Red Hat Satellite 6.12

Capsule Server のインストール

Red Hat Satellite Capsule Server のインストール

Last Updated: 2024-02-15

Red Hat Satellite 6.12 Capsule Server のインストール

Red Hat Satellite Capsule Server のインストール

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

法律上の通知

概要

本書では、Red Hat Satellite Capsule 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

RED HAT ドキュメントへのフィードバック (英語のみ) 3

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. ストレージ要件 5
1.3. ストレージのガイドライン 6
1.4. サポート対象オペレーティングシステム 7
1.5. ポートとファイアウォールの要件 8
1.6. CAPSULE SERVER から SATELLITE SERVER への接続の有効化 12
1.7. SATELLITE SERVER およびクライアントから CAPSULE SERVER への接続の有効化 13

第2章 CAPSULE SERVER のインストール 14
2.1. SATELLITE SERVER への登録 14
2.2. SATELLITE INFRASTRUCTURE サブスクリプションのアタッチ 15
2.3. リポジトリーの設定 17
2.4. CAPSULE SERVER パッケージのインストール 18
2.5. CHRONYD とシステムクロックの同期 18
2.6. SSL 証明書を使用した CAPSULE SERVER の設定 19
2.7. SATELLITE WEB UI での適切な組織および場所の CAPSULE SERVER への割り当て 24

第3章 CAPSULE SERVER での追加設定の実行 26
3.1. ホストの登録とプロビジョニングのための CAPSULE の設定 26
3.2. 外部 CAPSULE での KATELLO エージェントの有効化 26
3.3. プルクライアントのリモート実行の設定 27
3.4. CAPSULE SERVER での OPENSCAP の有効化 27
3.5. CAPSULE SERVER へのライフサイクル環境の追加 28
3.6. マネージドホスト上での電源管理の有効化 29
3.7. CAPSULE SERVER での DNS、DHCP、および TFTP の設定 29

第4章 外部サービスを使用した CAPSULE SERVER の設定 31
4.1. 外部 DNS を使用した CAPSULE SERVER の設定 31
4.2. CAPSULE SERVER での外部 DHCP の設定 32
4.3. CAPSULE SERVER での外部 TFTP の設定 36
4.4. 外部 IDM DNS を使用した CAPSULE SERVER の設定 36

第5章 CAPSULE を使用した DHCP の管理 44
5.1. DHCPD API の保護 44

第6章 CAPSULE を使用した DNS の管理 45

付録A CAPSULE SERVER のスケーラビリティーに関する考慮事項 46

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Bugzilla でチケットを作成することでフィードバックを送信できます。

1. Bugzilla のWeb サイトに移動します。
2. **Component** フィールドで、**Documentation** を使用します。
3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

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のオペレーティングシステムには、以下の要件が適用されます。

- x86_64 アーキテクチャー
- 最低 4 コア 2.0 GHz CPU
- Capsule Server が機能するには、最低 12 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 Capsule は正常に動作しないことがあります。
- 利用可能なすべての更新が適用されてインストールされているサポートされているオペレーティングシステム
- 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 は **UTF-8** エンコーディングのみをサポートします。地域が米国で言語が英語の場合、システム全体のロケール設定として **en_US.utf-8** を設定します。Red Hat Enterprise Linux でのシステムロケールの設定に関する詳細は、**Red Hat Enterprise Linux 8 での基本的なシステム設定の構成**の システムロケールの設定 を参照してください。

Satellite には、カスタマーポータルに Red Hat Satellite Infrastructure サブスクリプションマニフェストが必要です。Satellite では、satellite-capsule-6.x リポジトリーが有効化され、同期されている必要があります。カスタマーポータルで Red Hat サブスクリプションマニフェストを作成、管理、およびエクスポートするには、**Subscription Central** での 接続された Satellite Server のマニフェストの作成と管理 を参照してください。

Satellite Server および Capsule Server では、ホスト名の短縮名はサポートされません。カスタム証明書を使用する場合には、カスタム証明書の Common Name (CN) は短縮名ではなく完全修飾ドメイン名 (FQDN) である必要があります。これは Satellite のクライアントには適用されません。

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。

警告

Capsule のバージョンは、インストールされている Satellite のバージョンと一致する必要があります。異なるバージョンは指定できません。たとえば、Capsule バージョン 6.12 を Satellite バージョン 6.11 に登録することはできません。

Capsule Server は、新たにプロビジョニングしたシステムにインストールしておく。このシステムは、Capsule Server を実行する機能としてだけに使用するようにします。Capsule Server が作成するローカルのユーザーとの競合を回避するために、新たにプロビジョニングしたシステムには、外部アイデンティティープロバイダーで設定した以下のユーザーを使用することはできません。

- apache
- foreman-proxy
- postgres
- pulp
- puppet
- qdrouterd
- redis

Capsule Server のスケーリングの詳細は、Capsule Server のスケーラビリティーに関する考慮事項 を参照してください。

認定ハイパーバイザー

Capsule Server 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の両方で完全にサポートされています。認定ハイパーバイザーの詳細は、Certified Guest Operating Systems in Red Hat OpenStack Platform, Red Hat Virtualization, Red Hat OpenShift Virtualization and Red Hat Enterprise Linux with KVM を参照してください。

SELinux モード

SELinux は、Enforcing モードまたは Permissive モードのいずれかで有効化されている必要があります。無効化された SELinux でのインストールはサポートされません。

FIPS Mode

FIPS モードで稼働する Red Hat Enterprise Linux システムに、Satellite をインストールできます。Satellite のインストール後に FIPS モードを有効にすることはできません。詳細は、**Red Hat Enterprise Linux セキュリティー強化** の FIPS モードが有効な RHEL 8 システムのインストール を参照してください。

注記

Satellite は、DEFAULT および FIPS 暗号化ポリシーをサポートしています。FUTURE 暗号化ポリシーは、Satellite および Capsule のインストールではサポートされていません。

1.2. 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、想定のユースケースシナリオに基づいており、各環境ごとに異なることがあります。

ランタイムサイズは Red Hat Enterprise Linux 6、7、および 8 のリポジトリーと同期して測定されました。

Capsule Server 上の PostgreSQL データベースのサイズは、Satellite Server から同期されるライフサイクル環境、コンテンツビュー、またはリポジトリーの数が増加するにつれて大幅に増加する可能性が

あります。Satellite の最大環境では、Capsule Server の PostgreSQL ディレクトリーのサイズが大きくなり、Satellite Server 上の PostgreSQL ディレクトリーの 2 倍または縮小する可能性があります。

表1.1 Capsule 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/lib/pulp	1 MB	300 GB
/var/lib/pgsql	100 MB	20 GB
/usr	3 GB	適用外
/opt/puppetlabs	500 MB	適用外

1.3. ストレージのガイドライン

Capsule Server をインストールして効率性を向上させる場合は、以下のガイドラインを考慮してください。

- **/tmp** ディレクトリーを別のファイルシステムとしてマウントする場合は、**/etc/fstab** ファイルの **exec** マウントオプションを使用する必要があります。**/tmp** が、**noexec** オプションを指定してすでにマウントされている場合は、オプションを **exec** に変更して、ファイルシステムを再マウントする必要があります。これは、**puppetserver** サービスが機能するために必要です。
- 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/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/lib/pulp** ディレクトリーには、帯域幅が高く、レイテンシーの低いストレージを使用してください。Red Hat Satellite には I/O を大量に使用する操作が多数あるため、高レイ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。

storage-benchmark スクリプトを使用して、このデータを取得できます。**storage-benchmark** スクリプトの使用の詳細は、Impact of Disk Speed on Satellite Operations を参照してください。

ファイルシステムのガイドライン

- 入出力レイ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。

ログファイルのストレージ

ログファイルは、**/var/log/messages/**、**/var/log/httpd/**、および **/var/lib/foreman-proxy/openscap/content/** に書き込まれます。**logrotate** を使用して、これらのファイルのサイズを管理できます。

詳細は、How to use logrotate utility to rotate log files を参照してください。

ログメッセージに必要なストレージの正確な容量は、インストール環境および設定により異なります。

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  nfs  context="system_u:object_r:var_lib_t:s0"  1 2
```

NFS 共有がすで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。

```
# restorecon -R /var/lib/pulp
```

重複パッケージ

同じパッケージが異なるリポジトリーで重複して存在する場合には、ディスク上に一度しか保存されません。そのため、重複するパッケージを別のリポジトリーに追加するときに必要な追加ストレージが少なくて済みます。ストレージの多くは、**/var/lib/pulp/** ディレクトリーにあります。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。

シンボリックリンク

/var/lib/pulp/ 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 のすべての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。

1.4. サポート対象オペレーティングシステム

Capsule Server は、Capsule Server のインストール時に利用可能な最新バージョンの Red Hat Enterprise Linux 8 でサポートされています。EUS または z-stream を含む以前の Red Hat Enterprise Linux バージョンはサポートされません。

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 がサポートする方法であれば他の方法でもインストールできます。

以下のオペレーティングシステムはインストーラーでサポートされ、パッケージがあり、Satellite のデプロイ用にテストされています。

表1.2 satellite-installer でサポートされるオペレーティングシステム

オペレーティングシステム	アーキテクチャー	備考
Red Hat Enterprise Linux 8	x86_64 のみ	

Satellite をインストールする前に、可能な場合はすべてのオペレーティングシステムの更新を適用してください。

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設定やソフトウェアは含めないようにしてください。この制限は、ハード化や Red Hat 以外の他社のセキュリ

ティーソフトウェアが該当します。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、Capsule 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。

新しくプロビジョニングしたシステムに Capsule Server をインストールします。

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 では、このシステムを Capsule Server の実行以外に使用するサポートはしていません。

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で通信を行うには、ベースオペレーティングシステム上で、必要なネットワークポートが開放/解放されているようにしてください。また、ネットワークベースのファイアウォールでも、必要なネットワークポートを開放する必要があります。

Satellite Server と Capsule Server の間のポートがインストール開始前に開放されていない場合は、Capsule Server のインストールに失敗します。

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、以下のセクション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オペレーティング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 ではない Capsule のクライアントであるホストには、Satellite Server へのアクセスは必要ありません。Satellite トポロジーの詳細は、**Satellite の概要、概念、およびデプロイメントの考慮事項** の Capsule のネットワーク 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。

以下の表は、宛先ポートとネットワークトラフィックの方向を示しています。

表1.3 Capsule の受信トラフィック

送信先ポート	プロトコル	サービス	ソース	用途	説明
53	TCP および UDP	DNS	DSN サーバーおよびクライアント	名前解決	DNS (オプション)
67	UDP	DHCP	クライアント	動的 IP	DHCP (オプション)

送信先ポート	プロトコル	サービス	ソース	用途	説明
69	UDP	TFTP	クライアント	TFTP サーバー (オプション)	
443、80	TCP	HTTPS, HTTP	クライアント	コンテンツの取得	コンテンツ
443、80	TCP	HTTPS, HTTP	クライアント	コンテンツホスト登録	Capsule CA RPM のインストール
443	TCP	HTTPS	Red Hat Satellite	コンテンツミラーリング	管理
443	TCP	HTTPS	Red Hat Satellite	Capsule API	スマートプロキシー機能
443	TCP	HTTPS	クライアント	コンテンツホスト登録	開始 ファクトのアップロード インストールされたパッケージとトレースの送信
1883	TCP	MQTT	クライアント	プルベースの REX (オプション)	REX ジョブ通知用のコンテンツホスト (オプション)
5646、5647	TCP	AMQP	クライアント	Goferd メッセージバス	クライアントにメッセージの転送 (オプション) Qpid ディスパッチャーと通信する Katello エージェント
8000	TCP	HTTP	クライアント	プロビジョニングテンプレート	クライアントインストーラー、iPXE または UEFI HTTP ブートのテンプレート取得
8000	TCP	HTTP	クライアント	PXE ブート	インストール
8140	TCP	HTTPS	クライアント	puppet-agent	クライアントの更新 (オプション)

送信先ポート	プロトコル	サービス	ソース	用途	説明
8443	TCP	HTTPS	クライアント	コンテンツホスト登録	非推奨、アップグレード前にデプロイされたクライアントホストにのみ必要
9090	TCP	HTTPS	クライアント	エンドポイントの登録	外部 Capsule Server へのクライアント登録
9090	TCP	HTTPS	クライアント	OpenSCAP	クライアントの設定
9090	TCP	HTTPS	検出されたノード	検出	ホストの検出とプロビジョニング
9090	TCP	HTTPS	Red Hat Satellite	Capsule API	Capsule の機能

Satellite Server に直接接続されたマネージドホストは、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オペレーティングシステムが含まれます。

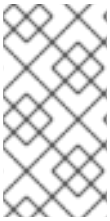
DHCP Capsule は、DHCP IPAM が設定されたサブネット内のホストに対して ICMP ping および TCP Echo 接続の試行を実行し、使用が検討されている IP アドレスが空いているかどうかを確認します。この動作は、**satellite-installer --foreman-proxy-dhcp-ping-free-ip=false** を使用してオフにできます。

表1.4 Capsule 送信トラフィック

送信先ポート	プロトコル	サービス	宛先	用途	説明
	ICMP	ping	クライアント	DHCP	解放されている IP チェック (オプション)
7	TCP	echo	クライアント	DHCP	解放されている IP チェック (オプション)
22	TCP	SSH	ターゲットホスト	リモート実行	ジョブの実行

送信先ポート	プロトコル	サービス	宛先	用途	説明
53	TCP および UDP	DNS	インターネット上の DNS サーバー	DNS サーバー	DNS レコードの解決 (オプション)
53	TCP および UDP	DNS	DNS サーバー	--capsule-dns	DNS 競合の検証 (オプション)
68	UDP	DHCP	クライアント	動的 IP	DHCP (オプション)
443	TCP	HTTPS	Satellite	Capsule	Capsule 設定管理 テンプレートの取得 OpenSCAP リモート実行結果のアップロード
443	TCP	HTTPS	Red Hat ポータル	SOS レポート	サポートケースの支援 (オプション)
443	TCP	HTTPS	Satellite	コンテンツ	同期
443	TCP	HTTPS	Satellite	クライアント通信	クライアントから Satellite への要求転送
443	TCP	HTTPS	Infoblox DHCP サーバー	DHCP 管理	DHCP に Infoblox を使用する場合、DHCP リースの管理 (オプション)
623			クライアント	電源管理	BMC のオン/オフ/サイクル/ステータス
5646	TCP	AMQP	Satellite Server	Katello Agent	Capsule の Qpid ディスパッチルーターへのメッセージの転送 (オプション)

送信先ポート	プロトコル	サービス	宛先	用途	説明
7911	TCP	DHCP、OMAPI	DHCP サーバー	DHCP	DHCP ターゲットは、**--foreman-proxy-dhcp-server** を使用して設定される。デフォルトは localhost。 ISC と **remote_isc** は、デフォルトが 7911 で、OMAPI を使用する設定可能なポートを使用する
8443	TCP	HTTPS	クライアント	検出	Capsule は、検出されたホストに再起動コマンドを送信する (オプション)



注記

ICMP から Port 7 UDP および TCP を拒否することはできませんが、破棄できます。DHCP Capsule は ECHO REQUEST をクライアントネットワークに送信し、IP アドレスが解放されていることを確認します。応答があると、IP アドレスの割り当てが回避されます。

1.6. CAPSULE SERVER から SATELLITE SERVER への接続の有効化

Satellite Server で、Capsule Server から Satellite Server に対する受信接続を有効にして、再起動後にもルールが保持されるようにする必要があります。

前提条件

- Capsule Server は Satellite Server のクライアントであることから、クライアントが Satellite との通信用に接続できるように、Satellite Server でファイアウォールルールが設定されていること。詳細は、**オンラインネットワーク環境からの Satellite Server のインストール** の クライアントから Satellite Server への接続の有効化 を参照してください。

手順

1. Satellite Server で、次のコマンドを入力して Capsule から Satellite への通信に使用するポートを開放します。

```
# firewall-cmd --add-port="5646/tcp"
```

2. 変更を永続化します。

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

1.7. SATELLITE SERVER およびクライアントから CAPSULE SERVER への接続の有効化

Capsule のインストール先のベースオペレーティングシステムで、Satellite Server およびクライアントから Capsule Server への受信接続を有効にして、再起動後にもこれらのルールが維持されるようにします。

手順

1. Capsule のインストール先のベースオペレーティングシステムで、次のコマンドを入力して、Satellite Server およびクライアントから Capsule Server への通信に使用するポートを開放します。

```
# 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="5647/tcp" \
--add-port="8140/tcp" \
--add-port="8443/tcp" \
--add-port="8000/tcp" --add-port="9090/tcp"
```

2. 変更を永続化します。

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

検証

- 以下のコマンドを入力します。

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

詳細は、**Red Hat Enterprise Linux 8 のネットワークの保護**の firewalld の使用および設定 を参照してください。

第2章 CAPSULE SERVER のインストール

Capsule Server をインストールする前に、お使いの環境がインストール要件を満たしていることを確認してください。詳細は、インストールのための環境準備 を参照してください。

2.1. SATELLITE SERVER への登録

この手順を使用して、Capsule Server をインストールするベースオペレーティングシステムを Satellite Server に登録します。

Red Hat Subscription Manifest の前提条件

- Satellite Server にマニフェストをインストールし、Capsule が所属する組織に適したリポジトリーが含まれている必要がある。
- マニフェストには、Capsule をインストールするベースオペレーティングシステムのリポジトリーと、Capsule に接続するクライアントが含まれている必要がある。
- リポジトリーは、同期されている必要がある。

マニフェストおよびリポジトリーに関する詳細は **コンテンツ管理** の Red Hat サブスクリプションの管理 を参照してください。

プロキシーとネットワークの前提条件

- Satellite Server のベースシステムは、Capsule のベースオペレーティングシステムのホスト名を解決できる必要があり、Capsule のベースシステムは Satellite Server のベースオペレーティングシステムのホスト名を解決できる必要がある。
- Capsule Server と Satellite Server の間でクライアント証明書認証を使用した HTTPS 接続が可能であることを確認します。Capsule Server と Satellite Server 間の HTTP プロキシーはサポートされていません。
- 要件に合わせてホストとネットワークベースのファイアウォールを設定する必要がある。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。ホスト登録機能、Satellite API、または Hammer CLI を使用して、ホストを Satellite に登録できます。

手順

1. Satellite Web UI で、**ホスト > ホストの登録** に移動します。
2. **Generate** をクリックして登録コマンドを作成します。
3. **ファイル** アイコンをクリックして、コマンドをクリップボードにコピーします。
4. 登録するホストにログインして、以前に作成したコマンドを実行します。
5. **/etc/yum.repos.d/redhat.repo** ファイルをチェックして、適切なリポジトリーが有効であることを確認します。

CLI 手順

1. Hammer CLI を使用してホスト登録コマンドを生成します。

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

ホストが Satellite Server の SSL 証明書を信頼しない場合は、登録コマンドに **--insecure** フラグを追加して SSL 検証を無効にすることができます。

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

2. 登録するホストにログインして、以前に作成したコマンドを実行します。
3. **/etc/yum.repos.d/redhat.repo** ファイルをチェックして、適切なリポジトリーが有効であることを確認します。

API の手順

1. Satellite API を使用してホスト登録コマンドを生成します。

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

ホストが Satellite Server の SSL 証明書を信頼しない場合は、登録コマンドに **--insecure** フラグを追加して SSL 検証を無効にすることができます。

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

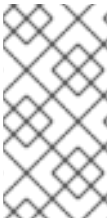
アクティベーションキーを使用すると、その環境を簡単に指定できます。詳細は、**コンテンツの管理** の アクティベーションキーの管理 を参照してください。

コマンドライン引数としてパスワードを入力するには、**username:password** 構文を使用します。これにより、パスワードがシェル履歴に保存される可能性があることに注意してください。

ホストの登録に関する情報は **ホストの管理** の Red Hat Satellite へのホストの登録 を参照してください。

2. 登録するホストにログインして、以前に作成したコマンドを実行します。
3. **/etc/yum.repos.d/redhat.repo** ファイルをチェックして、適切なリポジトリーが有効であることを確認します。

2.2. SATELLITE INFRASTRUCTURE サブスクリプションのアタッチ



注記

Satellite で SCA を有効にしている場合は、この手順をスキップしてください。subscription-manager を使用して Red Hat Satellite Infrastructure Subscription サブスクリプションを Capsule Server にアタッチする必要はありません。SCA の詳細は、Simple Content Access を参照してください。

Capsule Server の登録後に、サブスクリプションプール ID を特定して、利用可能なサブスクリプションをアタッチする必要があります。Red Hat Satellite Infrastructure サブスクリプションは、Red Hat Satellite および Red Hat Enterprise Linux コンテンツにアクセスできるようになります。Red Hat Enterprise Linux 7 では、Red Hat Software Collections (RHSCL) にアクセスできます。必要なサブスクリプションはこれだけです。

Red Hat Satellite Infrastructure は、Satellite (以前は Smart Management と呼ばれていました) を提供するすべてのサブスクリプションに含まれています。詳細は、**Red Hat ナレッジベース** の Satellite Infrastructure サブスクリプション MCT3718 MCT3719 を参照してください。

サブスクリプションがシステムに割り当てられていない場合には、利用可能として分類されます。利用可能な Satellite サブスクリプションが見つからない場合は、Red Hat ナレッジベースソリューション Red Hat Subscription Manager に登録されているクライアントが使用したサブスクリプションを把握するにはどうすればよいですか? 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します。

手順

1. Satellite Infrastructure サブスクリプションのプール ID を特定します。

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

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 Name:   Red Hat Satellite Infrastructure Subscription
Provides:            Red Hat Satellite
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64
                     Red Hat Satellite Capsule
                     Red Hat Ansible Engine
                     Red Hat Satellite with Embedded Oracle
                     Red Hat Satellite 5 Managed DB
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                     Red Hat Beta
                     Red Hat Software Collections Beta (for RHEL Server)
                     Red Hat Enterprise Linux Server
                     Red Hat Enterprise Linux for x86_64
                     Red Hat Satellite Proxy
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Discovery
SKU:                 MCT3718
Contract:
Pool ID:             8aca43dd771bf31101771c0231f906a5
Provides Management: Yes
Available:           10
Suggested:           1
Service Type:        L1-L3
```

```
Roles:
Service Level:      Premium
Usage:
Add-ons:
Subscription Type:  Standard
Starts:             11/11/2020
Ends:               11/11/2023
Entitlement Type:   Physical
```

2. サブスクリプションプール ID を書き留めます。上記の例と、実際のサブスクリプションプール ID は異なります。

3. Capsule Server の実行先のベースオペレーティングシステムに、Satellite Infrastructure サブスクリプションを割り当てます。SCA が Satellite Server で有効になっている場合は、この手順をスキップできます。

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

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。

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

4. オプション: Satellite Infrastructure サブスクリプションが割り当てられていることを確認します。

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

2.3. リポジトリーの設定

この手順を使用して、Capsule Server のインストールに必要なリポジトリーを有効にします。

1. すべてのリポジトリーを無効にします。

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

2. 以下のリポジトリーを有効にします。

```
# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms \
--enable=satellite-capsule-6.12-for-rhel-8-x86_64-rpms \
--enable=satellite-maintenance-6.12-for-rhel-8-x86_64-rpms
```

3. モジュールを有効にします。

```
# dnf module enable satellite-capsule:el8
```

注記

モジュール **satellite:el8** を有効にすると、**postgresql:10** および **ruby:2.5** との競合について警告が表示されます。これは、これらのモジュールが Red Hat Enterprise Linux 8 でデフォルトのモジュールバージョンに設定されているためです。モジュール **satellite:el8** には、モジュール **postgresql:12** および **ruby:2.7** への依存関係があり、**satellite:el8** モジュールで有効になります。これらの警告はインストールプロセスの失敗の原因にはならないため、安全に無視できます。Red Hat Enterprise Linux 8 のモジュールとライフサイクルストリームの詳細については、Red Hat Enterprise Linux Application Streams Life Cycle を参照してください。

注記

Red Hat Virtualization (RHV) がホストする仮想マシンとして、Capsule Server をインストールする場合は、**Red Hat Common** リポジトリーも有効にして、RHV ゲストエージェントとドライバーを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 Red Hat Enterprise Linux へのゲストエージェントとドライバーのインストール を参照してください。

4. メタデータを消去します。

```
# dnf clean all
```

5. オプション: 必要なリポジトリーが有効になっていることを確認します。

```
# dnf repolist enabled
```

2.4. CAPSULE SERVER パッケージのインストール

Capsule Server パッケージをインストールする前に、ベースオペレーティングシステムにインストールした全パッケージを更新する必要があります。

手順

Capsule Server をインストールするには、以下の手順を実行します。

1. すべてのパッケージを更新します。

```
# dnf update
```

2. Satellite Server パッケージをインストールします。

```
# dnf install satellite-capsule
```

2.5. CHRONYD とシステムクロックの同期

時間のずれを最小限に抑えるには、Capsule Server をインストールするベースオペレーティングシステムのシステムクロックを Network Time Protocol (NTP) サーバーと同期する必要があります。ベースオペレーティングシステムのクロックが正しく設定されていない場合には、証明書の検証に失敗する可能性があります。

chrony スイートの詳細は、**Red Hat Enterprise Linux 8 の基本的なシステム設定**の Chrony スイートを使用した NTP の設定 を参照してください。

手順

1. **chrony** パッケージをインストールします。

```
# dnf install chrony
```

2. **chronyd** サービスを起動して、有効にします。

```
# systemctl enable --now chronyd
```

2.6. SSL 証明書を使用した CAPSULE SERVER の設定

Red Hat Satellite は SSL 証明書を使用して、Satellite Server、外部 Capsule Server、全ホストの間の暗号化通信を有効にします。組織の要件によっては、デフォルトの証明書またはカスタムの証明書で Capsule Server を設定する必要があります。

- また、デフォルトの SSL 証明書を使用する場合には、外部 Capsule Server ごとに異なるデフォルトの SSL 証明書を設定する必要があります。詳細は、「デフォルトの SSL 証明書を使用した Capsule Server の設定」 を参照してください。
- また、カスタムの SSL 証明書を使用する場合には、外部 Capsule Server ごとに異なるカスタムの SSL 証明書を使用して設定する必要があります。詳細は、「カスタム SSL 証明書を使用した Capsule Server の設定」 を参照してください。

2.6.1. デフォルトの SSL 証明書を使用した Capsule Server の設定

本セクションを使用して、Satellite Server のデフォルトの証明局 (CA) が署名した SSL 証明書を使用して Capsule Server を設定します。

前提条件

- Capsule Server が Satellite Server に登録されている。詳細は、Satellite Server への登録 を参照してください。
- Capsule Server パッケージがインストールされている。詳細は、Capsule Server パッケージのインストール を参照してください。

手順

1. Satellite Server で Capsule Server の全ソース証明書ファイルを保存するには、**root** ユーザーのみがアクセスできるディレクトリーを作成します (例: **/root/capsule_cert**)。

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

2. Satellite Server で、Capsule Server の **/root/capsule_cert/capsule.example.com -certs.tar** 証明書アーカイブを生成します。

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

capsule-certs-generate コマンドが返す **satellite-installer** コマンドのコピーをメモし、Capsule Server に証明書をデプロイします。

capsule-certs-generate の出力例

```
output omitted
satellite-installer --scenario capsule \
--certs-tar-file "/root/capsule_cert/capsule.example.com-certs.tar" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule.example.com" \
--foreman-proxy-oauth-consumer-key "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f" \
--foreman-proxy-oauth-consumer-secret "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"
```

3. Satellite Server から、証明書アーカイブファイルを Capsule Server にコピーします。

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

4. Capsule Server で、証明書をデプロイするには、**capsule-certs-generate** コマンドにより返された **satellite-installer** コマンドを入力します。
Satellite へのネットワーク接続やポートをまだ開いていない場合は、**--foreman-proxy-register-in-foreman** オプションを **false** に設定すると、Capsule が Satellite へ接続を試行しなくなり、エラー報告がなくなります。ネットワークとファイアウォールを適切に設定したら、このオプションを **true** にして再度インストーラーを実行します。

重要

証明書のデプロイ後に、証明書のアーカイブファイルを削除しないでください。このアーカイブは、Capsule Server のアップグレード時などに必要になります。

2.6.2. カスタム SSL 証明書を使用した Capsule Server の設定

Satellite Server がカスタムの SSL 証明書を使用するように設定する場合は、この設定時に、外部の各 Capsule Server も、異なるカスタム SSL 証明書で設定する必要があります。

カスタム証明書を使用して Capsule Server を設定するには、Capsule Server ごとに以下の手順を実行します。

1. 「Capsule Server のカスタム SSL 証明書の作成」
2. 「カスタムの SSL 証明書の Capsule Server へのデプロイ」
3. 「ホストへの カスタム SSL 証明書のデプロイ」

2.6.2.1. Capsule Server のカスタム SSL 証明書の作成

Satellite Server で、Capsule Server 用にカスタムの証明書を作成します。Capsule Server 用のカスタムの SSL 証明書がすでにある場合には、以下の手順は省略してください。

手順

1. ソースの証明書ファイルすべてを保存するには、**root** ユーザーだけがアクセスできるディレクトリーを作成します。

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

2. 証明書署名要求 (CSR) に署名する秘密鍵を作成します。
秘密鍵は暗号化する必要がないことに注意してください。パスワードで保護された秘密鍵を使用する場合は、秘密鍵のパスワードを削除します。

この Capsule Server 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/capsule_cert/capsule_cert_key.pem 4096
```

3. CSR 用の **/root/capsule_cert/openssl.cnf** 設定ファイルを作成して、以下のコンテンツを追加します。

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

[ req_distinguished_name ]
CN = capsule.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 = capsule.example.com
```

4. CSR を生成します。

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

1. 秘密鍵へのパス
2. 設定ファイルへのパス

3 生成する CSR へのパス

5. 認証局 (CA) に証明書署名要求を送信します。Satellite Server と Capsule Server の証明書には同じ CA が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。証明書要求の送信方法にはさまざまなものがあるため、推奨される方法について CA にお問い合わせください。要求すると、CA 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ことになります。

2.6.2.2. カスタムの SSL 証明書の Capsule Server へのデプロイ

この手順を使用して、証明局が署名したカスタムの SSL 証明書で、Capsule Server を設定します。**capsule-certs-generate** コマンドにより返される、**satellite-installer** コマンドは、Capsule Server ごとに一意となっています。複数の Capsule Server に同じコマンドを使用しないでください。

前提条件

- Satellite Server は、カスタムの証明書で設定されている。詳細は、**オンラインネットワーク環境からの Satellite Server のインストール** の カスタムの SSL 証明書での Satellite Server の設定 を参照してください。
- Capsule Server が Satellite Server に登録されている。詳細は、Satellite Server への登録 を参照してください。
- Capsule Server パッケージがインストールされている。詳細は、Capsule Server パッケージのインストール を参照してください。

手順

1. Satellite Server で、カスタムの SSL 証明書の入力ファイルを検証します。

```
# katello-certs-check \
-t capsule -c /root/capsule_cert/capsule_cert.pem \ 1
-k /root/capsule_cert/capsule_cert_key.pem \ 2
-b /root/capsule_cert/ca_cert_bundle.pem 3
```

1 認証局が署名した Capsule Server の証明書ファイルへのパス

2 Capsule Server 証明書の署名に使用した秘密鍵へのパス

3 認証局バンドルへのパス

/root/capsule_cert/openssl.cnf 設定ファイルの証明書のコモンネーム **CN =** に、* のワイルドカードの値を設定した場合には、**katello-certs-check** コマンドに **-t capsule** オプションを追加する必要があります。

このコマンドに成功すると、**capsule-certs-generate** コマンド 2 つが返されます。このうちのいずれか 1 つを、Capsule Server の証明書アーカイブの生成に使用する必要があります。

katello-certs-check の出力例

```
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/capsule_cert/capsule_cert.pem" \
    --server-key "/root/capsule_cert/capsule_cert_key.pem" \
    --server-ca-cert "/root/capsule_cert/ca_cert_bundle.pem" \

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/capsule_cert/capsule_cert.pem" \
    --server-key "/root/capsule_cert/capsule_cert_key.pem" \
    --server-ca-cert "/root/capsule_cert/ca_cert_bundle.pem" \
    --certs-update-server
```

2. Satellite Server で、**katello-certs-check** コマンドの出力をもとに、要件に合わせて、**capsule-certs-generate** コマンドを入力し、新規または既存の Capsule の証明書を生成します。このコマンドで **$CAPSULE** を Capsule Server の FQDN に変更します。

3. **capsule-certs-generate** コマンドが返す **satellite-installer** コマンドのコピーをメモし、Capsule Server に証明書をデプロイします。

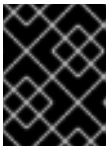
   capsule-certs-generate の出力例

```
output omitted
satellite-installer --scenario capsule \
--certs-tar-file "/root/capsule.example.com-certs.tar" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule.example.com" \
--foreman-proxy-oauth-consumer-key "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f" \
--foreman-proxy-oauth-consumer-secret "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"
```

4. Satellite Server から、証明書アーカイブファイルを Capsule Server にコピーします。

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

5. Capsule Server で、証明書をデプロイするには、**capsule-certs-generate** コマンドにより返された **satellite-installer** コマンドを入力します。
   Satellite へのネットワーク接続やポートをまだ開いていない場合は、**--foreman-proxy-register-in-foreman** オプションを **false** に設定すると、Capsule が Satellite へ接続を試行しなくなり、エラー報告がなくなります。ネットワークとファイアウォールを適切に設定したら、このオプションを **true** にして再度インストーラーを実行します。



重要

証明書のデプロイ後に、証明書のアーカイブファイルを削除しないでください。このアーカイブは、Capsule Server のアップグレード時などに必要になります。

2.6.2.3. ホストへの カスタム SSL 証明書のデプロイ

Capsule Server がカスタムの SSL 証明書を使用するよう設定した後に、Capsule Server に登録されている全ホストに **katello-ca-consumer** パッケージもインストールする必要があります。

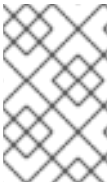
手順

- 各ホストに **katello-ca-consumer** パッケージをインストールします。

```
# dnf install http://capsule.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

2.7. SATELLITE WEB UI での適切な組織および場所の CAPSULE SERVER への割り当て

Capsule Server パッケージのインストール後に、組織または場所が複数ある場合には、Satellite Web UI で Capsule に正しい組織と場所を割り当てて Capsule が表示されるようにする必要があります。



注記

組み込み Capsule を使用して Satellite Server と同じ場所に Capsule を割り当てると、Red Hat Insights が Insights インベントリーをアップロードできなくなります。インベントリーのアップロードを有効化するには、両方の Capsule の SSH キーを同期します。

手順

1. Satellite Web UI にログインします。
2. 画面左上にある **Organization** リストから、**Any Organization** を選択します。
3. 画面左上にある **Location** リストから、**Any Location** を選択します。
4. Satellite Web UI で、**Hosts** > **All Hosts** に移動し、Capsule Server を選択します。
5. **Select Actions** 一覧から、**Assign Organization** を選択します。
6. **Organization** リストから、この Capsule を割り当てる組織を選択します。
7. **Fix Organization on Mismatch** をクリックします。
8. **Submit** をクリックします。
9. Capsule Server を選択します。**Select Actions** 一覧から、**Assign Location** を選択します。
10. **Location** リストから、この Capsule を割り当てる場所を選択します。
11. **Fix Location on Mismatch** をクリックします。
12. **Submit** をクリックします。
13. Satellite Web UI で、**Administer** > **Organizations** に移動して、Capsule を割り当てた組織をクリックします。
14. **Capsules** タブをクリックし、Capsule Server が **Selected items** 一覧に表示されていることを確認してから **Submit** をクリックします。
15. Satellite Web UI で、**Administer** > **Locations** に移動して、Capsule を割り当てたロケーションをクリックします。

16. **Capsules** タブをクリックし、Capsule Server が **Selected items** 一覧に表示されていることを確認してから **Submit** をクリックします。

検証

必要に応じて、Capsule Server が Satellite Web UI に正しく表示されているかどうかを検証できます。

1. **Organization** 一覧で、組織を選択します。
2. **Location** リストから場所を選択します。
3. Satellite Web UI で、**Hosts** > **All Hosts** に移動します。
4. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。

第3章 CAPSULE SERVER での追加設定の実行

以下の章を使用して、Capsule Server の追加設定を行います。

3.1. ホストの登録とプロビジョニングのための CAPSULE の設定

この手順を使用して Capsule を設定し、Satellite Server の代わりに Capsule Server を使用してホストを登録およびプロビジョニングできるようにします。

手順

- Satellite Server で、信頼できるプロキシーのリストに Capsule を追加します。
これは、Capsule によって設定された **X-Forwarded-For** HTTP ヘッダーを介して転送されるホストの IP アドレスを Satellite が認識するために必要です。デフォルトでは、セキュリティー上の理由から Satellite はローカルホストからのみ、この HTTP ヘッダーを認識します。信頼できるプロキシーを、Capsule の有効な IPv4 または IPv6 アドレス、またはネットワーク範囲として入力できます。

警告

潜在的なセキュリティーリスクが生じるため、広すぎるネットワーク範囲を使用しないでください。

以下のコマンドを入力します。このコマンドは、Satellite に現在保存されているリストを上書きすることに注意してください。したがって、以前に信頼できるプロキシーを設定している場合は、それらもコマンドに含める必要があります。

```
# satellite-installer \
--foreman-trusted-proxies "127.0.0.1/8" \
--foreman-trusted-proxies "::1" \
--foreman-trusted-proxies "My_IP_address" \
--foreman-trusted-proxies "My_IP_range"
```

localhost エントリーは必須です。省略しないでください。

検証

1. Satellite インストーラーの完全なヘルプを使用して、現在信頼できるプロキシーをリスト表示します。

```
# satellite-installer --full-help | grep -A 2 "trusted-proxies"
```

2. 現在のリストには、必要なすべての信頼できるプロキシーが含まれています。

3.2. 外部 CAPSULE での KATELLO エージェントの有効化

リモート実行は、コンテンツホスト上のパッケージを管理するための主要な方法です。非推奨の Katello エージェントを使用できるようにするには、各 Capsule で有効にする必要があります。

手順

- Katello エージェントインフラストラクチャーを有効にするには、以下のコマンドを入力します。

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-enable-katello-agent=true
```

3.3. プルクライアントのリモート実行の設定

デフォルトでは、リモート実行はスクリプトプロバイダーのトランスポートメカニズムとして SSH を使用します。ただし、リモート実行にはプルベースのトランスポート機能があり、インフラストラクチャーが Capsule からホストへの発信接続を禁止している場合に使用できます。

これは、Capsule の **pull-mqtt** モードと、ホスト上で実行されるプルクライアントで設定されます。**pull-mqtt** モードを Katello エージェントから移行するように設定します。これは、非推奨のプルベースの転送方法です。

注記

pull-mqtt モードは、スクリプトプロバイダーでのみ機能します。Ansible およびその他のプロバイダーは、引き続きデフォルトのトランスポート設定を使用します。

このモードは Capsule ごとに設定されます。一部の Capsule は **pull-mqtt** モードを使用するように設定できますが、その他の Capsule は SSH を使用します。この場合、特定のホスト上の 1 つのリモートジョブがプルクライアントを使用し、同じホスト上の次のジョブが SSH を使用する可能性があります。このシナリオを回避するには、すべての Capsule が同じモードを使用するように設定します。

手順

1. 関連する各 Capsule Server でプルベースのトランスポートを有効にします。

```
 # satellite-installer --scenario capsule \
--foreman-proxy-plugin-remote-execution-script-mode pull-mqtt
```

2. ポート 1883 で MQTT サービスを許可するようにファイアウォールを設定します。

```
# firewall-cmd --add-port="1883/tcp"
# firewall-cmd --runtime-to-permanent
```

3. **pull-mqtt** モードでは、ホストは、登録先の Capsule にジョブ通知をサブスクライブします。したがって、Satellite Server が同じ Capsule にリモート実行ジョブを送信するようにすることが推奨されます。これを行うには、Satellite Web UI で **Administer** > **Settings** の順に移動します。**Content** タブで、**リモート実行用に Capsule 経由で登録推奨** の値を **Yes** に設定します。

4. Capsule にプルベースのトランスポートを設定したら、各ホストでも設定する必要があります。詳細は、**ホストの管理** の リモート実行のトランスポートモード を参照してください。

3.4. CAPSULE SERVER での OPENSCAP の有効化

Satellite Server および Satellite Server に統合された Capsule では、デフォルトで OpenSCAP は有効になっています。外部 Capsule で OpenSCAP プラグインとコンテンツを使用する場合には、各 Capsule で OpenSCAP を有効にする必要があります。

手順

- OpenSCAP を有効にするには、次のコマンドを入力します。

```
# satellite-installer --scenario capsule \
--enable-foreman-proxy-plugin-openscap \
--foreman-proxy-plugin-openscap-puppet-module true
```

Puppet を使用してコンプライアンスポリシーをデプロイする場合は、最初に Puppet を有効にする必要があります。詳細は、**Red Hat Satellite で Puppet インテグレーションを使用した設定の管理** を参照してください。

3.5. CAPSULE SERVER へのライフサイクル環境の追加

Capsule Server でコンテンツ機能が有効な場合は、環境を追加して、Capsule が Satellite Server のコンテンツを同期し、コンテンツをホストシステムに提供できるようにする必要があります。

ライブラリー ライフサイクル環境は、CDN がリポジトリーを更新するたびに自動的に Capsule が同期をトリガーするため、Capsule Server に割り当てないでください。自動で同期される場合、Capsule 上の複数のシステムリソースや Satellite と Capsule 間のネットワーク帯域幅、および Capsule 上の利用可能なディスク領域が消費される可能性があります。

Satellite Server の Hammer CLI または Satellite Web UI を使用できます。

手順

1. Satellite Web UI で、**Infrastructure** > **Capsule** に移動し、ライフサイクルを追加する Capsule を選択します。
2. **Edit** をクリックしてから、**Life Cycle Environments** タブをクリックします。
3. 左側のメニューから、Capsule に追加するライフサイクル環境を選択し、**Submit** をクリックします。
4. Capsule のコンテンツを同期するには、**Overview** タブをクリックして **Synchronize** をクリックします。
5. **Optimized Sync** または **Complete Sync** を選択します。
同期の各タイプの定義については、リポジトリーの復旧 を参照してください。

CLI 手順

1. Satellite Server で、Capsule Server の全リストを表示するには、以下のコマンドを入力します。

```
# hammer capsule list
```

ライフサイクルを追加する Capsule の Capsule ID をメモします。

2. ID を使用して、Capsule の詳細を確認します。

```
# hammer capsule info \
--id My_capsule_ID
```

3. Capsule Server で利用可能なライフサイクル環境を表示するには、以下のコマンドを入力して、ID と組織名を書き留めます。

```
# hammer capsule content available-lifecycle-environments \
--id My_capsule_ID
```

4. ライフサイクル環境を Capsule Server に追加します。

```
# hammer capsule content add-lifecycle-environment \
--id My_capsule_ID \
--lifecycle-environment-id My_Lifecycle_Environment_ID
--organization "My_Organization"
```

Capsule Server に追加するライフサイクル環境ごとに繰り返します。

5. Satellite から Capsule にコンテンツを同期します。

- Satellite Server 環境のすべてのコンテンツを Capsule Server に同期するには、以下のコマンドを入力します。

```
# hammer capsule content synchronize \
--id My_capsule_ID
```

- Satellite Server から Capsule Server に特定のライフサイクル環境を同期するには、以下のコマンドを入力します。

```
# hammer capsule content synchronize \
--id My_capsule_ID
--lifecycle-environment-id My_Lifecycle_Environment_ID
```

3.6. マネージドホスト上での電源管理の有効化

Intelligent Platform Management Interface (IPMI) または類似するプロトコルを使用してマネージドホストで電源管理タスクを実行するには、Capsule Server でベースボード管理コントローラー (BMC) モジュールを有効にする必要があります。

前提条件

- すべてのマネージドホストには、BMC タイプのネットワークインターフェイスが必要である。Capsule Server 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。詳細は、**ホストの管理** の ベースボード管理コントローラー (BMC) インターフェイスの追加 を参照してください。

手順

- BMC を有効にするには、以下のコマンドを入力します。

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

3.7. CAPSULE SERVER での DNS、DHCP、および TFTP の設定

DNS、DHCP および TFTP サービスを Capsule Server で設定するには、お使いの環境に適したオプションを指定して **satellite-installer** コマンドを使用します。設定可能なオプションの全リストを表示するには、**satellite-installer --scenario satellite --help** コマンドを入力します。

設定を変更するには、**satellite-installer** コマンドを再び実行する必要があります。コマンドは複数回実行でき、実行するたびにすべての設定ファイルが変更された値で更新されます。

代わりに外部の DNS、DHCP および TFTP サービスを使用するには、*4章 外部サービスを使用した Capsule Server の設定* を参照してください。

Multihomed DHCP の詳細の追加

マルチホーム DHCP を使用する場合は、インストーラーに通知する必要があります。

前提条件

- DNS サーバーの適切なネットワーク名 (**dns-interface**) が用意されている必要がある。
- DHCP サーバーの適切なインターフェイス名 (**dhcp-interface**) が用意されている必要がある。
- 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する。

手順

- お使いの環境に適したオプションで、**satellite-installer** コマンドを入力してください。以下の例では、完全なプロビジョニングサービスの設定を示しています。

```
# satellite-installer --scenario capsule \
--foreman-proxy-dns true \
--foreman-proxy-dns-managed true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-reverse 2.0.192.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-additional-interfaces eth1 \
--foreman-proxy-dhcp-additional-interfaces eth2 \
--foreman-proxy-dhcp-range "192.0.2.100 192.0.2.150" \
--foreman-proxy-dhcp-gateway 192.0.2.1 \
--foreman-proxy-dhcp-nameservers 192.0.2.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername 192.0.2.3
```

DHCP、DNS および TFTP サービスの設定に関する情報は、**ホストのプロビジョニング** 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。

第4章 外部サービスを使用した CAPSULE SERVER の設定

Capsule Server で DNS、DHCP、および TFTP サービスを設定しない場合は、外部 DNS、DHCP、および TFTP サービスと連携させる Capsule Server の設定のセクションを使用します。

4.1. 外部 DNS を使用した CAPSULE SERVER の設定

外部 DNS を使用して Capsule Server を設定できます。Capsule Server は **nsupdate** ユーティリティー –を使用して、リモートサーバーで DNS レコードを更新します。

変更を永続的に保存するには、お使いの環境に適したオプションを指定して、**satellite-installer** コマンドを入力する必要があります。

前提条件

- 外部 DNS サーバーが設定されている必要がある。
- このガイドは、既存のインストールがあることを前提としています。

手順

1. 外部 DNS サーバーの **/etc/rndc.key** ファイルを Capsule Server にコピーします。

```
# scp root@dns.example.com:/etc/rndc.key /etc/foreman-proxy/rndc.key
```

2. 所有者、パーミッション、SELinux コンテキストを設定します。

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

3. **nsupdate** ユーティリティーをテストするには、ホストをリモートで追加します。

```
# echo -e "server DNS_IP_Address\n \
update add aaa.example.com 3600 IN A Host_IP_Address\n \
send\n" | nsupdate -k /etc/foreman-proxy/rndc.key
# nslookup aaa.example.com DNS_IP_Address
# echo -e "server DNS_IP_Address\n \
update delete aaa.example.com 3600 IN A Host_IP_Address\n \
send\n" | nsupdate -k /etc/foreman-proxy/rndc.key
```

4. **satellite-installer** コマンドを入力して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/dns.yml** ファイルに加えます。

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

5. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
6. Capsule Server を見つけて、**Actions** 列のリストから **Refresh** を選択します。

7. DNS 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。

4.2. CAPSULE SERVER での外部 DHCP の設定

外部の DHCP で Capsule Server を設定するには、以下の手順を実行します。

1. 「Capsule Server を使用するための外部 DHCP サーバーの設定」
2. 「外部 DHCP サーバーを使用した Satellite Server の設定」

4.2.1. Capsule Server を使用するための外部 DHCP サーバーの設定

Red Hat Enterprise Linux を実行する外部の DHCP サーバーを Capsule Server で使用できるように設定するには、ISC DHCP Service と Berkeley Internet Name Domain (BIND) ユーティリティーパッケージをインストールする必要があります。また、DHCP 設定とリースファイルを Capsule Server と共有する必要があります。この手順の例では、分散型の Network File System (NFS) プロトコルを使用して DHCP 設定とリースファイルを共有します。

注記

外部の DHCP サーバーとして dnsmasq を使用する場合には、**dhcp-no-override** の設定を有効にします。Satellite は **grub2/** サブディレクトリーの配下にある TFTP サーバーに設定ファイルを作成するので、この設定を必ず有効にしてください。**dhcp-no-override** 設定が無効な場合には、クライアントは root ディからブートローダーと設定をフェッチするのでエラーが発生する可能性があります。

手順

1. Red Hat Enterprise Linux ホストに、ISC DHCP Service と Berkeley Internet Name Domain (BIND) ユーティリティーパッケージをインストールします。

```
# dnf install dhcp-server bind-utils
```

2. 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
```

上記のコマンドを実行すると、2 つのファイルで設定されるキーペアが現在のディレクトリーに作成されます。

3. キーからシークレットハッシュをコピーします。

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

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

```
# 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 "jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

option routers の値は、外部の DHCP サービスと使用する Satellite または Capsule IP アドレスに置き換える点に注意してください。

5. キーファイルが作成されたディレクトリーから、2 つのキーファイルを削除します。

6. Satellite Server で各サブネットを定義します。定義済みのサブネットに DHCP Capsule は設定しないでください。
競合を回避するには、リースと予約範囲を別に設定します。たとえば、リース範囲を 192.168.38.10 から 192.168.38.100 に設定した場合には、Satellite Web UI で予約範囲を 192.168.38.101 から 192.168.38.250 に設定します。

7. DHCP サーバーに外部アクセスできるように、ファイアウォールを設定します。

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

8. Satellite Server で **foreman** ユーザーの UID と GID を指定します。

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

9. DHCP サーバーで、1 つ前の手順で定義した ID と同じ **foreman** ユーザーとグループを作成します。

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

10. 設定ファイルにアクセスできるように、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 サービスを有効にして開始します。

```
# systemctl enable --now dhcpd
```

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

```
# dnf install nfs-utils
# systemctl enable --now nfs-server
```

13. NFS を使用してエクスポートする DHCP 設定ファイルとリースファイルのディレクトリーを作成します。

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

14. 作成したディレクトリーにマウントポイントを作成するには、以下の行を **/etc/fstab** ファイルに追加します。

```
/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/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)
```

入力する IP アドレスは、外部 DHCP サービスで使用する Satellite または Capsule IP アドレスを指定する点に注意してください。

17. NFS サーバーをリロードします。

```
# exportfs -rva
```

18. ファイアウォールで DHCP omapi ポート 7911 を設定します。

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

19. オプション: NFS に外部からアクセスできるようにファイアウォールを設定します。クライアントは NFSv3 を使用して設定します。

```
# 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
```

4.2.2. 外部 DHCP サーバーを使用した Satellite Server の設定

外部 DHCP サーバーを使用して Capsule Server を設定できます。

前提条件

- 外部の DHCP サーバーを設定し、Capsule Server と DHCP 設定ファイルとリースファイルを共有していることを確認する。詳細は、「Capsule Server を使用するための外部 DHCP サーバーの設定」 を参照してください。

手順

1. **nfs-utils** パッケージをインストールします。

```
# dnf install nfs-utils
```

2. 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
```

3. ファイルの所有者を変更します。

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

4. NFS サーバーとの通信とリモートプロシージャコール (RPC: Remote Procedure Call) 通信パスを検証します。

```
# showmount -e DHCP_Server_FQDN
# rpcinfo -p DHCP_Server_FQDN
```

5. **/etc/fstab** ファイルに以下の行を追加します。

```
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

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
```

6. **/etc/fstab** でファイルシステムをマウントします。

```
# mount -a
```

7. **foreman-proxy** ユーザーがネットワークで共有したファイルにアクセスできることを確認するには、DHCP 設定ファイルとリースファイルを表示します。

```
# 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** コマンドを入力して、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=DHCP_Server_FQDN
```

9. 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。

4.3. CAPSULE SERVER での外部 TFTP の設定

外部 TFTP サービスを使用して Capsule Server を設定できます。

手順

1. NFS 用に TFTP ディレクトリーを作成します。

```
# mkdir -p /mnt/nfs/var/lib/tftpboot
```

2. **/etc/fstab** ファイルで以下の行を追加します。

```
TFTP_Server_IP_Address:/exports/var/lib/tftpboot /mnt/nfs/var/lib/tftpboot nfs
rw,vers=3,auto,nosharecache,context="system_u:object_r:tftpdir_rw_t:s0" 0 0
```

3. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

4. **satellite-installer** コマンドを入力して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/tftp.yml** ファイルに加えます。

```
# satellite-installer --foreman-proxy-tftp=true \
--foreman-proxy-tftp-root /mnt/nfs/var/lib/tftpboot
```

5. DHCP サービスとは異なるサーバーで TFTP サービスを実行している場合は、TFTP サービスを実行するサーバーの FQDN または IP アドレスに、**tftp_servername** 設定を更新します。

```
# satellite-installer --foreman-proxy-tftp-servername=TFTP_Server_FQDN
```

6. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
7. Capsule Server を見つけて、**Actions** 列のリストから **Refresh** を選択します。
8. TFTP 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 IDM DNS を使用した CAPSULE SERVER の設定

Satellite Server がホストの DNS レコードを追加する時には、まずどの Capsule が対象のドメインに DNS を提供しているかを判断します。次に、デプロイメントに使用する DNS サービスを提供するように設定された Capsule と通信し、レコードを追加します。ホストはこのプロセスには関与しません。そのため、IdM サーバーを使用して管理するドメインに DNS サービスを提供するように設定された Satellite または Capsule に IdM クライアントをインストールし、設定する必要があります。

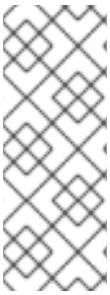
Capsule Server は、 Red Hat Identity Management (IdM) サーバーを使用して DNS サービスを提供するように設定できます。Red Hat Identity Management の詳細は、Linux Domain Identity, Authentication, and Policy Guide を参照してください。

Red Hat Identity Management (IdM) サーバーを使用して DNS サービスを提供するように Capsule Server を設定するには、以下の手順のいずれかを使用します。

- 「GSS-TSIG 認証を使用した動的 DNS 更新の設定」
- 「TSIG 認証を使用した動的 DNS 更新の設定」

内部 DNS サービスに戻すには、次の手順を使用します。

- 「内部 DNS サービス使用への復元」



注記

DNS の管理に、Capsule Server を使用する必要はありません。Satellite のレルム登録機能を使用しており、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。外部の IdM DNS とレルム登録を同時に使用して、Capsule Server を設定することはできません。レルム登録の設定の詳細は、**オンラインネットワーク環境への Satellite Server のインストール** 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。

4.4.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

RFC3645 で定義されている秘密鍵トランザクション (GSS-TSIG) 技術の一般的なセキュリティーサービスアルゴリズムを使用するように IdM サーバーを設定できます。IdM サーバーが GSS-TSIG 技術を使用するように設定するには、Capsule Server のベースオペレーティングシステムに IdM クライアントをインストールする必要があります。

前提条件

- 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。詳細は、**Identity Management のインストールガイド** の IdM のポート要件 を参照してください。
- IdM サーバーの管理者に問い合わせて、IdM サーバーでゾーンを作成するパーミッションが割り当てられた、IdM サーバーのアカウントを取得する。
- 応答ファイルのバックアップを作成する必要があります。応答ファイルが破損した場合に、元の状態に戻せるように、バックアップを使用できます。詳細は、Satellite Server の設定 を参照してください。

手順

GSS-TSIG 認証で動的 DNS 更新を設定するには、以下の手順を実行します。

IdM サーバーでの Kerberos プリンシパルの作成

1. IdM 管理者から取得したアカウントの Kerberos チケットを取得します。

```
# kinit idm_user
```

2. IdM サーバーでの認証に使用する Capsule Server 用の新規 Kerberos プリンシパルを作成します。

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

IdM クライアントのインストールおよび設定

1. デプロイメントの DNS サービスを管理する Satellite または Capsule のベースオペレーティングシステムで **ipa-client** パッケージをインストールします。

```
# satellite-maintain packages 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 チケットを取得します。

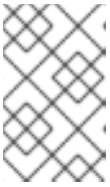
```
# kinit admin
```

4. 既存の **keytab** を削除します。

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

5. このシステムの **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. **dns.keytab** ファイルのグループと所有者を **foreman-proxy** に設定します。

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

7. オプション: **keytab** ファイルが有効であることを確認するには、以下のコマンドを入力します。

```
# 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** > **DNS** > **DNS Zones** に移動します。

   b. **Add** を選択し、ゾーン名を入力します。(例: **example.com**)

c. **Add and Edit** をクリックします。

d. 設定タブをクリックして **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** を **True** に設定します。

f. **Allow PTR sync** を有効にします。

g. **Save** をクリックして、変更を保存します。

2. 逆引きゾーンを作成して設定します。

a. **Network Services** > **DNS** > **DNS Zones** に移動します。

b. **Add** をクリックします。

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** を **True** に設定します。

g. **Save** をクリックして、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

1. **satellite-installer** コマンドを使用して、ドメインの DNS サービスを管理するように Satellite または Capsule を設定します。

- Satellite で以下のコマンドを入力します。

```
satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--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
```

- Capsule で、以下のコマンドを実行します。

```
satellite-installer --scenario capsule \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--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
```

satellite-installer コマンドを実行して Capsule 設定に変更を加えた後に、Satellite Web UI で変更のある Capsule ごとに設定を更新する必要があります。

Satellite Web UI での設定更新

1. Satellite Web UI で、**Infrastructure** > **Capsules** に移動し、Capsule Server を見つけて、**Actions** 列のリストから **Refresh** を選択します。
2. ドメインを設定します。
   a. Satellite Web UI で、**Infrastructure** > **Domains** 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 に設定されていることを確認します。
3. サブネットを設定します。
   a. Satellite Web UI で、**Infrastructure** > **Subnets** に移動し、サブネット名を選択します。
   b. **Subnet** タブで、**IPAM** を **None** に設定します。
   c. **Domains** タブで、IdM サーバーを使用して管理するドメインを選択します。
   d. **Capsules** タブで、**Reverse DNS Capsule** が、サブネットが接続されている Capsule に設定されていることを確認します。
   e. **Submit** をクリックして変更を保存します。

4.4.2. TSIG 認証を使用した動的 DNS 更新の設定

IdM サーバーが DNS (TSIG) テクノロジーの秘密鍵トランザクション認証を使用するように設定できます。このテクノロジーは、認証に **rndc.key** キーファイルを使用します。TSIG プロトコルについては RFC2845 に定義されています。

前提条件

- 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。詳細は **Linux Domain Identity, Authentication, and Policy Guide**の Port Requirements を参照してください。
- IdM サーバーで **root** 権限を取得する必要があります。
- デプロイメントに DNS サービスを提供するように Satellite Server または Capsule Server が設定されていることを確認する。
- デプロイメントの DNS サービスを管理する Satellite または Capsule のいずれかのベースオペレーティングシステムで DNS、DHCP および TFTP サービスを設定する必要がある。
- 応答ファイルのバックアップを作成しておく。応答ファイルが破損した場合に、元の状態に戻せるように、バックアップを使用できます。詳細は、Satellite Server の設定 を参照してください。

手順

TSIG 認証で動的 DNS 更新を設定するには、以下の手順を実行します。

IdM サーバーの DNS ゾーンに対する外部更新の有効化

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。

```
############################################################################

include "/etc/rndc.key";
controls  {
inet _IdM_Server_IP_Address_ port 953 allow { _Satellite_IP_Address_; } keys { "rndc-key";
};
};
############################################################################
```

2. **named** サービスをリロードして、変更を有効にします。

```
# systemctl reload named
```

3. IdM Web UI で、**Network Services** > **DNS** > **DNS Zones** に移動して、ゾーンの名前をクリックします。**Settings** タブで、以下の変更を適用します。

   a. **BIND update policy** ボックスで以下の内容を追加します。

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

   b. **Dynamic update** を **True** に設定します。

   c. **Update** をクリックして変更を保存します。

4. IdM サーバーから Satellite Server のベースオペレーティングシステムに **/etc/rndc.key** ファイルをコピーします。以下のコマンドを入力します。

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

5. **rndc.key** ファイルに適切な所有者、パーミッション、SELinux コンテキストを設定するには、以下のコマンドを入力します。

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

6. **foreman-proxy** ユーザーは、手動で **named** グループに割り当てます。通常、satellite-installer は **foreman-proxy** ユーザーが **named** UNIX グループに所属させますが、今回のシナリオでは、Satellite でユーザーとグループを管理していないので、**foreman-proxy** ユーザーを **named** グループに手作業で割り当てる必要があります。

```
# usermod -a -G named foreman-proxy
```

7. Satellite Server で以下の **satellite-installer** コマンドを入力して、Satellite が外部の DNS サーバーを使用するように設定します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="IdM_Server_IP_Address" \
--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更新のテスト

1. 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キーファイルと同じであることを確認します。

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

2. Satellite Server で、ホストのテスト DNS エントリーを作成します。(例: **192.168.25.1** の IdM サーバーに、**192.168.25.20** の A レコードを指定した **test.example.com** ホストなど)

```
# 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
```

3. 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
```

4. IdM Web UI でエントリーを参照するために、**Network Services** > **DNS** > **DNS Zones** に移動します。ゾーンの名前をクリックし、名前でホストを検索します。

5. 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
```

6. 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** エラーメッセージを返します。

4.4.3. 内部 DNS サービス使用への復元

Satellite Server および Capsule Server を DNS プロバイダーとして使用するように戻すことができま

す。外部の DNS を設定する前に作成した応答ファイルのバックアップを使用するか、応答ファイルのバックアップを作成します。アンサーファイルに関する詳細は、Satellite Server の設定 を参照してください。

手順

ドメインの DNS サーバーを管理するように設定する Satellite または Capsule Server で、以下の手順を実行します。

DNS サーバーとしての Satellite または Capsule の設定

- 外部の DNS を設定する前に応答ファイルのバックアップを作成済みの場合には、応答ファイルを復元して、**satellite-installer** コマンドを入力します。

```
# satellite-installer
```

- 応答ファイルの適切なバックアップがない場合には、ここで応答ファイルのバックアップを作成します。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するには、Satellite と Capsule で、以下の **satellite-installer** コマンドを入力します。

```
# satellite-installer \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1"
```

詳細は、Capsule Server での DNS、DHCP、および TFTP の設定 を参照してください。

satellite-installer コマンドを実行して Capsule 設定に変更を加えた後に、Satellite Web UI で変更のある Capsule ごとに設定を更新する必要があります。

Satellite Web UI での設定更新

1. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
2. 更新する各 Capsule で、**Actions** リストから **Refresh** を選択します。
3. ドメインを設定します。
    a. Satellite Web UI で、**Infrastructure** > **Domains** に移動し、設定するドメイン名をクリックします。
    b. **ドメイン** タブで、**DNS Capsule** を、サブネットの接続先の Capsule に設定します。
4. サブネットを設定します。
    a. Satellite Web UI で、**Infrastructure** > **Subnets** に移動し、サブネット名を選択します。
    b. **Subnet** タブで、**IPAM** を **DHCP** または **Internal DB** に設定します。
    c. **Domains** タブで、Satellite または Capsule で管理するドメインを選択します。
    d. **Capsules** タブで、**Reverse DNS Capsule** を、サブネットの接続先の Capsule に設定します。
    e. **Submit** をクリックして変更を保存します。

第5章 CAPSULE を使用した DHCP の管理

Capsule を使用して、Satellite を DHCP サービスと統合できます。Capsule には複数の DHCP プロバイダーがあり、Satellite を既存の DHCP インフラストラクチャーと統合したり、新しいインフラストラクチャーをデプロイしたりするために使用できます。Capsule の DHCP モジュールを使用して、利用可能な IP アドレスをクエリーし、新しい予約を追加し、既存の予約を削除できます。Capsule はサブネット宣言を管理できないことに注意してください。

利用可能な DHCP プロバイダー

- **dhcp_infoblox** - 詳細については、**ホストのプロビジョニング** の DHCP および DNS プロバイダーとしての infoblox の使用 を参照してください。
- **dhcp_isc** - OMAPI 上の ISC DHCP サーバー。詳細は、 **Capsule Server のインストール** の Capsule Server での DNS、DHCP、および TFTP の設定 を参照してください。
- **dhcp_remote_isc** - ネットワーク経由でリースがマウントされた OMAPI 上の ISC DHCP サーバー。詳細は、**Capsule Server のインストール** の Capsule Server で使用する外部 DHCP サーバーの設定 を参照してください。

5.1. DHCPD API の保護

Capsule は、dhcpd API を使用して DHCP デーモンと対話し、DHCP を管理します。デフォルトでは、dhcpd API はアクセス制御なしで任意のホストをリッスンします。**omapi_key** を追加して、基本的なセキュリティーを提供できます。

手順

1. 必要なパッケージをインストールします。

```
# dnf install bind-utils
```

2. キーを生成します。

```
# dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 512 -n HOST omapi_key
# cat Komapi_key.+*.private | grep ^Key|cut -d ' ' -f2-
```

3. **satellite-installer** を使用して dhcpd API を保護します。

```
# satellite-installer \
--foreman-proxy-dhcp-key-name "My_Name" \
--foreman-proxy-dhcp-key-secret "My_Secret"
```

第6章 CAPSULE を使用した DNS の管理

Satellite は Capsule を使用して DNS レコードを管理できます。DNS 管理には、既存の DNS ゾーンからの DNS レコードの更新および削除が含まれます。Capsule には複数の DNS プロバイダーがあり、Satellite を既存の DNS インフラストラクチャーと統合したり、新しいものをデプロイしたりするために使用できます。

DNS を有効にすると、Capsule は **dns_nsupdate** プロバイダーを使用して RFC 2136 に準拠する任意の DNS サーバーを操作できます。他のプロバイダーは、Infoblox の **dns_infoblox** など、より直接的な統合を提供します。

利用可能な DNS プロバイダー

- **dns_infoblox** - 詳細については、**ホストのプロビジョニング** の DHCP および DNS プロバイダーとしての infoblox の使用 を参照してください。
- **dns_nsupdate** - nsupdate を使用した動的 DNS 更新.詳細は、**ホストのプロビジョニング** で DHCP および DNS プロバイダーとしての infoblox の使用 を参照してください。
- **dns_nsupdate_gss** - GSS-TSIG による動的 DNS 更新。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。

付録A CAPSULE SERVER のスケーラビリティーに関する考慮事項

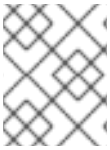
Satellite Server がサポート可能な Capsule Server の最大数には上限がありません。ただし、スケーラビリティーは非常に柔軟です (特に Puppet クライアントを管理する場合)。

Puppet クライアントを管理するときの Capsule Server のスケーラビリティーは、CPU の数、実行間隔の分散、および Puppet 管理リソースの数によって異なります。Capsule Server では、ある時点で同時実行される Puppet エージェントの上限数が 100 となっています。100 を超える Puppet エージェントを同時実行すると、503 HTTP エラーが発生します。

たとえば、実行が終了してから、次の実行が開始されるまでの任意のタイミングで、同時に実行される Puppet エージェントが 100 台未満で、Puppet エージェントの実行が均等に分散されていると仮定した場合に、CPU が 4 つ割り当てられた Capsule Server は最大で 1250-1600 台の Puppet クライアントに対応し、各 Puppet クライントに、中程度のワークロードである 10 個の Puppet クラスが割り当てられます。必要な Puppet クライアントの数により、Satellite のインストールは、Capsule Server の数をスケールアウトしてサポートします。

Puppet クライアントの管理時に Capsule Server をスケーリングする場合は、以下のことを前提とします。

- 外部 Puppet クライアントには、Satellite 統合 Capsule に直接報告するものはありません。
- 他のすべての Puppet クライアントは外部 Capsule に直接報告します。
- すべての Puppet エージェンの実行間隔が均等に分散されています。



注記

均等に分散されないと、Satellite Server をオーバーロードするリスクが高くなります。100 の同時要求の制限が適用されます。

以下の表は、推奨の 4 CPU を使用した場合のスケーラビリティーの制限を示しています。

表A.1 4 CPU を使用した場合の Puppet のスケーラビリティー

1つのホストあたりの Puppet 管理リソース数	実行間隔の分散
1	3000 - 2500
10	2400 - 2000
20	1700 - 1400

以下の表は、最小 2 CPU を使用した場合のスケーラビリティーの制限を示しています。

表A.2 2 CPU を使用した場合の Puppet のスケーラビリティー

1つのホストあたりの Puppet 管理リソース数	実行間隔の分散
1	1700 - 1450

1 つのホストあたりの Puppet 管理リソース数	実行間隔の分散
10	1500 - 1250
20	850 - 700