Red Hat Satellite 6.8

Configuring Capsules with a Load Balancer

Distributing load between Capsule Servers

Last Updated: 2022-08-05

Red Hat Satellite 6.8 Configuring Capsules with a Load Balancer

Distributing load between Capsule Servers

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

Legal Notice

Abstract

This guide describes how to configure Red Hat Satellite to use a load balancer to distribute load between Capsule Servers.

Table of Contents

CHAPTER 1. LOAD BALANCING SOLUTION ARCHITECTURE

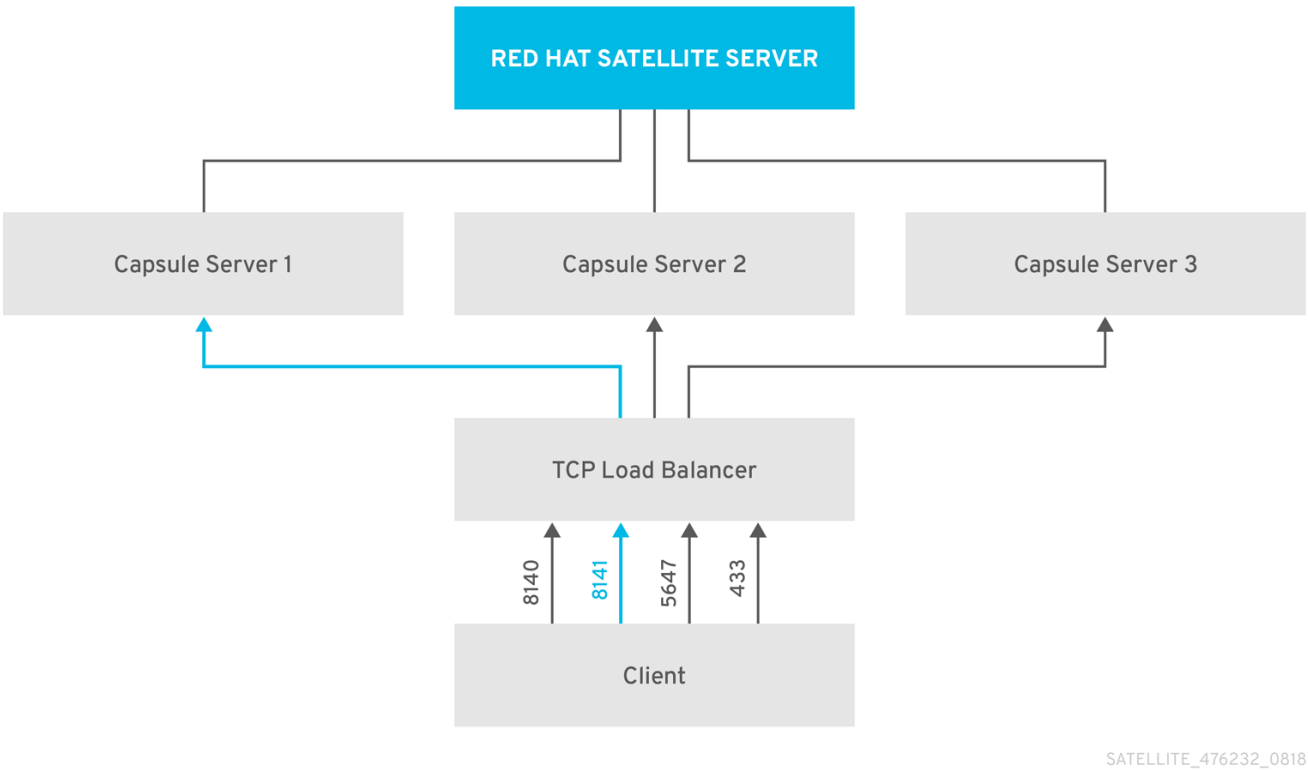
You can configure Satellite Server to use a load balancer to distribute client requests and network load across multiple Capsule Servers. This results in an overall performance improvement on Capsule Servers.

This guide outlines how to prepare Satellite Server and Capsule Server for load balancing, and provides guidelines on how to configure a load balancer and register clients in a load-balanced setup.

A load-balanced setup consists of the following components:

- Satellite Server
- Two or more Capsule Servers
- A load balancer
- Multiple clients

Figure 1.1. Satellite Load Balancing Solution Architecture



In a load-balanced setup, nearly all Capsule functionality continues to work as expected when one Capsule Server is down, for planned or unplanned maintenance. Load balancer works with the following services and features:

- Registration using **subscription-manager**
- Content Management with **yum** repositories
- Optional: Puppet

NOTE

In the load-balanced setup, a load balancer distributes load only for the services and features mentioned above. If other services, such as provisioning or virt-who, are running on the individual Capsules, you must access them directly through Capsules and not through the load balancer.

Managing Puppet Limitations

Puppet Certificate Authority (CA) management does not support certificate signing in a load-balanced setup. Puppet CA stores certificate information, such as the serial number counter and CRL, on the file system. Multiple writer processes that attempt to use the same data can corrupt it.

To manage this Puppet limitation, complete the following steps:

1. Configure Puppet certificate signing on one Capsule Server, typically the first system where you configure Capsule Server for load balancing.

2. Configure the clients to send CA requests to port 8141 on a load balancer.

3. Configure a load balancer to redirect CA requests from port 8141 to port 8140 on the system where you configure Capsule Server to sign Puppet certificates.

CHAPTER 2. LOAD BALANCING CONSIDERATIONS

Distributing load between several Capsule Servers prevents any one Capsule from becoming a single point of failure. Configuring Capsules to use a load balancer can provide resilience against planned and unplanned outages. This improves availability and responsiveness.

Consider the following guidelines when configuring load balancing:

- If you use Puppet, Puppet certificate signing is assigned to the first Capsule that you configure. If the first Capsule is down, clients cannot obtain Puppet content.
- This solution does not use Pacemaker or other similar HA tools to maintain one state across all Capsules. To troubleshoot issues, reproduce the issue on each Capsule, bypassing the load balancer.

Additional Maintenance Required for Load Balancing

Configuring Capsules to use a load balancer results in a more complex environment and requires additional maintenance.

The following additional steps are required for load balancing:

- You must ensure that all Capsules have the same Content Views and synchronize all Capsules to the same Content View versions
- You must upgrade each Capsule in sequence
- You must backup each Capsule that you configure regularly

Upgrading Capsule Servers in a Load Balancing Configuration

To upgrade Capsule Servers from 6.7 to 6.8, complete the Upgrading Capsule Servers procedure in *Upgrading and Updating Red Hat Satellite*. There are no additional steps required for Capsule Servers in a load balancing configuration.

CHAPTER 3. PREREQUISITES FOR CONFIGURING CAPSULE SERVERS FOR LOAD BALANCING

To configure Capsule Servers for load balancing, complete the following procedures described in *Installing Capsule Server*. Satellite does not support configuring existing Capsule Servers for load balancing.

1. Registering Capsule Server to Satellite Server
2. Attaching the Satellite Infrastructure Subscription
3. Configuring Repositories
4. Synchronizing the System Clock With chronyd
5. Installing Capsule Server Packages

CHAPTER 4. CONFIGURING CAPSULE SERVERS FOR LOAD BALANCING

This chapter outlines how to configure Capsule Servers for load balancing. Proceed to one of the following sections depending on your Satellite Server configuration:

1. Section 4.1, “Configuring Capsule Server with Default SSL Certificates for Load Balancing without Puppet”
2. Section 4.2, “Configuring Capsule Server with Default SSL Certificates for Load Balancing with Puppet”
3. Section 4.3, “Configuring Capsule Server with Custom SSL Certificates for Load Balancing without Puppet”
4. Section 4.4, “Configuring Capsule Server with Custom SSL Certificates for Load Balancing with Puppet”

Use different file names for the Katello certificates you create for each Capsule Server. For example, name the certificate archive file with the Capsule Server FQDN.

4.1. CONFIGURING CAPSULE SERVER WITH DEFAULT SSL CERTIFICATES FOR LOAD BALANCING WITHOUT PUPPET

The following section describes how to configure Capsule Servers that use default SSL certificates for load balancing without Puppet.

Complete this procedure on each Capsule Server that you want to configure for load balancing.

Procedure

1. On Satellite Server, generate Katello certificates for Capsule Server, for example:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule.example.com \
--certs-tar "/root/capsule.example.com-certs.tar" \
--foreman-proxy-cname loadbalancer.example.com
```

   Retain a copy of the example **satellite-installer** command that is output by the **capsule-certs-generate** command for installing the Capsule Server certificate.

2. Copy the certificate archive file from Satellite Server to Capsule Server.

```
# scp /root/capsule.example.com-certs.tar \
root@capsule.example.com:capsule.example.com-certs.tar
```

3. Append the following options to the **satellite-installer** command that you obtain from the output of the **capsule-certs-generate** command. Set the **--puppet-ca-server** option to point to the Capsule Server where you enter the command. You must install Puppet CA on your Capsule Servers, regardless of whether you intend to use it or not. Puppet is configured in its default single-node configuration.

```
--certs-cname                "loadbalancer.example.com" \
--puppet-dns-alt-names           "loadbalancer.example.com" \
```

```
--puppet-ca-server                    "capsule.example.com" \
--foreman-proxy-puppetca                 "true" \
--puppet-server-ca                    "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4. On Capsule Server, enter the **satellite-installer** command, for example:

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn      "satellite.example.com" \
--foreman-proxy-register-in-foreman      "true" \
--foreman-proxy-foreman-base-url         "https://satellite.example.com" \
--foreman-proxy-trusted-hosts            "satellite.example.com" \
--foreman-proxy-trusted-hosts            "capsule.example.com" \
--foreman-proxy-oauth-consumer-key       "oauth key" \
--foreman-proxy-oauth-consumer-secret    "oauth secret" \
--certs-tar-file                      "capsule.example.com-certs.tar" \
--puppet-server-foreman-url              "https://satellite.example.com" \
--certs-cname                         "loadbalancer.example.com" \
--puppet-dns-alt-names                   "loadbalancer.example.com" \
--puppet-ca-server                    "capsule.example.com" \
--foreman-proxy-puppetca                 "true" \
--puppet-server-ca                    "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4.2. CONFIGURING CAPSULE SERVER WITH DEFAULT SSL CERTIFICATES FOR LOAD BALANCING WITH PUPPET

The following section describes how to configure Capsule Servers that use default SSL certificates for load balancing with Puppet.

If you use Puppet in your Satellite configuration, you must complete the following procedures:

1. Configuring Capsule Server to Generate and Sign Puppet Certificates
2. Configuring Remaining Capsule Servers for Load Balancing

Configuring Capsule Server to Generate and Sign Puppet Certificates

Complete this procedure only for the system where you want to configure Capsule Server to generate and sign Puppet certificates for all other Capsule Servers that you configure for load balancing. In the examples in this procedure, the FQDN of this Capsule Server is **capsule-ca.example.com**.

1. On Satellite Server, generate Katello certificates for the system where you configure Capsule Server to generate and sign Puppet certificates:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule-ca.example.com \
--certs-tar "/root/capsule-ca.example.com-certs.tar" \
--foreman-proxy-cname loadbalancer.example.com
```

   Retain a copy of the example **satellite-installer** command that is output by the **capsule-certs-generate** command for installing the Capsule Server certificate.

2. Copy the certificate archive file from Satellite Server to Capsule Server:

```
# scp /root/capsule-ca.example.com-certs.tar \
root@capsule-ca.example.com:capsule-ca.example.com-certs.tar
```

3. Append the following options to the **satellite-installer** command that you obtain from the output of the **capsule-certs-generate** command:

```
--certs-cname                          "loadbalancer.example.com" \
--puppet-dns-alt-names                   "loadbalancer.example.com" \
--puppet-ca-server                     "capsule-ca.example.com" \
--foreman-proxy-puppetca                 "true" \
--puppet-server-ca                     "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4. On Capsule Server, enter the **satellite-installer** command, for example:

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn      "satellite.example.com" \
--foreman-proxy-register-in-foreman      "true" \
--foreman-proxy-foreman-base-url         "https://satellite.example.com" \
--foreman-proxy-trusted-hosts            "satellite.example.com" \
--foreman-proxy-trusted-hosts            "capsule-ca.example.com" \
--foreman-proxy-oauth-consumer-key       "oauth key" \
--foreman-proxy-oauth-consumer-secret    "oauth secret" \
--certs-tar-file         "capsule-ca.example.com-certs.tar" \
--puppet-server-foreman-url              "https://satellite.example.com" \
--certs-cname                          "loadbalancer.example.com" \
--puppet-dns-alt-names                   "loadbalancer.example.com" \
--puppet-ca-server                     "capsule-ca.example.com" \
--foreman-proxy-puppetca                 "true" \
--puppet-server-ca                     "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

5. On Capsule Server, generate Puppet certificates for all other Capsule Servers that you configure for load balancing, except this first system where you configure Puppet certificates signing:

```
# puppet cert generate capsule.example.com \
--dns_alt_names=loadbalancer.example.com
```

This command creates the following files on the system where you configure Capsule Server to sign Puppet certificates:

- **/etc/puppetlabs/puppet/ssl/certs/ca.pem**
- **/etc/puppetlabs/puppet/ssl/certs/*capsule.example.com*.pem**
- **/etc/puppetlabs/puppet/ssl/private_keys/*capsule.example.com*.pem**
- **/etc/puppetlabs/puppet/ssl/public_keys/*capsule.example.com*.pem**

Configuring Remaining Capsule Servers for Load Balancing

Complete this procedure on each Capsule Server excluding the system where you configure Capsule Server to sign Puppet certificates.

1. On Satellite Server, generate Katello certificates for Capsule Server:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule.example.com \
--certs-tar "/root/capsule.example.com-certs.tar" \
--foreman-proxy-cname loadbalancer.example.com
```

   Retain a copy of the example **satellite-installer** command that is output by the **capsule-certs-generate** command for installing the Capsule Server certificate.

2. Copy the certificate archive file from Satellite Server to Capsule Server:

```
# scp /root/capsule.example.com-certs.tar \
root@capsule.example.com:capsule.example.com-certs.tar
```

3. On Capsule Server, install the **puppetserver** package:

```
# satellite-maintain packages install puppetserver
```

4. On Capsule Server, create directories for puppet certificates:

```
# mkdir -p /etc/puppetlabs/puppet/ssl/certs/ \
/etc/puppetlabs/puppet/ssl/private_keys/ \
/etc/puppetlabs/puppet/ssl/public_keys/
```

5. On Capsule Server, copy the Puppet certificates for this Capsule Server from the system where you configure Capsule Server to sign Puppet certificates:

```
# scp root@capsule-ca.example.com:/etc/puppetlabs/puppet/ssl/certs/ca.pem \
/etc/puppetlabs/puppet/ssl/certs/ca.pem
# scp root@capsule-
ca.example.com:/etc/puppetlabs/puppet/ssl/certs/capsule.example.com.pem \
/etc/puppetlabs/puppet/ssl/certs/capsule.example.com.pem
# scp root@capsule-
ca.example.com:/etc/puppetlabs/puppet/ssl/private_keys/capsule.example.com.pem \
/etc/puppetlabs/puppet/ssl/private_keys/capsule.example.com.pem
# scp root@capsule-
ca.example.com:/etc/puppetlabs/puppet/ssl/public_keys/capsule.example.com.pem \
/etc/puppetlabs/puppet/ssl/public_keys/capsule.example.com.pem
```

6. On Capsule Server, change the directory ownership to user **puppet**, group **puppet** and set the SELinux contexts:

```
# chown -R puppet:puppet /etc/puppetlabs/puppet/ssl/
# restorecon -Rv /etc/puppetlabs/puppet/ssl/
```

7. Append the following options to the **satellite-installer** command that you obtain from the output of the **capsule-certs-generate** command:

```
--certs-cname "loadbalancer.example.com" \
--puppet-dns-alt-names "loadbalancer.example.com" \
--puppet-ca-server "capsule-ca.example.com" \
```

```
--foreman-proxy-puppetca                "false" \
--puppet-server-ca                    "false" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

8. On Capsule Server, enter the **satellite-installer** command, for example:

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn     "satellite.example.com" \
--foreman-proxy-register-in-foreman     "true" \
--foreman-proxy-foreman-base-url        "https://satellite.example.com" \
--foreman-proxy-trusted-hosts           "satellite.example.com" \
--foreman-proxy-trusted-hosts           "capsule.example.com" \
--foreman-proxy-oauth-consumer-key      "oauth key" \
--foreman-proxy-oauth-consumer-secret   "oauth secret" \
--certs-tar-file                      "capsule.example.com-certs.tar" \
--puppet-server-foreman-url             "https://satellite.example.com" \
--certs-cname                         "loadbalancer.example.com" \
--puppet-dns-alt-names                  "loadbalancer.example.com" \
--puppet-ca-server                    "capsule-ca.example.com" \
--foreman-proxy-puppetca                "false" \
--puppet-server-ca                    :false" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4.3. CONFIGURING CAPSULE SERVER WITH CUSTOM SSL CERTIFICATES FOR LOAD BALANCING WITHOUT PUPPET

The following section describes how to configure Capsule Servers that use custom SSL certificates for load balancing without Puppet.

4.3.1. Creating Custom SSL Certificates for Capsule Server

This procedure outlines how to create a configuration file for the Certificate Signing Request and include the load balancer and Capsule Server as Subject Alternative Names (SAN). Complete this procedure on each Capsule Server that you want to configure for load balancing.

Procedure

1. On Capsule Server, create a directory to contain all the source certificate files, accessible to only the **root** user:

```
# mkdir /root/capsule_cert
# cd /root/capsule_cert
```

2. Create a private key with which to sign the Certificate Signing Request (CSR).
Note that the private key must be unencrypted. If you use a password-protected private key, remove the private key password.

   If you already have a private key for this Capsule Server, skip this step.

```
# openssl genrsa -out /root/capsule_cert/capsule_cert_key.pem 4096
```

3. Create the certificate request configuration file with the following content:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt = no

[ req_distinguished_name ]
countryName=2 Letter Country Code
stateOrProvinceName=State or Province Full Name
localityName=Locality Name
0.organizationName=Organization Name
organizationalUnitName=Capsule Organization Unit Name
commonName=capsule.example.com 1
emailAddress=Email Address

[ req_ext ]
#authorityKeyIdentifier=keyid,issuer
#basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

[alt_names]                2
DNS.1 = loadbalancer.example.com
DNS.2 = capsule.example.com
```

1. The certificate's common name must match the FQDN of Capsule Server. Ensure to change this when running the command on each Capsule Server that you configure for load balancing. You can also set a wildcard value *****. If you set a wildcard value, you must add the **-t capsule** option when you use the **katello-certs-check** command.

2. Under **[alt_names]**, include the FQDN of the load balancer as **DNS.1** and the FQDN of Capsule Server as **DNS.2**.

4. Create a Certificate Signing Request (CSR) for the SAN certificate.

```
# openssl req -new \
-key /root/capsule_cert/capsule_cert_key.pem \ 1
-config SAN_config.cfg \                        2
-out /root/capsule_cert/capsule_cert_csr.pem    3
```

1. Capsule Server's private key, used to sign the certificate
2. The certificate request configuration file
3. Certificate Signing Request file

5. Send the certificate request to the Certificate Authority:
   When you submit the request, specify the lifespan of the certificate. The method for sending the certificate request varies, so consult the Certificate Authority for the preferred method. In response to the request, you can expect to receive a Certificate Authority bundle and a signed certificate, in separate files.

6. Copy the Certificate Authority bundle and Capsule Server certificate file that you receive from the Certificate Authority, and the Capsule Server private key to your Satellite Server.

7. On Satellite Server, validate the Capsule Server certificate input files:

```
# katello-certs-check \
-c /root/capsule_cert/capsule_cert.pem \ 1
-k /root/capsule_cert/capsule_cert_key.pem \ 2
-b /root/capsule_cert/ca_cert_bundle.pem 3
```

1. Capsule Server certificate file, provided by your Certificate Authority
2. Capsule Server's private key that you used to sign the certificate
3. Certificate Authority bundle, provided by your Certificate Authority

If you set the **commonName=** to a wildcard value *, you must add the **-t capsule** option to the **katello-certs-check** command.

Retain a copy of the example **capsule-certs-generate** command that is output by the **katello-certs-check** command for creating the Certificate Archive File for this Capsule Server.

4.3.2. Configuring Capsule Server with Custom SSL Certificates for Load Balancing without Puppet

Complete this procedure on each Capsule Server that you want to configure for load balancing.

Procedure

1. Append the following option to the **capsule-certs-generate** command that you obtain from the output of the **katello-certs-check** command:

```
--foreman-proxy-cname loadbalancer.example.com
```

2. On Satellite Server, enter the **capsule-certs-generate** command to generate Capsule certificates. For example:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule.example.com \
--certs-tar /root/capsule_cert/capsule.tar \
--server-cert /root/capsule_cert/capsule.pem \
--server-key /root/capsule_cert/capsule.pem \
--server-ca-cert /root/capsule_cert/ca_cert_bundle.pem \
--foreman-proxy-cname loadbalancer.example.com
```

Retain a copy of the example **satellite-installer** command from the output for installing the Capsule Server certificates.

3. Copy the certificate archive file from Satellite Server to Capsule Server:

```
# scp /root/capsule.example.com-certs.tar \
root@capsule.example.com:capsule.example.com-certs.tar
```

4. Append the following options to the **satellite-installer** command that you obtain from the output of the **capsule-certs-generate** command. Set the **--puppet-ca-server** option to point to the Capsule Server where you enter the command. You must install Puppet CA on your Capsule Servers, regardless of whether you intend to use it or not. Puppet is configured in its default single-node configuration.

```
--certs-cname                    "loadbalancer.example.com" \
--puppet-dns-alt-names           "loadbalancer.example.com" \
--puppet-ca-server               "capsule.example.com" \
--foreman-proxy-puppetca         "true" \
--puppet-server-ca               "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

5. On Capsule Server, enter the **satellite-installer** command, for example:

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn   "satellite.example.com" \
--foreman-proxy-register-in-foreman   "true" \
--foreman-proxy-foreman-base-url      "https://satellite.example.com" \
--foreman-proxy-trusted-hosts         "satellite.example.com" \
--foreman-proxy-trusted-hosts         "capsule.example.com" \
--foreman-proxy-oauth-consumer-key    "oauth key" \
--foreman-proxy-oauth-consumer-secret "oauth secret" \
--certs-tar-file                 "capsule.example.com-certs.tar" \
--puppet-server-foreman-url      "https://satellite.example.com" \
--certs-cname                    "loadbalancer.example.com" \
--puppet-dns-alt-names           "loadbalancer.example.com" \
--puppet-ca-server               "capsule.example.com" \
--foreman-proxy-puppetca         "true" \
--puppet-server-ca               "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4.4. CONFIGURING CAPSULE SERVER WITH CUSTOM SSL CERTIFICATES FOR LOAD BALANCING WITH PUPPET

The following section describes how to configure Capsule Servers that use custom SSL certificates for load balancing with Puppet.

4.4.1. Creating Custom SSL Certificates for Capsule Server

This procedure outlines how to create a configuration file for the Certificate Signing Request and include the load balancer and Capsule Server as Subject Alternative Names (SAN). Complete this procedure on each Capsule Server that you want to configure for load balancing.

Procedure

1. On Capsule Server, create a directory to contain all the source certificate files, accessible to only the **root** user:

```
# mkdir /root/capsule_cert
# cd /root/capsule_cert
```

2. Create a private key with which to sign the Certificate Signing Request (CSR).

Note that the private key must be unencrypted. If you use a password-protected private key, remove the private key password.

If you already have a private key for this Capsule Server, skip this step.

```
# openssl genrsa -out /root/capsule_cert/capsule.pem 4096
```

3. Create the certificate request configuration file with the following content:

```
[ req ]
default_bits       = 4096
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt = no

[ req_distinguished_name ]
countryName=2 Letter Country Code
stateOrProvinceName=State or Province Full Name
localityName=Locality Name
0.organizationName=Organization Name
organizationalUnitName=Capsule Organization Unit Name
commonName=capsule.example.com (1)
emailAddress=Email Address

[ req_ext ]
#authorityKeyIdentifier=keyid,issuer
#basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

[alt_names]                (2)
DNS.1 = loadbalancer.example.com
DNS.2 = capsule.example.com
```

(1) The certificate's common name must match the FQDN of Capsule Server. Ensure to change this when running the command on each Capsule Server. You can also set a wildcard value *****. If you set a wildcard value, you must add the **-t capsule** option when you use the **katello-certs-check** command.

(2) Under **[alt_names]**, include the FQDN of the load balancer as **DNS.1** and the FQDN of Capsule Server as **DNS.2**.

4. Create a Certificate Signing Request (CSR) for the SAN certificate:

```
# openssl req -new \
-key /root/capsule_cert/capsule.pem \ (1)
-config SAN_config.cfg \               (2)
-out /root/capsule_cert/capsule.pem    (3)
```

(1) Capsule Server's private key, used to sign the certificate

(2) The certificate request configuration file

(3) Certificate Signing Request file

5. Send the certificate request to the Certificate Authority:
   When you submit the request, specify the lifespan of the certificate. The method for sending the certificate request varies, so consult the Certificate Authority for the preferred method. In response to the request, you can expect to receive a Certificate Authority bundle and a signed certificate, in separate files.

6. Copy the Certificate Authority bundle and Capsule Server certificate file that you receive from the Certificate Authority, and the Capsule Server private key to your Satellite Server to validate them.

7. On Satellite Server, validate the Capsule Server certificate input files:

```
# katello-certs-check \
-c /root/capsule_cert/capsule.pem \ 1
-k /root/capsule_cert/capsule.pem \ 2
-b /root/capsule_cert/ca_cert_bundle.pem 3
```

1. Capsule Server certificate file, provided by your Certificate Authority
2. Capsule Server's private key that you used to sign the certificate
3. Certificate Authority bundle, provided by your Certificate Authority

   If you set the **commonName=** to a wildcard value ***, you must add the **-t capsule** option to the **katello-certs-check** command.

   Retain a copy of the example **capsule-certs-generate** command that is output by the **katello-certs-check** command for creating the Certificate Archive File for this Capsule Server.

4.4.2. Configuring Capsule Server with Custom SSL Certificates for Load Balancing with Puppet

If you use Puppet in your Satellite configuration, then you must complete the following procedures:

1. Configuring Capsule Server to Generate and Sign Puppet Certificates
2. Configuring Remaining Capsule Servers for Load Balancing

Configuring Capsule Server to Generate and Sign Puppet Certificates

Complete this procedure only for the system where you want to configure Capsule Server to generate Puppet certificates for all other Capsule Servers that you configure for load balancing. In the examples in this procedure, the FQDN of this Capsule Server is **capsule-ca.example.com**.

1. Append the following option to the **capsule-certs-generate** command that you obtain from the output of the **katello-certs-check** command:

```
--foreman-proxy-cname loadbalancer.example.com
```

2. On Satellite Server, enter the **capsule-certs-generate** command to generate Capsule certificates. For example:

```
# capsule-certs-generate \
```

```
--foreman-proxy-fqdn capsule-ca.example.com \
--certs-tar /root/capsule_cert/capsule-ca.tar \
--server-cert /root/capsule_cert/capsule-ca.pem \
--server-key /root/capsule_cert/capsule-ca.pem \
--server-ca-cert /root/capsule_cert/ca_cert_bundle.pem \
--foreman-proxy-cname loadbalancer.example.com
```

Retain a copy of the example **satellite-installer** command from the output for installing the Capsule Server certificates.

3. Copy the certificate archive file from Satellite Server to Capsule Server.

4. Append the following options to the **satellite-installer** command that you obtain from the output of the **capsule-certs-generate** command:

```
--puppet-dns-alt-names "loadbalancer.example.com" \
--puppet-ca-server "capsule-ca.example.com" \
--foreman-proxy-puppetca "true" \
--puppet-server-ca "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

5. On Capsule Server, enter the **satellite-installer** command, for example:

```
satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn "satellite.example.com" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule-ca.example.com" \
--foreman-proxy-oauth-consumer-key "oauth key" \
--foreman-proxy-oauth-consumer-secret "oauth secret" \
--certs-tar-file "certs.tgz" \
--puppet-server-foreman-url "https://satellite.example.com" \
--certs-cname "loadbalancer.example.com" \
--puppet-dns-alt-names "loadbalancer.example.com" \
--puppet-ca-server "capsule-ca.example.com" \
--foreman-proxy-puppetca "true" \
--puppet-server-ca "true" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

6. On Capsule Server, generate Puppet certificates for all other Capsules that you configure for load balancing, except this first system where you configure Puppet certificates signing:

```
# puppet cert generate capsule.example.com \
--dns_alt_names=loadbalancer.example.com
```

This command creates the following files on the Puppet certificate signing Capsule Server instance:

- **/etc/puppetlabs/puppet/ssl/certs/ca.pem**
- **/etc/puppetlabs/puppet/ssl/certs/capsule.example.com.pem**
- **/etc/puppetlabs/puppet/ssl/private_keys/capsule.example.com.pem**

- **/etc/puppetlabs/puppet/ssl/public_keys/capsule.example.com.pem**

Configuring Remaining Capsule Servers for Load Balancing

Complete this procedure for each Capsule Server excluding the system where you configure Capsule Server to sign Puppet certificates.

1. Append the following option to the **capsule-certs-generate** command that you obtain from the output of the **katello-certs-check** command:

```
--foreman-proxy-cname loadbalancer.example.com
```

2. On Satellite Server, enter the **capsule-certs-generate** command to generate Capsule certificates. For example:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule.example.com \
--certs-tar /root/capsule_cert/capsule.tar \
--server-cert /root/capsule_cert/capsule.pem \
--server-key /root/capsule_cert/capsule.pem \
--server-ca-cert /root/capsule_cert/ca_cert_bundle.pem \
--foreman-proxy-cname loadbalancer.example.com
```

   Retain a copy of the example **satellite-installer** command from the output for installing the Capsule Server certificates.

3. Copy the certificate archive file from Satellite Server to Capsule Server.

```
# scp /root/capsule.example.com-certs.tar \
root@capsule.example.com:capsule.example.com-certs.tar
```

4. On Capsule Server, install the **puppetserver** package:

```
# satellite-maintain packages install puppetserver
```

5. On Capsule Server, create directories for puppet certificates:

```
# mkdir -p /etc/puppetlabs/puppet/ssl/certs/ \
/etc/puppetlabs/puppet/ssl/private_keys/ \
/etc/puppetlabs/puppet/ssl/public_keys/
```

6. On Capsule Server, copy the Puppet certificates for this Capsule Server from the system where you configure Capsule Server to sign Puppet certificates:

```
# scp root@capsule-ca.example.com:/etc/puppetlabs/puppet/ssl/certs/ca.pem \
/etc/puppetlabs/puppet/ssl/certs/ca.pem
# scp root@capsule-
ca.example.com:/etc/puppetlabs/puppet/ssl/certs/capsule.example.com.pem \
/etc/puppetlabs/puppet/ssl/certs/capsule.example.com.pem
# scp root@capsule-
ca.example.com:/etc/puppetlabs/puppet/ssl/private_keys/capsule.example.com.pem \
/etc/puppetlabs/puppet/ssl/private_keys/capsule.example.com.pem
# scp root@capsule-
ca.example.com:/etc/puppetlabs/puppet/ssl/public_keys/capsule.example.com.pem \
/etc/puppetlabs/puppet/ssl/public_keys/capsule.example.com.pem
```

7. On Capsule Server, change the directory ownership to user **puppet**, group **puppet** and set the SELinux contexts:

```
# chown -R puppet:puppet /etc/puppetlabs/puppet/ssl/
# restorecon -Rv /etc/puppetlabs/puppet/ssl/
```

8. Append the following options to the **satellite-installer** command that you obtain from the output of the **capsule-certs-generate** command:

```
--certs-cname                          "loadbalancer.example.com" \
--puppet-dns-alt-names                   "loadbalancer.example.com" \
--puppet-ca-server                     "capsule-ca.example.com" \
--foreman-proxy-puppetca                 "false" \
--puppet-server-ca                     "false" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

9. On Capsule Server, enter the **satellite-installer** command, for example:

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn      "satellite.example.com" \
--foreman-proxy-register-in-foreman      "true" \
--foreman-proxy-foreman-base-url         "https://satellite.example.com" \
--foreman-proxy-trusted-hosts            "satellite.example.com" \
--foreman-proxy-trusted-hosts            "capsule.example.com" \
--foreman-proxy-oauth-consumer-key       "oauth key" \
--foreman-proxy-oauth-consumer-secret    "oauth secret" \
--certs-tar-file                       "capsule.example.com-certs.tar" \
--puppet-server-foreman-url              "https://satellite.example.com" \
--certs-cname                          "loadbalancer.example.com" \
--puppet-dns-alt-names                   "loadbalancer.example.com" \
--puppet-ca-server                     "capsule-ca.example.com" \
--foreman-proxy-puppetca                 "false" \
--puppet-server-ca                     "false" \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

CHAPTER 5. INSTALLING THE LOAD BALANCER

The following example provides general guidance for configuring an HAProxy load balancer. However, you can install any suitable load balancing software solution that supports TCP forwarding and sticky sessions.

1. On a Red Hat Enterprise Linux 7 host, install HAProxy:

```
# yum install haproxy
```

2. Install the following package that includes the **semanage** tool:

```
# yum install policycoreutils-python
```

3. Configure SELinux to allow HAProxy to bind any port:

```
# semanage boolean --modify --on haproxy_connect_any
```

4. Configure the load balancer to balance the network load for the ports as described in Table 5.1, "Ports Configuration for the Load Balancer". For example, to configure ports for HAProxy, edit the **/etc/haproxy/haproxy.cfg** file to correspond with the table.
You must configure sticky session on TCP port 443 to request yum metadata for RPM repositories from different Capsule Servers that you configure for load balancing.

Table 5.1. Ports Configuration for the Load Balancer

Service	Port	Mode	Balance Mode	Destination
HTTP	80	TCP	roundrobin	port 80 on all Capsule Servers
HTTPS	443	TCP	source	port 443 on all Capsule Servers
RHSM	8443	TCP	roundrobin	port 8443 on all Capsule Servers
AMQP	5647	TCP	roundrobin	port 5647 on all Capsule Servers
Puppet (*Optional*)	8140	TCP	roundrobin	port 8140 on all Capsule Servers
PuppetCA (*Optional*)	8141	TCP	roundrobin	port 8140 only on the system where you configure Capsule Server to sign Puppet certificates
SmartProxy (*Optional for OpenScap*)	9090	TCP	roundrobin	port 9090 on all Capsule Servers

Service	Port	Mode	Balance Mode	Destination
Docker (*Optional*)	5000	TCP	roundrobin	port 5000 on all Capsule Servers

5. Configure the load balancer to disable SSL offloading and allow client-side SSL certificates to pass through to back end servers. This is required because communication from clients to Capsule Servers depends on client-side SSL certificates.

6. Start and enable the HAProxy service:

```
# systemctl start haproxy
# systemctl enable haproxy
```

CHAPTER 6. REGISTERING CLIENTS

You can register a client running a Red Hat Enterprise Linux version 6, 7 or 8 operating system to Capsule Servers that you configure for load balancing. For more information about registering clients and configuring them to use Puppet, see Registering Hosts in the *Managing Hosts* guide.

To register clients, proceed to one of the following procedures:

- Section 6.1, “Registering Clients Using the Bootstrap Script”
- Section 6.2, “Registering Clients Manually”

6.1. REGISTERING CLIENTS USING THE BOOTSTRAP SCRIPT

To register clients, enter the following command on the client. You must complete the registration procedure for each client.

Prerequisite

Ensure that you install the bootstrap script on the client and change the script’s file permissions to executable. For more information, see Registering Hosts to Red Hat Satellite Using The Bootstrap Script in the *Managing Hosts* guide.

- On Red Hat Enterprise Linux 8, enter the following command:

```
# /usr/libexec/platform-python bootstrap.py \
--login=admin \
--server loadbalancer.example.com \
--organization="Your_Organization" \
--location="Your_Location" \
--hostgroup="Your_Hostgroup" \
--activationkey=your_activation_key \
--enablerepos=rhel-7-server-satellite-tools-6.8-rpms \
--puppet-ca-port 8141 \ 1
--force 2
```

1. Include the **--puppet-ca-port 8141** option if you use Puppet.
2. Include the **--force** option to register the client that has been previously registered to a standalone Capsule.

- On Red Hat Enterprise Linux 7, 6, or 5, enter the following command:

```
# python bootstrap.py --login=admin \
--server loadbalancer.example.com \
--organization="Your_Organization" \
--location="Your_Location" \
--hostgroup="Your_Hostgroup" \
--activationkey=your_activation_key \
--enablerepos=rhel-7-server-satellite-tools-6.8-rpms \
--puppet-ca-port 8141 \ 1
--force 2
```

1. Include the **--puppet-ca-port 8141** option if you use Puppet.

2 Include the **--force** option to register the client that has been previously registered to a standalone Capsule.

The script prompts for the password corresponding to the Satellite user name you entered with the **--login** option.

6.2. REGISTERING CLIENTS MANUALLY

To register clients manually, complete the following procedure on each client that you register.

Procedure

1. Remove the **katello-ca-consumer** package if it is installed:

```
# yum remove 'katello-ca-consumer*'
```

2. Install the **katello-ca-consumer** package from the load balancer:

```
# rpm -Uvh \
http://loadbalancer.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

3. Register the client and include the **--serverurl** and **--baseurl** options:

```
# subscription-manager register --org=Your_Organization \
--activationkey=Your_Activation_Key \
--serverurl=https://loadbalancer.example.com:8443/rhsm \
--baseurl=https://loadbalancer.example.com/pulp/repos
```

CHAPTER 7. PROMOTING SCAP CONTENT TO CLIENTS

The following section describes how to promote Security Content Automation Protocol (SCAP) content to clients registered to Capsule Servers that you configure for load balancing.

Prerequisites

- Ensure that you configure the SCAP content. For more information, see Configuring SCAP Content in *Administering Red Hat Satellite*.

Procedure

1. In the Satellite web UI, navigate to **Configure** > **Classes** and click **foreman_scap_client**.
2. Click the **Smart Class Parameter** tab.
3. In the pane to the left of the **Smart Class Parameter** window, click **port**.
4. In the **Default Behavior** area, select the **Override** check box.
5. From the **Key Type** list, select **integer**.
6. In the **Default Value** field, enter **9090**.
7. In the pane to the left of the **Smart Class Parameter** window, click **server**.
8. In the **Default Behavior** area, select the **Override** check box.
9. From the **Key Type** list, select **string**.
10. In the **Default Value** field, enter the FQDN of your load balancer. For example, ***loadbalancer.example.com***.
11. In the lower left of the **Smart Class Parameter** window, click **Submit**.
12. Add the puppet module that contains the **foreman_scap_client** puppet class to a Content View. Publish and promote this Content View to your client's environment.
13. If you want to verify the configuration, run the Puppet agent on the client to promote the changes. Do not run the Puppet agent on every client manually because the Puppet agent runs on the clients every 30 minutes.

```
# puppet agent -t --noop
```

14. On the client, verify that the **/etc/foreman_scap_client/config.yaml** file contains the following lines:

```
# Foreman proxy to which reports should be uploaded
:server: 'loadbalancer.example.com'
:port: 9090
```

Additional Resources

- For more information about adding puppet modules to Satellite Server, see Adding Puppet Modules to Red Hat Satellite 6 in the *Puppet Guide*.

- For more information about Content Views, see Managing Content Views in the *Content Management Guide*.

CHAPTER 8. VERIFYING THE LOAD BALANCING CONFIGURATION

You can verify the load balancing configuration by completing the following steps for each Capsule Server that you configure:

1. Shut down the base operating system for your Capsule Server.
2. Verify that content or subscription management features are available on the client registered to this Capsule. For example, enter the **subscription-manager refresh** command on the client.
3. Restart the base operating system for your Capsule Server.