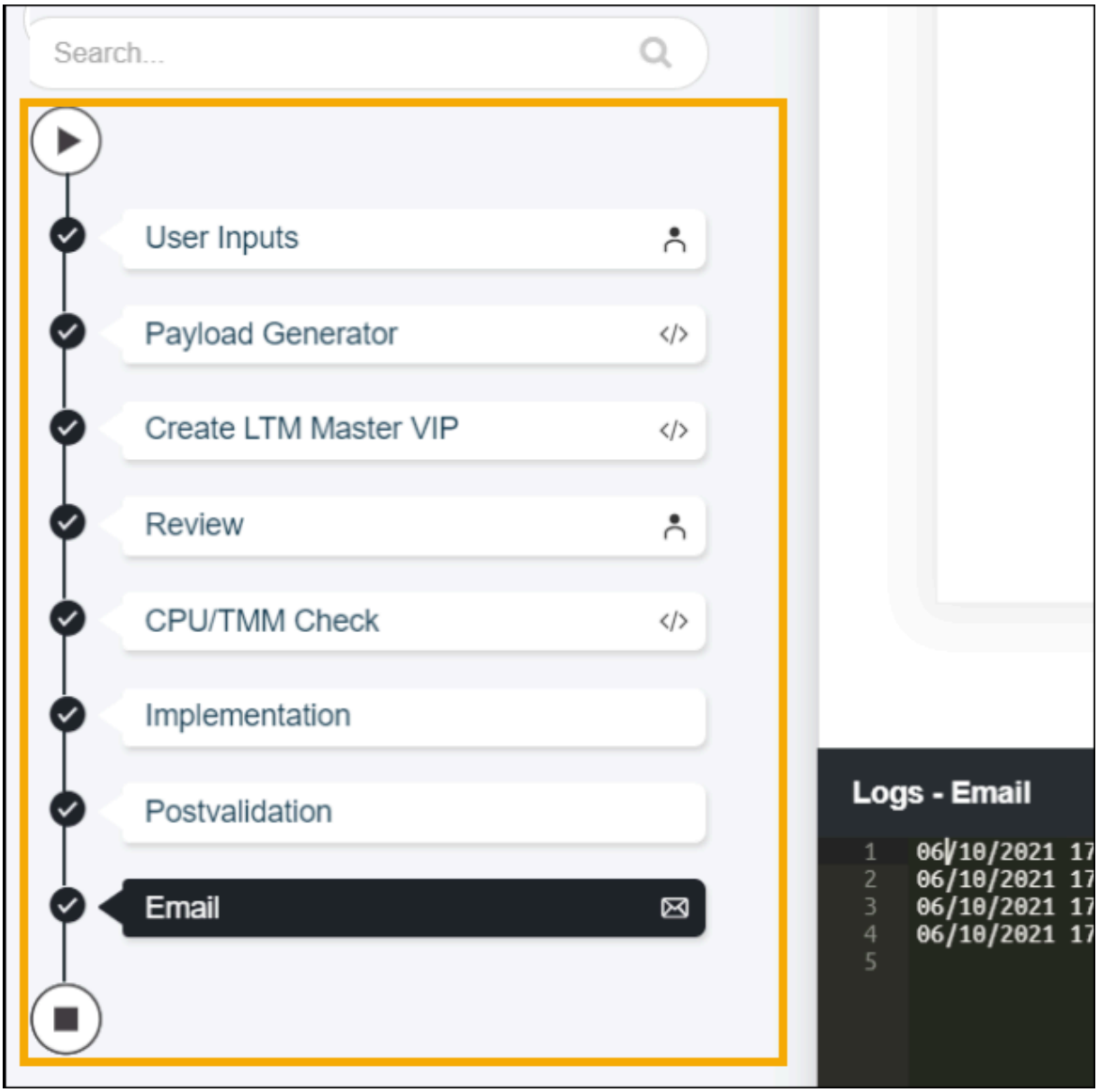
15. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
16. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    After the post-validation is successful, the workflow is created and an email triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Create F5 LTM VIP with Service Now

By running this workflow, you can create App Owner Centric VIP and integrate with ITSM – Service Now, where users can set Time Zone, Start time and End time for the ticket creation.

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover over the **Create F5 LTM VIP with Service Now** workflow.
4. Click the Run  button.

   The **Request > Create F5 LTM VIP with Service Now :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be added. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter/select the **Virtual Details**.

Virtual Details Section - Field and Description Table

Field	Description
Application Name	Enter the FQDN of the virtual server.

Do you need to Enable SSL Certificates?	Select any of the following options to enable SSL certificates. The default option is No-InSecure Site.
	• **No-InSecure Site**
	• **Yes** - Only the servers will have SSL certificates
	  • **No** – The only Client will have an SSL Certificate
	  • **Yes** - Both Client and Servers will have SSL certificates

Destination Address	Enter the destination IP address information for the virtual server.
Service Port	Enter a service port.

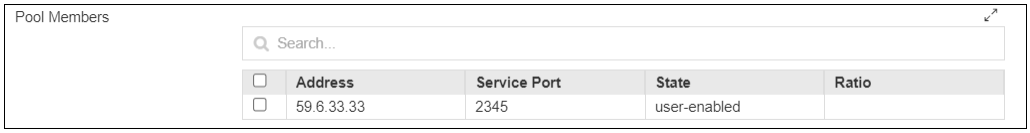
7. Enter/select the **Pool Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Pool Details Section - Field and Description Table

Field	Description
Load Balancing Method	The load balancing method used to select a pool in this WideIP. The default is round-robin.The methods are: • **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.
***Address**	Enter the IP address of the pool.
***Service Port**	Enter a service port.
State	The current state of the pool members. The states are: • user-enabled - when you select this option, the system sends traffic to this pool member regardless of the pool member's state. • user-disabled - when this option is selected, the pool member can handle only persistent or active connections.
Pool Members	Enter the IP address of the pool member. And then click the Add  button. Any number of pool numbers can be added to the pool. After adding the pool, you can manage them.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	

8. Enter/select the **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
Monitor / Health Check Type	Select the health monitors that are available to add for the pool: • **HTTP** • **TCP**
Request Method and URI	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. The default setting is GET /\r\n, which retrieves a default HTML file for a website. To retrieve a specific page from a website, specify a fully qualified path name.
Expected Response	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

9. Enter/select the **Snat Pool Details**.

Snat Pool Details Section - Field and Description Table

Field	Description
Snat Choice	Select the SNAT choice for any connections using this pool. The options are: • **AutoMap** -This option allows you to select a translation address from the available self-IP address. • **SNAT** - This option allows you to select a floating self IP as a translation address. When

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	this option is selected, the Snat Pool field appears. Select the Snat pool from the drop-down option:

10. Enter/select the **Persistence Details**.

Persistence Details Section - Field and Description Table

Field	Description
Persistence	This option allows you to use a pre-configured object that automatically enables persistence when you assign the profile to a virtual server. • **None** • **Create New** - create a new persistence by providing persistence details: • **Existing** - select the existing persistence details:

11. Select the **Change Management** details.

Change Management Section - Field and Description Table

Field	Description
***Do you want to integrate ServiceNow?**	By default, the No option is selected. If you want to integrate with ServiceNow, select Yes and provide the following details:  • **Timezone** – select the timezone from the drop-down list. • ***Start Time** – select the start date and time from the calendar. • ***End Time** – select the end date and time from the calendar.

12. Click **Submit**.
    The Confirmation popup opens.
13. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The form will be saved as **Open request** under **Request** > **My Request**.

OR

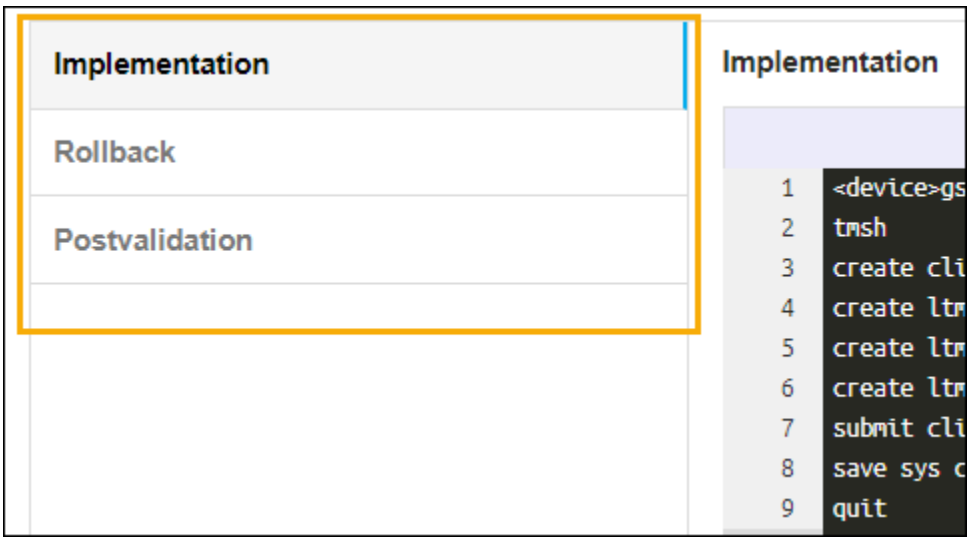
To submit the form, click **OK**.

The validation starts automatically and reaches the **Review** stage.

Note:

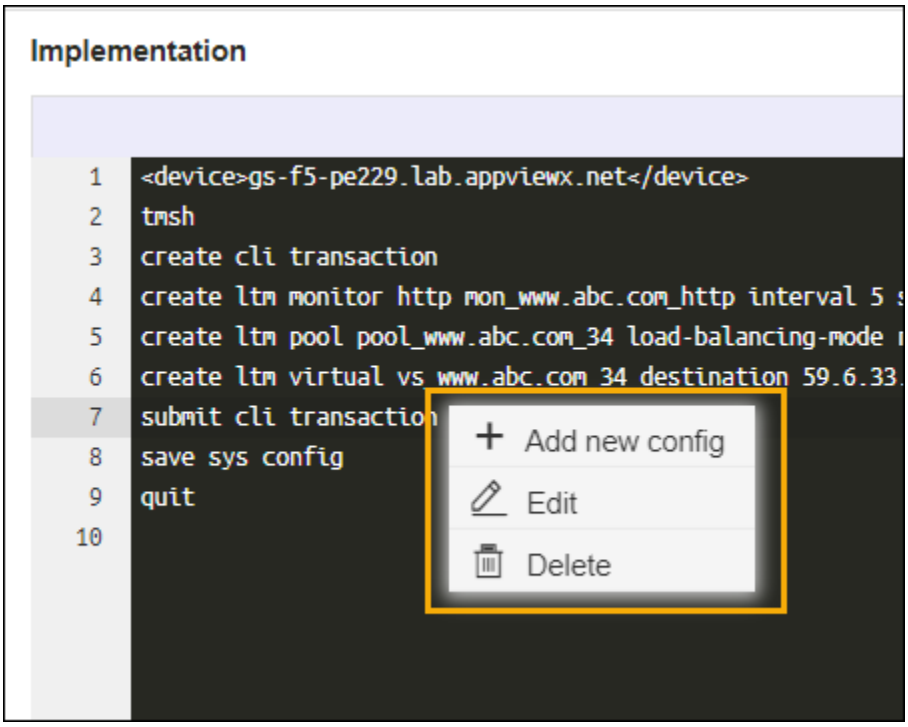
- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

14. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



15. (Optional) If you need to update any data at this stage, you can do so by clicking the right mouse button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



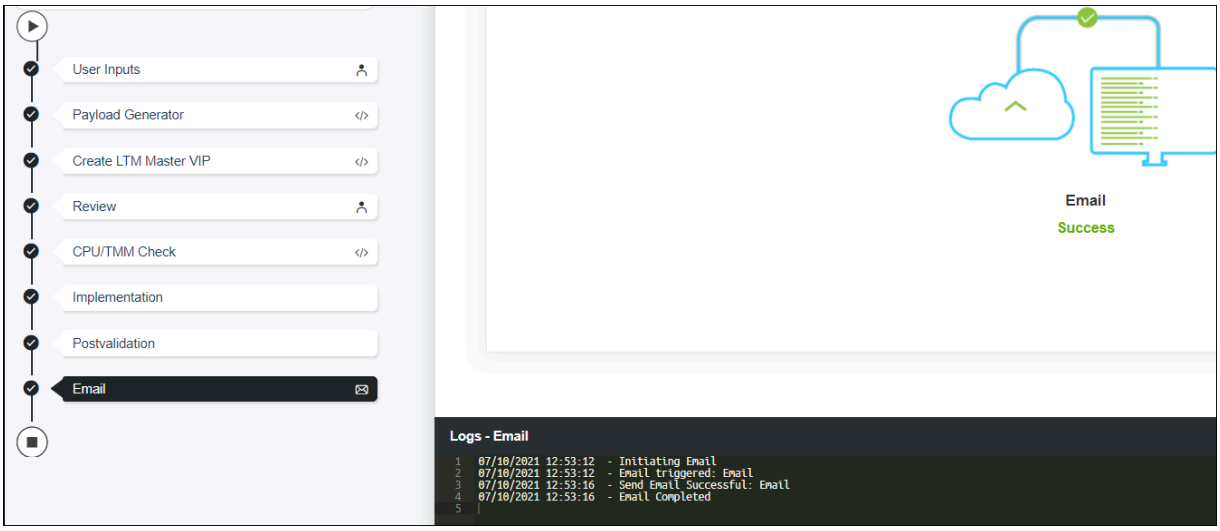
16. After the review, click the **Submit** button.
    The **Confirmation** dialog box is displayed.
17. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    After the post-validation is successful, the workflow is created and an email triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Create F5 LTM VIP Advanced

This workflow creates Extensive VIP with all basic objects in an F5 device and integrates with DNS - Infoblox/Bluecat for A record creation.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM VIP Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 LTM VIP Advanced :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Partition**	Partition in which objects will be created. If you want to create a new partition then click **Create New Partition** from the dropdown list and then provide a partition name.

6. Enter/select the **DNS Record Details**.

DNS Record Details Section - Field and Description Table

Field	Description
Do you want to create A record ?	The default option is **No**. To create A record in DNS select **Yes** and enter the **Vendor**, **DNS Device**, **Network View**, and **Subnet** details.

7. Enter/select the **Virtual Details**.

Virtual Details Section - Field and Description Table

Field	Description
FQDN/ Application Name	Enter the FQDN or application name of the virtual server.
Virtual Server Type	Select the virtual server type.
Redirect VIP	An IP forwarding virtual server accepts traffic that matches the virtual server address and forwards it to the destination IP address that is specified in the request rather than load balancing the traffic to a pool. • **No** (default) • **Yes** - select Yes to redirect VIP.
Destination Address	Enter the destination IP address information for the virtual server.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

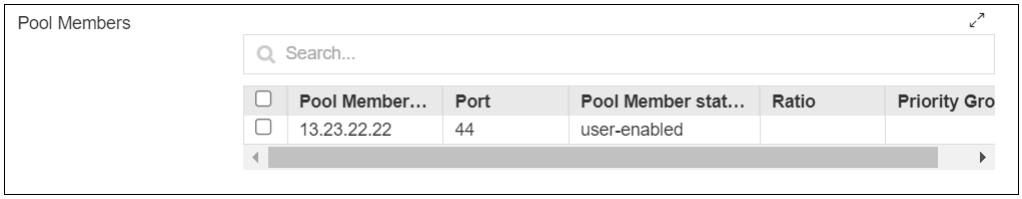
Field	Description
Service Port	Enter a service port.

8. Enter/select the **Pool and Monitor Details**.

Pool and Monitor Details Section - Field and Description Table

Field	Description
Pool	You can create a pool by providing the details for the pool or select an existing pool. Select the desired option: • **Create New** - this is the default option. When you select this option, you need to provide a Load Balancing Method, Address, Service Port, State, Priority Group, Pool Members, and Monitor. • **Existing** - when you select this option, the Select Pool field appears to select the existing pool from the drop-down option:
Load Balancing Method	The load balancing method is used to select a pool in this WideIP. The default is a round-robin. The methods are: • **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.
Address	Enter the IP address of the pool.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Service Port	Enter a service port.
State	The current state of the pool members. The states are: • user-enabled - when you select this option, the system sends traffic to this pool member regardless of the pool member's state. • user-disabled - when this option is selected, the pool member can handle only persistent or active connections.
Priority Group	A number representing the priority group for the pool members. To specify a priority, you must activate priority group usage when you create a new pool or when adding or removing pool members. When activated, the system load balances traffic according to the priority group number assigned to the pool member.
Pool Members	Enter the IP address of the pool member. And then click the Add  button. Any number of pool numbers can be added to the pool. After adding the pool, you can manage them. 
Monitor	An association between a health monitor and an entire pool. Select the desired Monitor option: • **No** - this option does not allow you to monitor the pool. • **Create New** - select this option to create a monitor. Input the Type, Interval, Interval, Timeout, and Monitor Destination. • **Existing** - select this option to choose the existing monitor option.

9. Enter/select the **Persistence Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Persistence Details Section - Field and Description Table

Field	Description
Persistence	This option allows you to use a pre-configured object that automatically enables persistence when you assign the profile to a virtual server. • **None** • **Create New** - create a new persistence by providing persistence details: • **Existing** - select the existing persistence details:
Fallback Persistence	It is a secondary persistence record for each connection. Select the desired following option: • **No** • **Yes**

10. Enter/select the **Snat Pool Details**.

Snat Pool Details Section - Field and Description Table

Field	Description
Snat Pool	Select the SNAT pool for any connections using this pool. The options are: • **Auto-Map**p -This option allows you to select a translation address from the available self-IP address. • **SNAT** - This option allows you to select a floating self IP as a translation address. When this option is selected, the Snat Pool field appears. Select the Snat pool from the drop-down option:

11. Enter/select the **Client SSL Details**.

Client SSL Details Section - Field and Description Table

Field	Description
Client SSL	Select the Client SSL options. The options are: • **No** (default) • **Create New** - This option allows you to upload a new Client SSL certificate. • **Existing** - This option allows you to select a Client SSL certificate from the drop-down option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

12. Enter/select the **Server SSL Details**.

Server SSL Details Section - Field and Description Table

Field	Description
Server SSL	Select the Client SSL options. The options are: • **No** (default) • **Create New** -This option allows you to upload a new Server SSL certificate. • **Existing**- This option allows you to select a Server SSL certificate from the drop-down option:

13. Enter/select the **Irule Details**.

Irule Details - Field and Description Table

Field	Description
IRule	Select the IRule options. The options are: • **No** (default) • **Create New** -This option allows you to upload a new IRule file. • **Existing**- This option allows you to select a IRule file from the drop-down option:

14. Enter/select the **Other profile Details**.
15. Click **Submit**.

    The **Confirmation** dialog box is displayed.
16. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

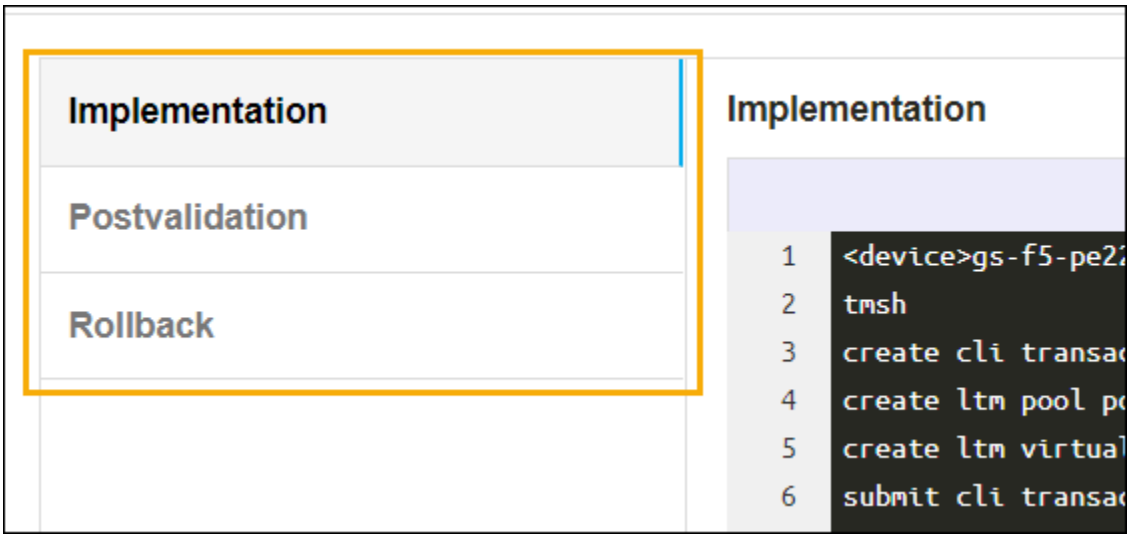
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

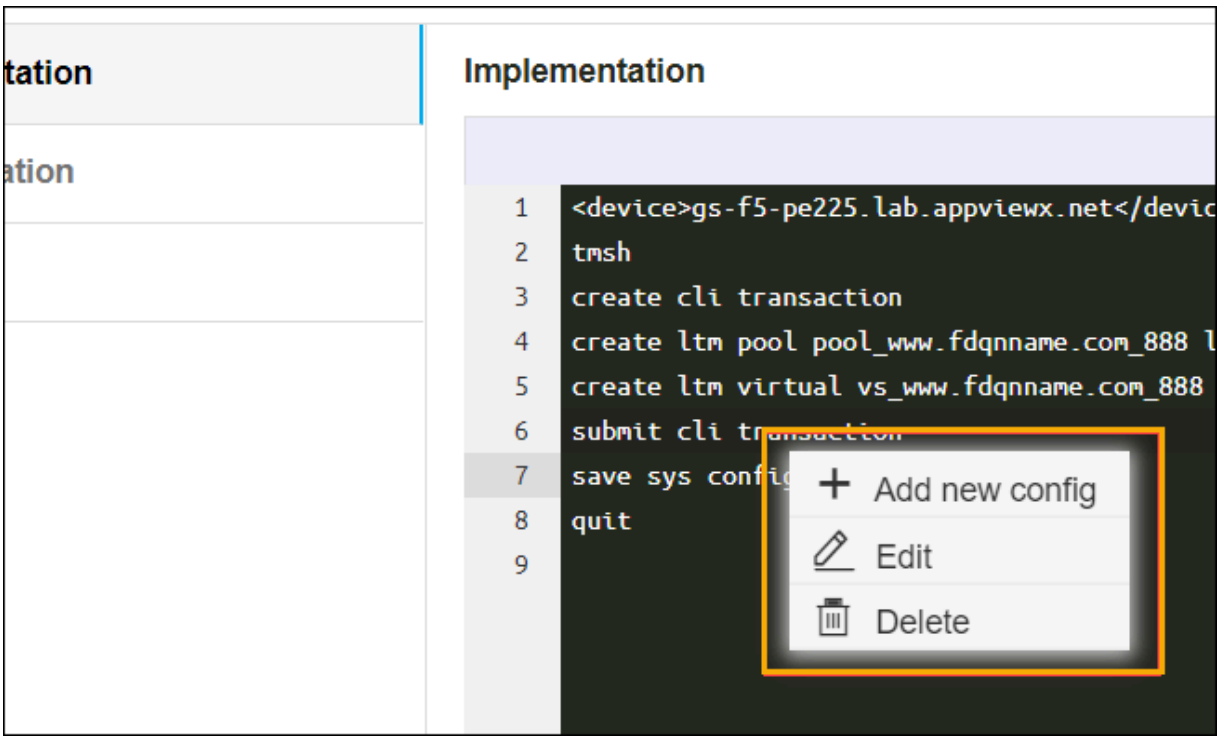
    To submit the form, click **OK**.

    The pre-validation starts automatically and reaches the **Review** stage.
17. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



18. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.



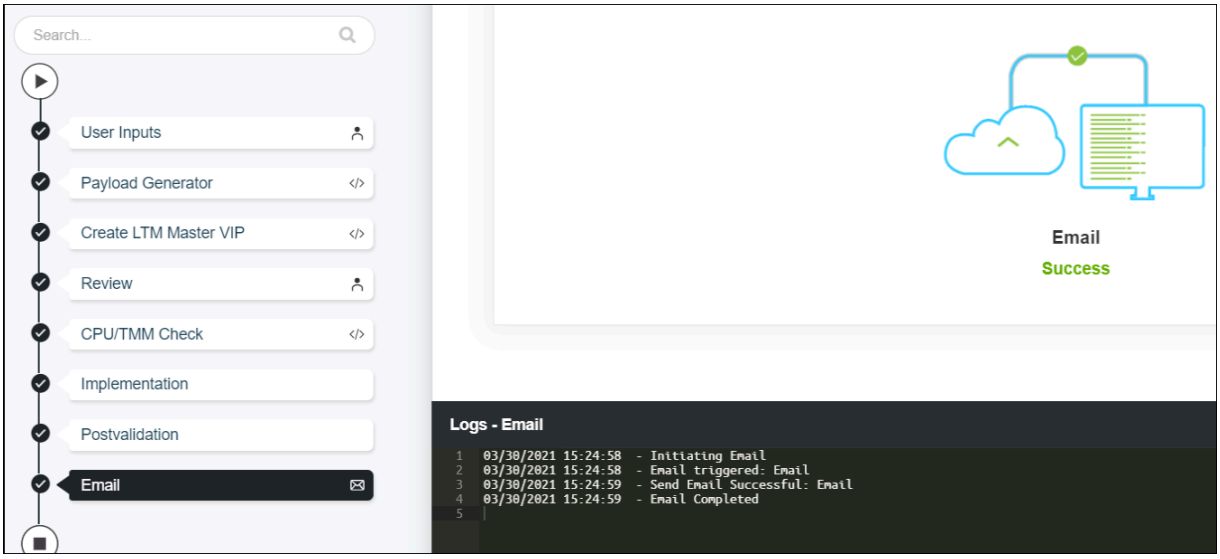
19. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
20. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

It takes a while to complete the post-validation.

After the post-validation is successful, the workflow is created and an email triggered to all the configured email IDs.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Create F5 LTM VIP with Partition

This workflow creates Extensive VIP with all basic objects in a F5 device on a selected Partition and integrates with DNS - Infoblox/Bluecat for A record creation.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM VIP with Partition** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 LTM VIP with Partition :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Partition**	Partition in which objects will be created. If you want to create a new partition then click **Create New Partition** from the dropdown list and then provide a partition name.

6. Enter/select the **Virtual Details**.

Virtual Details Section - Field and Description Table

Field	Description
***Application Name**	Enter the FQDN of the virtual server.

***Do you need to Enable SSL Certificates?**	Select any of the following options to enable SSL certificates. The default option is No-InSecure Site.
	• **No-InSecure Site**
	• **Yes** - Only the servers will have SSL certificates
	  • **No** – The only Client will have an SSL Certificate
	  • **Yes** - Both Client and Servers will have SSL certificates

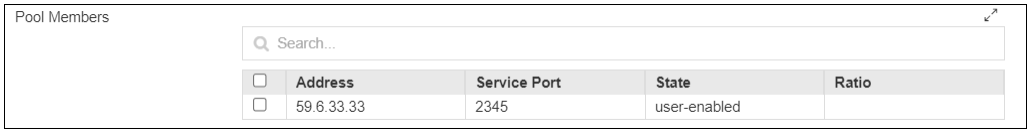
***Destination Address**	Enter the destination IP address information for the virtual server.
***Service Port**	Enter a service port.

7. Enter/select the **Pool Details**.

Pool Details Section - Field and Description Table

Field	Description
Load Balancing Method	The load balancing method is used to select a pool in this WideIP. The default is a round-robin. The methods are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.
***Address**	Enter the IP address of the pool.
***Service Port**	Enter a service port.
State	The current state of the pool members. The states are: • user-enabled - when you select this option, the system sends traffic to this pool member regardless of the pool member's state. • user-disabled - when this option is selected, the pool member can handle only persistent or active connections.
Pool Members	Enter the IP address of the pool member. And then click the Add  button. Any number of pool numbers can be added to the pool. After adding the pool, you can manage them. 

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

8. Enter/select the **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
Monitor / Health Check Type	Select the health monitors that are available to add for the pool: • **HTTP** • **TCP**
Request Method and URI	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. The default setting is GET /\r\n, which retrieves a default HTML file for a website. To retrieve a specific page from a website, specify a fully qualified path name
Expected Response	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

9. Enter/select the **Snat Pool Details**.

Snat Pool Details Section - Field and Description Table

Field	Description
Snat Choice	Select the SNAT choice for any connections using this pool. The options are: • **AutoMap** -This option allows you to select a translation address from the available self-IP address. • **SNAT** - This option allows you to select a floating self IP as a translation address. When this option is selected, the Snat Pool field appears. Select the Snat pool from the drop-down option:

10. Enter/select the **Persistence Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Persistence Details Section - Field and Description Table

Field	Description
Persistence	This option allows you to use a pre-configured object that automatically enables persistence when you assign the profile to a virtual server. The options for persistence are: • **None** • **Create New** - create a new persistence by providing persistence details: • **Existing** - select the existing persistence details:

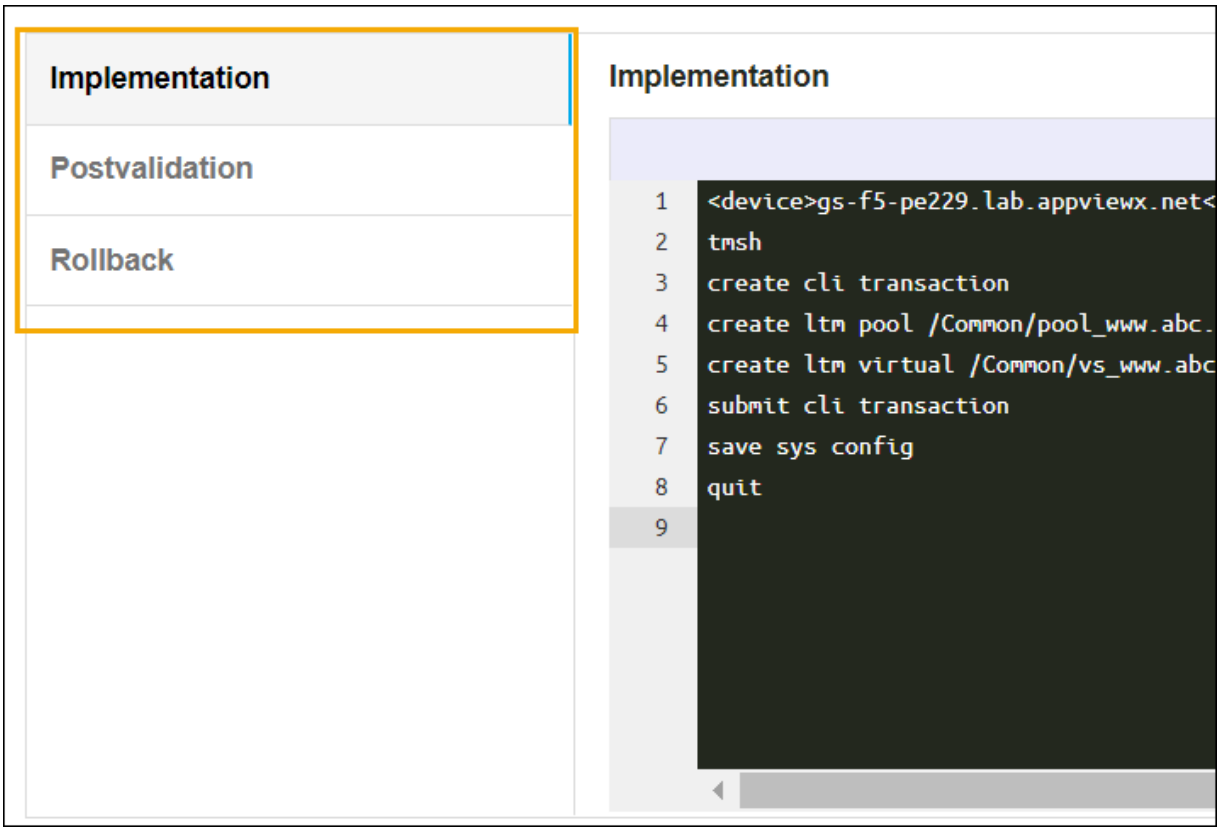
11. Click **Submit**.
    The Confirmation popup opens.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

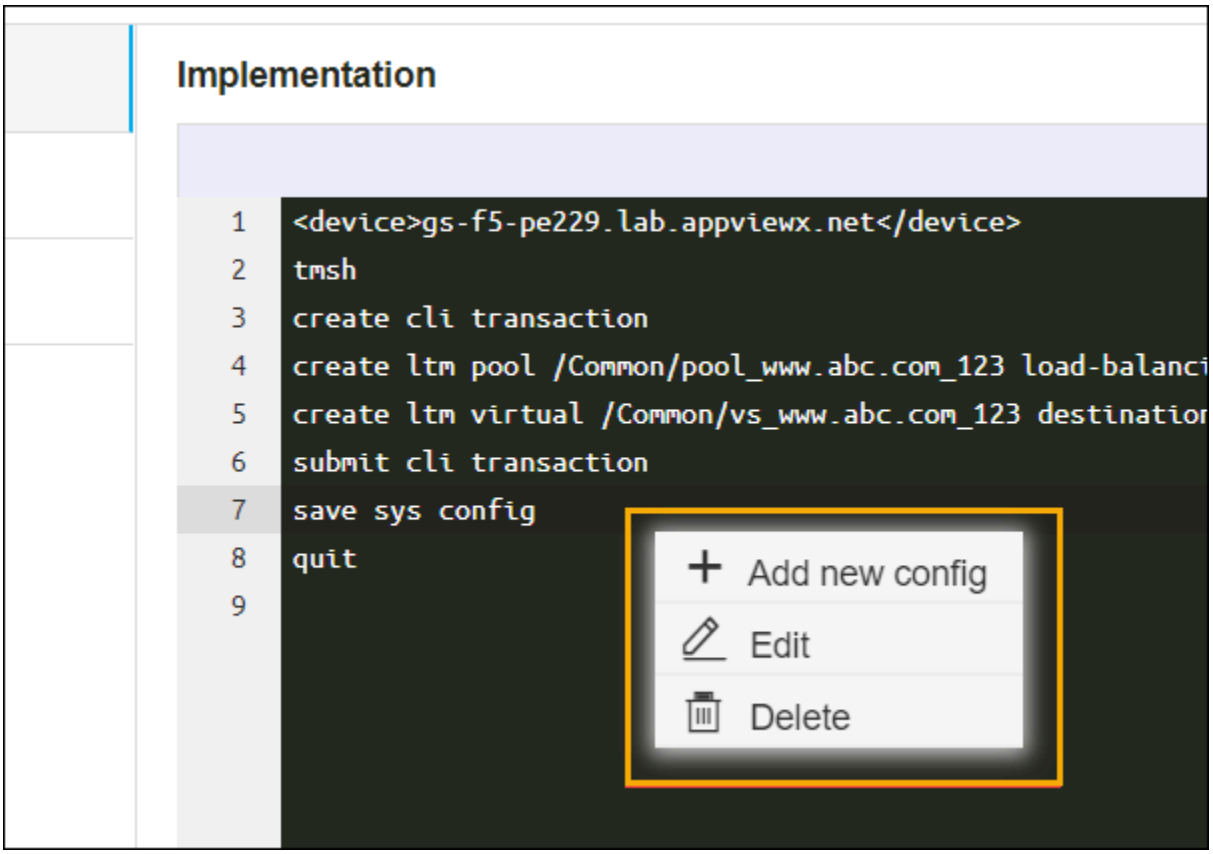
    The pre-validation starts automatically and reaches the **Review** stage.
13. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



14. (Optional) If you need to change any data at this stage, you can update by clicking the right mouse button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



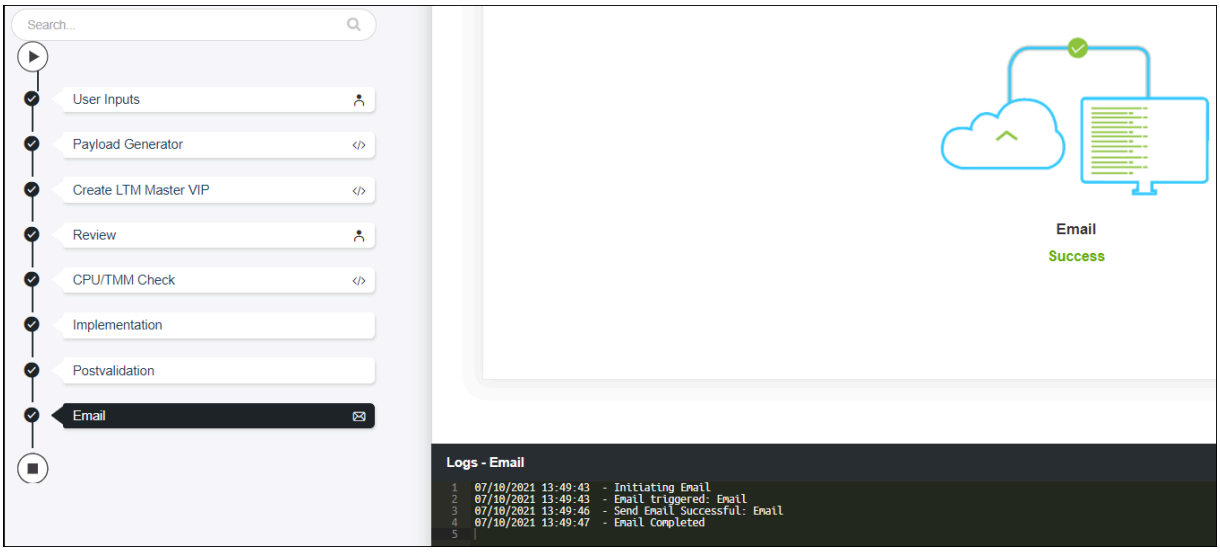
15. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
16. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    After the post-validation is successful, the workflow is created and an email triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

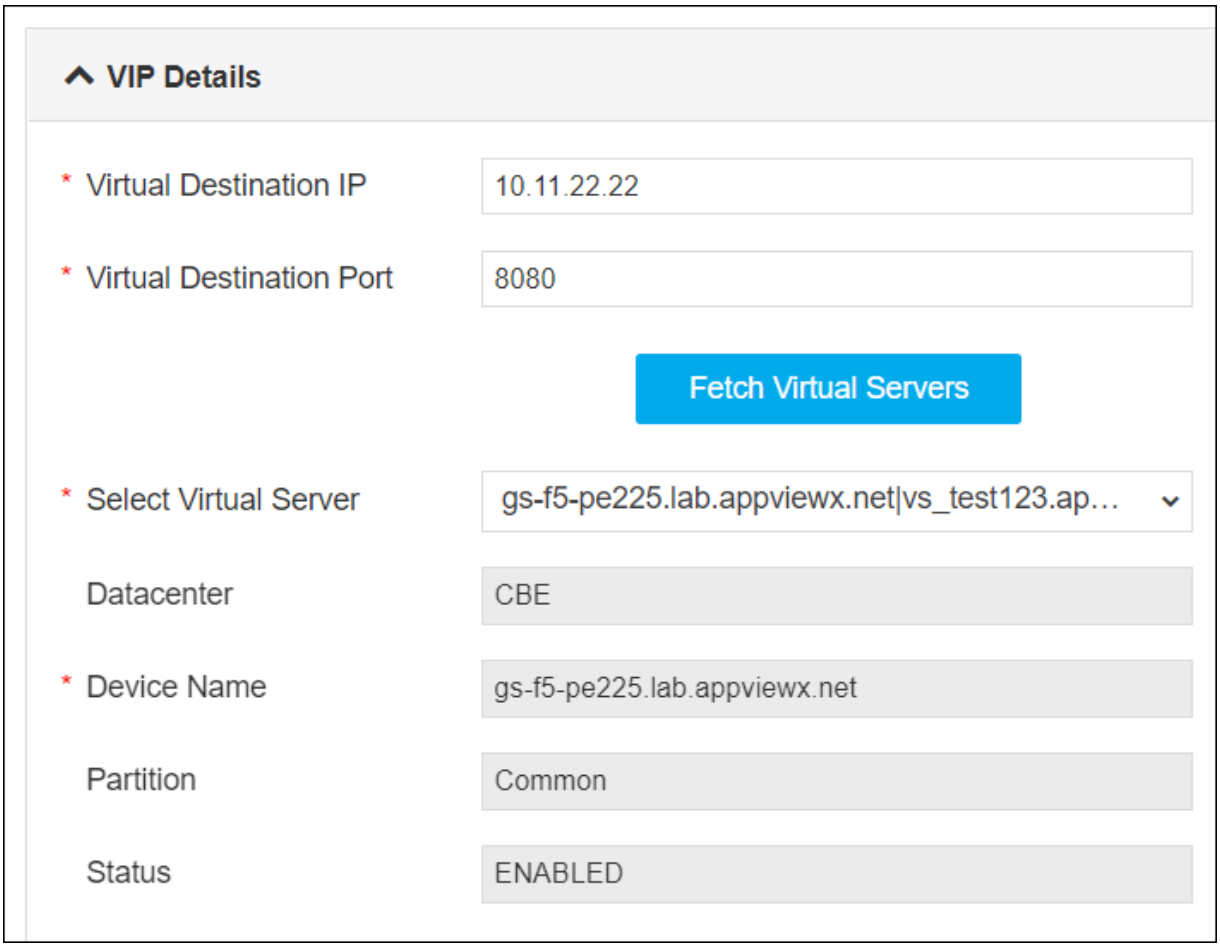
Delete F5 LTM VIP - Basic

This workflow deletes VIP along with its associated objects (Pool, Monitor, IRule, Profiles, Persistence, Snatpool, Policy) by searching the VIP using the Destination IP and Port across Managed Devices in AppViewX.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 LTM VIP - Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete F5 LTM VIP - Basic :: FormBuilder** page is displayed.
5. Enter the **Virtual Destination IP** and **Virtual Destination Port**, and then click **Fetch Virtual Servers**.
   The virtual server details are fetched for the given **Virtual Destination IP** and **Virtual Destination Port** and auto-populated in the rest of the fields.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Click **Submit**.

   The Confirmation dialog box is displayed.

7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

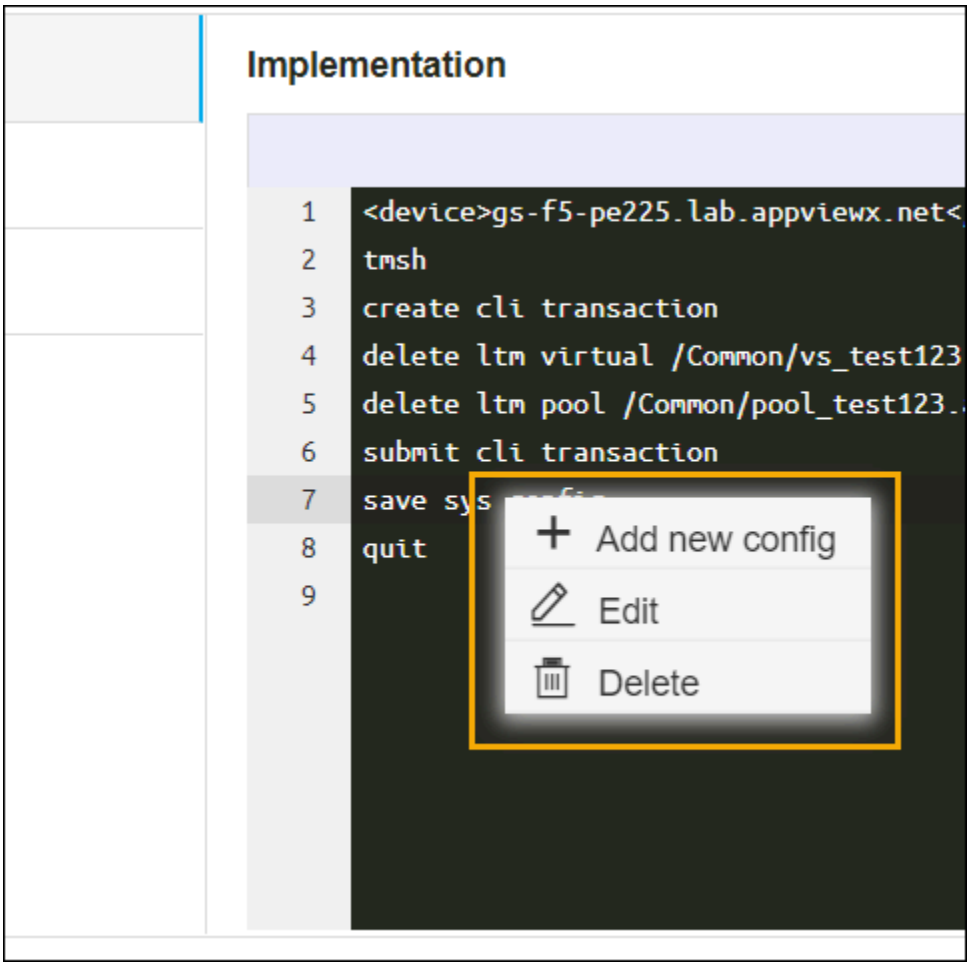
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



9. (Optional) If you need to change any data at this stage, you can update by clicking the right mouse button.



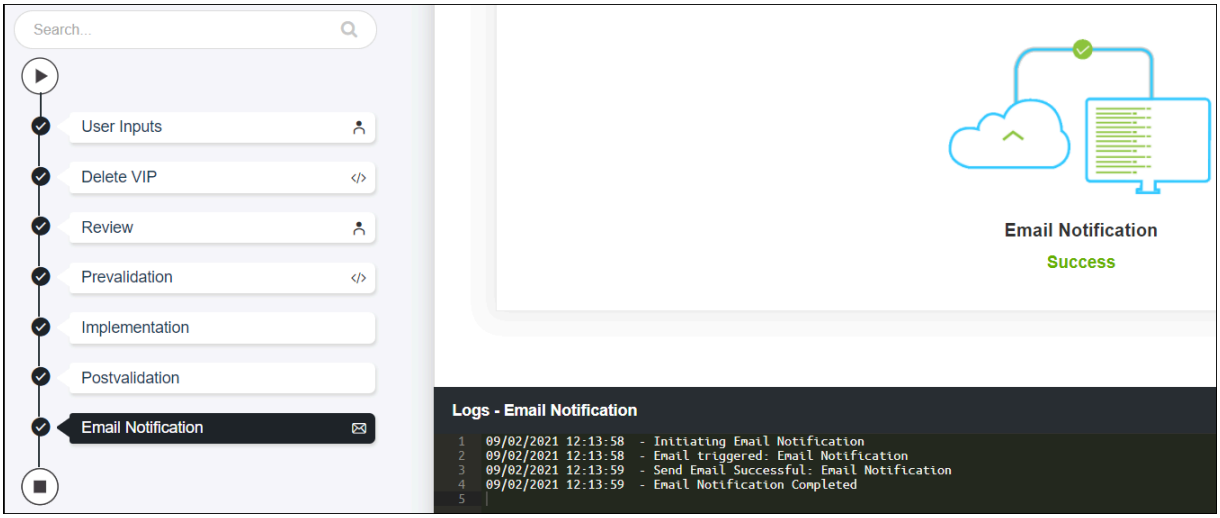
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
11. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    After the post-validation is successful, the workflow is created and an email triggered to all the configured email IDs.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Delete F5 LTM VIP - Advanced

This workflow deletes VIP along with its associated objects (Pool, Monitor, IRule, Profiles, Persistence, Snatpool, Policy) and multi-select the associated object types to consider for deletion.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 LTM VIP - Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.

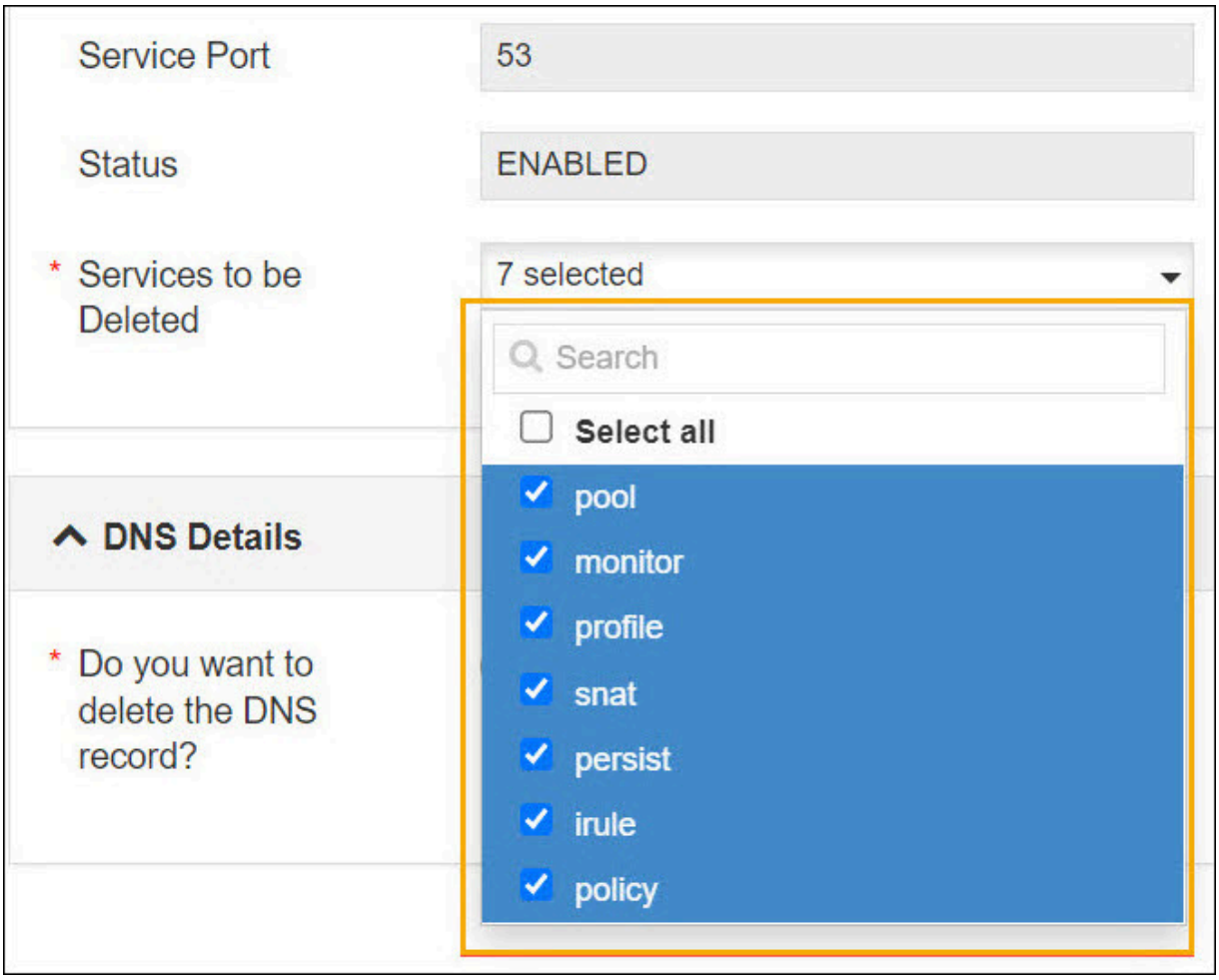
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click Run.

   The **Request > Delete F5 LTM VIP - Advanced :: FormBuilder** page is displayed.

5. From their respective dropdown lists, select the **Datacenter**, the **Device Name**, the **Partition**, and the **Virtual Server**.

   The services that are associated with the selected device are populated in the **Services to be Deleted** dropdown list.



6. Select all the services of the device or select/clear only the services of the devices that are to be deleted.
7. (Optional) Select the **DNS details** of the device, if DNS details of the device are required to be deleted.
8. Click **Submit**.

   The Confirmation dialog box is displayed.

9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

   The form will be saved as **Open request** under **Request** > **My Request**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

OR

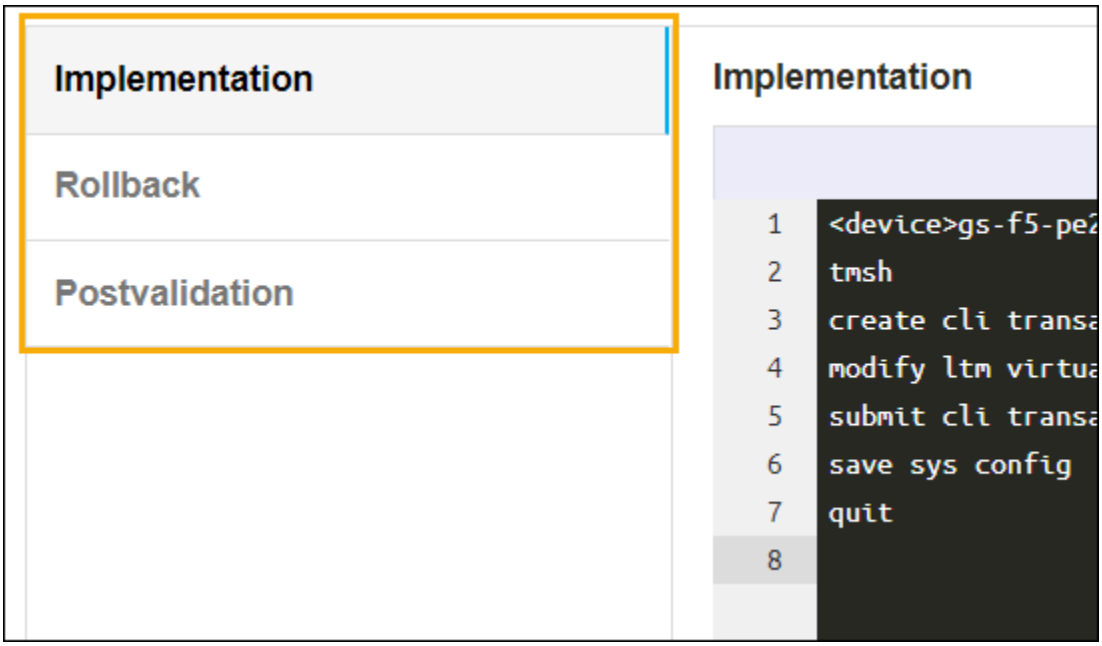
To submit the form, click **OK**.

The pre-validation starts automatically and reaches the **Disable Review** stage.

Note:

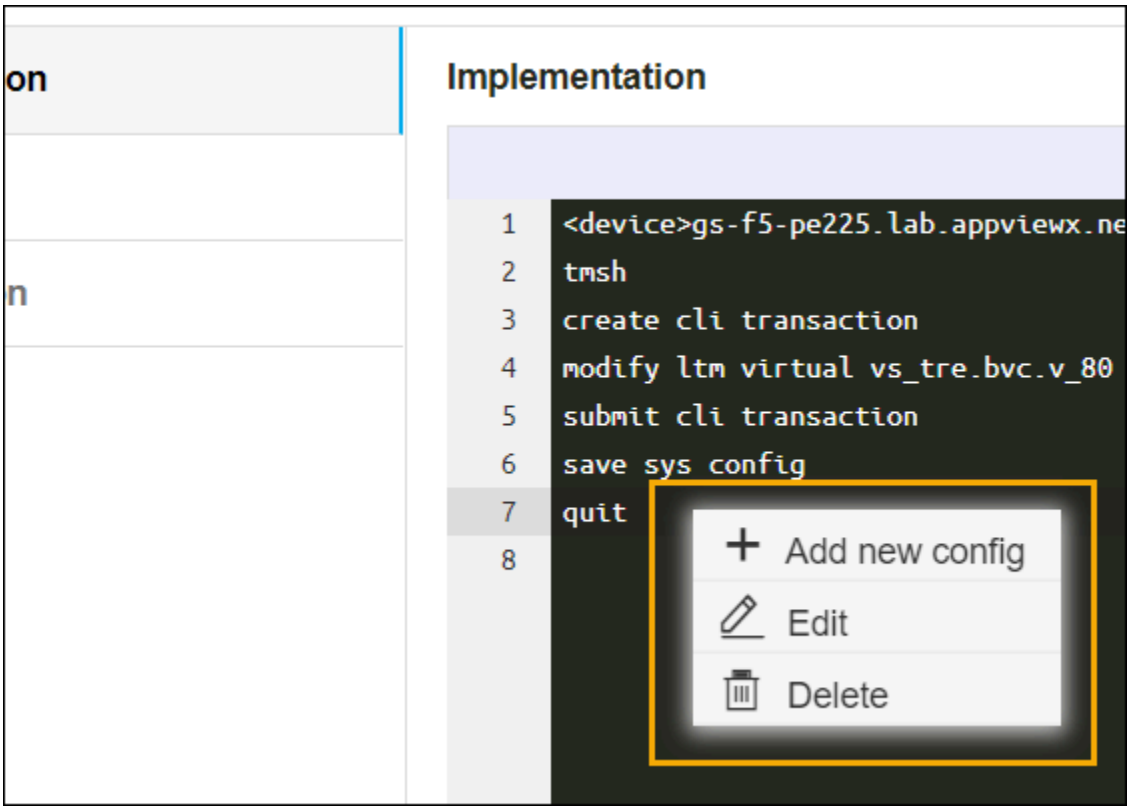
- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



11. (Optional) If you need to change any data at this stage, you can update by clicking the right mouse button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



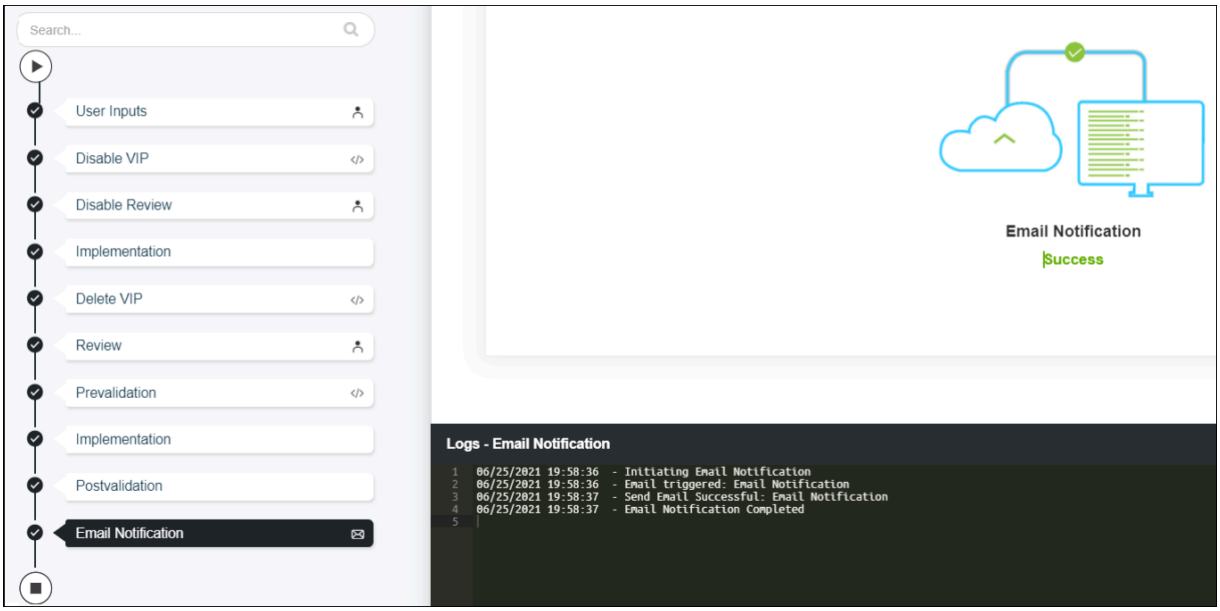
12. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
13. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    After the post-validation is successful, the workflow is created and an email triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Disable Unused F5 LTM VIP

This workflow disables VIPs that are unused for ‘n’ number of days.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable Unused F5 LTM VIP** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Disable Unused F5 LTM VIP :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option. **Note:** Selecting multiple devices is possible.
No of Days	Enter the no. of days to disable the virtual server.

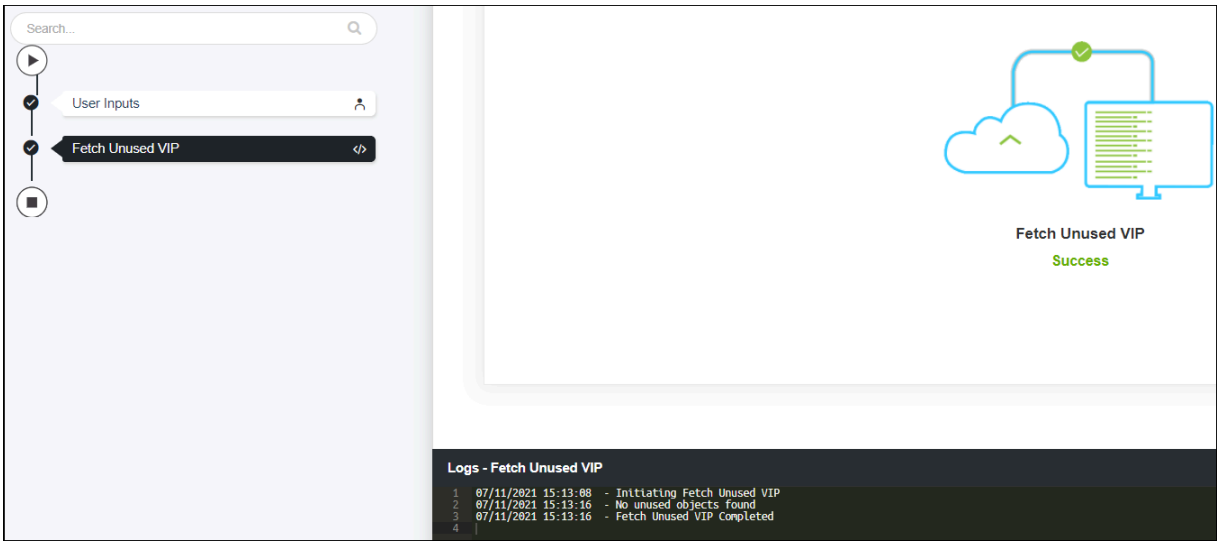
6. Click **Submit**.
   The Confirmation dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The workflow is successfully completed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Disable and Delete Unused F5 LTM VIP

This workflow disables and deletes the VIPs that are unused for 'n' number of days.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable and Delete Unused F5 LTM VIP** workflow. The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Disable and Delete Unused F5 LTM VIP :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

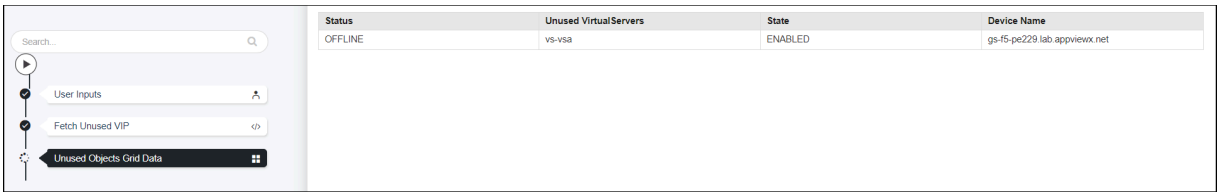
Field	Description
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option. **Note:** Selecting multiple devices is possible.
No of Days	Enter the no. of days for which the virtual server is not used.

6. Click **Submit**.
   The Confirmation dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The fetching unused VIPs starts automatically and reaches the **Unused Objects Grid Data** stage.



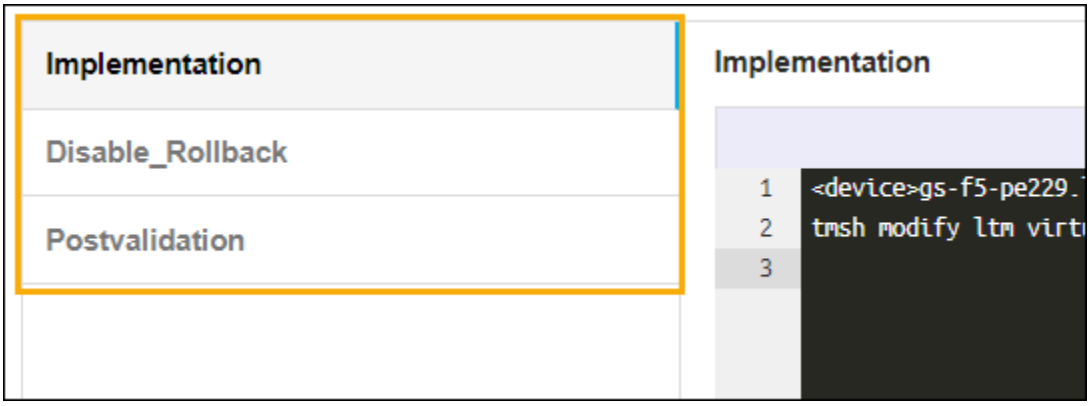
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.
- To download the Unused Objects Grid Data report, click the Download () button.

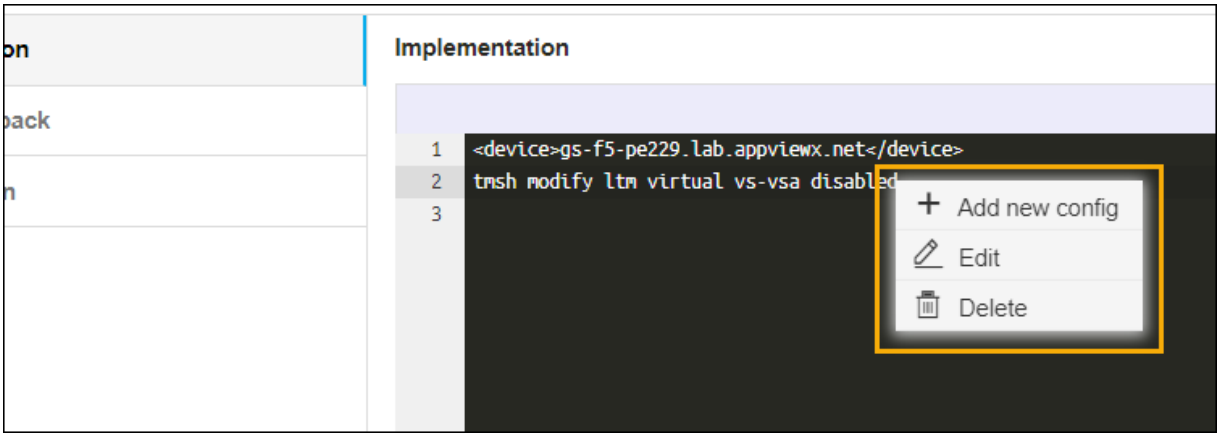
8. Click **Submit**.
   The **Confirmation** dialog box is displayed.
9. Click **Ok** to submit the form.
   The validation starts automatically and reaches the **Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

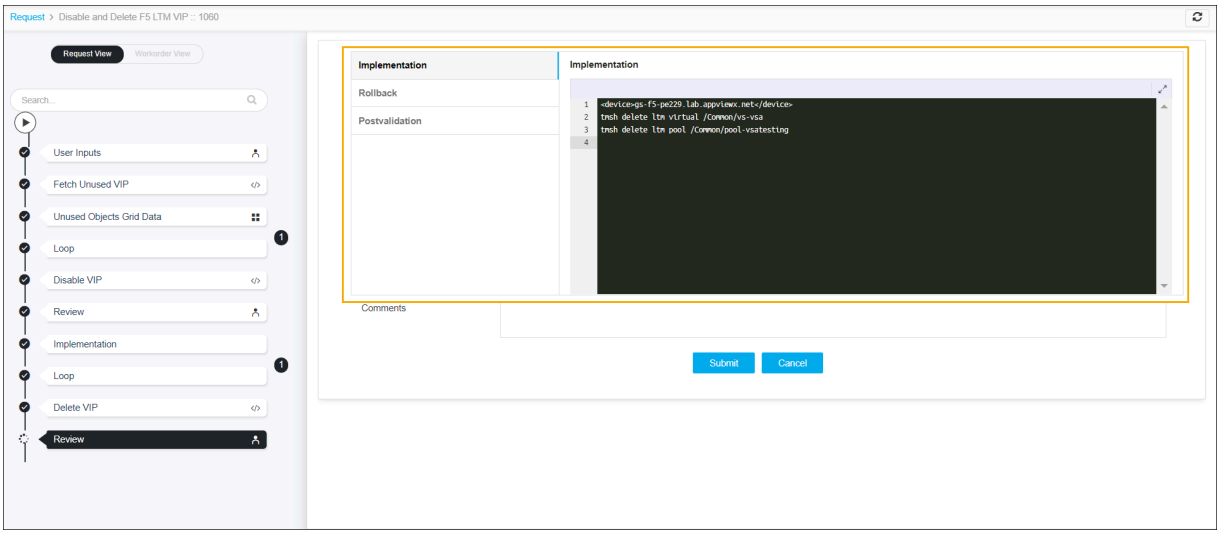


11. (Optional) If you need to change any data at this stage, you can update by clicking the right mouse button.

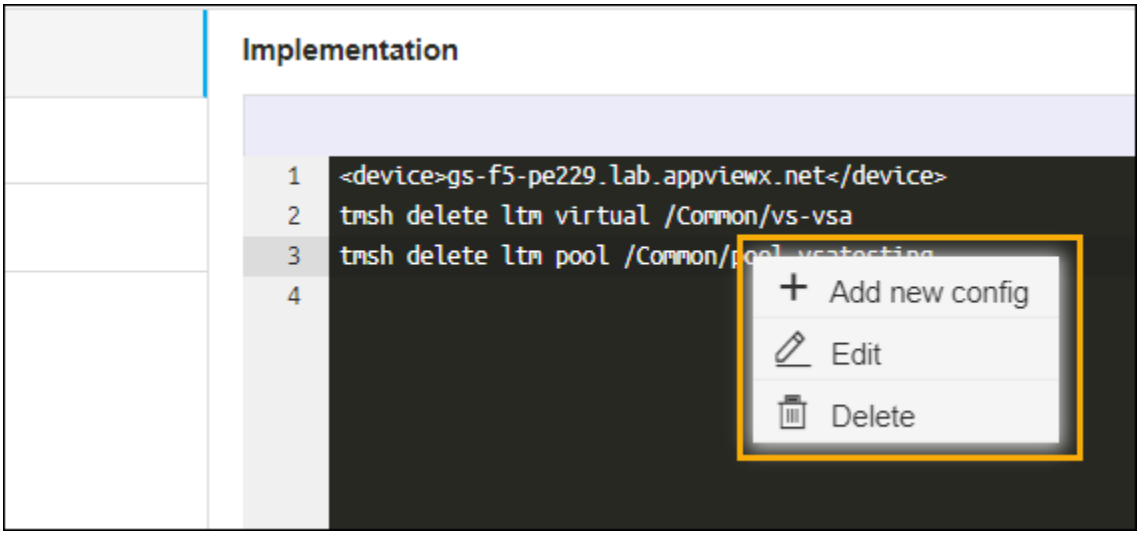


12. After the review, click **Submit**.
    The implementation process starts and reaches the **Review** stage.
13. Review the data, and then click **Submit** to implement the changes.
    The implementation starts.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

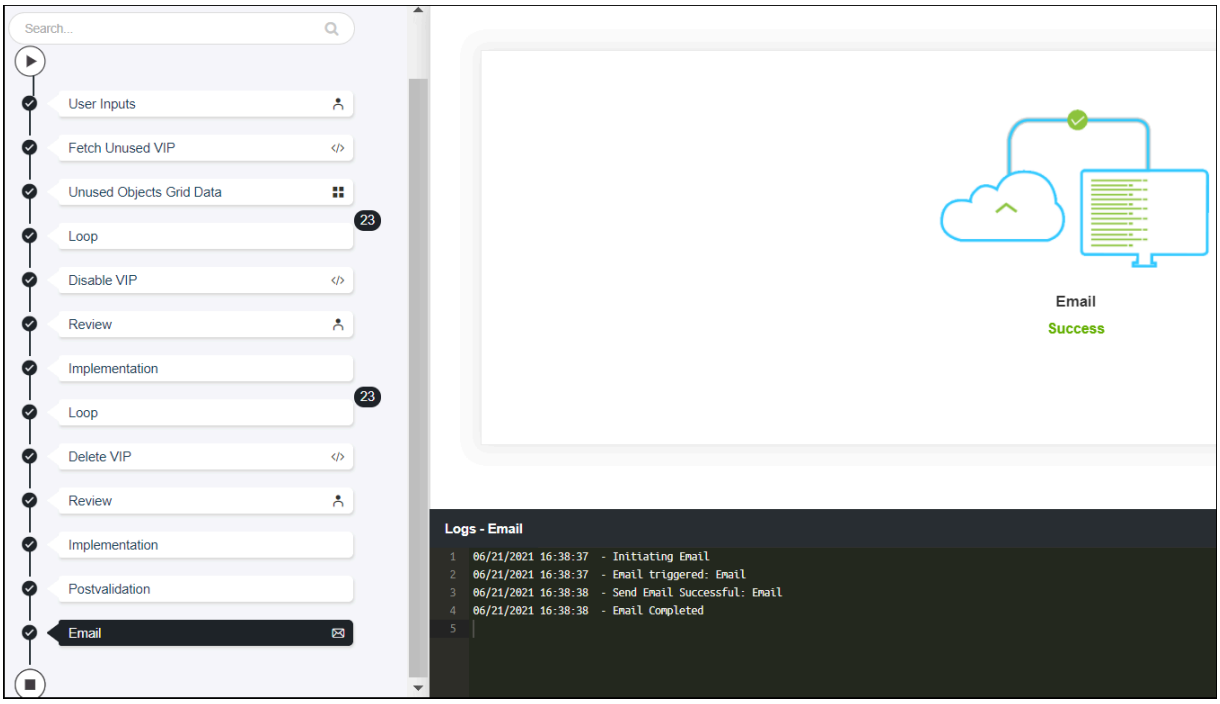


14. (Optional) If you need to change any data at this stage, you can update by clicking the right mouse button.



The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:

The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Get Unused LTM VIP and Delete

By running this workflow, you can delete VIPs that are unused for ‘n’ number of days.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Get Unused LTM VIP and Delete** workflow.

   The **Run** and **Schedule** buttons are displayed.
4. Click Run.

   The **Request > Get Unused LTM VIP and Delete :: FormBuilder** page is displayed.
5. Under **Device Details**, select the **Device Name**, and enter the **No of Days**.
6. Click **Submit**.

   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

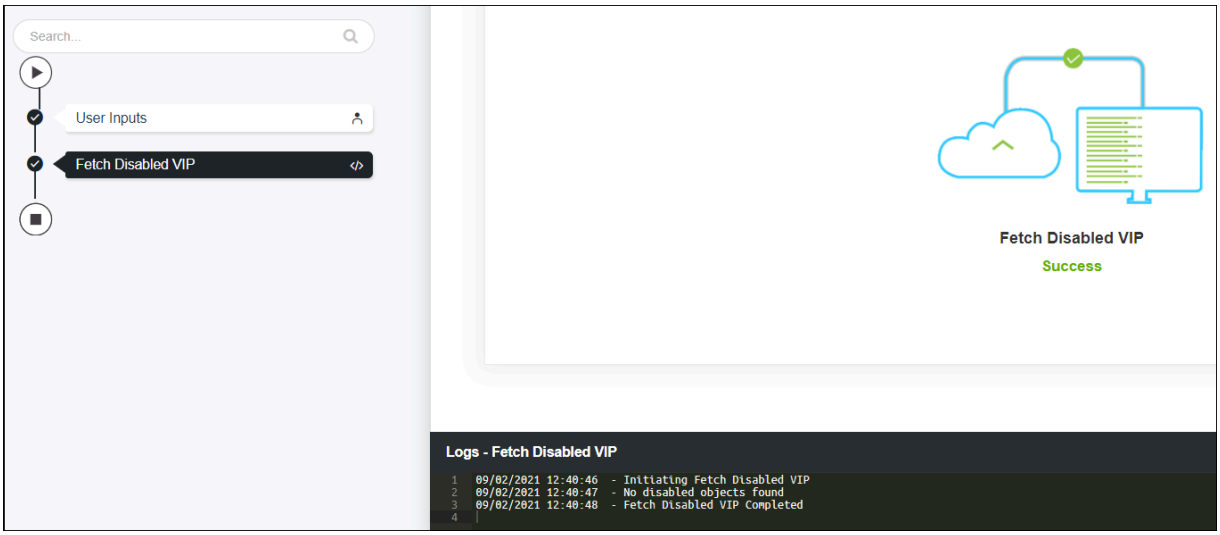
The form will be saved as **Open request** under **Request** > **My Request**.

OR

To submit the form, click **OK**.

The fetching process is automatically.

The unused LTM VIP for the given days is deleted.



Get Unused LTM VIP and Notify

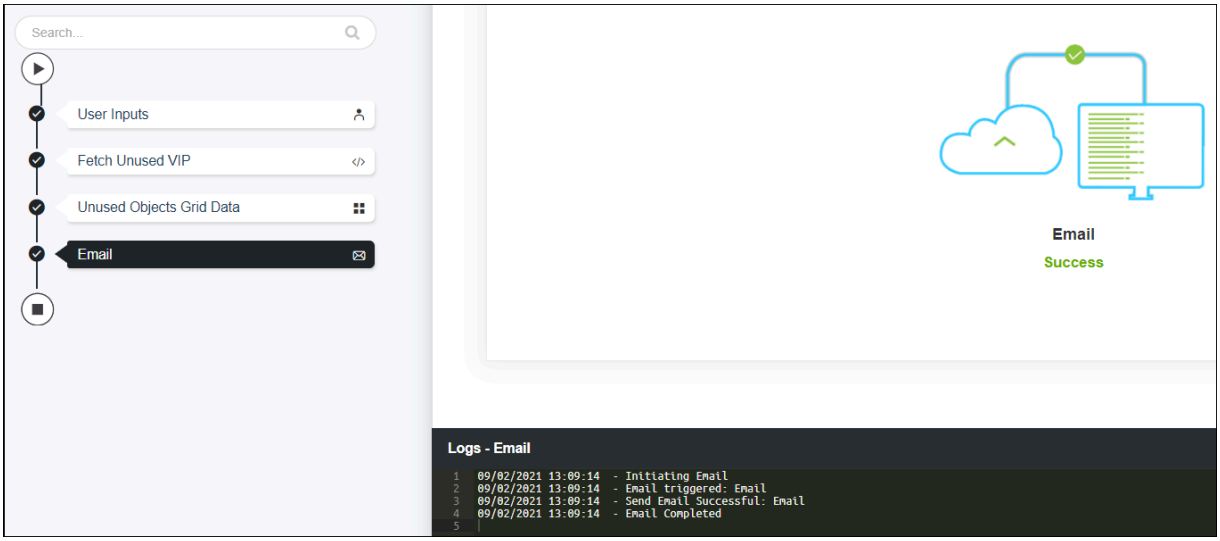
This workflow fetches the report of VIPs that are unused for ‘n’ number of days.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Get Unused LTM VIP and Notify** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Get Unused LTM VIP and Notify :: FormBuilder** page is displayed.
5. Under **Device Details**, select the **Device Name**, and enter the **No of Days**.
6. Click **Submit**.
   The fetching process is automatically.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The unused LTM VIP for the given days is fetched and sent reports via configured email addresses.



Disable and Delete Unused VIP-State/Status

This workflow fetches the unused VIPs' state and status and then disables or disables and deletes them.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable and Delete Unused VIP-State/Status** workflow.

   The **Run** and **Schedule** buttons are displayed.
4. Click 

.

   The **Request > Disable and Delete Unused VIP-State/Status :: FormBuilder** page is displayed.
5. Enter/select the workflow details.

Worflow Details - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option. **Note:** Selecting multiple devices is possible.
No of Days	Enter the no. of days for which the virtual server's state and status is not used.
***Action on VIPs**	The options are: • **Disable** - Choose this option to disable the unused VIPs. • **Disable and Delete** - Choose this option to disable and delete the unused VIPs. • **Disable Monitor and Delete** - Disables the identified unused VIPs, monitors them, and then deletes them, if they remain unused within the monitoring period. If you choose this option, provide the monitoring period in the **Monitoring Period** field. Enter a period within the range of 1 to 31 days.
***Unused VIP Metrics**	Choose the VIP's metrics for which the unused VIP has to be fetched.
** : Mandatory fields*	

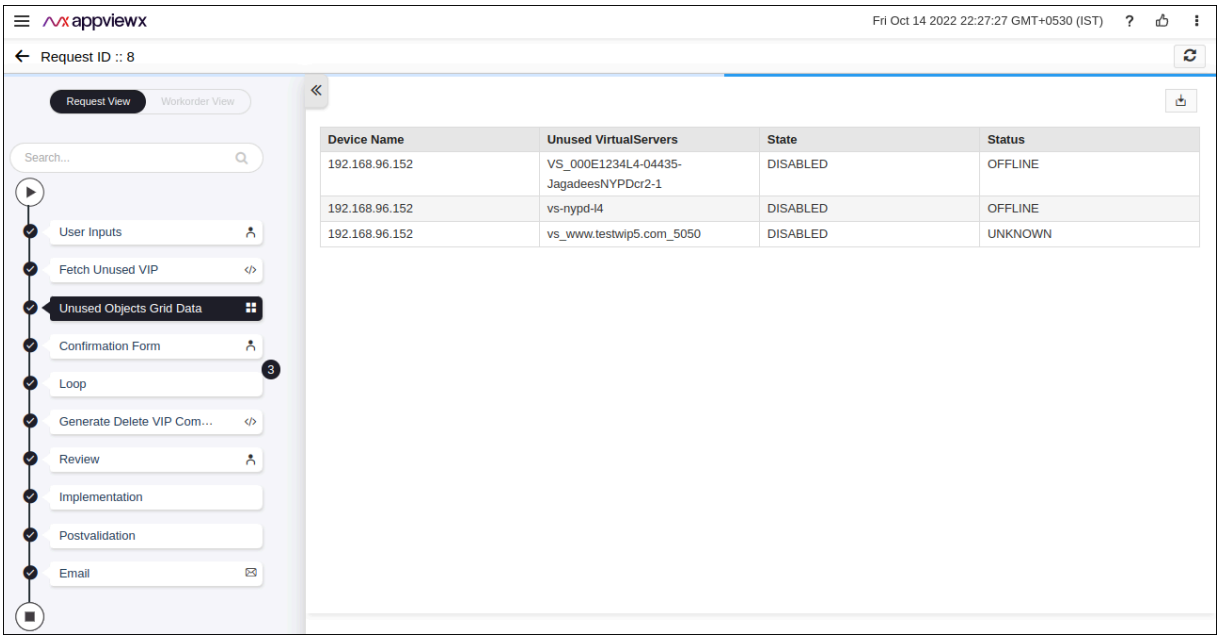
6. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The fetching unused VIPs starts automatically and reaches the **Unused Objects Grid Data** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:

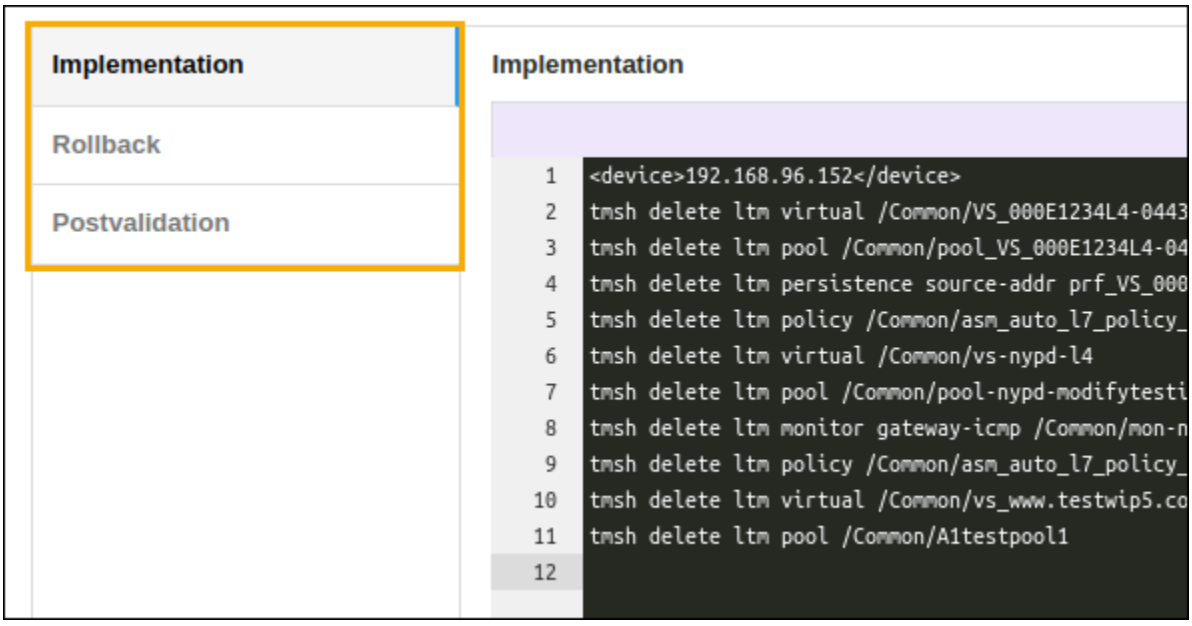
- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.
- To download the Unused Objects Grid Data report, click  .

7. Click **Submit**.

   The **Confirmation** dialog box is displayed.
8. Click **Ok** to submit the form.

   The validation starts automatically and reaches the **Review** stage.
9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

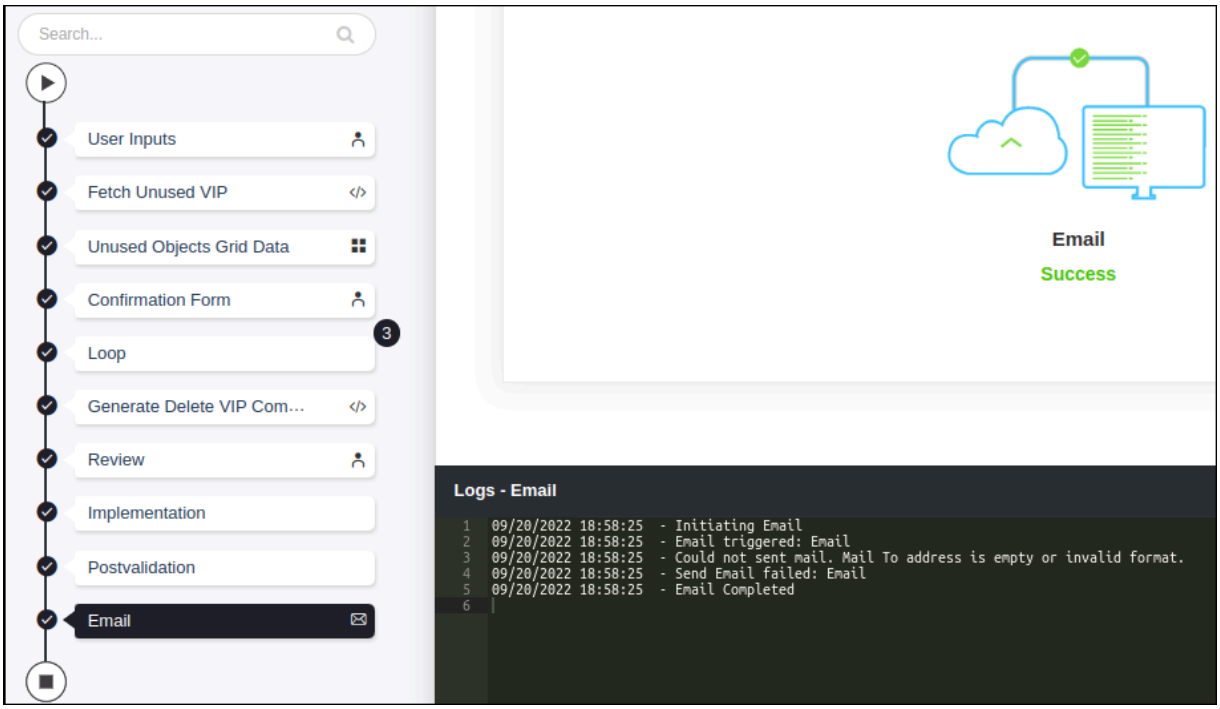
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



10. (Optional) If you need to update any data at this stage, right click the mouse button and select the required option.
11. After the review, click **Submit**.
    The implementation process starts and reaches the **Review** stage.

    The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Disable and Delete Unused VIP-State/Status/Statistics

This workflow fetches the unused VIPs' state, status, statistics, and then disables or disables and deletes them.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable and Delete Unused VIP-State/Status/Statistics** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Disable and Delete Unused VIP-State/Status/Statistics :: FormBuilder** page is displayed.
5. Enter/select the workflow details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Workflow Details - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option. **Note:** Selecting multiple devices is possible.
No of Days	Enter the no. of days for which the virtual server is not used.
***Action on VIPs**	The options are: • **Disable** - Choose this option to disable the unused VIPs. • **Disable and Delete** - Choose this option to disable and delete the unused VIPs.

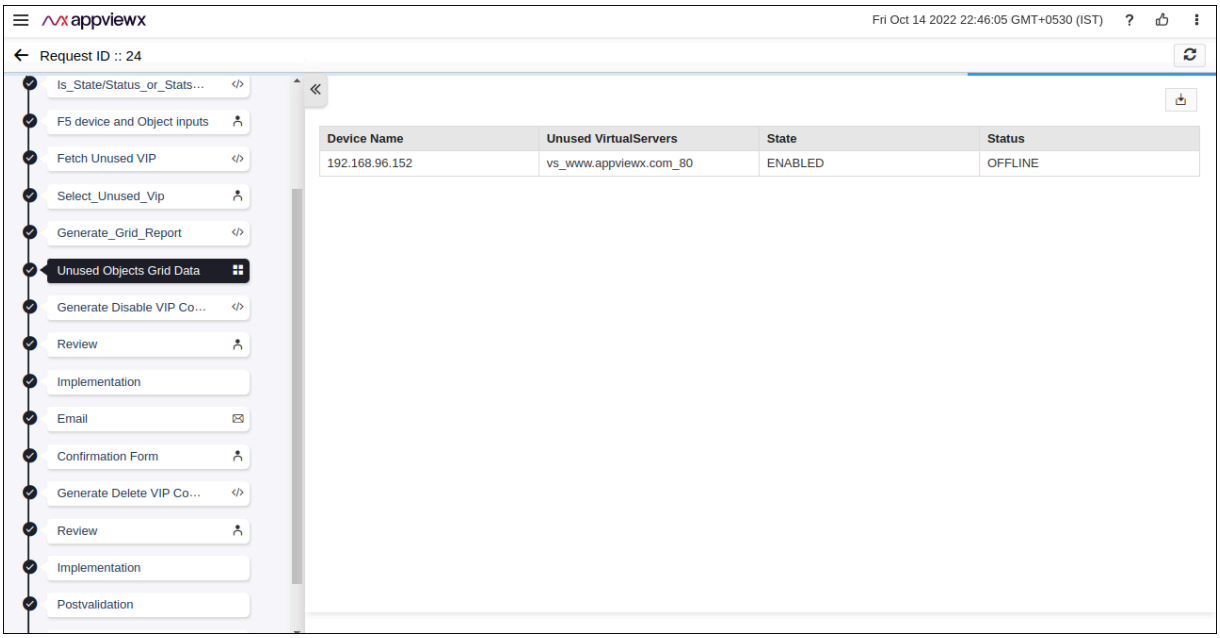
6. Click **Submit**.
   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The fetching of unused VIPs starts automatically and reaches the **Unused Objects Grid Data** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:

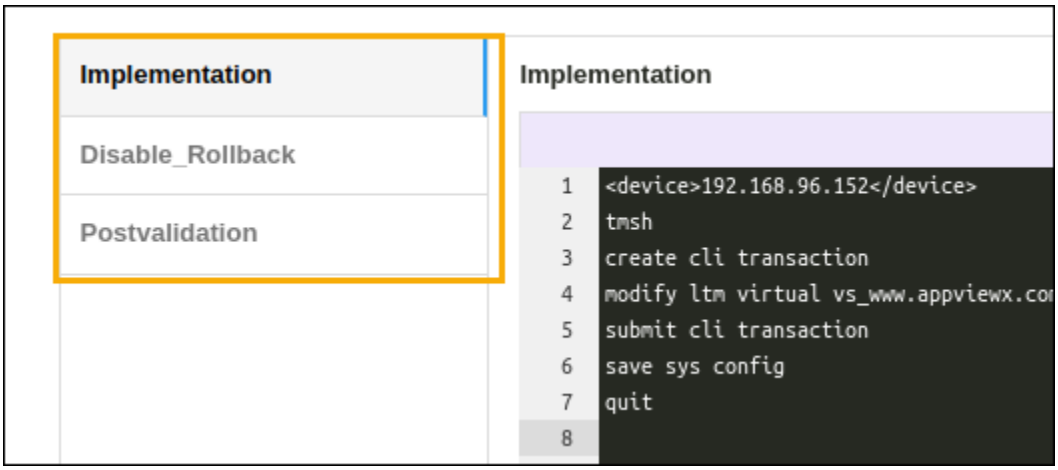
- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.
- To download the Unused Objects Grid Data report, click the Download ( ) button.

8. Click **Submit**.

   The **Confirmation** dialog box is displayed.
9. Click **Ok**.

   The validation starts automatically and reaches the **Review** stage.
10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

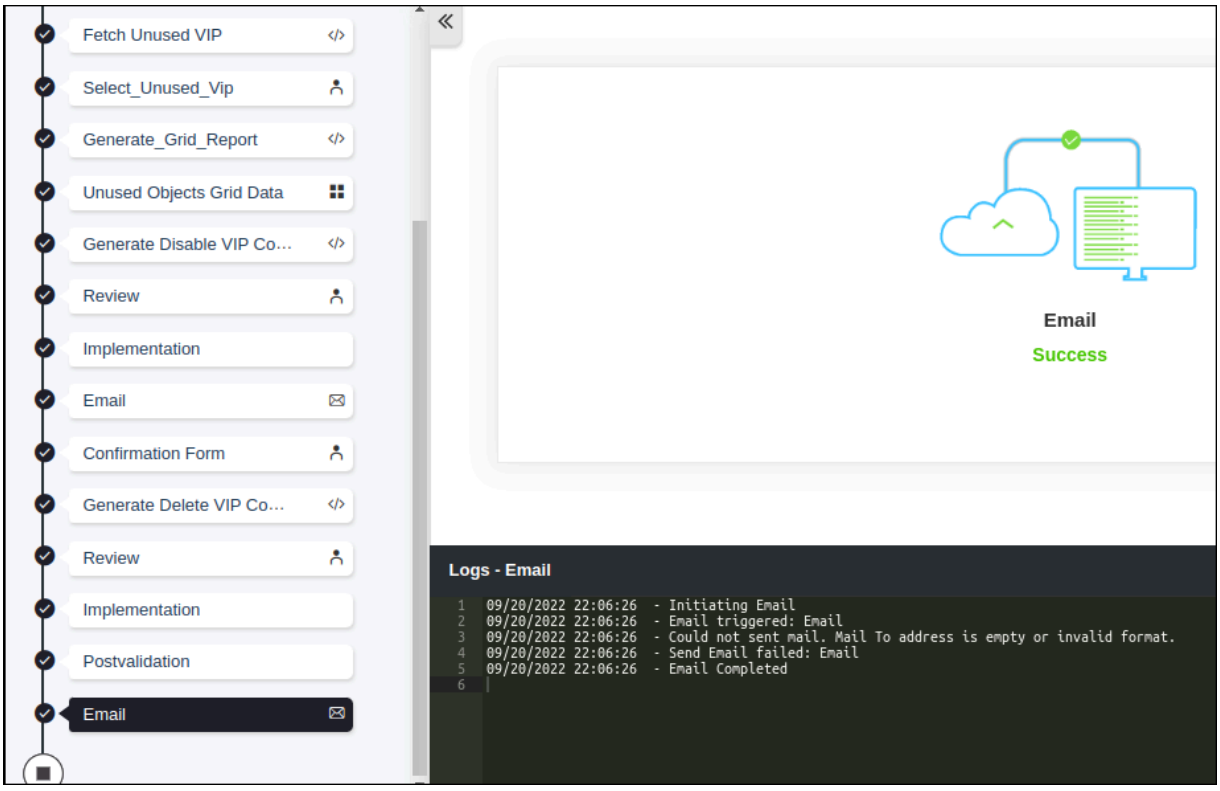
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



11. (Optional) If you need to update any data at this stage, right click the mouse button and select the required option.
12. After the review, click **Submit**.
    The implementation process starts and reaches the **Confirmation Form** stage.
13. Click **Click Submit to Delete**.
    The implementation starts and reaches the **Review** stage.
14. Review the data, and then click **Submit** to implement the changes.
    The implementation starts.

    The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Disable and Delete Unused VIP-Statistics

This workflow fetches the unused VIPs' statistics and then disables or disables and deletes them.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable and Delete Unused VIP-Statistics** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Disable and Delete Unused VIP-Statistics :: FormBuilder** page is displayed.
5. Enter/select the workflow details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Workflow Details - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option. **Note:** Selecting multiple devices is possible.
No of Days	Enter the no. of days for which the virtual server's state and status is not used.
***Action on VIPs**	The options are: • **Disable** - Choose this option to disable the unused VIPs. • **Disable and Delete** - Choose this option to disable and delete the unused VIPs.

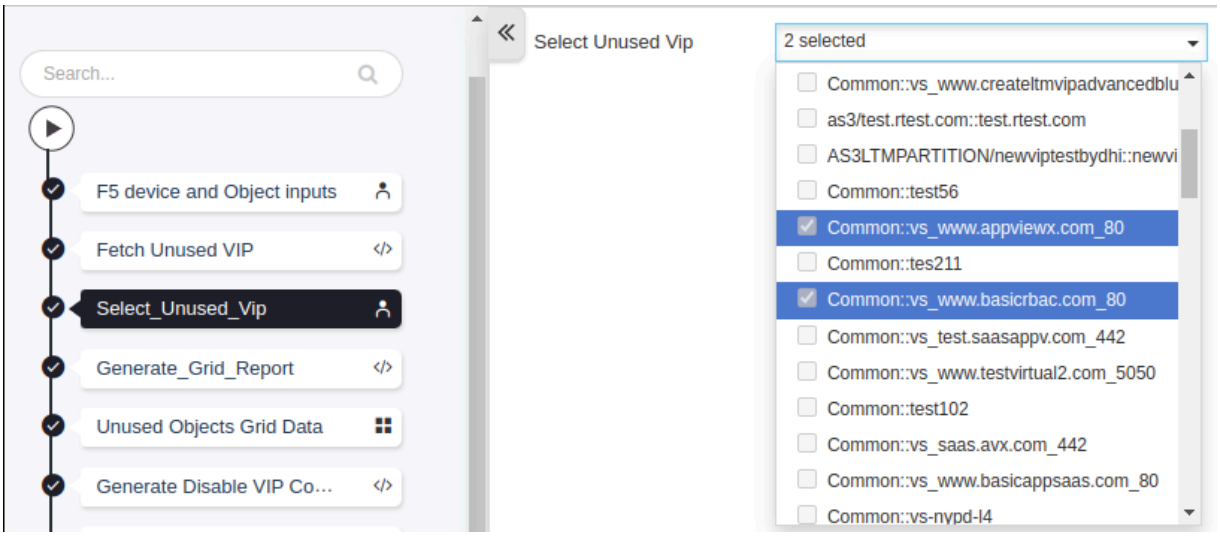
6. Click **Submit**.
   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The fetching of unused VIPs starts automatically and reaches the **Select_Unused_Vips** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

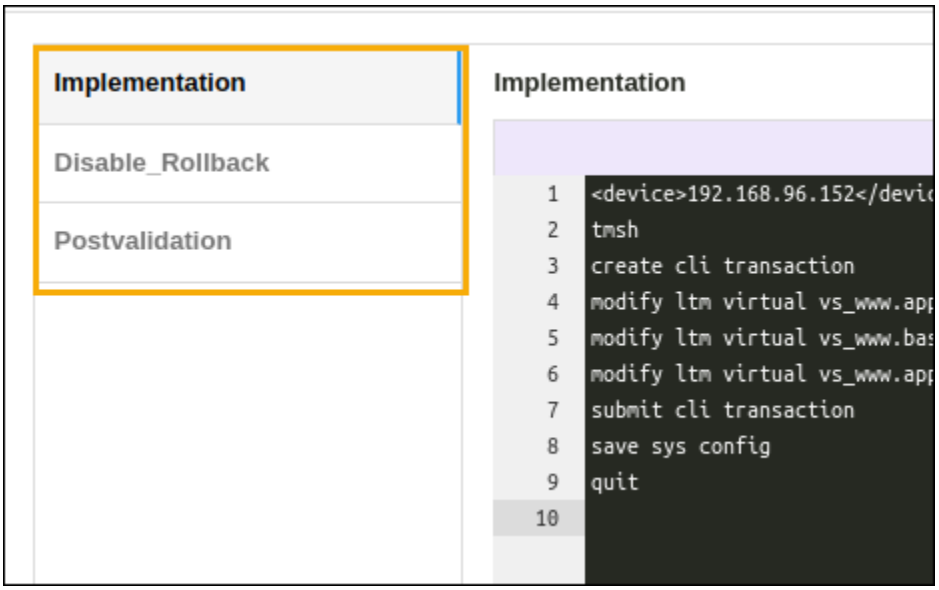


8. Select the required unused VIP(s) from the list, and click **Submit**.
   The generation of report starts automatically and reaches the **Unused Objects Grid Data** stage.

> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.
> - To download the Unused Objects Grid Data report, click  .

9. Click **OK** to submit the form.
10. Click **Submit**.
    The **Confirmation** dialog box is displayed.
11. Click **OK** to submit the form.
    The validation starts automatically and reaches the **Review** stage.
12. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



13. (Optional) If you need to change any data at this stage, right click and select the required option.
14. After the review, click **Submit**.

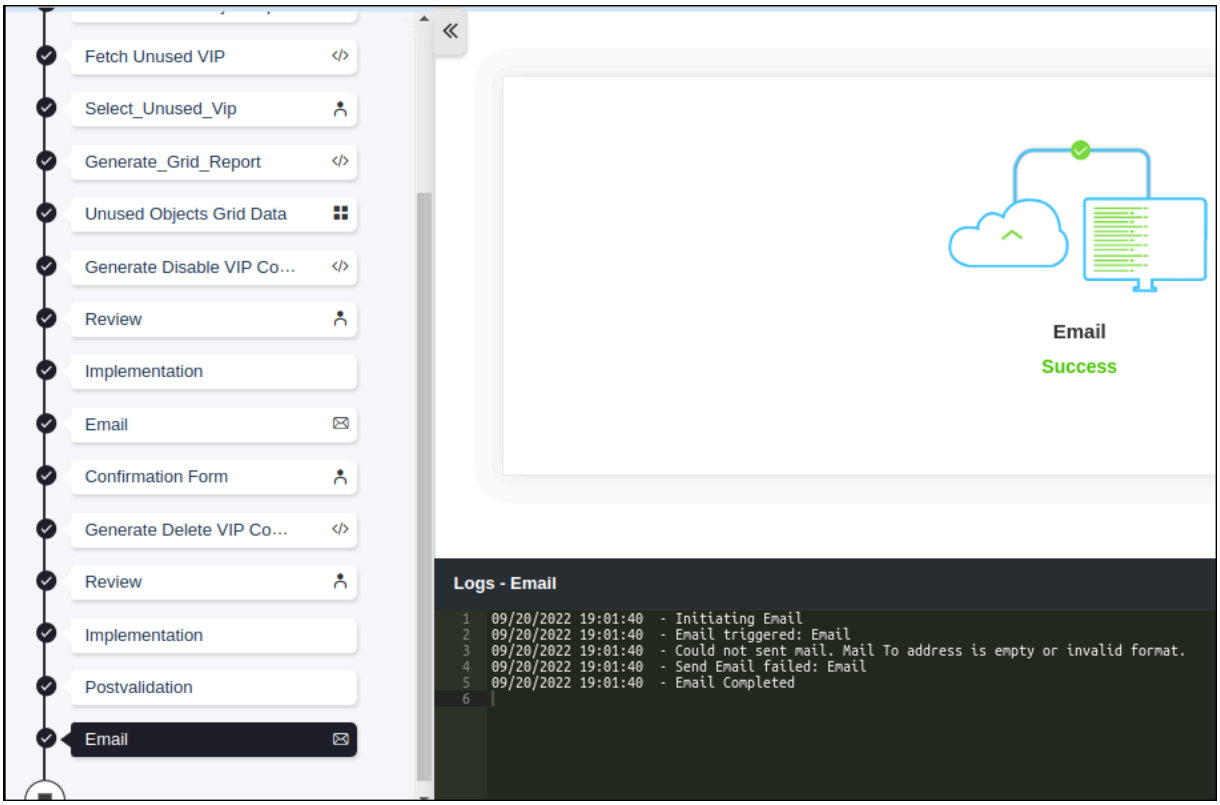
    The implementation process starts and reaches the **Confirmation Form** stage.
15. Click **Click Submit to Delete**.

    The implementation starts and reaches the **Review** stage.
16. After the review, click **Submit**.

    The implementation process starts.

    The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Notify Unused VIP-StateStatusStatistics

This workflow fetches the unused VIPs' and notifies through email.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Notify Unused VIP-StateStatusStatistics** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Notify Unused VIP-StateStatusStatistics :: FormBuilder** page is displayed.
5. Enter/select the workflow details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Workflow Details - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option. **Note:** Selecting multiple devices is possible.
No of Days	Enter the no. of days for which the virtual server's state and status is not used.
***Unused VIP Metrics**	Choose the VIP's metrics for which the unused VIP has to be fetched.

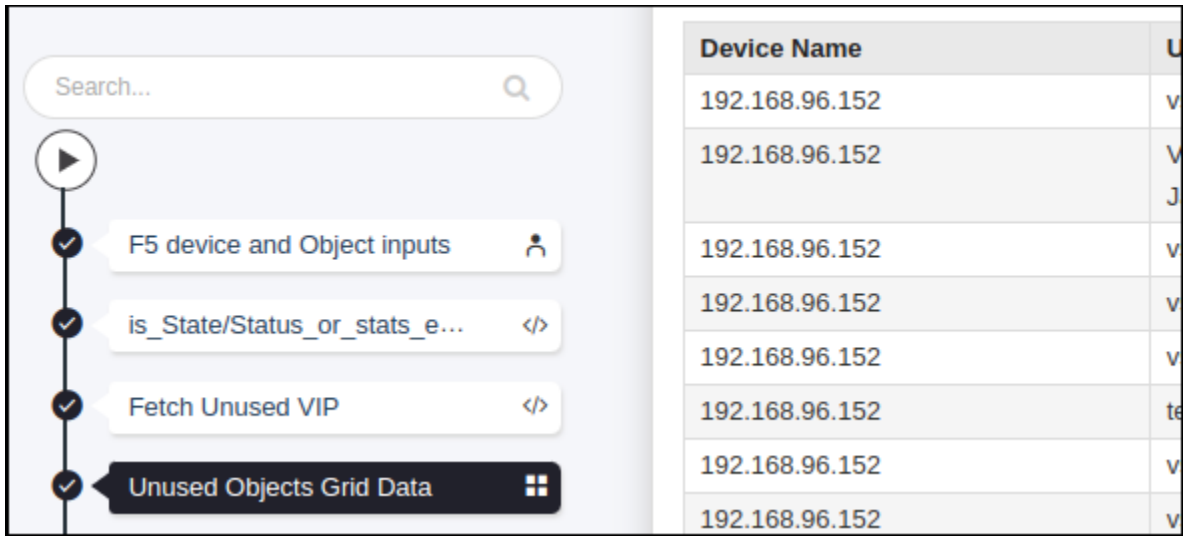
6. Click **Submit**.
   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The fetching of unused VIPs starts automatically and reaches the **Unused Objects Grid Data** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



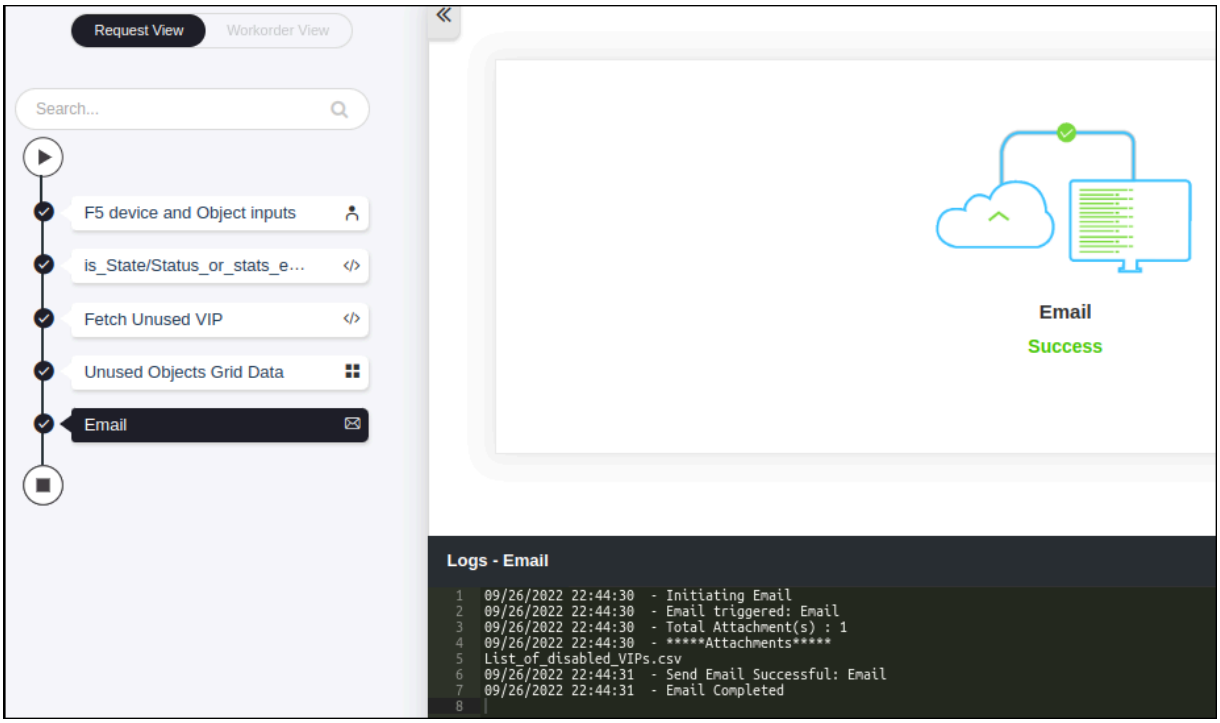
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.
- To download the Unused Objects Grid Data report, click the Download ( ) button.

8. Click **Submit**.
   The **Confirmation** dialog box is displayed.
9. Click **Ok** to submit the form.
   The validation starts automatically.

   The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

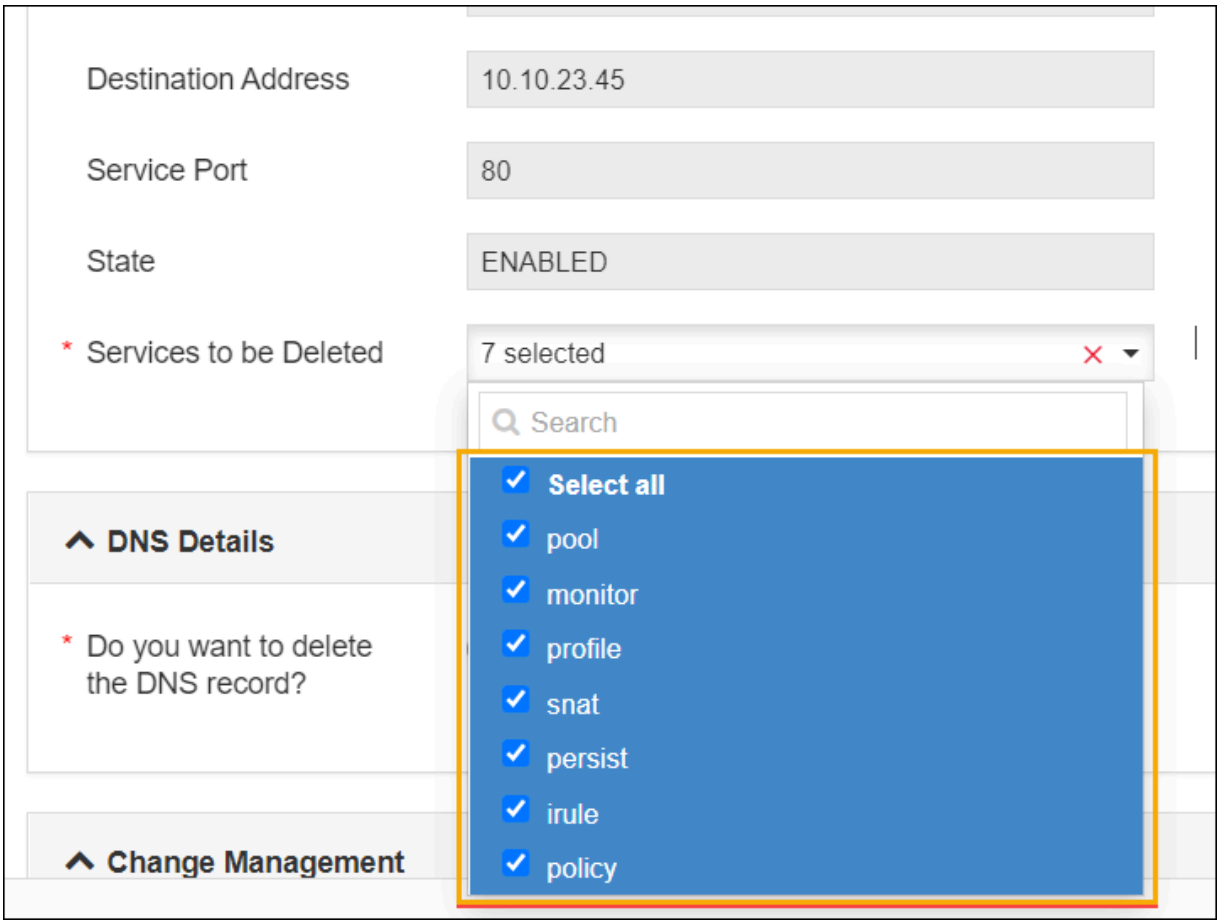
Delete F5 LTM VIP with Service Now

This workflow deletes VIP along with its associated objects(Pool, Monitor, IRule, Profiles, Persistence, Snatpool, Policy) and multi-select the associated object types to consider for deletion.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 LTM VIP with Service Now** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete F5 LTM VIP with Service Now :: FormBuilder** page is displayed.
5. Select the **Datacenter**, the **Device Name**, and the **Virtual Server** from the dropdown list.
   The services associated with the selected device are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Select all the services of the device or select/clear only the services of the devices that are to be deleted.
7. Select the **DNS Details** and **Change Management** details of the device, if DNS details and change management details of the device are required to be deleted.
8. Click **Submit**.
   The **Confirmation** dialog box is displayed.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Disable Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

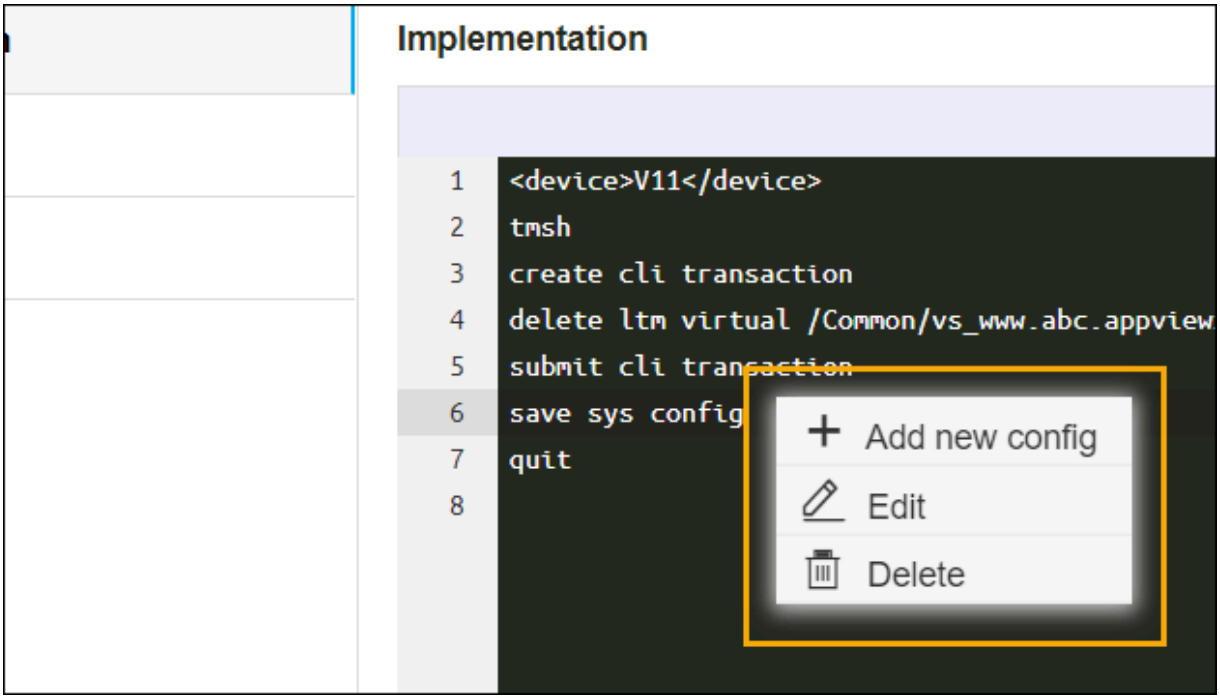
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



11. (Optional) If you need to change any data at this stage, right click and select the required option.



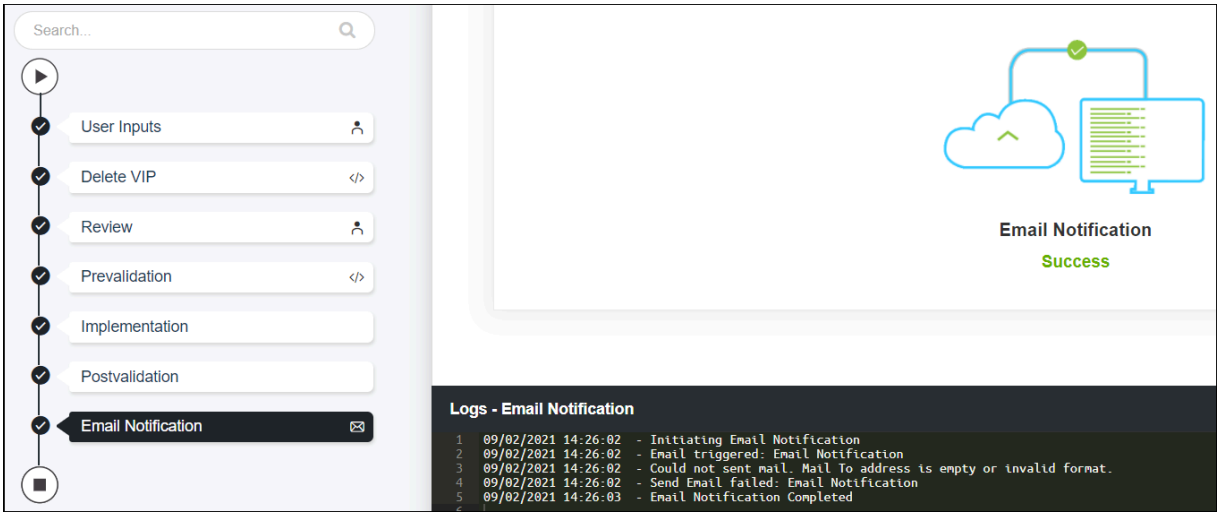
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

12. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, Click **Cancel**.

13. Click **Ok** to continue workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.



> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Manage F5 LTM Pool Members

This workflow selects VIP in a device and modifies (add/delete) associated pool members.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Manage F5 LTM Pool Members** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

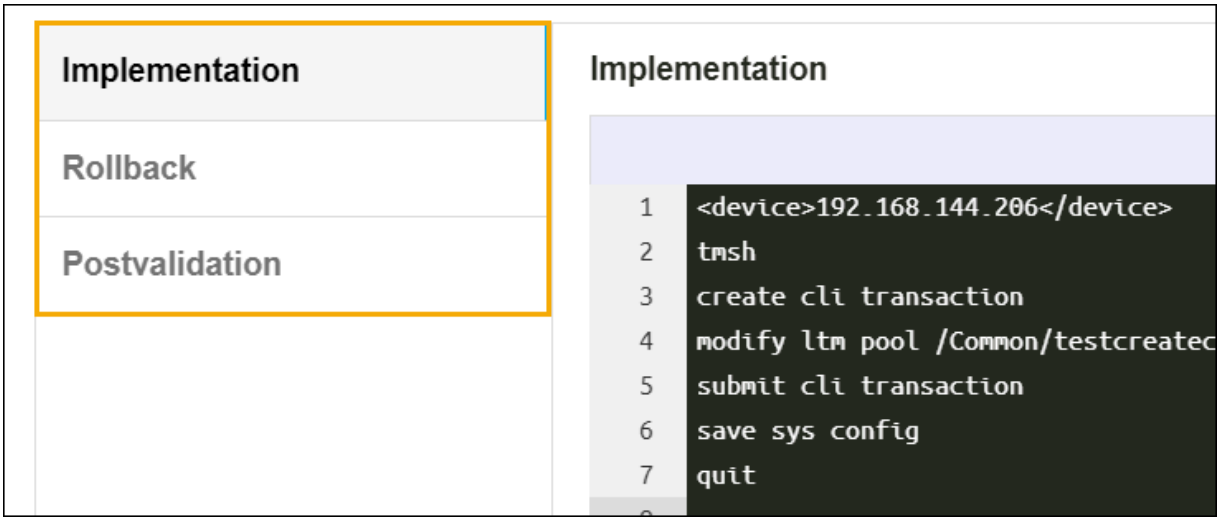
The **Request > Manage F5 LTM Pool Members :: FormBuilder** page is displayed.

5. Enter a valid IP address and port number in the **Destination Address** and **Service Port** field in the **Virtual Server Details** section.
6. Click **Get Virtual Server Details**.
   The devices are loaded.
7. Select the device name from the **Device Name** dropdown list.
   The other details of the device are retrieved automatically.
8. Click **Fetch Pool Members** to get the pool member of the selected device or manually enter the pool member details.
9. Click **Submit**.
   The **Confirmation** dialog box is displayed.
10. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

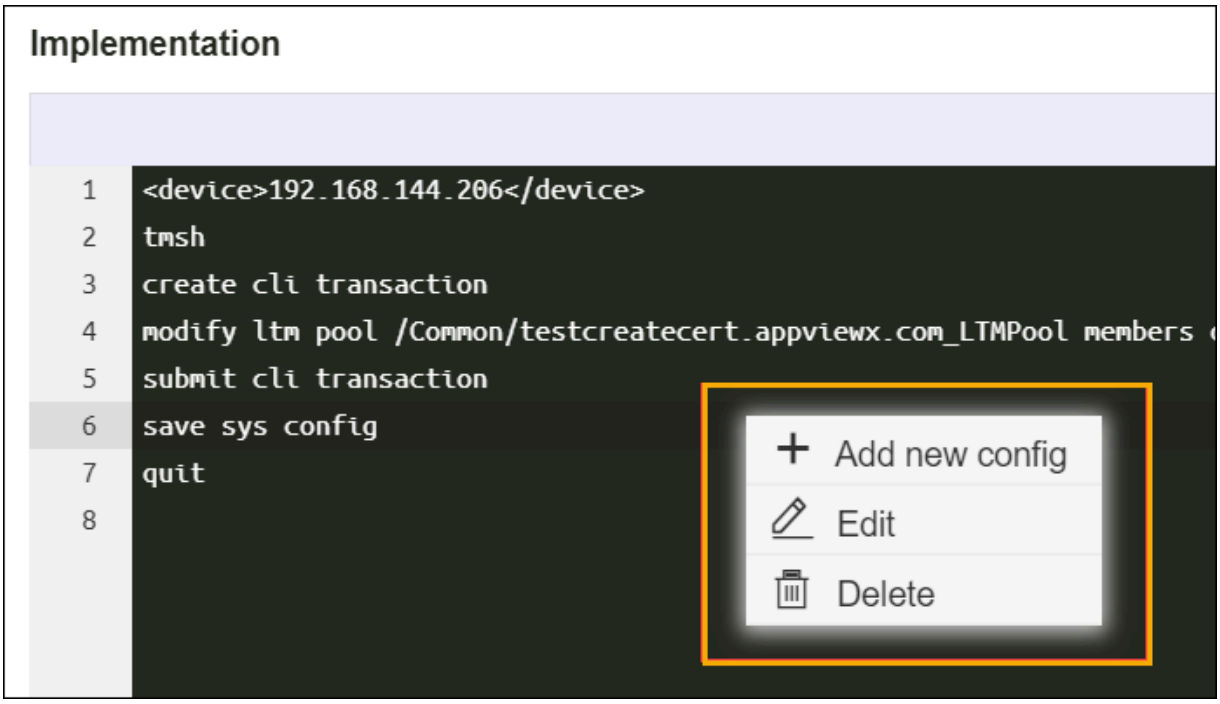
    To submit the form, click **OK**.

    The pre-validation starts automatically and reaches the **Review** stage.
11. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



12. (Optional) If you need to change any data at this stage, right click and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



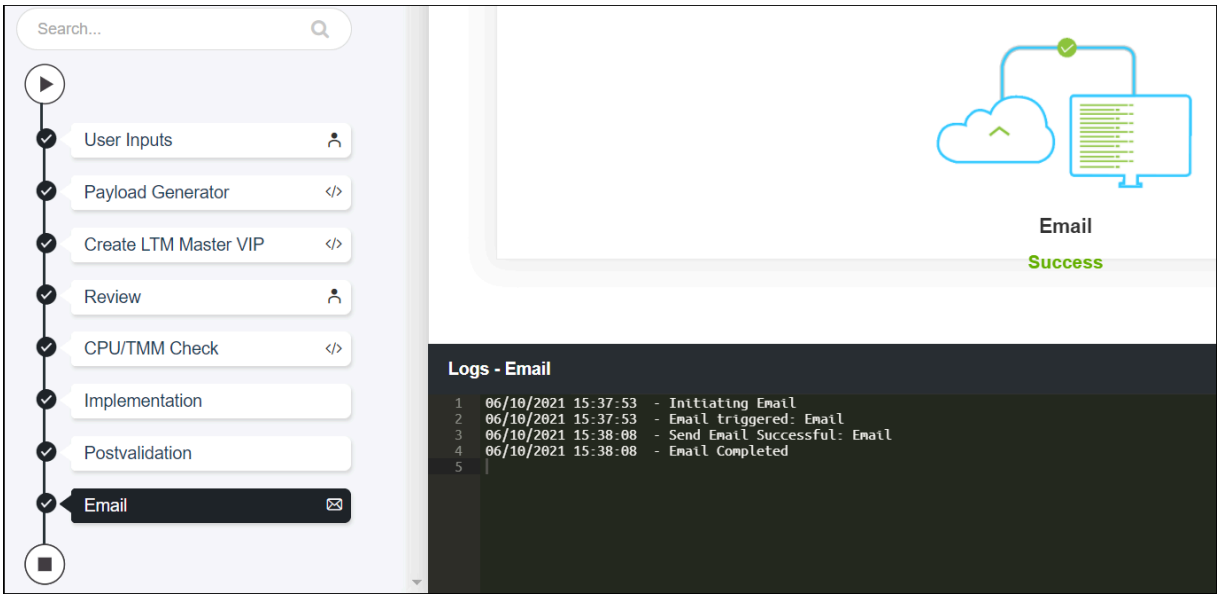
13. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, Click **Cancel**.

14. Click **Ok** to continue the workflow creation.
    It takes a while to complete the post-validation.

    After the successful post-validation, the workflow is created and the email triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view any validation stage, click on the respective stage.

Manage F5 LTM Pool members with Service Now

This workflow selects VIP in a device and modifies (add/delete) associated pool members.

To run this workflow:

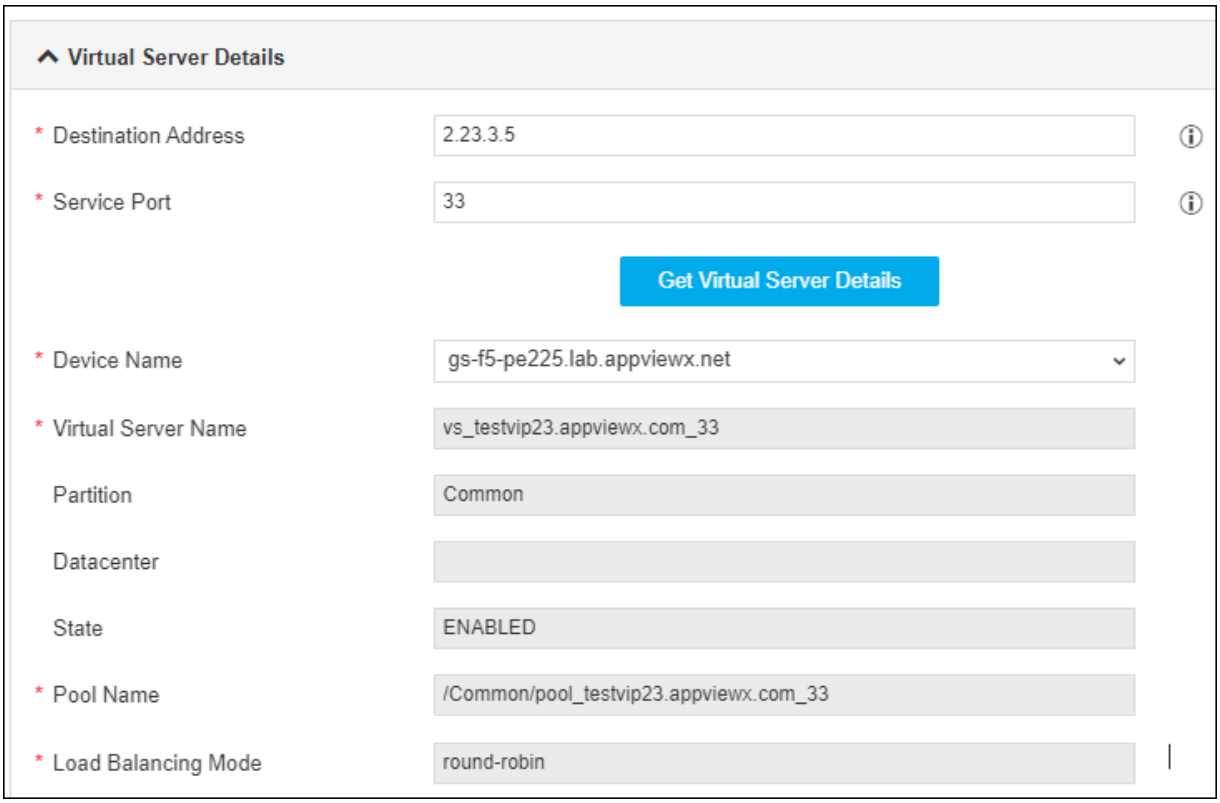
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Manage F5 LTM Pool members with Service Now** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Manage F5 LTM Pool members with Service Now :: FormBuilder** page is displayed.
5. Enter a valid IP address and port number in the **Destination Address** and **Service Port** field in the **Virtual Server Details** section.
6. Click **Get Virtual Server Details**.
   The devices are loaded.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The devices are fetched for the given destination address.

7. Select the device from the **Device Name** dropdown list.

   The virtual server details and pool details are fetched for the selected device.



8. Update the **Pool Member Details** as required.
9. Update the **Change Management** details as required.
10. Click **Submit**.

    The **Confirmation** dialog box is displayed.
11. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

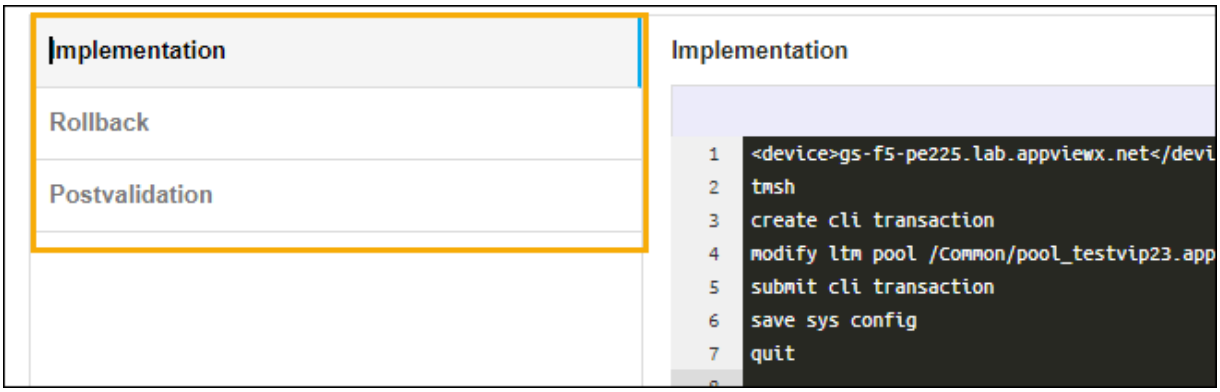
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

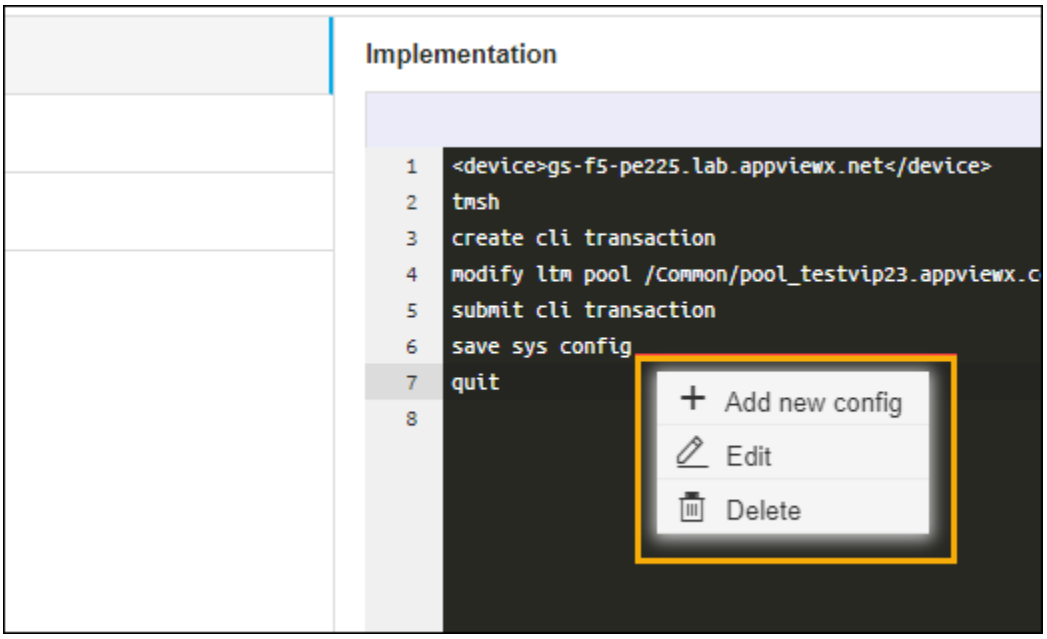
    To submit the form, click **OK**.

    The pre-validation starts automatically and reaches the **Review** stage.
12. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



13. (Optional) If you need to change any data at this stage, right click and select the required option.



14. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

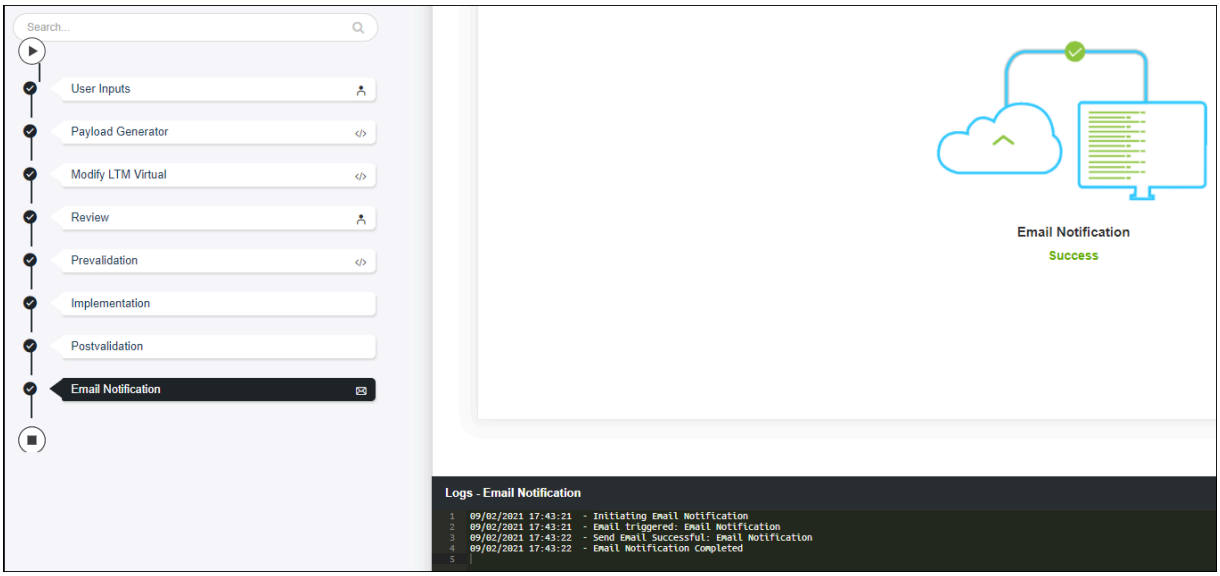
> **Note:**
> To stop running the workflow creation, click **Cancel**.

15. Click **Ok** to continue the workflow creation.

    It takes a while to complete the post-validation.

    After the successful post-validation, the workflow is created and the email triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view any validation stage, click on the respective stage.

Modify F5 LTM VIP Advanced

This workflow modifies VIP along with its associated objects(Pool, Monitor, IRule, Profiles, Persistence, Snatpool)

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify F5 LTM VIP Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Modify F5 LTM VIP Advanced :: FormBuilder** page is displayed.
5. Select the **Datacenter**, **Device Name**, and **Virtual Server** from the dropdown list, and then click the **Get Virtual Server config** button.
   The device details, such as SSL and Persistence, Rule, etc. are auto-populated.
6. Update the **SSL and Persistence**, **Rule**, **Protocol**, **Snat Pool**, **Mirror and Source-Port**, **Pool and Monitor**, and/or **Pool Members** details as required.
7. Click **Submit**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

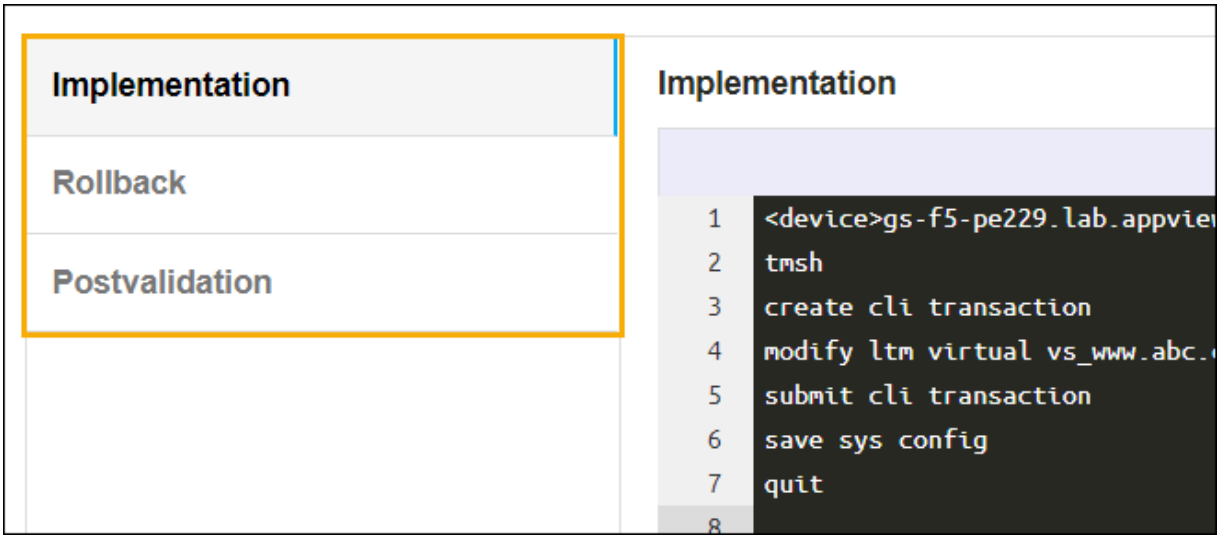
The **Confirmation** dialog box is displayed.

8. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

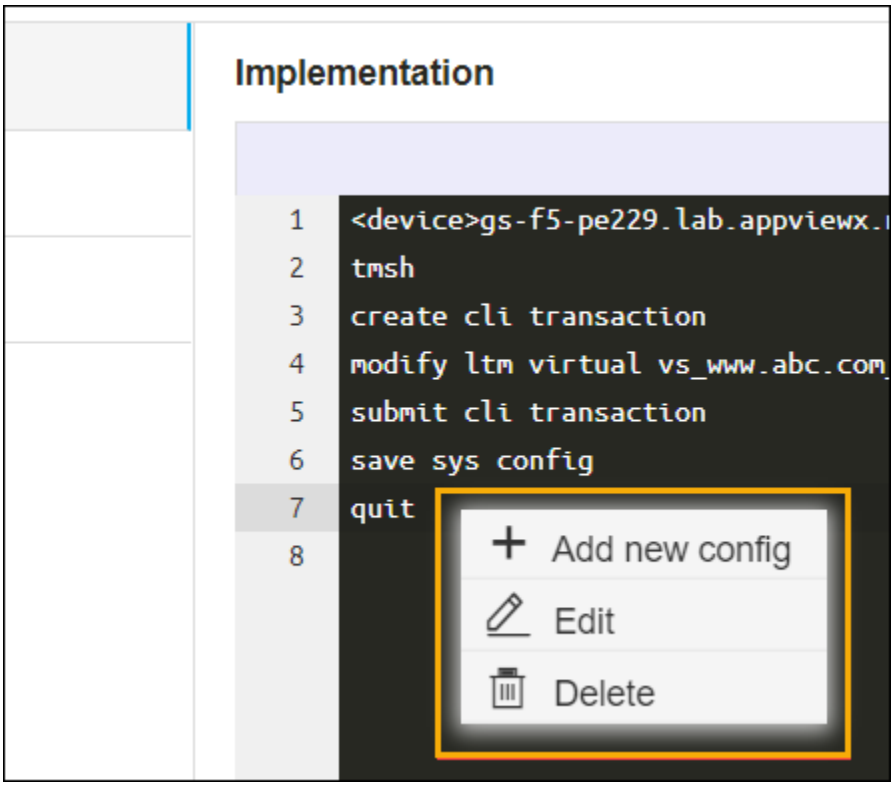
   To submit the form, click **OK**.

   The pre-validation starts automatically and reaches the **Review** stage.
9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



10. (Optional) If you need to change any data at this stage, right click and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



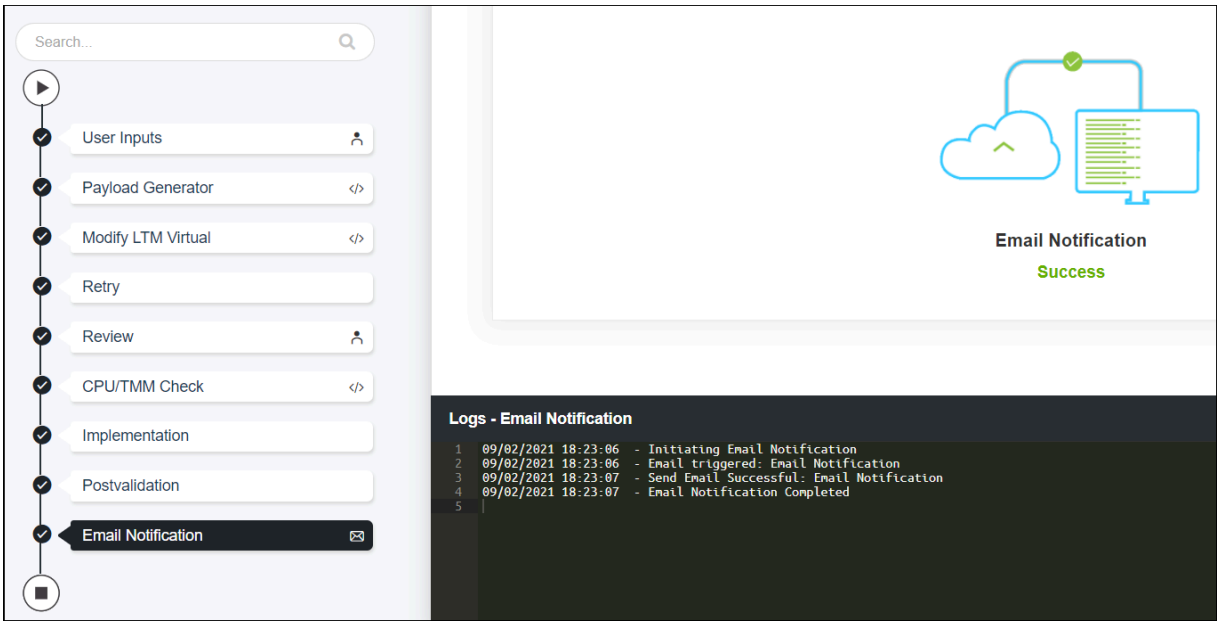
11. After the review, click **Submit**.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the post-validation.

    After the successful post-validation, the workflow is created and the email triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view any validation stage, click on the respective stage.

Create F5 LTM Multiple VIP

This workflow creates VIP with Single Pool and Single Monitor on Multiple Devices and also creates Multiple VIPs on the same Device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM Multiple VIP** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 LTM Multiple VIP :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter/select the **Virtual Details**.

Virtual Details Section - Field and Description Table

Field	Description
***Application Name**	Enter the application name of the virtual server.
***Virtual Server Type**	The attributes of this virtual server. The default is a Standard. • **Standard** - A virtual server that directs client traffic to a load balancing pool and is the most basic type of virtual server. When you first create the virtual server, you assign an existing default pool to it. From then on, the virtual server automatically directs traffic to that default pool. • **Performance (Layer 4)** - A virtual server that shares the same IP address as a node in an associated VLAN.
***Destination Address**	Enter the destination IP address information for the virtual server.
***Service Port**	Enter a service port.

7. Enter/select the **Pool Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Pool Details Section - Field and Description Table

Field	Description
Load Balancing Method	The load balancing method is used to select a pool in this WideIP. The default is a round-robin. The methods are: • **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.
Address	Enter the IP address of the pool.
Service Port	Enter a service port.
Pool Member Status	The current state of the pool members. The statuses are: • **user-enabled** - when you select this option, the system sends traffic to this pool member regardless of the pool member's state. • **user-disabled** - when this option is selected, the pool member can handle only persistent or active connections.
Priority Group	A number representing the priority group for the pool members. To specify a priority, you must activate priority group usage when you create a new pool or when adding or removing pool members. When activated, the system load balances traffic according to the priority group number assigned to the pool member.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Pool Members	Enter the IP address of the pool member. And then click the Add  button. Any number of pool numbers can be added to the pool. After adding the pool, you can manage them.

8. Enter/select the **Snat Details**.

Snat Details Section - Field and Description Table

Field	Description
Snat Choice	Select the SNAT choice for any connections using this pool. The options are: • **AutoMap** -This option allows you to select a translation address from the available self-IP address. • **SNAT** - This option allows you to select a floating self IP as a translation address.

9. Enter/select the **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
Monitor Type	Select the health monitors that are available to add for the pool: • **HTTP** • **HTTPS** • **TCP** • **TCPHALFOPEN** • **Gatewayicmp**
Send String	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. To retrieve a specific page from a website, specify a fully qualified path name.
Receive String	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. Click  to add the provided VIP details for a device.
    These details are created as scenario_1. You can add VIP details for another device by repeating the above procedure.
11. Once the VIP details are created for the desired devices, click **Submit**.
    The **Confirmation** dialog box is displayed.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

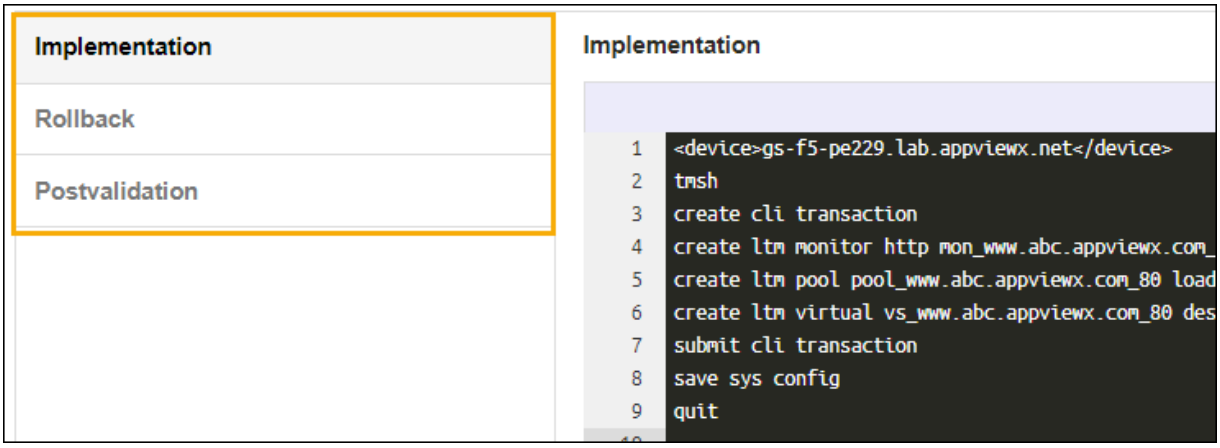
    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Review** stage.

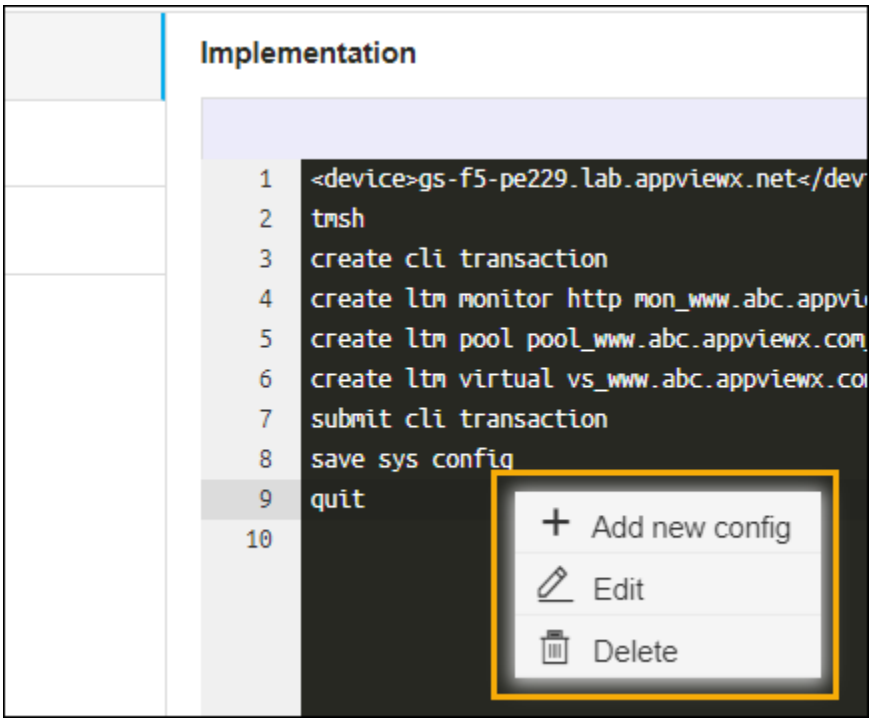
    Note:
    - If there is any failure at this stage, start a new request by providing the correct data.
    - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

13. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



14. (Optional) If you need to change any data at this stage, right click and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



15. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

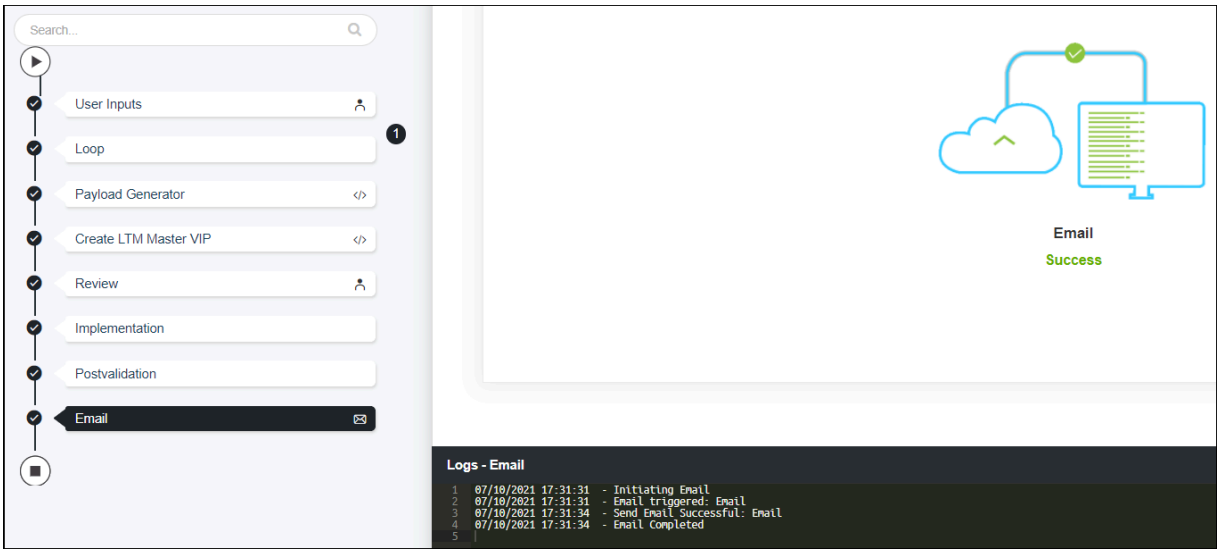
> **Note:**
> To stop running the workflow creation, click **Cancel**.

16. Click **Ok** to continue the workflow creation.

    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Create F5 LTM Monitors - Multiple Devices

This workflow creates a Single Monitor of Type (HTTP, HTTPS, GatewayICMP) with the same configuration on multiple devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM Monitors - Multiple Devices** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 LTM Monitors - Multiple Devices :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired devices from the drop-down option.

6. Enter/select the **Monitor Details**.

Monitor Details Section- Field and Description Table

Field	Description
***Monitor Type**	Select the monitor type from the drop-down option.
***Name**	Enter the name of the monitor.
Description	Enter the description for the monitor.
Interval	The frequency at which the system issues the monitor check when either the resource is down or the status of the resource is unknown. Enter the value for the interval in seconds.
Timeout	The number of seconds the target has in which to respond to the monitor request. Enter the timeout value in seconds.
Send String	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. The default setting is GET /\r\n, which retrieves a default HTML file for a website. To retrieve a specific page from a website, specify a fully qualified path name
Receive String	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.
Reverse	Instructs the system to mark the target resource down when the test is successful. This setting is useful. For example, if the content on your website's home page is dynamic and changes frequently, you may want to set up a reverse ECV service check that looks for the string errors. A match for this string meant that the webserver

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	was down. To use this option, you must specify values for Send String and Receive String. • disabled - Specifies that the monitor does not operate in reverse mode. • enabled - Specifies that the monitor operates in reverse mode.
Receive Disable String	This setting works like a Receive String, except that the system marks the node or pool member disabled when its response matches Receive Disable String but not Receive String. To use this setting, you must specify both Receive Disable String and Receive String.
Username	Enter the user name, if the monitored target requires authentication.
Password	Enter the password, if the monitored target requires authentication.
Transparent	The transparent mode of the monitor. A monitor in transparent mode directs traffic through the associated pool members or nodes (usually a router or firewall) to the aliased destination (that is, it probes the Alias Address-Alias Service Port combination specified in the monitor). If the monitor cannot successfully reach the aliased destination, the pool member or node through which the monitor traffic was sent is marked down. • **disabled** - Specifies that the monitor does not operate in transparent mode. • **enabled** - Specifies that the monitor operates in transparent mode.
Alias Address	Enter an alias IP address for the monitor to check on behalf of the pools or pool members with which the monitor is associated.
Alias Service Port	Enter an alias port or service for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.

7. Click **Submit**.
   The **Confirmation** dialog box is displayed.
8. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

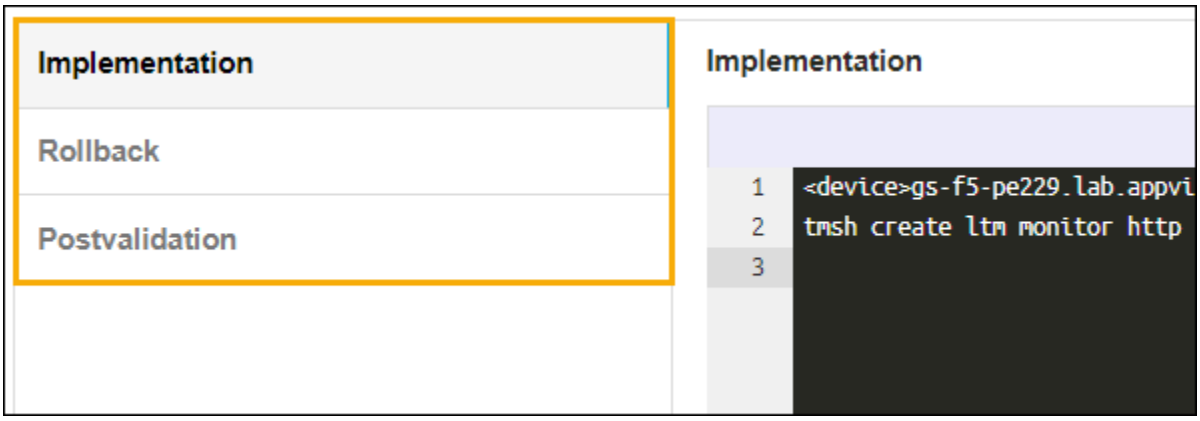
   The validation starts automatically and reaches the **Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

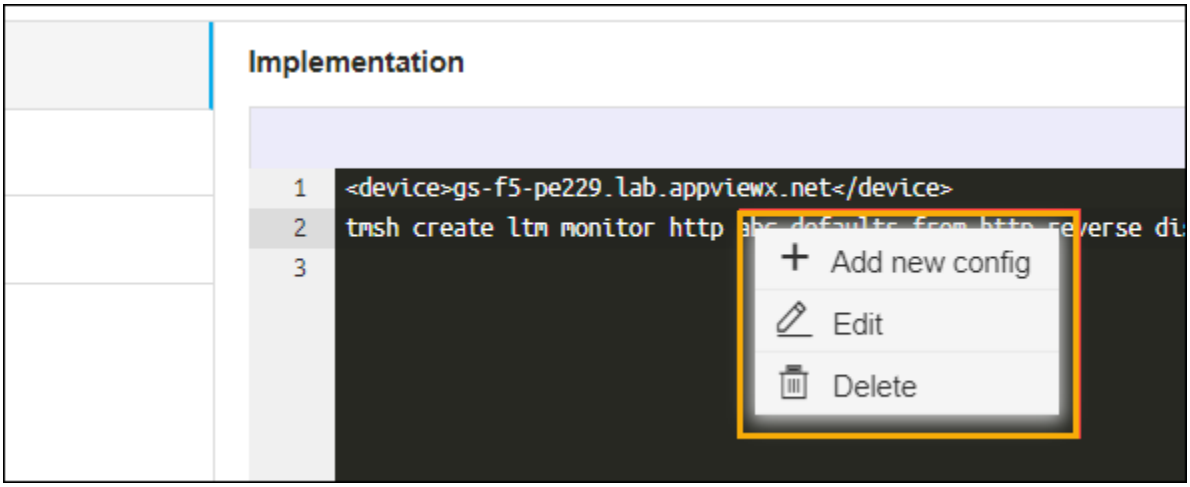
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



10. (Optional) If you need to change any data at this stage, right click and select the required option.



11. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

Note:

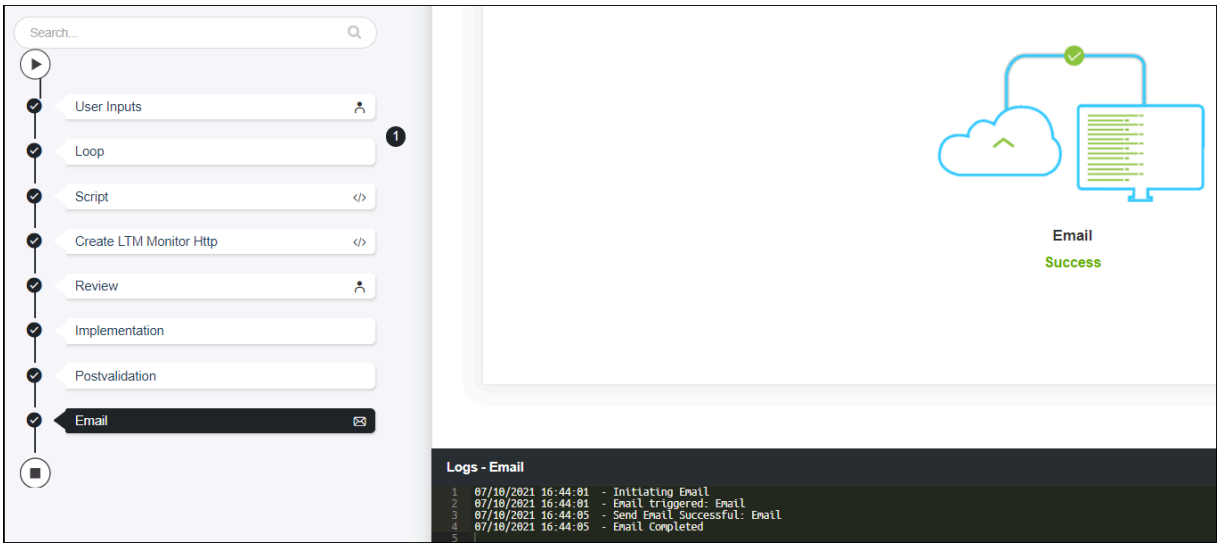
Click **Cancel** to stop running the workflow creation.

12. Click **Ok** to continue the workflow creation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

It takes a while to complete the request.

The workflow is created and the email is triggered to the configured email IDs.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Create F5 LTM Monitors

This workflow creates a Single Monitor of Type (HTTP, HTTPS, GatewayICMP) with all parameters on F5 device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP LTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM Monitors** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 LTM Monitors :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter/select the **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
***Monitor Type**	Select the monitor type from the drop-down option.
***Name**	Enter the name of the monitor.
Description	Enter the description for the monitor.
Interval	The frequency at which the system issues the monitor check when either the resource is down or the status of the resource is unknown. Enter the value for the interval in seconds.
Timeout	The number of seconds the target has in which to respond to the monitor request. Enter the timeout value in seconds.
Send String	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. The default setting is GET /\r\n, which retrieves a default HTML file for a website. To retrieve a specific page from a website, specify a fully qualified path name
Receive String	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Reverse	Instructs the system to mark the target resource down when the test is successful. This setting is useful. for example, if the content on your website home page is dynamic and changes frequently, you may want to set up a reverse ECV service check that looks for the string errors. A match for this string meant that the webserver was down. To use this option, you must specify values for Send String and Receive String. • disabled - Specifies that the monitor does not operate in reverse mode. • enabled - Specifies that the monitor operates in reverse mode.
Receive Disable String	This setting works like a Receive String, except that the system marks the node or pool member disabled when its response matches Receive Disable String but not Receive String. To use this setting, you must specify both Receive Disable String and Receive String.
Username	Enter the user name, if the monitored target requires authentication.
Password	Enter the password, if the monitored target requires authentication.
Transparent	The transparent mode of the monitor. A monitor in transparent mode directs traffic through the associated pool members or nodes (usually a router or firewall) to the aliased destination (that is, it probes the Alias Address-Alias Service Port combination specified in the monitor). If the monitor cannot successfully reach the aliased destination, the pool member or node through which the monitor traffic was sent is marked down. • **disabled** - Specifies that the monitor does not operate in transparent mode. • **enabled** - Specifies that the monitor operates in transparent mode.
Alias Address	Enter an alias IP address for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.
Alias Service Port	Enter an alias port or service for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.

7. Click **Submit**.
   The **Confirmation** dialog box is displayed.
8. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

OR

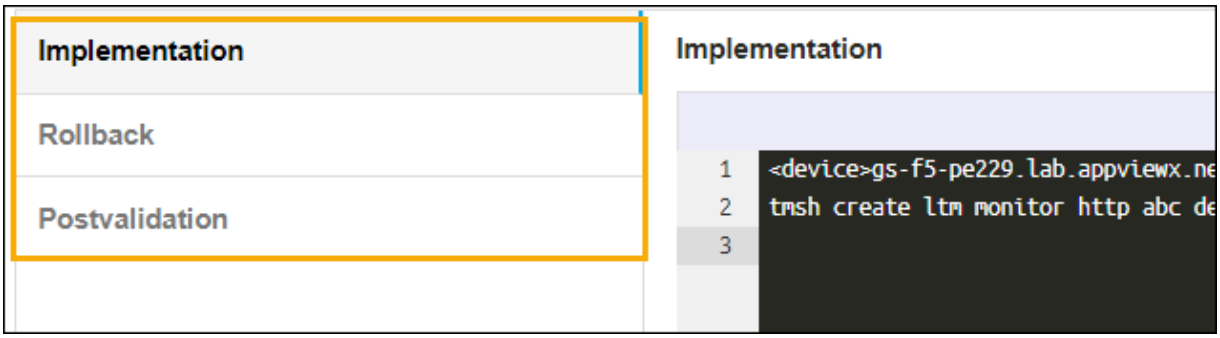
To submit the form, click **OK**.

The validation starts automatically and reaches the **Review** stage.

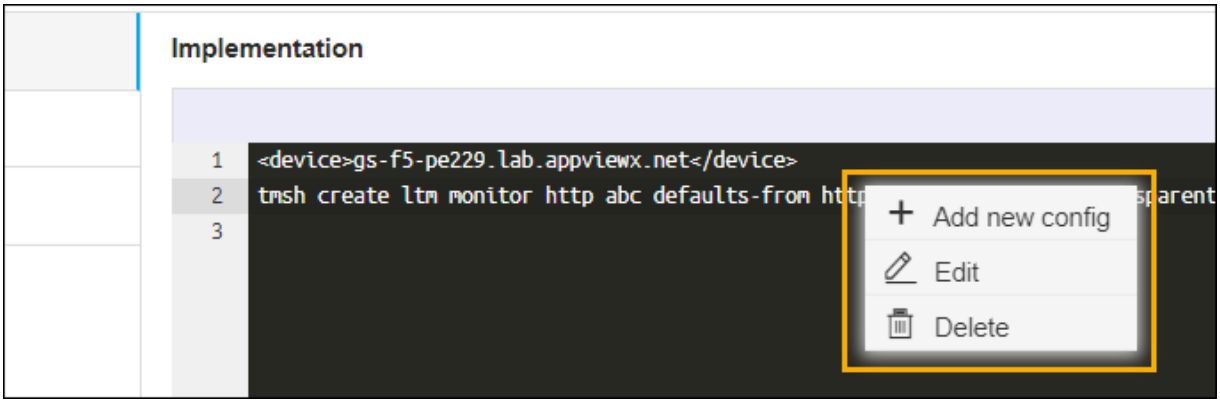
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



10. (Optional) If you need to change any data at this stage, right click and select the required option.



11. After the review, click **Submit**.

The **Confirmation** dialog box is displayed.

Note:

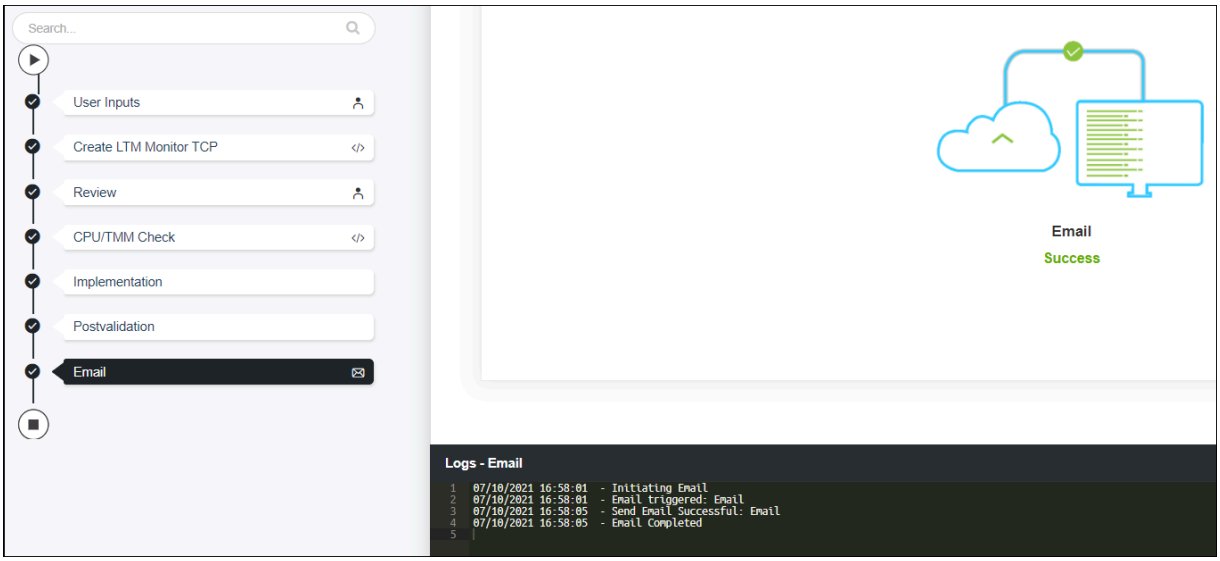
To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

It takes a while to complete the request.

The workflow is created and the email is triggered to the configured email IDs.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

F5 BIG-IP GTM

Create F5 GTM WideIP Basic

This workflow generates a GTM WideIP with a single pool for a device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 GTM WideIP Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 GTM WideIP Basic :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

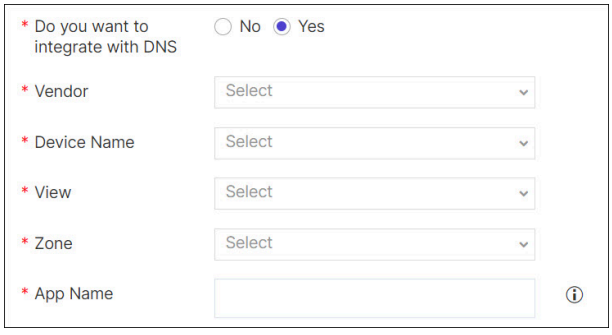
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Details Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Partition**	Partition in which objects will be created. If you want to create a new partition, then click **Create New Partition** from the dropdown list and then provide a partition name.

6. Enter/select the **DNS Details**.

DNS Details Section - Field and Description Table

Field	Description
***Do you want to integrate with DNS**	By default, the **No** option is selected. If you want to integrate with DNS for CNAME record creation, select **Yes**. When you select **Yes**, the following fields are displayed:  • **Vendor** - Select the DNS provider or the technology/service from the drop-down list to manage DNS records. The default vendors are: **Infoblox** and **bluecat**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **Device Name** - Select the **Device Name** from the drop-down list associated with the DNS record. • **View** - Select a view to which a DNS record belongs from the dropdown list. • **Zone** - Select a zone associated with a specific domain or subdomain from the dropdown list. • **App Name**- Enter the an application name associated with the DNS record. Ex: For the WideIP, name in the format www.<name>.com. **Note:** Avoid appending the zone at the end of the app name; it will be automatically included based on the previously specified zone. When you select DNS vendor as **bluecat**, you need to provide the following details:  If you select **bulecat** as the vendor, choose the configuration from the dropdown list, in addition to the previously specified details.

7. Enter/select the **General Properties**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

General Properties Section - Field and Description Table

Field	Description
***Wide IP Name**	Enter the WideIP Name in the FQDN format. For example, <yyy>.www.<name>.com.
***Type**	Select the pool/WideIP type. The types are: • **A** - the pool/Wide IP responds to A queries. The A pool/Wide IP is a mapping of an FQDN to a set of IPv4 virtual servers that host the domain's content, such as a Web site or an e-commerce site. You can also specify pool members' virtual server and ratio settings. • **CNAME** - the pool/Wide IP responds to CNAME queries. A CNAME pool/Wide IP is a mapping of an FQDN to its canonical name. This is the configuration of a static name or a Wide IP name. Static names support older configurations. Wide IP names allow service checking, enhanced load balancing, and CNAME chasing in the reply. You can also specify pool members' ratio.
Description	Enter the descriptive text to identify the WideIP.
State	It allows the system to use this wide IP and its resources for load balancing. The options are: • **enabled** (default) • **disabled**
***Do you want to add Alias ?**	Add Alias for the WideIP name. The options are: • **No** (default) • **Yes** – When you select **Yes**, the following fields are displayed to add alias details: Enter the alias name for WideIP in the Alias field. You can add any number of aliases for WideIP. After entering the alias in the Alias field, click  to add an alias. The added aliases are displayed under the Alias List. You can manage the added alias(es) to the list.

8. Enter/select the **Pool** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
The pool will be created with the name **pool_<wideip_name>**.

Pool Section - Field and Description Table

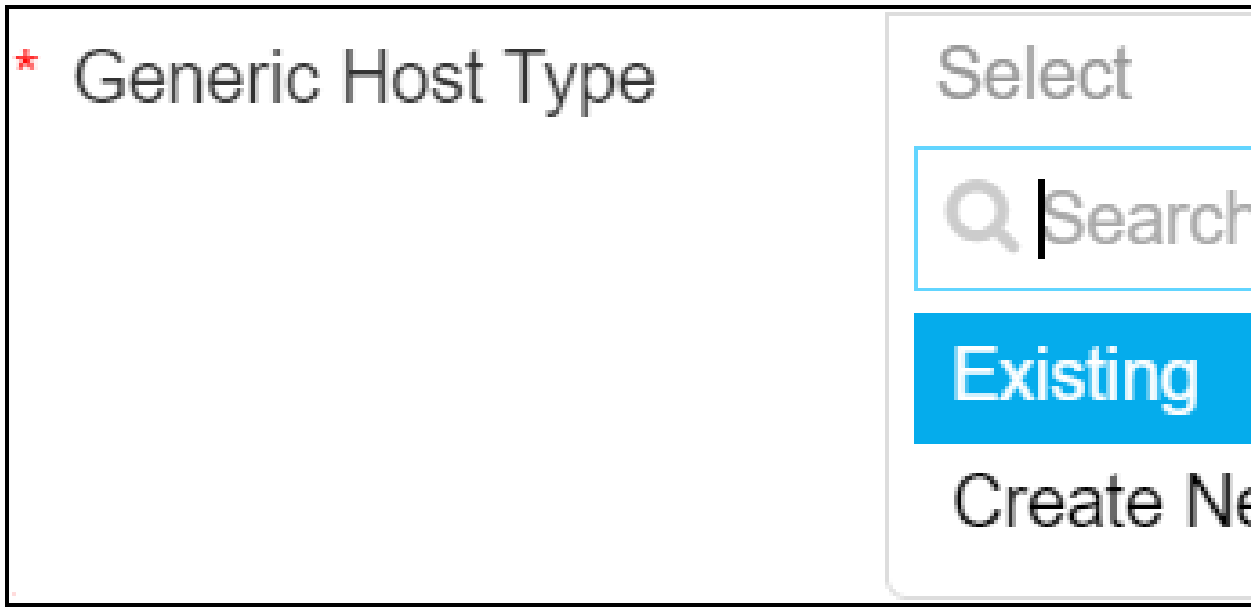
Field	Description
***Load Balancing Method**	The load balancing method is used to select a pool in this WideIP. The methods are: • **round-robin** - the system selects the pools sequentially. • **global-availability** -. the system selects a pool by following the order of the Pool List. The system repeatedly selects the first pool in the list for as long as its status is available. If the pool becomes unavailable for any reason, the system then repeatedly selects the next pool in the list, and so on. • **ratio** - the system selects a pool based on the ratio that you assign to the pool.
***Persistence**	By default, this option is disabled. When, a local DNS makes repetitive requests on behalf of a client, the system reconnects the client to the same resource as previous requests.
Preferred	Select the preferred load balancing method. The system tries this method first. **Note:** When you choose the **fallback-ip** option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Alternate	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP. **Note:** When you choose the **fallback-ip** option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Fallback	Select the alternate load balancing method. The system tries this method if the Preferred and Alternate methods are unsuccessful in picking the WideIP.
Fallback IP	When the **fallback-ip** option is selected as Preferred or Alternate load balancing method, this field will be displayed to add the Fallback IP. The **Fallback IP** is not applicable for the pool type **CNAME**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

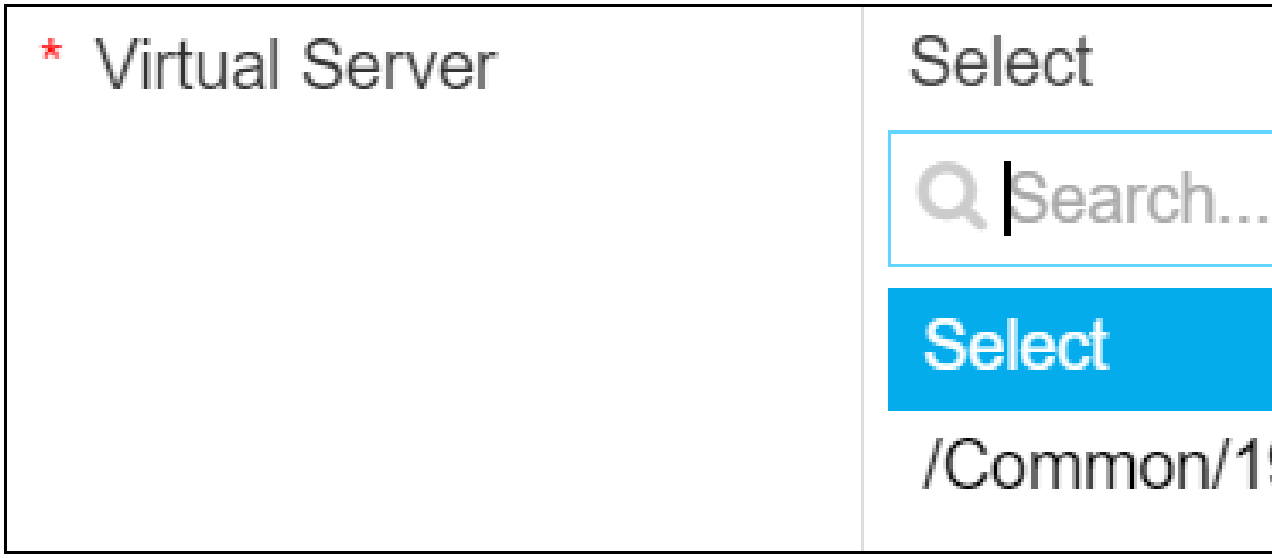
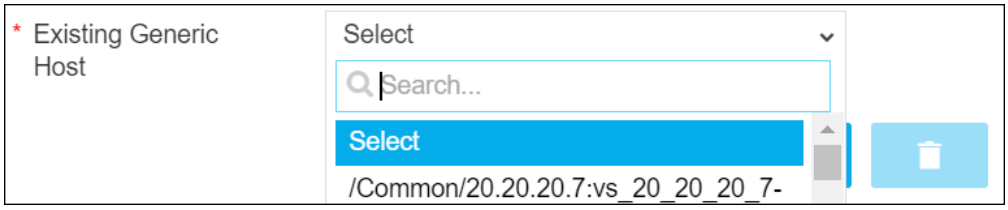
Field	Description
Health Monitor	The health monitors that the system uses to determine whether it can use this pool for load balancing. To get the fields to be monitored, click the **Retrieve** button, and then select the

9. Enter/select the **Pool Member Details**.

Pool Member Details Section - Field and Description Table

Field	
***Pool Member Type**	Select the pool member types. The pool member types are: • **Generic Host** – When you select this pool member type, you get the option to select the Generic Host Typ  • **Big IP System** – When you select this pool member type, you get the option to select the configured virtua

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
	 **Note:** It takes a while to load the servers.
***Generic Host Type**	When the pool member type is selected as Generic Host, you need to select the type of generic host. The g • **Existing** – This option allows to select the existing host. When you select this option, the existing generic  • **Create New** – This option allows you to create a generic host. When you select this, the following fields ar • ***Data Center** – select the datacenter of the pool member. • ***Pool Member IP** – enter the IP of the pool. • ***Port** – enter the port of the pool member.
***Virtual Server**	When the pool member type is selected as Big IP System. Select a virtual server from the dropdown list.
***Pool Member List**	Enter the pool member details, and then click the Add  button. The pool member details are a

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
	Note: For Pool with type CNAME, pool members can be either existing wide IP or a static wide IP.

10. Click **Submit**.
    The **Confirmation** dialog box is displayed.
11. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

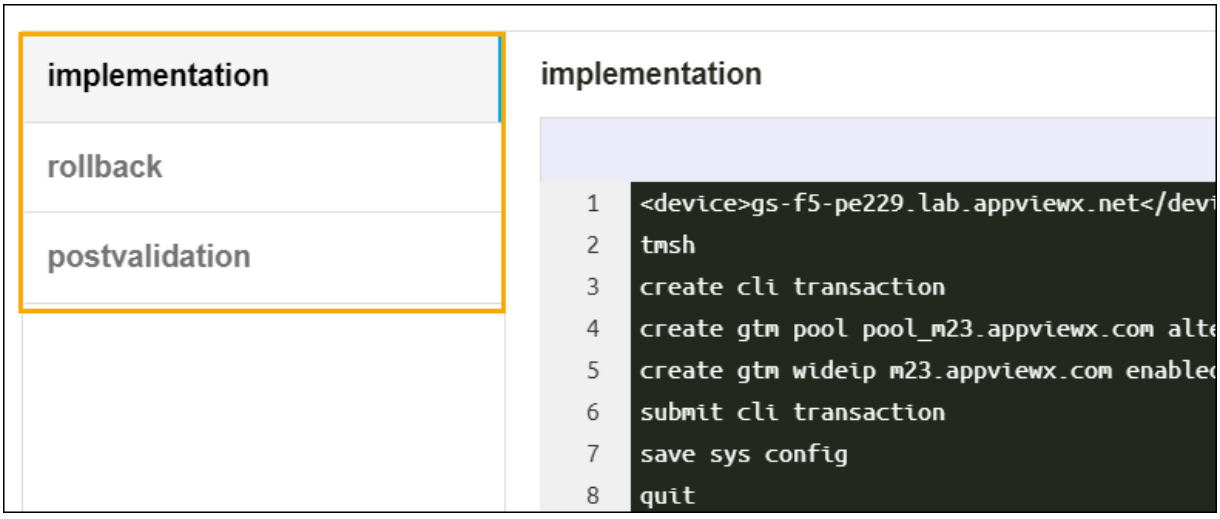
    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Review** stage.

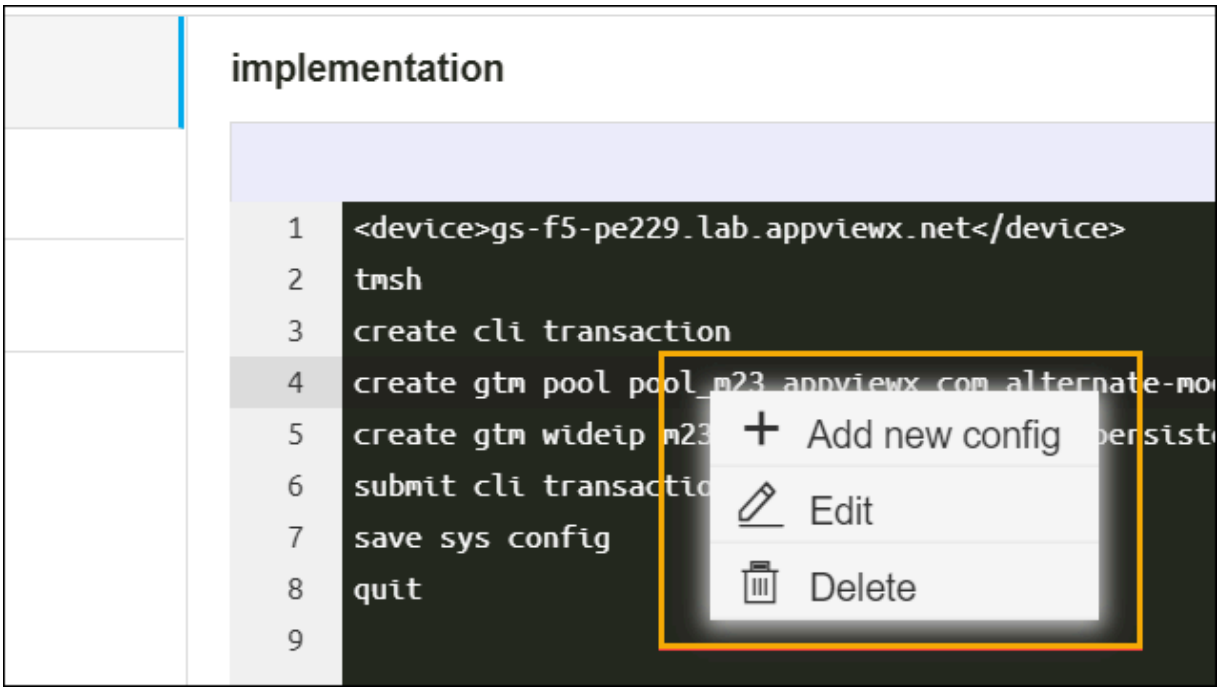
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

12. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



13. (Optional) If you need to change any data at this stage, right click and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



14. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

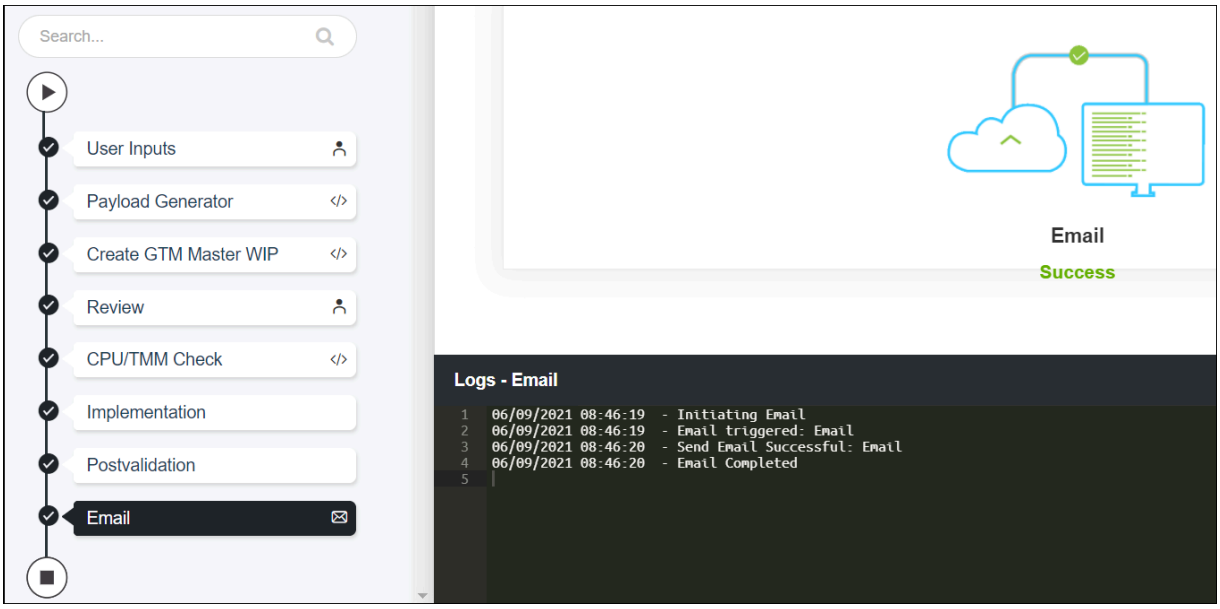
> **Note:**
> To stop running the workflow creation, click **Cancel**.

15. Click **Ok** to continue the workflow creation.

    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Create F5 GTM WideIP Advanced

This workflow generates a GTM WideIP with multiple pools for a device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 GTM WideIP Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 GTM WideIP Basic :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

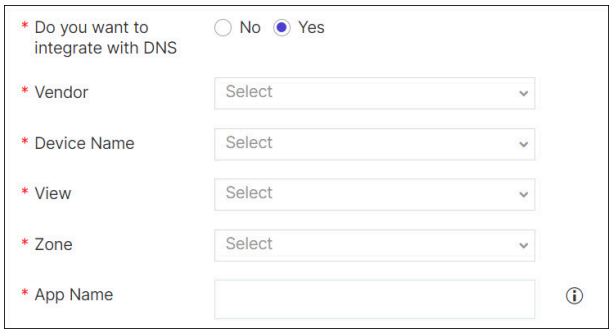
Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Partition**	Partition in which objects will be created. If you want to create a new partition, then click **Create New Partition** from the dropdown list and then provide a partition name.

6. Enter/select the **DNS Details**.

DNS Details Section - Field and Description Table

Field	Description
***Do you want to integrate with DNS**	By default, the **No** option is selected. If you want to integrate with DNS for CNAME record creation, select **Yes**. When you select **Yes**, the following fields are displayed:  • **Vendor** - Select the DNS provider or the technology/service from the drop-down list to manage DNS records. The default vendors are: **Infoblox** and **bluecat**. • **Device Name** - Select the **Device Name** from the drop-down list associated with the DNS record.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **View** - Select a view to which a DNS record belongs from the dropdown list. • **Zone** - Select a zone associated with a specific domain or subdomain from the dropdown list. • **App Name**- Enter the an application name associated with the DNS record. Ex: For the WideIP, name in the format www.<name>.com. **Note:** Avoid appending the zone at the end of the app name; it will be automatically included based on the previously specified zone. When you select DNS vendor as **bluecat**, you need to provide the following details:  If you select **bulecat** as the vendor, choose the configuration from the dropdown list, in addition to the previously specified details.

7. Enter/select the **General Properties**.

General Properties Section - Field and Description Table

Field	Description
***Wide IP Name**	Enter the WideIP Name in the FQDN format. For example, <yyy>.www.<name>.com.
***Type**	Select the pool/WideIP type. The types are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **A** - the pool/Wide IP responds to A queries. The A pool/Wide IP is a mapping of an FQDN to a set of IPv4 virtual servers that host the domain's content, such as a Web site or an e-commerce site. You can also specify pool members' virtual server and ratio settings. • **CNAME** - the pool/Wide IP responds to CNAME queries. A CNAME pool/Wide IP is a mapping of an FQDN to its canonical name. This is the configuration of a static name or a Wide IP name. Static names support older configurations. Wide IP names allow service checking, enhanced load balancing, and CNAME chasing in the reply. You can also specify pool members' ratio.
Description	Enter the descriptive text to identify the WideIP.
State	It allows the system to use this wide IP and its resources for load balancing. The options are: • **enabled** (default) • **disabled**
***Wide IP LB Method**	Select the load balancing method from the dropdown list.
***Wide IP Persistence**	Select one of the following options: • **Disabled** (default) - This option turns off Wide IP persistence for the specified Wide IP. In this mode, requests from the same client may be directed to different servers within the server pool without considering previous sessions. • **Enabled** - When Wide IP persistence is enabled, the load balancer ensures that subsequent requests from the same client are consistently directed to the same server within the server pool. This helps maintain session continuity and provides a predictable experience for users interacting with the application.
***Do you want to add Alias ?**	Add Alias for the WideIP name. The options are: • **No** (default) • **Yes** – When you select **Yes**, the following fields are displayed to add alias details:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	Enter the alias name for WideIP in the Alias field. You can add any number of aliases for WideIP. After entering the alias in the Alias field, click  to add an alias. The added aliases are displayed under the Alias List. You can manage the added alias(es) to the list.

8. Enter/select the **Pool** details.

> **Note:**
> The pool will be created with the name **pool_<wideip_name>**.

Pool Section - Field and Description Table

Field	Description
***Pool Name**	Enter the name of the pool.
Preferred	Select the preferred load balancing method. The system tries this method first. **Note:** When you choose the **fallback-ip** option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Alternate	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP. **Note:** When you choose the **fallback-ip** option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Fallback	Select the alternate load balancing method. The system tries this method if the Preferred and Alternate methods are unsuccessful in picking the WideIP.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

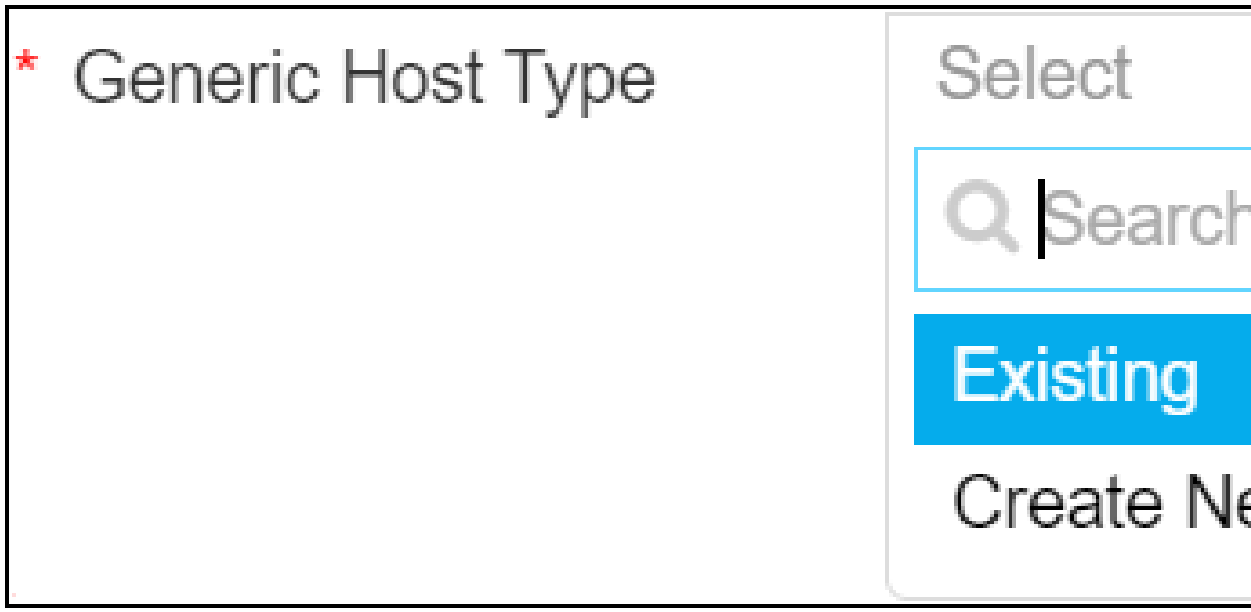
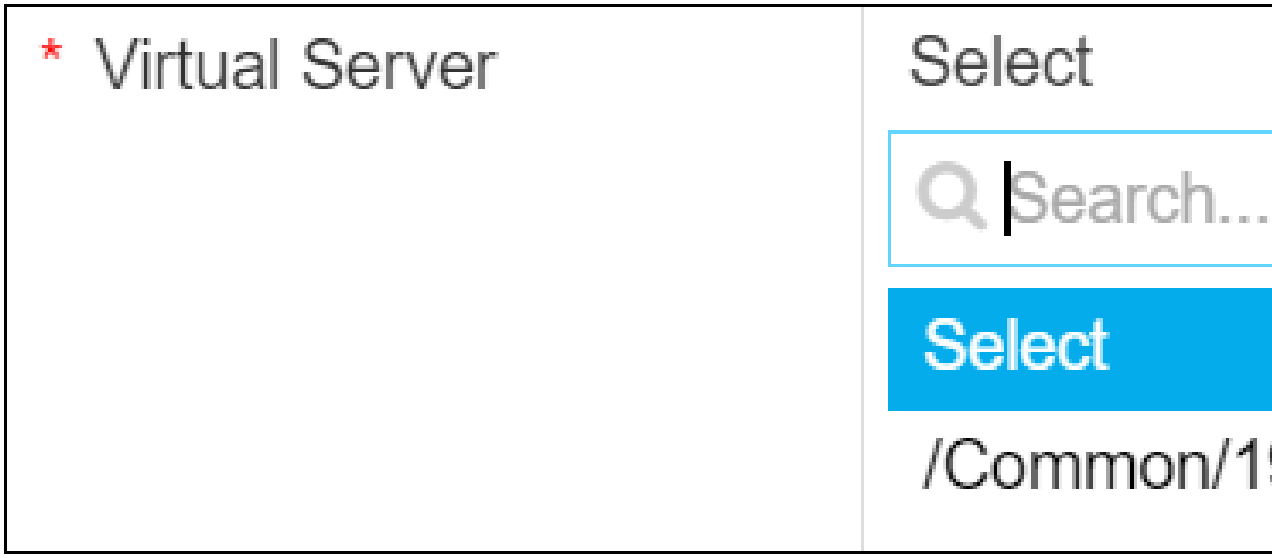
Field	Description
Fallback IP	When the **fallback-ip** option is selected as Preferred or Alternate load balancing method, this field will be displayed to add the Fallback IP. The **Fallback IP** is not applicable for the pool type **CNAME**.
***Choose Monitor**	Choose from the following options: • **No**: Select this option if you do not want to associate a health monitor with the current configuration. • **Create New**: Select this option to create a new health monitor specifically for the current configuration. • **Use Existing**: Select this option if you want to use an existing health monitor that has been previously configured.
***Pools**	Click  to add the pool details to the list. You can add multiple to pool to the list. After adding pool(s) to the list, you can manage them.

9. Enter/select the **Pool Member Details**.

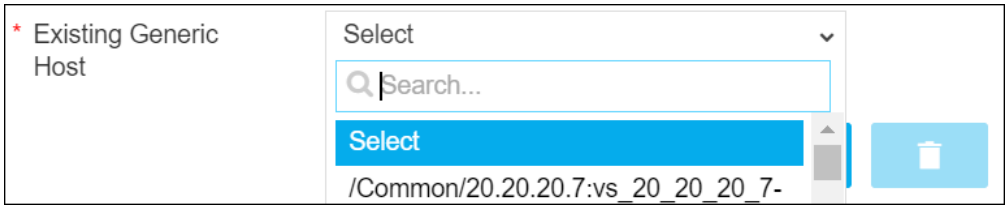
Pool Member Details Section - Field and Description Table

Field	
***Pool**	Click the  (**Retrieve**), and then select a pool from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
***Pool Member Type**	Select the pool member types. The pool member types are: • **Generic Host** – When you select this pool member type, you get the option to select the Generic Host Typ  • **Big IP System** – When you select this pool member type, you get the option to select the configured virtua  **Note:** It takes a while to load the servers.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
***Generic Host Type**	When the pool member type is selected as Generic Host, you need to select the type of generic host. The g • **Existing** – This option allows to select the existing host. When you select this option, the existing generic  • **Create New** – This option allows you to create a generic host. When you select this, the following fields ar • ***Data Center** – select the datacenter of the pool member. • ***Pool Member IP** – enter the IP of the pool. • ***Port** – enter the port of the pool member.
***Pool Member List**	Enter the pool member details, and then click the Add  button. The pool member details are a **Note:** For Pool with type CNAME, pool members can be either existing wide IP or a static wide IP.

10. On the **Validation Details** section, click **Form Validation**.
    The validation starts and the results are displayed in the Validation Status box.
11. Click **Submit**.
    The **Confirmation** dialog box is displayed.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

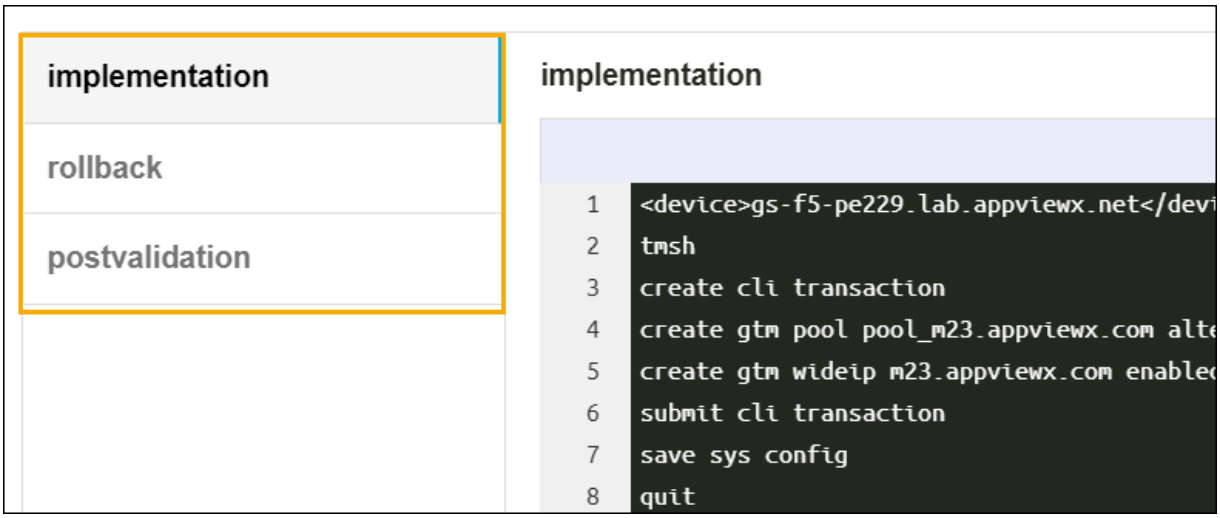
    The validation starts automatically and reaches the **Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

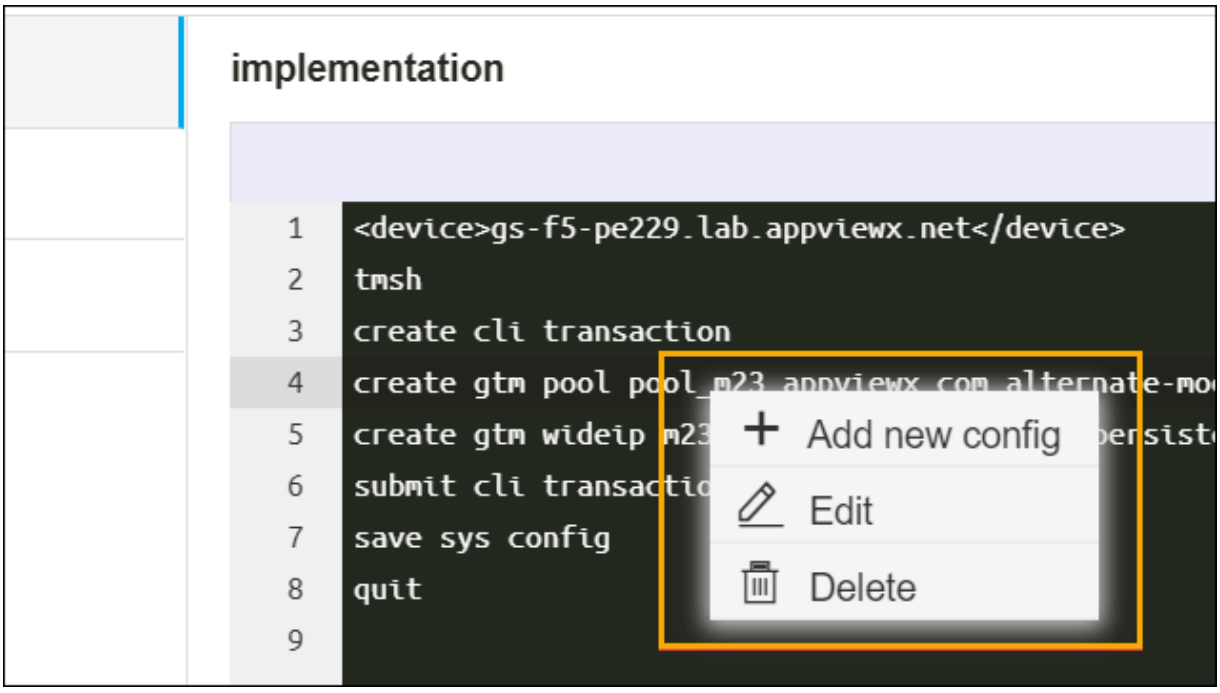
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

13. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



14. (Optional) If you need to change any data at this stage, right click and select the required option.



15. After the review, click **Submit**.

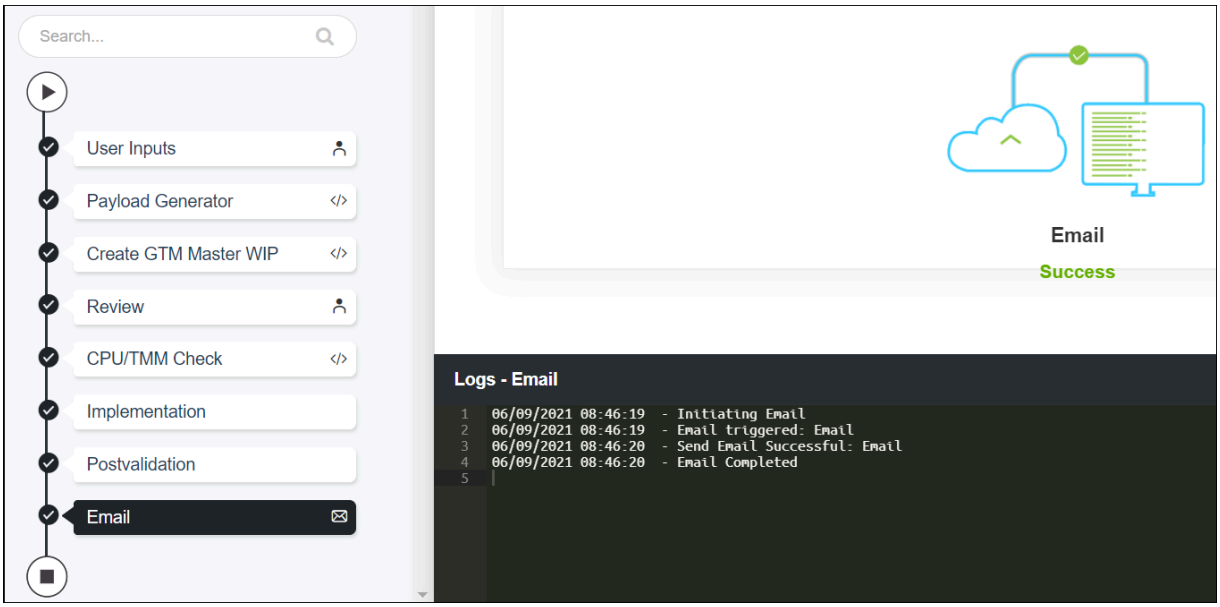
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

16. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.



> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Create F5 GTM WideIP with Topology

This workflow creates a WideIP with multiple Pools and chooses a topology record for the pools that are created.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 GTM WideIP with Topology** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click Run.

   The **Request > Create F5 GTM WideIP with Topology :: FormBuilder** page is displayed.

5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

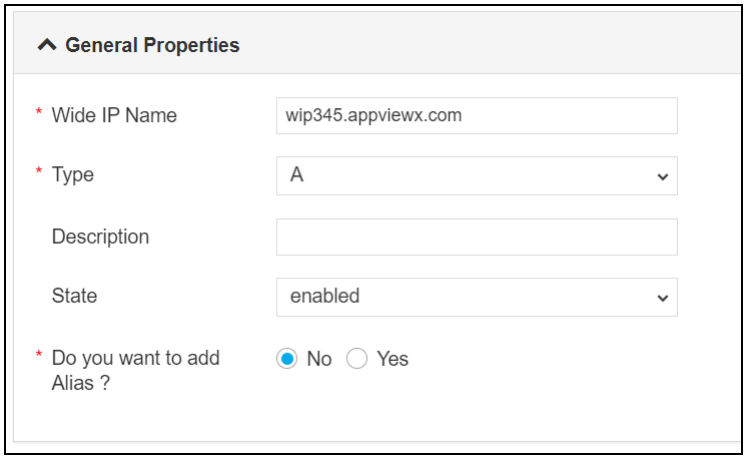
Field	Description
Datacenter	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter/select the **DNS Details**.

DNS Details Section - Field and Description Table

Field	Description
***Do you want to integrate with DNS**	By default, the No option is selected. If you want to integrate with DNS for CNAME record creation, select Y When you select Yes, the following fields are displayed:  Select the Vendor from the drop-down list. The default vendors are **Infoblox** and **bluecat**. When you select DNS vendor as **Infoblox**, you need to provide the following details: • Select the **Device Name** from the drop-down list. • Enter the **App Name** for the WideIP name in the format www.<name>.com. When you select DNS vendor as **bluecat**, you need to provide the following details:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

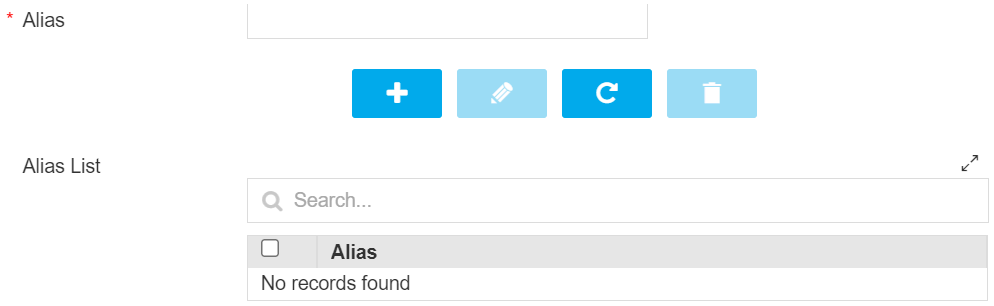
Field	Description
	 • Select the **Device Name** from the drop-down list. • Select the **Configuration** from the drop-down list. • Select the **View** for the selected configuration. The options are displayed based on the selected configura • Enter the **App Name** for the WideIP name in the format of www.<name>.com.

7. Enter/select the **General Properties**.

General Properties Section - Field and Description Table

Field	
***Wide IP Name**	Enter the WideIP Name in the FQDN format.
***Type**	Select the pool/WideIP type. The types are: • A - the pool/Wide IP responds to A queries. The A pool/Wide IP is a mapping of an FQDN to a set of IP • CNAME - the pool/Wide IP responds to CNAME queries. A CNAME pool/Wide IP is a mapping of an F
Description	Enter the descriptive text to identify the WideIP.
State	It allows the system to use this wide IP and its resources for load balancing. The options are: • enabled (default) • disabled
***Do you want to add Alias ?**	Add Alias for the WideIP name. The options are: • **No** (default) • **Yes** – When you select Yes the following fields are displayed to add alias details:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
	 Enter the alias name for WideIP in the Alias field. You can add any number of aliases for WideIP. After

8. Enter/select the **Pool** details.

Pool Section - Field and Description Table

Field	Description
Info	Information that is to be noted while entering the details in this section.
***Load Balancing Method**	The load balancing method is used to select a pool in this WideIP. The methods are: • round-robin - the system selects the pools sequentially. • global-availability -. the system selects a pool by following the order of the Pool List. The system repeatedly selects the first pool in the list for as long as its status is available. If the pool becomes unavailable for any reason, the system then repeatedly selects the next pool in the list, and so on. • ratio - the system selects a pool based on the ratio that you assign to the pool.
***Persistence**	By default, this option is disabled. When a local DNS makes repetitive requests on behalf of a client, the system reconnects the client to the same resource as previous requests.
***Pool Name**	Enter the pool name.
Preferred	Select the preferred load balancing method. The system tries this method first.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

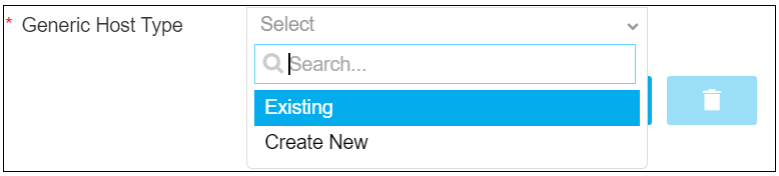
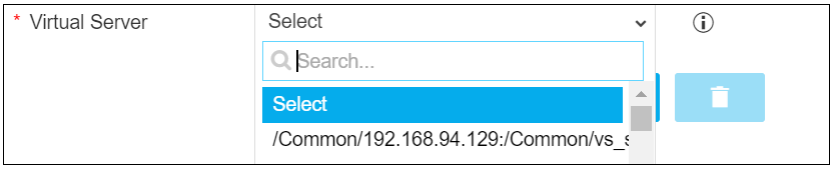
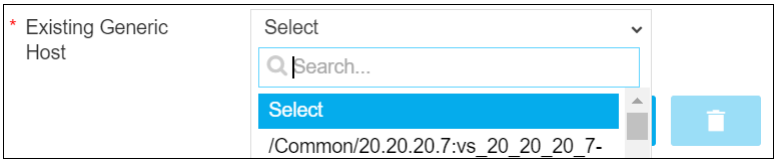
Field	Description
	Note: When you choose the fallback-ip option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Alternate	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP. **Note:** When you choose the fallback-ip option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Fallback	Select the alternate load balancing method. The system tries this method if the Preferred and Alternate methods are unsuccessful in picking the WideIP.
Fallback IP	When the fallback-ip option is selected as Preferred or Alternate load balancing method, this field will be displayed to add the Fallback IP. The Fallback IP is not applicable for the pool type CNAME.
Health Monitor	The health monitors that the system uses to determine whether it can use this pool for load balancing. To retrieve the fields to be monitored, click  , and then select the
Pools	Click  to add the pool details to the list. You can add multiple to the pool to the list. After adding pool(s) to the list, you can manage them.

9. Enter/select the **Pool Member Details**.

Pool Member Details Section - Field and Description Table

Field	Description
Info	Information that is to be noted while entering the details in this section.
Pool	Click the Retrieve () button to get the pool name.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Pool Member Type	Select the pool member types. The pool member types are: • **Generic Host** – When you select this pool member type, you get the option to select the Generic Host Typ  • **Big IP System** – When you select this pool member type, you get the option to select the configured virtua  **Note:** It takes a while to load the servers.
***Generic Host Type**	When the pool member type is selected as Generic Host, you need to select the type of generic host. The g • **Existing** – This option allows to select the existing host. When you select this option, the existing generic  • **Create New** – This option allows you to create a generic host. When you select this, the following fields ar • ***Data Center** – select the datacenter of the pool member. • ***Pool Member IP** – enter the IP of the pool. • ***Port** – enter the port of the pool member.
***Virtual Server**	When the pool member type is selected as Big IP System. Select a virtual server from the dropdown list.
***Pool Member List**	Enter the IP address of the pool member. And then click  . Any number of pool numbers can b

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. Enter/select the **Topology** details.

Topology Section - Field and Description Table

Field	Description
Request Source (Region)	The origination section of the topology record for the pool. Click  to retrieve the list of associated regions.
Destination (Pool)	The target section of the topology record for the pool.
Weight	The weight of the topology record. The system finds the weight of the first topology record that matches the server object (pool or pool member) and the local DNS. The system then assigns that weight as the topology score for that server object. The system load balances to the server object with the highest topology score. If the system finds no topology record that matches both the server object and the local DNS, then the system assigns that server object a zero score.
Topology Record Detail …	After providing the topology details for the pool, click  . The List of topologies is added for the pool. You can manage topology records after adding them.

11. Click **Submit**.
    The **Confirmation** dialog box is displayed.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

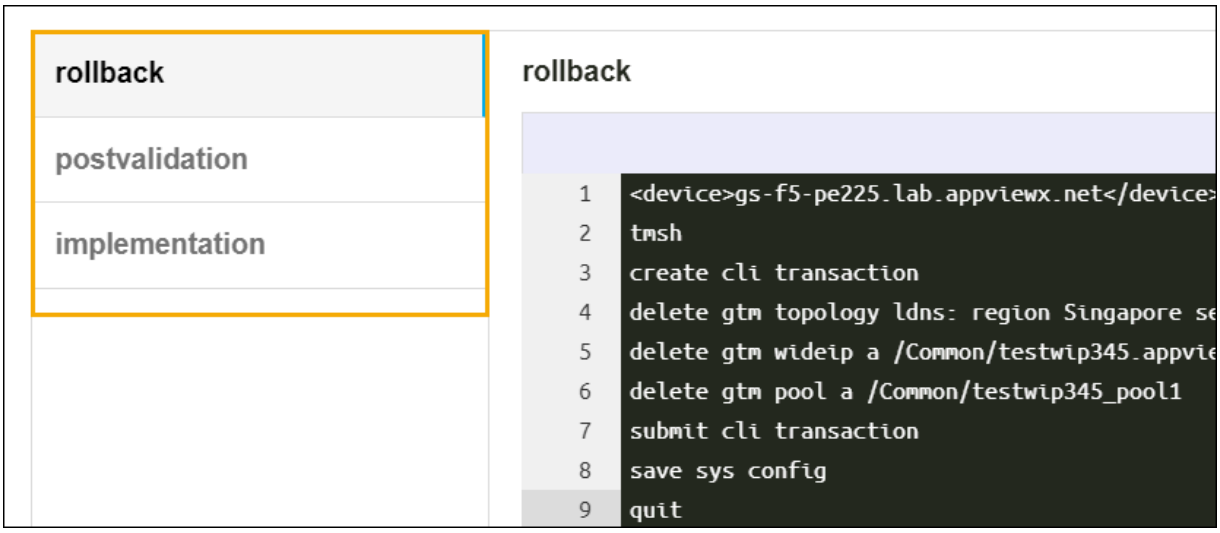
    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Review** stage.

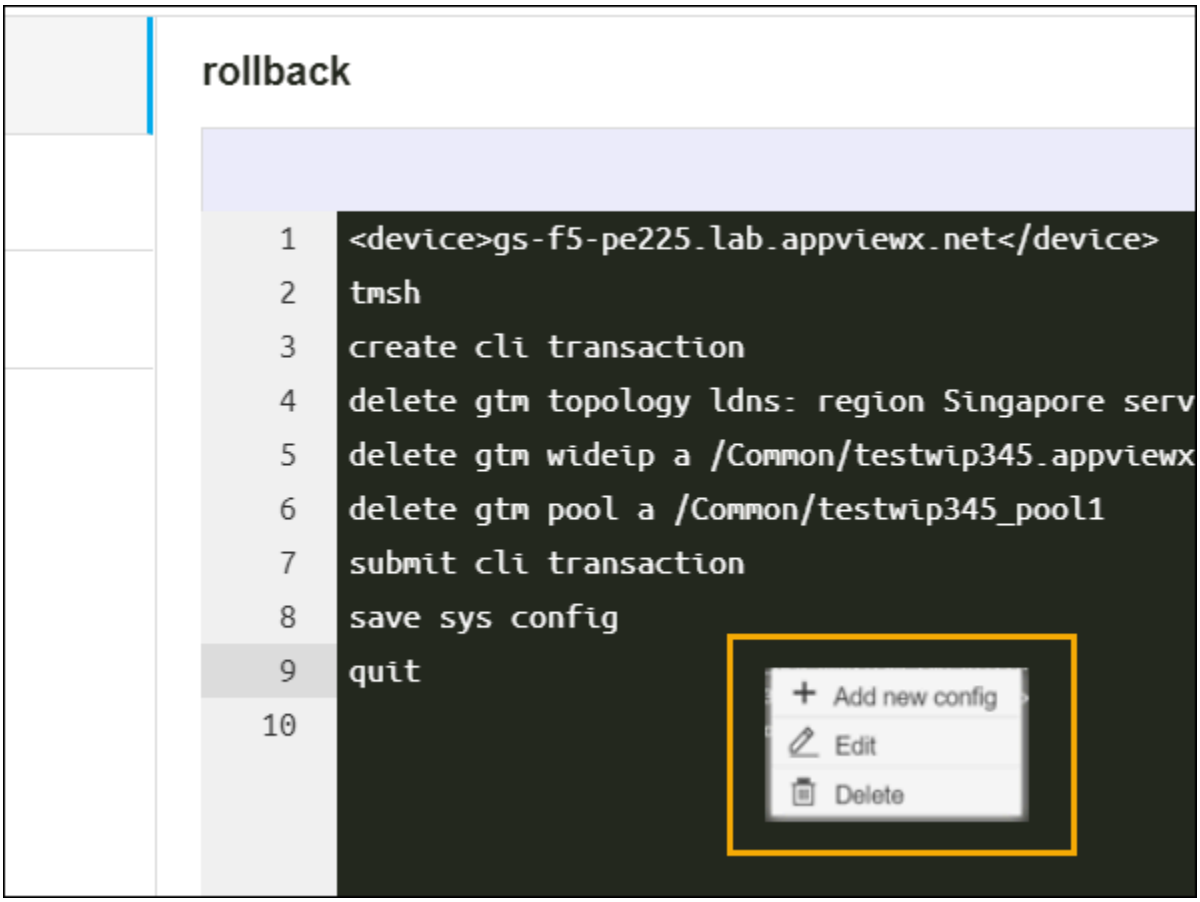
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

13. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



14. (Optional) If you need to change any data at this stage, right click and select the required option.



15. After the review, click **Submit**.

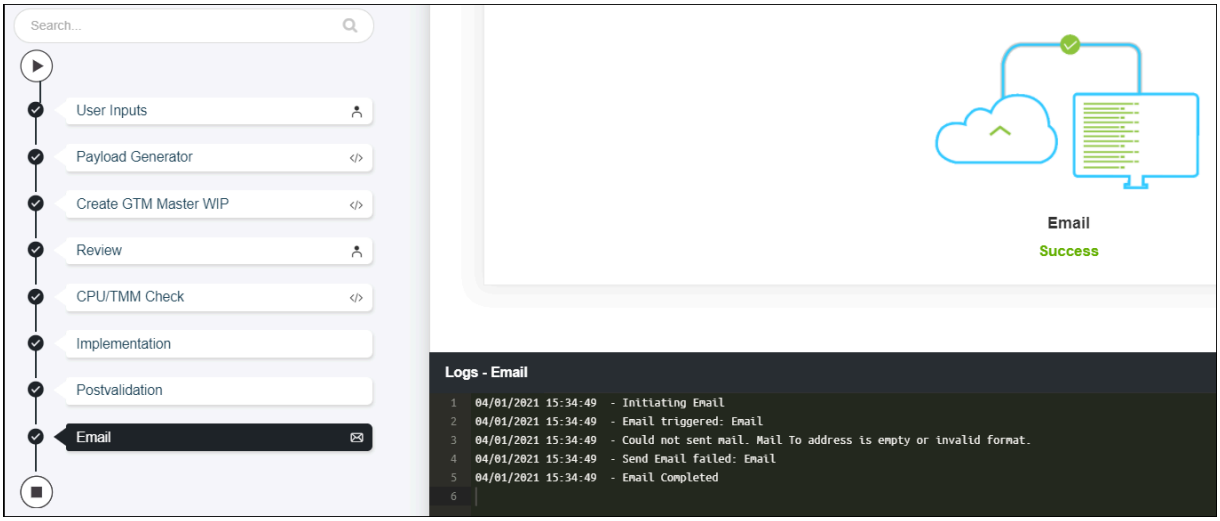
    The **Confirmation** dialog box is displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
To stop running the workflow creation, click **Cancel**.

16. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    After a successful post-validation, the workflow is created and the email triggered to the configured email IDs.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Create F5 GTM WideIP with Topology and Patition

This workflow creates a WideIP with multiple Pools and chooses a topology record for the pools that are created.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 GTM WideIP with Topology and Partition** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click Run.

   The **Request > Create F5 GTM WideIP with Topology and Partition :: FormBuilder** page is displayed.

5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

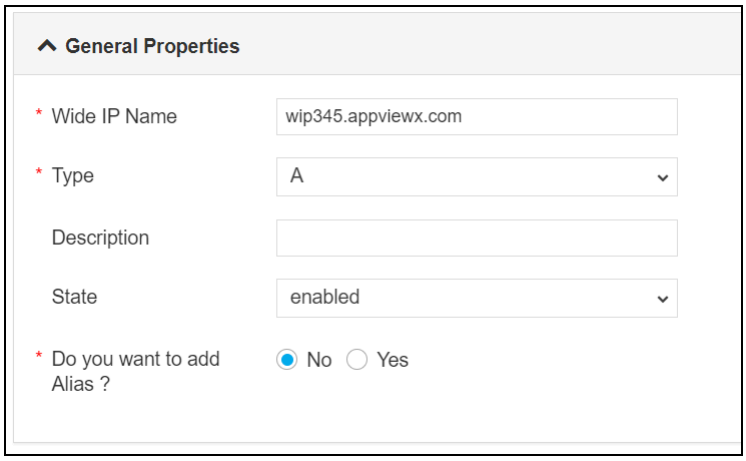
Field	Description
Datacenter	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Partition**	Select Partition from the dropdown list on which WideIP has to be created.

6. Enter/select the **DNS Details**.

DNS Details Section - Field and Description Table

Field	Description
***Do you want to integrate with DNS**	By default, the No option is selected. If you want to integrate with DNS for CNAME record creation, select Y When you select Yes, the following fields are displayed:  Select the Vendor from the drop-down list. The default vendors are **Infoblox** and **bluecat**. When you select DNS vendor as **Infoblox**, you need to provide the following details: • Select the **Device Name** from the drop-down list. • Enter the **App Name** for the WideIP name in the format www.<name>.com.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

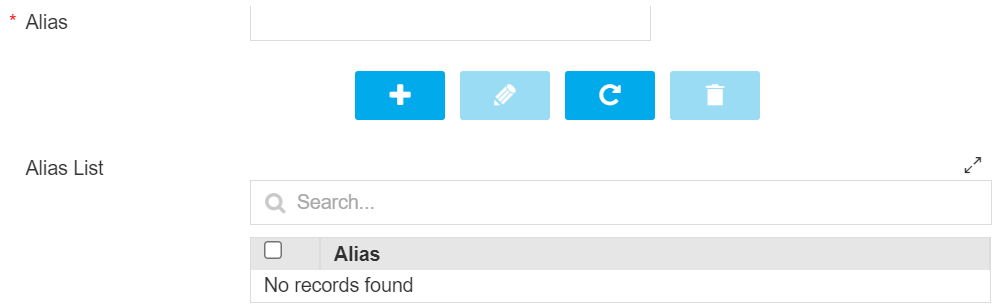
Field	Description
	When you select DNS vendor as **bluecat**, you need to provide the following details:  • Select the **Device Name** from the drop-down list. • Select the **Configuration** from the drop-down list. • Select the **View** for the selected configuration. The options are displayed based on the selected configura • Enter the **App Name** for the WideIP name in the format of www.<name>.com.

7. Enter/select the **General Properties**.

General Properties Section - Field and Description Table

Field	
***Wide IP Name**	Enter the WideIP Name in the FQDN format.
***Type**	Select the pool/WideIP type. The types are: • A - the pool/Wide IP responds to A queries. The A pool/Wide IP is a mapping of an FQDN to a set of IP • CNAME - the pool/Wide IP responds to CNAME queries. A CNAME pool/Wide IP is a mapping of an F
Description	Enter the descriptive text to identify the WideIP.
State	It allows the system to use this wide IP and its resources for load balancing. The options are: • enabled (default) • disabled

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
***Do you want to add Alias ?**	Add Alias for the WideIP name. The options are: • **No** (default) • **Yes** – When you select Yes the following fields are displayed to add alias details:  Enter the alias name for WideIP in the Alias field. You can add any number of aliases for WideIP. After

8. Enter/select the **Pool** details.

Pool Section - Field and Description Table

Field	Description
Info	Information that is to be noted while entering the details in this section.
***Load Balancing Method**	The load balancing method is used to select a pool in this WideIP. The methods are: • round-robin - the system selects the pools sequentially. • global-availability -. the system selects a pool by following the order of the Pool List. The system repeatedly selects the first pool in the list for as long as its status is available. If the pool becomes unavailable for any reason, the system then repeatedly selects the next pool in the list, and so on. • ratio - the system selects a pool based on the ratio that you assign to the pool.
***Persistence**	By default, this option is disabled. When a local DNS makes repetitive requests on behalf of a client, the system reconnects the client to the same resource as previous requests.
***Pool Name**	Enter the pool name.
Preferred	Select the preferred load balancing method. The system tries this method first.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

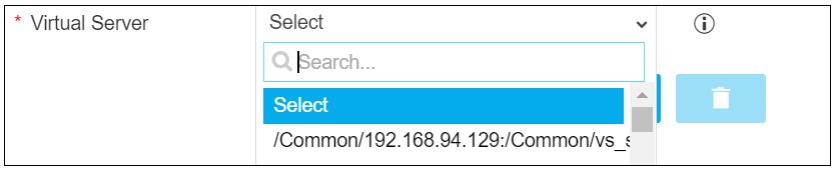
Field	Description
	Note: When you choose the fallback-ip option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Alternate	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP. **Note:** When you choose the fallback-ip option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Fallback	Select the alternate load balancing method. The system tries this method if the Preferred and Alternate methods are unsuccessful in picking the WideIP.
Pools	Click  to add the pool details to the list. You can add multiple to the pool to the list. After adding pool(s) to the list, you can manage them.

9. Enter/select the **Pool Member Details**.

Pool Member Details Section - Field and Description Table

Field	Description
Info	Information that is to be noted while entering the details in this section.
***Pool**	Click the Retrieve () button to get the pool name.
***Pool Member Type**	Select the pool member types. The pool member types are: • **Generic Host** – When you select this pool member type, you get the option to select the Generic Host Typ 

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **Big IP System** – When you select this pool member type, you get the option to select the configured virtua  **Note:** It takes a while to load the servers.
***Virtual Server**	When the pool member type is selected as Big IP System. Select a virtual server from the dropdown list.
***Pool Member List**	Enter the IP address of the pool member. And then click  . Any number of pool numbers can be ad

10. Enter/select the **Topology** details.

Topology Section - Field and Description Table

Field	Description
***Request Source (Region)**	The origination section of the topology record for the pool. Click  to retrieve the list of associated regions.
***Destination (Pool)**	The target section of the topology record for the pool.
***Weight**	The weight of the topology record. The system finds the weight of the first topology record that matches the server object (pool or pool member) and the local DNS. The system then assigns that weight as the topology score for that server object. The system load balances to the server object with the highest topology score. If the system finds no topology record that matches both the server object and the local DNS, then the system assigns that server object a zero score.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Topology Record Detail …**	After providing the topology details for the pool, click  . The List of topologies is added for the pool. You can manage topology records after adding them.

11. Click **Submit**.

    The **Confirmation** dialog box is displayed.

12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

    The form will be saved as **Open request** under **Request** > **My Request**.

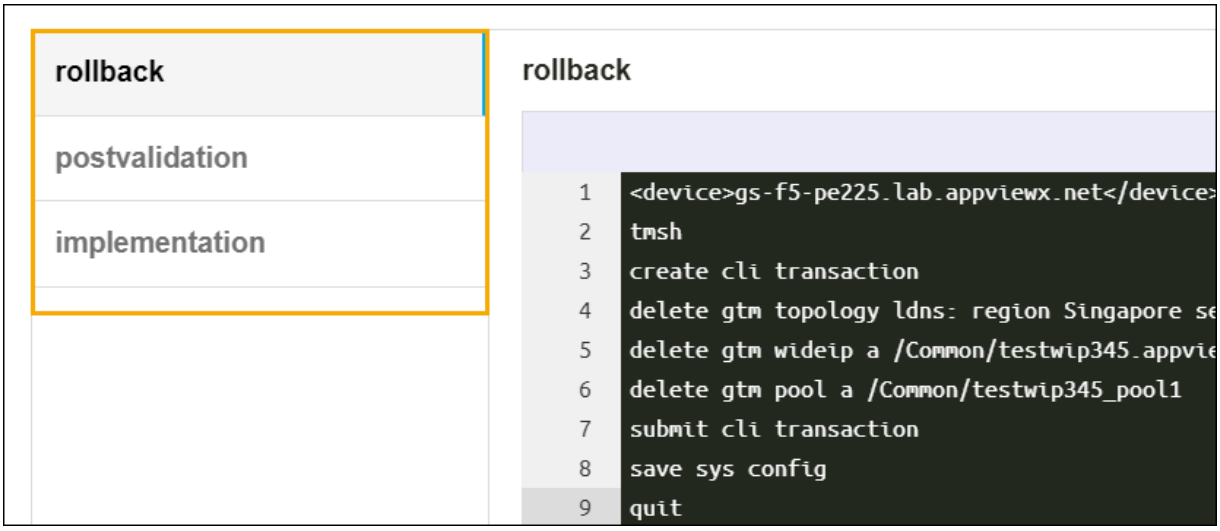
    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Review** stage.

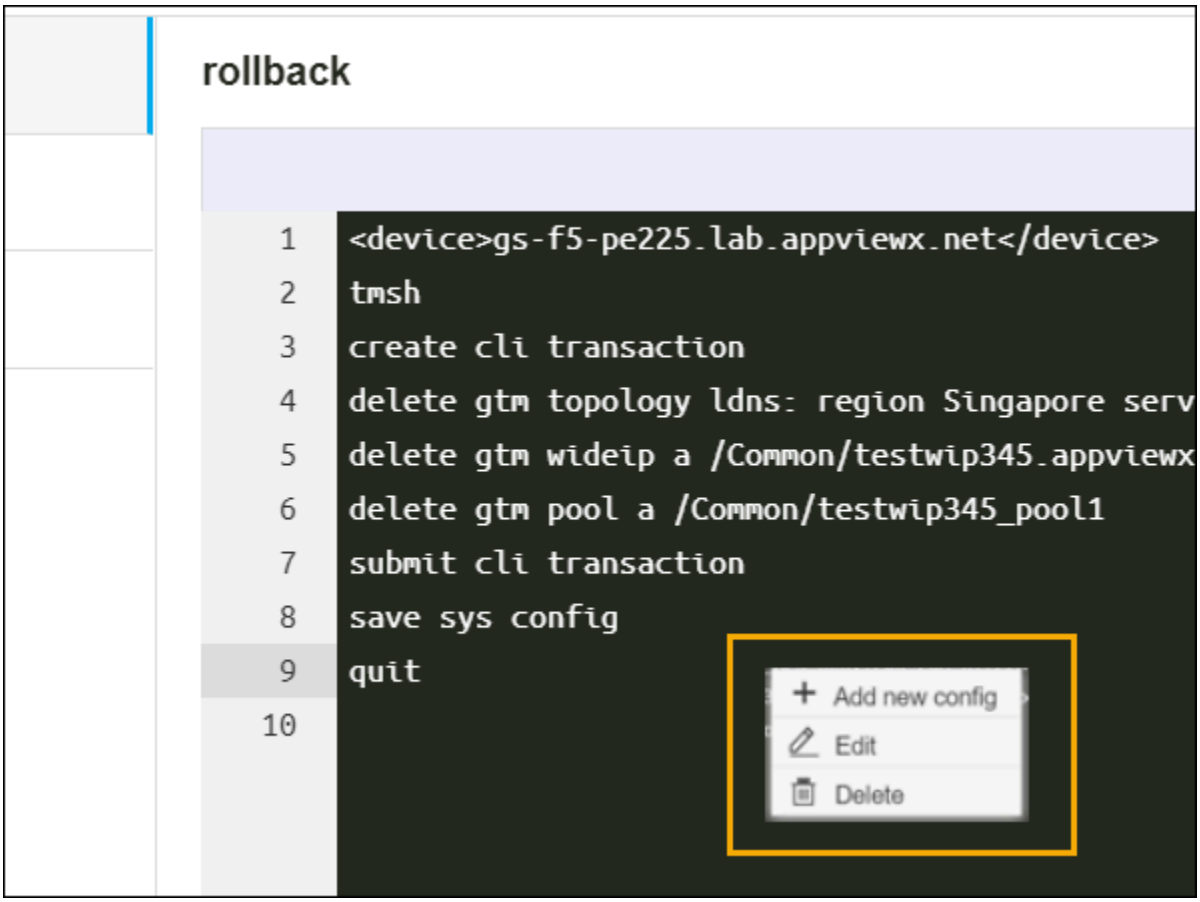
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

13. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



14. (Optional) If you need to change any data at this stage, right click and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



15. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

16. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    After a successful post-validation, the workflow is created and the email triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Create F5 GTM WideIP with Service Now

This workflow creates a WideIP with multiple Pools and integrates with ITSM – Service Now, where you can set Time Zone, Start time, and End time for the ticket creation.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 GTM WideIP with Service Now** workflow. The **Run** and **Schedule** buttons are displayed.
4. Click 

. The **Request > Create F5 GTM WideIP with Service Now :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
Datacenter	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter/select the **DNS Details**.

DNS Details Section - Field and Description Table

Field	Description
***Do you want to integrate with DNS**	By default, the **No** option is selected. If you want to integrate with DNS for CNAME record creation, select **Y** When you select **Yes**, the following fields are displayed:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

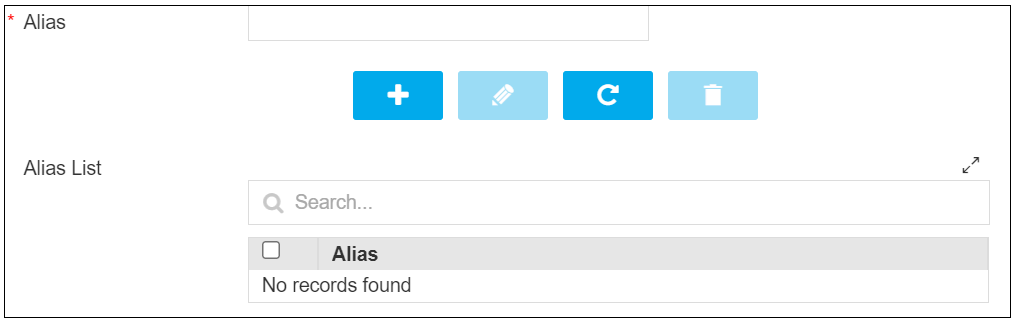
Field	Description
	 Select the **Vendor** from the drop-down list. The default vendors are: **Infoblox** and **bluecat**. When you select DNS vendor as **Infoblox**, you need to provide the following details: • Select the **Device Name** from the drop-down list. • Enter the **App Name** for the WideIP name. When you select DNS vendor as **bluecat**, you need to provide the following details:  • Select the **Device Name** from the drop-down list. • Select the **Configuration** from the drop-down list. • Select the **View** for the selected configuration. The options are displayed based on the selected configura • Enter the **App Name** for the WideIP name.

7. Enter/select the **General Properties**.

General Properties Section - Field and Description Table

Field	
***Wide IP Name**	Enter the WideIP Name in the FQDN format.
***Type**	Select the pool/WideIP type. The types are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	
	• A - the pool/Wide IP responds to A queries. The A pool/Wide IP is a mapping of an FQDN to a set of IP • CNAME - the pool/Wide IP responds to CNAME queries. A CNAME pool/Wide IP is a mapping of a FQ
Description	Enter the descriptive text to identify the WideIP.
State	It allows the system to use this wide IP and its resources for load balancing. The options are: • enabled (default) • disabled
***Do you want to add Alias ?**	Add Alias for the WideIP name. The options are: • **No** (default) • **Yes** – When you select **Yes** the following fields are displayed to add alias details:  Enter the alias name for WideIP in the Alias field. You can add any number of aliases for WideIP. After

8. Enter/select the **Pool** details.

Pool Section - Field and Description Table

Field	Description
Info	Information that are to be noted while entering the details in this section.
***Load Balancing Method**	The load balancing method is used to select a pool in this WideIP. The methods are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **round-robin** - the system selects the pools sequentially. • **global-availability** -. the system selects a pool by following the order of the Pool List. The system repeatedly selects the first pool in the list for as long as its status is available. If the pool becomes unavailable for any reason, the system then repeatedly selects the next pool in the list, and so on. • **ratio** - the system selects a pool based on the ratio that you assign to the pool.
***Persistence**	By default, this option is disabled. When enabled, a local DNS makes repetitive requests on behalf of a client, the system reconnects the client to the same resource as previous requests.
***Pool Name**	Name of the pool.
Preferred	Select the preferred load balancing method. The system tries this method first. **Note:** When you choose the fallback-ip option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Alternate	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP. **Note:** When you choose the fallback-ip option, the Fallback IP field will be displayed and you need to input the Fallback IP.
Fallback	Select the alternate load balancing method. The system tries this method if the Preferred and Alternate methods are unsuccessful in picking the WideIP.
Fallback IP	When the fallback-ip option is selected as Preferred or Alternate load balancing method, this field will be displayed to add the Fallback IP. The Fallback IP is not applicable for the pool type CNAME.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

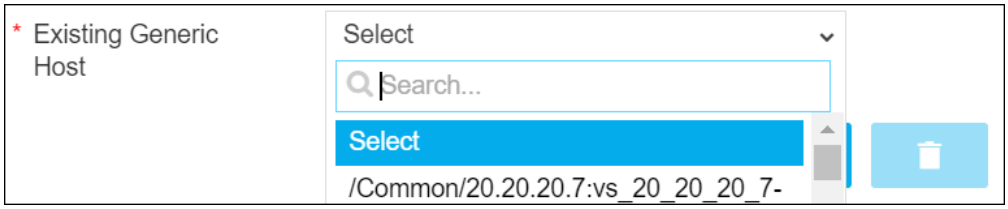
Field	Description
Health Monitor	The health monitors that the system uses to determine whether it can use this pool for load balancing. To get the fields to be monitored, click  , and then select the objects from the drop-down option.
***Pools**	Add multiple pools by clicking  . After adding pool(s) to the list. You can manage them.

9. Enter/select the **Pool Member Details**.

Pool Member Details Section - Field and Description Table

Field	Description
Info	Information that are to be noted while entering the details in this section.
***Pool**	Click  to select the pool name.
***Pool Member Type**	Select the pool member types. The pool member types are: • Generic Host – When you select this pool member type, you get the option to select the Generic Host Typ  • Big IP System – When you select this pool member type, you get the option to select the configured virtua

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	 **Note:** It takes a while to load the servers.
***Generic Host Type**	When the pool member type is selected as Generic Host, you need to select the type of generic host. The g Existing – This option allows to select the existing host. When you select this option, the existing generic hos  Create New – This option allows you to create a generic host. When you select this, the following fields are • ***Data Center** – select the datacenter of the pool member. • ***Pool Member IP** – enter the IP of the pool member. • ***Port** – enter the port of the pool member.
***Virtual Server**	When the pool member type is selected as Big IP System. Select a virtual server from the dropdown list.
***Pool Member List**	Enter the IP address of the pool member. And then click  . Any number of pool numbers can b

10. Enter/select the **Topology** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Topology Section - Field and Description Table

Field	Description
Info	Information that is to be noted while entering the details in this section.
***Request Source (Region)**	The origination section of the topology record for the pool. Click  to get the list of associated regions.
***Destination (Pool)**	The target section of the topology record for the pool.
***Weight**	The weight of the topology record. The system finds the weight of the first topology record that matches the server object (pool or pool member) and the local DNS. The system then assigns that weight as the topology score for that server object. The system load balances to the server object with the highest topology score. If the system finds no topology record that matches both the server object and the local DNS, then the system assigns that server object a zero score.
***Topology Record…**	After providing the topology details for the pool, click Add  icon. The List of topologies is added for the pool. You can manage topology records after adding them.

11. Enter/select the **Change Management** details.

Change Management Section - Field and Description Table

Field	Description
***Do you want to integrate ServiceNow?**	By default, the **No** option is selected. If you want to integrate with ServiceNow, select **Yes** and provide  • **Timezone** – select the timezone from the drop-down list. • **Start Time** – select the start date and time from the calendar. • **End Time** – select the end date and time from the calendar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

12. Click **Submit**.
    The **Confirmation** dialog box is displayed.
13. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Review** stage.

    Note:
    - If there is any failure at this stage, start a new request by providing the correct data.
    - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

14. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs. <screenshot>
15. (Optional) If you need to change any data at this stage, right click and select the required option.<screenshot>
16. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

    Note:
    To stop running the workflow creation, click **Cancel**.

17. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs. <screenshot>

    Note:
    The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Modify F5 GTM WideIP Advanced

This workflow modifies an existing WideIP and/or its Pool details, which is associated with the WideIP, and manages Pool members.

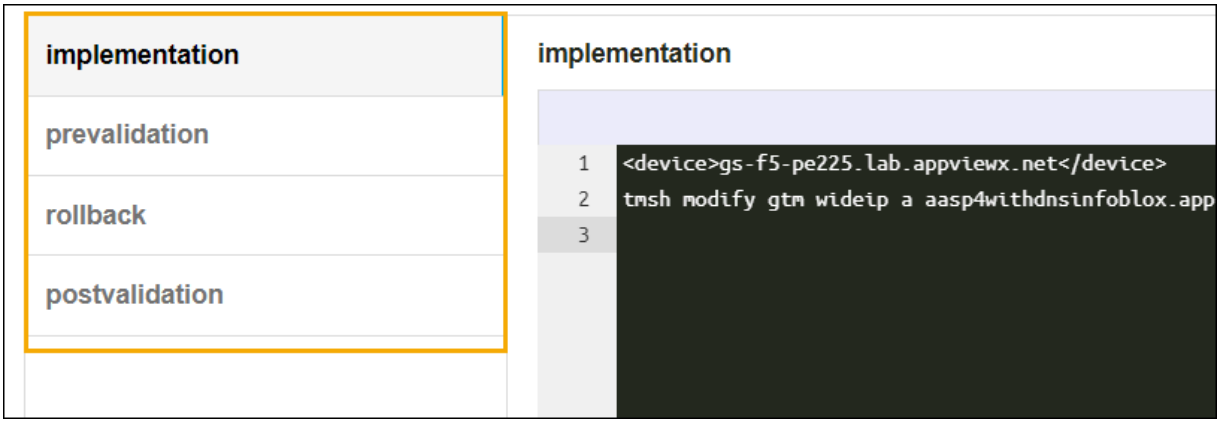
To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify F5 GTM WideIP Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Modify F5 GTM WideIP Advanced :: FormBuilder** page is displayed.
5. Enter/select the **Device Details**, **Wide IP Details**, **Pool**, and **Pool Member Details** of the device that are to be modified.
6. On the **Validation Details** section, click **Form Validation**.
   The validation starts and the results are displayed in the Validation Status box.
7. On the **RBAC Details** section, search and select the resource from the dropdown list.
8. If required, update the necessary details of Device, WIP, pool, and pool member, and then click **Submit**.
   The validation starts automatically and reaches the **Review** stage.

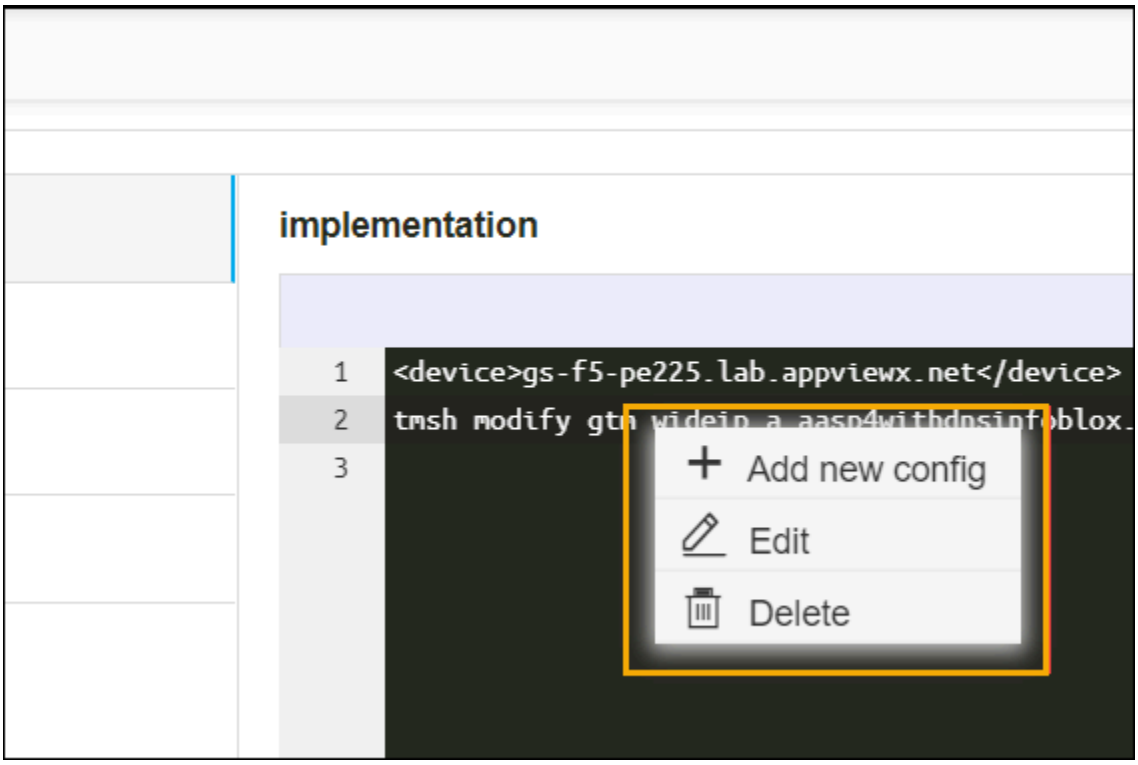
> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



10. (Optional) If you need to change any data at this stage, right click and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



11. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Delete F5 GTM WideIP Advanced

This workflow deletes an existing WideIP. On deleting a WideIP, associated Pool, Pool members, Topology, Monitor, and Servers with the WideIP are deleted if they are not associated with any other WideIP.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 GTM WideIP Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete F5 GTM WideIP Advanced :: FormBuilder** page is displayed.
5. For the device to be deleted, select the **Device Details** and **Wide IP Details**.
6. Click **Submit**.
   The validation starts automatically and reaches the **Review** stage.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

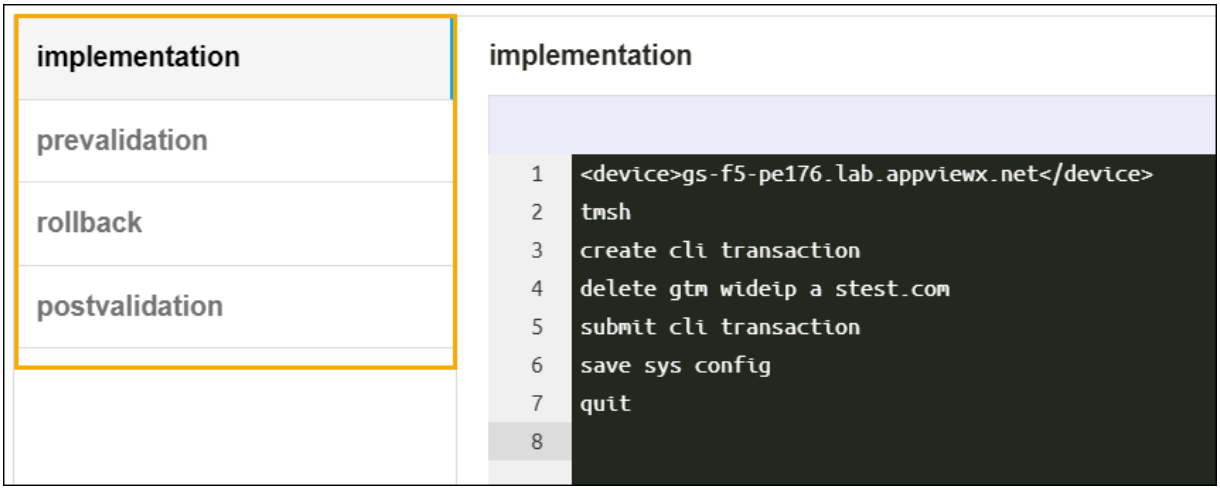
   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

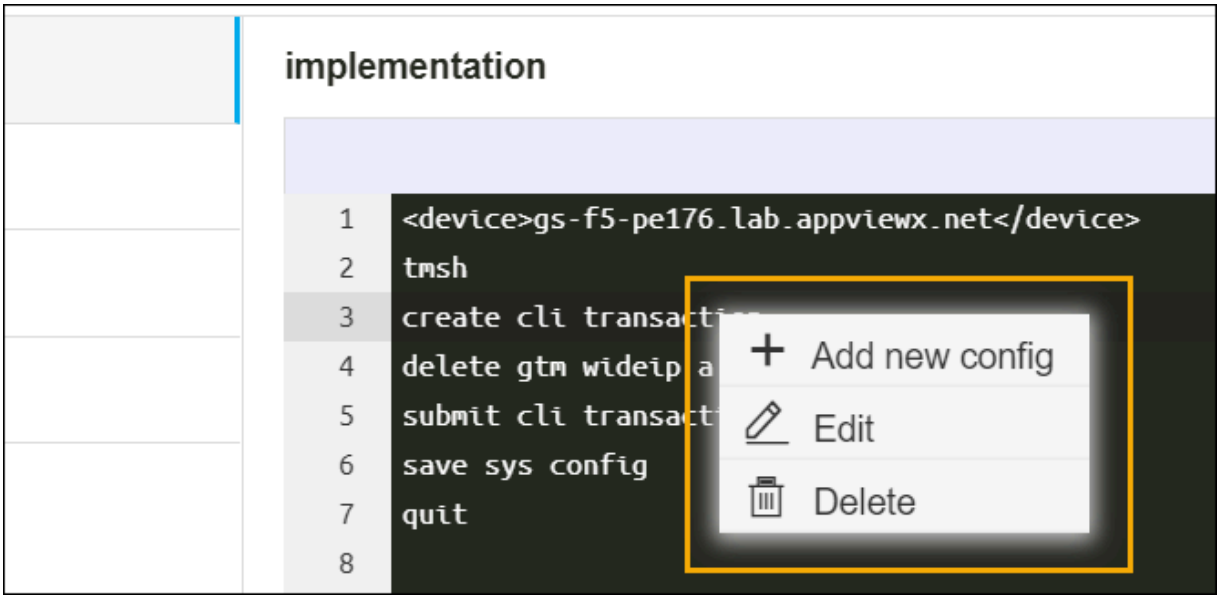
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

9. (Optional) If you need to change any data at this stage, right click and select the required option.



10. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

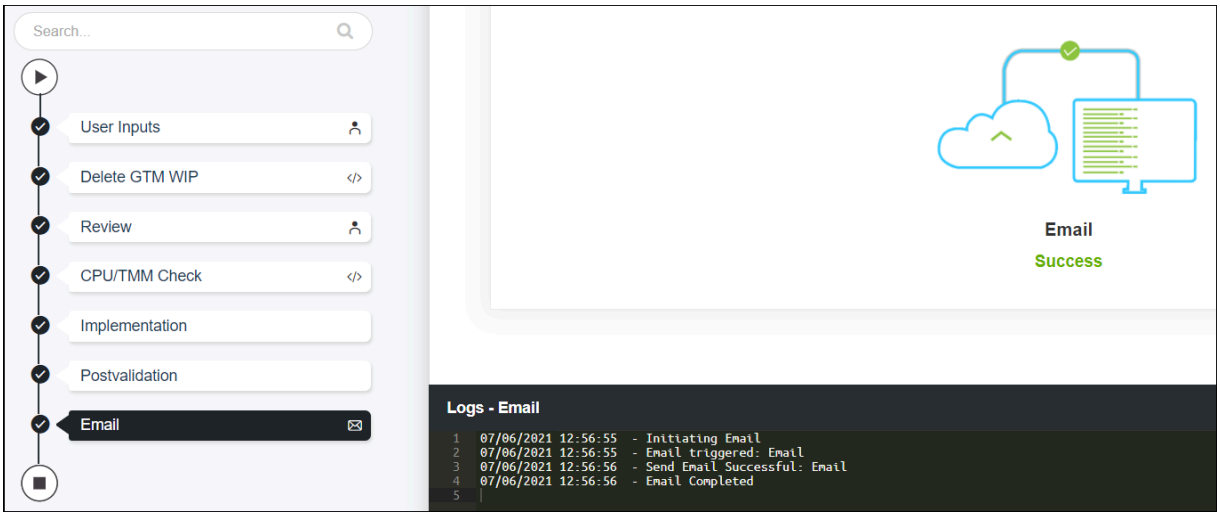
> **Note:**
> To stop running the workflow creation, click **Cancel**.

11. Click **Ok** to continue the workflow creation.

    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Delete F5 GTM WideIP with DNS

This workflow deletes an existing WideIP. On deleting a WideIP, associated Pool, Pool members, Topology, Monitor, and Servers with the WideIP are deleted if they are not associated with any other WideIP. Also, it deletes the associated CNAME record in the DNS – Infoblox/Bluecat.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 GTM WideIP with DNS** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete F5 GTM WideIP with DNS :: FormBuilder** page is displayed.
5. For the device to be deleted, select the **Device Details**, **Wide IP Details**, and **DNS Details**.
6. Click **Submit**.
   The validation starts automatically and reaches the **Review** stage.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

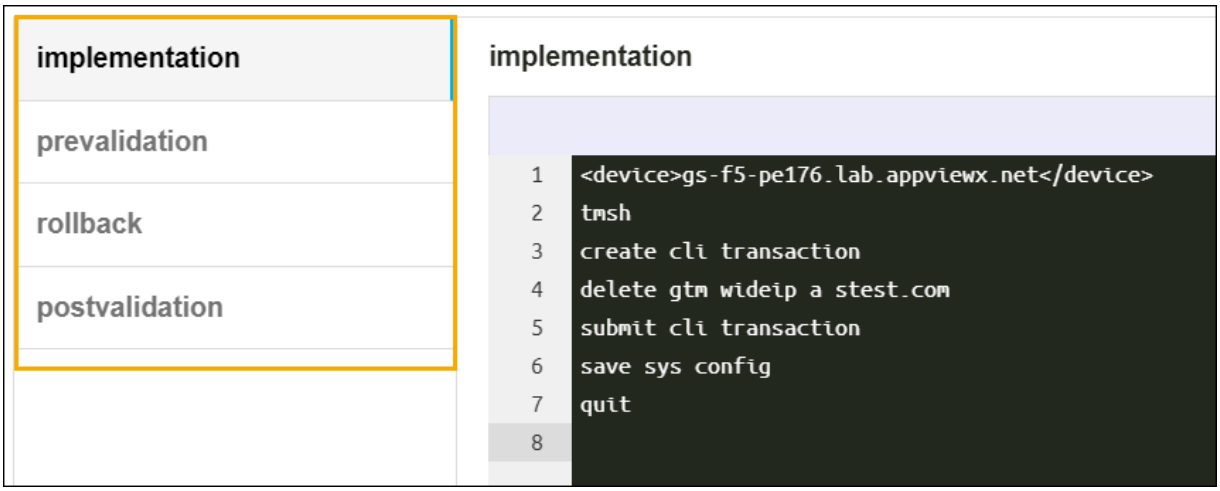
To submit the form, click **OK**.

The validation starts automatically and reaches the **Review** stage.

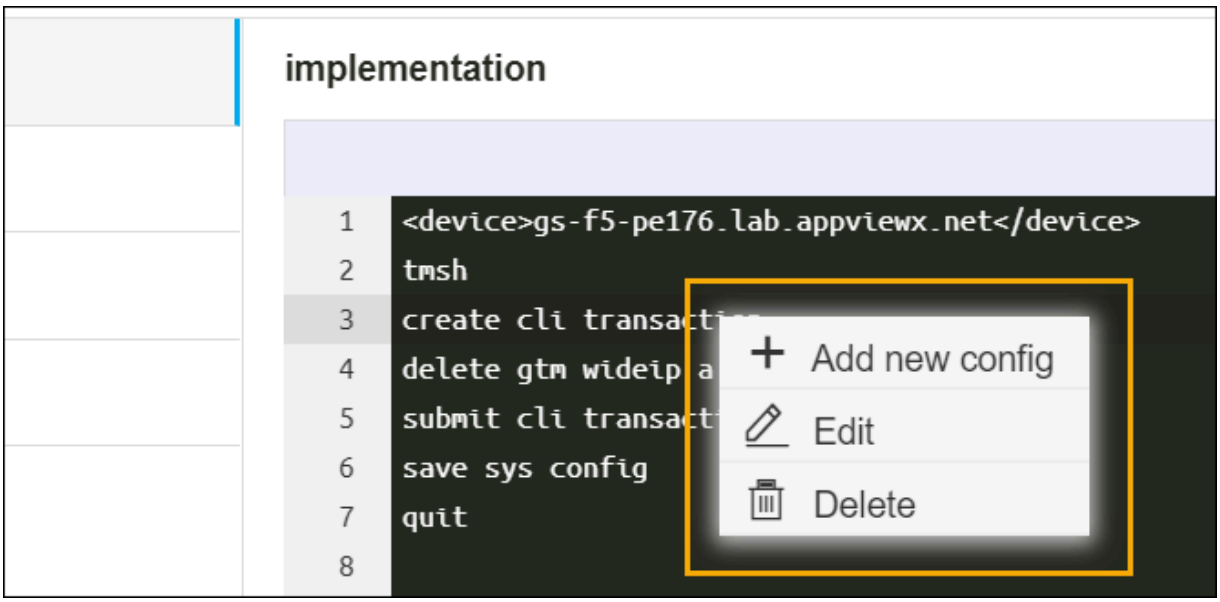
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.



9. (Optional) If you need to change any data at this stage, right click and select the required option.



10. After the review, click **Submit**.

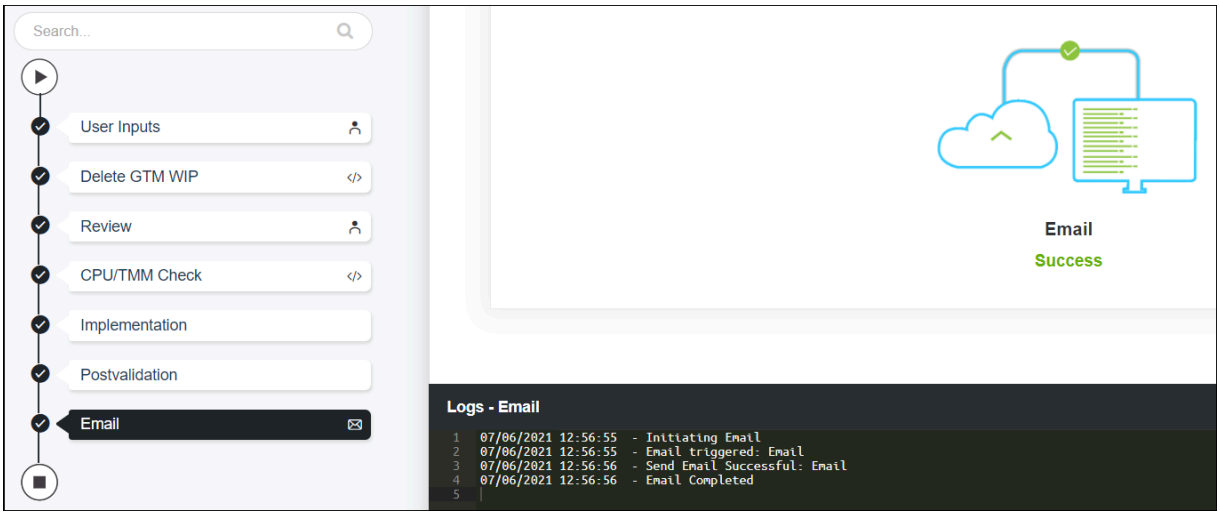
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

11. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.



> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Disable and Delete Unused WIP-State|Status

This workflow identifies and decommissions unused F5 GTM WideIPs (WIPs) based on the state or status. The process includes selecting the appropriate devices, identifying WIPs that have been unused for a specified duration, disabling those WideIPs, and decommissioning the associated virtual servers.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable and Delete Unused WIP-State|Status** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click 

.

   The **Request > Disable and Delete Unused WIP-State|Status :: FormBuilder** page is displayed.

5. For the WIPs to be identified, enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
Datacenter	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***No of Days**	Enter the number of days for which the WIPs have been unused. A WIP is considered **Unused** if any of the following conditions are met: • The WideIP Status is OFFLINE and the State is ENABLED. • The WideIP Status is OFFLINE and the State is DISABLED. • The WideIP Status is UNKNOWN and the State is DISABLED. • The WideIP Status is AVAILABLE and the State is DISABLED.
** : Mandatory fields*	

6. For the WIPs to be disabled or deleted based on the state or status, select the **Actions and Metrics**.

Actions and Metrics Section - Field and Description Table

Field	Description
***Action on WIPs**	Select one of the following actions to be performed on the WIPs identified: • **Disable** (default) - Disables the WIPs identified as unused based on the specified duration • **Disable and Delete** - Disables the identified unused WIPs and then deletes them. • **Disable Monitor and Delete** - Disables the identified unused WIPs, monitors them, and then deletes them, if they remain unused within the monitoring period. If you choose this option, provide the monitoring period in the **Monitoring Period** field. Enter a period within the range of 1 to 31 days.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Unused WIP Metrics**	The options are: • **Status** (default) - Status of the WIP to be considered as unused. • **State/Status** - State and status of the WIP to be considered as unused.
** : Mandatory fields*	

7. Enter/select the **Change Management** details.

Change Management Section - Field and Description Table

Field	Description
***Do you want to integrate ServiceNow?**	By default, the **No** option is selected. If you want to integrate with ServiceNow, select **Yes** and provide the following details: • **Timezone** – retrieve and select the timezone from the drop-down list. • **Start Time** – select the start date and time from the calendar. • **End Time** – select the end date and time from the calendar.
** : Mandatory fields*	

8. Click **Submit**.
   The validation starts automatically and reaches the **Review** stage.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

> **Note:**
> • If there is any failure at this stage, start a new request by providing the correct data.
> • Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.
11. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Disable and Delete Unused URL-State|Status

This workflow identifies and decommissions unused URLs (WIPs) on an F5 server. The process involves selecting the relevant devices, specifying the duration of inactivity, and then systematically decommissioning the unused URLs and their associated objects, starting from WIP down to LTM pool members.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP GTM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Disable and Delete Unused URL-State|Status** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

   .
   The **Request > Disable and Delete URL WIP-State|Status :: FormBuilder** page is displayed.
5. For the URLs to be identified, enter/select the **Device Details**.

Device Details Section - Field and Description Table

Field	Description
Datacenter	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are created without a datacenter in the Device Inventory, select the datacenter as **None**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***No of Days**	Enter the number of days for which the URLs have been unused. A URL is considered **Unused** if any of the following conditions are met: • The URL Status is OFFLINE and the State is ENABLED. • The URL Status is OFFLINE and the State is DISABLED. • The URL Status is UNKNOWN and the State is DISABLED. • The URL Status is AVAILABLE and the State is DISABLED.
** : Mandatory fields*	

6. For the URLs to be disabled or deleted based on the state or status, select the **Actions and Metrics**.

Actions and Metrics Section - Field and Description Table

Field	Description
***Action on URLs**	Select one of the following actions to be performed on the URLs identified: • **Delete** - Deletes the identified unused URLs • **Disable Monitor and Delete** - Monitors the identified unused URLs, and then deletes them, if they remain unused within the monitoring period. If you choose this option, provide the monitoring period in the **Monitoring Period** field. Enter a period within the range of 1 to 31 days.
***Unused URL Metrics**	The options are: • **Status** (default) - Status of the URL to be considered as unused. • **State/Status** - State and status of the URL to be considered as unused.
** : Mandatory fields*	

7. Click **Submit**.
   The validation starts automatically and reaches the **Review** stage.
8. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The validation starts automatically and reaches the **Review** stage.

> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.
10. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

11. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

    The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

F5 XC Distributed Cloud

Create F5 XC HTTP Load Balancer - Basic

This workflow creates a HTTP load balancer using the device name and portal FQDN, along with load balancer details such as type, name, domain name, port, and pool details.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 XC Distributed Cloud** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 XC HTTP Load Balancer - Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click 

.

   The **Request > Create F5 XC HTTP Load Balancer - Basic :: FormBuilder** page is displayed.

5. Enter/select **Device Details** details.

Device Details Section - Field and Description Table

Field	Description
***Device Name**	Select a device from the dropdown list.
***Portal FQDN**	Enter the F5 XC cloud platform application URL. For example, domain.ves.volterra.io.
**: Mandatory fields*	

6. Enter/select **LoadBalancer Details** details.

LoadBalancer Details Section - Field and Description Table

Field	Description
***Load Balancer Type**	Select a desired load balancer type from the dropdown list. The options are: • HTTP • HTTPS with Automatic Certificate • HTTPS with Custom Certificate
***Load Balancer Name**	Enter the name of the load balancer.
***Domain Name**	Enter the domain name.
***Port**	Enter the port number.
**: Mandatory fields*	

7. Enter/select **Origin Pool Details** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Origin Pool Details Section - Field and Description Table

Field	Description
***Create and Associate Origin Pool?**	To integrate with pool, select **Yes**, provide the necessary details, and then click  .
**: Mandatory fields*	

8. Click **Next**.

   The **Confirmation** dialog box is displayed.

> **Note:**
> - If you want to save this form and edit it later, click **Save Draft**, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click **Cancel**.

9. Click **OK**.

   A form opens to provide the certificate details.

10. Enter or select the certificate details in the form, and then click **Submit**.
11. Click **OK** to continue the workflow creation.

> **Note:**
> It takes a while to complete the request.

The workflow is created.

> **Note:**
> The stages of the request are shown on the left of the screen. To view a specific stage of the request, click on the corresponding stage.

Create F5 XC TCP Load Balancer - Basic

This workflow creates a TCP load balancer using the device name and portal FQDN, along with load balancer details such as type, name, domain name, port, and pool details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 XC Distributed Cloud** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 XC TCP Load Balancer - Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Create F5 XC TCP Load Balancer - Basic :: FormBuilder** page is displayed.
5. Enter/select **Device Details** details.

Device Details Section - Field and Description Table

Field	Description
***Device Name**	Select a device from the dropdown list.
***Portal FQDN**	Enter the F5 XC cloud platform application URL. For example, domain.ves.volterra.io.
**: Mandatory fields*	

6. Enter/select **LoadBalancer Details** details.

LoadBalancer Details Section - Field and Description Table

Field	Description
***Load Balancer Type**	Select a desired load balancer type from the dropdown list. The options are: • TCP • TCP over TCP with Automatic Certificate • TCP over TCP with Custom Certificate
***Load Balancer Name**	Enter the name of the load balancer.
***Domain Name**	Enter the domain name.
***Port**	Enter the port number.
**: Mandatory fields*	

7. Enter/select **Origin Pool Details** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Origin Pool Details Section - Field and Description Table

Field	Description
***Origin Pool Name**	Enter the name for the pool.
*Load Balancer Algorithm	The load balancing algorithm is used to select a pool to distribute traffic across multiple servers. The algorithms are: • Round Robin • Least Request • Ring Hash • Random • Load Balancer Override
*Origin Server Type	Select a server type from the drop-down list. And then click the Add  button. Any number of pool numbers can be added to the pool. After adding the pool, you can manage them.
*Origin Server Port	Enter the server port number.
*: Mandatory fields	

8. Click **Next**.
   The **Confirmation** dialog box is displayed.

> **Note:**
> • If you want to save this form and edit it later, click **Save Draft**, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> • If you want to cancel this form, click **Cancel**.

9. Click **OK**.
   A form opens to provide the certificate details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. Enter or select the certificate details in the form, and then click **Submit**.
11. Click **OK** to continue the workflow creation.

> **Note:**
> It takes a while to complete the request.

The workflow is created.

> **Note:**
> The stages of the request are shown on the left of the screen. To view a specific stage of the request, click on the corresponding stage.

Delete F5 XC HTTP Load Balancer - Basic

This workflow deletes a F5 XC HTTP load balancer.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 XC Distributed Cloud** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 XC HTTP Load Balancer - Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

   .
   The **Request > Delete F5 XC HTTP Load Balancer - Basic :: FormBuilder** page is displayed.
5. Enter/select **Device Details** details.

Device Details Section - Field and Description Table

Field	Description
***Device Name**	Select a device from the dropdown list.
***Portal FQDN**	Enter the F5 XC cloud platform application URL. For example, domain.ves.volterra.io.
**: Mandatory fields*	

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

6. Under the **LoadBalancer Details** section, click the **Fetch Load Balancer** button, and then choose a load balancer from the **Load Balancer Name** drop-down list.
7. Click **Submit**.
   The **Confirmation** dialog box is displayed.
8. (Optional) To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.
9. Click **Ok** to continue workflow creation.
   It takes a while to complete the request.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Delete F5 XC TCP Load Balancer - Basic

This workflow deletes a F5 XC TCP load balancer.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 XC Distributed Cloud** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete F5 XC TCP Load Balancer - Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Delete F5 XC TCP Load Balancer - Basic :: FormBuilder** page is displayed.
5. Enter/select **Device Details** details.

Device Details Section - Field and Description Table

Field	Description
***Device Name**	Select a device from the dropdown list.
***Portal FQDN**	Enter the F5 XC cloud platform application URL. For example, domain.ves.volterra.io.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
*: Mandatory fields	

6. Under the **LoadBalancer Details** section, click the **Fetch Load Balancer** button, and then choose a load balancer from the **Load Balancer Name** drop-down list.
7. Click **Submit**.
   The **Confirmation** dialog box is displayed.
8. (Optional) To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.
9. Click **Ok** to continue workflow creation.
   It takes a while to complete the request.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

F5 BIG-IP AS3

DNS

Create GTM Application Services

This workflow creates GTM Wide-IP along with GTM pool and monitor.

Prerequisites

- The AS3 service has to be enabled in the target F5 device.
- F5 device list displays only AS3 service enabled and active devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create GTM Application Services** workflow.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Run** and **Schedule** buttons are displayed.

4. Click Run.

   The **Request > Create GTM Application Services :: FormBuilder** page is displayed.

5. Enter/select the **General Properties**.

General Properties Section - Field and Description Table

Field	Description
****Device Name**	Select a device from the dropdown list. Only the devices that are AS3 enabled and active are displayed.
***Partition Type**	Select a partition type. The options are: • Create New Partition (default) - select this option to create a new partition. • Existing Partition - select this option to choose an existing partition. **Note:** Partition dropdown lists only the partitions created via AS3 triggered API calls.
***Enter Partition Name**	Enter the partition name.
***Application Type**	Select a application type. The options are: • Create New Application (default) - select this option to create a new application. • Existing Application - select this option to choose an existing partition.
***Enter Application Name**	Enter the name for the application.

6. Enter/select the **Wide IP** details.

Wide IP Section - Field and Description Table

Field	Description
WideIP FQDN	Enter the fully qualified domain name.
WideIP Type	Select Wide IP type from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. Enter/select the **Pool** details.

Pool Section - Field and Description Table

Field	Description
***Pool Name**	Enter the pool name.
***Pools Load Balancing Method**	Select a load balancing method for the pools.
Pool Members Preferred Load Balancing Method	Select a preferred load balancing method for the pool members. The system tries this method first.
Pool Members Fallback Load Balancing Method	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP.

8. Enter/select the **Pool Members** details.

Pool Members Section - Field and Description Table

Field	Description
***Pool member Type**	Select a pool member type from the deopdown list. The types are: • **Generic Host** (default) - When you select this option you need to select GTM server and virtual servers. • **BIG-IP System** -. When you select this option you need to select GTM server and existing VIP.
***GTM Server**	Select a GTM server from the dropdown list.
***Virtual Servers**	Select the virtual servers from the dropdown list

9. Enter/select the **Monitors** details.

Monitors Section - Field and Description Table

Field	Description
***Do you want to associate a HTTP/HTTPS monitor to the pool**	Select Yes to associate a HTTP/HTTP monitor, and then select monitor from the dropdown list otherwise select No.

10. Click **Submit**.

    The **Confirmation** dialog box is displayed.
11. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The form will be saved as **Open request** under **Request** > **My Request**.

OR

To submit the form, click **OK**.

The validation starts automatically and reaches the **Review** stage, after which the workflow is created.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

Note:

The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Modify GTM Application Services

This workflow modifies GTM Wide-IP along with GTM pool and monitor association.

Prerequisites

- The AS3 service has to be enabled in the target F5 device.
- F5 device list displays only AS3 service enabled and active devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify GTM Application Services** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

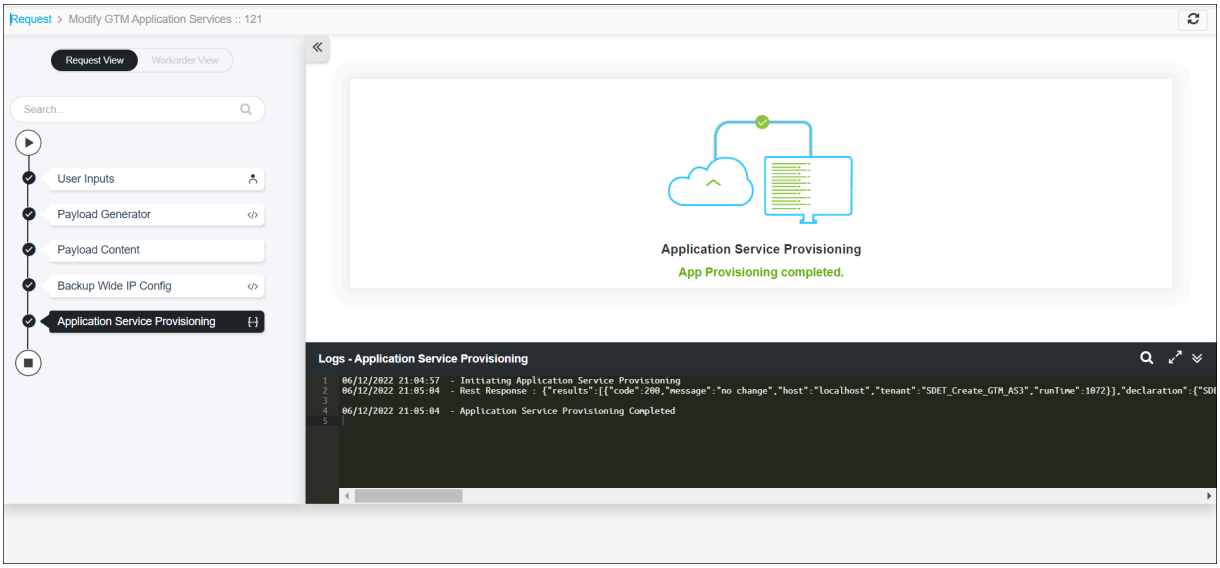
.
   The **Request > Modify GTM Application Services :: FormBuilder** page is displayed.
5. In the **General Properties** section, select the Device Name to modify, and then choose the Partition and Application details of the device that are to be modified.
6. Click **Get Wide IP** to get an existing Wide-IP and select the target Wide-IP.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. Fetch the associated pool by clicking **Get Pool Name** button, and then fetch the pool's details by clicking the **Get Pool Details** button to modify.
8. Fetch the pool member's configuration by clicking the **Get Pool Members** button to modify.
9. Update the necessary details, and then click the **Submit** button.
   The validation starts automatically stage.

> **Note:**
> - If you want to save this form to edit it later, click **Save Draft**, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click **Cancel**.

10. The workflow is created and GTM Wide-IP along with GTM pool and monitor association are updated.



> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Delete GTM Application Services

By running this workflow, you can delete a GTM Wide-IP along with GTM pool.

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

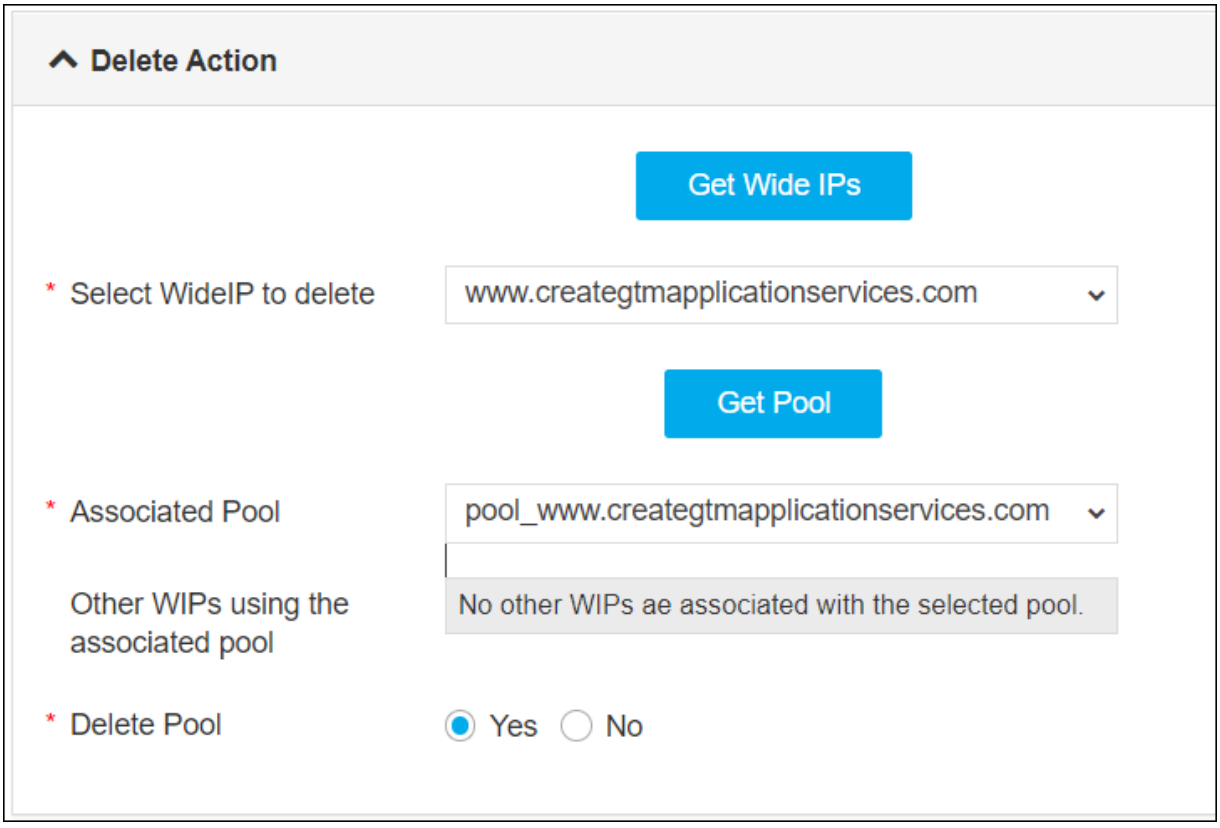
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete GTM Application Services** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Delete GTM Application Services :: FormBuilder** page is displayed.
5. In the **General Properties** section, select the Device Name, and then choose the Partition and Application details of the device that are to be deleted.
6. Fetch the following details:

   • Wide IP by clicking the **Get Wide IPs** button, and then select a Wide IP from the dropdown list.

   • Pool by clicking the **Get Pool** button, and then select an associated pool from the dropdown list.

   Click **Yes** to delete the associated pool otherwise select No.



7. Click the **Submit** button.
   The Confirmation popup opens.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button.

8. Click **Submit**.
   The **Confirmation** dialog box is displayed.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

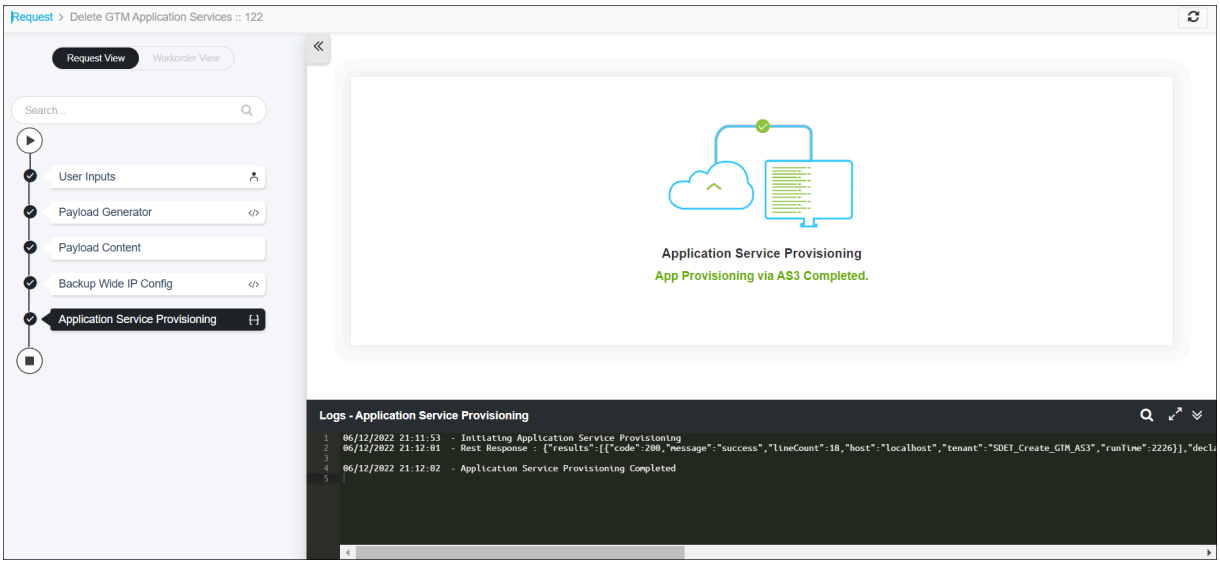
   The validation starts automatically.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

10. The workflow is created and the Wide IP along with its pool are deleted.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

LTM

Create LTM Application Services

This workflow creates virtual server along with pool and other objects using application services (AS3) in F5.

Prerequisites

- The AS3 service has to be enabled in the target F5 device.
- F5 device list displays only AS3 service enabled and active devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create LTM Application Services** workflow. The **Run** and **Schedule** buttons are displayed.
4. Click Run.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Request > Create LTM Application Services :: FormBuilder** page is displayed.

5. Enter/select the **Device and Partition Details**.

Device and Partition Details Section - Field and Description Table

Field	Description
***Device Name**	Select a device from the dropdown list. Only the devices that are AS3 enabled and active are displayed.
***Partition Type**	Select a partition type. The options are: • Create New Partition (default) - select this option to create a new partition. • Existing Partition - select this option to choose an existing partition. **Note:** Partition dropdown lists only the partitions created via AS3 triggered API calls.
***Enter Partition Name**	Enter the partition name.

6. Enter/select the required details in the **Enter Virtual Server Details** section.

Enter Virtual Server Details Section - Field and Description Table

Field	Description
***Virtual Server Name**	Enter the virtual server name.
***Virtual Server IP**	Select virtual server IP address.
***Virtual Server Port**	Enter virtual server port.

7. Enter/select the required deatils in the **Enter Pool Details** section.

Enter Pool Details Section - Field and Description Table

Field	Description
***Pool Name**	Enter the pool name.
***Pools Load Balancing Method**	Select a load balancing method for the pools.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Pool Member addresses**	Enter pool member' IP addresses.
***Pool Member Service Port**	Enter pool member service port.
***Pool Members**	Click the add button to add the pool member's details to the Pool Members table. **Note:** After adding the pool member's details to the Pool Members table, you can modify them.

8. Select the **Additional Details**.
9. Click **Submit**.
   The **Confirmation** dialog box is displayed.
10. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Review** stage, after which the workflow is created.

    Note:
    - If there is any failure at this stage, start a new request by providing the correct data.
    - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

    Note:
    The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

11. Click **Ok** to submit the form.
    The validation starts automatically after which the workflow is created.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

Modify LTM Application Services

This workflow modifies virtual server along with pool and other objects using application services (AS3) in F5.

Prerequisites

- The AS3 service has to be enabled in the target F5 device.
- F5 device list displays only AS3 service enabled and active devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify LTM Application Services** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Modify LTM Application Services :: FormBuilder** page is displayed.
5. In the **Device and Partition Details** section, select the Device Name to modify, and then choose the Partition of the device that are to be modified.
6. In the **Enter Virtual Server Details** section, click the **Get Virtuals** button to get the virtual server details and select the modify the virtual server details as needed.
7. In the **Enter Pool Details** section, fetch the associated pool by clicking the **Get Pool Name** button, and then fetch the pool's details by clicking the **Get Pool Details** button to modify.

   Note:
   Select the check box in the Pool Members table for the pool member to modify.

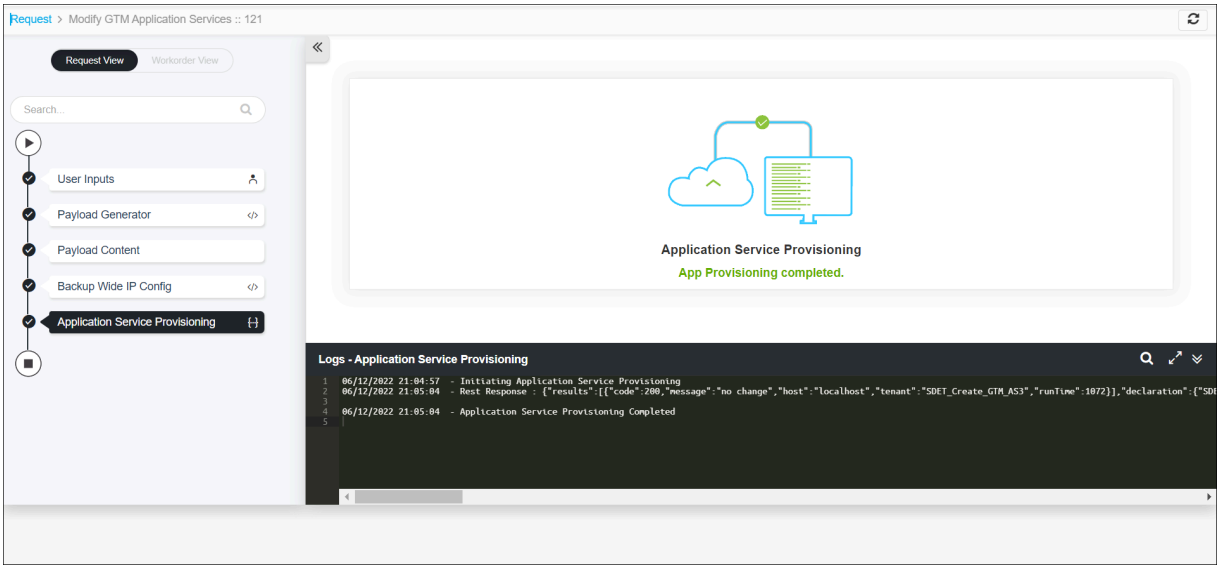
8. Update the necessary details, and then click the **Submit** button.
   The validation starts automatically stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button.

9. The workflow is created with modified details..



Note:

The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Delete LTM Application Services

By running this workflow, you can delete a virtual server along with pool.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete LTM Application Services** workflow. The **Run** and **Schedule** buttons are displayed.
4. Click Run.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Request > Delete LTM Application Services :: FormBuilder** page is displayed.

5. In the **General Properties** section, select the Device Name, and then choose the Partition and Application details of the device that are to be deleted.
6. Fetch the following details:

   - Wide IP by clicking the **Get Wide IPs** button, and then select a Wide IP from the dropdown list.
   - Pool by clicking the **Get Pool** button, and then select an associated pool from the dropdown list.

     Click **Yes** to delete the associated pool otherwise select No.

7. Click **Submit**.
   The **Confirmation** dialog box is displayed.
8. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage, after which the workflow is created.

   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

   Note:
   The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

9. Click **Ok** to submit the form.
   The validation starts automatically after which the workflow is created and the Wide IP along with its pool are deleted.

   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Fetch Application Services

This workflow fetches all the AS3 objects of a partition.

Prerequisites

- The AS3 service has to be enabled in the target F5 device.
- F5 device list displays only AS3 service enabled and active devices.

To run this workflow:

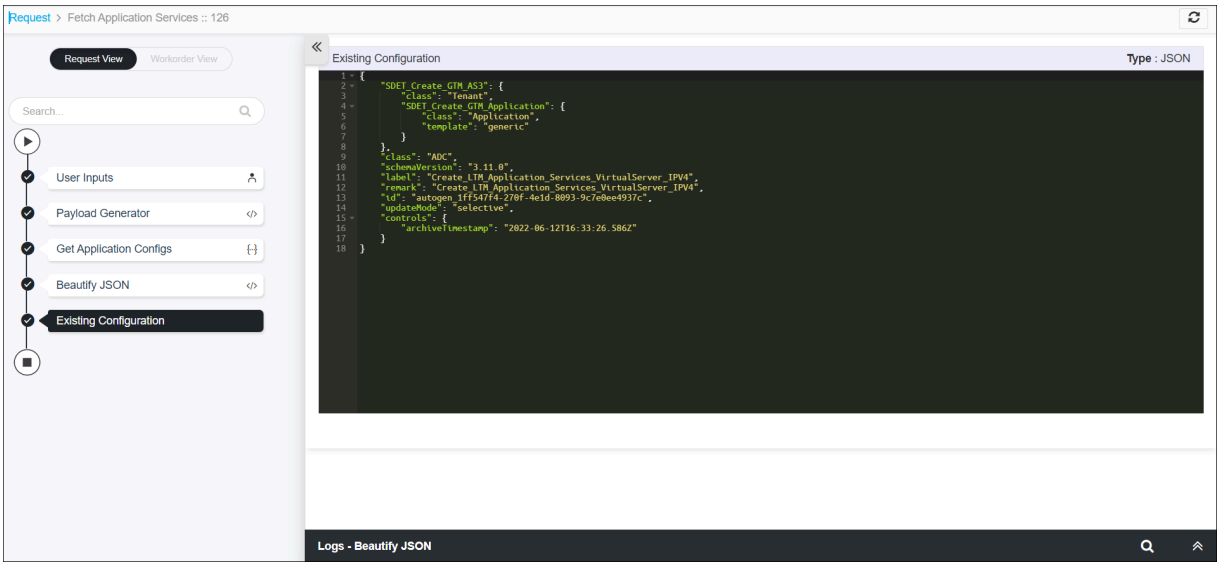
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP AS3** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Fetch Application Services** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Fetch Application Services :: FormBuilder** page is displayed.
5. In the **AS3 Details** section, select the Device Name and associated partition to fetch the AS3 objects.
6. Click the **Submit** button.
   The validation starts automatically stage.

   Note:
   - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
   - If you want to cancel this form, click the **Cancel** button.

7. The workflow is created and AS3 objects are fetched in the JSON format.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

F5 BIG IP ASM

Web App Vulnerability Remediation

This workflow create/selects a ASM Policy in F5 WAF Device to assess and remediate vulnerabilities

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG IP ASM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Web App Vulnerability Remediation** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Web App Vulnerability Remediation :: FormBuilder** page is displayed.
5. Enter/select the **Device and ASM Policy** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device and ASM Policy Section - Field and Description Table

Field	Description
***Device Name**	Select a device from the dropdown list for which the policy has to be created.

***ASM Policy Options**	Options to create a ASM policy. The options are:
	• **Create New** - select this option to create a policy newly and enter/select the following policy details:
	  • **ASM Policy Name** - enter the name for the policy.
	  • **Enforcement Mode** - select a enforcement mode from the dropdown list.
	  • **Do you want to associate a Parent Policy** - select Yes to associate a parent policy. Otherwise select No (default option).
	  • **Do you want to associate VIPs to the Policy** - select Yes to associate VIPs to the policy, and then select the desired VIPs from the **Virtual Servers** dropdown option. Otherwise select No (default option).
	• **Select Existing** - select this option to use the already created policy, and then select a policy from **ASM Policy** dropdown option.

6. Select the inputs for the **Upload Vulnerability Scan Report**.

Upload Vulnerability Scan Report Section - Field and Description Table

Field	Description
***Assessment Tool**	Select a assement tool from the dropdown list.
***Vulnerability Scan report**	Upload a vulnerability scan report in the `.xml` format.

7. Enter the **Email** ID in the **Send Assessment Report** section.
   Once the workflow is complete, the generated report will be sent to this email ID. More than one email ID can be added with comma separated.
8. Click **Submit**.
   The **Confirmation** dialog box is displayed.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

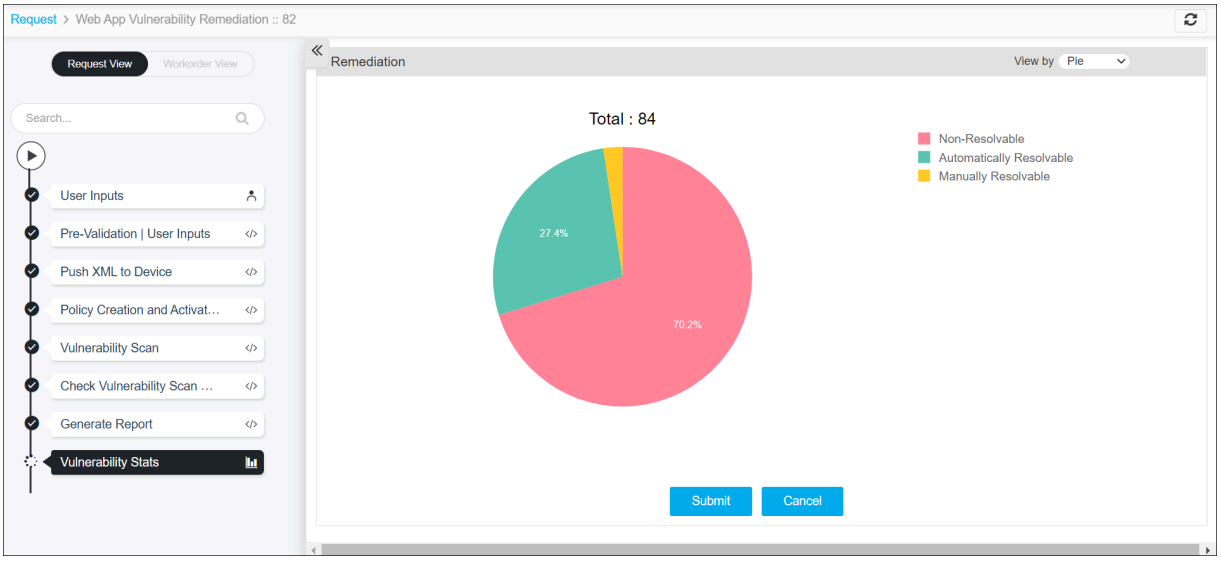
OR

To submit the form, click **OK**.

The validation starts automatically and reaches the **Review** stage.

> **Note:**
> • If there is any failure at this stage, start a new request by providing the correct data.
> • Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

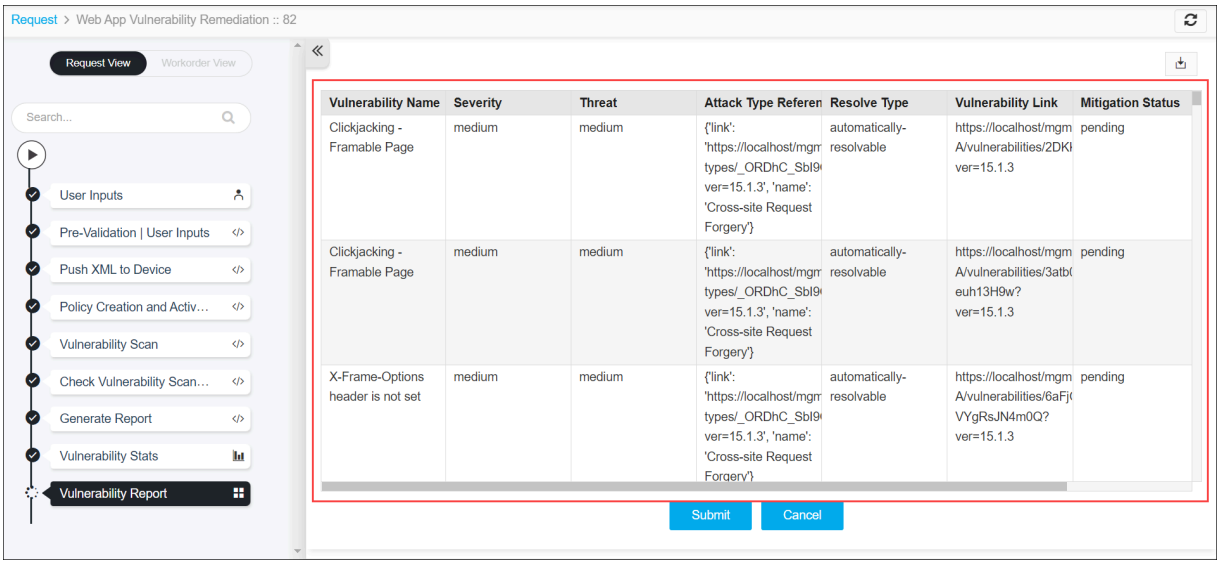
10. View the vulnerability status under the **Vulnerability Stats** tab.



> **Tip:**
> The view of this report can be changed from Pie chart or Donut view by selecting the option from the View by drop-down option.

11. Click **Submit**.
12. View the vulnerability report under the **Vulnerability Report** tab.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

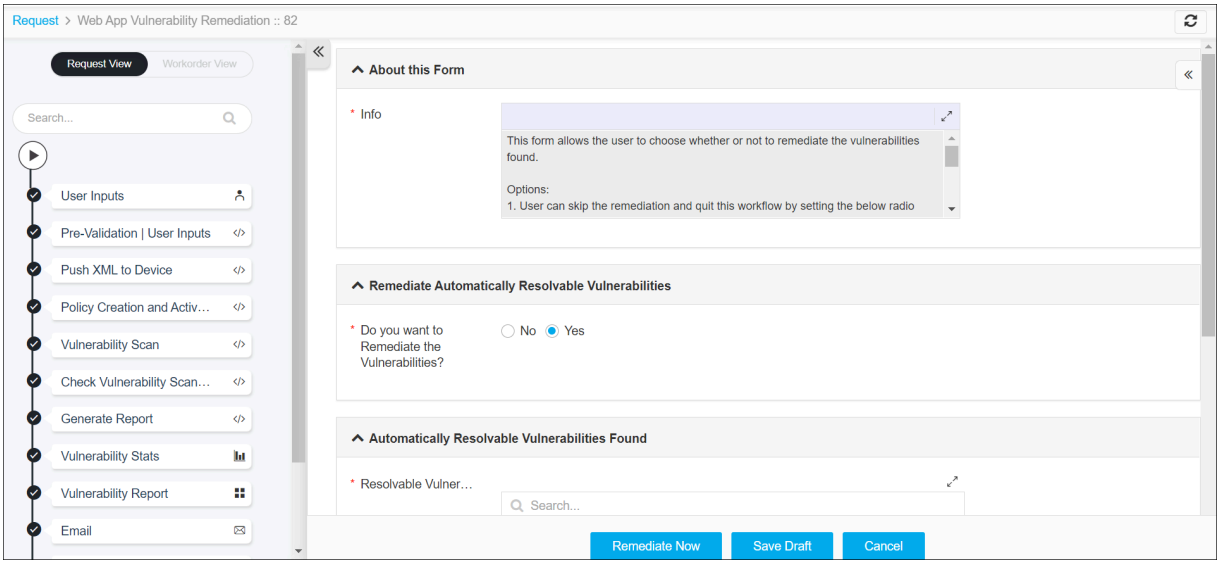


The report can be downloaded by clicking the Download () button.

13. Click the **Submit** button.

    The workflow reached the **Choice to Remediate** stage.

14. Select the option to resolve the vulnerabilities automatically or not.



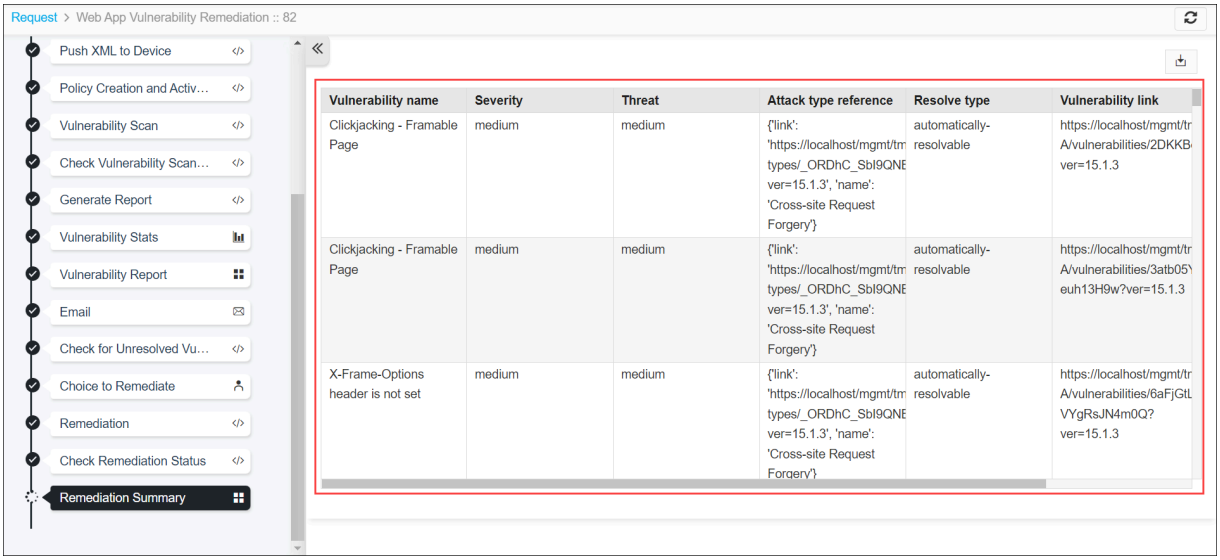
If you select

- **Yes**, the identified vulnerabilities will be resolved.
- **No**, the vulnerabilities will not be resolved.

15. Click **Remediate Now**.

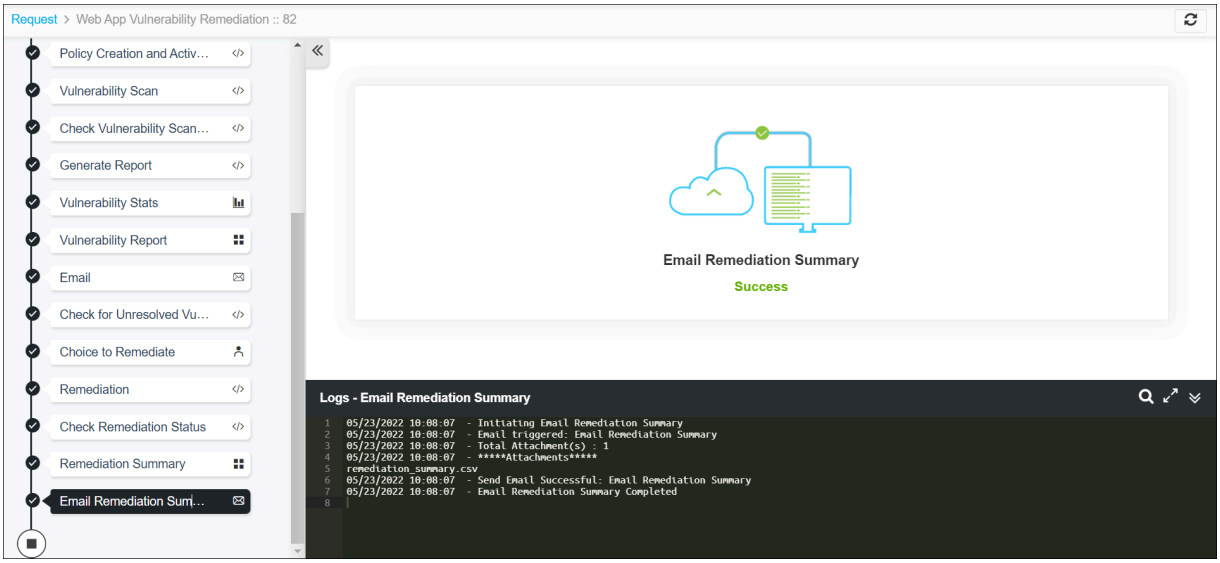
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

16. View the remediation summary under the **Remediation Summary** tab.



The report can be downloaded by clicking the Download () button.

17. It takes a while for the workflow completion and the email is triggered with remediation report.



Note:

The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

F5 BIG-IP

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Refer to the following workflows:

- F5 BIG-IP GTM
- F5 BIG-IP LTM

Infoblox IPAM

Prerequisites

Ensure the devices are added under Device Inventory > DNS.

Create Infoblox DNS Records – Basic

This workflow creates records in Infoblox and multiple record types (A, HOST, PTR, CNAME, TXT, MX) using the table. In the case of A, Host, and PTR record creation, IP can be fetched and reserved on the fly.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Infoblox IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create Infoblox DNS Records – Basic** workflow. The **Run** and **Schedule** buttons are displayed.
4. Click 

. The **Request > Create Infoblox DNS Records – Basic :: FormBuilder** page is displayed.
5. Enter/select **Device Info**.

Device Info Section - Field and Description Table

Field	Description
***Device Name**	Select the device name from the drop-down list.

6. Enter/select **Record Info**.

Record Info Section - Field and Description Table

Field	Description
***Network View**	Select the network view from the drop-down option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***DNS View**	Select the DNS view from the drop-down option.
***Record Type**	Select the record types from the drop-down option. The record types are A, HOST, PTR, CNAME, TXT, MX). In the case of A, Host, and PTR record creation, IP can be fetched and reserved on the fly. The record type A is shown.
***TTL**	Specify TTL (time to live) settings for Infoblox host records and resource records. TTL is the time that a name server is allowed to cache data. After the TTL expires, the name server is required to update the data.

7. If you cancel this request, remember to free the reserved IP by clicking the **Un-reserve IP** button.
8. Click **Submit**.
   The **Confirmation** dialog box is displayed.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

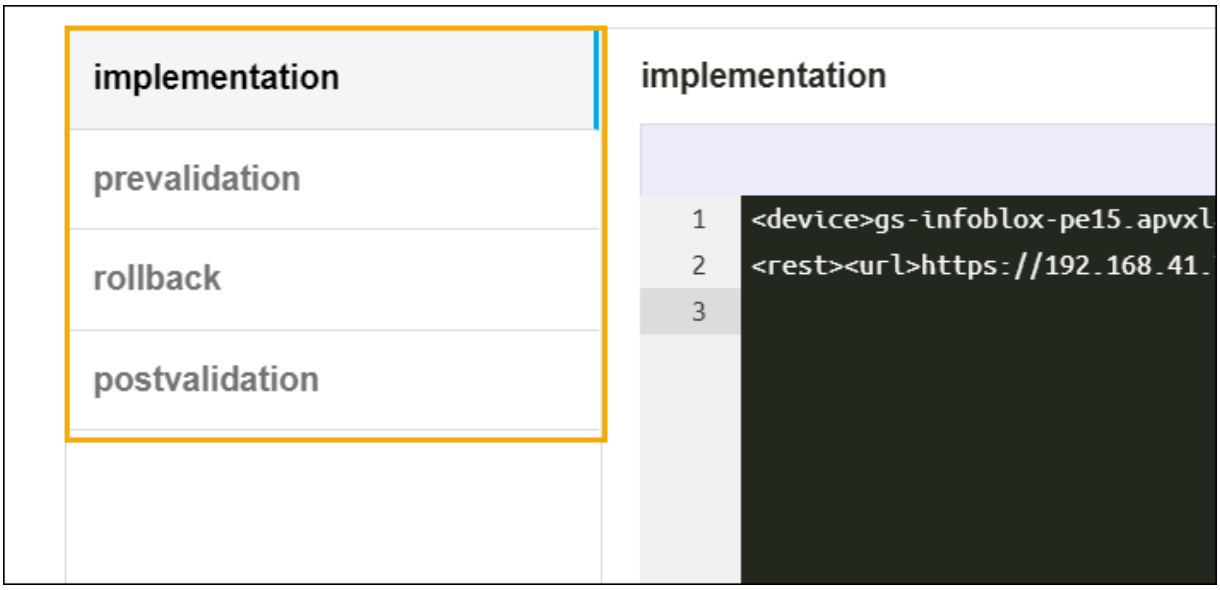
   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

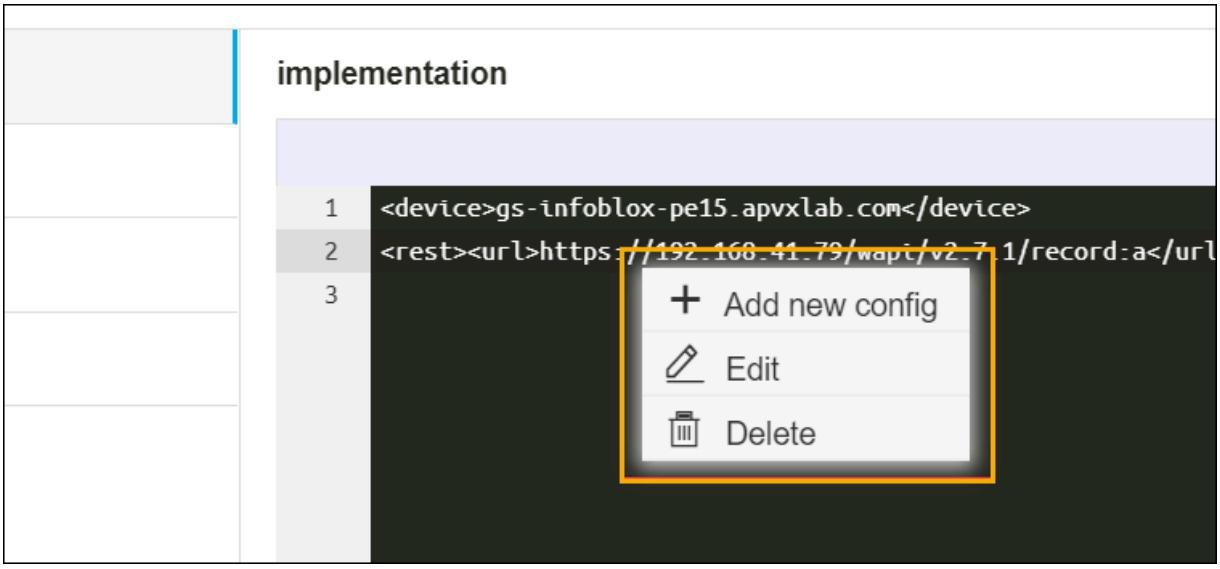
   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



11. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.



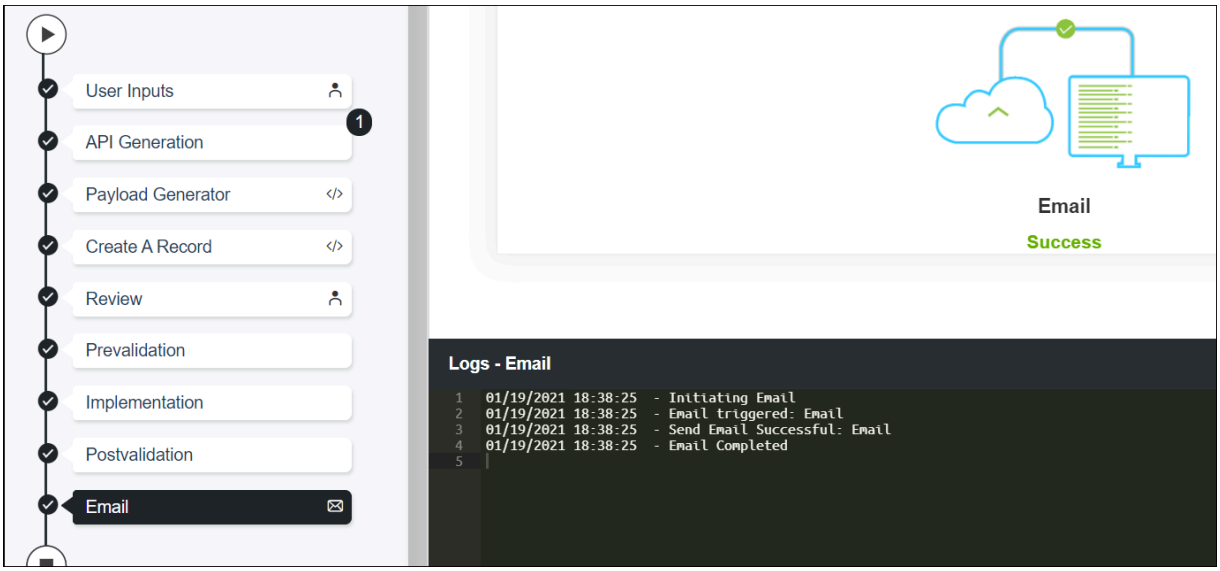
12. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
13. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    When the workflow is created, an email is triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

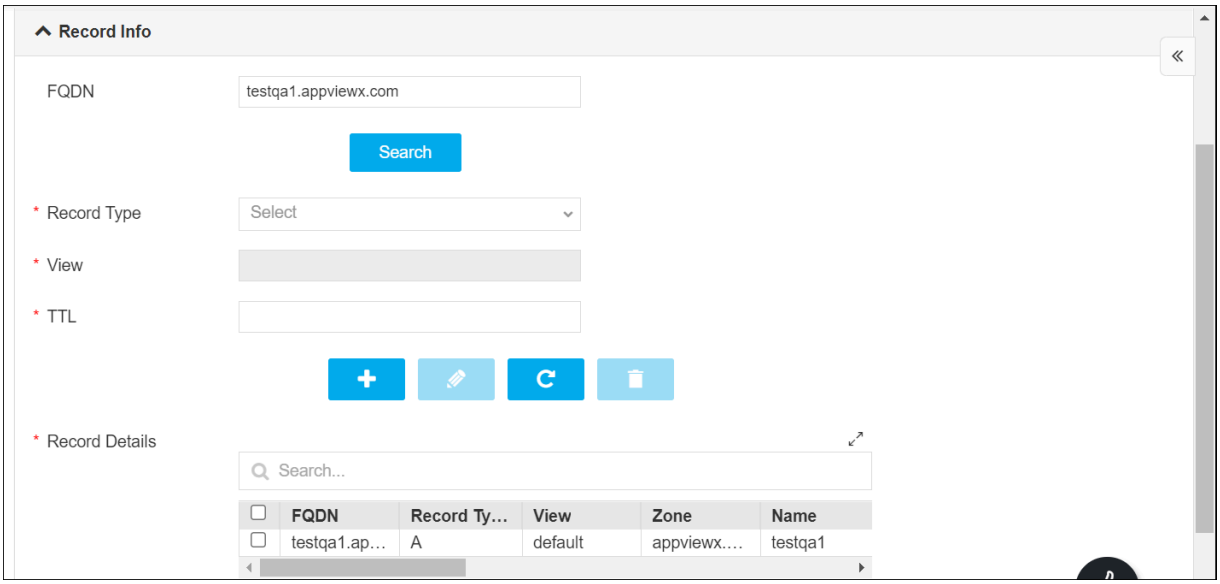
Modify Infoblox DNS Records - Basic

This workflow modifies existing records in Infoblox by fetching the existing records based on App Name. By using this workflow, you can modify multiple (A, HOST, PTR,CNAME,TXT,MX) records using the table.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Infoblox IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify Infoblox DNS Records - Basic** workflow. The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Modify Infoblox DNS Records - Basic :: FormBuilder** page is displayed.
5. Provide the necessary details of DNS record for a device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Click **Submit**.

   The **Confirmation** dialog box is displayed.

7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

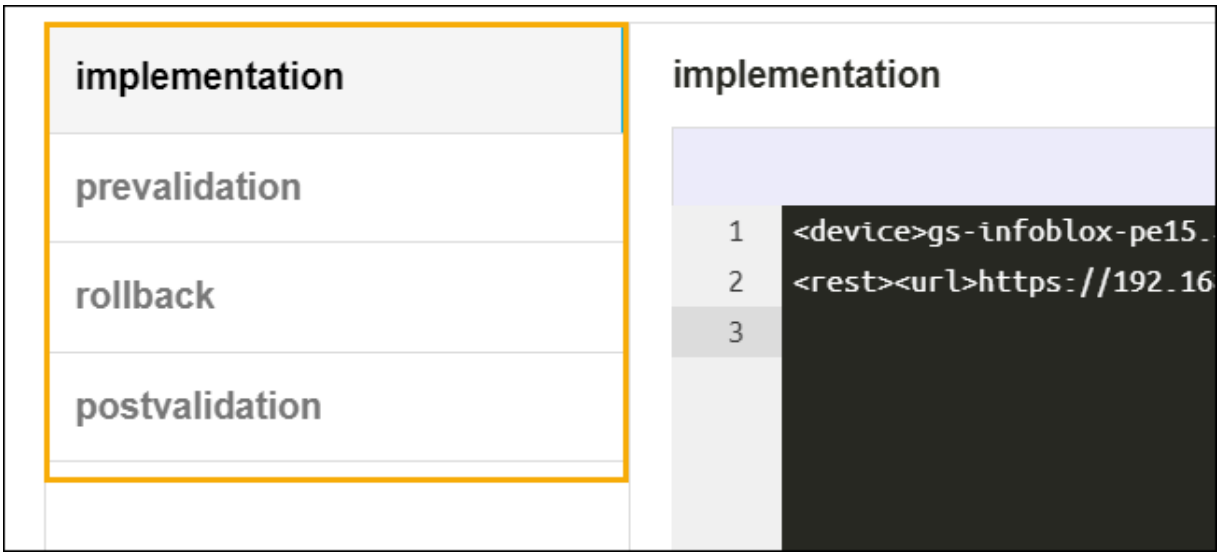
   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

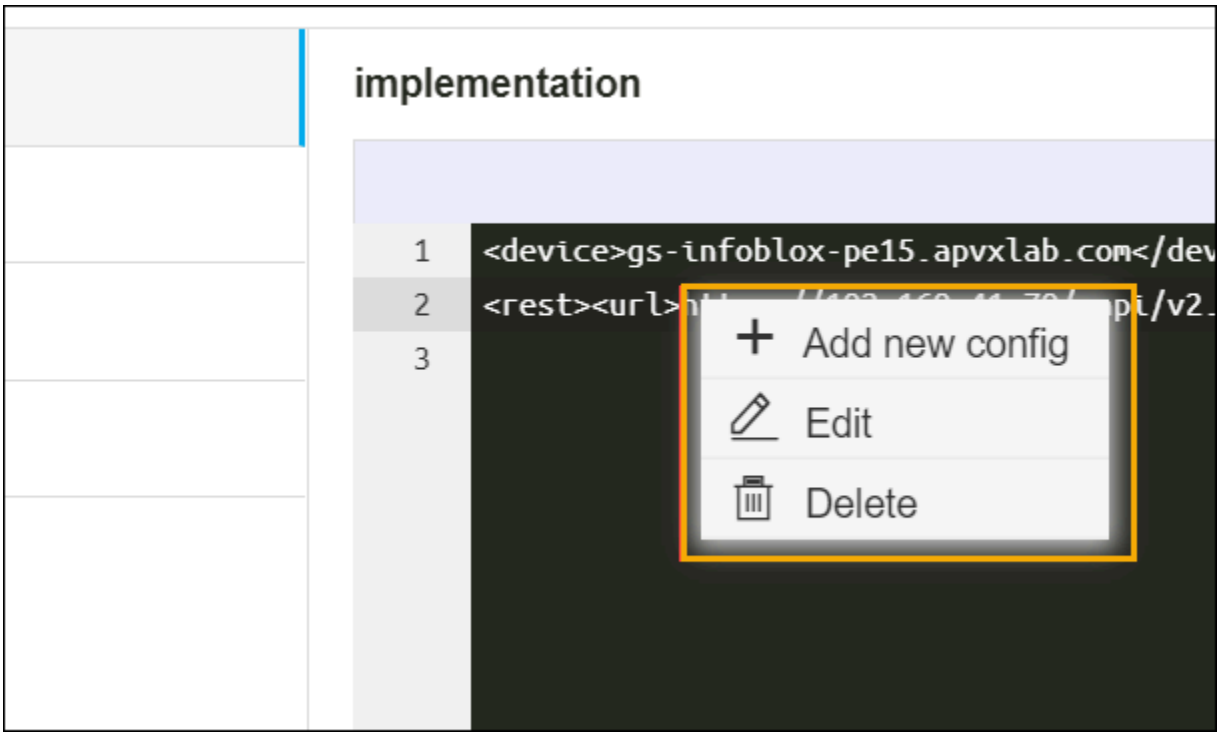
   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



9. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.



10. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

11. To continue with the workflow creation, click **Ok**

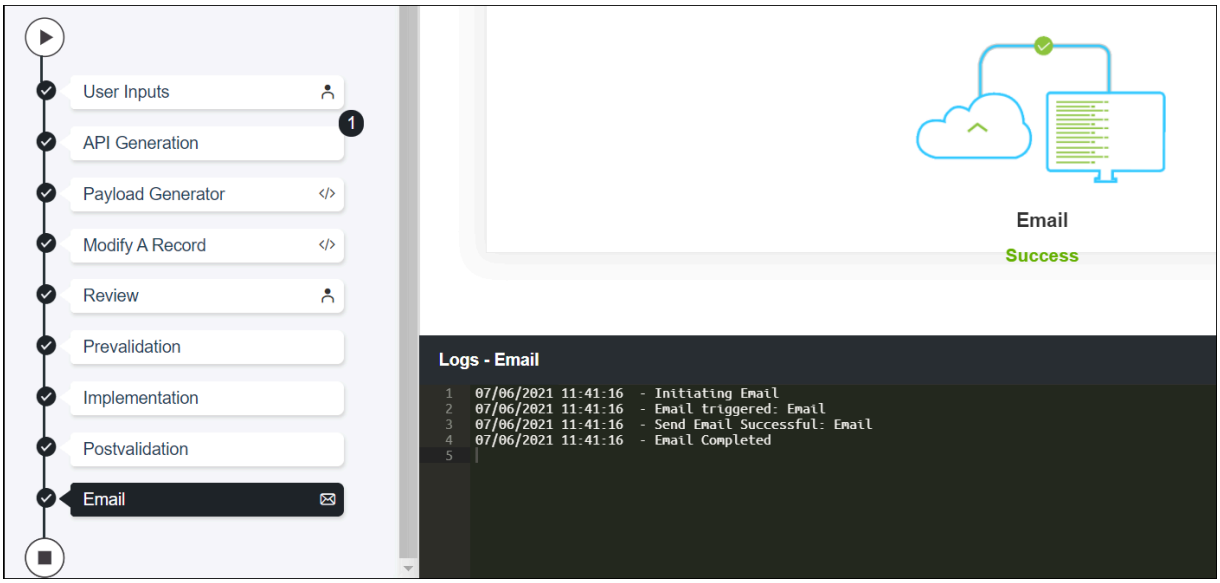
    OR

    To abort the workflow creation, click **Cancel**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

It takes a while to complete the post-validation.

When the workflow is created, an email is triggered to all the configured email IDs.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

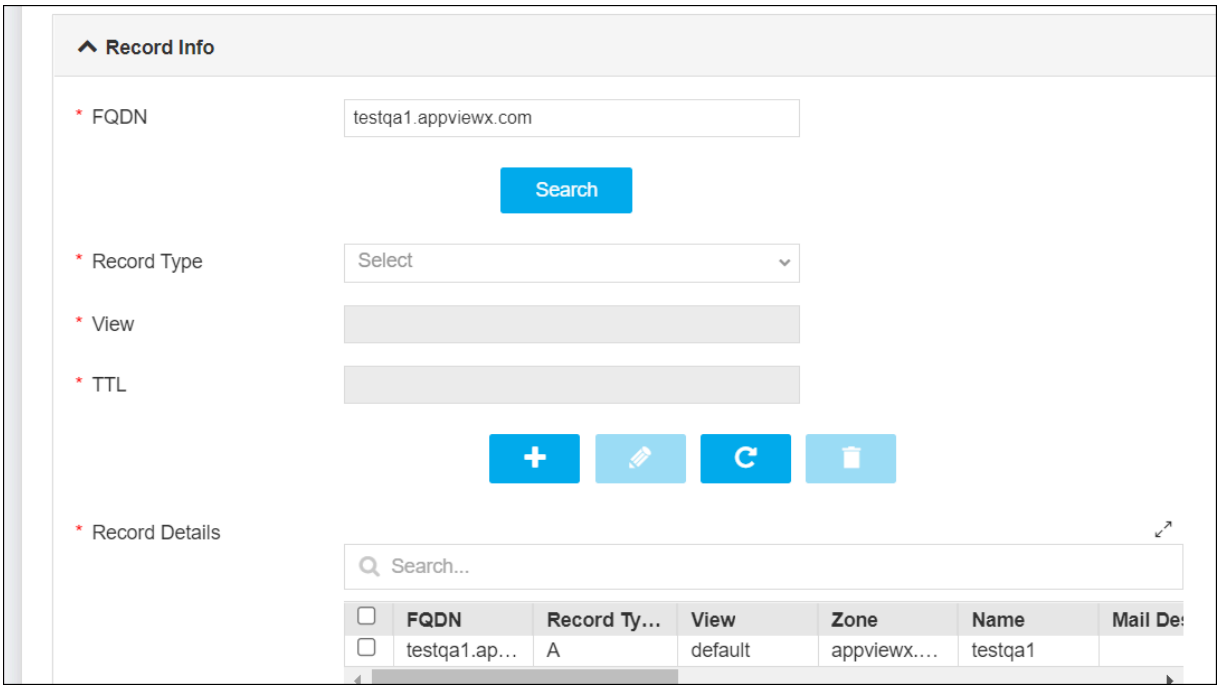
Delete Infoblox DNS Records - Basic

This workflow deletes existing records in Infoblox by fetching the existing records based on App Name. By using this workflow, you can modify multiple (A, HOST, PTR, CNAME, TXT, MX) records using the table.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Infoblox IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete Infoblox DNS Records - Basic** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete Infoblox DNS Records - Basic :: FormBuilder** page is displayed.
5. Provide the DNS record details of the device are to be deleted.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Click **Submit**.

   The **Confirmation** dialog box is displayed.

7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

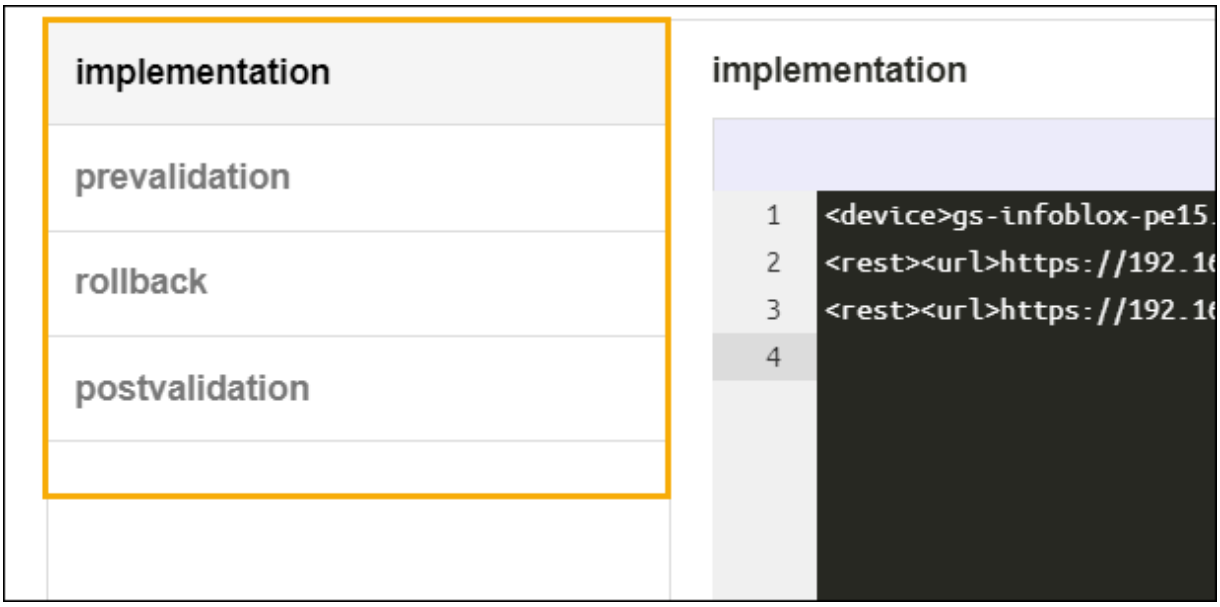
   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

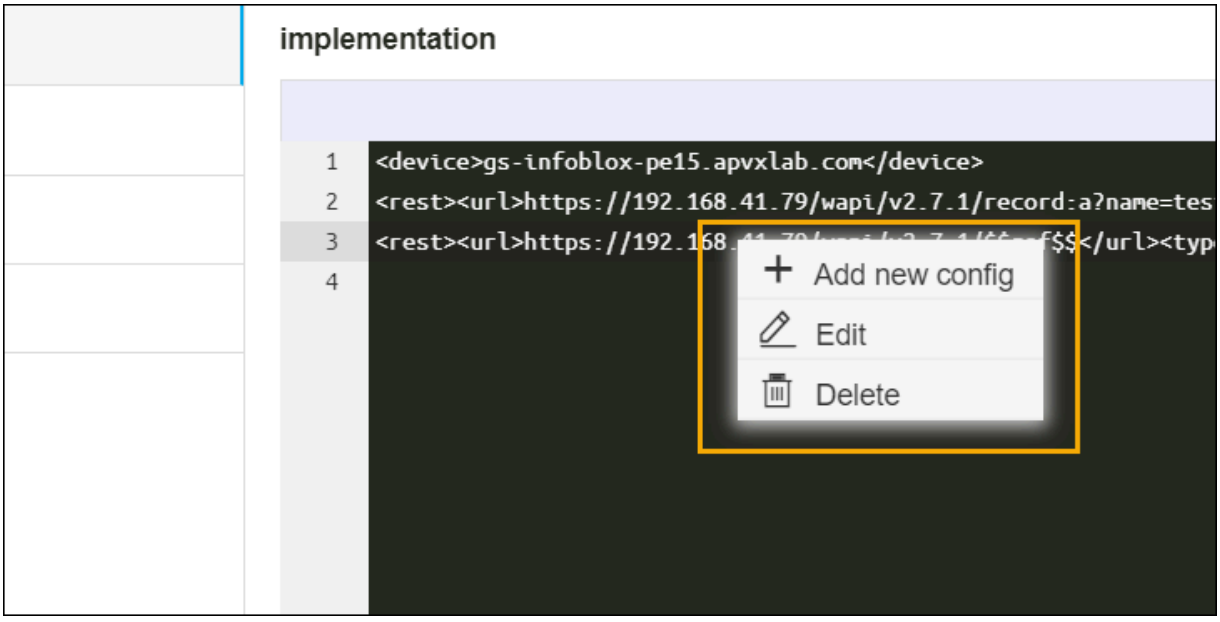
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



9. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.



10. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

11. To continue with the workflow creation, click **Ok**

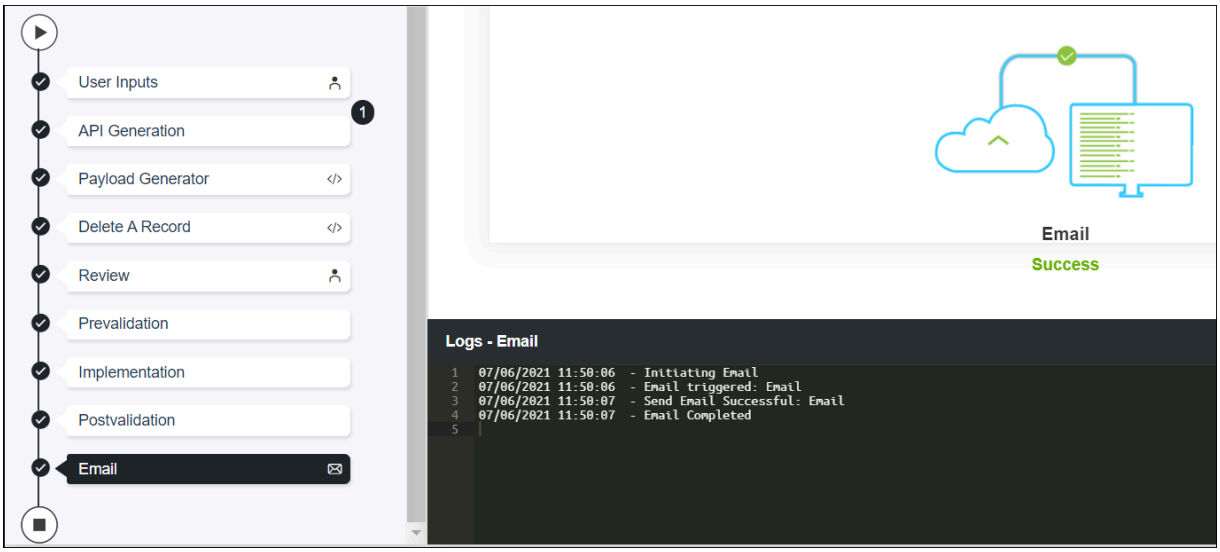
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

When the workflow is created, an email is triggered to all the configured email IDs.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Modify Infoblox DNS Record – Advanced

This workflow modifies the existing records in Infoblox by fetching the existing records based on FQDN. By using this workflow, you can modify multiple (A, HOST, PTR,CNAME,TXT,MX) records using the table.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Infoblox IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify Infoblox DNS Record – Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Modify Infoblox DNS Record – Advanced :: FormBuilder** page is displayed.
5. Modify the necessary details of existing DNS records for the device in Infoblox.
6. Click **Submit**.
   The **Confirmation** dialog box is displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

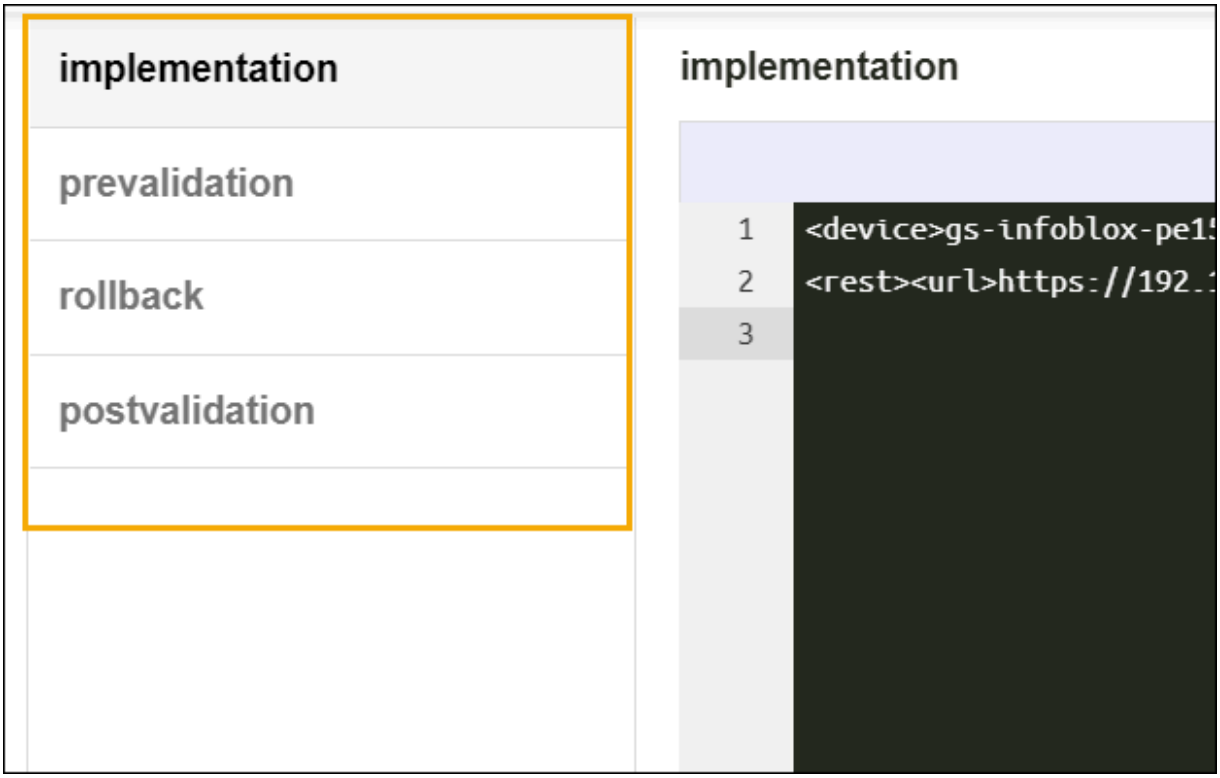
   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

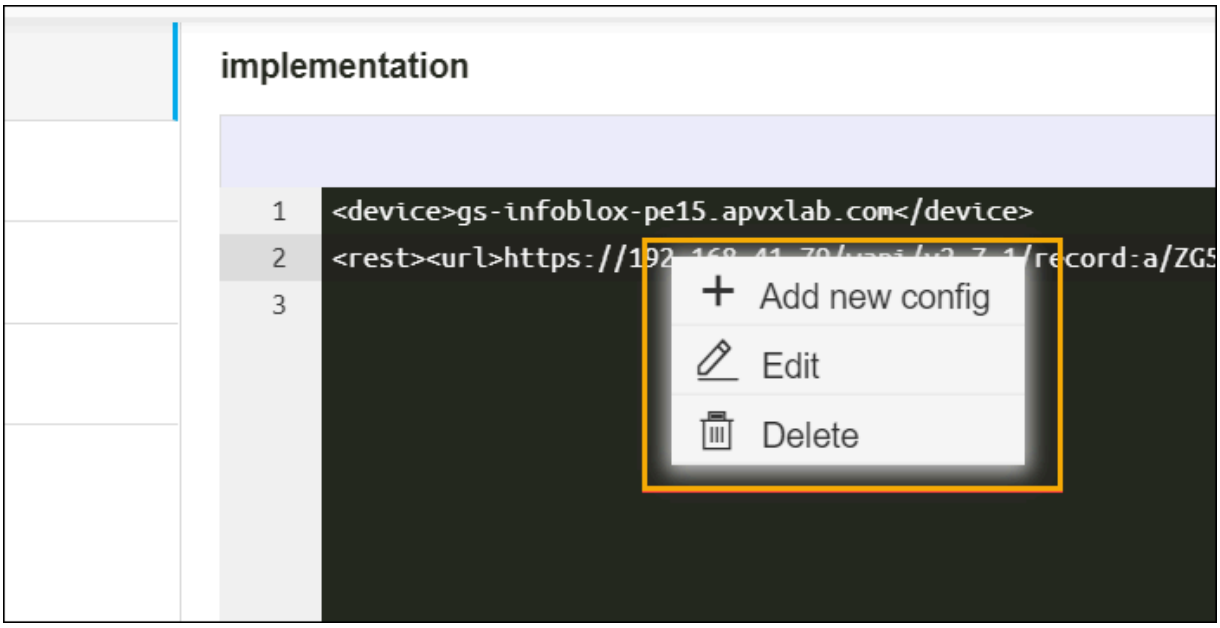
   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



9. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



10. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.

11. To continue with the workflow creation, click **Ok**

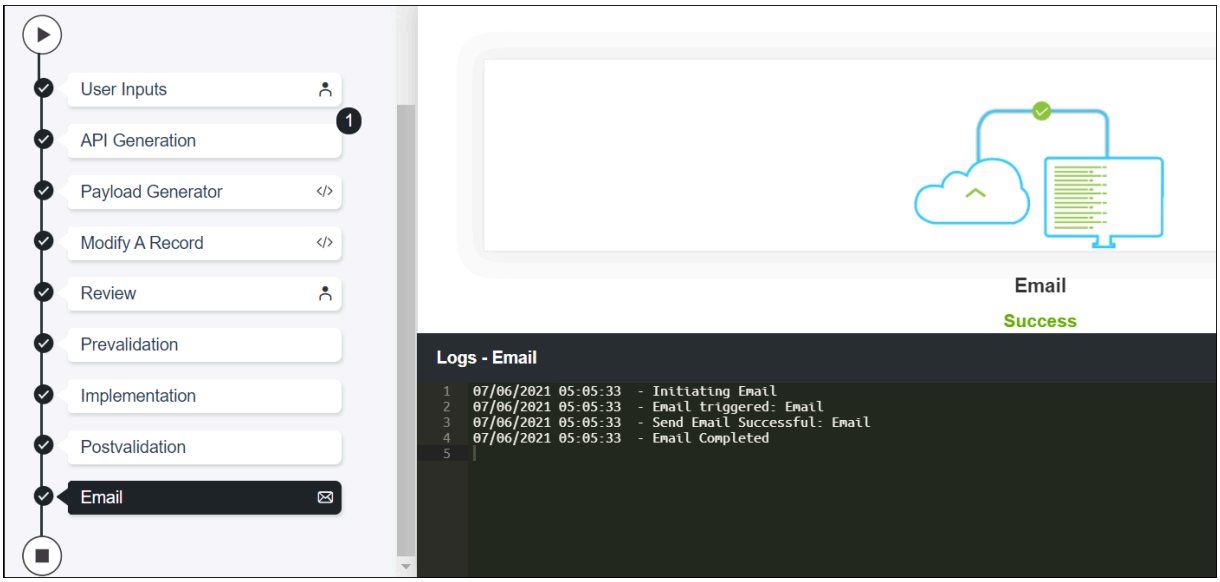
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    When the workflow is created, an email is triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Delete Infoblox DNS Records – Advanced

This workflow deletes existing records in Infoblox by fetching the existing records based on FQDN. By using this workflow, you can delete multiple (A, HOST, PTR, CNAME, TXT, MX) records.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Infoblox IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete Infoblox DNS Records – Advanced** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete Infoblox DNS Records – Advanced :: FormBuilder** page is displayed.
5. Provide the DNS record details of the device are to be deleted.
6. Click **Submit**.
   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

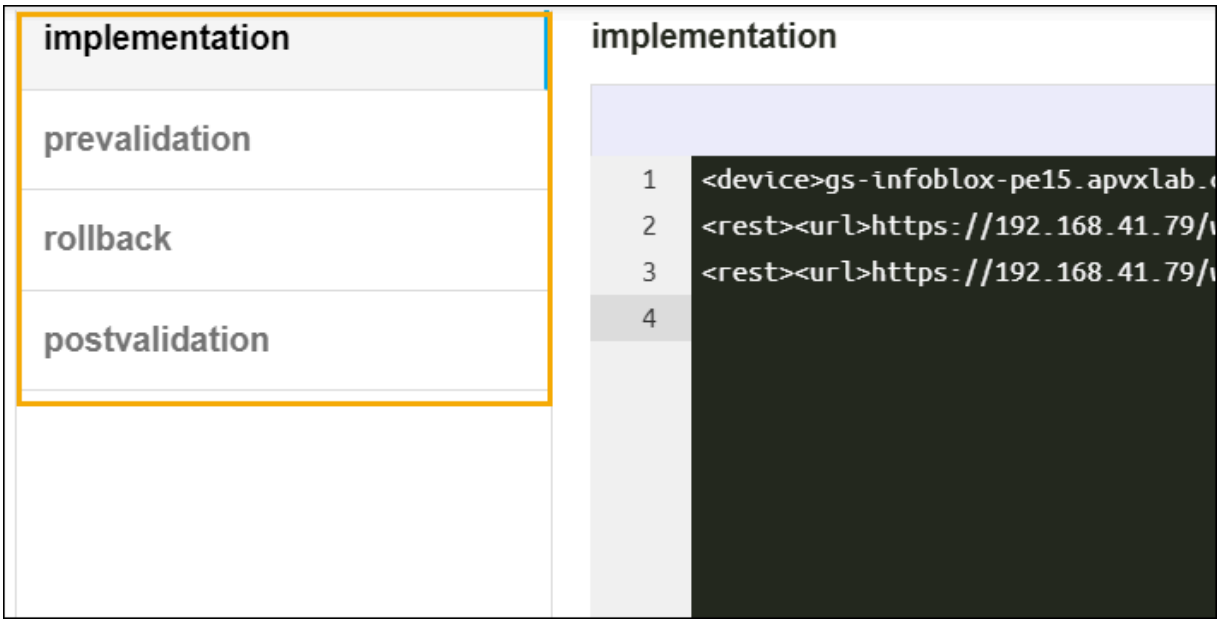
OR

To submit the form, click **OK**.

The validation starts automatically and reaches the **Review** stage.

> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



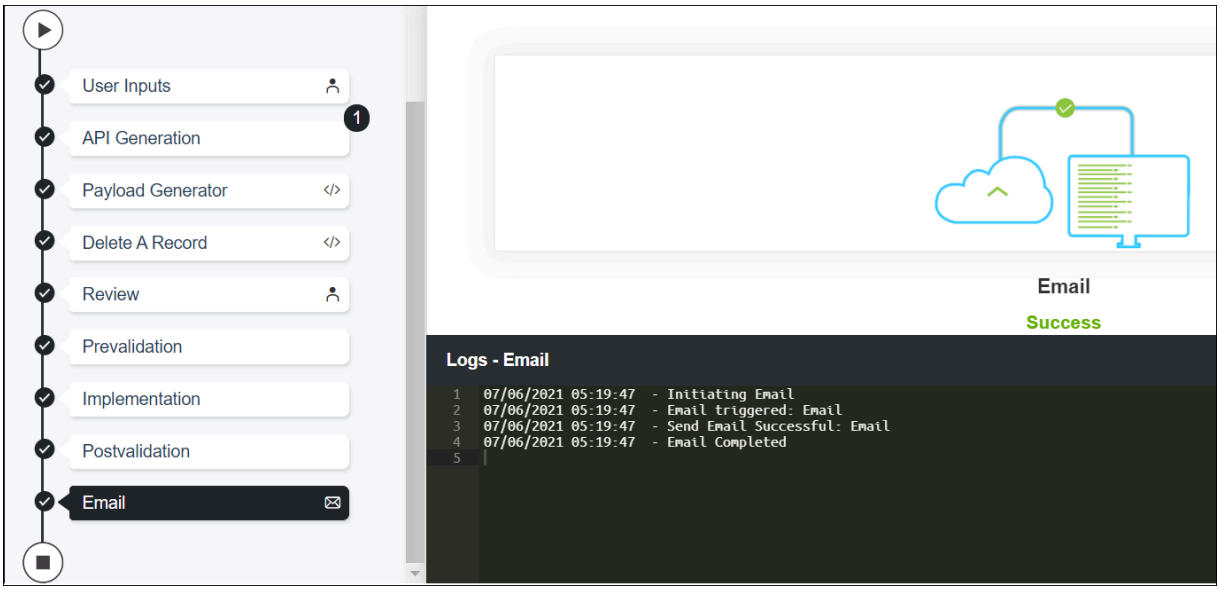
9. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.
10. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
11. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    When the workflow is created, an email is triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Bluecat IPAM

Prerequisites

Ensure the devices are added under Device Inventory > DNS.

Create BlueCat DNS Records

This workflow creates records in Bluecat. Also, it creates multiple (A, HOST, PTR,CNAME,TXT,MX) records. In the case of A, Host, PTR record creation, IP can be fetched and reserved on the fly.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Bluecat IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create BlueCat DNS Records** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create BlueCat DNS Records :: FormBuilder** page is displayed.
5. Enter/select **Device Info**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Info Section - Field and Description Table

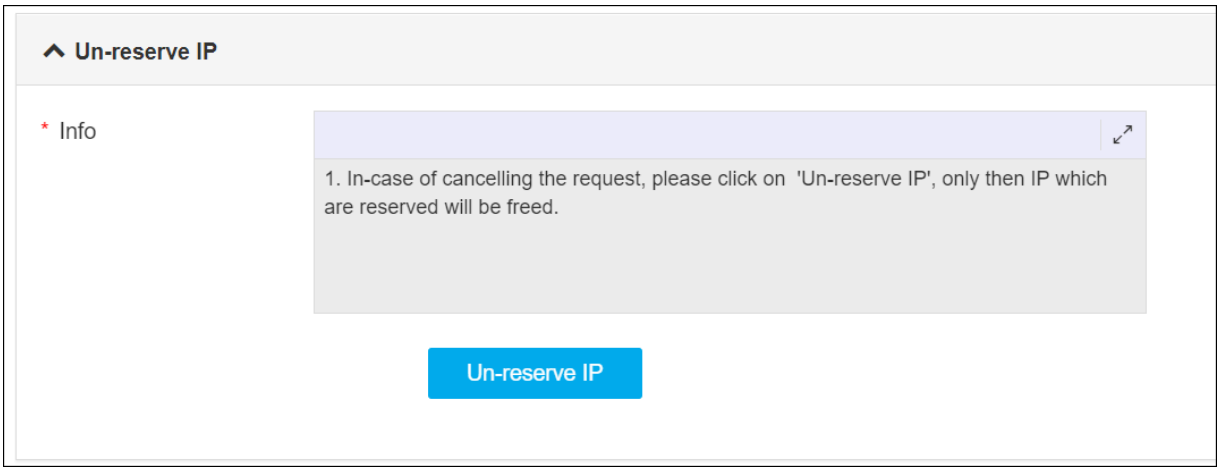
Field	Description
***Device Name**	Select the device name from the drop-down list.

6. Enter/select **Record Info**.

Record Info Section - Field and Description Table

Field	Description
***Network View**	Select the network view from the drop-down option.
***DNS View**	Select the DNS view from the drop-down option. DNS view is the container object for DNS zones and resource records.
***Zone**	Select the zone from the drop-down option. DNS zones represent your DNS zone of authority. For example, appviewx.com.
***Record Type**	Select the record types from the drop-down option. The record types are A, HOST, PTR, CNAME, TXT, MX). In the case of A, Host, and PTR record creation, IP can be fetched and reserved on the fly. Provide the details for record type.
***TTL**	Specify TTL (time to live) settings for Infoblox host records and resource records. TTL is the time that a name server is allowed to cache data. After the TTL expires, the name server is required to update the data.

7. If you cancel this request, remember to free the reserved IP by clicking the **Un-reserve IP** button.



8. Click **Submit**.

   The **Confirmation** dialog box is displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

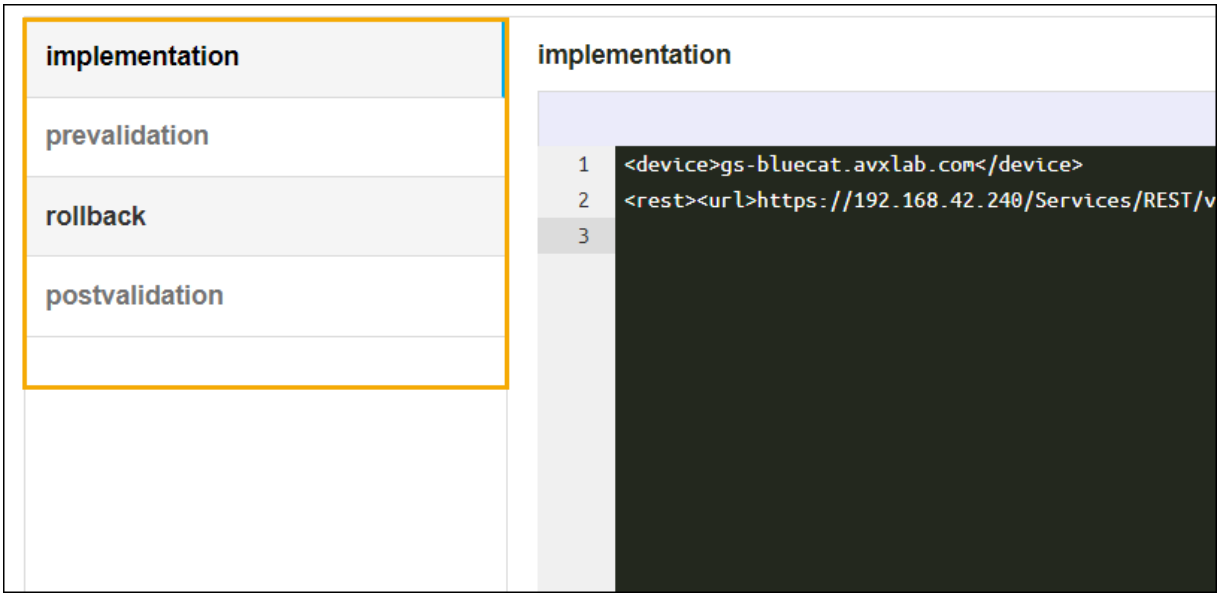
   OR

   To submit the form, click **OK**.

   The validation starts automatically and reaches the **Review** stage.

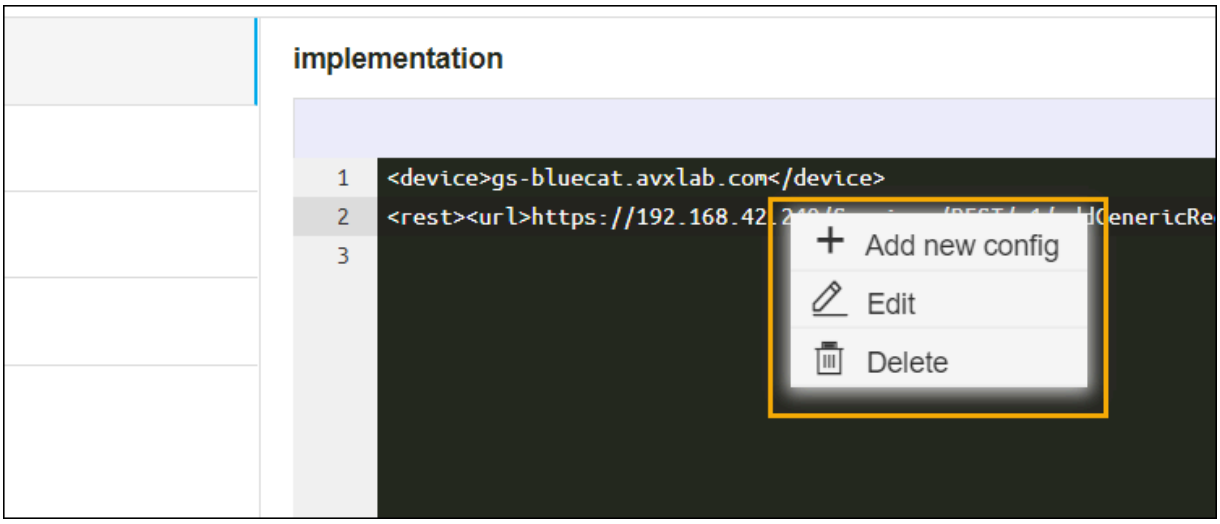
   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

10. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



11. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



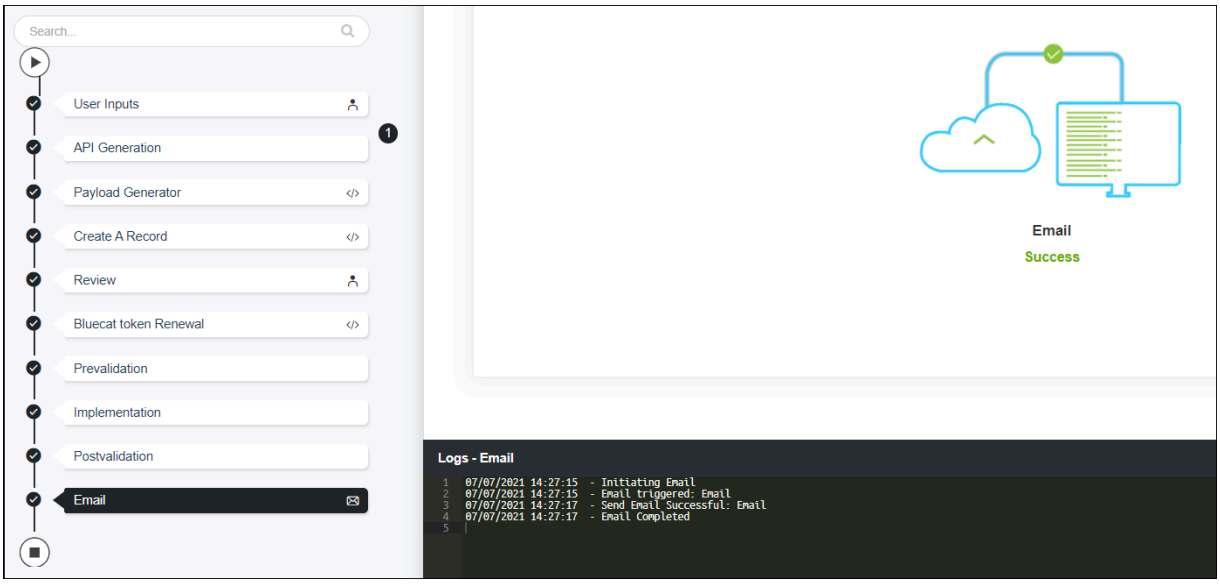
12. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
13. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    When the workflow is created, an email is triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

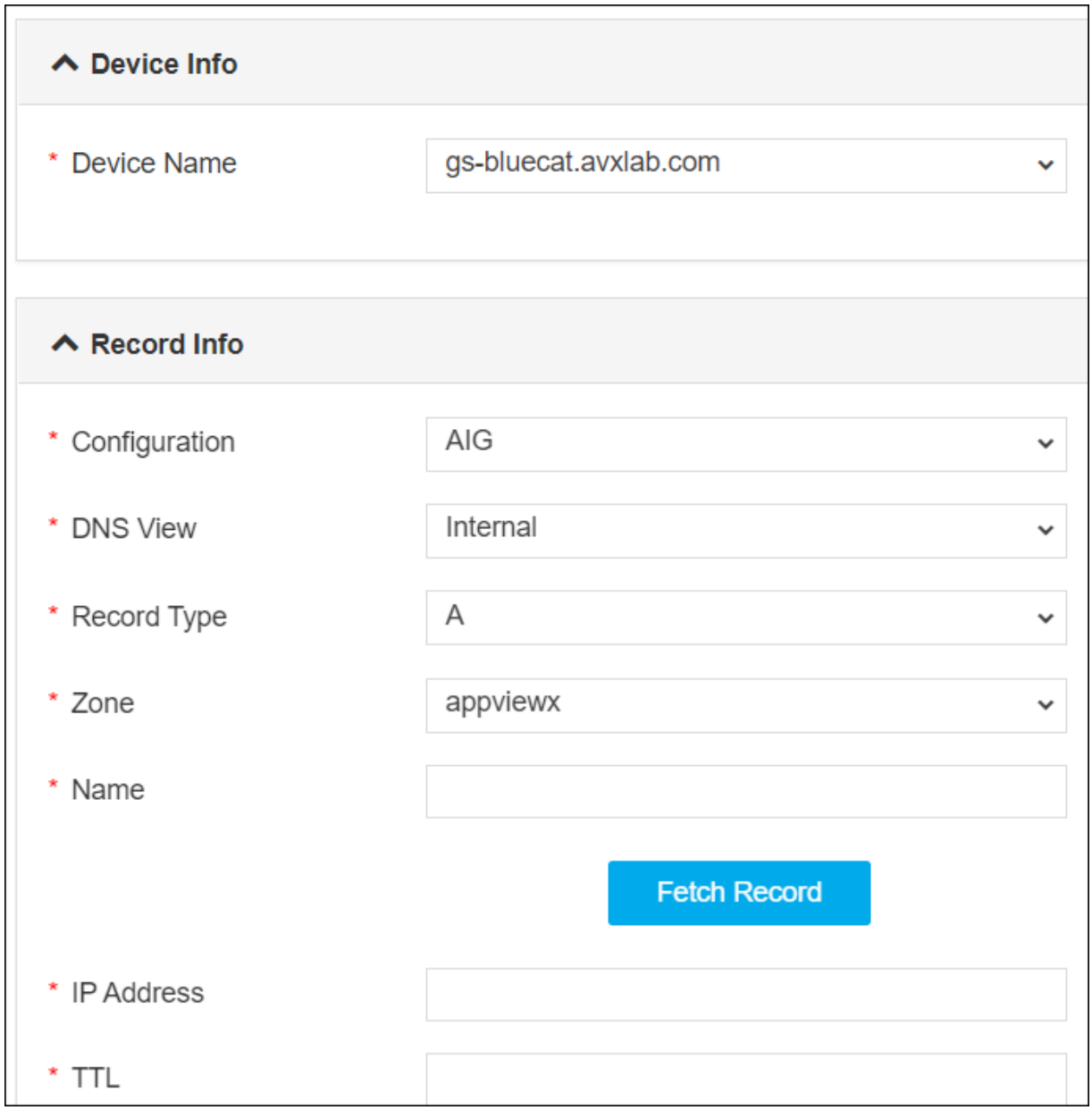
Modify BlueCat DNS Records

This workflow modifies existing records in Bluecat by fetching the existing records based on FQDN. It can also modify multiple (A, HOST, PTR, CNAME, TXT, MX) records.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Bluecat IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify BlueCat DNS Records** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Modify BlueCat DNS Records :: FormBuilder** page is displayed.
5. Provide the necessary details of DNS records for a device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Click **Submit**.

   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

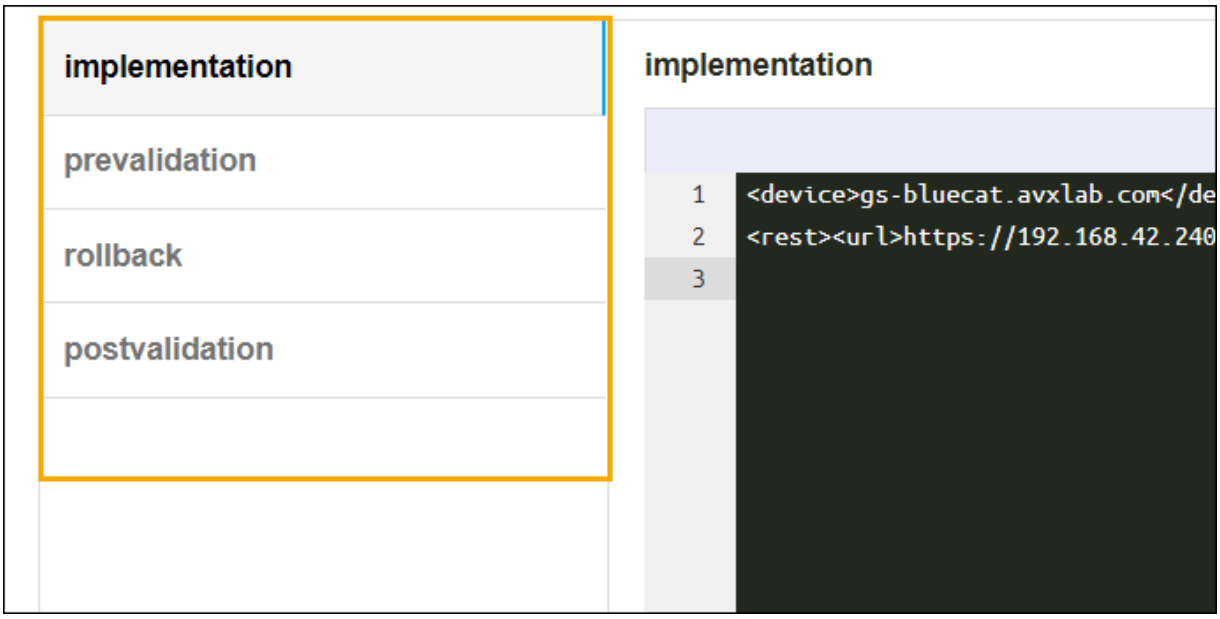
   The validation starts automatically and reaches the **Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

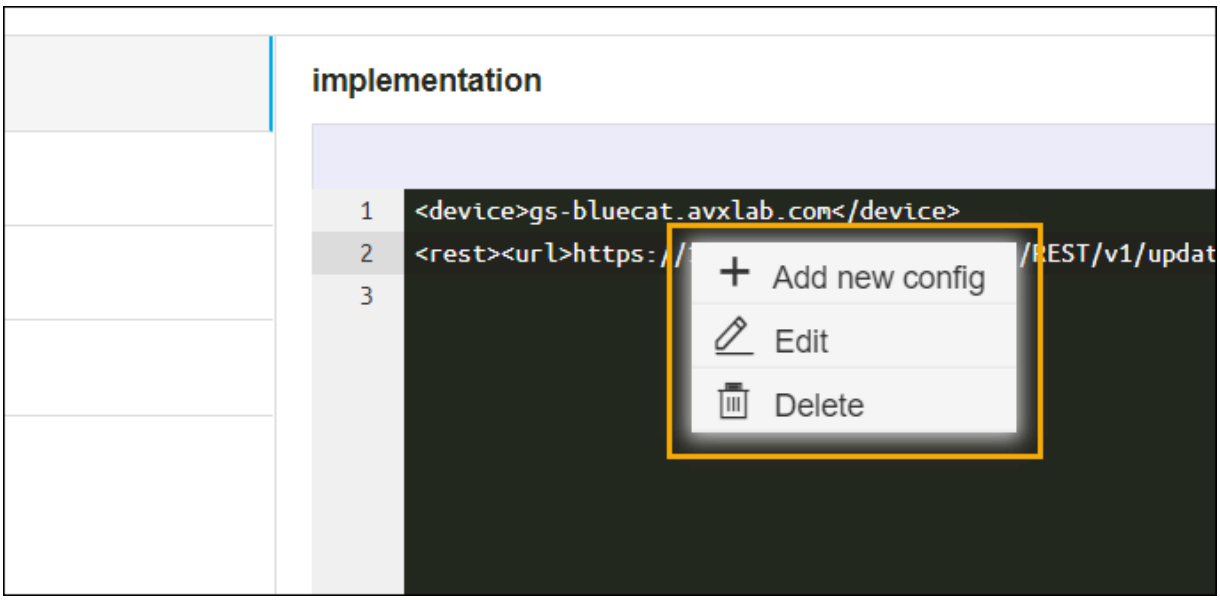
Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



9. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.



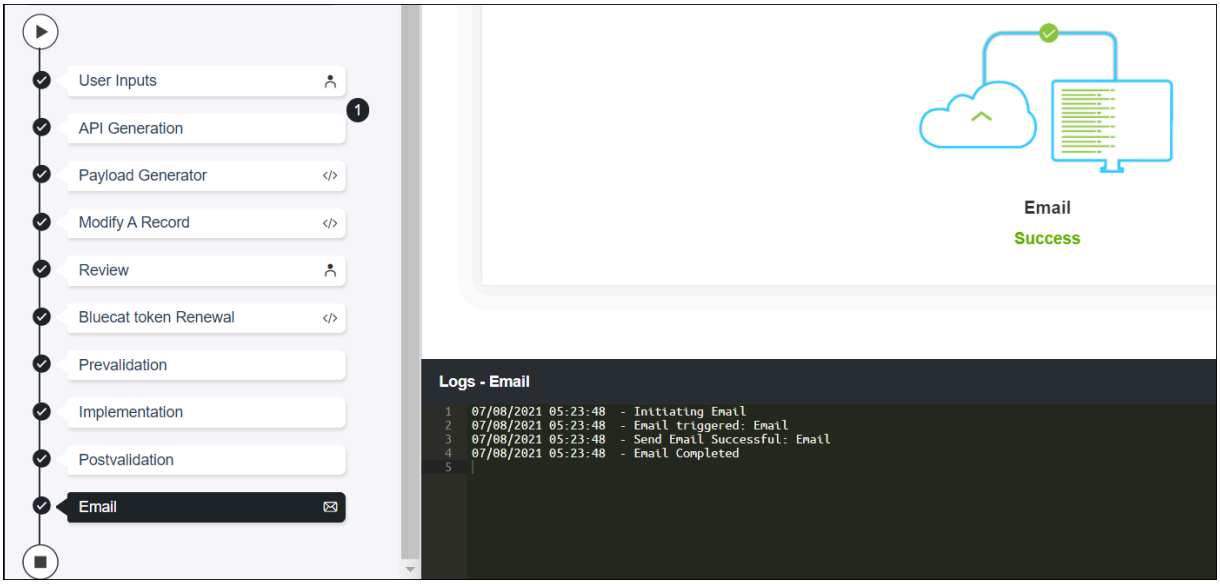
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. After the review, click **Submit**.
    The **Confirmation** dialog box is displayed.
11. To continue with the workflow creation, click **Ok**
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    When the workflow is created, an email is triggered to all the configured email IDs.



Note:
The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

Delete BlueCat DNS Record

This workflow deletes existing records in Bluecat by fetching the existing records based on FQDN. Also deletes multiple (A, HOST, PTR,CNAME,TXT,MX) records.

To run this workflow:

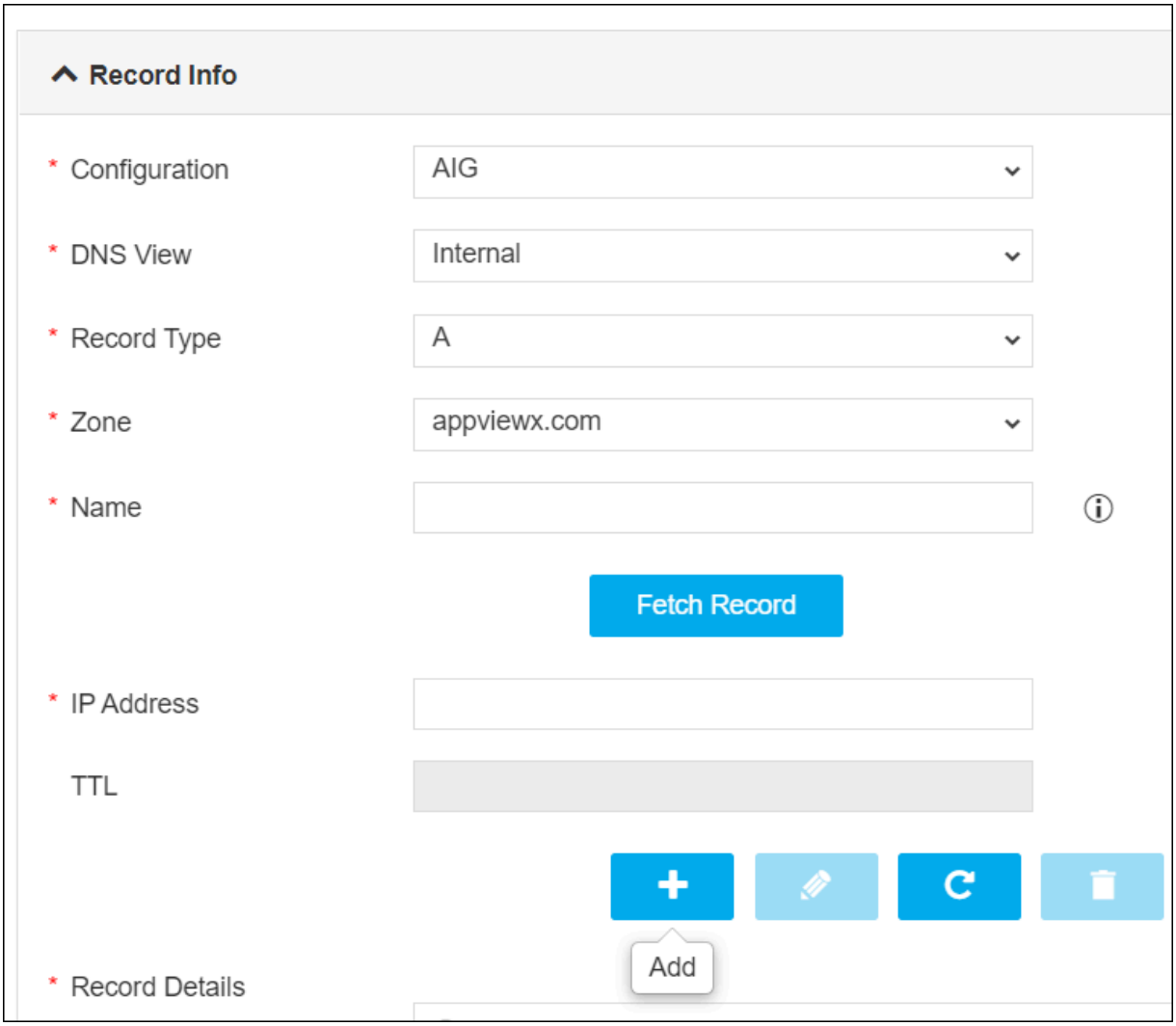
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Bluecat IPAM** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete BlueCat DNS Record** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click Run.

   The **Request > Delete BlueCat DNS Record :: FormBuilder** page is displayed.

5. Provide the necessary details of DNS records that are to be deleted.



6. Click **Submit**.

   The **Confirmation** dialog box is displayed.

7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

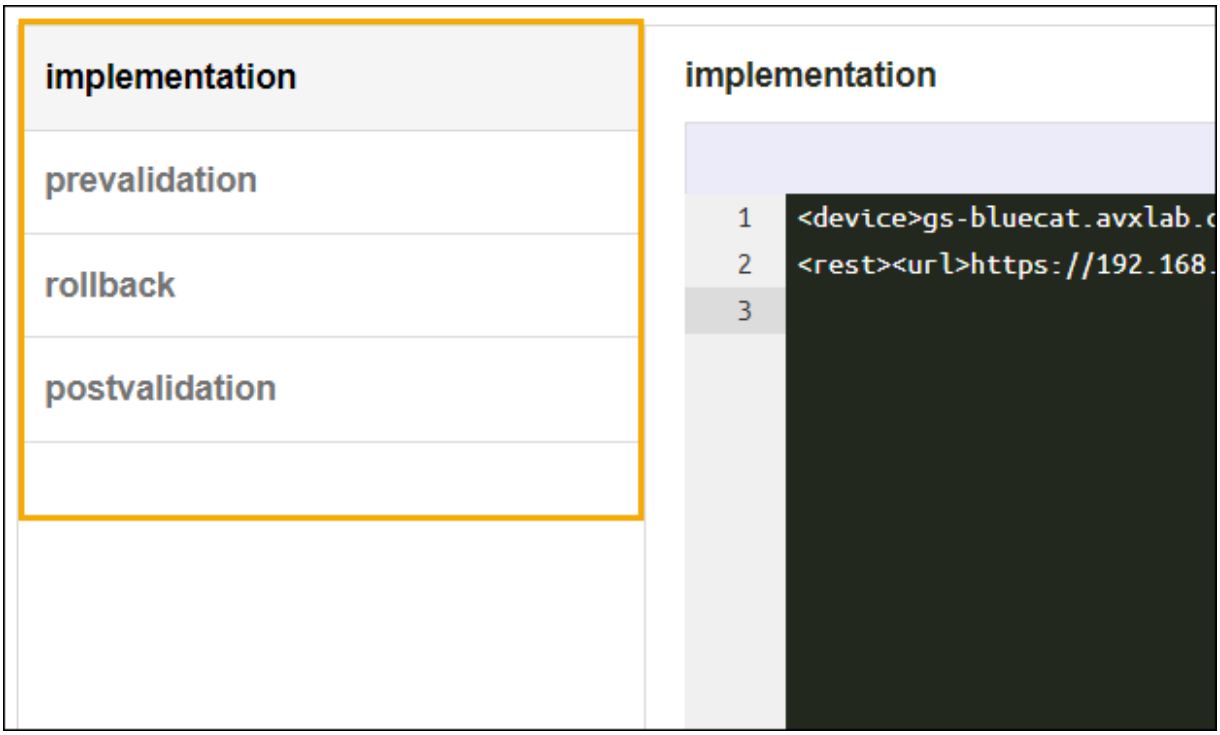
   The validation starts automatically and reaches the **Review** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

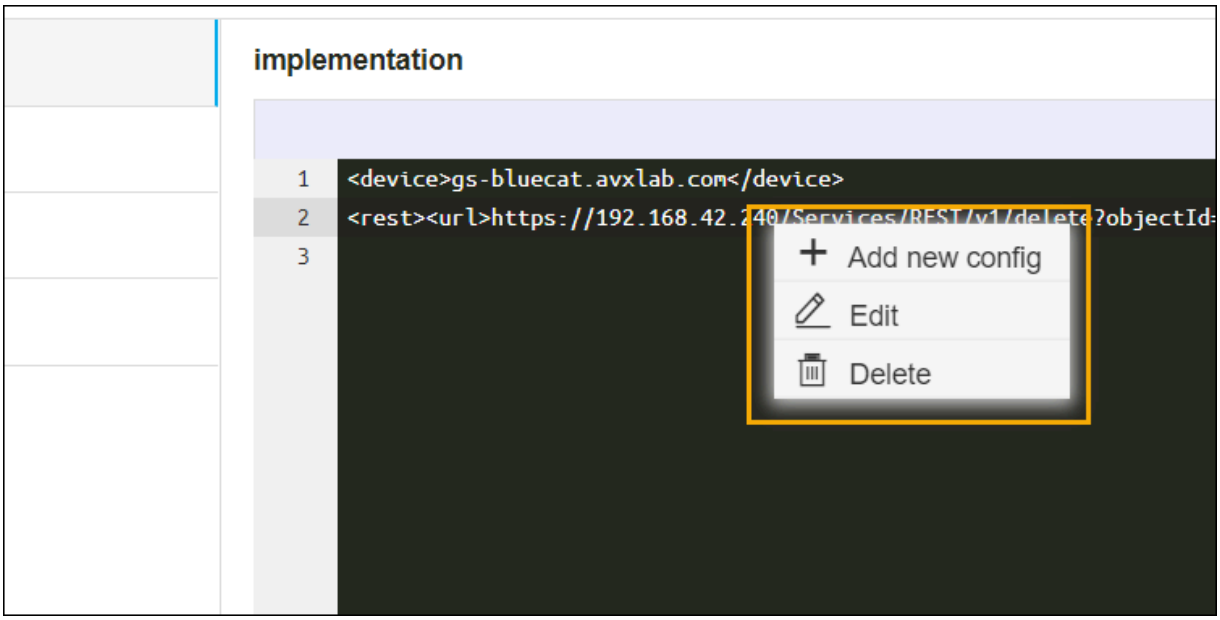
- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



9. (Optional) If you need to change any data at this stage, right click, as shown in the image below, and select the required option.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



10. After the review, click **Submit**.

    The **Confirmation** dialog box is displayed.
11. To continue with the workflow creation, click **Ok**

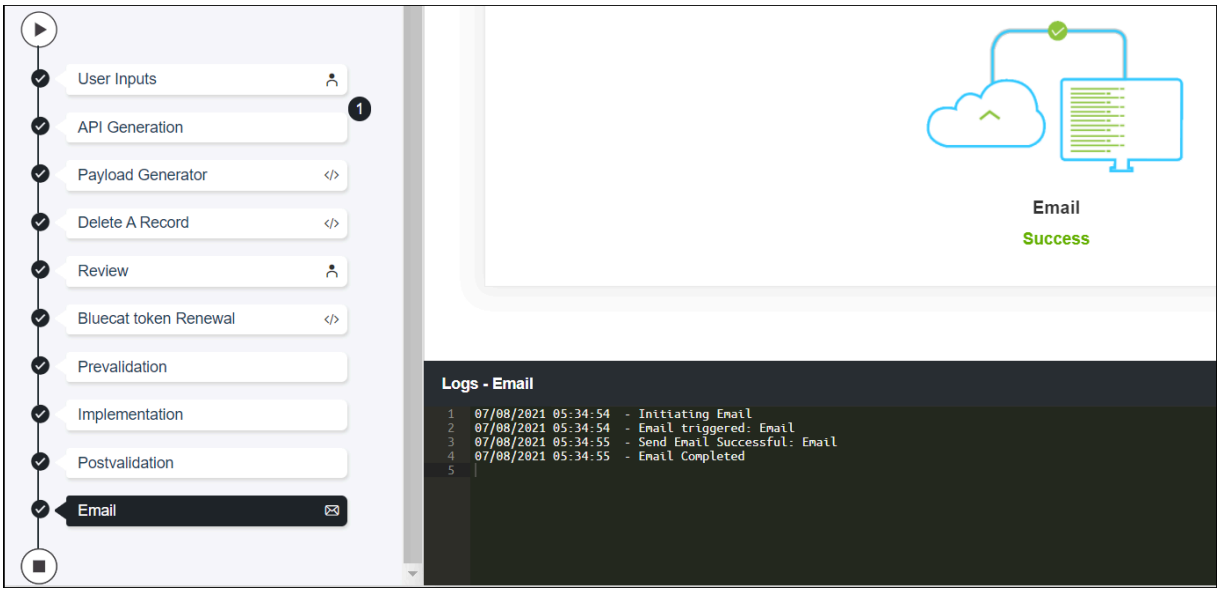
    OR

    To abort the workflow creation, click **Cancel**.

    It takes a while to complete the post-validation.

    When the workflow is created, an email is triggered to all the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:

The validation stages are shown in the left side of the screen. To view a validation stage, click on the respective stage.

F5 BIG-IP System

Automate F5 BIG-IP Upgrades

The workflows grouped under this sub-category include:

Software Upgrade on Standalone F5 Devices

Overview

This workflow upgrades the BIG-IP version from the existing version to a higher version as recommended by F5. For more information, refer to https://support.f5.com/csp/article/K13845.

Prerequisite

The prerequisites are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- The ISO files for the software upgrade must be available in the path **/shared/images** of the F5-BIG IP in which the upgrade is to be performed.
- The User using which F5 BIG-IP is Managed in AppViewX should have Advance shell Access.
- For License reactivation, the F5 device should have Internet access.

Running a Software Upgrade on Standalone Devices

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Software Upgrade on Standalone F5 Devices** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Software Upgrade on Standalone F5 Devices :: FormBuilder** page is displayed.
5. Enter/select **Device Selection** details.

Device Selection Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

The fields in the **Device Details** section are auto-populated.

6. Enter/select **Image and Volume Details**.

Field description for the Image and Volume Details section

Field	Description
***ISO File Name**	Click the retrieve button and then select a desired ISO file name from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***New Volume Name**	Enter the new volume name. The format of the volume name must be in the form of **HDX.Y**, **CFX.Y**, or **MDX.Y**.

7. Select **Renew License**.
   The possible values include:
   - **No**: This option does not allow you to renew the license. By default, this option is selected.
   - **Yes**: Select this option to generate
8. Click **Next**.
   The **Confirmation** dialog box is displayed.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

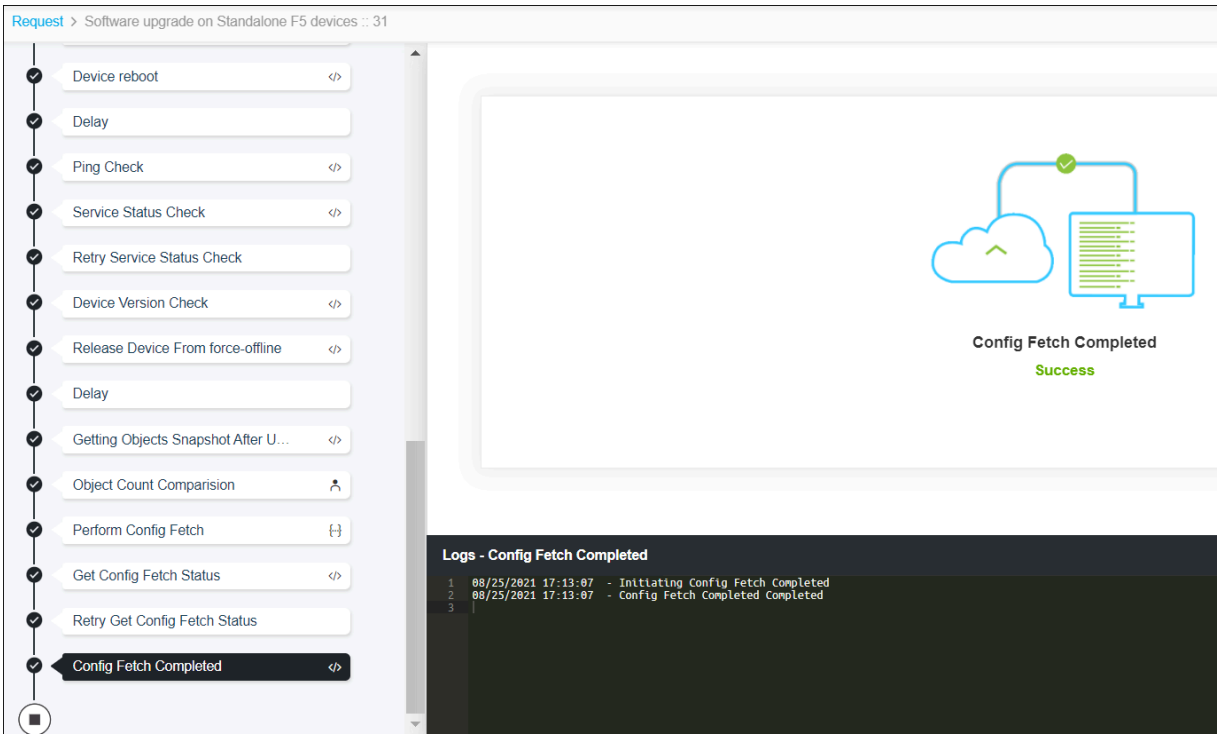
   To submit the form, click **OK**.

   The software upgrade starts. It takes a while to complete the software upgrade.

   Note:
   - A request is created for this workflow. To view the request, navigate to Menu > Request > All.
   - If there is any failure in running this workflow, start a new workflow by providing the correct data.
   - Alternatively, clone the request, update the correct input data in the form input, and submit the workflow.

10. After the manual check, click **Proceed**.
    The **Confirmation** dialog box is displayed.
11. Click **Ok** to submit the form.
12. After the **Object Count Comparison**, click **Proceed**.
    The **Confirmation** dialog box is displayed.
13. Click **Ok** to submit the form.
    The implementation process for software upgrade is completed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

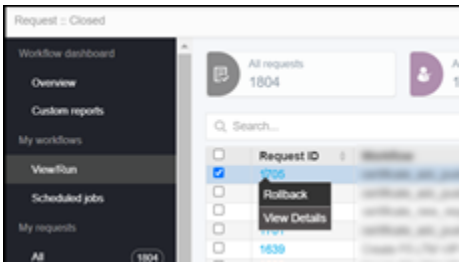


14. In case the request has to be reverted, execute the following steps:

    a. Go to the **Menu** > **Request** > **All**.

    b. Right-click the row on your desired Request ID.

    c. Select the **Rollback** option.



    d. Select **Yes** in the **Rollback** confirmation dialog box.

    A new request is triggered for performing rollback.

Software Upgrade on HA F5 Devices

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Overview

This workflow upgrades the BIG-IP version from the existing version to a higher version as recommended by HA F5. For more information, refer to https://support.f5.com/csp/article/K13845.

Prerequisites

The prerequisites are:

- The ISO files for the software upgrade must be available in both active and standby devices and the file should be in the path /shared/images of the F5-BIG IP in which upgrade to be performed.
- The User using which F5 BIG-IP is Managed in AppViewX should have Advance shell Access.
- For License reactivation, the F5 device should have Internet access.

Running Software Upgrade on HA F5 Devices

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Software Upgrade on F5 HA Devices** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Software Upgrade on F5 HA Devices :: FormBuilder** page is displayed.
5. Enter/select the **Device Selection** details.

Device Selection Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Standby Device**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Disk Space Threshold needed for installation	Specifies the threshold limit value for the installation. If the available disk space falls below the defined threshold value during installation, the installation process will be stopped automatically to prevent any potential issues due to insufficient disk space. The default threshold value is set to 15 units.
**: Mandatory fields*	

The fields in the **StandBy Device Details** and the **Active Device Details** sections are auto-populated when you click the **Fetch Standby device Installed Image** and **Fetch Active device Installed Image** buttons, respectively.

6. Enter/select **Image and Volume Details**.

Image and Volume Details Section - Field and Description Table

Field	Description
SFTP	When you use SFTP servers to store ISO files, you need to add the SFTP servers to the Inventory section under Others, with "sftp" selected as the model. The values are: • **Yes** - Select this option to store ISO files in SFTP servers. Then, select a SFTP sever from the **SFTP Server** drop-down menu and specify the path in the **ISO File Path** field. • **No** (default) - Select this option to store ISO files in AppViewX. For software upgrades, ensure that the ISO files are available in the following directory: **INSTALLATION_PATH/ appviewx_dependencies/vw/dependencies/bigip_images**. In a multi-node instance, make sure to upload ISO files to all nodes.
***ISO File Name**	Click the retrieve button, and then select a desired ISO file name from the dropdown list.
***New Volume Name**	Enter the new volume name. The format of the volume name must be in the form of HDX.Y, CFX.Y, or MDX.Y.
**: Mandatory fields*	

7. Verify the device and overall statuses under the **Device Optimality Validation** section to ensure that all devices are functioning optimally.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device Optimality Validation Section - Field and Description Table

Field	Description
Validate Device	Click this button to start the validation.
***Device Status**	Displays details about the status of the device.
***Overall Status**	Displays details about the overall status of the system.
**: Mandatory fields*	

8. Select **Generate Qkview File** option. Further options are:

   - If you select **Yes** for the **Generate Qkview File** option, you can then choose whether to upload the Qkview file to the iHealth Portal.
   - If you select **Yes** for **Upload to iHealth Portal**, you will need to provide the iHealth case number associated with the upload.
   - If you choose **No** for **Upload to iHealth Portal**, the Qkview file will not be uploaded to the iHealth Portal.

9. Select the following options in **Reactivate License Automatically**. field:
   - **No** (default): Select this option to renew the license manually.
   - **Yes**: Select this option to enables automatic license renewal. For automatic license reactivation, the F5 device must have Internet access.
10. Click **Next**.

    The **Confirmation** dialog box is displayed.
11. To save this form so you can edit it later, click **Save Draft** and then click **OK**.

    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

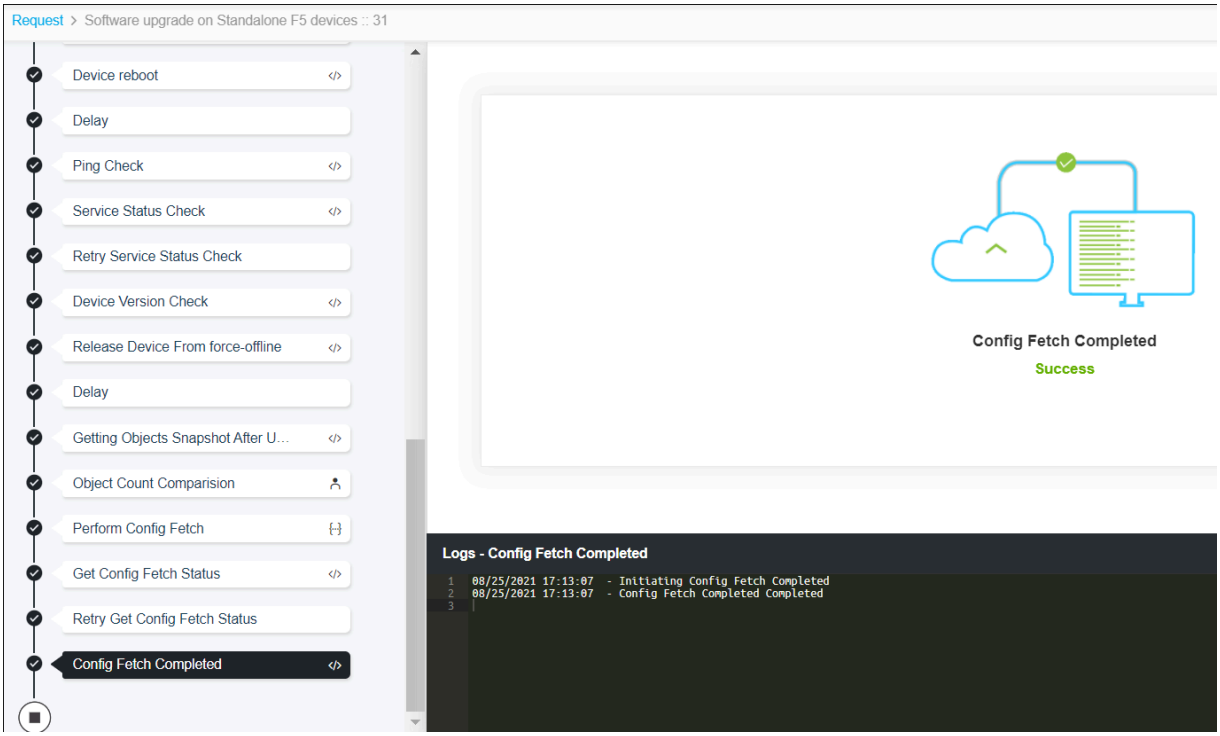
    The software upgrade starts. It takes a while to complete the software upgrade.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- A request is created for this workflow. To view the request, navigate to Menu > Request > All.
- If there is any failure in running this workflow, start a new workflow by providing the correct data.
- Alternatively, clone the request, update the correct input data in the form input, and submit the workflow.

12. After the manual check, click **Proceed**.
    The **Confirmation** dialog box is displayed.
13. Click **Ok** to submit the form.
14. After the **Object Count Comparison**, click **Proceed**.
    The **Confirmation** dialog box is displayed.
15. Click **Ok** to submit the form.
    The implementation process for software upgrade is completed.



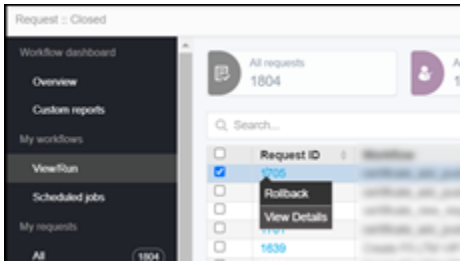
16. In case the request has to be reverted, execute the following steps:

    a. Go to the **Menu** > **Request** > **All**.

    b. Right-click the row on your desired Request ID.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

c. Select the **Rollback** option.



d. Select **Yes** in the **Rollback** confirmation dialog box.

A new request is triggered for performing rollback.

Software upgrade on Multiple HA F5 devices

The Workflow upgrades multiple Managed F5 HA pairs to next immediate version.

Prerequisites

- The ISO files for the software upgrade must be available in the path **/shared/images** of the F5-BIG IP in which the upgrade is to be performed.
- The User using which F5 BIG-IP is Managed in AppViewX should have Advance shell Access.
- For License reactivation, the F5 device should have Internet access.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Software upgrade on Multiple HA F5 devices** workflow.

   The **Run** and **Schedule** buttons are displayed.
4. Click 

.

   The **Request > Software upgrade on Multiple HA F5 devices :: FormBuilder** page is displayed.
5. Enter/select the **F5 HA Upgrade - Controller** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

F5 HA Upgrade - Controller Section - Field and Description Table

Field	Description
***Datacenter**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Standby Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the dropdown list.
Fetch Details	The fields in the **StandBy Device Details** and the **Active Device Details** sections are auto-populated when you click the **Fetch Details** button.
SFTP	When you use SFTP servers to store ISO files, you need to add the SFTP servers to the Inventory section under Others, with "sftp" selected as the model. The values are: • **Yes** - Select this option to store ISO files in SFTP servers. Then, select a SFTP sever from the **SFTP Server** drop-down menu and specify the path in the **ISO File Path** field. • **No** (default) - Select this option to store ISO files in AppViewX. For software upgrades, ensure that the ISO files are available in the following directory: **INSTALLATION_PATH/appviewx_dependencies/vw/dependencies/bigip_images**. In a multi-node instance, make sure to upload ISO files to all nodes.
***ISO File Name**	Click the retrieve button, and then select a desired ISO file name from the dropdown list.
***New Volume Name**	Enter the new volume name. The format of the volume name must be in the form of HDX.Y, CFX.Y, or MDX.Y.
Validate Device	Click this button to start the validation.
***Device Status**	Displays details about the status of the device.
***Overall Status**	Displays details about the overall status of the system.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Generate Qkview File**	Select **Yes** if required to generate Qkview file. The options are: • If you select **Yes** for the **Generate Qkview File** option, you can then choose whether to upload the Qkview file to the iHealth Portal. • If you select **Yes** for **Upload to iHealth Portal**, you will need to provide the iHealth case number associated with the upload. • If you choose **No** for **Upload to iHealth Portal**, the Qkview file will not be uploaded to the iHealth Portal.
***Reactivate License Automatically**	Select Yes if required to reactive license automatically. The options are: • **No** (default): Select this option to renew the license manually. • **Yes**: Select this option to enables automatic license renewal. For automatic license reactivation, the F5 device must have Internet access.
***Consolidated Device Table**	Click + to add the details to the Consolidated Device Table. After adding the details, it can be manage using edit, delete, or refresh buttons.
***Validate Request**	Click this button to retrieve the Request Trigger Status.
**: Mandatory fields*	

6. Click **Next**.
   The **Confirmation** dialog box is displayed.
7. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The software upgrade starts. It takes a while to complete the software upgrade.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

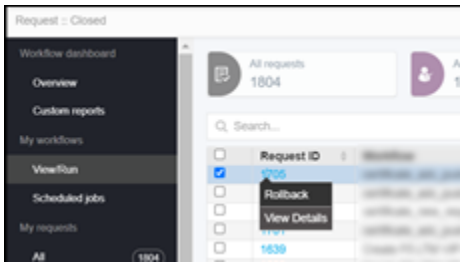
- A request is created for this workflow. To view the request, navigate to Menu > Request > All.
- If there is any failure in running this workflow, start a new workflow by providing the correct data.
- Alternatively, clone the request, update the correct input data in the form input, and submit the workflow.

8. After the manual check, click **Proceed**.
   The **Confirmation** dialog box is displayed.
9. Click **Ok** to submit the form.
10. After the **Object Count Comparison**, click **Proceed**.
    The **Confirmation** dialog box is displayed.
11. Click **Ok** to submit the form.
    The implementation process for software upgrade is completed.
12. In case the request has to be reverted, execute the following steps:

    a. Go to the **Menu** > **Request** > **All**.

    b. Right-click the row on your desired Request ID.

    c. Select the **Rollback** option.



    d. Select **Yes** in the **Rollback** confirmation dialog box.
    A new request is triggered for performing rollback.

Compliance - CVE Reporting and Golden Config Compliance

The workflows grouped under this sub-category are:

Fetch F5 BIG-IP CVEs

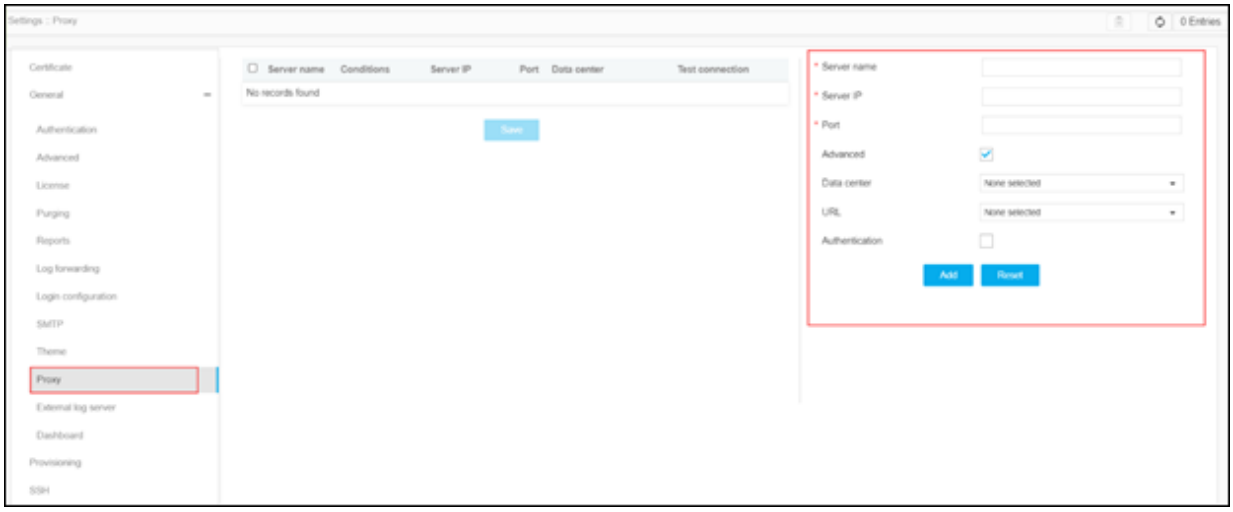
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Overview

This workflow fetches a list of CVE vulnerabilities from the Global site https://cve.mitre.org/ and stores them in the AppViewX Database. By using these vulnerabilities stored in the Database, the managed F5 load balancers in AppViewX can be validated to see whether they are vulnerable or not. This workflow only downloads the vulnerability data from the site and stores it in the database. The vulnerability validation and report generation can be done using the F5 BIG-IP CVE Reporting workflow. The vulnerabilities are identified for the Modules (LTM, GTM, ASM, AFM).

Prerequisites

- The VM (Virtual Machine) in which AppViewX is installed should have internet access or access to the proxy server. The proxy server should be configured in AppViewX. To configure, go to **Settings** > **General** > **Proxy**.



- You must be mapped to a role in which access to Collection, **Create/Modify** should be enabled to run this workflow.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Running F5 BIG-IP CVE Reporting

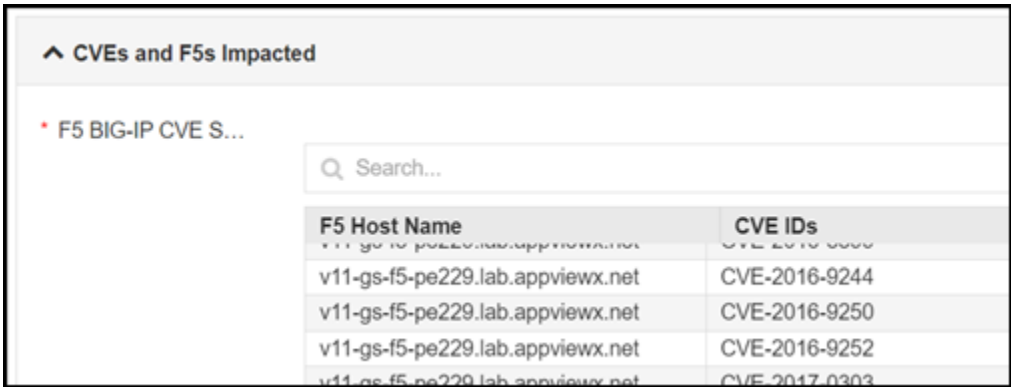
To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **F5 BIG-IP CVE Reporting** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > F5 BIG-IP CVE Reporting :: FormBuilder** page is displayed.
5. Enter/select the **Search Filter** details.

Search Filter Section - Field and Description Table

Field	Description
Search Filter	Select the search filter. The possible options are: • **F5 BIG-IP Version** – This option allows you to get the devices based on the product version. The product versions listed in the drop-down option are in the format **<major version>.x.x**. The vulnerabilities will be validated for the devices of selected product versions. • **F5 Big-IP Name** – This option allows you to get the devices based on the name.
***Available Device Versions**	Select the devices or product version from the drop-down list for which CVE validation to be triggered. The drop-down list is displayed if Search Filter is F5 BIG-IP Version.
***Available Devices**	Select the devices from the drop-down list for which CVE validation to be triggered. The drop-down list is displayed if Search Filter is F5 BIG-IP Name.

6. Click **Get CVE AFFECTED F5s** to get the devices affected by the CVE vulnerabilities.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The devices that are affected by the vulnerabilities are listed in the **CVEs and F5s Impacted** section.

7. If this report is to be sent via email, set the **Get Report By Email** option to **Yes**, and then enter the email IDs in the **Email_id** field.

   Note:
   Multiple email IDs can be added as comma-separated values.

8. Click **Submit For Detailed Report**.
9. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
   The form will be saved as **Open request** under **Request** > **My Request**.

   OR

   To submit the form, click **OK**.

   The report generation starts automatically. The CVE Report and CVE Severity Report are generated.
10. To view the CVE report, click the **CVE Report** tab from the left panel.

    The report can be downloaded by clicking  .
11. To view the CVE severity report, click the **CVE Severity Report** tab from the left panel.
    The view of this report can be changed from pie chart or donut view by selecting the option from the **View by** drop-down option.

    The CVE Severity Report displays the following details:
    - **Device Scanned** – Total number of scanned devices.
    - **Vulnerable Devices** – Total number of vulnerable devices in the devices scanned.
    - **Unique CVEs** - Unique vulnerability.
    - **Total CVEs** – Cumulative count of severities (Critical, Medium, High, etc.) reported for each module (LTM, GTM, AFM, and ASM) in the device. The severity reported for a module is identified only if the module is enabled in AppViewX while adding the device in the Inventory.

    The remediation versions are mentioned for each module (LTM/GTM/AFM/ASM) in the vulnerable devices, in the CVE Report tab. If required, trigger the F5 BIG-IP Software Upgrade on Standalone or F5 BIG-IP Software Upgrade on HA Devices to fix the vulnerabilities in the device.

F5 BIG-IP CVE Reporting

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Overview

This workflow validates the vulnerabilities against the device and generates reports. Before you run this workflow, make sure to run the Fetch F5 BIG-IP CVEs workflow to get the latest vulnerabilities list from the Global site.

Prerequisites

You must be mapped to a role in which access to Collection, **Create/Modify** should be enabled to run this workflow.



Running Fetch F5 BIG-IP CVEs

To run this workflow:

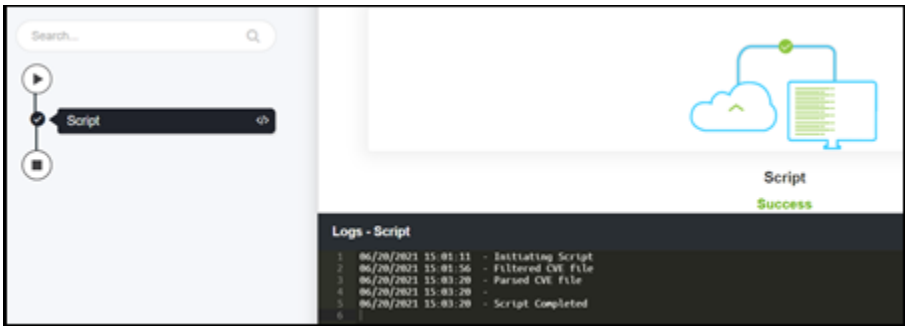
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Fetch F5 BIG-IP CVEs** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The script starts running.



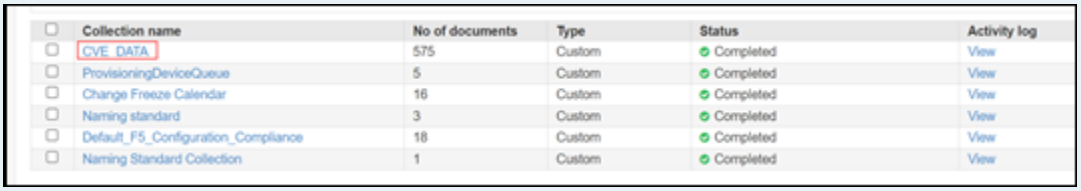
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Once the workflow is completed, the CVE vulnerabilities are fetched from the Global site and stored in the Database.



Note:

To view the list of stored vulnerabilities, go to **Menu** > **Collection**:



Collection name	No of documents	Type	Status	Activity log
CVE_DATA	575	Custom	Completed	View
ProvisioningDeviceQueue	5	Custom	Completed	View
Change Freeze Calendar	16	Custom	Completed	View
Naming standard	3	Custom	Completed	View
Default_F5_Configuration_Compliance	18	Custom	Completed	View
Naming Standard Collection	1	Custom	Completed	View

F5 BIG-IP Golden Config Compliance

Overview

This workflow validates a set of default parameters and custom parameters, if any, in the F5 load balancers managed in the AppViewX. Post validation, it generates a report of compliant and non-compliant parameters.

- **Complaint parameter** - The value of the parameter in the device matches with the value defined in the Compliance Policy in AppViewX for the parameter.

- **Non-complaint parameter** - The value of the parameter in the device does not match with the value defined in the Compliance Policy in AppViewX for the parameter.

This workflow provides an option to make the non-compliant parameters to compliant parameters in the F5 load balancer by writing the value as defined in the AppViewX. AppViewX ships the Standard Compliance policy Default_F5_Configuration_Compliance with Default parameters and sample values. The Standard Compliance policy file is non-editable. You can create Custom Compliance policies with Custom or selected Default parameters of your choice.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Default Parameters – The parameters supported by default in AppViewX are:
  - Command to read the values for the parameter.
  - Command to write the value for the parameter.
  - Command to revert the value for the parameter.

In F5, the commands are defined in AppViewX.

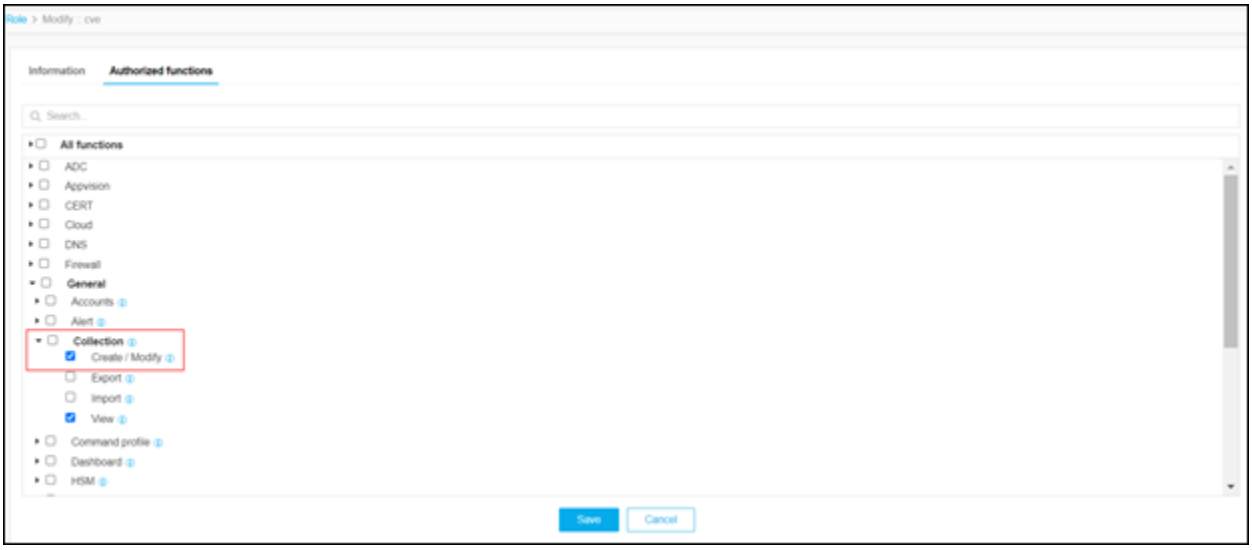
- Custom Parameters – The Parameters newly added by the user. The user must provide the command to read, the command to write, the command to revert, as an input in the Workflow.

Compliant or Non-compliant parameters are decided based on the logic below:

- A value is mapped to a parameter in the Policy.
- The Parameter complete config is taken from the device, as a response.
- If the value defined in the Policy is available in the response, then it is compliant, else non-compliant. For a few default parameters (syslog_param, cm-device-group-auto-sync, net-self-allow-service), the logic changes. For more details, click here.

Prerequisites

You must be mapped to a role in which access to Collection, **Create/Modify** should be enabled to run this workflow.



Running F5 BIG-IP Golden Config Compliance

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **F5 BIG-IP Golden Config Compliance** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > F5 BIG-IP Golden Config Compliance :: FormBuilder** page is displayed.
5. Select the **Select Compliance Policy** from the dropdown list.
   The compliance policy options are:
   - **Create New Compliance** – Select this option if you want to create a custom compliance policy file. When you select this option, the Compliance Policy Name field is shown in the form input to enter the name of the compliance policy file name.



   - **Default_F5_Configuration_Compliance** - Select this option to use the Standard Compliance policy file, which was shipped by AppViewX.
   - **<Custom Compliance Policy File Name>** - Other custom compliance policy files If any custom compliance policy file was created already.
6. Select the **Select Type** options are:
   - **Device Name** - When you select **Device Name** option, you get the option to select devices in the **Device Name** field for which the compliance policy validation is triggered.

> **Note:**
> Only the devices that are in managed and active status are displayed for the selection.

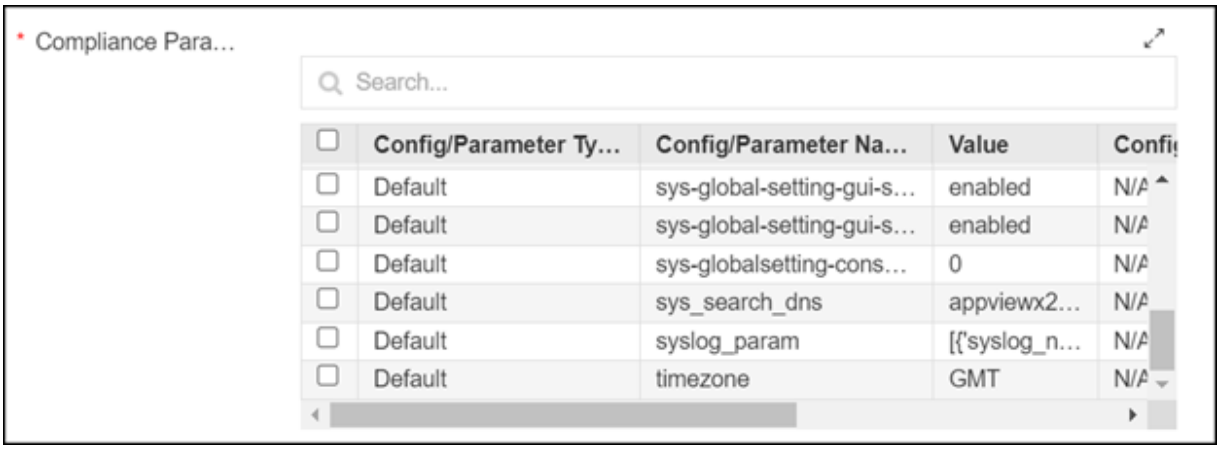
   - **Device Group** - When you select the Device Group option, you get the option to select the device group in the Device Group field for which the compliance policy validation is triggered.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:
Only the devices that are in managed and active status within the group are displayed.



7. After the **Device Name** selection or **Device Group** selection, the Compliance Parameter Table is updated with the parameters based on the selected compliance policy.

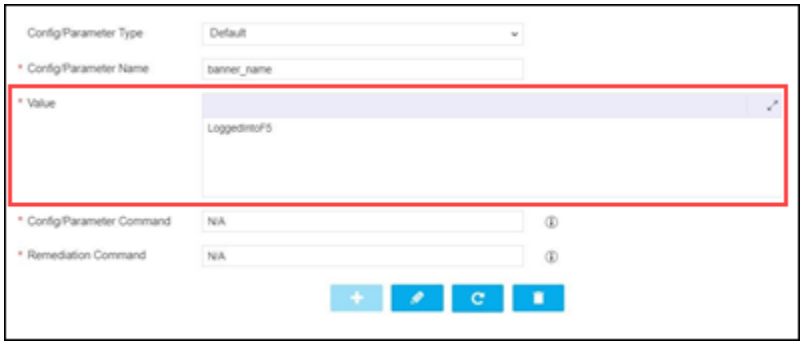


- When the **Create New Compliance** or **Default_F5_Configuration_Compliance** option is selected in the **Select Compliance Policy** field, the **Compliance Parameter Table** displays the Default parameters.
  - The **Default Parameters Value** in the **Compliance Parameter** table must be modified accordingly. To modify,
    - Select a Param:



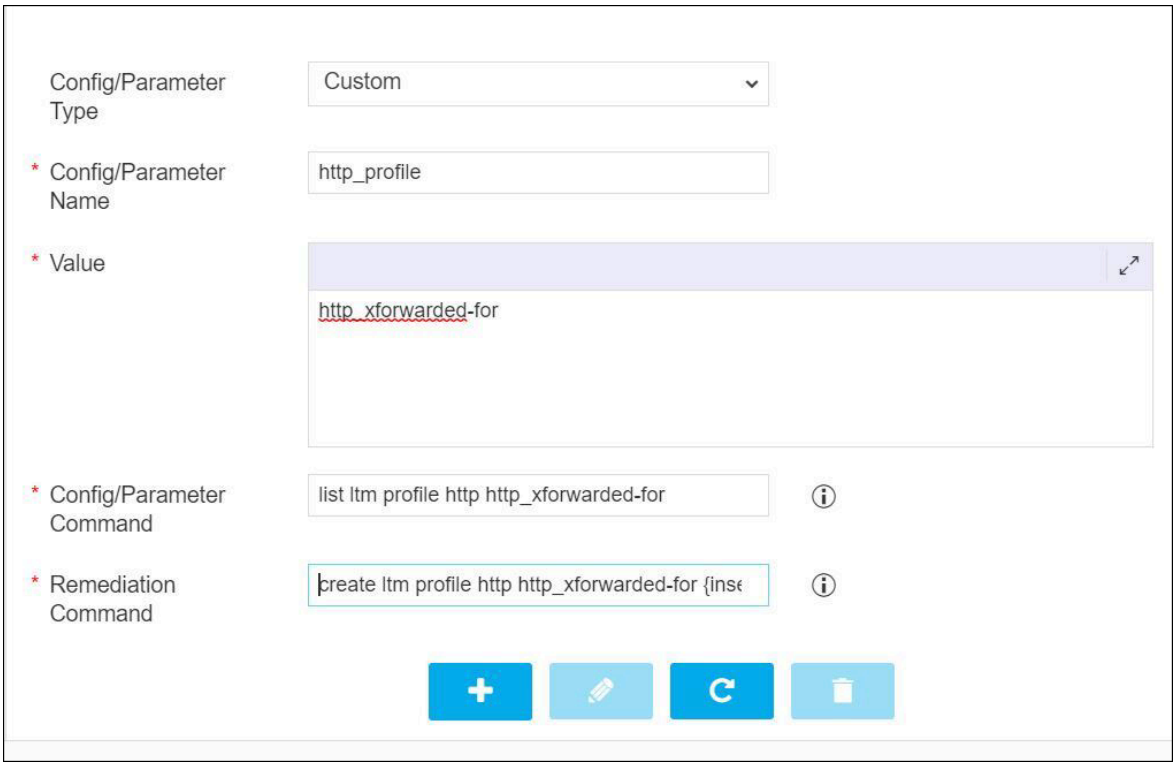
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Modify **Value** for a Param:



- Update the Value. To update the modified value into the table, click  **(Edit)**.

- New custom parameters can also be added. The custom parameter can be added by providing the details for it as follows:



- If the Default compliance policy is selected and the values are modified, save this as a new compliance policy by clicking the **Save as** option as **Yes**.
- If you select the **Save as** option as **No**, the modified values or added any custom parameter will not be saved as a new compliance policy.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- On selecting a specific compliance policy in the beginning of this form, the **Compliance Parameter Table** displays default and custom parameters as customized.
- If any values are modified or any new custom parameter is added to the table, then you can save them in the compliance policy by selecting the option Update Policy – **Yes**.

Note:

- You can only update the values for the default parameters and should not modify the Config/Parameter Name field for Default parameters. Modifying the Config/Parameter Name field might lead to incorrect validation.
- Any new Config/Param added as type Default will also be considered as Custom parameter.

8. In the Compliance Parameter Table, enter/select the details as required.

Compliance Parameter Table

Field	Description
***Config/ Parameter Type**	You can select either **Custom** or **Default** Config/Parameter type. For a Custom Compliance policy, the Config/Parameter type is automatically treated as Custom, even if default policy is configured under Custom parameter type. When you choose the **Custom** parameter type, the custom compliance policy is added to the collection for future reuse.
***Config/ Parameter Name**	Enter the config/parameter name for compliance.
***Value**	Enter the expected value in the device for the parameter.
***Config/ Parameter Command**	Enter the command to validate the parameter (Applicable only for Custom Parameter). For the Default parameter, the command is defined by AppViewX.
***Remediation Command**	Enter the command to remediate the parameter (Applicable only for the Custom parameter). For the Default parameter, the command is defined by AppViewX.
** : Mandatory fields*	

9. To add a new parameter to the Compliance Parameter Table, click Add () button. After adding the parameter, you can manage (update/delete) it.
10. Click the **Validate** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The validation starts and the results are displayed in the **Validation Status** box.

11. Click the **Submit** button.
    The Confirmation popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **All**.
> - If you want to cancel this form, click the **Cancel** button.

12. Click **Ok** to submit the form.
    The Compliance Check runs automatically and generates **Compliance Grid Report**.
13. At the **Compliance Statistics** stage, you can view the compliant and non-compliant parameters displayed in either the default chart view or a pie chart.
14. From the Compliance Grid Report, you get to know if the device parameters are Compliant or Non-Compliant.

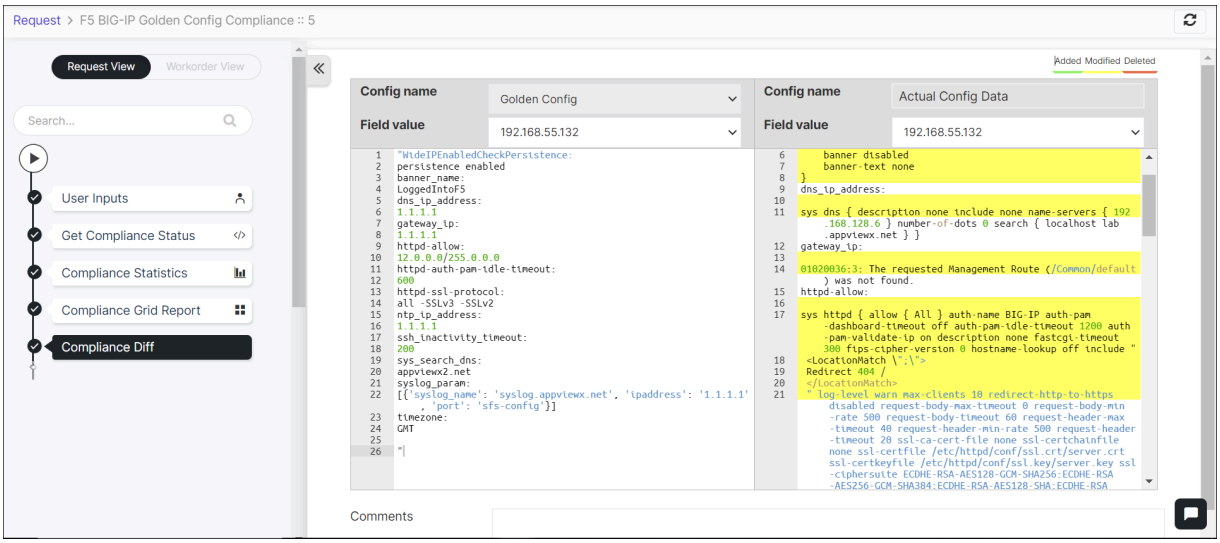
    You can download the **Compliance Grid Report** in the **.csv** or **.xlsx** format by clicking  **(Download)**.

    The summary of the Compliance Grid Report are:
    - The first column displays the device name.
    - The other columns display Config/Parameter names.
    - The report has two body rows. They represent the following:
    - First Row - the status of the Parameter in the device.
    - Second Row - the actual value of the parameter in the device.
15. Click the **Submit** button, and then click **Ok** in the confirmation pop-up.
    The **Compliance Diff check** runs.
16. The **Compliance Diff check** displays the details of non-compliant parameters for each device.
    The details are:
    - Parameter name and the value in the selected policy – shown in the left side of the diff Palate-Golden Config.
    - Parameter name and the value in the selected device - shown in the right side of the diff Palate-actual config.

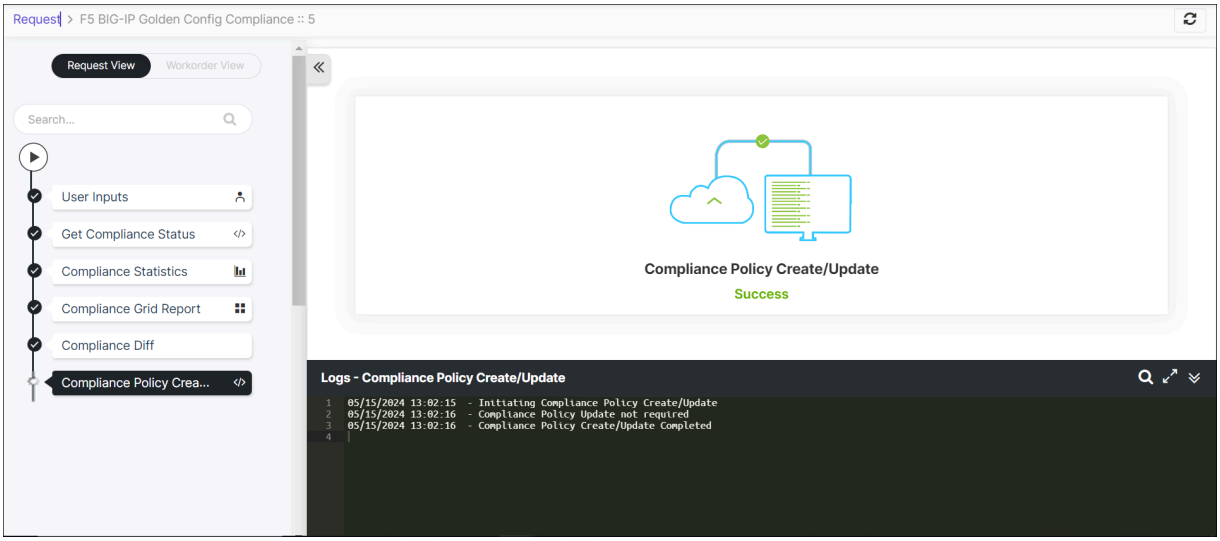
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

To view the compliance difference for the other devices, select a device from the field value drop-down list in the left and/or right side of the diff Palate.



17. Click the **Submit** button, and then click **Ok** in the confirmation pop-up.
18. The **Compliance Policy Create/Update** runs automatically:

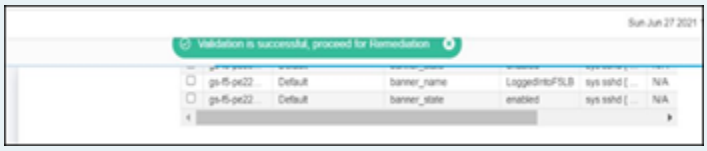


Note:

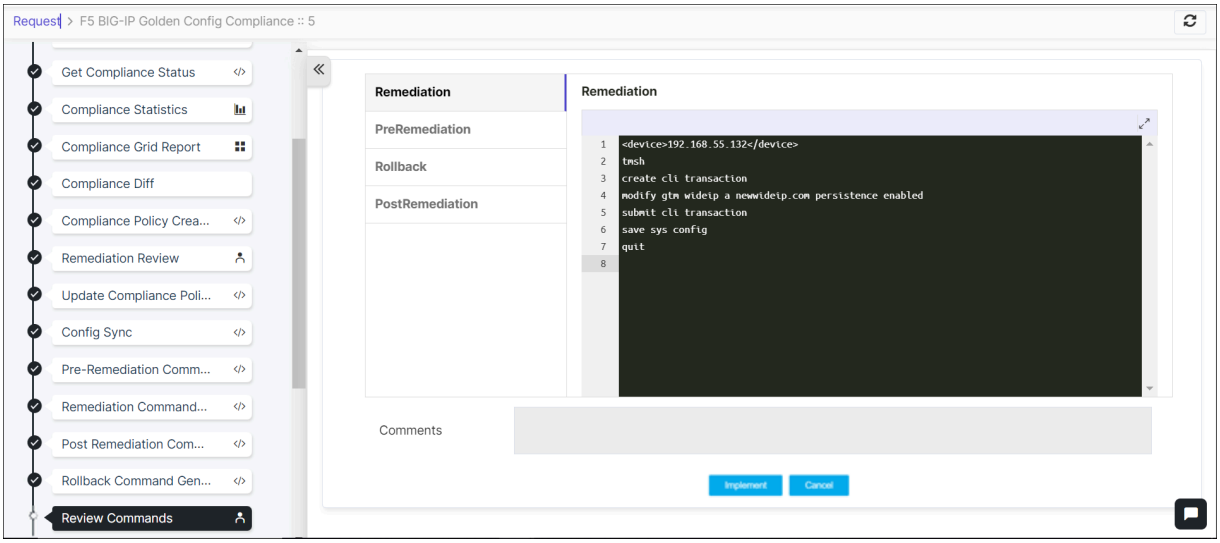
At this phase, the policy is created/updated if you have selected the option Create New Compliance, SaveAs – Yes, Update Policy -Yes in the form input.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

19. Remediation Review: The Page displays a list of the Non-Compliant Parameters. By default, two options are displayed: No, Yes.
    - **No** – By default this option is selected. When this option is selected and you **Submit**, the remediation will not be applied to the non-compliant parameters, the workflow ends.
    - **Yes** – When you select **Yes**, the remediation table is displayed for the non-compliant parameters.

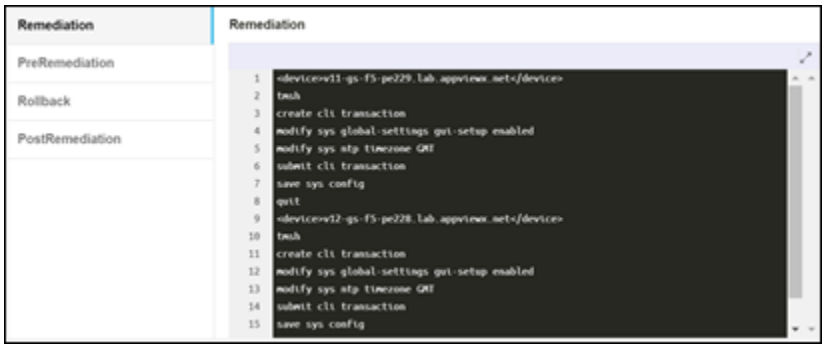
> **Note:**
> - In case any custom parameter is part of this remediation table, then update the rollback command in the **Rollback Command** field.
> - For default parameters, the remediation or rollback command is not required to be provided.
> - If the rollback command is not updated for the custom parameter, you do not get the rollback option in case to revert the device parameter to the older state.
> - Once the remediation command and rollback command are updated to the device, click the Update button.
> - Validate the remediation and rollback details by clicking the Validate button until the message Validation is Successful is seen.
>
> 

20. Click the **Submit** button, and then click **Ok** in the confirmation pop-up.
    The update of remediation and rollback commands run automatically and generates Review Commands to review and implement the changes.

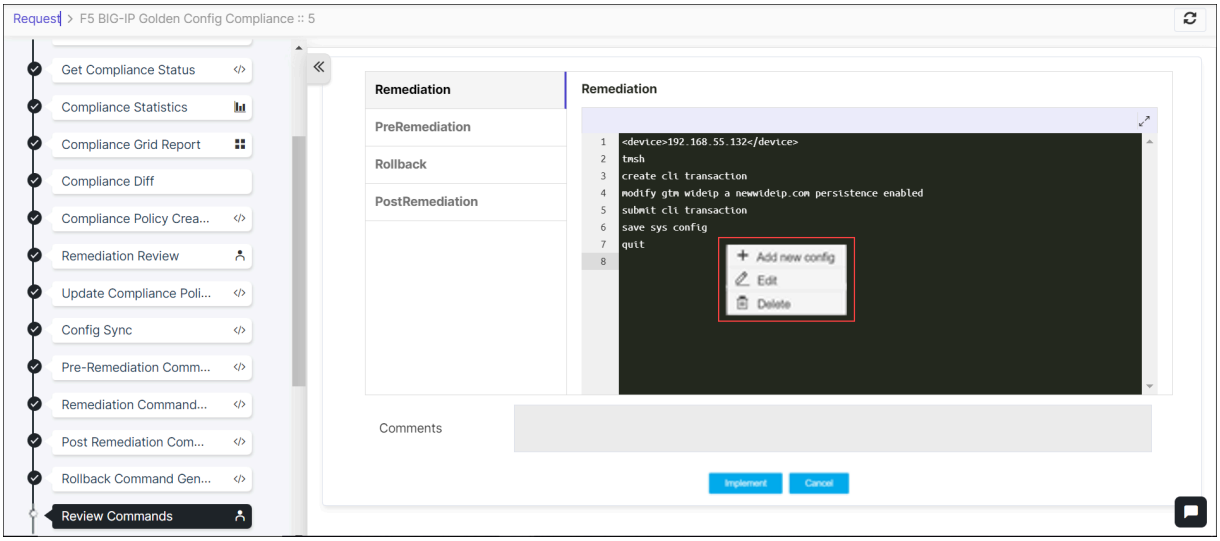


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

21. If required, review the commands that are about to be executed in the F5 load balances in the following tabs:
    - **Remediation**
    - **PreRemediation** – List the values of the parameter, prior Remediation.
    - **Rollback** – Revert the parameter's value to the previous state of Remediation.
    - **PostRemediation** - List the values of the parameter, post Remediation.



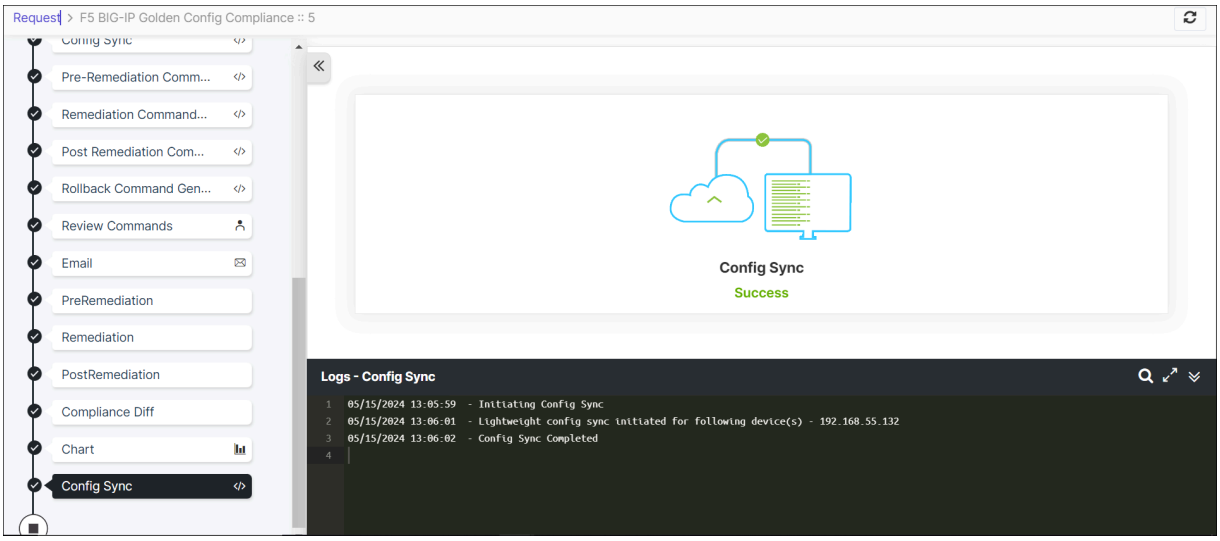
22. You can edit the commands at this stage, by clicking the right-mouse button in the desired tab.



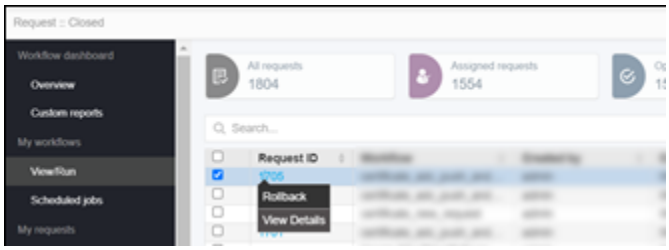
23. Once reviewed the commands, click the Implement button, and then click **Ok** in the confirmation pop-up.

    The implementation process completes. If email is configured, the email will be triggerred to the email IDs configured.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



24. In case the request has to be reverted, perform the following steps
    - Go to the **Menu** > **Request** > **All**.
    - Right-click the row on your desired Request ID.
    - Select the **Rollback** option.



    - Select Yes on the Rollback confirmation dialog.
    - A new request is triggered for performing rollback.
25. Mapping for the Default Parameter Name to Actual Parameter in Device:

Mapping for the Default Parameter Name to Actual Parameter in Device

Default Parameter Name	Actual Parameter

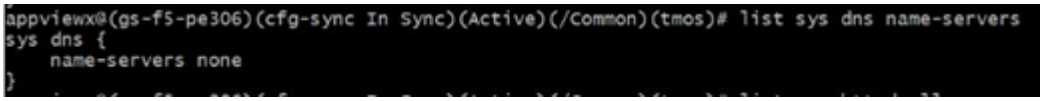
banner_name:	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys sshd banner-text
	sys sshd {
	banner-text none
	}

banner_state	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys sshd banner-text
	sys sshd {
	banner-text none
	}

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

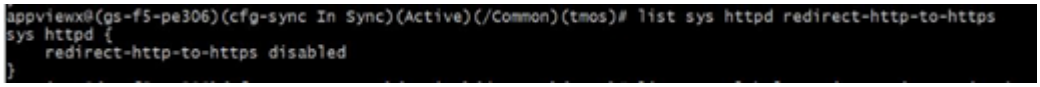
Default Parameter Name	Actual Parameter

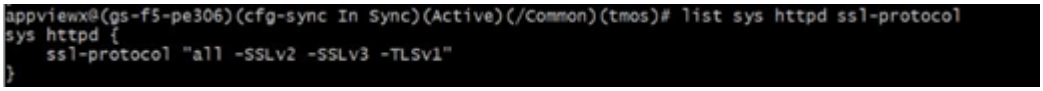
cm-device-group-auto-sync	Compliant or Non-Compliant is decided based on the below logic:
	
	In each of the cm device-group config, first it is checked whether “type sync-failover” is there. If it exists, then “auto-sync” value in the device is cross checked with the value defined in the Policy, if all match compliant, else non-compliant. Remediation happens only for Non-Complaints.
	 
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list cm device-group
	cm device-group HA_group {
	auto-sync enabled
	devices {
	gs-f5-pe305.lab.appviewx.net { }
	gs-f5-pe306.lab.appviewx.net { }
	}
	type sync-failover
	}
	cm device-group datasync-device-gs-f5-pe305.lab.appviewx.net-dg {
	auto-sync enabled
	devices {
	gs-f5-pe305.lab.appviewx.net { }
	gs-f5-pe306.lab.appviewx.net { }
	}
	full-load-on-sync true
	network-failover disabled
	}

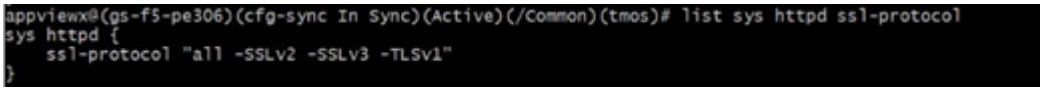
dns_ip_address	
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys dns name-servers
	sys dns {
	name-servers none
	}

gateway_ip	
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys httpd auth-pam-idle-timeout
	sys httpd {
	auth-pam-idle-timeout 1200
	}

httpd-allow	
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys httpd auth-pam-idle-timeout
	sys httpd {
	auth-pam-idle-timeout 1200
	}

httpd-auth-pam-idle-timeout	
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys httpd redirect-http-to-https
	sys httpd {
	redirect-http-to-https disabled
	}

httpd-redirect-http-to-https	
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys httpd ssl-protocol
	sys httpd {
	ssl-protocol "all -SSLv2 -SSLv3 -TLSv1"
	}

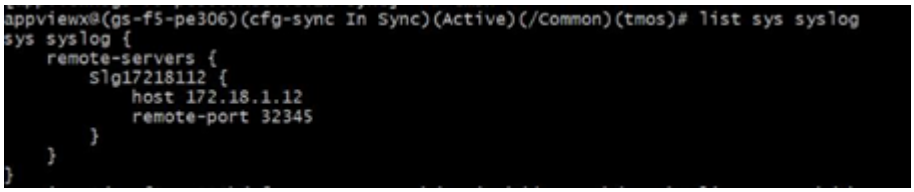
httpd-ssl-protocol	
	appviewx@(gs-f5-pe306)(cfg-sync In Sync)(Active)(/Common)(tmos)# list sys httpd ssl-protocol
	sys httpd {
	ssl-protocol "all -SSLv2 -SSLv3 -TLSv1"
	}

net-self-allow-service	The Parameter is compliant or Non-Compliant is decided based on the key “allow-service”. In each net self config, it is checked if the value of allow-service in the device matches with the one defined in AppViewX. If all config matches, it is Compliant, else

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Default Parameter Name	Actual Parameter
	it is Non-Compliant. Remediation happens only for the config that are Non-Compliant. Only Default Values are supported (none, default, all) and no custom values are supported. Example for custom value is tcp:8844. 
ntp_ip_address	
ssh_inactivity_timeout	
sys-global-setting-gui-security-banner	
sys-global-setting-gui-setup	
sys-globalsetting-console-inactivity-timeout	
sys_search_dns	
syslog_param	
syslog_param	Compliant or Non-Compliant is based on the points below:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Default Parameter Name	Actual Parameter
	• If the IP and Port defined in Golden config match with the device, then compliant. Then, no action is required. • If Syslog Name, IP, or port defined in Golden config does not match, then it is non-compliant. Then, update the IP or port. • · If the Syslog Name and IP or port defined in the golden config do not match, then create a new Syslog config. 
timezone	

F5 BIG-IP ZTP

HA

Launch F5 BIG-IP HA Pair in Azure

This workflow launches a failover High Availability (HA) Multi NIC F5 BIG-IP Virtual Machines on Azure. The existing VNET and Subnet are utilised to create two F5 devices with Standby and Active state on Azure. AppViewX will check if the deployment successfully completes. Once done, the device(s) details are fetched. The failover F5 devices are added into AppViewX and Email will be sent to the user. Option to do basic provisioning to the newly added F5 device. Active device will be chosen for the same.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP ZTP** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Launch F5 BIG-IP HA Pair in Azure** workflow. The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click 

.

   The **Request > Launch F5 BIG-IP HA Pair in Azure :: FormBuilder** page is displayed.

5. Enter/select **Azure Account Details**.

> **Note:**
> The entered tenant/application ID must have contributor/owner access to the Azure application provided.

Azure Account Details Section - Field and Description Table

Field	Description
Azure Subscription ID	Enter the subscription ID.
Tenant(Directory) ID	Enter Tenant ID.
Azure Client ID	Enter the client ID.
Azure Client Secret	Enter the client secret key.

6. Enter/select **Proxy Details**.

Proxy Details Section - Field and Description Table

Field	Description
Connect to Azure via Proxy	Allows to make connections via Proxy,. The options are: • **No** (default) - select this option to connect with Azure without proxy (Internet connection must be required). • **Yes** - select this option to connect with Azure via proxy, then click the **Get Proxies in Appviewx** button to fetch the proxy servers configured in AppViewX to be displayed in the **Proxies** dropdown list.
***Proxy**	Select a proxy servers configured in AppViewX from the dropdown list.

7. Enter/select **Azure Deployment information**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Azure Deployment information Section - Field and Description Table

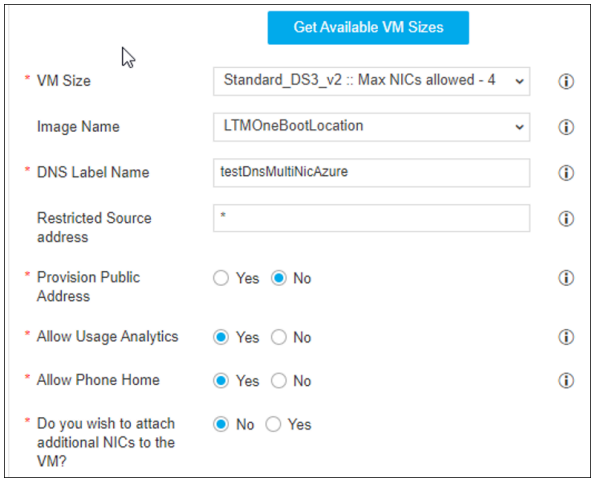
Field	Description
Get F5 on Azure Deployment collections	Click the **Get F5 on Azure Deployment collections** button to get the deployment templates from the AppViewX Collection.
Select Deployment Collection	Select a deployment collection from the dropdown list.
Select the Deployment Template	Select a deployment template type from the dropdown list.
Stack type	It is non editable field and stack type will be auto-filled from the selected deployment template.
F5 BIG-IP Version	It is non editable field and F5 BIG IP version will be auto-filled from the selected deployment template.
F5 BIG-IP Virtual Edition	It is non editable field and F5 BIG IP virtual edition will be auto-filled from the selected deployment template.
nic	Add n number of NICs using this template.
Content Version	It is non-editable field and content version will be auto-filled from the selected deployment template.

8. Enter/select **F5 Virtual Machine Details**.

F5 Virtual Machine Details Section - Field and Description Table

Field	Description
F5 Instance Name	Name of the F5 device name which was provided for a device while onboarding it in AppViewX Inventory.
Get available resource groups	Click this button to fetch the Azure resource groups of virtual network.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Virtual Network Resource Group	Select a virtual network resource group from the dropdown list.
Virtual Network	Select a virtual network from the resource group from the dropdown list. **Note:** Make sure that selected the VNET/ Management Subnet has connection established from and to AppViewX, only then the device can be communicated from AppViewX.
mgmt subnet name	Select a unique management subnet name from the dropdown list. Your subnet must have access to the internet.
External subnet name	Select an external subnet name.
Internal subnet name	Select an Internal subnet name.
mgmt IP-Address	Management IP address to use for the BIG-IP management IP address. By default, this field is filled with DYNAMIC to use a dynamically assigned IP address. Else provide an IP address.
Get Available VM Sizes	Click this button to fetch the available VM sizes. 

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
VM Size	Select the instance size of the virtual machine from the dropdown list.
Boot Location Type	The options are displayed in this field as it is available in the Azure marketplace for F5. Select a boot location type based on the requirement.
DNS Label Name	Enter a unique DNS Name for the Public IP address used to access the Virtual Machine.
Restricted Source address	This field restricts management access to a specific network or address. Enter an IP address or address range in CIDR notation or asterisk for all sources.
Provision Public Address	Provision to create the Public IP addresses for BIG-IP Management traffic. These addresses will be configured directly on the Azure network interfaces.
Allow Usage Analytics	This deployment can send anonymous template statistics to F5 to help us determine how to improve our solutions. If you select **No**, statistics are not sent.
Allow Phone Home	This deployment can provide F5 with high-level device use information to optimize development resources. If you select **No** the information is not sent.
Do you wish to attach additional NICs to the VM?	Click Yes to attach additional NICs to the VM otherwise select No.

9. Enter/select **F5 Login Info**.

F5 Login Info Section - Field and Description Table

Field	Description
Login User Name	Enter the user name for the virtual machine.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	Note: The user name must not be "admin".
Authentication Type	Type of authentication to use on the Virtual Machine, password based authentication or key based authentication.
Login Password	Enter the password with the following combination: Minimum eight characters, at least one uppercase letter, one lowercase letter, one number and one special character.

10. Enter/select **F5 License details**.

F5 License Details Section - Field and Description Table

Field	Description
Do you wish to activate license for the F5 device?	The options for activating the license: • **No** (default) - if you select this option, you cannot access F5 modules. • **Yes** - if you select this option, the Base registration key is enabled. Enter the registration key to activate the license.

11. The following table provides the field description for the **EBS Volume Details** section.
12. Click **Submit**.
    The **Confirmation** dialog box is displayed.
13. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The validation starts automatically and reaches the **Deployment Info** stage. In this page, the host details are shown. If required, you can download the details in the **.csv** or **.xls** format by clicking the  **(Download)** button.



14. Click the **Submit** button to submit the form.
    The validation starts automatically and reaches the **VM Status** stage.
15. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **High Availability F5 Devices Info** stage.
16. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **Basic Provisioning** stage.
17. Enter/select the **Basic VIP Creation** details.

Basic VIP Creation Section - Field and Description Table

Field	Description
VIP Name	Enter the name of the VIP.
Destination Address	Enter the destination IP address.
Port	Enter a port.
Pool Name	Enter the pool name.
Load Balancing Method	The load balancing method is used to distribute traffic across pool in this VIP. The default is round-robin. The methods are: • **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.

18. Enter/select **Pool Member Details**.

Pool Member Details Section - Field and Description Table

Field	Description
Member IP	Enter the member IP.
Port	Enter a port.
Pool Members	Click the add button to add the pool members to the table. After adding the pool member details to the list, you can manage them.

19. Enter/select **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
Monitor Type	Select the health monitors that are available to add for the pool: • HTTP • HTTPS • TCP

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• TCPHALFOPEN • Gatewaylcmp
Monitor Name	Enter the name for the monitor.
Send String	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. To retrieve a specific page from a website, specify a fully qualified path name.
Receive String	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

20. Enter/select **Other Settings**.

Other Settings Section - Field and Description Table

Field	Description
Banner Name	Enter the name for banner.
Banner State	Select a state for the banner. The possible states are: • enabled • disabled
DNS Address	Enter the DNS address.
NTP Address	Enter the NTP address.

21. Enter/select the **Syslog Setting** details.

Syslog Setting Section - Field and Description Table

Field	Description
Syslog Name	Enter the name for syslog.
Syslog Ipddress	Enter the syslog IP address.
Syslog Port	Enter the syslog port.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
SYSLOG	Click the add button to add the syslog settings to the list. After adding the syslog settings to the list, you can manage them.

22. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **Review** stage.
23. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.
24. After the review, click the **Submit** button.
    The Confirmation popup opens. It takes a while to complete the request.

> **Note:**
> To stop running the workflow creation, click **Cancel** .

> **Note:**
> The credentials to access BIG-IP Configuration Utility are:
> **Username:** admin
> **Password:** Contact the AppViewX support team.

Standalone

Launch F5 Standalone Single NIC in Azure

This workflow launches a Standalone Single-NIC F5 BIG-IP in Azure Cloud. A F5 device will be deployed in Azure. If new-stack template is selected, then vnets and subnets will be created. If existing-stack template is selected, the existing vnet and subnet will be utilized. AppViewX checks if the deployment successfully completed. Once completed, the device details are fetched. The F5 device is added into AppViewX and Email will be sent to the user. There is an option to do basic provisioning to the newly added F5 device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP ZTP** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Launch F5 Standalone Single NIC in Azure** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click Run.

   The **Request > Launch F5 Standalone Single NIC in Azure :: FormBuilder** page is displayed.

5. Enter/select the **Azure Account Details**.

> **Note:**
> The entered tenant/application ID must have contributor/owner access to the Azure application provided.

Azure Account Details Section - Field and Description Table

Field	Description
Azure Subscription ID	Enter the subscription ID.
Tenant(Directory) ID	Enter Tenant ID.
Azure Client ID	Enter the client ID.
Azure Client Secret	Enter the client secret key.

6. Enter/select the **Proxy Details**.

Proxy Details Section - Field and Description Table

Field	Description
***Connect to Azure via Proxy**	Allows to connect with Azure via proxy or not. The options are: • **No** (default) - select this option to connect with Azure without proxy (Internet connection must be required). • **Yes** - select this option to connect with Azure via proxy, then click the **Get Proxies in Appviewx** button to fetch the proxy servers configured in AppViewX to be displayed in the **Proxies** dropdown list.
***Proxy**	Select a proxy servers configured in AppViewX from the dropdown list.

7. Enter/select the **Azure Deployment information**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Azure Deployment information Section - Field and Description Table

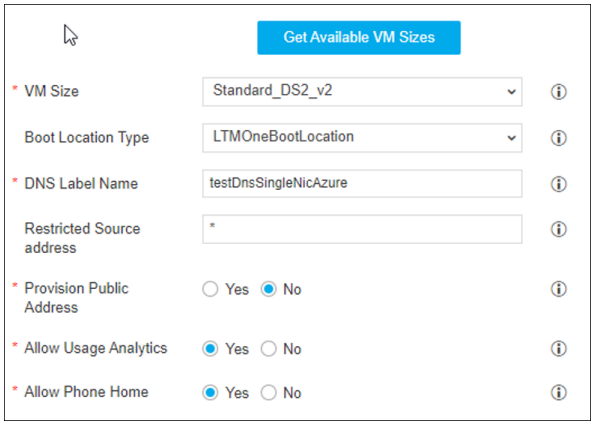
Field	Description
Get F5 on Azure Deployment collections	Click the **Get F5 on Azure Deployment collections** button to get the deployment templates from the AppViewX Collection.
Select Deployment Collection	Select a deployment collection from the dropdown list.
Select the Deployment Template	Select a deployment template type from the dropdown list.
Stack type	It is non editable field and stack type will be auto-filled from the selected deployment template.
F5 BIG-IP Version	It is non editable field and F5 BIG IP version will be auto-filled from the selected deployment template.
F5 BIG-IP Virtual Edition	It is non editable field and F5 BIG IP virtual edition will be auto-filled from the selected deployment template.
Content Version	It is non editable field and content version will be auto-filled from the selected deployment template.

8. Enter/select the **F5 Virtual Machine Details**.

F5 Virtual Machine Details - Field and Description Table

Field	Description
F5 Instance Name	Name of the F5 device/Azure instance which was provided for a device while onboarding it in AppViewX Inventory.
Get available resource groups	Click this button to fetch the Azure resource groups.
Virtual Network Resource Group	Select a virtual network resource group from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Virtual Network	Select a virtual network from the resource group from the dropdown list.
mgmt subnet name	Select a unique management subnet name from the dropdown list. Your subnet must have access to the internet. **Note:** Make sure that selected the VNET/ Management Subnet has connection established from and to AppViewX, only then the device can be communicated from AppViewX.
mgmt IP-Address	Management IP address to use for the BIG-IP management IP address. By default, this field is filled with DYNAMIC to use a dynamically assigned IP address. Else provide an IP address from management subnet.
Get Available VM Sizes	Click this button to fetch the available VM sizes. 
VM Size	Select the instance size of the virtual machine from the dropdown list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Boot Location Type	The options are displayed in this field as it is available in the Azure marketplace for F5. Select a boot location type based on the requirement.
DNS Label Name	Enter a unique DNS Name for the Public IP address used to access the Virtual Machine.
Restricted Source address	This field restricts management access to a specific network or address. Enter an IP address or address range in CIDR notation or asterisk for all sources.
Provision Public Address	Provision to create the Public IP addresses for BIG-IP Management traffic. These addresses will be configured directly on the Azure network interfaces.
Allow Usage Analytics	This deployment can send anonymous template statistics to F5 to help us determine how to improve our solutions. If you select **No**, statistics are not sent.
Allow Phone Home	This deployment can provide F5 with high-level device use information to optimize development resources. If you select **No** the information is not sent.

9. Enter/select **F5 Login Info**.

F5 Login Info Section - Field and Description Table

Field	Description
Login User Name	Enter the user name for the virtual machine. **Note:** The user name must not be "admin".
Authentication Type	Type of authentication to use on the Virtual Machine. The available option is password based authentication.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Login Password	Enter the password with the following combination: Minimum eight characters, at least one uppercase letter, one lowercase letter, one number and one special character.

10. Enter/select the **F5 License Activation**details.

F5 License Activation Section - Field and Description Table

Field	Description
Do you wish to activate license for the F5 device?	The options for activating the license: • **No** - if you select this option, you cannot access F5 modules. • **Yes** - if you select this option, the Base registration key is enabled. Enter the registration key to activate the license.

11. Click **Submit**.
    The **Confirmation** dialog box is displayed.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Deployment Info** stage. In this page, the host details are shown. If required, you can download the details in the **.csv** or **.xls** format by clicking the  **(Download)** button.
13. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **VM Status** stage.
14. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **Basic Provisioning** stage.
15. Enter/select the **Basic VIP Creation** details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Basic VIP Creation Section - Field and Description Table

Field	Description
VIP Name	Enter the name of the VIP.
Destination Address	Enter the destination IP address.
Port	Enter a port.
Pool Name	Enter the pool name.
Load Balancing Method	The load balancing method is used to distribute traffic across pool in this VIP. The default is round-robin. The methods are: • **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.

16. Enter/select the **Pool Member Details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Pool Member Details Section - Field and Description Table

Field	Description
Member IP	Enter the member IP.
Port	Enter a port.
Pool Members	Click the add button to add the pool members to the table. After adding the pool member details to the list, you can manage them.

17. Enter/select the **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
Monitor Type	Select the health monitors that are available to add for the pool: • HTTP • HTTPS • TCP • TCPHALFOPEN • Gatewaylcmp
Monitor Name	Enter the name for the monitor.
Send String	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. To retrieve a specific page from a website, specify a fully qualified path name.
Receive String	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

18. Enter/select the **Other Settings**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Other Settings Section - Field and Description Table

Field	Description
Banner Name	Enter the name for banner.
Banner State	Select a state for the banner. The possible states are: • enabled • disabled
DNS Address	Enter the DNS address.
NTP Address	Enter the NTP address.

19. Enter/select the **Syslog Setting** details.

Syslog Setting Section - Field and Description Table

Field	Description
Syslog Name	Enter the name for syslog.
Syslog Ipddress	Enter the syslog IP address.
Syslog Port	Enter the syslog port.
SYSLOG	Click the add button to add the syslog settings to the list. After adding the syslog settings to the list, you can manage them.

20. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **Review** stage.
21. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.
22. (Optional) If you need to change any data at this stage, right click and select the required option.
23. After the review, click the **Submit** button.
    The Confirmation popup opens.

> **Note:**
> Click **Cancel** to stop running the workflow creation.

24. Click **Ok** to complete the workflow creation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

It takes a while to complete the request.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Launch F5 Standalone Multi NIC in Azure

This workflow launches a Standalone Multi NIC F5 BIG-IP Virtual Machine on Azure. An F5 device will be deployed in Azure. If a new-stack template is selected, then vnets and subnets will be created. If an existing-stack template is selected, the existing vnet and subnet will be utilized. AppViewX checks if the deployment successfully completed. Once completed, the device details are fetched. The F5 device is added into AppViewX and Email will be sent to the user. Also, option to do basic provisioning to the newly added F5 device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **F5 BIG-IP ZTP** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Launch F5 Standalone Multi NIC in Azure** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Launch F5 Standalone Multi NIC in Azure :: FormBuilder** page is displayed.
5. Enter/select the **Azure Account Details**.

> **Note:**
> The entered tenant/application ID must have contributor/owner access to the Azure application provided.

Azure Account Details Section - Field and Description Table

Field	Description
Azure Subscription ID	Enter the subscription ID.
Tenant(Directory) ID	Enter Tenant ID.
Azure Client ID	Enter the client ID.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Azure Client Secret	Enter the client secret key.

6. Enter/select the **Proxy Details**.

Proxy Details Section - Field and Description Table

Field	Description
Connect to Azure via Proxy	Allows to connect with Azure via proxy or not. The options are: • **No** (default) - select this option to connect with Azure without proxy (Internet connection must be required). • **Yes** - select this option to connect with Azure via proxy, then click the **Get Proxies in Appviewx** button to fetch the proxy servers configured in AppViewX to be displayed in the **Proxies** dropdown list.
***Proxy**	Select a proxy servers configured in AppViewX from the dropdown list.

7. Enter/select the **Azure Deployment information**.

Azure Deployment information Section - Field and Description Table

Field	Description
Get F5 on Azure Deployment collections	Click the **Get F5 on Azure Deployment collections** button to get the deployment templates from the AppViewX Collection.
Select Deployment Collection	Select a deployment collection from the dropdown list.
Select the Deployment Template	Select a deployment template type from the dropdown list.
Stack type	It is non editable field and stack type will be auto-filled from the selected deployment template.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

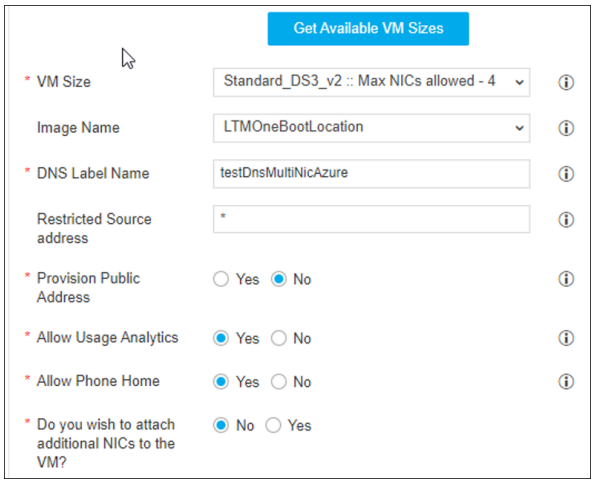
Field	Description
F5 BIG-IP Version	It is non editable field and F5 BIG IP version will be auto-filled from the selected deployment template.
F5 BIG-IP Virtual Edition	It is non editable field and F5 BIG IP virtual edition will be auto-filled from the selected deployment template.
nic	Add n number of NICs using this template.
Content Version	It is non-editable field and content version will be auto-filled from the selected deployment template.

8. Enter/select the **F5 Virtual Machine Details**.

F5 Virtual Machine Details Section - Field and Description Table

Field	Description
F5 Instance Name	Name of the F5 device name which was provided for a device while onboarding it in AppViewX Inventory.
Get available resource groups	Click this button to fetch the Azure resource groups of virtual network.
Virtual Network Resource Group	Select a virtual network resource group from the dropdown list.
Virtual Network	Select a virtual network from the resource group from the dropdown list. **Note:** Make sure that selected the VNET/ Management Subnet has connection established from and to AppViewX, only then the device can be communicated from AppViewX.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
mgmt subnet name	Select a unique management subnet name from the dropdown list. Your subnet must have access to the internet.
External subnet name	Select an external subnet name.
Internal subnet name	Select an Internal subnet name.
mgmt IP-Address	Management IP address to use for the BIG-IP management IP address. By default, this field is filled with DYNAMIC to use a dynamically assigned IP address. Else provide an IP address.
Get Available VM Sizes	Click this button to fetch the available VM sizes. 
VM Size	Select the instance size of the virtual machine from the dropdown list.
Boot Location Type	The options are displayed in this field as it is available in the Azure marketplace for F5. Select a boot location type based on the requirement.
DNS Label Name	Enter a unique DNS Name for the Public IP address used to access the Virtual Machine.
Restricted Source address	This field restricts management access to a specific network or address. Enter an IP address or address range in CIDR notation or asterisk for all sources.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Provision Public Address	Provision to create the Public IP addresses for BIG-IP Management traffic. These addresses will be configured directly on the Azure network interfaces.
Allow Usage Analytics	This deployment can send anonymous template statistics to F5 to help us determine how to improve our solutions. If you select **No**, statistics are not sent.
Allow Phone Home	This deployment can provide F5 with high-level device use information to optimize development resources. If you select **No** the information is not sent.
Do you wish to attach additional NICs to the VM?	Click Yes to attach additional NICs to the VM otherwise select No.

9. Enter/select the **F5 Login Info**.

F5 Login Info Section - Field and Description Table

Field	Description
Login User Name	Enter the user name for the virtual machine. **Note:** The user name must not be "admin".
Authentication Type	Type of authentication to use on the Virtual Machine, password based authentication or key based authentication.
Login Password	Enter the password with the following combination: Minimum eight characters, at least one uppercase letter, one lowercase letter, one number and one special character.

10. Enter/select the **F5 License details**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

F5 License details Section - Field and Description Table

Field	Description
Do you wish to activate license for the F5 device?	The options for activating the license: • **No** (default) - if you select this option, you cannot access F5 modules. • **Yes** - if you select this option, the Base registration key is enabled. Enter the registration key to activate the license.

11. Click **Submit**.
    The **Confirmation** dialog box is displayed.
12. To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

    OR

    To submit the form, click **OK**.

    The validation starts automatically and reaches the **Deployment Info** stage. In this page, the host details are shown. If required, you can download the details in the **.csv** or **.xls** format by clicking the  **(Download)** button.
13. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **VM Status** stage.
14. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **Basic Provisioning** stage.
15. Enter/select the **Basic VIP Creation** details.

Basic VIP Creation Section - Field and Description Table

Field	Description
VIP Name	Enter the name of the VIP.
Destination Address	Enter the destination IP address.
Port	Enter a port.
Pool Name	Enter the pool name.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Load Balancing Method	The load balancing method is used to distribute traffic across pool in this VIP. The default is round-robin. The methods are: • **round-robin** - the system selects the pools sequentially. • **least-connection-node** - The system passes a new connection to the node that has the least number of current connections out of all pools of which a node is a member. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the number of current connections per node, or the fastest node response time. • **least-connection-member** - The system passes a new connection to the node that has the least number of current connections in the pool. This method works best in environments where the servers or other equipment you are load balancing have similar capabilities. This is a dynamic load balancing method, distributing connections based on various aspects of real-time server performance analysis, such as the current number of connections per node or the fastest node response time. • **ratio-member** - The number of connections that each machine receives over time is proportionate to a ratio weight you define for each machine within the pool.

16. Enter/select the **Pool Member Details**.

Pool Member Details Section - Field and Description Table

Field	Description
Member IP	Enter the member IP.
Port	Enter a port.
Pool Members	Click the add button to add the pool members to the table. After adding the pool member details to the list, you can manage them.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

17. Enter/select the **Monitor Details**.

Monitor Details Section - Field and Description Table

Field	Description
Monitor Type	Select the health monitors that are available to add for the pool: • HTTP • HTTPS • TCP • TCPHALFOPEN • Gatewaylcmp
Monitor Name	Enter the name for the monitor.
Send String	The text string that the monitor sends to the target object. You must include \r\n at the end of a non-empty Send String. To retrieve a specific page from a website, specify a fully qualified path name.
Receive String	The regular expression representing the text string that the monitor looks for in the returned resource. The most common receive expressions contain a text string that is included in an HTML file on your site. The text string can be regular text, HTML tags, or image names, and the associated operation is not case-sensitive.

18. Enter/select the **Other Settings**.

Other Settings Section - Field and Description Table

Field	Description
Banner Name	Enter the name for banner.
Banner State	Select a state for the banner. The possible states are: • enabled • disabled
DNS Address	Enter the DNS address.
NTP Address	Enter the NTP address.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

19. Enter/select the **Syslog Setting** details.

Syslog Setting Section - Field and Description Table

Field	Description
Syslog Name	Enter the name for syslog.
Syslog Ipddress	Enter the syslog IP address.
Syslog Port	Enter the syslog port.
SYSLOG	Click the add button to add the syslog settings to the list. After adding the syslog settings to the list, you can manage them.

20. Click **Submit** to submit the form.
    The validation starts automatically and reaches the **Review** stage.
21. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs.
22. (Optional) If you need to change any data at this stage, right click and select the required option.
23. After the review, click the **Submit** button.
    The Confirmation popup opens.

> **Note:**
> Click **Cancel** to stop running the workflow creation.

24. Click **Ok** to complete the workflow creation.
    It takes a while to complete the request.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Migration

Netscaler to F5 BIG-IP

This workflow is designed to migrate load balancer (LB) configurations from Citrix NetScaler to F5 BIG-IP.

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Miration** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Netscaler to F5 BIG-IP** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Netscaler to F5 BIG-IP :: FormBuilder** page is displayed.
5. Enter/select the **Select Source Device - Citrix NetScaler** field information.

Select Source Device - Citrix NetScaler Section - Field and Description Table

Field	Description
***Select NetScaler Device**	Retrieve and select a NetScaler device.
***Select Partition**	Retrieve and select a partition.
***Select Application**	Retrieve and select an application.
Select GSLB Applications	Retrieve and select a GSLB application.
**: Mandatory field*	

6. Enter/select the **Select Destination Device - F5 BIG-IP**.

Select Destination Device - F5 BIG-IP Section - Field and Description Table

Field	Description
Fetch Active F5 Devices Managed in AppViewX	Click this button to select the target F5 device for migration.
***Select Target F5 Device**	Select a targeted F5 device.
Do you want to select partition in the F5 Device?	By default, No option is selected. To select a partition in the F5 device, click Yes option, and then select a parition.

7. On the **F5 Config Option** section, select a config push method.
8. On the **Pre-validation** section, click the **Validate** button.
   The Prevalidation Info are fetched.
9. Once the validation is successful, click **Submit**.
   The **Confirmation** dialog box is displayed.
10. (Optional) To save this form so you can edit it later, click **Save Draft** and then click **OK**.
    The form will be saved as **Open request** under **Request** > **My Request**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

11. Click **Ok** to complete the workflow creation.
    It takes a while to complete the request.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

GITOps

GitHub - Create New Project

This workflow creates a new project in GitHub Cloud.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **GITOps** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **GitHub - Create New Project** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > GitHub - Create New Project :: FormBuilder** page is displayed.
5. Enter/select the field information in the **GitHub - Create New Repository** section.

GitHub - Create New Repository Section - Field and Description Table

Field	Description
***Enter Repository Name**	Enter the repository name for the GitHub Project. The project will be created with this name.
***Enter Description**	Enter the Description for the GitHub Project.

6. Enter the field information in the **GitHub - Details** section.

GitHub - Details Section - Field and Description Table

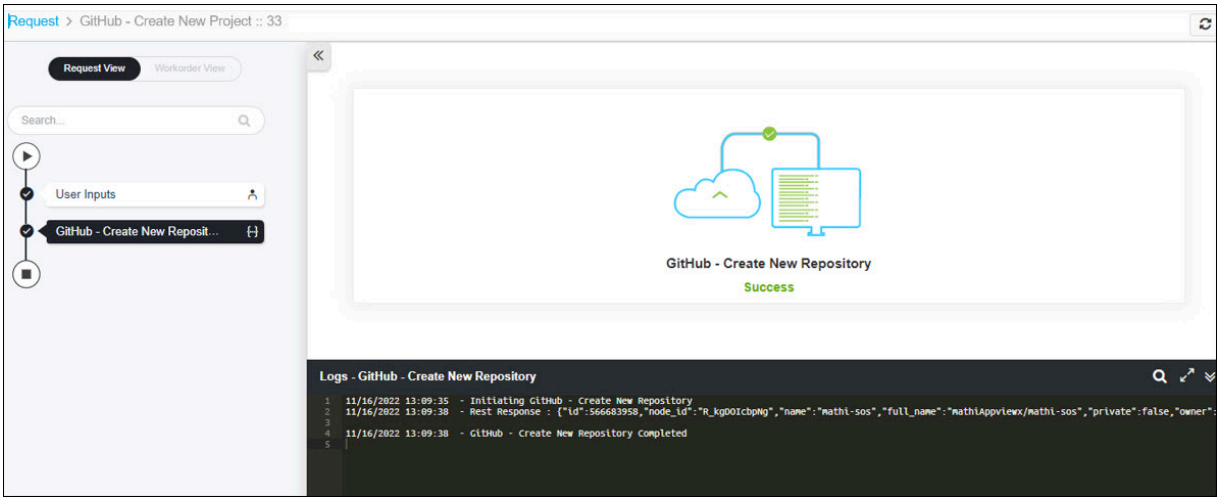
Field	Description
***Username**	Enter the valid username that has privelege to create projects.
***Password**	Enter the password or API token.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. Click the **Next** button.
   The **Confirmation** popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the **Confirmation** popup window. The form will be saved as **Open** request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

8. Click **Ok** to submit the form.
9. The workflow is completed and the new project is created in the GitHub.



> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

GitLab - Create New Project

This workflow creates a new project in GitLab Cloud/OnPrem.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **GITOps** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **GitLab - Create New Project** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click 

.

   The **Request > GitLab - Create New Project :: FormBuilder** page is displayed.
5. Enter the field information in the **GitLab - Create Projects** section.

   GitLab - Create Projects Section - Field and Description Table

Field	Description
***Enter GitLab Url**	Enter the GitLab URL.
***Enter Private Token**	Enter the private access token.
***Enter Project Name**	Enter the GitLab project name.

6. Click the **Next** button.

   The **Confirmation** popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the **Confirmation** popup window. The form will be saved as **Open** request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

7. Click **Ok** to submit the form.
8. The workflow is completed and the new project is created in the GitLab.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

GITOps - Create or Modify GTM Application Services

Prerequisites: F5 device must have AS3 service enabled and running.

This workflow pushes the AS3 configurations to an active GitLab repository. Review can be automated from Git using custom logic script.

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **GITOps** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **GITOps - Create or Modify GTM Application Services** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > GITOps - Create or Modify GTM Application Services :: FormBuilder** page is displayed.
5. Enter the field information in the **General Properties** section.
6. The following table provides the field description for the **General Properties** section.

General Properties Section - Field and Description Table

Field	Description
***Action**	Select the action to create or modify an application service. The options are: • **Create** (default) - select this option to create an application service for a device. • **Modify** - select this option to choose an existing application and partition.
***Device Name**	Select the device name from the drop-down list.
***Partition Type**	Partition in which objects will be created. Select a partition type. The options are: • **Create New Partition** (default) - select this option to create a new partition. • **Existing Partition** - select this option to choose an existing partition. **Note:** Partition dropdown lists only the partitions created via AS3 triggered API calls.
***Enter Partition Name**	Enter the partition name.
***Application Type**	Select a application type. The options are:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **Create New Application** (default) - select this option to create a new application. • **Existing Application** - select this option to choose an existing partition.
***Enter Application Name**	Enter the FQDN of the virtual server.

7. Enter the field information in the **Wide IP** section.

Wide IP Section - Field and Description Table

Field	Description
***WideIP FQDN**	Enter the fully qualified domain name.
***WideIP Type**	Select a Wide IP type from the dropdown list.

8. Enter the field information in the **Pool** section.

Pool Section - Field and Description Table

Field	Description
***Pool Name**	Enter the pool name.
***Pools Load Balancing Method**	Select a load balancing method for the pools.
Pool Members Preferred Load Balancing Method	Select a preferred load balancing method for the pool members. The system tries this method first.
Pool Members Alternate Load Balancing Method	Select the alternate load balancing method. The system tries this method if the Preferred method is unsuccessful in picking the WideIP.
Pool Members Fallback Load Balancing Method	Select the fallback load balancing method. The system tries this method if the Preferred and alternate methods are unsuccessful in picking the WideIP.

9. Enter the field information in the **Pool Members** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Pool Members Section - Field and Description

Field	Description
***Pool member Type**	Select a pool member type from the deopdown list. The types are: • Generic Host (default) - When you select this option you need to select GTM server and virtual servers. • Existing VIP -. When you select this option you need to select GTM server and existing VIP.
***GTM Server**	Select a GTM server from the dropdown list.

10. Enter the field information in the **Monitors** section.

Monitors Section - Field and Description Table

Field	Description
***Do you want to associate a HTTP/HTTPS monitor to the pool?**	Select Yes to associate a HTTP/HTTP monitor, and then select monitor from the dropdown list otherwise select No.

11. Click the **Submit** button.
    The **Confirmation** popup opens.

> **Note:**
> • If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the **Confirmation** popup window. The form will be saved as **Open** request under **Request** > **My Request**.
> • If you want to cancel this form, click the **Cancel** button.

12. Click **Ok** to submit the form.
13. The workflow is completed and the GTM application service is created/updated in the GitHub.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

GITOps - Create or Modify LTM Application Services

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

F5 device must have AS3 service enabled and running.

This workflow pushes the AS3 configurations to an active GitLab repository. Review can be automated from Git using custom logic script.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **GITOps** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **GITOps - Create or Modify LTM Application Services** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > GITOps - Create or Modify LTM Application Services :: FormBuilder** page is displayed.
5. Enter the field information in the **General Properties** section.

General Properties Section - Field and Description Table

Field	Description
***Action**	Select the action to create or modify an application service. The options are: • **Create** (default) - select this option to create an application service for a device. • **Modify** - select this option to choose an existing application and partition.
***Device Name**	Select the device name from the drop-down list.
***Partition Type**	Partition in which objects will be created. Select a partition type. The options are: • **Create New Partition** (default) - select this option to create a new partition. • **Existing Partition** - select this option to choose an existing partition. **Note:** Partition dropdown lists only the partitions created via AS3 triggered API calls.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Enter Partition Name**	Enter the partition name.
***Application Type**	Select a application type. The options are: • **Create New Application** (default) - select this option to create a new application. • **Existing Application** - select this option to choose an existing partition.
***Enter Application Name**	Enter the FQDN of the virtual server.

6. Enter the field information in the **Enter Virtual Server Details** section.

Enter Virtual Server Details- Field and Description Table

Field	Description
***Virtual Server Name**	Enter the virtual server name.
***Virtual Server IP**	Select virtual server IP address.
Virtual Server Port	Enter virtual server port.

7. Enter the field information in the **Enter Pool Details** section.

Enter Pool Details Section - Field and Description Table

Field	Description
***Pool Name**	Enter the pool name.
***Pool Members Load Balancing Method**	Select a load balancing method for the pools.
Pool Member Addresses	Enter pool member' IP addresses.
Pool Member Service Port	Enter pool member service port.
Pool Members	Click the add button to add the pool member's details to the Pool Members table.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	Note: After adding the pool member's details to the Pool Members table, you can modify them.

8. Select the field information in the **Additional Details** section.
9. Click the **Submit** button.
   The **Confirmation** popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the **Confirmation** popup window. The form will be saved as **Open** request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

10. Click **Ok** to submit the form.
11. The workflow is completed and the GTM application service is created/updated in the GitHub.

> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

GITOps - Delete LTM or GTM Application Services

F5 device must have AS3 service enabled and running.

This workflow deletes a LTM or GTM application service from GitLab device by fetching all the LTM or GTM application service from the device based on device name. You can delete any LTM or GTM application service from the list.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **GITOps** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **GITOps - Delete LTM or GTM Application Services** workflow.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The **Run** and **Schedule** buttons are displayed.

4. Click 

.

   The **Request > GITOps - Delete LTM or GTM Application Services :: FormBuilder** page is displayed.

5. Enter the field information in the **General Properties** section.

General Properties Section - Field and Description Table

Field	Description
***F5 Device**	Select a device from the dropdown list.
***Select the object to Delete**	Select an object from the dropdown list.
***Select Partition**	Select a partition from the dropdown list.
***Select Application**	Select an application from the dropdown list.

6. Enter the field information in the **Delete Action** section.

Delete Action Section - Field and Description Table

Field	Description
***Delete Pool**	Select any of the following options to delete a pool or not: • Yes • No

7. Click the **Submit** button.

Note:
- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button.

8. Click **Ok** to submit the form.

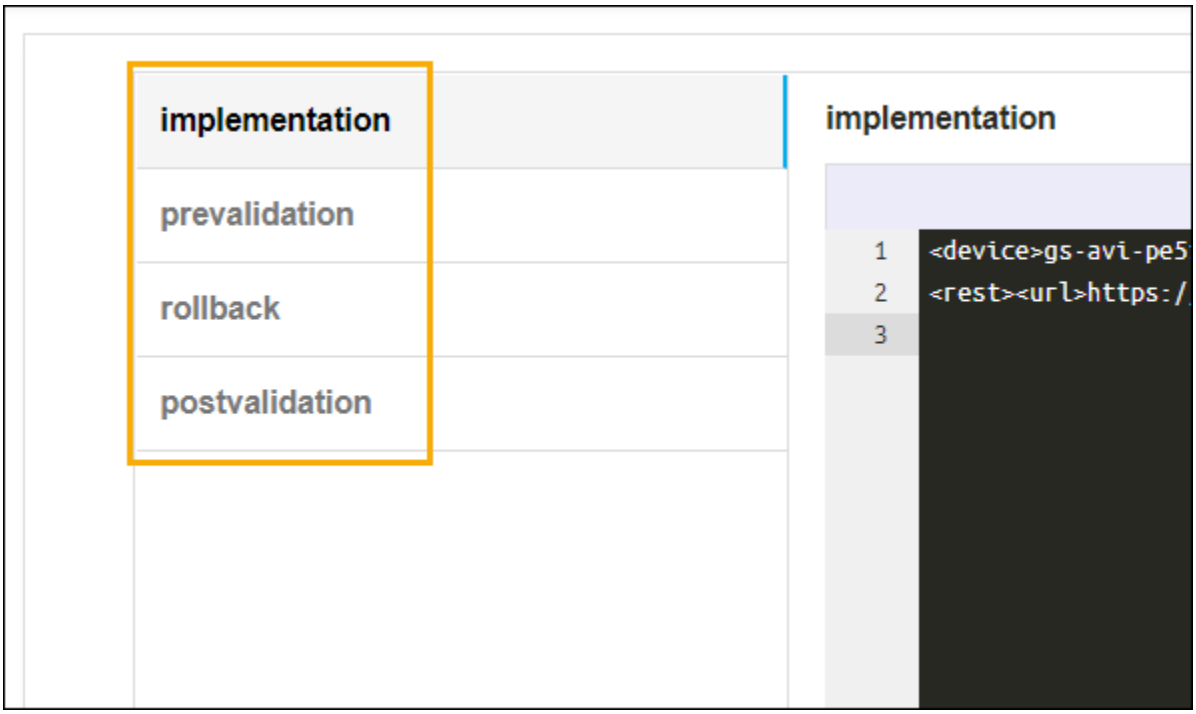
   The validation starts automatically and reaches the Review stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:

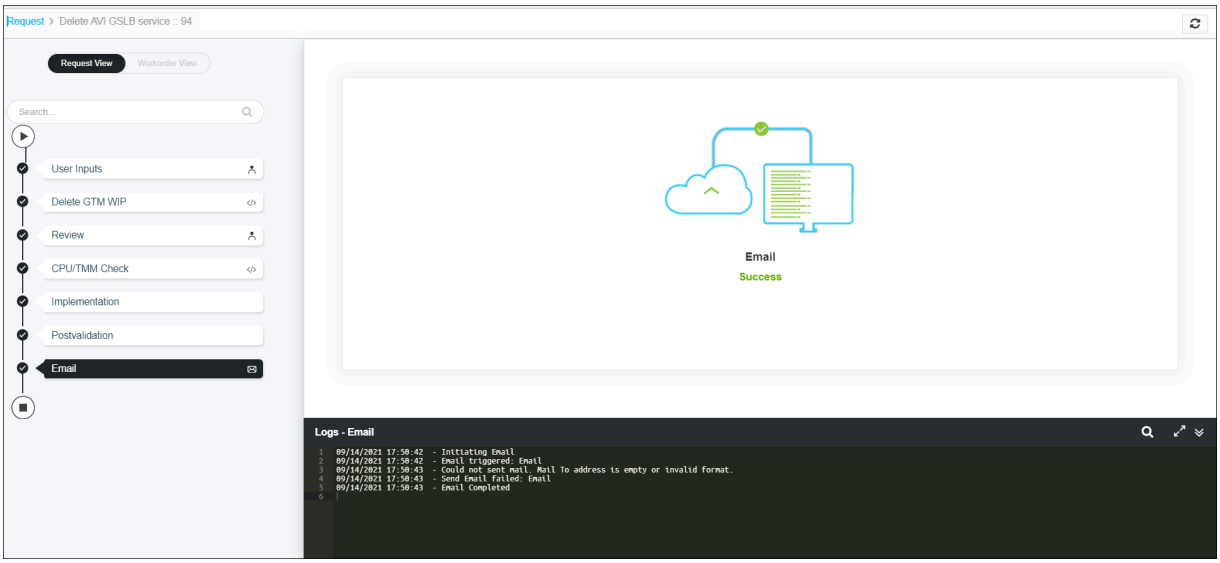


10. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
11. After the review, click the **Submit** button.
    The Confirmation popup opens.

Note:
To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
13. The workflow is created and the email is triggered to the configured email IDs.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Note:

The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

GitOps - Pull Config and Execute AS3

This workflow pulls the AS3 configurations from a file in GitHub repository.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **GITOps** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **GitOps - Pull Config and Execute AS3** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > GitOps - Pull Config and Execute AS3 :: FormBuilder** page is displayed.
5. Select the field information in the **AS3 Device** section.

AS3 Device Section - Filed and Description Table

Field	Description
***Select Device**	Select a device from the dropdown list.

6. Enter the field information in the **GitHub Repo Details** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

GitHub Repo Details Section - Field and Description Table

Field	Description
***Repo Owner**	Enter the name of the repository owner.
***Repo Name**	Enter the name of the repository.
***File Path**	Enter the file path.
***GitHub Username**	Enter GitHub username.
GitHub Password	Enter GitHub password.

7. Click the **Next** button.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

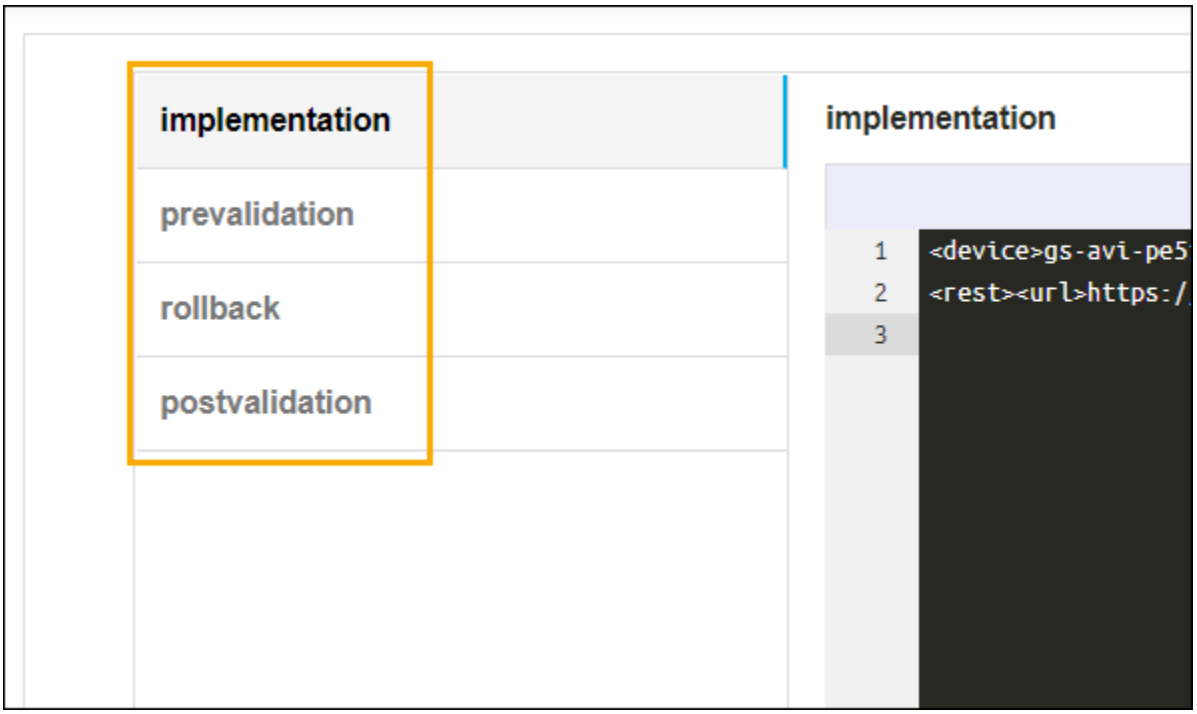
8. Click **Ok** to submit the form.

   The validation starts automatically and reaches the Review stage.

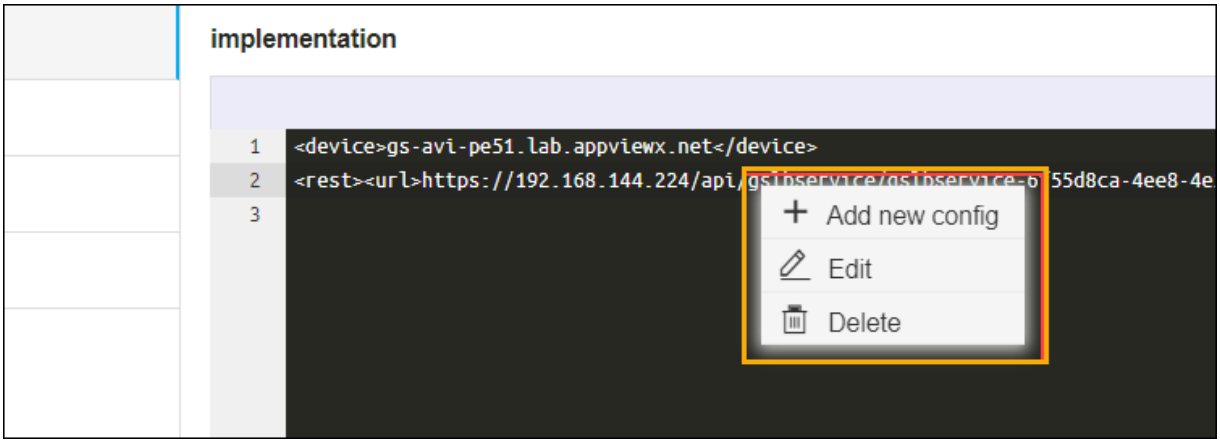
> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



10. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.



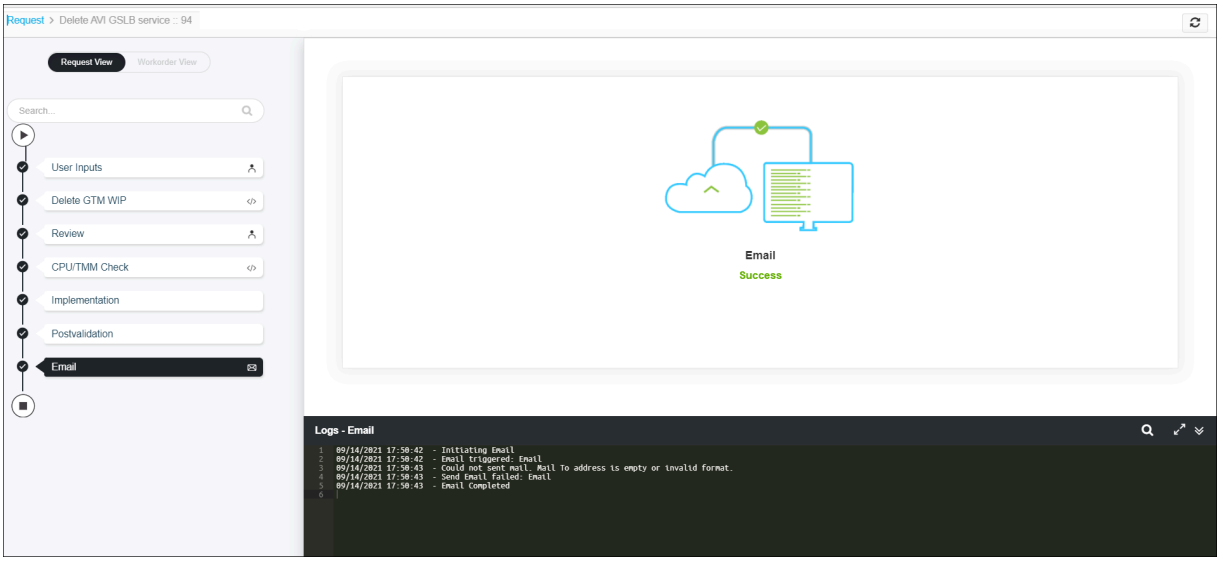
11. After the review, click the **Submit** button.
    The Confirmation popup opens.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

13. The workflow is created and the email is triggered to the configured email IDs.



Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Nginx Plus

Create NginxPlus LB Server

This workflow enables you to create HTTP or TCP load balancing server in existing or new source configuration.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Nginx Plus** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM VIP** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Create F5 LTM VIP :: FormBuilder** page is displayed.
5. Enter or select the field information in the **Device and Load balancing type** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Device and Load balancing type Section - Field and Description Table

Field	Description
***Data Center**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are added without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Source Config Option**	Select this option to choose an existing or new source config.
***Nginx Source Path**	Enter the path of the existing source config.
***Load Balancing Type**	Select the load balancing type: • **HTTP** • **TCP**
***Load Balancing Method**	Select a load balancing method from the dropdown list.
Server Name	Enter the server name, if the **Load Balancing Type** is selected as HTTP/

6. Enter or select the field information in the **LB server and UpStream** section.

LB server and UpStream Section - Field and Description Table

Field	Description
***LB server IP**	Enter the IP address of the Load balancing (LB) server.
***LB server Port**	Enter the port of the Load balancing (LB) server.
***Upstream Server**	Enter the upstream server name.
Health Check	Select a health check option:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **Active** (default) • **Passive**
***Upstream Server IP and Port**	Enter the upstream server IP address and port no. It must be in the format of *<IP Address>:<port no.>*.
***Weight**	The weight of the topology record. Enter the weight in the numerals.
***Server Status**	Select a server status from the dropdown list.
Server Type	Select a server type from the dropdown list: • **Normal** (default) • **Backup**
Fail Timeout	Enter the timeout value in seconds.
Max Fail	Enter the maximum allowed failures.
Slow Start	Enter the slow start value.
Upstream Server Grid	click the Add  button to add the server and upstream details to the Upstream Server Grid.

7. Enter or select the field information in the **Persistence** section.

Persistence Section - Field and Description Table

Field	Description
Sticky Cookie Persistence	By default, this option is disabled. When enabled Sticky cookie persistence helps to reconnect the server with the client to the same resource as previous requests.

8. Enter or select the field information in the **SSL** section.

SSL Section - Filed and Description Table

Field	Description
***Do you want to add SSL**	Select any of the following options to enable SSL certificates.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
	• **No** (default) - The only Client will have an SSL Certificate • **Yes** - Both Client and Servers will have SSL certificates

9. Click the **Submit** button.

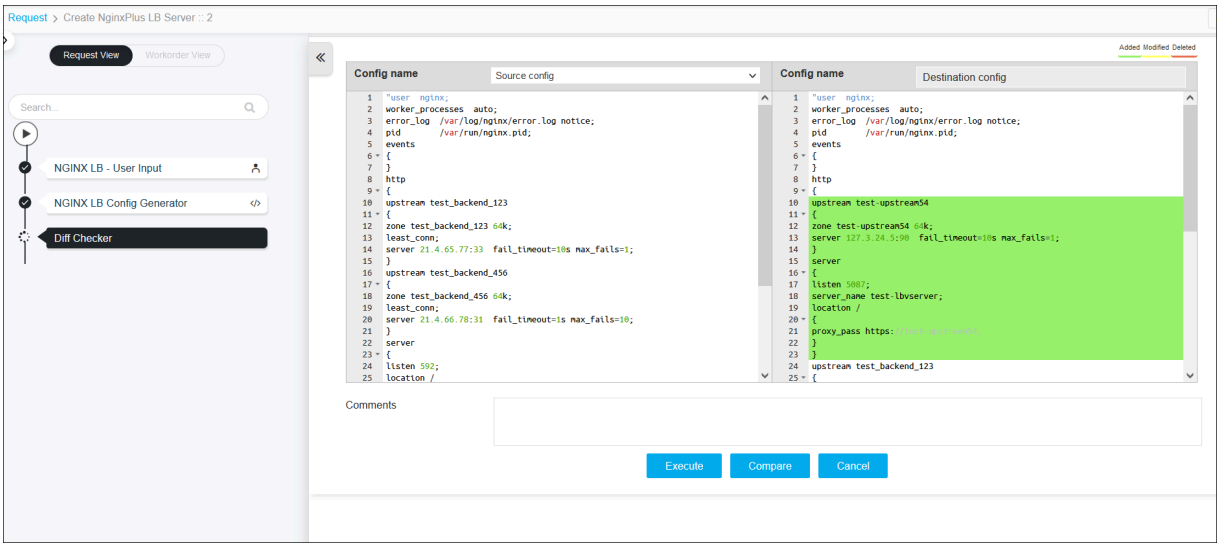
   The Confirmation popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

10. Click **Ok** to submit the form.

    The validation starts automatically and reaches the **Diff Checker** stage.

11. Compare the updates and click the **Execute** button.



    The validation starts automatically and reaches the **Backup Master Config** stage.

12. Click the **Submit** button.

    The validation starts automatically and reaches the **NGINX Config Check** stage.

13. Click the **Execute** button.

    The validation starts automatically and reaches the **NGINX Config Check Reload** stage.

14. Click the **Execute** button.

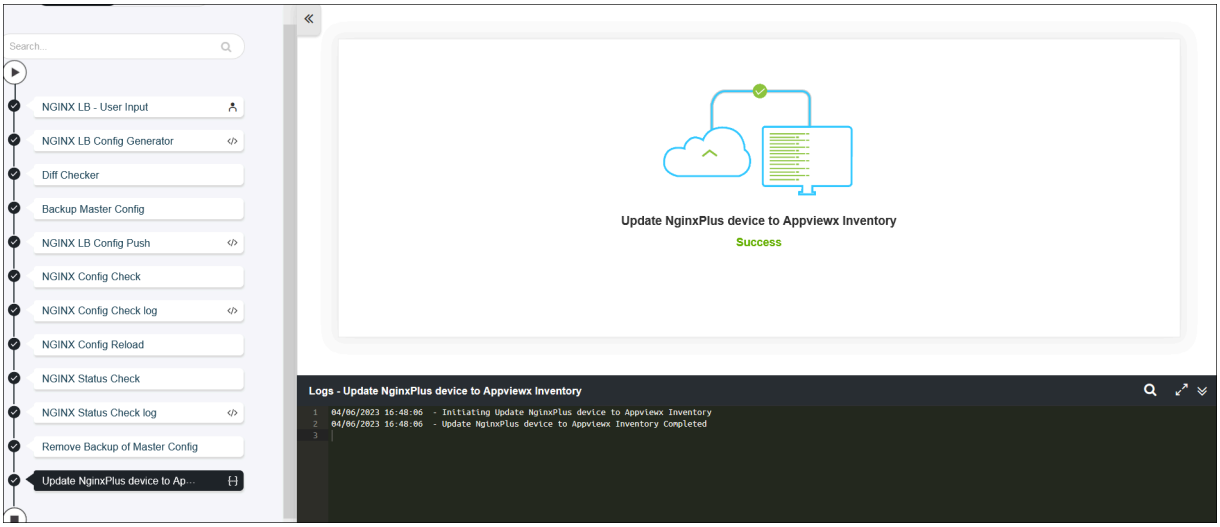
    The validation starts automatically and reaches the **NGINX Config Check Reload** stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

15. Click the **Execute** button.
    The validation starts automatically and reaches the **Remove Backup of Master Config** stage.
16. Click the **Submit** button.
    The Confirmation popup opens.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

17. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
18. The workflow is completed.



> **Note:**
> The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Create NginxPlus LB Server with Service Now

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Nginx Plus** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create NginxPlus LB Server with Service Now** workflow.
   The **Run** and **Schedule** buttons are displayed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Click Run.

   The **Request > Create NginxPlus LB Server with Service Now :: FormBuilder** page is displayed.

5. Enter or select the field information in the **Device and Load balancing type** section.

Device and Load balancing type Section - Field and Description able

Field	Description
***Data Center**	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices which are added without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device Name**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Source Config Option**	Select this option to choose an existing or new source config.
***Nginx Source Path**	Enter the path of the existing source config.
***Load Balancing Type**	Select the load balancing type: • **HTTP** • **TCP**
***Load Balancing Method**	Select a load balancing method from the dropdown list.
Server Name	Enter the server name, if the **Load Balancing Type** is selected as HTTP/

6. Enter or select the field information in the **LB server and UpStream** section.

LB server and UpStream Section - Field and Description Table

Field	Description
***LB server IP**	Enter the IP address of the Load balancing (LB) server.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***LB server Port**	Enter the port of the Load balancing (LB) server.
***Upstream Server**	Enter the upstream server name.
Health Check	Select a health check option: • **Active** (default) • **Passive**
***Upstream Server IP and Port**	Enter the upstream server IP address and port no. It must be in the format of *<IP Address>:<port no.>*.
***Weight**	The weight of the topology record. Enter the weight in the numerals.
***Server Status**	Select a server status from the dropdown list.
Server Type	Select a server type from the dropdown list: • **Normal** (default) • **Backup**
Fail Timeout	Enter the timeout value in seconds.
Max Fail	Enter the maximum allowed failures.
Slow Start	Enter the slow start value.
Upstream Server Grid	click the Add  button to add the server and upstream details to the Upstream Server Grid.

7. Enter or select the field information in the **Persistence** section.

Persistence Section - Field and Description Table

Field	Description
Sticky Cookie Persistence	Select one of the following options: • No (default) • Yes

8. Enter or select the field information in the **SSL** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

SSL Section - Table and Description Table

Field	Description
Sticky Cookie Persistence	By default, this option is disabled. When enabled Sticky cookie persistence helps to reconnect the server with the client to the same resource as previous requests.

9. Enter or select the field information in the **Change Management** section.

Change Management Section - Field and Description Table

Field	Description

*Do you want to integrate ServiceNow Ticket?	Select this option to create a service now ticket or not. The options are:
	• **No** (default)
	• **Yes** - select this option to create a ServiceNow ticket and select the follwing details:
	  • **Timezone** – select the timezone from the drop-down list.
	  • ***Start Time** – select the start date and time from the calendar.
	  • ***End Time** – select the end date and time from the calendar.

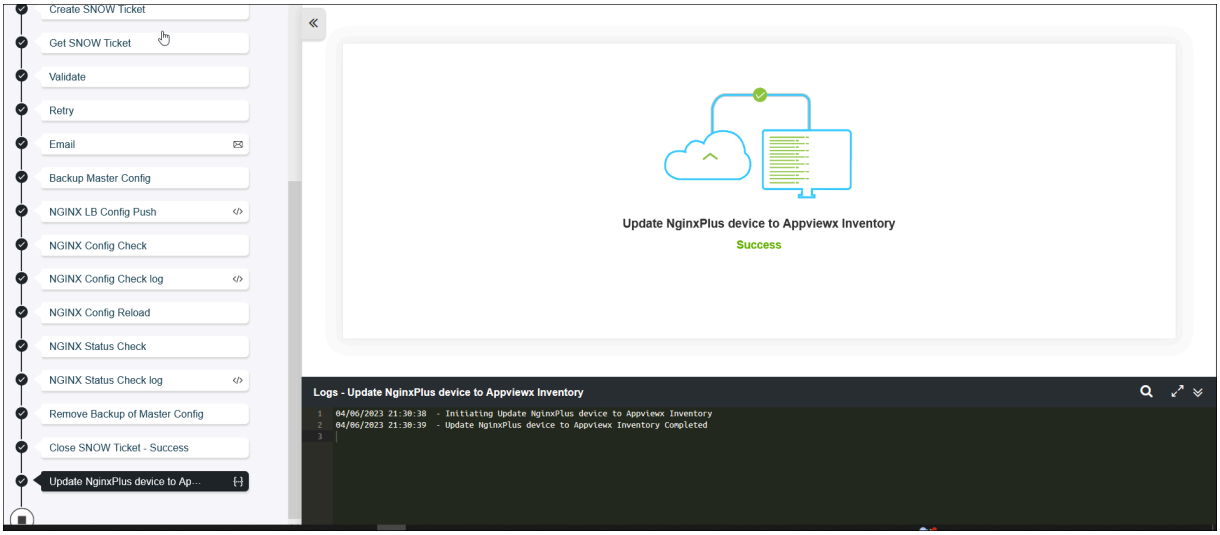
10. Click the **Submit** button.
    The Confirmation popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

11. In the **Diff Checker** stage, click the **Compare** button to view the updates, and then click the **Execute** button.
12. In the **Get SNOW Ticket** stage, click the **Execute** button.
13. In the **Backup Mater Config** stage, click the **Execute** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

14. In the **NGINX Config Check** stage, click the **Execute** button.
15. In the **NGINX Config Reload** stage, click the **Execute** button.
16. In the **NGINX Status Check** stage, click the **Execute** button.
17. In the **Remove Backup of Master Config** stage, click the **Submit** button.
18. The workflow is created and the email is triggered to the configured email IDs.



Note:
The stages of the request are shown in the left side of the screen. To view a particular stage of the request, click the respective stage.

Delete NginxPlus LB Server

This workflow enables you to delete HTTP or TCP LB Server and Upstream Servers.

To run this workflow:

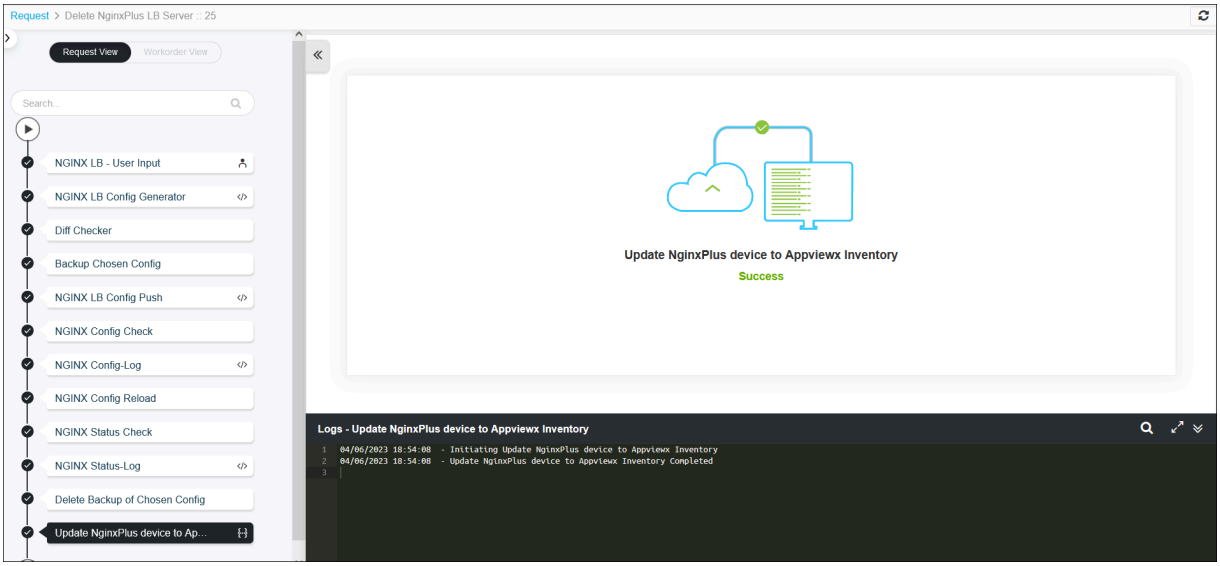
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Nginx Plus** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete NginxPlus LB Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete NginxPlus LB Server :: FormBuilder** page is displayed.
5. Select a datacenter, device, and Nginx source path in the **Device Details** section.
6. Choose the load balancing type and fetch and select the load balancing IP and Port in the **LB Server And Upstream** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. Click the **Submit** button.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

8. In the **Diff Checker** stage, click the <Compare> button to view the updates, and then click the **Submit** button.
9. In the **Backup Chosen Config** stage, click the <Submit> button.
10. In the **NGINX Config Check** stage, click the <Execute> button.
11. In the **NGINX Config Reload** stage, click the <Execute> button.
12. In the **NGINX Status Check** stage, click the <Submit> button.
13. In the **Delete Backup of Chosen Config** stage, click the <Execute> button.
14. The workflow is created.



> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Delete NginxPlus LB Server with Service Now

This workflow enables you to delete HTTP or TCP LB Server and Upstream Servers.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Nginx Plus** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete NginxPlus LB Server with Service Now** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Delete NginxPlus LB Server with Service Now :: FormBuilder** page is displayed.
5. Select a datacenter, device, and Nginx source path in the **Device Details** section.
6. Choose the load balancing type and fetch and select the load balancing IP and Port in the **LB Server And Upstream** section.
7. Choose **Yes** to create a ServiceNow ticket in the **Change Management** section. Otherwise choose **No**.
8. Click the **Submit** button.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

9. In the **Diff Checker** stage, click the **Compare** button to view the updates, and then click the **Execute** button.
10. In the **Get SNOW Ticket** stage, click the **Execute** button.
11. In the **Backup Mater Config** stage, click the **Execute** button.
12. In the **NGINX Config Check** stage, click the **Execute** button.
13. In the **NGINX Config Reload** stage, click the **Execute** button.
14. In the **NGINX Status Check** stage, click the **Execute** button.
15. In the **Remove Backup of Master Config** stage, click the **Submit**
16. The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Modify NginxPlus LB Server

Prerequisites

- /etc/nginx/ssl Folder must be present.
- In master file (/etc/nginx/nginx.conf), http block must be present for creating a HTTP LB.
- In master file (/etc/nginx/nginx.conf), stream block must be present for creating a TCP LB.

This workflow modifies a GSLB virtual server on a Citrix device by fetching the device details.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Nginx Plus** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create F5 LTM VIP** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

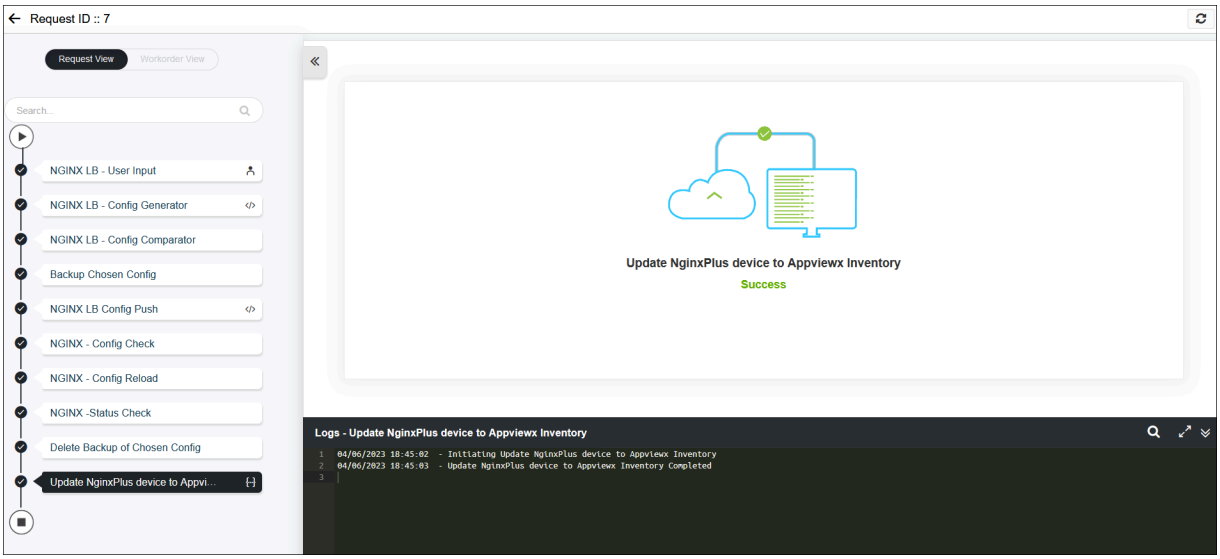
.
   The **Request > Create F5 LTM VIP :: FormBuilder** page is displayed.
5. In the **Device and Load Balancing Type** section, select the following:
   - Data Center
   - Device Name
   - Nginx Source Path
   - Modification Preference - select Virtual Server or Orphan Upstreams
   - Load Balacing Type - select HTTP or TCP
6. In the **Load Balancing Server and Upstream** section, modify the upstream server information as follows:
   - If you selected **Modification Preference** as a Virtual Server, fetch and select the LB server IP and Port, and modify the Upstream server information as needed.
   - If you selected **Modification Preference** as Orphan Upstreams, then fetch and modify the Upstream server information as needed.
7. Modify the Persistence and SSL options as needed.
8. Click the **Submit** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button. The form will be saved as Cancelled request under **Request** > **My Request**.

9. Click **Ok** to submit the form.
   The validation starts automatically.
10. In the **NGINX LB - Config Comparator** stage, compare the updated by clicking the **Compare** button, and then click the **Execute** button.
11. In the **Backup Chosen Config** stage, click the **Submit** button.
12. In the **NGINX Config Check** stage, click the **Execute** button.
13. In the **NGINX Config Reload** stage, click the **Execute** button.
14. In the **NGINX Status Check** stage, click the **Execute** button.
15. In the **Delete Backup of Chosen Config** stage, click the **Submit** button.
16. The workflow is created.



Note:

The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Modify NginxPlus LB Server with Service Now

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Prerequisites

- /etc/nginx/ssl Folder must be present.
- In master file (/etc/nginx/nginx.conf), http block must be present for creating a HTTP LB.
- In master file (/etc/nginx/nginx.conf), stream block must be present for creating a TCP LB.

This workflow modifies a GSLB virtual server on a Citrix device by fetching the device details.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Nginx Plus** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify NginxPlus LB Server with Service Now** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Modify NginxPlus LB Server with Service Now :: FormBuilder** page is displayed.
5. In the **Device and Load Balancing Type** section, select the following:
   - Data Center
   - Device Name
   - Nginx Source Path
   - Modification Preference - select Virtual Server or Orphan Upstreams
   - Load Balacing Type - select HTTP or TCP
6. In the **Load Balancing Server and Upstream** section, modify the upstream server information as follows:
   - If you selected Modification Preference as Virtual Server, fetch and select the LB server IP and Port, and modify the Upstream server information as needed.
   - If you selected Modification Preference as Orphan Upstreams, then fetch and modify the Upstream server information as needed.
7. Modify the Persistence, SSL, and Change Management options as needed.
8. In the **Diff Checker** stage, click the **Compare** button to view the updates, and then click the **Execute** button.
9. In the **Get SNOW Ticket** stage, click the **Execute** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

10. In the **Backup Mater Config** stage, click the **Execute** button.
11. In the **NGINX Config Check** stage, click the **Execute** button.
12. In the **NGINX Config Reload** stage, click the **Execute** button.
13. In the **NGINX Status Check** stage, click the **Execute** button.
14. In the **Remove Backup of Master Config** stage, click the **Submit**
15. The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Citrix SLB

Create Citrix SLB Virtual Server

This workflow creates a virtual server on Citrix LTM devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix SLB** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create Citrix SLB Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Create Citrix SLB Virtual Server :: FormBuilder** page is displayed.
5. Select the field information in the **Citrix Device Details** section.
6. The following table provides the field description for the **Citrix Device Details** section:

Citrix Device Details Section - Field and Description Table

Field	Description
Data center	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
***Citrix LTM Device**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

7. Enter or select the field information in the **Citrix Vip Details** section.

Citrix Vip Details Section - Field and Description Table

Field	Description
***Virtual Server Name**	Enter the name of the virtual server.
***Virtual Server IP**	Enter the VIP address.
***Virtual Server Port**	Enter a server port.

8. Enter or select the field information in the **Service or Service Group Details** section.

Service or Service Group Details Section - Field and Description Table

Field	Description
***Protocol**	Enter the name of the virtual server.
***Type**	Enter the VIP address.
***Service Name**	Enter a service port.
***Service IP Address**	Enter service IP address.
***Service Port**	Enter service port.
***Service Details**	Click Add () button to add the service(s) to the Service Details.

9. Enter or select the field information in the **Monitor** section of Form Input.

Monitor Section - Field and Description Table

Field	Description
***Monitor Settings**	By default, No option is selected. Select Yes to choose the health monitors.

10. Click the **Submit** button.

    The Confirmation popup opens.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

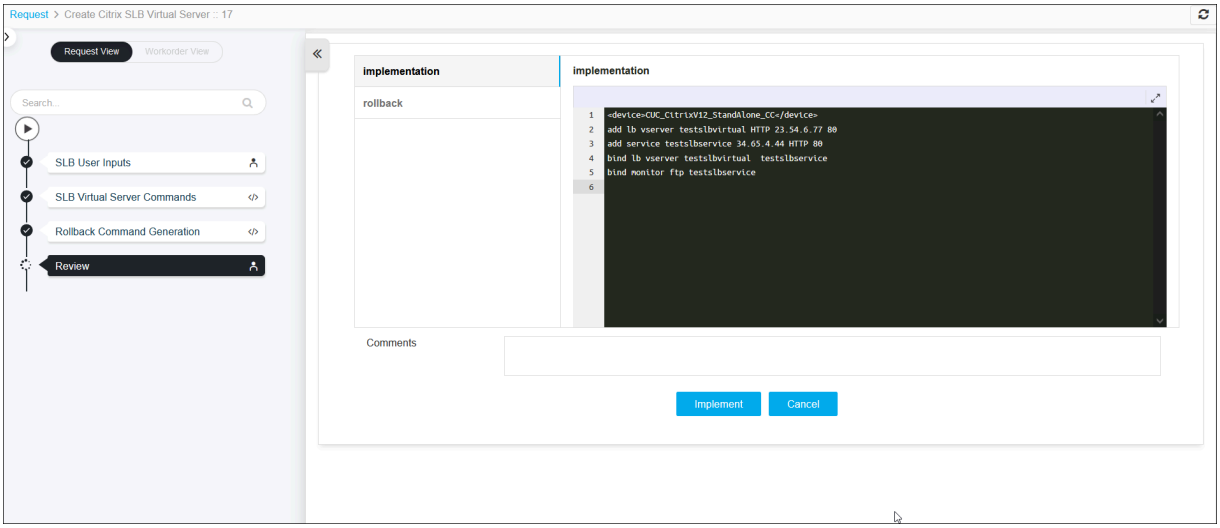
- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button.

11. Click **Ok** to submit the form.
    The validation starts automatically and reaches the Review stage.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

12. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



13. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
14. After the review, click the **Implement** button.
    The Confirmation popup opens.

Note:

To stop running the workflow creation, click **Cancel**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

15. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
16. The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Delete Citrix SLB Virtual Server

By running this workflow, you can delete a SLB virtual server from Citrix device by fetching all the SLB virtual server from the device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix SLB** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete Citrix SLB Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

   .
   The **Request > Delete Citrix SLB Virtual Server :: FormBuilder** page is displayed.
5. Select a datacenter, device, and virtual server in the **Device and Virtual Server Details** section.
6. Click the **Submit** button.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

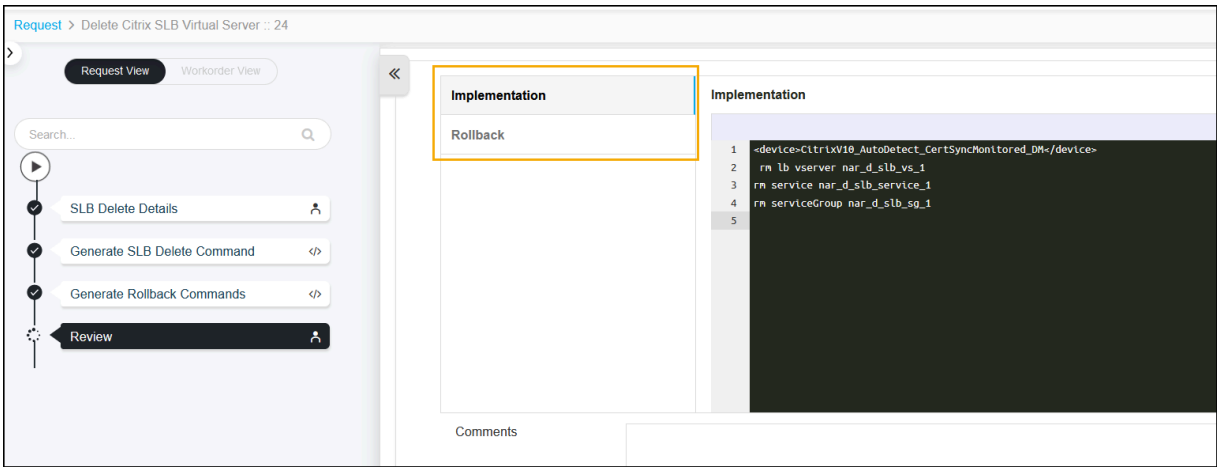
7. Click **Ok** to submit the form.
   The validation starts automatically and reaches the Review stage.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



9. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
10. After the review, click the **Implement** button.
    The Confirmation popup opens.

Note:

To stop running the workflow creation, click **Cancel**.

11. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
12. The workflow is created and the email is triggered to the configured email IDs.

Note:

The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Modify Citrix SLB Virtual Server

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

By running this workflow, you can modify a SLB virtual server on Citrix device by fetching the device details.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix SLB** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify Citrix SLB Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Modify Citrix SLB Virtual Server :: FormBuilder** page is displayed.
5. Fetch and select the details of device, virtual server, and service or service group.
6. Modify the necessary details of a virtual service for a device.
7. Click the **Submit** button.
   The Confirmation popup opens.

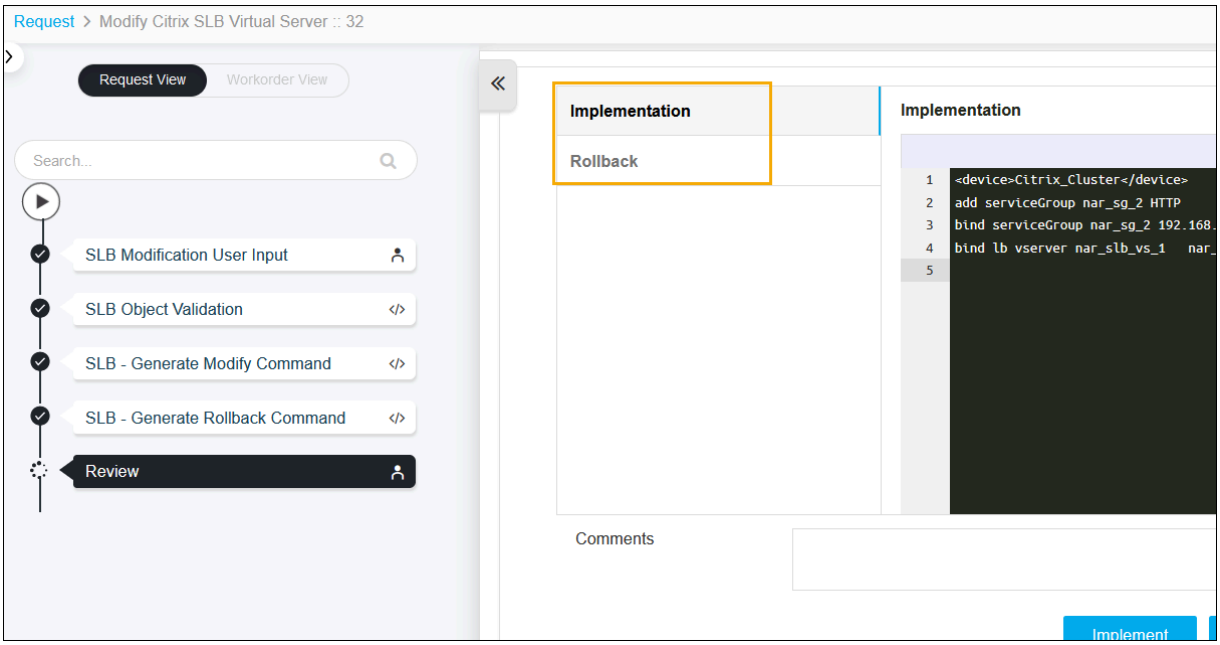
   Note:
   - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
   - If you want to cancel this form, click the **Cancel** button.

8. Click **Ok** to submit the form.
   The validation starts automatically and reaches the Review stage.

   Note:
   - If there is any failure at this stage, start a new request by providing the correct data.
   - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the implementation, rollback, and postvalidation tabs:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



10. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
11. After the review, click the **Implement** button.
    The Confirmation popup opens.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
13. The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Citrix GSLB

Create Citrix GSLB Virtual Server

This workflow creates a virtual server on Citrix devices.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix GSLB** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create Citrix GSLB Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Create Citrix GSLB Virtual Server :: FormBuilder** page is displayed.
5. Select the field information in the **Citrix Device Details** section.

Citrix Device Details Section - Field and Description Table

Field	Description
Data center	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Device**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.
***Site**	Select a site from dropdown list.

6. Enter or select the field information in the **GSLB Virtual Server Details** section.

GSLB Virtual Server Details Section - Field and Description Table

Field	Description
***GSLB Virtual Server Name**	Enter the name of the virtual server.
***Domain Name**	Enter the domain name.
Load Balancing Method	Select a load balancing method from the dropdown list.
Persistence Type	Select a persistence type from the dropdown list.
***Service Type**	Select a service type from the dropdown list.

7. Enter or select the field information in the **Service or Service Group Details** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Service or Service Group Details Section - Field and Description Table

Field	Description
***Type**	Select **Service** or **Service Group**.
***Service Name**	Enter a service name.
***Service Protocol**	Select service protocol from the dropdown list.
***Service IP**	Enter service IP address.
***Service Port**	Enter service port no.
***Service List**	Click Add () button to add the service(s) to the Service List.

8. Enter or select the field information in the **Monitor** section.

Monitor Section - Filed and Description Table

Field	Description
***Monitor Choice**	By default, No option is selected. Select Yes to choose the health monitors.

9. Click the **Submit** button.
   The Confirmation popup opens.

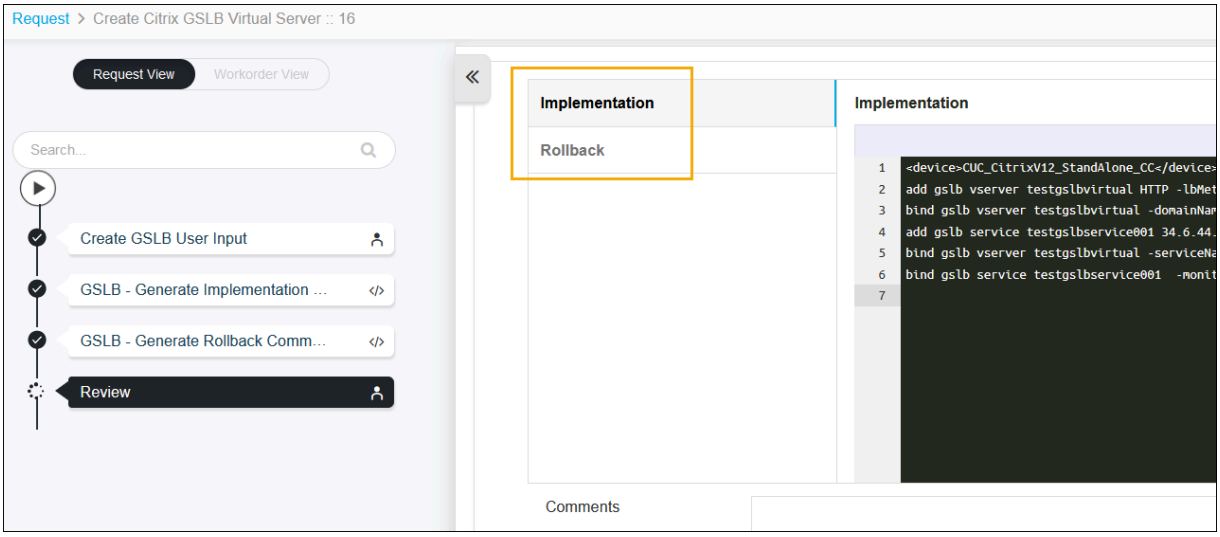
> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

10. Click **Ok** to submit the form.
    The validation starts automatically and reaches the Review stage.

> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

11. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



12. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
13. After the review, click the **Implement** button.
    The Confirmation popup opens.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

14. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
15. The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Delete Citrix GSLB Virtual Server

This workflow deletes a GSLB virtual server from Citrix device by fetching all the GSLB virtual server from the device.

To run this workflow:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

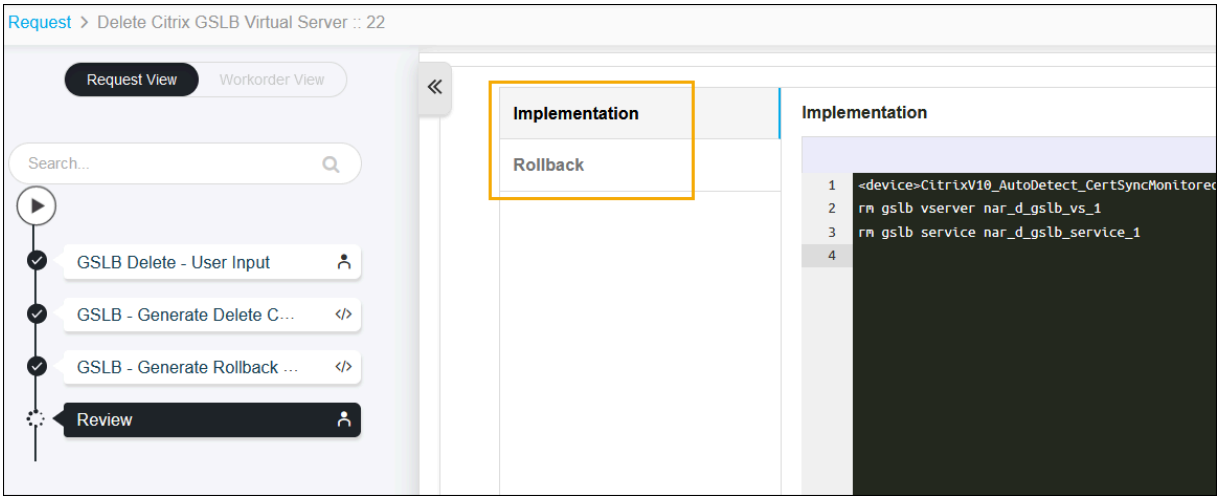
1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix GSLB** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete Citrix GSLB Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click Run.
   The **Request > Delete Citrix GSLB Virtual Server :: FormBuilder** page is displayed.
5. Fetch and select a device, and virtual server details in the **Citrix Device and Virtual Service Details** section.
6. Click the **Submit** button.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

7. Click **Ok** to submit the form.
   The validation starts automatically and reaches the Review stage.

> **Note:**
> - If there is any failure at this stage, start a new request by providing the correct data.
> - Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

9. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
10. After the review, click the **Impalement** button.
    The Confirmation popup opens.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

11. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
12. The workflow is completed and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Modify Citrix GSLB Virtual Server

This workflow modifies a GSLB virtual server on a Citrix device by fetching the device details.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix GSLB** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Modify Citrix GSLB Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Modify Citrix GSLB Virtual Server :: FormBuilder** page is displayed.
5. Fetch and select the details of device, virtual server, and service or service group.
6. Modify the necessary details of GSLB virtual server.
7. Click the **Submit** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button. The form will be saved as Cancelled request under **Request** > **My Request**.

8. Click **Ok** to submit the form.
   The validation starts automatically and reaches the Review stage.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

9. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:
10. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
11. After the review, click the **Implementation** button.
    The Confirmation popup opens.

Note:
To stop running the workflow creation, click **Cancel**.

12. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
13. The workflow is created and the email is triggered to the configured email IDs.

Note:
The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Citrix Context Switch

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Create Citrix CS Virtual Server

This workflow creates a context switch virtual server on Citrix devices.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix Context Switch** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Create Citrix CS Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Create Citrix CS Virtual Server :: FormBuilder** page is displayed.
5. Select the field information in the **Citrix Device Details** section.

Citrix Device Details Section - Field and Description

Field	Description
Data center	Displays the list of datacenters of the devices, which are created in the Device Inventory. Select the datacenter of a device from the drop-down option for which this request is to be created. For the devices, which are created without a datacenter in the Device Inventory, select the datacenter as **None**.
***Citric LTM Device**	Displays the list of devices associated with the selected datacenter. If the datacenter is selected as **None**, the devices that are created without datacenter details are listed. Select the desired device from the drop-down option.

6. Enter or select the field information in the **Citrix CS VIP Details** section.

Citrix CS VIP Details Section - Field and Description

Field	Description
***Virtual Server Name**	Enter the name of the virtual server.
***Protocol**	Select a protocol from the dropdown list.
IP Address Type	Select the IP address type.
***Virtual Server IP**	Enter the virtual server IP address.
***Virtual Server Port**	Enter the virtual server port no.

7. Enter or select the field information in the **Context switch policy binding** section.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Context switch policy binding Section - Field and Description Table

Field	Description
Select Policy	Click the **Fetch Existing Context Switch Policy** button, and then select a policy from the dropdown list.
Priority	By default. the priority is set to 120. If needed, you can update the value. It is mandatory for all the CS policy bindings. **Note:** Priority cannot be specified for URL-based context switching policy.
Target Load Balancing Virtual Server	Click the retrieve button and select a virtual server from the dropdown list.
CS Policy Binding	Click Add () button to add the policy to theCS Policy Binding.
Default Virtual Server Bound	Click the **Fetch available LB Servers** button, and then select a Default Virtual Server Bound from the dropdown list.

8. Enter or select the field information in the **Profiles** section of Form Input.

Profiles Section - Field and Description Table

Field	Description
Net Profile	Click the **Fetch Profiles** button, and then select a net profile from the dropdown list.
TCP Profile	Select a TCP profile from the dropdown list.
HTTP Profile	Select a HTTP profile from the dropdown list.

9. Click the **Submit** button.
   The Confirmation popup opens.

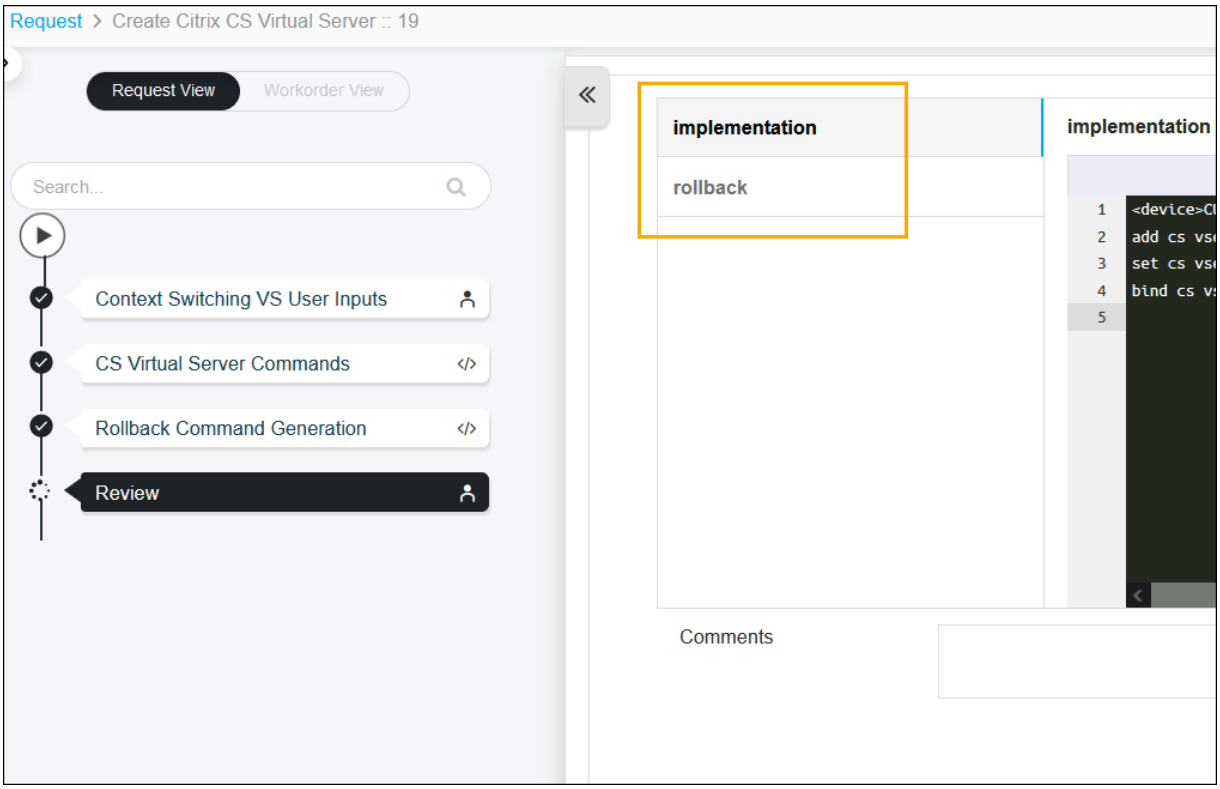
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
- If you want to cancel this form, click the **Cancel** button.

10. Click **Ok** to submit the form.

    The validation starts automatically and reaches the Review stage.



Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

11. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:
12. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
13. After the review, click the **Implement** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The Confirmation popup opens.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

14. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
15. The workflow is created and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Delete Citrix CS Virtual Server

This workflow deletes a context switch virtual server from Citrix device by fetching all the context switch virtual server from the device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix Context Switch** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Delete Citrix CS Virtual Server** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

   .
   The **Request > Delete Citrix CS Virtual Server :: FormBuilder** page is displayed.
5. Fetch and select a device, and virtual server details in the **Device and Virtual Service Details** section.
6. Click the **Submit** button.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **My Request**.
> - If you want to cancel this form, click the **Cancel** button.

7. Click **Ok** to submit the form.

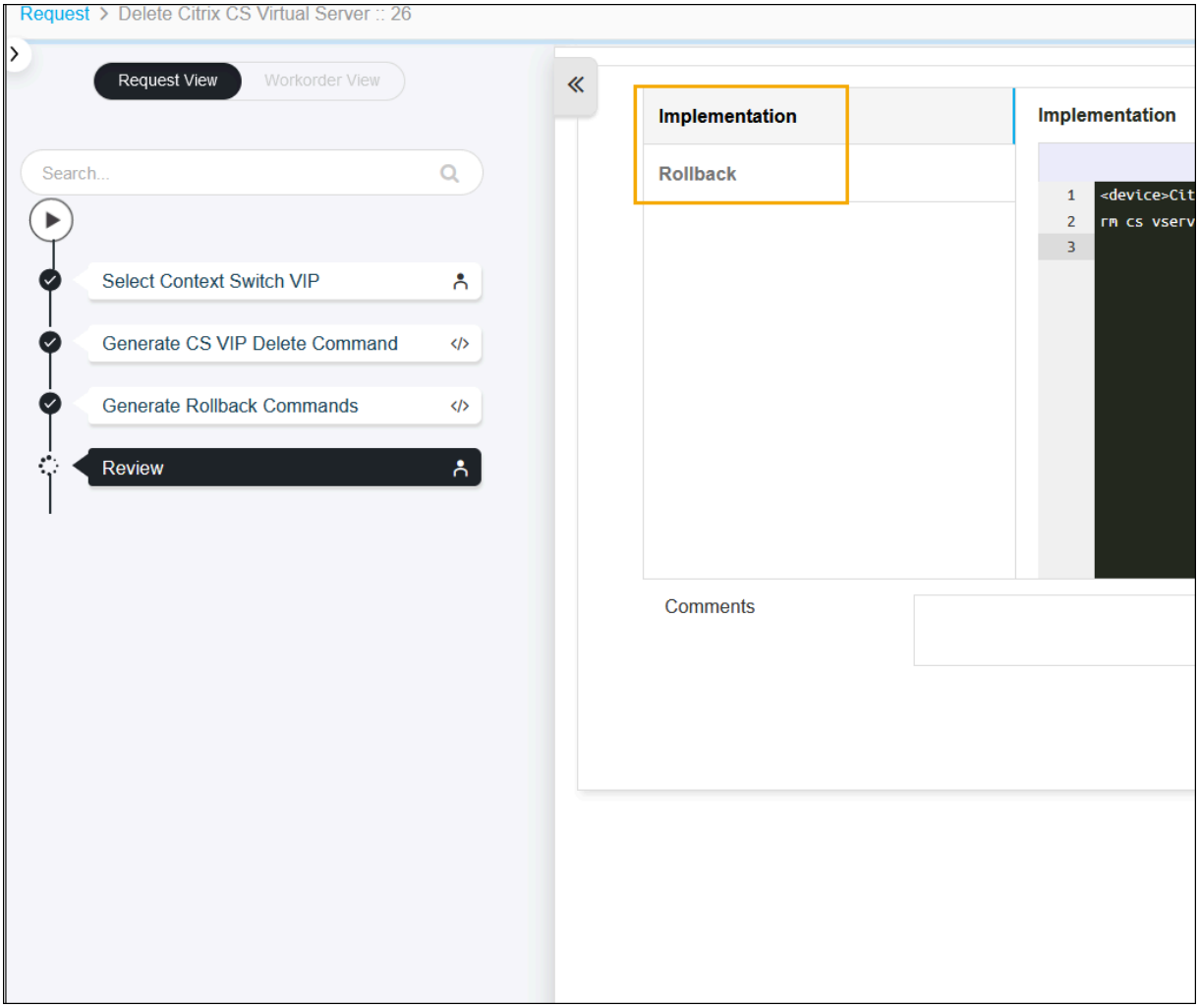
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The validation starts automatically and reaches the Review stage.

Note:

- If there is any failure at this stage, start a new request by providing the correct data.
- Alternatively, select the failed request and clone it. Update the correct input data in the form input, and submit the workflow again.

8. Review the input data under the **implementation**, **rollback**, and **postvalidation** tabs:



9. (Optional) If you need to update any data at this stage, you can do so by clicking the right-side of the mouse on the data and selecting the desired option.
10. After the review, click the **Impalement** button.
    The Confirmation popup opens.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

> **Note:**
> To stop running the workflow creation, click **Cancel**.

11. Click **Ok** to continue the workflow creation.
    It takes a while to complete the request.
12. The workflow is completed and the email is triggered to the configured email IDs.

> **Note:**
> The stages of the request are shown in the left-side of the screen. To view a particular stage of the request, click the respective stage.

Citrix NetScaler System

Citrix NetScaler Golden Config Compliance

This workflow validates a device against a compliance policy. A Standard Compliance policy "Citrix_Default_Compliance" with Default Parameters and Sample Values has been provided under the Collection. View this policy by navigating **menu** > **Collection**. Either you can select the default parameters or add custom commands to perform a compliance check and remediate. If the custom option is chosen, addition of rollback command is mandatory.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Citrix NetScaler System** category on the **View/Run Workflows** page.
3. On the popup window, hover the mouse over the **Citrix NetScaler Golden Config Compliance** workflow.
   The **Run** and **Schedule** buttons are displayed.
4. Click 

.
   The **Request > Citrix NetScaler Golden Config Compliance :: FormBuilder** page is displayed.
5. In the **Device Details** section, select a device, and then a compliance policy .
6. In the **Compliance Parameter** table, select or enter the following details:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Compliance Parameter Section - Field and Description Table

Field	Description
***Config/ Parameter Type**	You can select **Custom** or Default Config/Parameter type. For Custom Compliance policy, the Config/Parameter type is considered as **Custom** by default even if it is selected as **Default**.
***Config/ Parameter Name**	Enter the config/parameter name.
***Value**	Enter the expected value in the device for the parameter.

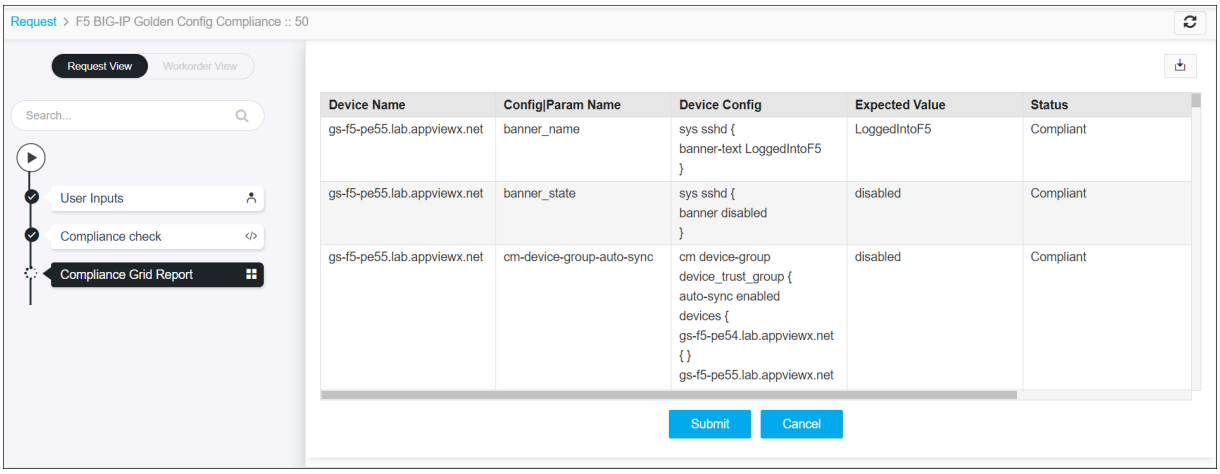
7. To add a new parameter to the Compliance Parameter Table, click Add () button After adding the parameter, you can manage (update/delete) it.
8. Click the **Submit** button.
   The Confirmation popup opens.

> **Note:**
> - If you want to save this form to edit it later, click the **Save Draft** button, and then click **Ok** in the Confirmation popup window. The form will be saved as Open request under **Request** > **All**.
> - If you want to cancel this form, click the **Cancel** button.

9. Click **Ok** to submit the form.
10. The Compliance Check runs automatically and generates **Compliance Grid Report**.
11. From the Compliance Grid Report, you get to know if the device parameters are Compliant or Non-Compliant.

    You can download the **Compliance Grid Report** in the `.csv` or `.xlsx` format by clicking the Download () button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



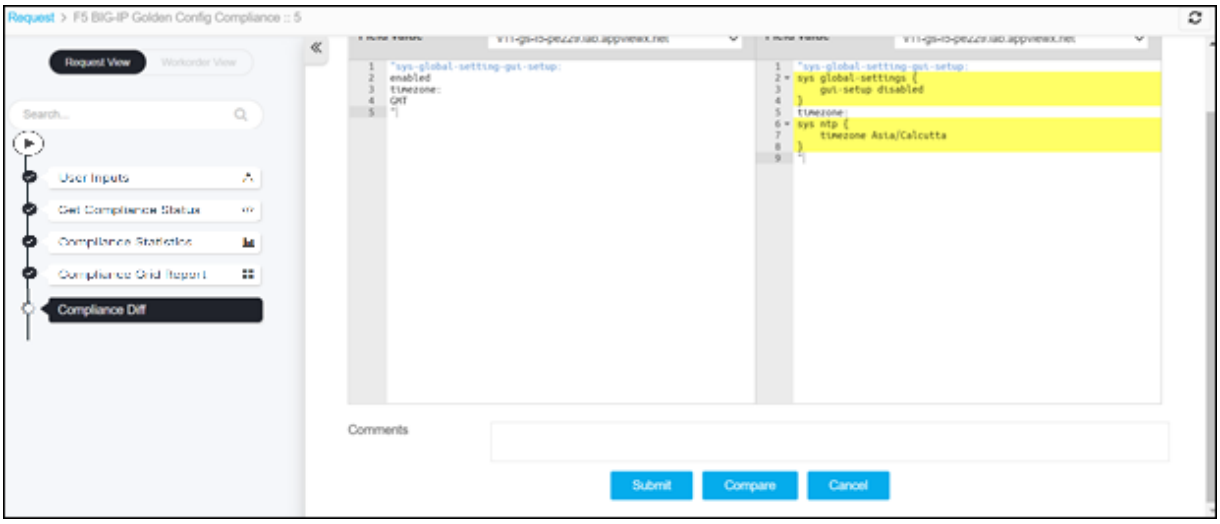
The summary of the Compliance Grid Report are:

- The first column displays the device name.
- The other columns display Config/Parameter names.
- The report has two body rows. They represent the following:
- First Row - the status of the Parameter in the device.
- Second Row - the actual value of the parameter in the device.

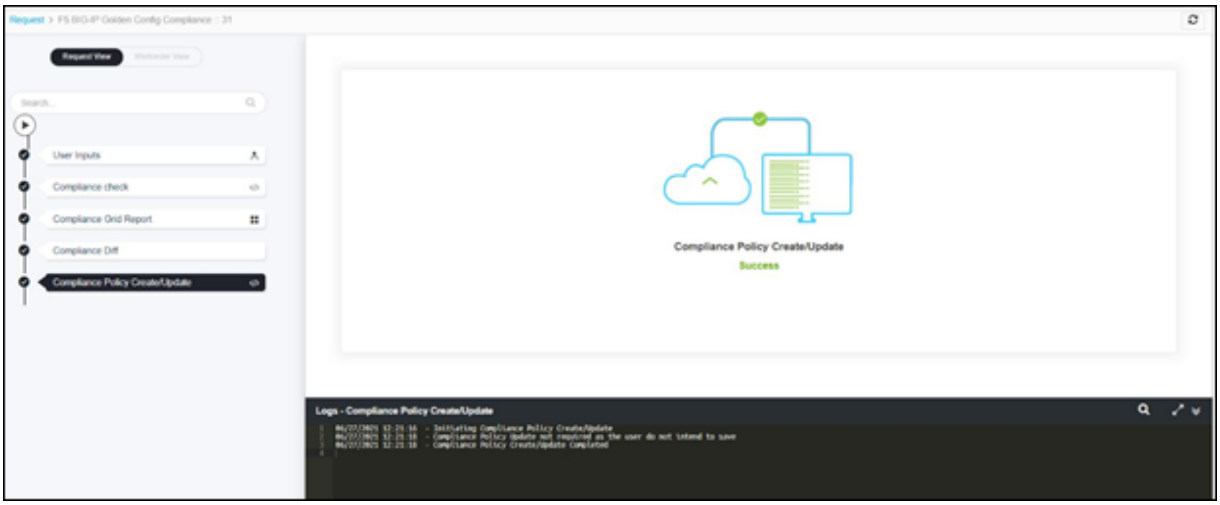
12. Click the **Submit** button, and then click **Ok** in the confirmation pop-up.
    The **Compliance Diff check** runs.
13. The **Compliance Diff check** displays the details of non-compliant parameters for each device.
14. The details are:
    - Parameter name and the value in the selected policy – shown in the left side of the diff Palate-Golden Config.
    - Parameter name and the value in the selected device - shown in the right side of the diff Palate-actual config.

Note:
To view the compliance difference for the other devices, select a device from the field value drop-down list in the left and/or right side of the diff Palate.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



15. Click the **Submit** button, and then click **Ok** in the confirmation pop-up.
16. The **Compliance Policy Create/Update** runs automatically:

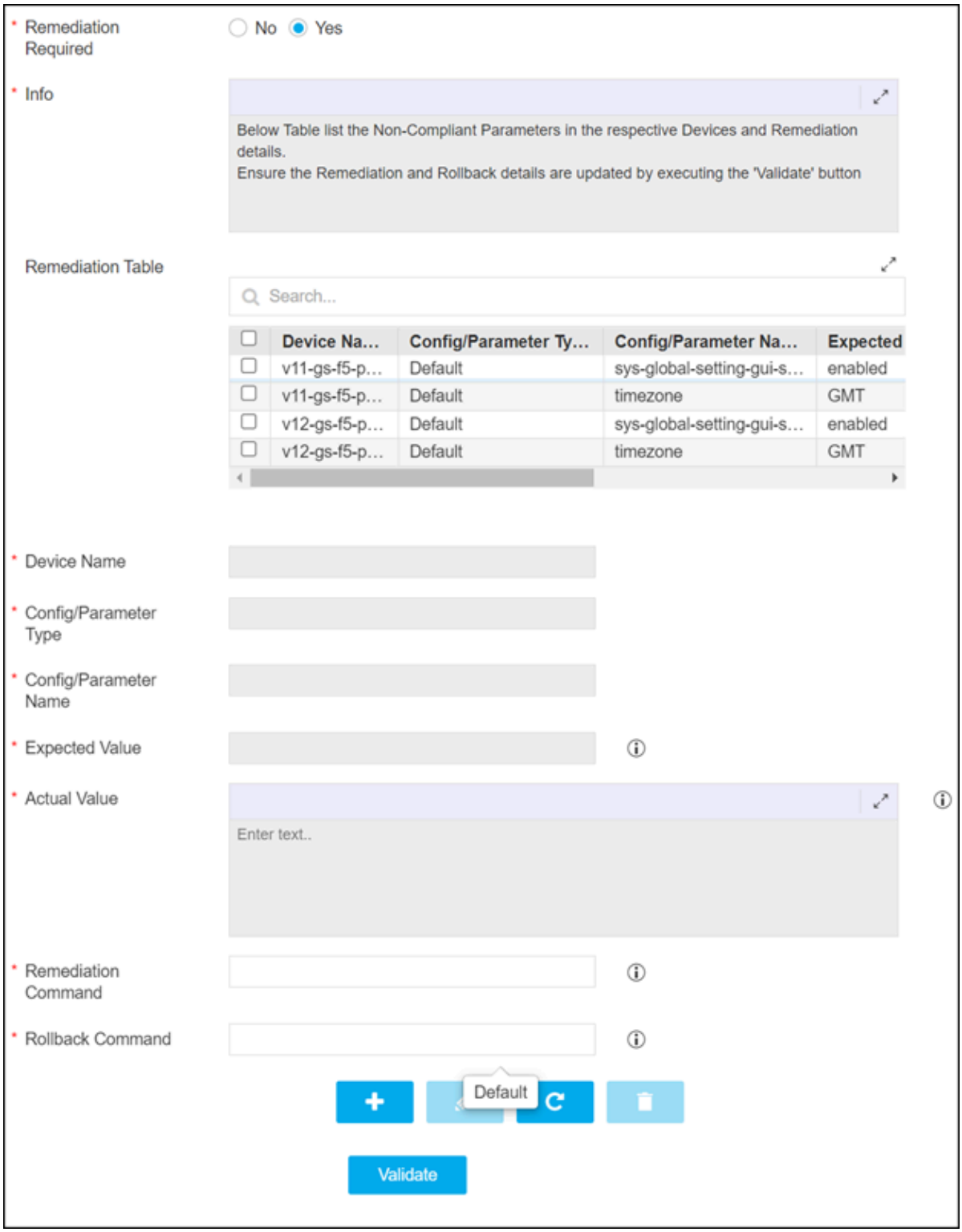


> **Note:**
> At this phase, the policy is created/updated if you have selected the option Create New Compliance, SaveAs – Yes, Update Policy -Yes in the form input.

17. Remediation Review: The Page displays a list of the Non-Compliant Parameters. By default, two options are displayed: No, Yes.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

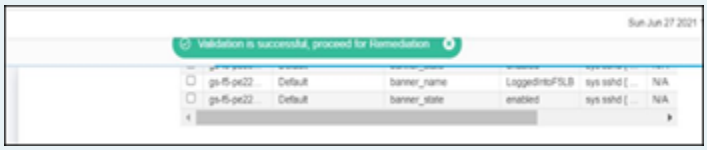
- **No** – By default this option is selected. When this option is selected and you **Submit**, the remediation will not be applied to the non-compliant parameters, the workflow ends.
- **Yes** – When you select **Yes**, the remediation table is displayed for the non-compliant parameters.



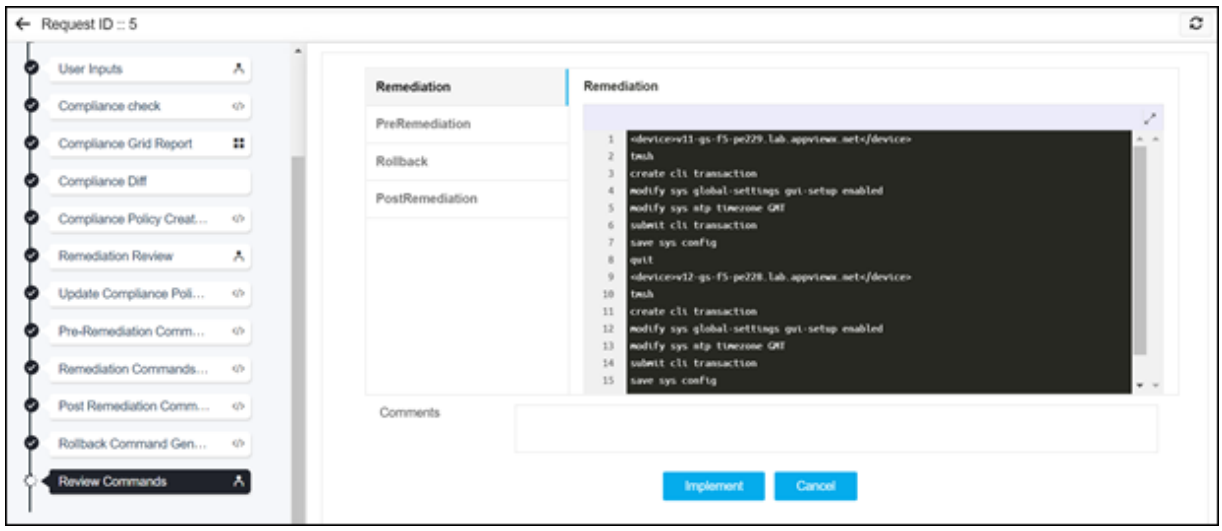
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Note:

- In case any custom parameter is part of this remediation table, then update the rollback command in the **Rollback Command** field.
- For default parameters, the remediation or rollback command is not required to be provided.
- If the rollback command is not updated for the custom parameter, you do not get the rollback option in case to revert the device parameter to the older state.
- Once the remediation command and rollback command are updated to the device, click the Update button.
- Validate the remediation and rollback details by clicking the Validate button until the message Validation is Successful is seen.



18. Click the **Submit** button, and then click **Ok** in the confirmation pop-up.
    The update of remediation and rollback commands run automatically and generates Review Commands to review and implement the changes.



19. If required, review the commands that are about to be executed in the F5 load balances in the following tabs:
    - **Remediation**
    - **PreRemediation** – List the values of the parameter, prior Remediation.
    - **Rollback** – Revert the parameter's value to the previous state of Remediation.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- **PostRemediation** - List the values of the parameter, post Remediation.



RSeries

RSeries Device Restore

This workflow allows to restore RSeries device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Hover the mouse over the **RSeries Device Restore** workflow on the **View/Run Workflows** page.
3. Click .
   The **Request > RSeries Device Restore :: FormBuilder** page is displayed.
4. In the **Device Details** table, select or enter the following details:

Device Details Section - Field and Description Table

Field	Description
***Host Device Name**	Select a host from the dropdown list.
***Host Backups**	Select a host backup from the dropdown list.
***Do You Want To Restore RSeries Tenants?**	By default, **No** is selected. To restore RSeries tenants, click **Yes**.
Primary Key	Enter the primary key. It must contain minimum six characters. The primary key is the key that is used to authenticate and secure the restoration process of the RSeries device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Salt	Enter a string of random characters used in combination with the primary key to create a unique hash. It must contain minimum six characters.
* - *Mandatory fields*	

5. Click the **Submit** button.
   The Confirmation popup opens.

> **Note:**
> To cancel this form, click the **Cancel** button.

6. Click **Ok** to submit the form.
   The workflow runs automatically and restore the host succesfully. The restored host will be found in the logs from **ADC+** > **ALERTS & LOGS** > **Logs**.

> **Note:**
> If there is any failure at this stage, start a new request by providing the correct data.

RSeries Device Backup

This workflow allows to backup RSeries device.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Hover the mouse over the **RSeries Device Backup** workflow on the **View/Run Workflows** page.
3. Click .
   The **Request > RSeries Device Backup :: FormBuilder** page is displayed.
4. In the **Device Details** table, select or enter the following details:

Device Details Section - Field and Description Table

Field	Description
***Host Device Name**	Select a host from the dropdown list.
***Do You Want To Backup RSeries Tenants?**	By default, **No** is selected. To backup RSeries tenants, click **Yes**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Field	Description
Primary Key	Enter the primary key. It must contain minimum six characters. The primary key is the key that is used to authenticate and secure the backup process of the RSeries device.
Salt	Enter a string of random characters used in combination with the primary key to create a unique hash. It must contain minimum six characters.
* - *Mandatory fields*	

5. Click the **Submit** button.
   The Confirmation popup opens.

> **Note:**
> To cancel this form, click the **Cancel** button.

6. Click **Ok** to submit the form.
   The workflow runs automatically and backup is created for the host successfully. The host backups will be managed from **ADC+** > **CONFIG MANAGEMENT** > **Backup Configuration**.

> **Note:**
> If there is any failure at this stage, start a new request by providing the correct data.

Reports

ADC Device Status Report

This workflow fetches the device status report and sends it via email if configured.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Reports** widget.
3. Hover the mouse over the **ADC Device Status Report** workflow on the **View/Run Workflows** page.
4. Click **Run**.
   The **Request > ADC Device Status Report :: FormBuilder** page is displayed.
5. In the **Device Details** table, select preferred device statuses from the Device Status dropdown.
6. Click the **Submit** button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

The Confirmation popup opens.

> **Note:**
> To cancel this form, click the **Cancel** button.

7. Click **Ok** to submit the form.
   The workflow runs automatically and fetches the report.

> **Note:**
> If there is any failure at this stage, start a new request by providing the correct data.

ADC Device Status Report

This workflow automates scheduling of the device status report and emails it if configured.

To run this workflow:

1. Go to **Menu** > **ADC+** > **AUTOMATION** > **Workflow Catalog** > **View/Run**
2. Click the **Reports** widget.
3. Hover the mouse over the **ADC Device Status Report** workflow on the **View/Run Workflows** page.
4. Click **Schedule**.
   The **ADC Device Status Report** popup window is displayed.
5. Enter the Name and Description, and then click Save & Continue.
6. Select the preferred device statuses from the **Device Status** dropdown, and then click **Save & Continue**.
7. Select the schedule frequency (once or repeat), and select the date range using the **Starts on** calendar.
8. Click the **Schedule** button.
   The workflow scheduled successfully.

Sample Workflow Creation

Sample Workflow Creation

Designing F5 BIG-IP Automation Workflow

This section covers the following procedures:

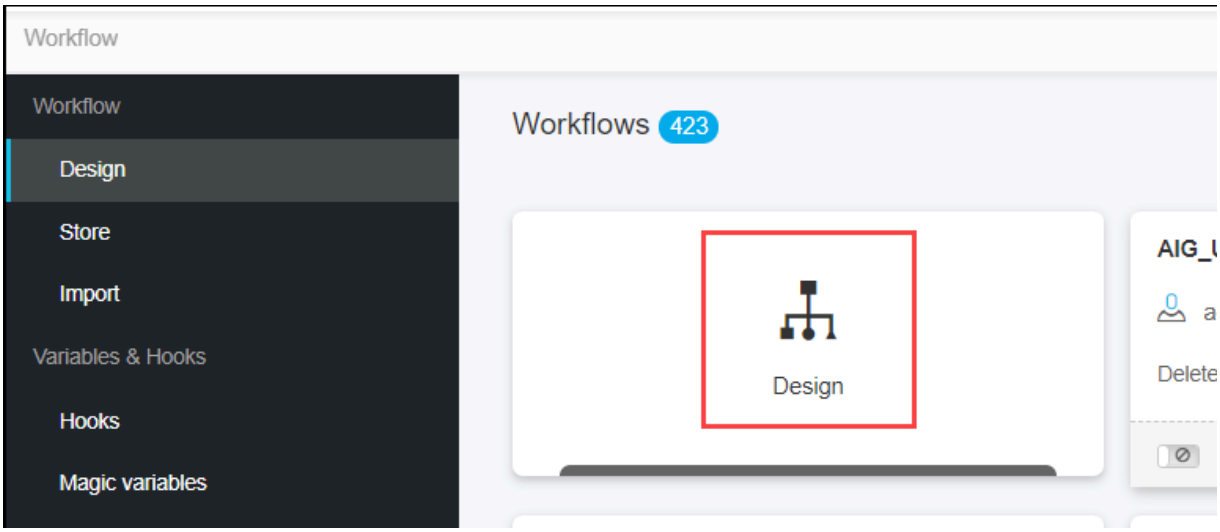
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Sample workflow creation with pool, pool members, and monitor
- Customize the variables for a task

Sample Workflow Creation with Pool, Pool Members, and Monitor

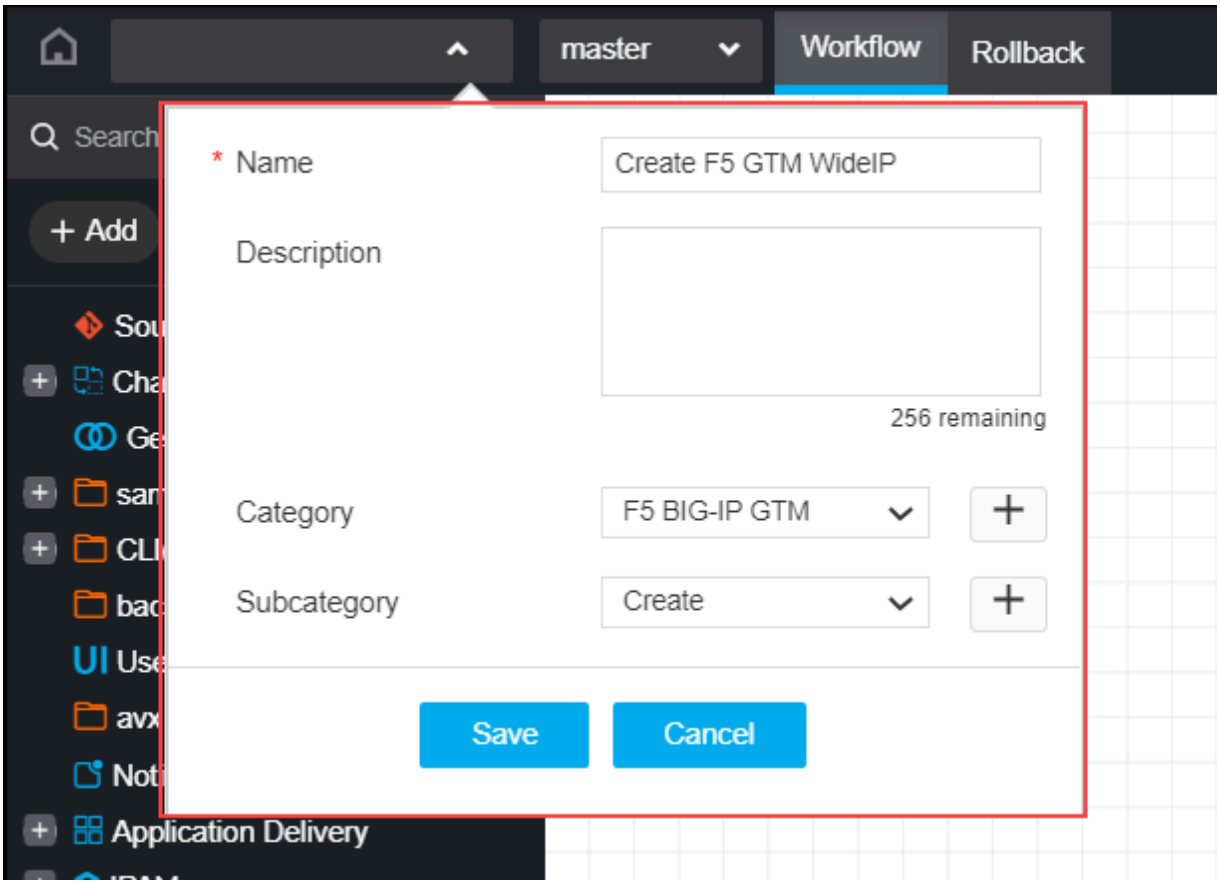
To create a WideIP with pool, pool members, and monitor,

1. Go to **Menu** > **Studio** > **Workflow**.
   The Workflows page appears.
2. Click **Design** to create a new workflow.



3. Enter/select the workflow design details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Field descriptions for the workflow design details

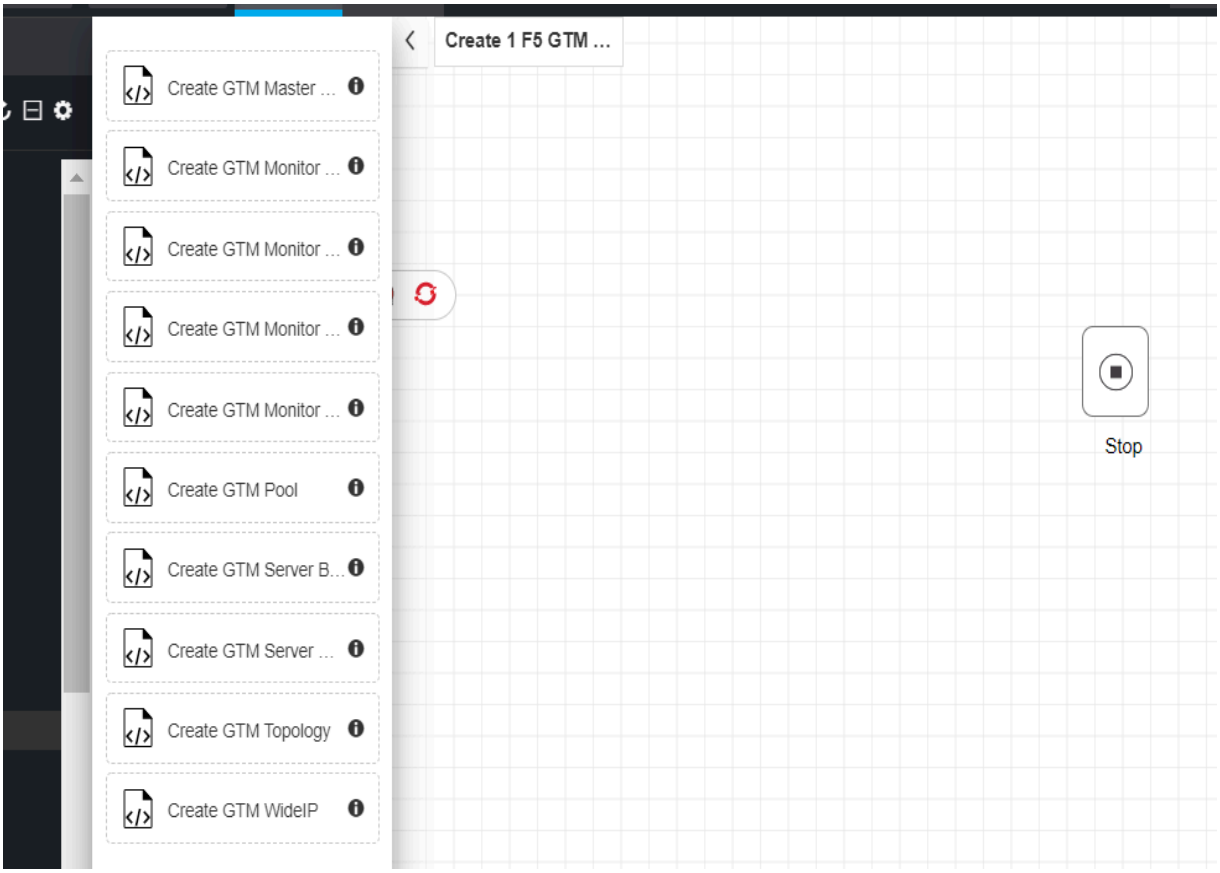
Field	Description
***Name**	Name of the workflow.
Description	Description for the workflow.
Category	Select the category as F5 BIG-IP GTM from the drop-down option.
Subcategory	Select sub-category for the workflows from the drop-down option. The possible options are: • Modify • Create • Default • Delete

4. Click the **Save** button.
   The workflow is created:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

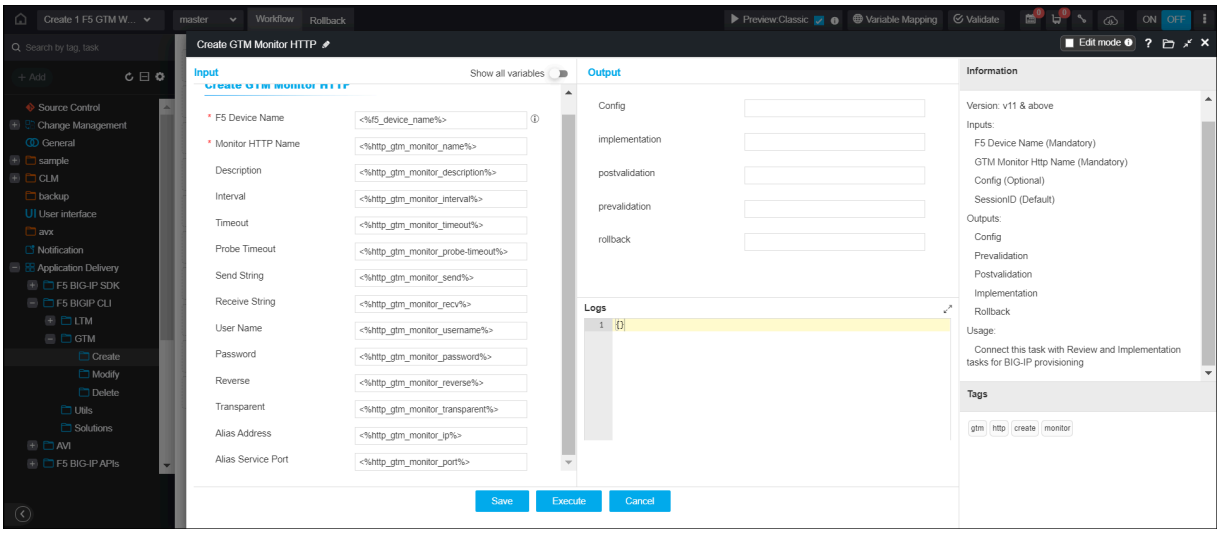


5. Go to **Application Delivery** > **F5 BIG IP CLI** > **GTM**.
6. Place any **Create GTM Monitor** task. For example, **Create GTM Monitor HTTP**.



The Input, Output, and Information data for **Create GTM Monitor** task are displayed:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

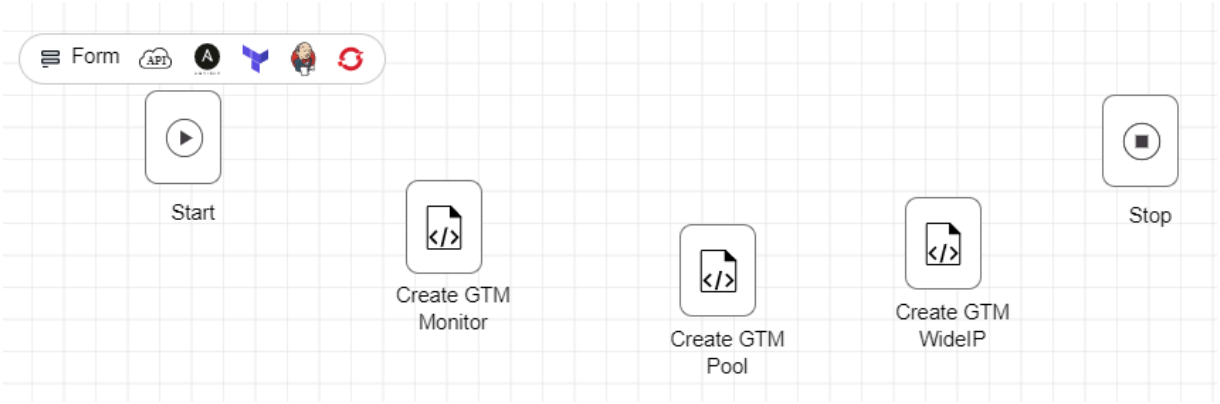


7. Click the **Save** button.

   The **Create GTM Monitor HTTP** task has been added to the workflow:

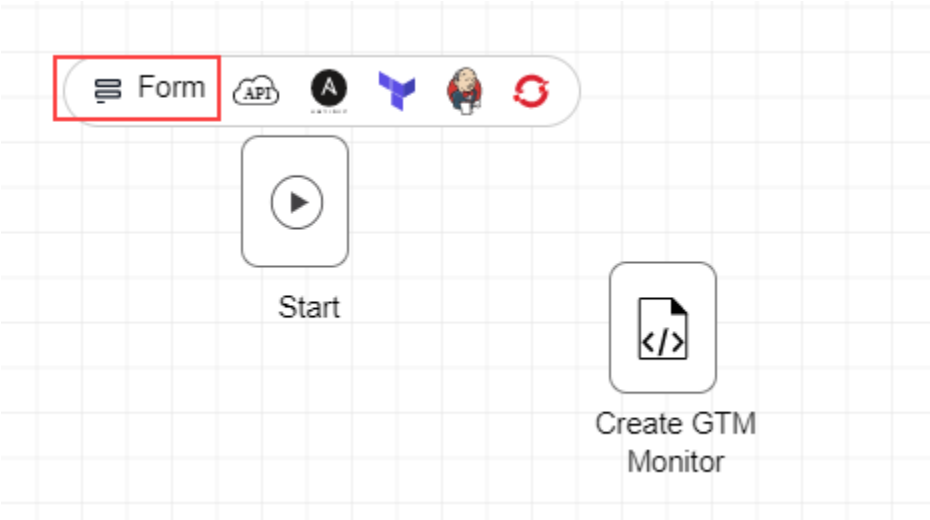


8. Similarly, place **Create GTM Pool** task and **Create GTM WideIP** task from the folder **Application Delivery** > **F5 BIG IP CLI** > **GTM**.

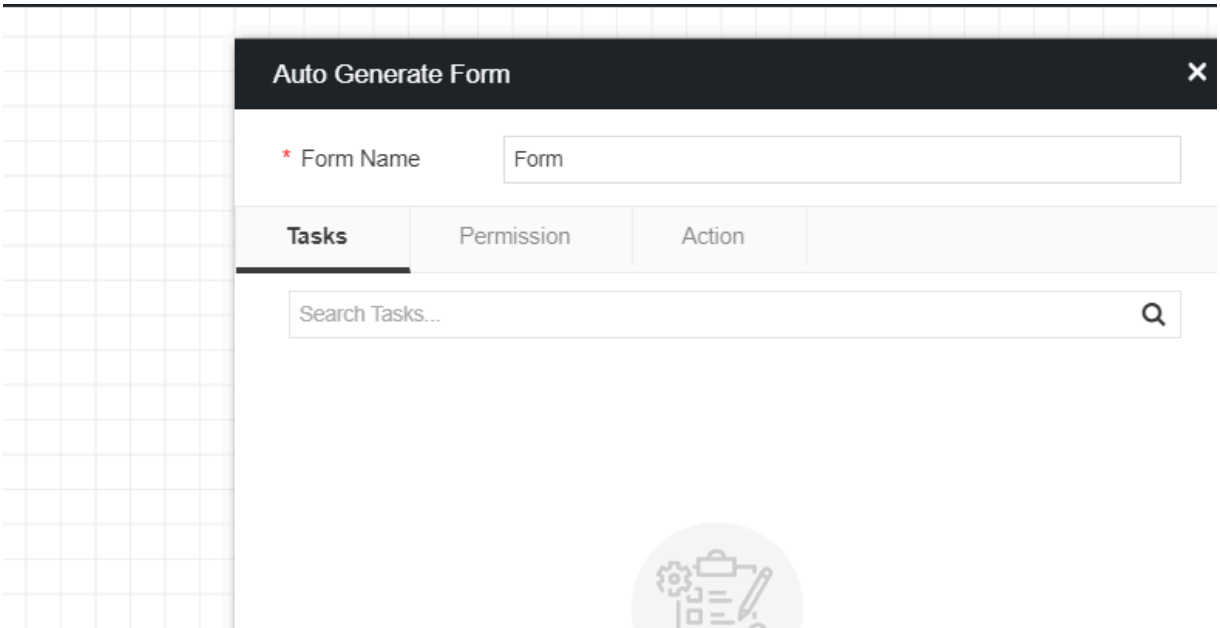


9. Click the **Form** in the Design page to auto-generate the form for the selected tasks.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

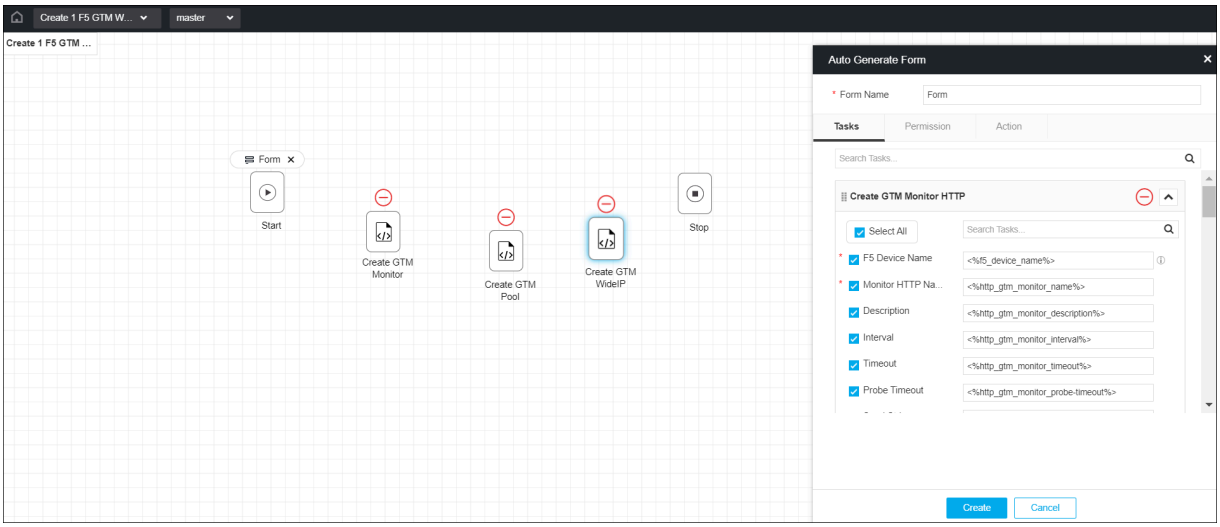


The **Auto Generated** Form appears:



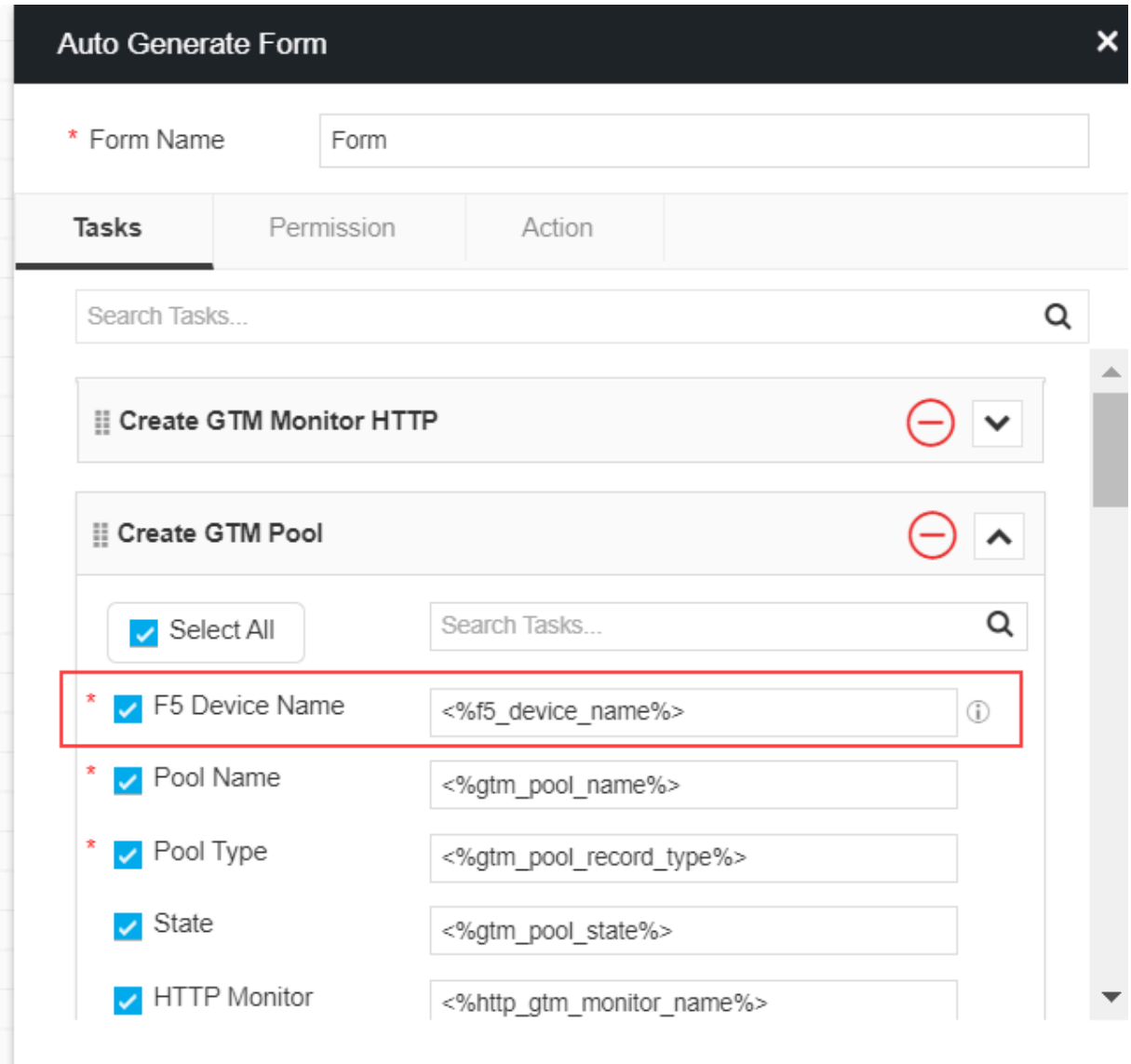
10. Click the add ( ) button of the **Create GTM Monitor HTTP** task, **Create GTM Pool** task and **Create GTM WideIP**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



The tasks are added to the Auto Generate Form:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

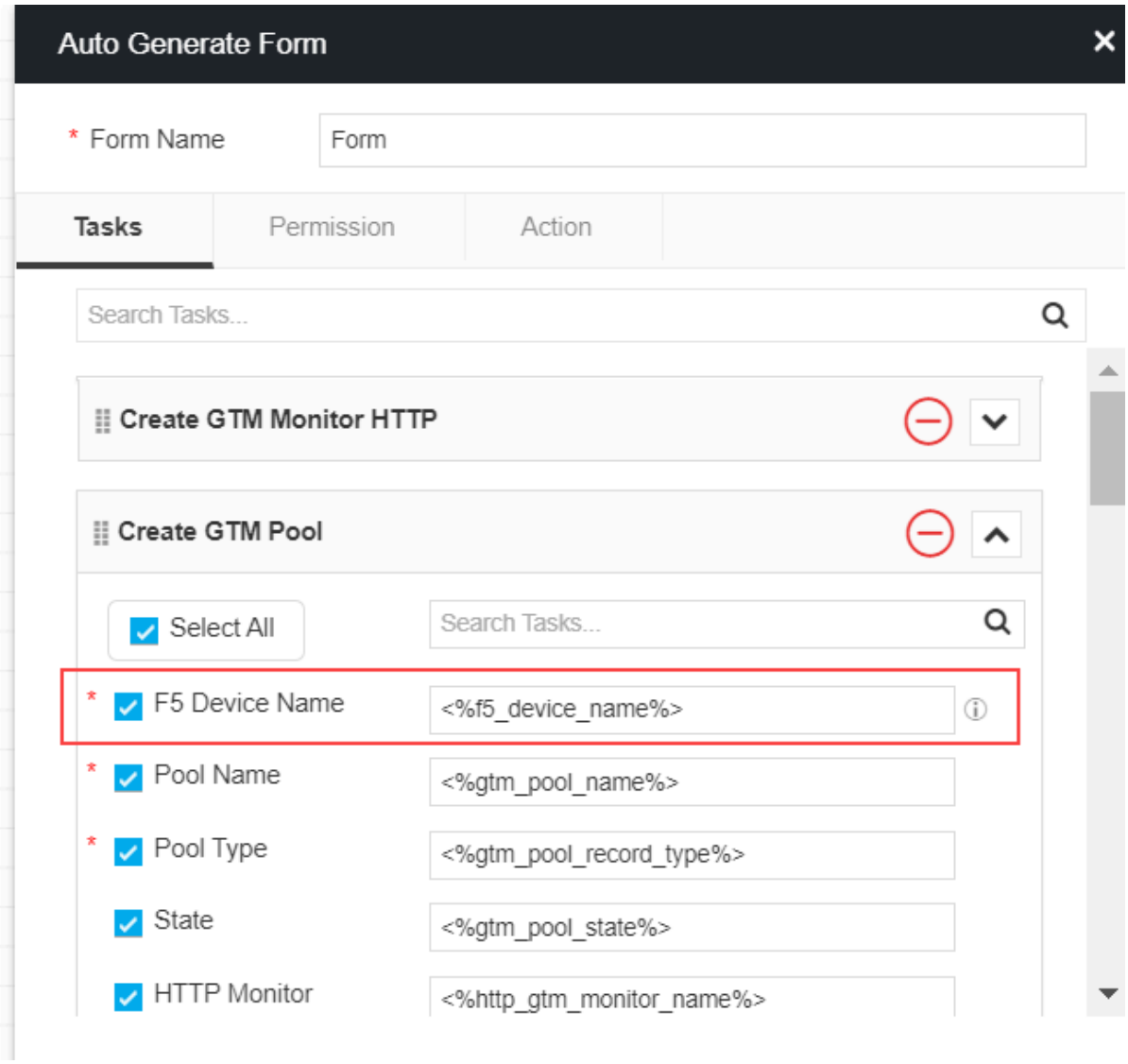


Note:
The sequence of the tasks is maintained based on the add button clicked sequence on the tasks. If you want to alter the sequence after adding the tasks to the Auto Generate Form, delete the task(s) from the Auto Generate Form and add them again.

11. Verify all the fields and select/clear the checkbox of the fields to add them to the form.
12. The following fields are not to be selected to avoid duplication being shown in the form.

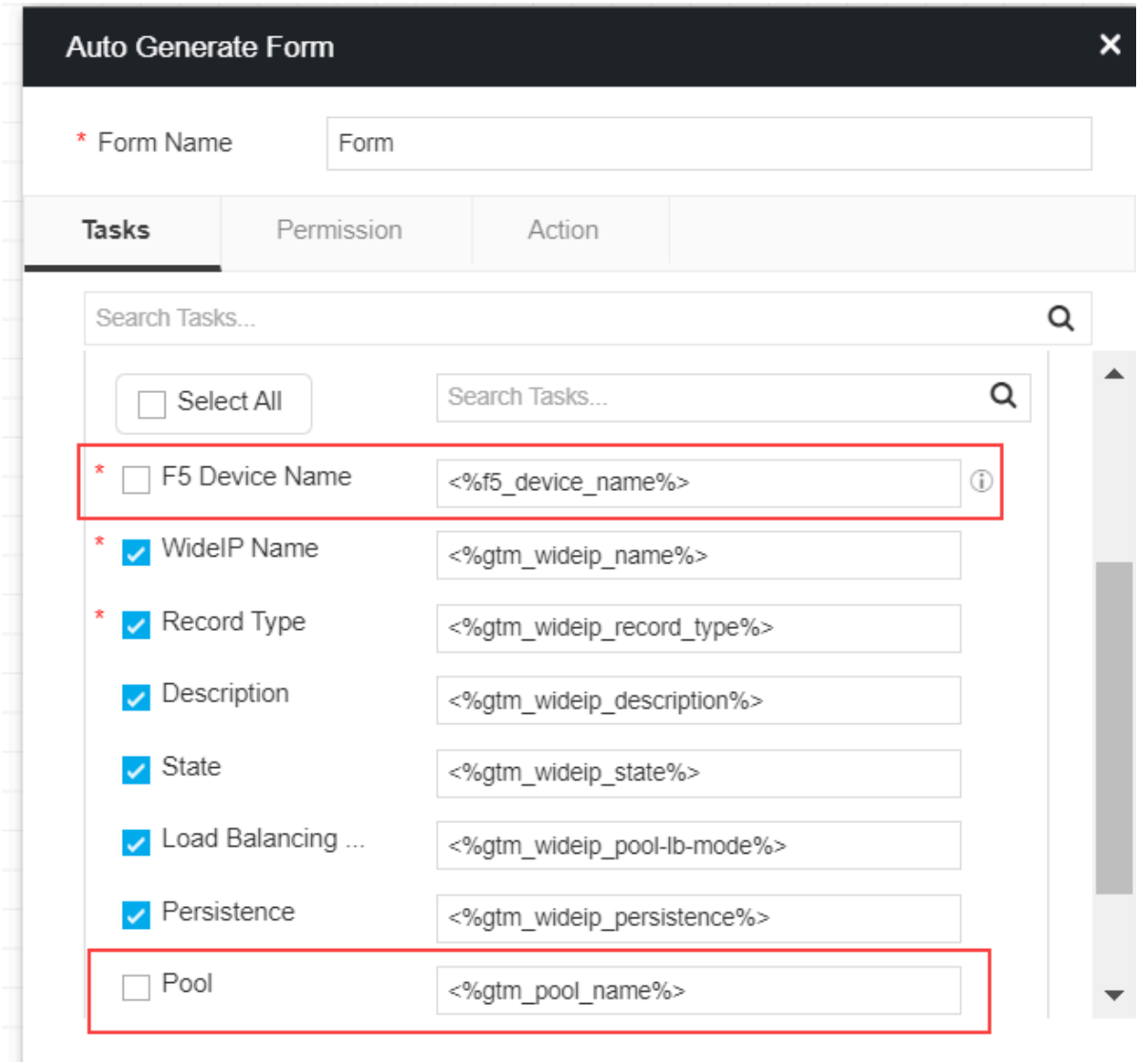
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Clear the checkbox from the **F5 Device Name** field from the **Create GTM Pool** task and **Create GTM WideIP**.



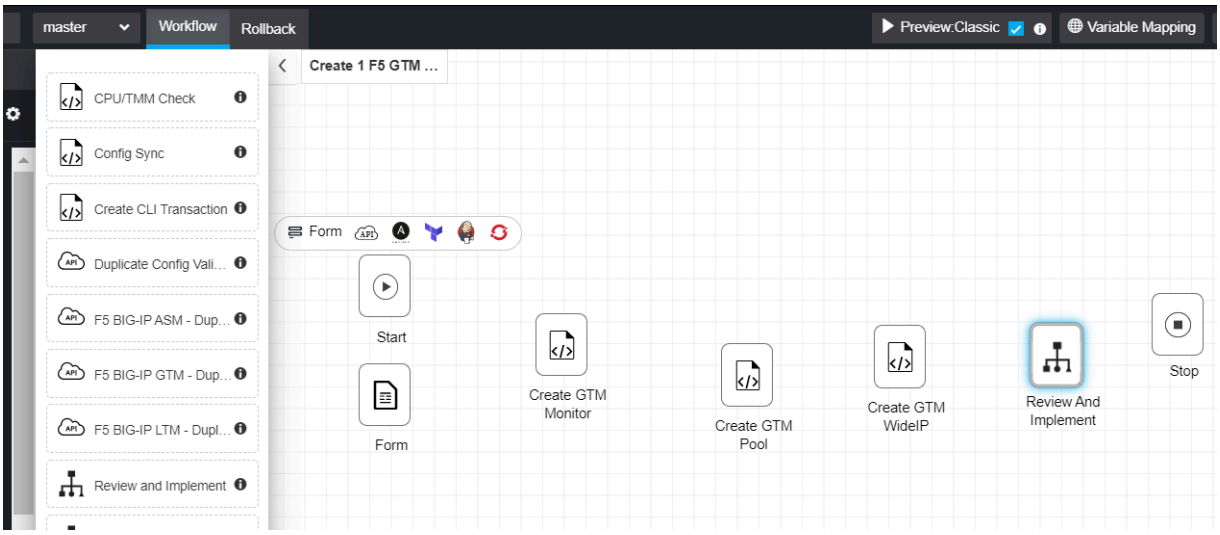
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Clear the checkbox from the **F5 Device Name** and **Pool** fields from the **Create GTM WideIP** task.

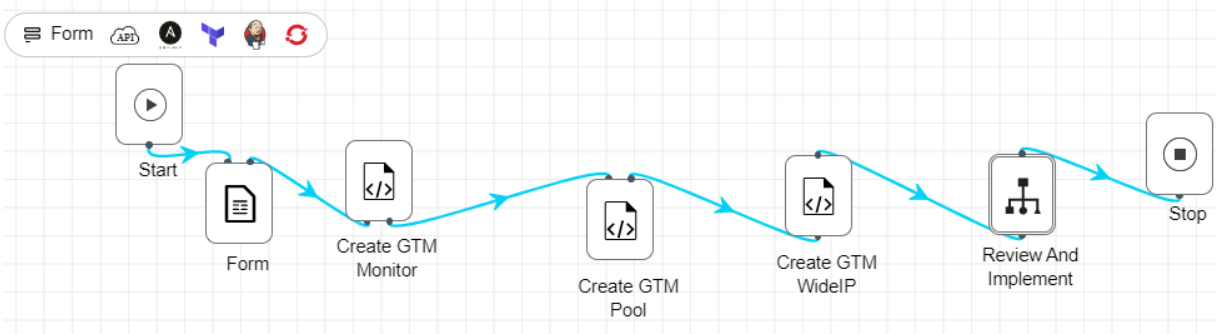


13. Click the **Create** button.
14. Go to **Application Delivery** > **F5 BIG IP CLI** > **Utils** folder, and then place **Review and Implement** subflow from the folder.

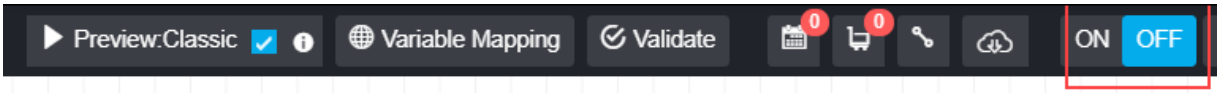
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



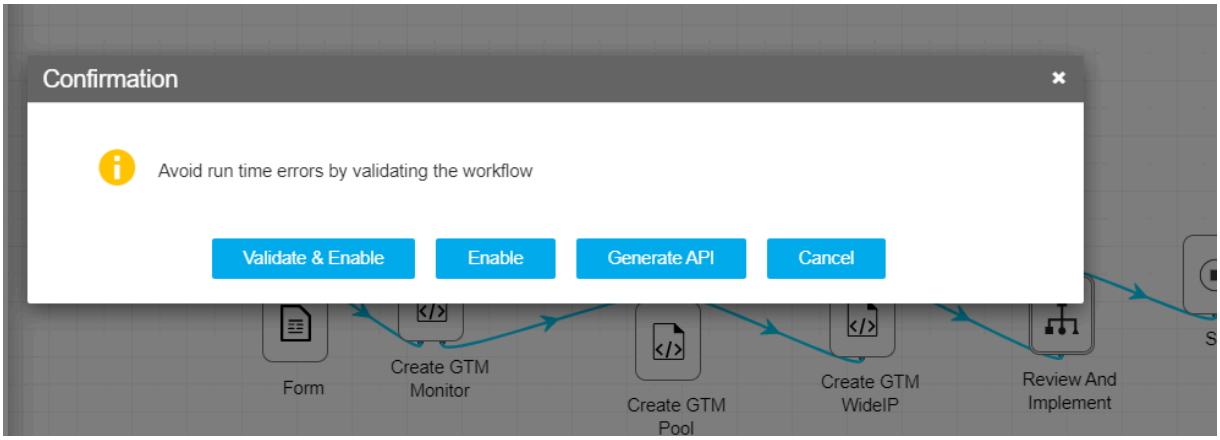
15. Connect all the tasks.



16. Enable the workflow by clicking the **ON** toggle button

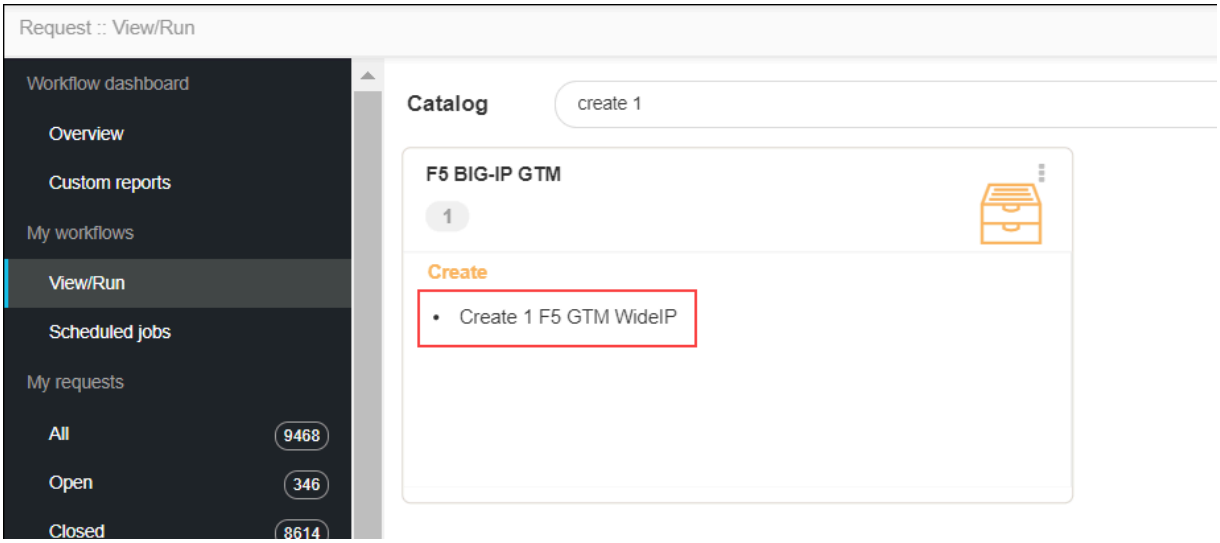


17. Click the **Validate & Enable** button.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

18. The workflow is added to the Workflow Catalog page.



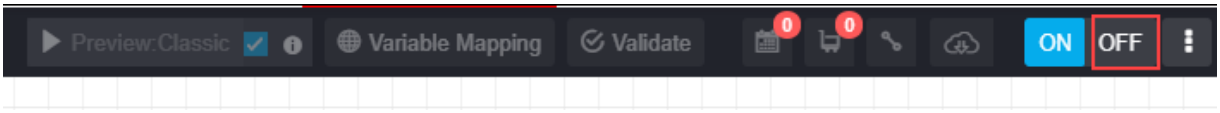
19. Run and verify the workflow.

Customize the Variables for a task

The variables can be customized for a task after creating a workflow also.

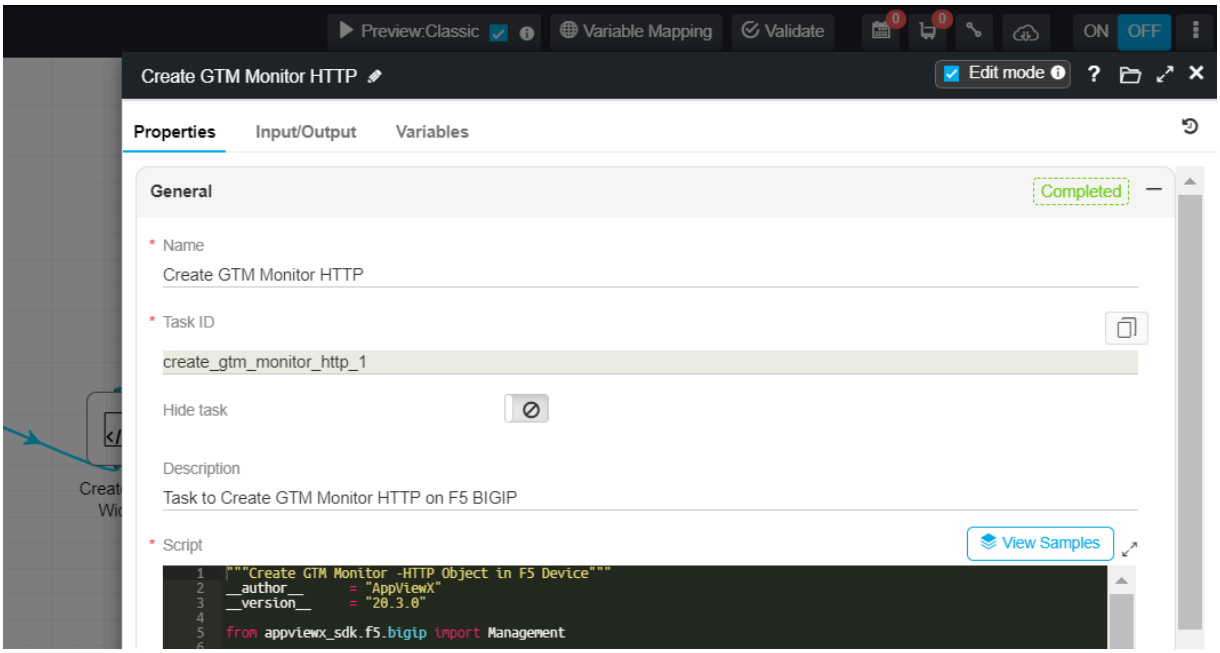
To customize the variables for a task,

1. Go to **Menu** > **Studio** > **Workflow**.
   The Workflows page appears:
2. Click the workflow for which the variables are to be customized.
3. Disable the workflow by clicking the **OFF** toggle button.

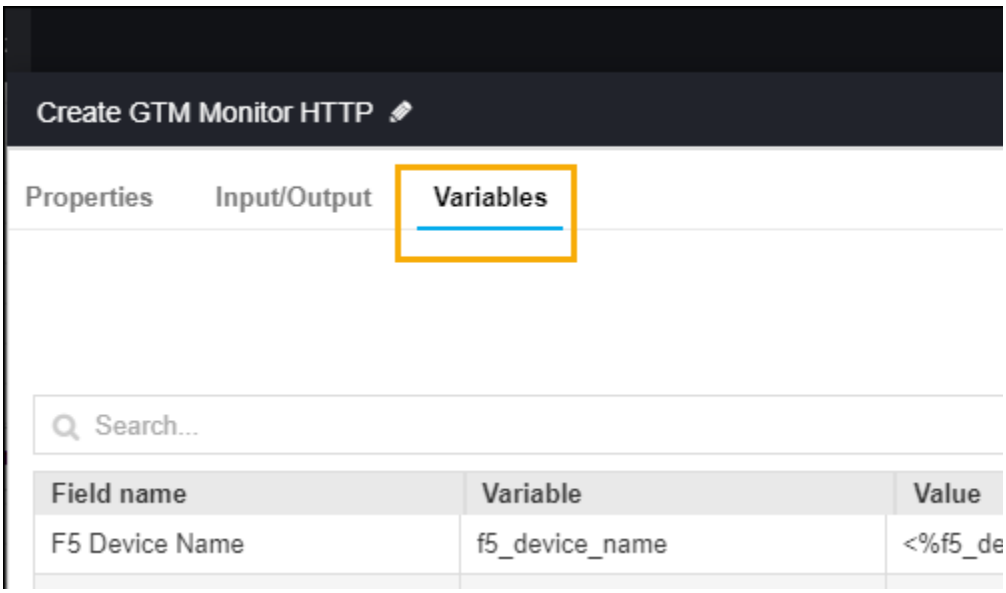


   The Confirmation popup appears.
4. Click **Yes** to confirm the disabling workflow.
5. Click twice on the desired task.
   The workflow Properties page appears:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



6. Click the **Variables** tab.



7. You can update any of the following details for the variables:
   - **Field Type** - select the field type from the drop-down option. For example, Description, Password, Dropdown, etc.
   - **Parent** - select the parent field from the drop-down option.
   - **Type** - select the type from the drop-down option. For example, String, JSON, Email, etc.
   - **Show variables** - enable or disable the variable by clicking the toggle button.
   - **Mandatory** - mark whether the field is mandatory or not by clicking the toggle button.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

8. Click the **Save** button, and then close the window.
   The updated variables are displayed in the form.

Designing Infoblox IPAM Automation Workflow

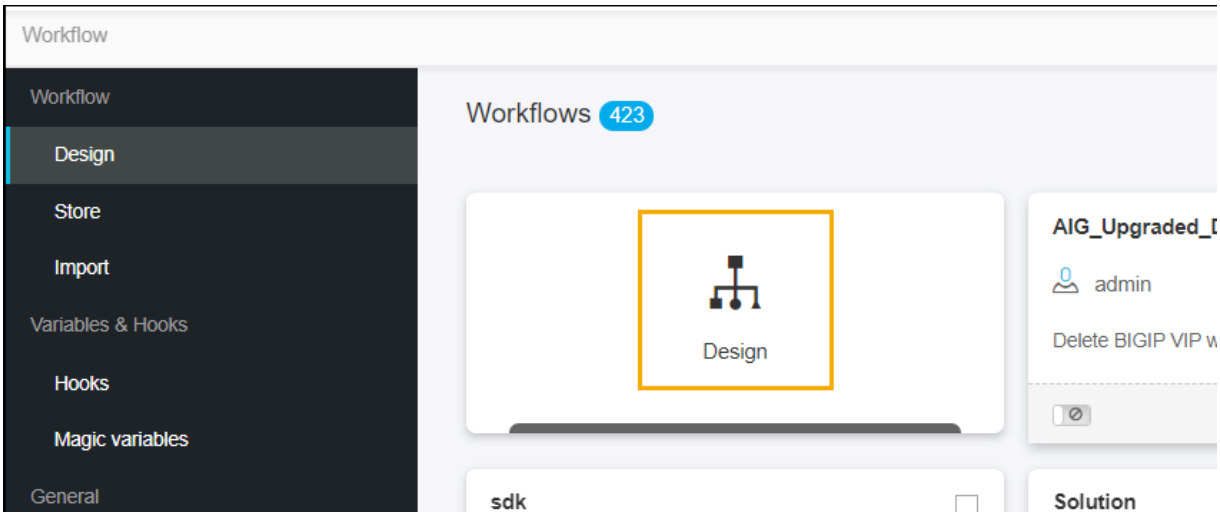
This section covers the following procedures:

- Sample workflow creation with A, CNAME, and PTR Records in Infoblox
- Customize the variables for a task

Sample Workflow Creation with A, CNAME, and PTR Records in Infoblox

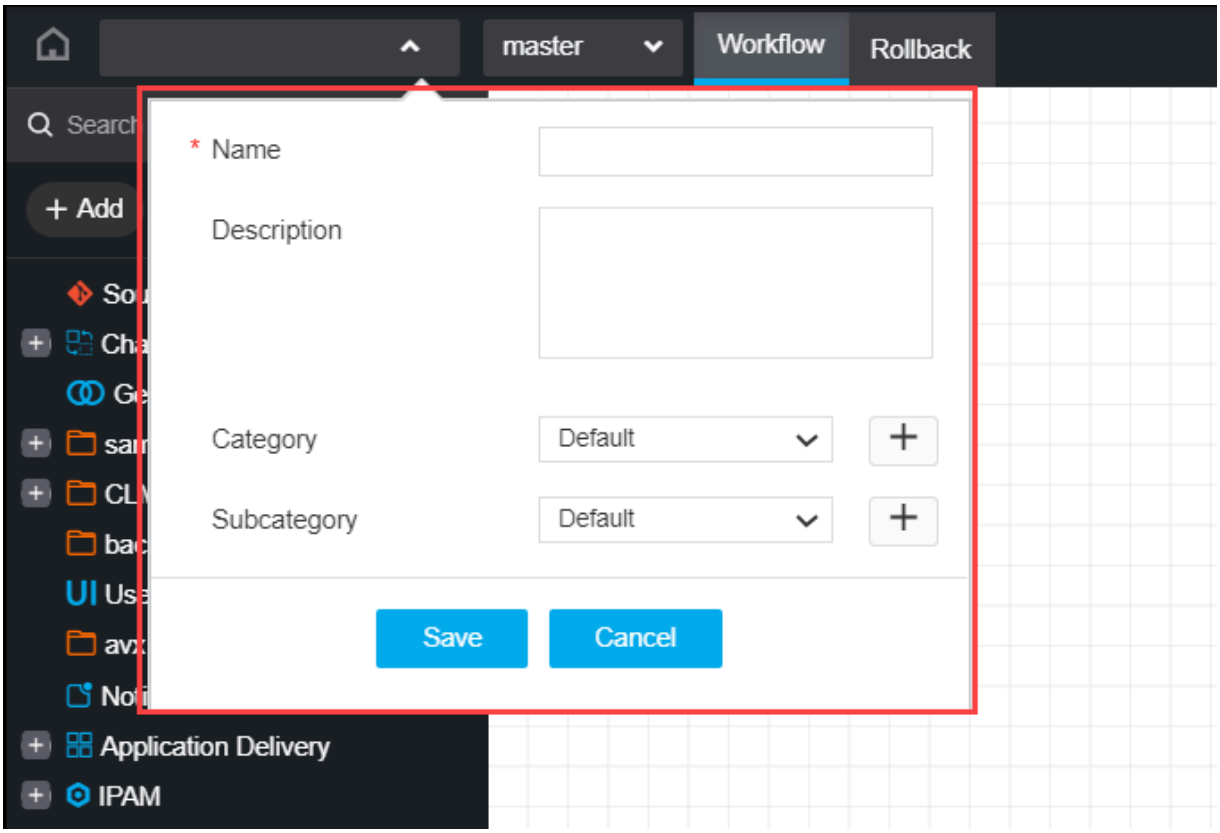
To create a sample workflow with A, CNAME, and PTR Records,

1. Go to **Menu** > **Studio** > **Workflow**.
   The Workflows page appears:
2. Click Design to create a new workflow.



3. Enter/select the workflow design details.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

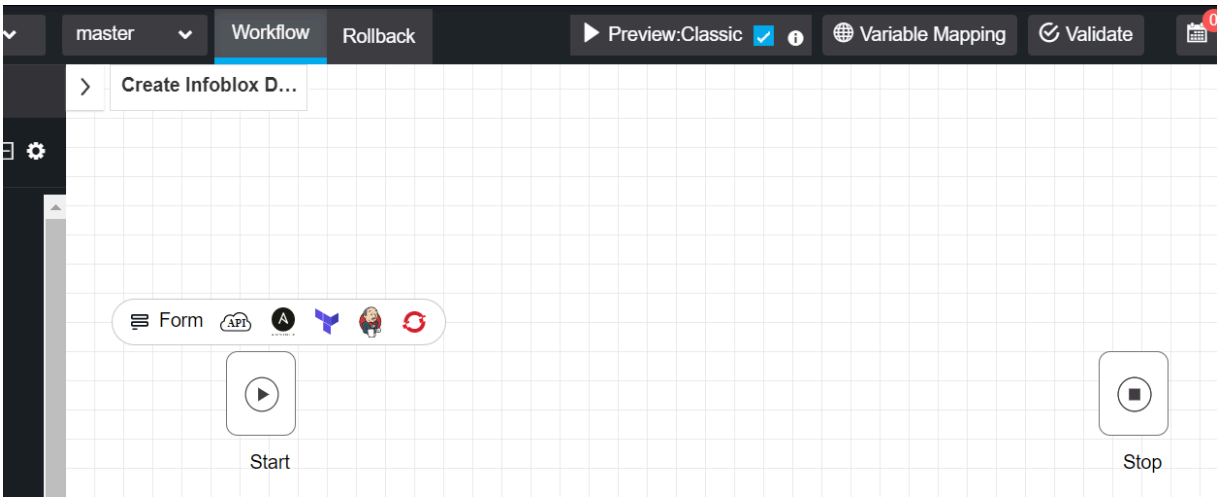


Field descriptions for the workflow design details

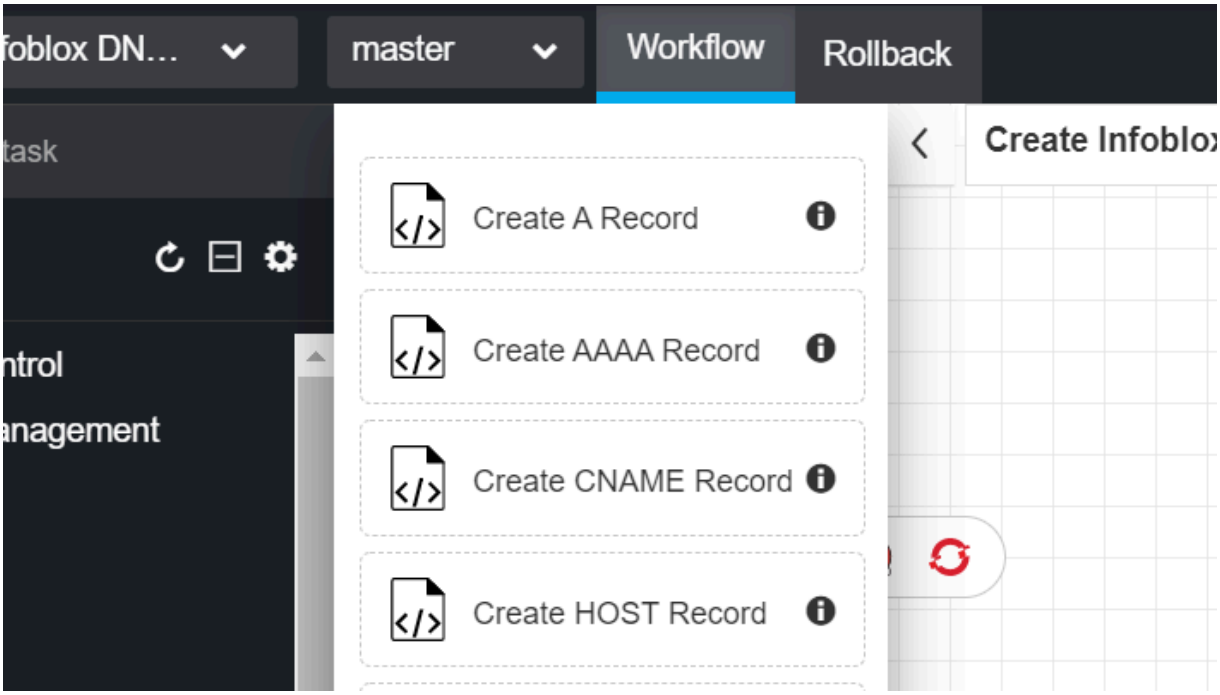
Field	Description
***Name**	Name of the workflow.
Description	Description for the workflow.
Category	Select the category from the drop-down option.
Subcategory	Select sub-category for the workflows from the drop-down option. The possible options are: • Modify • Create • Default • Delete

4. Click the **Save** button.
   The workflow is created:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

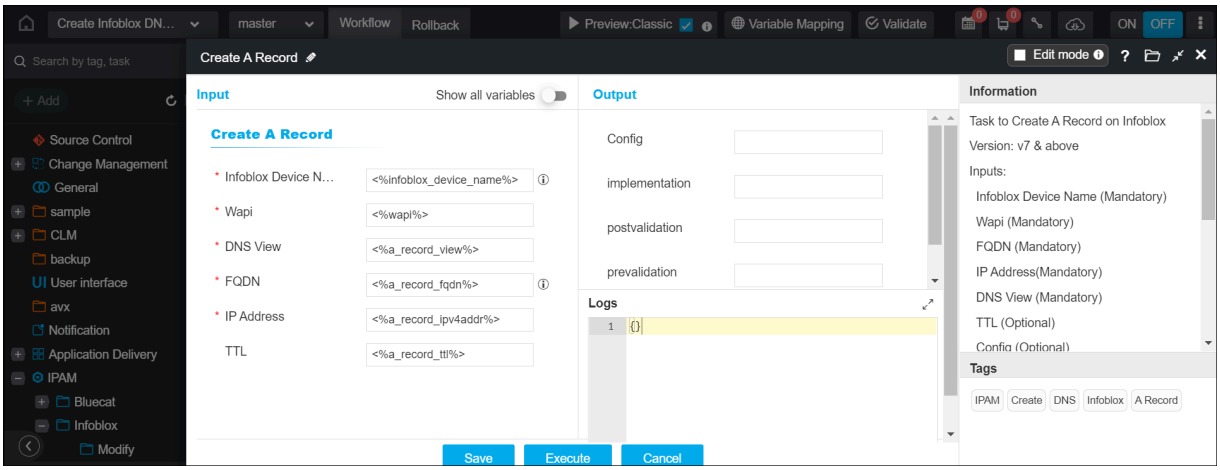


5. Go to **IPAM** > **Infoblox** > **Create** folder.
6. Place **Create A Record** task.



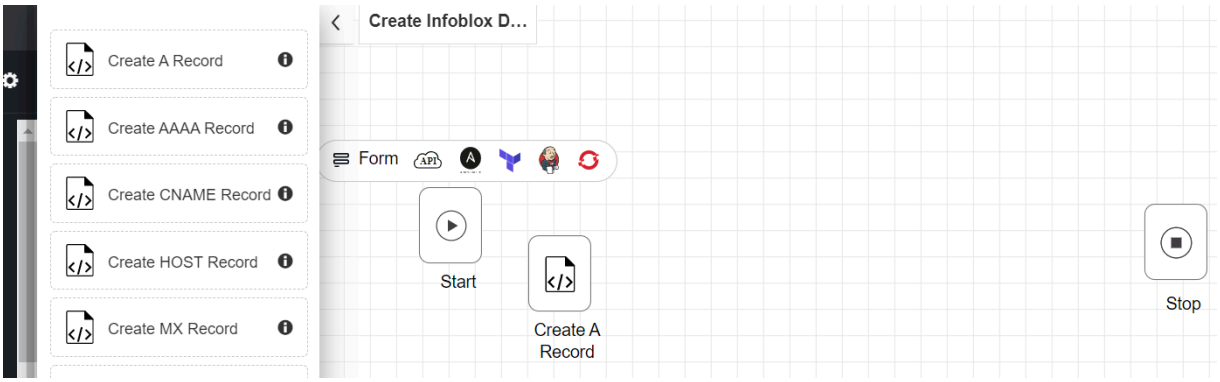
The Input, Output, and Information data for **Create A Record** task are displayed:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. Click the **Save** button.

   The **Create A Record** task has been added to the workflow:

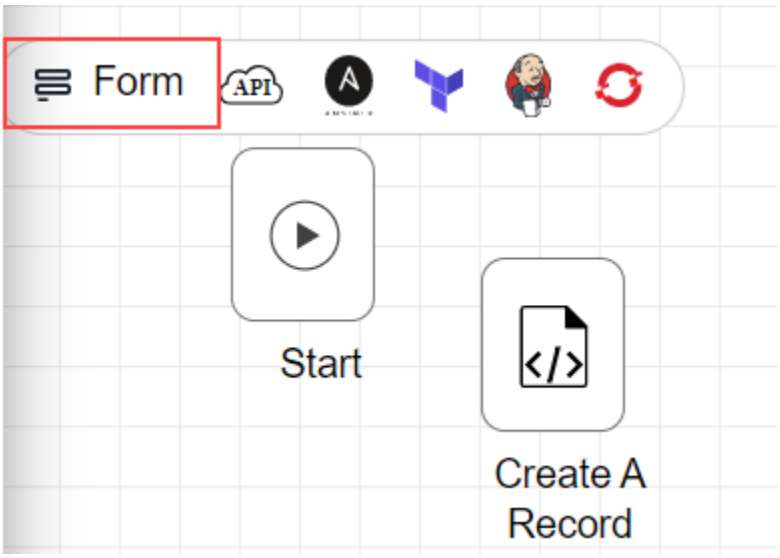


8. Similarly, place **Create CNAME Record** task and **Create PTR Record** task from the folder **IPAM** > **Infoblox** > **Create**.

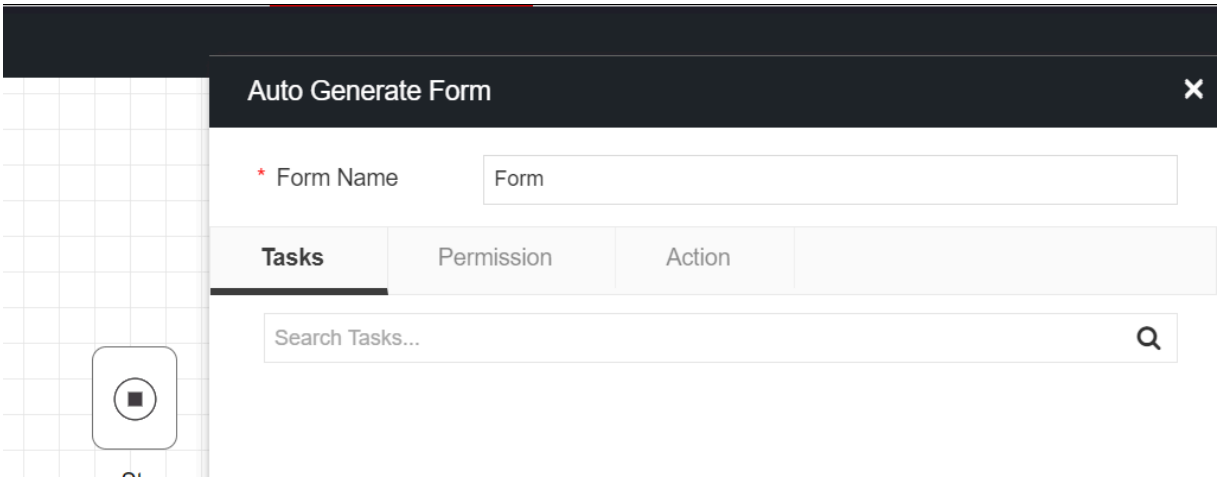


9. Click the **Form** in the Design page to auto-generate the form for the selected tasks.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

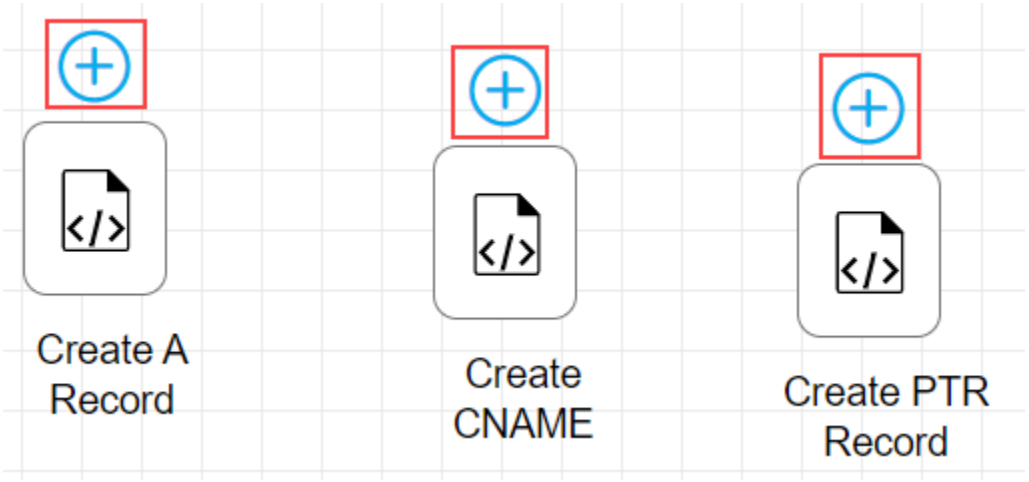


The Auto Generated Form appears:

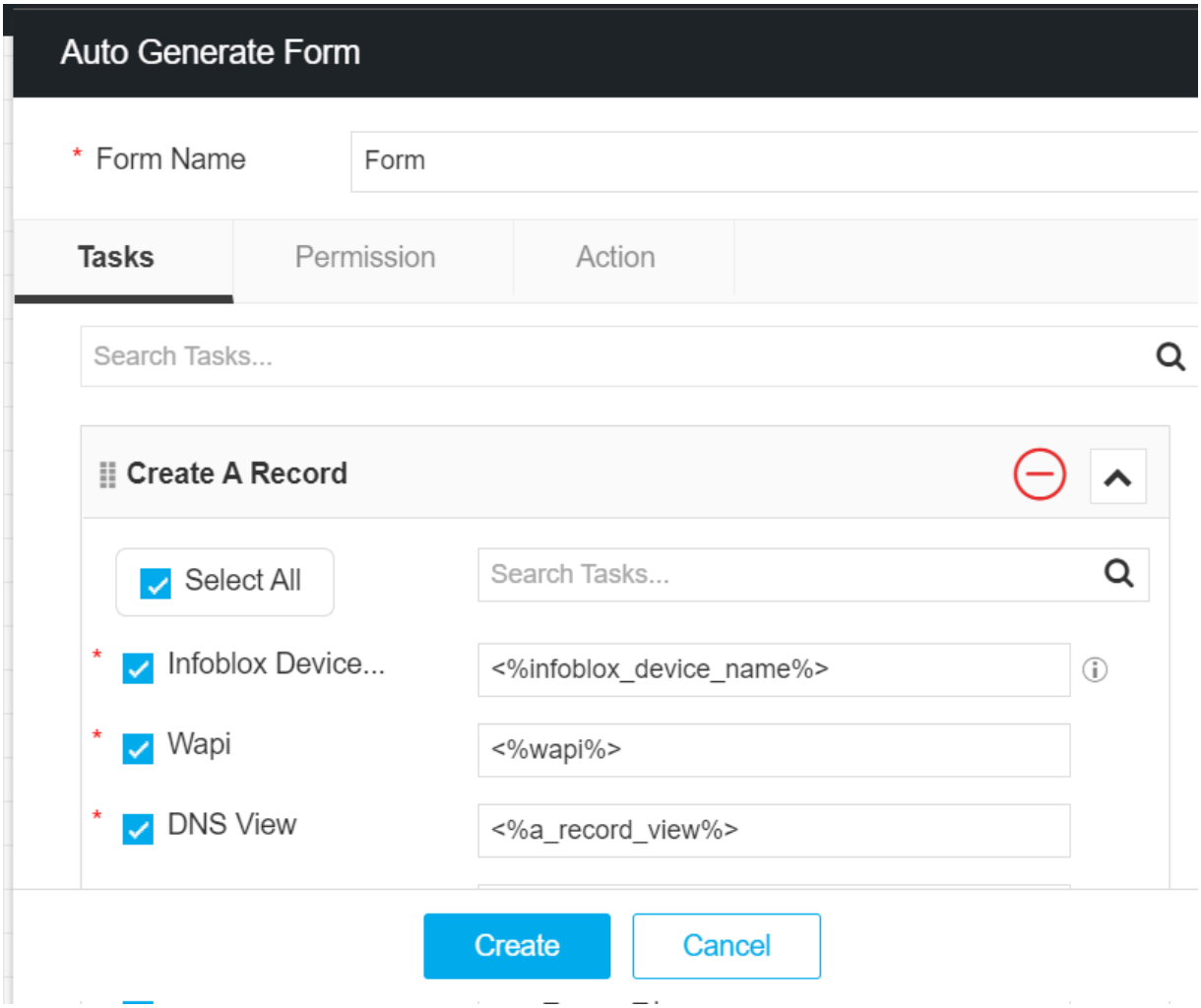


10. Click the add ( ) button of the **Create A Record**, **Create CNAME Record**, and **Create PTR Record** tasks.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



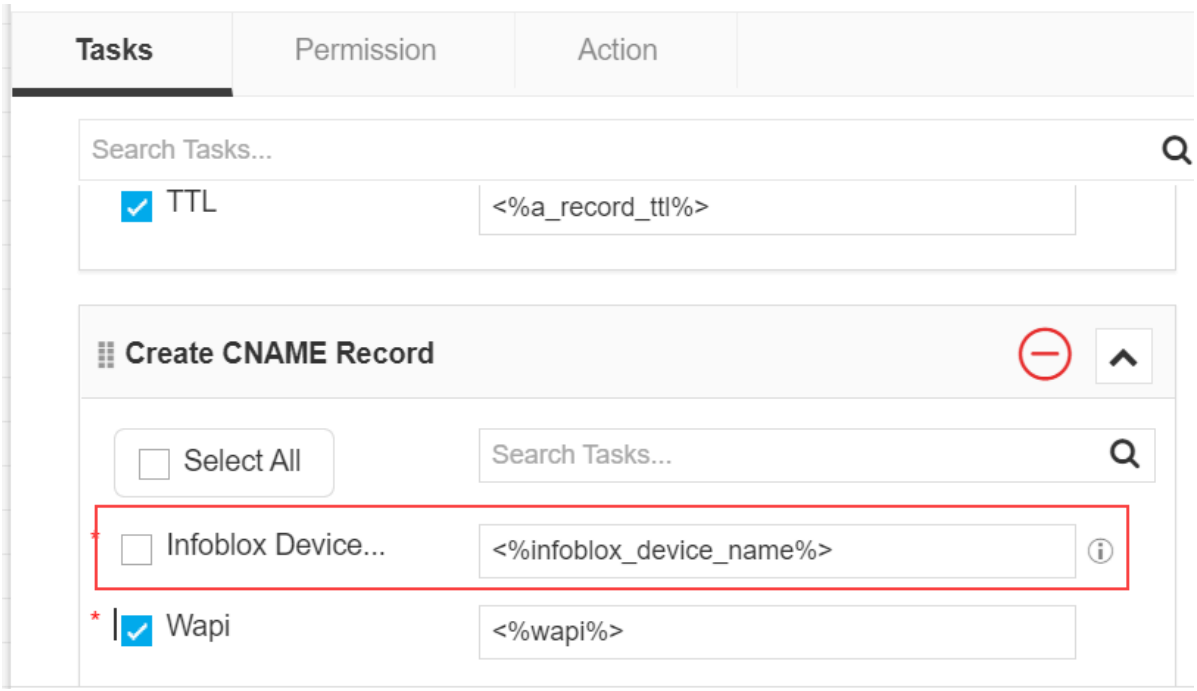
The tasks are added to the Auto Generate Form:



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

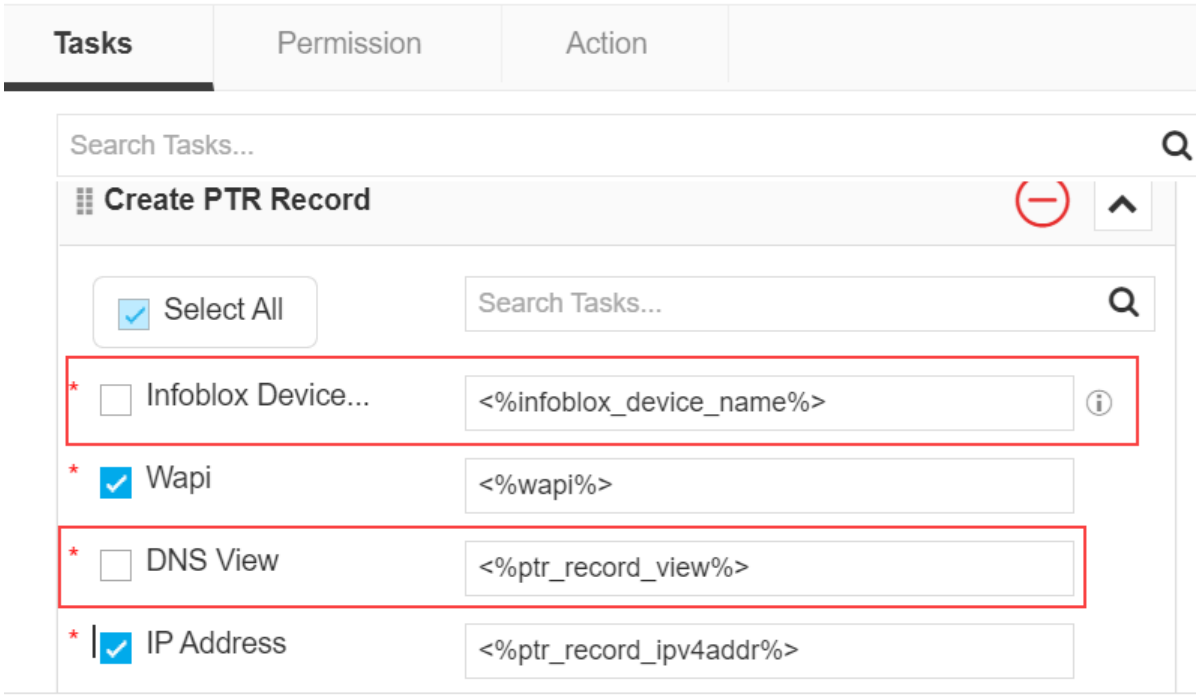
Note:
The sequence of the tasks is maintained based on the add button clicked sequence on the tasks. If you want to alter the sequence after adding the tasks to the Auto Generate Form, delete the task(s) from the Auto Generate Form and add them again.

11. Verify all the fields and select/clear the checkbox of the fields to add them to the form.
    The following fields are not to be selected to avoid duplication being shown in the form.
    - Clear the checkbox for the **Infoblox Device Name** field from the **Create CNAME Record** task and **Create PTR Record** task.

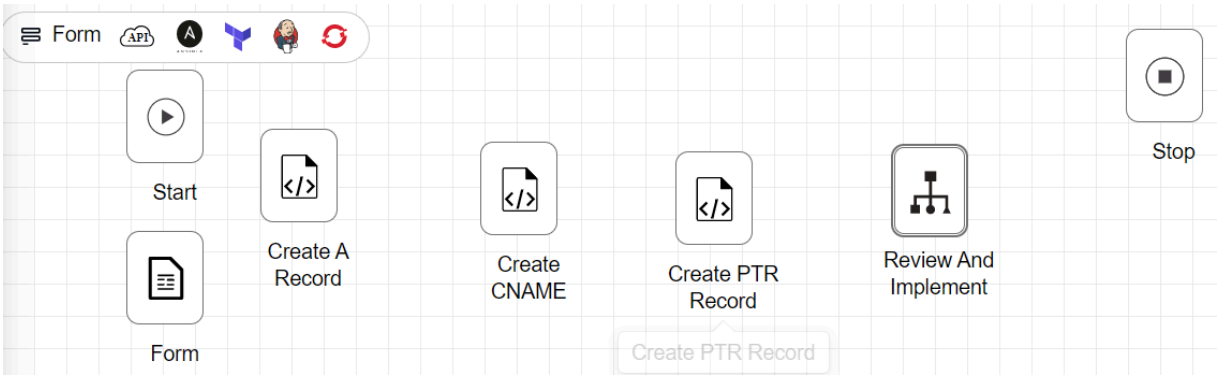


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

- Clear the checkbox for the **Infoblox Device Name** field and **DNS View** from the **Create PTR Record** task.

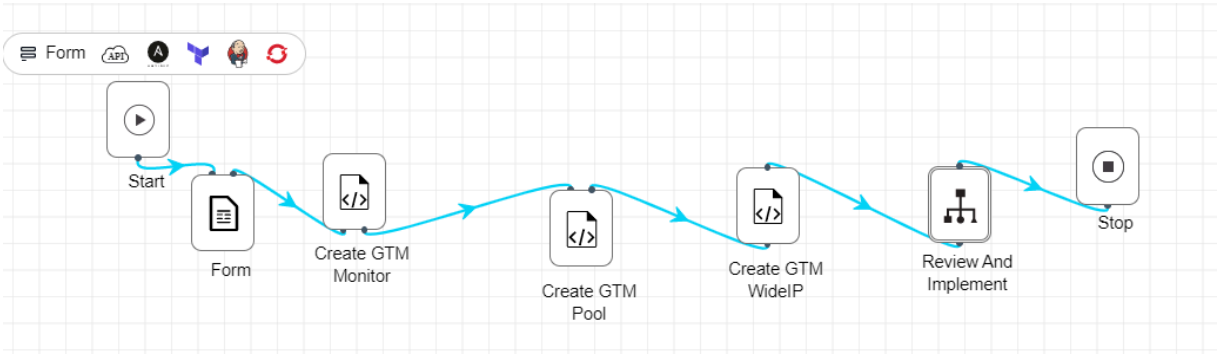


12. Click the **Create** button.
13. Go to **IPAM** > **Infoblox** > **Utils** folder, and then place **Review and Implement** subflow from the folder.

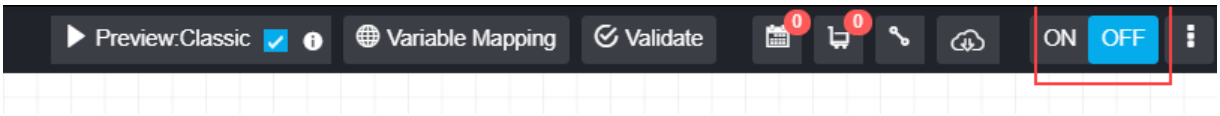


14. Connect all the tasks.

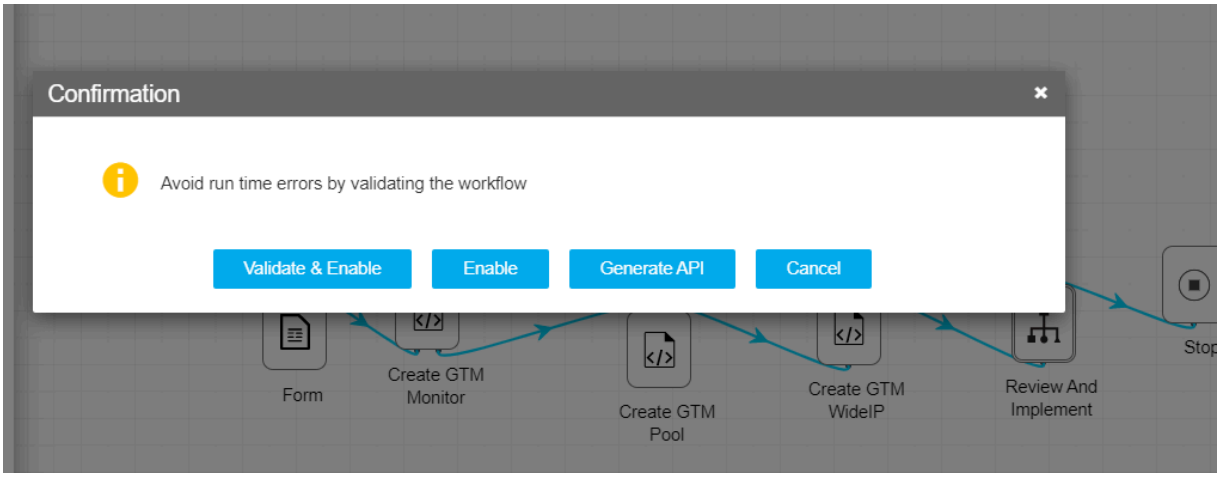
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



15. Enable the workflow by clicking the **ON** toggle button.

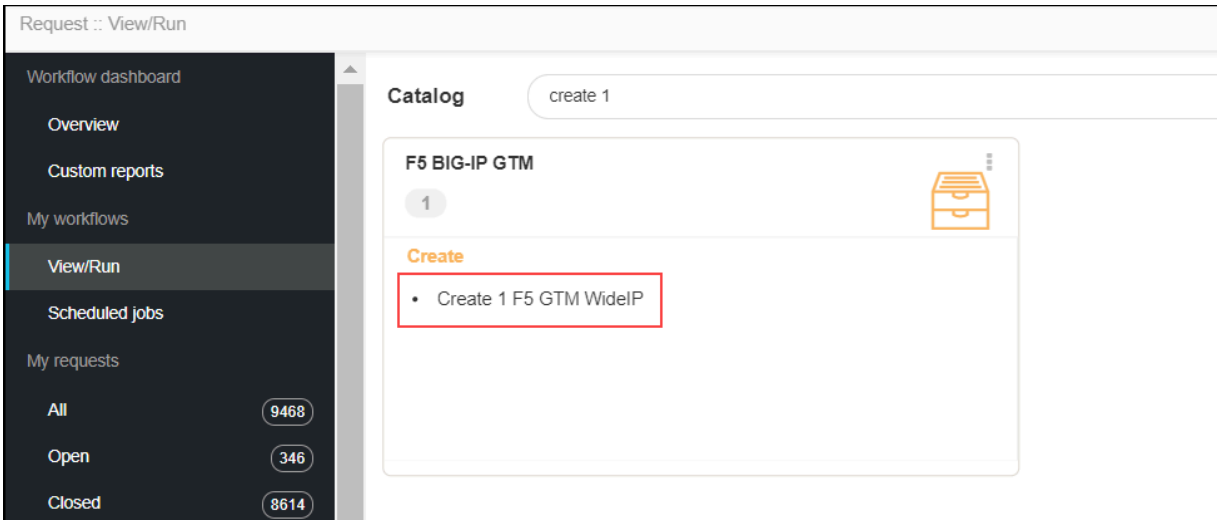


16. Click the **Validate & Enable** button.



The workflow is added to the Workflow Catalog page.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



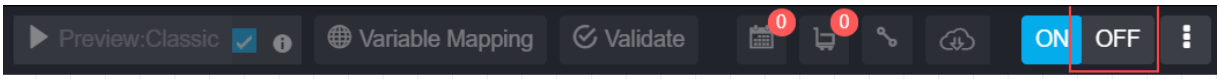
17. Run and verify the workflow.

Customize the Variables for a Task

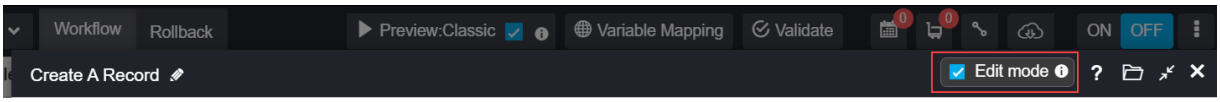
The variables can be customized for a task after creating workflow also.

To customize the variables for a task,

1. Go to **Menu** > **Studio** > **Workflow**.
   The Workflows page appears:
2. Click the workflow for which the variables are to be customized.
3. Disable the workflow by clicking the **OFF** toggle button.

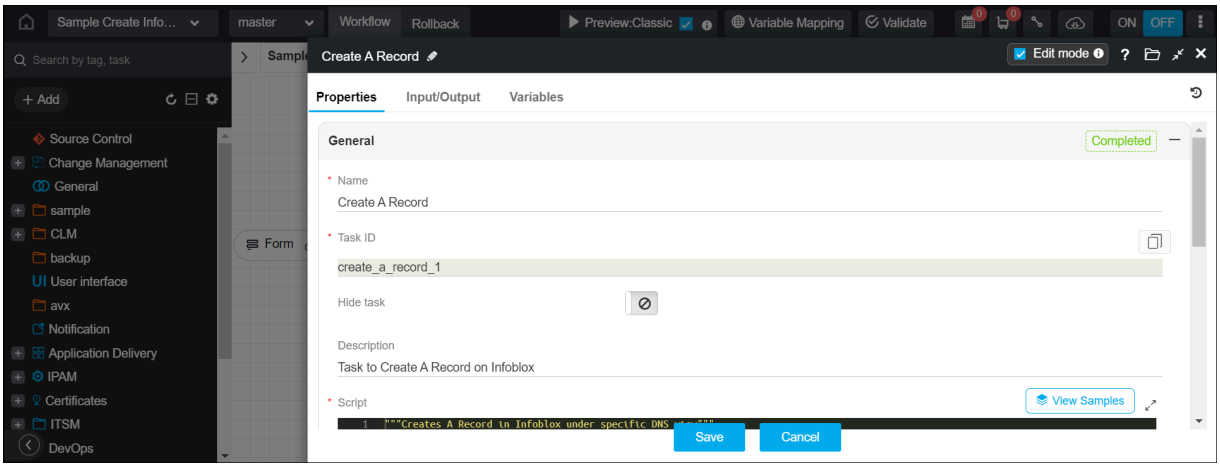


   The Confirmation popup appears.
4. Click **Yes** to confirm the disabling workflow.
5. Click twice on the desired task to customize the variables.
6. Make sure the **Edit Mode** check box is selected.

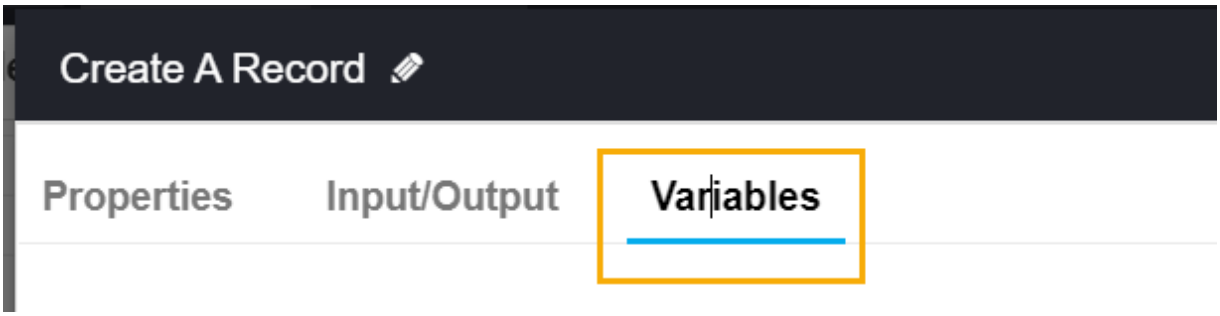


   The workflow Properties page appears:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



7. Click the **Variables** tab.



> **Note:**
> Make use of the scroll bar to view the details of the variables.

8. You can update any of the following details for the variables:

   - **Field Type** - select the field type from the drop-down option. For example, Description, Password, Dropdown, etc.
   - **Parent** - select the parent field from the drop-down option.
   - **Type** - select the type from the drop-down option. For example, String, JSON, Email, etc.
   - **Show variable**s - enable or disable the variable by clicking the toggle button.
   - **Mandatory** - mark whether the field is mandatory or not by clicking the toggle button.

9. Click the **Save** button, and then close the window.
   The updated variables are displayed in the form.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Designing BlueCat IPAM Automation Workflow

This section covers the following procedures:

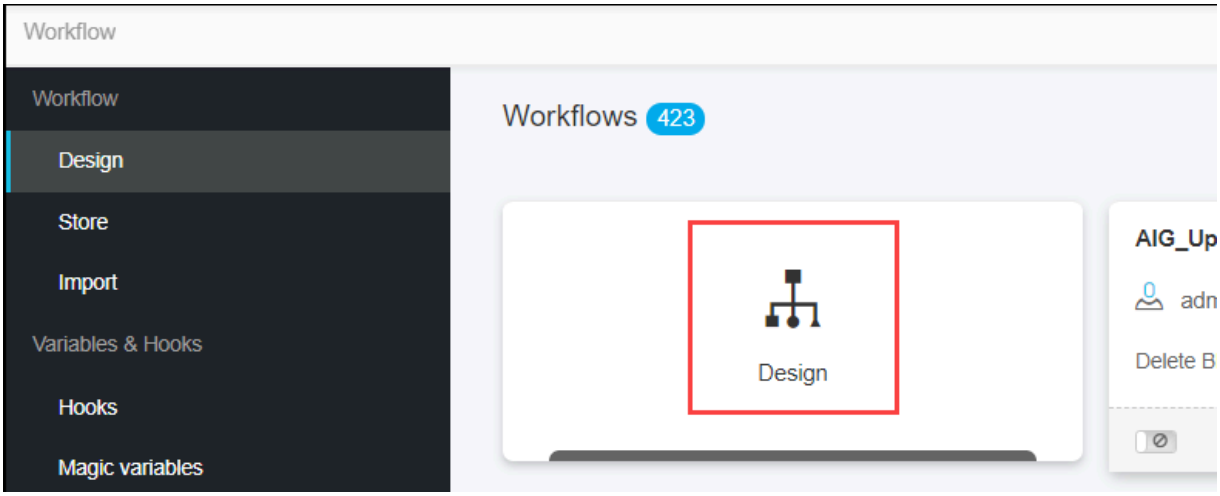
- Sample workflow creation with A, CNAME, and PTR Records in BlueCat
- Customize the variables for a task

Sample Workflow Creation with A, CNAME, and PTR Records in BlueCat

This section covers the sample procedures for creating a sample workflow with A, CNAME, and PTR Records in BlueCat.

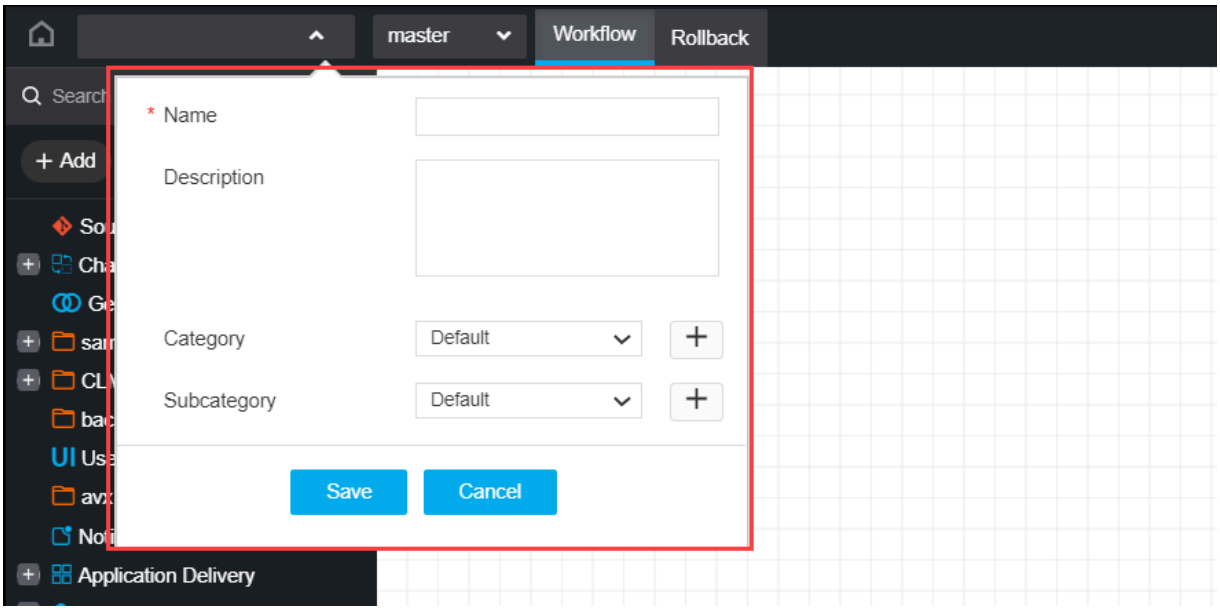
To create a sample workflow with A, CNAME, and PTR records,

1. Go to **Menu** > **Studio** > **Workflow**.
   The Workflows page is displayed.
2. Click **Design** to create a new workflow.



3. Enter/select the field information for the workflow design.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

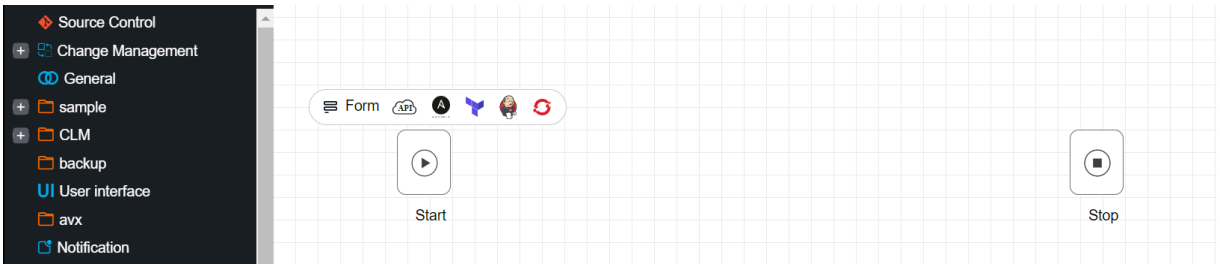


Field descriptions for the workflow design details

Field	Description
***Name**	Name of the workflow.
Description	Description for the workflow.
Category	Select the category from the drop-down option.
Subcategory	Select sub-category for the workflows from the drop-down option. The possible options are: • Modify • Create • Default • Delete

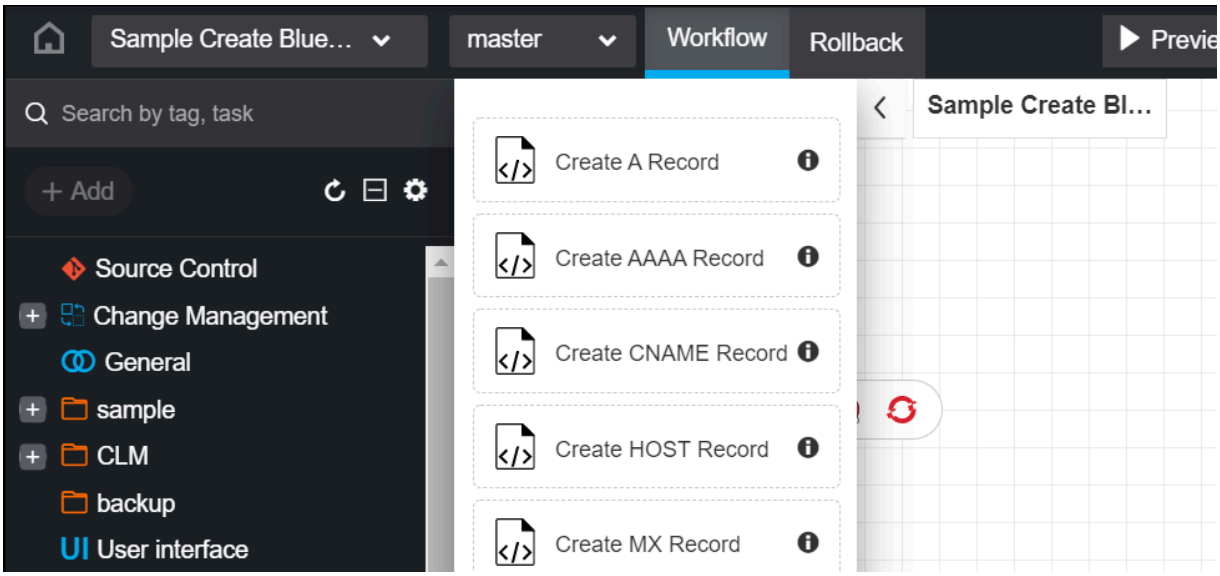
4. Click the **Save** button.

   The workflow is created.

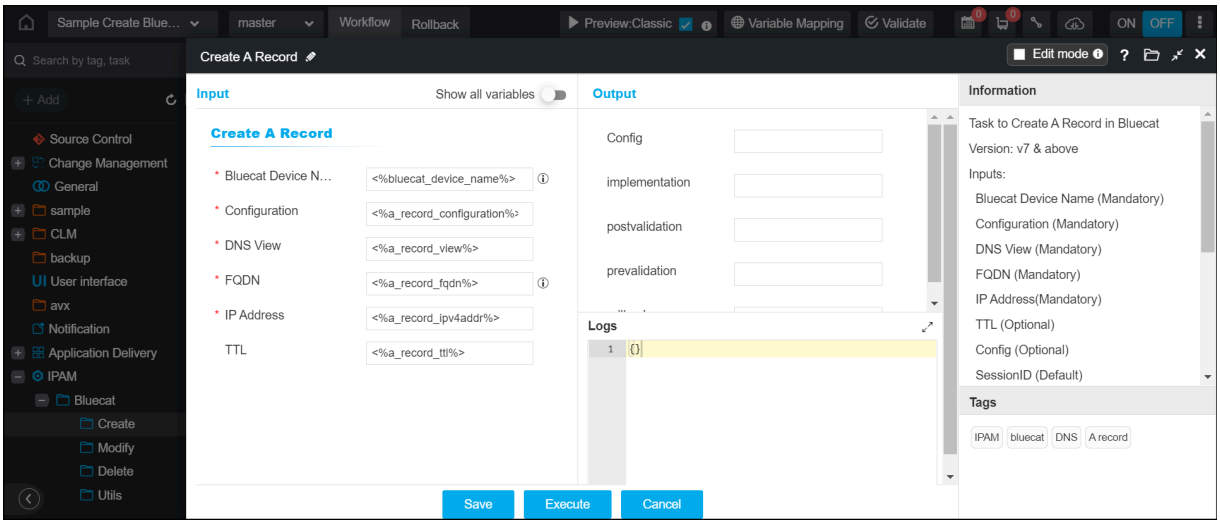


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

5. Go to the **IPAM** > **Bluecat** > **Create** folder.
6. Place **Create A Record** task.



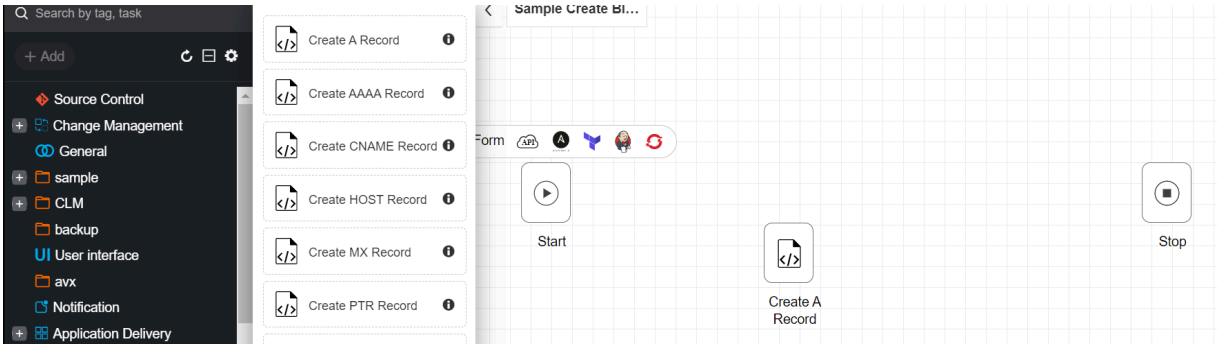
The Input, Output, and Information data for **Create A Record** task are displayed.



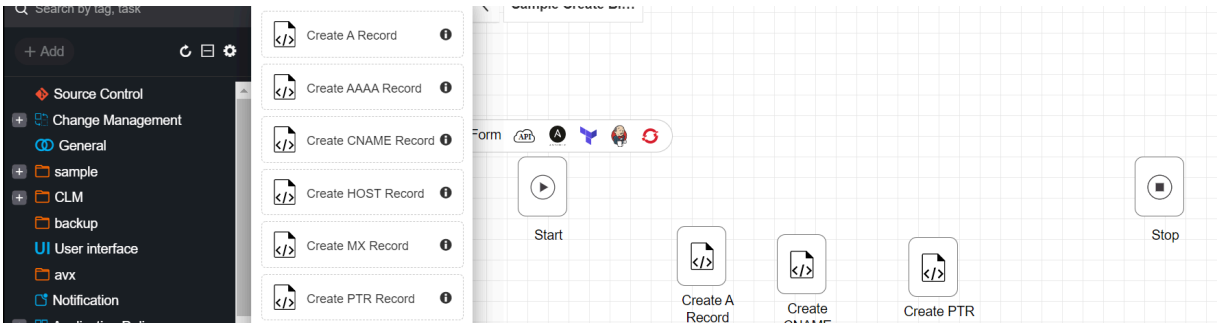
7. Click the **Save** button.

The **Create A Record** task has been added to the workflow:

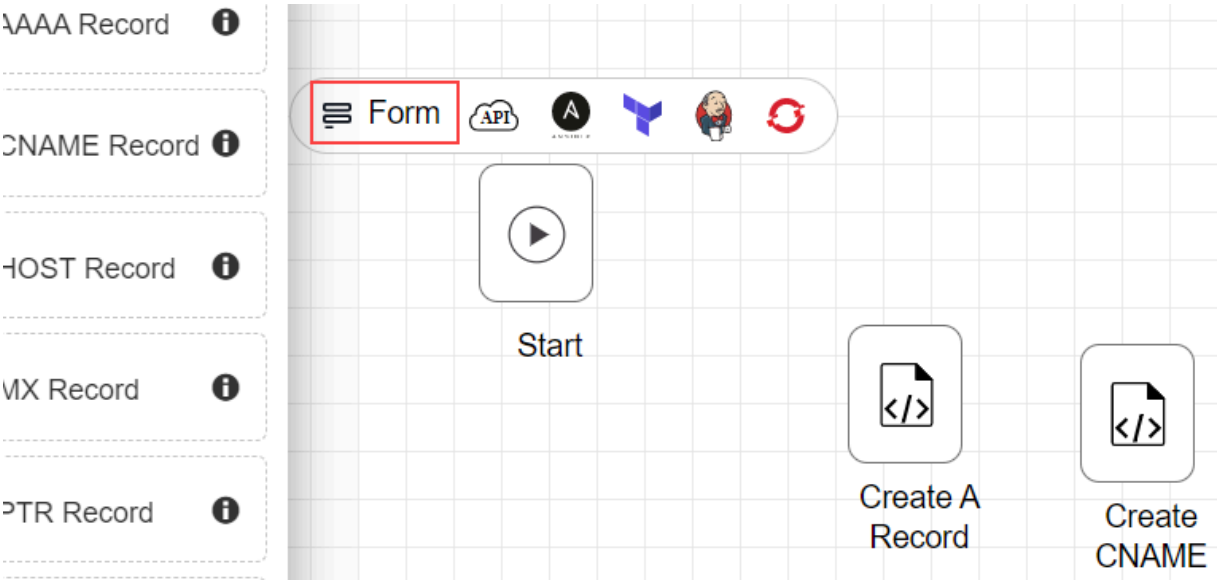
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



8. Similarly, place the **Create CNAME Record** task and **Create PTR Record** task from the folder **IPAM** > **Bluecat** > **Create**.

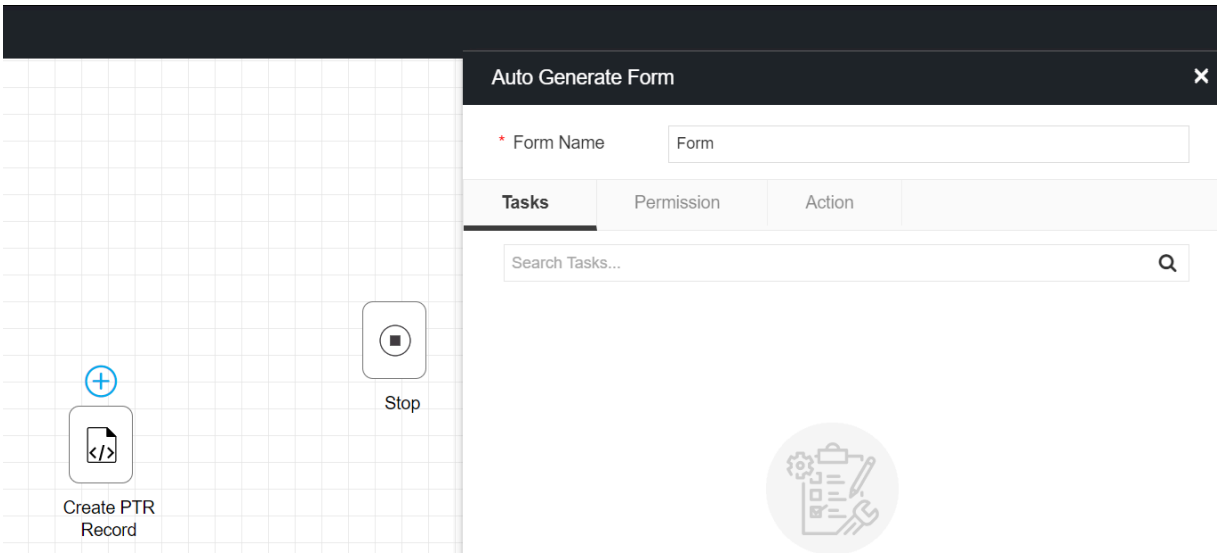


9. Click the **Form** in the Design page to auto-generate the form for the selected tasks.

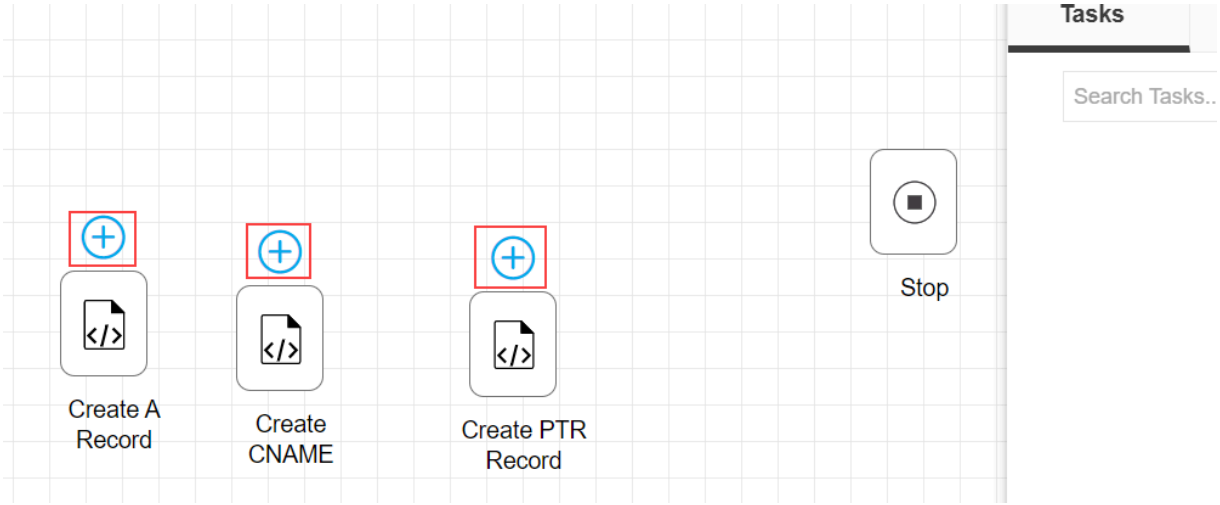


The **Auto Generate Form** appears:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

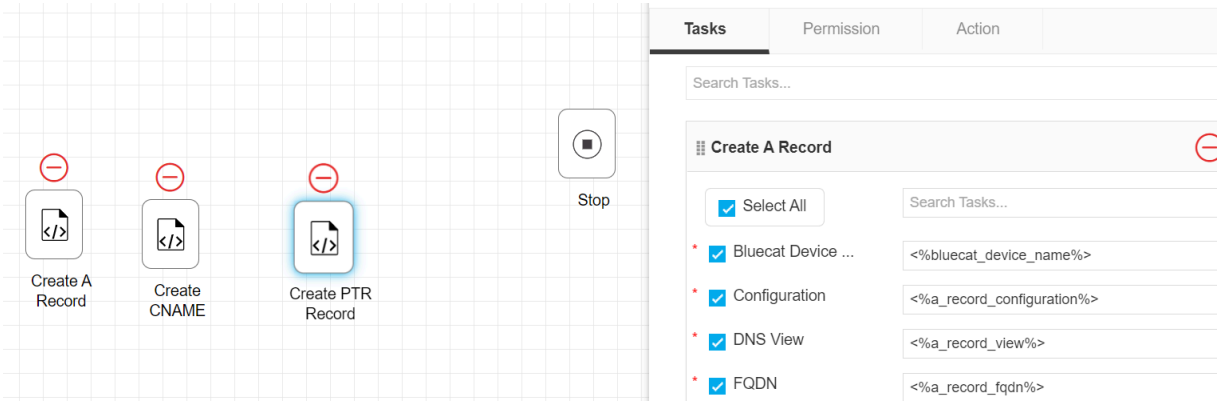


10. Click the add ( ) button of the **Create A Record** task, **Create CNAME Record** task, and **Create PTR Record** task.



The tasks are added to the Auto Generate Form:

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

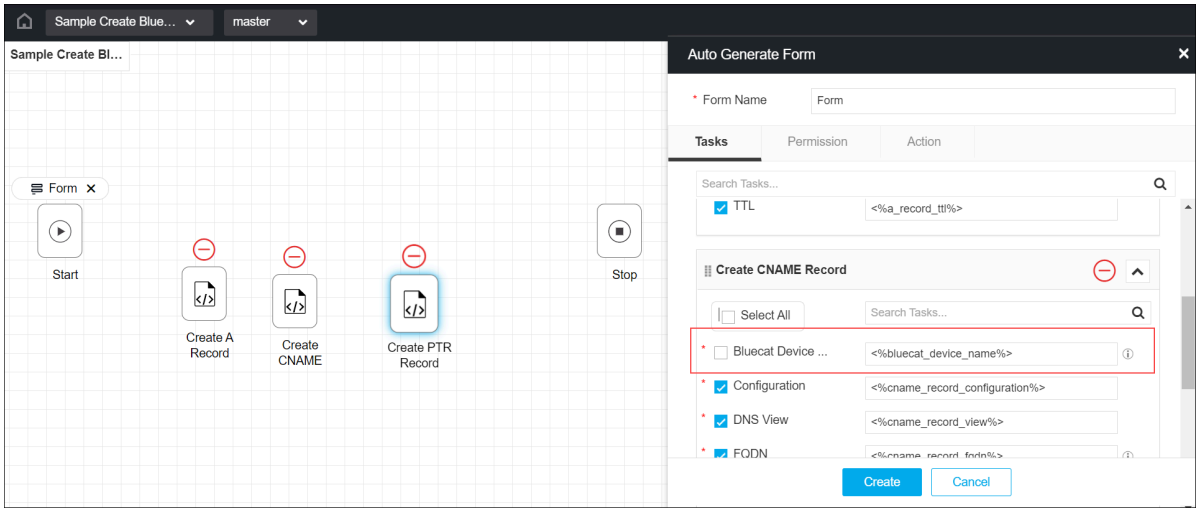


Note:

The sequence of the tasks is maintained based on the add button clicked sequence on the tasks. If you want to alter the sequence after adding the tasks to the Auto Generate Form, delete the task(s) from the Auto Generate Form and add again.

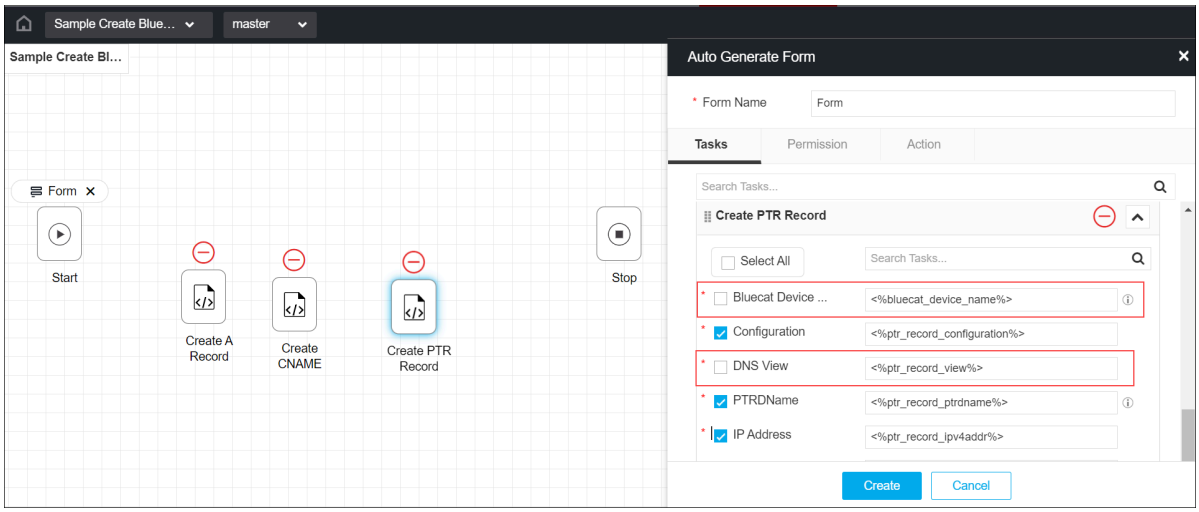
11. Verify all the fields and select/clear the checkbox of the fields to add them to the form.

    • Unselect the Bluecat Device Name field from the Create CNAME Record task.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

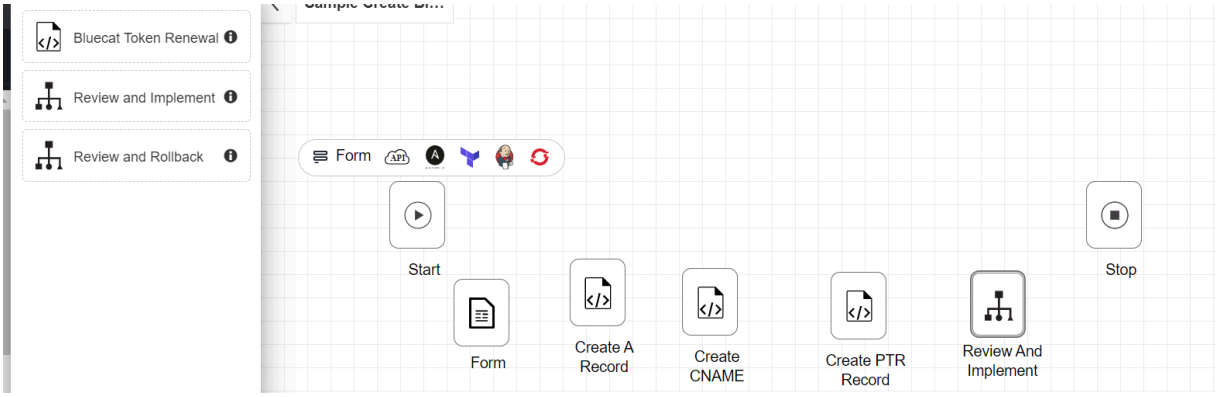
- Unselect the **Bluecat Device Name** field and **DNS View** from the **Create PTR Record** task.



Note:

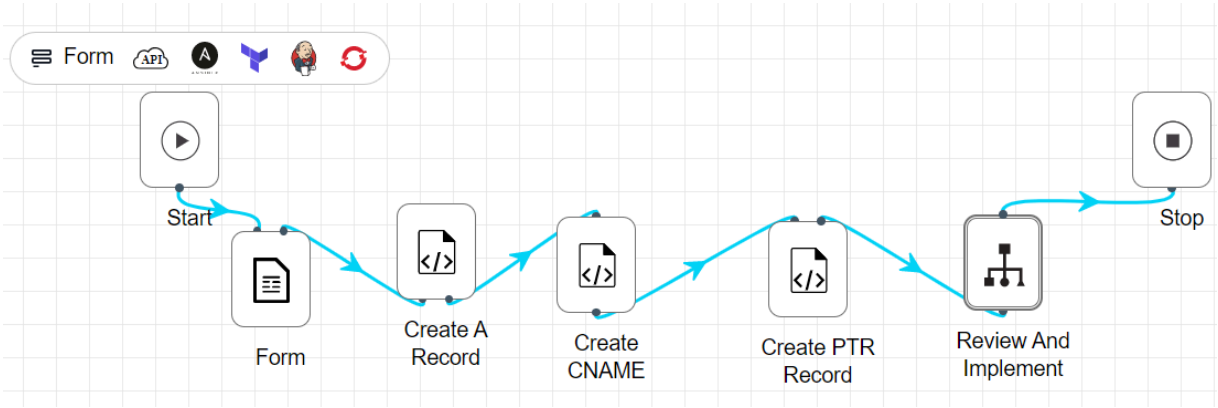
These fields are not to be selected to avoid duplication being shown in the form.

12. Click the **Create** button.
13. Go to **IPAM** > **Bluecat** > **Utils** folder, and then place **Review and Implement** subflow from the folder.

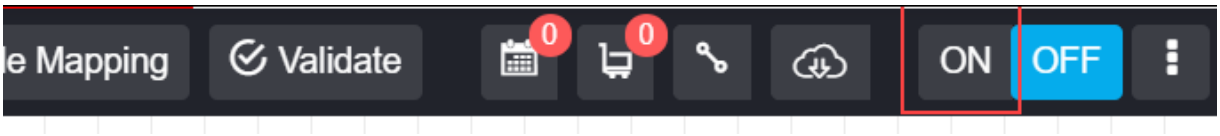


14. Connect all the tasks.

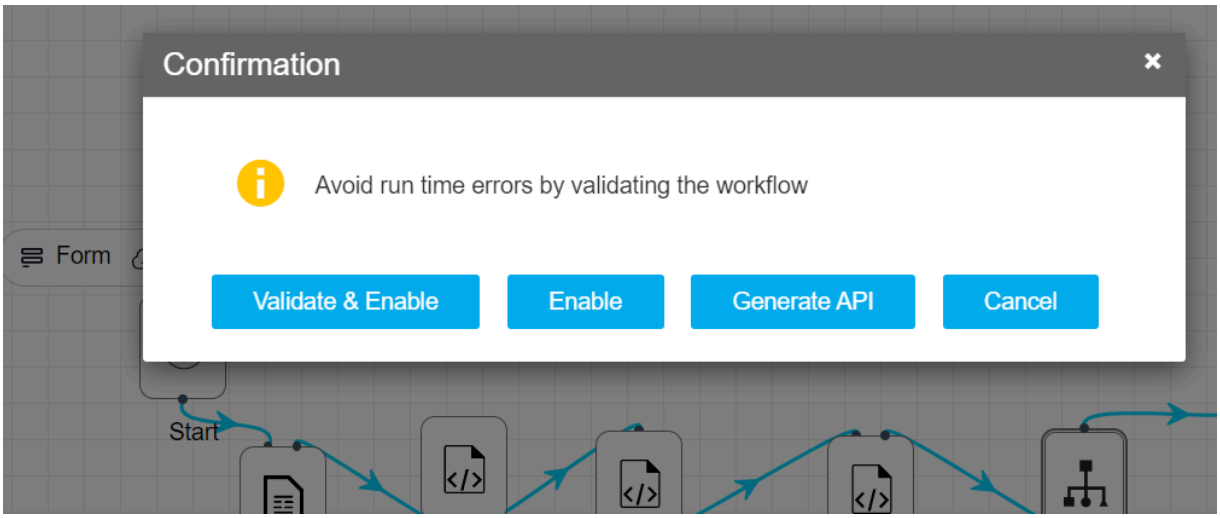
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



15. Enable the workflow by clicking the **ON** toggle button.

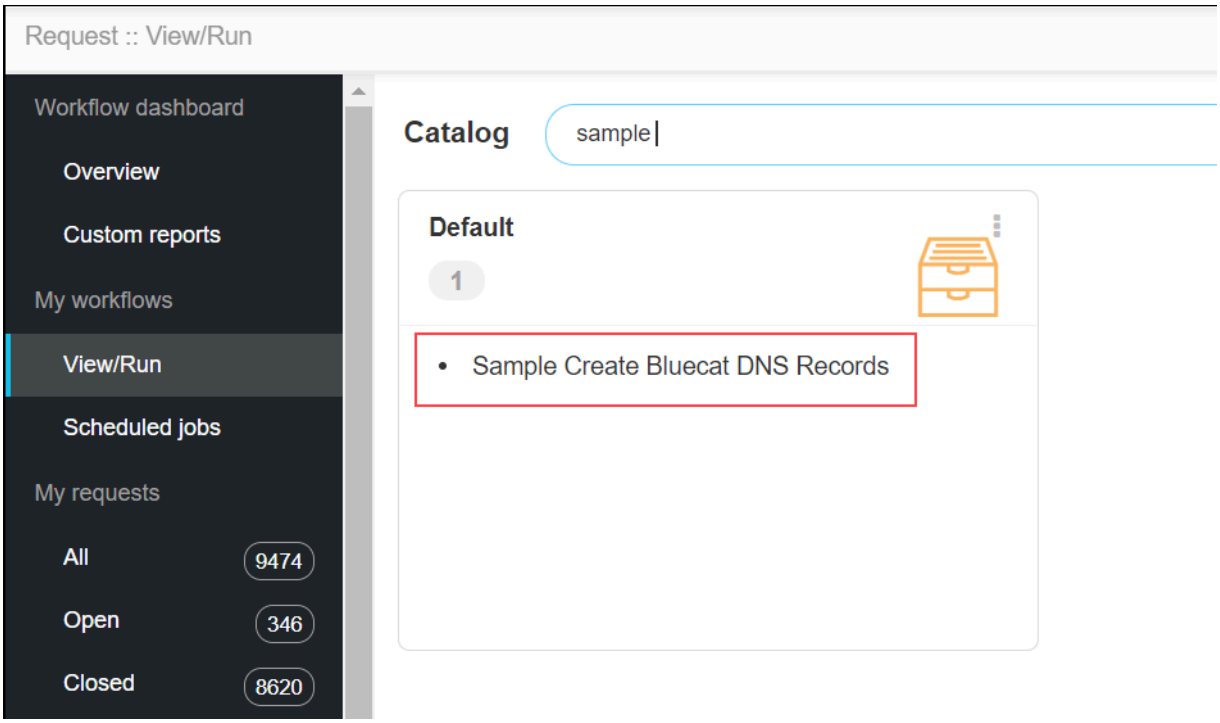


16. Click the **Validate & Enable** button.



The workflow is added to the Workflow Catalog page.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



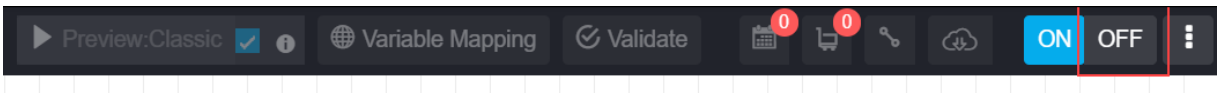
17. Run and verify the workflow.

Customize the Variable for a Task

The variables can be customized for a task after creating a workflow also.

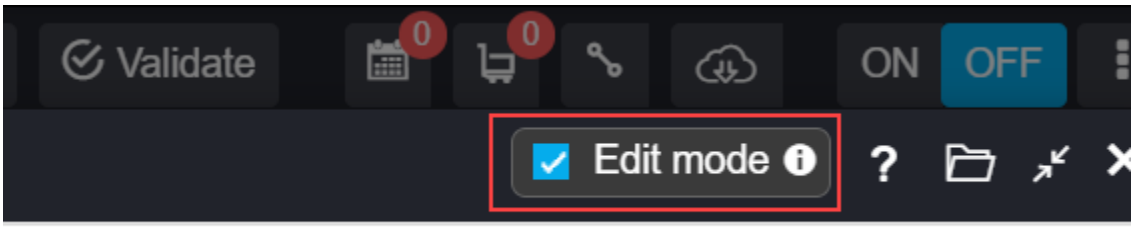
To customize the variables for a task,

1. Go to **Menu** > **Studio** > **Workflow**.
   The Workflows page appears.
2. Click the workflow for which the variables are to be customized.
3. Disable the workflow by clicking the **OFF** toggle button.

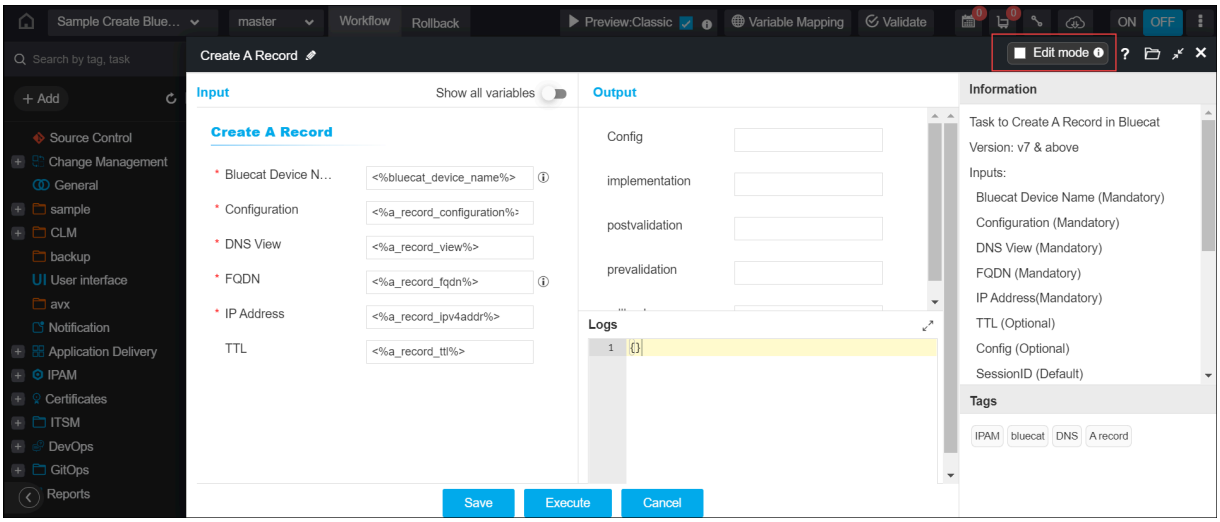


   The Confirmation popup appears.
4. Click **Yes** to confirm the disabling workflow.
5. Click twice on the desired task to customize the variables.
6. Make sure the **Edit Mode** check box is selected.

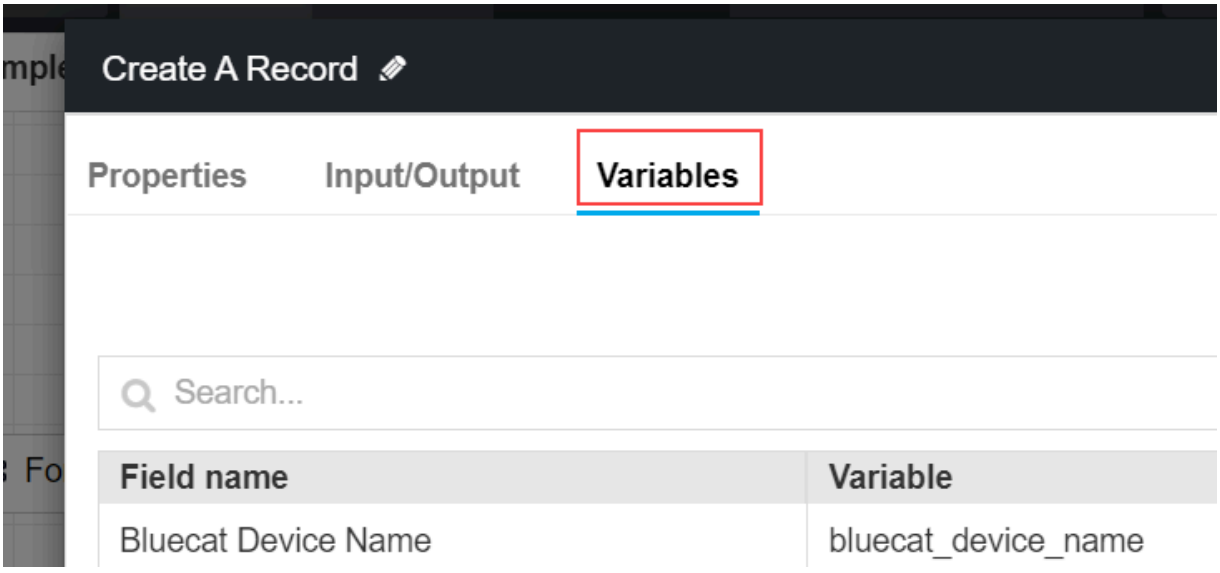
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



The workflow Properties page appears.



7. Click the **Variables** tab.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.

> **Note:**
> Make use of the scroll bar to view the details of the variables.

8. You can update any of the following details for the variables:
   - **Field Type** - select the field type from the drop-down option. For example, Description, Password, Dropdown, etc.
   - **Parent** - select the parent field from the drop-down option.
   - **Type** - select the type from the drop-down option. For example, String, JSON, Email, etc.
   - **Show variables** - enable or disable the variable by clicking the toggle button.
   - **Mandatory** - mark whether the field is mandatory or not by clicking the toggle button.
9. Click the **Save** button, and then close the window.
   The updated variables are displayed in the form.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Chapter 5: ADC+ Troubleshooting Guide

This document helps you to troubleshoot the common problems that you might encounter when using ADC functionalities. Effective troubleshooting documents depend on the availability of useful, detailed information. This guide will give more troubleshooting processes on Asset management and other ADC features.

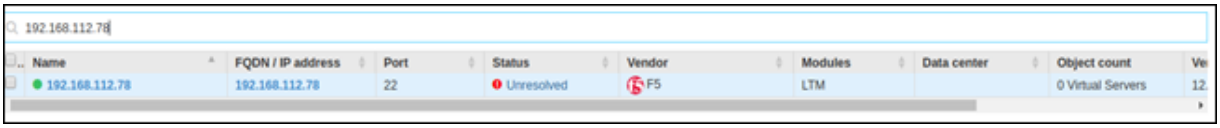
Overview

Asset Management Issues

Device addition in Unresolved/Failed State

The following recommended solutions are very common to all ADC vendors when device addition status moves to unresolved or Failed state. Vendor Specific issues and it's recommendations are given below.

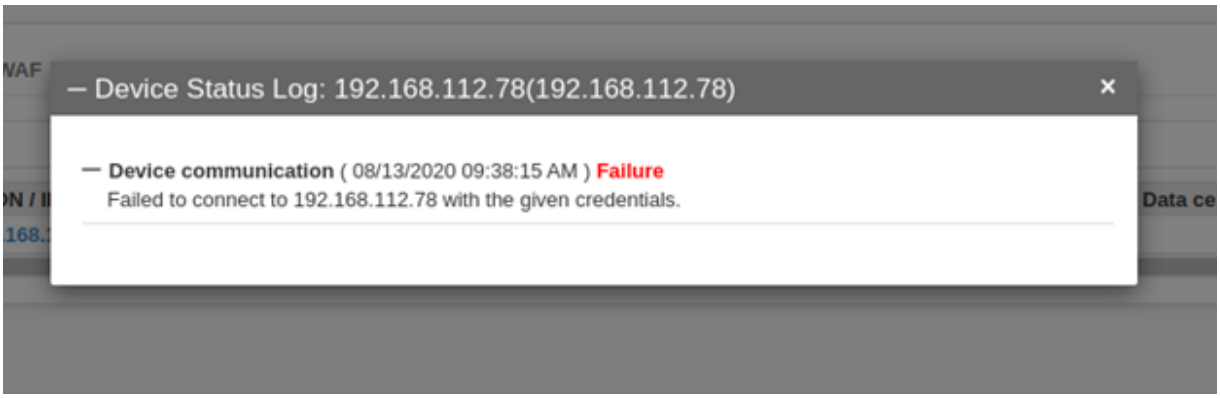
1. Log in to AppViewX.
2. Go to **Menu** > **ASSET MANAGEMENT** > **Device Inventory** and search for the device that was added.



3. Click on the status of that particular device.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Error Message	Possible Cause	Possible Solution
Failed to connect to <deviceIp> with given credentials.	Password of the device added might be wrong.	Click on the device name in the Inventory Page. Update the correct password and click Save.
Problem in checking shell access during terminal access verification.	This user might not have shell access permission in the device.	Check for shell access from AppViewX to the device with the same username/password used to add the device in AppViewX.
Modules - <Module names> not supported for version <version> by AppViewX.	Version might not be supported in AppViewX	Check for this particular version to be supported in the FSM document. Else contact Support team.
DeviceType does not match with the Provisioned Modules : <Module name>	There might be a mismatch in modules enabled in the device and the modules selected in AppViewX.	Cross check for which modules are selected in AppViewX for this particular device and those modules are enabled in the device or not. If there is a mismatch, select the module appropriately and save the device. Else contact Support team.
Error in parsing <Module names> <version> configuration files	Some error would have occurred during parsing in the code logic.	Contact Support team.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

A10 Device

1. Log in to AppViewX.
2. Go to **Menu** > **ASSET MANAGEMENT** > **Device Inventory** and search for the device that was added.
3. Click on the status of that particular device.

Error Message	Possible Cause	Possible Solution
Device communication failure with given enabled password.	Enable password of the device might not be provided or the provided Enable password might be wrong.	Click on the device name in the Inventory Page. Update the correct enable password of the device and click Save.
Device communication unsuccessful.	Password of the device might be wrong.	Click on the device name in the Inventory Page. Update the correct password and click Save.

Citrix Device

1. Log in to AppViewX.
2. Go to **Menu** > **ASSET MANAGEMENT** > **Device Inventory** and search for the device that was added.
3. Click on the status of that particular device.

Error Message	Possible Cause	Possible Solution
Device communication failed. Device Login Failed.	Password of the device might be wrong.	Click on the device name in the Inventory Page. Update the correct password and click Save.
AppViewX is unable to establish a secure connection to the NetScaler appliance.	The NetScaler management interface is bound to a 512-bit RSA certificate, which is considered insecure by modern Java security policies. Java enforces a minimum RSA key size of 2048 bits, causing the SSL handshake to fail with an "algorithm constraints" error.	a. Identify the SSL certificate bound to the NetScaler management service. b. Verify the key size of the certificate. c. Regenerate a new certificate with the following minimum parameters: • RSA Key Size: 2048 bits or higher • Supported signature algorithm (SHA-256 recommended) d. Bind the newly generated certificate to the NetScaler management service.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

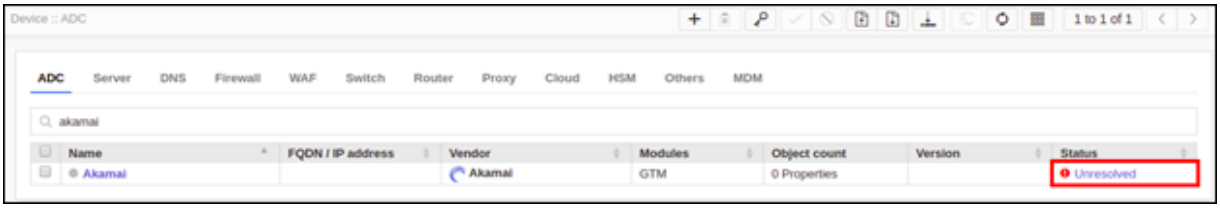
Error Message	Possible Cause	Possible Solution
		e. Restart the management service if required. f. Retry the connection from AppViewX.

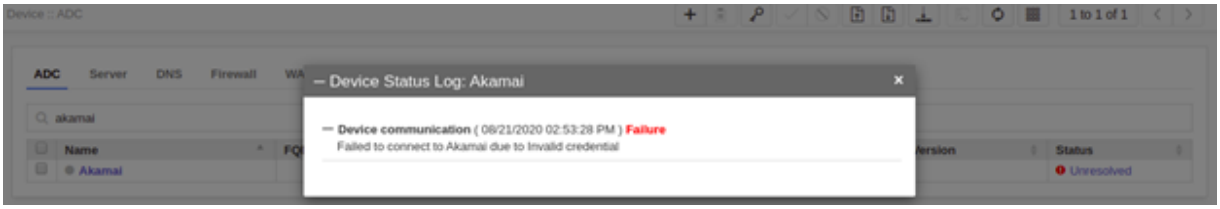
Akamai device

Akamai device addition will fail when the AppViewX node does not have an internet connection. For such issues, the recommended solutions are given below.

Error Message	Possible Cause	Possible Solution
Failed to connect to <Device Name> due to Invalid credential.	Proxy is not configured properly for an Akamai device.	Configure proxy for the Akamai device as shown above and trigger config fetch for the same device.

1. Log in to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory** and search for the device that was added.
3. Click on the status of that particular device.

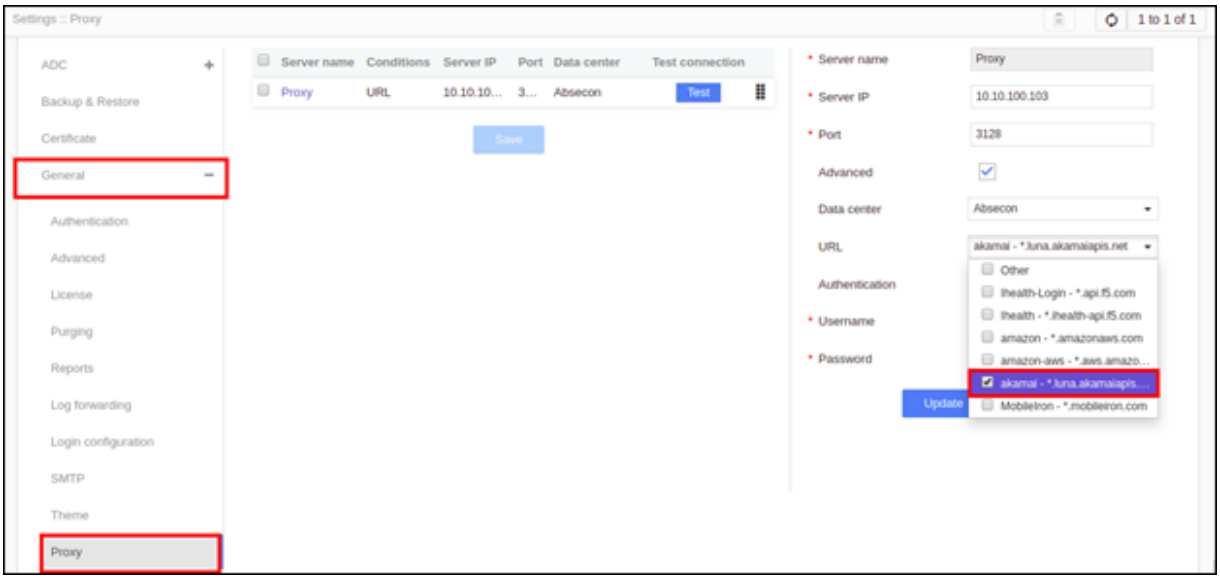




Recommendation:

1. Click **Main Menu > Settings > General > Proxy**.
2. From the **URL** list, select **akamai - *.luna.akamaiapis.net**.
3. Click **Update**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Click **Save**.

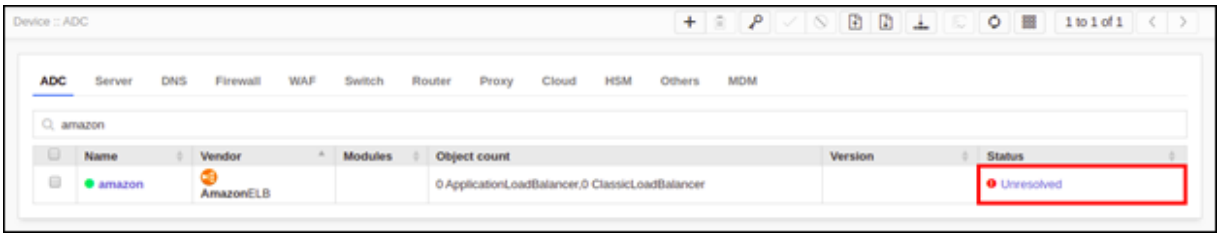
Amazon ELB

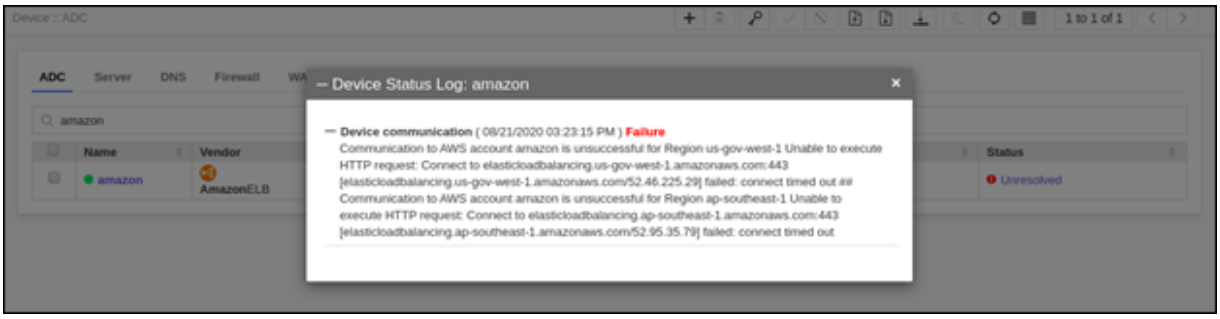
Amazon ELB device addition will fail when the AppViewX node does not have an internet connection. For such issues recommended solutions are given below.

Error Message	Possible Cause	Possible Solution
Communication to AWS account amazon is unsuccessful for Region <region> Unable to execute HTTP request: Connect to <device> failed: connect timed out ## Communication to AWS account amazon is unsuccessful for Region <region> Unable to execute HTTP request: Connect to <device> failed: connect timed out.	Proxy is not configured properly for an Amazon ELB device.	Configure proxy for the Amazon ELB device as shown above and trigger config fetch for the same device.

1. Log in to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory** and search for the device that was added.
3. Click on the status of that particular device.

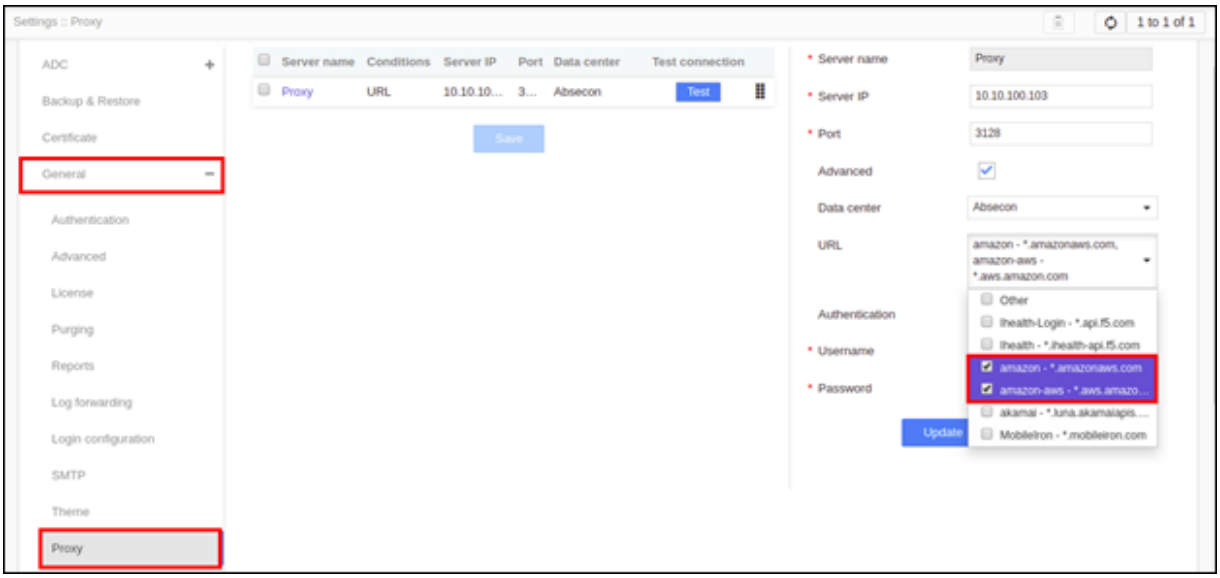
Copyright © 2025 AppViewX, Inc. All Rights Reserved.





Recommendation:

1. Click **Main Menu > Settings > General > Proxy**.
2. From the URL list, select **amazon -*.amazonaws.com** and **amazon-aws -*.aws.amazon.com**.
3. Click **Update**.



4. Click **Save**

WAF Devices

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Barracuda

This section helps you deal with the various issues faced by you during the use of Barracuda WAF functionalities.

Issues in Device communication

Error Message	Possible Cause	Possible Solution
Device communication failed	1. The given credential is incorrect. 2. The given IP/FQDN and Port may be incorrect. 3. Device not reachable.	1. Check and input the correct credential. 2. Check and input the correct IP/FQDN and Port. 3. Ensure the device is up and accessible.

Issues in Discovery

Error Message	Possible Cause	Possible Solution
Received failed response from batch	1. Device not running or not accessible. 2. No valid certificates available.	1. Ensure target device is up and running. 2. Check the device has valid certificate files.

Issues in Certificate Push

Error Message	Possible Cause	Possible Solution
Push failed due to invalid certificate	Push failed when pushing a certificate without key.	Ensure the certificate matches with the key associated in the device on pushing without the key file.
Device is not in managed state	Device is not available in “managed list”.	Please check the device status in device inventory.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

F5

This section helps you troubleshoot the common problems that you might encounter when using the F5 WAF functionalities. This guide will give more troubleshooting processes on F5 WAF certificate config fetch, discovery, CSR creation, backup, push, bind, rollback, and other actions associated with F5 WAF.

Issues in Fetch Config

Error Message	Possible Cause	Possible Solution
Authentication failed for the device	Invalid credentials specified for the device.	Provide valid credentials for the device.
Failed to fetch partitions from the device	Partition fetch failed.	1. Check the device partitions. 2. Please check the associated error message.
SSH connection failed	Unable to establish the SSH connection.	Please ensure that the device can be connected via SSH from AppViewX.

Issues in Discovery

Error Message	Possible Cause	Possible Solution
Please provide information as required	Discovery name is not given or length is less than 2 characters.	Enter a valid name with a minimum of 2 characters.
	Interval between batches info. is missing when execution type is sequential.	Provide a time interval between batches in minutes.
Please select a device	No device is selected in the Discover By section.	Select at least one device to discover certificates from.
Authentication failed for the device	Invalid credentials specified for the device.	Provide valid credentials for the device.
Failed to fetch partitions from the device	Partition fetch failed.	1. Check the device partitions. 2. Please check the error message associated.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
SSH connection failed	Unable to establish the SSH connection.	Please ensure that the device can be connected via SSH from AppViewX.

Issues in CSR Generation

Error Message	Possible Cause	Possible Solution
Authentication failed for the device	Invalid credentials specified for the device	Provide valid credentials for the device.
SSH connection failed	Unable to establish the SSH connection.	Please ensure that the device can be connected via SSH from AppViewX.
CSR Generation using HSM on this device version is not supported.	Version not supported.	Please use the supported version device to create CSR using HSM. Supported versions are v13 and v12.
1) CSR content unavailable in device 2) Requested CSR is not available in device 3) Csr could not be fetched	CSR content unavailable on device.	For some reasons CSR creation has failed, please check the logs to reproduce.
CSR already exists in this scope with the same Key reference name	The name is already used on the device.	Please use a different name.
Csr generation is device failed	CSR generation has failed.	CSR generation has failed. Please refer to the associated error message.
Thales HSM is not installed in the device. Please install and try again.	Thales is not installed in the device.	Please install Thales HSM and try again.
Invalid module. Please specify the valid module and try again.	Invalid module specified.	Specify a valid module and try again.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
The CSR <name> was not found in the device.	CSR content is not available in the name specified.	Please check the name in the CA connector.
NetHSM private key is not available/installed in device	NetHSM private key is not available/installed in the device.	Please install NetHSM private key on the device.

Issues in F5 WAF Backup, Push, Bind, and Rollback

Error Message	Possible Cause	Possible Solution
Unable to initiate request.	Pushing to device when certificate is unavailable, i.e, in a new state.	Push to device after certificate has been retrieved from CA.
	Previous work order is in progress and not completed.	Initiate push after previous work order is finished.
	AppConnector might not be in sync.	Synchronize the appConnector and retry.
Unable to initiate request, template is in disabled state	Given workflow is not in enabled state.	Enable the push/rollback workflow from the Workflow section.
User is not authorized	User does not have required permissions to push to the device.	Retry after getting the access for required action.
Application connector(s) not found	Application connector info was not found.	Provide the correct connectorId if not pushing using AppViewX UI.
Request associated with the application connector is in progress	Previous work order is in progress and not completed.	Initiate this request after the previous work order is finished.
Push not triggered or succeeded or No existing data available for backup process.	Rollback couldn’t proceed because push was not successful.	Only successfully pushed certificates can be rolled back.
Certificate not found.	Pushing to device when certificate is unavailable, i.e, in a new state.	Push to device after certificate has been retrieved from CA.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
Authentication failed for the device	Invalid credentials specified for the device.	Provide valid credentials for the device.
SSH connection failed	Unable to establish the SSH connection.	Please ensure that the device can be connected via SSH from AppViewX.
Invalid profile type	Specified profile type is not supported.	Please use client-SSL or server-SSL.
Certificate fetch failed	Certificate fetch failed.	Check the error message associated to know more about the problem.
Certificate already exists	Certificate is already available with the same.	Either change the name or enable overwrite in the application connector.
Key already exists	Key is already available with the same.	Either change the name of the key or enable overwrite in the application connector.
Certificate is expired	Certificate is expired.	Push the certificate which is currently valid.
Unknown certificate algorithm	Unknown certificate algorithm.	Push the certificates only with RSA or EC.
certificate is not yet valid	Certificate valid from value is greater than the current time.	Only valid certificates only can be pushed.

Imperva

This section will guide you to overcome the various issues faced by you during the use of Imperva WAF and its various functionalities.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Issues in Communicating to Imperva

Error Message	Possible Cause	Possible Solution
Communication to <ip> has failed	Invalid credentials Imperva WAF is down.	Please check if the correct Imperva credentials are provided. Please login to the SecureSphere console to check if Imperva is running.
Connection timeout	Internet / Proxy connection is not provided from AppViewX node. Firewall settings blocking the REST API request	Login to AppViewX portal and provide proxy details in Menu->Settings->General->Proxy and Check Firewall settings

Issues in Config Fetch

Error Message	Possible Cause	Possible Solution
Communication to <ip> has failed	Invalid credentials Imperva WAF is down.	Please check if a correct Imperva credentials are provided Please login to the SecureSphere to check if Imperva is running.
Device has 0 sites.	No sites configured in Imperva.	Please login to the SecureSphere to check if Sites are configured.
Device has 0 Server Group	No Server Groups configured in Imperva.	Please login to the SecureSphere to check if Server Groups are configured.
Device has 0 Services	No Services configured in Imperva	Please login to the SecureSphere to check if Services are configured.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Issues in Certificate Discovery

Error Message	Possible Cause	Possible Solution
Please provide information as required	1. Discovery name is not given or length is less than 2 characters. 2. Interval between batches info. is missing when execution type is sequential.	1. Enter a valid name with a minimum of 2 characters. 2. Provide a time interval between batches in minutes.
Please select a devices	No WAF account is selected in the “Discover By” section.	Select at least one device to discover certificates from.
Discovery Status Failed	Check if internet /Proxy connection is provided in AppViewX node.	Login to AppViewX portal and provide proxy details in Menu->Settings->General->Proxy.
No Certificates discovered from Imperva WAF	SSL Certificates are not discovered from Imperva due to API restrictions. Imperva Rest API doesn't allow the public Certificate String in the API responses. Only Certificate attributes like common name, signature, Expiry date and Issuer details are captured.	Check with AppViewX PM team / SME for more details

Issue in Push and Rollback

Error Message	Possible Cause	Possible Solution
An item with this name already exists - please select a different name	SSLKeyName should be unique.	Kindly ensure to provide a unique name in “Certificate File Name” field on Connector page.
Site not found	Unable to find the site to push the certificate	Please login to SecureSphere to check if the Site is available.

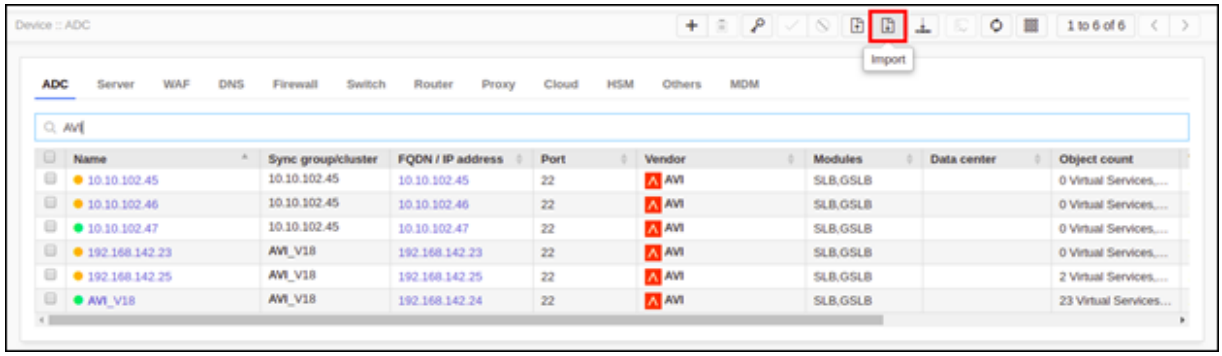
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
Server Group not found	Unable to find the Server Group to push the certificate	Please login to SecureSphere to check if the Server Group is available.
Services not found	Unable to find the Service to push the certificate	Please login to SecureSphere to check if the Service is available.

Device Inventory Action Issues

Issue in device import

1. Log in to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. To navigate to the **ADC Device Import** page, click **Import**.



4. Select a file having the extension as `xls` or `xlsx`.



5. Click **Upload**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
Import failed due to invalid column headers. Please refer to the import device instructions.	Import sheet has unsupported column headers.	Valid column headers are: device name, associated primary, sync group/cluster, ip address/fqdn, vendor, modules, data center, credential type, username, password, credential name, AppViewX group sync, enable password, port, vcmp guest, secondary/failover, token based authentication. Please validate the import sheet for these entries. Refer to the sample sheet as shown above.
Exception while Validating.	Import sheet contains invalid entries and it cannot be parsed.	Validate entries in the import sheet.
Exception while Validating.	The file has been removed during import.	Check if the import sheet exists.
Exception while Validating.	The sheet cannot be read due to insufficient permission.	Check for read permission to import sheets.

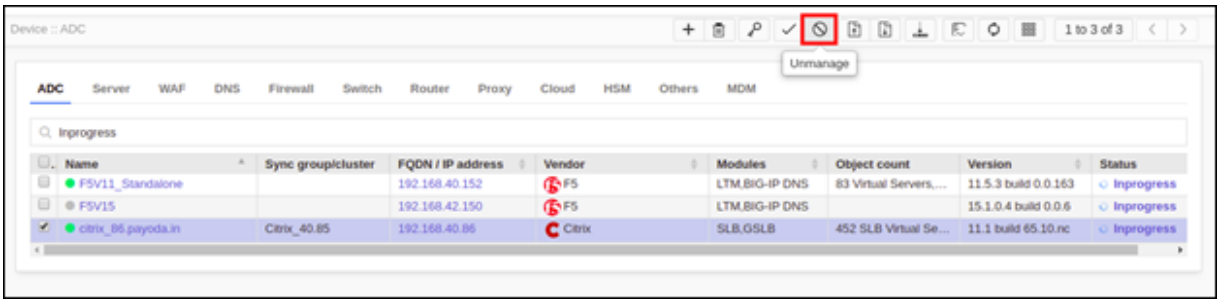
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

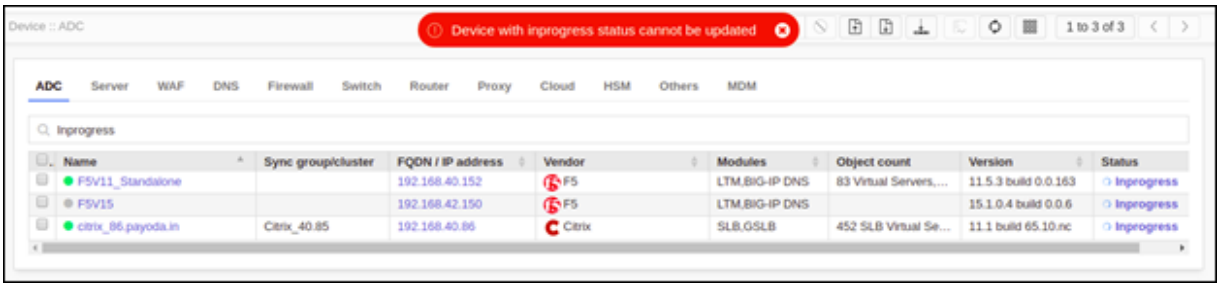
6. To download the sample import sheet, click the sample file icon.



Issue in Unmanaging a Device

1. Log in to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Select the device(s) that have their **Status** as **Inprogress** and click the **Unmanaged** icon.





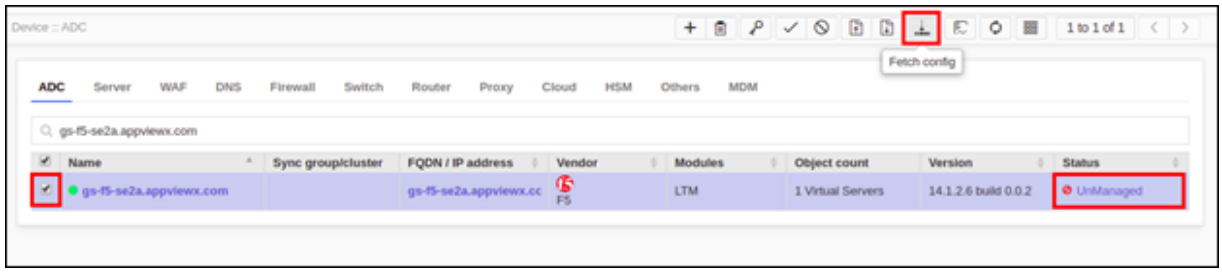
Error Message	Possible Cause	Possible Solution
Selected device(s) are in in-progress/unmanaged state. Operation cannot be performed.	Device cannot be unmanaged because it is already in unmanaged/ inprogress state.	Manage the device(s) in an unmanaged/inprogress state before performing action.
Config fetch not allowed for Unmanaged/Queued devices	One or more devices are in an inprogress/queued state and cannot	Wait until the device in inprogress/queued state

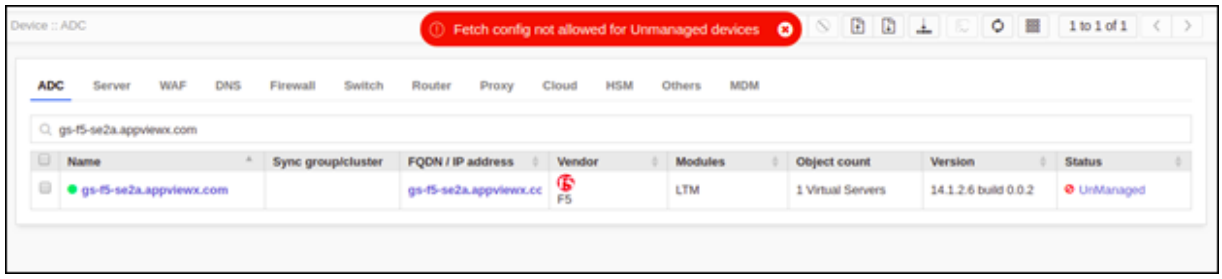
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
and Device(s) on which other operations is already in progress	be ignored as a stale state since the last updated time is within 300 seconds.	attains the timeout period of 300 seconds and perform action again.
Operation cannot be performed, Other operation is already in process on Device(s).	Update action cannot be performed on a device in an inprogress state.	Wait for the device to be resolved to a state before performing action.

Issue in Fetch Config

1. Log in to the AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Select the device(s) that have their **Status** as **UnManaged** and click the **config fetch** icon.

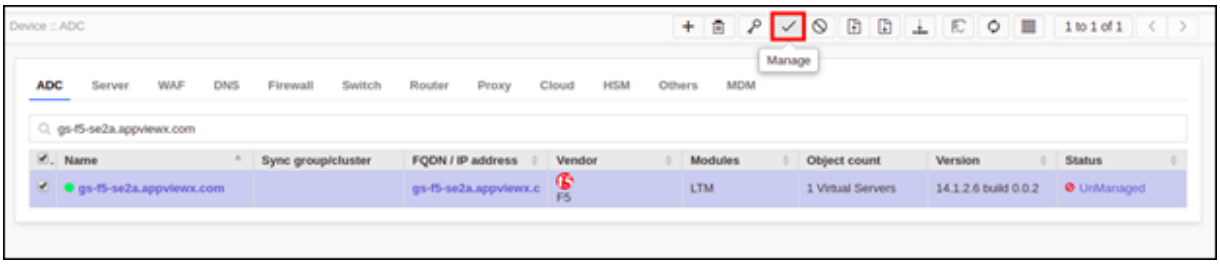




Error Message	Possible Cause	Possible Solution
Config fetch not allowed for Unmanaged/ Queued devices and Device(s) on which other operations are already in progress.	One or more devices went to an unmanaged state after the action was triggered.	Manage the device(s) in an unmanaged state before performing action.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

4. Select the unmanaged device(s) and click the **manage** icon and once device status updated to Managed we can trigger the fetch config action.



Device Backup and Restore Issues

Issue in schedule/trigger device backup

1. Login to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Search for the Device name.
4. Expand the group to check the generated backups.
5. If the backups are not generated, check the Audit Log to identify the root cause for backup failure.

Error Scenario	Possible Cause	Possible Solution
Unavailability of device version.	Device may be still in inprogress state, and yet to fetch the device version.	Wait for the device version check to be completed. Check in device inventory.
	Device may be in an unresolved state.	Check the device credentials.
Backup cannot be performed on an unresolved device.	Device is in an unresolved state.	Check for device credentials and manage the device.
Problem in establishing communication with devices.	Device is in managed state in AppViewX. But the credentials of the device may be changed now.	Check for the device credentials and update the latest credentials.
Scheduled Backup did not get triggered and backup is not generated.	Scheduler component is not running.	Please contact the administrator to start the scheduler component.

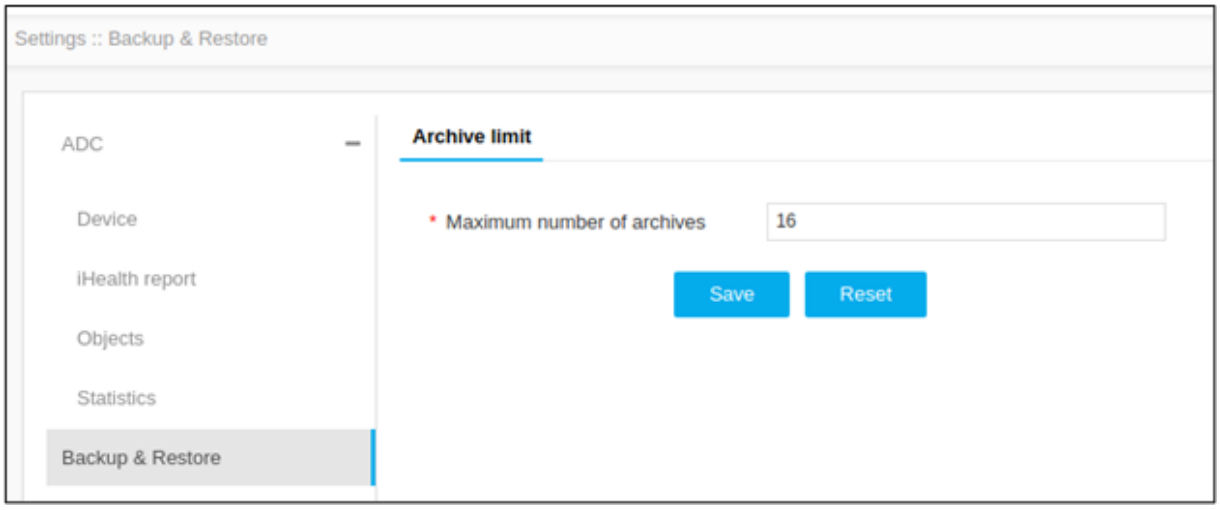
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

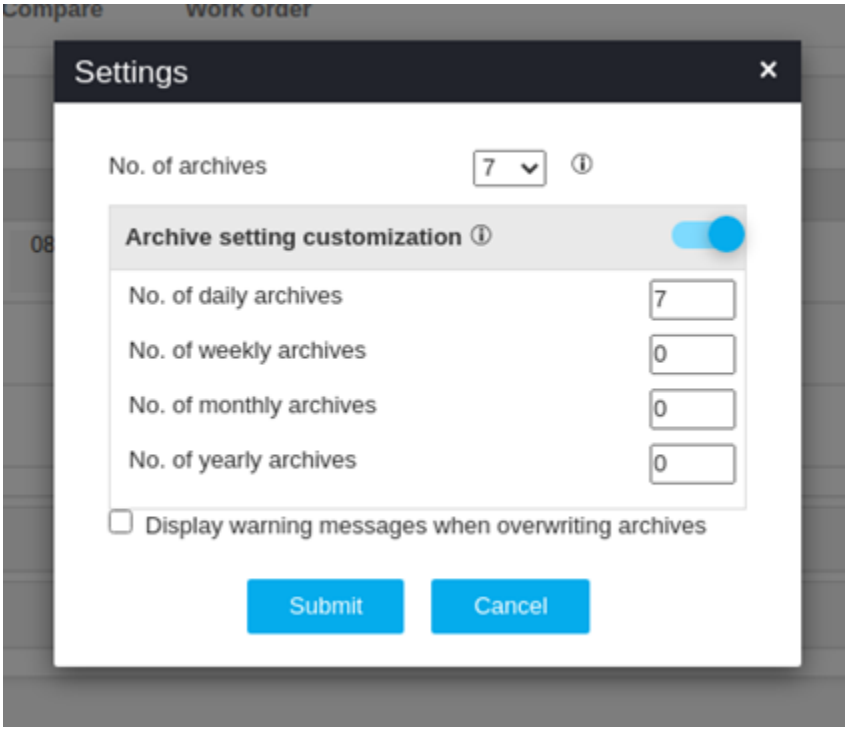
Unable to View the Old Backup Files

1. Login to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Expand the group to check the generated backups.
4. If old backups are missing, check the backup settings page.

Error Scenario	Possible Cause	Possible Solution
Old Backup files are missing.	Number of archives to be retained has been modified.	Maximum number of Archives to be maintained can be set in Settings > Backup & Restore Maximum value allowed is 16. More specific settings can be done at Settings available at Backup & Restore Page.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



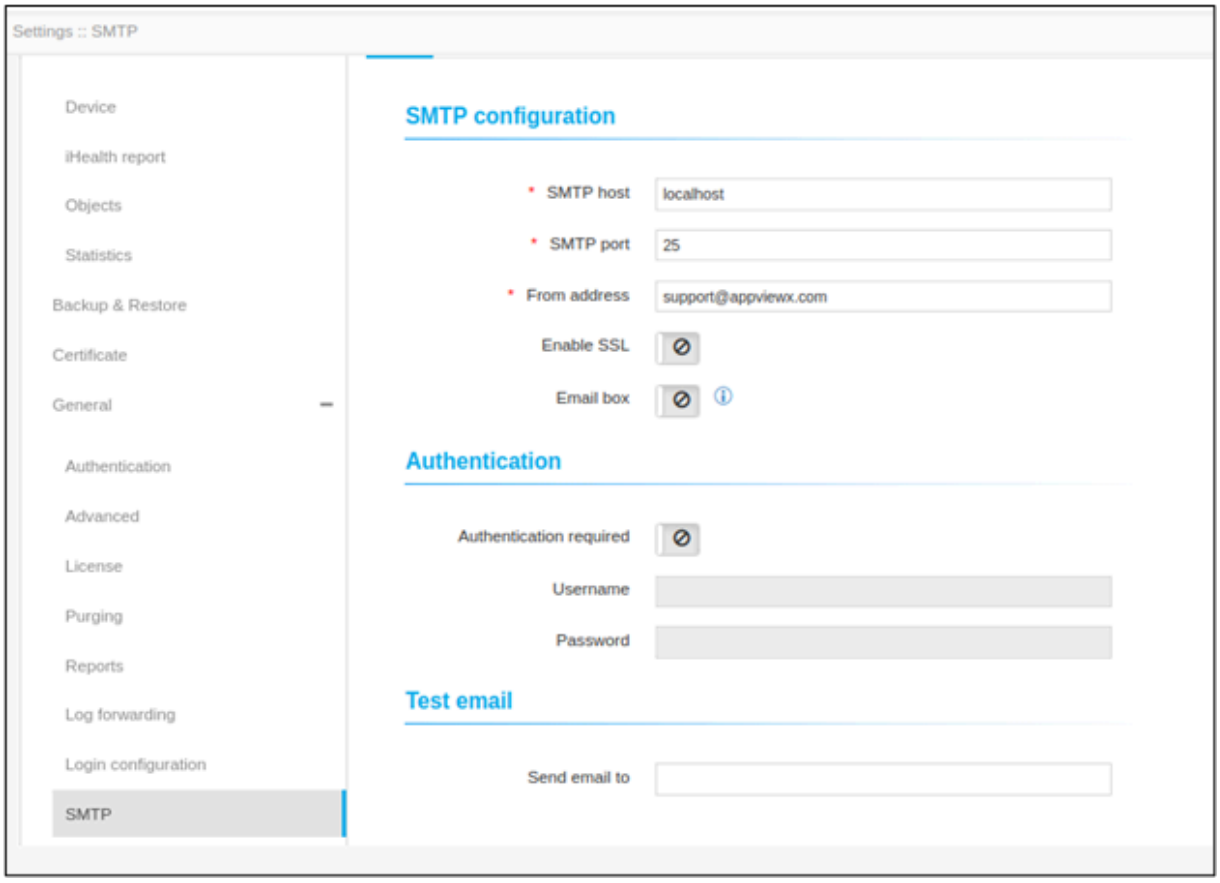


Issue in Email Notification for Configured for Backups

1. Login to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Expand the group to check the generated backups.
4. If the backup is generated and still no email notification was sent to the user.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Scenario	Possible Cause	Possible Solution
Backup Status mail is not received.	SMTP configuration is not configured.	Please update SMTP details in Menu > Settings > General > SMTP. Provide the details and check the configuration with Test email Configuration.
	Mail Id added may not be valid.	Check for the mail Id added.



Issue in Restoring a Device

1. Login to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Provide the device name.
4. Select one archive from generated backups.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

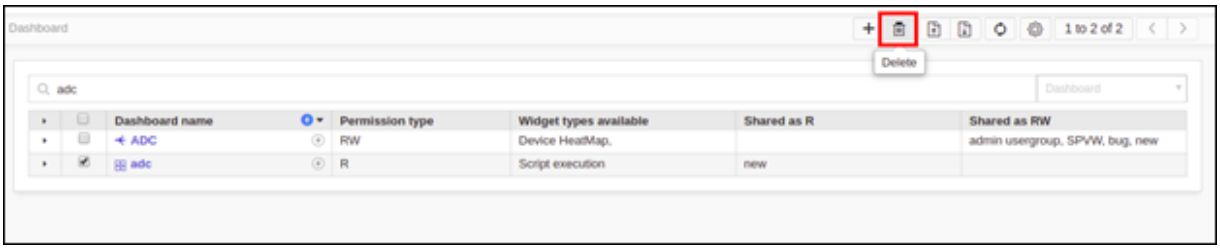
5. Click **Proceed**.
6. If a success message is not shown in the pop up, check the audit logs.

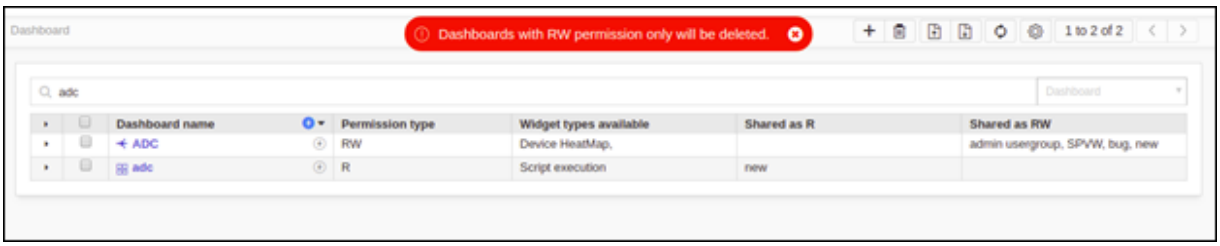
Error Scenario	Possible Cause	Possible Solution
Action cannot be performed on an unresolved device.	Device is in an unresolved state.	Check for device credentials and manage the device.
Problem in establishing communication with devices.	Device is in managed state in AppViewX. But the credentials of the device may be changed now.	Check for the device credentials and update the latest credentials.

Dashboard Inventory Action Issues

Issues in Dashboard Delete

1. Log in to AppViewX.
2. Click **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Select the dashboard(s) to delete.
4. Click **delete**.





Error Message	Possible Cause	Possible Solution
Only dashboards with R/W access will be deleted.	User only has Read permission for the dashboard and cannot delete the dashboard.	1. Update roles of user: Navigate to 'Account' -> 'User' -> 'Roles' and update user with required roles. Refer to the admin guide.

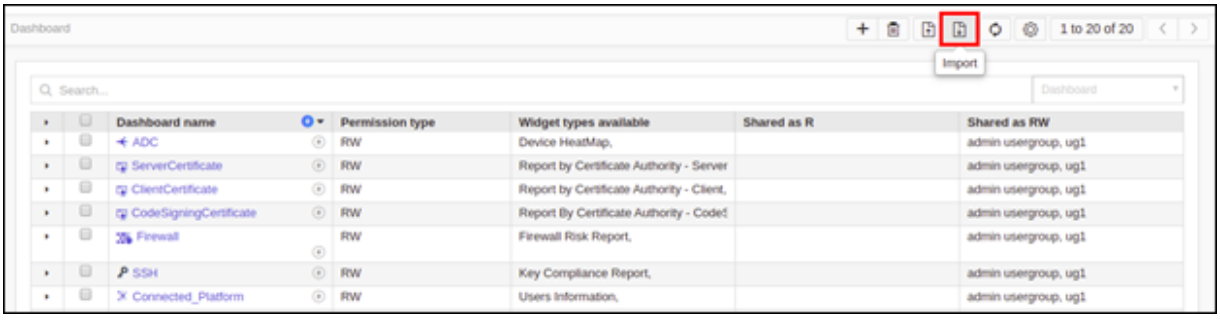
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
		2. Contact the user with RW access to the dashboard and request to 'share' roles with RW permission to the user. 3. Login with another user who has RW permission to dashboard and try again.

Issues in Dashboard Import

Process Failed by Invalid Dashboard

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To navigate to the dashboard import page, click the **import** icon.



4. Select the .zip file to import.
5. Click **Upload**.
6. Select the dashboard(s) and click **Submit**.

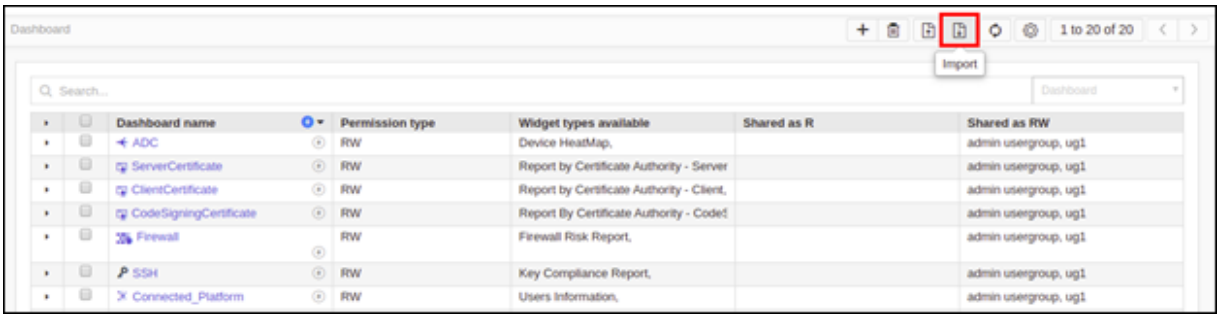


Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
Process Failed by Invalid Dashboard.	During dashboard import, the imported file has invalid entries and the user submitted the file in spite of the warning.	Navigate to 'Dashboard' -> click on 'import' icon -> under 'select file', click on 'browse' icon -> select file and click 'upload' button -> the dashboard details parsed should display a green symbol indicating valid import entries -> then click 'submit' button.

Import Zip File Validation Failed

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To navigate to the dashboard import page, click the **import** icon.



4. Select the .zip file to import.
5. Click **Upload**.



Error Message	Possible Cause	Possible Solution
Import zip file validation failed.	The imported file has invalid entries.	Import a proper zip file that has been exported from any appviewx node containing application widget.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
Widget details are invalid.	During device import, the imported file has invalid widget details.	Choose an import file that has valid widget details. Navigate to 'Dashboard' -> click on 'import' icon -> under 'select file', click on 'browse' -> select a valid import file -> click on 'upload' button -> the dashboard details parsed should display a green symbol indicating valid import entries -> then click 'submit' button. Refer to the admin guide.

Import Summary for Dashboard not Found

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To navigate to the dashboard import page, click the **import** icon.
4. Click the **Download Summary** icon.

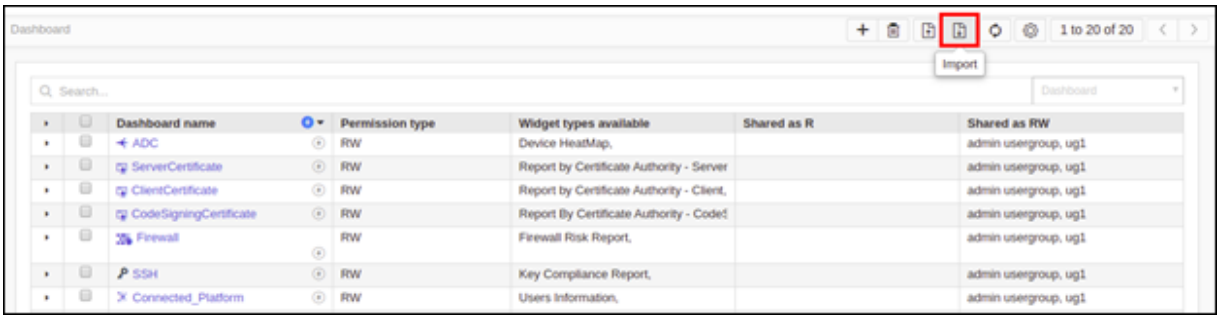


Error Message	Possible Cause	Possible Solution
Import summary for dashboard not found.	Something went wrong while populating the previous import details into the database.	Please perform operation when there is a previous successful import: Navigate to 'Dashboard' -> click on 'import' icon -> if there was a previous successful import, 'Download summary' icon will appear. click on it.

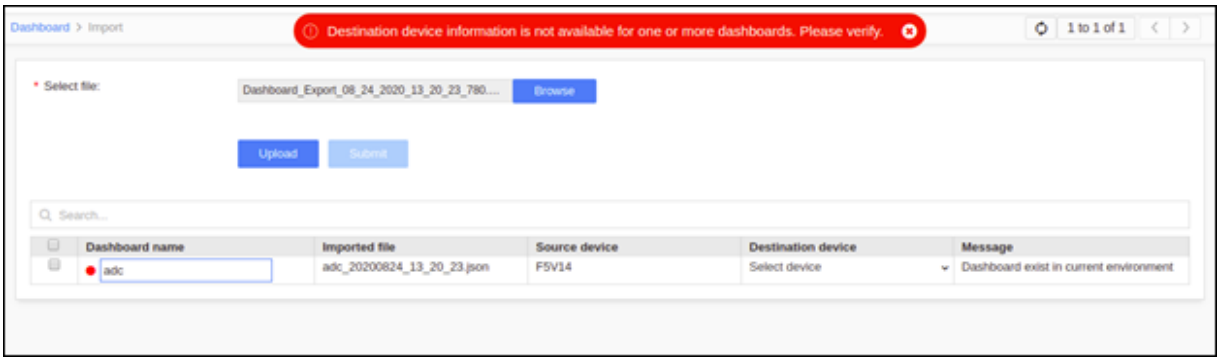
Destination device information is not available for dashboard(s).

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To navigate to the dashboard import page, click the **import** icon.

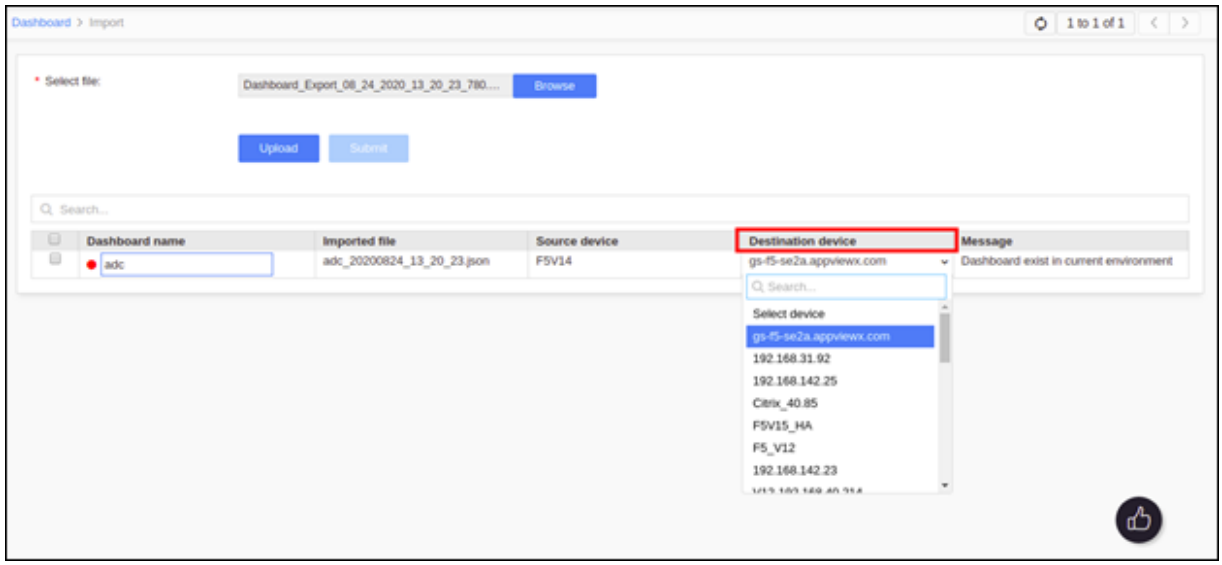
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Select the .zip file to import.
5. Click **Upload**.
6. Select the required dashboard(s).
7. Click **Submit**



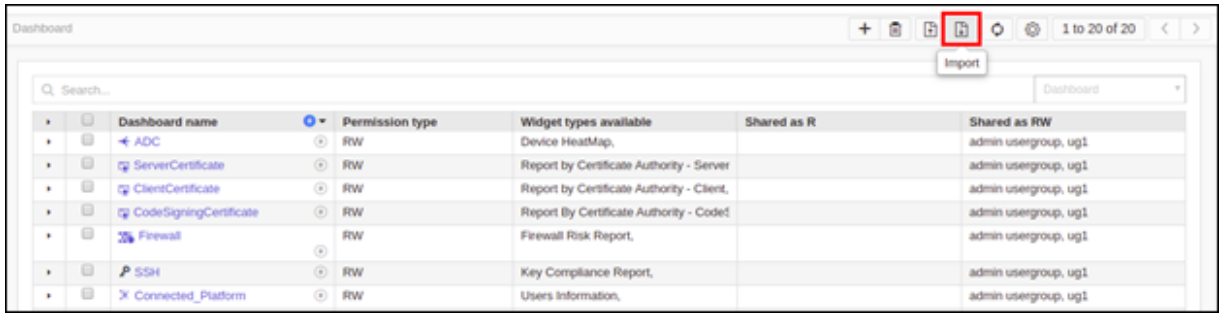
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Error Message	Possible Cause	Possible Solution
Destination device information is not available for one or more dashboard(s).	Destination device not selected for one or more dashboard(s).	Select the appropriate device in the destination device column as shown above.

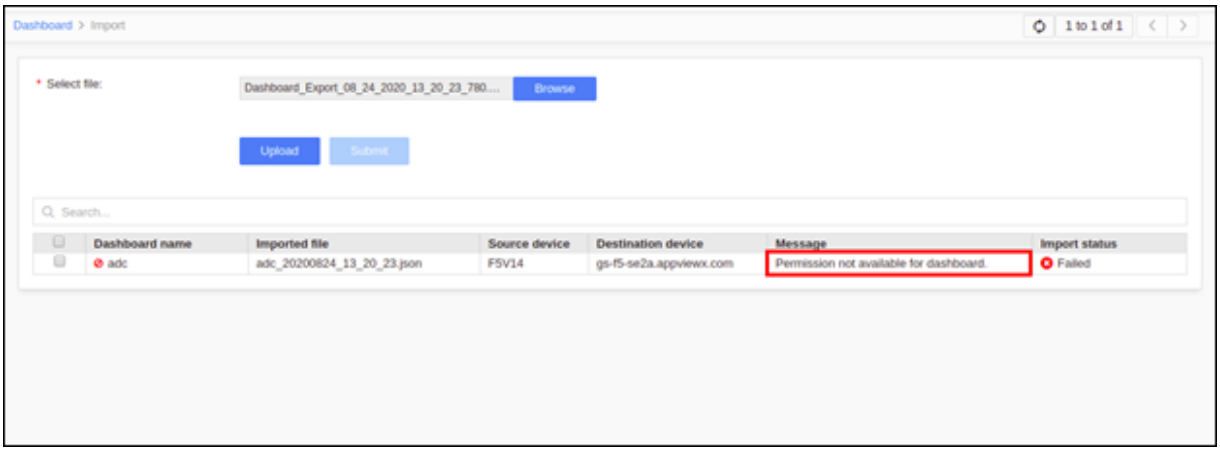
Read write permission not available for the given dashboard to import

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To navigate to the dashboard import page, click the **import** icon.



4. Select the .zip file to import.
5. Click **Upload**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



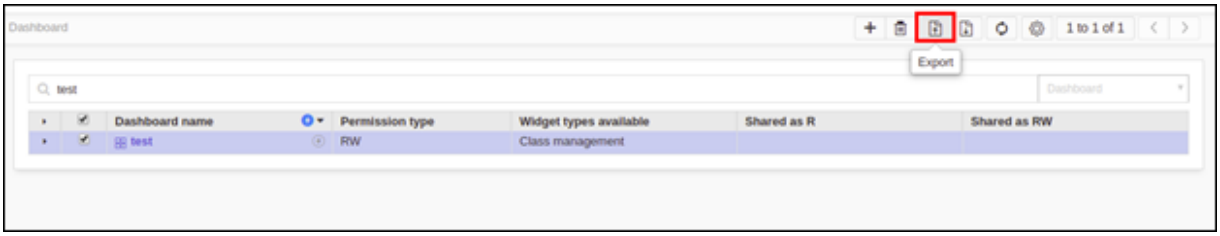
6. Select the required dashboard(s).
7. Click **Submit**.

Error Message	Possible Cause	Possible Solution
Read write permission not available for dashboard.	User only has Read permission for the dashboard and cannot export the dashboard.	Update roles of user: Navigate to 'Account' -> 'User' -> 'Roles' and update user with required roles. Refer to the admin guide.

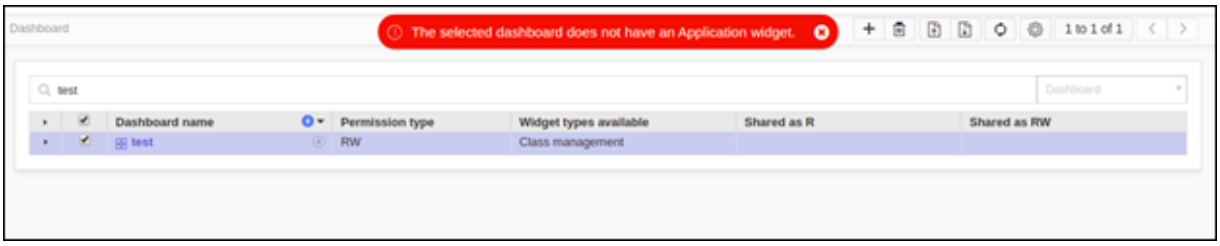
Issues in Dashboard Export

Unable to export the dashboard(s)

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Select the dashboard(s) to be exported and click the **export** icon.



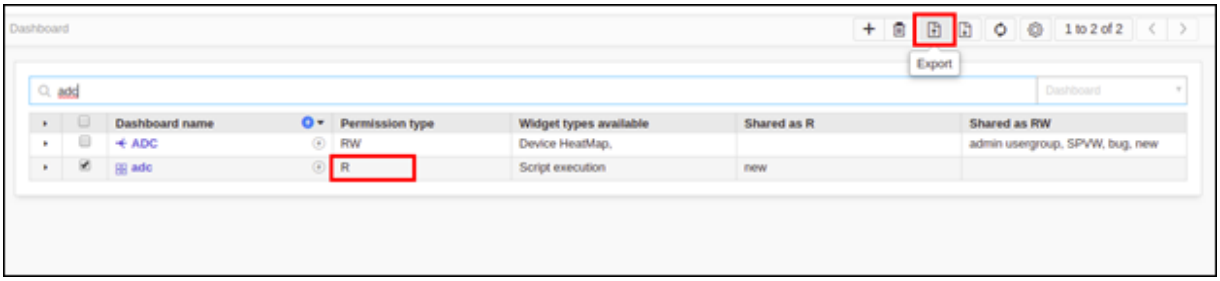
Copyright © 2025 AppViewX, Inc. All Rights Reserved.



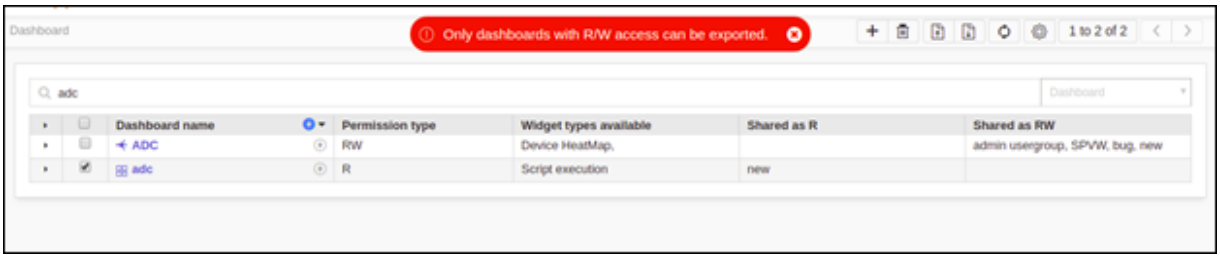
Error Message	Possible Cause	Possible Solution
Selected dashboard(s) does not have an Application Widget part of it.	One or more dashboard without application widget was selected for dashboard export.	Dashboard export operation can be performed only on the dashboard with application widget in it. Navigate to 'Dashboard' -> under dashboard grid list, select dashboard(s) with application widget in it -> click on 'export' icon.
Exception in Process Widget details.	Unable to process application widget details for export.	Select dashboard with valid application widget details. Navigate to 'Dashboard' -> under dashboard grid list, select dashboard(s) with application widget in it -> click on 'export' icon.
Exception Occured during zip file creation.	Dashboard details contain invalid content and it cannot be written to zip file during dashboard export.	Check if dashboard details contain valid entries.

Read Write Permission Unavailable for the Dashboard to Export

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Select the dashboard(s) to be exported and click the **export** icon.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



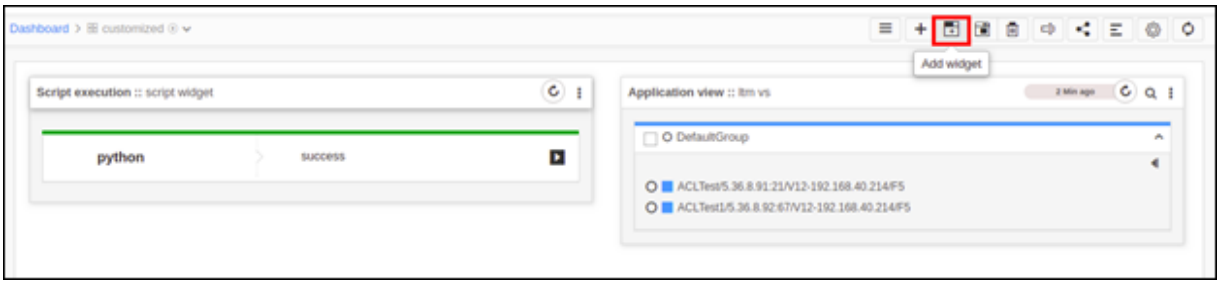
Error Message	Possible Cause	Possible Solution
Only dashboards with R/W access can be exported.	User only has Read permission for the dashboard and cannot export the dashboard.	a. Update roles of user: Navigate to 'Account' -> 'User' -> 'Roles' and update user with required roles. Refer to the admin guide. b. Contact the user with RW access to the dashboard and request to 'share' roles with RW permission to the user. c. Login with another user who has RW permission to dashboard and try again.
User not associated with any role.	During dashboard export, the user performing operation does not have any role.	Perform action using users with valid roles. Navigate to 'Dashboard' -> under dashboard grid list, select dashboard(s) to export -> click on 'export' icon.
Exception Occured during zip file creation.	File write location cannot be accessed to create an export file.	Ensure the file is written to location that has access permission.

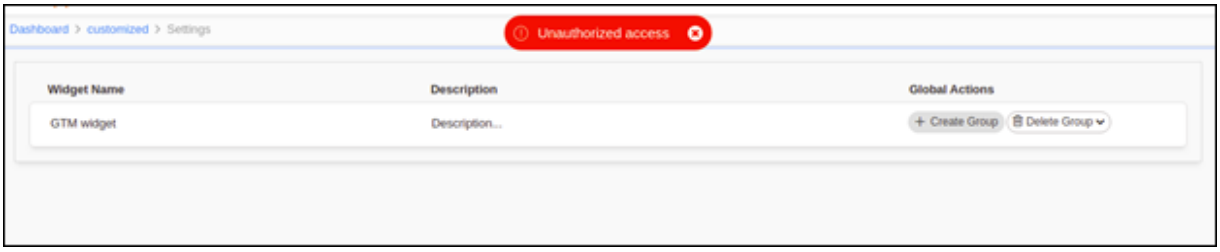
Issues in Dashboard Settings

Unauthorized Access to Add Widget(s)

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To create a new widget, click the **add widget** icon.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

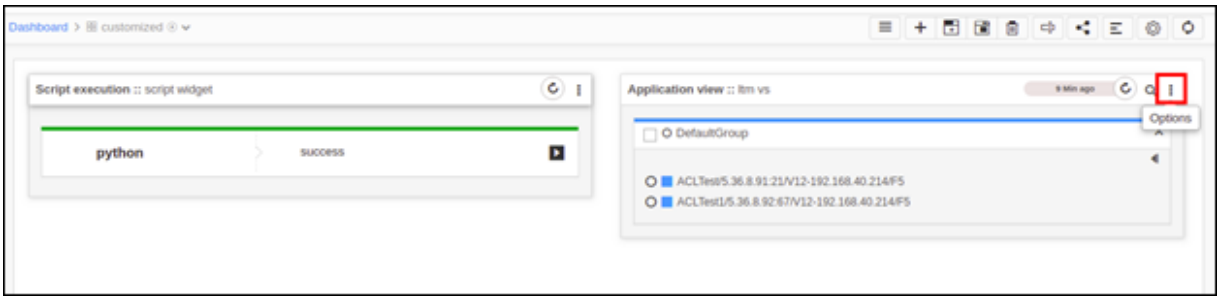




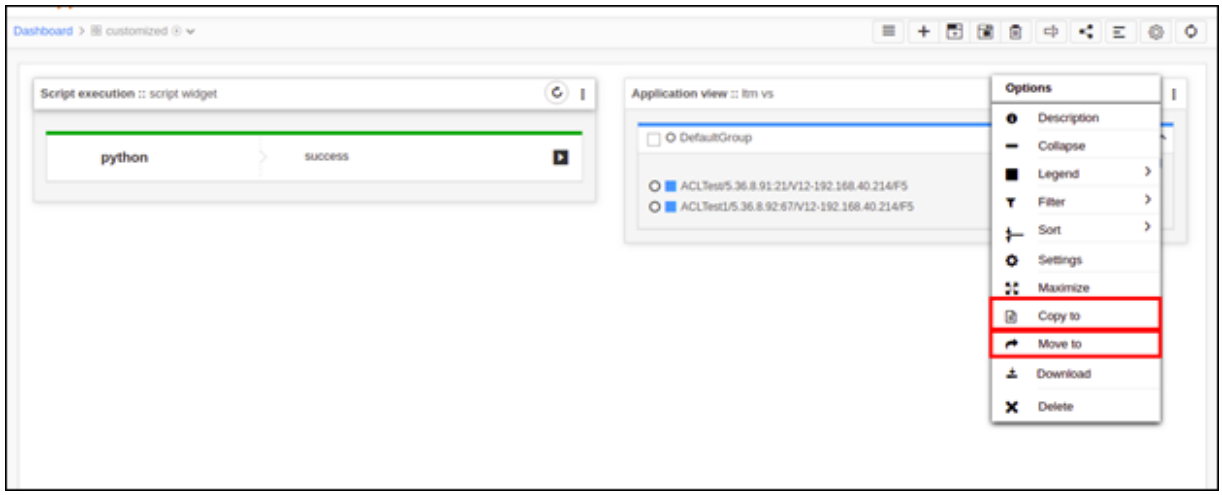
Error Message	Possible Cause	Possible Solution
Permission not available for dashboard.	User is not authorized to perform the operation on the dashboard.	a. Update permission to dashboard: Navigate to 'Account'-> 'User'-> 'Roles' and update users with roles that have R/W permission to dashboard. Refer to the admin guide. b. Login with another user who has access to the dashboard and try again.

Read Write Permission not available for dashboard to move/copy widget(s)

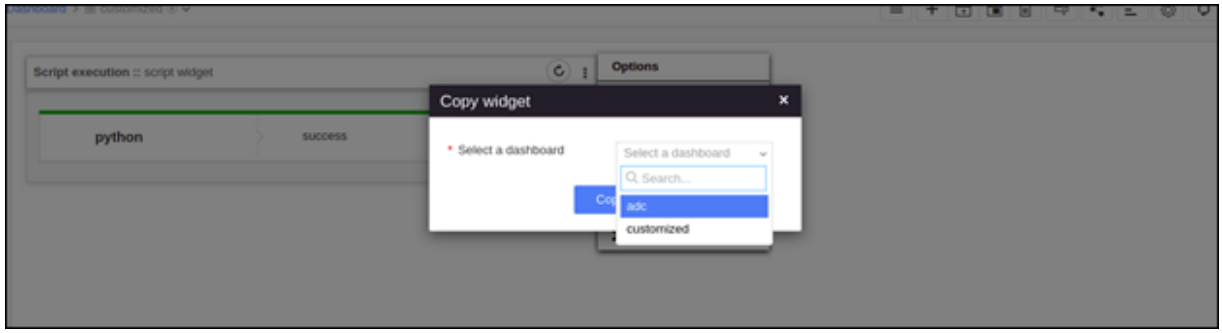
1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. To move or copy the widget, click **Options** and select the **move to/copy to** option.

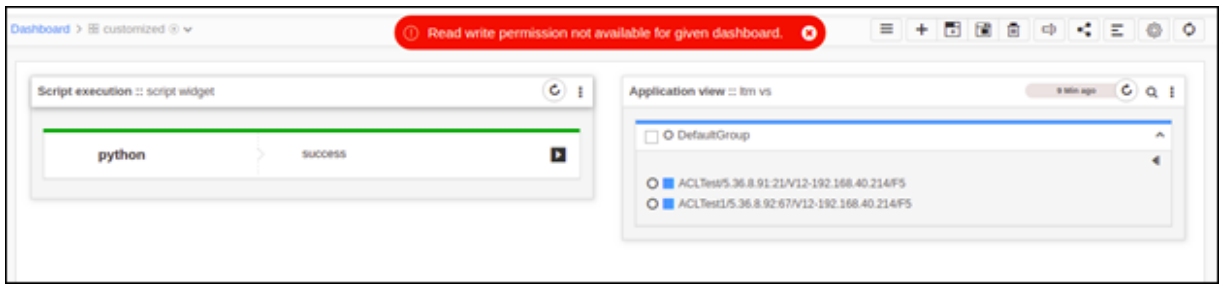


Copyright © 2025 AppViewX, Inc. All Rights Reserved.



4. Select the dashboard to which the widget should be copied or moved.





Error Message	Possible Cause	Possible Solution
Read write permission not available for given dashboard.	User only has Read permission for the dashboard and cannot perform update action on the dashboard.	Update roles of user: Navigate to 'Account' -> 'User' -> 'Roles' and update user with required roles. Refer to the admin guide. Contact the user with RW access to the dashboard and request to 'share' roles with RW permission to the user.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
	User does not have R/W permission to one of the dashboard during copy/move widget action.	Login with another user who has RW permission to dashboard and try again.

ADC Dashboard Specific Issues

Issue in Class Management - Search

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Search the ClassManagement Widget > Search for a class.

Error Message	Possible Cause	Possible Solution
Class names are not listed.	The device is Not Managed.	Trigger config fetch for the device and confirm LTM parsing in successfully completed.

Issue in Class Management - View Action

1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Search the ClassManagement Widget > Execute view Action.

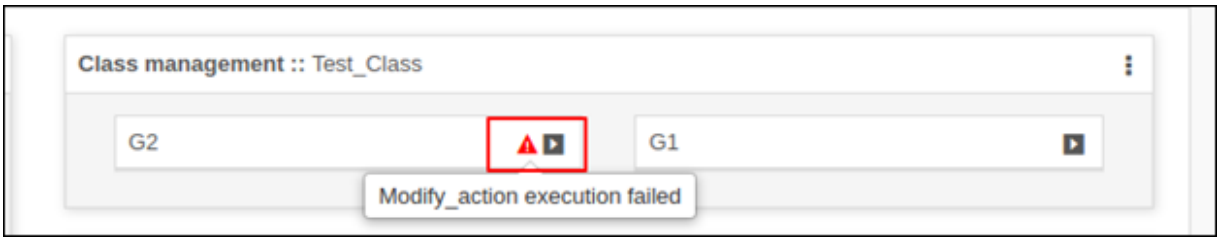
Error Message	Possible Cause	Possible Solution
The class name provided by the user does not exist.	The class object is not found on the device.	Check in the F5 device.
Unable to connect to device with the provided credentials. Displaying attributes from the database.	The device credentials might be changed.	Update the credentials for the device in Device Inventory.
Associated class not found in the device.	The class object is not found on the device.	Check in the F5 device.
Failed to retrieve class details. Refer log for more details.	There might be a problem in getting class data from the device.	Please check connectivity to the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Issue in Class Management - Modify Action

When the Modify Action is executed on a particular class, the newly added values are not reflected in the device.

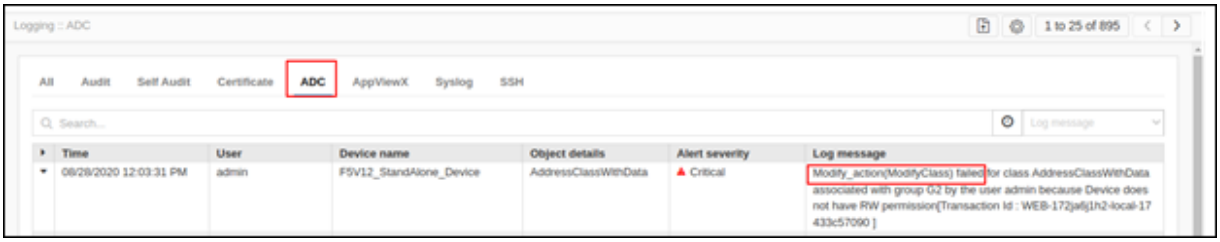
1. Log in to AppViewX.
2. Click **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Search the ClassManagement Widget > View log >and check for the Error Message for the Action failure.
4. Action Execution Failure is shown in Widget.



5. View Log shows the actual error message.

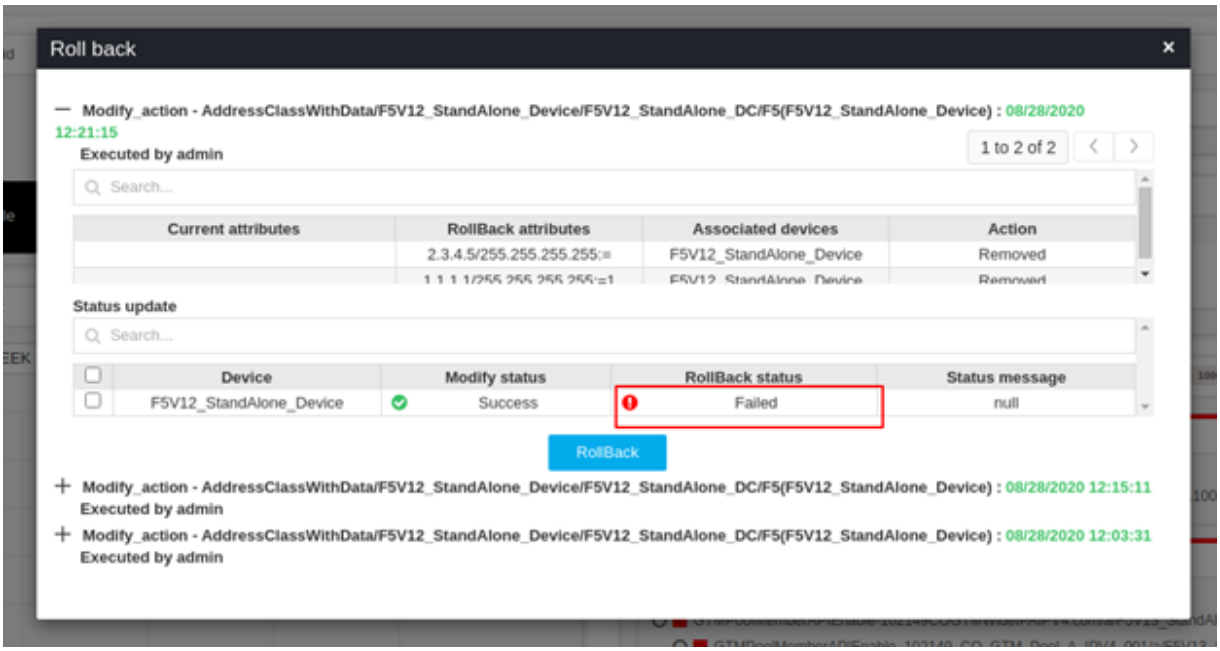


6. Navigate to **Main Menu** > **Logging** > **ADC**.



7. Search for the operation that was performed.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



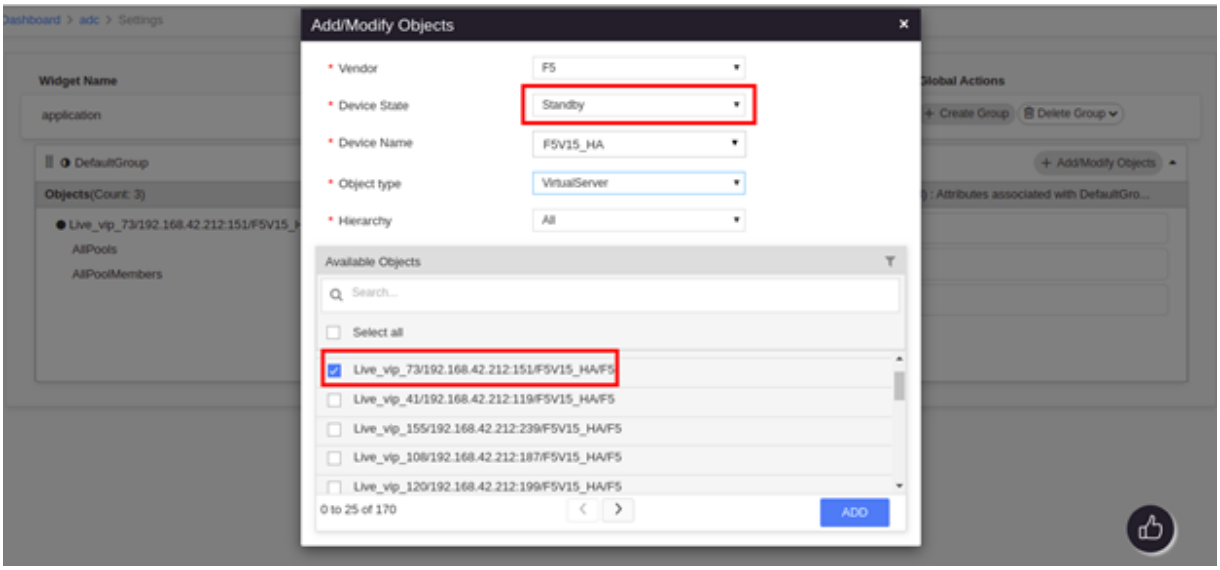
Error Message	Possible Cause	Possible Solution
(User: <userName>) Modify_action(ModifyClass) failed for class <className> associated with group <groupName> by the user <userName>	The class object is not found on the device.	Check for the presence of class in the F5 device.
Device does not have RW permission.	To execute Modify action, the user should have RW access for the device.	Check in **Menu**> **Account** > **Resources** >**Device** > Provide **RW** access for the device.
Failed to retrieve class details. Refer log for more details.	There might be a problem in getting class data from the device.	Please check connectivity to the device.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Issue in Class Management - Rollback Action

When Rollback Action is executed on a particular class, the newly added values are not removed in the device.

1. Log in to AppViewX.
2. Navigate to **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Search the ClassManagement Widget > View log > and check for the Error Message for the Action failure.



4. Navigate to **Main Menu** > **Logging** > **ADC**.
5. Search for the operation that was performed.

Error Message	Possible Cause	Possible Solution
(User: <userName>) Modify_action(Rollback) failed for class <className> associated with group <groupName> by the user <userName>	The class object is not found on the device.	Check for the presence of class in the F5 device.

Issue in Application Widget Actions

We have a set of actions that can be performed from the dashboard. For example, an object enable action can be performed by right clicking on the object as explained below.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

1. Log in to AppViewX.
2. Navigate to **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Search the Application Widget.
4. Right-click the required objects/groups.

Error Message	Possible Cause	Possible Solution
AppViewX throws No actions found error from UI	ACL permissions to RW on this object/Dashboard might not be allowed for this user role.	Check with admin whether this user has RW permissions to perform actions on this object/Device/Dashboard. If so, request for permission.
No actions are displayed when right click is performed on the object.	Actions would not have enabled on the Widget settings.	Go to the settings of the particular Widget, and check for actions being enabled below the actions tab. If not, enable the actions.
No actions are displayed when right click is performed on the object.	Enabled actions might not be applicable for the particular type of object on which action is tried to perform.	N/A

5. To determine the root cause for backup failure, check the **Audit Log**.
6. Navigate to **Main Menu > Logging > ADC**.
7. Search for the object name.

Error Scenario	Possible Cause	Possible Solution
Status fetch failure - Device communication failure	Device is in managed state in AppViewX. But the credentials of the device may be changed now.	Check for the device credentials and update the latest credentials.

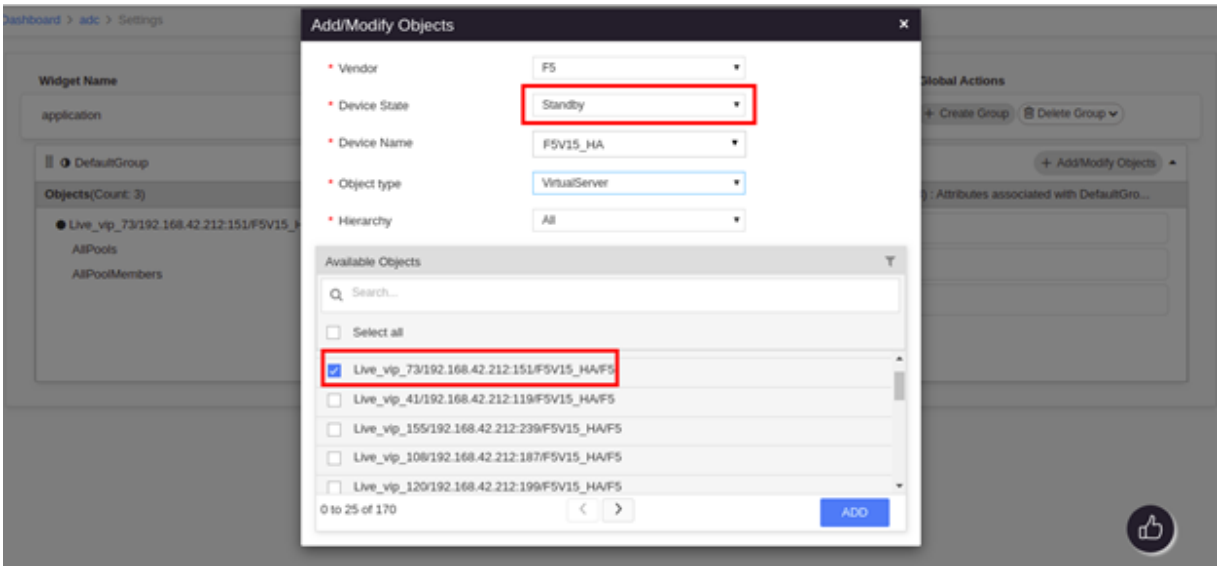
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Scenario	Possible Cause	Possible Solution
Action failed for the object - Southbound communication failed.	a. Device is in an unresolved state. b. avx_vendors plugin may not be running. c. Error while performing action.	a. Check for device credentials and manage the device. b. Check for avx_vendors plugin state. c. Check for the backend logs.
Object greyed out : The requested object not found.	Objects may be deleted from the device.	Check for the object in the F5 device.

Application Widget Settings

Configured with standby objects and it displays their corresponding active objects

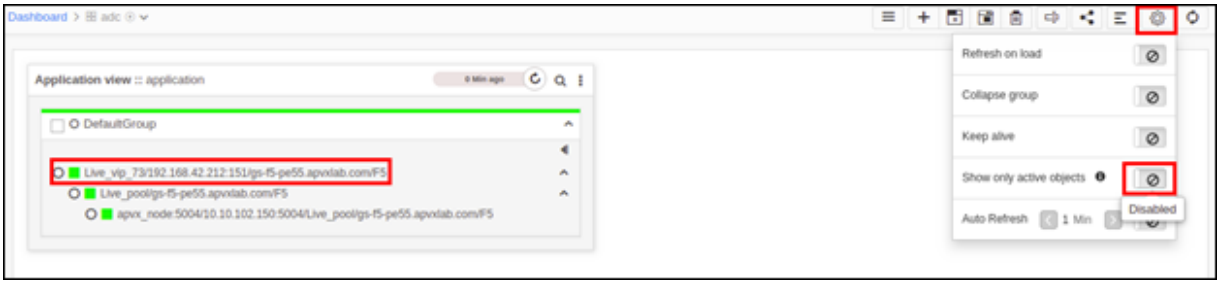
1. Log in to AppViewX.
2. Navigate to **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Create an application widget with standby objects.



It shows the active objects of the standby objects configured.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

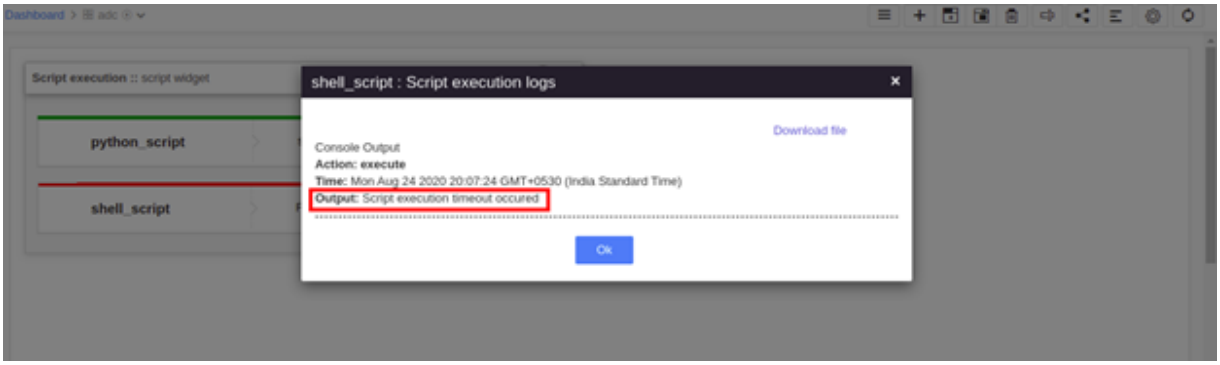
4. Click the **settings** icon and disable the **Show only active objects** option.



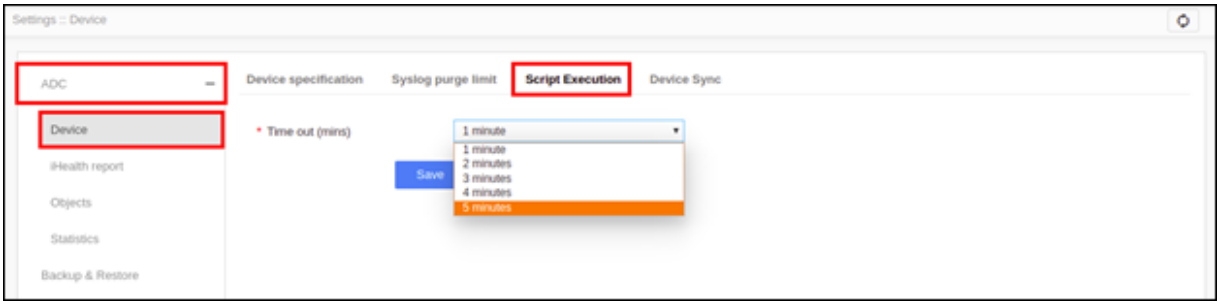
Script Execution Widget Actions

Script Execution Timeout Occurred

1. Log in to AppViewX.
2. Navigate to **Menu** > **TRAFFIC MANAGEMENT** > **Dashboards**.
3. Create a script execution widget.
4. Add a script.
5. Execute the script.



6. Navigate to **Main Menu > Settings > ADC > Device > Script Execution**



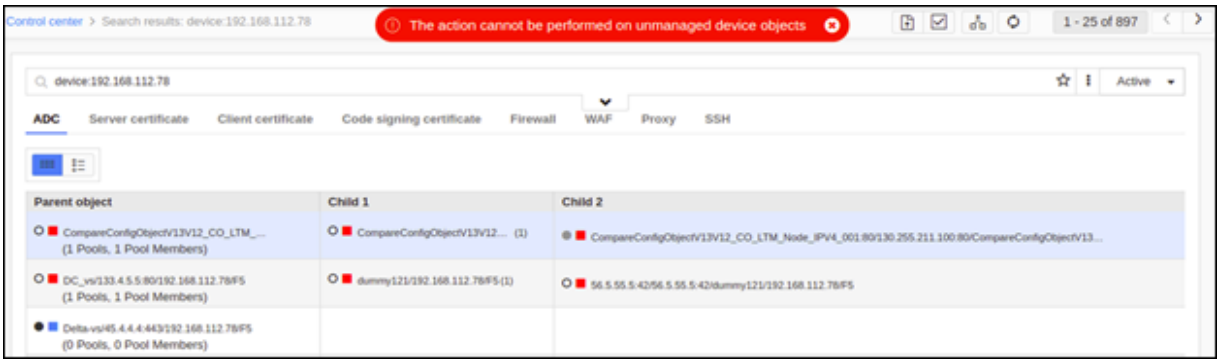
Copyright © 2025 AppViewX, Inc. All Rights Reserved.

7. Select the **Time out (mins)**.
8. Click **Save**.

Control Center Search and Action Issues

Issue while performing actions on an object

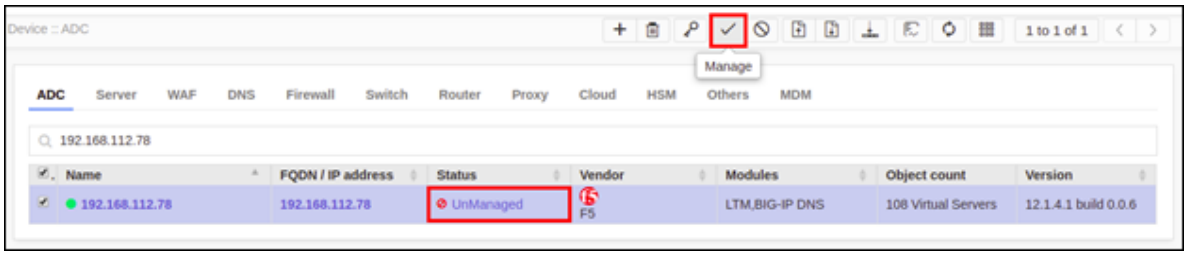
1. Action executed on an object throws an error message "*Action cannot be performed on unmanaged device object(s)*".



   a. Log in to AppViewX.

   b. Navigate to **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.

   c. Search for the device on which the action was performed.



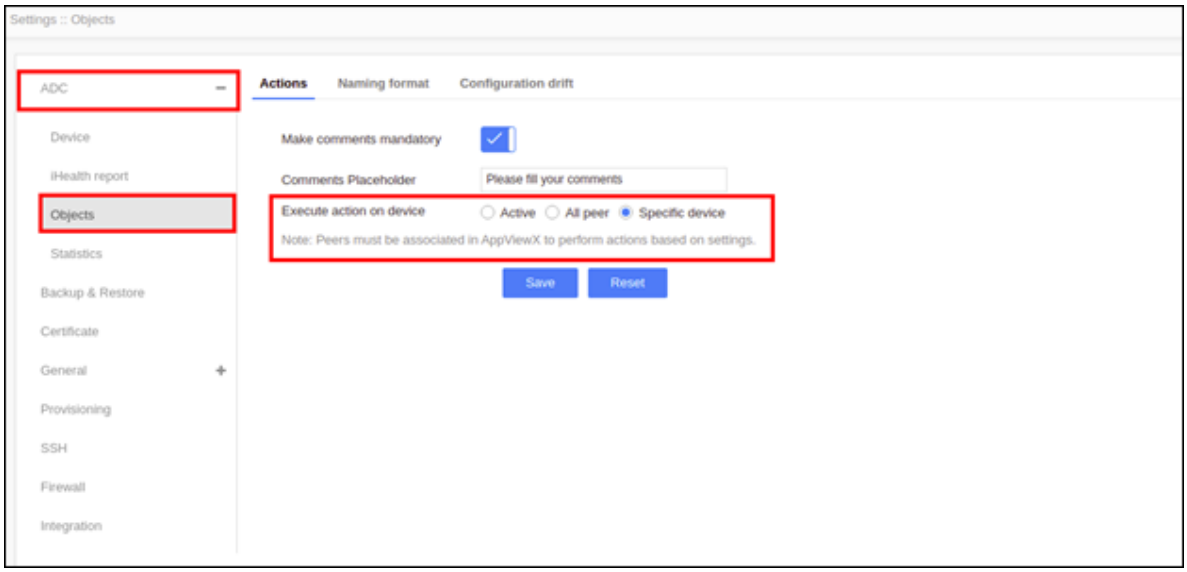
Error Message	Possible Cause	Possible Solution
Action cannot be performed on unmanaged device object(s).	Device is in an Unmanaged state.	Click the Manage icon as shown in the above image. After the device gets Managed, perform the required action on the respective object.

2. Action Executed on the Standby Object is not Reflected on the Active Object.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. Log in to AppViewX.

b. Navigate to **Main Menu > Settings > ADC > Objects > Actions**.



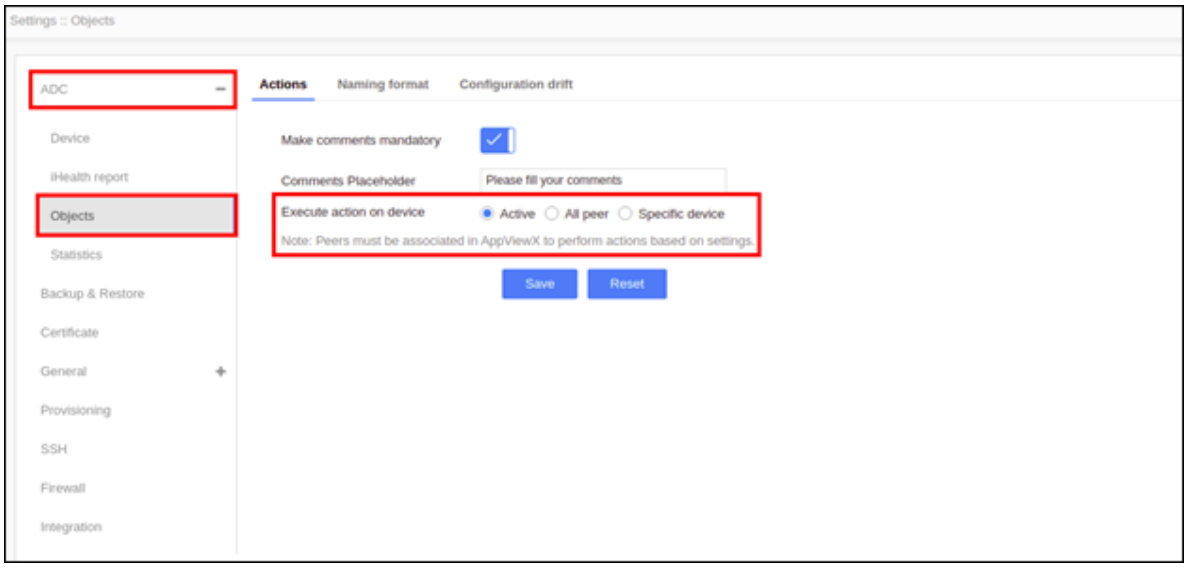
Error Message	Possible Cause	Possible Solution
No error	**Execute action on device** might be set to **Specific device**.	Change **Execute action on device** to **All Peer**and click **Save**.

3. Action is not Executed on the Standby Object Triggered

a. Log in to AppViewX.

b. Navigate to **Main Menu > Settings > ADC > Objects > Actions**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.



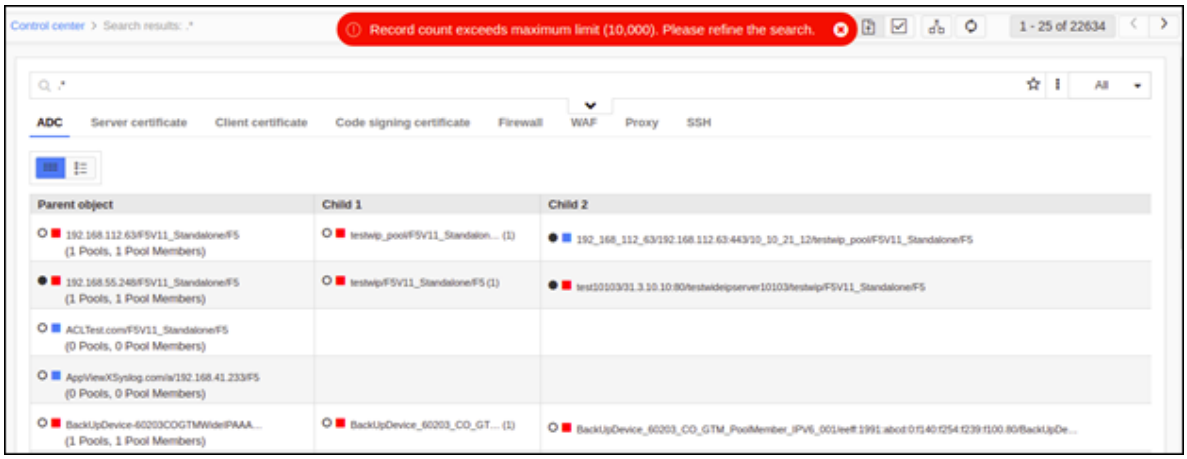
Error Message	Possible Cause	Possible Solution
No error	**Execute action on device** might be set to **Active**.	Change **Execute action on device** to **All peer**or **Specific device** and click **Save**.

Issues in Control Center Search

1. Result Exceeded the Maximum Record Size for the Control Center Export.
   a. Log in to AppViewX.
   b. Navigate to **Menu** > **ASSET MANAGEMENT** > **App Search**.
   c. Select the objects to export and click the **export** icon.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Error Message	Possible Cause	Possible Solution
Result exceeded the max record size.	Result size exceeded maximum limit. For Application View, the limit is 10000 and for Infrastructure View the limit is 300000.	Modify your search criteria to limit the search results.

2. No Content Found for Control Center Export

a. Log in to AppViewX.

b. Navigate to **Main Menu > Control center > ADC tab**.

c. Search for object(s) in the search bar and click the **export** icon.



Error Message	Possible Cause	Possible Solution
No contents found for search result.	The search criteria did not match any results.	Modify your search criteria to match results.

3. No Records Found

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. Log in to AppViewX.

b. Navigate to **Main Menu > Control center > ADC** tab.

c. Search for object(s) in the search bar.



Error Message	Possible Cause	Possible Solution
No records found are shown when I search for an object in the Control center.	The object might have been deleted in the device.	Check for the existence of the object in the device.
No records found are shown when I search for an object in the Control center.	If the object is present in the device and still it is not shown in the Control Center, the filter in the Control center page might be checked.	Check for the HA state of device(a drop down filter) in the Control center page. It will be available to the right of the search box. Change the value to **All**.
No records found are shown when I search for an object in the Control center.	ACL permissions to this object might not be allowed for the user role.	Check with the adminstrator whether the user has permissions to view the object/ device. If so, request for permission and provide a config fetch of the device.
No results found	The search criteria did not match any results.	Modify your search criteria to match results.
No results found	Search filters led to a larger result and 16mb cap limit is reached in Mongo Server.	Narrow your search criteria to limit the search results.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
No results found	User does not have permission to view the results.	Navigate to Account -> Resource -> Modify Resource -> Select Access Control -> Under Access Control tab assign permission to ADC Objects. Refer admin guide.

4. Object not Found

   a. Log in to AppViewX.

   b. Navigate to **Main Menu > Control center > ADC tab**.

   c. Search for object(s) in the search bar.

   d. Click the required object to open in the topology view.

Error Message	Possible Cause	Possible Solution
Object not found	The object(s) must have been removed after config sync.	Login to device and check if the object was deleted from the device.
Object not found	The device in which the object exists must have been deleted from AppViewX.	Navigate to 'Inventory' -> 'Device' -> 'ADC' , and check if the device has been deleted. If true, re-add device or request topology for objects in other available devices. Refer to the admin guide.
Object not found	The object(s) for which topology was requested is invalid.	1. If AppViewX is not in sync with the device, wait for midnight config sync. 2. If AppViewX is not in sync with the device, navigate to 'Inventory' -> 'Device' -> 'ADC' -> select the corresponding device -> click on 'config fetch' icon. Refer to the admin guide.

5. The Regex Expression is Invalid

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

a. Log in to AppViewX.

b. Navigate to **Main Menu > Control center > ADC**.

c. Search for object(s) in the search bar using a regex expression.



Error Message	Possible Cause	Possible Solution
The Expression is Invalid.	The keywords used for search is invalid.	Use the available keywords in the control center page. Navigate to Control center -> Click on the Expand icon near the Search bar and use one of the available search keys.
The Expression is Invalid.	The search term contains special characters ('(',')') that led to an invalid expression.	If the search term contains special characters, try to use a different filter.

6. Object Dictionary does not Contain Data for the Given Code.

a. Log in to AppViewX.

b. Navigate to **Main Menu > Control center > ADC**.

c. Search for object(s) in the search bar.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
Object dictionary does not contain data for the given code.	Script to populate the object dictionary was not executed.	Verify if AppViewX supports the object type specified by the code.
Object dictionary does not contain data for the given code.	AppViewX does not support the object type specified by the code.	Contact AppViewX support to execute the script to populate the object dictionary.

iHealth Report Generation Issues

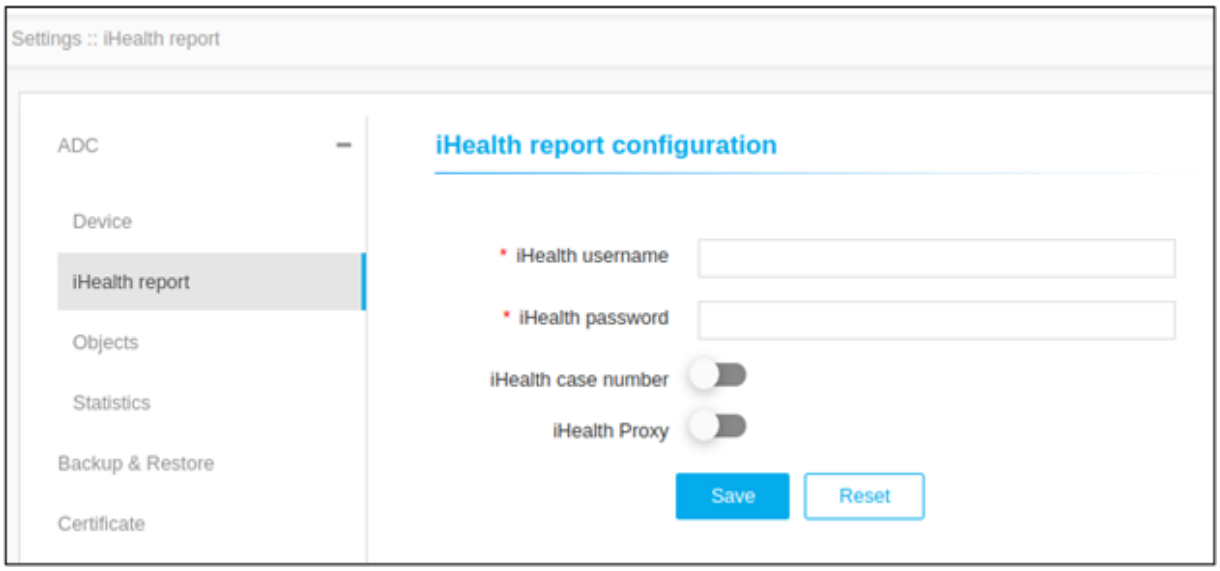
Unable to Generate iHealth Report

1. Log in to AppViewX.
2. Navigate to **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Select Device and click **Generate iHealth Report**.

Error Message	Possible Cause	Possible Solution
Please provide the valid credentials under AppViewX settings to generate the iHealth report.	iHealth settings like username, password and proxy are not configured in the **iHealth report configuration** page.	a. Navigate to **Menu > Settings > ADC > iHealth report**. b. Update **iHealth username**.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.

Error Message	Possible Cause	Possible Solution
		c. Update **iHealth password**. d. Enable **iHealth Proxy**.

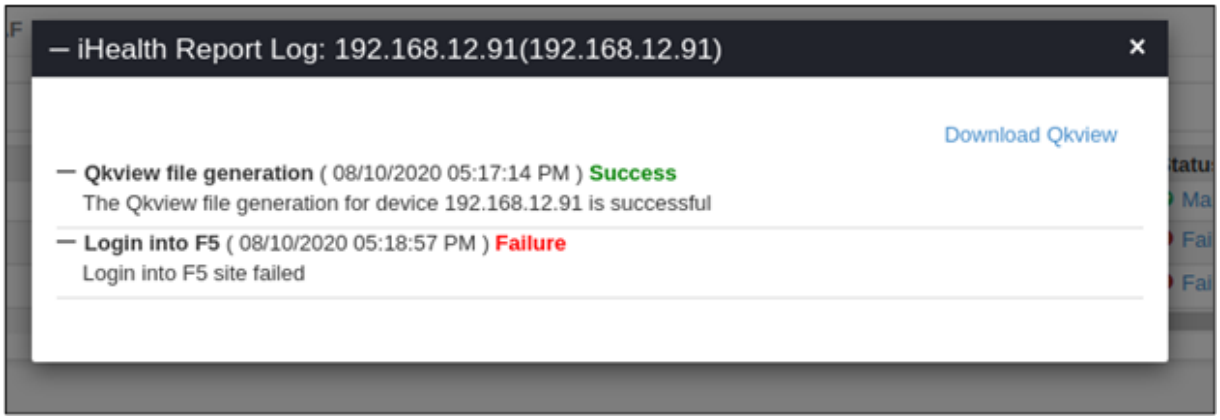


Issues in iHealth Report Generation

1. Log in to AppViewX.
2. Navigate to **Menu** > **ASSET MANAGEMENT** > **Device Inventory**.
3. Select the required device and click **Generate iHealth Report**.
4. Click the Report status to view the actual error report.



Copyright © 2025 AppViewX, Inc. All Rights Reserved.



Error Message	Possible Cause	Possible Solution
Report status is stuck in Queued state.	The `avx_vendors` plugin may be not running.	Please contact the administrator to restart the component.
Failed to connect to <device> with the given credentials.	The credentials provided for the device are wrong.	Update the credentials for the device.
Login into the F5 site failed.	Proxy is not configured or proxy credentials are updated wrongly.	a. Update the proxy values in **Settings > General > Proxy**. b. Select the ihealth Login urls in the **URL** list.

Copyright © 2025 AppViewX, Inc. All Rights Reserved.