

Teradici PCoIP Connection Manager and Security Gateway

The *PCoIP Connection Manager* and the *PCoIP Security Gateway* are components of Teradici Cloud Access Software, and are deployed together as a set. Multiple instances of the Connection Manager and Security Gateway can be deployed to handle mixed LAN and WAN access points or for scaling large systems.

Components in this release

The PCoIP Connection Manager and Security Gateway 22.01 is a combined release containing:

- PCoIP Connection Manager 22.01
- PCoIP Security Gateway 21.01.2

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.

Note

The PCoIP Security Gateway is not required for LAN access.

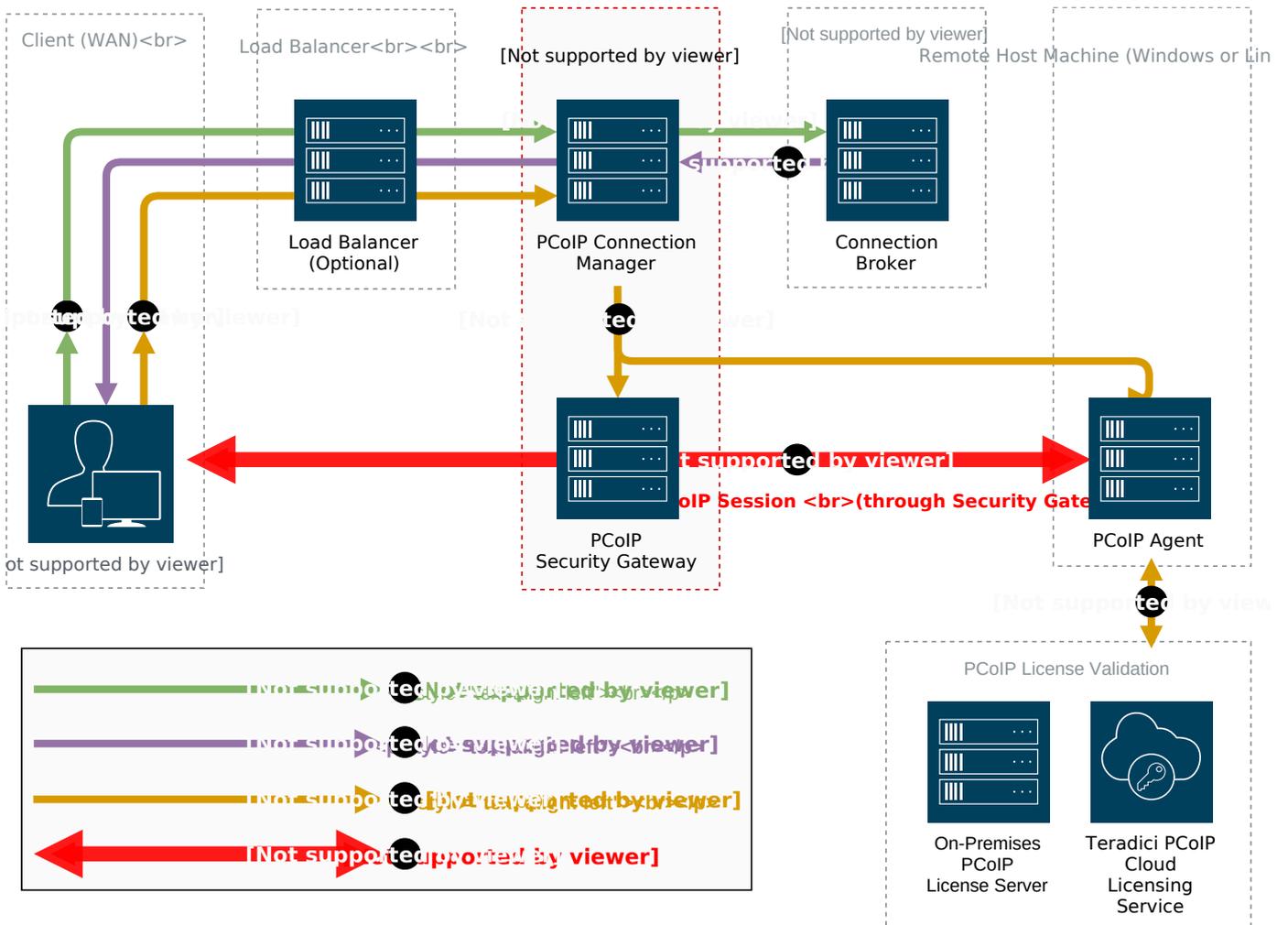
What's New in this Release?

Security Updates

Updates have been made in this release to enhance the security of the PCoIP Connection Manager and Security Gateway.

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, Teradici strongly recommends 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 the 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, leaving the Security Gateway enabled. The Connection Manager will handle PCoIP Connection establishment and the Security Gateway will secure the PCoIP session across the public internet.
- **Your desktops are on both a LAN and WAN:** Teradici recommends installing at least two groups of connection managers; one for internal access with the PCoIP Security Gateway disabled, and one for external access with the PCoIP Security Gateway 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

Coming soon

BETA

System Requirements

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

- 2 or more CPUs or vCPUs at 2.5 GHz or higher
- 4 GB of RAM
- 4 GB of swap space
- 4 GB of free disk space
- RHEL and CentOS 8 Stream. CentOS 7 is no longer supported.

The PCoIP Connection Manager and the PCoIP Security Gateway do not support IPv6.

If the connection broker is configured to identify resources by host name, then DNS must be available and configured so that the host names are resolvable from the server hosting the PCoIP Connection Manager and from the machines from which PCoIP clients will connect.

Install Dependencies for Offline Installation

PCoIP Connection Manager and PCoIP Security Gateway requires the following packages be installed and up to date prior to installing offline:

- gcc
- redhat-lsb
- java-1.8.0-openjdk.x86_64
- boost169
- authbind 2.1.2
- epel-release

These packages must be present on the remote workstation if an internet connection is not available. If an internet connection is available, the installer will obtain these packages automatically.

For information on performing an offline installation of the PCoIP Connection Manager and PCoIP Security Gateway, see [Installing PCoIP Connection Manager and PCoIP Security Gateway - Offline](#).

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 250 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 `/etc/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.

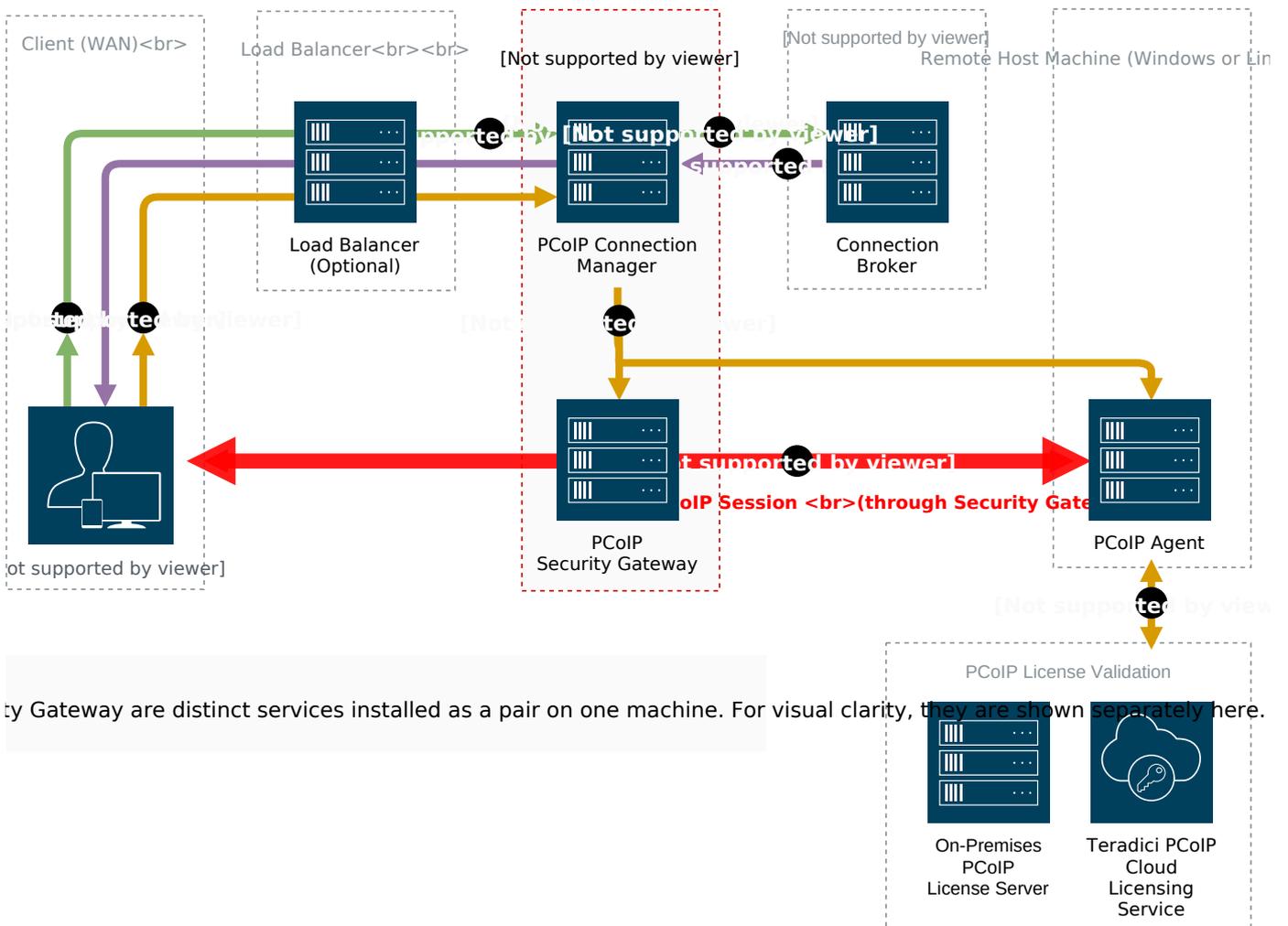
1100 Mbit/s is approximately the maximum bandwidth that can be achieved. Additional gains may be possible with larger sizing.

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.



PCoIP 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 `ExternalRoutableIP` configuration value from its paired PCoIP Security Gateway and passes it to the client. After the session is established, the client uses the provided IP address to communicate directly with the PCoIP Security Gateway.

Important: ExternalRoutableIP must point to the PCoIP Security Gateway

If the `ExternalRoutableIP` setting 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

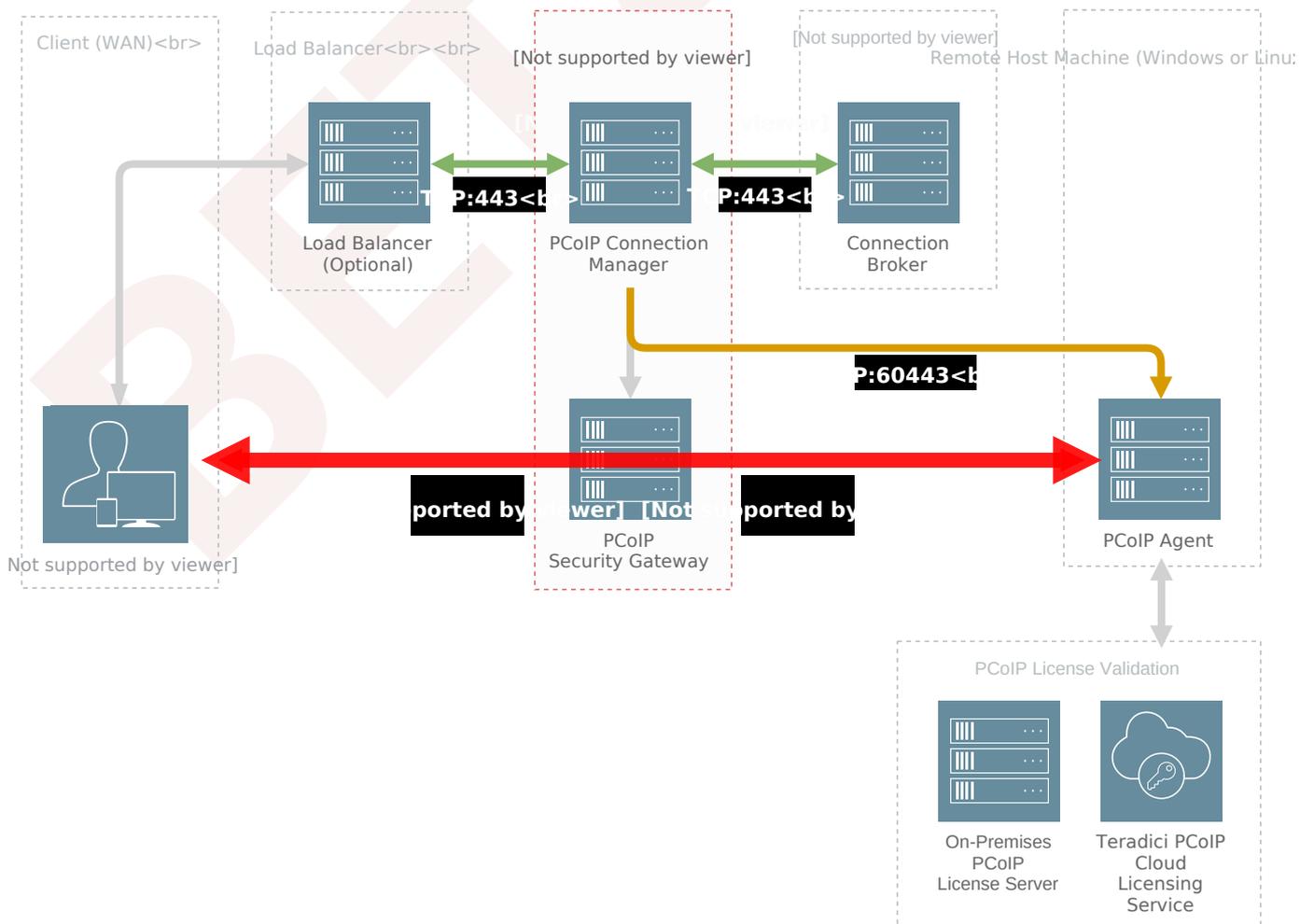
The machine with the PCoIP Connection Manager and Security Gateway on it must have a public IP address.

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.

Configuring Firewalls



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

Inbound Connections

Source	Port	Destination	Port	Description
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

By default, only SSH service is permitted in RHEL/CentOS 7. Use these commands to open the required ports for incoming traffic:

```
sudo iptables -I INPUT 1 -p tcp --dport 443 -j ACCEPT
sudo iptables -I INPUT 1 -p tcp --dport 4172 -j ACCEPT
sudo iptables -I INPUT 1 -p udp --dport 4172 -j ACCEPT
sudo service iptables save
```

Outbound Connections

If you also limit outbound connections, ensure that the following ports are open for outbound connections:

Port	Purpose
443 TCP	Used by the PCoIP Connection Manager to connect to third-party brokers. This is the default port for connections to third-party brokers. It is possible to use a different port by configuring it in the <code>ConnectionManager.conf</code> file.
60443 TCP	Used by the PCoIP Connection Manager to launch sessions on PCoIP Agents
4172 TCP/UDP	Used by the PCoIP Security Gateway to relay PCoIP session traffic from clients to PCoIP Agents

RHEL/CentOS permits 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. Teradici recommends that DHCP, DNS, and NTP are active for PCoIP Connection Manager operation.

Installing PCoIP Connection Manager and PCoIP Security Gateway

The PCoIP Connection Manager and PCoIP Security Gateway are bundled together in one package, available [from Teradici](#). The installation package includes:

- A setup script to install the RPMs and additional prerequisites.
- An RPM containing the PCoIP Connection Manager and PCoIP Security Gateway.
- An RPM containing third-party dependencies.

The PCoIP Connection Manager and the PCoIP Security Gateway must be installed together on a RHEL or CentOS 8 server with a single network interface.

Before You Begin

Before you proceed with installation, note the following:

- The PCoIP Connection Manager and the PCoIP Security Gateway do not support IPv6.
- 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 PCoIP client machines
- **Teradici highly recommends that you use NTP** to synchronize the time between components such as the PCoIP Connection Manager, PCoIP Security Gateway, PCoIP License Server, agents, and clients.

This enables logs to record a synchronized time across servers and sessions, which is extremely helpful in debugging and troubleshooting.

- You must have sudo (root) privileges to install the software.
- A text editor is required to configure the Connection Manager and Security Gateway. Any text editor will work.

Prepare the System Environment

First, verify that the following system environment requirements are met on your RHEL/CentOS 7 server:

- Your system meets or exceeds the [system requirements](#)
- Networking is configured to start on boot.
- NTP is enabled.
- Port 443 is available during installation.

Important: Uninstall the httpd service

Teradici highly recommends that you uninstall the `httpd` service (or any service or process that is bound to port 443) before installing the PCoIP Connection Manager. If Port 443 is in use by another service, PCoIP clients may not be able to connect to the PCoIP Connection Manager.

Install PCoIP Connection Manager and PCoIP Security Gateway

1. Download the PCoIP Connection Manager and Security Gateway package from [Teradici](#) and copy it into a temporary working directory.
2. Extract the archive:

```
unzip pcoip-cmsg_<version number>_el8.zip
```

Note: Included files

The `pcoip-cmsg__el8.zip` contains three files:

- `cm_setup.sh`
- `cm_sg-22.01.0-<build number>.x86_64.rpm`
- `cm_third_party_dependencies-22.01.0-<build number>.x86_64.rpm`

3. Run the installation script:

```
sudo sh ./cm_setup.sh
```

You can also navigate into the folder where the files have been extracted and run `cm_setup.sh` directly.

Setup script sets the system default to Java 8

The setup script will set the system default Java version to Java 8.

The `cm_setup.sh` script prepares the environment by installing compatible versions of OpenSSL, OpenJDK 8 (Runtime Environment), and Tomcat 8 with the Tomcat supporting libraries; then, it installs the RPM containing both the PCoIP Connection Manager and the PCoIP Security Gateway.

Some software is built from source during the setup procedure. To see exactly what the script does, open `cm_setup.sh` in a text editor and review the script content. You can customize this setup script for your system if required.

For a complete list of files and directories created during installation and operation of the PCoIP Connection Manager and PCoIP Security Gateway, see [PCoIP Connection Manager and PCoIP Security Gateway RPM Package Contents](#).

Important: Third-party dependencies

Currently-supported versions of included third-party components are listed in `/opt/Teradici/thirdparty/README`. Changing the version of any of these third-party components later might cause compatibility issues with other components.

Installing the PCoIP Connection Manager and PCoIP Security Gateway - Offline

If there is no internet connection in the target machine where PCoIP Connection Manager and PCoIP Security Gateway is going to be installed, you must create a temporary machine to download the PCoIP Connection Manager and PCoIP Security Gateway dependencies and then transfer them to the target machine. For information on these package dependencies, see [System Requirements](#).

Before You Begin

Before you proceed with installation, note the following:

- The PCoIP Connection Manager and the PCoIP Security Gateway do not support IPv6.
- 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 PCoIP client machines

Off-line Installation

The following steps outline how to install the PCoIP Connection Manager and PCoIP Security Gateway in an off-line environment:

1. Create a disposable internet-connected RHEL or CentOS 8 machine for downloading and inspecting packages. This machine should be identical to the PCoIP Connection Manager machine.
2. SSH into this new temporary machine. As the PCoIP Connection Manager and PCoIP Security Gateway now supports CentOS 8, Teradici recommends using `dnf` instead of `yum` for terminal commands.
3. Run the following command:

```
sudo dnf install -y wget
```

4. Create new directories to store downloaded package manager files:

Switch to root user: sudo -i

```
mkdir /root/packages/  
cd /root/packages/  
mkdir /root/packages/cm_sg  
mkdir /root/packages/updates  
mkdir /root/packages/authbind  
mkdir /root/packages/boost169
```

5. Download the required updates to the new packages/updates folder:

```
dnf update -y --downloadonly --downloadaddir=/root/packages/updates
```

6. Download the dependencies that are available in the package manager:

```
dnf download --resolve --downloadonly --arch x86_64 annobin unzip wget gcc  
gcc-c++ redhat-lsb java-1.8.0-openjdk.x86_64 expat --downloadaddir=/root/  
packages
```

7. Download authbind using the following command:

```
curl --output /root/packages/authbind/authbind_2.1.2.tar.gz http://  
ftp.debian.org/debian/pool/main/a/authbind/authbind_2.1.2.tar.gz
```

8. Download the Boost169 tar file using the following command:

```
wget -O /root/packages/boost169/boost_1_69_0.tar.gz http://sourceforge.net/  
projects/boost/files/boost/1.69.0/boost_1_69_0.tar.gz
```

9. Download the PCoIP Connection Manager and PCoIP Security Gateway package from [Teradici](#) and transfer it to the off-line machine using SCP. Move files cm_* into the folder ~/packages/cm_sg.

- a. Transfer the **packages** directory to the target machine through any method, for example via SCP or USB drive for on-premise machines.

- b. In the target machine, navigate to the **packages** folder and run the following commands for installing all dependencies: Switch to root user: `sudo -i`

```
# install updates and dependencies
cd /root/packages/updates
dnf install --disablerepo="*" -y *.rpm
cd /root/packages
dnf install --disablerepo="*" -y *.rpm

# install boost169
cd /root/packages/boost169
gunzip boost_1_69_0.tar.gz
tar -xvf boost_1_69_0.tar
cd boost_1_69_0

./bootstrap.sh --with-libraries=all --with-toolset=gcc

./b2 toolset=gcc --prefix=/usr/local
# it might take more than 15 min to compile

./b2 install --with=all

cp /usr/local/lib/lib* /usr/lib64

# install authbind
cd /root/packages/authbind
tar -xvf authbind_2.1.2.tar.gz
cd authbind
make
make install

touch /etc/authbind/byport/443
chmod 500 /etc/authbind/byport/443

cd /root/packages/cm_sg
# If you have pcoip_cmsg_<current_verson>.zip, run unzip command
unzip -j pcoip_cmsg_<current_version>.zip

chmod 550 /root/packages/cm_sg/cm_setup.sh
```

- c. Refer to steps 2 and 3 from the [Install PCoIP Connection Manager and PCoIP Security Gateway](#) section to complete the installation.

Updating the PCoIP Connection Manager and PCoIP Security Gateway

The PCoIP Connection Manager and PCoIP Security Gateway cannot be updated in place. To move to a new version, you must create a new machine, install and configure the software on it, then stop and destroy the old machine.

The method used to update the software differs by environment; follow the instructions for:

- [On-Premises PCoIP Connection Manager with an enabled PCoIP Security Gateway](#)
- [On-Premises PCoIP Connection Manager with a disabled PCoIP Security Gateway](#)
- [Public Cloud PCoIP Connection Manager and PCoIP Security Gateway](#)

Updating On-Premises PCoIP Connection Managers

On-premises systems can be deployed with or without the PCoIP Security Gateway. Follow the appropriate procedure for your environment.

Updating On-Premises Installations With PCoIP Security Gateway

If your system is located on-premises and the PCoIP Security Gateway is enabled, use this procedure.

To upgrade an on-premises PCoIP Connection Manager with the PCoIP Security Gateway enabled:

1. Build a new RHEL or CentOS VM and [install the PCoIP Connection Manager and PCoIP Security Gateway software](#) on it.
2. Configure the new PCoIP Connection Manager and PCoIP Security Gateway to match the old PCoIP Connection Manager and PCoIP Security Gateway:
 - Recreate the `/etc/ConnectionManager.conf` file on the new machine with identical settings.

- Recreate the `/etc/SecurityGateway.conf` file on the new machine with identical settings.
3. Install your custom certificates on the new machine.
 - Install the custom certificate for the PCoIP Connection Manager.
 - Install the custom certificate for the PCoIP Security Gateway.
 4. Disconnect the *new* PCoIP Connection Manager and PCoIP Security Gateway from the network and configure the local IP address to match the existing PCoIP Connection Manager and PCoIP Security Gateway.
 5. Shut down the existing PCoIP Connection Manager and PCoIP Security Gateway.
 6. Connect the new PCoIP Connection Manager and PCoIP Security Gateway to the network using the IP of the legacy PCoIP Connection Manager and PCoIP Security Gateway.
 7. Test a connection directly to the PCoIP Connection Manager and PCoIP Security Gateway external IP.

**Important: Powering off the PCoIP Connection Manager and PCoIP Security Gateway**

When you power off the existing PCoIP Connection Manager and PCoIP Security Gateway, any PCoIP sessions that are active and using the security gateway will be dropped and will need to be re-established.

**Load Balancer**

If you have a load balancer in front of a group of PCoIP Connection Manager and PCoIP Security Gateway virtual machines, then you can reconfigure the load balancer to stop sending new connections to a PCoIP Connection Manager and PCoIP Security Gateway virtual machine.

Updating On-Premises Installations Without PCoIP Security Gateway

If your system is located on-premises and the PCoIP Security Gateway is *disabled*, use this procedure.

To upgrade an on-premises PCoIP Connection Manager with the PCoIP Security Gateway disabled:

1. Build a new RHEL or CentOS VM and [install the PCoIP Connection Manager and PCoIP Security Gateway software](#) on it.
2. Recreate the `/etc/ConnectionManager.conf` file on the new machine with identical settings.
3. Install your custom certificates on the new PCoIP Connection Manager.
4. Add the IP address of the new PCoIP Connection Manager and PCoIP Security Gateway to the load balancer or round robin DNS.
5. Remove the IP address of the legacy PCoIP Connection Manager and PCoIP Security Gateway from the load balancer or round robin DNS.

Updating Public Cloud PCoIP Connection Managers

If your system is located in the public cloud, use this procedure. Public cloud deployments will always have the PCoIP Security Gateway enabled.

To upgrade the PCoIP Connection Manager and PCoIP Security Gateway in the public cloud:

1. Build a new RHEL or CentOS VM and [install the PCoIP Connection Manager and PCoIP Security Gateway software](#) on it.
2. Assign a new external IP to the VM and install any custom certificates required.
3. Configure the new PCoIP Connection Manager and PCoIP Security Gateway to match the old PCoIP Connection Manager and PCoIP Security Gateway:
 - Recreate the `/etc/ConnectionManager.conf` file on the new machine with identical settings.
 - Recreate the `/etc/SecurityGateway.conf` file on the new machine with identical settings.
4. Establish a new connection directly to the external IP to test that the PCoIP Connection Manager and PCoIP Security Gateway is correctly configured.
5. Add the new PCoIP Connection Manager and PCoIP Security Gateway to the cloud load balancer.

6. Repeat this process for each PCoIP Connection Manager and PCoIP Security Gateway that is being replaced.
7. Remove the legacy PCoIP Connection Manager and PCoIP Security Gateway from the load balancer.

Uninstalling PCoIP Connection Manager and PCoIP Security Gateway

Uninstalling the PCoIP Connection Manager and PCoIP Security Gateway also deletes configuration files such as `/etc/ConnectionManager.conf`, `/etc/SecurityGateway.conf`, and the `/opt/Teradici/certs` directory.

If necessary, back up your files before uninstalling.

Important: Both services are uninstalled

You cannot uninstall the PCoIP Security Gateway component by itself. Uninstalling the PCoIP Security Gateway also uninstalls the PCoIP Connection Manager.

To uninstall the PCoIP Connection Manager, PCoIP Security Gateway, and third-party dependencies:

```
sudo dnf remove cm_sg cm_third_party_dependencies
```

Some file structures and symbolic links are not deleted. If you plan to install a new version, you don't need to delete them. You can manually delete them if necessary.

To delete directories and symlinks:

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

Configuring the PCoIP Connection Manager

Configuration settings for the PCoIP Connection manager are stored in `/etc/ConnectionManager.conf` as key/value pairs separated by an equals sign. One setting is described per line.

To configure a setting, open `/etc/ConnectionManager.conf` with a text editor and add or change the appropriate lines. For example, to set a PCoIP License Server address, you would add this line:

```
LicenseServerAddress = <license-server-address>:<port>/request
```

Restart the PCoIP Connection Manager to apply your changes.

Configuration file formats and values are not validated

Configuration file formats and values are not validated. Incorrect configurations can result in components that do not work properly. Ensure you make backups before making changes.

Configuration Settings

Parameter	Default	Description
LogLevel	INFO	The minimum severity level of the messages written to the log. Acceptable values in increasing order of severity are: TRACE, DEBUG, INFO, WARN, and ERROR. Only messages that are at or above the configured LogLevel severity are logged. For information on log files, see PCoIP Connection Manager and Security Gateway Log Files .
BrokerType		Type of broker of the PCoIP Connection Manager is using: <div style="background-color: #f0f0f0; padding: 2px; margin: 5px 0;"><code>BrokerType = PCoIP</code></div> This is the only available option.

Parameter	Default	Description
PcoipAddress		<p>IP Address or FQDN of the PCoIP broker that the PCoIP Connection Manager uses to authenticate users and obtain resource information; for example:</p> <pre>PcoipAddress = 123.45.67.89:443</pre> <p>Port 443 is the default address for this connection, but another port can be used by configuring the port number with this parameter.</p>
SecurityGatewayEnabled	false	<p>If set to true, the PCoIP Connection Manager uses the PCoIP Security Gateway to establish sessions and directs clients to connect to their sessions via the PCoIP Security Gateway. The PCoIP Security Gateway must be enabled and configured. If set to false, the PCoIP Connection Manager directs clients to connect directly to the agent hosting the selected resource.</p>
LicenseServerAddress		<p>A single PCoIP License Server address and port number. Use the format <code>http://<license-server-address>:<port>/request</code>. Cannot be more than 1024 ASCII characters. Do not use <code><</code>, <code>></code>, or <code>&</code>.</p> <p>To use Teradici Cloud Licensing, leave this unset.</p>
ContentLengthEnabled	false	<p>If set to true, the PCoIP Connection Manager always sets the <code>Content-Length: XX</code> in the HTTP response header. If set to false the PCoIP Connection Manager sends HTTP responses using chunked encoding.</p>
BrokerMaxRespWaitSeconds	20	<p>The time in seconds to wait for a response from the broker (other than for authenticate or allocate resource responses) before timing out.</p>
BrokerMaxAllocateWaitSeconds	60	<p>The time in seconds to wait for a response from the broker to an allocate resource request before timing out.</p>
BrokerMaxAuthenticationWaitSeconds	30	<p>The time in seconds to wait for a response from the broker to an authenticate request before timing out.</p>
AgentMaxRespWaitSeconds	160	<p>The time in seconds to wait for a response from the PCoIP agent before timing out.</p>

Parameter	Default	Description
AgentCertCheck	false	If set to true, the PCoIP Connection Manager validates the certificate presented by agents during resource allocation.
AgentCertMinKeyLength	1024	When AgentCertCheck is true, specifies the required minimum public key length of the certificate presented by the agent. Ignored when AgentCertCheck is false. The minimum allowable length is 1024.
BrokerCertCheck	false	If set to true, the PCoIP Connection Manager validates the certificate presented by the broker during authentication and resource retrieval.
BrokerCertMinKeyLength	1024	When BrokerCertCheck is true, specifies the required minimum public key length of the certificate presented by the broker. Ignored when BrokerCertCheck is false. The minimum allowable length is 1024.
ClientSSLCipherBlackList		Lists the TLS cipher suites to be removed from the default list of cipher suites used for establishing a TLS connection to the PCoIP client. The cipher suites are specified by their RFC names and are separated by a colon. See PCoIP Connection Manager Supported TLS Cipher Suites . For example, specifying the 'TLS_RSA_WITH_AES_256_CBC_SHA: TLS_RSA_WITH_AES_128_CBC_SHA' string as the black list removes the specified two cipher suites.
ServerSSLCipherBlackList		Lists the TLS cipher suites to be removed from the default list of cipher suites used for establishing a TLS connection to the connection broker and the PCoIP agent. The cipher suites are specified by their RFC names and are separated by a colon. See PCoIP Connection Manager Supported TLS Cipher Suites . For example, specifying the 'TLS_RSA_WITH_AES_256_CBC_SHA: TLS_RSA_WITH_AES_128_CBC_SHA' string as the black list removes the specified two cipher suites.

Parameter	Default	Description
-----------	---------	-------------

ControlChannelTLSEnabled	true	If set to true, the PCoIP Connection Manager uses TLS to establish a secure connection with the PCoIP Security Gateway to send control commands. Otherwise, the PCoIP Connection Manager sends control commands in plain text. If set to true, the PCoIP Security Gateway must also be configured to use TLS. For more information, see TCPControlLinuxExtCert .
--------------------------	------	--

Configuring the PCoIP Security Gateway

Configuration settings for the PCoIP Security Gateway are stored in `/etc/SecurityGateway.conf` as key/value pairs separated by an equals sign. One setting is described per line.

To configure a setting, open `/etc/SecurityGateway.conf` with a text editor and add or change the appropriate lines. For example, to set the PCoIP Security Gateway's external IP address, you would add this line:

```
ExternalRoutableIP = ip-address-reachable-by-clients
```

Restart the PCoIP Security Gateway to apply your changes.

Configuration file formats and values are not validated

Configuration file formats and values are not validated. Incorrect configurations can result in components that do not work properly. Ensure you make backups before making changes.

Important: The Security Gateway must be enabled by the Connection Manager

To use the PCoIP Security Gateway, it must be enabled in `/etc/ConnectionManager.conf`:

```
SecurityGatewayEnabled = true
```

In `/etc/SecurityGateway.conf`, you must set the IP address clients will use to reach the PCoIP Security Gateway:

```
ExternalRoutableIP = ip-address-reachable-by-clients
```

PCoIP Security Gateway Configuration Settings

The configuration files for the PCoIP Security Gateway are located at `/etc/SecurityGateway.conf`. To apply changes, restart the PCoIP Security Gateway first, then restart the PCoIP Connection Manager.

PCoIP Security Gateway Configuration Settings

Parameter	Default	Description
ExternalRoutableIP		The externally routable IP address of the PCoIP Security Gateway. This is typically set to the static IP address assigned to the PCoIP Connection Manager. Do not set this to a loopback address.
LogLevel	2	The minimum severity level of messages written to the log. Acceptable values in increasing order of severity are: 0 (TRACE), 1 (DEBUG), 2 (INFO), 3 (WARN), 4 (ERROR). Only messages that are at or above the configured LogLevel severity are logged. For information on log files, see PCoIP Connection Manager and Security Gateway Log Files .
LogPath	\$TMPDIR or /tmp	Location of PCoIP Security Gateway log files.
MaxConnections	5000	Maximum number of connections. ulimit -n on Linux needs to be set to slightly more than double this number.
SSLCertPath	/opt/Teradici/certs	Location of certificate files used by the PCoIP Security Gateway.
SSLCertType	0	0 = Use an external certificate. If not configured, then generate and use a self-signed certificate. 1 = Use an external certificate. 2 = Generate and use a self-signed certificate.
SSLLinuxExtCA	CMCertificateCA.pem	SSLLinuxExtCert certificate chain file name.
SSLLinuxExtCert	CMCertificate.pem	File name of the public certificate (in base64-encoded PEM format) used to secure communication with PCoIP clients.

Parameter	Default	Description
SSLLinuxExtCertPhrase		Passphrase of the private key specified by SSLLinuxExtPriv. We strongly advise against encrypting the private key since doing so requires the pass phrase to be specified here in plain text.
SSLLinuxExtPriv	CMCertificateKey.pem	File name of the SSLLinuxExtCert certificate private key (in base64-encoded PEM format).
SSLCipherBlackList		Lists the TLS cipher suites to be removed from the default list of cipher suites used for establishing a TLS connection to the PCoIP client, the PCoIP server, and the connection manager. The cipher suites are specified by their RFC names and are separated by a colon. See PCoIP Connection Manager Supported TLS Cipher Suites . For example, specifying the "TLS_RSA_WITH_AES_256_CBC_SHA: TLS_RSA_WITH_AES_128_CBC_SHA" string as the black list removes the specified two cipher suites.
TCPControlLinuxExtCA	CMCertificateCA.pem	TCPControlLinuxExtCert certificate chain file name.
TCPControlLinuxExtCert	CMCertificate.pem	File name of the public certificate (in base64-encoded PEM format) used to secure communication with the PCoIP Connection Manager.
TCPControlLinuxExtCertPhrase		Passphrase of the private key specified by TCPControlLinuxExtPriv. We strongly advise against encrypting the private key since doing so requires the passphrase to be specified here in plain text.
TCPControlLinuxExtPriv	CMCertificateKey.pem	File name of the TCPControlLinuxExtCert certificate private key (in base64-encoded PEM format).

 **Security gateway secures connections to control channel**

If the TCPControlLinuxExtCA , TCPControlLinuxExtCert, and TCPControlLinuxExtPriv settings are all configured, then the security gateway secures connections to its control channel with TLS. If one or more of these settings are not specified, then the security gateway accepts plain text connections to its control channel. The connection manager uses TLS by default when establishing a connection to the security gateway control channel. For more information, see [PCoIP Connection Manager Configuration Settings](#).

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.

Agent and Broker Certificate Validation

Enable validation of certificate files

For production deployments, Teradici strongly recommends enabling validation of certificate files presented by PColP agents and broker.

In *brokered* systems, Teradici recommends the following:

- Install certificate files signed by a trusted certificate authority (CA) onto the agents and broker.
- Ensure the intermediate or root certificate from the CA is installed in the PColP Connection Manager's keystore. See [Importing Certificates into the Keystore](#).

Enabling Certificate Validation

To Enable PColP Connection Manager agent and broker certificate validation:

1. Open `/etc/ConnectionManager.conf` in a text editor and set the following values:

```
AgentCertCheck = true  
BrokerCertCheck = true
```

2. Save and close the editor.
3. Restart the PColP Connection Manager to implement the change:

```
sudo service connection_manager restart
```

Configure the agents and broker to present certificate chain

Ensure the agents and the broker are configured to present the complete certificate chain to clients (namely, the PCoIP Connection Manager). If none of the certificate files in the chain are signed by an intermediate or root certificate in the PCoIP Connection Manager's keystore, certificate validation will fail.

Using the PCoIP Connection Manager Keystore

To validate the agent and broker certificates, the PCoIP Connection Manager uses the Java system default keystore. The exact location of the keystore will vary depending on your Java installation and system configuration; in the Java home directory, the keystore path is typically `/etc/ssl/certs/java/cacerts`.

Importing Certificates Into the Keystore

To import a certificate into the keystore:

1. On the PCoIP Connection Manager server, open a command prompt.
2. Start the Java keytool:

```
sudo keytool -importcert -trustcacerts -file <path-to-certificate> -keystore  
<path-to-keystore> -alias <arbitrary-alias>
```

3. When prompted, enter the keystore password.
4. If the keytool cannot establish trust of the specified certificate, it displays the properties of the certificate followed by a prompt. In this case, verify you are importing the correct certificate and ensure that the certificate's constraints enable it to be used for certificate verification:

```
BasicConstraints:[
```

```
...
```

```
CA:true
```

```
...
```

```
]

```

5. At the *Trust this certificate?* prompt enter **y** and press **Enter** to complete the import.
6. Verify you get a confirmation that the certificate was added to keystore.

Certificate files do not need to be added to the keystore

Certificate files that the PCoIP Connection Manager and the PCoIP Security Gateway present to clients do not need to be added to the keystore, namely, CMCertificate.pem.

Managing the Keystore

Change your default password

Teradici strongly recommends changing the default password and using a password that conforms to your organization's security policy. Java's default keystore password is 'changeit'.

To list the certificates in the keystore:

```
keytool -list -v -keystore <path-to-keystore>
```

To determine whether a particular certificate is already installed to the keystore, it may be easier to search by Subject:

```
keytool -list -v -keystore <path-to-keystore> | grep "^Owner"
```

To change the keystore password:

```
keytool -storepasswd -keystore <path-to-keystore>
```

To remove a certificate from the keystore:

```
keytool -delete -alias <alias> -keystore <path-to-keystore>
```

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 1,024-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.

Certificate files must be readable by the `teradici_components` group.

If you are ready to replace your default self-signed certificates with your own signed certificates, proceed to [Signed Services 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 Windows 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 them however you need to to remain compliant.

These examples use openssl

The following procedures use openssl to create and manage certificates. If you use another tool, adapt these instructions accordingly.

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

Distinguished Name Field	Description	Example
Organization Name	The full legal name of your organization. Cannot use < > ~ ! @ # \$ % ^ * / \ () ? . , &	Teradici 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 teradiciplatform.teradici.com then the CSR must have the common name teradiciplatform.teradici.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
private.key	Contains the certificate's private key.
certificate.pem	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.
CAcertificate.pem	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.

Installing Certificate Files

To install the newly-created certificate files, copy them into the configured certificate folder of the PCoIP Connection Manager machine and restart the services.

We will copy three files: the signed certificate without the chain (**certificate.pem**), the signed certificate *with* the chain (**CAcertificate.pem**), and the key file (**private.key**).

To install new certificate files:

1. On the PCoIP Connection Manager machine, open a command prompt.

2. Rename the existing certificate files, preserving them as backups:

```
mv /opt/Teradici/certs/CMCertificate.pem /opt/Teradici/certs/
CMCertificate.pem.backup
mv /opt/Teradici/certs/CMCertificateCA.pem /opt/Teradici/certs/
CMCertificateCA.pem.backup
mv /opt/Teradici/certs/CMCertificateKey.pem /opt/Teradici/certs/
CMCertificateKey.pem.backup
```

3. Copy the new certificate files. These commands assume you've created these files using the instructions above; if you haven't, the source filenames shown here may be different.

```
cp ~/certs/certificate.pem /opt/Teradici/certs/CMCertificate.pem
cp ~/certs/CAcertificate.pem /opt/Teradici/certs/CMCertificateCA.pem
cp ~/certs/private.key /opt/Teradici/certs/CMCertificateKey.pem
```

The resulting files are renamed to **CMCertificate.pem**, **CMCertificateCA.pem**, and **CMCertificateKey.pem**. Note that the **.key** file is renamed to **.pem** by this copy operation.

4. Restart PColP Connection Manager components:

```
service security_gateway restart
service connection_manager restart
```

5. Once both services are back up, check the PColP Connection Manager log file to ensure the PColP Connection Manager web service started:

```
less /var/log/Teradici/ConnectionManager/catalina.out
```

Look for these lines in the output:

```
INFO: Initializing ProtocolHandler ["http-apr-443"]
INFO: Starting ProtocolHandler ["http-apr-443"]
```

Also verify that there are no lines beginning with **SEVERE:** , as they may indicate that the certificate failed to load.

6. Check the most recent PColP Security Gateway log file to ensure the PColP Security Gateway service started. To do this, we'll go into the log directory, list all the files, and then use **less** to view the most recent:

```
cd /var/log/Teradici/SecurityGateway/  
ls -l  
less <the_most_recent_filename>
```

Protecting Certificate Files

Once your certificate files have been created and installed, follow these guidelines to protect them.

To maintain client communications security:

- Ensure only root and the `teradici_components` group can read private keys.
- Ensure all certificate files are read-only.

To protect certificate files:

1. Log in to the server as an administrator.
2. Open a command prompt and issue these commands:

```
chown root:teradici_components /opt/Teradici/certs/CMCertificateKey.pem  
chmod 440 /opt/Teradici/certs/CMCertificateKey.pem  
chmod -w /opt/Teradici/certs/CMCertificate.pem  
chmod -w /opt/Teradici/certs/CMCertificateCA.pem
```

Configuring Certificate Location and File Names

By default, certificate files are created in `/opt/Teradici/certs` during installation. This location and file names do not normally need to be changed.

If your organization's security policy requires it, you can change the location or file name of certificate files. The PCoIP Connection Manager and the PCoIP Security Gateway certificate files may be located in different directories.

Once you have installed the the certificates automatically you can run `restart_components.sh` to restart the system and complete the update.

Customizing PCoIP Connection Manager Certificate Information

The PCoIP Connection Manager's certificate configuration is in the Tomcat server config file, located in `/opt/Teradici/thirdparty/tomcat/conf/server.xml`. Set the certificate file paths with the following attributes of the `<Connector>` element in `server.xml`:

- `SSLCertificateFile`
- `SSLCertificateKeyFile`
- `SSLCACertificateFile`

Customizing PCoIP Security Gateway Certificate Information

The PCoIP Security Gateway's certificate configuration is in its own configuration file, located in `/etc/SecurityGateway.conf`. Set the certificate file paths with the following attributes:

- `SSLLinuxExtCA`
- `SSLLinuxExtCert`
- `SSLLinuxExtPriv`
- `TCPControlLinuxExtCA`
- `TCPControlLinuxExtCert`
- `TCPControlLinuxExtPriv`

Ensure all certificate conform to your security policy

Protect the certificate and ensure all certificate files you use conform to your organization's security policy, including file ownership and permissions.

Self-Signed Certificates for Testing

A self-signed certificate can be used for testing and evaluation, and is provided by a default installation.

When using the self-signed certificate, PCoIP clients will connect but will indicate that the connection is insecure. Note that this warning appears because the certificate is not trusted; the connection may actually be secure, if the system is secured by other means (for example, if the entire system is deployed on a secured network).

For production systems, Teradici *highly recommends* replacing the self-signed certificate files with your own certificates, signed by a trusted Certificate Authority (CA).



Regenerate the self-signed certificate if you change your host name

If you use the default self-signed certificate and you change the system host name, you must [regenerate the self-signed certificate](#).

About the Default Certificate Files

By default, both the PCoIP Connection Manager and PCoIP Security Gateway use the same key/certificate pair located in `/opt/Teradici/certs`.

- **CMCertificate.pem** contains the leaf certificate that the server presents to the client during the TLS handshake. This certificate contains the public key that the client uses to encrypt the symmetric key. Both the server and the client use this symmetric key for encryption and decryption in subsequent communications.

This certificate secures the following ports:

- TCP port 443 for the PCoIP Connection Manager.
- TCP port 4172 for the PCoIP Security Gateway.

This certificate is presented as follows:

- The PCoIP Connection Manager presents this certificate file to PCoIP clients.
 - The PCoIP Security Gateway presents this certificate file to PCoIP clients and to the PCoIP Connection Manager.
- **CMCertificateCA.pem** contains the full chain of certificate files that the server presents to the client during the TLS handshake. For the client to establish trust of the leaf certificate, one or both of the following must be true:
 - At least one of the certificate files in the chain must be in the client's trust store.

- The certificate of the certificate authority (CA) used to sign the last certificate in the chain must be in the client's trust store.

Regenerating the Self-Signed Certificate

If you need to regenerate the self-signed certificate, use the `make_certs.sh` utility script. Include the `--install` option to generate and install the certificates automatically:

```
sudo ~/opt/Teradici/Management/bin/make_certs.sh --install
```

If you use the `--install` option, files will be installed in the `/opt/Teradici/certs` directory and overwrite any existing files with the same names. If the `/opt/Teradici/certs` directory does not exist, the script will create it with the following properties:

- Ownership: `root`
- Group: `teradici_components`
- Access: Readable and browsable only by `root` and `teradici_components` group members.

Installed files have these properties:

- Ownership: `root`
- Group: `teradici_components`
- Access: Readable only by `root` and `teradici_components` group members.

Once you have installed the the certificates automatically you can run `restart_components.sh` to restart the system and complete the update.

Administering the PCoIP Connection Manager and PCoIP Security Gateway

Starting or Stopping the PCoIP Connection Manager

To start, stop, or restart the PCoIP Connection Manager:

```
sudo service connection_manager start|stop|restart
```

Restarting the PCoIP Connection Manager may take up to a minute to complete. Clients cannot establish PCoIP sessions through it until restart is complete.

Starting or Stopping the PCoIP Security Gateway

To start, stop, or restart the PCoIP Security Gateway:

```
sudo service security_gateway start|stop|restart
```

Enable the PCoIP Security Gateway

If you deploy Teradici Platform over a WAN, using a PCoIP Security Gateway is highly recommended.

To enable the PCoIP Security Gateway:

1. Open `/etc/ConnectionManager.conf` in a text editor and ensure the following line is present:

```
SecurityGatewayEnabled = true
```

2. Save the file and exit the editor.
3. Open `/etc/SecurityGateway.conf` in a text editor and set the following field:

```
ExternalRoutableIP = <ip address reachable by clients>
```

4. Save the file and exit the editor.
5. Open `/etc/security/limits.conf` in a text editor and ensure that the PCoIP Security Gateway's file descriptor limits are set to 11000 or higher.

The file descriptor limits look like this:

```
security_gateway soft nfile 11000
security_gateway hard nfile 11000
```

6. Restart the PCoIP Connection Manager:

```
service connection_manager restart
```

7. Restart the PCoIP Security Gateway:

```
service security_gateway restart
```

Disable the PCoIP Security Gateway

If your users are behind your firewall and will not access their desktops from the WAN, you do not need to use the PCoIP Security Gateway. You can optionally disable it using this procedure.

To disable the PCoIP Security Gateway:

1. Open `/etc/ConnectionManager.conf` in a text editor and set `SecurityGatewayEnabled` to `false`:

```
SecurityGatewayEnabled = false
```

2. Save the file and exit the editor.
3. Restart the PCoIP Connection Manager:

```
service connection_manager restart
```

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 Teradici'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 connection broker and license server addresses.

Edit `/etc/ConnectionManager.conf` and set the following fields:

```
BrokerType = PCoIP  
  
PcoipAddress = <broker-ip-address-or-hostname>  
  
LicenseServerAddress = `http://<license-server-address>:<port>/request`
```

PCoIP Connection Manager and Security Gateway RPM Package Contents

The following tables show the files installed by the RPM packages:

Files and Directories created during installation of PCoIP Connection Manager third-party dependencies

File/Directory	Description
/opt/Teradici/thirdparty/tomcat	Tomcat
/opt/Teradici/thirdparty/openssl	OpenSSL
/opt/Teradici/thirdparty/tcnative	Tomcat Native
/opt/Teradici/thirdparty/apr	Apache Portable Runtime
/opt/Teradici/thirdparty/README	Readme file with instructions for building and updating these libraries

Files and directories created during installation and operation of PCoIP Connection Manager and PCoIP Security Gateway

File/Directory	Description
/etc/ConnectionManager.conf	PCoIP Connection Manager configuration file.
/etc/init.d/connection_manager	PCoIP Connection Manager service control script.
/etc/SecurityGateway.conf	PCoIP Security Gateway configuration file.
/etc/init.d/security_gateway	PCoIP Security Gateway service control script.

File/Directory	Description
<code>/opt/Teradici/certs</code>	Directory containing certificate files used by the PCoIP Connection Manager and PCoIP Security Gateway.
<code>/opt/Teradici/ConnectionManager</code>	Directory containing PCoIP Connection Manager version information and symbolic link to log files.
<code>/opt/Teradici/SecurityGateway</code>	Directory containing PCoIP Security Gateway version information, binaries, and symbolic link to log files.
<code>/opt/Teradici/Management</code>	Directory containing component management utilities.
<code>/opt/Teradici/thirdparty/tomcat/conf/catalina.properties</code>	Tomcat configuration file tailored for the PCoIP Connection Manager. Original renamed to catalina.properties.timestamp.original.
<code>/opt/Teradici/thirdparty/tomcat/conf/logging.properties</code>	Tomcat logging configuration file tailored for the PCoIP Connection Manager. Original renamed to logging.properties.timestamp.original.
<code>/opt/Teradici/thirdparty/tomcat/conf/server.xml</code>	Tomcat server configuration file tailored for the PCoIP Connection Manager. Original renamed to server.xml.timestamp.original.
<code>/opt/Teradici/thirdparty/tomcat/conf/tomcat-users.xml</code>	Tomcat user configuration file tailored for the PCoIP Connection Manager. Original renamed to tomcat-users.xml.timestamp.original.
<code>/opt/Teradici/thirdparty/tomcat/webapps</code>	Tomcat web application directory containing the PCoIP Connection Manager web application. Original renamed to webapps.timestamp.original.
<code>/opt/Teradici/thirdparty/tomcat/webapps/info</code>	PCoIP Connection Manager status page web application binary archive and directory containing meta information.
<code>/opt/Teradici/thirdparty/tomcat/webapps/pcoip-broker</code>	PCoIP Connection Manager web application binary archive and directory containing meta information.
<code>/var/log/Teradici/ConnectionManager</code>	Directory containing PCoIP Connection Manager log files.

File/Directory	Description
/var/log/Teradici/SecurityGateway	Directory containing PCoIP Security Gateway log files.

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.

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_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA

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_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA

Blacklisting Cipher Suites

Both the PCoIP Connection Manager and PCoIP Security Gateway can be configured to accept subsets of the full cipher suite list. This is done by blacklisting unwanted suites via configuration settings and restarting the respective service.

Blacklisting Cipher Suites for PCoIP Client Connections

You can limit the cipher suites accepted for incoming PCoIP client connections by using the `ClientSSLCipherBlackList` setting to blacklist unwanted suites. For more information, see [PCoIP Connection Manager Configuration Settings](#).

Changing the `ClientSSLCipherBlackList` setting updates cipher suite list

Changing the `ClientSSLCipherBlackList` and then restarting the PCoIP Connection Manager service causes the `SSLCipherSuite` variable in `/opt/Teradici/thirdparty/tomcat/conf/server.xml` to be updated with the revised cipher suite list. Tomcat uses the ciphers specified in `server.xml` for all its inbound connections.

Blacklisting Cipher Suites for Connection Broker and PCoIP Agent Connections

You can limit the cipher suites accepted for communications with a connection broker or PCoIP agent by using the `ServerSSLCipherBlackList` setting to blacklist unwanted suites. For more information, see [PCoIP Connection Manager Configuration Settings](#).

Blacklisting Cipher Suites for PCoIP Security Gateway Connections

You can configure the PCoIP Security Gateway to support a subset of the previous cipher suites. The `SSLCipherBlackList` setting enables removing cipher suites from the previous list. For more information, see [PCoIP Connection Manager Configuration Settings](#).

TLS Protocol Update

The PCoIP Connection Manager and Security Gateway 20.10 can run on RHEL/CentOS 8. As a result of running on RHEL/CentOS 8, the TLS protocol versions enabled will be TLS 1.2 and TLS 1.3. If you encounter the following error message regarding TLS 1.1 you may need to enable TLS 1.1: *"javax.net.ssl.SSLHandshakeException: Server chose TLSv1, but that protocol version is not enabled or not supported by the client"*.

If you want to enable TLS 1.1 you need to update the CentOS security policy to **LEGACY**.

To view the current security policy run the following command:

```
update-crypto-policies --show
```

To set the policy to **LEGACY** run the following command:

```
update-crypto-policies --set LEGACY
```

Troubleshooting Connectivity Issues

A common cause of PCoIP session connectivity issues is firewall misconfiguration. Use tools such as ssldump and tcpdump (for Linux) and Wireshark (for Windows) to verify that packets sent by a particular source are actually received at the intended destination.

Verifying Network Connectivity

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

Verifying PCoIP Client to PCoIP Connection Manager Connectivity

To use `ssldump` to verify PCoIP client to PCoIP Connection Manager connectivity on TLS port 443:

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

```
sudo ssldump -i eth0 host <client-ip-address> port 443
```

2. From the client, connect to the PCoIP Connection Manager.
3. Verify from `ssldump` output that the PCoIP Connection Manager is receiving data from the client.

Verifying PCoIP Connection Manager to Connection Broker Connectivity

To verify PCoIP Connection Manager to connection broker connectivity on TLS port 443:

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.

Verifying PCoIP Connection Manager to PCoIP Agent Connectivity

To verify PCoIP Connection Manager to agent connectivity on TLS port 60443:

1. On the virtual desktop host, 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.

Verifying PCoIP Client to PCoIP Security Gateway Connectivity

To verify that the server hosting the PCoIP Security Gateway is receiving session initiation data from the client on TLS port 4172:

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

```
sudo ssldump -i eth0 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.

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

Verifying PCoIP Security Gateway is Receiving UDP Traffic from the Client

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 eth0 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 tcpdump output that the PCoIP Security Gateway is receiving data from the client.

Verifying PCoIP Server is Receiving UDP Traffic from the Client

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 eth0 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 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

If you cannot connect to the PCoIP Connection Manager, ensure you had uninstalled the httpd service before you installed the connection manager. If the httpd service was installed when you install the PCoIP Connection Manager, you must uninstall the httpd service and then reinstall the PCoIP Connection Manager.

Verifying PCoIP Connection Manager Status

To verify the PCoIP Connection Manager and its components, issue the verification commands from the server hosting the PCoIP Connection Manager.

The PCoIP Connection Manager is a web application that runs under Tomcat—a Java application launched under the ownership of the connection_manager system user.

To verify Tomcat is operating, use the ps command to find processes running under the connection_manager user:

```
ps -fu connection_manager
```

If the process is running, you see output similar to the following:

```
UID  PID  PPID  C  STIME TTY      TIME CMD
498 2264    1  0  00:51 ?    00:01:10 /usr/bin/java -Djava.../tomcat/...
```

To verify PCoIP Connection Manager web application operation:

1. Establish a TLS connection with openssl s_client:

```
openssl s_client -crlf -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, it returns XML with an element. If the PCoIP Connection Manager is not operational, check these log files for errors:

- /var/log/Teradici/ConnectionManager/catalina.log.
- /var/log/Teradici/ConnectionManager/pcoip-connmgr_*.log.

Verifying PCoIP Security Gateway Status

If you have configured the PCoIP Connection Manager to use the PCoIP Security Gateway, use the `ps` command to find processes running under the `security_gateway` user:

```
ps -fu security_gateway
```

If the process is running, you see output similar to the following:

```
UID    PID  PPID  C  STIME TTY      TIME CMD
4172  4818  4816  0   22:43 ?        00:00:00 /opt/Teradici/SecurityGateway/...
```

The PCoIP Security Gateway listens for the PCoIP Connection Manager to notify it of pending connections on TCP port 50060.

When establishing a PCoIP session, you can use `tcpdump` to verify that the PCoIP Connection Manager is communicating with the PCoIP Security Gateway:

```
sudo tcpdump -i lo port 50060
```

To verify that the PCoIP Security Gateway is also waiting for TLS connections from PCoIP clients on port 4172:

```
openssl s_client -crlf -connect 127.0.0.1:4172 -server localhost
```

If the command fails to establish a TLS connection, check the current PCoIP Security Gateway log file for errors.

Troubleshooting Certificate Errors

 **Error messages may be caused by different issues**

Error messages in this topic might be caused by issues other than certificate errors.

If you have enabled agent or broker certificate validation, then you must:

- Install properly constructed, CA-signed certificate files to the agents and/or the broker.
- Import the appropriate CA-signed certificate into the keystore the PCoIP Connection Manager uses.

If the PCoIP Connection Manager receives an invalid certificate or is unable to establish trust of the certificate, users get one of the following error messages:

Error Message	Possible Cause
Connection to the broker lost	Occurs on connection to the PCoIP Connection Manager when the PCoIP Connection Manager cannot validate the certificate from the connection broker .
Command failed due to a PCoIP agent failure	Occurs after authentication when selecting a resource to connect to, when the PCoIP Connection Manager cannot validate the certificate from the PCoIP agent.

In addition to the previous error messages, the PCoIP Connection Manager writes an error message in the log file when a certificate validation failure occurs. The following table describes some of the exceptions that the PCoIP Connection Manager may log during certificate validation.

Exception and Message	Possible Cause
CertificateException The certificate presented by the server does not meet minimum key length requirement.	The key length of the leaf certificate presented by the broker or agent is less than the BrokerCertMinKeyLength or AgentCertMinKeyLength setting in <code>/etc/ConnectionManager.conf</code> .

Exception and Message	Possible Cause
CertificateException No subject alternative DNS name matching found.	The Subject Alternative Name attribute in the leaf certificate presented by the broker or agent does not match the host name of the broker or agent. If the Subject Alternative Name attribute is not present in the leaf certificate presented by the broker or agent, then the Common Name (CN) field of the certificate's Subject does not match the host name of the broker or agent.
CertificateExpiredException NotAfter:	The timestamps of a certificate in the chain presented by the broker or agent indicate the certificate has expired.
CertificateNotYetValidException NotBefore:	The timestamps of a certificate in the chain presented by the broker or agent indicate the certificate is not yet valid.
CertPathValidatorException Basic constraints check failed: this is not a CA certificate.	Either the root CA certificate or one of the intermediate CA certificate files in the chain presented by the broker or agent has not been authorized as a CA certificate – the CA Boolean of the certificate's Basic Constraints attribute has not been specified or is not 'true'.
CertPathValidatorException Signature check failed.	The signature of a certificate in the chain presented by the broker or agent does not match the content of the certificate – the content or signature may have been tampered with.
SunCertPathBuilderException Unable to find valid certification path to requested target.	One or more certificate files are missing from the chain presented by the broker or agent. Neither the root CA certificate nor any of the intermediate CA certificate in the chain presented by the broker or agent are present in the keystore. Either the root CA certificate or one of the intermediate CA certificate files in the chain presented by the broker or agent has not been authorized for signature verification – the keyCertSign bit has not been set in the certificate's Key Usage attribute.
ValidatorException Extended key usage does not permit use for TLS server authentication.	The Extended Key Usage attribute of the leaf certificate presented by the broker or agent is present but does not specify the Server Authentication purpose.

Troubleshooting Error Messages

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

Command failed due to a connection broker communication failure.

Possible cause	Resolution
The connection broker is down or unreachable	Ensure the broker server is up and the broker service is running.
Broker certificate validation is enabled but the broker certificate is invalid.	<p>Ensure the PCoIP Connection Manager can reach the broker.</p> <p>Ensure a properly constructed and valid certificate is installed on the broker.</p> <p>Ensure the certificate the broker presents has been imported to the keystore.</p>

Connection to the broker lost

Possible cause	Resolution
The connection broker is down or unreachable	Ensure the broker server is up and the broker service is running.
Broker certificate validation is enabled but the broker certificate is invalid.	<p>Ensure the PCoIP Connection Manager can reach the broker.</p> <p>Ensure a properly constructed and valid certificate is installed on the broker.</p> <p>Ensure the certificate the broker presents has been imported to the keystore.</p>

Command failed due to a PCoIP agent failure

Possible causes:

Possible cause	Resolution
The PCoIP agent may be down or unreachable.	<p>Ensure the host is up and the agent service is running.</p> <p>Ensure the PCoIP Connection Manager can reach the agent.</p>
Agent certificate validation is enabled but the agent certificate is invalid	<p>Ensure a properly constructed and valid certificate is installed on the agent.</p> <p>Ensure the certificate the broker presents has been imported to the keystore.</p>

Retrieving the Support Bundle

The following section outlines how to get the support bundle for the PCoIP Connection Manager and Security Gateway. To get the support bundle run the following command:

```
sudo /opt/Teradici/Management/bin/support-bundler.sh
```

To check the system usage run the following command:

```
sudo /opt/Teradici/Management/bin/support-bundler.sh -h
```

Both these commands must be run with superuser privileges. The generated bundle is stored in the */tmp* directory as a *.tar.gz* file. It should be named as */tmp/teradici-cmsg-support-bundle- $\{DATESTR\}$.tar.gz*.

The PCoIP Connection Manager and Security Gateway support bundle contains:

- Configuration files for the Connection Manager and Security Gateway.
- Third-party configurations files.
- Log files for the Connection Manager and Security Gateway.
- System information (CPU, memory size, disk size, network information, system logs and more).

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.

Warning: Maintain your log files

The PCoIP Connection Manager and the PCoIP Security Gateway do not monitor available disk space. To prevent service disruptions caused by a full hard drive, periodically delete old log files.

Logging Specifications

	PCoIP Connection Manager	PCoIP Security Gateway
Maximum log file size	25 MB	2 MB
Maximum number of log files	100	Unlimited
Old log files	Compressed	Not compressed
Log file rotation	Daily at midnight local time or when log file reaches maximum size	Daily at midnight local time or when log file reaches maximum size
Log file encryption	No	No
Pre-session logs	Yes	Yes

	PCoIP Connection Manager	PCoIP Security Gateway
In-session logs	No ¹	Yes

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

PCoIP Connection Manager Log File Locations

The default log directory for the PCoIP Connection Manager is `/var/log/Teradici/ConnectionManager/`. The following log files are stored in this location:

Type	Filename
Current PCoIP Connection Manager log file	<code>pcoip-connmgr_<timestamp>.log</code>
Archived PCoIP Connection Manager log files	<code>pcoip-connmgr_<timestamp>.log.gz</code>
Tomcat log file	<code>catalina.log</code>

For example, the default filepath of the *current* PCoIP Connection Manager log file is `/var/log/Teradici/ConnectionManager/pcoip-connmgr_<timestamp>.log`.

PCoIP Security Gateway Log File Locations

The default log directory for the PCoIP Security Gateway is `/var/log/Teradici/SecurityGateway/`. Only PCoIP Security Gateway logs are stored in this location:

Type	Filename
All log files	SecurityGateway_<timestamp>.log

The current log file for the PCoIP Security Gateway is the file at `/var/log/Teradici/SecurityGateway/SecurityGateway_<timestamp>.log` with the most recent timestamp.

Log Verbosity

PCoIP logs can capture log messages at several different verbosity levels, ranging from highly verbose informational messages to error-only reporting. The verbosity of logs in both the PCoIP Connection Manager and PCoIP Security Gateway can be customized by setting the `LogLevel` value in their respective `conf` files.

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

Changing the PCoIP Connection Manager Log Level

To configure the log level of the PCoIP Connection Manager:

1. Edit the PCoIP Connection Manager configuration file `/etc/ConnectionManager.conf`.
2. Find and modify the `LogLevel` value:

```
LogLevel = <log level value>
```

Where `<log level value>` is one of `TRACE`, `DEBUG`, `INFO`, `WARN`, or `ERROR`.

3. Restart the PCoIP Connection Manager to apply changes:

```
service connection_manager restart
```

Changing the PCoIP Security Gateway Log Level

PCoIP Security Gateway log levels are numeric, and correspond to the same log levels used in the PCoIP Connection Manager. The levels are:

- 0 : TRACE
- 1 : DEBUG
- 2 : INFO
- 3 : WARN
- 4 : ERROR

To configure the log level of the PCoIP Security Gateway:

1. Edit the PCoIP Connection Manager configuration file `/etc/SecurityGateway.conf`.
2. Find and modify the `LogLevel` value:

```
LogLevel = <log level value>
```

Where `<log level value>` is one of 0, 1, 2, 3, or 4.

3. Restart the PCoIP Connection Manager to apply changes:

```
service security_gateway restart
```

Periodically delete old log files

When you set the log level to trace or debug, the system may create a large volume of logs. To prevent service disruptions caused by a full hard drive, periodically delete old log files.

1. The PCoIP Connection Manager is only active during the pre-session phase.

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 Teradici.

Before creating a ticket, be prepared with the following:

- A detailed description of the problem
- Your PCoIP Connection Manager and PCoIP Security Gateway version numbers.

The Teradici 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 Teradici PCoIP Technical Support Service team. Teradici staff are heavily involved in the forums.

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