Teradici PCoIP® Connection Manager 1.8 and Security Gateway 1.14

Administrators' Guide

teradici®

TER1502010/A-1.8-1.14

Contents

Document History

Version	Date	Description
A	26th October 2017	• Guide updated to align with PCoIP License Server 2.0.

Who Should Read This Guide?

This guide provides information for system administrators who are installing, configuring, maintaining or troubleshooting the PCoIP Connection Manager 1.8 and the PCoIP Security Gateway 1.14 as an infrastructure component for use with Cloud Access Platform, as well as with third party connection brokers, for example Leostream. In this guide you'll learn about:

- Setting up Certificates
- Load Balancing Considerations
- Creating a PCoIP Connection Manager and Security Gateway Certificate
- Administering the PCoIP Connection Manager
- Troubleshooting the PCoIP Connection Manager and Security Gateway

Note: Understanding terms and conventions in Teradici guides
For information on the industry specific terms, abbreviations, text conventions, and graphic symbols used in this guide, see Using Teradici Product and Component Guides and the Teradici Glossary.

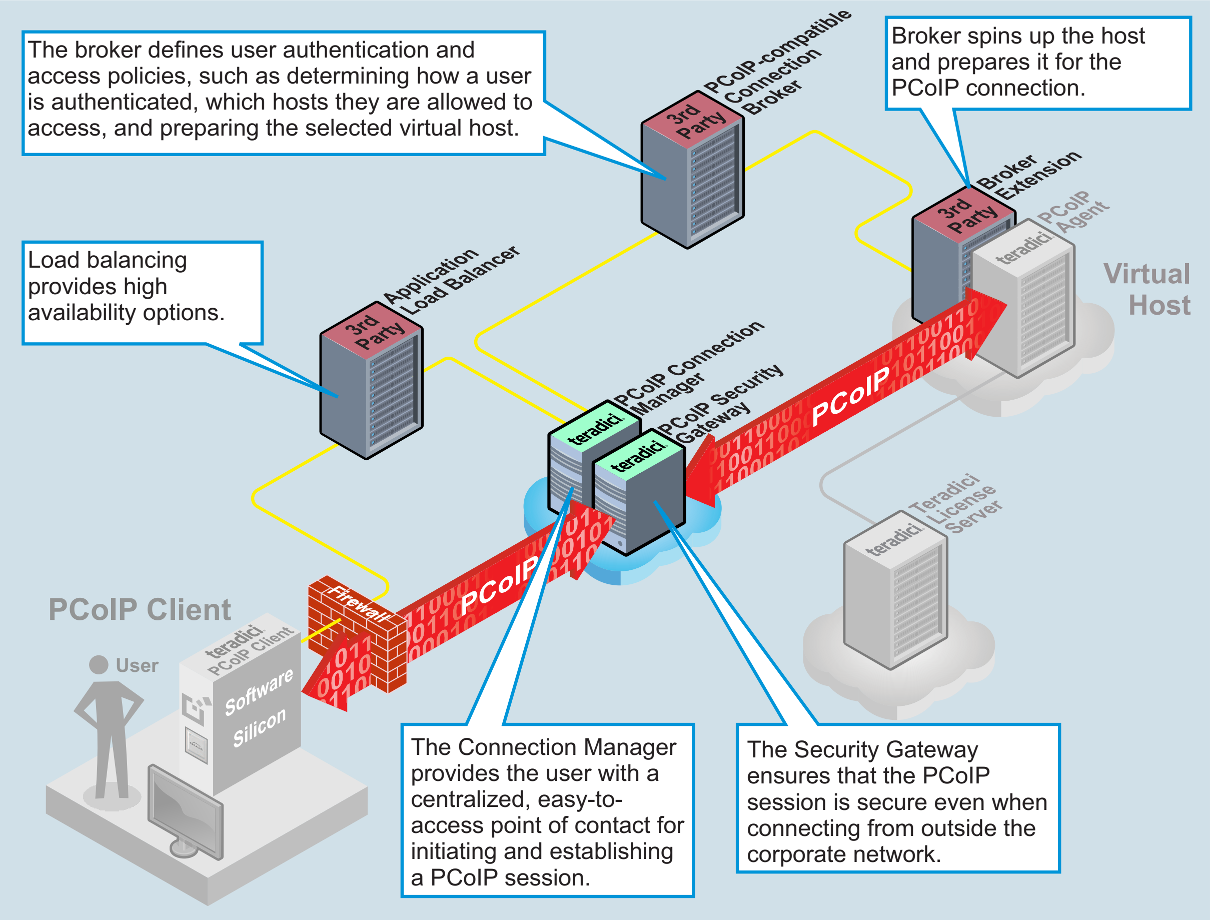
PCoIP Connection Manager and PCoIP Security Gateway Overview

The PCoIP Connection Manager and the PCoIP Security Gateway are both part of the Teradici Cloud Access Platform and are deployed in pairs.

The PCoIP Connection Manager enables the PCoIP client and the PCoIP agent to establish a remote desktop connection by creating a PCoIP session. It works with a connection broker to authenticate the user and query the desktop and applications. This can be a third-party connection broker from a partner, for example Leostream, or as part of the Teradici Cloud Access Software solution.

The PCoIP Security Gateway enables WAN users to securely access their remote desktops via the Internet without setting up a VPN connection. The PCoIP Security Gateway is not required for LAN access.

The diagram shown next illustrates an internal client accessing the PCoIP host using the PCoIP Connection Manager and the PCoIP Security Gateway.



The broker defines user authentication and access policies, such as determining how a user is authenticated, which hosts they are allowed to access, and preparing the selected virtual host.

Broker spins up the host and prepares it for the PCoIP connection.

Load balancing provides high availability options.

The Connection Manager provides the user with a centralized, easy-to-access point of contact for initiating and establishing a PCoIP session.

The Security Gateway ensures that the PCoIP session is secure even when connecting from outside the corporate network.

Brokered Connection

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.

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

- If all your desktops are on a LAN (internal access only), you may only need to install one PCoIP Connection Manager. You have the option of disabling the PCoIP Security Gateway since the PCoIP Security Gateway is not required for a LAN.

- If your desktops are on a WAN or 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 users use the internal PCoIP Connection Manager while external users use the external PCoIP Connection Manager with the PCoIP Security Gateway.
- If you serve a large number of desktops, or for high availability and load balancing, install more connection managers and implement load balancing.

For system requirements, see *System Requirements* on page 13.

Load Balancing

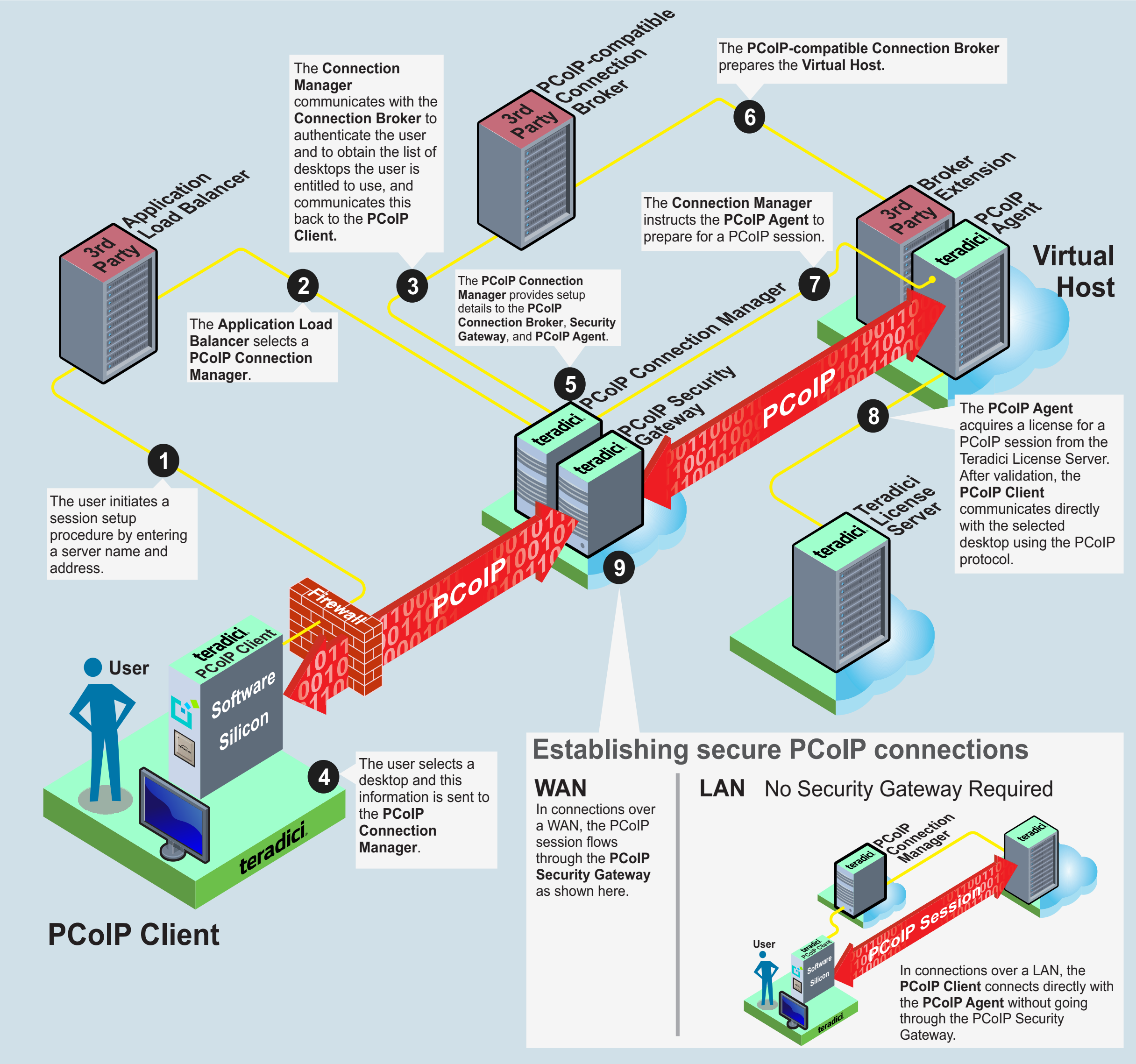
You can use an application load balancer in front of the PCoIP Connection Manager to distribute system load and optimize performance.

Note: Connection manager works with reverse proxy load balancing
The PCoIP Connection Manager is compatible with reverse proxy load balancing only. Do not use DNS round robin load balancing.

For more information, see *Load Balancing Considerations* on page 27.

Session Establishment

The diagram shown next illustrates the process of establishing a PCoIP session. The PCoIP Security Gateway is enabled to secure the connection and to proxy authorized traffic. Adapt this example for your installation.



1 The user initiates a session setup procedure by entering a server name and address.

2 The **Application Load Balancer** selects a **PCoIP Connection Manager**.

3 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, and communicates this back to the **PCoIP Client.**

4 The user selects a desktop and this information is sent to the **PCoIP Connection Manager**.

5 The **PCoIP Connection Manager** provides setup details to the **PCoIP Connection Broker**, **Security Gateway**, and **PCoIP Agent**.

6 The **PCoIP-compatible Connection Broker** prepares the **Virtual Host.**

7 The **Connection Manager** instructs the **PCoIP Agent** to prepare for a PCoIP session.

8 The **PCoIP Agent** acquires a license for a PCoIP session from the Teradici License Server. After validation, the **PCoIP Client** communicates directly with the selected desktop using the PCoIP protocol.

9

Establishing secure PCoIP connections

WAN In connections over a WAN, the PCoIP session flows through the **PCoIP Security Gateway** as shown here.

LAN No Security Gateway Required

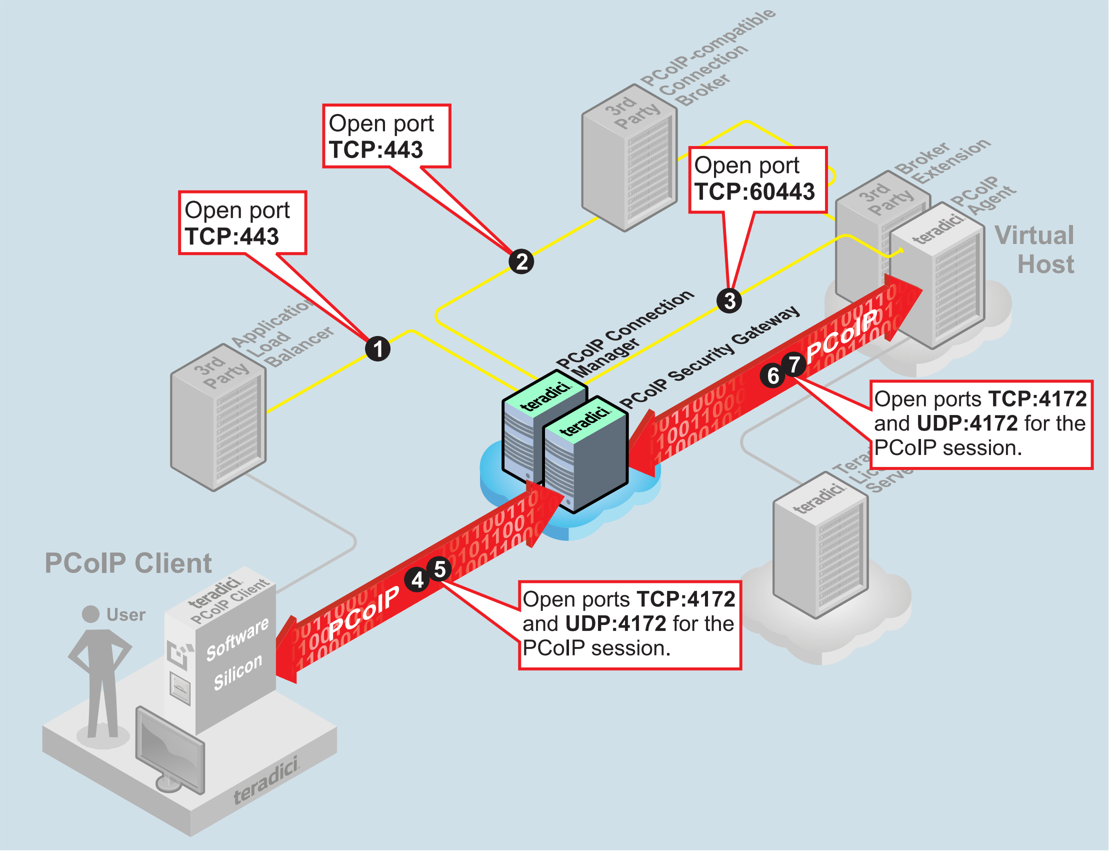
In connections over a LAN, the **PCoIP Client** connects directly with the **PCoIP Agent** without going through the PCoIP Security Gateway.

Overview of a session establishment using a brokered connection

To integrate and deploy your own or third-party brokering options, see the *Cloud Access Platform 2.10 Architecture Guide* about PCoIP brokering session messages and options.

Firewall Considerations

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

Ref	Source	Port	Destination	Port	Protocol	Description
❶	PCoIP client	*	PCoIP Connection Manager	443	TCP	PCoIP broker protocol. HTTPS access. By default, port 443 is enabled for client connection.
❷	PCoIP Connection Manager	*	Connection broker	443	TCP	PCoIP broker protocol. HTTPS access.
❸	PCoIP Connection Manager	*	PCoIP agent	60443	TCP	PCoIP agent protocol.

Ref	Source	Port	Destination	Port	Protocol	Description
❹	PCoIP client	*	PCoIP Security Gateway	4172	UDP	PCoIP user data.
❺	PCoIP client	*	PCoIP Security Gateway	4172	TCP	PCoIP control information.
❻	PCoIP Security Gateway	*	PCoIP agent	4172	TCP	PCoIP control information.
❼	PCoIP Security Gateway	55000	PCoIP agent	4172+	UDP	PCoIP user data. When deploying a host desktop using a standard or graphics agent, only port 4172 needs to be open. When deploying a host desktop using a multi-session agent, then multiple ports need to be open. Configure the number of open ports to be equal to or greater than the maximum number of sessions expected to be served by any of the multi-session agents.

Ensure these ports are open for inbound connections:

- 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.

On Red Hat Enterprise Linux (RHEL) 7 only SSH service is permitted by default. Use these commands to open ports to enable 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
```

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

- 443 TCP: used by the PCoIP Connection Manager to connect to third-party brokers.
- 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.

Note: Outbound traffic permitted by default
On Red Hat Enterprise Linux (RHEL) 7 , outbound traffic is permitted by default.

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 the PCoIP Security Gateway must be installed on the same server with Red Hat Enterprise Linux (RHEL) 7 installed with a single network interface. The instructions in this topic are for servers where RHEL 7.

The PCoIP Connection Manager and PCoIP Security Gateway are bundled together in one RPM.

Note: Use a text editor
You'll need a text file editor. If you like, install GNU nano or a text file editor of your choice.

System Requirements

The minimum 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.
- Red Hat Enterprise Linux (RHEL) 7.

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.

Caution: Use NTP to synchronize the time between components
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, and to assist with troubleshooting.

Preparing the PCoIP Connection Manager and PCoIP Security Gateway

To prepare the PCoIP Connection Manager and PCoIP Security Gateway:

1. Ensure Red Hat Enterprise Linux (RHEL) 7 is installed.
2. Ensure networking starts on boot.
3. Ensure that the NTP is enabled.
4. Ensure port 443 is available during installation.

Note: 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 or the client may not be able to connect to the PCoIP Connection Manager.

Installing PCoIP Connection Manager and PCoIP Security Gateway

To install the PCoIP Connection Manager and the PCoIP Security Gateway RPM package, you must have root access to run the installation script.

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.

Copy these files to a directory on the PCoIP Connection Manager server:

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

In the directory containing the installation files, use the command line to run the installation script:

```
sudo sh cm_setup.sh
```

Note: Setup script sets the system default to Java 8
The setup script will set the system default Java version to Java 8.

This 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 the PCoIP Connection Manager and the PCoIP Security Gateway. Some software is built from source during the setup procedure. For details, see the contents of this script. You can modify this setup script for your system.

`/opt/Teradici/thirdparty/README` lists the currently-supported version of third-party components in the third-party dependencies RPM. Changing the version of any of these third-party components might cause incompatibility with other components.

PCoIP Connection Manager and Security Gateway RPM Package Contents

The following tables show the files installed by the RPM packages. The RPM packages must be installed in this order:

1. Install the PCoIP Connection Manager third-party dependencies RPM.
2. Install the PCoIP Connection Manager and PCoIP Security Gateway RPM.

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.

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

Configuring the PCoIP Connection Manager

The commands in this topic are examples to illustrate how to configure the PCoIP Connection Manager and the PCoIP Security Gateway by editing their configuration files.

The commands in this procedure use these values:

- The public IP address of the PCoIP Security Gateway. In most cases this is the static IP address assigned to the PCoIP Connection Manager.
- Teradici PCoIP License Server:
  - Version 1.x format: `port-number@ip-address-or-hostname`
  - Version 2.x format: `http://<license-server-address>:<port>/request`
- The IP address of the connection broker (which supports the PCoIP broker protocol).

Caution: 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.

Configuring the Settings for the PCoIP Connection Manager

In most deployments, it is recommended that cloud-based licensing is adopted. Using this approach, PCoIP Agents in your domain are validated against cloud licensing services maintained by Teradici, thereby alleviating any need to configure or maintain your own licensing infrastructure. The PCoIP License Server can be used when the PCoIP Agent cannot access the internet. The PCoIP License Server can be hosted on-premises, for example ESXi, or on the public cloud, for example AWS or Azure. To use the PCoIP Connection Manager, 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 = See section above for PCoIP License Server 1.x
and 2.x formats.
```

For a complete list of settings that can be configured, see *PCoIP Connection Manager Configuration Settings* on page 32.

Configuring the PCoIP Security Gateway

The PCoIP Security Gateway enables users on a WAN to securely access their remote desktops from the Internet without the need to set up a VPN connection. The PCoIP Security Gateway is not needed for LAN access.

If you deploy Teradici Platform over a WAN, using a PCoIP Security Gateway is highly recommended. If Teradici Platform is not published over a WAN, configuring the PCoIP Security Gateway is optional. Restart the PCoIP Security Gateway and then the PCoIP Connection Manager to apply changes.

To use the PCoIP Security Gateway, ensure the following line is in `/etc/ConnectionManager.conf`:

```
SecurityGatewayEnabled = true
```

Edit **/etc/SecurityGateway.conf** and set the following fields:

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

Ensure the PCoIP Security Gateway's file descriptor limit is at least 15,000.

The limit is configured in `/etc/security/limits.conf` with the following lines:

For a complete list of settings that can be configured, see *Administering the PCoIP Security Gateway* on page 38.

Setting up 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.

☒ **Warning: Replace the self-signed certificate with your own certificate**
For production deployments, Teradici strongly recommends that you replace the self-signed certificate with your own that a trusted certificate authority (CA) has signed. Ensure all certificates you use conform to your organization's security policy.

For added security, consider using separate certificates for the PCoIP Connection Manager and the PCoIP Security Gateway.

You can create a certificate signing request (CSR) to get a signed public key certificate file or, for testing environments, you can create a self-signed certificate. For more information on creating certificates, see *Appendix: Creating a PCoIP Connection Manager and Security Gateway Certificate* on page 56.

Default Location of Certificate Files

By default, the PCoIP Connection Manager and the PCoIP Security Gateway use the same self-signed certificate files.

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.

The default certificate files are created in `/opt/Teradici/certs/opt/Teradici/certs` during RPM installation:

- `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.

- `CMCertificateKey.pem` contains the private key that the server uses to decrypt the symmetric key provided by the client.

Warning: Limit access to the private key
For security, limit access to the private key. Anyone with access to the key could capture and decrypt the data passing over the TLS connections established with the corresponding public key. Ensure access to all certificates conforms to your organization's security policy.

- `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.

  For an externally issued certificate, format the file as shown:

```
-----BEGIN CERTIFICATE -----
(Your server (leaf) certificate)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE -----
(Your intermediate CA certificate(s))
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE -----
(Your root CA certificate)
-----END CERTIFICATE-----
```

Note: Chain may contain intermediate CA certificate files
A chain may contain one or more intermediate CA certificate files.

For a self-signed certificate, copy `CMCertificate.pem` and rename it `CMCertificateCA.pem`.

To enable the PCoIP Connection Manager and the PCoIP Security Gateway to load and use the certificate files, ensure they are readable by the `teradici_components` group.

Note: 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.

Warning: Ensure all certificates conform to your security policy
Ensure all certificates you install conform to your organization's security policy, including file ownership and permissions.

Installing New Certificate Files

Install the new certificate by renaming the existing certificate files and copying the new certificate files to the same location. This example uses the files in `~/certs` created in *Appendix: Creating a PCoIP Connection Manager and Security Gateway Certificate* on page 56.

To install new certificate files:

1. In the PCoIP Connection Manager server, open a command prompt.
2. Rename the existing certificate files:

```
# 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:

```
# 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
```

4. Restart PCoIP Connection Manager components:

```
# service security_gateway restart
```

```
# service connection_manager restart
```

5. Check the PCoIP Connection Manager log file to ensure the PCoIP Connection Manager web service started:

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

   Look for these lines:

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

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

   Also check that it does not contain lines with `SEVERE:` as they may indicate that the certificate failed to load.

6. Check the PCoIP Security Gateway log file to ensure the PCoIP Security Gateway service started by checking the most recent file in `/var/log/Teradici/SecurityGateway`:

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

Warning: Protect the certificate files
After Installing new certificate files, ensure you see *Protecting Certificate Files* on page 24.

Protecting Certificate Files

Warning: Ensure all installed certificates follow your security policy
Ensure all certificates you install conform to your organization's security policy, including file ownership and permissions.

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
```

Regenerating the Self-Signed Certificate

Warning: Replace the self-signed certificate with your own certificate
For production deployments, we strongly recommend replacing the self-signed certificate with your own certificate that a trusted certificate authority (CA) has signed.

Note: Regenerate the self-signed certificate if you change host name
If you use the default self-signed certificate and you change the system host name, you must regenerate the self-signed certificate.

To use the `make_certs.sh` script to generate a new set of self-signed certificate files in the current directory:

```
cd ~
```

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

☒ **Warning: 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.

Regenerating and Installing the Self-Signed Certificate

Use the `--install` option to regenerate and install the certificate files at the same time. Files are installed in the `/opt/Teradici/certs` directory and overwrite any existing files with the same name.

If the `/opt/Teradici/certs` directory does not exist, the script creates 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

Optional: Configuring Certificate Location and File Names

By default, certificate files are created in `/opt/Teradici/certs` during installation and do not 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.

Configure the PCoIP Connection Manager certificate location and file names 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

Configure the following parameters in the *Administering the PCoIP Security Gateway* on page 38 section to specify the PCoIP Security Gateway certificate location and file names in `/etc/SecurityGateway.conf`.

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

☒ **Warning: Ensure all certificates 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.

Load Balancing Considerations

You can use load balancers in front of multiple connection managers and security gateways to distribute system load to optimize performance.

If you use load balancers, ensure that the `ExternalRoutableIP` setting for each PCoIP Security Gateway is directly reachable from clients.

During session establishment, the PCoIP Connection Manager provides the client with the configured `ExternalRoutableIP` of the PCoIP Security Gateway that's deployed on the same server. If that `ExternalRoutableIP` setting is incorrectly configured as the load balancer address, the load balancer can potentially direct the client to a PCoIP Security Gateway on a different server and the client won't be able to establish a session.

For more information, see the diagram in the *Firewall Considerations* on page 10 showing how the firewall and load balancer fit into the configuration.

Security Considerations

Warning: Follow your organization's security policy
For all security and certificate procedures, ensure you follow your organization's security policy.

Agent and Broker Certificate Validation

Warning: Enable validation of certificate files
For production deployments, Teradici strongly recommends enabling validation of certificate files presented by PCoIP agents and broker.

For a system using a PCoIP broker, 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 PCoIP Connection Manager's keystore. See *Importing Certificates into the Keystore* on page 28.
- Enable PCoIP Connection Manager agent and broker certificate validation. In `/etc/ConnectionManager.conf`, enter:

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

Caution: 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.

Importing Certificates into the Keystore

To validate the agent and broker certificates, the PCoIP Connection Manager uses the Java system default keystore. The exact location of the will vary depending on your Java installation and system configuration. In the Java home directory, the keystore path is typically:

<java-home>/<jre>/lib/security/cacerts

To import a certificate into the keystore so that the PCoIP Connection Manager can establish trust of the certificate signed by it:

1. In 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.

Note: 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

Warning: 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>
```

PCoIP Connection Manager Supported 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

Blacklisting Cipher Suites for the PCoIP Client Connections

You can configure the PCoIP Connection Manager to support a subset of the previous cipher suites. The `ClientSSLCipherBlackList` setting enables you to remove cipher suites from the previous list. For more information, see *PCoIP Connection Manager Configuration Settings* on page 32.

Note: 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 the Connection Broker and the PCoIP Agent Connections

You can configure the PCoIP Connection Manager to support a subset of the previous cipher suites for connections to the connection broker and to the PCoIP agents. The `ServerSSLCipherBlackList` setting enables you to remove cipher suites from the previous list. For more information, see *PCoIP Connection Manager Configuration Settings* on page 32.

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

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* on page 32.

Administering the PCoIP Connection Manager

This section contains information on how to start and stop, configure, uninstall, and upgrade your PCoIP Connection Manager.

Starting or Stopping the PCoIP Connection Manager

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

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

Note: Restarting the connection manager may take up to a minute
Restarting the PCoIP Connection Manager may take up to a minute to complete initialization. Clients cannot establish PCoIP sessions through it until initialization is complete.

PCoIP Connection Manager Configuration Settings

The configuration files for the PCoIP Connection Manager are located at `/etc/ConnectionManager.conf`. Restart the PCoIP Connection Manager to apply changes.

PCoIP Connection Manager 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* on page 45.
BrokerType		Type of the broker the PCoIP Connection Manager is using.

Parameter	Default	Description
PcoipAddress		Address of the PCoIP broker that the PCoIP Connection Manager uses to authenticate users and obtain resource information.
SecurityGatewayEnabled	false	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.
LicenseServerAddress		One or more license server addresses and port numbers. Use the format `port-number@ip-address-or-hostname` or `http://<license-server-address>:<port>/request`. Cannot be more than 1024 ASCII characters. Do not use `<`, `>`, and `&`. A local license server should be used for offline sites. Cloud licensing requires that the LicenseServerAddress be left blank.
ContentLengthEnabled	false	If set to true, the PCoIP Connection Manager always sets the `Content-Length: XX` in the HTTP response header. If set to false the PCoIP Connection Manager sends HTTP responses using chunked encoding.
BrokerMaxResp WaitSeconds	20	The time in seconds to wait for a response from the broker (other than for authenticate or allocate resource responses) before timing out.
BrokerMaxAllocate WaitSeconds	60	The time in seconds to wait for a response from the broker to an allocate resource request before timing out.
BrokerMaxAuthentication WaitSeconds	30	The time in seconds to wait for a response from the broker to an authenticate request before timing out.
AgentMaxResp WaitSeconds	160	The time in seconds to wait for a response from the PCoIP agent before timing out.
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.

Parameter	Default	Description
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* on page 30. 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* on page 30. 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.
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* on page 40.

Uninstalling PCoIP Connection Manager and PCoIP Security Gateway

This section provides information on how to uninstall PCoIP Connection Manager components. If necessary, back up your files before uninstalling them.

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.

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

```
sudo yum remove cm_sg
sudo yum remove 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.

To manually delete these files, execute:

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

Upgrading PCoIP Connection Manager and PCoIP Security Gateway

There is not an in-place upgrade available when moving to this version of the PCoIP Connection Manager and PCoIP Security Gateway, both the operating system as well as PCoIP Connection Manager and PCoIP Security Gateway require upgrading. Different procedures apply for on-premises and public cloud upgrades, as explained in the following sections.

On-Premises PCoIP Connection Manager with PCoIP Security Gateway enabled

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

1. Build a new RHEL 7 VM and install the PCoIP Connection Manager and PCoIP Security Gateway software. See *Installing PCoIP Connection Manager and PCoIP Security Gateway* on page 13.
2. Configure the PCoIP Connection Manager and PCoIP Security Gateway configuration files to match the existing PCoIP Connection Manager and PCoIP Security Gateway you are replacing. Custom certificates must also be installed. See *Administering the PCoIP Connection Manager* on page 32 and *Administering the PCoIP Security Gateway* on page 38.

3. 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.
4. Shut down the existing PCoIP Connection Manager and PCoIP Security Gateway.
5. 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.
6. Test a connection directly to the PCoIP Connection Manager and PCoIP Security Gateway external IP.

Note: 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.

Note: 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.

On-Premises PCoIP Connection Manager with PCoIP Security Gateway disabled

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

1. Build a new RHEL 7 VM and install the PCoIP Connection Manager and PCoIP Security Gateway software. See *Installing PCoIP Connection Manager and PCoIP Security Gateway* on page 13.
2. Configure the PCoIP Connection Manager and PCoIP Security Gateway configuration files to match the existing PCoIP Connection Manager and PCoIP Security Gateway you are going to replace. Custom certificates must also be installed. See *Administering the PCoIP Connection Manager* on page 32 and *Administering the PCoIP Security Gateway* on page 38.
3. Add the IP address of the new PCoIP Connection Manager and PCoIP Security Gateway to the load balancer or round robin DNS.
4. Remove the IP address of the legacy PCoIP Connection Manager and PCoIP Security Gateway from the load balancer or round robin DNS.

Upgrading the PCoIP Connection Manager and PCoIP Security Gateway in the Public Cloud

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

1. Build a new RHEL 7 VM and install the PCoIP Connection Manager and PCoIP Security Gateway software. See *Installing PCoIP Connection Manager and PCoIP Security Gateway* on page 13.
2. Assign a new external IP to the VM and install any custom certificates required.
3. Modify the configuration files to match the other PCoIP Connection Manager and PCoIP Security Gateway VM's using the assigned external IP. See *Administering the PCoIP Connection Manager* on page 32 and *Administering the PCoIP Security Gateway* on page 38.
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 and repeat for each PCoIP Connection Manager and PCoIP Security Gateway that is being replaced.
6. Remove the legacy PCoIP Connection Manager and PCoIP Security Gateway from the load balancer.

PCoIP Connection Manager Performance Metrics

These performance metrics are based on the minimum system requirements for the PCoIP Connection Manager and PCoIP Security Gateway.

Session Establishment Limits

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

- 40 sessions established per second.
- Up to 250 sessions established simultaneously.

Administering the PCoIP Security Gateway

This section contains information on how to start and stop and configure your PCoIP Security Gateway.

Starting or Stopping the PCoIP Security Gateway

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

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

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* on page 45.

Parameter	Default	Description
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.
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).

Parameter	Default	Description
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* on page 30. 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.

Parameter	Default	Description
TCPControlLinuxExtPriv	CMCertificateKey.pem	File name of the TCPControlLinuxExtCert certificate private key (in base64-encoded PEM format). **Note: 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* on page 32.

Uninstalling the PCoIP Security Gateway

Uninstalling the PCoIP Security Gateway also uninstalls the PCoIP Connection Manager.

To uninstall the PCoIP Security Gateway components, see *Uninstalling PCoIP Connection Manager and PCoIP Security Gateway* on page 34.

Upgrading PCoIP Connection Manager and PCoIP Security Gateway

There is not an in-place upgrade available when moving to this version of the PCoIP Connection Manager and PCoIP Security Gateway, both the operating system as well as PCoIP Connection Manager and PCoIP Security Gateway require upgrading. Different procedures apply for on-premises and public cloud upgrades, as explained in the following sections.

On-Premises PCoIP Connection Manager with PCoIP Security Gateway enabled

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

1. Build a new RHEL 7 VM and install the PCoIP Connection Manager and PCoIP Security Gateway software. See *Installing PCoIP Connection Manager and PCoIP Security Gateway* on page 13.
2. Configure the PCoIP Connection Manager and PCoIP Security Gateway configuration files to match the existing PCoIP Connection Manager and PCoIP Security Gateway you are replacing. Custom certificates must also be installed. See *Administering the PCoIP Connection Manager* on page 32 and *Administering the PCoIP Security Gateway* on page 38.
3. 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.
4. Shut down the existing PCoIP Connection Manager and PCoIP Security Gateway.
5. 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.
6. Test a connection directly to the PCoIP Connection Manager and PCoIP Security Gateway external IP.

Note: 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.

Note: 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.

On-Premises PCoIP Connection Manager with PCoIP Security Gateway disabled

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

1. Build a new RHEL 7 VM and install the PCoIP Connection Manager and PCoIP Security Gateway software. See *Installing PCoIP Connection Manager and PCoIP Security Gateway* on page 13.
2. Configure the PCoIP Connection Manager and PCoIP Security Gateway configuration files to match the existing PCoIP Connection Manager and PCoIP Security Gateway you are going to replace. Custom certificates must also be installed. See *Administering the PCoIP Connection Manager* on page 32 and *Administering the PCoIP Security Gateway* on page 38.
3. Add the IP address of the new PCoIP Connection Manager and PCoIP Security Gateway to the load balancer or round robin DNS.
4. Remove the IP address of the legacy PCoIP Connection Manager and PCoIP Security Gateway from the load balancer or round robin DNS.

Upgrading the PCoIP Connection Manager and PCoIP Security Gateway in the Public Cloud

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

1. Build a new RHEL 7 VM and install the PCoIP Connection Manager and PCoIP Security Gateway software. See *Installing PCoIP Connection Manager and PCoIP Security Gateway* on page 13.
2. Assign a new external IP to the VM and install any custom certificates required.
3. Modify the configuration files to match the other PCoIP Connection Manager and PCoIP Security Gateway VM's using the assigned external IP. See *Administering the PCoIP Connection Manager* on page 32 and *Administering the PCoIP Security Gateway* on page 38.
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 and repeat for each PCoIP Connection Manager and PCoIP Security Gateway that is being replaced.
6. Remove the legacy PCoIP Connection Manager and PCoIP Security Gateway from the load balancer.

PCoIP Security Gateway Performance Metrics

These performance metrics are based on the minimum system requirements for the PCoIP Connection Manager and PCoIP Security Gateway.

Session Limits

The PCoIP Security Gateway supports a maximum of 5,000 simultaneous sessions. The `MaxConnections` setting in `/etc/SecurityGateway.conf` controls this hard limit.

Bandwidth Limits

Based on the recommended minimum configuration, the PCoIP Security Gateway is capable of forwarding up to 400 Mbps of PCoIP session traffic (PCoIP UDP datagrams). This is a soft limit. To prevent performance degradation beyond this limit, use a PCoIP Security Gateway server with greater computing power.

Troubleshooting the PCoIP Connection Manager and Security Gateway

Use this section to troubleshoot the PCoIP Connection Manager and Security Gateway.

PCoIP Connection Manager and Security Gateway Log Files

Each PCoIP component logs its activities. Troubleshooting usually begins with checking PCoIP log files for error conditions or other indicators that show why the system may not be operating as expected.

Each PCoIP component is configured to log at a particular verbosity level and Teradici recommends using the default log level in a production deployment. The LogLevel setting in each component's configuration file controls the quantity and type of log messages written to the log file.

To assist with troubleshooting, all PCoIP components use the same 36-character hexadecimal string in the log files to correlate different components for a specific session.

When troubleshooting a problem, Teradici might recommend changing the log level for specific components to obtain more information from parts of the system.

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 files have these parameters:

	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	
Log file encryption	No	

Warning: Maintain 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.

For a complete list of settings that can be configured, see *PCoIP Connection Manager Configuration Settings* on page 32 and *Administering the PCoIP Security Gateway* on page 38.

Log File Default Location

Logs for components are required for troubleshooting depending on whether error conditions happen before the PCoIP session (pre-session) or during the session (in-session).

All log files are named with a timestamp suffix; the current log file for each component has the most recent timestamp.

See the default locations of the log files in the following table.

Component	Log File Default Location	Pre-Session	In-Session
PCoIP Connection Manager	/var/log/Teradici/ConnectionManager/pcoip-connmgr_*timestamp*.log	✓	
PCoIP Security Gateway	/var/log/Teradici/SecurityGateway/SecurityGateway_*timestamp*.log	✓	✓

The PCoIP Connection Manager default log file location includes:

- Current PCoIP Connection Manager log file (`pcoip-connmgr_`*`timestamp`*`.log`).
- Old, compressed PCoIP Connection Manager log files (`pcoip-connmgr_`*`timestamp`*`.log.gz`).
- Tomcat log file (`catalina.log`).

Log Level Configuration

This topic shows how to change the log level for the PCoIP Connection Manager and PCoIP Security Gateway.

PCoIP Connection Manager Log Level Configuration

Log levels are TRACE, DEBUG, INFO, WARN, ERROR.

To configure the log level of the PCoIP Connection Manager:

1. Edit the PCoIP Connection Manager configuration file `/etc/ConnectionManager.conf`.
2. Change `LogLevel =` *`TRACE/DEBUG/INFO/WARN/ERROR`*.
3. Restart the PCoIP Connection Manager to apply changes.

PCoIP Security Gateway Log Level Configuration

Log 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 Security Gateway configuration file `/etc/SecurityGateway.conf`.
2. Change `LogLevel =` *`0/1/2/3/4`*.
3. Restart the PCoIP Security Gateway .

Caution: 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.

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
PCoIP client to PCoIP Connection Manager	TLS port 443
PCoIP Connection Manager to connection broker	TLS port 443
PCoIP Connection Manager to PCoIP agent	TLS port 60443

If you are using a security gateway:

Connection	Port
PCoIP client to PCoIP Security Gateway	TLS port 4172, UDP port 4172
PCoIP Security Gateway to PCoIP agent	TLS port 4172, UDP port 4172+

If you are not using a security gateway:

Connection	Port
PCoIP client to PCoIP agent	TLS port 4172, UDP port 4172+

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 collectivity 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 ssldump 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 <error-resp> 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

Note: 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 `/etc/ConnectionManager.conf`.

Exception and Message	Possible Cause
CertificateException No subject alternative DNS name matching *<host-name>* 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: *<timestamp>*	The timestamps of a certificate in the chain presented by the broker or agent indicate the certificate has expired.
CertificateNotYetValidException NotBefore: *<timestamp>*	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. For more information, see *Importing Certificates into the Keystore* on page 28. 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

This topic lists some common PCoIP client error messages and their possible causes.

Error	Possible Cause and Resolution
Command failed due to a connection broker communication failure	• The connection broker might be down or unreachable. ◦ Ensure the broker server is up and the broker service is running. ◦ Ensure the PCoIP Connection Manager can reach the broker. See *Verifying PCoIP Connection Manager to Connection Broker Connectivity* on page 49. • Broker certificate validation is enabled but the broker certificate is invalid. ◦ Ensure a properly constructed and valid certificate is installed to the broker. See *Agent and Broker Certificate Validation* on page 28. ◦ Ensure the certificate the broker presents has been imported to the keystore. See *Importing Certificates into the Keystore* on page 28.
Connection to the broker lost	
Command failed due to a PCoIP agent failure	• The PCoIP agent might be down or unreachable. ◦ Ensure the host is up and the agent service is running. ◦ Ensure the PCoIP Connection Manager can reach the agent. See *Verifying PCoIP Connection Manager to PCoIP Agent Connectivity* on page 49. • Agent certificate validation is enabled but the agent certificate is invalid. ◦ Ensure a properly constructed and valid certificate is installed to the agent. See *Agent and Broker Certificate Validation* on page 28. ◦ Ensure the certificate the agent presents has been imported to the keystore. See *Importing Certificates into the Keystore* on page 28.

Appendix: Creating a PCoIP Connection Manager and Security Gateway Certificate

Warning: Ensure all certificates conform to your security policy
Ensure all certificates you use conform to your organization's security policy.

The PCoIP Connection Manager and PCoIP Security Gateway installation script generates a self-signed certificate; so there is no green padlock or https notation when loaded on a client or the zero client.

All settings and software in this topic are only examples. These are basic examples that do not show how to use certificate fields such as subject alternative names.

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

To create and install certificate files on the PCoIP Connection Manager and PCoIP Security Gateway:

1. Create a temporary staging directory, for example, `~/certs`.
2. See *Creating a Private Key and Certificate Signing Request* on page 56.
3. You can send the CSR to obtain a signed public key certificate .csr file (now a .crt file) or, for testing, you can create a self-signed certificate. See *Obtaining the Signed Public Key Certificate* on page 59 and *Creating a Self-signed Certificate* on page 59
4. Verify and convert certificate file format to get the three required certificate files. See *Verifying and Converting Certificate File Format to Get the Three Certificate Files* on page 61
5. Install and verify the new certificate files on the PCoIP Connection Manager and PCoIP Security Gateway. See *Installing Certificate Files on the PCoIP Connection Manager and Security Gateway* on page 62.

Creating a Private Key and Certificate Signing Request

This topic uses openssl as an example to generate two files:

- `private.key` contains a new private key.
- `server.csr` contains a certificate signing request (CSR).

Submit the CSR to a trusted certificate authority (CA) for signing: either a trusted third-party CA or a trusted internal CA.

The private key/signed certificate pair from the CA secure communication between the PCoIP Connection Manager/PCoIP Security Gateway and the client.

Depending on your security policy, you have the option of using two private key/signed certificate pairs on the server: one for the PCoIP Connection Manager and one for the PCoIP Security Gateway. In this example, we'll use the same private key/signed certificate pair for both the PCoIP Connection Manager and the PCoIP Security Gateway.

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.

To generate a private key and CSR:

1. In the PCoIP Connection Manager server, open a command prompt.
2. Create a temporary directory to store the certificate and change to this directory, for example, create a `certs` directory under the user's home directory:

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

   Note: Example creates certificate in the user's home directory
   This example creates the certificate in the user's home directory (`~/certs`). Later, the certificate will be moved to the appropriate location to complete the procedure.

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
   ```

The previous command generates a series of questions asking for information to be displayed in the certificate. You cannot use the following characters in the **Organization Name** or the **Organizational Unit Name**: < > ~ ! @ # $ % ^ * / \ ( ) ?.,&

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.	Teradici Corporation
Organization Unit Name	Department of your organization.	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	

To verify the details of the CSR request:

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

Obtaining the Signed Public Key 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 the file.

Creating a Self-signed Certificate

For evaluation and testing purposes, you may choose to create a self-signed certificate. If you choose to create a self-signed certificate, ensure you follow your organization's security policy.

Warning: Ensure all certificates conform to your security policy
Ensure all certificates you use conform to your organization's security policy.

This example creates two files on CentOS 6.8: `private.key` containing the private key and `certificate.pem` containing the self-signed public certificate.

Note: Some settings are required to ignore certificate errors
For a self-signed certificate, when connecting with a PCoIP client, some settings may be required to ignore certificate errors. See the client documentation for details.

To generate a private key file and CSR:

1. In the PCoIP Connection Manager server, open a command prompt.
2. Create a temporary directory to store the certificate and change to this directory, for example, create a `certs` directory under the user's home directory:

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

Note: Example creates certificate in the user's home directory
This example creates the certificate in the user's home directory (`~/certs`). Later, the certificate will be moved to the appropriate location to complete the procedure.

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 self-signed certificate using the sha384 hash algorithm. The private key file is `private.key` and the self-signed certificate is `certificate.pem`.

   ```
   openssl req -x509 -sha384 -nodes -days 365 -newkey rsa:3072
   -keyout private.key -out certificate.pem
   ```

The previous command generates a series questions asking for information to be displayed in the certificate. You cannot use the following characters in the **Organization Name** or the **Organizational Unit Name**: < > ~ ! @ # $ % ^ * / \ ( ) ?.,&

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.	Teradici Corporation
Organization Unit Name	Department of your organization.	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	

Verifying and Converting Certificate File Format to Get the Three Certificate Files

If you generated a self-signed certificate using the previous example, you can skip the verification. If your certificate is signed by a CA, you need to verify that the file is in the correct format.

In this example:

- 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(s) and root certificates) is named `CAcertificate.crt`.

To verify the certificate file (**certificate.crt** or **CAcertificate.crt**), copy the file to your **CentOS 6.8** machine and verify that the certificate signed by the CA is in base64-encoded PEM format:

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

Check that the output shows the content of the certificate without any errors.

If you get error messages, open the file in an editor such as GNU nano or vi and check:

- 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 does not begin and end with the correct lines, convert the file to PEM format:

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

To verify the new file:

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

If necessary, change the certificate file extension to .pem format:

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

Repeat the previous procedures for `CAcertificate.crt`.

If you created a self-signed certificate, create **CAcertificate.pem** by copying **certificate.pem** file:

```
cp certificate.pem CAcertificate.pem
```

You now have a public signed certificate in base64-encoded PEM format named `certificate.pem` (without the certificate chain) and `CAcertificate.pem` (with the certificate chain).

Install these three files on the PCoIP Connection Manager and PCoIP Security Gateway:

- `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.

Backing Up Your Private Key

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

Installing Certificate Files on the PCoIP Connection Manager and Security Gateway

In the previous example, we created one set of certificate files that can be used on both the PCoIP Connection Manager and PCoIP Security Gateway.

After creating the certificate files, install the new certificate files. See *Installing New Certificate Files* on page 22.

Teradici recommends keeping the certificate in the default location, but if your organization's security policy requires changing the certificate location or file name, you can specify different certificate location or file names. See *Optional: Configuring Certificate Location and File Names* on page 25.

Warning: Ensure all certificates conform to your security policy
Ensure all certificates you use conform to your organization's security policy.

The PCoIP Connection Manager and PCoIP Security Gateway installation script generates a self-signed certificate; so there is no green padlock or https notation when loaded on a client or the zero client.

All settings and software in this topic are only examples. These are basic examples that do not show how to use certificate fields such as subject alternative names.

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

To create and install certificate files on the PCoIP Connection Manager and PCoIP Security Gateway:

1. Create a temporary staging directory, for example, `~/certs`.
2. See *Creating a Private Key and Certificate Signing Request* on page 56.
3. You can send the CSR to obtain a signed public key certificate .csr file (now a .crt file) or, for testing, you can create a self-signed certificate. See *Obtaining the Signed Public Key Certificate* on page 59 and *Creating a Self-signed Certificate* on page 59.
4. Verify and convert certificate file format to get the three required certificate files. See *Verifying and Converting Certificate File Format to Get the Three Certificate Files* on page 61.
5. Install and verify the new certificate files on the PCoIP Connection Manager and PCoIP Security Gateway. See *Installing Certificate Files on the PCoIP Connection Manager and Security Gateway* on page 62.

Teradici Corporation

#101-4621 Canada Way, Burnaby, BC V5G 4X8 Canada

phone +1.604.451.5800 fax +1.604.451.5818

www.teradici.com

The information contained in this documentation represents the current view of Teradici Corporation as of the date of publication. Because Teradici must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Teradici, and Teradici cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. TERADICI MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Teradici may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Teradici, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Visit Notice of Intellectual Property Rights for more information.