Red Hat Satellite 6.4

Load Balancing Guide

Setting up a Load Balanced Capsule Pool

Last Updated: 2018-10-22

Red Hat Satellite 6.4 Load Balancing Guide

Setting up a Load Balanced Capsule Pool

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

Legal Notice

Abstract

The Load Balancing Guide describes how to set up a load balanced Red Hat Satellite Capsule Server pool. This guide is aimed primarily at Satellite administrators with sound networking knowledge and skills.

Table of Contents

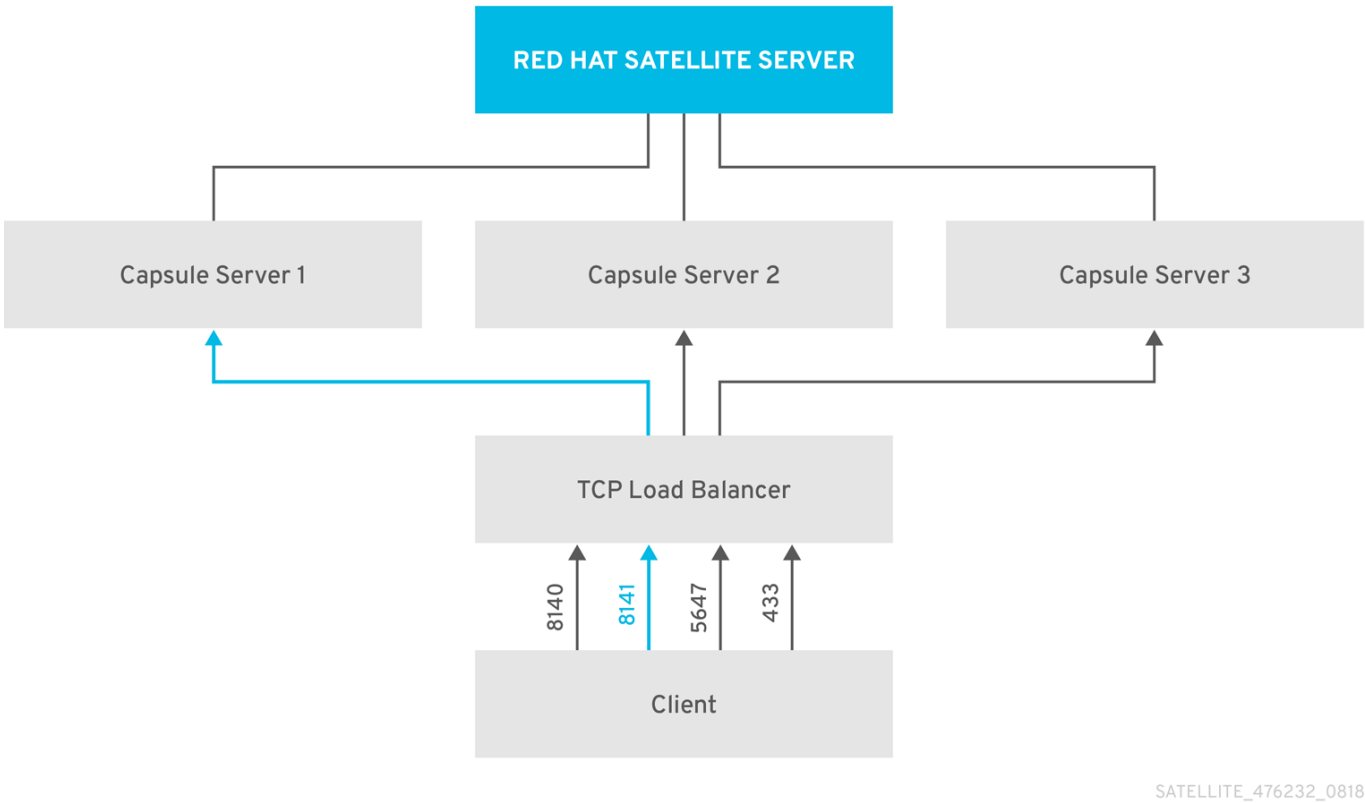
1. LOAD BALANCING SOLUTION OVERVIEW

You can set up a Satellite Server environment to load balance multiple Capsule Server instances across a cluster. To achieve that, you must configure a load balancer. The benefits are overall improved performance by efficiently distributing client requests and network load, therefore decreasing the burden on your Capsule Server instances.

A typical load balanced Capsule pool setup consists of the following components:

- An existing Satellite 6.4 environment,
- Two or more Capsule Server instances for the load balanced cluster,
- A load balancer,
- Multiple clients.

Figure 1. Satellite Load Balancing Solution Architecture



In a load balanced setup, nearly all Capsule functionality continues to work as expected when one Capsule Server instance is taken down, for either planned or unplanned maintenance. The following services and features are load balanced in this solution:

- Registration using **`subscription-manager`**,
- Content Management (**`yum`** repositories),
- Puppet (optional).

NOTE

If you use Puppet modules, the main caveat of the proposed solution is a known Puppet limitation where Puppet Certificate Authority (CA) management does not support load balanced certificate signing. Because Puppet CA stores certificate information, such as the serial number counter and CRL on the file system, when multiple writer processes attempt to use the same data, it can easily become corrupted.

As a workaround to manage this Puppet limitation, this solution proposes that you complete the following steps:

1. Configure Puppet certificate signing on one Capsule Server instance, typically the first instance in the cluster.
2. On the client machines, configure sending CA requests to port 8141 on the load balancer.
3. On the load balancer, configure this port to redirect to 8140 on the Capsule Server instance with Puppet certificate signing capabilities, typically the first instance in the cluster.

You can find additional information on how to prepare the Satellite Server and Capsule Server environments, as well as guidelines on how to configure a load balancer and register clients, in the following chapters of this document.

2. BEFORE YOU BEGIN

Balancing application requests reduces server load and prevents any one Capsule from becoming a single point of failure by improving availability and responsiveness. Setting up a load balanced Capsule pool can provide resilience against planned and unplanned outages.

Before implementing this solution, please ensure that you are aware of the following:

- If you use Puppet, bear in mind that Puppet certificate signing is assigned to the first Capsule in the pool. If the first Capsule in the pool is down, clients can not obtain Puppet content from the Capsule. In this case, Puppet certificate signing will become available when the first Capsule returns to service.
- This solution does not use pacemaker or other similar HA tools to maintain one state across all Capsules. To troubleshoot any issues, you must reproduce the issue on each Capsule, bypassing the load balancer.
- Setting up a load balanced capsule pool results in a more complex environment and additional maintenance.
    - You must ensure that all Capsules have the same content views and that they are synchronized to the same content view version.
    - You must upgrade each Capsule in sequence.
    - You must perform regular backups of each Capsule in the pool.

3. PREPARING SATELLITE SERVER AND CAPSULE SERVERS FOR LOAD BALANCING

The following chapter describes how to prepare Satellite Server and the Capsule Server machines that make up your Capsule Server load balanced pool.

Preparing Satellite Server

You must perform a typical installation of the minimum required version of Satellite Server for this solution. The minimum version requirement for Satellite Server is Satellite Server 6.4. For more information about installing Satellite Server, see Installing Satellite Server from a Connected Network.

Preparing Capsule Servers

To prepare each Capsule Server machine in the load balanced pool for installation, you must complete the following procedures described in the *Red Hat Satellite Installing Capsule Server*.

1. Registering Capsule Server to Satellite Server
2. Identifying and Attaching the Capsule Server Subscription
3. Configuring Repositories
4. Synchronizing Time
5. Installing Capsule Server

3.1. Completing the Capsule Server Installation for Load Balancing

The following chapter describes how to complete the Capsule Server installation on each of the machines that make up your Capsule Server load balanced pool.

If you use Puppet, then due to a known Puppet limitation, you must configure certificate signing on one Capsule Server instance, typically the first instance in the cluster. Therefore, you must complete both tasks:

1. Completing the installation for the certificate signing Capsule Server instance, typically the first instance in the cluster.
2. Completing the installation for all the remaining Capsule Server instances in the cluster.

If you don't use Puppet, then complete the Capsule Server installation as described in To Complete the Installation for the Remaining Capsule Servers: and omit the generating Puppet certificates step.

Prerequisite

You have completed the Preparing Capsule Server steps in the Section 3, "Preparing Satellite Server and Capsule Servers for Load Balancing" chapter.

To Complete the Installation for the Certificate Signing Capsule Server:

1. On Satellite Server, generate Katello certificates. Create the certificates archive with the included **`--foreman-proxy-cname`** option and load balancer host name:

   ```
   # capsule-certs-generate \
   --foreman-proxy-fqdn capsule01.example.com \
   --certs-tar "/root/capsule01.example.com-certs.tar" \
   --foreman-proxy-cname loadbalancer.example.com
   ```

   Note that each time this command is run, it generates a unique **`satellite-installer`** command. Note it down and save it for later use.

2. On Capsule Server, edit the **`custom-hiera.yaml`** file by adding the following line:

```
pulp::lazy_redirect_host: loadbalancer.example.com
```

3. Edit the **satellite-installer** command that was generated in a previous step, when generating Katello certificates, by appending the following lines to it:

```
--puppet-dns-alt-names loadbalancer.example.com \
--puppet-ca-server capsule01.example.com \
--foreman-proxy-puppetca true \
--puppet-server-ca true \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4. On Capsule Server, enter the edited **satellite-installer** command. The following is an 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                 "capsule01.example.com"
\
--foreman-proxy-oauth-consumer-key            "oauth key" \
--foreman-proxy-oauth-consumer-secret         "oauth secret" \
--foreman-proxy-content-pulp-oauth-secret     "katello oauth secret" \
--foreman-proxy-content-certs-tar             "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 capsule01.example.com \
--foreman-proxy-puppetca true \
--puppet-server-ca true \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

5. Generate Puppet Certificates for the remaining Capsules. Because Puppet certificate signing is assigned to the first Capsule in the pool, you must generate Puppet certificates for all the remaining Capsules in the pool except the first one. On Capsule Server, enter the following command:

```
# puppet cert generate capsule02.example.com \
--dns_alt_names=loadbalancer.example.com
```

Remember to edit the fully qualified domain name (fqdn) and run this command for every remaining Capsule in the load balanced pool.

To Complete the Installation for the Remaining Capsule Servers:

1. On Satellite Server, generate Katello certificates. Create the certificates archive with the included **--foreman-proxy-cname** option and load balancer name:

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule02.example.com \
--certs-tar "/root/capsule02.example.com-certs.tar" \
```

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

Note that each time this command is run, it generates a unique **satellite-installer** command. Note it down and save it for later use.

2. Generate Puppet Certificates. On Capsule Server, perform the following steps:

   i. Install the **puppetserver** RPM.

   ii. Copy the following files from the first Capsule Server instance to the current one (for example, the second, third and so on). Remember to edit the fqdn for every remaining Capsule:

   - **/etc/puppetlabs/puppet/ssl/certs/ca.pem**
   - **/etc/puppetlabs/puppet/ssl/certs/*capsule02.example.com*.pem**
   - **/etc/puppetlabs/puppet/ssl/private_keys/*capsule02.example.com*.pem**
   - **/etc/puppetlabs/puppet/ssl/public_keys/*capsule02.example.com*.pem**

   iii. Ensure the files are owned by user **puppet**, group **puppet** by entering the following command:

```
# chown -R puppet.puppet /etc/puppetlabs/puppet/ssl/
```

   iv. Ensure the SELinux contexts are set accordingly by entering the following command:

```
# restorecon -Rv /etc/puppetlabs/puppet/ssl/
```

3. On Capsule Server, edit the **custom-hiera.yaml** file by adding the following line:

```
pulp::lazy_redirect_host: loadbalancer.example.com
```

4. Edit the **satellite-installer** command that was generated in a previous step, when generating Katello certificates, by appending the following lines to it:

```
--puppet-dns-alt-names loadbalancer.example.com \
--puppet-ca-server capsule01.example.com \
--foreman-proxy-puppetca false \
--puppet-server-ca false \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

5. On Capsule Server, enter the edited **satellite-installer** command. The following is an 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                 "capsule02.example.com"
\
--foreman-proxy-oauth-consumer-key            "oauth key" \
--foreman-proxy-oauth-consumer-secret         "oauth secret" \
```

```
--foreman-proxy-content-pulp-oauth-secret     "katello oauth secret" \
--foreman-proxy-content-certs-tar             "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 capsule01.example.com \
--foreman-proxy-puppetca false \
--puppet-server-ca false \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

4. INSTALLING THE LOAD BALANCER

The following section provides a general guidance example on how to configure a HAProxy load balancer. You can install any other load balancer software solution that supports TCP forwarding and sticky sessions. It is up to you to decide on a suitable load balancer.

1. On a Red Hat Enterprise Linux 7 host, install the HAProxy RPM.
2. Configure SELinux to allow HAProxy to bind any port:

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

3. Configure the **`/etc/haproxy/haproxy.cfg`** file to balance all ports the following way:

Table 1. Ports Configuration for HAProxy

Service	Port	Mode	Balance Mode	Destination
HTTP	80	TCP	roundrobin	port 80 on all Capsules
HTTPS	443	TCP	source	port 443 on all Capsules
RHSM	8443	TCP	roundrobin	port 8443 on all Capsules
AMQP	5647	TCP	roundrobin	port 5647 on all Capsules
Puppet (*Optional*)	8140	TCP	roundrobin	port 8140 on all Capsules
PuppetCA (*Optional*)	8141	TCP	roundrobin	port 8140 on the first Capsule
SmartProxy (*Optional for OpenScap*)	9090	TCP	roundrobin	port 9090 on all Capsules

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

Note that the additional port for PuppetCA (port 8141) is forwarded only to the first Capsule.

5. REGISTERING CLIENTS

Any client running on a Red Hat Enterprise Linux version 6 or 7 base system can be registered to the load balanced Capsule pool. If the client has been previously registered to a standalone Capsule, you must register it to the load balanced Capsule pool instead.

There are two ways in which you can register clients to a load balanced Capsule pool:

- Registering clients using the bootstrap script,
- Manually registering clients.

It is recommended that you use the bootstrap script to register clients.

5.1. Registering Clients Using the Bootstrap Script

Prerequisite

You have installed the bootstrap script on the client machine and made it executable. For more information, see the Registering Hosts to Satellite 6 Using The Bootstrap Script section in *Managing Hosts*.

To Register Clients Using the Bootstrap Script:

To register clients to a load balanced Capsule, enter the following command on the client machine:

```
# 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-6.4-rpms \
--unmanaged \                    1
--puppet-ca-port 8141 \          2
--force                          3
```

1. Include the **`--unmanaged`** option if your hostgroup is not completely set up with all the OS and provisioning details.
2. Include the **`--puppet-ca-port 8141`** option if you use Puppet.
3. Include the **`--force`** option to register the client that has been previously registered to a standalone Capsule to the load balanced capsule instead.

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

NOTE

You must complete the registration procedure for each client machine.

5.2. Manually Registering Clients

1. If the katello-ca-consumer package has been previously installed on the host, you must remove it. On the client machine, enter the following command:

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

2. Install the katello-ca-consumer package, by entering the following command:

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

3. Register the system and include the **--serverurl** and **--baseurl** options by entering the following command:

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

4. Install `puppet-agent`by entering the following command:

```
# yum install puppet-agent
```

5. Edit the **/etc/puppetlabs/puppet/puppet.conf** file to add the following lines in the agent section:

```
server = loadbalancer.example.com
ca_server = loadbalancer.example.com
ca_port = 8141
```

6. Run **puppet agent** and include the **--noop** option by entering the following command:

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

7. In the Satellite Server web UI, sign the SSL certificate for Puppet by completing the following steps:

   a. Log in to the Satellite Server web UI.

   b. Navigate to **Infrastructure** > **Capsules**.

   c. In the **Actions** column for **capsule01**, click the **Edit** list and select **Certificates**.

   d. Click **Sign**.

e. Enter the **puppet agent** command again, to ensure it works as expected after signing the certificate:

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

6. VERIFYING THE LOAD BALANCING CONFIGURATION

After completing the configuration, you can verify it by performing the following actions:

1. Register a client machine to the load balanced Capsule pool.
2. Shut down one of the Capsule Server instances.
3. Verify that content or subscription management features are available to the client. For example, execute the **subscription-manager refresh** command, or **yum** commands.
4. Restart the previously shut down Capsule Server instance.
5. Repeat steps 2 - 4 until all Capsule Server instances in the load balanced pool have been shut down and restarted.