



U Y U N I

Uyuni 2024.07

クライアント設定ガイド

2024年07月 4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SLESおよびopenSUSEの機能	3
1.4. サポートされているSLE Microの機能	6
1.5. サポートされているSL Microクライアント機能	8
1.6. openSUSE Leap Microクライアントの機能	10
1.7. サポートされているAlibaba Cloud Linuxの機能	12
1.8. サポートされているAlmaLinuxの機能	14
1.9. サポートされているAmazon Linuxの機能	16
1.10. サポートされているCentOSの機能	18
1.11. サポートされているDebianの機能	20
1.12. サポートされているopenEulerの機能	22
1.13. サポートされているOracleの機能	24
1.14. サポートされているRaspberry Pi OSの機能	26
1.15. サポートされているRed Hat Enterprise Linuxの機能	28
1.16. サポートされているRocky Linuxの機能	30
1.17. サポートされているUbuntuの機能	32
2. 設定の基本	35
2.1. ソフトウェアチャンネル	35
2.1.1. SUSE Package Hubで提供されるパッケージ	36
2.1.2. AppStreamで提供されるパッケージ	36
2.1.3. EPELで提供されるパッケージ	36
2.1.4. SUSE Linux EnterpriseクライアントのUnified Installer更新チャンネル	37
2.1.5. ソフトウェアリポジトリ	37
2.1.6. ソフトウェア製品	38
2.1.7. AppStreamモジュール	39
2.2. ブートストラップリポジトリ	39
2.2.1. ブートストラップリポジトリの作成準備	39
2.2.2. 自動モードのオプション	39
2.2.3. ブートストラップリポジトリの手動生成	40
2.2.4. ブートストラップとカスタムチャンネル	42
2.3. アクティベーションキー	42
2.3.1. 再アクティベーションキー	44
2.3.2. アクティベーションキーのベストプラクティス	44
2.4. GPGキー	46
2.4.1. クライアントでGPGキーを信頼する	46
3. クライアントの接続メソッド	49
3.1. デフォルトの接続メソッド(Salt)	49
3.1.1. オンボードの詳細	50
3.2. SSH Push接続メソッド	50
3.2.1. 使用可能なパラメータ	52
3.2.2. アクションの実行	53
3.2.3. 今後の機能	53
3.3. SSH Push (トンネルを使用)接続メソッド	54
3.4. Salt Bundle	56
3.4.1. Salt Bundleの概要	56
3.4.2. Salt Bundleを使用してクライアントをMinionとして登録する	56
3.4.3. Salt BundleによるSSH Push	58
3.4.4. pipを使用したPythonパッケージによるSalt Bundleの拡張	58
4. クライアントの登録	60

4.1. 登録メソッド	60
4.1.1. Web UIでクライアントを登録する	60
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	62
4.1.3. コマンドラインでクライアントを登録する	66
4.2. SUSEクライアントの登録	69
4.2.1. SUSE Linux Enterpriseクライアントの登録	69
4.2.2. SLE Microクライアントの登録	73
4.2.3. SL Microクライアントの登録	77
4.3. openSUSEクライアントの登録	80
4.3.1. openSUSE Leapクライアントの登録	81
4.3.2. openSUSE Leap Microクライアントの登録	83
4.4. Alibaba Cloud Linuxクライアントの登録	85
4.4.1. Alibaba Cloud Linuxクライアントの登録	85
4.5. AlmaLinuxクライアントの登録	87
4.5.1. AlmaLinuxクライアントの登録	87
4.6. Amazon Linuxクライアントの登録	90
4.6.1. Amazon Linuxクライアントの登録	90
4.7. CentOSクライアントの登録	93
4.7.1. CentOSクライアントの登録	93
4.8. Debianクライアントの登録	98
4.8.1. Debianクライアントの登録	98
4.9. OpenEulerクライアントの登録	101
4.9.1. openEulerクライアントの登録	101
4.10. Oracleクライアントの登録	105
4.10.1. Oracle Linuxクライアントの登録	105
4.11. Raspberry Pi OSクライアントの登録	108
4.11.1. Raspberry Pi OSクライアントの登録	108
4.12. Red Hatクライアントの登録	112
4.12.1. CDNでRed Hat Enterprise Linuxクライアントを登録する	112
4.12.2. RHUIでRed Hat Enterprise Linuxクライアントを登録する	121
4.13. Rocky Linuxクライアントの登録	126
4.13.1. Rocky Linuxクライアントの登録	126
4.14. Ubuntuクライアントの登録	129
4.14.1. Ubuntu 20.04および22.04 クライアントの登録	129
4.15. プロキシへのクライアントの登録	134
4.15.1. プロキシ間でのクライアントの移動	134
4.15.2. プロキシからサーバへのクライアントの移動	136
4.15.3. Web UIを使用してクライアントをプロキシに登録する	136
4.15.4. コマンドラインでクライアントを登録する	137
4.16. パブリッククラウドでのクライアントの登録	137
4.16.1. 製品の追加とリポジトリの同期	137
4.16.2. オンデマンドイメージの準備	138
4.16.3. クライアントの登録	138
4.16.4. アクティベーションキー	139
4.16.5. Terraformによって作成されたクライアントの自動登録	139
5. クライアントのアップグレード	142
5.1. メジャーバージョンのアップグレード	142
5.1.1. マイグレーションの準備	142
5.1.2. 移行	145
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	145
5.2.1. アップグレードの準備	145
5.2.2. アップグレード	147
5.3. 製品移行	148
5.3.1. 単一システムの移行	148
5.3.2. 製品の大量移行	149
5.4. Uyuniクライアントのアップグレード	153

5.4.1. アップグレードの準備	153
5.4.2. アップグレード	153
6. クライアントの削除	155
6.1. Web UIでクライアントを削除する	155
6.2. コマンドラインでクライアントを削除する(APIコールを使用)	155
6.3. コマンドラインからのクライアントの削除	156
7. クライアントの操作	158
7.1. パッケージ管理	158
7.1.1. パッケージの検証	158
7.1.2. パッケージの比較	158
7.2. パッチ管理	159
7.2.1. パッチの作成	159
7.2.2. クライアントへのパッチの適用	161
7.3. システムのロック	162
7.3.1. クライアントのシステムのロック	162
7.3.2. パッケージのロック	162
7.4. 設定管理	164
7.4.1. 設定チャンネルの作成	165
7.4.2. 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	165
7.4.3. クライアントを設定チャンネルにサブスクライブする	166
7.4.4. 設定ファイルの比較	166
7.4.5. クライアントでのJinjaテンプレート	167
7.5. 電源管理	167
7.5.1. 電源管理とCobbler	168
7.6. カスタムシステム情報	168
7.7. システムセットマネージャ	169
7.7.1. SSMでベースチャンネルを変更する	170
7.8. システムグループ	171
7.8.1. グループの作成	171
7.8.2. グループにクライアントを追加する	172
7.8.3. グループの操作	172
7.9. システムの種類	173
8. オペレーティングシステムのインストール	174
8.1. 登録済みシステムを再インストールする	175
8.2. CD-ROMまたはUSBメモリを使用してインストールする	175
8.2.1. CobblerでISOイメージを構築する	176
8.2.2. KIWIでSUSE ISOイメージを構築する	176
8.2.3. CobblerでRed Hat ISOイメージを構築する	177
8.3. 自動インストールのディストリビューション	177
8.3.1. ISOイメージに基づくディストリビューション	177
8.3.2. RPMパッケージに基づくディストリビューション	178
8.3.3. 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	179
8.4. 自動インストールプロファイル	179
8.4.1. プロファイルを宣言する	180
8.4.2. AutoYaSTプロファイル	181
8.4.3. キックスタートプロファイル	182
8.4.4. テンプレートの構文	183
8.5. 無人プロビジョニング	185
8.6. 独自のGPGキーを使用する	186
8.6.1. PXEブート用の独自のGPGキー	186
8.6.2. CD-ROM内の独自のGPGキー	187
9. 仮想化	188
9.1. 仮想化ホストの管理	188
9.2. 仮想ゲストの作成	188
9.3. XenおよびKVMを使用した仮想化	189

9.3.1. ホストの設定	190
9.3.2. VMゲストの自動インストール	191
9.3.3. VMゲストの管理	195
10. 仮想ホストマネージャ	196
10.1. 仮想ホストマネージャおよびAmazon Web Services	196
10.1.1. Amazon EC2 VHMの作成	196
10.1.2. 仮想ホストマネージャのAWS許可	197
10.2. 仮想ホストマネージャおよびAzure	198
10.2.1. 前提条件	198
10.2.2. Azure VHMの作成	198
10.2.3. パーミッションの割り当て	199
10.2.4. Azure UUID	200
10.3. 仮想ホストマネージャおよびGoogle Compute Engine	200
10.3.1. 前提条件	200
10.3.2. GCE VHMの作成	200
10.3.3. パーミッションの割り当て	201
10.3.4. GCE UUID	202
10.4. Nutanixによる仮想化	202
10.4.1. VHMの設定	202
10.5. VMWareによる仮想化	203
10.5.1. VHMの設定	203
10.5.2. VMWareでのSSLエラーのトラブルシューティング	205
10.6. その他のサードパーティプロバイダを使用した仮想化	205
11. GNU Free Documentation License	208

クライアント設定ガイドの概要

更新: 2024-07-04

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、SUSE Linux Enterpriseまたはその他のLinuxオペレーティングシステム、および広範なハードウェアオプションと互換性があります。

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、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。

1.1. サポートされているクライアントシステム

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と機能

オペレーティングシステム	x86-64	ppc64le	IBM Z	aarch64	arm64 / armhf
SUSE Linux Enterprise 15、12	✓	✓	✓	✓	✗
SUSE Linux Enterprise Server for SAP 15、12	✓	✓	✗	✗	✗
SLE Micro	✓	✗	✗	✓	✗
SL Micro	✓	✗	✗	✓	✗
openSUSE Leap Micro	✓	✗	✗	✓	✗
openSUSE Leap 15	✓	✗	✗	✓	✗
Alibaba Cloud Linux 2	✓	✗	✗	✓	✗
AlmaLinux 9、8	✓	✗	✗	✓	✗
Amazon Linux 2003、2	✓	✗	✗	✓	✗
CentOS 7	✓	✓	✗	✓	✗

オペレーティングシステム	x86-64	ppc64le	IBM Z	aarch64	arm64 / armhf
Debian 12、11	✓	✗	✗	✗	✗
openEuler 22.03	✓	✗	✗	✗	✗
Oracle Linux 9、8、7	✓	✗	✗	✓	✗
Raspberry Pi OS 12	✗	✗	✗	✗	✓
Red Hat Enterprise Linux 9、8、7	✓	✗	✗	✗	✗
Rocky Linux 9、8	✓	✗	✗	✓	✗
Ubuntu 22.04、20.04	✓	✗	✗	✗	✗



(*) 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
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	Uyuni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マスター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	✓

1.3. サポートされているSLESおよびopenSUSE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。

この表のアイコンの意味は次のとおりです。

- ・ ✓: 機能は使用できません
- ・ ✗: 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

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	✓	✓	✓
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	✓	✓	✓
システムカスタムの状態	✓	✓	✓
グループカスタムの状態	✓	✓	✓
組織カスタムの状態	✓	✓	✓
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*	✓	✓	✓
仮想ゲストインストール(AutoYaST)、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	✓	✓	✓
仮想ゲスト管理	✓	✓	✓

機能	SUSE Linux Enterprise 12	SUSE Linux Enterprise 15	openSUSE 15
システムの配備(PXE/AutoYaST)	✓	✓	✓
システムの再配備(AutoYaST)	✓	✓	✓
接続メソッド	✗	✗	✓ 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	✓	✓	✓
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	✓	✓	✓
パッケージのプロファイル	✓ 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	✓ 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	✓ 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	✓	✓	✓
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	✓	✓	✓
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の機能



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(-----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。

- ✓: 機能は使用できません
- ✗: 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

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	✓
オペレーティングシステムパッケージ	✓
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)	✓
リモートコマンド	✓
システムパッケージの状態	✓
システムカスタムの状態	✓
グループカスタムの状態	✓
組織カスタムの状態	✓
システムセットマネージャ(SSM)	✓

機能	SLE Micro
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*	✓
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	✓ ZeroMQ
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	?
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	✓
スナップショットとプロファイル	✓ 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**	✓
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	✓
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、-----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。

- ✓: 機能は使用できません
- ✗: 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

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

機能	SLE Micro
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
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)	?
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. openSUSE Leap Microクライアントの機能



openSUSE Leap MicroはSUSEコミュニティによってサポートされます。

この表のアイコンの意味は次のとおりです。

- ✓: 機能は使用できません
- ✗: 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

表 6. openSUSE Leap 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	openSUSE Leap Micro
クライアント	✓
オペレーティングシステムパッケージ	✓
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)	✓
リモートコマンド	✓
システムパッケージの状態	✓
システムカスタムの状態	✓
グループカスタムの状態	✓
組織カスタムの状態	✓
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*	✓

機能	openSUSE Leap Micro
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	✓ ZeroMQ
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	?
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	✓
スナップショットとプロファイル	✓ 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	✓
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	✓
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。

1.7. 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。

この表のアイコンの意味は次のとおりです。

- ✓: 機能は使用できません
- ✗: 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

表 7. 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	✓
オペレーティングシステムパッケージ	✓
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)	✓
リモートコマンド	✓
システムパッケージの状態	✓
システムカスタムの状態	✓
グループカスタムの状態	✓
組織カスタムの状態	✓
システムセットマネージャ(SSM)	✓
製品移行	N/A
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	Alibaba Cloud 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	✓ 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	✗
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	✓
スナップショットとプロファイル	✓ 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	✓
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	✓
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。

1.8. 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。

- ✓: 機能は使用できません
- ✗: 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

表 8. 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	✓ (plain AlmaLinux)	✓ (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	✓	✓
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	✓	✓
システムカスタムの状態	✓	✓
グループカスタムの状態	✓	✓
組織カスタムの状態	✓	✓
システムセットマネージャ(SSM)	✓	✓
製品移行	なし	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	✓	✓

機能	AlmaLinux 9	AlmaLinux 8
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	✓ ZeroMQ、Salt-SSH	✓ 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	✓	✓
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	✓	✓
スナップショットとプロファイル	✓ 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	✓ 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	✓	✓
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	✓	✓
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。

1.9. 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。

- ✓: 機能は使用できません
- ✗: 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

表 9. 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	Amazon Linux 2023
クライアント	✓	✓
オペレーティングシステムパッケージ	✓	✓
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)	✓	✓
リモートコマンド	✓	✓
システムパッケージの状態	✓	✓
システムカスタムの状態	✓	✓
グループカスタムの状態	✓	✓
組織カスタムの状態	✓	✓
システムセットマネージャ(SSM)	✓	✓
製品移行	なし	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	Amazon Linux 2023
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	✓ ZeroMQ、Salt-SSH	✓ 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	✗	✗
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	✓	✓
スナップショットとプロファイル	✓ 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	✓ 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	✓	✓
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	✓	✓
AppStreams	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。

- ✓: 機能は使用できません
- ✗: 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

表 10. 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)	✓ (エラータにはサードパーティサービスが必要)
リモートコマンド	✓
システムパッケージの状態	✓
システムカスタムの状態	✓
グループカスタムの状態	✓
組織カスタムの状態	✓
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*	✓
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	CentOS 7
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	✓ 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	✗
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	✓
スナップショットとプロファイル	✓ 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	✓
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	✓
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。

1.11. 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。

- ✓: 機能は使用できません
- ✗: 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

表 11. 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	✓	✓
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	✓	✓
システムカスタムの状態	✓	✓
グループカスタムの状態	✓	✓
組織カスタムの状態	✓	✓
システムセットマネージャ(SSM)	✓	✓
製品移行	なし	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	✓	✓

機能	Debian 11	Debian 12
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	✓ ZeroMQ、Salt-SSH	✓ 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	?	?
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	✓	✓
パッケージのプロファイル	✓ 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	✓ 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	✓	✓
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	✓	✓
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。

1.12. サポートされているopenEulerの機能

この表には、openEuler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。openEulerはopenEuler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

表 12. openEuler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	openEuler 22.03
クライアント	✓ (plain openEuler)
システムパッケージ	openEuler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	✓
リモートコマンド	✓
システムパッケージの状態	✓
システムカスタムの状態	✓
グループカスタムの状態	✓
組織カスタムの状態	✓
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*	✓
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	openEuler 22.03
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	✓ 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	✓
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	✓
スナップショットとプロファイル	✓ 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	✓
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	✓
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

より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。

- ✓: 機能は使用できません
- ✗: 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

表 13. 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	✓	✓	✓
オペレーティングシステムパッケージ	✓	✓	✓
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)	✓	✓	✓
リモートコマンド	✓	✓	✓
システムパッケージの状態	✓	✓	✓
システムカスタムの状態	✓	✓	✓
グループカスタムの状態	✓	✓	✓
組織カスタムの状態	✓	✓	✓
システムセットマネージャ(SSM)	✓	✓	✓
製品移行	なし	なし	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	Oracle Linux 7	Oracle Linux 8	Oracle 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	✗	✓	✓
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	✓ ZeroMQ、Salt-SSH	✓ ZeroMQ、Salt-SSH	✓ 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	✗	✗	✗
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	✓	✓	✓
スナップショットとプロファイル	✓ 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	✓ 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	✓ 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	✓	✓	✓
Docker buildhost	✗	✗	✗
OSでのDockerイメージの構築	✗	✗	✗
Kiwi buildhost	✗	✗	✗

機能	Oracle Linux 7	Oracle Linux 8	Oracle Linux 9
OSでのKiwiイメージの構築	✗	✗	✗
繰り返しアクション	✓	✓	✓
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Raspberry Pi OSの機能

この表には、Raspberry Pi OS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。Raspberry Pi OSはRaspberry Pi OS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。

- ✓: 機能は使用できません
- ✗: 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

表 14. Raspberry Pi OS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	Raspberry Pi OS 12
クライアント	✓
システムパッケージ	Raspberry Pi OS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	?
リモートコマンド	✓
システムパッケージの状態	✓

機能	Raspberry Pi OS 12
システムカスタムの状態	✓
グループカスタムの状態	✓
組織カスタムの状態	✓
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*	✓
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	✓ 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	?
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	✓
パッケージのプロファイル	✓ 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	✓
Docker buildhost	?
OSでのDockerイメージの構築	✓
Kiwi buildhost	✗

機能	Raspberry Pi OS 12
OSでのKiwiイメージの構築	✗
繰り返しアクション	✓
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。

1.15. 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。



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。

- ✓: 機能は使用できません
- ✗: 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

表 15. 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	✓	✓	✓
システムパッケージ	Red Hat	Red Hat	Red Hat
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	✓	✓	✓

機能	RHEL 7	RHEL 8	RHEL 9
システムパッケージの状態	✓	✓	✓
システムカスタムの状態	✓	✓	✓
グループカスタムの状態	✓	✓	✓
組織カスタムの状態	✓	✓	✓
システムセットマネージャ(SSM)	✓	✓	✓
製品移行	なし	なし	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	✓ ZeroMQ、Salt-SSH	✓ ZeroMQ、Salt-SSH	✓ 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	✓	✓	✓
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نس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	✓	✓	✓

機能	RHEL 7	RHEL 8	RHEL 9
スナップショットとプロファイル	✓ 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	✓ 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	✓ 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	✓	✓	✓
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	✓	✓	✓
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。

1.16. 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。

- ✓: 機能は使用できません
- ✗: 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

表 16. 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	✓ (plain Rocky Linux)	✓ (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	✓	✓
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	✓	✓
システムカスタムの状態	✓	✓
グループカスタムの状態	✓	✓
組織カスタムの状態	✓	✓
システムセットマネージャ(SSM)	✓	✓
製品移行	なし	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	✓ ZeroMQ、Salt-SSH	✓ 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	✓	✓
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	✗	✗

機能	Rocky Linux 8	Rocky Linux 9
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	✓	✓
スナップショットとプロファイル	✓ 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	✓ 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	✓	✓
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	✓	✓
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。

1.17. 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。

- ✓: 機能は使用できません

- ・ **✗:** 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

表 17. 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 20.04	Ubuntu 22.04
クライアント	✓	✓
システムパッケージ	Canonical	Canonical
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	✓	✓
システムカスタムの状態	✓	✓
グループカスタムの状態	✓	✓
組織カスタムの状態	✓	✓
システムセットマネージャ(SSM)	✓	✓
製品移行	なし	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	✓ ZeroMQ、Salt-SSH	✓ 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	?	?
SCAP監査	?	?
パッケージの確認	✗	✗
パッケージのロック	✓	✓

機能	Ubuntu 20.04	Ubuntu 22.04
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	✓	✓
パッケージのプロファイル	✓ 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	✓ 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	✓	✓
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	✓	✓
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。

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。

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

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。

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 18. 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のディレクトリ
SUSE Linux Enterprise Server	/etc/zypp/repos.d
openSUSE	/etc/zypp/repos.d
Red Hat Enterprise Linuxと類似の派生物	/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

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。  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ェックが付きます。 また、include recommendedトグルがオフになっている場合、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.  をクリックし、製品の同期が完了するまで待機します。

2.1.7. AppStreamモジュール

AppStreamチャンネルをサブスクリーブしているクライアントを管理する場合、Uyuniにより、現在有効になっている一連のAppStreamモジュールに応じて、クライアントで利用可能なパッケージがフィルタされます。

プロシージャ: 有効なAppStreamモジュールの管理

1. UyuniのWeb UIで、**システム** › **ソフトウェア** › **AppStreams**に移動します。
2. モジュラーストリームの表から、クライアントで有効化されているストリームを選択します。
3. **[[変更]の適用]** をクリックします。
4. 次のページで、変更内容のリストを確認し、いつ変更するかを選択します。
5. **[[確認]** をクリックして、変更の適用をスケジュールします。

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クライアントを登録するために必要な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. **[同期]** タブに移動し、**[同期]** をクリックし、**[同期]** 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

- 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
```

- 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
```

- 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

- 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
```

- 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

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。

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を使用し、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。

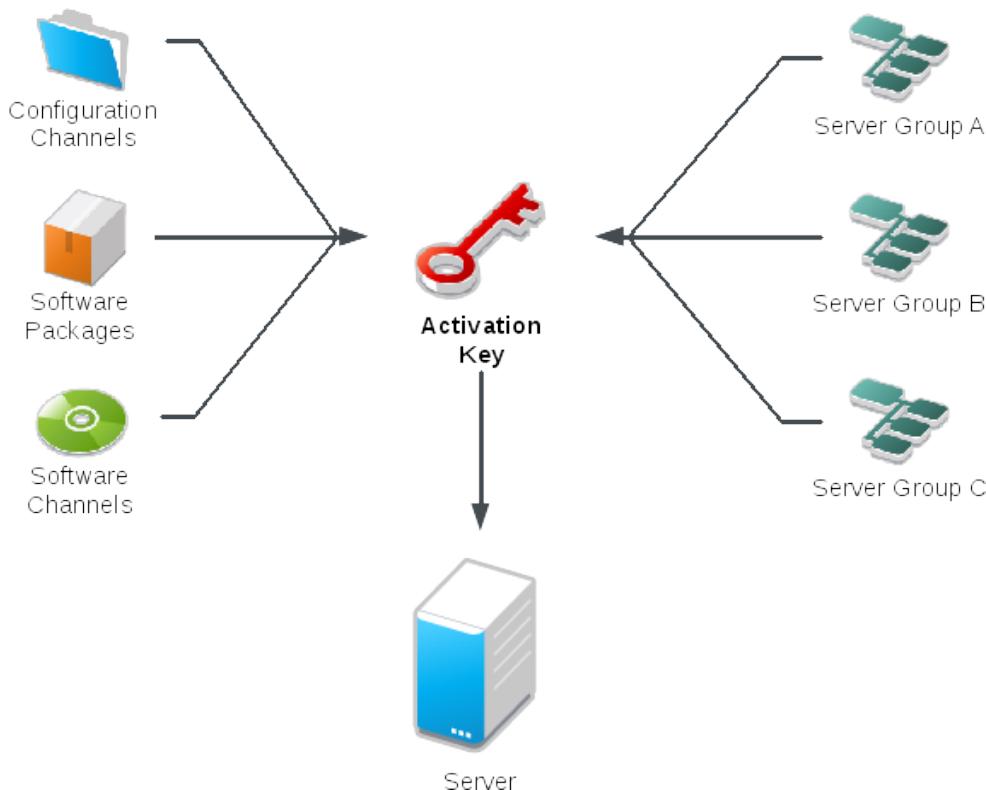
Uyuniでは、アクティベーションキーは、ラベルの付いた一連の設定です。アクティベーションキーに関連付けられている設定は、すべて適用できます。そのためには、キーのラベルをパラメータにしてブートストラップスクリプトに追加します。アクティベーションキーラベルをブートストラップスクリプトと組み合わせて使用することをお勧めします。ブートストラップスクリプトが実行されると、そのラベルに関連付けられているすべての設定が、スクリプトを実行しているシステムに適用されます。

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。



プロシージャ: 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. [キーの一の作成] ボタンをクリックします。
3. [-----] ページの [-----] 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. [-----] 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 SP5では「SLES15-SP5」と入力します。



SUSE製品の [-----] 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. [Base Channels] 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、[-----] チェックボックスにチェックを付けます。

8. [-----] は [-----] のままにすることをお勧めします。
9. [-----] 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. [-----] 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。



[-----] 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. 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、[-----] 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詳細タブに移動します。「[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。

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"
```

venv-salt-minionまたは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。



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。

2.3.2. 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-XXXXXXXXXXXXXXの親チャンネルを使用しないでください。この設定では、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ごとに、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。
- [---] セクションのパッケージがインストールされます。
- [---] 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
```

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。

クライアントは、ソフトウェアチャンネルに入力されたGPGキーの情報を使用して信頼済みのキーを管理するようになります。GPGキーの情報が含まれるソフトウェアチャンネルをクライアントに割り当てるとき、チャンネルを更新したとき、またはこのチャンネルから最初のパッケージをインストールしたときに、その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。

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、システムによってインポートされて信頼されます。



Debianベースのシステムはメタデータのみに署名するため、单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。

2.4.1.1. ユーザ定義のGPGキー

ユーザは、クライアントに配備するカスタムの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
```

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--URL] に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

2.4.1.2. 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

- Uyuni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。

- 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. クライアントの接続メソッド

Uyuniサーバがクライアントと通信する接続メソッドは3つあります。これらはSaltプロトコルに基づきます。

使用するSalt接続メソッドは、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。

デフォルト(Salt)

デフォルトの選択肢であり、特別な必要がない限り、この接続メソッドを推奨します。詳細については、see **Client-configuration** › **Contact-methods-default**を参照してください。

SSH Push

ネットワークの制限のためにクライアントがサーバとの接続を確立できない場合にのみ有用です。Salt Bundleでのみサポートされます。この接続メソッドには重大な制限があります。詳細については、**Client-configuration** › **Contact-methods-saltssh**を参照してください。

SSH Push (トンネルを使用)

SSH Pushと同じですが、セキュリティ保護された通信トンネルを使用します。詳細については、**Client-configuration** › **Contact-methods-saltssh-tunnel**を参照してください。

Salt通信ソフトウェアのオプションの実装:

Salt Bundle

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。したがって、Salt接続メソッドは、クライアントにインストールされているソフトウェアから独立しています。Salt Bundleはデフォルトで使用されます。これは、SSH PushまたはSSH Push(トンネルを使用)の接続メソッドでサポートされている唯一の実装です。詳細については、**Client-configuration** › **Contact-methods-saltbundle**を参照してください。

Salt Minion

クライアントシステムにインストールされるSaltソフトウェアです。



いわゆる従来の接続プロトコルは、Uyuni 5.0以降ではサポートされなくなりました。Uyuni 4から5にアップグレードする前に、従来のプロキシを含む既存の従来のクライアントをSaltに移行するか、Saltプロキシで置き換える必要があります。

従来のUyuni 4クライアントからSaltクライアントへの移行の詳細については、<https://documentation.suse.com/suma/4.3/en/suse-manager/client-configuration/contact-methods-migrate-traditional.html>を参照してください。

3.1. デフォルトの接続メソッド(Salt)

特別な必要がない限り、Saltプロトコルを使用するデフォルトの接続メソッドを推奨します。Saltの全般的な詳細については、**Specialized-guides** › **Salt**を参照してください。

ソフトウェア更新は一般的にサーバからクライアントにプッシュされます。接続はクライアントから開始されます。つまり、クライアント上ではなくサーバ上でポートを開く必要があります。SaltクライアントはSalt minionとしても知られます。Uyuni Serverはすべてのクライアントにデーモンをインストールします。

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、SSH Pushを接続メソッドとして設定できます。この接続メソッドでは、クライアントをDMZと呼ばれるファイアウォール保護ゾーンに配置できます。SSH Pushの詳細については、[Client-configuration › Contact-methods-saltssh](#)を参照してください。

3.1.1. オンボードの詳細

Saltには、Minionのキーを保持する専用のデータベースがあります。このデータベースとUyuniのデータベースとの同期を保つ必要があります。キーがSaltで受け入れられるとすぐに、Uyuniでオンボーディングプロセスが開始されます。

このオンボーディングプロセスでは、minion_idとmachine_idを検索することで、既存のシステムがUyuniのデータベース内で検索されます。この結果に応じて、次のようなシナリオが考えられます。

- 何も見つからない場合、新しいシステムが作成されます。
- minion_idまたはmachine_idを持つエントリが見つかった場合、そのシステムが、新しいシステムに一致するように移行されます。
- 両方のエントリに一致する項目があり、それらが同じシステム上にない場合、オンボーディングプロセスは中止されてエラーが発生します。
- この場合、管理者が少なくとも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。

3.2. SSH Push接続メソッド

SSH Push (ssh-push)は、SaltクライアントがUyuni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があります。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(Uyuniサーバなど)はDMZ内にはありません。

SSH Pushは、デーモンエージェントをクライアントにインストールできない場合にも使用されます。



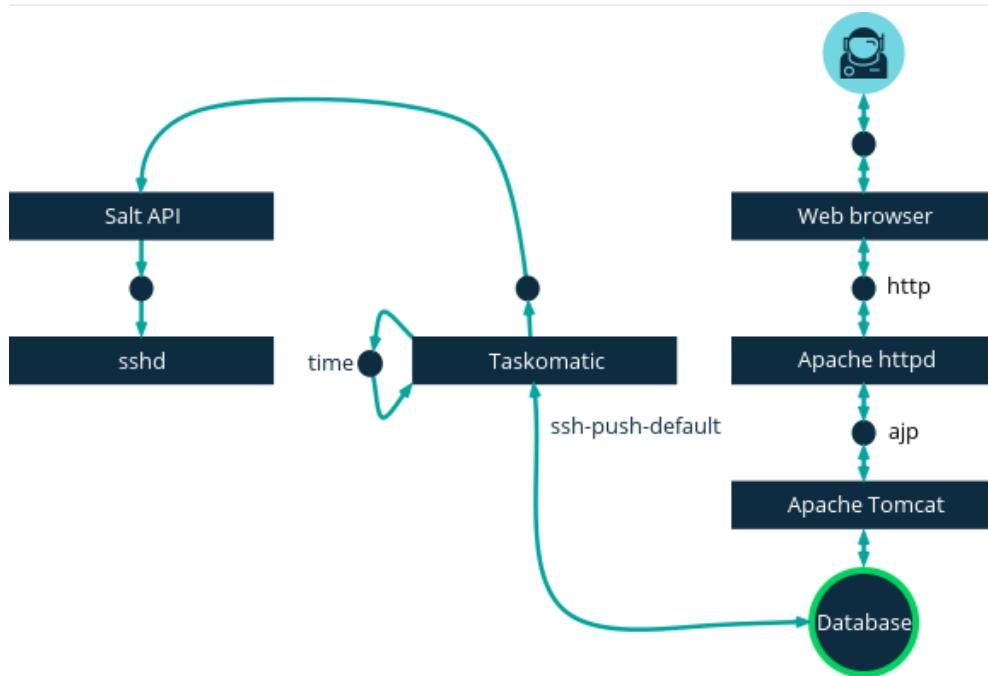
SSH Pushメソッドには重大な制限があります。SSH Pushは適切にスケールせず、標準のSaltメソッド(default)よりも多くのサーバリソースとネットワーク帯域幅を消費します。Push SSHメソッドは、大規模なセットアップ(1000クライアント以上)では一切サポートされません。

サーバは、SSH Push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。



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 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 Push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、Uyuniクライアントで実行されるプロセスを表します。



SSH Pushを使用するには、SSHデーモンがクライアントで動作していて、Uyuniサーバで動作しているsalt-apiデーモンによって接続できる必要があります。また、必須のPythonバージョンをSalt Bundleとともにクライアントシステムに配備する必要があります。

次の登録手順を開始する前に、SSH Push接続メソッドが設定された状態でアクティベーションキーを定義します。このメソッドでは、HTTPSでのサーバとの直接接続が存在する必要があります。

Web UIまたはAPIを使用して、これらのクライアントをUyuniサーバに登録する必要があります。次のプロシージャまたは例を参照してください。

プロシージャ: SSH Pushを使用したクライアントの登録

1. Uyuni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 Push接続メソッドが設定された状態で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。

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'})
```

3.2.1. 使用可能なパラメータ

SSH Push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。

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ための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を設定することもできます。

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Uyuni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. 各クライアントシステムで、sudoersファイルを編集します。

```
sudo visudo
```

4. 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/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。

5. Uyuni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>
```

この設定パラメータを変更した後、salt-secrets-config.service、tomcat.service、taskomatic.serviceなどのサービスを再起動する必要があります。必要なサービスすべてを対象とするには、Uyuniサーバをrootとして再起動することをお勧めします。

```
mgradm restart
```

3.2.2. アクションの実行

SSH Push機能は、taskomaticを使用し、salt-sshを使用してスケジュールされた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。SSH Push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。

3.2.3. 今後の機能

SSH Push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。

3.2.3.1. 詳細情報

- Salt SSH全般については、**Specialized-guides** › **Salt**および<https://docs.saltstack.com/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。

3.3. SSH Push (トンネルを使用)接続メソッド

SSH Push (トンネルを使用)は、クライアントでUyuni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(Uyuniサーバなど)はDMZ内にはありません。

このSSH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Uyuni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。



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 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。



トンネルを使用して、暗号化されたトンネルを介したサーバへのアクセスを提供します。SSH Pushクライアント(トンネルを使用)に割り当てられたリポジトリは、このトンネルを介してのみ提供されます。したがって、リポジトリを利用できるのはトンネルの動作中だけであるため、クライアントシステムから直接、パッケージマネージャツールを使用することはできません。つまり、セッションがサーバによって開始された場合にのみ、アクセスが可能です。クライアント上でのすべてのパッケージ管理操作はサーバ側からのみ実行できます。

SSHでのトンネル接続では、HTTPS経由のトンネル用のポート番号が必要です。デフォルトで使用されるポート番号は`1233`です。この番号を上書きするには、1024よりも大きいカスタムポート番号を/etc/rhn/rhn.confに追加します。

```
ssh_push_port_https = high_port
```

この設定パラメータを変更した後に、salt-secrets-config.service、tomcat.service、taskomatic.serviceなどのサービスを再起動する必要があります。必要なすべてのサービスを対象とするには、rootとしてspacewalk-serviceを再起動することをお勧めします。

```
spacewalk-service restart
```

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. 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。
2. 各クライアントシステムで、sudoersファイルを編集します。

```
sudo visudo
```

3. 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/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。

4. Uyuni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>
```

この設定パラメータを変更した後に、salt-secrets-config.service、tomcat.service、taskomatic.serviceなどのサービスを再起動する必要があります。必要なすべてのサービスを対象とするには、rootとしてspacewalk-serviceを再起動することをお勧めします。

```
spacewalk-service restart
```

Web UIまたはAPIを使用して、これらのクライアントをUyuniサーバに登録する必要があります。

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、再アクティベーションキーを使用して再登録する必要があります。

- 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。
- ブートストラップの詳細については、[client-configuration:activation-keys.pdf](#)を参照してください。

例: SSH Push (トンネルを使用)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-tunnel`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-tunnel'})
```

3.4. Salt Bundle

3.4.1. 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。

Uyuni Salt Master以外のSalt Masterに接続されている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。

3.4.2. 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インターフ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。

3.4.2.1. Salt Minionを使用したSalt Bundleの使用

Salt Bundleは、Uyuni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、Uyuni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の設定](#)を参照してください。



- Uyuni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。

3.4.2.2. 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

- 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、`mgr_avoid_venv_salt_minion` pillarをTrueに設定します。

3.4.3. Salt BundleによるSSH Push

Salt Bundleは、クライアントに対してSSH Push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。SSH Push (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インターパ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SSH Pushセッションを処理するために使用されるため、SSH Push (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、推奨もサポートもされておらず、開発目的でのみ存在しています。`/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。SSH Push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。
- SSH Push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SSH Push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。

3.4.4. 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。



Chapter 4. クライアントの登録

クライアントをUyuni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Uyuni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。



UyuniサーバのベースOSをUyuni自体に登録しないでください。UyuniサーバのベースOSは個別に管理するか、別個のUyuniサーバを使用して管理する必要があります。

Uyuniサーバコンテナを管理するには、`mgradm`ツールを使用します。

4.1. 登録メソッド

クライアントをUyuniサーバに登録する方法は複数あります。

- クライアントの数が少ない場合、Uyuni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、または多数のクライアントを登録する必要がある場合は、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。
- 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Uyuni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。



Uyuniサーバをこのサーバ自体に登録しないでください。Uyuniサーバは個別に管理するか、別のUyuni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

Web UIを使用してクライアントをブートストラップする場合、[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ます。Salt SSHは、Salt Bundleおよびそれに含まれてい

るPythonインターペリターを使用します。したがって、その他のPythonインターペ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。



Uyuniサーバをこのサーバ自体に登録しないでください。Uyuniサーバは個別に管理するか、別のUyuni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. UyuniのWeb UIで、**システム**、**ブートストラップ**に移動します。
2. [IPアドレス] 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ポート番号] 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. [ユーザ名] 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、[SSH密鑑] フィールドで、[SSH密鑑] 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密鑑をアップロードします。SSH密鑑でパスフレーズが必要な場合、[SSH密鑑パスフレーズ] 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、[パスワード] 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厳格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. [次へ] 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Uyuniサーバから削除されます。



Uyuni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。

4.1.1.1. ローカルで割り当てられたリポジトリの取り扱い

Uyuni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、Uyuni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](#)のマニュア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。

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一部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キーを使用してそのベースチャンネルに割り当てるようになります。

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、Uyuni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、および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の詳細については、[Client-configuration > Bootstrap-repository](#)を参照してください。



GPGキーおよびUyuniクライアントツール

Uyuniクライアントツールで使用されるGPGキーは、デフォルトでは信頼されません。ブートストラップスクリプトを作成するとき、`ORG_GPG_KEY`パラメータを使用して公開鍵の指紋を含むファイルへのパスを追加します。



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。

4.1.2.1. `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、Uyuni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。Uyuni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_KEY>`および`<EDITED_NAME>`は使用する値に置き換えます。

```
mgr-bootstrap --activation-keys=<ACTIVATION_KEY> --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(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。

4.1.2.2. Web UIでのブートストラップスクリプトの作成

Uyuni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. Uyuni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. 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。
3. **[新]**をクリックしてスクリプトを作成します。
4. 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Uyuni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、see **Administration > Ssl-certs**を参照してください。

4.1.2.3. 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。 Uyuni 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、 [...] ボックスで、 [...] 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. Uyuni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=

```

- 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、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。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。

4.1.2.4. 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登録する

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

- Uyuniサーバにrootとしてログインし、コマンドプロンプトで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/
```

- 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
```

- 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。Uyuni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。クライアントがリストされていない場合は、UyuniのWeb UIで**Salt** > **キー**に移動して、クライアントキーが受け入れられているかどうかを確認します。



- Uyuni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. コマンドラインでクライアントを登録する

4.1.3.1. 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Uyuni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Uyuni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。

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。

4.1.3.2. 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、Uyuni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. Uyuni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>'
```

4.1.3.3. クライアントの設定

クライアント(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、Uyuni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. Uyuni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、<https://docs.saltstack.com/en/latest/ref/configuration/minion.html>を参照してください。

4.2. SUSEクライアントの登録

SUSE Linux EnterpriseクライアントをUyuniサーバに登録する方法は複数あります。

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Uyuni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。



Uyuniサーバをこのサーバ自身に登録しないでください。Uyuniサーバは個別に管理するか、別のUyuni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. SUSE Linux Enterpriseクライアントの登録

このセクションでは、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。

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。

4.2.1.1. 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Uyuni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 19. SLE製品 - WebUI

OSバージョン	製品名
SUSE Linux Enterprise Server 15 SP6	SUSE Linux Enterprise Server 15 SP6 x86_64
SUSE Linux Enterprise Server 15 SP5	SUSE Linux Enterprise Server 15 SP5 x86_64

OSバージョン	製品名
SUSE Linux Enterprise Server 15 SP4	SUSE Linux Enterprise Server 15 SP4 x86_64
SUSE Linux Enterprise Server 15 SP3	SUSE Linux Enterprise Server 15 SP3 x86_64
SUSE Linux Enterprise Server 15 SP2	SUSE Linux Enterprise Server 15 SP2 x86_64
SUSE Linux Enterprise Server 15 SP1	SUSE Linux Enterprise Server 15 SP1 x86_64
SUSE Linux Enterprise Server 12 SP5	SUSE Linux Enterprise Server 12 SP5 x86_64

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製品 - CLI

OSバージョン	ベースチャンネル
SUSE Linux Enterprise Server 15 SP6	sle-product-sles15-sp6-pool-x86_64
SUSE Linux Enterprise Server 15 SP5	sle-product-sles15-sp5-pool-x86_64
SUSE Linux Enterprise Server 15 SP4	sle-product-sles15-sp4-pool-x86_64
SUSE Linux Enterprise Server 15 SP3	sle-product-sles15-sp3-pool-x86_64
SUSE Linux Enterprise Server 15 SP2	sle-product-sles15-sp2-pool-x86_64
SUSE Linux Enterprise Server 15 SP1	sle-product-sles15-sp1-pool-x86_64
SUSE Linux Enterprise Server 12 SP5	sle-product-sles15-sp5-pool-x86_64

古い製品のチャンネル名を見つけるには、Uyuni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]
```

手順: 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. Uyuni 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。

クライアントツールを追加するには、コマンドプロンプトからこれらのチャンネルを追加します。

表 21. SUSE Linux Enterprise チャンネル - CLI

OSバージョン	クライアントチャンネル
SUSE Linux Enterprise Server 15 SP6	sles15-sp6-uyuni-client
SUSE Linux Enterprise Server 15 SP5	sles15-sp5-uyuni-client
SUSE Linux Enterprise Server 15 SP4	sles15-sp4-uyuni-client
SUSE Linux Enterprise Server 15 SP3	sles15-sp3-uyuni-client
SUSE Linux Enterprise Server 15 SP2	sles15-sp2-uyuni-client
SUSE Linux Enterprise Server 15 SP1	sles15-sp1-uyuni-client
SUSE Linux Enterprise Server 12 SP5	sles12-sp5-uyuni-client

手順: 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. Uyuni 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。

4.2.1.2. 同期ステータスの確認

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. [→] タブに移動し、[→] をクリックし、[→] 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。



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。

4.2.1.3. 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。



SUSE Linux Enterprise Server 15とSUSE Linux Enterprise Server 12クライアントの両方で同じGPGキーを使用します。正しいキーはsle12-gpg-pubkey-39db7c82.keyと呼ばれます。

4.2.1.4. 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。

4.2.2. SLE Microクライアントの登録

このセクションでは、x86-64、arm64、およびIBM Z (s390x) アーキテクチャで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、最新の不变の開発者向け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。

4.2.2.1. ソフトウェアチャンネルの追加

SLE MicroクライアントをUyuni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 22. 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
SLE Micro 5.3 s390x	SUSE Linux Enterprise Micro 5.3 s390x

OSバージョン	製品名
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. 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. 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 5.5 arm64	sle-micro-5.5-pool-arm64	sle-micro-5.5-updates-arm64
SLE Micro 5.5 IBM Z (s390x)	sle-micro-5.5-pool-s390x	sle-micro-5.5-updates-s390x
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
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

OSバージョン	ベースチャンネル	更新チャンネル
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

手順: 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. Uyuni 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。

クライアントツールを追加するには、コマンドプロンプトからこれらのチャンネルを追加します。

表 24. SLE Microチャンネル - CLI

OSバージョン	クライアントチャンネル
SLE Micro 5.4	sle-micro-5.4-uyuni-client
SLE Micro 5.3	suse-micro-5.3-uyuni-client
SLE Micro 5.2	suse-microos-5.2-uyuni-client
SLE Micro 5.1	sle-microos-5.1-uyuni-client

手順: 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. Uyuni 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。

4.2.2.2. 同期ステータスの確認

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. [-----] タブに移動し、[----] をクリックし、[-----] 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。

4.2.2.3. 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。

ブートストラッププロセスに続いて、SLE Microクライアントで自動ブートが無効化されます。この変更は、SaltがブートストラップのSaltの状態を適用した結果を中継できる前に発生していた、断続的な自動再起動が原因で実装されました。

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
```

4.2.2.4. 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。Uyuni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。

Uyuni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、Uyuni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、Uyuniでスケジュールしたりできます。

4.2.2.4.1. 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、Uyuni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。

4.2.3. 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、最新の不变の開発者向け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。

4.2.3.1. ソフトウェアチャンネルの追加

SL MicroクライアントをUyuni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 25. SL Micro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. 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 26. SL Micro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

手順: 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. Uyuni 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>
```

- 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>
```

- 続行前に、同期が完了していることを確認してください。

4.2.3.2. 同期ステータスの確認

プロシージャ: Web UIから同期の進捗状況を確認する

- 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。
- [...] タブに移動し、 [...] をクリックし、 [...] 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

- 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
```

- 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。

4.2.3.3. 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、デフォルトの再起動マネージャのメンテナنس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。

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
```

4.2.3.4. 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。 Uyuni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。

Uyuni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、Uyuni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、Uyuniでスケジュールしたりできます。

4.2.3.4.1. 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、Uyuni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。

4.3. openSUSEクライアントの登録

openSUSEおよびopenSUSE Leap MicroのクライアントをUyuni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Uyuni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。



Uyuniサーバをこのサーバ自体に登録しないでください。Uyuniサーバは個別に管理するか、別のUyuni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クライアントの登録について説明します。Uyuniは、Saltを使用するopenSUSE Leap 15クライアントをサポートします。

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。

4.3.1.1. ソフトウェアチャンネルの追加

openSUSEクライアントをUyuni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 27. openSUSEチャンネル - CLI

OSバージョン	openSUSE Leap 15.6	openSUSE Leap 15.5	openSUSE Leap 15.4
ベースチャンネル	opensuse_leap15_6	opensuse_leap15_5	opensuse_leap15_4
クライアントチャンネル	opensuse_leap15_6-uyuni-client	opensuse_leap15_5-uyuni-client	opensuse_leap15_4-uyuni-client
更新チャンネル	opensuse_leap15_6-updates	opensuse_leap15_5-updates	opensuse_leap15_4-updates
非OSSチャンネル	opensuse_leap15_6-non-oss	opensuse_leap15_5-non-oss	opensuse_leap15_4-non-oss
非OSS更新チャンネル	opensuse_leap15_6-non-oss-updates	opensuse_leap15_5-non-oss-updates	opensuse_leap15_4-non-oss-updates
バックポート更新チャンネル	opensuse_leap15_6-backports-updates	opensuse_leap15_5-backports-updates	opensuse_leap15_4-backports-updates
SLE更新チャンネル	opensuse_leap15_6-sle-updates	opensuse_leap15_5-sle-updates	opensuse_leap15_4-sle-updates

手順: 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

- Uyuni 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>
```

- 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>
```

- 続行前に、同期が完了していることを確認してください。

4.3.1.2. 同期ステータスの確認

プロシージャ: Web UIから同期の進捗状況を確認する

- 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。
- [同期] タブに移動し、[同期] をクリックし、[同期] 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

- 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
```

- 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。

4.3.1.3. 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。

4.3.1.4. 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。

4.3.2. openSUSE Leap Microクライアントの登録

このセクションでは、x86-64およびaarch64アーキテクチャでopenSUSE Leap Micro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。

openSUSE Leap 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、最新の不变の開発者向けOSプラットフォームと統合します。

openSUSE Leap 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。

表 28. openSUSEチャンネル - CLI

OSバージョン	openSUSE Leap Micro 5.5	openSUSE Leap Micro 5.4	openSUSE Leap Micro 5.3
ベースチャンネル	opensuse_micro5_5	opensuse_micro5_4	opensuse_micro5_3
クライアントチャンネル	opensuse_micro5_5-uyuni-client	opensuse_micro5_4-uyuni-client	opensuse_micro5_3-uyuni-client
SLE更新チャンネル	opensuse_micro5_5-sle-updates	opensuse_micro5_4-sle-updates	opensuse_micro5_3-sle-updates

手順: 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

- Uyuni 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>
```

- 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>
```

- 続行前に、同期が完了していることを確認してください。

4.3.2.1. 同期ステータスの確認

プロシージャ: Web UIから同期の進捗状況を確認する

- 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。
- [...] タブに移動し、 [...] をクリックし、 [...] 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

- 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
```

- 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 Leap Micro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。

4.3.2.2. クライアントの登録

openSUSE Leap Microクライアントは、登録後に再起動が必要です。



ブートストラッププロセス後に、openSUSE Leap Micro上で自動ブートが無効化されます。この変更は、SaltがブートストラップのSaltの状態を適用した結果を中継できるようになる前に発生する可能性があった断続的な自動再起動が原因で実装されました。

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に及ぶ場合があります。openSUSE Leap Microの登録を高速化するには、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. Alibaba Cloud Linuxクライアントの登録

Alibaba Cloud LinuxクライアントをUyuni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Uyuni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。

4.4.1. Alibaba Cloud Linuxクライアントの登録

このセクションでは、Alibaba Cloud Linux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。



一部のAlibaba Cloud Linux 2インスタンスでは、正常に登録するために2回の試行が必要になります。

4.4.1.1. ソフトウェアチャンネルの追加

Alibaba Cloud LinuxクライアントをUyuni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チャンネルは次のとおりです。

表 29. Alibaba Cloud Linuxチャンネル - CLI

OSバージョン	コアチャンネル	更新チャンネル	クライアントチャンネル
Alibaba Cloud Linux 2	alibaba-2	alibaba-2-updates	alibaba-2-uyuni-client

手順: 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. Uyuni 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。



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。

4.4.1.2. 同期ステータスの確認

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. [→→→→] タブに移動し、[→→] をクリックし、[→→→→] 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。

4.4.1.3. アクティベーションキーの作成

Alibaba Cloud 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。

4.4.1.4. 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。

一部のAlibaba Cloud Linux 2インスタンスは、最初の試行で登録に失敗します。これはAlibaba Cloud Linux 2イメージの既知のバグのためです。

「python-urlgrabber3」パッケージは、Python pipパッケージとRPMパッケージの両方で提供されており、最初の登録試行時に競合が発生する可能性があります。

インスタンスが影響を受けるいずれかのイメージバージョンに基づいている場合、クライアントは2回目の登録試行で正しく登録されます。

4.5. AlmaLinuxクライアントの登録

AlmaLinuxクライアントをUyuni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Uyuni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。

4.5.1. 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クライアントの登録について説

明します。



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。

4.5.1.1. ソフトウェアチャンネルの追加



AlmaLinuxクライアントのUyuniへの登録は、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Uyuniに登録する必要があります。

AlmaLinuxクライアントをUyuni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。

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. AlmaLinux 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	AppStreamチャンネル
AlmaLinux 9	almalinux9	almalinux9-uyuni-client	almalinux9-appstream
AlmaLinux 8	almalinux8	almalinux8-uyuni-client	almalinux8-appstream

手順: 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. Uyuni 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。このとき、正しいアーキテクチャを指定してください:

```
spacewalk-common-channels \
-a <architecture> \
<base_channel_name> \
<child_channel_name_1> \
<child_channel_name_2> \
... <child_channel_name_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>-<architecture>
```

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。



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。



AlmaLinux 9およびAlmaLinux 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、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Uyuni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、Uyuni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。

4.5.1.2. 同期ステータスの確認

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. [同期] タブに移動し、[同期] をクリックし、[同期] 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。

4.5.1.3. 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。

4.5.1.4. 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はソフトウェアチャンネルを使用できないため、チャンネルをクライアントに割り当てるかどうかの判断は、キーを信頼するかどうかの判断に依存します。

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

4.5.1.5. 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。

4.5.1.6. 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。

4.6. Amazon Linuxクライアントの登録

Amazon LinuxクライアントをUyuni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Uyuni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。

4.6.1. 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クライアントの登録について説明します。



AWSで作成するとき、Amazon Linux 2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。Amazon Linux 2インスタンスを作成する場合は、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。

Amazon Linux 2023はこの影響を受けません。

4.6.1.1. ソフトウェアチャンネルの追加

Amazon LinuxクライアントをUyuni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。

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 31. Amazon Linux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
Amazon Linux 2023	amazonlinux2023	amazonlinux2023-uyuni-client
Amazon Linux 2	amazonlinux2-core	amazonlinux2-uyuni-client



また、Amazon Linux 2インスタンスでは、Amazon LinuxインスタンスでDockerを使用する場合は、必ず「amazonlinux2-extra-docker」チャンネルを追加して同期してください。



Amazon Linux 2023では、Amazon Linuxインスタンスでカーネルライブパッチを使用する予定の場合は、必ず「amazonlinux2023-kernel-livepatch」チャンネルも追加して同期してください。

手順: 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. Uyuni 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。



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。

4.6.1.2. 同期ステータスの確認

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. [同期] タブに移動し、[同期] をクリックし、[同期] 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。

4.6.1.3. 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。

4.6.1.4. 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。

4.6.1.5. 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。

4.7. CentOSクライアントの登録

CentOSクライアントをUyuni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Uyuni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。

4.7.1. 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クライアントの登録について説明します。



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Uyuni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。



CentOSクライアントのUyuniへの登録は、`http://<UyuniServer>/police`で`allow`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CentOSクライアントをUyuniに登録する必要があります。

4.7.1.1. ソフトウェアチャンネルの追加

CentOSクライアントをUyuni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。

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. CentOS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	更新/Appstreamチャンネル
CentOS 7	centos7	centos7-uyuni-client	centos7-updates

手順: 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. Uyuni 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。このとき、正しいアーキテクチャを指定してください:

```
spacewalk-common-channels \
-a <architecture> \
<base_channel_name> \
<child_channel_name_1> \
<child_channel_name_2> \
... <child_channel_name_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>-<architecture>
```

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。



`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。

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Uyuni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、Uyuni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。

4.7.1.2. 同期ステータスの確認

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. [→] タブに移動し、[→] をクリックし、[→] 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。

4.7.1.3. 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。

4.7.1.4. 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。

4.7.1.5. 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

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。

4.7.1.6. エラータの管理



このセクションはコンテナ化されていないUyuniにのみ適用され、CentOS 7がサポート終了になった後に削除されます。

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。



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カ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Uyuni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. Uyuni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。

4.8. Debianクライアントの登録

DebianクライアントをUyuni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Uyuni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。

4.8.1. 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クライアントの登録について説明します。

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。

4.8.1.1. 登録の準備

DebianクライアントをUyuni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、Uyuni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Uyuniサーバと日時が同期されている必要があります。

4.8.1.2. ソフトウェアチャンネルの追加

DebianクライアントをUyuni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チャンネルは次のとおりです。

表 33. Debian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	更新チャンネル	セキュリティチャンネル
Debian 12	debian-12-pool-amd64-uyuni	debian-12-amd64-uyuni-client	debian-12-amd64-main-updates-uyuni	debian-12-amd64-main-security-uyuni
Debian 11	debian-11-pool-amd64-uyuni	debian-11-amd64-uyuni-client	debian-11-amd64-main-updates-uyuni	debian-11-amd64-main-security-uyuni

手順: 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. Uyuni 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。

4.8.1.3. 同期ステータスの確認

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. [同期] タブに移動し、[同期] をクリックし、[同期] 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。



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。

4.8.1.4. 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。



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。

UyuniサーバにすでにインポートされているGPGキーを確認するには、次のコマンドを実行します。

```
mgrctl exec -- gpg --homedir /var/lib/spacewalk/gpgdir --list-keys
```

新しいGPGキーをインポートするには、次のコマンドを実行します。

```
mgradm gpg add <filename>.gpg
```

4.8.1.5. 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。

4.8.1.6. 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. OpenEulerクライアントの登録

openEulerクライアントをUyuni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Uyuni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。

4.9.1. openEulerクライアントの登録

このセクションでは、openEuler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。



AWSで作成するとき、openEuler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。

4.9.1.1. ソフトウェアチャンネルの追加

openEulerクライアントをUyuni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。

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. openEulerチャンネル - CLI

OSバージョン	コアチャンネル	クライアントチャンネル	更新チャンネル	EPOLチャンネル	Everythingチャンネル
openEuler 22.03	openeuler2203	openeuler2203-uyuni-client	openeuler2203-update	openeuler2203-epol	openeuler2203-everything

手順: 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

- Uyuni 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>
```

- 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>
```

- 続行前に、同期が完了していることを確認してください。



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。

4.9.1.2. 同期ステータスの確認

プロシージャ: Web UIから同期の進捗状況を確認する

- 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。
- [...] タブに移動し、 [...] をクリックし、 [...] 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

- 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。

4.9.1.3. アクティベーションキーの作成

openEuler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。

4.9.1.4. クライアントでGPGキーを信頼する

4.9.1.5. 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。

クライアントは、ソフトウェアチャンネルに入力されたGPGキーの情報を使用して信頼済みのキーを管理するようになります。GPGキーの情報が含まれるソフトウェアチャンネルをクライアントに割り当てると、チャンネルを更新したとき、またはこのチャンネルから最初のパッケージをインストールしたときに、その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。

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、システムによってインポートされて信頼されます。



Debianベースのシステムはメタデータのみに署名するため、单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。

4.9.1.5.1. ユーザ定義のGPGキー

ユーザは、クライアントに配備するカスタムの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
```

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URL] に値file:///etc/pki/rpm-gpg/my_first_gpg.keyを追加します。

4.9.1.5.2. 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

- Uyuni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。
- 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。必要な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。

4.9.1.6. クライアントの登録

openEulerクライアントは、その他すべてのクライアントと同じ方法で登録されます。詳細については、**Client-configuration > Registration-overview**を参照してください。

4.10. Oracleクライアントの登録

Oracle LinuxクライアントをUyuni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Uyuni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。
- 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。

4.10.1. 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クライアントの登録について説明します。



Unbreakable Linux Network (ULN)リポジトリとUyuni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。

4.10.1.1. ソフトウェアチャンネルの追加

Oracle LinuxクライアントをUyuni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。

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 35. Oracle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	更新チャンネル
Oracle Linux 9	oraclelinux9	oraclelinux9-uyuni-client	oraclelinux9-appstream
Oracle Linux 8	oraclelinux8	oraclelinux8-uyuni-client	oraclelinux8-appstream
Oracle Linux 7	oraclelinux7	oraclelinux7-uyuni-client	-

手順: 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

- Uyuni 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>
```

- 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>
```

- 続行前に、同期が完了していることを確認してください。



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。



Oracle Linux 9およびOracle Linux 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。

4.10.1.2. 同期ステータスの確認

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. [同期] タブに移動し、[同期] をクリックし、[同期] 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。

4.10.1.3. 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。

4.10.1.4. 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。

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
```

4.10.1.5. 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. Raspberry Pi OSクライアントの登録

Raspberry Pi OSクライアントをUyuni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Uyuni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。

4.11.1. Raspberry Pi OSクライアントの登録

このセクションでは、Raspberry Pi OS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。

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Raspberry Pi OSクライアントで使用できます。

4.11.1.1. 登録の準備

Raspberry Pi OSクライアントをUyuni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、Uyuni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Uyuniサーバと日時が同期されている必要があります。

4.11.1.2. ソフトウェアチャンネルの追加

Raspberry Pi OSクライアントをUyuni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、arm64とarmhf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。

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. Raspberry Pi OSチャンネル - CLI

チャンネルの説明	arm64	armhf
ベースチャンネル	raspberrypios-12-pool-arm64-uyuni	raspberrypios-12-pool-armhf-uyuni
クライアントチャンネル	raspberrypios-12-arm64-uyuni-client	raspberrypios-12-armhf-uyuni-client
更新チャンネル	raspberrypios-12-arm64-main-updates-uyuni	-
コントリビューションチャンネル	raspberrypios-12-arm64-contrib-uyuni	raspberrypios-12-armhf-contrib-uyuni
非フリーチャンネル	raspberrypios-12-arm64-non-free-uyuni	raspberrypios-12-armhf-non-free-uyuni
非フリーファームウェアチャンネル	raspberrypios-12-arm64-non-free-firmware-uyuni	-
Raspberryチャンネル	raspberrypios-12-arm64-raspberry-uyuni	raspberrypios-12-armhf-raspberry-uyuni
コントリビューション更新	raspberrypios-12-arm64-contrib-updates-uyuni	-
非フリー更新	raspberrypios-12-arm64-non-free-updates-uyuni	-
非フリーファームウェア更新	raspberrypios-12-arm64-non-free-firmware-updates-uyuni	-
セキュリティメインチャンネル	raspberrypios-12-arm64-main-security-uyuni	-
セキュリティコントリビューションチャンネル	raspberrypios-12-arm64-contrib-security-uyuni	-
セキュリティ非フリーチャンネル	raspberrypios-12-arm64-non-free-security-uyuni	-
セキュリティ非フリーファームウェアチャンネル	raspberrypios-12-arm64-non-free-firmware-security-uyuni	-
RPIチャンネル	-	raspberrypios-12-armhf-rpi-uyuni

手順: 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. Uyuni 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。



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。



Raspberry Pi OSクライアントをブートストラップする前に、新しいチャンネルはすべて完全に同期されている必要があります。

4.11.1.3. 同期ステータスの確認

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. [→→→→] タブに移動し、[→→→] をクリックし、[→→→→] 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。



Raspberry Pi OS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。

4.11.1.4. アクティベーションキーの作成

Raspberry Pi OS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。

4.11.1.5. 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。



Raspberry Pi OS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Raspberry Pi OS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。

Raspberry Pi OS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。

UyuniサーバにすでにインポートされているGPGキーを確認するには、次のコマンドを実行します。

```
mgrctl exec -- gpg --homedir /var/lib/spacewalk/gpgdir --list-keys
```

新しいGPGキーをインポートするには、次のコマンドを実行します。

```
mgradm gpg add <filename>.gpg
```

4.11.1.6. rootアクセス

Raspberry Pi OS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。

4.11.1.7. 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. 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Uyuni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Uyuni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。

4.12.1. CDNでRed Hat Enterprise Linuxクライアントを登録する

このセクションでは、Red Hat Enterprise Linuxオペレーティングシステムを実行しているRed Hatコンテンツデリバリネットワーク(CDN)の登録について説明します。

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Uyuni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。

4.12.1.1. 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 Hat CDN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Uyuni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. Uyuni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Uyuni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. Uyuni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メータを設定します。
 - [----] フィールドにEntitlement-Cert-dateと入力します。
 - [-----] フィールドで、SSLを選択します。

- [-----] 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. [キ一の作成] 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一覧を設定します。

- [----] フィールドにEntitlement-key-dateと入力します。

- [----] フィールドで、SSLを選択します。

- [-----] 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. [キ一の作成] をクリックします。

6. [格納されているキーまたは証明書Red Hat証明書用] を次のパラメータを設定します。

- [----] フィールドにredhat-uepと入力します。

- [----] フィールドで、SSLを選択します。

- [-----] 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. [キ一の作成] をクリックします。

4.12.1.2. ソフトウェアチャンネルの追加

Red HatクライアントをUyuni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チャンネルは次のとおりです。

表 37. Red Hat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
Red Hat 9	rhel9-pool-uyuni	-	rhel9-uyuni-client
Red Hat 8	rhel8-pool-uyuni	-	rhel8-uyuni-client
Red Hat 7	rhel7-pool-uyuni	-	rhel7-uyuni-client

手順: 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

- Uyuni 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。このとき、正しいアーキテクチャを指定してください:

```
spacewalk-common-channels \
-a <architecture> \
<base_channel_name> \
<child_channel_name_1> \
<child_channel_name_2> \
... <child_channel_name_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>-<architecture>
```

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。



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。

4.12.1.3. 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Uyuni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 38. 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

オプション	設定
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. Uyuni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] [ボ] [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] [ボ] [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 39. Red Hatカスタムチャンネル

OSバージョン	ベースチャンネル
Red Hat 9	rhel9-pool-uyuni
Red Hat 8	rhel8-pool-uyuni
Red Hat 7	rhel7-pool-uyuni

プロシージャ: カスタムチャンネルの作成

1. Uyuni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. 「[--]」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 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。

モジュー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. Uyuni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。

4.12.1.4. 同期ステータスの確認

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. [同期] タブに移動し、[同期] をクリックし、[同期] 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。



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. UyuniサーバのWeb UIで、**設定**、**チャンネル**に移動します。
2. [状態チャレンジの作成] をクリックします。
 - [同期] フィールドにsubscription-manager: disable yum pluginsと入力します。
 - [同期] フィールドにsubscription-manager-disable-yum-pluginsと入力します。
 - [同期] フィールドにsubscription-manager: disable yum pluginsと入力します。
 - [SLS同期] フィールドは空白のままにします。
3. [設定チャレンジの作成] 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
 - [同期] フィールドに/etc/yum/pluginconf.d/subscription-manager.confと入力します。
 - [同期] 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' -----]」をクリックします。
 - 「-----」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-----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. Uyuni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。
 - 「-----」フィールドにrhel-systemsと入力します。
 - 「-----」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. 「-----」タブをクリックします。
5. 「-----」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Uyuni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. Uyuni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. [-----] タブ、[----] 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。

4.12.1.5. 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。



Red Hatカスタムチャンネルの場合、[GPG-----] フィールドを確認する場合は、[GPG-----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。

4.12.1.6. 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. RHUIでRed Hat Enterprise Linuxクライアントを登録する

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クライアントを登録する方法について説明します。

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。



Uyuni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。

4.12.2.1. 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Uyuni 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、このタスクが自動化されました。詳細については、[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Uyuni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の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。

4.12.2.2. 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

- Uyuni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。
- ページセクション **[PAYG→→→→→→→→→→]** から始めます。
- [→→→]** フィールドに、説明を追加します。
- ページセクション **[SSH→→→→→→→→→→]** に移動します。
- [→→→→→]** フィールドに、Uyuni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。
- [SSH→→→→→]** 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。
- [→→→→→]** 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。
- [→→→→→→→→]** フィールドに、パスワードを入力します。
- [SSH→→→→→]** フィールドに、インスタンスキーを入力します。
- [SSH→→→→→→→→→→→→→→→→]** 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要塞接続データの追加](#)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要塞接続データの追加

- ページセクション **[→→→SSH→→→→→→→→]** に移動します。
- [→→→→→]** フィールドに、要塞のホスト名を入力します。
- [SSH→→→→→]** フィールドに、要塞のポート番号を入力します。
- [→→→→→]** フィールドに、要塞のユーザ名を入力します。
- [→→→→→→→→]** フィールドに、要塞のパスワードを入力します。

6. [SSH] フィールドに、要塞キーを入力します。
7. [SSH] 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 [...] ページに戻ります。 更新された接続ステータスは、 [...] 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、 [...] > [...] > 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、 [...] 列に「 [...] 」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 [[guimenu] > [...] > [...] > PAYG] に表示され、 [...] 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、 [...] > [...] > [...] 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。

4.12.2.3. ソフトウェアチャンネルの追加

Red HatクライアントをUyuni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チャンネルは次のとおりです。

表 40. Red Hat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
Red Hat 7	rhel7-pool-uyuni	-	rhel7-uyuni-client
Red Hat 8	rhel8-pool-uyuni	-	rhel8-uyuni-client

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
Red Hat 9	rhel9-pool-uyuni	-	rhel9-uyuni-client

手順: 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. Uyuni 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。このとき、正しいアーキテクチャを指定してください:

```
spacewalk-common-channels \
-a <architecture> \
<base_channel_name> \
<child_channel_name_1> \
<child_channel_name_2> \
... <child_channel_name_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>-<architecture>
```

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。



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。

4.12.2.4. 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Uyuni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 41. Red Hatカスタムチャンネル

OSバージョン	ベースチャンネル
Red Hat 7	rhel7-pool-uyuni
Red Hat 8	rhel8-pool-uyuni
Red Hat 9	rhel9-pool-uyuni

プロシージャ: カスタムチャンネルの作成

1. Uyuni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. [-->] 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 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. Uyuni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。

4.12.2.5. 同期ステータスの確認

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. 「[同期]」タブに移動し、「[同期]」をクリックし、「[同期]」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。



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。

4.12.2.6. 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。

4.12.2.7. 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. Rocky Linuxクライアントの登録

Rocky LinuxクライアントをUyuni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Uyuni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。

4.13.1. 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クライアントの登録について説明します。



Rocky LinuxクライアントのUyuniへの登録は、~~-----~~ポリシーで~~-----~~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Uyuniに登録するために、SELinuxを無効にする必要はありません。

4.13.1.1. ソフトウェアチャンネルの追加

Rocky LinuxクライアントをUyuni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。

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. Rocky Linux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	AppStreamチャンネル
Rocky Linux 9	rockylinux9	rockylinux9-uyuni-client	rockylinux9-appstream
Rocky Linux 8	rockylinux8	rockylinux8-uyuni-client	rockylinux8-appstream

手順: 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. Uyuni 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。このとき、正しいアーキテクチャを指定してください:

```
spacewalk-common-channels \
-a <architecture> \
<base_channel_name> \
<child_channel_name_1> \
<child_channel_name_2> \
... <child_channel_name_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>-<architecture>
```

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。



`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。



Rocky Linux 8およびRocky Linux 9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。



上流のチャンネルとUyuni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、Uyuni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。

Rocky Linux 8で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。

4.13.1.2. 同期ステータスの確認

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. [同期] タブに移動し、[同期] をクリックし、[同期] 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。

4.13.1.3. 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。

4.13.1.4. 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。

4.13.1.5. 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。

4.13.1.6. 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。

4.14. Ubuntuクライアントの登録

UbuntuクライアントをUyuni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Uyuni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。

4.14.1. Ubuntu 20.04および22.04 クライアントの登録

このセクションでは、Ubuntu 20.04 LTSおよび22.04 LTS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。

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。



Canonicalは、Uyuniを保証またはサポートしていません。

4.14.1.1. ソフトウェアチャンネルの追加

UbuntuクライアントをUyuni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チャンネルは次のとおりです。

表 43. Ubuntuチャンネル - CLI

OSバージョン	ベースチャンネル	メインチャンネル	更新チャンネル	セキュリティチャンネル	クライアントチャンネル
Ubuntu 22.04	ubuntu-2204-pool-amd64-uyuni	ubuntu-2204-amd64-main-uyuni	ubuntu-2204-amd64-main-updates-uyuni	ubuntu-2204-amd64-main-security-uyuni	ubuntu-2204-amd64-uyuni-client
Ubuntu 20.04	ubuntu-2004-pool-amd64-uyuni	ubuntu-2004-amd64-main-uyuni	ubuntu-2004-amd64-main-updates-uyuni	ubuntu-2004-amd64-main-security-uyuni	ubuntu-2004-amd64-uyuni-client

バージョン20.04では次のUniverseチャンネルも必要です。

表 44. Ubuntu 20.04 Universeチャンネル - CLI

Ubuntu 20.04	
Universeチャンネル	ubuntu-2004-amd64-universe-uyuni
Universe更新チャンネル	ubuntu-2004-amd64-universe-updates-uyuni
Universeセキュリティ更新チャンネル	ubuntu-2004-amd64-universe-security-uyuni
Universeバックポートチャンネル	ubuntu-2004-amd64-universe-backports-uyuni

手順: 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. Uyuni 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。



Ubuntuクライアントをブートストラップする前に、新しいチャンネルはすべて完全に同期されている必要があります。

4.14.1.2. Ubuntu ESMパッケージのミラーリング

Canonicalは、<https://ubuntu.com/pro>[Ubuntu Pro]のユーザと顧客向けに<https://ubuntu.com/security/esm>[Expanded Security Maintenance] (**ESM**)パッケージを提供しています。これらのパッケージは、複数のオペレーティングシステムコンポーネントと選択したアプリケーションを対象として、より長期のメンテナンス(10~12年)を提供します。

これらのリポジトリは、Ubuntu Proに登録されているシステムから必要なGPGキーと個人用ペアラートークンを抽出すれば、Uyuni内でも同期できます。

4.14.1.2.1. GPGキーとペアラートークンの抽出

UbuntuホストをUbuntu Proに登録します。個人用登録トークンは、<https://ubuntu.com/pro/dashboard>[Ubuntu Proダッシュボード]で確認できます。このためには<https://login.ubuntu.com/>[Ubuntu Oneアカウント]が必要です。

```
sudo apt-get install ubuntu-adantage-tools
sudo pro attach <personal_token>
```

登録後、ファイル`/etc/apt/auth.conf.d/90ubuntu-adantage`でペアラートークンを確認できます。

```
machine esm.ubuntu.com/apps/ubuntu/ login bearer password <token> #
ubuntu-pro-client
machine esm.ubuntu.com/infra/ubuntu/ login bearer password <token> #
ubuntu-pro-client
```



リポジトリごとに1つの専用のペアラートークンが使用されます。

Uyuni内で次のリポジトリを設定します。

4.14.1.2.2. Ubuntu ESMリポジトリの設定

リポジトリを作成するには、次のURLを使用します。

表 45. Ubuntu ESMリポジトリ

URL	説明
<a href="https://bearer:<token>@esm.ubuntu.com/infra/ubuntu/dists/<release>-infra-updates/main/binary-<arch>/">https://bearer:<token>@esm.ubuntu.com/infra/ubuntu/dists/<release>-infra-updates/main/binary-<arch>/	オペレーティングシステムの機能更新
<a href="https://bearer:<token>@esm.ubuntu.com/infra/ubuntu/dists/<release>-infra-security/main/binary-<arch>/">https://bearer:<token>@esm.ubuntu.com/infra/ubuntu/dists/<release>-infra-security/main/binary-<arch>/	オペレーティングシステムのセキュリティ更新
<a href="https://bearer:<token>@esm.ubuntu.com/apps/ubuntu/dists/<release>-apps-updates/main/binary-<arch>/">https://bearer:<token>@esm.ubuntu.com/apps/ubuntu/dists/<release>-apps-updates/main/binary-<arch>/	アプリケーションの機能更新
<a href="https://bearer:<token>@esm.ubuntu.com/apps/ubuntu/dists/<release>-apps-security/main/binary-<arch>/">https://bearer:<token>@esm.ubuntu.com/apps/ubuntu/dists/<release>-apps-security/main/binary-<arch>/	アプリケーションのセキュリティ更新

`<token>`は個人用ベアラートークンに置き換えてください。また、`arch`および`release`も次のいずれかの値に置き換える必要があります。

表 46. Ubuntu ESMのアーキテクチャとリリース

アーキテクチャ	リリース
amd64、arm64、armel、armhf、i386、powerpc 、ppc64el、s390x	bionic、focal、jammy、noble、trusty、xenial

Uyuniでリポジトリを同期するには、対応するGPGキー(ubuntu-advantage-esm-infra-trusty.gpg、ubuntu-advantage-esm-apps.gpg)をインポートする必要があります。これらのキーは、Ubuntu Proに登録されているシステムの`/etc/apt/trusted.gpg.d`にあります。これらのファイルをコピーしてUyuniシステムにコピーし、次のようにインポートします。

```
mgradm gpg add /path/to/gpg.key
```

既に同期済みのUbuntu親チャンネルの下に適切な子チャンネルを作成します。その後、リポジトリを同期できます。



ここで説明する手順を使用するとサブスクリプションの制限を回避できますが、これはサービス利用規約に違反し、法的な結果を招く場合があります。常に、使用するシステムの数に対して十分な数のサブスクリプションが存在する必要があります。

4.14.1.3. 同期ステータスの確認

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. [→] タブに移動し、[→] をクリックし、[→] 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。



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。

4.14.1.4. 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。

4.14.1.5. 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。



This issue happens with self-installed versions of Ubuntu. If the default user has been granted administrative privileges during installation time, a password is required to perform privilege escalation using sudo. With cloud instances this does not happen because cloud-init automatically creates a file under /etc/sudoers.d and grants privilege escalation through sudo without the need for a password.

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
```



このプロセッサにより 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。

4.14.1.6. 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。

4.15. プロキシへのクライアントの登録

プロキシサーバは、クライアントのためにプ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 Uyuni 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。クライアントのある Uyuni プロキシから別のプロキシまたは Uyuni サーバに移動する方法もあります。

Web UI 内では、クライアント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、[...] タブの [...] 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、[...] タブの [...] サブタブを選択できます。

4.15.1. 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、クライアント (Salt) および Salt SSH ブ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。

手順: プロキシ間でクライアントまたは Salt SSH プッシュクライアントを移動する

1. Uyuni の Web UI で、プロキシ間で移動するクライアントの [クライアント] ページに移動します。
2. [クライアント] タブに移動します。次に [クライアント] 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. [クライアント] 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 で複数のクライアントまたは Salt SSH プッシュクライアントをプロキシ間で移動する

1. Uyuni 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. [クライアント] 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。

4.15.1.1. 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 プッシュクライアントで異なります。

4.15.1.1.1. デフォルトのクライアント

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。

クライアント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。

4.15.1.1.2. 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。

4.15.2. プロキシからサーバへのクライアントの移動

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→→→を選択します。

4.15.3. 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を使用してクライアントをUyuni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の作成の詳細については、[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. UyuniのWeb UIで、**システム › ブートストラップ**に移動します。
2. [→→→→] 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→→→→] 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. [→→→→] 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、[→→→→→→→] 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秘密鍵認証の場合、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. [→→→→→→→→→→→→→→→→] 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. [→→→→→] 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厳格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.  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。

4.15.4. コマンドラインでクライアントを登録する

Web UIの代わりに、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できます。この手順では、登録する前にSaltパッケージを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ビ化しておく必要があります。

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 `PROXY123.EXAMPLE.COM` 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>'
```

4.16. パブリッククラウドでのクライアントの登録

Uyuni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。

4.16.1. 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Uyuni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、 [Specialized-guides > Public-cloud-guide](#) を参照してください。

4.16.2. 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。Uyuni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。

手順: 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. 自動登録のトリガー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
```

4.16.3. クライアントの登録

UyuniのWeb UIで、**システム**、**ブートストラップ**に移動し、[マシン名]、[SSHマシン名]、[マシン名]、および[マシン名]の各フィールドに入力します。[マシン名]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、Uyuniではホストを検索できません。

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、または多数のクライアントを登録する必要がある場合、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。

4.16.4. アクティベーションキー

アクティベーションキーを使用し、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。

4.16.5. 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Uyuni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

4.16.5.1. 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Uyuni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。

4.16.5.2. 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。

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、Uyuni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。

登録を取り消したopenSUSE Leapクライアントのアップグレードの詳細については、**Client-configuration** › **Client-upgrades-uyuni**を参照してください。

5.1. 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宣言
3. アクティベーションキーの作成
4. 自動インストールプロファイルの作成

5.1.1.1. インストールメディアの準備

手順: インストールメディアを準備する

1. コンテナホストで、インストールソースが含まれるISOイメージをダウンロードします。

- mgradmを使用して、ISOイメージからインストールデータをインポートします。

```
mgradm distribution copy <image_name>.iso <image_name>
```

- mgradmで報告されたディストリビューションのパスをメモしてください。このファイルパスは、ディストリビューションをUyuniに対して宣言するときに必要です。



このイメージは、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。

詳細については、[Client-configuration > Autoinst-distributions](#)を参照してください。

5.1.1.2. 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

手順: 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

- Uyuni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。
- [-----] をクリックし、次のフィールドに入力します。

[-----] フィールドに、自動インストールの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。たとえば、「sles15sp6-x86_64」と入力します。
 [-----] フィールドに、保存済みのインストールメディアへのパスを入力します。 * 対応する [-----] 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。 * [-----] 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。

- [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。

5.1.1.3. アクティベーションキーの作成

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

- Uyuni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、[-----] をクリックします。
- キーの説明を入力します。
- 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。
- オプション: 使用量を制限する場合、[-----] テキストフィールドに値を入力します。
- SLE-Product-SLES15-SP6-Pool for x86_64ベースチャンネルを選択します。
- オプション: [-----] 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. [-->] タブをクリックし、必要なチャンネルを選択します。
9. [キーの一更新] をクリックします。

5.1.1.4. 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。

AutoYaSTの有効な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. Uyuni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、以下を参照してください。

<https://github.com/SUSE/manager-build-profiles/tree/master/AutoYaST>を参照してください。

2. [-->] 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ができます。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、
[-->] 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、 [-->] 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>
```

5.1.2. 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. UyuniサーバのWeb UIで、[マニフェスト] 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. [マニフェスト] 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キップしてかみ替へます] ボタンをクリックします。
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、Uyuni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。



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

- 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。

- 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。
- 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。
- 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

- 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。

- 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: 自動インストール可能なディストリビューションの作成

- 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。

- 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。
- [-----] 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

- 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。

- 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_prefi
x-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. Uyuni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、[...] 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、コンテナ内でシェルに移動し、次のコマンドを実行します。

```
mgrctl term
```

1. 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

製品を移行すると、SLEベースのクライアントシステムをサービスパック(SP)レベルから最新版にアップグレードできます。たとえば、SUSE Linux Enterprise Server 15 SP1をSUSE Linux Enterprise Server 15 SP2に移行できます。

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
- openSUSE Leap 15.1からSUSE Linux Enterprise Server 15 SP1
- 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/en-us/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、Uyuni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 [...] 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。移行プロセージャが始まると、ロールバックできません。

プロセージャ: 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. [ベースチャンネル] ページから、正しいベースチャンネルを選択し、[追加] および追加のベースチャンネルを含めます。
5. オプション: [インストール] 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、[インストール] 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、Uyuni 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。

5.3.2.1. 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。

手順: 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

```

5.3.2.2. 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

- Uyuni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。
- 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** 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

- 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
```

- 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
```

5.3.2.2.1. 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、Uyuni Web UIからシステムを再起動できます。

5.4. Uyuniクライアントのアップグレード

このセクションでは、例としてopenSUSE Leapを使用します。

5.4.1. アップグレードの準備

プロシージャ: クライアントのアップグレードの準備

- Uyuni 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 \
opensuse_leap15_4\
opensuse_leap15_4-non-oss \
opensuse_leap15_4-non-oss-updates \
opensuse_leap15_4-updates \
opensuse_leap15_4-uyuni-client
```

- spacewalk-repo-syncを使用して、すべてのチャンネルを完全に同期します。 リポジトリのURLが定義済みの場合、[installation-and-upgrade:proxy-uyuni.pdf](#)で続行します。
- UyuniサーバのWeb UIで、**ソフトウェア**、**管理**、**チャンネル**に移動し、Uyuni Client Tools for openSUSE Leap 15.4 (x86_64)チャンネル名をクリックします。
- 右上隅の**[チャンネルの管理]**をクリックします。
- [リポジトリ] タブをクリックし、[External - Uyuni Client Tools for openSUSE Leap 15.3 (x86_64)] を選択します。
- [Update Repositories]** (リポジトリの更新) をクリックします。
- リポジトリ、同期サブタブに移動し、**[今すぐ同期]**をクリックします。
- openSUSE Leap 15.4 (x86_64)およびExternal - openSUSE Leap 15.3 (x86_64)で同じ操作をします。

openSUSE Leap 15.4 (x86_64)を展開し、パッケージが入力されているすべての子チャンネルを表示します。

5.4.2. アップグレード

クライアントをアップグレードするには、ソフトウェアリポジトリを置き換え、ソフトウェアを更新し、最後にクライアントを再起動します。

プロシージャ: クライアントのアップグレード

- UyuniサーバのWeb UIで、**[システム]** に移動し、クライアントの名前をクリックします。
- ソフトウェア**、**ソフトウェアチャンネル**をクリックし、一覧にリストされているopenSUSE Leap 15.5チャンネルをベースチャンネルとして選択します。

3. [――――――――――――] ペインで、15.5子チャンネルを選択します。
4. [次へ]、[――――――――――――――――]、[確認]をクリックします。
5. ソフトウェア、パッケージ、アップグレードをクリックし、クライアントで更新するパッケージをすべて選択してから、[パッケージのアップグレード]をクリックし、詳細を確認し、[確認]をクリックして更新を終了します。
6. クライアントを再起動します。

多数のクライアントを更新する必要がある場合、Uyuniサーバでこのコマンドシーケンスのアクションチェーンを作成できます。アクションチェーンを使用して、複数のクライアントで同時に更新を実行できます。

Chapter 6. クライアントの削除

クライアントをUyuniサーバから削除する必要がある場合、次の操作を実行できます。

- ・ Web UIを使用して削除する
- ・ コマンドラインからクライアントを削除する

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. Uyuni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. **[クライアントの削除]**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、Uyuni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。



クライアントをクリーンアップすると、Saltが無効化されて、可能であればサービスが停止されるだけです。パッケージはアンインストールされません。

6.2. コマンドラインで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



このプロセスはUyuniクライアントのみを対象としています。Uyuni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コマンドを使用します。

プロシージャ: SLES 12および15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 /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 - 手動登録のクリーンアップ

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
rm /etc/zypp/repos.d/susemanager\channels.repo /etc/venv-salt-
minion/*
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。

Chapter 7. クライアントの操作

登録、アップグレード、ソフトウェアのインストール、またはクライアントの削除に加え、他の操作も実行できます。Uyuniのクライアントは個別に管理することも、クライアントのセットとして管理することもできます。クライアントのセットは、一回限りの操作のためにシステムセットマネージャを使用してグループ化するか、システムグループを使用して恒久的にグループ化できます。

高度なクライアント設定オプションについては、一般的な設定管理を利用できます。カスタムシステム情報の取得、電源のオン/オフ、およびUyuni Web UIを使用したクライアントの再起動を行うことができます。

以下のセクションでは、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。



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。

クライアントでパッケージを管理するには、[メニュー] 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。

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. パッケージの検証

クライアントにインストールしたパッケージがインストール元のデータベースの現在の状態と一致していることを確認できます。インストールしたパッケージのメタデータが、ファイルのチェックサム、ファイルサイズ、パーミッション、オーナー、グループ、タイプなど、データベースの情報と比較されます。

プロシージャ: インストールしたパッケージの検証

1. UyuniのWeb UIで、[メニュー] に移動し、パッケージをインストールした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** > **検証**サブタブに移動します。
2. 検証するパッケージを選択し、**[選択したパッケージの検証]**をクリックします。
3. 検証を完了したら、**システム** > **イベント** > **履歴**に移動し、結果を表示します。

7.1.2. 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

- UyuniのWeb UIで、[...]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。
- [システム] [プロファイル] の [作成] をクリックします。
- 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

- UyuniのWeb UIで、[...]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。
- 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。

7.2. 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

- Uyuni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。
- [...]セクションで、次の詳細を使用します。
 - [...] フィールドにパッチの短い説明を入力します。
 - [...] 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。
 - [...] 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。
 - [...] 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 [...] を選択します。
 - [...] 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 [...] フィールドで選択します。
 - [...] 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。

- 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: [] 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。
- [] 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。
- [] 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。
- [] 、 [] 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. [] 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、パッケージの関連づけが保持され、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

- UyuniのWeb UIで、**パッチ** > **パッチの複製**に移動します。
- [] 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。
- 複製する1つまたは複数のパッチを選択し、[] をクリックします。
- 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。
- 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

- Uyuni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。
- パッケージ** > **追加**タブに移動します。
- [] 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、[] をクリックします。[All 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。
- 含めるパッケージを確認し、[] をクリックします。
- パッケージの詳細を確認し、[] をクリックしてパッチに適用します。
- 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] 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. UyuniのWeb UIで、**パッチ**タブに移動します。
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] 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. Uyuni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. Uyuni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[編集]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. Uyuni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、[クライアント] 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、[]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。

7.3.1. クライアントのシステムのロック

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、実行コマンドが無効になり、黄色のバナーが [→→→→→→→→→→] 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: クライアントのシステムのロック

1. UyuniのWeb UIで、ロックするクライアントの [→→→→→→→→→→] ページに移動します。
2. [→] タブに移動し、システムのロックの式にチェックを付け、[]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、[]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.saltstack.com/en/latest/topics/blackout/index.html>を参照してください。

7.3.2. 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。 SUSE Linux EnterpriseおよびopenSUSE (zyppベース)のクライアントと、Red Hat Enterprise LinuxまたはDebianのクライアントを区別する必要があります。

7.3.2.1. 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。Uyuni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. 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。

7.3.2.2. 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、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。Uyuni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`をインストールし、Red Hat Enterprise Linux 8または9システムでは、`python3-dnf-plugin-versionlock`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。

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。Uyuni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、Uyuni Serverにサブスクライブしている任意のクライアントに適用できます。

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。

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。

次の表に、サポートされている機能を示します。各記号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、後日使用できる場合と、使用できない場合があります。

表 47. 設定管理でサポートされる機能

機能	ステータス
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. 設定チャンネルの作成

7.4.1.1. 一元的な設定チャンネル

新しいセントラル設定チャンネルを作成するには:

プロシージャ: セントラル設定チャンネルの作成

1. Uyuni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。

7.4.1.2. Saltの状態チャンネル

設定チャンネルを使用して、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. Uyuni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~~~~~] 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。

7.4.2. 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. Uyuni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、[-----/-----] 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. [――――――] フィールドおよび [――――――――――――――――] にファイルの [――――――] および [――――――――] 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. [――――――――――] 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。

7.4.3. 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、[――] 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. Uyuni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: [――――――――――――――――――――] 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 **[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 **[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 **[既存のサブスクライブーションを置換]** 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. **[サブスクライブーションの適用]** 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。

7.4.4. 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、Uyuni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. Uyuni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: [~~~~~] 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。

7.4.5. 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
```

7.5. 電源管理

Uyuni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Uyuni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、Uyuni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. Uyuni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、**プロジェクト**、**電源管理**タブに移動します。

2. [――――――] 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。Uyuni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. 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key:valueペアで定義され、クライアントに割り当てるすることができます。たとえば、特定のプロセッサに対してkey:value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することができます。カスタムシステム情報は分類され、Uyuni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. Uyuni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. [――――――――] 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. [――] 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。

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、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. UyuniのWeb UIで、[カスタム情報]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. [→] 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。

7.7. 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場合と、サポートされない場合があります。

表 48. 使用可能なSSMアクション

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	✓
パッケージの確認	✗
グループの作成	✓

システムのリスト	✓
グループの管理	✓
チャンネルのメンバーシップ	✓
チャンネルの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	✓
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。

システムグループに移動し、操作するグループにチェックを付け、[イ] [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 [-----] アイコンをクリックします。



SSMの詳細は、Uyuni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。

7.7.1. 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. Uyuni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. [-----] 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、[-----] 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. [-----] 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、[-----] 、[-----] 、または [-----] 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。

7.8. 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、Webサーバーソフトウェアが含まれるすべてのクライアント用に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、Webサーバーソフトウェアが含まれるすべての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、Webサーバーソフトウェアが含まれるすべてのSLESクライアントの更新を実行したりできます。

7.8.1. 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. Uyuni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。

7.8.2. 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. Uyuni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. Uyuni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. Uyuni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、[→→→→→] タブに移動します。
3. 参加するグループを見つけ、[→→→] 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、[→→→→→] タブに移動します。

7.8.3. 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。

Uyuni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ご操作可能です。グループを選択し、**[ヨリオングで作業する]**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]**

[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、Webサーバソフトウェアが含まれるすべてのクライアント用に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フェクションは、Webサーバソフトウェアが含まれるすべてのSLESクライアントになります。

7.9. 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は、すべてのクライアントでSaltです。

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――――――――――――、ビルドホストとして動作するクライアントでは――――――――――――――――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、

[――――――――――――――――――] を選択し、**[エントリーツールメントの追加]**または**[エントリートル] [除]**をクリックします。

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。 Uyuni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Uyuni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、Uyuni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Uyuni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。

Uyuni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
- ・ Uyuni 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Uyuni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。

- ・ Uyuni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。

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。

Uyuni 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。

Uyuniの自動インストール機能は、Cobblerという名前のソフトウェアによって実行されます。Cobblerの詳細については、<https://cobbler.readthedocs.io>を参照してください。



SUSEは、Uyuni Web UIまたはUyuni 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。

手順: 登録済みのクライアントを再インストールする

1. Uyuni 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. 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**[-----]**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. CD-ROMまたはUSBメモリを使用してインストールする

Uyuni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。

8.2.1. 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。

```
cobbler buildiso --iso=/path/to/boot.iso --distro=<your-distro-label>
```



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によって保存されている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。次に例を示します。

```
cobbler buildiso --systems="system1 system2 system3" \
--profiles="<your-profile1-label> <your-profile2-label> <your-profile3-label>" --distro=<your-distro-label>
```



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。

8.2.2. 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Uyuni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。

8.2.3. 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。

8.3. 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。

Uyuniは、mgradmツールを使用してソースからサーバコンテナにインストールファイルをコピーします。

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、Uyuni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。

8.3.1. 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。

手順: インストールメディアからのファイルのインポート

1. mgradmを使用してISOイメージからインストールデータをインポートします。

```
# mgradm distribution copy <image_name>.iso <image_name>
```

- mgradmで報告されたディストリビューションの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Uyuniに対して宣言するときに必要です。

8.3.1.1. ディストリビューションの自動検出と登録

mgradmは、ディストリビューションの名前を自動的に検出してサーバに登録できます。提供されているISOイメージに.treeinfoファイルが含まれている必要があります。

手順: 自動検出と登録を使用してディストリビューションファイルをインポートする

- mgradmを使用します。

```
# mgradm distribution copy --api-user <username> --api-password <password> <image_name>.iso
```

8.3.2. 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。

手順: インストールパッケージからファイルを抽出する

- Uyuni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。
- 次のコマンドを使用して別のルートにパッケージをインストールし、再起動しなくても済むようにすることができます。

```
# mkdir /opt/tftpinstall
# zypper --installroot /opt/tftpinstall install tftpboot-
installation-SLE-Micro-5.5-x86_64
```

- コマンドls -d /opt/tftpinstall/usr/share/tftpboot-installation/*を使用してインストールファイルを見つけます。
- mgradmを使用してインストールファイルをコピーします。

```
# mgradm distribution copy /opt/tftpinstall/usr/share/tftpboot-
installation/SLE-Micro-5.5-x86_64 SLE-Micro-5.5-x86_64
```

- mgradmツールで報告されたディストリビューションの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Uyuniに対して宣言するときに必要です。
- mgradmツールが完了したら、/opt/tftpinstallディレクトリを削除できます。

この手順では、Uyuni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。 Uyuniの [-----] 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。

8.3.3. 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。

手順: 自動インストールのディストリビューションの宣言

1. Uyuni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. [-----] をクリックし、次のフィールドに入力します。
 - [-----] 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。
 - [-----] フィールドに、Uyuni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 [-----] 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。
 - [-----] 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、[-----] 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、[-----] フィードに`insecure=1`を追加するか、**Client-configuration** › **Autoinst-ownpgpkey**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。

8.4. 自動インストールプロファイル

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。

Uyuni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openSUSE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。

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。

8.4.1. 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、Uyuni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。

手順: 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. Uyuni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|一|ト/AutoYaST|フ|ア|イ|ル|を|ア|ッ|フ|ロ|ー|ド]** をクリックします。
3. **[~~~~~]** 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. **[~~~~~]** 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. **[~~~~~]** 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 **[~~~]** を選択します。
6. 自動インストールファイルの内容を **[~~~~~]** フィールドにコピーするか、または **[~~~~~]** 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. Uyuni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. [名前] 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] 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. [Uyuni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] で、Uyuni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、

[最新ディストリビューション] 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Uyuni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、Uyuni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。

-----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、[ディストリビューション] 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。

8.4.2. 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-----、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-----、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。

Uyuni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>を使用すると、子チャンネルをインストールに追加できます。例については、<https://doc.opensuse.org/projects/autoyast/#Software-Selections-additional>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。
 - 「<software>」の例について
は、<https://doc.opensuse.org/projects/autoyast/#CreateProfile-Software>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Uyuni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。

8.4.3. 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。

手順: ファイル保持一覧の作成

1. UyuniのWeb UIで、**自動テンプレート保持**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。

8.4.4. 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、Uyuni Web UI内で作成できます。 プロファイル内の[...]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。

8.4.4.1. 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[...]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、 [...] 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[...]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。

8.4.4.2. コードスニペット

Uyuni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。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')
```

8.4.4.3. 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)
```

8.5. 無人プロビジョニング

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。

手順: 手動で宣言したプロファイルからの再インストール

1. Uyuni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ーフェース名に、<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`のAPIドキュメントを参照してください。

8.6. 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。

8.6.1. 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。



サーバコンテナ内でシェルにアクセスするには、コンテナホストで`mgrctl term`を実行します。

手順: 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 tftpboot/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
tftpboot/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 tftpboot
find . | cpio -o -H newc | xz --check=crc32 -c >> /path/to/initrd
```

8.6.2. 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。

手順: 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

Uyuniを使用して、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、Uyuni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Uyuni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、Uyuni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。 Uyuni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Uyuni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Uyuni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 [System] 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。 [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。 [System] システムタイプがリストされていない場合、[System] の式を初期化します。 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 [System] 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 [System] タブを使用できます。 [System] 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

Uyuni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. Uyuni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、 [System] タブに移動します。
2. [System] セクションで、次の詳細を入力します。
 - [System] 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。
 - [完全・部分的] 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。
 - [ゲストディスク] フィールドに、ゲストディスクの最大サイズ制限(MB単位)を入力します。
 - [vCPU] で、ゲストのvCPUの数を入力します。
 - [エミュレートCPU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。
 - [インストールツール] 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。
[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ネットワークインターフェースの詳細を入力します。[MAC] 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. [グラフィックスドライバ] 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。

Uyuni 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つでプロシージャに従います。
 - ノード名 フィールドがチェックされていることを確認します。
 - VM構成 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ーへのパスを入力します。
 - ストレージ プール フィールドに、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. 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Uyuniで直接管理できます。

まず、Uyuniサーバで仮想ホストを設定する必要があります。追加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。

9.3.1. 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。

-----式は、ホストの初期化を支援します。
-----を参照してください。

詳細については、[client-configuration:virt-xenkm.pdf](#)

9.3.1.1. 背景情報

Uyuni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、調整する必要はありません。ただし、ゲストのVNCコンソールに非rootユーザとしてアクセスする場合、設定変更を実行する必要があります。VNCコンソールの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。

Uyuni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、そのスクリプトを使用してホストをUyuni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。

9.3.1.2. -----の初期化

-----式で、ホストを初期化します。

プロシージャ: -----の初期化

1. UyuniのWeb UIで、ホストの [-----] ページに移動し、[-----] タブをクリックします。
2. -----式を選択し、[-----] をクリックします。
3. -----サブタブをクリックします。
4. 設定を確認して、[-----] 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
```

9.3.2. 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。

ゲスト登録先のVMホストでアクティベーションキーがそれぞれのゲストで必要です。アクティベーションキーには、`XXXXXXXXXXXXXX`のエンタイトルメントと`XXXXXXXXXXXXXX`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Uyuni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の詳細については、**Client-configuration > Registration-bootstrap**を参照してください。

9.3.2.1. 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

Uyuniからクライアントを自動インストールできる自動インストール可能なディストリビューションをVMに作成する必要があります。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から使用できたり、ループマウントされたISOイメージで使用できます。

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、Red Hat Enterprise LinuxまたはSUSE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、SUSE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 49. 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
SUSE	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、リモートリソース上、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、全ユーザが読み取り可能であることを確認してください。

プロシージャ: 自動インストール可能なディストリビューションの作成

1. Uyuni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. [固有名前] セクションで、次のパラメータを使用します。
 - [固有名前] 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。

- [――――――――――] 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。
- [――――――――――――] 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。
- [――――――――――――] 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。
- [――――――――――――] 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`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 [――――――――――――――] 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

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するとき、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。

9.3.2.2. 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Uyuni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**、**新しいkickstart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. Uyuni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. [-----] セクションで、次のパラメータを使用します。
 - [-----] 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。
 - [-----] 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。
 - [-----] フィールドで、関連するゲストの種類を選択します(KVMなど)。ここでは、[Xen] 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、[-----] フィールドを空白のままにします。
 - [-----] 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 [-----] 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するとき、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 [-----] 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。 [-----] 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。

9.3.2.3. 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とき、Uyuni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. Uyuni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Uyuni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。

9.3.2.4. 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. Uyuni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. [――――――] タブ、[――――――――――――] 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キップ]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。

9.3.3. VMゲストの管理

Uyuni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Uyuni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。

Uyuni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。
[...] 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、Uyuni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。

Uyuni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。
- 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。

10.1. 仮想ホストマネージャおよび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、Uyuni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Uyuniサーバ上で動作します。

`virtual-host-gatherer-libcloud`パッケージをUyuni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. Uyuni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。

- [Name] 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. [-----] ページで、新しいVHMを選択します。
6. [-----] 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Uyuni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. 仮想ホストマネージャおよび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、Uyuni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Uyuniサーバ上で動作します。

`virtual-host-gatherer-libcloud`パッケージをUyuni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. Uyuni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。
 - [Name] 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. [VMs] ページで、新しいVHMを選択します。
6. [VMs] ページで、[評価] 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/virtualMachines/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、Uyuni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Uyuniサーバに報告できます。

```
13f56399-bd52-4150-9748-7190aae1ff21
```

10.3. 仮想ホストマネージャおよび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、Uyuni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Uyuniサーバ上で動作します。

VHMを実行するには、Uyuni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Uyuni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Uyuni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. Uyuni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。
 - [名前] 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Uyuniサーバのローカルパスを入力します。
 - [Project 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. [**VM**] ページで、新しいVHMを選択します。
6. [**評価**] 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、Uyuni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Uyuniサーバに報告できます。

152986662232938449

10.4. Nutanixによる仮想化

Uyuni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、Uyuni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。

10.4.1. VHMの設定

仮想ホストマネージャ(VHM)はUyuniサーバ上で動作します。

virtual-host-gatherer-NutanixパッケージをUyuni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、Uyuni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. Uyuni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。
 - **[名前]** フィールドにVHMのカスタム名を入力します。
 - **[完全修飾ドメイン名]** 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。

- [――――――] 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)。
- [――――――――] 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。
- [――――――――――] 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. [――――――――――――――――] ページで、新しいVHMを選択します。

6. [――――――――――] 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、――――――――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Uyuni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、**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Uyuni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。

10.5. VMWareによる仮想化

Uyuni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、Uyuni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。

10.5.1. VHMの設定

仮想ホストマネージャ(VHM)はUyuniサーバ上で動作します。

VHMを実行するには、Uyuni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Uyuni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Uyuni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。

ユーザ、ロール、およびパーミッションの詳細については、VMWare vSphereのドキュメント(<https://docs.vmware.com/en/VMware-vSphere/index.html>)を参照してください。

プロシージャ: VMWare VHMの作成

1. Uyuni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-based Virtual Host Manager]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。
 - [名前]フィールドにVHMのカスタム名を入力します。
 - [完全修飾ドメイン名(FQDN)] 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。
 - [ESXi API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)。
 - [VMホストユーザー名] 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。
 - [VMホストユーザーのパスワード] フィールドに、VMホストユーザー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. [VMWare-based Virtual Host Manager]ページで、新しいVHMを選択します。
6. [データの更新]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、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、**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Uyuni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。

10.5.2. 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、Uyuni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Uyuni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. Uyuni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
```

10.6. 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Uyuni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Uyuniサーバからアクセスできる場所に保存します。
3. Uyuni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、[-----]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。

- [名前] 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設定ファイ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. [新しいVHM] ページで、新しいVHMを選択します。

8. [新しいVHM] ページで、[評価] 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": {
        "vCenter": "564d6d90-459c-2256-8f39-3cb2bd24b7b0"
      }
    }
  }
}
```

```
"49737e0a-c9e6-4ceb-aef8-6a9452f67cb5": "4230c60f-3f98-  
2a65-f7c3-600b26b79c22",  
    "5a2e4e63-a957-426b-bfa8-4169302e4fdb": "42307b15-1618-  
0595-01f2-427ffcd88e",  
        "NSX-gateway": "4230d43e-aafe-38ba-5a9e-3cb67c03a16a",  
        "NSX-13gateway": "4230b00f-0b21-0e9d-dfde-  
6c7b06909d5f",  
            "NSX-service": "4230e924-b714-198b-348b-25de01482fd9"  
        }  
    }  
}  
}
```

詳細については、Uyuni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、ハイパーテイプの「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".