CAS Manager 22.04

CAS Manager is a Teradici management plane enabling users to configure, manage and monitor brokering of remote workstations. CAS Manager enables highly-scalable and cost-effective Cloud Access Software deployments by managing cloud compute costs by brokering PCoIP connections to remote workstations, see Cloud Access Software for supported hosts.

CAS Manager is offered in 2 variants – as a Teradici managed Service, and as an installable instance deployed and managed by the users in their on-premises or cloud environments.

This document covers the installable instance variant of CAS Manager.

For information on CAS Manager as a Service, see CAS Manager as a Service.

Where Do I Begin?

CAS Manager is a collection of microservices, and each microservice operates from its own docker container. These container images are deployed on a local lightweight Kubernetes (k3s) cluster, on a virtual machine. This cluster is set up on the virtual machine as part of the installation.

Before you begin installing CAS Manager, it is important to understand what other components are required by the CAS Manager to enable end to end brokering:

- CAS Manager
- MongoDB
- Hashicorp Vault/Azure Key Vault
- Cloud Access Connector
- Teradici PCoIP Registration Key
- Teradici PCoIP Client
- Teradici PCoIP Agent

MongoDB is the local data store that hosts all CAS Manager information, configurations and settings.

Hashicorp Vault is the secret storage where CAS Manager can store and encrypt all the secrets and keys.

Azure Key Vault is the cloud service from Microsoft that enables the secure storage of, and access to, secrets.

Cloud Access Connector is an access hub that facilitates PCoIP connections to remote desktops and workstations by providing user authentication, entitlement and security gateway services. Later in this document it will be referred to as the "Connector". It is installed on a separate VM that resides in your environment. Based on your requirements, you may need more than a single Connector. Please ensure you have read all the installation guidelines and prerequisites in the Connector section.

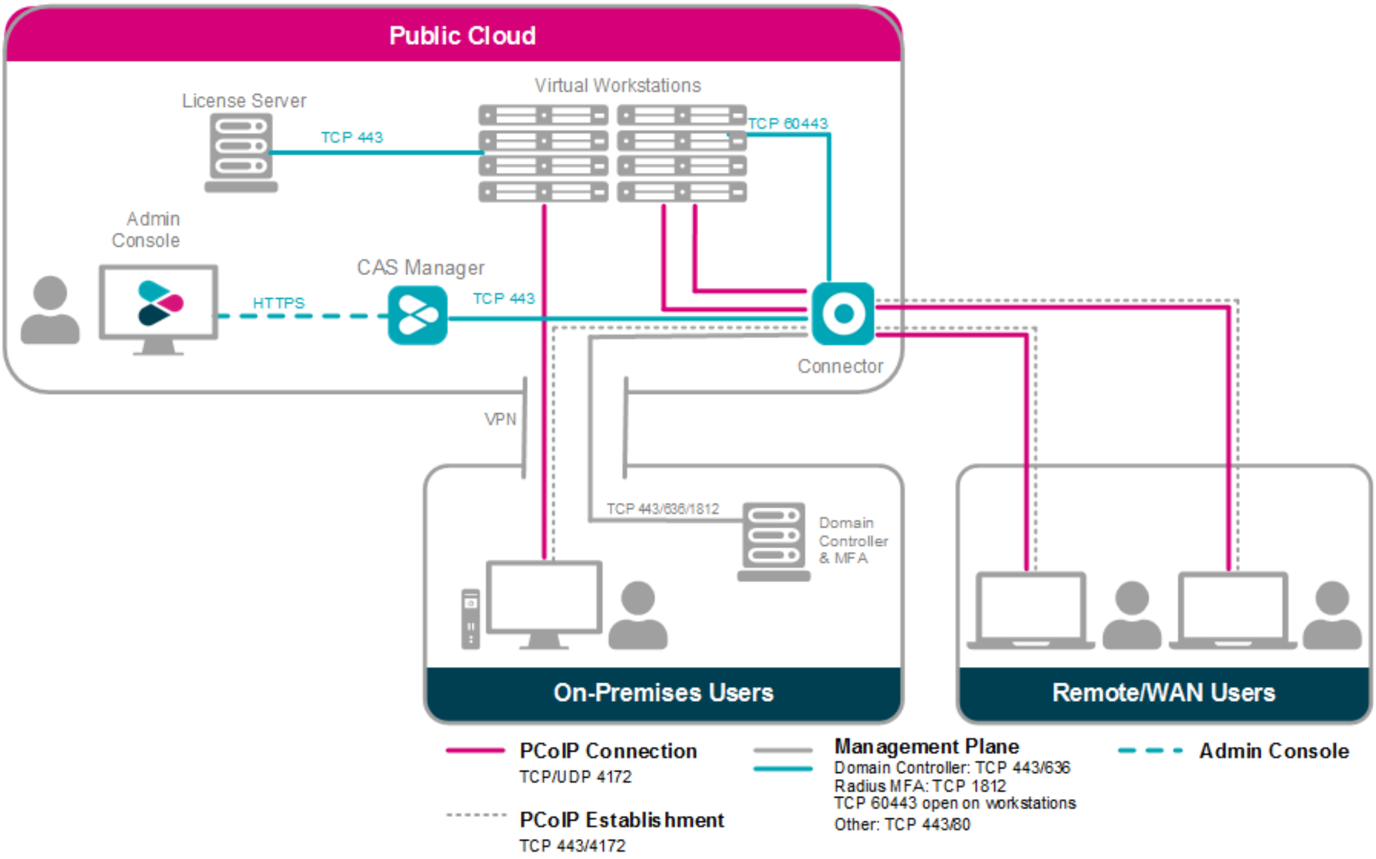
Where Do I Install CAS Manager?

The following architectural diagrams depict where CAS Manager can be installed in multiple infrastructures – be it the Public Cloud, On-Premises or a Hybrid deployemnt.

Please pay close attention to the number of Connectors required based on your setup, and the ports you may need to configure to allow PCoIP traffic (pre-session and in-session). These ports are outlined in the Ports and Connections table.

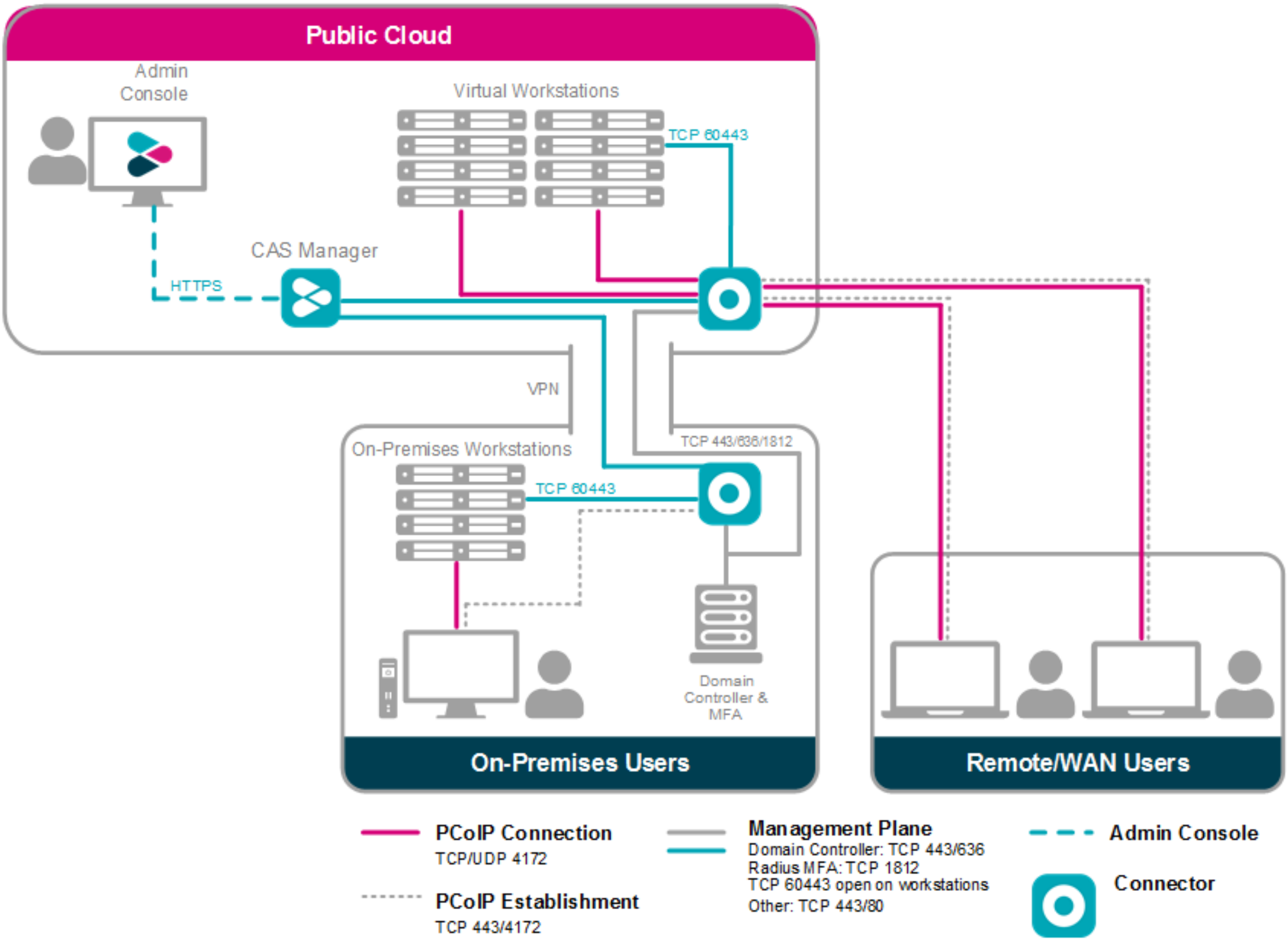
Public Cloud Deployment

The following diagram illustrates a public cloud deployment with CAS Manager.



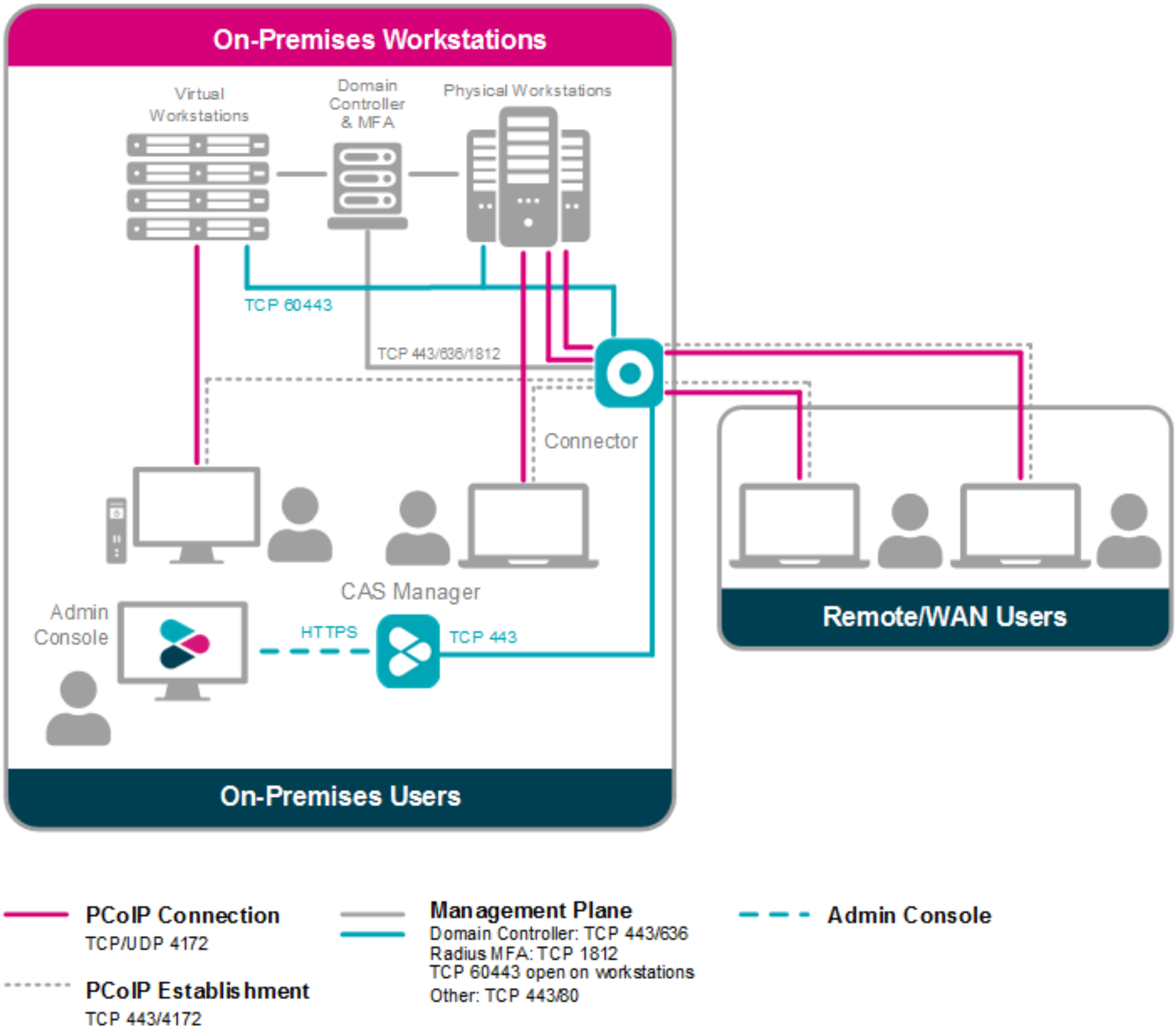
Hybrid Deployment

The following diagram illustrates a hybrid deployment where CAS Manager is deployed in the Public Cloud.



On-Premises Deployment

The following diagram illustrates an on-premises deployment with CAS Manager.



Ports and Connections

CAS Manager requires certain ports to be open to enable connections between the other components such as Connector, MongoDB, Vault etc. For detailed breakdown of the ports and connection descriptions for Connector, see Firewall and Load Balancing Considerations.

The following table outlines the required ports and connections for CAS Manager:

Component	Allow	Port/ Protocol	Source/Destination Component	Description
CAS Manager	Inbound	443/TCP	From administrative web browsers, HTTP request clients and Connector.	To enable access to CAS Manager.
CAS Manager	Outbound	443/TCP	To the public license server.	Validates the CAS registration code.
CAS Manager	Outbound	8200/TCP	To external Vault.	Stores CAS Manager secrets.
CAS Manager	Outbound	27017/ TCP	To external MongoDB.	Stores CAS Manager data.
CAS Manager	Outbound	636/TCP	To Domain Controller.	Authenticates users to CAS Manager.
CAS Manager	Outbound	53/UDP	To DNS.	Domain name resolution.

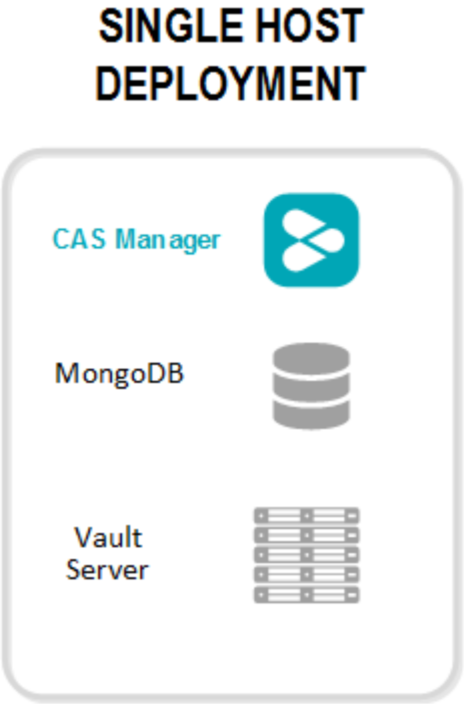
What Deployment Topology Can I Use?

In terms of deployment topologies and scenarios, CAS Manager is flexible and can be deployed in a single host, or with multiple hosts, depending on your organization's network environment and operational requirements. The possible deployment topologies are outlined below. Connector(s) are not included in these diagrams, they will be deployed on additional host(s) separately.

Single Host Deployment

This deployment configuration is when CAS Manager and MongoDB and Vault server are running on a single host, it can be deployed on a virtual machine on any cloud or on-premise. It should be used for getting started with CAS Manager for initial prototyping or smaller scale production deployments. If you use this configuration for production environment you must ensure there is a backup and restore process in place. This is necessary to minimize the loss of data and to minimize down time.

For information on installing CAS Manager as part of a single host deployment, see Installing CAS Manager - Default Configuration.

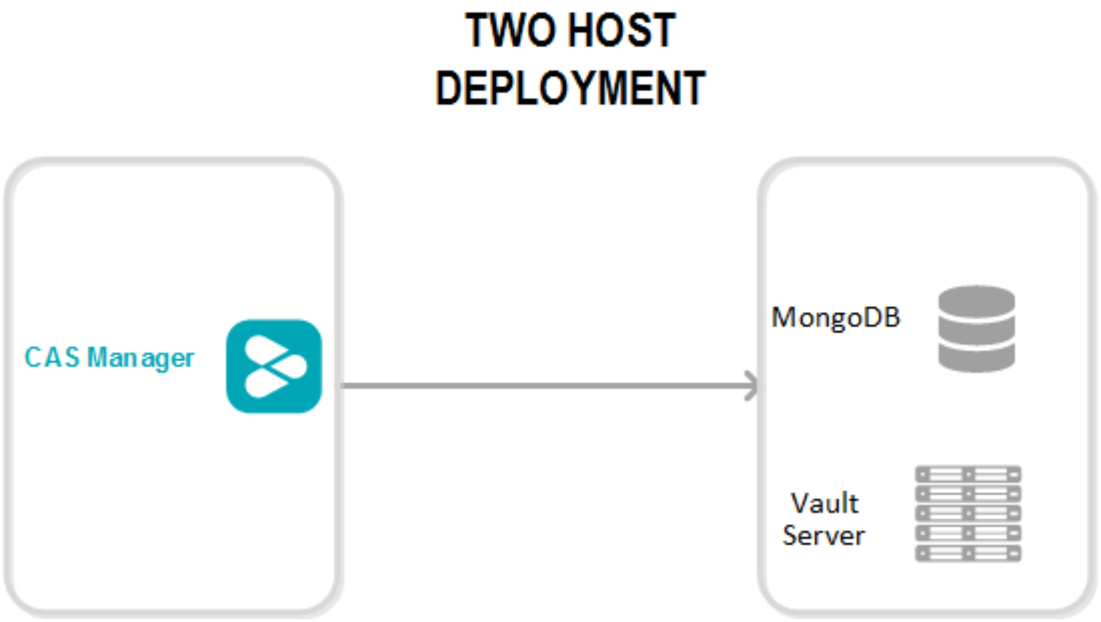


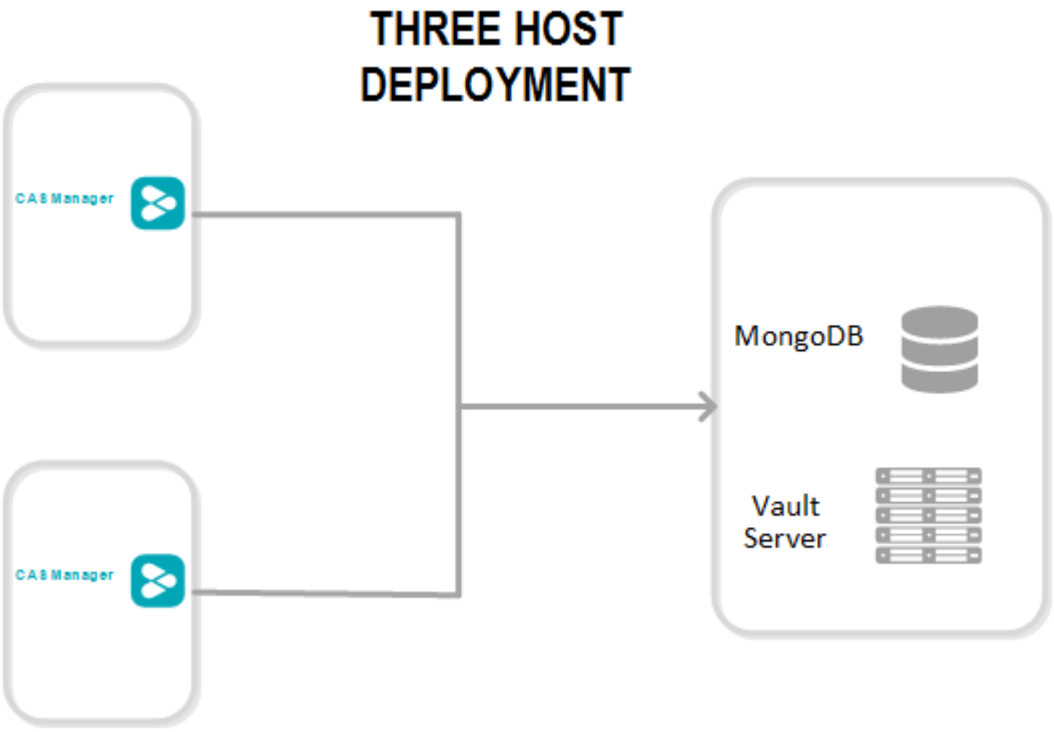
Two/Three Host Deployment

This deployment configuration is when CAS Manager, MongoDB and Vault server are running on separate hosts. By hosting the database and secret storage on a separate machine, it reduces the risk of data loss in the case of CAS Manager server failure. This configuration enables high-availability and scalability for CAS Manager by deploying multiple instances of CAS Manager. This configuration has the following limitations:

- With only one instance of MongoDB and Vault deployed, high-availability is not available to the data persistence layer, and a backup and restore process must be in place for the server hosting MongoDB and Vault to minimize data loss.
- You can configure this deployment on virtual machines hosted on-premises or on any cloud.
- This configuration requires a certain level of technical knowledge around MongoDB and Vault to properly deploy and operate these external components. For detailed deployment instructions on installing and configuring MongoDB and Vault in a single virtual machine to be used by CAS Manager, see the following KB article.

For information on installing CAS Manager as part of a two/three host deployment, see Installing CAS Manager - External Configuration.



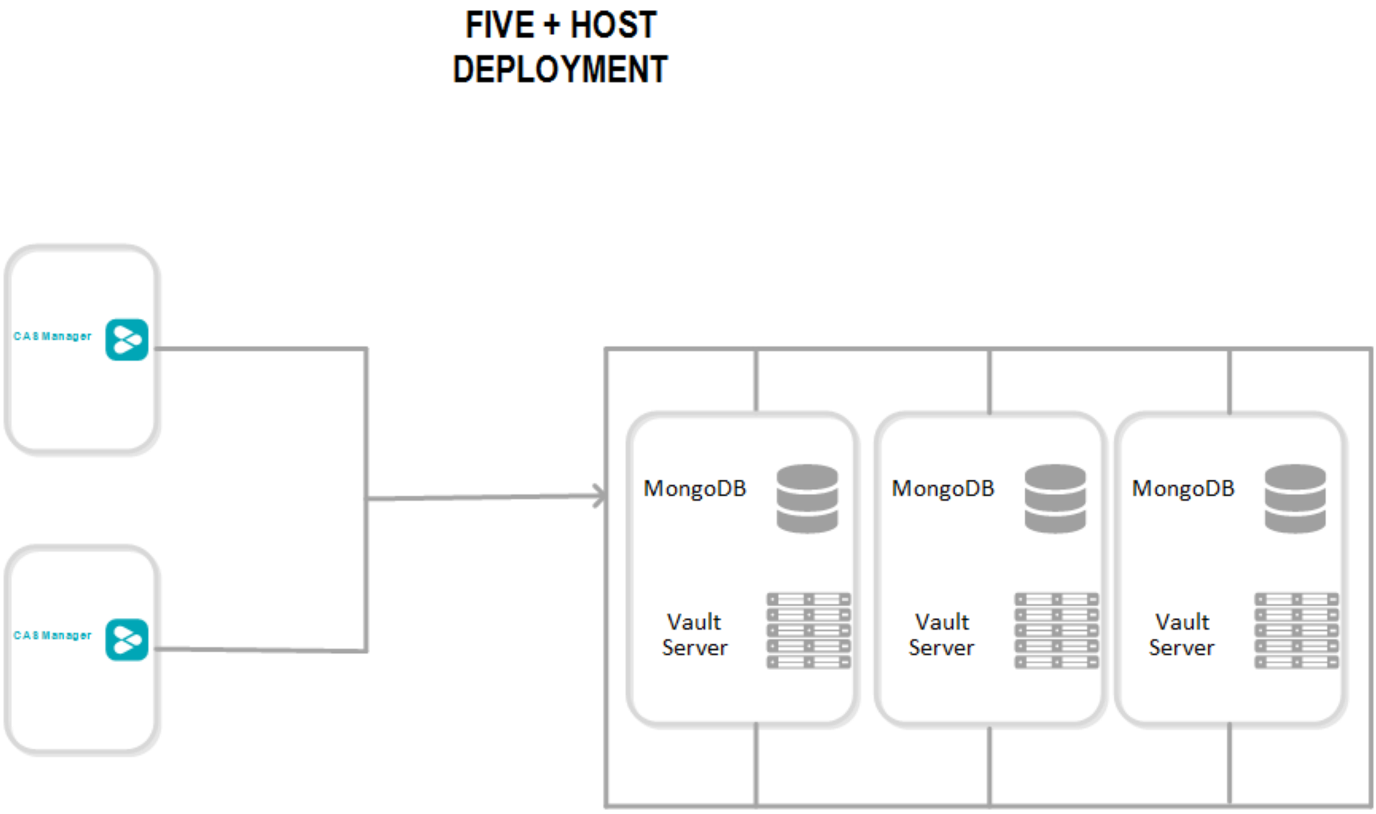


Five or more Hosts Deployment

This deployment configuration provides high-availability for both CAS Manager, and MongoDB and Vault server which are on separate hosts. In this configuration two or more CAS Manager instances provides high-availability using a load balancer. The hosts that contain the MongoDB and Vault server provide a basic high-availability for data persistence with a failure tolerant of 1. This configuration requires the following working knowledge:

- This is a complex environment and requires you to have working knowledge of installing, configuring and operating the MongoDB and Vault server services in a high-availability setup. Visit MongoDB and Hashicorp Vault official documentation sites for detailed instructions on how to carry out these steps.

For information on installing CAS Manager as part of a five or more host deployment, see Installing CAS Manager - External Configuration.



How Do I Install CAS Manager?

You need to setup and install a dedicated virtual machine which will host CAS Manager. This virtual machine needs to meet certain system requirements which are outlined in the sections below. If you are using an external MongoDB and secret storage you need to prepare these components before installing CAS Manager, and then configure them afterwards. The available configurations are outlined below.

Connector Installation

Once you have installed CAS Manager using either of the configurations below, you need to install the Connector. This should take roughly **1 hour** to complete.

Using a Default Database and Secret Storage

This is the default installation of CAS Manager where an instance of MongoDB and Vault is deployed as part of the installation. Installation of these components is seamlessly built into the CAS Manager installer. This configuration does not scale beyond a single CAS Manager instance and does not support high availability. For more information on this configuration, see Installing CAS Manager - Default Configuration.

Installation Time

Installing CAS Manager with the default database and secret storage should take roughly **45 minutes** to complete. It should take a further **1 hour** to install the Connector.

Using an External Database and Secret Storage

With CAS Manager you can prepare and install your own instances of MongoDB and Vault, or you can use an Azure Key Vault service, on a different virtual machine, by following the guidelines in the installation section. This enables you to upgrade or re-install CAS Manager, and makes a high-availability service available. For more information on this configuration, see Installing CAS Manager - External Database and Secret Storage Configuration.

Production Environments

Installing CAS Manager with an external database and secret storage should take roughly **2 hours** to complete. It should take a further **1 hour** to install the Connector.

Installing CAS Manager - Default Configuration

The following section outlines how to install CAS Manager with the default database and secret storage.

Installation Time

The default configuration of CAS Manager uses an internal Vault and MongoDB. It generates self-signed TLS certificates to use for it's gateway. It will take roughly **45 minutes** to complete the installation.

Data Migration

CAS Manager does not do any data migration when configuring your database and secret storage application. Any data stored when CAS Manager is used with the default database and secret storage configuration, will not be transferred if the same CAS Manager instance is re-configured to run with an external database and secret storage.

Firewall and Load Balancing Considerations

For information on firewall and load balancing ports connected to CAS Manager and Connector, see Firewall and Load Balancing Considerations

Preparing the CAS Manager Virtual Machine

The following section outlines how to prepare the system requirements, firewall configurations and proxy configurations on the CAS Manager virtual machine:

System Requirements

You need to prepare a virtual machine that has the following requirements:

- Operating System: RHEL 8 and Rocky Linux 8.
- Minimum 8 GB RAM
- 4 CPU

- 60 GB Storage: If you are using LVM and `/var` is mounted on a separate volume, that volume must have 30GB or more in order for the installation to succeed and for CAS Manager to function properly.
- Active Directory permissions set to **List contents** and **Read all properties**. If you do not set these permissions you will be unable to connect to specific remote workstations.

Firewall Configuration

You must ensure your firewall is established and configured properly. Ensure port 443 is enabled in the firewall rules for the VM that CAS Manager is running on.

Configure the firewall that the virtual network CAS Manager is running by following the commands below:

1. Login to the CAS Manager VM by ssh from a bash shell as ***root***.
2. Check and confirm if firewalld is active by running the following command:

   ```
   sudo systemctl status firewalld
   ```

3. If `firewalld` is active, follow the steps outlined below for firewall configuration. If `firewalld` is inactive, and your organization does not require firewall on the CAS Manager VM, then skip the firewall configuration steps below and proceed to the remaining steps.
4. Run the following commands to configure the firewall:

```
sudo firewall-cmd --permanent --add-port=6443/tcp # virtual network flannel

sudo firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16 # This subnet is for the pods

sudo firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16 # This subnet is for the services

sudo firewall-cmd --reload
```

Proxy Configuration Variables

If HTTP/HTTPS proxy is used, then `HTTP_PROXY` , `HTTPS_PROXY` and `NO_PROXY` must be set. For `NO_PROXY` , specific IP addresses or domain names of service that are internal must be added. IP address ranges like "10.0.0.0/8" will not work; exact IP addresses or domain names must be used for `NO_PROXY` for the traffic to be routed through the proxy to work properly. The outlined variables need to be set in the `/etc/environment` file.

The following steps outline how to modify this file to add these variables:

1. Run the following command to edit the `/etc/environment/` file in vi. You could also use vim or nano:

   ```
   sudo vi /etc/environment
   ```

2. Update the file to include the following environment variables.

   ```
   HTTPS_PROXY="http://hostname_of_proxy:port"
   HTTP_PROXY="http://hostname_of_proxy:port"
   NO_PROXY=[list of all host names that should not go through the proxy, such
   as: localhost, 127.0.0.1, 0.0.0.0, ip_address_of_mongo]
   ALL_PROXY="http://hostname_of_proxy:port"
   https_proxy="http://hostname_of_proxy:port"
   http_proxy="http://hostname_of_proxy:port"
   no_proxy=”=[list of all host names that should not go through the proxy,
   such as: localhost, 127.0.0.1, 0.0.0.0, ip_address_of_mongo]
   all_proxy="http://hostname_of_proxy:port"
   ```

3. Save the file. Once you install CAS Manager you can configure it to use the proxy configuration. From this new terminal, proceed with the installation steps. The proxy configuration will be implemented when CAS Manager is installed.

Cloud Access Software Registration Code

Once you have a Cloud Access Software subscription Teradici will email a registration code to you. To contact sales and enquire about attaining a Cloud Access Software subscription, see Contact Sales.

Installing CAS Manager

The following section outlines how to install CAS Manager with the default database and secret storage. These steps should be performed on the target machine by connecting via SSH or console.

⚠ **System Requirements and Prerequisite Steps**

Before installing CAS Manager please ensure you have read through the system requirements, and configured the necessary prerequisites outlined above. Failure to do this will result in an unsuccessful installation of CAS Manager.

1. Add CAS Manager Repository

The virtual machine you are adding the repo to must have access to the internet. If it doesn't, you will be unable to download and install the required files.

To access the scripts and to configure and add the RHEL and Rocky Linux repository, select the **Downloads and scripts** option here from the CAS Manager support site.

Run the following command to confirm `teradici-cas-manager` repos were added into dnf repo.

```
dnf repolist --enabled teradici-cas-manager*
```

The output from this command should list the repo id, names as outlined in the example below:

```
repo id                                             repo name
teradici-cas-manager-beta                           teradici-cas-manager-beta
teradici-cas-manager-beta-noarch                    teradici-cas-manager-beta-
noarch
teradici-cas-manager-beta-source                    teradici-cas-manager-beta-
source
```

2. SELinux Configuration

SELinux policies are required for persistent storage and container logging on CAS Manager. If SELinux policies are not found, data stored in CAS Manager will be lost when the CAS Manager Machine is shut down.

Once configured, and the installation has verified SELinux, all CAS Manager related data will persist when the target machine hosting CAS Manager is re-booted. To check if SELinux is already installed on your system, run the following command:

```
sudo dnf list installed | grep cas-manager-selinux
```

The output from this command will notify if you if `selinux` is already running on your system. If it is not then you need to run the following commands to install the SELinux policies:

1. Run the following command to install the SELinux policies and set the basic framework for persistent database and Vault:

   ```
   sudo dnf install -y selinux-policy-base container-selinux
   ```

2. Run the following command to install a specific version of SELinux that has been tested for K3s:

   ```
   sudo dnf install -y https://github.com/k3s-io/k3s-selinux/releases/download/
   v1.1.stable.1/k3s-selinux-1.1-1.el8.noarch.rpm
   ```

3. Run the following command to install SELinux from the CAS Manager repo:

   ```
   sudo dnf install -y cas-manager-selinux
   ```

3. Install CAS Manager

Run the following command to install CAS Manager:

```
sudo dnf install -y cas-manager
```

The installer will install CAS Manager, as well as all external components required.

These external components are:

- k3s
- MongoDB (data store)
- Vault (secret store from HashiCorp)
- A self-signed SSL certificate for HTTPS access

Vault Data Encryption

The Vault data that is installed as part of the CAS Manager installation, is installed on the CAS Manager virtual machine, and is encrypted at rest. It is recommended that you take appropriate measures to secure access to the filesystem. For information on this, see the Filesystem Storage Backend section of the HashiCorp Vault guide.

The installation process takes 5-10 minutes to complete, depending on your network connection speed and other environment variables. During this process, CAS Manager is running a health check every 15 seconds to confirm that all required services are deployed and running successfully before reporting that the installation is complete.

Once the installation has been successful you should see a message stating **CAS Manager installation complete**. The IP address of your CAS Manager instance will also be displayed. The CAS Manager version that has been installed will also be displayed.

If the installation appears unhealthy, you should generate a support bundle and send this to Teradici for investigation. For more information on generating a support bundle, see Support Bundle. For more information on monitoring and assessing the health status of CAS Manager, see Health Status.

Generated Credentials

The installer will automatically generate a password. This password is important as it will be required when accessing the Admin Console. This password can be found in the *temp-creds.txt* file which is located at `/opt/teradici/casm/temp-creds.txt`. This location will be displayed in the CLI window once the the installation has been successful, as seen in the image above.

Generated Self-Signed Certificates

The installer will automatically generate several certificates to ensure that internal communication within the CAS Manager and communication to the CAS Manager itself are done over encrypted TLS connections. These certificates will be automatically generated as needed when CAS Manager is initially installed or when upgrades are done. If for whatever reason you do not wish to upgrade, certificates will need to be periodically renewed, see TLS Certificates for steps on how to do this.

4. Configure CAS Manager to use Proxy

The following section outlines the steps involved in enabling the proxy configuration with CAS Manager:

1. If the proxy environment variables were not set before installing CAS Manager, please see the Proxy Configuration Variables section above for the steps involved in setting these variables. If you already have these variables set, continue to step 2.
2. Establish a new ssh/shell session.
3. Configure CAS Manager to use the proxy configuration by running the following command:

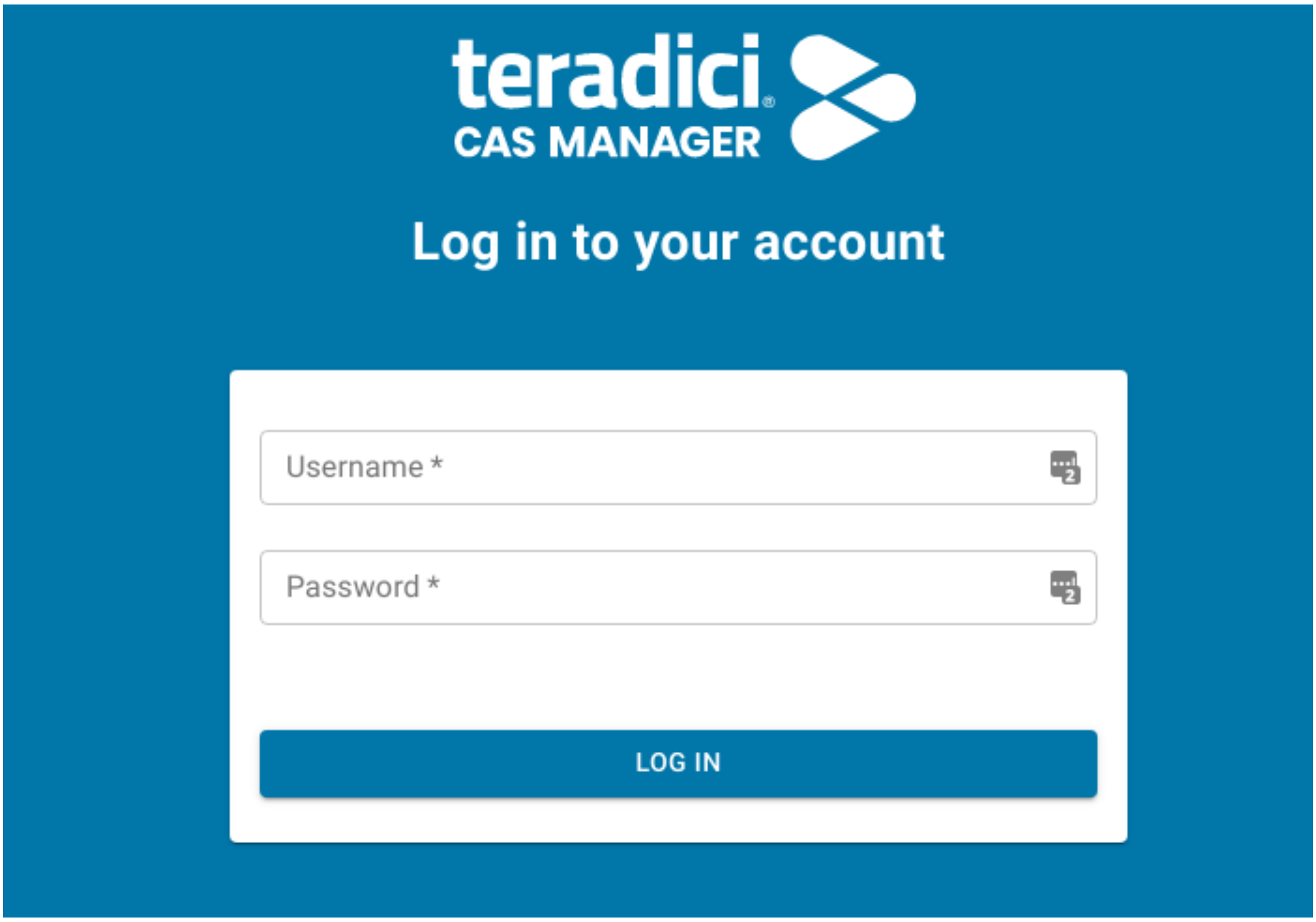
```
sudo /usr/local/bin/cas-manager configure --enable-proxy
```

5. Access the Admin Console

The following section outlines how to access and unlock the CAS Admin Console.

1. Open a web browser and go to https://{public-or-private-ip-address-of-cas-manager}. This is the external IP address of the target machine that CAS Manager has been installed on. You

will be presented with the CAS Manager login page.



2. Use the following credentials to begin setting up the admin user:

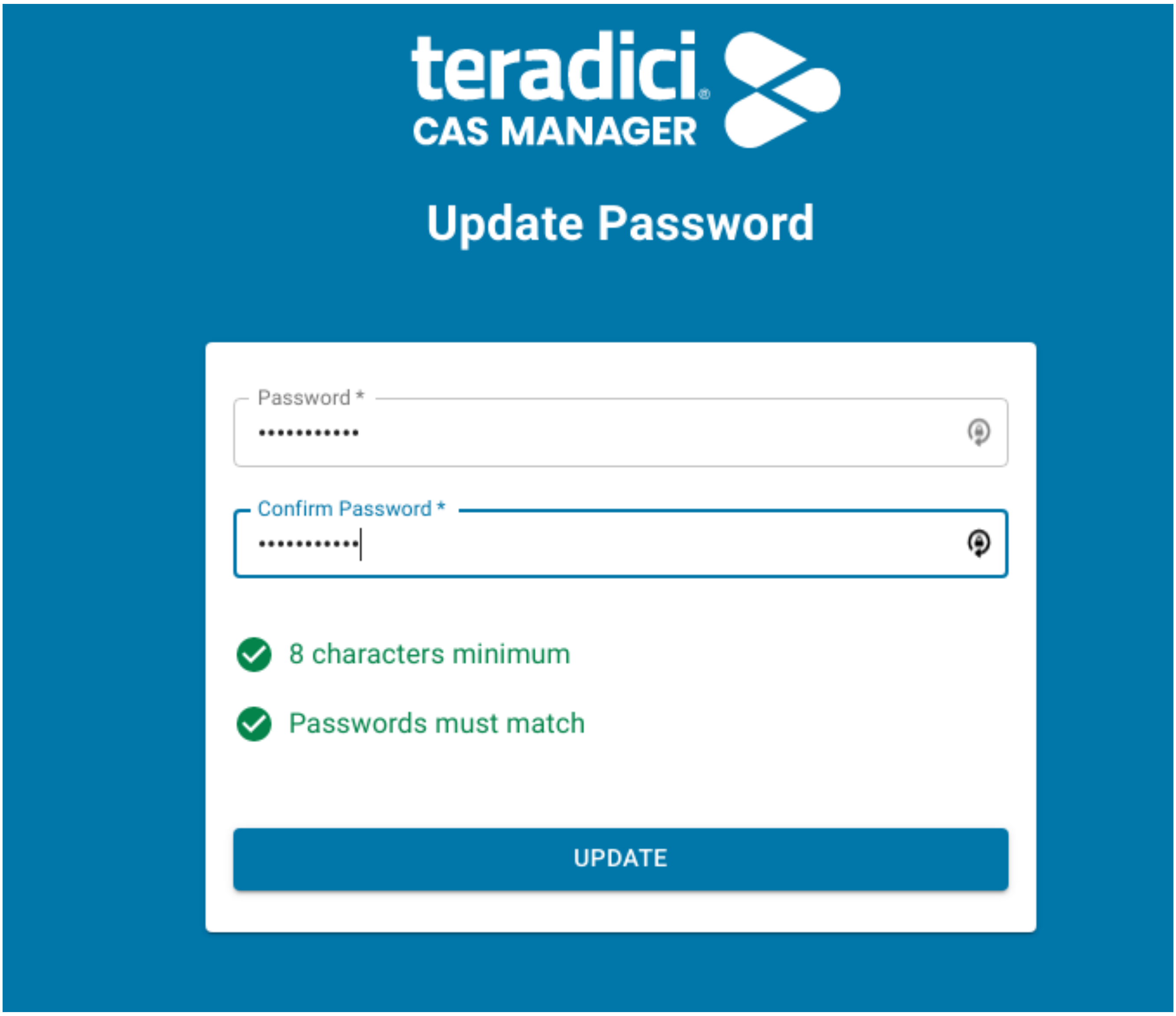
username: adminUser

password: The password generated by the installer.The initial password can be found at */opt/teradici/casm/temp-creds.txt*. You can run the following command to view the password:

```
sudo cat /opt/teradici/casm/temp-creds.txt
```

3. Upon successful login, you will be required to immediately change this password. The new password will be stored in the Vault. Do not change the configuration to connect to a different

Vault after resetting the password.



Update Password

Password *
••••••••••••

Confirm Password *
••••••••••••

✓ 8 characters minimum

✓ Passwords must match

After updating the password you will be able to use CAS Manager as the **adminUser** user.

To unlock the Admin Console enter your Cloud Access Software registration code into the Unlock dialog that appears when you first log-in. CAS Manager will verify the registration code and then create a new deployment on your behalf. For further information on using the Admin Console, see Admin Console.

6. CAS Manager dnf Repo Management

By default, CAS Manager will install any updates that are available, when you update all managed packages with the following command:

```
dnf upgrade
```

or

```
dnf update
```

This system wide update will include any new CAS Manager version updates. If you do not want this system wide update, the CAS Manager repo(s) should be disabled once installation is complete. The following section outlines how to lock the CAS Manager in the dnf repo.

Locking CAS Manager version in the dnf repo

The following command will lock the CAS Manager version in the dnf repo:

```
sudo dnf config-manager --set-disabled teradici-cas-manager*
```

You can confirm the settings by running the following command:

```
dnf repolist teradici-cas-manager*
```

The output from this command should list the repo id, names and their status, as outlined in the example below:

```
repo id                                         repo
name                                       status
teradici-cas-manager                            teradici-cas-
manager                            disabled
teradici-cas-manager-noarch                     teradici-cas-manager-
noarch                     disabled
teradici-cas-manager-source                     teradici-cas-manager-
source                     disabled
```

Installing the Connector

Once you have installed the CAS Manager you can install Connector(s) by following the instructions outlined in the Installing the Connector section.

Installing CAS Manager - External Configuration

This section outlines how to install CAS Manager and to configure an external database and secret storage. If you have already installed CAS Manager with the default configuration you can skip this section.

With this configuration CAS Manager supports high availability and scaling beyond a single virtual machine.

⚠ **Installation Time**

Installing and configuring CAS Manager to run with an external database and secret storage should take roughly **2 hours** to complete. It should take a further **1 hour** to install the Connector.

⚠ **Data Migration**

CAS Manager does not do any data migration when configuring your database and secret storage application. Any data stored when CAS Manager is used with the default database and secret storage configuration, will not be transferred if the same CAS Manager instance is re-configured to run with an external database and secret storage.

✎ **Firewall and Load Balancing Considerations**

For information on firewall and load balancing ports connected to CAS Manager and Connector, see Firewall and Load Balancing Considerations

Preparing the CAS Manager Virtual Machine

The following section outlines how to prepare the system requirements, firewall configurations and proxy configurations on the CAS Manager virtual machine:

System Requirements

You need to prepare a virtual machine that has the following requirements:

- Operating System: RHEL 8 and Rocky Linux 8.
- Minimum 8 GB RAM
- 4 CPU
- 60 GB Storage: If you are using LVM and `/var` is mounted on a separate volume, that volume must have 30GB or more in order for the installation to succeed and for CAS Manager to function properly.
- Active Directory permissions set to **List contents** and **Read all properties**. If you do not set these permissions you will be unable to connect to specific remote workstations.

Firewall Configuration

You must ensure your firewall is established and configured properly. Ensure port 443 is enabled in the firewall rules for the VM that CAS Manager is running on.

Configure the firewall that the virtual network CAS Manager is running by following the commands below:

1. Login to the CAS Manager VM by ssh from a bash shell as ***root***.
2. Check and confirm if firewalld is active by running the following command:

   ```
   sudo systemctl status firewalld
   ```

3. If `firewalld` is active, follow the steps outlined below for firewall configuration. If `firewalld` is inactive, and your organization does not require firewall on the CAS Manager VM, then skip the firewall configuration steps below and proceed to the remaining steps.
4. Run the following commands to configure the firewall:

```
sudo firewall-cmd --permanent --add-port=6443/tcp # virtual network flannel

sudo firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16 # This subnet is for the pods
```

```
sudo firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16 # This
subnet is for the services

sudo firewall-cmd --reload
```

Proxy Configuration Variables

If HTTP/HTTPS proxy is used, then `HTTP_PROXY` , `HTTPS_PROXY` and `NO_PROXY` must be set. For `NO_PROXY` , specific IP addresses or domain names of service that are internal must be added. IP address ranges like "10.0.0.0/8" will not work; exact IP addresses or domain names must be used for `NO_PROXY` for the traffic to be routed through the proxy to work properly. The outlined variables need to be set in the `/etc/environment` file.

The following steps outline how to modify this file to add these variables:

1. Run the following command to edit the `/etc/environment/` file in vi. You could also use vim or nano:

   ```
   sudo vi /etc/environment
   ```

2. Update the file to include the following environment variables.

   ```
   HTTPS_PROXY="http://hostname_of_proxy:port"
   HTTP_PROXY="http://hostname_of_proxy:port"
   NO_PROXY=[list of all host names that should not go through the proxy, such
   as: localhost, 127.0.0.1, 0.0.0.0, ip_address_of_mongo]
   ALL_PROXY="http://hostname_of_proxy:port"
   https_proxy="http://hostname_of_proxy:port"
   http_proxy="http://hostname_of_proxy:port"
   no_proxy=”=[list of all host names that should not go through the proxy,
   such as: localhost, 127.0.0.1, 0.0.0.0, ip_address_of_mongo]
   all_proxy="http://hostname_of_proxy:port"
   ```

3. Save the file. Once you install CAS Manager you can configure it to use the proxy configuration. From this new terminal, proceed with the installation steps. The proxy configuration will be implemented when CAS Manager is installed.

Cloud Access Software Registration Code

Once you have a Cloud Access Software subscription Teradici will email a registration code to you. To contact sales and enquire about attaining a Cloud Access Software subscription, see Contact Sales.

By default, CAS Manager will install a database and secret storage on the same virtual machine. If you plan to use an external database and secret storage, which Teradici recommends for scaling, continue with the steps outlined below to prepare the external database and secret store.

Preparing an External Database and Secret Storage

The following sections outline how to prepare a secret storage application and MongoDB that can be configured to work with CAS Manager.

Verified Versions

The table below outlines the versions of MongoDB and Vault that are verified with CAS Manager:

CAS Manager Version	Vault Version	MongoDB Version
21.03	1.4.2	4.0.8
21.07	1.7.1	4.2.14
21.10	1.7.1	4.2.14

Preparing a Secret Storage Application

It is possible to use either Hashicorp Vault or Azure Key Vault, depending on your environment and needs, for secret and key encryption and storage with CAS Manager. Once you have successfully installed CAS Manager you will need to configure CAS Manager to use the defined secret store. Please be aware that you can only configure one secret storage option with CAS Manager.

The sections below outline the prerequisite steps required to prepare these secret stores:

- Preparing Azure Key Vault
- Preparing Hashicorp Vault

You can't configure the secret storage application to work with CAS Manager until you have successfully installed CAS Manager. Please complete the installation and then perform the required configurations.

Preparing an External Database

The following section provides guidelines and best practices involved when preparing and deploying a production MongoDB solution with CAS Manager.

Reference Instructions for MongoDB and Vault Configuration

For detailed deployment instructions on installing and configuring MongoDB and Vault in a single virtual machine to be used by CAS Manager, see the following KB article. This KB article outlines in detail how to install and configure an instance of MongoDB and an instance of Vault on the same virtual machine. This KB article should be used in conjunction with the installation steps outlined in this section.

⚠ **Reference Steps Only**

All configuration steps outlined should be used as a reference only. For specific details, visit the vendor's official documentation and knowledge base. For information on the main reference list for MongoDB, see https://docs.mongodb.com/manual/administration/.

Guidelines and Best Practices

The following are some of the guidelines and best practices that Teradici encourages when deploying a MongoDB to work with CAS Manager:

- Ensure the machine is deployed in a secure subnet with no public facing access.
- Ensure that the host firewalls are leveraged to control inbound and outbound traffic.
- MongoDB only needs to be accessible to the CAS Manager and to administrators so it is better to be overly restrictive when granting access, and follow the rules of granting least privilege access.

- CAS Manager cannot connect to an external MongoDB from behind a proxy.
- Remote desktop or SSH access to the system should be disallowed altogether if possible - realistically this is highly unlikely - or heavily restricted to essential users only, with a security-conscious configuration (e.g. add certificates for RDP, use passphrase-protected SSH keys and disallow password based authentication, change default SSH port, etc).
- Keep the host OS patched and up to date to ensure security fixes are deployed.
- It is best to use the latest stable version of MongoDB to ensure there are as few vulnerabilities, bugs, and issues as possible.
- It is best to maintain a regular update cadence for both MongoDB and the host machine in order to maintain latest security fixes.
- It is best to run MongoDB on a Long Term Support variant of Linux (ex, RHEL x86_64 or Ubuntu x86_64) VM.
- In order to maintain data integrity, it is best to run Mongo with Journaling enabled (enabled by default) in a geographically distributed replica set.
- Regular backups are also important to ensure CAS Manager can be restored in case of a crash. To keep MongoDB secure, it is important to create the appropriate admin accounts for granting access and ensuring that all communication is done over a secured TLS connect. Details for creating an appropriate service account can be found in the official MongoDB documentaton, as well as:
    - Details for enabling data encryption at rest.
    - How to enable TLS on the MongoDB server.
    - Additional tips for hardening the system.

Installing CAS Manager

The following section outlines how to install CAS Manager. These steps should be performed on the target machine by connecting via SSH or console.

> ⚠ **System Requirements and Prerequisite Steps**
>
> Before installing CAS Manager please ensure you have read through the system requirements, and configured the necessary prerequisites outlined above. Failure to do this will result in an unsuccessful installation of CAS Manager.

1. Add CAS Manager Repository

The virtual machine you are adding the repo to must have access to the internet. If it doesn't, you will be unable to download and install the required files.

To access the scripts and to configure and add the RHEL and Rocky Linux repository, select the **Downloads and scripts** option here from the CAS Manager support site.

Run the following command to confirm `teradici-cas-manager` repos were added into dnf repo.

```
dnf repolist --enabled teradici-cas-manager*
```

The output from this command should list the repo id, names as outlined in the example below:

```
repo id                                         repo name
teradici-cas-manager-beta                       teradici-cas-manager-beta
teradici-cas-manager-beta-noarch                teradici-cas-manager-beta-
noarch
teradici-cas-manager-beta-source                teradici-cas-manager-beta-
source
```

2. SELinux Configuration

SELinux policies are required for persistent storage and container logging on CAS Manager. If SELinux policies are not found, data stored in CAS Manager will be lost when the CAS Manager Machine is shut down.

Once configured, and the installation has verified SELinux, all CAS Manager related data will persist when the target machine hosting CAS Manager is re-booted. To check if SELinux is already installed on your system, run the following command:

```
sudo dnf list installed | grep cas-manager-selinux
```

The output from this command will notify if you if `selinux` is already running on your system. If it is not then you need to run the following commands to install the SELinux policies:

1. Run the following command to install the SELinux policies and set the basic framework for persistent database and Vault:

```
sudo dnf install -y selinux-policy-base container-selinux
```

2. Run the following command to install a specific version of SELinux that has been tested for K3s:

   ```
   sudo dnf install -y https://github.com/k3s-io/k3s-selinux/releases/download/
   v1.1.stable.1/k3s-selinux-1.1-1.el8.noarch.rpm
   ```

3. Run the following command to install SELinux from the CAS Manager repo:

   ```
   sudo dnf install -y cas-manager-selinux
   ```

3. Install CAS Manager

Run the following command to install CAS Manager:

```
sudo dnf install -y cas-manager
```

The installer will install CAS Manager, as well as all external components required.

These external components are:

- k3s
- A self-signed SSL certificate for HTTPS access

The installation process takes 5-10 minutes to complete, depending on your network connection speed and other environment variables. During this process, CAS Manager is running a health check every 15 seconds to confirm that all required services are deployed and running successfully before reporting that the installation is complete.

Once the installation has been successful you should see a message stating **CAS Manager installation complete**. The IP address of your CAS Manager instance will also be displayed. The CAS Manager version that has been installed will also be displayed.

If the installation appears unhealthy, you should generate a support bundle and send this to Teradici for investigation. For more information on generating a support bundle, see Support

Bundle. For more information on monitoring and assessing the health status of CAS Manager, see Health Status.

Generated Credentials

The installer will automatically generate a password. This password is important as it will be required when accessing the Admin Console. This password can be found in the *temp-creds.txt* file which is located at `/opt/teradici/casm/temp-creds.txt`. This location will be displayed in the CLI window once the the installation has been successful, as seen in the image above.

Generated Self-Signed Certificates

The installer will automatically generate several certificates to ensure that internal communication within the CAS Manager and communication to the CAS Manager itself are done over encrypted TLS connections. These certificates will be automatically generated as needed when CAS Manager is initially installed or when upgrades are done. If for whatever reason you do not wish to upgrade, certificates will need to be periodically renewed, see TLS Certificates for steps on how to do this.

4. Configure CAS Manager to use Proxy

The following section outlines the steps involved in enabling the proxy configuration with CAS Manager:

1. If the proxy environment variables were not set before installing CAS Manager, please see the Proxy Configuration Variables section above for the steps involved in setting these variables. If you already have these variables set, continue to step 2.
2. Establish a new ssh/shell session.
3. Configure CAS Manager to use the proxy configuration by running the following command:

```
sudo /usr/local/bin/cas-manager configure --enable-proxy
```

5. Configure CAS Manager to use a Secret Storage Application

Once you have successfully installed CAS Manager you must configure it to use the secret store you prepared in the prerequisite steps prior to installing CAS Manager. You need to have prepared the selected secret storage application before installing CAS Manager, as outlined in the Preparing a Secret Storage Application section above. For information on how to configure CAS Manager to work with these secret stores, see the following sections based on what type of secret storage you prepared:

- Configuring CAS Manager with Azure Key Vault
- Configuring CAS Manager with Hashicorp Vault

6. Configure CAS Manager to use MongoDB

Once you have successfully installed CAS Manager you must configure it to use the external MongoDB you prepared in the prerequisite steps prior to installing CAS Manager.

The following section outlines how to configure CAS Manager to use MongoDB:

1. SSH to your target machine where you installed CAS Manager.
2. Create a file that contains the following data:

```
{
  "db-connection-string": "mongodb://<username>:<password>@<address>/
<db_name>",
  "db-enable-tls": true,
  "db-skip-verify-cert": false
}
```

URL Encoding

If the username or password contain any of the following special characters: /, ?, #, [], @, %, those characters must be converted using URL encoding in the MongoDB connection string. For example, if you defined user 'casmuser' with password 'Password%' in MongoDB, then in CAS Manager the `db-connection-string` for MongoDB would look like this:

```
mongodb://casmuser:Password%25@ip_of_mongodb:27017/name_of_mongodb
```

If you require more characters to be encoded, or want to test encoding or decoding your data, see https://www.urlencoder.org/.

3. Replace the following place holders with your own values:
   - username: Username of the MongoDB user that CAS Manager will authenticate MongoDB requests.

- password: Password for the MongoDB user referenced in "username".
- address: Address to the MongoDB server.
- db_name: Name of the MongoDB database that CAS Manager will use. Note that if no db name is specified, the db named "test" will be used.

4. Run the following command to configure CAS Manager to use MongoDB:

```
sudo /usr/local/bin/cas-manager configure --config-file path-to-your-config-
file
```

⚠ "MongoDB Database Name

If no database name is provided as part of the connection string, a default name "test" will be used instead, for example:

db-connection-string:"mongodb://user:pass@mongo:27017/ will result in the creation of a database with the name "test".

If you provided a database name then that will be used, for example:

db-connection-string:"mongodb://user:pass@mongo:27017/casm_db will result in "casm_db" being used as the name.

After running this command, CAS Manager will validate the configuration by attempting to query the MongoDB server. If the request is successful, then CAS Manager will be configured to use this MongoDB. The configure command should only take a few minutes to complete.

Here's an example of creating a user for the CAS Manager Database "casm""

```
use casm_db
db.createUser(
  {
    user: "casmanager",
    pwd: passwordPrompt(), // or cleartext password
    roles: [ {db: "casm_db", role:"readWrite"} ], // user only needs readWrite
Access to casm DB,
    authenticationRestrictions: [
        {
          clientSource: [
            "<CASM-IP>", // IP address of the CASM Host
            "10.42.0.0/24" // Subnet for the CASM pods
          ],
          serverAddress: ["<MongoDB IP>"] // IP for the MongoDB server
        }
      ],
    }
)
```

The connection string for this user would be:

```
mongodb://casmanager:<password>@<MongoDB IP>/casm_db
```

> **Configuration Templates**
>
> Teradici provides configuration template files and parameters that can be generated and used when configuring your MongoDB, see Configuration Templates.

6.1 Connecting a MongoDB with Self-Signed TLS Certificates

CAS Manager allows for the option to provide a database connection string, a flag to enable/ disable TLS, a flag to enabled/disable TLS cert validation, and also provide a custom Certificate Authority certificate for the MongoDB Server certificate. This is only recommended during proof-of-concept testing. In this mode, TLS must be enabled and certificate validation must be carried out. A server certificate signed by a public Certificate Authority is also highly recommended.

> **Tested on CentOS Only**
>
> The following steps have been tested on CentOS. These steps may not work, or work differently, on different systems.

The following steps outline how to connect a MongoDB that uses self-signed TLS certificates:

1. SSH to your target machine where you installed CAS Manager.
2. Create a file that contains the following data:

```
{
    "db-connection-string": "mongodb://<username>:<password>@<address>/
<db_name>",
    "db-enable-tls": true,
    "db-ca-cert-file": "/path/to/mongo/TLS/custom/certificate/authority",
    "db-skip-verify-cert": false
}
```

3. Replace the following place holders with your own values:
    - "db-connection-string": Follow the same guidelines as mentioned above.
    - "db-ca-cert-file": Path to MongoDB's custom Certificate Authority's public certificate, in PEM format, if one is used. This is only required to validate self-signed certificates or certificates signed by a non-public Certificate Authority.
4. Run the following command to configure CAS Manager to use MongoDB:

    ```
    sudo /usr/local/bin/cas-manager configure --config-file path-to-your-config-file
    ```

5. If you want to skip certificate verification, include `"db-skip-verify-cert": true` in your configuration file. Please note that this is not secure and is not recommended for production use cases:

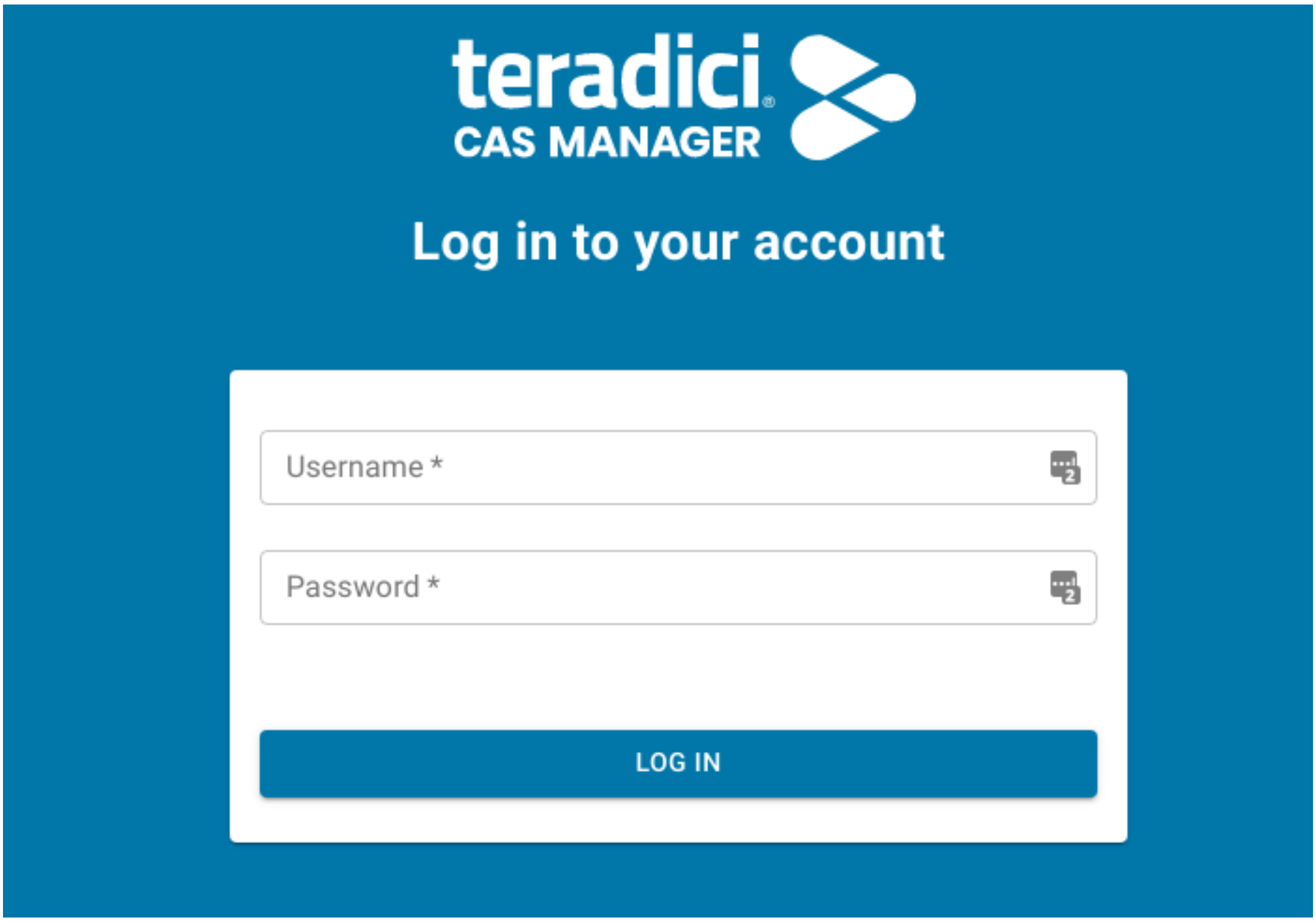
    ```
    {
        "db-connection-string": "mongodb://<username>:<password>@<address>/<db_name>",
        "db-enable-tls": true,
        "db-ca-cert-file": "/path/to/mongo/TLS/custom/certificate/authority",
        "db-skip-verify-cert": true
    }
    ```

7. Accessing the Admin Console

The following section outlines how to access and unlock the CAS Admin Console.

1. Open a web browser and go to https://{public-or-private-ip-address-of-cas-manager}. This is the external IP address of the target machine that CAS Manager has been installed on. You

will be presented with the CAS Manager login page.



2. Use the following credentials to begin setting up the admin user:

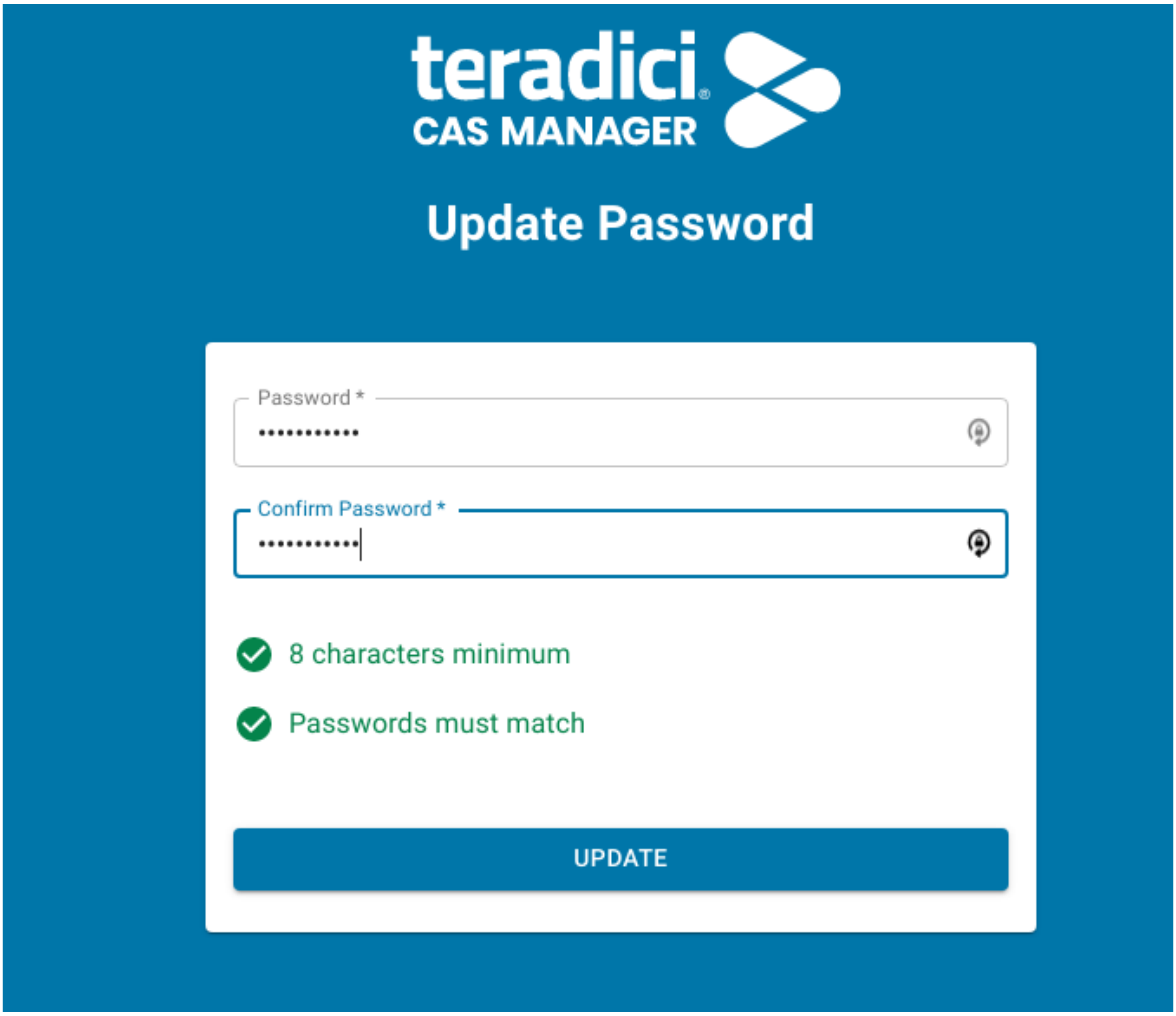
username: adminUser

password: The password generated by the installer.The initial password can be found at */opt/teradici/casm/temp-creds.txt*. You can run the following command to view the password:

```
sudo cat /opt/teradici/casm/temp-creds.txt
```

3. Upon successful login, you will be required to immediately change this password. The new password will be stored in the Vault. Do not change the configuration to connect to a different

Vault after resetting the password.



✓ 8 characters minimum

✓ Passwords must match

After updating the password you will be able to use CAS Manager as the **adminUser** user.

To unlock the Admin Console enter your Cloud Access Software registration code into the Unlock dialog that appears when you first log-in. CAS Manager will verify the registration code and then create a new deployment on your behalf. For further information on using the Admin Console, see Admin Console.

8. CAS Manager dnf Repo Management

By default, CAS Manager will install any updates that are available, when you update all managed packages with the following command:

```
dnf upgrade
```

or

```
dnf update
```

This system wide update will include any new CAS Manager version updates. If you do not want this system wide update, the CAS Manager repo(s) should be disabled once installation is complete. The following section outlines how to lock the CAS Manager in the dnf repo.

Locking CAS Manager version in the dnf repo

The following command will lock the CAS Manager version in the dnf repo:

```
sudo dnf config-manager --set-disabled teradici-cas-manager*
```

You can confirm the settings by running the following command:

```
dnf repolist teradici-cas-manager*
```

The output from this command should list the repo id, names and their status, as outlined in the example below:

```
repo id                                         repo
name                                       status
teradici-cas-manager                            teradici-cas-
manager                            disabled
teradici-cas-manager-noarch                     teradici-cas-manager-
noarch                     disabled
teradici-cas-manager-source                     teradici-cas-manager-
source                     disabled
```

Installing the Connector

Once you have installed the CAS Manager you can install Connector(s) by following the instructions outlined in the Installing the Connector section.

Installing CAS Manager - Darksite Installation

In cases where CAS Manager needs to be installed in a **darksite** (ie, an environment where there is no internet access, also known as **airgap** or **offline** environment) you will need to download the darksite dependency files, transfer them to the target darksite machine, and then run the darksite installation script.

> **Legacy Scripts and Tarball Files**
>
> Before you install a new darksite version of CAS Manager, you must remove all legacy CAS Manager scripts and tarball files.

Prerequisite Steps

> **RHEL/Rocky Linux OS version**
>
> For CAS Manager Darksite installation, the minimum RHEL/Rocky Linux operating system version is 8.5.

Follow the prerequisite steps outlined here: CAS Manager System Requirements to prepare your target machine. It is important to read and address all the prerequisites outlined. Once you have completed these steps and prepared the target machine, return to this page and continue with the installation.

Download and Transfer Dependencies

Once you have addressed all the prerequisite steps, you need to download and transfer the dependencies files.

The combined dependencies will take up 1.8 GB approximately. You must ensure you have a media device that can transfer all required components to the target darksite virtual machine;

these components can be removed from the target virtual machine once installation is complete. The required packages and dependencies to complete a darksite installation are:

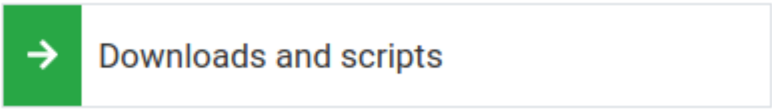
- Darksite installation bash script: A bash script that will set up and install required components as well as the CAS Manager.
- RedHat Package Manager (RPM) dependencies: A yum repo that contains all RPM dependencies required to install CAS Manager. Only the dependencies that are missing will be installed but this package contains all system requirements.
- CAS Manager container images: A `.tar.gz` file containing all CAS Manager container images required for running the CAS Manager to be loaded onto the system.

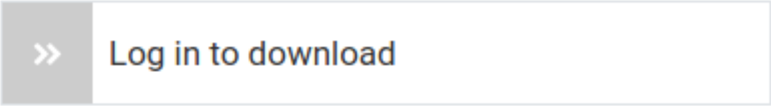
All of the packages and dependencies are bundled in a tar archive file.

1. Downloading the File

Once you have access to a virtual machine with internet access, download the darksite installation package directly as a `tar.gz` file from teradici.com CAS Manager product release and follow the steps below:

- Click the **Downloads and scripts** button.



> **Note: An account is required**
>
> If you are not logged in, you will see a log in prompt instead:
>
> 
>
>
> You can create an account when you click this button if you do not already have one.

- Read and accept the Teradici End User License Agreement.
- To download the darksite package, click **Download linux** Under **Darksite packages**.

2. Transferring the File

Once you have downloaded the `.tar.gz` file, transfer it to the target darksite machine. For system requirements of the target virtual machine, see here.

To transfer the file you can copy it onto a media device, such as a USB drive or a DVD, and then connect that device to the target darksit machine. You can also connect the target darksite machine to another machine via SSH or FTP, and complete a network file transfer. This method may not be viable for some darksite networks.

3. Extracting the File

Once the `.tar.gz` file has been transferred to the target darksite machine, extract the downloaded file by running the following command:

```
sudo tar xzvf cas-manager-offline_Linux.tar.gz
```

That creates a new folder called `cas-manager-offline_VERSION_linux` .

Change drive on the command prompt into the new folder. You can see two files, one is the dependency tarball file and the other is the install bash script file.

4. Install CAS Manager Darksite

Checking and Removing CAS Manager Repository Files

CAS Manager Darksite doesn't need to download images from online repository, this usually works as expected on a clean VM without additional step here. However, if there is repository set by the flags or repository file, the installer will try to download from what the target is set to. To avoid this you must make sure the relevant repository files are removed from the VM.

- To check if the repository present in the system, run `sudo ls -la /etc/yum.repos.d`
- If you have any repository name that start from ***teradici-cas-manager***, you need to remove them by running `sudo rm /etc/yum.repos.d/NAME_OF_THE_REPO` .

Run the following command for the installation:

```
sudo ./install.sh
```

This installs the CAS Manager and all the other dependencies.

The command above will replace steps 1-3 in either the External Configuration or Default Configuration installation steps. Please follow the steps after this with the configuration you wish to use.

Upgrading CAS Manager

When upgrading CAS Manager there are two options available.

1. **In-place upgrade within the maintenance window:** You can run an in-place upgrade through dnf for CAS Manager. Depending on the configuration you implemented, this will mean a period of downtime which can range from a few seconds to a few minutes.
2. **Zero downtime upgrade via a new VM**. The second option involves installing CAS Manager on a new virtual machine, and configuring it to connect to the same external database and secret storage. If done correctly this can result in zero downtime.

The steps involved in both options are outlined below.

Which Option Should I Choose?

The upgrade option you choose depends on the amount of downtime you are willing to experience and how your CAS Manager instance has been deployed and setup. The following are some use cases that outline which option to use:

- If you have a single CAS Manager server connecting to external database and secret storage, it is recommended to install CAS Manager on a new virtual machine. If you don't have a new virtual machine then run the in-place upgrade on the existing virtual machine with the understanding that there will be some downtime during this upgrade.
- If you have multiple CAS Manager servers connecting to the same external database and secret storage, it is recommended to run an in-place upgrade on each CAS Manager server, one at a time. There should not be any downtime as long as one CAS Manager server is up and running.
- If the database and secret storage is on the same virtual machine as CAS Manager, you must run an in-place upgrade. This is to ensure that the data persists after the upgrade has been completed. There will be some downtime during this upgrade.

Running an In-Place Upgrade

> **CAS Manager Downtime**
>
> The CAS Manager virtual machine that is undergoing an in-place upgrade will not be available during the upgrade. This can take anywhere from a few seconds to a few minutes, depending on the number of services that need to be upgraded and the speed of download when retrieving new versions from the repo. If this is the only CAS Manager server you have, the new connections will not be established until the upgrade is completed successfully.

1. Update CAS Manager dnf Repository

There are two cases you need to update CAS Manager dnf repository:

- You want to upgrade to a CAS Manager version that is on different repo, such as upgrading from a GA version to a beta version
- You have installed version 21.03 when EL7 was specified for download.

If your cases are not these, please skip this section and go to 2. Unlocking CAS Manager Version.

The following section outlines how to update the CAS Manager dnf repository as part of running an in-place upgrade.

1. SSH to the CAS Manager virtual machine.
2. Remove the existing dnf repo for previous CAS Manager with the following command.

   ```
   sudo rm /etc/dnf.repos.d/teradici-cas-manager.repo
   ```

3. To access the scripts and to configure and add the RHEL and Rocky Linux repository, select the **Downloads and scripts** option here from the CAS Manager support site.

Run the following command to confirm `teradici-cas-manager` repos were added into dnf repo.

```
dnf repolist --enabled teradici-cas-manager*
```

The output from this command should list the repo id, names as outlined in the example below:

```
repo id                                             repo name
teradici-cas-manager-beta                           teradici-cas-manager-beta
teradici-cas-manager-beta-noarch                    teradici-cas-manager-beta-
noarch
teradici-cas-manager-beta-source                    teradici-cas-manager-beta-
source
```

2. Unlocking CAS Manager Version

If you have skipped above step Update CAS Manager dnf Repository , make sure the existing CAS Manager repo is enabled. Follow the command below to unlock the CAS Manager repo if it was disabled to lock the version before.

The following command will unlock the CAS Manager version in the dnf repo:

```
sudo dnf config-manager --set-enable teradici-cas-manager*
```

You can confirm the settings with the following command:

```
dnf repolist teradici-cas-manager*
```

The output from this command should list the repo id, names and their status, as outlined in the example below:

```
repo id                                         repo
name                                        status
teradici-cas-manager                            teradici-cas-
manager                             enabled
teradici-cas-manager-noarch                     teradici-cas-manager-
noarch                      enabled
teradici-cas-manager-source                     teradici-cas-manager-
source                      enabled
```

3. Upgrade CAS Manager via dnf

The following steps outline how to perform an in-place upgrade of CAS Manager through dnf:

1. Before running an in-place upgrade you should backup the entire virtual machine by taking a snapshot. This snapshot will help to rollback if anything wrong happens during the upgrade.

2. SSH to the CAS Manager virtual machine.
3. Run the following command:

   ```
   sudo dnf upgrade -y cas-manager-selinux cas-manager
   ```

   If there is no new version available, and you therefore currently have the latest version, you will see a message outlining this when you run this command.
4. Once you have run the upgrade command you need to wait until all CAS Manager services have been updated and are healthy. To manually verify the health status of CAS Manager run:

   ```
   sudo /usr/local/bin/cas-manager diagnose --health
   ```

5. Verify that CAS Manager has been updated by running the following command:

   ```
   dnf list --installed 'cas-manager*'
   ```

This will display the current installed CAS Manager version.

Installing CAS Manager on a new Virtual Machine

The following steps outline how to install CAS Manager on a new virtual machine, and configure it to connect to the same external database and secret storage application:

1. Before performing an upgrade you should backup the database and secret storage application you used when installing CAS Manager. If you intend to install CAS Manager on a new virtual machine, backup the configuration file so that it can be used on the CAS Manager instance.
2. Follow the installation steps outlined here to install a new instance of CAS Manager in a new virtual machine to replace the existing one. You must configure it to be identical to the old CAS Manager instance. It needs to connect to the same MongoDB, and secret storage application, as well as having the same certificate and network configurations.
3. Change your DNS to point to the new CAS Manager instance.

4. Reconfigure your Connector to connect to the new CAS Manager if necessary:
    - If you installed your Connector with `--cam-url=https://Fully-Qualified-Domain-Name-of-CAS-M` :
        - Change your DNS entry for the FQDN to point to the new CAS Manager's static IP.
        - In the Connector virtual machine, flush the DNS cache to use the latest DNS. This ensures that there will be zero downtime as the Connector will be able to connect to the new CAS Manager instance:

          ```
          sudo systemd-resolve --flush-caches
          ```
    - If you installed your Connector with `--cam-url=https://ip-address-of-cas-manager` :
        - Update your Connector to use the latest CAS Manager's IP address.
        - Log into the Connector virtual machine and run the following command to update the CAS Manager IP:

          ```
          sudo usr/sbin/cloud-access-connector update --cam-url https://<New
          CAS-Manager IP>
          ```
5. In the Connector, ping the FQDN to verify it can find the new CAS Manager instance.
6. Remove the old CAS Manager virtual machine.

Removing the CAS Manager Virtual Machine

The following steps outline how to remove the CAS Manager virtual machine:

1. Save the configuration setting files used in the current version of CAS Manager. For example, save them as a ***all-configurations.json*** file. If the current CAS Manager has proxy configured, save all the proxy environment variables also.
2. Run the following commands to remove the CAS Manager.

   ```
   sudo dnf remove -y cas-manager-selinux cas-manager
   sudo rm -rf /opt/teradici     # Remove the cas-manager files
   ```
3. Delete the CAS Manager VM.

Configuration Template Files and Parameters

The following section outlines the configuration template files and parameters available with CAS Manager.

Configuration Template Files

The following section outlines how to generate template files that can be used to fill in various CAS Manager configurations:

1. SSH to your machine where you installed CAS Manager.
2. Run one of the following commands listed in the code block below to generate configuration template files:

```
# Generate the following config templates: mongo-template.json
sudo /usr/local/bin/cas-manager generate --mongo

# Generate the following config templates: tls-template.json
sudo /usr/local/bin/cas-manager generate --tls

# Generate the following config templates: vault-template.json
sudo /usr/local/bin/cas-manager generate --vault

# Generate the following config templates: all-templates.json, mongo-
template.json, tls-template.json, vault-template.json
sudo /usr/local/bin/cas-manager generate --all-templates
```

Configuration Parameters

The following table contains all of the parameters that can be used in a configuration file to configure CAS Manager.

Parameter	Type	Description	Required
vault-type	string	Specifies the type of secret store that CAS Manager should use. Currently, "vault" is the only supported value for this parameter.	Required for updating Vault configuration.
vault-url	string	URL of the Vault server.	Required for updating Vault configuration.
vault-secret-path	string	Vault secret path where secrets are stored.	Required for updating Vault configuration.
vault-token	string	Token used to authenticate requests to the Vault server.	Required for updating Vault configuration.
vault-ca-cert-file	string	Path to the file containing a PEM-formatted CA certificate that will be used to validate the Vault server's certificate.	Required if the Vault server is using self-signed certificates.
vault-skip-verify-cert	boolean	If true, CAS Manager will not validate the Vault server's TLS certificate. This is not secure and is not recommended for production deployments.	Not required. Defaults to "false".
db-connection-string	string	URL of the MongoDB server.	Required for updating MongoDB configuration.
db-enable-tls	boolean	If false, requests to MongoDB will not be encrypted. Setting this parameter to false is not secure and is not recommended for production deployments.	Not required. Defaults to "true".
db-skip-verify-cert	boolean	If true, CAS Manager will not validate the MongoDB server's TLS certificate. This is not secure and is not recommended for production deployments.	Not required. Defaults to "false".

Parameter	Type	Description	Required
db-ca-cert-file	string	Path to the file containing a PEM-formatted CA certificate that will be used to validate the MongoDB server's certificate.	Required if the MongoDB server is using self-signed certificates.
tls-key-file	string	Path to the file containing a PEM-formatted TLS key that will be used by CAS Manager.	Required for updating TLS certificates used by CAS Manager.
tls-cert-file	string	Path to the file containing PEM-formatted TLS certificate that will be used by CAS Manager.	Required for updating TLS certificates used by CAS Manager.
skip-validate-reg-code	boolean	If true, skip validating PCoIP registration code when creating or updating a deployment.	Not required unless CAS Manager is blocking all internet traffic. Defaults to "false".

Configuring Custom TLS Certificates

By default, CAS Manager is deployed using self-signed TLS certificates. Teradici recommends using a custom TLS certificate for using CAS Manager in production. You should renew and maintain these certificates as required. CAS Manager supports X509 certificates in PEM format, the certificate file must only include a single certificate, the CA bundle is not supported. The TLS key must not require Password.

To configure CAS Manager to use custom TLS certificates, or to update CAS Manager to use a new TLS certificate, follow the steps outlined below:

1. Create a file called ***tls-config.json*** with the following contents:

   ```
   {
     "tls-key-file": "<path to a file containing your TLS certificate key>",
     "tls-cert-file": "<path to a file containing your TLS certificate>"
   }
   ```

2. Update the TLS configuration by running the following command:

   ```
   sudo /usr/local/bin/cas-manager configure --config-file tls-config.json
   ```

   This command will update the CAS Manager services automatically, and this will take a few minutes to complete.

Internal TLS Certificates

When you are using internal MongoDB and Vault for data storage, in order to keep CAS Manager's internal communication secure, the installer will also generate a set of self-signed TLS certificates to be used for encrypting internal communication within CAS Manager. By default these certificates will expire 2 years from when they are generated.

In order to ensure that the CAS Manager uptime is not interrupted unexpectedly it is important to ensure that these certificates do not expire. This can be done by:

- Upgrading CAS Manager regularly. These certificates will be regenerated by the installer during the upgrade process if they are close to expiring so upgrading at a regular cadence (eg, once or twice a year) will ensure everything remains operational.
- If you do not want to upgrade CAS Manager and only want to use a version that you have qualified yourself, and that may exceed the TLS certificates expiration time, you can either:
    - Periodically re-deploy the CAS Manager instance you are running so that new certificates are generated regularly.
    - Run the command to re-generate certificates periodically. See Internal Certificate Generation below for steps on how to do this.
- Monitor when the certificates are going to expire and plan to regenerate them beforehand. You can do this by either running the CAS Manager diagnose command or checking the CAS Manager health probe's logs. Run the following command to generate this health check:

  ```
  /usr/local/bin/cas-manager diagnose --health
  ```

  This health check will assess the Mongo Database and Vault connections. A warning message will be logged if the certificates are close to expiring and an error will be logged if they have expired. For example,

  ```
  ...
  INFO .. Connections:
  INFO .... MongoDB=Healthy
  WARN ...... Mongo Certificate Valid From=2021-08-17 19:35:42 Mongo
  Certificate Valid Until=2021-09-18 19:35:42
  INFO .... vault=Healthy
  WARN ...... Vault Certificate Valid From=2021-08-17 19:35:42 Vault
  Certificate Valid Until=2021-09-18 19:35:42
  ...
  ```

  In order to check the logs for the CAS Manager health probe, run

```
/usr/local/bin/kubectl get jobs -o jsonpath='{.items[?
(@.spec.template.metadata.labels.name=="casm-health-probe")].metadata.name}'
--sort-by=.metadata.creationTimestamp | rev | cut -d' ' -f 1 | rev | xargs -
I % /usr/local/bin/kubectl logs jobs/%
```

This command will return the last completed CAS Manager health probe's logs and will state when the certificates will be expiring. For example:

```
Secret Provider type is Vault
Vault certificate is valid from Tue Aug 17 19:35:42 2021 until Sat Sep 18 19:
35:42 2021
Vault status - Initialized: True, Sealed: False
Vault is healthy
MongoDB certificate is valid from Tue Aug 17 19:35:42 2021 until Sat Sep 18
19:35:42 2021
MongoDB is healthy
CASM is healthy
```

These commands will show the expiration date for the Mongo Database and Vault in both the default MongoDB/Vault mode or external MongoDB/Vault mode. For external MongoDB/Vault modes, you need to manually change the certificates yourself on the external instances since CAS Manager does not have the necessary permissions or functionality to do that for you.

Internal Certificate Generation

In the case where the certificate has expired or is about to expire, and you do not wish to upgrade your CAS Manager instance, you can generate internal certificates by running the following command:

```
/usr/local/bin/cas-manager configure --generate-certs
```

Once you have run this command, check the output of the diagnostics health command or the CAS Manager health probe as shown in point 3 above. Please note that this will only update the certificates within CAS Manager, so if you are using an external MongoDB and/or external Vault with TLS enabled, this command will not affect the external Database or Vault's certificates.

Multi-Admin Support

Once CAS Manager is installed, a local **adminUser** user is created to manage CAS Manager. Optionally, Active Directory or SAML integration can be configured to support additional admin users. If you don't configure either of these integrations, **adminUser** will be the only admin user.

Active Directory Integration

The Active Directory used to enable multi-admin in CAS Manager does not need to be the same Actve Directory that was used by the Connector to manage the users for the remote workstation.

The Active Directory Domain Controller machine must be accessible from the target machine that CAS Manager is installed on over the LDAPS port (TCP 636). Typically, this is only the case if both machines are on the same LAN.

To enable multiple users to manage CAS Manager, an Active Directory LDAPS configuration must be added in the Active Directory as outlined here.

⚠ **CAS Manager Integration**

The following steps are applicable for configuring an Active Directory with CAS Manager only. They are not applicable to integrating with CAS Manager as a Service.

⚠ **Active Directory Configuration Permissions**

Only the **adminUser** has the permissions to configure the Active Directory with CAS Manager from the CAS Manager Admin Console.

DNS and Name Resolution

You must ensure that you can resolve your AD domain and controller. For information on how to install and edit resolve.conf, and configure DNS name resolution, see Configuring DNS Name Resolution.

The following steps outline how to configure Active Directory integration in the CAS Manager Admin Console:

1. Go to the CAS Manager Admin Console and log-in using your CAS Manager admin credentials.
2. Click on **adminUser** from the user account tab and then click on **Multi-Admin Settings**.
3. Click on the **Active Directory Configuration** tab.
4. Enter the Active Directory configuration information:
    - Domain Controller URL: This is the URL where your domain controller is hosted, for example ldaps://dc.example.com.
    - Admin Connection DN: This is the distinguished name (DN) of a user within your AD that is able to search for users. Microsoft AD supports using UPN format for logging in. For example cn=casm_admin,cn=Users,dc=example,dc=com or casm_admin@example.com.
    - Admin Password: This is the password for the admin user defined by "Admin connection DN".
    - CASM group DN: This is the DN of a group in your AD. Only users that belong to this group will be able to authenticate to CAS Manager. For example, cn=CASM Admins,cn=Users,dc=example,dc=com.
    - Search Base DN: This is the DN of the container in your AD where we will search for user's to authenticate. For example, cn=Users,dc=example,dc=com
5. Click **SAVE** to save the configuration.

User's from the Active Directory belonging to the CASM group will now be able to navigate to the Cloud Acces Manager Admin Console login page and authenticate using their Active Directory credentials.

SAML Integration

If the Active Directory Domain Controller cannot be accessed, you can alternatively enable Active Directory users to login by enabling the Admin Console's SAML integration.

The following steps outline how to enable SAML integration and configuration of IDP settings, admins and groups access and general configuration information:

1. Go to the Admin Console.
2. Log-in using your CAS Manager admin credentials.
3. After logging into the Admin Console click on **adminUser** from the user account tab and then click on **Multi Admin settings** to open the preferences page.
4. Click on the **SAML** tab.
5. Enter the SAML configuration information:
    - The first section contains auto-generated information about the login URLs and IDP:
        - **CAS Manager login page**: A link to the page for multi-administrator login to the Admin Console
        - **Direct login via identity provider**: An endpoint to which multi-admin sign-in requests can be sent
        - **Assertion Consumer Service URL**: The callback URL provided to the IDP to which user information is sent once the IDP has authorized the user
        - **Audience URL**: The entity ID that the IDP can use to identify the Admin Console
    - The second section contains IDP settings that can be updated to manage the SAML configuration within the CAS Manager:
        - **Identity Provider Login URL**: The IDP endpoint to which SAML authentication requests are sent
        - **Identity Provider Certificate**: The public certificate of the IDP used to verify the signature of the IDP. You can also upload a .xml file that contains your IDP information.
    - The third section enables you to add new admins as well as displaying all existing admins that are allowed to login via an IDP. To enable the access for a single user, visit the **Allowed admins** tab, enter their e-mail, and click the **Add Admin** button.
    - The fourth section enables you to add new groups as well as displaying all existing groups that are allowed to login via an IDP. To enable the access for a group of users, visit the **Allowed groups** tab, enter the claim type and group claim and click **Add Group**. The claim type informs CAS Manager how the group is returned in the SAML assertion by your

IDP. The group claim matches against the group either in the Group Name claim or in the Group ID claim returned in the SAML assertion for a user based on the claim type defined for the group.

A user's access via SAML can be enabled or disabled on either the Allowed admins or Allowed groups tabs.

Configuring the Active Directory for CAS Manager

CAS Manager uses Lightweight Directory Access Protocol (LDAP) or Secure Lightweight Directory Access Protocol (LDAPS) with Active Directory servers for user authentication. LDAPS is recommended to give you a more secure environment, through the use of an Active Directory Certificate, which should be available before activating the Active Directory configuration.

The following section details how to configure and add an existing Active Directory with CAS Manager. You must have an existing Active Directory to use with CAS Manager.

Test LDAPS

The first step is to test LDAPS. For information on adding a self-signed certificate to enable LDAPs, see the following KB article.

The following command outlines how to test LDAPS through PowerShell as an Admin. Enter the name of your domain controller in place of `dc1.example.com` :

```
openssl s_client -connect "dc1.example.com":636
```

If you see a certificate successfully returned, then LDAPS for the Active Directory is configured and functioning.

Configure Active Directory for CAS Manager

The following steps outline how to configure the Active Directory for CAS Manager:

1. Open the system Control Panel and select **Administrative Tools**.
2. Click **Active Directory Users and Computers** from the list of options. If you don't have an Active Directory installed, then this option will not appear.
3. Create the following groups and users within the ***Users*** folder:

   New Group: **TESTGROUP** New User: **testUser**

The group and user names used above are just examples and can be replaced with any names you choose.

4. Once you have created this new group and user you need to access the CAS Manager Admin Console and configure the Active Directory. For information on how to do this, see Active Directory Integration.

If you are using the CAS Manager as a simple broker without power management, the Active Directory user you select will need to have read permission to query the Active Directory. A simple Domain Users group will suffice. If you are using the CAS Manager with power management features enabled, please see the following section of the CAS Manager Administrator's guide, here.

AWS Configuration

The following page outlines how to enable AWS features through the AWS management console on CAS Manager. The first step is to create a policy that can be attached to a service account. This service account will allow CAS Manager to manage resources within the provided AWS account.

Roles and Permissions for AWS

Prior to creating and assigning a permissions policy, you need to ensure that it contains the following permissions:

- **Service**: EC2
- **Actions**:
  - List: DescribeInstances
  - Write: RebootInstances StartInstances StopInstances TerminateInstances

There are additional permissions needed to verify that the policy has all the required permissions before being added to a deployment:

- **Service**: IAM
- **Actions**:
  - List: ListAttachedUserPolicies ListUserPolicies
  - Read: GetUser GetUserPolicy GetPolicy GetPolicyVersion SimulatePrincipalPolicy

If the user tries to add an AWS policy that doesn't have these permissions, CAS Manager will still add the policy but will not validate that it has the required permissions.

Please note the permissions required for AWS configuration with CAS Manager as a Service are different to the permissions required for CAS Manager. See AWS Permissions Policies for CAS Manager as a Service for information on these permissions. Currently, the permissions required for Azure and GCP configuration are the same between CAS Manager and CAS Manager as a Service.

Create a CAS Manager Policy in AWS

The following steps outline how to create the required AWS policy that you can attach to a AWS User to manage AWS resources:

1. Go to the IAM Management page in the AWS management console.
2. From the sidebar, click **Policies**.
3. Click **Create policy**.
4. For **Service** click **EC2** from the list of services.
5. Under **Access level** expand the **List** section and select **DescribeInstances**.
6. Under **Access level** expand the **Write** section and select the following permissions:
   - **RebootInstances**
   - **StartInstances**
   - **StopInstances**
   - **TerminateInstances**
7. For **Service** click **IAM** from the list of services.
8. Under **Access level** expand the **Read** section and select the following permissions:
   - **GetUser**
   - **SimulatePrincipalPolicy**
9. For **Resources** click **All resources**.
10. Leave **Request conditions** blank and click **Review policy**.
11. Give the newly created policy a name and click **Create policy**.

Create CAS Manager Service Account for AWS

This service account will have the ability to perform required actions in AWS. This will let the service account manage resources that the user has access to.

The following steps outline how to create the CAM service account:

1. Go to the IAM Management page in the AWS management console.

2. From the sidebar, click **Users**.
3. Click **Add user**.
4. Give the user a name and select **Programmatic access** as the Access type.
5. Click **Next: Permissions**.
6. Click **Attach existing policies directly** and search for the policy you created above that has EC2 permissions and select it. Optionally, you can add a tag to this role.
7. Click **Next:Review**.
8. Click **Create user**
9. Copy the **User name**, **Access key ID** and **Secret access key** credentials and save them to a secure location.

Add the AWS Service Account to a CAS Manager Deployment

The next step requires you to add the AWS service account you have created from the previous steps in the AWS management console to CAS Manager. This service account will have the CAM policy created in the previous step.

The following steps outline how to add the information to CAS Manager:

1. Log in to CAS Manager.
2. Select the CAS Manager deployment ou want to add the AWS service account to.
3. Click **Edit Deployment**.
4. Click the **Cloud service accounts** tab and open the AWS container.
5. Enter the **User name**, **Access key ID** and **Secret access key** values that you saved previously in the AWS form.
6. Click **Submit**.

CAS Manager will have be able to manage AWS machines that get added to this deployment.

Adding a Sumo Logic Log Collector

The following section details how to add a Sumo Logic log collector to CAS Manager. For information on Sumo Logic, see here. In order to add the log collector you must have a CAS Managers instance, and a Sumo Logic account that has the permissions levels required to create log collectors.

1. SSH to the CAS Manager host and create the Sumo Logic configuration file:

```
cd ~
vim sources.json
```

2. Paste in the following information:

```
{
"api.version": "v1",
"sources": [
    {
        "name":  "test", # <<< Replace this with your own or leave it as is
        "category":  "casm/test", # <<< Replace this with your own category
or leave it as is
        "automaticDateParsing":  true,
        "multilineProcessingEnabled":  false,
        "useAutolineMatching":  false,
        "forceTimeZone":  false,
        "timeZone":  "Etc/UTC",
        "filters":  [
                    ],
        "cutoffTimestamp":  0,
        "encoding":  "UTF-8",
        "pathExpression":  "/var/log/containers/*.log", # <<< this tells
sumologic which file paterns to ingest. We only care about the logs. Leave
this as is.
        "blacklist":  [
                      ],
        "sourceType":  "LocalFile",
        "alive":  false
    }
]
}
```

3. Download the Sumo Logic Collector:

   ```
   curl "https://collectors.sumologic.com/rest/download/linux/64" -o
   SumoCollector.sh
   sudo chmod +x SumoCollector.sh
   ```

4. Install the Sumo Logic Collector. For more information on installing the Sumo Collector, see here.

Once you have access to the CAS Manager host, you need to perform one of the following:

- Installation using an installation token

Installation tokens can be created by going to **Administration>Security>Installation Tokens** in the Sumo Logic web app and adding a token. Once you have completed this, run the following command:

```
sudo ./SumoCollector.sh -q -Vsumo.token_and_url=<Your-Installation-Token> -
VsyncSources=/path/to/sources.json
```

- Installation using an access key

Access keys can be created by going to the **Preferences** page in the Sumo Logic web app and adding an Access key. You will then be able to copy the accessID and accessKey. Once you have completed this, run the following command:

```
sudo ./SumoCollector.sh -q -Vsumo.accessid=<accessid> -
Vsumo.accesskey=<accesskey> -VsyncSources=/path/to/sources.json
```

Once you have the Sumo Logic collector installed, you can log into Sumo Logic and access the logs for this collector.

Preparing a Secret Storage Application

The following section outlines the steps involved in preparing specific secret storage applications prior to installing CAS Manager. Once you have CAS Manager installed, you can configure the secret storage application to work with CAS Manager.

Preparing Azure Key Vault

The following section outlines how to prepare Azure Key Vault for key and secret encryption and storage with CAS Manager.

Before configuring CAS Manager to use the Azure Key Vault you need to complete the following steps:

1. Create an Azure service principal that is able to read, write and delete secrets from/to the Azure Key Vault. For information on how to create this service principal, see App Objects and Service Principals.
2. Create an Azure Key Vault. For information on how to create an Azure Key Vault, see Quickstart: Create a key vault using the Azure Portal.

Once you have completed the pre-requisite steps above, return to the Installing CAS Manager - External Database and Secret Storage Configuration and complete the installation of CAS Manager.

Preparing Hashicorp Vault

The following section outlines how to prepare Vault for key and secret encryption and storage with CAS Manager.

Deploying Vault with Consul and Integrated Storage (Raft)

For information on setting up a Vault server using Consul as a storage backend, see Hashicorp's official deployment guide see Vault using Consul. This guide demonstrates how to deploy a Vault in a high availability mode.

HashiCorp's recommendations for a production level deployment of Vault can be found here Production Level Deployment.

Hashicorp's official deployment guide for setting up a Vault server using Integrated Storage (Raft) as a storage backend can be found here Vault with Raft Storage.

Reference Instructions for MongoDB and Vault Configuration

For detailed deployment instructions on installing and configuring MongoDB and Vault in a single virtual machine to be used by CAS Manager, see the following KB article. This KB article outlines in detail how to install and configure an instance of MongoDB and an instance of Vault on the same virtual machine. This KB article should be used in conjunction with the installation steps outlined in this section.

All configuration steps outlined should be used as a reference only. For specific details user's should visit the vendors official documentation and knowledge base.

The following steps outline how to prepare Vault to be used by CAS Manager. You can skip these steps if you have setup Vault and prepared it by following the KB article linked above. If you have not gone through the KB above and have already installed and configured the Vault server, following the vendors official documentation site, follow the steps below to add specific Vault configurations required for CAS Manager:

1. Initialize the Vault. For information on initializing the Vault, see Initializing the Vault.
2. Unseal the Vault. For information on sealing and unsealing the Vault, see Seal/Unseal.
3. Enable the secrets path expected by CAS Manager by running the following command:

   ```
   vault login
   vault secrets enable -version=2 -path=secret/ kv
   ```

4. Create a Vault policy called "casm-policy":

   ```
   vault policy write casm-policy - << EOF
   path "secret/data/*" {
     capabilities = ["create", "update", "read", "delete", "list"]
   ```

```
}
EOF
```

The output for this command should be:

```
Success! Uploaded policy: casm-policy
```

You can validate the policy by running the following command:

```
vault policy read casm-policy
```

5. Create a role to be used by CAS Manager by running the following command:

   ```
   vault write auth/token/roles/casm-role allowed_policies="casm-policy"
   period="768h"
   ```

   This command will create a token role with the casm policy created above. Any token created using this role will be valid for 32 days, if not renewed. If the token is renewed, then its validation period will be reset back to 32 days. This period should be set in accordance with your security guidelines and should be configured to be as low as possible. The output of this command should be:

   ```
   Success! Data written to: auth/token/roles/casm-role
   ```

6. Create a periodic token to be used by CAS Manager by running the following command:

   ```
   vault token create -role=casm-role -orphan
   ```

   This command will create a periodic token which are useful when the token in question is intended to be used by a long-running process or application. For more information on creating Vault tokens, see Vault Tokens. The output of this command should be:

   ```
   Key                    Value
   ---                    -----
   token                  <your token is here>
   token_accessor         <your token accessor is here>
   token_duration         768h
   token_renewable        true
   token_policies         ["casm" "default"]
   ```

```
identity_policies    []
policies             ["casm" "default"]
```

Once you have completed the pre-requisite steps above, return to the Installing CAS Manager - External Database and Secret Storage Configuration) and complete the installation of CAS Manager.

Configuring a Secret Storage Application

The following section outlines the steps involved in configuring specific secret storage applications to work with CAS Manager. You must have prepared the secret storage applications prior to installing CAS Manager.

> ⚠ **Data Migration between Secret Stores Prohibited**
>
> CAS Manager does not support any data migration between secret stores. If CAS Manager is originally configured with another secret store application, for example Hashicorp Vault, and is then configured to use Azure Key Vault, CAS Manager will not have access to the data stored in the original Vault. CAS Manager will not be able to retrieve any stored passwords, so login requests will fail. Teradici recommends using the same secret store throughout the lifetime of a CAS Manager instance.

Configuring Azure Key Vault

Follow the steps below to configure CAS Manager to use the Azure Key Vault as its secret storage application. CAS Manager must be installed before the command can be run:

1. SSH to the target machine where you installed CAS Manager.
2. Create a config file that contains the following parameters and information:

   ```
   {
   "vault-type": "azure",
   "key-vault-url": "<URL of Azure Key Vault>",
   "azure-client-id": "<client id of Azure service principal>",
   "azure-client-secret": "<client secret of Azure service principal>",
   "azure-tenant-id": "<tenant id of Azure service principal>"
   }
   ```

3. Run the following command to implement the config file:

   ```
   sudo /usr/bin/local/cas-manager configure --config-file <config file name>
   ```

Configuring Hashicorp Vault

Follow the steps below to configure CAS Manager to use Hashicorp Vault as its secret storage application. CAS Manager must be installed before the command can be run:

1. SSH to the target machine where you installed CAS Manager.
2. Create a file that contains the following data:

```
{
        "vault-type": "vault",
        "vault-url": "https://<vault_address>",
        "vault-token": "<vault_token>",
        "vault-secret-path": <in this example: secret/data>
}
```

3. Replace the following place holders with your own values:
   - vault_address: IP address or domain name of the Vault server.
   - vault_token: The access token generated on the Vault server that will be used by CAS Manager to access the Vault.
4. Run the following command to configure CAS Manager to use Vault:

```
sudo /usr/local/bin/cas-manager configure --config-file path-to-your-config-
file
```

After running this command, CAS Manager will validate the configuration by attempting to query the Vault's health status. If the request is successful, then CAS Manager will be configured to use this Vault. The configure command should only take a few minutes to complete. To verify that the connection to the Vault is healthy, run the following command:

```
sudo /usr/local/bin/cas-manager diagnose --health

## It will show the following if Vault is healthy:
[2021-01-25T22:49:02Z]  INFO .. Connections:
[2021-01-25T22:49:02Z]  INFO .... Vault=Healthy

## It will show the following if Vault is unhealthy:
[2021-01-26T01:47:10Z]  INFO .. Connections:
[2021-01-26T01:47:10Z] ERROR .... Vault=Vault service is unreachable
```

```
[2021-01-26T01:47:10Z] ERROR .. Overall Health=CAS Manager is in Unhealthy
state because Vault is unhealthy
```

Connecting to a Vault server with Self-Signed TLS Certificates

> ⚠ **Tested on CentOS Only**
>
> The following steps have been tested on CentOS. These steps may not work, or work differently, on different systems.

The following steps outline how to connect to a Vault server that uses self-signed TLS certificates:

1. Create a file called ***vault-config.json*** that contains the following:

```
{
        "vault-type": "vault",
        "vault-url": "https://<vault_address>",
        "vault-token": "<vault_token>",
        "vault-ca-cert-file": "<vault_ca_cert_file>",
        "vault-skip-verify-cert": false,
        "vault-secret-path": "secret/data"
}
```

2. Replace the following place holders with your own values:
   - vault_address (string): IP address or domain name of the Vault server.
   - vault_token (string): The access token generated on the Vault server that will be used by CAS Manager to access the Vault.
   - vault_ca_cert_file (string): The path to the file containing the CA certificate for your self-signed certificate.
3. Run the following command to update CAS Manager to use Vault:

```
sudo /usr/local/bin/cas-manager configure --config-file path-to-your-config-
file
```

4. If you want to skip certificate verification, include `"vault-skip-verify-cert":true` in your configuration file. Please note that this is not secure and is not recommended for production use cases:

```
{
        "vault-type": "vault",
        "vault-url": "https://<vault_address>",
        "vault-token": "<vault_token>",
        "vault-ca-cert-file": "<vault_ca_cert_file>",
        "vault-skip-verify-cert": true,
        "vault-secret-path": "secret/data"
}
```

Vault Token Auto-Renewal

CAS Manager does not renew the Vault token by default. You can manually set-up auto-renewal by configuring the `vault-config.json` file. For more information on renewing Vault tokens, see Vault Token Renewal.

CAS Manager can automatically renew the Vault token. You need to enable the setting in the `vault-config.json` file, and set the interval you wish the token to renew at. To enable this feature, add the Vault token auto-renew settings as follows:

1. Edit the `vault-config.json` file with the following settings:

```
{
        "vault-type": "vault",
        "vault-url": "https://<vault_address>",
        "vault-token": "<vault_token>",
        "vault-secret-path": <in this example: secret/data>
        "vault-enable-token-renew": true,
        "vault-token-renew-interval": "<crontab expression: eg @hourly,
@daily, @weekly, @monthly>"
}
```

2. Set the auto-renew token setting appropriately. The `vault-token-renew-interval` is a cron tab string. It can either be in a descriptor format as outlined in the above example, or you can set it to your own custom cron tab expression. The cron tab expression needs to be in the following format:

```
"<minute> <hour> <day-of-month> <month> <day-of-week>"
```

Vault Token Renewal Interval

To ensure that the Vault token is kept alive, the renewal needs to be able to occur multiple times before the token expires. If the token expires in a week, then you need to renew at least twice a week, if it expires every day, it needs to be renewed every few hours.

Vault Data Migration

CAS Manager does not do any data migration between different Vault configurations. If CAS Manager is updated to use a new Vault configuration, it will no longer be able to access the data from the previous configuration. If the admin user's password had been updated using a prior Vault configuration, you will no longer be able to login. To fix this, do one of the following:

- If you have access to the old Vault, migrate the data from the old Vault to the new Vault.
- Find the key that CAS Manager is using to look for the admin password in the Vault and then manually store the password in the Vault at that location. To find the key, stream the logs for the secretmgmt service by running the following command:

  ```
  /usr/local/bin/kubectl logs -l app=secretmgmt -f
  ```

  Log in to CAS Manager using the adminUser account and look for the log that includes the route `/internal/secrets/admin-XXX` . The password is expected to be at `<secret path>/admin-XXX` in the Vault, where ***secret path*** is the path defined by "vault-secret-path" in your CAS Manager config-file.
- Update to use a new MongoDB, or drop the `standaloneAdmins` collection in your MongoDB. WARNING: this will cause you to lose all of your CAS Manager data.

Backing up and Restoring CAS Manager Data

The following sections outline the steps to backup the data stored using in-cluster data storage applications, and used by CAS Manager. It also outlines how to restore this data from the created archive, as well as how to then migrate this data to another virtual machine.

Backing up CAS Manager Data

The following section outlines how to back up the data stored and used by CAS Manager, by creating an encrypted archive of the data.

> ⚠ **Backup Command for In-Cluster Storage Only**
>
> The backup command can only be run if CAS Manager is using in-cluster data storage for both Vault and MongoDB. This command will not work if CAS Manager is using an external Vault or MongoDB.

To backup the data run the following command in an SSH terminal:

```
sudo /usr/local/bin/cas-manager backup
```

This will create an encrypted archive of the Vault and MongoDB data used by CAS Manager. If successful, the backup archive file and decryption key file locations will be displayed in the terminal.

The backup archive file will be stored in the *`/opt/teradici/casm/backups/`* directory. The decryption key will be stored in the *`/opt/teradici/casm/.private/backup.key`*. CAS Manager will only create a new decryption key if one does not already exist. If there is an existing decryption key then it will continue to use it.

Once the file and key has been created, you will need to change the ownership of the file to the SSH user using the `chown` command. You can move the files to a specific directory, change the owner, and the correct permissions will be assigned. The following script is an example of this command:

```
ssh user1@machine1
sudo mv /opt/teradici/casm/backups/<archive_name> ~/backup.tar
sudo chown user1:user1 ~/backup.tar
exit

ssh user2@machine2
scp user1@machine1:~/backup.tar .
```

You will also need to copy the decryption key and change its ownership. The following script is an example of this command:

```
ssh user1@machine1
sudo cp /opt/teradici/casm/.private/backup.key ~/
sudo chown user1:user1 ~/backup.key
exit

ssh user@machine2
scp user1@machine1:~/backup.key .
```

Restoring CAS Manager Data

Once you have backed up the data to the encrypted archive, you need to restore this data. If the restore command fails, CAS Manager will attempt to try and restore using a backup archive that is automatically created right after the restore command has been run. If this is successful, CAS Manager will have the same data as before the restore was attempted. To skip the rollback feature, add `--skip-rollback` to the arguments of the restore function.

> ⚠ **Restore Command for In-Cluster Storage Only**
>
> The restore command can only be run if CAS Manager is using in-cluster data storage for both Vault and MongoDB. This command will not work if CAS Manager is using an external Vault or MongoDB.

To restore the data run the following command in an SSH terminal:

```
sudo /usr/local/bin/cas-manager restore --archive <path to archive file>  --key
<path to key file>
```

If you do not specify a key, CAS Manager will attempt to restore using the key found at *`/opt/teradici/casm/.private/backup.key`*.

Moving CAS Manager Data

It is possible to backup data on one CAS Manager virtual machine, and then restore the resulting archive on a separate CAS Manager virtual machine. Once you have backed up the data successfully on the first CAS Manager virtual machine, you can move the encrypted archive to another virtual machine.

You must ensure you have a machine that has SSH access to the virtual machines that host each of the CAS Manager instances. You must first copy the encrypted archive and decryption key to this machine, then you can move the data from this intermediary machine to the new CAS Manager virtual machine.

The following steps outline how to move CAS Manager data:

1. Run the following command to copy the encrypted archive to a machine that has SSH access to the CAS Manager instance:

   ```
   scp <username>@<casm1_URL>:/opt/teradici/casm/backups/<archive_name> <path
   on machine that contains casm backups>
   ```

2. Run the following command to copy the decryption key to the same machine that has SSH access to the CAS Manager instance:

   ```
   scp <username>@<casm1_URL>:/opt/teradici/casm/.private/backup.key <path on
   machine that contains casm backup keys>
   ```

3. Run the following command to move the encrypted archive file from the machine to the host of the new CAS Manager virtual machine:

   ```
   scp <path on machine that contains casm backups>/<archive_name>
   <username>@<casm2_URL>:<path on casm2 host that contains backups>
   ```

4. Run the following command to move the decryption key file from the machine to the host of the new CAS Manager virtual machine:

```
scp <path on machine that contains casm backup keys>/backup.key
<username>@<casm2_URL>:<path on casm2 host that contains backup keys>
```

Migrating from a Default to External Configuration

The following section outlines the steps involved in migrating the data stored in the internal data storage applications as part of a default configuration of CAS Manager, to MongoDB and Vault instances in the external configuration mode.

Prerequisites for Migrating CAS Manager Data

> ⚠ **Migration Commands are for Internal Storage Only**
>
> The migration commands can only be run if CAS Manager is using internal data storage for Vault or MongoDB as part of the default configuration. The Vault migration commands will not work if CAS Manager is already using an external Vault. The MongoDB commands commands will not work if CAS Manager is already using an external MongoDB.

1. Create the configuration files for the target MongoDB and Vault you are migrating CAS Manager data to. To create blank configuration files run the following command in an SSH terminal:

   ```
   /usr/local/bin/cas-manager generate --vault --mongo
   ```

   This will create `mongo-template.json` and `vault-template.json` files which will be created in the `config-templates/` directory within the current directory. The commands output will contain the full path to the files for reference.

2. Input the required parameters for the configuration file by following the instructions here.
3. In order for our migration scripts to be able to read from these files, please install the `jq` utility by running the following command in an SSH terminal:

   ```
   sudo dnf install -y jq
   ```

Migrating Internal MongoDB Data

The following steps outline how to migrate the internal MongoDB data to the external MongoDB instance as part of an external configuration of CAS Manager.

> ⚠ **Migration Commands are for Internal Storage Only**
>
> The migration commands can only be run if CAS Manager is using internal data storage for MongoDB. This command will not work if CAS Manager is using an external MongoDB instance.

Once you have configured the `config-templates/mongo-template.json`, you can run the following commands to migrate the data from the internal storage to the external MongoDB instance.

1. Run the following command in an SSH terminal to set the configuration for where to migrate the data to:

```
# Set path to Mongo Configuration file
export PATH_TO_MONGO_CONFIG='config-templates/mongo-template.json'
```

2. Run the migration script. Ensure that the `DEST_MONGO_DB` is set correctly. It should match the database specified by the MongoDB connection string in the configuration file.

```
# Run Commands to migrate MongoDB data from internal MongoDB to external
MongoDB
/usr/local/bin/kubectl exec -it deployments/mongo -- bash -c "
#!/bin/sh
set -e

# If destination DB is different from default (casmdb), set it accordingly.
export DEST_MONGO_DB='casmdb';

# Get connection string from mongo configuration file.
export DEST_MONGO_CONNECTION_STRING=$(jq '."db-connection-string"' $
{PATH_TO_MONGO_CONFIG});

# Check if TLS is enabled for external MongoDB
if [[ $(jq '."db-enable-tls"' ${PATH_TO_MONGO_CONFIG}) == 'true' ]]; then
    export MONGO_TLS='--ssl --tlsInsecure'
fi
```

```
# Get internal MongoDB's credentials
export MONGO_ADMIN=$(/usr/local/bin/kubectl get secrets/mongo-secret --
template={{.data.username}} | base64 -d);
export MONGO_DB=$(/usr/local/bin/kubectl get secrets/mongo-secret --
template={{.data.dbname}} | base64 -d);
export MONGO_PWD=$(/usr/local/bin/kubectl get secrets/mongo-secret --
template={{.data.password}} | base64 -d);

$(cat << 'EOF'
# Check if TLS is enabled for internal MongoDB. This file is volume mounted
in K8S manifest when TLS is required for mongo.
if [[ -f /certs/tls_combined.crt ]]; then
    export INTERNAL_MONGO_TLS='--ssl --tlsInsecure'
fi
rm -rf /export/
mkdir -p /export/

# Dump data from internal MongoDB
mongodump ${INTERNAL_MONGO_TLS} -u $MONGO_ADMIN -p $MONGO_PWD --db $MONGO_DB
--gzip --archive=/export/mongo.archive
# Restore dumped data to external MongoDB instance
mongorestore ${MONGO_TLS} --uri="${DEST_MONGO_CONNECTION_STRING}" --drop --
gzip --nsInclude=$MONGO_DB.* --nsFrom=$MONGO_DB.* --nsTo=$DEST_MONGO_DB.* --
archive=/export/mongo.archive

# Clean up
rm -rf /export/
EOF
)"
```

Once this command is complete, the last line logged by `mongorestore` will display a message similar to the following:

```
6 document(s) restored successfully. 0 document(s) failed to restore.
```

3. Run the following command to apply the external MongoDB configuration to complete the migration:

```
# Point CASM instance to External MongoDB
/usr/local/bin/cas-manager configure --config-file ${PATH_TO_MONGO_CONFIG}
```

After running this command, there may be some momentary down time as the database is switched over. Once the command is complete, CAS Manager should be functional. If for whatever reason you need to re-run the migration commands, you need to run the following command to start the internal MongoDB:

```
/usr/local/bin/kubectl scale deployments/mongo --replicas=1
```

Common issue are that the `DEST_MONGO_DB` environment variable set in the script and the database specified by the external MongoDB connection string in the configuration file do not match, or there are permissions issues with the credentials in the connection string. Applying the MongoDB configuration again will disable to the internal MongoDB.

Migrating Internal Vault Data

The following steps outline how to migrate the internal Vault data to the external Vault instance as part of an external configuration of CAS Manager.

> ⚠ **Migration Command is for Internal Storage Only**
>
> The migration commands can only be run if CAS Manager is using internal data storage for Vault. This command will not work if CAS Manager is using an external Vault instance.

Once you have configured the `config-templates/vault-template.json`, you can run the following commands to migrate the data from the internal storage to the to the external Vault instance.

1. Run the following command in an SSH terminal to set the configuration for where to migrate the data to:

   ```
   # Set path to Vault Configuration file
   export PATH_TO_VAULT_CONFIG='config-templates/vault-template.json'
   ```

2. Create a backup of the internal Vault's token:

   ```
   # Create backup of internal vault's token in case something fails
   /usr/local/bin/kubectl create secret generic clustervaulttoken --from-
   literal=token="$(/usr/local/bin/kubectl get secret vault-secret --
   template={{.data.roottoken}} | base64 -d)" --from-literal=address="$(/usr/
   local/bin/kubectl get secrets app --template={{.data.VAULT_ADDRESS}} |
   base64 -d)"
   ```

3. Run the migration script:

```
# Run Commands to migrate Vault data from internal Vault to external Vault
/usr/local/bin/kubectl exec -it deployments/vault -- sh -c "
#!/bin/sh
set -e

# Get target Vault settings from configuration file.
export DEST_VAULT=$(jq '."vault-url"' ${PATH_TO_VAULT_CONFIG});
export DEST_VAULT_TOKEN=$(jq '."vault-token"' ${PATH_TO_VAULT_CONFIG});
export DEST_SECRET_PATH=$(jq '."vault-secret-path"' ${PATH_TO_VAULT_CONFIG});

# Set existing vault settings
export VAULT_ADDR=$(/usr/local/bin/kubectl get secret clustervaulttoken --
template={{.data.address}} | base64 -d);
export VAULT_TOKEN=$(/usr/local/bin/kubectl get secret clustervaulttoken --
template={{.data.token}} | base64 -d);
export VAULT_SECRET_PATH='secret/';
export VAULT_SKIP_VERIFY='true';

# Dump secrets in json format
$(cat << 'EOF'
rm -rf /export/
mkdir -p /export/
for key in $( vault kv list ${VAULT_SECRET_PATH} | tail +3  )
do
    dest=/export/$key.json
    # Don't copy sub-folders
    if [[ $(echo $key | grep -E '/\s*$') ]]
    then
        continue;
    fi
    mkdir -p /export/${key%/*}
    echo \"get ${VAULT_SECRET_PATH}$key\"
    vault kv get -format=json -field=data  ${VAULT_SECRET_PATH}$key > $dest;
done

# Copy secrets to destination vault
export VAULT_ADDR=${DEST_VAULT}
export VAULT_TOKEN=${DEST_VAULT_TOKEN}
export DEST_SECRET_PATH=$(echo ${DEST_SECRET_PATH} | sed -e 's|\(.*\)data|\1|
g')
for secret_file in $( ls /export/*.json   ); do
    key_file_name=$(basename -- \"$secret_file\")
    key_name=${key_file_name%.*}
    echo \"put ${DEST_SECRET_PATH}$key_name\"
    vault kv put ${DEST_SECRET_PATH}$key_name @$secret_file;
done

# Clean up
rm -rf /export/
EOF
)"
```

On successful completion, the output will display a message similar to the following:

```
"get secret/60f9f0455234e00881fd00a2"
"get secret/admin-60f9f0365234e066b4fd00a1"
"get secret/secret-management-service-health"
"put secret/60f9f0455234e00881fd00a2"
Key                Value
---                -----
created_time       2021-07-22T22:27:59.961440121Z
deletion_time      n/a
destroyed          false
version            1
"put secret/admin-60f9f0365234e066b4fd00a1"
Key                Value
---                -----
created_time       2021-07-22T22:28:00.088969023Z
deletion_time      n/a
destroyed          false
version            1
"put secret/secret-management-service-health"
Key                Value
---                -----
created_time       2021-07-22T22:28:00.207620136Z
deletion_time      n/a
destroyed          false
version            1
```

4. Run the following command to apply the external Vault configuration to complete the migration:

```
# Point CASM instance to External Vault
/usr/local/bin/cas-manager configure --config-file ${PATH_TO_VAULT_CONFIG}
```

After running this command, there may be some momentary down time as the vault is switched over. Once the command is complete, CAS Manager should be functional. If for whatever reason you need to re-run the migration commands, run the following command to start the internal Vault:

```
/usr/local/bin/kubectl scale deployments/vault --replicas=1
/usr/local/bin/kubectl patch cronjobs vaultunseal -p '{"spec" : {"suspend" :
false }}'
sleep 60
```

A common issue is that the destination secret path is incorrect or the Vault has been sealed. If there is a problem please check the configuration and try again.

5. If everything is okay, delete the backup of the internal Vault's token by running the following command:

   ```
   /usr/local/bin/kubectl delete secret clustervaulttoken
   ```

   Once this is deleted you will no longer be able to access data from the internal Vault.

Overview

The Cloud Access Connector is an access hub installed in the customer environment which facilitates PCoIP Client connections to remote workstations. It operates in conjunction with the Teradici CAS Manager to provide user authentication and entitlement for remote workstation access, including MFA. It enables secure connectivity between users and the remote workstations by eliminating the need for a dedicated VPN by providing NAT sevices for external users.

The Connector currently runs on an Ubuntu server, we are adding the support to run on Rocky Linux 8 or RHEL 8 starting from **Connector version 22.04.0**. At HP, we are in a constant endeavour to simplify and unify our OS support strategy. To that effect, Anyware Manager and Connector will only support RHEL / Rocky Linux (8.0 & Above) starting H2 of 2023, and we will be EOLing Ubuntu CAC (actual Date TBD). You will be notified six months before a migration path from Ubuntu to RHEL / Rocky Linux for new Anyware Connectors.

The Connector enables CAS Manager to broker desktops or workstations located in AWS, Google Cloud, Microsoft Azure and on-premises environments. Based on customers' infrastructure, they may need more than one Connector. The Connector communicates with the CAS Manager which orchestrates and manages Cloud Access deployments.

You are required to have a valid registration code for Teradici Cloud Access Software to be able to successfully deploy CAS Manager. This code will be sent to you via email from Teradici and looks like ABCDEF1234@AB12-C345-D67E-89FG. For more information on Cloud Access Software, see Cloud Access Software.

Creating the Connector Server

The following sections outline how to create the Connector servers on Ubuntu and RHEL/Rocky Linux that are nescessary to run the Connectors. It also outlines the environment conditions that must be met for a successful installation.

Cloud Access Connector on Ubuntu

Cloud Access Connector is software that runs within an Ubuntu server and enables secure connectivity between users and the remote workstations. Connector runs in the customer environment such as on-premises, AWS and Google Cloud. The Connector communicates with the CAS Manager which orchestrates and manages Cloud Access deployments.

Creating the Connector Server

The Connector runs on an Ubuntu server (called the Connector server).

Create a dedicated Ubuntu server with the following specifications:

- Ubuntu Server 18.04.
- At least 4GB RAM.
- 30GB available storage or more.
- 2 vCPUs or more.

Once you have setup a dedicated virtual machine for the Connector, please ensure the following environment conditions are met:

- You must have access to the internet.
- You must have an Active Directory (AD) user account located in the designated Connector domain admins group, in order to log into the Admin Console.
- The server must be able to resolve the AD domain.
- You must be able to access the server using SSH.
- You must have superuser (sudo) privileges on the server.

- The networking information of the server (including the IP address) must not change while the Connector is operational.
- The server must have a single network interface and IP address. If the server has multiple network interfaces, the Connector will fail to install.
- If you are deploying Ubuntu on ESXi, you must install open-vm-tools to enable the ESXi host to communicate with the Connector server.
- The Connector runs on the following supported domain controller servers:
    - Windows 2016 Server with secure LDAP (LDAPS) enabled.
    - Windows 2012 R2 Server with secure LDAP (LDAPS) enabled.
    - Windows 2019 Server with secure LDAP (LDAPS) enabled.

For information on the session establishment and session bandwidth limits when working with external connections, see here.

Creating a DNS record

If you want to create a DNS record for the Connector, you need to obtain an SSL certificate with its FQDN and provide it (along with the key) when installing the Connector. This will avoid SSL certificate verification warnings.

Verifying the Connector Server

To verify your Connector server network configuration, SSH into the machine and ping the domain and a remote workstation in the domain. You should get a positive response from both attempts:

```
ping <domain FQDN>
ping <remote workstation FQDN>
```

If any of your attempts to verify these components fails, the DNS settings on the Connector server might be misconfigured. For more information on DNS configuration, see Configuring Network Settings in Ubuntu 18.04.

Cloud Access Connector on RHEL/Rocky Linux

{! cloud_access_connector/include/cac_differences.md !}

The following section outlines the minimum requirements for installing the Connector on Rocky Linux and RHEL. You need an operating system with the following specifications:

- Operating System: Rocky Linux 8 or RHEL 8.
- Minimum 8 GB RAM
- 4 CPU
- 60 GB Storage
    - If you are using LVM and **/var** is mounted on a separate volume, that volume must have 30 GB or more. This is to ensure that the installation process succeeds and CAS Manager can function at an optimum level.

Once you have setup a dedicated virtual machine for the Connector, please ensure the following environment conditions are met:

- You must have access to the internet.
- You must have an Active Directory (AD) user account located in the designated Connector domain admins group, in order to log into the Admin Console.
- The virtual machine must have ports TCP 443, and ports TCP/UDP 4172 enabled, Please check Firewall Load Balancing Considerations for additional port and firewall information.
- You must have console access to the virtual machine.
- The server must be able to resolve the AD domain.
- You must be able to access the server using SSH.
- You must have superuser (sudo) privileges on the server.
- The networking information of the server (including the IP address) must not change while the Connector is operational.
- The Connector runs on the following supported domain controller servers:
    - Windows 2016 Server with secure LDAP (LDAPS) enabled.
    - Windows 2012 R2 Server with secure LDAP (LDAPS) enabled.
    - Windows 2019 Server with secure LDAP (LDAPS) enabled.

Verifying the Connector Server

To verify your Connector server network configuration, SSH into the machine and ping the domain and a remote workstation in the domain. You should get a positive response from both attempts:

```
ping <domain FQDN>
ping <remote workstation FQDN>
```

DNS and Name Resolution

You must ensure that you can resolve your AD domain and controller. For information on how to install and edit resolve.conf, and configure DNS name resolution, see Configuring DNS Name Resolution.

Enabling Connections over WAN

If the Connector server will be accessed outside the domain, it must be configured for external access (this step is only required if you want to enable remote access to the workstations without requiring a VPN):

- The server must have a public IP address. This can be done via bi-directional NAT mapping.
- The `--external-client-cidr` flag takes priority over the `--internal-client-cidr` . The default for the `--internal-client-cidr` is 10.0.0.0/8,172.16.0.0/12,192.168.0.0/16. Any source that does not match to a `--internal-client-cidr` will default to an external connection.

For example `--external-client-cidr 0.0.0.0/0` will treat everything as an external connection, to reset to the default behaviour you would need to enter the following command and flag parameters:

```
./cloud-access-connector update --internal-client-cidr 10.0.0.0/8 --internal-
client-cidr 172.16.0.0/12 --internal-client-cidr 192.168.0.0/16
```

When setting connections from a firewall or security gateway to be external, the internal CIDR will treat connections under a certain range as internal. For example the following example will treat connections originating from under the 10.11.12.0/24 CIDR except 10.11.12.1 as internal:

```
./cloud-access-connector update --internal-client-cidr 10.11.12.0/24 --external-
client-cidr 10.11.12.1/32
```

- Port 443 TCP and 4172 UDP/TCP need to be open. Session set-up is done through port 443 and in-session traffic runs through port 4172. - The `--external-pcoip-ip` flag sets the IPv4 address for the Connector for external connections. If this value is not set, the external IPv4 address will be determined automatically. This is an optional setting that can be used when installing the Connector.

For information on the session establishment and session bandwidth limits when working with external connections, see here.

Reboot the server after NAT changes

If the NAT is configured after the Connector has been installed, reboot the Connector server.

Active Directory Service Accounts

The following sections outline the Active Directory (AD) Service Account permissions required for installing the Connector. It also outlines the steps required to set these permissions.

Permissions Required to Install the Connector

There are no mandatory permissions required for the AD Service Account to install the Connector. You can optionally delegate the **Reset user passwords and force password change at next logon** task in the Delegation of Control Wizard panel. For steps on how to delegate the password reset task to the AD Service Account, see Permissions to Change and Reset Passwords.

Delegating this task will enable users to change and reset their passwords while connecting to the remote workstations. If this is not set, the user will receive an error.

⚠ **Higher AD Service Account Permissions**

If the user has a higher level of permissions than the AD Service Account, then you will experience password change errors even if the delegation is configured as outlined above.

✎ **Domain Controller certificates**

If all DC certificates have expired, the Connector will stop working. An error indicator will display on the Connectors page when a Connector has a DC with expired certificates.

A warning indicator that details the current state of the DC certs will display on the same page when a Connector has a certificate that less than a week away from expiring.

For information on how to create and install a self-signed certificate on a Windows 2016 AD server to test LDAP connections, see KB 1707.

Permissions to Change and Reset Passwords

The following steps outline how to delegate the **Reset user passwords and force password change at next logon** task in the Delegation of Control Wizard:

1. Open the **Active Directory Users and Computers** application.
2. Select the user or group you want to delegate, and click **Delegate Control**.
3. Click **Next**.
4. Click **Add** and enter the username or group name that will be granted reset permission.
5. Click **OK**.
6. Click **Next**.
7. Select **Delegate the following common tasks** and select the **Reset user passwords and force password change at next logon** task.
8. Click **Finish**.

During Installation

When the Connector is installed, you will be prompted for the following information:

- The AD Service Account username.
- The AD Service Account password.

Permissions Required to Provision Remote Workstations

Before provisioning a remote workstation you need to ensure that the AD Service Account is correctly configured. This should be a different AD Service Account to the account used when installing the Connector. The AD Service Account needs to have specific permissions, for information on these permissions and how to configure them, see Provisioning Remote Workstations.

Assigning an SSL Certificate

You can assign an SSL certificate to the Connector during installation. This will prevent certificate verification errors when connecting to the CAS Manager or CAS Manager as a Service Interface through your browser. It will also prevent the PCoIP client from reporting an insecure connection when establishing a PCoIP session.

The certificate you provide must be signed and validated by a root certificate that the client trusts. The certificate must be combined or bundled with the intermediate certificates in PEM format and copied, along with the key, to the Connector server prior to installation.

For an example of how to create a self-signed certificate, see Creating a self-signed certificate on a Windows 2016 Active Directory Server. For an example of a method to install a certificate on your Active Directory, see Installing a certificate on your Active Directory server to enable LDAPS.

The DNS needs to be setup so that 'casm.test.com' for example, is registered to the public IP address of the application gateway.

When the Connector is installed, you will be prompted for the following information:

- The full path and filename of the SSL key
- The full path and filename of the SSL certificate

If you do not wish to specify a certificate when installing the Connector, you can bypass this by entering the command line option `--self-signed` (which is recommended strictly for testing purposes). If you decide to use a certificate later, Teradici recommends creating a new Connector and deleting the old one. For information on updating SSL certificates, see Updating the Cloud Access Connector.

Installing the Cloud Access Connector on Ubuntu

The Cloud Access Connector can be installed on Ubuntu 18.04, Rocky Linux 8 or RHEL 8. The section's below outline how to download and install the Connector on these operating systems.

> **Upgrading a Connector**
>
> It is not possible to upgrade a Connector installed on Ubuntu to a Connector installed on RHEL or Rocky Linux. To replace a Connector installed on Ubuntu, you must install the RHEL/Rocky Linux Connector on a new virtual machine and configure it exactly the same as the existing Connector on Ubuntu.

Installing the Connector on Ubuntu

The following section outlines how to download and install the Connector on **Ubuntu 18.04**. There are three main steps involved in this process:

- Downloading the Connector installer files.
- Generating a Connector token.
- Installing the Connector.

Prerequisite Steps

For instructions and documentation on the Connector prerequisite steps when installing on Ubuntu, see Connector System Requirements.

It is important to read and address all the prerequisites outlined.

Downloading the Connector

The following section outlines how to download the installer files for the Connector. First, connect to the machine and download the Connector files. The commands below will download the Connector archive, and extract it.

You need to ensure that you have a customer account created on teradici.com to access the download information.

Downloading the Installer from teradici.com

The following steps outline the current process that enables you to download the installer directly from teradici.com as a tar.gz file or else run the shell script from teradici.com:

1. SSH into the machine:

   ```
   ssh <username>@<server-ip-address>
   ```

2. Download the installer from Teradici:
   - Open a web browser and navigate to the Downloads and Scripts tab on the Teradici support site.
   - Download the installer and upload it to the machine or run the shell script provided to download the installer to the machine.
3. Unpackage the installer:
   - Previously the installer was extracted into the ~/v2connector directory. This location has now changed. Run the following command to extract the installer to /usr/sbin/:

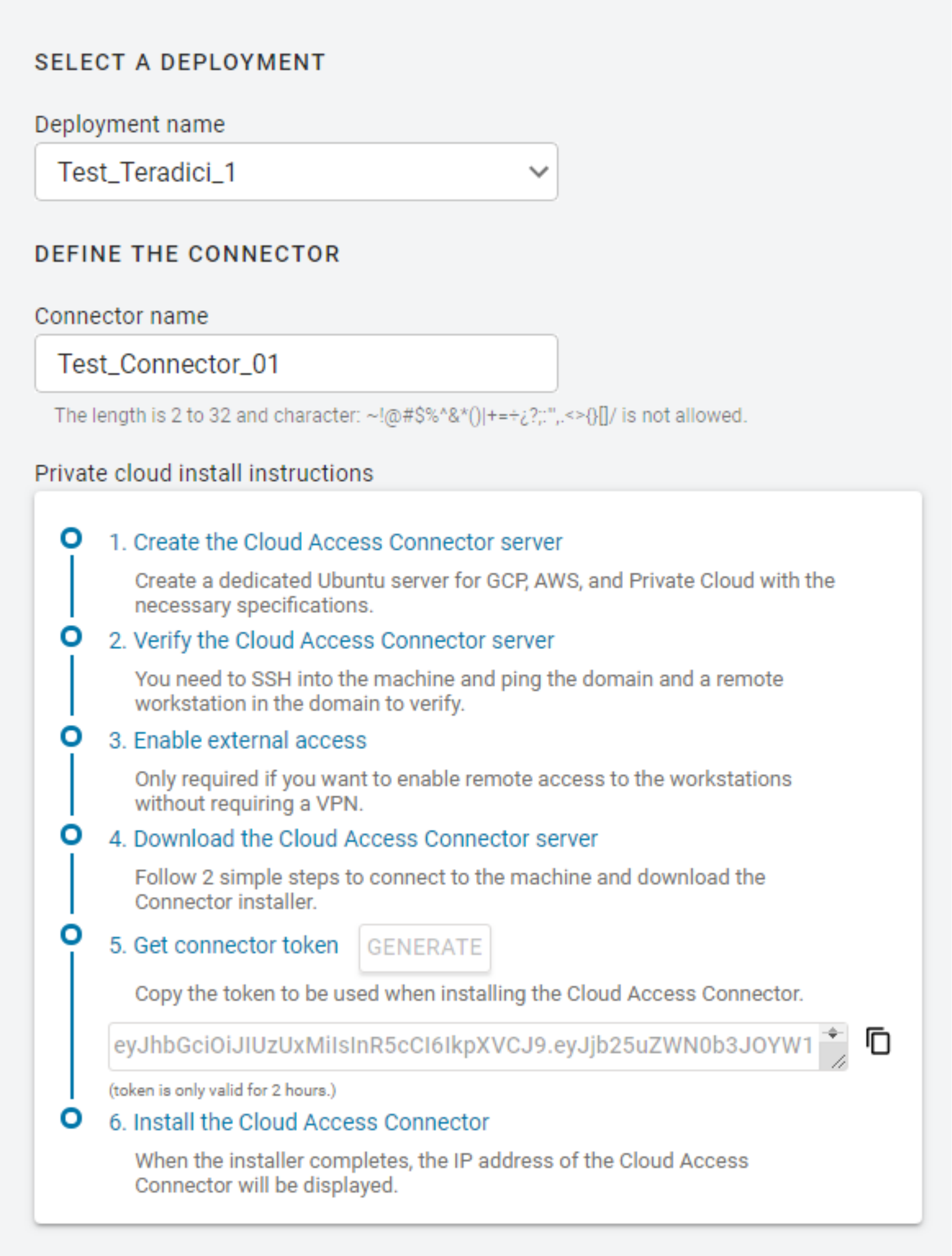
     ```
     sudo tar xzvf <PATH TO FILE>/cloud-access-
     connector_<version>_Linux.tar.gz -C /
     ```

Generating a Connector Token

You are required to have a Connector token when installing the Connector. You need to create or have created a deployment prior to obtaining a token. For information on how to log into the Admin Console, see Admin Console Connection. The following section outlines how to obtain a Connector token using the Admin Console:

1. Click **Connectors** from the console sidebar.
2. Click the add connector button (**+** sign located beside **Connectors** heading) to display the connector creation panel.

3. Enter the following information:
    - Select the deployment you want to add the Connector to. If you do not have an existing deployment you need to create one.
    - Enter the name of the Connector.
    - Follow the step by step instructions outlined below.



SELECT A DEPLOYMENT

Deployment name

Test_Teradici_1

DEFINE THE CONNECTOR

Connector name

Test_Connector_01

The length is 2 to 32 and character: ~!@#$%^&*()|+=÷¿?;:",.<>{}[]/ is not allowed.

Private cloud install instructions

1. Create the Cloud Access Connector server

Create a dedicated Ubuntu server for GCP, AWS, and Private Cloud with the necessary specifications.

2. Verify the Cloud Access Connector server

You need to SSH into the machine and ping the domain and a remote workstation in the domain to verify.

3. Enable external access

Only required if you want to enable remote access to the workstations without requiring a VPN.

4. Download the Cloud Access Connector server

Follow 2 simple steps to connect to the machine and download the Connector installer.

5. Get connector token GENERATE

Copy the token to be used when installing the Cloud Access Connector.

eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJjb25uZWN0b3JOYW1

(token is only valid for 2 hours.)

6. Install the Cloud Access Connector

When the installer completes, the IP address of the Cloud Access Connector will be displayed.

4. Click **GENERATE**.
5. Copy the Connector token by click the copy icon.
6. Click **CLOSE** the exit the panel.

You can now use this Connector token when prompted during installation.

Installing the Connector

Once the files are downloaded and the access token is set, you can install the Connector. If you are not already connected, connect to the machine via SSH and navigate to the `/usr/sbin` directory. For information on installing the Connector for CAS Manager and CAS Manager as a Service, see:

- CAS Manager
- CAS Manager as a Service
- Installing for Testing

If you are installing a Connector into a CAS Manager deployment that has other Connectors already deployed, the configuration of the latest Connector installed will take precedence.

Latest Installer Version

Ensure that you are using the latest installer prior to installing or upgrading the Connector. If you are not using the latest installer, you may see one of the following errors or warnings:

- The installer is out of date. Please obtain the latest version and try again. See Downloading the Connector for instructions.
- The installer is out of date. Please download the latest version from teradici.bintray.com/cloud-access-connector/cloud-access-connector-0.1.1.tar.gz and try again.
- A newer version is available. Please go to Downloading the Connector to obtain the latest.

For information on troubleshooting Connector installer issues related to this distribution change, see Installer Issues.

DNS and Name Resolution

You must ensure that you can resolve your AD domain and controller. For information on how to install and edit resolve.conf, and configure DNS name resolution, see Configuring DNS Name Resolution.

Installing the Connector for CAS Manager

Once you have downloaded the Connector installer and have obtained a Connector token, run the following command to install Connector to the CAS Manager instance you have just installed. The first line of this command maps the Connector token to a variable in the shell:

```
export token=<token from CAS Manager admin console>
sudo cloud-access-connector install \
-t $token \
--casm-url=https://ip-address-of-cas-manager \
--external-pcoip-ip public.ipv4.clients.connect.to \
--casm-insecure
```

- When you are installing the Connector for CAS Manager you need to ensure that you enable and specify the `--casm-url` flag. This flag specifies the CAS Manager URL that the Connector connects to. If it is not specified by default it will point to https://cas.teradici.com.
- The `--external-pcoip-ip` flag is highly recommended to use in order to explicitly set the public IP that PCoIP Clients will connect to during PCoIP sessions. This is the public IP that the Connector is listening to on port 4172. The installer will reach out to cas.teradici.com and first try to automatically resolve the external IP; if this fails, or is not able to resolve the correct IP, this flag is required. In the case that the Connector machine doesn't have an internet connection, for example in a dark site environment, or the ingress and egress internet traffic are running through different public IPs, this flag is required.
- The `--casm-insecure` flag is only required when the Connector is connecting to a CAS Manager that is using self-signed certificates. If CAS Manager is using trusted TLS certificates signed by a public CA, then users will not need to use the `--casm-insecure` command.
- The `-casm-ca-cert` flag can used to provide the PEM formatted public certificate for the private CA used to sign the CAS Manager certificate. This flag is useful if the Connector fails to fetch a certificate from the CAS Manager.

Ensure that you use the options and flags that best suit your system architecture and requirements. If required values are not provided on the command line, you will be prompted for them. For additional flags and options, see Installation Flags and Options.

Installing the Connector for CAS Manager as a Service

Install the Connector for CAS Manager as a Service by running the following command:

```
cd /usr/sbin
sudo ./cloud-access-connector install
```

Ensure that you use the options and flags that best suit your system architecture and requirements. If required values are not provided on the command line, you will be prompted for them. For additional flags and options, see Installation Flags and Options.

Cloud Access Connector - Troubleshooting

If there is an issue installing the Cloud Access Connector or an existing Connector is failing, please see the troubleshooting section on Cloud Access Connector Connectivity. Within this section there are steps to check the following:

- Remote Workstation connections
- Active Directory connections
- Cloud Access Connector component information

You can also view the following KB article here which provides a list of troubleshooting steps for common issues related to installing the Connector. For information on installer errors related to a change in the distribution system, see Installer Issues.

Installation Flags and Options

The following flags can be used to provide values at the command line. If they are omitted from the command and are required, you will be prompted for them:

Flag	Type	Description
CAS Manager		
`--casm-url`	String	Required for CAS Manager, Specifies the CAS Manager URL that the Connector connects to. If this is not specified it will point to https://cas.teradici.com by default, which is the URL for CAS Manager as a Service.
`--casm-ca-cert`	String	Enables users to supply a CA certificate for CAS Manager to enable the Connector to connect to a CAS Manager instance using self-signed certificates.

Flag	Type	Description
`--casm-insecure`	String	Is required when the Connector is connecting to a CAS Manager instance that is using self-signed certificates. If CAS Manager is using trusted TLS certificates signed by a public CA, then users will not need to use the this command.
`--ldaps-ca-cert`	String	Enables users to supply a CA certificate for the connection to Active Directory over LDAPS.
`--self-signed`	String	Installs the Connector with self-signed certificates. This mode is not secure and is intended for testing. The `--insecure` flag is still supported.
Connector		
`--token` (`-t`)	String	Required. The token generated for CAS Manager.
`--accept-policies`	—	Automatically accept the EULA and Privacy Policy.
`--force-install`	String	Replaces any existing Connector installation.
`--debug`	String	This flag can be run if you initial install of the Connector fails. It provides a detailed output of the Connector installation. This is useful for self-troubleshooting or to provide to the Teradici support team when logging a support ticket.
`--local-license-server-url`	String	Sets the URL for PCoIP License Server to be used for PCoIP Sessions. If this is not provided, ensure that the Cloud License Server is registered on the PCoIP Agent. Example: *--local-license-server-url http://10.10.10.10:7070/request*. For more information on the PCoIP License Server, see PCoIP License Server.

Flag	Type	Description
`--pool-group`	String	Specifies one or more Active Directory groups, by entering the distinguished name (DN), to be assigned to pools for remote workstation management (eg, --pool-group 'CN=GroupPool1,CN=Users,DC=sample,DC=com' --pool-group 'CN=GroupPool2,CN=Users,DC=sample,DC=com'). By providing all the existing pools groups in the Connector settings would get replaced by the user specified ones. When running this command you need to run it with **adconfig**. Example: ***sudo ./cloud-access-connector adconfig --pool-group***.
`--setup-docker-image`	String	Specifies the docker image to be used from the setup container. This is intended to be used for debugging purposes and is not recommended to be used without guidance from Teradici support. Usage without guidance could result in failed installations.
`--docker-registry`	String	This is an optional flag that enables users to specify the docker image registry that they want to use when installing or updating a Connector. If an option is not specified, the default registry ***docker.cloudsmith.io/teradici/cloud-access-connector*** will be used. This is intended to be used for debugging purposes and is not recommended to be used without guidance from Teradici support. Usage without guidance from Teradici could result in failed installations.
`--prune-image`	Boolean	Removes all unused docker images on this machine to reclaim more disk space. **Warning:** This command will remove all unused images under Connector and other services, if any. This is equivalent to the `docker image prune` command.
`--preferred-name`	String	This is an optional flag to determine if the hostname or machine name should be displayed to identify the remote workstations, the default is set to display machine name.

Firewall

Flag	Type	Description
`--https-proxy`	String	Specify the URL for a HTTPS proxy (overrides related proxy settings in environment variables)
`--connector-network-cidr`	String	This is the CIDR to use for the Connector's docker network. The default docker network subnet is 10.101.0.0/16.
`--internal-client-cidr`	String	The CIDR for PCoIP Clients that connect to remote workstations directly. It is possible to specify multiple `--internal-client-cidr` networks.
`--external-client-cidr`	String	The CIDR for PCoIP Clients that connect to remote workstations through the Security Gateway. If external CIDRs settings are set, internal settings must be explicitly set. It is possible to specify multiple `--external-client-cidr` networks.
PCoIP Software Client		
`--retrieve-agent-state`	Boolean	Enables the broker to retrieve the agent state for unmanaged and managed remote workstations. The default value for this flag is false. The available states are ***In Session***, ***Ready***, ***Starting***, ***Stopping***, ***Stopped*** and ***Unknown***. The value of this flag can either be true or false.
`--show-agent-state`	Boolean	Controls if the agent state is displayed as part of the remote workstation name in the PCoIP Client. The default value for this flag is true. Setting the value of this flag to true and the `--retrieve-agent-state` flag to false will result in no agent state displaying.

Flag	Type	Description
`--external-pcoip-ip`	String	Sets the public IP for PCoIP Client to PCoIP Agent connection. This is the public IP that the Connector is listening to on port 4172. The installer will reach out to cas.teradici.com and first try to automatically resolve the external IP; if this fails, or is not able to resolve the correct IP, this flag is required. In the case that the Connector machine doesn't have an internet connection, for example in a dark site environment, or the ingress and egress internet traffic are running through different public IPs, this flag is required. For more information on external network access, see Enabling External Network Access.
Domain		
`--domain`	String	The AD domain that remote workstations will join.
`--sa-user`	String	The Active Directory service account username.
`--sa-password`	String	The Active Directory service account password.
`--domain-controller`	String	Specifies one or more domain controllers to use with the Connector.
`--users-filter`	String	The filter to search for users within Active Directory. Specify multiple filters with multiple options. Default user filter: (&(objectCategory=person)(objectClass=user)).
`--computers-filter`	String	The filter to search for computers within Active Directory. Specify multiple filters with multiple options. Default computer filter: (&(primaryGroupID=515)(objectCategory=computer)).
`--users-dn`	StringArray	The base DN to search for users within AD. Specify multiple DNs with multiple options. Newly provided base DN(s) will automatically replace previous base DN(s). This field is looking for user's within the user-defined DN and SGs.

Flag	Type	Description
`--computers-dn`	StringArray	The base DN to search for computers within AD. Specify multiple DNs with multiple options. Newly provided base DN(s) will automatically replace previous base DN(s).
`--sync-interval`	uint8	The interval (in minutes) for how often to sync AD users and computers with the CASM Service.
MFA		
`--enable-mfa`	String	Installs with multi-factor authentication enabled.
`--radius-server`	String	The FQDN or IP address of the RADIUS server to use for MFA. This flag is optional.
`--radius-port`	String	The RADIUS server port. If not specified, the default port (1812) is used. If `--radius-server` is specifed then this flag is optional.
`--radius-secret`	String	The shared secret used for configuring RADIUS authentication. If `--radius-server` is specifed then this flag is required.
Certificates		
`--ssl-key`	String	The full path and filename of the SSL key to use. The `--self-signed` flag overrides this flag.
`--ssl-cert`	String	The full path and filename of the SSL certificate (in PEM format) to use. The `--self-signed` flag overrides this flag.

Troubleshooting the Connector

If you encounter issues when attempting to install the Connector, please see the Troubleshooting section for information on how to potentially diagnose the specific issue. You can also view the following KB article here which provides a list of troubleshooting steps for common issues related to installing the Connector. For information on installer errors related to a change in the distribution system, see Installer Issues.

Multi-Factor Authentication

When installing the Connector you can enable multi-factor authentication (MFA) by running the `--enable-mfa` flag. MFA will be disabled by default. If you want MFA to only apply to external connections, you should have separate Connectors. One Connector should be for external connections, where MFA is enabled, and one for internal or direct connections, where MFA is disabled. For steps on how to install the Connector with MFA bypassed for internal connections, see Installing the Connector for Internal Connections.

For external facing Connectors you should apply firewall and network settings, such as placing it in a DMZ for example. For external facing Connectors, ensure that you set `--external-client-cidr` to `0.0.0.0/0` so that everything through this Connector is treated as an external connection. It is not recommended to rely on the IP range to manage authentication levels, and for better security you should use separate Connectors.

Ensure that you use the options and flags that best suit your system architecture and requirements. If required values are not provided on the command line, you will be prompted for them. For additional flags and options, see Installation Flags and Options.

Installing the Connector for Internal Connections

The following steps outline how to install the Connector for internal connections to bypass MFA:

1. Prepare a virtual machine in your private network that meets the system requirements with the following sub-steps:
    - Skip the step for preparing the system for external access.
    - Skip the step for setting up MFA.
2. Install the Connector with the following sub-steps:
    - Do not set the Public IP using the `--external-pcoip-ip` flag. The Connector will instead return the virtual machines IP address.
    - No MFA flag is required as MFA is disabled by default.
3. Once you have installed the Connector connect to a remote workstation with a PCoIP Software Client with the following sub-step:
    - In the ***Host Address or Code*** field enter the private IP of the internal Connector you just installed and log-in.

If you want to use the same url for an external Connector as an internal Connector, for example connector.domain.com, you must set-up an internal/private DNS. In this DNS create an entry called connector.domain.com and map it to the private IP of the internal Connector. User's will then be able to connect to this entry by entering connector.domain.com in the ***Host Address or Code*** field in the PCoIP Client. The internal connection will connect to the internal Connector, and the external connection will connect to the external Connector.

Connecting to a Remote Workstation with a PCoIP Client

After successfully installing a Connector, you can initiate a session to connect to a remote workstation with a PCoIP Software Client. Teradici enables customers to use multi-factor authentication for these PCoIP Client sessions. The following steps outline how to connect to a remote workstation using the PCoIP Software Client:

1. Double-click the PCoIP Client desktop icon or program file `PCoIPClient` to launch the application.
2. In the ***Host Address or Code*** field, enter one of the following:
    - For direct connections, provide the address of the host machine.
    - For managed connections, provide the address of the connection manager.
3. Click **NEXT**.
4. Select your domain and enter the credentials for the remote workstation. If you have enabled MFA then you will be prompted for the 2nd factor passcode. The method of how this passcode is communicated depends on the provider you used. It is usually either a One Time Password or push notification.
5. Click **LOGIN**.
6. If your login is successful you should be able to select the remote workstation and connect to it. Please note that if you have a single remote workstation, that remote workstation is automatically selected and the connection is initiated immeadiately. In this case you will not be presented with a remote workstation selection screen.

For more information about the PCoIP Software Client, please see the following PCoIP Software Client guides:

- PCoIP Software Client for Windows

- PCoIP Software Client for macOS
- PCoIP Software Client for Linux

Upgrading the Cloud Access Connector on Ubuntu

The following section outlines how to update the Connector on Ubuntu.

When updating an installed Connector you must download the latest version of the Connector installer. For information on how to download the Connector installer, see Connector Installation. All parameters persist from installation using pre-defined configurations and do not need to be updated unless new configurations are required. For more information on this please see the Persistent Parameters section below.

Once you have downloaded the latest installer, run the update command:

```
cd /usr/sbin
sudo cloud-access-connector update
```

Internal IP Address

As part of the update command the Connector will send its internal IP address to CAS Manager. Previously, this only occured during installation.

Latest Installer Version

Ensure that you are using the latest installer prior to installing or updating the Connector. See Installing a Connector. Please note that older installs and updates may still be in the legacy directory at `~/v2connector` . If you are not using the latest installer, you may see one of the following errors or warnings:

- The installer is out of date. Please obtain the latest version and try again. See Downloading the Connector for instructions.
- A newer version is available. Please go to Downloading the Connector to obtain the latest.

For information on troubleshooting Connector installer issues related to this distribution change, see Installer Issues.

Persistent Parameters

Parameters can persist from installation through an update using the pre-defined configurations. As part of the update command, the Connector will search and read from the existing configuration and use the pre-existing information as part of the update.

If you wish to update any parameters with new information as part of the update, you can add these parameters when you are running the update command, for example, if you wanted to update the domain controller you would run the following command:

```
cd /usr/sbin
sudo cloud-access-connector update --domain-controller mydomain.com
```

If you do not add domain controllers during the update, any domain controllers that have been previously saved in the configuration will be used. If there are no domain controllers saved, the system will do an auto-discovery to find which domain controllers could be used.

Expired User Credentials

Be aware that you have a `--sa-user` or `--sa-password` that are expired and you do not add the new credentials to the update, then the update will fail. Please ensure these credentials are valid when performing an update of the Connector.

Installation Flags and Options

For detailed information on the installation flags and options that you can pass during installation, see the Installation Flags and Options table in the Connector on Ubuntu installation instructions.

Enabling MFA While Updating

You can enable MFA to the Connector with the `--enable-mfa` flag when performing an update. You need to have the following information:

- RADIUS server IP address or FQDN.
- RADIUS shared secret for configuring RADIUS authentication.

```
sudo ./cloud-access-connector update --enable-mfa
```

If you do not provide the locations of your RADIUS server and RADIUS shared secret, you will be prompted to do so.

Removing MFA While Updating

You can disable MFA from the Connector with the `--disable-mfa` flag when performing an update.

```
sudo ./cloud-access-connector update --disable-mfa
```

Updating SSL Certificates

Before updating SSL certificates, ensure that you aware of the requirments for creating and updating certificates, see Assigning a Certificate to the Connector. You can update your Connectors SSL certificate and key by running the following command and specifying your SSL certificate and SSL key information:

```
sudo ./cloud-access-connector update --ssl-cert path/to/cert --ssl-key path/to/
key
```

Certificate format

The SSL certificate must be a PEM file. A CRT formatted file will not work with the update command above.

This command will enable you update your SSL certificate information without having to re-install the Connector. This command also enables you to change your self-signed certificate to a signed certificate.

Domain Controller Certificates

If all DC certificates have expired, the Cloud Access Connector will stop working. An error indicator will display on the **Connectors** page when a Cloud Access Connector has a DC with expired certificates.

A warning indicator that details the current state of the DC certs will display on the same page when a Cloud Access Connector has a certificate that less than a week away from expiring.

Multi-Factor Authentication (MFA)

CAS Connector supports Multi-Factor Authentication (MFA) for PCoIP client sessions. The CAS Connector MFA implementation is based on the RADIUS protocol. Customers can leverage their existing RADIUS server installation to enable MFA for CAS Connector deployments. The following MFA scenario's have been tested with specific versions of the MFA software in question. Different versions may not yield the same results and may lead to different behavior.

Multi-Factor Authentication with Duo

Duo Authentication Version

The Connector was tested with Duo version **2.4.21**.

In regards Duo authentication, the following information is configured in the `authproxy.cfg` file. When installing the Connector it will require the following information to configure the Duo Radius server:

- Radius Client IP (Connector IP)
- Radius Server Port
- Radius Shared Secret
- Duo authentication settings (ikey, skey and api host)

Multi-Factor Authentication PCoIP Client Support

Android PCoIP clients do not presently support RADIUS MFA.

For information on enabling Duo authentication with CAS Connector, see CAS Connector Multi-Factor Authentication.

Multi-Factor Authentication with Azure

Microsoft Azure MFA Component Versions

Teradici tested the Connector with Microsoft Azure MFA on **November 15th 2019** with the following components.

Teradici component versions:

- PCoIP Software Client for Windows 19.11.
- Connector with MFA flag enabled.
- PCoIP Standard/Graphics Agent 19.11.

3rd party component versions:

- Azure Active Directory Premium or Microsoft 365 Business offering to use Azure MFA.
- Network Policy Server (NPS) acting as the RADIUS server.
- NPS extension **1.0.1.32**.
- Microsoft Authenticator App **1911.7724** (Android/iOS).

Using different versions may result in different behavior and has not been tested by Teradici.

Azure MFA can successfully be used as a 2nd factor tool for authenticating into the Connector. The following components are required to enable this MFA set-up:

- Azure Active Directory Premium or Microsoft 365 Business offering to use Azure MFA.
- Network Policy Server (NPS) acting as the RADIUS server.
- NPS extension **1.0.1.32** for Azure MFA sending requests from NPS to Azure MFA cloud service.
- Microsoft Authenticator App **1911.7724** (Android/iOS) to receive Push or to generate a Passcode.

Generated Passcode is not usable with Connector and Azure MFA

Only Microsoft Authenticator App Push Notification is supported due to Azure MFA using Modern Authentication. Selecting **Send Me a Push** or **Submit Passcode** triggers a push notification on your Microsoft Authenticator App. You will successfully connect to your PCoIP Session once you approve the push on your Android/iOS device.

For further information on configuring the required 3rd party components to enable Azure MFA with Connector, see CAS Connector Multi-Factor Authentication.

Third-Party Multi-Factor Authentication

It is possible to integrate third-party MFA applications with CAS Manager and Cloud Access Software. Teradici has tested MFA integrations with certain applications and versions of Cloud Access Software, within specific environments. The links outlined below point to knowledge base articles that outline the processes involved in setting up these specific integrations.

> ⚠ **Third-Party MFA Information**
>
> The knowledge base articles contain steps and processes that were accurate at the time of testing. Third-party applications may get updated or change without notice, and not behave as the instructions describe. In this case, please let Teradici know so that we can review, investigate and update these procedures. Different combinations or versions of applications may behave differently than described, or may not work. CAS Manager is also compatible with MFA solutions that support RADIUS. However, not all functions may work, and some of these solutions using RADIUS may behave differently and not work.

- Cloud Access Software - Okta MFA Integration in GCP

Overview

This section outlines beta features and enhancements that have not yet been pushed to the production version of CAS Manager. As a result, these features may change as they are developed, and they will not be supported by Teradici Global Support Services. Features in the beta version are considered not yet ready for full production and you use them at your own risk.

The following beta features for CAS Manager are currently documented:

- Installing the Cloud Access Connector on RHEL/Rocky Linux
- Upgrading the Cloud Access Connector on RHEL/Rocky Linux

Once these features have been fully developed and moved to the production version of CAS Manager, they will be removed from this section and added to the main Administrators' guide.

Installing the Cloud Access Connector on RHEL/Rocky Linux

You can configure the firewall, setup the system, download and install the Anyware connector on RHEL/Rocky Linux. If you are currently using Connector on Ubuntu, it is important to read and understand the differences Connector on RHEL/Rocky Linux introduced, To find out the side by side comparison, see Difference between Cloud Access Connector on Ubuntu and RHEL/Rocky Linux.

⚠ **Beta Feature**

Please be aware that the Cloud Access Connector on RHEL/Rocky Linux is only currently available in a beta version. As such, this feature may change as it is developed, and it will not be supported by Teradici Global Support Services. Features in the beta version are considered not yet ready for full production and you use them at your own risk.

Items	Connector on RHEL/Rocky Linux
Operating System	RHEL/Rocky Linux 8.x
Packaging	RPM Package
Deployment	Kubernetes
Connector Configuration	Configuration files and/or command line flags and parameters.

Required Configuration Flags	`--token`
	`--domain`
	`--sa-user`
	`--sa-password`
	`--accept-policies`
	`--self-signed` (or `--tls-key` and `--tls-cert` must be provided)
	`--ldaps-ca-cert`
	`--computers-dn` (for the first Connector of the deployment)
	`--users-dn` (for the first Connector of the deployment)

Items	Connector on RHEL/Rocky Linux
MFA Configuration	All connection requests both internal and external will require MFA credentials to be entered.
AD Synchronization	If computer (`--computers-dn`) and/or users DNs (`--users-dn`) parameters are not provided during installation, and no AD configuration is returned from CAS Manager, AD sync will **not** sync all objects from the AD to the CAS Manager.
AD LDAPS Certificate	The AD CA certificate must be provided to the installer by entering the information with the `--ldaps-ca-cert` parameter or by editing the configuration file.
Diagnose Commands	In addition to the `diagnose` command, you can check the health of the Connector by running the `--health` command, and can create a support bundle by running the `--support-bundle` command.
Key path and Certificate path flag	`--tls-key` and `--tls-cert`
Installation Commands	Add the repository and install the Connector RPM with the following command: `sudo yum install -y cas-connector` . Configure the Connector with flags or configuration files using the following command: `sudo /usr/local/bin/cas-connector configure {flags or path to config file}`. The `configure` command will fail with a missing parameter error if the mandatory flags or parameters are missing, the mandatory flags are `--token` , `--domain` , `-sa-user` , `--sa-password` , `--ldaps-ca-cert` , `--self-signed` (or must provide `--tls-key` and `--tls-cert`).
Update Configuration Files	Update the configuration file or specify flags to be updated: `sudo /usr/local/bin/cas-connector configure {flags or path to config file}`.
Upgrade Commands	`sudo dnf update cas-connector` and `sudo /usr/local/bin/cas-connector upgrade`
Internal/External Session Detection	In some cases the Connector on RHEL/Rocky Linux will work without any special configuration, but if you know the Connector on RHEL/Rocky Linux is only for WAN connections, it is strongly recommended to set the `--internal-client-cidr` to `""` to avoid issues. In some cases where the Connector on Ubuntu worked without configuring this setting, you may now be required to do so (eg, GCP Network Load Balancer).

The following sections outlines how to download and install the Connector on **Rocky Linux and RHEL**. There are five main steps involved in this process:

1. Adding the Connector repository
2. Configuring the SELinux components
3. Installing the RPM
4. Generating the Connector Token
5. Configuring the Connector.

Prerequisite Steps

For instructions and documentation on the Connector prerequisite steps when installing on RHEL/ Rocky Linux, see Connector System Requirements. It is important to read and address all the prerequisites outlined.

> **Before you begin**
>
> If you are currently using the Anyware Connector on Ubuntu, it is important to read and understand what the differences are between the Connector on Ubuntu and Connector on RHEL/Rocky Linux so you can prepare the installation correctly to minimize errors during installation.
>
> For more information, see Difference between Anyware Connector on Ubuntu and RHEL/Rocky Linux

Firewall Configuration

You must ensure your firewall is established and configured properly. Ensure you have port TCP 443 and port TCP/UDP 4172 enabled on the virtual machine that the Connector is running on. Please check Firewall Load Balancing Considerations for additional port and firewall information.

You need to configure the firewall of the virtual network the Connector is running in by following the commands below:

1. Login to the Connector virtual machine by ssh from a bash shell as ***root***.
2. Check and confirm if firewalld is active by running the following command:

```
sudo systemctl status firewalld
```

3. If `firewalld` is active, follow the steps outlined below for firewall configuration. If `firewalld` is inactive, and your organization does not require firewall on the Connector virtual machine, then skip the firewall configuration steps below and proceed to the remaining steps.
4. Run the following commands to configure the firewall:

```
sudo firewall-cmd --permanent --add-port=6443/tcp # virtual network flannel
sudo firewall-cmd --permanent --add-port=4172/tcp # PCoIP SG port
sudo firewall-cmd --permanent --add-port=4172/udp # PCoIP SG port
sudo firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16 # This
subnet is for the pods
sudo firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16 # This
subnet is for the services
sudo firewall-cmd --reload
```

1. Adding the Connector Repository

The virtual machine you are adding the repo to must have access to the internet. If it doesn't, you will be unable to download and install the required files.

To access the scripts and to configure and add the RHEL and Rocky Linux repository, select the Downloads and scripts option here and select the **Cloud Access Connector (RHEL/Rocky Linux)** option. Click **Downloads and scripts** and copy the script to add the Connector repo.

Once you have copied the curl command you need to run it to download the repository.

2. Configuring SELinux Policies

The following SELinux policies enable persistent storage and container logging on the Connector. If SELinux policies are not found, data stored in the Connector will be lost when the virtual machine is shut down.

Once configured, and the installation has verified SELinux, all Connector related data will persist when the target machine hosting the Connector is re-booted. To check if `selinux` is already installed on your system, run the following command:

```
sudo dnf list installed | grep cas-manager-selinux
```

The output from this command will notify if you if `selinux` is already running on your system. If it is not then you need to run the following commands to install the SELinux policies:

1. Run the following command to install the SELinux policies and set the basic framework for persistent database and Vault:

   ```
   sudo dnf install -y selinux-policy-base container-selinux
   ```

2. Run the following command to install a specific version of SELinux that has been tested for K3s:

   ```
   sudo dnf install -y https://github.com/k3s-io/k3s-selinux/releases/download/
   v1.1.stable.1/k3s-selinux-1.1-1.el8.noarch.rpm
   ```

3. Run the following command to install SELinux from the CAS Manager repo:

   ```
   sudo dnf install -y cas-manager-selinux
   ```

3. Installing the Connector RPM

Once you have installed and configured the SELinx policies you must install the Connector RPM and configuration files.

Run the following command to install the Connector RPM, the sample configuration files will be generated once the install is done:

```
sudo dnf install -y cas-connector
```

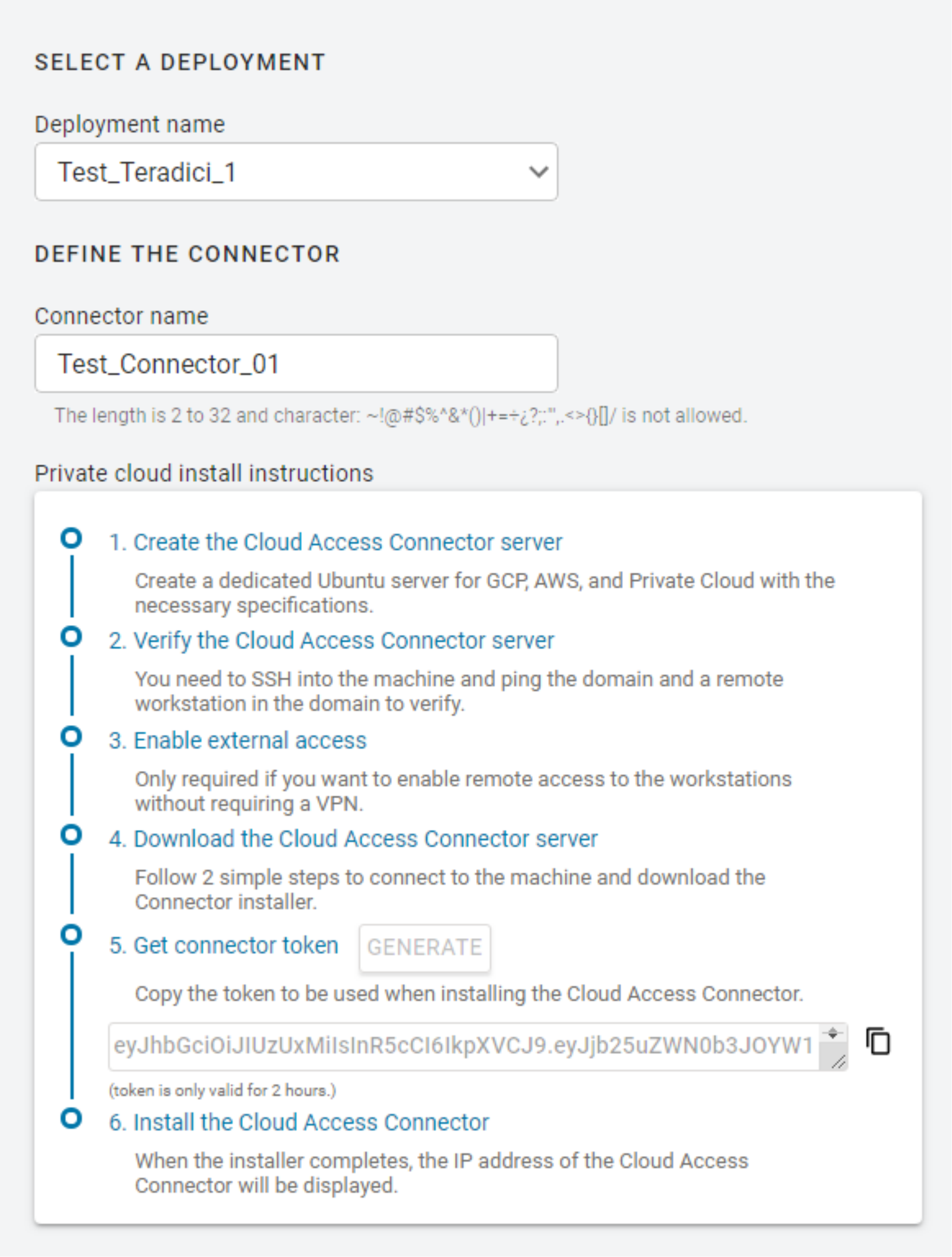
4. Generating a Connector Token

You must generate a Connector token using the Admin Console. The steps outlined below must be performed on the target virtual machine.

You need to create or have created a deployment prior to obtaining a token. For information on how to log into the Admin Console, see Admin Console Connection. The following section outlines how to obtain a Connector token using the Admin Console:

1. Click **Connectors** from the console sidebar.
2. Click the add connector button (**+** sign located beside **Connectors** heading) to display the connector creation panel.
3. Enter the following information:
    - Select the deployment you want to add the Connector to. If you do not have an existing deployment you need to create one.
    - Enter the name of the Connector.

- Follow the step by step instructions outlined below.



SELECT A DEPLOYMENT

Deployment name

Test_Teradici_1

DEFINE THE CONNECTOR

Connector name

Test_Connector_01

The length is 2 to 32 and character: ~!@#$%^&*()|+=÷¿?;:",.<>{}[]/ is not allowed.

Private cloud install instructions

1. Create the Cloud Access Connector server

   Create a dedicated Ubuntu server for GCP, AWS, and Private Cloud with the necessary specifications.

2. Verify the Cloud Access Connector server

   You need to SSH into the machine and ping the domain and a remote workstation in the domain to verify.

3. Enable external access

   Only required if you want to enable remote access to the workstations without requiring a VPN.

4. Download the Cloud Access Connector server

   Follow 2 simple steps to connect to the machine and download the Connector installer.

5. Get connector token GENERATE

   Copy the token to be used when installing the Cloud Access Connector.

   eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJjb25uZWN0b3JOYW1

   (token is only valid for 2 hours.)

6. Install the Cloud Access Connector

   When the installer completes, the IP address of the Cloud Access Connector will be displayed.

4. Click **GENERATE**.
5. Copy the Connector token by click the copy icon.
6. Click **CLOSE** the exit the panel.

You can now use this Connector token when prompted during installation.

5. Configuring the Connector with CAS Manager - Example Commands

The following section provides example configuration commands for configuring the Connector with CAS Manager and CAS Manager as a Service. These example commands use flags, but the same parameters can be configured using the configuration files also.

For information on how to configure the Connector on RHEL/Rocky Linux using the configuration files, see Configuring the Cloud Access Connector with Configuration Files.

- CAS Manager
- CAS Manager as a Service

Configuring the Connector for CAS Manager

Once you have installed the Connector RPM, and have generated a Connector token from the CAS Manager installed in your enterprise network, run the following commands to configure the Connector to work with the CAS Manager in your enterprise network. The first line of this command maps the Connector token to a variable in the shell:

```
export token=<token from CAS Manager admin console>
sudo /usr/local/bin/cas-connector configure \
--casm-url=https://ip-address-of-cas-manager \
--token $token \
--domain myexample.com \
--sa-user AD-service-account-user-name \
--sa-password AD-service-account-password \
--ldaps-ca-cert PEM-formatted-CA-certificate-for-the-connection-to-AD-over-LDAPS
\
--external-pcoip-ip public.ipv4.clients.connect.to \
--self-signed \
--accept-policies \
--casm-insecure
```

- When you are installing the Connector for CAS Manager you need to ensure that you enable and specify the `--casm-url` flag. This flag specifies the CAS Manager URL that the Connector connects to. If it is not specified by default it will point to https://cas.teradici.com.

- The `--external-pcoip-ip` flag is highly recommended to use in order to explicitly set the public IP that PCoIP Clients will connect to during PCoIP sessions. This is the public IP that the Connector is listening to on port 4172. The installer will reach out to cas.teradici.com and first try to automatically resolve the external IP; if this fails, or is not able to resolve the correct IP, this flag is required. In the case that the Connector machine doesn't have an internet connection, for example in a dark site environment, or the ingress and egress internet traffic are running through different public IPs, this flag is required.
- The `--casm-insecure` flag is only required when the Connector is connecting to a CAS Manager that is using self-signed certificates. If CAS Manager is using trusted TLS certificates signed by a public CA, then users will not need to use the `--casm-insecure` command.
- The `-casm-ca-cert` flag can used to provide the PEM formatted public certificate for the private CA used to sign the CAS Manager certificate. This flag is required when CAS Manager is using a custom certificate that is not signed by a public CA.
- If this is the first Connector installed in the deployment, use `--computers-dn` and/or `--users-dn` flags to sync AD objects to CAS Manager. The additional Connector in the same deployment will be able to pull the DN(s) configuration from CAS Manager without needing to provide these flags.
- If `--self-signed` flag is not used, you must use `--tls-key` and `--tls-cert` flags to provide the full path and filename of the TLS key and PEM formatted TLS certificate to use.

Ensure that you use the options and flags that best suit your system architecture and requirements. If required values are not provided on the command line, you will be prompted for them. For additional flags and options, see Installation Flags and Options.

Configuring the Connector for CAS Manager as a Service

Once you have installed the Connector RPM, and have generated a Connector token from the CAS Manager as a Service, run the following commands to configure the Connector to work with the CAS Manager as a Service. The first line of this command maps the Connector token to a variable in the shell:

```
export token=<token from CAS Manager Admin console>
/usr/local/bin/cas-connector configure \
--token $token \
--domain myexample.com \
--sa-user AD-service-account-user-name \
--sa-password AD-service-account-password \
--ldaps-ca-cert PEM-formatted-CA-certificate-for-the-connection-to-AD-over-LDAPS
```

```
\
--accept-policies \
--self-signed
```

- If this is the first Connector installed in the deployment, use `--computers-dn` and/or `--users-dn` flags to sync AD objects to CAS Manager. The additional Connector in the same deployment will be able to pull the DN(s) configuration from CAS Manager without needing to provide these flags.
- The `-casm-ca-cert` flag can used to provide the PEM formatted public certificate for the private CA used to sign the CAS Manager certificate. This flag is useful if the Connector fails to fetch a certificate from the CAS Manager.
- If `--self-signed` flag is not used, you must use `--tls-key` and `--tls-cert` flags to provide the full path and filename of the TLS key and PEM formatted TLS certificate to use.

Ensure that you use the options and flags that best suit your system architecture and requirements. If required values are not provided on the command line, you will be prompted for them. For additional flags and options, see Installation Flags and Options.

Cloud Access Connector - Troubleshooting

If there is an issue installing the Cloud Access Connector or an existing Connector is failing, please see the troubleshooting section on Cloud Access Connector Connectivity. Within this section there are steps to check the following:

- Remote Workstation connections
- Active Directory connections
- Cloud Access Connector component information

You can also view the following KB article here which provides a list of troubleshooting steps for common issues related to installing the Connector. For information on installer errors related to a change in the distribution system, see Installer Issues.

Multi-Factor Authentication

When you enable MFA for the Connector for RHEL/Rocky Linux, all PCoIP Clients authenticated through the Connector will be prompted to enter MFA credentials. Previously, only the external PCoIP Clients were prompted for MFA information

When installing the Connector you can enable multi-factor authentication (MFA) by running the `--enable-mfa` flag. MFA will be disabled by default. If you want MFA to only apply to external connections, you should have separate Connectors. One Connector should be for external connections, where MFA is enabled, and one for internal or direct connections, where MFA is disabled. For steps on how to install the Connector with MFA bypassed for internal connections, see Installing the Connector for Internal Connections.

For external facing Connectors you should apply firewall and network settings, such as placing it in a DMZ for example. For external facing Connectors, ensure that you set `--external-client-cidr` to `0.0.0.0/0` so that everything through this Connector is treated as an external connection. It is not recommended to rely on the IP range to manage authentication levels, and for better security you should use separate Connectors.

Ensure that you use the options and flags that best suit your system architecture and requirements. If required values are not provided on the command line, you will be prompted for them. For additional flags and options, see Installation Flags and Options.

Installation Flags and Options

For detailed information on the installation flags and the configuration file parameters that you can pass during installation, see the table outlined below:

Configuration File Parameter*	Type	Description
casManager		
caCertPath	`--casm-ca-cert`	Enables users to supply a CA certificate for CAS Manager to enable the Connector to trust the certificate in order to connect to the CAS Manager instance.
insecure	`--casm-insecure`	This flag is required when the Connector is connecting to a CAS Manager instance that is using self-signed certificates, and you want to turn off the verification of the certificate.

Configuration File Parameter*	Type	Description
url	`--casm-url`	This flag is required for CAS Manager, Specifies the CAS Manager URL that the Connector connects to. If this is not specified it will point to https://cas.teradici.com by default, which is the URL for CAS Manager as a Service.
casConnector		
acceptPolicies	`--accept-policies`	Automatically accept the EULA and Privacy Policy.
token	`--token(-t)`	Required. The token generated from CAS Manager for Connector to create a service account to connect to CAS Manager.
httpsProxy	`--https-proxy`	Specify the URL for a HTTPS proxy (overrides related proxy settings in environment variables).
casConnector/tls		
selfSigned	`--self-signed`	This mode is not secure and intended for testing only. PCoIP client will receive a untrusted warning when connecting to the Connector. The previous `--insure` flag is still supported
keypath	`--tls-key`	The full path and filename of the TLS key to use. The `--self-signed` flag overrides this flag. A string parameter.
certPath	`--tls-cert`	The full path and filename of the TLS certificate (in PEM format) to use. The `--self-signed` flag overrides this flag. A string parameter.
casConnector/connector		

Configuration File Parameter*	Type	Description
externalClientCidrs	`--external-client-cidr`	The CIDR for PCoIP Clients that connect to remote workstations through the Security Gateway. If external CIDRs settings are set, internal settings must be explicitly set. It is possible to specify multiple `--external-client-cidr` networks. A string parameter.
internalClientCidrs	`--internal-client-cidr`	The CIDR for PCoIP Clients that connect to remote workstations directly. It is possible to specify multiple `--internal-client-cidr` networks. A string parameter.
ip	`--external-pcoip-ip`	Sets the public IP for PCoIP Client to PCoIP Agent connection. This is the public IP that the Connector is listening to on port 4172. The installer will reach out to cas.teradici.com and first try to automatically resolve the external IP; if this fails, or is not able to resolve the correct IP, this flag is required. In the case that the Connector machine doesn't have an internet connection, for example in a dark site environment, or the ingress and egress internet traffic are running through different public IPs, this flag is required. For more information on external network access, see Enabling External Network Access. A string parameter.
licenseServerUrl	`--local-license-server-url`	Lets the URL for PCoIP License Server to be used for PCoIP Sessions. If this is not provided, ensure that the Cloud License Server is registered on the PCoIP Agent. Example: `--local-license-server-url http://10.10.10.10:7070/request` . For more information on the PCoIP License Server, see PCoIP License Server. A string parameter.

Configuration File Parameter*	Type	Description
retrieveAgentState	`--retrieve-agent-state`	Enables the broker to retrieve the agent state for unmanaged and managed remote workstations. The default value for this flag is false. The available states are **In Session**, **Ready**, **Starting**, **Stopping**, **Stopped** and **Unknown**. The value of this flag can either be true or false. A boolean parameter.
showAgentState	`--show-agent-state`	Controls if the agent state is displayed as part of the remote workstation name in the PCoIP Client. The default value for this flag is true. Setting the value of this flag to true and the `--retrieve-agent-state` flag to false will result in no agent state displaying. A boolean parameter.
casConnector/multifactorAuthentication		
enable	`--enable-mfa`	Enables multi-factor authentication. Multi-factor authentication will be enabled for all connections, both internal and external. Internal users will be required to enter the multi-factor authentication code for the Connector when connecting to the PCoIP Client. It is recommended to install separate Connectors for internal vs external connections. A boolean parameter.
port	`--radius-port`	This is the RADIUS server port. If not specified, the default port (1812) is used. If `--radius-server` is specified, then this flag is optional. A string parameter.
server	`--radius-server`	The FQDN or IP address of the RADIUS server to use for MFA. This flag is optional. A string parameter.

Configuration File Parameter*	Type	Description
sharedSecret	`--radius-secret`	The shared secret used for configuring RADIUS authenticaton. If `--radius-server` is specified then this flag is required. A string parameter.
deployment/domain/DomainName		
name	`--domain`	The AD domain that the remote workstations will join. A string parameter.
deployment/domain/domainControllers		
domainControllers	`--domain-controller`	This flag specifies one or more domain controllers to use with the Connector. To specify multiple domain controllers use the following format: `--domain-controller dc1.domain.com`, `--domain-controller dc2.domain.com`, `--domain-controller dc3.domain.com`. A string parameter.
deployment/domain/serviceAccount		
username	`--sa-user`	The AD service account username. A string parameter.
password	`--sa-password`	The AD service account password. A string parameter.
deployment/domain/adSynch		
computerdns	`--computers-dn`	The base DN to search for computers within AD for AD sync. Can specify multiple DNs with multiple options. Required if this is the first Connector in the deployment. see the differences between the Connectors at the top of this page for details. Newly provided base DN(s) will automatically replace previous base DN(s).

Configuration File Parameter*	Type	Description
computerfilters	`--computers-filter`	The filter to search for computers within Active Directory. Specify multiple filters with multiple options. Default computer filter: (&(primaryGroupID=515)(objectCategory=computer)). A string parameter.
usersdns	`--users-dn`	The base DN to search for users within AD. Specify multiple DNs with multiple options. Newly provided base DN(s) will automatically replace previous base DN(s). The base DN to search for computers within the AD for AD sync. You can specify multiple DNs with multiple options. Required if this is the first Connector in the deployment. See the table above on the differences between the Connectors for more information. Newly provided base DN(s) will automatically replace previous base DN(s). A string array parameter.
usersfilters	`--users-filter`	The filter to search for users within Active Directory. Specify multiple filters with multiple options. Default user filter: (&(objectCategory=person)(objectClass=user)). A string parameter.
interval	`--sync-interval`	The interval (in minutes) for how often to sync AD users and computers with the CASM Service. A uint8 parameter.
deployment/domain		
caCertPath	`--ldaps-ca-cert`	Required, to supply a CA certificate for the connection to AD over LDAPS. A string parameter.

Configuration File Parameter*	Type	Description
poolgroups	`--pool-group`	Specifies one or more Active Directory groups, by entering the distinguished name (DN), to be assigned to pools for remote workstation management (eg, --pool-group 'CN=GroupPool1,CN=Users,DC=sample,DC=com' --pool-group 'CN=GroupPool2,CN=Users,DC=sample,DC=com'). A string parameter.
N/A	`--preferred-name`	This is an optional flag to determine if the hostname or machine name should be displayed to identify the remote workstations, the default is set to display machine name.

*These configuration parameters are only applicable for the Connector on RHEL/Rocky Linux.

Troubleshooting the Connector

If you encounter issues when attempting to install the Connector, please see the Troubleshooting section for information on how to potentially diagnose the specific issue. You can also view the following KB article here which provides a list of troubleshooting steps for common issues related to installing the Connector. For information on installer errors related to a change in the distribution system, see Installer Issues.

Installing the Connector for Internal Connections

The following steps outline how to install the Connector for internal connections to bypass MFA:

1. Prepare a virtual machine in your private network that meets the system requirements with the following sub-steps:
   - Skip the step for preparing the system for external access.
   - Skip the step for setting up MFA.
2. Install the Connector with the following sub-steps:
   - Do not set the Public IP using the `--external-pcoip-ip` flag. The Connector will instead return the virtual machines IP address.

- No MFA flag is required as MFA is disabled by default.

3. Once you have installed the Connector connect to a remote workstation with a PCoIP Software Client with the following sub-step:
    - In the ***Host Address or Code*** field enter the private IP of the internal Connector you just installed and log-in.

If you want to use the same url for an external Connector as an internal Connector, for example connector.domain.com, you must set-up an internal/private DNS. In this DNS create an entry called connector.domain.com and map it to the private IP of the internal Connector. User's will then be able to connect to this entry by entering connector.domain.com in the ***Host Address or Code*** field in the PCoIP Client. The internal connection will connect to the internal Connector, and the external connection will connect to the external Connector.

Connecting to a Remote Workstation with a PCoIP Client

After successfully installing a Connector, you can initiate a session to connect to a remote workstation with a PCoIP Software Client. Teradici enables customers to use multi-factor authentication for these PCoIP Client sessions. The following steps outline how to connect to a remote workstation using the PCoIP Software Client:

1. Double-click the PCoIP Client desktop icon or program file `PCoIPClient` to launch the application.
2. In the ***Host Address or Code*** field, enter one of the following:
    - For direct connections, provide the address of the host machine.
    - For managed connections, provide the address of the connection manager.
3. Click **NEXT**.
4. Select your domain and enter the credentials for the remote workstation. If you have enabled MFA then you will be prompted for the 2nd factor passcode. The method of how this passcode is communicated depends on the provider you used. It is usually either a One Time Password or push notification.
5. Click **LOGIN**.
6. If your login is successful you should be able to select the remote workstation and connect to it. Please note that if you have a single remote workstation, that remote workstation is

automatically selected and the connection is initiated immeadiately. In this case you will not be presented with a remote workstation selection screen.

For more information about the PCoIP Software Client, please see the following PCoIP Software Client guides:

- PCoIP Software Client for Windows
- PCoIP Software Client for macOS
- PCoIP Software Client for Linux

Upgrading the Cloud Access Connector on RHEL/Rocky Linux

The following section outlines how to upgrade the Connector on RHEL/Rocky Linux. It is a 2-step process to upgrade the Connector, as outlined in the steps below:

Beta Feature

Please be aware that the Cloud Access Connector on RHEL/Rocky Linux is only currently available in a beta version. As such, this feature may change as it is developed, and it will not be supported by Teradici Global Support Services. Features in the beta version are considered not yet ready for full production and you use them at your own risk.

Upgrading a Connector

It is not possible to upgrade a Connector installed on Ubuntu to a Connector installed on RHEL or Rocky Linux. To replace a Connector installed on Ubuntu, you must install the RHEL/Rocky Linux Connector on a new virtual machine and configure it exactly the same as the existing Connector on Ubuntu.

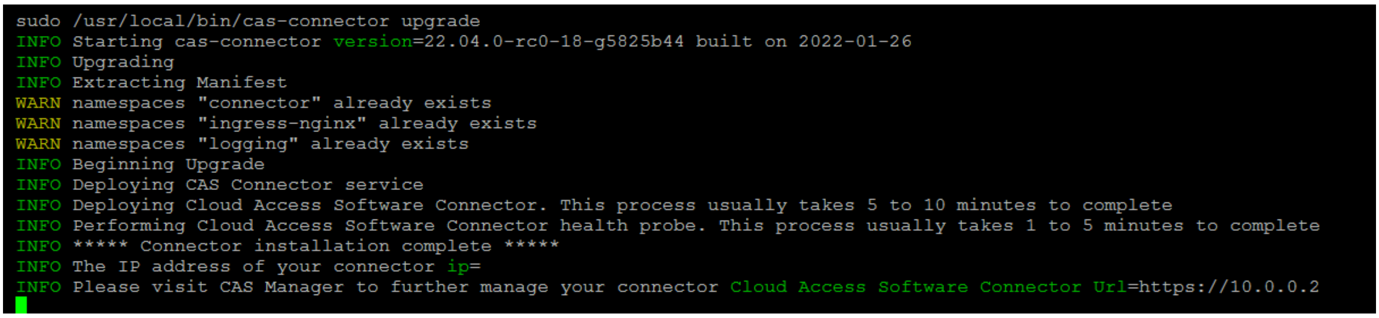
1. You must install the new version of the Connector RPM:

   ```
   sudo dnf upgrade -y cas-connector
   ```

2. Run the following command to upgrade the Connector with the current configuration:

   ```
   sudo /usr/local/bin/cas-connector upgrade
   ```

Once you have successfully upgraded the Connector you should see a response similar to the example output outlined below.



```
sudo /usr/local/bin/cas-connector upgrade
INFO Starting cas-connector version=22.04.0-rc0-18-g5825b44 built on 2022-01-26
INFO Upgrading
INFO Extracting Manifest
WARN namespaces "connector" already exists
WARN namespaces "ingress-nginx" already exists
WARN namespaces "logging" already exists
INFO Beginning Upgrade
INFO Deploying CAS Connector service
INFO Deploying Cloud Access Software Connector. This process usually takes 5 to 10 minutes to complete
INFO Performing Cloud Access Software Connector health probe. This process usually takes 1 to 5 minutes to complete
INFO ***** Connector installation complete *****
INFO The IP address of your connector ip=
INFO Please visit CAS Manager to further manage your connector Cloud Access Software Connector Url=https://10.0.0.2
```

Scaling and PCoIP Session Limits

When using CAS Manager there are certain session establishment and session bandwidth limits when dealing with external connections.

The following table outlines the RAM, vCPU and correlated estimated bandwidth support:

vCPUs	RAM	Estimated Bandwidth
2vCPU	7.5 GB RAM	~ 365 Mbit/s
4vCPU	15 GB RAM	~ 830 Mbit/s
8vCPU	30 GB RAM	~ 1100 Mbit/s

Estimated Bandwidth

These are estimated bandwidth levels. The bandwidth can vary based on the host, OS, CSP, etc.

1100 Mbit/s is approximately the maximum bandwidth that can be achieved. Additional gains may be possible with larger sizing.

CAS Connector Multi-Factor Authentication

Duo Authentication

If you wish to use Duo authentication with CAS Connector you will be required to setup an authentication server provided by Duo. For more information on this, see Duo Authentication Proxy - Reference.

Duo Authentication Version

The Connector was tested with Duo version 2.4.21.

The following are key items in the ***authproxy.cfg*** file that are relevant for the CAS Connector configuration:

```
[duo_only_client]
[radius_server_duo_only]
ikey=<integration key for duo>
skey=<secret key for duo>
api_host=<host used for duo>
radius_ip_1=<cac connection server ip>
radius_secret_1=<shared secret for above>
radius_ip_2=<cac connection server ip2>
radius_secret_2=<shared secret for above>
port=1812
```

For further information on the above integration, see RADIUS Duo Only.

Azure MFA Authentication

Microsoft Azure MFA Component Versions

Teradici tested the Connector with Microsoft Azure MFA on **November 15th 2019** with the following components.

Teradici component versions:

- PCoIP Software Client for Windows 19.11.
- Connector with MFA flag enabled.
- PCoIP Standard/Graphics Agent 19.11.

3rd party component versions:

- Azure Active Directory Premium or Microsoft 365 Business offering to use Azure MFA.
- Network Policy Server (NPS) acting as the RADIUS server.
- NPS extension **1.0.1.32**.
- Microsoft Authenticator App **1911.7724** (Android/iOS).

Using different versions may result in different behavior and has not been tested by Teradici.

Azure MFA Configuration

If you wish to use Azure MFA with the Connector you need to configure a number of 3rd party components. The following steps outline this process:

1. From within the Azure portal click **Azure AD**.
2. Click **Enable MFA for target users**.
3. Install the Microsoft Authenticator App on an Android or iOS mobile device.
4. Ensure that if the users requiring MFA are not yet populated in Azure AD, that you setup Azure AD Connect to sync On-Premises users to Azure.
5. Install Network Policy and Access role on Windows Server 2016 or 2019.
6. Install Network Policy Server (NPS) extension for Azure MFA.
7. Register NPS to Active Directory to enable it to query the list of users.

Once you have registered the NPS you need to configure the server. The following steps outline the NPS configuration:

1. From within the NPS console click **RADIUS Clients**.
2. Add the Connector IP address and Shared Secret and click **OK**. Azure MFA
3. Click **Policies** > **Connection Request Policies** and add a new policy name and click **OK**.
4. From the **Conditions** tab add the Client IPv4 Address of the Connector and click **OK**. Azure MFA 1
5. From the **Settings** tab under **Authentication** click *Accept users without validating credentials*.
6. Restart NPS services to enable these changes to take effect.

Firewall and Load Balancing Considerations

CAS Manager and the Connector require certain ports to be open to enable connections between the CAS Manager, Connector, Remote Workstations, as well as other components.

Ports and Component Connections

Component	Allow	Port/Protocol	Source/ Destination Component	Descriptions
Connector	Inbound	443 TCP	From PCoIP Clients and administrative web browsers.	For users to negotiate connections to their remote workstations. For accessing the Management Interface for (legacy) management of CAS Manager.
Connector	Outbound	443 TCP	To CAM Service, PCoIP Cloud License Server and to SumoLogic.	To sync AD information to the CAM service and call CAS Manager APIs related to negotiating PCoIP sessions. To verify license activation code during the Connector installation. For log aggregation for support purposes.
Connector	Outbound	60443 TCP	To remote workstations.	Prepares PCoIP Agents for a new user session.
Connector	Inbound	4172 TCP/UDP	From PCoIP Clients.	For PCoIP Sessions with users that are outside of the corporate network.
Connector	Outbound	4172 TCP/UDP	To remote workstations.	For PCoIP Sessions with users that are outside of the corporate network.
Connector	Outbound	636 TCP	To Domain Controllers.	To authenticate users, and query user and computer information.

Component	Allow	Port/Protocol	Source/ Destination Component	Descriptions
Connector	Outbound	1812 UDP (This port is configurable)	To RADIUS Server.	For authentication against RADIUS Server.
Connector	Outbound	53 UDP	To DNS.	Domain name resolution.
PCoIP License Server	Inbound	7070 TCP (This port is configurable)	From remote workstations.	For license activation and verification from PCoIP Agent if the PCoIP License Server is used instead of the Cloud License Server.

Port and Component Notes:

- Port **443 TCP** is not required if the PCoIP License Server is used in place of the Cloud License Server.
- The RADIUS Server is optionally configured.
- See the PCoIP License Server guide for changing port and configuring TLS encryption.

Health Check Endpoint

The following URI endpoint can be used for the CAS Manager and Connector's health check:

```
/health
```

You can use the `curl` command to verify the health check status and run it on a console. The following command is an example of using the `curl` command:

```
curl -k https://cac-machine.local:443/health
```

- If the command is successful, you will see the following response:

  ```
  {"code":200,"status":"success"}
  ```

- If the command fails, you will see the following response:

```
{"code":500,"status":"Error","reason":"Cannot communicate with broker"}
```

The following table outlines the list of possible errors and the associated status codes for the `/health` endpoint:

Status Code	Status	Example	Issue
200	success	{"code":200,"status":"success"}	N/A
500	error	{"code":500,"status":"Error","reason":"Cannot communicate with broker"}	Failure to communicate to Broker.
500	error	{"code":500,"status":"Error","reason":"Security Gateway is enabled but does not respond"}	Failure to communicate to Security Gateway.
500	error	{"code":500,"status":"Error","reason":"[error-related-for-configuration]"}	Misconfiguration for the Connection Manager

Using the Cloud Access Connector with a Web Proxy

If web access is being blocked to the machines in your environment the Connector will not work. In order to give the Connector machine access to the required resources from the internet, a web proxy server is required. The web proxy server must support the HTTP Connect method and it must be enabled. Both HTTP and HTTPS traffic will be proxied through the same proxy server.

Using the Connector on Ubuntu with a Web Proxy

The following steps outline how to use the Connector on Ubuntu with a web proxy:

1. Set up a web proxy with access to the Internet, for example Squid.
2. Ensure that HTTP Connect is enabled on the web proxy. For Squid for example, the config file may look like this:

```
# Allowed Source IPs (ie, machines with 10.xxx.xxx.xxx IPs)
acl localnet src 10.0.0.0/8 # RFC1918 possible internal network

# Allowed ports to proxy traffic (Default)
acl SSL_ports port 443
acl Safe_ports port 80      # http
acl Safe_ports port 21      # ftp
acl Safe_ports port 443     # https
acl Safe_ports port 70      # gopher
acl Safe_ports port 210     # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280     # http-mgmt
acl Safe_ports port 488     # gss-http
acl Safe_ports port 591     # filemaker
acl Safe_ports port 777     # multiling http
# Enable HTTP Connect
acl CONNECT method CONNECT


# Default Squid http_access settings
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
```

```
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager


# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost


# And finally deny all other access to this proxy
http_access deny all

# Squid normally listens to port 3128
http_port 3128
# Leave coredumps in the first cache dir (Default)
coredump_dir /var/spool/squid
# Default Refresh patterns
refresh_pattern ^ftp:       1440    20% 10080
refresh_pattern ^gopher:    1440    0%  1440
refresh_pattern -i (/cgi-bin/|\?) 0 0%  0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
refresh_pattern .       0   20% 4320
```

3. To test that the proxy is working correctly, using SSH, open a terminal on the Connector host machine and run the following set of commands:

```
# Running curl to cam should time out since the host should not be able to
route to the internet
$ curl cam.teradici.com
curl: (7) Failed to connect to cam.teradici.com port 80: Connection timed out
$ curl https://cam.teradici.com
curl: (7) Failed to connect to cam.teradici.com port 443: Connection timed
out
# Setting the proxy settings in the environment for curl to test that it
works for HTTP and HTTPS traffic
$ export http_proxy=http://<ip-of-proxy-server>:<proxy-port (default 3128)>
$ curl cam.teradici.com
<html>
<head><title>308 Permanent Redirect</title></head>
<body bgcolor="white">
<center><h1>308 Permanent Redirect</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

```
$ export https_proxy=$http_proxy
$ curl https://cam.teradici.com
<!doctype html><html lang="en"><head><meta charset="utf-8"><meta
name="viewport" content="width=device-width,initial-scale=1,shrink-to-
fit=no"><meta name="theme-color" content="#000000"><link rel="manifest"
href="/manifest.json"><link rel="shortcut icon" href="/
favicon.ico"><title>Cloud Access Manager</title><link href="/static/css/main.
27391ea7.css" rel="stylesheet"></head><body><noscript>You need to enable
JavaScript to run this app.</noscript><div id="root" class="full-height"></
div><script type="text/javascript" src="/static/js/main.45a05db7.js"></
script></body></html>
# Clear the settings from the environment
$ unset http_proxy
$ unset https_proxy
```

4. To run the installer with the proxy settings, you can apply them in the environment:

```
# Installer will read proxy setting from environment if http_proxy,
https_proxy, HTTP_PROXY, or HTTPS_PROXY are set
$ export https_proxy=http://<ip-of-proxy-server>:<proxy-port (default 3128)>
$ ./cloud-access-connector configure ...
```

   or through the command line option:

```
$ ./cloud-access-connector configure --https-proxy http://<ip-of-proxy-
server>:<proxy-port (default 3128)> ...
```

5. The installer should run as normal and configure the containers with the web proxy settings provided.

Proxy Passwords are not Supported

Proxy passwords are not supported with the Connector at this time.

Using the Connector on Rocky Linux/RHEL with a Web Proxy

The following steps outline how to use the Connector on Rocky Linux/RHEL with a web proxy:

1. Set up a web proxy with access to the Internet, for example Squid.
2. Ensure that HTTP Connect is enabled on the web proxy. For Squid for example, the config file may look like this:

```
# Allowed Source IPs (ie, machines with 10.xxx.xxx.xxx IPs)
acl localnet src 10.0.0.0/8 # RFC1918 possible internal network

# Allowed ports to proxy traffic (Default)
acl SSL_ports port 443
acl Safe_ports port 80      # http
acl Safe_ports port 21      # ftp
acl Safe_ports port 443     # https
acl Safe_ports port 70      # gopher
acl Safe_ports port 210     # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280     # http-mgmt
acl Safe_ports port 488     # gss-http
acl Safe_ports port 591     # filemaker
acl Safe_ports port 777     # multiling http
# Enable HTTP Connect
acl CONNECT method CONNECT


# Default Squid http_access settings
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager


# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost


# And finally deny all other access to this proxy
http_access deny all

# Squid normally listens to port 3128
http_port 3128
# Leave coredumps in the first cache dir (Default)
coredump_dir /var/spool/squid
# Default Refresh patterns
refresh_pattern ^ftp:       1440    20% 10080
refresh_pattern ^gopher:    1440    0%  1440
refresh_pattern -i (/cgi-bin/|\?) 0 0%  0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
refresh_pattern .       0   20% 4320
```

3. To test that the proxy is working correctly, using SSH, open a terminal on the Connector host machine and run the following set of commands:

```
# Running curl to cam should time out since the host should not be able to
route to the internet
$ curl cam.teradici.com
curl: (7) Failed to connect to cam.teradici.com port 80: Connection timed out
$ curl https://cam.teradici.com
curl: (7) Failed to connect to cam.teradici.com port 443: Connection timed
out
# Setting the proxy settings in the environment for curl to test that it
works for HTTP and HTTPS traffic
$ export http_proxy=http://<ip-of-proxy-server>:<proxy-port (default 3128)>
$ curl cam.teradici.com
<html>
<head><title>308 Permanent Redirect</title></head>
<body bgcolor="white">
<center><h1>308 Permanent Redirect</h1></center>
<hr><center>nginx</center>
</body>
</html>
$ export https_proxy=$http_proxy
$ curl https://cam.teradici.com
<!doctype html><html lang="en"><head><meta charset="utf-8"><meta
name="viewport" content="width=device-width,initial-scale=1,shrink-to-
fit=no"><meta name="theme-color" content="#000000"><link rel="manifest"
href="/manifest.json"><link rel="shortcut icon" href="/
favicon.ico"><title>Cloud Access Manager</title><link href="/static/css/main.
27391ea7.css" rel="stylesheet"></head><body><noscript>You need to enable
JavaScript to run this app.</noscript><div id="root" class="full-height"></
div><script type="text/javascript" src="/static/js/main.45a05db7.js"></
script></body></html>
# Clear the settings from the environment
$ unset http_proxy
$ unset https_proxy
```

4. To run the installer with the proxy settings, you can apply them in the environment:

```
# Installer will read proxy setting from environment if http_proxy,
https_proxy, HTTP_PROXY, or HTTPS_PROXY are set
$ export https_proxy=http://<ip-of-proxy-server>:<proxy-port (default 3128)>
$ ./cas-connector configure ...
```

or through the command line option:

```
$ ./cas-connector configure --https-proxy http://<ip-of-proxy-server>:<proxy-
port (default 3128)> ...
```

5. The installer should run as normal and configure the containers with the web proxy settings provided.

Proxy Passwords are not Supported

Proxy passwords are not supported with the Connector at this time.

Configuring the Active Directory for Cloud Access Connector

Teradici recommends having a single Active Directory configuration for a single deployment, which means all Connectors within that deployment should be configured to the same AD. If you want to have multiple Connectors with different Active Directory settings then you need to ensure that each Connector belongs to a separate deployment. If you create two Connectors that are associated with the same deployment then both will use the same Active Directory sync settings, and the configuration of the last Connector created will take precedence.

Configuring User and Computer Active Directory Distinguished Names

The Connector can optionally be configured to use specific Distinguished Names (DNs) when querying Active Directory for users and computers. This has been extended to be available when running the `update` command in addition to the `install` command.

The following is an example of the DN string format: `CN=CASM Admins,CN=Users,DC=example,DC=com` . You can also configure the frequency at which the Connector syncs this data with the CASM service, as outlined in the following table:

Flag	Type	Description
`--users-dn`	String	The base DN to search for users within Active Directory. This option may be specified multiple times to provide multiple DNs.
`--computers-dn`	String	The base DN to search for computers within Active Directory. This option may be specified multiple times to provide multiple DNs.
`--sync-interval`	String	The interval time in minutes for how often to sync Active Directory users and computers with the CASM service. It must be at least five minutes.

Flag	Type	Description
`--users-filter`	String	The filter to search for users within Active Directory. Specify multiple filters with multiple options. Default user filter: (&(objectCategory=person)(objectClass=user)). An example for a user group filter: (&(objectCategory=person)(objectClass=user)(memberOf:1.2.840.113556.1.4.1941:=CN=PCoIP Users Group,CN=Users,DC=example,DC=com)).
`--computers-filter`	String	The filter to search for computers within Active Directory. Specify multiple filters with multiple options. Default computer filter: (&(primaryGroupID=515)(objectCategory=computer)).

These flags outlined are optional and may be provided with the `install` or `update` commands. If you are updating a Connector you only need to provide these flags if you want to changing the DN settings associated with that Connector. If you do not add these flags when performing an update then the Connector will retain the same settings.

You can reset user or computer DNs to their default values by providing an explicit DN with a wider scope than the original DN used.

Configuring Active Directory Pool Groups

A set of command line flags enables users to update Active Directory pool groups. These flags apply changes to the Active Directory settings of the Connector.

By providing the following flags the appropriate update gets applied to the Connector settings. If no command-line option is provided, the Connector will display all available options for this operation.

Flag	Type	Description
`--cam-insecure`	String	Skips certificate validation when connecting to CAS Manager as a Service. This option should only be used when connecting to CAS Manager as a Service deployed with self-signed certificates.

Flag	Type	Description
`--add-pool-group`	String	Adds specified Active Directory group to the existing pool group settings. By providing all the existing pools groups in the Connector, settings would get replaced by the user specified ones.
`--remove-pool-group`	String	Removes specified pool Active Directory group by its DN. **This flag is not supported by the Connector on Rocky Linux/RHEL**.
`--clear-pools-groups`	String	Clears all pools Active Directory groups. This operation is exclusive and cannot be combined with `--remove-pool-group` or `--add-pool-group` . **This flag is not supported by the Connector on Rocky Linux/RHEL**
`--get-cam-settings`	String	Prints all CAS Manager as a Service settings to Admin console.

Floating Workstation Assignments

Floating workstation assignments is a feature of the Connector v78 or higher, which enables a user's entitlement to a workstation to be temporary. The remote workstation can be used by multiple users. Floating workstation assignments enables remote workstations that are part of a Remote Workstation Pool, to be assigned to a user for the duration of the PCoIP session. Once this session has been disconnected, the remote workstation will be automatically unassigned, and will be available for other users to connect.

This feature is useful for managing persistent remote workstations that are shared by multiple users and that have expensive software and applications, such as video editing, video proofing, etc. Multiple users can access the same remote workstation and utilize these applications. It can be used for project based remote workstations, where remote workstations are associated with projects instead of users. Teams can log into the project and access a specific remote workstation for that project. This also enables organizations to enforce logical separation of remote workstations.

The following sections outlines the steps involved in enabling this feature.

Create a Floating Pool

The next step is to create a floating pool group from the Admin Console.

1. Open the **Workstations Pools** page and click the **+** icon to create a new pool.
2. Select **Floating** for the workstation assignment policy, name the pool and click **CREATE**.
3. Click on the newly created pool from the Pools menu.
4. Click **ADD REMOTE WORKSTATIONS** to add workstations to the pool and click **SAVE**.

Remote Workstation Limit

There is a limit of 200 remote workstations in a floating pool. This feature will work with a larger number of remote workstations, but assignment timing may vary as a result.

Assign Users to the Pool

Once you have enabled session tracking and created and added remote workstations to your pool, you now need to add specific users. Only specified users can establish PCoIP sessions to remote workstations in the pool.

1. Click on the newly created pool from the Pools menu.
2. Click **ADD USERS** from the top menu, select the users you want to add and click **SAVE**.

Once you have completed these steps any user from the pool will be able to get any available remote workstation from the same pool on login. Once the PCoIP session has been disconnected, the remote workstation will automatically become available for future connections.

Session Disconnection

Please note that remote workstations will remain assigned to a user for **approximately 25 minutes** after the PCoIP session has been disconnected.

⚠ **Limited Support for PCoIP Agents for Linux**

PCoIP sessions to PCoIP Agents for Linux must be logged off before another user can connect. If the session is not logged off, the user will see a 6604 Error. If you observe this error, reboot the remote workstation.

Configuring the Broker Response Timeout

The following section outlines how to increase and set the *BROKER_MAX_WAIT_SECONDS* parameter for the Connector.

1. Make a copy of the *docker-compose.yaml* file by running the following command:

```
cp /opt/connector_data/docker-compose.yaml /opt/connector_data/docker-compose-new.yaml
```

2. Edit `/opt/connector_data/docker-compose.yaml` to add an environmental variable.

   - If you have an active Security Gateway, find the *cmsg* service section and add the following parameter and enter a value in seconds:

```
BROKER_MAX_RESP_WAIT_SECONDS=<required_value>
```

   - If you don't have an active Security Gateway, find the *cm* service section and add the following parameter and enter a value in seconds:

```
BROKER_MAX_RESP_WAIT_SECONDS=<required_value>
```

3. Update the Connector by running the following command:

```
cloud-access-connector update --compose-file /opt/connector_data/docker-compose-new.yaml
```

Transitioning Cloud Access Connectors

It is not possible to migrate directly from the Connector on Ubuntu to the Connector on RHEL/Rocky Linux as they run on different operating systems. The best method to transition to the Connector on RHEL/Rocky Linux is to create this new Connector using the same CAS Manager deployment you used for the Connector on Ubuntu.

> ⚠ **Minimize Transition Downtime**
>
> The Connector on RHEL/Rocky Linux can co-exist with the Connector on Ubuntu in the same CAS Manager deployment. To minimize downtime, it is recommended that you run the newly created RHEL/Rocky Linux Connector for a period of time to ensure it is working properly before decommissioning the Connector on Ubuntu.

MFA Configuration

When you enable MFA for the Connector for RHEL/Rocky Linux, all PCoIP Clients authenticated through the Connector will be prompted to enter MFA credentials. Previously with the Connector on Ubuntu, internal and external clients had different MFA configurations. If you want to have the same MFA configuration for the Connector on RHEL/Rocky Linux as the Connector on Ubuntu, you must install multiple Connectors on RHEL/Rocky Linux.

Transitioning the Connector

The following steps outline how to transition to the Connector on RHEL/Rocky Linux:

1. Before you install the Connector, ensure you have met all the required prerequisite steps. For instructions and documentation on the Connector prerequisite steps when install on RHEL/Rocky Linux, see Connector System Requirements. It is important to read and address all the prerequisites outlined.
2. Review the differences between the Connector on RHEL/Rocky Linux and Ubuntu, as outlined here.
3. Prepare your Connector configuration files. For information on configuring your Connector, see Configuring the Connector.

4. When generating the Cloud Access Connector token for the new Connector, ensure you use the same CAS Manager deployment as the existing Connector on Ubuntu. For information on generating the Connector tokekn, see Generating a Connector Token.
5. Install and configure the Connector on RHEL/Rocky Linux. For information on installing and configuring the Connector, see Installing the Connector.

Testing the Connector

Once you have installed the Connector on RHEL/Rocky Linux you should test it and ensure it has been correctly installed and configured. The following steps outline how to test the Connector:

1. Run the following command to check that all installed services are running:

```
sudo /usr/local/bin/kubectl get pods -c connector
```

2. Access the virtual machine where you have the PCoIP Client installed and establish a connection using the Connector IP or FQDN. If this connection is successful then it confirms the Connector has been installed correctly. If it is not, you should re-check the virtual machine configuration and Connector configuration.

Adding the Connector to a Load Balancer

If the original Connector on Ubuntu was configured with a load balancer, you need to add the new Connector on RHEL/Rocky Linux to the load balancer. Once you have tested the Connector on RHEL/Rocky Linux install, and are happy that it was successful, you can remove the Connector on Ubuntu.

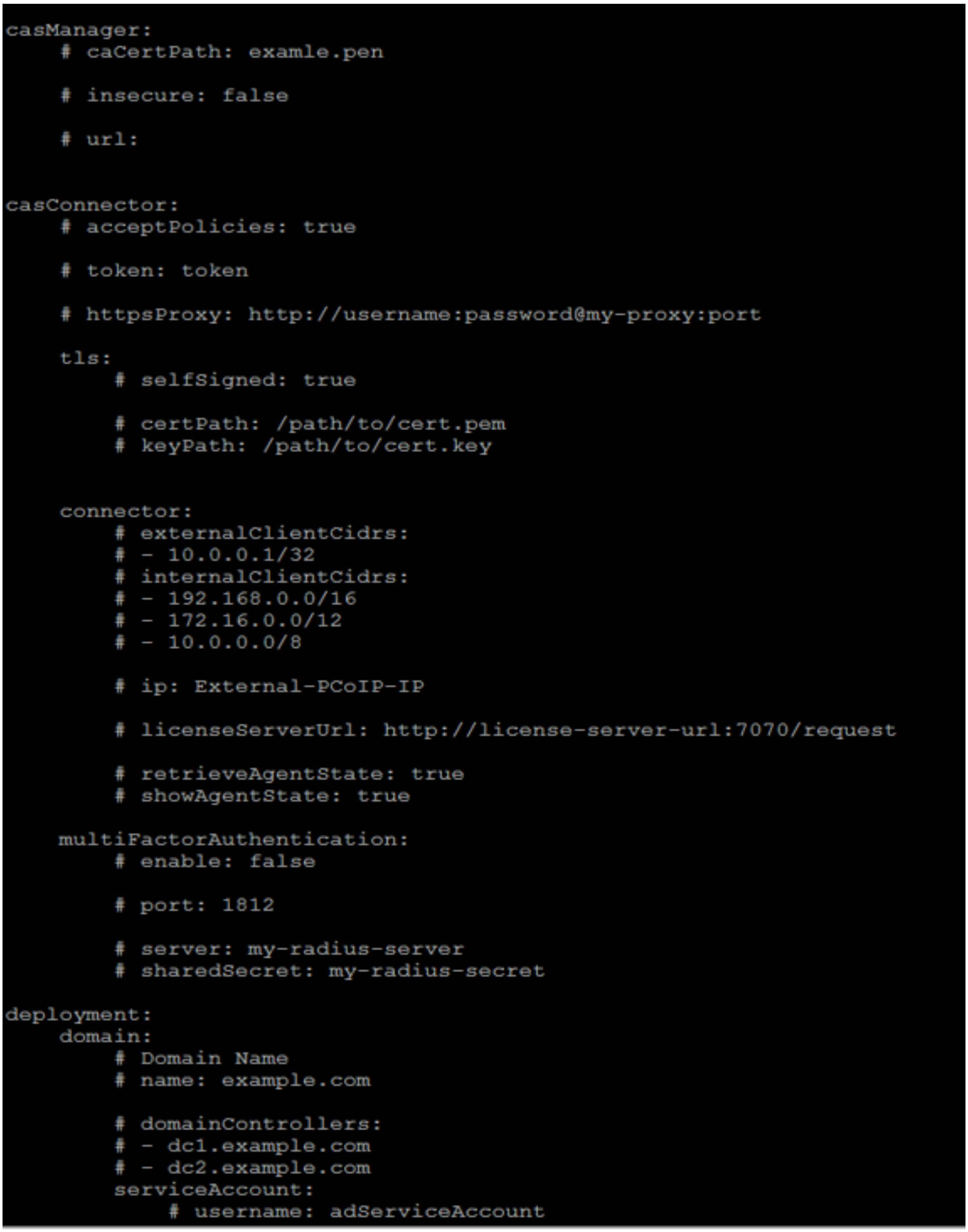
Configuring the Cloud Access Connector with Configuration Files

The following section outlines how to configure the Connector on RHEL/Rocky Linux using the configuration files that are downloaded as part of the Connector install. The Connector looks for two configuration files by default for configuration:

- `casc-secrets.yaml`
- `casc-configuration.yaml`

Both files can be found in the */etc/cas-connector/* directory. These files will be generated in the default directory when you install the Connector RPM. The `casc-configuration.yaml` file contains all of the installation configuration parameters that will persist after installation, the `casc-secrets.yaml` file contains sensitive information and data, such as passwords, that will be reset after a successful installation for security considerations.

It's recommended that you put sensitive data in the `casc-secrets.yaml` file instead of in the `casc-configuration.yaml` file or any custom configuration files so that they are not stored in plain text after installation. A sample configuration file is displayed below.



```
casManager:
    # caCertPath: examle.pen

    # insecure: false

    # url:


casConnector:
    # acceptPolicies: true

    # token: token

    # httpsProxy: http://username:password@my-proxy:port

    tls:
        # selfSigned: true

        # certPath: /path/to/cert.pem
        # keyPath: /path/to/cert.key


    connector:
        # externalClientCidrs:
        # - 10.0.0.1/32
        # internalClientCidrs:
        # - 192.168.0.0/16
        # - 172.16.0.0/12
        # - 10.0.0.0/8

        # ip: External-PCoIP-IP

        # licenseServerUrl: http://license-server-url:7070/request

        # retrieveAgentState: true
        # showAgentState: true

    multiFactorAuthentication:
        # enable: false

        # port: 1812

        # server: my-radius-server
        # sharedSecret: my-radius-secret

deployment:
    domain:
        # Domain Name
        # name: example.com

        # domainControllers:
        # - dc1.example.com
        # - dc2.example.com
        serviceAccount:
            # username: adServiceAccount
```

There are three available options for configuring the Connector:

- Default configuration files.
- Custom configuration files.
- A combination of default configuration files and command line flags.

⚠ **Required flags for Connector Configuration**

The following flags are mandatory, and must be included in the `casc-configuration.yaml` configuration file when configuring the Connector and running the `cas-connector configure` command:

- `--token`
- `--domain`
- `--sa-user`
- `--sa-password`
- `--ldaps-ca-cert`
- `--self-signed` (or you must provide `--tls-key` or `--tls-cert`)

Configuration using the Default Configuration Files

This option uses the default configuration files that were installed as part of the Connector RPM install process.

`casc-secrets.yaml` file reset

The `casc-secrets.yaml` file will be reset to its default values once you complete the configuration. This is a safety measure to ensure the secrets information is protected. A sample configuration file can be found in the */etc/cas-connector/* directory.

1. Configure both the configuration files in the */etc/cas-connector/* directory.
2. Run the following command to configure the Connector using the default configuration files:

```
sudo /usr/local/bin/cas-connector configure
```

Ensure that you use the options, flags or configuration parameters that best suit your system architecture and requirements. Ensure that you use the mandatory flags in the `casc-configuration.yaml` configuration file that are outlined in the note above. If required values are not provided on the command line or configuration file, you will be prompted for them. For additional flags, options or parameters, see Installation Flags and Options.

Configuration with Custom Configuration Files

This option uses custom configuration files to configure the Connector. You need to configure your custom configuration file and then run the command outlined below.

1. Configure the configuration files you wish to use.
2. Run the following command:

```
sudo /usr/local/bin/cas-connector configure --config-file
<PATH_OF_CONFIG_FILE>
```

Configuration with a Combination of Configuration Files and Flags

This option uses a combination of command line flags and configuration files. If the command line flags being used already exist in the configuration files, then the command line flag will take precedence.

1. Configure the configuration files you wish to use.
2. Run the following command:

```
sudo /usr/local/bin/cas-connector configure --config-file
<PATH_OF_CONFIG_FILE> <FLAGS>
```

Ensure that you use the mandatory flags in the `casc-configuration.yaml` configuration file that are outlined in the note above.

Difference between Cloud Access Connector on Ubuntu and RHEL/Rocky Linux

Cloud Access Connector on RHEL/Rocky Linux stands out from the Connector on Ubuntu. Here is a feature comparison between two Connectors.

Items	Connector on Ubuntu	Connector on RHEL/Rocky Linux
Operating System	Ubuntu 18.04	RHEL/Rocky Linux 8.x
Packaging	`tar` file	RPM Package
Deployment	Docker Swarm	Kubernetes
Connector Configuration	Command line flags and parameters.	Configuration files and/or command line flags and parameters.

Required Configuration Flags	`--token`
	`--domain`
	`--sa-user`
	`sa-password`
	`--accept-policies`
	`--self-signed` (or `--ssl-key` and `--ssl-cert`)
		`--token`
		`--domain`
		`--sa-user`
		`--sa-password`
		`--accept-policies`
		`--self-signed` (or `--tls-key` and `--tls-cert` must be provided)
		`--ldaps-ca-cert`
		`--computers-dn` (for the first Connector of the deployment)
		`--users-dn` (for the first Connector of the deployment)

MFA Configuration	When MFA is enabled all connection requests from internal PCoIP Clients will have MFA bypassed.	All connection requests both internal and external will require MFA credentials to be entered.

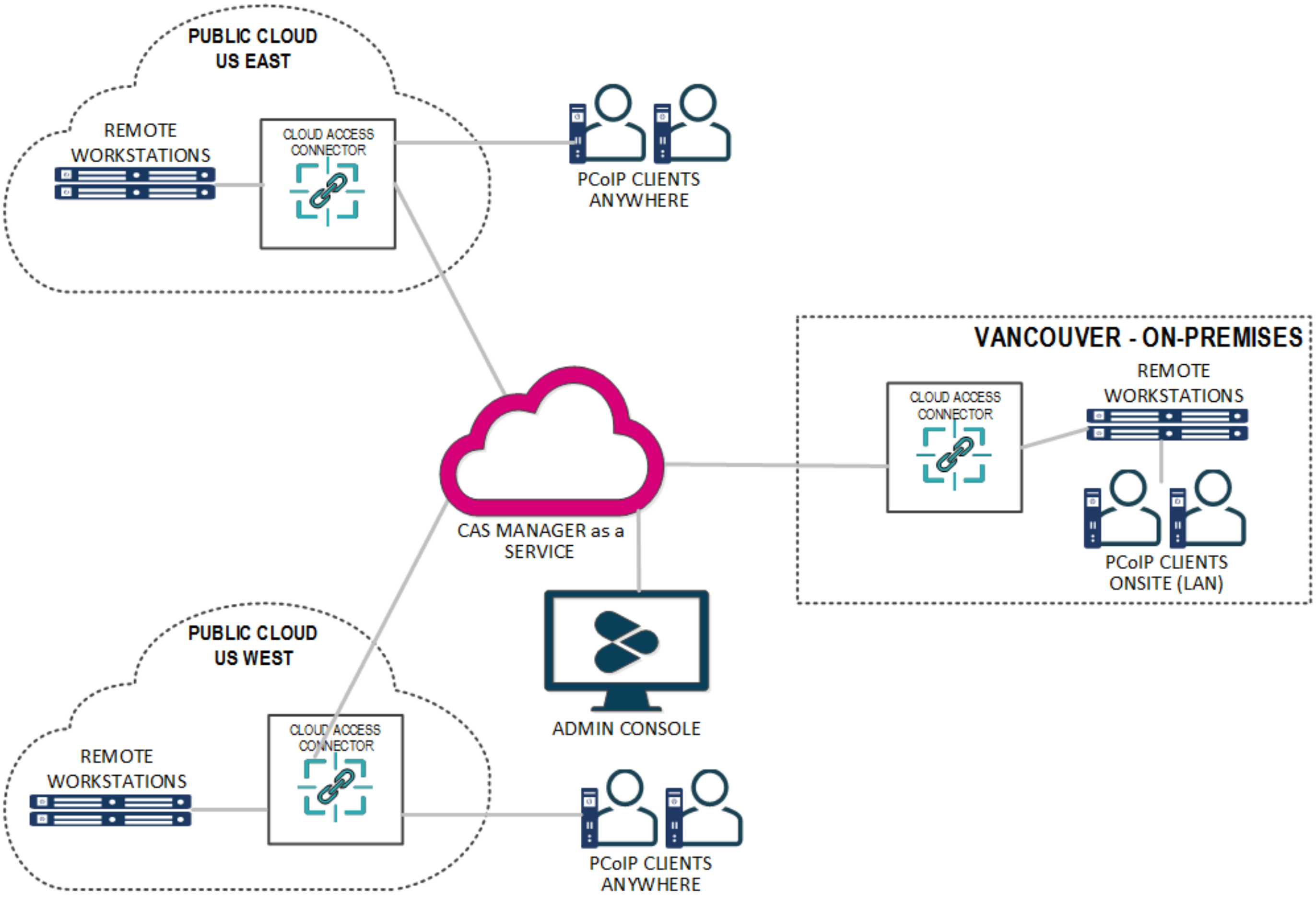
Items	Connector on Ubuntu	Connector on RHEL/Rocky Linux
AD Synchronization	If computer (`--computers-dn`) and/or users DNs (`--users-dn`) parameters are not provided during installation, and no AD configuration is returned from CAS Manager, AD sync will sync all objects from the AD to the CAS Manager.	If computer (`--computers-dn`) and/or users DNs (`--users-dn`) parameters are not provided during installation, and no AD configuration is returned from CAS Manager, AD sync will **not** sync all objects from the AD to the CAS Manager.
AD LDAPS Certificate	If `--ldaps-ca-cert` is not provided during installation, the AD CA certificate is automatically collected by the Connector by connecting to each DC on the LDAPS port, and the certificate is saved to the Connectors CA certificate store.	The AD CA certificate must be provided to the installer by entering the information with the `--ldaps-ca-cert` parameter or by editing the configuration file.
Diagnose Commands	You can diagnose remote workstation connectivity, and Active Directory connectivity by running the `diagnose` command.	In addition to the `diagnose` command, you can check the health of the Connector by running the `--health` command, and can create a support bundle by running the `--support-bundle` command.
Key path and Certificate path flag	`--ssl-key` and `--ssl-cert`	`--tls-key` and `--tls-cert`

Items	Connector on Ubuntu	Connector on RHEL/Rocky Linux
Installation Commands	Download the installer from teradici.com and extract the package, then run the install command with the required flags: `sudo /usr/sbin/cloud-access-connector install {flags}`. The installer will then prompt for mandatory flags if you do not provide them in the command.	Add the repository and install the Connector RPM with the following command: `sudo yum install -y cas-connector`. Configure the Connector with flags or configuration files using the following command: `sudo /usr/local/bin/cas-connector configure {flags or path to config file}`. The `configure` command will fail with a missing parameter error if the mandatory flags or parameters are missing, the mandatory flags are `--token`, `--domain`, `` `-sa-user ``, `--sa-password`, `--ldaps-ca-cert`, `--self-signed` (or must provide `--tls-key` and `--tls-cert`).
Update Configuration Files	`cd /usr/sbin sudo cloud-access-connector update {flags to be updated}`	Update the configuration file or specify flags to be updated: `sudo /usr/local/bin/cas-connector configure {flags or path to config file}.`
Upgrade Commands	`cd /usr/sbin sudo cloud-access-connector update {flags to be updated}`	`sudo dnf update cas-connector` and `sudo /usr/local/bin/cas-connector upgrade`
Internal/External Session Detection	Typically the Connector on Ubuntu will work without any special configuration, but in some cases you may need to explicitly set the `--internal-client-cidr` and `--external-client-cidr` so that sessions get treated correctly (eg, NATing external connections from a Firewall).	In some cases the Connector on RHEL/Rocky Linux will work without any special configuration, but if you know the Connector on RHEL/Rocky Linux is only for WAN connections, it is strongly recommended to set the `--internal-client-cidr` to `""` to avoid issues. In some cases where the Connector on Ubuntu worked without configuring this setting, you may now be required to do so (eg, GCP Network Load Balancer).

Overview

The Admin Console enables you to create deployments, connectors and remote workstations all within a single console and from a single interface (UI). You can track all these components from the interface of the console, as well as monitor and manage all aspects of your deployment infrastructure. You can access support, release notes and get service status information from the Admin Console also. The Admin Console works with both CAS Manager, and CAS Manager as a Service.

The diagram below outlines a connection workflow for a cloud deployment using the Admin Console with CAS Manager as a Service.



Connecting to the Admin Console

The following section outlines how to access and connect to the Admin Console for CAS Manager, and CAS Manager as a Service.

Connecting to CAS Manager

Once you have unlocked the Admin Console, open a web browser and go to https://public-or-private-ip-address-of-cas-manager to login with the default "adminUser". If you have configured multi-admin support, login with your enterprise identity provider account that has the required admin permission for CAS Manager.

Connecting to CAS Manager as a Service

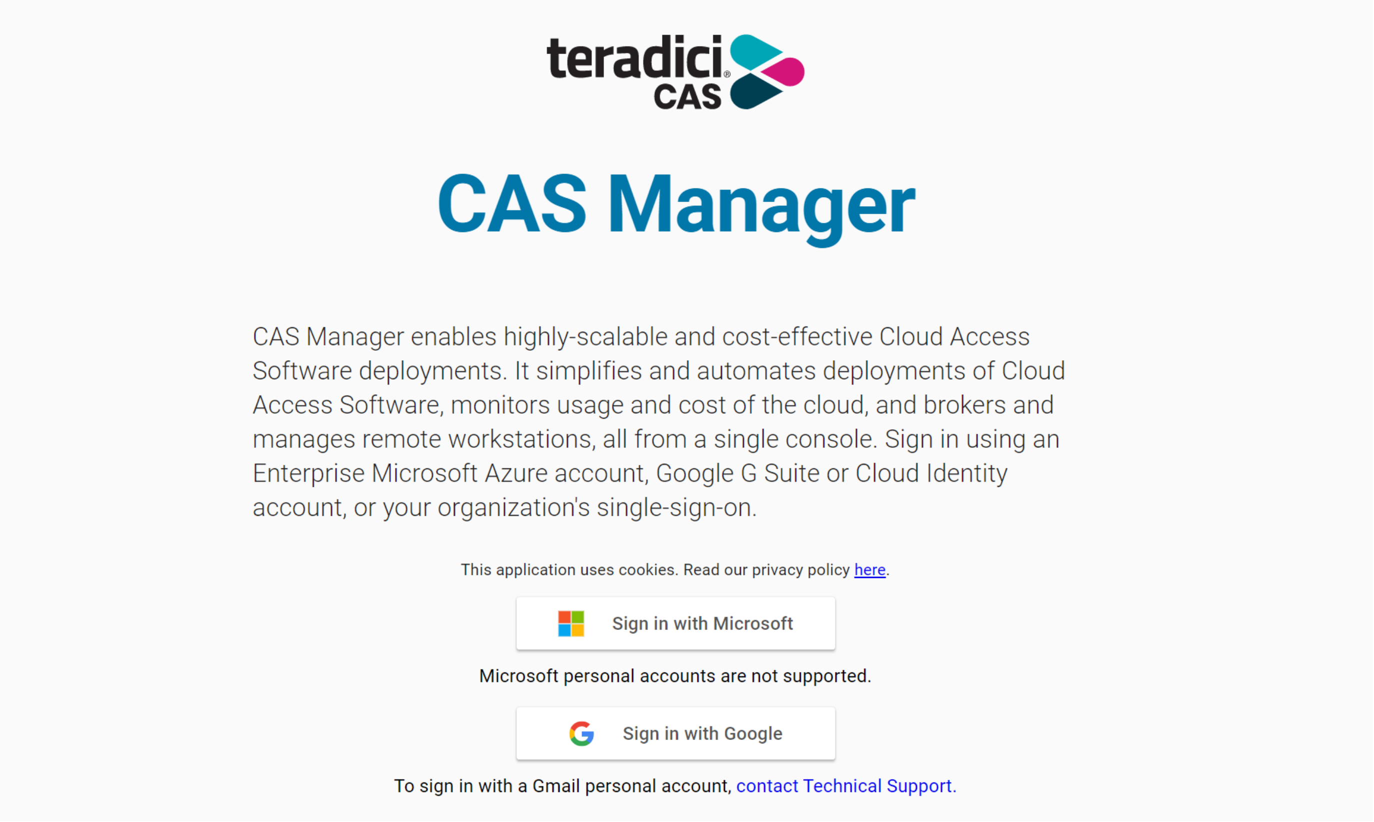
Go to the Admin Console login page and log in with your Enterprise Microsoft Azure account, or if you are logging in through Google, a G Suite or Cloud Identity account. Enter your credentials to access the Admin Console.

Email Account Support with CAS Manager

CAS Manager supports two types of email accounts:

- Company email accounts registered with Google G Suite.
- Company email accounts registered with Microsoft Azure Active Directory services. For more information on this account type, see Microsoft Azure Active Directory Authentication.

Personal Gmail accounts are not supported by default and need to be approved by Teradici before being used. For access to CAS Manager with a personal Gmail account, contact Teradici support. CAS Manager as a Service does not support Microsoft personal email accounts.



CAS Manager

CAS Manager enables highly-scalable and cost-effective Cloud Access Software deployments. It simplifies and automates deployments of Cloud Access Software, monitors usage and cost of the cloud, and brokers and manages remote workstations, all from a single console. Sign in using an Enterprise Microsoft Azure account, Google G Suite or Cloud Identity account, or your organization's single-sign-on.

This application uses cookies. Read our privacy policy here.

Microsoft personal accounts are not supported.

To sign in with a Gmail personal account, contact Technical Support.

If you encounter issues logging into the Admin Console, it could be for one of the following reasons:

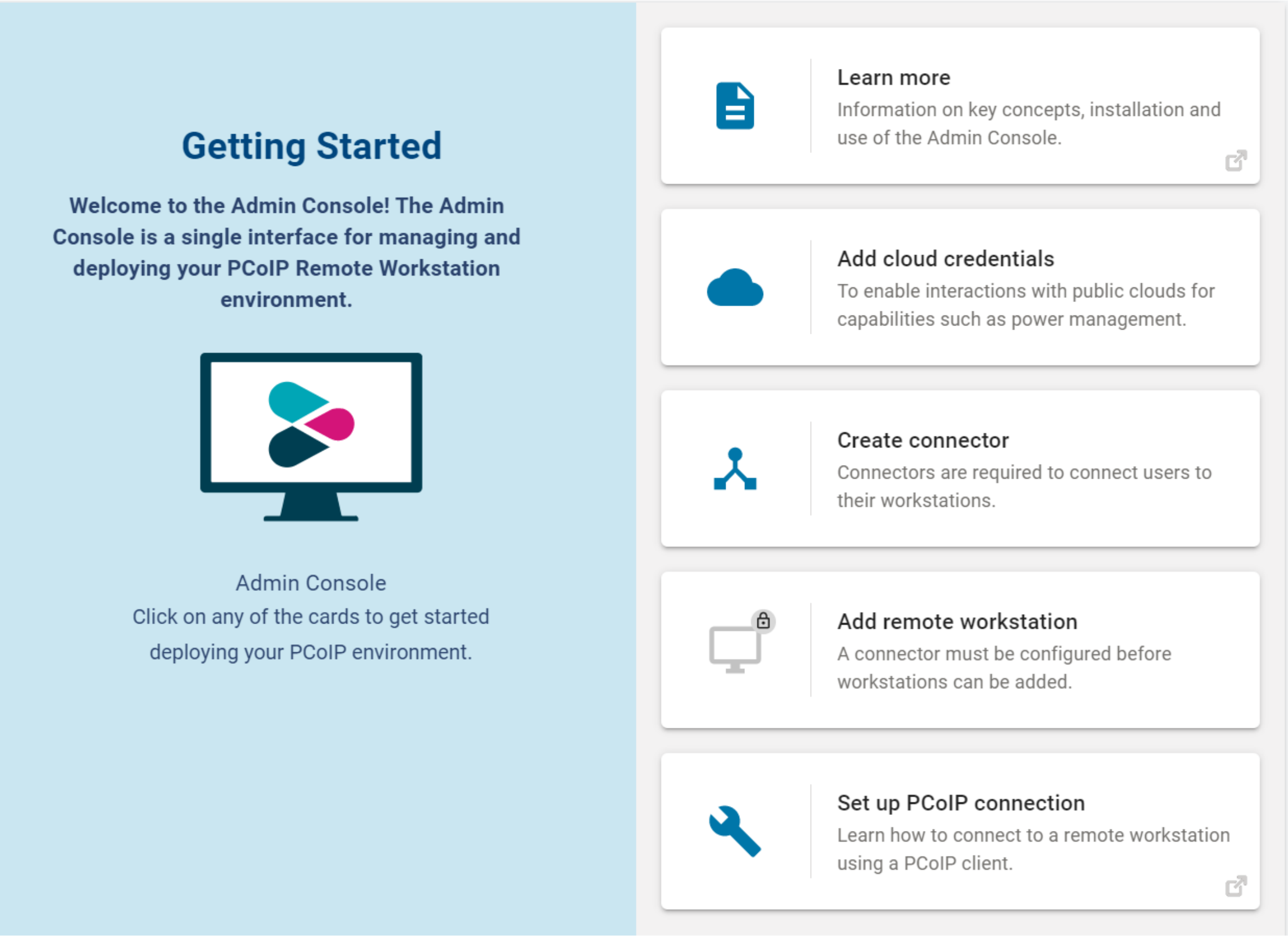
- The account being used is a personal account and has not been approved by Teradici.
- Cookies have been blocked on https://cam.teradici.com/.
- Pop-ups have been blocked on https://cam.teradici.com/.

If you continue to experience issues logging into the Admin Console, contact Teradici Support.

Admin Console Dashboard

Once you log into the Admin Console you will see the dashboard page. This dashboard acts as a quick-start guide which points to where you can create deployments, create Connectors, add remote workstations as well as provide links to useful information within the CAS Manager documentation.

You can return to the dashboard page at any time by clicking the **Dashboard** option from the console sidebar.



Getting Started

Welcome to the Admin Console! The Admin Console is a single interface for managing and deploying your PCoIP Remote Workstation environment.

Admin Console
Click on any of the cards to get started deploying your PCoIP environment.

Learn more
Information on key concepts, installation and use of the Admin Console.

Add cloud credentials
To enable interactions with public clouds for capabilities such as power management.

Create connector
Connectors are required to connect users to their workstations.

Add remote workstation
A connector must be configured before workstations can be added.

Set up PCoIP connection
Learn how to connect to a remote workstation using a PCoIP client.

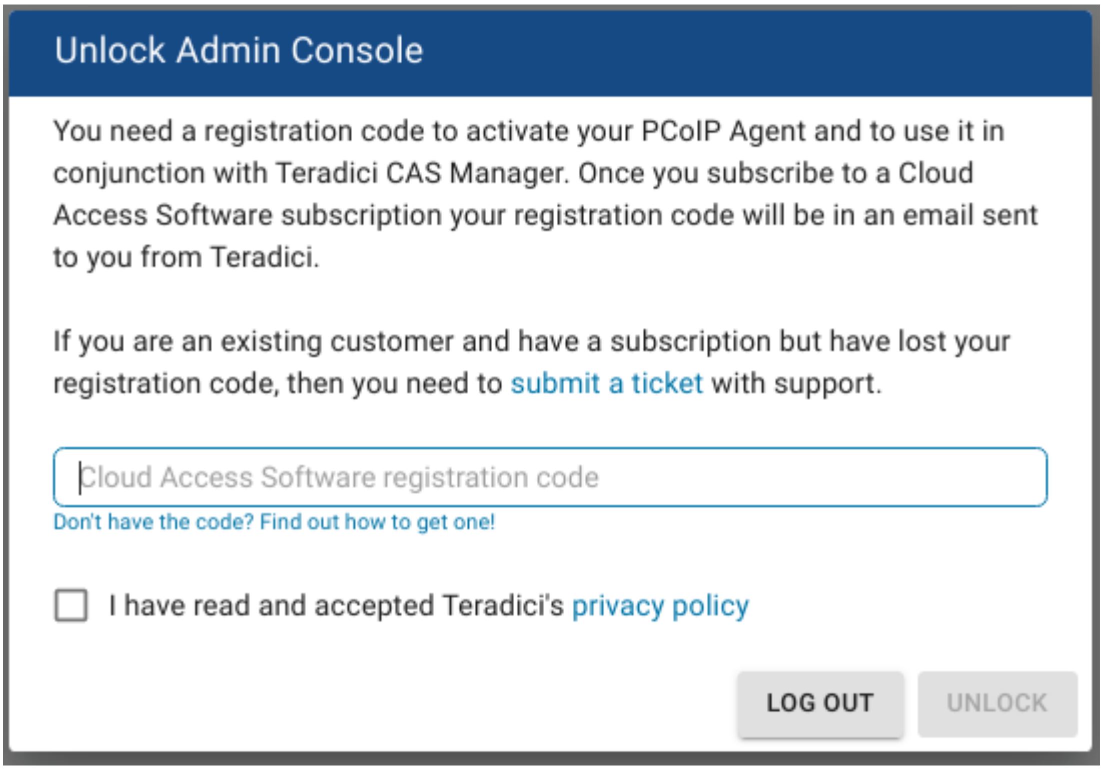
Configuring the Admin Console

On the **Deployments**, **Connectors** and **Remote Workstations** pages you can control which columns are visible and in which order they appear for the listed resources. To change your column options, select COLUMNS from the page heading and select which columns you wish to make visible. The format you select will be preserved when you log back into the Admin Console.

Managing Deployments

The following section outlines how to create a deployment using the Admin console:

1. If you do not have any existing deployments (first time log-in) you will be prompted to enter your CAS Software registration code. Once you enter the code it will automatically generate your first deployment and take you to the **Edit Deployment** page.

   Unlock Admin Console

   You need a registration code to activate your PCoIP Agent and to use it in conjunction with Teradici CAS Manager. Once you subscribe to a Cloud Access Software subscription your registration code will be in an email sent to you from Teradici.

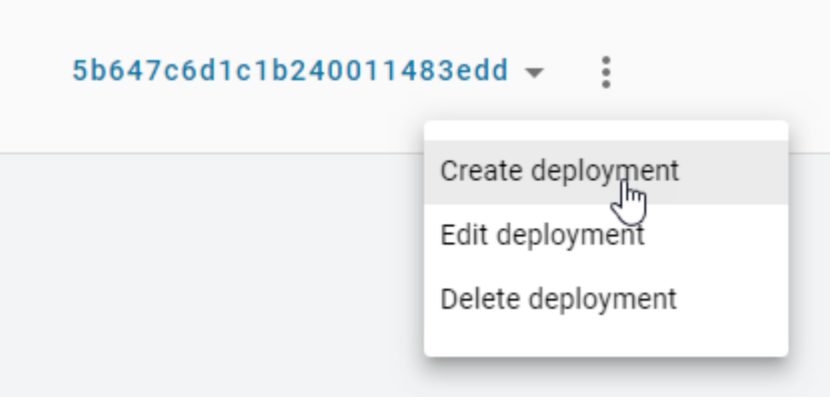
   If you are an existing customer and have a subscription but have lost your registration code, then you need to submit a ticket with support.

   Cloud Access Software registration code

   Don't have the code? Find out how to get one!

   □ I have read and accepted Teradici's privacy policy

2. If you have existing deployments you can click **Create deployment** from the kebab options at the top of the page to take you to the **Create Deployment** page.

3. Enter the following information:
    - Enter the deployment name.
    - Enter your PCoIP registration code. Please store this code in a secure location as it cannot be retrieved later.
    - Click **CREATE**.

The deployment has now been created and you can edit the deployment by configuring deployment service accounts, cloud service accounts and Connector settings.

Cloud Service Accounts

You can now enter cloud service account credentials for AWS, Azure and GCP if you are working in those environments and want to enable CAS Manager to perform certain functions, such as power management. If you are not using AWS, Azure, and GCP then you do not need to enter this information.

> **Cloud Service Account Credentials**
>
> These credentials are used in places where the CAS Manager as a Service interacts with your cloud environment to perform actions such as powering a remote workstation on or off. If credentials are not provided, remote workstations in that cloud can still be added to CAS Manager as a Service and users can still be entitled to the remote workstation and start a PCoIP session, but CAS Manager as a Service cannot perform functions such as power on and off.

Entering these credentials is optional and enables you to access extra functionality and control over the remote workstations within the deployment on the cloud provider of your choice.

> ⚠ **Domain Controllers in a Single Deployment**
>
> You cannot deploy multiple Connectors against different Domain Controllers within the same deployment. This will cause the Connectors to crash.

AWS Cloud Credentials

The following sections outline how to managed and configure AWS cloud information for CAS Manager and CAS Manager as a Service. Please note the permissions required for CAS Manager as a Service are different to the permissions for CAS Manager.

AWS Cloud Credentials for CAS Manager

To configure AWS Cloud Credentials for CAS Manager, see the AWS Configuration section.

AWS Cloud Credentials for CAS Manager as a Service

Through the Admin Console you can generate a CAS Manager Account ID and External ID that can be used when creating an AWS role through the AWS Management Console. The following steps outline how to generate a CAS Manager Account ID and External ID:

1. In the Admin Console select the deployment you wish to use.
2. Click **Edit Deployment**.
3. Click **Cloud Service Accounts**.
4. Select AWS and click **Generate**. Ensure you copy the CAS Manager Account ID and External ID and save them to your clipboard.

> ⚠ **AWS Role Creation and Permission Policy**
>
> You must create a role in your AWS account which CAS Manager as a Service is able to assume. You must use the Account ID and External IDs when creating the AWS role. For more information on creating roles in AWS, see here.

Once you have entered the CAS Manager Account ID and External ID and created the AWS role, you will need to create a permissions policy for CAS Manager as a Service that contains the following permissions:

- **Service**: EC2
- **Actions**:
    - List: DescribeInstances
    - Write: RebootInstances StartInstances StopInstances TerminateInstances

There are additional permissions needed to verify that the role has all the required permissions before being added to a deployment:

- **Service**: IAM
- **Actions**:
    - Read: GetUser SimulatePrincipalPolicy

The following is an example of how the permissions set should look in a JSON format:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:RebootInstances",
                "ec2:DescribeInstances",
                "ec2:TerminateInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "iam:GetUser",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

If the user tries to add an AWS role that doesn't have these permissions, CAS Manager as a Service will still add the role but will not validate that it has the required permissions. You can now associate a permissions policy to this role.

1. Once you have created the role in AWS, copy and paste the role ARN and enter it into the Role ARN field in the Admin Console.
2. Click **Submit**.

For information on the AWS Service Account roles and permission policies with CAS Manager as a Service, see here.

Azure Cloud Credentials

For Azure you need to enter the Tenant ID, Subscription ID, Client ID and Client Secret.

For information on how to create a new Client Secret from Azure, see here.

> ⚠ **Azure Client Secret**
>
> Once you generate the client secret you need to copy it straight away as it will not be available again from Microsoft. If you have an expired client secret you need to delete it and then create a new secret and assign it to that deployment.

For information on the Azure Service Account and permission requirements with CAS Manager, see here.

GCP Cloud Credentials

You can enable GCP cloud credentials by entering the GCP client email, Project ID and Private Key and clicking **Submit**. You can also upload the JSON Key file with the GCP cloud credentials.

For more information on GCP Cloud Service Accounts with CAS Manager, see here.

Editing an Existing Deployment

The creation date, computer and users DNs and the interval time in minutes that it syncs with the Active Directory for the deployment are also displayed when you go to edit a specific deployment.

You can search for specific deployments by name by using the search bar in the table toolbar.

You can edit the deployment name, update the registration code and GCP or Azure cloud service account credentials of an existing deployment through the Admin Console. A menu item has been added to the table toolbar that enables you to create, edit, delete and view all existing deployments:

1. Click the dropdown menu from the top of the page and select the deployment.
2. Select the deployment and click the kebab option under the **ACTIONS** column to edit the deployment.

3. Update the deployment name, registration code, GCP or Azure credentials and then click SAVE.

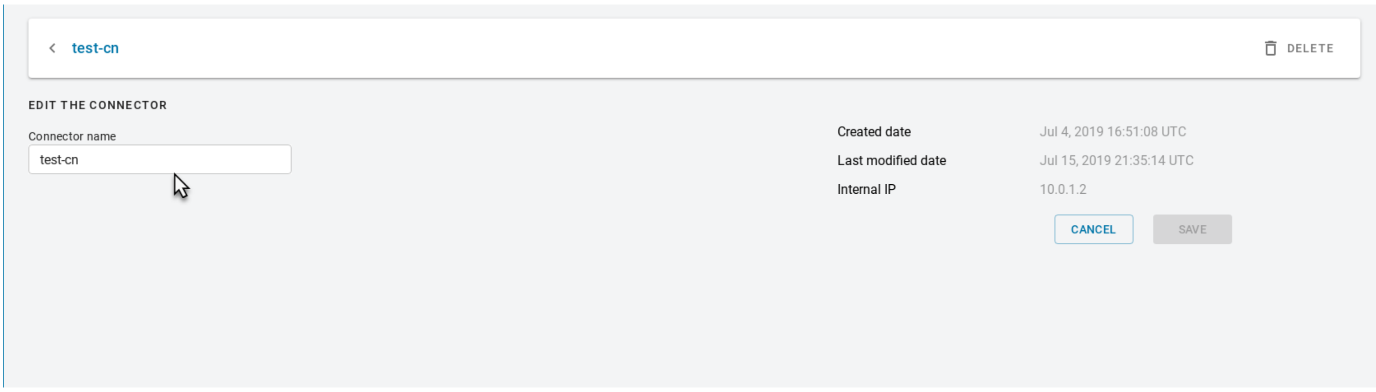
The updated information and credentials will now be associated with this deployment.

Editing a Cloud Access Connector

Once you have created a Connector you can edit its name by clicking on the Connector directly from the **Connectors** page or by clicking on **Edit** from the kebab associated with it on the **Connectors** page.

You can search for specific Connectors by name by using the search bar in the table toolbar.

Enter the new name and click **Save**.



EDIT THE CONNECTOR

Connector name: test-cn

Created date	Jul 4, 2019 16:51:08 UTC
Last modified date	Jul 15, 2019 21:35:14 UTC
Internal IP	10.0.1.2

Domain Controller Certificates

If all DC certificates have expired, the Cloud Access Connector will stop working. An error indicator will display on the **Connectors** page when a Cloud Access Connector has a DC with expired certificates.

A warning indicator that details the current state of the DC certs will display on the same page when a Cloud Access Connector has a certificate that less than a week away from expiring.

Cloud Access Connector - Troubleshooting

If there is an issue installing the Cloud Access Connector or an existing Connector is failing, please see the troubleshooting section on Cloud Access Connector Connectivity. Within this section there are steps to check the following:

- Remote Workstation connections
- Active Directory connections
- Cloud Access Connector component information

You can also view the following KB article here which provides a list of troubleshooting steps for common issues related to installing the Connector. For information on installer errors related to a change in the distribution system, see Installer Issues.

Provisioning a Remote Workstation

The following section outlines how to provision a remote workstation using the Admin Console.

Pre-Defined Images and Templates

If you wish to use your own custom images or templates, you must create and manage those outside of CAS Manager and create your remote workstation outside of CAS Manager also. Once you have created a remote workstation you can add it to your deployment in CAS Manager for brokering and management.

Before provisioning a remote workstation you need to ensure that the Active Directory domain is correctly configured. This should be a different AD Service Account to the account used when installing the Connector. The AD Service Account needs to have the following permissions:

- Create Computer Objects
- Delete Computer Objects

The permissions on the Computer Objects must be set to:

- Read All Properties
- Write All Properties
- Read Permissions
- Modify Permissions
- Change Passwords
- Reset User Passwords
- Validated write to DNS host name
- Validate write to service principal name

These permissions are required so that the remote workstations are able to join the domain account. Without these permissions the remote workstation will still be provisioned, but there will be an issue when adding it to the domain.

Permissions to Create and Delete Computer Objects

The following section outlines how to add permissions to create and delete computer objects through the OU permissions dialog:

1. Go to the security tab of the OU you want to give permissions to.
2. Right-click the relevant OU and click **Properties**.
3. Go to the security tab and click **Advanced**.
4. Click **Add** and browse to your user account. As stated above you need to add the user account to the OU.
5. Select **This object and all descendant objects** and select the following permissions:
    - Create Computer Objects
    - Delete Computer Objects
6. Click **OK**.

Permissions on the Computer Objects

The following section outlines how to select permissions on the computer objects through the OU permissions dialog:

1. Go to the security tab of the OU you want to give permissions to.
2. Right-click the relevant OU and click **Properties**.
3. Go to the security tab and click **Advanced**.
4. Click **Add** and browse to your user account. As stated above you need to add the user account to the OU.
5. Limit the **Apply Onto** scope to **Descendant Computer objects** and select the following settings:
    - Read All Properties
    - Write All Properties
    - Read Permissions
    - Modify Permissions

- Validated write to DNS host name
- Validated write to service principal name

6. Click OK.

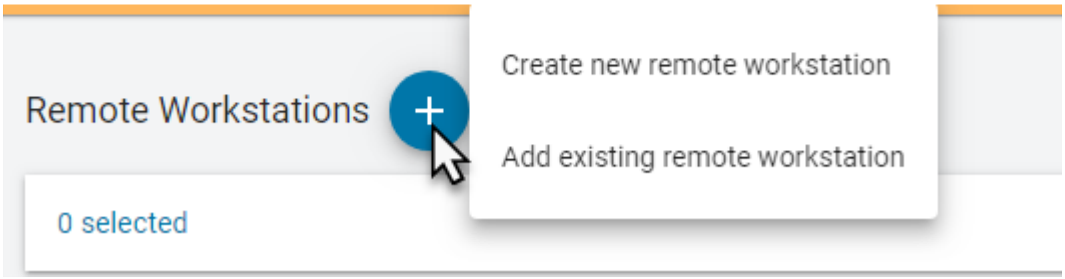
The validated write to DNS host and service principal name permissions are required so that the DNS record for a remote workstation can be created after it is domain joined.

For information on which Cloud Service accounts can perform certain features, please consult the Service Account Requirements section.

Provisioning a Remote Workstation

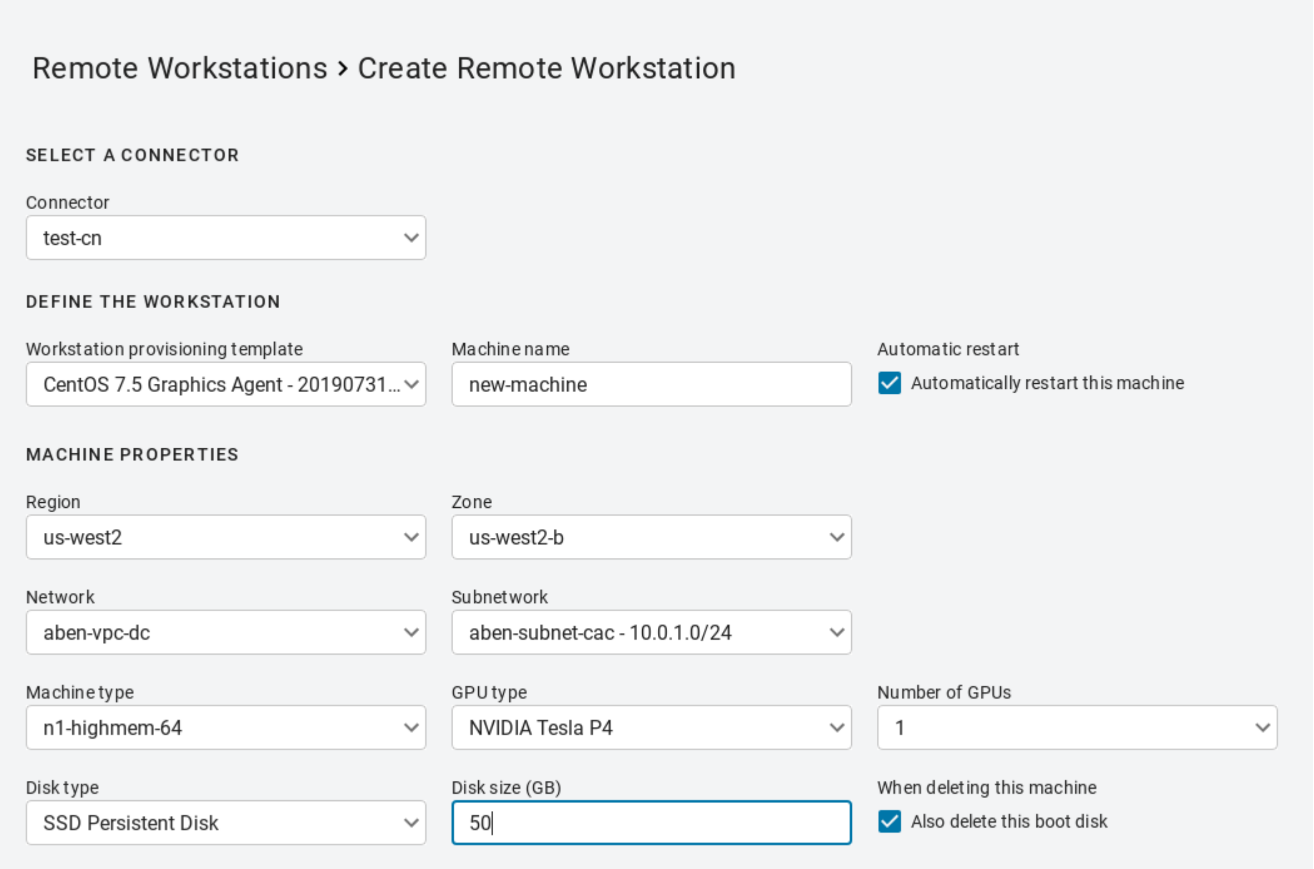
You must have a valid cloud service account to enable this feature. The following steps outline how to provision a remote workstation:

1. Click **Workstations** from the Admin Console sidebar.
2. Click **Create new remote workstation** from the add remote workstation icon.


Remote Workstations
+
Create new remote workstation
Add existing remote workstation
0 selected

3. Select an existing Connector from the dropdown menu.
4. Select a provisioning template from the dropdown menu and give your remote workstation a machine name. You can also choose whether you want to enable an automatic restart of the workstation. Compute engine can automatically restart remote workstation instances if they are terminated for non- user intitiated reasons, such as maintenance events, hardware failures, software failures, etc.

5. Enter the remote workstations network, region and disk properties. An example of what this information may look like is shown below:



Remote Workstations > Create Remote Workstation

SELECT A CONNECTOR

Connector: test-cn

DEFINE THE WORKSTATION

Workstation provisioning template: CentOS 7.5 Graphics Agent - 20190731...

Machine name: new-machine

Automatic restart: ☑ Automatically restart this machine

MACHINE PROPERTIES

Region: us-west2

Zone: us-west2-b

Network: aben-vpc-dc

Subnetwork: aben-subnet-cac - 10.0.1.0/24

Machine type: n1-highmem-64

GPU type: NVIDIA Tesla P4

Number of GPUs: 1

Disk type: SSD Persistent Disk

Disk size (GB): 50

When deleting this machine: ☑ Also delete this boot disk

Public IP or Cloud NAT Requirement

Provisioning will fail unless the machine has a public IP or Cloud NAT.

Remote Workstation Machine Name

Due to NetBIOS and a Windows limitation, the remote workstations machine name must be 15 characters or less. Failure to do this may result in issues with your remote workstation connection.

6. Enter the Active Directory information for the remote workstation. The service account must have permission to join computers to the domain.
7. Once you have entered all required information, click **CREATE**.

The remote workstation will now appear in the table of available machines on the **Workstations** page.

Active Directory Information

Active Directory information is only used during provisioning to join the remote workstation in question to the domain. This information is not saved by the CAS Manager. The remote workstation is joined to the active directory domain configured in the Connector.

Metadata Retrieval and Storage Information

All provisioned remote workstations have `--metadata enable-guest-attributes=TRUE` set. This is set to facilitate the passing of data at provisioning time. For more information, see https://cloud.google.com/compute/docs/storing-retrieving-metadata.

IdleShutDown Agent Configuration

IdleShutDown Agent is configured so that the remote workstation will shutdown when it is idle. For more information on installing and configuring this feature, see Configuring Idle Shutdown.

Workstation Pools

You can create workstation pools within the CAS Manager Admin Console. A workstation pool is a group of remote workstations. To simplify user access management, a user group or individual users can simply be assigned to a workstation pool.

A floating pool is a remote workstation pool that uses a floating workstation assignment policy. With this pool, a user is entitled to a pool and its remote workstations rather than a single remote workstation. When a user is assigned a remote workstation, this assignment is ephemeral. Once the user disconnects from the remote workstation there is a holding time where it will still be assigned to that user. This holding time can be configured by an admin user. Once this holding period expires, the user is unassigned to the remote workstation and it is added back to the pool and is available for re-assignment.

Once you log in to the CAS Manager Admin Console, you are automatically assigned a remote workstation within the pool, depending on the assignment policy of the pool. When creating a remote workstation pool, the assignment policy can be selected.

Use Cases for Floating Workstation Assignment Policy

The following section outlines potential use cases for the floating workstation assignment policy:

Two user's and two Windows workstations

- User 1 attempts to log in with a PCoIP Client.
- Windows workstation 1 is successfully assigned to User 1.
- User 2 attempts to log in with a PCoIP Client.
- Windows workstation 2 is successfully assigned to User 2.

Two user's and a single Windows workstation

- User 1 attempts to log in with a PCoIP Client.
- Windows workstation 1 is successfully assigned to User 1.
- User 1 closes the PCoIP Client.

- User 2 then attempts to log in and is presented with an error message stating "Resource does not exist in CAM Service".
- User 2 attempts to log in 25 minutes later. The assignment holding time for this example is 20 minutes, which is the default minimum assignment holding.
- Windows workstation 1 is successfully assigned to User 2. The Windows session is taken over by User 2.

These use cases also apply for RHEL/CentOS workstations.

The following section outlines how to create a floating workstation pool from the CAS Manager Admin Console:

Creating a Workstation Pool

You can add multiple workstation pools to specific deployments. Each workstation pool will list the remote workstations, users and user groups within that pool. The following steps outline how to create a workstation pool, and choose the floating pool assignment policy:

1. Click on **Workstation Pools** from the CAS Manager Admin Console sidebar.
2. Click the **+** icon to create a new workstation pool.
3. Name the pool and choose the **Floating** option for the workstation assignment policy. The workstation assignment policy determines how workstations belonging to the workstation pool are assigned to users when they log in. This assignment policy can not be modified after the workstation pool has been created.
4. Enable **Session Tracking** by selecting the toggle.
5. Add the workstation holding time in minutes. Once this holding period expires, the user is unassigned to the remote workstation and it is added back to the pool and is available for re-assignment.
6. Name the pool and click **CREATE**.

There are two possible options for the workstation assignment policy:

Persistent: This is the default policy. Once a user logs in they are automatically and persistently assigned a remote workstation within the pool.

Floating: With the floating policy, once a user disconnects their PCoIP session, the remote workstation will be automatically unassigned from the user, and the remote workstation will become available for other users to connect to.

Adding Remote Workstations to a Workstation Pool

Remote workstations and users within a workstation pool are a subset of the available remote workstations and users within a specific deployment. As a result of this, you will only be able to add remote workstations and users that have already been created in CAS Manager. The following steps outline how to add remote workstations to a workstation pool:

1. Click on **Workstation Pools** from the CAS Manager Admin Console sidebar.
2. Select the workstation pool you created in the previous section to display the edit pools page.
3. Click **Add Remote Workstations** and add remote workstations to the pool.

Adding Users to a Workstation Pool

Only specified users can establish PCoIP sessions to remote workstations in the workstation pool. If a remote workstation is available (not assigned to a user) it will be automatically assigned to the user. The following steps outline how to add users to a workstation pool:

1. Select the workstation pool and click **Add Users/Add Groups**.
2. Search for the users you want to add, select them and click **SAVE**.

Once users and remote workstations are added to the workstation pool, users from the workstation pool get available workstations upon log in. Once the PCoIP Session is disconnected, the remote workstation will become automatically available for future connections or continue to be assigned to the user depending on the workstation pool assignment policy.

Features and Known Limitations

There are certain limitations associated with this feature, as outlined in the following list:

- This feature is only supported in Connector(s) version 78 or higher.

- If all remote workstations have been assigned, and no remote workstations are available, users will see the following error during session establishment "Resource does not exist on CAM Service".
- Remote workstations will remain assigned to a user for approximately 20 minutes after the PCoIP session has been disconnected by default. This time period is configurable through assignment holding time, and has to be longer than 20 minutes.
- The current limit is 200 remote workstations in a floating pool. The feature will work with a larger number of remote workstations, but the assignment timing may vary.
- Limited support for Linux Agents: When establishing a PCoIP session to Linux Agents, the session must be logged off before another user can connect. If the session is not logged off, the user will see a 6604 error message. To resolve this error reboot the remote workstation. This issue is being worked on.
- When connecting to a PCoIP Agent for Windows, if a previous user has been connected, the other user will see the Windows Switch Users screen. They will then be prompted to enter their credentials again before accessing the desktop.
- For Windows remote workstations that are power managed through CAS Manager, the Idle Shutdown Service for the PCoIP Agent can be installed and configured. Once this service is installed, the remote workstation will automatically power off after the PCoIP session has been disconnected. During PCoIP session establishment, the remote workstation will be powered on. For information on installing and configuring the Idle Shutdown Service, see below.

Idle Shutdown Service

The following section outlines the steps to install and configure the Idle Shutdown Service for each Windows remote workstation:

1. Connect to the remote workstation via the PCoIP Client. Ensure you have admin permissions.
2. Copy the following PowerShell script:

```
$idleTimerRegKeyValue = <idle-time-in-minutes>
$enableAutoShutdown = <$true-or-$false>

# Detect agent type
$is64 = $false
```

```
$serviceName = "CAMIdleShutdown"
$path = "C:\Program Files (x86)\Teradici\PCoIP Agent\bin"
if (!(Test-Path -path $path))  {
    $path = "C:\Program Files\Teradici\PCoIP Agent\bin"
    $is64 = $true
}
cd $path

# Install Service
$ret = .\IdleShutdownAgent.exe -install
# Check for success
if( !$? ) {
    $msg = "Failed to install {0} because: {1}" -f $serviceName, $ret
    Write-Host $msg
    throw $msg
}

# Configure Service
$idleTimerRegKeyPath =
"HKLM:SOFTWARE\WOW6432Node\Teradici\CAMShutdownIdleMachineAgent"
if ($is64) {
    $idleTimerRegKeyPath =
"HKLM:SOFTWARE\Teradici\CAMShutdownIdleMachineAgent"
}
$idleTimerRegKeyName = "MinutesIdleBeforeShutdown"
if (!(Test-Path $idleTimerRegKeyPath)) {
    New-Item -Path $idleTimerRegKeyPath -Force
}
New-ItemProperty -Path $idleTimerRegKeyPath -Name $idleTimerRegKeyName -
Value $idleTimerRegKeyValue -PropertyType DWORD -Force

# Disable service if desired
$svc = Get-Service -Name $serviceName
if (!$enableAutoShutdown) {
    $msg = "Attempting to disable {0} service" -f $serviceName
    Write-Host $msg
    try {
        if ($svc.Status -ne "Stopped") {
            Start-Sleep -s 15
            $svc.Stop()
            $svc.WaitForStatus("Stopped", 180)
        }
        Set-Service -InputObject $svc -StartupType "Disabled"
        $status = if ($?) { "succeeded" } else { "failed" }
        $msg = "Disabling {0} service {1}" -f $svc.ServiceName, $status
        Write-Host $msg
    }
    catch {
        throw "Failed to disable CAMIdleShutdown service."
    }
}
```

If the remote workstation was provisioned by CAS Manager then the `idle timer` is already configured. In this case the `Install Service` section of the above commands can be skipped.

3. Set `$idleTimerRegKeyValue` to be between 5 and 10000, this is the number of minutes before the remote workstation is considered idle after the last PCoIP session was disconnected.
4. Set `$enableAutoShutdown` to `$true`.
5. Save the PowerShell script with a *.ps1* file extension and run the script. After the script has been successfully executed, the idle shutdown timer should be running. The remote workstation will now shutdown after the specified number of minutes of inactivity, as configured in step 3, once the user disconnects.

For more information on the Idle Service Shutdown, see Installing and Configuring CAS Manager as a Service in the CAS Manager as a Service guide.

Auto Log-Off Service

When a user disconnects their PCoIP session from a Linux PCoIP Agent, a different user is unable to connect unless the existing remote workstation user session is terminated. This will result in the remote workstation being locked, and unusable in a floating pool assignment, since a different user cannot log-in.

The auto log-off service enables you to bypass this issue by terminating a user session after the PCoIP session has been terminated. The auto log-off service monitors the **pcoip-server** process every minute. If it is not an active process then it samples the CPU load involved and if it is below a certain level for a certain amount of minutes, the script terminates the **pcoip-desktop-child** process which emulates a user logging off.

The auto log-off service disconnects a user if following criteria are met:

- No active PCoIP session detected (**pcoip-server** process is terminated).
- CPU utilization is less than 20% (`CPUUtilizationLimit`) for over 20 minutes (`MinutesIdleBeforeLogOff`).

- Sampling rate is 1 minute (`OnUnitActiveSec`).

Installing and Configuring the Auto Log-Off Service

You must have a CentOS/RHEL 7.8 virtual machine or Ubuntu virtual machine installed in order to run this service.

CentOS/RHEL Virtual Machine

- Run the following command to install the `pcoip-agent-autologoff` service on a CentOS/RHEL virtual machine:

  ```
  sudo yum install pcoip-agent-autologoff
  ```

Ubuntu Virtual Machine

- Run the following command to install the `pcoip-agent-autologoff` service on a Ubuntu virtual machine:

  ```
  sudo apt-get install pcoip-agent-autologoff
  ```

Once you have installed the service you can manage it via the `pcoip-agent-autologoff-mgmt` script. This script is located in */opt/teradici/pcoip-agent-autologoff/pcoip-agent-autologoff-mgmt*. The following commands must be executed either from the script path, or using the full path of the script.

The following table outlines the options you can use to manage the auto log-off service:

Option	Description
`--enable`	Enable the service.
`--disable`	Disable the service.
`--change-params`	Modify CPU utilization limit (CPUUtilizationLimit) and Idle time before logging off (MinutesIdleBeforeLogOff).
`--change-timer`	Modify polling interval (OnUnitActiveSec). This value sets how often the service runs.

Option	Description
`--show-logs`	Shows last 100 log messages.
`--follow-logs`	Shows live log messages.
`--help`	Shows the tool help page.

The default settings are shown in the table below. It is possible to modify these settings after the auto log-off service has been installed and configured:

Setting	Default	Description
`MinutesIdleBeforeLogOff`	20 minutes	Number of minutes the remote workstation must be considered idle before it logs a user off. The timer only starts when a user is not in PCoIP session.
`CPUUtilizationLimit`	20%	Value between 0 and 100 representing CPU utilization percentage. If average CPU utilization is below this value, the machine is considered idle, and will log-off if maintained for `MinutesIdleBeforeLogOff` .
`OnUnitActiveSec`	1 Minute	Polling interval in minutes for checking the CPU utilization.

Enabling the Auto Log-Off Service

The following section outlines how to enable the auto log-off service.

1. To enable the service run the following command:

   ```
   sudo pcoip-agent-autologoff-mgmt --enable
   ```

2. To disable the service run the following command:

   ```
   sudo pcoip-agent-autologoff-mgmt --disable
   ```

Updating the Auto Log-Off Service Configuration

The following section outlines how to update the auto log-off service configuration.

- Run the following command to change `MinutesIdleBeforeLogOff` or `CPUUtilizationLimit` :

  ```
  sudo pcoip-agent-autologoff-mgmt --change-params

  # follow the prompt to apply changes to the service
  ```

- Run the following command to change `OnUnitActiveSec` :

  ```
  sudo pcoip-agent-autologoff-mgmt --change-timer

  # follow the prompt to apply changes to the service
  ```

- Run the following command to show the log history:

  ```
  sudo pcoip-agent-autologoff-mgmt --show-logs
  ```

- Run the following command to follow the logs:

  ```
  sudo pcoip-agent-autologoff-mgmt --follow-logs
  ```

- Run the following command to display help information:

  ```
  sudo pcoip-agent-autologoff-mgmt --help
  ```

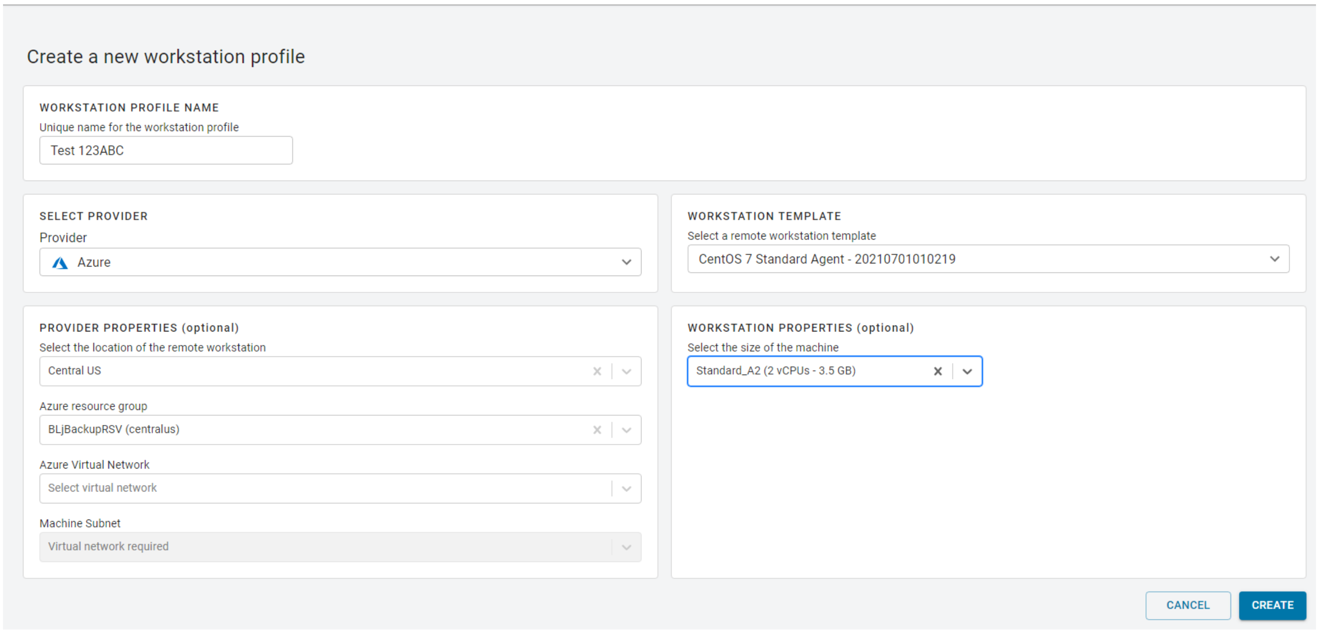
Workstation Profiles

Workstation profiles logically group remote workstation provisioning information. Profiles define a set of workstation deployment settings that will be shared by any workstation deployed from this profile. This feature enables administrators to streamline deploying workstations and improves the experience of deploying a large number of identical remote workstations.

Creating a Workstation Profile

The following steps outlines how to create a workstation profile from the Admin Console. Once you have created a workstation profile, you can use it to deploy a remote workstation.

1. Click on **Workstation Profiles** from the Admin Console sidebar.
2. Click the **+** icon to create a new workstation profile.
3. Enter a unique name for the workstation profile.
4. Select the cloud provider you wish to use. Currently Azure and GCP are supported.
5. Select a remote workstaion template. The remote workstation template defines the base OS and provisioning steps for the remote workstation you will deploy. The only required field is the **Resource template ID** parameter.



Create a new workstation profile

WORKSTATION PROFILE NAME
Unique name for the workstation profile
Test 123ABC

SELECT PROVIDER
Provider
Azure

WORKSTATION TEMPLATE
Select a remote workstation template
CentOS 7 Standard Agent - 20210701010219

PROVIDER PROPERTIES (optional)
Select the location of the remote workstation
Central US

Azure resource group
BLjBackupRSV (centralus)

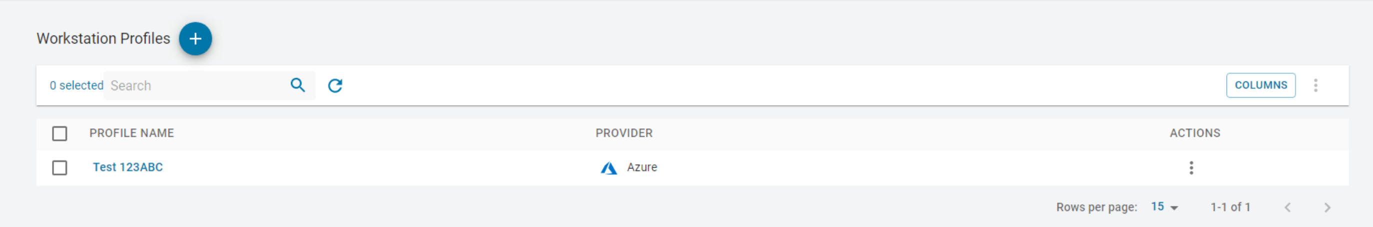
Azure Virtual Network
Select virtual network

Machine Subnet
Virtual network required

WORKSTATION PROPERTIES (optional)
Select the size of the machine
Standard_A2 (2 vCPUs - 3.5 GB)

6. Once you entered all the required information, click **CREATE**.

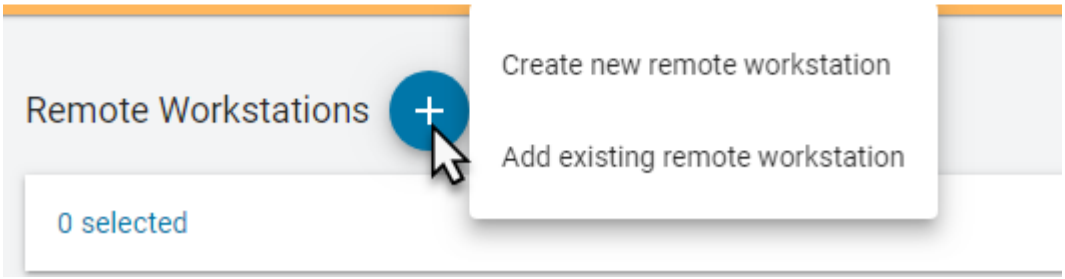
The workstation profile you created will now be visible in the list of workstation profiles. You can delete any workstation profiles from this view.



Deploying a Remote Workstation with a Workstation Profile

Once you have created a workstation profile you can use it to populate the **Create Workstation** page. The following steps outline how to use the workstation profile when creating a new remote workstation:

1. Click **Workstations** from the Admin Console sidebar.
2. Click **Create new remote workstation** from the add remote workstation icon.



Remote Workstations
Create new remote workstation
Add existing remote workstation
0 selected

3. Select the cloud provider you wish to use. This must be the same cloud provider you entered when you created the workstation profile.
4. From the **WORKSTATION PROFILE** tab select the workstation profile you created to auto-fill the page with the provider properties, workstation template and workstation properties.
5. If you chose Azure as your cloud provider, enter the remote workstation username and password.
6. Enter the Active Directory account and password. The Active Directory information is only used during provisioning to join the remote workstation to the domain. This information will not be saved by the CAS Manager or Admin Console.
7. Click **DEPLOY**.

The remote workstation has now been created and deployed and will be visible on the **Workstations** page. Using a specific workstation profile ensures that the identical information is used for all remote workstations created.

Workstation profile checkbox

If you deploy a remote workstation and do not use a workstation profile, you will be prompted by a message asking if you want to use this configuration as a workstation profile. You can select to not show this message again, but then you will not be prompted to create a workstation profile for future deployments.

SAML Configuration with CAS Manager

What is SAML?

SAML stands for Security Assertion Markup Language (SAML) and is a standard which Identity Providers use to communicate authorization credentials to different Service Providers. This enables users to manage one set of credentials to authenticate with different services.

SAML enables federated login to several services by passing authorization credentials between services. A SAML flow has three main roles:

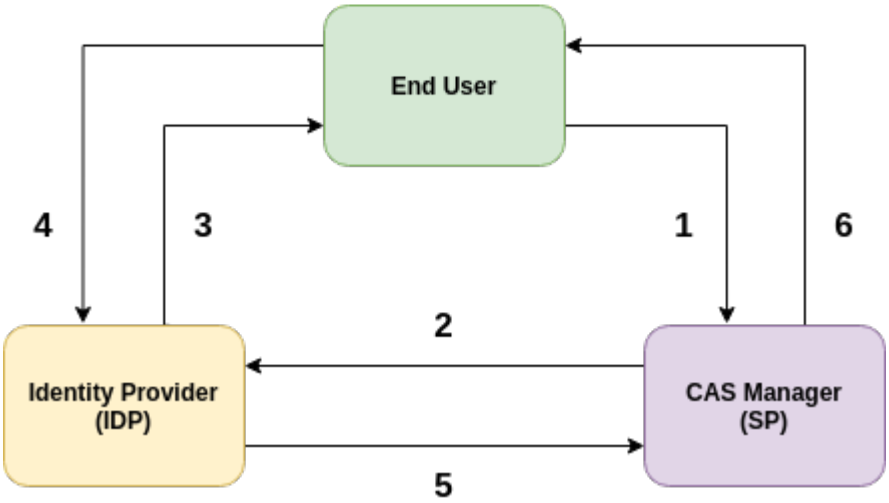
- **End User**: A user who is trying to access a service using federated login credentials
- **Identity Provider (IDP)**: An identity provider performs the authentication about the end users identity and sends the necessary data to the service provider along with any other access control data in the form of **SAML Assertions**. Popular examples are Azure Active Directory and Okta.
- **Service Provider (SP)**: A service provider is the system that requests authentication from an identity provider to authorize an end user. **CAS Manager plays the role of a SP**

SAML Assertions

SAML Assertions are XML documents that the IDP sends to a given SP to validate user authorization. There are three different types of SAML Assertions:

- **Authentication**: This assertion provides user identity and the time at which a user was authenticated and the method of authentication that was used.
- **Attribute**: This assertion passes the SAML attributes about the user to the service provider. There can be more than one attribute assertions in a SAML response.
- **Authorization**: This assertion is the decision that determines if the user was successfully authorized to access the service or not by the IDP. Most common causes of failed authorization are incorrect password and/or insufficient access to the service the end user tried to access.

CAS Manager Initiated SAML Authentication Flow



In the diagram above the following is happening

1. An end user wants to login to CAS Manager. The user uses the SSO link for CAS Manager.
2. CAS Manager requests the configured IDP for the SAML response for the user.
3. IDP requests the user to login and verifies credentials.
4. User logs in with the desired credentials to IDP.
5. The IDP now sends a SAML response to CAS Manager based on the user provided credentials.
6. CAS Manager validates the SAML response and ***SAML Attribute Assertions for CAS Manager*** received from the IDP, and then grants access to the end user.

SAML Attribute Assertions for CAS Manager

CAS Manager checks for the following attributes in the SAML response received from the configured IDP:

- **NameID**: CAS Manager verifies the NameID attribute, which is used to uniquely identify a user. The NameID value is typically a user's UPN or email.
- **Group Attributes**: CAS Manager can also verify a user's group membership from properties in the AttributeStatement of the SAML Assertion. The ***Group attribute name*** (configured in the ***Allowed Groups*** tab on the ***Multi Admin Setting*** page of the Admin Console) specifies the

name of the Attribute where the groups are returned. The AttributeValue can match either a *Group ID* or *Group Name* based on how an Allowed Group was created in the Multi-Admin Settings page.

CAS Manager will allow access to a user through a SAMl configuration if the user is in the list of **Allowed Admins** in CAS Manager or the user is a member of one or more of the **Allowed Groups** in your IDP. Hence if you need to revoke a user's access to CAS Manager through a SAMl configuration, you will need to remove the user from the **Allowed Admins** list in CAS Manager and remove the user's membership from any **Allowed Groups** through your IDP.

Configure CAS Manager as a SAML Service Provider to Enable Multi-Admin

The following section outlines the steps to setup and configure SAML for CAS Manager using the CAS Manager Admin Console:

1. From the account icon click **Multi Admin Settings** to create a new multi-admin configuration.
2. Register CAS Manager as a SP with your IDP. You can obtain the **Assertion Consumer Service URL** and **Audience URL** from the **Configuration Info** section. This information should be used to configure your IDP to recognize CAS Manager as a SP.
3. Configure CAS Manager to be able to connect to your IDP. Obtain the **Identity Provider Login URL** and **Identity Provider Certificate** from your IDP and configure the **IDP Settings** section accordingly. Alternatively you can also upload an ***IDP XML Metadata*** file in the **IDP Settings** section.
4. Enable Multi-Admin configuration to use configured IDP. Make sure that your configuration is enabled by toggling the switch at the bottom of the **Configuration Info** section and confirm that you see the ***Configuration is enabled*** message.
5. Configure CAS Manager Assertion Attributes:
    - To allow individual user as admin, go to the **Allowed Admins** section and add the UPN associated to that user. CAS manager validates the UPN against the **NameId** SAML assertion attribute in the SAML response received from the IDP.
    - To allow user groups. Go to the **Allowed Groups** section and configure the **Group Attributes** accordingly. This configures CAS Manager to validate the **Group Name** and/or **Group ID** SAML attribute assertions in the SAML response received from the IDP.

  - You can configure either **Allowed Admins** or **Allowed Groups** or both in the **Multi-Admin Settings**.

6. Allowed users can now access CAS Manager by opening the **CAS Manager login page** URL which is available in the **Configuration Info** section. Alternatively, users can also directly login via the IDP using the **Direct login via identity provider** URL also available on the **Configuration Info** section.

Configuration Information

This section contains auto-generated information about the login URLs and IDP:

- **CAS Manager login page**: A link to the page for multi-administrator login to the Admin Console. This is the SSO link used by the end user in **Step 1** of SAML auth flow diagram
- **Direct login via identity provider**: An endpoint to which multi-admin sign-in requests can be sent. This is the login page for the configured IDP.
- **Assertion Consumer Service URL**: The callback URL provided to the IDP to which user information is sent once the IDP has authorized the user. This is the CAS Manager endpoint that the IDP sends the SAML response to in **Step 5** of the SAML auth flow diagram
- **Audience URL**: The entity ID that the IDP can use to identify the Admin Console.

IDP Settings

This section contains IDP settings that can be updated to manage the SAML configuration within CAS Manager:

- **Identity Provider Login URL**: The IDP endpoint to which SAML authentication requests are sent. This endpoint is the one that CAS Manager sends the SAML login request to in **Step 2** of SAML authentication flow diagram above.
- **Identity Provider Certificate**: The public certificate of the IDP used to verify the signature of the IDP.

You can also upload a .xml file that contains your IDP information.

Allowed Admins

This section enables you to add new admins and displays all existing admins that are allowed to login via your IDP. To add a new admin, enter their e-mail, and click the Add Admin button.

Allowed Groups

This section enables you to add new groups and displays all existing groups that are allowed to login via your IDP. To enable the access for a group of users, enter the ***claim type*** and ***group claim*** and click Add Group.

- The ***claim type*** informs CAS Manager how the group is returned in the SAML attribute assertions in the SAML response received from your IDP.
- The ***group claim*** matches against the group either in the Group Name claim or in the Group ID claim received in the SAML attribute assertions for a user based on the ***claim type*** defined for the group.

Service Account and API Access

CAS Manager as a Service provides direct API access in the CAS Manager as a Service service. API's are an advanced way of interacting with the service, which enables you to integrate it into your business systems or to automate your use of the service for your specific needs.

> **Teradici Advantage Partner Program**
>
> To access and use the CAS Manager as a Service APIs, you must be a member of the Teradici Advantage Partner Program (TAPP) or have been pre-approved by Teradici. Contact Teradici here for more information.

Service Accounts: There are two types of service accounts that you can create with the Admin Console:

CAS Manager Service Accounts

The CAS Manager service account is an account that is created from the Admin Console for the purpose of creating future deployments and deployment service accounts through the CAS Manager as a Service APIs. The CAS Manager service account cannot perform any actions within a deployment, and so further actions to a deployment require the deployment service account, which is outlined below. For information on creating a CAS Manager service account, see here.

Deployment Service Accounts

Deployment service accounts are specific accounts that can only perform actions against the deployment, such as adding remote workstations. The deployment in this case is the deployment the service account is created within. They cannot perform actions against any other deployment. For information on creating a deployment service account, see here.

API Access Token

The API Access Token can be used to enable a user to operate at a level above deployments, such as creating a new deployment. The API Access Token is only valid for a limited period of time. This token also acts as an authorization token that can be used when performing an account ownership transfer, as outlined in the Account Ownership section of the CAS Manager as a Service guide.

For more detailed information on accessing the CAS Manager as a Service APIs, see https://cam.teradici.com/api/docs.

Creating a CAS Manager Service Account

You can create a CAS Manager service account from within the Admin Console. The following steps outline how to create a CAS Manager service account.

1. Click on your account name and select **CAS Manager service account**.
2. Click the **+** icon from the CAM service account page and name your new account.
3. Once you have created the CAS Manager service account download the JSON file or copy the key id. Ensure that you store the file securely as this key cannot be recovered if lost.
4. Go to the Service Account Keys section of the CAS Manager as a Service API documentation for the required APIs to use this key to create a deployment.

Creating and Assigning a Deployment Service Account

You can create and assign a deployment service account to a deployment through the **Deployments** option within the CAS Manager as a Service Admin Console. The following steps outline how to add a deployment service account to an existing deployment:

1. Click on your deployment from the console dropdown to display your existing deployments.
2. Click the kebab icon and click **Edit deployment** to display the deployment properties page.
3. Under the **Deployment Service Accounts** tab click the **+** sign to create a service account.
4. Once the service account has been created it will return service account information. This information should be saved as a JSON file in a secure location, as it can only be retrieved once. It will return a CAS Manager as a Service API token that you can use to query the CAS Manager as a Service APIs. This token is only authorized to access resources associated to the deployment that service account is associated with.

All deployment service accounts associated with a specific deployment will be listed on the deployment page. You can delete deployment service accounts from this page. For information on using the deployment service accounts and deployment service keys with the CAS Manager as a Service APIs, see here.

Obtaining a CAS Manager as a Service API Access Token

API access tokens permit you to enable other tools and applications to interact with CAS Manager as a Service through public APIs. The access token has tenant level permissions, which enables you to access all of a user's resources from any deployment.

To obtain a CAS Manager as a Service API Access token:

- Click **Get API token** from the user account icon within the Admin Console. You will receive the following message:

You need to copy the token as it will expire after a period of time.

> **Teradici Advantage Partner Program**
>
> To access and use the CAS Manager as a Service APIs, you must be a member of the Teradici Advantage Partner Program (TAPP) or have been pre-approved by Teradici. Contact Teradici here for more information.

Adding a Remote Workstation

You can add an existing remote workstation you created within the Admin Console, or one created in your cloud environment to a deployment. You can also view and add available resource groups if the remote workstation has valid cloud credentials. The remote workstation must have a PCoIP Agent installed on it and be visible to the Connector. You must have a valid CAS Software registration code and the remote workstation, and user, must be part of the deployments active directory domain. Any remote workstations that have a PCoIP Agent installed must be domain joined.

The following steps outlines how to add an existing remote workstation to your deployment using the Admin Console:

1. Click **Workstations** from the console sidebar.
2. Click the Add Remote Workstation button and click **Add existing remote workstation** to display the Add a Remote Workstation panel.
3. Select a Cloud Services Provider.
    - If your remote workstation has AWS credentials select the AWS region.
    - If your remote workstation has Azure credentials you can view and select available resource groups from the resource groups tab.
    - If your remote workstation has GCP credentials select the GCP region where your remote workstation resides, as well as the GCP zone.
    - If your remote workstation is on the Private Cloud you can search for, and add, these remote workstations. They must be domain joined and have a PCoIP Agent installed. If you want to add remote workstations that are not domain joined, you can click DEFINE YOUR OWN MACHINES and enter the name of the remote workstation and add it. The Connector can connect to this remote workstation by a FQDN or an IP address. If you are using an IP address, ensure it is static or persistently assigned to the remote workstation in question.
4. Select the remote workstations you want to add.
5. Select how you want to manage adding users to these remote workstations. You can individually select users, add users later or use workstations pools.

6. Click **SAVE**.

The remote workstation should now appear on the **Workstations** page.

Editing a Remote Workstation

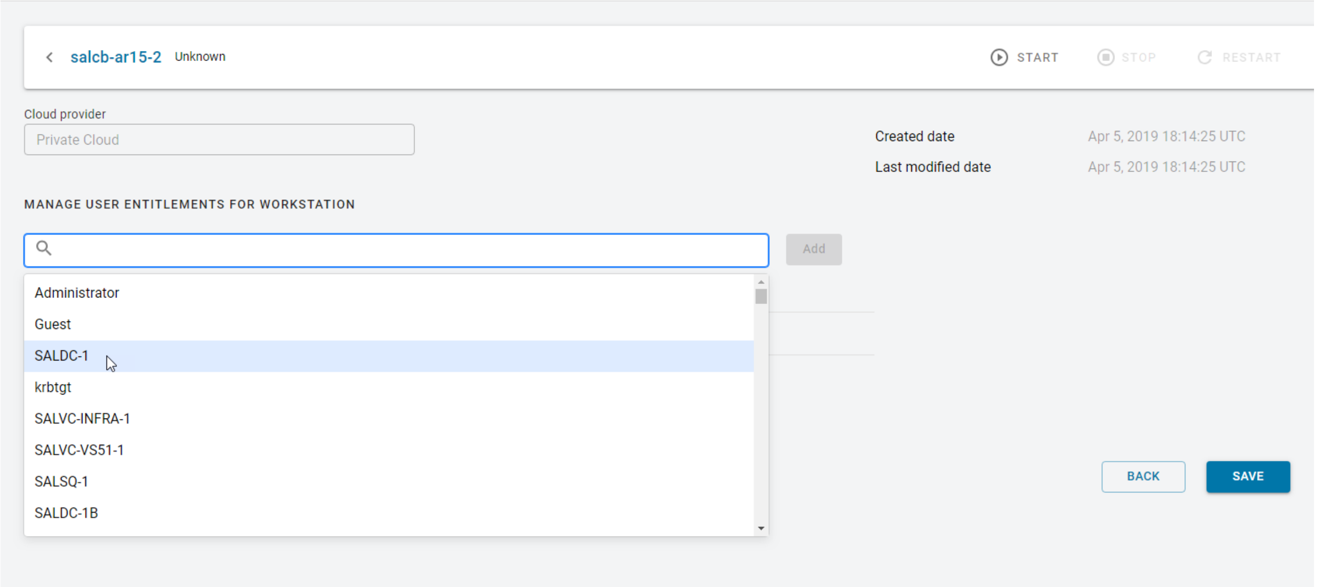
Once you have created a remote workstation within the Admin Console you can manage and reconfigure it directly from the **Remote Workstations** page.

You can search for specific remote workstations by name by using the search bar in the table toolbar.

Entitling Users

Once you have created a remote workstation you can entitle users from the active directory account to specific remote workstations. The following section outlines how to entitle users:

1. Click the kebab option under the **ACTIONS** column to edit the desired remote workstation.
2. Click **Edit**.
3. Select the search bar and select the user you want to entitle:



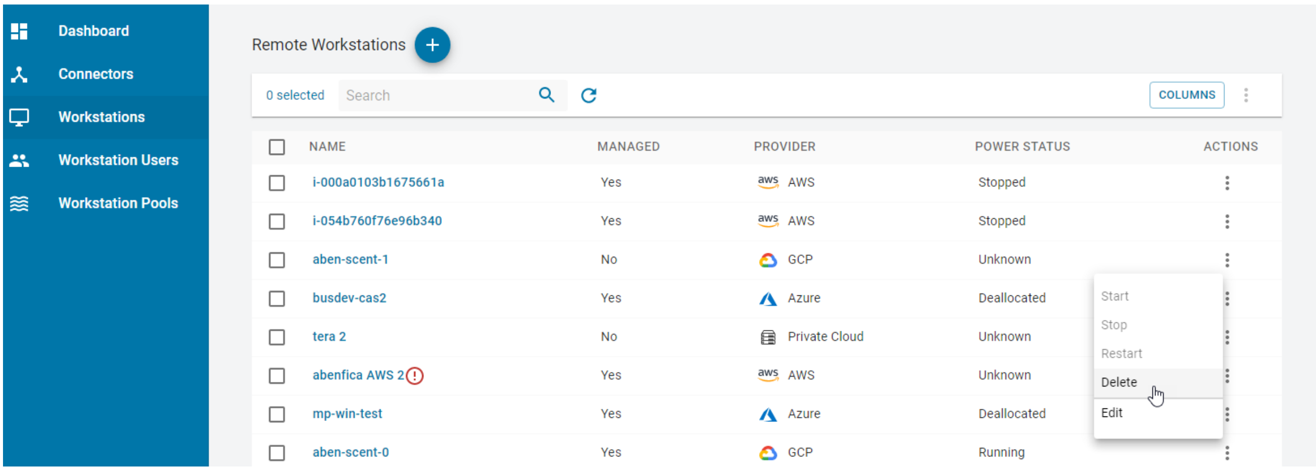
4. Click **Add** and then **SAVE**.

The user you entitled will appear in the ***USER*** column on the Remote Workstations page for that particular remote workstation.

Deleting Remote Workstations from the Public Cloud

You can delete existing remote workstations from AWS, Azure, and GCP from thes Admin Console. Only remote workstations that exist in AWS, Azure, and GCP and are part of deployments that have valid cloud credentials can be deleted.

1. Click **Workstations** from the console sidebar to display your existing remote workstations.
2. Click the kebab option under the **ACTIONS** column.
3. Click **Delete**.



NAME	MANAGED	PROVIDER	POWER STATUS	ACTIONS
i-000a0103b1675661a	Yes	AWS	Stopped	⋮
i-054b760f76e96b340	Yes	AWS	Stopped	⋮
aben-scent-1	No	GCP	Unknown	⋮
busdev-cas2	Yes	Azure	Deallocated	⋮
tera 2	No	Private Cloud	Unknown	⋮
abenfica AWS 2	Yes	AWS	Unknown	⋮
mp-win-test	Yes	Azure	Deallocated	⋮
aben-scent-0	Yes	GCP	Running	⋮

4. Click **CONFIRM** from the resulting pop-up message.

The process for deleting the remote workstation has now begun. It is also possible to bulk delete more than one remote workstation at a time by selecting multiple remote workstations to delete from the Admin Console.

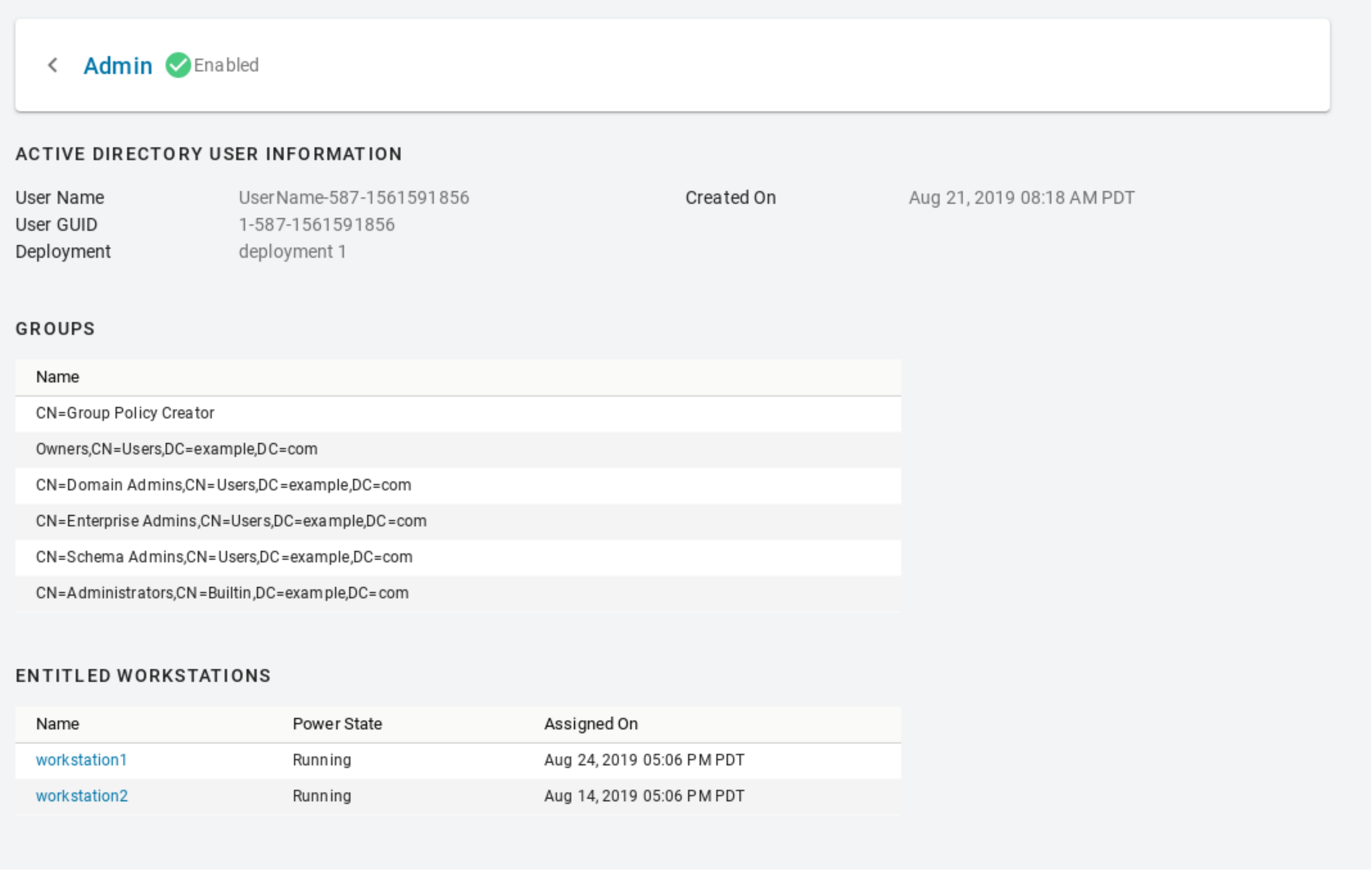
The remote workstation will disappear immediately from the Admin Console and can take 5-10 minutes to be deleted from the CAS Manager and public cloud. You should monitor the workstation in your cloud provider to ensure a successful completion. You will be notified in the Admin Console on the whether the deletion was successfull or not.

Viewing Remote Workstation Users

You can view all available Workstation users in your active directory by selecting the **Workstation Users** page. You can search for specific users by name with the search field in the toolbar. You can obtain the following user information for specific users:

- User Name
- User GUID
- Deployment
- Directory status
- User Groups
- Date of creation

When you select a specific user you will be shown all user groups and entitled remote workstations associated with this user:



ACTIVE DIRECTORY USER INFORMATION

User Name	UserName-587-1561591856	Created On	Aug 21, 2019 08:18 AM PDT
User GUID	1-587-1561591856		
Deployment	deployment 1		

GROUPS

Name
CN=Group Policy Creator
Owners,CN=Users,DC=example,DC=com
CN=Domain Admins,CN=Users,DC=example,DC=com
CN=Enterprise Admins,CN=Users,DC=example,DC=com
CN=Schema Admins,CN=Users,DC=example,DC=com
CN=Administrators,CN=Builtin,DC=example,DC=com

ENTITLED WORKSTATIONS

Name	Power State	Assigned On
workstation1	Running	Aug 24, 2019 05:06 PM PDT
workstation2	Running	Aug 14, 2019 05:06 PM PDT

This gives your an overview of a specific user's entitlements and deployment information and can be useful for troubleshooting issues.

Updating Cloud Provider Information

Remote workstations that have been added into CAS Manager, or created by CAS Manager, can be associated to a cloud provider. This enables CAS Manager to use the credentials for that cloud provider to access the remote workstation and enable power management. The cloud provider in which the remote workstation resides in can be changed. This can be done if either the remote workstation has been moved, or if the workstation was set to the Private Cloud, and you want to update it and assign it to the actual cloud provider.

Editing the cloud provider and zone information will not change the location of the remote workstation. This feature enables CAS Manager to point to a different location to verify the remote workstation exists in the specified zone. If you do not have valid cloud credentials for a cloud provider you will not be able to change the remote workstation cloud provider.

The following section outlines how to update a remote workstation on the private cloud and associate it to a workstation in a public cloud:

1. Click the kebab option under the ACTIONS column to edit the desired remote workstation.
2. Click Edit.
3. From the CLOUD INFORMATION panel click EDIT PROVIDER.
4. Select the cloud provider the remote workstation belongs to.
5. Select the region, resource group and zone, depending on the cloud provider, the remote workstation resides in.
6. Select the remote workstation and update the provider.

If you enter the correct cloud provider and zone for the remote workstation you will receive a notification that it has been updated. The new zone, cloud provider and information will be listed on this page also.

If you enter an incorrect zone then you will receive an error message stating that the remote workstation does not exist in the entered zone.

Setting Time and Date

You can configure the time zone, time format and date format within the Admin Console. This enables you to ensure the time zone is set to your local time zone or else to the time zone into which your remote workstations are deployed. The current date and time format provided by the web browser will be the default preference used.

The following steps outline how to set date and time preferences:

1. Click **Preferences** from the user account icon within the Admin Console.
2. Select the desired Date format, Time zone and Time format.
3. Click **SAVE**.

The new date and time preferences will now be applied globally where applicable across the entire Admin Console.

Activity Log

The CAS Manager activity log enables you to view a record of all activity and operations performed in your CAS Manager environment. You can choose whether to show all records or just the records from a selected deployment. To view the activity log from the Admin Console:

1. Click the user account icon within the Admin Console.
2. Click **Activity Log** to display the activity log for that deployment.

The logs will show the date, user account, source and activity details.

You can search for logs based on specific operations that occured. You can download all the logs available in CAS Manager by clicking the **Download CSV** button. For information on CAS Manager levels and how they impact the activity log, see CAS Manager.

> **Activity Log Expiration Timeframe**
>
> The Activity Log in the Admin Console contains short-term data, up to 7 days. After 7 days the log data expires. To maintain your long term storage Teradici recommends downloading the .csv file regularly.

Accessing the Activity Log through CAS Manager APIs

CAS Manager offers a RESTful API as an alternative to using the Admin Console. It allows for programmatic management and automation of resources in CAS Manager deployments.

The following API page details how you can obtain these Activity Logs using the CAS Manager APIs: https://cam.teradici.com/api/docs#tag/Activity-Logs

The Get activity logs and download activity logs API calls enable users to get the logs and download them as a .csv file.

CAS Manager Support Bundle

If you encounter an issue installing the CAS Manager or with the application itself, it is possible to generate a support bundle that can be sent to the Teradici support team to investigate and resolve.

To generate the support bundle, run the following command:

```
sudo /usr/local/bin/cas-manager diagnose --support-bundle
```

If this command is successful, a tar.gz file will be located under the */tmp* folder with a name formatted as follows:

/tmp/cas-manager-support-bundle-yyyymmddThhmmssZ.tar.gz

yyyymmddThhmmssZ represents the date and time the support bundle was created.

Support Bundle Information and Logs

The support bundle will collect the various information from the system and then zip the files into a `.tar.gz` file in the */tmp* directory.

Once you unzip the file the structure will be as follows:

- files/etc folder contains files with OS level information:
  - issue file will contain a copy of all contents from the */etc/issue* file.
  - os-release file will contain all operating system identification data that was found in the */usr/lib/os-release* file.
- files/var/log/cas-manager folder collects all of the CAS Manager log files, for example generate, configure, diagnose, install logs.

The **out** folder collects outputs from running various commands to expose the details of relevant system information and CAS Manager backend services, as outlined below:

- os folder will contain the following files:
  - dmesg.out file will contain the output of command `dmesg`.

  - **ls_-l@var@crash.out** file will contain the output of the command `ls -l /var/crash`.
  - **pgrep_-l_k3s.out** file will contain the output of the command `pgrep -l k3s`.
  - **ps_wwauxfx.out** file will contain the output of the command `ps wwauxfZ`.
  - **ss-aux.out** file will contain the output of the command `ss -ax`.
  - **who.out** file will contain the output of the command `who`.
- The **firewall** folder will contain the outputs of the command `firewall-cmd --list-services` .
- The **network** folder will display the following network files:
  - **netstat_-Wnap.out** file will contain the output of the command `netstat -Wnap`.
  - **ip_add.out** file will contain the output of the command `ip add`.
  - **selinux** folder will contain the following files:
    - **semodule_-l.out** file will contain the output of the command `semodule -l`.
    - **sestatus.out** file will contain the output of the command `sestatus`.
- The **deployments** folder will contain the following files:
  - **kubectl_get_deployment.out** file is the output of the command `kubectl get deployments` which list the status of all deployments for CAS Manager services.
  - The description of the deployment for each of the deployments through the output of the command `kubectl describe`.
- **pods** folder will contain a describe pod file for each pod.
- **logs** folder will display the log files for each pod.
- **services** folder will contain a Describe Service file for each service.
- **secrets** folder will display the output of the command `kubectl get secrets`.

CAS Manager Health Status

In the case that there is an issue with CAS Manager, the diagnose health command will provide an overview of CAS Manager's health. The following command will provide a list of services that are in healthy and unhealthy state. The command will allow the user to determine the services that are unhealthy and run more specific diagnosis on the unhealthy service:

```
sudo /usr/local/bin/cas-manager diagnose --health
```

The diagnose command lists all CAS Manager services, but not all services are essential. The essential services for CAS Manager are listed below. If any of these services are in an unhealthy state, the overall health status will be unhealthy:

- "activitylog"
- "activitylogconsumer"
- "authorization"
- "camadminconsolega"
- "connectors"
- "connectorsworker"
- "deploymentmgmt"
- "deploymentworker"
- "docs"
- "kafka"
- "machinemgmt"
- "machinemgmtdeleteworker"
- "machinemgmtworker"
- "machinemonitor"
- "machinemonitorworker"
- "poolmgmt"
- "poolmgmtworker"

- "secretmgmt"
- "redis"
- "resourcetemplates"
- "resourcetemplatestore"
- "userentitlement"
- "userentitlementworker"

Vault Issues

If you suddenly start getting errors when using CAS Manager features, it is possible the Vault token used in your CAS Manager deployment has expired. To diagnose, try the following options:

1. Run the following command to follow the logs for the secret management service:

   ```
   kubectl logs -l app=secretmgmt -f
   ```

2. While streaming the secretmgmt logs, try logging in to CAS Manager. If you see the following message in the logs, your Vault token may have expired:

   ```
   {"message":"Permission denied","level":"error"}
   ```

3. To confirm that the Vault token has expired, run the following command in the location you have the Vault CLI installed:

   ```
   vault token lookup <your CAS Manager Vault token>
   ```

4. If you get the following message after running this command, then your CAS Manager token has expired or become invalid:

   ```
   Error looking up token: Error making API request.

   URL: POST https://<your Vault address>/v1/auth/token/lookup
   Code: 403. Errors:

   * bad token
   ```

To fix this issue, create a renewable token and update your CAS Manager's Vault configuration to use that token. To avoid the Vault token from prematurely expiring again, follow the steps outlined here to set up automatic renewal for your Vault token.

Cloud Access Connector Installer Issues

Teradici moved to a new distribution system and on December 31, 2020 the legacy system was shut down. As a result of this change some errors may occur for users with Connectors that were installed from older installer versions.

In almost all cases downloading the latest installer version and running the `cloud-access-connector install`, `cloud-access-connector update` or `cloud-access-connector diagnose` commands with the new install should work.

Error Messages

The following are a list of potential error messages that a user may encounter as a result of the change of distribution system:

Attempting to download the installer

Error 1

```
user@vm:~$ mkdir ~/v2connector && cd ~/v2connector
user@vm:~/v2connector$ curl -LO https://teradici.bintray.com/cloud-access-
connector/cloud-access-connector-0.1.1.tar.gz
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 41 0 41 0 0 141 0 --:--:-- --:--:-- --:--:-- 141
user@vm:~/v2connector$ tar xzvf cloud-access-connector-0.1.1.tar.gz

gzip: stdin: not in gzip format
tar: Child returned status 1
tar: Error is not recoverable: exiting now
```

Error 2

```
user@vm:~$ mkdir ~/v2connector && cd ~/v2connector
user@vm:~/v2connector$ curl -LO https://teradici.bintray.com/cloud-access-
connector/cloud-access-connector-0.1.1.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--
0curl: (6) Could not resolve host: teradici.bintray.com
```

Running older installer versions

If the Connector installer was installed before December 11, it may generate some of the following errors:

Error 3

```
[2020-12-22T20:36:57Z]  INFO Verifying installer version
[2020-12-22T20:36:57Z] ERROR yaml: unmarshal errors:
  line 1: cannot unmarshal !!str `The req...` into docker.ComposeConfig
```

Error 4

```
[2020-12-22T20:07:09Z]  INFO Downloading compose file
[2020-12-22T20:07:09Z]  INFO curl: (60) SSL certificate problem: self signed
certificate
[2020-12-22T20:07:09Z]  INFO More details here: https://curl.haxx.se/docs/
sslcerts.html
[2020-12-22T20:07:09Z]  INFO
[2020-12-22T20:07:09Z]  INFO curl failed to verify the legitimacy of the server
and therefore could not
[2020-12-22T20:07:09Z]  INFO establish a secure connection to it. To learn more
about this situation and
[2020-12-22T20:07:09Z]  INFO how to fix it, please visit the web page mentioned
above.
[2020-12-22T20:07:09Z] ERROR exit status 60
```

Error 5

```
[2020-12-22T20:24:06Z]  INFO Configuring Docker Daemon
[2020-12-22T20:24:06Z]  INFO populateTrustAndKeyStore: Pulling setup container
and populating Java trust and key store
[2020-12-22T20:24:07Z]  INFO Error response from daemon: error parsing HTTP 404
response body: invalid character '<' looking for beginning of value: "<!doctype
html><html lang=\"en\"><head><title>HTTP Status 404 – Not Found</title><style
type=\"text/css\">body {font-family:Tahoma,Arial,sans-serif;} h1, h2, h3, b
{color:white;background-color:#525D76;} h1 {font-size:22px;} h2 {font-size:
16px;} h3 {font-size:14px;} p {font-size:12px;} a {color:black;} .line {height:
1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status
404 – Not Found</h1><hr class=\"line\" /><p><b>Type</b> Status Report</
p><p><b>Description</b> The origin server did not find a current representation
```

```
for the target resource or is not willing to disclose that one exists.</p><hr
class=\"line\" /><h3>Apache Tomcat/8.5.59</h3></body></html>"
[2020-12-22T20:24:07Z] ERROR exit status 1
```

Error 6

```
[2021-02-11T01:02:42Z] INFO Error response from daemon: Head https://teradici-
docker-registry.bintray.io/v2/diagnostics/manifests/stable: unauthorized:
Unauthorized
[2021-02-11T01:02:42Z] ERROR exit status 1
```

Cloud Access Connector Connectivity Issues

CAS Manager provides some diagnosic checks that can be used to troubleshoot the cause of issues you may be experiencing with your Connector. Run the following command:

```
cd /usr/sbin
sudo ./cloud-access-connector diagnose
```

Please note that older installs and updates may still be in the legacy directory at `~/v2connector` .

This command can also be used to verify that your Connector has been correctly configured. The diagnostic checks cover Remote Workstation connectivity and Active Directory connectivity.

The following table lists the flags associated with this command:

Flag	Description
`--rw`	The Remote Workstation FQDN
`--ad`	Verify connectivity to currently configured Active Directory server
`-h` `--help`	help for diagnose
`--debug`	This flag can be run if you initial install of the Connector fails. It provides a detailed output of the Connector installation. This is useful for self-troubleshooting or to provide to the Teradici support team when logging a support ticket.

Common Installation Issues with the Connector

For information on issues relating to failed Connector installations, Teradici has a KB article that details troubleshooting steps for common issues related to installing the Connector, see here.

⚠ **Connector Upgrade and Diagnose Issues**

Several previous versions of Connector installers are no longer compatible with our latest infrastucture upgrades. When you run the configure or diagnose commands with these older versions you may receive errors such as "*Error response from daemon: GET https://docker.cloudsmith.io/......: unauthorized*" for example. If this occurs you need to download the latest version of the Connector installer from here.

Remote Workstation Connectivity Check

This command will attempt to connect to the specified remote workstation on the ports required for establishing a PCoIP session. It checks to ensure that the PCoIP Agent is running on the remote workstation.

Example command to diagnose remote workstation connectivity issues:

```
sudo ./cloud-access-connector diagnose --rw fqdn.of.my.rw
```

Check Passes

- Your Connector is able to resolve the FQDN of the remote workstation and connect to it.
- The PCoIP Agent is running and responding on the remote workstation.

Check Fails

If the check fails it may be as a result of one or more of the following issues:

- Firewall or network routing rules or restrictions may be in place.
- A failure has occured and the FQDN of the remote workstation cannot be resolved.
- The PCoIP Agent on the remote workstation is not running or is unresponsive.

Active Directory Connectivity Check

This command will attempt to connect to the Active Directory domain controller that was provided during installation using those same credentials.

Example command to diagnose Active Directory connectivity issues:

```
sudo ./cloud-access-connector diagnose --ad
```

Check Passes

- The Connector is able to resolve the FQDN of the domain controller and authenticate to it.

Check Fails

If the check fails it may be as a result of one or more of the following issues:

- Firewall or network routing rules or restrictions may be in place.
- A failure has occured and the FQDN of the domain controller cannot be resolved.
- The Active Directory server may be unresponsive.
- The check was unable to authenticate to the Active Directory server.

Cloud Access Connector Log Collection

The following section outlines how to view the logs and view the status of the Connector services and installer. This information can help troubleshoot issues relating to the Connector.

To view the status of all services in the Connector run the following command:

```
sudo docker service ls
```

To get logs from services run the following command:

```
sudo docker service logs [service]
```

The following list details the important services:

- `connector_activedirectorysync`
- `connector_brokerexternal`
- `connector_brokerinternal`
- `connector_cm`
- `connector_cmsg`
- `connector_connectorgateway`
- `connector_healthcheck`
- `connector_managementinterface`
- `connector_sumologic`

The installer and update logs are saved for the installer in */var/log/cloud-access-connector/*.

Retrieving Cloud Access Connector Version Numbers

Understanding the version number of a Connector can be useful when troubleshooting issues and to ensure you are running the latest version of the Connector. The Connector, from version 67 on, has a single version number. Previously, the installer and connector version numbers were different. These have now been combined to display a single version number going forward.

Connector Installer Version

The installer is used for installing, updating and diagnosing issues related to the Connector installation process. It can be updated at the same time as the Connector, and also updated independent of the Connector as updates are made to improve installation specific areas. A change in the version of the installer does not require an upgrade to the Connector itself.

If you have downloaded the Connector installer you can obtain the version number by running one of the following commands (depending on where it has been copied to):

```
./cloud-access-connector --version
```

or

```
/usr/sbin/cloud-access-connector --version
```

A successful response is outlined below:

```
cloud-access-connector v66.0.63_9606001760
```

The first command should be used if you are currently in the same directory as the installer. This is more common for older versions of the installer. The second command is the location where newer versions of the installer have been copied to.

The Connector installer version also appears at the top of the output when you run an installation:

```
user@vm:~$ sudo /usr/sbin/cloud-access-connector install
[2021-02-03T17:16:01Z]  INFO Set docker registry as: docker.cloudsmith.io/
teradici/cloud-access-connector
[2021-02-03T17:16:01Z]  INFO Starting cloud-access-connector
version=v66.0.63_9606001760
...
```

It will also be logged by the installer:

```
user@vm:/var/log$ sudo more /var/log/cloud-access-connector/
install_2021-02-03T17-16-01.log
time="2021-02-03T17:16:01Z" level=info msg="Starting cloud-access-connector"
version=v66.0.63_9606001760
time="2021-02-03T17:17:14Z" level=error msg="You must accept the EULA and
Privacy Policy to continue."
...
```

You can also view the Connector installer version number from the Teradici download site when viewing the download filename.

Connector Version

The Connector version, sometimes referred to as the YAML or compose file, denotes the combination of containers that make up a particular release of the Connector. The primary location to view the version of your running Connector is from the Connectors page in the Admin Console.

This version number represents a combination of specific versions of services that run on the Connector. For example, version 42 of the Connector includes the PCoIP Connection Manager 21.01.0. When troubleshooting issues, this version is used by Teradici's support team to inform them as to which version of each service is running on your Connector.

> **Legacy Connector Versions**
>
> You should ensure that you keep this version as up to date as possible. Teradici is continuously enhancing, adding features, fixing bugs and improving the overall security of the Connector. If you have a version that is v38 or lower, you should update your Connector as previous versions were integrated with an installer that predates our current Connector download location, and further installs or updates from that legacy installer may not work correctly.

If you are unable to access the Admin Console, you can obtain the version of the Connector from the configuration file itself, as outlined in the below example:

```
user@vm:/var/log$ cat /var/local/teradici/docker-compose.yaml | grep
CACV2_VERSION
      CACV2_VERSION: 42
```

DNS Name Resolution Configuration on Ubuntu

Configuring a DNS Name Resolution

The first step to configuring the Connector is to ensure that there is a solid connection between the Connector and the Active Directory Domain Controller. You need to ensure that you can route from this machine to the Domain Controller and that there is nothing to prevent port 443 (https) and port 636 (LDAPS) connecting between the two systems.

In the following example the IP of the Domain Controller is 10.162.0.42.

> **.yaml format file**
>
> In step 4 below this is a .yaml format file and the whitespaces are critical. Tabs cannot be used in a .yaml file. Please format it exactly as it is displayed in the above example.

1. Install Resolvconf

   ```
   sudo apt update
   sudo apt install resolvconf
   ```

2. Edit the resolvconf file

   ```
   Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
   DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
   127.0.0.53 is the systemd-resolved stub resolver.
   run "systemd-resolve --status" to see details about the actual nameservers.
   search teradici.local
   nameserver 10.162.0.42
   ```

3. Restart the resolvconf service

   ```
   sudo service resolvconf restart
   ```

4. Edit Netplan Config

```
sudo nano /etc/netplan/50-cloud-init.yaml
This file is generated from information provided by the datasource. Changes
to it will not persist across an instance reboot. To disable cloud-init's
network configuration capabilities, write a file
/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
network: {config: disabled}
network:
ethernets:
ens4:
dhcp4: true
match:
macaddress: 42:01:0a:a2:00:29
set-name: ens4
nameservers:
search: [teradici.local]
addresses: [10.162.0.42]
version: 2
```

5. Restart Netplan

```
sudo netplan apply
```

6. Test DNS

```
ping <name-of-domain-controller>
```

7. If the response is successful, you should receive a message similar to the example below:

```
PING example.com (172.217.14.206): 56 data bytes
64 bytes from 172.217.14.206: icmp_seq=0 ttl=118 time=16.622 ms
64 bytes from 172.217.14.206: icmp_seq=1 ttl=118 time=50.675 ms
64 bytes from 172.217.14.206: icmp_seq=2 ttl=118 time=27.682 ms
64 bytes from 172.217.14.206: icmp_seq=3 ttl=118 time=19.886 ms
^C
--- example.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
```

Cloud Access Connector Support Bundle on RHEL/ Rocky Linux

⚠ **Beta Feature**

Please be aware that the Cloud Access Connector on RHEL/Rocky Linux is only currently available in a beta version. As such, this feature may change as it is developed, and it will not be supported by Teradici Global Support Services. Features in the beta version are considered not yet ready for full production and you use them at your own risk.

If you encounter an issue with the Connector on RHEL/Rocky Linux, it is possible to generate a support bundle that can be sent to the Teradici support team to investigate and resolve.

To generate the support bundle, run the following command:

```
sudo /usr/local/bin/cas-connector diagnose --support-bundle
```

If this command is successful, a tar.gz file will be located under the */tmp/* directory with a name formatted as follows:

/tmp/cas-connector-support-bundle-(date)-(time).tar.gz

Support Bundle Information and Logs

The support bundle will collect the various information from the system and then zip the files into a `.tar.gz` file in the */tmp* directory.

Once you unzip the file the structure will be as follows:

- files/etc folder contains files with OS level information:
    - issue file will contain a copy of all contents from the */etc/issue* file.
    - os-release file will contain all operating system identification data that was found in the */usr/lib/os-release* file.
- files/var/log/cas-connector folder collects all of the CAS Connector log files, for example generate, configure, diagnose, install logs.

The **out** folder collects outputs from running various commands to expose the details of relevant system information and CAS Connector backend services, as outlined below:

- **os** folder will contain the following files:
    - **dmesg.out** file will contain the output of command `dmesg` .
    - **ls_-l@var@crash.out** file will contain the output of the command `ls -l /var/crash` .
    - **pgrep_-l_k3s.out** file will contain the output of the command `pgrep -l k3s` .
    - **ps_wwauxfx.out** file will contain the output of the command `ps wwauxfZ` .
    - **ss-aux.out** file will contain the output of the command `ss -ax` .
    - **who.out** file will contain the output of the command `who` .
- The **firewall** folder will contain the outputs of the command `firewall-cmd --list-services` .
- The **network** folder will display the following network files:
    - **netstat_-Wnap.out** file will contain the output of the command `netstat -Wnap` .
    - **ip_add.out** file will contain the output of the command `ip add` .
    - **selinux** folder will contain the following files:
        - **semodule_-l.out** file will contain the output of the command `semodule -l` .
        - **sestatus.out** file will contain the output of the command `sestatus` .
- The **deployments** folder will contain the following files:
    - **kubectl_get_deployment.out** file is the output of the command `kubectl get deployments` which list the status of all deployments for CAS Manager services.
    - The description of the deployment for each of the deployments through the output of the command `kubectl describe` .
- **pods** folder will contain a describe pod file for each pod.
- **logs** folder will display the log files for each pod.
- **services** folder will contain a Describe Service file for each service.
- **secrets** folder will display the output of the command `kubectl get secrets` .

Cloud Access Connector Health Status on RHEL/ Rocky Linux

⚠ **Beta Feature**

Please be aware that the Cloud Access Connector on RHEL/Rocky Linux is only currently available in a beta version. As such, this feature may change as it is developed, and it will not be supported by Teradici Global Support Services. Features in the beta version are considered not yet ready for full production and you use them at your own risk.

In the case that there is an issue with the Connector on RHEL/Rocky Linux, the diagnose health command will provide an overview of Connector's health. The following command will provide a list of services that are in healthy and unhealthy states. The command will enable you to determine the services that are unhealthy and run more specific diagnosis on the unhealthy service:

```
sudo /usr/local/bin/cas-connector diagnose --health
```

The diagnose command lists all Connector service, these services for the Connector are listed below. If any of these services are in an unhealthy state, the overall health status will be unhealthy:

- "adsync"
- "broker"
- "cm"
- "cmsg"
- "connectorgateway"
- "sg"
- "healthcheck"
- "rwtelemetry"

DNS Name Resolution Configuration on RHEL/ Rocky Linux

Beta Feature

Please be aware that the Cloud Access Connector on RHEL/Rocky Linux is only currently available in a beta version. As such, this feature may change as it is developed, and it will not be supported by Teradici Global Support Services. Features in the beta version are considered not yet ready for full production and you use them at your own risk.

In order to install and configure CAS Manager or Connector on the RHEL or Rocky Linux machine, it's important to ensure that there is a solid connection between the machine and the Active Directory Domain Controller. You need to ensure that you can route from this machine to the Domain Controller and that there is nothing to prevent port 443 (https) and port 636 (LDAPS) connecting between the two systems.

The following steps are to ensure DNS settings are configured properly on the machine for CAS Manager or Connector to operate. The sample IP of the Domain Controller is `10.162.0.42` for `example-domain.com` :

1. Disable auto-configuration of DNS settings in order to prevent setting being overwritten on reboot. In this example the device name is `eth0` :

   ```
   nmcli device modify eth0 ipv4.ignore-auto-dns yes
   ```

   You may also need to disable this on the connection level in some cases. In this example the connection name is `eth0` :

   ```
   nmcli connection modify eth0 ipv4.ignore-auto-dns yes
   ```

2. Edit the Network Configuration scripts. Add the `DNS1` for the IP address for Active Directory's DNS server (typically the Domain Controller itself) and optionally `DNS2` for the fallback DNS server. You can optionally add `DOMAIN` for a DNS suffix (typically the Domain name):

   ```
   sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0
   TYPE=Ethernet
   PROXY_METHOD=none
   ```

```
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=ens192
UUID=dfe16427-21f1-429c-99cb-a1e9b42be181
DEVICE=ens192
ONBOOT=yes
DNS1=10.162.0.42
DOMAIN=example-domain.com
PEERDNS=no
```

3. Run the following command to restart the Network Manager:

```
sudo systemctl restart NetworkManager
```

4. Check the `/etc/resolv.conf` file to make sure that the desired DNS servers and search suffixes are there. It is important the the `nameservers` are the AD DNS or else the machine will fail to connect to the Domain Controller(s):

```
cat /etc/resolv.conf
# Generated by NetworkManager
search example-domain.com
nameserver 10.162.0.42
```

5. Test DNS by pinging the domain, in this example `example-domain.com` is the domain name:

```
ping example-domain.com
```

6. If the response is successful, you should receive a message similar to the example below:

```
PING example-domain.com (10.162.0.42): 56 data bytes
64 bytes from 10.162.0.42: icmp_seq=0 ttl=118 time=16.622 ms
64 bytes from 10.162.0.42: icmp_seq=1 ttl=118 time=50.675 ms
64 bytes from 10.162.0.42: icmp_seq=2 ttl=118 time=27.682 ms
64 bytes from 10.162.0.42: icmp_seq=3 ttl=118 time=19.886 ms
^C
```

```
--- example-domain.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
```

7. Reboot the machine and check that the DNS settings in `/etc/resolv.conf` persist and that you can still ping the domain as done in steps 4-6.

Applying Host Machines DNS settings to K3S

The host machine's DNS settings are copied from `/etc/resolv.conf` and applied to the CAS Manager and/or CAS Connector when the CAS K3S service starts. Because of this it is important that settings are correct on boot. You will need to either reboot the machine or restart the K3S service to apply the DNS settings to the CAS Manager or CAS Connector if changes are made post installation or configuration.

It is best to ensure DNS settings are correct before installing and CAS software on the machine.

Network Connectivity Issues

The Connector provides some diagnosic checks that can be used to troubleshoot the cause of issues you may be experiencing. Run the following command:

```
sudo /usr/local/bin/cas-connector diagnose
```

This command can also be used to verify that your Connector has been correctly configured. The diagnostic checks cover Remote Workstation connectivity and Active Directory connectivity.

The following table lists the flags associated with this command:

Flag	Description
`--rw`	The Remote Workstation FQDN
`--ad`	Verify connectivity to currently configured Active Directory server
`-h` `--help`	help for diagnose
`--debug`	This flag can be run if you initial install of the Connector fails. It provides a detailed output of the Connector installation. This is useful for self-troubleshooting or to provide to the Teradici support team when logging a support ticket.

Common Installation Issues with the Connector

For information on issues relating to failed Connector installations, Teradici has a KB article that details troubleshooting steps for common issues related to installing the Connector, see here.

Remote Workstation Connectivity Check

This command will attempt to connect to the specified remote workstation on the ports required for establishing a PCoIP session. It checks to ensure that the PCoIP agent is running on the remote workstation.

Example command to diagnose remote workstation connectivity issues:

```
sudo /usr/local/bin/cas-connector diagnose --rw fqdn.of.my.rw
```

Check Passes

- Your Connector is able to resolve the FQDN of the remote workstation and connect to it.
- The PCoIP agent is running and responding on the remote workstation.

Check Fails

If the check fails it may be as a result of one or more of the following issues:

- Firewall or network routing rules or restrictions may be in place.
- A failure has occured and the FQDN of the remote workstation cannot be resolved.
- The PCoIP agent on the remote workstation is not running or is unresponsive.
- There may be an issue with the DNS name. For more information on this, and how to potentially resolve this issue, see DNS Name Resolution Configuration on RHEL/Rocky Linux.

Active Directory Connectivity Check

This command will attempt to connect to the Active Directory domain controller that was provided during installation using those same credentials.

Example command to diagnose Active Directory connectivity issues:

```
sudo /usr/local/bin/cas-connector diagnose --ad
```

Check Passes

- The Connector is able to resolve the FQDN of the domain controller and authenticate to it.

Check Fails

If the check fails it may be as a result of one or more of the following issues:

- Firewall or network routing rules or restrictions may be in place.

- A failure has occured and the FQDN of the domain controller cannot be resolved.
- The Active Directory server may be unresponsive.
- The check was unable to authenticate to the Active Directory server.
- There may be an issue with the DNS name. For more information on this, and how to potentially resolve this issue, see DNS Name Resolution Configuration on RHEL/Rocky Linux.

Getting Support

If you are having trouble, help is available. This section contains information about contacting Teradici support and connecting with the Teradici user community.

Contacting Support

If you encounter problems installing or using Teradici technology, you can:

- Browse the Teradici Knowledge Base.
- Submit a Support Ticket.

The Teradici Community Forum

The PCoIP Community Forum allows users to have conversations with other IT professionals to learn how they resolved issues, find answers to common questions, have peer group discussions on various topics, and access the Teradici PCoIP Technical Support Service team. Teradici staff are heavily involved in the forums.

To join the Teradici community, visit the Teradici Knowledge Center.

Release Notes

To view the latest release notes for CAS Manager, see CAS Manager Release Notes.