



U Y U N I

Uyuni 2022.11

クライアント設定ガイド

2023年01月23日



目次

クライアント設定ガイドの概要	1
1. サポートされているクライアントと機能	2
1.1. サポートされているクライアントシステム	2
1.2. サポートされているツールパッケージ	3
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	4
1.4. サポートされているSLE MicroおよびopenSUSE MicroOSクライアントの機能	6
1.5. サポートされているAlibaba Cloud Linuxの機能	9
1.6. サポートされているAlmaLinuxの機能	11
1.7. サポートされているAmazon Linuxの機能	13
1.8. サポートされているCentOSの機能	15
1.9. サポートされているDebianの機能	17
1.10. サポートされているOracleの機能	19
1.11. サポートされているRed Hat Enterprise Linuxの機能	21
1.12. サポートされているRocky Linuxの機能	24
1.13. サポートされているUbuntuの機能	26
2. 設定の基本	29
2.1. ソフトウェアチャンネル	29
2.1.1. SUSE Package Hubで提供されるパッケージ	30
2.1.2. AppStreamで提供されるパッケージ	30
2.1.3. EPELで提供されるパッケージ	30
2.1.4. SUSE Linux EnterpriseクライアントのUnified Installer更新チャンネル	31
2.1.5. ソフトウェアリポジトリ	31
2.1.6. ソフトウェア製品	32
2.2. ブートストラップリポジトリ	33
2.2.1. ブートストラップリポジトリの作成準備	33
2.2.2. 自動モードのオプション	33
2.2.3. ブートストラップリポジトリの手動生成	34
2.2.4. ブートストラップとカスタムチャンネル	36
2.3. アクティベーションキー	36
2.3.1. 複数のアクティベーションキーの結合	38
2.3.2. 再アクティベーションキー	38
2.3.3. アクティベーションキーのベストプラクティス	39
2.4. GPGキー	41
2.4.1. クライアントでGPGキーを信頼する	41

クライアント設定ガイドの概要

更新: 2023-01-23

クライアントの登録は、Uyuniインストール後の最初の手順であり、Uyuni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。

Uyuni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。

Chapter 1. サポートされているクライアントと機能

Uyuni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。

1.1. サポートされているクライアントシステム

従来のクライアントおよびSaltクライアントでサポートされているオペレーティングシステムを次の表に示します。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります。



クライアントオペレーティングシステムのバージョンおよびSPレベルは、Uyuni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。



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	x86-64, ppc64le, IBM Z, ARM	✓	✓
SUSE Linux Enterprise 12	x86-64, ppc64le, IBM Z, ARM	✓	✓
SUSE Linux Enterprise Server for SAP 15	x86-64, ppc64le	✓	✓
SUSE Linux Enterprise Server for SAP 12	x86-64, ppc64le	✓	✓
SLE Micro	x86-64, ppc64le, aarch64	✗	✓
openSUSE Leap 15	x86-64, aarch64	✓	✓
Alibaba Cloud Linux 2	x86-64, aarch64	✗	✓

Operating System	Architecture	Traditional Clients	Salt Clients
AlmaLinux 9	x86-64, ppc64le, IBM Z, aarch64	✗	✓
AlmaLinux 8	x86-64, aarch64	✗	✓
Amazon Linux 2	x86-64, aarch64	✗	✓
CentOS 7	x86-64, ppc64le, aarch64	✓	✓
Debian 11	x86-64	✗	✓
Debian 10	x86-64	✗	✓
Oracle Linux 9	x86-64, aarch64	✗	✓
Oracle Linux 8	x86-64, aarch64	✗	✓
Oracle Linux 7	x86-64, aarch64	✓	✓
Red Hat Enterprise Linux 9	x86-64	✗	✓
Red Hat Enterprise Linux 8	x86-64	✗	✓
Red Hat Enterprise Linux 7	x86-64	✓	✓
Rocky Linux 9	x86-64, aarch64, ppc64le, s390x	✗	✓
Rocky Linux 8	x86-64, aarch64	✗	✓
Ubuntu 22.04	amd64	✗	✓
Ubuntu 20.04	amd64	✗	✓
Ubuntu 18.04	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。

表 2. Spacewalkのユーティリティ

ツール名	説明	サポートの有無
<code>spacewalk-common-channels</code>	SUSE Customer Center で提供されないチャンネルを追加します	✓
<code>spacewalk-hostname-rename</code>	Uyuniサーバのホスト名を変更します	✓
<code>spacewalk-clone-by-date</code>	特定の日までにチャンネルを複製します	✓
<code>spacewalk-sync-setup</code>	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	✓
<code>spacewalk-manage-channel-lifecycle</code>	チャンネルのライフサイクルを管理します	✓

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亞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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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
登録	✓	✓	Salt
パッケージのインストール	✓	✓	Salt

機能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
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
Uyuni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	✓	✓	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

機能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
システムのロック	Traditional	Traditional	✗
メンテナスウィンドウ	✓	✓	✓
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およびopenSUSE MicroOSクライアントの機能



SLE MicroおよびopenSUSE MicroOS クライアントのサポートは、テスト目的のテクノロジプレビューとして提供されるものであり、この段階ですべての機能が完全に動作するわけではありません。この機能は、Uyuniの後続バージョンですべてサポートされる予定です。運用システムでは、この機能を使用しないでください。



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。 openSUSE MicroOSはSUSEコミュニティでサポートされています。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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およびopenSUSE MicroOS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	openSUSE MicroOS
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
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
製品の移行	?	Salt
基本的な仮想ゲスト管理*	?	Salt
高度な仮想ゲスト管理*	?	Salt
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	✗	Salt

機能	SLE Micro	openSUSE MicroOS
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
システムの配備(PXE/キックスター ト)	?	Salt
システムの再配備(キックスタート)	×	Salt
接続メソッド	Salt: ZeroMQ	Salt: ZeroMQ
Uyuniプロキシでの操作	Salt	Salt
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	Salt	Salt
CVE監査(CVE IDが必要)	Salt	Salt
SCAP監査	?	?
パッケージの確認	?	?
パッケージのロック	Salt	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	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	Salt	Salt
繰り返しアクション	Salt	Salt
AppStream	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。

1.5. サポートされているAlibaba Cloud Linuxの機能

この表には、Alibaba Cloud 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。 Alibaba Cloud Linux はAlibaba Cloudでサポートされています。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. Alibaba Cloud 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	Alibaba Cloud 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*	?

機能	Alibaba Cloud Linux 2
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、Salt-SSH
Uyuni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Alibaba Cloud Linuxで利用できますが、サポートされていません。

1.6. 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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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	なし	なし
基本的な仮想ゲスト管理*	Salt	Salt

機能	AlmaLinux 9	AlmaLinux 8
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
Uyuni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	×	×
パッケージのロック	×	×
システムのロック	×	×
メンテナスウィンドウ	✓	✓
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

*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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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

機能	Amazon Linux 2
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	×
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
Uyuni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。

1.8. 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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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
グループカスタムの状態	Salt
組織カスタムの状態	Salt

機能	CentOS 7
システムセットマネージャ(SSM)	✓
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*	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
Uyuni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	✗
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

機能	CentOS 7
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。

1.9. 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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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 10	Debian 11
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

機能	Debian 10	Debian 11
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
Uyuni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	×	×
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	?	?

機能	Debian 10	Debian 11
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。

1.10. 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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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

機能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
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
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	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
Uyuni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
SCAP監査	✓	Salt	Salt
パッケージの確認	Traditional	✗	✗

機能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
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

*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Red Hat Enterprise Linuxの機能

This table lists the availability of various features on native Red Hat Enterprise Linux clients.



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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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
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

機能	RHEL 7	RHEL 8	RHEL 9
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
Uyuni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	✓	✓

機能	RHEL 7	RHEL 8	RHEL 9
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。

1.12. 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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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

機能	Rocky Linux 8	Rocky Linux 9
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)	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
Uyuni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	×	×
パッケージのロック	?	?
システムのロック	×	×
メンテナスウィンドウ	✓	✓
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	×	×

機能	Rocky Linux 8	Rocky Linux 9
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

*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。

この表のアイコンに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、後日使用できる場合と、使用できない場合があります
- 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. 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

機能	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
クライアント	✓	✓	✓
システムパッケージ	Canonical	Canonical	Canonical
登録	Salt	Salt	Salt
パッケージのインストール	Salt	Salt	Salt

機能	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
パッチの適用	✓	✓	✓
リモートコマンド	Salt	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	なし
基本的な仮想ゲスト管理*	Salt	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	✗	✗	✗
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
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	Salt: ZeroMQ、Salt-SSH
Uyuniプロキシでの操作	Salt	Salt	Salt
動作チェーン	Salt	Salt	Salt
ステージング(パッケージの事前ダウンロード)	Salt	Salt	Salt
重複パッケージの報告	Salt	Salt	Salt
CVE監査	?	?	?
SCAP監査	?	?	?
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	✗	✗	✗
システムのスナップショット	✗	✗	✗
設定ファイルの管理	Salt	Salt	Salt

機能	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
パッケージプロファイル	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	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	Salt	Salt	Salt
Docker buildhost	?	?	?
OSでのDockerイメージの構築	Salt	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	Salt
AppStream	なし	なし	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。

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、Uyuni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、Uyuni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。

- Uyuniのインストールの詳細については、[Installation-and-upgrade > Install-uyuni](#)を参照してください。
- Uyuniのセットアップの詳細については、[Installation-and-upgrade > Uyuni-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。

Uyuni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、関連づ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。

Uyuni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、チャンネルを変更又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。

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リプション先にできる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。

Uyuni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、Uyuni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リプション先にできる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Uyuni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。

Uyuni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。

2.1.5.1. 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、Uyuni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 14. ローカルリポジトリの場所

Client Operating System	Local Repository Directory
SUSE Linux Enterprise Server	<code>/etc/zypp/repos.d</code>
openSUSE	<code>/etc/zypp/repos.d</code>
Red Hat Enterprise Linux	<code>/etc/yum.repos.d/</code>
CentOS	<code>/etc/yum.repos.d/</code>
Ubuntu	<code>/etc/apt/sources.list.d/</code>
Debian	<code>/etc/apt/sources.list.d/</code>

2.1.6. ソフトウェア製品

Uyuni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、Uyuni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** 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ます。 Uyuni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. Uyuni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

エックが付きます。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、Uyuni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から同期の進捗状況を確認する

1. UyuniのWeb UIで、**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。
2. [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. Uyuni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
```

2.2.2.1. 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、Uyuni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. Uyuni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
```

2.2.3.1. 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/oracleでも、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

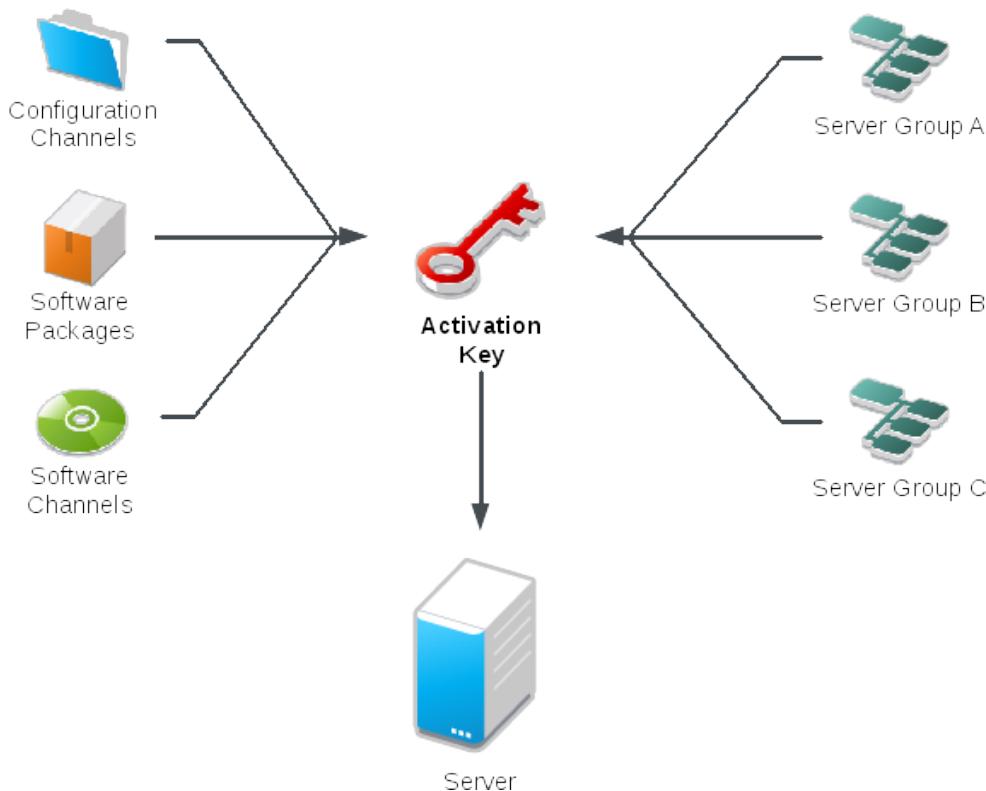
アクティベーションキーは従来のクライアントとSaltクライアントで使用し、クライアントが正しいソフトウェアのエンタイトルメントを持ち、適切なチャンネルに接続して関連グループに加入するようします。それぞれのアクティベーションキーは、キーを作成するときに設定できる組織にひもづけされます。

Uyuni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。



プロセッジヤ: アクティベーションキーの作成

1. Uyuni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. [新規] 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。

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、[reference:admin/setup-wizard.pdf](#)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。

Uyuni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れます。 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Uyuni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、再アクティベーションタブに移動します。[新規]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Uyuni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Uyuni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するには、Uyuni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

プロンバーシップ、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。

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き、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いくと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 clients use now GPG key information entered for a software channel to manage the trusted keys. When a software channel with GPG key information is assigned to a client, the key gets trusted as soon as the channel is refreshed or the first package gets installed from this channel.

ソフトウェアチャンネルのセットであるGPGキーのURLが存在している必要があります。 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。

The GPG keys for the Client Tools Channels of Red Hat based clients are deployed on the client into [/etc/pki/rpm-gpg/](#) and can be referenced with file URLs.