Red Hat Satellite 6.15

使用负载均衡器配置 Capsule

在 Capsules 之间分配负载

Last Updated: 2024-06-21

Red Hat Satellite 6.15 使用负载均衡器配置 Capsule

在 Capsules 之间分配负载

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

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。由于这项工作的艰巨性，这些变化正在尽可能地逐步更新。详情请查看 CTO Chris Wright 的信息。

对红帽文档提供反馈

我们感谢您对我们的文档提供的信息。请让我们了解如何改进文档。

您可以通过在 Bugzilla 中记录一个 ticket 来提交反馈：

1. 导航到 Bugzilla 网站。
2. 在 **Component** 字段中，使用 **Documentation**。
3. 在 **Description** 字段中，输入您要改进的建议。包括文档相关部分的链接。
4. 点 **Submit Bug**。

第 1 章 负载平衡解决方案架构

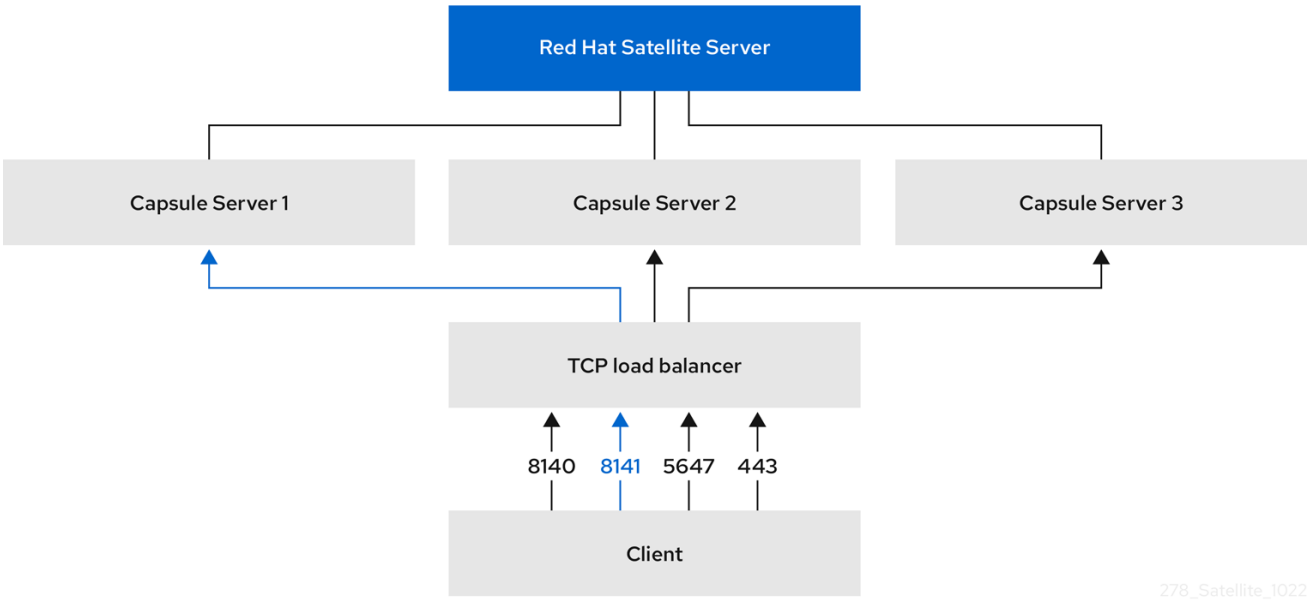
您可以将 Satellite 服务器配置为使用负载均衡器来在多个胶囊服务器之间分发客户端请求和网络负载。这会提高胶囊服务器上的整体性能。

本指南概述了如何准备 Satellite 服务器和胶囊服务器以进行负载平衡，并提供了有关如何在负载均衡设置中配置负载均衡器和注册客户端的指南。

负载均衡的设置由以下组件组成：

- Satellite 服务器
- 两个或多个 Capsule 服务器
- 负载均衡器
- 多个客户端

图 1.1. Satellite 负载均衡解决方案架构



在负载均衡设置中，对于计划或计划外维护，几乎所有胶囊功能都会继续按预期工作。负载均衡器可用于以下服务和特性：

- 使用 **subscription-manager**注册
- 使用 Yum 存储库管理内容
- 可选： Puppet

注意

在负载均衡设置中，负载均衡器只为上述服务和功能分配负载。如果其他服务（如 provisioning 或 virt-who）正在单独的 Capsules 上运行，则必须通过 Capsules 直接访问它们，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。尝试使用同一数据的多个写器进程可能会损坏它。

要管理此 Puppet 限制，请完成以下步骤：

1. 在一个胶囊服务器上配置 Puppet 证书签名请求，通常是配置胶囊服务器以进行负载平衡的第一个系统。
2. 配置客户端，将 CA 请求发送到负载均衡器上的端口 8141。
3. 配置负载平衡器，将端口 8141 的 CA 请求重定向到将胶囊服务器配置为签署 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点故障。将 Capsule 配置为使用负载均衡器可以提供对计划和计划外中断的恢复能力。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准则：

- 如果使用 Puppet，则会将 Puppet 证书签名请求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中的一个状态。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

将 Capsule 配置为使用负载均衡器会导致更复杂的环境，并需要额外的维护。

负载均衡需要额外的步骤：

- 您必须确保所有 Capsules 具有相同的内容视图，并将所有 Capsules 同步到相同的内容视图版本
- 您必须按顺序升级每个 Capsule
- 您必须备份您定期配置的每个 Capsule

在负载均衡配置中升级 Capsule 服务器

要将 Capsule 服务器从 6.14 升级到 6.15，请完成将 *Red Hat Satellite 升级到 6.15 中的 升级 胶囊服务器* 流程。在负载平衡配置中，胶囊服务器不需要额外的步骤。

第 3 章 为负载均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 *安装胶囊服务器* 中描述的以下步骤。Satellite 不支持为负载平衡配置现有的胶囊服务器。

1. 将 Capsule 服务器注册到 Satellite 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 Capsule 服务器软件包

第 4 章 配置胶囊服务器以进行负载平衡

本章概述了如何配置胶囊服务器以进行负载平衡。根据您的 Satellite 服务器配置，继续以下部分之一：

- 第 4.1 节 “使用默认 SSL 证书配置 Capsule 服务器，以便在没有 Puppet 的情况下进行负载平衡”
- 第 4.2 节 “使用默认 SSL 证书配置 Capsule 服务器，以使用 Puppet 进行负载平衡”
- 第 4.3.2 节 “使用自定义 SSL 证书配置 Capsule 服务器，以便在没有 Puppet 的情况下进行负载平衡”
- 第 4.4 节 “使用自定义 SSL 证书配置 Capsule 服务器，以使用 Puppet 进行负载平衡”

对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 CAPSULE 服务器，以便在没有 PUPPET 的情况下进行负载平衡

下面的部分论述了如何配置使用默认 SSL 证书进行负载平衡的胶囊服务器，而无需 Puppet。在您要为负载均衡配置的每个 Capsule 服务器上完成这个步骤。

流程

1. 在 Satellite 服务器上，为 Capsule 服务器生成 Katello 证书：

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

保留用于安装 Capsule 服务器证书的 Capsule- **certs-generate 命令输出的示例 satellite-installer** 命令的副本。

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. 将以下选项附加到您从 Capsule- **certs-generate 命令的输出中获取的 satellite- installer** 命令中：

```
--certs-cname "loadbalancer.example.com" \
--enable-foreman-proxy-plugin-remote-execution-script
```

4. 在 Capsule 服务器上，输入 **satellite-installer** 命令：

```
# satellite-installer --scenario capsule \
--certs-cname "loadbalancer.example.com" \
--certs-tar-file "capsule.example.com-certs.tar" \
--enable-foreman-proxy-plugin-remote-execution-script \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-oauth-consumer-key "oauth key" \
--foreman-proxy-oauth-consumer-secret "oauth secret" \
```

```
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule.example.com"
```

4.2. 使用默认 SSL 证书配置 CAPSULE 服务器，以使用 PUPPET 进行负载平衡

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

如果在 Satellite 配置中使用 Puppet，您必须完成以下步骤：

1. 第 4.2.1 节 “使用默认 SSL 证书配置胶囊服务器来生成和签署 Puppet 证书”
2. 第 4.2.2 节 “使用默认 SSL 证书配置剩余的 Capsule 服务器以进行负载平衡”

4.2.1. 使用默认 SSL 证书配置胶囊服务器来生成和签署 Puppet 证书

仅针对您要配置 Capsule 服务器的系统为您配置进行负载平衡的所有其他胶囊服务器生成并签署 Puppet 证书。

流程

1. 在 Satellite 服务器上，为配置 Capsule 服务器以生成和签署 Puppet 证书的系统生成 Katello 证书：

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

保留用于安装 Capsule 服务器证书的 Capsule- **certs-generate 命令输出的示例 satellite-installer** 命令的副本。

2. 将证书存档文件从 Satellite 服务器复制到 Capsule 服务器：

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

3. 将以下选项附加到您从 Capsule- **certs-generate 命令的输出中获取的 satellite- installer** 命令中：

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

4. 在 Capsule 服务器上，输入 **satellite-installer** 命令：

```
# satellite-installer --scenario capsule \
--certs-cname "loadbalancer.example.com" \
--certs-tar-file "capsule-ca.example.com-certs.tar" \
```

```
--enable-foreman-proxy-plugin-remote-execution-script \
--enable-puppet \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-oauth-consumer-key "oauth key" \
--foreman-proxy-oauth-consumer-secret "oauth secret" \
--foreman-proxy-puppetca "true" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule-ca.example.com" \
--puppet-ca-server "capsule-ca.example.com" \
--puppet-dns-alt-names "loadbalancer.example.com" \
--puppet-server true \
--puppet-server-ca "true"
```

5. 在 Capsule 服务器上停止 Puppet 服务器：

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

6. 为您为负载平衡配置的所有其他胶囊服务器生成 Puppet 证书，但配置 Puppet 证书的第一个系统除外：

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

此命令将配置 Capsule 服务器以签署 Puppet 证书的系统上创建以下文件：

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

4.2.2. 使用默认 SSL 证书配置剩余的 Capsule 服务器以进行负载平衡

在每个胶囊服务器上完成此流程，不包括将胶囊服务器配置为签署 Puppet 证书的系统。

流程

1. 在 Satellite 服务器上，为 Capsule 服务器生成 Katello 证书：

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

保留用于安装 Capsule 服务器证书的 Capsule- **certs-generate 命令输出的示例 satellite-installer** 命令的副本。

2. 将证书存档文件从 Satellite 服务器复制到 Capsule 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 服务器上，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 服务器上，从配置 Capsule 服务器的系统中复制此 Capsule 服务器的 Puppet 证书，以签署 Puppet 证书：

```
# 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/certs/ca.pem
/etc/puppetlabs/puppet/ssl/certs/ca.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 服务器上，将 **/etc/puppetlabs/puppet/ssl/** 目录所有权改为用户 **puppet** 和组 **puppet** ：

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

7. 在 Capsule 服务器上，为 **/etc/puppetlabs/puppet/ssl/** 目录设置 SELinux 上下文：

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

8. 将以下选项附加到您从 Capsule- **certs-generate 命令的输出中获取的 satellite- installer** 命令中：

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

9. 在 Capsule 服务器上，输入 **satellite-installer** 命令：

-

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

4.3. 使用自定义 SSL 证书配置 CAPSULE 服务器，以便在没有 PUPPET 的情况下进行负载平衡

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

4.3.1. 为 Capsule 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，并将负载均衡器和 Capsule 服务器作为 Subject Alternative Names (SAN)包含。在您要为负载均衡配置的每个 Capsule 服务器上完成这个步骤。

流程

1. 要存储所有源证书文件，请创建一个只能被 **root** 用户访问的目录：

```
# mkdir /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。如果您使用密码保护的私钥，请删除私钥密码。

   如果您已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. 为 CSR 创建 **/root/capsule_cert/openssl.cnf** 配置文件并包含以下内容：

```
[ req ]
req_extensions = v3_req
distinguished_name = req_distinguished_name
x509_extensions = usr_cert
prompt = no

[ req_distinguished_name ]
commonName = capsule.example.com (1)

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
```

```
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 Capsule 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，确保更改此设置。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. 可选： 如果要向 CSR 添加可辨识名称(DN)详情，请在 **[req_distinguished_name]** 部分添加以下信息：

```
[req_distinguished_name]
CN = capsule.example.com
countryName = My_Country_Name (1)
stateOrProvinceName = My_State_Or_Province_Name (2)
localityName = My_Locality_Name (3)
organizationName = My_Organization_Or_Company_Name
organizationalUnitName = My_Organizational_Unit_Name (4)
```

1. 两个字母代码
2. 全名
3. 全名（例如：New York）
4. 负责证书的部门（示例：IT 部门）

5. 生成 CSR：

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

1. 私钥的路径
2. 配置文件的路径
3. 要生成的 CSR 的路径

6. 将证书签名请求发送到证书颁发机构(CA)。同一 CA 必须为 Satellite 服务器和 Capsule 服务器签名证书。
   提交请求时，指定证书的寿命。发送证书请求的方法会有所不同，因此请查阅 CA 查看首选方法。为了响应请求，您可以在单独的文件中接收 CA 捆绑包和签名证书。

7. 将您从证书颁发机构接收的证书颁发机构和胶囊服务器证书文件复制到您的 Satellite 服务器。

8. 在 Satellite 服务器上，验证 Capsule 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. 由您的证书颁发机构提供的 Capsule 服务器证书文件
2. 用于为证书签名的 Capsule 服务器私钥
3. 由您的证书颁发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。

保留示例 **Capsule -certs-generate** 命令的副本，该命令由 **katello-certs-check** 命令输出，以便为此胶囊服务器创建证书归档文件。

4.3.2. 使用自定义 SSL 证书配置 Capsule 服务器，以便在没有 Puppet 的情况下进行负载平衡

下面的部分论述了如何配置使用自定义 **SSL** 证书进行负载平衡的胶囊服务器，而无需 **Puppet**。在您要为负载均衡配置的每个 **Capsule** 服务器上完成这个步骤。

流程

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

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

2. 在 **Satellite** 服务器上，输入 **Capsule -certs-generate** 命令来生成 **Capsule** 证书：

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

从输出保留示例 **satellite-installer** 命令的副本，用于安装 **Capsule** 服务器证书。

3. 将证书存档文件从 Satellite 服务器复制到 Capsule 服务器：

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

4. 将以下选项附加到您从 Capsule- certs-generate 命令的输出中获取的 satellite- installer 命令中：

```
--certs-cname "loadbalancer.example.com" \
--enable-foreman-proxy-plugin-remote-execution-script
```

5. 在 Capsule 服务器上，输入 satellite-installer 命令：

```
# satellite-installer --scenario capsule \
--certs-cname "loadbalancer.example.com" \
--certs-tar-file "capsule.example.com-certs.tar" \
--enable-foreman-proxy-plugin-remote-execution-script \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-oauth-consumer-key "oauth key" \
--foreman-proxy-oauth-consumer-secret "oauth secret" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule.example.com"
```

4.4. 使用自定义 SSL 证书配置 CAPSULE 服务器，以使用 PUPPET 进行负载平衡

如果在 Satellite 配置中使用 Puppet，则必须完成以下步骤：

1. 第 4.4.2 节 “使用自定义 SSL 证书配置 Capsule 服务器来生成和签署 Puppet 证书”

2. 第 4.4.3 节 “使用自定义 SSL 证书配置剩余的 Capsule 服务器以进行负载平衡”

4.4.1. 为 Capsule 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，并将负载均衡器和 Capsule 服务器作为 Subject Alternative Names (SAN)包含。在您要为负载均衡配置的每个 Capsule 服务器上完成这个步骤。

流程

1. 要存储所有源证书文件，请创建一个只能被 root 用户访问的目录：

```
# mkdir /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。如果您使用密码保护的私钥，请删除私钥密码。

如果您已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. 为 CSR 创建 /root/capsule_cert/openssl.cnf 配置文件并包含以下内容：

```
[ req ]
req_extensions = v3_req
distinguished_name = req_distinguished_name
x509_extensions = usr_cert
prompt = no

[ req_distinguished_name ]
commonName = capsule.example.com (1)

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
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 Capsule 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，确保更改此设置。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. **可选：如果要向 CSR 添加可辨识名称(DN)详情，请在 [req_distinguished_name] 部分添加以下信息：**

```
[req_distinguished_name]
CN = capsule.example.com
countryName =My_Country_Name (1)
stateOrProvinceName = My_State_Or_Province_Name (2)
localityName = My_Locality_Name (3)
organizationName = My_Organization_Or_Company_Name
organizationalUnitName = My_Organizational_Unit_Name (4)
```

(1) **两个字母代码**

(2) **全名**

(3) **全名（例如：New York）**

(4) **负责证书的部门（示例：IT 部门）**

5. **生成 CSR：**

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

(1) **私钥的路径**

(2) **配置文件的路径**

(3)

要生成的 CSR 的路径

6. 将证书签名请求发送到证书颁发机构(CA)。同一 CA 必须为 Satellite 服务器和 Capsule 服务器签名证书。

提交请求时，指定证书的寿命。发送证书请求的方法会有所不同，因此请查阅 CA 查看首选方法。为了响应请求，您可以在单独的文件中接收 CA 捆绑包和签名证书。

7. 将您从证书颁发机构接收的证书颁发机构和胶囊服务器证书文件复制到您的 Satellite 服务器。

8. 在 Satellite 服务器上，验证 Capsule 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 由您的证书颁发机构提供的 Capsule 服务器证书文件

2 用于为证书签名的 Capsule 服务器私钥

3 由您的证书颁发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。

保留示例 Capsule -certs-generate 命令的副本，该命令由 katello-certs-check 命令输出，以便为此胶囊服务器创建证书归档文件。

4.4.2. 使用自定义 SSL 证书配置 Capsule 服务器来生成和签署 Puppet 证书

仅针对您要配置 Capsule 服务器的系统，为您配置的所有其他 Capsule 服务器生成 Puppet 证书，以便进行负载平衡。

流程

1. 将以下选项附加到从 katello -certs-check 命令的输出中获取的 Capsule-certs-generate 命令中：

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

2. 在 Satellite 服务器上，输入 Capsule -certs-generate 命令来生成 Capsule 证书：

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

从输出保留示例 satellite-installer 命令的副本，用于安装 Capsule 服务器证书。

3. 将证书存档文件从卫星服务器复制到胶囊服务器。

4. 将以下选项附加到您从 Capsule- certs-generate 命令的输出中获取的 satellite- installer 命令中：

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

5. 在 Capsule 服务器上，输入 satellite-installer 命令：

```
# satellite-installer --scenario capsule \
--certs-cname "loadbalancer.example.com" \
--certs-tar-file "certs.tgz" \
--enable-foreman-proxy-plugin-remote-execution-script \
--enable-puppet \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
```

```
--foreman-proxy-oauth-consumer-key "oauth key" \
--foreman-proxy-oauth-consumer-secret "oauth secret" \
--foreman-proxy-puppetca "true" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule-ca.example.com" \
--puppet-ca-server "capsule-ca.example.com" \
--puppet-dns-alt-names "loadbalancer.example.com" \
--puppet-server true \
--puppet-server-ca "true"
```

6. **在 Capsule 服务器上，为您配置的所有其他 Capsule 生成 Puppet 证书，以用于负载平衡，但第一个配置 Puppet 证书签名请求的系统除外：**

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

此命令在 Puppet 证书签名请求上创建以下文件：

- **/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**

4.4.3. 使用自定义 SSL 证书配置剩余的 Capsule 服务器以进行负载平衡

为每个胶囊服务器完成此流程，不包括将胶囊服务器配置为签署 Puppet 证书的系统。

流程

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

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

2. **在 Satellite 服务器上，输入 Capsule -certs-generate 命令来生成 Capsule 证书：**

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

从输出保留示例 satellite-installer 命令的副本，用于安装 Capsule 服务器证书。

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 服务器上，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 服务器上，从配置 Capsule 服务器的系统中复制此 Capsule 服务器的 Puppet 证书，以签署 Puppet 证书：**

```
# 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/certs/ca.pem
/etc/puppetlabs/puppet/ssl/certs/ca.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 服务器上，将 /etc/puppetlabs/puppet/ssl/ 目录所有权改为用户 puppet 和组 puppet ：

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

8. 在 Capsule 服务器上，为 /etc/puppetlabs/puppet/ssl/ 目录设置 SELinux 上下文：

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

9. 将以下选项附加到您从 Capsule- certs-generate 命令的输出中获取的 satellite- installer 命令中：

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

10. 在 Capsule 服务器上，输入 satellite-installer 命令：

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

第 5 章 为主机注册设置负载均衡器

在使用主机注册功能时，您可以将 Satellite 配置为通过负载均衡器注册客户端。

您将能够将主机注册到负载平衡器，而非胶囊。负载均衡器将决定在请求时注册主机的胶囊。在注册后，主机上的订阅管理器将被配置为通过负载均衡器管理内容。

前提条件

- 您已在所有 Capsule 服务器上配置了 SSL 证书。如需更多信息，请参阅 第 4 章 *配置胶囊服务器以进行负载平衡*。
- 在所有 Capsule 服务器上都启用了注册和模板插件：

```
# satellite-installer --scenario capsule \
--foreman-proxy-registration true \
--foreman-proxy-templates true
```

流程

1. 在所有 Capsule 服务器上，使用 satellite-installer 设置注册和模板 URL：

```
# satellite-installer --scenario capsule \
--foreman-proxy-registration-url "https://loadbalancer.example.com:9090" \
--foreman-proxy-template-url "https://loadbalancer.example.com:8000"
```

2. 在 Satellite Web UI 中，进入到 Infrastructure > Capsules。
3. 对于每个胶囊，单击 Actions 列中的下拉菜单，再选择 Refresh。

第 6 章 安装负载均衡器

以下示例提供了使用 Red Hat Enterprise Linux 8 服务器配置 HAProxy 负载均衡器的通用指导。但是，您可以安装支持 TCP 转发的任何合适的负载平衡软件解决方案。

流程

1. 安装 HAProxy：

```
# dnf install haproxy
```

2. 安装以下包含 semanage 工具的软件包：

```
# dnf install policycoreutils-python-utils
```

3. 配置 SELinux 以允许 HAProxy 绑定任何端口：

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

4. 将负载均衡器配置为平衡端口的网络负载，如 表 6.1 “负载均衡器的端口配置” 所述。例如，要为 HAProxy 配置端口，请编辑 /etc/haproxy/haproxy.cfg 文件以与表对应。如需更多信息，*请参阅红帽知识库中的* 带有 Satellite 6 的 HAProxy 负载均衡器的 haproxy.cfg 配置示例。

表 6.1. 负载均衡器的端口配置

Service	端口	模式	平衡模式	目的地
HTTP	80	TCP	roundrobin	所有 Capsule 服务器上的端口 80
HTTPS 和 RHSM	443	TCP	source	所有 Capsule 服务器上的端口 443
Anaconda 用于模板检索	8000	TCP	roundrobin	所有胶囊服务器上的端口 8000
Puppet (*可选*)	8140	TCP	roundrobin	所有胶囊服务器上的端口 8140
PuppetCA (*可选*)	8141	TCP	roundrobin	只有在将 Capsule 服务器配置为签署 Puppet 证书的系统中，端口 8140

Service	端口	模式	平衡模式	目的地
用于主机注册和可选 OpenSCAP 的 Capsule HTTPS	9090	TCP	roundrobin	所有 Capsule 服务器上的端口 9090

5. **将负载均衡器配置为禁用 SSL 卸载，并允许客户端 SSL 证书传递给后端服务器。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 enable --now haproxy
```

第 7 章 验证负载平衡配置

使用这个流程验证每个 Capsule 服务器的负载平衡配置。

流程

1. 关闭 Capsule 服务器的基本操作系统。

2. 验证注册到此胶囊的客户端上是否有内容或订阅管理功能。例如，在客户端上输入 **subscription-manager refresh** 命令。

3. 为您的胶囊服务器重启基础操作系统。

第 8 章 将客户端注册到负载均衡器

要平衡来自客户端的网络流量的负载，您必须将客户端注册到负载均衡器。

要注册客户端，请执行以下步骤之一：

- 第 8.1 节 “使用主机注册注册客户端”
- 第 8.2 节 “（已弃用） 使用 bootstrap 脚本注册客户端”

8.1. 使用主机注册注册客户端

您可以使用 Satellite Web UI、Hammer CLI 或 Satellite API 中的主机注册功能在 Satellite 中注册主机。如需更多信息，请参阅管理 主机 中的 *注册主机* 。

先决条件

- 您已为主机注册设置了负载均衡器。如需更多信息，请参阅 第 5 章 *为主机注册设置负载均衡器*。

流程

1. 在 Satellite Web UI 中，进入到 Hosts > Register Host。
2. 从 Capsule 下拉列表中，选择负载平衡器。
3. 选择 Force 来注册之前注册到胶囊服务器的主机。
4. 从 Activation Keys 列表中，选择要分配给您的主机的激活密钥。
5. 点 Generate 来创建注册命令。

6. 点 *files* 图标将命令复制到您的剪贴板。

7. 使用 SSH 连接到您的主机，并运行注册命令。

8. 检查 /etc/yum.repos.d/redhat.repo 文件，并确保启用了适当的存储库。

CLI 过程

1. 使用 Hammer CLI 生成主机注册命令：

```
# hammer host-registration generate-command \
--activation-keys "My_Activation_Key"
```

如果您的主机不信任 Satellite 服务器的 SSL 证书，您可以通过在注册命令中添加 **--insecure** 标志来禁用 SSL 验证。

```
# hammer host-registration generate-command \
--activation-keys "My_Activation_Key" \
--insecure true
```

包含 **--smart-proxy-id** *My_Capsule_ID* 选项。您可以使用您为主机注册负载均衡配置的任何胶囊服务器的 ID。Satellite 将自动将负载平衡器应用到注册命令。

包含 **--force** 选项，以注册之前注册到胶囊服务器的主机。

2. 使用 SSH 连接到您的主机，并运行注册命令。

3. 检查 /etc/yum.repos.d/redhat.repo 文件，并确保启用了适当的存储库。

API 过程

1. 使用 Satellite API 生成主机注册命令：

```
# curl -X POST https://satellite.example.com/api/registration_commands \
--user "My_User_Name" \
```

```
-H 'Content-Type: application/json' \
-d '{ "registration_command": { "activation_keys": ["My_Activation_Key_1,
My_Activation_Key_2"] }}'
```

如果您的主机不信任 Satellite 服务器的 SSL 证书，您可以通过在注册命令中添加 --insecure 标志来禁用 SSL 验证。

```
# curl -X POST https://satellite.example.com/api/registration_commands \
--user "My_User_Name" \
-H 'Content-Type: application/json' \
-d '{ "registration_command": { "activation_keys": ["My_Activation_Key_1,
My_Activation_Key_2"], "insecure": true }}'
```

使用激活码来简化指定环境。如需更多信息，请参阅管理内容中的管理 *激活码* 。

包含 { "smart_proxy_id": *My_Capsule_ID* }。您可以使用您为主机注册负载均衡配置的任何胶囊服务器的 ID。Satellite 将自动将负载平衡器应用到注册命令。

包含 { "force": true }，以注册之前注册到胶囊服务器的主机。

要将密码作为命令行参数输入密码，请使用 username:password 语法。请记住，这会在 shell 历史记录中保存密码。或者，您可以使用临时个人访问令牌而不是密码。要在 Satellite Web UI 中生成令牌，请导航到 My Account > Personal Access Tokens。

2. **使用 SSH 连接到您的主机，并运行注册命令。**

3. 检查 **/etc/yum.repos.d/redhat.repo 文件，并确保启用了适当的存储库。**

8.2. （已弃用） 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。

先决条件

- **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，请参阅管理主机中的 使用启动脚本将主机注册到 Red Hat Satellite。**

流程

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

```
# /usr/libexec/platform-python bootstrap.py \
--activationkey="My_Activation_Key" \
--enablerepos=satellite-client-6-for-rhel-8-<arch>-rpms \ 1
--force \ 2
--hostgroup="My_Host_Group" \
--location="My_Location" \
--login=admin \
--organization="My_Organization" \
--puppet-ca-port 8141 \ 3
--server loadbalancer.example.com
```

1. 将 **<arch >** 替换为客户端架构，如 **x86**。
2. 包含 **--force** 选项，以注册之前注册到独立胶囊的客户端。
3. 如果使用 **Puppet**，请包含 **--puppet-ca-port 8141** 选项。

- 在 Red Hat Enterprise Linux 7 或 6 中，输入以下命令：

```
# python bootstrap.py --login=admin \
--activationkey="My_Activation_Key" \
--enablerepos=rhel-7-server-satellite-client-6-rpms \
--force \ 1
--hostgroup="My_Host_Group" \
--location="My_Location" \
--organization="My_Organization" \
--puppet-ca-port 8141 \ 2
--server loadbalancer.example.com
```

1. 包含 **--force** 选项，以注册之前注册到独立胶囊的客户端。
2.

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

第 9 章 通过负载均衡器传播 SCAP 内容

如果使用 OpenSCAP 来管理客户端的安全合规性，您必须将 SCAP 客户端配置为将 ARF 报告发送到负载均衡器而不是 Capsule。配置过程取决于您选择部署合规策略的方法。

9.1. 使用 ANSIBLE 部署传播 SCAP 内容

使用这个流程，您可以通过 Ansible 部署方法范围内的负载均衡器提升安全内容自动化协议(SCAP)内容。

先决条件

- 确保您已为 Ansible 部署策略配置了 Satellite。如需更多信息，请参阅管理安全合规 中的配置合规策略部署方法。

流程

1. 在 Satellite web UI 中，进入到 Configure > Ansible > Variables。
2. 搜索 foreman_scap_client_port 变量，再单击其名称。
3. 在 Default Behavior 区域中，确保选择了 Override 复选框。
4. 在 Parameter Type 列表中，确保选择了 整数。
5. 在 Default Value 字段中，输入 9090。
6. 在 Specify Matchers 区域中，删除所有覆盖默认值的 matchers。
7. 点 Submit。
8. 搜索 foreman_scap_client_server 变量，再单击其名称。

9. 在 Default Behavior 区域中，确保选择了 Override 复选框。

10. 在 Parameter Type 列表中，确保选择了 字符串。

11. 在 Default Value 字段中，输入负载均衡器的 FQDN，如 loadbalancer.example.com。

12. 在 Specify Matchers 区域中，删除所有覆盖默认值的 matchers。

13. 点 Submit。

14. 使用 Ansible 继续部署合规策略。如需更多信息，请参阅：

    - 管理安全合规 中的使用 Ansible 在主机组中部署策略
    - 在管理 安全合规中的使用 Ansible 在主机上部署策略

验证

- 在客户端上，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
```

9.2. 使用 PUPPET 部署传播 SCAP 内容

使用此流程，您可以通过 Puppet 部署方法范围内的负载均衡器提升安全内容自动化协议(SCAP)内容。

先决条件

- 确保您为 Puppet 部署策略配置了 Satellite。如需更多信息，请参阅管理安全合规 中的配置合规策略部署方法。

流程

1. *在 Satellite Web UI 中，进入到 Configure > Puppet ENC > Classes。*
2. *单击 foreman_scap_client。*
3. *单击 Smart Class Parameter 选项卡。*
4. *在 Smart Class Parameter 窗口左侧的窗格中，单击 端口。*
5. *在 默认行为 区域中，选中 覆盖 复选框。*
6. *从 Key Type 列表中，选择 整数。*
7. *在 Default Value 字段中，输入 9090。*
8. *在 Smart Class Parameter 窗口左侧的窗格中，单击 server。*
9. *在 默认行为 区域中，选中 覆盖 复选框。*
10. *从 Key Type 列表中，选择 字符串。*
11. *在 Default Value 字段中，输入负载均衡器的 FQDN，如 loadbalancer.example.com。*
12. *在 Smart Class Parameter 窗口左下角，单击 Submit。*
13. *使用 Puppet 继续部署合规策略。如需更多信息，请参阅：*

- *使用 Puppet 在主机组中部署策略，以管理安全合规性*

- *使用 Puppet 在主机上部署策略，以管理安全合规性*

验证

- *在客户端上，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
```