SUSE Manager 4.3

クライアント設定ガイド

Contents

クライアント設定ガイドの概要

更新: 2025-12-12

クライアントの登録は、SUSE Managerインストール後の最初の手順であり、SUSE Managerで費やす時間のほとんどは、これらのクライアントを管理する時間です。

SUSE　Managerは、広範なクライアント技術と互換性があります。広範なハードウェアオプションを使用して、従来のクライアントまたはSaltクライアントをインストールし、SUSE　Linux　Enterpriseまたはその他のLinuxオペレーティングシステムを実行できます。

サポートされているクライアントおよび機能の一覧については、**Client-configuration　›　Supported-features**を参照してください。

このガイドでは、異なるクライアントを登録して設定する方法に関して手動の方法と自動の方法の両方について説明します。

このバージョンのSUSE　Managerは、Saltおよび従来のクライアント(一部のオペレーティングシステム)と互換性があります。 SUSEは、次のSUSE Manager 4.3リリースで従来のクライアントを廃止する予定です。

SUSE　Manager　4.3以降のリリースでは、従来のクライアントはサポートされません(2023 年にリリース予定)。

すべての新しい配備でSaltクライアントを排他的に使用し、既存の従来のクライアントをSaltに移行することをお勧めします。

Chapter 1. サポートされているクライアントと機能

SUSE Managerは、さまざまなクライアント技術と互換性があります。 広範なハードウェアオプションを使用して、従来のクライアントまたはSaltクライアントをインストールし、SUSE Linux Enterpriseまたはその他のLinuxオペレーティングシステムを実行できます。

このセクションには、サポートされているクライアントシステムのまとめが含まれています。それぞれのクライアントで使用できる機能の詳細な一覧については、次のページを参照してください。

このバージョンのSUSE Managerは、Saltおよび従来のクライアント(一部のオペレーティングシステム)と互換性があります。 SUSEは、次のSUSE Manager 4.3リリースで従来のクライアントを廃止する予定です。

SUSE Manager 4.3以降のリリースでは、従来のクライアントはサポートされません(2023 年にリリース予定)。

すべての新しい配備でSaltクライアントを排他的に使用し、既存の従来のクライアントをSaltに移行することをお勧めします。

1.1. サポートされているクライアントシステム

クライアントオペレーティングシステムは、そのオペレーティングシステムを提供する組織によってサポートされています。 バージョンおよびSPレベルは、SUSE Managerでサポートされる全般的なサポート(通常またはLTSS)の条件を基準にする必要があります。 サポートされている製品バージョンの詳細については、https://www.suse.com/lifecycleを参照してください。

サポートされているクライアントオペレーティングシステムを次の表に示します。 この表のアイコンの意味は次のとおりです。

- ✔このオペレーティングシステムを実行しているクライアントはSUSEでサポートされています
- ✖このオペレーティングシステムを実行しているクライアントはSUSEではサポートされていません
- ?クライアントは検討中であり、後日利用可能になるかどうかは未定です。

クライアントで実行されているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。

表 1. サポートされているクライアントシステム

Operating System	Architecture	Traditional Clients	Salt Clients
SUSE Linux Enterprise 15, 12	x86-64, ppc64le, IBM Z, aarch64	✔	✔
SUSE Linux Enterprise Server for SAP 15, 12	x86-64, ppc64le	✔	✔

Operating System	Architecture	Traditional Clients	Salt Clients
SLE Micro	x86-64, aarch64, s390x	✗	✓
SL Micro 6.1, 6.0	x86-64, aarch64, s390x	✗	✓
openSUSE Leap 15	x86-64, aarch64	✓	✓
SUSE Liberty Linux 9, 8, 7	x86-64	✗	✓
SUSE Linux Enterprise Server ES 8, 7	x86-64	✗	✓
AlmaLinux 9, 8	x86-64, aarch64	✗	✓
Amazon Linux 2	x86-64, aarch64	✗	✓
CentOS 7	x86-64, aarch64	✓	✓
Debian 12	x86-64	✗	✓
Open Enterprise Server 24.4, 23.4	x86-64	✓	✓
Oracle Linux 9, 8, 7	x86-64, aarch64	✗	✓
Red Hat Enterprise Linux 9, 8, 7	x86-64	✗	✓
Rocky Linux 9, 8	x86-64, aarch64	✗	✓
Ubuntu 24.04, 22.04	amd64	✗	✓

DebianとUbuntuは、x86-64アーキテクチャをamd64としてリストします。

配布がサポート終了になると、サポートが廃止されたと見なされる3か月の猶予期間に入ります。　その期間が過ぎると、製品はサポート対象外と見なされます。　サポートは、努力ベースでのみ提供される場合があります。

サポート終了日の詳細については、https://endoflife.software/operating-systemsを参照してください。

1.2. サポートされているツールパッケージ

spacewalk-utilsパッケージおよび**spacewalk-utils-extras**パッケージは、追加のサービスおよび機能を提供できます。

SUSE　Manager**spacewalk-utils**パッケージはSUSEでは完全にサポートされていて、次のツールが含まれています。

表 2. Spacewalkのユーティリティ

ツール名	説明	サポートの有無
spacewalk-common-channels	SUSE Customer Center で提供されないチャンネルを追加します	✔
spacewalk-hostname-rename	SUSE Managerサーバのホスト名を変更します	✔
spacewalk-clone-by-date	特定の日までにチャンネルを複製します	✔
spacewalk-sync-setup	ISSマスタおよびスレーブの組織マッピングを設定します	✔
spacewalk-manage-channel-lifecycle	チャンネルのライフサイクルを管理します	✔

SUSE Manager **spacewalk-utils-extras**パッケージはSUSEでは限定的にサポートされています。

1.3. サポートされているSUSEおよびopenSUSEクライアントの機能

この表には、SUSEおよびopenSUSEクライアントのさまざまな機能の使用可否がリストされています。 この表では、SLES、SLED、SUSE Linux Enterprise Server for SAP、SUSE Linux Enterprise Server for HPCなど、SUSE Linux Enterpriseオペレーティングシステムのすべての亜種について記載しています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 SUSE Linux EnterpriseはSUSEでサポートされています。 openSUSEはSUSEコミュニティでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✘: 機能は使用できません
- ?: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 3. SUSEおよびopenSUSEオペレーティングシステムでサポートされている機能

機能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
クライアント	✔	✔	✔
システムパッケージ	SUSE	SUSE	openSUSEコミュニティ

機能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
登録	✔	✔	Salt
パッケージのインストール	✔	✔	Salt
パッチの適用	✔	✔	Salt
リモートコマンド	✔	✔	Salt
システムパッケージの状態	Salt	Salt	Salt
システムカスタムの状態	Salt	Salt	Salt
グループカスタムの状態	Salt	Salt	Salt
組織カスタムの状態	Salt	Salt	Salt
システムセットマネージャ(SSM)	✔	✔	Salt
製品移行	✔	✔	Salt
基本的な仮想ゲスト管理✱	✔	✔	Salt
高度な仮想ゲスト管理✱	Salt	Salt	Salt
仮想ゲストインストール(AutoYaST)、ホストOSとして	Traditional	Traditional	✘
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	Salt	Salt	Salt
仮想ゲスト管理	Salt	Salt	Salt
システムの配備(PXE/AutoYaST)	✔	✔	✔
システムの再配備(AutoYaST)	✔	✔	Salt
接続メソッド	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	✔	✔	Salt

機能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
動作チェーン	✔	✔	Salt
ステージング(パッケージの事前ダウンロード)	✔	✔	Salt
重複パッケージの報告	✔	✔	Salt
CVE監査	✔	✔	Salt
SCAP監査	✔	✔	Salt
パッケージの確認	Traditional	Traditional	✖
パッケージのロック	Salt	Salt	Salt
システムのロック	Traditional	Traditional	✖
メンテナンスウィンドウ	✔	✔	✔
システムのスナップショット	Traditional	Traditional	✖
設定ファイルの管理	✔	✔	Salt
パッケージプロファイル	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✔	✔	✔
モニタリングサーバ	Salt	Salt	Salt
監視対象クライアント	Salt	Salt	Salt
Docker buildhost	Salt	Salt	?
OSでのDockerイメージの構築	Salt	Salt	Salt
Kiwi buildhost	Salt	?	✖
OSでのKiwiイメージの構築	Salt	?	✖
繰り返しアクション	Salt	Salt	Salt
AppStream	なし	なし	なし
Yomi	✖	✔	✔

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.4. サポートされているSLE Microクライアント機能

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。SLE MicroはSUSEでサポートされています。

現時点では、SLE Microは正規のミニオンとして(**デフォルト**接続メソッド)のみサポートされています。Salt SSHクライアント(**salt-ssh**接続メソッド)として管理できるように目指しています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 4. SLE Microオペレーティングシステムでサポートされている機能

機能	SLE Micro
クライアント	Salt
オペレーティングシステムパッケージ	Salt
登録	Salt
パッケージのインストール	Salt
パッチの適用(CVE IDが必要)	Salt
リモートコマンド	Salt
システムパッケージの状態	Salt
システムカスタムの状態	Salt
グループカスタムの状態	Salt

機能	SLE Micro
組織カスタムの状態	Salt
システムセットマネージャ(SSM)	Salt
製品移行	Salt
基本的な仮想ゲスト管理✱	?
高度な仮想ゲスト管理✱	?
仮想ゲストインストール(キックスタート)、ホストOSとして	✖
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	?
システムの配備(PXE/キックスタート)	?
システムの再配備(キックスタート)	✖
接続メソッド	Salt: ZeroMQ
SUSE Managerプロキシでの操作	Salt
動作チェーン	Salt
ステージング(パッケージの事前ダウンロード)	?
重複パッケージの報告	Salt
CVE監査(CVE IDが必要)	Salt
SCAP監査	?
パッケージの確認	?
パッケージのロック	Salt
システムのロック	?
メンテナンスウィンドウ	?
システムのスナップショット	✖
設定ファイルの管理	Salt
スナップショットとプロファイル	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	Salt
モニタリングサーバ	✖
監視対象クライアント✱✱	Salt

機能	SLE Micro
Docker buildhost	✗
OSでのDockerイメージの構築	✗
Kiwi buildhost	✗
OSでのKiwiイメージの構築	Salt
繰り返しアクション	Salt
AppStream	なし
Yomi	?

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

✱✱ SLE Microでは、Node exporterとBlackbox exporterのみが利用できます。

1.5. サポートされているSL Microクライアント機能

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。SL MicroはSUSEでサポートされています。

SL Microは、**デフォルト**接続メソッドのみのSaltクライアントとして現在サポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✗: 機能は使用できません
- ?: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 5. SLE Microオペレーティングシステムでサポートされている機能

機能	SLE Micro
クライアント	Salt
オペレーティングシステムパッケージ	Salt
登録	Salt
パッケージのインストール	Salt
パッチの適用(CVE IDが必要)	Salt
リモートコマンド	Salt
システムパッケージの状態	Salt
システムカスタムの状態	Salt
グループカスタムの状態	Salt
組織カスタムの状態	Salt
システムセットマネージャ(SSM)	Salt
製品移行	Salt
基本的な仮想ゲスト管理✱	?
高度な仮想ゲスト管理✱	?
仮想ゲストインストール(キックスタート)、ホストOSとして	✗
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	?
システムの配備(PXE/キックスタート)	?
システムの再配備(キックスタート)	✗
接続メソッド	Salt: ZeroMQ
SUSE Managerプロキシでの操作	Salt
動作チェーン	Salt
ステージング(パッケージの事前ダウンロード)	?
重複パッケージの報告	Salt
CVE監査(CVE IDが必要)	Salt
SCAP監査	?
パッケージの確認	?
パッケージのロック	Salt

機能	SLE Micro
システムのロック	?
メンテナンスウィンドウ	?
システムのスナップショット	✗
設定ファイルの管理	Salt
スナップショットとプロファイル	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	Salt
モニタリングサーバ	✗
監視対象クライアント✱✱	Salt
Docker buildhost	✗
OSでのDockerイメージの構築	✗
Kiwi buildhost	✗
OSでのKiwiイメージの構築	Salt
繰り返しアクション	Salt
AppStream	なし
Yomi	?

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

✱✱ SL Microでは、Node exporterとBlackbox exporterのみが利用できます。

1.6. サポートされているSUSE Liberty Linuxの機能

この表には、SUSE Liberty Linuxクライアントのさまざまな機能の使用可否がリストされています。

SUSE Liberty Linuxクライアントは、SUSE Linux Enterprise Server with Expanded Support (SLES ES)、Liberty、RES、またはRed Hat Expanded Supportとも呼ばれます。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 SUSE Liberty LinuxはSUSEでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 6. SUSE Liberty Linuxオペレーティングシステムでサポートされている機能

機能	SUSE Liberty Linux 7	SUSE Liberty Linux 8	SUSE Liberty Linux 9
クライアント	✔	Salt	Salt
システムパッケージ	SUSE	SUSE	SUSE
登録	✔	Salt	Salt
パッケージのインストール	✔	Salt	Salt
パッチの適用	✔	Salt	Salt
リモートコマンド	✔	Salt	Salt
システムパッケージの状態	Salt	Salt	Salt
システムカスタムの状態	Salt	Salt	Salt
グループカスタムの状態	Salt	Salt	Salt
組織カスタムの状態	Salt	Salt	Salt
システムセットマネージャ(SSM)	Salt	Salt	Salt
製品移行	✔✱	なし	なし
基本的な仮想ゲスト管理✱✱	✔	Salt	Salt
高度な仮想ゲスト管理✱✱	Salt	Salt	Salt

機能	SUSE Liberty Linux 7	SUSE Liberty Linux 8	SUSE Liberty Linux 9
仮想ゲストインストール(キックスタート)、ホストOSとして	Traditional	✗	✗
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	✓	Salt	Salt
システムの配備(PXE/キックスタート)	✓	Salt	Salt
システムの再配備(キックスタート)	✓	✗	✗
接続メソッド	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	✓	Salt	Salt
動作チェーン	✓	Salt	Salt
ステージング(パッケージの事前ダウンロード)	✓	Salt	Salt
重複パッケージの報告	✓	Salt	Salt
CVE監査	✓	Salt	Salt
SCAP監査	✓	Salt	Salt
パッケージの確認	Traditional	✗	✗
パッケージのロック	✓	?	?
システムのロック	Traditional	✗	✗
メンテナンスウィンドウ	✓	✓	Salt
システムのスナップショット	Traditional	Salt	Salt
設定ファイルの管理	✓	Salt	Salt
スナップショットとプロファイル	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✓	Salt	Salt

機能	SUSE Liberty Linux 7	SUSE Liberty Linux 8	SUSE Liberty Linux 9
モニタリングサーバ	✖	✖	✖
監視対象クライアント	Salt	Salt	Salt
Docker buildhost	✖	✖	✖
OSでのDockerイメージの構築	?	?	?
Kiwi buildhost	✖	✖	✖
OSでのKiwiイメージの構築	✖	✖	✖
定期的なアクション	Salt	Salt	Salt
AppStream	なし	✔	✔
Yomi	なし	なし	なし

*SUSE Liberty Linux LTSSに対応

**仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.7. サポートされているAlmaLinuxの機能

この表には、AlmaLinuxクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 AlmaLinuxはAlmaLinuxコミュニティでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ?: 機能は検討中であり、後日利用可能になるかどうかは未定です

- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 7. AlmaLinuxオペレーティングシステムでサポートされている機能

機能	AlmaLinux 9	AlmaLinux 8
クライアント	Salt (plain AlmaLinux)	Salt (plain AlmaLinux)
システムパッケージ	AlmaLinuxコミュニティ	AlmaLinuxコミュニティ
登録	Salt	Salt
パッケージのインストール	Salt	Salt
パッチの適用	Salt	Salt
リモートコマンド	Salt	Salt
システムパッケージの状態	Salt	Salt
システムカスタムの状態	Salt	Salt
グループカスタムの状態	Salt	Salt
組織カスタムの状態	Salt	Salt
システムセットマネージャ(SSM)	Salt	Salt
製品移行✱	Salt	Salt
基本的な仮想ゲスト管理✱✱	Salt	Salt
高度な仮想ゲスト管理✱✱	Salt	Salt
仮想ゲストインストール(キックスタート)、ホストOSとして	✖	✖
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	Salt	Salt
システムの配備(PXE/キックスタート)	Salt	Salt
システムの再配備(キックスタート)	Salt	Salt
接続メソッド	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	Salt	Salt
動作チェーン	Salt	Salt
ステージング(パッケージの事前ダウンロード)	Salt	Salt
重複パッケージの報告	Salt	Salt

機能	AlmaLinux 9	AlmaLinux 8
CVE監査	Salt	Salt
SCAP監査	Salt	Salt
パッケージの確認	✗	✗
パッケージのロック	✗	✗
システムのロック	✗	✗
メンテナンスウィンドウ	✓	✓
システムのスナップショット	✗	✗
設定ファイルの管理	Salt	Salt
スナップショットとプロファイル	Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	Salt	Salt
モニタリングサーバ	✗	✗
監視対象クライアント	Salt	Salt
Docker buildhost	✗	✗
OSでのDockerイメージの構築	✗	✗
Kiwi buildhost	✗	✗
OSでのKiwiイメージの構築	✗	✗
繰り返しアクション	Salt	Salt
AppStream	✓	✓
Yomi	なし	なし

✱ SUSE Liberty Linuxに対応

✱✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.8. サポートされているAmazon Linuxの機能

この表には、Amazon Linuxクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 Amazon LinuxはAmazonでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています

表 8. Amazon Linuxオペレーティングシステムでサポートされている機能

機能	Amazon Linux 2
クライアント	Salt
オペレーティングシステムパッケージ	Salt
登録	Salt
パッケージのインストール	Salt
パッチの適用(CVE IDが必要)	Salt
リモートコマンド	Salt
システムパッケージの状態	Salt
システムカスタムの状態	Salt
グループカスタムの状態	Salt
組織カスタムの状態	Salt
システムセットマネージャ(SSM)	Salt
製品移行	なし
基本的な仮想ゲスト管理✱	❓
高度な仮想ゲスト管理✱	❓
仮想ゲストインストール(キックスタート)、ホストOSとして	✖

機能	Amazon Linux 2
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	?
システムの配備(PXE/キックスタート)	?
システムの再配備(キックスタート)	?
接続メソッド	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	Salt
動作チェーン	Salt
ステージング(パッケージの事前ダウンロード)	Salt
重複パッケージの報告	Salt
CVE監査(CVE IDが必要)	Salt
SCAP監査	Salt
パッケージの確認	✗
パッケージのロック	✗
システムのロック	✗
メンテナンスウィンドウ	✓
システムのスナップショット	✗
設定ファイルの管理	Salt
スナップショットとプロファイル	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	?
モニタリングサーバ	✗
監視対象クライアント	Salt
Docker buildhost	Salt
OSでのDockerイメージの構築	Salt
Kiwi buildhost	Salt
OSでのKiwiイメージの構築	Salt
繰り返しアクション	Salt
AppStream	なし
Yomi	なし

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

✱従来のスタックはAmazon Linuxで利用できますが、サポートされていません。

1.9. サポートされているCentOSの機能

この表には、CentOSクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 CentOSはCentOSコミュニティでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ?: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 9. CentOSオペレーティングシステムでサポートされている機能

機能	CentOS 7
クライアント	✔ (plain CentOS)
システムパッケージ	CentOSコミュニティ
登録	✔
パッケージのインストール	✔
パッチの適用(CVE IDが必要)	✔ (エラータで必要なサードパーティサービス))
リモートコマンド	✔
システムパッケージの状態	Salt
システムカスタムの状態	Salt

機能	CentOS 7
グループカスタムの状態	Salt
組織カスタムの状態	Salt
システムセットマネージャ(SSM)	✔
製品移行✱	✔
基本的な仮想ゲスト管理✱✱	Salt
高度な仮想ゲスト管理✱✱	Salt
仮想ゲストインストール(キックスタート)、ホストOSとして	Traditional
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	✔
システムの配備(PXE/キックスタート)	✔
システムの再配備(キックスタート)	✔
接続メソッド	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	✔
動作チェーン	✔
ステージング(パッケージの事前ダウンロード)	✔
重複パッケージの報告	✔
CVE監査(CVE IDが必要)	✔
SCAP監査	✔
パッケージの確認	Traditional
パッケージのロック	✔
システムのロック	Traditional
メンテナンスウィンドウ	✔
システムのスナップショット	Traditional
設定ファイルの管理	✔
スナップショットとプロファイル	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✔
モニタリングサーバ	✖

機能	CentOS 7
監視対象クライアント	Salt
Docker buildhost	✗
OSでのDockerイメージの構築	✗
Kiwi buildhost	✗
OSでのKiwiイメージの構築	✗
繰り返しアクション	Salt
AppStream	なし
Yomi	なし

✱ SUSE Liberty Linuxに対応

✱✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.10. サポートされているDebianの機能

この表には、Debianクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。　DebianはDebianコミュニティでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✗: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能は Saltクライアントでのみサポートされています。

表 10. Debianオペレーティングシステムでサポートされている機能

機能	Debian 11	Debian 12
クライアント	✔	✔
システムパッケージ	Debian コミュニティ	Debian コミュニティ
登録	Salt	Salt
パッケージのインストール	Salt	Salt
パッチの適用	?	?
リモートコマンド	Salt	Salt
システムパッケージの状態	Salt	Salt
システムカスタムの状態	Salt	Salt
グループカスタムの状態	Salt	Salt
組織カスタムの状態	Salt	Salt
システムセットマネージャ(SSM)	Salt	Salt
製品移行	なし	なし
基本的な仮想ゲスト管理✱	Salt	Salt
高度な仮想ゲスト管理✱	Salt	Salt
仮想ゲストインストール(キックスタート)、ホストOSとして	✖	✖
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	Salt	Salt
システムの配備(PXE/キックスタート)	✖	✖
システムの再配備(キックスタート)	✖	✖
接続メソッド	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	Salt	Salt
動作チェーン	Salt	Salt
ステージング(パッケージの事前ダウンロード)	Salt	Salt
重複パッケージの報告	Salt	Salt
CVE監査	?	?
SCAP監査	?	?
パッケージの確認	✖	✖

機能	Debian 11	Debian 12
パッケージのロック	✔	✔
システムのロック	✖	✖
メンテナンスウィンドウ	✔	✔
システムのスナップショット	✖	✖
設定ファイルの管理	Salt	Salt
パッケージのプロファイル	Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✔	✔
モニタリングサーバ	✖	✖
モニタリングクライアント	Salt	Salt
Docker buildhost	?	?
OSでのDockerイメージの構築	Salt	Salt
Kiwi buildhost	✖	✖
OSでのKiwiイメージの構築	✖	✖
繰り返しアクション	Salt	Salt
AppStream	なし	なし
Yomi	なし	なし

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.11. サポートされているOpen Enterprise Serverの機能

この表には、Open Enterprise Serverクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステム

> を提供している組織によってサポートされています。 Open Enterprise ServerはMicro FocusまたはOpenTextでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 11. Open Enterprise Serverオペレーティングシステムでサポートされている機能

機能	Open Enterprise Server 24.4	Open Enterprise Server 23.4
クライアント	✔	✔
システムパッケージ	SUSE	SUSE
登録	✔	✔
パッケージのインストール	✔	✔
パッチの適用	✔	✔
リモートコマンド	✔	✔
システムパッケージの状態	Salt	Salt
システムカスタムの状態	Salt	Salt
グループカスタムの状態	Salt	Salt
組織カスタムの状態	Salt	Salt
システムセットマネージャ(SSM)	✔	✔
製品移行	✔	✔
基本的な仮想ゲスト管理✱	✔	✔
高度な仮想ゲスト管理✱	Salt	Salt
仮想ゲストインストール(AutoYaST)、ホストOSとして	Traditional	Traditional
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	Salt	Salt
仮想ゲスト管理	Salt	Salt

機能	Open Enterprise Server 24.4	Open Enterprise Server 23.4
システムの配備(PXE/AutoYaST)	✔	✔
システムの再配備(AutoYaST)	✔	✔
接続メソッド	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	✔	✔
動作チェーン	✔	✔
ステージング(パッケージの事前ダウンロード)	✔	✔
重複パッケージの報告	✔	✔
CVE監査	✔	✔
SCAP監査	✔	✔
パッケージの確認	Traditional	Traditional
パッケージのロック	Salt	Salt
システムのロック	Traditional	Traditional
メンテナンスウィンドウ	✔	✔
システムのスナップショット	Traditional	Traditional
設定ファイルの管理	✔	✔
パッケージプロファイル	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✔	✔
モニタリングサーバ	Salt	Salt
監視対象クライアント	Salt	Salt
Docker buildhost	Salt	Salt
OSでのDockerイメージの構築	Salt	Salt
Kiwi buildhost	?	?
OSでのKiwiイメージの構築	?	?
繰り返しアクション	Salt	Salt
AppStream	なし	なし

機能	Open Enterprise Server 24.4	Open Enterprise Server 23.4
Yomi	✔	✔

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.12. サポートされているOracleの機能

この表には、Oracle Linuxクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 Oracle LinuxはOracleでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✘: 機能は使用できません
- ?: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています

表 12. Oracle Linuxオペレーティングシステムでサポートされている機能

機能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
クライアント	✔	Salt	Salt
オペレーティングシステムパッケージ	✔	Salt	Salt
登録	✔	Salt	Salt
パッケージのインストール	✔	Salt	Salt
パッチの適用(CVE IDが必要)	✔	Salt	Salt

機能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
リモートコマンド	✔	Salt	Salt
システムパッケージの状態	Salt	Salt	Salt
システムカスタムの状態	Salt	Salt	Salt
グループカスタムの状態	Salt	Salt	Salt
組織カスタムの状態	Salt	Salt	Salt
システムセットマネージャ(SSM)	✔	Salt	Salt
製品移行✱	✔	Salt	Salt
高度な仮想ゲスト管理✱✱	Salt	Salt	Salt
基本的な仮想ゲスト管理✱✱	Salt	Salt	Salt
仮想ゲストインストール(キックスタート)、ホストOSとして	Traditional	✖	✖
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	✔	Salt	Salt
システムの配備(PXE/キックスタート)	✔	Salt	Salt
システムの再配備(キックスタート)	✔	Salt	Salt
接続メソッド	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	✔	Salt	Salt
動作チェーン	✔	Salt	Salt
ステージング(パッケージの事前ダウンロード)	✔	Salt	Salt
重複パッケージの報告	✔	Salt	Salt
CVE監査(CVE IDが必要)	✔	Salt	Salt

機能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
SCAP監査	✔	Salt	Salt
パッケージの確認	Traditional	✖	✖
パッケージのロック	✔	?	?
システムのロック	Traditional	✖	✖
メンテナンスウィンドウ	✔	✔	✔
システムのスナップショット	Traditional	✖	✖
設定ファイルの管理	✔	Salt	Salt
スナップショットとプロファイル	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✔	Salt	Salt
モニタリングサーバ	✖	✖	✖
監視対象クライアント	Salt	Salt	Salt
Docker buildhost	✖	✖	✖
OSでのDockerイメージの構築	✖	✖	✖
Kiwi buildhost	✖	✖	✖
OSでのKiwiイメージの構築	✖	✖	✖
繰り返しアクション	Salt	Salt	Salt
AppStream	なし	✔	✔
Yomi	なし	なし	なし

✱ SUSE Liberty Linuxに対応

✱✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイク

ルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.13. サポートされているRed Hat Enterprise Linuxの機能

この表には、Red Hat Enterprise Linuxクライアントのさまざまな機能の使用可否がリストされています(拡張サポートがない場合)。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 Red Hat Enterprise LinuxはRed Hatでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ?: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 13. Red Hat Enterprise Linuxオペレーティングシステムでサポートされている機能

機能	RHEL 7	RHEL 8	RHEL 9
クライアント	✔	Salt	Salt
システムパッケージ	Red Hat	Red Hat	Red Hat
登録	✔	Salt	Salt
パッケージのインストール	✔	Salt	Salt
パッチの適用	✔	Salt	Salt
リモートコマンド	✔	Salt	Salt
システムパッケージの状態	Salt	Salt	Salt
システムカスタムの状態	Salt	Salt	Salt
グループカスタムの状態	Salt	Salt	Salt
組織カスタムの状態	Salt	Salt	Salt

機能	RHEL 7	RHEL 8	RHEL 9
システムセットマネージャ(SSM)	Salt	Salt	Salt
製品移行	なし	なし	なし
基本的な仮想ゲスト管理✱	✔	Salt	Salt
高度な仮想ゲスト管理✱	Salt	Salt	Salt
仮想ゲストインストール(キックスタート)、ホストOSとして	Traditional	✖	✖
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	✔	Salt	Salt
システムの配備(PXE/キックスタート)	✔	Salt	Salt
システムの再配備(キックスタート)	✔	Salt	Salt
接続メソッド	Traditional: OSAD、RHNSD、SSH-push。Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	✔	Salt	Salt
動作チェーン	✔	Salt	Salt
ステージング(パッケージの事前ダウンロード)	✔	Salt	Salt
重複パッケージの報告	✔	Salt	Salt
CVE監査	✔	Salt	Salt
SCAP監査	✔	Salt	Salt
パッケージの確認	Traditional	✖	✖
パッケージのロック	✔	?	?
システムのロック	Traditional	✖	✖
メンテナンスウィンドウ	✔	✔	✔
システムのスナップショット	Traditional	✖	✖

機能	RHEL 7	RHEL 8	RHEL 9
設定ファイルの管理	✔	Salt	Salt
スナップショットとプロファイル	Traditional. Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✔	Salt	Salt
モニタリングサーバ	✖	✖	✖
監視対象クライアント	Salt	Salt	Salt
Docker buildhost	✖	✖	✖
OSでのDockerイメージの構築	?	?	?
Kiwi buildhost	✖	✖	✖
OSでのKiwiイメージの構築	✖	✖	✖
繰り返しアクション	Salt	Salt	Salt
AppStream	なし	✔	✔
Yomi	なし	なし	なし

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.14. サポートされているRocky Linuxの機能

この表には、Rocky Linuxクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 Rocky LinuxはRocky Linuxコミュニティでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 14. Rocky Linuxオペレーティングシステムでサポートされている機能

機能	Rocky Linux 8	Rocky Linux 9
クライアント	Salt (plain Rocky Linux)	Salt (plain Rocky Linux)
システムパッケージ	Rocky Linuxコミュニティ	Rocky Linuxコミュニティ
登録	Salt	Salt
パッケージのインストール	Salt	Salt
パッチの適用	Salt	Salt
リモートコマンド	Salt	Salt
システムパッケージの状態	Salt	Salt
システムカスタムの状態	Salt	Salt
グループカスタムの状態	Salt	Salt
組織カスタムの状態	Salt	Salt
システムセットマネージャ(SSM)	Salt	Salt
製品移行✱	Salt	Salt
基本的な仮想ゲスト管理✱✱	Salt	Salt
高度な仮想ゲスト管理✱✱	✔	✔
仮想ゲストインストール(キックスタート)、ホストOSとして	✖	✖
仮想ゲストインストール(イメージテンプレート)、ホストOSとして	Salt	Salt
システムの配備(PXE/キックスタート)	Salt	Salt
システムの再配備(キックスタート)	Salt	Salt
接続メソッド	Salt: ZeroMQ、Salt-SSH	Salt: ZeroMQ、Salt-SSH
SUSE Managerプロキシでの操作	Salt	Salt

機能	Rocky Linux 8	Rocky Linux 9
動作チェーン	Salt	Salt
ステージング(パッケージの事前ダウンロード)	Salt	Salt
重複パッケージの報告	Salt	Salt
CVE監査	Salt	Salt
SCAP監査	Salt	Salt
パッケージの確認	✗	✗
パッケージのロック	?	?
システムのロック	✗	✗
メンテナンスウィンドウ	✓	✓
システムのスナップショット	✗	✗
設定ファイルの管理	Salt	Salt
パッケージのプロファイル	Salt: プロファイルはサポートされていますが、同期はサポートされていません	Salt: プロファイルはサポートされていますが、同期はサポートされていません
電源管理	✓	✓
モニタリングサーバ	✗	✗
監視対象クライアント	Salt	Salt
Docker buildhost	✗	✗
OSでのDockerイメージの構築	✗	✗
Kiwi buildhost	✗	✗
OSでのKiwiイメージの構築	✗	✗
繰り返しアクション	Salt	Salt
AppStream	✓	✓
Yomi	なし	なし

✱ SUSE Liberty Linuxに対応

✱✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再

開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

1.15. サポートされているUbuntuの機能

この表には、Ubuntuクライアントのさまざまな機能の使用可否がリストされています。

クライアントで実行しているオペレーティングシステムは、オペレーティングシステムを提供している組織によってサポートされています。 UbuntuはCanonicalでサポートされています。

この表のアイコンの意味は次のとおりです。

- ✔: 機能はSaltクライアントと従来のクライアントの両方で使用できます
- ✖: 機能は使用できません
- ❓: 機能は検討中であり、後日利用可能になるかどうかは未定です
- Traditional: 機能は従来のクライアントでのみサポートされています
- Salt: 機能はSaltクライアントでのみサポートされています。

表 15. Ubuntuオペレーティングシステムでサポートされている機能

Feature	Ubuntu 22.04	Ubuntu 24.04
Client	✔	✔
System packages	Ubuntu Community	Ubuntu Community
Registration	Salt	Salt
Install packages	Salt	Salt
Apply patches	✔	✔
Remote commands	Salt	Salt
System package states	Salt	Salt
System custom states	Salt	Salt
Group custom states	Salt	Salt
Organization custom states	Salt	Salt
System set manager (SSM)	Salt	Salt
Product migration	N/A	N/A

Feature	Ubuntu 22.04	Ubuntu 24.04
Basic Virtual Guest Management ✱	Salt	Salt
Advanced Virtual Guest Management ✱	Salt	Salt
Virtual Guest Installation (Kickstart), as Host OS	✗	✗
Virtual Guest Installation (image template), as Host OS	Salt	Salt
System deployment (PXE/Kickstart)	✗	✗
System redeployment (Kickstart)	✗	✗
Contact methods	Salt: ZeroMQ, Salt-SSH	Salt: ZeroMQ, Salt-SSH
Works with SUSE Manager Proxy	Salt	Salt
Action chains	Salt	Salt
Staging (pre-download of packages)	Salt	Salt
Duplicate package reporting	Salt	Salt
CVE auditing	?	?
SCAP auditing	?	?
Package verification	✗	✗
Package locking	✓	✓
Maintenance Windows	✓	✓
System locking	✗	✗
System snapshot	✗	✗
Configuration file management	Salt	Salt
Package profiles	Salt: Profiles supported, Sync not supported	Salt: Profiles supported, Sync not supported
Power management	✓	✓
Monitoring server	✗	✗
Monitored clients	Salt	Salt
Docker buildhost	?	?

Feature	Ubuntu 22.04	Ubuntu 24.04
Build Docker image with OS	Salt	Salt
Kiwi buildhost	✖	✖
Build Kiwi image with OS	✖	✖
Recurring Actions	Salt	Salt
AppStreams	N/A	N/A
Yomi	N/A	N/A

✱仮想ゲスト管理:

この表では、仮想ゲスト管理は基本と高度に分割されています。

基本的な仮想ゲスト管理には、VMのリスト化、低速更新、VMのライフサイクルアクション(起動、停止、再開、一時停止)、およびVM vCPUとメモリの変更が含まれています。

高度な仮想ゲスト管理には、基本的な仮想ゲスト管理のすべての機能に加えて、高速更新、VMライフサイクルアクション(削除、リセット、電源オフ)、VMディスクの変更、ネットワークグラフィカル表示の変更、およびグラフィカル表示の設定が含まれています。

Chapter 2. 設定の基本

広範な操作を利用できるようにするためにクライアント登録のための環境を準備するには、SUSE Managerで複数の手順を実行する必要があります。

このセクションには、SUSE Managerを正しくインストールおよびセットアップした後の 環境操作をサポートするために必要な初期設定手順のまとめが記載されています。

- SUSE Managerのインストールの詳細については、**Installation-and-upgrade › Install-server-unified**を参照してください。
- SUSE Managerのセットアップの詳細については、**Installation-and-upgrade › Server-setup**を参照してください。

2.1. ソフトウェアチャンネル

チャンネルは、ソフトウェアパッケージをグループ化する方法です。 ソフトウェアパッケージはリポジトリによって提供され、リポジトリはチャンネルに関連付けられています。 クライアントをソフトウェアチャンネルにサブスクライブすると、クライアントは、これに関連付けられたソフトウェアをインストールし、更新できます。

SUSE Managerでは、チャンネルはベースチャンネルと子チャンネルに分割されます。 この方法でチャンネルを編成すると、互換性のあるパッケージのみが各システムにインストールされるようになります。 クライアントは、1つのベースチャンネルのみをサブスクライブして、登録中にクライアントのオペレーティングシステムおよびアーキテクチャに基づいて割り当てる必要があります。 ベンダによって提供される有料チャンネルでは、関連付けされたサブスクリプションを持っている必要があります。

ベースチャンネルは、特定のオペレーティングシステムの種類、バージョン、およびアーキテクチャのために構築されたパッケージから構成されています。 たとえば、SUSE Linux Enterprise Server 15 x86-64のベースチャンネルには、そのオペレーティングシステムおよびアーキテクチャと互換性のあるソフトウェアのみが含まれています。

子チャンネルはベースチャンネルに関連付けられていて、ベースチャンネルと互換性のあるパッケージのみを提供します。 システムは、ベースチャンネルの複数の子チャンネルにサブスクライブできます。 システムがベースチャンネルに割り当てられている場合、そのシステムは関連する子チャンネルをインストールできます。 たとえば、システムがSUSE Linux Enterprise Server 15 **x86_64** ベースチャンネルに割り当てられている場合、互換性のあるベースチャンネルまたは関連する子チャンネルのいずれかから利用できるパッケージのみインストールまたは更新できます。

SUSE ManagerのWeb UIで、**ソフトウェア › チャンネル一覧 › すべて**に移動して、利用できるチャンネルをブラウズできます。 **ソフトウェア › 管理 › チャンネル**に移動して、チャンネルを変更または新しいチャンネルを作成できます。

カスタムチャンネルなどチャンネルを使用する方法の詳細については、**Administration › Channel-management**を参照してください。

ブートストラップ後のインストーラ更新チャンネルの処理

> クライアントシステムがブートストラップされたら、**インストーラ更新**チャンネルを削除する必要があります。標準の更新チャンネルにはすでに必要な更新が含まれているため、このチャンネルは冗長になります。
>
> さらに、移行中は、このチャンネルは不要であり、使用しないでください。

2.1.1. SUSE Package Hubで提供されるパッケージ

SUSE Package HubはSUSE Linux Enterprise製品の拡張機能で、openSUSEコミュニティで提供する追加オープンソースソフトウェアを提供しています。

> SUSE Package Hubのパッケージは、openSUSEコミュニティによって提供されます。パッケージはSUSEではサポートされていません。

クライアントでSUSE Linux Enterpriseオペレーティングシステムを使用している場合、SUSE Package Hub拡張機能を有効にして、これらの追加パッケージにアクセスできます。 アクセスすると、クライアントのサブスクライブ先にできるSUSE Package Hubチャンネルが提供されます。

SUSE Package Hubは多数のパッケージを提供しており、大量のディスク容量を使用してパッケージの同期に長時間かかる場合があります。 提供するパッケージが必要でない場合、SUSE Package Hubを有効にしないでください。

サポートされていないパッケージを誤ってインストールまたは更新しないためには、最初にすべてのSUSE Package Hubパッケージを拒否するコンテンツライフサイクル管理戦略の実装をお勧めします。 その後、必要なパッケージを明示的に有効にできます。 コンテンツライフサイクル管理の詳細については、**Administration › Content-lifecycle**を参照してください。

2.1.2. AppStreamで提供されるパッケージ

Red Hatベースのクライアントの場合、AppStreamから追加パッケージを利用できます。 ほとんどの場合、AppStreamパッケージでは、必要なソフトウェアをすべて持っていることを確認する必要があります。

SUSE ManagerのWeb UIでAppStreamパッケージを管理している場合、パッケージの更新に関して相反する推奨事項が表示される場合があります。 これは、SUSE Managerでモジュールのメタデータを正しく解釈できないことが原因です。 コンテンツライフサイクル管理(CLM)のAppStreamフィルタを使用して、AppStreamリポジトリを非モジュール型リポジトリに変換して、一部の更新操作で使用できます。 CLM AppStreamフィルタの詳細については、**Administration › Content-lifecycle-examples**を参照してください。

2.1.3. EPELで提供されるパッケージ

Red Hatベースのクライアントの場合、EPEL(エンタープライズ版Linux用の追加パッケージ)から追加パッケージを利用できます EPELはオプションのパッケージリポジトリで、追加ソフトウェアが提供されます。

> EPELのパッケージは、Fedoraコミュニティによって提供されます。 このパッケージはSUSEではサポートされていません。

クライアントでRed Hatオペレーティングシステムを使用している場合、EPEL拡張機能を有効にして、これ

らの追加パッケージにアクセスできます。　　アクセスすると、クライアントのサブスクライブ先にできるEPELチャンネルが提供されます。

EPELは多数のパッケージを提供しており、大量のディスク容量を使用してパッケージの同期に長時間かかる場合があります。 提供するパッケージが必要でない場合、EPELリポジトリを有効にしないでください。

サポートされていないパッケージを誤ってインストールまたは更新しないためには、最初にすべてのEPELパッケージを拒否するコンテンツライフサイクル管理(CLM)戦略の実装をお勧めします。　その後、必要なパッケージを明示的に有効にできます。　コンテンツライフサイクル管理の詳細については、**Administration › Content-lifecycle**を参照してください。

2.1.4. SUSE Linux EnterpriseクライアントのUnified Installer更新チャンネル

このチャンネルは、オペレーティングシステムをインストールする前に、Unified Installerが最新であることを確認するためにUnified Installerで使用されます。 すべてのSUSE Linux Enterprise製品は、インストール中にインストーラ更新チャンネルにアクセスできる必要があります。

SUSE Linux Enterprise Serverクライアントでは、更新を含む製品を追加するときにデフォルトでインストーラ更新チャンネルが同期します。また、これらの製品チャンネルで自動インストールディストリビューションを作成するときに有効になります。

SUSE Linux Enterprise for SAPなどその他すべてのSUSE Linux Enterpriseの亜種では、インストーラ更新チャンネルを手動で追加する必要があります。 そのためには、適切なSUSE Linux Enterprise Serverインストーラ更新チャンネルをこれらの SUSE Linux Enterprise亜種のベースチャンネルの下に複製します。 チャンネルを複製した後にこれらのSUSE Linux Enterprise亜種の自動インストールディストリビューションを作成するとき、そのインストーラ更新チャンネルが自動的に使用されます。

2.1.5. ソフトウェアリポジトリ

リポジトリはソフトウェアパッケージを収集するために使用されます。　ソフトウェアリポジトリにアクセスできる場合、リポジトリが提供するソフトウェアをインストールできます。　　1つ以上のリポジトリをSUSE Managerのソフトウェアチャンネルに関連付け、クライアントをそのチャンネルに割り当て、クライアントのパッケージにインストールして更新する必要があります。

SUSE Managerのほとんどのデフォルトチャンネルは、正しいリポジトリに関連付けられています。 カスタムチャンネルを作成している場合、アクセスできるリポジトリまたは自分で作成したリポジトリを関連付ける必要があります。

カスタムリポジトリおよびチャンネルの詳細については、**Administration › Custom-channels**を参照してください。

ローカルリポジトリの場所

Saltクライアントでローカルリポジトリを設定して、SUSE Managerチャンネルが提供しないパッケージを提供できます。

ほとんどの場合、クライアントシステムはローカルリポジトリを必要としません。　ロー

> カルリポジトリを使用すると、クライアントで使用できるパッケージがどれかという問題が発生する可能性があります。 この問題が発生すると、予期しないパッケージがインストールされる場合があります。

ローカルリポジトリは、オンボーディング中に無効になります。

Saltクライアントの場合、チャンネル状態が実行されるたびにローカルリポジトリが無効になります。 たとえば、highstateを適用したり、パッケージアクションを実行したりする場合などです。

オンボーディング後もローカルリポジトリを有効にしておく必要がある場合は、影響を受けるSaltクライアントに対して次のpillarを設定する必要があります。

/srv/pillar/top.slsファイルを編集します。

```
base:
  'minionid':
    - localrepos
```

/srv/pillar/localrepos.slsファイルを編集します。

```
mgr_disable_local_repos: False
```

クライアントがオンボードを完了した後、ローカルリポジトリを次の場所に追加できます。

表 16. ローカルリポジトリの場所

クライアントのオペレーティングシステム	ローカルリポジトリーのディレクトリ
SUSE Linux Enterprise Server	**/etc/zypp/repos.d**
openSUSE	**/etc/zypp/repos.d**
SUSE Linux Enterprise Server Expanded Support	**/etc/yum.repos.d/**
Red Hat Enterprise Linux	**/etc/yum.repos.d/**
CentOS	**/etc/yum.repos.d/**
Ubuntu	**/etc/apt/sources.list.d/**
Debian	**/etc/apt/sources.list.d/**

2.1.6. ソフトウェア製品

SUSE Managerでは、製品でソフトウェアを使用できます。 SUSEサブスクリプションでは、さまざまな製品にアクセスできます。製品には、SUSE ManagerのWeb UIで**管理 › セットアップウィザード › 製品**に移動してブラウズし、選択できます。

製品には、任意の数のソフトウェアチャンネルが含まれています。 **製品チャンネルの表示** アイコンをクリックし、製品に含まれているチャンネルを表示します。 製品を追加して正常に同期すると、製品で提供してい

るチャンネルにアクセスできるようになり、SUSE　Managerサーバとクライアントで製品のパッケージを使用できます。

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

詳細については、**Installation-and-upgrade › Setup-wizard**を参照してください。

2.2. ブートストラップリポジトリ

ブートストラップリポジトリには、ブートストラップ中にSaltまたは従来のクライアントを登録するために必要なパッケージと、クライアントにSaltをインストールするためのパッケージが含まれています。 製品を同期するとき、ブートストラップリポジトリは、自動的に作成され、SUSE　Managerサーバに再生成されます。

2.2.1. ブートストラップリポジトリの作成準備

同期する製品を選択するとき、ブートストラップリポジトリは、必須のチャンネルすべてが完全にミラーリングされるとすぐに自動的に作成されます。

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

2.2.2. 自動モードのオプション

ブートストラップリポジトリの自動作成動作を変更できます。 このセクションでは、さまざまな設定を説明します。

フラッシュモード::

フラッシュモード

デフォルトでは、既存のリポジトリは、最新パッケージでのみ更新されます。 代わりに、必ず空のリポジトリで始まるように設定できます。 この動作を有効にするには、**/etc/rhn/rhn.conf**で次の値を追加または編集します。

```
server.susemanager.bootstrap_repo_flush = 1
```

自動モード::

自動モード

デフォルトでは、ブートストラップリポジトリの自動再生成は有効になっています。 無効にするには、**/etc/rhn/rhn.conf**で次の値を追加または編集します。

```
server.susemanager.auto_generate_bootstrap_repo = 0
```

ブートストラップデータファイルの設定

このツールは、各ディストリビューションに必要なパッケージに関する情報を含むデータファイルを使用します。 データファイルは**/usr/share/susemanager/mgr_bootstrap_data.py**に保存されています。 SUSEはこのファイルを定期的に更新します。 このファイルを変更する場合、直接編集しないでください。 代わりに、同じディレクトリにコピーを作成し、コピーを編集します。

```
cd /usr/share/susemanager/
cp mgr_bootstrap_data.py my_data.py
```

変更したら、SUSE Managerを設定して新しいファイルを使用します。 **/etc/rhn/rhn.conf**でこの値を追加または編集します。

```
server.susemanager.bootstrap_repo_datamodule = my_data
```

次の更新時、SUSEの新しいデータによって、新しいデータファイルではなく元のデータファイルが上書きされます。 SUSEによって行われた変更を使用して新しいファイルを最新に保つ必要があります。

2.2.3. ブートストラップリポジトリの手動生成

デフォルトでは、ブートストラップリポジトリは毎日再生成されます。 コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

プロシージャ: SUSE Linux Enterpriseのブートストラップリポジトリの生成

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、次のコマンド用のブートストラップリポジトリを作成するために使用できるディストリビューションをリストします。

```
mgr-create-bootstrap-repo -l
```

2. 製品ラベルとして適切なリポジトリ名を使用して、ブートストラップリポジトリを作成します。

```
mgr-create-bootstrap-repo -c SLE-version-x86_64
```

3. または、利用可能なディストリビューション一覧のディストリビューション名の横に表示されている番号を使用します。

クライアントリポジトリは**/srv/www/htdocs/pub/repositories/**にあります。

複数の製品(SLESとSLES for SAPなど)をミラーリング済みの場合、またはカスタムチャンネルを使用している場合、ブートストラップリポジトリを作成するときに使用する親チャンネルを指定する必要が生じる場合があります。 これは、あらゆる状況で必須ではありません。 たとえば、SLES 15の一部のバージョンには共通のコードベースがあるため、親チャンネルを指定する必要はありません。 このプロシージャは、ご使用の環境で必要な場合のみ使用します。

オプションのプロシージャ: ブートストラップリポジトリの親チャンネルの指定

1. 利用できる親チャンネルを確認します。

```
mgr-create-bootstrap-repo -c SLE-15-x86_64
Multiple options for parent channel
found. (親チャンネルの複数にオプションが表示されます。) Please use option
--with-parent-channel <label> and choose one of: (オプション --with-parent-channel
<label>を使用し、次のいずれかを選択してください。)
- sle-product-sles15-pool-x86_64
- sle-product-sles_sap15-pool-x86_64
- sle-product-sled15-pool-x86_64
```

2. 適切な親チャンネルを指定します。

```
mgr-create-bootstrap-repo -c SLE-15-x86_64 --with-parent-channel sle-product-sled15-
pool-x86_64
```

複数アーキテクチャを含むリポジトリ

複数の異なるアーキテクチャを含むブートストラップリポジトリを作成している場合、すべてのアーキテクチャが正しく更新されることに注意を払う必要があります。 たとえば、SLEのx86-64アーキテクチャおよびIBM Zアーキテクチャは、同じブートストラップリポジトリURL(**/srv/www/htdocs/pub/repositories/sle/15/2/bootstrap/**)を使用します。

フラッシュオプションを有効にすると、複数のアーキテクチャのブートストラップリポジトリを生成しようとしても、生成されるアーキテクチャは1つのみです。 この動作を回避するには、追加のアーキテクチャを

作成するとき、コマンドプロンプトで**--no-flush**オプションを使用します。 次に例を示します。

```
mgr-create-bootstrap-repo -c SLE-15-SP2-x86_64
mgr-create-bootstrap-repo --no-flush -c SLE-15-SP2-s390x
```

2.2.4. ブートストラップとカスタムチャンネル

カスタムチャンネルを使用している場合、**mgr-create-bootstrap-repo**コマンドを使用して**--with-custom-channels**オプションを使用できます。 この場合、使用する親チャンネルも指定する必要があります。

カスタムチャンネルを使用すると、ブートストラップリポジトリの自動作成が失敗する場合があります。 この場合、リポジトリを手動で作成する必要があります。

カスタムチャンネルの詳細については、**Administration › Custom-channels**を参照してください。

2.3. アクティベーションキー

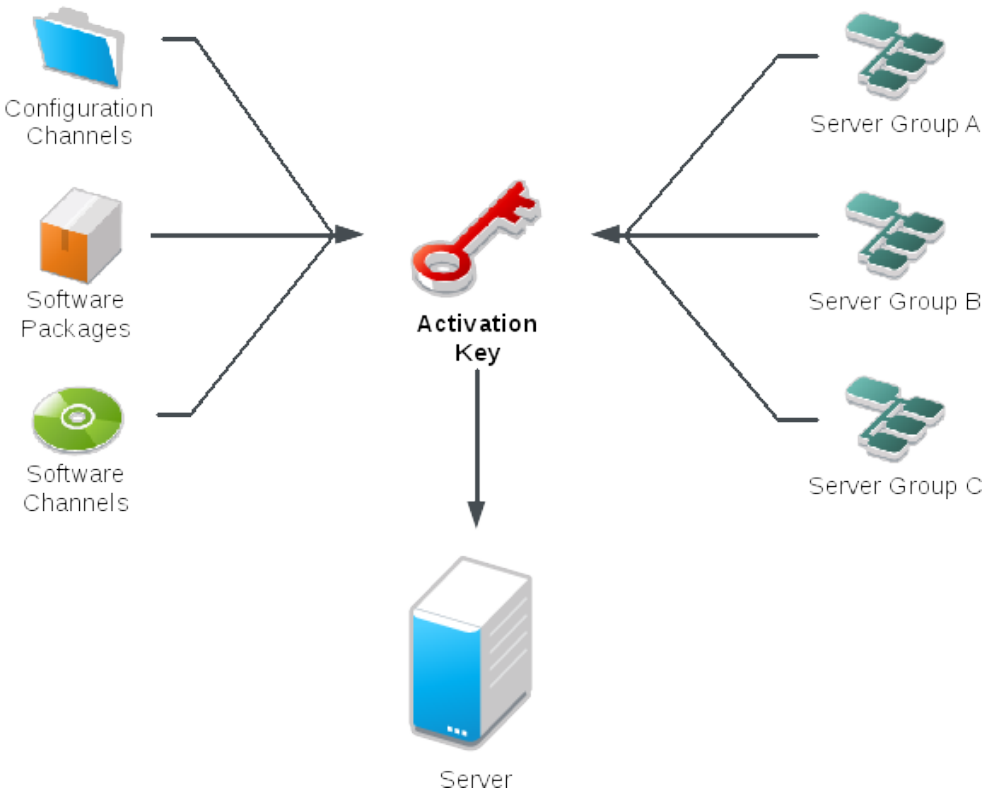
アクティベーションキーは従来のクライアントとSaltクライアントで使用し、クライアントが正しいソフトウェアのエンタイトルメントを持ち、適切なチャンネルに接続して関連グループに加入するようします。 それぞれのアクティベーションキーは、キーを作成するときに設定できる組織にひもづけされます。

SUSE Managerでは、アクティベーションキーは、ラベルの付いた一連の設定です。 アクティベーションキーに関連付けられている設定は、すべて適用できます。そのためには、キーのラベルをパラメータにしてブートストラップスクリプトに追加します。 アクティベーションキーラベルをブートストラップスクリプトと組み合わせて使用することをお勧めします。 ブートストラップスクリプトが実行されると、そのラベルに関連付けられているすべての設定が、スクリプトを実行しているシステムに適用されます。

アクティベーションキーは以下を指定できます。

- チャンネルの割り当て
- システムの種類またはアドオンのエンタイトルメント
- 接続メソッド
- 設定ファイル
- インストールするパッケージ
- システムグループの割り当て

アクティベーションキーは、クライアント登録時に使用され、再使用されることはありません。 アクティベーションキーで指定する内容に関係なく、クライアントは登録後、任意の方法で変更できます。 アクティベーションキーとクライアントの関連付けは、履歴を残すためだけに記録されます。



プロシージャ: アクティベーションキーの作成

1. SUSE ManagerのWeb UIで、管理者として**システム › アクティベーションキー**に移動します。
2. **[キーの作成]**ボタンをクリックします。
3. ［**アクティベーションキーの詳細**］ページの［**説明**］フィールドにアクティベーションキーの説明を入力します。
4. ［**キー**］フィールドにアクティベーションキーの名前を入力します。 たとえば、SUSE Linux Enterprise Server 15 SP4では**SLES15-SP4**と入力します。

   SUSE製品の［**キー**］フィールドではカンマまたは二重引用符を使用しないでください。 ただし、Red Hat製品ではカンマを使用する必要があります。

   - 他のすべての文字を使用できますが、**<>(){}**(スペースを含む)は自動的に削除されます。
   - フィールドが空のままの場合、ランダムな文字列が生成されます。

5. ［**ベースチャンネル**］ドロップダウンボックスで、適切なベースソフトウェアチャンネルを選択し、関連する子チャンネルへのデータ入力を許可します。 詳細については、Setup Wizardと**Administration › Custom-channels**を参照してください。
6. 必要な子チャンネルを選択します(必須のSUSE Managerツールや更新チャンネルなど)。
7. いずれかのオプションを有効にする必要がある場合は、［**付属エンタイトルメント**］チェックボックスにチェックを付けます。

8. ［**接続メソッド**］は［**デフォルト**］のままにすることをお勧めします。
9. ［**汎用デフォルト**］設定には、チェックを入れないようにすることをお勧めします。
10. **[アクティベーションキーの作成]**をクリックしてアクティベーションキーを作成します。
11. ［**設定ファイルの展開**］チェックボックスをチェックし、このキーの設定管理を有効にし、**[アクティベーションキーの更新]**をクリックしてこの変更を保存します。

［**設定ファイルの展開**］チェックボックスは、アクティベーションキーを作成するまで表示されません。 設定管理を有効にする必要がある場合、前に戻ってボックスにチェックを付けます。

2.3.1. 複数のアクティベーションキーの結合

従来のクライアントでブートストラップスクリプトを実行するとき、アクティベーションキーを結合できます。 キーを結合すると、システムにインストールされる機能をより詳細に制御でき、大規模な環境や複雑な環境でのキーの重複が軽減されます。

アクティベーションキーの結合は、従来のクライアント上でのみ動作します。 Saltクライアントはアクティベーションキーの結合をサポートしていません。 Saltクライアントで結合キーを使用する場合、最初のキーのみ使用されます。

コマンドプロンプトまたはシングル自動インストールプロファイルで複数のアクティベーションキーを指定できます。

SUSE Managerサーバのコマンドプロンプトで、**rhnreg_ks**コマンドを使用し、カンマでキーの名前を区切ります。 Kickstartプロファイルで複数のキーを指定するには、**システム › 自動インストール**に移動し、使用するプロファイルを編集します。

値が競合するとクライアントの登録に失敗するため、アクティベーションキーを結合するときには注意してください。 結合する前に次の値で情報の競合がないことを確認してください。

- ソフトウェアパッケージ
- ソフトウェアの子チャンネル
- 設定チャンネル。

競合は、検出されると次のように処理されます。

- ベースソフトウェアチャンネルの競合: 登録は失敗します。
- システムの種類の競合: 登録は失敗します。
- **enable configuration** フラグの競合: 設定管理が有効になります。
- 一方のキーがシステム固有のキーである場合: 登録は失敗します。

2.3.2. 再アクティベーションキー

クライアントを再登録してすべてのSUSE Manager設定を再取得するために、再アクティベーションキーを1回だけ使用できます。 再アクティベーションキーはクライアント固有で、システムID、履歴、グループ、およびチャンネルが含まれています。

再アクティベーションキーを作成するには、［**システム**］に移動し、再アクティベーションキーを作成するクライアントをクリックし、**詳細 › 再アクティベーション**タブに移動します。 **[新しいキーの生成]**をクリックして再アクティベーションキーを作成します。 後で使用できるようにキーの詳細を書き留めます。 特定のシステムIDに関連付けられていない通常のアクティベーションキーと異なり、ここで作成されるキーは、**システム › アクティベーションキー**ページに表示されません。

Saltクライアントの場合、再アクティベーションキーを作成した後、**/etc/salt/minion.d/susemanager.conf**の**management_key** grainとして使用できます。次に例を示します。

```
grains:
    susemanager:
        management_key: "re-1-daf44db90c0853edbb5db03f2b37986e"
```

salt-minionプロセスを再起動して再アクティベーションキーを適用します。

ブートストラップスクリプトで再アクティベーションキーを使用できます。 ブートストラップスクリプトの詳細については、**Client-configuration › Registration-bootstrap**を参照してください。

従来のクライアントの場合、再アクティベーションキーを作成した後、**rhnreg_ks**コマンドラインユーティリティでこのキーを使用できます。 このコマンドを実行すると、クライアントが再登録され、そのSUSE Manager設定が復元されます。 従来のクライアントでは、再アクティベーションキーをアクティベーションキーと結合して、単一システムプロファイルで複数のキーの設定を集約できます。 次に例を示します。

```
rhnreg_ks --server=<server-url>/XMLRPC \
          --activationkey=<reactivation-key>,<activationkey> \
          --force
```

既存のSUSE Managerプロファイルでクライアントを自動インストールすると、そのプロファイルは、再アクティベーションキーを使用して、システムを再登録し、その設定を復元します。 プロファイルベースの自動インストール実行中は、このキーを再生成、削除、または使用しないでください。 このような操作を実行すると、自動インストールは失敗します。

2.3.3. アクティベーションキーのベストプラクティス

デフォルトの親チャンネル

SUSEマネージャのデフォルトの親チャンネルを使用しないでください。 この設定では、SUSE Managerは、インストールされるオペレーティングシステムに最適な親チャンネルを強制的に選択します。その場合、予期しない動作が発生する可能性があります。 代わりに、それぞれのディストリビューションおよびアーキテクチャに固有のアクティベーションキーを作成することをお勧めします。

アクティベーションキーによるブートストラップ

ブートストラップスクリプトを使用している場合、各スクリプトにアクティベーションキーを作成することを検討してください。 作成によって、チャンネルの割り当て、パッケージのインストール、システムグループメンバーシップ、および設定チャンネルの割り当ての整合性を取ることができます。 登録後にシステムで手動操作する必要も減ります。

LTSSクライアントのブートストラップ

LTSSサブスクリプションでクライアントをブーストラッピングする場合は、アクティベーションキーの作成中にLTSSチャンネルを含めます。

帯域幅の要件

アクティベーションキーを使用すると、登録時にソフトウェアが自動ダウンロードされることがあります。この動作は、帯域幅に制約がある環境では望ましくない場合があります。

次のオプションによって帯域幅使用条件が作成されます。

- SUSE Product Poolチャンネルを割り当てると、対応する製品ディスクリプタパッケージが自動インストールされます。
- ［**パッケージ**］セクションのパッケージがインストールされます。
- ［**設定**］セクションのSaltの状態によっては、その内容に応じてダウンロードがトリガされる場合があります。

キーラベルの命名

読んで理解しやすい名前をアクティベーションキーに入力しないと、システムが数値の文字列を自動生成するため、キーの管理が困難になる場合があります。

キーを追跡できるようにアクティベーションキーの命名規則を検討してください。 組織のインフラストラクチャに関係がある名前を付けておくと、複雑な操作の実行も簡単になります。

キーラベルを作成する場合、次のヒントを考慮してください。

- OSの名前(必須): キーには、設定を指定するOS名を必ず含める必要があります
- アーキテクチャ名(推奨): 会社で稼働しているアーキテクチャ(たとえば、x86_64)が複数ある場合、アーキテクチャの種類をラベルに含めることをお勧めします。
- サーバの種類の名前: このサーバの使用目的。
- 場所名: サーバの配置場所(部屋、ビル、部署)。
- 日付: 保守期間(四半期など)。
- カスタム名: 組織のニーズに合う命名規則。

アクティベーションキーラベルの名前の例:

```
sles15-sp4-web_server-room_129-x86_64
```

```
sles15-sp4-test_packages-blg_502-room_21-ppc64le
```

SUSE製品の［**キー**］フィールドではカンマを使用しないでください。 ただし、Red Hat製品ではカンマを使用する必要があります。 詳細については、**Reference › Systems**を参照してください。

含めるチャンネル

アクティベーションキーを作成するときは、このキーに関連付けられているソフトウェアチャンネルも考慮する必要があります。 キーには、特定のベースチャンネルを割り当てる必要があります。 デフォルトのベースチャンネルの使用はお勧めしません。 詳細については、**Client-configuration › Registration-overview**でインストールしているクライアントオペレーティングシステムを参照してください。

2.4. GPGキー

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

ほとんどの場合、クライアントにソフトウェアをインストールできるようにGPG設定を調整する必要はありません。

RPMパッケージに直接署名することはできますが、Debianベースのシステムではメタデータにのみ署名し、チェックサムのチェーンを使用してパッケージを保護します。 RPMベースのほとんどのシステムでは、署名されたパッケージに加え、署名されたメタデータも使用します。

2.4.1. クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアントの**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。 拡張サポートクライアントのGPGキーの場合も同様です。 ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートさ

れ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、Saltファイルシステムに GPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys** pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。 **ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。 ［**GPGキーのURL**］に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。
2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

```
uyuni-gpg-pubkey-0d20833e.key
```

以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

Chapter 3. クライアントの管理方法

SUSE Managerサーバがクライアントと通信する方法は多数あります。 使用方法は、クライアントのタイプおよびネットワークアーキテクチャによって決まります。

SUSE Managerデーモン(**rhnsd**)は従来のクライアントシステムで動作し、SUSE Managerと定期的に接続し、新しい更新および通知を確認します。 Saltクライアントには適用されません。

SSHでのプッシュおよびSalt SSHでのプッシュメソッドは、クライアントがSUSE Managerサーバに直接アクセスできない環境で使用されます。 この環境では、DMZと呼ばれるファイアウォール保護ゾーンにクライアントはあります。 内部ネットワークとの接続を開くことを認可されているシステム(SUSE Managerサーバなど)はDMZ内にはありません。

OSADは、SUSE Managerと従来のクライアントの間の代替の接続メソッドです。 OSADでは、スケジュールされているアクションを従来のクライアントがすぐに実行できます。 Saltクライアントには適用されません。

3.1. Saltクライアントの接続メソッド

ほとんどの場合、Saltクライアントは、デフォルトのブートストラップメソッドで正確に登録されます。

非接続設定でSaltクライアントを使用する必要がある場合、Salt SSHでのプッシュを設定できます。 この環境では、DMZと呼ばれるファイアウォール保護ゾーンにクライアントはあります。 Salt SSH接続メソッドの詳細については、**Client-configuration › Contact-methods-saltssh**を参照してください。

Saltクライアントを手動で設定してSUSE Managerサーバに接続する必要がある場合は、正しいネットワーク詳細を使用してSaltクライアントの設定ファイルを編集します。 Salt minion設定ファイルの接続メソッドの詳細については、**Client-configuration › Registration-cli**を参照してください。

3.1.1. オンボードの詳細

Saltには、minionのキーを保持するための独自のデータベースがあります。これは、SUSE Managerデータベースと同期を保つ必要があります。 Saltでキーが受け入れられると、SUSE Managerでオンボードプロセスがただちに開始されます。 オンボードプロセスは、**minion_id**と**machine-id**を検索して、SUSE Managerデータベース内で既存のシステムを探します。 何も見つからない場合は、新しいシステムが作成されます。**minion_id**または**machine-id**のエントリが見つかった場合、新しいシステムに合わせてシステムが移行されます。 両方のエントリとの一致が見つかり、一致したのが同一のシステムではない場合、オンボードはエラーで中断されます。 この場合、管理者は少なくとも1つのシステムを削除して競合を解決する必要があります。

3.1.2. Salt SSHでのプッシュ

Salt SSHでのプッシュは、SaltクライアントがSUSE Managerサーバに直接アクセスできない環境で使用されます。 この環境では、DMZと呼ばれるファイアウォール保護ゾーンにクライアントはあります。 内部ネットワークとの接続を開くことを認可されているシステム(SUSE Managerサーバなど)はDMZ内にはありません。

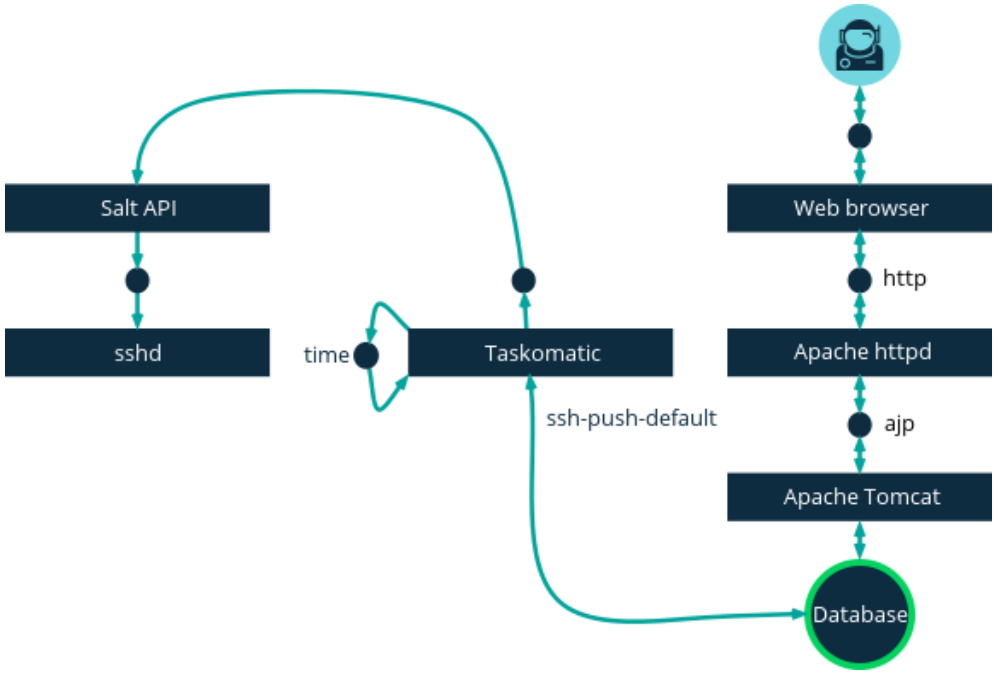
Salt SSHでのプッシュメソッドは、DMZにあるクライアントに内部ネットワークのSUSE Managerサーバか

ら暗号化トンネルを作成します。　すべてのアクションおよびイベントが実行された後、トンネルはクローズします。

サーバは、Salt　SSHを使用して、定期的にクライアントに接続し、チェックインし、スケジュールされたアクションおよびイベントを実行します。 Salt SSHの詳細については、**Specialized-guides › Salt**を参照してください。

この接続方法は、Saltクライアントでのみ動作します。　従来のクライアントでは、SSHでのプッシュを使用します。

次のイメージは、Salt　SSHでのプッシュプロセスのパスを示しています。　**Taskomatic**ブロックの左側のアイテムはすべて、SUSE Managerクライアントで実行されるプロセスを表します。



Salt　SSHでのプッシュを使用するには、クライアントで動作しているSSHデーモンが必要で、SUSE Managerサーバで動作している**salt-api**デーモンによって接続できます。　また、Pythonは、リモートシステムで使用できてSaltでサポートされているバージョンである必要があります。

Red Hat Enterprise Linux 5、CentOS 5、およびこれら以前のバージョンは、サポートされていないバージョンのPythonを使用しているため、サポートされません。

プロシージャ: Salt SSHでのプッシュを使用したクライアントの登録

1. SUSE　ManagerのWeb　UIで、**システム › ブートストラップ**に移動し、該当するフィールドに入力します。
2. SSHでのプッシュ接続メソッドが設定されたアクティベーションキーを選択します。　アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。
3. ［**Manage System Completely via SSH**］（SSHでシステムを完全に管理する）チェックボックスにチェックを付けます。

4. **[ブートストラップ]**をクリックして、登録を開始します。
5. **システム › 概要**に移動して、システムが正しく登録されたことを確認します。

使用可能なパラメータ

Salt SSHでのプッシュを設定している場合、ホスト、アクティベーションキー、パスワードなど、システムを登録するときに使用するパラメータを変更できます。 このパスワードはブートストラップでのみ使用し、どこにも保存されません。 今後のSSHセッションではすべて、キー/証明書のペアで認可されます。 これらのパラメータは**システム › ブートストラップ**で設定されます。

sudoユーザなどシステム全体で使用される永続パラメータも設定できます。 sudoユーザの設定について詳しくは、**Client-configuration › Contact-methods-pushssh**を参照してください。

アクションの実行

Salt SSHでのプッシュ機能は、taskomaticを使用し、**salt-ssh**を使用してスケジュール済みのアクションを実行します。 taskomaticジョブは、スケジュールされたアクションを定期的に確認して実行します。 従来のクライアントにおけるSSHでのプッシュと異なり、Salt SSHでのプッシュ機能は、スケジュールされたアクションに基づいて、完全な**salt-ssh**コールを実行します。

デフォルトでは、20個のSalt SSHアクションを同時に実行できます。 同時実行できるアクションの個数を増やすことができます。そのためには、次の行を設定ファイルに追加し、**parallel_threads**の値を調整します。 問題の発生を回避するために、同時実行アクション数を低い値に保つことをお勧めします。

```
taskomatic.sshminion_action_executor.parallel_threads = <number>
org.quartz.threadPool.threadCount = <value of parallel_threads + 20>
```

1つのクライアントで同時実行できるアクションの個数とtaskomaticで使用されるワーカスレッドの合計数が調整されます。 複数のクライアントでアクションを実行する必要がある場合、アクションは常に各クライアントで順次実行されます。

クライアントがプロキシ経由で接続されている場合、プロキシの**MaxSessions**設定を調整する必要があります。 この場合、平行接続数を総クライアント数の3倍に設定します。

今後の機能

Salt SSHでのプッシュでサポートされていない機能があります。 これらの機能はSalt SSHクライアント上では動作しません。

- OpenSCAPの監査
- ビーコン。次の結果になります。
  - **zypper**を使用してシステムのパッケージをインストールしても、パッケージ更新が呼び出されません。
  - 仮想ホストシステムがSalt SSHベースの場合、仮想ホスト関数(たとえば、ゲストへのホスト)が動作しません。

詳細情報

- Salt SSH全般については、**Specialized-guides › Salt** and https://docs.saltproject.io/en/latest/topics/ssh/を参照してください。
- SSHキーのローテーションについては、specialized-guides:salt/salt-ssh.pdfを参照してください。

3.1.3. Salt Bundle

Salt Bundleの概要

Salt Bundleは、Salt Minion、Python 3、必須のPythonモジュール、およびライブラリが含まれている1つのバイナリパッケージです。

Salt BundleはPython 3に付属していて、Saltを実行するためのすべての要件です。したがって、Salt Bundleは、システムソフトウェアとしてクライアントにインストールされているPythonバージョンを使用しません。 Salt Bundleは、指定のSaltバージョンの要件を満たさないクライアントにインストールできます。

SUSE Manager Salt Master以外のSalt MasterにSalt Minionを実行するシステムでSalt Bundleを使用することもできます。

Salt Bundleを使用してクライアントをMinionとして登録する

Salt Bundleを使用した登録方法は推奨の登録方法です。 このセクションでは、現在の実装の利点と制約について説明します。 Salt Bundleは、Salt、Python 3、およびSaltが依存しているPythonモジュールで構成されている**venv-salt-minion**として提供されます。 Web UIを使用したブートストラップもSalt Bundleを使用しているため、Web UIを使用したブートストラップはPythonに依存しません。 Salt Bundleを使用すると、クライアントがPythonインタープリターまたはモジュールを提供する必要がなくなります。

新しいクライアントをブートストラップする場合、Salt Bundleを使用した登録がデフォルトの方法です。 既存のクライアントをSalt Bundleの方式に切り替えることができます。 切り替える場合、**salt-minion**パッケージおよびその依存関係はインストールされたままになります。

Salt Minionを使用したSalt Bundleの使用

Salt Bundleは、SUSE Managerサーバ以外のSalt Masterによって管理されているSalt Minionと同時に使用できます。 Salt Bundleが、SUSE ManagerサーバがSalt Bundleの設定ファイルを管理するクライアントにインストールされる場合、**salt-minion**の設定ファイルは管理されません。詳細については、Salt Bundle configurationを参照してください。

- SUSE Managerサーバ以外のSalt Masterによって管理されているSalt Minionを使用してクライアントをブートストラップするには、ブートストラップスクリプトを生成するときに**mgr-bootstrap --force-bundle**を使用するか、またはブートストラップスクリプトで**FORCE_VENV_SALT_MINION**を**1**に設定することをお勧めします。
- Web UI **mgr_force_venv_salt_minion**を**true**に設定してブートストラップする場合、pillarをグローバルに指定できます。 詳細については、**Specialized-guides › Salt**を参照してください。

Salt MinionからSalt Bundleへの切り替え

salt-minionから**venv-salt-minion**に切り替えるためにSalt状態**util.mgr_switch_to_venv_minion**を使用できます。　移行プロセスのトラブルを回避するために、**venv-salt-minion**への切り替えは2ステップで実行することをお勧めします。

プロシージャ: util.mgr_switch_to_venv_minionを使用して状態をvenv-salt-minionに切り替える

1. まず、pillarを指定せずに**util.mgr_switch_to_venv_minion**を適用します。 **venv-salt-minion**に切り替わり、設定ファイルなどがコピーされます。　元の**salt-minion**の設定およびそのパッケージはクリーンアップされません。

```
salt <minion_id> state.apply util.mgr_switch_to_venv_minion
```

util.mgr_switch_to_venv_minionを適用し、**mgr_purge_non_venv_salt**を**True**に設定して**salt-minion**を削除し、**mgr_purge_non_venv_salt_files**を**True**に設定して**salt-minion**に関するすべてのファイルを削除します。　この2番目の手順によって、最初の手順が処理されたことが保証され、古い設定ファイルおよび古くなった**salt-minion**パッケージが削除されます。

```
salt <minion_id> state.apply util.mgr_switch_to_venv_minion
pillar='{"mgr_purge_non_venv_salt_files": True, "mgr_purge_non_venv_salt": True}'
```

切り替えの最初の手順をスキップして2番目の手順を実行すると、クライアント側でコマンドを実行するために使用される**salt-minion**を停止する必要があるため、状態適用プロセスは失敗する可能性があります。

他方、Salt Bundleのインストールを回避して代わりに**salt-minion**の使用を続けることも可能です。 この場合、次のいずれかのオプションを指定します。

- **--no-bundle**オプションを指定して**mgr-bootstrap**を実行します。
- 生成されたブートストラップスクリプトで**AVOID_VENV_SALT_MINION**を**1**に設定します。
- ブートストラップ状態の場合、**/srv/pillar/salt_bundle_config.sls**の**mgr_avoid_venv_salt_minion** pillarを**True**に設定します。

Salt BundleによるSalt SSH

Salt Bundleは、クライアントに対してSalt SSHアクションを実行するときにも使用されます。

シェルスクリプトは、Saltコマンドを実行する前に**venv-salt-minion**をインストールせずにSalt Bundleをターゲットシステムに配備します。Salt BundleにはSaltのコードベース全体が含まれているため、**salt-thin**は配備されません。Salt SSH (Web UIを使用するブートストラップを含む)は、バンドル内でPython 3インタープリターを使用します。ターゲットシステムには他のPythonインタープリターがインストールされている必要はありません。

Bundleを使用して配備されたPython 3は、クライアントでSalt SSHセッションを処理するために使用されるため、Salt SSH (Web UIを使用したブートストラップを含む)は、システムにインストールされているPython

に依存しません。

salt-thinの使用はフォールバック方法として有効にできますが、クライアントにPython 3をインストールする必要があります。 この方法は、開発目的でのみ存在しており、お勧めもサポートもしません。**/etc/rhn/rhn.conf**設定ファイルで**web.ssh_use_salt_thin**を**true**に設定します。

- ブートストラップリポジトリーは、Web UIを使用してクライアントをブートストラップする前に作成済みである必要があります。Salt SSHは、検出したターゲットオペレーティングシステムに基づいてブートストラップリポジトリーから取得されたSalt Bundleを使用しています。 詳細については、client-configuration:bootstrap-repository.pdfを参照してください。
- Salt SSHは、**/var/tmp**を使用して、Salt Bundleを配備し、バンドルされたPythonを使用してクライアント上でSaltコマンドを実行しています。 したがって、**noexec**オプションを指定して**/var/tmp**をマウントしないでください。 ブートストラッププロセスがクライアントに到達するためにSalt SSHを使用しているため、**/var/tmp**が**noexec**オプションでマウントされたクライアントをWeb UIでブートストラップすることはできません。

pipを使用したPythonパッケージによるSalt Bundleの拡張

Salt Bundleには**pip**が含まれており、バンドルされているSalt Minionの機能を追加のPythonパッケージで拡張できます。

デフォルトで、**salt <minion_id> pip.install <package-name>**は、**<package_name>**で指定されたPythonパッケージを**/var/lib/venv-salt-minion/local**にインストールします。

必要に応じて、**venv-salt-minion.service**の環境変数**VENV_PIP_TARGET**を設定することで、パス**/var/lib/venv-salt-minion/local**を上書きできます。 サービスにはsystemdのドロップイン設定ファイルを使用することをお勧めします。 設定ファイル**/etc/systemd/system/venv-salt-minion.service.d/10-pip-destination.conf**で実行できます。

```
[Service]
Environment=VENV_PIP_TARGET=/new/path/local/venv-salt-minion/pip
```

pipを使用してインストールしたPythonパッケージは、Salt Bundleの更新時に変更されません。 更新後にこのようなパッケージが使用可能で機能するようにするために、Salt Bundleの更新後に適用されるSaltの状態でパッケージをインストールすることをお勧めします。

3.2. 従来のクライアントの接続メソッド

従来のクライアントは、さまざまなメソッドでSUSE Managerサーバと通信できます。

SUSE Managerデーモン(**rhnsd**)は従来のクライアントシステムで動作し、SUSE Managerと定期的に接続し、新しい更新および通知を確認します。

SSHでのプッシュは、クライアントでSUSE　Managerサーバに直接接続できない環境で使用されます。　この環境では、DMZと呼ばれるファイアウォール保護ゾーンにクライアントはあります。 内部ネットワークとの接続を開くことを認可されているシステム(SUSE Managerサーバなど)はDMZ内にはありません。

OSADは、SUSE Managerと従来のクライアントの間の代替の接続メソッドです。 OSADでは、スケジュールされているアクションを従来のクライアントがすぐに実行できます。

SUSE　Manager　4.3リリースでは、従来のクライアントは非推奨になりました。　SUSE Manager 4.3より後のリリースでは、従来のクライアントと従来のプロキシはサポートされなくなります。これは2023年に予定されています。 新しいすべての配備ではSaltクライアントとSaltプロキシのみを使用し、既存の従来のクライアントとプロキシはSaltに移行することをお勧めします。

+ 従来のクライアントからSalt minionに移行するとき、その前に登録したクライアントを削除する必要はありません。 Salt minionとしてそれらを登録するだけでSaltは従来のクライアントで必要な手順を実行します。　従来のクライアントを削除済みの場合、Salt minionとしての登録はできなくなります。**Administration › Troubleshooting**を参照してください。

3.2.1. SUSE Managerデーモン(rhnsd)

SUSE　Managerデーモン(**rhnsd**)は従来のクライアントシステムで動作し、SUSE　Managerと定期的に接続し、新しい更新および通知を確認します。 Saltクライアントには適用されません。

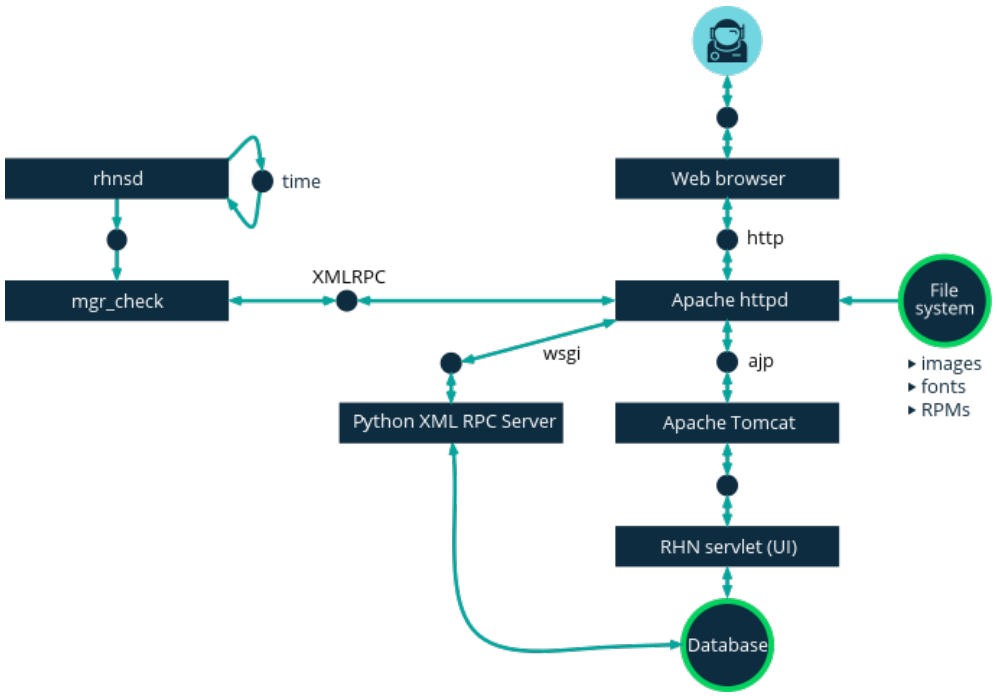
これは、SUSE Linux Enterprise 11およびRed Hat Enterprise Linux Server 6でのみ使用されます。その理由はこれらのシステムはsystemdを使用しないためです。　後者のオペレーティングシステムでは、systemdタイマー(**rhnsd.timer**)が使用され、**rhnsd.service**によって制御されます。

/etc/init.d/rhnsdでデーモンを起動します。

デフォルトでは4時間ごとに新しいアクションの確認が行われます。　つまり、スケジュールされたアクションをクライアントで実行するには時間がかかることがあります。

更新を確認するために、**rhnsd**は、**/usr/sbin/**にある外部の**mgr_check**プログラムを実行します。　このプログラムは、SUSE Managerへのネットワーク接続を確立する小さいアプリケーションです。 SUSE Managerデーモンは、ネットワークポートをリッスンせず、ネットワークに直接問い合わせしません。　すべてのネットワークアクティビティは、**mgr_check**ユーティリティによって実行されます。

次の図は、デフォルトの**rhnsd**プロセスのパスの概要を示しています。 **Python XMLRPC server**ブロックの左側のアイテムはすべて、SUSE Managerクライアントで実行されるプロセスを表します。



rhnsdの設定

rhnsd　初期化スクリプトには、クライアントシステムの設定ファイルが**/etc/sysconfig/rhn/rhnsd**にあります。

デーモンの重要なパラメータはそのチェックイン頻度です。 デフォルトの間隔は4時間(240分)です。 設定可能な最短間隔は1時間(60分)です。 間隔を1時間未満に設定すると、デフォルトの4時間(240分)に戻ります。

rhnsd 設定ファイルを変更する場合、次のコマンドをrootとしてデーモンを再起動し、変更を取得します。

```
/etc/init.d/rhnsd restart
```

rhnsd のステータスを表示するには、次のコマンドをrootとして実行します。

```
/etc/init.d/rhnsd status
```

SUSE Linux Enterprise 12以降では、デフォルトの間隔は**/etc/systemd/system/timers.target.wants/rhnsd.timer**で設定されます。このセクションでは次のようになります。

```
[Timer]
OnCalendar=00/4:00
RandomizedDelaySec=30min
```

systemctlを使用して**rhnsd.timer**の上書きドロップインファイルを作成できます。

```
systemctl edit rhnsd.timer
```

たとえば、2時間の間隔を設定する場合、次のようになります。

```
[Timer]
OnCalendar=00/2:00
```

ファイルは**/etc/systemd/system/rhnsd.timer.d/override.conf**として保存されます。

systemdタイマーの詳細については、**systemd.timer**および**systemctl**のマニュアルを参照してください。

OSAD

OSADは、SUSE Managerと従来のクライアントの間の代替の接続メソッドです。 デフォルトでは、SUSE Managerは**rhnsd**を使用します。これは、4時間ごとにサーバに接続し、スケジュールされたアクションを実行します。 OSADでは、スケジュールされているアクションを従来のクライアントがすぐに実行できます。

rhnsdに加えて、OSADを使用します。 **rhnsd**を無効にすると、クライアントは24時間後に未確認と表示されます。

OSADには異なるコンポーネントがいくつかあります。

- **osa-dispatcher**サービスは、サーバで動作し、データベースチェックを使用して、クライアントをpingする必要があるかどうか、およびアクションを実行する必要があるかどうかを判定します。
- **osad**サービスは、クライアントで動作します。このサービスは、**osa-dispatcher**からpingに応答し、**mgr_check**を実行して指示された場合にはアクションを実行します。
- **jabberd**サービスは、クライアントとサーバの間の通信に**XMPP**プロトコルを使用するデーモンです。**jabberd**サービスは認証も処理します。
- **mgr_check**ツールは、クライアントで動作し、アクションを実行します。 このツールは、**osa-dispatcher**サービスから通信によってトリガされます。

osa-dispatcherは、クエリを定期的に実行し、クライアントがネットワークアクティビティを最後に示したタイミングを確認します。 最近アクティビティを示していないクライアントを見つけた場合、**jabberd**を使用して、SUSE Managerサーバで登録されているクライアントのすべてで実行している**osad**インスタンスをすべてpingします。 **osad**インスタンスは、**jabberd**を使用してpingに応答します。これは、サーバ上のバックグラウンドで実行されています。 **osa-dispatcher**が応答を受信すると、クライアントをオンラインとしてマークします。 **osa-dispatcher**が一定の時間内に応答を受信できない場合、クライアントをオフラインとしてマークします。

OSAD対応のシステムでアクションをスケジュールすると、タスクはすぐに実行されます。 **osa-dispatcher**は、実行する必要があるアクションのクライアントを定期的に確認します。 実行保留中アクションが見つかった場合、**jabberd**を使用してクライアントで**mgr_check**を実行し、次にアクションを実行します。

OSADクライアントは、サーバの完全修飾ドメイン名(FQDN)を使用して**osa-dispatcher**サービスと通信します。

osad通信にはSSLが必要です。 SSL証明書が利用できない場合、クライアントシステムのデーモンは接続できません。 ファイアウォールルールで必要なポートの許可が設定されていることを確認します。 詳細につい

ては、**Installation-and-upgrade › Ports**を参照してください。

プロシージャ: OSADの有効化

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**osa-dispatcher**サービスを起動します。

```
systemctl start osa-dispatcher
```

2. それぞれのクライアントで、**mgr-osad**パッケージを［**ツール**］子チャンネルからインストールします。 **mgr-osad**パッケージはクライアントにのみインストールする必要があります。 **mgr-osad**パッケージをSUSE Managerサーバにインストールする場合、**osa-dispatcher**パッケージと競合します。
3. それぞれのクライアントでrootとして**osad**サービスを起動します。

```
systemctl start osad
```

osadおよび**osa-dispatcher**はサービスとして実行されるため、**stop**、**restart**、**status**などの標準コマンドを使用して、管理できます。

それぞれのOSADコンポーネントは、ローカル設定ファイルを使用して設定されます。 すべてのOSADコンポーネントのデフォルトの設定パラメータを保存することをお勧めします。

Component	場所	設定ファイルへのパス
osa-dispatcher	サーバ	**/etc/rhn/rhn.conf** セクション: **OSAの設定**
osad	クライアント	**/etc/sysconfig/rhn/osad.conf**
osad ログファイル	クライアント	**/var/log/osad**
jabberd ログファイル	両方	**/var/log/messages**

OSADのトラブルシューティング

OSADクライアントをサーバに接続できない場合、または**jabberd**サービスでポート5552への応答に時間がかかる場合、開いているファイルの数が超過していることが原因である可能性があります。

それぞれのクライアントは、常に開いているサーバTCP接続が1つ必要で、1つのファイルハンドラを使用します。 現在開いているファイルハンドラの数が**jabberd**での使用が許可されている最大ファイル数を超えると、**jabberd**はリクエストをキューに入れ、接続を拒否します。

この問題を解決するには、**jabberd**のファイル制限数を増やします。そのためには、**/etc/security/limits.conf**設定ファイルを編集して次の行を追加します。

```
jabber soft nofile 5100
jabber hard nofile 6000
```

使用している環境で必要な制限数を計算するには、ソフト制限はクライアント数に100を加算し、ハード制

限は現在のクライアント数に1000を加算します。

上記の例では、現在のクライアント数を5000と想定しているため、ソフト制限は5100、ハード制限は6000です。

/etc/jabberd/c2s.xmlファイルの**max_fds**パラメータを選択ハード制限で更新する必要もあります。

```
<max_fds>6000</max_fds>
```

3.2.2. SSHでのプッシュ

SSHでのプッシュは、従来のクライアントでSUSE　Managerサーバに直接接続できない環境で使用されます。 この環境では、DMZと呼ばれるファイアウォール保護ゾーンにクライアントはあります。 内部ネットワークとの接続を開くことを認可されているシステム(SUSE Managerサーバなど)はDMZ内にはありません。

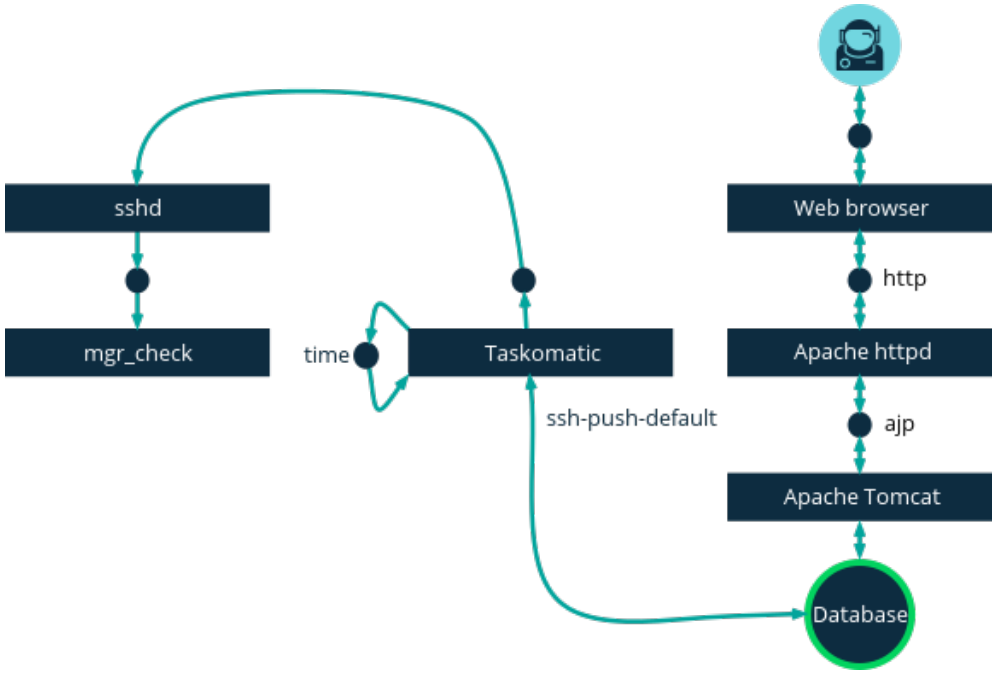
SSHでのプッシュメソッドは、DMZにあるクライアントに内部ネットワークのSUSE　Managerサーバから暗号化トンネルを作成します。　すべてのアクションおよびイベントが実行された後、トンネルはクローズします。

サーバは、SSHを使用して、定期的にクライアントに接続し、チェックインし、スケジュールされたアクションおよびイベントを実行します。

この接続方法は、従来のクライアントでのみ動作します。 Saltクライアントでは、Salt SSHでのプッシュを使用します。

プロビジョニングモデルを使用したシステムの再インストールは、SSHでのPushで管理されているクライアントでは現在サポートされていません。

次のイメージは、SSHでのプッシュプロセスのパスを示しています。　**Taskomatic**ブロックの左側のアイテムはすべて、SUSE Managerクライアントで実行されるプロセスを表します。



SSHでのトンネル接続では、使用できる2つのポート番号が必要です。1つ目はHTTPのトンネル用、2つ目はHTTPSでのトンネル用です(HTTPは登録プロセス中のみ必要です)。 デフォルトで使用されるポート番号は **1232** と **1233** です。 これらの番号を上書きするには、1024よりも大きい2つのカスタムポート番号を**/etc/rhn/rhn.conf**に追加します。

```
ssh_push_port_http = high_port_1
ssh_push_port_https = high_port_2
```

IPアドレスではなくホスト名を使用してクライアントに接続する場合、次のオプションを設定します。

```
ssh_push_use_hostname = true
```

クライアント接続を同時平行して開くために使用するスレッドの数も調整できます。 デフォルトでは、2つの同時平行スレッドが使用されます。 **/etc/rhn/rhn.conf**で**taskomatic.ssh_push_workers**を設定します。

```
taskomatic.ssh_push_workers = number
```

セキュリティ上の理由から、SSHでsudoを使用して、rootとしてではなく非特権ユーザとしてシステムにアクセスする必要がある場合があります。

プロシージャ: 非特権SSHアクセスの設定

1. 最新の**spacewalk-taskomatic**パッケージおよび**spacewalk-certs-tools**パッケージがSUSE Managerサーバにインストールされていることを確認してください。
2. それぞれのクライアントシステムで、適切な非特権ユーザを作成します。
3. それぞれのクライアントシステムで、**/etc/sudoers**ファイルを開き、次の行をコメントアウトします。

```
#Defaults targetpw   # ask for the password of the target user i.e. root
```

```
#ALL    ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults  targetpw'!
```

4. それぞれのクライアントシステムの **User privilege specification** セクションで、次の行を追加します。

```
<user> ALL=(ALL) NOPASSWD:/usr/sbin/mgr_check
<user> ALL=(ALL) NOPASSWD:/home/<user>/enable.sh
<user> ALL=(ALL) NOPASSWD:/home/<user>/bootstrap.sh
```

5. それぞれのクライアントシステムの**/home/<user>/.bashrc**ファイルで、次の行を追加します。

```
PATH=$PATH:/usr/sbin
export PATH
```

6. SUSE Managerサーバの**/etc/rhn/rhn.conf**設定ファイルで、次の行を追加または修正して、非特権ユーザ名を含めます。

```
ssh_push_sudo_user = <user>
```

クライアントがDMZにあり、サーバに接続できないため、**mgr-ssh-push-init**ツールを使用してクライアントをSUSE Managerサーバに登録する必要があります。

ツールを使用するには、クライアントのホスト名またはIPアドレス、およびSUSE Managerサーバの有効なブートストラップスクリプトへのパスが必要です。 ブートストラップの詳細については、**Client-configuration › Registration-bootstrap**を参照してください。

ブートストラップスクリプトは、SSHでのプッシュで設定されたスクリプトとアクティベーションキーを関連付ける必要があります。 アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

始める前に、SSHトンネルに使用するポートを指定済みであることを確認する必要があります。 ポート番号を変更する前にクライアントを登録した場合、再登録する必要があります。

SSHでのプッシュによって管理されているクライアントはサーバに直接接続できません。 **mgr-ssh-push-init**ツールを使用している場合、**rhnsd**デーモンは無効になっています。

プロシージャ: SSHでのプッシュを使用したクライアントの登録

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、次のコマンドを実行します。

```
# mgr-ssh-push-init --client <client> --register \
/srv/www/htdocs/pub/bootstrap/bootstrap_script --tunnel
```

   オプション: トンネルを使用しない場合、**--tunnel**オプションを削除できます。

2. オプション: **ssh_push_sudo_user**を定義済みの場合、**--notty**オプションを追加してrootのパスワードを使用できます。

3. SSH接続がアクティブであることを確認します。

```
# ssh -i /root/.ssh/id_susemanager -R <high_port>:<susemanager>:443 \
<client> zypper ref
```

例: SSHでのプッシュへのAPIアクセス

APIを使用して、使用する接続メソッドを管理できます。 このPythonコードの例では、接続メソッドが**ssh-push**に設定されます。

有効な値は次のとおりです。

- **default** (pull)
- **ssh-push**
- **ssh-push-tunnel**

```
client = xmlrpclib.Server(SUMA_HOST + "/rpc/api", verbose=0)
key = client.auth.login(SUMA_LOGIN, SUMA_PASSWORD)
client.system.setDetails(key, 1000012345, {'contact_method' : 'ssh-push'})
```

クライアントを登録済みで、SSHでのプッシュを使用するようにする場合、追加手順が必要です。 **mgr-ssh-push-init**ツールを使用してクライアントを設定できます。

プロシージャ: 登録済みシステムをSSHでのプッシュに移行する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、クライアントを設定します。

```
# mgr-ssh-push-init --client <client> \
/srv/www/htdocs/pub/bootstrap/bootstrap_script --tunnel
```

2. SUSE ManagerのWeb UIを使用して、クライアントの接続メソッドを **ssh-push** または **ssh-push-tunnel** に変更します。
3. オプション: 既存のアクティベーションキーを編集する必要がある場合、次のコマンドでできます。

```
client.activationkey.setDetails(key, '1-mykey', {'contact_method' : 'ssh-push'})
```

プロキシを使用して接続するクライアントにもSSHでのプッシュを使用できます。 始める前にプロキシが更新されていることを確認してください。

プロシージャ: SSHでのプッシュを使用したクライアントをプロキシに登録する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、クライアントを設定します。

```
# mgr-ssh-push-init --client <client> \
/srv/www/htdocs/pub/bootstrap/bootstrap_script --tunnel
```

2. SUSE Managerサーバのコマンドプロンプトで、SSHキーをプロキシにコピーします。

```
mgr-ssh-push-init --client <proxy>
```

3.3. 従来のクライアントをSaltクライアントに移行する

システムを従来のクライアントからSaltクライアントに移行するには、ユーザはSaltブートストラップスクリプトを作成し、それにクライアント システムを再登録します。

これは次のようにして実行できます。

- クライアントの再アクティベーションキーを生成し、
- 特定のアクティベーションキーを使用してブートストラップスクリプトを作成し、
- ブートストラップスクリプトを実行し、前述の再アクティベーションキーを使用してクライアントを登録します。

3.3.1. 再アクティベーションキーの生成

再アクティベーションキーを使用すると、クライアントを再登録し、SUSE Managerのすべての設定を回復することができます。 再アクティベーションキーの作成方法の詳細については、client-configuration:activation-keys.pdfを参照してください。

3.3.2. ブートストラップスクリプトの作成

ブートストラップスクリプトは、Web UIまたはコマンドラインから作成できます。 ブートストラップスクリプトの作成の詳細については、**Client-configuration › Registration-bootstrap**を参照してください。適切なアクティベーションキーを事前に生成するには、**Client-configuration › Activation-keys**を参照してください。

ブートストラップスクリプトを作成する際に、事前に生成したアクティベーションキーを使用します。

ブートストラップスクリプトが生成されると、サーバの**/srv/www/htdocs/pub/bootstrap**ディレクトリに保存されます。 または、HTTPS経由でブートストラップスクリプトにアクセスできます。 **<example.com>**をSUSE Managerサーバのホスト名に置き換えます。

```
https://<example.com>/pub/bootstrap/bootstrap.sh
```

ブートストラップスクリプトは、Saltクライアントの**venv-salt-minion**がブートストラップリポジトリにある場合にはこれをインストールしようとし、ブートストラップリポジトリにSalt bundleがない場合には**salt-minion**をインストールしようとします。 **salt-minion**が必要な場合、Salt bundleをインストールせずに**salt-minion**の使用を続けることができます。

詳細については、**Client-configuration › Contact-methods-saltbundle**を参照してください。

3.3.3. ブートストラップスクリプトの実行

最後のステップとして、ブートストラップスクリプトを実行してクライアントを移行および登録します。

プロシージャ: ブートストラップスクリプトの実行

1. SUSE Managerサーバにrootとしてサインインします。 コマンドプロンプトで、ブートストラップディレクトリに変更します。

```
cd /srv/www/htdocs/pub/bootstrap/
```

2. 次のコマンドを実行して、クライアントでブートストラップスクリプトを実行します。 **EXAMPLE.COM**をクライアントのホスト名に置き換え、**REACT_KEY**を再アクティベーションキーに置き換えます。

```
cat bootstrap-migrate-to-salt.sh | ssh root@EXAMPLE.COM REACTIVATION_KEY=REACT_KEY
/bin/bash
```

3. または、クライアントで次のコマンドを実行します。

```
curl -Sks https://server_hostname/pub/bootstrap/bootstrap-migrate-to-salt.sh |
REACTIVATION_KEY=REACT_KEY /bin/bash
```

   このスクリプトは、必要な依存関係をダウンロードします。

4. 新しいSalt minionを探す前に、必ず新しいSaltキーを受け入れてください。SUSE ManagerのWeb UIを開いて、**Salt › キー**に移動し、クライアントキーを受け入れることができます。
5. スクリプトの実行が完了すると、クライアントが正しく登録されたかどうかを確認できます。そのためには、SUSE ManagerのWeb UIを開き、**システム › 概要**に移動します。新しいクライアントが、管理システムタイプではなく、 Saltでリストされていることを確認してください。

SUSE Managerを使用してクライアントに新しいパッケージまたは更新がインストールされると、エンドユーザライセンスアグリーメント(EULA)が自動的に受け入れられます。 パッケージのEULAを確認するには、Web UIでパッケージ詳細ページを開きます。

ブートストラップスクリプトの使用に関する詳細については、**Client-configuration › Registration-bootstrap**を参照してください。

Chapter 4. クライアントの登録

クライアントをSUSE Managerサーバに登録する方法は複数あります。 このセクションでは、使用できるさまざまなメソッドについて説明します。 クライアントで実行するオペレーティングシステム固有の情報も含まれています。

始める前に次の項目を確認してください。

- クライアントで登録前にSUSE Managerサーバと日時が正しく同期されている。
- アクティベーションキーを作成済みである。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

ブートストラップ後のインストーラ更新チャンネルの処理

クライアントシステムがブートストラップされたら、**インストーラ更新**チャンネルを削除する必要があります。 標準の更新チャンネルにはすでに必要な更新が含まれているため、このチャンネルは冗長になります。

さらに、移行中は、このチャンネルは不要であり、使用しないでください。

SUSE Managerサーバをこのサーバ自体に登録しないでください。 SUSE Managerサーバは個別に管理するか、別のSUSE Managerサーバを使用して管理する必要があります。 複数のサーバを使用する方法の詳細については、**Specialized-guides › Large-deployments**を参照してください。

4.1. クライアント登録メソッド

クライアントをSUSE Managerサーバに登録する方法は複数あります。

- Saltクライアントの場合、SUSE ManagerのWeb UIを使用してクライアントを登録することをお勧めします。 詳細については、**Client-configuration › Registration-webui**を参照してください。
- プロセスをより詳細に制御したい場合、多数のクライアントを登録する必要がある場合、または従来のクライアントを登録している場合、ブートストラップスクリプトの作成をお勧めします。 詳細については、**Client-configuration › Registration-bootstrap**を参照してください。
- Saltクライアントで、さらに詳細にプロセスを制御するには、コマンド行でsingleコマンドを実行すると便利です。 詳細については、**Client-configuration › Registration-cli**を参照してください。

クライアントは、登録する前にSUSE Managerサーバと日時が正しく同期されている必要があります。

まず、アクティベーションキーを作成してから、ブートストラップスクリプトまたはコマンドラインメソッドを使用する必要があります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

SUSE Managerサーバをこのサーバ自体に登録しないでください。 SUSE Managerサーバは個別に管理するか、別のSUSE Managerサーバを使用して管理する必要がありま

> す。 複数のサーバを使用する方法の詳細については、**Specialized-guides › Large-deployments**を参照してください。

4.1.1. Web UIでクライアントを登録する

SUSE ManagerのWeb UIでクライアントを登録する方法はSaltクライアントでのみ動作します。

Web UIを使用してSaltクライアントをブートストラップしている場合、**Specialized-guides › Salt**を使用してクライアントでブートストラッププロセスを実行しています。 Salt SSHは、Salt Bundleおよびそれに含まれているPythonインタープリターを使用します。したがって、その他のPythonインタープリターをクライアントにインストールする必要はありません。

> Salt Bundleはブートストラップリポジトリに付属しているため、クライアントでブートストラッププロセスを開始する前にリポジトリを作成する必要があります。 シェルスクリプトは、クライアントのオペレーティングシステムを検出し、適切なブートストラップリポジトリからSalt Bundleを配備し、ブートストラップスクリプトと同じロジックを使用します。 詳細については、ブートストラップリポジトリの作成準備を参照してください。

> SUSE Managerサーバをこのサーバ自体に登録しないでください。 SUSE Managerサーバは個別に管理するか、別のSUSE Managerサーバを使用して管理する必要があります。 複数のサーバを使用する方法の詳細については、**Specialized-guides › Large-deployments**を参照してください。

プロシージャ: Web UIでクライアントを登録する

1. SUSE ManagerのWeb UIで、**システム › ブートストラップ**に移動します。
2. ［**ホスト**］フィールドに、ブートストラップするクライアントの完全修飾ドメイン名(FQDN)を入力します。
3. ［**SSHポート**］フィールドに、クライアントを接続してブートストラップするために使用するSSHポート番号を入力します。 デフォルトでは、SSHポートは**22**です。
4. ［**ユーザ**］フィールドに、クライアントにログインするユーザ名を入力します。 デフォルトでは、ユーザ名は**root**です。
5. SSHでクライアントをブートストラップするには、［**認証**］フィールドで、［**SSH機密鍵**］にチェックを付け、クライアントへのログインに使用するSSH秘密鍵をアップロードします。 SSH秘密鍵でパスフレーズが必要な場合、［**SSH機密鍵のパスフレーズ**］フィールドに入力します。パスフレーズがない場合には空白のままにします。
6. パスワードでクライアントをブートストラップするには、［**認証**］フィールドで、［**パスワード**］にチェックを付け、クライアントへのログインに使用するパスワードを入力します。
7. ［**アクティベーションキー**］フィールドで、クライアントのブートストラップに使用するソフトウェアチャンネルに関連付けられているアクティベーションキーを選択します。 詳細については、**Client-configuration › Activation-keys**を参照してください。
8. オプション: ［**プロキシ**］フィールドで、クライアントの登録先にするプロキシを選択します。

9. デフォルトでは、［**Disable SSH Strict Key Host Checking**］（SSH厳密キーホストの確認を無効にする）チェックボックスにチェックが付いています。 このチェックボックスにチェックが付いていると、ブートストラッププロセスは、手動認証なしでSSHホストキーを自動的に受け入れます。
10. オプション: ［**Manage System Completely via SSH**］（SSHでシステムを完全に管理する）チェックボックスにチェックを付けます。 このオプションにチェックを付けると、サーバへの接続にSSHを使用するようにクライアントは設定され、その他の接続方法は設定されません。
11. **[ブートストラップ]**をクリックして、登録を開始します。

ブートストラッププロセスが完了したら、クライアントは［**システム › システム一覧**］にリストされます。

SSH秘密鍵は、ブートストラッププロセス中のみ保存されます。 秘密鍵は、ブートストラップが完了するとすぐにSUSE Managerサーバから削除されます。

SUSE Managerを使用してクライアントに新しいパッケージまたは更新がインストールされると、エンドユーザライセンスアグリーメント(EULA)が自動的に受け入れられます。 パッケージのEULAを確認するには、Web UIでパッケージ詳細ページを開きます。

ローカルで割り当てられたリポジトリの取り扱い

SUSE Managerがサービスを提供しないクライアントにリポジトリを直接割り当てることは、一般的なユースケースではありません。 問題の原因になる可能性があります。 したがって、Saltを介してブートストラップすることで、ブートストラッププロセスの開始時にすべてのローカルリポジトリを無効にします。

その後、Highstateやパッケージのインストールを実行するなど、チャンネルの状態を使用するたびに、ローカルに割り当てられたすべてのリポジトリが再び無効になります。

クライアントで使用されるすべてのソフトウェアパッケージは、SUSE Managerがサービスを提供するチャンネルから取得される必要があります。 カスタムチャンネルの作成の詳細については、**Administration › Custom-channelsの カスタムチャンネル**を参照してください。

4.1.2. ブートストラップスクリプトを使用してクライアントを登録する

ブートストラップスクリプトでクライアントを登録すると、パラメータを制御できるようになり、同時に多数のクライアントを登録する必要がある場合に役立ちます。 このメソッドは、Saltクライアントと従来のクライアントの両方で動作します。

ブートストラップスクリプトを使用してクライアントを登録するには、まずブートストラップスクリプトのテンプレートを作成することをお勧めします。このテンプレートはその後コピーして変更できます。 作成したブートストラップスクリプトは、登録時にクライアントで実行され、必要なパッケージがすべてクライアントに展開されていることを確認します。 ブートストラップスクリプトに含まれているパラメータがあります。このパラメータによって、クライアントシステムを、アクティベーションキーやGPGキーなどそのベースチャンネルに割り当てることができるようになります。

リポジトリ情報を注意深く確認して、ベースチャンネルリポジトリと一致していることを確認することが重要です。 リポジトリ情報が正確に一致しないと、ブートストラップスクリプトは正しいパッケージをダウンロードできません。

すべてのクライアントにブートストラップリポジトリが必要です。製品が同期されると、SUSE Managerサーバ上で自動的に作成および再生成されます。 ブートストラップリポジトリには、クライアントにSaltをインストールするためのパッケージ、Saltまたは従来のクライアントを登録するためのパッケージが用意されています。 ブートストラップリポジトリの作成については、**Client-configuration › Bootstrap-repository**を参照してください。

デフォルトではopenSUSE Leap 15およびSLE 15はPython 3を使用します。Python 2に基づくブートストラップスクリプトは、openSUSE Leap 15システムおよびSLE 15システム用に再作成する必要があります。Python 2を使用してopenSUSE Leap 15システムまたはSLE 15システムを登録する場合、ブートストラップスクリプトは失敗します。

mgr-bootstrapでのブートストラップスクリプトの作成

mgr-bootstrapコマンドは、カスタムブートストラップスクリプトを生成します。 ブートストラップスクリプトは、SUSE Managerクライアントシステムによって初期登録と設定を簡素化するために使用されます。

引数**--activation-keys**および**--script**は唯一の必須の引数です。 SUSE Managerサーバでは、コマンドラインでrootとして、必須の引数を指定してコマンドを実行します。 **<ACTIVATION_KEYS>**および**<EDITED_NAME>**は使用する値に置き換えます。

```
mgr-bootstrap --activation-keys=<ACTIVATION_KEYS> --script=bootstrap-<EDITED_NAME>.sh
```

mgr-bootstrapコマンドには、ほかにもオプションがいくつかあり、特定のホスト名を設定したり、特定のGPGキーを設定したり、登録方法(従来、salt-minion、またはsalt-bundle)を設定したりできます。

詳細については、**mgr-bootstrap**のマニュアルページを参照するか、**mgr-bootstrap --help**を実行してください。

Web UIでのブートストラップスクリプトの作成

SUSE ManagerのWeb UIを使用して、編集できるブートストラップスクリプトを作成できます。

プロシージャ: ブートストラップスクリプトの作成

1. SUSE ManagerのWeb UIで、**管理 › マネージャ設定 › ブートストラップスクリプト**に移動します。
2. ［**SUSE Manager Configuration - Bootstrap**］（SUSEマネージャ設定 - ブートストラップ）ダイアログで、従来のクライアントをインストールしている場合、［**Saltを使用するブートストラップ**］チェックボックスのチェックを外します。 Saltクライアントでは、チェックを付けたままにします。
3. 必須フィールドには、前のインストール手順から取り出した値が事前に入力されています。 各設定の詳細については、**Reference › Admin**を参照してください。
4. **[更新]**をクリックしてスクリプトを作成します。
5. ブートストラップスクリプトが生成され、サーバの**/srv/www/htdocs/pub/bootstrap**ディレクトリに保存されます。 または、HTTPS経由でブートストラップスクリプトにアクセスできます。**<example.com>**をSUSE Managerサーバのホスト名に置き換えます。

```
https://<example.com>/pub/bootstrap/bootstrap.sh
```

ブートストラップスクリプトのSSLを無効にしないでください。 Web UIで［**Enable SSL**］（SSLの有効化）がチェックされていること、または設定 **USING_SSL=1** がブートストラップスクリプトに存在していることを確認してください。 SSLを無効にすると、登録プロセスでカスタムSSL証明書が必要です。

カスタム証明書の詳細については、**Administration › Ssl-certs**を参照してください。

ブートストラップスクリプトの編集

作成したブートストラップスクリプトのテンプレートをコピーして変更し、カスタマイズできます。 SUSE Managerで使用するためにブートストラップスクリプトを編集するときの最小要件は、アクティベーションキーを含めることです。 ほとんどのパッケージはGPGで署名されているため、信頼できるGPGキーをシステムで用意してインストールすることも必要です。

このプロシージャでは、アクティベーションキーの正確な名前を知っている必要があります。 **ホーム › 概要**に移動し、［**タスク**］ボックスで、［**アクティベーションキーの管理**］をクリックします。 チャンネル用に作成したすべてのキーがこのページに一覧表示されます。 ブートストラップスクリプトで使用するキーのフルネームを、キーフィールドに表示されているように正確に入力する必要があります。 アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

プロシージャ: ブートストラップスクリプトの変更

1. SUSE Managerサーバのコマンドラインでrootとして、ブートストラップディレクトリを次のように変更します。

```
cd /srv/www/htdocs/pub/bootstrap/
```

2. 各クライアントで使用するブートストラップスクリプトのテンプレートのコピーを2つ作成し、名前を変更します。

```
cp bootstrap.sh bootstrap-sles12.sh
cp bootstrap.sh bootstrap-sles15.sh
```

3. 変更するために**bootstrap-sles15.sh**を開きます。 次のテキストが表示されるまで下方にスクロールします。 ファイルに「**exit 1**」がある場合、その行の先頭にハッシュまたはポンド記号(#)を入力してコメントアウトします。 これによって、スクリプトがアクティブになります。 **ACTIVATION_KEYS=**フィールドにこのスクリプトのキーの名前を入力します。

```
echo "Enable this script: comment (with #'s) this block (or, at least just"
echo "the exit below)"
echo
#exit 1

# can be edited, but probably correct (unless created during initial install):
# NOTE: ACTIVATION_KEYS *must* be used to bootstrap a client machine.
ACTIVATION_KEYS=1-sles15
```

```
ORG_GPG_KEY=
```

4. 完了したら、ファイルを保存し、2つ目のブートストラップスクリプトでこの手順を繰り返します。

デフォルトでは、ブートストラップスクリプトは、Saltクライアントの**venv-salt-minion**がブートストラップリポジトリーにある場合にはこれをインストールしようとし、ブートストラップリポジトリーにSalt Bundleがない場合には**salt-minion**をインストールしようとします。 何らかの理由で**salt-minion**が必要な場合、Salt Bundleをインストールせずに**salt-minion**の使用を続けることができます。

詳細については、**Client-configuration › Contact-methods-saltbundle**を参照してください。

クライアントの接続

スクリプトの作成を完了したら、このスクリプトを使用してクライアントを登録できます。

プロシージャ: ブートストラップスクリプトの実行

1. SUSE Managerでrootとしてログインします。 コマンドプロンプトでブートストラップディレクトリに変更します。

```
cd /srv/www/htdocs/pub/bootstrap/
```

2. 次のコマンドを実行して、クライアントでブートストラップスクリプトを実行します。 **EXAMPLE.COM**をクライアントのホスト名に置き換えます。

```
cat bootstrap-sles15.sh | ssh root@EXAMPLE.COM /bin/bash
```

3. または、クライアントで次のコマンドを実行します。

```
curl -Sks https://server_hostname/pub/bootstrap/bootstrap-sles15.sh | /bin/bash
```

問題を回避するには、ブートストラップスクリプトが **bash**を使用して実行されていることを確認してください。

このスクリプトは、前に作成したリポジトリディレクトリにある必要な依存関係をダウンロードします。

4. スクリプトの実行が完了すると、クライアントが正しく登録されたかどうかを確認できます。そのためには、SUSE ManagerのWeb UIを開き、**システム › 概要**に移動して、新しいクライアントがリストされていることを確認します。

5. スクリプトを使用してSaltクライアントを登録する場合は、SUSE ManagerのWeb UIを開いて、**Salt › Keys**に移動し、クライアントキーを受け入れます。

SUSE Managerを使用してクライアントに新しいパッケージまたは更新がインストール

> されると、エンドユーザライセンスアグリーメント(EULA)が自動的に受け入れられます。 パッケージのEULAを確認するには、Web UIでパッケージ詳細ページを開きます。

4.1.3. コマンドラインで登録する(Salt)

Saltクライアントの手動登録

ほとんどの場合、Saltクライアントは、デフォルトのブートストラップメソッドで正確に登録されます。 ただし、Saltを使用してクライアントをSUSE Managerサーバに手動で登録できます。そのためには、クライアントでSalt Minionファイルを編集し、サーバの完全修飾ドメイン名(FQDN)を指定します。 このメソッドは、サーバで受信するポート4505および4506を使用します。 このメソッドではSUSE Managerサーバの設定は不要です。ただし、上記のポートを開いている必要があります。

> コマンドラインで従来のクライアントを登録することもできますが、その手順は長くなります。 この手順についてはここでは説明しません。 ブートストラップスクリプトプロシージャを使用して従来のクライアントを登録します。 詳細については、registration-bootstrap.pdfを参照してください。

このプロシージャでは、登録する前に**venv-salt-minion** (Salt bundle)または**salt-minion**パッケージをSaltクライアントにインストール済みである必要があります。 両方ともさまざまな場所で設定ファイルを使用しますが、ファイル名は同じままです。 systemdサービスファイル名は異なります。

> この方法でブートストラップを実行できるのは、クライアントツールチャンネルまたは公式のSUSEディストリビューションの一部として**salt-minion**を使用する場合のみです。

Salt Bundleの設定

Salt Bundle (venv-salt-minion)

- **/etc/venv-salt-minion/**
- **/etc/venv-salt-minion/minion**
- **/etc/venv-salt-minion/minion.d/NAME.conf**
- systemdサービスファイル: **venv-salt-minion.service**

Salt bundleの詳細については、**Client-configuration › Contact-methods-saltbundle**を参照してください。

プロシージャ: Salt Bundle設定ファイルでクライアントを登録する

1. Saltクライアントで**minion**設定ファイルを開きます。 設定ファイルは次の場所にあります。

```
/etc/venv-salt-minion/minion
```

または

```
/etc/venv-salt-minion/minion.d/NAME.conf
```

2. ファイルで、SUSE Managerサーバまたはプロキシのfqdnと、アクティベーションキー(存在する場合)を追加または編集します。以下にリストされている他の設定パラメータも追加します。

```
マスタ: SERVER.EXAMPLE.COM

grains:
  susemanager:
    activation_key: "<Activation_Key_Name>"

server_id_use_crc: adler32
enable_legacy_startup_events: False
enable_fqdns_grains: False
```

3. **venv-salt-minion**サービスを再起動します。

```
systemctl restart venv-salt-minion
```

4. SUSE Managerサーバで、新しいクライアントキーを受け入れます。**<client>**をクライアントの名前に置き換えます。

```
salt-key -a '<client>'
```

Salt Minionの設定

Salt Minion (salt-minion)

- **/etc/salt/**
- **/etc/salt/minion**
- **/etc/salt/minion.d/NAME.conf**
- systemdサービスファイル: **salt-minion.service**

プロシージャ: Salt Minion設定ファイルでクライアントを登録する

1. Saltクライアントで**minion**設定ファイルを開きます。 設定ファイルは次の場所にあります。

```
/etc/salt/minion
```

または

```
/etc/salt/minion.d/NAME.conf
```

2. ファイルで、SUSE Managerサーバまたはプロキシのfqdnと、アクティベーションキー(存在する場合)を追加または編集します。以下にリストされている他の設定パラメータも追加します。

```
マスタ: SERVER.EXAMPLE.COM

grains:
  susemanager:
```

```
    activation_key: "<Activation_Key_Name>"

server_id_use_crc: adler32
enable_legacy_startup_events: False
enable_fqdns_grains: False
```

3. **salt-minion**サービスを再起動します。

```
systemctl restart salt-minion
```

4. SUSE Managerサーバで、新しいクライアントキーを受け入れます。**<client>**をクライアントの名前に置き換えます。

```
salt-key -a '<client>'
```

Salt minion設定ファイルの詳細については、Salt Minionの設定を参照してください。

4.2. SUSEクライアントの登録

SUSE Linux EnterpriseおよびSUSE Linux Enterprise Server with Expanded Supportの各クライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーを作成済みである必要もあります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

SUSE Managerサーバをこのサーバ自体に登録しないでください。 SUSE Managerサーバは個別に管理するか、別のSUSE Managerサーバを使用して管理する必要があります。 複数のサーバを使用する方法の詳細については、**Specialized-guides › Large-deployments**を参照してください。

4.2.1. 以前の登録ステータスの確認

クライアントが以前にSCC、SMT、またはRMTに登録されていた場合は、クライアントをSUSE Managerサーバに移行する前に、**SUSEConnect**を使用してそのような登録を削除します。 まず古い登録を削除することを忘れないでください。 削除しないと、クライアントがSUSE ManagerでSUSEオペレーティングシステムの新しいバージョンに移行された後に、不要になったリポジトリがクライアントに追加されます。

SUSE Linux Enterprise 12または15クライアントを登録解除するには、クライアント上でrootとして次のコマンドを実行します。

```
SUSEConnect --de-register
SUSEConnect --cleanup
```

詳細なクリーンアップコマンドおよび検証手順の詳細については、https://www.suse.com/de-de/

support/kb/doc/?id=000019054を参照してください。

4.2.2. SUSE Linux Enterpriseクライアントの登録

このセクションでは、次のSUSE Linux Enterpriseオペレーティングシステムを実行しているクライアントの登録について説明します。

- SUSE Linux Enterprise Server 15 SP1
- SUSE Linux Enterprise Server 15 SP2
- SUSE Linux Enterprise Server 15 SP3
- SUSE Linux Enterprise Server 15 SP4
- SUSE Linux Enterprise Server 15 SP5
- SUSE Linux Enterprise Server 15 SP6
- SUSE Linux Enterprise Server 15 SP7

以下を含むすべてのSUSE Linux Enterprise製品を準備する際には、この章の手順を使用してください。

- SUSE Linux Enterprise Server for SAP
- SUSE Linux Enterprise Desktop
- SUSE Linux Enterprise
- SUSE Linux Enterprise Real Time

これらの手順は、古いSUSE Linux Enterpriseバージョンおよびサービスパックにも使用できます。

ソフトウェアチャンネルの追加

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

SUSE Linux EnterpriseクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

このプロシージャで必要な製品は次のとおりです。

表 17. SLE製品 - WebUI

OS Version	Product Name
SUSE Linux Enterprise Server 12 SP5	SUSE Linux Enterprise Server 12 SP5 x86_64
SUSE Linux Enterprise Server 15 SP1	SUSE Linux Enterprise Server 15 SP1 x86_64
SUSE Linux Enterprise Server 15 SP2	SUSE Linux Enterprise Server 15 SP2 x86_64
SUSE Linux Enterprise Server 15 SP3	SUSE Linux Enterprise Server 15 SP3 x86_64

OS Version	Product Name
SUSE Linux Enterprise Server 15 SP4	SUSE Linux Enterprise Server 15 SP4 x86_64
SUSE Linux Enterprise Server 15 SP5	SUSE Linux Enterprise Server 15 SP5 x86_64
SUSE Linux Enterprise Server 15 SP6	SUSE Linux Enterprise Server 15 SP6 x86_64
SUSE Linux Enterprise Server 15 SP7	SUSE Linux Enterprise Server 15 SP7 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 18. SLE製品 - CLI

OS Version	Base Channel
SUSE Linux Enterprise Server 12 SP5	sle-product-sles12-sp5-pool-x86_64
SUSE Linux Enterprise Server 15 SP1	sle-product-sles15-sp1-pool-x86_64
SUSE Linux Enterprise Server 15 SP2	sle-product-sles15-sp2-pool-x86_64
SUSE Linux Enterprise Server 15 SP3	sle-product-sles15-sp3-pool-x86_64
SUSE Linux Enterprise Server 15 SP4	sle-product-sles15-sp4-pool-x86_64
SUSE Linux Enterprise Server 15 SP5	sle-product-sles15-sp5-pool-x86_64
SUSE Linux Enterprise Server 15 SP6	sle-product-sles15-sp6-pool-x86_64
SUSE Linux Enterprise Server 15 SP7	sle-product-sles15-sp7-pool-x86_64

古い製品のチャンネル名を見つけるには、SUSE Managerサーバのコマンドプロンプトで root になり、**mgr-sync**コマンドを使用します:

```
mgr-sync list --help
```

次に、関心のある引数を指定します。たとえば、**channels**を指定します:

```
mgr-sync list channels [-c]
```

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

SUSE Linux Enterpriseチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.2.3. SLE Microクライアントの登録

このセクションでは、次のSLE Microオペレーティングシステム 5.1、5.2、5.3、5.4、および5.5 x86-64、ARM64、およびIBM Z (s390x)を実行しているクライアントの登録について説明します。

SLE Microは、エッジコンピューティング向けに構築された、極めて信頼性が高く軽量なオペレーティングシステムです。SUSE Linux Enterpriseのエンタープライズレベルの強化されたセキュリティおよびコンプライアンスコンポーネントを活用し、最新の不変の開発者向けOSプラットフォームと統合します。

SLE Microはトランザクション更新を使用します。トランザクション更新はアトミックであり(すべての更新はすべての更新が成功した場合にのみ適用されます)、ロールバックをサポートします。システムが再起動されるまで変更はアクティブ化されないため、実行中のシステムには影響しません。この情報は、**システム › 詳細 › 概要**サブタブに表示されます。

トランザクション更新と再起動の詳細については、https://documentation.suse.com/sles/html/SLES-all/cha-transactional-updates.htmlを参照してください。

ソフトウェアチャンネルの追加

SLE MicroクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

このプロシージャで必要な製品は次のとおりです。

表 19. SLE Micro製品 - WebUI

OSバージョン	製品名
SLE Micro 5.5 x86-64	SUSE Linux Enterprise Micro 5.5 x86_64
SLE Micro 5.5 ARM64	SUSE Linux Enterprise Micro 5.5 aarch64
SLE Micro 5.5 s390x	SUSE Linux Enterprise Micro 5.5 s390x
SLE Micro 5.4 x86-64	SUSE Linux Enterprise Micro 5.4 x86_64
SLE Micro 5.4 ARM64	SUSE Linux Enterprise Micro 5.4 aarch64
SLE Micro 5.4 s390x	SUSE Linux Enterprise Micro 5.4 s390x
SLE Micro 5.3 x86-64	SUSE Linux Enterprise Micro 5.3 x86_64
SLE Micro 5.3 ARM64	SUSE Linux Enterprise Micro 5.3 aarch64

OSバージョン	製品名
SLE Micro 5.3 s390x	SUSE Linux Enterprise Micro 5.3 s390x
SLE Micro 5.2 x86-64	SUSE Linux Enterprise Micro 5.2 x86_64
SLE Micro 5.2 ARM64	SUSE Linux Enterprise Micro 5.2 aarch64
SLE Micro 5.2 s390x	SUSE Linux Enterprise Micro 5.2 s390x
SLE Micro 5.1 x86-64	SUSE Linux Enterprise Micro 5.1 x86_64
SLE Micro 5.1 ARM64	SUSE Linux Enterprise Micro 5.1 aarch64
SLE Micro 5.1 s390x	SUSE Linux Enterprise Micro 5.1 s390x

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 20. SLE Micro製品 - CLI

OSバージョン	ベースチャンネル	更新チャンネル
SLE Micro 5.5 x86-64	sle-micro-5.5-pool-x86_64	sle-micro-5.5-updates-x86_64
SLE Micro 5.4 x86-64	sle-micro-5.4-pool-x86_64	sle-micro-5.4-updates-x86_64
SLE Micro 5.4 ARM64	sle-micro-5.4-pool-arm64	sle-micro-5.4-updates-arm64
SLE Micro 5.4 IBM Z (s390x)	sle-micro-5.4-pool-s390x	sle-micro-5.4-updates-s390x
SLE Micro 5.3 x86-64	sle-micro-5.3-pool-x86_64	sle-micro-5.3-updates-x86_64
SLE Micro 5.3 ARM64	sle-micro-5.3-pool-arm64	sle-micro-5.3-updates-arm64
SLE Micro 5.3 IBM Z (s390x)	sle-micro-5.3-pool-s390x	sle-micro-5.3-updates-s390x
SLE Micro 5.2 x86-64	suse-microos-5.2-pool-x86_64	suse-microos-5.2-updates-x86_64
SLE Micro 5.2 ARM64	suse-microos-5.2-pool-aarch64	suse-microos-5.2-updates-aarch64

OSバージョン	ベースチャンネル	更新チャンネル
SLE Micro 5.2 IBM Z (s390x)	suse-microos-5.2-pool-s390x	suse-microos-5.2-updates-s390x
SLE Micro 5.1 x86-64	suse-microos-5.1-pool-x86_64	suse-microos-5.1-updates-x86_64
SLE Micro 5.1 ARM64	suse-microos-5.1-pool-aarch64	suse-microos-5.1-updates-aarch64
SLE Micro 5.1 IBM Z (s390x)	suse-microos-5.1-pool-s390x	suse-microos-5.1-updates-s390x

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

クライアントの登録

SLE Microクライアントは、登録後に再起動が必要です。 登録が完了すると、再起動が自動的にスケジュールされますが、デフォルトの再起動マネージャのメンテナンスウィンドウに従って実行されます。 このウィンドウは、クライアントが登録されてから数時間後に表示される場合があります。 登録を高速化し、システムがシステムリストに表示されるようにするには、登録スクリプトの終了後にクライアントを手動で再起動することをお勧めします。

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

SLE Microシステムでブートストラップスクリプトを使用する場合は、スクリプトの証明書セクションに次のコンテンツがあることを確認します。

```
ORG_CA_CERT=RHN-ORG-TRUSTED-SSL-CERT
ORG_CA_CERT_IS_RPM_YN=0
```

ブートストラップスクリプトを直接編集して設定を追加するか、次のパラメータを使用してブートストラップスクリプトを作成します。

```
mgr-bootstrap --script=bootstrap-sle-micro.sh \
    --ssl-cert=/srv/www/htdocs/pub/RHN-ORG-TRUSTED-SSL-CERT
```

SLE Microの再起動

SLE Microはトランザクションシステムです。トランザクション更新は通常、いくつかの再起動方法をサポートしています。 SUSE Managerで管理されるシステムの再起動には、**systemd**を使用することをお勧めします。他の方法を使用すると、望ましくない動作が発生する可能性があります。

SUSE Managerでトランザクションシステムをブートストラップする場合、**systemd**が再起動方法(**REBOOT_METHOD**)として設定されます(システムがデフォルト設定の場合)。 このような設定により、SUSE Managerが再起動アクションを制御でき、必要に応じて再起動をすぐに実行したり、SUSE Managerでスケジュールしたりできます。

背景情報

デフォルトでは、クライアントのインストール中の再起動方法は**auto**に設定されています。 **auto**ブート方法では、サービスが実行されている場合、**rebootmgrd**を使用して、設定されたポリシーに従ってシステムを再起動します。 ポリシーにより、すぐに再起動することも、メンテナンスウィンドウ中に再起動することもできます。 詳細については、**rebootmgrd**(8)のマニュアルページを参照してください。 それ以外の場合で**rebootmgrd**が実行されていない場合、SUSE Managerは**systemctl reboot**を呼び出します。

systemdとは異なる方法を使用すると、望ましくない動作が発生する可能性があります。

4.2.4. SL Microクライアントの登録

このセクションでは、SL Microオペレーティングシステム 6.0 x86-64、ARM64、および IBM Z (s390x)を実行しているクライアントの登録について説明します。

SL Microは、エッジコンピューティング向けに構築された、極めて信頼性が高く軽量なオペレーティングシステムです。SUSE Linux Enterpriseのエンタープライズレベルの強化されたセキュリティおよびコンプライアンスコンポーネントを活用し、最新の不変の開発者向けOSプラットフォームと統合します。

SL Microはトランザクション更新を使用します。トランザクション更新はアトミックであり(すべての更新はすべての更新が成功した場合にのみ適用されます)、ロールバックをサポートします。システムが再起動されるまで変更はアクティブ化されないため、実行中のシステムには影響しません。この情報は、**システム › 詳細 › 概要**サブタブに表示されます。

トランザクション更新と再起動の詳細については、https://documentation.suse.com/sles/html/SLES-all/cha-transactional-updates.htmlを参照してください。

ソフトウェアチャンネルの追加

SL MicroクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

このプロシージャで必要な製品は次のとおりです。

表 21. SL Micro 6.1製品 - WebUI

OSバージョン	製品名
SL Micro 6.1 x86-64	SUSE Linux Micro 6.1 x86_64
SL Micro 6.1 ARM64	SUSE Linux Micro 6.1 arch64
SL Micro 6.1 s390x	SUSE Linux Micro 6.1 s390x

表 22. SL Micro 6.0製品 - WebUI

OSバージョン	製品名
SL Micro 6.0 x86-64	SUSE Linux Micro 6.0 x86_64
SL Micro 6.0 ARM64	SUSE Linux Micro 6.0 arch64
SL Micro 6.0 s390x	SUSE Linux Micro 6.0 s390x

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 23. SL Micro 6.1製品 - CLI

OSバージョン	ベースチャンネル
SL Micro [microversion] x86-64	sl-micro-6.1-pool-x86_64

表 24. SL Micro 6.0製品 - CLI

OSバージョン	ベースチャンネル
SL Micro 6.0 x86-64	sl-micro-6.0-pool-x86_64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルを

クリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

クライアントの登録

SL Microクライアントは、登録後に再起動が必要です。 登録が完了すると、再起動が自動的にスケジュールされますが、この再起動はデフォルトの再起動マネージャのメンテナンスウィンドウに従って実行されます。このウィンドウは、クライアントが登録されてから数時間後になる場合があります。迅速に登録してシステムをシステム一覧に表示するには、登録スクリプトの完了後にクライアントを手動で再起動することをお勧めします。

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

SL Microシステムでブートストラップスクリプトを使用する場合は、スクリプトの証明書セクションに次のコンテンツがあることを確認します。

```
ORG_CA_CERT=RHN-ORG-TRUSTED-SSL-CERT
ORG_CA_CERT_IS_RPM_YN=0
```

ブートストラップスクリプトを直接編集して設定を追加するか、次のパラメータを使用してブートストラップスクリプトを作成します。

```
mgr-bootstrap --script=bootstrap-sl-micro.sh \
    --ssl-cert=/srv/www/htdocs/pub/RHN-ORG-TRUSTED-SSL-CERT
```

SL Microの再起動

SL Microはトランザクションシステムです。トランザクション更新は通常、いくつかの再起動方法をサポートしています。 SUSE Managerで管理されるシステムの再起動には、**systemd**を使用することをお勧めします。他の方法を使用すると、望ましくない動作が発生する可能性があります。

SUSE Managerでトランザクションシステムをブートストラップする場合、**systemd**が再起動方法(**REBOOT_METHOD**)として設定されます(システムがデフォルト設定の場合)。 このような設定により、SUSE Managerが再起動アクションを制御でき、必要に応じて再起動をすぐに実行したり、SUSE Managerでスケジュールしたりできます。

背景情報

デフォルトでは、クライアントのインストール中の再起動方法は**auto**に設定されています。 **auto**ブート方法では、サービスが実行されている場合、**rebootmgrd**を使用して、設定されたポリシーに従ってシステムを再起動します。 ポリシーにより、すぐに再起動することも、メンテナンスウィンドウ中に再起動することもできます。 詳細については、**rebootmgrd**(8)のマニュアルページを参照してください。 それ以外の場合で**rebootmgrd**が実行されていない場合、SUSE Managerは**systemctl reboot**を呼び出します。

systemdとは異なる方法を使用すると、望ましくない動作が発生する可能性があります。

4.2.5. SUSE Liberty Linuxクライアントの登録

このセクションでは、SUSE Liberty Linuxオペレーティングシステムを実行している従来のクライアントおよびSaltクライアントの登録について説明します。 SUSE Liberty LinuxクライアントはRed Hat Enterprise LinuxまたはCentOSに基づいています。

SUSE Liberty Linuxクライアントは、SUSE Linux Enterprise Server with Expanded Support (SLESES)、Liberty、RES、またはRed Hat Expanded Supportとも呼ばれます。

SUSEによって提供されるSUSE Liberty Linuxソフトウェアチャンネルは、パッケージの更新のみを提供します。パッケージそのものは提供しません。 SUSE Liberty Linuxクライアントを登録するには、SUSE Liberty Linux製品(概要は以下を参照)を登録して必要なベースチャンネルを作成し、必要なRed HatまたはCentOSパッケージをカスタム子チャンネルとしてインポートする必要があります。 SUSE Liberty Linuxソフトウェアチャンネルで提供される更新を適用する前に、初期パッケージをRed HatまたはCentOSから直接取得する必要があります。

- SUSE Liberty LinuxリポジトリのURLはSUSE Customer Centerから入手できます
- パッケージおよびメタデータはSUSEから提供されます
- サポートされている製品については、サポートテーブルとリリースノートを参照してください。

ユーザは、Red HatまたはCentOSのベースメディアリポジトリおよびインストールメディアをアクセスできるようにする必要があります。

使用しているすべてのSUSE Liberty Linuxシステムに対してSUSEのサポートを取得する必要があります。

従来のクライアントはSUSE Liberty Linux 8または9では使用できません。 SUSE Liberty Linux 8および9クライアントはSaltクライアントとしてのみサポートされます。

ソフトウェアチャンネルの追加

SUSE Liberty Linuxクライアントでは、必要なパッケージの一部がRed Hat Enterprise LinuxまたはCentOSのインストールメディアに含まれています。 SUSE Liberty Linuxクライアントを登録するには、その前にこれらのパッケージをインストールする必要があります。

SUSE Liberty Linux製品はSUSE Customer Centerによって提供されます。 これには、クライアントツールパッケージも含まれています。

SUSE Liberty LinuxクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

2つの異なるチャンネルセットを選択する必要があります。一方はSUSE Liberty Linux用、他方はクライアントツール用です。

正しいSUSE Liberty Linuxチャンネルに関連付けられているアクティベーションキーが必要です。 アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

このプロシージャで必要な製品は次のとおりです。

表 25. ES製品 - WebUI

OSバージョン	製品名
SUSE Liberty Linux 7	SUSE Linux Enterprise Server with Expanded Support 7 x86_64
SUSE Liberty Linux LTSS 7	SUSE Linux Enterprise Server with Expanded Support LTSS 7 x86_64
SUSE Liberty Linux LTSS for Oracle 7	SUSE Linux Enterprise Server with Expanded Support LTSS for Oracle 7 x86_64
SUSE Liberty Linux 8	RHELまたはSLES ESまたはCentOS 8 Baseおよびおよび SUSE Linux Enterprise Server with Expanded Support 8 x86_64
SUSE Liberty Linux 9	RHELまたはSLES ESおよびLiberty 9 x86_64

SUSE Managerには、追加のソフトウェアが含まれているツールチャンネルが必要です。 このプロシージャは次のツールチャンネルを作成します。

表 26. ESツールチェンネル

OSバージョン	ベースチャンネル	ツールチャンネル
SUSE Liberty Linux LTSS 7	RHEL Expanded Support LTSS 7	RES-7-SUSE-Manager-Tools for x86_64 LBT7
SUSE Liberty Linux LTSS for Oracle 7	RHEL Expanded Support LTSS for Oracle 7	RES-7-SUSE-Manager-Tools for x86_64 LBTOL7
SUSE Liberty Linux 7	RHEL Expanded Support 7	RES7-SUSE-Manager-Tools x86_64
SUSE Liberty Linux 8	RHELまたはSLES ESまたはCentOS 8 Base	RES8-Manager-Tools-Pool for x86_64およびRES8-Manager-Tools-Updates for x86_64
SUSE Liberty Linux 9	RHELおよびLiberty 9 Base	EL9-Manager-Tools-Pool for x86_64およびEL9-Manager-Tools-Updates for x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

このプロシージャで必要なチャンネルは次のとおりです。

表 27. ESチャンネル - CLI

OSバージョン	ベースチャンネル	クライアントチャンネル	ツールチャンネル
SUSE Liberty Linux 7	rhel-x86_64-server-7	-	res7-suse-manager-tools-x86_64
SUSE Liberty Linux LTSS 7	res-7-ltss-updates-x86_64	-	res-7-suse-manager-tools-x86_64-lbt7
SUSE Liberty Linux LTSS for Oracle 7	res-7-ol-ltss-updates-x86_64	-	res-7-suse-manager-tools-x86_64-lbtol7
SUSE Liberty Linux 8	rhel8-pool-x86_64	-	res8-manager-tools-pool-x86_64
SUSE Liberty Linux 9	el9-pool-x86_64	-	el9-manager-tools-pool-x86_64

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE Manager

> のWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。
>
> または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。 CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

ベースメディアの追加

SUSE Liberty Linuxソフトウェアチャンネルは、パッケージの更新のみを提供します。パッケージそのものは提供しません。SUSE Liberty Linuxクライアントを登録するには、まずSUSE Liberty Linux製品(概要は以下を参照)を登録してベースチャンネルを作成し、必要なRed HatまたはCentOS パッケージをカスタム子チャンネルとしてインポートする必要があります。SUSE Liberty Linuxから更新を適用する前に、初期パッケージをRed HatまたはCentOSから直接取得する必要があります。重要な点として、Red Hatサブスクリプションを保持する必要がありませんが、移行に伴ってRed Hat への継続的な支払い義務が生じるかどうかを法務部門に問い合わせてください。必要なパッケージをすべて揃えるため、最小限のイメージまたはJeOSイメージではなく、完全なDVDイメージを使用してください。

SUSE Managerカスタムチャンネルを使用して、Red Hat Enterprise LinuxまたはCentOSのメディアを設定できます。 ベースメディアのすべてのパッケージは、子チャンネルにミラーリングする必要があります。

チャンネルの名前は自由に選択できます。

プロシージャ: カスタムチャンネルの作成

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › チャンネル**に移動します。
2. **[チャンネルの作成]**をクリックし、チャンネルに適切なパラメータを設定します。
3. ［**親チャンネル**］フィールドで、適切なベースチャンネルを選択します。
4. **[チャンネルの作成]**をクリックします。
5. 作成する必要があるすべてのチャンネルで繰り返します。 各カスタムリポジトリに1つのカスタムチャンネルが必要です。

該当するすべてのチャンネルとリポジトリを作成したことを確認できます。そのためには、**ソフトウェア › チャンネル一覧 › すべて**に移動します。

> Red Hat 9およびRed Hat 8クライアントでは、ベースチャンネルとAppStreamチャンネルの両方を追加します。 両方のチャンネルのパッケージが必要です。 両方のチャンネルを追加しないと、パッケージ不足のためブートストラップリポジトリを作成できません。

モジュラーチャンネルを使用している場合は、クライアントでPython3.6モジュールストリームを有効にする必要があります。Python 3.6を提供しない場合、**spacecmd**パッケージのインストールは失敗します。

プロシージャ: ベースメディアをカスタムチャンネルに追加する

1. SUSE Managerサーバのコマンドプロンプトでrootとしてベースメディアイメージを**/tmp/**ディレクトリにコピーします。
2. メディアコンテンツを含むディレクトリを作成します。 **<os_name>**を**sll7**、**sll8**、または**sll9**のいずれかに置き換えます。

```
mkdir -p /srv/www/htdocs/pub/<os_name>
```

3. イメージをマウントします。

```
mount -o loop /tmp/<iso_filename> /srv/www/htdocs/pub/<os_name>
```

4. 前に作成した子チャンネルにパッケージをインポートします。

```
spacewalk-repo-sync -c <channel-label> -u
file:///srv/www/htdocs/pub/<os_name>/<repopath>/
```

オプション: ベースチャンネルをコンテンツURLから追加する

または、Red Hat CDNまたはCentOSが提供するコンテンツURLにアクセスできる場合、カスタムリポジトリを作成してパッケージをミラーリングできます。

このプロシージャに必要な詳細は次のとおりです。

表 28. ESカスタムリポジトリ設定

オプション	パラメータ
リポジトリURL	Red Hat CDNまたはCentOSによって提供されるコンテンツURL
署名済みメタデータがあるかどうか	すべてのRed Hatエンタープライズリポジトリのチェックを外します
SSL CA証明書	**redhat-uep** (Red Hatのみ)
SSLクライアント証明書	**Entitlement-Cert-date** (Red Hatのみ)
SSLクライアントキー	**Entitlement-Key-date** (Red Hatのみ)

プロシージャ: カスタムリポジトリの作成

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › リポジトリ**に移動します。
2. **[リポジトリの作成]**をクリックし、リポジトリに適切なパラメータを設定します。
3. **[リポジトリの作成]**をクリックします。
4. 作成する必要があるすべてのリポジトリで繰り返します。

すべてのチャンネルを作成済みの場合、これらのチャンネルを、作成したリポジトリと関連付けできます。

プロシージャ: チャンネルのリポジトリとの関連付け

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › チャンネル**に移動し、関連付けるチャンネルをクリックします。
2. ［**リポジトリ**］タブに移動し、このチャンネルと関連付けるリポジトリにチェックを付けます。
3. **[リポジトリの更新]**をクリックし、チャンネルとリポジトリを関連付けます。
4. 関連付ける必要があるすべてのチャンネルとすべてのリポジトリを繰り返します。
5. オプション: ［**同期**］タブに移動し、このリポジトリの同期の繰り返しスケジュールを設定します。
6. **[今すぐ同期]**をクリックし、すぐに同期を開始します。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

SUSE Liberty Linuxチャンネルは非常に大きいことがあります。 最初のチャンネル同期は数時間かかる場合があります。

最初の同期が完了したとき、このチャンネルを使用する前にチャンネルを複製することをお勧めします。 この操作を実行すると、元の同期データのバックアップを作成できます。

SUSE Liberty Linuxクライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

Enterprise Linux (EL)クライアントをSUSE Liberty Linuxに移行する

Red Hat Enterprise LinuxやCentOSなどのEnterprise Linux (EL)クライアントがすでにSUSE Managerのminionとして登録されており、ユーザがそれをSUSE Liberty Linuxに移行したい場合は、再アクティベーションキーを使用して、移行を推進するアクティベーションキーを適用できます。

再アクティベーションキーの詳細については、client-configuration:activation-keys.pdfを参照してください。

再アクティベーションキーはminionごとにあり、Web UIまたはAPIを使用して生成できます。 詳細については、https://documentation.suse.com/suma/4.3/api/suse-manager/api/system.html#apidoc-system-obtainReactivationKey-loggedInUser-sidを参照してください。

クライアントを再アクティベーションするには、ユーザはクライアント上でブートストラップスクリプトを実行し、再アクティベーションキーを環境変数として渡します。 例:

```
REACTIVATION_KEY=<KEY> ./bootstrap_liberate9.sh
```

もう1つの方法は、**/etc/venv-salt-minon/minion.d/susemanager.conf**(または**/etc/salt-minon/minion.d/susemanager.conf**)にあるSaltクライアント設定ファイルに特別なフラグを追加することです。次のコンテンツを参照してください(このコンテンツを既存のコンテンツと結合します):

```
grains:
    susemanager:
        activation_key: "<KEY_ID>"
        management_key: "MINION_REACTIVATION_KEY"
```

susemanager.confファイルを変更した後、**salt-minion**サービスをSaltサーバ上で再起動する必要があります。 デフォルトでは以下を使用します。

```
systemctl restart venv-salt-minon
```

レガシのSaltの場合は、以下を使用します。

```
systemctl restart salt-minon
```

liberate formula

liberate formulaを使用してEnterprise Linux (EL)クライアントをSUSE Liberty Linuxに移行します。 詳細については、**Specialized-guides › Salt**を参照してください。

4.3. openSUSEクライアントの登録

openSUSEクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーを作成済みである必要もあります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

SUSE Managerサーバをこのサーバ自体に登録しないでください。 SUSE Managerサーバは個別に管理するか、別のSUSE Managerサーバを使用して管理する必要があります。 複数のサーバを使用する方法の詳細については、**Specialized-guides › Large-deployments**を参照してください。

4.3.1. openSUSE Leapクライアントの登録

このセクションでは、openSUSEオペレーティングシステムを実行しているSaltクライアントの登録について説明します。 SUSE Managerは、Saltを使用するopenSUSE Leap 15クライアントをサポートします。 従来のクライアントはサポートされていません。

ブートストラップは、リポジトリの設定やプロファイルの更新の実行など、openSUSEクライアントの起動および初期状態の実行のためにサポートされています。

ソフトウェアチャンネルの追加

openSUSEクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

現在サポートされているアーキテクチャは、「**x86_64**」と「**aarch64**」です。 サポートされている製品およびアーキテクチャの完全な一覧については、**Client-configuration › Supported-features**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

たとえば、「**x86_64**」アーキテクチャを使用する場合は、次の製品が必要です。

表 29. openSUSE製品 - WebUI

OSバージョン	製品名
openSUSE Leap 15.6	openSUSE Leap 15.6 x86_64
openSUSE Leap 15.5	openSUSE Leap 15.5 x86_64
openSUSE Leap 15.4	openSUSE Leap 15.4 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチ

ェックが付きます。　また、**include　recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。

3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 30. openSUSEチャンネル - CLI

OSバージョン	ベースチャンネル
openSUSE Leap 15.6	opensuse-leap-15.6-pool
openSUSE Leap 15.5	opensuse-leap-15.5-pool
openSUSE Leap 15.4	opensuse-leap-15.4-pool

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。　［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

openSUSEチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.4. AlmaLinuxクライアントの登録

AlmaLinuxクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーも作成しておく必要があります。

- アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。
- AlmaLinuxからSUSE Liberty Linuxへの移行の詳細については、client-configuration:clients-sleses.pdfを参照してください。

4.4.1. AlmaLinuxクライアントの登録

このセクションでは、AlmaLinuxオペレーティングシステムを実行しているSaltクライアントの登録について説明します。

- AlmaLinux repository URLs are available from SUSE Customer Center.
- パッケージおよびメタデータはSUSEではなくAlmaLinux OS Foundationから提供されます。
- サポートされている製品については、**Client-configuration › Supported-features-almalinux**にあるリリースノートとサポートテーブルを参照してください。

従来のクライアントはAlmaLinuxでは使用できません。 AlmaLinuxクライアントはSalt

クライアントとしてのみサポートされます。

AWSで作成するとき、AlmaLinuxインスタンスには、**/etc/machine-id**で常に同じ**machine-id** IDが割り当てられます。 インスタンスを作成した後に、必ず**machine-id**を再生成してください。 詳細については、**Administration › Troubleshooting**を参照してください。

ソフトウェアチャンネルの追加

AlmaLinuxクライアントのSUSE Managerへの登録は、**ターゲット**ポリシーで**適用**されるデフォルトのSELinux設定でテストされます。 SELinuxを無効にしてAlmaLinuxクライアントをSUSE Managerに登録する必要があります。

AlmaLinuxクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

現在サポートされているアーキテクチャは、「 **x86_64**」と「 **aarch64**」です。バージョン9では、ppc64leとs390xも追加でサポートされます。 サポートされている製品およびアーキテクチャの完全な一覧については、**Client-configuration › Supported-features**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

たとえば、「**x86_64**」アーキテクチャを使用する場合は、次の製品が必要です。

表 31. AlmaLinux製品 - WebUI

OSバージョン	製品名
AlmaLinux 9	AlmaLinux 9 x86_64
AlmaLinux 8	AlmaLinux 8 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 32. AlmaLinux チャンネル - CLI

OSバージョン	ベースチャンネル
AlmaLinux 9	almalinux9-x86_64
AlmaLinux 8	almalinux8-x86_64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

モジュラーチャンネルを使用している場合は、AlmaLinux 8クライアントでPython 3.6モジュールストリームを有効にする必要があります。Python 3.6を提供しない場合、**spacecmd**パッケージのインストールは失敗します。

上流のチャンネルとSUSE Managerチャンネルの間のAppStreamチャンネルで利用できるパッケージ数に不一致が発生する場合があります。 また、同時に別の場所で追加したチャンネルを比較すると、数値が異なる場合もあります。 AlmaLinuxでリポジトリを管理する方法が原因です。 AlmaLinuxでは新しいバージョンがリリースされると古いバージョンのパッケージが削除されますが、SUSE Managerでは経過年数に関係なくすべてのバージョンが保持されます。

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE ManagerのWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。

または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。 CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

アクティベーションキーの作成

AlmaLinuxチャンネルと関連付けられているアクティベーションキーを作成する必要があります。

アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアントの**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。 拡張サポートクライアントのGPGキーの場合も同様です。 ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートされ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、Saltファイルシステムに GPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys** pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。 **ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。 ［**GPGキーのURL**］に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。
2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

   ```
   uyuni-gpg-pubkey-0d20833e.key
   ```

   以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、

> ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

> クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

エラータの管理

AlmaLinuxクライアントを更新するとき、パッケージには更新に関するメタデータが含まれています。

4.5. Amazon Linuxクライアントの登録

Amazon LinuxクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーを作成済みである必要もあります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

4.5.1. Amazon Linuxクライアントの登録

このセクションでは、Amazon Linuxオペレーティングシステムを実行している従来のクライアントおよびSaltクライアントの登録について説明します。

> - Amazon Linux repository URLs are available from SUSE Customer Center.
> - パッケージおよびメタデータはSUSEではなくAmazonから提供されます。
> - サポートされている製品については、**Client-configuration › Supported-features-amazon**にあるリリースノートとサポートテーブルを参照してください。

従来のクライアントはAmazon Linux 2では使用できません。 Amazon Linux 2クライアントはSaltクライアントとしてのみサポートされます。

AWSで作成するとき、Amazon Linuxインスタンスには、**/etc/machine-id**で常に同じ**machine-id** IDが割り当てられます。 インスタンスを作成した後に、必ず**machine-id**を再生成してください。 詳細については、**Administration › Troubleshooting**を参照してください。

ソフトウェアチャンネルの追加

Amazon LinuxクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

現在サポートされているアーキテクチャは、「 **x86_64**」と「 **aarch64**」です。 サポートされている製品およびアーキテクチャの完全な一覧については、**Client-configuration › Supported-features**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

たとえば、「**x86_64**」アーキテクチャを使用する場合は、次の製品が必要です。

表 33. Amazon Linux製品 - WebUI

OSバージョン	製品名
Amazon Linux 2	Amazon Linux 2 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 34. Amazon Linuxチャンネル - CLI

OSバージョン	ベースチャンネル
Amazon Linux 2	amazonlinux2-core-x86_64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

アクティベーションキーの作成

Amazon Linuxチャンネルと関連付けられているアクティベーションキーを作成する必要があります。

アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアントの**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。 拡張サポートクライアントのGPGキーの場合も同様です。 ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートされ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、SaltファイルシステムにGPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys** pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。 **ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。 ［**GPGキーのURL**］に値file:///etc/

pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE　Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。
2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

```
uyuni-gpg-pubkey-0d20833e.key
```

以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.6. CentOSクライアントの登録

CentOSクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーも作成しておく必要があります。

- アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。
- CentOSからSUSE Liberty Linuxへの移行の詳細については、client-configuration:clients-sleses.pdfを参照してください。

4.6.1. CentOSクライアントの登録

このセクションでは、CentOSオペレーティングシステムを実行している従来のクライアントおよびSaltクライアントの登録について説明します。

- CentOS repository URLs are available from SUSE Customer Center.
- パッケージおよびメタデータはSUSEではなくCentOSから提供されます。
- サポートされている製品については、**Client-configuration › Supported-features-centos**にあるリリースノートとサポートテーブルを参照してください。

CentOSクライアントは、CentOSに基づいていて、SUSE Linux Enterprise Server with Expanded Support、RES、Red Hat、またはExpanded Supportとは関係がありません。 CentOSベースメディアリポジトリとCentOSインストールメディアへのアクセス管理、およびSUSE Managerサーバ のCentOSコンテンツデリバリネットワークへの接続は、ユーザが行います。

ソフトウェアチャンネルの追加

CentOSクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

現在サポートされているアーキテクチャは、「 **x86_64**」と「 **aarch64**」です。 サポートされている製品およびアーキテクチャの完全な一覧については、**Client-configuration › Supported-features**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

たとえば、「**x86_64**」アーキテクチャを使用する場合は、次の製品が必要です。

表 35. CentOS製品 - WebUI

OSバージョン	製品名
CentOS 7	CentOS 7 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 36. CentOSチャンネル - CLI

OSバージョン	ベースチャンネル
CentOS 7	centos7-x86_64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

モジュラーチャンネルを使用している場合は、クライアントでPython3.6モジュールストリームを有効にする必要があります。Python 3.6を提供しない場合、**spacecmd**パッケージのインストールは失敗します。

上流のチャンネルとSUSE Managerチャンネルの間のAppStreamチャンネルで利用できるパッケージ数に不一致が発生する場合があります。 また、同時に別の場所で追加したチャンネルを比較すると、数値が異なる場合もあります。 CentOSでリポジトリを管理する方法が原因です。 CentOSでは新しいバージョンがリリースされると古いバージョンのパッケージが削除されますが、SUSE Managerでは経過年数に関係なくすべてのバージョンが保持されます。

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE ManagerのWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。

> または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。 CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

アクティベーションキーの作成

CentOSチャンネルと関連付けられているアクティベーションキーを作成する必要があります。

アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアント

の**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。　拡張サポートクライアントのGPGキーの場合も同様です。　ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートされ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、Saltファイルシステムに GPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys**　pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。　**ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。　［**GPGキーのURL**］に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE　Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。

2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

```
uyuni-gpg-pubkey-0d20833e.key
```

以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

エラータの管理

CentOSクライアントを更新するとき、パッケージには更新に関するメタデータは含まれていません。 サードパーティのエラータサービスを使用してこの情報を取得できます。

ここで説明するサードパーティのエラータサービスであるCEFSは、コミュニティによって提供され、維持管理されます。 これはSUSEではサポートされていません。

CEFSの作成者は、パッチまたはエラータを、利便性向上を目指して努力ベースで提供していますが、これが正確であることや最新であることを保証していません。 つまり、パッチ日が正しくない場合があります。また、発行されたデータが1か月以上遅れて示されたことが少なくとも1回ありました。 このような場合の情報については、https://github.com/stevemeier/cefs/issues/28#issuecomment-656579382およびhttps://github.com/stevemeier/cefs/issues/28#issuecomment-656573607を参照してください。

> パッチデータに問題または遅れがあると、信頼できないパッチ情報がSUSE　Managerサーバにインポートされる場合があります。その結果、レポート、監査、CVEの更新、またはその他のパッチ関連の情報も誤りになります。セキュリティ関連の要件や証明書の条件に応じて、パッチデータを独立して確認する方法や、異なるオペレーティングシステムを選択する方法など、このサービスを使用する方法の代替方法を検討してください。

プロシージャ: エラータサービスのインストール

1. SUSE Managerサーバでコマンドプロンプトからrootとして**sle-module-development-tools**モジュールを追加します。

```
SUSEConnect --product sle-module-development-tools/15.2/x86_64
```

2. エラータサービスの依存関係をインストールします。

```
zypper in  perl-Text-Unidecode
```

3. **/etc/rhn/rhn.conf**で次の行を追加または編集します。

```
java.allow_adding_patches_via_api = centos7-x86_64-updates,centos7-x86_64,centos7-
x86_64-extras
```

4. Tomcatを再起動します。

```
systemctl restart tomcat
```

5. エラータスクリプト用のファイルを作成します。

```
touch /usr/local/bin/cent-errata.sh
```

6. 新しいファイルを編集してこのスクリプトを含め、必要に応じてリポジトリの詳細を編集します。 このスクリプトは、外部のエラータサービスからエラータの詳細をフェッチして展開し、詳細を発行します。

```
#!/bin/bash
mkdir -p /usr/local/centos
cd /usr/local/centos
rm *.xml
wget -c http://cefs.steve-meier.de/errata.latest.xml
wget -c https://www.redhat.com/security/data/oval/v2/RHEL7/rhel-7.oval.xml.bz2
bzip2 -d rhel-7.oval.xml.bz2
wget -c http://cefs.steve-meier.de/errata-import.tar
tar xvf errata-import.tar
chmod +x /usr/local/centos/errata-import.pl
export SPACEWALK_USER='<adminname>';export SPACEWALK_PASS='<password>'
/usr/local/centos/errata-import.pl --server '<servername>' \
--errata /usr/local/centos/errata.latest.xml  \
--include-channels=centos7-x86_64-updates,centos7-x86_64,centos7-x86_64-extras \
```

```
--publish --rhsa-oval /usr/local/centos/rhel-7.oval.xml
```

7. スクリプトを毎日実行するようcronジョブを設定します。

```
ln -s /usr/local/bin/cent-errata.sh /etc/cron.daily
```

このツールの詳細については、https://cefs.steve-meier.de/を参照してください。

4.7. Debianクライアントの登録

DebianクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーを作成済みである必要もあります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

SUSE Managerサーバをこのサーバ自体に登録しないでください。 SUSE Managerサーバは個別に管理する必要があります。

4.7.1. Debianクライアントの登録

このセクションでは、Debianオペレーティングシステムを実行しているSaltクライアントの登録について説明します。

DebianはSaltクライアントでのみサポートされています。 従来のクライアントはサポートされていません。

ブートストラップは、初期状態の実行およびプロファイルの更新のためにDebianクライアントで使用できます。

- Debian repository URLs are available from SUSE Customer Center.
- パッケージおよびメタデータはSUSEではなくDebianから提供されます。
- サポートされている製品については、**Client-configuration › Supported-features-debian**にあるリリースノートとサポートテーブルを参照してください。

DebianはSaltクライアントでのみサポートされています。 従来のクライアントはサポートされていません。

ブートストラップは、初期状態の実行およびプロファイルの更新のためにDebianクライアントで使用できます。

登録の準備

DebianクライアントをSUSE Managerサーバに登録するには、その前に準備が必要です。

- DNSが正しく設定されていることを確認し、クライアントのエントリを提供します。 または、適切なエントリを使用して、SUSE Managerサーバとクライアントの両方で**/etc/hosts**ファイルを設定できます。
- クライアントは、登録する前にSUSE Managerサーバと日時が同期されている必要があります。

ソフトウェアチャンネルの追加

DebianクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

このプロシージャで必要な製品は次のとおりです。

表 37. Debian製品 - WebUI

OSバージョン	製品名
Debian 11	Debian 11
Debian 12	Debian 12

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 38. Debianチャンネル - CLI

OSバージョン	製品名
Debian 11	debian-11-pool-amd64
Debian 12	debian-12-pool-amd64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

Debianチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

Debianクライアントをインストールするには、複数のGPGキーが必要な場合があります。

サードパーティのDebianリポジトリを同期する場合は、適切なGPGキーをサーバにインポートする必要があります。 GPGキーがない場合、同期は失敗します。

Debianリポジトリの場合、メタデータのみが署名されます。 したがって、ソフトウェアチャンネルのGPGキーをインポートする必要はありません。 パッケージはUyuniによって再署名されません。

SUSE Managerサーバにすでにインポートされているの GPGキーを確認するには、次のコマンドを実行します。

```
sudo gpg --homedir /var/lib/spacewalk/gpgdir --list-keys
```

新しいGPGキーをインポートするには、**--import**パラメータを使用します。

```
sudo gpg --homedir /var/lib/spacewalk/gpgdir --import <filename>.gpg
```

rootアクセス

Debianのrootユーザはデフォルトで SSHアクセスが無効になっています。

標準ユーザを使用してオンボードできるようにするには、**sudoers**ファイルを編集する必要があります。

プロシージャ: rootユーザアクセスの許可

1. クライアントで、**sudoers**ファイルを編集します。

   ```
   sudo visudo
   ```

   この行を**sudoers**ファイルの末尾に追加して**sudo**アクセス権をユーザに付与します。 Web UIでクライアントをブートストラップしているユーザの名前で**<user>**を置き換えます。

   ```
   <user>  ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2, /usr/bin/python3,
   /var/tmp/venv-salt-minion/bin/python
   ```

   このプロシージャによりrootアクセス権が付与されます。クライアントの登録に必要なパスワードは不要です。 クライアントは正常にインストールされると、root特権で実行されるため、アクセス権は不要です。 クライアントを正しくインストールした後、**sudoers**ファイルからこの行を削除することをお勧めします。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.8. Oracleクライアントの登録

Open Enterprise Serverクライアントを SUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーも作成しておく必要があります。

- アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

4.8.1. Open Enterprise Serverクライアントの登録

このセクションでは、Open Enterprise Serverオペレーティングシステムを実行しているクライアントの登録について説明します。

ソフトウェアチャンネルの追加

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

Open Enterprise Serverクライアントを SUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

このプロシージャで必要な製品は次のとおりです。

表 39. OES製品 - WebUI

OSバージョン	製品名
Open Enterprise Server 24.4	Open Enterprise Server 24.4 x86_64
Open Enterprise Server 23.4	Open Enterprise Server 23.4 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。　このプロシージャで必要なチャンネルは次のとおりです。

表 40. OES製品 - CLI

OSバージョン	ベースチャンネル	名前
Open Enterprise Server 24.4	oes24.4-pool-x86_64	OES24.4-Pool for x86_64
Open Enterprise Server 23.4	oes23.4-pool-x86_64"	OES23.4-Pool for x86_64

古い製品のチャンネル名を見つけるには、SUSE Managerサーバのコマンドプロンプトで root になり、**mgr-sync**コマンドを使用します:

```
mgr-sync list --help
```

次に、関心のある引数を指定します。たとえば、**channels**を指定します:

```
mgr-sync list channels [-c]
```

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

   ```
   mgr-sync add channel <channel_label_1>
   mgr-sync add channel <channel_label_2>
   mgr-sync add channel <channel_label_n>
   ```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

   ```
   mgr-sync sync --with-children <channel_name>
   ```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。　［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

SUSE Linux Enterpriseチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.9. Oracleクライアントの登録

Oracle LinuxクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーを作成済みである必要もあります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

Oracle LinuxからSUSE Liberty Linuxへの移行の詳細については、client-configuration:clients-sleses.pdfを参照してください。

4.9.1. Oracle Linuxクライアントの登録

このセクションでは、Oracle Linuxオペレーティングシステムを実行している従来のクライアントおよびSaltクライアントの登録について説明します。

- Oracle Linux repository URLs are available from SUSE Customer Center.
- パッケージおよびメタデータはSUSEではなくOracleから提供されます。
- Oracle Linuxベースメディアリポジトリはhttps://yum.oracle.com/から無料でダウンロードできます。
- サポートされている製品については、**Client-configuration › Supported-features-oracle**にあるリリースノートとサポートテーブルを参照してください。

Unbreakable Linux Network (ULN)リポジトリとSUSE Managerを直接同期することは現在サポートされていません。 ULNのOracleローカルディストリビューションを使用する必要があります。 ローカルULNミラーの設定の詳細については、https://docs.oracle.com/en/operating-systems/oracle-linux/software-management/sfw-mgmt-UseSoftwareDistributionMirrors.html#local-uln-mirror で提供されているOracleのドキュメントを参照してください。

従来のクライアントはOracle Linux 9および8では使用できません。 Oracle Linux 9およびOracle Linux 8クライアントはSaltクライアントとしてのみサポートされます。

ソフトウェアチャンネルの追加

Oracle Linuxクライアントを SUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

現在サポートされているアーキテクチャは、「 **x86_64**」と「 **aarch64**」です。 サポートされている製品およびアーキテクチャの完全な一覧については、**Client-configuration › Supported-features**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

たとえば、「**x86_64**」アーキテクチャを使用する場合は、次の製品が必要です。

表 41. Oracle製品 - WebUI

OSバージョン	製品名
Oracle Linux 9	Oracle Linux 9 x86_64
Oracle Linux 8	Oracle Linux 8 x86_64
Oracle Linux 7	Oracle Linux 7 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 42. Oracleチャンネル - CLI

OSバージョン	ベースチャンネル
Oracle Linux 9	oraclelinux9-x86_64
Oracle Linux 8	oraclelinux8-x86_64
Oracle Linux 7	oraclelinux7-x86_64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

モジュラーチャンネルを使用している場合は、クライアントでPython3.6モジュールストリームを有効にする必要があります。Python 3.6を提供しない場合、**spacecmd**パッケージのインストールは失敗します。

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE ManagerのWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。

または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。 CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

アクティベーションキーの作成

Oracle Linuxチャンネルと関連付けられているアクティベーションキーを作成する必要があります。

アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアントの**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。 拡張サポートクライアントのGPGキーの場合も同様です。 ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートされ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、Saltファイルシステムに GPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys** pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。 **ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。 ［**GPGキーのURL**］に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。
2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

   ```
   uyuni-gpg-pubkey-0d20833e.key
   ```

   以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、

> ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

> Oracle Linux 9およびOracle Linux 8クライアントの場合、以下を使用します
>
> ```
> ol8-gpg-pubkey-82562EA9AD986DA3.key
> ```
>
> Oracle Linux 7クライアントの場合、以下を使用します
>
> ```
> ol67-gpg-pubkey-72F97B74EC551F0A3.key
> ```

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

> Oracle Linux Linux用のLiberty 7 LTSSのブートストラップリポジトリを生成する際には、チャンネル**RES-7-BASE-Updates for x86_64 LBTOL7** (ラベル: **res-7-base-updates-x86_64-lbtol7**)がミラーリングされていることを確認してください。

4.10. Red Hatクライアントの登録

Red Hatコンテンツデリバリネットワーク(CDN)またはRed Hat更新インフラストラクチャ(RHUI)を使用してRed Hat Enterprise Linuxクライアントを SUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーも作成しておく必要があります。

- アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。
- Red Hat Enterprise LinuxからSUSE Liberty Linuxへの移行の詳細については、client-configuration:clients-sleses.pdfを参照してください。

4.10.1. CDNでRed Hat Enterprise Linuxクライアントを登録する

SUSE Linux Enterprise Server with Expanded Supportを使用するのではなく、Red Hat Enterprise Linuxクライアントを直接実行している場合、Red Hatソースを使用してパッケージを取得および更新する必要があ

ります。　このセクションでは、Red　Hatコンテンツデリバリネットワーク(CDN)を使用して、Red　Hat Enterprise Linuxオペレーティングシステムを実行している従来のクライアントおよびSaltクライアントを登録する方法について説明します。

従来のクライアントはRed　Hat　Enterprise　Linux 6および7でのみサポートされています。　Red　Hat Enterprise Linux 8およびRed Hat Enterprise Linux 9クライアントはSaltクライアントとしてサポートされています。

代わりにRed　Hat更新インフラストラクチャ(RHUI)を使用する方法については、**Client-configuration › Clients-rh-rhui**を参照してください。

Red　Hat　Enterprise　Linuxクライアントは、Red　Hatに基づいていて、SUSE　Linux Enterprise Server with Expanded Support、RES、またはSUSE Linux Enterprise Serverとは関係がありません。 Red Hatベースメディアリポジトリ とRHELインストールメディアへのアクセス管理、およびSUSE ManagerサーバのRed Hatコンテンツデリバリネットワークへの接続は、ユーザが行います。　使用しているすべてのRHELシステムに対してRed Hatのサポートを取得する必要があります。 これを実行しないと、Red Hatの条項に違反となる場合があります。

エンタイトルメントと証明書のインポート

Red Hatクライアントには、Red Hat認証局(CA)、エンタイトルメント証明書、およびエンタイトルメントキーが必要です。

エンタイトルメント証明書には、有効期限が埋め込まれていて、この期限はサポートサブスクリプションの期間と一致しています。 中断を回避するには、サポートサブスクリプション期間の終わりのたびにこのプロセスを繰り返す必要があります。

Red　Hatには、サブスクリプション割り当てを管理するためのサブスクリプションマネージャツールが用意されています。 このツールはローカルに実行され、インストール済みの製品およびサブスクリプションを追跡します。 クライアントは、サブスクリプションマネージャで登録して証明書を取得する必要があります。

Red Hatクライアントは、URLを使用してリポジトリを複製します。 URLは、Red Hatクライアントを登録した場所に応じて変わります。

Red Hatクライアントは次の3種類の方法で登録できます。

- redhat.comにあるRed Hatコンテンツデリバリネットワーク(CDN)
- Red Hatサテライトサーバ
- クラウドのRed Hat更新インフラストラクチャ(RHUI)

このガイドでは、Red　HatCDNに登録されるクライアントについて説明します。 リポジトリコンテンツの認可済みサブスクリプションを使用して、1つ以上のシステムがCDNに登録されている必要があります。

代わりにRed　Hat更新インフラストラクチャ(RHUI)を使用する方法については、**Client-configuration › Clients-rh-rhui**を参照してください。

クライアントシステムのサテライト証明書では、サテライトサーバおよびサブスクリプションが必要です。 サテライト証明書を使用するクライアントはSUSE Managerサーバではサポートされていません。

エンタイトルメント証明書には、有効期限が埋め込まれていて、この期限はサポートサブスクリプションの期間と一致しています。 中断を回避するには、サポートサブスクリプション期間の終わりのたびにこのプロセスを繰り返す必要があります。

Red Hatには、サブスクリプション割り当てを管理するためのサブスクリプションマネージャツールが用意されています。 このツールはクライアントシステムでローカルに実行され、インストール済みの製品およびサブスクリプションを追跡します。 サブスクリプションマネージャを使用してredhat.comを登録し、このプロシージャに従って証明書を取得します。

プロシージャ: クライアントをサブスクリプションマネージャに登録する

1. クライアントシステムのコマンドプロンプトで、サブスクリプションマネージャツールを使用して登録します。

```
subscription-manager register
```

プロンプトが表示されたら、Red Hatポータルのユーザ名とパスワードを入力します。

2. コマンドを実行します。

```
subscription-manager activate
```

3. SUSE Managerサーバがアクセスできる場所にエンタイトルメント証明書とキーをクライアントシステムからコピーします。

```
cp /etc/pki/entitlement/ /<example>/entitlement/
```

エンタイトルメント証明書とキーの両方ともファイル拡張子は**.pem**です。 キーにはファイル名にも**key**が含まれています。

4. Red Hat CA証明書ファイルをクライアントシステムから、エンタイトルメント証明書およびキーと同じWebの場所にコピーします。

```
cp /etc/rhsm/ca/redhat-uep.pem /<example>/entitlement
```

Red Hatクライアントでリポジトリを管理するには、CAおよびエンタイトルメント証明書をSUSE Managerサーバにインポートする必要があります。 この操作を実行するには、インポートプロシージャを3回実行して、3つのエントリを作成する必要があります。 エンタイトルメント証明書、エンタイトルメントキーおよびRed Hat証明書にそれぞれ1つずつです。

プロシージャ: 証明書をサーバにインポートする

1. SUSE ManagerサーバのWeb UIで、**システム › 自動インストール › GPGキーとSSLキー**に移動します。
2. **[格納されているキーまたは証明書の作成]**をクリックして、エンタイトルメント証明書用に次のパラメータを設定します。
   - ［**説明**］フィールドに**Entitlement-Cert-date**と入力します。
   - ［**タイプ**］フィールドで、**SSL**を選択します。
   - ［**アップロードするファイルの選択**］フィールドで、エンタイトルメント証明書を保存した場所をブラウズし、**.pem**証明書ファイルを選択します。
3. **[キーの作成]**をクリックします。
4. **[格納されているキーまたは証明書の作成]**をクリックして、エンタイトルメントキー用に次のパラメータを設定します。
   - ［**説明**］フィールドに**Entitlement-key-date**と入力します。
   - ［**タイプ**］フィールドで、**SSL**を選択します。
   - ［**アップロードするファイルの選択**］フィールドで、エンタイトルメントキーを保存した場所をブラウズし、**.pem**キーファイルを選択します。
5. **[キーの作成]**をクリックします。
6. **[格納されているキーまたは証明書の作成]**をクリックして、Red　Hat証明書用に次のパラメータを設定します。
   - ［**説明**］フィールドに**redhat-uep**と入力します。
   - ［**タイプ**］フィールドで、**SSL**を選択します。
   - ［**アップロードするファイルの選択**］フィールドで、Red Hat証明書を保存した場所をブラウズし、証明書ファイルを選択します。
7. **[キーの作成]**をクリックします。

ソフトウェアチャンネルの追加

Red HatクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

SUSE Managerサブスクリプションでは、SUSE Linux Enterprise Server with Expanded Supportのツールチャンネルを使用できます(Red Hat拡張サポートまたはRESとも呼ばれます)。 クライアントツールチャンネルを使用してブートストラップリポジトリを作成する必要があります。　このプロシージャは、Saltクライアントと従来のクライアントの両方に適用されます。

このプロシージャで必要な製品は次のとおりです。

表 43. Red Hat製品 - WebUI

OSバージョン	製品名
Red Hat 7	RHEL7 Base x86_64
Red Hat 8	RHELまたはSLES ESまたはCentOS 8 Base
Red Hat 9	RHELおよびLiberty 9 Base

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE ManagerのWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。

または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。 CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

カスタムリポジトリおよびチャンネルの準備

Red Hat CDNからソフトウェアをミラーリングするには、URLでCDNにリンクされているカスタムチャンネルおよびリポジトリをSUSE Managerに作成する必要があります。 Red Hatポータルでこれらの製品を正しく動作させるには、該当製品のエンタイトルメントが必要です。 サブスクリプションマネージャツールを使用して、ミラーリングするリポジトリのURLを取得できます。

```
subscription-manager repos
```

これらのリポジトリURLを使用して、カスタムリポジトリを作成できます。 クライアントを管理するために必要なコンテンツのみミラーリングできます。

Red Hatポータルに正しいエンタイトルメントがある場合、Red Hatリポジトリのカスタムバージョンのみ作成できます。

このプロシージャに必要な詳細は次のとおりです。

表 44. Red Hatカスタムリポジトリ設定

オプション	設定
リポジトリURL	Red Hat CDNによって提供されるコンテンツURL
署名済みメタデータがあるかどうか	すべてのRed Hatエンタイトルメントリポジトリのチェックを外します
SSL CA証明書	**redhat-uep**
SSLクライアント証明書	**Entitlement-Cert-date**
SSLクライアントキー	**Entitlement-Key-date**

プロシージャ: カスタムリポジトリの作成

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › リポジトリ**に移動します。
2. **[リポジトリの作成]**をクリックし、リポジトリに適切なパラメータを設定します。
3. **[リポジトリの作成]**をクリックします。
4. 作成する必要があるすべてのリポジトリで繰り返します。

このプロシージャで必要なチャンネルは次のとおりです。

表 45. Red Hatカスタムチャンネル

OSバージョン	ベース製品	ベースチャンネル
Red Hat 7	RHEL7 Base x86_64	rhel7-pool-x86_64
Red Hat 8	RHELまたはSLES ESまたはCentOS 8 Base	rhel8-pool-x86_64
Red Hat 9	RHELおよびLiberty 9 Base	el9-pool-x86_64

プロシージャ: カスタムチャンネルの作成

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › チャンネル**に移動します。
2. **[チャンネルの作成]**をクリックし、チャンネルに適切なパラメータを設定します。
3. ［**親チャンネル**］フィールドで、適切なベースチャンネルを選択します。
4. **[チャンネルの作成]**をクリックします。
5. 作成する必要があるすべてのチャンネルで繰り返します。　各カスタムリポジトリに1つのカスタムチャンネルが必要です。

該当するすべてのチャンネルとリポジトリを作成したことを確認できます。そのためには、**ソフトウェア › チャンネル一覧 › すべて**に移動します。

Red Hat 9およびRed Hat 8クライアントでは、ベースチャンネルとAppStreamチャンネルの両方を追加します。 両方のチャンネルのパッケージが必要です。 両方のチャンネル

> を追加しないと、パッケージ不足のためブートストラップリポジトリを作成できません。

モジュラーチャンネルを使用している場合は、クライアントでPython3.6モジュールストリームを有効にする必要があります。Python 3.6を提供しない場合、**spacecmd**パッケージのインストールは失敗します。

すべてのチャンネルを作成済みの場合、これらのチャンネルを、作成したリポジトリと関連付けできます。

プロシージャ: チャンネルのリポジトリとの関連付け

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › チャンネル**に移動し、関連付けるチャンネルをクリックします。
2. ［**リポジトリ**］タブに移動し、このチャンネルと関連付けるリポジトリにチェックを付けます。
3. **[リポジトリの更新]**をクリックし、チャンネルとリポジトリを関連付けます。
4. 関連付ける必要があるすべてのチャンネルとすべてのリポジトリを繰り返します。
5. オプション: ［**同期**］タブに移動し、このリポジトリの同期の繰り返しスケジュールを設定します。
6. **[今すぐ同期]**をクリックし、すぐに同期を開始します。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

> Red Hat Enterprise Linuxチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

プロシージャ: オプション: Salt状態を作成して設定ファイルを展開する

1. SUSE ManagerサーバのWeb UIで、**設定 › チャンネル**に移動します。

2. **[状態チャンネルの作成]**をクリックします。
   - ［**名前**］フィールドに**subscription-manager: disable yum plugins**と入力します。
   - ［**ラベル**］フィールドに**subscription-manager-disable-yum-plugins**と入力します。
   - ［**説明**］フィールドに**subscription-manager: disable yum plugins**と入力します。
   - ［**SLSコンテンツ**］フィールドは空白のままにします。
3. **[設定チャンネルの作成]**をクリックします
4. **[設定ファイルの作成]**をクリックします
   - ［**ファイル名/パス**］フィールドに**/etc/yum/pluginconf.d/subscription-manager.conf**と入力します。
   - ［**ファイルの内容**］フィールドに次のように入力します。

```
[main]
enabled=0
```

5. **[設定ファイルの作成]**をクリックします
6. ［**Saltファイルシステムパス**］フィールドの値をメモします。
7. 設定チャンネルの名前をクリックします。
8. ［**'init.sls' ファイルの表示/編集**］をクリックします
   - ［**ファイルの内容**］フィールドに次のように入力します。

```
configure_subscription-manager-disable-yum-plugins:
  cmd.run:
     - name: subscription-manager config --rhsm.auto_enable_yum_plugins=0
     - watch:
       - file: /etc/yum/pluginconf.d/subscription-manager.conf
   file.managed:
     - name: /etc/yum/pluginconf.d/subscription-manager.conf
     - source: salt:///etc/yum/pluginconf.d/subscription-manager.conf
```

9. **[設定ファイルの更新]**をクリックします。

Salt状態を作成して設定ファイルを展開するのプロシージャはオプションです。

プロシージャ: Red Hat Enterprise Linuxクライアントのシステムグループの作成

1. SUSE ManagerサーバのWeb UIで、**システム › システムグループ**に移動します。
2. **[グループの作成]**をクリックします。
   - ［**名前**］フィールドに**rhel-systems**と入力します。
   - ［**説明**］フィールドに**All RHEL systems**と入力します。
3. **[グループの作成]**をクリックします。

4. ［**状態**］タブをクリックします。
5. ［**設定チャンネル**］タブをクリックします。
6. 検索ボックスに**subscription-manager: disable yum plugins**と入力します。
7. **[検索]**をクリックして状態を表示します。
8. **Assign**列で状態のチェックボックスをクリックします。
9. **[変更点の保存]**をクリックします。
10. **[確認]**をクリックします。

RHELシステムをSUSE Managerに追加済みの場合、これらを新しいシステムグループに割り当て、highstateを適用します。

プロシージャ: システムグループをアクティベーションキーに追加する

RHELシステムで使用したアクティベーションキーを変更して、上記で作成したシステムグループに含めます。

1. SUSE ManagerサーバのWeb UIで、**システム › アクティベーションキー**に移動します。
2. RHELシステムで使用されるそれぞれのアクティベーションキーをクリックします。
3. ［**グループ**］タブ、［**参加**］サブタブに移動します。
4. ［**Select rhel-systems**］（RHELシステムを選択）にチェックを付けます。
5. **[選択されたグループに参加]**をクリックします。

クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアントの**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。 拡張サポートクライアントのGPGキーの場合も同様です。 ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートされ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、SaltファイルシステムにGPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys** pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。 **ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。 ［**GPGキーのURL**］に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。
2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

```
uyuni-gpg-pubkey-0d20833e.key
```

以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

Red Hatカスタムチャンネルの場合、［**GPGチェックの有効化**］フィールドを確認する場合は、［**GPGキーのURL**］フィールドに値を入力する必要があります。 Red Hat minionのディレクトリ**/etc/pki/rpm-gpg**に保存されているGPGキーのファイルURLを使用できます。

例: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

Red Hat製品署名キーの完全なリストについては、https://access.redhat.com/security/team/keyを参照してください。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.10.2. RHUIでRed Hat Enterprise Linuxクライアントを登録する

SUSE Linux Enterprise Server with Expanded Supportを使用するのではなく、Red Hat Enterprise Linuxクライアントを直接実行している場合、Red Hatソースを使用してパッケージを取得および更新する必要があります。

このセクションでは、Red Hat更新インフラストラクチャ(RHUI)を使用して、Red Hat Enterprise Linuxオペレーティングシステムを実行している従来のクライアントおよびSaltクライアントを登録する方法について

説明します。

従来のクライアントはRed Hat Enterprise Linux 7でのみ使用できます。 Red Hat Enterprise Linux 8およびRed Hat Enterprise Linux 9クライアントはSaltクライアントとしてサポートされています。

Amazon EC2などのパブリッククラウドでクライアントを実行している場合は、この方法を使用します。

RHUIをRed Hatコンテンツデリバリネットワーク(CDN)と組み合わせて使用して、Red Hat Enterprise Linuxサブスクリプションを管理できます。 Red Hat CDNの使用については、**Client-configuration › Clients-rh-cdn**を参照してください。

Red Hat Enterprise LinuxクライアントはRed Hatに基づいていて、SUSE Linux Enterprise Server with Expanded Support、RES、またはSUSE Linux Enterprise Serverとは関係ありません。

SUSE ManagerサーバのRed Hat更新インフラストラクチャへの接続はユーザが行います。 このRHUI証明書を使用して更新したすべてのクライアントは、正しくライセンス供与されている必要があります。クラウドプロバイダに確認し、詳細については、Red Hatのサービス条項を確認してください。

RHUIで登録されたRed Hat Enterprise Linuxクライアントの電源がオフになっている場合、Red Hatが証明書を無効と宣言する場合があります。 この場合、クライアントの電源を再度オンにするか、新しいRHUI証明書を取得する必要があります。

エンタイトルメントと証明書のインポート

以前は、証明書とエンタイトルメントデータマニュアルをSUSE Manager Serverにインポートする必要がありました。 SUSE PAYGインスタンスと同じメカニズムを使用して、このタスクを自動化しました。**Installation-and-upgrade › Connect-payg**も参照してください。

このガイドでは、Red Hat更新インフラストラクチャ(RHUI)に登録されるクライアントについて説明します。 リポジトリコンテンツの認可済みサブスクリプションを使用して、1つ以上のシステムがRHUIに登録されている必要があります。

代わりにRed Hatコンテンツデリバリネットワーク(CDN)を使用する方法については、**Client-configuration › Clients-rh-cdn**を参照してください。

クライアントシステムのサテライト証明書では、サテライトサーバおよびサブスクリプションが必要です。 サテライト証明書を使用するクライアントはSUSE Managerサーバではサポートされていません。

PAYG接続は、最新の認証データを取得するために、定期的にクライアントをチェックします。 クライアントが実行され続け、定期的に更新されることが重要です。 これが行われない場合、リポジトリの同期はある時点で認証エラーにより失敗します。

接続する前に、Red Hat 7インスタンスを更新してください。

Red Hat 9インスタンスを接続するには、暗号ポリシー**LEGACY**で設定する必要がありま

> す。 **sudo update-crypto-policies --set LEGACY**を実行して、それに応じて設定します。

Red Hat更新インフラストラクチャへの接続

プロシージャ: 新しいRed Hatインスタンスの接続

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › PAYG**に移動し、**[PAYGの追加]**をクリックします。
2. ページセクション［**PAYGの接続の説明**］から始めます。
3. ［**説明**］フィールドに、説明を追加します。
4. ページセクション［**インスタンスSSH接続データ**］に移動します。
5. ［**ホスト**］フィールドに、SUSE Managerから接続するインスタンスのDNSまたはIPアドレスを入力します。
6. ［**SSHポート**］フィールドに、ポート番号を入力するか、デフォルト値22を使用します。
7. ［**ユーザ**］フィールドに、クラウドで指定されているユーザ名を入力します。
8. ［**パスワード**］フィールドに、パスワードを入力します。
9. ［**SSH機密鍵**］フィールドに、インスタンスキーを入力します。
10. ［**SSH機密鍵のパスフレーズ**］フィールドに、キーパスフレーズを入力します。

> 認証キーは常にPEM形式である必要があります。

インスタンスに直接接続していないが、SSH要塞を介して接続している場合は、プロシージャ: SSH要塞接続データを追加するに進みます。

それ以外の場合は、プロシージャ: Red Hat接続の終了 に進みます。

プロシージャ: SSH要塞接続データを追加する

1. ページセクション［**要塞SSH接続データ**］に移動します。
2. ［**ホスト**］フィールドに、要塞のホスト名を入力します。
3. ［**SSHポート**］フィールドに、要塞のポート番号を入力します。
4. ［**ユーザ**］フィールドに、要塞のユーザ名を入力します。
5. ［**パスワード**］フィールドに、要塞のパスワードを入力します。
6. ［**SSH機密鍵**］フィールドに、要塞キーを入力します。
7. ［**SSH機密鍵のパスフレーズ**］フィールドに、要塞キーのパスフレーズを入力します。

プロシージャ: Red Hat接続の終了でセットアップを完了します。

プロシージャ: Red Hat接続の終了

1. 新しいRed Hat接続データの追加を完了するには、**[作成]**をクリックします。
2. PAYG接続データの［**詳細**］ページに戻ります。　更新された接続ステータスは、［**情報**］という名前の上部セクションに表示されます。
3. 接続ステータスは、［**管理 > セットアップウィザード > Pay-as-you-go**］画面にも表示されます。
4. インスタンスの認証データが正しい場合、［**ステータス**］列に「**資格情報が正常に更新されました**」と表示されます。

いずれかの時点で無効なデータが入力された場合、新しく作成されたインスタンスは［**管理 > セットアップウィザード > PAYG**］に表示され、［**ステータス**］列にエラーメッセージが表示されます。

サーバで認証データが利用可能になるとすぐに、接続されているインスタンスで利用可能なすべてのリポジトリにリポジトリが追加されました。　リポジトリは、［**ソフトウェア > 管理 > リポジトリ**］で確認できます。

Red Hat接続は、デフォルトで組織1が所有するカスタムリポジトリを作成します。 別の組織が自動生成リポジトリを所有する必要がある場合は、**/etc/rhn/rhn.conf**で**java.rhui_default_org_id**を設定します。

これはリポジトリを定義して更新するだけです。　管理対象クライアントにリポジトリを使用する場合は、ソフトウェアチャンネルを指定して、リポジトリを接続する必要があります。

ソフトウェアチャンネルの追加

Red HatクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

SUSE Managerサブスクリプションでは、SUSE Linux Enterprise Server with Expanded Supportのツールチャンネルを使用できます(Red Hat拡張サポートまたはRESとも呼ばれます)。 クライアントツールチャンネルを使用してブートストラップリポジトリを作成する必要があります。　このプロシージャは、Saltクライアントと従来のクライアントの両方に適用されます。

このプロシージャで必要な製品は次のとおりです。

表 46. Red Hat製品 - WebUI

OSバージョン	製品名
Red Hat 9	RHELおよびLiberty 9 Base
Red Hat 8	RHELまたはSLES ESまたはCentOS 8 Base
Red Hat 7	RHEL7 Base x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE ManagerのWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。

または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。 CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

カスタムチャンネルの準備

RHUIからソフトウェアをミラーリングするには、自動生成リポジトリにリンクされたカスタムチャンネルをSUSE Managerに作成する必要があります。

このプロシージャで必要なチャンネルは次のとおりです。

表 47. Red Hatカスタムチャンネル

OSバージョン	ベース製品	ベースチャンネル
Red Hat 9	RHEL and Liberty 9 Base	el9-pool-x86_64
Red Hat 8	RHEL or SLES ES or CentOS 8 Base	rhel8-pool-x86_64
Red Hat 7	RHEL7 Base x86_64	rhel7-pool-x86_64

プロシージャ: カスタムチャンネルの作成

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › チャンネル**に移動します。
2. **[チャンネルの作成]**をクリックし、チャンネルに適切なパラメータを設定します。
3. ［**親チャンネル**］フィールドで、適切なベースチャンネルを選択します。
4. **[チャンネルの作成]**をクリックします。
5. 作成する必要があるすべてのチャンネルで繰り返します。 各カスタムリポジトリに1つのカスタムチャンネルが必要です。

該当するすべてのチャンネルとリポジトリを作成したことを確認できます。そのためには、**ソフトウェア › チャンネル一覧 › すべて**に移動します。

Red Hat 9およびRed Hat 8クライアントでは、ベースチャンネルとAppStreamチャンネルの両方を追加します。 両方のチャンネルのパッケージが必要です。 両方のチャンネルを追加しないと、パッケージ不足のためブートストラップリポジトリを作成できません。

すべてのチャンネルを作成済みの場合、これらのチャンネルを、作成したリポジトリと関連付けできます。

プロシージャ: チャンネルのリポジトリとの関連付け

1. SUSE ManagerサーバのWeb UIで、**ソフトウェア › 管理 › チャンネル**に移動し、関連付けるチャンネルをクリックします。
2. ［**リポジトリ**］タブに移動し、このチャンネルと関連付けるリポジトリにチェックを付けます。
3. **[リポジトリの更新]**をクリックし、チャンネルとリポジトリを関連付けます。
4. 関連付ける必要があるすべてのチャンネルとすべてのリポジトリを繰り返します。
5. オプション: ［**同期**］タブに移動し、このリポジトリの同期の繰り返しスケジュールを設定します。
6. **[今すぐ同期]**をクリックし、すぐに同期を開始します。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

Red Hat Enterprise Linuxチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルは使用できないため、クライアントにチャンネルを割り当てるかどうかは、キーを信頼するかどうかによって決まります。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.11. Rocky Linuxクライアントの登録

Rocky LinuxクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーも作成しておく必要があります。

- アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。
- Rocky LinuxからSUSE Liberty Linuxへの移行の詳細については、client-configuration:clients-sleses.pdfを参照してください。

4.11.1. Rocky Linuxクライアントの登録

このセクションでは、Rocky Linuxオペレーティングシステムを実行しているSaltクライアントの登録について説明します。

従来のクライアントはRocky Linuxでは使用できません。 Rocky LinuxクライアントはSaltクライアントとしてのみサポートされます。

- Rocky Linux repository URLs are available from SUSE Customer Center.
- パッケージおよびメタデータはSUSEではなくRocky Enterprise Software

Foundationから提供されます。

- サポートされている製品については、 **Client-configuration › Supported-features-rocky**にあるリリースノートとサポートテーブルを参照してください。

Rocky LinuxクライアントのSUSE Managerへの登録は、**ターゲット**ポリシーで**適用**されるデフォルトのSELinux設定でテストされます。 Rocky LinuxクライアントをSUSE Managerに登録するために、SELinuxを無効にする必要はありません。

ソフトウェアチャンネルの追加

Rocky LinuxクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

現在サポートされているアーキテクチャは、「 **x86_64**」と「 **aarch64**」です。バージョン9では、ppc64leとs390xも追加でサポートされます。サポートされている製品およびアーキテクチャの完全な一覧については、**Client-configuration › Supported-features**を参照してください。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

たとえば、「**x86_64**」アーキテクチャを使用する場合は、次の製品が必要です。

表 48. Rocky Linux製品 - WebUI

OSバージョン	製品名
Rocky Linux 9	Rocky Linux 9 x86_64
Rocky Linux 8	Rocky Linux 8 x86_64

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 49. Rocky Linuxチャンネル - CLI

OSバージョン	ベースチャンネル
Rocky Linux 9	rockylinux9-x86_64
Rocky Linux 8	rockylinux8-x86_64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

上流のチャンネルとSUSE　Managerチャンネルの間のAppStreamチャンネルで利用できるパッケージ数に不一致が発生する場合があります。　また、同時に別の場所で追加したチャンネルを比較すると、数値が異なる場合もあります。　Rocky　Linuxでリポジトリを管理する方法が原因です。　Rocky　Linuxでは新しいバージョンがリリースされると古いバージョンのパッケージが削除されますが、SUSE　Managerでは経過年数に関係なくすべてのバージョンが保持されます。

AppStreamリポジトリにはモジュールパッケージが用意されています。 SUSE ManagerのWeb UIに正しくないパッケージ情報が表示されます。 Web UIまたはAPIを使用してモジュールリポジトリから直接インストールまたはアップグレードするようなパッケージ操作は実行できません。

または、Salt状態を使用してSaltクライアントでモジュラーパッケージを管理したり、クライアントで**dnf**コマンドを使用することもできます。　CLMの詳細については、**Administration › Content-lifecycle**を参照してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。

2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。　［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

アクティベーションキーの作成

Rocky Linuxチャンネルと関連付けられているアクティベーションキーを作成する必要があります。

アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

クライアントでGPGキーを信頼する

オペレーティング システムは、独自のGPGキーを直接信頼するか、少なくとも最小限のシステムでインストールされて出荷されます。 ただし、別のGPGキーで署名されたサードパーティのパッケージは手動で処理する必要があります。 クライアントは、GPGキーを信頼していなくても正常にブートストラップできます。 ただし、キーが信頼されるまで、新しいクライアントツールパッケージをインストールしたり、更新したりできません。

Saltクライアントは、ソフトウェアチャンネル用に入力されたGPGキー情報を使用して、信頼できるキーを管理するようになりました。 GPGキー情報を持つソフトウェアチャンネルがクライアントに割り当てられると、チャンネルが更新されるか、このチャンネルから最初のパッケージがインストールされるとすぐに、キーが信頼されます。

ソフトウェアチャンネルページのGPGキーのURLには、「空白」で区切られた複数のキーのURLを含めることができます。 ファイルURLの場合は、ソフトウェアチャンネルを使用する前に、GPGキーファイルをクライアントに配備する必要があります。

Red Hatベースのクライアントのクライアントツールチャンネル用GPG キーは、クライアントの**/etc/pki/rpm-gpg/**に配備され、ファイルURLで参照できます。 拡張サポートクライアントのGPGキーの場合も同様です。 ソフトウェアチャンネルがクライアントに割り当てられている場合にのみ、インポートされ、システムによって信頼されます。

Debianベースのシステムはメタデータのみに署名するため、単一チャンネルに追加のキーを指定する必要はありません。 **Administration › Repo-metadata**の「独自のGPGキーを使用する」で説明されているように、ユーザが独自のGPGキーを設定してメタデータに署名すると、そのキーの配備と信頼が自動的に実行されます。

ユーザ定義のGPGキー

ユーザは、クライアントに配備する独自のGPGキーを定義できます。

いくつかのpillarデータを提供し、SaltファイルシステムにGPGキーファイルを提供することで、自動的にクライアントに配備されます。

これらのキーは、RPMベースのオペレーティングシステムでは**/etc/pki/rpm-gpg/**に、Debianシステムでは**/usr/share/keyrings/**に配備されます。

キーを配備するクライアントのpillarキー**custom_gpgkeys**を定義し、キーファイルの名前を一覧にします。

```
cat /srv/pillar/mypillar.sls
custom_gpgkeys:
  - my_first_gpg.key
  - my_second_gpgkey.gpg
```

さらに、Saltファイルシステムでは、**gpg**という名前のディレクトリを作成し、**custom_gpgkeys** pillarデータで指定された名前のGPGキーファイルを保存します。

```
ls -la /srv/salt/gpg/
/srv/salt/gpg/my_first_gpg.key
/srv/salt/gpg/my_second_gpgkey.gpg
```

これでキーは**/etc/pki/rpm-gpg/my_first_gpg.key**および**/etc/pki/rpm-gpg/my_second_gpgkey.gpg**でクライアントに配備されます。

最後のステップでは、ソフトウェアチャンネルのGPGキーのURLフィールドにURLを追加します。 **ソフトウェア › 管理 › チャンネル**に移動し、変更するチャンネルを選択します。 ［**GPGキーのURL**］に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

ブートストラップスクリプトのGPGキー

プロシージャ: ブートストラップスクリプトを使用してクライアントでGPGキーを信頼する

1. SUSE Managerサーバのコマンドプロンプトで、**/srv/www/htdocs/pub/**ディレクトリの内容を確認します。 このディレクトリには、使用できるすべての公開鍵が含まれています。 登録クライアントに割り当てるチャンネルに適用するキーをメモします。
2. 関連するブートストラップスクリプトを開き、**ORG_GPG_KEY=**パラメータを見つけて、必要なキーを追加します。 次に例を示します。

   ```
   uyuni-gpg-pubkey-0d20833e.key
   ```

   以前保存したキーを削除する必要はありません。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

エラータの管理

Rocky Linuxクライアントを更新するとき、パッケージには更新に関するメタデータが含まれています。

4.12. Ubuntuクライアントの登録

UbuntuクライアントをSUSE Managerサーバに登録できます。 そのメソッドおよび詳細は、クライアントのオペレーティングシステムによって異なります。

始める前に、クライアントでSUSE Managerサーバと日時が正しく同期していることを確認してください。

アクティベーションキーを作成済みである必要もあります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

4.12.1. Ubuntuクライアントの登録

このセクションでは、Ubuntuオペレーティングシステムを実行しているクライアントの登録について説明します。

- Ubuntu repository URLs are available from SUSE Customer Center.
- パッケージおよびメタデータはSUSEではなくUbuntuから提供されます。
- Canonicalは、SUSE Managerを保証またはサポートしていません。
- サポートされている製品については、**Client-configuration › Supported-features-ubuntu**にあるリリースノートとサポートテーブルを参照してください。

UbuntuはSaltクライアントでのみサポートされています。 従来のクライアントはサポートされていません。

ブートストラップは、リポジトリの設定やプロファイルの更新の実行など、Ubuntuクライアントの起動およ

び初期状態の実行のためにサポートされています。　ただし、Ubuntuのrootユーザはデフォルトで無効になっているため、ブートストラップを使用するには、Pythonの**sudo**特権がある既存ユーザが必要です。

ソフトウェアチャンネルの追加

UbuntuクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

このプロシージャで必要な製品は次のとおりです。

表 50. Ubuntu製品 - WebUI

OSバージョン	製品名
Ubuntu 24.04	Ubuntu 24.04
Ubuntu 22.04	Ubuntu 22.04

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 51. Ubuntuチャンネル - CLI

OSバージョン	ベースチャンネル
Ubuntu 24.04	ubuntu-2404-amd64-main-amd64
Ubuntu 22.04	ubuntu-2204-amd64-main-amd64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**mgr-sync** コマンドを特定のチャンネルに対して実行します:

```
mgr-sync add channel <channel_label_1>
mgr-sync add channel <channel_label_2>
```

```
mgr-sync add channel <channel_label_n>
```

2. 同期は自動的に開始されます。 チャンネルを手動で同期する場合、次のコマンドを使用します。

```
mgr-sync sync --with-children <channel_name>
```

3. 続行前に、同期が完了していることを確認してください。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

Ubuntuチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

rootアクセス

UbuntuのrootユーザはデフォルトでSSHアクセスが無効になっています。

標準ユーザを使用してオンボードできるようにするには、**sudoers**ファイルを編集する必要があります。

この問題は、自己インストールされたバージョンのUbuntuで発生します。インストール時にデフォルトのユーザが管理特権を付与されている場合、**sudo**を使用して特権エスカレーションを実行するにはパスワードが必要です。クラウドインスタンスでは、**cloud-init**が**/etc/sudoers.d**の下にファイルを自動的に作成し、パスワードを必要とせずに**sudo**を介して特権エスカレーションを付与するため、この問題は発生しません。

rootユーザアクセスの許可

プロシージャ: rootユーザアクセスの許可

1. クライアントで、**sudoers**ファイルを編集します。

```
sudo visudo
```

この行を**sudoers**ファイルの末尾に追加して**sudo**アクセス権をユーザに付与します。 Web UIでクライアントをブートストラップしているユーザの名前で**<user>**を置き換えます。

```
<user>  ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2, /usr/bin/python3,
/var/tmp/venv-salt-minion/bin/python
```

このプロシージャによりrootアクセス権が付与されます。クライアントの登録に必要なパスワードは不要です。 クライアントは正常にインストールされると、root特権で実行されるため、アクセス権は不要です。 クライアントを正しくインストールした後、**sudoers**ファイルからこの行を削除することをお勧めします。

SSH経由でインストール時に作成されたユーザとしてブートストラップする

SSH経由でUbuntuクライアントをブートストラップするには、インストール時に作成されたユーザをsudoグループに追加する必要があります。 次にクライアントからブートストラップスクリプトを実行します。

プロシージャ: インストール時に作成されたユーザのsudoグループへの追加とSSH経由でのブートストラップ

1. クライアントで、rootとしてコマンドラインから実行します(**<username>**をインストール時に作成されたユーザの名前に置き換えます)。

```
sudo usermod -aG sudo <username>
```

2. コマンドラインからクライアントシステムをブートストラップします(**<SERVER_FQDN>**をSUSE Managerサーバの完全修飾ドメイン名に置き換えます)。

```
sudo su -
curl -Sks https://<SERVER_FQDN>/pub/bootstrap/bootstrap-script.sh | /bin/bash
```

Ubuntuは上記のように**sudo su -**を発行した後に、クライアントシステムのコマンドラインから実行される対応するUbuntuブートストラップスクリプトを使用してのみブートストラップできます。 Ubuntuではデフォルトでrootユーザが無効になっており、Web

UIがUbuntuのインストール時に作成されたユーザの特権昇格を許可していないため、SUSE Manager Web UI経由でブートストラップできません。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.12.2. Ubuntu 18.04クライアントの登録

SUSE Managerは、Saltを使用するUbuntu 20.04 LTSおよび22.04 LTSクライアントをサポートしています。 Ubuntu 20.04および22.04を実行しているSaltクライアントを登録する方法については、**Client-configuration › Clients-ubuntu**を参照してください。

Ubuntu 18.04はサポートが終了しました。

配布がサポート終了になると、サポートが廃止されたと見なされる3か月の猶予期間に入ります。 その期間が過ぎると、製品はサポート対象外と見なされます。 サポートは、努力ベースでのみ提供される場合があります。

サポート終了日の詳細については、https://endoflife.software/operating-systemsを参照してください。

- Ubuntu 18.04リポジトリのURLはSUSE Customer Centerで使用できません。
- パッケージおよびメタデータはSUSEではなくUbuntuから提供されます。
- サポートされている製品については、サポートテーブルとリリースノートを参照してください。
- Canonicalは、SUSE Managerを保証またはサポートしていません。

UbuntuはSaltクライアントでのみサポートされています。 従来のクライアントはサポートされていません。

ブートストラップは、リポジトリの設定やプロファイルの更新の実行など、Ubuntuクライアントの起動および初期状態の実行のためにサポートされています。 ただし、Ubuntuのrootユーザはデフォルトで無効になっているため、ブートストラップを使用するには、Pythonの**sudo**特権がある既存ユーザが必要です。

このセクションでは、Ubuntu 18.04 LTSオペレーティングシステムを実行しているSaltクライアントの登録について説明します。

ソフトウェアチャンネルの追加

UbuntuクライアントをSUSE Managerサーバに登録する前に、必要なソフトウェアチャンネルを追加して同

期する必要があります。

次のセクションでは、**x86_64**アーキテクチャに基づく説明が多いです。 必要に応じて他のアーキテクチャに置き換えてください。

このプロシージャで必要な製品は次のとおりです。

表 52. Ubuntu製品 - WebUI

OSバージョン	製品名
Ubuntu 18.04	Ubuntu 18.04

WebUIを使用してUbuntu 18.04チャンネルを追加する場合、Ubuntuチャンネル - CLIに記載されているように、CLIを使用してubuntu-1804-amd64-mainチャンネルおよびubuntu-1804-amd64-main-updatesチャンネルも追加する必要があります。

プロシージャ: ソフトウェアチャンネルの追加

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード › 製品**に移動します。
2. 検索バーを使用してクライアントのオペレーティングシステムおよびアーキテクチャに適切な製品を探し、適切な製品にチェックを付けます。 こうすることによって、すべての必須チャンネルに自動的にチェックが付きます。 また、**include recommended**トグルがオンになっている場合、すべての推奨チャンネルにもチェックが付きます。 矢印をクリックして関連製品の一覧を表示し、必要な追加製品にチェックが付いていることを確認します。
3. **[製品の追加]**をクリックし、製品の同期が完了するまで待機します。

または、コマンドプロンプトでチャンネルを追加できます。 このプロシージャで必要なチャンネルは次のとおりです。

表 53. Ubuntuチャンネル - CLI

OSバージョン	ベースチャンネル
Ubuntu 18.04	ubuntu-18.04-pool-amd64

プロシージャ: コマンドプロンプトからのソフトウェアチャンネルの追加

1. SUSE Manager サーバのコマンドプロンプトで root になり、**spacewalk-common-channels** コマンドを特定のチャンネルに対して実行します:

```
spacewalk-common-channels \
<base_channel_label> \
<child_channel_label_1> \
<child_channel_label_2> \
... <child_channel_label_n>
```

2. 自動同期がオフになっている場合は、チャンネルを同期します。

```
spacewalk-repo-sync -p <base_channel_label>
```

3. 続行前に、同期が完了していることを確認してください。

表 54. Ubuntuチャンネル - CLI

OSバージョン	メインチャンネル	セキュリティチャンネル	更新チャンネル
Ubuntu 18.04	ubuntu-1804-amd64-main	ubuntu-1804-amd64-main-security	ubuntu-1804-amd64-main-updates

このメソッドを使用して追加されたチャンネルはデフォルトでは定期的に同期されません。 SUSE Manager Web UIを使用して同期スケジュールを設定できます。 **ソフトウェア › 管理 › チャンネル**に移動し、追加したチャンネルをクリックし、**リポジトリ › 同期**サブタブを選択します。 同期スケジュールを毎日または毎週に設定し、**[スケジュール]**をクリックします。

spacewalk-common-channelsによって提供されるクライアントツールのチャンネルの提供元はUyuniです。SUSEではありません。

同期ステータスの確認

プロシージャ: Web UIからの同期の進捗状況の確認

1. SUSE ManagerのWeb UIで、**管理 › セットアップウィザード**に移動し、［**製品**］タブを選択します。 このダイアログには、同期中の各製品の完了バーが表示されます。
2. 代わりに、**ソフトウェア › 管理 › チャンネル**に移動し、リポジトリに関連付けられているチャンネルをクリックします。 ［**リポジトリ**］タブに移動し、［**同期**］をクリックし、［**同期状態**］をクリックします。

プロシージャ: コマンドプロンプトから同期の進捗状況を確認する

1. SUSE Managerサーバのコマンドプロンプトで、rootとして、**tail**コマンドを使用して同期ログファイルを確認します。

```
tail -f /var/log/rhn/reposync/<channel-label>.log
```

2. それぞれの子チャンネルは、同期の進捗中にそれぞれのログを生成します。 同期が完了したことを確認するには、ベースチャンネルと子チャンネルのログファイルをすべて確認する必要があります。

Ubuntuチャンネルは非常に大きいことがあります。 同期に数時間かかる場合があります。

GPGキーの管理

クライアントではGPGキーを使用して、ソフトウェアパッケージをインストールする前にパッケージ認証の確認が行われます。 信頼されているソフトウェアのみクライアントにインストールできます。

クライアントのセキュリティにとってGPGキーを信頼することは重要です。 必要かつ信頼できるキーを決定するのは管理者のタスクです。 GPGキーが信頼されていない場合、ソフトウェアチャンネルをクライアントに割り当てることはできません。

GPGキーの詳細については、**Client-configuration › Gpg-keys**を参照してください。

rootアクセス

UbuntuのrootユーザはデフォルトでSSHアクセスが無効になっています。

標準ユーザを使用してオンボードできるようにするには、**sudoers**ファイルを編集する必要があります。

この問題は、自己インストールされたバージョンのUbuntuで発生します。インストール時にデフォルトのユーザが管理特権を付与されている場合、**sudo**を使用して特権エスカレーションを実行するにはパスワードが必要です。クラウドインスタンスでは、**cloud-init**が**/etc/sudoers.d**の下にファイルを自動的に作成し、パスワードを必要とせずに**sudo**を介して特権エスカレーションを付与するため、この問題は発生しません。

プロシージャ: rootユーザアクセスの許可

1. クライアントで、**sudoers**ファイルを編集します。

```
sudo visudo
```

この行を**sudoers**ファイルの末尾に追加して**sudo**アクセス権をユーザに付与します。 Web UIでクライアントをブートストラップしているユーザの名前で**<user>**を置き換えます。

```
<user>  ALL=NOPASSWD: /usr/bin/python, /usr/bin/python2, /usr/bin/python3,
/var/tmp/venv-salt-minion/bin/python
```

このプロシージャによりrootアクセス権が付与されます。クライアントの登録に必要なパスワードは不要です。 クライアントは正常にインストールされると、root特権で実行されるため、アクセス権は不要です。 クライアントを正しくインストールした後、**sudoers**ファイルからこの行を削除することをお勧めします。

クライアントの登録

クライアントを登録するには、ブートストラップリポジトリが必要です。 デフォルトでは、ブートストラップリポジトリは自動的に作成され、すべての同期製品に対して毎日再生成されます。 次のコマンドを使用して、コマンドプロンプトからブートストラップリポジトリを手動で作成できます。

```
mgr-create-bootstrap-repo
```

クライアントの登録については、**Client-configuration › Registration-overview**を参照してください。

4.13. クライアントをプロキシに登録する

プロキシサーバは、Saltおよび従来のクライアントの両方のためにブローカおよびパッケージキャッシュとして動作できます。 クライアントをプロキシに登録する動作は、クライアントをSUSE Managerサーバに直接登録する動作に似ていますが、いくつかの相違点があります。

Web UI、コマンドラインにおけるコマンド、またはブートストラップスクリプトを使用してSaltクライアントをプロキシに登録するための情報が次の各セクションに記載されています。 また、ブートストラップスクリプトを使用して従来のクライアントを登録するための情報も含まれています。 クライアントをあるSUSE Managerプロキシから別のプロキシまたはSUSE Managerサーバに移動する方法もあります。

Web UI内では、Saltクライアントと従来のクライアントの両方に関する情報をプロキシページに示します。 **システム › システム一覧 › プロキシ**でプロキシの名前をクリックしてプロキシに接続するクライアントの一覧を表示し、［**詳細**］タブの［**プロキシ**］サブタブを選択できます。

システム › すべてでクライアントの名前をクリックしてSaltクライアントのチェーンされたプロキシの一覧を表示し、［**詳細**］タブの［**接続**］サブタブを選択できます。

4.13.1. プロキシ間でのクライアントの移動

登録プロセスを繰り返すことなく、SaltおよびSaltSSHプッシュクライアントをプロキシ間で移動することができます。

チェーンされたプロキシを移動することはできません。 チェーンされたプロキシを移動する代わりに、新しいプロキシを作成し、クライアントを移動して、古いプロキシを削除します。

従来のクライアントをプロキシ間で移動する場合、最初から登録プロセスを繰り返す必要があります。

手順: プロキシ間でSaltまたはSaltSSHプッシュクライアントを移動する

1. SUSE ManagerのWeb UIで、プロキシ間で移動するクライアントの［**システムの詳細**］ページに移動します。
2. ［**接続**］タブに移動します。次に［**プロキシの変更**］リンクをたどって、ドロップダウンメニューを表示します。
3. ［**新しいプロキシ**］ドロップダウンメニューから、クライアントの移動先のプロキシを選択し、**[プロキシの変更]**をクリックします。

手順: SSMで複数のSaltまたはSalt SSHプッシュクライアントをプロキシ間で移動する

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、移動するそれぞれのクライアントを確認します。クライアントがシステムセットマネージャに追加されます。
2. **システム › システムセットマネージャ**に移動し、［**その他 › プロキシ**］タブに移動します。

3. ［**新しいプロキシ**］ドロップダウンメニューから、クライアントの移動先のプロキシを選択し、**[プロキシの変更]**をクリックします。

system.changeProxy API呼び出しでも同じ機能を利用できます。

背景情報

この機能の効果は、通常のSaltクライアントとSalt SSH Pushクライアントで異なります。

通常のSaltクライアント

この機能は、Salt状態アクションをスケジュールします。これにより、**susemanager.conf** Saltクライアント設定ファイルの**master:**設定が新しいプロキシを指すように変更されます。次に、この機能はSaltクライアントを再起動します。

susemanager.confファイルを手動で編集して**master:**を変更しても同じ効果があり、この方法もサポートされています。

minionが再起動して新しいプロキシ経由で再接続すると、サーバはデータベース内のプロキシパスを更新し、チャンネルURLを更新するための別のアクションをスケジュールします。

Salt SSHプッシュクライアント

この機能はデータベース内のプロキシパスをただちに更新し、チャンネルURLを更新するための新しいアクションがスケジュールされます。

4.13.2. プロキシからサーバへのクライアントの移動

Saltクライアントをプロキシからサーバに移動する場合は、プロキシリストから**なし**を選択します。

従来のクライアントをサーバに移動する場合、最初から登録プロセスを繰り返す必要があります。

4.13.3. Web UIを使用してクライアントをプロキシに登録する

Web UIを使用してSaltクライアントをSUSE Managerプロキシに登録できます。

SLE以外のクライアント全般およびバージョン15より前のバージョンのSLEクライアントではブートストラップリポジトリが必要です。　ブートストラップリポジトリには、クライアントにSaltをインストールするためのパッケージ、Saltまたは従来のクライアントを登録するためのパッケージが用意されています。　ブートストラップリポジトリの作成については、**Client-configuration › Bootstrap-repository**を参照してください。

プロシージャ: Web UIを使用してクライアントをプロキシに登録する

1. SUSE ManagerのWeb UIで、**システム › ブートストラップ**に移動します。
2. ［**ホスト**］フィールドに、ブートストラップするクライアントの完全修飾ドメイン名(FQDN)を入力します。

3. ［**SSHポート**］フィールドに、クライアントを接続してブートストラップするために使用するSSHポート番号を入力します。 デフォルトでは、SSHポートは**22**です。
4. ［**ユーザ**］フィールドに、クライアントにログインするユーザ名を入力します。 デフォルトでは、ユーザ名は**root**です。
5. ［**Authentication Method**］（認証メソッド）フィールドで、クライアントのブートストラップに使用する認証メソッドを選択します。
   - パスワード認証の場合、［**パスワード**］フィールドに、パスワードを入力してクライアントにログインします。
   - SSH機密鍵認証の場合、秘密鍵と関連パスフレーズを入力します。 ブートストラッププロセスの実行が完了するまでの間のみ、この鍵は保存されます。
6. ［**アクティベーションキー**］フィールドで、クライアントのブートストラップに使用するソフトウェアチャンネルに関連付けられているアクティベーションキーを選択します。
7. ［**プロキシ**］フィールドで、登録先にするプロキシサーバを選択します。
8. デフォルトでは、［**Disable SSH Strict Key Host Checking**］（SSH厳密キーホストの確認を無効にする）チェックボックスにチェックが付いています。 このチェックボックスにチェックが付いていると、ブートストラッププロセスは、手動認証なしでSSHホストキーを自動的に受け入れます。
9. オプション: ［**Manage System Completely via SSH**］（SSHでシステムを完全に管理する）チェックボックスにチェックを付けます。 このオプションにチェックを付けると、サーバへの接続にSSHを使用するようにクライアントは設定され、その他の接続方法は設定されません。
10. **[ブートストラップ]**をクリックして、登録を開始します。

ブートストラッププロセスが完了したら、クライアントは［**システム › システム一覧**］にリストされます。

4.13.4. コマンドラインで登録する(Salt)

Web UIの代わりに、コマンドラインを使用して、Saltクライアントをプロキシに登録できます。 このプロシージャでは、登録する前にSaltパッケージをSaltクライアントにインストール済みである必要があります。 SLE 12ベースのクライアントでは、**Advanced Systems Management**モジュールもアクティブ化しておく必要があります。

コマンドラインで従来のクライアントを登録することもできますが、その手順は長くなります。 この手順についてはここでは説明しません。 ブートストラップスクリプトプロシージャを使用して従来のクライアントを登録します。 詳細については、client-proxy-script.pdfを参照してください。

プロシージャ: コマンドラインを使用してクライアントをプロキシに登録する

1. 次の場所にあるクライアント設定ファイルを選択します。

```
/etc/salt/minion
```

または

```
/etc/salt/minion.d/NAME.conf
```

これはminionファイルと呼ばれることもあります。

2. プロキシFQDNを **マスタ** としてクライアント設定ファイルに追加します。

```
master: PROXY123.EXAMPLE.COM
```

3. **salt-minion**サービスを再起動します。

```
systemctl restart salt-minion
```

4. サーバで、新しいクライアントキーを受け入れます。**<client>**をクライアントの名前に置き換えます。

```
salt-key -a '<client>'
```

4.13.5. ブートストラップスクリプトを使用して登録する(Saltと従来版)

ブートストラップスクリプトを使用してSUSE Managerを介してSaltクライアントおよび従来のクライアントを登録できます。 これは、SUSE Managerサーバでクライアントを登録する方法とほぼ同じです。 相違点は、コマンドラインツールを使用してSUSE Managerプロキシにブートストラップスクリプトを作成することです。 その後、ブートストラップスクリプトは、必要な情報をクライアントにすべて展開します。 ブートストラップスクリプトには、アクティベーションキーまたはGPGキーなどのパラメータが必要です。 これらのパラメータはそれぞれの設定によって決まります。

プロシージャ: ブートストラップスクリプトを使用してクライアントをプロキシに登録する

1. Web UIを使用してSUSE Managerサーバにクライアントアクティベーションキーを作成します。 詳細については、**Client-configuration › Activation-keys**を参照してください。
2. プロキシで、**mgr-bootstrap**コマンドラインツールをrootとして実行します。 必要に応じて、追加のコマンドラインスイッチを使用して。ブートストラップスクリプトを調整します。 Saltクライアントではなく従来のクライアントをインストールするには、**--traditional**スイッチを使用してください。

   使用できるオプションを表示するには、コマンドラインに**mgr-bootstrap --help**と入力します。

```
mgr-bootstrap --activation-keys=key-string
```

3. オプション: 結果として得られたブートストラップスクリプトを編集します。
4. クライアントで直接ブートストラップスクリプトを実行するか、または**ssh**を使用してプロキシからブートストラップスクリプトを実行します。 **<bootstrap>**をブートストラップスクリプトの名前に置き換え、 **<client.example.com>**をクライアントのホスト名に置き換えます。

```
cat <bootstrap> | ssh root@<client.example.com> /bin/bash
```

4.14. パブリッククラウドでのクライアントの登録

SUSE Managerサーバを設定すると、クライアントの登録を開始できます。

4.14.1. 製品の追加とリポジトリの同期

クライアントに対応する製品をすでに追加し、リポジトリをSUSE　Managerに同期していることを確認してください。　これは、クライアントの登録に使用されるブートストラップリポジトリを作成するために必要です。

詳細については、installation-and-upgrade:pubcloud-setup.pdfを参照してください。

4.14.2. オンデマンドイメージの準備

SUSEによって提供されるオンデマンドイメージから起動するインスタンスは自動的に登録され、更新されたインフラストラクチャおよびSUSE Linux Enterpriseモジュールはアクティブ化されます。 SUSE Managerクライアントとしてオンデマンドイメージを使用するには、使用を始める前にこの自動化を無効にする必要があります。

プロシージャ: オンデマンドイメージの準備

1. オンデマンドインスタンスにログインします。
2. コマンドプロンプトでrootとして、登録データとリポジトリを削除します。

```
registercloudguest --clean
```

3. 自動登録のトリガサービスを削除します。

```
systemctl disable guestregister.service
```

4. Microsoft Azureでは、無効にする必要がある追加サービスがあります。

```
systemctl disable regionsrv-enabler-azure.timer
```

SUSE ManagerをSUSE Customer Centerに登録する手順については、**Installation-and-upgrade › Server-setup**を参照してください。

4.14.3. クライアントの登録

SUSE ManagerのWeb UIで、**システム › ブートストラップ**に移動し、［**ホスト**］、［**SSHポート**］、［**ユーザ**］、および［**パスワード**］の各フィールドに入力します。　［**ホスト**］フィールドで安定版FQDNを使用していることを確認してください。別の有効期間が短いFQDNをパブリッククラウドで使用している場合、SUSE Managerではホストを検索できません。

従来のクライアントをブートストラップしようとしている場合、クライアントにログインしている間にサーバのホスト名を解決できることを確認してください。　サーバ

> のFQDNをクライアントの**/etc/hosts**ローカル解決ファイルに追加する必要があります。 サーバのローカルIPアドレスで**hostname -f**コマンドを使用していることを確認してください。

パブリッククラウドのイメージでは通常、ユーザ名とパスワードでSSHにログインできません。証明書でのみSSHにログインできます。 Web UIからブートストラップを使用する場合、ユーザ名とSSHキーによるSSHへのログインを有効にする必要があります。 この操作を実行するには、**システム › ブートストラップ**に移動し、認証メソッドを変更します。

クラウドプロバイダがMicrosoft Azureの場合、ユーザ名とパスワードでログインできます。 この操作を実行するには、AzureUserがrootとしてパスワードなしでコマンドを実行できる必要があります。 この操作を実行するには、**/etc/sudoers.d/waagent**ファイルを開き、次の行を追加または編集します。

```
AzureUser ALL=(ALL) NOPASSWD: ALL
```

> AzureUserがrootとしてパスワードなしでコマンドを実行できると、セキュリティ上のリスクが生じます。 この方法の使用はテストのみにしてください。 運用システムでは実行しないでください。

ブートストラッププロセスが正常に完了したら、クライアントは**システム › システム一覧**にリストされます。

- プロセスをより詳細に制御したい場合、多数のクライアントを登録する必要がある場合、または従来のクライアントを登録している場合、ブートストラップスクリプトを作成します。 詳細については、**Client-configuration › Registration-bootstrap**を参照してください。
- Saltクライアントで、さらに詳細にプロセスを制御するには、コマンド行でsingleコマンドを実行すると便利です。 詳細については、**Client-configuration › Registration-cli**を参照してください。
- パブリッククラウドイメージ(AWS AMIなど)から起動されたクライアントを登録する場合、追加の設定をして、相互に上書きしないようにする必要があります。 複製を登録する方法の詳細については、**Administration › Troubleshooting**を参照してください。

4.14.4. アクティベーションキー

アクティベーションキーは従来のクライアントとSaltクライアントで使用し、クライアントが正しいソフトウェアのエンタイトルメントを持ち、適切なチャンネルに接続して関連グループに加入するようします。 それぞれのアクティベーションキーは、キーを作成するときに設定できる組織にひもづけされます。

アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

4.14.5. Terraformによって作成されたクライアントの自動登録

Terraformによって作成された新しいクライアントはSUSE Managerに自動的に登録できます。 次の2つの登録方法があります。

- **cloud-init**ベースの登録

- リモート実行プロビジョナーベースの登録

cloud-initベースのクライアント登録

新しく作成された仮想マシンを自動的に登録するには、**cloud-init**を活用して登録することをお勧めします。このソリューションでは、ホストへのSSH接続を設定する必要がありません。 また、クライアントの作成に使用するツールに関係なく使用することができます。

ユーザは、マシンをSUSE Managerに自動的に登録するために、Terraformを使用してイメージを展開するときにユーザデータのセットを渡すことができます。 **user_data**はブートストラップ時にマシンを初めて起動したときのみ1回だけ実行されます。

cloud-initを使用してクライアントを登録する前にユーザは以下を設定する必要があります。

- ブートストラップスクリプト: 詳細については、**Client-configuration › Registration-bootstrap**を参照してください。
- アクティベーションキー: 詳細については、**Client-configuration › Activation-keys**を参照してください。

次のコマンドを実行すると、ブートストラップスクリプトがダウンロードされ、新しいマシン作成時にそのマシンが登録されます。 これを**cloud-init**設定に追加する必要があります。

```
curl -s http://hub-server.tf.local/pub/bootstrap/bootstrap-default.sh | bash -s
```

user_dataが更新されてプロビジョニングが変更されると必ず、Terraformはそのマシンを破棄してから新しいIPなどを使用して再作成します。

AWSでの**cloud-init**に関する詳細については、https://registry.terraform.io/providers/hashicorp/template/latest/docs/data-sources/cloudinit_configを参照してください。

cloud-initの例については、https://registry.terraform.io/providers/hashicorp/cloudinit/latest/docs/data-sources/cloudinit_config#example-usageを参照してください。

remote-execプロビジョナーベースの登録

新しく作成した仮想マシンの自動登録の2番目のソリューションではTerraformの**remote-exec**プロビジョナーを使用します。

remote-execプロビジョナーは新しく作成されたマシンとやり取りします。 これは、SSH接続を開き、そのマシンでコマンドを実行できます。

リモート実行プロビジョナーを使用してクライアントを登録するとき、ユーザは、Terraformを実行しているマシンが、新しい仮想マシン作成後にそのマシンにアクセスできることを確認する必要があります。

その他の要件は、[cloud-initベースのクライアントの登録]と同じです。

- ブートストラップスクリプト: 詳細については、**Client-configuration › Registration-bootstrap**を参照してください。
- アクティベーションキー: 詳細については、**Client-configuration › Activation-keys**を参照してください。

次のコマンドを実行すると、ブートストラップスクリプトがダウンロードされ、新しいマシン作成時にそのマシンが登録されます。 これは、実行するリモートコマンドとして定義する必要があります。

```
curl -s http://hub-server.tf.local/pub/bootstrap/bootstrap-default.sh | bash -s
```

remote-execプロビジョナーの詳細については、https://www.terraform.io/docs/provisioners/remote-exec.htmlを参照してください。

Chapter 5. クライアントのアップグレード

クライアントは、基盤となるオペレーティングシステムのバージョン設定システムを使用し、定期的なアップグレードが必要です。

SUSEオペレーティングシステムを使用するSCC登録クライアントの場合、SUSE ManagerのWeb UI内でアップグレードを実行できます。 サポートされているSUSE Linux Enterprise 15のアップグレードパスは、https://documentation.suse.com/sles/15-SP4/html/SLES-all/cha-upgrade-paths.htmlを参照してください。

SLE 12を実行しているクライアントをSLE 15にアップグレードするには、アップグレードは自動化されていますが、アップグレードを始める前に準備手順を実行する必要があります。 詳細については、**Client-configuration › Client-upgrades-major**を参照してください。

コンテンツライフサイクルマネージャを使用してクライアントのアップグレードを自動化することもできます。 詳細については、**Client-configuration › Client-upgrades-lifecycle**を参照してください。

サービスパックのアップグレード、openSUSE Leapのマイナーバージョンのアップグレード、openSUSE LeapからSUSE Linux Enterpriseへの移行などの製品の移行の詳細については、**Client-configuration › Client-upgrades-product-migration**を参照してください。

5.1. クライアント - メジャーバージョンのアップグレード

クライアントには、インストールされているオペレーティングシステムで利用できる最新のサービスパック(SP)があり、最新の更新がすべて適用されている必要があります。 システムが最新でありすべての更新が正しくインストールされていることを開始前に確認してください。

アップグレードは、YaSTおよびAutoYaSTによって制御されます。Zypperは使用しません。

5.1.1. マイグレーションの準備

クライアントをSLE 12からSLE 15 に移行する前に、次の作業を行う必要があります。

1. インストールメディアの準備
2. 自動インストールのディストリビューションの作成
3. アクティベーションキーの作成
4. AutoYaSTプロファイルのアップロード

プロシージャ: インストールメディア(SLE 15 SP2など)の準備

1. SUSE Managerサーバで、SLE 15 SP2インストールメディア用にローカルディレクトリを作成します。

```
mkdir -p /srv/images/sle15sp2
```

2. インストールソースでISOイメージをダウンロードし、ISOイメージをサーバにマウントします。

```
mount -o loop DVD1.iso /mnt/
```

3. マウントしたISOからローカルファイルシステムにすべてコピーします。

```
cp -r /mnt/* /srv/images/sle15sp2
```

4. コピーが完了したら、ISOイメージをアンマウントします。

```
umount /mnt
```

このイメージはUnified Installerであり、複数の自動インストールのディストリビューション用に使用できます。

プロシージャ: 自動インストールのディストリビューションの作成

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › ディストリビューション**に移動し、**[ディストリビューションの作成]**をクリックします。
2. ［**自動インストール可能なディストリビューションの作成**］セクションで、次のパラメータを使用します。
   - ［**ディストリビューションラベル**］セクションに、ディストリビューションの固有の名前を入力します。 半角の英字、数字、ハイフン、ピリオド、および下線のみを使用し、5文字以上にします。たとえば、**sles15sp2-x86_64**です。
   - ［**ツリーパス**］フィールドに、インストールソースへの絶対パスを入力します。 たとえば、**/srv/images/sle15sp2**です。
   - ［**ベースチャンネル**］フィールドで、**SLE-Product-SLES15-SP2-Pool for x86_64**を選択します。
   - ［**インストーラ生成**］フィールドで、［**SUSE Linux Enterprise 15**］を選択します。
   - ［**カーネルオプション**］フィールドに、インストールでブート時にカーネルに渡すオプションを入力します。 **install=**パラメータおよび**self_update=0 pt.options=self_update**パラメータはデフォルトで追加されます。
   - インストールしたシステムを初めてブートするときにカーネルに渡すオプションを［**カーネルの後のオプション**］セクションに入力します。
3. **[自動インストール可能なディストリビューションの作成]**をクリックして保存します。

古いSLE 12 ベースチャンネルから新しいSLE 15チャンネルに切り替えるには、アクティベーションキーが必要です。

プロシージャ: アクティベーションキーの作成

1. SUSE ManagerサーバのWeb UIで、**システム › アクティベーションキー**に移動し、［**キーの作成**］をクリックします。

2. キーの説明を入力します。
3. キーを入力するか空白のままにし、自動キーを生成します。
4. オプション: 使用量を制限する場合、［**使用量**］テキストフィールドに値を入力します。
5. **SLE-Product-SLES15-SP2-Pool for x86_64**ベースチャンネルを選択します。
6. オプション: ［**付属エンタイトルメント**］を選択します。 詳細については、https://documentation.suse.com/sles/15-SP4/html/SLES-all/article-modules.htmlを参照してください。
7. **[アクティベーションキーの作成]**をクリックします。
8. ［**子チャンネル**］タブをクリックし、必要なチャンネルを選択します。
9. **[キーの更新]**をクリックします。

5.1.2. 自動インストールプロファイルの作成

自動インストールプロファイルには、システムをインストールするために必要なインストールデータおよび設定データがすべて含まれています。 インストール完了後に実行するスクリプトを含めることもできます。 手始めに使用できるスクリプトのサンプルは、https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaSTを参照してください。

有効なAutoYaSTアップグレード設定については、

https://doc.opensuse.org/projects/autoyast/#CreateProfile-upgradeを参照してください。

プロシージャ: 自動インストールプロファイルの作成

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › プロファイル**に移動し、自動インストールプロファイルのスクリプトをアップロードします。

   手始めに使用できるスクリプトのサンプルは、

   https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaSTを参照してください。

2. ［**カーネルオプション**］フィールドに**autoupgrade=1**と入力します。

   Y2DEBUG=1オプションを含めることもできます。 デバッグ設定は不要ですが、問題が発生したときの調査に役立ちます。

   Azureクラウドで実行されているクライアントは、［**カーネルオプション**］に**textmode=1 console=ttyS0**を追加する必要があります。

3. 自動インストールプロファイルを貼り付けるか、またはファイルアップロードフィールドを使用します。
4. **[作成]**をクリックして保存します。
5. アップロードしたプロファイルで変数を設定する必要がある場合、**システム › 自動インストール › プロ**

ファイルに移動し、編集するプロファイルを選択し、［**変数**］タブに移動します。

次のフォーマットを使用して必要な変数を指定します。

```
<key>=<value>
```

5.1.3. 移行

自動インストールプロファイルで参照するチャンネルがすべて使用可能で完全に同期していることを開始前に確認してください。

/var/log/rhn/reposync/<channel-label>.logでミラーリングの進捗状況を監視できます。

プロシージャ: 移行

1. SUSE ManagerサーバのWeb UIで、［**システム**］に移動し、アップグレードするクライアントを選択します。
2. ［**プロビジョニング**］タブに移動し、アップロードした自動インストールプロファイルを選択します。
3. **[自動インストールをスケジュールしてから終了する]**をクリックします。 システムによって、必要なファイルがダウンロードされ、ブートローダのエントリが変更され、再起動され、アップグレードが開始されます。

クライアントは、SUSE Managerサーバと次に同期するときに、再インストールジョブを受け取ります。 再インストールジョブは、新しいカーネルパッケージおよびinitrdパッケージをフェッチします。 また、新しいカーネルパッケージおよびinitrdパッケージへのポインタを含む新しい**/boot/grub/menu.lst** (GRUB Legacy)または**/boot/grub2/grub.cfg** (GRUB 2)を書き込みます。

クライアントが次にブートするとき、grubを使用して、新しいカーネルとそのinitrdをブートします。 PXEブートはこのプロセス中に使用されません。

ジョブがフェッチされた約3分後に、クライアントは再起動するためにシャットダウンします。

Saltクライアントでは、移行が完了した後、**spacewalk/minion_script**スニペットを使用してクライアントを再登録します。

5.2. コンテンツライフサイクルマネージャを使用したアップグレード

管理するSUSE Linux Enterprise Serverクライアントが多数ある場合、コンテンツライフサイクルマネージャを使用してインプレースアップグレードを自動化できます。

5.2.1. アップグレードの準備

クライアントをアップグレードするには、その前に次の準備を行う必要があります。

- コンテンツライフサイクルプロジェクトの作成
- アクティベーションキーの作成
- 自動インストールのディストリビューションの作成
- 自動インストールプロファイルの作成

プロシージャ: コンテンツライフサイクルプロジェクトの作成

1. ディストリビューション用のコンテンツライフサイクルプロジェクトを作成します。

   詳細については、**Administration › Content-lifecycle**を参照してください。

2. プロジェクトの名前は、短いがわかりやすい名前にします。
3. ディストリビューションに必要なソースチャンネルモジュールをすべて含めます。
4. 必要に応じてフィルタを追加し、1つ以上の環境を設定します。

プロシージャ: アクティベーションキーの作成

1. ディストリビューション用のアクティベーションキーを作成します。

   詳細については、**Client-configuration › Activation-keys**を参照してください。

2. フィルタされたプロジェクトチャンネルのすべてがアクティベーションキーに含まれていることを確認します。

プロシージャ: 自動インストールのディストリビューションの作成

1. 移行するベースチャンネルごとに自動インストールのディストリビューションを作成します。

   詳細については、**Client-configuration › Autoinst-distributions**を参照してください。

2. コンテンツライフサイクルプロジェクトの名前を表すディストリビューションラベルを付けます。
3. ［**インストーラ生成**］フィールドで、使用しているSLESのバージョンを選択します。

プロシージャ: 自動インストールプロファイルの作成

1. アップグレードするディストリビューションおよびサービスパックごとに自動インストールプロファイルを作成します。

   詳細については、**Client-configuration › Autoinst-profiles**を参照してください。

2. Saltクライアントと従来のクライアントには異なるプロファイルを使用する必要があります。
3. プロファイルで変数を使用して、異なるライフサイクル環境を区別できます。

自動インストールプロファイルのサンプルについては、https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaSTを参照してください。

インプレースアップグレードを自動化するための自動インストールプロファイルでこれらの次の変数を使用します。

リスト 1. 例: 自動インストールプロファイルで使用する変数

```
registration_key=1-15sp1-demo-test
org=1
channel_prefix=15sp1-demo-test
distro_label=15sp1-demo-test
```

リスト 2. 例: 自動インストールプロファイルで使用するエントリ

```
 <listentry>
     <ask_on_error config:type="boolean">true</ask_on_error>
     <media_url>https://$redhat_management_server/ks/dist/child/$channel_prefix-sle-
module-web-scripting15-sp1-pool-x86_64/$distro_label</media_url>
     <name>$channel_prefix SLE-Module-Web-Scripting15-SP1 Pool for x86_64  </name>
     <product>Web Scripting Module 15 SP1 x86_64 Pool</product>
 </listentry>
```

5.2.2. アップグレード

サーバをアップグレードする準備ができたら、クライアントをプロビジョニングできます。

プロシージャ: クライアントのプロビジョニング

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、プロビジョニングするクライアントを選択してシステムセットマネージャに追加します。
2. **システム › システムセットマネージャ › 概要**に移動し、［**プロビジョニング**］タブをクリックします。
3. 使用する自動インストールプロファイルを選択します。

PXEを使用できるクライアントでは、そのクライアントをプロビジョニングするとすぐに移行が自動化されます。その他のすべてのクライアントでは、Cobblerを使用してアップグレードを実行できます。

プロシージャ: Cobblerを使用してクライアントをアップグレードする

1. コマンドプロンプトで、rootとして、利用できるCobblerプロファイルを確認します。

   ```
   cobbler profile list
   ```

2. 選択したプロファイルおよびディストリビューションでISOファイルを構築します。

   ```
   cobbler buildiso --iso=/tmp/SLE_15-sp1.iso --profiles=SLE_15-sp1:1:Example
   --distro=SLE_15-sp1
   ```

   CD-ROMを使用したクライアントのプロビジョニングの詳細については、**Client-configuration › Autoinst-cdrom**を参照してください。

5.3. 製品移行

Product migration allows you to upgrade SLE-based client systems from an Service Pack (SP) level to a later one. For example, you can migrate SUSE Linux Enterprise Server 15 SP1 to SUSE Linux Enterprise Server 15 SP2. You can also upgrade clients such as Red Hat Enterprise Linux or CentOS to SUSE Liberty Linux.

製品移行では、クライアント上で有効化され、移行される必要のあるすべてのリポジトリが有効である必要があります。有効でない場合、移行は中止されます。

製品の移行は、同じメジャーバージョン内でアップグレードするためのものです。 SUSE Linux Enterprise Server 12からSUSE Linux Enterprise Server 15への移行には、製品の移行は使用できません。 メジャーアップグレードの詳細については、**Client-configuration › Client-upgrades-major**を参照してください。

openSUSE Leapを新しいマイナーバージョンまたは対応するSUSE Linux Enterprise Server SPレベルに移行することもできます。次に例を示します。

- openSUSE Leap 15.1から15.2、または
- openSUSE Leap 15.1からSUSE Linux Enterprise Server 15 SP1、または
- openSUSE Leap 15.5からSUSE Linux Enterprise Server 15 SP5

SUSE Linux Enterprise Server 12以降では、SUSE Customer Centerがサービスパックを提供している場合、SUSEはサービスパックのスキップをサポートしています。 たとえば、SUSE Linux Enterprise Server 15からSP2にアップグレードできます。SP1はインストールされません。

サポートされているSUSE Linux Enterprise Serverアップグレードパスについては、https://documentation.suse.com/sles/15-SP4/html/SLES-all/cha-upgrade-paths.html#sec-upgrade-paths-supportedを参照してください。

移行中、SUSE Managerは、インストール前に必要なライセンス(EULA)を自動的に受け入れます。

5.3.1. 単一システムの移行

製品の移行を開始する前に:

- 保留中の更新やパッチがないことを確認してください。 クライアントシステムの**詳細 › 概要**ページの［ **システムステータス** ］を確認し、提供されているすべての更新またはパッチをインストールします。 クライアントシステムが最新でない場合、製品移行が失敗する可能性があります。
- ターゲット製品のすべてのチャンネルが完全に同期されていることを確認してください。 Web UIで同期ステータスを確認するには、**管理 › セットアップウィザード › 製品**ページに移動します。
- 万一に備えて、作業システムのバックアップを用意してください。 製品の移行にはロールバック機能はありません。 移行プロシージャが始まると、ロールバックできません。

プロシージャ: 単一システムの移行の実行

1. **システム › 概要**ページからクライアントを選択します。
2. クライアントのシステム詳細ページから、**ソフトウェア › 製品移行**タブに移動します。
3. ターゲットの移行パスを選択し、**[チャンネルの選択]**をクリックします。
4. ［**製品移行 - チャンネル**］ページから、正しいベースチャンネルを選択し、**必須の子チャンネル**および追加の**オプション子チャンネル**を含めます。
5. オプション: ［**ベンダー変更を許可する**］にチェックを付け、ベンダを変更したパッケージをインストールできるようにします。 チェックを付けると、移行の開始前に詳細を示す通知が表示されます。

openSUSE LeapをSUSE Linux Enterprise Serverに移行するには、［**ベンダー変更を許可する**］オプションにチェックを付ける必要があります。

6. チャンネルを正しく設定したら**[移行のスケジュール]**をクリックします。

5.3.2. 製品の大量移行

多数のクライアントを次のSPバージョンに移行する場合、SUSE Manager APIコールを使用できます。

spacecmdコマンドラインツールは、**system_scheduleproductmigration**サブコマンドを提供します。 このコマンドを使用して、多数のクライアントの次のマイナーバージョンへの移行をスケジュールできます。

製品の大量移行の実行

製品の大量移行操作は危険です。 プロセスは徹底的にテストする必要があります。 少なくとも、最初に予行演習を行ってください。

システムを意図せずにアップグレードしないように注意してください。

プロシージャ: 製品の大量移行の実行

1. 実行可能な移行ターゲットをリストし、移行するシステムIDをメモします。

```
spacecmd api -- system.listMigrationTargets -A 1000010001
```

2. それぞれのシステムIDに対して、**listMigrationTarget**を呼び出し、目的の製品が使用可能であることを確認します。
   - システムIDに使用可能なターゲットがある場合は、**system.scheduleProductMigration**を呼び出します。
   - 目的のターゲットを使用できない場合、そのシステムをスキップします。
3. 次のテンプレートを環境に合わせます。

```
target = '[....]'
basechannel = 'channel-label'
 system_ids = [1, 2, 3]

session = auth.login(user, pass)
```

```
for system in system_ids
  if system.listMigrationTargets(session, system).ident == target
    system.scheduleProductMigration(session, system, target, basechannel, [], False,
<now>)
  else
    print "Cannot migrate to requested target -- skipping system"
  endif
endfor
```

例: SLES 15 SP2からSLES 15 SP3

この例では、大量移行を容易にするためにグループが一時的に作成されます。

プロシージャ: 製品の大量移行グループの作成

1. SUSE ManagerのWeb UIで、**システム › システムグループ**に移動し、**[グループの作成]**をクリックします。
2. グループに**mpm-target-sles15sp3**という名前を付けます。
   - 同じベースチャンネルにサブスクライブしているシステムのみを、作成したグループに追加する必要があります。 この例では、**SLE-Product-SLES15-SP2-Pool for x86_64**にサブスクライブされているシステムのみをグループに追加する必要があります。

     グループへのクライアントの追加の詳細については、client-configuration:system-groups.pdf を参照してください。

プロシージャ: グループへのシステムの追加

1. 次のコマンドを実行して、グループ内のすべてのシステムのターゲットを取得します。

```
spacecmd -- system_listmigrationtargets group:mpm-target-sles15sp3
```

2. コマンドは「ID」の文字列を出力します。
   - **すべての** システムについて報告されるターゲットのみを選択してください。
   - 文字列は、他のコマンドの**MIGRATIONTARGET**の識別子です。

spacecmdサブコマンド**system_scheduleproductmigration**と**system_listmigrationtargets**は、グループの一部であるすべてのシステムをループしています。

グループに100台のシステムがある場合は、100個のスケジュールされたアクションが表示されます。

グループ内のすべてのシステムは同じ移行ターゲットをサポートする必要があります。

プロシージャ: 大量移行コマンドの実行

1. **system_scheduleproductmigration**コマンドの構文は次のとおりです。

```
spacecmd -- system_scheduleproductmigration <SYSTEM> <BASE_CHANNEL_LABEL> \
    <MIGRATION_TARGET> [options]
```

2. この例では、グループ**mpm-target-sles15sp3**内のすべてのシステムをSLES 12 SP2からSLES 15 SPにアップグレードするには、コマンドラインで次のように入力します。

```
spacecmd -- system_scheduleproductmigration group:mpm-target-sles15sp3 \
    sle-product-sles15-sp3-pool-x86_64 "[190,203,195,1242]" -d
```

必須の構文の説明

system_scheduleproductmigrationの構文の使用方法とオプションを表示するには、次のコマンドを実行します。

```
spacecmd system_scheduleproductmigration help
```

<SYSTEM>

この例では、作成したグループを使用して、そのグループからすべてのシステムを選択します。

```
group:mpm-target-sles15sp3
```

<BASE_CHANNEL_LABEL>

これは、ターゲットベースチャンネルのラベルです。 この場合、システムはSLES 15 SP3にアップグレードされており、ラベルは**sle-product-sles15-sp3-pool-x86_64**です。

現在ミラーリングされているすべてのベースチャンネルのリストを表示するには、次のコマンドを実行します。

```
spacecmd softwarechannel_listbasechannels.
```

現在のベースチャンネルで使用可能なターゲットでない限り、チャンネルにアップグレードできないことに注意してください。

<MIGRATION_TARGET>

グループ**group:mpm-target-sles15sp3**内のシステムのこの値を特定するには、次のコマンドを実行します。

```
spacecmd -- system_listmigrationtargets group:mpm-target-sles15sp3
```

MIGRATION_TARGETパラメータは、次の形式で渡す必要があります。角括弧の副次作用を防ぐために、シェルの引用符が必要であることに注意してください。

```
"[190,203,195,1242]"
```

オプション

- **-s** START_TIME
- **-d** 予行演習モード(実際には処理を行わない) (実際の移行の前に予行演習を行うことを推奨)
- **-c** (子チャンネル) (カンマ区切りで子チャンネルのラベルを指定する 。空白は使用しないこと)

  この場合、**-d**オプションが含まれています。このオプションは予行演習が成功した後に削除できます。

成功した場合、スケジュールされたシステムごとのコマンド出力は次のようになります。

```
Scheduling Product migration for system mpm-sles152-1
Scheduled action ID: 66
```

グループ内の特定のシステムのWeb UIで、アクション (この場合は予行演習)を追跡することもできます。 クライアントのシステム詳細ページから、**イベント › 履歴**に移動します。 予行演習中に障害が発生した場合は、システムを調査する必要があります。

すべて問題がなければ、コマンドから**-d**オプションを削除して、実際の移行を実行できます。

移行が完了したら、SUSE Manager Web UIからシステムを再起動できます。

Chapter 6. クライアントの削除

SUSE Managerサーバからクライアントを削除する必要がある場合、Web UIを使用して削除できます。 コマンドラインからクライアントを削除することもできます。 これらのプロシージャは、従来のクライアントとSaltクライアントの両方で動作します。

6.1. Web UIでクライアントを削除する

プロシージャ: クライアントの削除

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、削除するクライアントを選択します。
2. **[システムの削除]**をクリックします。
3. 詳細を確認し、**[プロファイルの削除]**をクリックして確認します。
4. Saltクライアントの場合、SUSE Managerは、追加の設定をクリーンアップしようとします。 クライアントに接続できない場合、削除をキャンセルするオプションや、設定ファイルをクリーンアップせずにクライアントを削除するオプションがあります。

システムセットマネージャを使用して複数のクライアントを削除することもできます。 システムセットマネージャの詳細については、**Client-configuration › System-set-manager**を参照してください。

従来のクライアントを削除した後にこれを自動的にクリーンアップすることはできません。 手動で行う必要があります。 さらに、Saltクライアントをクリーンアップしてもsaltが無効になり、可能な場合はサービスが停止するだけです。 パッケージはアンインストールされません。

通常、従来のクライアントを削除せずにこれをSaltクライアントに移行します。 Saltは、従来のクライアントがあることを自動的に検出し、必要な変更を行います。 ただし、従来のクライアントをすでに削除していて、Saltクライアントとして再度登録する場合は、**Administration › Troubleshooting**を参照してください。

6.2. コマンドラインでSaltクライアントを削除する(API コールを使用)

プロシージャ: サーバからのクライアントの削除

1. FQDN (完全修飾ドメイン名)を持つクライアントを削除します。

```
spacecmd system_delete FQDN
```

spacecmd system_deleteはSaltキーも削除します。

system_deleteは以下のオプションを提供します。

```
usage: system_delete [options] <SYSTEMS>
```

```
    options:
          -c TYPE - Possible values:
             *  'FAIL_ON_CLEANUP_ERR' - fail in case of cleanup error,
             *  'NO_CLEANUP' - do not cleanup, just delete,
             *  'FORCE_DELETE' - try cleanup first but delete server anyway
            in case of error
```

6.3. コマンドラインからのクライアントの削除

6.3.1. Saltクライアント

このプロセスはSUSE Managerクライアントのみを対象としています。SUSE Managerサーバ自体では実行しないでください。

Red Hat Enterprise Linux、Debian、またはクローンを実行しているクライアントでは、次のプロシージャを実行しないでください。 **zypper**の代わりに、**yum**、**dnf**、**apt**のような同等のパッケージャコマンドを使用します。

プロシージャ: SLES 12および15 Saltクライアントの削除

1. salt-minionサービスを停止します。

```
systemctl stop salt-minion
```

2. リポジトリと設定ファイルを削除します。

```
rm -f /etc/zypp/repos.d/susemanager\:bootstrap.repo
rm -f /etc/zypp/repos.d/susemanager\:channels.repo
rm -r /etc/sysconfig/rhn/
rm -r /etc/salt/
```

3. クライアントパッケージを削除します。

```
zypper rm salt salt-minion python*-salt sle-manager-tools-release
```

プロシージャ: Salt Bundleクライアント - 登録の手動クリーンアップ

1. 登録解除するには、次のコマンドを実行します。

```
systemctl stop venv-salt-minion
zypper rm -y venv-salt-minion sle-manager-tools-release
rm -f /etc/zypp/repos.d/susemanager\:bootstrap.repo
rm -f /etc/zypp/repos.d/susemanager\:channels.repo
rm -r /etc/venv-salt-minion/*
```

Salt Bundleについては、**Client-configuration › Contact-methods-saltbundle**を参照してください。

このプロセスはSUSE Managerクライアントのみを対象としています。SUSE Managerサーバ自体では実行しないでください。

Red Hat Enterprise Linux、Debian、またはクローンを実行しているクライアントに対して、文字通り次のプロシージャを実行しないでください。**zypper**の代わりに、**yum**、**dnf**、**apt**などの同等のパッケージャコマンドを使用してください。

プロシージャ: 従来のSLES 12および15クライアント - 手動クリーンアップ

1. osadサービスを停止します(使用されている場合)。

```
systemctl stop osad
```

SLES 12クライアントで、次のパッケージがインストールされている場合は削除します。 これは最初に試す必要があります(**osad**パッケージがインストールされていない場合は、コマンドラインに一覧表示しないでください)。

+

```
zypper rm spacewalksd spacewalk-check zypp-plugin-spacewalk \
spacewalk-client-tools osad python2-zypp-plugin-spacewalk \
python2-spacewalk-check python2-spacewalk-client-setup
```

1. SLES 15クライアントでは、次のパッケージがインストールされている場合は削除します。

```
zypper rm spacewalk-client-setup mgr-daemon spacewalk-check \
zypp-plugin-spacewalk mgr-osad python3-zypp-plugin-spacewalk \
python3-spacewalk-check python3-spacewalk-client-setup
```

2. 次の出力が表示されます。

```
Refreshing service 'spacewalk'.
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following packages are going to be REMOVED:
  spacewalk-check spacewalk-client-setup spacewalksd zypp
plugin-python osad

5 packages to remove.
After the operation, 301.0 KiB will be freed.
Continue? [y/n/?] (y):
```

上記のRPMパッケージはクライアント固有であるため、削除する必要があります。 これが失敗した場合は、手動で削除する必要があります。 **rpm -e** コマンドは、上記の **zypper rm** コマンドが失敗しない限り使用しないでください。

3. これが完了したら、/etc/sysconfig/rhn/systemidファイルを削除する必要があります。 このファイルはクライアントマシン上にのみ存在し、SUSE Managerへの登録に使用されます。

```
rm /etc/sysconfig/rhn/systemid
```

4. 設定されているspacewalkチャンネルは次のコマンドで削除する必要があります。

```
rm /etc/zypp/repos.d/spacewalk*
```

5. 最後に、リポジトリが適切に設定されていることを確認します。 サーバでそれらを更新し、一覧表示します。

```
zypper ref -s
zypper lr
```

spacewalkを指すリポジトリがまだ存在する場合は、次のコマンドで削除します。

```
zypper repos -d
zypper removerepo <ID of the repo in the output from previous command>
```

Chapter 7. クライアントの操作

クライアントの登録、アップグレード、または削除に加えて、他の操作を実行できます。

SUSE　Managerクライアントは、システムセットマネージャ、システムグループまたは設定管理を使用して、個別に管理またはグループに編成できます。

SUSE Manager Web UI を使用して、カスタム システム情報の取得、設定のスナップショットの管理、またはクライアントの電源オン、電源オフ、および再起動を行うことができます。

このセクションでは、これらの各操作について詳細に説明します。

7.1. パッケージ管理

クライアントは、パッケージを使用してソフトウェアをインストール、アンインストール、およびアップグレードします。

クライアントでパッケージを管理するには、［**システム**］に移動し、管理するクライアントをクリックし、**システム › ソフトウェア › パッケージ**サブタブに移動します。 このセクションで使用できるオプションは、選択したクライアントのタイプ、および現在のチャンネルのサブスクリプションによって異なります。

パッケージをインストールまたはアップグレードするとき、ライセンスまたはEULAは自動的に受け入れられます。

ほとんどのパッケージ管理アクションは、アクションチェーンに追加できます。 アクションチェーンの詳細については、**Reference › Schedule**を参照してください。

7.1.1. プロファイルを使用したパッケージの比較

保存されているプロファイルでクライアントにインストールされたパッケージを比較できます。または別のクライアントにインストールされたパッケージと比較できます。 比較を実行するとき、選択したクライアントを一致するように変更できます。

パッケージをプロファイルと比較するには、プロファイルを保存済みである必要があります。 プロファイルは、現在インストールされているクライアントのパッケージから作成されます。 プロファイルを作成したら使用して、同じインストール済みパッケージで別のクライアントをインストールできます。

プロシージャ: 保存されているプロファイルの作成

1. SUSE ManagerのWeb UIで、［**システム**］に移動し、プロファイルのベースになっているクライアントをクリックし、**システム › ソフトウェア › パッケージ › プロファイル**サブタブに移動します。
2. **[システムプロファイルの作成]**をクリックします。
3. プロファイルの名前と説明を入力し、**[プロファイルの作成]**をクリックします。

プロシージャ: クライアントパッケージの比較

1. SUSE ManagerのWeb UIで、［**システム**］に移動し、比較するクライアントをクリックし、**システム › ソフトウェア › パッケージ › プロファイル**サブタブに移動します。 保存されているプロファイルを比較するには、プロファイルを選択し、**[比較]**をクリックします。
2. 別のクライアントと比較するには、クライアント名を選択し、**[比較]**をクリックし、2つのクライアントの差異一覧を表示します。

7.1.2. 孤立したパッケージ

孤立したパッケージとは、SUSE　Managerによって同期され、ソフトウェアチャンネルに関連付けられていないパッケージです。 したがって、孤立したパッケージは通常、SUSE Managerクライアントでは利用できず、ユーザは追加の作業を行わずにそのようなパッケージをインストールすることはできません。

次のいずれかのイベントの結果として、パッケージが孤立する可能性があります。

- 同期されたリポジトリはパッケージを削除します。 **strict mode**チャンネル設定では、SUSE Managerはそのようなパッケージをチャンネルからリンク解除しますが、パッケージは削除しません。
- リポジトリはパッケージを新しいバージョンに置き換えるので、以前のバージョンは削除されます。
- reposyncプロセスが中断されたので(たとえば、容量不足の例外が発生したため)、ダウンロードされたパッケージがチャンネルに関連付けられませんでした。
- ユーザはパッケージを手動でアップロードしましたが、どのチャンネルにも関連付けませんでした。

孤立したパッケージはユーザ環境で容量を消費し、チャンネルに関連付けられていないためクライアントに簡単に配布できません。 孤立したパッケージは、ブートストラップや顧客固有の手順のような特定のワークフローでは意味を持つ場合があります。

Web UIで孤立したパッケージを表示するには、**ソフトウェア › Manage › Packages › view Packages in no channel**(管理 > パッケージ > チャンネルのないパッケージを表示)をクリックします

SUSE Managerは、孤立したパッケージを検索しパッケージ組織IDを1に変更する、taskomaticジョブを定期的に実行します。 つまり、孤立したパッケージはtaskomaticジョブの実行後にのみ削除できます。 削除できない孤立したパッケージが発生した場合は、24時間待ってから再度削除を試みてください。

孤立したパッケージの削除:

- 個々のパッケージを対象とするコマンドラインツールを使用する場合。 例:

```
spacecmd package_remove zypper-1.14.52-150400.1.9.x86_64
```

- 孤立したパッケージすべてを一度に対象とするコマンドラインツールを使用する場合:

```
spacecmd package_removeorphans
```

7.2. パッチ管理

組織内でカスタムパッチを使用してクライアントを管理できます。 この方法では、カスタムチャンネルのパッケージのパッチ警告の発行、パッチインストールのスケジュール、組織間のパッチの管理を実行できます。

7.2.1. パッチの作成

カスタムパッチを使用するには、パッチを作成し、これにパッケージを追加し、1つまたは複数のチャンネルに追加する必要があります。

プロシージャ: カスタムパッチの作成

1. SUSE ManagerのWeb UIで、**パッチ › パッチの管理**に移動し、**[パッチの作成]**をクリックします。
2. ［**パッチの作成**］セクションで、次の詳細を使用します。
   - ［**概要**］フィールドにパッチの短い説明を入力します。
   - ［**アドバイザリ**］フィールドにパッチのラベルを入力します。 組織の命名規則を使用して、パッチの管理を簡単にすることをお勧めします。
   - ［**アドバイザリリリース**］フィールドにパッチのリリース番号を入力します。 たとえば、このパッチの最初のバージョンの場合、**1**を使用します。
   - ［**アドバイザリタイプ**］フィールドで、使用するパッチのタイプを選択します。 たとえば、エラーを修正するパッチには［**バグ修正アドバイザリ**］を選択します。
   - ［**セキュリティアドバイザリ**］のアドバイザリタイプを選択した場合、使用するセキュリティレベルを［**アドバイザリの重要度**］フィールドで選択します。
   - ［**製品**］フィールドに、パッチが参照する製品の名前を入力します。
   - オプション: ［**Author**］（作成者）フィールドにパッチの作成者の名前を入力します。
   - ［**トピック**］、［**説明**］、［**ソリューション**］の各フィールドにパッチに関する追加情報を入力します。
3. オプション: ［**バグ**］セクションで、次の詳細を使用して関連するバグの情報を指定します。
   - ［**ID**］フィールドにバグ番号を入力します。
   - ［**概要**］フィールドにバグの短い説明を入力します。
   - ［**Bugzilla URL**］フィールドにバグのアドレスを入力します。
   - ［**キーワード**］フィールドにバグに関するキーワードを入力します。 複数のキーワードの間ではカンマを使用してください。
   - ［**リファレンス**］、［**メモ**］の各フィールドにバグに関する追加情報を入力します。
   - 新しいパッチを追加するチャンネルを1つまたは複数選択します。
4. **[パッチの作成]**をクリックします。

既存のパッチを複製してパッチを作成することもできます。 複製では、パッケージの関連付けが保持され、

パッチの発行が簡素化されます。

プロシージャ: パッチの複製

1. SUSE ManagerのWeb UIで、**パッチ › パッチの複製**に移動します。
2. ［**潜在的に適用できる可能性のあるパッチを表示**］フィールドで、複製するパッチのソフトウェアチャンネルを選択します。
3. 複製する1つまたは複数のパッチを選択し、**[パッチの複製]**をクリックします。
4. 複製したパッチを追加するチャンネルを1つまたは複数選択します。
5. 詳細を確認し、複製を開始します。

パッチを作成したら、そのパッチにパッケージを割り当てることができます。

プロシージャ: パッチにパッケージを割り当てる

1. SUSE ManagerのWeb UIで、**パッチ › パッチの管理**に移動し、パッチのアドバイザリ名をクリックしてパッチの詳細を表示します。
2. **パッケージ › 追加**タブに移動します。
3. ［**チェンネル**］フィールドで、パッチに割り当てるパッケージが含まれているソフトウェアチャンネルを選択し、**[パッケージの表示]**をクリックします。 ［**All managed packages**］（管理しているすべてのパッケージ）を選択して、すべてのチャンネルで使用できるパッケージを表示できます。
4. 含めるパッケージを確認し、**[パッケージの追加]**をクリックします。
5. パッケージの詳細を確認し、**[確認]**をクリックしてパッチに適用します。
6. **パッケージ › 一覧表示/削除**タブに移動し、パッケージが正しく割り当てられていることを確認します。

パッケージをパッチに割り当てると、パッチキャッシュが更新され、変更が反映されます。 キャッシュの更新には数分かかる場合があります。

既存のパッチの詳細を変更する必要がある場合、［**パッチ管理**］ページから実行できます。

プロシージャ: 既存のパッチ警告の編集および削除

1. SUSE ManagerのWeb UIで、**パッチ › パッチの管理**に移動します。
2. パッチのアドバイザリ名をクリックし、パッチの詳細を表示します。
3. 必要に応じて変更し、**[パッチの更新]**をクリックします。
4. パッチを削除するには、［**パッチ管理**］ページでパッチを選択し、**[パッチの削除]**をクリックします。パッチの削除には、数分かかる場合があります。

7.2.2. クライアントへのパッチの適用

パッチの用意ができたら、クライアントに適用できます。その際、単独または他のパッチと一緒に適用できます。

パッチ内の各パッケージは、1つまたは複数のチャンネルの一部です。　クライアントがチャンネルにサブスクライブされていない場合、更新はインストールされません。

対象の更新より新しいバージョンのパッケージがクライアントにすでにインストールされている場合、その更新はインストールされません。　古いバージョンのパッケージがクライアントにインストールされている場合、そのパッケージはアップグレードされます。

プロシージャ: 適用可能なすべてのパッチを適用する

1. SUSE ManagerのWeb UIで、**システム › 概要**に移動し、更新するクライアントを選択します。
2. **ソフトウェア › パッチ**タブに移動します。
3. **[すべてを選択]**をクリックして適用可能なすべてのパッチを選択します。
4. **[パッチの適用]**をクリックしてクライアントを更新します。

管理者特権でサインインしている場合、クライアントに対してより大規模なバッチアップグレードも実行できます。

プロシージャ: 1つのパッチを複数のクライアントに適用する

1. SUSE ManagerのWeb UIで、**パッチ › パッチリスト**に移動します。
2. 適用するパッチを見つけ、そのパッチの**システム**列で番号をクリックします。
3. パッチの適用先にするクライアントを選択し、**[パッチの適用]**をクリックします。
4. クライアントの一覧を確認し、更新を実行します。

プロシージャ: 複数のパッチを複数のクライアントに適用する

1. SUSE ManagerのWeb UIで、**システム › 概要**に移動し、更新するクライアントにチェックを付け、システムセットマネージャに追加します。
2. **システム › システムセットマネージャ**に移動し、［**パッチ**］タブに移動します。
3. クライアントに適用するパッチを選択し、**[パッチの適用]**をクリックします。
4. 更新する日時をスケジュールし、**[確認]**をクリックします。
5. 更新の進捗を確認するには、**スケジュール › 待機中の動作**に移動します。

スケジュールされたパッケージ更新は、各クライアントで設定された接続メソッドを使用してインストールされます。　詳細については、**Client-configuration › Contact-methods-intro**を参照してください。

7.3. システムのロック

システムのロックは、クライアントでアクションが発生しないようにするために使用されます。　たとえば、システムのロックは、クライアントの更新または再起動が行われないようにします。　これは、運用ソフトウェアを実行しているクライアントに対して、または不注意による変更が行われないようにするために役に立ちます。 アクションを実行する準備ができたときにシステムのロックを無効にできます。

システムのロックは、従来のクライアントおよびSaltクライアントでは異なる方法で実装されています。

7.3.1. 従来のクライアントのシステムのロック

従来のクライアントがロックされると、Web UIを使用してアクションをスケジュールすることができず、**システム › システム一覧**にあるクライアントの名前の横に南京錠のアイコンが表示されます。

プロシージャ: 従来のクライアントのシステムのロック

1. SUSE ManagerのWeb UIで、ロックするクライアントの［**システムの詳細**］ページに移動します。
2. ［**ロックの状態**］で、**[Lock this system]**（このシステムをロックする）をクリックします。 このクライアントは、**[Unlock this system]**（このシステムのロックを解除する）をクリックするまでロックされたままになります。

ロックされた従来のクライアントでは、リモートコマンド、自動化されているパッチの更新など、一部のアクションは実行できます。 自動化されているパッチの更新を停止するには、クライアントの［**システムの詳細**］ページに移動し、［**プロパティ**］タブで［**自動パッチ更新**］のチェックを外します。

7.3.2. Saltクライアントのシステムのロック

Saltクライアントがロックされている場合、または停止モードになっている場合、アクションをスケジュールできず、Salt実行コマンドが無効になり、黄色のバナーが［**システムの詳細**］ページに表示されます。 このモードでは、Web UIまたはAPIを使用してロックされているクライアントのアクションをスケジュールできますが、アクションは失敗します。

ロックメカニズムはSalt SSHクライアントでは使用できません。

プロシージャ: Saltクライアントのシステムのロック

1. SUSE ManagerのWeb UIで、ロックするクライアントの［**システムの詳細**］ページに移動します。
2. ［**式**］タブに移動し、システムのロックの式にチェックを付け、**[保存]**をクリックします。
3. **式 › System Lock**（システムのロック）タブに移動し、［**Lock system**］（システムのロック）にチェックを付け、**[保存]**をクリックします。 このページでは、クライアントがロックされているときに特定のSaltモジュールを有効にすることもできます。
4. 変更した場合、highstateを適用する必要がある場合があります。 この場合、Web UIのバナーで通知されます。 システムのロック式を削除するまで、クライアントはロックされたままです。

Saltの停止モードの詳細については、https://docs.saltproject.io/en/latest/topics/blackout/index.htmlを参照してください。

7.3.3. パッケージのロック

パッケージのロックは複数のクライアントで使用できますが、さまざまな機能セットを使用できます。 次のものを区別する必要があります。

1. SUSE Linux EnterpriseおよびopenSUSE (zyppベース)とRed Hat Enterprise LinuxまたはDebian クライアント、
2. 従来のクライアントとSalt クライアント。

Zyppベースのシステムでのパッケージのロック

パッケージのロックを使用して、ソフトウェアパッケージの未認可インストールや未認可アップグレードを防止します。 パッケージがロックされている場合、インストールできないことを示す南京錠のアイコンが表示されます。 ロックされているパッケージをインストールしようとすると、イベントログにエラーとしてレポートされます。

ロックされているパッケージは、インストール、アップグレード、または削除できません。SUSE Manager のWeb UIを使用しても、パッケージマネージャを使用してクライアントマシンで直接操作しても同様です。ロックされているパッケージは、依存関係のあるパッケージも間接的にロックします。

Zypperパッケージマネージャを備えたシステムでは、従来のクライアントとSaltクライアントでパッケージのロックを使用できます。

プロシージャ: パッケージのロックの使用

1. 管理対象システムで**ソフトウェア › パッケージ › ロック**タブに移動し、使用できるパッケージの一覧を表示します。
2. ロックするパッケージを選択し、**[Request Lock]**（ロックのリクエスト）をクリックします。 ロックをアクティブ化する日時を選択します。 デフォルトでは、できるだけ早くロックをアクティブ化します。 ロックはすぐにはアクティブにできないことに注意してください。
3. パッケージのロックを外すには、ロック解除するパッケージを選択し、**[Request Unlock]**（ロック解除のリクエスト）をクリックします。 ロックをアクティブ化する場合と同様に、日時を選択します。

Red Hat Enterprise LinuxやDebianのようなシステムでのパッケージのロック

一部のRed Hat Enterprise LinuxやDebianのようなシステムでは、Saltクライアントでパッケージのロックを使用できます。

Red Hat Enterprise LinuxやDebianのようなシステムでは、パッケージのロックは、ソフトウェアパッケージの未認可アップグレードや削除を防止するためにのみ使用されます。 パッケージがロックされている場合、変更できないことを示す南京錠のアイコンが表示されます。 ロックされているパッケージを変更しようとすると、イベントログにエラーとしてレポートされます。

ロックされているパッケージは、アップグレードまたは削除できません。SUSE ManagerのWeb UIを使用しても、パッケージマネージャを使用してクライアントマシンで直接操作しても同様です。 ロックされているパッケージは、依存関係のあるパッケージも間接的にロックします。

プロシージャ: パッケージのロックの使用

1. Red Hat Enterprise Linux 7システムでは、**yum-plugin-versionlock**パッケージを**root**としてインストールします。 Red Hat Enterprise Linux 8システムでは、**python3-dnf-plugin-versionlock**パッケージを**root**としてインストールします。 Debianシステムでは、**apt**ツールにロック機能が含まれています。

2. 管理対象システムで**ソフトウェア › パッケージ › ロック**タブに移動し、使用できるパッケージの一覧を表示します。
3. ロックするパッケージを選択し、**[Request Lock]**（ロックのリクエスト）をクリックします。 ロックをアクティブ化する日時を選択します。 デフォルトでは、できるだけ早くロックをアクティブ化します。 ロックはすぐにはアクティブにできないことに注意してください。
4. パッケージのロックを外すには、ロック解除するパッケージを選択し、**[Request Unlock]**（ロック解除のリクエスト）をクリックします。 ロックをアクティブ化する場合と同様に、日時を選択します。

7.4. 設定管理

クライアントそれぞれを手動で設定するのではなく、設定ファイルおよびチャンネルを使用してクライアントの設定を管理できます。

以下の機能の一部は、従来のクライアントのみが使用できます。 Saltクライアントでサポートされている機能については、以下の表を参照してください。

設定パラメータは、スクリプト化され、設定ファイルに保存されます。 SUSE ManagerのWeb UIを使用して設定ファイルを直接書き込むことができます。または、別の場所にあるファイルをアップロードまたはリンクできます。

設定ファイルは一元管理またはローカルで管理できます。 一元管理された設定ファイルは、グローバル設定チャンネルで提供され、SUSE Managerサーバにサブスクライブされるクライアントに適用できます。 ローカル管理された設定ファイルは、一元管理された設定ファイルを上書きするために使用されます。 このファイルは、設定管理特権がないSUSE Managerユーザにとって特に便利ですが、管理しているクライアントを変更する必要があります。

設定チャンネルは、設定ファイルの編成に使用されます。 クライアントを設定チャンネルにサブスクライブして、必要に応じて設定ファイルを展開できます。

設定ファイルはバージョン管理されるため、設定を追加し、クライアントで設定をテストし、必要に応じて前のリビジョンにロールバックできます。 設定チャンネルを作成したら、さまざまな設定ファイル間の比較や同じ設定ファイルの異なるリビジョン間の比較も実行できます。

設定ファイルは一元管理またはローカルで管理できます。 一元管理された設定ファイルはグローバル設定チャンネルによって提供されます。 ローカル管理された設定ファイルはSUSE Managerに直接作成またはアップロードされます。

使用できる設定管理機能はSaltクライアントと従来のクライアントで異なります。 次の表は、それぞれのクライアントタイプでサポートされる機能を示しています。

この表のアイコンの意味は次のとおりです。

- ✔: この機能はSUSEでサポートされています
- ✖: この機能はSUSEではサポートされていません
- ?: この機能は検討中であり、後日サポートされるかどうかは未定です。

表 55. 設定管理でサポートされる機能

機能	Salt	従来
グローバル設定チャンネル	✔	✔
ファイルの展開	✔	✔
ファイルの比較	?	✔
ローカル管理ファイル	✔ (Salt機能を使用)	✔
サンドボックスファイル	✖	✔
Highstateの適用	✔	✖
クライアントからのファイルのインポート	✖	✔
Jinjaテンプレート	✔	✖
設定マクロ	✔ (Salt機能(grains、pillarデータなど)を使用)	✔

7.4.1. 設定管理用に従来のクライアントを準備する

従来のクライアントでは、設定管理を使用するために追加の準備操作が必要です。 AutoYaSTまたはKickstartを使用して従来のクライアントをインストールした場合、おそらく適切なパッケージがすでに準備されています。 その他の従来のクライアントでは、そのクライアントのオペレーティングシステム用に関連するツールの子チャンネルがインストールされていることを確認します。 ソフトウェアチャンネルの詳細については、**Client-configuration › Channels**を参照してください。

必要なパッケージは次のとおりです。

- **mgr-cfg**: すべての**mgr-cfg-***パッケージで必要なベースライブラリおよび関数
- **mgr-cfg-actions**: SUSE Managerを使用してスケジュールされた設定アクションを実行するために必要です。
- **mgr-cfg-client**: 設定管理システムのクライアントの機能へのコマンドラインインターフェースを提供します。
- **mgr-cfg-management**: SUSE Manager設定を管理するためのコマンドラインインターフェースを提供します。

システム › アクティベーションキーに移動し、ブートストラップ中に使用するアクティベーションキーをクリックし、［**設定ファイルの配備**］オプションにチェックを付けることによってブートストラッププロセス中にこれらのパッケージをインストールできます。 アクティベーションキーの詳細については、**Client-configuration › Activation-keys**を参照してください。

7.4.2. 設定チャンネルの作成

新しい一元的な設定チャンネルを作成するには:

プロシージャ: 一元的な設定チャンネルの作成

1. SUSE ManagerのWeb UIで、**設定 › チャンネル**に移動し、**[設定チャンネルの作成]**をクリックします。
2. チャンネルの名前を入力します。
3. チャンネルのラベルを入力します。　このフィールドには、半角の英字、数字、ハイフン(-)、および下線(_)のみを含める必要があります。
4. 他のチャンネルから区別できるようにチャンネルの説明を入力します。
5. **[設定チャンネルの作成]**をクリックして新しいチャンネルを作成します。

設定チャンネルを使用して、SaltクライアントのSaltの状態を管理することもできます。

プロシージャ: Saltの状態チャンネルの作成

1. SUSE ManagerのWeb UIで、**設定 › チャンネル**に移動し、**[状態チャンネルの作成]**をクリックします。
2. チャンネルの名前を入力します。
3. チャンネルのラベルを入力します。　このフィールドには、半角の英字、数字、ハイフン(-)、および下線(_)のみを含める必要があります。
4. 他のチャンネルから区別できるようにチャンネルの説明を入力します。
5. **init.sls**ファイルの［**SLSコンテンツ**］を入力します。
6. **[設定チャンネルの作成]**をクリックして新しいチャンネルを作成します。

7.4.3. 設定ファイル、ディレクトリ、またはシンボリックリンクの追加

設定チャンネルを作成済みの場合、設定ファイル、ディレクトリ、またはシンボリックリンクを追加できます。

プロシージャ: 設定ファイル、ディレクトリ、またはシンボリックリンクの追加

1. SUSE ManagerのWeb UIで、**設定 › チャンネル**に移動し、設定ファイルに追加する設定チャンネルの名前をクリックし、**ファイルの追加 › ファイルの作成**サブタブに移動します。
2. ［**ファイルタイプ**］フィールドで、テキストファイル、ディレクトリ、またはシンボリックリンクを作成するかどうかを選択します。
3. ［**ファイル名/パス**］フィールドで、ファイルを展開する場所への絶対パスを入力します。
4. シンボリックリンクを作成している場合、［**シンボリックリンクの目的ファイル名/パス**］フィールドにターゲットのファイルおよびパスを入力します。
5. ［**所有権**］フィールドおよび［**ファイルアクセス許可モード**］にファイルの［**ユーザ名**］および［**グループ名**］を入力します。

6. クライアントでSELinuxが有効になっている場合、［**SELinux contexts**］（SELinuxコンテキスト）を設定して、必要なファイル属性(ユーザ、ロール、ファイルタイプなど)を有効にできます。
7. 設定ファイルにマクロが含まれている場合、マクロの先頭および末尾をマークする記号を入力します。
8. ［**ファイルの内容**］テキストボックスで設定ファイルの内容を入力し、スクリプトドロップダウンボックスを使用して、適切なスクリプト言語を選択します。
9. **[設定ファイルの作成]**をクリックします。

7.4.4. クライアントを設定チャンネルにサブスクライブする

個々のクライアントを設定チャンネルにサブスクライブできます。そのためには、**システム › システム一覧**に移動し、サブスクライブするクライアントを選択し、［**設定**］タブに移動します。 複数のクライアントを設定チャンネルにサブスクライブするには、システムセットマネージャ(SSM)を使用できます。

プロシージャ: 複数のクライアントを設定チャンネルにサブスクライブする

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、操作するクライアントを選択します。
2. **システム › システムセットマネージャ**に移動し、**設定 › チャンネルにサブスクライブ**サブタブに移動し、使用できる設定チャンネルの一覧を表示します。
3. オプション: ［**現在サブスクライブしているシステム**］列で番号をクリックして、設定チャンネルに現在サブスクライブされているクライアントを表示します。
4. サブスクライブ先の設定チャンネルを確認し、**[続行]**をクリックします。
5. 上下矢印を使用して設定チャンネルをランクします。 設定の競合が設定チャンネルで発生した場合、一覧の上の方にあるチャンネルが優先されます。
6. 選択したクライアントにチャンネルを適用する方法を決定します。**[最も低い順位でサブスクライブ]**をクリックして、現在サブスクライブしているチャンネルより低い優先度で新しいチャンネルを追加します。**[最も高い順位でサブスクライブ]**をクリックして、現在サブスクライブしているチャンネルより高い優先度で新しいチャンネルを追加します。**[既存のサブスクリプションを置換]**をクリックして、既存のチャンネルを削除し、新しいチャンネルに置き換えます。
7. **[サブスクリプションの適用]**をクリックします。

新しい設定チャンネルの優先度が既存のチャンネルと競合する場合、重複チャンネルが削除され、新しい優先度に応じて置き換えられます。 クライアントの設定優先度をアクションで順序変更する場合、Web UIでは続行する前に変更を確認する必要があります。

7.4.5. 設定ファイルの比較

システムセットマネージャ(SSM)を使用して、SUSE Managerサーバに保存されている設定ファイルを使用してクライアントに展開された設定ファイルを比較することもできます。

プロシージャ: 設定ファイルの比較

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動して、比較する設定ファイルにサブスクライブされているクライアントを選択します。

2. **システム › システムセットマネージャ**に移動し、**設定 › ファイルの比較**サブタブに移動し、使用できる設定チャンネルの一覧を表示します。
3. オプション: ［**システム**］列で番号をクリックして、設定ファイルに現在サブスクライブされているクライアントを表示します。
4. 比較する設定ファイルを確認し、**[ファイルの比較をスケジュール]**をクリックします。

7.4.6. SaltクライアントでのJinjaテンプレート

Jinjaテンプレートは、Saltクライアントで可能です。 Jinjaはピラーまたはグレインからの変数を提供します。 これらは、設定ファイルまたはSalt状態で使用できます。

詳細については、次の例のhttps://docs.saltproject.io/salt/user-guide/en/latest/topics/jinja.htmlを参照してください。

```
{% if grains.os_family == 'RedHat' %}
  {% set dns_cfg = '/etc/named.conf' %}
{% elif grains.os_family == 'Debian' %}
  {% set dns_cfg = '/etc/bind/named.conf' %}
{% else %}
  {% set dns_cfg = '/etc/named.conf' %}
{% endif %}
dns_conf:
  file.managed:
    - name: {{ dns_cfg }}
    - source: salt://dns/files/named.conf
```

7.4.7. 従来のクライアントにおける設定ファイルのマクロ

1つのファイルを保存して同じ設定を共有できることは便利ですが、同じ設定ファイルの多数のバリエーションまたはシステム固有の詳細(ホスト名やMACアドレスなど)のみが異なる設定ファイルが必要になる場合があります。 この場合、設定ファイル内でマクロまたは変数を使用できます。 マクロまたは変数を使用すると、数百さらには数千のバリエーションを含む単一のファイルをアップロードし、配布することができます。 カスタムシステム情報用の変数に加えて、次の標準マクロがサポートされています。

```
rhn.system.sid
rhn.system.profile_name
rhn.system.description
rhn.system.hostname
rhn.system.ip_address
rhn.system.custom_info(key_name)
rhn.system.net_interface.ip_address(eth_device)
rhn.system.net_interface.netmask(eth_device)
rhn.system.net_interface.broadcast(eth_device)
rhn.system.net_interface.hardware_address(eth_device)
rhn.system.net_interface.driver_module(eth_device)
```

この機能を使用するには、［**設定チャンネルの詳細**］ページで設定ファイルをアップロードまたは作成します。 ［**設定チャンネルの詳細**］ページを開き、選択したサポート対象のマクロを含めます。 変数を並べて記述するために使用するデリミタは、［**マクロ開始デリミタ**］フィールドと［**マクロ終了デリミタ**］フィールドで設定したデリミタと必ず一致させ、ファイル内の他の文字と競合しないようにしてください。 デリミタは長さを2文字とし、パーセント(**%**)記号を含めないことをお勧めします。

たとえば、IPアドレスとホスト名のみが異なるすべてのサーバに適用される1つのファイルがあるとします。 サーバごとに別の設定ファイルを管理する代わりに、**server.conf**などの単一のファイルを作成して、IPアドレスとホスト名のマクロを含めることができます。

```
hostname={| rhn.system.hostname |}
ip_address={| rhn.system.net_interface.ip_address(eth0) |}
```

SUSE Manager Web UIのスケジュールされたアクションまたはSUSE Manager設定クライアント(**mgrcfg-client**)のコマンドラインのいずれかを通じて、そのファイルを個々のシステムに配布する場合、変数は、SUSE Managerのシステムプロファイルに記録されているシステムのホスト名とIPアドレスに置き換えられます。 この例では、展開されるバージョンは次のようになります。

```
hostname=test.example.domain.com
ip_address=177.18.54.7
```

カスタムシステム情報を取得するには、カスタム情報マクロ(**rhn.system.custom_info**)にキーラベルを挿入します。 たとえば、「asset」というラベルを付けたキーを作成した場合、設定ファイルのカスタム情報マクロにそのキーラベルを追加して、そのキーラベルを含むシステムを値に代入することができます。 このマクロは次のようになります。

```
asset={@ rhn.system.custom_info(asset) @}
```

そのキーの値を含むシステムにこのファイルが展開されると、マクロが変換され、次のような文字列が生成されます。

```
asset=Example#456
```

(エラーを防ぐために必要な場合などに)デフォルト値を含めるには、カスタム情報マクロに次のようにデフォルト値を付加することができます。

```
asset={@ rhn.system.custom_info(asset) = 'Asset #' @}
```

値を含むシステムでは、このデフォルトはその値によって上書きされます。

システム管理を支援するために、SUSE Managerクライアントコンピュータ上でSUSE Manager設定マネージャ(**mgrcfg-manager**)を利用できます。 このツールは、システムに依存しないため、ファイルを変換または変更しません。 **mgrcfg-manager**コマンドは、システム設定に依存しません。 バイナリファイルを変更することはできません。

7.5. 電源管理

SUSE ManagerのWeb UIを使用して、電源オン、電源オフ、およびクライアントの再起動を実行できます。

この機能は、IPMIまたはRedfishプロトコルを使用し、Cobblerプロファイルを使用して管理されます。 選択したクライアントには、これらのプロトコルのいずれかをサポートしている電源管理コントローラがある必

要があります。

Redfishの場合、クライアントとSUSE Managerサーバの間に有効なSSL接続を確立できることを確認してください。Redfish管理コントローラのSSLサーバ証明書を署名するために使用される認証局を信頼している必要があります。CA証明書は**.pem**フォーマットで、SUSE Managerサーバの**/etc/pki/trust/anchors/**に保存される必要があります。証明書を保存したら、**update-ca-certificate**を実行します。

プロシージャ: 電源管理を有効にする

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、管理するクライアントを選択し、**プロビジョニング › 電源管理**タブに移動します。
2. ［**タイプ**］フィールドで、使用する電源管理プロトコルを選択します。
3. 電源管理サーバの詳細を入力し、適切なボタンをクリックしてアクションを実行し、**[保存のみ]**をクリックし、アクションを実行せずに詳細を保存します。

電源管理アクションを複数のクライアントに同時に適用できます。そのためには、クライアントをシステムセットマネージャに追加します。 システムセットマネージャの使用法の詳細については、**Client-configuration › System-set-manager**を参照してください。

7.5.1. 電源管理とCobbler

電源管理機能を初めて使用するとき、Cobblerシステムレコードが自動的に作成されます(まだクライアントに存在しない場合)。自動作成されたシステムレコードは、ネットワークから起動できず、ダミーのシステムイメージへの参照が含まれています。Cobblerがプロファイルまたはイメージのないシステムレコードを現時点でサポートしていないため、この参照は必要です。

Cobbler電源管理は、フェンスエージェントツールを使用して、IPMI以外のプロトコルをサポートしています。SUSE Managerでは、IPMIプロトコルとRedfishプロトコルのみがサポートされています。クライアントを設定して、その他のプロトコルを使用できます。そのためには、**rhn.conf**設定ファイルの**java.power_management.types**設定パラメータにフェンスエージェント名をカンマ区切りリストとして追加します。

7.6. 設定のスナップショット

スナップショットは、設定時点でのクライアントのパッケージプロファイル、設定ファイル、およびSUSE Manager設定を記録します。 古いスナップショットにロールバックして、前の設定に戻すことができます。

スナップショットは、従来のクライアントでのみサポートされます。 Saltクライアントでは、この機能をサポートしていません。

スナップショットは、一部のアクション実行後に自動的に取得されます。 いつでもスナップショットを手動で取得することもできます。 クライアントで元に戻せないかもしれないアクションを実行する前に、その時点のスナップショットがあることを確認することをお勧めします。

スナップショットはデフォルトで有効になっています。 自動スナップショットは無効にできます。そのためには、**rhn.conf**設定ファイルの**enable_snapshots=0**を設定します。

スナップショットを管理するには、**システム › システム一覧**に移動して、管理するクライアントを選択します。 選択したクライアントで現在のスナップショットをすべて一覧表示するには、**プロビジョニング › スナップショット**タブに移動します。 スナップショットに記録された変更の詳細を表示するには、スナップショットの名前をクリックします。 **プロビジョニング › スナップショット**タブでサブタブを使用して、選択したスナップショットにロールバックした変更を表示できます。

- グループメンバーシップ
- チャンネルサブスクリプション
- インストールされたパッケージ
- 設定チャンネルサブスクリプション
- 設定ファイル
- スナップショットタグ

スナップショットを使用して、クライアントに対するほとんどの変更をロールバックできますが、すべてではありません。 たとえば、複数の更新はロールバックできません。 また、製品の移行はロールバックできません。 クライアントでアップグレードを実行する前に、必ずバックアップを取ってください。

7.6.1. スナップショットタグ

スナップショットタグを使用すると、わかりやすい説明をスナップショットに追加できます。 タグを使用して、動作したことがわかっている最後の設定、成功したアップグレードなどスナップショットに関する追加情報を記録できます。

スナップショットタグを管理するには、**システム › システム一覧**に移動して、管理するクライアントを選択します。 選択したクライアントで現在のスナップショットタグをすべて一覧表示するには、**プロビジョニング › スナップショットタグ**タブに移動します。 ［**システムタグの作成**］をクリックし、説明を入力し、**[現在のスナップショットにタグ付け]**ボタンをクリックします。

7.6.2. 大規模インストールのスナップショット

SUSE Managerで保持できるスナップショットの上限数はありません。 つまり、クライアント、パッケージ、チャンネル、および設定変更を追加すると、スナップショットを保存するデータベースが大きくなります。

数千のクライアントを含む大規模インストールがある場合、古いスナップショットを定期的に削除するように、SUSE Manager APIを使用して、繰り返しスケジュールに繰り返しクリーンアップスクリプトを使用できます。 または、この機能を無効にできます。そのためには、**enable_snapshots=0**(**rhn.conf**設定ファイル内)を設定します。

7.7. カスタムシステム情報

クライアントに関してカスタマイズしたシステム情報を含めることができます。 システム情報は「キー:値」ペアで定義され、クライアントに割り当てることができます。 たとえば、特定のプロセッサに対して「キー:

値」ペアを定義してから、そのプロセッサがインストールされているすべてのクライアントにそのキーを割り当てることができます。 カスタムシステム情報は分類され、SUSE ManagerのWeb UIを使用して検索できます。

始める前に、カスタム情報を保存できるキーを作成する必要があります。

プロシージャ: カスタムシステム情報のキーの作成

1. SUSE ManagerのWeb UIで、**システム › カスタムシステム情報**に移動し、**[キーの作成]**をクリックします。
2. ［**キーラベル**］フィールドにキーの名前を追加します。 スペースは使用しません。 例: **intel-x86_64-quadcore**。
3. ［**説明**］フィールドに必要な追加情報を入力します。
4. 必要な各キーで操作を繰り返します。

Saltクライアントの場合、この情報はSalt pillarを使用して利用できます。 次のようなコマンドを使用して、Saltクライアントからこの情報を取得できます。

```
salt $minionid pillar.get custom_info:key1
```

このコマンドは、次のような出力になります。

```
$minionid:
    val1
```

カスタムシステム情報キーを作成するとき、キーをクライアントに適用できます。

プロシージャ: カスタム情報キーをクライアントに適用する

1. SUSE ManagerのWeb UIで、［**システム**］に移動し、カスタム情報を適用するクライアントをクリックし、**詳細 › カスタム情報**タブに移動します。
2. **[値の作成]**をクリックします。
3. 適用する値を見つけ、キーラベルをクリックします。
4. ［**値**］フィールドに追加情報を入力します。
5. **[キーの更新]**をクリックしてカスタム情報をクライアントに適用します。

設定管理の詳細については、**Client-configuration › Configuration-management**を参照してください。

7.8. システムセットマネージャ

システムセットマネージャ(SSM)は、同時に複数のクライアントでアクションを実行するために使用します。 SSMで一時的なクライアントセットが作成されます。これは、多数のクライアントに適用する必要がある1回限定アクションに便利です。 より永続的なセットが必要な場合、代わりにシステムグループの使用を検討してください。 システムグループの詳細については、**Client-configuration › System-groups**を参照し

てください。

SSMで使用できるアクションを次の表に示します。 この表のアイコンの意味は次のとおりです。

- ✔: このアクションはこのクライアントタイプ用にSSMで使用できます
- ✖: このアクションはこのクライアントタイプ用にSSMで使用できません
- ?: このアクションはこのクライアントタイプ用に検討中であり、後日サポートされるかどうかは未定です。

表 56. 使用可能なSSMアクション

アクション	従来	Salt
システムのリスト	✔	✔
パッチのインストール	✔	✔
パッチの更新のスケジュール	✔	✔
パッケージのアップグレード	✔	✔
パッケージのインストール	✔	✔
パッケージの削除	✔	✔
パッケージの検証	✔	✖
グループを作成する	✔	✔
グループの管理	✔	✔
チャンネルのメンバーシップ	✔	✔
チャンネルサブスクリプション	✔	✖
チャンネルの展開/diff	✔	✖
クライアントの自動インストール	✔	✖
スナップショットのタグ	✔	✖
リモートコマンド	✔	✖
電源管理	✔	✖
システム設定の更新	✔	✔
ハードウェアプロファイルの更新	✔	✔
パッケージのプロファイルの更新	✔	✔
カスタム値の設定/削除	✔	✔
クライアントの再起動	✔	✔

アクション	従来	Salt
クライアントの別の組織への移行	✔	✔
クライアントの削除	✔	✔

SSMのクライアントの選択は複数の方法で実行できます。

- **システム › システム一覧**に移動し、操作するクライアントにチェックを付けます。
- **システム › システムグループ**に移動し、操作するシステムグループで**[SSMで使用]**をクリックします。
- **システム › システムグループ**に移動し、操作するグループにチェックを付け、**[グループでの作業]**をクリックします。

操作するクライアントを選択したら、**システム › システムセットマネージャ**に移動し、上部のメニューバーにある［**システムが選択されました**］アイコンをクリックします。

SSMの詳細は、SUSE ManagerのWeb UIで別の部分にある詳細と若干異なる場合があります。 SSMでは、使用できるすべての更新が表示されます。 そのため、最新バージョンではないかもしれないパッケージをアップグレードされる場合があります。

7.8.1. SSMでベースチャンネルを変更する

SSMを使用して、複数のクライアントのベースチャンネルを同時に変更できます。

ベースチャンネルを大幅に変更すると、影響を受けるクライアントで使用できるパッケージおよびパッチが変更されます。 注意して使用してください。

プロシージャ: SSMを使用して複数のクライアントのベースチャンネルを変更する

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、操作するクライアントにチェックを付け、**システム › システムセットマネージャ**に移動します。
2. ［**チャンネル**］サブタブに移動します。
3. リストで現在のベースチャンネルを見つけ、［**システム**］列に表示されている数字が正しいことを確認します。 この列の数字をクリックして、変更するクライアントの詳細を表示できます。
4. ［**必要なベースチャンネル**］フィールドで新しいベースチャンネルを選択し、**[次へ]**をクリックします。
5. 子チャンネルそれぞれで、［**変更なし**］、［**サブスクライブ**］、または［**サブスクライブの中止**］を選択し、**[次へ]**をクリックします。
6. 変更内容を確認し、いつ変更するかを選択します。
7. **[確認]**をクリックして、変更をスケジュールします。

7.9. システムグループ

システムグループを使用して、多数のクライアントの管理を簡単にできます。 グループは、更新、設定チャンネル、Saltの状態、または方式の適用など、一括アクションをクライアントで実行するために使用できます。

使用している環境で動作する方法でクライアントをグループに編成できます。 たとえば、オペレーティングシステムがインストールされているクライアント、クライアントがある物理的な場所、または処理しているワークロードの種類を編成できます。 クライアントは、任意の数のグループに属することができるため、さまざまな方法でグループを定義できます。

クライアントをグループに編成している場合、1つまたは複数のグループのすべてのクライアントの更新を実行できます。または、複数のグループに属しているクライアントの更新を実行できます。 たとえば、すべてのSaltクライアント用に1つのグループを定義し、すべてのSLESクライアント用に別のグループを定義できます。 その後、すべてのSaltクライアントの更新を実行したり、両グループに属しているクライアントを使用してすべてのSalt SLESクライアントの更新を実行できます。

7.9.1. グループの作成

グループを使用してクライアントを編成する前にグループを作成する必要があります。

プロシージャ: 新しいシステムグループの作成

1. SUSE ManagerのWeb UIで、**システム › システムグループ**に移動します。
2. **[グループの作成]**をクリックします。
3. 新しいグループの名前と説明を指定します。
4. **[グループの作成]**をクリックしてグループを保存します。
5. 必要な各グループで操作を繰り返します。

7.9.2. グループにクライアントを追加する

個々のクライアントをグループに追加したり、複数のクライアントを同時に追加できます。

プロシージャ: 1つのクライアントのグループへの追加

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、追加するクライアントの名前をクリックします。
2. **グループ › 参加**タブに移動します。
3. 参加するグループを確認し、**[選択したグループに参加]**をクリックします。

プロシージャ: 複数のクライアントのグループへの追加

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、クライアントを追加するグループの名前をクリックします。

2. ［**ターゲットシステム**］タブに移動します。
3. 追加するクライアントを確認して、**[システムの追加]**をクリックします。

プロシージャ: SSMで複数のクライアントをグループに追加する

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、追加するそれぞれのクライアントを確認します。クライアントがシステムセットマネージャに追加されます。
2. **システム › システムセットマネージャ**に移動し、［**グループ**］タブに移動します。
3. 参加するグループを見つけ、［**追加**］にチェックを付けます。
4. **[メンバーの変更]**をクリックします。
5. **[確認]**をクリックして、選択したグループにクライアントを追加します。

システムセットマネージャの詳細については、**Client-configuration › System-set-manager**を参照してください。

グループに属しているクライアントを確認できます。そのためには、**システム › システムグループ**に移動し、グループの名前をクリックし、［**システム**］タブに移動します。 または、システムグループをグラフィカル表示できます。そのためには、**システム › 可視化 › システムのグループ化**に移動します。

7.9.3. グループの操作

クライアントをグループに編成すると、グループを使用して更新を管理できます。 Saltクライアントの場合、グループにあるクライアントのすべてに状態と方式を適用できます。

SUSE ManagerのWeb UIで、**システム › システムグループ**に移動します。 グループ内のいずれかのクライアントに適用できる更新がある場合、リストにアイコンが表示されます。 グループ内のいずれかのクライアントで更新ステータスの確認が無効になっている場合、リストには疑問符アイコンが表示されます。 アイコンをクリックすると、適用できる更新に関する詳細情報が表示され、クライアントに適用されます。

同時に複数のグループを操作することも可能です。 操作するグループを選択し、**[ユニオンで作業する]**をクリックし、すべての選択グループですべてのクライアントを選択します。

または、両方のグループに属しているクライアントを操作できます。 2つ以上のグループを選択し、**[インターセクションで作業する]**をクリックし、選択したすべてのグループに存在しているクライアントのみ選択します。 たとえば、すべてのSaltクライアント用に1つのグループがあり、すべてのSLESクライアント用に別のグループがあるとします。 これらのグループのインターセクションはすべてのSalt SLESクライアントになります。

7.10. システムの種類

クライアントは、システムの種類で分類されます。 各クライアントは、両方のベースシステムの種類を備えることができ、アドオンシステムの種類が割り当てられます。

ベースシステムの種類には、従来のクライアントでは**管理**、Saltクライアントでは**Salt**が含まれます。

アドオンシステムの種類には、仮想ホストとして動作するクライアントでは**仮想ホスト**、ビルドホストとして動作するクライアントでは**コンテナビルドホスト**が含まれます。

アドオンシステムの種類は調整できます。そのためには、**システム › システム一覧 › システムの種類**に移動します。　アドオンシステムの種類を変更するクライアントにチェックを付け、［**付属エンタイトルメント**］を選択し、**[エンタイトルメントの追加]**または**[エンタイトルメントの削除]**をクリックします。

ベースシステムの種類を**管理**から**Salt**に変更することもできます。そのためには、クライアントを再登録します。

ベースシステムの種類を変更するには、クライアントを再登録する必要があります。　この操作を実行すると、クライアントのすべてのカスタマイズまたは設定は削除されますが、イベント履歴は保持されます。 クライアントのダウンタイムも発生します。

従来のクライアントからSaltクライアントへの移行については、**Client-configuration › Contact-methods-migrate-traditional**を参照してください。

Chapter 8. オペレーティングシステムのインストール

一般に、すでに動作しているクライアントを登録します。 SUSE Managerに登録する直前にコンピュータに手動でインストールするか、環境にSUSE Managerを追加する前にインストールされた既存のシステムを使用できます。

または、SUSE Managerを使用して、1回の手順でオペレーティングシステムをインストールしてSUSE Managerに登録することもできます。 この方法では一部または全部が自動化されているため、インストーラの質問に答える時間を節約することができます。これは、特にインストールと登録が必要な多くのクライアントがある場合に役立ちます。

SUSE Managerからオペレーティングシステムをインストールするには次のような複数の方法があります。

- 登録済みのクライアントでインプレースインストールを行う
- PXEブートを使用してネットワークを通じてインストールする
- インストール用CD-ROMまたはUSBメモリを作成し、そのメディアでコンピュータをブートする
- SUSE Manager for Retailソリューションの一部としてインストールする。

インプレースでの再インストール方法は、以前のオペレーティングシステムがクライアントにすでにインストールされており、クライアントがSUSE Managerにすでに登録されていることを前提としています。

インプレースインストール方法については、**Client-configuration › Autoinst-reinstall**を参照してください。

ネットワークブートによるインストール方法は、フォーマットされていないコンピュータで動作します。 ただし、これは次のような特定のネットワーク構成のみで実行できます。

- SUSE Managerサーバまたはそのプロキシのいずれかが、インストール対象のコンピュータと同じローカルネットワーク上にあるか、経路にあるすべてのルータを中継できるDHCPリレーに対応している。
- 新しいDHCPサーバをセットアップするか、既存のDHCPサーバを設定することができる。
- インストール対象のクライアントがPXEブートに対応しており、PXEブートを実行するように設定することができる。

ネットワークブート方法については、**Client-configuration › Autoinst-pxeboot**を参照してください。

リムーバブルメディアを使用する方法では、このようなネットワーク上の制約を受けません。 しかし、この方法はコンピュータがCD-ROMまたはUSBメモリを読み取ることができ、各メディアからブートできることを前提としています。 また、クライアントコンピュータに対する物理的なアクセスも必要です。

リムーバブルメディアを使用する方法については、**Client-configuration › Autoinst-cdrom**を参照してください。

SUSE Manager for Retailアプローチについては、**Retail › Retail-overview**を参照してください。

Ubuntuクライアントと Debianクライアントの自動インストールはサポートされていません。 これらのオペレーティングシステムは、手動でインストールする必要があります。

The autoinstallation features of SUSE Manager are based on a software named Cobbler. For more information about Cobbler, see https://cobbler.readthedocs.io/en/latest/.

SUSEは、SUSE Manager Web UIまたはSUSE Manager APIで使用できるCobblerの機能のみをサポートしています。 Cobblerがサポートする唯一のコマンドラインコマンドは**buildiso**です。 ここにはサポートされている機能のみが記載されています。

8.1. 登録済みシステムを再インストールする

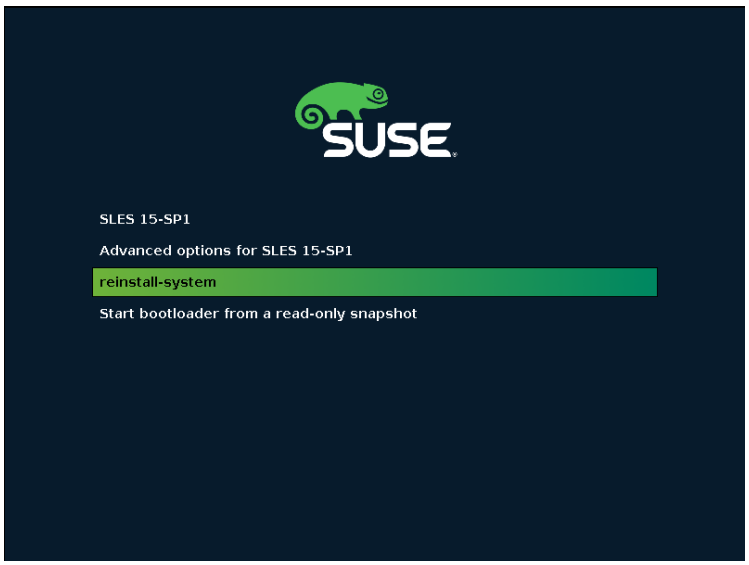
インプレースでの再インストールは、ローカルクライアントシステムから開始します。 したがって、クライアントがネットワークを通じてPXEブートを実行できる必要はありません。

登録済みのクライアントをインプレースで再インストールするには、自動インストールのディストリビューションと自動インストールプロファイルを定義する必要があります。 詳細については、**Client-configuration › Autoinst-distributions**と**Client-configuration › Autoinst-profiles**を参照してください。

自動インストールプロファイルと自動インストールのディストリビューションを定義したら、再インストールを実行できます。

プロシージャ: 登録済みのクライアントを再インストールする

1. SUSE Manager Web UIで、**システム › システム一覧**に移動し、再インストールするクライアントを選択し、**プロビジョニング › 自動インストール › スケジュール**サブタブに移動します。
2. 作成した自動インストールプロファイルを選択し、必要に応じてプロキシを選択して、**[自動インストールをスケジュールしてから終了する]**をクリックします。
3. クライアントが従来のクライアントであり、osadを設定していない場合は、ジョブがフェッチされるまで待つ必要があります。
4. **プロビジョニング › 自動インストール › セッションの状態**に移動するか、クライアント上で直接、インストールの進行状況を監視できます。 クライアントが再起動したら、［ブート］メニューで［**システムの再インストール**］という新しいオプションを選択します。

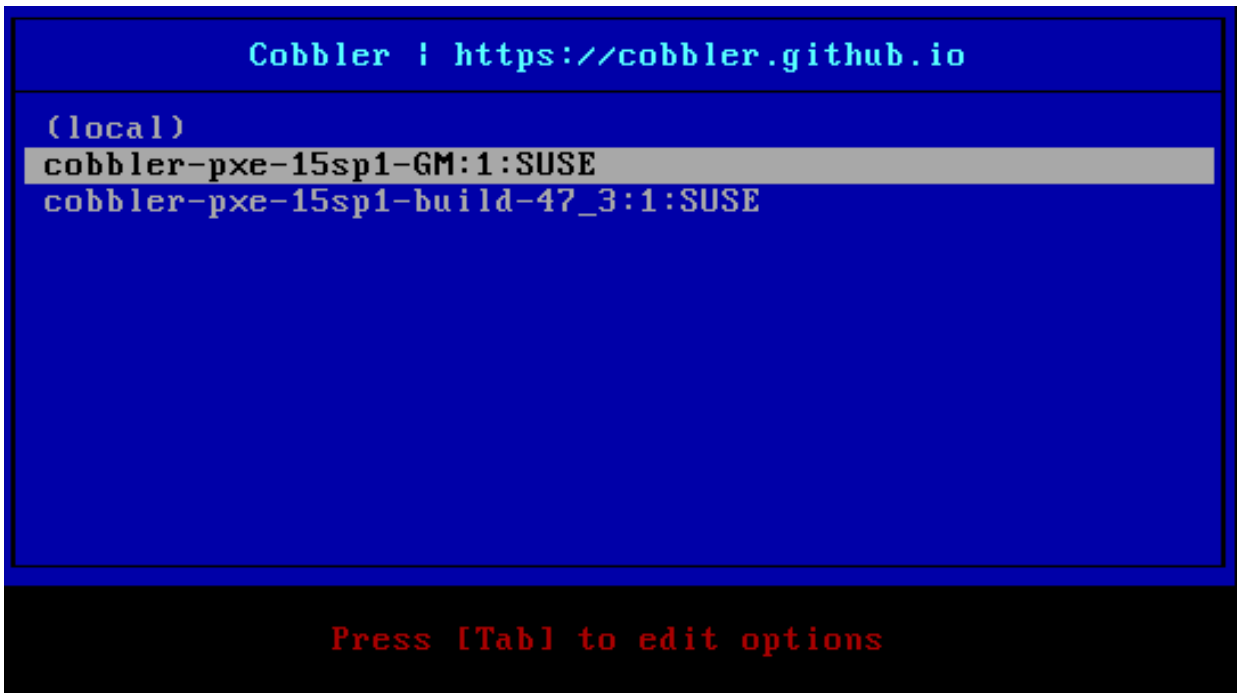


その後、インストールはHTTPプロトコルを通じて進められます。

8.2. ネットワークを通じてインストールする(PXEブート)

ネットワークブートによるインストール中に、次の処理が実行されます。

1. クライアントがPXEモードでブートされます。
2. DHCPサーバが、IPアドレスとマスク、インストールサーバのアドレス、サーバ上のブートローダファイルの名前をクライアントに提供します。
3. クライアントは、インストールサーバからTFTPプロトコルを通じてブートローダファイルをダウンロードし、実行します。
4. クライアントは、インストールに使用できるプロファイルをメニューから選択するか、いずれかのプロファイルで自動インストールを開始します。
5. クライアントは、TFTPプロトコルを通じて、そのプロファイルと一致するディストリビューション用のカーネルと初期RAMディスクをダウンロードします。
6. インストールカーネルは、インストールプログラム、KickstartまたはAutoYaSTを起動します。　それ以降、カーネルは、HTTPプロトコルを通じてサーバで提供されるリソースを使用します。
7. ディストリビューションは、KickstartまたはAutoYaSTプロファイルに従って自動的にインストールされます。
8. プロファイルは、従来のクライアントまたはSaltクライアントとしてクライアントをSUSE　Managerサーバに登録するコードスニペットを呼び出します。



インストールサーバは、SUSE Managerサーバまたはそのプロキシのいずれかにすることができです。 プロキシからインストールするには、事前にサーバとプロキシの間でTFTPツリーを同期させる必要があります。

DHCPサーバは、ホスト名、ルータのアドレス、ドメインネームサーバのアドレスなど、その他の構成情報をクライアントに提供することもあります。 この情報の一部は、インストールサーバをドメイン名で指定した場合など、自動インストールのために必要になることがあります。

[PXEブート]メニューで、最初の選択肢は**[ローカルブート]**です。 これを選択すると、ブートプロセスはローカルディスクドライブから続行されます。 特定の時間内にプロファイルが選択されなかった場合は、このオプションが自動的に選択されます。 これは、プロファイルを選択する人間のオペレータがいない場合に、自動インストールが開始されないようにするための安全対策です。

または、手動での操作なしで、いずれかのプロファイルからインストールを自動的に開始することもできます。 これは「無人プロビジョニング」と呼ばれます。

「ベアメタル」機能は、PXEブートに基づく無人プロビジョニングの一種です。 このシナリオでは、ブートローダファイルはSUSE Managerサーバでクライアントを登録するだけで、インストールは開始しません。インプレースでの再インストールを後で実行することができます。

手順: PXEブートによるインストール

1. DHCPサーバを準備します。DHCPサーバを準備するを参照してください。
2. 自動インストールのディストリビューションを準備します。**Client-configuration › Autoinst-distributions**を参照してください。
3. 自動インストールプロファイルを準備します。**Client-configuration › Autoinst-profiles**を参照してください。
4. クライアントを再起動し、インストールするプロファイルを選択します。

その他の一部の手順はオプションです。プロキシをインストールサーバとして使用するには、プロキシを使用してTFTPツリーを同期するを参照してください。無人プロビジョニングについては、**Client-configuration › Autoinst-unattended**を参照してください。

8.2.1. DHCPサーバを準備する

PXEブートプロセスは、DHCPを使用して、TFTPサーバを検索します。 SUSE Managerサーバまたはそのプロキシは、TFTPサーバとして機能させることができます。

このためには、ネットワークのDHCPサーバへの管理アクセス権が必要です。 TFTPブートサーバとしてインストールサーバをポイントするようにDHCP設定ファイルを編集します。

例: ISC DHCPサーバを設定する

1. DHCPサーバでrootとして**/etc/dhcpd.conf**ファイルを開きます。
2. クライアントに対する宣言を次のように変更します。

```
host myclient { (...)
                next-server 192.168.2.1;
                filename "pxelinux.0"; }
```

1. ファイルを保存し、**dhcpd**サービスを再起動します。

この例では、**192.168.2.1**でPXEクライアント**myclient**をインストールサーバに指定し、**pxelinux.0**ブートローダファイルの取得を指示しています。

DHCPサーバがSUSE Managerに登録されている場合は、その代わりにDHCPd式を使用して設定することもできます。

例: DHCPd式を使用したISC DHCPサーバの設定

1. **システム › システム一覧**に移動し、変更するクライアントを選択し、［**式**］タブに移動してDHCPd式を有効にします。
2. 式の［**Dhcpd**］タブに移動し、［**次のサーバ**］フィールドに、インストールサーバのホスト名またはIPアドレスを入力します。
3. ［**Filename EFI**］（ファイル名EFI）フィールドで、**grub/shim.efi**と入力し、EFI PXEのサポートを有効にします。
4. ［**ファイル名**］フィールドで、**pxelinux.0**と入力し、従来のBIOSのサポートを有効にします。
5. **[Save Formula]**（式の保存）をクリックして設定を保存します。
6. highstateを適用します。

セキュアブートを使用しない場合は、［**Filename EFI**］（ファイル名EFI）フィールドに**grub/shim.efi**ではなく**grub/grubx86.efi**と入力します。

異なるアーキテクチャについては、表異なるアーキテクチャ用のGRUB EFIバイナリ

名を参照してください。

Cobblerで管理されるDHCPの使用は、SUSE Managerではサポートされていません。

これですべてのホストにグローバルPXEサーバが設定されますが、ホストごとに設定することもできます。DHCPd式の詳細については、**Specialized-guides › Salt**を参照してください。

8.2.2. プロキシを使用してTFTPツリーを同期する

SUSE Managerサーバ上のTFTPツリーはSUSE Managerプロキシと同期させることができます。 同期するには、HTTPSポート443を開く必要があります。

プロキシを追加するたびに、ツリーの同期に時間がかかるようになります。

手順: サーバとプロキシ間のTFTPの同期

1. SUSE Managerサーバのコマンドプロンプトで、rootとして**susemanager-tftpsync**パッケージをインストールします。

```
zypper install susemanager-tftpsync
```

1. SUSE Managerプロキシのコマンドプロンプトで、rootとして**susemanager-tftpsync-recv**パッケージをインストールします。

```
zypper install susemanager-tftpsync-recv
```

1. プロキシでrootとして**configure-tftpsync.sh**スクリプトを実行します。 スクリプトは、SUSE Managerサーバおよびプロキシのホスト名およびIPアドレス、プロキシの**tftpboot**ディレクトリの場所の詳細についてインタラクティブに問い合わせます。詳細については、**configure-tftpsync.sh --help**コマンドを使用してください。
2. サーバでrootとして**configure-tftpsync.sh**スクリプトを実行します。

```
configure-tftpsync.sh proxy1.example.com proxy2.example.com
```

3. サーバで**cobbler sync**コマンドを実行して、プロキシにファイルをプッシュします。 プロキシが正しく設定されていないとこの操作は失敗します。

プロキシのリストを後で変更する場合、**configure-tftpsync.sh**スクリプトを使用して編集できます。

設定済みのプロキシを再インストールしてすべてのファイルを再プッシュする場合、**cobbler sync**を呼び出す前に**/var/lib/cobbler/pxe_cache.json**でキャッシュファイルを削除する必要があります。

8.2.3. 異なるアーキテクチャ用のGRUB EFIバイナリ名

表 57. 異なるアーキテクチャ用のGRUB EFIバイナリ名

アーキテクチャ	GRUB EFIバイナリ名
aarch64	grubaa64.efi
x86-64	grubx86.efi
ppc64le	grub.ppc64le

8.3. CD-ROMまたはUSBメモリを使用してインストールする

SUSE Managerにまだ登録されていないクライアントで、PXEを通じたネットワークブートを選択できない場合は、ブート可能なCD-ROMまたはUSBメモリを使用してシステムをインストールできます。

このようなリムーバブルメディアを作成する1つの方法は、Cobblerを使用することです。 Cobblerを使用してISOイメージを作成する方法については、CobblerでISOイメージを構築するを参照してください。

SUSEシステムでは、多くの場合、KIWIを使用してISOイメージを準備することが推奨されます。 詳細については、KIWIでSUSE ISOイメージを構築するを参照してください。

いずれの場合も、生成されたイメージをCD-ROMまたはUSBメモリに書き込みます。

8.3.1. CobblerでISOイメージを構築する

Cobblerは、一連のディストリビューション、カーネル、およびメニューが含まれているISOブートイメージを作成できます。これはPXEインストールと同じように動作します。

CobblerによるISOの構築はIBM Zではサポートされていません。

CobblerでISOイメージを作成するには、PXEを通じてネットワークブートを使用する場合と同様に、ディストリビューションとプロファイルを作成する必要があります。ディストリビューションの作成については、**Client-configuration › Autoinst-distributions**を参照してください。プロファイルの作成については、**Client-configuration › Autoinst-profiles**を参照してください。

Cobblerの**buildiso**コマンドは、ブートISOの名前および出力場所を定義するパラメータを取ります。

buildisoコマンドを実行する場合、**--distro**でディストリビューションを指定することは必須です。 **--iso**は出力の場所です。

```
cobbler buildiso --iso=/path/to/boot.iso --distro=<your-distro-label>
```

Web UIに表示されているだけでなく、Cobblerで一覧表示されているディストリビューションラベルとプロファイルラベルを使用する必要があります。 Cobblerによって保存されたディストリビューションとプロファイルの名前を一覧表示するには、次のコマンドを実行します。

```
cobbler distro list
cobbler profile list
```

ブートISOには、すべてのプロファイルおよびシステムがデフォルトで含まれています。 **--profiles**オプションと**--systems**オプションで、使用するプロファイルおよびシステムを制限できます。

```
cobbler buildiso --systems="system1 system2 system3" \
  --profiles="<your-profile1-label> <your-profile2-label> <your-profile3-label>" \
  --distro=<your-distro-label>
```

--espを使用すると、セキュアブートで構築されたブートISOを有効にすることができます。 EFIシステムパーティション(ESP)を明示的に指定します。 デフォルトでは、CobblerはESPパーティションを生成し、セキュアブートを無効にします。

```
cobbler buildiso \
  --esp="/usr/share/tftpboot-installation/SLE-15-SP4-x86_64/boot/x86_64/efi" \
  --iso=/path/to/boot.iso --distro=<your-distro-label>
```

プロシージャ: efiの検出

1. **cobbler distro list**を実行して、**distro**名を取得します。

   ```
   cobbler distro list
   ```

   このコマンドは、**sles15-sp4:1:SUSE**などの文字列を出力します。

2. **cobbler distro report**を実行して、**distro**ファイルの場所に関する情報を取得します。

   ```
   cobbler distro report --name sles15-sp4:1:SUSE
   ```

   このコマンドは、次のようなレポートを出力します。

   ```
   Name                   : sles15-sp4:1:SUSE
   Architecture           : x86_64
   ...output omitted...
   Initrd                 : /usr/share/tftpboot-installation/SLE-15-SP4-
   x86_64/boot/x86_64/loader/initrd
   Kernel                 : /usr/share/tftpboot-installation/SLE-15-SP6-
   x86_64/boot/x86_64/loader/linux
   ...output omitted...
   ```

3. ブートディレクトリのコンテンツを参照します。 上記の例では、ブートパーティションは**/usr/share/tftpboot-installation/SLE-15-SP4-x86_64/boot/x86_64/efi**ファイルですが、これはISOディストリビュータによって異なる場合があります。

ISOイメージをパブリック**tmp**ディレクトリに書き込むことができない場合、**/usr/lib/systemd/system/cobblerd.service**でsystemd設定を確認してください。

8.3.2. KIWIでSUSE ISOイメージを構築する

KIWIはイメージ作成システムです。 KIWIを使用して、SUSEシステムのインストール用にターゲットシステ

ムで使用するブート可能なISOイメージを作成することができます。　システムを再起動または電源オンするとき、このイメージからブートし、AutoYaST設定をSUSE　Managerから読み込み、AutoYaSTプロファイルに応じてSUSE Linux Enterprise Serverをインストールします。

ISOイメージを使用するには、システムをブートし、プロンプトに**autoyast**と入力します(AutoYaSTブートのラベルを**autoyast**のままにしていることを想定しています)。 `Enter`キーを押してAutoYaSTのインストールを開始します。

KIWIの詳細については、http://doc.opensuse.org/projects/kiwi/doc/を参照してください。

8.3.3. CobblerでRed Hat ISOイメージを構築する

詳細については、client-configuration:autoinst-cdrom.pdfを参照してください。

8.4. 自動インストールのディストリビューション

自動インストールプロセスでは、インストールを開始するために複数のファイルが必要です。　必要なファイルには、Linuxカーネル、初期RAMディスク、およびインストールモードでオペレーティングシステムをブートするために必要なその他のファイルが含まれます。

DVDイメージから必要なファイルを抽出できます。 詳細については、ISOイメージに基づくディストリビューションを参照してください。

または、**tftpboot-installation**パッケージをインストールすることもできます。 詳細については、RPMパッケージに基づくディストリビューションを参照してください。

また、これらのファイルと同じオペレーティングシステムバージョン用に、SUSE　Managerサーバでベースチャンネルを同期させておく必要があります。

ファイルの準備が整い、ベースチャンネルを同期したら、ディストリビューションを宣言する必要があります。 この操作により、インストールファイルがベースチャンネルに関連付けられます。 ディストリビューションは、1つ以上のインストールプロファイルによって参照されることがあります。　詳細については、自動インストールのディストリビューションを宣言するを参照してください。

8.4.1. ISOイメージに基づくディストリビューション

この方法では、クライアントにインストールするオペレーティングシステムのインストールメディアがあることを前提としています。 このメディアは通常DVD **.iso**イメージです。これには、Linuxカーネル、**initrd**ファイル、およびインストールモードでオペレーティングシステムをブートするために必要なその他のファイルが含まれています。

プロシージャ: インストールメディアからのファイルのインポート

1. インストールメディアをSUSE　Managerサーバにコピーします。　SUSEオペレーティングシステムの場合、https://www.suse.com/download/からインストールメディアをダウンロードできます。
2. ISOイメージをループマウントして、そのコンテンツをどこかにコピーします。

```
# mount -o loop,ro <image_name>.iso /mnt
# mkdir -p /srv/www/distributions
# cp -a /mnt /srv/www/distributions/<image_name>
# umount /mnt
```

ファイルパスをメモしておいてください。 このファイルパスは、ディストリビューションをSUSE Managerに対して宣言するときに必要です。

8.4.2. RPMパッケージに基づくディストリビューション

この方法は、SUSEシステムで動作します。 インストールシステム用にあらかじめパッケージされたファイルを使用するため、インストールメディアからコンテンツをインポートするよりも簡単です。

プロシージャ: インストールパッケージからファイルを抽出する

1. SUSE Managerサーバに、名前が**tftpboot-installation**で始まるパッケージをインストールします。 このパッケージの正確な名前は、**zypper se tftpboot-installation**コマンドで確認できます
2. **ls -d /usr/share/tftpboot-installation/***コマンドで、インストールファイルの場所を確認します。 ファイルパスをメモしておいてください。 このファイルパスは、ディストリビューションをSUSE Managerに対して宣言するときに必要になります。

この手順では、SUSE Managerサーバに搭載されているものと同じバージョンのオペレーティングシステムをインストールする準備をします。 クライアントに異なるオペレーティングシステムやバージョンをインストールする場合は、**tftpboot-installation-***パッケージを、これが属するディストリビューションから手動で取得する必要があります。 SUSE Managerの［**パッケージ検索**］入力ボックスで、名前が**tftpboot-installation**で始まるパッケージを検索し、そのパッケージの詳細を確認します。ここには、**/var/spacewalk/**以下のローカルパスが表示されます。

8.4.3. 自動インストールのディストリビューションを宣言する

自動インストールファイルを抽出した後の次の手順は、自動インストールのディストリビューションの宣言です。

プロシージャ: 自動インストールのディストリビューションの宣言

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › ディストリビューション**に移動します。
2. ［**ディストリビューションの作成**］をクリックし、次のフィールドに入力します。
   - ［**ディストリビューションラベル**］フィールドに、自動インストール可能なディストリビューションを識別するための名前を入力します。
   - ［**ツリーパス**］フィールドに、SUSE Managerサーバに保存されているインストールメディアへのパスを入力します。
   - 対応する［**ベースチャンネル**］を選択します。 このチャンネルはインストールメディアと一致する必要があります。
   - ［**インストーラ生成**］を選択します。 これはインストールメディアと一致する必要があります。

- オプション: このディストリビューションをブートするときに使用するカーネルオプションを指定します。 カーネルオプションを指定する方法は複数あります。 ここにはディストリビューションに当てはまるオプションのみを追加します。

3. **[自動インストール可能なディストリビューションの作成]**をクリックします。

準備したインストールファイルには、インストールする必要があるパッケージが含まれていない可能性があります。 必要なパッケージが含まれていない場合は、［**カーネルオプション**］フィールドに**useonlinerepo=1**を追加します。

パッケージリポジトリには、署名されていないことがあるメタデータが含まれています。 メタデータが署名されていない場合は、［**カーネルオプション**］フィードに**insecure=1**を追加するか、**Client-configuration › Autoinst-owngpgkey**の説明に従って独自のGPGキーを使用します。

これらの関連のオプションは、フルDVDの代わりに「オンラインインストーラ」ISOイメージを使用する場合や、**tpboot-installation**パッケージを使用する場合などに必要です。

自動インストールのディストリビューションを管理するには、**システム › 自動インストール › ディストリビューション**に移動します。

SUSE Linux Enterpriseクライアントと同じ方法でSUSE Managerプロキシを自動インストールできます。 SUSE Linux Enterpriseインストールメディアを使用して、**SLE-Product-SUSE-Manager-Proxy-4.2-Pool for x86_64**ベースチャンネルを選択していることを確認してください。

8.4.4. ディストリビューションとプロファイルのカーネルオプションの処理

SUSE Managerは、割り当てたカーネルオプションを組み合わせることができます。 これは特別な継承ロジックを使用して行われます。 これに関連するオブジェクト タイプは3つあります。

1. ディストリビューション(または略して "Distros")
2. プロファイル
3. システム

最後のカーネルオプションに影響を与える4つ目の特別なポイントとして、Cobbler設定ファイル**/etc/cobbler/settings.yaml**が挙げられます。 Cobbler設定ファイルは、すべてのディストリビューションにデフォルトのカーネルオプションを定義します。 これは、SUSE Managerのコンテキストではサポートされていません。

カーネルオプションを効果的に管理するには、生の値と解決済みの値を理解することが極めて重要です。

1. **生の値**: これらは、特定のCobbler項目に直接割り当てられ、Cobblerの内部データベースにそのまま保存される値を指します。
2. **解決済みの値**: これらの値は、Cobbler項目の継承階層を考慮して実行時に動的に生成されます。

オプションの前に**!**を付けると、そのオプションは最終的なカーネルコマンドラインから削除されます。

SUSE Managerは、プロファイルとシステムの両方のカーネルオプションを管理します。 そのため、Distroのカーネルオプションのみを編集することができます。

例

基本的な継承の例

ディストリビューションの生の値

```
install=http://uyuni.server/ks/dist/SLES15SP4 self_update=0
```

プロファイルの生の値

```
console=tty1
```

システムの生の値

```
console=ttyS0
```

このプロファイルを継承するシステムの**解決済みの**値

```
install=http://uyuni.server/ks/dist/SLES15SP4 self_update=0 console=ttyS0
```

オプション削除の例

ディストリビューションの生の値

```
install=http://uyuni.server/ks/dist/SLES15SP4 self_update=0
```

プロファイルの生の値

```
console=tty1
```

システムの生の値

```
!self_update
```

このプロファイルを継承するシステムの**解決済みの**値

```
install=http://uyuni.server/ks/dist/SLES15SP4 console=ttyS0
```

8.5. 自動インストールプロファイル

SUSE Manager内では、インストールするクライアントのオペレーティングシステムに応じて、2つの異なる

種類のプロファイルを使用できます。

- SUSE Linux Enterpriseクライアントまたはopen SUSEクライアントの場合、AutoYaSTを使用します。
- Red Hat Enterprise Linuxクライアントの場合、Kickstartを使用します。

自動インストールプロファイルによって、オペレーティングシステムをインストールする方法が決定されます。 たとえば、インストーラに渡す追加のカーネルパラメータを指定できます。

プロファイルの最も重要な部分は、*自動インストールファイル*です。 インストールを手動で実行する場合、パーティション設定、ネットワーク情報、ユーザの詳細などの情報をインストーラに提供する必要があります。 自動インストールファイルは、スクリプト形式でこの情報を提供する方法です。 このタイプのファイルは、*回答ファイル*と呼ばれることもあります。

異なるオペレーティングシステムでクライアントをインストールする場合、AutoYaSTプロファイルとKickstartプロファイルの両方を使用できます。

- プロファイルを宣言する方法については、プロファイルを宣言するを参照してください。
- AutoYaSTプロファイルについては、AutoYaSTプロファイルを参照してください。
- Kickstartプロファイルについては、Kickstartプロファイルを参照してください。

プロファイルに含まれる自動インストールファイルには、変数とコードスニペットを格納できます。 変数とコードスニペットについては、テンプレートの構文を参照してください。

8.5.1. プロファイルを宣言する

自動インストールファイルとディストリビューションの準備ができたら、プロファイルを作成して、SUSE Managerサーバで自動インストールを管理できます。プロファイルにより、選択したこのディストリビューションのインストール方法が決定されます。プロファイルを作成する1つの方法はAutoYaSTファイルまたはKickstartファイルをアップロードする方法です。または、Kickstartのみの場合、Web UIウィザードを使用できます。

プロシージャ: アップロードによる自動インストールプロファイルの作成

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › プロファイル**に移動します。
2. **[キックスタート/AutoYaSTファイルをアップロード]**をクリックします。
3. ［**ラベル**］フィールドにプロファイルの名前を入力します。 スペースは使用しません。
4. ［**自動インストールツリー**］フィールドで、このプロファイルに使用する自動インストールのディストリビューションを選択します。
5. ［**仮想化タイプ**］フィールドで、このプロファイルに使用する仮想化の種類を選択します。または、このプロファイルを使用して新しい仮想マシンを作成しない場合には［**なし**］を選択します。
6. 自動インストールファイルの内容を［**ファイルの内容**］フィールドにコピーするか、または［**アップロードするファイル**］フィールドを使用してファイルを直接アップロードします。

   ここに記載する詳細については、AutoYaSTプロファイルまたはKickstartプロファイルを参照してく

ださい。

7. **[作成]**をクリックしてプロファイルを作成します。

プロシージャ: ウィザードでKickstartプロファイルを作成する

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › プロファイル**に移動します。
2. **[キックスタートプロファイルを作成]**をクリックします。
3. ［**ラベル**］フィールドにプロファイルの名前を入力します。 スペースは使用しません。
4. ［**ベースチャンネル**］フィールドで、このプロファイルに使用するベースチャンネルを選択します。このフィールドは利用できるディストリビューションから入力されます。 必要なベースチャンネルが利用できない場合、ディストリビューションを正しく作成したことを確認してください。
5. ［**仮想化タイプ**］フィールドで、このプロファイルに使用する仮想化の種類を選択します。または、仮想化しない場合には［**なし**］を選択します。
6. **[次へ]**をクリックします。
7. ［**配信ファイルの場所**］で、SUSE Managerサーバにインストールするインストールメディアへのパスを入力します。
8. **[次へ]**をクリックします。
9. クライアントのrootユーザのパスワードを入力します。
10. **[完了]**をクリックします。
11. 新しいプロファイルの詳細を確認し、必要に応じてカスタマイズします。

自動インストールプロファイルを作成している場合、［**このベースチャンネルに最新のツリーを常に使用します**］にチェックを付けることができます。 この設定では、指定ベースチャンネルに関連付けられた最新ディストリビューションをSUSE Managerで自動選択できます。 新しいディストリビューションを後で追加する場合、SUSE Managerは、最後に作成または変更されたディストリビューションを使用します。

仮想化の種類を変更すると、通常、プロファイルのブートローダおよびパーティションオプションを変更する必要があります。 この操作を実行すると、カスタマイズが上書きされます。 新しい設定または変更した設定を保存前に確認します。そのためには、［**パーティション設定**］タブに移動します。

ディストリビューションとプロファイルのカーネルオプションは統合されます。

自動インストールプロファイルの詳細および設定を変更できます。そのためには、**システム › 自動インストール › プロファイル**に移動し、編集するプロファイルの名前をクリックします。 または、**システム › システム一覧**に移動し、プロビジョニングするクライアントを選択し、**プロビジョニング › 自動インストール**サブタブに移動します。

8.5.2. AutoYaSTプロファイル

AutoYaSTプロファイルは、プロファイルを識別する**ラベル**、自動インストールのディストリビューションをポイントする**自動インストールツリー**、さまざまなオプション、最も重要なAutoYaSTインストールファイルで構成されます。

AutoYaSTインストールファイルは、AutoYaSTインストーラに指示を与えるXMLファイルです。AutoYaSTでは、「制御ファイル」と呼ばれます。 AutoYaSTインストールファイルの構文の詳細については、https://doc.opensuse.org/projects/autoyast/#cha-configuration-installation-optionsを参照してください。

SUSEには、独自のカスタムファイルの雛形として使用できるAutoYaSTインストールファイルのテンプレートが用意されています。 このテンプレートは、https://github.com/SUSE/manager-build-profilesの**AutoYaST**ディレクトリにあります。 各プロファイルを使用するにはその前に、一部の変数を設定する必要があります。 スクリプトに含まれている**README**ファイルを確認して、必要な変数を判別してください。 AutoYaSTスクリプトで変数を使用する方法の詳細については、変数を参照してください。

Provided AutoYaST templates do not set any user password. Consider setting up root and other user accounts and passwords or other means of authentication. For more information about user accounts in the AutoYaST profiles, see https://doc.opensuse.org/projects/autoyast/#Configuration-Security-users-and-groups.

SUSE Managerでインストールするための AutoYaSTインストールファイルで、最も重要なセクションを次に示します。

- **<add-on>**では、インストールに子チャンネルを追加できます 。

```
https://doc.opensuse.org/projects/autoyast/#Software-Selections-additional with an
``<add-on>`` exampleを参照してください。
```

- **<general>$SNIPPET('spacewalk/sles_no_signature_checks')</general>**は、署名のチェックを無効にします。
- **<software>**によって、Unified Installerに製品を指定できます。

```
https://doc.opensuse.org/projects/autoyast/#Software-Selections-additional with an
``<add-on>`` exampleを参照してください。
```

- **<init-scripts config:type="list">$SNIPPET('spacewalk/minion_script ')</init-scripts>**は、クライアントをSaltクライアントとしてSUSE Managerに登録できるようにします。

AutoYaSTの詳細については、https://doc.opensuse.org/projects/autoyast/を参照してください。

AutoYaSTに代わるSaltベースの最近のプロファイルには、Yomiがあります。 Yomiについては、**Specialized-guides › Salt**を参照してください。

8.5.3. キックスタートプロファイル

Kickstartプロファイルには、多数の設定オプションがあります。 プロファイルを作成するには、プロファイルをアップロードするか、専用のウィザードを使用します。

Kickstartプロファイルでは、ファイル保持一覧を使用できます。 Kickstartで再インストールするクライアン

トにあるカスタム設定ファイルが多数ある場合、リストにしてこれらのファイルを保存し、そのリストをKickstartプロファイルに関連付けることができます。

プロシージャ: ファイル保持一覧の作成

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › ファイル保持**に移動し、**[ファイル保持一覧の作成]**をクリックします。
2. 適切なラベルを入力し、保存するすべてのファイルおよびディレクトリへの絶対パスをリストします。
3. **[一覧の作成]**をクリックします。
4. Kickstartプロファイルにファイル保持一覧を含めます。
5. **システム › 自動インストール › プロファイル**に移動して編集するプロファイルを選択し、**システムの詳細 › ファイル保持**サブタブに移動して、含めるファイル保持一覧を選択します。

ファイル保持一覧の合計サイズは1 MBに制限されています。 **/dev/hda1**や**/dev/sda1**などの特殊なデバイスは保持できません。 ファイル名とディレクトリ名のみ使用できます。正規表現のワイルドカードは使用できません。

Kickstartの詳細については、Red Hatのドキュメントを参照してください。

8.5.4. テンプレートの構文

インストールファイルの一部は、インストール中に置き換えられます。 変数は単一の値に置き換えられ、コードスニペットはテキストのセクション全体に置き換えられます。 エスケープされた記号やセクションは置き換えられません。

CobblerはCheetahと呼ばれるテンプレートエンジンを使用して、このような置き換えを実行できます。このメカニズムにより、システムごとにプロファイルを手動で作成する必要なく、多数のシステムを再インストールできます。

自動インストールの変数やコードスニペットは、SUSE Manager Web UI内で作成できます。 プロファイル内の［**自動インストールファイル**］タブでは、置き換えの結果を確認できます。

- 変数については、変数を参照してください。
- コードスニペットについては、コードスニペットを参照してください。
- エスケープ記号またはテキストブロックについては、エスケープを参照してください。

変数

自動インストールの変数は、KickstartプロファイルおよびAutoYaSTプロファイルに値を代入するために使用できます。 変数を定義するには、プロファイルから［**変数**］サブタブに移動し、テキストボックスで**name=value**ペアを作成します。

たとえば、クライアントのIPアドレスを格納する変数と、ゲートウェイのアドレスを格納する変数を作成できます。 次に、作成した変数は、同じプロファイルからインストールされるすべてのクライアントに対して定義できます。 このためには、［**変数**］テキストボックスに次の行を追加します。

```
ipaddr=192.168.0.28
gateway=192.168.0.1
```

変数を使用するには、プロファイルで値の前に**$**記号を付けて値を代入します。　たとえば、Kickstartファイルの［**ネットワーク**］部分は次のようになることがあります。

```
network --bootproto=static --device=eth0 --onboot=on --ip=$ipaddr \
  --gateway=$gateway
```

$ipaddr**は**192.168.0.28**に解決され、**$gatewayは**192.168.0.1**に解決されます。

インストールファイルでは、変数は階層的に使用します。　システム変数はプロファイル変数より優先され、プロファイル変数はディストリビューション変数より優先されます。

コードスニペット

SUSE Managerには、多数の定義済みコードスニペットが付属しています。**システム › 自動インストール › 自動インストールスニペット**に移動し、既存のスニペットの一覧を表示します。

自動インストールファイルの**$SNIPPET()**マクロに挿入してスニペットを使用します。　たとえば、Kickstartでは次のようになります。

```
$SNIPPET('spacewalk/rhel_register_script')
```

または、AutoYaSTでは次のようになります。

```
<init-scripts config:type="list">
  $SNIPPET('spacewalk/sles_register_script')
</init-scripts>
```

このマクロはCobblerによって解析され、スニペットの内容に置き換えられます。独自のコードスニペットを保存して、後で自動インストールファイルで使用することもできます。**[スニペットの作成]**をクリックして、新しいコードスニペットを作成します。

この例では、一般的なハードドライブのパーティション設定のKickstartスニペットが設定されます。

```
clearpart --all
part /boot --fstype ext3 --size=150 --asprimary
part / --fstype ext3 --size=40000 --asprimary
part swap --recommended

part pv.00 --size=1 --grow

volgroup vg00 pv.00
logvol /var --name=var vgname=vg00 --fstype ext3 --size=5000
```

たとえば、次のようにスニペットを使用します。

```
$SNIPPET('my_partition')
```

エスケープ

自動インストールファイルには、**$(example)**のようなシェルスクリプト変数が含まれています。コンテンツはバックスラッシュ(円記号)でエスケープする必要があります**\$(example)**。 **$**記号をエスケープすると、テンプレートエンジンは記号を内部変数として評価しなくなります。

コードフラグメントやスクリプトなどのテキストブロックは、**\#raw**ディレクティブおよび**\#end raw**ディレクティブで囲むことによってエスケープできます。 次に例を示します。

```
#raw
#!/bin/bash
for i in {0..2}; do
 echo "$i - Hello World!"
done
#end raw
```

#記号の後にスペースがある行はコメントとして扱われるため、評価されません。 次に例を示します。

```
# start some section (this is a comment)
echo "Hello, world"
# end some section (this is a comment)
```

8.6. 無人プロビジョニング

「ベアメタル」機能を使用すると、汎用PXEブートイメージを使用して、ローカルネットワークに接続した直後に新しいコンピュータを登録することができます。 次に、SUSE Manager Web UIに移動して、このコンピュータにプロファイルを割り当てます。 次にクライアントをブートしたときに、そのプロファイルに従ってオペレーティングシステムがインストールされます。 ベアメタルプロビジョニングについては、ベアメタルプロビジョニングを参照してください。

ベアメタル機能を使用したくない場合は、SUSE Managerでシステムを手動で宣言することもできます。SUSE Manager APIを使用すると、ベアメタル機能で収集されたかのように、システムに対するシステムレコードを作成することができます。 APIを使用したシステムの宣言については、システムレコードを手動で作成するを参照してください。

8.6.1. ベアメタルプロビジョニング

ベアメタルプロビジョニングオプションを有効にしている場合、SUSE Managerネットワークに接続されているクライアントは、電源をオンにするとすぐに組織に自動追加されます。 この処理が完了すると、クライアントはシャットダウンし、［**システム**］一覧に表示され、インストールする準備ができます。

手順: ベアメタル機能を有効にする

1. SUSE ManagerのWeb UIで、**管理 › マネージャ設定 › ベアメタルシステム**に移動します。
2. **[この組織に対する追加の有効化]**をクリックします。

電源をオンにした新しいクライアントは、ベアメタル機能を有効にした管理者が所属している組織に追加されます。 このようなクライアントは「ブートストラップ」タイプであり、通常のクライアントにするためにプロビジョニングが必要です。

新しいクライアントを追加する組織を変更するには、ベアメタル機能を無効にし、新しい組織の管理者としてログインし、機能を再有効化します。 登録済みのシステムは別の組織に移行できます。そのためには［**移行**］タブを使用します。

この方法で登録するクライアントでは、システムセットマネージャ(SSM)を使用できます。 ただし、オペレーティングシステムがまだインストールされていないため、このようなクライアントでは使用できないSSM機能があります。 これは、この方法で登録したシステムを含む混合セットにも当てはまります。 セットのすべてのクライアントがプロビジョニングされると、すべての機能をセットで使用できるようになります。 SSMの詳細については、**Client-configuration › System-set-manager**を参照してください。

手順: 「ブートストラップ」タイプのクライアントをプロビジョニングする

1. SUSE Manager Web UIで、［**システム**］に移動し、プロビジョニングするクライアントを選択し、**プロビジョニング › 自動インストール**タブに移動します。
2. 使用するAutoYaSTプロファイルを選択し、**[PXEインストール設定の作成]**をクリックします。 このオプションを選択すると、Cobblerでシステムエントリが作成されます。
3. クライアントの電源をオンにします。

サーバは、TFTPを使用して新しいクライアントをプロビジョニングするため、プロビジョニングを正常に実行するには適切なポートおよびネットワークが正しく設定されている必要があります。

8.6.2. システムレコードを手動で作成する

APIコールを使用して、MACアドレスによって識別されるクライアントと自動インストールプロファイル間の関連付けを宣言できます。 次にシステムを再起動したときに、指定したプロファイルに基づいてインストールが開始されます。

プロシージャ: 手動で宣言したプロファイルからの再インインストール

1. SUSE Managerサーバのコマンドプロンプトで、**system.createSystemRecord** APIコールを使用します。 この例では、**name**をクライアントの名前に、**<profile>**をプロファイルラベルに、**<iface>**を**eth0**などのクライアント上のインタフェース名に、**<hw_addr>**を**00:25:22:71:e7:c6**などのクライアントのハードウェアアドレスに置き換えます。

```
$ spacecmd api -- --args '["<name>", "<profile>", "", "", \
  [ {"name": "<iface>", "mac": "<hw_addr>"} ]]' \
  system.createSystemRecord
```

2. クライアントの電源をオンにします。 ネットワークからブートすると、インストール用の正しいプロファイルが選択されます。

このコマンドによって、Cobblerでシステムレコードが作成されます。カーネルオプション、クライアントのIPアドレス、クライアントのドメイン名など、追加のパラメータを指定することもできます。詳細につい

ては、**createSystemRecordコール**のAPIドキュメントを参照してください。

8.7. 独自のGPGキーを使用する

自動インストールのために使用しているリポジトリに署名されていないメタデータがある場合は、通常、自動インストールのディストリビューションのオプションとして**insecure=1**カーネルパラメータを使用し、AutoYaSTインストールファイルで**spacewalk/sles_no_signature_checks**コードスニペットを使用する必要があります。

より安全な代替方法は、独自のGPGキーを提供することです。

この操作は、SUSEクライアントにのみ適用されます。

手順: 独自のGPGキーを追加する

1. GPGキーを作成します。
2. このキーを使用して、パッケージのメタデータに署名します。
3. インストールメディアの初期RAMディスクにこのキーを追加します。
   - キーを作成し、そのキーを使用してメタデータに署名する方法については、**Administration › Repo-metadata**を参照してください。
   - ネットワークブートに使用するインストールメディアにキーを追加する方法については、PXEブート用の独自のGPGキーを参照してください。
   - CD-ROMからのブートに使用するインストールメディアにキーを追加する方法については、CD-ROM内の独自のGPGキーを参照してください。

新しいGPGキーを使用してメタデータに署名した場合、オンボード済みのクライアントはこの新しいキーを認識しません。 クライアントを登録する前に、メタデータに署名することが理想的です。

リポジトリを使用するオンボード済みのクライアントの場合、修正方法は、そのクライアントでGPGキーのチェックを無効にすることです。

8.7.1. PXEブート用の独自のGPGキー

PXEブートプロセスで使用される初期RAMディスク(**initrd**)には、通常SUSEのGPGキーのみが格納されています。 パッケージをチェックするために使用できるように、このファイルに独自のキーを追加する必要があります。

プロシージャ: 初期RAMディスクにGPGキーを追加する

1. GPGキーを見つけるためにブートプロセス中に使用されるものと同じパスにディレクトリを作成します。

```
mkdir -p tftproot/usr/lib/rpm/gnupg/keys
```

2. **.asc**サフィックスを付けてこのディレクトリにGPGキーをコピーします。

```
cp /srv/www/htdocs/pub/mgr-gpg-pub.key tftproot/usr/lib/rpm/gnupg/keys/mgr-gpg-
pub.asc
```

3. 最上位のディレクトリ内で、コンテンツをパッケージ化し、インストールメディアファイルの一部である**initrd**に追加します。

```
cd tftproot
find . | cpio -o -H newc | xz --check=crc32 -c >> /path/to/initrd
```

8.7.2. CD-ROM内の独自のGPGキー

mksusecdユーティリティでインストールイメージを修正できます。 このユーティリティは、Development Toolsモジュールに含まれています。

プロシージャ: インストールISOイメージにGPGキーを追加する

1. GPGキーを見つけるためにブートプロセス中に使用されるものと同じパスにディレクトリを作成します。

```
mkdir -p initrdroot/usr/lib/rpm/gnupg/keys
```

2. **.asc**サフィックスを付けてこのディレクトリにGPGキーをコピーします。

```
cp /srv/www/htdocs/pub/mgr-gpg-pub.key initrdroot/usr/lib/rpm/gnupg/keys/mgr-gpg-
pub.asc
```

3. **mksusecd**で既存のISOイメージを修正します。

```
mksusecd --create <new-image>.iso --initrd initrdroot/ <old-image>.iso
```

Chapter 9. 仮想化

SUSE Managerを使用して、通常の従来のクライアントまたはSaltクライアントに加えて、仮想化されたクライアントを管理できます。 この種のインストールでは、仮想ホストは、SUSE Managerサーバにインストールされ、任意の数の仮想ゲストを管理します。 このインストールを選択すると、複数の仮想ホストをインストールし、ゲストのグループを管理できます。

仮想化されたクライアントにある機能の範囲は、選択したサードパーティ仮想化プロバイダによって決まります。

XenおよびKVMのホストおよびゲストはSUSE Managerで直接管理できます。 そうすると、AutoYaSTまたはKickstartを使用してホストおよびゲストを自動インストールし、Web UIでゲストを管理できます。

VMware vSphere、Nutanix AHVなどのVMwareでは、SUSE Managerは、仮想ホストマネージャ(VHM)を設定し、VMを制御する必要があります。 そうするとホストおよびゲストを制御できますが、XenおよびKVMで可能な制御方法より限定されます。 SUSE Managerは、VMware vSphereまたはNutanix AHVでVMを作成または編集できません。

その他のサードパーティ仮想化プロバイダはSUSE Managerでは直接サポートされていません。 ただし、プロバイダでVMのJSON設定ファイルをエクスポートできる場合、その設定ファイルをSUSE Managerにアップロードし、VHMで管理できます。

VHMを使用して仮想化を管理する方法の詳細については、**Client-configuration › Vhm**を参照してください。

9.1. 仮想化ホストの管理

開始前に、仮想化ホストとして使用するクライアントで［**仮想化ホスト**］システムタイプが割り当てられていることを確認してください。 従来のクライアントとSaltクライアントの両方を仮想化ホストとして使用できます。 **システム › システム一覧**に移動し、仮想化ホストとして使用するクライアントの名前をクリックします。 ［**仮想化ホスト**］システムタイプがリストされていない場合は、**仮想化ホスト**式を初期化します。 詳細については、client-configuration:virt-xenkvm.pdfを参照してください。

クライアントに［**仮想化ホスト**］システムタイプがある場合、クライアントの［システムの詳細］ページで［**仮想化**］タブを使用できます。 ［**仮想化**］タブでは、仮想ゲストを作成して管理し、ストレージプールおよび仮想ネットワークを管理できます。

9.2. 仮想ゲストの作成

SUSE ManagerのWeb UI内で仮想ゲストを仮想化ホストに追加できます。

プロシージャ: 仮想ゲストの作成

1. SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、仮想化ホストの名前をクリックし、［**仮想化**］タブに移動します。
2. ［**全般**］セクションで、次の詳細を入力します。

- ［**ゲスト**］サブタブで、**[Create Guest]**（ゲストの作成）をクリックします。
- ［**名前**］フィールドにゲストの名前を入力します。
- ［**ハイパーバイザ**］フィールドで、使用するハイパーバイザを選択します。
- ［**仮想マシンのタイプ**］フィールドで、完全仮想化または部分的仮想化のいずれかを選択します。
- ［**最大メモリ**］フィールドに、ゲストディスクの最大サイズ制限(MiB単位)を入力します。
- ［**仮想CPU数**］で、ゲストのvCPUの数を入力します。
- ［**アーキテクチャ**］フィールドで、ゲストで使用するエミュレートCPUアーキテクチャを選択します。 デフォルトでは、選択したアーキテクチャは仮想ホストと一致しています。
- ［**自動インストールプロファイル**］フィールドで、ゲストのインストールに使用する自動インストールツールを選択します。 自動インストールを使用しない場合、このフィールドを空白のままにします。

3. ［**ディスク**］セクションで、クライアントで使用する仮想ディスクの詳細を入力します。 ［**ソーステンプレートのイメージURL**］フィールドで、オペレーティングシステムのイメージへのパスを入力したことを確認してください。 これを実行しないと、ゲストのディスクは空になります。
4. ［**ネットワーク**］セクションで、クライアントで使用する仮想ネットワークインタフェースの詳細を入力します。［**MACアドレス**］フィールドを空白のままにして、MACアドレスを生成します。
5. ［**グラフィックス**］セクションで、クライアントで使用するグラフィックスドライバの詳細を入力します。
6. ゲストを作成する時間をスケジュールし、**[作成]**をクリックしてゲストを作成します。
7. 新しい仮想ゲストは、正常に作成されるとすぐに開始されます。

SUSE Manager Web UI内のペースメーカークラスタに仮想ゲストを追加することもできます。

プロシージャ: クラスタ管理対象仮想ゲストの作成

1. 次の追加項目を使用して、クラスタのノードの1つで**仮想ゲストの作成**プロシージャに従います。
   - ［**クラスタリソースとして定義**］フィールドがチェックされていることを確認します。
   - ［**VM定義用のクラスタ共有フォルダへのパス**］フィールドに、ゲスト構成が保存されるすべてのクラスタノードによって共有されるフォルダへのパスを入力します。
   - すべてのディスクが、すべてのクラスタノードによって共有されるストレージプールに配置されていることを確認してください。

クラスタによって管理される仮想ゲストは、ライブマイグレーションできます。

9.3. SUSEのサポートとVMゾーン

パブリッククラウドプロバイダは、リージョンを使用して、仮想マシンを提供しているデータセンターにおける地理上の物理的場所を定義します。 たとえば、**米国東部**や**アジア**です。

リージョンがさらにゾーンに分割されます。 たとえば、**米国東部**リージョンには、**us-east-2a**、**us-east-2b**

などが含まれる場合があります。

SUSEは、仮想マシンのゾーンを使用して、提供する適切なサブスクリプションを決定します。 すべてのVMが同じゾーンで提供される場合、**1-2仮想マシン**サブスクリプションの条件になります。

VMが異なるゾーンで提供される場合、それらが同じリージョン内であっても、**1-2仮想マシン**サブスクリプションの条件を満たさない場合があります。 この場合、サブスクリプションを注意深く確認してください。

BYOSインスタンス(Bring-your-own-subscription)の場合、インストールされているすべての製品がサブスクリプションマッチャに渡されます。パブリッククラウドのインスタンスがPAYG (Pay-as-you-go)の場合、そのベース製品はサブスクリプションマッチャのカウントから除外されます。

インスタンスが PAYGかBYOSかの計算は、登録時またはハードウェア更新アクション実行時に行われます。

詳細については、https://www.suse.com/products/terms_and_conditions.pdfを参照するか、SUSEにお問い合わせください。

9.4. XenおよびKVMを使用した仮想化

XenおよびKVMの仮想化クライアントはSUSE Managerで直接管理できます。

まず、SUSE Managerサーバで仮想ホストを設定する必要があります。 今後の仮想ホストおよび仮想ゲストのAutoYaSTまたはKickstartを使用して自動インストールを設定できます。

このセクションでは、インストール後に仮想ゲストを管理する方法についても説明します。

9.4.1. ホストの設定

VMホストでXenまたはKVMを設定する方法は、関連するゲストで使用するオペレーティングシステムによって決まります。

SUSEオペレーティングシステムについては、https://documentation.suse.com/sles/15-SP4/html/SLES-all/book-virtualization.htmlでSLES仮想化に関するガイドを参照してください。

Red Hat Enterprise Linuxオペレーティングシステムについては、使用バージョンに応じてRed Hatのドキュメントを参照してください。

仮想化ホスト式は、ホストの初期化を支援します。 詳細については、client-configuration:virt-xenkvm.pdfを参照してください。

背景情報

SUSE Managerは、**libvirt**を使用してゲストをインストールして管理します。 ホストに**libvirt-daemon**パッケージがインストールされている必要があります。 ほとんどの場合、デフォルト設定で十分であるため、調整する必要はありません。 ただし、非rootユーザとしてゲストのVNCコンソールにアクセスする場合、設定

変更を実行する必要があります。　設定方法の詳細については、ご使用のオペレーティングシステム用のマニュアルを参照してください。

SUSE Managerサーバでブートストラップスクリプトが必要です。 ブートストラップスクリプトには、ホストのアクティベーションキーを含める必要があります。 GPGキーも含めてセキュリティを強化することをお勧めします。　ブートストラップスクリプトの作成については、**Client-configuration › Registration-bootstrap**を参照してください。

ブートストラップスクリプトの準備ができたら、ホストで実行し、SUSE Managerサーバに登録します。 クライアントの登録の詳細については、**Client-configuration › Registration-overview**を参照してください。

仮想化ホストの初期化

仮想化ホスト式で、ホストを初期化します。

プロシージャ: 仮想化ホストの初期化

1. SUSE ManagerのWeb UIで、ホストの［**システムの詳細**］ページに移動し、［**式** ］タブをクリックします。
2. **仮想化ホスト**式を選択し、**[保存]**をクリックします。
3. **仮想化ホスト**サブタブをクリックします。
4. 設定を確認して、**[式の保存]**をクリックします。
5. 変更を有効にするには、Highstateを適用します。
6. **salt-minion**サービスを再起動し、新しい設定を有効にします。

```
systemctl restart salt-minion
```

従来のクライアントの場合、デフォルトでは、VMホストは**rhnsd**サービスを使用して、スケジュールされたアクションを確認します。　確認は4時間ごとに実行され、多数のクライアントが存在する環境の負荷を均等化します。 そのため、アクションの実行前に、最大4時間の遅延が発生する可能性があります。 VMゲストを管理している場合、この長時間の遅延は常に(特にゲストの再起動の際には)理想的ではありません。　この問題に対処するには、**rhnsd**サービスを無効にして**osad**サービスを有効にできます。 **osad**サービスは、jabberプロトコルを使用してコマンドを受け取り、すぐにコマンドを実行します。

rhnsdサービスを無効にして**osad**デーモンを有効にするには、次のコマンドをrootユーザとして実行します。

```
service rhnsd stop
service rhnsd disable
```

```
service osad enable
service osad start
```

9.4.2. VMゲストの自動インストール

AutoYaSTまたはKickstartを使用して、XenおよびKVMのゲストを自動的にインストールして登録できます。SUSE Linux EnterpriseまたはopenSUSEクライアントの場合はAutoYaSTを使用し、Red Hat Enterprise Linuxクライアントの場合はKickstartを使用してください。

ゲストを登録するVMホストと各ゲストのアクティベーション キーが必要です。

アクティベーションキーには、**プロビジョニング**のエンタイトルメントと**仮想化プラットフォーム**のエンタイトルメントが必要です。 アクティベーションキーは、**mgr-virtualization-host**パッケージおよび**mgr-osad**パッケージにもアクセスできる必要があります。 アクティベーションキーの作成の詳細については、**Client-configuration › Activation-keys**を参照してください。

インストール後にSUSE Managerでゲストを自動的に登録する場合、ブートストラップスクリプトを作成する必要があります。 ブートストラップスクリプトの作成については、**Client-configuration › Registration-bootstrap**を参照してください。

VMゲストの自動インストールは、従来のクライアントとして設定されている場合のみ機能します。 Saltクライアントはテンプレートディスクイメージを使用して作成できます。AutoYaSTまたはKickstartを使用して作成することはできません。

自動インストール可能なディストリビューションの作成

SUSE Managerからクライアントを自動インストールできるようにするには、自動インストール可能なディストリビューションをVMホストに作成する必要があります。 このディストリビューションは、マウントされたローカルディレクトリやリモートディレクトリ、またはループマウントされたISOイメージから使用できます。

自動インストール可能なディストリビューションの設定は、SLESまたはRed Hat Enterprise Linuxオペレーティングシステムをゲストで使用しているかどうかによって異なります。 Red Hat Enterprise Linuxインストールのパッケージは、関連するベースチャンネルからフェッチされます。 SUSEシステムをインストールするパッケージは、自動インストール可能なディストリビューションからフェッチされます。 したがって、SLESシステムでは、自動インストール可能なディストリビューションは、完全なインストールソースである必要があります。

表 58. 自動インストール可能なディストリビューションのパス

オペレーティングシステムの種類	カーネルの場所	initrdの場所
Red Hat Enterprise Linux	**images/pxeboot/vmlinuz**	**images/pxeboot/initrd.img**
SLES	**boot/<arch>/loader/initrd**	**boot/<arch>/loader/linux**

すべてのケースで、ベースチャンネルが自動インストール可能なディストリビューションと一致していることを確認してください。

始める前に、使用しているVMホストでインストールメディアを使用できることを確認してください。 これは、ネットワークリソース上、ローカルディレクトリ内、またはループマウントされたISOイメージ内にある場合があります。 また、すべてのファイルおよびディレクトリが全世界で読み取ることができることを確認

してください。

プロシージャ: 自動インストールのディストリビューションの作成

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › ディストリビューション**に移動し、**[ディストリビューションの作成]**をクリックします。
2. ［**自動インストール可能なディストリビューションの作成**］セクションで、次のパラメータを使用します。
   - ［**ディストリビューションラベル**］セクションに、ディストリビューションの固有の名前を入力します。 半角の英字、数字、ハイフン(-)、ピリオド(.)、および下線(_)のみを使用し、5文字以上にしてください。
   - ［**ツリーパス**］フィールドに、インストールソースへの絶対パスを入力します。
   - ［**ベースチャンネル**］フィールドで、インストールソースと一致するチャンネルを選択します。 このチャンネルは、非SUSEインストール環境用のパッケージソースとして使用されます。
   - ［**インストーラ生成**］フィールドで、インストールソースと一致するオペレーティングシステムのバージョンを選択します。
   - ［**カーネルオプション**］フィールドに、インストールでブート時にカーネルに渡すオプションを入力します。 **install=**パラメータおよび**self_update=0 pt.options=self_update**パラメータはデフォルトで追加されます。
   - インストールしたシステムを初めてブートするときにカーネルに渡すオプションを［**カーネルの後のオプション**］セクションに入力します。
3. **[自動インストール可能なディストリビューションの作成]**をクリックして保存します。

自動インストール可能なディストリビューションを作成したら、これを編集できます。そのためには、**システム › 自動インストール › ディストリビューション**に移動し、編集するディストリビューションを選択します。

自動インストールプロファイルの作成およびアップロード

自動インストールプロファイルには、システムをインストールするために必要なインストールデータおよび設定データがすべて含まれています。 インストール完了後に実行するスクリプトを含めることもできます。

KickstartプロファイルはSUSE ManagerのWeb UIを使用して作成できます。そのためには、**システム › 自動インストール › プロファイル**に移動し、**[新しいキックスタートプロファイルを作成]**をクリックし、プロンプトに従って操作します。

AutoYaSTまたはKickstartの自動インストールプロファイルを手動で作成することもできます。 SUSEには、独自のカスタムファイルの雛形として使用できるAutoYaSTインストールファイルのテンプレートが用意されています。 これは、https://github.com/SUSE/manager-build-profilesにあります。

AutoYaSTを使用してSLESをインストールする場合、次のスニペットも含める必要があります。

```
<products config:type="list">
  <listentry>SLES</listentry>
```

```
</products>
```

- AutoYaSTの詳細については、client-configuration:autoinst-profiles.pdfを参照してください。
- Kickstartについては、client-configuration:autoinst-profiles.pdfを参照するか、Red Hatのインストール関連ドキュメントを参照してください。

プロシージャ: 自動インストールプロファイルのアップロード

1. SUSE ManagerのWeb UIで、**システム › 自動インストール › プロファイル**に移動し、**[キックスタート/Autoyastファイルをアップロード]**をクリックします。
2. ［**自動インストールプロファイルを作成**］セクションで、次のパラメータを使用します。
   - ［**ラベル**］フィールドにプロファイルの一意の名前を入力します。 半角の英字、数字、ハイフン(-)、ピリオド(.)、および下線(_)のみを使用し、7文字以上にしてください。
   - ［**自動インストールツリー**］フィールドで、前に作成した自動インストール可能なディストリビューションを選択します。
   - ［**仮想化タイプ**］フィールドで、関連するゲストの種類を選択します(**KVM仮想化ゲスト**など)。 ここでは、［**Xen仮想化ホスト**］を選択しないでください。
   - オプション: 自動インストールプロファイルを手動で作成する場合、［**ファイルの内容**］フィールドに直接入力できます。 ファイルを作成済みの場合、［**ファイルの内容**］フィールドを空白のままにします。
   - ［**アップロードするファイル**］フィールドで、**[Choose File]**（ファイルの選択）をクリックし、システムダイアログを使用して、アップロードするファイルを選択します。 ファイルが正常にアップロードされると、ファイル名が［**アップロードするファイル**］フィールドに表示されます。
   - アップロードしたファイルの内容が［**ファイルの内容**］フィールドに表示されます。 編集する必要がある場合、直接編集できます。
3. **[作成]**をクリックして変更を保存し、プロファイルを保存します。

自動インストールプロファイルを作成したら、これを編集できます。そのためには、**システム › 自動インストール › プロファイル**に移動し、編集するプロファイルを選択します。**[作成]**をクリックして、必要な変更を行い、設定を保存します。

既存のKickstartプロファイルの［**仮想化タイプ**］を変更する場合、ブートローダおよびパーティションのオプションも変更する場合があり、カスタム設定を上書きすることもあります。 ［**パーティション設定**］タブを注意深く確認して、変更前にこれらの設定を確認してください。

ゲストを自動的に登録する

VMゲストを自動的にインストールする場合、ゲストはSUSE Managerには登録されません。 ゲストをインストールしてすぐに自動的に登録する場合、ブートストラップスクリプトを呼び出してゲストを登録する自動インストールプロファイルにセクションを追加できます。

このセクションでは、ブートストラップスクリプトを既存のAutoYaSTプロファイルに追加する手順について

説明します。

ブートストラップスクリプトの作成の詳細については、**Client-configuration › Registration-bootstrap**を参照してください。 Kickstartでこの作業を行う方法については、Red Hatのインストール関連ドキュメントを参照してください。

プロシージャ: ブートストラップスクリプトをAutoYaSTプロファイルに追加する

1. 登録するVMゲストのアクティベーションキーがブートストラップスクリプトに含まれていることを確認してください。これはホストの**/srv/www/htdocs/pub/bootstrap_vm_guests.sh**にあります。
2. SUSE ManagerのWeb UIで、**システム › 自動インストール › プロファイル**に移動し、このスクリプトを関連付けるAutoYaSTプロファイルを選択します。
3. ［**ファイルの内容**］フィールドで、次のスニペットをファイルの末尾(**</profile>**タグの直前)に追加します。 スニペットのIPアドレス例**192.168.1.1**を、使用中のSUSE Managerサーバの正しいIPアドレスに置き換えてください。

```
<scripts>
  <init-scripts config:type="list">
    <script>
      <interpreter>shell</interpreter>
      <location>
        http://192.168.1.1/pub/bootstrap/bootstrap_vm_guests.sh
      </location>
    </script>
  </init-scripts>
</scripts>
```

4. ［**更新**］をクリックして変更を保存します。

AutoYaSTプロファイルに**<scripts>**セクションがすでに含まれている場合、2つ目のセクションを追加しないでください。 既存の**<scripts>**セクション内にブートストラップスニペットを配置します。

VMゲストの自動インストール

すべての設定が完了したら、VMゲストの自動インストールを開始できます。

各VMホストが同時にインストールできるゲストは1つだけです。 複数の自動インストールをスケジュールしている場合、前のインストールが完了する前に次のインストールが始まらないようにスケジュールしてください。 ゲストのインストールが別のインストールの実行中に開始すると、実行中のインストールはキャンセルされます。

1. SUSE ManagerのWeb UIで、**システム › 概要**に移動し、ゲストをインストールするVMホストを選択します。
2. ［**仮想化**］タブ、［**プロビジョニング**］サブタブに移動します。
3. 使用する自動インストールプロファイルを選択し、ゲストの一意の名前を指定します。
4. 該当する場合にはプロキシを選択し、スケジュールを入力します。

5. ゲストのハードウェアのプロファイルおよび設定オプションを変更するには、**[高度なオプション]**をクリックします。
6. **[自動インストールをスケジュールしてから終了する]**をクリックして完了します。

9.4.3. VMゲストの管理

SUSE ManagerのWeb UIを使用して、CPUやメモリの割り当て調整、シャットダウン、再起動のようなアクションなど、VMゲストを管理できます。

そのためには、XenまたはKVM VMホストをSUSE Managerサーバに登録し、**libvirtd**サービスをホストで実行する必要があります。 従来のクライアントでは、**mgr-cfg-actions**パッケージをSUSE Managerサーバにインストールする必要もあります。

SUSE ManagerのWeb UIで、**システム › システム一覧**に移動し、管理するゲストのVMホストをクリックします。 ［**仮想化**］タブに移動し、このホストに登録されているすべてのゲストを表示し、管理機能にアクセスします。

Web UIを使用してVMゲストを管理する方法の詳細については、**Reference › Systems**を参照してください。

Chapter 10. 仮想ホストマネージャ

仮想ホストマネージャ(VHM)は、さまざまなクライアントの種類から情報を収集するために使用します。

VHMを使用して、プライベートクラウドまたはパブリッククラウドのインスタンスを収集し仮想化グループに編成できます。　このように編成された仮想化クライアントを使用して、Taskomaticは、クライアントのデータを収集し、SUSE ManagerのWeb UIに表示します。 VHMを使用すると、仮想化されたクライアントでサブスクリプションマッチングを使用することもできます。

SUSE　Managerサーバに VHMを作成して使用し、使用可能なパブリッククラウドのインスタンスを評価できます。 VHMを使用して、Kubernetesで作成したクラスタを管理することもできます。

- Amazon Web ServicesでVHMを使用する方法の詳細については、**Client-configuration › Vhm-aws**を参照してください。
- Microsoft AzureでVHMを使用する方法の詳細については、**Client-configuration › Vhm-azure**を参照してください。
- Google Compute EngineでVHMを使用する方法の詳細については、**Client-configuration › Vhm-gce**を参照してください。
- KubernetesでVHMを使用する方法の詳細については、**Client-configuration › Vhm-kubernetes**を参照してください。
- NutanixでVHMを使用する方法の詳細については、**Client-configuration › Vhm-nutanix**を参照してください。
- VMWare vSphereでVHMを使用する方法の詳細については、**Client-configuration › Vhm-vmware**を参照してください。
- その他のホストでVHMを使用する方法の詳細については、**Client-configuration › Vhm-file**を参照してください。

SUSE　Managerの仮想化サーバエンタイトルメントを持つホストごとに仮想化アドオンサブスクリプションが必要です。

10.1. VHMおよびAmazon Web Services

仮想ホストマネージャ(VHM)を使用して、Amazon Web Services (AWS)からインスタンスを収集できます。

VHMを使用すると、SUSE Managerは、クラスタに関する情報を取得して報告できます。 VHMの詳細については、**Client-configuration › Vhm**を参照してください。

10.1.1. Amazon EC2 VHMの作成

仮想ホストマネージャ(VHM)はSUSE Managerサーバ上で動作します。

virtual-host-gatherer-libcloudパッケージをSUSE　Managerサーバにインストール済みであることを確認してください。

プロシージャ: Amazon EC2 VHMの作成

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. **[作成]**をクリックし、ドロップダウンメニューから［**Amazon EC2**］を選択します。
3. ［**Add an Amazon EC2 Virtual Host Manager**］（Amazon EC2仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。
   - ［**ラベル**］フィールドにVHMのカスタム名を入力します。
   - ［**Access Key ID**］（アクセスキーID）フィールドに、Amazonが提供するアクセスキーIDを入力します。
   - ［**Secret Access Key**］（秘密アクセス鍵）フィールドに、Amazonインスタンスに関連付けられた秘密アクセス鍵を入力します。
   - ［**Region**］（リージョン）フィールドに、使用するリージョンを入力します。
   - ［**Zone**］（ゾーン）フィールドに、VMが存在するゾーンを入力します。 これは、サブスクリプションマッチングを動作させるために必要です。 リージョンおよびゾーンの設定の詳細については、client-configuration:virtualization.pdfを参照してください。
4. **[作成]**をクリックして変更を保存し、VHMを作成します。
5. ［**仮想ホストマネージャ**］ページで、新しいVHMを選択します。
6. ［**プロパティ**］ページで、**[データの更新]**をクリックし、新しいVHMを評価します。

評価されたオブジェクトおよびリソースを表示するには、**システム › システム一覧 › 仮想システム**に移動します。

Amazonパブリッククラウドで動作しているインスタンスは、UUIDをSUSE Managerサーバに報告します。その際のフォーマットは、**i**に17桁の16進数をつなげたものです。

```
I1234567890abcdef0
```

10.1.2. 仮想ホストマネージャのAWS許可

セキュリティ上の理由から、タスクを実行するために可能な限り最小限の権限を常に付与してください。AWSに接続するユーザに過度な許可を持つアクセスキーを使用することはお勧めしません。

SUSE ManagerがAWSから必要な情報を収集するには、VHMにEC2インスタンスとアドレスを記述する許可が必要です。これを許可する1つの方法は、このタスクに固有の新しいIAMユーザ(IDおよびアクセス管理)を作成し、次のようにポリシーを作成して、ユーザにアタッチすることです。

```
{
    "Version": "2012-10-17",
    "Statement":[
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAddresses",
                "ec2:DescribeInstances"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

特定のリージョンへのアクセスを制限することで、許可をさらに制限できます。 詳細については、https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ExamplePolicies_EC2.html#iam-example-read-onlyを参照してください。

10.2. VHMとAzure

仮想ホストマネージャ(VHM)を使用して、Microsoft Azureからインスタンスを収集できます。

VHMを使用すると、SUSE Managerは、使用している仮想マシンに関する情報を取得して報告できます。VHMの詳細については、**Client-configuration › Vhm**を参照してください。

10.2.1. 前提条件

作成したVHMは、Azure VMにアクセスするために、正しいパーミッションが割り当てられている必要があります。

サブスクリプション管理者としてAzureアカウントにログインし、Azureユーザアカウントとアプリケーションが正しいグループに属していることを確認してください。 アプリケーションが属しているグループによって、そのロールが決まり、パーミッションが決まります。

10.2.2. Azure VHMの作成

仮想ホストマネージャ(VHM)はSUSE Managerサーバ上で動作します。

virtual-host-gatherer-libcloudパッケージをSUSE Managerサーバにインストール済みであることを確認してください。

プロシージャ: Azure VHMの作成

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. **[作成]**をクリックし、ドロップダウンメニューから［**Azure**］を選択します。
3. ［**Add an Azure Virtual Host Manager**］（Azure仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。
   - ［**ラベル**］フィールドにVHMのカスタム名を入力します。
   - ［**Subscription ID**］（サブスクリプションID）フィールドに、**Azure portal > Services > Subscriptions**ページにあるサブスクリプションIDを入力します。
   - ［**Application ID**］（アプリケーションID）フィールドに、このアプリケーションを登録したときに収集したアプリケーションIDを入力します。
   - ［**Tenant ID**］（テナントID）フィールドに、このアプリケーションを登録したときに収集し

たAzureが提供するテナントIDを入力します。

- ［**Secret Key**］（秘密鍵）フィールドに、Azureインスタンスに関連付けられた秘密鍵を入力します。
- ［**Zone**］（ゾーン）フィールドに、VMが存在するゾーンを入力します。たとえば、西ヨーロッパの場合、**westeurope**と入力します。 これは、サブスクリプションマッチングを動作させるために必要です。

4. **[作成]**をクリックして変更を保存し、VHMを作成します。
5. ［**仮想ホストマネージャ**］ページで、新しいVHMを選択します。
6. ［**プロパティ**］ページで、**[データの更新]**をクリックし、新しいVHMを評価します。

評価されたオブジェクトおよびリソースを表示するには、**システム › システム一覧 › 仮想システム**に移動します。

10.2.3. パーミッションの割り当て

パーミッションが正しく設定されていない場合、**virtual-host-gatherer**を実行すると次のようなエラーが発生する場合があります。

```
General error: (一般エラー:) [AuthorizationFailed] The client 'client_name' with object id
'object_ID' does not have authorization to perform action
'Microsoft.Compute/virtualMachines/read' over scope '/subscriptions/not-very-secret-
subscription-id' or the scope is invalid. ([AuthorizationFailed]オブジェクトID「
object_ID」のクライアント「client_name」には、アクション「Microsoft.Compute/virtualMachine
s/read」をスコープ「/subscriptions/not-very-secret-subscription-
id」を超えて実行する権限がありません。またはスコープが無効です。) If access was recently
granted, please refresh your
credentials. (アクセスが最近付与された場合は、資格情報を更新してください。)
```

正しい資格情報を判断するには、SUSE Managerサーバのプロンプトで次のコマンドを実行します。

```
virtual-host-gatherer -i input_azure.json -o out_azure.json -vvv
```

input_azure.jsonファイルには次の情報が含まれています。

```
[
    {
         "id": "azure_vhm",
         "module": "Azure",
         "subscription_id": "subscription-id",
         "application_id": "application-id",
         "tenant_id": "tenant-id",
         "secret_key": "secret-key",
         "zone": "zone"
     }
 ]
```

10.2.4. Azure UUID

Azureパブリッククラウドで実行されているインスタンスは、このUUIDをSUSE　Managerサーバに報告できます。

```
13f56399-bd52-4150-9748-7190aae1ff21
```

10.3. VHMおよびGoogle Compute Engine

仮想ホストマネージャ(VHM)を使用して、Google　Compute　Engine　(GCE)からインスタンスを収集できます。

VHMを使用すると、SUSE　　Managerは、使用している仮想マシンに関する情報を取得して報告できます。VHMの詳細については、**Client-configuration › Vhm**を参照してください。

10.3.1. 前提条件

作成したVHMは、GCE　VMにアクセスするために、正しいパーミッションが割り当てられている必要があります。

Googleクラウドプラットフォームのアカウントに管理者としてログインし、クラウドのIDおよびアクセス管理(IAM)ツールを使用して、サービスアカウントに適切なロールがあることを確認してください。

10.3.2. GCE VHMの作成

仮想ホストマネージャ(VHM)はSUSE Managerサーバ上で動作します。

VHMを実行するには、SUSE Managerサーバでポート443がオープンになっていて、クライアントにアクセスする必要があります。

virtual-host-gatherer-libcloudパッケージをSUSE　Managerサーバにインストール済みであることを確認してください。

開始する前に、GCEパネルにログインし、証明書ファイルをダウンロードします。　　このファイルをSUSE Managerサーバにローカルに格納し、パスをメモします。

プロシージャ: GCE VHMの作成

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. **[作成]**をクリックし、ドロップダウンメニューから［**Google Compute Engine**］を選択します。
3. ［**Add a Google Compute Engine Virtual Host Manager**］（Google Compute Engineの仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。
    - ［**ラベル**］フィールドにVHMのカスタム名を入力します。
    - ［**Service　Account　Email**］（サービスアカウントメール）フィールドに、サービスアカウントに関連付けられているメールアドレスを入力します。

- ［**Cert Path**］（証明書のパス）フィールドに、GCEパネルからダウンロードしたキーへのSUSE Managerサーバのローカルパスを入力します。
- ［**プロジェクトID**］フィールドに、GCEインスタンスで使用するプロジェクトIDを入力します。
- ［**Zone**］（ゾーン）フィールドに、VMが存在するゾーンを入力します。 これは、サブスクリプションマッチングを動作させるために必要です。

4. **[作成]**をクリックして変更を保存し、VHMを作成します。
5. ［**仮想ホストマネージャ**］ページで、新しいVHMを選択します。
6. ［**プロパティ**］ページで、**[データの更新]**をクリックし、新しいVHMを評価します。

評価されたオブジェクトおよびリソースを表示するには、**システム › システム一覧 › 仮想システム**に移動します。

10.3.3. パーミッションの割り当て

パーミッションが正しく設定されていない場合、**virtual-host-gatherer**を実行すると次のようなエラーが発生する場合があります。

```
ERROR: (エラー:) {'domain': 'global', 'reason': 'forbidden', 'message': "Required
'compute.zones.list' permission for 'projects/project-id'"}
ERROR: (エラー:) Could not connect to the Google Compute Engine Public Cloud using
specified credentials. (指定した資格情報を使用してGoogle Compute
Engineのパブリッククラウドに接続できませんでした。)
```

正しい資格情報を判断するには、SUSE Managerサーバのプロンプトで次のコマンドを実行します。

```
virtual-host-gatherer -i input_google.json -o out_google.json -vvv
```

input_google.jsonファイルには次の情報が含まれています。

```
[
    {
         "id": "google_vhm",
         "module": "GoogleCE",
         "service_account_email": "mail@example.com",
         "cert_path": "secret-key",
         "project_id": "project-id",
         "zone": "zone"
     }
 ]
```

10.3.4. GCE UUID

Googleパブリッククラウドで実行されているインスタンスは、このUUIDをSUSE Managerサーバに報告できます。

```
152986662232938449
```

10.4. VHMとKubernetes

仮想ホストマネージャ(VHM)を使用して、Kubernetesクラスタを管理できます。

VHMを使用すると、SUSE Managerは、クラスタに関する情報を取得して報告できます。 VHMの詳細については、**Client-configuration › Vhm**を参照してください。

KubernetesでSUSE Managerを使用するには、SUSE Managerサーバがコンテナ管理用に設定されていて、必要なすべてのチャンネルがあり、登録されているコンテナビルドホストが利用できる必要があります。

次の要件もあります。

- 1つ以上のKubernetesのクラスタをネットワーク上で使用できる。
- **virtual-host-gatherer-Kubernetes**パッケージがSUSE Managerサーバにインストールされている。
- Kubernetesバージョン1.5.0以上。
- コンテナビルドホストにDockerバージョン1.12以上がある。

10.4.1. Kubernetes VHMの作成

Kubernetesクラスタは、SUSE ManagerにVHMとして登録されています。

Kubernetesクラスタを登録して認可する**kubeconfig**ファイルが必要です。 Kubernetesのコマンドラインツールである**kubectl**を使用して**kubeconfig**ファイルを取得できます。 **kubectl config view --flatten=true**は、VHMに必要に応じて証明書ファイルが埋め込まれた設定を提供します。

プロシージャ: Kubernetes VHMの作成

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. **[作成]**をクリックし、［**Kubernetesクラスタ**］を選択します。
3. ［**Add a Kubernetes Virtual Host Manager**］（Kubernetes仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。
   - ［**ラベル**］フィールドにVHMのカスタム名を入力します。
   - Kubernetesクラスタに必要データが含まれている**kubeconfig**ファイルを選択します。
4. ［**コンテキスト**］フィールドで、クラスタに適切なコンテキストを選択します。 これは**kubeconfig**ファイルで指定されています。
5. **[作成]**をクリックします。

プロシージャ: クラスタのノードを表示する

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. Kubernetesクラスタを選択します。
3. [**Schedule refresh data**]（データの更新をスケジュールする）をクリックしてノードのデータを更新します。

ノードのデータの更新には数分かかる場合があります。 更新された情報を表示するには、ブラウザウィンドウを更新する必要がある場合があります。

接続および認証の問題は**gatherer.log**にログされます。

登録中にはノードのデータは更新されません。 データを表示するにはデータを手動で更新する必要があります。

10.4.2. イメージランタイムデータの取得

SUSE ManagerのWeb UIでKubernetesイメージに関するランタイムデータを表示できます。そのためには、**イメージ › イメージリスト**に移動します。

イメージリストの表には、3つの列があります。

- **リビジョン**:

  SUSE Managerによってビルドされたイメージをリビルドするたびに、または外部でビルドされたイメージをインポートするたびに増加するシーケンス番号。

- **ランタイム**:

  登録されたクラスタの各イメージにおける実行中インスタンスの全般的な状態。

- **インスタンス**:

  SUSE Managerで登録されているすべてのクラスタでこのイメージを実行しているインスタンスの数。数値の横のポップアップアイコンをクリックして数値の内訳を表示できます。

［**ランタイム**］列には、次の状態メッセージのいずれかが表示されます。

- **全てのインスタンスがSUSE Managerと同期できています**:

  実行中のすべてのインスタンスがSUSE Managerによって追跡されているイメージの同じビルドを実行しています。

- **古いインスタンスが見つかりました**:

  インスタンスの一部が古いビルドのインスタンスを実行しています。 イメージを再展開する必要があるかもしれません。

- **情報無し**:

  SUSE Managerに含まれているイメージデータとインスタンスイメージのチェックサムが一致していません。 イメージを再展開する必要があるかもしれません。

プロシージャ: イメージの構築

1. SUSE ManagerのWeb UIで、**イメージ › ストア**に移動します。
2. [**作成**]をクリックしてイメージストアを作成します。
3. **イメージ › プロファイル**に移動します。
4. [**作成**]をクリックしてイメージプロファイルを作成します。 Kubernetesへの展開に適したdockerファイルを使用する必要があります。
5. **イメージ › ビルド**に移動して、新しいプロファイルでイメージをビルドします。
6. イメージを登録済みのKubernetesクラスタのいずれかに展開します。 この操作は**kubectl**ツールを使用して実行できます。

更新データは、**イメージ › イメージリスト**にあるイメージリストに表示されます。

プロシージャ: 以前展開したイメージのインポート

1. SUSE ManagerのWeb UIで、**イメージ › イメージストア**に移動します。
2. インポートするイメージを所有しているレジストリがない場合、追加します。
3. **イメージ › イメージリスト**に移動し、[**インポート**]をクリックします。
4. 各フィールドに入力し、作成したイメージストアを選択し、[**インポート**]をクリックします。

インポートしたデータは、**イメージ › イメージリスト**にあるイメージリストに表示されます。

プロシージャ: 以前展開したイメージの再ビルド

1. SUSE ManagerのWeb UIで、**イメージ › イメージリスト**に移動し、再ビルドするイメージが含まれている行を探し、**[詳細]**をクリックします。
2. ［**ビルド状態**］セクションに移動し、**[再ビルド]**をクリックします。 再ビルドの完了には少し時間がかかります。

再ビルドが正常に完了すると、**イメージ › イメージリスト**のイメージリストでイメージのランタイム状態が更新されます。 インスタンスが前のビルドのインスタンスを実行していることをこれは示しています。

再ビルドできるのは、元々SUSE Managerでビルドされたイメージのみです。 インポートしたイメージは再ビルドできません。

プロシージャ: 追加のランタイムデータの取得

1. SUSE ManagerのWeb UIで、**イメージ › イメージリスト**に移動し、実行中のインスタンスが含まれている行を探し、**[詳細]**をクリックします。
2. ［**概要**］タブに移動します。 ［**イメージの情報**］セクションには、［**ランタイム**］フィールドと［**インスタンス**］フィールドにデータがあります。
3. ［**ランタイム**］タブに移動します。 このセクションには、登録されているすべてのクラスタでこのイメージを実行しているKubernetesポッドに関する情報が含まれています。 このセクションの情報を次に示します。
   - ポッドの名前。
   - ポッドがあるネームスペース。
   - 指定されているポッドのコンテナのランタイム状態。

10.4.3. パーミッションと証明書

SUSE Managerでは**kubeconfig**ファイルにすべての証明書データが埋め込まれている場合、このファイルのみ使用できます。

SUSE ManagerからのAPIコールは次のとおりです。

- **GET /api/v1/pods**
- **GET /api/v1/nodes**

SUSE Managerの最小推奨パーミッションは次のとおりです。

- すべてのノードをリストするClusterRole:

```
resources: ["nodes"]
verbs: ["list"]
```

- すべてのネームスペースのポッドをリストするClusterRole(ロールのバインドはネームスペースを制限してはいけません):

```
resources: ["pods"]
verbs: ["list"]
```

/podsが403の応答を返した場合、SUSE Managerはクラスタ全体を無視します。

For more information on working with RBAC Authorization, see https://kubernetes.io/docs/reference/access-authn-authz/rbac/.

10.5. Nutanixによる仮想化

SUSEはNutanixエコシステムパートナーで、SUSE Managerは"Nutanix AHV integrated"カテゴリでNutanix対応の認証を取得しています。

Nutanixの統合の詳細については、https://www.nutanix.com/partners/technology-alliances/suseを参照してください。

SUSE ManagerではNutanix AHV仮想マシンを使用できます。そのためには、仮想ホストマネージャ(VHM)を設定します。 まず、SUSE Managerサーバで VHMを設定し、使用できるVMホストを評価する必要があります。

10.5.1. VHMの設定

仮想ホストマネージャ(VHM)はSUSE Managerサーバ上で動作します。

virtual-host-gatherer-NutanixパッケージをSUSE Managerサーバにインストール済みであることを確認してください。

VHMを実行するには、SUSE Managerサーバでポート9440がオープンになっていて、Nutanix Prism Element APIにアクセスする必要があります。

プロシージャ: Nutanix VHMの作成

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. **[作成]**をクリックし、［**Nutanix AHV**］を選択します。
3. ［**Add a Nutanix AHV Virtual Host Manager**］（Nutanix AHV仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。

- ［**ラベル**］フィールドにVHMのカスタム名を入力します。
- ［**ホスト名**］フィールドに、完全修飾ドメイン名(FQDN)またはホストIPアドレスを入力します。
- ［**ポート**］フィールドに、使用するPrism Element APIポートを入力します(**9440**など)。
- ［**ユーザ名**］フィールドに、VMホストに関連付けられているユーザ名を入力します。
- ［**パスワード**］フィールドに、VMホストユーザに関連付けられているパスワードを入力します。

4. **[作成]**をクリックして変更を保存し、VHMを作成します。
5. ［**仮想ホストマネージャ**］ページで、新しいVHMを選択します。
6. ［**プロパティ**］ページで、**[データの更新]**をクリックし、新しいVHMを評価します。

評価されたオブジェクトおよびリソースを表示するには、**システム › システム一覧 › 仮想システム**に移動します。

HTTPSを使用してブラウザからNutanix Prism APIサーバに接続すると、**無効な証明書**エラーがログされる場合があります。 このエラーが発生すると、仮想ホストマネージャからのデータの更新は失敗します。 Nutanix APIサーバでは、(自己証明書ではなく)有効なSSL証明書が必要です。

Nutanix SSL証明書にカスタムCA認証局を使用している場合、カスタムCA証明書をSUSE Managerサーバの**/etc/pki/trust/anchors**にコピーします。 証明書を再度信頼します。そのためには、コマンドラインで**update-ca-certificates**コマンドを実行し、spacewalkサービスを再起動します。

VHMが作成されて設定されると、Taskomaticは、データ収集を自動的に実行します。 データ収集を手動で実行する場合、**システム › 仮想ホストマネージャ**に移動し、適切なVHMを選択して**[データの更新]**をクリックします。

APIを使用してVHMに接続して仮想ホストの情報をリクエストできる**virtual-host-gatherer**というツールがSUSE Managerに付属しています。 **virtual-host-gatherer**は、オプションモジュールの概念を維持していて、各モジュールが特定のVHMを有効にします。 このツールは、Taskomaticによって毎晩自動的に呼び出されます。 **virtual-host-gatherer**ツールのログファイルは**/var/log/rhn/gatherer.log**にあります。

10.6. VMwareによる仮想化

SUSE Managerでは、ESXiやvCenterなどのVMware vSphere仮想マシンを使用できます。そのためには、仮想ホストマネージャ(VHM)を設定します。

まず、SUSE ManagerサーバでVHMを設定し、使用できるVMホストを評価する必要があります。 次に、Taskomaticは、VMのAPIを使用してデータ収集を開始できます。

10.6.1. VHMの設定

仮想ホストマネージャ(VHM)はSUSE Managerサーバ上で動作します。

VHMを実行するには、SUSE Managerサーバでポート443がオープンになっていて、VMware APIにアクセス

する必要があります。

VMwareホストは、アクセスロールとパーミッションを使用して、ホストおよびゲストへのアクセスを制御します。 VHMで評価するVMwareのオブジェクトまたはリソースに少なくとも**read-only**パーミッションがあることを確認してください。　任意のオブジェクトまたはリソースを除外する場合、除外対象に**no-access**というマークを付けます。

新しいホストをSUSE　Managerに追加している場合、ユーザおよびオブジェクトに割り当てられているロールおよびパーミッションをSUSE Managerで評価する必要があるかどうかを検討する必要があります。

For more information on users, roles, and permissions, see the VMware vSphere documentation: https://techdocs.broadcom.com/us/en/vmware-cis/vsphere.html

プロシージャ: VMware VHMの作成

1. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
2. **[作成]**をクリックし、［**VMwareベース**］を選択します。
3. ［**Add a VMware-based Virtual Host Manager**］（VMwareベースの仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。
   - ［**ラベル**］フィールドにVHMのカスタム名を入力します。
   - ［**ホスト名**］フィールドに、完全修飾ドメイン名(FQDN)またはホストIPアドレスを入力します。
   - ［**ポート**］フィールドに、使用するESXi APIポートを入力します(**443**など)。
   - ［**ユーザ名**］フィールドに、VMホストに関連付けられているユーザ名を入力します。
   - ［**パスワード**］フィールドに、VMホストユーザに関連付けられているパスワードを入力します。
4. **[作成]**をクリックして変更を保存し、VHMを作成します。
5. ［**仮想ホストマネージャ**］ページで、新しいVHMを選択します。
6. ［**プロパティ**］ページで、**[データの更新]**をクリックし、新しいVHMを評価します。

評価されたオブジェクトおよびリソースを表示するには、**システム › システム一覧 › 仮想システム**に移動します。

HTTPSを使用してブラウザからESXiサーバに接続すると、**無効な証明書**エラーがログされる場合があります。　このエラーが発生すると、仮想ホストサーバからのデータの更新は失敗します。　この問題を修正するには、ESXiサーバから証明書を抽出して**/etc/pki/trust/anchors**にコピーします。　証明書を再度信頼します。そのためには、コマンドラインで**update-ca-certificates**コマンドを実行し、spacewalkサービスを再起動します。

VHMが作成されて設定されると、Taskomaticは、データ収集を自動的に実行します。　データ収集を手動で実行する場合、**システム › 仮想ホストマネージャ**に移動し、適切なVHMを選択して**[データの更新]**をクリックします。

APIを使用してVHMに接続して仮想ホストの情報をリクエストできる**virtual-host-gatherer**というツール

がSUSE　Managerに付属しています。　**virtual-host-gatherer**は、オプションモジュールの概念を維持していて、各モジュールが特定のVHMを有効にします。　このツールは、Taskomaticによって毎晩自動的に呼び出されます。 **virtual-host-gatherer**ツールのログファイルは**/var/log/rhn/gatherer.log**にあります。

10.6.2. VMwareでのSSLエラーのトラブルシューティング

VMwareのインストール環境を設定中にSSLエラーが発生した場合、VMwareからCA証明書をダウンロードし、SUSE Managerで信頼する必要があります。

プロシージャ: VMware CA証明書を信頼する

1. VMwareインストール環境からCA証明書をダウンロードします。 そのためには、vCenterのWeb UIにログインし、**[Download trusted root CA certificates]**（信頼できるルートCA証明書のダウンロード）をクリックします。
2. ダウンロードしたCA証明書ファイルが**.zip**フォーマットの場合、アーカイブを抽出します。 証明書ファイルには拡張子として番号が含まれています。 たとえば、**certificate.0**のようになります。
3. 証明書ファイルをSUSE　Managerサーバにコピーし、**/etc/pki/trust/anchors/**ディレクトリに保存します。
4. コピーした証明書のファイル名のサフィックスを**.crt**または**.pem**に変更します。
5. SUSE Managerサーバのコマンドプロンプトで、CA証明書のレコードを更新します。

```
update-ca-certificates
```

10.7. その他のサードパーティプロバイダを使用した仮想化

Xen、KVMまたはVMware以外のサードパーティ仮想化プロバイダを使用する場合、JSON設定ファイルをSUSE Managerにインポートできます。

同様に、APIへの直接アクセスを提供しないVMwareインストール環境の場合、ファイルベースのVHMが基本的な管理機能を提供します。

このオプションは、**virtual-host-gatherer**ツールを使用して作成されたファイルをインポートするためのものです。 手動で作成したファイル用には設計されていません。

プロシージャ: JSONファイルのエクスポートとインポート

1. VMネットワークで**virtual-host-gatherer**を実行してJSON設定ファイルをエクスポートします。
2. 生成されたファイルをSUSE Managerサーバからアクセスできる場所に保存します。
3. SUSE ManagerのWeb UIで、**システム › 仮想ホストマネージャ**に移動します。
4. **[作成]**をクリックし、［**ファイルベース**］を選択します。
5. ［**Add a file-based Virtual Host Manager**］（ファイルベースの仮想ホストマネージャの追加）セクションで、次のパラメータを使用します。

- ［**ラベル**］フィールドにVHMのカスタム名を入力します。
- ［**Url**］フィールドに、エクスポートするJSON設定フィルへのパスを入力します。

6. **[作成]**をクリックして変更を保存し、VHMを作成します。
7. ［**仮想ホストマネージャ**］ページで、新しいVHMを選択します。
8. ［**プロパティ**］ページで、**[データの更新]**をクリックし、新しいVHMを評価します。

リスト 3. 例: エクスポートするJSON設定ファイル

```
{
    "examplevhost": {
         "10.11.12.13": {
             "cpuArch": "x86_64",
             "cpuDescription": "AMD Opteron(tm) Processor 4386",
             "cpuMhz": 3092.212727,
             "cpuVendor": "amd",
             "hostIdentifier": "'vim.HostSystem:host-182'",
             "name": "11.11.12.13",
             "os": "VMware ESXi",
             "osVersion": "5.5.0",
             "ramMb": 65512,
             "totalCpuCores": 16,
             "totalCpuSockets": 2,
             "totalCpuThreads": 16,
             "type": "vmware",
             "vms": {
                 "vCenter": "564d6d90-459c-2256-8f39-3cb2bd24b7b0"
             }
         },
         "10.11.12.14": {
             "cpuArch": "x86_64",
             "cpuDescription": "AMD Opteron(tm) Processor 4386",
             "cpuMhz": 3092.212639,
             "cpuVendor": "amd",
             "hostIdentifier": "'vim.HostSystem:host-183'",
             "name": "10.11.12.14",
             "os": "VMware ESXi",
             "osVersion": "5.5.0",
             "ramMb": 65512,
             "totalCpuCores": 16,
             "totalCpuSockets": 2,
             "totalCpuThreads": 16,
             "type": "vmware",
             "vms": {
                 "49737e0a-c9e6-4ceb-aef8-6a9452f67cb5":  "4230c60f-3f98-2a65-f7c3-
600b26b79c22",
                 "5a2e4e63-a957-426b-bfa8-4169302e4fdb":  "42307b15-1618-0595-01f2-
427ffcddd88e",
                 "NSX-gateway": "4230d43e-aafe-38ba-5a9e-3cb67c03a16a",
                 "NSX-l3gateway":  "4230b00f-0b21-0e9d-dfde-6c7b06909d5f",
                 "NSX-service": "4230e924-b714-198b-348b-25de01482fd9"
             }
         }
     }
 }
```

詳細については、SUSE Managerサーバの**virtual-host-gatherer**の関数リファレンスを参照してください。

```
man virtual-host-gatherer
```

このパッケージの **README** ファイルには、ハイパーバイザの「種類」などに関する背景情報が記載されています。

```
/usr/share/doc/packages/virtual-host-gatherer/README.md
```

この関数リファレンスおよび **README** ファイルには、設定ファイルのサンプルも含まれています。

Chapter 11. GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections

then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum

below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this

License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled "GNU
   Free Documentation License.
```