

PCoIP Connection Manager and Security Gateway Administrators' Guide

24.03

Table of contents

Overview	5
HP PCoIP Connection Manager and Security Gateway	5
About the PCoIP Connection Manager	5
About the PCoIP Security Gateway	5
Establishing a PCoIP Connection With the Connection Manager and Security Gateway	6
Deployment Scenarios	7
What's New in This Release	8
System Requirements	9
Installation Prerequisites	9
PCoIP Connection Manager and PCoIP Security Gateway Performance Limits	11
PCoIP Connection Manager Limits	11
PCoIP Security Gateway Limits	11
System Planning	13
Session Establishment	13
Load Balancing	14
Configuring Firewalls	15
Configuring Docker Network	18
Installing	19
Installing for Online Environments	19
Before You Begin	19
Install PCoIP Modern Connection Manager and PCoIP Security Gateway	20
Installation Flags and Options	22
About Docker	27
Installing for Offline Environments	30
Before You Begin	30
Downloading Offline Installation Bundle	30

Installing PCoIP Connection Manager and the PCoIP Security Gateway	31
Installation Flags and Options	32
Enabling or Disabling the PCoIP Security Gateway	37
Creating the Installation Bundle	38
Updating the PCoIP Connection Manager and PCoIP Security Gateway	40
Updating an Online Installation	41
Uninstalling PCoIP Connection Manager and PCoIP Security Gateway	42
Configuring	44
Configuring the PCoIP Connection Manager and PCoIP Security Gateway	44
Configuration Flags and Options	45
Security and Certificates	49
Security Considerations	49
Creating, Installing, and Managing Certificates	50
Federated Authentication	56
Federated Authentication using OAuth2	56
Single Sign-On	67
Troubleshooting Federated Authentication	90
Reference	114
Using a PCoIP License Server with the Connection Manager	114
Using a PCoIP License Server with the PCoIP Connection Manager	114
PCoIP Connection Manager and Security Gateway RPM Package Contents	115
TLS Cipher Suites	116
TLS Versions	116
PCoIP Connection Manager TLS Cipher Suites	116
PCoIP Security Gateway Supported TLS Cipher Suites	116
Troubleshooting	118
Troubleshooting Connectivity Issues	118
Network Connectivity Problems	118

Troubleshooting Certificate Errors	130
Error messages	130
Troubleshooting Error Messages	131
PCoIP Connection Manager and Security Gateway Log Files	132
Log Maintenance	132
Sensitive Information in Logs	132
Log File Locations	132
Log Verbosity	133
Contacting Support	135
The HP Community Forum	135
Generating a Support Bundle	136

Overview

HP PCoIP Connection Manager and Security Gateway

The *PCoIP Connection Manager* and the *PCoIP Security Gateway* are components of HP Anyware, and can be deployed together as a set or individually. Multiple instances of the Connection Manager and/or the Security Gateway can be deployed to handle mixed LAN and WAN access points, enable security gateway failover, or for scaling large systems.

Components in this release

The PCoIP Connection Manager and Security Gateway 24.03 is a combined release containing:

- PCoIP Connection Manager 24.03
- PCoIP Security Gateway 23.04

About the PCoIP Connection Manager

The *PCoIP Connection Manager* enables connections between PCoIP clients and PCoIP agents installed on remote desktops. It uses a required third-party connection broker to authenticate users, query available desktops and applications, and then establish a PCoIP connection between the client and the selected desktop.

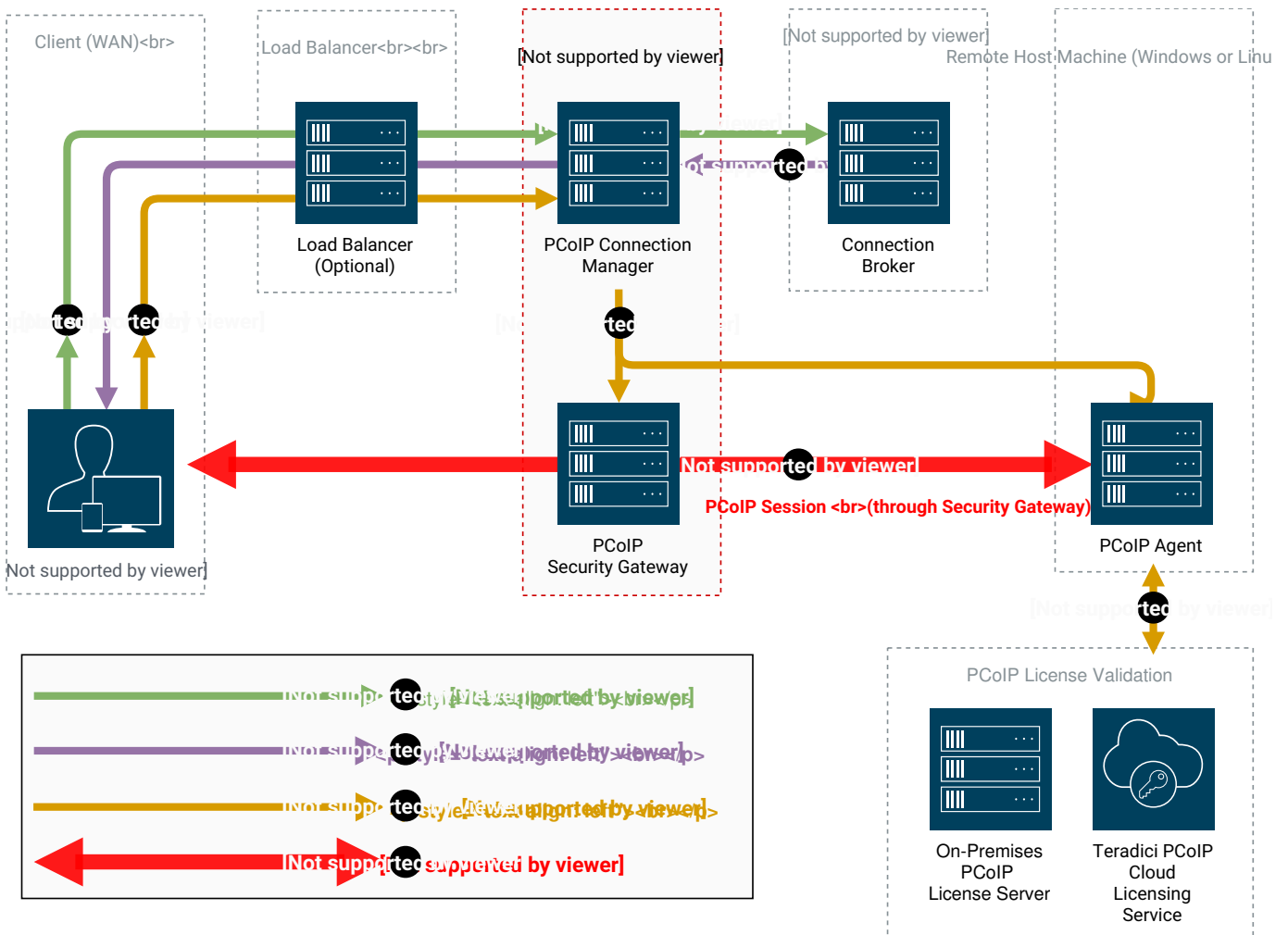
About the PCoIP Security Gateway

The *PCoIP Security Gateway* enables WAN users to securely access their remote desktops via the Internet without a VPN connection. You can optionally deploy multiple PCoIP Security Gateways so that if the gateway being used by a PCoIP session becomes unavailable, the session is automatically transferred to the next available gateway. To use this feature, configure the Connection Manager using the `--external-sg-ip` flag with the addresses of the failover security brokers.

Note
The PCoIP Security Gateway is not required for LAN access.

Establishing a PCoIP Connection With the Connection Manager and Security Gateway

The diagram shown next illustrates a brokered connection to the PCoIP host machine using the PCoIP Connection Manager and the PCoIP Security Gateway.



⚠ Caution: A dedicated server is strongly recommended

Since the PCoIP Connection Manager is a component that handles authentication data for users connecting to virtual desktops, we strongly recommend installing the PCoIP Connection Manager and PCoIP Security Gateway on a dedicated server that is accessible only by authorized system administrators according to your organization's security policy.

Deployment Scenarios

Depending on your deployment scenario, you can install the PCoIP Connection Manager with the PCoIP Security Gateway disabled.

- **All your desktops are on a LAN (internal access only):** you may only need to install one PCoIP Connection Manager. Since a PCoIP Security Gateway isn't required for LAN connections, you can optionally disable it.
- **All your desktops are on a WAN:** Install one PCoIP Connection Manager, and enabling one or more Security Gateways. The Connection Manager handles PCoIP Connection establishment and the Security Gateway(s) secures the PCoIP session across the public internet.
- **Your desktops are on both a LAN and WAN:** We recommend installing at least two groups of connection managers; one for internal access with the PCoIP Security Gateway disabled, and one for external access with one or more PCoIP Security Gateway(s) enabled. You can set up the DNS so that internal and external users are routed to the appropriate connection manager.
- **If you are exceeding the [system specifications](#) or have high availability requirements:** If you serve a large number of desktops, or require high availability, install additional connection managers and implement load balancing.

What's New in This Release

Release 24.03 of the PCoIP Connection Manager and PCoIP Security Gateway includes the following:

System Requirements

The minimum system requirements for a PCoIP Connection Manager and PCoIP Security Gateway are:

- 2 or more CPUs or vCPUs, 2.5 GHz or higher
- 4 GB of RAM
- 4 GB of swap space
- 10 GB of free disk space in var directory

Supported operating systems:

- RHEL 8
- Rocky Linux 8

If the connection broker is configured to identify resources by host name, then DNS must be available in PCoIP Connection Manager and the PCoIP Broker.

Installation Prerequisites

The PCoIP Connection Manager and PCoIP Security Gateway depends on the following packages:

- **Docker 20.10.0** or higher

Install or update OpenSSL version to **OpenSSL 3.0** or higher.

Project dependencies must be installed on the production machine *before* installing the PCoIP Connection Manager and PCoIP Security Gateway.

Caution: Dependencies in offline environments

If your deployment will be running in an environment that is not connected to the public internet (a *dark site*), you must download the package dependencies, transfer them to the production machine, and install them before installing the PCoIP Connection Manager and PCoIP Security Gateway.

i Open SSL Minimum Requirements

The following procedures use openssl to create and manage certificates. If you use another tool, adapt these instructions accordingly. The minimum Open SSL version on your virtual machine is 3.0.

PCoIP Connection Manager and PCoIP Security Gateway Performance Limits

The following statistics represent the performance limits of the PCoIP Connection Manager and PCoIP Security Gateway with a *minimum* system configuration. You can exceed these limits, unless indicated, with more powerful systems.

PCoIP Connection Manager Limits

Session Establishment Limits

Based on the minimum connection manager system requirements, the PCoIP Connection Manager can establish the following number of sessions:

- 40 simultaneous *in-process* session establishment sequences
- Up to 400 simultaneous client communications

PCoIP Security Gateway Limits

Session Limits

Each PCoIP Security Gateway supports a maximum of **5,000** simultaneous sessions. You can lower this limit by [changing the `MaxConnections` setting](#) in `/opt/teradici/pcoipcm_data/data/SecurityGateway.conf`. If you need to support more than 5,000 simultaneous sessions, deploy additional PCoIP Connection Manager and PCoIP Security Gateways behind a load balancer.

Bandwidth Limits

When using the PCoIP Connection Manager and Security Gateway 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 for all combined concurrent sessions:

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

Estimated Bandwidth

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

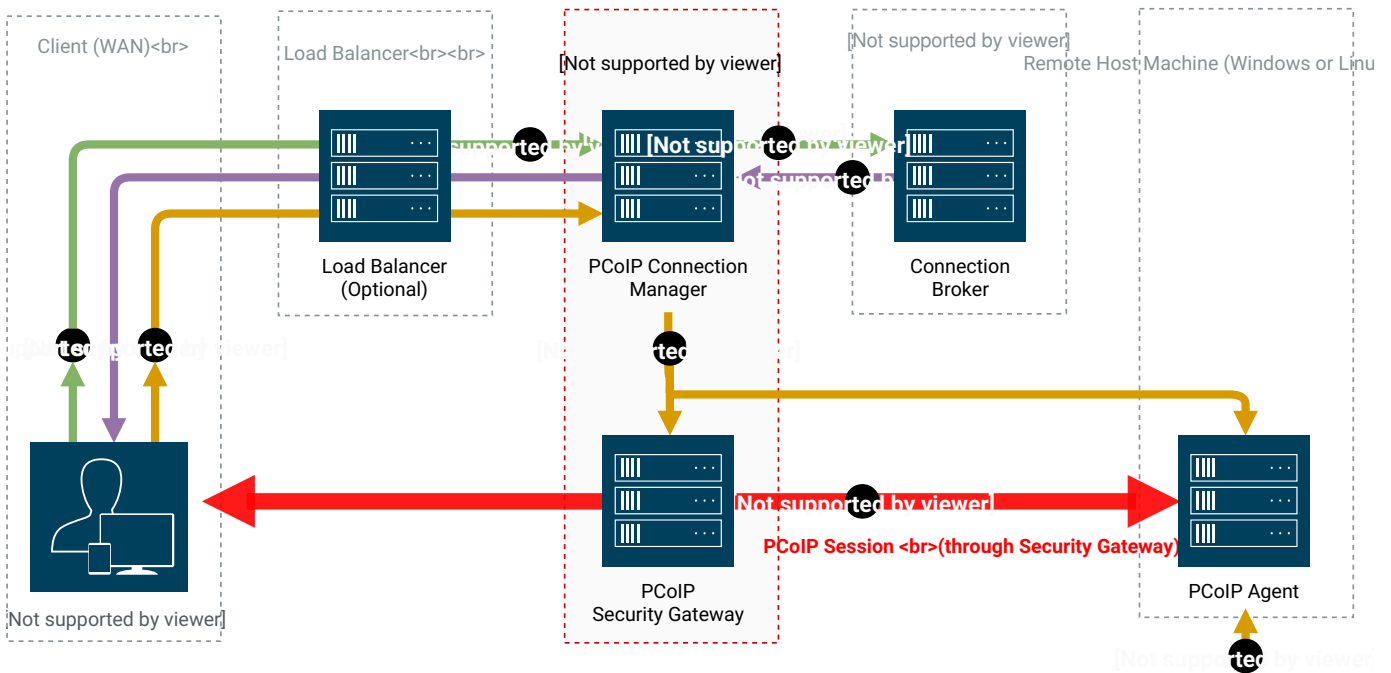
1100 Mbit/s is approximately the maximum bandwidth that can be achieved. Additional gains may be possible with larger sizing.

System Planning

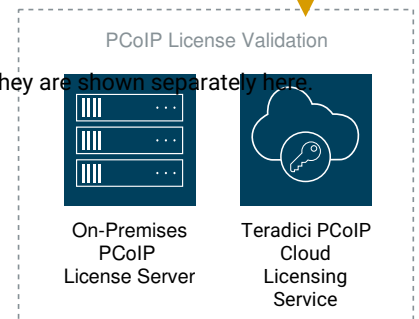
Before deploying the PCoIP Connection Manager and PCoIP Security Gateway, ensure you understand the PCoIP session establishment process and how [load balancers](#) and [firewalls](#) fit in.

Session Establishment

Here's the sequence of events involved in establishing a PCoIP session in a typical brokered scenario. In this example, the PCoIP client is outside the firewall, so the PCoIP Security Gateway is enabled to secure the connection and to proxy authorized traffic.



Security Gateway are distinct services installed as a pair on one machine. For visual clarity, they are shown separately here.



1. A user provides a server name and address to their PCoIP client, which passes the data to the **PCoIP Connection Manager** (this can be relayed through a load balancer, as shown here).

2. The *Connection Manager* communicates with the **Connection Broker** to authenticate the user and to obtain the list of desktops the user is entitled to use.
3. The *Connection Broker* passes the list of desktops back to the the **PCoIP Client**.
4. The user selects a desktop from the client UI, and their choice is passed back to the **PCoIP Connection Manager**.
5. The *PCoIP Connection Manager* prepares the **PCoIP Security Gateway** and the requested desktop's **PCoIP Agent**.
6. The *PCoIP Agent* acquires a session license from a licensing service (either the **PCoIP Cloud Licensing Service** or the a local **PCoIP License Server**).
7. The PCoIP session is established. The **PCoIP Client** now communicates directly with the selected desktop using the PCoIP Protocol.

 **Note: PCoIP Security Gateway in LAN systems**

The PCoIP Security Gateway secures PCoIP communications through the firewall. In systems where PCoIP clients are on the WAN, PCoIP traffic is relayed through the PCoIP Security Gateway. When the entire PCoIP system is on your company LAN, the PCoIP Security Gateway is unnecessary and the PCoIP Client and PCoIP agent communicate directly.

Load Balancing

You can use load balancers in front of multiple connection managers and security gateways to distribute system load to optimize performance. The load balancer must support the following:

- HTTPS
- Sticky sessions by the jsessionid

During session establishment, the PCoIP Connection Manager retrieves the public IP addresses of the PCoIP Security Gateways and passes them to the client. After the session is established, the client uses a provided IP address to communicate directly with a PCoIP Security Gateway.

Important: The PCoIP Security Gateway's public IP address must be set during installation

When a PCoIP Security Gateway is installed using the `--enable-security-gateway` flag, its public IP address is set using the `--external-pcoip-ip` flag during installation.

If the public IP address is configured to point to the *load balancer* instead of the *PCoIP Security Gateway*, the load balancer may direct the client to a PCoIP Security Gateway on the wrong server. If this happens, the client will not be able to establish a session.

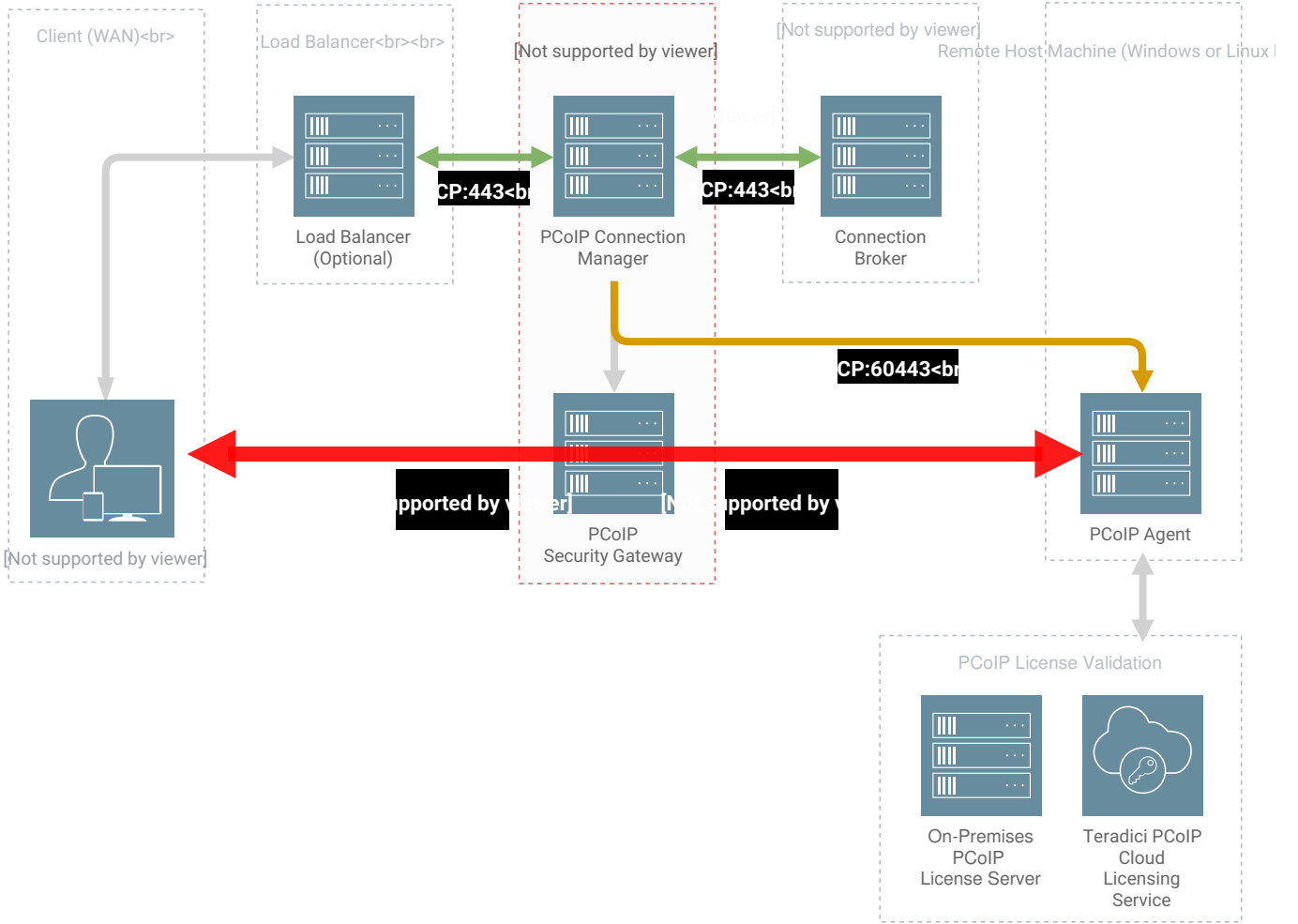
Public IP Address

The machine(s) with a PCoIP Connection Manager and/or a Security Gateway on it must have a public IP address if it is directly accessed from WAN.

To see how load balancers fit into firewall configurations, refer to [Configuring Firewalls](#).

Configuring Firewalls

If there is a firewall on the PCoIP Connection Manager server, ensure ports for PCoIP traffic are open so that users can access their desktop. The illustration shown next shows the default port numbers.



Firewall recommendations for establishing a PCoIP Session

Source	Port	Destination	Port	Description
PCoIP Client	*	PCoIP Connection Manager	TCP: 443	PCoIP broker protocol (HTTPS)
PCoIP Connection Manager	*	Connection broker	TCP: 443	PCoIP broker protocol (HTTPS)
PCoIP Connection Manager	*	PCoIP Agent	TCP: 60443	PCoIP agent protocol
PCoIP Client	*	PCoIP Security Gateway	UDP: 4172	PCoIP user data
PCoIP Client	*	PCoIP Security Gateway	TCP: 4172	PCoIP control information
PCoIP Security Gateway	*	PCoIP Agent	TCP: 4172	PCoIP control information
PCoIP Security Gateway	UDP: 55000	PCoIP Agent	UDP: 4172	PCoIP user data. <i>When deploying a desktop with a PCoIP agent, only port 4172 needs to be open.</i>

Inbound Connections

Ensure these ports are open for inbound connections:

Port	Purpose
443 TCP	Used by clients to connect to the PCoIP Connection Manager
4172 TCP/UDP	Used by authorized clients to connect to the PCoIP Security Gateway

Instructions for opening these ports are included in the [installation procedures](#).

Note that RHEL 8 and Rocky Linux 8 permit all outbound traffic by default.

 Important: Other required services may need open outbound ports

If the PCoIP Connection Manager is on a network behind a firewall that blocks outbound connections, ensure that the required ports for other required operating system services are open. We recommend that DHCP, DNS, and NTP are active for PCoIP Connection Manager operation.

Configuring Docker Network

The default docker network environment for the PCoIP Connection Manager and the PCoIP Security Gateway is assigned to `10.101.0.0/24`.

If your company network CIDR overlaps `10.101.0.0/24`, please use option `--docker-network-cidr` to provide a new network CIDR for docker during installation / updating. Addresses from any of the following CIDR classes can be used:

```
Class A: 10.0.0.0 to 10.255.255.255.  
Class B: 172.16.0.0 to 172.31.255.255.  
Class C: 192.168.0.0 to 192.168.255.255.
```

for example: `pcoip-cmsg-setup install --docker-network-cidr 172.16.0.0/24`

Installing

Installing for Online Environments

The following sections outline how to install the Modern Connection Manager and Security Gateway 24.03.

Before You Begin

Before you proceed with installation, note the following:

- **Docker must be installed** before you begin. For instructions, see [About Docker](#).
- Make sure ports TCP:80, TCP:443, TCP:4172, and UDP:4172 are open:

```
firewall-cmd --add-port 80/tcp
firewall-cmd --add-port 443/tcp
firewall-cmd --add-port 4172/tcp
firewall-cmd --add-port 4172/udp
```

- If you will be using IPv6, set up the required port forwarding rules:

```
# Add port forwarding rules
firewall-cmd --add-forward-port=port=443:proto=tcp:toport=8443
firewall-cmd --add-forward-port=port=80:proto=tcp:toport=8080
firewall-cmd --add-rich-rule='rule family=ipv6 forward-port protocol=tcp
port=443 to-port=8443'
firewall-cmd --add-rich-rule='rule family=ipv6 forward-port protocol=tcp
port=80 to-port=8080'

# Make the new settings persistent
firewall-cmd --runtime-to-permanent
```

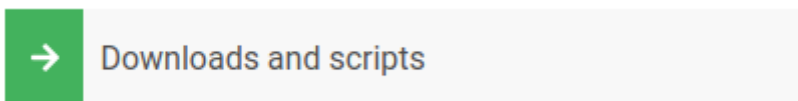
- If your environment has `podman` or `buildah` installed, uninstall them before proceeding.

```
sudo dnf erase podman buildah -y
```

Install PCoIP Modern Connection Manager and PCoIP Security Gateway

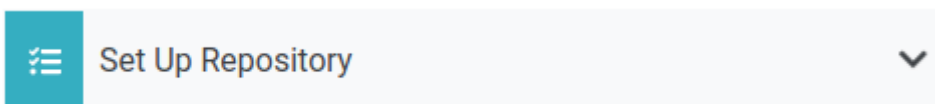
1. On the machine that hosts the PCoIP Connection Manager and/or the PCoIP Security Gateway, open a browser and go to the PCoIP Connection Manager and PCoIP Security Gateway [download page](#).

2. Click **Downloads and scripts**:



If you see a login button instead, click it to log into the site and then proceed.

3. Accept the End User License Agreement, then click **Set Up Repository**:



The window expands and show the setup scripts for each supported operating system. Copy the command for your system to the clipboard.

4. Open a console window and paste in the command you copied in the previous step. You may need to press to execute it.

The command fetches a configuration script from our servers and runs it locally, setting up and configuring the repository on the local machine.

5. Install the PCoIP Connection Manager and PCoIP Security Gateway package:

```
sudo dnf install pcoip-cmsg-setup
```

6. After the package is installed locally, run the `pcoip-cmsg-setup install` command with the required flags to complete installation.

```
sudo pcoip-cmsg-setup install <installation_flags>
```


Important: Required installation flags

There are a number of options and settings available. You can invoke the `install` command with the `--help` flag to list them:

```
pcoip-cmsg-setup install --help
```

They are also listed in the [next section](#).

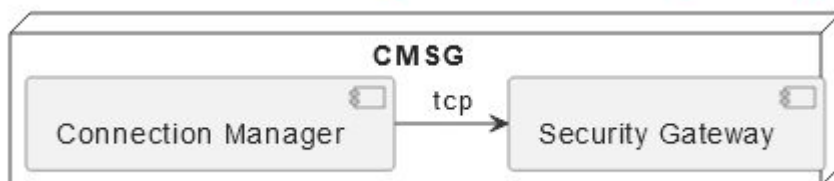
The `install` command prompts you for required parameters that have not been supplied via flags.

Installing Components Individually

- To install only the PCoIP Connection Manager use `--enable-security-gateway=false`.
- To install only the PCoIP Security Gateway use `--enable-connection-manager=false`.
- Otherwise both the PCoIP Connection Manager and PCoIP Security Gateway are installed by default.

Deployment Scenarios

- PCoIP Connection Manager and PCoIP Security Gateway deployed together: This is the default option when installing. There is no gateway failover in this deployment.



- PCoIP Connection Manager and PCoIP Security Gateways deployed separately: There is gateway failover in this scenario.



- PCoIP Connection Manager and PCoIP Security Gateways deployed together and separately: There is gateway failover in this scenario.



Installation Flags and Options

The following flags can be used to provide values at the command line. Flags that are required are identified in the description.

Boolean values should be provided as either `true` or `false`, lowercased, as in this example:

```
--example-flag=true
```

Flag	Type	Description
<code>--accept-policies</code>	Boolean	Automatically accepts the EULA and Privacy Policy. Required.
<code>--broker-url</code>	String	The URL of the PCoIP Broker, specified either as a <code>https://:</code> or <code>https://[:]</code> . Required.
<code>--ca-cert</code>	String	The full path and filename of the custom Certificate Authority's public certificate to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--compose-file</code>	String	Specify the full path to a local docker-compose file.
<code>--darksite-bundle-path</code>	string	The path of darksite install bundle to be used for darksite installation
<code>--docker-password</code>	String	Password to login to private registry.
<code>--docker-registry</code>	String	Specifies the HP source for Anyware Connector images to be install from. Debugging only: This is intended to be used for debugging purposes and should not be used without guidance from HP support. Using this flag incorrectly can result in failed installations.
<code>--docker-username</code>	String	Username to login to private registry.
<code>--enable-collaboration</code>	Boolean	Allow multiple PCoIP clients to collaborate on a PCoIP agent. (Default=true)
<code>--enable-ipv6</code>	Boolean	Enables IPv6 connections (Default=false). To enable IPv6 use <code>--enable-ipv6=true</code> . To disable IPv6 use <code>--enable-ipv6=false</code> , or omit this flag.
<code>--external-pcoip-ip</code>	StringArray	Sets the public IP address of Security Gateway. If <code>--enable-ipv6</code> is true, this option may be used twice (once for IPv4 and once for IPv6).

Flag	Type	Description
		Required if PCoIP Security Gateway is enabled
<code>--enable-security-gateway</code>	Boolean	Enable and use the PCoIP Security Gateway (Default=true).
<code>--help</code>		Lists all available flags.
<code>--host-address</code>	stringArray	Sets the host FQDN/IP address. The option may be used twice (once for the IP address and once for the FQDN)
<code>--ignore-disk-req</code>	Boolean	Ignore the check for the minimum disk space requirement.
<code>--license-server-url</code>	String	The address of the locally installed PCoIP License Server. Example: <code>https://<license-server-address>:<port></code>
<code>--self-signed</code>	Boolean	Automatically generate self-signed SSL cert and key for testing purposes. If specified, <code>--ssl-key</code> and <code>--ssl-cert</code> options are ignored.
<code>--ssl-cert</code>	String	The full path and filename of the SSL certificate to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--ssl-key</code>	String	The full path and filename of the SSL key to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--docker-network-cidr</code>		Sets CIDR for Connection Manager's docker network for services. If default docker network IP range is conflict with intranet, this option should be used to solve the confliction
<code>--debug</code>	String	

Flag	Type	Description
		Sets the log verbosity higher to help with debugging installation issues.
<code>--enable-connection-manager</code>	Boolean	Enable and use the PCoIP Connection Manager (Default=true).
<code>--external-sg-ip</code>	StringArray	Sets public IP addresses of external Security Gateways to enable gateway failover if a Security Gateway becomes unavailable. IP address should be provided in the format <code>--external-sg-ip=ipAddr1 --external-sg-ip=ipAddr2...</code>
<code>--jwt-verifying-cert</code>	String	The full path and filename of the certificate that the Security Gateway should use to validate the JWT token.
<code>--jwt-signing-key</code>	String	The full path and filename of the key to sign a JWT. It is used by the Connection Manager for signing the JWT token.

Federated Authentication Flags

Flag	Type	Description
<code>--enable-oauth</code>	Boolean	Enables Oauth authentication. (Default=false)
<code>--id-provider-url</code>	String	Sets the identity provider URL. Example: <code>--id-provider-url https://provider-1234567890.id.provider.com</code> . This flag is required if <code>--enable-oauth</code> is true.
<code>--oauth-client-id</code>	String	Gets the Client ID from the Identity Provider. This flag is also required if <code>--enable-oauth</code> is "true".

Federated Authentication Single Sign-On Flags

Flag	Type	Description
<code>--fa-url</code>	String	Override the fhe Federated Auth Broker URL provided to the PCoIP Agent. This flag can be used if auto-detection is not correcting determining the connector address. for example https://cac-vm-fqdn:port
<code>--enable-sso</code>	Boolean	Enables SSO. (Default=False)
<code>--sso-signing-csr-ca</code>	String	Path to copy intermediate CA Certificate.
<code>--sso-signing-csr-key</code>	String	Path to the intermediate key.
<code>--sso-signing-crl</code>	String	Path to a certificate revocation list.
<code>--sso-enrollment-url</code>	String	Gets the URL to the Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-domain</code>	String	Domain of the user to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-username</code>	String	Username for accessing Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-password</code>	String	Password for the username to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-certificate-template-name</code>	String	Name of the certificate template that Active Directory Certification Authority Web Enrollment Service uses to sign CSR.

About Docker

The PCoIP Connection Manager and PCoIP Security Gateway depends on Docker 20.10.0 or higher, which must be installed on the machine before you install the PCoIP Connection Manager and PCoIP Security Gateway.

If you have not installed Docker yet, [install it now](#).

If you are not sure if Docker is installed, or are not sure what Docker version you have, [verify your Docker version](#) first.

Verifying Docker Version

To verify your Docker installation and version:

1. SSH into the machine.
2. Open a console window and run the following command:

```
sudo docker -v
```

- If Docker is *not* installed, this command will produce an error. Installation instructions are provided in the [next section](#).
- If you see a version number that is *lower* than 20.10.0, you must uninstall Docker and then reinstall the supported version. Instructions for [uninstalling](#) and [installing](#) are provided in the next section.
- If you see a version number that is equal to or higher than 20.10.0, you have a compatible version of Docker already installed and can skip to PCoIP Connection Manager and PCoIP Security Gateway installation.

Uninstalling Docker

You'll only need to do this if you have an unsupported version of Docker already on the machine. If you haven't installed Docker yet, skip this section.

To uninstall Docker:

1. SSH into the machine.
2. Open a console window and run the following command:

```
sudo dnf remove docker docker-client docker-client-latest docker-common  
docker-latest docker-latest-logrotate docker-logrotate docker-engine  
docker-ce docker-ce-cli containerd.io runc
```

3. When uninstalling is complete, proceed to [Installing Docker](#).

Installing Docker

To install Docker:

If you do not have Docker installed, or if the Docker version is too low, install it using the following procedure:

1. SSH into the machine that hosts the PCoIP Connection Manager and/or PCoIP Security Gateway.
2. Open a console window, and run the following command. This removes the `podman` and `buildah` packages if they are installed (these packages conflict with Docker):

```
sudo dnf remove podman buildah
```

3. Run the following commands in the same console window. Note that if you copy and paste these commands into the console, you may need to press `Enter` again to execute the last command:

```
sudo dnf install -y dnf-utils  
sudo dnf config-manager --add-repo https://download.docker.com/linux/  
centos/docker-ce.repo  
sudo dnf install docker-ce docker-ce-cli containerd.io
```

4. Confirm installation:

```
sudo docker -v
```

Installing for Offline Environments

If the PCoIP Connection Manager and PCoIP Security Gateway machine does not have a connection to the public internet, you must create a temporary internet-connected machine to download a pre-created offline installation bundle and then transfer the bundle to the production machine.

For information on bundle dependencies, see [System Requirements](#).

Before You Begin

Before you proceed with installation, note the following:

- If your connection broker is configured to identify resources by host name, then DNS must be available and configured as follows:
 - Host names must be resolvable from the PCoIP Connection Manager server.
 - Host names must be resolvable from the PCoIP broker.

Downloading Offline Installation Bundle

You'll need a temporary machine with internet access.

1. On the temporary machine, open a browser and go to the PCoIP Connection Manager and PCoIP Security Gateway [download page](#), and download the installation bundle.
2. Transfer the installation bundle to the production machine using any acceptable method, such as a USB flash drive or SCP.

Note: Create Offline Bundle

If you preferred to create your own offline bundle for specific reasons, you can follow [bundle creation](#). However, we recommend using the pre-created offline installation bundle.

Installing PCoIP Connection Manager and the PCoIP Security Gateway

To install the PCoIP Connection Manager and the PCoIP Security Gateway:

1. SSH into the production machine.
2. Navigate to the directory where you placed the installer bundle.
3. Extract the bundle and move into the newly-created `teradici-pcoip-cmsg-bundle` directory:

```
tar xzvf pcoip-cmsg-setup_darksite-<version>.el8.tar.gz
```

```
cd teradici-pcoip-cmsg-bundle
```

4. Run the `pcoip-cmsg-setup-offline.sh` script to complete the installation

- **To install dependencies and follow the setup prompts to setup PCoIP Connection Manager and the PCoIP Security Gateway:**

```
sudo ./pcoip-cmsg-setup-offline.sh
```

and skip the next step.

- **To install dependencies and run `pcoip-cmsg-setup` later to setup PCoIP Connection Manager and the PCoIP Security Gateway:**

```
sudo ./pcoip-cmsg-setup-offline.sh -d
```

5. Move back up one directory level and then install the PCoIP Connection Manager and PCoIP Security Gateway:

```
cd ..  
sudo pcoip-cmsg-setup install --darksite-bundle-path teradici-pcoip-cmsg-bundle <installation_flags>
```

Important: Required installation flags

There are a number of options and settings available. You can invoke the `install` command with the `--help` flag to list them:

```
pcoip-cmsg-setup install --help
```

They are also listed in the [next section](#).

The `install` command will prompt you for required parameters that have not been supplied via flags.

Installation Flags and Options

The following flags can be used to provide values at the command line. Flags that are required are identified in the description.

Boolean values should be provided as either `true` or `false`, lowercased, as in this example:

```
--example-flag=true
```

Flag	Type	Description
<code>--accept-policies</code>	Boolean	Automatically accepts the EULA and Privacy Policy. Required.
<code>--broker-url</code>	String	The URL of the PCoIP Broker, specified either as a <code>https://:</code> or <code>https://[:]</code> . Required.
<code>--ca-cert</code>	String	The full path and filename of the custom Certificate Authority's public certificate to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--compose-file</code>	String	Specify the full path to a local docker-compose file.
<code>--darksite-bundle-path</code>	string	The path of darksite install bundle to be used for darksite installation
<code>--docker-password</code>	String	Password to login to private registry.
<code>--docker-registry</code>	String	Specifies the HP source for Anyware Connector images to be install from. Debugging only: This is intended to be used for debugging purposes and should not be used without guidance from HP support. Using this flag incorrectly can result in failed installations.
<code>--docker-username</code>	String	Username to login to private registry.
<code>--enable-collaboration</code>	Boolean	Allow multiple PCoIP clients to collaborate on a PCoIP agent. (Default=true)
<code>--enable-ipv6</code>	Boolean	Enables IPv6 connections (Default=false). To enable IPv6 use <code>--enable-ipv6=true</code> . To disable IPv6 use <code>--enable-ipv6=false</code> , or omit this flag.
<code>--external-pcoip-ip</code>	StringArray	Sets the public IP address of Security Gateway. If <code>--enable-ipv6</code> is true, this option may be used twice (once for IPv4 and once for IPv6).

Flag	Type	Description
		Required if PCoIP Security Gateway is enabled
<code>--enable-security-gateway</code>	Boolean	Enable and use the PCoIP Security Gateway (Default=true).
<code>--help</code>		Lists all available flags.
<code>--host-address</code>	stringArray	Sets the host FQDN/IP address. The option may be used twice (once for the IP address and once for the FQDN)
<code>--ignore-disk-req</code>	Boolean	Ignore the check for the minimum disk space requirement.
<code>--license-server-url</code>	String	The address of the locally installed PCoIP License Server. Example: <code>https://<license-server-address>:<port></code>
<code>--self-signed</code>	Boolean	Automatically generate self-signed SSL cert and key for testing purposes. If specified, <code>--ssl-key</code> and <code>--ssl-cert</code> options are ignored.
<code>--ssl-cert</code>	String	The full path and filename of the SSL certificate to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--ssl-key</code>	String	The full path and filename of the SSL key to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--docker-network-cidr</code>		Sets CIDR for Connection Manager's docker network for services. If default docker network IP range is conflict with intranet, this option should be used to solve the confliction
<code>--debug</code>	String	

Flag	Type	Description
		Sets the log verbosity higher to help with debugging installation issues.
<code>--enable-connection-manager</code>	Boolean	Enable and use the PCoIP Connection Manager (Default=true).
<code>--external-sg-ip</code>	StringArray	Sets public IP addresses of external Security Gateways to enable gateway failover if a Security Gateway becomes unavailable. IP address should be provided in the format <code>--external-sg-ip=ipAddr1 --external-sg-ip=ipAddr2...</code>
<code>--jwt-verifying-cert</code>	String	The full path and filename of the certificate that the Security Gateway should use to validate the JWT token.
<code>--jwt-signing-key</code>	String	The full path and filename of the key to sign a JWT. It is used by the Connection Manager for signing the JWT token.

Federated Authentication Flags

Flag	Type	Description
<code>--enable-oauth</code>	Boolean	Enables Oauth authentication. (Default=false)
<code>--id-provider-url</code>	String	Sets the identity provider URL. Example: <code>--id-provider-url https://provider-1234567890.id.provider.com</code> . This flag is required if <code>--enable-oauth</code> is true.
<code>--oauth-client-id</code>	String	Gets the Client ID from the Identity Provider. This flag is also required if <code>--enable-oauth</code> is "true".

Federated Authentication Single Sign-On Flags

Flag	Type	Description
<code>--fa-url</code>	String	Override the the Federated Auth Broker URL provided to the PCoIP Agent. This flag can be used if auto-detection is not correcting determining the connector address. for example https://cac-vm-fqdn:port
<code>--enable-sso</code>	Boolean	Enables SSO. (Default=False)
<code>--sso-signing-csr-ca</code>	String	Path to copy intermediate CA Certificate.
<code>--sso-signing-csr-key</code>	String	Path to the intermediate key.
<code>--sso-signing-crl</code>	String	Path to a certificate revocation list.
<code>--sso-enrollment-url</code>	String	Gets the URL to the Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-domain</code>	String	Domain of the user to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-username</code>	String	Username for accessing Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-password</code>	String	Password for the username to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-certificate-template-name</code>	String	Name of the certificate template that Active Directory Certification Authority Web Enrollment Service uses to sign CSR.

Enabling or Disabling the PCoIP Security Gateway

By default, the PCoIP Security Gateway is enabled when the bundle is installed. This configuration is highly recommended for deployments where users will connect over the WAN. If your users are behind a firewall and do not access their desktops from the WAN, you may not need the PCoIP Security Gateway.

If you are sure that you do not need the PCoIP Security Gateway, reinstall the bundle using the `--enable-security-gateway=false` flag.

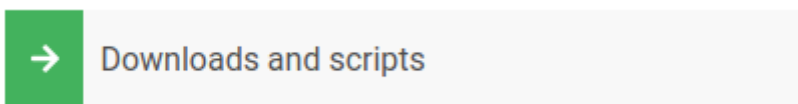
To reenble the PCoIP Security Gateway, reinstall the bundle using the default options.

Creating the Installation Bundle

First, you'll download the package and dependencies to a temporary internet-connected machine, create an installation bundle.

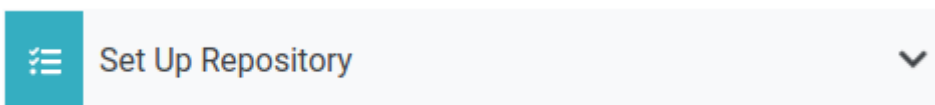
To create the offline installation bundle:

1. Install [Docker](#) onto the temporary machine.
2. On the temporary, open a browser and go to the PCoIP Connection Manager and PCoIP Security Gateway [download page](#).
3. Click **Downloads and scripts**:



If you see a login button instead, click it to log into the site and then proceed.

4. Accept the End User License Agreement, then click **Set Up Repository**:



The window will expand and show the setup scripts for each supported operating system. Copy the command for your system to the clipboard.

5. Open a console window and paste in the command you copied in the previous step. You may need to press to execute it.

The command fetches a configuration script from our servers and runs it locally, setting up and configuring the repository on the local machine.

6. Install **pcoip-cmsg-setup**

```
sudo dnf install pcoip-cmsg-setup
```

7. Find and note the rpm name for the setup package. We will use this name when creating the offline bundle next.

```
sudo dnf info pcoip-cmsg-setup
```

The rpm name will similar to this: `pcoip-cmsg-setup-<version>-<release>`.

8. Create the offline install bundle:

```
sudo pcoip-cmsg-setup create-darksite-bundle --pcoip-cmsg-rpm-path <rpm
name>
```

...where `<rpm name>` is the name you noted in the previous step.

The process will create a tarball called `teradici-pcoip-cmsg-bundle.tar.gz`.

Once this process has completed successfully, you can dispose of the temporary machine.

Updating the PCoIP Connection Manager and PCoIP Security Gateway

The PCoIP Connection Manager and the PCoIP Security Gateway components can be installed in an offline or online deployment, depending on your environment. The procedure for updating these components is different for both scenarios.

Note: Load Balancer

If you have a load balancer in front of a group of PCoIP Connection Manager and PCoIP Security Gateway virtual machines, you can configure the load balancer to stop sending new connections to the PCoIP Connection Manager and PCoIP Security Gateway while you are updating.

Updating an Offline Installation

If your deployment is offline (dark site), use this procedure.

1. [Download a new installation bundle](#) .
2. Transfer the installation bundle to the production machine using any acceptable method, such as a USB flash drive or SCP.
3. Extract the bundle and move into the newly-created teradici-pcoip-cmsg-bundle directory:

```
tar xzvf pcoip-cmsg-setup_darksite-<version>.el8.tar.gz
cd teradici-pcoip-cmsg-bundle/dependencies
```

4. Install the PCoIP Connection Manager

```
sudo dnf install --allowmissing pcoip-cmsg-setup*.rpm --disablerepo="*" -y
```

5. Move back up one directory level and then install the PCoIP Connection Manager and PCoIP Security Gateway:

```
cd ../../..  
sudo pcoip-cmsg-setup install --darksite-bundle-path teradici-pcoip-cmsg-  
bundle <installation_flags>
```

Updating an Online Installation

To upgrade a PCoIP Connection Manager and PCoIP Security Gateway that can reach the public internet:

Important: Installation flags are required

If installation flags are absent, or are different from the original installation, the configuration on the new machine will be different.

1. Update the package:

```
dnf upgrade pcoip-cmsg-setup -y
```

2. Reinstall the package:

```
pcoip-cmsg-setup install <installation_flags>
```

To downgrade to an earlier version:

1. Downgrade the package:

```
dnf downgrade pcoip-cmsg-setup -y
```

2. Reinstall the package:

```
pcoip-cmsg-setup install <installation_flags>
```

Uninstalling PCoIP Connection Manager and PCoIP Security Gateway

If you want to remove the PCoIP Connection Manager and PCoIP Security Gateway completely from the production machine, open a console and run the following commands:

1. Close out running Docker containers:

```
sudo docker stack rm pcoipcm
sudo docker swarm leave --force
```

2. Remove Docker images

```
sudo docker rmi -f $(sudo docker images --format "{{.ID}} {{.Repository}}"
| grep -E */pcoip-cm | awk '{ print $1 }')
sudo docker rmi -f $(sudo docker images --format "{{.ID}} {{.Repository}}"
| grep -E */sg | awk '{ print $1 }')
```

3. Remove the setup files and repository information:

```
sudo dnf remove pcoip-cmsg-setup
sudo rm -f /etc/yum.repos.d/teradici-pcoip-cmsg.repo
```

4. Clean up files and directories:

```
sudo rm -rf /opt/teradici
sudo rm -rf /var/log/Teradici
```

5. Optionally remove Docker, if it will no longer be needed:

```
sudo docker system prune -f -a # remove all unused images
sudo systemctl stop docker # stop Docker
sudo systemctl disable docker # Prevent Docker from running on reboot
sudo dnf remove docker-ce docker-ce-cli containerd.io # uninstall Docker
Engine
```

6. Optionally remove the Docker repository:

```
sudo rm -f /etc/yum.repos.d/docker-ce.repo
```

Configuring

Configuring the PCoIP Connection Manager and PCoIP Security Gateway

You can configure the PCoIP Connection Manager and/or the Security Gateway using the `pcoip-cmsg-setup configure` command.

The general syntax is:

```
sudo pcoip-cmsg-setup configure <flags>
```

For example, to specify a broker url, you would open a console window and enter the following:

```
sudo pcoip-cmsg-setup configure --broker-url https://<example>
```


Configuration Flags and Options

The following flags can be used to provide values at the command line.

Flag	Type	Description
<code>--broker-url</code>	String	The URL of the PCoIP Broker, specified either as a <code>https://:</code> or <code>https://:</code> or <code>https://[]:</code> . Required.
<code>--clear-host-address</code>	Boolean	Clears the host address.
<code>--ca-cert</code>	String	The full path and filename of the custom Certificate Authority's public certificate to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--clear-trusted-license</code>	Boolean	Clears trusted license certificate and key.
<code>--compose-file</code>	String	Specify the full path to a local docker-compose file.
<code>--docker-password</code>	String	Password to login to private registry.
<code>--docker-registry</code>	String	Specifies the HP source for Anyware Connector images to be install from. Debugging only: This is intended to be used for debugging purposes and should not be used without guidance from HP support. Using this flag incorrectly can result in failed installations.
<code>--docker-username</code>	String	Username to login to private registry.
<code>--enable-collaboration</code>	Boolean	Allow multiple PCoIP clients to collaborate on a PCoIP agent. (default true)
<code>--external-pcoip-ip</code>	StringArray	Sets the public IP addresses of VM which hosts Security Gateway. This option can be used twice, once for IPv4 and once for IPv6 (if using). Required if PCoIP Security Gateway is enabled.
<code>--help</code>		Display configuration help.
<code>--host-address</code>	stringArray	Sets the host FQDN/IP address. The option may be used twice (once for the IP address and once for the FQDN)
<code>--license-server-url</code>	String	The address of the locally installed PCoIP License Server. Example: <code>https://<license-server-address>:<port></code>
<code>--ssl-cert</code>	String	The full path and filename of the SSL certificate to be used in the PCoIP Connection Manager and PCoIP Security Gateway. Required if <code>--self-signed</code> is not used.
<code>--ssl-key</code>	String	The full path and filename of the SSL key to be used in the PCoIP Connection Manager and PCoIP Security

Flag	Type	Description
		Gateway. Required if <code>--self-signed</code> is not used.
<code>--trusted-license-cert</code>	String	Trusted Customer License certificate path. Defaults to /opt/teradici/pcoipcm_data/certs/tcl-cert.crt).
<code>--trusted-license-cert-key</code>	String	Trusted Customer License certificate key path. Defaults to /opt/teradici/pcoipcm_data/certs/tcl-cert.key.
<code>--docker-network-cidr</code>	String	Sets CIDR for Connection Manager's docker network for services.
<code>--enable-horizon</code>	Boolean	Enables/Disables HP Anyware to be brokered with VMware Horizon (Default=false).
<code>--external-sg-ip</code>	StringArray	Sets public IP addresses of external Security Gateways to enable gateway failover if a Security Gateway becomes unavailable. IP address should be provided in the format <code>--external-sg-ip=ipAddr1 --external-sg-ip=ipAddr2...</code>
<code>--jwt-verifying-cert</code>	String	The full path and filename of the certificate that the Security Gateway should use to validate the JWT token.
<code>--jwt-signing-key</code>	String	The full path and filename of the key to sign a JWT. It is used by the Connection Manager for signing the JWT token.

Federated Authentication Flags

Flag	Type	Description
<code>--enable-oauth</code>	Boolean	Enables Oauth authentication. (Default=False)
<code>--id-provider-url</code>	String	Sets the identity provider URL. Example: <code>--id-provider-url https://provider-1234567890.id.provider.com</code> . This flag is required if <code>--enable-oauth</code> is true.
<code>--oauth-client-id</code>	String	Gets the Client ID from the Identity Provider. This flag is also required if <code>--enable-oauth</code> is "true".

Federated Authentication Single Sign-On Flags

Flag	Type	Description
<code>--fa-url</code>	String	Override the fhe Federated Auth Broker URL provided to the PCoIP Agent. This flag can be used if auto-detection is not correcting determining the connector address. for example https://cac-vm-fqdn:port
<code>--enable-ss0</code>	Boolean	Enables SSO. (Default=False)
<code>--sso-signing-csr-ca</code>	String	Path to copy intermediate CA Certificate.
<code>--sso-signing-csr-key</code>	String	Path to the intermediate key.
<code>--sso-signing-crl</code>	String	Path to a certificate revocation list.
<code>--sso-enrollment-url</code>	String	Gets the URL to the Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-domain</code>	String	Domain of the user to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-username</code>	String	Username for accessing Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-password</code>	String	Password for the username to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-certificate-template-name</code>	String	Name of the certificate template that Active Directory Certification Authority Web Enrollment Service uses to sign CSR.

Security and Certificates

Security Considerations

All certificate files must be in base64-encoded PEM format.

Follow your organisation's security policy

For all security and certificate procedures, ensure you follow your organisation's security policy.

Creating, Installing, and Managing Certificates

In order to establish secure TLS connections with clients, certificates must be configured for the PCoIP Connection Manager and the PCoIP Security Gateway. If the required certificate files are not present or they are improperly configured, clients will not be able to connect and users will not be able to establish PCoIP sessions.

Only certificates with RSA private keys having at least 2,048-bit length are supported. RSA private keys having at least 3,072-bit length are recommended. Certificates with DSA private keys are not supported. Certificates that include an MD5-based digital signature algorithm are not supported.

Both the PCoIP Connection Manager and PCoIP Security Gateway support wildcard certificates which can be used on multiple PCoIP Connection Manager and PCoIP Security Gateway servers.

If you are ready to replace your default self-signed certificates with your own signed certificates, proceed to [Signed Certificates for Production](#).

Ensure all certificate files follow your security policy

Protect the regenerated certificate and ensure all certificate files you use conform to your organization's security policy.

Default Certificate

The PCoIP Connection Manager and PCoIP Security Gateway installation script generates a self-signed certificate during installation to facilitate testing. **This should be replaced with your own certificate, signed by a trusted Certificate Authority (CA), when deploying a production system.**

By default, both the PCoIP Connection Manager and the PCoIP Security Gateway use the same private key and signed certificate; if your security policy requires it, each service can use its own key/certificate pair instead. If two sets of certificates are required, follow these procedures twice to generate two key/certificate pairs and [configure the PCoIP Security Gateway](#) appropriately.

Copying certificates from a Window system to a Linux system

When copying certificates from a Windows system to a Linux system, line endings might be incorrect. Check that the certificate text is formatted correctly.

Signed Certificates for Production

Production systems should use your own certificates, signed by a trusted certificate authority (CA). The following sections describe the process of creating, signing, and installing certificates.

At a high level, the process is:

1. [Generate a new private key and certificate signing request](#).
2. [Submit the CSR to a trusted certificate authority \(CA\)](#) for signing, either internal or third-party.
3. [Verify and convert the resulting certificate files](#) to the .pem format.
4. [Install the certificates](#) on the PCoIP Connection Manager and Security Gateway machine, restart both services, and inspect their log files to verify that the certificates are working and that all services have started.
5. [Protect the certificate files and access](#).

Danger: These instructions are examples

The following procedures are working examples. Before following them, you should be sure they conform to your organization's security policies. Modify these procedures as necessary.

Open SSL Minimum Requirements

The following procedures use openssl to create and manage certificates. If you use another tool, adapt these instructions accordingly. The minimum Open SSL version on your virtual machine is 1.1.1.

CREATING CERTIFICATE FILES

First, generate a new private key and CSR (certificate signing request).

To generate a private key and CSR:

1. On the PCoIP Connection Manager server, open a command prompt.
2. Create a temporary directory to store the certificate and move into it.

This example uses `~/certs`, which creates a `certs` directory under your home directory, but you can create it anywhere you like:

```
mkdir ~/certs
cd ~/certs
```

3. Generate a private key file and CSR according to your organization's security policy.

This example creates an RSA 3072-bit private key and a CSR requesting a sha384 hash algorithm. The private key file is `private.key` and the CSR file is `server.csr`.

```
openssl req -new -newkey rsa:3072 -sha384 -nodes -keyout private.key -out
server.csr
```


When running this command, you will be prompted for information to be displayed in the certificate.

Distinguished Name Field	Description	Example
Country Name	The two-letter ISO abbreviation for your country	CA for Canada
State or Province Name	The unabbreviated name of the state or province where your organization is legally located.	British Columbia
Locality Name	The city where your organization is legally located.	Burnaby
Organization Name	The full legal name of your organization. Cannot use < > ~ ! @ # \$ % ^ * / \ () ? . , &	EasyShop Corporation
Organization Unit Name	Department of your organization. Cannot use < > ~ ! @ # \$ % ^ * / \ () ? . , &	Global Support Services
Common Name	The fully qualified domain name (FQDN) of your server. This must be an exact match or, in the case of a wild card, an asterisk (*) before the domain.	If your PCoIP Connection Manager address is easyshop.easybuzz.com then the CSR must have the common name easyshop.easybuzz.com. If you plan on having a wildcard certificate for use on multiple PCoIP Connection Manager servers, then prefix the domain with an asterisk (*). In this example: *.teradici.com.
Email Address	Leave blank	
A challenge password	Leave blank	
An optional company name	Leave blank	

You should now have two files in your ~/certs folder; private.key and server.csr.

You can verify the details of the CSR request using the following command:

```
openssl req -noout -text -in ~/certs/server.csr
```

OBTAINING THE SIGNED PUBLIC KEY CERTIFICATE

Next, use your CSR request to obtain a public signed certificate. Submit `server.csr` to a trusted CA following your organization's security policy. Follow the CA's instructions to obtain the public signed certificate.

If your CA offers the public signed certificate both with and without the certificate chain, download both. If they do not offer a certificate file including the certificate chain, refer to your CA's documentation on how to build it.

When you have received the signed files, copy them into your working directory (`~/certs`).

VERIFYING AND CONVERTING CERTIFICATE FILE FORMAT

Before installing your certificate, you must verify that it's in the correct format and convert it to .

These instructions assume the following:

- You have copied the files received from the CA to `~/certs`.
- The public certificate signed by the CA *without* the certificate chain is named `certificate.crt`.
- The public certificate signed by the CA *with* the certificate chain (intermediary and root certificates) is named `CACertificate.crt`.

To verify the certificate file format:

Verify the `certificate.crt` file:

```
openssl x509 -in certificate.crt -text -noout
```

- If you don't see any errors, change the file extension from `.crt` to `.pem`:

```
mv certificate.crt certificate.pem
```

- If you DO see errors, open the certificate file in a text editor and verify the following:
 - There are no extra characters at the end of lines
 - The file starts with `-----BEGIN CERTIFICATE-----`
 - The file ends with `-----END CERTIFICATE-----`

If the file doesn't begin and end with the required lines, it's in the wrong format. Convert it to PEM:

```
openssl x509 -inform der -in certificate.crt -out certificate.pem
```

Verify the newly renamed file:

```
openssl x509 -in certificate.pem -text -noout
```

Repeat these steps for `CAcertificate.crt` (the certificate that includes the certificate chain).

When you are done, you should have two `.pem` files and one private key file in the `~/certs` directory:

File	Explanation
<code>private.key</code>	Contains the certificate's private key.
<code>certificate.pem</code>	Contains a public certificate signed by a CA without the certificate chain. This is presented to PCoIP clients when they connect to the PCoIP Connection Manager during authentication and resource allocation.
<code>CAcertificate.pem</code>	Contains the certificate chain, including any intermediate and root certificate. Self-signed certificates do not have any root or intermediate certificate.

Important: Back up your certificate and private key

Back up the private key and certificate in a secure location according to your organization's security policy.

Federated Authentication

Federated Authentication using OAuth2

Federated Authentication Overview

Federated User Authentication enables organizations to use their own Identity Provider (IdP) as the source to verify the identity and to authenticate a user before permitting them to select a remote workstation. Once the desired workstation is selected, the user needs to provide the username and password to authenticate at the remote workstation.

Federated Authentication with Single Sign-On (SSO)

Federated Authentication is a feature that permits using the IdP to authenticate to the point of selecting your desktop from the list of workstations, and you need not to authenticate again to log in. If you are interested in this functionality, please discuss with your HP account representative.

PREREQUISITES

To use the Federated Authentication Functionality, you must meet the following criteria:

- CMSG 23.01 or later.
- HP PCoIP Client version 23.01.0 or later
- An Identity Provider that supports OAuth2
- A custom or third-party broker that supports Federated User Authentication using the PCoIP Broker Protocol

NEXT STEPS

In order to successfully configure Federated Authentication, you need to follow the steps below in order:

1. Configure a third-party IDP.
 - [Configure Okta](#)

- [Configure Azure Active Directory](#)

2. [Enable Federated Authentication](#)

Configuring Okta IDP

Okta IDP is third party identity provider, a service that manages user accounts. Adding IDPs in Okta enables your end users to self-register with your custom applications by authenticating the user who is trying to Login. To enable Federated Authentication, you need to configure Okta IDP.

⚠ Okta Documentation Reference

The configuration steps listed below are based on the most recent documentation release by Okta. These steps are subject to change provided there is a change or upgrade in the Okta documentation. For more information, see [Okta Documentation](#).

⚠ IDP Configuration Subject to Change

The configuration instructions below are provided as an example with Okta IDP. They are provided as-is. The method of configuration could change outside of the control of HP. Additionally, other IDPs could have different steps required and may use different terms to describe the requirements.

In most IdPs, the settings include terms like:

- Creating an App Integration
- OAuth2 or OIDC or OpenId Connect sign-in method
- Native Application application type
- The Grant type is Authorization Code
- And the redirect URL would be: pcoip//oauth/

After completing the setup within your IdP, you must have the following information for future configurations:

- The authorization URL of your identity provider
- A Client ID

TO CONFIGURE OKTA

1. Login to Okta on the link [here](#).
2. Go to **Applications** section on the left pane and select **Create App Integration**.

The screenshot shows the Okta administrator interface. On the left is a navigation menu with the Okta logo at the top. The menu items are: Dashboard (dropdown), Directory (dropdown), Applications (dropdown), Applications (selected), Security (dropdown), Reports (dropdown), and Settings (dropdown). The main content area is titled "Applications" and contains a search bar, two buttons: "Create App Integration" and "Browse App Catalog", and a table. The table has a search bar at the top and a "STATUS" header. The table content is as follows:

STATUS		
ACTIVE	4	
INACTIVE	1	

3. In the **Create a new app integration** window, select **OIDC-OpenID Connect** as the sign-in method and **Native Application** as the Application type.

Applications

[Create App Integration](#) [Browse App Catalog](#) [Assign Users to App](#) [More ▾](#)

Create a new app integration ✕

Sign-in method
[Learn More](#) [🔗](#)

- OIDC - OpenID Connect**
Token-based OAuth 2.0 authentication for Single Sign-On (SSO) through API endpoints. Recommended if you intend to build a custom app integration with the Okta Sign-In Widget.
- SAML 2.0**
XML-based open standard for SSO. Use if the Identity Provider for your application only supports SAML.
- SWA - Secure Web Authentication**
Okta-specific SSO method. Use if your application doesn't support OIDC or SAML.
- API Services**
Interact with Okta APIs using the scoped OAuth 2.0 access tokens for machine-to-machine authentication.

Application type
What kind of application are you trying to integrate with Okta?

Specifying an application type customizes your experience and provides the best configuration, SDK, and sample recommendations.

- Web Application**
Server-side applications where authentication and tokens are handled on the server (for example, Go, Java, ASP.Net, Node.js, PHP)
- Single-Page Application**
Single-page web applications that run in the browser where the client receives tokens (for example, Javascript, Angular, React, Vue)
- Native Application**
Desktop or mobile applications that run natively on a device and redirect users to a non-HTTP callback (for example, iOS, Android, React Native)

[Cancel](#) [Next](#)


4. Click **Next**.


5. In the **New Native App Integration** window, enter a name in the **App integration name** field.

New Native App Integration

General Settings

App integration name

Logo (Optional) 


Grant type [Learn More](#) 

Client acting on behalf of a user

- Authorization Code
- Interaction Code
- Refresh Token
- Resource Owner Password
- SAML 2.0 Assertion
- Device Authorization
- Token Exchange
- Implicit (hybrid)


Sign-in redirect URIs Allow wildcard * in sign-in URI redirect.

Okta sends the authentication response and ID token for the user's sign-in request to these URIs

[Learn More](#) 

Sign-out redirect URIs (Optional)

After your application contacts Okta to close the user session, Okta redirects the user to one of these URIs.

[Learn More](#) 

Assignments

Controlled access

Select whether to assign the app integration to everyone in your org, only selected group(s), or skip assignment until after app creation.

- Allow everyone in your organization to access
- Limit access to selected groups
- Skip group assignment for now

6. Check the **Authorization Code** option as Grant type.

7. Enter `pcoip://oauth/` as the **Sign-in redirect URIs**.

8. In the **Assignments** section, select the **Skip group assignment for now** option.
9. Click **Save**.

Okta IDP is now Configured.

 **HP Anyware supports other IDPs**

You can use other identity providers that support OAuth2 other than Okta, however, they have not been validated by HP and they may or may not work, or instructions and terms may vary.

Configuring Azure Active Directory

Azure Active Directory is a third-party identity provider (IdP) that can be configured to work with Anyware Manager. This permits Azure to be used as the source of authentication for any user attempting to connect to a connector in order to get a list of remote workstations to connect to.

In most IdPs, the settings include terms like:

- Creating an App Integration
- OAuth2 or OIDC or OpenId Connect sign-in method
- Native Application application type
- The Grant type is Authorization Code
- And the redirect URL would be: pcoip//oauth/

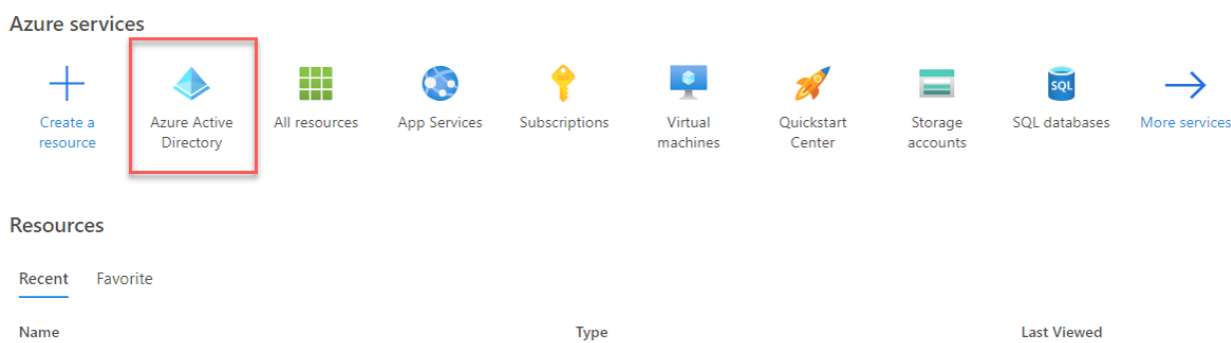
⚠ IDP Configuration Subject to Change

The configuration instructions below are provided as an example with Azure Active Directory IDP. They are provided as-is. The method of configuration could change outside of the control of HP. Additionally, other IdPs could have different steps required and may use different terms to describe the requirements.

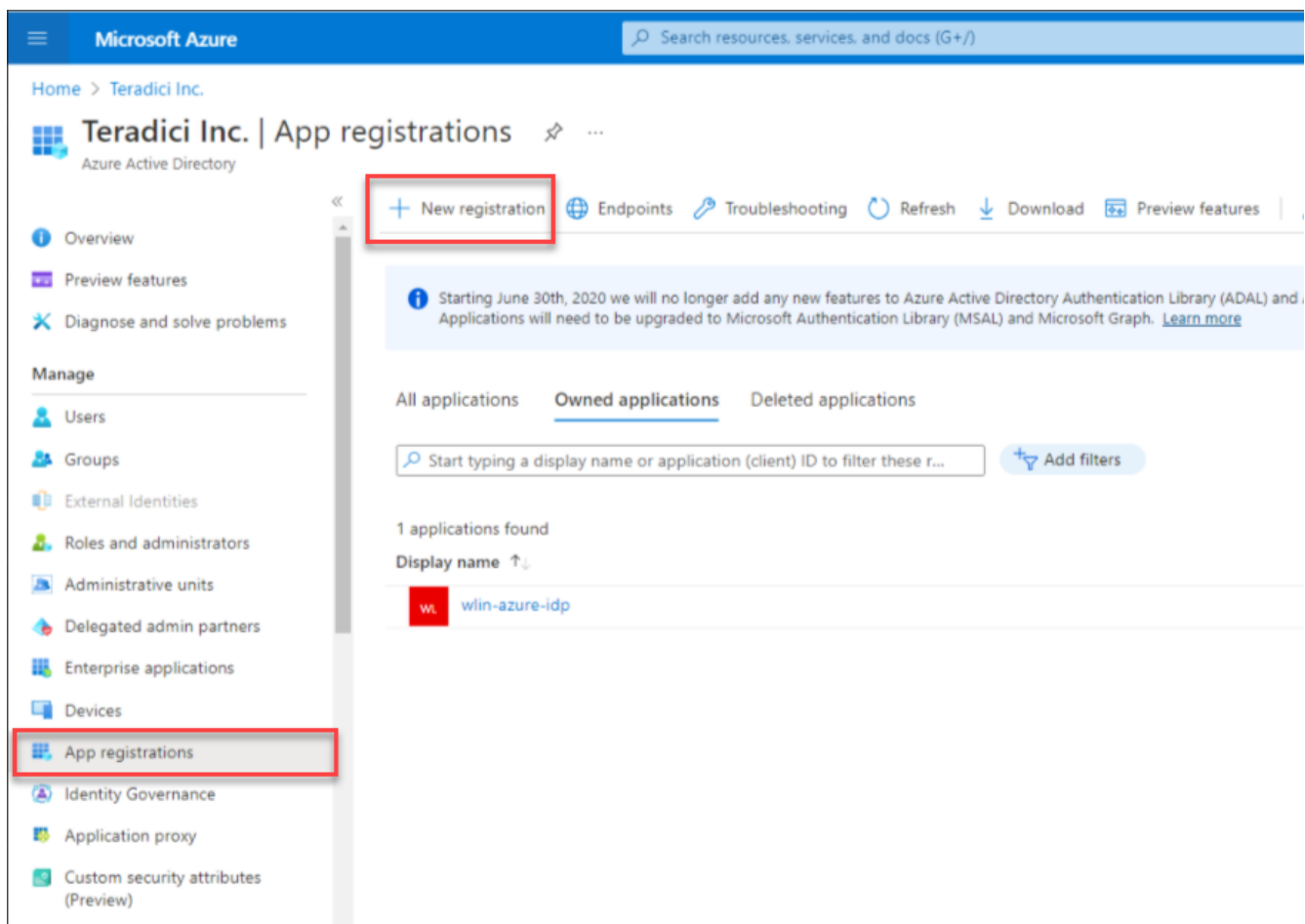
CONFIGURE AZURE ACTIVE DIRECTORY

To configure:

1. Login to Microsoft Azure and Select the **Azure Active Directory** component.



2. From the left pane, select **App registrations** and click **New registration**.



3. Enter the application name, supported account types, and the redirect URL (optional).
4. Click **Register**.

Microsoft Azure Search resources, services, and docs (G+)

Home > Teradici Inc. | App registrations >

Register an application

*** Name**
The user-facing display name for this application (this can be changed later).
my-aad-idp ✓

Supported account types
Who can use this application or access this API?
 Accounts in this organizational directory only (Teradici Inc. only - Single tenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
 Personal Microsoft accounts only
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.
 Public client/native (mobile ... | pcoip://oauth/ ✓

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

5. In the **App registrations** page, Click **Endpoints** and make a note of the client ID and the IDP URL for future configurations.

Search resources, services, and docs (G+)

Delete Endpoints Preview features

Essentials
 Display name :
 Application (client) ID : 3307
 Object ID : 3f8c...3f8c
 Directory (tenant) ID : 2cf18
 Supported account types : My organization only

Endpoints

- OAuth 2.0 authorization endpoint (v2)
https://login.microsoftonline.com/.../oauth2/v2.0/authorize
- OAuth 2.0 token endpoint (v2)
https://login.microsoftonline.com/.../oauth2/v2.0/token
- OAuth 2.0 authorization endpoint (v1)
https://login.microsoftonline.com/.../oauth2/authorize
- OAuth 2.0 token endpoint (v1)
https://login.microsoftonline.com/.../oauth2/token
- OpenID Connect metadata document
https://login.microsoftonline.com/715c7...2cf18/v2.0/.well-known/openid-configuration

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Micro...

Enable Federated Authentication for CMSG

IDP Configuration

Configuration parameters below are all obtained during the configuration of the IDP. Before configuring Federated Authentication, please ensure that you have an active third-party identity provider configured correctly.

If you are installing a new CMSG:

- Run the command: `sudo pcoip-cmsg-setup install [...other settings...] --enable-oauth true --id-provider-url https://id.provider.com --oauth-client-id XXXXXXXXXX`

If you are configuring an existing CMSG:

- Run the command: `sudo pcoip-cmsg-setup configure [...other settings...] --enable-oauth true --id-provider-url https://id.provider.com --oauth-client-id XXXXXXXXXX`

For more information on installing or updating CMSG, see [Installing for Online Environments](#) and [Updating CMSG](#).

INSTALLATION FLAGS

Flag	Type	Description
<code>--enable-oauth</code>	Boolean	Enables Oauth authentication. (Default=False)
<code>--id-provider-url</code>	String	Sets the identity provider URL. Example: <code>--id-provider-url https://provider-1234567890.id.provider.com</code> . This flag is required if <code>--enable-oauth</code> is true.
<code>--oauth-client-id</code>	String	Gets the Client ID from the Identity Provider. This flag is also required if <code>--enable-oauth</code> is "true".

Single Sign-On

Single Sign-On Overview

Federated User Authentication enables organizations to use their own Identity Provider (IdP) as the source to verify the identity and to authenticate a user before permitting them to select remote workstation. By Configuring Single-Sign-On, the user does not need to authenticate and directly connects to the remote workstation once the desired workstation is selected.

Federated Authentication with Single Sign-On (SSO)

Single Sign-On is a feature that permits using the IdP to authenticate to the point of selecting your desktop from the list of workstations, and you need not to authenticate again to log in.

PREREQUISITES

To use the Federated Authentication Functionality, you must meet the following criteria:

- CMSG 23.04 or later.
- HP PCoIP Client version 23.01.0 or later
- HP PCoIP Windows Agent 23.01.0 or later (SSO is not supported on Linux or MacOS)
- An Identity Provider that supports OAuth2
- A custom or third-party broker that supports Federated User Authentication using the PCoIP Broker Protocol

NEXT STEPS

In order to successfully configure Single Sign-On, you need to follow the steps below in order:

1. Configure a third-party IDP.
 - [Configure Okta](#)
 - [Configure Azure Active Directory](#)
2. [Prepare for SSO](#)
3. [Enable Federated Authentication with SSO](#)

Configuring IDP for Single Sign-On

Before you start preparing for Single Sign-On, ensure that you configure an IDP to enable Federated Authentication.

- For more information on Okta IDP configuration, see [Configuring Okta IDP](#).
- For more information on Azure Active Directory configuration, see [Configuring Azure Active Directory](#).

Preparing for Single Sign-On

Configuring Single Sign-On (SSO) prevents you from authenticating twice, when you need to access the workstation and when you want to make a connection to the remote workstation.

⚠ Certificate Authority Instructions

The instructions assume you have a Certification Authority (CA) in your environment and your remote workstations use it to verify certificates. If you do not have a Certification Authority, See <https://learn.microsoft.com/en-us/windows-server/networking/core-network-guide/cncg/server-certs/install-the-certification-authority>. Instructions for generating a signed intermediate certificate and private key can vary from CA to CA, or even between versions of the same CA. Please reference your CA documentation for further instructions.

ENROLLMENT OPTIONS

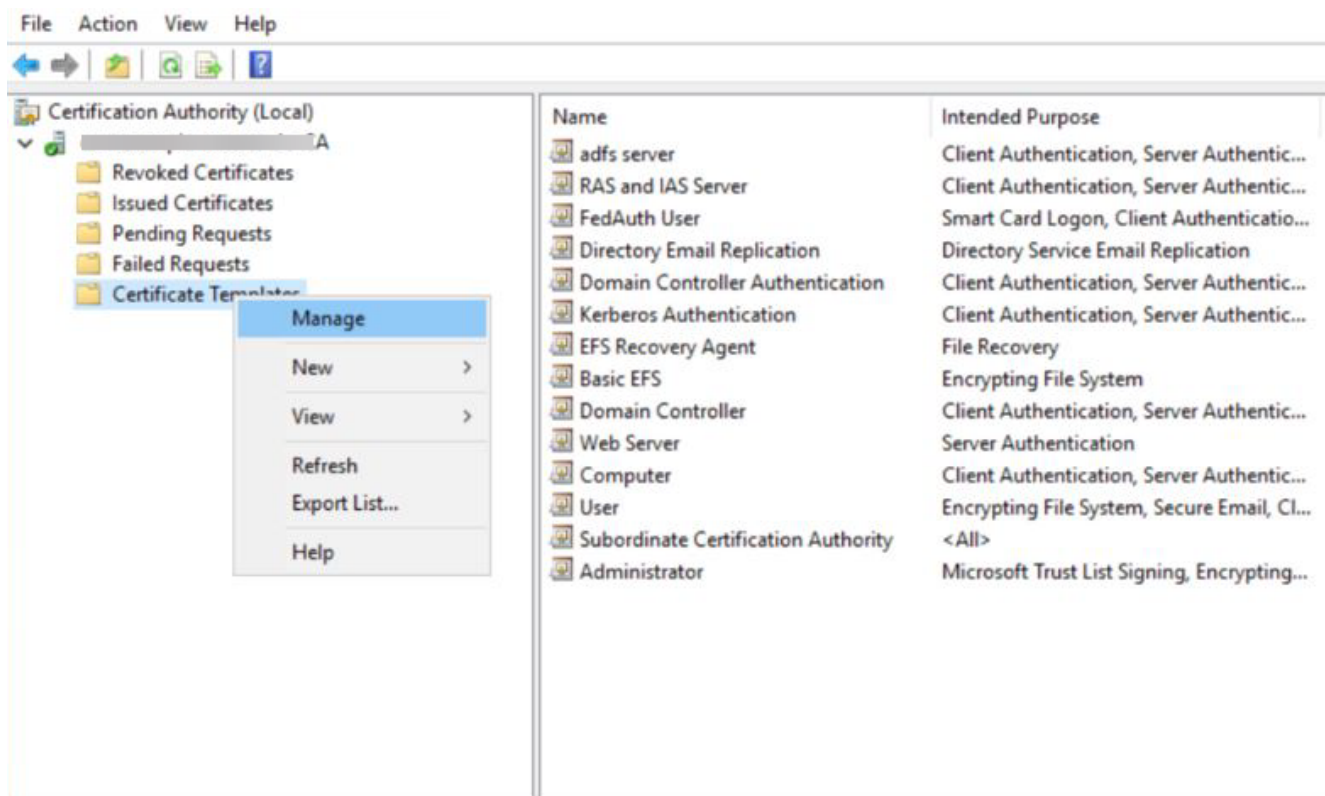
In order to support Single Sign On, CMSG needs to be able to enroll user in the Certification Authority and supports two options to enroll user:

- [By Active Directory Certification Authority Web Enrollment](#)
- [By private key and certificate of the Certification Authority](#)

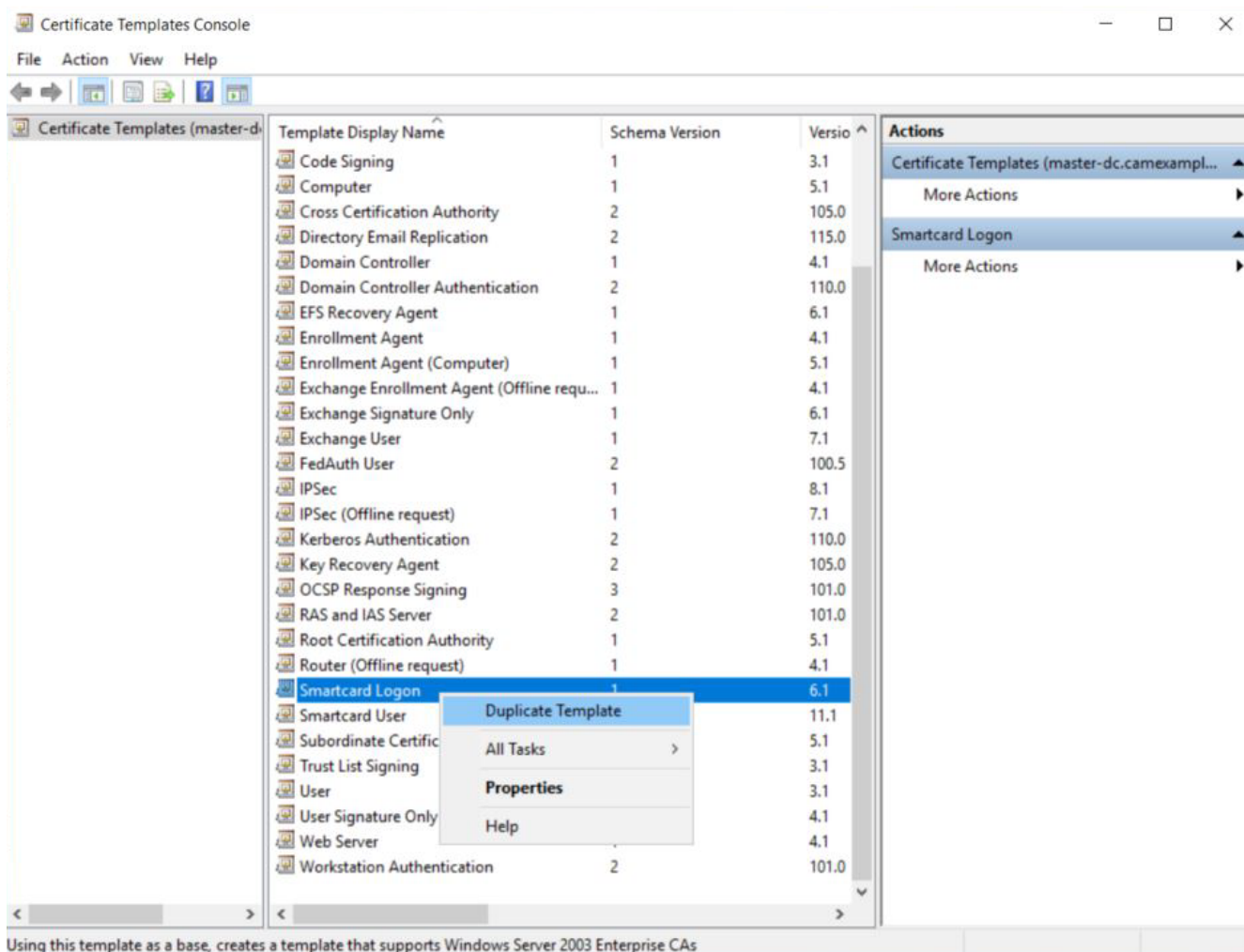
By Active Directory Certification Authority Web Enrollment

Create Certificate Authority Template

1. Log on to the Certificate Authority resource.
2. Open Certificate Authority MMC (`certsrv.msc`).
3. Right click the **Certificate Templates** and select **Manage**.



4. **Certificates Templates Console** window is now open. Right click **Smartcard User** and select **Duplicate Template**.



5. Navigate to the **General** tab and rename the template to a desired name and take note of the name as it is required during CMSG installation. Change the **Validity Period** and **Renewal Period** to minimum such as 1 hours and 0 hours respectively.
6. Navigate to **Request Handling** tab and change the purpose to **Signature and smartcard logon**. The **Certificate Templates** information box appears. Click **Yes** to close it.

Properties of New Template ✕

Subject Name	Server	Issuance Requirements
Superseded Templates	Extensions	Security
Compatibility	General	Request Handling
	Cryptography	Key Attestation

Purpose: Signature and smartcard logon ▾

- Delete revoked or expired certificates (do not archive)
- Include symmetric algorithms allowed by the subject
- Archive subject's encryption private key

Allow private key to be exported

Renew with the same key (*)

For automatic renewal of smart card certificates, use the existing key if a new key cannot be created (*)

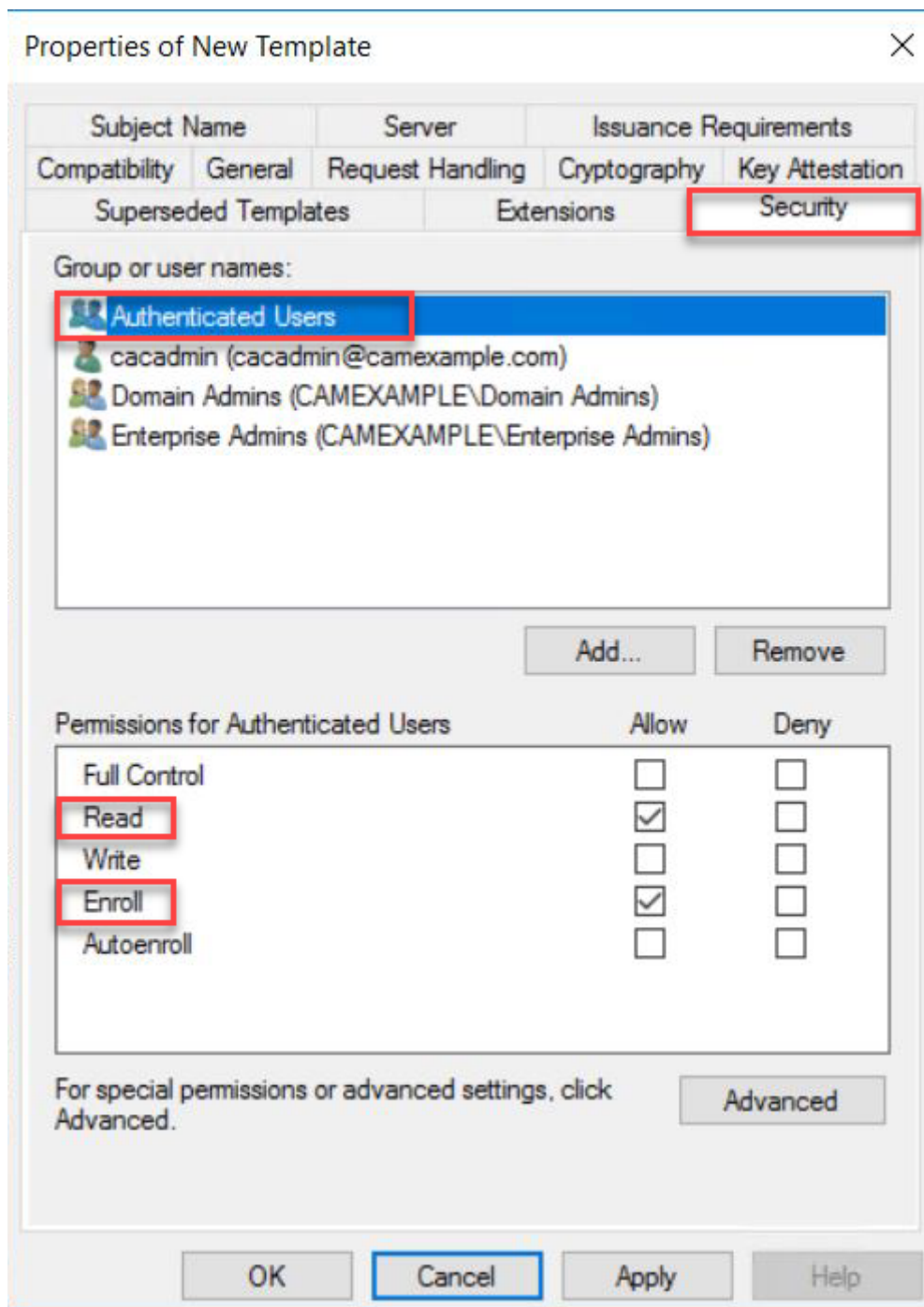
Do the following when the subject is enrolled and when the private key associated with this certificate is used:

- Enroll subject without requiring any user input
- Prompt the user during enrollment
- Prompt the user during enrollment and require user input when the private key is used

* Control is disabled due to [compatibility settings](#).

OK Cancel Apply Help

7. Navigate to **Security** tap and select **Read** and **Enroll** as **Allow** for **Authenticated Users**.



8. Navigate to **Subject Name** tab and select **Supply in the request**. A warning text box appears and click **OK** to close the warning text box.

Properties of New Template

Compatibility | General | Request Handling | Cryptography | Key Attestation

Superseded Templates | Extensions | Security

Subject Name | Server | Issuance Requirements

Supply in the request

Use subject information from existing certificates for autoenrollment renewal requests (*)

Build from this Active Directory information

Select this option to enforce consistency among subject names and to simplify certificate administration.

Subject name format:

None

Include e-mail name in subject name

Include this information in alternate subject name:

E-mail name

DNS name

User principal name (UPN)

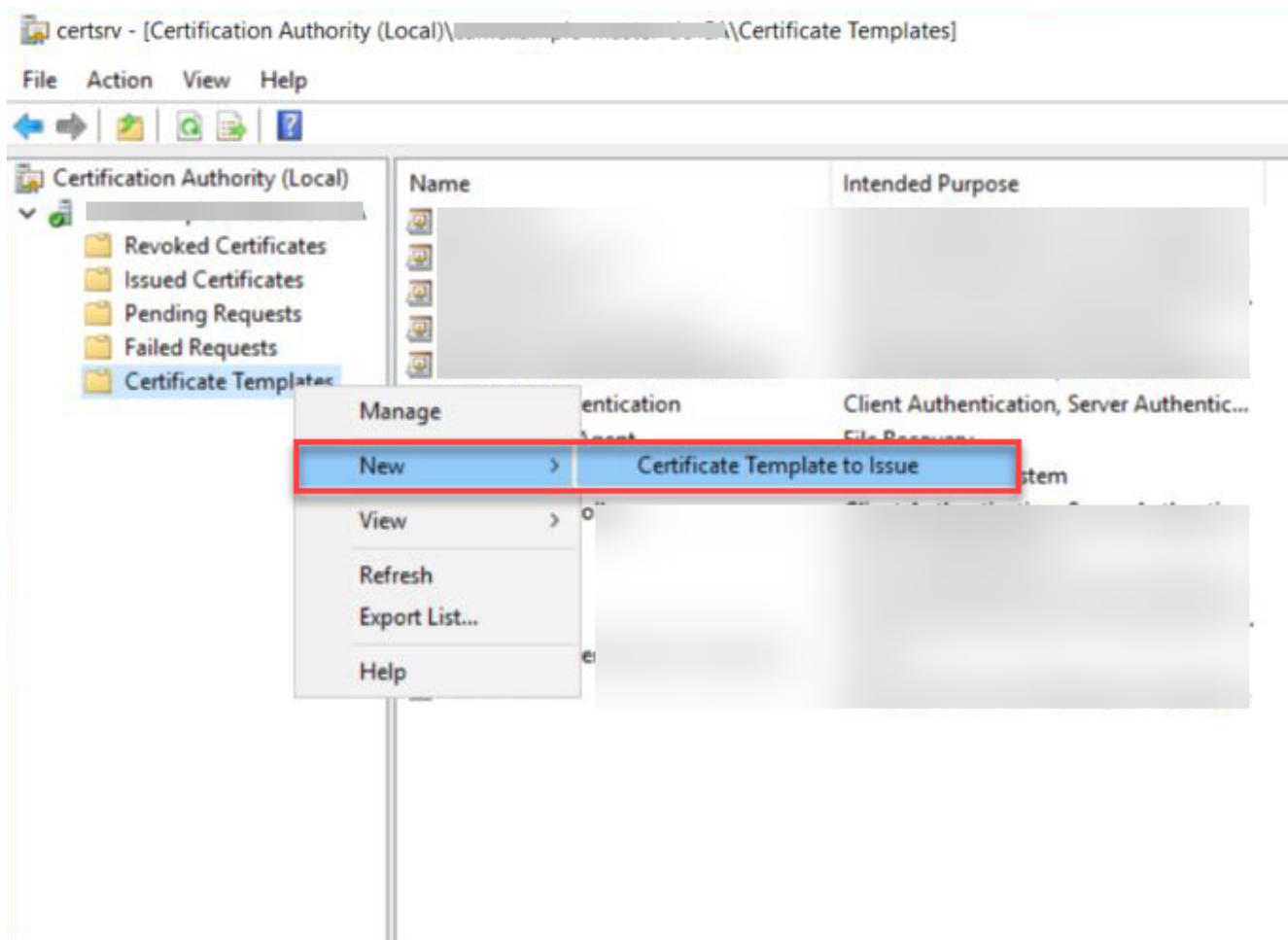
Service principal name (SPN)

* Control is disabled due to [compatibility settings](#)

OK Cancel Apply Help

9. Click **Apply** and then **OK** to finish creating the template.

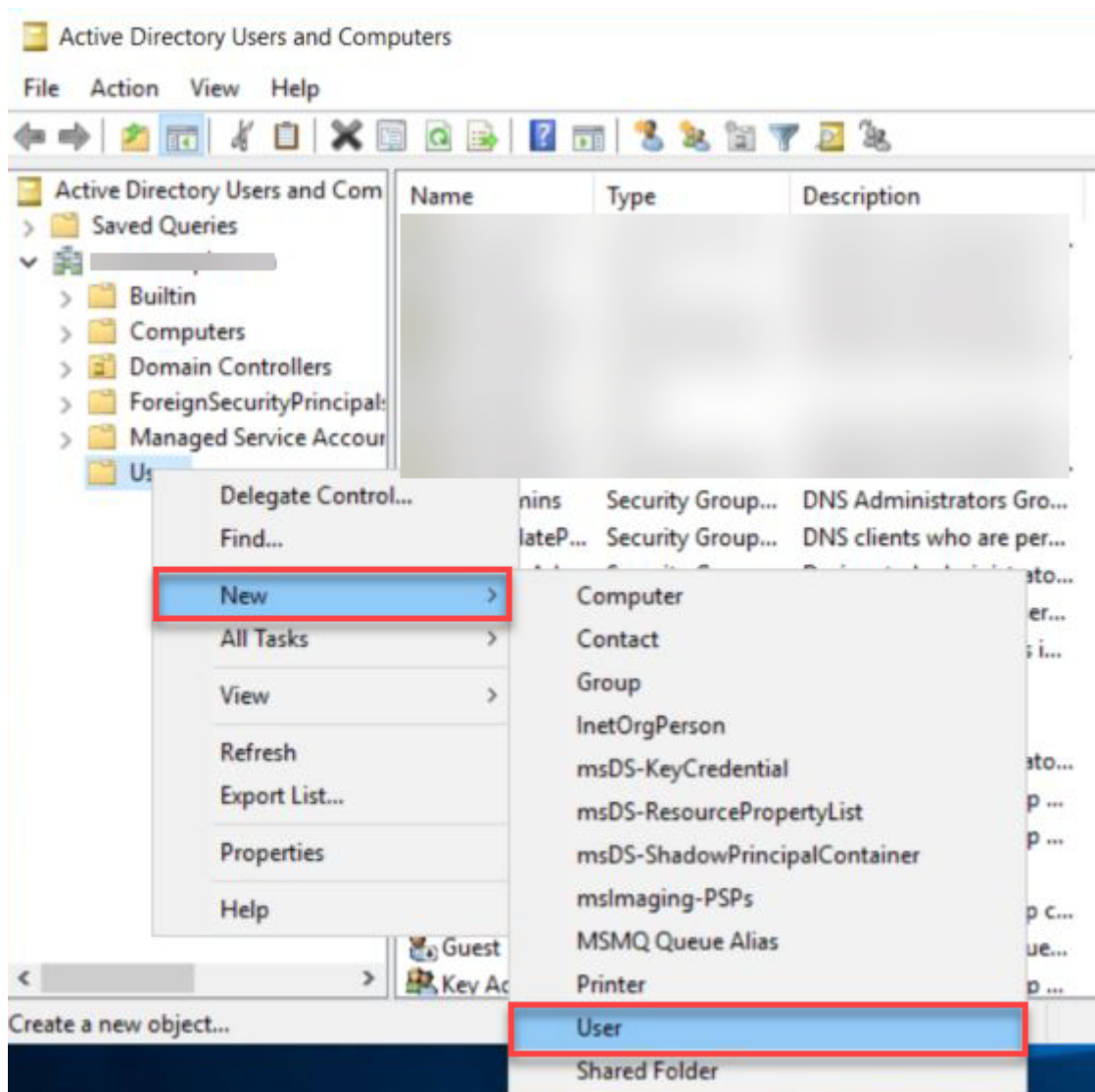
10. Right click the **Certificate Templates**, select **New** and click **Certificate Template to Issue**.



11. Select the template created above and click **OK** to add the template to CA.

Create a user who will have the permission to request Certificate

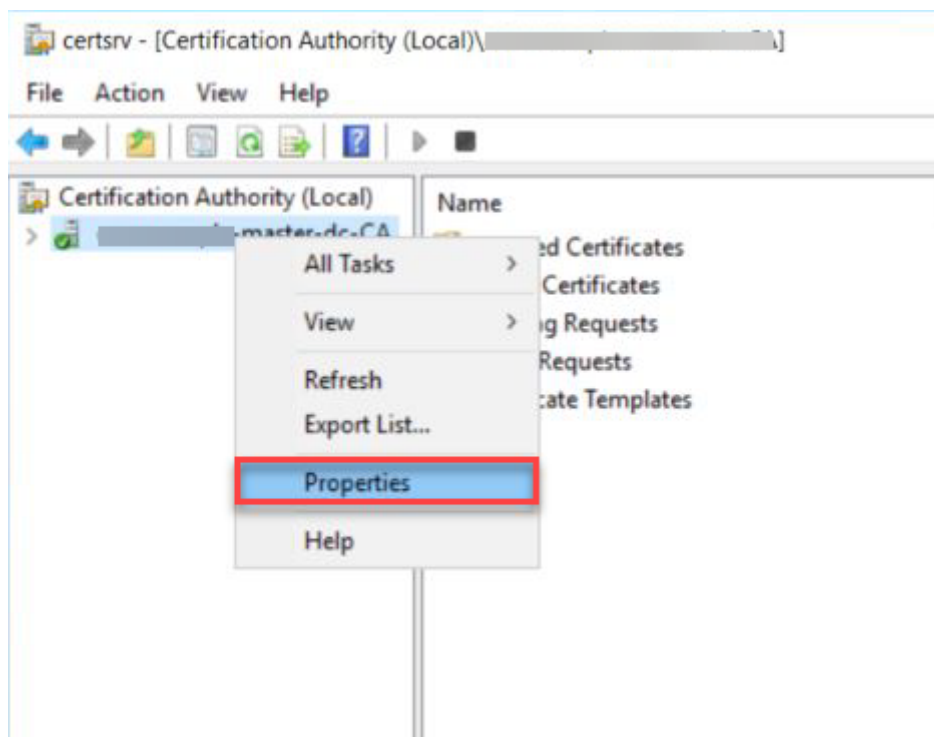
1. Logon to Domain Controller, open **Active Directory Users and Computers**.
2. Go to **\$Domain** and select **Users**.
3. Right click **Users** select **New** and click **Use**.



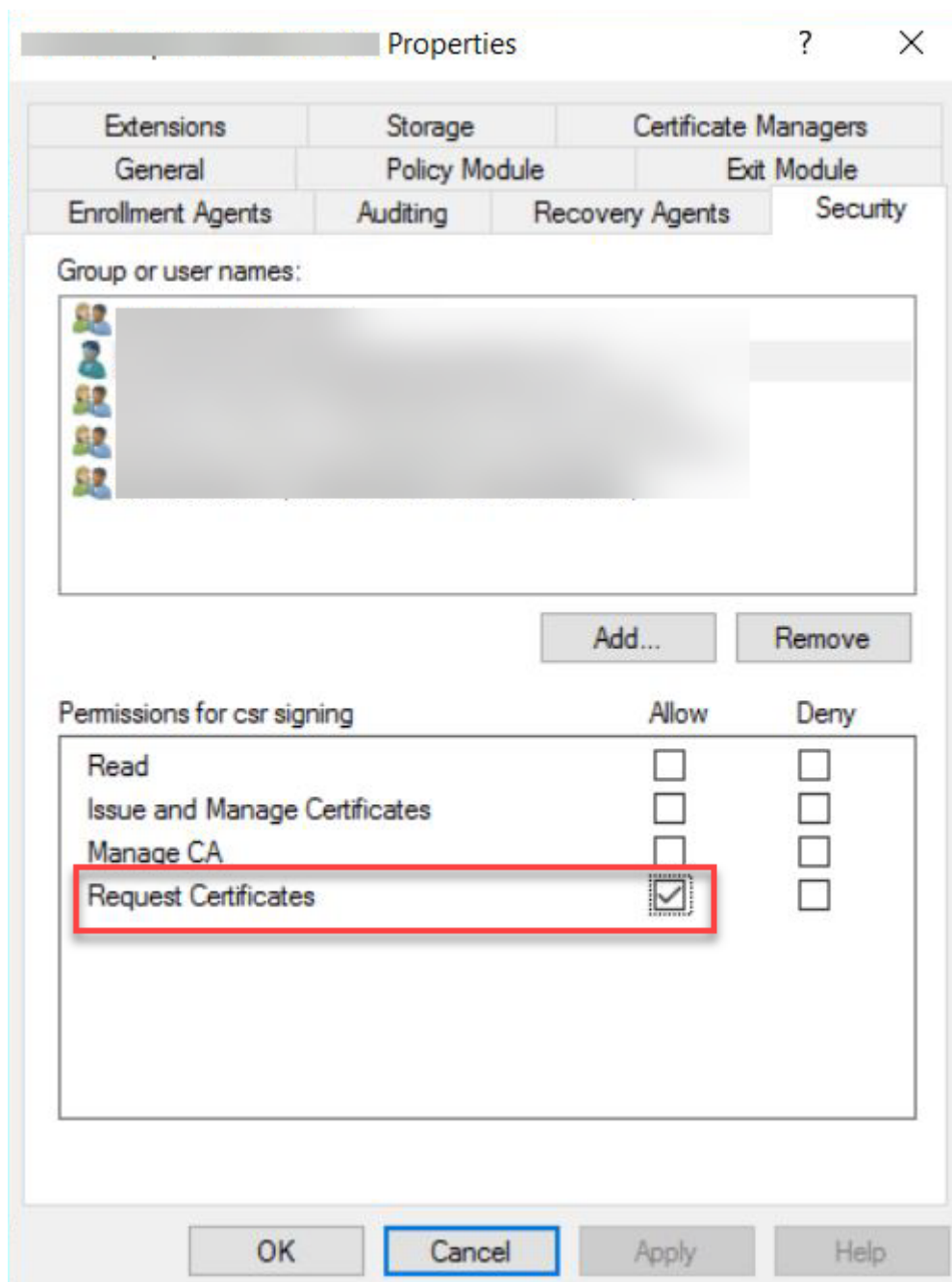
4. Enter the required information such as First name, Last name, User Logon name ...etc and click on **Next**.
5. Enter the Password for the user and click **Next**.
6. Note the username and password as it is required during CMSG installation.
7. Click on **Finish** to create the user.

Grant user the permission to request Certificate

1. Log on to the Certificate Authority machine
2. Open Certificate Authority MMC (`certsrv.msc`)
3. Right click the CA and select **Properties**.

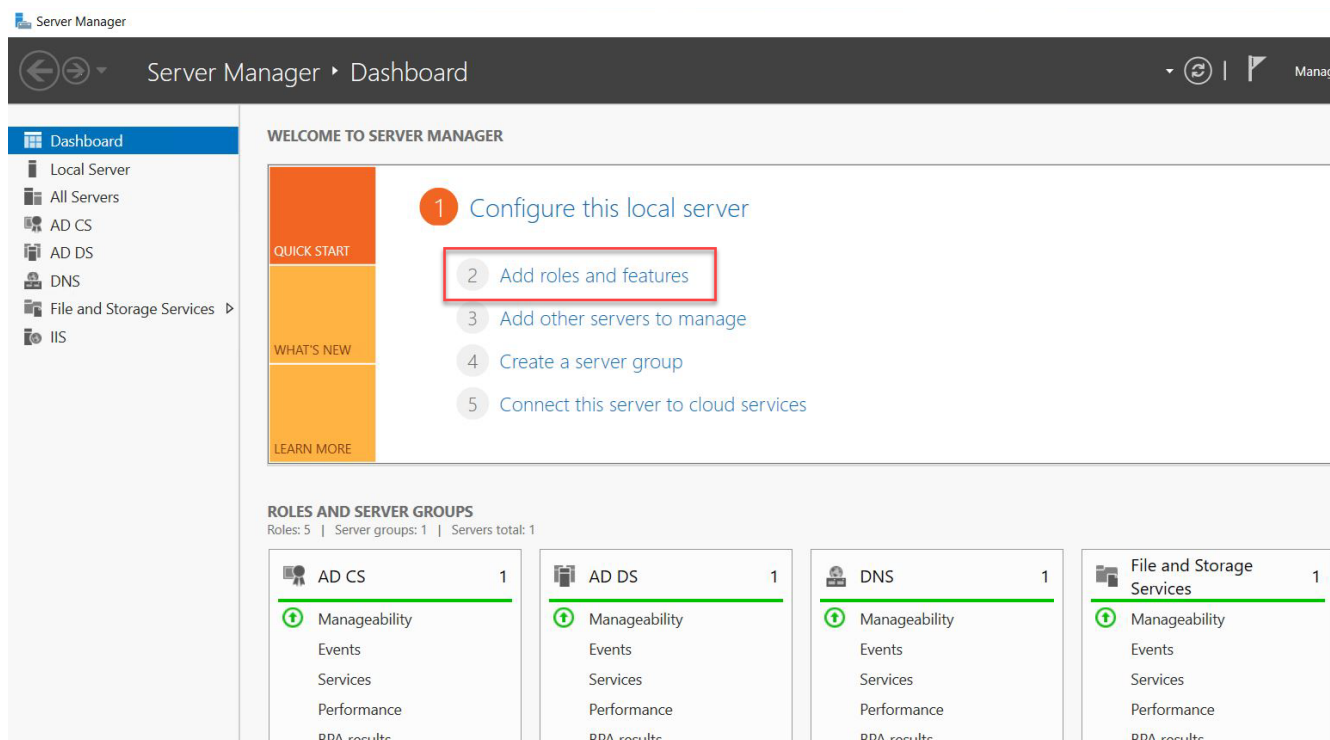


4. Navigate to **Security** tab and click **Add...** and add the user created above.
5. Ensure the user added is allowed to **Request Certificates**.



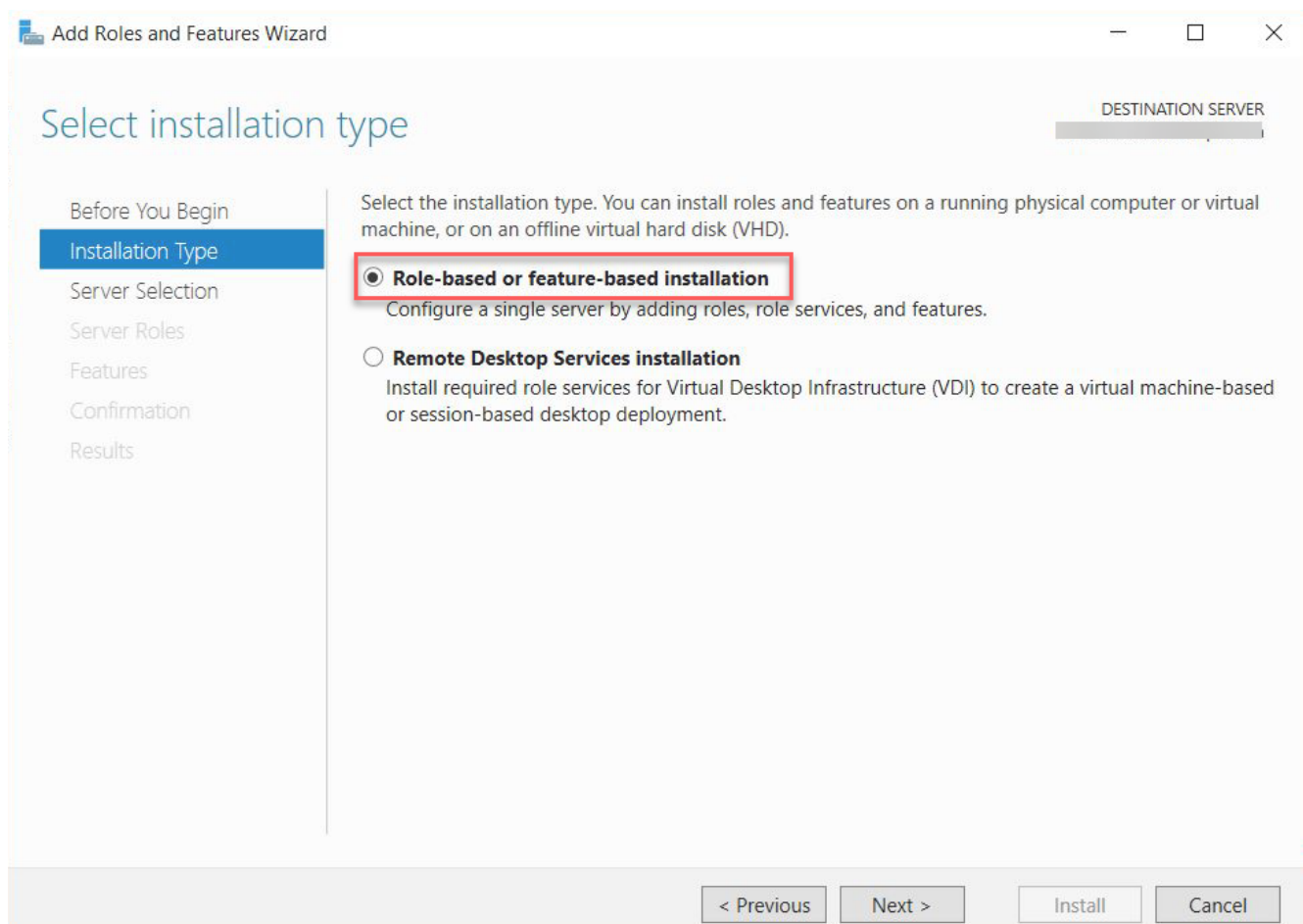
Set up Active Directory Certification Authority Web Enrollment

1. On a Windows Server machine where the Certification Authority is installed, select **Add roles and features** on the Server Manager window.

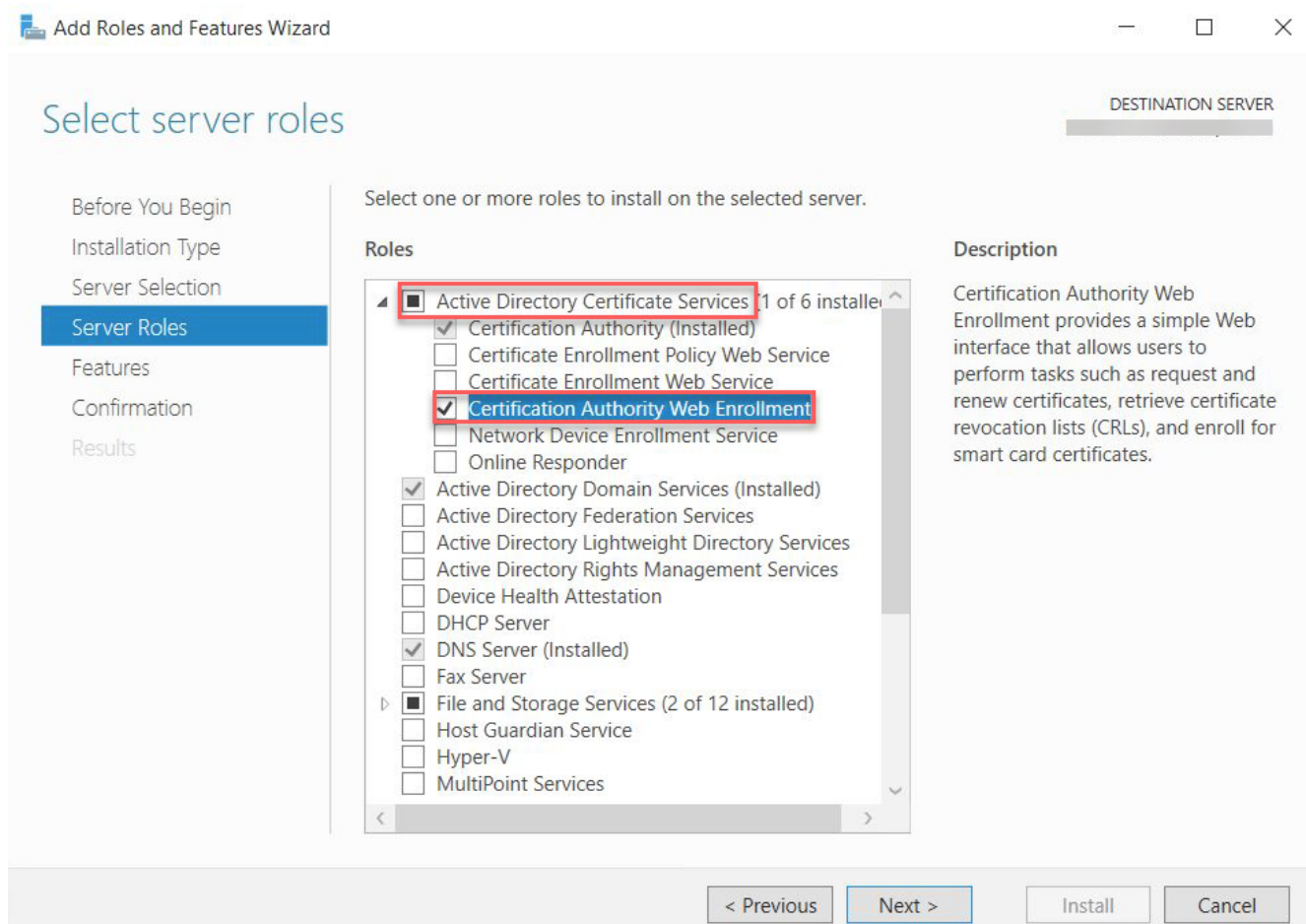


2. Click **Next** on the **Before you begin** window.

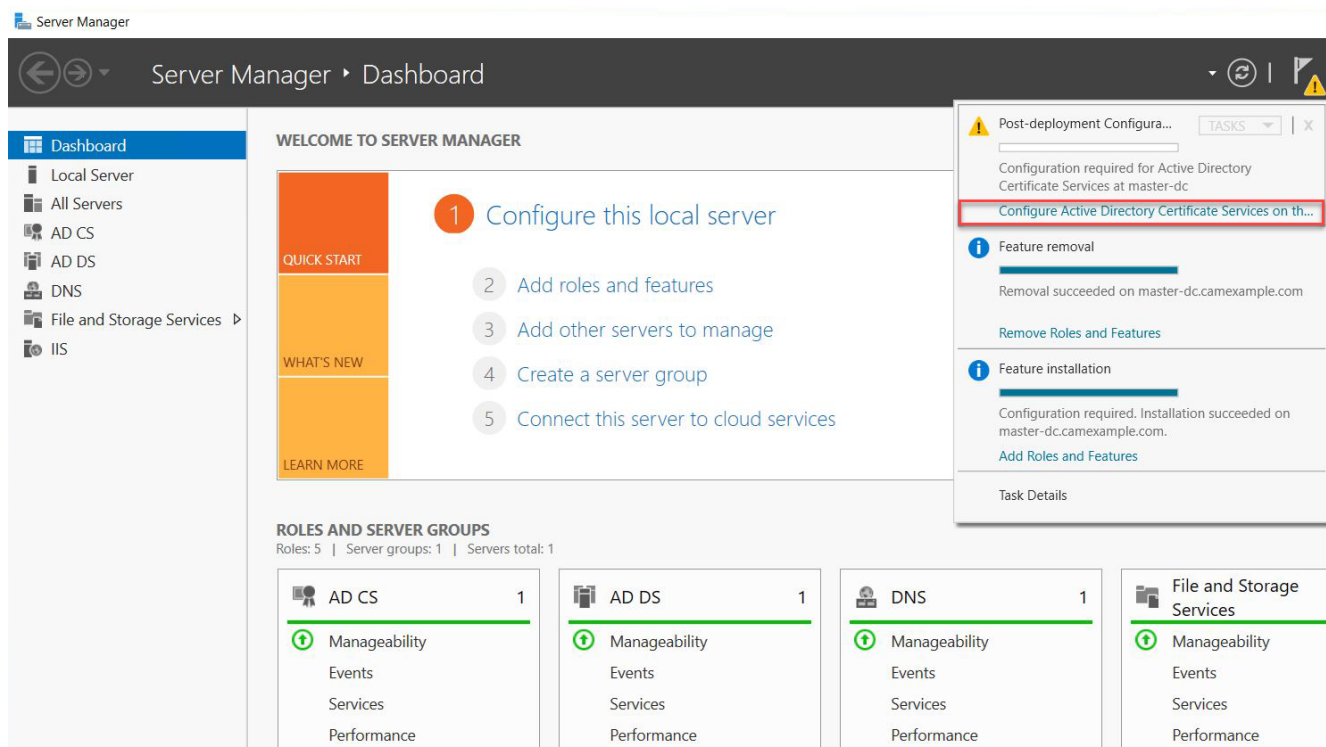
3. Select **Role-based or feature-based installation** on the **Installation Type** page.



4. Select a server from the server pool and press **Next**.
5. On the **Server Roles** page, expand **Active Directory Certificate Services** section, and select **Certificate Authority Web Enrollment**. Click **Next**.

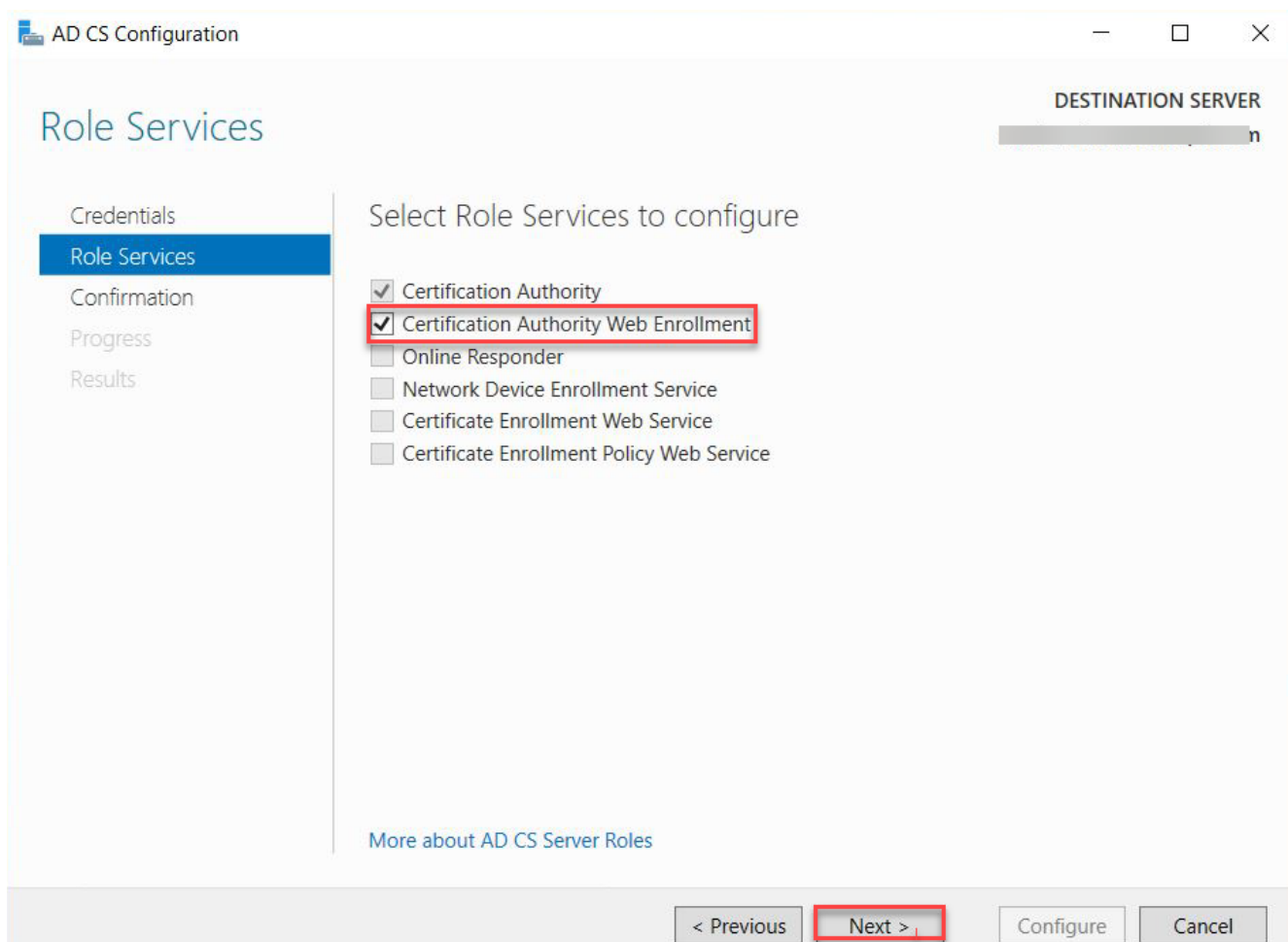


6. On the **Features** page, Click **Next**.
7. On the **Confirmation** page, select **Restart the destination server automatically if required** and press **Install**.
8. After installation, go to the **notification** tab and click **Configure Active Directory Certificate Services**.



9. On the **Credentials** page, input the Credentials and click **Next**.

10. On the **Role Services** page, select **Certification Authority Web Enrollment** and Click **Next**.



11. On the **Confirmation** page, click **Configure** to finish configuration.

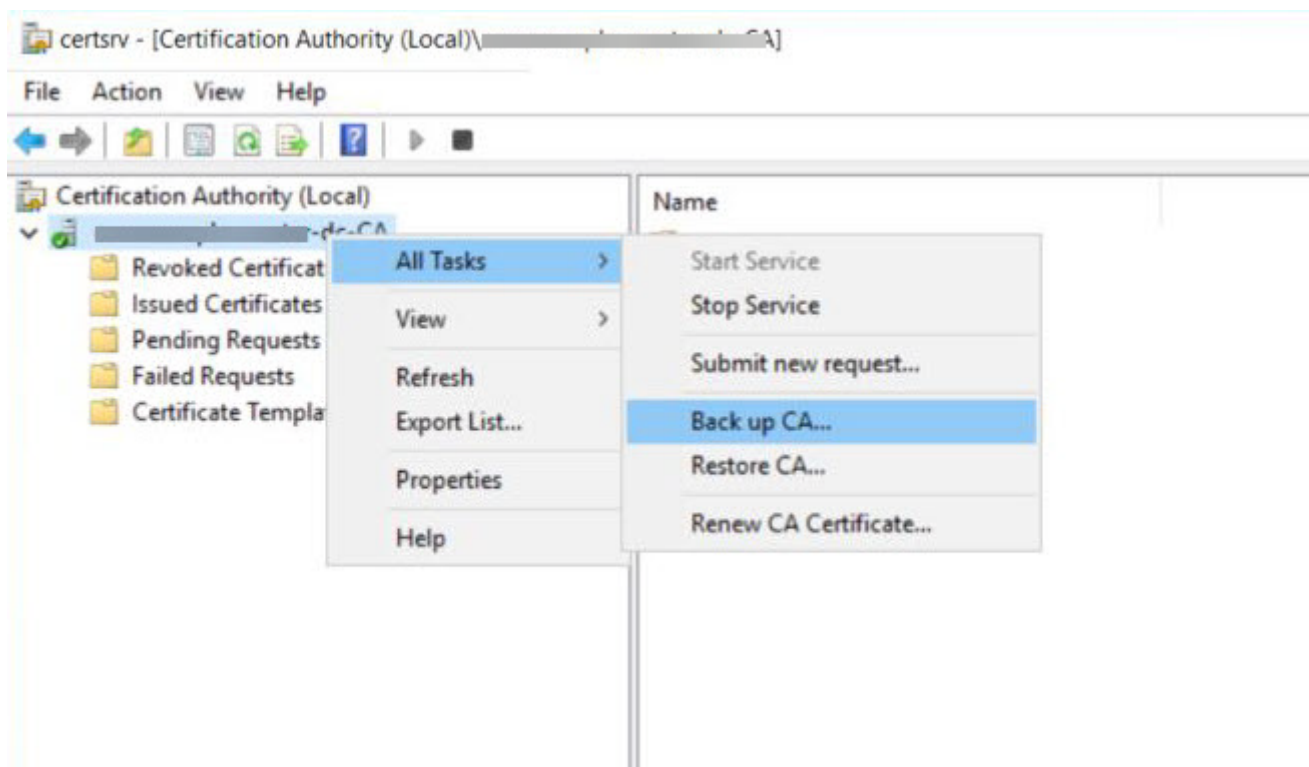
By private key and certificate of the Certification Authority

Working with your Certification Authority (CA) you will need to obtain:

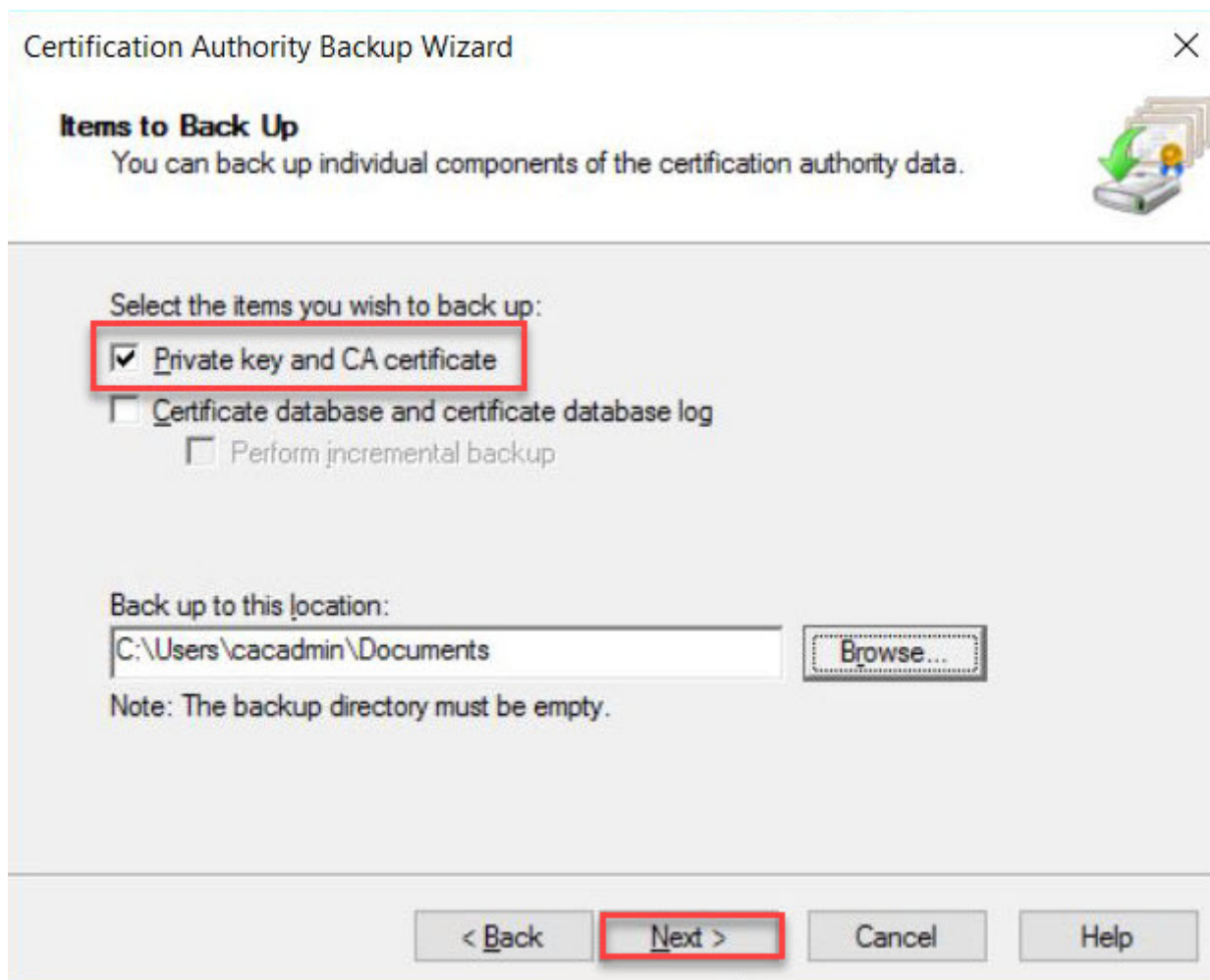
- Certificate of Intermediate CA
- Private Key of Intermediate CA
- Certificate Revocation List (CRL) file of the Intermediate CA

Export private key and certificate of the Intermediate Windows CA (Microsoft Windows Server 2019 Datacenter)

1. Log on to the Certificate Authority resource.
2. Open Certificate Authority MMC (`certsrv.msc`).
3. Right-click the CA in the tree, select **All Tasks** and click **Back up CA....**



4. In the **Certification Authority Backup Wizard** window, click **Next**.
5. In the **Items to Back Up** section, select **Private key and CA certificate** and click on **Browse...** to choose a location to save the file. Click on **Next** to go to next step.



6. Click **Finish** to finish exporting the private key and certificate of the CA. **Note:** The private key and certificate are in a single `p12` file.

Extract the private key and certificate from `p12` file:

On a resource such as Linux VM that has `openssl` available:

- Extract private key with `openssl`. Run the following command and enter password when prompted:

```
openssl pkcs12 -in <your .p12 file name>.p12 -nocerts -nodes -out <your private key file name>.key
```

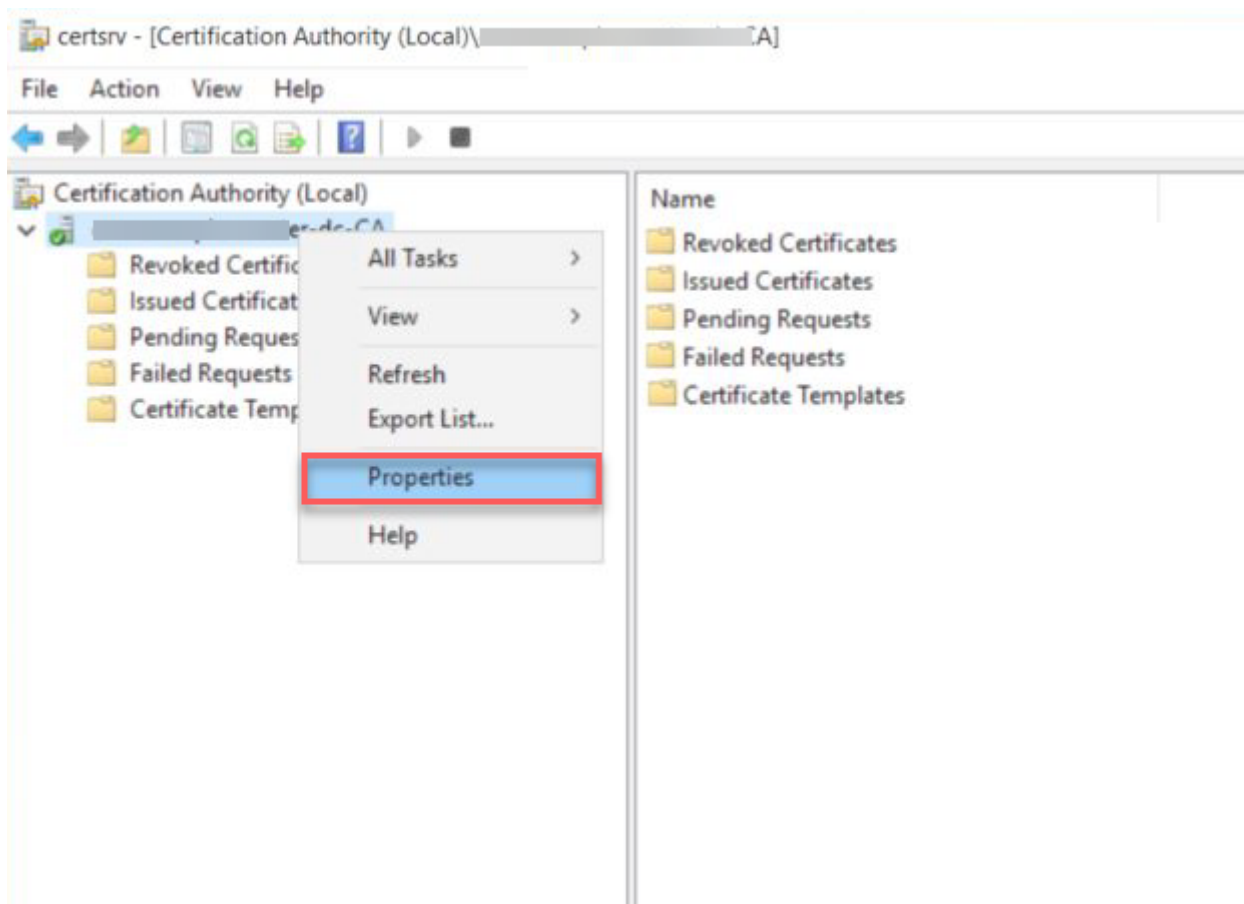
- Extract certificate with `openssl`. Run the following command and enter password when prompted:

```
openssl pkcs12 -in <your .p12 file name>.p12 -clcerts -nokeys -out <your certificate file name>.crt
```

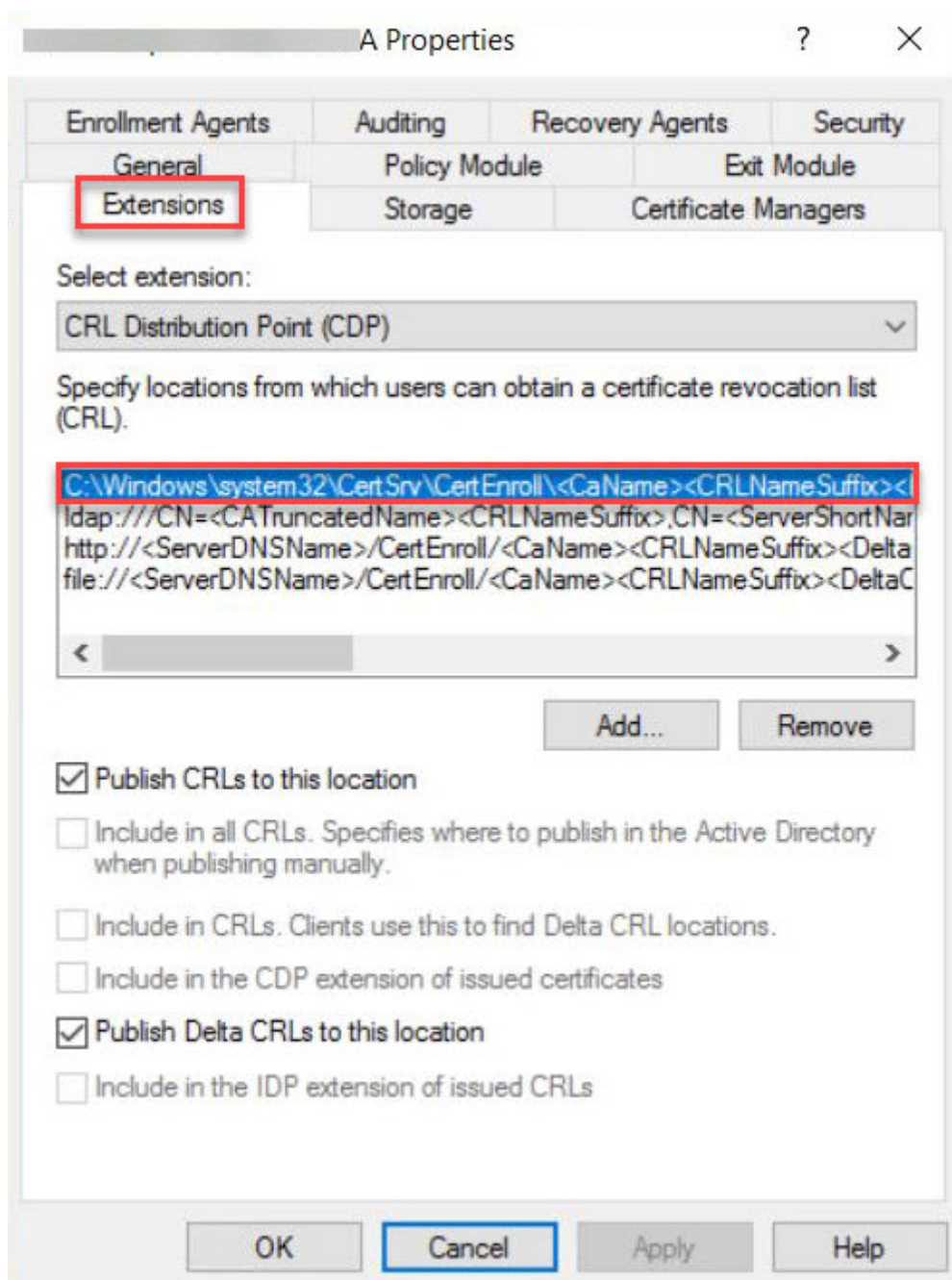
Locate Certificate Revocation List (CRL) file of the Intermediate Windows CA (Microsoft Windows Server 2019 Datacenter)

Perform the following steps:

1. Log on to the Certificate Authority resource, run `certsrv.msc` from command line to launch Certification Authority.
2. Right click the CA name and select **Properties**.



3. Select the **Extensions** tab, and take note of the `.cerl` path. In this example, it is `C:\Windows\System32\CertSrv\CertEnroll\<CA name>.cerl`.



After you have obtained the files, they should be uploaded via SFTP (using a tool such as SCP) to your CMSSG and ensure that they are available for future configurations.

Enable Federated Authentication for CMSG with SSO

IDP Configuration

Configuration parameters below are all obtained during the configuration of the IDP. Before configuring Federated Authentication, please ensure that you have an active third-party identity provider configured correctly.

To enroll by the private key and certificate of the Certification Authority:

For more information on all the enrollment options, see [Preparing for Single Sign-On](#)

Private Key and CA requirement

Ensure that you have the PEM files for the signed certificate, private key and certificate revocation list from the above instructions on Preparing for Single Sign-On, and have uploaded them to the CMSG.

Passphrase Protection

Passphrase protection for CA certificates is not supported.

If you are installing a new CMSG:

- Run this command: `sudo pcoip-cmsg-setup install [...other settings...] --enable-oauth true --id-provider-url https://id.provider.com --oauth-client-id XXXXXXXXXX --enable-ssso true --sso-signing-csr-ca <path to pem> --sso-signing-csr-key <path to pem> --sso-signing-crl <path to crl>`

If you are configuring an existing CMSG:

Empty flags

For configuring an existing CMSG, extra flags are included with "" as a value. These are provided to ensure those settings are cleared from the CMSG. If those settings had never been configured then those flags are not necessary to provide.

- Run this command:

```
sudo pcoip-cmsg-setup configure [...other settings...] --enable-oauth true --id-provider-url https://id.provider.com --oauth-client-id XXXXXXXXX --enable-sso true --sso-signing-csr-ca <path to pem> --sso-signing-csr-key <path to pem> --sso-signing-crl <path to crl> --sso-enrollment-url "" --sso-enrollment-domain "" --sso-enrollment-username "" --sso-enrollment-password "" --sso-enrollment-certificate-template-name ""
```

To enroll via Active Directory Certification Authority Web Enrollment:

If you are installing a new CMSG:

- Run this command: `sudo pcoip-cmsg-setup install [...other settings...] --enable-oauth true --id-provider-url https://id.provider.com --oauth-client-id XXXXXXXXX --enable-sso true --sso-enrollment-url "$Enrollment_URL" --sso-enrollment-domain "$Domain" --sso-enrollment-username "<username>" --sso-enrollment-password "<password>" --sso-enrollment-certificate-template-name "<template name>"`

If you are configuring an existing CMSG:

Empty flags

For configuring an existing CMSG, extra flags are included with "" as a value. These are provided to ensure those settings are cleared from the CMSG. If those settings had never been configured then those flags are not necessary to provide.

- Run this command: `sudo pcoip-cmsg-setup configure [...other settings...] --enable-oauth true --id-provider-url https://id.provider.com --oauth-client-id XXXXXXXXX --enable-sso true --sso-enrollment-url "$Enrollment_URL" --sso-enrollment-domain "$Domain" --sso-enrollment-username "<username>" --sso-enrollment-password "<password>" --sso-enrollment-certificate-template-name "<template name>" --sso-signing-csr-ca "" --sso-signing-csr-key "" --sso-signing-crl ""`

Installation Flags

Federated Authentication Flags

Flag	Type	Description
<code>--enable-oauth</code>	Boolean	Enables Oauth authentication. (Default=false)
<code>--id-provider-url</code>	String	Sets the identity provider URL. Example: <code>--id-provider-url https://provider-1234567890.id.provider.com</code> . This flag is required if <code>--enable-oauth is true</code> .
<code>--oauth-client-id</code>	String	Gets the Client ID from the Identity Provider. This flag is also required if <code>--enable-oauth is "true"</code> .

Federated Authentication Single Sign-On Flags

Flag	Type	Description
<code>--fa-url</code>	String	Override the fhe Federated Auth Broker URL provided to the PCoIP Agent. This flag can be used if auto-detection is not correcting determining the connector address. for example https://cac-vm-fqdn:port
<code>--enable-sso</code>	Boolean	Enables SSO. (Default=False)
<code>--sso-signing-csr-ca</code>	String	Path to copy intermediate CA Certificate.
<code>--sso-signing-csr-key</code>	String	Path to the intermediate key.
<code>--sso-signing-crl</code>	String	Path to a certificate revocation list.
<code>--sso-enrollment-url</code>	String	Gets the URL to the Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-domain</code>	String	Domain of the user to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-username</code>	String	Username for accessing Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-password</code>	String	Password for the username to access Active Directory Certification Authority Web Enrollment Service.
<code>--sso-enrollment-certificate-template-name</code>	String	Name of the certificate template that Active Directory Certification Authority Web Enrollment Service uses to sign CSR.

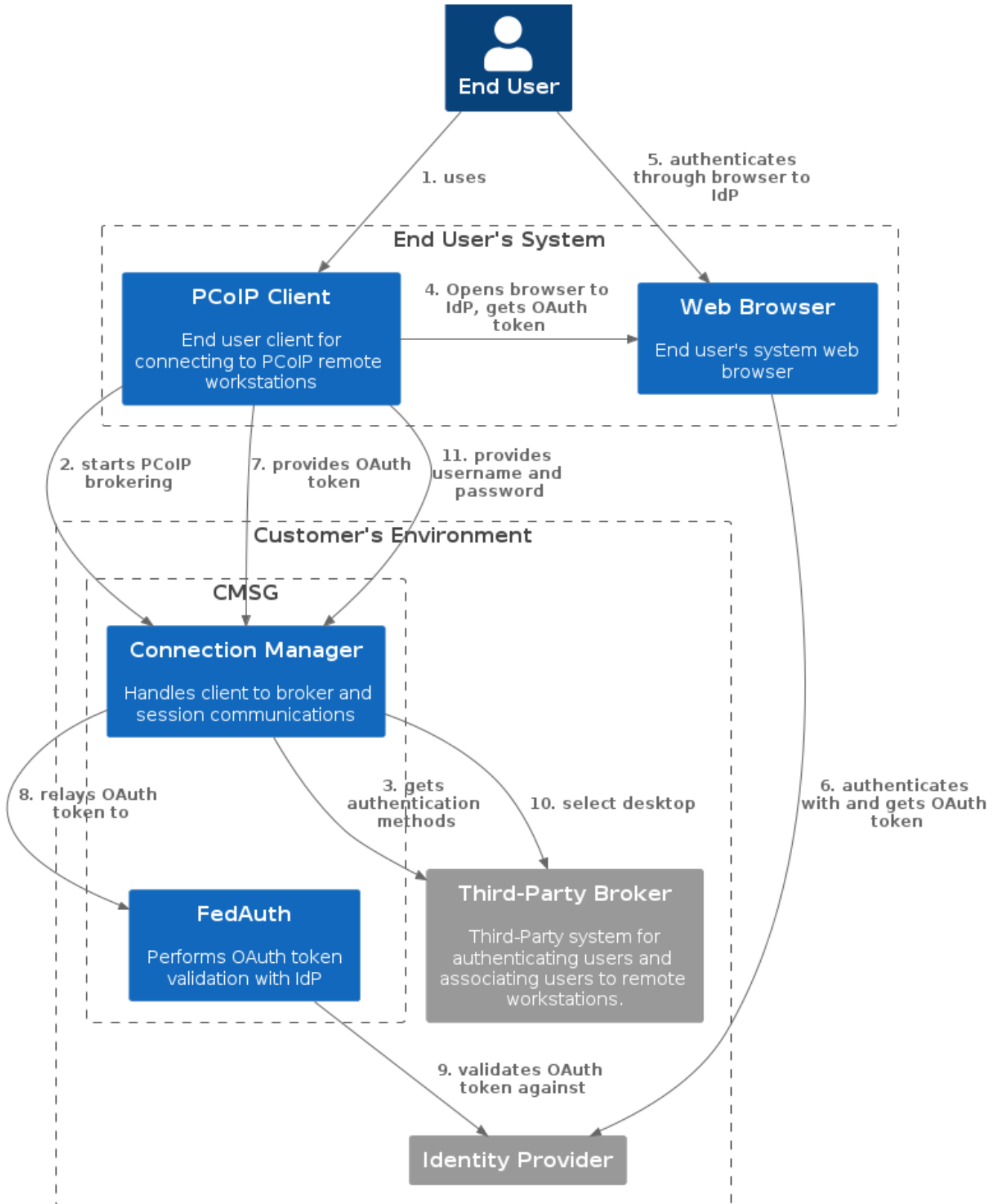
Troubleshooting Federated Authentication

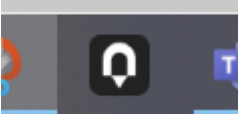
Federated Authentication Process Overview

Diagrams are provided that describe the steps that occur during authentication and session establishment through a CMSG for Federated User Authentication with or without Single Sign-On configured. The diagrams are numbered, so the flow can be followed by the numbers to determine which components are in use at any given step in the process, and instructions are provided for how to obtain logs from those components in the event of a failure.

AUTHENTICATION PROCESS

The diagram shows the process of authentication up until just before the start of a PCoIP Session for the case where Federated Authentication is configured, and Single Sign-On is not.



Step	Visual
1	
2	

Step**Visual**

Saved connections

To begin, select a connection.

3, 4

Step**Visual**

Saved connections

To begin, select a connection.



Step	Visual
5, 6	

Step	Visual

dev- ██████████ okta.com

Connecting to

Sign in with your ██████████996 account to access PCoIP Client



Sign In

Username

Password

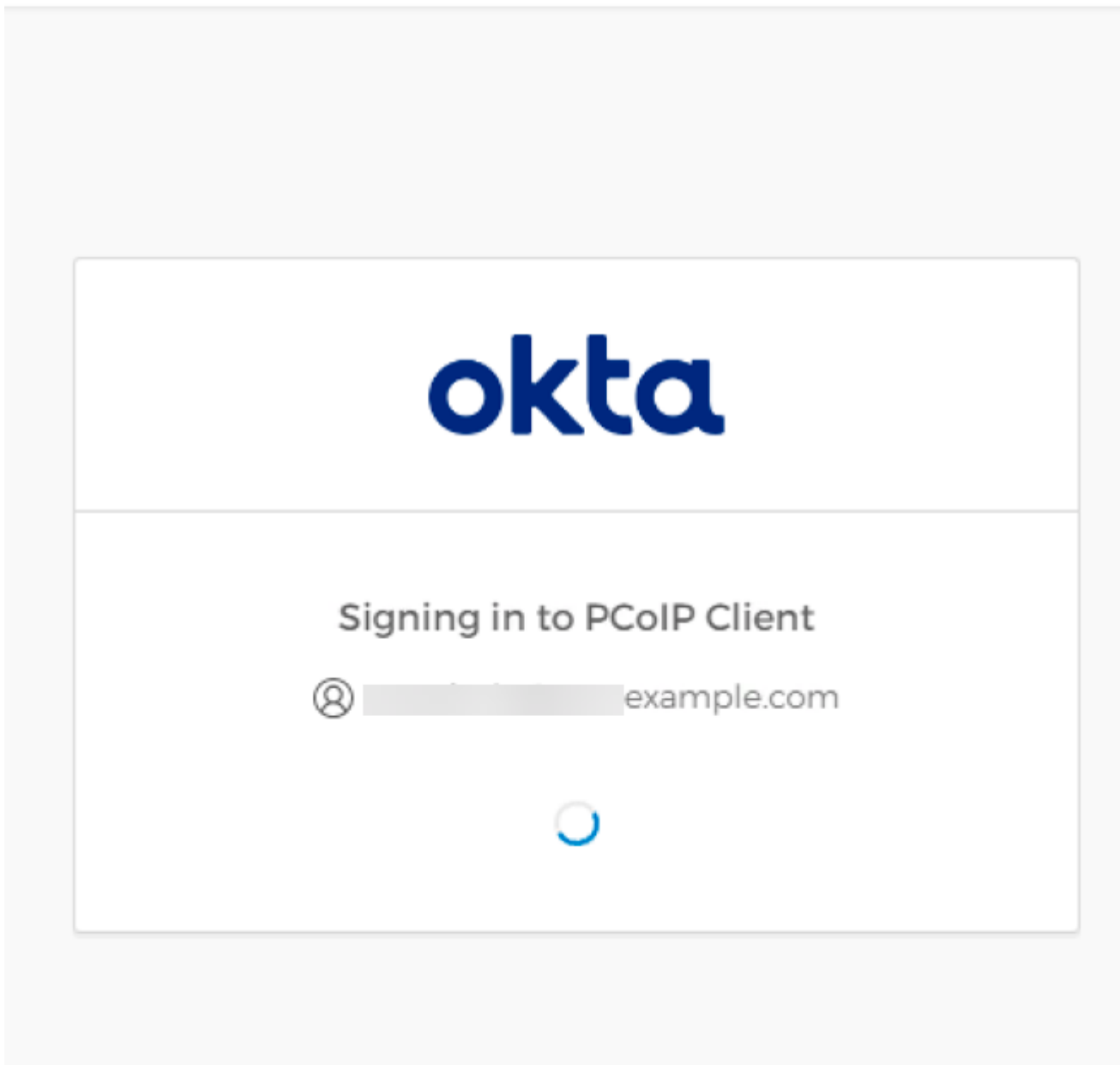
Keep me signed in

[Forgot password?](#)

[Help](#)

Step	Visual
------	--------

Connecting to 
Sign in with your okta- account to access PCoIP Client



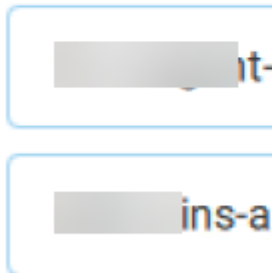
7	NA
---	----

Step	Visual
8, 9	

Step	Visual
------	--------

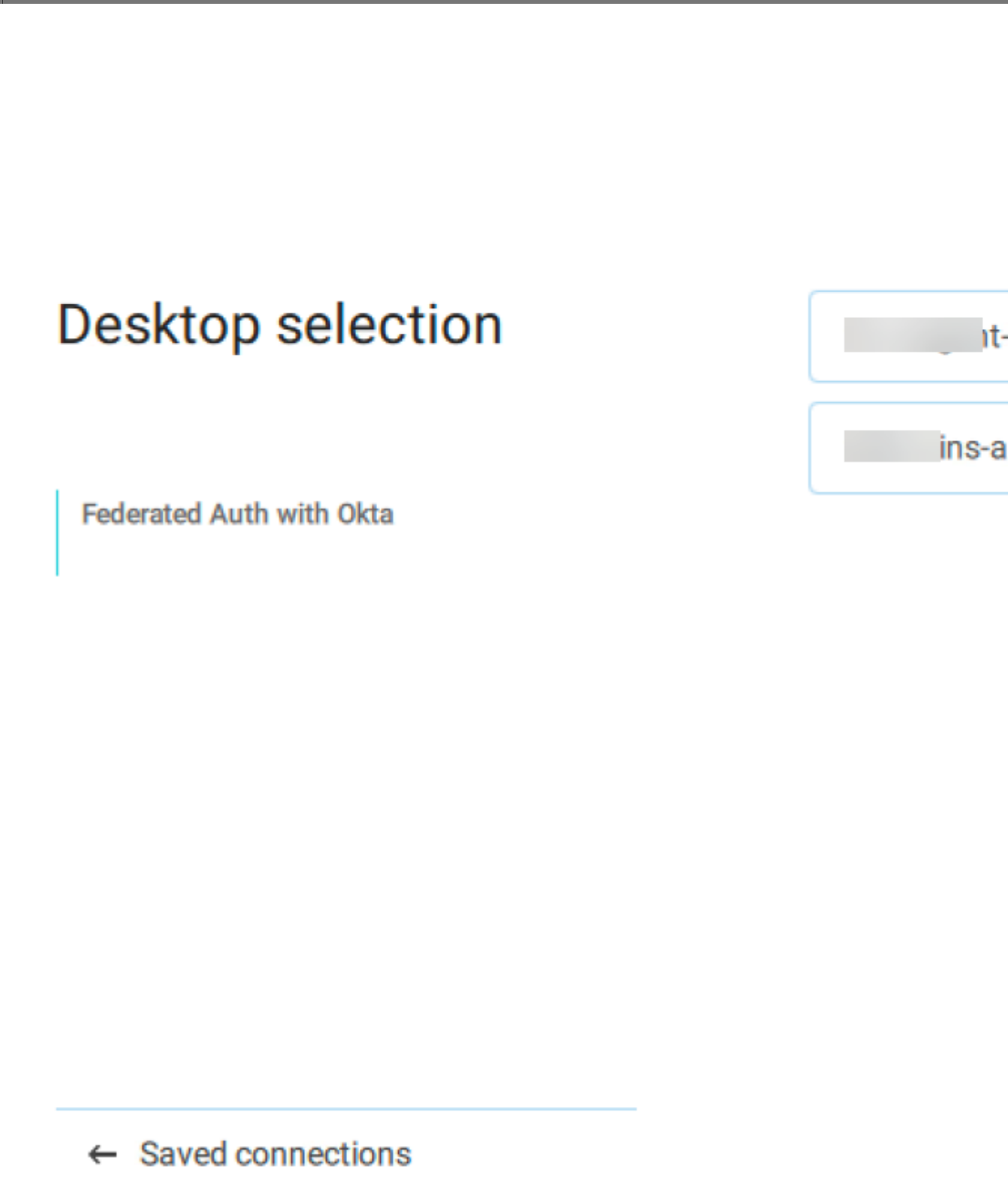
Desktop selection

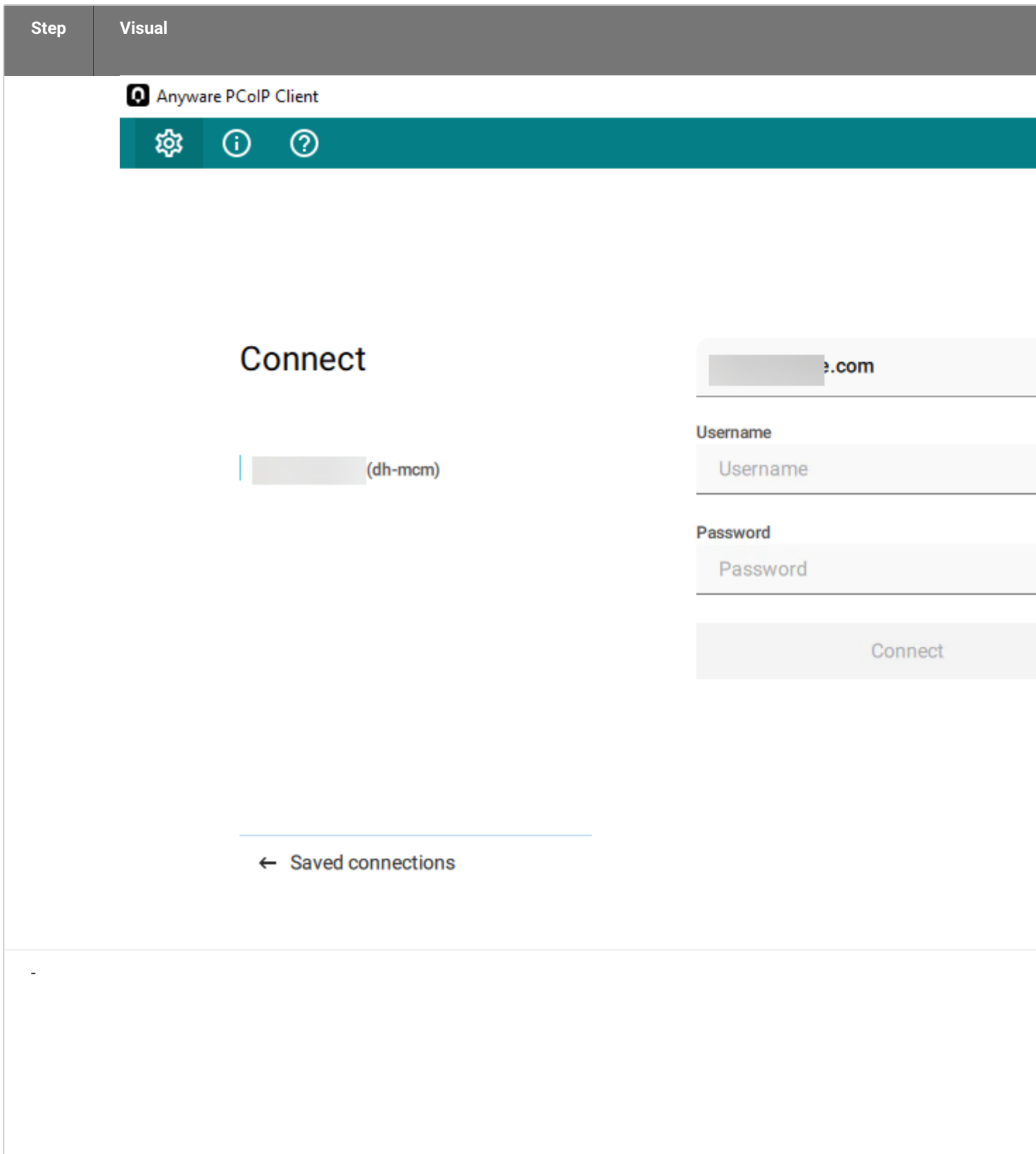
Federated Auth with Okta



← Saved connections



Step	Visual
10	 <p>The screenshot shows a software interface with a dark header. The main content area has a large heading "Desktop selection" in blue. Below it, there is a section titled "Federated Auth with Okta" with a vertical blue line to its left. On the right side, there are two rounded rectangular buttons with blue borders. The top button has a greyed-out label ending in "it-". The bottom button has a greyed-out label ending in "ins-a". At the bottom of the interface, there is a blue horizontal line and a button with a left-pointing arrow and the text "Saved connections".</p>
11	

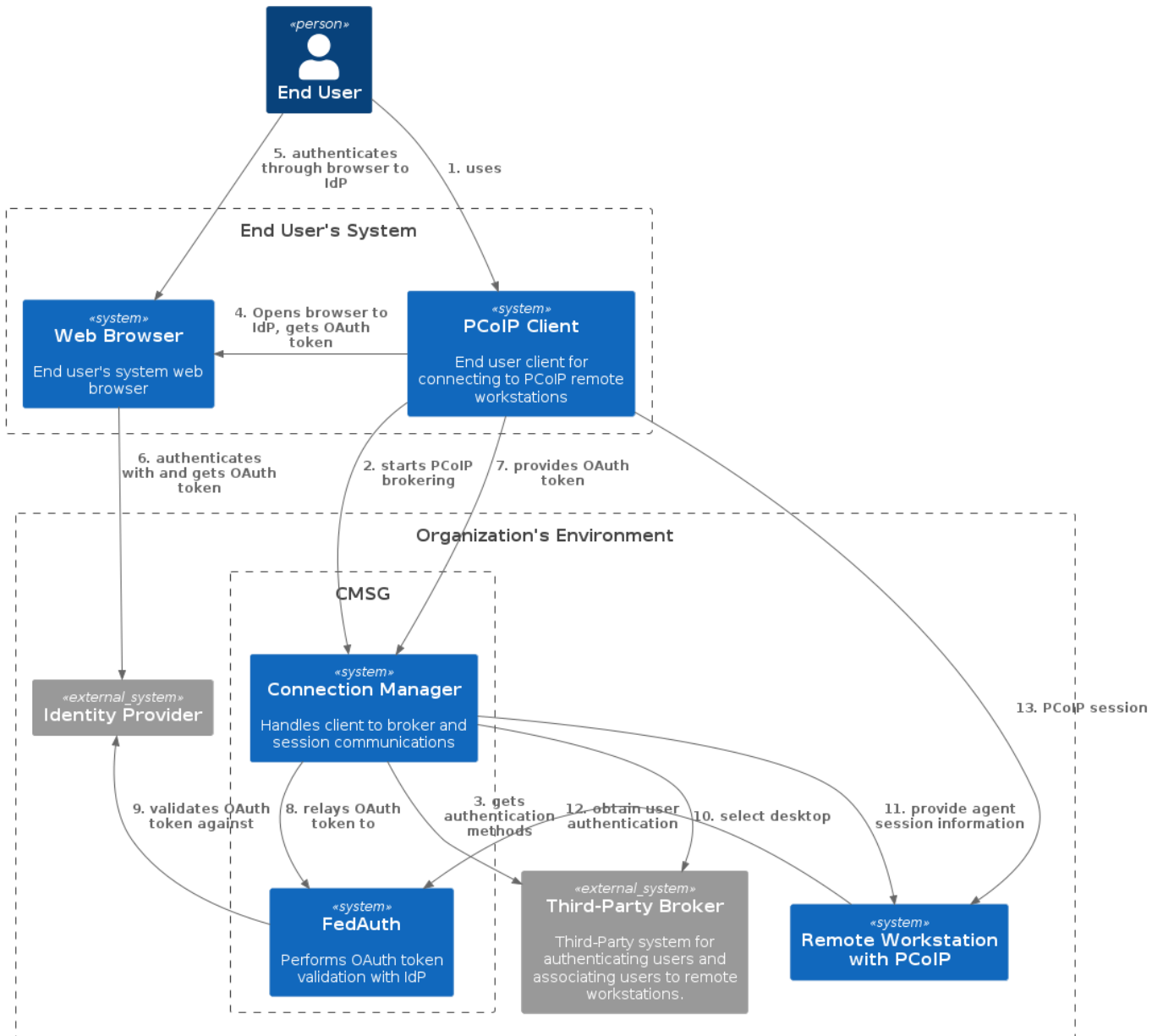


Single Sign-On

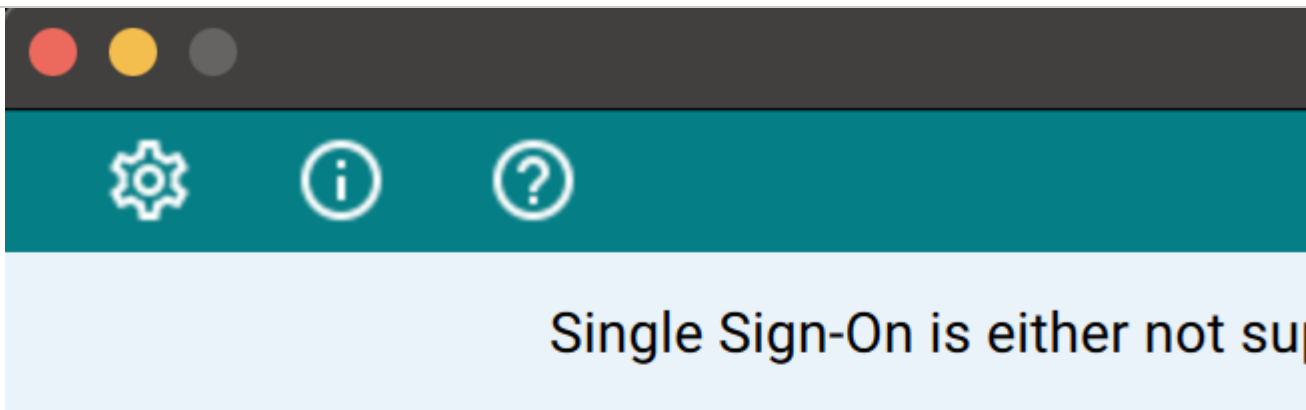
The diagram describes the steps to authenticate to a CMSG and select a desired remote workstation desktop using Federated User Authentication with Single Sign-On.

The table continues from the table above for Federated Authentication and adds the steps for where Single Sign-On is configured and attempted.

Federated User Authentication with SSO



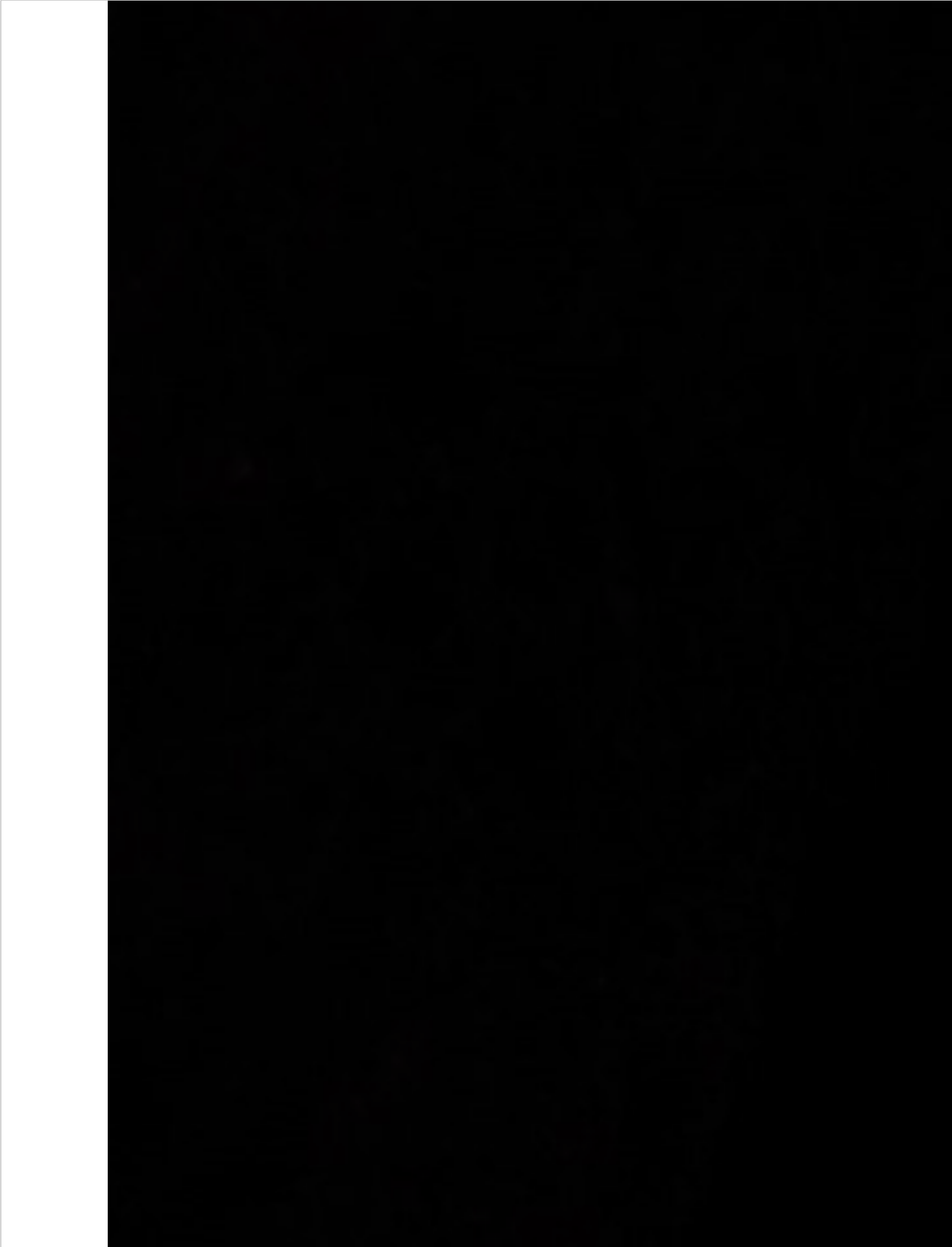
Step	Visual
11	



Connect

| Federated Auth with Okta

Step	Visual
12	



Step	Visual
13	

OBTAINING LOGS

The table above describes the components that may contain logs to describe errors if a failure occurs. This section provides information or references to how to obtain logs for each HP provided component:

• PCoIP Client

- Windows: https://www.teradici.com/web-help/pcoip_client/windows/current/support/logs/
- Linux: https://www.teradici.com/web-help/pcoip_client/linux/current/support/logs/
- MacOS: https://www.teradici.com/web-help/pcoip_client/mac/current/support/logs/

• Connector

- Overview: https://www.teradici.com/web-help/pcoip_connection_manager_security_gateway/current/troubleshooting/log_files/

• **Connection Manager:** `sudo docker service logs pcoipcm_cm`

• **Federated Auth service:** `sudo docker service logs pcoipcm_fa`

 **Agent logs are only necessary in troubleshooting if using SSO**

• PCoIP Agent

- Windows Standard Agent: https://www.teradici.com/web-help/pcoip_agent/standard_agent/windows/current/admin-guide/diagnostics/locating-log-files/
- Windows Graphics Agent: https://www.teradici.com/web-help/pcoip_agent/graphics_agent/windows/current/admin-guide/diagnostics/locating-log-files/

Reference

Using a PCoIP License Server with the Connection Manager

Using a PCoIP License Server with the PCoIP Connection Manager

In most cases, PCoIP licenses are validated automatically using HP's Cloud Licensing Service. In deployments where PCoIP agents cannot reach the public internet, a PCoIP License Server can be used to handle license validation instead. PCoIP License servers can be hosted on-premises or in any public or private cloud environment.

To use the PCoIP Connection Manager with a PCoIP License Server, you must configure the PCoIP Connection Manager with the address of the license server and the address of the connection broker.

Use the `install` and `configure` commands to configure the PCoIP License Server information:

```
pcoip-cmsg-setup install --license-server-url https://<license-server-address>:<port>
pcoip-cmsg-setup configure --license-server-url https://<license-server-address>:<port>
```

For more information about the PCoIP License Server, see the following guides:

- [PCoIP License Server Administrators' Guide \(Online deployments\)](#)
- [PCoIP License Server Administrators' Guide \(Offline deployments\)](#)

PCoIP Connection Manager and Security Gateway RPM Package Contents

The following table shows the files installed by the RPM packages:

Folder or file path	Type	Description
/sbin/pcoip-cmsg-setup	File	binary installation file
/usr/local/teradici/conf/docker-compose.yaml	File	docker-compose template file
/usr/local/teradici/conf/docker-compose-ipv6.yaml	File	docker-compose file for IPv6
/usr/local/teradici/conf/	Folder	configuration files templates
/usr/local/teradici/licenses/	Folder	license information files

After installation, the following files and folders are present:

Folder or file path	Type	Description
/var/log/Teradici/pcoip-cmsg-setup/	Folder	pcoip-cmsg-setup installation log files
/var/lib/docker/containers	Folder	All docker-related folders and files
/opt/teradici/pcoipcm_data/docker-compose.yaml	File	Template file used to generate docker containers
/opt/teradici/pcoipcm_data/certs/	Folder	Required certificates
/opt/teradici/pcoipcm_data/data/	Folder	
/opt/teradici/pcoipcm_data/data/SecurityGateway.conf	File	Security Gateway configuration file
/opt/teradici/pcoipcm_data/data/config.properties	File	Connection Manager configuration file
/opt/teradici/pcoipcm_data/data/cmsg.env	File	Environment file that configures shared data for both Connection Manager and Security Gateway

TLS Cipher Suites

This page contains information about the TLS Cipher Suites used by the PCoIP Connection Manager and PCoIP Security Gateway, and instructions for restricting the full list to subsets if desired.

TLS Versions

The PCoIP Connection Manager and Security Gateway supports TLS 1.2 and TLS 1.3.

PCoIP Connection Manager TLS Cipher Suites

The PCoIP Connection Manager supports the following cipher suites for the TLS connections from the PCoIP client, to the connection broker, and to the PCoIP Agent (in decreasing order of preference):

- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- TLS_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_GCM_SHA256

PCoIP Security Gateway Supported TLS Cipher Suites

The PCoIP Security Gateway supports the following cipher suites for TLS connections, in decreasing order of preference:

- TLS_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384

- TLS_RSA_WITH_AES_128_GCM_SHA256

Troubleshooting

Troubleshooting Connectivity Issues

Network Connectivity Problems

Connectivity issues are often caused by firewall misconfiguration. If you are unable to establish PCoIP connections, verify that sent packets are actually being received at the intended destination.

Useful tools for troubleshooting this type of issue are **ssldump** or **tcpdump** (for Linux), or **Wireshark** (for Windows).

The network connections between the following endpoints all need to be operational for a PCoIP session to be successful.

Connection	Port	Source
PCoIP Connection Manager	443 TCP	PCoIP Client
Connection Broker	443 (configurable) TCP	PCoIP Connection Manager
PCoIP Agent	60443 TCP	PCoIP Connection Manager
When Security Gateway is enabled		
PCoIP Security Gateway	4172 TCP/UDP	PCoIP Client
PCoIP Agent	4172 TCP/UDP	PCoIP Security Gateway
When Security Gateway is disabled (Direct Connection)		
PCoIP Agent	4172 UDP/TCP	PCoIP Client

Methods for testing communication between components on required ports are given next:

- [PCoIP Client to PCoIP Connection Manager](#)
- [PCoIP Connection Manager to Connection Broker](#)
- [PCoIP Connection Manager to PCoIP Agent](#)

- [PCoIP Client to PCoIP Security Gateway](#)
- [PCoIP Security Gateway from PCoIP Client \(UDP\)](#)
- [PCoIP Agent from PCoIP Client \(UDP\)](#)

Methods are also given for verifying the component availability:

- [Verifying PCoIP Agent availability](#)
- [Verifying Connection Broker availability](#)
- [Verifying PCoIP Connection Manager Web Application Availability](#)

Connectivity from PCoIP Client to PCoIP Connection Manager

This check looks for traffic between the PCoIP Client and the PCoIP Connection Manager on TLS port 443.

To verify connectivity from a PCoIP Client to PCoIP Connection Manager:

1. On the server hosting the PCoIP Connection Manager, start `ssldump`:

```
sudo ssldump -i <interface> host <client-ip-address> port 443
```

2. From the client, connect to the PCoIP Connection Manager.
3. In the `ssldump` output, look for packets originating from the PCoIP Client.

Connectivity from PCoIP Connection Manager to Connection Broker

This check looks for traffic between the PCoIP Connection Manager and a broker on TLS port 443.

To verify connectivity from the PCoIP Connection Manager to a connection broker connectivity:

1. On the server hosting the connection broker, use `ssldump` or **Wireshark** to capture packets from the PCoIP Connection Manager on TLS port 443.
2. From the client, connect to the PCoIP Connection Manager.
3. Try to authenticate.
4. Verify from `ssldump` or Wireshark output that the connection broker is receiving data from the PCoIP Connection Manager.

Connectivity from PCoIP Connection Manager to PCoIP Agent

This check looks for traffic between the PCoIP Connection Manager and a PCoIP Agent on TLS port 60443.

To verify PCoIP Connection Manager to agent collectivity:

1. On the PCoIP agent machine, use ssldump or Wireshark to capture packets from the PCoIP Connection Manager on TLS port 60443.
2. From the client, connect to the PCoIP Connection Manager.
3. Try to authenticate and establish a session.
4. Select a resource and connect.
5. Verify from ssldump or Wireshark output that the PCoIP agent is receiving data from the PCoIP Connection Manager.

Connectivity from PCoIP Client to PCoIP Security Gateway


This check looks for traffic between the PCoIP client and the PCoIP Security Gateway on TLS port 4172.

To verify that the PCoIP Security Gateway server is receiving session initiation data from the PCoIP client:

1. On the server hosting the PCoIP Security Gateway, start ssldump:

```
sudo ssldump -i <interface> host [client-ip-address] and port 4172
```

2. From the client, connect to the PCoIP Connection Manager.
3. Try to authenticate and establish a session.
4. Select a resource and connect.
5. Verify from ssldump output that the PCoIP Security Gateway is receiving data from the client.

 **Note: Firewall must allow traffic on UDP:4172**

If the firewall is configured to enable TCP traffic over port 4172 but not UDP traffic, then the ssldump output will show packets but you won't be able to establish a PCoIP session.

Connectivity (UDP) to PCoIP Security Gateway from PCoIP Client

This check looks for UDP traffic between the PCoIP Security Gateway and the PCoIP client on TLS port 4172.

To verify that the PCoIP Security Gateway is receiving UDP traffic from the PCoIP client:

1. On the server hosting the PCoIP Security Gateway, start tcpdump:

```
sudo tcpdump -i <interface> host [client-ip-address] and -n udp port 4172
```

2. From the client, connect to the PCoIP Connection Manager.
3. Try to authenticate and establish a session.
4. Select a resource and connect.
5. Verify from ssldump output that the PCoIP Security Gateway is receiving data from the client.

Connectivity (UDP) to PCoIP Agent from PCoIP Client

This check looks for UDP traffic between the PCoIP Security Gateway and the PCoIP client on TLS port 4172.

To verify that the PCoIP server is receiving UDP traffic from the PCoIP client:

1. On the server hosting the PCoIP server, start tcpdump:

```
sudo tcpdump -i <interface> host [server-ip-address] and -n udp port 4172
```

2. From the client, connect to the PCoIP Connection Manager.
3. Try to authenticate and establish a session.
4. Select a resource and connect.
5. Verify from ssldump output that the PCoIP server is receiving data from the client.

Verifying PCoIP Agent Availability

Ensure your DNS is configured correctly, then verify you can establish and maintain a connection to the agent.

For each virtual desktop host in your deployment or RDS farm, verify that you can establish TLS connections from the server hosting the PCoIP Connection Manager to the PCoIP agent listening on ports 4172 and 60443:

```
openssl s_client -connect <host-ip-address>:4172
openssl s_client -connect <host-ip-address>:60443
```

Verifying Connection Broker Availability

If you are using a connection broker and the firewall is configured correctly, then verify you can establish a TLS connection from the server hosting the PCoIP Connection Manager to the connection broker listening on port 443:

```
openssl s_client -connect <broker-ip-address>:443
```

Verifying PCoIP Connection Manager and Security Gateway Status

To verify the PCoIP Connection Manager and/or the PCoIP Security Gateway is running, SSH into the host machine.

- List the running services in Docker: to verify that the `pcoipcm_cm` and/or `pcoipcm_sg` services are running:

```
docker service ls
```

You should see the `pcoipcm_cm` and/or `pcoipcm_sg` service in the response, similar to the following example:

ID	NAME	MODE	REPLICAS	PORTS
34iefjidf	pcoipcm_cm	replicated	1/1	
	cloudaccessmanager.azurecr.io/pcoip-cm:5-release			
54ibvbdt	pcoipcm_sg	replicated	1/1	
	cloudaccessmanager.azurecr.io/sg:3-release			

Verifying PCoIP Connection Manager Web Application Status:

1. Establish a TLS connection using the `openssl s_client` command:

```
openssl s_client -connect 127.0.0.1:443
```

2. When the SSL connection is established, copy and paste the following text to issue a dummy HTTP POST command:

```
POST /pcoip-broker/xml HTTP/1.1
Host: localhost
Content-type: text/xml; charset=UTF-8
Content-Length: 39
<?xml version="1.0" encoding="UTF-8"?>
```


- If the PCoIP Connection Manager is operational, you will receive an XML response containing an `<error-resp>` element.

- If the PCoIP Connection Manager **is not** operational, check the logs of the PCoIP Connection Manager and PCoIP Security Gateway containers:

```
docker logs <CONTAINER_ID>
```

PCoIP Client cannot launch a session to the Agent with Horizon Broker

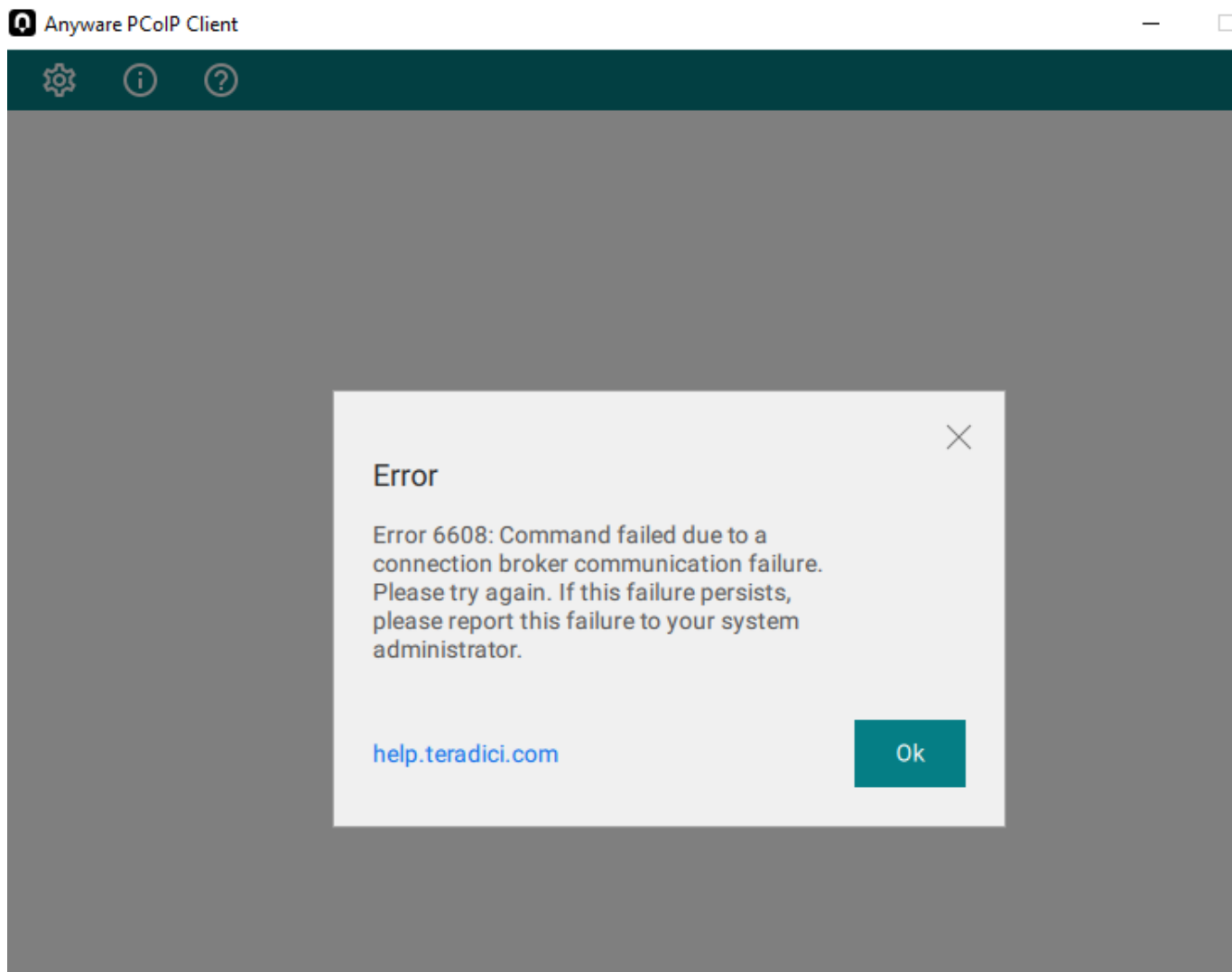
In this scenario, ensure that the Horizon Agent and the PCoIP Agent are on the same machine. There are some cases where in you could come across some errors. Some of those situations are as follows:

 **Windows Logs for PcoIP Agent and Horizon Broker can be found in c : \ProgramData\VMware\VDM\logs .**

Case 1:

1. Navigate to **Servers** and click **Connection servers**.
2. Select the Connection Server and check the **Use secure tunnel connection to machine** checkbox and click **Use Blast Secure Gateway for all Blast connections to machine**.

This results in a following error:



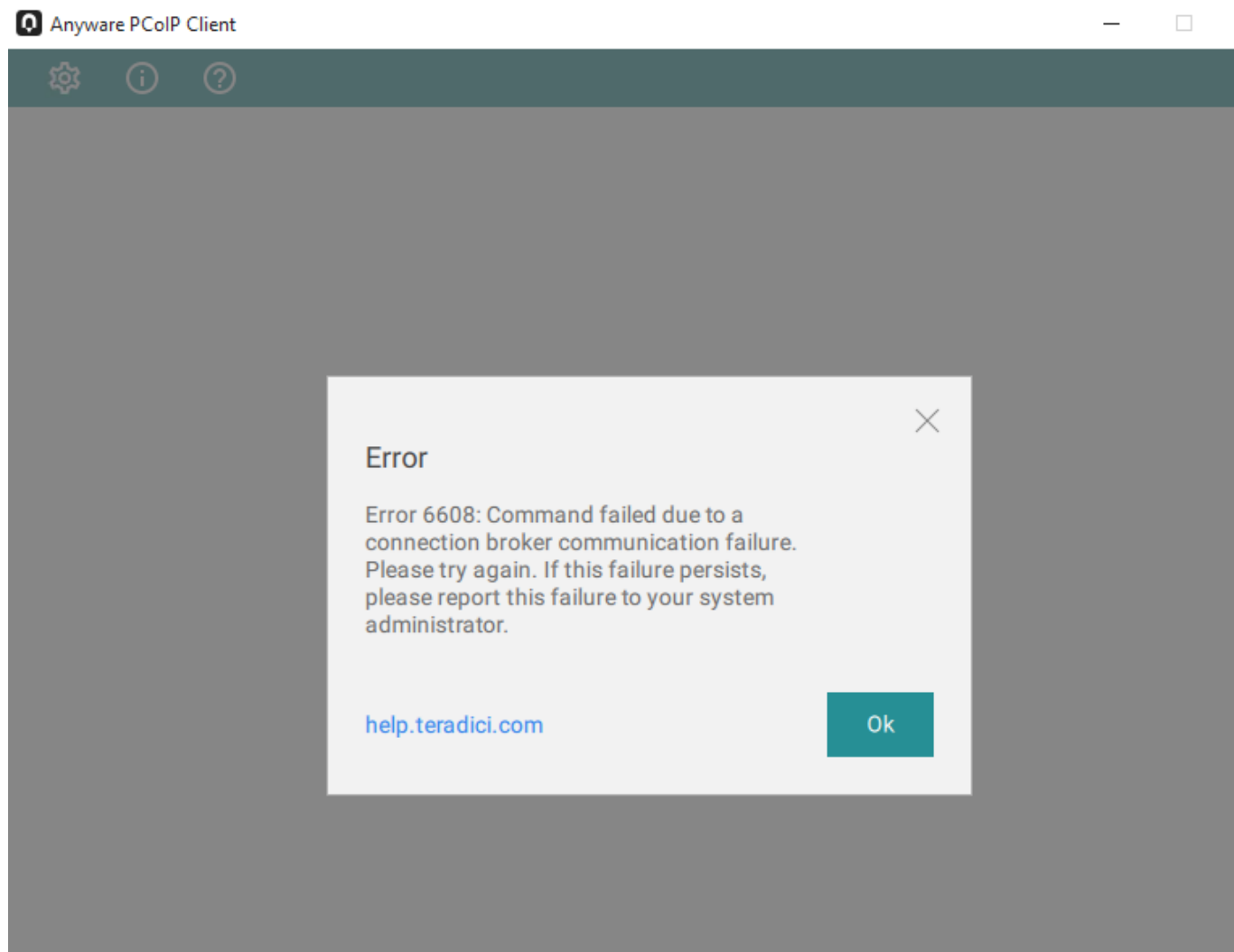
Logs:

```
2023-05-09T16:01:42.745Z 89c2df80-d0b0-103b-8473-000000000000 s=0002 c=0011
ERROR [app=MCM]baseclient.ClientRequest.<anonymous>: socket hang up
2023-05-09T16:01:44.747Z 89c2df80-d0b0-103b-8473-000000000000 s=0002 c=0011
WARN [app=MCM]baseclient.Timeout._onTimeout: Retry attempt #1 of 3. Reason:
timeout
2023-05-09T16:02:44.756Z 89c2df80-d0b0-103b-8473-000000000000 s=0002 c=0011
ERROR [app=MCM]baseclient.ClientRequest.<anonymous>: socket hang up
ERROR [app=MCM]baseclient.retry: Failed to communicate with Broker [https://
10.128.24.7:443] because of timeout.
  <err-detail>Timeout to communicate with Broker</err-detail>  </error-
resp> </pcoip-client>
2023-05-09T16:04:56.788Z 89c2df80-d0b0-103b-8473-000000000000 s=0002 c=0011
ERROR [app=MCM]baseclient.ClientRequest.<anonymous>: socket hang up
```

Case 2:

1. Navigate to **Servers** and click **Connection servers**.
2. Select the Connection Server and check the **Use secure tunnel connection to machine** checkbox and click **Use Blast Secure Gateway for only HTML Access connections to machine**.

This results in a following error:



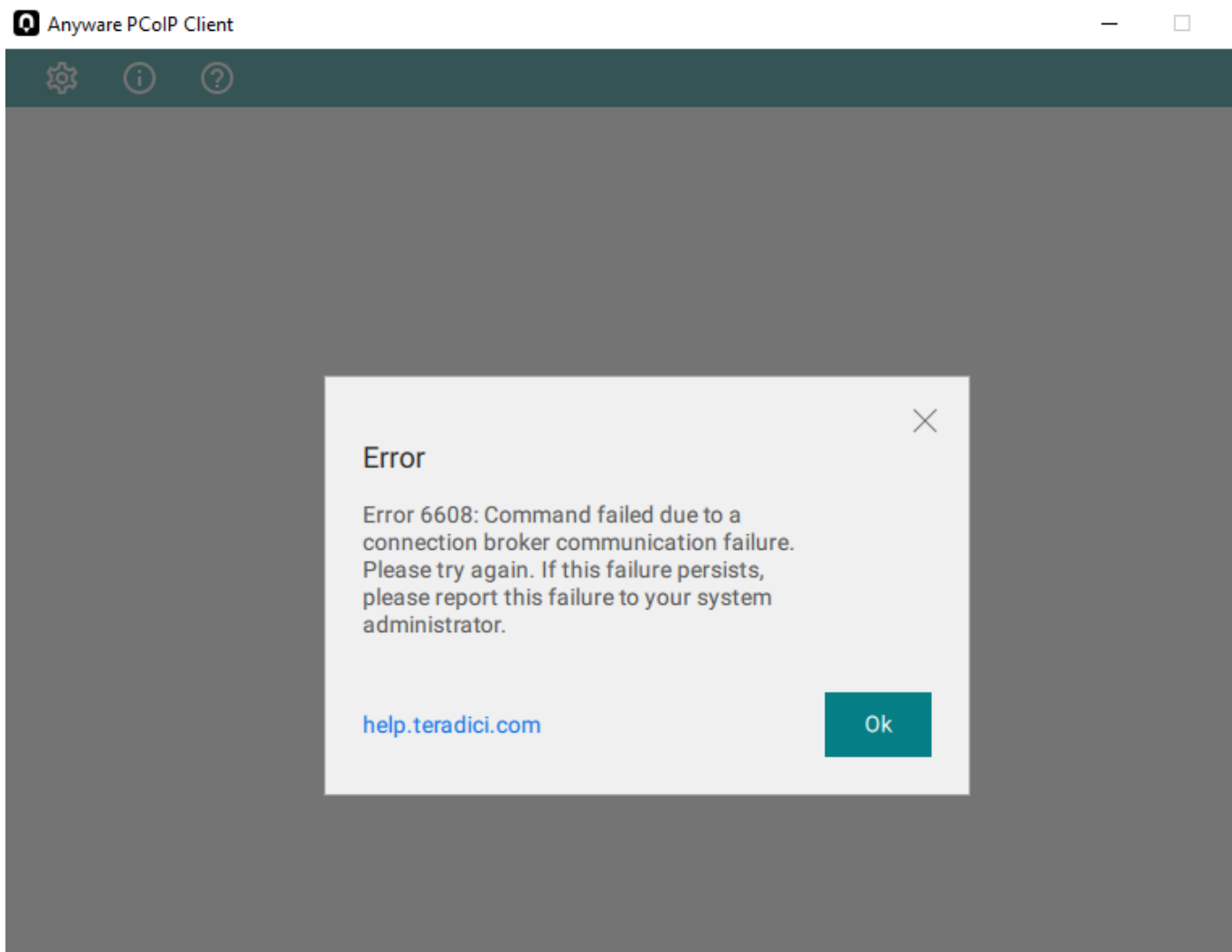
Logs:

```
2023-05-09T16:12:48.450Z 13bf7800-d0b2-103b-9dc1-000000000000 s=0003 c=0016
ERROR [app=MCM]baseclient.ClientRequest.<anonymous>: socket hang up
2023-05-09T16:12:50.450Z 13bf7800-d0b2-103b-9dc1-000000000000 s=0003 c=0016
WARN [app=MCM]baseclient.Timeout._onTimeout: Retry attempt #1 of 3. Reason:
timeout
2023-05-09T16:16:02.475Z 13bf7800-d0b2-103b-9dc1-000000000000 s=0003 c=0016
INFO [app=MCM]brokerprotocol.pbpErrorHandler: Sending response to client
(error-resp): <?xml version="1.0" encoding="UTF-8"?> <pcoip-client
version="2.1"> <error-resp <result> <result-id> <err-
detail>Timeout to communicate with Broker</err-detail> </error-resp> </
pcoip-client>
2023-05-09T16:16:02.478Z 13bf7800-d0b2-103b-9dc1-000000000000 s=0003 c=0016
ERROR [app=MCM]baseclient.ClientRequest.<anonymous>: socket hang up
```

Other Setting 1:

1. Navigate to **Settings** and Click **Global Settings**.
2. Select **Send Domain List** and uncheck **Hide domain list in client user interface**.

This results in a following error:



Logs:

```

<screen>          <name>windows-password</name>          <params>
<param>          <name>domain</name>          <values>
<value>*DefaultDomain*</value>          </values>          </
param>          </pa
rams>          </screen>          </authentication>          </configuration> </broker>
//Here in the above log , while authentication it's not the correct domain,
it's sending a default domain which led to Authentication failure
//broker doesn't send us the correct domain name

.....

2023-05-09T16:22:52.292Z 9e54a700-d0b3-103b-b725-000000000000 s=0005 c=0026
INFO
[app=MCM]horizonauthenticateresponse.HorizonAuthenticateResponse.handleErrorResp
Horizon authentication failed with error code AUTHENTICATION_FAILED, e
rror message Authentication failure and user message Authentication can not
proceed (Domain name is invalid).
2023-05-09T16:22:52.293Z 9e54a700-d0b3-103b-b725-000000000000 s=0005 c=0026
INFO [app=MCM]requestprocessor.processRequest: Sending response to client
(authenticate-resp): <?xml version="1.0" encoding="UTF-8"?> <pcoip-client
version="2.1"
> <authenticate-resp method="password"> <result> <result-
id>AUTH_FAILED_UNKNOWN_USERNAME_OR_PASSWORD</result-id> <result-
str>Authentication failure. Authentication can not proceed (Domain name is
invalid).</result-str>
</result> </authenticate-resp> </pcoip-client>

```

Other Setting 2:

1. Navigate to **Settings** and Click **Global Settings**.
2. Select **Send Domain List** and check **Hide domain list in client user interface**.

This settings run successfully.

Troubleshooting Certificate Errors

Error messages may be caused by different issues

Errors discussed here might not be caused by certificate problems.

Error messages

Failed to get broker / agent response due to: read ECONNRESET

If the broker or agent certificate key bit length is less than 2048, the PCoIP Connection Manager may not be able to establish a connection. Ensure that all components are using certificates of 2048 bits or more.

Troubleshooting Error Messages

Some common PCoIP client error messages and their possible causes are listed here.

Error occurred while communicating with broker

Possible cause	Resolution
The connection broker may be down or unreachable	Ensure the broker server is up and the broker service is running Ensure the broker server is resolvable by DNS

Timeout occurred while communicating with broker

Possible cause	Resolution
The port which the connection broker listens on may be blocked	Ensure the port which the broker server listen on is not blocked by firewall Ensure the broker server is functional

Error occurred while communicating with agent

Possible cause	Resolution
The PCoIP agent may be down or or unreachable.	Ensure the host is up and the agent service is running. Ensure the host is resolvable by DNS

Timeout occurred while communicating with agent

Possible causes:

Possible cause	Resolution
The port which the PCoIP agent listen on may be blocked.	Ensure the port of the PCoIP agent listen on is not blocked by firewall

PCoIP Connection Manager and Security Gateway Log Files

Each PCoIP component logs its activities and stores the log files locally. Troubleshooting behavior problems usually begins with an examination of PCoIP log files for error conditions or other system health indicators.

All PCoIP components use an identical, session-specific ID in their respective log files, allowing you to separate individual sessions and aggregate messages from multiple components within a session. The session ID is a 36-character hexadecimal string.

Log Maintenance

Logs and log rotation for both the PCoIP Connection Manager and PCoIP Security Gateway are managed automatically by Docker.

Sensitive Information in Logs

Sensitive information such as passwords, session cookies, and other session data that can potentially be used to gain unauthorized access is either obscured or not logged. Non-sensitive, unique session identifiers such as user names and IP addresses are logged as these often help with troubleshooting.

Log File Locations

Docker stores logs from its containers in `/var/lib/docker/containers`. You can check logs based on the CONTAINER_ID for the PCoIP Connection Manager and PCoIP Security Gateway.

If the PCoIP Connection Manager and PCoIP Security Gateway is running, you can also use the following command to check logs:

```
docker logs <MSG_CONTAINER_ID>
```


...where `<MSG_CONTAINER_ID>` is the container ID for the PCoIP Connection Manager and PCoIP Security Gateway.

Log Verbosity

PCoIP logs can capture log messages at several different verbosity levels, ranging from highly verbose informational messages to error-only reporting.

We recommend using the default verbosity log level in production deployments. When troubleshooting a problem, we might recommend changing the log level for specific components to obtain more information from parts of the system.

Security Gateway log levels cannot be changed

The log levels for the Security Gateway are not configurable.

Note: Increasing verbosity will reduce history

Increasing log verbosity will generate more and larger log files, which will then reach the system limits and be aged out more quickly.

Changing the PCoIP Connection Manager Log Level

The available log levels for the PCoIP Connection Manager, from most verbose to least verbose, are:

- debug
- info
- warn
- error

To configure the log level of the PCoIP Connection Manager:

1. Open `/opt/teradici/pcoipcm_data/docker-compose.yaml` in a text editor.

2. Add or modify the `LOG_LEVEL` value as an environment variable under the Connection Manager service:

```
LOG_LEVEL = <log level value>
```

Where `<log level value>` is one of `debug`, `info`, `warn`, or `error`.

3. Use the `config` command to apply changes:

```
pcoip-cmsg-setup config --compose-file /opt/teradici/pcoipcm_data/docker-compose.yaml
```

Contacting Support

If you encounter any problems installing, configuring, or running the PCoIP Connection Manager and PCoIP Security Gateway, you can create a [support ticket](#) with HP.

Before creating a ticket, be prepared with the following:

- A detailed description of the problem
- Your PCoIP Connection Manager and PCoIP Security Gateway version number. You can find this by opening a console window and running the command:

```
pcoip-cmsg-setup --version
```

- A support bundle, which contains log files and other diagnostic information we can use to help solve the problem. See [Generating a Support Bundle](#) for more information.

The HP Community Forum

The PCoIP Community Forum enables 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 HP PCoIP Technical Support Service team. HP staff are heavily involved in the forums.

To visit the HP community, go to the [HP Knowledge Center](#).

Generating a Support Bundle

HP may request a support bundle from your system in order to troubleshoot and diagnose PCoIP issues. The support file is an archive containing logs and other diagnostic data that can help support diagnose your problem.

To generate the support bundle:

1. Open a console window
2. Run the following command:

```
pcoip-cmsg-setup diagnose --support-bundle
```