What is CAS Manager?

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 deployment.

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 Firewalls and Load Balancing Considerations.

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

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.

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.

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.

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.

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.
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:

   ```bash
   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:

   ```bash
   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"
http_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.
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 to configure and add the RHEL repository, select the Downloads and Scripts option here from the CAS Manager support site.

Rocky Linux Repository

To force the specific repository for Rocky Linux 8, run the following command:

```
curl -1sLf https://dl.teradici.com/<RepositoryAccessToken>/cas-manager/cfg/setup/bash.rpm.sh | sudo -E distro=el codename=8 bash
```

You can get the Repository Access Token by selecting the Downloads and Scripts option here from the CAS Manager support site. Copy the token in the download script and replace in the command above.

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

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

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

<table>
<thead>
<tr>
<th>repo id</th>
<th>repo name</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta</td>
<td>teradici-cas-manager-beta</td>
</tr>
<tr>
<td>beta-noarch</td>
<td>teradici-cas-manager-beta-noarch</td>
</tr>
<tr>
<td>source</td>
<td>teradici-cas-manager-beta-source</td>
</tr>
</tbody>
</table>
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. Run the following commands to install and SELinux:

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

   ```
   sudo yum 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 yum install -y https://github.com/k3s-io/k3s-selinux/releases/download/v0.2.stable.1/k3s-selinux-0.2-1.el7_8.noarch.rpm
   ```

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

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

3. Install CAS Manager

Run the following command to install CAS Manager:

```
sudo yum 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 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
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.

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 Yum Repo Management

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

**Locking CAS Manager version in the yum repo**

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

```bash
sudo yum config-manager --set-disabled teradici-cas-manager*
```

You can confirm the settings by running the following command:

```bash
yum repolist teradici-cas-manager*
```

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

<table>
<thead>
<tr>
<th>repo id</th>
<th>repo name</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>teradici-cas-manager</td>
<td>teradici-cas-manager</td>
<td>disabled</td>
</tr>
<tr>
<td>teradici-cas-manager-noarch</td>
<td>teradici-cas-manager-noarch</td>
<td>disabled</td>
</tr>
<tr>
<td>teradici-cas-manager-source</td>
<td>teradici-cas-manager-source</td>
<td>disabled</td>
</tr>
</tbody>
</table>

**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.

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.

**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:
   ```bash
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:
   ```bash
   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:

<table>
<thead>
<tr>
<th>CAS Manager Version</th>
<th>Vault Version</th>
<th>MongoDB Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.03</td>
<td>1.4.2</td>
<td>4.0.8</td>
</tr>
<tr>
<td>21.07</td>
<td>1.7.1</td>
<td>4.2.14</td>
</tr>
<tr>
<td>21.10</td>
<td>1.7.1</td>
<td>4.2.14</td>
</tr>
</tbody>
</table>

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.
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.

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/.
• 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 documentation, 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 to configure and add the RHEL repository, select the Downloads and Scripts option here from the CAS Manager support site.

Rocky Linux Repository

To force the specific repository for Rocky Linux 8, run the following command:

```bash
curl -1sLf https://dl.teradici.com/<RepositoryAccessToken>/cas-manager/cfg/setup/bash.rpm.sh | sudo -E distro=el codename=8 bash
```
You can get the **Repository Access Token** by selecting the Downloads and Scripts option [here](#) from the CAS Manager support site. Copy the token in the download script and replace in the command above.

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

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

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

<table>
<thead>
<tr>
<th>repo id</th>
<th>repo name</th>
</tr>
</thead>
<tbody>
<tr>
<td>teradici-cas-manager-beta</td>
<td>teradici-cas-manager-beta</td>
</tr>
<tr>
<td>teradici-cas-manager-beta-noarch noarch</td>
<td>teradici-cas-manager-beta</td>
</tr>
<tr>
<td>teradici-cas-manager-beta-source source</td>
<td>teradici-cas-manager-beta</td>
</tr>
</tbody>
</table>

### 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. Run the following commands to install and SELinux:

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

   ```
   sudo yum 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 yum install -y https://github.com/k3s-io/k3s-selinux/releases/download/v0.2.stable.1/k3s-selinux-0.2-1.el7_8.noarch.rpm
   ```

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

Run the following command to install CAS Manager:

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

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 installation has been successful, as seen in the image above.
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
   }
   ```

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:

   ```
   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/.

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
   ```

© 2021 Teradici
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"

```javascript
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:

```
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.
```

© 2021 Teradici
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.

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:

   ```json
   {
     "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.

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 Yum Repo Management

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

**Locking CAS Manager version in the yum repo**

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

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

You can confirm the settings by running the following command:

```
yum repolist teradici-cas-manager*
```

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

```
<table>
<thead>
<tr>
<th>repo id</th>
<th>repo name</th>
<th>repo status</th>
</tr>
</thead>
<tbody>
<tr>
<td>teradici-cas-manager</td>
<td>teradici-cas-manager</td>
<td>disabled</td>
</tr>
<tr>
<td>teradici-cas-manager-noarch</td>
<td>teradici-cas-manager-noarch</td>
<td>disabled</td>
</tr>
<tr>
<td>teradici-cas-manager-source</td>
<td>teradici-cas-manager-source</td>
<td>disabled</td>
</tr>
</tbody>
</table>
```

**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.
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 yum 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 Yum Repository**

There are two cases you need to update CAS Manager yum 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 yum repository as part of running an in-place upgrade.

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

   ```
   sudo rm /etc/yum.repos.d/teradici-cas-manager.repo
   ```
3. To access the scripts to configure and add the RHEL repository, select the Downloads and Scripts option [here](#) from the CAS Manager support site.

   **Rocky Linux Repository**

   To force the specific repository for Rocky Linux 8, run the following command:

   ```
   curl -1sf https://dl.teradici.com/<RepositoryAccessToken>/cas-manager/cfg/setup/bash.rpm.sh | sudo -E distro=el codename=8 bash
   ```
You can get the **Repository Access Token** by selecting the Downloads and Scripts option here from the CAS Manager support site. Copy the token in the download script and replace in the command above.

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

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

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

<table>
<thead>
<tr>
<th>repo id</th>
<th>repo name</th>
</tr>
</thead>
<tbody>
<tr>
<td>teradici-cas-manager-beta</td>
<td>teradici-cas-manager-beta</td>
</tr>
<tr>
<td>teradici-cas-manager-beta-noarch</td>
<td>teradici-cas-manager-beta-noarch</td>
</tr>
<tr>
<td>teradici-cas-manager-beta-source</td>
<td>teradici-cas-manager-beta-source</td>
</tr>
</tbody>
</table>

### 2. Unlocking CAS Manager Version

If you have skipped above step **Update CAS Manager Yum 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 yum repo:

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

You can confirm the settings with the following command:

```
yum repolist teradici-cas-manager*
```

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

<table>
<thead>
<tr>
<th>repo id</th>
<th>repo name</th>
<th>status</th>
</tr>
</thead>
<tbody>
<tr>
<td>teradici-cas-manager</td>
<td>teradici-cas-manager</td>
<td>enabled</td>
</tr>
<tr>
<td>teradici-cas-manager-noarch</td>
<td>teradici-cas-manager-noarch</td>
<td>enabled</td>
</tr>
</tbody>
</table>
3. Upgrade CAS Manager via Yum

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

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 yum 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:

   ```
   yum 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 
     ```bash
     --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:
       ```bash
       sudo systemd-resolve --flush-caches
       ```
   - If you installed your Connector with 
     ```bash
     --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:
       ```bash
       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.
3. Delete the CAS Manager VM.

```
sudo yum remove -y cas-manager-selinux cas-manager
sudo rm -rf /opt/teradici  # Remove the cas-manager files
```
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.
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":

### 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 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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>token</td>
<td>&lt;your token is here&gt;</td>
</tr>
<tr>
<td>token_accessor</td>
<td>&lt;your token accessor is here&gt;</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>token_duration</th>
<th>768h</th>
</tr>
</thead>
<tbody>
<tr>
<td>token_renewable</td>
<td>true</td>
</tr>
<tr>
<td>token_policies</td>
<td>[&quot;casm&quot; &quot;default&quot;]</td>
</tr>
<tr>
<td>identity_policies</td>
<td>[]</td>
</tr>
<tr>
<td>policies</td>
<td>[&quot;casm&quot; &quot;default&quot;]</td>
</tr>
</tbody>
</table>
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:

   ```json
   {
   "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:

   ```bash
   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:

   ```json
   {
     "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:

   ```bash
   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:

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

# It will show the following if Vault is healthy:

```
[2021-01-25T22:49:02Z] INFO .. Connections:
```

# 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
```
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 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:

```json
{
  "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 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.

---

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

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

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`.

---

**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.

---

© 2021 Teradici
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:

   ```bash
   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:

   ```bash
   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:

   ```bash
   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:

   ```bash
   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:

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

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.
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:

```
    $ 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
```

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:

```
    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
```

© 2021 Teradici
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:

   ```bash
   # 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:

   ```bash
   # 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

```
#!/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=$(kubectl get secret clustervaulttoken --template={{.data.address}} | base64 -d);
export VAULT_TOKEN=$(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
EOF)
```

```
# 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
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" : true}}'
```
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:

```bash
false }
sleep 60
```

/usr/local/bin/kubectl delete secret clustervaultttoken

Once this is deleted you will no longer be able to access data from the internal Vault.
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. Connector is software that runs within an Ubuntu server and enables secure connectivity between users and the remote workstations by eliminating the need for a dedicated VPN by providing NAT services for external users.

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.
System Requirements

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:

```bash
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](#).
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:

```bash
./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:

```bash
./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.
Multi-Factor Authentication

When you install the Connector you can specify whether the PCoIP session uses Multi-Factor Authentication (MFA) during authentication or not. The Connector can be integrated with your RADIUS server. To do this you will need to provide the following information during the Connector installation:

- The FQDN or IP address of the RADIUS server.
- The RADIUS server port. If this port is not specified the default port (1812) will be used.
- The shared secret used for configuring RADIUS authentication.

If you do not enable MFA when installing the Connector, you can enable it later when performing an update, see Updating the Cloud Access Connector. For more information on Connector MFA, see Multi-Factor Authentication.
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

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

- Downloading the Connector installer files.
- Obtaining an authorization token.
- Installing the Connector.

Prerequisite Steps

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

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

1. 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 /
```

---

2. Obtaining the 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.

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.

3. 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.

3.1 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. For more information on configuring custom trusted certificates for CAS Manager, see TLS Certificates.

• The `--casm-ca-cert` flag can be 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.

### 3.2 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.
Installing the Connector for Testing with CAS Manager

To install the Connector with MFA enabled and in insecure mode for testing, you would run this command (note that we are providing the --enable-mfa flag but not the RADIUS server information, so prompts will appear to collect it):

```
sudo ./cloud-access-connector install -t $token --enable-mfa --self-signed
```

When the installer completes, the IP address of the Connector will be displayed and you will be directed to go to https://cas.teradici.com to begin managing your deployments, connectors and remote workstations.

**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

**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:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--casm-url</td>
<td>String</td>
<td>Required for CAS Manager, Specifies the CAS Manager URL that the Connector connects to. If this is not specified it will point to <a href="https://cas.teradici.com">https://cas.teradici.com</a> by default, which is the URL for CAS Manager as a Service.</td>
</tr>
<tr>
<td>--casm-ca-cert</td>
<td>String</td>
<td>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.</td>
</tr>
</tbody>
</table>
### 3. Installing the Connector

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--casm-insecure</code></td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td><code>--ldaps-ca-cert</code></td>
<td>String</td>
<td>Enables users to supply a CA certificate for the connection to Active Directory over LDAPS.</td>
</tr>
<tr>
<td><code>--self-signed</code></td>
<td>String</td>
<td>Installs the Connector with self-signed certificates. This mode is not secure and is intended for testing. The <code>--insecure</code> flag is still supported.</td>
</tr>
</tbody>
</table>

**Connector**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--token (-t)</code></td>
<td>String</td>
<td>Required. The token generated for CAS Manager.</td>
</tr>
<tr>
<td><code>--accept-policies</code></td>
<td>-</td>
<td>Automatically accept the EULA and Privacy Policy.</td>
</tr>
<tr>
<td><code>--force-install</code></td>
<td>String</td>
<td>Replaces any existing Connector installation.</td>
</tr>
<tr>
<td><code>--debug</code></td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td><code>--local-license-server-url</code></td>
<td>String</td>
<td>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: <code>--local-license-server-url http://10.10.10.7070/request</code> For more information on the PCoIP License Server, see PCoIP License Server.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--add-pool-group</code></td>
<td>String</td>
<td>Specifies one or more Active Directory groups, by entering the distinguished name (DN), to be assigned to pools for remote workstation management (eg, <code>--pool-group 'CN=GroupPool1,CN=Users,DC=sample,DC=com' --pool-group 'CN=GroupPool2,CN=Users,DC=sample,DC=com'</code>). 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 <code>adconfig</code>. Example: <code>sudo ./cloud-access-connector adconfig --add-pool-group</code>.</td>
</tr>
<tr>
<td><code>--setup-docker-image</code></td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td><code>--docker-registry</code></td>
<td>String</td>
<td>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 <code>docker.cloudsmith.io/teradici/cloud-access-connector</code> 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.</td>
</tr>
<tr>
<td><code>--prune-image</code></td>
<td>Boolean</td>
<td>Removes all unused docker images on this machine to reclaim more disk space. <strong>Warning:</strong> This command will remove all unused images under Connector and other services, if any. This is equivalent to the <code>docker image prune</code> command.</td>
</tr>
</tbody>
</table>

**Firewall**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--https-proxy</code></td>
<td>String</td>
<td>Specify the URL for a HTTPS proxy (overrides related proxy settings in environment variables)</td>
</tr>
<tr>
<td><code>--connector-network-cidr</code></td>
<td>String</td>
<td>This is the CIDR to use for the Connector's docker network. The default docker network subnet is 10.101.0.0/16.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--internal-client-cidr</td>
<td>String</td>
<td>The CIDR for PCoIP Clients that connect to remote workstations directly. It is possible to specify multiple networks.</td>
</tr>
<tr>
<td>--external-client-cidr</td>
<td>String</td>
<td>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 networks.</td>
</tr>
</tbody>
</table>

**PCoIP Software Client**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--retrieve-agent-state</td>
<td>Boolean</td>
<td>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 <em>In Session, Ready, Starting, Stopping, Stopped</em> and <em>Unknown</em>. The value of this flag can either be true or false.</td>
</tr>
<tr>
<td>--show-agent-state</td>
<td>Boolean</td>
<td>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 <code>--retrieve-agent-state</code> flag to false will result in no agent state displaying.</td>
</tr>
<tr>
<td>--external-pcoip-ip</td>
<td>String</td>
<td>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 <a href="#">Enabling External Network Access</a>.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--domain</td>
<td>String</td>
<td>The AD domain that remote workstations will join.</td>
</tr>
<tr>
<td>--sa-user</td>
<td>String</td>
<td>The Active Directory service account username.</td>
</tr>
<tr>
<td>--sa-password</td>
<td>String</td>
<td>The Active Directory service account password.</td>
</tr>
<tr>
<td>--domain-controller</td>
<td>String</td>
<td>Specifies one or more domain controllers to use with the Connector.</td>
</tr>
<tr>
<td>--users-filter</td>
<td>String</td>
<td>The filter to search for users within Active Directory. Specify multiple filters with multiple options. Default user filter: (&amp;(objectCategory=person)(objectClass=user)).</td>
</tr>
<tr>
<td>--computers-filter</td>
<td>String</td>
<td>The filter to search for computers within Active Directory. Specify multiple filters with multiple options. Default computer filter: (&amp;(primaryGroupID=515)(objectCategory=computer)).</td>
</tr>
<tr>
<td>--users-dn</td>
<td>StringArray</td>
<td>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.</td>
</tr>
<tr>
<td>--computers-dn</td>
<td>StringArray</td>
<td>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).</td>
</tr>
<tr>
<td>--sync-interval</td>
<td>uint8</td>
<td>The interval (in minutes) for how often to sync AD users and computers with the CASM Service.</td>
</tr>
</tbody>
</table>

**MFA**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--enable-mfa</td>
<td>String</td>
<td>Installs with multi-factor authentication enabled.</td>
</tr>
<tr>
<td>--radius-server</td>
<td>String</td>
<td>The FQDN or IP address of the RADIUS server to use for MFA. This flag is optional.</td>
</tr>
</tbody>
</table>
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 2\textsuperscript{nd} 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 immediately. 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
Updating and Reconfiguring the Connector

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 occurred 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:

```bash
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

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:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--casm-url</code></td>
<td>String</td>
<td>Required for CAS Manager, Specifies the CAS Manager URL that the Connector connects to. If this is not specified it will point to <code>https://cas.teradici.com</code> by default, which is the URL for CAS Manager as a Service.</td>
</tr>
<tr>
<td><code>--casm-ca-cert</code></td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--casm-insecure</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>--ldaps-ca-cert</td>
<td>String</td>
<td>Enables users to supply a CA certificate for the connection to Active Directory over LDAPS.</td>
</tr>
<tr>
<td>--self-signed</td>
<td>String</td>
<td>Installs the Connector with self-signed certificates. This mode is not secure and is intended for testing. The <code>--insecure</code> flag is still supported.</td>
</tr>
</tbody>
</table>

**Connector**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--token (<code>-t</code>)</td>
<td>String</td>
<td>Required. The token generated for CAS Manager.</td>
</tr>
<tr>
<td>--accept-policies</td>
<td></td>
<td>Automatically accept the EULA and Privacy Policy.</td>
</tr>
<tr>
<td>--force-install</td>
<td>String</td>
<td>Replaces any existing Connector installation.</td>
</tr>
<tr>
<td>--debug</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>--local-license-server-url</td>
<td>String</td>
<td>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: <code>--local-license-server-url http://10.10.10.7070/request</code>. For more information on the PCoIP License Server, see PCoIP License Server.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--add-pool-group</td>
<td>String</td>
<td>Specifies one or more Active Directory groups, by entering the distinguished name (DN), to be assigned to pools for remote workstation management (e.g., --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 <code>adconfig</code>. Example: <code>sudo ./cloud-access-connector adconfig --add-pool-group</code></td>
</tr>
<tr>
<td>--setup-docker-image</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>--docker-registry</td>
<td>String</td>
<td>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 <code>docker.cloudsmith.io/teradici/cloud-access-connector</code> 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.</td>
</tr>
<tr>
<td>--prune-image</td>
<td>Boolean</td>
<td>Removes all unused docker images on this machine to reclaim more disk space. <strong>Warning</strong>: This command will remove all unused images under Connector and other services, if any. This is equivalent to the <code>docker image prune</code> command.</td>
</tr>
<tr>
<td><strong>Firewall</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--https-proxy</td>
<td>String</td>
<td>Specify the URL for a HTTPS proxy (overrides related proxy settings in environment variables)</td>
</tr>
<tr>
<td>--connector-network-cidr</td>
<td>String</td>
<td>This is the CIDR to use for the Connector's docker network. The default docker network subnet is 10.101.0.0/16.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--internal-client-cidr</code></td>
<td>String</td>
<td>The CIDR for PCoIP Clients that connect to remote workstations directly. It is possible to specify multiple <code>--internal-client-cidr</code> networks.</td>
</tr>
<tr>
<td><code>--external-client-cidr</code></td>
<td>String</td>
<td>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 <code>--external-client-cidr</code> networks.</td>
</tr>
</tbody>
</table>

**PCoIP Software Client**

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--retrieve-agent-state</code></td>
<td>Boolean</td>
<td>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.</td>
</tr>
<tr>
<td><code>--show-agent-state</code></td>
<td>Boolean</td>
<td>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 <code>--retrieve-agent-state</code> flag to false will result in no agent state displaying.</td>
</tr>
<tr>
<td><code>--external-pcoip-ip</code></td>
<td>String</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Domain**
<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--domain</td>
<td>String</td>
<td>The AD domain that remote workstations will join.</td>
</tr>
<tr>
<td>--sa-user</td>
<td>String</td>
<td>The Active Directory service account username.</td>
</tr>
<tr>
<td>--sa-password</td>
<td>String</td>
<td>The Active Directory service account password.</td>
</tr>
<tr>
<td>--domain-controller</td>
<td>String</td>
<td>Specifies one or more domain controllers to use with the Connector.</td>
</tr>
<tr>
<td>--users-filter</td>
<td>String</td>
<td>The filter to search for users within Active Directory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify multiple filters with multiple options. Default user filter: (&amp;(objectCategory=person)(objectClass=user)).</td>
</tr>
<tr>
<td>--computers-filter</td>
<td>String</td>
<td>The filter to search for computers within Active Directory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify multiple filters with multiple options. Default computer filter: (&amp;(primaryGroupID=515)(objectCategory=computer)).</td>
</tr>
<tr>
<td>--users-dn</td>
<td>StringArray</td>
<td>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.</td>
</tr>
<tr>
<td>--computers-dn</td>
<td>StringArray</td>
<td>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).</td>
</tr>
<tr>
<td>--sync-interval</td>
<td>uint8</td>
<td>The interval (in minutes) for how often to sync AD users and computers with the CASM Service.</td>
</tr>
<tr>
<td>MFA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--enable-mfa</td>
<td>String</td>
<td>Installs with multi-factor authentication enabled.</td>
</tr>
<tr>
<td>--radius-server</td>
<td>String</td>
<td>The FQDN or IP address of the RADIUS server to use for MFA. This flag is optional.</td>
</tr>
</tbody>
</table>
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 requirements 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.
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:

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>RAM</th>
<th>Estimated Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>2vCPU</td>
<td>7.5 GB RAM</td>
<td>~ 365 Mbit/s</td>
</tr>
<tr>
<td>4vCPU</td>
<td>15 GB RAM</td>
<td>~ 830 Mbit/s</td>
</tr>
<tr>
<td>8vCPU</td>
<td>30 GB RAM</td>
<td>~ 1100 Mbit/s</td>
</tr>
</tbody>
</table>

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.
# 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

<table>
<thead>
<tr>
<th>Component</th>
<th>Allow</th>
<th>Port/Protocol</th>
<th>Source/Destination Component</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector</td>
<td>Inbound</td>
<td>443 TCP</td>
<td>From PCoIP Clients and administrative web browsers.</td>
<td>For users to negotiate connections to their remote workstations. For accessing the Management Interface for (legacy) management of CAS Manager.</td>
</tr>
<tr>
<td>Connector</td>
<td>Outbound</td>
<td>443 TCP</td>
<td>To CAM Service, PCoIP Cloud License Server and to SumoLogic.</td>
<td>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.</td>
</tr>
<tr>
<td>Connector</td>
<td>Outbound</td>
<td>60443 TCP</td>
<td>To remote workstations.</td>
<td>Prepares PCoIP Agents for a new user session.</td>
</tr>
<tr>
<td>Connector</td>
<td>Inbound</td>
<td>4172 TCP/UDP</td>
<td>From PCoIP Clients.</td>
<td>For PCoIP Sessions with users that are outside of the corporate network.</td>
</tr>
<tr>
<td>Connector</td>
<td>Outbound</td>
<td>4172 TCP/UDP</td>
<td>To remote workstations.</td>
<td>For PCoIP Sessions with users that are outside of the corporate network.</td>
</tr>
<tr>
<td>Connector</td>
<td>Outbound</td>
<td>636 TCP</td>
<td>To Domain Controllers.</td>
<td>To authenticate users, and query user and computer information.</td>
</tr>
</tbody>
</table>
**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.

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

The following steps outline how to use the Connector 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
```

© 2021 Teradici
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:

```bash
# 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

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

```bash
$ export https_proxy=$http_proxy
$ curl https://cam.teradici.com
```

or through the command line option:

```bash
$ ./cloud-access-connector install --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:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--users-dn</code></td>
<td>String</td>
<td>The base DN to search for users within Active Directory. This option may be specified multiple times to provide multiple DNs.</td>
</tr>
<tr>
<td><code>--computers-dn</code></td>
<td>String</td>
<td>The base DN to search for computers within Active Directory. This option may be specified multiple times to provide multiple DNs.</td>
</tr>
<tr>
<td><code>--sync-interval</code></td>
<td>String</td>
<td>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.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--cam-insecure</code></td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>Flag</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--add-pool-group</td>
<td>String</td>
<td>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.</td>
</tr>
<tr>
<td>--remove-pool-group</td>
<td>String</td>
<td>Removes specified pool Active Directory group by its DN.</td>
</tr>
<tr>
<td>--clear-pools-groups</td>
<td>String</td>
<td>Clears all pools Active Directory groups. This operation is exclusive and cannot be combined with --remove-pool-group or --add-pool-group.</td>
</tr>
<tr>
<td>--get-cam-settings</td>
<td>String</td>
<td>Prints all CAS Manager as a Service settings to Admin console.</td>
</tr>
</tbody>
</table>
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.

**Applicable for CAS Manager as a Service Only**

Currently, the floating workstation assignment feature is only available for Connectors deployed to CAS Manager as a Service. It is not currently supported with CAS Manager.

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**.
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.

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.

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.
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:

```bash
# 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.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>vault-type</td>
<td>string</td>
<td>Specifies the type of secret store that CAS Manager should use. Currently, &quot;vault&quot; is the only supported value for this parameter.</td>
<td>Required for updating Vault configuration.</td>
</tr>
<tr>
<td>vault-url</td>
<td>string</td>
<td>URL of the Vault server.</td>
<td>Required for updating Vault configuration.</td>
</tr>
<tr>
<td>vault-secret-path</td>
<td>string</td>
<td>Vault secret path where secrets are stored.</td>
<td>Required for updating Vault configuration.</td>
</tr>
<tr>
<td>vault-token</td>
<td>string</td>
<td>Token used to authenticate requests to the Vault server.</td>
<td>Required for updating Vault configuration.</td>
</tr>
<tr>
<td>vault-ca-cert-file</td>
<td>string</td>
<td>Path to the file containing a PEM-formatted CA certificate that will be used to validate the Vault server's certificate.</td>
<td>Required if the Vault server is using self-signed certificates.</td>
</tr>
<tr>
<td>vault-skip-verify-cert</td>
<td>boolean</td>
<td>If true, CAS Manager will not validate the Vault server's TLS certificate. This is not secure and is not recommended for production deployments.</td>
<td>Not required. Defaults to &quot;false&quot;.</td>
</tr>
<tr>
<td>db-connection-string</td>
<td>string</td>
<td>URL of the MongoDB server.</td>
<td>Required for updating MongoDB configuration.</td>
</tr>
<tr>
<td>db-enable-tls</td>
<td>boolean</td>
<td>If false, requests to MongoDB will not be encrypted. Setting this parameter to false is not secure and is not recommended for production deployments.</td>
<td>Not required. Defaults to &quot;true&quot;.</td>
</tr>
<tr>
<td>db-skip-verify-cert</td>
<td>boolean</td>
<td>If true, CAS Manager will not validate the MongoDB server's TLS certificate. This is not secure and is not recommended for production deployments.</td>
<td>Not required. Defaults to &quot;false&quot;.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Description</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>db-ca-cert-file</td>
<td>string</td>
<td>Path to the file containing a PEM-formatted CA certificate that will be used to validate the MongoDB server's certificate.</td>
<td>Required if the MongoDB server is using self-signed certificates.</td>
</tr>
<tr>
<td>tls-key-file</td>
<td>string</td>
<td>Path to the file containing a PEM-formatted TLS key that will be used by CAS Manager.</td>
<td>Required for updating TLS certificates used by CAS Manager.</td>
</tr>
<tr>
<td>tls-cert-file</td>
<td>string</td>
<td>Path to the file containing PEM-formatted TLS certificate that will be used by CAS Manager.</td>
<td>Required for updating TLS certificates used by CAS Manager.</td>
</tr>
<tr>
<td>skip-validate-reg-code</td>
<td>boolean</td>
<td>If true, skip validating PCoIP registration code when creating or updating a deployment.</td>
<td>Not required unless CAS Manager is blocking all internet traffic. Defaults to &quot;false&quot;.</td>
</tr>
</tbody>
</table>
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.

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:

   ```json
   {
     "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]}.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
```
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 Active 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.

<table>
<thead>
<tr>
<th>CAS Manager Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Active Directory Configuration Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only the adminUser has the permissions to configure the Active Directory with CAS Manager from the CAS Manager Admin Console.</td>
</tr>
</tbody>
</table>

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:

```json
{
  "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 patterns to ingest. We only care about the logs. Leave this as is.
      "blacklist": [
      ],
      "sourceType": "LocalFile",
      "alive": false
    }
  ]
}
```
3. Download the Sumo Logic Collector:

```bash
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:

  ```bash
  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:

  ```bash
  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.
Multi-Factor Authentication Integration

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
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.
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.

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.

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:

```json
{
   "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.

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
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.

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 initiated 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]

5. Enter the Active Directory information for the remote workstation. The service account must have permission to join computers to the domain.

6. 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.

---

**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.
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 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. Once the user logs in, they are automatically and persistently assigned a remote workstation within the pool.

Creating a Workstation Pool

The following steps outline how to create a workstation pool and add remote workstations and users to it:

1. Click on Workstation Pools from the Admin Console sidebar.
2. Click the + icon to create a new workstation pool.
3. Enter a Pool Name, and click **Create**. You will now have a workstation pool where you can add remote workstations, users and user groups.

![Test Pool](image)

4. From the workstation pool page click **ADD REMOTE WORKSTATIONS** and search for an available workstation to add to the pool and click **SAVE**. 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.

5. From the workstation pool page click **ADD USERS** and/or **ADD GROUPS** search for available users and user groups to add to the pool and click **SAVE**.

You can add multiple workstation pools to specific deployments. Each workstation pool will list the remote workstations, users and user groups within that pool, as well as component information for these remote workstations and users.
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.
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.
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.

**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.

---

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.
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:

   ![User Entitlements](image)

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 the 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**.

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 successful or not.
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

<table>
<thead>
<tr>
<th>User Name</th>
<th>User GUID</th>
<th>Created On</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name-587-1561591856</td>
<td>1-587-1561591856</td>
<td>Aug 21, 2019 08:18 AM PDT</td>
</tr>
</tbody>
</table>

### GROUPS

- 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=Schemas Admins, CN=Users, DC=example, DC=com
- CN=Administrators, CN=Build, DC=example, DC=com

### ENTITLED WORKSTATIONS

<table>
<thead>
<tr>
<th>Name</th>
<th>Power State</th>
<th>Assigned On</th>
</tr>
</thead>
<tbody>
<tr>
<td>workstation1</td>
<td>Running</td>
<td>Aug 24, 2019 05:06 PM PDT</td>
</tr>
<tr>
<td>workstation2</td>
<td>Running</td>
<td>Aug 14, 2019 05:06 PM PDT</td>
</tr>
</tbody>
</table>
This gives your an overview of a specific user’s entitlements and deployment information and can be useful for troubleshooting issues.
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 occurred. 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.

© 2021 Teradici
DNS Name Resolution Configuration

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.

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
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
```
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-yyyyymmddThhmmssZ.tar.gz
```

`yyyyymmddThhmmssZ` 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`.

© 2021 Teradici
- `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 wwauxfxZ`.
- `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`.
  - The seinux 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`.
- The pods folder will contain a describe pod file for each pod.
- The logs folder will display the log files for each pod.
- The services folder will contain a Describe Service file for each service.
- The secrets folder will display the output of the command `kubectl get secrets`. 


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"
- "machinemgmtpoolworker"
- "machinemonitor"
- "machinemonitorworker"
- "poolmgmt"
- "poolmgmtworker"
• "secretmgmt"
• "redis"
• "resourcetemplates"
• "resourcetemplatestore"
• "userentitlement"
• "userentitlementworker"
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:

Attempted 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
Running older installer versions

If the Connector installer was installed before December 11, it may generate some of the following errors:

Error 3

```shell
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

```shell
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 curl failed to verify the legitimacy of the server and therefore could not establish a secure connection to it. To learn more about this situation and how to fix it, please visit the web page mentioned above.
2020-12-22T20:07:09Z  ERROR exit status 60
```

Error 5

```shell
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
```

© 2021 Teradici
for the target resource or is not willing to disclose that one exists.</p>

---

**Apache Tomcat/8.5.59**

---

**ERROR exit status 1**

---

**Error 6**

---


---

**ERROR exit status 1**
Cloud Access Connector Connectivity Issues

CAS Manager provides some diagnostic checks that can be used to troubleshoot the cause of issues you may be experiencing with your Connector. Run the following command:

```bash
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:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--rw</td>
<td>The Remote Workstation FQDN</td>
</tr>
<tr>
<td>-ad</td>
<td>Verify connectivity to currently configured Active Directory server</td>
</tr>
<tr>
<td>-h</td>
<td>help for diagnose</td>
</tr>
<tr>
<td>--debug</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 ./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 occurred 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
```

Connector Upgrade and Diagnose Issues

Several previous versions of Connector installers are no longer compatible with our latest infrastructure 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.
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 occurred 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:
It will also be logged by the installer:

```
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
...
```

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
```
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:
   
   ```bash
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:
   
   ```json
   {
   "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:
   
   ```bash
   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:
   
   ```markdown
   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.
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.