



Red Hat Satellite 6.5

オンラインネットワークからの Satellite Server のインストール

オンラインネットワークからの Red Hat Satellite Server のインストール

Red Hat Satellite 6.5 オンラインネットワークからの Satellite Server のインストール

オンラインネットワークからの Red Hat Satellite Server のインストール

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, the Red Hat 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 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

本書では、オンラインネットワークから Red Hat Satellite Server のインストール方法、初期設定の実行方法、および外部サービスの設定方法を説明します。

目次

第1章 インストールのための環境準備	4
1.1. システム要件	4
1.2. ストレージの要件とガイドライン	5
1.2.1. ストレージ要件	5
1.2.2. ストレージのガイドライン	5
1.3. サポート対象オペレーティングシステム	7
1.4. サポート対象ブラウザ	7
1.5. ポートとファイアウォールの要件	8
1.6. クライアントから SATELLITE SERVER への接続の有効化	11
1.7. ファイアウォール設定の確認	11
1.8. DNS 解決の検証	11
第2章 SATELLITE SERVER のインストール	13
2.1. オンラインネットワークからの SATELLITE SERVER のインストール	13
2.1.1. Red Hat Subscription Management への登録	13
2.1.2. Satellite サブスクリプションの確認、およびホストへのサブスクリプションの割り当て	14
2.1.3. リポジトリの設定	16
2.1.4. Satellite サーバーパッケージのインストール	16
2.2. 初期設定の実行	17
2.2.1. 時間の同期	17
2.2.2. ホストオペレーティングシステムへの SOS パッケージのインストール	17
2.2.3. インストールオプションの指定	17
2.2.3.1. 手動による初期設定	18
2.2.3.2. 応答ファイルを使用した初期設定の自動実行	19
2.2.4. カスタマーポータルでサブスクリプションの割り当ての作成	20
2.2.5. 割り当てへのサブスクリプションの追加	20
2.2.6. カスタマーポータルからのサブスクリプションマニフェストのエクスポート	20
2.2.7. Satellite Server へのサブスクリプションマニフェストのインポート	20
第3章 SATELLITE SERVER での追加設定の実行	22
3.1. SATELLITE TOOLS リポジトリのインストール	22
3.2. HTTP プロキシを使用した SATELLITE SERVER の設定	23
3.3. 全 SATELLITE HTTP 要求での HTTP プロキシの使用	24
3.4. 管理対象ホスト上での電源管理の有効化	25
3.5. SATELLITE SERVER で DNS、DHCP、および TFTP の設定	25
3.6. 管理対象外ネットワークに対して DNS、DHCP、および TFTP の無効化	27
3.7. SATELLITE SERVER で送信メールの設定	27
3.8. カスタムサーバー証明書を使用した SATELLITE SERVER の設定	29
3.8.1. Satellite Server 向けの SSL 証明書の取得	30
3.8.2. Satellite Server の SSL 証明書の検証	32
3.8.3. カスタム証明書パラメーターを使用した Satellite インストーラーの実行	33
3.8.4. Satellite Server に接続されたすべてのホストへの新しい証明書のインストール	34
3.9. SATELLITE での外部データベースの使用	34
3.9.1. 外部データベースとして MongoDB を使用する際の注意点	34
3.9.2. 外部データベースとして PostgreSQL を使用する際の注意点	35
3.9.3. 概要	36
3.9.4. MongoDB のインストール	36
3.9.5. PostgreSQL のインストール	37
3.10. MONGOD へのアクセスの制限	39
第4章 外部サービスの設定	41
4.1. 外部 DNS を使用した SATELLITE の設定	41

4.2. DNS サービスの開始と起動	43
4.3. SATELLITE SERVER での外部 DHCP の設定	43
4.4. 外部 TFTP での SATELLITE SERVER の設定	47
4.4.1. ファイアウォールでの TFTP への外部アクセスの設定	48
4.5. SATELLITE または CAPSULE での外部 IDM DNS の設定	48
4.5.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定	49
4.5.2. TSIG 認証を使用した動的 DNS 更新の設定	52
4.5.3. 内部 DNS サービス使用への復元	55
第5章 SATELLITE SERVER のアンインストール	57
第6章 AMAZON WEB SERVICES 上での RED HAT SATELLITE の実行	58
6.1. ユースケースにおける留意点	58
6.1.1. 機能するユースケース	58
6.1.2. 機能しないユースケース	59
6.2. デプロイメントシナリオ	59
6.3. 前提条件	62
6.3.1. Amazon Web Service の前提条件	62
6.3.2. Red Hat Cloud の前提条件	63
6.3.3. Red Hat Satellite 固有の前提条件	63
6.3.4. Red Hat Satellite のインストール準備	64
6.4. AWS での SATELLITE SERVER のインストール	64
6.5. AWS での CAPSULE のインストール	64
6.6. ブートストラップスクリプトを使ったホストの SATELLITE への登録	64
付録A 大規模デプロイメントに関する考慮事項	65
付録B RED HAT SATELLITE へのカスタム設定の適用	69
B.1. PUPPET 実行で上書きされた手動変更の復元	69

第1章 インストールのための環境準備

1.1. システム要件

ネットワーク接続されたベースシステムには、以下の要件が適用されます。

- 64 ビットアーキテクチャー
- Red Hat Enterprise Linux 7 Server の最新バージョン
- 最低 4 コア 2.0 GHz CPU
- Satellite Server が機能するには、最低 20 GB のメモリーが必要です。また、最低 4 GB のスワップ領域が推奨されます。最低値よりも少ないメモリーで実行している Satellite は正常に動作しないことがあります。
- 一意なホスト名 (小文字、数字、ドット (.)、ハイフン (-) を使用できます)
- 現在の Red Hat Satellite サブスクリプション
- 管理ユーザー (root) アクセス
- システム umask 0022
- 完全修飾ドメイン名を使用した完全な正引きおよび逆引きの DNS 解決

Satellite Server または Capsule Server をインストールする前に、環境がインストール要件を満たしていることを確認する必要があります。

Satellite Server は、新規にプロビジョニングされたシステムにインストールする必要があり、このシステムで Satellite Server の実行以外の機能は提供されません。



注記

Red Hat Satellite Server と Capsule Server のバージョンは同じでなければなりません。たとえば、Satellite 6.2 Server は 6.5 Capsule Server を実行できず、Satellite 6.5 Server は 6.2 Capsule Server を実行できません。Satellite Server と Capsule Server のバージョンが一致しないと、警告なしで Capsule Server が失敗します。



注記

自己登録の Satellites はサポートされません。

コンテンツホストが多数ある場合には、「[大規模なデプロイメントに関する考慮事項](#)」を参照して、お使いの環境が正しく設定されていることを確認してください。

Capsule Server のスケーリングの詳細については、「[Capsule Server のスケーラビリティに関する考慮事項](#)」を参照してください。

認定ハイパーバイザー

Red Hat Satellite は、Red Hat Enterprise Linux の実行をサポートするハイパーバイザーで稼働する物理システムおよび仮想マシン両方を完全にサポートしています。認定ハイパーバイザーに関する詳細は、「[Red Hat Enterprise Linux の実行が認定されているハイパーバイザー](#)」を参照してください。

FIPS モード

FIPS モードで稼働する Red Hat Enterprise Linux システムに、Satellite Server および Capsule Server をインストールできます。詳細は、『Red Hat Enterprise Linux セキュリティーガイド』の「[FIPS モードの有効化](#)」を参照してください。

1.2. ストレージの要件とガイドライン

このセクションでは、最小ストレージ要件を紹介し、Satellite Server と Capsule Server のインストールのストレージに関するガイドラインについて説明します。

1.2.1. ストレージ要件

以下の表には、特定のディレクトリーのストレージ要件が詳細に記載されています。これらの値は、想定ユースケースシナリオに基づいており、個別の環境に応じて異なることがあります。

ランタイムサイズは Red Hat Enterprise Linux 5、6、および 7 のリポジトリーと同期して測定されています。

表1.1 オンラインでの Satellite Server インストールのストレージ要件

ディレクトリー	インストールサイズ	ランタイムサイズ
/var/cache/pulp/	1M バイト	20 GB
/var/lib/pulp/	1 MB	500 GB
/var/lib/mongodb/	3.5 GB	50 GB
/var/lib/qpidd/	25 MB	適用外
/var/log/	10 MB	250 MB
/var/lib/pgsql/	100 MB	10 GB
/var/spool/squid/	0 MB	10 GB
/usr	3 GB	適用外
/opt	500 MB	適用外
/opt/puppetlabs	180 MB	適用外

1.2.2. ストレージのガイドライン

Satellite Server をインストールして効率性を向上する場合には、以下のガイドラインを考慮してください。

- Satellite Server データと Capsule Server データの多くは **/var** ディレクトリーに格納されるため、LVM ストレージに **/var** をマウントして、システムがスケーリングできるようにしてください。

- `/var/lib/pulp/` ディレクトリーと `/var/lib/mongodb/` ディレクトリーには、ハードディスクドライブ (HDD) ではなく、帯域幅が高く、レーテンシーの低いストレージおよび SSD (ソリッドステートドライブ) を使用するよう to してください。Red Hat Satellite には I/O を大量に使用する操作が多数あるため、高レーテンシーで低帯域幅のストレージを使用すると、パフォーマンス低下の問題が発生します。インストールに、毎秒 60 - 80 メガバイトのスピードがあることを確認してください。**fiio** ツールを使用すると、このデータが取得できます。**fiio** ツールの詳細な使用法は、Red Hat ナレッジベースのソリューション「[Impact of Disk Speed on Satellite 6 Operations](#)」を参照してください。
- `/var/lib/qpidd/` ディレクトリーでは、**goferd** サービスが管理するコンテンツホスト 1 つに対して使用される容量は 2 MB を少し超えます。たとえば、コンテンツホストの数が 10,000 個の場合、`/var/lib/qpidd/` に 20 GB のディスク容量が必要になります。
- `/var/cache/pulp/` と `/var/lib/pulp/` ディレクトリーに同じボリュームを使用することで、同期後に `/var/cache/pulp/` から `/var/lib/pulp/` にコンテンツを移動する時間を短縮できます。

ファイルシステムのガイドライン

- XFS ファイルシステムは、**ext4** では存在する inode の制限がないため、Red Hat Satellite 6 では XFS ファイルシステムを使用してください。Satellite は多くのシンボリックリンクを使用するため、**ext4** とデフォルトの数の inode を使用する場合は、システムで inode が足りなくなる可能性が高くなります。
- MongoDB は従来の I/O を使用してデータファイルにアクセスしないので、MongoDB では NFS を使用しないでください。また、NFS でデータファイルとジャーナルファイルの両方がホストされている場合にはパフォーマンスの問題が発生します。NFS を使用する必要がある場合は、`/etc/fstab` ファイルで **bg**、**noatime**、および **noatime** のオプションを使用してボリュームをマウントします。
- 入出力レーテンシーが高すぎるため、GFS2 ファイルシステムは使用しないでください。

NFS マウントを使用する場合の SELinux の考慮事項

NFS 共有を使用して `/var/lib/pulp` ディレクトリーをマウントすると、SELinux は同期プロセスをブロックします。これを避けるには、以下の行を `/etc/fstab` に追加して、ファイルシステムテーブル内の `/var/lib/pulp` ディレクトリーの SELinux コンテキストを指定します。

```
nfs.example.com:/nfsshare /var/lib/pulp/content nfs
context="system_u:object_r:httpd_sys_rw_content_t:s0" 1 2
```

NFS 共有が既にマウントされている場合は、上記の方法を使用して再マウントし、以下のコマンドを入力します。

```
# chcon -R system_u:object_r:httpd_sys_rw_content_t:s0 /var/lib/pulp
```

重複パッケージ

異なるリポジトリーで重複するパッケージは、ディスク上に一度しか格納されないため、重複するパッケージを含む追加リポジトリーに必要なストレージが少なく済みます。ストレージの多くは、`/var/lib/mongodb/` ディレクトリーおよび `/var/lib/pulp/` ディレクトリーに使用されます。これらのエンドポイントは手動で設定できません。ストレージの問題を回避するために、ストレージが `/var` ファイルシステムで利用可能であることを確認してください。

一時的なストレージ

`/var/cache/pulp/` ディレクトリーは、同期中にコンテンツを一時的に保管するために使用されます。

RPM 形式のコンテンツの場合、このディレクトリーには保管されるファイルは最大 5 RPM になります。各ファイルは、同期後に `/var/lib/pulp/` ディレクトリーに移動します。デフォルトでは、同時に最大 8 個の RPM コンテンツ同期タスクを実行でき、それぞれに対して最大 1GB のメタデータが使用されます。

ソフトウェアコレクション

ソフトウェアコレクションは、`/opt/rh/` ディレクトリーと `/opt/theforeman/` ディレクトリーにインストールされます。

`/opt` ディレクトリーへのインストールには、root ユーザーによる書き込みパーミッションおよび実行パーミッションが必要です。

シンボリックリンク

`/var/lib/pulp/` および `/var/lib/mongodb/` にはシンボリックリンクは使用できません。

同期された RHEL ISO

RHEL コンテンツの ISO を Satellite に同期する予定の場合には、Red Hat Enterprise Linux の全マイナーバージョンも同期することに注意してください。これに対応するため、Satellite に適切なストレージを設定するようにプランニングする必要があります。

1.3. サポート対象オペレーティングシステム

オペレーティングシステムは、ディスク、ローカル ISO イメージ、キックスタート、または Red Hat がサポートする他の任意の方法でインストールできます。Red Hat Satellite Server と Red Hat Satellite Capsule Server は、Satellite 6.5 のインストール時に入手可能な Red Hat Enterprise Linux 7 Server の最新バージョンでのみサポートされています。EUS または z-stream を含む Red Hat Enterprise Linux の以前のバージョンはサポートされません。

Red Hat Satellite Server および Red Hat Satellite Capsule Server には、**@Base** パッケージグループを含む Red Hat Enterprise Linux インストールが必要です。他のパッケージセットの変更や、サーバーの運用に直接必要でないサードパーティーの構成やソフトウェアは含めないようにしてください。機能強化や Red Hat 以外のセキュリティーソフトウェアもこの制限に含まれます。インフラストラクチャーにこのようなソフトウェアが必要な場合は、Satellite Server が完全に機能することを最初に確認し、その後でシステムのバックアップを作成して、Red Hat 以外のソフトウェアを追加します。

新たにプロビジョニングされたシステムで、Satellite Server および Capsule Server をインストールし、Capsule Server は Red Hat コンテンツ配信ネットワーク (CDN) に登録しないでください。Red Hat は、Satellite 以外を実行するシステムの使用はサポートしません。

1.4. サポート対象ブラウザ

以下の Web ブラウザーは完全にサポートされます。

- Firefox バージョン 39 以降
- Chrome バージョン 28 以降

以下の Web ブラウザーは部分的にサポートされます。Satellite Web UI インターフェースは正常に機能しますが、特定のデザイン要素が期待どおりに表示されないことがあります。

- Firefox バージョン 38
- Chrome バージョン 27

- Internet Explorer バージョン 10 および 11



注記

Satellite Server の Web UI とコマンドラインインターフェースは、英語、ポルトガル語、中国語(簡体)、中国語(繁体)、韓国語、日本語、イタリア語、スペイン語、ロシア語、フランス語、ドイツ語に対応しています。

1.5. ポートとファイアウォールの要件

Satellite アーキテクチャーのコンポーネントが通信できるようにするには、Satellite をインストールするベースのオペレーティングシステムで、特定のネットワークポートを開放し、ネットワークベースのファイアウォールを無効にしておく必要があります。

以下の表では、宛先ポートおよびネットワークトラフィックの方向を紹介します。この情報を使用して、ネットワークベースのファイアウォールを設定してください。クラウドソリューションによっては、ネットワークベースのファイアウォールと同様にマシンが分離されるので、特にマシン間の通信ができるように設定する必要があります。アプリケーションベースのファイアウォールを使用する場合には、アプリケーションベースのファイアウォールで、テーブルに記載のアプリケーションすべてを許可して、ファイアウォールに既知の状態にするようにしてください。可能であれば、アプリケーションのチェックを無効にして、プロトコルをベースにポートの通信を開放できるようにしてください。

統合 Capsule

Satellite Server には Capsule が統合されており、Satellite Server に直接接続されたホストは、以下の表のコンテキストでは Satellite のクライアントになります。これには、Capsule Server が実行されているベースシステムが含まれます。

Capsule のクライアント

Satellite と統合された Capsule ではなく、Capsule のクライアントであるホストには、Satellite Server へのアクセスが必要ありません。Satellite トポロジーの詳細は『Red Hat Satellite 6 のプランニング』の「[Capsule ネットワーク](#)」を参照してください。

使用している設定に応じて、必要なポートは変わることがあります。

表1.2 Red Hat CDN 通信に対する Satellite のポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	サブスクリプション管理サービス (access.redhat.com) と Red Hat CDN (cdn.redhat.com) への接続。

Satellite がネットワークから切断されている場合を除き、Satellite Server には Red Hat CDN へのアクセスが必要になります。Red Hat CDN (cdn.redhat.com) で使用されている IP アドレスの一覧は、Red Hat カスタマーポータルナレッジベース記事「[Red Hat が公開している CIDR の一覧](#)」を参照してください。

表1.3 Satellite へのブラウザーベースユーザーインターフェース向けポート

ポート	プロトコル	サービス	用途
-----	-------	------	----

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Satellite へのブラウザベース UI アクセス
80	TCP	HTTP	Satellite に Web UI でアクセスするための HTTPS へのリダイレクション (オプション)

表1.4 Satellite に通信するクライアント向けポート

ポート	プロトコル	サービス	用途
80	TCP	HTTP	Anaconda、yum、Katello 証明書およびテンプレートの取得向け、iPXE ファームウェアのダウンロード向け
443	TCP	HTTPS	サブスクリプション管理サービス、yum、Telemetry サービス、Katello エージェントへの接続向け
5647	TCP	amqp	Satellite の Qpid ディスパッチルーターと通信する Katello エージェント
8000	TCP	HTTPS	キックスタートテンプレートをホストにダウンロードする Anaconda、iPXE ファームウェアのダウンロード向け
8140	TCP	HTTPS	マスター接続に対する Puppet エージェント
9090	TCP	HTTPS	プロビジョニング時の検出イメージや、リモート実行 (Rex) 設定の SSH キーをコピーするための Satellite Server との通信で使用するために、統合 Capsule で Smart Proxy に SCAP レポートを送信
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続
53	TCP および UDP	DNS	Satellite の統合 Capsule の DNS サービスへのクライアント DNS クエリー (オプション)

ポート	プロトコル	サービス	用途
67	UDP	DHCP	Satellite の統合 Capsule ブロードキャストと、Satellite 統合 Capsule からプロビジョニングするクライアントに対する DHCP ブロードキャストを行うクライアント (オプション)
69	UDP	TFTP	プロビジョニングのために Satellite の統合 Capsule から PXE ブートイメージファイルをダウンロードするクライアント (オプション)

Satellite Server に直接接続された管理対象ホストは、統合された Capsule のクライアントとなるため、このコンテキストではクライアントになります。これには、Capsule Server が稼働しているベースシステムが含まれます。

表1.5 Capsule に通信する Satellite 向けポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Capsule の Pulp サーバーへの接続
9090	TCP	HTTPS	Capsule のプロキシへの接続
80	TCP	HTTP	bootdisk のダウンロード (オプション)

表1.6 オプションのネットワークポート

ポート	プロトコル	サービス	用途
22	TCP	SSH	Remote Execution (Rex) および Ansible 向けの Satellite および Capsule からの通信
443	TCP	HTTPS	vCenter のコンピュートリソースに対する Satellite からの通信
5000	TCP	HTTP	OpenStack のコンピュートリソースまたは実行中のコンテナに対する Satellite からの通信
22, 16514	TCP	SSH, SSL/TLS	libvirt のコンピュートリソースに対する Satellite からの通信
389, 636	TCP	LDAP, LDAPS	LDAP およびセキュアな LDAP 認証ソースに対する Satellite からの通信

ポート	プロトコル	サービス	用途
5900~5930	TCP	SSL/TLS	ハイパーバイザー向け Web UI の NoVNC コンソールに対する Satellite からの通信

1.6. クライアントから SATELLITE SERVER への接続の有効化

Satellite Server の内部 Capsule のクライアントである Capsule とコンテンツホストは、Satellite のホストベースのファイアウォールとすべてのネットワークベースのファイアウォールを介したアクセスを必要とします。

本セクションでは、Satellite をインストールする Red Hat Enterprise Linux 7 システム上のホストベースのファイアウォールの設定と、クライアントからの受信接続を有効にし、これらの設定をシステムの再起動後にも保持する方法について説明します。使用するポートの詳細は「[ポートとファイアウォールの要件](#)」を参照してください。

ファイアウォールの設定

1. クライアントから Satellite の通信用のポートを開放するには、Satellite をインストールするベースシステムで以下のコマンドを入力します。

```
# firewall-cmd \
--add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="9090/tcp"
```

2. 変更を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

1.7. ファイアウォール設定の確認

`firewall-cmd` コマンドを使用して、ファイアウォール設定の変更を確認できます。

ファイアウォール設定の確認

```
# firewall-cmd --list-all
```

詳細情報は、『Red Hat Enterprise Linux 7 セキュリティーガイド』の「[firewalld の概要](#)」を参照してください。

1.8. DNS 解決の検証

完全修飾ドメイン名を使用して完全な正引きおよび逆引き DNS 解決を検証すると、Satellite のインストール中の問題を回避できます。

ホスト名とローカルホストが正しく解決されることを確認します。

```
# ping -c1 localhost
# ping -c1 `hostname -f` # my_system.domain.com
```

名前解決に成功すると、以下のような出力が表示されます。

```
# ping -c1 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.043 ms

--- localhost ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.043/0.043/0.043/0.000 ms

# ping -c1 `hostname -f`
PING hostname.gateway (XX.XX.XX.XX) 56(84) bytes of data.
64 bytes from hostname.gateway (XX.XX.XX.XX): icmp_seq=1 ttl=64 time=0.019 ms

--- localhost.gateway ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.019/0.019/0.019/0.000 ms
```

静的および一時的なホスト名との不一致を避けるには、次のコマンドを入力して、システム上のすべてのホスト名を設定します。

```
# hostnamectl set-hostname name
```

詳細は、『Red Hat Enterprise Linux 7 ネットワークガイド』の「[hostnamectl を使ったホスト名の設定](#)」を参照してください。



警告

Satellite 6 の運用には名前解決が非常に重要です。Satellite が完全修飾ドメイン名を適切に解決できないと、多くのオプションが失敗します。これらのオプションには、コンテンツ管理、サブスクリプション管理、およびプロビジョニングがあります。

第2章 SATELLITE SERVER のインストール

本章では、Red Hat Satellite Server のインストール、初期設定、マニフェストの作成およびインストール、および追加設定の実行について説明します。

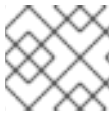
Satellite Server は、以下の 2 つのインストール方法があります。

接続インストール

Satellite Server のインストールに必要なパッケージは、Red Hat コンテンツ配信ネットワーク (CDN) から直接取得できます。CDN を使用すると、システムは常に最新のアップデートを受信できます。

非接続インストール

外部のコンピューターを使用してパッケージの ISO イメージをダウンロードして、それを Satellite Server のインストール先のシステムにコピーする必要があります。非接続環境が必要な場合にのみ、ISO イメージを使用してください。ISO イメージには最新のアップデートが含まれていない場合があります。



注記

Satellite Server をそれ自体に登録することはできません。

2.1. オンラインネットワークからの SATELLITE SERVER のインストール

オンラインネットワークから Satellite Server をインストールする場合には、Red Hat コンテンツ配信ネットワーク から直接パッケージとアップデートを取得できます。

Satellite 6 インストールスクリプトは Puppet をベースとしていることに留意してください。つまり、インストールスクリプトを複数回実行すると、手動での設定変更が上書きされることがあります。この問題を回避し、適用する変更を特定するには、インストールスクリプトを実行時に **--noop** 引数を使用します。この引数により、実際の変更は行われません。潜在的な変更は **/var/log/katello-installer.log** に書き込まれます。

ファイルは常にバックアップされるため、不要な変更は復元することができます。たとえば、katello-installer ログには、Filebucket に関する以下のようなエントリーが示されます。

```
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed /etc/dhcp/dhcpd.conf to puppet with sum
622d9820b8e764ab124367c68f5fa3a1
```

以前のファイルは以下のように復元できます。

```
# puppet filebucket -l \
restore /etc/dhcp/dhcpd.conf 622d9820b8e764ab124367c68f5fa3a1
```

2.1.1. Red Hat Subscription Management への登録

Red Hat Subscription Management にホストを登録すると、ユーザーが利用可能なサブスクリプションにホストを登録して、サブスクリプションのコンテンツを使用できるようになります。これには、Red Hat Enterprise Linux、Red Hat Software Collection (RHSC)、Red Hat Satellite などのコンテンツが含まれます。

Red Hat コンテンツ配信ネットワークにシステムを登録します。プロンプトが表示されたら、カスタマーポータルユーザー名とパスワードを入力します。

-

```
# subscription-manager register
```

このコマンドを実行すると、以下のような出力が表示されます。

```
# subscription-manager register
Username: user_name
Password:
The system has been registered with ID: 541084ff2-44cab-4eb1-9fa1-7683431bcf9a
```

2.1.2. Satellite サブスクリプションの確認、およびホストへのサブスクリプションの割り当て

ホストの登録後に、利用可能な Satellite サブスクリプションを識別し、割り当てる必要があります。Satellite サブスクリプションは、Satellite コンテンツ、Red Hat Enterprise Linux、Red Hat Software Collections (RHSC)、および Red Hat Satellite へのアクセスを提供します。これは、必要な唯一のサブスクリプションです。各 Red Hat サブスクリプションはプール ID によって識別されます。

1. Satellite サブスクリプションを特定します。

```
# subscription-manager list --all --available --matches 'Red Hat Satellite Infrastructure Subscription'
```

サブスクリプションがシステムにアタッチされていない場合には、利用可能として分類されません。利用可能な Satellite サブスクリプションを見つけることができない場合は、Red Hat ナレッジベースソリューション [「How do I figure out which subscriptions have been consumed by clients registered under Red Hat Subscription Manager?」](#) を参照して、スクリプトを実行し、サブスクリプションが別のシステムで使用されているかどうかを確認できます。

出力は以下のようになります。

```
Subscription Name: Red Hat Satellite Infrastructure Subscription
Provides:          Red Hat Beta
                  Red Hat Satellite
                  Red Hat Satellite with Embedded Oracle
                  Red Hat Satellite 5 Managed DB
                  Red Hat Satellite Proxy
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Satellite Capsule
                  Red Hat Software Collections Beta (for RHEL Server)
                  Red Hat Enterprise Linux High Availability for x86_64
                  Red Hat Ansible Engine
                  Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                  Red Hat Enterprise Linux Server
SKU:               MCT3718
Contract:
Account:           540155
Serial:            6960416325892182336
Pool ID:           8a85f999655dc45a01658350d80d7164
Provides Management: Yes
Active:            True
Quantity Used:    1
Service Level:    Premium
Service Type:     L1-L3
Status Details:   Subscription is current
```

```
Subscription Type: Standard
Starts:            08/28/2018
Ends:              09/19/2019
System Type:      Physical
```

- Satellite ホストに割り当てることができるように、プール ID をメモします。実際に使用するプール ID は、この例で使用されているものとは異なります。
- Satellite サーバーにサブスクリプションを割り当てるには、実際のプール ID を指定して以下のコマンドを実行します。

```
# subscription-manager attach --pool=pool_id
```

出力は以下のようになります。

```
Successfully attached a subscription for: Red Hat Satellite
```

- サブスクリプションが正しく割り当てられたことを確認するには、以下のコマンドを実行します。

```
# subscription-manager list --consumed
```

この出力では、以下のような内容が表示されます。

```
+-----+
Consumed Subscriptions
+-----+
Subscription Name:  Red Hat Satellite Infrastructure Subscription
Provides:          Red Hat Beta
                   Red Hat Satellite
                   Red Hat Satellite with Embedded Oracle
                   Red Hat Satellite 5 Managed DB
                   Red Hat Satellite Proxy
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Satellite Capsule
                   Red Hat Software Collections Beta (for RHEL Server)
                   Red Hat Enterprise Linux High Availability for x86_64
                   Red Hat Ansible Engine
                   Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                   Red Hat Enterprise Linux Server
SKU:               MCT3718
Contract:
Account:           540155
Serial:            6960416325892182336
Pool ID:           8a85f999655dc45a01658350d80d7164
Provides Management: Yes
Active:            True
Quantity Used:    1
Service Level:    Premium
Service Type:     L1-L3
Status Details:   Subscription is current
Subscription Type: Standard
Starts:           08/28/2018
Ends:             09/19/2019
System Type:      Physical
```

■

2.1.3. リポジトリの設定

1. 既存のリポジトリをすべて無効にします。

```
# subscription-manager repos --disable "*"

```

2. Red Hat Satellite、Red Hat Enterprise Linux、Red Hat Software Collections の各リポジトリを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-server-rhsc-7-rpms \
--enable=rhel-7-server-satellite-6.5-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-ansible-2.6-rpms

```



注記

Red Hat Satellite を Red Hat Virtualization (RHV) でホストされる仮想マシンとしてインストールする場合は、**Red Hat Common** リポジトリを有効にして、RHV ゲストエージェントとドライバーもインストールする必要があります。詳細は『[仮想マシン管理ガイド](#)』の「[ゲストエージェントおよびドライバーのインストール](#)」を参照してください。

3. Red Hat Subscription Manager が特定のオペレーティングシステムリリースを使用しないようにします。

```
# subscription-manager release --unset

```

4. Red Hat 以外の **yum** リポジトリからのすべてのメタデータを消去します。

```
# yum clean all

```

5. リポジトリが有効になっていることを確認します。

```
# yum repolist enabled

```

2.1.4. Satellite サーバーパッケージのインストール

Satellite サーバーパッケージをインストールする前に、すべてのパッケージを更新する必要があります。インストール後に、サーバー証明書の設定、ユーザー名、パスワード、デフォルトの組織および場所の設定を含む Satellite サーバーの初期設定を実行する必要があります。

1. すべてのパッケージを更新します。

```
# yum update

```

2. インストールパッケージをインストールします。

```
# yum install satellite

```

3. 「[初期設定の実行](#)」を参照して、インストーラースクリプトを実行し、Satellite Server の初期設定を行います。

2.2. 初期設定の実行

本セクションでは、Red Hat Satellite Server インストール時のホストオペレーティングシステムの初期設定について説明します。時間の同期、`sos` パッケージのインストール、インストールオプションの指定などが含まれます。

作業を進める前に、使用している環境に適切なマニフェストまたはパッケージを確認します。マニフェストについての詳細は『[Red Hat Satellite コンテンツ管理ガイド](#)』の「[サブスクリプションの管理](#)」を参照してください。

2.2.1. 時間の同期

時刻の誤差を最小化するには、ホストオペレーティングシステムで時刻シンクロナイザーを起動し、有効にする必要があります。システムの時刻が正しくないと、証明書の検証に失敗することがあります。

NTP ベースの時刻シンクロナイザーは `chronyd` と `ntpd` の 2 種類利用できます。`chronyd` 実装は、特に、頻繁に一時停止するシステムと、ネットワークから断続的に切断されるシステムに推奨されます。`ntpd` 実装は、`chronyd` でまだサポートされていないプロトコルまたはドライバーに対するサポートが必要な場合にのみ使用してください。

`ntpd` と `chronyd` の違いについては、『[システム管理者のガイド](#)』の「[ntpd と chronyd の違い](#)」を参照してください。

chronyd を使用した時間の同期

1. `chronyd` をインストールします。

```
# yum install chrony
```

2. `chronyd` サービスを起動して、有効にします。

```
# systemctl start chronyd
# systemctl enable chronyd
```

2.2.2. ホストオペレーティングシステムへの SOS パッケージのインストール

ホストオペレーティングシステムには `sos` パッケージをインストールする必要があります。`sos` パッケージを使用すると、Red Hat Enterprise Linux システムから設定と診断情報を収集できます。また、Red Hat テクニカルサポートでサービスリクエストを開く際に必要な初期システム分析を提供することもできます。`sos` の使用の詳細は、カスタマーポータルナレッジベース「[Red Hat Enterprise Linux 4.6 以降における sosreport の役割と取得方法](#)」を参照してください。

`sos` パッケージをインストールします。

```
# yum install sos
```

2.2.3. インストールオプションの指定

Satellite Server は `satellite-installer` インストールスクリプトを使用してインストールし、初期設定の一部として自動または手動で Satellite を設定します。

以下のいずれかの設定方法を選択します。

- 自動設定: この方法は、インストールスクリプトの実行時に応答ファイルを使用して設定プロセスを自動化することで実行します。応答ファイルとは、コマンドやスクリプトによって読み込まれるパラメーター一覧が含まれているファイルです。デフォルトの Satellite 応答ファイルは、`/etc/foreman-installer/scenarios.d/satellite-answers.yaml` です。使用する応答ファイルは、`/etc/foreman-installer/scenarios.d/satellite.yaml` 設定ファイル内の `answer_file` ディレクティブで設定します。
応答ファイルを使用したインストールスクリプトによる初期設定の実行方法は「[応答ファイルを使用した初期設定の自動実行](#)」を参照してください。
- 手動設定: 1つ以上のコマンドオプションが含まれるインストールスクリプトを実行します。コマンドオプションは、対応するデフォルトの初期設定オプションを上書きし、Satellite 応答ファイルに記録されます。必要なオプションを設定するために、スクリプトは何回でも実行することができます。
コマンドラインオプションのあるインストールスクリプトによる初期設定の実行方法は「[手動による初期設定](#)」を参照してください。



注記

Satellite インストーラーの実行時に使用するオプションによっては、設定が完了するのに数分かかることがあります。管理者は、応答ファイルを見ることで、両方の方法でこれまでに使用されたオプションを確認できます。

2.2.3.1. 手動による初期設定

初期設定では、組織、場所、ユーザー名、およびパスワードが作成されます。初期設定後に、必要に応じて追加の組織と場所を作成できます。初期設定では、MongoDB および PostgreSQL データベースも同じサーバーにインストールします。デプロイメントによっては、外部のデータベースを使用する方がパフォーマンスが向上する可能性があります。

インストールプロセスの完了には、数十分かかることがあります。システムにリモートで接続する場合は、リモートシステムから切断された場合にインストールの進捗を確認できるよう、通信セッションの一時中断または再接続を許可できる `screen` などのユーティリティーの使用を検討してください。Red Hat ナレッジベースの記事「[How to use the screen command](#)」には `screen` のインストールについて記載されています。詳細は `screen` の man ページを参照してください。インストールコマンドを実行しているシェルへの接続が切断された場合は、`/var/log/foreman-installer/satellite.log` のログを参照してプロセスが正常に完了したかどうかを確認します。

Satellite Server の手動設定

`satellite-installer --scenario satellite --help` コマンドを使用して利用可能なオプションとすべてのデフォルト値を表示します。値を指定しない場合は、デフォルト値が使用されます。

`--foreman-initial-organization` オプションには、意味のある値を指定することが推奨されます。たとえば会社名を指定できます。値に一致する内部ラベルが作成されますが、このラベルは後で変更できません。値を指定しない場合は、ラベルが `Default_Organization` の `Default Organization` という名前の組織が作成されます。組織名は変更できますが、ラベルは変更できません。

デフォルトでは、インストーラーが設定するすべての設定ファイルが Puppet によって管理されます。`satellite-installer` を実行すると、Puppet が管理するファイルに手動で加えられた変更が初期値で上書きされます。Satellite Server は、デフォルトでは、サービスとして実行している Puppet エージェントを使用してインストールされます。必要に応じて、`--puppet-runmode=none` オプションを使用して、Satellite Server で Puppet エージェントを無効にできます。

DNS ファイルと DHCP ファイルを手動で管理する場合には、`--foreman-proxy-dns-managed=false`

オプションと `--foreman-proxy-dhcp-managed=false` オプションを使用して、各サービスに関連するファイルが Puppet で管理されないようにします。他のサービスにカスタム設定を適用する方法は「[付録B Red Hat Satellite へのカスタム設定の適用](#)」を参照してください。

Satellite で外部データベースを使用する場合には、`satellite-installer` ツールを実行する前に、外部データベースを設定して参照する必要があります。詳しい情報は、『[オンラインネットワークからの Satellite Server のインストール](#)』の「[Satellite での外部データベースの使用](#)」を参照してください。

```
# satellite-installer --scenario satellite \
--foreman-initial-organization "initial_organization_name" \
--foreman-initial-location "initial_location_name" \
--foreman-admin-username admin_user_name \
--foreman-admin-password admin_password
```

このスクリプトは、進捗を表示し、`/var/log/foreman-installer/satellite.log` にログを記録します。

2.2.3.2. 応答ファイルを使用した初期設定の自動実行

応答ファイルを使用すると、カスタマイズされたオプションでインストールを自動化できます。最初の応答ファイルには、部分的に情報が入力されます。応答ファイルには、`satellite-installer` スクリプトの初回実行後に、インストール向けの標準的なパラメーター値が入力されます。いつでも Satellite Server の設定は変更できます。

ネットワークの変更の場合は、可能な限り、IP アドレスの代わりに FQDN を使用する必要があります。

応答ファイルを使用した Satellite Server の自動設定

1. デフォルトの応答ファイル `/etc/foreman-installer/scenarios.d/satellite-answers.yaml` をローカルファイルシステムの場所にコピーします。

```
# cp /etc/foreman-installer/scenarios.d/satellite-answers.yaml \
/etc/foreman-installer/scenarios.d/my-answer-file.yaml
```

2. 設定可能なすべてのオプションを表示するには、`satellite-installer --scenario satellite --help` コマンドを実行します。
3. 応答ファイルのコピーを開き、ご使用の環境に適した値を編集し、ファイルを保存します。
4. `/etc/foreman-installer/scenarios.d/satellite.yaml` ファイルを開き、カスタム応答ファイルを参照する応答ファイルエントリを編集します。

```
:answer_file: /etc/foreman-installer/scenarios.d/my-answer-file.yaml
```

5. `satellite-installer` スクリプトを実行します。

```
# satellite-installer --scenario satellite
```

6. 切断環境でインストールしている場合は、ISO イメージをアンマウントします。

```
# umount /media/sat6
# umount /media/rhel7-server
```

2.2.4. カスタマーポータルでサブスクリプションの割り当ての作成

サブスクリプション情報は、Red Hat カスタマーポータルでアクセスできます。そこでサブスクリプション割り当てを使用して、Red Hat Satellite サーバーなどのオンプレミス管理アプリケーションで使用するサブスクリプションを割り当てることができます。

1. ブラウザーで <https://access.redhat.com/> を開き、Red Hat アカウントでログインします。
2. カスタマーポータルの左上にある **サブスクリプション** に移動します。
3. **サブスクリプション割り当て** に移動します。
4. **新規サブスクリプションの割り当てを作成** をクリックします。
5. **名前** フィールドに名前を入力します。
6. **タイプ** の一覧から、お使いの Satellite Server に一致するタイプとバージョンを選択します。
7. **作成** をクリックします。

2.2.5. 割り当てへのサブスクリプションの追加

以下の手順では、サブスクリプションを割り当てに追加する方法について説明します。

1. **サブスクリプション割り当て** に移動します。
2. 変更するサブスクリプションの名前を選択します。
3. **サブスクリプション** タブをクリックします。
4. **サブスクリプションの追加** をクリックします。
5. Red Hat 製品サブスクリプションの一覧が表示されます。各製品に対する**エンタイトルメントの数量**を入力します。
6. **送信** をクリックして割り当てを完了します。

割り当てにサブスクリプションを追加したら、マニフェストファイルをエクスポートします。

2.2.6. カスタマーポータルからのサブスクリプションマニフェストのエクスポート

1つ以上のサブスクリプションがあるサブスクリプション割り当てを表示し、以下のいずれかからマニフェストをエクスポートできます。

- **サブスクリプション** セクションの **詳細** タブから **マニフェストのエクスポート** ボタンをクリックします。
- **サブスクリプション** タブから **マニフェストのエクスポート** ボタンをクリックします。

マニフェストをエクスポートすると、カスタマーポータルにより、選択したサブスクリプション証明書がエンコードされ、.zip アーカイブが作成されます。作成した .zip アーカイブはサブスクリプションのマニフェストで、Satellite サーバーにアップロードできます。

2.2.7. Satellite Server へのサブスクリプションマニフェストのインポート

Red Hat Satellite 6 Web UI と CLI は、マニフェストをインポートする手段を提供します。

Web UI をご利用の場合

1. コンテキストが、使用する組織に設定されていることを確認します。
2. コンテンツ > Red Hat サブスクリプション に移動します。
3. マニフェストの管理 をクリックして、組織のマニフェストページを表示します。
4. ファイルの選択 をクリックしてサブスクリプションマニフェストを選択し、アップロード をクリックします。

CLI をご利用の場合

Red Hat Satellite 6 CLI を使用するには、マニフェストが Satellite Server 上にある必要があります。ローカルクライアントシステムで、マニフェストを Satellite Server にコピーします。

```
[user@client ~]$ scp ~/manifest_file.zip root@satellite.example.com:~/.
```

次に、以下のコマンドを使用してインポートします。

```
[root@satellite ~]# hammer subscription upload \  
--file ~/manifest_file.zip \  
--organization "organization_name"
```

数分後、CLI により、マニフェストのインポートに成功したことが報告されます。

上記の手順を完了すると、リポジトリを有効にして Red Hat コンテンツをインポートできるようになります。これは、後に続くいくつかの手順での前提条件になります。詳細は、『Red Hat Satellite コンテンツ管理ガイド』の「Red Hat コンテンツのインポート」を参照してください。

第3章 SATELLITE SERVER での追加設定の実行

3.1. SATELLITE TOOLS リポジトリのインストール

Satellite Tools リポジトリは、Satellite Server に登録されたクライアント向けの **katello-agent** パッケージと **puppet** パッケージを提供します。クライアントのリモートアップデートを許可するために、katello エージェントをインストールすることが推奨されます。Capsule Server のベースシステムは Satellite Server のクライアントであるため、katello エージェントもインストールする必要があります。

Web UI を使用した Satellite Tools リポジトリのインストール手順:

1. Satellite Web UI で、**コンテンツ > Red Hat サブスクリプション** に移動します。
2. 検索フィールドを使用して **Red Hat Satellite Tools 6.5 (for RHEL 7 Server) (RPMs)** のリポジトリ名を入力します。
3. 利用可能なリポジトリペインで、**Red Hat Satellite Tools 6.5 (for RHEL 7 Server) (RPMs)** をクリックして、リポジトリセットを展開します。
Red Hat Satellite Tools 6.5の項目が表示されていない場合は、カスタマーポータルから取得したサブスクリプションマニフェストにその項目が含まれないことが原因として考えられます。この問題を修正するには、カスタマーポータルにログインし、これらのリポジトリを追加し、サブスクリプションマニフェストをダウンロードして、Satellite にインポートします。
4. **x86_64** エントリーでは、**有効化** アイコンをクリックして、リポジトリを有効にします。

ホストで実行している Red Hat Enterprise Linux の各サポート対象メジャーバージョンに対して Satellite Tools リポジトリを有効にします。Red Hat リポジトリの有効後に、このリポジトリの製品が自動的に作成されます。

CLI をご利用の場合

Satellite Tools リポジトリを追加します。

```
# hammer repository-set enable --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--basearch='x86_64' \
--name 'Red Hat Satellite Tools 6.5 (for RHEL 7 Server) (RPMs)'
```

Web UI を使用した Satellite Tools リポジトリの同期手順:

1. **コンテンツ > 同期ステータス** に移動します。
同期可能な製品リポジトリのリストが表示されます。
2. 製品コンテンツの横にある矢印をクリックして利用可能なコンテンツを表示します。
3. 同期するコンテンツを選択します。
4. **今すぐ同期** をクリックします。

CLI をご利用の場合

Satellite Tools リポジトリを同期します。

```
$ hammer repository synchronize --organization "initial_organization_name" \
```

```
--product 'Red Hat Enterprise Linux Server' \
--name 'Red Hat Satellite Tools 6.5 for RHEL 7 Server RPMs x86_64' \
--async
```

3.2. HTTP プロキシを使用した SATELLITE SERVER の設定

ネットワークで HTTP プロキシを使用している場合は、Red Hat コンテンツ配信ネットワーク (CDN) または別のコンテンツソースへの要求送信に HTTP プロキシを使用するように Satellite Server を設定できます。ネットワークの変更が原因で接続が失われるのを回避するために、可能な限り IP の代わりに FQDN を使用します。

以下の手順では、Satellite のコンテンツダウンロード専用のプロキシを設定します。

認証メソッド

Basic 認証のみがサポートされます。ユーザー名とパスワード情報を `--katello-proxy-url` オプションに追加するか、`--katello-proxy-username` および `--katello-proxy-password` オプションを使用します。

HTTP プロキシを使用した Satellite Server の設定

1. `http_proxy`、`https_proxy`、および `no_proxy` の変数が設定されていないことを確認します。

```
# unset http_proxy
# unset https_proxy
# unset no_proxy
```

2. HTTP プロキシオプションを使用して `satellite-installer` を実行します。

```
# satellite-installer --scenario satellite \
--katello-proxy-url=http://myproxy.example.com \
--katello-proxy-port=8080 \
--katello-proxy-username=proxy_username \
--katello-proxy-password='proxy_password'
```

3. Satellite Server が Red Hat CDN に接続し、リポジトリを同期できることを確認します。

- a. ネットワークゲートウェイと HTTP プロキシで、以下のホスト名に対して TCP を有効にします。

ホスト名	ポート	プロトコル
subscription.rhsm.redhat.com	443	HTTPS
cdn.redhat.com	443	HTTPS
*.akamaiedge.net	443	HTTPS
cert-api.access.redhat.com (Red Hat Insights を使用している場合)	443	HTTPS
api.access.redhat.com (Red Hat Insights を使用している場合)	443	HTTPS

Satellite Server は、SSL で安全に Red Hat CDN と通信します。SSL インターセプトプロキシを使用すると、この通信が妨害されます。これらのホストは、プロキシでホワイトリスト化されている必要があります。

Red Hat CDN (cdn.redhat.com) で使用されている IP アドレスの一覧は、Red Hat カスタマーポータルナレッジベース記事「[Red Hat が公開している CIDR の一覧](#)」を参照してください。

- b. Satellite Server の `/etc/rhsm/rhsm.conf` ファイルで、以下の詳細を記入します。

```
# an http proxy server to use (enter server FQDN)
proxy_hostname = myproxy.example.com

# port for http proxy server
proxy_port = 8080

# user name for authenticating to an http proxy, if needed
proxy_user =

# password for basic http proxy auth, if needed
proxy_password =
```

カスタムポートに関する SELinux の考慮事項

SELinux を使用すると、Red Hat Satellite 6 と Red Hat Subscription Manager のアクセスが、特定ポートに限定されます。HTTP キャッシュの TCP ポートは、8080、8118、8123、および 10001~10010 になります。SELinux のタイプ `http_cache_port_t` がないポートを使用する場合は、以下のステップを行います。

1. 以下のコマンドを実行して、SELinux で HTTP キャッシュに許可されているポートを確認します。

```
# semanage port -l | grep http_cache
http_cache_port_t tcp 8080, 8118, 8123, 10001-10010
[output truncated]
```

2. 以下のコマンドを実行して、SELinux が HTTP キャッシュにポート (たとえば、8088) を許可するよう設定します。

```
# semanage port -a -t http_cache_port_t -p tcp 8088
```

3.3. 全 SATELLITE HTTP 要求での HTTP プロキシの使用

Satellite Server は、HTTP および HTTPS をブロックするファイアウォールの内側に設定する必要がある場合に、コンピュートリソースなどの外部システムとの通信に使用するプロキシを設定してください。

プロビジョニングにコンピュートリソースを使用し、コンピュートリソースと、異なる HTTP プロキシを併用する場合には、コンピュートリソースに設定したプロキシではなく、Satellite 通信すべてに設定したプロキシが優先されます。

Satellite からの HTTP 送信接続すべてに、HTTP プロキシを設定するには、以下の手順を実行します。

1. Satellite Web UI で、**管理** > **設定** に移動します。

2. **HTTP(S) プロキシ** の行で、隣接する **Value** 列を選択して、プロキシの URL を入力します。
3. チェックのアイコンをクリックして変更を保存します。

プロキシ化された要求を受信しないようにホストを除外する手順

Satellite HTTP または HTTPS 要求に HTTP プロキシを使用する場合は、プロキシ経由で通信しないように、特定のホストを除外できます。

プロキシ経由の通信を使用しないように、ホスト1つまたは複数を除外するには、以下の手順を実行します。

1. Satellite Web UI で、**管理 > 設定** に移動します。
2. **ホスト以外の HTTP(S) プロキシ** の行で、隣接する **Value** の列を選択して、プロキシ要求から除外する、1つまたは複数のホストの名前を入力します。
3. チェックのアイコンをクリックして変更を保存します。

3.4. 管理対象ホスト上での電源管理の有効化

Satellite Server でベースボード管理コントローラー (BMC) を有効にすると、IPMI (Intelligent Platform Management Interface) または類似したプロトコルを使用して、管理対象ホストで電源管理コマンドを使用できます。

BMC サービスを使用すると、さまざまな電源管理タスクを実行できます。この機能の基礎となるプロトコルは IPMI です (BMC 機能とも呼ばれます)。IPMI は、ホストの CPU から独立して実行する専用プロセッサに接続された管理対象ハードウェア上で、特別なネットワークインターフェースを使用します。多くのインスタンスで、BMC 機能はシャーシ管理の一部として、シャーシベースのシステムに組み込まれます (シャーシの専用モジュール)。

BMC サービスの詳細は『**ホストの管理**』の「[追加のネットワークインターフェースの設定](#)」を参照してください。

作業開始前の準備

- すべての管理対象ホストに **BMC** タイプのネットワークインターフェースが搭載されている必要があります。Satellite はこの NIC を使用して適切な認証情報をホストに渡します。

管理対象ホスト上での電源管理の有効化

1. オプションを使用してインストーラーを実行し、BMC を有効にします。

```
# satellite-installer --foreman-proxy-bmc "true" \  
--foreman-proxy-bmc-default-provider "freeipmi"
```

3.5. SATELLITE SERVER で DNS、DHCP、および TFTP の設定

Satellite Server では、DNS、DHCP、および TFTP を設定できます。

外部サービスを設定する場合は、詳細について「[Satellite Server での外部サービスの設定](#)」を参照してください。

これらのサービスを手動で管理するために Satellite でサービスを無効にする場合は、詳細について「[管理対象外ネットワークに対して DNS、DHCP、および TFTP を無効化](#)」を参照してください。

設定可能な全オプションを表示するには、**satellite-installer --scenario satellite --help** コマンドを実行します。

作業開始前の準備

- ネットワーク管理者に連絡して正しい設定が行われていることを確認します。
- 以下の情報を用意する必要があります。
 - DHCP IP アドレス範囲
 - DHCP ゲートウェイ IP アドレス
 - DHCP ネームサーバー IP アドレス
 - DNS 情報
 - TFTP サーバー名
- ネットワークの変更の場合は、可能な限り、IP アドレスの代わりに FQDN を使用します。

Satellite Server での DNS、DHCP、および TFTP の設定

1. 使用している環境に適切なオプションを使用して **satellite-installer** を実行します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns true \
--foreman-proxy-dns-managed true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-forwarders 172.17.13.1 \
--foreman-proxy-dns-reverse 13.17.172.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "172.17.13.100 172.17.13.150" \
--foreman-proxy-dhcp-gateway 172.17.13.1 \
--foreman-proxy-dhcp-nameservers 172.17.13.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername $(hostname)
```

DHCP、DNS および TFTP サービスの情報は、『[プロビジョニングガイド](#)』の「[ネットワークサービスの設定](#)」セクションを参照してください。

このスクリプトは、進捗を表示し、`/var/log/foreman-installer/satellite.log` にログを記録します。`/etc/foreman-installer/scenarios.d/satellite-answers.yaml` ファイルで、`admin_password` パラメーターなど、使用する設定を表示できます。



注記

設定を変更するには、**satellite-installer** を再び実行する必要があります。スクリプトは複数回実行でき、すべての設定ファイルが変更された値で更新されます。

3.6. 管理対象外ネットワークに対して DNS、DHCP、および TFTP の無効化

TFTP、DHCP および DNS サービスを手動で管理する場合には、Satellite がオペレーティングシステム上でこれらのサービスを管理しないようにし、オーケストレーションを無効にして、DHCP および DNS バリデーションエラーを回避する必要があります。ただし、Satellite ではオペレーティングシステムのバックエンドサービスは削除されません。

手順

オペレーティングシステム上で Satellite による DHCP、DNS、および TFTP サービスを管理を回避し、オーケストレーションを無効にするには、次の手順を実行します。

1. Satellite Server で以下のコマンドを入力します。

```
# satellite-installer --foreman-proxy-dhcp false \  
--foreman-proxy-dns false \  
--foreman-proxy-tftp false
```

2. Satellite Web UI で、インフラストラクチャー > Capsule に移動し、サブネットを選択します。
3. Capsules タブで、DHCP Capsule、TFTP Capsule、および 逆引き DNS Capsule を選択します。
4. インフラストラクチャー > ドメイン に移動し、ドメインを選択します。
5. DNS Capsule フィールドの内容を消去します。
6. オプション: サードパーティーが提供する DHCP サービスを使用する場合は、以下のオプションを渡すように DHCP サーバーを設定します。

```
Option 66: IP_address_of_Satellite_or_Capsule  
Option 67: /pxelinux.0
```

DHCP オプションの詳細は [「RFC 2132」](#) を参照してください。



注記

Satellite 6 は、Capsule が該当するサブネットとドメインに設定されていない場合にオーケストレーションを実行しません。Capsule の関連付けを有効または無効にした場合に、想定のリコードと設定ファイルが存在しないと、既存のホストのオーケストレーションコマンドが失敗することがあります。オーケストレーションを有効にするために Capsule を関連付ける場合は、今後、ホストの削除に失敗しないように、既存の Satellite ホストに対して必要な DHCP レコード、DNS レコード、TFTP ファイルが所定の場所にあることを確認します。

3.7. SATELLITE SERVER で送信メールの設定

Satellite Server からメールメッセージを送信するには、SMTP サーバーまたは **sendmail** コマンドのいずれかを使用できます。

前提条件

前回のリリースからアップグレードしている場合は、設定ファイル `/usr/share/foreman/config/email.yaml` の名前を変更するか削除して、`httpd` サービスを再起動してください。例を示します。

```
# mv /usr/share/foreman/config/email.yaml \
/usr/share/foreman/config/email.yaml-backup
# systemctl restart httpd
```

Satellite Server で送信メールの設定

1. Satellite web UI で、**管理** → **設定** に移動します。
2. **Email** タブをクリックして、希望する配信方法に一致する設定オプションを設定します。変更は即座に反映されます。
 - a. 以下の例は、SMTP サーバーを使用する場合の設定オプションの例を示しています。

表3.1 配信方法に SMTP サーバーを使用する例

名前	値例
配信方法	SMTP
SMTP アドレス	smtp.example.com
SMTP 認証	ログイン
SMTP HELO/EHLO ドメイン	example.com
SMTP パスワード	パスワード
SMTP ポート	25
SMTP ユーザー名	satellite@example.com

SMTP ユーザー名 と **SMTP パスワード** では、SMTP サーバーのログイン認証情報を指定します。

- a. 以下の例では、`gmail.com` が SMTP サーバーとして使用されています。

表3.2 gmail.com を SMTP サーバーとして使用する例

名前	値例
配信方法	SMTP
SMTP アドレス	smtp.gmail.com
SMTP 認証	plain

名前	値例
SMTP HELO/EHLO ドメイン	smtp.gmail.com
SMTP enable StartTLS auto	あり
SMTP パスワード	パスワード
SMTP ポート	587
SMTP ユーザー名	user@gmail.com

- c. 以下の例では、**sendmail** コマンドが配信方法として使用されています。

表3.3 配信方法に sendmail を使用する例

名前	値例
配信方法	Sendmail
Sendmail の引数	-i -t -G

Sendmail の引数 では、**sendmail** コマンドに渡すオプションを指定します。デフォルト値は、**-i -t** です。詳細は、**sendmail 1** の man ページを参照してください。

3. TLS 認証を使用する SMTP サーバーで電子メールを送信する場合は、以下のいずれかの手順を実行してください。

- SMTP サーバーの CA 証明書を信頼済みとしてマークします。このようにマークするには、Satellite Server で以下のコマンドを実行します。

```
# cp mailca.crt /etc/pki/ca-trust/source/anchors/
# update-ca-trust enable
# update-ca-trust
```

ここで、**mailca.crt** は SMTP サーバーの CA 証明書です。

- 別の方法では、web UI の **SMTP enable StartTLS auto** オプションを **No** に設定します。
4. **Test email** をクリックしてユーザーのメールアドレスにテストメッセージを送信し、設定が機能していることを確認します。メッセージの送信に失敗する場合は、web UI でエラーが表示されます。詳細については、**/var/log/foreman/production.log** のログを確認してください。



注記

個々のユーザーまたはユーザーグループに対する電子メール通知の設定は、『Red Hat Satellite の管理』の「[電子メール通知の設定](#)」を参照してください。

3.8. カスタムサーバー証明書を使用した SATELLITE SERVER の設定

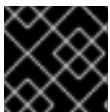
SSL 証明書は、情報を保護し、通信を安全にするために使用されます。Red Hat Satellite 6 は自己署名 SSL 証明書を作成し、Satellite Server、外部の Capsule Server、およびすべてのホスト間で暗号化された通信を有効します。必要に応じて、デフォルト証明書をカスタム証明書に置き換えることができます。これらの自己署名証明書を使用する代わりに、外部の信頼できる企業である認証局が発行したカスタム SSL 証明書をインストールすることもできます。たとえば、会社のセキュリティーポリシーで、認証局から SSL 証明書を取得することが規定されている場合があります。証明書を取得するには、「[Satellite Server 向けの SSL 証明書の取得](#)」にあるように証明書署名要求 (CSR) を作成して認証局に送信します。すると、署名済み SSL 証明書が送られてきます。

Satellite サーバーでカスタム証明書を使用するには、これらの手順を完了します。

1. [「Satellite Server 向けの SSL 証明書の取得](#)」
2. [「Satellite Server の SSL 証明書の検証](#)」
3. [「カスタム証明書パラメーターを使用した Satellite インストーラーの実行](#)」
4. [「Satellite Server に接続されたすべてのホストへの新しい証明書のインストール](#)」
5. Satellite Server に外部の Capsule Server を登録した場合には、『[CAPSULE SERVER のインストール](#)』の「[カスタムサーバー証明書を使用した Capsule Server の設定](#)」に移動して、Capsule Server がカスタムの証明書を使用するように設定します。

3.8.1. Satellite Server 向けの SSL 証明書の取得

Satellite Server 向けのカスタム SSL 証明書がすでにある場合は、この手順を省略します。



重要

SSL 証明書には、PEM エンコードを使用してください。

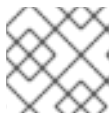
手順

Satellite Server のカスタム SSL 証明書を取得するには、次の手順を実行します。

1. **root** ユーザーのみがアクセスできる、全ソース証明書ファイルの保存用のディレクトリーを作成します (例: `/root/sat_cert`)。

```
# mkdir /root/sat_cert
```

2. 証明書署名要求 (CSR) を署名する秘密鍵を作成します。



注記

Satellite Server 向けの秘密鍵がすでにある場合は、この手順を省略します。

```
# openssl genrsa -out /root/sat_cert/satellite_cert_key.pem 4096
```

3. 証明書署名要求 (CSR) 用に `/root/sat_cert/openssl.cnf` 設定ファイルを作成し、以下の内容を追加します。[`req_distinguished_name`] セクションで、組織の情報を入力します。



注記

証明書の Common Name (CN) およびは Subject Alternative Name (SAN) の DNS.1 は、証明書を使用するサーバーの完全修飾ドメイン名 (FQDN) と同じでなければなりません。要求しているのが Satellite Server の証明書の場合には、これは Satellite Server の FQDN で、Capsule サーバーの証明書の場合には、Capsule サーバーの FQDN です。

サーバーの FQDN を確認するには、該当するサーバーでコマンド **hostname -f** を実行します。

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

[ req_distinguished_name ]
C = Country Name (2 letter code)
ST = State or Province Name (full name)
L = Locality Name (eg, city)
O = Organization Name (eg, company)
OU = The division of your organization handling the certificate
CN = satellite.example.com

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

[ usr_cert ]
basicConstraints=CA:FALSE
nsCertType = client, server, email
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
nsComment = "OpenSSL Generated Certificate"
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

[ alt_names ]
DNS.1 = satellite.example.com
```

4. 証明書署名要求 (CSR) を作成します。

```
# openssl req -new \
-key /root/sat_cert/satellite_cert_key.pem \
-out /root/sat_cert/satellite_cert_csr.pem \
-config /root/sat_cert/openssl.cnf
```

5. 証明局に証明書署名要求を送信します。同じ証明局が Satellite Server と Capsule Server の証明書に署名する必要があります。
要求を送信する場合は、証明書の有効期限を指定してください。証明書要求を送信する方法は異なるため、推奨の方法について認証局にお問い合わせください。要求への応答で、認証局バンドルと署名済み証明書を別々のファイルで受け取るようになります。

3.8.2. Satellite Server の SSL 証明書の検証

以下の例のように、必要なパラメーターを使用して **katello-certs-check** コマンドを入力します。これにより、カスタム証明書に必要な入力ファイルが検証され、これらを Satellite サーバー、すべての Capsule サーバー、および Satellite で管理されているホストにインストールするために必要なコマンドが出力されます。

1. カスタム SSL 証明書入力ファイルを検証します。ファイルに一致するようファイル名を変更します。

```
# katello-certs-check \
  -c /root/sat_cert/satellite_cert.pem \ 1
  -k /root/sat_cert/satellite_cert_key.pem \ 2
  -b /root/sat_cert/ca_cert_bundle.pem 3
```

- 1 認証局により署名された Satellite Server 向けの証明書ファイル
- 2 証明書を署名するために使用する Satellite Server の秘密鍵
- 3 認証局バンドル

katello-certs-check の出力例

```
Checking server certificate encoding:
[OK]

Checking expiration of certificate:
[OK]

Checking expiration of CA bundle:
[OK]

Checking if server certificate has CA:TRUE flag
[OK]

Checking to see if the private key matches the certificate:
[OK]

Checking CA bundle against the certificate file:
[OK]

Checking Subject Alt Name on certificate
[WARNING]

The /root/sat_cert/satellite_cert.pem does not contain a Subject Alt Name.
Checking Key Usage extension on certificate for Key Encipherment
[OK]

Validation succeeded

To use them inside a NEW $CAPSULE, run this command:

capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" \
```

```
--certs-tar "~/${CAPSULE}-certs.tar" \
--server-cert "/root/sat_cert/satellite_cert.pem" \
--server-key "/root/sat_cert/satellite_cert_key.pem" \
--server-ca-cert "/root/sat_cert/bundle2.crt" \
```

To use them inside an EXISTING \$CAPSULE, run this command INSTEAD:

```
capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" \
--certs-tar "~/${CAPSULE}-certs.tar" \
--server-cert "/root/sat_cert/satellite_cert.pem" \
--server-key "/root/sat_cert/satellite_cert_key.pem" \
--server-ca-cert "/root/sat_cert/bundle2.crt" \
--certs-update-server
```

3.8.3. カスタム証明書パラメーターを使用した Satellite インストーラーの実行

この時点で SSL 証明書が作成され、Red Hat Satellite 6 で使用できることが確認されました。次の手順は、カスタム SSL 証明書を Satellite Server とそのすべてのホストにインストールすることです。

この手順は、Satellite Server がすでにインストールされているかどうかに応じて、少し異なります。Satellite Server がすでにインストールされている場合は、既存の証明書を証明書アーカイブの証明書で更新する必要があります。

このセクションのコマンドは、「[Satellite Server の SSL 証明書の検証](#)」で説明されたように **katello-certs-check** コマンドの出力を使用します。**katello-certs-check** の出力は、ターミナルにコピーアンドペーストできます。

1. インストールの状況に応じて、**satellite-installer** コマンドを実行します。
 - a. Satellite がすでにインストールされている場合は、Satellite サーバーで以下のコマンドを実行します。

```
# satellite-installer --scenario satellite \
--certs-server-cert /root/sat_cert/satellite_cert.pem \
--certs-server-key /root/sat_cert/satellite_cert_key.pem \
--certs-server-ca-cert /root/sat_cert/ca_cert_bundle.pem \
--certs-update-server --certs-update-server-ca
```

このコマンドの重要なパラメーターは **--certs-update-server** と **--certs-update-server-ca** です。これにより、サーバーの SSL 証明書と認証局を更新するよう指定されます。すべてのインストーラーのパラメーターの簡単な説明は、**satellite-installer --scenario satellite -help** コマンドを実行します。



注記

satellite-installer コマンドにおけるファイルはすべて、相対パス名ではなく完全パス名を使用します。インストーラーにより、すべてのファイルのパスと名前が記録されます。インストーラーを異なるディレクトリーから再び実行する場合は、元のファイルを見つけることができないため、失敗します。

- b. Satellite をまだインストールしていない場合は、Satellite Server で以下のコマンドを実行します。

```
# satellite-installer --scenario satellite \
--certs-server-cert /root/sat_cert/satellite_cert.pem \
```

```
--certs-server-key /root/sat_cert/satellite_cert_key.pem \  
--certs-server-ca-cert /root/sat_cert/ca_cert_bundle.pem
```



注記

satellite-installer コマンドにおけるファイルはすべて、相対パス名ではなく完全パス名を使用します。インストーラーにより、すべてのファイルのパスと名前が記録されます。インストーラーを異なるディレクトリーから再び実行する場合は、元のファイルを見つけることができないため、失敗します。

2. 証明書をホストにインストールする前に証明書が Satellite サーバーに正常にインストールされていることを確認します。Satellite サーバーへのネットワークアクセスがあるコンピュータで、Web ブラウザーを起動し、URL <https://satellite.example.com> に移動して、証明書の詳細を参照します。

3.8.4. Satellite Server に接続されたすべてのホストへの新しい証明書のインストール

カスタム SSL 証明書が Satellite サーバーにインストールされたので、Satellite サーバーに登録されている各ホストにもインストールする必要があります。すべての該当するホストで以下のコマンドを実行します。

1. ホスト上で現行の **katello-ca-consumer** パッケージを削除します。

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

2. ホストにカスタム SSL 証明書をインストールします。

```
# yum localinstall http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

3.9. SATELLITE での外部データベースの使用

Red Hat Satellite のインストールプロセスの一部として、**satellite-installer** コマンドは MongoDB および PostgreSQL のデータベースを Satellite と同じサーバー上にインストールします。Satellite のデプロイメントによっては、外部データベースがサーバーの負荷を軽減する場合があります。ただし、Satellite Server のパフォーマンスに影響を与える可能性のある要素は多数あります。外部データベースに移動すると、固有の問題に対応できない可能性があります。

外部データベースに MongoDB と PostgreSQL のどちらのデータベースが使用できるかについては、要件によって異なります。

Red Hat では、外部データベースのメンテナンスのサポートやそのためのツールは提供していません。これにはバックアップ、アップグレード、データベースのチューニングが含まれます。外部データベースを使用しているお客様は、外部データベースをサポート、メンテナンスする独自のデータベース管理者が必要になります。

お使いの Satellite デプロイメントで外部データベースを必要とする場合は、以下の情報を使用して、Satellite から外部データベースにポイントするように設定します。

3.9.1. 外部データベースとして MongoDB を使用する際の注意点

Pulp は MongoDB データベースを使用します。MongoDB を外部データベースとして使用する場合は、以下の情報を参照してお使いの Satellite 設定にこのオプションが適しているかどうかを判定してください。

外部 MongoDB の利点

- Satellite 上の空きメモリーと空き CPU が増えます。
- Satellite 操作にマイナスの影響をもたらすことなく MongoDB サーバーのシステムを調整する柔軟性が得られます。

外部 MongoDB のマイナス点

- デプロイメントの複雑性が増し、問題解決がより困難になります。
- 外部 MongoDB サーバーだと、パッチおよびメンテナンス対象に新たなシステムが加わることとなります。
- Satellite または Mongo データベースサーバーのいずれかにハードウェアまたはストレージ障害が発生すると、Satellite が機能しなくなります。
- Satellite と外部データベースサーバーの間でレイテンシーが発生すると、パフォーマンスに影響が出ます。

お使いの Mongo データベースが遅いと感じられる場合は、Red Hat サポートチームと協力して問題解決に当たることができます。Satellite 6 での設定問題や既存のパフォーマンス問題については、外部データベースサーバーに移行したとしても解決が期待できないものもあります。Red Hat サポートチームは既知の問題を調査するほか、Satellite エンジニアリングチームとも協力して根本原因を見つけ出します。

3.9.2. 外部データベースとして PostgreSQL を使用する際の注意点

Foreman、Katello、および Candlepin は PostgreSQL データベースを使用します。PostgreSQL を外部データベースとして使用する場合は、以下の情報を参照してお使いの Satellite 設定にこのオプションが適しているかどうかを判定してください。

外部 PostgreSQL の利点

- Satellite 上の空きメモリーと空き CPU が増えます。
- PostgreSQL データベースで **shared_buffers** を高い値に設定しても、Satellite 上の他のサービスの妨げるリスクがありません。
- Satellite 操作にマイナスの影響をもたらすことなく PostgreSQL サーバーのシステムを調整する柔軟性が得られます。

外部 PostgreSQL のマイナス点

- デプロイメントの複雑性が増し、問題解決がより困難になります。
- 外部 PostgreSQL サーバーだと、パッチおよびメンテナンス対象に新たなシステムが加わることとなります。
- Satellite または PostgreSQL データベースサーバーのいずれかにハードウェアまたはストレージ障害が発生すると、Satellite が機能しなくなります。
- Satellite と外部データベースサーバーの間でレイテンシーが発生すると、パフォーマンスに影響が出ます。

お使いの Satellite 上の PostgreSQL データベースがパフォーマンスを低下させていることが疑われる場

合は、[Satellite 6: How to enable postgres query logging to detect slow running queries](#) を参照して遅いクエリがあるかどうか判定します。1秒以上かかるクエリがある場合は、通常、大規模インストールのパフォーマンスが原因であることが多く、外部データベースに移行しても問題解決が期待できません。遅いクエリがある場合は、Red Hat サポートチームまでお問い合わせください。

3.9.3. 概要

Satellite 用にリモートデータベースを作成して使用するには、以下の手順を実行します。

1. 「[ストレージの要件とガイドライン](#)」を使用して、外部データベースのストレージ要件をプランニングします。
2. PostgreSQL で Foreman および Candlepin 用のデータベースを準備し、Foreman と Candlepin の所有権を持つ専用ユーザーを作成します。
3. `pulp_database` を所有している `pulp` ユーザーで MongoDB を準備します。
4. 最初の手順に従い、Satellite をインストールし、データベースが Satellite からアクセスできることを確認します。
5. `satellite-installer` のパラメーターを、新規データベースを参照するように編集し、`satellite-installer` を実行します。

Red Hat Enterprise Linux Server 7 をデータベースインストール用に準備する

「[ストレージの要件とガイドライン](#)」のストレージ要件を満たす、最新の Red Hat Enterprise Linux Server 7 で、新たにシステムをプロビジョニングする必要があります。

Red Hat Software Collections および Red Hat Enterprise Linux のサブスクリプションでは、外部データベースと Satellite を併用する場合に、正しいサービスレベルアグリーメントが提供されません。外部データベースに使用するベースシステムに、Satellite サブスクリプションもアタッチする必要があります。

1. 「[Satellite サブスクリプションの特定およびホストへのアタッチ](#)」にある説明を参照して、Satellite サブスクリプションをサーバーにアタッチします。
2. MongoDB および PostgreSQL サーバーを Red Hat Enterprise Linux Server 7 にインストールするには、すべてのリポジトリを無効にし、以下のリポジトリのみを有効にする必要があります。

```
# subscription-manager repos --disable '*'
# subscription-manager repos --enable=rhel-server-rhsc7-7-rpms \
--enable=rhel-7-server-rpms
```

3.9.4. MongoDB のインストール

インストール可能な MongoDB は、内部データベースのインストール中に `satellite-installer` ツールでインストールされたものと同じバージョンの MongoDB のみになります。MongoDB はサポート対象のバージョンであれば、Red Hat Software Collections (RHSC) リポジトリからまたは外部ソースからインストールすることが可能です。Satellite は MongoDB バージョン 3.4 をサポートしています。

1. MongoDB をインストールするには、以下のコマンドを入力します。

```
# yum install rh-mongodb34 rh-mongodb34-syspaths
```

2. `rh-mongodb34` サービスを起動して有効にします。


```
# systemctl start rh-mongodb34-mongod
# systemctl enable rh-mongodb34-mongod
```

3. **pulp_database** データベース用に、MongoDB に Pulp ユーザーを作成します。

```
# mongo pulp_database \
--eval "db.createUser({user:'pulp',pwd:'pulp_password',roles:[{role:'dbOwner',
db:'pulp_database'},{ role: 'readWrite', db: 'pulp_database'}]})"
```

4. **/etc/opt/rh/rh-mongodb34/mongod.conf** ファイルでバインド IP を指定します。

```
bindIp: your_mongodb_server_bind_IP:::1
```

5. **/etc/opt/rh/rh-mongodb34/mongod.conf** ファイルを編集して **security** セクションの認証を有効にします。

```
security:
  authorization: enabled
```

6. **rh-mongodb34-mongod** サービスを再起動します。

```
# systemctl restart rh-mongodb34-mongod
```

7. MongoDB にポート 27017 を開きます。

```
# firewall-cmd --add-port=27017/tcp
# firewall-cmd --runtime-to-permanent
```

8. Satellite Server から、データベースにアクセスできることをテストします。接続に成功した場合には、コマンドは **1** を返します。

```
# scl enable rh-mongodb34 " mongo --host mongo.example.com \
-u pulp -p pulp_password --port 27017 --eval 'ping:1' pulp_database"
```

3.9.5. PostgreSQL のインストール

インストール可能な PostgreSQL は、内部データベースのインストール中に **satellite-installer** ツールでインストールされたものと同じバージョンの PostgreSQL のみになります。Satellite がサポートするのは、Red Hat Enterprise Linux Server 7 リポジトリから入手可能な特定バージョンの PostgreSQL のみになります。PostgreSQL はサポート対象のバージョンであれば、**rhel-7-server-rpms** リポジトリからまたは外部ソースからインストールすることが可能です。サポート対象の PostgreSQL バージョンを格納しているリポジトリについての情報は、『[Package Manifest](#)』を参照してください。

1. PostgreSQL をインストールするには、以下のコマンドを入力します。

```
# yum install postgresql-server
```

2. PostgreSQL サービスを初期化して起動し、有効にするには、以下のコマンドを実行します。

```
# postgresql-setup initdb
# systemctl start postgresql
# systemctl enable postgresql
```

3. `/var/lib/pgsql/data/postgresql.conf` ファイルを編集します。

```
# vi /var/lib/pgsql/data/postgresql.conf
```

4. `#` を削除して、着信接続をリッスンするようにします。

```
listen_addresses = '*'
```

5. `/var/lib/pgsql/data/pg_hba.conf` ファイルを編集します。

```
# vi /var/lib/pgsql/data/pg_hba.conf
```

6. 以下の行を追加します。

```
host all all satellite_server_ip/24 md5
```

7. PostgreSQL サービスを再起動して、変更を適用します。

```
# systemctl restart postgresql
```

8. 外部 PostgreSQL サーバーで `postgresql` ポートを開きます。

```
# firewall-cmd --add-service=postgresql  
# firewall-cmd --runtime-to-permanent
```

9. `postgres` ユーザーに切り替え、PostgreSQL クライアントを起動します。

```
$ su - postgres -c psql
```

10. Satellite と Candlepin 用にそれぞれ、データベース、および専用ロールを作成します。

```
CREATE USER "foreman" WITH PASSWORD 'Foreman_Password';  
CREATE USER "candlepin" WITH PASSWORD 'Candlepin_Password';  
CREATE DATABASE foreman OWNER foreman;  
CREATE DATABASE candlepin OWNER candlepin;
```

11. Satellite Server から、データベースにアクセスできることをテストします。接続に成功した場合には、コマンドは `1` を返します。

```
# PGPASSWORD='Foreman_Password' psql -h postgres.example.com -p 5432 -U  
foreman -d foreman -c "SELECT 1 as ping"  
# PGPASSWORD='Candlepin_Password' psql -h postgres.example.com -p 5432 -U  
candlepin -d candlepin -c "SELECT 1 as ping"
```

12. Satellite 用にリモートデータベースをインストールして設定するには以下のコマンドを入力します。

```
satellite-installer --scenario satellite \  
  --foreman-db-host postgres.example.com \  
  --foreman-db-password Foreman_Password \  
  --foreman-db-database foreman \  
  --katello-candlepin-db-host postgres.example.com \  
  --katello-candlepin-db-password Candlepin_Password
```

```
--katello-candlepin-db-name candlepin \
--katello-candlepin-db-password Candlepin_Password \
--katello-candlepin-manage-db false \
--katello-pulp-db-username pulp \
--katello-pulp-db-password pulp_password \
--katello-pulp-db-seeds mongo.example.com:27017 \
--katello-pulp-db-name pulp_database
```

データベースのステータスをクエリーします。たとえば、以下のコマンドに **--only** と **postgresql** または **rh-mongodb34-mongod** を追加して実行します。

PostgreSQL の場合は、以下のコマンドを実行します。

```
# foreman-maintain service status --only postgresql
```

MongoDB の場合は、以下のコマンドを実行します。

```
# foreman-maintain service status --only rh-mongodb34-mongod
```

3.10. MONGODB へのアクセスの制限

データ損失の危険を減らすために、MongoDB データベースデーモン **mongod** へのアクセスは **apache** ユーザーと **root** ユーザーにだけ許可する必要があります。

Satellite Server と Capsule Server で **mongod** へのアクセスを制限するには、以下のコマンドを使用します。

1. ファイヤーウォールを設定します。

```
# firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p \
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p \
tcp -m tcp --dport 28017 -j DROP
```

2. 変更を永続化します。

```
█ # firewall-cmd --runtime-to-permanent
```

第4章 外部サービスの設定

一部の環境には DNS、DHCP、および TFTP サービスがすでに存在するため、これらのサービスを提供するために Satellite Server を使用する必要はありません。DNS、DHCP、または TFTP を提供するために外部サーバーを使用する場合は、Satellite Server で使用するよう設定できます。

これらのサービスを手動で管理するために Satellite でサービスを無効にする場合は、詳細について「[管理対象外ネットワークに対して DNS、DHCP、および TFTP を無効化](#)」を参照してください。

4.1. 外部 DNS を使用した SATELLITE の設定

DNS サービスを提供するために Satellite が外部サーバーを使用するよう設定できます。

1. Red Hat Enterprise Linux Server をデプロイし、ISC DNS サービスをインストールします。

```
# yum install bind bind-utils
```

2. ドメインの設定ファイルを作成します。

以下の例では、ドメイン **virtual.lan** を1つのサブネット 192.168.38.0/24 として設定し、**capsule** という名前のセキュリティーキーを設定して、フォワーダーを Google のパブリック DNS アドレス (8.8.8.8 および 8.8.4.4) に設定します。192.168.38.2 は DNS サーバーの IP アドレスで、192.168.38.1 は、Satellite Server または Capsule Server の IP アドレスになります。

```
# cat /etc/named.conf
include "/etc/rndc.key";

controls {
    inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "capsule"; };
    inet 192.168.38.2 port 953 allow { 192.168.38.1; 192.168.38.2; } keys { "capsule"; };
};

options {
    directory "/var/named";
    forwarders { 8.8.8.8; 8.8.4.4; };
};

include "/etc/named.rfc1912.zones";

zone "38.168.192.in-addr.arpa" IN {
    type master;
    file "dynamic/38.168.192-rev";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};

zone "virtual.lan" IN {
    type master;
    file "dynamic/virtual.lan";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};
```

設定ファイルの **inet** 行は、1つの行として入力する必要があります。

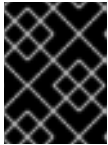
3. キーファイルを作成します。

```
# ddns-confgen -k capsule
```

このコマンドが完了するまで、しばらく時間がかかることがあります。

4. キーセクションから出力をコピーし、**/etc/rndc.key** という名前の別のファイルに貼り付けます。

```
# cat /etc/rndc.key
key "capsule" {
    algorithm hmac-sha256;
    secret "GeBbgGoLedEAAwNQPtPh3zP56MJbkwM84UJDtaUS9mw=";
};
```



重要

これは、DNS サーバー設定を変更するために使用するキーです。root ユーザーのみが読み書きできるようにする必要があります。

5. ゾーンファイルを作成します。

```
# cat /var/named/dynamic/virtual.lan
$ORIGIN .
$TTL 10800 ; 3 hours
virtual.lan IN SOA service.virtual.lan. root.virtual.lan. (
    9 ; serial
    86400 ; refresh (1 day)
    3600 ; retry (1 hour)
    604800 ; expire (1 week)
    3600 ; minimum (1 hour)
)
NS service.virtual.lan.
$ORIGIN virtual.lan.
$TTL 86400 ; 1 day
capsule A 192.168.38.1
service A 192.168.38.2
```

6. 逆引きゾーンファイルを作成します。

```
# cat /var/named/dynamic/38.168.192-rev
$ORIGIN .
$TTL 10800 ; 3 hours
38.168.192.in-addr.arpa IN SOA service.virtual.lan. root.38.168.192.in-addr.arpa. (
    4 ; serial
    86400 ; refresh (1 day)
    3600 ; retry (1 hour)
    604800 ; expire (1 week)
    3600 ; minimum (1 hour)
)
NS service.virtual.lan.
$ORIGIN 38.168.192.in-addr.arpa.
```

```
$TTL 86400 ; 1 day
1 PTR capsule.virtual.lan.
2 PTR service.virtual.lan.
```

ASCII 以外の他の文字は使用しないでください。

4.2. DNS サービスの開始と起動

1. 構文を検証します。

```
# named-checkconf -z /etc/named.conf
```

2. サーバーを起動します。

```
# systemctl restart named
```

3. 新しいホストを追加します。

以下のコマンドでは、ホストの例 192.168.38.2 を使用しています。この値は、ご使用の環境に合わせて変更してください。

```
# echo -e "server 192.168.38.2\n \
update add aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

4. DNS サービスが新しいホストを解決できることを確認します。

```
# nslookup aaa.virtual.lan 192.168.38.2
```

5. 必要な場合は、新しいエントリーを削除します。

```
# echo -e "server 192.168.38.2\n \
update delete aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

6. DNS サービスへの外部アクセスのためにファイアウォールを設定します (ポート 53 上の UDP および TCP)。

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
&& firewall-cmd --runtime-to-permanent
```

4.3. SATELLITE SERVER での外部 DHCP の設定

本セクションでは、Red Hat Satellite Server が外部 DHCP サーバーを使用する設定について説明します。

DHCP サーバーの設定、DHCP 設定およびリースファイルの共有

1. Red Hat Enterprise Linux Server をデプロイし、ISC DHCP サービスおよび BIND (Berkeley Internet Name Domain) をインストールします。

```
# yum install dhcp bind
```

2. 空のディレクトリーでセキュリティートークンを生成します。

```
# dnssec-keygen -a HMAC-MD5 -b 512 -n HOST omapi_key
```

上記のコマンドが完了するまで、しばらく時間がかかることがあります。安全性が高くない概念実証のデプロイメントでは、非ブロック型乱数ジェネレーターを使用できます。

```
# dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 512 -n HOST omapi_key
```

これにより、キーペアが現行ディレクトリーに2つのファイルで作成されます。

3. キーからシークレットハッシュをコピーします。

```
# cat Komapi_key.+*.private |grep ^Key|cut -d ' ' -f2
```

4. すべてのサブネットに対して **dhcpd** 設定ファイルを編集し、キーを追加します。例を示します。

```
# cat /etc/dhcp/dhcpd.conf
default-lease-time 604800;
max-lease-time 2592000;
log-facility local7;

subnet 192.168.38.0 netmask 255.255.255.0 {
  range 192.168.38.10 192.168.38.100;
  option routers 192.168.38.1;
  option subnet-mask 255.255.255.0;
  option domain-search "virtual.lan";
  option domain-name "virtual.lan";
  option domain-name-servers 8.8.8.8;
}

omapi-port 7911;
key omapi_key {
  algorithm HMAC-MD5;
  secret "jN5E5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

5. 2つのキーファイルを、それらを作成したディレクトリーから削除します。

6. Satellite Server で各サブネットを定義します。

競合を回避するために、リース範囲と予約範囲は別々に設定することが推奨されます。たとえば、リース範囲を 192.168.38.10 から 192.168.38.100 とし、予約範囲 (Satellite Web UI で定義済み) を 192.168.38.101 から 192.168.38.250 とします。ここでは、定義されたサブネットに DHCP Capsule を設定しないでください。

7. ファイアウォールで DHCP サーバーへの外部アクセスを設定します。

```
# firewall-cmd --add-service dhcp \
&& firewall-cmd --runtime-to-permanent
```

8. Capsule Server の foreman ユーザーの UID 番号と GID 番号を確認します。


```
# id -u foreman
993
# id -g foreman
990
```

9. ユーザー ID とグループ ID が同じユーザーとグループを、DHCP サーバーに作成します。

```
# groupadd -g 990 foreman
# useradd -u 993 -g 990 -s /sbin/nologin foreman
```

10. 設定ファイルを読み取り可能にするために、読み取りおよび実行フラグを復元します。

```
# chmod o+rx /etc/dhcp/
# chmod o+r /etc/dhcp/dhcpd.conf
# chattr +i /etc/dhcp/ /etc/dhcp/dhcpd.conf
```

11. DHCP サービスを起動します。

```
# systemctl start dhcpd
```

12. NFS を使用して DHCP 設定およびリースファイルをエクスポートします。

```
# yum install nfs-utils
# systemctl enable rpcbind nfs-server
# systemctl start rpcbind nfs-server nfs-lock nfs-idmapd
```

13. NFS を使用して、エクスポートする DHCP 設定およびリースファイルを作成します。

```
# mkdir -p /exports/var/lib/dhcpd /exports/etc/dhcp
```

14. **/etc/fstab** ファイルに以下の行を追加して、新規作成ディレクトリー用にマウントポイントを作成します。

```
/var/lib/dhcpd /exports/var/lib/dhcpd none bind,auto 0 0
/etc/dhcp /exports/etc/dhcp none bind,auto 0 0
```

15. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

16. **/etc/exports** に以下の行が存在することを確認します。

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)
/exports/etc/dhcp 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
/exports/var/lib/dhcpd 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

17. NFS サーバーをリロードします。

```
# exportfs -rva
```

- ファイアウォールで Satellite Server 向けの DHCP omapi ポート 7911 を設定します。

```
# firewall-cmd --add-port="7911/tcp" \
&& firewall-cmd --runtime-to-permanent
```

- 必要な場合は、ファイアウォールで NFS への外部アクセスを設定します。
クライアントは NFSv3 を使用して設定されます。

- firewalld** デーモンの NFS サービスを使用してファイアウォールを設定します。

```
# firewall-cmd --zone public --add-service mountd \
&& firewall-cmd --zone public --add-service rpc-bind \
&& firewall-cmd --zone public --add-service nfs \
&& firewall-cmd --runtime-to-permanent
```

Satellite Server の設定

- NFS クライアントをインストールします。

```
# yum install nfs-utils
```

- NFS 用の DHCP ディレクトリーを作成します。

```
# mkdir -p /mnt/nfs/etc/dhcp /mnt/nfs/var/lib/dhcpd
```

- ファイルの所有者を変更します。

```
# chown -R foreman-proxy /mnt/nfs
```

- NFS サーバーとの通信と RPC 通信パスを検証します。

```
# showmount -e your_DHCP_server_FQDN
# rpcinfo -p your_DHCP_server_FQDN
```

- /etc/fstab** ファイルに以下の行を追加します。

```
your_DHCP_server_FQDN:/exports/etc/dhcp /mnt/nfs/etc/dhcp nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcp_etc_t:s0" 0 0

your_DHCP_server_FQDN:/exports/var/lib/dhcpd /mnt/nfs/var/lib/dhcpd nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcpd_state_t:s0" 0 0
```

- /etc/fstab** 上のファイルシステムをマウントします。

```
# mount -a
```

- 関連するファイルを読み取ります。

```
# su foreman-proxy -s /bin/bash
bash-4.2$ cat /mnt/nfs/etc/dhcp/dhcpd.conf
bash-4.2$ cat /mnt/nfs/var/lib/dhcpd/dhcpd.leases
bash-4.2$ exit
```

8. **satellite-installer** スクリプトを実行して、以下の永続的な変更を `/etc/foreman-proxy/settings.d/dhcp.yml` ファイルに加えます。

```
# satellite-installer --foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=your_DHCP_server_FQDN
```

9. foreman-proxy サービスを再起動します。

```
# systemctl restart foreman-proxy
```

10. Satellite Server Web インターフェースにログインします。
11. インフラストラクチャー > **Capsules** に移動します。適切な Capsule Server を見つけ、**アクション** ドロップダウンリストから **更新** を選択します。この結果、DHCP 機能が現れます。
12. DHCP サービスに適切なサブネットとドメインを関連付けます。

4.4. 外部 TFTP での SATELLITE SERVER の設定

以下の手順を使用して、外部の TFTP サービスを使用して Satellite Server を設定します。

DHCP、DNS および TFTP サービスの情報は、『[プロビジョニングガイド](#)』の「[NAT を介した TFTP サービスの使用](#)」セクションを参照してください。

作業開始前の準備

- NFS が設定され、NFS への外部アクセスのためにファイアウォールが設定されている必要があります。「[Satellite Server での外部 DHCP の設定](#)」を参照してください。

Satellite Server での外部 TFTP の設定

1. TFTP サーバーをインストールし、有効にします。

```
# yum install tftp-server syslinux
```

2. **tftp.socket** ユニットの有効にし、アクティベートします。

```
# systemctl enable tftp.socket
# systemctl start tftp.socket
```

3. PXELinux 環境を設定します。

```
# mkdir -p /var/lib/tftpboot/{boot,pxelinux.cfg,grub2}
# cp /usr/share/syslinux/{pxelinux.0,menu.c32,chain.c32} \
/var/lib/tftpboot/
```

- SELinux ファイルコンテキストを復元します。

```
# restorecon -RvF /var/lib/tftpboot/
```

- NFS を使用してエクスポートする TFTP ディレクトリーを作成します。

```
# mkdir -p /exports/var/lib/tftpboot
```

- 新しく作成されたマウントポイントを `/etc/fstab` ファイルに追加します。

```
/var/lib/tftpboot /exports/var/lib/tftpboot none bind,auto 0 0
```

- `/etc/fstab` のファイルシステムをマウントします。

```
# mount -a
```

- `/etc/exports` に以下の行があることを確認します。

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)
```

```
/exports/var/lib/tftpboot 192.168.38.1(rw,async,no_root_squash,no_subtree_check,nohide)
```

最初の行は DHCP 設定に共通で、このシステムで以前の手順を完了すると作成されます。

- NFS サーバーをリロードします。

```
# exportfs -rva
```

4.4.1. ファイアウォールでの TFTP への外部アクセスの設定

- ファイアウォールを設定します (ポート 69 上の UDP)。

```
# firewall-cmd --add-port="69/udp" \
&& firewall-cmd --runtime-to-permanent
```

4.5. SATELLITE または CAPSULE での外部 IDM DNS の設定

Red Hat Satellite は、Red Hat Identity Management (IdM) サーバーを使って DNS サービスを提供するように設定できます。これには 2 つ方法があり、その両方でトランザクションキーを使用します。Red Hat Identity Management の詳細は『[Linux ドメイン ID、認証、およびポリシーガイド](#)』を参照してください。

1 つ目の方法では、[RFC3645](#) で定義された **generic security service algorithm for secret key transaction** (GSS-TSIG) 技術を使用してプロセスを自動化する IdM クライアントをインストールします。この方法では、Satellite Server か Capsule のベースシステムに IdM クライアントをインストールし、Satellite 管理者が使用するアカウントを IdM サーバーの管理者が作成する必要があります。詳細は「[GSS-TSIG 認証を使用した動的 DNS 更新の設定](#)」を参照してください。

2 つ目の方法である **secret key transaction authentication for DNS**(TSIG) では、認証に **rndc.key** を使用します。root 権限で IdM サーバーにアクセスして BIND 設定ファイルを編集する必要があります。Satellite Server に **BIND** ユーティリティーをインストールし、システム間で **rndc.key** をコピーしま

す。この技術は、[RFC2845](#) で定義されています。詳細は「[TSIG 認証を使用した動的 DNS 更新の設定](#)」を参照してください。



注記

DNS の管理には、Satellite を使用する必要はありません。Satellite のレム登録機能を使用していて、プロビジョニングされたホストが自動的に IdM に登録されている場合は、`ipa-client-install` スクリプトでクライアント用に DNS レコードが作成されます。このため、以下の手順とレム登録は、同時に使用することはできません。レム登録の詳細は『[Red Hat Satellite の管理](#)』の「[プロビジョニングされたホストの外部認証](#)」を参照してください。

IdM クライアントのインストール先

Satellite Server がホスト用に DNS レコードを追加する際には、まずどの Capsule がそのドメインの DNS を提供しているかを判断します。その後 Capsule と通信し、レコードを追加します。ホスト自体はこのプロセスに関与していません。つまり、IdM クライアントをインストールして設定する Satellite または Capsule は、IdM サーバーを使って管理するドメインに DNS サービスを提供するように現在設定されているものにすべきということになります。

4.5.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

この例では、Satellite Server の設定は以下のようになります。

ホスト名	satellite.example.com
ネットワーク	192.168.55.0/24

IdM サーバーの設定は以下のようになります。

ホスト名	idm1.example.com
ドメイン名	example.com

作業開始前の準備

1. IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されていることを確認します。詳細は『[Linux ドメイン ID、認証、およびポリシーガイド](#)』の「[ポート要件](#)」を参照してください。
2. IdM サーバーに、IdM サーバーにゾーンを作成するパーミッションのあるアカウントを作成します。
3. Satellite または外部 Capsule がドメインの DNS を管理していることを確認します。
4. Satellite または外部 Capsule が正常に機能していることを確認します。
5. 新たにインストールしたシステムの場合は、まず本ガイドにあるインストール手順を完了させます。特に、DNS と DHCP の設定は完了させてください。
6. 変更を元に戻す場合に備えて、応答ファイルのバックアップを作成します。詳細は「[インストールオプションの指定](#)」を参照してください。

IdM サーバー上で Kerberos プリンシパルの作成

1. Kerberos チケットがあることを確認します。

```
# kinit idm_user
```

ここでの **idm_user** は、IdM 管理者が作成したアカウントになります。

2. IdM サーバーに認証する際に使用する Satellite または Capsule 用の新規 Kerberos プリンシパルを作成します。

```
# ipa service-add capsule/satellite.example.com
```

IdM クライアントのインストールと設定

以下の手順は、ドメインの DNS サービスを管理している Satellite または Capsule サーバーで行います。

1. IdM クライアントパッケージをインストールします。

```
# yum install ipa-client
```

2. インストールスクリプトとそれに続くプロンプトを実行して、IdM クライアントを設定します。

```
# ipa-client-install
```

3. Kerberos チケットがあることを確認します。

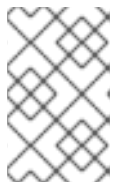
```
# kinit admin
```

4. 既存の keytab を削除します。

```
# rm /etc/foreman-proxy/dns.keytab
```

5. このシステム用に作成された keytab を取得します。

```
# ipa-getkeytab -p capsule/satellite.example.com@EXAMPLE.COM \  
-s idm1.example.com -k /etc/foreman-proxy/dns.keytab
```



注記

サービス中の元のシステムと同じホスト名を持つスタンバイシステムに keytab を追加する際には、**r** オプションを追加します。これにより、新規の認証情報が生成されることを防ぎ、元のシステムの認証情報が無効になります。

6. 以下のように、**foreman-proxy** への keytab ファイルのグループと所有者を設定します。

```
# chown foreman-proxy:foreman-proxy /etc/foreman-proxy/dns.keytab
```

7. 必要に応じて、keytab が有効か確認します。

```
# kinit -kt /etc/foreman-proxy/dns.keytab \
capsule/satellite.example.com@EXAMPLE.COM
```

IdM web UI での DNS ゾーンの設定

1. 管理するゾーンを作成して、設定します。
 - a. **Network Services (ネットワークサービス) > DNS > DNS Zones (DNS ゾーン)** に移動します。
 - b. **Add** を選択し、ゾーン名を入力します。この例では、**example.com** になります。
 - c. **Add and Edit** をクリックします。
 - d. 設定タブの **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

```
grant capsule\047satellite.example.com@EXAMPLE.COM wildcard * ANY;
```

- e. **Dynamic update** が **True** に設定されていることを確認します。
 - f. **Allow PTR sync** を有効にします。
 - g. **Save** を選択して、変更を保存します。
2. 逆引きゾーンを作成、設定します。
 - a. **Network Services (ネットワークサービス) > DNS > DNS Zones (DNS ゾーン)** に移動します。
 - b. **Add** を選択します。
 - c. **Reverse zone IP network** を選択して、CIDR 形式でネットワークアドレスを追加し、逆引き参照を有効にします。
 - d. **Add and Edit** をクリックします。
 - e. **Settings** タブの **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

```
grant capsule\047satellite.example.com@EXAMPLE.COM wildcard * ANY;
```

- f. **Dynamic update** が **True** に設定されていることを確認します。
 - g. **Save** を選択して、変更を保存します。

ドメインの DNS サービスを管理する Satellite または Capsule Server の設定

- Satellite Server のベースシステムでは、以下を実行します。

```
satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
```

```
--foreman-proxy-dns-tsig-principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-reverse="55.168.192.in-addr.arpa" \
--foreman-proxy-dns-zone=example.com \
--foreman-proxy-dns-ttl=86400
```

- Capsule Server のベースシステムでは、以下を実行します。

```
satellite-installer --scenario capsule \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
--foreman-proxy-dns-tsig-principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-reverse="55.168.192.in-addr.arpa" \
--foreman-proxy-dns-zone=example.com \
--foreman-proxy-dns-ttl=86400
```

Satellite または Capsule のプロキシーサービスを再起動します。

```
# systemctl restart foreman-proxy
```

Satellite web UI での設定更新

インストールスクリプトを実行して Capsule に変更を加えた後に、Satellite が該当する各 Capsule の設定をスキャンするようにします。

1. インフラストラクチャー > Capsules (スマートプロキシー) に移動します。
2. 更新する Capsule で、アクション ドロップダウンメニューから **更新** を選択します。
3. ドメインを設定します。
 - a. インフラストラクチャー > ドメイン に移動し、ドメイン名を選択します。
 - b. ドメイン タブで、DNS Capsule が、サブネットが接続されている Capsule に選択されていることを確認します。
4. サブネットを設定します。
 - a. インフラストラクチャー > サブネット に移動し、サブネット名を選択します。
 - b. サブネット タブで、IPAM を None に設定します。
 - c. ドメイン タブで、IdM サーバーが管理するドメインが選択されていることを確認します。
 - d. Capsules タブで、Reverse DNS Capsule、サブネットが接続されている Capsule に選択されていることを確認します。
 - e. **送信** をクリックして変更を保存します。

4.5.2. TSIG 認証を使用した動的 DNS 更新の設定

この例では、Satellite Server の設定は以下のようになります。

IP アドレス	192.168.25.1
ホスト名	satellite.example.com

IdM サーバーの設定は以下のようになります。

ホスト名	idm1.example.com
IP アドレス	192.168.25.2
ドメイン名	example.com

作業開始前の準備

1. IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されていることを確認します。詳細は『Linux ドメイン ID、認証、およびポリシーガイド』の「[ポート要件](#)」を参照してください。
2. IdM サーバーで **root** 権限を取得します。
3. Satellite または外部 Capsule がドメインの DNS を管理していることを確認します。
4. Satellite または外部 Capsule が正常に機能していることを確認します。
5. 新たにインストールしたシステムの場合は、まず本ガイドにあるインストール手順を完了させます。特に、DNS と DHCP の設定は完了させてください。
6. 変更を元に戻す場合に備えて、応答ファイルのバックアップを作成します。詳細は「[インストールオプションの指定](#)」を参照してください。

IdM サーバーの DNS ゾーンに対する外部アップデートの有効化

1. IdM サーバーで、以下の内容を `/etc/named.conf` ファイルの先頭に追加します。

```
// This was added to allow Satellite Server at 192.168.25.1 to make DNS updates.
#####
include "/etc/rndc.key";
controls {
inet 192.168.25.2 port 953 allow { 192.168.25.1; } keys { "rndc-key"; };
};
#####
```

2. **named** をリロードして、変更を有効にします。

```
# systemctl reload named
```

3. IdM Web UI で、**Network Services (ネットワークサービス) > DNS > DNS Zones (DNS ゾーン)** に移動します。ゾーンの名前を選択します。**Settings (設定)** タブで、以下の手順を実行します。
 - a. **BIND update policy (BIND アップデートポリシー)** ボックスで以下の内容を追加します。

```
grant "rndc-key" zonesub ANY;
```

- b. **Dynamic update** が **True** に設定されていることを確認します。
 - c. **Update (更新)** をクリックして変更を保存します。
4. 以下のように、IdM サーバーから Satellite のベースシステムへ **/etc/rndc.key** ファイルをコピーします。

```
# scp /etc/rndc.key root@satellite.example.com:/etc/rndc.key
```

5. 所有者、パーミッション、SELinux コンテキストが正しいことを確認します。

```
# restorecon -v /etc/rndc.key
# chown -v root:named /etc/rndc.key
# chmod -v 640 /etc/rndc.key
```

6. Satellite Server で以下のようにインストールスクリプトを実行し、外部 DNS サーバーを使用します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="192.168.25.2" \
--foreman-proxy-keyfile=/etc/rndc.key \
--foreman-proxy-dns-ttl=86400
```

IdM サーバーの DNS ゾーンに対する外部アップデートのテスト

1. テストのために **nsupdate** とともに **bind-utils** をインストールします。

```
# yum install bind-utils
```

2. Satellite Server 上の **/etc/rndc.key** ファイルのキーが IdM サーバーで使用されているものと同じであることを確認します。

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "secret-key==";
};
```

3. Satellite Server で、ホスト向けのテスト DNS エントリーを作成します (たとえば、**192.168.25.1** の IdM サーバー上に **192.168.25.20** の A レコードがあるホスト **test.example.com**)。

```
# echo -e "server 192.168.25.1\n \
update add test.example.com 3600 IN A 192.168.25.20\n \
send\n" | nsupdate -k /etc/rndc.key
```

4. Satellite Server で、DNS エントリーをテストします。

```
# nslookup test.example.com 192.168.25.1
```

```
Server: 192.168.25.1
Address: 192.168.25.1#53
```

```
Name: test.example.com
Address: 192.168.25.20
```

5. IdM Web UI でエントリーを参照するために、**Network Services (ネットワークサービス) > DNS > DNS Zones (DNS ゾーン)** に移動します。ゾーンの名前を選択し、名前でホストを検索します。
6. 正常に解決されたら、テスト DNS エントリーを削除します。

```
# echo -e "server 192.168.25.1\n \
update delete test.example.com 3600 IN A 192.168.25.20\n \
send\n" | nsupdate -k /etc/rndc.key
```

7. DNS エントリーが削除されたことを確認します。

```
# nslookup test.example.com 192.168.25.1
```

レコードが正常に削除されている場合は、上記の **nslookup** コマンドが失敗し、SERVFAIL エラーメッセージが出力されます。

4.5.3. 内部 DNS サービス使用への復元

Satellite Server と Capsule Server を DNS プロバイダーとして使用するように戻すには、以下の手順に従います。

ドメインの DNS を管理する Satellite または Capsule Server

- 外部 DNS への変更前に応答ファイルをバックアップした場合は、応答ファイルを復元して、インストールスクリプトを実行します。

```
# satellite-installer
```

- 応答ファイルのバックアップがない場合は、現行の応答ファイルでバックアップを作成し、以下にあるように Satellite および Capsules でインストールスクリプトを実行します。応答ファイルに関する情報は、「[インストールオプションの指定](#)」を参照してください。

応答ファイルを使用せずに **Satellite** または **Capsule** を DNS サーバーとして設定

```
# satellite-installer \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1" \
--foreman-proxy-dns-tsig-principal="foremanproxy/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab
```

詳しい情報は、「[Capsule Server での DNS、DHCP および TFTP の設定](#)」を参照してください。

Satellite web UI での設定更新

インストールスクリプトを実行して Capsule に変更を加えた後に、Satellite が該当する各 Capsule の設定をスキャンするようにします。

1. インフラストラクチャー > **Capsules (スマートプロキシ)** に移動します。
2. 更新する Capsule で、**アクション** ドロップダウンメニューから **更新** を選択します。
3. ドメインを設定します。
 - a. **インフラストラクチャー > ドメイン** に移動し、ドメイン名を選択します。
 - b. **ドメイン** タブで、**DNS Capsule** が、サブネットが接続されている Capsule に選択されていることを確認します。
4. サブネットを設定します。
 - a. **インフラストラクチャー > サブネット** に移動し、サブネット名を選択します。
 - b. **サブネット** タブで、**IPAM** を **DHCP** または **Internal DB** に設定します。
 - c. **ドメイン** タブで、Satellite または Capsule が管理するドメインが選択されていることを確認します。
 - d. **Capsules** タブで、**Reverse DNS Capsule**、サブネットが接続されている Capsule に選択されていることを確認します。
 - e. **送信** をクリックして変更を保存します。

第5章 SATELLITE SERVER のアンインストール

Satellite Server または Capsule Server は、不要になったらアンインストールできます。

Satellite Server をアンインストールすると、ターゲットシステムで使用されたすべてのアプリケーションが削除されます。アプリケーションまたはアプリケーションデータを Satellite Server 以外の目的で使用する場合は、削除する前にそれらの情報をバックアップする必要があります。

作業開始前の準備

katello-remove スクリプトを実行すると、2つの警告が出され、システムのすべてのパッケージと設定ファイルを削除する前に確認が求められます。



警告

このスクリプトは、以下の重要なパッケージを含む、多くのパッケージおよび設定ファイルを削除します。

- httpd (apache)
- mongoddb
- tomcat
- puppet
- Ruby
- rubygems
- すべての Katello および Foreman パッケージ

Satellite Server のアンインストール

1. Satellite Server をアンインストールします。

```
# katello-remove
```

第6章 AMAZON WEB SERVICES 上での RED HAT SATELLITE の実行

本章では、Red Hat Satellite Server および Capsules を Amazon Web Services (AWS) Elastic Compute Cloud (Amazon EC2) にインストールする準備について説明します。

AWS 上での Satellite および Capsule インストールで利用可能なアーキテクチャ設定は「[デプロイメントシナリオ](#)」を参照してください。

前提条件 セクションでは、Red Hat Satellite のインストールに際しての Red Hat と Amazon Web リソースの準備について説明しています。

サブスクリプション

すべてのサブスクリプションがパブリッククラウド環境で実行できるわけではありません。サブスクリプションを実行できるかどうかの適格性は、「[Cloud Access Page](#)」を参照してください。新たな組織を作成して、その組織に新たなマニフェストをインポートすることもできます。詳細は『[コンテンツ管理ガイド](#)』の「[組織の作成](#)」を参照してください。

6.1. ユースケースにおける留意点

Amazon Web サービスは画像のみのサービスなので、Satellite のユースケースによっては、Amazon Web サービス環境では動作しない、または追加設定が必要な場合があります。AWS で Satellite を使用する予定がある場合は、使用するユースケースのシナリオが AWS 環境で利用可能であることを確認してください。

6.1.1. 機能するユースケース

以下の Red Hat Satellite ユースケースは、AWS で機能することがわかっています。

- [サブスクリプション管理](#)
- [コンテンツ管理](#)
- [エラータ管理](#)
- [ホストの設定](#)
- [Red Hat Insights](#)
- [コンテナのプロビジョニング](#)
- [IdM 経由のレルム統合](#)
- [OpenSCAP](#)
- [リモート実行](#)

マルチホームの Satellite および Capsule

Satellite で異なるホスト名を持つ複数のインタフェースを使用する場合は、Satellite Server と Satellite Capsule Server の CA 証明書に追加設定を行う必要があります。この設定での Satellite のデプロイをご希望の場合は、Red Hat までご連絡ください。

Satellite Server または Capsule Server の内部および外部 DNS ホスト名が異り、かつ Satellite Server と Capsule Server を配置するロケーション間にサイト間 VPN 接続がない場合は、この設定を実行する必要があります。

オンデマンドのコンテンツソース

オンデマンドダウンロードポリシーを使用すると、Satellite を実行する Red Hat Enterprise Linux サーバーのストレージフットプリントを削減できます。ダウンロードポリシーを **オンデマンド** に設定すると、コンテンツホストの要求時に、Satellite Server または Capsule Server にコンテンツが同期されません。

詳細は『[Red Hat Satellite コンテンツ管理ガイド](#)』の「[Red Hat コンテンツのインポート](#)」を参照してください。

6.1.2. 機能しないユースケース

AWS では、DHCP を管理できません。このため、Satellite Server のほとんどのキックスタートと PXE プロビジョニングモデルは使用できないことになります。これには、以下のものが含まれます。

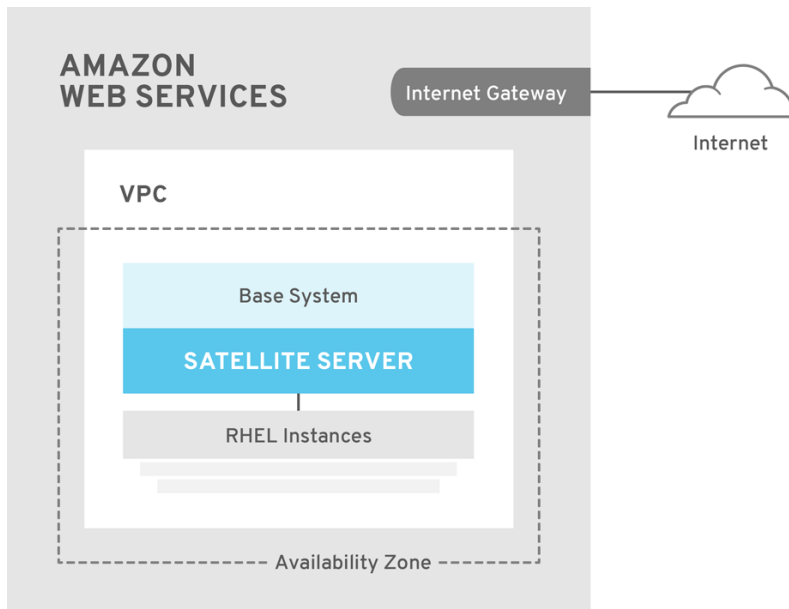
- PXE プロビジョニング
- 検出および検出ルール
- ISO プロビジョニングメソッド
 - PXE-Less Discovery (iPXE) (PXE を使用しない検出)
 - Per-host ISO
 - Generic ISO
 - Full-host ISO

6.2. デプロイメントシナリオ

Amazon Web Services では、以下の 3 つの Red Hat Satellite デプロイメントシナリオがあります。

1. 単一領域設定
2. オンプレミスと AWS 領域の接続
3. 異なる領域の接続

シナリオ 1: 単一領域設定



SATELLITE_465517_0118

Amazon Web Services における最もシンプルな Satellite Server 設定は、Satellite Server とコンテンツホストの両方が同一領域内かつ仮想プライベートクラウド (VPC) 内にあるという構成です。

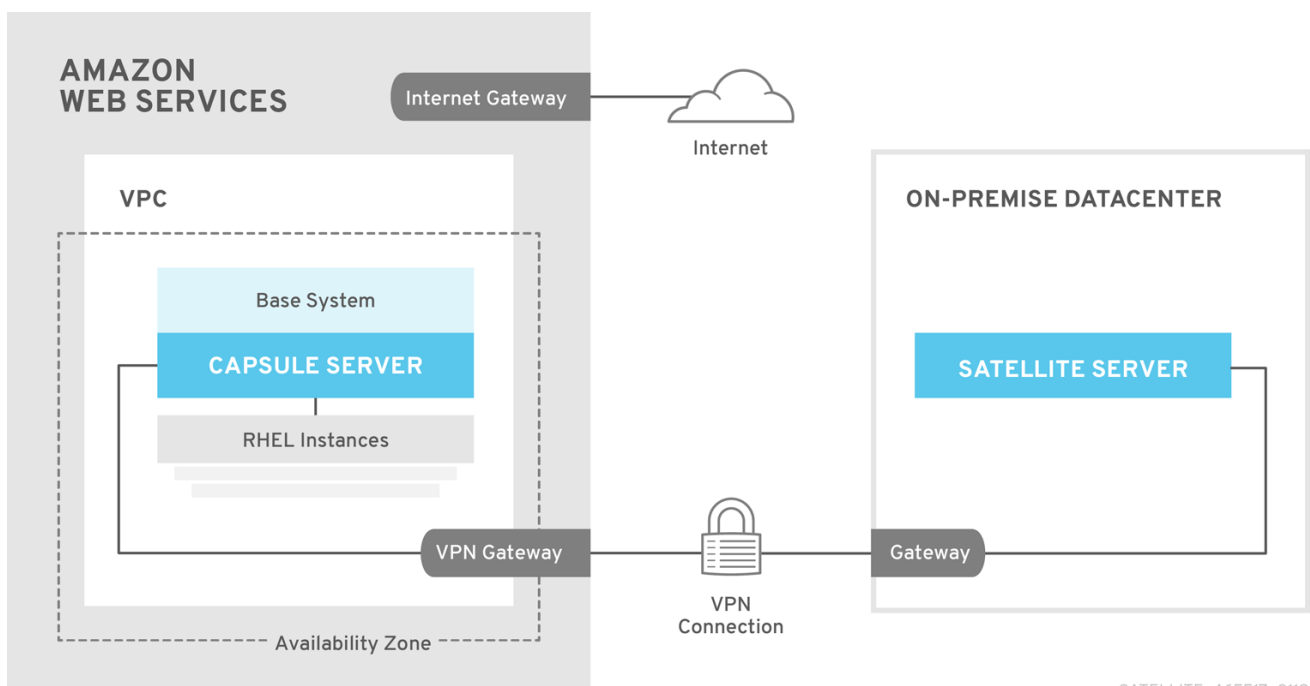
別のアベイラビリティゾーンを使用することもできます。

シナリオ 2: オンプレミスと AWS 領域の接続

この場合は、オンプレミスのロケーションと Capsule を設置する AWS 領域の間に VPN 接続を作成します。

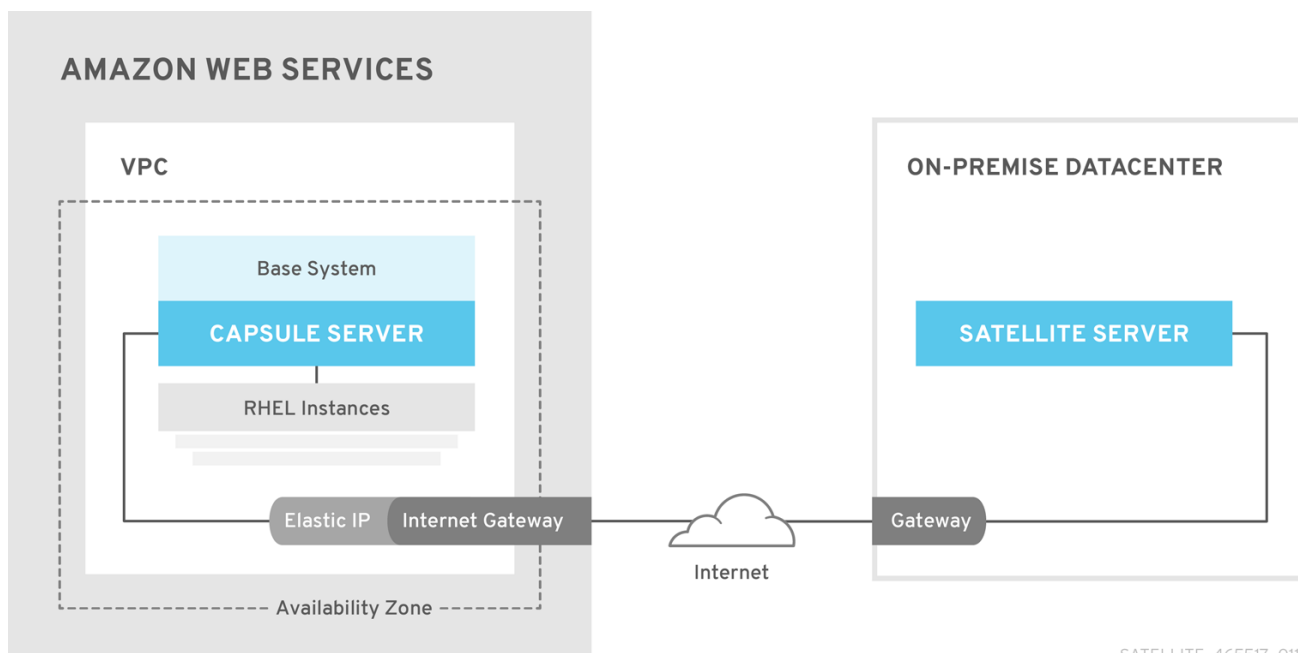
Capsule Server を実行するインスタンスを登録すれば、Satellite Server の外部のホスト名を使用することも可能です。

オプション 1: AWS 領域とオンプレミスデータセンターをつなぐサイト間 VPN 接続



SATELLITE_465517_0118

オプション 2: 外部 DNS ホスト名を使用した直接接続



シナリオ 3: 異なる領域の接続

この場合は、別の領域をつなぐサイト間 VPN 接続を作成し、Capsule Server を実行するインスタンスを Satellite Server に登録する際に、内部 DNS ホスト名を使用できるようにします。

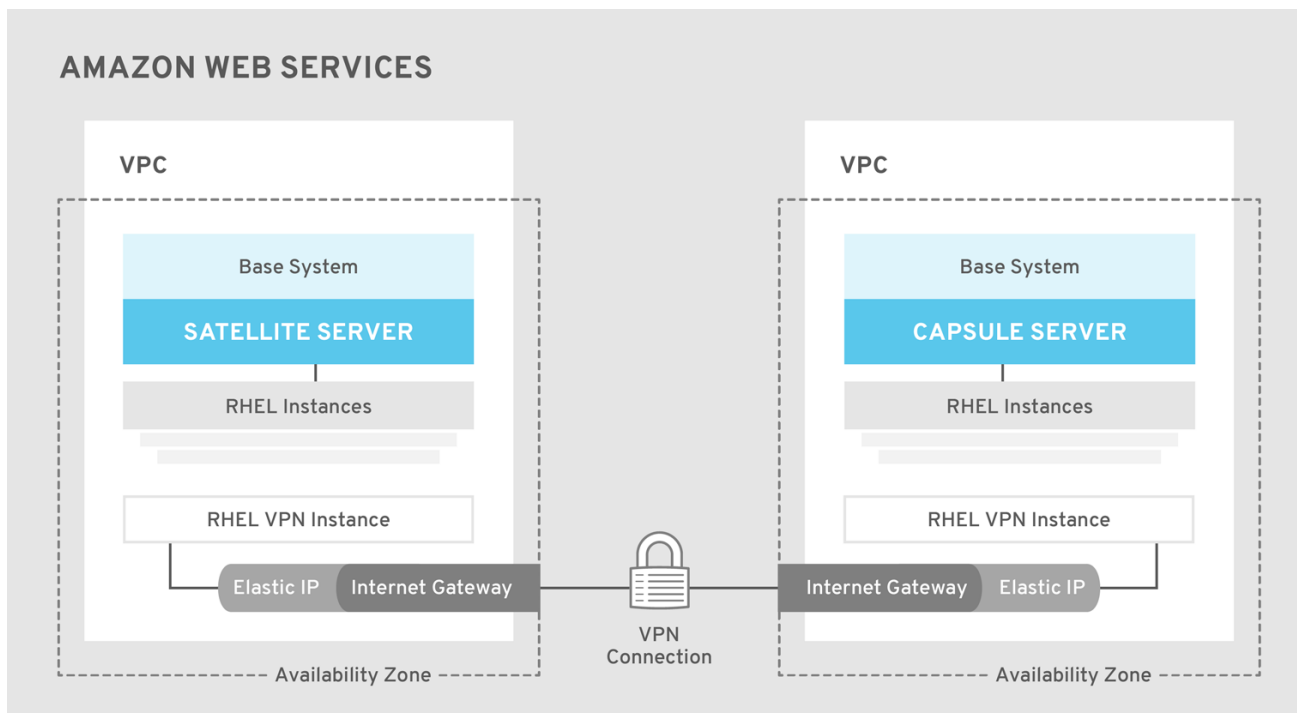
サイト間 VPN 接続を確立しない場合には、Capsule Server を実行するインスタンスを Satellite Server に登録する際に、外部 DNS ホスト名を使用します。



注記

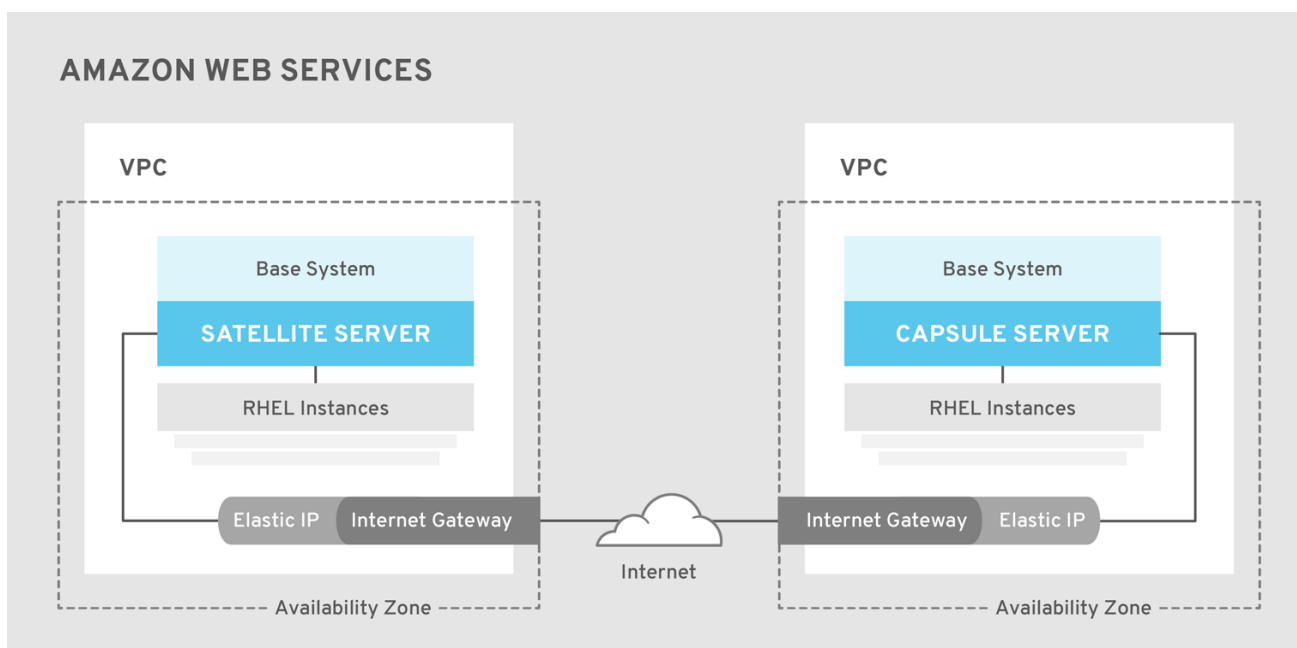
ほとんどのパブリッククラウドプロバイダーでは、領域へのデータ転送や、単一領域内でのアベイラビリティゾーン間でのデータ転送には課金されませんが、領域からインターネットへのデータ転送には課金されます。

オプション 1: AWS 領域間をつなぐサイト間 VPN 接続



SATELLITE_465517_0118

オプション 2: 外部 DNS ホスト名を使用した直接接続



SATELLITE_465517_0118

6.3. 前提条件

Red Hat Satellite および Capsule のインストールと登録前には、Amazon Web Services (AWS) でアカウントを設定し、AWS 上で Red Hat Enterprise Linux インスタンスを作成して起動する必要があります。

6.3.1. Amazon Web Service の前提条件

本ガイドでは、ユーザーが Amazon Web Services の以下の点について作業可能な知識を備えていることを前提としています。

- AWS での Red Hat Enterprise Linux イメージの作成とアクセス。
- AWS セキュリティーにおけるネットワークアクセスの編集。
- EC2 インスタンスの作成と EBS ボリュームの作成。
- インスタンスの開始。
- AWS での仮想マシンのインポートおよびエクスポート。
- AWS Direct Connect の使用。

Satellite と Capsule を AWS 環境にインストールするには、お使いの AWS 設定が「システム要件」を満たしている必要があります。

Amazon Web Services と関連用語に関する情報は、[Amazon Elastic Compute Cloud ドキュメント](#) を参照してください。

Amazon Web Services Direct Connect に関する詳細情報は、「[AWS Direct Connect とは何ですか](#)」を参照してください。

6.3.2. Red Hat Cloud の前提条件

本ガイドでは、以下のステップを完了していることが必要になります。

- Red Hat Cloud Access での登録
- 使用する Red Hat サブスクリプションの移行
- AWS インスタンスを作成し、Red Hat Enterprise Linux 仮想マシンをそのインスタンスにデプロイすること
- お使いのサブスクリプションが Red Hat Cloud に移行可能であることを確認。詳細は「[Red Hat Cloud Access プログラムの詳細](#)」を参照してください。

Red Hat Enterprise Linux の AWS でのデプロイに関する情報は、「[How to Locate Red Hat Cloud Access Gold Images on AWS EC2](#)」を参照してください。

6.3.3. Red Hat Satellite 固有の前提条件

- Amazon EC2 インスタンスのタイプが「[1章インストールのための環境準備](#)」にあるシステムおよびストレージ要件を満たしていることを確認します。ベストパフォーマンスをお求めの場合は、AWS Storage Optimized インスタンスをお使いください。
- 「[1章インストールのための環境準備](#)」を参照して、適切なストレージを把握して、AWS EBS ボリュームに割り当てます。
- ブートボリュームとは別の EBS ボリューム上に同期コンテンツを保存します。
- 同期した EBS ボリュームをオペレーティングシステム内で個別にマウントします。
- オプションで、`mongodb` ディレクトリーなどの他のデータを別個の EBS ボリュームに保存します。
- Satellite Server と Capsule Server が外部 DNS ホスト名を使用して通信するようにするには、インスタンスに関連付けられた AWS セキュリティーグループで、通信用に必要なポートを開きます。

6.3.4. Red Hat Satellite のインストール準備

お使いの AWS 環境で、以下のステップを完了させます。

1. Red Hat Enterprise Linux AMI の EC2 インスタンスを起動します。
2. 新規作成のインスタンスに接続します。

Red Hat Gold Image を使用している場合は、以下のように RHUI クライアントを削除して、**product-id.conf** の **enabled** パラメーターを **1** に設定します。

```
# yum -y remove rh-amazon-rhui-client*
# yum clean all
# cat << EOF > /etc/yum/pluginconf.d/product-id.conf
> [main]
> enabled=1
> EOF
```

6.4. AWS での SATELLITE SERVER のインストール

お使いの AWS 環境で、以下のステップを完了させます。

1. 新規インスタンスに接続します。
2. 「[Satellite Server のインストール](#)」を参照して、Satellite Server をインストールします。

6.5. AWS での CAPSULE のインストール

お使いの AWS 環境で、以下のステップを完了させます。

1. 新規インスタンスに接続します。
2. Capsule Server をインストールします。詳細は、「[Capsule Server のインストール](#)」を参照してください。

6.6. ブートストラップスクリプトを使ったホストの SATELLITE への登録

Satellite Server および Capsule Server をインストールしたら、ブートストラップスクリプトを使用して、EC2 インスタンス上のコンテンツホストを Satellite に登録する必要があります。

ブートストラップスクリプトの使用方法は、『[ユーザーガイド](#)』の「[ブートストラップスクリプトを使ったホストの Satellite への登録](#)」を参照してください。

Katello Agent をインストールします。詳細は、「[Katello Agent のインストール](#)」を参照してください。

付録A 大規模デプロイメントに関する考慮事項

Apache 向けファイル記述子の最大数の増加

登録されているコンテンツホストの数が 800 を超える場合、Apache では複数のシステムレベルの制限に到達し、新しいコンテンツホストの登録に失敗することがあります。この問題を回避するには、大量のコンテンツホストをデプロイする前に、ファイル記述子の制限を緩和する必要があります。

1. `/etc/systemd/system/httpd.service.d/limits.conf` ファイルを作成し、以下のテキストを挿入します。

```
[Service]
LimitNOFILE=65536
```

2. 変更をユニットに適用します。

```
# systemctl daemon-reload
```

3. Satellite のサービスを再起動します。

```
# foreman-maintain service restart
```

qpid 向けファイル記述子の最大数の増加

1100 を超えるコンテンツホストでエラータ更新のために `goferd` を実行すると、`qpid` でシステムレベルの制限に到達し、登録に失敗することがあります。この問題を回避するには、大量のコンテンツホストをデプロイする前に、ファイル記述子の制限を緩和する必要があります。

qpid 向けファイル記述子の最大数の増加

1. `/etc/systemd/system/qpid.service.d/limits.conf` ファイルを作成し、以下のテキストを挿入します。

```
[Service]
LimitNOFILE=65536
```

2. 変更をユニットに適用します。

```
# systemctl daemon-reload
# systemctl restart qpid.service
```

共有バッファと作業メモリの増加

`shared_buffer` と `work_mem` はそれぞれ、**256M** と **4M** に増やすことができます。

1. Red Hat Enterprise Linux 7 の場合は、`/var/lib/pgsql/data/postgresql.conf` ファイルを作成し、以下のテキストを挿入します。

```
work_mem = 4MB
shared_buffers = 256MB
```

2. `postgresql` サービスを再起動します。

```
# systemctl restart postgresql
```

同時コンテンツホスト登録の増加

システムレベルの制限への到達を回避するために、最大 250 の同時コンテンツホストを処理するようグローバルパッセンジャーキュー制限を増加できます。

グローバルパッセンジャーキューの上限を増やすには、以下の手順を実行します。

1. 最大パッセンジャープールサイズを、Satellite Server で利用可能な物理 CPU コアの 1.5 倍に調整します。
たとえば、Satellite Server にコアが 16 個ある場合、最大パッセンジャープールサイズは 24 になります。この数は例として使用しているため、実際にはご使用の環境に応じた数を使用する必要があります。
2. `/etc/httpd/conf.d/passenger.conf` ファイルを編集して以下のテキストに一致するよう `IfModule` スタンザを更新します。

```
<IfModule mod_passenger.c>
  PassengerRoot /usr/share/gems/gems/passenger-
  4.0.18/lib/phusion_passenger/locations.ini
  PassengerRuby /usr/bin/ruby
  PassengerMaxPoolSize 24
  PassengerMaxRequestQueueSize 200
  PassengerStatThrottleRate 120
</IfModule>
```

3. Foreman Passenger アプリケーション設定ファイル `/etc/httpd/conf.d/05-foreman-ssl.conf` を編集して、以下のテキストに一致するよう `PassengerAppRoot` で始まるスタンザを更新します。

```
PassengerAppRoot /usr/share/foreman
PassengerRuby /usr/bin/tfm-ruby
PassengerMinInstances 6
PassengerStartTimeout 90
PassengerMaxPreloaderIdleTime 0
PassengerMaxRequests 10000
PassengerPreStart https://example.com
```

4. Puppet Passenger アプリケーション設定ファイル `/etc/httpd/conf.d/25-puppet.conf` を編集して以下のテキストを仮想ホスト定義の最後に追加します。

```
PassengerMinInstances 6
PassengerStartTimeout 90
PassengerMaxPreloaderIdleTime 0
PassengerMaxRequests 10000
PassengerPreStart https://example.com:8140
```

5. `/var/lib/pgsql/data/postgresql.conf` ファイルで最大接続数を変更します。

```
max_connections = 500
```

6. postgresql サービスを再起動します。

```
# systemctl restart postgresql
```

qdrouterd 向けオープンファイルの最大数の増加

登録されているコンテンツホストの数が 1000 を超える場合、**qdrouterd** はオープンファイルのデフォルトの最大数に到達することがあります。この問題を回避するには、Satellite サーバーとすべての外部 Capsule サーバーのオープンファイルの最大数を増加します。

- 以下の式を使用して、オープンファイルの必要な最大数を計算します。

```
(3 x コンテンツホストの数) + 100
```

たとえば、1020 のコンテンツホストの場合、新しい最大数は 3160 ((3 x 1020) + 100) に設定します。

- /etc/systemd/system/qdrouterd.service.d/limits.conf** ファイルを作成し、以下のテキストを追加します。

```
[Service]
LimitNOFILE=maximum_number_of_files
```

- 変更をユニットに適用します。

```
# systemctl daemon-reload
```

- Satellite のサービスを再起動します。

```
# foreman-maintain service restart
```

ホスト登録の遅れを減少

Satellite と、登録した各ホストの間の接続は、証明書を使用することで保護されています。ホストを登録すると、Satellite は、識別証明書と Puppet 証明書の 2 つを作成します。各証明書の作成に使用されるアルゴリズムでは、Red Hat Enterprise Linux カーネルのランダムなデータが必要になります。ホストを登録する際に十分なエントロピーが利用できない場合は、エントロピーのレベルが十分になるまで遅延が発生します。環境が非常に大きく、10,000 台以上のホストを使用する場合は、エントロピーが足りなくなるため、ホスト登録にかかる速度が遅くなる可能性があります。Linux カーネルで利用可能なエントロピーを増やし、Satellite のパフォーマンスをリスクを減らすための方法をいくつか使用できます。

デフォルトでは、Linux カーネルは、乱数のソースとして **/dev/random** デバイスを使用します。このデバイスはブロックデバイスで、エントロピーの数が少なく、乱数出力を正しく生成できないと判断した場合には、乱数の生成を停止します。それがこの待ち時間となり、ホストの登録に遅れが発生します。この問題を解決するには、ブロックデバイスではない、**/dev/urandom** デバイスを使用します。

ハードウェアサーバーの中には、ハードウェアの乱数ジェネレーターが含まれるものもあります。その乱数ジェネレーターを Red Hat Enterprise Linux カーネルがサポートする場合は、それを乱数のソースとして使用できます。詳細は、ハードウェアベンダーのドキュメントを参照してください。

仮想マシンで Satellite をホストしている場合には、ハードウェアサーバーの乱数ジェネレーターを、それがホストする仮想マシンで利用できるようにするものもあります。Red Hat Virtualization には、Red Hat Virtualization ホストからのエントロピーへの KVM 仮想マシンアクセスを提供する virtio RNG (乱数ジェネレーター) デバイスがあります。Red Hat Enterprise Linux 7.0 を実行しているゲストには、**rngd** をインストールして設定する必要があります。Red Hat Enterprise Linux 7.1 以降を実行しているゲストでは、ゲストのカーネルが、必要に応じてホストからエントロピーを取得します。ホストの乱数ジェネレーターをゲストと共有している場合は、ハードウェアの乱数ジェネレーターを使用することが推奨されます。

ゲストにおける乱数ジェネレーターの詳細は Red Hat Enterprise Linux 7 『[仮想化の導入および管理ガイド](#)』の「[乱数ジェネレーターデバイス](#)」を参照してください。その他のハイパーバイザーについては、ベンダーのドキュメントを参照してください。

乱数ジェネレーターデーモン (`rngd`) の詳細は Red Hat Enterprise Linux 7 の 『[セキュリティーガイド](#)』の「[乱数ジェネレーターの使用](#)」を参照してください。

付録B RED HAT SATELLITE へのカスタム設定の適用

satellite-installer を使用して初めて Satellite をインストールして設定する際には、**--foreman-proxy-dns-managed=false** と **--foreman-proxy-dhcp-managed=false** のインストーラーフラグを使用することで、DNS および DHCP 設定ファイルが Puppet に管理されないようにできます。インストーラーの初回実行時のこれらのフラグを指定しない場合は、インストーラーを再実行すると手動による変更がすべて上書きされるので、アップグレードする際に再実行できます。変更が上書きされても、復元手順を実行すると、手動での変更は復元できます。詳細は『インストールガイド』の「[Puppet 実行で上書きされた手動変更の復元](#)」を参照してください。

カスタム設定に利用可能なすべてのインストーラーフラグを表示するには、**satellite-installer --scenario satellite --full-help** を実行します。Puppet クラスには、Satellite インストーラーに公開されていないものもあります。これらのクラスを手動で管理して、インストーラーが値を上書きしないようにするには、設定ファイル `/etc/foreman-installer/custom-hiera.yaml` にエントリーを追加して設定値を指定します。この設定ファイルは YAML 形式で、**<puppet class>::<parameter name>: <value>** という形式を 1 行あたり 1 エントリーで記入します。このファイルで指定した設定値は、インストーラーを再起動しても維持されます。

一般的な例を示します。

- Apache で ServerTokens ディレクティブが製品名のみを返すようにするには、以下のようにします。

```
apache::server_tokens: Prod
```

- Apache サーバー署名をオフにするには、以下のようにします。

```
apache::server_signature: Off
```

- Pulp で pulp ワーカーの数を設定するには、以下のようにします。

```
pulp::num_workers: 8
```

Satellite インストーラー用の Puppet モジュールは、`/usr/share/foreman-installer/modules` と `/usr/share/katello-installer-base/modules` に保存されています。クラス、パラメーター、および値を調べるには、**.pp** ファイル (例: `moduleName/manifests/example.pp`) を確認してください。別の方法では、**grep** コマンドでキーワード検索を実行します。

値の設定によっては、Red Hat Satellite のパフォーマンスや機能に影響が出る意図しない結果がもたらされる場合があります。設定を適用する前に変更の影響を考慮して、実稼働以外の環境で最初に変更をテストしてください。実稼働以外の Satellite 環境がない場合は、Satellite インストーラーを **--noop** と **--verbose** のオプションを追加して実行します。変更によって問題が発生する場合は、該当箇所を `custom-hiera.yaml` から削除し、Satellite インストーラーを再実行します。特定の値を変更することが安全かどうかを確認する場合は、Red Hat サポートにお問い合わせください。

B.1. PUPPET 実行で上書きされた手動変更の復元

Puppet 実行で手動による設定が上書きされた場合でも、ファイルを元の状態に戻すことができます。以下の例では、Puppet 実行で上書きされた DHCP 設定ファイルを復元します。

1. 復元するファイルをコピーします。こうすることで、アップグレードに必要な変更があるか、ファイル間で比較できます。これは DNS や DHCP サービスでは一般的ではありません。

```
# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.backup
```

2. ログファイルを確認して、上書きされたファイルの md5sum をメモします。たとえば、以下のようになります。

```
# journalctl -xe
...
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed /etc/dhcp/dhcpd.conf to puppet
with sum 622d9820b8e764ab124367c68f5fa3a1
...
```

3. 上書きされたファイルを復元します。

```
# puppet filebucket restore --local --bucket \
/var/lib/puppet/clientbucket /etc/dhcp/dhcpd.conf \ 622d9820b8e764ab124367c68f5fa3a1
```

4. バックアップしたファイルと復元されたファイルを比べます。復元されたファイルに、アップグレードに必要な変更を追加します。