Table of contents

Chapter 15. Configuring custom SSL/TLS certificates

You can configure the undercloud to use SSL/TLS for communication over public endpoints. However, if want to you use a SSL certificate with your own certificate authority, you must complete the following configuration steps.

Back to top

15.1. Initializing the signing host

The signing host is the host that generates and signs new certificates with a certificate authority. If you have never created SSL certificates on the chosen signing host, you might need to initialize the host so that it can sign new certificates.

Procedure

1. The `/etc/pki/CA/index.txt` file contains records of all signed certificates. Check if this file exists. If it does not exist, create an empty file:

   ```
   $ sudo touch /etc/pki/CA/index.txt
   ```

2. The `/etc/pki/CA/serial` file identifies the next serial number to use for the next certificate to sign. Check if this file exists. If the file does not exist, create a new file with a new starting value:

Back to top

```
$ echo '1000' | sudo tee /etc/pki/CA/serial
```

15.2. Creating a certificate authority

Normally you sign your SSL/TLS certificates with an external certificate authority. In some situations, you might want to use your own certificate authority. For example, you might want to have an internal-only certificate authority.

Procedure

1. Generate a key and certificate pair to act as the certificate authority:

```
$ openssl genrsa -out ca.key.pem 4096
$ openssl req  -key ca.key.pem -new -x509 -days 7300 -extensions
```

Back to top

```
    v3_ca -out ca.crt.pem
```

1. The `openssl req` command asks for certain details about your authority. Enter these details at the prompt.

These commands create a certificate authority file called `ca.crt.pem`.

15.3. Adding the certificate authority to clients

For any external clients aiming to communicate using SSL/TLS, copy the certificate authority file to each client that requires access to your Red Hat OpenStack Platform environment.

Procedure

Back to top

1. Copy the certificate authority to the client system:

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
```

2. After you copy the certificate authority file to each client, run the following command on each client to add the certificate to the certificate authority trust bundle:

```
$ sudo update-ca-trust extract
```

15.4. Creating an SSL/TLS key

Back to top

Enabling SSL/TLS on an OpenStack environment requires an SSL/TLS key to generate your certificates. This procedure shows how to generate this key.

Procedure

1. Run the following command to generate the SSL/TLS key (`server.key.pem`):

```
$ openssl genrsa -out server.key.pem 2048
```

15.5. Creating an SSL/TLS certificate signing request

Complete the following procedure to create a certificate signing request.

Back to top

Procedure

1. Copy the default OpenSSL configuration file:

```
$ cp /etc/pki/tls/openssl.cnf .
```

2. Edit the new `openssl.cnf` file and configure the SSL parameters to use for the director. An example of the types of parameters to modify include:

```
[req]
distinguished_name = req_distinguished_name
req_extensions = v3_req
```

Show more

Back to top

Set the `commonName_default` to one of the following entries:

- If using an IP address to access the director over SSL/TLS, use the `undercloud_public_host` parameter in `undercloud.conf`.
- If using a fully qualified domain name to access the director over SSL/TLS, use the domain name.

Add `subjectAltName = @alt_names` to the `v3_req` section.

Edit the `alt_names` section to include the following entries:

- `IP` - A list of IP addresses that clients use to access the director over SSL.
- `DNS` - A list of domain names that clients use to access the director over SSL. Also include the Public API IP address as a DNS entry at the end of the `alt_names` section.

ⓘ Note

Back to top

For more information about `openssl.cnf`, run the `man openssl.cnf` command.

3. Run the following command to generate a certificate signing request (`server.csr.pem`):

```
$ openssl req -config openssl.cnf -key server.key.pem -new -out server.csr.pem
```

Ensure that you include your OpenStack SSL/TLS key with the `-key` option.

This command results in an `server.csr.pem` file, which is the certificate signing request. Use this file to create your OpenStack SSL/TLS certificate.

Back to top

15.6. Creating the SSL/TLS certificate

This procedure shows how to generate the certificate for your OpenStack environment. This requires the following files:

`openssl.cnf`

The customized configuration file specifying the v3 extensions.

`server.csr.pem`

The certificate signing request to generate and sign the certificate with a certificate authority.

`ca.crt.pem`

The certificate authority, which signs the certificate.

`ca.key.pem`

The certificate authority private key.

Back to top

Procedure

1. Run the following command to create a certificate for your undercloud or overcloud:

   ```
   $ sudo openssl ca -config openssl.cnf -extensions v3_req -days
   3650 -in server.csr.pem -out server.crt.pem -cert ca.crt.pem -
   keyfile ca.key.pem
   ```

   This command uses the following options:

   `-config`

   Use a custom configuration file, which is our `openssl.cnf` file with v3 extensions.

   `-extensions v3_req`

   Enabled v3 extensions.

Back to top

-days

Defines how long in days until the certificate expires.

-in '

The certificate signing request.

-out

The resulting signed certificate.

-cert

The certificate authority file.

-keyfile

The certificate authority private key.

Back to top

This command creates a new certificate named `server.crt.pem`. Use this certificate in conjunction with your OpenStack SSL/TLS key

15.7. Adding the certificate to the undercloud

Complete the following steps to add your OpenStack SSL/TLS certificate to the undercloud trust bundle.

Procedure

1. Run the following command to combine the certificate and key:

   ```
   $ cat server.crt.pem server.key.pem > undercloud.pem
   ```

   This command creates a `undercloud.pem` file.

Back to top

2. Copy the `undercloud.pem` file to a location within your `/etc/pki` directory and set the necessary SELinux context so that HAProxy can read it:

```
$ sudo mkdir /etc/pki/undercloud-certs
$ sudo cp ~/undercloud.pem /etc/pki/undercloud-certs/.
$ sudo semanage fcontext -a -t etc_t "/etc/pki/undercloud-
certs(/.*)?"
```

Show more

3. Add the `undercloud.pem` file location to the `undercloud_service_certificate` option in the `undercloud.conf` file:

```
undercloud_service_certificate = /etc/pki/undercloud-
certs/undercloud.pem
```

Back to top

4. Ensure you add the certificate authority that signed the certificate to the undercloud’s list of trusted Certificate Authorities so that different services within the undercloud have access to the certificate authority:

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
$ sudo update-ca-trust extract
```

Continue installing the undercloud.

Previous

Next

Back to top

We help Red Hat users innovate and achieve their goals with our products and services with content they can trust. Explore our recent updates.

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. For more details, see the Red Hat Blog.

About Red Hat

We deliver hardened solutions that make it easier for enterprises to work across platforms and environments, from the core datacenter to the network edge.

Theme

System default

About Red Hat

Back to top

Jobs

Events

Locations

Contact Red Hat

Red Hat Blog

Inclusion at Red Hat

Cool Stuff Store

Red Hat Summit

Privacy statement

Terms of use

All policies and guidelines

Digital accessibility

Cookie preferences

Back to top

Back to top