



SSH+ Guides

Version: 2024.1.0.0

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.....	vii
Chapter 1. SSH+ Admin Guide.....	9
Introduction to SSH+.....	9
Getting Started.....	10
System Requirements.....	11
Installation Instructions.....	13
On-Premise Installation Instructions.....	13
SaaS Installation Instructions.....	17
Accessing SSH+ Features.....	22
Configuring Provision Settings.....	23
Getting Started	26
Access Requests Hub.....	28
Setting ACF and ACL Permissions.....	29
Setting ACF Permissions for Roles.....	29
Setting ACL Permissions to Infra Access Groups	30
Setting ACL Permissions to Key Compliance Groups.....	31
Adding or Deleting Regex Patterns.....	32
Adding Access to Users.....	33
Overview.....	33
Approving/Rejecting Access Requests.....	34
Glossary.....	35
Chapter 2. SSH+ User Guide.....	36
Introduction to SSH+.....	36
System Requirements.....	37
Accessing SSH+ Features.....	39
Discovering Keys.....	40
Overview.....	40

Network Scan.....	41
Managed Devices.....	50
Discovery Status.....	54
Scheduler.....	59
Managing Devices/Hosts	60
Overview.....	60
Adding Credentials.....	60
Host Inventory.....	62
Adding Server	70
Adding Cloud.....	74
Actions	79
Access Control.....	81
Overview.....	81
Requesting Access to Terminals.....	81
Access Requests Hub.....	84
Approving Access Requests.....	85
Viewing Terminal Access Control Page.....	85
Accessing Host Terminals.....	87
Configuring Provision Settings.....	88
Setting ACL Permissions to Resources.....	91
Setting ACL Permissions to Infra Access Groups	92
Setting ACL Permissions to Key Compliance Groups.....	93
Adding or Deleting Regex Patterns.....	94
Adding Infra Access Groups.....	95
Infra Access Group.....	95
Adding Infra Access Group.....	96
Removing Hosts from Infra Access Group.....	97
Viewing Infra Access Group.....	97
Managing Host Key and User Key Inventories.....	98

Overview.....	98
Key Inventory.....	99
Dashboard.....	111
Reports.....	111
Remediation Actions.....	116
Creating Key Policy and Group.....	118
Overview.....	118
Key Policy.....	118
Key Compliance Group.....	120
Creating Host Policy and Group.....	122
Creating Host Policy.....	122
Glossary.....	123
Chapter 3. SSH+ API Guide.....	124
Understanding the AppViewX SSH+ API.....	125
Authentication using a User Account.....	128
Retrieve session ID using login API.....	129
Using Session ID for further API calls.....	133
Authentication using a Service Account.....	137
Retrieve Access Token using get-service-token API.....	137
Using Access Token in the header for further API calls.....	141
Add New Host.....	144
Search Hosts.....	154
Search Host Keys.....	162
Search User Keys.....	173
Search Access Groups.....	191
SSH Create CA.....	198
SSH Download CA.....	204
Search CA.....	208
SSH Create Certificate.....	214

SSH Download KRL.....	222
SSH Get Hosts From Infra Access Group.....	227
Request access to infra access groups.....	232
Trigger Network Scan for Range of IP Addresses	237
Search Hosts from Discovery.....	247
Search User Keys From Discovery.....	256
Search Host Keys From Discovery.....	274
Fetch Discovery Status.....	286
Rotate SSH Keys.....	292
Restore SSH Keys.....	296
Rollback SSH Keys.....	301
Delete SSH Keys.....	306
Search Backup User Keys.....	310
Search Backup Host Keys.....	321
Fetch Status of an Infra Access Group Access Request.....	331
Fetch Risk Report.....	340
Fetch Allowed Principals.....	346
Remove Allowed Principals.....	352
Revoke Certificate.....	358

Preface

Revision History

Revision	Description	Date
1.0	Initial Release of AppViewX 2024.0.0 SSH+	October 2024

About the Documentation

This guide talks about the complete functionality of the AppViewX SSH Key and Host Management solution. With the help of this guide, you can manage:

- Host discovery and onboarding
- SSH key and SSH certificate discovery
- Key compliance group
- Host compliance group
- Infra access group
- Access control
- Remediation

Audience

This guide is intended for CISO, PKI Security, and Application Teams.

All the people belonging to the groups mentioned below should use this guide.

IT Operations	CISO, CIO, CTO, Server & Application Administration
Security	IAM Administration
Risk Management	Risk, Compliance, Audit Administration

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

The following text conventions are used in this document:

Convention	Description
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in the text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>codeblock</code>	Indicates commands with a paragraph, URLs, codes in examples, text that appears on the screen, or text that you enter.

Chapter 1: SSH+ Admin Guide

Introduction to SSH+

As application infrastructures grow, so do security threats. Organizations have to find newer ways for protecting their data and granting access to the right users and devices to avoid security threats and breaches. While the traditional approach used password authentication, it proved to be insecure. This is where AppViewX SSH+ comes into play.

AppViewX SSH+ is a fully-automated application infra-access management and SSH key lifecycle management solution that allows you to centrally discover, manage, and protect SSH keys with access across hybrid multi-cloud environments. It also helps simplify access management, stay compliant and mitigate risks with SSH+.

AppViewX offers visibility and SSH access management across traditional on-premises data centers and cloud-hosted infrastructures.

Risks of improper SSH Management

Since there is no governing body to regulate the use of SSH keys, there is an element of risk involved. As SSH keys are generated on a need basis, several keys may be discarded and left unmanaged when they are no longer of use. Without an inventory, managing these keys and revoking their access pose a security threat to large organizations for potential back-door entry into the network, data theft, or breaches.

Improper SSH key management can lead to unauthorized access, compliance violations, identity and access management issues, data breaches, operational disruption, and reputation damage. To mitigate these risks, organizations should implement proper SSH key management practices, including secure key storage, regular key rotation, and access controls.

What Enterprises Need

AppViewX conducted multiple surveys to identify the core features and functionality needed to address SSH management challenges. SSH and Identity and Access Management (IAM) Administrators highlighted the following requirements:

- Discover keys from standard and non-standard locations
- Identify and report non-standard and non-compliant keys
- Visibility of keys and the users of these keys
- Revoke access to non-compliant and non-standard keys

- Rotation and distribution of keys
- Self-service SSH access requests
- Support for cloud and legacy on-premise infrastructure
- Centralized SSH Certificate Authority

How AppViewX Can Help

AppViewX SSH+ key lifecycle management is a fully automated solution that discovers and manages enterprise SSH infrastructure. It can identify and mitigate risks associated with poorly managed passwordless access management.

AppViewX SSH+ features include:

- **Centralized Discovery and Visibility**
 - The solution offers on-demand scans to discover SSH keys across multi-vendor, hybrid network infrastructures, and map trust relationships to determine access privileges.
 - The consolidated inventory provides a central console to view and manage all SSH keys and hosts.
- **Risk Scorecard and One-Click Remediation**
 - The solution proactively identifies and remediates risks associated with inactive, weak, orphan, or suspicious keys using an intuitive SSH scorecard dashboard.
 - The one-click remediation feature enables instant deletion or regeneration of keys.

Getting Started

Know your Deployments for SSH+

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.



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.](#)



System Requirements

SSH within the Application Infrastructure

Application infrastructure refers to all the components required to deliver an application and its functions and services to the customer. Although each application is unique, certain common components can be identified that are typically implemented to support application capabilities and service delivery.

One of the most common components of a typical application infrastructure is Linux hosts.

Modern application infrastructure leverages multiple hosting platforms such as on-premise data centers, private clouds, and public clouds offered by third-party hosting providers.

The application infrastructure components communicate with each other to enable service delivery. This can leverage SSH communication. Additionally, administrators of these application infrastructures will need to SSH into these hosts to perform maintenance.

Additionally, security and audit compliance requirements necessitate constant awareness of who has access to what and the maintenance of best security practices for the communications.


Hardware

Ensure that you have, at minimum, the following hardware with the given specifications before proceeding with the installation:

• Single Node Deployment Requirements

Node	Bare Minimum		
	CPU	RAM	Hard Disk Space
Single node	8	32GB	500GB

• Multi-Node Deployment Requirements

Node	Bare Minimum		
	CPU	RAM	Hard Disk Space
Multi-node (master node)	4	4GB	100GB
<div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: One node for a single master installation and a minimum of three nodes for multi-master installation. </div>			
Multi-node (worker node)	8	32GB	500GB

• Platform Bare Minimum Requirements

Supported Virtualization Platforms	Versions	vCPU	RAM	HDD
VM Server, VMware ESXi	5.5 or later	8v	32GB	1TB

Operating System

Both single node and multi-node installations of AppViewX are supported on the following operating systems:

- RHEL 8.5
- RHEL 8.6
- RHEL 8.7
- Ubuntu 20.04

Browser

Following is the browser requirements to use the AppViewX SSH+ node:

Browser	Version
Firefox	v74.0.1 (64-bit) or later
Google Chrome	v85.0.4183.83 (64-bit) or later

Installation Instructions

On-Premise Installation Instructions

This chapter provides the step-by-step instruction for on-premise deployment.

On-Premise Installation Steps

Step No	Step Name	Mandatory
1	Working with Prerequisites	Yes
2	Configuring Firewall Ports	Yes
3	Configuring Elevated Access	Yes

On-Premise Installation Steps (continued)

Step No	Step Name	Mandatory
4	Running the Prerequisite Tool	Yes
5	Deploying the AppViewX Virtual Appliance	No
6	Performing a Single Node or Standalone Installation	No
7	Performing a Multi-node or High Availability Installation	No
8	Configuring the appviewx.conf file	Yes
9	Configuring POD and Service IP CIDR	No
10	Verifying the Installation	Yes
11	Activating SSH	Yes
12	Accessing the AppViewX Graphical User Interface	Yes
13	Adding Third-party Libraries	No

Activating SSH+

License Management Software tracks software installed throughout the enterprise and ensures legal licenses for its usage. The software helps you to obtain the license key, upload the license key, and troubleshoot the license issues. License management is an essential element of software asset management (SAM).

To access the application for SSH+, send an email to help@appviewx.com with the hostname of the node in which the application is installed.

Uninstalling AppViewX

Users can uninstall AppViewX when they want to migrate into another environment. They can also uninstall AppViewX when it is no longer required.

To uninstall an application package safely:

1. Open the terminal window.
2. To navigate to the **appviewx_kubernetes** directory, execute the following command:

```
cd /home/appviewx/appviewx_kubernetes/scripts/uninstall
```

3. To start the uninstallation process, execute the following command:

```
/uninstall.sh
```

4. Enter the node's credentials when prompted.

5. Reboot all the nodes after completion of the AppViewX uninstallation.

Troubleshooting

Whenever the AppViewX installation fails, you will get an error stating that some script execution failed.

• Prerequisites not met

Please check for all the items below.

- port not opened
- insufficient disk/CPU
- time not in sync
- packages not found
- hostname incorrect in configuration

• Error while installing the docker

If a customer brings in a custom OS, the Linux packages that AppViewX includes with the installer may not be compatible with the OS. In such situations, you may need to install the appropriate package to continue. This can be observed from the log messages that indicate an error while installing a package.

• Error while installing the docker

Occasionally, we have observed intermittent errors from the OS during the installation of Docker. If you encounter an error at this stage, please attempt to uninstall the application, reboot all nodes, and then proceed with the installation.

• Docker gets uninstalled from the CAGateway

Root cause: Although we removed the "uninstall docker" commands from our scripts, we discovered that Docker relies on containerd, which is used as a runtime in the product. The scripts also include steps to remove containerd in the install, uninstall, and upgrade scripts, which cannot be avoided. This ultimately results in the removal of Docker as well. Additionally, the containerd version used in the product conflicts with the pre-existing containerd version of Docker on the server.

Docker and the AppViewX application cannot co-exist in the same server as it is tightly coupled with containerd. The manually installed docker will be removed during every maintenance activity such as install, uninstall and infra upgrade.

- **Context deadline exceeded in consul after the FP3 patching process**

For setups with high network latency or slow I/O, after the FP3 patch process, the consul may be stuck in 1/2 stage, causing the vault to go in a crash loop back. If you encounter this, check the consul logs using the command

```
kubectl logs consul-consul-server-0 -n avx
```

If the logs specify “**context deadline exceeded**,” then increase the timeout in consul by the following steps:

1. Navigate to `<installer location>/appviewx_kubernetes/yaml/appviewx_vault/consul/chart/vaules.yaml`
2. Edit **consulAPITimeout: 5s** (old value) to **consulAPITimeout: 10s** (new value)
3. Save the changes.



Note:

Increase this timeout only based on the latency.

- **Error while initializing the kube master/worker**

In certain cases, when uninstallation does not clean up the data properly, we may observe errors while initializing kube master and worker. In such cases, perform an uninstall, reboot all the nodes and then go ahead with the install. Additionally, there are cases where the installation fails due to port connectivity issues. If a failure occurs in this stage, check if ports 6443, 10250, 2379 and 2380 are opened properly.

- **Error while initializing the mongodb chart**

This specific error occurs after a timeout of 5 minutes to initialize the mongodb charts. This error occurs when the pods are not able to communicate between themselves. Use the following commands to verify that:

```
kubectl describe statefulset -n avx mongo-shardeddb
```

For any connectivity issues, the output of this command will display the specific error stating connection timed out.

- **Node is enabled with IPv6 but the application is not.**

Verify the output of the command:

```
ifconfig | grep -i inet6
```

If an IPv6 address is displayed, it is necessary to enable IPv6 in the appviewx.conf file. Failure to do so may result in communication issues.

- **IP in IP tunneling is not enabled**

If the IP in IP traffic is disabled, which means that the IPv4 protocol is not permitted, we will encounter the same problem. The prerequisite check script does not identify this, so we need to verify it separately to confirm.

- **Error while installing the AppViewX plugins**

If an error occurs during the installation of AppViewX plugins, it is likely due to an error in the configuration file. You may observe an error such as `Upload failed: scp`, in such cases re-trigger `plugins_install.sh` to install the plugins. Likewise, ensure to review the configuration file carefully and proceed with the execution of `plugins_install.sh` to install only the plugins.

Migrating CentOS to Ubuntu/RHEL

For detailed instructions, refer to [Migrating CentOS to Ubuntu/RHEL](#).

SaaS Installation Instructions

For detailed instructions, refer to [AppViewX SaaS Setup Guides](#).

AppViewX Software as a Service

For more information, refer to [AppViewX Software as a Service](#) and [SaaS Architecture Guide](#).

AppViewX SaaS Onboarding and Getting Started Guide

For more information, refer to [AppViewX SaaS Onboarding and Getting Started Guide](#).

Features of the AppViewX Cloud Connector

For more information, refer to [Features of the AppViewX Cloud Connector](#).

System Requirements

For more information, refer to System Requirements for [Setting up the AppViewX Cloud Connector](#).

Setting Up the AppViewX Cloud Connector

- **Setting Up the AppViewX Cloud Connector:** For more details, click [here](#).
- **Setting up the AppViewX Cloud Connector via OVA using the Automated Script:** For more details, click [here](#).
- **Setting up the AppViewX Cloud Connector via OVA using AppViewX User Interface (Manual):** For more details, click [here](#).

Installing the AppViewX Cloud Connector



Note:

The following steps assume that:

- All system prerequisites are fulfilled by the host machine.
- The AppViewX Cloud Connector installer (downloaded in the above step) is securely copied via SCP/SFTP to the host machine where the AppViewX Cloud Connector is to be installed.

1. To extract the installer, from the downloaded package, extract the tar.gz file using the command given below: `tar -zxvf <filename>.tar.gz`

For example: `tar -zxvf pesrv07-test-94-99-appviewx-appviewx-net-cloud-connector.tar.gz`

2. On the node where the AppViewX Cloud Connector agent will be installed, from the extracted installation package, run the `./install.sh` script.

The script will check if the installation prerequisites for the AppViewX Cloud Connector have been fulfilled.



Note:

Ensure that the license file is placed in the same location as the `install.sh` script. If the license file is placed in another location, run the `install.sh` script using the following command:

```
./install.sh <complete path of the license file with the filename>
```

On successful verification of the prerequisites, you will be prompted to specify if you want to manage f5 BIG-IP devices and if you need auto-enrollment of the certificates.

3. Enter the required input value:

**Important:**

If you choose to **not enable** any of the following features, you will have to reinstall the AppViewX Cloud Connector to enable them later.

- a. If you want to manage f5 BIG-IP devices, enter **y/n** for yes/no, respectively.
- b. Choose from the available installation modes for configuring the cloud connector:
 - **Standard mode:** This is the default installation mode, enabling basic Cloud Connector features, including Auto enrollment gateways for certificate management.
 - **Advanced mode:** In this mode, you gain access to advanced functionalities such as managing F5 devices, receiving Syslog from endpoints, and SSH Web Terminal for SSH+.

Read the descriptions for the standard and advanced modes, and enter your choice. (Enter 1 for standard mode and 2 for advanced mode.)

- c. If you need auto-enrollment of the certificate using one of the following supported auto-enrollment protocols, enter **y/n** for yes/no, respectively.
 - If you choose **y** (yes) here, enter the required protocol(s) name. SSH server is installed.
 - If you choose **n**(no), you will see this prompt:

Do you want to enable SSH Terminal Server for using SSH terminal usecase (y/n)?

If you choose y, SSH server is installed.

- d. If you want to enable Syslog receiver for a near-real time configuration updates from the devices, enter **y/n** for yes/no, respectively. For configuring Syslog reception, refer to Platform User guide section, [Syslog Reception](#).

In case you have an older version of AppViewX on cloud and want to make use of Syslog capabilities for ADC, you must manually activate the Syslog flag by setting `SYSLOG_ENABLED=true` in the path `ccpath/deps/properties`.

4. Enter the sudo password.

After the relevant details have been entered, the installation proceeds. Installation logs, according to the outcome of the installation, are displayed.

Given below are sample installation logs:

```
Loaded image: rancher/k3s:v1.23.3-k3s1
Loaded image: rancher/k3d-tools:5.2.2
Loaded image: rancher/mirrored-pause:3.6
```

```
[36mINFO[0m[0000] [SimpleConfig] Hostnetwork selected - disabling injection of docker host into the cluster, server load balancer and setting the api port to
the k3s default
[33mWARN[0m[0000] No node filter specified
[33mWARN[0m[0000] No node filter specified
[33mWARN[0m[0000] No node filter specified
[36mINFO[0m[0000] Prep: Network
[36mINFO[0m[0000] Re-using existing network 'host' (8bebb4ae61001f74487d0aa6b315396405d0127c938da1206614d113295ae139)
[36mINFO[0m[0000] Created volume 'k3d-cc-images'
[36mINFO[0m[0000] Starting new tools node...
[36mINFO[0m[0000] Starting Node 'k3d-cc-tools'
[36mINFO[0m[0001] Creating node 'k3d-cc-server-0'
[36mINFO[0m[0001] Using the k3d-tools node to gather environment information
[36mINFO[0m[0001] Starting cluster 'cc'
[36mINFO[0m[0001] Starting servers...
[36mINFO[0m[0001] Starting Node 'k3d-cc-server-0'
[36mINFO[0m[0033] All agents already running.
[36mINFO[0m[0033] All helpers already running.
[36mINFO[0m[0033] Cluster 'cc' created successfully!
[36mINFO[0m[0034] You can now use it like this:
kubectf cluster-info
Cluster setup is completed. Will start the deployment shortly...
Importing the required images...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images ['/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/avx-mid-server-base-22.1.0.0.tar'] into node
'k3d-cc-server-0'...
[36mINFO[0m[0024] Successfully imported image(s)
[36mINFO[0m[0024] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images ['/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/k3d-tools-5.2.2.tar'] into node 'k3d-cc-server-0'...
[36mINFO[0m[0005] Successfully imported image(s)
[36mINFO[0m[0005] Successfully imported 1 image(s) into 1 cluster(s)
Import in progress...
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
```

```

[36mINFO[0m[0000] Importing images [/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-coredns-coredns-1.8.6.tar] into
node 'k3d-cc-server-0'...
[36mINFO[0m[0007] Successfully imported image(s)
[36mINFO[0m[0007] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images [/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-local-path-provisioner-v0.0.21.tar] into node
'k3d-cc-server-0'...
[36mINFO[0m[0004] Successfully imported image(s)
[36mINFO[0m[0004] Successfully imported 1 image(s) into 1 cluster(s)
[36mINFO[0m[0000] Importing image(s) into cluster 'cc'
[36mINFO[0m[0000] Importing images from 1 tarball(s)...
[36mINFO[0m[0000] Importing images [/home/appviewx/CCTEST/deps/tools/mid-server-docker-image/rancher-mirrored-pause-3.6.tar] into node
'k3d-cc-server-0'...
[36mINFO[0m[0003] Successfully imported image(s)
[36mINFO[0m[0003] Successfully imported 1 image(s) into 1 cluster(s)
Deploying the Cloud Connector...
NAME: avx-mid-server-starter
LAST DEPLOYED: Mon May 30 15:51:13 2022
NAMESPACE: cc
STATUS: deployed
REVISION: 1
NOTES:
1. It may take a couple of minutes for the Cloud Connector to be up.

kubectl get pod --namespace cc
*****
* Congratulations!!! The installation completed successfully. *
* Please wait till the Cloud Connector is up and running. *
*****
(1%) Cloud Connector status: Running
[32m Cloud Connector is up and running. (B[m

```







Troubleshooting:

For installation errors, refer to the [Troubleshooting](#) section.

The AppViewX Cloud Connector consists of two important components—the starter plugin and the platform. The starter plugin component is installed along with the AppViewX Cloud Connector, in the same installation process.

When installed, the starter plugin is used to initiate the download of the platform component. The platform component is used to host business use cases related to the AppViewX Cloud Connector.

When the platform component download is in progress, it is indicated by the  symbol prefixed to the platform component version number in the AppViewX Cloud Connector inventory details

 21.1.0.0 . A completed download/upgrade is indicated by the  symbol in the same location
 21.1.0.1 .

**Note:**

Based on the internet bandwidth and the number of cloud connectors being installed, the downloading of the cloud connector may vary between 5 to 15 minutes.

Prerequisites for Managing ADC Devices

For more information, refer to [Prerequisites for Managing ADC Devices](#).

Installing the AppViewX Windows Gateway

For more information, refer to [Installing the AppViewX Windows Gateway](#).

Managing the AppViewX Cloud Connector

For more information, refer to [Managing the AppViewX Cloud Connector](#).


Troubleshooting the AppViewX Cloud Connector

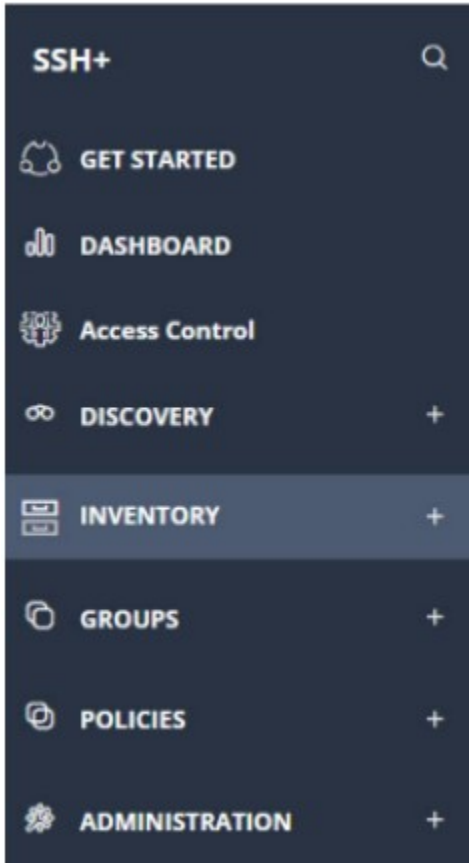
For more information, refer to [Troubleshooting the AppViewX Cloud Connector](#).

Accessing SSH+ Features

You have to access the SSH+ node to access the various functions provided by it.

To access SSH+:

1. Log into AppViewX with valid credentials.
2. Hover the mouse pointer over  (**Menu**) icon on the top-left corner of the screen.
3. From the left pane, click **SSH+**. You can now see the different menus of **SSH+** on the left hand side of the page.



4. From the left pane, expand any of the nodes to see that page.

**Note:**


For detailed functionalities, refer to the [SSH+ User Guide](#).

Configuring Provision Settings

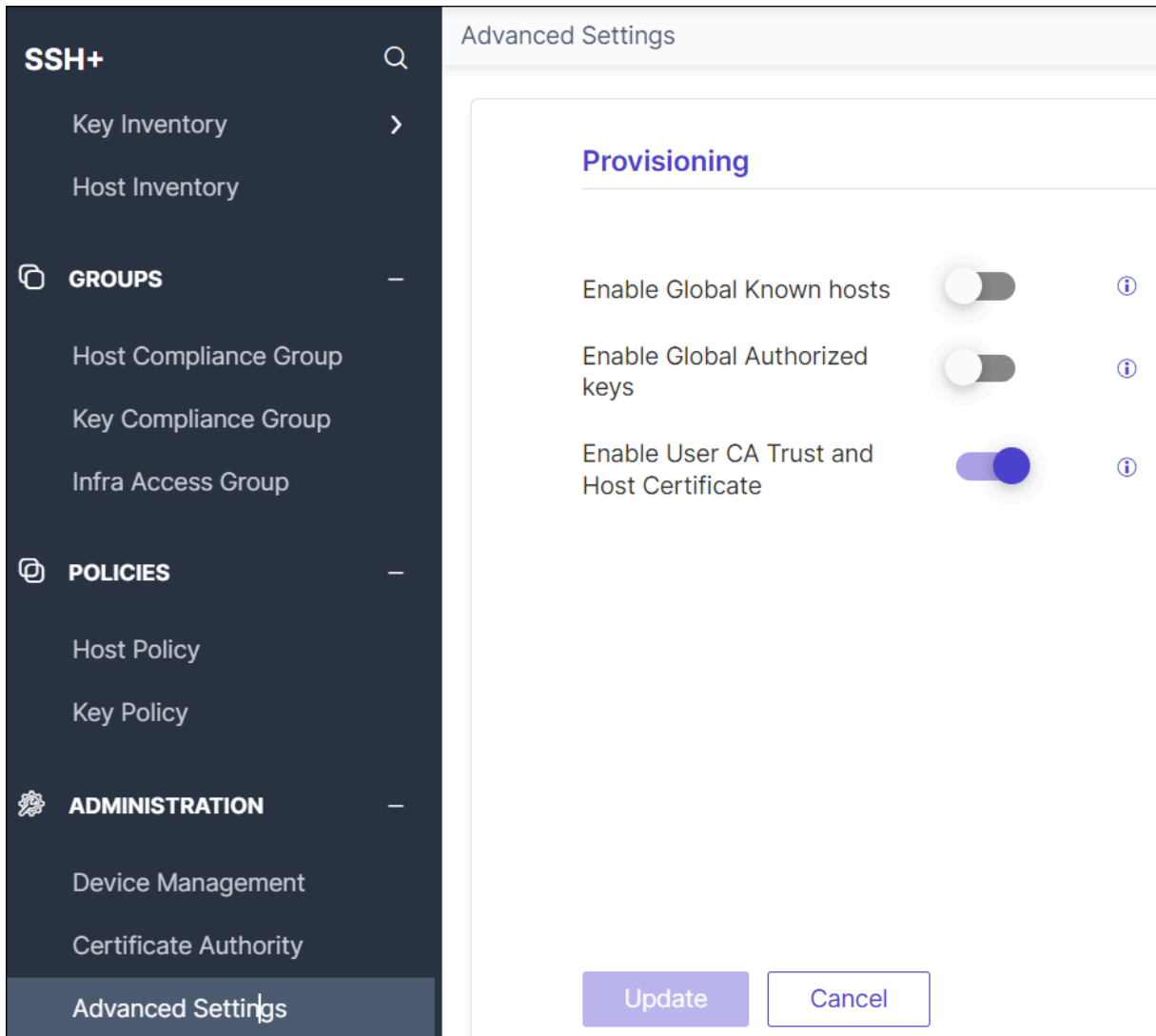
You can gain comprehensive control over key and certificate provisioning with the options available under **Advanced Settings > Provisioning** section. The options are:

- **Provision to Enable Global Known hosts**
- **Provision to Enable Global Auth keys**
- **Provision to Enable User CA Trust and Host Certificate**

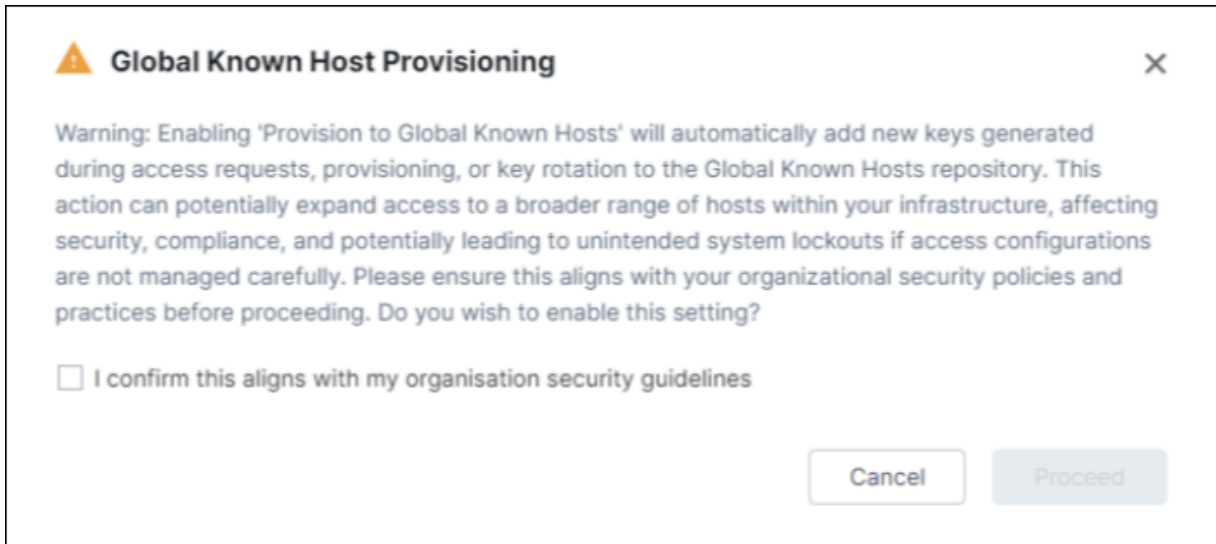
Procedure

1. Go to  (Menu) icon > **SSH+** > **Administration** > **Provisioning**.

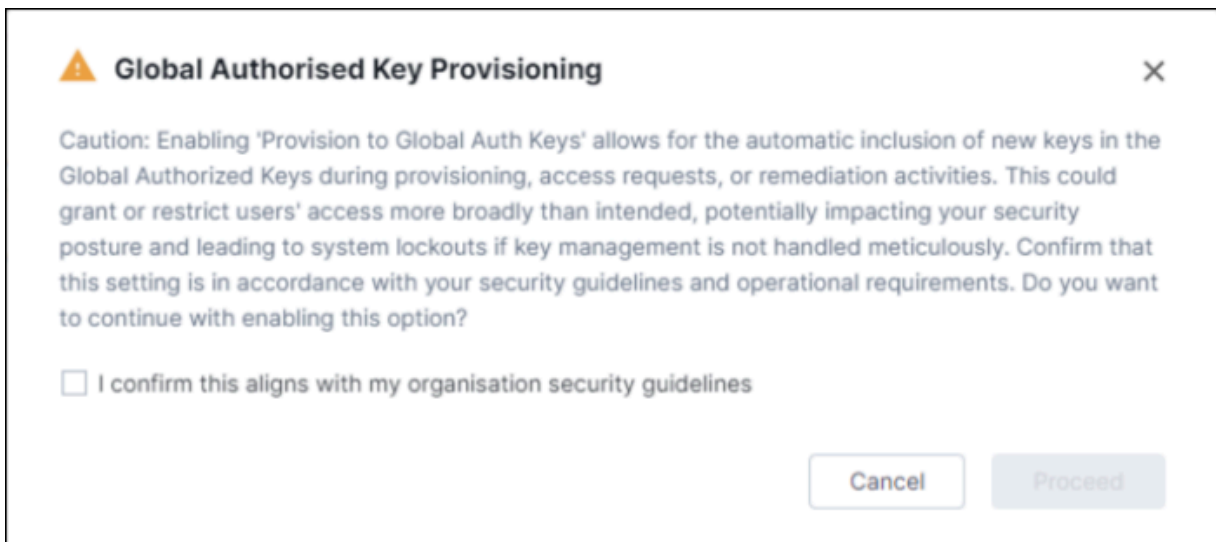
The **Provisioning** page is displayed.



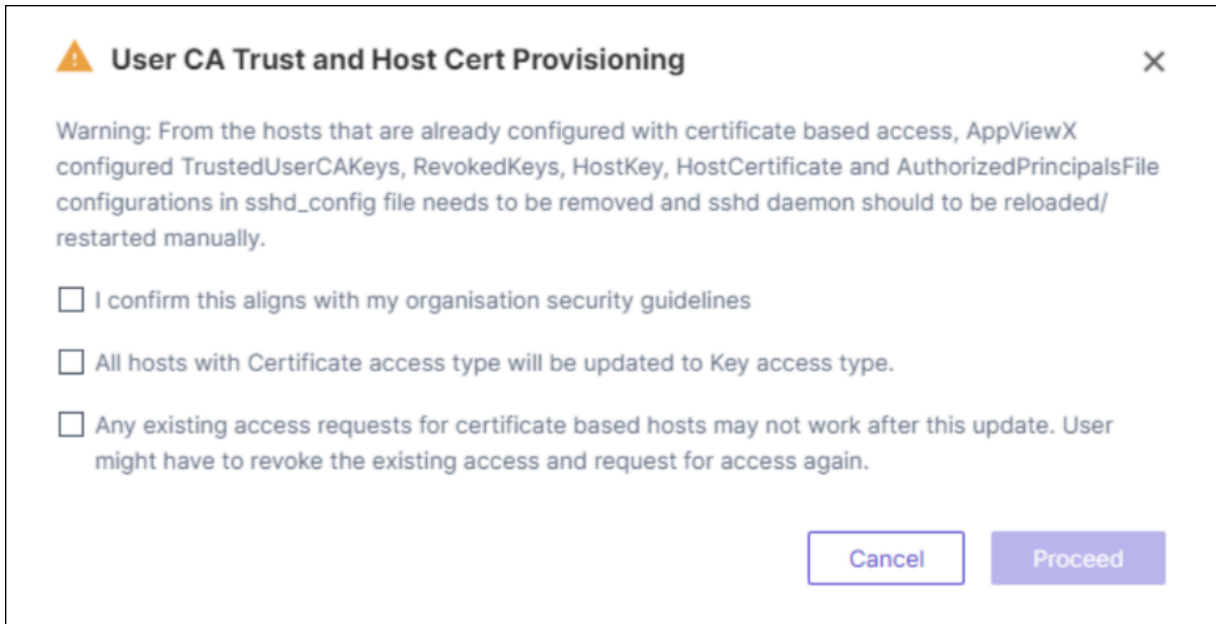
2. **Enable Global Known hosts:** By default, this option is disabled. Enabling this option allows to generate a Global Known Hosts configuration, ensuring that all new keys/certificates generated during access requests, provisioning, or remediation activities (such as key rotation) are added to the Global Known Hosts repositories. Enabling this option may have implications for your network.



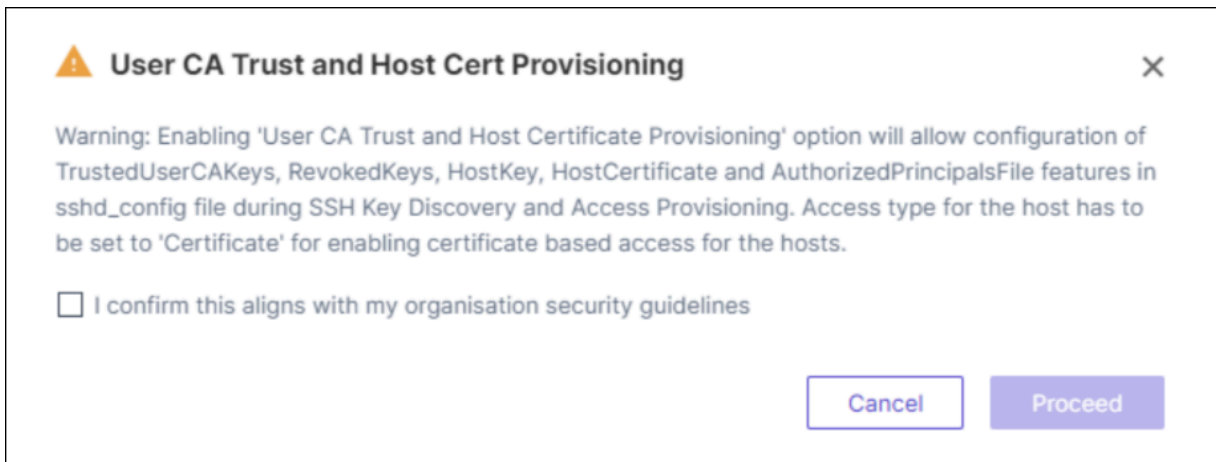
3. **Enable Global Authorized keys:** By default, this option is disabled. Enabling this option allows to establish a Global Auth Keys configuration, ensuring that all new keys/certificates generated during access requests, provisioning, or remediation activities (such as key rotation) are added to the Global Authorised Keys repositories. Enabling this option may have implications for their network.



4. **Enable User CA Trust and Host Certificate:** By default, this option is enabled. This option allows certificate based authentication for the devices. If you disable this option, the following **User CA Trust and Host Certificate** warning window opens. Select the checkboxes in the warning window, and then click **Proceed**.



If you enable this option again, you get the following **User CA Trust and Host Certificate** warning window opens. Confirm the checkbox, and then click **Proceed**.



5. Click **Update**. A message, *Provisioning settings saved successfully*, appears.

Getting Started

This section explains the SSH+ management workflow as seen on the Get Started page:

Onboarding Users

AppViewX offers comprehensive support for Role and Resource-Based Access Control (RBAC). RBAC is a method of restricting AppViewX functions, and managing and monitoring network resources in AppViewX based on the roles of individual users within an enterprise. 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.

For more information on configuring role and RBAC, refer to the Section, [Configuring Role and Resource-Based Access Control \(RBAC\)](#) in the Platform Guide.

Discovery and Visibility

You can gain full visibility of SSH user and host keys by discovering them within your network infrastructure.

- **Discover Hosts and Keys from IP network:** You can discover user or/and host keys configured on your server by creating and running scans on your network using IP address or subnet. The Discovery scans your network (on the default SSH enabled port 22) for SSH keys configured on your server. You can map the discovered keys to the selected key compliance groups and manage/monitor them.

Clicking this link takes you to the **Discovery** page. See [Using IP Range Option](#).

- **Risk Report Dashboard:** The risk reports are generated based on user and host configuration, key usage, and access requests once the discovery is complete. This information can help identify potential security risks, compliance issues, or other areas for improvement.

Clicking this link takes you to the **Dashboard** page. See [Reports](#).

- **User Key Inventory:** You can see the total number of weak, shared, orphan, and suspicious keys in the key discovery status. Click the number hyperlink to drill down on the metrics. This helps you track the progress of the key discovery efforts, identify any potential security risks, and prioritize the remediation actions.

Clicking this link takes you to the **Key Inventory** page. See [Viewing User/Host Key Inventory](#).

- **Host Key Inventory:** You can see the total number of weak and shared keys in the key discovery status. Click the number hyperlink to drill down on the metrics. This helps you track the progress of the key discovery efforts, identify any potential security risks, and prioritize the remediation actions.

Clicking this link takes you to the **Key Inventory** page. See [Viewing User/Host Key Inventory](#).

Access Management

You can ensure secure access control and authorization within your network infrastructure.

- **Onboard AWS Hosts:** Clicking this link takes you to the **Device Management > Device :: Cloud** page. See [Adding Cloud](#).
- **Host onboarding settings:** Clicking this link takes you to the **Host Inventory** page. See [Adding Host](#).
- **Manage Host Access Groups:** Clicking this link takes you to the **Infra Access Groups** page. See [Adding Infra Access Group](#).
- **Request Access:** Clicking this link takes you to the **Access Control** page. See [Accessing Requests](#)

Compliance

You can use key policies for configuring key rotation to ensure secure and standardized key management practices.

- **Configure Key Compliance Policy:** Clicking this link takes you to the **Key Policy** page. See [Key Policy](#).
- **Configure Host Configuration Policy:** Clicking this link takes you to the **Host Policy** page. See [Creating Host Policy](#).

Additional Settings

- **Onboard Network Hosts:** Clicking this link takes you to the **Device Management > Device :: Server** page. See [Adding Server](#).

Access Requests Hub

Any user with administrator privileges can use Access Request Hub if you have ACF permissions enabled. This is a full-featured tool for access request management enhancing security governance, operational efficiency, and compliance through detailed oversight and control of user access within the system.

1. Go to  (**Menu**) icon > **SSH+ > Manage Access > Access Requests Hub**.

The **Access Requests Hub** page is displayed.

You can view the details of the access requests such as requestor, requested for, app infra name, access mode, access duration, approver, and requested date.

2. You can approve or reject requests that are in *Pending Approval* status by hovering over the access request to see the **Approve or Reject** button as shown.

Requestor	Requested For	App Infra Name	Access Mode	Access Duration	Approved By	Requested C
admin	admin	RW ssh_demo	AppViewX Terminal	1 hours	admin	<input type="button" value="Approve"/> <input type="button" value="Reject"/>

On approving the request, the **Access Status** column changes to **Accessible**, and the **Approve or Reject** button changes to **Revoke**.

- You can revoke access requests that are in **Accessible** status by hovering over the access request to see the **Revoke** button as shown.

Requestor	Requested For	App Infra Name	Access Mode	Access Duration	Approved By
admin	admin	RW test_access_revoke_key	AppViewX Terminal	1 hours	Revoke

- You can use the **Search** option to search access requests by requester, requested for, app infra name, access mode, access duration, and approver.
- You can sort and filter access requests by clicking the hyperlink on the top panel that displays the number of access requests by their statuses such as accessible, partial, pending, denied, expired, failed, and revoked. Clicking the hyperlink fetches those access requests and displays details of the access requests.

Setting ACF and ACL Permissions


This section explains how you can set authorized functions (ACF) permissions for users based on their roles. Depending on the role and the ACF permissions set, the user can access some or all of the SSH+ functionalities.

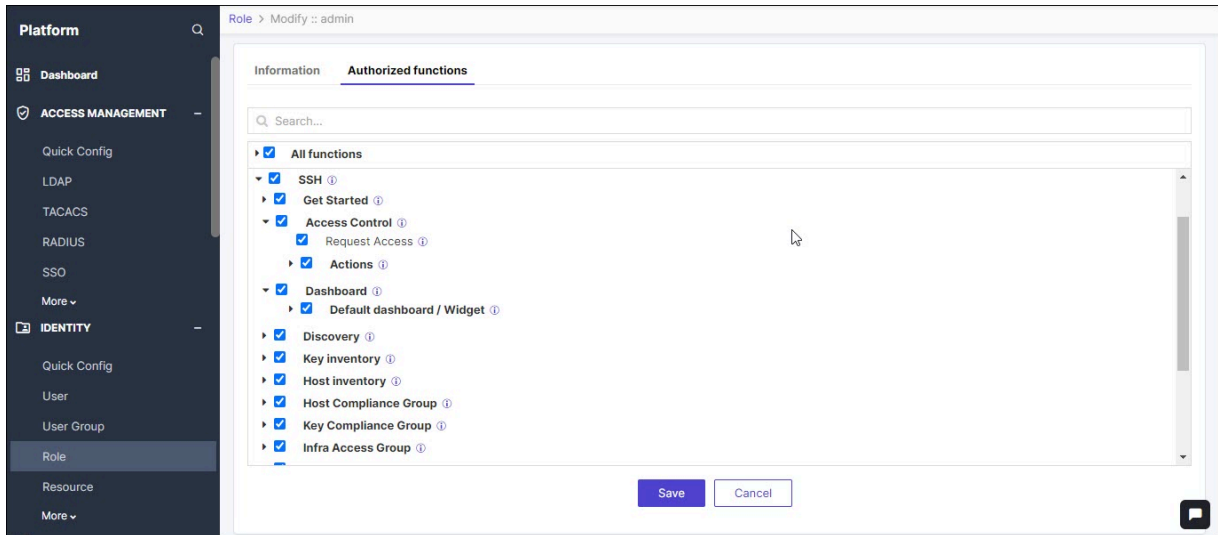
You can also assign/unassign resources access control list (ACL) permissions for infra access groups and key compliance groups. Depending on the ACL permissions set, resources will have partial, full, or no access to the groups and related actions.

Setting ACF Permissions for Roles

As administrators, you can set authorized functions (ACF) permissions for users based on their roles. Depending on the role and the ACF permissions set, the user can access some or all of the SSH+ functionalities.

To set ACF permissions for a role:

- Go to  (**Menu**) icon > **Platform** > **Identity** > **Role**.
- Click the role for which you want to set ACF permission.
- Select **Authorized functions**.
- Select and expand SSH to select and unselect the functions.



5. Click **Save**.


While most of the changes are implemented instantaneously, it takes approximately 3 minutes for changes in **Access Control** function to take effect.

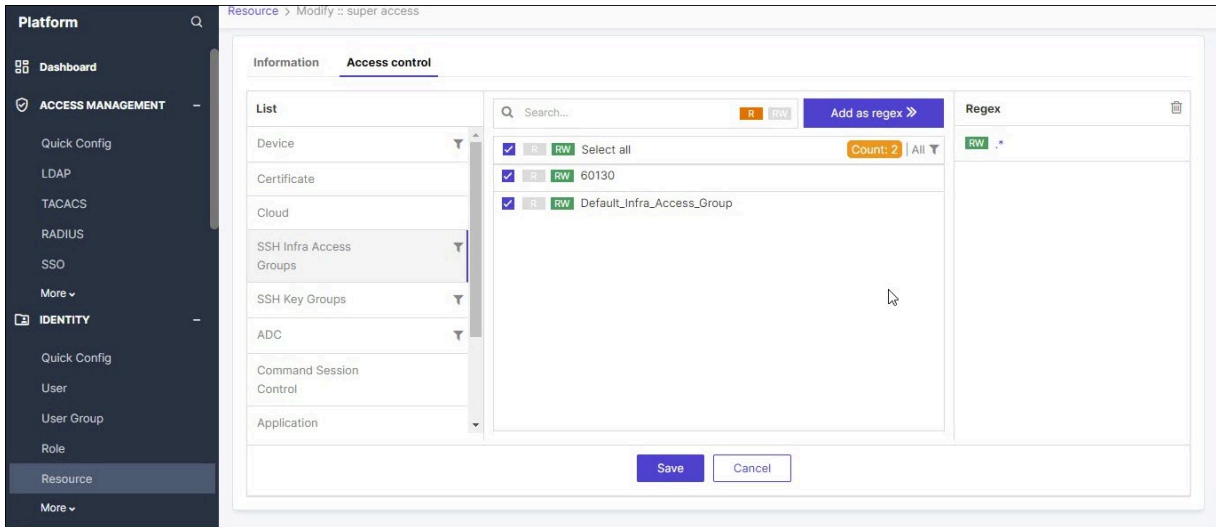
Setting ACL Permissions to Infra Access Groups

You can assign/unassign resources the following access control list (ACL) permissions to infra access groups:

- **RW** denotes that users have *Read-Write* permission to the infra access group along with hyperlinks to request access to the group on the **Access Control** page. Users can modify or delete infra access groups.
- **R** denotes that users only have *Read* permissions to the infra access groups. Users cannot request access or modify or delete infra access groups.
- If R or RW permission is not assigned, then users will not be able to view the infra access groups.

To assign ACL permission to infra access groups:

1. Go to  (**Menu**) icon > **Platform** > **Identity** > **Resource**.
2. Click a resource name.
3. Click **Access Control** tab and select **SSH Infra Access Groups** from the list.



The infra access groups are displayed on the right.

4. Select the infra access group for which you want to assign permission by clicking the check box.
5. Click **R** or **RW** button to assign the read or read-write permission.
6. Click **Save**.

A message, *Successfully assigned/unassigned SSH groups data for resource super access*, appears.



Note:


On deleting a resource, all ACL permissions from the associated infra access groups are also removed.

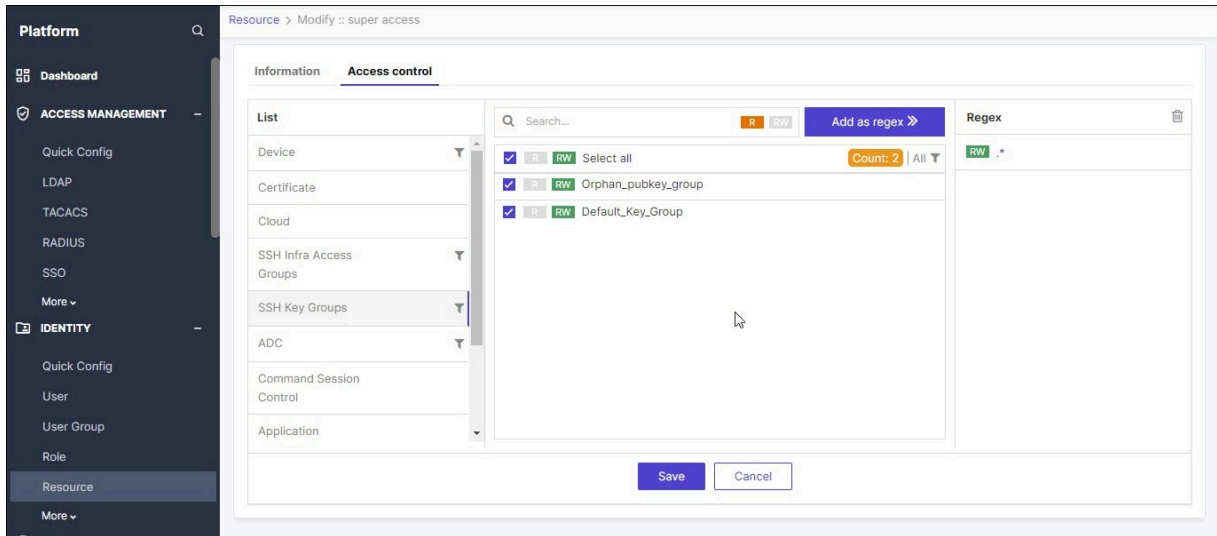
Setting ACL Permissions to Key Compliance Groups

You can assign/unassign resources the following access control list (ACL) permissions to key compliance groups:

- **RW** denotes that users have *Read-Write* permissions to the key compliance groups. Users can modify and delete key compliance groups. Users can view the keys associated with the key compliance groups and perform any of the actions in the inventory.
- **R** denotes that users have *Read* permissions to only view details displayed on the **Key Compliance Group** page. Users can only view the keys associated with the key compliance groups but cannot perform any action in the inventory.
- If R or RW permission is not assigned, then users will not be able to view the key compliance groups and the keys associated with them.

To assign ACL permission to key compliance groups:

1. Go to  (Menu) icon > **Platform** > **Identity** > **Resource**.
2. Click a resource name.
3. Click **Access Control** tab and select **SSH Key Groups** from the list.



The key groups are displayed on the right.

4. Select the key group for which you want to assign permission by clicking the check box.
5. Click **R** or **RW** button to assign the read or read-write permission.
6. Click **Save**.

A message, *Successfully assigned/unassigned SSH groups data for resource super access*, appears.



Note:


On deleting a resource, all ACL permissions from the associated key compliance groups are also removed.

Adding or Deleting Regex Patterns

Regex stands for regular expression; it is a string used to define filters. The string can contain a part of the device name or a key scan instance. These expressions are stored in the registry.

With Regex (Regular Expression) support, you can dynamically map user groups to infra access groups and key groups within the SSH+ module. This feature allows for the automatic mapping of newly created infra access groups and key groups that match specified Regex patterns, thus streamlining the process and reducing manual overhead.

To add or delete Regex patterns:

1. Go to  (**Menu**) icon > **Platform** > **Identity** > **Resource**.
2. Click a resource name.
3. Click **Access Control** tab and select **SSH Infra Access Groups** or **SSH Key Groups** from the list.
4. Type a regex pattern in the **Search** box and click **Add as regex**.


Examples of regex pattern include:

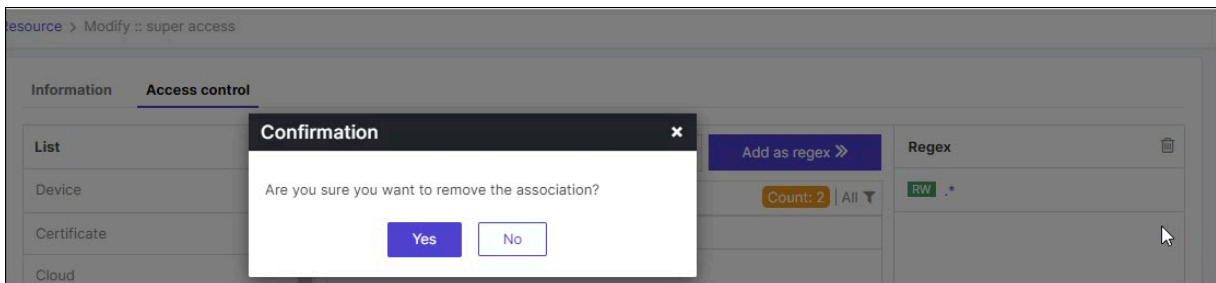
- `.*<text>`
- `<text>.*`
- `<text>.*<text>`
- `.*`

where `<text>` can be alphabets, numbers, and special characters to create these entries in the pages.

A message that the Regex pattern is successfully added for admin appears. Based on the matched regex, permission is automatically added to that resource. The newly added regex appears under the Regex column on the RHS of the page.

5. To individually delete a regex pattern, select the regex pattern from the **Regex** column and click **X**

against the name. To delete all regex patterns, click  (Delete) icon. A confirmation message appears as shown.



Click **Yes** and a message that it is successfully deleted from the matching groups appears.

Adding Access to Users

Overview

With access control, you can control and manage users and user lists who have access to the infra access groups and the actions they perform on the hosts. This helps to keep the infra access hosts secure.


You can view all the access requests from users on the **Get Started** page in the **Approvals** section. You can approve/reject the requests efficiently and effectively based on the business justification and for the

time duration requested in the access request form. On approving an access request, the user is granted access to the specified infra access group and hosts for the requested duration. The users will be able to perform authorized tasks such as running scripts, executing commands, and troubleshooting issues on those resources for that duration. On rejecting an access request, the user is denied access to the requested resources and will not be able to perform any actions.

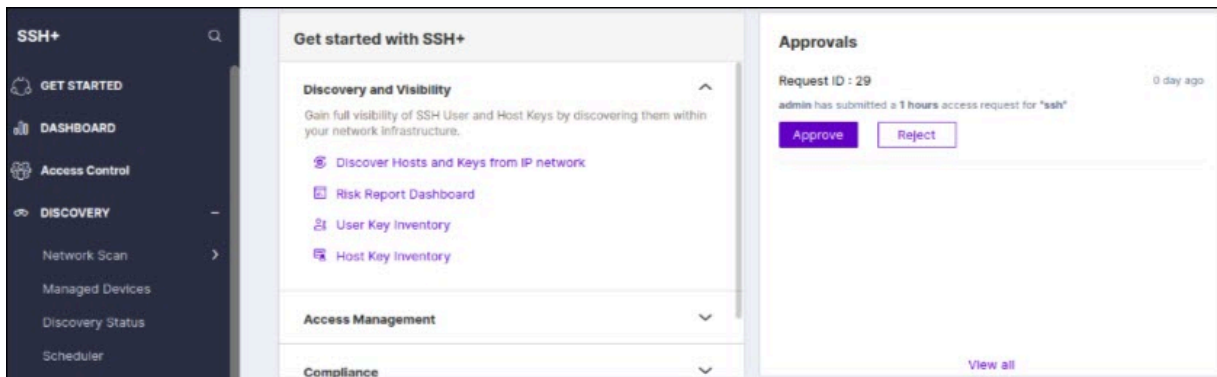
Approving/Rejecting Access Requests

All requests for terminal access are displayed on the **Get Started** page in the **Approvals** section.

To approve or reject access request:

1. Go to  (Menu) icon > **SSH+** > **Get Started**.

The access requests are displayed in the **Approvals** section.



2. Based on the requestors and their access rights, you can approve or reject the request.

On approving the request, the **Access Status** column on the **Terminal Access Control** page turns to:

- **Accessible:** Status when key provisioning is successful for all hosts in the infra access groups.
- **Partially Accessible:** Status when key provisioning fails for some hosts in the infra access groups.
- **Failed:** Status when key provisioning fails for all hosts in the infra access groups.

On approving an access request, the user is granted access to the specified infra access group and hosts for the requested duration. The users will be able to perform authorized tasks on those resources. On rejecting an access request, the user is denied access to the requested resources.



Note:

You can approve or reject access requests from **Access Requests Hub**. For more information, see [Access Requests Hub](#).

Glossary

Term definition

Term	Definition
SSH	Secure Socket Shell (SSH), also known as simply Secure Shell, is a cryptographic protocol used to enable secure access to remote servers and devices over the internet using SSH keys, certificates, or passwords.
SSH key	SSH keys are used to encrypt communication with a remote system. SSH keys usually come in pairs comprising a public and a private key and are used to grant access to authorized personnel to critical systems such as cloud, on-premise servers, and network devices.
Host key	A host key is a key that is used to identify the server. It is generated by the server and shared with the client during the initial connection setup. The client uses this key to verify the identity of the server before establishing a connection.
User key	A user key is a public key that is associated with a particular user account on the host. It is used to authenticate the user and establish a secure connection with the server.
Public key	A public key is used to encrypt data and verify digital signatures. It can be freely distributed, and anyone can use it to encrypt data or verify digital signatures. It is also used to establish a secure connection between the client and the server.
Private key	A private key is a secret key that is used to decrypt data and create digital signatures. It must be kept secret and never shared with anyone. The private key is used to authenticate the user and establish a secure connection with the server.
Suspicious key	A key without a known client association.
Shared key	A key used by more than one user.
Orphan key	A key that is found on a non-standard client file-folder path and does not have a known server.
SSH key rotation	The process of replacing the old key with a new one that adheres to the SSH key policy.
Weak key	A key that is generated using a weaker algorithm and size.

Chapter 2: SSH+ User Guide

Introduction to SSH+

As application infrastructures grow, so do security threats. Organizations have to find newer ways for protecting their data and granting access to the right users and devices to avoid security threats and breaches. While the traditional approach used password authentication, it proved to be insecure. This is where AppViewX SSH+ comes into play.

AppViewX SSH+ is a fully-automated application infra-access management and SSH key lifecycle management solution that allows you to centrally discover, manage, and protect SSH keys with access across hybrid multi-cloud environments. It also helps simplify access management, stay compliant and mitigate risks with SSH+.

AppViewX offers visibility and SSH access management across traditional on-premises data centers and cloud-hosted infrastructures.

Risks of improper SSH Management

Since there is no governing body to regulate the use of SSH keys, there is an element of risk involved. As SSH keys are generated on a need basis, several keys may be discarded and left unmanaged when they are no longer of use. Without an inventory, managing these keys and revoking their access pose a security threat to large organizations for potential back-door entry into the network, data theft, or breaches.

Improper SSH key management can lead to unauthorized access, compliance violations, identity and access management issues, data breaches, operational disruption, and reputation damage. To mitigate these risks, organizations should implement proper SSH key management practices, including secure key storage, regular key rotation, and access controls.

What Enterprises Need

AppViewX conducted multiple surveys to identify the core features and functionality needed to address SSH management challenges. SSH and Identity and Access Management (IAM) Administrators highlighted the following requirements:

- Discover keys from standard and non-standard locations
- Identify and report non-standard and non-compliant keys
- Visibility of keys and the users of these keys
- Revoke access to non-compliant and non-standard keys

- Rotation and distribution of keys
- Self-service SSH access requests
- Support for cloud and legacy on-premise infrastructure
- Centralized SSH Certificate Authority

How AppViewX Can Help

AppViewX SSH+ key lifecycle management is a fully automated solution that discovers and manages enterprise SSH infrastructure. It can identify and mitigate risks associated with poorly managed passwordless access management.

AppViewX SSH+ features include:

- **Centralized Discovery and Visibility**
 - The solution offers on-demand scans to discover SSH keys across multi-vendor, hybrid network infrastructures, and map trust relationships to determine access privileges.
 - The consolidated inventory provides a central console to view and manage all SSH keys and hosts.
- **Risk Scorecard and One-Click Remediation**
 - The solution proactively identifies and remediates risks associated with inactive, weak, orphan, or suspicious keys using an intuitive SSH scorecard dashboard.
 - The one-click remediation feature enables instant deletion or regeneration of keys.

System Requirements

SSH within the Application Infrastructure

Application infrastructure refers to all the components required to deliver an application and its functions and services to the customer. Although each application is unique, certain common components can be identified that are typically implemented to support application capabilities and service delivery.

One of the most common components of a typical application infrastructure is Linux hosts.

Modern application infrastructure leverages multiple hosting platforms such as on-premise data centers, private clouds, and public clouds offered by third-party hosting providers.

The application infrastructure components communicate with each other to enable service delivery. This can leverage SSH communication. Additionally, administrators of these application infrastructures will need to SSH into these hosts to perform maintenance.

Additionally, security and audit compliance requirements necessitate constant awareness of who has access to what and the maintenance of best security practices for the communications.


Hardware

Ensure that you have, at minimum, the following hardware with the given specifications before proceeding with the installation:

• Single Node Deployment Requirements

Node	Bare Minimum		
	CPU	RAM	Hard Disk Space
Single node	8	32GB	500GB

• Multi-Node Deployment Requirements

Node	Bare Minimum		
	CPU	RAM	Hard Disk Space
Multi-node (master node)	4	4GB	100GB
<div style="border: 1px solid #00aaff; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: One node for a single master installation and a minimum of three nodes for multi-master installation. </div>			
Multi-node (worker node)	8	32GB	500GB

• Platform Bare Minimum Requirements

Supported Virtualization Platforms	Versions	vCPU	RAM	HDD
VM Server, VMware ESXi	5.5 or later	8v	32GB	1TB

Operating System

Both single node and multi-node installations of AppViewX are supported on the following operating systems:

- RHEL 8.5
- RHEL 8.6
- RHEL 8.7
- Ubuntu 20.04

Browser

Following is the browser requirements to use the AppViewX SSH+ node:

Browser	Version
Firefox	v74.0.1 (64-bit) or later
Google Chrome	v85.0.4183.83 (64-bit) or later

Accessing SSH+ Features




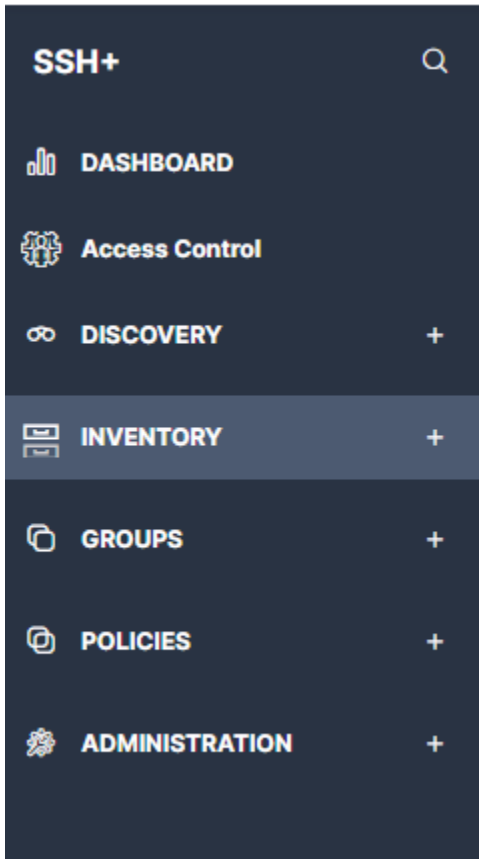
Note:

The SSH+ features accessible to you is based on the ACF permissions enabled by the administrator.

You have to access the SSH+ node to access the various functions provided by it.

To access SSH+:

1. Log into AppViewX with valid credentials.
2. Hover the mouse pointer over  (**Menu**) icon on the top-left corner of the screen.
3. From the left pane, click **SSH+**. You can now see the different menus of **SSH+** on the left hand side of the page.



4. From the left pane, expand any of the nodes to see that page.

Discovering Keys

Overview

Before you begin: You can access this functionality only if you have the ACF permissions enabled for your role.

SSH keys are installed to grant and protect access to privileged accounts. When initially deployed on a device, the device is configured to change privileged account passwords; however, if the devices are deployed after the SSH keys are installed, changing the passwords does not stop SSH keys from working thus rendering the privileged account insecure. To make it secure, these keys must be found so you can remove them and make the accounts secure again.

From the **Discovery** page, you can:

- Discover keys configured by creating and running scans on your network using IP range or subnet option. You can map the discovered keys to the selected key compliance groups and manage/monitor them. See [Network Scan](#).
- Discover keys on the devices you configured by creating and running scans on your devices. If the key already exists in the key inventory under another key group, then only the additional location/filename/filepath details is updated. The keys are not updated to the group that was provided in the discovery creation details. See [Managed Devices](#).
- Fetch the details and the status of the discovery such as the discovery method, action, recurrence, status along with the start and end time. See [Discovery Status](#).
- Create, customize, or delete the scheduler to run the discoveries. See [Scheduler](#).
- Fetch the key discovery status of the user and host keys, risk report, details of the user and host keys and the hosts. See [Viewing Discovery Summary](#).

Network Scan

You can discover user or/and host keys configured on your server by creating and running scans on your network using IP address or subnet. The Discovery scans your network (on the default SSH enabled port 22) for SSH keys configured on your server. You can map the discovered keys to the selected key compliance groups and manage/monitor them.




Note:

- Only live IPs (hosts) discovered from the IP range/subnet will be reflected in the discovery summary.
- Any key with a passphrase created by pre-OpenSSH 7.8 versions can NOT be discovered and parsed by AppViewX.


Using IP Range Option



To discover keys using the IP Range option:



1. Go to  (**Menu**) icon > **SSH+** > **Discovery** > **Network Scan** > **IP Range**.
The **Discover** page is displayed.
2. Enter the following details:

Field description for Discover IP Range section

Field	Description
Discover By	
*Select	Select one of the options: <ul style="list-style-type: none"> • Instant: To discover the keys immediately. By default, the Instant option is selected. • Scheduled: To schedule the discovery of keys on a specific date and time.
Scheduler (This section appears only if you have selected the Discovery option as <i>Scheduled</i>)	
*Schedule Name	Enter a unique name. This helps you identify it easily.
Description	Enter details pertaining to the scheduling discovery purpose.
*Starts On	Under the Starts On , set the time to start the run. You can customize the date, month, year, and time by clicking the Calendar icon.
*Repeat Every	Schedule discovery can be set to repeat discovery after every 5 minutes or can be customized per your requirement.
*End Date	Select one of the options to end the scheduled discovery: <ul style="list-style-type: none"> • Never: To keep the scheduled discovery going. • On: To select the end date when the scheduled discovery has to stop. • After: To stop the scheduled discovery after a certain number of occurrences.
Discover SSH Keys	
*Discovery Name	This field appears on selecting the Instant discovery option. Enter a unique name. This helps you identify it easily.
Description	This field appears on selecting the Instant discovery option. Enter the details pertaining to the discovery stating the purpose.
*Start IP	Enter the start range of the IP address for discovery.
*End IP	Enter the end range of the IP address for discovery.
*Ip(S) Per Batch	Select a value from the dropdown list. Based on this value, the subnet provided is split into multiple batches for the discovery process.

Field	Description
* Ports	By default, the port is 22. You can enter a port number from where the keys have to be discovered.
* Access Type	Select Key or Certificate . <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: The Certificate option can be disabled or enabled by configuring the Enable User CA Trust and Host Certificate toggle button under Advanced Settings. </div>
* DataCenter	Select a datacenter to connect to the host(s).
* Credential Type	Select one of the options: <ul style="list-style-type: none"> • Manual entry: Enter the username and password. • Credential List: Select the credential details that are already stored in the credential inventory page.
* Credential Name	This field appears only if you have selected Credential Type as <i>Credential List</i> .
* Login Type	Select one of the options: <ul style="list-style-type: none"> • Password: Enter username and password. • Identity Key: Click Upload and the Upload SSH Private Key window opens. Browse for the key file and fill out all the fields. Enter passphrase.
Sudoer User	Enable this checkbox if you want: <ul style="list-style-type: none"> • privileges to perform actions on discovery, provisioning, and remediation. • to discover keys for all users configured in the host.
* Access Elevation	This field appears only on selection of Sudoer User .
* Discover	Select one or both of the options: <ul style="list-style-type: none"> • User Keys: To discover user keys. • Host Keys: To discover host keys.
* Application Infra Access Group	Groups with RW permission will be visible in the Application Infra Access Group field.

Field	Description
	<p>Select the Application Infra Access Group(s) to which you want to map the onboarded host.</p>
Key Compliance Group	<p>Groups with RW permission will be visible in the Key Compliance Group field.</p> <p>Select the required Key Compliance Group to which you want to map the discovered user keys. The discovered keys are associated with the selected Key Compliance Group.</p> <div data-bbox="574 659 1419 932" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>The key group selection simplifies the grouping of the discovered keys and checks the discovered keys for key compliance. The keys are checked for compliance based on the policy of the key group it is associated with.</p> </div>
*Scan Type	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Default: The system scans the default ssh folders. • Full: The system scans the entire location. You can enter the files name/path that you want to exclude from the discovery for non-standard location. Make sure to select the Sudoer User checkbox. • Directory: The system performs default scan along with directory scan in the specified directory. Enter the file name/path you want to exclude/include for non-standard location. <div data-bbox="574 1360 1419 1499" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Changing the scan type clears the File Path table.</p> </div>
File Path	<p>This field is enabled only if you select Full or Directory as your Scan Type.</p> <p>Enter the file/s name/path that you want to exclude/include scan (only for directory scan) from the discovery for non-standard location.</p> <p>File path should always start with /.</p>

Field	Description
Operation	<p>This field is enabled only if you select Full or Directory as your Scan Type.</p> <p>Select one of the options:</p> <ul style="list-style-type: none"> • Exclude: Disables the scan in the file/s name/path location entered in File Path. • Include: Enables the scan only in the file/s name/path location entered in File Path. <div data-bbox="574 638 1419 863" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Multiple folder/path entries can be entered for scan, which are displayed in the consecutive table with respect to File Path and Operation.</p> </div>
<div data-bbox="235 932 1419 1066" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Fields indicated with red asterisk (*) symbol are mandatory.</p> </div>	

3. Click **Add**.

The **File Path** table is populated with the operation.

4. In **Inventory Action**, select one of the options:

- **Do Not Move:** To avoid the movement of newly discovered keys in the inventory.
- **Manage:** To allow the system to manage the newly discovered keys, which are moved to the inventory with **Managed** status.
- **Monitor:** To allow the system to monitor the newly discovered keys, which are moved to the inventory with **Monitored** status.

5. Click **Discover**.

The discovery runs per the settings and the key scan instance is added to the discovery inventory with the **Status** as *In Progress* until the discovery is completed. The **Status** in the discovery inventory changes to *Successful* or *Failed* depending on the outcome of the scan.




Note:

Only live IPs (hosts) discovered from the IP range will be reflected in the discovery summary.


Using Subnet Option



To discover keys with the subnet option:



1. Go to  (Menu) icon > **SSH+** > **Discovery** > **Network Scan** > **Subnet**.
The **Discover** page is displayed.
2. Enter the following details:

Field description for Discover Subnet section

Field	Description
Discover By	
*Select	Select one of the options: <ul style="list-style-type: none"> • Instant: To discover the keys immediately. By default, the Instant option is selected. • Scheduled: To schedule the discovery of keys on a specific date and time.
Scheduler (This section appears only if you have selected the Discovery option as <i>Scheduled</i>)	
*Schedule Name	Enter a unique name. This helps you identify it easily.
Description	Enter the details pertaining to the scheduling discovery purpose.
*Starts On	Under the Starts On , set the time to start the run. You can customize the date, month, year, and time by clicking the Calendar icon.
*Repeat Every	Scheduled discovery can be set to repeat discovery after every 5 minutes or can be customized per your requirement.
*End Date	Select one of the options to end the scheduled discovery: <ul style="list-style-type: none"> • Never: To keep the scheduled discovery going. • On: To select the end date when the scheduled discovery has to stop. • After: To stop the scheduled discovery after a certain number of occurrences.
Discover SSH Keys	
*Discovery Name	This field appears on selecting the Instant discovery option. Enter a unique name.
Description	This field appears on selecting the Instant discovery option.

Field	Description
	Enter the details pertaining to the discovery stating the purpose.
*Network	Enter the IP address of the network. For example, 192.168.1.1/24
*Subnets Per Batch Of Discovery	Select a value from the dropdown list. Based on this value, the subnet provided is split into multiple batches for the discovery process.
*Ports	By default, the port is 22. You can enter a port number from where the keys have to be discovered.
*Access Type	<p>Select Key or Certificate.</p> <div style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The Certificate option can be disabled or enabled by configuring the Enable User CA Trust and Host Certificate toggle button under Advanced Settings.</p> </div>
*DataCenter	Select a datacenter to connect to the host(s).
*Credential Type	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Manual entry: Enter the username and password. • Credential List: Select the credential details that are already stored in the credential inventory page.
*Credential Name	This field appears only if you have selected Credential Type as <i>Credential List</i> .
*Login Type	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Password: Enter username and password. • Identity Key: Click Upload and the Upload SSH Private Key window opens. Browse for the key file and fill out all the fields. Enter passphrase.
Sudoer User	<p>Enable this checkbox if you want:</p> <ul style="list-style-type: none"> • privileges to perform actions on discovery, provisioning, and remediation. • to discover keys for all users configured in the host.
*Access Elevation	This field appears only on selection of Sudoer User .
*Discover	Select one or both of the options:

Field	Description
	<ul style="list-style-type: none"> • User Keys: To discover user keys. • Host Keys: To discover host keys.
* Application Infra Access Group	<p>Groups with RW permission will be visible in the Application Infra Access Group field.</p> <p>Select the Application Infra Access Group(s) to which you want to map the onboarded host.</p>
Key Compliance Group	<p>Groups with RW permission will be visible in the Key Compliance Group field.</p> <p>Select the required Key Compliance Group to which you want to map the discovered user keys. The discovered keys are associated with the selected Key Compliance Group.</p> <div data-bbox="574 884 1419 1157" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: The key group selection simplifies the grouping of the discovered keys and checks the discovered keys for key compliance. The keys are checked for compliance based on the policy of the key group it is associated with.</p> </div>
* Scan Type	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Default: The system scans the default ssh folders. • Full: The system scans the entire location. You can enter the files name/path that you want to exclude from the discovery for non-standard location. Make sure to select the Sudoer User checkbox. • Directory: The system performs default scan along with directory scan in the specified directory. Enter the file name/path you want to exclude/include for non-standard location. <div data-bbox="574 1587 1419 1717" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: Changing the scan type clears the File Path table.</p> </div>
File Path	This field is enabled only if you select Full or Directory as your Scan Type .

Field	Description
	<p>Enter the file/s name/path that you want to exclude/include scan (only for directory scan) from the discovery for non-standard location.</p> <p>File path should always start with /.</p>
Operation	<p>This field is enabled only if you select Full or Directory as your Scan Type.</p> <p>Select one of the options:</p> <ul style="list-style-type: none"> • Exclude: Disables the scan in the file/s name/path location entered in File Path. • Include: Enables the scan only in the file/s name/path location entered in File Path. <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Multiple folder/path entries can be entered for scan, which are displayed in the consecutive table with respect to File Path and Operation.</p> </div>
<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note: Fields indicated with red asterisk (*) symbol are mandatory.</p> </div>	

3. Click **Add**.

The **File Path** table is populated with the operation.

4. In **Inventory Action**, select one of the options:

- **Do Not Move:** To avoid the movement of newly discovered keys in the inventory.
- **Manage:** To allow the system to manage the newly discovered keys, which are moved to the inventory with **Managed** status.
- **Monitor:** To allow the system to monitor the newly discovered keys, which are moved to the inventory with **Monitored** status.

5. Click **Discover**.

The discovery runs per the settings and the key scan instance is added to the discovery inventory with the **Status** as *In Progress* until the discovery is completed. The **Status** in the discovery inventory changes to *Successful* or *Failed* depending on the outcome of the scan.


**Note:**

Only live IPs (hosts) discovered from the subnet will be reflected in the discovery summary.

Managed Devices


You can discover keys by creating and running scans on your configured devices. The Discovery scans these devices for SSH keys configured on your server. You can map the discovered keys to the selected key compliance groups and manage/monitor them.



To discover keys using managed devices option:



1. Go to  (Menu) icon > **SSH+ > Discovery > Managed Devices**.
The **Managed Devices > Discover** page is displayed.
2. Enter the following details:

Field description for Discover Managed Devices section

Field	Description
Discover By	
*Select	Select one of the options: <ul style="list-style-type: none"> • Instant: To discover the keys immediately. By default, Instant option is selected. • Scheduled: To schedule the discovery of keys on a specific date and time.
Scheduler (This section appears only if you have selected the Discovery option as <i>Scheduled</i>)	
*Schedule Name	Enter a unique name. This helps you identify it easily.
Description	Enter details pertaining to the scheduling discovery purpose.
*Starts On	Under the Starts On , set the time to start the run. You can customize the date, month, year, and time by clicking the Calendar icon.
*Repeat Every	Schedule discovery can be set to repeat discovery after every 5 minutes or can be customized per your requirement.

Field	Description
*End Date	Select one of the options to end the scheduled discovery: <ul style="list-style-type: none"> • Never: To keep the scheduled discovery going. • On: To select the end date when the scheduled discovery has to stop. • After: To stop the scheduled discovery after a certain number of occurrences.
Discover SSH Keys	
*Discovery Name	This field appears only on selecting the Instant discovery option. Enter a unique name. This helps you identify it easily.
Description	This field appears only on selecting the Instant discovery option. Enter details pertaining to the discovery stating the purpose.
<p>A list of added and managed devices is displayed. Only devices with status as <i>Managed</i> and those that have RW permission are displayed in the list.</p> <p>From the list of managed device(s), select the Managed Device(s). The selected device(s) is the source of discovery.</p> <p>To select all the managed devices, select Select all. All the managed devices are the source of discovery.</p> <p>To understand the functionality of Regex, see Using Regex Feature.</p>	
*Ip(S) Per Batch	Select a value from the dropdown list. Based on this value, the subnet provided is split into multiple batches for the discovery process
*Discover	Select one or both of the options: <ul style="list-style-type: none"> • User Keys: To discover user keys. • Host Keys: To discover host keys.
Key Compliance Group	Select the required Key Compliance Group to which you want to map the discovered keys. The discovered keys are associated with the selected Key Compliance Group . <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The key group selection simplifies the grouping of the discovered keys and checks the discovered keys for key </div>

Field	Description
	 compliance. The keys are checked for compliance based on the policy of the key group it is associated with.
*Scan Type	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Default: The system scans the default ssh folders. • Full: The system scans the entire location. You can enter the files name/path that you want to exclude from the discovery for non-standard location. • Directory: The system performs default scan along with directory scan in the specified directory. Enter the file name/path you want to exclude/include for non-standard location. <div data-bbox="597 808 1417 940" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Changing the scan type clears the File Path table. </div>
File Path	<p>This field is enabled only if you select Full or Directory as your Scan Type.</p> <p>Enter the file/s name/path that you want to exclude/include scan (only for directory scan) from the discovery for non-standard location.</p> <p>File path should always start with '/'.</p>
Operation	<p>This field is enabled only if you select Full or Directory as your Scan Type.</p> <p>Select from the following options:</p> <ul style="list-style-type: none"> • Exclude: Disables the scan in the file/s name/path location entered in File Path. • Include: Enables the scan only in the file/s name/path location entered in File Path.

Field	Description
	 Note: Multiple folder/path entries can be entered for scan, which are displayed in the consecutive table with respect to File Path and Operation .
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	

3. Click **Add**.

The **File Path** table is populated with the operation.

4. In **Inventory Action**, select one of the options:

- **Do Not Move:** To avoid the movement of newly discovered keys in the inventory.
- **Manage:** To allow the system to manage the newly discovered keys, which are moved to the inventory with **Managed** status.
- **Monitor:** To allow the system to monitor the newly discovered keys, which are moved to the inventory with **Monitored** status.

5. Click **Discover**.

The discovery runs per the settings and the key scan instance is added to the discovery inventory with the **Status** as *In Progress* until the discovery is completed. The **Status** in the discovery inventory changes to *Completed* or *Failed* depending on the outcome of the scan.

Using Regex Feature

Regex stands for regular expression; it is a string used to define filters. The string can contain a part of the device name or a key scan instance. These expressions are stored in the registry and can be used to select the devices/key scan instances in future. This feature is available for discovering keys using the managed devices.

It enables you to filter your records on the strings mentioned in **Regex**.

1. Go to **Managed Devices > Discover** page.

The **Discover Managed Devices** page is displayed.

2. On the left side is the list of the devices/key scan instances.

- In the search bar, enter an expression.
- Click **Add as regex >>**.

The expression is added in the list on the right hand side.




Note:

These expressions are stored in the registry and can be used to select the devices/key scan instances in future.

You can use this register of expressions (Regex) for all future managed devices.

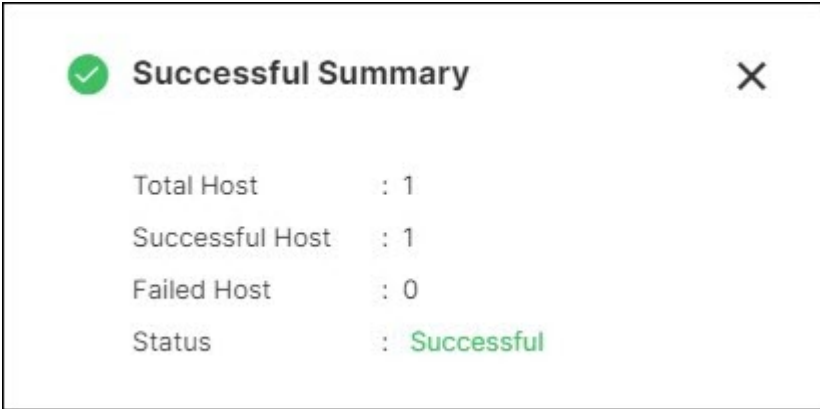
Discovery Status

To discover keys using the discovery status option:

- Go to  (**Menu**) icon > **SSH+ > Discovery > Discovery Status**.
The **Discovery Status** page is displayed.
- Displays the following details:

Field description for Discovery Status section

Field	Description
Discovery Name	<p>Displays unique discovery name. This helps you identify it easily.</p> <p>This field is clickable and opens a discovery report for the host with the following discovery status:</p> <ul style="list-style-type: none"> • Successful: Displays all tabs such as Hosts, Host Keys, User Keys, and Risk Report. Hosts can have statuses such as <i>Managed</i> or <i>Monitored</i>. • Failed: Displays only the Hosts tab. Each host has a <i>Failed</i> status, which is a hyperlink. Clicking it displays the reason for failure in a pop-up window. • Partial: Displays all tabs. The Hosts tab displays statuses such as <i>Managed</i>, <i>Monitored</i>, or <i>Failed</i>. The <i>Failed</i> status is a hyperlink. Clicking it displays the reason for failure in a pop-up window.
Discovery Mode	Displays the mode of discovery (IP range, subnet, or managed devices).
Discover action	Displays the discovery action as Instant or Scheduled .
Recurrence Type	Displays the frequency of the run of the discovery key instance.

Field	Description								
Status	<p>Displays the discovery status which are Successful, Failed, or Partial.</p> <p>You can click the status to open a pop-up window displaying a summary of the discovery results.</p> <div data-bbox="571 436 1386 842" style="border: 1px solid black; padding: 10px; margin: 10px 0;">  <p>A pop-up window titled "Successful Summary" with a green checkmark icon and a close button (X). It displays the following summary:</p> <table border="0" style="width: 100%;"> <tr> <td>Total Host</td> <td>: 1</td> </tr> <tr> <td>Successful Host</td> <td>: 1</td> </tr> <tr> <td>Failed Host</td> <td>: 0</td> </tr> <tr> <td>Status</td> <td>: Successful</td> </tr> </table> </div> <p>To check the host status, click the Discovery Name > Hosts > Host status.</p> <div data-bbox="571 978 1419 1327" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>+ Device status log: TestDevice1(192.168.60.129) ×</p> <hr/> <p>— Network Status: Reachability Check (01/08/2024 11:26:57 AM) Success Successfully connected to TestDevice1</p> <hr/> <p>— Device communication (01/08/2024 11:26:57 AM) Success Communication with TestDevice1 verified: Login valid</p> <hr/> <p>— SSH Keys Discovery from Device (01/08/2024 11:27:56 AM) Success SSH keys successfully discovered on TestDevice1</p> <hr/> <p>— SSH Host Key and/or Cert Provisioning and UserCA trust (01/08/2024 11:27:56 AM) Skipped SSH Key/Cert Provisioning skipped for TestDevice1 - Possible reasons: Non-sudoer host, Host in Unmanaged/Failed state, Mandatory field missing.</p> </div> <p>For more details on the error messages, see Error Messages.</p>	Total Host	: 1	Successful Host	: 1	Failed Host	: 0	Status	: Successful
Total Host	: 1								
Successful Host	: 1								
Failed Host	: 0								
Status	: Successful								
Start Time	<p>Displays the Start Time--the time set to start the discovery run. You can customize it if the Recurrence Type is one of these:</p> <ul style="list-style-type: none"> • Daily • Weekly • Monthly • Yearly 								
End Time	<p>Displays the date and time when the discovery ends.</p>								
Description	<p>Displays the details pertaining to the discovery stating the purpose.</p>								


What to do next:

- To schedule a discovery using the same input parameters, click **Actions > Rediscover**.
- To delete a discovery, select the checkbox against the **Discovery Name(s)** that has to be deleted and click **Actions > Delete**.

The selected discovery is deleted from the AppViewX database.

Viewing Discovery Summary

To view the discovery summary:

1. Go to  (**Menu**) icon > **SSH+ > Discovery > Discovery Status**.

The **Discovery Status** page is displayed.

2. Click the **Discovery Name** link.

The **Discovery Summary** of that discovery is displayed.

Summary

The summary report provides information about the discovery of the keys in the default branch. It contains the cumulative results of all successful discoveries.

Discovery summary mainly contains four reports:

- **Key Discovery Summary:** This widget gives a count of the discovered host keys and the user keys.

Color Code	Description
Orange	Displays the number of discovered host keys.
Blue	Displays the number of discovered user keys.

Clicking the widget redirects you to the **Host Keys/User Keys** tab.

- **Hosts:** This widget gives a count of the existing hosts and the newly discovered hosts.

Color Code	Description
Orange	Displays the number of existing hosts.
Blue	Displays the number of newly discovered hosts.

Clicking the widget redirects you to the **Hosts** tab.

- **User Keys:** This widget gives a count of the newly discovered user keys, missing user keys, and unchanged user keys in the device.

Clicking the number hyperlink redirects you to the **User Keys** tab.

- **Host Keys:** This widget gives a count of the newly discovered host keys, missing host keys, and unchanged host keys in the device.

Clicking the number hyperlink redirects you to the **Host Keys** tab.

Hosts

The Hosts tab displays the total number of compliant, non-compliant, weak kex algorithm hosts, weak ciphers, and weak mac algorithm hosts in the host discovery status. Click the number hyperlink to drill down on the metrics.

This helps you monitor the progress of the host discovery efforts, identify the compliance gaps, and prioritize the remediation actions.

Host Keys

The Host Keys tab displays the total number of weak, shared, orphan, and suspicious keys in the key discovery status. Click the number hyperlink to drill down on the metrics.

This helps you track the progress of the key discovery efforts, identify any potential security risks, and prioritize the remediation actions.

User Keys

The User Keys tab displays the total number of weak, shared, orphan, and suspicious keys in the key discovery status. Click the number hyperlink to drill down on the metrics.

This helps you track the progress of the key discovery efforts, identify any potential security risks, and prioritize the remediation actions.

Risk Report

The Risk Report tab contains the same field information as the Dashboard. The only difference is that while Dashboard shows the reports for all the discovered keys and also an option to perform remediation, you can use this page to fetch all the reports for the selected discovery. See [Reports](#).


Error Messages

Error Messages


Device Status	Status	Message
Network Status: Reachability Check	Success	Successfully connected to <deviceName>.
	Failed	Unable to connect to <deviceName>: Host not reachable.
	Failed	Connection to <deviceName> refused. Possible causes: Incorrect IP Address or Port, Service Not Running, Configuration Issues.
	Failed	FQDN <fqdn> resolution failed.
Device Communication	Success	Communication with <deviceName> verified: Login valid.
	Failed	Connection to <deviceName> failed: Invalid login credentials.
	Failed	Connection to <deviceName> failed: Invalid login credentials. <exceptionCaught>.
	Failed	Connection to <deviceName> refused. Possible causes: Incorrect IP Address or Port, Service Not Running, Configuration Issues.
SSH Keys Discovery from Device	Success	SSH keys successfully discovered on <deviceName>.
	Failed	SSH Key discovery on <deviceName> failed. Error encountered during the discovery process: <exceptionCaught during discovery>.
SSH Host Key and/or Cert Provisioning and UserCA Trust	Skipped	SSH Key/Cert Provisioning skipped for <deviceName>. Possible reasons: Non-sudoer host, Host in Unmanaged/Failed state, Mandatory field missing.
	Failed	SSH Host Key/Cert or UserCA Trust provisioning failed on <deviceName>. Possible reasons: Unable to create certificate for the given key, or unable to create a host certificate. Failures in building provisioning requests for the device.

Scheduler

To discover keys using the Scheduler option:

1. Go to  (Menu) icon > **SSH+** > **Discovery** > **Scheduler**.
The **Scheduler** page is displayed.
2. Displays the following details:

Field description for Scheduler section

Field	description
Schedule Name	Displays unique name for the schedule.
Discovery Mode	Displays the mode of discovery (IP range, subnet, or managed devices).
Recurrence Type	Displays the frequency of the run of the scheduled discovery key instance.
Last Execution Time	Displays the date and time of the previous scheduled discovery occurrence details.
Status	<p>Displays the discovery status which are Completed, Scheduled, and Paused.</p> <div style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Scheduled discovery can be paused or resumed by clicking the pause or resume icon before the occurrence of the discovery.</p> </div>
Description	Displays the details pertaining to the discovery stating the purpose.
Next Execution Time	Displays the date and time of the next scheduled discovery occurrence details.

What to do next:

- To modify the scheduled discovery, click **Discovery Status::Scheduler > Modify**.
- To delete a discovery, select the checkbox next to the **Schedule name** that has to be deleted and click **Actions > Delete**.

The selected discovery is deleted from the AppViewX database.

Managing Devices/Hosts

Overview

Before you get started: You can access this functionality only if you have the ACF permissions enabled for your role.

You can configure and manage devices (AWS Cloud and Linux servers), enable certificate sync for AppViewX to connect with customer's accounts and discover certificates, enable SSH sync for AppViewX to connect with customer's accounts, and discover host and user keys.

From the **Managing Devices** page, you can:

- Configure devices (AWS Cloud and Linux servers). See [Adding Cloud](#) or [Adding Server](#).
- Set user name and passwords to make the devices secure for SSH management. See [Adding Credentials](#).
- Perform action such as export, import, manage, unmanage, or delete a server, or fetch configuration from a server. See [Actions](#).

Adding Credentials

To add credentials for any device:

1. Go to  (**Menu**) icon > **SSH+ > Administration > Device Management**.

The **Device::Server** page is displayed.

2. Select the device from the tabs.

3. Click the  (**Credentials**) icon in the command bar.


The **Credentials** page is displayed. If credentials are set up, a list of credential names with details is displayed in the table.

4. To add a new credential, click + (**Add**) icon in the command bar.

The **Add Credential** page is displayed with default credentials fields for AppViewX.


- To set credentials for **AppViewX**:

Field description for AppViewX Credential Details section

Field	Description
* Credential name	Enter a suitable credential name.
* User name	Enter a suitable username.
Credential type (Password)	Select one of the options: <ul style="list-style-type: none"> • Password: Enter password and secondary password. • Identity key: Enter identity key and passphrase.
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	

- To set credentials for **CyberArk**:

Field description for CyberArk Credential Details section

Field Name	Description
* Credential name	Enter a suitable credential name
Type	Select one of the options: <ul style="list-style-type: none"> • Device: Enter user name, App ID, and user type. • Amazon (AWS/ELB): Enter AWS IAM user name, App ID, and AWS access key ID.
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	




Note:

To configure the API Settings for CyberArk, click the **Cyberark API Settings** button on the right of the screen.

- To set credentials for **Thycotic**:

Field description for Thycotic Credential Details section

Field Name	Description
* Credential name	Enter a suitable credential name.
* API Profile	Select the desired API profile from the dropdown list.
Secret Type	Select one of the options: <ul style="list-style-type: none"> • Device: Enter user name. • Amazon (AWS/ELB): Enter AWS IAM user name.
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	

**Note:**

To configure the API Settings for Thycotic, click the **Thycotic API Settings** button on the right of the screen.

Host Inventory

Hosts can be discovered, added, modified, deleted, and decommissioned. The most important feature of this module is discovering the **cloud host**. **Cloud host** management helps simplify grouping and access to the host.


- You can discover the hosts that are a part of the AWS cloud.
- The devices can be automatically grouped together based on *AWS Tags* during the run of a cloud host discovery host instance.
- The automatic grouping option can be enabled using a toggle button.
- Alternatively, you can manually create an Access Group by grouping the devices as per your discretion.

Adding Host

**Note:**

AppViewX SSH+ currently supports only adding servers as hosts.


To add a host:


1. Go to  (**Menu**) icon > **SSH+ > Inventory > Host Inventory**.
2. On the command bar, click **+ Add Host**.


The **SSH+::Host Inventory > Add Host** page is displayed.


3. Enter the following details:

Field description for Add Host section

Field	Description
General information	
*Category	Select Server . Selecting the Server option displays the Port field.
*Vendor	Select Linux .
*Device Name	Enter the name of the device. Displays the port used while configuring the device.
Port	This is a non-editable field.
Jump Server Client	By default, this is turned off. Turning on the toggle button allows you to identify the host as a jump server. client. The application infra access groups selected also get mapped to this jump server. client.
*Access Type	Select Key or Certificate . <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: The Certificate option can be disabled or enabled by configuring the Enable User CA Trust and Host Certificate toggle button under Advanced Settings.</p> </div>
*FQDN / IP Address	Enter the FQDN or the IP address of the host.
*DataCenter	Select a datacenter to connect to the host.
*Inventory Action	Select one of the options:

Field	Description
	<ul style="list-style-type: none"> • Do Not Move: To avoid the movement of newly discovered keys in the inventory. • Manage: To allow the system to manage the newly discovered keys, which are moved to the inventory with Managed status. • Monitor: To allow the system to monitor the newly discovered keys, which are moved to the inventory with Monitored status.
*Discover	Select one or both of the options: <ul style="list-style-type: none"> • User Keys: To discover user keys. • Host Keys: To discover host keys.
*Scan Type	Select one of the options: <ul style="list-style-type: none"> • Default: The system scans the default ssh folders. • Full: The system scans the entire location. You can enter the files name/path that you want to exclude from the discovery for non-standard location. Make sure to select the Sudoer User checkbox. • Directory: The system performs default scan along with directory scan in the specified directory. Enter the file name/path you want to exclude/include for non-standard location. <div data-bbox="581 1094 1419 1224" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Changing the scan type clears the File Path table. </div>
File Path	This field is enabled if you select Full or Directory as your Scan Type . Enter the file/s name/path that you want to exclude/include scan (only for directory scan) from the discovery for non-standard location. File path should always start with /.
Operation	This field is enabled if you select Full or Directory as your Scan Type . Select one of the options: <ul style="list-style-type: none"> • Exclude: Disables the scan in the file/s name/path location entered in File Path. • Include: Enables the scan only in the file/s name/path location entered in File Path.

Field	Description
	 Note: Multiple folder/path entries can be entered for scan, which are displayed in the consecutive table with respect to File Path and Operation .
Click Add . The File Path table is populated with the results.	
Credentials	
*Credential Type	Select one of the options: <ul style="list-style-type: none"> • Manual entry: Enter username and password. • Credential List: Select the credential details that are already stored in the credential inventory page.
Login Using	Select one of the options: <ul style="list-style-type: none"> • Password: Enter username and password. • Identity Key: Click Upload and the Upload SSH Private Key window opens. Browse for the key file and fill out all the fields. Enter passphrase.
Sudoer User	Enable this checkbox if you want: <ul style="list-style-type: none"> • privileges to perform actions on discovery, provisioning, and remediation. • to discover keys for all users configured in the host.
*Access Elevation	This field appears only on selection of Sudoer User .
Assign group	
*Host Compliance Group	Groups with RW permission will be visible in the Host Compliance Group field.
*Application Infra Access Group	Groups with RW permission will be visible in the Application Infra Access Group field. Only users with ACF permission can create an infra access group by entering a name in the text box and pressing Enter. Select the required Application Infra Access Group to which you want to map the onboarded host. The onboarded hosts are associated with the selected Application Infra Access Group.

Field	Description
	The Application Infra Access Group selection simplifies the grouping of the onboarded hosts and checks the onboarded hosts for user compliance. The onboarded hosts are checked for compliance based on the policy of the Application Infra Access Group it is associated with.
*Key Compliance Group	Groups with RW permission will be visible in the Key Compliance Group field.
<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: Fields indicated with red asterisk (*) symbol are mandatory. </div>	


4. Click **Create**.

The host is created in the host inventory.

Viewing Host Inventory

The Host Inventory tab displays the total number of hosts with weak kex algorithms, weak ciphers, and weak mac algorithms and host discovery status. Click the number hyperlink to get the details of the host with weak kex algorithms/weak ciphers/weak mac algorithms. This inventory helps you monitor the progress of the host discovery for host management, identify the compliance gaps, and prioritize the remediation actions.

To view the details of the host inventory:





- Go to  (**Menu**) icon > **SSH+** > **Inventory** > **Host Inventory**.

The **Host Inventory** page is displayed.

Field description for Host Inventory section

Field	Description
Device name	Displays the unique name provided for the device.
FQDN/IP address	Displays the FQDN/IP address of the host.
Host name	Displays the name of the host.


Field	Description														
Host Permission	<p>Displays the count of infra access groups for which access control list (ACL) permissions are enabled for the hosts. Hosts with R (Read) and RW (Read-Write) permissions are only displayed. Clicking the hyperlink opens up a pop-up window displaying the infra access group and the related permissions as shown:</p> <div data-bbox="532 499 1349 1255" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: right; margin: 0;">Groups And Permissions ✕</p> <p style="text-align: right; margin: 0;">1 to 8 of 8 < ></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Group Name</th> <th style="text-align: left;">Permission</th> </tr> </thead> <tbody> <tr> <td>Default_Infra_Access_Group</td> <td style="text-align: center;">RW</td> </tr> <tr> <td>groupPermission</td> <td style="text-align: center;">RW</td> </tr> <tr> <td>hostGroup4</td> <td style="text-align: center;">R</td> </tr> <tr> <td>iodeji23</td> <td style="text-align: center;">RW</td> </tr> <tr> <td>newGRP1</td> <td style="text-align: center;">RW</td> </tr> <tr> <td>newPerm32</td> <td style="text-align: center;">RW</td> </tr> </tbody> </table> </div> <p>For example, if a host has ACL permissions enabled in eight of the infra access groups, then the count is shown as 8. In this example, the host has six RW and two R permissions enabled in different infra access groups, it is shown as RW 8. This will be the case if a host has only one RW and six R permissions as RW outweighs R permission.</p>	Group Name	Permission	Default_Infra_Access_Group	RW	groupPermission	RW	hostGroup4	R	iodeji23	RW	newGRP1	RW	newPerm32	RW
Group Name	Permission														
Default_Infra_Access_Group	RW														
groupPermission	RW														
hostGroup4	R														
iodeji23	RW														
newGRP1	RW														
newPerm32	RW														
Group	Displays the host compliance group associated with the host.														
Allowed Principals	Displays the count of users who have access to a particular host (certificate-based access type) based on the access provided by the administrator using Access Control . Clicking the count hyperlink opens the Host Level Access Settings window where administrator can remove host access by selecting the users. Click the confirmation message and provide a comment, which is displayed in the audit logs.														

Field	Description
Host Status	<p>Displays the status of the host as:</p> <ul style="list-style-type: none"> • Managed: When SSH Host Key and/or Cert Provisioning and UserCA trust operations are completed, it displays  Managed , else it displays  Managed . If the host status displays as managed with a warning symbol, it may be because the you did not check the 'sudo' option during discovery or host addition. <div data-bbox="558 604 1419 827" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: To resolve this, modify the host in the SSH+ host inventory to include 'sudo.' If this issue is not resolved, the access request functionality may not work successfully.</p> </div> <ul style="list-style-type: none"> • In-Progress: Status when host addition/discovery is in progress. • Failed: Status when host communication failed. • Unresolved: Status when the device is not reachable. <p>Clicking the hyperlink displays the device status.</p>
Jump Server Client	Displays the devices acting as a jump server. client. Client device allows the users to connect to other devices in the infra access group.
Access Type	Displays key or certificate.
User	Displays the name of the user onboarding the device.
Port	Displays the port number that is communicating with the host.
Vendor	Displays the operating system vendor associated with the host.
Last sync time	Displays the last sync date and time with the host.
Instance Id	<p>Displays the ID automatically generated by AWS when you launch a new EC2 instance.</p> <div data-bbox="537 1633 1419 1808" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: This is applicable only for EC2 instances discovered by adding the AWS cloud account.</p> </div>

Actions on Host Inventory

You can perform any of following actions by selecting the checkbox against the host name and clicking the **Actions** dropdown menu on the **Host Inventory** page:

Action description on Host Inventory page

Action	Description
Modify	Only users with RW permission can edit the selected host details.
Delete	Only users with RW permission can delete the active hosts from the host inventory.
Credentials	<p>You can view the credentials of the host keys that are discovered on the device. You can also add, modify, or delete the credentials. See Adding Credentials.</p> <p>You cannot delete the default or active credentials.</p>
Fetch Keys	Only users with RW permission can re-fetch all the keys from the host.
Export	You can export hosts from the host inventory to .csv or .xls format.
Rotate Host Certificate	<div style="border: 1px solid #ffc107; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> Important:</p> <ul style="list-style-type: none"> • Only users with RW permission can perform this action. • This action is enabled only for devices/hosts with host status as <i>Managed</i> with sudo access. • If the <i>global known host</i> is not present, then AppViewX will create one with the right CA definition. Rotate the host key along with the host certificate from the host inventory to update the public host key in the known hosts. </div> <p>A confirmation pop-up window appears warning about the implications of the rotation process. On confirmation, the existing host certificate is deleted and a new one is created in its place for the selected host in the <i>global known host</i> file.</p>

Adding Server



Note:

AppViewX SSH+ currently supports addition of only Linux servers.

To add a server:

1. Go to  (**Menu**) icon > **SSH+** > **Administration** > **Device Management**.

The **Device::Server** page is displayed.


2. On the command bar, click + (**Add**) icon to add a new server.


The **Device::Server > Add** page is displayed.

3. Select **Linux** from the **Vendors** list.
4. Enter the following details:

Field description for Device Details section


Field	Description
Server details	
*Server name	Enter a unique name for the server. This helps you identify it easily.
*IP address/FQDN	Enter the IP address/FQDN.
Data center	Select a data center from the dropdown list.
Communication mode	Select SSH.
*SSH Port	By default, the port is 22. You can choose to enter a port number.
Cert sync	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Managed: AppViewX connects to the customer's server account and discovers host and user keys. These keys are added to the host and key inventory. Users with the relevant permissions can then perform the required keys-related actions. • Monitored: AppViewX connects to the customer's server account and discovers host and user keys. These keys are added to the host and key inventory where the users are allowed to only view the keys. • Ignored: Certificate discovery is ignored.


Field	Description
	<div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: The Cert sync option is applicable only if the CERT+ license is active; otherwise, it can be ignored. </div>
Credentials	
*Credential Type	Select one of the options: <ul style="list-style-type: none"> • Manual entry: Enter the username and password. • Credential List - AppViewX: Select the credential details that are already stored in the credential inventory page. • SSH: Enter the username, browse and upload the identity key along with its passphrase.
Service account credentials	
Username	Enter the user name.
Password	Enter the password.
Vendor Specific Details	
Access Elevation	By default, the value is None . Select a value from the dropdown list.
Discover Formats	Enter a value to filter the formats to be discovered from the device. By default, all standard formats are discovered.
Certificate details	
Certificate Directory	Provide the directory from where the certificates must be discovered. By default, the system scans for certificates from all the directories.
Scan type	Select one of the options: <ul style="list-style-type: none"> • Default: The system scans for supported certificate formats such as pem, crt, cer, der, kdb, jks, p12, p7, pfx, and adds them to the certificate inventory. • Aggressive: The system scans for all keystore files with non-standard extensions.
*Operation	Select one the options: <ul style="list-style-type: none"> • Exclude: Disables the scan in the specified certificate directory. • Include: Enables the scan only in the specified certificate directory.

Field	Description
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	


5. Scroll down to the **SSH Details** section. By default, the **SSH Sync Enable** toggle button is turned off.
6. Click the **SSH Sync Enable** toggle button to enable SSH sync.
7. Click **Customise** to modify the default settings.
8. Enter the following fields:

Field description for SSH Details section

Field	Description
*Inventory Action	Select one of the options: <ul style="list-style-type: none"> • Do Not Move: To avoid the movement of newly discovered keys in the inventory. • Manage: To allow the system to manage the newly discovered keys, which are moved to the inventory with Managed status. • Monitor: To allow the system to monitor the newly discovered keys, which are moved to the inventory with Monitored status.
*Discover	Select one or both of the options: <ul style="list-style-type: none"> • User Keys: To discover user keys. • Host Keys: To discover host keys.
Scan Type	Select one of the options: <ul style="list-style-type: none"> • Default: The system scans the default ssh folders. • Full: The system scans the entire location. You can enter the files name/path that you want to exclude from the discovery for non-standard location. • Directory: The system performs default scan along with directory scan in the specified directory. Enter the file name/path you want to exclude/include for non-standard location. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Changing the scan type clears the File Path table. </div>

Field	Description
File Path	<p>This field appears if you select Full or Directory as your Scan Type.</p> <p>Enter the file/s name/path that you want to exclude/include scan (only for directory scan) from the discovery for non-standard location.</p> <p>File path should always start with '/'.</p>
Operation	<p>This field appears if you select Full or Directory as your Scan Type.</p> <p>Select one of the options:</p> <ul style="list-style-type: none"> • Exclude: Disables the scan in the file/s name/path location entered in File Path. • Include: Enables the scan only in the file/s name/path location entered in File Path. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Multiple folder/path entries can be entered for scan, which are displayed in the consecutive table with respect to File Path and Operation.</p> </div>
*Host Compliance Group	<p>Groups with RW permission will be visible in the Host Compliance Group field.</p> <p>Select the required Host Compliance Group to which you want to map the discovered hosts and host keys. The discovered hosts and host keys are associated with the selected host compliance group.</p>
*Key Compliance Group	<p>Groups with RW permission will be visible in the Key Compliance Group field.</p> <p>Select the required Key Compliance Group to which you want to map the discovered keys. The discovered keys are associated with the selected key compliance group.</p> <p>The key group selection simplifies the grouping of the discovered keys and checks the discovered keys for key compliance. The keys are checked for compliance based on the policy of the key group it is associated with.</p>

Field	Description
* Application Infra access group	Groups with RW permission will be visible in the Application Infra Access Group field. Only users with ACF permission can create an infra access group by entering a name in the text box and pressing Enter. Select the Application Infra Access Group(s) to which you want to map the onboarded host.

 **Note:**
Fields indicated with red asterisk (*) symbol are mandatory.

9. Click **Save**.

The host is created and displayed in the host inventory.



Note:

Server inventory allows the addition of multiple vendors configured under the same hostname/IP, but only one hostname/IP is added in the SSH+ inventory to avoid duplication of SSH actions.

What to do next:

- To add credentials to the server, see [Adding Credentials](#).
- To perform any of the actions such as export, import, manage, unmanage, or delete a server, or fetch configuration from a server, see [Actions](#).

Adding Cloud



Note:

AppViewX SSH+ currently supports addition of only AWS cloud EC2 servers.

1. Go to  (Menu) icon > **SSH+** > **Administration** > **Device Management**.

The **Device::Server** page is displayed.

2. Click the **Cloud** tab.


The **Device::Cloud** page is displayed.


3. On the command bar, click + (**Add**) icon to add a new cloud device.


The **Device::Cloud > Add** page is displayed. By default, AWS is selected from the **Vendors** list.

4. Enter the following fields:

Field description for AWS Device Details section

Field	Description
Basic Information	
*Account Type	Select Cross or Federated to authenticate using the assumed role.
*Account Name	Enter a unique name. It cannot be an account name that is already in the cloud inventory. Name can be alphanumeric and contain hyphen (-) and period (.).
*Account Number	Enter a valid AWS account number.
Account Description	Enter a description that helps identify your account from the cloud inventory.
Proxy Required	Select the checkbox if you want to create it as a proxy.
*Default Region	Select the region from the dropdown list for API communication.
*Data Center	Select a datacenter to connect to the host.
Credentials	
*Credential type	Select one of the options: <ul style="list-style-type: none"> • Manual Entry: Enter username and password. • Credential List: Select the credential details that are already stored in the credential inventory page.
*Access Key ID	Enter the access key ID.
*Secret Access Key	Enter the secret access key. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: To authenticate requests, use both the access key ID and the secret access key. </div>
Discover resources	
Auto Discover Resources	By default, this is turned off. Turn on the toggle button to discover all cross or federated/child accounts of the provided master account details.


Field	Description
Advanced Settings	By default, this is turned off. Turn on the toggle button to customize the auto-discovery process.
*Auto Discovery Mode	Select one or both of the options.
*Service	Select EC2 (EC2 instance) from the dropdown list.
*Service Region	Click Fetch Region to fetch the service regions for the provided account information.
Cert Sync	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Managed: AppViewX connects to the customer's server account and discovers host and user keys. These keys are added to the host and key inventory. Users with the relevant permissions can then perform the required keys-related actions. • Monitored: AppViewX connects to the customer's server account and discovers host and user keys. These keys are added to the host and key inventory where the users are allowed to only view the keys. • Ignored: Certificate discovery is ignored. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The Cert Sync option is applicable only if the CERT+ license is active; otherwise, it can be ignored.</p> </div>
Auto Sync	By default, this is turned off. Turn on the toggle button to auto sync based on trigger or schedule.
EC2 Services	
Communication mode	Keep the default selection.
Certificate Discovery Mode	Keep the default selection.
*S3 Deployment Type	Enter the S3 deployment type that can be a centralized S3 bucket.
*S3 Bucket Name	Click the Settings icon and fill out the mandatory fields in the ARN Advanced Settings window that pops up.



Field	Description
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	

5. Scroll down to the **SSH Details** section. By default, the **SSH Sync Enable** toggle button is turned off.
6. Click the **SSH Sync Enable** toggle button to enable SSH sync.
7. Click **Customise** to modify the default settings.
8. Enter the following fields:

Field description for SSH Details section

Field	Description
*Inventory Action	Select one of the options: <ul style="list-style-type: none"> • Do Not Move: To avoid the movement of newly discovered keys in the inventory. • Manage: To allow the system to manage the newly discovered keys, which are moved to the inventory with Managed status. • Monitor: To allow the system to monitor the newly discovered keys, which are moved to the inventory with Monitored status.
*Host Compliance Group	Groups with RW permission will be visible in the Host Compliance Group field. Select the required Host Compliance Group to which you want to map the discovered hosts and host keys. The discovered hosts and host keys are associated with the selected host compliance group.
*Key Compliance Group	Groups with RW permission will be visible in the Key Compliance Group field. Select the required Key Compliance Group to which you want to map the discovered keys. The discovered keys are associated with the selected key compliance group. The key group selection simplifies the grouping of the discovered keys and checks the discovered keys for key compliance. The keys

Field	Description
	are checked for compliance based on the policy of the key group it is associated with.
Scan Type	<p>Select one of the options:</p> <ul style="list-style-type: none"> • Default: The system scans the default ssh folders. • Full: The system scans the entire location. You can enter the files name/path that you want to exclude from the discovery for non-standard location. • Directory: The system performs default scan along with directory scan in the specified directory. Enter the file name/path you want to exclude/include for non-standard location. <div data-bbox="548 772 1419 905" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: Changing the scan type clears the File Path table.</p> </div>
*Discover	<p>Select one or both of the options:</p> <ul style="list-style-type: none"> • User Keys: To discover user keys. • Host Keys: To discover host keys.
File Path	<p>This field appears if you select Full or Directory as your Scan Type.</p> <p>Enter the file/s name/path that you want to exclude/include scan (only for directory scan) from the discovery for non-standard location.</p> <p>File path should always start with '/'.</p>
Operation	<p>This field appears if you select Full or Directory as your Scan Type.</p> <p>Select one of the options:</p> <ul style="list-style-type: none"> • Exclude: Disables the scan in the file/s name/path location entered in File Path. • Include: Enables the scan only in the file/s name/path location entered in File Path.

Field	Description
	 Note: Multiple folder/path entries can be entered for scan, which are displayed in the consecutive table with respect to File Path and Operation .
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	





What to do next:

- To add credentials to the cloud device, see [Adding Credentials](#).
- To perform any of the actions such as export, import, manage, unmanage, or delete a server, or fetch configuration from a server, see [Actions](#).






Actions

To do any of the following actions on the command bar for Server inventory, select the checkbox against the device.



Action Descriptions on Command Bar

Icon	Action	Description
	Terminal connect	Click to open the terminal connect page of the device.
	Delete	Click Yes on the confirmation popup window. The device is deleted from the inventory.
	Credential	Click to add credentials for the server. See Adding Credentials .
	Manage	Click Yes on the confirmation popup window. Configuration fetch is triggered, and the device status is changed to <i>Managed</i> in the device inventory.

Action Descriptions on Command Bar (continued)

Icon	Action	Description
		 Note: With the introduction of the Cert License Enforcement; Server Auto discovery of EC2 instances will be on-boarded with certificate status as <i>Monitored</i> , even if <i>Managed</i> state is configured in the device account configuration.
	Unmanage	Click Yes on the confirmation popup window. Configuration fetch is triggered, and the device status is changed to <i>Unmanaged</i> in the device inventory.
	Export	Select one of the options: <ul style="list-style-type: none"> • 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 screen. • Columns to modify data and import: Select this option if you want to export device details to make modifications and then import the data into the device inventory. The data is exported to an Excel (.xls) format.
	Import	<ol style="list-style-type: none"> 1. Click Import to be redirected to the import cloud page. 2. Download the sample <.csv> or <.xls> file. 3. Update the details. 4. Click to browse and upload the files. The Cloud details are updated in the cloud inventory.
	Fetch	A popup message, Fetch config has been triggered for the device appears. The configuration is fetched from the device.

Action Descriptions on Command Bar (continued)

Icon	Action	Description
	Device Settings	<p>Click to change the Certificate details section.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: The Device Settings option is applicable only if the CERT+ license is active; otherwise, it can be ignored.</p> </div>

Access Control

Overview

Before you begin: You can access this functionality only if you have the ACF permissions enabled for your role.


With access control, users with RW permission can access terminals to manage and monitor all the hosts on your network from a single platform to perform various tasks such as running scripts, executing commands, and troubleshooting issues.

You can access terminals using the **Open with Password** option if you know the password or click the **Name** hyperlink of the infra access group once you have been granted access to the group (See [Requesting Access to Terminals](#)).

Requesting Access to Terminals



Before you begin: Only users with RW permission can access hyperlinks to infra access groups for which you can request access.

To request access to terminal:

- Go to  (**Menu**) icon > **SSH+ > Access Control**.
The **Terminal Access Control** page is displayed.
- Click the **Name** hyperlink of the infra access group to which you want to request access.
The access request form is displayed.
- Enter the following fields:

Field description for Access Request section

Field	Description
* Access Mode	<p>Depending on your ACF permissions, select one of the options:</p> <ul style="list-style-type: none"> • AppViewX Terminal: Select this option if you want to connect to the hosts in the Infra access group via AppViewX Terminal. • Jump Server: Client: Select this option if you have mapped the infra access group to a jump server client or a set of jump server. clients. The jump servers clients are populated in the Device name box. You can search or select the ones you want access to. • Unmanaged Clients: Select this option to request access for a client not managed by AppViewX. You can select between generating keys via AppViewX or uploading your own public keys thus ensuring secure and customized access to unmanaged clients. <p>On clicking Download, the SSH keys are downloaded along with the host CA trust certificate that is to be added in the known host file. If the known host file exists, then copy the content from the downloaded file into it; in case the known host file is not present, then copy and paste the known_host file itself.</p> <p>The download option has a disabled PEM format field and an optional password field when AppViewX is selected for key generation. The signed keys are downloadable as a PEM file from the Infra Access Group inventory as long as the access request is valid. AppViewX signs the user-uploaded key with a certificate validity for the requested access duration.</p> <p>The user can choose to protect the downloaded keys (zip file) with a password or leave them unprotected. The download option is disabled when the PEM format field without a password for the upload key is selected.</p>
* Application Infra Name	This is the infra access group for which you are requesting access. This is a non-editable field.
* Access For	Select one of the options:

Field	Description
	<ul style="list-style-type: none"> • Self if you are requesting access for yourself • User for requesting access for another user. On selecting this option, the User field is displayed. Enter a user name or select user from the dropdown list.
*Available Jump Server(s) Client(s)	This field is displayed on selecting the Jump Server Client option. Select one or more jump servers clients from the section. Click the Refresh icon to see the latest list.
*Access Duration Type	Select Hours or Days .
*Access Duration Value	<p>You can key in a value in the text field after selecting Hours or Days option.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: The access duration kicks in from the time the administrator grants access to the time requested. Once the access duration elapses, the access is revoked.</p> </div>
Comments	Enter the reason for requesting access to the infra access group.
*Key Generation	<p>This field is displayed on selecting the Unmanaged Clients option.</p> <p>You can select between generating keys via AppViewX or uploading your own public keys thus ensuring secure and customized access to unmanaged clients.</p>
<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: Fields indicated with red asterisk (*) symbol are mandatory.</p> </div>	

4. Click **Request Access**.

A banner message, "*Access Request created successfully*" appears. The Access Status changes to *Pending Approval*. Once the administrator approves the request, the Access Status changes to *Approved*; else, it changes to *Access Denied*.

What to do next

You can directly access the terminal by clicking the **Name** hyperlink of the infra access group. See [Accessing Host Terminals](#).

Access Requests Hub

Any user with administrator privileges can use access request hub if you have ACF permissions enabled. This is a full-featured tool for access request management enhancing security governance, operational efficiency, and compliance through detailed oversight and control of user access within the system.

1. Go to  (**Menu**) icon > **SSH+ > Manage Access > Access Requests Hub**.

The **Access Requests Hub** page is displayed.

You can view the details of the access requests such as requestor, requested for, app infra name, access mode, access duration, approver, and requested date.

2. You can approve or reject requests that are in *Pending Approval* status by hovering over the access request to see the **Approve or Reject** button as shown.

Requestor	Requested For	App Infra Name	Access Mode	Access Duration	Approved By	Requested C
admin	admin	RW ssh_demo	AppViewX Terminal	1 hours	adm	Approve Reject

On approving the request, the **Access Status** column changes to **Accessible**, and the **Approve or Reject** button changes to **Revoke**.

3. You can revoke access requests that are in **Accessible** status by hovering over the access request to see the **Revoke** button as shown.

Requestor	Requested For	App Infra Name	Access Mode	Access Duration	Approved By	Requested C
admin	admin	RW test_access_revoke_key	AppViewX Terminal	1 hours		Revoke

4. You can use the **Search** option to search access requests by requester, requested for, app infra name, access mode, access duration, and approver.
5. You can sort and filter access requests by clicking the hyperlink on the top panel that displays the number of access requests by their statuses such as accessible, partial, pending, denied, expired, failed, and revoked. Clicking the hyperlink fetches those access requests and displays details of the access requests.

Approving Access Requests



Note:

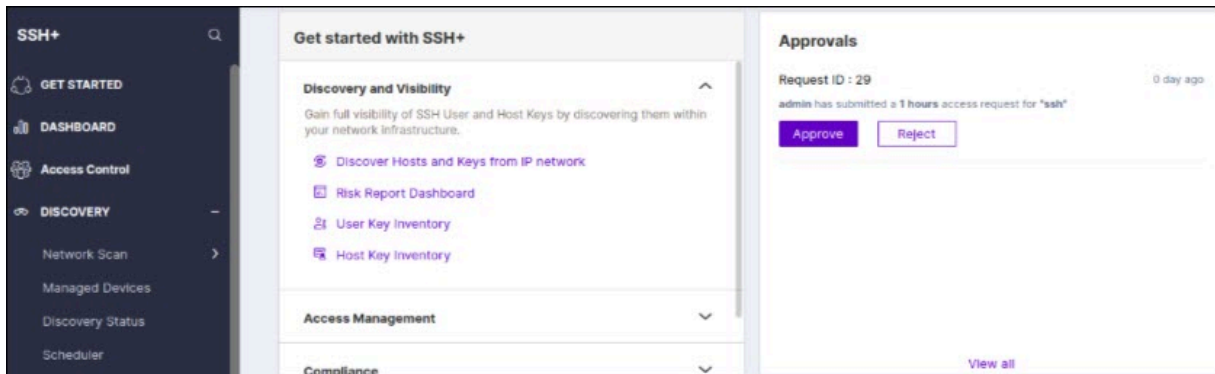
To approve or reject access requests, the ACF permissions must be enabled by the administrator.

You may also see all requests for terminal access on the **Get Started** page in the **Approvals** section.

To approve or reject access request:

1. Go to (**Menu**) icon > **SSH+** > **Get Started**.

The access requests are displayed in the **Approvals** section.



2. Based on the requestors and their access rights, you can approve or reject the request.

On approving the request, the **Access Status** column on the **Terminal Access Control** page turns to:

- **Accessible:** Status when key provisioning is successful for all hosts in the infra access groups.
- **Partially Accessible:** Status when key provisioning fails for some hosts in the infra access groups.
- **Failed:** Status when key provisioning fails for all hosts in the infra access groups.

On approving an access request, the user is granted access to the specified infra access group and hosts for the requested duration. The users will be able to perform authorized tasks on those resources. On rejecting an access request, the user is denied access to the requested resources.

Viewing Terminal Access Control Page

To view the Terminal Access Control page:

- Go to (**Menu**) icon > **SSH+** > **Access Control**.

The **Terminal Access Control** page is displayed.

Field description for Terminal Access Control page

Field	Description
Name	Displays infra access groups based on your RW permission. Click the hyperlink to request access to the group. See Requesting Access to Terminals .
Host(s) Count	Displays the count of the hosts associated with the infra access group for which you have access.
Access Status	<p>Displays the access status of the infra access group:</p> <ul style="list-style-type: none"> • N/A: Initial status when you do not have access to the infra access group. • Pending Approval: Status when you have sent an access request and are awaiting approval from the administrator. <p>When the administrator approves access to the infra access group, the status is set to Approved; else, it is marked as Access Denied.</p> <ul style="list-style-type: none"> • Provisioning In Progress: Status when key provisioning begins for all hosts in the infra access groups. • Accessible: Status when key provisioning is successful for all hosts in the infra access groups. • Partially Accessible: Status when key provisioning fails for some hosts in the infra access groups. • Failed: Status when key provisioning fails for all hosts in the infra access groups. • Expired: Status when your access to the group is expired for the requested duration. You can request access for the same group again by raising an access request. <p>Clicking the Access Status link against an infra access group displays a pop-up message.</p>
Logs	Click View to open the log. The log displays details about the user who has access to the infra access group, access mode (whether SSH if it was accessed after granting approval or credential if password was used to access the terminal), status (whether access is active or expired), when the access request was initiated, when the access was terminated, and for how long the access was granted.

Accessing Host Terminals


You can access the host terminals in any of the following ways:

- If you have the password to the infra access server, then click the **Open with Password** option on the **Access Request** page.
- If you do not have the password to the infra access server, then request access as described in [Requesting Access to Terminals](#). Once access is granted, you can:
 - Click the **Name** hyperlink of the infra access group.
 - Access the terminals from the **Device Management** page or the **Host Inventory** page by clicking the



(**Terminal Connect**) icon.

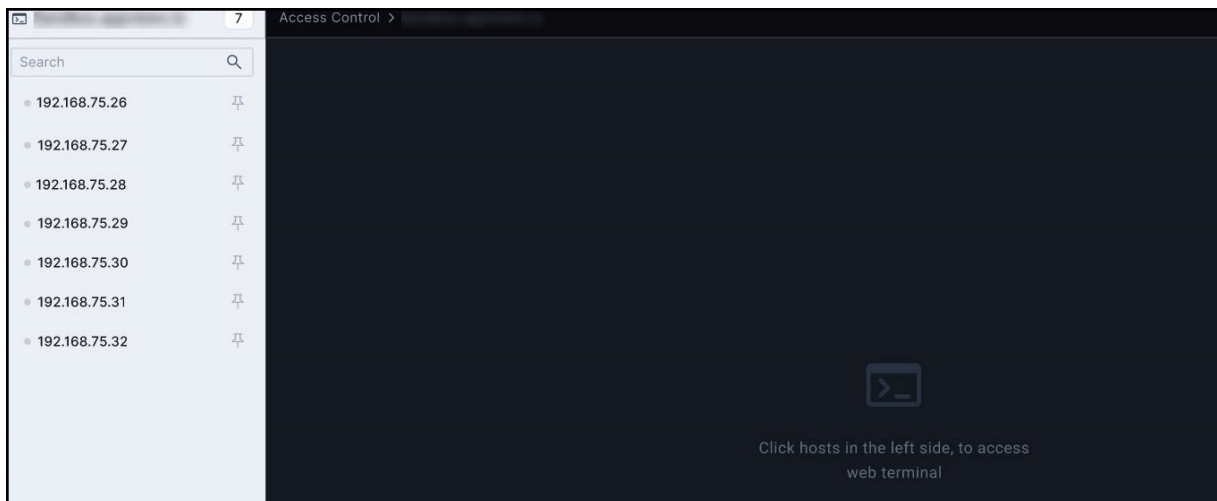
To access the host terminal:

1. Go to  (**Menu**) icon > **SSH+** > **Access Control**.

The **Terminal Access Control** page is displayed.

2. Click the **Name** hyperlink of the infra access group.



The terminal page opens with all the associated hosts as shown.



3. Click any of the hosts to access the web terminal.
4. Enter your credentials.

You can now perform various tasks such as running scripts, executing commands, and troubleshooting issues.

**Note:**

- You can access more than one host at a time within each infra access group. On selecting multiple hosts, the hosts open in the split screen view. You can alternate between split or tab view by clicking using () icon on the RHS of the page.
- You can pin only four terminals for better access and management using the  (**Pin**) icon.

Configuring Provision Settings

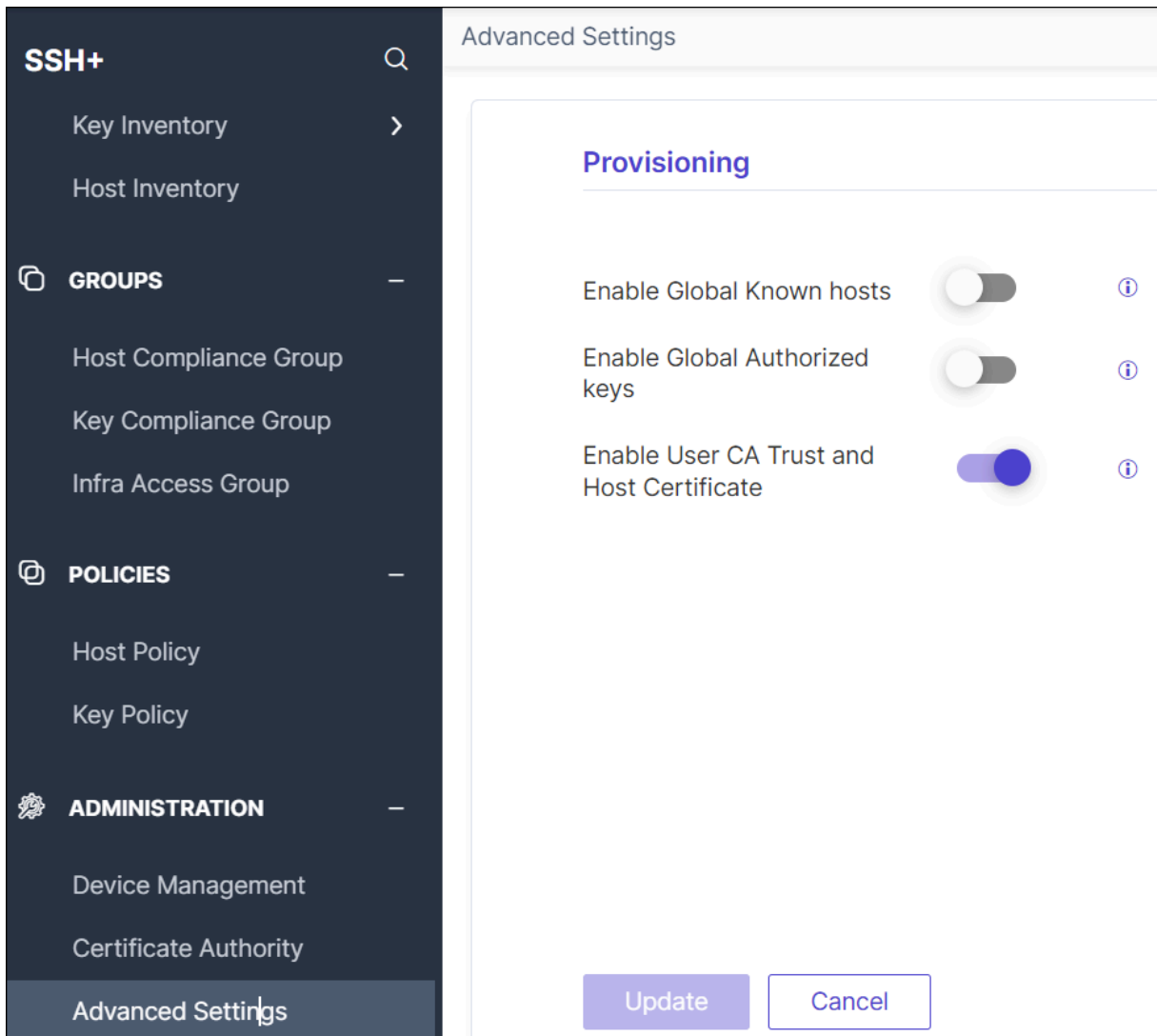
You can gain comprehensive control over key and certificate provisioning with the options available under **Advanced Settings > Provisioning** section. The options are:

- **Provision to Enable Global Known hosts**
- **Provision to Enable Global Auth keys**
- **Provision to Enable User CA Trust and Host Certificate**

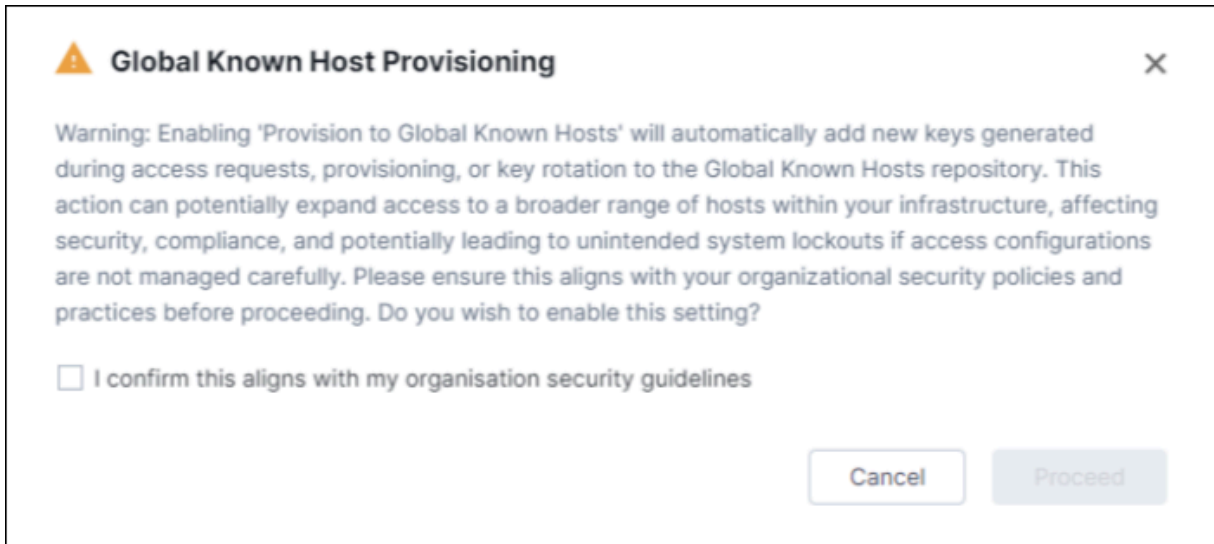
Procedure

1. Go to  (**Menu**) icon > **SSH+ > Administration > Advanced Settings**.

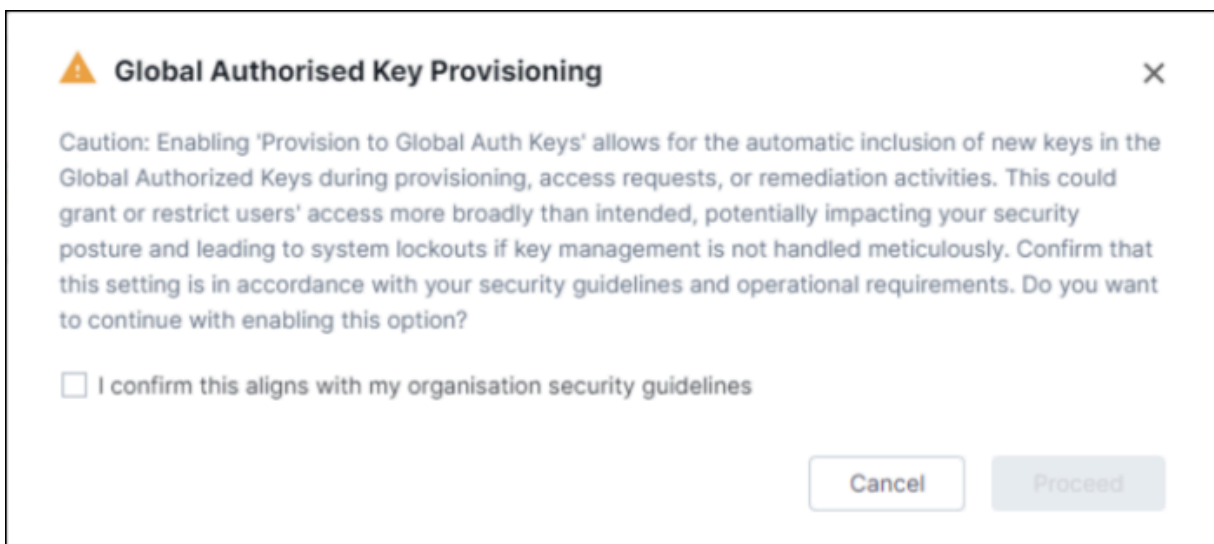
The **Advanced Settings** page is displayed.



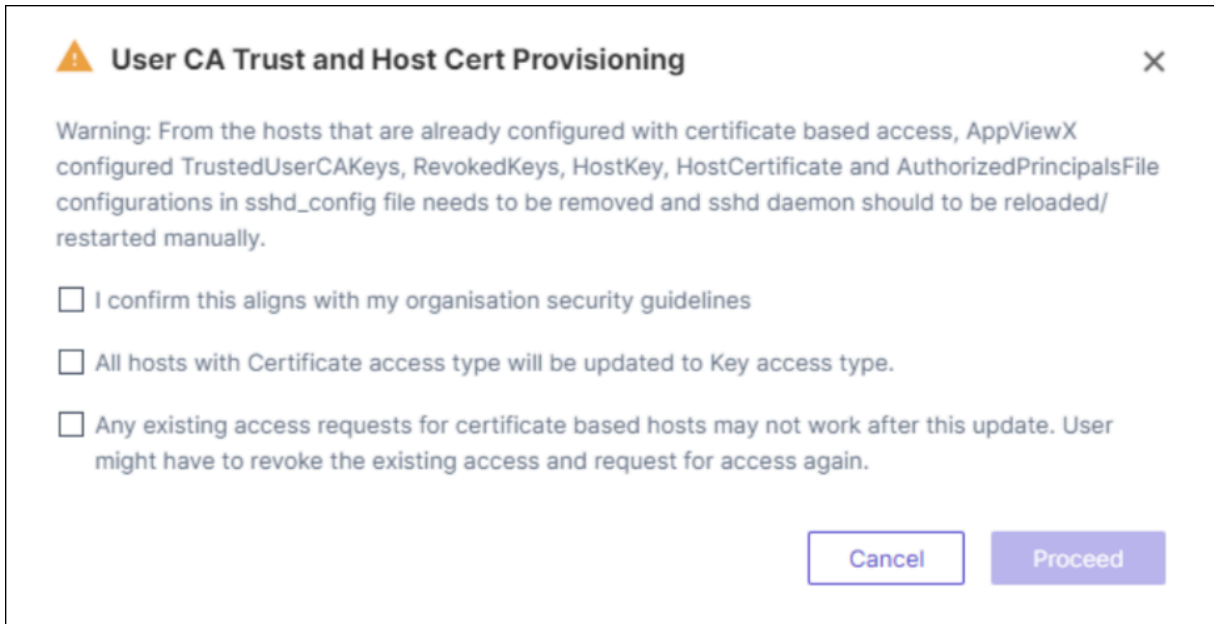
- 2. Enable Global Known hosts:** By default, this option is disabled. Enabling this option allows to generate a Global Known Hosts configuration, ensuring that all new keys/certificates generated during access requests, provisioning, or remediation activities (such as key rotation) are added to the Global Known Hosts repositories. Enabling this option may have implications for your network.



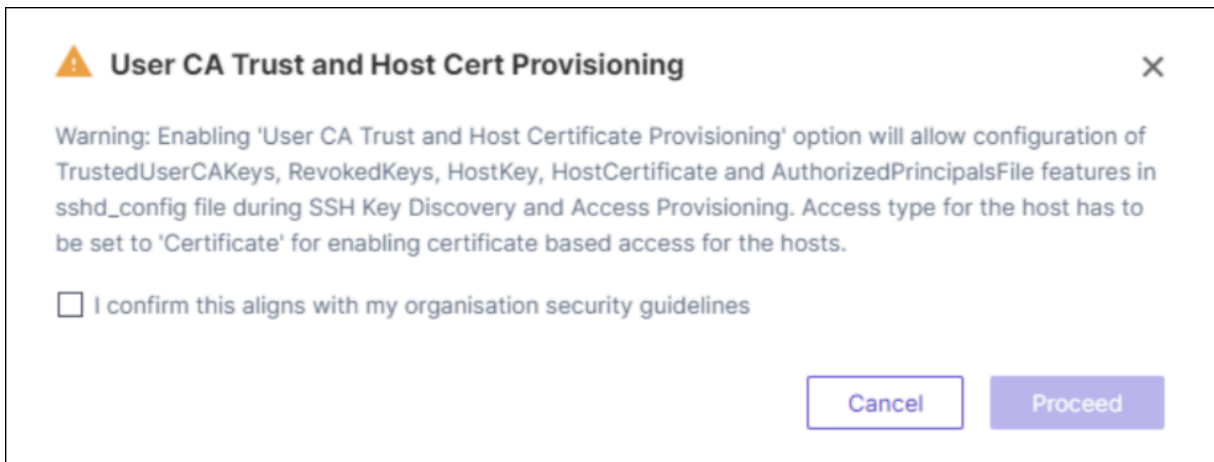
3. **Enable Global Authorized keys:** By default, this option is disabled. Enabling this option allows to establish a Global Auth Keys configuration, ensuring that all new keys/certificates generated during access requests, provisioning, or remediation activities (such as key rotation) are added to the Global Authorised Keys repositories. Enabling this option may have implications for their network.



4. **Enable User CA Trust and Host Certificate:** By default, this option is enabled. This option allows certificate based authentication for the devices. If you disable this option, the following **User CA Trust and Host Certificate** warning window opens. Select the desired options, and then click **Proceed**.



If you enable this option again, you get the following **User CA Trust and Host Certificate** warning window opens. Confirm the checkbox, and then click **Proceed**.



5. Click **Update**. A message, *Provisioning settings saved successfully*, appears.

Setting ACL Permissions to Resources


You can also assign/unassign resources access control list (ACL) permissions for infra access groups and key compliance groups. Depending on the ACL permissions set, resources will have partial, full, or no access to the groups and related actions.

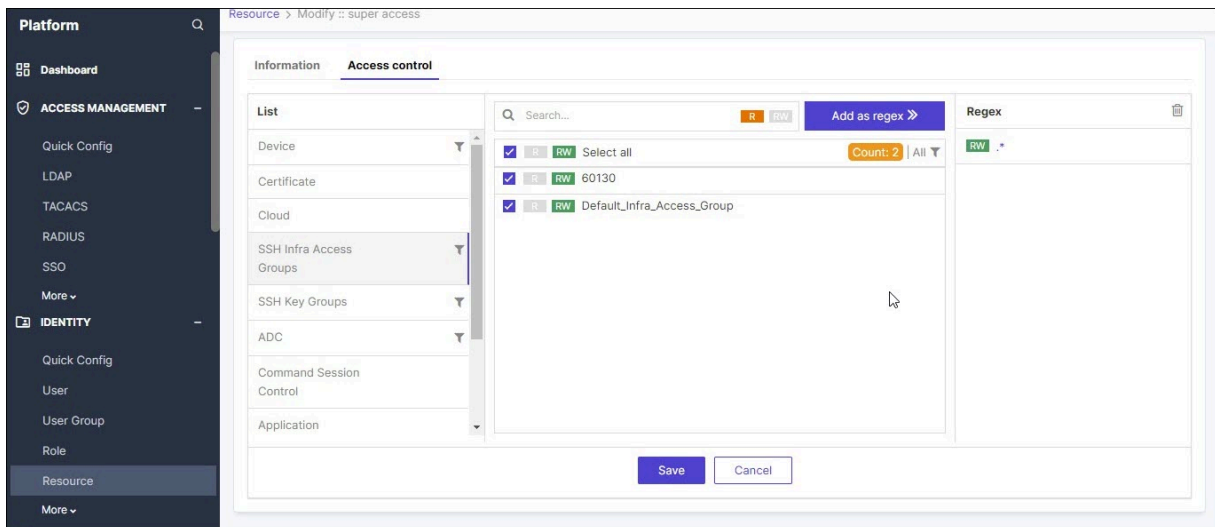
Setting ACL Permissions to Infra Access Groups

You can assign/unassign resources the following access control list (ACL) permissions to infra access groups:

- **RW** denotes that users have *Read-Write* permission to the infra access group along with hyperlinks to request access to the group on the **Access Control** page. Users can modify or delete infra access groups.
- **R** denotes that users only have *Read* permissions to the infra access groups. Users cannot request access or modify or delete infra access groups.
- If neither R nor RW permission is assigned, then users will not be able to even view the infra access group.

To assign ACL permission to infra access groups:

1. Go to  (Menu) icon > **Platform** > **Identity** > **Resource**.
2. Click a resource name.
3. Click **Access Control** tab and select **SSH Infra Access Groups** from the list.



The infra access groups are displayed on the right.

4. Select the infra access group for which you want to assign permission by clicking the check box.
5. Click **R** or **RW** button to assign the read or read-write permission.
6. Click **Save**.

A message, *Successfully assigned/unassigned SSH groups data for resource super access*, appears.



Note:

On deleting a resource, all ACL permissions from the associated infra access groups are also removed.

Setting ACL Permissions to Key Compliance Groups

You can assign/unassign resources the following access control list (ACL) permissions to key compliance groups:

- **RW** denotes that users have *Read-Write* permissions to the key compliance groups. Users can modify and delete key compliance groups. Users can view the keys associated with the key compliance groups and perform any of the actions in the inventory.
- **R** denotes that users have *Read* permissions to only view details displayed on the **Key Compliance Group** page. Users can only view the keys associated with the key compliance groups but cannot perform any action in the inventory.
- If R or RW permission is not assigned, then users will not be able to view the key compliance groups and the keys associated with them.

To assign ACL permission to key compliance groups:

1. Go to  (Menu) icon > **Platform** > **Identity** > **Resource**.
2. Click a resource name.
3. Click **Access Control** tab and select **SSH Key Groups** from the list.

The screenshot displays the 'Access control' configuration page for the resource 'super access'. The left sidebar shows the navigation menu with 'Platform' selected, and 'Identity' > 'Resource' highlighted. The main content area is divided into 'Information' and 'Access control' tabs, with 'Access control' active. Under 'Access control', the 'SSH Key Groups' section is expanded. A table lists three key groups: 'Select all', 'Orphan_pubkey_group', and 'Default_Key_Group'. Each row has a checked checkbox and a permission indicator 'RW'. A search bar at the top of the list shows 'Count: 2' and 'All'. To the right, the 'Regex' field contains the pattern '*'. At the bottom of the configuration area, there are 'Save' and 'Cancel' buttons.

The key groups are displayed on the right.

4. Select the key group for which you want to assign permission by clicking the check box.
5. Click **R** or **RW** button to assign the read or read-write permission.
6. Click **Save**.

A message, *Successfully assigned/unassigned SSH groups data for resource super access*, appears.




Note:

On deleting a resource, all ACL permissions from the associated key compliance groups are also removed.

Adding or Deleting Regex Patterns

With Regex (Regular Expression) support, you can dynamically map user groups to infra access groups and key groups within the SSH+ module. This feature allows for the automatic mapping of newly created infra access groups and key groups that match specified Regex patterns, thus streamlining the process and reducing manual overhead.

To add or delete Regex patterns:

1. Go to  (**Menu**) icon > **Platform** > **Identity** > **Resource**.
2. Click a resource name.
3. Click **Access Control** tab and select **SSH Infra Access Groups** or **SSH Key Groups** from the list.
4. Type a regex pattern in the **Search** box and click **Add as regex**.


Examples of regex pattern include:

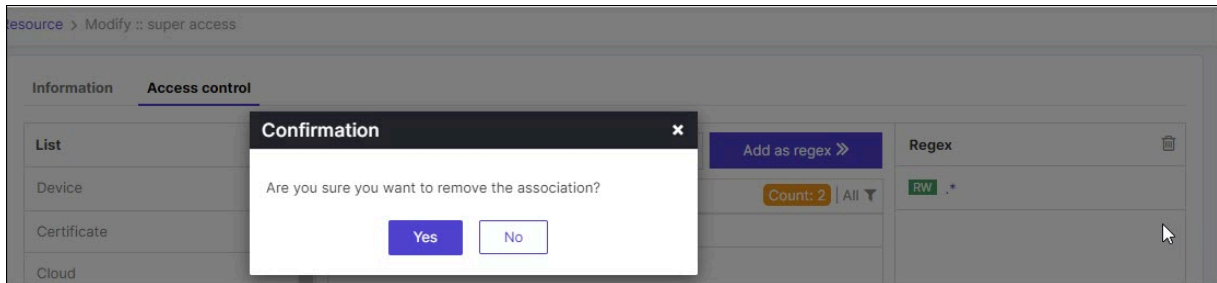
- `.*<text>`
- `<text>.*`
- `<text>.*<text>`
- `.*`

where `<text>` can be alphabets, numbers, and special characters to create these entries in the pages.

A message that the Regex pattern is successfully added for admin appears. Based on the matched regex, permission is automatically added to that resource. The newly added regex appears under the Regex column on the RHS of the page.

5. To individually delete a regex pattern, select the regex pattern from the **Regex** column and click **X**

against the name. To delete all regex patterns, click  (Delete) icon. A confirmation message appears as shown.



Click **Yes** and a message that it is successfully deleted from the matching groups appears.

Adding Infra Access Groups

Infra Access Group

- Creating an access group lets you enable access to a user or a group of users to all the hosts in the access group with a single instruction/selection.
- A host can be associated with one or more access groups. A host can remain unassociated with an access group as well.
- Infra access groups are created in different ways:
 - Dynamic addition of new Infra access group during host addition in SSH host inventory or Server inventory (excluding AWS EC2 servers) and SSH discovery

- **Manually created through the Infra Access Group tab**

Manually created access groups are just groups and do not have the above-mentioned intelligence. It is meant to create access group on-premises devices as well. See [Adding Infra Access Group](#).

- **Auto-create during the cloud host scan**


Auto-created access groups are created automatically by reading the AWS tags of the host during cloud host discovery. The system forms the groups dynamically based on these AWS tags. The cloud host discovery does a periodic scan.

If Hosts are added to an existing infra access group that is in "Accessible" state or "Partially accessible", The host is automatically provisioned with access and respective users assigned to the infra access group can access the respective host.. (Refer Access Request hub). If Hosts are removed from an infra access group that is in "Accessible" state or "Partially accessible", The application ensures withdrawal of access that was provisioned to the host previously and respective users assigned to the infra access group loses access to the specific host.

Adding Infra Access Group

Before you begin: You can access this functionality only if you have the ACF permissions enabled for your role.

To add an infra access group:

1. Go to  (Menu) icon > **SSH+** > **Groups** > **Infra Access Group**.


The **Infra Access Group** page is displayed.

2. On the command bar, click **+Add New Groups**.

The **Infra Access Group > Create** page is displayed.

3. In the **General Information** section, enter the following:

Field description for General Information section

Field	Description
*Group Name	Enter a unique name.
Description	Enter details regarding the group stating the purpose.
Managed Devices/ Instances	Select the managed devices/instances to be associated with the group.
Jump Server Client	Select the jump servers clients to be associated with the group.
<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: Fields indicated with red asterisk (*) symbol are mandatory. </div>	

4. Click **Create**.

An Infra Access Group is created and displayed in the inventory.

5. Click the number displayed in the **Associated Machine Count** column.

The popup window displays the **List of Devices** associated with the machine.


What to do next:

Users with RW permissions can modify or delete an infra user group by selecting the checkbox against the group name and selecting **Modify** or **Delete** from the **Actions** menu.

Removing Hosts from Infra Access Group

You can remove one or multiple hosts from an infra access group thus revoking access to those hosts (if access was granted previously) enhancing security and maintain proper access control within the system.

To remove hosts from an infra access group:

1. Go to  (**Menu**) icon > **SSH+ > Groups > Infra Access Group**.

The **Infra Access Group** page is displayed.

2. Select the infra access groups from which you want to remove the hosts by selecting the checkbox against the group.
3. On the command bar, click **Actions > Modify**.

The **Infra Access Group > Modify** page is displayed.

4. Scroll down to **Managed Devices / instances** and select the hosts you want to revoke access.
5. Click **Update**.

A workflow is triggered in the background and once the status of the service request is displayed as *Completed*, the hosts are removed from the infra access group.



Tip:

You can also remove hosts from the **Host Inventory** by selecting the hosts for which you want to revoke access. Go to **Actions > Modify** and scroll down to **Application Infra Access Group** and delete the infra access groups from which you want to remove the host.


Viewing Infra Access Group

To view infra access group:

1. Go to  (**Menu**) icon > **SSH+ > Groups > Infra Access Group**. The **Infra Access Group** page is displayed.

Field description for Infra Access Group section

Field	Description
Group Name	Displays the name of the infra access group. The infra access group can have the following label based on your ACL permissions:

Field	Description
	<ul style="list-style-type: none"> • RW denotes that you have <i>Read-Write</i> permission to the infra access group along with hyperlinks to request access to the group on the Access Control page. You can modify or delete infra access groups. • R denotes that you only have <i>Read</i> permissions to the infra access groups. Users cannot request access or modify or delete infra access groups. • If R or RW permission is not assigned, then you will not be able to view the infra access groups.
Description	Displays the details of the key compliance group.
Associated Machine Count	Click the hyperlink to open a pop-up window displaying all the hosts associated with the infra access group.
Client	Displays the client associated with the infra access group.
<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: Fields indicated with red asterisk (*) symbol are mandatory. </div>	

Managing Host Key and User Key Inventories

Overview

Before you begin: You can access this functionality only if you have the ACF permissions enabled for your role.

SSH uses SSH keys to encrypt communication with a remote system. SSH keys usually come in pairs comprising a public and a private key and are used to grant access to authorized personnel to critical systems such as cloud, on-premise servers, and network devices. The public key (or host key) can be freely shared and is used to encrypt data sent to remote server or user; the private key (or user key) must only be with the user and be kept secret as it is used to decrypt data sent from remote server or user. It is generated by the local machine and kept in a secure location.

This chapter guides you through all the actions that can be carried out on the keys. Actions on keys such as deletion, status change, export, and upload of keys are possible.





From the **Inventory** page, you can:

- View details of the user key and host key. See [Viewing User/Host Key Inventory](#).
- Delete, change status, export, upload user key or host key. See [Actions in Key Inventory](#).
- Discover and add hosts to the host inventory. See [Adding Host](#).
- View details of the hosts. See [Viewing Host Inventory](#).
- Decommission inactive hosts or delete active hosts, add credentials to the host server, fetch host key from the host inventory, and export details of the host to a .csv or .xls format. See [Actions in Key Inventory](#).

Key Inventory

The keys inventory displays the details about the keys. You can also perform various actions on the keys by selecting them from the **Actions** dropdown list. Selecting any action directs you to the respective action page under the **Action** section.

The **User Key Inventory** page displays:

- Key Pair with Passphrase indicated by ()
- Key Pair without Passphrase indicated by ()
- Public Key Only indicated by ()
- Private Key Only indicated by ()


The **Host Key Inventory** page displays:

- Active hosts indicated by green. These hosts can be deleted.
- Inactive hosts indicated by red.

Viewing User/Host Key Inventory

The User/Host Keys tab displays the total number of weak, shared, orphan, and suspicious keys in the key discovery status. Click the number hyperlink to drill down on the metrics. This helps you track the progress of the key discovery efforts, identify any potential security risks, and prioritize the remediation actions.

To view the user/host key inventory:

1. Go to  (**Menu**) icon > **SSH+** > **Inventory** > **Key Inventory**.
2. Select **User Key Inventory** or **Host Key Inventory**.

The **SSH+::User Key** page is displayed.

Field Description in User/Host Key Inventory

Field	Description
Key name	Displays the auto-generated unique name created for the key.
Certificate Count	<p>Displays the number of certificates associated with the key. Click the hyperlink to see a popup window with the following fields:</p> <ul style="list-style-type: none"> • Principals: Principals contain the identities associated with a certificate. A principal can be a host or a username associated with the certificate.. • CA name: Displays the CA name associated with the key. • Serial Number: Displays the serial number of the key. • Certificate Status: Displays the certificate status of the key. • Valid From: Displays the start date of the key validity. • Valid To: Displays the end date of the key validity. • Expires In: Displays how long before the key expires. • Extensions: Displays the extensions of the key.
Key Compliance Group/ Host Compliance Group	<p>Displays the name of the group associated with the key. The key/host compliance group can have the following label based on your ACL permissions:</p> <ul style="list-style-type: none"> • R denotes that you have <i>Read</i> permission to the key/host compliance group, although there is no hyperlink available to request access to the key/host compliance group; however, you can view the logs. • RW denotes that you have <i>Read-Write</i> permission to the key/host compliance group along with hyperlinks to request access to the key/host compliance. You can modify and delete the key/host compliance groups, and view the logs. • If you do not have either R or RW permission, then you will not be able to view the key/host compliance group.
Encryption	This field is applicable only for the user key. Displays the encryption type of the key.
Length	Displays the bit-length of the key.
Age	Displays the age of the key.

Field	Description
	For example, if the key was created 5 days earlier, it displays as <i>5 Days</i> .
Client Endpoint(s)	Displays the count of client machines associated with the key. You can view the list of the hosts associated with the key as a client machine.
Host Endpoint(s)	Displays the count of host machines associated with the key. You can view the list of the hosts associated with the key as a host machine.
Risk Status	Displays the status of the key as weak, shared, orphan, or suspicious.
Status	Displays the status of the key. The statuses are: <ul style="list-style-type: none"> • Managed • Monitored
Associated Users	This field is applicable only for the user key inventory. Displays the users associated with the key.
File Path(s)	Displays where the key file is located on the host.
Comment	Displays any comments with regards to the key.
Validity	Displays the validity of the key.
Fingerprint	Displays the fingerprint of the key.





Actions on User Key/Host Key Inventory

You can perform the following actions from the **Key Inventory** page.



Action description on User/Host Key Inventory page

Action	Description
Change status	Users with RW permission can change the status of a key to Managed or Monitored .

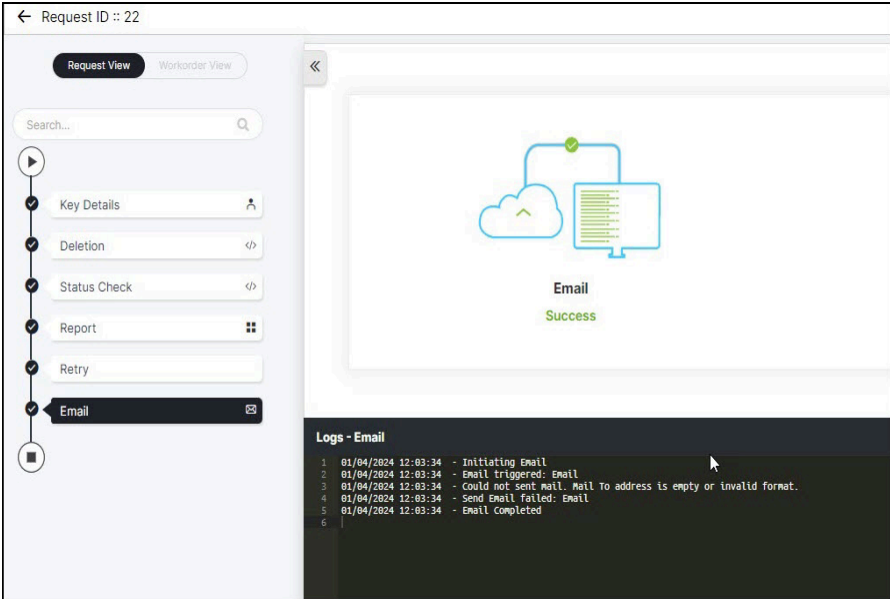
Action description on User/Host Key Inventory page (continued)

Action	Description																
Export	You can export the user or host key details from their respective inventory in .csv or .xls format.																
Upload User SSH key	<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  Note: This field appears only for User Key Inventory. </div> <p>Field description for Upload SSH key section</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>*Key File</td> <td>Click Search icon to browse for the file.</td> </tr> <tr> <td>*Key Group</td> <td>Select key group from the dropdown list. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: A key is linked to a key group, and this key group is further connected to a policy. Based on the selection of the key group, it is determined if the key needs a work order approval. The key is also checked for compliance with the key policy associated with the key group. </div> </td> </tr> <tr> <td>*Key Name</td> <td>Enter a unique name for the key to facilitate easy identification.</td> </tr> <tr> <td>Passphrase</td> <td>Enter a passphrase</td> </tr> <tr> <td>Confirm Passphrase</td> <td>Enter the passphrase again to confirm.</td> </tr> <tr> <td>*Validity</td> <td>Select validity from the dropdown list. This determines the duration for which the key is valid.</td> </tr> <tr> <td>Comment</td> <td>Enter remarks specific to the key.</td> </tr> </tbody> </table>	Field	Description	*Key File	Click Search icon to browse for the file.	*Key Group	Select key group from the dropdown list. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: A key is linked to a key group, and this key group is further connected to a policy. Based on the selection of the key group, it is determined if the key needs a work order approval. The key is also checked for compliance with the key policy associated with the key group. </div>	*Key Name	Enter a unique name for the key to facilitate easy identification.	Passphrase	Enter a passphrase	Confirm Passphrase	Enter the passphrase again to confirm.	*Validity	Select validity from the dropdown list. This determines the duration for which the key is valid.	Comment	Enter remarks specific to the key.
Field	Description																
*Key File	Click Search icon to browse for the file.																
*Key Group	Select key group from the dropdown list. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: A key is linked to a key group, and this key group is further connected to a policy. Based on the selection of the key group, it is determined if the key needs a work order approval. The key is also checked for compliance with the key policy associated with the key group. </div>																
*Key Name	Enter a unique name for the key to facilitate easy identification.																
Passphrase	Enter a passphrase																
Confirm Passphrase	Enter the passphrase again to confirm.																
*Validity	Select validity from the dropdown list. This determines the duration for which the key is valid.																
Comment	Enter remarks specific to the key.																






Action description on User/Host Key Inventory page (continued)

Action	Description	
	Field	Description
Revoke		<div data-bbox="548 422 1409 552" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: Fields indicated with red asterisk (*) symbol are mandatory. </div> <p>Users with RW permission can revoke certificates that are associated with keys that have a private key and key pair (public + private). If the selection has even one key that is a public key, then revoke is disabled.</p>
Rotate		<div data-bbox="548 636 1409 766" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: This field appears only for User Key Inventory. </div> <p>Users with RW permission can rotate selected user keys or host keys based on the rotation configuration outlined in their corresponding key policies. Keys selected for rotation are automatically backed up and stored in a secure encrypted format in the Recently Rotated Keys. The details of backup are available in the audit log. On successful completion of backup, a message appears in the audit log, <i>Backup completed for the <key type> for action <action> with name <key name> with fingerprint <key fingerprint> with group name <key group name> by the user <user name></i>.</p> <p>On selecting keys for rotation, a confirmation message appears. On confirming, the rotate operation is triggered via workflow. To check the status and reports, go to Automation > Service Request > All and select your request from All requests.</p>




Action description on User/Host Key Inventory page (continued)

Action	Description
	 <p>The newly rotated key adheres to the following naming convention: KEYTYPE_TIMESTAMP, where key type denotes the encryption algorithm of the key while timestamp is when you have rotated the key in the yyyyMMdd_HHmss_SSS_counter format where:</p> <ul style="list-style-type: none"> • <i>yyyy</i> denotes the year • <i>MM</i> denotes the month • <i>dd</i> denotes the date • <i>HH</i> denotes the hours • <i>mm</i> denotes the minutes • <i>ss</i> denotes the seconds • <i>SSS</i> denotes the milliseconds • <i>counter</i> denotes the number of keys being rotated <p>For example, ECDSA_20230908_123456_789_1 implies that the rotated key follows the ECDSA algorithm and was generated on September 8, 2023, at 12:34:56.789 GMT.</p> <p>Upon successful rotation of the key, the Comments field is updated.</p>



Action description on User/Host Key Inventory page (continued)

Action	Description
	<p> Important:</p> <p>Best practices before rotating host keys:</p> <div data-bbox="618 485 1403 814" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <p>The following points are applicable when the Enable Global Known hosts option is enabled under  (Menu) icon > SSH+ > Administration > Advanced Settings. Enabling this option may have implications for your network.</p> </div> <ol style="list-style-type: none"> 1. If the <i>global known host</i> file is not present, then AppViewX will create one in the root folder by including all public keys from users in the global known host file. 2. Prior to host key rotation, update the <i>global known host</i> file. 3. The old public key is deleted and the new key is replaced in the <i>global known host</i> file. <p>Best practices before rotating user keys:</p> <div data-bbox="618 1230 1403 1560" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <p>The following points are applicable when the Enable Global Authorized keys option is enabled under  (Menu) icon > SSH+ > Administration > Advanced Settings. Enabling this option may have implications for your network.</p> </div>

Action description on User/Host Key Inventory page (continued)

Action	Description
	<div data-bbox="537 327 1417 646" style="border: 1px solid #ccc; padding: 10px; background-color: #fff9c4;">  <p>1. If the <i>global authorized key</i> file is not present, then AppViewX will create one in the root folder for each login user with privileged user permission.</p> <p>2. Prior to user key rotation, update the <i>global authorized key</i> file.</p> <p>3. The old public key is deleted and the new key is replaced in the <i>global authorized key</i> file.</p> </div> <div data-bbox="537 684 1417 947" style="border: 1px solid #ccc; padding: 10px; background-color: #fff9c4; margin-top: 10px;">  <p>CAUTION:</p> <p>Rotating keys can result in access loss and authentication problems if AppViewX does not have access to all the infrastructure information. Proceed with caution and ensure proper backup and alternative authentication methods are in place.</p> </div>
Delete	<p>Users with RW permission can:</p> <ul style="list-style-type: none"> • Delete from Endpoints: Deletes the keys from the host endpoints. Keys selected for deletion from endpoints are automatically backed up and stored in a secure encrypted format in the database. The details of backup are available in the audit log. On successful completion of backup, a message appears in the audit log, Backup completed for the <key type> for action <action> with name <key name> with fingerprint <key fingerprint> with group name <key group name> by the user <user name>. <div data-bbox="558 1465 1417 1791" style="border: 1px solid #ccc; padding: 10px; background-color: #e1f5fe; margin-top: 10px;">  <p>Note:</p> <ul style="list-style-type: none"> • If you try deleting keys from hosts with only one key, then a warning message about the potential service disruption is displayed. • On selecting keys for deletion from endpoints, a confirmation message appears. On confirming, the delete operation is triggered via workflow. To check the status and reports, go to </div>


Action description on User/Host Key Inventory page (continued)

Action	Description
	<p data-bbox="574 331 1393 401">  Automation > Service Request and select your request from All requests. </p> <div data-bbox="659 415 1406 1010" style="border: 1px solid #ccc; padding: 10px;"> <p data-bbox="667 422 776 443">Request ID :: 22</p> <p data-bbox="711 470 857 491">Request View Workorder View</p> <p data-bbox="678 527 873 548">Search...</p> <ul data-bbox="678 604 873 842" style="list-style-type: none"> <li data-bbox="678 604 873 632">Key Details <li data-bbox="678 653 873 680">Deletion <li data-bbox="678 701 873 728">Status Check <li data-bbox="678 749 873 777">Report <li data-bbox="678 798 873 825">Retry <li data-bbox="678 846 873 873">Email <div data-bbox="911 457 1398 835" style="border: 1px solid #ccc; padding: 10px; text-align: center;">  <p data-bbox="1117 695 1154 716">Email</p> <p data-bbox="1117 716 1154 737">Success</p> </div> <pre data-bbox="922 842 1398 1003" style="font-family: monospace; font-size: 0.8em;"> Logs - Email 1 01/04/2024 12:03:34 - Initiating Email 2 01/04/2024 12:03:34 - Email triggered: Email 3 01/04/2024 12:03:34 - Could not send mail. Mail to address is empty or invalid format. 4 01/04/2024 12:03:34 - Send Email failed: Email 5 01/04/2024 12:03:34 - Email completed 6 </pre> </div> <p data-bbox="537 1108 1393 1178"> <ul style="list-style-type: none"> • Delete from Inventory: Deletes the keys from the AppViewX inventory and not from the actual hosts. </p>

Actions on Key Groups

You can perform any of the following actions from the user key and the host key inventories:

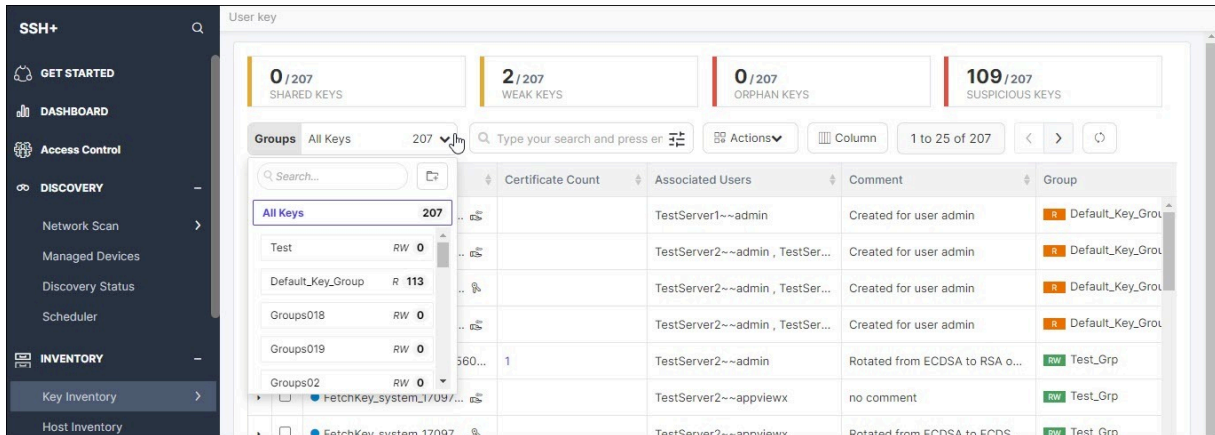
- Filter based on key groups
- Search based on key groups
- Create new key groups and associate keys to them


1. Go to  (Menu) icon > **SSH+ > Inventory > Key Inventory**.
2. Select **User Key Inventory** or **Host Key Inventory**.

The **SSH+::User Key Inventory/Host Key Inventory** page is displayed.

- To filter the key groups, click the **Groups** dropdown list and select a key group from the list. By default, **All Keys** is selected. The key groups are sorted in an alphabetical order displaying key count against each of the key group.

The filtered key group is displayed.



- To search further among the key group results, you can provide a search criterion in the **Search** box within the **Groups** dropdown list.
- To create a new key compliance group and associate keys with it, click  (**Create a group**) icon beside the **Search** bar within the **Groups** dropdown list.



The **Key Compliance Group > Create** page is displayed.

- Follow the steps as explained in [Adding Key Compliance Group](#) to create a group.

Advanced Search

With the advanced search functionality, you can enable precise identification and management of keys. This not only enhances the platform's usability and security but also facilitates efficient key management process.

The SSH+ Key Inventory search supports more granular and flexible searches across various key attributes, including client endpoints, host endpoints, associated users, encryption types, key length, key name, group, file path, fingerprint, and comment.

- Go to  (**Menu**) icon > **SSH+ > Inventory > Key Inventory**.
- Select **User Key Inventory** or **Host Key Inventory**.
The **SSH+::User Key Inventory/Host Key Inventory** page is displayed.
- Click  (**Advanced Search**) icon in the **Search** bar.

The **Advanced Search** window appears. By default, client endpoints, host endpoints, associated users, encryption types, key length, key name, file path, fingerprint, and comment are selected. To add or remove or reposition the search attributes, click **Configure Advance Search Attributes**.

4. Select the **AND** or **OR** condition. By default, **AND** is selected.
5. Select **is**, or **is not**, or **contains** from the dropdown list of each field and specify the search value. By default, **is** is selected.


You can specify more than one search value using commas.

6. Click **Search**.

The search results are displayed.

Actions on Recently Deleted Keys

All the keys that are deleted from the user key and host inventories are available in **Recently Deleted Keys**. You can access this functionality only if you have the ACF permissions enabled for your role.

1. Go to  (**Menu**) icon > **SSH+** > **Inventory** > **Key Inventory**.
2. Select **Recently Deleted Keys**.

The **Recently Deleted Keys** page is displayed.

3. You can perform any of the following actions:
 - **Viewing deleted keys:** The deleted user or host keys from endpoints are made available in the **User Keys** or **Host Keys** for a period of 30 days. The inventory lists key details including associated users, key type, client and host endpoints, who deleted the key, deletion date, and the days left before a key is permanently deleted.
 - **Restoring deleted keys:** To restore keys that were deleted by mistake or are needed again, simply select them and select **Restore** from the **Actions** menu. A message appears, *You are about to restore the selected key(s) back to their original location(s). This action will reinstate the key(s) and make them active again. Are you sure you want to proceed with the restoration?* Click **Confirm Restoration** to proceed.

Audit logs for the action when:

- **restoration succeeded:** Restore action triggered by the user <username> for uuids <uuid list> completed successfully for backup <user or host key>
- **restoration failed:** Restore action triggered by the user <username> for uuids <uuid list> failed for backup <user or host keys>



Note:

<uuid> is the unique ID of the key.

- **Deleting keys permanently:** You can permanently delete keys before the 30-day window by selecting the keys and then selecting **Delete Permanently** from the **Actions** menu. A message appears, *You are about to permanently delete the selected key(s). This action cannot be undone, and the key(s) will be irrecoverably removed from the system. Are you sure you want to proceed with permanent deletion?* Click **Confirm Deletion** to proceed.

Audit logs for the action when:

- **delete succeeded:** Delete action triggered by the user <username> for uuids <uuid list> completed successfully for backup <user or host keys>


- **delete failed:** Delete action triggered by the user <username> for uuids <uuid list> failed for backup <user or host keys>

**Note:**

<uuid> is the unique ID of the key.

Actions on Recently Rotated Keys

All the keys that are rotated from the user key and host inventories are available in **Recently Rotated Keys**. You can access this functionality only if you have the ACF permissions enabled for your role.

1. Go to  (Menu) icon > **SSH+ > Inventory > Key Inventory**.
2. Select **Recently Rotated Keys**.

The **Recently Rotated Keys** page is displayed.

3. You can perform any of the following actions:
 - **View rotated keys:** The rotated user or host keys from endpoints are made available in the **User Keys** or **Host Keys**. The inventory lists key details including key name, associated users, client and host endpoints, fingerprint, who rotated the key, and the date when rotated.
 - **Rollback rotated keys:** To rollback keys that were rotated by mistake, simply select them and select **Rollback** from the **Actions** menu. A message appears, *Are you sure you want to rollback to the previous key? This action will revert the selected key(s) to their former state. Please confirm to proceed.* Click **Yes** to proceed.

**Note:**

An info icon appears beside the rotated key. Clicking the info icon takes you to the user/host key inventory where you can see the active key in the device. On performing rollback, that key is pushed and the active key is removed.

Dashboard

Reports

You can use this dashboard to look at the risks associated with SSH and their severity. This dashboard provides detailed information identifying critical and high device risks for administrators to mitigate and secure before management of a device is compromised.

**Note:**

The reports visible to you is based on your assigned ACL permissions.

Risk Report

The Risk Report provides information about risks and overall visibility of cumulative results of successful discoveries of hosts performed within the SSH+ product.

Click the **Count** hyperlink to fetch more details such as key name, group, length, age, encryption and so on based on the key type.

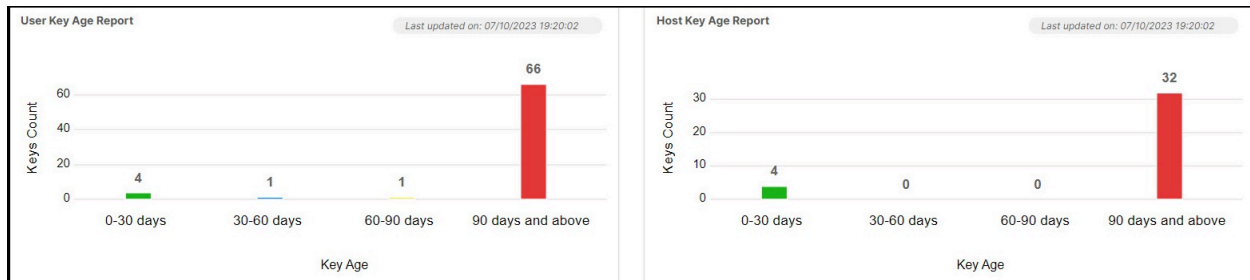
**Note:**

The Count field displays key count only from key compliance groups that the user has permission to access.

You can identify and remediate weak, shared, suspicious, and orphan host/user keys from the dashboard to keep the infrastructure secure. To perform remediation actions, see [Remediation Actions](#). For description of keys, see [Glossary](#).

Key Age Reports

Displays the following reports in a bar chart:



- **User Key Age Report:** A bar chart displaying the groupwise number of keys (Y-axis) based on the following user key age (X-axis):
 - 0 to 30 days
 - 30 to 60 days
 - 60 to 90 days
 - 90 days and above
- **Host Key Age Report:** A bar chart displaying the groupwise number of keys (Y-axis) based on the following host key age (X-axis):

- 0 to 30 days
- 30 to 60 days
- 60 to 90 days
- 90 days and above

Click the graph on the widget to fetch details of the client endpoints, host endpoints, associated users, key name, and the group.

You can also rotate and delete keys from hosts with multiple keys through the user and host key age report thus mitigating service disruptions. On selecting keys for rotation or deletion from an endpoint, a confirmation message appears. On confirming, the operation is triggered via workflow. To check the status and reports, go to **Automation > Service Request** and select your request from **All requests**.

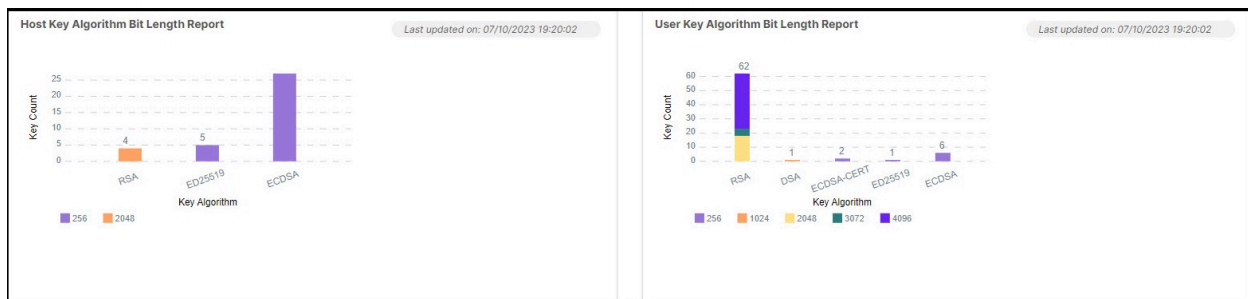


Note:

The age data is only fetched for private and public keys that are available in the host as a file. Age data is not fetched for keys that are part of grouped data like `authorized_key` files or `known_host` files.

Key Algorithm Bit Length Reports

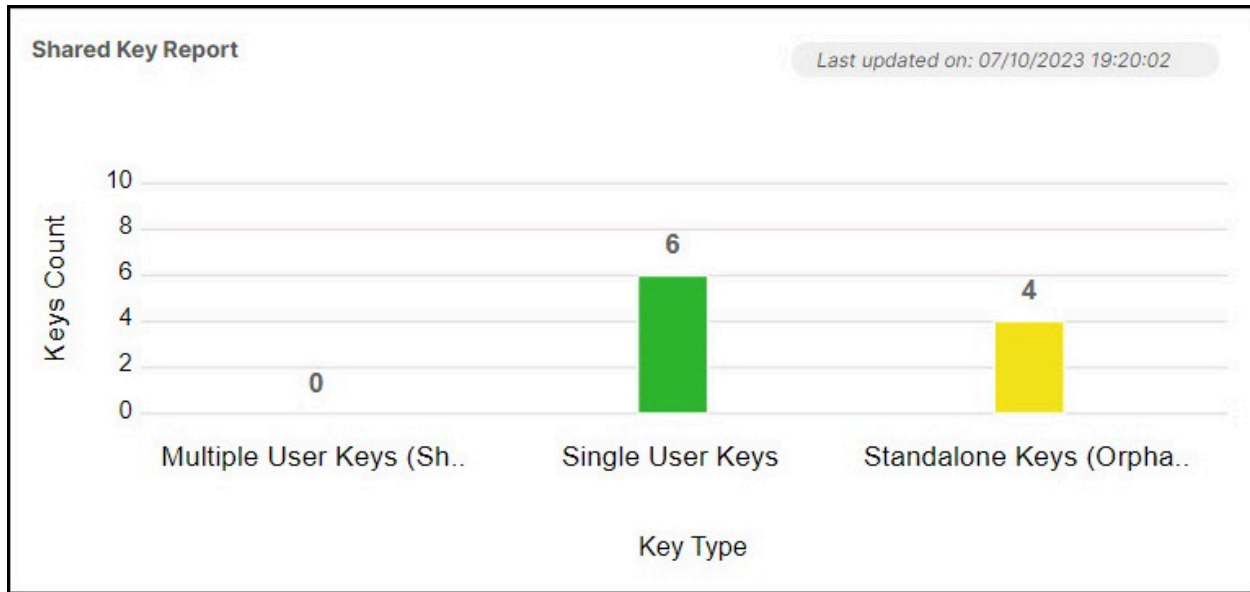
Displays a stacked bar chart to filter keys based on key bit length and key algorithm:



- **User Key Algorithm Bit Length Report:** A stacked bar chart displaying the groupwise number of key size (Y-axis) based on the user key size and algorithm (X-axis). The user key sizes are 256, 384, 521, 1024, 2048, 3072, 4096 while the key algorithms are RSA, DSA, RSA1, ED25519, ECDSA.
- **Host Key Algorithm Bit Length Report:** A stacked bar chart displaying the groupwise number of key size (Y-axis) based on the host key size and algorithm (X-axis). The host key sizes are 256 and 2048 while the key algorithms are RSA, ED25519, ECDSA.

Shared Key Report

Displays the key count (Y-axis) based on the following key type (X-axis) as a bar chart.

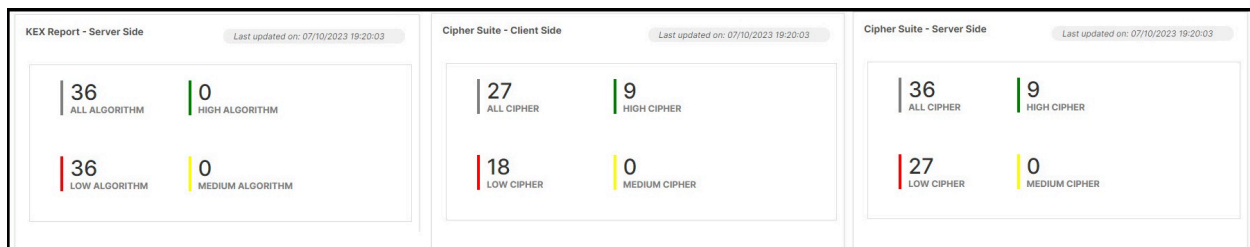


- **Multiple User Keys (Shared):** The number keys that are associated with multiple users and in the key group.
- **Single User Keys:** The number keys that are associated with a single user and in the key group.
- **Standalone Keys (Orphan):** The number of keys on standalone machines associated with the key group.

Click the graph on the widget to fetch details of the client endpoints, host endpoints, associated users, key name, and the host group.

Server-Side Reports

Displays the following reports:

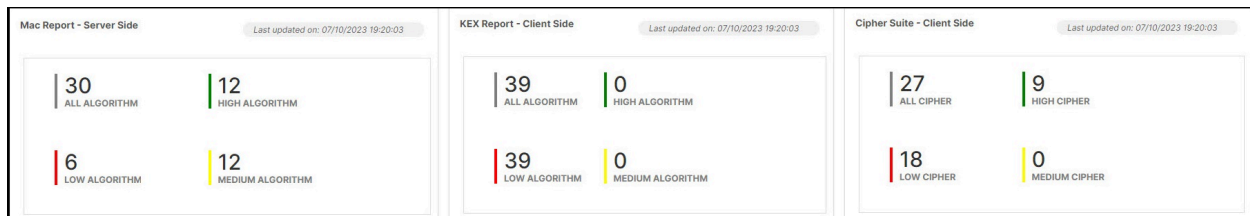


- **Cipher suite report:** This report provides the following numerical representation of All Cipher, High Cipher, Low Cipher, and Medium Cipher for the hosts added in the SSH host inventory.
- **KEX report:** This report provides the following numerical representation of All Algorithm, High Algorithm, Low Algorithm, and Medium Algorithm for the hosts added in the SSH host inventory.
- **Mac report:** This report provides the following numerical representation of All Algorithm, High Algorithm, Low Algorithm, and Medium Algorithm for the hosts added in the SSH host inventory.

Hover the mouse over the numbers and click it to fetch details of the host name, IP address/FQDN, algorithm, OS type, version, and group.

Client-Side Reports

Displays the following reports:



- **Cipher suite report:** This report provides the following numerical representation of All Cipher, High Cipher, Low Cipher, and Medium Cipher for the hosts added in the SSH host inventory.
- **KEX report:** This report provides the following numerical representation of All Algorithm, High Algorithm, Low Algorithm, and Medium Algorithm for the hosts added in the SSH host inventory.
- **Mac report:** This report provides the following numerical representation of All Algorithm, High Algorithm, Low Algorithm, and Medium Algorithm for the hosts added in the SSH host inventory.

Hover the mouse over the numbers and click it to fetch details of the host name, IP address/FQDN, algorithm, OS type, version, and host group.

Host Certificate Expiry Report

A bar chart displaying the number of host certificates (Y-axis) based on their expiration (X-axis):



- 0 to 10 days
- 10 to 30 days
- 30 to 60 days
- 60 days and above

Actions


- Click **Export** to export the details to .csv or .xls format.
- Click **View in Inventory** to view the inventory listing of the categories on that page.

Remediation Actions

Before you begin:

- You can access this functionality only if you have the ACF permissions enabled for your role.
- To perform any of the remediation actions, you must have RW permissions assigned to you.

You can identify and remediate weak, shared, suspicious, and orphan host/user keys from the dashboard to keep the infrastructure secure.

1. Go to  (**Menu**) icon > **SSH+ > Dashboard**.

The **Dashboard** page is displayed.

2. On the **Risk Report**, click **Remediation Option** against the host/user key category.

A popup window of the selected host/user key is displayed along with the details of the key name, host endpoints, associated users, and file path.

3. Click the checkbox against the key(s) you want to remediate.
4. You can perform one of the following actions:
 - a. **Rotate:** On selecting keys for rotation, a confirmation message appears. On confirming, the rotate operation is triggered via workflow. To check the status and reports, go to **Automation > Service Request** and select your request from **All requests**.


The selected keys are regenerated and pushed to the host endpoints.

 **Important:**

Best practices before rotating host keys:



Note:


The following points are applicable when the **Enable Global Known hosts** option is enabled under  (**Menu**) icon > **SSH+ > Administration > Advanced Settings**. Enabling this option may have implications for your network.

- i. If the *global known host* file is not present, then AppViewX will create one in the root folder by including all public keys from users in the global known host file.
- ii. Prior to host key rotation, update the *global known host* file.
- iii. The old public key is deleted and the new key is replaced in the *global known host* file.

Best practices before rotating user keys:



Note:

The following points are applicable when the **Enable Global Authorized keys** option is enabled under  (**Menu**) icon > **SSH+ > Administration > Advanced Settings**. Enabling this option may have implications for your network.

- i. If the *global authorized key* file is not present, then AppViewX will create one in the root folder for each login user with privileged user permission.
- ii. Prior to user key rotation, update the *global authorized key* file.
- iii. The old public key is deleted and the new key is replaced in the *global authorized key* file.

- b. **Delete:** On selecting keys for deletion from endpoint, a confirmation message appears. On confirming, the delete operation is triggered via workflow. To check the status and reports, go to

Automation > Service Request and select your request from **All requests**. The selected keys are deleted from the host endpoints and the key inventory.

- c. **Acknowledge:** The selected keys are acknowledged and excluded from the risk report for the duration specified in the associated key policy. The keys, however, continue to be present in the key inventory.

Selecting any of the actions opens a confirmation window.

5. Click **Confirm** to proceed.

Creating Key Policy and Group

Overview

Before you begin: You can access this functionality only if you have the ACF permissions enabled for your role.

Policy is configured to define the attributes of a key to belong to the key group. A key is compliant only if it matches the attributes defined in the associated policy.

From the **Policies** page, you can add, modify, or delete a key policy. See [Creating Key Policy](#).

You can bring together keys under a key group. A key can be part of only one group associated with a policy (key policy for the key group). This group to policy association can be done from the key inventory or from the policy create/edit page. By default, the system has a default key group to accommodate all keys that are not manually associated with any other group. See [Adding Key Compliance Group](#).

Key Policy

The key policy plays an important part while generating an SSH key.

You cannot delete the default key policy, but you have the option to change the settings. The default policy is associated with all the key groups.



Note:

You can associate a particular policy with a single key group or multiple key groups. The key configuration is based on the associated policy. The discovered keys are checked for compliance against the policy associated with them. The system marks the key as non-compliant in the inventory if the compliance check fails.

Creating Key Policy

To create a key policy:


1. Go to  (Menu) icon > **SSH+** > **Policies** > **Key Policy**.

The **Key Policy** page is displayed.

2. On the command bar, click **+Create policy**.
3. Enter the following details:

Field description for Key Policy section

Field	Description
Policy details	
*Policy Name	Enter a unique name for the policy.
Description	Enter details of the policy stating the purpose.
Compliance Configuration	
*Key Algorithm	Select a value from the dropdown list. You can select more than one value.
*Key Size	Select a value from the dropdown list. You can select more than one value.
Rotation Configuration	
*Key Rotation Period	Select a value from the dropdown list. For example, if you select 180 days from the dropdown list, then the key will be rotated after 180 days.
*Key Algorithm	Select a key algorithm that specifies which host key types are allowed to be used for the SSH connection.
*Key Size	Select the size of the key used in the key algorithm.
Host Certificate Auto Rotate Settings	
*Auto Rotate Host Certificates before	Select a value from the dropdown list to initiate certificate rotation before its expiration. By default, this value is 10 days.

Field	Description
 Note: Fields indicated with red asterisk (*) symbol are mandatory.	

4. Click **Create**.

A key policy is created and added to the key inventory.

What to do next:

- Modify or delete a key policy by selecting the checkbox against the policy name and selecting **Modify** or **Delete** from the **Actions** menu.
- Associate the policy with a key compliance group. See [Adding Key Compliance Group](#).

Key Compliance Group

By default, all the LDAP users are mapped to the Default Requestor group. A user can be associated with a single user group.

Adding Key Compliance Group

To add a key compliance group:

1. Go to  (**Menu**) icon > **SSH+** > **Groups** > **Key Compliance Group**.

The **Key Compliance Group** page is displayed.


2. On the command bar, click **+Add New Groups**.

The **Key Compliance Group > Create** page is displayed.

3. In the **General Information** section, enter the following:

Field description for General Information section

Field	Description
*Group Name	Enter a unique name. This helps you identify it easily.
Description	Enter details regarding the group stating the purpose.
*Requestor	Select the requestor(s) to be associated with the group.

Field	Description
*Requestor Policy	Select the required policy to be associated with the requestor group.
<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: Fields indicated with red asterisk (*) symbol are mandatory. </div>	

4. Click **Create**.

A key compliance group is created and displayed in the inventory.

What to do next:

- Users with RW permission can modify or delete a key compliance group by selecting the checkbox against the group name and selecting **Modify** or **Delete** from the **Actions** menu.

Viewing Key Compliance Group


To view a key compliance group:

1. Go to  (**Menu**) icon > **SSH+** > **Groups** > **Key Compliance Group**. The **Key Compliance Group** page is displayed.

Field description for General Information section


Field	Description
Group Name	Displays the name of the key compliance group.
Description	Displays the details of the key compliance group.
Count	<p>The Count field displays key count of keys only from the Key Compliance Groups that the user has permission to access.</p> <p>Click the hyperlink to open a pop-up window displaying all the keys associated with the key compliance group.</p>

Field	Description
Policy Associated	Displays the policy associated with the key compliance group.

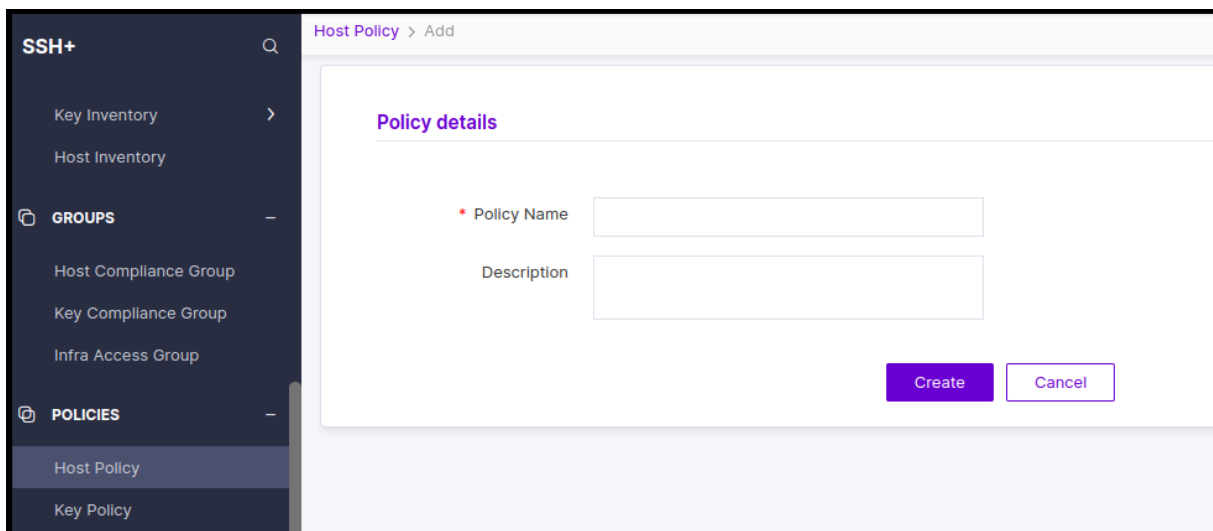
 **Note:**
Fields indicated with red asterisk (*) symbol are mandatory.

Creating Host Policy and Group

Creating Host Policy

1. Go to  (Menu) icon > **SSH+** > **Policies** > **Host Policy**.
2. Click **Create Policy**.

The **Host Policy** page is displayed.



3. Enter a unique policy name.
4. [Optional] Add description for the policy.

Click **Create**. The host policy is created.



Note:

Complete Host policy features will be available in a future release.

Glossary

Term definition


Term	Definition
SSH	Secure Socket Shell (SSH), also known as simply Secure Shell, is a cryptographic protocol used to enable secure access to remote servers and devices over the internet using SSH keys, certificates, or passwords.
SSH key	SSH keys are used to encrypt communication with a remote system. SSH keys usually come in pairs comprising a public and a private key and are used to grant access to authorized personnel to critical systems such as cloud, on-premise servers, and network devices.
Host key	A host key is a key that is used to identify the server. It is generated by the server and shared with the client during the initial connection setup. The client uses this key to verify the identity of the server before establishing a connection.
User key	A user key is a public key that is associated with a particular user account on the host. It is used to authenticate the user and establish a secure connection with the server.
Public key	A public key is used to encrypt data and verify digital signatures. It can be freely distributed, and anyone can use it to encrypt data or verify digital signatures. It is also used to establish a secure connection between the client and the server.
Private key	A private key is a secret key that is used to decrypt data and create digital signatures. It must be kept secret and never shared with anyone. The private key is used to authenticate the user and establish a secure connection with the server.
Suspicious key	A key without a known client association.
Shared key	A key used by more than one user.
Orphan key	A key that is found on a non-standard client file-folder path and does not have a known server.
SSH key rotation	The process of replacing the old key with a new one that adheres to the SSH key policy.
Weak key	A key that is generated using a weaker algorithm and size.

Chapter 3: SSH+ API Guide

This guide provides information on the APIs to use for various SSH actions.

API Reference

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:

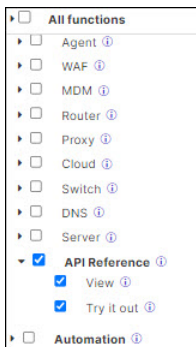
1. Go to **Platform** module **IDENTITY > Role**.

You will be redirected to **Role** page.

2. Click on the role name to enable the ACF permission.

You will be redirected to the **Modify :: [RoleName]** page, with the **Information** tab open by default.

3. Switch to the **Authorized Functions** tab and expand the **Platform** by clicking  (**Expand**) icon.
4. To assign **API Reference**, select the checkbox beside **API Reference**.



5. 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 SSH+ 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.

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>?<gwsources>
```

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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsources:** 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?gwsouce=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
}
```

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.

HTTP Code	Response Message
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.

Authentication using a User Account

For accessing APIs using a User account:

- **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.

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.

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.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **execute-hook**
- **gwsources:** Source or origin of a gateway, for example: **external**.

Input Parameters

	Description
username	(Mandatory) Use login name of the user.
<i>Header</i>	Type: String Example: "admin"
password	(Mandatory) Password for the username.
<i>Header</i>	Type: String Example: "AppViewX@123"
otp	(Mandatory only if MFA is enabled) If MFA is enabled, enter the OTP received on your registered email ID in the header.
<i>Header</i>	

Input Parameters (continued)

	Description
	<p>Multifactor authentication (MFA) is a security mechanism that requires users to provide two or more verification factors to gain access to a resource</p> <p>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.</p> <p>Type: String</p> <p>Example: "OTP : 609700"</p>
Content-Type <i>Header</i>	<p>(Mandatory) The parameter should be set to <code>application/json</code> to specify the nature of the data in the payload.</p> <p>Type: String</p> <p>Example: "application/json"</p>
gwsources <i>Query</i>	<p>(Mandatory) Source from which the request is triggered. The values can be:</p> <ul style="list-style-type: none"> • web • external <p>Type: String</p>

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: <ul style="list-style-type: none"> • SUCCESS • FAILURE
appStatusCode	An application-specific status code, if applicable.
statusDescription	Description of the status, if available.
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

Login to the application with a username and password.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/login?gwsorce=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,
    "termsAccepted": true,
    "passwordExpiryMsg": ""
  },
  "message": "Login successful.",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

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.

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:	/role
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/role?&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.

- **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
sessionId <i>Header</i>	(Mandatory) Use session ID retrieved from login API, if username and password are not provided. Type: <i>String</i> Example: "sessionId": "ce7f1a14-2bf9-4e4a-89a8-bc780a255813"
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. The values can be: <ul style="list-style-type: none"> • web • external Type: <i>String</i>
Payload <i>String</i>	(Mandatory) Input data for request body in application/json format. For payload details, see Payload section.

Payload

Name	Description
name <i>String</i>	(Mandatory) Name of the role to be added. Example: "role_1"
description <i>String</i>	(Optional) Description of the role to be added. Example: "Adding a new role"

Response Structure

- **Status Code:** 201 Created
- **Message:** Role added successfully
- **Headers:**
 - **Content-Type:** application/json

Response Parameters

Name	Description
response	Contains the response attributes for role added successfully.
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
201 Created	null	Role added successfully.
409 Conflict	ACCT_RO_002	Role name already exists
400 Bad Request	VALIDATION_ERROR_0004	'name' should have at least '2' characters, Mandatory Field 'name' is missing or empty.
400 Bad Request	ACCT_RO_015	Role name invalid.

Sample Request/Response

Use Case

Using the session ID acquired from the login API to execute subsequent API calls, specifically for adding a role API.

Sample Request

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/role?gwsouce=external
```

Request Payload

```
{
  "payload": {
    "name": "role_01",
    "description": "Adding a new role"
  }
}
```

Sample Response

```
{
  "response": "Role added successfully",
  "message": "Role added successfully",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Authentication using a Service Account

For accessing APIs using a Service account:

- **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**.

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
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**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

Input Parameters

	Description
Authorization <i>Header</i>	(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: <i>String</i> Example: "admin"
Content-Type <i>Header</i>	(Mandatory) The parameter should be set to <code>application/json</code> to specify the nature of the data in the payload. Type: <i>String</i> Example: "application/json"
grant_type <i>Payload</i>	(Mandatory) Payload Type should be "Form". The value of the param should be "Client_Credentials". Type: <i>Text</i>

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.


```

"appStatusCode": null,
"tags": {},
"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.

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:	/resource
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/resource?gwsouce=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.

- **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	(Mandatory) Use token retrieved from login API.
<i>Header</i>	<p>Type: <i>String</i></p> <p>Example: eyJ0eXAI0iJKV1QiLCJhbGciOiJIUz11NiJ9.eyJzdWliOiJwbGF0Zm9y bSIsImF1ZCI6ImF2eCIsImNsaWVudElkljoiOTcwNzRINDEtOGFmOS00NTZkLTlhNjQtZjB jNGJiOTA4MDQ4IiwiaXNzIjoieXZ4IiwiaXhwIjozNjUwMzY5MzY3LCJncmFudCB0eXBllj oiY2xpZW50X2NyZWRIbnRpYWxzIn0.HZnkuUEjXleqJWqpqi NWFHqIDI7GYf4cWx 6VwbjGD_0</p>
gwsource	(Mandatory) Source from which the request is triggered. The values can be:
<i>Query</i>	<ul style="list-style-type: none"> • web • external <p>Type: <i>String</i></p>
Payload	(Mandatory) Input data for request body in application/json format. For payload details, see Payload section.
<i>String</i>	

Payload

Name	Description
name	(Mandatory) Name of the resource to create. Name cannot be duplicated.
<i>String</i>	Example: "resource_1"
description	(Optional) Description of the resource.
<i>String</i>	Example: "This is a sample resource."

Response Structure

- **Status Code:** 201 Created
- **Message:** Resource added successfully
- **Headers:**
 - **Content-Type:** application/json

Response Parameters

Name	Description
response	Contains the response attributes for resource added successfully.
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
201 Created	null	Resource added successfully
409 Conflict	RBAC_RE_005	Resource with the given name already exists
400 Bad Request	VALIDATION_ERROR_0004	'name' should have at least '2' characters, Mandatory Field 'name' is missing or empty
400 Bad Request	VALIDATION_ERROR_0004	Invalid "name".

HTTP Code	appStatusCode	Response Message
401 Unauthorized	AVX_GW_012	Unauthorized access, reason - Invalid Token
407 Proxy Authentication Required	AVX_GW_011	Session validation failed, reason - Session information is missing.

Sample Request/Response

Use Case

Add a resource using API with Access Token.

Sample Request

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/resource?gwsource=external
```

Request Payload

```
{
  "payload": {
    "name": "resource_1",
    "description": "This is a sample resource."
  }
}
```

Sample Response

```
{
  "response": "Resource added successfully",
  "message": "Resource added successfully",
  "appStatusCode": null,
  "tags": null,
  "headers": null
}
```

Add New Host

The API will add a new host in host inventory.

Before you begin

Before attempting to add a new host, the user has to ensure the following:

- The host does not exist already.

Request Structure

Endpoint:	/ssh/host/create
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/host/create?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: The value of the param should be 'application/json'.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
data	(Mandatory) Contains request parameters to create and save the host. Type: Data

Data

Name	Description
groupName	(Mandatory) Contains request parameters to create and save the host. Type: String
categoryType	(Mandatory) Category type of the host Type: String Possible Values: Server
vendorType	(Mandatory) Type of the vendor Type: String
ipAddress	(Mandatory) IP address of the host

Data (continued)

Name	Description
	Type: String
fqdn	(Mandatory) Fully qualified domain name of the host
	Type: String
dataCenter	(Mandatory) Datacenter of the host
	Type: String
isClient	(Mandatory) Is the host client machine or not
	Type: Boolean
accessType	(Mandatory) Access type of the host
	Type: String
	Possible values: Key, Certificate
port	(Mandatory) Port number to connect to the host from
	Type: Number
deviceName	(Mandatory) Device name of the host
	Type: String
hostName	(Mandatory) Host name
	Type: String
loginType	(Mandatory) Login type for the host
	Type: String
	Possible values: Password, Identity Key
fileContent	(Mandatory) Identity Key file content, applicable only if "Identity Key" loginType is selected

Data (continued)

Name	Description
	Content-Type: application/octet-stream
fileName	(Mandatory) Name of the Identity key file, applicable only if “Identity Key” loginType is selected Type: String
userName	(Mandatory) Username to login to the host Type: String
password	(Mandatory) Password to login to the host Type: String
isSudoUser	(Mandatory) Is sudo user or not Type: Boolean
credentialType	(Mandatory) Credential type for authentication to login to the host Type: String Possible values: Manual Entry, Credential List - AppViewX, Credential List - CyberArk, Credential List - Thycotic Secret
credentialName	(Mandatory) Credential name (null for “Manual Entry” credentialType) Type: String
accessGroups	(Mandatory) List of access groups where the host belongs to Type: List
userComplianceGroup	(Mandatory) User compliance group of the host Type: String
inventoryAction	(Mandatory) Inventory action for the host

Data (continued)

Name	Description
	Type: String Possible values: Manage, Monitor, Do Not Move
accessElevation	(Mandatory) Access elevation of the user Type: String Possible Values: sudo, dzdo
sshSyncKeyDetail	(Mandatory) SSH sync key detail of the host Type: SshSyncKeyDetail

SshSyncKeyDetails

Name	Description
sshScanType	(Mandatory) SSH scan type Type: String Possible values: Default, Full, Directory
sshScanDetails	(Mandatory) Details of SSH scan Type: List
discoverKeyType	(Mandatory) List of key types to be discovered Type: List
appInfraAccessGroup	(Mandatory) List of app infra access groups where the host belongs to Type: List

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response message Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

response	Host created and saved successfully. Type: String
-----------------	---

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Host created and saved successfully.
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To add new host using `add_new_host` API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/host/create?gwsouce=external
```

Sample Request 1

```
{
  "data":
  {
    "groupName": "Default_Host_Group",
    "categoryType": "client",
    "vendorType": "linux",
    "ipAddress": "1.1.1.1",
    "fqdn": "",
    "dataCenter": "absecon",
    "isClient": false,
    "accessType": "Certificate",
    "port": "22",
    "deviceName": "test-device",
    "hostName": "",
    "loginType": "password",
    "userName": "appviewx",
    "isSudoUser": true,
    "credentialType": "Manual Entry",
    "credentialName": null,
    "accessGroupDeviceType": "Host",
    "accessGroups": ["Default_Infra_Access_Group"],
    "userComplianceGroup": "Default_Key_Group",
    "clientServerAccessGroups": []
  }
}
```

```

"sshSyncKeyDetail":{"sshScanType":"default","sshScanDetails":[],"discoverKeyType":["User Keys","Host
Keys"],"appInfraAccessGroup":["Default_Infra_Access_Group"]},
  "inventoryAction":"manage","password":"dummy pwd","accessElevation":"sudo"}
}

```

Sample Request 2

```

{
  "fileContent": (binary),
  "fileName": "demo",
  "data":
  {"groupName":"Default_Host_Group",
  "categoryType":"client",
  "vendorType":"linux",
  "ipAddress":"1.1.1.1",
  "fqdn":"",
  "dataCenter":"absecon",
  "isClient":false,
  "accessType":"Certificate",
  "port":"22",
  "deviceName":"test-device",
  "hostName":"",
  "loginType":"file",
  "userName":"appviewx",
  "isSudoUser":true,
  "credentialType":"Manual Entry",
  "credentialName":null,
  "accessGroupDeviceType":"Host",
  "accessGroups":["Default_Infra_Access_Group"],
  "userComplianceGroup":"Default_Key_Group",
  "clientServerAccessGroups":[],
  "sshSyncKeyDetail":{"sshScanType":"default","sshScanDetails":[],"discoverKeyType":["User Keys","Host
Keys"],"appInfraAccessGroup":["Default_Infra_Access_Group"]},
  "inventoryAction":"manage","password":"dummy pwd","accessElevation":"sudo"}
}

```

Sample Response

```
{
  "response": "Host created and saved successfully.",
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Hosts](#)
- [SSH Create CA](#)

Search Hosts

The API will search hosts and its information from the host inventory.

Before you begin

Before attempting to search hosts from the host inventory, the user has to ensure the following:

- Hosts must be added in AppViewX.

Request Structure

Endpoint:	/ssh/search/hosts
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hosts?gwsouce=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.

Request Parameters (continued)

Name	Description
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to find host information Type: Input
filter	(Mandatory) Filter parameters to find host information Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find host information Type: String

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted Type: String
sortOrder	(Mandatory) Order to be sorted Possible values: asc, desc
start	(Mandatory) Start count of the hosts to be fetched Type: String
max	(Mandatory) Count of the hosts to be fetched Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the hosts Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of host information which matches the search criteria Type: List
iTotalDisplayRecords	Total number of host available for the search criteria

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Host information retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To search for hosts using **search_hosts** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hosts?gwsource=external
```

Sample Request

```
{
  "input": {
    "freeSearch": "pe-cert-apvx-node05"
  },
}
```

```

"filter": {
  "sortColumn": "none",
  "sortOrder": "desc",
  "start": "0",
  "max": "100"
}
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "sshHost": {
          "credentialType": "Manual Entry",
          "fqdn": "pe-cert-apvx-node05.lab.appviewx.net.",
          "ipAddress": "192.168.60.133",
          "userName": "appviewx",
          "loginType": "password",
          "groupName": "Default_Host_Group",
          "deviceName": "pe-cert-apvx-node05",
          "port": "22",
          "accessElevation": "sudo",
          "hostName": "",
          "fingerPrint": "",
          "status": "Unresolved",
          "displayName": "192.168.60.133",
          "vendorType": "linux",
          "category": "Host",
          "keyWords": [
            "Unresolved",
            "22",
            "NA",
            "192.168.60.133",
            "pe-cert-apvx-node05",
            "appviewx",
            "Default_Host_Group",

```

```

"Host",
"sam"
],
"groupIds": [
  "5767bcdc3465bfbf73e44726"
],
"createdTime": 1716221229762,
"adcAccess": false,
"isClient": false,
"isBastion": false,
"deviceType": [
  "NA"
],
"deviceStatusList": [
  {
    "title": "Network Status: Reachability Check",
    "message": "FQDN pe-cert-apvx-node05.lab.appviewx.net. resolution failed.",
    "status": "Failed",
    "timeStamp": 1716221234111
  }
],
"isSudoUser": true,
"isValidSudo": false,
"lastSyncTime": 1716221229526,
"sshKeysCount": 0,
"certificatesCount": 0,
"communicationMode": "SSH",
"accessGroups": [
  "sam"
],
"accessGroupDeviceType": "Host",
"clientServerAccessGroups": [],
"categoryType": "server",
"userComplianceGroup": "Default_Key_Group",
"inventoryAction": "manage",
"sshSyncKeyDetail": {
  "autoCreateHPGroup": false,

```

```

    "sshSyncEnabled": false,
    "appInfraAccessGroup": [
      "sam"
    ],
    "sshScanType": "default",
    "sshScanDetails": [],
    "discoverKeyType": [
      "User Keys",
      "Host Keys"
    ],
    "fromSSHSubsystem": false
  },
  "dataCenter": "absecon",
  "accessType": "Certificate",
  "discoverySeqId": 0,
  "discoveryBatchNo": 0,
  "allowedPrincipalCount": 0,
  "sshOrigin": true,
  "update": false,
  "new": false,
  "selected": false,
  "active": true,
  "_id": "664b752d6c117a54017bc2f3"
},
"groups": [
  {
    "Default_Host_Group": "RW"
  }
],
"displayName": "192.168.60.133",
"fingerPrint": "",
"permission": "RW",
"groupPermissions": {
  "sam": "RW"
},
"_id": "664b752d6c117a54017bc2f3"
}

```

```

],
  "iTotalDisplayRecords": 1
},
"message": "Host information retrieved successfully",
"appStatusCode": null,
"tags": {},
"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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Host Keys](#)
- [Search User Keys](#)
- [SSH Create CA](#)

Search Host Keys

The API will search host keys and its information from the host key inventory.

Before you begin

Before attempting to searching host keys from the host key inventory, the user has to ensure the following:

- Host keys must be present in AppViewX.

Request Structure

Endpoint:	/ssh/search/hostKeys
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?gwsources=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username Type: String

Request Parameters (continued)

Name	Description
<i>Header</i>	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to fetch host keys Type: Input
filter	(Mandatory) Filter parameters to fetch host keys Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find host key information

Input (continued)

Name	Description
	Type: String
keywordSearch	(Optional) Keyword and value to search and retrieve host key information Example: {"keyname":"RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted Type: String
sortOrder	(Mandatory) Order to be sorted Possible values: asc, desc
start	(Mandatory) Start count of the host keys to be fetched Type: String
max	(Mandatory) Count of the host keys to be fetched Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the host keys Type: response

Response Parameters (continued)

Name	Description
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of host key information which matches the search criteria Type: List
iTotalDisplayRecords	Total number of host key available for the search criteria

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Host keys retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To search for host keys using **search_host_keys** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?gwsouce=external
```

Sample Request 1

```
{
  "input": {
    "freeSearch": "RotateKeys_admin_1716367184410-B0-001"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Request 2

```
{
  "input": {
    "keywordSearch":{"keyname":"RotateKeys_admin_1716367184410-B0-001"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
  }
}
```

```

"max": "100"
}
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "sshkey": {
          "type": "ECDSA",
          "name": "RotateKeys_admin_1716367184410-B0-001",
          "bitLength": "256",
          "passPhrase": null,
          "passPhraseCryptKey": null,
          "comment": "Created for host 192.168.60.130",
          "privateKey": null,
          "cryptKeyForPrivateKey": null,
          "publicKey": null,
          "fingerPrint": "+1cEqAhMLynje99v9hjeB3QEooh1sqEPQH1303d5P8o",
          "keyStatus": "Managed",
          "compliance": null,
          "groupName": "Default_Key_Group",
          "validity": null,
          "period": "lifetime",
          "hsmDeviceName": null,
          "uuid": "81da80bf-e20f-358e-8606-34b63d8d61c1",
          "keyWords": [
            "discoveredKeys",
            "Created for host 192.168.60.130",
            "256",
            "pe-cert-apvx-node02",
            "RotateKeys_admin_1716367184410-B0-001",
            "ECDSA",
            "Compliant"
          ],
          "active": true,
          "fileName": null,

```

```
"privateKeyFilePermission": null,  
"publicFilePermission": null,  
"createdBy": "discoveredKeys",  
"displayName": null,  
"awsDiscoveredKey": true,  
"userName": null,  
"associatedUsers": [],  
"clientFingerprint": null,  
"modifiedBy": null,  
"modifiedTime": 1716367224,  
"fingerPrints": null,  
"isModified": null,  
"colorCode": null,  
"oldPassPhrase": null,  
"oldPassPhraseCryptKey": null,  
"awsFingerPrint": null,  
"sourceIp": [],  
"destIp": [  
  "192.168.60.130"  
],  
"createdTime": 1716366952000,  
"isPushAutomatically": null,  
"isRotateAutomatically": null,  
"autoRotate": null,  
"workOrderDetail": null,  
"groupIds": [  
  "5767bcef3465bfbf73e44727"  
],  
"readWrite": false,  
"keyPath": null,  
"renewDate": 0,  
"expiryDate": 0,  
"isExpired": false,  
"hostUserName": null,  
"workOrderMandate": null,  
"initiatedTime": 1716366952000,  
"complianceDescription": null,
```

```

"agentUuid": null,
"keyType": null,
"filePaths": [
  "pe-cert-apvx-node02~/~/etc/ssh/appviewxssh/ssh_host_ECDSA_key",
  "pe-cert-apvx-node02~/~/etc/ssh/appviewxssh/ssh_host_ECDSA_key.pub"
],
"symLinks": [],
"privateKeyFileProps": {
  "readable": "true",
  "executable": "false",
  "writable": "true"
},
"publicKeyFileProps": {
  "readable": "true",
  "executable": "false",
  "writable": "true"
},
"clientMachineNames": [],
"serverMachineNames": [
  "pe-cert-apvx-node02"
],
"keyFoundInDiscoverySeqIdRanges": [
  {
    "from": 30,
    "to": 30
  }
],
"firstDiscovery": 1716367184461,
"previousDiscovery": 0,
"currentDiscoveredTime": 1716367184461,
"certificate": [
  {
    "fingerprint": "+1cEqAhMLynje99v9hjeB3QEooh1sqEPQH1303d5P8o",
    "certType": "Host",
    "rawFPString": "ECDSA-CERT SHA256:+1cEqAhMLynje99v9hjeB3QEooh1sqEPQH1303d5P8o",
    "signingCA": "ECDSA SHA256:k/k8+W/SlzdaK0ajyat/l3FVpnWpPpndaP0qTM7lf7M (using ecdsa-sha2-nistp256)",
    "keyId": "pe-cert-apvx-node02.lab.appviewx.net",
  }
]

```

```

"serialNumber": "3515119686734784",
"rawCertType": "ecdsa-sha2-nistp256-cert-v01@openssh.com host certificate",
"validFrom": 1716366944000,
"validTo": 1747902944000,
"validity": 364,
"validityUnit": "days",
"expiresIn": "364 days",
"principals": [
  "192.168.60.130",
  "pe-cert-apvx-node02.lab.appviewx.net"
],
"cryptKeyForCertContent": "opj82wtc1bylx4igskt7ra724",
"certStatus": "Active",
"filePaths": [
  {
    "hostName": "pe-cert-apvx-node02",
    "paths": [
      "pe-cert-apvx-node02-~/etc/ssh/appviewxssh/ssh_host_ECDSA_key-cert.pub"
    ]
  }
],
"onlyForWebTerminal": false
}
],
"sharedType": "single",
"excludeFromSharedKeyReportEndTime": 0,
"excludeFromWeakKeyReportEndTime": 0,
"excludeFromOrphanKeyReportEndTime": 0,
"excludeFromSuspiciousKeyReportEndTime": 0,
"discoveryIdWithNewState": {
  "30": true
},
"eligibleForRollback": false,
"backupData": null,
"sharedKey": false,
"weakKey": false,
"riskKey": false,

```

```

"discovered": true,
"accessRequest": false,
"new": true,
"keyDownload": false,
"privateKeyDeleted": false,
"publicKeyDeleted": false,
"keyFilePermission": [
  {
    "user": null,
    "userHomeDirectory": null,
    "userGroup": null,
    "filePath": null,
    "deviceName": null,
    "fileProperties": null
  }
],
"selected": false,
"upload": false,
"passphraseValidated": false,
"hasPrivateKey": true,
"_id": "664daf7868cf79570aab88b5"
},
"age": "0 day",
"clientMachineNames": null,
"serverMachineNames": null,
"groupPermission": [
  {
    "Default_Key_Group": "RW"
  }
],
"permission": "RW",
"compliance": "Compliant",
"createdTime": 0,
"displayName": "RotateKeys_admin_1716367184410-B0-001",
"hostComplianceGroup": null,
"joinedHostGroups": null,
"hostName": null,

```

```

"associatedUsers": null,
"colorCode": "newKeys",
"complianceDescription": "",
"hostInfos": null,
"keyComplianceGroup": null,
"accessGroup": null,
"selected": false,
"_id": null
}
],
"iTotalDisplayRecords": 1,
"serverTime": 1716377205029
},
"message": "User keys 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.

**Note:**

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Hosts](#)
- [Search User Keys](#)
- [SSH Create CA](#)

Search User Keys

The API will search user keys and its information from the user key inventory.

Before you begin

Before attempting to searching user keys from the user key inventory, the user has to ensure the following:

- User keys must be present in AppViewX.

Request Structure

Endpoint: `/ssh/search/userKeys`

**Note:**

The same API has been used for the following:



- [Backup User Keys Search](#)
- [User Keys Search from discovery](#)

Type: POST**Sample URL:** `https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/userKeys?gwsource=external`To understand the elements of the sample URL, click [here](#).**Headers:****Content-Type:** application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)

Request Parameters (continued)

Name	Description
<i>Query</i>	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to fetch user keys Type: Input
filter	(Mandatory) Filter parameters to fetch user keys Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find user key information Type: String
keywordSearch	(Optional) Keyword and value to search and retrieve user key information Example: {"keyname": "RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted Type: String

Filter (continued)

Name	Description
sortOrder	(Mandatory) Order to be sorted Possible values: asc, desc
start	(Mandatory) Start count of the user keys to be fetched Type: String
max	(Mandatory) Count of the user keys to be fetched Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the user keys Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of user key information which matches the search criteria

Response (continued)

Name	Description
Type: List	
iTotalDisplayRecords	Total number of user key available for the search criteria

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	User keys retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To search for user keys using **search_user_keys** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/userKeys?gwsouce=external
```

Sample Request 1

```
{
  "input": {
    "freeSearch": "FetchKey_admin_1716211635851-B0-022"
  },
  "filter": {
    "sortColumn": "none",
```

```

"sortOrder": "desc",
"start": "0",
"max": "100"
}
}

```

Sample Request 2

```

{
  "input": {
    "keywordSearch":{"keyname":"FetchKey_admin_1716211635851-B0-022"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "sshkey": {
          "type": "RSA",
          "name": "FetchKey_admin_1716211635851-B0-022",
          "bitLength": "1024",
          "passPhrase": null,
          "passPhraseCryptKey": null,
          "comment": "demo_may7",
          "privateKey": null,
          "cryptKeyForPrivateKey": null,
          "publicKey": null,
          "fingerPrint": "HXtVvirhDbslJottYGjpQsFgKL/Q1KtSxWx4A+dIMhl",
          "keyStatus": "Managed",
          "compliance": null,
          "groupName": "Default_Key_Group",

```

```
"validity": null,
"period": "lifetime",
"hsmDeviceName": null,
"uuid": "14383066-7481-3cbc-947f-39cc19419278",
"keyWords": [
  "FetchKey_admin_1716211635851-B0-022",
  "RSA",
  "discoveredKeys",
  "1024",
  "pe-cert-apvx-node02",
  "demo_may7",
  "Compliant"
],
"active": true,
"fileName": null,
"privateKeyFilePermission": null,
"publicFilePermission": null,
"createdBy": "discoveredKeys",
"displayName": null,
"awsDiscoveredKey": true,
"userName": null,
"associatedUsers": [
  "pe-cert-apvx-node02~~appviewx"
],
"clientFingerprint": null,
"modifiedBy": null,
"modifiedTime": 1716367224,
"fingerPrints": null,
"isModified": null,
"colorCode": null,
"oldPassPhrase": null,
"oldPassPhraseCryptKey": null,
"awsFingerPrint": null,
"sourceIp": [
  "192.168.60.130"
],
"destIp": [],
```

```

"createdTime": 1715086510000,
"isPushAutomatically": null,
"isRotateAutomatically": null,
"autoRotate": null,
"workOrderDetail": null,
"groupIds": [
  "5767bcef3465bfbf73e44727"
],
"readWrite": false,
"keyPath": null,
"renewDate": 0,
"expiryDate": 0,
"isExpired": false,
"hostUserName": null,
"workOrderMandate": null,
"initiatedTime": 1715086510000,
"complianceDescription": null,
"agentUuid": null,
"keyType": "REGULAR",
"filePaths": [
  "pe-cert-apvx-node02~/~/home/appviewx/.ssh/demomay7",
  "pe-cert-apvx-node02~/~/home/appviewx/.ssh/demomay7.pub"
],
"symLinks": [
  {
    "deviceName": "pe-cert-apvx-node02",
    "sourcePath": "/home/appviewx/.ssh/demomay7",
    "targetPath": "/home/appviewx/.ssh/appviewxssh/id_RSA_20240507_125500_639_0"
  },
  {
    "deviceName": "pe-cert-apvx-node02",
    "sourcePath": "/home/appviewx/.ssh/demomay7.pub",
    "targetPath": "/home/appviewx/.ssh/appviewxssh/id_RSA_20240507_125500_639_0.pub"
  }
],
"privateKeyFileProps": {
  "readable": "true",

```

```

    "executable": "false",
    "writable": "true"
  },
  "publicKeyFileProps": {
    "readable": "true",
    "executable": "false",
    "writable": "true"
  },
  "clientMachineNames": [
    "pe-cert-apvx-node02"
  ],
  "serverMachineNames": [],
  "keyFoundInDiscoverySeqIdRanges": [
    {
      "from": 21,
      "to": 21
    },
    {
      "from": 26,
      "to": 26
    },
    {
      "from": 28,
      "to": 30
    }
  ],
  "firstDiscovery": 1716211635872,
  "previousDiscovery": 1716366860849,
  "currentDiscoveredTime": 1716367184461,
  "certificate": [],
  "sharedType": "single",
  "excludeFromSharedKeyReportEndTime": 0,
  "excludeFromWeakKeyReportEndTime": 0,
  "excludeFromOrphanKeyReportEndTime": 0,
  "excludeFromSuspiciousKeyReportEndTime": 0,
  "discoveryIdWithNewState": {
    "21": true,

```

```

"26": true,
"28": true,
"29": true,
"30": true
},
"eligibleForRollback": false,
"backupData": null,
"sharedKey": false,
"weakKey": true,
"riskKey": true,
"discovered": true,
"accessRequest": false,
"new": true,
"keyDownload": false,
"privateKeyDeleted": false,
"publicKeyDeleted": false,
"keyFilePermission": [
{
"user": "appviewx",
"userHomeDirectory": "/home/appviewx",
"userGroup": "appviewx",
"filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7.pub",
"deviceName": "pe-cert-apvx-node02",
"fileProperties": [
{
"readable": true,
"writable": false,
"executable": false,
"permissionFor": "Group"
},
{
"readable": true,
"writable": false,
"executable": false,
"permissionFor": "Others"
}
]
}
]

```

```

    "readable": true,
    "writable": true,
    "executable": false,
    "permissionFor": "Owner"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    }
  ]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",

```

```

"userGroup": "appviewx",
"filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
"deviceName": "pe-cert-apvx-node02",
"fileProperties": [
  {
    "readable": true,
    "writable": true,
    "executable": false,
    "permissionFor": "Owner"
  },
  {
    "readable": false,
    "writable": false,
    "executable": false,
    "permissionFor": "Others"
  },
  {
    "readable": false,
    "writable": false,
    "executable": false,
    "permissionFor": "Group"
  }
],
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    },

```

```
{
  "readable": true,
  "writable": true,
  "executable": false,
  "permissionFor": "Owner"
},
{
  "readable": false,
  "writable": false,
  "executable": false,
  "permissionFor": "Group"
}
],
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7.pub",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    },
    {
      "readable": true,
      "writable": false,
      "executable": false,

```

```

    "permissionFor": "Others"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02~/home/appviewx/.ssh/demomay7.pub",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    },
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    }
  ]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02~/home/appviewx/.ssh/demomay7.pub",
  "deviceName": "pe-cert-apvx-node02",

```

```
"fileProperties": [  
  {  
    "readable": true,  
    "writable": false,  
    "executable": false,  
    "permissionFor": "Group"  
  },  
  {  
    "readable": true,  
    "writable": false,  
    "executable": false,  
    "permissionFor": "Others"  
  },  
  {  
    "readable": true,  
    "writable": true,  
    "executable": false,  
    "permissionFor": "Owner"  
  }  
]  
},  
{  
  "user": "appviewx",  
  "userHomeDirectory": "/home/appviewx",  
  "userGroup": "appviewx",  
  "filePath": "pe-cert-apvx-node02-~/home/appviewx/.ssh/demomay7",  
  "deviceName": "pe-cert-apvx-node02",  
  "fileProperties": [  
    {  
      "readable": true,  
      "writable": true,  
      "executable": false,  
      "permissionFor": "Owner"  
    },  
    {  
      "readable": false,  
      "writable": false,
```

```

    "executable": false,
    "permissionFor": "Others"
  },
  {
    "readable": false,
    "writable": false,
    "executable": false,
    "permissionFor": "Group"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    },
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    }
  ]
}
]

```

```

    },
    {
      "user": "appviewx",
      "userHomeDirectory": "/home/appviewx",
      "userGroup": "appviewx",
      "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7.pub",
      "deviceName": "pe-cert-apvx-node02",
      "fileProperties": [
        {
          "readable": true,
          "writable": false,
          "executable": false,
          "permissionFor": "Group"
        },
        {
          "readable": true,
          "writable": false,
          "executable": false,
          "permissionFor": "Others"
        },
        {
          "readable": true,
          "writable": true,
          "executable": false,
          "permissionFor": "Owner"
        }
      ]
    },
    ],
    "selected": false,
    "upload": false,
    "passphraseValidated": false,
    "hasPrivateKey": true,
    "_id": "664b4fd9e8eaab668f8a51d9"
  },
  "age": "14 days",
  "clientMachineNames": null,

```

```

"serverMachineNames": null,
"groupPermission": [
  {
    "Default_Key_Group": "RW"
  }
],
"permission": "RW",
"compliance": "Compliant",
"createdTime": 0,
"displayName": "FetchKey_admin_1716211635851-B0-022",
"hostComplianceGroup": null,
"joinedHostGroups": null,
"hostName": null,
"associatedUsers": null,
"colorCode": "lifetime",
"complianceDescription": "",
"hostInfos": null,
"keyComplianceGroup": null,
"accessGroup": null,
"selected": false,
"_id": null
}
],
"iTotalDisplayRecords": 1,
"serverTime": 1716376908710
},
"message": "User keys 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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Host Keys](#)
- [SSH Create CA](#)

Search Access Groups

The API will search access groups and its information from the access group inventory.

Before you begin

Before attempting to searching access groups from the access groups inventory, the user has to ensure the following:

- Access groups should be present in AppViewX.

Request Structure

Endpoint:	/ssh/app-infra-group/list
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/app-infra-group/list?gwsouce=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.

Request Parameters (continued)

Name	Description
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to fetch access groups Type: Input
filter	(Mandatory) Filter parameters to fetch access groups Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find access group information Type: String

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted Type: String
sortOrder	(Mandatory) Order to be sorted

Filter (continued)

Name	Description
	Possible values: asc, desc
start	(Mandatory) Start count of the access groups to be fetched Type: String
max	(Mandatory) Count of the access groups to be fetched Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the access groups Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of access group information which matches the search criteria Type: List

Response (continued)

Name	Description
iTotalDisplayRecords	Total number of access group available for the search criteria

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Access groups retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To search for access groups using **search_access_groups** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/app-infra-group/list?gwsouce=external
```

Sample Request

```
{
  "input": {
    "freeSearch": "Default_Infra_Access_Group"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
  }
}
```

```

"start": "0",
"max": "100"
}
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "name": "Default_Infra_Access_Group",
        "description": "Default Infra Access group",
        "creationType": "Auto",
        "associatedHostCount": 1,
        "rolePermissionMap": {
          "super access": "RW"
        },
        "keyWords": [
          "Auto",
          "Default Infra Access group",
          "Default_Infra_Access_Group"
        ],
        "userCAName": "Default-Infra-Access-Group.user.ca",
        "hostCAName": "appviewx.ssh.host.ca",
        "permission": "RW",
        "selected": false,
        "_id": "66027e3b6a91df51fa2c5d62"
      }
    ],
    "iTotalDisplayRecords": 1
  },
  "message": "Access groups 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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search CA](#)
- [SSH Create CA](#)
- [SSH Download CA](#)
- [SSH Get Hosts From Infra Access Group](#)

SSH Create CA

The API will create CA based on the given request.

Before you begin

Before attempting to creating CAs, the user has to ensure the following:

- Proper values given in the request.

Request Structure

Endpoint:	/ssh/ca/create
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/ca/create?gwsouce=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.

Request Parameters (continued)

Name	Description
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be 'application/json' .
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
name	(Mandatory) Name of the CA Type: String
validityValue	(Mandatory) Validity of the certificate Type: String
validityUnit	(Mandatory) Validity unit of the certificate Type: String
caType	(Mandatory) Type of the CA Type: String
comment	(Optional) Comments Type: String
bitLength	(Mandatory) Bit length for CA

Name	Description
	Type: String
algorithm	(Mandatory) Algorithm for CA
	Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the CAs
	Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response
	Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response
	Type: String
tags	More info in case of failure response

Response

Name	Description
id	ID of the CA
	Type: String
name	Name of the CA
	Type: String
caType	CA Type of the CA

Response (continued)

Name	Description
	Type: String
comment	Comment
	Type: String
renewEnabled	Renewal enabled for CA
	Type: String
publicKey	Public key of the CA
	Type: String
keyRevocationListStatus	KRL status of the CA
	Type: String
algorithm	Algorithm of the CA
	Type: String
bitLength	Bit length of the CA
	Type: String
validityValue	Validity of the certificate
	Type: String
validityUnit	Validity unit of the CA
	Type: String
expiryDate	Expiry date of the CA
	Type: String

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Success
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	ERR-SSH_NB-262	Invalid Key Algorithm or BitLength Possible remediation: Make sure to provide proper key type and bit length
	ERR-SSH_NB-212	CA name already exists Possible remediation: Please make sure to provide different CA name

Sample Request/Response

Use Case

To create CA using `create_CA` API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/ca/create?gwsource=external
```

Sample Request

```
{
  "name": "dev-ssh.user.ca",
  "validityValue": "12",
  "validityUnit": "months",
  "caType": "user",
  "comment": "This is a sample env user CA",
  "bitLength": "256",
  "algorithm": "ECDSA"
}
```

```
}

```

Sample Response

```
{
  "response": {
    "id": "664fc8ea0eea9f4a4438d1f0",
    "name": "dev-ssh.user.ca",
    "validityValue": 12,
    "validityUnit": "months",
    "caType": "User",
    "comment": "This is a sample env user CA",
    "renewEnabled": false,
    "publicKey": "dummy key",
    "cryptKeyForPublicKey": "dummy key",
    "keyRevocationListStatus": "Not created",
    "algorithm": "ECDSA",
    "bitLength": 256,
    "status": "Active",
    "expiryDate": 1748044799000
  },
  "message": "Success",
  "appStatusCode": "SSH-NB-200",
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [SSH Download CA](#)
- [SSH Create Certificate](#)

SSH Download CA

The API will download CA based on the given request.

Before you begin

Before attempting to downloading CAs, the user has to ensure the following:

- CA should be present in AppViewX.

Request Structure

Endpoint:	/ssh/ca/download
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/ca/download?gwsouce=external

To understand the elements of the sample URL, click [here](#).

Headers:**Content-Type:** application/json**Request Parameters**

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
caName <i>queryParam</i>	(Mandatory) Name of the CA Type: String
encode	(Mandatory) Encode response

Request Parameters (continued)

Name	Description
<code>queryParam</code>	Type: Boolean

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
<code>response</code>	Contains the response attributes for the CA content Type: <code>response</code>
<code>message</code>	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
<code>appStatusCode</code>	Application-specific status code for the response. Will be non-null for failure response Type: String
<code>tags</code>	More info in case of failure response

Response

Name	Description
<code>content</code>	Content of the CA Type: String

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	CA downloaded successfully

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	ERR-SSH_NB-247	Invalid query param found for <FieldName> Possible remediation: Please make sure to provide valid values
	ERR-SSH_NB450	Failed to download a SSH CA Certificate. Possible remediation: Please make sure to provide proper values in request

Sample Request/Response**Use Case**

To download CA using **download_CA** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/ca/download?gwsource=external
```

Sample URL

<https://pe-ssh-apvx-n4.lab.appviewx.net:31443/avxapi/ssh/ca/download?caName=dev-ssh.user.ca&encode=false&gwsource=internal>

Sample Response

CA content will be downloaded

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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

Search CA

The API will search CAs and its information.

Before you begin

Before attempting to searching CAs, the user has to ensure the following:

- CAs should be present in AppViewX.

Request Structure

Endpoint:	<code>/ssh/ca/list</code>
------------------	---------------------------

Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/ca/list?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request

Request Parameters (continued)

Name	Description
<i>Body</i>	Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to fetch CAs Type: Input
filter	(Mandatory) Filter parameters to fetch CAs Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find CA information Type: String

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted Type: String
sortOrder	(Mandatory) Order to be sorted Possible values: asc, desc
start	(Mandatory) Start count of the CAs to be fetched Type: String
max	(Mandatory) Count of the CAs to be fetched

Filter (continued)

Name	Description
	Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the CAs Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of CA information which matches the search criteria Type: List
iTotalDisplayRecords	Total number of CAs available for the search criteria

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	CAs retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response

Use Case

To search for CAs using **search_CAs** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/ca/list?gwsouce=external
```

Sample Request

```
{
  "input": {
    "freeSearch": "testVendor.user.ca"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Response

```
{
  "response": {
    "data": [
      {
        "id": "664b4f18e8eaab668f8a5199",
        "name": "testVendor.user.ca",
        "validityValue": 5,
        "validityUnit": "years",
        "caType": "User",
        "renewEnabled": false,
        "keyRevocationListStatus": "Not created",
        "algorithm": "ECDSA",
        "bitLength": 256,
        "status": "Active",
        "expiryDate": 1874015999000
      }
    ],
    "iTotalDisplayRecords": 1
  },
  "message": "CAs retrieved successfully",
  "appStatusCode": "SSH-NB-200",
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search CA](#)

SSH Create Certificate

The API will create certificates based on the given request.

Before you begin

Before attempting to creating certificates, the user has to ensure the following:

- CAs should be present in AppViewX.

Request Structure

Endpoint:	/ssh/cert/create
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/cert/create?gwsouce=external

To understand the elements of the sample URL, click [here](#).

Headers:**Content-Type:** application/json**Request Parameters**

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
publicKey	(Mandatory) Public key to create certificate Type: String
validityValue	(Mandatory) Validity of the certificate Type: String
validityUnit	(Mandatory) Validity unit of the certificate Type: String
certificateIdentity	(Mandatory) Identity of the certificate Type: String
caName	(Mandatory) CA name of the certificate Type: String
principals	(Mandatory) Principals of the certificate Type: List
certType	(Mandatory) Type of the certificate Type: String (User or Host)
validFrom	(Optional) Certificate valid from value Type: Long
validTo	(Optional) Certificate valid to value Type: Long

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the certificate Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
fingerPrint	Fingerprint of the certificate Type: String
certType	Type of the certificate Type: String
rawFPString	Raw fingerprint of the certificate Type: String
signingCA	Signing CA of the certificate Type: String
keyId	Key ID of the certificate Type: String
serialNumber	Serial number of the certificate

Response (continued)

Name	Description
	Type: String
rawCertType	Raw cert type of the certificate
	Type: String
validFrom	Valid from value of the certificate
	Type: String
validTo	Valid to value of the certificate
	Type: String
validity	Validity of the certificate
	Type: String
validityUnit	Validity unit of the certificate
	Type: String
principals	Principals of the certificate
	Type: String
extensions	Extensions of the certificate
	Type: String
certContent	Content of the certificate
	Type: String
caName	CA name of the certificate
	Type: String
certStatus	Status of the certificate
	Type: String

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Success
401 Unauthorized	AVX_GW_003	<p>Authentication failed, reason - Invalid Credentials</p> <p>Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.</p>
400 Bad Request	ERR-SSH_NB-268	<p>Valid certificate already exists for the given Key</p> <p>Possible remediation: Valid certificate is available for the given public key. Please make sure to use different public key</p>
	ERR-SSH_NB-268	<p>Valid certificate already exists for the given Key</p> <p>Possible remediation: Valid certificate is available for the given public key. Please make sure to use different public key</p>
	ERR-SSH_NB-267	<p>Selected CA is not in Active status</p> <p>Possible remediation: Please make sure to provide active CA name</p>
	ERR-SSH_NB-263	<p>Selected CA cannot sign requested Cert Type</p> <p>Possible remediation: Please provide proper CA name</p>
	ERR-SSH_NB-266	<p>Validity start cannot exceed validity end</p> <p>Possible remediation: Possible remediation: Make sure to provide start value less than end value</p>
	ERR-SSH_NB-269	<p>validityValue must be greater than zero</p>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: validityValue field should not be 0 or less than 0
	ERR-SSH_NB-264	Certificate validity cannot exceed CA validity Possible remediation: Make sure to give the certificate validity less than CA validity
	ERR-SSH_NB-273	Certificate validity end cannot be past date Possible remediation: Make sure to give the certificate validity in future
404 Not Found	ERR-SSH-NB-350	No CA found with given name or ID Possible remediation: Please make sure to provide the available CA name

Sample Request/Response**Use Case**

To create certificate using `create_certificate` API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/cert/create?gwsouce=external
```

Sample Request

```
{
  "publicKey": "dummy key",
  "validityValue": 1,
  "validityUnit": "months",
  "certificateIdentity": "admin",
  "caName": "testVendor.user.ca",
  "principals": ["admin"],
  "certType": "User"
}
```

Sample Response

```
{
  "response": {
    "fingerPrint": "AwQpO14LR7d1e9BubBGWcDmloifmrxC3M2xyceDO2s",
    "certType": "User",
    "rawFPString": "ECDSA-CERT SHA256:AwQpO14LR7d1e9BubBGWcDmloifmrxC3M2xyceDO2s",
    "signingCA": "ECDSA SHA256:A/Pfc4Se53vBzllstXbTWVmRy5u7n8mBiySryI+UZgl (using ecdsa-sha2-nistp256)",
    "keyId": "admin",
    "serialNumber": "3515398573506688",
    "rawCertType": "ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate",
    "validFrom": 1716503060000,
    "validTo": 1719181460000,
    "validity": 30,
    "validityUnit": "days",
    "principals": [
      "admin"
    ],
    "extensions": {
      "permitX11Forwarding": true,
      "permitAgentForwarding": true,
      "permitPortForwarding": true,
      "permitPty": true,
      "permitUserRc": true
    },
    "certContent": "dummy cert",
    "caName": "testPermG1.user.ca",
    "certStatus": "Active",
    "filePaths": [],
    "onlyForWebTerminal": false
  },
  "message": "Success",
  "appStatusCode": "SSH-NB-200",
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [SSH Download KRL](#)
- [SSH Get Hosts From Infra Access Group](#)

SSH Download KRL

The API will download KRL based on the given request.

Before you begin

Before attempting to downloading KRL, the user has to ensure the following:

- KRL should be present in AppViewX.

Request Structure

Endpoint:	/ssh/krl/download
Type:	GET
Sample URL:	<p>https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/krl/download?caName=dev-ssh.user.ca&gwsourc=external</p> <p>To understand the elements of the sample URL, click here.</p>
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if username and password are not provided.</p>
username	(Mandatory) AppViewX login username
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if sessionId is not provided.</p>
password	(Mandatory) AppViewX login password
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if sessionId is not provided.</p>
Content-Type	(Mandatory) Specifies the nature of the data in the payload

Request Parameters (continued)

Name	Description
<i>Header</i>	<p>Type: String</p> <p>Constraints: The value of the param should be 'application/json'.</p>
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	<p>Type: String</p>
caName	(Mandatory) Name of the CA
<i>queryParam</i>	<p>Type: String</p>

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	<p>Contains the response attributes for the KRL file</p> <p>Type: response</p>
message	<p>Success message of the action or failure description in case of error. Will be non-null for failure response</p> <p>Type: String</p>
appStatusCode	<p>Application-specific status code for the response. Will be non-null for failure response</p> <p>Type: String</p>
tags	More info in case of failure response

Response

Name	Description
content	Content of the KRL Type: String
name	Name of the CA Type: String

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	KRL file downloaded successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	ERR-SSH_NB-247	Invalid query param found for <FieldName> Possible remediation: Please make sure to provide valid values
	ERR-CA-515	KRL file does not exist or file size mismatch Possible remediation: Please make sure to provide proper values in request

Sample Request/Response**Use Case**

To download KRL using **download_KRL** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/krl/download?gwsouce=external
```

Sample URL

<https://pe-ssh-apvx-n4.lab.appviewx.net:31443/avxapi/ssh/krl/download?caName=dev-ssh.user.ca&gwsource=internal>

Sample Response

```
{
  "response": {
    "name": "dev-ssh.user.ca",
    "content": "content"
  },
  "message": "",
  "appStatusCode": "",
  "tags": {},
  "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.

**Note:**

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **/ssh/host/create**
- **gwsource**: Source or origin of a gateway, for example: **external**.

SSH Get Hosts From Infra Access Group

The API will be used to retrieve the host information from the access group.

Before you begin

Before attempting to get the host information from access group, the user has to ensure the following:

- Access group is present in the request.

Request Structure

Endpoint:	/ssh/app-infra-group/hosts
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/app-infra-group/hosts?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.

Request Parameters (continued)

Name	Description
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be 'application/json' .
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
groupName <i>queryParam</i>	(Mandatory) Access group name Type: String
isClient <i>queryParam</i>	(Mandatory) Fetch client hosts Type: Boolean
fetchTerminalPreference <i>queryParam</i>	(Mandatory) Fetch terminal preference Type: Boolean

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the host attributes Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
hosts	List of host information Type: List
applInfraAccessGroupName	Name of the access group Type: String

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Success
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
400 Bad Request	ERR-SSH_NB-247	Invalid Key Algorithm or BitLength Possible remediation: Make sure to provide proper key type and bit length

Sample Request/Response**Use Case**

To get hosts from infra access groups using `get_hosts_from_infra_access_groups` API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/app-infra-group/hosts?gwsource=external
```

Sample URL

https://pe-ssh-apvx-n4.lab.appviewx.net:31443/avxapi/ssh/app-infra-group/hosts?groupName=Default_Infra_Access_Group&isClient=true&fetchTerminalPreference=true&gwsource=external

Sample Response

```
{
  "response": {
    "hosts": [
      {
        "deviceName": "pe-cert-apvx-node01.lab.appviewx.net",
        "hostName": "pe-cert-apvx-node01.lab.appviewx.net",
        "ipAddress": "192.168.60.129",
        "status": "Managed",
        "isSudoUser": true,
        "isValidSudo": true,
        "isClient": true,
        "accessType": "Certificate",
        "vendor": "CentOS Linux",
        "vendorType": "linux",
        "communicationMode": "SSH",
        "pinned": false
      }
    ]
  }
}
```

```

    ],
    "appInfraAccessGroupName": "Default_Infra_Access_Group"
  },
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Access Groups](#)

Request access to infra access groups

The API will request access for an infra access group.

Before you begin

Before attempting to request access for an infra access group, the user has to ensure the following:

- At least one host should be added to the infra access group.

Request Structure

Endpoint:	/ssh/access/request
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/access/request?gwsouce=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	AppViewX login username
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.

Request Parameters (continued)

Name	Description
password	AppViewX login password
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	Specifies the nature of the data in the payload
<i>Header</i>	Type: String Constraints: The value of the param should be 'application/json' .
gwsource	Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
Payload	Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
requestAccessMode	Type of access mode required Type: String Possible values: AppViewX Terminal, Client, Unmanaged Client
accessGroup	Name of the infra access group for which access request has to be created. Type: String
accessDuration	Duration for which access for the infra access group is required Type: String Possible values: Hours, Days

Name	Description
comments	Comments Type: String
sshKeyGenerationType	Type of ssh key generation (applicable only for Unmanaged Client access request mode) Type: String Possible values: Generate via AppViewX, Upload Key
accessFor	Specify to whom the access is requested for Type: String Possible values: Self, User

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the hosts Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

status	Submitted
---------------	-----------

Response (continued)**Type:** String**accessRequestId**

Access Request Id of the access request

Type: String**Status Codes****Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Access requested successfully for an infra access group
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Sample Request**

```
{
  "accessGroup": "Default_Infra_Access_Group",
  "accessDuration": "7 hours",
  "requestAccessMode": "AppViewX Terminal",
  "comments": "cert-based-access-type",
  "accessFor": "User",
  "userName": "appviewx"
```

```
}

```

Sample Response

```
{
  "response":
  {
    "status": "Submitted",
    "accessRequestId": "66504bd9fe56006bcfedd006"
  },
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.

**Note:**

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **/ssh/host/create**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- [Search Access Groups](#)

Trigger Network Scan for Range of IP Addresses

The API will initiate network scans for specified IP address ranges.

Before you begin

N/A

Request Structure

Endpoint:	/ssh/discovery/create
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/discovery/create?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String
	Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
isUpdate	(Mandatory) Is this an update to already existing discovery or not
<i>queryParam</i>	Type: Boolean
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
data	Contains request parameters to trigger a new discovery.

Name	Description
------	-------------

Type: Data

Data

Name	Description
------	-------------

name (Mandatory) Name of the discovery to be triggered

Type: String

description (Optional) Description of the discovery to be triggered

Type: String

discoveryMode (Mandatory) Mode of the discovery

Type: String

Possible values: ipRange, subnet

scheduleType (Mandatory) Schedule type of the discovery

Type: String

Possible values: instant, scheduled

keyGroupName (Mandatory) Name of the Key compliance group to which the hosts in discovery should be added

Type: String

inventoryAction (Mandatory) Inventory action for the host

Type: String

Possible values: Manage, Monitor, Do Not Move

ipRangeBean (Mandatory) Contains the details of IP ranges to be discovered

Type: [IpRangeBean](#)

hostGroupNames (Mandatory) List of host compliance group names the hosts in discovery belong to

Data (continued)

Name	Description
	Type: List
isSudoUser	(Mandatory) Is sudo user or not Type: Boolean
accessElevation	(Mandatory) Access elevation of the user Type: String Possible values: sudo, dzdo
accessType	(Mandatory) Access type of the hosts to be discovered Type: String Possible values: Key, Certificate
sshSyncKeyDetail	(Mandatory) SSH sync key detail of the host Type: SshSyncKeyDetail
IpRangeBean	
Name	Description
startIp	(Mandatory) Start IP of the IP range to be discovered Type: String
endIp	(Mandatory) End IP of the IP range to be discovered Type: String
ipPerBatch	(Mandatory) Number of IP addresses to be discovered per batch Type: Number Possible values: 1, 2, 4, 8, 16, 32, 64, 128

IpRangeBean (continued)

Name	Description
isSelectPort	(Mandatory) List of app infra access groups where the host belongs to Type: String
port	(Mandatory) Port number to connect to the host from Type: Number
userName	(Mandatory) Username to login to the host Type: String
loginType	(Mandatory) Login type for the host Type: String Possible values: Password, Identity Key
fileContent	(Mandatory) Identity Key file content, applicable only if “ <i>Identity Key</i> ” loginType is selected Content-Type: application/octet-stream
fileName	(Mandatory) Name of the Identity key file, applicable only if “ <i>Identity Key</i> ” loginType is selected Type: String
password	(Mandatory) Password to login to the host Type: String
credentialType	(Mandatory) Credential type for authentication to login to the host Type: String Possible values: Manual Entry, Credential List - AppViewX, Credential List - CyberArk, Credential List - Thycotic Secret

IpRangeBean (continued)

Name	Description
credentialName	(Mandatory) Credential name (null for "Manual Entry" credentialType) Type: String
dataCenter	Name of the data center the hosts belong to Type: String

SshSyncKeyDetails

Name	Description
sshScanType	(Mandatory) SSH scan type Type: String Possible values: Default, Full, Directory
sshScanDetails	(Mandatory) Details of the SSH scan Type: List
discoverKeyType	(Mandatory) List of key types to be discovered Type: List
applnfraAccessGroup	(Mandatory) List of app infra access groups where the host belongs to Type: List

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response message

Response Parameters (continued)

Name	Description
	Type: String
message	Success message of the action or failure description in case of error. Will be non-null for failure response
	Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response
	Type: String
tags	More info in case of failure response

Response

response	Scheduler has been triggered successfully.
	Type: String

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Scheduler has been triggered successfully.
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To trigger network scan for range of IP addresses using **trigger_network_scan_for_range_of_IP_addresses** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/discovery/create?gwsouce=external
```

Sample Request 1

```
{
  "data":{
    "name":"test_2",
    "description":"",
    "discoveryMode":"ipRange",
    "scheduleType":"instant",
    "keyGroupName":"Default_Key_Group",
    "inventoryAction":"manage",
    "ipRangeBean":
    {
      "startIp":"1.1.1.1",
      "endIp":"1.1.1.4",
      "ipPerBatch":"1",
      "isSelectPort":"custom",
      "port":"22",
      "userName":"appviewx",
      "loginType":"password",
      "password":"dummy pwd",
      "credentialType":"Manual Entry",
      "credentialName":null,
      "dataCenter":"absecon",
      "hostGroupNames":["Default_Host_Group"],
```

```

"isSudoUser":true,
"accessElevation":"sudo",
"source":"IP_Range_Discovery",
"accessType":"Certificate",
"sshSyncKeyDetail":
{"sshScanType":"default",
"sshScanDetails":[]},
"discoverKeyType":["User Keys","Host Keys"],
"appInfraAccessGroup":["Default_Infra_Access_Group"]
}

```

Sample Request 2

```

{
  "data": {
    "name": "test",
    "description": "",
    "discoveryMode": "subnet",
    "scheduleType": "instant",
    "keyGroupName": "Default_Key_Group",
    "inventoryAction": "manage",
    "subnetBean": {
      "network": "10.12.11.0/24",
      "superNet": "25",
      "isSelectPort": "custom",
      "port": "22",
      "userName": "appviewx",
      "loginType": "password",
      "password": "dummyPwd",
      "credentialType": "Manual Entry",
      "credentialName": null,
      "dataCenter": "absecon"
    },
    "hostGroupNames": [
      "Default_Host_Group"
    ],
    "isSudoUser": true,
    "accessElevation": "sudo",

```

```

"source": "Subnet_Scan_Discovery",
"accessType": "Certificate",
"sshSyncKeyDetail": {
  "sshScanType": "default",
  "sshScanDetails": [],
  "discoverKeyType": [
    "User Keys",
    "Host Keys"
  ],
  "appInfraAccessGroup": [
    "Default_Infra_Access_Group"
  ]
}
}
}

```

Sample Response

```

{
  "response": "Scheduler has been triggered successfully.",
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsource:** Source or origin of a gateway, for example: **external**.

Search Hosts from Discovery

The API will search hosts and its information from the existing discovery.

Before you begin

Before attempting to searching hosts from the discovery, the user has to ensure the following:

- Discovery should be added in AppViewX.

Request Structure

Endpoint: `/ssh/search/hosts`



Note:

The same API has been used for the following with the additional query params:



• Host Search

Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hosts?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external)

Request Parameters (continued)

Name	Description
	Type: String
discoveryName	Discovery Name. (E.g. SampleDiscovery)
<i>Query</i>	If this value not given, hosts will be fetched from host inventory.
	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	
	Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to find host information from discovery
	Type: Input
filter	(Mandatory) Filter parameters to find host information from discovery
	Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find host information from discovery
	Type: String

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted.

Filter (continued)

Name	Description
	Type: String
sortOrder	(Mandatory) Order to be sorted. Possible values: asc, desc
start	(Mandatory) Start count of the host keys to be fetched from discovery. Type: String
max	(Mandatory) Count of the hosts to be fetched from discovery. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the host keys Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of host information which matches the search criteria Type: List
iTotalDisplayRecords	Total number of host available for the search criteria

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Host information retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response

Use Case

To search host "pe-cert-apvx-node05" in the discovery using **/search/hosts** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hosts?keysFrom=rotatedInventory&gwsouce=externalhttps://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?keysFrom=rotatedInventory&gwsouce=external
```

Sample Request 1

```
{
  "input": {
```

```

"freeSearch": "pe-cert-apvx-node05"
},
"filter": {
  "sortColumn": "none",
  "sortOrder": "desc",
  "start": "0",
  "max": "100"
}
}

```

Sample Request 2

```

{
  "input": {
    "keywordSearch":{"keyname":"RotateKeys_admin_1716367184410-B0-001"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "sshHost": {
          "credentialType": "Manual Entry",
          "fqdn": "pe-cert-apvx-node05.lab.appviewx.net.",
          "ipAddress": "192.168.60.133",
          "userName": "appviewx",
          "loginType": "password",
          "groupName": "Default_Host_Group",
          "deviceName": "pe-cert-apvx-node05",
          "port": "22",
          "accessElevation": "sudo",

```

```

"hostName": "",
"fingerPrint": "",
"status": "Unresolved",
"displayName": "192.168.60.133",
"vendorType": "linux",
"category": "Host",
"keyWords": [
  "Unresolved",
  "22",
  "NA",
  "192.168.60.133",
  "pe-cert-apvx-node05",
  "appviewx",
  "Default_Host_Group",
  "Host",
  "sam"
],
"groupIds": [
  "5767bcd3465bfbf73e44726"
],
"createdTime": 1716221229762,
"adcAccess": false,
"isClient": false,
"isBastion": false,
"deviceType": [
  "NA"
],
"deviceStatusList": [
  {
    "title": "Network Status: Reachability Check",
    "message": "FQDN pe-cert-apvx-node05.lab.appviewx.net. resolution failed.",
    "status": "Failed",
    "timeStamp": 1716221234111
  }
],
"isSudoUser": true,
"isValidSudo": false,

```

```

"lastSyncTime": 1716221229526,
"sshKeysCount": 0,
"certificatesCount": 0,
"communicationMode": "SSH",
"accessGroups": [
  "sam"
],
"accessGroupDeviceType": "Host",
"clientServerAccessGroups": [],
"categoryType": "server",
"userComplianceGroup": "Default_Key_Group",
"inventoryAction": "manage",
"sshSyncKeyDetail": {
  "autoCreateHPGroup": false,
  "sshSyncEnabled": false,
  "appInfraAccessGroup": [
    "sam"
  ],
  "sshScanType": "default",
  "sshScanDetails": [],
  "discoverKeyType": [
    "User Keys",
    "Host Keys"
  ],
  "fromSSHSubsystem": false
},
"dataCenter": "absecon",
"accessType": "Certificate",
"discoverySeqId": 0,
"discoveryBatchNo": 0,
"allowedPrincipalCount": 0,
"sshOrigin": true,
"update": false,
"new": false,
"selected": false,
"active": true,
"_id": "664b752d6c117a54017bc2f3"

```

```

    },
    "groups": [
      {
        "Default_Host_Group": "RW"
      }
    ],
    "displayName": "192.168.60.133",
    "fingerPrint": "",
    "permission": "RW",
    "groupPermissions": {
      "sam": "RW"
    },
    "_id": "664b752d6c117a54017bc2f3"
  }
],
"ITotalDisplayRecords": 1
},
"message": "Host information retrieved successfully",
"appStatusCode": null,
"tags": {},
"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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsource:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Hosts](#)
- [Search Host Keys](#)

Search User Keys From Discovery

The API will search user keys and its information from the backup keys.

Before you begin

Before attempting to searching user keys from the discovery, the user has to ensure the following:

- Discovery should be present in AppViewX.

Request Structure

Endpoint: `/ssh/search/userKeys`



Note:

The same API has been used for the following:



- [User Keys Search](#)
- [Backup User Keys Search](#)

Type: POST

Sample URL: `https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/userKeys?gwsource=external`

To understand the elements of the sample URL, click [here](#).

Headers:

Content-Type: application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)

Request Parameters (continued)

Name	Description
<i>Query</i>	Type: String
discoveryName	Discovery Name (E.g. SampleDiscovery).
<i>Query</i>	If this value not given, hosts will be fetched from user keys inventory. Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
input	(Optional) Input Parameters to fetch user keys from discovery. Type: Input
filter	(Mandatory) Filter parameters to fetch user keys from discovery. Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find user key information from discovery Type: String
keywordSearch	(Optional) Keyword and value to search and retrieve user key . Example: {"keyname": "RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted. Type: String
sortOrder	(Mandatory) Order to be sorted. Possible values: asc, desc
start	(Mandatory) Start count of the user keys to be fetched from discovery. Type: String
max	Count of the user keys to be fetched from backup inventory. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the user keys Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of user key information which matches the search criteria. Type: List
iTotalDisplayRecords	Total number of user keys available for the search criteria.

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	User keys retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response

Use Case

To search user key "FetchKey_admin_1716211635851-B0-022" in the discovery using **/search/userKeys** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/userKeys?gwsource=external
```

Sample Request 1

```
{
  "input": {
    "freeSearch": "FetchKey_admin_1716211635851-B0-022"
  }
}
```

```

},
"filter": {
  "sortColumn": "none",
  "sortOrder": "desc",
  "start": "0",
  "max": "100"
}
}

```

Sample Request 2

```

{
  "input": {
    "keywordSearch":{"keyname":"FetchKey_admin_1716211635851-B0-022"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "sshkey": {
          "type": "RSA",
          "name": "FetchKey_admin_1716211635851-B0-022",
          "bitLength": "1024",
          "passPhrase": null,
          "passPhraseCryptKey": null,
          "comment": "demo_may7",
          "privateKey": null,
          "cryptKeyForPrivateKey": null,
          "publicKey": null,
          "fingerprint": "HXtVvirhDbslJottYGjpQsFgKL/Q1KISxWx4A+dIMhl",

```

```
"keyStatus": "Managed",
"compliance": null,
"groupName": "Default_Key_Group",
"validity": null,
"period": "lifetime",
"hsmDeviceName": null,
"uuid": "14383066-7481-3cbc-947f-39cc19419278",
"keyWords": [
  "FetchKey_admin_1716211635851-B0-022",
  "RSA",
  "discoveredKeys",
  "1024",
  "pe-cert-apvx-node02",
  "demo_may7",
  "Compliant"
],
"active": true,
"fileName": null,
"privateKeyFilePermission": null,
"publicFilePermission": null,
"createdBy": "discoveredKeys",
"displayName": null,
"awsDiscoveredKey": true,
"userName": null,
"associatedUsers": [
  "pe-cert-apvx-node02~~appviewx"
],
"clientFingerprint": null,
"modifiedBy": null,
"modifiedTime": 1716367224,
"fingerPrints": null,
"isModified": null,
"colorCode": null,
"oldPassPhrase": null,
"oldPassPhraseCryptKey": null,
"awsFingerPrint": null,
"sourceIp": [
```

```

"192.168.60.130"
],
"destIp": [],
"createdTime": 1715086510000,
"isPushAutomatically": null,
"isRotateAutomatically": null,
"autoRotate": null,
"workOrderDetail": null,
"groupIds": [
  "5767bcef3465bfbf73e44727"
],
"readWrite": false,
"keyPath": null,
"renewDate": 0,
"expiryDate": 0,
"isExpired": false,
"hostUserName": null,
"workOrderMandate": null,
"initiatedTime": 1715086510000,
"complianceDescription": null,
"agentUuid": null,
"keyType": "REGULAR",
"filePaths": [
  "pe-cert-apvx-node02~~~/home/appviewx/.ssh/demomay7",
  "pe-cert-apvx-node02~~~/home/appviewx/.ssh/demomay7.pub"
],
"symLinks": [
  {
    "deviceName": "pe-cert-apvx-node02",
    "sourcePath": "/home/appviewx/.ssh/demomay7",
    "targetPath": "/home/appviewx/.ssh/appviewxssh/id_RSA_20240507_125500_639_0"
  },
  {
    "deviceName": "pe-cert-apvx-node02",
    "sourcePath": "/home/appviewx/.ssh/demomay7.pub",
    "targetPath": "/home/appviewx/.ssh/appviewxssh/id_RSA_20240507_125500_639_0.pub"
  }
]

```

```
],
"privateKeyFileProps": {
  "readable": "true",
  "executable": "false",
  "writable": "true"
},
"publicKeyFileProps": {
  "readable": "true",
  "executable": "false",
  "writable": "true"
},
"clientMachineNames": [
  "pe-cert-apvx-node02"
],
"serverMachineNames": [],
"keyFoundInDiscoverySeqIdRanges": [
  {
    "from": 21,
    "to": 21
  },
  {
    "from": 26,
    "to": 26
  },
  {
    "from": 28,
    "to": 30
  }
],
"firstDiscovery": 1716211635872,
"previousDiscovery": 1716366860849,
"currentDiscoveredTime": 1716367184461,
"certificate": [],
"sharedType": "single",
"excludeFromSharedKeyReportEndTime": 0,
"excludeFromWeakKeyReportEndTime": 0,
"excludeFromOrphanKeyReportEndTime": 0,
```

```
"excludeFromSuspiciousKeyReportEndTime": 0,
"discoveryIdWithNewState": {
  "21": true,
  "26": true,
  "28": true,
  "29": true,
  "30": true
},
"eligibleForRollback": false,
"backupData": null,
"sharedKey": false,
"weakKey": true,
"riskKey": true,
"discovered": true,
"accessRequest": false,
"new": true,
"keyDownload": false,
"privateKeyDeleted": false,
"publicKeyDeleted": false,
"keyFilePermission": [
  {
    "user": "appviewx",
    "userHomeDirectory": "/home/appviewx",
    "userGroup": "appviewx",
    "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7.pub",
    "deviceName": "pe-cert-apvx-node02",
    "fileProperties": [
      {
        "readable": true,
        "writable": false,
        "executable": false,
        "permissionFor": "Group"
      },
      {
        "readable": true,
        "writable": false,
        "executable": false,
```

```
    "permissionFor": "Others"
  },
  {
    "readable": true,
    "writable": true,
    "executable": false,
    "permissionFor": "Owner"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    }
  ]
},
```

```

{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    }
  ],
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": false,
      "writable": false,

```

```

    "executable": false,
    "permissionFor": "Others"
  },
  {
    "readable": true,
    "writable": true,
    "executable": false,
    "permissionFor": "Owner"
  },
  {
    "readable": false,
    "writable": false,
    "executable": false,
    "permissionFor": "Group"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7.pub",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    }
  ]
}

```

```

    "readable": true,
    "writable": false,
    "executable": false,
    "permissionFor": "Others"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02~/home/appviewx/.ssh/demomay7.pub",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    },
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    }
  ]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",

```

```
"userGroup": "appviewx",
"filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7.pub",
"deviceName": "pe-cert-apvx-node02",
"fileProperties": [
  {
    "readable": true,
    "writable": false,
    "executable": false,
    "permissionFor": "Group"
  },
  {
    "readable": true,
    "writable": false,
    "executable": false,
    "permissionFor": "Others"
  },
  {
    "readable": true,
    "writable": true,
    "executable": false,
    "permissionFor": "Owner"
  }
],
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    }
  ],
```

```
{
  "readable": false,
  "writable": false,
  "executable": false,
  "permissionFor": "Others"
},
{
  "readable": false,
  "writable": false,
  "executable": false,
  "permissionFor": "Group"
}
],
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02--~/home/appviewx/.ssh/demomay7",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": false,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    },
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    },
    {
      "readable": false,
      "writable": false,
      "executable": false,
    }
  ]
}
```

```

    "permissionFor": "Group"
  }
]
},
{
  "user": "appviewx",
  "userHomeDirectory": "/home/appviewx",
  "userGroup": "appviewx",
  "filePath": "pe-cert-apvx-node02~/home/appviewx/.ssh/demomay7.pub",
  "deviceName": "pe-cert-apvx-node02",
  "fileProperties": [
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Group"
    },
    {
      "readable": true,
      "writable": false,
      "executable": false,
      "permissionFor": "Others"
    },
    {
      "readable": true,
      "writable": true,
      "executable": false,
      "permissionFor": "Owner"
    }
  ]
}
],
"selected": false,
"upload": false,
"passphraseValidated": false,
"hasPrivateKey": true,
"_id": "664b4fd9e8eaab668f8a51d9"

```

```

    },
    "age": "14 days",
    "clientMachineNames": null,
    "serverMachineNames": null,
    "groupPermission": [
      {
        "Default_Key_Group": "RW"
      }
    ],
    "permission": "RW",
    "compliance": "Compliant",
    "createdTime": 0,
    "displayName": "FetchKey_admin_1716211635851-B0-022",
    "hostComplianceGroup": null,
    "joinedHostGroups": null,
    "hostName": null,
    "associatedUsers": null,
    "colorCode": "lifetime",
    "complianceDescription": "",
    "hostInfos": null,
    "keyComplianceGroup": null,
    "accessGroup": null,
    "selected": false,
    "_id": null
  }
],
"iTotalDisplayRecords": 1,
"serverTime": 1716376908710
},
"message": "User keys 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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsource:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search User Keys](#)

Search Host Keys From Discovery


The API will search host keys and its information from the discovery.

Before you begin

Before attempting to searching host keys from the host key inventory, the user has to ensure the following:

- Discovery should be present in AppViewX.

Request Structure

Endpoint:	/ssh/search/hostKeys
	<div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: The same API has been used for the following: <ul style="list-style-type: none"> • Host Keys Search • Backup Host Keys Search </div>
Type:	POST
Sample URL:	<p>https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?gwsouce=external</p> <p>To understand the elements of the sample URL, click here.</p>
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if username and password are not provided.</p>
username	(Mandatory) AppViewX login username
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if sessionId is not provided.</p>
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be 'application/json' .
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
discoveryName <i>Query</i>	Discovery Name. (E.g. SampleDiscovery) If this value not given, hosts will be fetched from host keys inventory. Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
input	(Optional) Input Parameters to fetch host keys from discovery. Type: Input
filter	(Mandatory) Filter parameters to fetch host keys from discovery. Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find host key information from discovery. Type: String
keywordSearch	(Optional) Keyword and value to search and retrieve host key. Example: {"keyname": "RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted. Type: String
sortOrder	(Mandatory) Order to be sorted. Possible values: asc, desc
start	(Mandatory) Start count of the host keys to be fetched from discovery. Type: String
max	(Mandatory) Count of the host keys to be fetched from discovery. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the host keys Type: response

Response Parameters (continued)

Name	Description
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of host key information which matches the search criteria. Type: List
iTotalDisplayRecords	Total number of host keys available for the search criteria.

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Host keys retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To search host key “RotateKeys_admin_1716367184410-B0-001” in the discovery using **/search/hostKeys** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?keysFrom=rotatedInventory&gwsource=externalhttps://
<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?keysFrom=rotatedInventory&gwsource=external
```

Sample Request 1

```
{
  "input": {
    "freeSearch": "RotateKeys_admin_1716367184410-B0-001"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Request 2

```
{
  "input": {
    "keywordSearch":{"keyname":"RotateKeys_admin_1716367184410-B0-001"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
  }
}
```

```

"start": "0",
"max": "100"
}
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "sshkey": {
          "type": "ECDSA",
          "name": "RotateKeys_admin_1716367184410-B0-001",
          "bitLength": "256",
          "passPhrase": null,
          "passPhraseCryptKey": null,
          "comment": "Created for host 192.168.60.130",
          "privateKey": null,
          "cryptKeyForPrivateKey": null,
          "publicKey": null,
          "fingerprint": "+1cEqAhMLynje99v9hjeB3QEooh1sqEPQH1303d5P8o",
          "keyStatus": "Managed",
          "compliance": null,
          "groupName": "Default_Key_Group",
          "validity": null,
          "period": "lifetime",
          "hsmDeviceName": null,
          "uuid": "81da80bf-e20f-358e-8606-34b63d8d61c1",
          "keyWords": [
            "discoveredKeys",
            "Created for host 192.168.60.130",
            "256",
            "pe-cert-apvx-node02",
            "RotateKeys_admin_1716367184410-B0-001",
            "ECDSA",
            "Compliant"
          ],
          "active": true,

```

```
"fileName": null,  
"privateKeyFilePermission": null,  
"publicFilePermission": null,  
"createdBy": "discoveredKeys",  
"displayName": null,  
"awsDiscoveredKey": true,  
"userName": null,  
"associatedUsers": [],  
"clientFingerprint": null,  
"modifiedBy": null,  
"modifiedTime": 1716367224,  
"fingerPrints": null,  
"isModified": null,  
"colorCode": null,  
"oldPassPhrase": null,  
"oldPassPhraseCryptKey": null,  
"awsFingerPrint": null,  
"sourceIp": [],  
"destIp": [  
  "192.168.60.130"  
],  
"createdTime": 1716366952000,  
"isPushAutomatically": null,  
"isRotateAutomatically": null,  
"autoRotate": null,  
"workOrderDetail": null,  
"groupIds": [  
  "5767bcef3465bfbf73e44727"  
],  
"readWrite": false,  
"keyPath": null,  
"renewDate": 0,  
"expiryDate": 0,  
"isExpired": false,  
"hostUserName": null,  
"workOrderMandate": null,  
"initiatedTime": 1716366952000,
```

```

"complianceDescription": null,
"agentUuid": null,
"keyType": null,
"filePaths": [
  "pe-cert-apvx-node02~/~/etc/ssh/appviewxssh/ssh_host_ECDSA_key",
  "pe-cert-apvx-node02~/~/etc/ssh/appviewxssh/ssh_host_ECDSA_key.pub"
],
"symLinks": [],
"privateKeyFileProps": {
  "readable": "true",
  "executable": "false",
  "writable": "true"
},
"publicKeyFileProps": {
  "readable": "true",
  "executable": "false",
  "writable": "true"
},
"clientMachineNames": [],
"serverMachineNames": [
  "pe-cert-apvx-node02"
],
"keyFoundInDiscoverySeqIdRanges": [
  {
    "from": 30,
    "to": 30
  }
],
"firstDiscovery": 1716367184461,
"previousDiscovery": 0,
"currentDiscoveredTime": 1716367184461,
"certificate": [
  {
    "fingerprint": "+1cEqAhMLynje99v9hjeB3QEooh1sqEPQH1303d5P8o",
    "certType": "Host",
    "rawFPString": "ECDSA-CERT SHA256:+1cEqAhMLynje99v9hjeB3QEooh1sqEPQH1303d5P8o",
    "signingCA": "ECDSA SHA256:k/k8+W/SlzdaK0ajyat/l3FVpnWpPpndaP0qTM7lf7M (using ecdsa-sha2-nistp256)",
  }
]

```

```

"keyId": "pe-cert-apvx-node02.lab.appviewx.net",
"serialNumber": "3515119686734784",
"rawCertType": "ecdsa-sha2-nistp256-cert-v01@openssh.com host certificate",
"validFrom": 1716366944000,
"validTo": 1747902944000,
"validity": 364,
"validityUnit": "days",
"expiresIn": "364 days",
"principals": [
  "192.168.60.130",
  "pe-cert-apvx-node02.lab.appviewx.net"
],
"cryptoKeyForCertContent": "opj82wtc1bylx4igskt7ra724",
"certStatus": "Active",
"filePaths": [
  {
    "hostName": "pe-cert-apvx-node02",
    "paths": [
      "pe-cert-apvx-node02--/etc/ssh/appviewxssh/ssh_host_ECDSA_key-cert.pub"
    ]
  }
],
"onlyForWebTerminal": false
}
],
"sharedType": "single",
"excludeFromSharedKeyReportEndTime": 0,
"excludeFromWeakKeyReportEndTime": 0,
"excludeFromOrphanKeyReportEndTime": 0,
"excludeFromSuspiciousKeyReportEndTime": 0,
"discoveryIdWithNewState": {
  "30": true
},
"eligibleForRollback": false,
"backupData": null,
"sharedKey": false,
"weakKey": false,

```

```

"riskKey": false,
"discovered": true,
"accessRequest": false,
"new": true,
"keyDownload": false,
"privateKeyDeleted": false,
"publicKeyDeleted": false,
"keyFilePermission": [
  {
    "user": null,
    "userHomeDirectory": null,
    "userGroup": null,
    "filePath": null,
    "deviceName": null,
    "fileProperties": null
  }
],
"selected": false,
"upload": false,
"passphraseValidated": false,
"hasPrivateKey": true,
"_id": "664daf7868cf79570aab88b5"
},
"age": "0 day",
"clientMachineNames": null,
"serverMachineNames": null,
"groupPermission": [
  {
    "Default_Key_Group": "RW"
  }
],
"permission": "RW",
"compliance": "Compliant",
"createdTime": 0,
"displayName": "RotateKeys_admin_1716367184410-B0-001",
"hostComplianceGroup": null,
"joinedHostGroups": null,

```

```

"hostName": null,
"associatedUsers": null,
"colorCode": "newKeys",
"complianceDescription": "",
"hostInfos": null,
"keyComplianceGroup": null,
"accessGroup": null,
"selected": false,
"_id": null
}
],
"iTotalDisplayRecords": 1,
"serverTime": 1716377205029
},
"message": "User keys 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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Hosts](#)

Fetch Discovery Status

The API will fetch discovery status information.

Before you begin

Before attempting to fetch discovery status, the user has to ensure the following:

- Discovery should be present in AppViewX.

Request Structure

Endpoint:	/ssh/discovery/status
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/discovery/status?gwsouce=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
input	(Optional) Input Parameters to fetch discovery status.

Name	Description
	Type: Input
filter	(Mandatory) Filter parameters to fetch discovery status. Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to fetch discovery status. Type: String
keywordSearch	(Optional) Keyword and value to search and retrieve discovery. Example: {"keyname": "RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	(Mandatory) Column name to be sorted. Type: String
sortOrder	(Mandatory) Order to be sorted. Possible values: asc, desc
start	(Mandatory) Start count of the discovery status. Type: String
max	Count of the discovery status. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the discovery status Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of discovery information which matches the search criteria Type: List
iTotalDisplayRecords	Total number of discovery available for the search criteria

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Discovery information retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To fetch status of the discovery “TestDiscovery” using **/discovery/status** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/discovery/status?gwsource=external
```

Sample Request

```
{
  "input": {
    "freeSearch": "TestDiscovery"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Response

```
{
  "response": {
    "data": [
      {
        "discoveryName": "TestDiscovery",
        "displayDiscoveryName": "TestDiscovery",
        "discoveryMode": "managedDevices",
        "scheduleType": "instant",
        "status": "Completed",
      }
    ]
  }
}
```

```

"startTime": 1717135034277,
"endTime": 1717135074037,
"description": "",
"discoveryId": "665966ba1500c06eadf8c022",
"recurrenceType": "NA",
"totalHosts": 1,
"successfulHosts": 1,
"failedHosts": 0,
"_id": "665966badd048f3caec5da37"
}
],
"iTotalDisplayRecords": 1
},
"message": "Discovery information 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.

**Note:**

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

Rotate SSH Keys

The API will rotate ssh keys on the device.

Before you begin

Before attempting to rotate keys, the user has to ensure the following:

- Key should be present in AppViewX.

Request Structure

Endpoint:	/ssh/key/rotate
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/rotate?gwsouce=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String
	Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
type	(Mandatory) Type of the key.
	Possible values: User Keys, Host Keys
	Type: String

Name	Description
uuids	(Mandatory) Uuid of the keys to be rotated. Type: List

Response Structure

200 Accepted returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the status of action performed Type: String
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 Accepted	NA	Rotate Operation has been triggered
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
400 Bad Request	AVX-VLDTN-001	<p>Mandatory field is missing or invalid values specified - <<field name>></p> <p>Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.</p>

Sample Request/Response**Use Case**

To rotate key “1f4158e4-c12d-3097-a058-7bb2817e132b” using **/key/rotate** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/rotate?gwsouce=external
```

Sample Request

```
{
  "type": "User Keys",
  "uuids": ["1f4158e4-c12d-3097-a058-7bb2817e132b"]
}
```

Sample Response

```
{
  "response": "Rotate operation has been triggered",
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Restore SSH Keys](#)
- [Rollback SSH Keys](#)
- [Delete SSH Keys](#)

Restore SSH Keys

The API will restore ssh keys on the device.

Before you begin

Before attempting to restore keys, the user has to ensure the following:

- Key should be present in Recently deleted inventory of AppViewX.

Request Structure

Endpoint:	/ssh/key/restore
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/restore?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be ' application/json '.

Request Parameters (continued)

Name	Description
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
type	(Mandatory) Type of the key. Possible values: User Keys, Host Keys Type: String
uuids	(Mandatory) Uuid of the keys to be restored. Type: List

Response Structure

200 Accepted returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the status of action performed Type: String
message	Success message of the action or failure description in case of error. Will be non-null for failure response

Response Parameters (continued)

Name	Description
	Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 Accepted	NA	Restore Operation has been triggered for backup keys
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To restore key "1f4158e4-c12d-3097-a058-7bb2817e132b" using **/key/restore** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/restore?gwsources=external
```

Sample Request

```
{
  "type": "User Keys",
  "uuids": ["1f4158e4-c12d-3097-a058-7bb2817e132b"]
}
```

Sample Response

```
{
  "response": "Restore Operation has been triggered for backup keys",
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **/ssh/host/create**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- [Rotate SSH Keys](#)
- [Restore SSH Keys](#)
- [Rollback SSH Keys](#)
- [Delete SSH Keys](#)

Rollback SSH Keys

The API will rollback ssh keys on the device.

Before you begin

Before attempting to rollback keys, the user has to ensure the following:

- Key should be present in Recently rotated inventory of AppViewX.

Request Structure

Endpoint:	/ssh/key/rollback
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/rollback?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String
	Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
discoveryName	Discovery Name (E.g. SampleDiscovery).
<i>Query</i>	If this value not given, hosts will be fetched from user keys inventory.
	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
type	(Mandatory) Type of the key. Possible values: User Keys, Host Keys Type: String
uuids	(Mandatory) Uuid of the keys to be rolled back. Type: List
keyInventory	Specified UUID are from key inventory or not. The values are: <ul style="list-style-type: none"> • true - UUIDs of ssk keys from inventory. • false - UUIDs of keys from rotated keys inventory.

Response Structure

200 Accepted returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the status of action performed Type: String
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 Accepted	NA	Rollback Operation has been triggered for backup keys
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response

Use Case

To rollback key "1f4158e4-c12d-3097-a058-7bb2817e132b" using **/key/rollback** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/rollback?gwsource=external
```

Sample Request

```
{
  "type": "User Keys",
  "uuids": ["1f4158e4-c12d-3097-a058-7bb2817e132b"],
  "keyInventory": false
}
```

Sample Response

```
{
  "response": "Rollback Operation has been triggered for backup keys",
  "message": null,
  "appStatusCode": null,
}
```

```
"tags": {},
"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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Restore SSH Keys](#)
- [Rotate SSH Keys](#)
- [Delete SSH Keys](#)

Delete SSH Keys

The API will delete ssh keys on the inventory, device or from deleted inventory.

Before you begin

Before attempting to delete keys, the user has to ensure the following:

- Key should be present in Recently deleted inventory of AppViewX. (or)
- Key should be present in the keys inventory of AppViewX.

Request Structure

Endpoint:	/ssh/key/delete
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/delete?gwsources=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be 'application/json' .
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
discoveryName <i>Query</i>	Discovery Name (E.g. SampleDiscovery). If this value not given, hosts will be fetched from user keys inventory. Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
type	(Mandatory) Type of the key. Possible values: User Keys, Host Keys Type: String
uuids	(Mandatory) Uuid of the keys to be deleted. Type: List

Name	Description
deleteFrom	<p>The place from where the keys to be deleted.</p> <p>inventory - Deletes from keys inventory</p> <p>backupInventory - Deletes from deleted keys inventory</p> <p>endpoint - Deletes from keys inventory and the device</p> <p>Type: String</p>

Response Structure

200 Accepted returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	<p>Contains the status of action performed</p> <p>Type: String</p>
message	<p>Success message of the action or failure description in case of error. Will be non-null for failure response</p> <p>Type: String</p>
appStatusCode	<p>Application-specific status code for the response. Will be non-null for failure response</p> <p>Type: String</p>
tags	<p>More info in case of failure response</p>

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 Accepted	NA	Deletion Operation has been triggered
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.
400 Bad Request	SSH-KEY-REM-013	Remediation could not be progressed as the user does not have Read-Write access.

Sample Request/Response**Use Case**

To delete key “1f4158e4-c12d-3097-a058-7bb2817e132b” using **/key/delete** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/key/delete?gwsouce=external
```

Sample Request

```
{
  "type": "User Keys",
  "uuids": ["1f4158e4-c12d-3097-a058-7bb2817e132b"],
  "deleteFrom": "inventory"
}
```

Sample Response

```
{
  "response": "Deletion Operation has been triggered",
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [SSH Create CA](#)

Search Backup User Keys


The API will search user keys and its information from the backup keys.

Before you begin

Before attempting to searching user keys from the backup, the user has to ensure the following:

- Keys should be backed up and present in AppViewX.

Request Structure

Endpoint:	/ssh/search/userKeys
	<div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: The same API has been used for the following: <ul style="list-style-type: none"> • User Keys Search • User Keys Search from discovery </div>
Type:	POST
Sample URL:	<p>https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/userKeys?gwsource=external</p> <p>To understand the elements of the sample URL, click here.</p>
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if username and password are not provided.</p>
username	(Mandatory) AppViewX login username
<i>Header</i>	<p>Type: String</p> <p>Constraints: Required if sessionId is not provided.</p>

Request Parameters (continued)

Name	Description
password <i>Header</i>	(Mandatory) AppViewX login password Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload Type: String Constraints: The value of the param should be 'application/json' .
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
keysFrom <i>Query</i>	Inventory Name Possible Values: rotatedInventory or deletedInventory If this value not given, hosts will be fetched from user key inventory. Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request Type: Payload

Payload

Name	Description
input	(Optional) Input Parameters to fetch user keys from backup inventory. Type: Input

Name	Description
filter	(Mandatory) Filter parameters to fetch user keys from backup inventory. Type: Filter

Input

Name	Description
freeSearch	Search text to find user key information from backup inventory. Type: String
keywordSearch	Keyword and value to search and retrieve user key information. Example: {"keyname": "RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	Column name to be sorted. Type: String
sortOrder	Order to be sorted. Possible values: asc, desc
start	Start count of the user keys to be fetched from backup inventory. Type: String
max	Count of the user keys to be fetched from backup inventory. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the user keys Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of user key information which matches the search criteria. Type: List
totalCount	Total number of user keys available for the search criteria

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	User keys retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case**

To search backup user key “FetchKey_admin_1716211635851-B0-022” using **/search/userKeys** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/userKeys?gwsouce=external
```

Sample Request 1

```
{
  "input": {
    "freeSearch": "FetchKey_admin_1716211635851-B0-022"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Request 2

```
{
  "input": {
    "keywordSearch":{"keyname":"FetchKey_admin_1716211635851-B0-022"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

```
}
}
```

Sample Response

```
{
  "response": {
    "statusCode": 200,
    "message": "Rotated Keys retrieved successfully for action ",
    "data": [
      {
        "type": "ECDSA",
        "name": "FetchKey_system_1717200035373-B0-001",
        "bitLength": "256",
        "comment": "Rotated from ECDSA to ECDSA on 2024-05-31 14:59:28 GMT",
        "fingerPrint": "OZo/vOaX+O21a+Z5dS5ssSSVKbfWNG7G5a2D+S0mXA0",
        "keyStatus": "Managed",
        "groupName": "Default_Key_Group",
        "period": "lifetime",
        "uuid": "5cc7d433-7745-3667-8aa1-596f9e607bb9",
        "keyWords": [
          "discoveredKeys",
          "256",
          "Rotated from ECDSA to ECDSA on 2024-05-31 14:59:28 GMT",
          "pe-cert-apvx-node02.lab.appviewx.net",
          "userKeys",
          "OZo/vOaX+O21a+Z5dS5ssSSVKbfWNG7G5a2D+S0mXA0",
          "FetchKey_system_1717200035373-B0-001",
          "ECDSA"
        ],
        "active": true,
        "createdBy": "discoveredKeys",
        "awsDiscoveredKey": true,
        "associatedUsers": [
          "pe-cert-apvx-node02.lab.appviewx.net--admin"
        ],
        "modifiedTime": 1717200075,
        "sourceIp": [
          "192.168.60.130"
        ]
      }
    ]
  }
}
```

```

    ],
    "destIp": [],
    "createdTime": 1717167614000,
    "groupIds": [
      "5767bcef3465bfbf73e44727"
    ],
    "readWrite": false,
    "renewDate": 0,
    "expiryDate": 0,
    "isExpired": false,
    "initiatedTime": 1717167614000,
    "filePaths": [
      "pe-cert-apvx-node02.lab.appviewx.net--~/home/admin/.ssh/id_ECDSA_appviewx_dev-group"
    ],
    "symLinks": [
      {
        "deviceName": "pe-cert-apvx-node02.lab.appviewx.net",
        "sourcePath": "/home/admin/.ssh/id_ECDSA_appviewx_dev-group",
        "targetPath": "/home/admin/.ssh/appviewxssh/id_ECDSA_20240531_145928_527_1"
      }
    ],
    "privateKeyFileProps": {
      "readable": "true",
      "executable": "false",
      "writable": "true"
    },
    "clientMachineNames": [
      "pe-cert-apvx-node02.lab.appviewx.net"
    ],
    "serverMachineNames": [],
    "keyFoundInDiscoverySeqIdRanges": [
      {
        "from": 33,
        "to": 33
      }
    ],
    "firstDiscovery": 1717200035397,

```

```

"previousDiscovery": 0,
"currentDiscoveredTime": 1717200035397,
"certificate": [],
"sharedType": "single",
"excludeFromSharedKeyReportEndTime": 0,
"excludeFromWeakKeyReportEndTime": 0,
"excludeFromOrphanKeyReportEndTime": 0,
"excludeFromSuspiciousKeyReportEndTime": 0,
"discoveryIdWithNewState": {
  "33": true
},
"eligibleForRollback": false,
"parentUuid": "51b51de5-3428-3ce7-b99d-9ba59a2cda81",
"parentFingerPrint": "HXbeNF4F6HdaKjiiULgkF99YKgjC9YZsPyrn/TNlc1k",
"parentBitLength": "256",
"parentAlgorithm": "ECDSA",
"backupAction": "Rotate",
"audit": {
  "createdOn": 1717222406512,
  "createdBy": "admin",
  "updatedOn": 0
},
"deletionDate": 0,
"backupKeyType": "userKeys",
"actionCompleted": true,
"sshdRemovedStatusWithPath": {
  "/home/admin/.ssh/id_ECDSA_appviewx_dev-group": false
},
"sharedKey": false,
"accessRequest": false,
"discovered": true,
"new": true,
"keyDownload": false,
"privateKeyDeleted": false,
"publicKeyDeleted": false,
"keyFilePermission": [
  {

```

```

"user": "admin",
"userHomeDirectory": "/home/admin",
"userGroup": "admin",
"filePath": "pe-cert-apvx-node02.lab.appviewx.net--~/home/admin/.ssh/id_ECDSA_appviewx_dev-group",
"deviceName": "pe-cert-apvx-node02.lab.appviewx.net",
"fileProperties": [
  {
    "readable": false,
    "writable": false,
    "executable": false,
    "permissionFor": "Group"
  },
  {
    "readable": false,
    "writable": false,
    "executable": false,
    "permissionFor": "Others"
  },
  {
    "readable": true,
    "writable": true,
    "executable": false,
    "permissionFor": "Owner"
  }
]
},
"selected": false,
"weakKey": false,
"riskKey": false,
"upload": false,
"passphraseValidated": false,
"hasPrivateKey": true,
"_id": "665a64cb5744cb4ca8c54fd4"
}
],
"totalCount": 1

```

```

},
"message": "User keys 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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search User Keys](#)

Search Backup Host Keys


The API will search host keys and its information from the backup inventory.

Before you begin

Before attempting to searching host keys from the backup inventory, the user has to ensure the following:

- Keys should be backed up and present in AppViewX.

Request Structure

Endpoint:	/ssh/search/hostKeys
	<div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: The same API has been used for the following:</p> <ul style="list-style-type: none"> • Host Keys Search • Host Keys Search From Discovery </div>
Type:	POST
Sample URL:	<p>https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?gwsource=external</p> <p>To understand the elements of the sample URL, click here.</p>
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload
<i>Header</i>	Type: String
	Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
keysFrom	Inventory Name
<i>Query</i>	Possible Values: rotatedInventory or deletedInventory. If this value not given, hosts will be fetched from user key inventory.
	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request
<i>Body</i>	Type: Payload

Payload

Name	Description
input	(Optional) Input Parameters to fetch host keys from backup inventory. Type: Input
filter	(Mandatory) Filter parameters to fetch host keys from backup inventory. Type: Filter

Input

Name	Description
freeSearch	Search text to find host key information from backup inventory. Type: String
keywordSearch	Keyword and value to search and retrieve host key information. Example: {"keyname": "RotateKeys_admin_1716367661908-B0-001"}

Filter

Name	Description
sortColumn	Column name to be sorted. Type: String
sortOrder	Order to be sorted. Possible values: asc, desc
start	Start count of the host keys to be fetched from backup inventory. Type: String
max	Count of the host keys to be fetched from backup inventory.

Filter (continued)

Name	Description
	Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the host keys Type: response
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Response

Name	Description
data	List of host key information which matches the search criteria. Type: List
totalCount	Total number of host keys available for the search criteria.

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Host keys retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>> Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response

Use Case

To search backup host key "RotateKeys_admin_1716367184410-B0-001" using **/search/hostKeys** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?keysFrom=rotatedInventory&gwsource=externalhttps://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/search/hostKeys?keysFrom=rotatedInventory&gwsource=external
```

Sample Request 1

```
{
  "input": {
    "freeSearch": "RotateKeys_admin_1716367184410-B0-001"
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

```
}

```

Sample Request 2

```
{
  "input": {
    "keywordSearch":{"keyname":"RotateKeys_admin_1716367184410-B0-001"}
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Response

```
{
  "response": {
    "httpStatusCode": 200,
    "message": "Deleted Keys retrieved successfully for action ",
    "data": [
      {
        "type": "ECDSA",
        "name": "FetchKey_admin_1717154137217-B0-126",
        "bitLength": "521",
        "comment": "no comment",
        "fingerPrint": "fMCeH0raF/oeufXk4f9KaP7W8y4WJF1XWfincaSEDE",
        "keyStatus": "Managed",
        "groupName": "Default_Key_Group",
        "period": "lifetime",
        "uuid": "40cd00ae-76ef-3bfc-bd58-32959dc38453",
        "keyWords": [
          "pe-cert-apvx-node01.lab.appviewx.net",
          "discoveredKeys",
          "521",
          "no comment",
          "fMCeH0raF/oeufXk4f9KaP7W8y4WJF1XWfincaSEDE",
          "FetchKey_admin_1717154137217-B0-126",

```

```

    "hostKeys",
    "ECDSA"
  ],
  "active": true,
  "createdBy": "discoveredKeys",
  "awsDiscoveredKey": true,
  "associatedUsers": [],
  "modifiedTime": 1717395307,
  "sourceIp": [
    "192.168.60.129"
  ],
  "destIp": [],
  "createdTime": 0,
  "groupIds": [
    "5767bcef3465bfbf73e44727"
  ],
  "readWrite": false,
  "renewDate": 0,
  "expiryDate": 0,
  "isExpired": false,
  "initiatedTime": 0,
  "filePaths": [
    "pe-cert-apvx-node01.lab.appviewx.net-~/root/.ssh/known_hosts"
  ],
  "symLinks": [],
  "publicKeyFileProps": {
    "readable": "true",
    "executable": "false",
    "writable": "true"
  },
  "clientMachineNames": [
    "pe-cert-apvx-node01.lab.appviewx.net"
  ],
  "serverMachineNames": [],
  "keyFoundInDiscoverySeqIdRanges": [
    {
      "from": 22,

```

```
"to": 23
},
{
  "from": 25,
  "to": 25
},
{
  "from": 27,
  "to": 28
},
{
  "from": 30,
  "to": 30
},
{
  "from": 32,
  "to": 32
},
{
  "from": 34,
  "to": 34
},
{
  "from": 37,
  "to": 37
},
{
  "from": 39,
  "to": 40
}
],
"firstDiscovery": 1717154137239,
"previousDiscovery": 1717372835282,
"currentDiscoveredTime": 1717395222733,
"certificate": [],
"sharedType": "standalone",
"excludeFromSharedKeyReportEndTime": 0,
```

```
"excludeFromWeakKeyReportEndTime": 0,
"excludeFromOrphanKeyReportEndTime": 0,
"excludeFromSuspiciousKeyReportEndTime": 0,
"discoveryIdWithNewState": {
  "22": true
},
"eligibleForRollback": false,
"parentUuid": "6659b187c07be2338021f391",
"backupAction": "Delete",
"expireAt": 1717403655705,
"audit": {
  "createdOn": 1717403655705,
  "createdBy": "admin",
  "updatedOn": 0
},
"deletionDate": 1719995655705,
"backupKeyType": "hostKeys",
"actionCompleted": false,
"sharedKey": false,
"accessRequest": false,
"discovered": true,
"new": false,
"keyDownload": false,
"privateKeyDeleted": false,
"publicKeyDeleted": false,
"keyFilePermission": [],
"selected": false,
"weakKey": false,
"riskKey": false,
"upload": false,
"passphraseValidated": false,
"hasPrivateKey": false,
"_id": "6659b187c07be2338021f391"
}
],
"totalCount": 1
},
```

```

"message": "Host keys 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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Search Hosts](#)
- [Search Host Keys](#)

Fetch Status of an Infra Access Group Access Request

The API will fetch status of the given Infra Access Group's access request.

Before you begin

Before attempting to status of an access request, the user has to ensure the following:

- Access should be requested for the required Infra Access Group.
- Once an access request is submitted, make a note of *accessRequestId* returned from the response of access request creation.

Request Structure

Endpoint:	/ssh/access/list
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/access/list?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username. Type: String

Request Parameters (continued)

Name	Description
<i>Header</i>	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password.
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload.
<i>Header</i>	Type: String Constraints: The value of the param should be 'application/json' .
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
accessRequestId	ID of the access request.
<i>queryParam</i>	Type: String
Payload	(Mandatory) Parameters to view access requests
<i>Body</i>	Type: Payload

Payload

Name	Description
input	(Optional) Input parameters to find host information. Type: Input
filter	(Mandatory) Filter parameters to find host information. Type: Filter

Input

Name	Description
freeSearch	(Optional) Search text to find host information Type: String

Filter

Name	Description
sortColumn	(Mandatory)Column name to be sorted. Type: String
sortOrder	(Mandatory)Order to be sorted Type: String Possible values: asc, desc
start	(Mandatory)Start count of the hosts to be fetched. Type: String
max	(Mandatory) Count of the hosts to be fetched. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the access requests. Type: Response
message	Success message of the action or failure description in case of error. Will be non-null for failure response.

Response Parameters (continued)

Name	Description
	Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response. Type: String
tags	More info in case of failure response.

Response

Name	Description
data	List of access requests information which matches the search criteria. Type: List
iTotalDisplayRecords	Total number of access requests available.

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Access request information retrieved successfully
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	AVX-VLDTN-001	Mandatory field is missing or invalid values specified - <<field name>>

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.

Sample Request/Response**Use Case 1**

To fetch access request details of a particular access request using **/access/list** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/access/list?accessRequestId=<accessRequestId>&gwsouce=external
```

Sample Request

```
{
  "input": {
    "freeSearch": ""
  },
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "100"
  }
}
```

Sample Response

```
{
  "response": {
    "data": [
      {
        "userName": "appviewx",
        "accessGroup": "test",
        "accessDuration": "7 days",
        "accessDurationUnit": "days",

```

```

    "accessDurationValue": 7,
    "approvedBy": "admin",
    "requestStatus": "Approved",
    "requestAccessMode": "AppViewX Terminal",
    "accessStatus": "Accessible",
    "workflowId": "28",
    "workorderId": "0",
    "totalNumberOfHosts": 2,
    "successfulHosts": 2,
    "failedHosts": 0,
    "provisioningHosts": 0,
    "requestTimestamp": 1722596465924,
    "validToTimestamp": 1723201297070,
    "reviewTimestamp": 1722596495920,
    "certSerialNumber": "3527877650586052",
    "keyFingerPrint": "9nW1AAU+KK+5G1m2tEtCkX90AfpqfvC53IYeES2aRQ",
    "keyUUID": "9edb74bc-f85e-398d-b373-2626bf3323dc",
    "comments": "cert-based-access-type",
    "accessFor": "User",
    "requestedBy": "admin",
    "permission": "RW",
    "_id": "66acbc7124bdf3a45598ba6"
  }
],
  "iTotalDisplayRecords": 5
},
"message": null,
"appStatusCode": null,
"tags": {},
"headers": null
}

```

Use Case 2

To fetch first 2 access request details using **/access/list** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/access/list?accessRequestId=<accessRequestId>&gwsouce=external
```

Sample Request

```
{
  "input": {},
  "filter": {
    "sortColumn": "none",
    "sortOrder": "desc",
    "start": "0",
    "max": "2"
  }
}
```

Sample Response

```
{
  "response": {
    "data": [
      {
        "userName": "appviewx",
        "accessGroup": "test_ca_removal",
        "accessDuration": "7 days",
        "accessDurationUnit": "days",
        "accessDurationValue": 7,
        "requestStatus": "Pending Approval",
        "requestAccessMode": "AppViewX Terminal",
        "accessStatus": "Pending Approval",
        "workflowId": "185",
        "workorderId": "0",
        "totalNumberOfHosts": 0,
        "successfulHosts": 0,
        "failedHosts": 0,
        "provisioningHosts": 0,
        "requestTimestamp": 1717408016639,
        "validToTimestamp": 0,
        "reviewTimestamp": 0,
        "comments": "cert-based-access-type",
        "accessFor": "User",
        "requestedBy": "admin",
        "permission": "RW",
      }
    ]
  }
}
```

```
"_id": "665d911043a8710b62e6b7cb"
},
{
  "userName": "admin",
  "accessGroup": "test_ca_removal",
  "accessDuration": "1 hours",
  "accessDurationUnit": "hours",
  "accessDurationValue": 1,
  "approvedBy": "admin",
  "requestStatus": "Approved",
  "requestAccessMode": "AppViewX Terminal",
  "sshKeyGenerationType": "Generate via AppViewX",
  "accessStatus": "Expired",
  "workflowId": "184",
  "workorderId": "0",
  "totalNumberOfHosts": 1,
  "successfulHosts": 1,
  "failedHosts": 0,
  "provisioningHosts": 0,
  "requestTimestamp": 1717400190275,
  "validToTimestamp": 1717403801805,
  "reviewTimestamp": 1717400199329,
  "certSerialNumber": "1234535692720887",
  "keyFingerPrint": "nJ0PZqE1mjvPcJER2bA5SF5m0lsun9AQfExtKT+ouLo",
  "keyUUID": "192292cf-e695-3202-8d10-11e644a3996d",
  "comments": "",
  "accessFor": "Self",
  "requestedBy": "admin",
  "permission": "RW",
  "_id": "665d727e5517745f50c0b8e2"
}
```

```

],
  "iTotalDisplayRecords": 5
},
{
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **/ssh/host/create**
- **gwsource**: Source or origin of a gateway, for example: **external**.

Fetch Risk Report

The API will retrieve risk reports for data analysis and reporting purposes.

Before you begin

NA

Request Structure

Endpoint:	/ssh/report/risk
Type:	GET
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/report/risk?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username.
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password.
	Type: String

Request Parameters (continued)

Name	Description
<i>Header</i>	Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload.
<i>Header</i>	Type: String Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
reportName	Name of the report to be fetched.
<i>queryParam</i>	Type: String Possible Values: vulnerabilityReport, keyAgeReport, hostKeyAgeReport, hostKeyAlgorithmBitLengthReport, keyAlgorithmBitLengthReport, keySummaryReport, hostCertValidityReport, cipherSuiteServerSide, cipherSuiteClientSide, kexReportServerSide, kexReportClientSide, macReportServerSide, macReportClientSide
Payload	None
<i>Body</i>	

Response Structure

200 Accepted returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response of the vulnerability report Type: Response

Response Parameters (continued)

Name	Description
message	Success message of the action or failure description in case of error. Will be non-null for failure response. Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response. Type: String
tags	More info in case of failure response.

Response

Name	Description
keys	List of keys with vulnerability information. Type: List
lastUpdateTime	Last updated time of the report. Type: Long
securityScore	Security score information of the keys. Type: Number
totalKeys	Total number of keys available. Type: Number
totalWeakKeys	Total number of weak keys available. Type: Number

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 Accepted	NA	Success
401 Unauthorized	AVX_GW_003	<p>Authentication failed, reason - Invalid Credentials</p> <p>Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.</p>
400 Bad Request	AVX-VLDTN-001	<p>Mandatory field is missing or invalid values specified - <<field name>></p> <p>Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.</p>

Sample Request/Response

Use Case

To fetch risk reports using `/report/risk` API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/report/risk?reportName=reportname&gwsouce=external
```

Sample Request

```
None
```

Sample Response

```
{
  "response": {
    "keys": [
      {
        "category": "Suspicious Keys",
        "keysCount": 111,

```

```
"impact": "High"
},
{
  "category": "Shared User Keys",
  "keysCount": 1,
  "impact": "High"
},
{
  "category": "KEX Server Side Weak Algorithms",
  "keysCount": 31,
  "impact": "High"
},
{
  "category": "KEX Client Side Weak Algorithms",
  "keysCount": 36,
  "impact": "High"
},
{
  "category": "Shared Host Keys",
  "keysCount": 3,
  "impact": "High"
},
{
  "category": "Orphan Keys",
  "keysCount": 5,
  "impact": "Medium"
},
{
  "category": "Server Side Weak Ciphers",
  "keysCount": 11,
  "impact": "Medium"
},
{
  "category": "Client Side Weak Ciphers",
  "keysCount": 11,
  "impact": "Medium"
},
}
```

```

{
  "category": "Weak User Keys",
  "keysCount": 2,
  "impact": "Low"
},
{
  "category": "MAC Server Side Weak Algorithms",
  "keysCount": 6,
  "impact": "Low"
},
{
  "category": "MAC Client Side Weak Algorithms",
  "keysCount": 6,
  "impact": "Low"
},
{
  "category": "Weak Host Keys",
  "keysCount": 4,
  "impact": "Low"
}
],
"lastUpdateTime": 1717662901079,
"securityScore": 36.32568,
"totalKeys": 401,
"totalWeakKeys": 123
},
"message": "Success",
"appStatusCode": "SSH-NB-200",
"tags": {},
"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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsorce:** Source or origin of a gateway, for example: **external**.

Fetch Allowed Principals

The API will fetch the authorized principals configured in the host.

Before you begin

Before attempting to search principals for the host, make sure that the host has been provisioned with a cert based access request.

Request Structure

Endpoint:	/ssh/host/principal/list
Type:	POST

Sample URL:	<code>https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/host/principal/list?gwsource=external</code>
	To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId <i>Header</i>	(Mandatory) Session ID received after login. Type: String Constraints: Required if username and password are not provided.
username <i>Header</i>	(Mandatory) AppViewX login username. Type: String Constraints: Required if sessionId is not provided.
password <i>Header</i>	(Mandatory) AppViewX login password. Type: String Constraints: Required if sessionId is not provided.
Content-Type <i>Header</i>	(Mandatory) Specifies the nature of the data in the payload. Type: String Constraints: The value of the param should be ' application/json '.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request.

Request Parameters (continued)

Name	Description
	Type: Payload

Payload

Name	Description
input	(Mandatory) Input parameters to find host principal. Type: Input
filter	(Mandatory) Filter parameters to find host information. Type: Filter

Input

Name	Description
freeSearch	Search text to find host information. Type: String
deviceName	Name of the device for which principals need to be fetched. Type: String

Filter

Name	Description
sortColumn	Column name to be sorted. Type: String Possible values: deviceName, userName
sortOrder	Order to be sorted. Type: String

Filter (continued)

Name	Description
	Possible values: asc, desc
start	Start count of the users to be fetched. Type: String
max	Count of the users to be fetched. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response attributes for the hosts Type: Response
message	Success message of the action or failure description in case of error. Will be non-null for failure response. Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response. Type: String
tags	More info in case of failure response.

Response

Name	Description
data	List of user/principal information which matches the search criteria.

Name	Description
	Type: List
iTotalDisplayRecords	Total number of users available for the search criteria.

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Principal removal action has been triggered.
401 Unauthorized	AVX_GW_003	<p>Authentication failed, reason - Invalid Credentials</p> <p>Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.</p>
400 Bad Request	AVX-VLDTN-001	<p>Mandatory field is missing or invalid values specified - <<field name>></p> <p>Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.</p>

Sample Request/Response

Use Case

To fetch allowed principals for the host using **/host/principal/list** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/host/principal/list?gwsource=external
```

Sample Request

```
{
  "input": {
```

```

"freeSearch": "",
"deviceName": "TestServer1"
},
"filter": {
  "max": "25",
  "start": "0",
  "sortColumn": "principals",
  "sortOrder": "asc"
}
}

```

Sample Response

```

{
  "response": {
    "data": [
      {
        "deviceName": "TestServer1",
        "userName": "admin",
        "principals": [
          "admin"
        ],
        "audit": {
          "createdOn": 1716966974381,
          "createdBy": "system",
          "updatedOn": 0
        },
        "_id": "6656d63e9925e474582662ec"
      }
    ],
    "iTotalDisplayRecords": 1
  },
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.



Note:

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsouce:** Source or origin of a gateway, for example: **external**.

What's Next

- [Remove Allowed Principals](#)

Remove Allowed Principals

The API will trigger removal of the principals from the authorized principals file of the end device.

Before you begin

Before attempting to search principals for the host, make sure that the host has been provisioned with a cert based access request so that the principals are available for the host.

Request Structure

Endpoint:	/ssh/host/principal/remove
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/host/principal/remove?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username.
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password.
<i>Header</i>	Type: String Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload.
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: The value of the param should be 'application/json'.
gwsource <i>Query</i>	(Mandatory) Source from which the request is triggered. (E.g. external) Type: String
Payload <i>Body</i>	(Mandatory) Contains all the parameters to be sent in the request body for the post request. Type: Payload

Payload

Name	Description
users	(Mandatory) List of users for whom principal need to be removed. Type: List
deviceName	(Mandatory) Name of device from which principal has to be removed. Type: String
comment	(Optional) Reason for principal removal Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response message of the API. Type: String

Response Parameters (continued)

Name	Description
message	Success message of the action or failure description in case of error. Will be non-null for failure response. Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response. Type: String
tags	More info in case of failure response.

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response message for principal removal action.
message	Success message of the action or failure description in case of error. Will be non-null for failure response Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response Type: String
tags	More info in case of failure response

Status Codes**Status Codes and Description**

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Principal removal action has been triggered.

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
401 Unauthorized	AVX_GW_003	<p>Authentication failed, reason - Invalid Credentials</p> <p>Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.</p>
400 Bad Request	VALIDATION_ERROR_0004	<p>Mandatory field <<field name>> is missing or or empty</p> <p>Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.</p>
404 Not Found	ERR-SSH-NB-252	<p>SSH device not found.</p> <p>Possible remediation: Check and ensure that deviceName specified in the input refers to a device that exists in the SSH Host Inventory.</p>
400 Bad Request	ERR-SSH_NB-293	<p>Device not in successfully managed status. Action not allowed.</p> <p>Possible remediation: Check and ensure that deviceName specified in the input refers to a device that exists in the SSH Host Inventory.</p>

Sample Request/Response**Use Case**

To removed the allowed principals using **/host/principal/remove** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/host/principal/remove?gwsource=external
```

Sample Request

```
{
  "users":["admin"],
  "deviceName":"TestServer1",
  "comment":"Check"
}
```

Sample Response

```
{
  "response": "Principal removal action has been triggered",
  "message": null,
  "appStatusCode": null,
  "tags": {},
  "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.

**Note:**

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi**: Path parameter value (static) that is part of the endpoint's URL
- **Endpoint**: Endpoint of the API, for example: **/ssh/host/create**
- **gwsource**: Source or origin of a gateway, for example: **external**.

What's Next

- [Fetch Allowed Principals](#)

Revoke Certificate

The API will trigger revocation of the SSH certificate.

Before you begin

Before attempting to revoke the certificate, make sure that the certificate is in the *Active* state.

Request Structure

Endpoint:	/ssh/cert/revoke
Type:	POST
Sample URL:	https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/cert/revoke?gwsource=external To understand the elements of the sample URL, click here .
Headers:	
Content-Type:	application/json

Request Parameters

Name	Description
sessionId	(Mandatory) Session ID received after login.
<i>Header</i>	Type: String

Request Parameters (continued)

Name	Description
	Constraints: Required if username and password are not provided.
username	(Mandatory) AppViewX login username.
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
password	(Mandatory) AppViewX login password.
<i>Header</i>	Type: String
	Constraints: Required if sessionId is not provided.
Content-Type	(Mandatory) Specifies the nature of the data in the payload.
<i>Header</i>	Type: String
	Constraints: The value of the param should be ' application/json '.
gwsource	(Mandatory) Source from which the request is triggered. (E.g. external)
<i>Query</i>	Type: String
Payload	(Mandatory) Contains all the parameters to be sent in the request body for the post request.
<i>Body</i>	Type: Payload

Payload

Name	Description
caName	(Mandatory) Name of the CA that signed the certificate.
	Type: List
groupName	(Mandatory) Name of the key compliance group to which the certificate is assigned.

Name	Description
	Type: String
serialNumber	(Optional) Serial number of the certificate. Type: String Constraints: Required if certificate is not provided.
certificate	(Optional) Certificate content Type: String Constraints: Required if serialNumber is not provided.
reason	(Optional) Reason for revocation. Type: String

Response Structure

200 OK returns string of type application/json with the following body params.

Response Parameters

Name	Description
response	Contains the response message of the API. Type: Revoke Response
message	Success message of the action or failure description in case of error. Will be non-null for failure response. Type: String
appStatusCode	Application-specific status code for the response. Will be non-null for failure response. Type: String
tags	More info in case of failure response.

Revoke Response

revokedRequests	List of revoked requests. Type: List
failedRequests	List of failed requests. Type: List

Status Codes

Status Codes and Description

HTTP Status code	appStatusCode	Message and Possible remediation
200 OK	NA	Success
401 Unauthorized	AVX_GW_003	Authentication failed, reason - Invalid Credentials Possible remediation: Ensure that valid username and password or valid sessionId is provided as the header param.
400 Bad Request	VALIDATION_ERROR_0004	Mandatory field <<field name>> is missing or or empty Possible remediation: Check and ensure that valid value is provided for <<field name>> field in the request.
400 Bad Request	ERR-SSH_NB-270	Max request size exceeded::[Max request size allowed is 25] Possible remediation: Check and ensure that the number of requests specified in the payload is less than 25.
417 Expectation Failed	ERR-SSH-NB-298	One or more groups do not have permission to perform this action.

Status Codes and Description (continued)

HTTP Status code	appStatusCode	Message and Possible remediation
		Possible remediation: Check and ensure that the user has access to the specified key compliance group.

Sample Request/Response**Use Case**

To revoke SSH certificate using **revoke_certificate** API.

Request URL

```
https://<IP/HostName/TenantName>:<GWPORT>/avxapi/ssh/cert/revoke?gwsouce=external
```

Sample Request

```
[
  {
    "serialNumber":"3527182657550928",
    "caName":"Default-Infra-Access-Group.user.ca",
    "reason":"test",
    "groupName":"Default_Key_Group"
  },
  {
    "certificate":"<actual certificate content should be populated here>",
    "caName":"TestHostCA",
    "reason":"test",
    "groupName":"Default_Key_Group"
  }
]
```

Sample Response

```
{
  "response": {
    "revokedRequests": [
      {
```

```

    "caIdentifier": "Default-Infra-Access-Group.user.ca",
    "error": null,
    "certificate": "<actual certificate content will be present here>",
    "serialNumber": "3527182657550928",
    "revoked": true
  }
],
"failedRequests": [
  {
    "caIdentifier": "TestHostCA",
    "error": "No CA found with given name or Id",
    "certificate": "<actual certificate content will be present here>",
    "serialNumber": null,
    "revoked": false
  }
]
},
"message": "Success",
"appStatusCode": "SSH-NB-200",
"tags": {},
"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.

**Note:**

GWPORT is not required for SaaS setups.

Example: **31443**

- **avxapi:** Path parameter value (static) that is part of the endpoint's URL
- **Endpoint:** Endpoint of the API, for example: **/ssh/host/create**
- **gwsource:** Source or origin of a gateway, for example: **external**.

What's Next

- [SSH Create Certificate](#)