

# Anyware Connector Administrators Guide

**23.12**

# Table of Contents

Anyware Connector 23.12	4
What's New in the Anyware Connector (RHEL) for 23.12	5
Support for RHEL/Rocky Linux 9	5
Improved flexibility when using Active Directory integration	5
Anyware Connector VM Requirements	6
Installation and Upgrade	7
Prerequisites	7
Creating the Connector Server	7
Confirming the CIDR for Connector Cluster	11
DNS Name Resolution Configuration on RHEL/Rocky Linux	12
Preparing Security Certificates	15
Installing the Connector on RHEL/Rocky Linux	18
Prerequisite Steps	18
1. Adding the Connector Repository	19
2. Configuring SELinux Policies	20
3. Installing the Connector RPM	20
4. Generating a Connector Token	21
5. Configuring the Connector-Example Commands	23
5. Connecting to a Remote Workstation with a PCoIP Client	40
Upgrading the Connector on RHEL/Rocky Linux	42
Installing the Connector on RHEL/Rocky Linux- Darksite Installation	43
Prerequisite Steps	43
Reference	45
Anyware Connector Features	45
Difference between Anyware Connector on Ubuntu and RHEL/Rocky Linux	49

Transitioning Anyware Connectors	53
MFA Configuration	53
Transitioning the Connector	53
Testing the Connector	54
Adding the Connector to a Load Balancer	54
Proxy Configuration for Anyware Connector	55
Scaling and PCoIP Session Limits	56
Firewall and Load Balancing Considerations	57
Configuring the Active Directory for Anyware Connector	60
Configuring User and Computer Active Directory Distinguished Names	60
Configuring Active Directory Pool Groups	61
Security	63
Anyware Connector Multi-Factor Authentication	63
Duo Authentication	63
Azure MFA Authentication	64
Third-Party Multi-Factor Authentication	67
Troubleshooting	68
Anyware Connector Support Bundle on RHEL/Rocky Linux	68
Support Bundle Information and Logs	68
Anyware Connector Health Status on RHEL/Rocky Linux	71
Network Connectivity Issues	72
Remote Workstation Connectivity Check	72
Active Directory Connectivity Check	73
Configuring DNS Name Resolution	75
Getting Support	78
Contacting Support	78
The HP Community Forum	78
Release Notes	79

# Anyware Connector 23.12

Anyware Connector is an access hub installed on the customer environment that facilitates PCoIP Client connections to remote workstations. It operates in conjunction with HP Anyware Manager to provide user authentication and entitlement for remote workstation access, including MFA. It enables secure connectivity between users and the remote workstations by eliminating the need for a dedicated Virtual private network (VPN) by providing Network Address Translation (NAT) services for external users.

Anyware Connector enables Anyware Manager to broker desktops or workstations based in AWS, Google Cloud, Microsoft Azure, and on-premises environments. Based on your infrastructure, you may need more than one instance of the Connector. The Connector communicates with the Anyware Manager that orchestrates and manages system deployments.

You are required to have a valid registration code for HP Anyware Software to successfully deploy Anyware Manager. This code is sent to you via email from HP and looks like ABCDEF1234@AB12-C345-D67E-89FG. For more information on Anyware Software, see [Anyware Software](#).

The Anyware Connector runs on RHEL/Rocky Linux starting from Connector version 22.04.0. At HP, our constant endeavour is to simplify and unify our OS support strategy. To that effect, Anyware Manager and Anyware Connector will only support RHEL/Rocky Linux (8.0 or later) starting H2 of 2023, and we will be EOLing Ubuntu CAC (actual Date TBD). You will be notified six months before a migration path from Ubuntu to RHEL/Rocky Linux for new Anyware Connector.

# What's New in the Anyware Connector (RHEL) for 23.12

## Support for RHEL/Rocky Linux 9

This release introduces support for RHEL 9 and Rocky Linux 9.

## Improved flexibility when using Active Directory integration

This release introduces the ability to enable or disable AD Authentication and AD Sync individually. Previously, these features were switched on and off together. For more information, see [--enable-ad-authentication](#) and [--enable-ad-sync](#) flags documented in [Installation Flags and Options](#).

# Anyware Connector VM Requirements

The Anyware Connector runs on a RHEL or Rocky Linux VM with the following minimum requirements:

	Minimum requirement
<b>Operating System</b>	RHEL 8 or 9; Rocky Linux 8 or 9
<b>Memory</b>	4GB+ RAM
<b>CPUs</b>	4+ vCPUs
<b>Storage</b>	<p>30GB+ available storage.</p> <hr/> <p>If you are using LVM and <code>/var</code> is mounted on a separate volume, <code>/var</code> must have 30 GB+ of available storage.</p>

There are additional network and firewall configuration requirements; see [Creating the Connector Server](#) for complete installation and configuration instructions.

# Installation and Upgrade

## Prerequisites

### Creating the Connector Server

This section outlines steps to create the Connector server on RHEL/Rocky Linux and other system requirements that are required for installation.

#### MINIMUM REQUIREMENTS

Create a virtual machine that meets or exceeds the following requirements:

	Minimum requirement
<b>Operating System</b>	RHEL 8 or 9; Rocky Linux 8 or 9
<b>Memory</b>	4GB+ RAM
<b>CPUs</b>	4+ vCPUs
<b>Storage</b>	30GB+ available storage.  If you are using LVM and <code>/var</code> is mounted on a separate volume, <code>/var</code> must have 30 GB+ of available storage.

#### NETWORK REQUIREMENTS

Once you have setup a dedicated Virtual Machine(VM) for the Connector, please ensure the following environment conditions are met:

- You must have access to the internet for an online installation. For Darksite installation see, [Installing the Connector on RHEL/Rocky Linux- Darksite Installation](#)
- The virtual machine needs the following port configuration:
  - Port 4172 configured for inbound/outbound TCP/UDP traffic
  - Port 443 configured for outbound TCP traffic

- Port 443 configured for inbound TCP traffic if the connector is accepting connections from PCoIP clients external to the network. For additional port and firewall information, see [Firewall Load Balancing Considerations](#).
- You must have console access to the virtual machine using SSH.
- The server must be able to resolve the AD domain.
- You must have superuser (sudo) privileges on the virtual machine.
- The networking configuration of the server (including the IP address) must not change while the Connector is operational.

## FIREWALL CONFIGURATION

Before you configure firewall, please ensure the following conditions are met:

- The Virtual Machine must have port TCP 443 and TCP/UDP 4172 enabled in its firewall rules
- Within virtual network in the VM, the Firewalld is configured properly for Anyware Connector to run within the Virtual Machine.
- You can confirm it by running the following command:

```
sudo systemctl status firewalld
```

If the firewalld status is 'active', make sure you execute the following commands to configure firewall correctly. If the firewalld status is 'inactive' and your organization does not require firewall on the Anyware Connector VM, then please skip the step below.

Commands to configure firewall:

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



## DISABLE SWAP

Connector is built on K3s, and it's strongly recommended to disable swap on a Linux system to avoid memory issue in a production environment. It is recommended to disable swap on a Linux system to avoid memory issue.

You can do the following to disable swap:

- If this is a new install and you want to disable swap permanently on the Connector server:
  - Edit the `/etc/fstab` file and add '#' in front of any line that contains the word 'swap'.
- If you have an existing Connector and is running into memory issue, run the following command to disable swap immediately. (This is not retained after a system reboot):
  - `sudo swapoff -a`

If Swap is required for any reason, it should be greater or equal to the size of the RAM. There is no guarantee that it works, so it is strongly recommended to disable it.

## ENABLING CONNECTIONS OVER WAN

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

To enable external PCoIP connections:

- The remote server should have a public IP address. This can be done via bi-directional NAT mapping. During the installation, you should use the `--external-pcoip-ip` flag to set the IPv4 address for the Connector for external connections.
- By default `--enable-security-gateway` is set to true forcing all sessions to go through security gateway to allow external users to connect to their workstations, if your environment consists of internal users, the Security Gateway can be disabled by passing `--enable-security-gateway=false`.

## VERIFYING THE CONNECTOR SERVER

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

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

## DNS and Name Resolution

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

## Confirming the CIDR for Connector Cluster

Once the Connector server is verified, you need to confirm the CIDR for the Connector Cluster.

### CONFIRM THE CIDR FOR CONNECTOR CLUSTER

- The default CIDR used for the Connector's k3s network are: 10.42.0.0/16, 10.43.0.0/16, 10.43.0.10.
- The CIDR must not be in conflict with customer's enterprise network CIDRs, where the Connector Virtual Machine is accessible.
- If any of the default Connector Cluster's CIDR is in conflict, confirm the desired CIDRs to be used by Connector that are Not in conflict:
  - `--cluster-cidr` : this is to set cluster CIDR, default is 10.42.0.0/16.
  - `--service-cidr`: this is to set service CIDR, default is 10.43.0.0/16.
  - `--cluster-dns`: this is to set cluster dns ip address, default is 10.43.0.10, it has to be part of of the `--service-cidr`.
- Record the new CIDRs if the default is in conflict, they are required for Connector configuration during installation, For example, to change the cluster CIDR to 192.168.10.0 and service to 172.16.0.0. The configure command example: `sudo anyware-connector configure --cluster-cidr 192.168.10.0/24 --service-cidr 172.16.0.0/16 --cluster-dns 172.16.0.10.`

# DNS Name Resolution Configuration on RHEL/Rocky Linux

To install and configure Anyware Connector on the RHEL or Rocky Linux machine, its important to have a connection between the machine and the Active Directory Domain Controller.

## CHECK THAT THE DNS NAME RESOLUTION WORKS AS EXPECTED

1. Check the `/etc/resolv.conf` file to ensure that the desired DNS servers and search suffixes are present.

```
cat /etc/resolv.conf
# Generated by NetworkManager
search example-domain.com
nameserver 10.162.0.42
```

2. Test the DNS by pinging the Domain, in this example example-domain.com:

```
ping example-domain.com
```

3. If the response is successful, you should receive a message similar to the example below:

```
PING example-domain.com (10.162.0.42): 56 data bytes
64 bytes from 10.162.0.42: icmp_seq=0 ttl=118 time=16.622 ms
64 bytes from 10.162.0.42: icmp_seq=1 ttl=118 time=50.675 ms
64 bytes from 10.162.0.42: icmp_seq=2 ttl=118 time=27.682 ms
64 bytes from 10.162.0.42: icmp_seq=3 ttl=118 time=19.886 ms
^C
--- example-domain.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
```

4. Restart the Virtual Machine(VM) and check if the DNS settings in `/etc/resolv.conf` persist and that you can still ping the domain as shown in steps 2-3 above. If it does not work, please follow the steps in [Configure DNS Settings](#) below.

### Applying Host VM's DNS settings to K3S

The host Virtual Machine's DNS settings are copied from `/etc/resolv.conf` and applied to the Anyware Manager and/or Connector whichever is installed when the K3S service starts. Hence, it is important that settings are correct after a restart. You need to reboot the Virtual Machine or restart the K3S service to apply the DNS settings to the Anyware Manager or Connector whichever is installed, if changes are made post installation or configuration.

## CONFIGURE DNS SETTINGS

If the DNS Name resolution work as expected, please skip the steps below.

To ensure DNS settings are configured properly on the machine for Anyware Manager or Connector to operate, please perform the following steps(the sample IP of the Domain Controller is 10.162.0.42 for `example-domain.com`):

1. Disable auto-configuration of DNS settings bto prevent overwriting on reboot. In this example the device name is `eth0`.

```
nmcli device modify eth0 ipv4.ignore-auto-dns yes
```

You also need to disable this on the connection level in some cases. In this example the connection name is `eth0`.

```
nmcli connection modify eth0 ipv4.ignore-auto-dns yes
```

2. Add the DNS1 for the IP address for Active Directory's DNS server (typically the Domain Controller itself) and optionally DNS2 for fallback DNS server and optionally DOMAIN for a DNS suffix (typically the Domain name) in the network configuration scripts.

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=ens192
UUID=dfe16427-21f1-429c-99cb-a1e9b42be181
DEVICE=ens192
ONBOOT=yes
DNS1=10.162.0.42
DOMAIN=example-domain.com
PEERDNS=no
```

3. Restart the Network Manager.

```
sudo systemctl restart NetworkManager
```

4. Follow steps in the [Check that the DNS Name Resolution works properly](#) section to confirm the DNS name resolution works properly.

## Preparing Security Certificates

To ensure the communications between Anyware Connector and external entities are trusted and secured, the following certificates are recommended:

- Certificate for establishing LDAPs connection from Connector to AD , typically it is the DC certificate
- Certificate for the Connector to establish HTTPs connection from PCoIP client to Connector for Login
- Certificate for Anyware Manager to establish HTTPs connection from Connector to Anyware Manager installed locally

These certificates are not required, especially for testing purposes, there is option to bypass some of them, however it is recommended to have them for production use.

## Domain Controller Certificates

### CONFIGURING *LDAPS-CA-CERT* FLAG

Domain Controller Certificate is required for secure and trusted communication to the Active Directory using LDAPs. By default the certificate is signed by a private Certificate Authority(CA). However for the Connector to validate the certificate and communicate securely with the Active Directory the certificate should be signed by a Public Certificate Authority(CA). If verifying Active Directory certificate is required use `--ldaps-ca-cert` to pass Active Directory root certificate, in the case where validating the certificate is not required use `--ldaps-insecure` flag to skip verification.

Anyware Connector runs with the following supported Domain Controller servers:

- Windows 2016 Server with secure LDAP (LDAPS) enabled.
- Windows 2012 R2 Server with secure LDAP (LDAPS) enabled.
- Windows 2019 Server with secure LDAP (LDAPS) enabled.

It is recommended to provide Domain Controller or Domain's root certificate. Alternatively you can provide the public certificate for the leaf certificate for the Domain Controllers instead, leaf certificate is valid for a shorter time such as 1 year than the CA cert, which usually is valid for 5 years. For more information, see [How to create and install a self-signed certificate on a Windows 2016 Active Directory server to enable LDAPS](#).

If you don't have the CA cert, you can get the leaf certificate by running the following command:

```
openssl s_client -connect domain-controller.domain.com:636
```

Here the `domain-controller.domain.com` is the Domain Controller's Fully qualified domain name and 636 is the LDAPS port.

### CONFIGURING *LDAPS-INSECURE* FLAG

LDAPS with a root Certificate is the recommended way to use Anyware Connector. This way, communication from the Connector to the Active Directory is done using a secure TLS connection. If you do not wish to install the CA cert or want to skip certification verification for testing purposes, you can use `--ldaps-insecure` flag. This flag helps you establish a encrypted connection between the Connector and Active Directory however, that connection is not validated.

### CONFIGURING *ENABLE-PLAINTEXT-LDAP* FLAG

For non production environment, LDAP could be used instead of LDAPS to avoid setting up certificates. LDAP is non secure protocol and message between the Connector and Active Directory are sent in plain text.

To enable the LDAP mode, use the following flag:

```
--enable-plaintext-ldap
```

#### **Domain Controller certificates expiry**

When all the LDAPS certificates expire, the Connector stops working and displays an error message on the Connectors page. Also, a warning message that details the current state of the certificates is displayed on the same page when a Connector has a certificate that is less than a week away from expiring.

## Connector TLS Certificate

Connector TLS certificate is required for secure and trusted connection between PCoIP client and Anyware Connector, you can bypass this using `--self-signed` (or `--insecure`) flag which generates a self-signed certificate and key for the Connector. However, the PCoIP clients gets insecure warning when establishing a connection, which is recommended strictly only for testing purposes. For production use, you should assign a TLS certificate to the Connector during



installation. This prevents insecure connection errors when connecting to Anyware Connector, Anyware manager is not affected by this certificate.

## Anyware Manager Certificate

Anyware Manager Certificate could be required and obtained using `--manager-ca-cert` flag for secure and trusted connection from Connector to the Manager. You don't need to provide Anyware Manager certificate if:

- You are using Anyware Manager as it uses a certificate signed by a public CA
- You are using a trusted TLS certificates signed by a public CA when connecting to Anyware Manager.

If Anyware Manager is installed with self-signed certificate or a certificate signed by a public CA that is not trusted by the Connector, you need to provide Anyware Manager Certificate unless `--manager-insecure` flag is used to skip certificate validation for testing purpose.

## Expected Certificate file

The certificate supported by the Connector has certain requirements. They are as follows:

### THE ANYWARE CONNECTOR SUPPORTS THE CERTIFICATE FILE IN THE FOLLOWING FORM:

- A certificate in the **PEM** format as shown below:

```
-----BEGIN CERTIFICATE-----  
base64encodedcertdata  
-----END CERTIFICATE-----
```

- A certificate file including only a single certificate. For example: - A single self-signed certificate - A root CA certificate - A single leaf certificate that is signed by an existing root CA

### THE ANYWARE CONNECTOR DOESN'T SUPPORT THE CERTIFICATE FILE IN FOLLOWING FORM:

- A bundle certificate that includes multiple certificates such as root, intermediate, or leaf certificate.
- Leaf certificate that is signed by a different or an untrusted root CA by the Connector.

# Installing the Connector on RHEL/Rocky Linux

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

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

1. [Adding the Connector repository](#)
2. [Configuring the SELinux components](#)
3. [Installing the RPM](#)
4. [Generating the Connector Token](#)
5. [Connecting to a Remote Workstation with a PCoIP Client](#)

## Prerequisite Steps

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

### Before you begin

If you are currently using the Anyware Connector on Ubuntu, it is important to read and understand what the differences are between the Connector on Ubuntu and Connector on RHEL/Rocky Linux so you can prepare the installation correctly to minimize errors during installation.

For more information, see [Difference between Anyware Connector on Ubuntu and RHEL/Rocky Linux](#).

# 1. Adding the Connector Repository

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

## Checking Existing Repositories for Anyware Connector

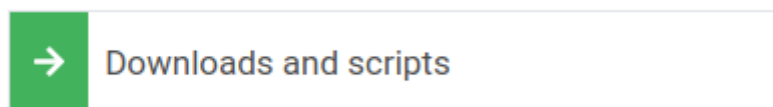
If the Anyware Connector was installed previously on your virtual machine, there could be existing repos related to it on your system. Run the command below to check all existing repos related to Anyware Connector (Skip this step if Anyware Connector was never installed on your virtual machine).

```
dnf repolist teradici-anyware-manager*
```

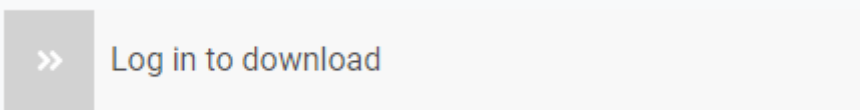
Check the current Anyware Connector repo to make sure it is the desired repo that you want to use for installation. If there are unwanted repositories on your VM, see [Repository Management](#) to remove them.

## Adding a Repository

1. To access the scripts and to configure and add the RHEL and Rocky Linux repository, select the **Downloads and scripts** option from the [Anyware Manager support site](#).

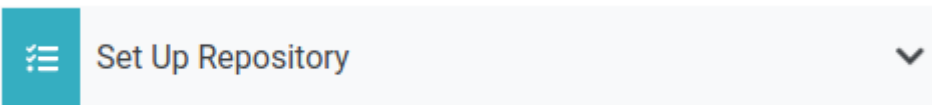


If you see a login button as such:



Click it to log into the site and then proceed.

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



3. The window expands and displays the setup scripts. Copy the `curl` script to clipboard.

4. Paste command on the target machine where you want to install Anyware Connector and press `Enter`.

## 2. Configuring SELinux Policies

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

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

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

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

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

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

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

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

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

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

## 3. Installing the Connector RPM

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

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

```
sudo dnf install -y anyware-connector
```

## 4. Generating a Connector Token

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

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

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

- Follow the step by step instructions outlined below.

## Connectors > Create a new connector

Connector name

Length must be between 2 and 32 characters. The following characters are not allowed:  
~!@#\$%^&\*()|+=÷¿?,:;"'.,<>{}[]/\_

Private cloud install instructions

### 1. Create the Anyware Connector server

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

### 2. Verify the Anyware Connector server

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

### 3. Enable external access


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

### 4. Download the Anyware Connector server

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

### 5. Get connector token

GENERATE



Copy the token to be used when installing the Anyware Connector.

eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJvaWQiOiI1YzljMTJlZT  
IzYmE3MDAwMWYyYjdhMjEiLCJkZXBsbnV5ZW50SWQoIl1YzljMTJl

(token is only valid for 1 hour.)

### 6. Install the Anyware Connector

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

CLOSE

4. Click **GENERATE**.
5. Copy the Connector token by click the copy icon.

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

## 5. Configuring the Connector-Example Commands

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

### Configuring the Connector for Anyware Manager


Once you have installed the Connector RPM, and have generated a Connector token from the Anyware Manager installed in your enterprise network, run the following commands to configure the Connector to work with the Anyware Manager in your enterprise network. The first line for these commands maps the Connector token to a variable in the shell, the '' for the string values are not required if there are no special chars in the string.

#### MINIMUM CONFIGURATION SAMPLE COMMAND FOR QUICK START

The following command with dummy values configures a Connector with minimum flags to work with the Anyware Manager in your enterprise network. Communications with external integrations such as PCoIP clients, Active Directory server, etc are not secure without certificate validation are not secure without certificate validation, this should only be used for testing purpose.

```
export token=<token from Anyware Manager Admin Console>
/usr/local/bin/anyware-connector configure \
--manager-url 'https://ipv4.Anyware.Manager.Installable' \
--token $token \
--domain 'testlab.internal' \
--accept-policies \
--enable-ad-sync=false \
--ldaps-insecure \
```

You can use the minimum command for testing or base installation excluding additional configurations. When editing a workstation, you should manually add workstations from the Admin Console and add a user assignment by the user's UPN as the domain users or computers are synced.

 **The ability to manually add a user assignment by the user's UPN is supported only in Anyware Manager as a Service combined with Anyware Connector RHEL/Rocky Linux 23.06 or later or Ubuntu Connector version 164 or later.**

## TYPICAL CONFIGURATION SAMPLE COMMAND

```
export token=<token from Anyware Manager admin console>
sudo /usr/local/bin/anyware-connector configure \
--manager-url 'https://ipv4.Anyware.Manager.Installable' \
--token $token \
--domain 'testlab.internal' \
--sa-user 'sampleuser' \
--sa-password 'Passwordstring' \
--ldaps-ca-cert '/home/rocky/DC-Cert.pem' \
--computers-dn 'CN=Computers,DC=testlab,DC=internal' \
--users-dn 'CN=Users,DC=testlab,DC=internal' \
--external-pcoip-ip 'public.ipv4.sg.ip' \
--self-signed \
--accept-policies \
--manager-insecure \
--debug
```

- `--manager-url` Sets the Anyware Manager URL that the Connector will connect to. If this flag is not provided, the connector will connect to Anyware Manager as a Service (`https://cas.teradici.com`). **Required** for instances of Anyware Manager Installable.
- *Installable only:* The `--manager-ca-cert` flag provides the PEM-formatted public certificate for the root CA used to sign the Anyware Manager certificate. This flag is required when Anyware Manager Installable is using a custom certificate that is not signed by a public CA. This flag has no effect when connecting to Anyware Manager as a Service.
- *Installable only:* The `--manager-insecure` flag is required if the target Anyware Manager is using self-signed certificates. This flag is not required if Anyware Manager is using trusted TLS certificates signed by a public CA. This flag has no effect when connecting to Anyware Manager as a Service.
- `--external-pcoip-ip`: Explicitly sets the public IP for PCoIP Client to PCoIP Agent connection, and is highly recommended as a best practice. If not provided, the installer will attempt to automatically resolve the external IP by reaching out to `https://cas.teradici.com`. This flag is required if automatic resolution fails, or if the Connector does not have a connection to the public internet.
- The first time you install a connector in a deployment, use `--computers-dn` and/or `--users-dn` flags to sync AD objects to Anyware Manager. Subsequent connectors installed in the same deployment will automatically inherit the configuration from Anyware Manager.

If these flags are not provided, the AD sync synchronises all objects from the AD to the Anyware Manager.



- If `--self-signed` flag is not used, you should use `--tls-key` and `--tls-cert` flags to provide the full path and filename of the TLS key and PEM formatted TLS certificate to use.
- If `--ldaps-ca-cert` flag is not used, you should use either `--ldaps-insecure` to skip certificate validation, or `--enable-ldap-plaintext` for test purposes.

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

## Additional Configurations for the Anyware Connector

### ⚠ Updating the Connector

When updating configurations for Anyware Connector using "Configure" command, it restarts to apply the updated configurations and all the active sessions going through the connector are disconnected. This requires users to log in again and reconnect.

## MULTI-FACTOR AUTHENTICATION

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

### ⚠ Multi-Factor Authentication for the Connector

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

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

## INSTALLING THE CONNECTOR FOR INTERNAL CONNECTIONS

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

1. Prepare a virtual machine in your private network that meets the [system requirements](#) with the following sub-steps:
  - Skip the step for preparing the system for external access.
  - Skip the step for setting up MFA.
2. Install the Connector with the following sub-steps:
  - If you don't have external users, then you could disable security gateway by passing `--enable-security-gateway=false`, otherwise it's set to true enabled by default.
  - Do not set the Public IP using the `--external-pcoip-ip` flag. The Connector returns the virtual machines IP address.
  - No MFA flag is required as MFA is disabled by default.
3. Once you have installed the Connector connect to a remote workstation with a [PCoIP Software Client](#) with the following sub-step:
  - In the *Host Address or Code* field enter the private IP of the internal Connector you just installed and log-in.

## INSTALLING THE CONNECTOR FOR EXTERNAL CONNECTIONS

The following steps outline how to install the Connector for external connections:

1. Prepare a virtual machine in your private network that meets the [system requirements](#) with the following sub-steps:
  - Skip the step for preparing the system for internal access.
2. Install the Connector with the following sub-steps:
  - Set the Public IP using the `--external-pcoip-ip` flag.
3. Once you have installed the Connector, connect to a remote workstation with a [PCoIP Software Client](#) with the following sub-step:
  - In the *Host Address or Code* field enter the IP address or DNS name of the external Connector you just installed and log-in.

## UPDATING CIDR FOR CONNECTOR CLUSTER

The default CIDR for Connector Cluster are as follows:

- 10.42.0.0/16 cluster CIDR
- 10.43.0.0/16 Service CIDR
- 10.43.0.10 Cluster DNS

If the default CIDRs conflict with your internal network, use the following flags to update the cluster with different CIDR.

To update, run the following command:

```
sudo anyware-connector configure --cluster-cidr <IP Address> --service-cidr  
<IP Address> --cluster-dns <IP Address>
```

Example Command with dummy values:

```
sudo anyware-connector configure --cluster-cidr 192.168.10.0/24 --service-  
cidr 172.16.0.0/16 --cluster-dns 172.16.0.10
```

## Installation Flags and Options

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

### Groups of flags

The flags are here categorized by their configuration groups:

### States for Boolean Flags

The state of all the Boolean Flags is interpreted as follows:

- `--boolean-flag` means "true".
- `--boolean-flag=true` means "true".
- `--boolean-flag=false` means "false".
- `--boolean-flag anytext` uses default as "true".

## ANYWARE MANAGER FLAGS

Flag	Config Key	Description
<code>--manager-url</code>	<code>manager.url</code>	<p>(String) The Anyware Manager URL that the Connector will connect to.</p> <hr/> <p>If this flag is not provided, the connector will connect to Anyware Manager as a Service (<a href="https://cas.teradici.com">https://cas.teradici.com</a>). <b>Required</b> for instances of Anyware Manager Installable.</p>
<code>--manager-ca-cert</code>	<code>manager.CaCertPath</code>	(String) Provide an Anyware Manager CA certificate, which the Connector uses to establish trust and connect to Anyware Manager.
<code>--manager-insecure</code>	<code>manager.Insecure</code>	(Boolean) Turns off verification of the CA certificate. Use this flag when connecting to an Anyware Manager instance that uses self-signed certificates.

## CONNECTOR FLAGS

Flag	Config Key	Description
--token	<code>connector.token</code>	( <i>String</i> ) <b>Required.</b> The token generated from Anyware Manager for Connector to create a service account to connect to Anyware Manager.
--accept-policies	<code>connector.AcceptPolicies</code>	( <i>Boolean</i> ) Automatically accept the <a href="#">EULA</a> and <a href="#">Privacy Policy</a> .
--push-config-to-manager	<code>connector.PushConfigToManager</code>	( <i>Boolean</i> ) Send (non-sensitive) configuration data to Anyware Manager, to be used by future Connector installations.
--pull-config-from-manager	<code>connector.PullConfigFromManager</code>	( <i>Boolean</i> ) Include this flag to retrieve Connector configuration data from Anyware Manager.

## CONNECTOR MULTIFACTORAUTHENTICATION FLAGS

Flag	Config Key	Description
-- enable-mfa	<code>connector.MultiFactorAuthentication.enable</code>	<i>(Boolean)</i> This flag can be used if you wish to enable multi-factor authentication. Multi-factor authentication will be enabled for all connections, both internal and external. Internal users will be required to enter the multi-factor authentication code for the Connector when connecting to the PCoIP Client. It is recommended to install separate Connectors for internal vs external connections.
-- radius-server	<code>connector.MultiFactorAuthentication.Server</code>	<i>(String)</i> The FQDN or IP address of the RADIUS server to use for MFA. Optional.
-- radius-port	<code>connector.MultiFactorAuthentication.Port</code>	<i>(String)</i> This is the RADIUS server port. If not specified, the default port (1812) is used. If <code>--radius-server</code> is specified, then this flag is optional.
-- radius-secret	<code>connector.MultiFactorAuthentication.SharedSecret</code>	<i>(String)</i> The shared secret used for configuring RADIUS authentication. If <code>--radius-server</code> is specified then this flag is required.

## CONNECTOR TLS FLAGS

Flag	Config Key	Description
<code>--self-signed</code>	<code>connector.tls.self-signed</code>	<i>(Boolean)</i> This mode is not secure, and should only be used for testing. PCoIP clients will receive a "untrusted" warning when connecting to the Connector. The previous <code>--insure</code> flag is still supported.
<code>--tls-cert</code>	<code>connector.tls.certpath</code>	<i>(String)</i> The full path and filename of the TLS certificate (in PEM format) to use. <i>If <code>--self-signed</code> is used, this flag has no effect.</i>
<code>--tls-key</code>	<code>connector.tls.keypath</code>	<i>(String)</i> The full path and filename of the TLS key to use. <i>If <code>--self-signed</code> is used, this flag has no effect.</i>

## CONNECTOR SESSION FLAGS



Flag	Config Key	Description
-- external- pcoip-ip	<code>connector.session.ip</code>	<p>(String) Manually sets the public IP for PCoIP Client to PCoIP Agent connection. If not provided, the installer will attempt to automatically resolve the external IP by reaching out to <code>https://cas.teradici.com</code>.</p> <hr/> <p>This flag is required if automatic resolution fails, or if the Connector does not have a connection to the public internet.</p>
--local- license- server-url	<code>connector.session.licenseServerUrl</code>	<p>(String) Sets the URL for PCoIP License Server to be used for PCoIP Sessions. If this is not provided, ensure that the Cloud License Server is registered on the PCoIP Agent. Example: <code>--local-license-server-url http://10.10.10.10:7070/request</code>. For more information on the PCoIP License Server, see <a href="#">PCoIP License Server</a>.</p>
--show- agent- state	<code>connector.session.showagentstate</code>	<p>(Boolean) Specifies whether the agent state is displayed as part of the remote workstation name in the PCoIP Client. The default value for this flag is true. This setting has no effect if <code>--retrieve-agent-state</code> is false.</p>
--retrieve- agent- state	<code>connector.session.retrieveagentstate</code>	<p>(Boolean) Enables the broker to retrieve the agent state for unmanaged and managed remote workstations. The default value for this flag is false. The available states are <b>In Session</b>, <b>Ready</b>, <b>Starting</b>, <b>Stopping</b>, <b>Stopped</b> and <b>Unknown</b>.</p>
-- preferred- name	<code>connector.session.preferredname</code>	<p>(String) This is an optional flag to determine if the hostname or machine name should be displayed to identify the remote workstations, the default is set to display machine name.</p>
--enable- security- gateway	<code>connector.session.enablesecuritygateway</code>	<p>(Boolean) By default the security gateway for external traffic is set to true. For internal traffic disable this feature using</p>

Flag	Config Key	Description
		the <code>--enable-security-gateway=false</code> flag.

## CONNECTOR SESSION TRUSTEDCUSTOMERLICENSING FLAGS

Flag	Config Key	Description
<code>--trusted-customer-license-cert</code>	<code>connector.session.TrustedCustomerLicensing.certpath</code>	(String) The full path, including filename, of the Trusted Customer licensing certificate file.
<code>--trusted-customer-license-key</code>	<code>connector.session.TrustedCustomerLicensing.keypath</code>	(String) The full path, including filename, of the Trusted Customer licensing key file.
<code>--clear-trusted-customer-license</code>	<code>connector.session.TrustedCustomerLicensing.clear</code>	(Boolean) Clear the Trusted Customer Licensing flag and key from the Connector configuration.

## CONNECTOR CLUSTERNETWORK FLAGS

Flag	Config Key	Description
<code>--service-cidr</code>	<code>connector.ClusterNetwork.serviceCidr</code>	(String) The IPv4 network CIDRs to use for service container IPs.
<code>--cluster-cidr</code>	<code>connector.ClusterNetwork.clusterCidr</code>	(String) The IPv4 network CIDRs to use for container pod IPs.
<code>--cluster-dns</code>	<code>connector.ClusterNetwork.clusterDns</code>	(String) The IPv4 address for DNS for cluster pods. Must be within the range indicated by <code>--cluster-cidr</code> .

## CONNECTOR DOMAIN FLAGS

Flag	Config Key	Description
--domain	<code>connector.Domain.Name</code>	(String) The AD domain that the remote workstations will join.
--enable-ad-authentication	<code>connector.Domain.EnableAuthentication</code>	(Boolean) Enables AD authentication. Default: <code>true</code>
--domain-controller	<code>connector.Domain.DomainControllers</code>	(String) This flag specifies one or more domain controllers to use with the Connector. To specify multiple domain controllers use the following format: <code>--domain-controller dc1.domain.com, --domain-controller dc2.domain.com, --domain-controller dc3.domain.com</code> .
--pool-group	<code>connector.domain.poolGroups</code>	(String) Specifies one or more Active Directory groups, by entering the distinguished name (DN) to be assigned to pools for remote workstation management (eg, <code>--pool-group 'CN=GroupPool1,CN=Users,DC=sample,DC=com'</code> or <code>--pool-group 'CN=GroupPool2,CN=Users,DC=sample,DC=com'</code> ).
--enable-plaintext-ldap	<code>connector.domain.enableLdapMode</code>	(Boolean) Connections to Active Directory will be made using plaintext LDAP instead of encrypted LDAPS. This is meant only for testing, do NOT use it in production.
--ldaps-ca-cert	<code>connector.domain.caCertPath</code>	(String) To supply a CA certificate for the connection to AD over LDAPS.
--ldaps-insecure	<code>connector.domain.insecure</code>	(Boolean) Skip certificate validation when connecting to the Active Directory using LDAPS. This option should only be used when connecting to the Active Directory deployed with self signed certificates. This will be ignored if a CA cert is provided.

## CONNECTOR DOMAIN SERVICEACCOUNT FLAGS

Flag	Config Key	Description
--sa-user	<code>connector.domain.serviceAccount.username</code>	(String) The AD service account username.
--sa-password	<code>connector.domain.serviceAccount.password</code>	(String) The AD service account password.

## CONNECTOR DOMAIN ADSYNC FLAGS

Flag	Config Key	Description
<code>--enable-ad-sync</code>	<code>connector.domain.adsync.enable</code>	(Boolean) Enable Active Directory synchronization.
<code>--users-dn</code>	<code>connector.domain.adsync.UserDns</code>	<p>(String array) The base DN to search for users within AD. Specify multiple DNs with multiple options. Newly provided base DN(s) will automatically replace previous base DN(s). The base DN to search for computers within the AD for AD sync.</p> <p>You can specify multiple DNs with multiple options. See the table above on the differences between the Connectors for more information.</p> <p>Newly provided base DN(s) automatically replaces previous base DN(s).</p>
<code>--users-filter</code>	<code>connector.domain.adsync.UserFilters</code>	<p>(String) The filter to search for users within Active Directory. Specify multiple filters with multiple options. The default user filter is</p> <p><code>(&amp;(objectCategory=person)(objectClass=user))</code>.</p>
<code>--computers-dn</code>	<code>connector.domain.adsync.ComputerDns</code>	<p>(String) The base DN to search for computers within AD for AD sync. Can specify multiple DNs with multiple options. See the differences between the Connectors at the top of this page for details. Newly provided base DN(s) will automatically replace previous base DN(s).</p>
<code>--computers-filter</code>	<code>connector.domain.adsync.CompuysterFilters</code>	<p>(String) The filter to search for computers within Active Directory. Specify multiple filters with multiple options. The default computer filter is</p> <p><code>(&amp;(primaryGroupID=515)(objectCategory=computer))</code>.</p>
<code>--sync-interval</code>	<code>connector.domain.adsync.Interval</code>	(uint8) The interval (in minutes) for how often to sync AD users and computers with the Anyware Service.

## CONNECTOR OAUTH FLAGS

Flag	Config Key	Description
<code>--enable-oauth</code>	<code>connector.oauth.enabled</code>	(Boolean) Enables OAuth authentication.
<code>--id-provider-url</code>	<code>connector.oauth.IDProviderUrl</code>	(String) Sets the identity provider URL. Example: <code>--id-provider-url https://provider-1234567890.okta.com</code> . This flag is required if <code>--enable-oauth is true</code> .
<code>--oauth-client-id</code>	<code>connector.oauth.IdpAppClientId</code>	(String) Sets the Client ID from the Identity Provider. This flag is required if <code>--enable-oauth is true</code> .
<code>--oauth-flow-code</code>	<code>connector.oauth.OauthFlowCode</code>	(String) Set the desired OAuth flow / grant type. Currently, only <code>OAUTH_FLOW_CODE_WITH_PKCE</code> is supported, and is used by default.
<code>--fa-url</code>	<code>connector.oauth.FaUrl</code>	(String) The Federated Auth Broker URL. for example <a href="https://cac-vm-fqdn:port">https://cac-vm-fqdn:port</a>
<code>--oauth-server-ca</code>	<code>connector.oauth.OauthServerTrustCaPath</code>	(String) The full path, including file name, of the OAuth Server CA certificate.

## CONNECTOR SSO FLAGS

Flag	Config Key	Description
--enable-sso	<code>connector.oauth.EnableSSO</code>	(Boolean) Enable Single Sign-On.
--sso-enrollment-url	<code>connector.oauth.SSOEnrollmentUrl</code>	(String) Sets the URL to the Active Directory Certification Authority Web Enrollment Service.
--sso-enrollment-certificate-template-name	<code>connector.oauth.SSOEnrollmentCertificateTemplate</code>	(String) Name of the certificate template that Active Directory Certificate Services (AD CS) uses to sign CSR.
--sso-enrollment-domain	<code>connector.oauth.SSOEnrollDomain</code>	(String) Domain of the user to access Active Directory Certification Authority Web Enrollment Service.
--sso-enrollment-username	<code>connector.oauth.SSOEnrollAccount</code>	(String) Username for accessing Active Directory Certification Authority Web Enrollment Service.
--sso-enrollment-password	<code>connector.oauth.SSOEnrollPassword</code>	(String) Password for the username to access Active Directory Certification Authority Web Enrollment Service.

## CONNECTOR OAUTH OPENSSL FLAGS

Flag	Config Key	Description
--sso-signing-csr-ca	<code>connector.oauth.opensslCsrSign.CaCertPath</code>	Path to copy intermediate CA Certificate.
--sso-signing-csr-key	<code>connector.oauth.opensslCsrSign.CaKeyPath</code>	Path to the intermediate key.
--sso-signing-csr-crl	<code>connector.oauth.opensslCsrSign.CaCRLPath</code>	Path to a certificate revocation list.

## 5. Connecting to a Remote Workstation with a PCoIP Client

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

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

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

- [PCoIP Software Client for Windows](#)



- [PColP Software Client for macOS](#)
- [PColP Software Client for Linux](#)

# Upgrading the Connector on RHEL/Rocky Linux

The Anyware connector on RHEL/Rocky Linux can be upgraded in a 2-step process outlined as follows:

## Upgrading a Connector

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

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

```
sudo dnf upgrade -y anyware-connector
```

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

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

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

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

# Installing the Connector on RHEL/Rocky Linux- Darksite Installation

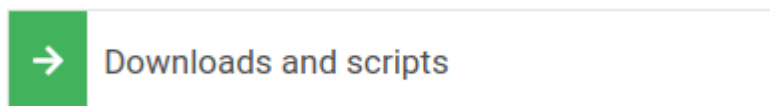
In cases where the Connector needs to be installed in a **Darksite** (i.e. an environment without internet access, also known as **airgap** or **offline** environment) you can download the installation files, transfer them to the target Darksite machine, and then run the installation script.

## Prerequisite Steps

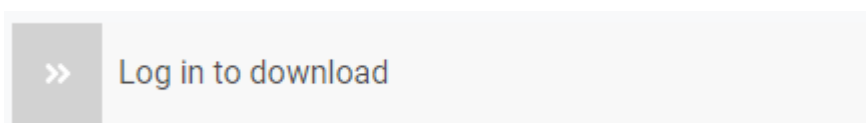
Before you begin, see [Connector System Requirement](#) to prepare your target machine.

### 1. Download and Transfer the Installation Files

To access the scripts and to configure and add the RHEL and Rocky Linux repository, select the **Downloads and scripts** option from the [Anyware Manager support site](#).



If you see a login button as such:



Click it to log into the site and then proceed. Once logged in, read and accept the End User License Agreement and download the `.tar.gz` file.

### 2. Transferring the File

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

To transfer the file you can copy it onto a media device, such as a USB drive or a DVD, and then connect that device to the target darksit machine. You can also connect the target darksite machine

to another machine via SSH or FTP, and complete a network file transfer. This method may not be viable for some darksite networks.

### 3. Extract the Installation File

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

```
sudo tar xzvf anyware-connector-offline_Linux.tar.gz
```

When you extract the file, a new folder called anyware-connector-offline\_linux is available and it contains the following two files:

- anyware-connector-offline-deps.tar.gz
- install.sh

The `install.sh` file contains the installation bash script and the `anyware-connector-offline-deps.tar.gz` file contains the RPM dependencies and Anyware Connector images.

### 4. Install Anyware Connector Offline

To install the Anyware Connector offline, run the installation script:

```
cd /PATH_OF_EXTRACTED_INSTALLATION_FILES  
sudo ./install.sh
```

Anyware Connector RPM is now installed.

#### Installation Errors

The Darksite installation could fail due to a package conflicting or checksum error if there is an existing package already present on the target machine. To resolve this, you need to delete the conflicting package and re-run the installation script.

### 5. Configure Anyware Connector

To install the remaining components and configure Anyware Connector, see [Configuring Anyware Connector](#)

# Reference

## Anyware Connector Features

### **Updating the Connector**

When updating Anyware Connector, it restarts to apply the updated configurations and all the active sessions going through the connector are disconnected. This requires users to log in again and reconnect. The only configuration update that does not require a Connector restart is running the diagnose command.

Flags mentioned in this section apply to the `configure` connector command, unless otherwise noted.

Items	Connector on RHEL/Rocky Linux
<b>Operating System</b>	RHEL/Rocky Linux 8.x
<b>Packaging</b>	RPM Package
<b>Connector Configuration</b>	Configuration files and/or command line flags and parameters.
<b>Required Configuration Flags</b>	<pre>--token --domain --sa-user and --sa-password, or --ad-sync=false --accept-policies --self-signed (or --tls-key and --tls-cert must be provided) --ldaps-ca-cert (or --ldaps-insecure or --enable-ldap-plaintext) --external-pcoip-ip 'public.ipv4.sg.ip'</pre>
<b>MFA Configuration</b>	All connection requests both internal and external will require MFA credentials to be entered.
<b>Federated User Authentication/SSO</b>	Supported
<b>Active Directory LDAPS Certificate</b>	<p>The Active Directory CA certificate must be provided to the installer by entering the information with the <code>--ldaps-ca-cert</code> parameter or by setting it in the configuration file.</p> <p>Skip the certificate validation when connecting to the Active Directory using the following flag <code>--ldaps-insecure</code>.</p> <p>For testing purposes the AD connection can use ldap in the plaintext form with the <code>--enable-ldap-plaintext</code> flag.</p>
<b>Key path and Certificate path flag</b>	<code>--tls-key</code> and <code>--tls-cert</code>
<b>Installation Commands</b>	<p>Add the repository and install the Connector RPM with the following command: <code>sudo dnf install -y anyware-connector</code>.</p> <p>Configure the Connector with flags or configuration files using the following command: <code>sudo /usr/local/bin/anyware-connector configure {flags or path to config file}</code>.</p> <p>The <code>configure</code> command fails with a missing parameter error if the mandatory flags or parameters are missing, the mandatory flags are <code>--token</code>, <code>--domain</code>, <code>--sa-user</code>, <code>--sa-password</code>, <code>--ldaps-ca-cert</code> (or must provide <code>ldaps-insecure</code> or <code>--enable-ldap-plaintext</code>), <code>--self-signed</code> (or must provide <code>--tls-key</code> and <code>--tls-cert</code>).</p>
<b>Update Configuration</b>	Update the configuration using: <code>sudo /usr/local/bin/anyware-connector configure {flags or path to config file} command</code> .

Items	Connector on RHEL/Rocky Linux
<b>Upgrade Commands</b>	<code>sudo dnf update anyware-connector</code> and <code>sudo /usr/local/bin/anyware-connector upgrade</code>
<b>Internal/External Session Detection</b>	In most cases the Connector on RHEL/Rocky Linux works without any special configuration, but if you know the Connector on RHEL/Rocky Linux is only for LAN connections, it is recommended to set the <code>--enable-security-gateway</code> flag to false by using <code>--enable-security-gateway=false</code> .
<b>Diagnose Commands</b>	The <i>diagnose</i> command has two flags: <code>--health</code> to check the overall status of Anyware Connector, and <code>--support-bundle</code> to generate a support bundle.



# Difference between Anyware Connector on Ubuntu and RHEL/Rocky Linux

Anyware Connector on RHEL/Rocky Linux stands out from the Connector on Ubuntu. Here is a feature comparison between two Connectors.

Items	Connector on Ubuntu	Connector on RHEL/Rocky Linux
<b>Operating System</b>	Ubuntu 18.04	RHEL/Rocky Linux 8, 9
<b>Packaging</b>	tar file	RPM Package
<b>Deployment</b>	Docker Swarm	Kubernetes
<b>Connector Configuration</b>	Configuration files and/or command line flags and parameters.	Configuration files and/or command line flags and parameters.
<b>Required Configuration Flags</b>	<pre>--token --domain --sa-user --sa-password --accept-policies --self-signed (or --ssl-key and --ssl-cert)</pre>	<pre>--token --domain --accept-policies --enable-ad-sync=false --ldaps-insecure (or --ldaps-ca-cert or --enable-ldap-plaintext) --external-pcoip-ip 'public.ipv4.sg.ip'</pre>
<b>MFA Configuration</b>	When MFA is enabled all connection requests from internal PCoIP Clients have MFA bypassed.	All connection requests both internal and external will require MFA credentials to be entered.
<b>Federated User Authentication/SSO</b>	Supported	Supported
<b>Active Directory LDAPS Certificate</b>	If <code>--ldaps-ca-cert</code> is not provided during installation, the Active Directory CA certificate is automatically collected by the Connector by connecting to each DC on the LDAPS port, and the certificate is saved to the Connectors CA certificate store.	<p>The Active Directory CA certificate must be provided to the installer by entering the information with the <code>--ldaps-ca-cert</code> parameter or by editing the configuration file.</p> <p>Skip the certificate validation when connecting to the Active Directory using the following flag <code>--ldaps-insecure</code>.</p> <p>For testing purposes the Active Directory connection can use ldap in the plaintext form with the <code>--enable-ldap-plaintext</code> flag.</p>
<b>Active Directory Service Accounts</b>	The Active Directory service account username and password is required.	The Active Directory service account is optional.
<b>Diagnose Commands</b>	You can diagnose remote workstation connectivity, and Active Directory connectivity	Diagnose commands has two flags <code>--health</code> to check the overall status of Anyware Connector and <code>--support-bundle</code> to generate support bundle.

Items	Connector on Ubuntu	Connector on RHEL/Rocky Linux
	by running the <code>diagnose</code> command.	
<b>Key path and Certificate path flag</b>	<code>--ssl-key</code> and <code>--ssl-cert</code>	<code>--tls-key</code> and <code>--tls-cert</code>
<b>Installation Commands</b>	<p>Download the installer from our website and extract the package, then run the install command with the required flags: <code>sudo /usr/sbin/cloud-access-connector install {flags}</code>.</p> <p>The installer prompts for mandatory flags if you do not provide them in the command.</p>	<p>Add the repository and install the Connector RPM with the following command:  <code>sudo dnf install -y anyware-connector</code>.</p> <p>Configure the Connector with flags or configuration files using the following command: <code>sudo /usr/local/bin/anyware-connector configure {flags or path to config file}</code>. The <code>configure</code> command will fail with a missing parameter error if the mandatory flags or parameters are missing, the mandatory flags are <code>--token</code>, <code>--domain</code>, <code>--sa-user</code>, <code>--sa-password</code>, <code>--self-signed</code> (or must provide <code>--tls-key</code> and <code>--tls-cert</code>), <code>--ldaps-ca-cert</code> (or <code>--ldaps-insecure</code> or <code>--enable-ldap-plaintext</code>).</p>
<b>Update Configuration</b>	<code>cd /usr/sbin sudo cloud-access-connector update {flags to be updated}</code>	Update the configuration using: <code>sudo /usr/local/bin/anyware-connector configure {flags or path to config file}</code> command.
<b>Upgrade Commands</b>	<code>cd /usr/sbin sudo cloud-access-connector update {flags to be updated}</code>	<code>sudo dnf update anyware-connector</code> and <code>sudo /usr/local/bin/anyware-connector upgrade</code>
<b>Internal/External Session Detection</b>	Typically the Connector on Ubuntu works without any special configuration, but in some cases you may need to explicitly set the <code>--internal-client-cidr</code> and <code>--external-client-cidr</code> so that sessions get treated correctly (eg, NATing external connections from a Firewall).	In most cases the Connector on RHEL/Rocky Linux works without any special configuration, but if you know the Connector on RHEL/Rocky Linux is only for LAN connections, it is recommended to set the <code>--enable-security-gateway</code> flag to false by using <code>--enable-security-gateway=false</code>
<b>Connector Cluster Network</b>	<code>--connector-network-cidr</code> is the CIDR flag to use for	There are three flags to use for the Connector's network. They are; <code>--cluster-cidr</code> to set

Items	Connector on Ubuntu	Connector on RHEL/Rocky Linux
	the Connector's docker network. The default docker network subnet is 10.101.0.0/16.	cluster CIDR,default is 10.42.0.0/16, <code>--service-cidr</code> to set service CIDR, default is 10.43.0.0/16. and <code>--cluster-dns</code> to set cluster dns ip address, default is 10.43.0.10, it has to be part of of the service-cidr.

# Transitioning Anyware Connectors

It is not possible to migrate directly from the Anyware Connector on Ubuntu to the Anyware Connector on RHEL/Rocky Linux as they run on different operating systems. The best method to transition to the Connector on RHEL/Rocky Linux is to create this new Connector using the same Anyware Manager deployment you used for the Connector on Ubuntu.

## Minimize Transition Downtime

The Connector on RHEL/Rocky Linux can co-exist with the Connector on Ubuntu in the same Anyware Manager deployment. To minimize downtime, it is recommended that you run the newly created RHEL/Rocky Linux Connector for a period of time to ensure it is working properly before decommissioning the Connector on Ubuntu.

## MFA Configuration

When you enable MFA for the Connector for RHEL/Rocky Linux, all PCoIP Clients authenticated through the Connector will be prompted to enter MFA credentials. Previously with the Connector on Ubuntu, internal and external clients had different MFA configurations. If you want to have the same MFA configuration for the Connector on RHEL/Rocky Linux as the Connector on Ubuntu, you must install multiple Connectors on RHEL/Rocky Linux.

## Transitioning the Connector

The following steps outline how to transition to the Connector on RHEL/Rocky Linux:

1. Before you install the RHEL Connector, ensure you have met all the required prerequisite steps. For instructions and documentation on the Connector prerequisite steps when installing the RHEL Connector, see [Connector System Requirement](#). It is important to read and address all the prerequisites outlined.
2. Review the differences between the Connector on RHEL/Rocky Linux and Ubuntu, as outlined [here](#).
3. Prepare your Connector configuration files. For information on configuring your Connector, see [Configuring the Connector](#).

4. When generating the Anyware Connector token for the new Connector, ensure you use the same Anyware Manager deployment as the existing Connector on Ubuntu. For information on generating the Connector token, see [Generating a Connector Token](#).
5. Install and configure the Connector on RHEL/Rocky Linux. For information on installing and configuring the Connector, see [Installing the Connector](#).

## Testing the Connector

Once you have installed the Connector on RHEL/Rocky Linux you should test it and ensure it has been correctly installed and configured. The following steps outline how to test the Connector:

1. Run the following command to check that all installed services are running:

```
sudo /usr/local/bin/kubectl get pods -c connector
```

2. Access the virtual machine where you have the PCoIP Client installed and establish a connection using the Connector IP or FQDN. If this connection is successful then it confirms the Connector has been installed correctly. If it is not, you should re-check the virtual machine configuration and Connector configuration.

## Adding the Connector to a Load Balancer

If the original Connector on Ubuntu was configured with a load balancer, you need to add the new Connector on RHEL/Rocky Linux to the load balancer. Once you have tested the Connector on RHEL/Rocky Linux install, and are happy that it was successful, you can remove the Connector on Ubuntu.

# Proxy Configuration for Anyware Connector

## Disable Proxy Configuration

Once all installations and configurations are done, Anyware Connector could run into some issues related to proxy. To avoid this, the proxy must be disabled on the Anyware Connector workstations. To disable proxy:

1. Check the proxy settings saved in the cluster:

```
sudo /usr/local/bin/kubectl edit secret proxysettings -o yaml -n connector
```

2. Look for strings such as `HTTP_PROXY`, `HTTPS_PROXY`, `NO_PROXY`.
3. Press the letter "i" to edit the keys to set them to empty string by doing the following:

```
HTTP_PROXY: ""  
HTTPS_PROXY: ""  
NO_PROXY: ""
```

if you do not see one of those strings change the rest to empty string "".

4. Save the new configuration by pressing "ESC" then ":" then "wq".

# Scaling and PColP Session Limits

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

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

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

## Estimated Bandwidth

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

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



# Firewall and Load Balancing Considerations

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

## Ports and Component Connections

Component	Allow	Port/Protocol	Source/ Destination Component	Descriptions
Connector	Inbound	443 TCP	From PCoIP Clients and administrative web browsers.	For users to negotiate connections to their remote workstations. For accessing the Management Interface for (legacy) management of Anyware Manager.
Connector	Outbound	443 TCP	To CAM Service, <a href="#">PCoIP Cloud License Server</a> and to <a href="#">SumoLogic</a> .	To sync AD information to the CAM service and call Anyware Manager APIs related to negotiating PCoIP sessions. To verify license activation code during the Connector installation. For log aggregation for support purposes.
Connector	Outbound	60443 TCP	To remote workstations.	Prepares PCoIP Agents for a new user session.
Connector	Inbound	4172 TCP/UDP	From PCoIP Clients.	For PCoIP Sessions with users that are outside of the corporate network.
Connector	Outbound	4172 TCP/UDP	To remote workstations.	For PCoIP Sessions with users that are outside of the corporate network.
Connector	Outbound	636 TCP	To Domain Controllers.	To authenticate users, and query user and computer information.
Connector	Outbound	1812 UDP (This port is configurable)	To RADIUS Server.	For authentication against RADIUS Server.
Connector	Outbound	53 TCP/UDP	To DNS.	Domain name resolution.
PCoIP License Server	Inbound	7070 TCP (This port is configurable)	From remote workstations.	For license activation and verification from PCoIP Agent if the PCoIP License Server is used instead of the Cloud License Server.

## PORT AND COMPONENT NOTES:

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

## HEALTH CHECK ENDPOINT

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

```
/health
```

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

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

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

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

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

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

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

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

# Configuring the Active Directory for Anyware Connector

We recommend having a single Active Directory configuration for a single deployment, that means all Connectors within that deployment should be configured to the same Active Directory. If you want to have multiple Connectors with different Active Directory settings then you need to ensure that each Connector belongs to a separate deployment. If you create two Connectors that are associated with the same deployment then both will use the same Active Directory sync settings, and the configuration of the last Connector created will take precedence.

## Configuring User and Computer Active Directory Distinguished Names

The Connector can optionally be configured to use specific Distinguished Names (DNs) when querying Active Directory for users and computers. This has been extended to be available when running the `update` command in addition to the `install` command.

The following is an example of the DN string format: `CN=Manager Admins,CN=Users,DC=example,DC=com`. You can also configure the frequency at which the Connector syncs this data with the Manager service, as outlined in the following table:

Flag	Type	Description
<code>--users-dn</code>	String	The base DN to search for users within Active Directory. This option may be specified multiple times to provide multiple DNs.
<code>--computers-dn</code>	String	The base DN to search for computers within Active Directory. This option may be specified multiple times to provide multiple DNs.
<code>--sync-interval</code>	String	The interval time in minutes for how often to sync Active Directory users and computers with the Manager service. It must be at least five minutes.
<code>--users-filter</code>	String	The filter to search for users within Active Directory. Specify multiple filters with multiple options. Default user filter: <code>(&amp;(objectCategory=person)(objectClass=user))</code> . An example for a user group filter: <code>(&amp;(objectCategory=person)(objectClass=user)(memberOf: 1.2.840.113556.1.4.1941:=CN=PCoIP Users Group,CN=Users,DC=example,DC=com))</code> .
<code>--computers-filter</code>	String	The filter to search for computers within Active Directory. Specify multiple filters with multiple options. Default computer filter: <code>(&amp;(primaryGroupID=515)(objectCategory=computer))</code> .

These flags outlined are optional and may be provided with the `install` or `update` commands. If you are updating a Connector you only need to provide these flags if you want to changing the DN settings associated with that Connector. If you do not add these flags when performing an update then the Connector will retain the same settings.

You can reset user or computer DNs to their default values by providing an explicit DN with a wider scope than the original DN used.

## Configuring Active Directory Pool Groups

A set of command line flags enables users to update Active Directory pool groups. These flags apply changes to the Active Directory settings of the Connector.

By providing the following flags the appropriate update gets applied to the Connector settings. If no command-line option is provided, the Connector will display all available options for this operation.

Flag	Type	Description
<code>--manager-insecure</code>	String	Skips certificate validation when connecting to Anyware Manager as a Service. This option should only be used when connecting to Anyware Manager as a Service deployed with self-signed certificates.
<code>--add-pool-group</code>	String	Adds specified Active Directory group to the existing pool group settings. By providing all the existing pools groups in the Connector, settings would get replaced by the user specified ones.
<code>--remove-pool-group</code>	String	Removes specified pool Active Directory group by its DN. <b>This flag is not supported by the Connector on Rocky Linux/RHEL.</b>
<code>--clear-pools-groups</code>	String	Clears all pools Active Directory groups. This operation is exclusive and cannot be combined with <code>--remove-pool-group</code> or <code>--add-pool-group</code> . <b>This flag is not supported by the Connector on Rocky Linux/RHEL</b>
<code>--get-cam-settings</code>	String	Prints all Anyware Manager as a Service settings to Admin console.

# Security

## Anyware Connector Multi-Factor Authentication

### Duo Authentication

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

#### Duo Authentication Version

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

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

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

For further information on the above integration, see [RADIUS Duo Only](#).

# Azure MFA Authentication

## Microsoft Azure MFA Component Versions

The Connector was tested with Microsoft Azure MFA on **November 15<sup>th</sup> 2019** with the following components:

HP component versions:

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

3<sup>rd</sup> party component versions:

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

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

## Azure MFA Configuration

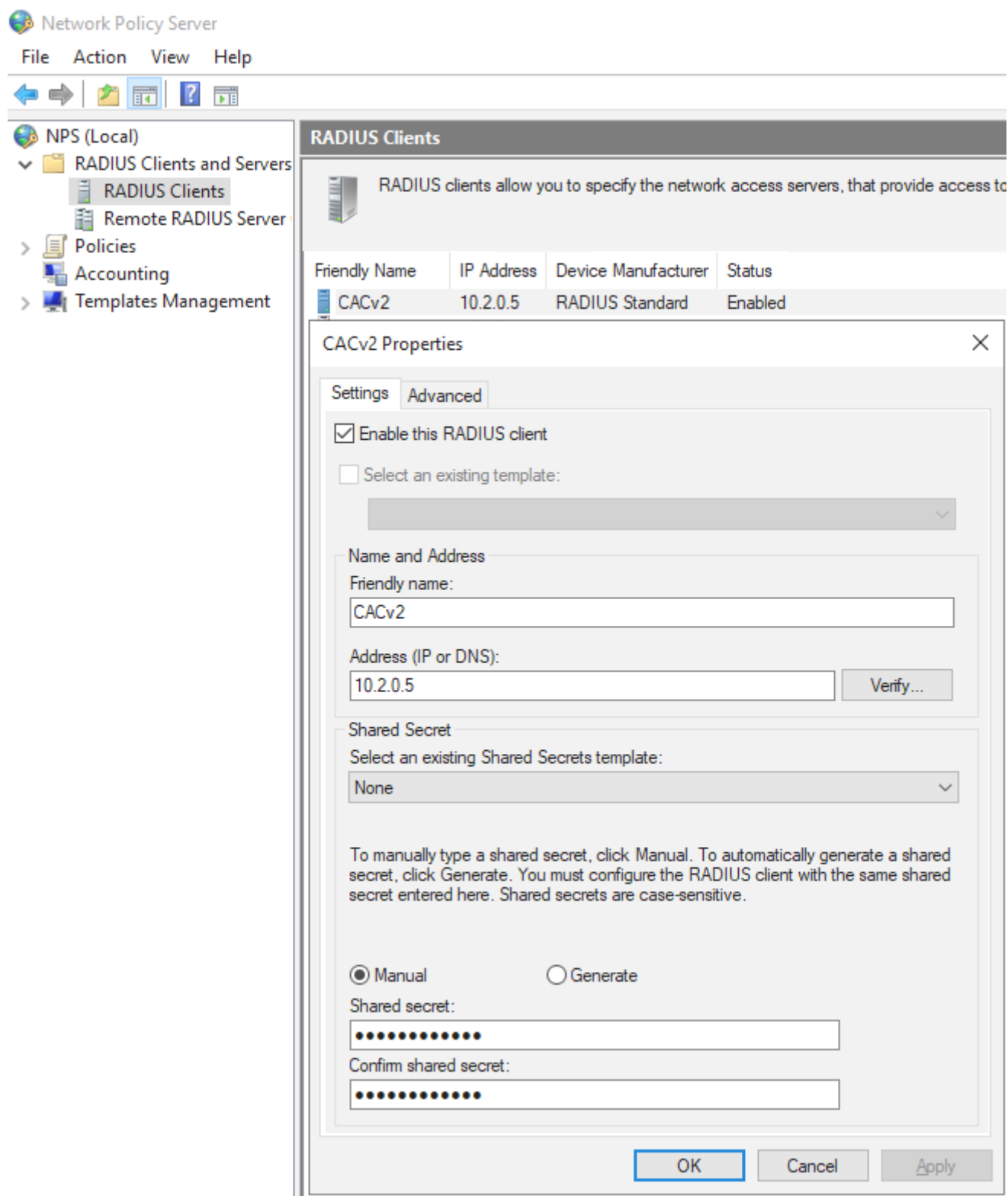
If you wish to use Azure MFA with the Connector you need to configure a number of 3<sup>rd</sup> party components. The following steps outline this process:

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



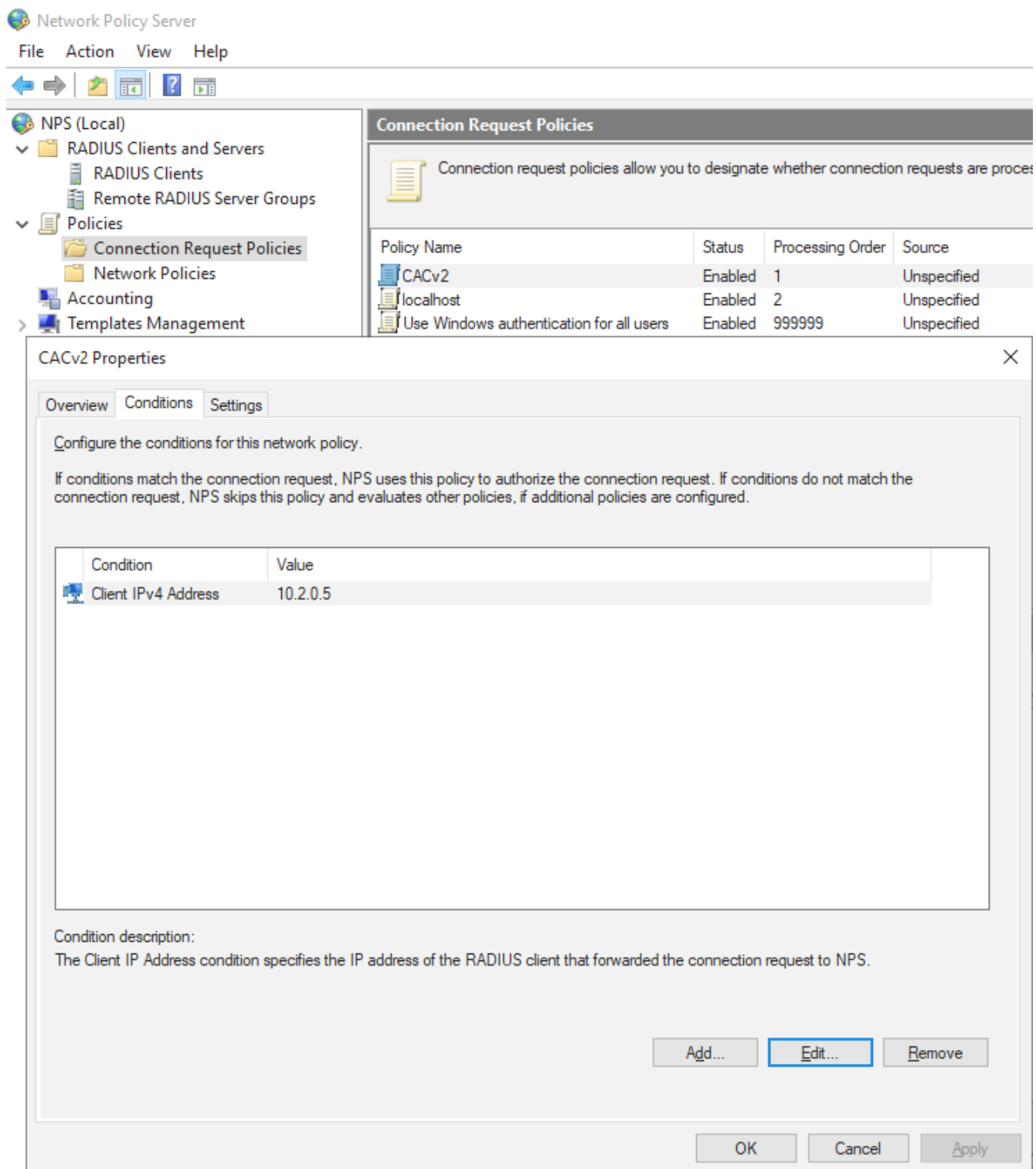
Once you have registered the NPS you need to configure the server. The following steps outline the NPS configuration:

1. From within the NPS console click **RADIUS Clients**.
2. Add the Connector IP address and Shared Secret and click **OK**.



3. Click **Policies > Connection Request Policies** and add a new policy name and click **OK**.

4. From the **Conditions** tab add the Client IPv4 Address of the Connector and click **OK**.



5. From the **Settings** tab under **Authentication** click *Accept users without validating credentials*.

6. Restart NPS services to enable these changes to take effect.

# Third-Party Multi-Factor Authentication

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

## Third-Party MFA Information

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

- [Anyware Software - Okta MFA Integration in GCP](#)

# Troubleshooting

## Anyware Connector Support Bundle on RHEL/Rocky Linux

If you encounter an issue with the Anyware Connector on RHEL/Rocky Linux, it is possible to generate a support bundle that can be sent to the HP support team to investigate and resolve.

To generate the support bundle, run the following command:

```
sudo /usr/local/bin/anyware-connector diagnose --support-bundle
```

When this command is successful, a `tar.gz` file is located under the `/tmp/` directory with a name formatted as follows:

**`/tmp/anyware-connector-support-bundle-(date)-(time).tar.gz`**

## Support Bundle Information and Logs

The support bundle collects the various information from the system and then zip the files into a `.tar.gz` file in the `/tmp` directory.

Once you unzip the file the structure is as follows:

- **files/etc** folder contains files with OS level information:
  - **issue** file contains a copy of all contents from the `/etc/issue` file.
  - **os-release** file contains all operating system identification data that was found in the `/usr/lib/os-release` file.
  - **\*\*systemd/system/k3s.service\*\*** folder contains the information about `k3s.service` configuration.
- **files/var/log/anyware-connector** folder collects all of the Anyware Connector log files, for example generate, configure, diagnose, install logs.

The **out** folder collects outputs from running various commands to expose the details of relevant system information and Anyware Connector backend services, as outlined below:

- **os** folder contains the following files:
  - **dmesg.out** file contains the output of command `dmesg`.
  - **ls\_l@var@crash.out** file contains the output of the command `ls -l /var/crash`.
  - **pgrep\_l\_k3s.out** file contains the output of the command `pgrep -l k3s`.
  - **ps\_wwauxfx.out** file contains the output of the command `ps wwauxfZ`.
  - **ss-aux.out** file contains the output of the command `ss -ax`.
  - **who.out** file contains the output of the command `who`.
  - **dnf\_list--installed--disablerepo=\*.out** file contains the packages that are currently installed on the system and their versions.
- The **firewall** folder contains the outputs of the command `firewall-cmd --list-services`.
- The **network** folder displays the following network files:
  - **netstat\_Wnap.out** file contains the output of the command `netstat -Wnap`.
  - **ip\_add.out** file contains the output of the command `ip add`.
  - **selinux** folder contains the following files:
    - **semodule\_l.out** file contains the output of the command `semodule -l`.
    - **sestatus.out** file contains the output of the command `sestatus`.
- The **deployments** folder contains the following files:
  - **kubectl\_get\_deployment.out** file is the output of the command `kubectl get deployments` that lists the status of all deployments for Anyware Connector services.
  - The description of the deployment for each of the deployments through the output of the command `kubectl describe`.
  - The deployments information is retrieved from the following namespaces: `kube-system`, `kube-public`, `kube-node-lease`, `connector`, `logging`, and `ingress-nginx`.
- The **pods** folder contains the following files:
  - **kubectl\_get\_pods.out** file is the output of the command `get pods` that lists the status of all pods for Anyware Connector services.
  - The description of a pod for each of the pods through the output of the command `kubectl describe pod`

- The pod information is retrieved from the following namespaces: `kube-system`, `kube-public`, `kube-node-lease`, `connector`, `logging`, and `ingress-nginx`.
- **logs** folder contains the following files:
  - The log files for each pod.
  - The logs files are retrieved from the following namespaces: `kube-system`, `kube-public`, `kube-node-lease`, `connector`, `logging`, and `ingress-nginx`.
- The **services** folder contains the following files:
  - **kubectrl\_get\_services.out** file is the output of the command `get services` that lists the status of all services for Anyware Connector services.
  - The description of a service for each of the services through the output of the command `kubectrl describe services`
  - The services information is retrieved from the following namespaces: `kube-system`, `kube-public`, `kube-node-lease`, `connector`, `logging`, and `ingress-nginx`.
- **secrets** folder displays the output of the command `kubectrl get secrets` and displays the following files:
  - **kubectrl\_get\_secrets.out** file is the output of the command `get secrets` that lists the status of all services for Anyware Connector services.
  - The description of a secrets through the output of the command `kubectrl describe secrets`.
  - The secrets information is retrieved from the following namespaces: `kube-system`, `kube-public`, `kube-node-lease`, `connector`, `logging`, and `ingress-nginx`.

# Anyware Connector Health Status on RHEL/Rocky Linux

In the case that there is an issue with the Connector on RHEL/Rocky Linux, the diagnose health command will provide an overview of Connector's health. The following command will provide a list of services that are in healthy and unhealthy states. The command will enable you to determine the services that are unhealthy and run more specific diagnosis on the unhealthy service:

```
sudo /usr/local/bin/anyware-connector diagnose --health
```

The diagnose command lists all Connector service, these services for the Connector are listed below. If any of these services are in an unhealthy state, the overall health status will be unhealthy:

- "adsync"
- "broker"
- "cm"
- "cmsg"
- "sg"
- "healthcheck"
- "rwtelemetry"

 **Running Diagnose command does not impact the active sessions.**

# Network Connectivity Issues

The Anyware Connector provides some diagnostic checks that can be used to troubleshoot the cause of issues you may be experiencing. Run the following command:

```
sudo /usr/local/bin/anyware-connector diagnose
```

This command can also be used to verify that your Connector has been correctly configured. The diagnostic checks cover Remote Workstation connectivity and Active Directory connectivity.

The following table lists the flags associated with this command:

Flag	Description
<code>--rw</code>	The Remote Workstation FQDN
<code>--ad</code>	Verify connectivity to currently configured Active Directory server
<code>-h --help</code>	help for diagnose
<code>--debug</code>	This flag can be run if you initial install of the Connector fails. It provides a detailed output of the Connector installation. This is useful for self-troubleshooting or to provide to the HP support team when logging a support ticket.

## Common Installation Issues with the Connector

For information on issues relating to failed Connector installations, We have a KB article that details troubleshooting steps for common issues related to installing the Connector, see [here](#).

## Remote Workstation Connectivity Check

This command will attempt to connect to the specified remote workstation on the ports required for establishing a PCoIP session. It checks to ensure that the PCoIP agent is running on the remote workstation.



Example command to diagnose remote workstation connectivity issues:

```
sudo /usr/local/bin/anyware-connector diagnose --rw fqdn.of.my.rw
```

### Check Passes

- Your Connector is able to resolve the FQDN of the remote workstation and connect to it.
- The PColP agent is running and responding on the remote workstation.

### Check Fails

If the check fails it may be as a result of one or more of the following issues:

- Firewall or network routing rules or restrictions may be in place.
- A failure has occurred and the FQDN of the remote workstation cannot be resolved.
- The PColP agent on the remote workstation is not running or is unresponsive.
- There may be an issue with the DNS name. For more information on this, and how to potentially resolve this issue, see [DNS Name Resolution Configuration on RHEL/Rocky Linux](#).

## Active Directory Connectivity Check

This command will attempt to connect to the Active Directory domain controller that was provided during installation using those same credentials.

Example command to diagnose Active Directory connectivity issues:

```
sudo /usr/local/bin/anyware-connector diagnose --ad
```

### Check Passes

- The Connector is able to resolve the FQDN of the domain controller and authenticate to it.

### Check Fails

If the check fails it may be as a result of one or more of the following issues:

- Firewall or network routing rules or restrictions may be in place.
- A failure has occurred and the FQDN of the domain controller cannot be resolved.
- The Active Directory server may be unresponsive.

- The check was unable to authenticate to the Active Directory server.
- There may be an issue with the DNS name. For more information on this, and how to potentially resolve this issue, see [DNS Name Resolution Configuration on RHEL/Rocky Linux](#).

# Configuring DNS Name Resolution

In order to install and configure Anyware Manager or Connector on the RHEL or Rocky Linux machine, it's important to ensure that there is a solid connection between the machine and the Active Directory Domain Controller. You need to ensure that you can route from this machine to the Domain Controller and that there is nothing to prevent port 443 (https) and port 636 (LDAPS) connecting between the two systems.

The following steps are to ensure DNS settings are configured properly on the machine for Anyware Manager or Connector to operate. The sample IP of the Domain Controller is `10.162.0.42` for `example-domain.com`:

1. Disable auto-configuration of DNS settings in order to prevent setting being overwritten on reboot. In this example the device name is `eth0`:

```
nmcli device modify eth0 ipv4.ignore-auto-dns yes
```

You may also need to disable this on the connection level in some cases. In this example the connection name is `eth0`:

```
nmcli connection modify eth0 ipv4.ignore-auto-dns yes
```

2. Edit the Network Configuration scripts. Add the `DNS1` for the IP address for Active Directory's DNS server (typically the Domain Controller itself) and optionally `DNS2` for the fallback DNS server. You can optionally add `DOMAIN` for a DNS suffix (typically the Domain name):

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=ens192
UUID=dfe16427-21f1-429c-99cb-a1e9b42be181
DEVICE=ens192
ONBOOT=yes
DNS1=10.162.0.42
DOMAIN=example-domain.com
PEERDNS=no
```

3. Run the following command to restart the Network Manager:

```
sudo systemctl restart NetworkManager
```

4. Check the `/etc/resolv.conf` file to make sure that the desired DNS servers and search suffixes are there. It is important the the `nameservers` are the AD DNS or else the machine will fail to connect to the Domain Controller(s):

```
cat /etc/resolv.conf
# Generated by NetworkManager
search example-domain.com
nameserver 10.162.0.42
```

5. Test DNS by pinging the domain, in this example `example-domain.com` is the domain name:

```
ping example-domain.com
```

6. If the response is successful, you should receive a message similar to the example below:

```
PING example-domain.com (10.162.0.42): 56 data bytes
64 bytes from 10.162.0.42: icmp_seq=0 ttl=118 time=16.622 ms
64 bytes from 10.162.0.42: icmp_seq=1 ttl=118 time=50.675 ms
64 bytes from 10.162.0.42: icmp_seq=2 ttl=118 time=27.682 ms
64 bytes from 10.162.0.42: icmp_seq=3 ttl=118 time=19.886 ms
^C
--- example-domain.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
```

7. Reboot the machine and check that the DNS settings in `/etc/resolv.conf` persist and that you can still ping the domain as done in steps 4-6.

#### Applying Host Machines DNS settings to K3S

The host machine's DNS settings are copied from `/etc/resolv.conf` and applied to the Anyware Manager and/or CAS Connector when the CAS K3S service starts. Because of this it is important that settings are correct on boot. You will need to either reboot the machine or restart the K3S service to apply the DNS settings to the Anyware Manager or CAS Connector if changes are made post installation or configuration.

It is best to ensure DNS settings are correct before installing and CAS software on the machine.

# Getting Support

If you are having trouble, help is available. This section contains information about contacting HP support and connecting with the HP user community.

## Contacting Support

If you encounter problems installing or using HP technology, you can:

- Browse the [HP Knowledge Base](#).
- Submit a [Support Ticket](#).

## The HP Community Forum

The PColP Community Forum allows users to have conversations with other IT professionals to learn how they resolved issues, find answers to common questions, have peer group discussions on various topics, and access the HP PColP Technical Support Service team. Our staff is heavily involved in the forums.

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

# Release Notes

To view the latest release notes for Anyware Connector, see [Anyware Connector Release Notes](#).