Red Hat Satellite 6.4

ロードバランシングガイド

負荷を分散した Capsule プールの設定

Last Updated: 2019-01-10

Red Hat Satellite 6.4 ロードバランシングガイド

負荷を分散した Capsule プールの設定

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

法律上の通知

Copyright © 2019 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

ロードバランシングガイドは、負荷を分散した Red Hat Satellite Capsule Server のプールを設定する方法について説明します。本書は、ネットワークに関する知識とスキルを十分に持つ Satellite 管理者を主な対象としています。

目次

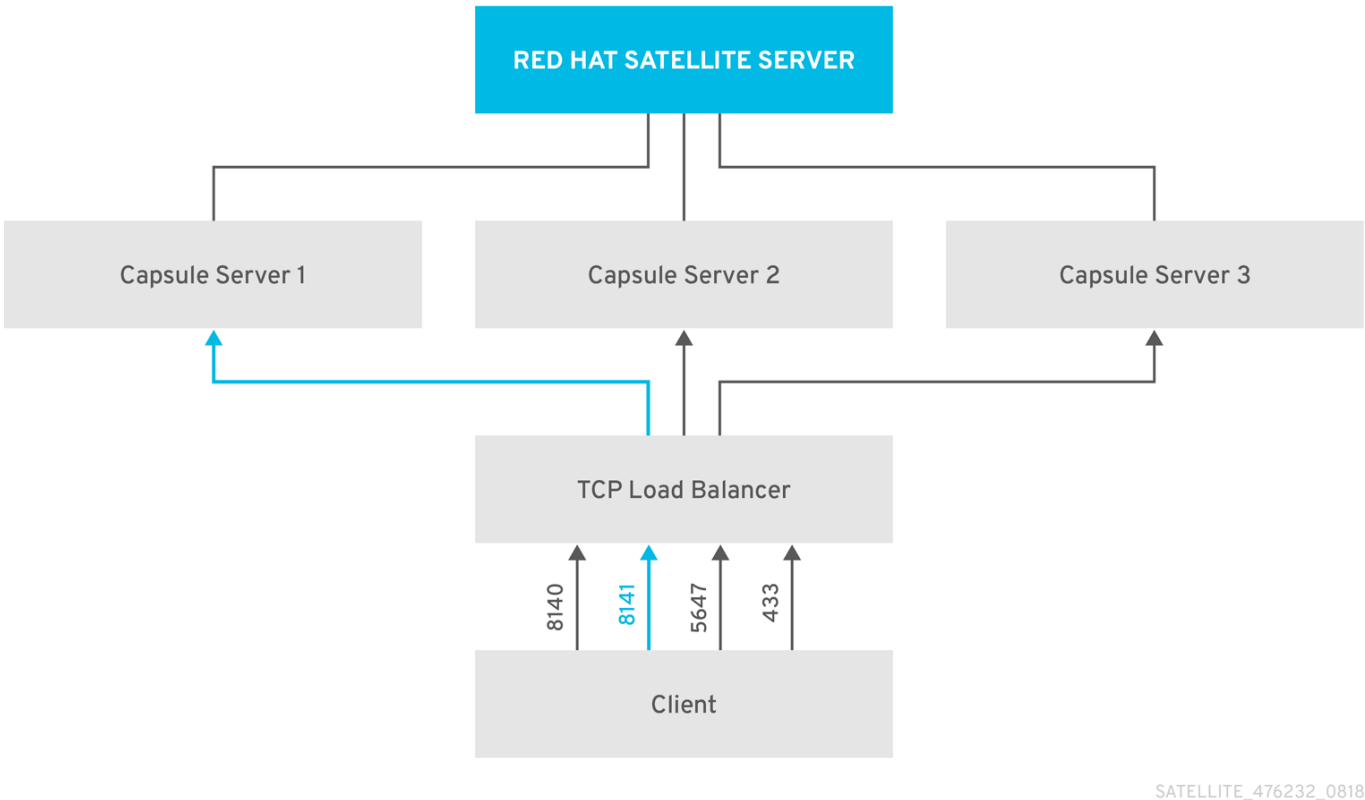
1. 負荷分散ソリューションの概要

Satellite Server 環境は、クラスター全体の複数の Capsule Server インスタンスの負荷を分散させる形で設定することが可能です。これを実現するには、ロードバランサーを設定する必要があります。クライアントのリクエストとネットワーク負荷が効率的に分散することでパフォーマンスが向上するため、Capsule Server インスタンスの負担が軽減されます。

負荷分散型の Capsule プール構成は通常、以下のコンポーネントで構成されます。

- 既存の Satellite 6.4 環境
- 負荷分散クラスター用の複数の Capsule Server インスタンス
- ロードバランサー
- 複数のクライアント

図1 Satellite 負荷分散ソリューションのアーキテクチャー



負荷分散型の設定では、ある Capsule Server インスタンスが定期または予定外のメンテナンスで停止されても、ほとんどすべての Capsule 機能は想定通りに動作し続けます。このソリューションでは、以下のサービスと機能の負荷が分散されます。

- **`subscription-manager`** を使用した登録
- コンテンツ管理 (**`yum`** リポジトリー)
- Puppet (オプション)

注記

Puppet モジュールを使用している場合は、Puppet 認証機関 (CA) 管理が負荷分散型の証明書署名に対応していないという既知の制限があります。Puppet CA はシリアル番号カウンターや証明書失効リストといった証明書情報をファイルシステムに保存するため、複数の書き込みプロセスが同一データを使おうとすると、データが簡単に破損するおそれがあります。

この Puppet の制限に対する回避策として、このソリューションでは以下のステップを実行してください。

1. ある Capsule Server インスタンス (通常はクラスター内の最初のインスタンス) で Puppet 証明書署名を設定する。
2. クライアントマシン上で、CA リクエスト送信をロードバランサーのポート 8141 に設定する。
3. ロードバランサーで、このポートを Puppet 証明書署名機能のある Capsule Server インスタンス (通常はクラスター内の最初のインスタンス) 上の 8140 にリダイレクトする。

Satellite Server および Capsule Server 環境の準備方法、ロードバランサーの設定方法、クライアントの登録方法についての詳細情報は、本ガイドの以下の章を参照してください。

2. 作業開始前の準備

アプリケーションリクエストを分散すると可用性と応答性が高まることと、サーバーの負荷を減らすことができるほか、特定の Capsule が単一障害点になることを防ぐこともできます。負荷分散の Capsule プールを設定すると、定期および予定外の停止に対する復元力が備えられます。

本ガイドのソリューションを実装する前に、以下の点を確認してください。

- Puppet を使用している場合は、Puppet 証明書署名がプール内の最初の Capsule に割り当てられることに注意してください。このプール内の最初の Capsule がダウンすると、クライアントは Capsule から Puppet コンテンツを取得できなくなります。この場合 Puppet 証明書署名は、最初の Capsule がサービス提供可能な状態に戻ると利用できるようになります。
- 本ガイドのソリューションでは、全 Capsule をある状態に維持するために pacemaker や他の同様の HA ツールは使用しません。問題のトラブルシュートに際しては、ロードバランサーを迂回して、各 Capsule 上で問題を再生成する必要があります。
- 負荷を分散した Capsule プールを設定すると、より複雑な環境となり、新たなメンテナンスが必要になります。
  - すべての Capsule で同じコンテンツビューがあり、同一のコンテンツビューバージョンに同期されている必要があります。
  - 各 Capsule を順にアップグレードする必要があります。
  - プール内の各 Capsule で定期的なバックアップを行う必要があります。

3. SATELLITE SERVER および CAPSULE SERVER での負荷分散への準備

本章では、負荷を分散した Capsule Server プールを構成する Satellite Server と Capsule Server マシンの準備方法について説明します。

Satellite Server の準備

このソリューションでは、必要最小限の Satellite Server のバージョンをインストールする必要があります。Satellite Server 6.4 以上のバージョンが必要になります。Satellite Server のインストールに関する詳細情報は、オンラインネットワークからの SATELLITE SERVER のインストール を参照してください。

Capsule Server の準備

負荷を分散したプールでの Capsule Server マシンを準備するには、**Red Hat Satellite Capsule Server のインストール** にある以下の手順を完了する必要があります。

1. Satellite Server への登録
2. Capsule Server サブスクリプションの識別と割り当て
3. リポジトリーの設定
4. 時間の同期
5. CAPSULE SERVER のインストール

3.1. 負荷分散型 Capsule Server インストール

本章では、負荷を分散した Capsule Server プールを構成するマシンでの Capsule Server のインストール方法について説明します。

Puppet を使用している場合は、Puppet の既知の制限のため、ある Capsule Server インスタンス (通常はクラスター内の最初のインスタンス) で証明書署名を設定する必要があります。このため、以下の 2 つのタスクを実行してください。

1. 証明書署名の Capsule Server インスタンス (通常はクラスター内の最初のインスタンス) をインストールする。
2. クラスター内の残りの Capsule Server インスタンスをインストールする。

Puppet を使用していない場合は、残りの Capsule Server のインストールを完了する にある説明に従って Capsule Server をインストールし、Puppet 署名生成のステップは省略します。

前提条件

「Satellite Server および Capsule Server での負荷分散への準備」 の章にある Capsule Server 準備のステップを終えていること。

証明書署名の Capsule Server のインストールを完了するには、以下の手順に従います。

1. Satellite Server で Katello 証明書を生成します。**`--foreman-proxy-cname`** オプションにロードバランサーのホスト名を含めて証明書アーカイブを作成します。

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

   このコマンドを実行すると、毎回一意の **`satellite-installer`** コマンドが生成されます。後で使用するので、これを書き留めます。

2. Capsule Server 上で、**`custom-hiera.yaml`** ファイルに以下の行を追加します。

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

3. ステップ 1 で生成された **satellite-installer** コマンドに、以下の行を追加します。

```
--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. Capsule Server で、編集した **satellite-installer** コマンドを実行します。例を示します。

```
# 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. 残りの Capsule で Puppet 証明書を生成します。プール内の最初の Capsule に Puppet 証明書署名が割り当てられているので、プール内のそれ以外の Capsule 用に Puppet 証明書を生成する必要があります。Capsule Server で以下のコマンドを実行します。

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

完全修飾ドメイン名 (FQDN) を編集して、ロードバランスプール内の残りの各 Capsule でこのコマンドを実行します。

残りの Capsule Server のインストールを完了する

1. Satellite Server で Katello 証明書を生成します。**--foreman-proxy-cname** オプションにロードバランサー名を含めて証明書アーカイブを作成します。

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

このコマンドを実行すると、毎回一意の **satellite-installer** コマンドが生成されます。後で使用するので、これを書き留めます。

2. Puppet 証明書を生成します。Capsule Server で以下のステップを実行します。

   i. **puppetserver** RPM をインストールします。

   ii. 最初の Capsule Server インスタンスから残りの各インスタンス (つまり、2 つ目、3 つ目など) に以下のファイルをコピーします。各 Capsule に合わせて FQDN を編集します。

   - **/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. 以下のコマンドを実行して、ファイルの所有者がユーザー **puppet**、グループ **puppet** になるようにします。

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

   iv. 以下のコマンドを実行して、SELinux コンテキストを適切に設定します。

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

3. Capsule Server 上で、**custom-hiera.yaml** ファイルに以下の行を追加します。

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

4. ステップ 1 で生成された **satellite-installer** コマンドに、以下の行を追加します。

```
--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. Capsule Server で、編集した **satellite-installer** コマンドを実行します。例を示します。

```
# 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. ロードバランサーのインストール

以下のセクションでは、HAProxy ロードバランサーの設定方法に関する一般的な例を示します。TCP 転送およびスティッキーセッションをサポートしているロードバランサーソフトウェアソリューションであれば、どのロードバランサーをインストールしても構いません。

1. Red Hat Enterprise Linux 7 ホストで、HAProxy RPM をインストールします。
2. SELinux で HAProxy がどのポートでもバインドできるように設定します。

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

3. **/etc/haproxy/haproxy.cfg** ファイルを以下のように設定して、すべてのポートのバランスをとります。

表1 HAProxy のポート設定

サービス	ポート	モード	バランスモード	宛先
HTTP	80	TCP	roundrobin	全 Capsule のポート 80
HTTPS	443	TCP	source	全 Capsule のポート 443
RHSM	8443	TCP	roundrobin	全 Capsule のポート 8443
AMQP	5647	TCP	roundrobin	全 Capsule のポート 5647
Puppet (**オプション**)	8140	TCP	roundrobin	全 Capsule のポート 8140
PuppetCA (**オプション**)	8141	TCP	roundrobin	最初の Capsule のポート 8140
SmartProxy (**OpenScap のオプション**)	9090	TCP	roundrobin	全 Capsule のポート 9090
Docker (**オプション**)	5000	TCP	roundrobin	全 Capsule のポート 5000

PuppetCA 用の追加ポート (ポート 8141) は最初の Capsule にのみ転送されることに注意してください。

5. クライアントの登録

Red Hat Enterprise Linux バージョン 6 または 7 のベースシステムで稼働しているクライアントはどれも、負荷分散型 Capsule プールに登録することができます。クライアントが以前にスタンドアロンの Capsule に登録していた場合は、そのクライアントを負荷分散型 Capsule プールに登録する必要があります。

クライアントを負荷分散型 Capsule プールに登録するには、以下の 2 つの方法があります。

- ブートストラップスクリプトを使った登録
- 手動でのクライアントの登録

ブートストラップスクリプトを使った登録方法が推奨されます。

5.1. ブートストラップスクリプトを使ったクライアントの登録

前提条件

クライアントマシンにブートストラップスクリプトをインストールして、実行可能としていること。詳細については、**ホストの管理** の ブートストラップスクリプトを使用したホストの Satellite 6 への登録 セクションを参照してください。

ブートストラップスクリプトを使ったクライアント登録

クライアントを負荷分散型 Capsule に登録するには、以下のコマンドをクライアントマシンで実行します。

```
# 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. hostgroup がすべての OS およびプロビジョニング詳細で設定されていない場合は、**--unmanaged** オプションを含めます。
2. Puppet を使用している場合は、**--puppet-ca-port 8141** オプションを含めます。
3. 登録するクライアントが以前にスタンドアロン Capsule に登録されていた場合は、**--force** オプションを含めます。

このスクリプトでは、**--login** オプションを使用して入力した Satellite ユーザー名のパスワード入力が求められます。

注記

登録手続きはすべてのクライアントマシンで完了する必要があります。

5.2. 手動でのクライアントの登録

1. ホストに katello-ca-consumer パッケージがインストールされている場合は、これを削除する必要があります。クライアントマシンで以下のコマンドを実行してください。

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

2. 以下のコマンドを実行して katello-ca-consumer パッケージをインストールします。

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

3. **--serverurl** と **--baseurl** オプションを含めて以下のコマンドを実行し、システムを登録します。

```
# 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. 以下のコマンドを実行して `puppet-agent` をインストールします。

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

5. **/etc/puppetlabs/puppet/puppet.conf** ファイルの agent セクションに以下の行を追加します。

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

6. **--noop** オプションを **puppet agent** コマンドに含めて、実行します。

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

7. Satellite Server web UI で以下のステップを実行して、Puppet 用の SSL 証明書に署名します。
   a. Satellite Server Web UI にログインします。
   b. **インフラストラクチャー** > **Capsule** に移動します。
   c. **capsule01** の **アクション** コラムで **編集** リストをクリックして **証明書** を選択します。
   d. **署名** をクリックします。
   e. 再度 **puppet agent** コマンドを実行して、証明書の署名後に機能することを確かめます。

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

6. ロードバランサー設定の確認

設定が完了したら、以下のアクションを実行して確認します。

1. 負荷分散型 Capsule プールにクライアントマシンを登録します。
2. いずれかの Capsule Server インスタンスをシャットダウンします。
3. クライアントでコンテンツまたはサブスクリプション管理機能が利用可能であることを確かめます。たとえば、**`subscription-manager refresh`** コマンドや **`yum`** コマンドを実行します。
4. シャットダウンした Capsule Server インスタンスを再起動します。
5. 負荷分散型プール内のすべての Capsule Server でステップ 2 - 4 を繰り返します。