Red Hat Satellite 6.10

使用负载均衡器配置胶囊

在胶囊服务器之间分布负载

Last Updated: 2024-02-15

Red Hat Satellite 6.10 使用负载均衡器配置胶囊

在胶囊服务器之间分布负载

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

法律通告

摘要

本指南介绍了如何将 Red Hat Satellite 配置为使用负载均衡器在胶囊服务器之间分发负载。

目录

第 1 章 负载均衡解决方案架构

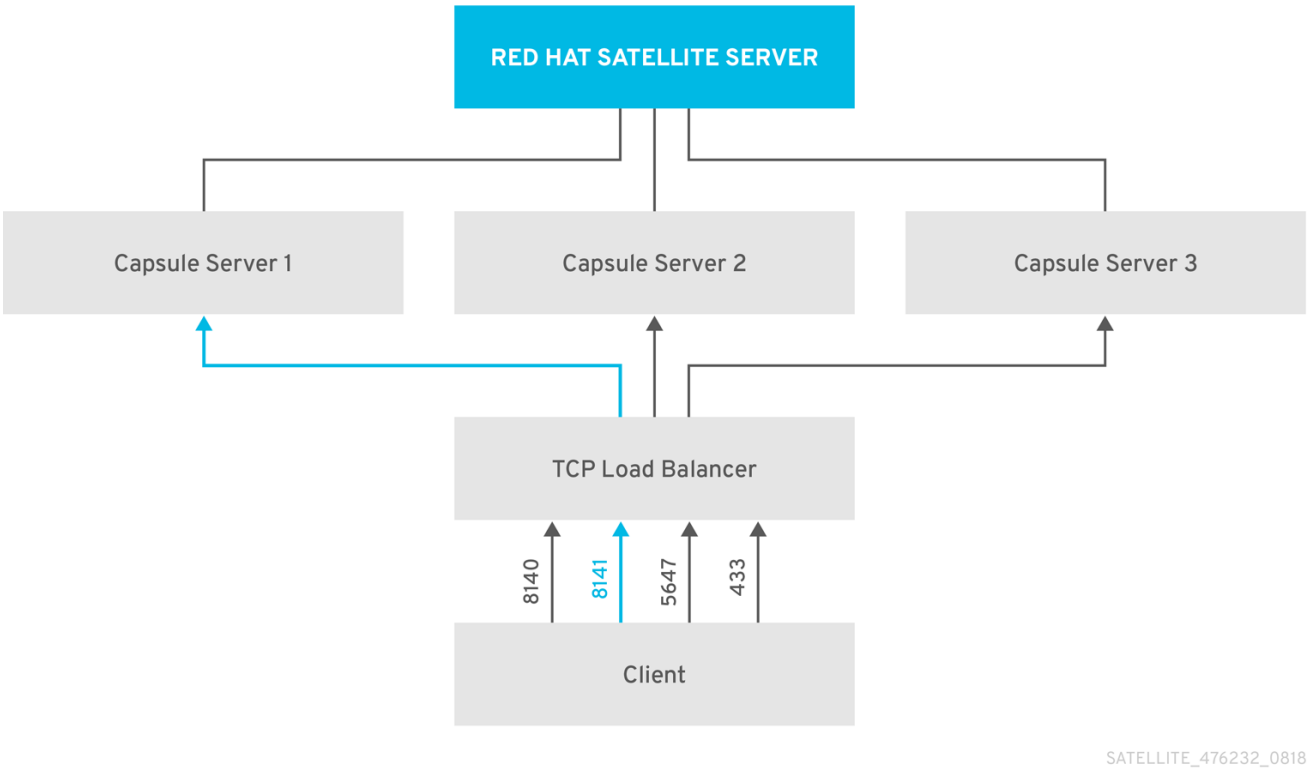
您可以将卫星服务器配置为使用负载平衡器在多个胶囊服务器之间分发客户端请求和网络负载。这会对胶囊服务器产生整体性能提升。

本指南概述了如何准备卫星服务器和胶囊服务器以进行负载平衡，并提供了有关如何配置负载均衡器并在负载均衡的设置中注册客户端的指导。

负载均衡的设置由以下组件组成：

- Satellite Server
- 两个或多个胶囊服务器
- 一个负载均衡器
- 多个客户端

图 1.1. Satellite 负载平衡解决方案架构



在负载均衡的设置中，当一个胶囊服务器停机或计划外维护时，几乎所有胶囊功能都可以继续按预期工作。负载均衡器具有以下服务和特性：

- 使用 **subscription-manager**进行注册
- 使用 **yum** 软件仓库管理内容
- 可选：Puppet

注意

在负载均衡的设置中，负载均衡器只会为上述服务和功能分发负载。如果其它服务（如 provisioning 或 virt-who）在各个胶囊上运行，则必须直接通过胶囊方式访问这些服务，而不是通过负载均衡器访问。

管理 Puppet 限制

Puppet 证书颁发机构(CA)管理不支持负载均衡的设置中的证书签名请求。Puppet CA 在文件系统中存储证书信息，如序列号计数器和 CRL。试图使用相同数据的多个写入程序可能会损坏数据。

要管理此 Puppet 限制，请完成以下步骤：

1. 在一台胶囊服务器上配置 Puppet 证书签名请求，通常是您配置用于负载平衡的首个系统。
2. 配置客户端，以将 CA 请求发送到负载均衡器上的端口 8141。
3. 配置负载均衡器，以将 CA 请求从端口 8141 重定向到您要配置为签署 Puppet 证书的系统上的端口 8140。

第 2 章 负载均衡注意事项

在多个胶囊服务器之间分布负载可防止任何一个胶囊成为单一故障点。将胶囊配置为使用负载平衡器提供对计划和计划外中断的弹性。这提高了可用性和响应能力。

在配置负载均衡时请考虑以下指南：

- 如果使用 Puppet，Puppet 证书签名请求将分配给您配置的第一个胶囊。如果第一个胶囊停机，客户端无法获取 Puppet 内容。
- 此解决方案不使用 Pacemaker 或其他类似 HA 工具来维护所有胶囊之间的一个状态。要排除问题，请在每个胶囊上重现问题，绕过负载均衡器。

负载均衡所需的额外维护

配置胶囊以使用负载均衡器会导致更复杂的环境，需要进行额外的维护。

负载平衡需要以下额外步骤：

- 您必须确保所有 Capsules 具有相同的 Content Views，并将所有胶囊同步到同一内容视图版本
- 您必须按顺序升级每个胶囊
- 您必须定期备份每个配置的胶囊

在负载均衡配置中升级胶囊服务器

要将胶囊服务器从 6.9 升级到 6.10，请完成 *更新和更新 Red Hat Satellite 中的 升级 胶囊服务器 步骤*。在负载平衡配置中，胶囊服务器不需要额外的步骤。

第 3 章 为负载平衡配置胶囊服务器的先决条件

要配置胶囊服务器以进行负载平衡，请完成以下步骤，如 *安装胶囊服务器* 中所述。卫星不支持为负载平衡配置现有的胶囊服务器。

1. 将胶囊服务器注册到卫星服务器
2. 附加 Satellite 基础架构订阅
3. 配置软件仓库
4. 使用 chronyd 同步系统时钟
5. 安装胶囊服务器软件包

第 4 章 为负载平衡配置胶囊服务器

本章概述了如何配置用于负载平衡的胶囊服务器。根据您的 Satellite 服务器配置，继续执行以下部分之一：

1. 第 4.1 节 “使用默认 SSL 证书配置胶囊服务器以在没有 Puppet 的情况下进行负载平衡”
2. 第 4.2 节 “使用默认 SSL 证书配置胶囊服务器以使用 Puppet 负载平衡”
3. 第 4.3 节 “使用自定义 SSL 证书配置胶囊服务器以在没有 Puppet 的情况下进行负载平衡”
4. 第 4.4 节 “使用自定义 SSL 证书配置胶囊服务器以使用 Puppet 负载平衡”

为您为每个胶囊服务器创建的 Katello 证书使用不同的文件名。例如，将证书存档文件命名为 Capsule Server FQDN。

4.1. 使用默认 SSL 证书配置胶囊服务器以在没有 PUPPET 的情况下进行负载平衡

下面的部分论述了如何配置不使用 Puppet 的情况下使用默认 SSL 证书进行负载平衡的胶囊服务器。

在您要为负载均衡配置的每个胶囊服务器上完成这个步骤。

流程

1. 在卫星服务器上，为 Capsule Server 生成 Katello 证书，例如：

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

保留 satellite-installer 命令示例 **satellite-installer** 命令的副本，它通过 **capsule-certs-generate** 命令安装 Capsule Server 证书。

2. 将证书存档文件从卫星服务器复制到胶囊服务器。

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

3. 将以下选项附加到 **satellite-installer** 命令中，从 **capsule-certs-generate** 命令的输出中获取。将 **--puppet-ca-server** 选项设置为指向您输入命令的胶囊服务器。您必须在胶囊服务器上安装 Puppet CA，无论您是否打算使用它。Puppet 在默认的单节点配置中配置。

```
--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. 在 Capsule Server 上，输入 **satellite-installer** 命令，例如：

```
# satellite-installer --scenario capsule \
```

```
--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. 使用默认 SSL 证书配置胶囊服务器以使用 PUPPET 负载平衡

下面的部分论述了如何配置使用默认 SSL 证书进行 Puppet 负载平衡的胶囊服务器。

如果在 Satellite 配置中使用 Puppet，您必须完成以下步骤：

1. 配置胶囊服务器以生成和签署 Puppet 证书
2. 为负载平衡配置重新平衡

配置胶囊服务器以生成和签署 Puppet 证书

对于您要配置用于负载均衡的所有其他胶囊服务器的系统，请完成此步骤。在这一流程中的示例中，此胶囊服务器的 FQDN 为 **capsule-ca.example.com**。

1. 在卫星服务器上，为您配置 Capsule Server 以生成并签名 Puppet 证书的系统生成 Katello 证书：

```
# 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
```

   保留 satellite-installer 命令示例 **satellite-installer** 命令的副本，它通过 **capsule-certs-generate** 命令安装 Capsule Server 证书。

2. 将证书存档文件从 Satellite 服务器复制到胶囊服务器：

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

3. 将以下选项附加到 **satellite-installer** 命令中，从 **capsule-certs-generate** 命令的输出中获取：

```
--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. 在 Capsule Server 上，输入 **satellite-installer** 命令，例如：

```
# satellite-installer --scenario capsule \
--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. 在 Capsule Server 上，停止 Puppet 服务器：

```
# puppet resource service puppetserver ensure=stopped
```

6. 为您为负载平衡配置的其他所有胶囊服务器生成 Puppet 证书，除了配置 Puppet 证书签名的第一个系统外：

```
# puppetserver ca generate --certname capsule.example.com \
--subject-alt-names loadbalancer.example.com --ca-client
```

此命令会在配置胶囊服务器以为 Puppet 证书签名的系统中创建以下文件：

- **/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**

7. 恢复 Puppet 服务器：

```
# puppet resource service puppetserver ensure=running
```

为负载平衡配置重新平衡

在每个胶囊服务器上完成这个步骤，不包括您配置 Capsule Server 以为 Puppet 证书签名的系统。

1. 在卫星服务器上，为 Capsule Server 生成 Katello 证书：

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

保留 satellite-installer 命令示例 **satellite-installer** 命令的副本，它通过 **capsule-certs-generate** 命令安装 Capsule Server 证书。

2. 将证书存档文件从 Satellite 服务器复制到胶囊服务器：

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

3. 在 Capsule Server 上，安装 **puppetserver** 软件包：

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

4. 在 Capsule Server 上，为 puppet 证书创建目录：

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

5. 在 Capsule Server 上，从配置胶囊服务器的系统中复制此胶囊服务器的 Puppet 证书，以为 Puppet 证书签名：

```
# 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. 在 Capsule Server 上，将目录所有权更改为用户 **puppet**、组 **puppet** 并设置 SELinux 上下文：

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

7. 将以下选项附加到 **satellite-installer** 命令中，从 **capsule-certs-generate** 命令的输出中获取：

```
--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. 在 Capsule Server 上，输入 **satellite-installer** 命令，例如：

```
# satellite-installer --scenario capsule \
--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. 使用自定义 SSL 证书配置胶囊服务器以在没有 PUPPET 的情况下进行负载平衡

下面的部分论述了如何配置在没有 Puppet 的情况下使用自定义 SSL 证书进行负载平衡的胶囊服务器。

4.3.1. 为胶囊服务器创建自定义 SSL 证书

此流程概述了如何为证书签名请求创建配置文件，并包含负载均衡器和胶囊服务器，作为主题备用名称 (SAN)。在您要为负载均衡配置的每个胶囊服务器上完成这个步骤。

流程

1. 在 Capsule Server 上，创建包含所有源证书文件的目录，仅可供 **root** 用户访问：

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

2. 创建为证书签名请求(CSR)签名的私钥。
请注意，私钥必须未加密的。如果您使用受密码保护的私钥，请删除私钥密码。

   如果您已拥有此胶囊服务器的私钥，请跳过这一步。

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

3. 使用以下内容创建证书请求配置文件：

```
[ 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. 证书的通用名称必须与胶囊服务器的 FQDN 匹配。在您为负载平衡配置的每个胶囊服务器上运行命令时，请确保进行更改。您还可以设置通配符值 *。如果设置通配符值，您必须在使用 **katello -certs-check 命令时添加 -t** Capsule 选项。
2. 在 **[alt_names]** 下，将负载均衡器的 FQDN 包含为 **DNS.1**，并将胶囊服务器的 FQDN 作为 **DNS.2**。

4. 为 SAN 证书创建证书签名请求(CSR)。

```
# 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. 胶囊服务器的私钥，用于为证书进行签名
2. 证书请求配置文件
3. 证书签名请求文件

5. 将证书请求发送到证书颁发机构：
提交请求时，指定证书的寿命。发送证书请求的方法各不相同，因此请查阅证书颁发机构以获取首选方法。为了响应请求，预计可以在单独的文件中接收证书颁发机构捆绑包和签名证书。

6. 将您从认证机构接收的证书颁发机构和胶囊服务器证书文件复制到卫星服务器上。

7. 在卫星服务器上，验证胶囊服务器证书输入文件：

```
# 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. 由您的认证机构提供的胶囊服务器证书文件
2. 用于为证书签名的胶囊服务器私钥
3. 由认证机构提供的证书颁发机构捆绑包

如果将 **commonName=** 设置为通配符值 *，您必须将 **-t Capsule 选项添加到 katello-certs-check** 命令中。

保留由 **katello-certs-check** 命令输出的示例 **capsule-certs-generate** 命令的副本，用于为这个胶囊服务器创建证书存档文件。

4.3.2. 使用自定义 SSL 证书配置胶囊服务器以在没有 Puppet 的情况下进行负载平衡

在您要为负载均衡配置的每个胶囊服务器上完成这个步骤。

流程

1. 将以下选项附加到您从 **katello-certs-check** 命令的输出中获取的 **capsule-certs-generate** 命令：

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

2. 在卫星服务器上，输入 **capsule-certs-generate** 命令来生成胶囊证书。例如：

```
# 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
```

   保留用于安装 Capsule Server 证书的输出中的 example **satellite-installer** 命令的副本。

3. 将证书存档文件从 Satellite 服务器复制到胶囊服务器：

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

4. 将以下选项附加到 **satellite-installer** 命令中，从 **capsule-certs-generate** 命令的输出中获取。将 **--puppet-ca-server** 选项设置为指向您输入命令的胶囊服务器。您必须在胶囊服务器上安装 Puppet CA，无论您是否打算使用它。Puppet 在默认的单节点配置中配置。

```
--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. 在 Capsule Server 上，输入 **satellite-installer** 命令，例如：

```
# satellite-installer --scenario capsule \
--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. 使用自定义 SSL 证书配置胶囊服务器以使用 PUPPET 负载平衡

下面的部分论述了如何配置使用自定义 SSL 证书进行 Puppet 负载平衡的胶囊服务器。

4.4.1. 为胶囊服务器创建自定义 SSL 证书

此流程概述了如何为证书签名请求创建配置文件，并包含负载均衡器和胶囊服务器，作为主题备用名称 (SAN)。在您要为负载均衡配置的每个胶囊服务器上完成这个步骤。

流程

1. 在 Capsule Server 上，创建包含所有源证书文件的目录，仅可供 **root** 用户访问：

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

2. 创建为证书签名请求(CSR)签名的私钥。
请注意，私钥必须未加密的。如果您使用受密码保护的私钥，请删除私钥密码。

   如果您已拥有此胶囊服务器的私钥，请跳过这一步。

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

3. 使用以下内容创建证书请求配置文件：

```
[ 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. 证书的通用名称必须与胶囊服务器的 FQDN 匹配。在每一胶囊服务器上运行 命令时，请确保进行更改。您还可以设置通配符值 *。如果设置通配符值，您必须在使用 **katello -certs-check 命令时添加 -t** Capsule 选项。
2. 在 **[alt_names]** 下，将负载均衡器的 FQDN 包含为 **DNS.1**，并将胶囊服务器的 FQDN 作为 **DNS.2**。

4. 为 SAN 证书创建证书签名请求(CSR)：

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

1. 胶囊服务器的私钥，用于为证书进行签名
2. 证书请求配置文件
3. 证书签名请求文件

5. 将证书请求发送到证书颁发机构：
提交请求时，指定证书的寿命。发送证书请求的方法各不相同，因此请查阅证书颁发机构以获取首选方法。为了响应请求，预计可以在单独的文件中接收证书颁发机构捆绑包和签名证书。

6. 将您从认证机构接收的证书颁发机构和胶囊服务器证书文件复制到卫星服务器上以验证它们。

7. 在卫星服务器上，验证胶囊服务器证书输入文件：

```
# 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. 由您的认证机构提供的胶囊服务器证书文件
2. 用于为证书签名的胶囊服务器私钥
3. 由认证机构提供的证书颁发机构捆绑包

   如果将 **commonName=** 设置为通配符值 *，您必须将 **-t Capsule 选项添加到 katello-certs-check** 命令中。

   保留由 **katello-certs-check** 命令输出的示例 **capsule-certs-generate** 命令的副本，用于为这个胶囊服务器创建证书存档文件。

4.4.2. 使用自定义 SSL 证书配置胶囊服务器以使用 Puppet 负载平衡

如果在 Satellite 配置中使用 Puppet，则必须完成以下步骤：

1. 配置胶囊服务器以生成和签署 Puppet 证书
2. 为负载平衡配置重新平衡

配置胶囊服务器以生成和签署 Puppet 证书

对于您要配置用于负载平衡的所有其他胶囊服务器的系统，请完成此步骤。在这一流程中的示例中，此胶囊服务器的 FQDN 为 **capsule-ca.example.com**。

1. 将以下选项附加到您从 **katello-certs-check** 命令的输出中获取的 **capsule-certs-generate** 命令：

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

2. 在卫星服务器上，输入 **capsule-certs-generate** 命令来生成胶囊证书。例如：

```
# 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
```

   保留用于安装 Capsule Server 证书的输出中的 example **satellite-installer** 命令的副本。

3. 将证书存档文件从卫星服务器复制到胶囊服务器。

4. 将以下选项附加到 **satellite-installer** 命令中，从 **capsule-certs-generate** 命令的输出中获取：

```
--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. 在 Capsule Server 上，输入 **satellite-installer** 命令，例如：

```
satellite-installer --scenario capsule \
--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. 在 Capsule Server 上，为您配置用于负载平衡的所有其他胶囊生成 Puppet 证书，但这是您配置 Puppet 证书签名的第一个系统：

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

此命令在 Puppet 证书签名请求实例上创建以下文件：

- **/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**

为负载平衡配置重新平衡

为每个胶囊服务器完成这个步骤，包括配置胶囊服务器以为 Puppet 证书签名的系统。

1. 将以下选项附加到您从 **katello-certs-check** 命令的输出中获取的 **capsule-certs-generate** 命令：

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

2. 在卫星服务器上，输入 **capsule-certs-generate** 命令来生成胶囊证书。例如：

```
# 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
```

保留用于安装 Capsule Server 证书的输出中的 example **satellite-installer** 命令的副本。

3. 将证书存档文件从卫星服务器复制到胶囊服务器。

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

4. 在 Capsule Server 上，安装 **puppetserver** 软件包：

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

5. 在 Capsule Server 上，为 puppet 证书创建目录：

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

6. 在 Capsule Server 上，从配置胶囊服务器的系统中复制此胶囊服务器的 Puppet 证书，以为 Puppet 证书签名：

```
# 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. 在 Capsule Server 上，将目录所有权更改为用户 **puppet**、组 **puppet** 并设置 SELinux 上下文：

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

8. 将以下选项附加到 **satellite-installer** 命令中，从 **capsule-certs-generate** 命令的输出中获取：

```
--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. 在 Capsule Server 上，输入 **satellite-installer** 命令，例如：

```
# satellite-installer --scenario capsule \
--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
```

第 5 章 安装 LOAD BALANCER

以下示例提供了配置 HAProxy 负载均衡器的一般指南。但是，您可以安装支持 TCP 转发和粘性会话的任何合适的负载均衡软件解决方案。

1. 在 Red Hat Enterprise Linux 7 主机上安装 HAProxy：

```
# yum install haproxy
```

2. 安装包含 **semanage** 工具的以下软件包：

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

3. 将 SELinux 配置为允许 HAProxy 绑定任何端口：

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

4. 配置负载均衡器以平衡端口的网络负载，如 表 5.1 “Load Balancer 的端口配置” 所述。例如，要为 HAProxy 配置端口，请编辑 **/etc/haproxy/haproxy.cfg** 文件与表对应。
您必须在 TCP 端口 443 上配置粘性会话，从您为负载均衡配置的不同胶囊服务器请求 RPM 存储库的 yum 元数据。

表 5.1. Load Balancer 的端口配置

Service	端口	模式	平衡模式	目的地
HTTP	80	TCP	roundrobin	所有胶囊服务器上的端口 80
HTTPS	443	TCP	source	all Capsule Servers 上的端口 443
RHSM	8443	TCP	roundrobin	所有胶囊服务器上的端口 8443
AMQP	5647	TCP	roundrobin	所有胶囊服务器上的端口 5647
Puppet（*可选*）	8140	TCP	roundrobin	所有胶囊服务器上的端口 8140
PuppetCA (*Optional*)	8141	TCP	roundrobin	仅在配置 Capsule Server 以为 Puppet 证书签名的系统中的端口 8140
Cryostat（*可选，对于 OpenScap*）	9090	TCP	roundrobin	所有胶囊服务器上的端口 9090

5. 配置负载均衡器以禁用 SSL 卸载并允许客户端 SSL 证书传递到后端服务器。这是必要的，因为从客户端到胶囊服务器的通信取决于客户端 SSL 证书。

6. 启动并启用 HAProxy 服务：

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

第 6 章 注册客户端

您可以将运行 Red Hat Enterprise Linux 6、7 或 8 操作系统的客户端注册到您为负载均衡配置的胶囊服务器。有关注册客户端并将其配置为使用 Puppet 的更多信息，请参阅 *《管理主机》指南中的注册主机*。

要注册客户端，请执行以下步骤之一：

- 第 6.1 节 “使用 Bootstrap 脚本注册客户端”
- 第 6.2 节 “手动注册客户端”

6.1. 使用 BOOTSTRAP 脚本注册客户端

要注册客户端，请在客户端上输入以下命令。您必须为每个客户端完成注册步骤。

前提条件

确保在客户端上安装 bootstrap 脚本，并将脚本的文件权限更改为可执行。如需更多信息，请参阅 管理主机指南中的使用引导脚本将主机注册到 Red Hat Satellite。

- *在 Red Hat Enterprise Linux 8 中输入以下命令：*

```
# /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-6.10-rpms \
--puppet-ca-port 8141 \    (1)
--force                    (2)
```

(1) *如果使用 Puppet，请包含* ***--puppet-ca-port 8141*** *选项。*

(2) *包含* ***--force*** *选项，以注册之前注册到单机胶囊的客户端。*

- *在 Red Hat Enterprise Linux 7、6 或 5 中输入以下命令：*

```
# 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-6.10-rpms \
--puppet-ca-port 8141 \    (1)
--force                    (2)
```

(1) *如果使用 Puppet，请包含* ***--puppet-ca-port 8141*** *选项。*

(2) *包含* ***--force*** *选项，以注册之前注册到单机胶囊的客户端。*

*该脚本提示与您使用 **--login** 选项输入的 Satellite 用户名对应的密码。*

6.2. 手动注册客户端

要手动注册客户端，请在您注册的每个客户端上完成以下步骤。

流程

1. *如果已安装，请删除 **katello-ca-consumer** 软件包：*

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

2. *从负载均衡器安装 **katello-ca-consumer** 软件包：*

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

3. *注册客户端并包含 **--serverurl** 和 **--baseurl** 选项：*

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

第 7 章 将 SCAP 内容提升到客户端

下面的部分论述了如何将安全内容自动化协议(SCAP)内容提升至您为负载平衡配置的胶囊服务器的客户端。

先决条件

- 确保配置 SCAP 内容。如需更多信息，请参阅 管理 Red Hat Satellite 中的配置 SCAP 内容。

流程

1. 在 Satellite Web UI 中，导航到 **Configure** > **Classes**，再点击 **foreman_scap_client**。
2. 单击 **Smart Class Parameter** 选项卡。
3. 在 **智能类参数** 窗口左侧的窗格中，单击 **端口**。
4. 在 **Default Behavior** 区域，选中 **Override** 复选框。
5. 在 **Key Type** 列表中，选择 **整数**。
6. 在 **Default Value** 字段中，输入 **9090**。
7. 在 **智能类参数** 窗口左侧的窗格中，单击 **server**。
8. 在 **Default Behavior** 区域，选中 **Override** 复选框。
9. 在 **Key Type** 列表中，选择 **字符串**。
10. 在 **Default Value** 字段中输入负载均衡器的 FQDN。例如, **loadbalancer.example.com**.
11. 在 **Smart Class Parameter** 窗口左下角，单击 **Submit**。
12. 将包含 **foreman_scap_client** puppet 类的 puppet 模块添加到内容视图。发布此内容视图并将其提升到客户端的环境。
13. 如果要验证配置，请在客户端上运行 Puppet 代理以提升更改。不要在每个客户端手动运行 Puppet 代理，因为 Puppet 代理每 30 分钟在客户端上运行。

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

14. 在客户端中，验证 **/etc/foreman_scap_client/config.yaml** 文件是否包含以下行：

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

其它资源

- 有关添加 Puppet 模块到卫星服务器的更多信息，请参阅 Puppet 指南中的 将 Puppet 模块添加到 Red Hat Satellite 6 中。
- 有关内容视图的更多信息，请参阅 内容 管理指南中的管理内容视图。

第 8 章 验证负载平衡配置

您可以通过为每个配置的每个胶囊服务器完成以下步骤来验证负载平衡配置：

1. 关闭您的胶囊服务器的基础操作系统。
2. 验证内容或订阅管理功能在注册到此胶囊的客户端上可用。例如，在客户端中输入 **subscription-manager refresh** 命令。
3. 为您的胶囊服务器重启基础操作系统。