



U Y U N I

Uyuni 2022.11

管理者ガイド

2022年11月21日



目次

管理ガイドの概要	1
1. 動作	2
1.1. 定期的なアクション	2
1.2. 動作チェーン	3
1.3. リモートコマンド	4
2. 認証方法	7
2.1. シングルサインオン(SSO)による認証	7
2.1.1. 前提条件	7
2.1.2. SSOの有効化	8
2.2. PAMによる認証	9
3. バックアップと復元	11
3.1. Uyuniのバックアップ	11
3.2. smdbaによるデータベースの管理	13
3.3. smdbaによるデータベースのバックアップ	14
3.3.1. データベースの手動バックアップの実行	14
3.3.2. 自動バックアップのスケジュール	15
3.4. バックアップからの復元	16
3.5. アーカイブログ設定	16
3.6. 占有されているデータベーススペースの概要を取得する	17
3.7. データベースの移動	18
3.8. クラッシュしたルートパーティションからの回復	19
3.9. データベース接続情報	20
4. コンテンツのステージング	21
4.1. コンテンツステージングの有効化	21
4.2. コンテンツステージングの設定	21
5. チャンネル管理	23
5.1. チャンネル管理	23
5.2. チャンネルの削除	23
5.3. カスタムチャンネル	24
5.3.1. カスタムチャンネルおよびリポジトリの作成	24
5.3.2. カスタムチャンネル同期	28
5.3.3. カスタムチャンネルへのパッケージとパッチの追加	29
5.3.4. カスタムチャンネルの管理	29
6. コンテンツライフサイクル管理	31
6.1. コンテンツライフサイクルプロジェクトの作成	31
6.2. フィルタタイプ	32
6.2.1. フィルタの <code>rule</code> パラメータ	33
6.3. フィルタテンプレート	33
6.3.1. SUSE製品に基づくライブパッチ処理	33
6.3.2. システムに基づくライブパッチ処理	34
6.3.3. デフォルトのAppStreamモジュール	35
6.4. コンテンツライフサイクルプロジェクトの構築	36
6.5. 環境のプロモート	37
6.6. 環境にクライアントを割り当てる	37
6.7. コン텐ツライフサイクル管理の例	37
6.7.1. 月次パッチサイクルのプロジェクトの作成	37
6.7.2. 既存の月次パッチサイクルの更新	40
6.7.3. ライブパッチ処理でプロジェクトを強化	40
6.7.4. ライブパッチ処理用の新しいカーネルバージョンに切り替える	41
6.7.5. AppStreamフィルタ	42
7. 切断されたセットアップ	44

7.1. RMTの同期	44
7.2. SMTの同期.....	45
7.3. 必須チャンネル.....	46
7.4. 切断されたサーバの同期.....	47
8. ディスク容量の管理	48
8.1. 監視対象ディレクトリ.....	48
8.2. しきい値.....	48
8.3. サービスのシャットダウン.....	49
8.4. スペースチェックの無効化.....	49
9. イメージの構築と管理	50
9.1. イメージの構築の概要.....	50
9.2. コンテナイメージ.....	50
9.2.1. 要件	51
9.2.2. 構築ホストの作成	51
9.2.3. コンテナ用アクティベーションキーの作成	52
9.2.4. イメージストアの作成	52
9.2.5. イメージプロファイルの作成	53
9.2.6. イメージの構築	57
9.2.7. イメージの取り込み	57
9.2.8. トラブルシューティング	58
9.3. OSイメージ	58
9.3.1. 要件	58
9.3.2. 構築ホストの作成	59
9.3.3. OSイメージ用アクティベーションキーの作成	61
9.3.4. イメージストアの作成	62
9.3.5. イメージプロファイルの作成	63
9.3.6. イメージの構築	66
9.3.7. トラブルシューティング	67
9.3.8. 制限事項	67
9.4. ビルドイメージのリスト	68
10. インフラストラクチャ保守タスク	69
10.1. サーバ	69
10.1.1. クライアントツール	69
10.2. サーバ間同期スレーブサーバ	69
10.3. モニタリングサーバ	70
10.4. プロキシ	70
11. サーバ間同期	71
11.1. サーバ間同期 - バージョン2	72
11.1.1. ISSパッケージのインストール	72
11.1.2. コンテンツ同期	72
11.1.3. データベース接続設定	73
11.1.4. 既知の制限事項	73
12. SUSE Managerによるライブパッチ処理	74
12.1. ライブパッチ処理用のチャンネルの設定	74
12.1.1. ライブパッチ処理用のspacewalk-manage-channel-lifecycleを使用する	74
12.2. SLES 15でのライブパッチ処理	76
12.3. SLES 12でのライブパッチ処理	78
13. メンテナンスウィンドウ	80
13.1. メンテナンススケジュールタイプ	82
13.2. 制限されたアクションと制限されないアクション	83
14. <code>mgr-sync</code> の使用	84
15. PrometheusとGrafanaを使用したモニタリング	86
15.1. PrometheusとGrafana	86
15.1.1. Prometheus	86

15.1.2. Prometheus Exporters	86
15.1.3. Grafana	87
15.2. モニタリングサーバの設定.....	87
15.2.1. Prometheusのインストール.....	87
15.2.2. Grafanaのインストール.....	89
15.3. Uyuniのモニタリングの設定	90
15.3.1. サーバ自己監視	90
15.3.2. 管理対象システムの監視	93
15.3.3. Grafanaパスワードの変更	94
15.4. ネットワーク境界	94
15.4.1. リバースプロキシのセットアップ	95
15.5. セキュリティ	95
15.5.1. TLS証明書の生成	96
16. 組織	97
16.1. 組織の管理	97
16.1.1. 組織ユーザ	98
16.1.2. 組織の作成	98
16.1.3. 組織の設定	98
16.2. 状態の管理	98
16.2.1. 設定チャンネルの管理	98
17. パッチ管理	99
17.1. 撤回されたパッチ	99
17.1.1. チャンネルクローン	99
17.1.2. パッチの共有	100
18. レポートの生成	101
18.1. <code>spacewalk-report</code> の使用	101
18.2. <code>spacewalk-report</code> およびレポーティングデータベース	101
18.3. 使用可能なレポートのリスト	102
19. セキュリティ	107
19.1. クライアントをマスター検証指紋に設定する	107
19.2. リポジトリメタデータの署名	107
19.3. ミラーソースパッケージ	109
19.4. OpenSCAPによるシステムセキュリティ	110
19.4.1. SCAPについて	110
19.4.2. SCAPスキャンのためのクライアントの準備	110
19.4.3. OpenSCAPコンテンツファイル	111
19.4.4. OpenSCAPプロファイルの検索	112
19.4.5. 監査スキャンの実行	114
19.4.6. スキャン結果	115
19.4.7. 修復	115
19.5. 監査	119
19.5.1. CVE監査	120
19.5.2. CVEステータス	121
20. SSL証明書	122
20.1. 自己署名SSL証明書	123
20.1.1. 既存のサーバ証明書の再作成	123
20.1.2. 新しいCAおよびサーバ証明書の作成	123
20.2. SSL証明書のインポート	124
20.2.1. 新しいインストール用証明書のインポート	125
20.2.2. 新しいプロキシインストール用の証明書のインポート	125
20.2.3. 証明書を置き換える	126
20.3. HTTP Strict Transport Security	127
21. サブスクリプションマッチング	129
21.1. クライアントをサブスクリプションにピン設定する	130

22. タスクスケジュール	131
23. 変更ログの調整	134
24. ユーザー	135
24.1. アカウントの無効化と削除	135
24.2. 管理者ロール	135
24.3. ユーザ許可とシステム	136
24.4. ユーザとチャンネルの許可	136
24.5. ユーザのデフォルト言語	137
24.5.1. ユーザデフォルトのインターフェイステーマ	138
25. トラブルシューティング	139
25.1. 自動インストールのトラブルシューティング	139
25.2. ベアメタルシステムのトラブルシューティング	139
25.3. サポート終了製品のブートストラップリポジトリのトラブルシューティング	140
25.4. クライアントが複製したSaltクライアントのトラブルシューティング	141
25.5. 破損したリポジトリのトラブルシューティング	141
25.6. パッケージが競合するカスタムチャンネルのトラブルシューティング	141
25.7. FQDNS grainの無効化のトラブルシューティング	142
25.8. ディスク容量のトラブルシューティング	143
25.9. ファイアウォールのトラブルシューティング	143
25.10. WAN接続を介したUyuniサーバとプロキシ間の長い同期時間に関するトラブルシューティング	144
25.11. 無効なクライアントのトラブルシューティング	147
25.12. サーバ間同期のトラブルシューティング	147
25.13. ローカル発行者証明書のトラブルシューティング	148
25.14. ログインタイムアウトのトラブルシューティング	148
25.15. メール設定のトラブルシューティング	149
25.16. noexecで/tmpをマウントする場合のトラブルシューティング	149
25.17. noexecで/var/tmpをマウントする場合のトラブルシューティング	150
25.18. 十分なディスク容量がない場合のトラブルシューティング	150
25.19. 通知のトラブルシューティング	150
25.20. OSADとjabberdのトラブルシューティング	151
25.21. パッケージの不整合のトラブルシューティング	152
25.22. プロキシ経由のリポジトリの問題のトラブルシューティング	152
25.23. grainを開始イベントに渡す場合のトラブルシューティング	152
25.24. プロキシの接続およびFQDNのトラブルシューティング	153
25.25. クローンクライアントの登録のトラブルシューティング	153
25.26. Web UIからの登録が失敗し、エラーが表示されない場合のトラブルシューティング	156
25.27. 従来のクライアントを削除した後、Salt minionとして登録する場合のトラブルシューティング	157
25.28. 従来のRed Hatクライアントの登録のトラブルシューティング	157
25.29. Red Hat CDNチャンネルと複数の証明書のトラブルシューティング	157
25.30. Uyuniサーバの名前変更のトラブルシューティング	159
25.31. xref:troubleshooting/tshoot-retrying-setup-target-system.adoc[ターゲットシステムの設定を再試行する場合のトラブルシューティング]	160
25.32. RPC接続タイムアウトのトラブルシューティング	160
25.33. ダウンと表示されるSaltクライアントとDNS設定のトラブルシューティング	161
25.34. Saltboot Formulaのトラブルシューティング	161
25.35. スキーマのアップグレードが失敗する場合のトラブルシューティング	162
25.36. 同期のトラブルシューティング	163
25.37. Taskomaticのトラブルシューティング	164
25.38. Web UIの読み込みが失敗する場合のトラブルシューティング	165
26. GNU Free Documentation License	166

管理ガイドの概要

更新: 2022-11-21

このドキュメントでは、Uyuniサーバで管理タスクを実行する方法について説明します。

Chapter 1. 動作

クライアントに対するアクションは、さまざまな方法で管理することができます。

Saltクライアントの場合、自動化された定期的なアクションをスケジュールして、指定されたスケジュールでクライアントにハイステートを適用できます。定期的なアクションは、個々のクライアント、システムグループ内のすべてのクライアント、または組織全体に適用できます。

Saltクライアントと従来のクライアントの両方で、アクションチェーンを作成することにより、特定の順序で実行されるアクションを設定できます。アクションチェーンは事前に作成および編集でき、適切な時間に実行するようにスケジュールできます。

また、1つ以上のSaltクライアントに対してリモートコマンドを実行することもできます。リモートコマンドを使用すると、個々のSaltクライアント、または検索語に一致するすべてのクライアントにコマンドを発行できます。

1.1. 定期的なアクション

個々のSaltクライアント、または組織内のすべてのクライアントに定期的なアクションを適用することができます。

手順: 新しい定期的なアクションを作成する

1. 個々のクライアントに定期的なアクションを適用するには、[システム]に移動し、クライアントをクリックしてスケジュールを設定し、**状態 > 繰り返し状態タブ**に移動します。
2. システムグループに定期的なアクションを適用するには、**システム > システムグループ**に移動し、スケジュールを設定するグループを選択して、**状態 > 繰り返し状態タブ**に移動します。
3. [作成]をクリックします。
4. 新しいスケジュールの名前を入力します。
5. 定期的なアクションの頻度を選択します。
 - **毎時:** 各時間の分を入力します。たとえば、**15**は毎時15分にアクションを実行します。
 - **毎日:** 毎日の時間を選択します。たとえば、**01:00**は、Uyuniサーバのタイムゾーンで、毎日0100にアクションを実行します。
 - **毎週:** 指定した時刻に毎週アクションを実行する曜日と時刻を選択します。
 - **毎月:** 指定した時刻に毎月アクションを実行する日付と時刻を選択します。
 - **独自のクオーツ書式オプション:** 独自のクオーツ文字列を入力します。たとえば、毎月毎週土曜日の0215に定期的なアクションを実行するには、次のように入力します。

```
0 15 2 ? * 7
```

6. オプション[テストモード]スイッチを切り替えて、テストモードでスケジュールを実行します。

7. Click **[スケジュールの作成]** をクリックして保存し、既存のスケジュールの完全なリストを表示します。

組織管理者は、組織内のすべてのクライアントに定期的なアクションを設定および編集できます。 [ホーム > 組織 > 繰り返し状態](#) に移動して、組織全体に適用されるすべての定期的なアクションを表示します。

Uyuni管理者は、すべての組織のすべてのクライアントの定期的なアクションを設定および編集できます。 [管理 > 組織](#) に移動し、管理する組織を選択して、[状態 > 繰り返し状態](#) タブに移動します。



定期的なアクションは、Saltクライアントでのみ使用できます。 グループまたは組織の従来のクライアントは無視されます。

1.2. 動作チェーン

クライアントに対して多数の連続動作を実行する必要がある場合は、順序を確実に反映するための動作チェーンを作成できます。

デフォルトでは、ほとんどのクライアントはコマンドが発行されるとすぐに動作を実行します。 場合によつては、動作に時間がかかることがあります。これは、その後に実行された動作が失敗することを意味します。 たとえば、クライアントに再起動を指示してから2番目のコマンドを発行すると、再起動がまだ行われているため、2番目の動作が失敗する可能性があります。 アクションが正しい順序で実行されるようにするには、動作チェーンを使用します。



トランザクション更新システムの場合、動作チェーンは、再起動動作があるまで単一スナップショット内で実行されます。 これによりいくつかの制限が発生する可能性があります。

詳細については、[Client-configuration > Clients-slemicro](#) および [Client-configuration > Clients-microos](#) を参照してください。

動作チェーンは、従来のクライアントとSaltクライアントの両方で使用できます。 動作チェーンには、次の動作を任意の数、任意の順序で含めることができます。

- [システムの詳細 > リモートコマンド](#)
- [システムの詳細 > システムの再起動をスケジュール](#)
- [システムの詳細 > 状態 > highstate](#)
- [システムの詳細 > ソフトウェア > パッケージ > 一覧表示/削除](#)
- [システムの詳細 > ソフトウェア > パッケージ > インストール](#)
- [システムの詳細 > ソフトウェア > パッケージ > アップグレード](#)
- [システムの詳細 > ソフトウェア > パッチ](#)
- [システムの詳細 > ソフトウェア > ソフトウェアチャンネル](#)
- [システムの詳細 > 設定](#)

・ イメージ > ビルド

プロシージャ: 新しい動作チェーンの作成

1. Uyuni Web UIで、動作チェーンで実行する最初の動作に移動します。たとえば、クライアントの [シス テム の 詳 細] に移動し、[シス テム の 再 起 動 を スケ ジュ ー ル] をクリックします。
2. [以 下 に 追 加] フィールドをオンにし、追加する動作チェーンを選択します。
 - これが最初の動作チェーンの場合は、[新 し い 動 作 チ ェ ー ン] を選択します。
 - 動作チェーンがすでに存在する場合は、リストから選択します。
 - 既存の動作チェーンがすでにあるが、新しい動作チェーンを作成する場合は、作成する新しい動作チェーンの名前の入力を開始します。
3. 動作を確認します。動作はすぐには実行されず、新しい動作チェーンが作成され、これを確認する青いバーが画面の上部に表示されます。
4. [以 下 に 追 加] フィールドをオンにし、追加する動作チェーンの名前を選択して、動作チェーンに動作の追加を続行します。
5. 動作を追加し終えたら、スケジュール > 動作チェーンに移動して、リストから動作チェーンを選択します。
6. 動作をドラッグして正しい位置にドロップすることで、動作の順序を変更します。青いプラス記号をクリックして、動作が実行されるクライアントを表示します。[保 存] をクリックして、変更を保存します。
7. 実行する動作チェーンの時刻をスケジュールし、[保 存] と [スケ ジュ ー ル] のいずれかをクリックせずにページを離れる場合、未保存の変更はすべて破棄されます。



動作チェーン内の1つの動作が失敗すると、動作チェーンが停止し、これ以上の動作は実行されません。

スケジュール > 待機中の動作に移動して、動作チェーンからスケジュール済みの動作を確認できます。

1.3. リモートコマンド

リモートでコマンドを実行するようにクライアントを設定できます。これにより、クライアントに直接アクセスすることなく、スクリプトまたは個々のコマンドをクライアントに発行できます。

この機能はSaltクライアントで自動的に有効になるため、これ以上の設定を行う必要はありません。従来のクライアントでは、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登録し、リモートコマンドを有効にしている場合はこの機能が有効になります。代わりに、このプロシージャを使用して手動で有効にすることもできます。

開始する前に、インストールされているオペレーティングシステムに適したツールの子チャンネルにクライアントがサブスクライブされていることを確認してください。ソフトウェアチャンネルへのサブスクライブの詳細については、Client-configuration > Channelsを参照してください。



トランザクション更新システムの場合は、リモートコマンドが単一スナップショット内で実行されることを考慮します。これによりいくつかの制限が発生する可能性があります。

詳細については、[Client-configuration](#) > [Clients-slemicro](#)および[Client-configuration](#) > [Clients-microos](#)を参照してください。

プロシージャ: リモートコマンドを受け入れるように従来のクライアントを設定する

1. クライアントのコマンドプロンプトで、パッケージマネージャを使用して `rhncfg`、`rhncfg-client`、および `rhncfg-actions` パッケージをインストールします(まだインストールされていない場合)。例:

```
zypper in rhncfg rhncfg-client rhncfg-actions
```

2. クライアントのコマンドプロンプトで、rootとして、ローカル設定ディレクトリのパスを作成します。

```
mkdir -p /etc/sysconfig/rhn/allowed-actions/script
```

3. 新しいディレクトリに `run` という空のファイルを作成します。このファイルは、Uyuniサーバにリモートコマンドを実行するための許可を与えます。

```
touch /etc/sysconfig/rhn/allowed-actions/script/run
```



Saltクライアントの場合、リモートコマンドはクライアントの `/tmp/` ディレクトリから実行されます。リモートコマンドが正確に機能するようにするには、`noexec` オプションを指定して `/tmp` をマウントしないでください。 詳細については、[Administration](#) > [Troubleshooting](#)を参照してください。



[リモートコマンド] ページから実行されるすべてのコマンドは、クライアント上でrootとして実行されます。ワイルドカードを使用して、任意の数のシステムでコマンドを実行できます。コマンドを発行する前に、必ず十分注意してコマンドを確認してください。

プロシージャ: 従来のクライアントでのリモートコマンドの実行

1. WebUIで、[システム]に移動して、リモートコマンドを実行するクライアントをクリックし、[詳細](#) > [リモートコマンド](#)タブに移動します。
2. [ユーザーとして実行する] フィールドで、コマンドを実行するクライアント上のユーザーのユーザー名を入力します。[グループとして実行する] フィールドのグループ名を使用して、コマンドを実行するグループを指定できます。
3. オプショナルタイムアウト] フィールドに、コマンドのタイムアウト期間を秒単位で入力します。この期間内にコマンドが実行されない場合、このコマンドは実行されません。

4. [コマンドラベル] フィールドに、コマンドの名前を入力します。
5. [スクリプト] フィールドに、実行するコマンドまたはスクリプトを入力します。
6. コマンドを実行する日時を選択するか、リモートコマンドを動作チェーンに追加します。
7. リモートコマンドをスケジュールするには、をクリックします。

動作チェーンの詳細については、[Reference > Schedule](#)を参照してください。

プロシージャ: Saltクライアントでのリモートコマンドの実行

1. [Salt > リモートコマンド](#)に移動します。
2. 最初のフィールドで、@記号の前に、発行するコマンドを入力します。
3. 2番目のフィールドで、@記号の後に、コマンドを発行するクライアントを入力します。個々のクライアントの minion-id を入力することも、ワイルドカードを使用してクライアントの範囲をターゲットにすることもできます。
4. をクリックして、ターゲットにしたクライアントを確認し、正しい詳細情報を使用していることを確認します。
5. をクリックして、ターゲットクライアントにコマンドを発行します。

Chapter 2. 認証方法

Uyuniは、いくつかの異なる認証方法をサポートしています。このセクションでは、pluggable authentication modules (PAM)およびシングルサインオン(SSO)について説明します。

2.1. シングルサインオン(SSO)による認証

Uyuniは、Security Assertion Markup Language (SAML) 2プロトコルを実装することで、シングルサインオン(SSO)をサポートしています。

シングルサインオンは、ユーザが1組の資格情報を使用して複数のアプリケーションにアクセスできるようにする認証プロセスです。 SAMLは、認証および許可データを交換するためのXMLベースの規格です。 SAML IDサービスプロバイダ(IdP)は、Uyuniなどのサービスプロバイダ(SP)に認証および許可サービスを提供します。 Uyuniは、シングルサインオンを有効にする必要がある3つのエンドポイントを公開します。

UyuniのSSOは以下をサポートします。

- ・ SSOを使用したログイン。
- ・ サービスプロバイダが開始したシングルログアウト(SLO)、およびIDサービスプロバイダのシングルログアウトサービス(SLS)を使用してログアウトする。
- ・ アサーションとnameIDの暗号化。
- ・ アサーションの署名。
- ・ AuthNRequest、LogoutRequest、およびLogoutResponderによるメッセージ署名。
- ・ アサーションコンシューマサービスエンドポイントの有効化。
- ・ シングルログアウトサービスエンドポイントの有効化。
- ・ SPメタデータ(署名可能)の発行。

UyuniのSSOは以下をサポートしません。

- ・ IDサービスプロバイダ(IdP)の製品の選択と実装。
- ・ 他の製品のSAMLサポート(各製品のドキュメントで確認してください)。

SSOの実装例については、[Administration > Auth-methods-sso-example](#)を参照してください。



デフォルトの認証方法からシングルサインオンに変更する場合、新しいSSO資格情報はWeb UIにのみ適用されます。`mgr-sync`や`spacecmd`などのクライアントツールは、引き続きデフォルトの認証方式でのみ動作します。

2.1.1. 前提条件

開始する前に、これらのパラメータを使用して外部IDサービスプロバイダを設定しておく必要があります。手順については、IdPのドキュメントを確認してください。



IdPには、**uid**と呼ばれるIdPユーザドメインのユーザ名を含むSAML:Attributeが必要です。SAML:Attributeで渡された**uid**属性は、シングルサインオンを有効にする前にUyuniユーザベースで作成する必要があります。

以下のエンドポイントが必要です。

- アサーションコンシューマサービス(ACS): SAMLメッセージを受け入れてサービスプロバイダへのセッションを確立するエンドポイント。
UyuniのACSのエンドポイントは`https://server.example.com/rhn/manager/sso/acs`です
- シングルログアウトサービス(SLS): IdPからログアウト要求を開始するエンドポイント。 UyuniのSLSのエンドポイントは`https://server.example.com/rhn/manager/sso/sls`です
- メタデータ: SAMLのUyuniメタデータを取得するエンドポイント。 Uyuniのメタデータのエンドポイントは`https://server.example.com/rhn/manager/sso/metadata`です

ユーザ **orgadmin** を使用したIdPによる認証が成功した後で、**orgadmin** ユーザがUyuniに存在する場合は、**orgadmin** ユーザとしてUyuniにログインします。

2.1.2. SSOの有効化



SSOの使用は、他のタイプの認証と相互に排他的であり、有効か無効のいずれかです。SSOはデフォルトで無効になっています。

プロシージャ: SSOの有効化

- ユーザがまだUyuniに存在しない場合は、まず作成してください。
- `/etc/rhn/rhn.conf`を編集して、次の行をファイルの最後に追加します。

```
java.sso = true
```

- `/usr/share/rhn/config-defaults/rhn_java_sso.conf`で、カスタマイズするパラメータを見つけます。カスタマイズするパラメータを`/etc/rhn/rhn.conf`に挿入し、それらのパラメータの前に**java.sso**を付けています。たとえば、`/usr/share/rhn/config-defaults/rhn_java_sso.conf`で以下を見つけます。

```
onelogin.saml2.sp.assertion_consumer_service.url = https://YOUR-  
PRODUCT-HOSTNAME-OR-IP/rhn/manager/sso/acs
```

カスタマイズするには、オプション名の前に**java.sso**を付けて、対応するオプションを`/etc/rhn/rhn.conf`に作成します。

```
java.sso.onelogin.saml2.sp.assertion_consumer_service.url =  
https://YOUR-PRODUCT-HOSTNAME-OR-IP/rhn/manager/sso/acs
```

変更する必要があるすべての出現を見つけるには、プレースホルダ **YOUR-PRODUCT** および **YOUR-IDP-ENTITY** をファイル内で検索します。すべてのパラメータには、その意味についての簡単な説明が含まれています。

- spacewalkサービスを再起動して変更を取得します。

```
spacewalk-service restart
```

UyuniのURLにアクセスすると、認証を要求されたSSO用IdPにリダイレクトされます。認証に成功すると、Uyuni Web UIにリダイレクトされ、認証されたユーザとしてログインします。SSOを使用したログインで問題が発生した場合は、Uyuniのログで詳細情報を確認してください。

2.2. PAMによる認証

Uyuniは、pluggable authentication modules (PAM)を使用したネットワークベースの認証システムをサポートしています。PAMは、Uyuniを集中認証メカニズムと統合できるようにする一連のライブラリであり、複数のパスワードを覚える必要がなくなります。Uyuniは、LDAP、Kerberos、およびPAMを使用するその他のネットワークベースの認証システムをサポートしています。

プロシージャ: PAMの有効化

- /etc/pam.d/susemanager** にPAMサービスファイルを作成します。ファイル名は小文字で指定し、**tomcat** ユーザが読み取り可能である必要があります。このファイルは、Uyuniが正しいPAM設定ファイルをロードするために使用されます。

```
#%PAM-1.0
auth    include      common-auth
account include    common-account
password include   common-password
session include    common-session
```

リスト 1. Uyuniサーバのコマンドプロンプトで、rootとして **sss** PAMモジュールを追加します。

```
pam-config -a --sss
```

このコマンドはモジュールを **/etc/pam.d/common-auth** 設定ファイルに追加します。このファイルを直接編集することはお勧めしません。

- 次の行を **/etc/rhn/rhn.conf** に追加してサービスファイルの使用を強制します。

```
pam_auth_service = susemanager
```

この例では、PAMサービスファイルは **susemanager** と呼ばれます。

3. 設定の変更後、Uyuniサービスを再起動します。
4. Uyuni Web UIで、menu:ユーザ [ユーザの作成]に移動し、新しいユーザまたは既存のユーザがPAMで認証されるようにします。
5. [**Pluggable Authentication Modules (PAM)**] チェックボックスをオンにします。 パスワードとパスワードの確認フィールドの下にあります。



Uyuni Web UIのパスワードを変更すると、Uyuniサーバのローカルパスワードのみが変更されます。 PAMがそのユーザに対して有効になっている場合、ローカルパスワードはまったく使用されない可能性があります。 たとえば、先に記載した例では、Kerberosパスワードは変更されません。 ネットワークサービスのパスワード変更メカニズムを使用して、これらのユーザのパスワードを変更します。

システム全体の認証を設定するには、YaSTを使用できます。 `yast2-auth-client` パッケージをインストールする必要があります。

PAMの設定の詳細については、SUSE Linux Enterprise Serverセキュリティガイドに、他のネットワークベースの認証方法でも機能する一般的な例が含まれています。 また、Active Directoryサービスを設定する方法についても説明します。 詳細については、<https://documentation.suse.com/sles/15-SP3/html/SLES-all/part-auth.html>を参照してください。

Chapter 3. バックアップと復元

Uyuniのインストールを定期的にバックアップして、データの損失を防ぎます。 Uyuniは、インストールされているプログラムと設定だけでなくデータベースにも依存しているため、インストールのすべてのコンポーネントをバックアップすることが重要です。 この章には、バックアップする必要のあるファイルに関する情報が含まれており、データベースバックアップを管理するための **smdba** ツールについて説明しています。また、システム障害が発生した場合のバックアップからの復元に関する情報も含まれています。



使用するバックアップ方法にかかわらず、現在のインストールで使用している容量の3倍以上の空き容量が必要です。容量が不足するとバックアップが失敗する可能性があるため、頻繁に確認してください。

3.1. Uyuniのバックアップ

Uyuniのインストールをバックアップする最も包括的な方法は、関連するファイルとディレクトリをバックアップすることです。これにより、バックアップの管理にかかる時間を節約でき、障害発生時に再インストールと再同期をより速く実行できます。ただし、この方法ではディスク領域がかなり必要になるため、バックアップの実行に時間がかかることがあります。



必要なファイルとディレクトリのみをバックアップする場合は、次のリストを使用します。このプロセスをよりシンプルで包括的なものにするには、ここで指定したディレクトリだけでなく **/etc** と **/root** ディレクトリ全体をバックアップすることをお勧めします。一部のファイルは、関連するSUSE Manager機能を実際に使用している場合にのみ存在します。

- **/etc/cobbler/**
- **/etc/dhcp.conf**
- **/etc/fstab** および必要なISOマウントポイント。

UUIDが変更されている場合は、それに応じて、**fstab** エントリを更新していることを確認します。

- **/etc/rhn/**
- **/etc/salt**
- **/etc/sudoers**
- **/etc/sysconfig/rhn/**
- **/root/.gnupg/**
- **/root/.ssh**

このファイルはSSHトンネルまたはSSH **push** を使用している場合に存在します。また **id-susemanager** キーのコピーを保存しておく必要があります。

- **/root/ssl-build/**

- `/srv/formula_metadata`
- `/srv/pillar`
- `/srv/salt`
- `/srv/susemanager`
- `/srv/tftpboot/`
- `/srv/www/cobbler`
- `/srv/www/htdocs/pub/`
- `/srv/www/os-images`
- `/var/cache/rhn`
- `/var/cache/salt`
- `/var/lib/cobbler/`
- `/var/lib/cobbler/templates/` (バージョン4.0より前は`/var/lib/rhn/kickstarts/`です)
- `/var/lib/Kiwi`
- `/var/lib/rhn/`
- `/var/run/pgsql/`
- `/var/lib/salt/`
- `/var/run/salt/`
- `/var/spacewalk/`
- スクリプト、KickstartまたはAutoYaSTプロファイル、カスタムRPMなどのカスタムデータを含むディレクトリ。



データベースをバックアップする必要があります。バックアップは `smdba` ツールで実行できます。

プロシージャ: 手動バックアップからの復元

1. Uyuniサーバを再インストールします。 詳細については、[Installation-and-upgrade > Install-server-unified](#)を参照してください。
2. Uyuniサーバを `yast2 susemanager_setup`を使用して設定します。 詳細については、[Installation-and-upgrade > Server-setup](#)を参照してください。
3. UyuniWeb UIを使用するか、コマンドプロンプトで `mgr-sync`ツールを使用して、Uyuniリポジトリを再同期します。 製品を再登録するか、登録およびSSL証明書生成セクションをスキップするかを選択できます。
4. `/root/ssl-build/rhn-org-htpd-ssl-key-pair-MACHINE_NAME-VER-REL.noarch.rpm` パッケージを再インストールします。
5. 次回 `rhn-search` サービスを開始するときに検索インデックスの再作成をスケジュールします。このコマンドはデバッグメッセージのみを生成し、エラーメッセージは生成しません。

```
rhn-search cleanindex
```

6. `/var/spacewalk/packages/` を復元する必要があるかどうかを確認します。`/var/spacewalk/packages/` がバックアップになかった場合は、復元する必要があります。ソースリポジトリが使用できる場合は、完全なチャンネル同期を使用して `/var/spacewalk/packages/` を復元できます。

```
mgr-sync refresh --refresh-channels
```

3.2. smdbaによるデータベースの管理

ローカルPostgreSQLデータベースを管理するには、`smdba` ツールを使用します。データベースのバックアップと復元、およびバックアップの管理を行うことができます。また、データベースのステータスを確認したり、再起動などの管理タスクを実行したりするために使用できます。`smdba` ツールの詳細については、[\[reference:cli-smdba\]](#)を参照してください。

`smdba` ツールは、ローカルPostgreSQLデータベースでのみ動作し、リモートアクセスされたデータベースやOracleデータベースでは動作しません。



`smdba` ツールにはシステムの変更を実行するために `sudo` アクセスが必要です。開始する前に次の行の `/etc/sudoers` ファイルをチェックして、`admin` ユーザに対して `sudo` アクセスを有効にしていることを確認してください。

```
admin    ALL=(postgres) /usr/bin/smdba
```

次のコマンドを使用して、データベースのランタイムステータスを確認します。

```
smdba db-status
```

このコマンドは、次のように、`online` または `offline` を返します。

```
Checking database core...      online
```

データベースの起動と停止は次のコマンドを使用して実行できます。

```
smdba db-start
```

および

```
smdba db-stop
```

3.3. smdbaによるデータベースのバックアップ

smdba ツールは継続的なアーカイブバックアップを実行します。このバックアップ方法では、現在のセッション中にデータベースに行われたすべての変更のログと、一連の従来のバックアップファイルが結合されます。クラッシュが発生した場合は、最初にディスク上の最新のバックアップファイルからデータベースの状態が復元され、次に現在のセッションのログが正確に再生されて、データベースが現在の状態に戻されます。データベースを実行している状態で **smdba** を使用した継続的なアーカイブバックアップが実行されるため、ダウンタイムは必要ありません。

このバックアップ方法は安定していて、通常は整合性のあるスナップショットを作成しますが、多くのストレージ容量を占有する可能性があります。バックアップに使用できる現在のデータベースサイズの3倍以上の容量があることを確認します。`/var/lib/pgsql/` に移動し、`df -h` を実行することで、現在のデータベースサイズを確認できます。

また、**smdba** ツールは、アーカイブを管理し、最新のバックアップと、ログの現在のアーカイブのみを保持します。ログファイルの最大ファイルサイズはわずか16MBであるため、ファイルがこのサイズに達すると新しいログファイルが作成されます。新しいバックアップを作成するたびに、以前のバックアップがページされてディスク領域が解放されます。`cron` を使用して **smdba** のバックアップをスケジュールし、ストレージが効果的に管理されるようにして、障害が発生した場合に備えて常にバックアップを準備しておくことをお勧めします。

3.3.1. データベースの手動バックアップの実行

smdba ツールはコマンドラインから直接実行できます。インストール直後、または設定に重大な変更を加えた場合は、データベースの手動バックアップを実行することをお勧めします。



smdba が初めて実行される場合、またはバックアップの場所を変更した場合は、アーカイブを実行する前にデータベースを再起動する必要があります。この結果、わずかなダウンタイムが発生します。通常のデータベースバックアップでは、ダウンタイムは必要ありません。

プロシージャ: データベースの手動バックアップの実行

- バックアップ用の永続的なストレージ容量を割り当てます。この例では、`/var/spacewalk/` にあるディレクトリを使用しています。これはバックアップの永続的なターゲットになるため、常にサーバからアクセスできるようにしてください。
- バックアップの場所で、バックアップのディレクトリを作成します。

rootとして:

```
install -d -o postgres -g postgres -m 700 /var/spacewalk/db-backup
```

3. バックアップの場所に適切な許可が設定されていることを確認します。

```
chown postgres:postgres /var/spacewalk/db-backup
```

4. 最初に、バックアップを作成するには、**enable** オプションセットを指定して、**smdba backup-hot** コマンドを実行します。これにより、指定されたディレクトリにバックアップが作成されます。必要に応じて、データベースを再起動します。

```
smdba backup-hot --enable=on --backup-dir=/var/spacewalk/db-backup
```

このコマンドはデバッグメッセージを生成し、次の出力で正常に終了します。

```
INFO: Finished
```

5. バックアップファイルが **/var/spacewalk/db-backup** ディレクトリに存在することを確認し、バックアップが成功したことを確認してください。

3.3.2. 自動バックアップのスケジュール

smdba を使用してデータベースバックアップを実行するためにシステムをシャットダウンする必要はありません。ただし、大規模な操作であるため、バックアップの実行中にデータベースのパフォーマンスが低下する可能性があります。中断を最小限に抑えるために、定期的なデータベースバックアップを低トラフィック期間にスケジュールすることをお勧めします。



バックアップに使用できる現在のデータベースサイズの3倍以上の容量があることを確認します。 **/var/lib/pgsql/** に移動し、**df -h** を実行することで、現在のデータベースサイズを確認できます。

プロシージャ: 自動バックアップのスケジュール

1. バックアップのディレクトリを作成し、適切な許可を(rootとして)設定します。

```
install -m 700 -o postgres -g postgres /var/spacewalk/db-backup
```

2. **/etc/cron.d/db-backup-mgr** を開くか、存在しない場合は作成し、次の行を追加して、cronジョブを作成します。

```
0 2 * * * root /usr/bin/smdba backup-hot --enable=on --backup
-dir=/var/spacewalk/db-backup
```

3. バックアップディレクトリを定期的に確認し、バックアップが期待通りに機能していることを確認しま

す。

3.4. バックアップからの復元

smdba ツールを使用すると、障害が発生した場合にはバックアップから復元できます。

プロシージャ: バックアップからの復元

1. データベースをシャットダウンします。

```
smdba db-stop
```

2. 復元プロセスを開始して、完了するまで待機します。

```
smdba backup-restore start
```

3. データベースを再起動します。

```
smdba db-start
```

4. RPMとデータベース間に違いがあるかどうかを確認します。

```
spacewalk-data-fsck
```

3.5. アーカイブログ設定

アーカイブログを使用すると、データベース管理ツール **smdba** でホットバックアップを実行できます。Uyuniに埋め込みデータベースがある場合、デフォルトでアーカイブログは有効になっています。

PostgreSQLは限られた数のアーカイブ ログを維持します。 デフォルト設定を使用すると、サイズが16MiBの約64個のファイルが保存されます。

ユーザを作成し、チャンネルを同期します。

- SLES12-SP2-Pool-x86_64
- SLES12-SP2-Updates-x86_64
- SLE-Manager-Tools12-Pool-x86_64-SP2
- SLE-Manager-Tools12-Updates-x86_64-SP2

PostgreSQLは、追加で約1GBのデータを生成します。 したがって、バックアップ戦略について考え、定期的にバックアップを作成することが重要です。

アーカイブログは `/var/lib/pgsql/data/pg_xlog/` (postgresql)に保存されます。

3.6. 占有されているデータベーススペースの概要を取得する

データベース管理者は、サブコマンド `space-overview` を使用して、占有されていたテーブルスペースに関するレポートを取得します。例:

```
smdba space-overview
```

出力:

```
SUSE Manager Database Control. Version 1.5.2
Copyright (c) 2012 by SUSE Linux Products GmbH

Tablespace | Size (Mb) | Avail (Mb) | Use %
-----+-----+-----+-----
postgres   | 7          | 49168      | 0.013
susemanager | 776        | 48399      | 1.602
```

`smdba` コマンドはPostgreSQLで使用できます。より詳細なレポートの場合は、`space-tables` サブコマンドを使用します。 テーブルとそのサイズを一覧にします。例:

```
smdba space-tables
```

出力:

```
SUSE Manager Database Control. Version 1.5.2
Copyright (c) 2012 by SUSE Linux Products GmbH

Table           | Size
-----+-----
public.all_primary_keys | 0 bytes
public.all_tab_columns | 0 bytes
public.allserverkeywordsincereboot | 0 bytes
public.dblink_pkey_results | 0 bytes
public.dual | 8192 bytes
public.evr_t | 0 bytes
public.log | 32 kB
...
```

3.7. データベースの移動

データベースを別の場所に移動できます。たとえば、データベースのストレージ容量が少なくなっている場合です。

プロシージャ: データベースの移動

1. Uyuniのデフォルトのストレージ場所は `/var/lib/pgsql/` です。たとえば、`/storage/postgres/` に移動する場合は、次のように進みます。
2. コマンドプロンプトで、rootとして、実行中のデータベースを停止します。

```
rcpostgresql stop
```

実行中のspacewalkサービスをシャットダウンします。

```
spacewalk-service stop
```

3. `-a, --archive` オプションを使用して、現在の作業ディレクトリ構造を `cp` でコピーします。例:

```
cp --archive /var/lib/pgsql/ /storage/postgres/
```

このコマンドは、`/var/lib/pgsql/` の内容を `/storage/postgres/pgsql/` にコピーします。



`/var/lib/pgsql` ディレクトリの内容は同じままにしておく必要があります。同じままでないと、Uyuniデータベースが正常に動作しない可能性があります。また、十分な空きディスク容量があることも確認する必要があります。

4. 新しいデータベースディレクトリをマウントします。

```
mount /storage/postgres/pgsql
```

5. 新しいディレクトリに移動し、次のコマンドを実行して、所有権が`root:root`ではなく、`postgres:postgres`であることを確認します。

```
cd /storage/postgres/pgsql/
ls -l
```

出力:

```
total 8
drwxr-x--- 4 postgres postgres 47 Jun 2 14:35 ./
```

6. `etc/fstab` を編集して、サーバfstabに新しいデータベースのマウント場所を追加します。

7. 次のコマンドを使用して、データベースを起動します。

```
rcpostgresql start
```

8. spacewalkサービスを開始します。

```
spacewalk-service start
```

3.8. クラッシュしたルートパーティションからの回復

ルートパーティションがクラッシュした場合は、追加のステップでUyuniサーバを再起動できます。 このセクションでは、`/var/lib/pgsql` および `/var/spacewalk/` にマウントされた、データベースとチャンネルに別のパーティションを使用してサーバをセットアップしていることを前提としています。



システムの新規インストール後、ほとんどのユーザとグループは異なるIDを取得します。ほとんどのバックアップシステムでは、IDの代わりに名前が保存され、正しい所有権と許可でファイルが復元されます。ただし、既存のパーティションをマウントする場合は、所有権を新しいシステムに合わせる必要があります。

プロシージャ: クラッシュしたルートパーティションからの回復

1. Uyuniをインストールします。`/var/spacewalk` および `/var/lib/pgsql` パーティションはマウントしないでください。インストールが完了するまで待ってから、次のステップに進みます。

2. データベースをシャットダウンします。

```
rcpostgresql stop
```

3. サービスをシャットダウンします。

```
spacewalk-service stop
```

4. `/var/spacewalk` および `/var/lib/pgsql` パーティションをマウントします。

5. [Uyuniのバックアップ](#)に一覧にされているディレクトリを復元します。

6. データベースを起動します。

```
rcpostgresql start
```

7. spacewalkサービスを開始します。

```
spacewalk-service start
```

Uyuniは、データベースや同期されたチャンネルを失うことなく正常に動作するはずです。

3.9. データベース接続情報

/etc/rhn/rhn.conf で次の変数を追加または編集して、Uyuniデータベースに接続する情報を設定できます。

```
db_backend = postgresql
db_user = susemanager
db_password = susemanager
db_name = susemanager
db_host = localhost
db_port = 5432
db_ssl_enabled =
```

Chapter 4. コンテンツのステージング

ステージングは、クライアントがインストール前にパッケージを事前にダウンロードするために使用されます。これにより、パッケージのインストールをスケジュールされたらすぐに開始できるため、メンテナンスウィンドウに必要な時間を短縮できます。

4.1. コン텐ツステージングの有効化

組織全体のコンテンツステージングを管理できます。Uyuni Web UIで、[管理 > 組織](#)に移動して、使用可能な組織のリストを表示する前に組織名をクリックし、[ステージング] ボタンをクリックして、この組織のクライアントがパッケージデータをステージングできるようにします。



組織を作成および管理するには、Uyuni管理者としてログインする必要があります。

`/etc/sysconfig/rhn/up2date` を編集し、次の行を追加または編集して、コマンドプロンプトでステージングを有効にすることもできます。

```
stagingContent=1
stagingContentWindow=24
```

`stagingContentWindow` パラメータは時間単位で表される時間値で、ダウンロードの開始時間を決定します。これは、スケジュールされたインストールまたは更新時間までの時間数です。この例では、インストール時間の24時間前にコンテンツがダウンロードされます。ダウンロードの開始時間は、システムで選択した連絡方法によって異なります。割り当てられた連絡方法は、次の `mgr_check` が実行される時間を設定します。

次回、アクションがスケジュールされると、パッケージは自動的にダウンロードされますが、インストールはされません。スケジュールされた時間に、ステージングされたパッケージがインストールされます。

4.2. コンテンツステージングの設定

コンテンツステージングを設定するために使用される次の2つのパラメータがあります。

- ・ `salt_content_staging_advance` は、コンテンツステージングウィンドウが開くまでの時間です(時間単位)。これは、インストールが開始されるまでの時間数で、この時間に達するとパッケージのダウンロードを開始できます。
- ・ `salt_content_staging_window` は、コンテンツステージングウィンドウの期間(時間単位)です。これは、インストールが開始されるまでにクライアントがパッケージをステージングする必要がある時間数です。

たとえば `salt_content_staging_advance` が6時間に設定されていて、`salt_content_staging_window` が2時間に設定されている場合、ステージングウィンドウはインストール時間の6時間前に開き、2時間開いたままになります。インストールが開始されるまで、残りの4時間以内にパッケージはダウンロードされません。

`salt_content_staging_advance` および `salt_content_staging_window` の両方に同じ値を設定する場合、インストールが開始されるまでパッケージをダウンロードできます。

`/usr/share/rhn/config-defaults/rhn_java.conf` で、コンテンツステージングパラメータを設定します。

デフォルト値:

- `salt_content_staging_advance: 8 hours`
- `salt_content_staging_window: 8 hours`



これらのパラメータを正しく機能させるには、コンテンツステージングを有効にする必要があります。

Chapter 5. チャンネル管理

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。

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グループ化され、ベースチャンネルはオペレーティングシステムのタイプ、バージョン、およびアーキテクチャ別にグループ化され、子チャンネルは関連するベースチャンネルと互換性があります。 クライアントがベースチャンネルに割り当てられている場合、そのシステムでは関連する子チャンネルのみをインストールできます。 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ます。

ソフトウェアチャンネルは、リポジトリを使用してパッケージを提供します。 チャンネルはUyuniのリポジトリをミラーリングし、パッケージ名やその他のデータはUyuniデータベースに保存されます。 1つのチャンネルに関連付けられたリポジトリはいくつでも持つことができます。 これらのリポジトリのソフトウェアは、クライアントを適切なチャンネルにサブスクライブすることでクライアントにインストールできます。

クライアントはベースチャンネルにのみ割り当てることができます。 その後、クライアントは、そのベースチャンネルとその子チャンネルのいずれかに関連付けられたリポジトリからパッケージをインストールまたは更新できます。

Uyuniには、Uyuniを実行するために必要なすべてのものを提供する、多数のベンダチャンネルが用意されています。 Uyuniの管理者とチャンネル管理者には、チャンネル管理権限があり、これにより、独自のカスタムチャンネルを作成して管理することができます。 環境で独自のパッケージを使用する場合は、カスタムチャンネルを作成できます。 カスタムチャンネルはベースチャンネルとして使用することも、ベンダベースチャンネルに関連付けることもできます。

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

5.1. チャンネル管理

デフォルトでは、すべてのユーザがシステムにチャンネルをサブスクライブできます。 Web UIを使用してチャンネルに制限を実装できます。

プロシージャ: チャンネルへのサブスクライバアクセスの制限

1. Uyuni Web UIで、**ソフトウェア** > **チャンネルリスト**に移動して、編集するチャンネルを選択します。
2. [ユーザー毎のサブスクリプション制限] セクションを見つけて、[この組織内のみがこのチャンネルにサブスクリー~~ト~~**[更新]**新巻戻す]をオーバーレイを保存します。
3. [サブスクリバ] タブに移動して、必要に応じてユーザを選択または選択解除します。

5.2. チャンネルの削除

コマンドプロンプトからベンダソフトウェアチャンネルを削除できます。

カスタムチャンネルの削除については、Administration > Custom-channelsを参照してください。

プロシージャ: ベンダチャンネルの削除

- Uyuniサーバのコマンドプロンプトで、rootとして、使用できるベンダチャンネルを一覧にし、削除するチャンネルをメモします。

```
mgr-sync list チャネル
```

- チャンネルを削除します。

```
spacewalk-remove-channel -c <channel-name>
```

- 同期してチャンネルを削除します。

```
mgr-sync sync channel <channel-name>
```

5.3. カスタムチャンネル

カスタムチャンネルを使用すると、クライアントを更新するために使用できる、独自のソフトウェアパッケージとリポジトリを作成できます。また、環境内でサードパーティベンダが提供するソフトウェアを使用することもできます。

このセクションでは、カスタムチャンネルを作成、管理、および削除する方法について詳細に説明します。カスタムチャンネルを作成および管理できるようにするには、管理者権限が必要です。

5.3.1. カスタムチャンネルおよびリポジトリの作成

カスタムチャンネルを作成する前に、関連付けるベースチャンネルと、コンテンツに使用するリポジトリを決定します。

クライアントシステムにインストールする必要があるカスタムソフトウェアパッケージがある場合は、カスタム子チャンネルを作成して管理できます。システムにチャンネルを割り当てる前に、Uyuni Web UIでチャンネルを作成し、パッケージのリポジトリを作成する必要があります。



クライアントシステムと互換性のないパッケージを含む子チャンネルを作成しないでください。

ベンダによって提供されたパッケージを使用する場合は、ベースチャンネルとしてベンダチャンネルを選択できます。または [なし] を選択して、カスタムチャンネルをベースチャンネルにします。

プロシージャ: カスタムチャンネルの作成

- Uyuni Web UIで、[ソフトウェア](#) > [管理](#) > [チャンネル](#)に移動して、[\[Create Channel\]](#) (チャンネルの作成) をクリックします。
- [[ソフトウェアチャンネルの作成](#)] ページで、チャンネルに名前([My Tools SLES 15 SP1 x86_64](#))、およびラベル(たとえば、[my-tools-sles15sp1-x86_64](#))を付けます。ラベルにはスペースまたは

は大文字を含めないでください。

3. [親チャンネルドロップダウン]で、関連するベースチャンネル(たとえば、SLE-Product-SLES15-SP1-Pool for x86_64)を選択します。パッケージに互換性のある親チャンネルを選択していることを確認します。
4. [アーキテクチャ]ドロップダウンで、適切なハードウェアアーキテクチャ(たとえば、x86_64)を選択します。
5. ご使用の環境に応じて、連絡先の詳細、チャンネルアクセス制御、およびGPGフィールドに追加情報を入力します。
6. [チャンネルの作成]をクリックします。

カスタムチャンネルでは、追加のセキュリティ設定が必要になる場合があります。多くのサードパーティベンダーは、GPGを使用してパッケージをセキュリティ保護しています。カスタムチャンネルでGPGで保護されたパッケージを使用する場合は、メタデータの署名に使用されたGPGキーを信頼する必要があります。次に、[メタデータは署名されていますか？] チェックボックスをオンにして、パッケージメタデータのGPGキーと照合します。

リモートチャンネルおよびリポジトリがGPGキーで署名されている場合、これらのGPGキーをインポートして信頼することができます。たとえば、Uyuniサーバのコマンドラインから `spacewalk-repo-sync` を実行します。

```
/usr/bin/spacewalk-repo-sync -c <channellabelname> -t yum
```

基盤となる `zypper` コールは、利用可能な場合はキーをインポートします。Web UIはこの機能を提供していません。

これはミラーリングするリポジトリが特別な方法で設定され、署名の横のリポジトリに「キー」が提供されている場合にのみ機能します。これは、Open Build Service (OBS)によって生成されたすべてのリポジトリに該当します。その他のリポジトリについては、特別な準備ステップが必要です(以下を参照)。



デフォルトでは、新しいチャンネルを[GPGチェックの有効化] フィールドがオフになっています。チャンネルにカスタムパッケージとアプリケーションを追加する場合は、このフィールドをオフにして、署名されていないパッケージをインストールできるようにしてください。パッケージが信頼されていないソースからのものである場合、GPGチェックを無効にするとセキュリティリスクが生じます。



従来のRed Hat Enterprise Linux 7またはSLES Expanded Support 7クライアントを登録している場合、署名されていないパッケージでエラーが発生する場合があります。 詳細については、Administration > Troubleshootingを参照してください。

リポジトリが有効なソフトウェアリポジトリである場合は、Web UIを使用してUyuniに追加することのみできます。必要なリポジトリメタデータが使用できることを事前に確認してください。これに関しては、`createrepo` や `reprepro` などのツールが便利です。`mgr-push` はリポジトリを作成せずに単一のRPMをチャンネルにプッシュするのに役立ちます。 詳細については、`createrepo_c` および `reprepro` のマニュアルペ

ページを参照してください。

プロシージャ: ソフトウェアリポジトリの追加

1. Uyuni Web UIで、**ソフトウェア > 管理 > リポジトリ**に移動し、**[リポジトリの作成]**をクリックします。
2. **[リポジトリの作成]**ページで、リポジトリにラベル(たとえば、`my-tools-sles15sp1-x86_64-repo`)を付けます。
3. **[リポジトイサル]**に、**repodata** ファイル(たとえば、`file:///opt/mytools/`)を含むディレクトリへのパスを指定します。 このフィールドでは、任意の有効なアドレス指定プロトコルを使用できます。
4. **[メタデータは署名されていますか？]** チェックボックスをオフにします。
5. オプション: リポジトリでクライアント証明書認証が必要な場合は、SSLフィールドに入力します。
6. **[リポジトリの作成]**をクリックします。

先に示したプロシージャは、ミラーリングするリポジトリが署名の横にあるリポジトリに「キー」を提供する場合にのみ機能します。 これはOBSによって生成されたすべてのリポジトリに該当しますが、通常、OBSによって提供されていないオペレーティングシステムの他のリポジトリには該当しません。

使用するリポジトリにGPGキーがない場合は、自分でGPGキーを指定し、GPGキーをキーリングに手動でインポートできます。 **gpg** コマンドラインツールを使用して `/var/lib/spacewalk/gpgdir` キーリングにキーをインポートすると、永続的に保存されます。 chroot環境が消去されても、キーは保持されます。

プロシージャ: GPGキー付きで、ソフトウェアリポジトリの作成

1. キーをpubキーリングにインポートするgpgコマンドは次のとおりです。

```
gpg --keyring /var/lib/spacewalk/gpgdir --import /path/to/gpg.key
gpg --edit-key <keyid>
```



uyuni_suite、**uyuni_component**、および **uyuni_arch** のクエリパラメータを使用して、Debianの非フラットリポジトリを追加します。

uyuni_suite

必須です。 Debianのドキュメントでは、これは **distribution** とも呼ばれています。 このパラメータでは、aptソースを指定します。 このパラメータを指定しない場合は、元のアプローチが使用されます。 パラメータの末尾が / の場合、リポジトリはフラットとして識別されます。

uyuni_component

オプションです。 このパラメータは1つのコンポーネントのみを指定できます。 コンポーネントを一覧にすることはできません。 aptソースエントリでは複数のコンポーネントを指定できますが、Uyuniでは指定できません。 代わりに、コンポーネントごとに個別のリポジトリを作成する必要があります。

uyuni_arch

オプションです。省略すると、データベースからチャンネルのSQLクエリを使用してアーキテクチャ名が計算されます。チャンネルのアーキテクチャと一致しない場合は、明示的に **uyuni_arch** を指定します(アーキテクチャの命名があいまいな場合があります)。

ここにいくつかの例があります。

表 1. Debian非フラットリポジトリマッピング

Type	Source line / URL
apt source line	<code>deb https://pkg.jenkins.io/ debian-stable binary/</code>
URL mapping	<code>https://pkg.jenkins.io/ debian-stable?</code> <code>uyuni_suite=binary/</code>
apt source line	<code>deb https://deb.debian.org/ debian/ dists stable main</code>
URL mapping	<code>https://deb.debian.org/ debian/ dists?</code> <code>uyuni_suite=stable& uyuni_component=main</code>

Debianリポジトリ定義フォーマットに関する背景情報はこちらをご覧ください。
この情報は、<https://wiki.debian.org/DebianRepository/Format#Overview>に基づいています。

リポジトリ定義フォーマットは次のとおりです。

```
deb uri suite [component1] [component2] [...]
```

例:

```
deb https://deb.debian.org/debian/dists stable main
```

または

```
deb https://pkg.jenkins.io/debian-stable binary/
```

スイートとコンポーネントのペアごとに、仕様ではベースURL + **suite component**に基づいて計算される個別のURLを定義しています。

プロシージャ: リポジトリをチャンネルに割り当てる

1. ソフトウェア > 管理 > チャンネルに移動して、新たに作成されたカスタムチャンネルの名前をクリックし、[リポジトリ] タブに移動して、新しいリポジトリをカスタムチャンネルに割り当てます。

2. チャンネルに割り当てるリポジトリがオンになっていることを確認し、[リポジトリの更新]をクリックします。
3. デフォルトでは、同期プロセスはすぐに開始されます。チャンネル同期の詳細については、以下を参照してください。

プロシージャ: アクティベーションキーへのカスタムチャンネルの追加

1. Uyuni Web UIで、**システム > アクティベーションキー**に移動して、カスタムチャンネルを追加するキーを選択します。
2. [詳細] タブの [子チャンネル] リストで、関連付けるチャンネルを選択に応じて、複数のチャンネルを選択できます。
3. [アクティベーションキーの更新] をクリックします。

5.3.2. カスタムチャンネル同期

重要な更新を見逃さないようにするために、SUSEでは、カスタムチャンネルをリモートリポジトリの変更に合わせて最新の状態に保つことをお勧めします。

デフォルトでは、作成したすべてのカスタムチャンネルに対して同期が自動的に行われます。特に、次のことが発生します。

- ・ UIから、または `spacewalk-common-channels` を使用して、リポジトリをチャンネルに追加した後
- ・ 毎日のタスク `mgr-sync-refresh-default` の一部として、すべてのカスタムおよびベンダチャンネルを同期します。

このデフォルトの動作を無効にするには、`/etc/rhn/rhn.conf` に以下を設定します。

```
java.unify_custom_channel_management = 0
```

このプロパティをオフにすると、同期は自動的に実行されません。カスタムチャネルを最新の状態に保つには、次の操作を行う必要があります。

- ・ [同期] タブに移動し、[今すぐ同期] をクリックして手動で同期し、
- ・ [リポジトリ] タブから自動同期スケジュールを設定します。

プロセス開始時には、チャンネルの同期が終了したかどうかを確認するいくつかの方法があります。

- ・ UyuniのWeb UIで、**管理セットアップウィザード**に移動し、[製品] タブを選択します。このダイアログには、同期中の各製品の完了バーが表示されます。
- ・ 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。menu:[リポジトリ同期]タブに移動します。リポジトリ名の横に [同期状態] が表示されます。
- ・ コマンドプロンプトで同期ログファイルを確認します。

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

同期の進行中に各子チャンネルは独自のログを生成します。ベースチャンネルと子チャンネルのすべてのログファイルを確認して、同期が完了していることを確認する必要があります。

5.3.3. カスタムチャンネルへのパッケージとパッチの追加

既存のチャンネルからクローンを作成せずに新しいカスタムチャンネルを作成する場合、そのチャンネルにはパッケージやパッチは含まれません。Uyuni Web UIを使用して、必要なパッケージとパッチを追加できます。

カスタムチャンネルには、クローンまたはカスタムのパッケージまたはパッチのみを含めることができます。チャンネルの基本アーキテクチャに一致する必要があります。カスタムチャンネルに追加されたパッチは、チャンネルに存在するパッケージに適用する必要があります。

プロシージャ: カスタムチャンネルへのパッケージの追加

1. Uyuni Web UIで、**ソフトウェア > 管理 > チャンネル**に移動して、[パッケージ] タブに移動します。
2. オプション: [一覧表示/削除] タブに移動して、現在チャンネル内にあるすべてのパッケージを表示します。
3. [追加] タブに移動して、チャンネルに新しいパッケージを追加します。
4. パッケージを提供する親チャンネルを選択し、[パッケージの表示] をクリックして、リストに入力します。
5. カスタムチャンネルに追加するパッケージをオンにして、[パッケージの追加] をクリックします。
6. 選択に問題がなければ、[追加の確認] をクリックして、チャンネルにパッケージを追加します。
7. オプション: ソフトウェア > 管理 > チャンネルに移動し、パッケージ > 比較タブに移動して、現在のチャンネルのパッケージを別のチャンネルのパッケージと比較できます。2つのチャンネルを同じにするには、[違いをマージする] ボタンをクリックし、競合を解決します。

プロシージャ: カスタムチャンネルへのパッチの追加

1. Uyuni Web UIで、**ソフトウェア > 管理 > チャンネル**に移動して、[パッチ] タブに移動します。
2. オプション: [一覧表示/削除] タブに移動して、現在チャンネル内のすべてのパッチを表示します。
3. [追加] タブに移動し、追加するパッチの種類を選択して、チャンネルに新しいパッチを追加します。
4. パッチを提供する親チャンネルを選択し、[関連づけられたパッチの表示] をクリックして、リストに入力します。
5. カスタムチャンネルに追加するパッチをオンにして、[確認] をクリックします。
6. 選択に問題がなければ、[確認] をクリックして、チャンネルにパッチを追加します。

5.3.4. カスタムチャンネルの管理

Uyuni管理者とチャンネル管理者は任意のチャンネルを変更または削除できます。

チャンネルを変更または削除する権限を他のユーザに付与するには、[ソフトウェア > 管理 > チャンネル](#)に移動して、編集するチャンネルを選択しま [**マネージャ**] タブに移動して、権限を付与するユーザをオンにします。 [**[更新]**] をクリックして、変更を保存します。



一連のクライアントに割り当てられているチャンネルを削除すると、削除されたチャンネルに関連付けられているすべてのクライアントのチャンネル状態が直ちに更新されます。 これは、変更がリポジトリファイルに正確に反映されるようになります。

Web UIを使用してUyuniチャンネルを削除することはできません。 カスタムチャンネルのみを削除できます。

プロシージャ: カスタムチャンネルの削除

1. Uyuni Web UIで、[ソフトウェア > 管理 > チャンネル](#)に移動して、削除するチャンネルを選択します。
2. [**[ソフトウェア] チャンネルの削除**] をクリックします。
3. [**チャンネルの削除**] ページで、削除するチャンネルの詳細を確認し、[**システムのサブス**
ト] チェックボックスをオンにして、まだサブスクライブされている可能性のあるシステムからカスタムチャンネルを削除します。
4. [**チャンネルの削除**] をクリックします。

チャンネルを削除しても、削除されたチャンネルの一部であるパッケージは自動的に削除されません。 チャンネルが削除されたパッケージを更新することはできません。

Uyuni Web UIで、チャンネルに関連付けられていないパッケージを削除できます。 [ソフトウェア > 管理 > パッケージ](#)に移動して、削除するパッケージをオンにして、 [**[パッケージ] の削除**] をクリックします。

Chapter 6. コンテンツライフサイクル管理

コンテンツライフサイクル管理では運用クライアントを更新する前にパッケージをカスタマイズおよびテストできます。これは、限られたメンテナスウィンドウ中に更新を適用する必要がある場合に特に役立ちます。

コンテンツライフサイクル管理では、ソフトウェアチャンネルをソースとして選択し、必要に応じて環境に合わせて調整して、運用クライアントにインストールする前に徹底的にテストすることができます。

ベンダーチャンネルを直接変更することはできませんが、パッケージとカスタムパッチを追加または削除することで、それらのチャンネルのクローンを作成し、クローンを変更することができます。これらのクローンチャンネルをテストクライアントに割り当てて、クライアントが期待どおりに動作することを確認できます。その後、すべてのテストに合格すると、運用サーバにプロモートさせることができます。

これは、ソフトウェアチャンネルがライフサイクルで移動できる一連の環境を通じて実現されます。ほとんどの環境ライフサイクルには、少なくともテスト環境と運用環境が含まれていますが、必要な数の環境を持つことができます。

このセクションでは、基本的なコンテンツライフサイクル手順、および使用可能なフィルタについて説明します。より具体的な例については、Administration > Content-lifecycle-examplesを参照してください。

6.1. コンテンツライフサイクルプロジェクトの作成

コンテンツライフサイクルを設定するには、プロジェクトから開始する必要があります。プロジェクトでは、ソフトウェアチャンネルソース、パッケージの検索に使用されるフィルタ、およびビルド環境について定義します。

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

- Uyuni UIで、**コンテンツライフサイクル**に移動し、**[プロジェクトの作成]**をクリックします。
- [ラベル]** フィールドに、プロジェクトのラベルを入力。**[ラベル]** フィールドには、小文字、数字、ピリオド、ハイフン、およびアンダースコアのみを入力できます。
- [名前]** フィールドに、プロジェクトのわかりやすい名前を入力します。
- [作成]**ボタンをクリックして、プロジェクトを作成し、プロジェクトページに戻ります。
- [Attach/Detach Sources]** (ソースの割り当て/取り外し) をクリックします。
- [ソース]** ダイアログで、ソースタイプを選択し、プロジェクトのベースチャンネルを選択します。チャンネルが必須か推奨されるかに関する情報を含め、選択したベースチャンネルで使用可能な子チャンネルが表示されます。
- 必要な子チャンネルをオンにし、**[保存]**をクリックして、プロジェクトページに戻ります。選択したソフトウェアチャンネルが表示されているはずです。
- [フィルタの割り当て]**をクリックします。
- [フィルタ]** ダイアログで、プロジェクトに割り当てるフィルタを選択し、新しいフィルタを作成するには、**[新規]**をクリックします。

10. [環境] の [追加] をクリックします。
11. [環境ライフサイクル] ダイアログで、最初の環境に、名前、ラベル、および説明を付けて、[保存] をクリックします。ラベル] フィールドには、小文字、数字、ピリオド、ハイフン、およびアンダースコアのみを入力できます。
12. ライフサイクルのすべての環境が完了するまで、環境の作成を続行しません時に [前に挿入] フィールドで環境を選択することで、ライフサイクルの環境の順序を選択できます。

6.2. フィルタタイプ

Uyuniでは、プロジェクトの構築に使用するコンテンツを制御するために、さまざまなタイプのフィルタを作成できます。 フィルタを使用すると、ビルドに含めるパッケージまたはビルドから除外するパッケージを選択できます。 たとえば、すべてのカーネルパッケージを除外したり、一部のパッケージの特定のリリースのみを含めることができます。

サポートされているフィルタは次のとおりです。

- ・ パッケージのフィルタリング
 - by name (名前別)
 - by name, epoch, version, release, and architecture (名前、エポック、バージョン、リリース、およびアーキテクチャ別)
 - by provided name (指定された名前別)
- ・ パッチフィルタリング
 - by advisory name (アドバイザリ名別)
 - by advisory type (アドバイザリタイプ別)
 - by synopsis (概要別)
 - by keyword (キーワード別)
 - by date (日付別)
 - by affected package (影響を受けるパッケージ別)
- ・ モジュール
 - by stream (ストリーム別)



パッケージの依存は、コンテンツのフィルタリング中には解決されません。

フィルタで使用できるマッチャーは複数あります。 使用できるマッチャーは、選択したフィルタ タイプによって異なります。

使用できるマッチャーは次のとおりです。

- ・ 含む

- ・ 一致（正規表現の形式を取る必要があります）
- ・ 等しい
- ・ 新しい
- ・ 新しいか等しい
- ・ 古いか等しい
- ・ 古い
- ・ 新しいか等しい

6.2.1. フィルタの rule パラメータ

各フィルタには 許 可 または 拒 否 のいずれかに設定できる rule パラメータがあります。ルタは次のように処理されます。

- ・ パッケージまたはパッチが 拒 否 フィルタを満たす場合は、結果から除外されます。
- ・ パッケージまたはパッチが 許 可 フィルタを満たす場合は、結果に含まれます（拒 否 フィルタによって除外された場合でも）。

この動作は、一般的な 拒 否 フィルタを使用して多数のパッケージまたはパッチを除外し、特定の 許 可 フィルタで特定のパッケージまたはパッチを「チェリーピック」する場合に役立ちます。



コンテンツフィルタは組織内でグローバルなものであり、プロジェクト間で共有できます。



プロジェクトにすでに構築されたソースが含まれている場合、環境を追加すると、既存のコンテンツが自動的に入力されます。コンテンツは、サイクルの以前の環境から引き出されます（存在していた場合）。以前の環境がない場合は、プロジェクトソースが再度構築されるまで空のままになります。

6.3. フィルタテンプレート

いくつかの一般的なシナリオでフィルタの作成を支援するため、Uyuniにはフィルタテンプレートが用意されています。これらのテンプレートを適用すると、特定のユースケースに合わせてカスタマイズされた一連のフィルタを事前に作成できます。

このセクションでは、使用可能なテンプレートとその使用方法について説明します。

6.3.1. SUSE製品に基づくライブパッチ処理

ライブパッチ処理を含むプロジェクトでは、ライブパッチパッケージのみがクライアントに更新として提供されるように、定期的な将来のカーネルパッケージを除外する必要があります。一方で、システムの整合性を維持するために、すでにインストールされている通常のカーネルパッケージを含める必要があります。

このテンプレートを適用すると、この動作を実現するために必要な3つのフィルタが作成されます。

- ベースカーネルバージョンと同じ **kernel-default** パッケージを含むパッチを許可する
- reboot_suggested** キーワードを含むパッチを拒否する
- installhint(reboot-needed)** という名前を提供するパッケージを含むパッチを拒否する

ライブパッチ処理プロジェクトの設定方法の詳細については、[administration:content-lifecycle-examples.pdf](#)を参照してください。

プロシージャ: テンプレートの適用

- Uyuni Web UIで、**コンテンツライフサイクル** > **フィルタ**に移動して、**[Create Filter]** (フィルタの作成) をクリックします。
- ダイアログで、**[Use a template]** (テンプレートを使用) をクリックします。それに応じて入力が変更されます。
- [プレフィックス] フィールドに名前のプレフィックスを入力し、**この値は、テンプレートによって生成された個々のフィルタの名前の前に追加されます。** テンプレートがプロジェクトのコンテキストで適用されている場合、このフィールドにはプロジェクトラベルが事前に入力されます。
- [テンプレート] フィールドで、**[SUSE 製品に基づくライブパッチ処理]**を選択します。
- [製品] フィールドで、ライブパッチ処理を設定する製品を選択します。
- [カーネル] フィールドで、選択した製品で使用可能なバージョンのリストからカーネルバージョンを選択します。以降の通常のカーネルパッチを拒否するフィルタは、このバージョンに基づきます。
- [[保存]**をクリックして、フィルタを作成します。
- コンテンツライフサイクル > プロジェクトに移動して、プロジェクトを選択します。
- [[フィルタの割り当て]** > **[取り外し]**をクリックします。
- 指定したプレフィックスを持つフィルタを3つ選択し、**[[保存]**をクリックします。

6.3.2. システムに基づくライブパッチ処理

特定の登録済みシステムにインストールされているカーネルバージョンに基づいてライブパッチ処理プロジェクトを設定する場合、「システムに基づくライブパッチ処理」テンプレートを使用できます。

このテンプレートを適用すると、この動作を実現するために必要な3つのフィルタが作成されます。

- ベースカーネルバージョンと同じ **kernel-default** パッケージを含むパッチを許可する
- reboot_suggested** キーワードを含むパッチを拒否する
- installhint(reboot-needed)** という名前を提供するパッケージを含むパッチを拒否する

ライブパッチ処理プロジェクトの設定方法の詳細については、[administration:content-lifecycle-examples.pdf](#)を参照してください。

プロシージャ: テンプレートの適用

- Uyuni Web UIで、**コンテンツライフサイクル** > **フィルタ**に移動して、**[Create Filter]** (フィルタの作成) をクリックします。

2. ダイアログで、[Use a template] (テンプレートを使用) をクリックします。それに応じて入力が変更されます。
3. [プレフィックス] フィールドに、名前のプレフィックスを入力し、またはテンプレートによって作成されたすべてのフィルタの名前の前に追加されます。テンプレートがプロジェクトのコンテキストで適用されている場合は、このフィールドにはプロジェクトラベルが事前に入力されます。
4. [テンプレート] フィールドで、[固有のシステムに基づくライブパッチ処理] を選択します。
5. [システム] フィールドで、リストからシステムを選択するか、システム名の入力を開始して、オプションを絞り込みます。
6. [カーネル] フィールドで、選択したシステムにインストールされているバージョンのリストからカーネルバージョンを選択します。以降の通常のカーネルパッチを拒否するフィルタは、このバージョンに基づきます。
7. [保存] をクリックして、フィルタを作成します。
8. コンテンツライフサイクル > プロジェクトに移動して、プロジェクトを選択します。
9. [ファイルタの割り当て] をクリックします。
10. 指定したプレフィックスを持つフィルタを3つ選択し、[保存] をクリックします。

6.3.3. デフォルトのAppStreamモジュール

プロジェクトに含まれているモジュラリポジトリですべてのモジュールを使用できるようにする場合は、このフィルタテンプレートを使用してモジュールを自動的に追加できます。

このテンプレートを適用すると、モジュールごとにAppStreamフィルタとそのデフォルトストリームが作成されます。

このプロセスがプロジェクトのページから実行された場合、フィルタは自動的にプロジェクトに追加されます。追加されない場合は、作成されたフィルタをコンテンツライフサイクル > フィルタに一覧表示し、必要に応じて任意のプロジェクトに追加できます。

個々のフィルタを編集して別のモジュールストリームを選択したり、ターゲットリポジトリからそのモジュールを除外するために完全に削除したりできます。



すべてのモジュールストリームが相互に互換性がある訳ではないため、個々のストリームを変更すると、モジュール間依存関係を正常に解決できない可能性があります。この場合、プロジェクトの詳細ページのフィルタペインに問題を説明するエラーが表示され、すべてのモジュールの選択が互換性を持つようになるまでビルボタンは無効になります。

AppStreamリポジトリをコンテンツライフサイクル管理で設定する方法の詳細については、[administration:content-lifecycle-examples.pdf](#)を参照してください。

プロシージャ: テンプレートの適用

1. UyuniのWeb UIで、Content Lifecycle > Projectsに移動し、プロジェクトを選択します。

2. [フィルタ] セクションで、[フィルタの割り当て]/[取り外し]をクリックし、[新規]をクリックします。
3. ダイアログで、[Use a template] (テンプレートを使用) をクリックします。それに応じて入力が変更されます。
4. [プレフィックス] フィールドに、名前のプレフィックスを入力し、またはテンプレートによって作成されたすべてのフィルタの名前の前に追加されます。テンプレートがプロジェクトのコンテキストで適用されている場合は、このフィールドにはプロジェクトラベルが事前に入力されます。
5. [テンプレート] フィールドで、[デフォルトのAppStream モジュール] を選択します。
6. [チャーンネル] フィールドで、モジュールを取得するモジュラーチャンネルを選択します。このドロップダウンには、モジュラーチャンネルのみが表示されます。
7. [保存] をクリックして、フィルタを作成します。
8. [フィルタ] セクションまでスクロールして、新しく割り当てられたAppStreamフィルタを確認します。
9. 個々のフィルタを編集/削除して、ニーズに合わせてプロジェクトを調整できます。

6.4. コンテンツライフサイクルプロジェクトの構築

プロジェクトを作成し、環境を定義し、ソースとフィルタを割り当てたら、初めてプロジェクトを構築できます。

構築によって、割り当てられたソースにフィルタが適用され、プロジェクトの最初の環境にそれらが複製されます。

複数のコンテンツプロジェクトのソースとして同じベンダーチャンネルを使用できます。この場合、Uyuniはクローンチャンネルごとに新しいパッチクローンを作成しません。代わりに、1つのパッチクローンがすべてのクローンチャンネル間で共有されます。これにより、パッチが撤回された場合やパッチ内のパッケージが変更された場合など、ベンダがパッチを変更すると問題が発生する可能性があります。コンテンツプロジェクトの1つを構築すると、コンテンツプロジェクトの他の環境や組織内の他のコンテンツプロジェクトチャンネルにチャンネルがある場合でも、クローンパッチを共有するすべてのチャンネルがデフォルトでオリジナルと同期されます。この動作は、組織の設定で自動パッチ同期をオフにすることで変更できます。パッチを共有しているすべてのチャンネルのパッチを後で手動で同期するには、[ソフトウェア > 管理 > チャンネル](#)に移動し、同期するチャンネルをクリックして、[同期] サブタブに移動します。パッチを手動で同期しても、パッチを共有するすべての組織チャンネルに影響します。

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構築

1. Uyuni Web UIで、[コンテンツライフサイクル > プロジェクト](#)に移動し、構築するプロジェクトを選択します。
2. 割り当てられたソースとフィルタを確認し、[ビルド] をクリックします。
3. このビルドの変更または更新を説明するバージョンメッセージを提供します。
4. [環境ライフサイクル] セクションでビルドの進行状況を監視できます。

ビルドが完了すると、環境バージョンが1つ増え、ソフトウェアチャンネルなどのビルドソースをクライアントに割り当てるすることができます。

6.5. 環境のプロモート

プロジェクトが構築されると、構築されたソースを順次、環境にプロモートできます。

プロセージャ: 環境のプロモート

- Uyuni Web UIで、**コンテンツライフサイクル** > **プロジェクト**に移動し、作業するプロジェクトを選択します。
- [環境ライフサイクル] セクションで、後継環境にプロモートする環境を見つけ、[Promote] (プロモート) をクリックします。
- [環境ライフサイクル] セクションでビルトの進行状況を監視できます。

6.6. 環境にクライアントを割り当てる

コンテンツライフサイクルプロジェクトを構築およびプロモートすると、Uyuniはソフトウェアチャンネルのツリーを作成します。これを環境に追加するには、クライアントの [システムの詳細] ページの **ソフトウェア** > **ソフトウェアチャンネル** を使用して、ベースソフトウェアチャンネルと子ソフトウェアチャンネルをクライアントに割り当てます。



新たに追加されたクローンチャンネルはクライアントに自動的に割り当てられません。ソースを追加またはプロモートする場合は、チャンネル割り当てを手動で確認して更新する必要があります。

自動割り当ては、今後のバージョンでUyuniに追加される予定です。

6.7. コンテンツライフサイクル管理の例

このセクションでは、コンテンツライフサイクル管理の使用方法の一般的な例をいくつか示します。これらの例を使用して、独自にカスタマイズされた実装を構築します。

6.7.1. 月次パッチサイクルのプロジェクトの作成

月次パッチサイクルのプロジェクト例は、以下で構成されます。

- By Date** (日付別) フィルタの作成
- プロジェクトへのフィルタの追加
- 新しいプロジェクトビルトへのフィルタの適用
- プロジェクトからのパッチの除外
- プロジェクトにパッチを含める

6.7.1.1. **By Date** (日付別) フィルタの作成

指定した日付以降にリリースされたすべてのパッチは **By Date** (日付別) フィルタによって除外されます。このフィルタは、月次パッチサイクルに従うコンテンツライフサイクルプロジェクトに役立ちます。

プロシージャ: **By Date** (日付別) フィルタの作成

1. Uyuni Web UIで、**コンテンツライフサイクル** > **フィルタ**に移動して、**[Create Filter]** (フィルタの作成) をクリックします。
2. **[フィルタ名]** フィールドに、フィルタの名前を入力しますたとえば**Exclude patches by date** (日付別にパッチを除外する)。
3. **[フィルタタイプ]** フィールドで、**[Patch (Issue date)]** (パッチ(発行日)) を選択します。
4. **[マッチャード]** フィールドでは、**[later or equal]** (それ以降) が自動選択されます。
5. 日時を選択します。
6. **[[保]存]** をクリックします。

6.7.1.2. プロジェクトへのフィルタの追加

プロシージャ: プロジェクトへのフィルタの追加

1. Uyuni Web UIで、**コンテンツライフサイクル** > **プロジェクト**に移動して、リストからプロジェクトを選択します。
2. **[フィルタの割り当て]** リンクをクリックして、すべての使用可能なフィルタを表示します。
3. 新しい **[Exclude patches by date]** (日付別にパッチを除外する) を選択します。
4. **[[保]存]** をクリックします。

6.7.1.3. 新しいプロジェクトビルトへのフィルタの適用

新しいフィルタがフィルタリストに追加されますが、プロジェクトに適用する必要があります。 フィルタを適用するには、最初の環境を構築する必要があります。

プロシージャ: フィルタの使用

1. **[ビルド]** をクリックして、最初の環境を構築します。
2. オプション: メッセージを追加します。 メッセージを使用して、ビルト履歴を追跡できます。
3. テストサーバで新しいチャンネルを使用して、フィルタが正しく機能していることを確認します。
4. **[Promote]** (プロモート) をクリックして、次の環境にコンテンツを移動します。 多数のフィルタがある場合、または非常に複雑な場合は、ビルトに時間がかかります。

6.7.1.4. プロジェクトからのパッチの除外

テストは、問題を発見するのに役立つ場合があります。 問題が見つかった場合は、**by date** (日付別) フィルタの前にリリースされた問題のパッチを除外します。

プロシージャ: パッチの除外

1. Uyuni Web UIで、**コンテンツライフサイクル** > **フィルタ**に移動して、**[Create Filter]** (フィルタの作成) をクリックします。
2. **[フィルタ名]** フィールドに、フィルタの名前を入力しますたとえば、**Exclude openjdk patch**

(openjdkパッチの除外)。

3. [フィルタタイプ] フィールド **Patch (Advisory Name)**] (パッチ(アドバイザリ名)) を選択します。
4. [マッチャード] フィールドで、[等しい] を選択します。
5. [アドバイザリ名] フィールドに、アドバイザリの名前を入力または既存の **SUSE-15-2019-1807** を選択します。
6. [保存] をクリックします。
7. コンテンツライフサイクル > プロジェクトに移動して、プロジェクトを選択します。
8. [フィルタの割り当て / 取り外し] リンクをクリックして **Exclude openjdk patch (openjdkの除外)** を選択し、[保存] をクリックします。

[ビルド] ボタンを使用してプロジェクトを再構築すると、新しいフィルタが、以前に追加した **by date**] (日付別) フィルタとともに使用されます。

6.7.1.5. プロジェクトにパッチを含める

この例では、セキュリティアラートを受信しています。重要なセキュリティパッチが、現在作業している月の最初の日から数日後にリリースされました。新しいパッチの名前は **SUSE-15-2019-2071** です。この新しいパッチを環境に含める必要があります。



[許可] フィルタ規則は、[拒否] フィルタ規則の除外機能を上書きし詳細。については、Administration > Content-lifecycle を参照してください。

プロシージャ: プロジェクトにパッチを含める

1. Uyuni Web UIで、コンテンツライフサイクル > フィルタに移動して、[Create Filter] (フィルタの作成) をクリックします。
2. [フィルタ名] フィールドに、フィルタの名前を入力しますたとえば **include kernel security fix** (カーネルセキュリティ修正を含める)。
3. [フィルタタイプ] フィールド **Patch (Advisory Name)**] (パッチ(アドバイザリ名)) を選択します。
4. [マッチャード] フィールドで、[等しい] を選択します。
5. [アドバイザリ名] フィールドに、「**SUSE-15-2019-2071**」と入力し、[許可] をオンにします。
6. [保存] をクリックして、フィルタを保存します。
7. コンテンツライフサイクル > プロジェクトに移動して、リストからプロジェクトを選択します。
8. [フィルタの割り当て / 取り外し] リンクをクリックして **Include kernel security patch (カーネルセキュリティパッチを含める)** を選択します。
9. [保存] をクリックします。
10. [ビルド] をクリックして、環境を再構築します。

6.7.2. 既存の月次パッチサイクルの更新

月次パッチサイクルが完了すると、次の月のパッチサイクルを更新できます。

プロシージャ: 月次パッチサイクルの更新

1. [**by date**] (日付別) フィールドで、フィルタの日付を次の月に変更します。 または、新しいフィルタを作成して、プロジェクトへの割り当てを変更します。
2. **SUSE-15-2019-1807** の除外フィルタをプロジェクトから取り外すことができるかどうか確認します。この問題を解決するために使用できる新しいパッチがある場合があります。
3. 以前に追加した [**許 可**] フィルタを取り外します。 パッチはデフォルトで含まれています。
4. プロジェクトを再構築して、来月のパッチを適用した新しい環境を作成します。

6.7.3. ライブパッチ処理でプロジェクトを強化

このセクションでは、ライブパッチ処理用の環境を作成するためのフィルタの設定について説明します。

ライブパッチ処理を使用する準備をしている場合には、いくつかの重要な考慮事項があります

- ・ システムでカーネルバージョンを1つだけ使用してください。 ライブパッチ処理パッケージは、特定のカーネルとともにインストールされます。
- ・ ライブパッチ処理の更新は、1つのパッチとして出荷されます。
- ・ 新しいシリーズのライブパッチ処理カーネルを開始するカーネルパッチごとに、**要 再 起 動** フラグが表示されません。カーネルパッチには、ライブパッチ処理ツールが付属しています。 それらをインストールしたら、来年になる前に少なくとも1回システムを再起動する必要があります。
- ・ インストール済みカーネルバージョンに一致するライブパッチ更新のみをインストールします。
- ・ ライブパッチは、スタンダードアロンパッチとして提供されます。 現在インストールされているカーネルバージョンよりも新しいカーネルバージョンの通常のカーネルパッチをすべて除外する必要があります。



6.7.3.1. より新しいカーネルバージョンのパッケージを除外する

この例では、**SUSE-15-2019-1244** パッチでシステムを更新します。 このパッチには、**kernel-default-4.12.14-150.17.1-x86_64** が含まれています。

より新しいバージョンの **kernel-default** を含むすべてのパッチを除外する必要があります。

プロシージャ: より新しいカーネルバージョンのパッケージを除外する

1. Uyuni Web UIで、**コンテンツライフサイクル** フィルタに移動して、[**Create Filter**] (フィルタの作成) をクリックします。

2. [フ ィ ル タ 名] フィールドに、フィルタの名前を入力しますたとえば**Exclude kernel greater than 4.12.14-150.17.1** (4.12.14-150.17.1より新しいカーネルを除外する)。
3. [フ ィ ル タ タ イ プ] フィールド **Contains Package** (パッチ(パッケージを含む)) を選択します。
4. [マ ッ チ ャ ー] フィールドで**version greater than** (指定したものより新しいバージョン) を選択します。
5. [パ ッ ケ ー ジ 名] フィールドに、「**kernel-default**」と入力します。
6. [エ ポ ッ ク] フィールドを空のままにします。
7. [バ ー ジ ョ ン] フィールドに、「**4.12.14**」と入力します。
8. [リ リ ー ス] フィールドに、「**150.17.1**」と入力します。
9. [保 存] をクリックして、フィルタを保存します。
10. コンテンツライフサイクル > プロジェクトに移動して、プロジェクトを選択します。
11. [フ ィ ル タ の 割 り 当 て / 取 り 外 し] をクリックします。
12. [Exclude kernel greater than 4.12.14-150.17.1] (4.12.14-150.17.1より新しいカーネルを除外する) を選択して、[保 存] をクリックします。

[ビルド] をクリックすると、新しい環境が作成されま新しい環境には、インストールしたバージョンまでのすべてのカーネルパッチが含まれています。



より新しいカーネルバージョンのすべてのカーネルパッチは削除されます。 ライブパッチ処理カーネルは、シリーズの最初でない限り、引き続き使用できます。



このプロセッジヤはフィルタテンプレートを使用して自動化できます。 ライブパッチ処理フィルタテンプレートの適用方法の詳細について
は、[administration:content-lifecycle.pdf](#)を参照してください。

6.7.4. ライブパッチ処理用の新しいカーネルバージョンに切り替える

特定のカーネルバージョン用のライブパッチ処理は、1年間のみ利用できます。 1年後、システムのカーネルを更新する必要があります。 以下の環境変更を実行します。

プロセッジヤ: 新しいカーネルバージョンに切り替える

1. アップグレードするカーネルバージョンを決定します。 例: **4.12.14-150.32.1**
2. 新しいカーネルバージョンフィルタを作成します。
3. 以前のフィルタ [Exclude kernel greater than 4.12.14-150.17.1] (4.12.14-150.17.1より新しいカーネルを除外する) を取り外し、新しいフィルタを割り当てます。

[ビルド] をクリックして、環境を再構築しま新しい環境には、選択した新しいカーネルバージョンまでのすべてのカーネルパッチが含まれています。 これらのチャンネルを使用しているシステムには、インストールに使用できるカーネル更新があります。 アップグレードを実行した後で、システムを再起動する必要があります。 新しいカーネルは1年間有効です。 その年にインストールされたすべてのパッケージは、現在の

ライブパッチ処理カーネルフィルタと一致します。

6.7.5. AppStreamフィルタ

Red Hat Enterprise Linux 8クライアントを使用している場合は、Red Hat Enterprise Linux AppStreamリポジトリなどのモジュラーリポジトリから直接インストールやアップグレードなどのパッケージ操作を実行できません。AppStreamフィルタを使用して、モジュラーリポジトリを通常のリポジトリに変換できます。これはパッケージをリポジトリに保持し、モジュールのメタデータを削除することにより行われます。その結果作成されたリポジトリは、通常のリポジトリと同じ方法でUyuniで使用できます。

AppStreamフィルタはターゲットリポジトリに含める単一のモジュールストリームを選択します。複数のフィルタを追加して、複数のモジュールストリームを選択できます。

CLMプロジェクトでAppStreamフィルタを使用しない場合は、モジュラーソースのモジュールメタデータは未加工のままで、ターゲットリポジトリには同じモジュールメタデータが含まれます。CLMプロジェクトで少なくとも1つのAppStreamフィルタが有効になっている限り、すべてのターゲットリポジトリが通常のリポジトリに変換されます。

AppStreamフィルタを使用するには、**Red Hat Enterprise Linux AppStream**などのモジュラーリポジトリを含むCLMプロジェクトが必要です。開始する前に、必要なモジュールがソースとして含まれていることを確認してください。

プロシージャ: AppStreamフィルタの使用

1. Uyuni Web UIで、Red Hat Enterprise Linux 8 CLMプロジェクトに移動します。プロジェクトにAppStreamチャンネルが含まれていることを確認します。
2. (フィルタの作成) をクリックし、次のパラメータを使用します。
 - [フィルタ名] フィールドには、新しいフィルタの名前を入力します。
 - [フィルタタイプ] フィールドで、[モジュール(ストリーム)] を選択します。
 - [モジュール名] フィールドに、モジュール名を入力します。たとえば、**postgresql**。
 - [ストリーム] フィールドに、目的のストリームの名前を入力します。たとえば、**10**のフィールドを空のままにする場合、モジュールのデフォルトのストリームが選択されます。
3. をクリックして、新しいフィルタを作成します。
4. コンテンツライフサイクル › プロジェクトに移動して、プロジェクトを選択します。
5. フィルタの割り当て / 取り外しをクリックし、新しいAppStreamフィルタを選択して、 をクリックします。

[Create/Edit Filter] (フィルタの作成/編集) フォームのブラウズ機能を使用して、モジュラーチャンネルで使用可能なモジュールストリームのリストからモジュールを選択できます。

プロシージャ: 使用可能なモジュールストリームのブラウズ

1. Uyuni Web UIで、Red Hat Enterprise Linux 8 CLMプロジェクトに移動します。プロジェクトにAppStreamチャンネルが含まれていることを確認します。
2. (フィルタの作成) をクリックし、次のパラメータを使用します。

- [フィルタ名] フィールドには、新しいフィルタの名前を入力します。
 - [フィルタタイプ] フィールドで、[モジュール(ストリーム)] を選択します。
3. [Browse available modules] (使用可能なモジュールをブラウズ) をクリックして、すべてのモジュラーチャンネルを表示します。
4. モジュールとストリームをブラウズするチャンネルを選択します。
- [モジュール名] フィールドで、検索するモジュール名の入力を開始するか、リストから選択します。
 - [ストリーム] フィールドで、検索するストリーム名の入力を開始するか、リストから選択します。



チャンネル選択はモジュールのブラウズのみを目的としています。選択したチャンネルはフィルタとともに保存されず、CLMプロセスには影響しません。

ターゲットリポジトリに含める他のモジュールストリーム用に追加のAppStreamフィルタを作成できます。選択したストリームが依存するモジュールストリームは自動的に含まれます。



競合する、互換性のない、または欠落しているモジュールストリームを指定しないように注意してください。たとえば、同じモジュールから2つのストリームを選択すると無効になります。

Web UIの[ビルド]ボタンを使用してCLMプロジェクトを構築する場合、ターゲットリポジトリは、選択したモジュールストリームからのパッケージを含む、モジュールを含まない通常のリポジトリです。

Chapter 7. 切断されたセットアップ

Uyuniは、インターネット接続できない場合は、切断された環境で使用できます。

リポジトリミラーリングツール(RMT)はSUSE Linux Enterprise 15以降で使用できます。RMTは、古いSUSE Linux Enterpriseインストールで使用できる、サブスクリプション管理ツール(SMT)に代わるものです。

切断されたUyuniセットアップでは、RMTまたはSMTは外部ネットワークを使用してSUSE Customer Centerに接続します。すべてのソフトウェアチャンネルとリポジトリは、リムーバブルストレージデバイスに同期されます。その後、ストレージデバイスを使用して、切断されたUyuniのインストールを更新できます。

このセットアップにより、Uyuniのインストールをオフラインで、切断された環境のままにできます。



Uyuniサーバを直接管理するには、RMTまたはSMTインスタンスを使用する必要があります。カスケードで2番目のRMTまたはSMTインスタンスを管理するために使用することはできません。

RMTの詳細については、<https://documentation.suse.com/sles/15-SP3/html/SLES-all/book-rmt.html>を参照してください。

7.1. RMTの同期

SUSE Linux Enterprise 15インストールでRMTを使用して、SUSE Linux Enterprise 12以降を実行しているクライアントを管理できます。

Uyuniのインストールごとに専用RMTインスタンスを設定することをお勧めします。

プロシージャ: RMTの設定

1. RMTインスタンスで、RMTパッケージをインストールします。

```
zypper in rmt-server
```

2. YaSTを使用してRMTを設定します。

```
yast2 rmt
```

3. プロンプトに従ってインストールを完了します。

RMTの設定の詳細については、<https://documentation.suse.com/sles/15-SP3/html/SLES-all/book-rmt.html>を参照してください。

プロシージャ: RMTとSCCの同期

1. RMTインスタンスで、組織で使用可能なすべての製品とリポジトリを一覧にします。

```
rmt-cli products list --all
rmt-cli repos list --all
```

- 組織で使用可能なすべての更新を同期します。

```
rmt-cli sync
```

systemdを使用して定期的に同期するようにRMTを設定することもできます。

- 必要な製品を有効にします。たとえば、SLES 15を有効にするには、次のようにします。

```
rmt-cli product enable sles/15/x86_64
```

- リムーバブルストレージに同期されたデータをエクスポートします。この例では、ストレージメディアは **/mnt/usb** にマウントされます。

```
rmt-cli export data /mnt/usb
```

- リムーバブルストレージに有効なリポジトリをエクスポートします。

```
rmt-cli export settings /mnt/usb
rmt-cli export repos /mnt/usb
```



外部ストレージが、RMTユーザが書き込み可能なディレクトリにマウントされていることを確認します。RMTユーザ設定は、**/etc/rmt.conf** の `cli` セクションで変更できます。

7.2. SMTの同期

SMTはSUSE Linux Enterprise 12に含まれ、SUSE Linux Enterprise 10以降を実行しているクライアントを管理するために使用できます。

SMTでは、リポジトリとパッケージを同期するために、SMTインスタンス上にローカルミラーディレクトリを作成する必要があります。

SMTのインストールおよび設定に関する詳細については、<https://documentation.suse.com/sles/12-SP5/html/SLES-all/book-smt.html>を参照してください。

プロシージャ: SMTとSCCの同期

1. SMTインスタンスで、データベース置換ファイルを作成します。

```
smt-sync --createdbreplacementfile /tmp/dbrepl.xml
```

2. リムーバブルストレージに同期されたデータをエクスポートします。この例では、ストレージメディアは `/mnt/usb` にマウントされます。

```
smt-sync --todir /mnt/usb
smt-mirror --dbreplfile /tmp/dbrepl.xml --directory /mnt/usb \
--fromlocalsmt -L /var/log/smt/smt-mirror-export.log
curl https://scc.suse.com/suma/product_tree.json -o
/mnt/usb/product_tree.json
```



外部ストレージがRMTユーザによって書き込み可能なディレクトリにマウントされていることを確認します。 `/etc/smt.conf` のSMTユーザ設定を変更できます。

7.3. 必須チャンネル

Uyuniが指定されたチャンネルを同期できるようにするには、対応するUyuniクライアントツールチャンネルが必要です。これらのチャンネルが有効でない場合、Uyuniはその製品を検出できない場合があります。

次のコマンドを実行して、これらの必須チャンネルを有効にします。

SLES 12およびSLES for SAPやSLE HPCなどのSLES 12に基づく製品

RMT: `rmt-cli products enable sle-manager-tools/12/x86_64`

SMT: `smt repos -p sle-manager-tools,12,x86_64`

SLES 15およびSLES for SAPやSLE HPCなどのSLES 15に基づく製品

RMT: `rmt-cli products enable sle-manager-tools/15/x86_64`

SMT: `smt repos -p sle-manager-tools,15,x86_64`

次に、チャンネルをミラーリングしてエクスポートします。

他のディストリビューションまたはアーキテクチャを有効にすることができます。 製品チャンネルまたはリポジトリのミラーリングを有効にする方法の詳細については、次のドキュメントを参照してください。

RMT

<https://documentation.suse.com/sles/15-SP4/html/SLES-all/cha-rmt-mirroring.html#sec-rmt-mirroring-enable-disable>

SMT

<https://documentation.suse.com/sles/12-SP5/single-html/SLES-smt/index.html#smt-mirroring-manage-domirror>

7.4. 切断されたサーバの同期

SUSE Customer Centerデータでロードされたリムーバブルメディアがある場合は、それを使用して切断されたサーバを同期できます。

プロシージャ: 切断されたサーバの同期

1. Uyuniサーバにリムーバブルメディアアデバイスをマウントします。 この例では、マウントポイントは **/media/disk** です。
2. **/etc/rhn/rhn.conf** を開き、次の行を追加または編集して、マウントポイントを定義します。

```
server.susemanager.frommdir = /media/disk
```

3. Tomcatサービスを再起動します。

```
systemctl restart tomcat
```

4. ローカルデータを更新します。

```
mgr-sync refresh
```

5. 同期を実行します。

```
mgr-sync list channels
mgr-sync add channel channel-label
```



同期に使用するリムーバブルディスクは常に同じマウントポイントで使用できる必要があります。ストレージメディアがマウントされていない場合は、同期をトリガしないでください。これにより、データが破損します。

Chapter 8. ディスク容量の管理

ディスク容量が不足すると、Uyuniデータベースとファイル構造に深刻な影響を及ぼす可能性があり、場合によっては回復できなくなります。

Uyuniは、空きディスク容量のために一部のディレクトリを監視します。 監視するディレクトリと作成される警告を変更できます。 すべての設定は、`/etc/rhn/rhn.conf` 設定ファイルで行われます。

監視対象ディレクトリのいずれかの使用可能容量が警告しきい値を下回ると、設定された電子メールアドレスにメッセージが送信され、サインインページの上部に通知が表示されます。

8.1. 監視対象ディレクトリ

デフォルトでは、Uyuniは以下のディレクトリを監視します。

- `/var/lib/pgsql`
- `/var/spacewalk`
- `/var/cache`
- `/srv`

監視するディレクトリは、`spacecheck_dirs` パラメータで変更できます。 スペースで区切って、複数のディレクトリを指定できます。

例:

```
spacecheck_dirs = /var/lib/pgsql /var/spacewalk /var/cache /srv
```

8.2. しきい値

デフォルトでは、Uyuniは、監視対象ディレクトリで使用可能な合計容量の10%未満になると警告を作成します。 また、監視対象ディレクトリの空き容量が5%未満になると、クリティカルアラートが作成されます。

これらのアラートしきい値は `spacecheck_free_alert` および `spacecheck_free_critical` パラメータを使用して変更できます。

例:

```
spacecheck_free_alert = 10
spacecheck_free_critical = 5
```

8.3. サービスのシャットダウン

デフォルトでは、クリティカルアラートしきい値に達すると、Uyuniはspacewalkサービスをシャットダウンします。

この動作は、**spacecheck_shutdown** パラメータで変更できます。**true** の値はシャットダウン機能を有効にします。 その他の値は無効にします。

例:

```
spacecheck_shutdown = true
```

8.4. スペースチェックの無効化

スペースチェックツールはデフォルトで有効になっています。 次のコマンドを使用して完全に無効にできます。

```
systemctl stop spacewalk-diskcheck.timer  
systemctl disable spacewalk-diskcheck.timer
```

spacewalk-diskcheck.timer を無効にすると、アラートのしきい値に達した場合に定期的な電子メールアラートが停止しますが、警告通知はサインインページの上部に表示されます。

Chapter 9. イメージの構築と管理

9.1. イメージの構築の概要

Uyuniでは、システム管理者はコンテナおよびOSイメージを構築し、結果をイメージストアにプッシュできます。ワークフローは次のようにになります。

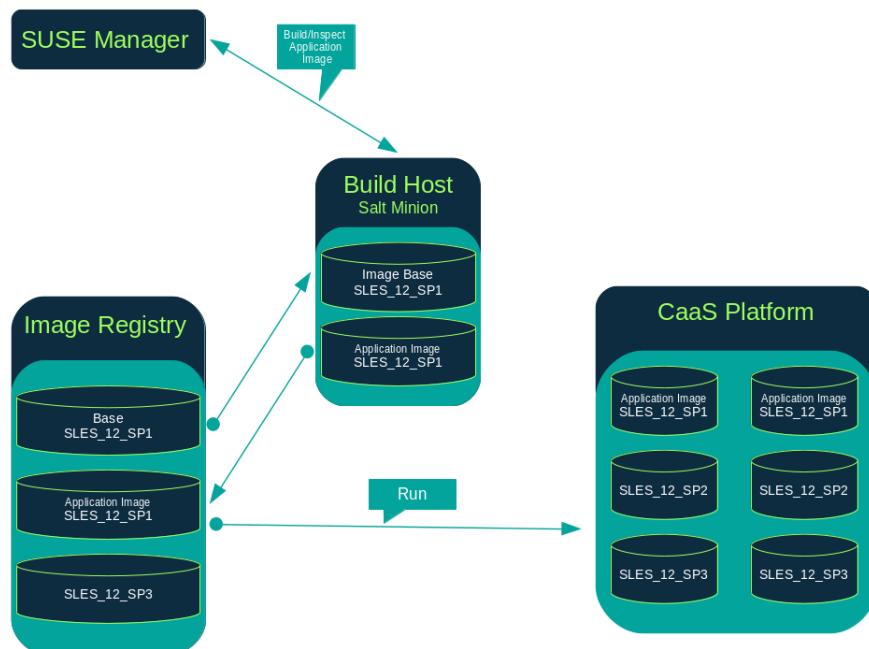
1. イメージストアを定義します
2. イメージプロファイルを定義し、それをソース(gitリポジトリまたはディレクトリのいずれか)に関連付けてます
3. イメージを構築します
4. イメージをイメージストアにプッシュします

Uyuniでは、次の2つの異なるビルドタイプ(dockerfile、およびKiwiイメージシステム)をサポートしています。

Kiwiビルドタイプは、システムイメージ、仮想イメージ、およびその他のイメージの構築に使用されます。Kiwiビルドタイプのイメージストアは、サーバ上の `/srv/www/os-images` にあるファイルシステムディレクトリとして事前定義されています。Uyuniは、`//<SERVER-FQDN>/os-images` からHTTPS経由でイメージストアを提供します。イメージストアの場所は固有であり、カスタマイズできません。

イメージは常に `/srv/www/os-image/<organization id>` に保存されます。

9.2. コンテナイメージ



9.2.1. 要件

コンテナ機能は、SUSE Linux Enterprise Server 12以降を実行しているSaltクライアントで使用できます。開始する前に、ご使用の環境が次の要件を満たしていることを確認してください。

- dockerfileと設定スクリプトを含む発行済みのgitリポジトリ。リポジトリはパブリックにもプライベートにもでき、GitHub、GitLab、またはBitBucketでホストする必要があります。
- Dockerレジストリなどの適切に設定されたイメージストア。

コンテナの詳細については、以下を参照してください。

- <https://documentation.suse.com/sles/15-SP3/html/SLES-all/book-container.html>

9.2.2. 構築ホストの作成

Uyuniでイメージを構築するには、構築ホストを作成して設定する必要があります。コンテナビルドホストは、SUSE Linux Enterprise 12以降を実行しているSaltクライアントです。このセクションでは、構築ホストの初期設定について説明します。



構築ホスト上のオペレーティングシステムは、ターゲットイメージ上のオペレーティングシステムと一致する必要があります。

たとえば、SUSE Linux Enterprise Server 15 (SP2以降)のOSバージョンを実行している構築ホスト上にSUSE Linux Enterprise Server 15ベースのイメージを構築します。SUSE Linux Enterprise Server 12 SP4またはSUSE Linux Enterprise Server 12 SP3 OSバージョンを実行している構築ホスト上にSUSE Linux Enterprise Server 12ベースのイメージを構築します。

クロスアーキテクチャビルドはサポートされていません。

Uyuni Web UIから、次のステップを実行して、構築ホストを設定します。

- システム > 概要ページから、構築ホストとして指定されるSaltクライアントを選択します。
- 選択したクライアントの [システムの詳細] ページから、コンテナモジュールを割り当てます。
ウェア > ソフトウェアチャンネルに移動して、コンテナモジュール(たとえば、[SLE-Module-Containers15-Pool] と [SLE-Module-Containers15-Updates]) [サブスクリプション] の [変更] をクリックして確定します。
- 高級ページから属性付 属 エンタイトルメント] リストから コンテナビルドホスト を有効にし、[プロパティの更新] をクリックして確定します。
- highstate を適用して必要なすべてのパッケージをインストールします。システムの詳細ページで、状態highstateを選択し、[highstate の適用] をクリックします。または、Uyuniサーバのコマンドラインからhighstateを適用します。

```
salt '$your_client' state.highstate
```

9.2.3. コンテナ用アクティベーションキーの作成

Uyuniを使用して構築されたコンテナは、イメージを構築するときに、アクティベーションキーに関連付けられたチャンネルをリポジトリとして使用します。このセクションでは、この目的のためにアドホックアクティベーションキーを作成する方法について説明します。



- コンテナを構築するには、`SUSE Manager Default`以外のチャンネルに関連付けられているアクティベーションキーが必要です。

Create Activation Key

Activation Key Details

Systems registered with this activation key will inherit the settings listed below.

Description:

Use this to describe what kind of settings this key will reflect on systems that use it. If left blank, this field will be filled in 'None'.

Key:

Activation key can contain only numbers [0-9], letters [a-z A-Z], '.', '-' and '.'.

Leave blank for automatic key generation. Note that the prefix is an indication of the SUSE Manager organization the key is associated with.

Usage:

Leave blank for unlimited use.

Base Channel:

Choose "SUSE Manager Default" to allow systems to register to the default SUSE Manager provided channel that corresponds to the installed SUSE Linux version. Instead of the default, you may choose a particular SUSE provided channel or a custom base channel, but if a system using this key is not compatible with the selected channel, it will fall back to its SUSE Manager Default channel.

Add-On System Types:

- Container Build Host
- Virtualization Host

Contact Method:

Universal Default:

Tip: Only one universal default activation key may be set for this organization. By setting this key as universal default, you will remove universal default status from the current universal default key if it exists. If this key is set as universal default, then newly-registered systems to your organization will inherit the properties of this key.

Create Activation Key

1. システム > アクティベーションキーを選択します。

2. [アクション] の [作成] をクリックします。

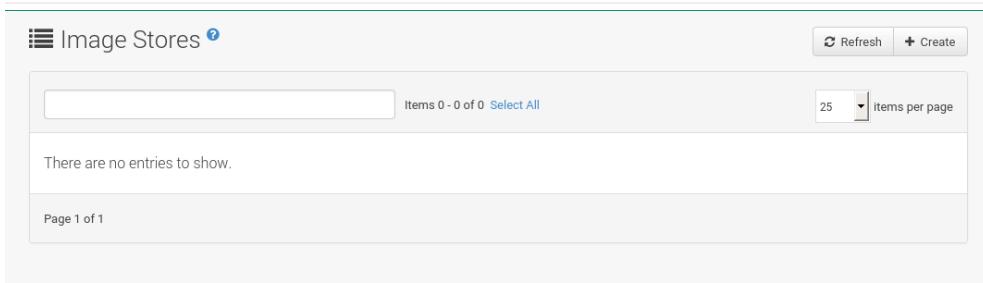
3. [説明] と [キー] 名を入力またはダウンメニューを使用してこのキーと関連付ける [ベースチャンネル] を選択します。

4. [アクション] の [作成] で確定します。

詳細については、[bp.key.management]を参照してください。

9.2.4. イメージストアの作成

すべての構築されたイメージは、イメージストアにプッシュされます。このセクションでは、イメージストアの作成について説明します。



1. イメージストアを選択します。
2. 「作成」をクリックして、新しいストアを作成します。

Create Image Store

Store Type *: Registry

Label *:

URI *:

Use credentials

Username *:

Password *:

+ Create **Clear fields**

3. 「ラベル」フィールドにイメージストアの名前を定義します。
4. コンテナレジストリホスト(内部か外部)の完全修飾ドメイン名(FQDN)として、「URI」フィールドに入力して、イメージレジストリへのパスを指定します。

registry.example.com

レジストリURI を使用して、すでに使用されているレジストリのイメージストアを指定することもできます。

registry.example.com:5000/myregistry/myproject

1. 「作成」をクリックして、新しいイメージストアを追加します。

9.2.5. イメージプロファイルの作成

すべてのコンテナイメージは、構築手順を含む、イメージプロファイルを使用して構築されます。このセクションでは、Uyuni Web UIでイメージプロファイルを作成する方法について説明します。

The screenshot shows a table header with columns for 'Label', 'Image Type', 'Target Image Store', 'Path', and 'Activation Key'. Below the table, there is a message: 'There are no entries to show.' and a footer indicating 'Page 1 of 1'.

プロセッジヤ: イメージプロファイルの作成

1. イメージプロファイルを作成するには、**イメージ > プロファイル**を選択し、**[作成]**をクリックします。

Create Image Profile

Label *:

Image Type *: Dockerfile

Target Image Store *: Select an image store

Path *: Format: giturl#branch:dockerfile_location

Activation Key: None

Custom Info Values: Create additional custom info values

+ Create **Clear fields**

2. [ラベル] フィールドに入力して、イメージプロファイルの名前を指定します。



コンテナイメージタグが`myproject/myimage`などの形式である場合は、イメージストアのレジストリURIに`/myproject`サフィックスが含まれていることを確認します。

3. [イメージタイプ] としてdockerfileを使用します。
4. ドロップダウンメニューを使用して、[ターゲットのイメージファイルからレジストリ]を選択します。
5. [パス] フィールドに、GitHub、GitLab、またはBitBucketリポジトリのURLを入力します。 URLはhttp、https、またはトークン認証URLである必要があります。 以下の形式のいずれかを使用します。

GitHubパスオプション

- GitHubシングルユーザプロジェクトリポジトリ

```
https://github.com/USER/project.git#branchname:folder
```

- GitHub組織プロジェクトリポジトリ

```
https://github.com/ORG/project.git#branchname:folder
```

- GitHubトークン認証

gitリポジトリがプライベートな場合、認証を含むようにプロファイルのURLを変更します。 GitHubトークンで認証するには次のURL形式を使用します。

```
https://USER:<AUTHENTICATION_TOKEN>@github.com/USER/project.git#master
:/container/
```

GitLabパスオプション

- GitLabシングルユーザプロジェクトリポジトリ

```
https://gitlab.example.com/USER/project.git#master:/container/
```

- GitLabグループプロジェクトリポジトリ

```
https://gitlab.example.com/GROUP/project.git#master:/container/
```

- GitLabトークン認証

gitリポジトリがプライベートで、パブリックにアクセスできない場合は、認証を含むようにプロファイルのgit URLを変更する必要があります。 GitLabトークンで認証するには、次のURL形式を使用します。

```
https://gitlab-ci-
token:<AUTHENTICATION_TOKEN>@gitlab.example.com/USER/project.git#maste
r:/container/
```



gitブランチを指定しない場合は、デフォルトで`master`ブランチが使用されます。 `folder`が指定されていない場合、イメージソース(dockerfileソース)はGitHubまたはGitLabチェックアウトのルートディレクトリにあると想定されます。

1. [アクティベーションキー]ボタンを選択します。キーは、プロファイルを使用するイメージが正しいチャンネルとパッケージに確実に割り当てられるようにします。



アクティベーションキーをイメージプロファイルに関連付けると、プロファイルを使用するすべてのイメージで正しいソフトウェアチャンネルとチャンネル内のすべてのパッケージが確実に使用されるようになります。

2. [作成]ボタンをクリックします。

Dockerfileソースの例

再利用できるイメージプロファイルは<https://github.com/SUSE/manager-build-profiles>で公開されています。



ARG パラメータは、構築されたイメージがUyuniが提供する目的のリポジトリに関連付けられていることを確認します。 **ARG** パラメータを使用すると、構築ホスト自体で使用されるSUSE Linux Enterprise Serverのバージョンとは異なる可能性のあるSUSE Linux Enterprise Serverのイメージバージョンを構築することもできます。

例:リポジトリファイルを示す**ARG repo** パラメータと **echo** コマンドは、目的のチャンネルバージョンの正しいパスを作成し、リポジトリファイルに挿入します。

リポジトリは、イメージプロファイルに割り当てたアクティベーションキーによって決定されます。

```
FROM registry.example.com/sles12sp2
MAINTAINER Tux Administrator "tux@example.com"

### Begin: These lines Required for use with {productname}

ARG repo
ARG cert

# Add the correct certificate
RUN echo "$cert" > /etc/pki/trust/anchors/RHN-ORG-TRUSTED-SSL-CERT.pem

# Update certificate trust store
RUN update-ca-certificates

# Add the repository path to the image
RUN echo "$repo" > /etc/zypp/repos.d/susemanager:dockerbuild.repo

### End: These lines required for use with {productname}

# Add the package script
ADD add_packages.sh /root/add_packages.sh

# Run the package script
RUN /root/add_packages.sh

# After building remove the repository path from image
RUN rm -f /etc/zypp/repos.d/susemanager:dockerbuild.repo
```

カスタム情報のキーと値のペアをDocker Buildargsとして使用する

カスタム情報のキーと値のペアを割り当てて、イメージプロファイルに情報を添付できます。さらに、これ

らのキーと値のペアは、Dockerビルドコマンドに`buildargs`として渡されます。

使用可能なカスタム情報キーと追加のキーの作成に関する詳細については、[Reference > Systems](#)を参照してください。

9.2.6. イメージの構築

イメージを構築するには、2つの方法があります。左側のナビゲーションバーから**イメージ > ビルド**を選択するか、**イメージ > プロファイルリスト**のビルトアイコンをクリックします。

Build Image

Version: latest

Image Profile: Select an image profile

Build Host: Select a build host

Earliest: 05.06. 18:00 CEST

Add to: new action chain

Build

プロシージャ: イメージのビルド

1. **イメージ > ビルド**を選択します。
2. デフォルトの **latest** (コンテナにのみ関連する)以外のバージョンを使用する場合は、別のタグ名を追加します。
3. [Build Profile] (ビルトプロファイル) と [構築ホスト] を選択します。



ビルトフィールドの右側にある [プロファイル概要] は強調表示します。プロファイルを選択すると、選択したプロファイルに関する詳細情報がこの領域に表示されます。

4. ビルドをスケジュールするには、[スケジュール] ボタンをクリックします。

9.2.7. イメージの取り込み

任意のイメージの取り込みおよび検査が可能です。左側のナビゲーションバーから**イメージ > イメージリスト**を選択します。[取り込み] ダイアログのテキストボックスに入力します。処理が完了すると、取り込まれたイメージが [イメージリスト] ページに一覧表示されます。

プロシージャ: イメージの取り込み

イメージリストから、[取り込み] をクリックして、[イメージの取り込み] ダイアログを開きます。

2. [イメージの取り込み] ダイアログで、次のフィールドに入力します。

イメージストア

検査のためにイメージがプルされるレジストリ。

イメージ名

レジストリのイメージの名前。

イメージバージョン

レジストリのイメージのバージョン。

構築ホスト

イメージをプルして検査する構築ホスト。

アクティベーションキー

イメージが検査されるソフトウェアチャンネルへのパスを提供するアクティベーションキー。

3. 確定するには、をクリックします。

イメージのエントリがデータベースに作成され、Uyuniの**Inspect Image**（イメージの検査）アクションがスケジュールされます。

処理が完了すると、取り込まれたイメージが **イメージリスト** に表示され、取り込まれたことを示す異なるアイコンが **[ビルド]** 列に表示されま、取り込まれたイメージのステータスアイコンはイメージの **[概要]** タブにも表示されます。

9.2.8. トラブルシューティング

イメージを操作する場合の既知の問題がいくつかあります。

- レジストリまたはgitリポジトリにアクセスするためのHTTPS証明書はカスタム状態ファイルによってクライアントに配備する必要があります。
- Dockerを使用したSSH gitアクセスは現在サポートされていません。

9.3. OSイメージ

OSイメージは、Kiwiイメージシステムによって構築されます。出力イメージはカスタマイズ可能で、PXE、QCOW2、LiveCD、またはその他のタイプのイメージにすることができます。

Kiwiビルドシステムの詳細については、[https://doc.opensuse.org/projects/kiwi/doc/\[Kiwiのドキュメント\]](https://doc.opensuse.org/projects/kiwi/doc/[Kiwiのドキュメント]) を参照してください。

9.3.1. 要件

Kiwiイメージ構築機能は、SUSE Linux Enterprise Server 12およびSUSE Linux Enterprise Server 11を実行しているSaltクライアントで利用できます。

Kiwiイメージ設定ファイルおよび設定スクリプトは、以下の場所のいずれかからアクセスできる必要があります。

ます。

- ・ Gitリポジトリ
- ・ HTTPでホストされたtarball
- ・ ローカル構築ホストディレクトリ

gitで提供される完全なKiwiリポジトリの例については、<https://github.com/SUSE/manager-build-profiles/tree/master/OSImage>を参照してください



Kiwiで構築されたOSイメージを実行しているホストには、少なくとも1GBのRAMが必要です。ディスク容量は、イメージの実際のサイズによって異なります。 詳細については、基になるシステムのドキュメントを参照してください。



構築ホストはSaltクライアントである必要があります。構築ホストを従来のクライアントとしてインストールしないでください。

9.3.2. 構築ホストの作成

Uyuniであらゆる種類のイメージを構築するには、構築ホストを作成して設定します。OSイメージ構築ホストは、SUSE Linux Enterprise Server 15 (SP2以降)、SUSE Linux Enterprise Server 12 (SP3以降)、またはSUSE Linux Enterprise Server 11 SP4で実行されているSaltクライアントです。

このプロセッサでは、構築ホストの初期設定について説明します。



構築ホスト上のオペレーティングシステムは、ターゲットイメージ上のオペレーティングシステムと一致する必要があります。

たとえば、SUSE Linux Enterprise Server 15 (SP2以降)のOSバージョンを実行している構築ホストにSUSE Linux Enterprise Server 15ベースのイメージを構築します。SUSE Linux Enterprise Server 12 SP4またはSUSE Linux Enterprise Server 12 SP3 OSバージョンを実行している構築ホストにSUSE Linux Enterprise Server 12ベースのイメージを構築します。SUSE Linux Enterprise Server 11 SP4 OSバージョンを実行している構築ホストにSUSE Linux Enterprise Server 11ベースのイメージを構築します。

クロスアーキテクチャビルドはできません。たとえば、SUSE Linux Enterprise Server 15 SP3を実行しているRaspberry PI (aarch64アーキテクチャ)構築ホストにRaspberry PI SUSE Linux Enterprise Server 15 SP3を構築する必要があります。

Uyuni Web UIで構築ホストを設定します。

1. システム > 概要ページから構築ホストとして指定するクライアントを選択します。

2. プロセスの詳細移動して、[付属エンタイトル: OS] トイメージビルドホスト [] を有効にします。[プロファイル] の [更新] で確定します。

System Name: d186.suse.de

Base System Type: Salt

Add-On System Types:

- Container Build Host
- OS Image Build Host

Description: OS Image Build Host (for KIWI images)

Facility Address:

City:

State/Province:

Country: None

Building:

3. システムの詳細 > ソフトウェア > ソフトウェアチャンネルに移動し、構築ホストのバージョンに応じて必要なソフトウェアのチャンネルを有効にします。

- SUSE Linux Enterprise Server 11構築ホストでは、Uyuniクライアントツール(**SLE-Manager-Tools11-Pool** と **SLE-Manager-Tools11-Updates**)が必要です。
 - SUSE Linux Enterprise Server 12構築ホストでは、Uyuniクライアントツール(**SLE-Manager-Tools12-Pool** と **SLE-Manager-Tools12-Updates**)が必要です。
 - SUSE Linux Enterprise Server 15構築ホストでは、SUSE Linux Enterprise Serverモジュール **SLE-Module-DevTools15-SP2-Pool** と **SLE-Module-DevTools15-SP2-Updates**が必要です。スケジュールを設定し、**[確認]**をクリックします。
4. `highstate`を適用してKiwiと必要なすべてのパッケージをインストールします。 システムの詳細ページで、状態 **highstate** を選択し、**[highstate] の [適用]** をクリックします。または、Uyuniサーバのコマンドラインからhighstateを適用します。

```
salt '$your_client' state.highstate
```

Uyuni Webサーバのパブリック証明書RPM

構築ホストのプロビジョニングでは、Uyuni証明書RPMを構築ホストにコピーします。 この証明書はUyuniによって提供されるリポジトリにアクセスするために使用されます。

証明書は、`mgr-package-rpm-certificate-osimage` パッケージスクリプトでRPMにパッケージ化されます。 パッケージスクリプトは新しいUyuniのインストール中に自動的に呼び出されます。

spacewalk-certs-tools パッケージをアップグレードすると、アップグレードします。ただし、証明書パスが変更

「**アップグレード プロジェクトの完了後に、`--ca-certfullpathcertificate`**」を使用してパッケージスクリプトを手動で呼び出します。

リスト 2. パッケージスクリプトの呼び出し例

```
/usr/sbin/mgr-package-rpm-certificate-osimage --ca-cert-full-path
/root/ssl-build/RHN-ORG-TRUSTED-SSL-CERT
```

証明書を含むRPMパッケージは、次のようなsalt-accessibleディレクトリに保存されます。

```
/usr/share/susemanager/salt/images/rhn-org-trusted-ssl-cert-osimage-1.0-
1.noarch.rpm
```

証明書を含むRPMパッケージは、ローカル構築ホストリポジトリで提供されます。

```
/var/lib/Kiwi/repo
```

ビルドソースに Uyuni SSL証明書を含むRPMパッケージを指定し、Kiwiの設定に **bootstrap** セクションの必須パッケージとして **rhn-org-trusted-ssl-cert-osimage** が含まれていることを確認します。

リスト 3. config.xml



```
...
<packages type="bootstrap">
  ...
    <package name="rhn-org-trusted-ssl-cert-osimage"
bootinclude="true" />
  </packages>
  ...

```

9.3.3. OSイメージ用アクティベーションキーの作成

イメージの構築時にOSイメージがリポジトリとして使用できるチャンネルに関連付けられたアクティベーションキーを作成します。

アクティベーションキーはOSイメージの構築に必須です。



OSイメージを構築するには、「SUSEマネージャデフォルト」以外のチャンネルに関連付けられたアクティベーションキーが必要です。

Create Activation Key ?

Activation Key Details

Systems registered with this activation key will inherit the settings listed below.

Description:	<input type="text"/>	Use this to describe what kind of settings this key will reflect on systems that use it. If left blank, this field will be filled in 'None'.
Key:	<input type="text" value="1-"/>	Activation key can contains only numbers [0-9], letters [a-z A-Z], '.', '_' and '.'. Leave blank for automatic key generation. Note that the prefix is an indication of the SUSE Manager organization the key is associated with.
Usage:	<input type="text"/>	Leave blank for unlimited use.
Base Channel:	<input type="text" value="SUSE Manager Default"/>	Choose "SUSE Manager Default" to allow systems to register to the default SUSE Manager provided channel that corresponds to the installed SUSE Linux version. Instead of the default, you may choose a particular SUSE provided channel or a custom base channel, but if a system using this key is not compatible with the selected channel, it will fall back to its SUSE Manager Default channel.
Add-On System Types:	<input type="checkbox"/> Container Build Host <input type="checkbox"/> Virtualization Host	
Contact Method:	<input type="text" value="Default"/>	
Universal Default:	<input type="checkbox"/>	Tip: Only one universal default activation key may be set for this organization. By setting this key as universal default, you will remove universal default status from the current universal default key if it exists. If this key is set as universal default, then newly-registered systems to your organization will inherit the properties of this key.
Create Activation Key		

1. Web UIで、**システム > アクティベーションキー**を選択します。

2. **[キーの作成]**をクリックします。

3. **[説明]**、**[キー]**の名前を入力し、ドロップダウンボックスを使用してキーに関連付ける**[ベースチャネル]**を選択します。

4. **[アクティベーションキーの作成]**で確定します。

詳細については、[\[bp.key.management\]](#)を参照してください。

9.3.4. イメージストアの作成

OSイメージには、大量のストレージ容量が必要になる場合があります。したがって、OSイメージストアは、ルートパーティションとは別の、独自のパーティションまたはBtrfsサブボリュームに配置することをお勧めします。デフォルトでは、イメージストアは/**/srv/www/os-images**にあります。



システム、仮想、およびその他のイメージの構築に使用されるKiwiビルドタイプのイメージストアは、まだサポートされていません。

イメージは常に **/srv/www/os-images/<organizationid>** で保存され、HTTP/HTTPs **https://<susemanager_host>/os-images/<organizationid>** を介してアクセスできます。

9.3.5. イメージプロファイルの作成

Web UIを使用してイメージプロファイルを管理します。

プロシージャ: イメージプロファイルの作成

1. イメージプロファイルを作成するには、**イメージプロファイル**から選択し、**[作成]**をクリックします。

2. [ラベル] フィールドに、「イメージプロファイル」の名前を入力します。
3. [イメージタイプ] として [Kiwi] を使用します。
4. イメージストアは自動的に選択されます。
5. Kiwi設定ファイルを含むディレクトリに [設定URL] を入力します。
 - a. git URI
 - b. HTTPS tarball
 - c. 構築ホストローカルディレクトリへのパス
6. 必要に応じて [Kiwi オプション] を入力し、複数のプロファイルが指定されている場合、`--profile <name>`を使用してアクティブなプロファイルを選択します。他のオプションについては、Kiwiのドキュメントを参照してください。
7. [アクティベーションキー] により、プロファイルを使用したイメージが正しいチャンネルとパッケージに確実に割り当てられます。



アクティベーションキーをイメージプロファイルに関連付け、イメージプロファイルで正しいソフトウェアチャンネルとパッケージが使用されるようにします。

8. [作成] ボタンで確定します。

ソースフォーマットオプション

- ・ リポジトリへのgit/HTTP(S) URL

構築するイメージのソースを含むgitリポジトリへのURL。リポジトリのレイアウトによって、URLは次のようにになります。

```
https://github.com/SUSE/manager-build-profiles
```

URLの`#`文字の後にブランチを指定できます。この例では、`master`ブランチを使用します。

```
https://github.com/SUSE/manager-build-profiles#master
```

`:`文字の後のイメージソースを含むディレクトリを指定できます。
は、`OSImage/POS_Image-JeOS6`を使用します。

この例で

```
https://github.com/SUSE/manager-build-
profiles#master:OSImage/POS_Image-JeOS6
```

- ・ tarballへのHTTP(S) URL

WebサーバでホストされているtarアーカイブへのURL(圧縮または非圧縮)。

```
https://myimagesourceserver.example.org/MyKiwiImage.tar.gz
```

- ・ 構築ホスト上のディレクトリへのパス

Kiwiビルドシステムソースを含むディレクトリへのパスを入力します。このディレクトリは選択した構築ホスト上に存在する必要があります。

```
/var/lib/Kiwi/MyKiwiImage
```

9.3.5.1. Kiwiソースの例

Kiwiソースは少なくとも `config.xml` で構成されています。通常、`config.sh` と `images.sh` も存在しま

す。 ソースには`root`サブディレクトリの下の最終イメージにインストールするファイルも含めることができます。

Kiwiビルドシステムについては、[https://doc.opensuse.org/projects/kiwi/doc/\[Kiwiのドキュメント\]](https://doc.opensuse.org/projects/kiwi/doc/[Kiwiのドキュメント])を参照してください。

SUSEでは、[https://github.com/SUSE/manager-build-profiles\[SUSE/manager-build-profiles\]](https://github.com/SUSE/manager-build-profiles[SUSE/manager-build-profiles])パブリックGitHubリポジトリで、完全に機能するイメージソースの例を提供しています。

リスト 4. JeOS config.xmlの例

```
<?xml version="1.0" encoding="utf-8"?>

<image schemaversion="6.1" name="POS_Image_JeOS6">
  <description type="system">
    <author>Admin User</author>
    <contact>noemail@example.com</contact>
    <specification>SUSE Linux Enterprise 12 SP3 JeOS</specification>
  </description>
  <preferences>
    <version>6.0.0</version>
    <packagemanager>zypper</packagemanager>
    <bootsplash-theme>SLE</bootsplash-theme>
    <bootloader-theme>SLE</bootloader-theme>

    <locale>en_US</locale>
    <keytable>us.map.gz</keytable>
    <timezone>Europe/Berlin</timezone>
    <hwclock>utc</hwclock>

    <rpm-excludedocs>true</rpm-excludedocs>
    <type boot="saltboot/suse-SLES12" bootloader="grub2"
checkprebuilt="true" compressed="false" filesystem="ext3" fsmountoptions
="acl" fsnocheck="true" image="pxe" kernel cmdline="quiet"></type>
  </preferences>
  <!-- CUSTOM REPOSITORY
  <repository type="rpm-dir">
    <source path="this://repo"/>
  </repository>
  -->
  <packages type="image">
    <package name="patterns-sles-Minimal"/>
    <package name="aaa_base-extras"/> <!-- wouldn't be SUSE without
that ;-->
    <package name="kernel-default"/>
    <package name="salt-minion"/>
```

```

...
</packages>
<packages type="bootstrap">
...
<package name="sles-release"/>
<!-- this certificate package is required to access {productname}
repositories
and is provided by {productname} automatically --&gt;
&lt;package name="rhn-org-trusted-ssl-cert-osimage" bootinclude=
"true"/&gt;

&lt;/packages&gt;
&lt;packages type="delete"&gt;
&lt;package name="mtools"/&gt;
&lt;package name="initviocons"/&gt;
...
&lt;/packages&gt;
&lt;/image&gt;
</pre>

```

9.3.6. イメージの構築

Web UIを使用してイメージを構築する2つの方法があります。 **イメージ > ビルド**を選択するか、**イメージ > プロファイルリスト**のビルドアイコンをクリックします。

The screenshot shows the 'Build Image' dialog box. It includes fields for Version (set to 'latest'), Image Profile (dropdown menu showing 'Select an image profile' with 'No profile selected' message), Build Host (dropdown menu), Earliest (calendar date 05.06.2023 at 18:00 CEST), and Add to (dropdown menu showing 'new action chain'). A green 'Build' button is at the bottom.

プロシージャ: イメージのビルド

1. **イメージ > ビルド**を選択します。
2. デフォルトの **latest** (コンテナのみに適用)以外のバージョンが必要な場合は別のタグ名を追加します。
3. [イメージプロファイル] および [構築ホスト] を選択します。



[プロファイル概要] がビルドフィールドの右側に表示されます。ファイルを選択したら、選択したプロファイルに関する詳細情報がここに表示されます。

4. ビルドをスケジュールするには、[スケジュール]ボタンをクリックします。



ビルドサーバは、イメージ構築プロセス中にどのような形式のオートマウンタも実行できません。必要に応じて、Gnomeセッションがrootとして実行されていないことを確認します。オートマウンタが実行されている場合、イメージのビルトは正常に終了しますが、イメージのチェックサムが異なるためエラーが発生します。

イメージが正常に構築されると、検査フェーズが開始されます。検査フェーズ中に、SUSE Managerではイメージに関する情報を収集します。

- ・ イメージにインストールされているパッケージのリスト
- ・ イメージのチェックサム
- ・ イメージタイプと他のイメージの詳細



構築されたイメージタイプが`PXE`の場合は、Saltピラーも生成されます。イメージのピラーはデータベースに保存され、Saltサブシステムは生成されたイメージに関する詳細にアクセスできます。詳細には、イメージファイルの場所と提供場所、イメージのチェックサム、ネットワークブートに必要な情報などが含まれます。

生成されたピラーはすべての接続されているクライアントで使用できます。

9.3.7. トラブルシューティング

イメージを構築するには、いくつかの依存ステップが必要です。ビルトが失敗した場合は、Salt状態の結果とビルトログを調査することで、失敗の原因を特定できます。ビルトが失敗した場合は、次のチェックを実行できます。

- ・ 構築ホストがビルトソースにアクセスできる
- ・ 構築ホストとUyuniサーバの両方にイメージ用の十分なディスク容量がある
- ・ アクティベーションキーには正しいチャンネルが関連付けられている
- ・ 使用されるビルトソースが有効である
- ・ Uyuniのパブリック証明書を含むRPMパッケージは最新で`/usr/share/susemanager/salt/images/rhn-org-trusted-ssl-cert-osimage-1.0-1.noarch.rpm`から入手できます。パブリック証明書RPMを更新する方法の詳細については、[構築ホストの作成](#)を参照してください。

9.3.8. 制限事項

このセクションには、イメージを操作するときのいくつかの既知の問題が含まれています。

- ・ HTTPソースまたはgitリポジトリへのアクセスに使用されるHTTPS証明書は、カスタム状態ファイルによってクライアントに配備するか、手動で設定する必要があります。
- ・ Kiwiベースのイメージの取り込みはサポートされていません。

9.4. ビルドイメージのリスト

使用できるビルドイメージを一覧にするには、[イメージ > イメージリスト](#)を選択します。すべてのイメージのリストが表示されます。

Type	Name	Version	Revision	Updates	Patches	Packages	Build	Last Modified	Actions
Container Image	suse_key	latest	15	0	0	0	✓	21 hours ago	
OS Image	Building profile: suse_os_image	-	-	0	0	0	✗	2 days ago	
OS Image	POS_Image_JeOS7_uyuni	7.0.0	1	✓	0	0	✓	5 days ago	

Page 1 of 1

[Go to OS image directory listing](#)

イメージに関する表示されたデータには、イメージの[名前]、その[バージョン]、[リビジョン]、おビルドの[ステータス]が含まれます。イメージに使用できる可能性のあるパッチやパッケージの更新のリストを使用してイメージの更新ステータスを確認することもできます。

OSイメージの場合、[名前]および[バージョン]フィールドはkiwiソースから作成され、ビルドが成功したときに更新されます。ビルド中またはビルドが失敗した後は、これらのフィールドにはプロファイル名に基づく一時的な名前が表示されます。

[リビジョン]はビルドが成功するたびに自動的に増えます。OSイメージの場合、複数のリビジョンがストア内に共存できます。

コンテナイメージの場合、ストアには最新リビジョンのみが保持されます。以前のリビジョン(パッケージ、パッチなど)に関する情報は保存され、[非推奨の表示]チェックボックスを使用して一覧にすることができます。

イメージの[詳細]ボタンをクリックすると、詳細ビューが表示されます。詳細ビューには、関連するパッチの正確なリスト、イメージ内にインストールされたすべてのパッケージのリスト、およびビルドログが含まれます。

[削除]ボタンをクリックすると、リストからイメージが削除されま~~せ~~た、関連するピラー、OSイメージストアからのファイル、および非推奨のリビジョンも削除されます。



パッチおよびパッケージリストは、ビルド後の検査状態が正常だった場合にのみ使用できます。

Chapter 10. インフラストラクチャ保守タスク

スケジュールされたダウンタイム期間で作業する場合、Uyuniサーバの重要なダウンタイムの前、その最中、およびその後に行う必要があるすべての作業を覚えておくことが困難な場合があります。 サーバ間同期スレーブサーバやUyuniプロキシなどのUyuniサーバ関連のシステムも影響を受けるため、考慮する必要があります。

SUSEでは、常にUyuniインフラストラクチャを更新し続け続けることをお勧めします。 これには、サーバ、プロキシ、構築ホストが含まれます。 Uyuniサーバを更新し続けないと、必要な場合に環境の一部を更新できない場合があります。

このセクションには、ダウンタイム期間のチェックリストと、各ステップの実行に関する詳細情報へのリンクが含まれています。

10.1. サーバ

1. 最新の更新を適用します。 [Installation-and-upgrade > Server-intro](#)を参照してください。
2. 必要に応じて、最新のサービスパックにアップグレードします。
3. `spacewalk-service status` を実行し、必要なすべてのサービスが稼働しているかどうかを確認します。

データベーススキーマのアップグレードとPostgreSQLのマイグレーションについては、[Installation-and-upgrade > Db-intro](#)を参照してください。

パッケージマネージャを使用して更新をインストールできます。 YaSTの使用方法については、<https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-onlineupdate-you.html>を参照してください。 zypperの使用方法については、<https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-sw-cl.html#sec-zypper>を参照してください。

デフォルトでは、Uyuniサーバに対していくつかの更新チャンネルが設定され、有効になっています。 新規および更新されたパッケージは自動的に使用可能になります。

10.1.1. クライアントツール

サーバを更新する際には、クライアント上の一ツのツールも更新することを検討してください。 クライアント上で `salt-minion`、`zypper`、および他の関連する管理パッケージを更新することは厳密な要件ではありませんが、一般的にはベストプラクティスです。 たとえば、サーバの保守更新では、メジャーな新しいSaltバージョンが導入される場合があります。 その後、Saltクライアントは引き続き動作しますが、後で問題が発生する可能性があります。 これを回避するには、`salt-minion` パッケージが利用可能になったら常に更新してください。 SUSEは、`salt-minion` が常に安全に更新できるようにします。

10.2. サーバ間同期スレーブサーバ

サーバ間同期スレーブサーバを使用している場合は、Uyuniサーバ更新が完了した後で更新してください。

サーバ間同期の詳細については、[Administration > Iss](#)を参照してください。

10.3. モニタリングサーバ

Prometheusにモニタリングサーバを使用している場合は、Uyuniサーバの更新が完了した後で更新してください。

モニタリングの詳細については、[Administration > Monitoring](#)を参照してください。

10.4. プロキシ

プロキシは、Uyuniサーバの更新が完了したらすぐに更新する必要があります。

一般的に、別のバージョンのサーバに接続されたプロキシの実行はサポートされていません。 唯一の例外は、サーバが最初に更新されることが予想される更新期間の場合で、プロキシは以前のバージョンを一時的に実行できます。

特にバージョン4.0 から4.1に移行する場合は、最初にサーバをアップグレードしてから任意のプロキシをアップグレードしてください。

詳細については、[Installation-and-upgrade > Proxy-intro](#)を参照してください。

Chapter 11. サーバ間同期

複数のUyuniがインストールされている場合は、コンテンツと許可が必ず一致しているようにする必要があります。 サーバ間同期(ISS)を使用すると、複数のUyuniサーバを接続して、最新のままにすることができます。

ISSを設定するには、一方のUyuniサーバをマスターとして定義し、他方をスレーブとして定義する必要があります。 競合する設定が存在する場合は、システムがマスター設定を優先します。



ISSマスターはスレーブが接続されているという理由でのみマスターです。 これは、スレーブを定義して、最初にISSマスターを設定する必要があることを意味します。 その後、ISSスレーブをマスターに接続して設定できます。

プロシージャ: ISSマスターの設定

1. Uyuni Web UIで、**管理ターゲットセットアップ**に移動し、**[新規]** **[スレーブの追加]**をクリックします。
2. **[スレーブの詳細の編集]** ダイアログで、ISSマスターの最初のスレーブに関する次の詳細を提供します。
 - **[スレーブの完全修飾ドメイン名]** フィールドに、ISSスレーブのFQDNを入力します。`server2.example.com`。
 - **[スレーブの同期を許可しますか?]** チェックボックスをオンにして、スレーブがマスターとなるようにします。
 - **[すべての組織をスレーブに同期しますか?]** チェックボックスをオンにして、すべてのスレーブに同期します。
3. **[作成]**をクリックして、ISSスレーブを追加します。
4. **[選択された組織のエクスポートを許可します]** セクションで、このスレーブをマスターできる組織をオンにし、**[組織を許可]**をクリックします。

ISSスレーブを設定する前に、適切なCA証明書があることを確認する必要があります。

プロシージャ: マスターCA証明書をISSスレーブにコピーする

1. ISSマスターで、`/srv/www/htdocs/pub/RHN-ORG-TRUSTED-SSL-CERT`にあるCA証明書を見つけ、ISSスレーブに転送できるコピーを作成します。
2. ISSスレーブで、CA証明書ファイルを `/etc/pki/trust/anchors/` ディレクトリに保存します。

証明書をコピーしたら、ISSスレーブを設定できます。

プロシージャ: ISSスレーブの設定

1. Uyuni Web UIで、**管理ターゲットセットアップ**に移動し、**[新規]** **[マスターの追加]**をクリックします。
2. **[新規マスターの詳細]** ダイアログで、サーバがISSマスターとして使用するための詳細を提供します。
 - **[マスターの完全修飾ドメイン名]** フィールドに、このスレーブのISSマスターのFQDNを入れます。

す。例: `server1.example.com`。

このマスターのCA証明書のファイル名] フィールドに、ISSマスターのCA証明書への絶対パスを入力します。これは `/etc/pki/trust/anchors/RHN-ORG-TRUSTED-SSL-CERT` である必要があります。

3. [新規マスターの追加] をクリックし、このマスターにISSスレーブを追加します。

プロシージャ: ISSセットアップの完了

1. ISSスレーブのコマンドプロンプトで、ISSマスターと同期します。

```
mgr-inter-sync
```

2. オプション: 単一チャンネルを同期するには、次のコマンドを使用します。

```
mgr-inter-sync -c <channel-name>
```

3. Uyuni Web UIで、**管理 > ISS設定 > Configure Master-to-Slave Mappings** (マスターとスレーブのマッピングの設定) に移動して、同期する組織を選択します。

11.1. サーバ間同期 - バージョン2

複数のUyuniがインストールされている場合は、サーバ間でコンテンツをコピーする必要があります。サーバ間同期(ISS)を使用すると、あるサーバ(ソース)からデータをエクスポートし、別の(ターゲット)サーバにインポートできます。これは、ハブ配備シナリオや切断されたセットアップに役立ちます。



バージョン2のISS実装では、SUSEはマスター/スレーブの概念を削除しました。コンテンツは、Uyuniサーバ間で任意の方向にエクスポートおよびインポートできます。

11.1.1. ISSパッケージのインストール

ISSを使用するには、ソースサーバとターゲットサーバに `inter-server-sync` パッケージをインストールする必要があります。

11.1.2. コンテンツ同期

プロシージャ: ソースサーバへのデータのエクスポート

1. ソースサーバのコマンドラインで、ISSエクスポートコマンドを実行します。 `-h` オプションでは、詳細なヘルプを提供します。

```
inter-server-sync export -h
```

エクスポートプロシージャにより、インポートプロシージャに必要なデータをすべて含む出力ディレクトリが作成されます。

プロシージャ: エクスポートディレクトリをターゲットサーバにコピーする

- ソースサーバのコンテンツをターゲットサーバに同期する必要があります。 コマンドラインで、rootとして、次のコマンドを実行します。

```
rsync -r <PATH_EXPORTED_DIR> root@<TARGET_SERVER>:~/
```

すべてのコンテンツがコピーされたら、そのインポートを開始します。

プロシージャ: ターゲットサーバへのデータのインポート

- ターゲットサーバのコマンドラインで、ISSインポートコマンドを実行します。 **-h** オプションにより詳細なヘルプが提供されます。

```
inter-server-sync import -h
```

11.1.3. データベース接続設定

データベース接続設定は **/etc/rhn/rhn.conf** からデフォルトでロードされます。プロパティファイルの場所はパラメータ **--serverConfig** で上書きできます。

11.1.4. 既知の制限事項

- ソースサーバとターゲットサーバは同じバージョンである必要がある
- エクスポートとインポートの組織名は同じである必要がある

Chapter 12. SUSE Managerによるライブパッチ処理

カーネル更新を実行するには、通常、システムの再起動が必要です。共通脆弱性識別子(CVE)パッチはできるだけ早く適用する必要がありますが、ダウンタイムを許容できない場合は、ライブパッチ処理を使用してこれらの重要な更新を挿入し、再起動の必要性をスキップできます。

ライブパッチ処理を設定するプロシージャはSLES 12とSLES 15ではわずかに異なります。このセクションでは両方のプロシージャについて説明します。

12.1. ライブパッチ処理用のチャンネルの設定

完全なカーネルパッケージを更新するたびに再起動が必要です。したがって、ライブパッチ処理を使用しているクライアントは、割り当てられているチャンネルで新しいカーネルを使用できないことが重要です。ライブパッチ処理を使用しているクライアントは、ライブパッチ処理チャンネルで実行中のカーネルの更新を取得します。

ライブパッチ処理用のチャンネルを管理するには次の2つの方法があります。

コンテンツライフサイクル管理を使用して製品ツリーのクローンを作成し、実行中のバージョンより新しいカーネルバージョンを削除します。このプロシージャは、[content-lifecycle-examples.pdf](#)で説明されています。これは推奨される解決策です。

または、`spacewalk-manage-channel-lifecycle`ツールを使用します。このプロシージャはより手動であり、Web UIと同様にコマンドラインツールが必要です。このプロシージャはSLES 15 SP1のこのセクションで説明されていますが、SLE 12 SP4以降でも機能します。

12.1.1. ライブパッチ処理用のspacewalk-manage-channel-lifecycleを使用する

複製されたベンダチャンネルは、開発の場合は **dev**、運用の場合は **testing** または **prod** で始まる必要があります。このプロシージャでは **dev** クローンチャンネルを作成し、そのチャンネルを **testing** にプロモートさせます。

プロシージャ: ライブパッチ処理チャンネルの複製

1. クライアントのコマンドプロンプトで、rootとして、現在のパッケージチャンネルツリーを取得します。

```
# spacewalk-manage-channel-lifecycle --list-channels
Spacewalk Username: admin
Spacewalk Password:
Channel tree:

1. sles15-sp3-pool-x86_64
  \__ sle-live-patching15-pool-x86_64-sp3
  \__ sle-live-patching15-updates-x86_64-sp3
  \__ sle-manager-tools15-pool-x86_64-sp3
  \__ sle-manager-tools15-updates-x86_64-sp3
  \__ sles15-sp3-updates-x86_64
```

2. **spacewalk-manage-channel** コマンドに **init** 引数を指定して、元のベンダーチャンネルの新しい開発クローンを自動的に作成します。

```
spacewalk-manage-channel-lifecycle --init -c sles15-sp3-pool-x86_64
```

3. **dev-sles15-sp3-updates-x86_64** がチャンネルリストで使用できることを確認します。

作成した **dev** クローンチャンネルを確認し、再起動が必要なカーネル更新をすべて削除します。

プロシージャ: クローンチャンネルから非ライブカーネルパッチを削除する

1. システム一覧からクライアントを選択し、[カーネル] フィールドに表示されるバージョンをメモして、現在のカーネルバージョンを確認します。
2. Uyuni Web UIで、システム > 概要からクライアントを選択し、ソフトウェア > 管理 > チャンネルタブに移動 **dev-sles15-sp3-updates-x86_64** を選択し、[パッチ] タブに移動して、[パッチの一覧] [表示] [削除] をクリックします。
3. 検索バーに「カーネル」と入力し、クライアントが現在使用しているカーネルに一致するカーネルバージョンを特定します。
4. 現在インストールされているカーネルより新しいすべてのカーネルバージョンを削除します。

これでチャンネルにライブパッチ処理を適用するように設定され、**testing** にプロモートできるようになりました。このプロシージャでは、ライブパッチ処理の子チャンネルもクライアントに追加し、適用できるようになります。

プロシージャ: ライブパッチ処理チャンネルのプロモート

1. クライアントのコマンドプロンプトで、`root`として、`dev-sles15-sp3-pool-x86_64` チャンネルを新しい`testing` チャンネルにプロモートして複製します。

```
# spacewalk-manage-channel-lifecycle --promote -c dev-sles15-sp3-pool-x86_64
```

2. Uyuni Web UIで、**システム** > **概要**からクライアントを選択し、**ソフトウェア** > **ソフトウェアチャンネル**タブに移動します。
3. 新しい **test-sles15-sp3-pool-x86_64** カスタムチャンネルを確認してベースチャンネルを変更し、両方の対応するライブパッチ処理の子チャンネルを確認します。
4. **[次へ]** をクリックして、詳細が正しいことを確認して、**[確認]** をクリックして、変更を保存します。

これで、使用できるCVEパッチを選択して表示し、ライブパッチ処理でこれらの重要なカーネル更新を適用できるようになりました。

12.2. SLES 15でのライブパッチ処理

SLES 15以降のシステムでは、ライブパッチ処理は **klp livepatch** ツールで管理されます。

開始する前に、以下を確認します。

- ・ Uyuniが完全に更新されている。
- ・ SLES 15 (SP1以降)を実行している1つ以上のSaltクライアントがある。
- ・ SLES 15 SaltクライアントはUyuniに登録されている。
- ・ ライブパッチ処理の子チャンネルを含む、アーキテクチャに適したSLES 15チャンネルにアクセスできる。
- ・ クライアントが完全に同期されている。
- ・ クライアントをライブパッチ処理用に準備されているクローンチャンネルに割り当てる。 準備の詳細については、**Administration** > **Live-patching-channel-setup**を参照してください。

プロシージャ: ライブパッチ処理の設定

1. **システム** > **概要**からライブパッチ処理で管理するクライアントを選択し、**ソフトウェア** > **パッケージ** > **インストール**タブに移動します。 **kernel-livepatch** パッケージを検索して、インストールします。

Package Name	Architecture
kernel-livepatch-4_12_14-195-default-4-10.1	x86_64
<input checked="" type="checkbox"/> kernel-livepatch-4_12_14-197_10-default-1-3.3.1	x86_64
kernel-livepatch-4_12_14-197_4-default-3-2.1	x86_64
kernel-livepatch-4_12_14-197_7-default-2-2.1	x86_64
kernel-livepatch-tools-1.1-9.5	x86_64
kernel-livepatch-tools-devel-1.1-9.5	x86_64

2. highstateを適用してライブパッチ処理を有効にし、クライアントを再起動します。
3. ライブパッチ処理で管理するクライアントごとに繰り返します。
4. ライブパッチ処理が正しく有効化されていることを確認するには、**システム > システム一覧**からクライアントを選択し、[カーネル] フィールドに [ライブパッチ] が表示されていることを確認します。

プロシージャ: ライブパッチのカーネルへの適用

1. Uyuni Web UIで、**システム > 概要**からクライアントを選択します。画面の上部のバナーに、クライアントに使用できる重要なパッケージ数と、重要ではないパッケージ数が表示されます。

2. [重大] をクリックすると、使用可能な重大なパッチのリストが表示されます。
3. [Important: Security update for the Linux kernel] (重要: Linuxカーネル用のセキュリティ更新) という概要のパッチを選択します。セキュリティバグには該当する場合はCVE番号も含まれます。
4. オプション: 適用するパッチのCVE番号がわかっている場合は、**監査 > CVE監査**で検索し、必要なクライアントにパッチを適用します。



すべてのカーネルパッチがライブパッチであるわけではありません。非ライブカーネルパッチは [セキュリティアイコン] の横にある [要再起動] アイコンで示されます。これらのパッチでは常に再起動が必要です。



ライブパッチを適用することで、すべてのセキュリティ問題を修正できるわけではありません。一部のセキュリティ問題は、カーネルの完全な更新を適用することによってのみ修正でき、再起動が必要です。これらの問題に割り当てられたCVE番号は、ライブパッチには含まれていません。CVE監査では、この要件が表示されます。

12.3. SLES 12でのライブパッチ処理

SLES 12システムでは、ライブパッチ処理はkGraftで管理されます。kGraftの使用に関する詳細については、<https://documentation.suse.com/sles/12-SP4/html/SLES-all/cha-kgraft.html>を参照してください。

開始する前に、以下を確認します。

- Uyuniが完全に更新されている。
- SLES 12 (SP1以降)を実行している1つ以上のSaltクライアントがある。
- SLES 12 SaltクライアントがUyuniに登録されている。
- ライブパッチ処理の子チャンネルを含む、アーキテクチャに適したSLES 12チャンネルにアクセスできる。
- クライアントが完全に同期されている。
- クライアントをライブパッチ処理用に準備されているクローンチャンネルに割り当てる。準備の詳細については、[Administration > Live-patching-channel-setup](#)を参照してください。

プロシージャ: ライブパッチ処理の設定

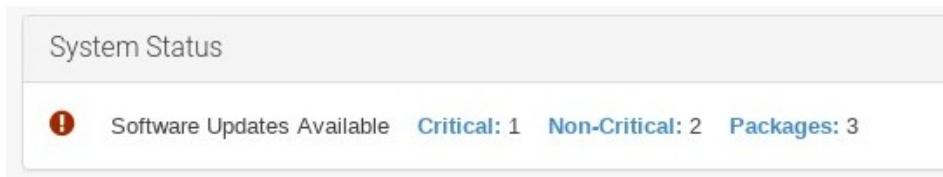
- システム > 概要**からライブパッチ処理を使用して管理するクライアントを選択し、システムの詳細ページで、**ソフトウェア > パッケージ > インストール**タブに移動します。**kgraft**パッケージを検索して、インストールします。

Package Name	Architecture
kgraft-1.0-22.1	x86_64
kgraft-devel-1.0-22.1	x86_64
kgraft-manual_en-12.8.2	noarch
kgraft-patch-3_12_32-25-default-1.2.7	x86_64
kgraft-patch-3_12_32-25-xen-1.2.7	x86_64

- highstateを適用してライブパッチ処理を有効にし、クライアントを再起動します。
- ライブパッチ処理で管理するクライアントごとに繰り返します。
- ライブパッチ処理が正常に有効化されていることを確認するには、**システム > システム一覧**からクライアントを選択し、[ライブパッチ処理]が[カーネル]フィールドに表示されていることを確認します。

プロセッジャ: ライブパッチのカーネルへの適用

1. Uyuni Web UIで、[システム > 概要](#)からクライアントを選択します。画面の上部のバナーに、クライアントに使用できる重要なパッケージ数と、重要ではないパッケージ数が表示されます。



2. **[重]** をクリックすると、使用可能な重大なパッチのリストが表示されます。
3. **[Important: Security update for the Linux kernel]** (重要: Linuxカーネル用のセキュリティ更新) という概要のパッチを選択します。セキュリティバグには該当する場合はCVE番号も含まれます。
4. オプション: 適用するパッチのCVE番号がわかっている場合は、[監査 > CVE監査](#)で検索し、必要なクライアントにパッチを適用します。



すべてのカーネルパッチがライブパッチであるわけではありません。非ライブカーネルパッチは [セキュリティ] の横にある [要再起動] アイコンで示されます。これらのパッチでは常に再起動が必要です。



ライブパッチを適用しても、すべてのセキュリティ問題を修正できるわけではありません。一部のセキュリティの問題は、カーネルの完全な更新を適用することでのみ修正でき、再起動が必要です。これらの問題に割り当てられたCVE番号は、ライブパッチには含まれていません。CVE監査ではこの要件が表示されます。

Chapter 13. メンテナスウィンドウ

Uyuniのメンテナスウィンドウ機能を使用すると、スケジュールされたメンテナスウィンドウ期間中にアクションを実行するようにスケジュールできます。メンテナスウィンドウスケジュールを作成してクライアントに適用すると、指定した期間外に一部のアクションを実行できなくなります。



メンテナスウィンドウは、システムロックとは異なる方法で動作します。システムロックは必要に応じてオンまたはオフに切り替えられますが、メンテナスウィンドウではアクションを許可する期間が定義されます。また、許可されたアクションと制限されたアクションが異なります。システムロックの詳細については、[Client-configuration > System-locking](#)を参照してください。

メンテナスウィンドウには、カレンダーとスケジュールの両方が必要です。カレンダーは、定期的なイベントを含むメンテナスウィンドウイベントの日付と時刻を定義し、`ical`形式にする必要があります。スケジュールは、カレンダーで定義されたイベントを使用して、メンテナスウィンドウを作成します。スケジュールを作成する前に、アップロード用の`ical`ファイルを作成するか、`ical`ファイルにリンクしてカレンダーを作成する必要があります。

スケジュールを作成したら、Uyuniサーバに登録されているクライアントに割り当てることができます。メンテナススケジュールが割り当てられているクライアントは、メンテナスウィンドウ以外に制限されたアクションを実行できません。

制限されたアクションによってクライアントが大幅に変更され、クライアントの実行を停止させる可能性があります。制限されたアクションの例は次のとおりです。

- ・ パッケージのインストール
- ・ クライアントのアップグレード
- ・ 製品の移行
- ・ highstateアプリケーション(Saltクライアント用)

制限されたアクションは、安全であるとみなされ、クライアントに問題が発生する可能性が低いマイナーアクションです。制限されたアクションの例は次のとおりです。

- ・ パッケージプロファイルの更新
- ・ ハードウェアの更新
- ・ ソフトウェアチャンネルのサブスクライブ

開始する前に、アップロード用の`ical`ファイルを作成するか、`ical`ファイルにリンクしてカレンダーを作成する必要があります。`ical`ファイルは、Microsoft Outlook、Google Calendar、KOrganizerなどのお好みのカレンダーツールで作成できます。

プロシージャ: 新しいメンテナスカレンダーのアップロード

1. UyuniWeb UIで、[スケジュール > メンテナスウィンドウ > カレンダー](#)に移動し、[\[新規\]](#)をクリックします。

- [カレンダー名] セクションに、カレンダーの名前を入力します。
- ical ファイルへのURLを指定するか、ファイルを直接アップロードします。
- [Create Calendar] (カレンダーの作成) をクリックして、カレンダーを保存します。

プロシージャ: 新しいスケジュールの作成

- Uyuni Web UIで、スケジュールメンテナンスウィンドウ、スケジュールに移動して、[作成]をクリックします。
- [スケジュール名] セクションに、スケジュールの名前を入力します。
- オプシカル: ファイルに複数のスケジュールに適用するイベントが含まれている場合は、[マルチ] をオンにします。
- このスケジュールに割り当てるカレンダーを選択します。
- [スケジュールの作成]をクリックして、スケジュールを保存します。

プロシージャ: スケジュールをクライアントに割り当てる

- Uyuni Web UIで、システム > システム一覧に移動して、スケジュールに割り当てるクライアントを選択 [System Properties] (システムのプロパティ) パネルを見つけて、[これら(の)プロパティ]を編集します。または、システムセットマネージャに移動し、その他メンテナスウィンドウタブを使用して、システムセットマネージャからクライアントを割り当てることができます。
- [システムの詳細を編集] ページで、[メンテナンススケジュール] フィールドをスケジュールの名前を選択します。
- [プロパティの更新]をクリックし、メンテナンススケジュールを割り当てます。



新しいメンテナンススケジュールをクライアントに割り当てるとき、クライアントですでにいくつかの制限されたアクションがスケジュールされている可能性があり、これらが新しいメンテナンススケジュールと競合する可能性があります。この場合は、Web UIにエラーが表示され、スケジュールをクライアントに割り当てることができません。この問題を解決するには、スケジュールを割り当てるときに、[影響する動作のキャンセル]オプションを有効にして新規メンテナンススケジュールと競合する、以前にスケジュールされた動作がすべてキャンセルされます。

メンテナンスウィンドウを作成したら、メンテナンスウィンドウ中に実行される、パッケージの更新などの制限されたアクションをスケジュールできます。

プロシージャ: パッケージのアップグレードのスケジュール

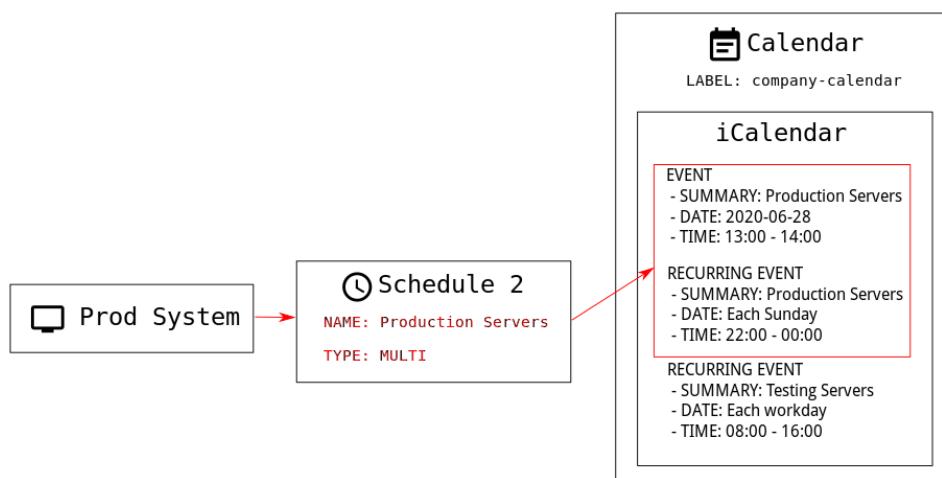
- Uyuni Web UIで、システム > システム一覧に移動して、アップグレードするクライアントを選択し、ソフトウェア > パッケージ > アップグレードタブに移動します。
- リストからアップグレードするパッケージを選択し、[パッケージのアップグレード]をクリックします。
- [guimenu] メンテナンス] ワイドドロップ、クライアントがアップグレードの実行に使用するメンテナンスウィンドウを選択します。

4. [確認] をクリックして、パッケージのアップグレードをスケジュールします。

13.1. メンテナンススケジュールタイプ

カレンダーを作成すると、1回限りのイベントまたは繰り返し発生するイベントのいずれかを含むイベントが多数含まれます。各イベントには [概要] フィールドが含まれます。1つのカレンダーに対して複数のメンテナンススケジュールを作成する場合は、[概要] フィールドを使用して、それぞれのイベントを指定できます。

たとえば、運用サーバのスケジュールを作成し、テストサーバに別のスケジュールを作成したい場合があります。この場合、運用サーバのイベントには **SUMMARY: Production Servers** を指定し、テストサーバのイベントには **SUMMARY: Testing Servers** を指定します。



スケジュールには、シングルとマルチの2つのタイプがあります。 カレンダーに複数のスケジュールに適用されるイベントが含まれている場合は [マルチ]を選択し、カレンダーファイルで使用した [概要] フィールドに従ってスケジュールに名前を付ける必要があります。

プロセッジヤ: マルチスケジュールの作成

1. UyuniWeb UIで、**スケジュール** > **メンテナンス** ウィンドウ > **スケジュール** に移動して、[作成] をクリックします。
2. [スケジュール名] セクションに、スケジュールの名前を入力または [概要] フィールドに一致していることを確認します。
3. [マルチ] オプションをオンにします。
4. このスケジュールに割り当てるカレンダーを選択します。
5. [スケジュールの作成] をクリックして、スケジュールを保存します。
6. 次のスケジュールを作成するには、[作成] をクリックします。
7. [スケジュール名] セクションに、2番目のスケジュールの名前を入力または [概要] フィールドに一致していることを確認します。
8. [マルチ] オプションをオンにします。
9. [スケジュールの作成] をクリックして、スケジュールを保存します。

10. 作成する必要がある各スケジュールごとに繰り返します。

13.2. 制限されたアクションと制限されないアクション

このセクションには、制限されたアクションと制限されないアクションのリスト全体が含まれています。

制限されたアクションにより、クライアントが大幅に変更され、クライアントの実行が停止する可能性があります。制限されたアクションは、メンテナンスウィンドウ中にのみ実行できます。制限されたアクションは次のとおりです。

- ・ パッケージ操作(たとえば、パッケージのインストール、更新、または削除)
- ・ パッチの更新
- ・ クライアントの再起動
- ・ トランザクションのロールバック
- ・ 設定管理の変更タスク
- ・ highstateの適用(Saltクライアント用)
- ・ 自動インストールと再インストール
- ・ リモートコマンド
- ・ 製品移行
- ・ クラスター操作



Saltクライアントの場合は、[Salt > リモートコマンド](#)に移動することで、いつでもリモートコマンドを直接実行できます。これは、Saltクライアントがメンテナンスウィンドウ中であるかどうかに関係なく適用されます。リモートコマンドの詳細については、[Administration > Actions](#)を参照してください。

制限されないアクションは、安全であるとみなされ、クライアントに問題が発生する可能性が少ないマイナーアクションです。アクションが制限されない場合、定義では、無制限であり、いつでも実行できます。

Chapter 14. `mgr-sync` の使用

`mgr-sync` ツールは、コマンドプロンプトで使用されます。Web UIでは必ずしも使用できるとは限らないUyuniを使用するための機能を提供します。`mgr-sync` の主な用途は、SUSE Customer Centerへの接続、製品およびパッケージ情報の取得、Uyuniサーバと同期するためのチャンネルの準備です。

このツールは、SUSEサポートサブスクリプションで使用するように設計されています。
openSUSE、CentOS、Ubuntuなどのオープンソースディストリビューションには必要ありません。

`mgr-sync` に使用できるコマンドおよび引数を以下の表に一覧表示しています。`mgr-sync` コマンドには次の構文を使用します。

```
mgr-sync [-h] [--version] [-v] [-s] [-d {1,2,3}]  
{list,add,refresh,delete}
```

表 2. `mgr-sync` コマンド

コマンド	説明	使用例
list	チャンネル、組織の資格情報、または製品を一覧表示する	<code>mgr-sync list channels</code>
add	チャンネル、組織の資格情報、または製品を追加する	<code>mgr-sync add channel <channel_name></code>
refresh	製品、チャンネル、およびサブスクリプションのローカルコピーを更新する	<code>mgr-sync refresh</code>
delete	ローカルシステムから既存のSCC組織の資格情報を削除する	<code>mgr-sync delete credentials</code>
sync	指定されたチャンネルを同期するか、空白のままの場合は尋ねる	<code>mgr-sync sync channel <channel_name></code>

コマンドに固有のオプションの全リストを表示するには、次のコマンドを使用します。

```
mgr-sync <command> --help
```

表 3. `mgr-sync` オプション引数

オプション	短縮オプション	説明	使用例
help	<code>h</code>	コマンドの使用方法とオプションを表示する	<code>mgr-sync --help</code>
version	N/A	現在インストールされているバージョンの <code>mgr-sync</code> を表示する	<code>mgr-sync --version</code>

オプション	短縮オプション	説明	使用例
verbose	v	詳細な出力を提供する	<code>mgr-sync --verbose refresh</code>
store-credentials	s	資格情報をローカルの隠しファイルに保存する	<code>mgr-sync --store-credentials</code>
debug	d	追加のデバッグ情報をログに記録する。レベル1、2、3が必要です。3は、最も多くのデバッグ情報を提供します。	<code>mgr-sync -d 3 refresh</code>
no-sync	N/A	<code>add</code> コマンドと一緒に使用して、同期を開始せずに製品またはチャンネルを追加する	<code>mgr-sync --no-sync add <channel_name></code>

`mgr-sync` のログは次の場所にあります。

- `/var/log/rhn/mgr-sync.log`
- `/var/log/rhn/rhn_web_api.log`

Chapter 15. PrometheusとGrafanaを使用したモニタリング

PrometheusとGrafanaを使用して、Uyuni環境を監視できます。Uyuniサーバとプロキシはself-health metricsを提供できます。また、Saltクライアント上に多数のPrometheus exportersをインストールして管理することもできます。

PrometheusとGrafanaパッケージは以下のUyuniクライアントツールに含まれています。

- SUSE Linux Enterprise 12
- SUSE Linux Enterprise 15
- openSUSE Leap 15.x

Uyuniサーバとは別のマシンにPrometheusとGrafanaをインストールする必要があります。管理対象のSalt SUSEクライアントをモニタリングサーバとして使用することをお勧めします。他のクライアントはモニタリングサーバとしてサポートされていません。

Prometheusはプルメカニズムを使用してメトリックをフェッチするため、サーバは監視対象クライアントへのTCP接続を確立できる必要があります。クライアントには対応するオープンポートがあり、ネットワークを介して到達できる必要があります。または、リバースプロキシを使用して接続を確立することもできます。

15.1. PrometheusとGrafana

15.1.1. Prometheus

Prometheusはオープンソースのモニタリングツールで、リアルタイムメトリックを時系列データベースに記録するために使用されます。メトリックはHTTP経由でプルされるため、高いパフォーマンスと拡張性が実現します。

Prometheusメトリックは時系列データ、または同じグループやディメンションに属するタイムスタンプ付きの値です。メトリックはその名前とラベルセットで固有に識別されます。

metric name	labels	timestamp	value
http_requests_total{status="200", method="GET"}		@1557331801.111	42236

監視対象の各アプリケーションまたはシステムは、コードインストルメンテーションまたはPrometheus exportersを使用して、上記の形式でメトリックを公開する必要があります。

15.1.2. Prometheus Exporters

Exportersは、サードパーティシステムからメトリックをPrometheusメトリックとしてエクスポートするの

に役立つライブラリです。 Exportersは、Prometheusメトリックを直接使用して特定のアプリケーションまたはシステムをインストルメントできない場合に役立ちます。 複数のexportersを監視対象ホスト上で実行して、ローカルメトリックをエクスポートできます。

Prometheusコミュニティは公式exportersのリストを提供し、さらに多くのものをコミュニティの貢献として見つけることができます。
詳細およびexportersの詳細なリストについては、<https://prometheus.io/docs/instrumenting/exporters/>を参照してください。

15.1.3. Grafana

Grafanaは、データの視覚化、モニタリング、および分析を行うためのツールです。 一定期間の特定のメトリックを表すパネルを使用してダッシュボードを作成するために使用されます。 Grafanaは通常、Prometheusと一緒に使用されますが、Elasticsearch、MySQL、PostgreSQL、Influx DBなどの他のデータソースもサポートしています。 Grafanaの詳細については、<https://grafana.com/docs/>を参照してください。

15.2. モニタリングサーバの設定

モニタリングサーバを設定するには、PrometheusとGrafanaをインストールして設定する必要があります。

15.2.1. Prometheusのインストール

モニタリングサーバがSaltクライアントの場合、Uyuni Web UIを使用してPrometheusパッケージをインストールできます。 インストールできない場合は、モニタリングサーバに手動でパッケージをダウンロードしてインストールできます。 Prometheusソフトウェアは、UyuniプロキシおよびUyuni for Retailブランチサーバでも使用できます。



Prometheusはデータの保存にPOSIXファイルシステムを想定しています。
非POSIX準拠ファイルシステムはサポートされていません。 NFSファイルシステムは明示的にサポートされていません。

プロシージャ: Web UIを使用したPrometheusのインストール

1. Uyuni Web UIで、Prometheusをインストールするシステムの詳細ページを開き、[Formula] タブに移動します。
2. [Prometheus] チェックボックスをオンにしてエニタリング式を有効にし、[保] をクリックします。
3. 上部メニューの [Prometheus] タブに移動します。
4. [Uyuni サーバ] セクションに、有効なUyuni API資格情報を入力します。入力した資格情報でモニタリングするシステムセットにアクセスできることを確認してください。
5. 必要に応じて、他の設定オプションをカスタマイズします。
6. [Save Formula] (Formulaの保存) をクリックします。
7. highstateを適用し、正常に完了したことを確認します。
8. Prometheusインターフェースが正しくロードされることを確認します。ブラウザで、ポート9090で、PrometheusがインストールされているサーバのURLに移動します(たとえ

ば、<http://example.com:9090>）。

モニタリング式の詳細については、[Specialized-guides > Salt](#)を参照してください。

プロシージャ: Prometheusを手動でインストールおよび設定する

- モニタリングサーバで、`golang-github-prometheus-prometheus`パッケージをインストールします。

```
zypper in golang-github-prometheus-prometheus
```

- Prometheusサービスを有効にします。

```
systemctl enable --now prometheus
```

- Prometheusインターフェース が正しくロードされていることを確認します。 ブラウザで、ポート9090で、PrometheusがインストールされているサーバのURLに移動します(たとえば、<http://example.com:9090>)。

- `/etc/prometheus/prometheus.yml`にある設定ファイルを開いて、この設定情報を追加します。

`server.url` をUyuni サーバ のURI で置き換え、Uyuni 資格情報に一致するようにスワード`] フィールドを調整します。

```
# {productname} self-health metrics
scrape_configs:
- job_name: 'mgr-server'
  static_configs:
    - targets:
        - 'server.url:9100' # Node exporter
        - 'server.url:9187' # PostgreSQL exporter
        - 'server.url:5556' # JMX exporter (Tomcat)
        - 'server.url:5557' # JMX exporter (Taskomatic)
        - 'server.url:9800' # Taskomatic
    - targets:
        - 'server.url:80' # Message queue
  labels:
    __metrics_path__: /rhn/metrics

# Managed systems metrics:
- job_name: 'mgr-clients'
  uyuni_sd_configs:
    - host: "http://server.url"
      username: "admin"
      password: "admin"
```

5. 設定ファイルを保存します。
6. Prometheusサービスを再起動します。

```
systemctl restart prometheus
```

Prometheus設定オプションの詳細について
は、<https://prometheus.io/docs/prometheus/latest/configuration/configuration/>にある公式Prometheus
ドキュメントを参照してください。

15.2.2. Grafanaのインストール

モニタリングサーバがSaltクライアントの場合は、Uyuni Web UIを使用してGrafanaパッケージをインストールできます。インストールできない場合は、モニタリングサーバにパッケージを手動でダウンロードしてインストールできます。



GrafanaはUyuniプロキシでは使用できません。

プロシージャ: Web UIを使用したGrafanaのインストール

1. Uyuni Web UIで、Grafanaがインストールされるシステムの詳細ページを開き、[Formula] タブに移動します。
2. [Grafana] チェックボックスをオンにして、モニタリング式を有効にし、[保存] をクリックします。
3. 上部メニューの [Grafana] タブに移動します。
4. [Enable and configure Grafana] (Grafanaの有効化と設定) セクションに、Grafanaにログインするために使用する管理者資格情報を入力します。
5. [Datasources] (データソース) セクションで、Prometheus URLフィールドがPrometheusが実行されているシステムを指していることを確認します。
6. 必要に応じて、他の設定オプションをカスタマイズします。
7. [Save Formula] (Formulaの保存) をクリックします。
8. highstateを適用し、正常に完了したことを確認します。
9. Grafanaインターフェースが正しくロードされていることを確認します。ブラウザで、ポート3000で、GrafanaがインストールされているサーバのURLに移動します(たとえば、<http://example.com:3000>)。



Uyuniには、サーバのセルフヘルス、基本的なクライアントモニタリングなどのためのダッシュボードがあらかじめ構築されています。プロビジョニングするダッシュボードは、式の設定ページで選択できます。

モニタリング式の詳細については、[Specialized-guides > Salt](#)を参照してください。

プロシージャ: Grafanaの手動インストール

1. **grafana** パッケージをインストールします。

```
zypper in grafana
```

2. Grafanaサービスを有効にします。

```
systemctl enable --now grafana-server
```

3. Grafanaインターフェースが正しくロードされていることを確認します。 ブラウザで、ポート3000で、GrafanaがインストールされているサーバのURLに移動します(たとえば、<http://example.com:3000>)。



Grafanaを手動でインストールおよび設定する方法の詳細については、<https://grafana.com/docs>を参照してください。

フォームがあるモニタリング式の詳細については、[Specialized-guides > Salt](#)を参照してください。

15.3. Uyuniのモニタリングの設定

Uyuni 4以降では、サーバがPrometheus self-health metricsを公開できるようにして、クライアントシステムにexportersをインストールして設定することもできます。

15.3.1. サーバ自己監視

Server self-health metricsは、ハードウェア、オペレーティングシステム、およびUyuniの内部を対象としています。これらのメトリックは、Prometheus exportersと組み合わされた、Javaアプリケーションのインストルメンテーションによって利用可能になります。

以下のexporterパッケージはUyuniサーバに同梱されています。

- Node exporter: **golang-github-prometheus-node_exporter**。 <https://github.com/prometheus/>

[node_exporter](#)を参照してください。

- PostgreSQL exporter: [prometheus-postgres_exporter](#)。 https://github.com/wrouesnel/postgres_exporterを参照してください。
- JMX exporter: [prometheus-jmx_exporter](#)。 https://github.com/prometheus/jmx_exporterを参照してください。
- Apache exporter: [golang-github-lusitaniae-apache_exporter](#)。 https://github.com/Lusitaniae/apache_exporterを参照してください。

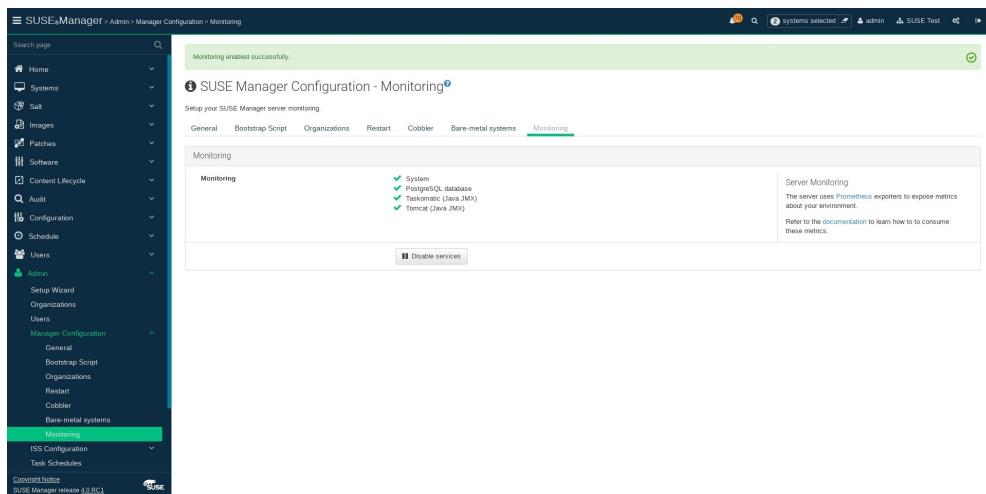
以下のexporterパッケージはUyuniプロキシに同梱されています。

- Node exporter: [golang-github-prometheus-node_exporter](#)。 https://github.com/prometheus/node_exporterを参照してください。
- Squid exporter: [golang-github-boynux-squid_exporter](#)。 <https://github.com/boynux/squid-exporter>を参照してください。

exporterパッケージはUyuniサーバおよびプロキシに事前インストールされていますが、各systemdサービスはデフォルトで無効になっています。

プロシージャ: 自己監視の有効化

- Uyuni Web UIで、**管理** > **マネージャ設定** > **モニタリング**に移動します。
- [**Enable services**] (サービスの有効化) をクリックします。
- TomcatとTaskomaticを再起動します。
- ポート9090で、PrometheusサーバのURLに移動します(たとえば、<http://example.com:9090>)。
- PrometheusUIで、menu:[ステータス>Targets (ターゲット)]に移動して、**mgr-server**グループのすべてのエンドポイントが起動していることを確認します。
- Web UIを使用してGrafanaもインストールしている場合、サーバインサイトがUyuniサーバダッシュボードに表示されます。





Web UIを使用して有効にできるのは、サーバのセルフヘルスモニタリングだけです。プロキシのメトリックは、Prometheusによって自動的に収集されません。プロキシでセルフヘルスモニタリングを有効にするには、exportersを手動でインストールして有効にする必要があります。

以下の関連メトリックがUyuniサーバに収集されます。

表 4. PostgreSQL exporter (ポート9187)

メトリック	タイプ	説明
pg_stat_database_tup_fetched	カウンタ	クエリによってフェッチされた行数
pg_stat_database_tup_inserted	カウンタ	クエリによって挿入された行数
pg_stat_database_tup_updated	カウンタ	クエリによって更新された行数
pg_stat_database_tup_deleted	カウンタ	クエリによって削除された行数
mgr_serveractions_completed	ゲージ	完了したアクションの数
mgr_serveractions_failed	ゲージ	失敗したアクションの数
mgr_serveractions_picked_up	ゲージ	ピックアップされたアクションの数
mgr_serveractions_queued	ゲージ	キューに入れられたアクションの数

表 5. JMX exporter (Tomcatポート5556、Taskomaticポート5557)

メトリック	タイプ	説明
java_lang_Threading_ThreadCount	ゲージ	アクティブなスレッドの数
java_lang_Memory_HeapMemoryUsage_used	ゲージ	現在のヒープメモリ使用量

表 6. サーバメッセージキュー(ポート80)

メトリック	タイプ	説明
message_queue_thread_pool_threads	カウンタ	これまでに作成されたメッセージキュースレッドの数
message_queue_thread_pool_threads_active	ゲージ	現在アクティブなメッセージキュースレッドの数
message_queue_thread_pool_task_count	カウンタ	これまでに送信されたタスクの数
message_queue_thread_pool_completed_tasks_count	カウンタ	これまでに完了したタスクの数

表 7. Taskomaticスケジューラ(ポート9800)

メトリック	タイプ	説明
taskomatic_scheduler_threads	カウンタ	これまでに作成されたスケジューラスレッドの数
taskomatic_scheduler_threads_active	ゲージ	現在アクティブなスケジューラスレッドの数

メトリック	タイプ	説明
taskomatic_scheduler_completed_task_count	カウンタ	これまでに完了したタスクの数

15.3.2. 管理対象システムの監視

Prometheus metrics exportersは、式を使用してSaltクライアントにインストールおよび設定できます。 パッケージはUyuniクライアントツールチャンネルから入手でき、Uyuni Web UIで直接有効化および設定できます。

これらのexportersは管理対象システムにインストールできます。

- Node exporter: **golang-github-prometheus-node_exporter**。 https://github.com/prometheus/node_exporterを参照してください。
- PostgreSQL exporter: **prometheus-postgres_exporter**。 https://github.com/wrouesnel/postgres_exporterを参照してください。
- Apache exporter: **golang-github-lusitaniae-apache_exporter**。 https://github.com/Lusitaniae/apache_exporterを参照してください。

exportersをインストールして設定したら、Prometheusを使用して監視対象システムからメトリックを収集できます。 Web UIを使用してモニタリングサーバを設定している場合、メトリック収集は自動的に行われます。

プロシージャ: クライアントでのPrometheus Exportersの設定

- Uyuni Web UIで、監視対象のクライアントの詳細ページを開き、menu:Formulaタブに移動します。
- [Prometheus Exporters] formulaで、[有効] チェックボックスをオンにします。
- [保存] をクリックします。
- Formula > Prometheus Exportersタブに移動します。
- 有効にするexportersを選択し、必要に応じて引数をカスタマイズします。[アドレス] フィールドは、コロンで始まるポート番号(:9100)か、完全に解決可能なアドレス(example:9100)のいずれかを受け入れます。
- [Save Formula] (Formulaの保存) をクリックします。
- highstateを適用します。



モニタリング式は、対応するグループ内の個々のシステムに使用されているのと同じ設定を適用することによって、システムグループにも設定できます。

モニタリング式の詳細については、[Specialized-guides > Salt](#)を参照してください。

15.3.3. Grafanaパスワードの変更

Grafana/パスワードを変更するには、Grafanaのドキュメントに記載されているステップに従います。

- <https://grafana.com/docs/grafana/latest/administration/user-management/user-preferences/#change-your-grafana-password>

Grafana管理者パスワードを紛失した場合は、次のコマンドで **root** としてリセットできます。

```
grafana-cli --configOverrides cfg:default.paths.data=/var/lib/grafana
--homepath /usr/share/grafana admin reset-admin-password <new_password>
```

15.4. ネットワーク境界

Prometheusはプルメカニズムを使用してメトリックをフェッチするため、サーバは監視対象クライアントとのTCP接続を確立できる必要があります。デフォルトでは、Prometheusは次のポートを使用します。

- Node exporter: 9100
- PostgreSQL exporter: 9187
- Apache exporter: 9117

また、Prometheusを実行するホストとは異なるホストでアラートマネージャを実行している場合は、ポート9093も開く必要があります。

クラウドインスタンスにインストールされているクライアントの場合、モニタリングサーバにアクセスできるセキュリティグループに必要なポートを追加できます。

または、Prometheusインスタンスをexportersのローカルネットワークにデプロイし、フェデレーションを設定することもできます。これにより、メインのモニタリングサーバはローカルのPrometheusインスタンスから時系列をスクレイピングできます。この方法を使用する場合は、Prometheus APIポート(9090)を開くだけで済みます。

Prometheusフェデレーションの詳細について

は、<https://prometheus.io/docs/prometheus/latest/federation/>を参照してください。

ネットワーク境界から要求をプロキシすることもできます。PushProxのようなツールは、プロキシとクライアントをネットワークバリアの両側に配備し、PrometheusがNATなどのネットワークトポロジ全体で動作できるようにします。

PushProxの詳細については、<https://github.com/RobustPerception/PushProx>を参照してください。

15.4.1. リバースプロキシのセットアップ

Prometheusはプルメカニズムを使用してメトリックをフェッチするため、サーバは監視対象クライアント上の各exporterへのTCP接続を確立できる必要があります。ファイアウォール設定を簡素化するため、exportersにリバースプロキシを使用して、単一のポートですべてのメトリックを公開できます。

プロシージャ: リバースプロキシを使用したPrometheus Exportersのインストール

1. Uyuni Web UIで、監視対象のシステムの詳細ページを開き、[Formula] タブに移動します。
2. [Prometheus Exporters] チェックボックスをオンにして、exporter formulaを有効にし、[保存] をクリックします。
3. 上部メニューの [Prometheus Exporters] タブに移動します。
4. [Enable reverse proxy] (リバースプロキシを有効にする) オプションをオンにして、有効なリバースプロキシポート番号を入力します。たとえば、9999。
5. 必要に応じて他のexportersをカスタマイズします。
6. [Save Formula] (Formulaの保存) をクリックします。
7. highstateを適用し、正常に完了したことを確認します。

モニタリング式の詳細については、[Specialized-guides > Salt](#)を参照してください。

15.5. セキュリティ

PrometheusサーバとPrometheus node exporterは、TLS暗号化と認証でエンドポイントを保護するための組み込みメカニズムを提供します。Uyuni Web UIは、関連するすべてのコンポーネントの設定を簡素化します。TLS証明書は、ユーザが提供して配備する必要があります。Uyuniでは、次のセキュリティモデルを有効にします。

- Node exporter: TLS暗号化とクライアント証明書ベースの認証
- Prometheus: TLS暗号化と基本認証

利用可能なすべてのオプションの設定の詳細については、[Specialized-guides > Salt](#)を参照してください。

15.5.1. TLS証明書の生成

デフォルトでは、Uyuniはモニタリング設定を保護するための証明書を提供しません。 セキュリティを提供するために、自己署名された、またはサードパーティの認証局(CA)によって署名されたカスタム証明書を生成またはインポートできます。

このセクションでは、SUSE Manager CAで自己署名したPrometheusおよびNode exporter minionsのクライアント/サーバ証明書を生成する方法について説明します。

プロシージャ: サーバ/クライアントTLS証明書の作成

1. Uyuniサーバのコマンドプロンプトで、次のコマンドを実行します。

```
rhn-ssl-tool --gen-server --dir="/root/ssl-build" --set
  -country="COUNTRY" \
  --set-state="STATE" --set-city="CITY" --set-org="ORGANIZATION" \
  --set-org-unit="ORGANIZATION UNIT" --set-email="name@example.com" \
  --set-hostname="minion.example.com" --set-cname="minion.example.com"
  --no-rpm
```

set-cname パラメータが、Saltクライアントの完全修飾ドメイン名(FQDN)であることを確認します。複数のエイリアスが必要な場合は、**set-cname** パラメータを複数回使用できます。

2. **server.crt** および **server.key** ファイルをSaltminionにコピーし、**prometheus** ユーザに読み取りアクセスを提供します。

Chapter 16. 組織

組織は、Uyuni内でユーザのアクセスと許可を管理するために使用されます。

ほとんどの環境では、単一の組織で十分です。ただし、より複雑な環境では、複数の組織が必要な場合があります。ビジネス内の物理的な場所ごとに、またはさまざまなビジネス機能ごとに組織を設定したい場合があります。

組織を作成したら、ユーザを作成して組織に割り当てることができます。次に、組織レベルで許可を割り当てるすることができます。この許可は、組織に割り当てられたすべてのユーザにデフォルトで適用されます。

PAMやシングルサインオンなど、新しい組織の認証方法を設定することもできます。認証の詳細については、Administration > Auth-methodsを参照してください。



組織を作成および管理するには、Uyuni管理者としてログインする必要があります。

プロセージャ: 新しい組織の作成

1. Uyuni Web UIで、管理 > 組織に移動して、[組織の作成] をクリックします。
2. [組織の作成] ダイアログで、次のフィールドに入力します。
 - [Organization Name] (組織名) フィールドに、新しい組織の名前を入力します。名前は3～128文字である必要があります。
 - [希望のログイン] フィールドに、組織の管理者に使用するログイン名を入力します。既存の管理者アカウントを使用して、新しい組織にサインインすることはできません。
 - [希望のパスワード] フィールドに、新しい組織の管理者のパスワードを再度入力して確認します。パスワードの強度はパスワードフィールドの下の色付きバーで示されます。
 - [電子メール] フィールドに、新しい組織の管理者の電子メールアドレスを入力します。
 - [ファーストネーム(名)] フィールドで、挨拶文を選択し、新しい組織の管理者の名前を入力します。
 - [ラストネーム(姓)] フィールドに、新しい組織の管理者の姓を入力します。
3. [組織の作成] をクリックします。

16.1. 組織の管理

Uyuni Web UIで、管理 > 組織に移動して、利用可能な組織のリストを表示します。管理する組織の名前をクリックします。

管理 > 組織セクションから、組織のユーザ、信頼、設定、および状態を管理するタブにアクセスできます。



組織を管理できるのは管理者のみです。組織を管理するには、変更する組織の正しい管理者としてサインインしていることを確認してください。

16.1.1. 組織ユーザ

[ユーザ] タブに移動して、組織に関連付けられているすべてのユーザとそのロールのリストを表示します。ユーザ名をクリックすると、ユーザを追加、変更、または削除する [ユーザ] メニューに移動します。

16.1.2. 組織の作成

[信頼] タブに移動して、信頼されている組織を追加または削除します組織間で信頼を確立することにより、それらの組織間でコンテンツを共有し、ある組織から別の組織にクライアントを転送できるようになります。

16.1.3. 組織の設定

[設定] タブに移動し、組織の設定を管理しますこれには、ステージングされたコンテンツの使用、およびSCAPファイルの使用が含まれます。

コンテンツステージングの詳細については、Administration > Content-stagingを参照してください。

OpenSCAPの詳細については、Reference > Auditを参照してください。

16.2. 状態の管理

[状態] タブに移動して、組織のすべてのクライアントのSalt状態を管理します状態を使用すると、グローバルセキュリティポリシーを定義したり、すべてのクライアントに共通の管理ユーザを追加したりできます。

Salt状態の情報については、Specialized-guides > Saltを参照してください。

16.2.1. 設定チャンネルの管理

組織全体に適用する設定チャンネルを選択できます。設定チャンネルは、Uyuni Web UIで設定 > チャンネルに移動して作成できます。Uyuni Web UIを使用して組織に設定チャンネルを適用します。

プロシージャ: 組織への設定チャンネルの適用

1. Uyuni Web UIで、ホーム > 所属企業/組織 > 設定チャンネルに移動します。
2. 検索機能を使用して、名前でチャンネルを見つけます。
3. 適用するチャンネルをオンにし、[変更点の保存]をクリックします。データベースに保存されますが、チャンネルに変更は適用されません。
4. [適用]をクリックして変更を適用します。これにより、組織内のすべてのクライアントに変更が適用されるようにタスクがスケジュールされます。

Chapter 17. パッチ管理

この章では、パッチ管理に関するさまざまなトピックについて説明します。

17.1. 撤回されたパッチ

ベンダから新しいパッチがリリースされると、テストで特定されなかつたいくつかのシナリオでは、パッチに望ましくない副作用(セキュリティ、安定性)が生じる可能性があります。これが発生すると(非常にまれ)、ベンダは通常、新しいパッチをリリースします。このパッチには、そのベンダが実施している内部プロセスによっては、数時間または数日かかる場合があります。

SUSEでは、「撤回されたパッチ」と呼ばれる新しいメカニズム(2021年)を導入し、このようなパッチのアドバイザリーステータスを(「最終」または「安定」ではなく)「撤回」に設定することで、ほぼ即座にこのようなパッチを取り消します。



アドバイザリーステータス属性が「撤回」に設定されている場合、パッチは「撤回」されています。パッケージが「撤回された」パッチに属している場合、そのパッケージは「撤回」されています。

撤回されたパッチまたはパッケージは、Uyuniのシステムにはインストールできません。撤回されたパッケージをインストールする唯一の方法は、`zypper install` を使用して手動で実行し、正確なパッケージバージョンを指定することです。例:

```
zypper install vim-8.0.1568-5.14.1
```

パッチおよびパッケージの撤回ステータスは、UyuniのWeb UIの⑧アイコンで示されます。たとえば、以下を参照してください。

- ・ チャンネルのパッケージのリスト
- ・ チャンネルのパッチのリスト

システムにインストールされているパッチまたはパッケージが撤回されると、そのシステムのインストール済みパッケージリストに⑨アイコンも表示されます。Uyuniでは、このようなパッチまたはパッケージをダウングレードする方法は提供されていません。

17.1.1. チャンネルクローン

複製されたチャンネルを使用する場合は、元のチャンネルからクローンへの撤回されたアドバイザリーステータスの伝播に注意する必要があります。

ベンダチャンネルを組織に複製すると、チャンネルパッチも複製されます。

ベンダがチャンネル内のパッチを撤回し、Uyuniがこのチャンネルを同期する(たとえば、夜間ジョブと同期する)と、「撤回」属性はクローンパッチに伝播されず、クローンチャンネルにサブスクライブされたクライアントによって監視されません。属性をクローンチャネルに伝播するには、次のいずれかの方法を使用しま

す。

- ・ パッチ同期(ソフトウェア > 管理 > **cloned channel** (クローンチャンネル) > パッチ > 同期)。この機能を使用すると、クローンチャンネルのパッチの属性をオリジナルに合わせることができます。
- ・ コンテンツライフサイクル管理。コンテンツライフサイクル管理のコンテキストでのクローンチャンネルの詳細については、**Client-configuration** > **Channels**を参照してください。

17.1.2. パッチの共有

組織内で複数のベンダチャンネルのクローンを作成する場合、パッチは複数回複製されるのではなく、クローンチャンネル間で共有されます。その結果、(パッチ同期機能または先述のコンテンツライフサイクル管理を使用して)複製されたパッチを同期すると、複製されたパッチを使用するすべてのチャンネルでその変更が確認されます。

例:

1. 2つのコンテンツライフサイクル管理プロジェクト **prj1** と **prj2** を検討します。
2. これらのプロジェクトの両方に2つの環境 **dev** と **test** があります。
3. これらのプロジェクトの両方にソースチャンネルとして設定されたベンダチャンネルがあります。
4. このシナリオのすべてのチャンネル(合計4つのクローンチャネル)は、ベンダチャンネルの最新の状態に合わせて調整されます。
5. ベンダがソースチャンネル内のパッチを撤回し、夜間ジョブがそのパッチをUyuniに同期します。
6. 4つのチャンネルのいずれも、パッチを直接使用するのではなく、パッチクローンを使用しているため、この変更を認識しません。
7. パッチを同期するとすぐに(これら2つのプロジェクトのいずれかを構築するか、または4つのクローンチャネルのいずれかでパッチ同期機能を使用する)、パッチの共有により、*すべて*のクローンチャンネルがそのパッチを撤回されたものと認識します。

Chapter 18. レポートの生成

Uyuniでは、ユーザがさまざまなレポートを作成できます。これらのレポートはサブスクライブしたシステム、ユーザ、および組織のインベントリを取得するのに役立ちます。レポートの使用は、特に多数のシステムを管理している場合は、Uyuni Web UIから手動で情報を収集するより簡単な場合が多いです。

コマンドラインツール `spacewalk-report` を使用して事前設定されたレポートを生成することができますが、[Specialized-guides](#) > [Large-deployments](#)を導入することで、完全にカスタマイズされたレポートを生成することもできます。これは、SQL言語をサポートする任意のレポーティングツールをレポーティングデータベースに接続し、データを直接抽出することで実現できます。データの可用性と構造の詳細については、レポーティングデータベーススキーマのドキュメントを参照してください。

18.1. `spacewalk-report` の使用

レポートを生成するには、`spacewalk-reports` パッケージをインストールしている必要があります。`spacewalk-report` コマンドを使用すると、Uyuni全体のコンテンツ、システム、およびユーザリソースに関するレポートを整理して表示できます。



`Specialized-guides`>`Large-deployments`の導入により、`spacewalk-report` はデフォルトでレポーティングデータベースからデータを収集するようになりました。 詳細については [spacewalk-report](#) および [レポーティングデータベース](#) を参照してください。

次に関するレポートを生成できます。

1. システムインベントリ: Uyuniに登録されているすべてのシステムを一覧にします。
2. パッチ: 登録されているシステムに関連するすべてのパッチを一覧にします。重大度、および特定のパッチに適用されるシステムでパッチをソートできます。
3. ユーザ: すべての登録済みユーザおよび特定のユーザに関連するすべてのシステムを一覧にします。

CSV形式でレポートを取得するには、サーバのコマンドプロンプトで次のコマンドを実行します。

```
spacewalk-report <report_name>
```

18.2. `spacewalk-report` およびレポーティングデータベース

`spacewalk-report` は、デフォルトで新しいレポーティングデータベースを使用してデータを抽出します。これは、新しく生成されたレポートには、データの構造と形式にいくつかの相違があることを意味します。すべてのレポートに共通する相違点は次のとおりです。

- ・ レポートデータはリアルタイムでは変更されませんが、スケジュールされたタスクの実行によってのみ更新されます。
- ・ データの重複が削除され、以前は「多値」と見なされていた列には、;で区切られた複数の値が含まれるようになりました。これはまた、コマンドラインオプション `--multival-on-rows` と `--multival`

`-separator` が新しいレポートには適用されなくなったことも意味します。これは、これらの動作がデフォルトになったためです。

- すべてのレポートで、新しい列の `mgm_id` と `synced_date` が導入され、ハブシナリオの管理サーバと、アプリケーションデータベースから情報が最後に更新された時刻を識別します。
- すべての布尔値は、`1 / 0` 値ではなく、`True / False` で表されるようになりました。
- 列 `org_id` は、数値識別子ではなく組織名を含む、`organization` に置き換えされました。
- 「サーバ」という用語は「システム」に置き換えられました。したがって、たとえば、列 `server_id` は `system_id` と呼ばれるようになりました。

レポート固有の変更については、[使用可能なレポートのリスト](#) を参照してください。



この変更された動作によって問題が発生した場合、新しいオプション `--legacy-report` を使用して、アプリケーションデータベースに対して実行される古いレポートにフォールバックできます。

hubレポートの詳細については、[Specialized-guides > Large-deployments](#) を参照してください。

18.3. 使用可能なレポートのリスト

この表は、使用可能なレポートを一覧表示します。

表 8. `spacewalk-report` レポート

レポート	呼び出される手段	説明	レポートティングデータベースを使用するか	具体的な違い
アクション	<code>actions</code>	すべてのアクション。	はい	列 <code>id</code> が ``action_id`` と呼ばれるようになりました
アクティベーションキー	<code>activation-keys</code>	すべてのアクティベーションキー、およびそれらに関連付けられたエンタイトルメント、チャネル、設定チャネル、システムグループ、およびパッケージ。	いいえ	
アクティベーションキー: チャンネル	<code>activation-keys-channels</code>	すべてのアクティベーションキーおよび各キーに関連付けられたエンティティ。	いいえ	

レポート	呼び出される手段	説明	レポートイングデータベースを使用するか	具体的な違い
アクティベーションキー: 設定	<code>activation-keys-config</code>	すべてのアクティベーションキーおよび各キーに関連付けられた設定チャンネル。	いいえ	
アクティベーションキー: サーバグループ	<code>activation-keys-groups</code>	すべてのアクティベーションキーおよび各キーに関連付けられたシステムグループ。	いいえ	
アクティベーションキー: パッケージ	<code>activation-keys-packages</code>	すべてのアクティベーションキーおよび各キーが配備できるパッケージ。	いいえ	
チャンネルパッケージ	<code>channel-packages</code>	チャンネル内のすべてのパッケージ。	はい	
チャンネルレポート	<code>channels</code>	指定されたチャンネルの詳細レポート。	はい	
クローンチャンネルレポート	<code>cloned-channels</code>	クローンチャンネルの詳細レポート。	はい	
設定ファイル	<code>config-files</code>	ファイルの内容とファイル情報を含む、全組織の全設定ファイルのリビジョン。	いいえ	
最新の設定ファイル	<code>config-files-latest</code>	ファイルの内容とファイル情報を含む、全組織の最新設定ファイルのリビジョン。	いいえ	
カスタムチャンネル	<code>custom-channels</code>	特定の組織が所有するすべてのチャンネルのチャンネルメタデータ。	はい	列 <code>id</code> は <code>channel_id</code> と呼ばれるようになりました
カスタム情報	<code>custom-info</code>	クライアントカスタム情報。	はい	
チャンネルのパッチ	<code>errata-channels</code>	チャンネル内のすべてのパッチ。	はい	
パッチの詳細	<code>errata-list</code>	登録済みクライアントに影響するすべてのパッチ。	はい	
すべてのパッチ	<code>errata-list-all</code>	すべてのパッチ。	いいえ	

レポート	呼び出される手段	説明	レポートイングデータベースを使用するか	具体的な違い
クライアント用パッチ	<code>errata-systems</code>	適用可能なパッチおよび影響を受ける登録済みクライアント。	はい	
ホストゲスト	<code>host-guests</code>	ホストおよびゲストのマッピング。	はい	
非アクティブなクライアント	<code>inactive-systems</code>	非アクティブなクライアント。	はい	必須パラメータは「 <code>しきい値</code> 」と呼ばれます。
システムインベントリ	<code>inventory</code>	サーバに登録されているクライアントと、ハードウェアおよびソフトウェアの情報。	はい	列 <code>osad_status</code> は削除されました。
キックスタートスクリプト	<code>kickstart-scripts</code>	詳細を含む、すべてのキックスタートスクリプト。	いいえ	
キックスタートツリー	<code>kickstartable-trees</code>	キックスタートツリー。	いいえ	"
すべてのアップグレード可能バージョン	<code>packages-updates-all</code>	アップグレード可能なすべてのより新しいパッケージバージョン。	はい	
最新のアップグレード可能なバージョン	<code>packages-updates-newest</code>	アップグレード可能な最新のパッケージバージョン。	はい	
プロキシの概要	<code>proxies-overview</code>	すべてのプロキシとそれぞれに登録されたクライアント。	はい	
リポジトリ	<code>repositories</code>	関連付けられたSSLの詳細およびフィルタを含むすべてのリポジトリ。	いいえ	
SCAPの結果	<code>scap-scan</code>	OpenSCAP <code>sccdf</code> 評価の結果。	はい	
SCAPの結果	<code>scap-scan-results</code>	別の形式でのOpenSCAP <code>sccdf</code> 評価の結果。	はい	
システムデータ	<code>splice-export</code>	スプライス統合に必要なクライアントデータ。	いいえ	

レポート	呼び出される手段	説明	レポートイングデータベースを使用するか	具体的な違い
システム通貨	<code>system-currency</code>	登録済みクライアントごとに使用可能なパッチの数。	いいえ	
システムの追加パッケージ	<code>system-extra-packages</code>	クライアントがサブスクリーブしているチャンネルから利用できない、すべてのクライアントにインストールされたすべてのパッケージ。	はい	
システムグループ	<code>system-groups</code>	システムグループ。	はい	
システムグループのアクティベーションキー	<code>system-groups-keys</code>	システムグループのアクティベーションキー。	いいえ。	
システムグループのシステム	<code>system-groups-systems</code>	システムグループのクライアント。	はい	
システムグループユーザ	<code>system-groups-users</code>	システムグループおよびシステムグループに対する許可を持つユーザ。	いいえ	
履歴: システム	<code>system-history</code>	クライアントごとのイベント履歴。	はい	
履歴: チャンネル	<code>system-history-channels</code>	チャンネルイベント履歴。	はい	
履歴: 設定	<code>system-history-configuration</code>	設定イベント履歴	はい	列 <code>created_date</code> は削除されました。
履歴: エンタイトルメント	<code>system-history-entitlements</code>	システムエンタイトルメントイベント履歴。	はい	
履歴: エラータ	<code>system-history-errata</code>	エラータイベント履歴。	はい	列 <code>created_date</code> は削除されました。
履歴: キックスター	<code>system-history-kickstart</code>	キックスタートイベント履歴。	はい	列 <code>created_date</code> は削除されました。
履歴: パッケージ	<code>system-history-packages</code>	パッケージイベント履歴。	はい	列 <code>created_date</code> は削除されました。
履歴: SCAP	<code>system-history-scap</code>	OpenSCAPイベント履歴。	はい	列 <code>created_date</code> は削除されました。

レポート	呼び出される手段	説明	レポートイングデータベースを使用するか	具体的な違い
MD5証明書	<code>system-md5-certificates</code>	MD5チェックサムを持つ証明書を使用するすべての登録済みクライアント。	いいえ	
インストール済みパッケージ	<code>system-packages-installed</code>	クライアントにインストールされたパッケージ。	はい	
システムプロファイル	<code>system-profiles</code>	サーバに登録されているすべてのクライアントと、ソフトウェアおよびシステムグループ情報。	いいえ	
ユーザ	<code>users</code>	Uyuniに登録されたすべてのユーザ。	はい	列 <code>organization_id</code> は削除されました。
MD5ユーザ	<code>users-md5</code>	MD5暗号化パスワードを使用している全組織の全ユーザと、その詳細とロール。	はい	列 <code>organization_id</code> は削除されました。
管理されるシステム	<code>users-systems</code>	個々のユーザが管理できるクライアント。	はい	列 <code>organization_id</code> は削除されました。

個々のレポートに関する詳細については、オプション `--info` または `--list-fields-info` およびレポート名を指定して、`spacewalk-report` を実行します。これにより、レポートに使用可能なフィールドの説明とリストが表示されます。

プログラムの呼び出しとオプションの詳細については、`spacewalk-report(8)` マニュアルページ、および `spacewalk-report` コマンドの `--help` パラメータを参照してください。

Chapter 19. セキュリティ

19.1. クライアントをマスター検証指紋に設定する

高度に安全なネットワーク設定では、Saltクライアントが特定のマスターに接続していることを確認したい場合があります。 クライアントからマスターへの検証を設定するには、`/etc/salt/minion` 設定ファイル内にマスターの指紋を入力します。次のプロシージャを参照してください。

プロシージャ: マスターの指紋をクライアントに追加する

1. マスターのコマンドプロンプトで、rootとして、次のコマンドを使用して、`master.pub` の指紋を見つけます。

```
salt-key -F master
```

クライアントで、`/etc/salt/minion` 設定ファイルを開きます。次の行のコメントを解除し、指紋の例を置き換えて、マスターの指紋を入力します。

```
master_finger: 'ba:30:65:2a:d6:9e:20:4f:d8:b2:f3:a7:d4:65:11:13'
```

2. salt-minionサービスを再起動します。

```
# systemctl restart salt-minion
```

クライアントからのセキュリティの設定について

は、<https://docs.saltstack.com/en/latest/ref/configuration/minion.html>を参照してください。

19.2. リポジトリメタデータの署名

リポジトリメタデータを署名できるようにするにはカスタムGPGキーが必要です。

プロシージャ: カスタムGPGキーの生成

1. rootユーザとして、`gpg` コマンドを使用して、新しいキーを生成します。

```
gpg --gen-key
```

2. プロンプトが表示されたら、サイズが2048ビットのRSAをキータイプとして選択し、キーの適切な有効期限を選択します。新しいキーの詳細を確認して、「y」と入力して確定します。
3. プロンプトが表示されたら、キーに関連付けられた名前と電子メールアドレスを入力します。必要に応じて、キーの識別に役立つコメントを追加することもできます。ユーザIDに問題がなければ、「o」と入力して確定します。

4. プロンプトが表示されたら、キーを保護するパスフレーズを入力します。
5. キーは自動的にキーリングに追加されます。キーリングにキーを一覧表示して確認できます。

```
gpg --list-keys
```

6. テキストエディタでファイルを開き、次の行を追加して、キーリングのパスワードを `/etc/rhn/signing.conf` 設定ファイルに追加します。

```
GPGPASS="password"
```

GPGキーの更新については、Administration > Troubleshootingを参照してください。

mgr-sign-metadata-ctl コマンドを使用してコマンドラインでメタデータ署名を管理できます。

プロシージャ: メタデータ署名の有効化

1. 使用するキーの短い識別子を知っている必要があります。 使用可能な公開鍵を短い形式で一覧表示できます。

```
gpg --keyid-format short --list-keys
...
pub    rsa2048/3E7BFE0A 2019-04-02 [SC] [expires: 2021-04-01]
      A43F9EC645ED838ED3014B035CFA51BF3E7BFE0A
uid      [ultimate] SUSE Manager
sub    rsa2048/118DE7FF 2019-04-02 [E] [expires: 2021-04-01]
```

2. **mgr-sign-metadata-ctl** コマンドを使用してメタデータ署名を有効化します。

```
mgr-sign-metadata-ctl enable 3E7BFE0A
OK. Found key 3E7BFE0A in keyring.
DONE. Set key 3E7BFE0A in /etc/rhn/signing.conf.
DONE. Enabled metadata signing in /etc/rhn/rhn.conf.
DONE. Exported key 4E2C3DD8 to /srv/susemanager/salt/gpg/mgr-
keyring.gpg.
DONE. Exported key 4E2C3DD8 to /srv/www/htdocs/pub/mgr-gpg-pub.key.
NOTE. For the changes to become effective run:
      mgr-sign-metadata-ctl regen-metadata
```

3. このコマンドを使用して設定が正しいことを確認できます。

```
mgr-sign-metadata-ctl check-config
```

- サービスを再起動し、メタデータの再生成をスケジュールして変更を取得します。

```
mgr-sign-metadata-ctl regen-metadata
```

mgr-sign-metadata-ctl コマンドを使用して他のタスクを実行することもできます。 **mgr-sign-metadata-ctl --help** を使用して、完全なリストを表示します。

リポジトリメタデータ署名はグローバルオプションです。 有効にすると、サーバ上のすべてのソフトウェアチャンネルで有効になります。 これは、サーバに接続されているすべてのクライアントが、パッケージをインストールまたは更新できるようにするために、新しいGPGキーを信頼する必要があることを意味します。

プロシージャ: クライアントへのGPGキーのインポート

1. GPGキーをクライアントに配備すると、salt状態で動作します。
2. Uyuni Web UIを使用してhighstateを適用します。

GPGキーのトラブルシューティングの詳細については、Administration > Troubleshootingを参照してください。

19.3. ミラーソースパッケージ

独自のパッケージをローカルに構築する場合、または法的な理由でパッケージのソースコードが必要な場合は、Uyuniサーバ上のソースパッケージをミラーリングできます。



ソースパッケージをミラーリングすると、大量のディスク容量が消費される可能性があります。

プロシージャ: ソースパッケージのミラーリング

1. **/etc/rhn/rhn.conf** 設定ファイルを開いて、次の行を追加します。

```
server.sync_source_packages = 1
```

2. Spacewalkサービスを再起動して、変更を取得します。

```
spacewalk-service restart
```

現在、この機能はすべてのリポジトリに対してグローバルにのみ有効にできます。 ミラーリングに個々のリポジトリを選択することはできません。

この機能を有効にすると、次のリポジトリ同期後に、ソースパッケージがUyuni Web UIで使用できるようになります。これらはバイナリパッケージのソースとして表示され、Web UIから直接ダウンロードできます。Web UIを使用して、ソースパッケージをクライアントにインストールすることはできません。

19.4. OpenSCAPによるシステムセキュリティ

UyuniはOpenSCAPを使用してクライアントを監査します。任意のクライアントのコンプライアンススキャンをスケジュールして表示できます。

19.4.1. SCAPについて

Security Content Automation Protocol (SCAP)は、コミュニティのアイデアから派生した相互運用可能な仕様を統合したものです。これは、エンタープライズシステムのシステムセキュリティを維持するためには、National Institute of Standards and Technology (NIST)によって維持されている一連の仕様です。

SCAPは、システムのセキュリティを維持するための標準化されたアプローチを提供するために作成されました。また、使用される標準は、コミュニティや企業のビジネスニーズを満たすために継続的に変更されます。新しい仕様はNISTのSCAPリリースサイクルによって管理され、一貫性のある再現可能な改訂ワークフローを提供します。詳細については、以下を参照してください。

- <http://scap.nist.gov/timeline.html>
- <https://csrc.nist.gov/projects/security-content-automation-protocol>
- <https://www.open-scap.org/features/standards/>
- <https://ncp.nist.gov/repository?scap>

UyuniはOpenSCAPを使用してSCAP仕様を実装します。OpenSCAPは、Extensible Configuration Checklist Description Format (XCCDF)を利用した監査ツールです。XCCDFは、チェックリストの内容を表現する標準的な方法であり、セキュリティチェックリストを定義します。また、Common Platform Enumeration (CPE)、Common Configuration Enumeration (CCE)、Open Vulnerability and Assessment Language (OVAL)などの他の仕様と組み合わせて、SCAP検証済みの製品で処理できるSCAP表現のチェックリストを作成します。

OpenSCAPはSUSEセキュリティチームが作成したコンテンツを使用して、パッチの存在を確認します。OpenSCAPは、システムセキュリティ設定をチェックし、標準と仕様に基づいたルールを使用して、システムに侵害の兆候がないかどうかを検査します。SUSEセキュリティチームの詳細については、<https://www.suse.com/support/security>を参照してください。

19.4.2. SCAPスキャンのためのクライアントの準備

開始する前に、SCAPスキャン用にクライアントシステムを準備する必要があります。



OpenSCAP監査は、SSH接続メソッドを使用するSaltクライアントでは使用できません。



スキャンクライアントは、スキャンするクライアントのメモリと計算能力を大量に消費する可能性があります。Red Hatクライアントの場合、スキャンする各クライアントで少なくとも2GBのRAMが使用可能であることを確認してください。

従来のクライアントとSaltクライアントの場合は、開始する前にOpenSCAPスキャナとSCAPセキュリティガイド(コンテンツ)パッケージをインストールしてください。オペレーティングシステムに応じて、これらのパッケージはベースオペレーティングシステムまたはUyuniクライアントツールのいずれかに含まれています。

次の表に、クライアントオペレーティングシステムに応じて必要なパッケージを一覧表示します。

表 9. OpenSCAPパッケージ

オペレーティングシステム	スキャナ	コンテンツ
SLES	openscap-utils	scap-security-guide
openSUSE	openscap-utils	scap-security-guide
RHEL	openscap-utils	scap-security-guide-redhat
CentOS	openscap-utils	scap-security-guide-redhat
Oracle Linux	openscap-utils	scap-security-guide-redhat
Ubuntu	libopenscap8	scap-security-guide-ubuntu
Debian	libopenscap8	scap-security-guide-debian

RHEL 7および互換システムでは、**scap-security-guide**パッケージが提供されています。このパッケージには、古い内容が含まれています。Uyuniクライアントツールにある**scap-security-guide-redhat**パッケージを使用することをお勧めします。



SUSEは、異なるOpenSCAPプロファイル用の**scap-security-guide**パッケージを提供しています。**scap-security-guide**の現在のバージョンでは、SUSEは次のプロファイルをサポートしています。

- ・ SUSE Linux Enterprise Server 12および15のDISA STIGプロファイル
- ・ SUSE Linux Enterprise Server 12および15のPCI-DSSプロファイル
- ・ SUSE Linux Enterprise Server 12および15のHIPAAプロファイル

CISプロファイルなどの他のプロファイルは、コミュニティが提供するものであり、SUSEによって公式にサポートされていません。

SUSE以外のオペレーティングシステムの場合、含まれるプロファイルはコミュニティが提供します。これらはSUSEによって公式にサポートされていません。

19.4.3. OpenSCAPコンテンツファイル

OpenSCAPIは、SCAPコンテンツファイルを使用してテストルールを定義します。これらのコンテンツファイルは、XCCDFまたはOVAL標準に基づいて作成されます。『SCAPセキュリティガイド』に加えて、公開さ

れているコンテンツファイルをダウンロードして、要件に合わせてカスタマイズできます。 デフォルトのコンテンツファイルテンプレート用のSCAPセキュリティガイドパッケージをインストールできます。 または、XCCDFまたはOVALに精通している場合は、独自のコンテンツファイルを作成できます。



テンプレートを使用してSCAPコンテンツファイルを作成することをお勧めします。 独自のカスタムコンテンツファイルを作成して使用する場合は、自己責任で作成してください。 カスタムコンテンツファイルを使用してシステムが破損した場合、SUSEのサポートを受けられない可能性があります。

コンテンツファイルを作成したら、ファイルをクライアントに転送する必要があります。 これは、物理的なストレージメディアを使用して他のファイルを移動するのと同じ方法で行うことができます。 または、Salt (たとえば、[salt-cp](https://docs.saltproject.io/en/latest/ref/cli/salt-cp.html)) や[salt-cp](https://docs.saltproject.io/en/latest/ref/file_server/index.html) [Salt File Server])、**ftp** や **scp** を使用してネットワークを介して移動することもできます。

Uyuniで管理しているクライアントにコンテンツファイルを配布するパッケージを作成することをお勧めします。 パッケージの整合性を確認するために、パッケージに署名して検証することができます。 詳細については、Administration > Custom-channelsを参照してください。

19.4.4. OpenSCAPプロファイルの検索

オペレーティングシステムによって、使用可能なOpenSCAPコンテンツファイルとプロファイルが異なります。 1つのコンテンツファイルに複数のプロファイルを含めることができます。

RPMベースのオペレーティングシステムでは、次のコマンドを使用して、使用可能なSCAPファイルの場所を決定します。

```
rpm -ql <scap-security-guide-package-name-from-table>
```

DEBベースのオペレーティングシステムでは、次のコマンドを使用して、使用可能なSCAPファイルの場所を決定します。

```
dpkg -L <scap-security-guide-package-name-from-table>
```

ニーズに合う1つの SCAPコンテンツファイルを特定したら、クライアントに使用可能なプロファイルを一覧にします。

```

oscap info ./usr/share/xml/scap/ssg/content/ssg-sle15-ds-1.2.xml
Document type: Source Data Stream
Imported: 2021-03-24T18:14:45

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-sle15-xccdf-1.2.xml
Generated: (null)
Version: 1.2
Checklists:
    Ref-Id: scap_org.open-scap_cref(ssg-sle15-xccdf-1.2.xml)
        Status: draft
        Generated: 2021-03-24
        Resolved: true
        Profiles:
            Title: CIS SUSE Linux Enterprise 15 Benchmark
            Id:
    xccdf_org.ssgproject.content_profile_cis
        Title: Standard System Security Profile for SUSE
        Linux Enterprise 15
        Id:
    xccdf_org.ssgproject.content_profile_standard
        Title: DISA STIG for SUSE Linux Enterprise 15
        Id:
    xccdf_org.ssgproject.content_profile_stig
        Referenced check files:
            ssg-sle15-oval.xml
            system:
http://oval.mitre.org/XMLSchema/oval-definitions-5
            ssg-sle15-ocil.xml
            system:
http://scap.nist.gov/schema/ocil/2

https://ftp.suse.com/pub/projects/security/oval/suse.linux.enterprise.15.
xml
            system:
http://oval.mitre.org/XMLSchema/oval-definitions-5
Checks:
    Ref-Id: scap_org.open-scap_cref(ssg-sle15-oval.xml)
    Ref-Id: scap_org.open-scap_cref(ssg-sle15-ocil.xml)
    Ref-Id: scap_org.open-scap_cref(ssg-sle15-cpe-oval.xml)
Dictionaries:
    Ref-Id: scap_org.open-scap_cref(ssg-sle15-cpe-dictionary.xml)

```

スキャンを実行するためのファイルパスとプロファイルをメモします。

19.4.5. 監査スキャンの実行

コンテンツファイルをインストールまたは転送したら、監査スキャンを実行できます。 監査スキャンは、Uyuni Web UIを使用してトリガできます。 Uyuni APIを使用して、定期的なスキャンをスケジュールすることもできます。

プロシージャ: Web UIからの監査スキャンの実行

1. Uyuni Web UIで、**システム** > **システム**一覧に移動して、スキャンするクライアントを選択します。
2. [監査] タブ、および [スケジュール] サブタブに移動します。
3. [XCCDF ドキュメントへのパス] フィールドに、クライアントで使用するSCAPテンプレートとプロファイルのパラメータを入力します。例:

```
Command: /usr/bin/oscap xccdf eval
Command-line arguments: --profile
xccdf_org.ssgproject.content_profile_standard
Path to XCCDF document: /usr/share/xml/scap/ssg/content/ssg-sle15-ds-
1.2.xml
```

1. スキャンは、クライアントの次にスケジュールされた同期時に実行されます。

XCCDFコンテンツファイルはリモートシステムで実行される前に検証されます。 コンテンツファイルに無効な引数が含まれている場合は、テストに失敗します。

プロシージャ: APIからの監査スキャンの実行

1. 開始する前に、スキャンするクライアントにPythonおよびXML-RPCライブラリがインストールされていることを確認します。
2. 既存のスクリプトを選択するか、**system.scap.scheduleXccdfScan** を使用してシステムスキャンをスケジュールするスクリプトを作成します。例:

```
#!/usr/bin/python
client = xmlrpclib.Server('https://server.example.com/rpc/api')
key = client.auth.login('username', 'password')
client.system.scap.scheduleXccdfScan(key, <1000010001>,
'<path_to_xccdf_file.xml>',
'--profile <profile_name>')
```

この例では:
* **<1000010001>** はシステムID(sid)です。
* **<path_to_xccdf_file.xml>** は、クライアント上のコンテンツファイルの場所へのパスです。たとえば、**/usr/share/xml/scap/ssg/content/ssg-sle15-ds-1.2.xml**。
* **<profile_name>** は **oscrap** コマンドの追加の引数です。たとえば、**united_states_government_configuration_baseline** (USGCB)を使用します。

3. コマンドプロンプトから、スキャンするクライアント上でスクリプトを実行します。

19.4.6. スキャン結果

実行したスキャンに関する情報は、Uyuni Web UIにあります。結果の表を表示するには、[監査 > OpenSCAP](#) > [全スキャン](#)に移動します。この表のデータの詳細については、[Reference > Audit](#)を参照してください。

スキャンに関する詳細情報を使用できるようにするには、クライアントで有効にする必要があります。Uyuni Web UIで、[管理 > 組織](#)に移動し、クライアントが属する組織をクリックします。[設定] タブに移動し、[詳細なSCAP ファイルのアップロードを有効にする] にチェックを入れます。次に、追加情報を含む追加のHTMLファイルが生成されます。結果には、次のような追加の行が表示されます。

```
Detailed Results: xccdf-report.html xccdf-results.xml scap-yast2sec-oval.xml.result.xml
```

コマンドラインからスキャン情報を取得するには、[spacewalk-report](#) コマンドを使用します。

```
spacewalk-report system-history-scap
spacewalk-report scap-scan
spacewalk-report scap-scan-results
```

Uyuni APIを使用して、[system.scap](#) ハンドラを使用して結果を表示することもできます。

19.4.7. 修復

クライアントシステムを強化するために、修復用のbashスクリプトとAnsible playbookが同じSCAPセキュリティガイドパッケージで提供されます。例:

リスト 5. bashスクリプト

```
/usr/share/scap-security-guide/bash/sle15-script-cis.sh
/usr/share/scap-security-guide/bash/sle15-script-standard.sh
/usr/share/scap-security-guide/bash/sle15-script-stig.sh
```

リスト 6. Ansible playbook

```
/usr/share/scap-security-guide/ansible/sle15-playbook-cis.yml
/usr/share/scap-security-guide/ansible/sle15-playbook-standard.yml
/usr/share/scap-security-guide/ansible/sle15-playbook-stig.yml
```

クライアントシステムでAnsibleを有効にした後、リモートコマンドまたはAnsibleを使用して実行できます。

19.4.7.1. Bashスクリプトを使用して修復を実行する

scap-security-guide パッケージをすべてのターゲットシステムにインストールします。 詳細については、[ansible-setup-control-node.pdf](#)を参照してください。

パッケージ、チャンネル、スクリプトは、オペレーティングシステムと配布ごとに異なります。 [修復Bashスクリプトの例](#)セクションに例を一覧表示しています。

19.4.7.1.1. リモートコマンドとして単一システムでBashスクリプトを実行する

単一システムでリモートコマンドとしてBashスクリプトを実行します。

1. システム > 概要タブから、インスタンスを選択します。 次に、詳細 > リモートコマンドで、Bashスクリプトを次のように記述します。

```
#!/bin/bash
chmod +x -R /usr/share/scap-security-guide/bash
/usr/share/scap-security-guide/bash/sle15-script-standard.sh
```

2. [Schedule] をクリック。



配布とバージョンの間でフォルダ名とスクリプト名が変わります。 [修復Bashスクリプトの例](#)セクションに例を一覧表示しています。

19.4.7.1.2. 複数のシステムでシステムセットマネージャを使用してbashスクリプトを実行する

複数のシステムでリモートコマンドとして一度にBashスクリプトを実行します。

1. システムグループが作成されたら [システムグループ] をクリックし、テーブルかSSM で選択します。
2. [システムセットマネージャー] の下で、次のようなBashスクリプトを記述します。

```
#!/bin/bash
chmod +x -R /usr/share/scap-security-guide/bash
/usr/share/scap-security-guide/bash/sle15-script-standard.sh
```

3. [Schedule] をクリック。

19.4.7.2. 修復Bashスクリプトの例

19.4.7.2.1. SUSE Linux Enterprise openSUSEおよび亞種

SUSE Linux EnterpriseおよびopenSUSEスクリプトデータの例。

表 10. SUSE Linux Enterprise openSUSE

パッケージ	scap-security-guide
チャンネル	SLE12: SLES12 Updates SLE15: SLES15 Module Basesystemの更新
Bashスクリプトフォルダ	/usr/share/scap-security-guide/bash/
Bashスクリプト	opensuse-script-standard.sh sle12-script-standard.sh sle12-script-stig.sh sle15-script-cis.sh sle15-script-standard.sh sle15-script-stig.sh

19.4.7.2.2. Red Hat Enterprise LinuxおよびCentOS Bashスクリプトデータ

Red Hat Enterprise LinuxおよびCentOSスクリプトデータの例。



centos7-updatesの**scap-security-guide**には、Red Hat Enterprise Linuxスクリプトのみが含まれています。

表 11. Red Hat Enterprise Linux CentOSおよび亜種

パッケージ	scap-security-guide-redhat
チャンネル	SUSE Managerツール+
Bashスクリプトフォルダ	/usr/share/scap-security-guide/bash/

Bashスクリプト

centos7-script-pci-dss.sh
centos7-script-standard.sh
centos8-script-pci-dss.sh
centos8-script-standard.sh
fedora-script-ospp.sh
fedora-script-pci-dss.sh
fedora-script-standard.sh
ol7-script-anssi_nt28_enhanced.sh
ol7-script-anssi_nt28_high.sh
ol7-script-anssi_nt28_intermediary.sh
ol7-script-anssi_nt28_minimal.sh
ol7-script-cjis.sh
ol7-script-cui.sh
ol7-script-e8.sh
ol7-script-hipaa.sh
ol7-script-ospp.sh
ol7-script-pci-dss.sh
ol7-script-sap.sh
ol7-script-standard.sh
ol7-script-stig.sh
ol8-script-anssi_bp28_enhanced.sh
ol8-script-anssi_bp28_high.sh
ol8-script-anssi_bp28_intermediary.sh
ol8-script-anssi_bp28_minimal.sh
ol8-script-cjis.sh
ol8-script-cui.sh
ol8-script-e8.sh
ol8-script-hipaa.sh
ol8-script-ospp.sh
ol8-script-pci-dss.sh
ol8-script-standard.sh
rhel7-script-anssi_nt28_enhanced.sh
rhel7-script-anssi_nt28_high.sh
rhel7-script-anssi_nt28_intermediary.sh
rhel7-script-anssi_nt28_minimal.sh
rhel7-script-C2S.sh
rhel7-script-cis.sh
rhel7-script-cjis.sh
rhel7-script-cui.sh
rhel7-script-e8.sh
rhel7-script-hipaa.sh
rhel7-script-ncp.sh
rhel7-script-ospp.sh
rhel7-script-pci-dss.sh
rhel7-script-rhelh-stig.sh
rhel7-script-rhelh-vpp.sh
rhel7-script-rht ccp.sh
rhel7-script-standard.sh
rhel7-script-stig_gui.sh
rhel7-script-stig.sh
rhel8-script-anssi_bp28_enhanced.sh

19.4.7.2.3. Ubuntu Bashスクリプトデータ

Ubuntuスクリプトデータの例。

表 12. Ubuntu

パッケージ	scap-security-guide-ubuntu
チャンネル	SUSE Managerツール
Bashスクリプトフォルダ	/usr/share/scap-security-guide/
Bashスクリプト	ubuntu1804-script-anssi_np_nt28_average.sh ubuntu1804-script-anssi_np_nt28_high.sh ubuntu1804-script-anssi_np_nt28_minimal.sh ubuntu1804-script-anssi_np_nt28_restrictive.sh ubuntu1804-script-cis.sh ubuntu1804-script-standard.sh ubuntu2004-script-standard.sh

19.4.7.2.4. Debian Bashスクリプトデータ

Debianスクリプトデータの例。

表 13. Debian

パッケージ	scap-security-guide-debian
チャンネル	SUSE Managerツール
Bashスクリプトフォルダ	/usr/share/scap-security-guide/bash
Bashスクリプト	debian10-script-anssi_np_nt28_average.sh debian10-script-anssi_np_nt28_high.sh debian10-script-anssi_np_nt28_minimal.sh debian10-script-anssi_np_nt28_restrictive.sh debian10-script-standard.sh debian11-script-anssi_np_nt28_average.sh debian11-script-anssi_np_nt28_high.sh debian11-script-anssi_np_nt28_minimal.sh debian11-script-anssi_np_nt28_restrictive.sh debian11-script-standard.sh

19.5. 監査

Uyuniでは、一連の監査タスクを通じてクライアントを追跡できます。 クライアントがすべてのパブリックセキュリティパッチ(CVE)を適用して最新の状態になっていることを確認し、サブスクリプションマッチングを実行して、OpenSCAPを使用して仕様のコンプライアンスを確認できます。

Uyuni Web UIで、 [監 査] に移動して、監査タスクを実行します。

19.5.1. CVE監査

CVE(共通脆弱性識別子)は、一般に知られているセキュリティの脆弱性に対する修正です。



CVEが利用可能になったらすぐにクライアントに適用する必要があります。

各CVEには、識別番号、脆弱性の説明、および詳細情報へのリンクが含まれています。CVE識別番号は、**CVE-YEAR-XXXX**の形式を使用します。

Uyuni Web UIで、**監査 > CVE監査**に移動して、すべてのクライアントとその現在のパッチステータスのリストを表示します。

デフォルトでは、CVEデータは毎日2300に更新されます。CVE監査を開始する前に、データを更新して最新のパッチが適用されていることを確認することをお勧めします。

プロシージャ: CVEデータの更新

1. Uyuni Web UIで、**管理 > タスクスケジュール**に移動し、**cve-server-channels-default**スケジュールを選択します。
2. **[cve-server-channels-bunch]**をクリックします。

[1] [回]の[み]の[実]行[スケジュール]をクリックして、タスクをスケジュール。CVE監査を続行する前に、タスクを完了させてください。

プロシージャ: パッチステータスの確認

1. Uyuni Web UIで、**監査 > CVE監査**に移動します。
2. オプション特定のCVEのパッチステータスを確認するには、**[CVE番号]**フィールドにCVEIDを入力します。
3. 検索するパッチステータスを選択するか、すべてのステータスをオンのままにしてすべてを検索します。
4. **[監査サバ]**をクリックしてすべてのシステムを確認するか、**[監査イメージ]**をクリックしてすべてのイメージを確認します。

このページで使用されるパッチステータスアイコンの詳細については、**Reference > Audit**を参照してください。

各システムについて、**[次の動作]**列には、脆弱性に対処するために実行する必要がある情報が表示されます。該当する場合は、候補チャンネルまたはパッチのリストも表示されます。さらにパッチ処理を行うためにシステムを**[システムセット]**に割り当てることもできます。

Uyuni APIを使用して、クライアントのパッチステータスを確認できます。
audit.listSystemsByPatchStatus APIメソッドを使用します。このメソッドの詳細については、『Uyuni APIガイド』を参照してください。

19.5.2. CVEステータス

クライアントのCVEステータスは通常、**影響を受ける** | **not affected (パッチ適用済み)** のいずれかです。これらのステータスは、Uyuniで使用できる情報にのみ基づいています。

Uyuni内で、次の定義が適用されます。

特定の脆弱性の影響を受けるシステム

脆弱性がマークされた関連パッチ内の同じパッケージのバージョンより前のバージョンのパッケージがインストールされているシステム。

特定の脆弱性の影響を受けないシステム

脆弱性がマークされた関連パッチにも含まれているパッケージがインストールされていないシステム。

特定の脆弱性に対するパッチが適用されたシステム

脆弱性がマークされた関連パッチ内の同じパッケージのバージョン以上のバージョンのパッケージがインストールされているシステム。

関連パッチ

関連するチャンネルでUyuniによって知られているパッチ。

関連するチャンネル

システムに割り当てられたUyuniによって管理されるチャンネル、システムに割り当てられたクローンチャンネルのオリジナルチャンネル、システムにインストールされている製品にリンクされたチャンネル、またはシステムの過去または将来のサービスパックチャンネル。



Uyuni内で使用されている定義のため、状況によってはCVE監査結果が正しくない場合があります。たとえば、管理されていないチャンネル、管理されていないパッケージ、または準拠していないシステムが誤って報告される可能性があります。

Chapter 20. SSL証明書

Uyuniでは、SSL証明書を使用して、クライアントが正しいサーバに登録されていることを確認します。

SSLを使用してUyuniサーバに登録するすべてのクライアントは、サーバ証明書に対して検証することにより、適切なサーバに接続していることを確認します。このプロセスはSSLハンドシェークと呼ばれます。

SSLハンドシェーク中に、クライアントはサーバ証明書のホスト名が予期しているホスト名と一致することを確認します。クライアントは、サーバ証明書が信頼できるかどうかを確認する必要があります。

認証局(CA)は、他の証明書に署名するために使用される証明書です。証明書が有効であると見なされ、クライアントが証明書と正常に照合できるようにするには、すべての証明書が認証局(CA)によって署名されている必要があります。

SSL認証が正しく機能するためには、クライアントがルートCAを信頼する必要があります。これは、ルートCAがすべてのクライアントにインストールされる必要があることを意味します。

SSL認証のデフォルトの方法は、Uyuniで自己署名証明書を使用することです。この場合、Uyuniはすべての証明書を生成し、ルートCAはサーバ証明書に直接署名しています。

別 の方法は、中間CAを使用する方法です。この場合、ルートCAは中間CAに署名します。中間CAは任意の数の他の中間CAに署名することができ、最後のCAはサーバ証明書に署名します。これはチェーン証明書と呼ばれます。

チェーン証明書で中間CAを使用している場合は、ルートCAがクライアントにインストールされ、サーバ証明書がサーバにインストールされます。SSLハンドシェーク中に、クライアントはルートCAとサーバ証明書の中間証明書のチェーン全体を検証できる必要があります。そのため、クライアントはすべての中間証明書にアクセスできる必要があります。

これを実現するには、主に2つの方法があります。以前のバージョンのUyuniでは、デフォルトですべての中間CAがクライアントにインストールされます。ただし、サーバ上でサービスを設定してクライアントに提供することも可能です。この場合、SSLハンドシェーク中に、サーバはサーバ証明書とすべての中間CAを提示します。このメカニズムは現在、新しいデフォルト設定として使用されています。

デフォルトでは、Uyuniは中間CAなしの自己署名証明書を使用します。セキュリティを強化するために、サードパーティのCAを手配して証明書に署名できます。サードパーティのCAは、証明書に含まれる情報が正しいことを確認するためにチェックを実行します。通常、このサービスには年会費がかかります。サードパーティのCAを使用すると、証明書のスプーフィングが難しくなり、インストールに対する保護が強化されます。サードパーティのCAによって署名された証明書がある場合は、Uyuniのインストール環境にインポートできます。

このマニュアルでは、SSL証明書の使用について2つのステップで説明します。

1. Uyuniツールを使用して自己署名証明書を作成する方法
2. Uyuniサーバまたはプロキシに証明書を配備する方法

証明書が独自のPKIや外部PKIなどのサードパーティのインスタンスによって提供されている場合は、ステップ1をスキップできます。

- 自己署名証明書の作成方法詳細については、Administration > Ssl-certs-selfsignedを参照してください。
- 証明書インポート方法の詳細については、Administration > Ssl-certs-importedを参照してください。

20.1. 自己署名SSL証明書

デフォルトでは、Uyuniは自己署名証明書を使用します。この場合、証明書はUyuniによって作成され、署名されます。この方法では、証明書の詳細が正しいことを保証するために独立した認証局を使用しません。サードパーティCAは、証明書に含まれる情報が正しいことを確認するためにチェックを実行します。サードパーティCAの詳細については、Administration > Ssl-certs-importedを参照してください。

このセクションでは、新規または既存のインストールで自己署名証明書を作成または再作成する方法について説明します。

SSLキーおよび証明書のホスト名はそれらを配備するマシンの完全修飾ホスト名に一致する必要があります。

20.1.1. 既存のサーバ証明書の再作成

既存の証明書の有効期限が切れているか、何らかの理由で動作を停止している場合は、既存のCAから新しいサーバ証明書を生成できます。

プロシージャ: 既存のサーバ証明書の再作成

- Uyuniサーバのコマンドプロンプトで、サーバ証明書を再生成します。

```
rhn-ssl-tool --gen-server --dir="/root/ssl-build" --set
-country="COUNTRY" \
--set-state="STATE" --set-city="CITY" --set-org="ORGANIZATION" \
--set-org-unit="ORGANIZATION UNIT" --set-email="name@example.com" \
--set-hostname="susemanager.example.com" --set-cname="example.com"
```

set-cname パラメータがUyuniサーバの完全修飾ドメイン名であることを確認します。複数のエイリアスが必要な場合は複数回、**set-cname** パラメータを使用できます。

秘密鍵とサーバ証明書はディレクトリ `/root/ssl-build/susemanager/` に `server.key` と `server.crt` としてあります。最後のディレクトリの名前は、`--set-hostname` オプションで使用されるホス

20.1.2. 新しいCAおよびサーバ証明書の作成



ルートCAを置き換える必要がある場合は注意してください。サーバとクライアントの間の信頼チェーンを切断する可能性があります。その場合は、管理ユーザがすべてのクライアントにログインしてCAを直接配備する必要があります。

プロシージャ: 新しい証明書の作成

- Uyuniサーバのコマンドプロンプトで、古い証明書ディレクトリを新しい場所に移動します。

```
mv /root/ssl-build /root/old-ssl-build
```

- 新しいCA証明書を生成します。

```
rhn-ssl-tool --gen-ca --dir="/root/ssl-build" --set-country="COUNTRY" \
--set-state="STATE" --set-city="CITY" --set-org="ORGANIZATION" \
--set-org-unit="ORGANIZATION UNIT" --set-common-name="SUSE Manager CA \
Certificate" \
--set-email="name@example.com"
```

- 新しいサーバ証明書を生成します。

```
rhn-ssl-tool --gen-server --dir="/root/ssl-build" --set \
-country="COUNTRY" \
--set-state="STATE" --set-city="CITY" --set-org="ORGANIZATION" \
--set-org-unit="ORGANIZATION UNIT" --set-email="name@example.com" \
--set-hostname="susemanager.example.top" --set-cname="example.com"
```

set-cname パラメータがUyuniサーバの完全修飾ドメイン名であることを確認します。複数のエイリアスが必要な場合は複数回、**set-cname** パラメータを使用できます。

ホスト名とcnameを使用して、各プロキシのサーバ証明書も生成する必要があります。

20.2. SSL証明書のインポート

このセクションでは、新しいUyuniのインストールにSSL証明書を設定する方法、および既存の証明書を置き換える方法について説明します。

開始する前に、以下があることを確認します。

- 認証局(CA) SSLパブリック証明書。 CAチェーンを使用している場合は、すべての中間CAも使用できる必要があります。
- SSLサーバ秘密鍵
- SSLサーバ証明書

すべてのファイルがPEM形式である必要があります。

SSLサーバ証明書のホスト名は、配備先マシンの完全修飾ホスト名と一致している必要があります。ホスト名は、証明書の **X509v3 Subject Alternative Name** セクションで設定できます。環境で必要な場合は、複数のホスト名を一覧にすることもできます。

サードパーティの機関は通常、中間CAを使用して、要求されたサーバ証明書に署名します。この場合、チェーン内のすべてのCAが使用できる必要があります。中間CAを指定するために使用できる追加のパラメータまたはオプションがない場合は、すべてのCA (ルートCAおよび中間CA)が1つのファイルに保存されるように注意してください。

20.2.1. 新しインストール用証明書のインポート

デフォルトで、Uyuniは自己署名証明書を使用します。初期セットアップを完了した後、デフォルトの証明書を、インポートされた証明書に置き換えることができます。

プロシージャ: 新しいUyuniサーバに証明書をインポートする

1. **Installation-and-upgrade > Install-intro** の手順に従って、Uyuniサーバをインストールします。
2. **Installation-and-upgrade > Server-setup** に従って、初期セットアップを完了します。
3. コマンドプロンプトで、SSL環境変数に証明書ファイルの場所を指定します。

```
export CA_CERT=<path_to_CA_certificates_file>
export SERVER_KEY=<path_to_web_server_key>
export SERVER_CERT=<path_to_web_server_certificate>
```

4. Uyuniのセットアップを完了します。

```
yast susemanager_setup
```

セットアップ中に証明書の詳細を入力するように求められたら、ランダムな値を入力します。これらの値はコマンドプロンプトで指定した値で上書きされます。



環境変数をエクスポートしたのと同じシェルから **yast susemanager_setup** コマンドを実行します。

20.2.2. 新しいプロキシインストール用の証明書のインポート

デフォルトでは、Uyuniプロキシは自己署名証明書を使用します。初期セットアップを完了した後で、デフォルトの証明書を、インポートされた証明書に置き換えることができます。

プロシージャ: 新しいUyuniプロキシに証明書をインポートする

1. **Installation-and-upgrade > Install-intro** の手順に従って、Uyuniプロキシをインストールします。
2. **Installation-and-upgrade > Proxy-setup** に従って、初期セットアップを完了します。

3. コマンドプロンプトで、次のコマンドを実行します。

```
configure-proxy.sh
```

4. [Do you want to import existing certificates?] (既存の証明書をインポートしますか?) プロンプトが表示されたら、「y」と入力します。
5. プロンプトに従ってセットアップを完了します。



サーバとプロキシのすべてのサーバ証明書に署名するには、同じ認証局を使用します。異なるCAで署名された証明書は一致しません。

20.2.3. 証明書を置き換える

Uyuniのインストールでアクティブな証明書を新しい証明書に置き換えることができます。 証明書を置き換えるには、インストールされているCA証明書を新しいCAに置き換えてから、データベースを更新します。

プロシージャ: 既存の証明書を置き換える

1. Uyuniサーバのコマンドプロンプトで、コマンド`mgr-ssl-cert-setup`を呼び出して、パラメータとして証明書を提供します。

```
mgr-ssl-cert-setup --root-ca-file=<Path_to_Root_CA_Certificate>
--server-cert-file=<Server_Cert_File> --server-key
-file=<Server_Key_File>
```

中間CAは、`--root-ca-file`で指定されたファイルで使用することも、`--intermediate-ca-file`で指定することもできます。`--intermediate-ca-file`オプションは複数回指定できます。このコマンドは、提供されたファイルに対していくつかのテストを実行し、それらが有効であり、要求されたユースケースに使用できるかどうかをテストします。

1. サービスを再起動して変更を取得します。

```
spacewalk-service stop
systemctl restart postgresql.service
spacewalk-service start
```

2. 新しいCAでデータベースを更新します。

```
/usr/bin/rhn-ssl-dbstore --ca-cert=<Path_to_Root_CA_Certificate>
```

プロキシを使用している場合は、各プロキシのホスト名とcnameを使用してサーバ証明書RPMを生成する必

要があります。 証明書を置き換えるには、Uyuniプロキシでも`mgr-ssl-cert-setup`を使用する必要があります。UyuniプロキシにはPostgreSQLデータベースがないため、`spacewalk-service restart`のみで十分です。

ルートCAが変更された場合は、Uyuniに接続されているすべてのクライアントに配備する必要があります。

プロシージャ: Saltクライアント上のルートCAのデプロイ

1. Uyuni Web UIで、**システム > 概要**に移動します。
2. すべてのSaltクライアントをチェックして、システムセットマネージャに追加します。
3. **システム > システムセットマネージャ > 概要**に移動します。
4. [状態] フィールドで、**[適用]**をクリックして、システムの状態を適用します。
5. [highstate] ページで、**[highstate] の [適用]**をクリックして、クライアントに変更を伝播します。

20.2.3.1. 従来のクライアントの追加処理

従来のクライアントは非推奨であり、Saltクライアントに置き換える必要があります。

Uyuniに接続されている従来の管理対象クライアントがまだ存在する場合にCAを置き換える必要がある場合は、いくつかの追加ステップが必要です。

重要なことは、Uyuniサーバおよびプロキシで新しいCAが有効にされても、クライアントが切断されないことです。「古い」ルートCA証明書と「新しい」ルートCA証明書を影響を受けるクライアントに配備し、それらを信頼します。設定チャンネルを使用して証明書ファイルをクライアントに配備し、リモートコマンド機能を使用してtrust storeを再生成します。

Uyuniサーバおよびプロキシで新しい証明書を有効にした後、接続が機能しているかどうか、およびクライアントでアクションをスケジュールできるかどうかをテストします。この場合、「古い」ルートCAをクライアントから削除できます。

20.3. HTTP Strict Transport Security

HTTP Strict Transport Security ([HSTS](#))は、プロトコルダウングレード攻撃やクッキーハイジャックなどの中間者攻撃からWebサイトを保護するのに役立つポリシーメカニズムです。

UyuniではHSTSを有効にできます。Uyuniサーバに対して有効にするには、次の手順に従います。

1. `'/etc/apache2/conf.d/zz-spacewalk-www.conf` を編集します。
2. `# Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"` 行のコメントを解除します。
3. `systemctl restart apache2` を使用してApacheを再起動します。

Uyuniプロキシに対して有効にするには、次の手順に従います。

1. `'/etc/apache2/conf.d/spacewalk-proxy.conf` を編集します。

2. `# Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"` 行のコメントを解除します。
3. `systemctl restart apache2` を使用してApacheを再起動します。

重要: Uyuniで生成されたデフォルトのSSL証明書または自己署名証明書を使用してHSTSを有効にすると、このような証明書に署名するために使用されたCAがブラウザによって信頼されていない限り、ブラウザはHTTPSを使用した接続を拒否します。 Uyuniで生成されたSSL証明書を使用している場合は、`http://<SERVER-HOSTNAME>/pub/RHN-ORG-TRUSTED-SSL-CERT` にあるファイルをすべてのユーザのブラウザにインポートすることでこの証明書を信頼することができます。

Chapter 21. サブスクリプションマッチング

SUSE製品にはSUSE Customer Center (SCC)によって管理されるサブスクリプションが必要です。 Uyuniは、SCCアカウントに対して登録済みの全クライアントのサブスクリプションステータスをチェックする夜間レポートを実行します。 このレポートには、どのクライアントがどのサブスクリプションを使用しているか、残りのサブスクリプション数と使用できるサブスクリプション数、および現在のサブスクリプションがないクライアントに関する情報が表示されます。

監査 > サブスクリプションマッチングに移動して、レポートを表示します。

[Subscriptions Report] (サブスクリプションレポート) タブには、現在のサブスクリプションと期限切れサブスクリプションに関する情報が表示されます。

[Unmatched Products Report] (一致しない製品レポート) タブには、現在のサブスクリプションがないクライアントのリストが表示されます。 これには、一致できなかったクライアント、Uyuniに現在登録されてないクライアントが含まれます。 このレポートには、製品名とまだ一致していないシステムの数が含まれます。

[ピン] タブでは、関連するサブスクリプションに個々のクライアントを関連付けることができます。これは、サブスクリプションマネージャがクライアントをサブスクリプションに自動的に正常に関連付けない場合に特に役立ちます。

[メッセージ] タブでは、マッチングプロセス中に発生したエラーが表示されます。

レポートは.csv形式でダウンロードすることも、コマンドプロンプトから /var/lib/spacewalk/subscription-matcher/ ディレクトリでアクセスすることもできます。

デフォルトでは、サブスクリプションマッチャーは毎日午前0時に実行されます。 これを変更するには、管理 > タスクスケジュールに移動し、[gatherer-matcher-default] をクリックします。 必要に応じてスケジュールを変更し、 をクリックします。

レポートは現在のクライアントと現在のサブスクリプションしか一致させることができないため、一致が時間の経過とともに変化することがあります。 同じクライアントが常に同じサブスクリプションと一致するわけではありません。 これは、新規クライアントの登録または登録解除、もしくはサブスクリプションの追加または期限切れが原因である可能性があります。

サブスクリプションマッチャーは、アカウント内のサブスクリプションの条項によって制限される、不一致の製品数を自動的に減らそうとします。 ただし、不完全なハードウェア情報、不明な仮想マシンホストの割り当て、または不明なパブリッククラウドで実行されているクライアントがある場合は、利用可能なサブスクリプションが十分でないことがマッチャーに表示されることがあります。 正確性を確保するために、常にUyuniに含まれるクライアントに関する完全なデータがあることを確認してください。



サブスクリプションマッチャーは、常にクライアントとサブスクリプションを正確に一致させるとは限りません。 これは監査に代わるものではありません。

21.1. クライアントをサブスクリプションにピン設定する

サブスクリプションマッチャーが特定のクライアントと正しいサブスクリプションを自動的に一致させない場合は、それらを手動でピン設定することができます。 ピンを作成すると、サブスクリプションマッチャーは特定のサブスクリプションを特定のシステムまたはシステムのグループと一致させることを優先します。

ただし、マッチャーは常にピンを尊重するわけではありません。 これは、利用可能なサブスクリプション、およびサブスクリプションをクライアントに適用できるかどうかによって異なります。 さらに、サブスクリプションの条項に違反する一致が発生した場合、またはピンが無視された場合にマッチャーがより正確な一致を検出した場合、ピンは無視されます。

新しいピンを追加するには、**[[ピン]の[追]加]**をクリックし、ピンを設定するクライアントを選択します。



定期的に、または多数のクライアントにピンの設定を使用することはお勧めしません。 Subscription Matcherツールは、一般的にほとんどのインストールに十分な精度があります。

Chapter 22. タスクスケジュール

管理 > タスクスケジュールに、すべての事前定義済みのタスクバンチが一覧にされます。

SUSE Manager Schedules [?](#)

Below is a list of defined schedules. A schedule defines frequency, how often a predefined bunch shall be triggered.

1 - 23 of 23

25 [▼](#) items per page

Schedule name ? :	Frequency	Active From	Bunch
auto-errata-default	0 5/10 * * * ?	2018-06-05 11:40:50 CEST	auto-errata-bunch
channel-repo-data-default	0 * * * ?	2018-06-05 11:40:50 CEST	channel-repo-data-bunch
cleanup-data-default	0 0 23 ? * *	2018-06-05 11:40:50 CEST	cleanup-data-bunch
clear-tasklogs-default	0 0 23 ? * *	2018-06-05 11:40:50 CEST	clear-tasklogs-bunch
cobbler-sync-default	0 * * * ?	2018-06-05 11:40:50 CEST	cobbler-sync-bunch
compare-configs-default	0 0 23 ? * *	2018-06-05 11:40:50 CEST	compare-configs-bunch
cve-server-channels-default	0 0 23 ? * *	2018-06-05 11:40:51 CEST	cve-server-channels-bunch
daily-status-default	0 0 23 ? * *	2018-06-05 11:40:50 CEST	daily-status-bunch
errata-cache-default	0 * * * ?	2018-06-05 11:40:50 CEST	errata-cache-bunch
errata-queue-default	0 * * * ?	2018-06-05 11:40:50 CEST	errata-queue-bunch
gatherer-matcher-default	0 0 0 ? * *	2018-06-05 11:40:51 CEST	gatherer-matcher-bunch
kickstart-cleanup-default	0 0/10 * * * ?	2018-06-05 11:40:50 CEST	kickstart-cleanup-bunch
kickstartfile-sync-default	0 0/10 * * * ?	2018-06-05 11:40:50 CEST	kickstartfile-sync-bunch
mgr-register-default	0 0/15 * * * ?	2018-06-05 11:40:50 CEST	mgr-register-bunch
mgr-sync-refresh-default	0 6 1 ? * *	2018-06-05 11:40:51 CEST	mgr-sync-refresh-bunch
minion-action-cleanup-default	0 0 * * * ?	2018-06-05 11:40:50 CEST	minion-action-cleanup-bunch
package-cleanup-default	0 0/10 * * * ?	2018-06-05 11:40:50 CEST	package-cleanup-bunch
reboot-action-cleanup-default	0 0 * * * ?	2018-06-05 11:40:50 CEST	reboot-action-cleanup-bunch
sandbox-cleanup-default	0 5 4 ? * *	2018-06-05 11:40:50 CEST	sandbox-cleanup-bunch
session-cleanup-default	0 0/15 * * * ?	2018-06-05 11:40:50 CEST	session-cleanup-bunch
ssh-push-default	0 * * * ?	2018-06-05 11:40:50 CEST	ssh-push-bunch
token-cleanup-default	0 0 0 ? * *	2018-06-05 11:40:51 CEST	token-cleanup-bunch
uuid-cleanup-default	0 0 * * * ?	2018-06-05 11:40:51 CEST	uuid-cleanup-bunch

SUSE Manager Schedules (SUSE Managerスケジュール) > スケジュール名をクリックして、スケジュール名 > 基本的なスケジュール詳細を開き、無効にしたり、頻度を変更したりできます。 [Edit Schedule] (スケジュールの編集) をクリックして、設定でスケジュールを更新します。 スケジュールを削除するには、右上隅の [スケジュールの削除] をクリックします。



- Uyuniが正常に動作するために不可欠であるため、スケジュールが必要であると確信している場合にのみ、スケジュールを無効にするか削除してください。

バンチ名をクリックすると、そのバンチタイプのランとそのステータスのリストが表示されます。 開始時間のリンクをクリックすると、スケジュール名 > 基本的なスケジュール詳細に戻ります。

たとえば、次の事前定義済みのタスクバンチはデフォルトでスケジュールされおり、設定できます。

channel-repo-data-default:

リポジトリメタデータファイルを(再)生成します。

cleanup-data-default:

古いパッケージ変更ログをクリーンアップし、データベースから時系列データを監視します。

clear-tasklogs-default:

ジョブタイプに応じて、指定した日数よりも古いタスクエンジン(taskomatic)履歴データをデータベースからクリアします。

cobbler-sync-default:

UyuniからCobblerに配布データとプロファイルデータを同期します。 Cobblerによる自動インストールの詳細については、[Client-configuration](#) > [Autoinst-intro](#)を参照してください。

compare-configs-default:

設定チャンネルに保存されている設定ファイルと、すべての設定対応サーバに保存されているファイルを比較します。 比較を確認するには、menu:[システム]タブをクリックし、対象のシステムを選択します。 [設定](#) > [ファイルの比較](#)に移動します。 詳細については、[reference:systems/system-details/configuration.pdf](#)を参照してください。

cve-server-channels-default:

[監査](#) > [CVE監査](#)ページに結果を表示するために使用される、事前に計算された内部CVEデータを更新します。 [監査](#) > [CVE監査](#)ページの検索結果は、このスケジュールの最後の実行に更新されます。 詳細については、[Reference](#) > [Audit](#)を参照してください。

daily-status-default:

関連するアドレスに日次レポート電子メールを送信します。 特定のユーザの通知を設定する方法の詳細については、[Reference](#) > [Users](#)を参照してください。

errata-cache-default:

各サーバの更新が必要なパッケージを検索するために使用される、内部パッチキャッシュデータベーステーブルを更新します。 また、これにより、特定のパッチに関心がある可能性のあるユーザに通知メールが送信されます。 パッチの詳細については、[Reference](#) > [Patches](#)を参照してください。

errata-queue-default:

自動更新(パッチ)を受信するように設定されているサーバのキューに入れます。

kickstart-cleanup-default:

古いキックスタートセッションデータをクリーンアップします。

kickstartfile-sync-default:

設定ウィザードによって作成されたキックスタートプロファイルに対応するCobblerファイルを生成します。

mgr-forward-registration-default:

SUSE Customer Center (SCC)とクライアント登録データを同期します。デフォルトでは、新規、変更済み、または削除済みクライアントデータが転送されます。 `/etc/rhn/rhn.conf`で同期セットを無効にするには、次のコマンドを実行します。

```
server.susemanager.forward_registration = 0
```

mgr-sync-refresh-default:

SUSE Customer Center (SCC)と同期します(**mgr-sync-refresh**)。デフォルトでは、すべてのカスタムチャンネルもこのタスクの一部として同期されます。 カスタムチャンネル同期の詳細については、[administration:custom-channels.pdf](#)を参照してください。

minion-action-cleanup-default:

ファイルシステムから古いクライアントアクションデータを削除します。 最初に、対応する結果を検索して、未完了の可能性のあるアクションを完了しようとします。 これらの結果は、Saltジョブキャッシュに保存されます。 サーバがアクションの結果を見逃した場合、未完了のアクションが発生する可能性があります。 アクションが正常に完了した場合は、実行されたスクリプトファイルなどのアーティファクトが削除されます。

package-cleanup-default:

ファイルシステムから古いパッケージファイルを削除します。

reboot-action-cleanup-default:

6時間以上保留中の再起動アクションは失敗としてマークされ、関連データがデータベース内でクリーンアップされます。 再起動アクションのスケジュール設定の詳細については、[reference:systems/system-details/sd-provisioning.pdf](#)を参照してください。

sandbox-cleanup-default:

`sandbox_lifetime` 設定パラメータ(デフォルトでは3日)よりも古いサンドボックス設定ファイルとチャンネルをクリーンアップします。 サンドボックスファイルは、システムまたは開発中のファイルからインポートされたファイルです。 詳細については、[reference:systems/system-details/sd-configuration.pdf](#)を参照してください。

session-cleanup-default:

古いWebインターフェイスセッションをクリーンアップします。通常は、ユーザがログインし、ログアウトする前にブラウザを閉じたときに一時的に保存されるデータです。

ssh-push-default:

クライアントが`SSH Push`連絡方法で設定されている場合、クライアントにSSH経由でUyuniにチェックインするようにプロンプトを表示します。

token-cleanup-default:

Saltクライアントがパッケージとメタデータをダウンロードするために使用する期限切れのリポジトリトークンを削除します。

Chapter 23. 変更ログの調整

一部のパッケージには、変更ログエントリの長いリストがあります。このデータはデフォルトでダウンロードされますが、必ずしも保持すべき役立つ情報とは限りません。ダウンロードする変更ログのメタデータ量を制限し、ディスク容量を節約するために、ディスクに保持するエントリ数を制限できます。

この設定オプションは、`/etc/rhn/rhn.conf` 設定ファイルにあります。パラメータのデフォルトは `0` で、無制限を意味します。

```
java.max_changelog_entries = 0
```

このパラメータを設定すると、新しいパッケージが同期されるときにのみ有効になります。

このパラメータを変更した後で、`spacewalk-service restart` を使用してサービスを再起動します。

キャッシュされたデータを削除して再生成し、古いデータを削除したい場合があります。



キャッシュされたデータの削除と再生成には時間がかかる場合があります。 使用するチャンネル数と削除するデータ量に応じて、数時間かかる可能性があります。 タスクはTaskomaticによってバックグラウンドで実行されるため、操作が完了するまでUyuniを使用し続けることができますが、多少のパフォーマンスの低下が予想されます。

コマンドラインからキャッシュされたデータを削除して、再生成を要求できます。

```
spacewalk-sql -i
```

次に、SQLデータベースプロンプトで、次のように入力します。

```
DELETE FROM rhnPackageRepodata;
INSERT INTO rhnRepoRegenQueue (id, CHANNEL_LABEL, REASON, FORCE)
(SELECT sequence_nextval('rhn_repo_regen_queue_id_seq'),
C.label,
'cached data regeneration',
'Y'
FROM rhnChannel C);
\q
```

Chapter 24. ユーザー

Uyuni管理者は新しいユーザの追加、許可の付与、ユーザの無効化または削除を行うことができます。 多数のユーザを管理している場合は、ユーザをシステムグループに割り当てて、グループレベルで許可を管理できます。 言語やテーマのデフォルトなど、Web UIのシステムデフォルトを変更することもできます。



[**ユ
ー
ザ**] メニューは、Uyuni管理者アカウントでログインしている場合にのみ使用できます。

Uyuniユーザを管理するには、ユーザ[ユーザー一覧 > すべて]に移動し、Uyuniサーバのすべてのユーザを表示します。 リスト内の各ユーザにはユーザ名、リアル名、割り当てられたロール、ユーザが最後に署名した日付、ユーザの現在のステータスが表示 [**ユ
ー
ザ
ー
の
作
成**] をクリックして、新しいユーザアカウントを作成します。 ユーザ名をクリックして、[User Details] (ユーザ詳細) ページに移動します。

新規ユーザを組織に追加するには、[**ユ
ー
ザ
ー
の
作
成**] をクリックして、新しいユーザの詳細を確認し、[**登
録**] をクリックします。

24.1. アカウントの無効化と削除

ユーザアカウントが不要になった場合は、無効化または削除できます。 無効化されたユーザアカウントは、いつでも再有効化できます。 削除されたユーザアカウントは表示されず、取得できません。

ユーザは独自のアカウントを無効化できます。 ただし、ユーザに管理者ロールがある場合は、アカウントを無効化する前に、ロールを削除する必要があります。

無効化されたユーザはUyuni Web UIにログインしたり、アクションをスケジュール したりできません。 無効化の前にユーザによってスケジュールされたアクションは、アクションキューに残ります。 無効化されたユーザはUyuni管理者によって再有効化できます。

24.2. 管理者ロール

ユーザは複数の管理者ロールを保持できます。 また、任意の管理者ロールを保持するユーザは、いつでも複数存在することができます。 常に少なくとも1人の有効なUyuni管理者が必要です。

Uyuni管理者ロールを除く、ユーザの管理者ロールを変更するには、**ユーザ** > **ユーザー一覧** > **すべて**に移動して、変更するユーザを選択し、必要に応じて管理者ロールをオンまたはオフにします。

ユーザのUyuni管理者ロールを変更するには、**管理者**に移動して、必要に応じて、[**Uyuni 管理者?**] をオンまたはオフにします。

表 14. ユーザ管理者ロールの許可

ロール名	説明
システムグループユーザ	すべてのユーザに関連付けられた標準ロール。
Uyuni管理者	他のユーザの権限の変更を含む、すべての機能を実行できます。

ロール名	説明
組織管理者	アクティベーションキー、設定、チャンネル、およびシステムグループを管理します。
アクティベーションキー管理者	アクティベーションキーを管理します。
イメージ管理者	イメージプロファイル、ビルド、およびストアを管理します。
設定管理者	システム設定を管理します。
チャンネル管理者	チャンネルをグローバルにサブスクライブ可能にしたり、新しいチャンネルを作成したりするなど、ソフトウェアチャンネルを管理します。
システムグループ管理者	システムグループの作成と削除、既存のグループへのクライアントの追加、グループへのユーザアクセスの管理など、システムグループを管理します。

24.3. ユーザ許可とシステム

クライアントを管理するシステムグループを作成している場合は、管理するグループをユーザに割り当てることができます。

ユーザをシステムグループに割り当てるには、**ユーザ > ユーザ一覧**に移動して、編集するユーザ名をクリックし、[システムグループ割り当てる]タブに移動します。して、[デフォルトの更新]をクリックします。

ユーザの1つ以上のデフォルトのシステムグループを選択することもできます。ユーザが新しいクライアントを登録すると、デフォルトで選択したシステムグループに割り当てられます。これにより、ユーザは新たに登録されたクライアントに直ちにアクセスできます。

外部グループを管理するには、**ユーザ > System Group Configuration** (システムグループ設定) に移動して、[External Authentication](外部認証) タブに移動します。[外部グループの作成]をクリックして、新しい外部グループを作成します。グループに名前を付けて、適切なシステムグループに割り当てます。

システムグループの詳細については、**Reference > Systems**を参照してください。

ユーザが管理できる個々のクライアントを確認するには、**ユーザ > ユーザ一覧**に移動し、編集するユーザ名をクリックして、[システム]タブに移動します。タスクを実行するには、リストからクライアントを選択し、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詳細については、**Client-configuration > System-set-manager**を参照してください。

24.4. ユーザとチャンネルの許可

チャンネルからコンテンツを消費するサブスクライバとして、またはチャンネル自体を管理できる管理者として、ユーザを組織内のソフトウェアチャンネルに割り当てることができます。

ユーザをチャンネルにサブスクライブするには、**ユーザ** > **ユーザー**一覧に移動し、編集するユーザ名をクリックして、**チャンネルの権限**タブに移動します。割り当てるチャンネルをオンにして、**許可の更新**をクリックします。

ユーザにチャンネル管理許可を付与するには、**ユーザ** > **ユーザー**一覧に移動し、編集するユーザ名をクリックして、**チャンネルの管理**タブに移動します。割り当てるチャンネルをオンにして、**許可の更新**をクリックします。

リスト内的一部のチャンネルはサブスクライブできない場合があります。これは通常、ユーザ管理者のステータス、またはチャンネルのグルーバ設定が原因です。

24.5. ユーザのデフォルト言語

新しいユーザを作成するときに、Web UIで使用する言語を選択できます。ユーザを作成した後で、**ホーム** > **設定**に移動して、言語を変更できます。

デフォルト言語は、**rhn.conf**設定ファイルで設定されます。
デフォルト言語を変更するには、**/etc/rhn/rhn.conf**ファイルを開いて、次の行を追加または編集します。

```
web.locale = <LANGCODE>
```

パラメータが設定されていない場合、デフォルト言語は**en_US**です。

Uyuniで使用可能な言語は次のとおりです。

表 15. 使用可能な言語コード

言語コード	言語	ダイアレクト
bn_IN	バングラ語	インド
ca	カタロニア語	
de	ドイツ語	
en_US	英語	米国
es	スペイン語	
fr	フランス語	
gu	グジャラート語	
hi	ヒンディー語	
it	イタリア語	
ja	日本語	
ko	韓国語	
pa	パンジャブ語	
pt	ポルトガル語	

言語コード	言語	ダイアレクト
pt_BR	ポルトガル語	ブラジル
ru	ロシア語	
ta	タミル語	
zh_CN	中国語	本土、簡体字
zh_TW	中国語	台湾、繁体字



Uyuniの翻訳はコミュニティによって提供されており、不正確または不完全である可能性があります。翻訳が利用できない場合、Web UIはデフォルトで英語(en_US)になります。

24.5.1. ユーザデフォルトのインターフェイステーマ

デフォルトでは、Uyuni Web UIはインストールした製品に適切なテーマを使用します。テーマを変更して、UyuniまたはSUSE Managerの色を反映できます。SUSE Managerのテーマでは、ダークオプションも使用できます。

rhn.conf 設定ファイルでデフォルトのテーマを変更できます。デフォルトのテーマを変更するには、**/etc/rhn/rhn.conf** ファイルを開いて、次の行を追加または編集します。

```
web.theme_default = <THEME>
```

表 16. 使用可能なWebUIテーマ

テーマ名	色	スタイル
susemanager-light	SUSE Manager	ライト
susemanager-dark	SUSE Manager	ダーク
uyuni	Uyuni	ライト

Chapter 25. トラブルシューティング

このセクションには、Uyuniで発生する可能性のある共通の問題、およびそれらの問題を解決するためのソリューションが含まれています。

25.1. 自動インストールのトラブルシューティング

ベースチャンネルによっては、新しい自動インストールプロファイルは、必要なパッケージがないチャンネルにサブスクライブされる場合があります。

自動インストールを動作させるには、次のパッケージが必要です。

- `pyOpenSSL`
- `rhnlib`
- `libxml2-python`
- `spacewalk-koan`

この問題を解決するには、まず次の点について確認してください。

- 自動インストールプロファイルのベースチャンネルに関するツールソフトウェアチャンネルを組織およびユーザーで使用できることを確認します。
- ツールチャンネルが子チャンネルとしてUyuniで使用できることを確認します。
- 関連するチャンネルで、必要なパッケージと依存関係が使用可能であることを確認します。

25.2. ベアメタルシステムのトラブルシューティング

ネットワークのベアメタルシステムが [システム] リストに自動的に追加されない場合、まず次の点について確認してください。

- `pxe-default-image` パッケージがインストールされている必要があります。
- ファイルのパスおよびパラメータが正しく設定されている必要があります。`rhn.conf` 設定ファイルで指定された場所に、`pxe-default-image` で提供される `vmlinuz0` ファイルと `initrd0.img` ファイルがあることを確認します。
- ベアメタルシステムをUyuniサーバに接続するネットワーク機器が正しく動作していて、そのサーバからUyuniサーバのIPアドレスにアクセスできることを確認します。
- プロビジョニングするベアメタルシステムは、ブートシーケンスでPXEブートが有効になっている必要があります、オペレーティングシステムをブートしていない必要があります。
- DHCPサーバは、ブート中にDHCPリクエストに応答する必要があります。PXEブートメッセージで次の点を確認してください。
 - DHCPサーバが目的のIPアドレスを割り当てている。
 - DHCPサーバがUyuniサーバのIPアドレスをブート用に `next-server` として割り当てている。

- Cobblerが実行中であり検出機能が有効なことを確認してください。

ブート後すぐに青いCobblerメニューが表示される場合、検出が開始されています。 正常に完了しない場合、自動シャットダウンを一時的に無効にして、問題の診断に活用してください。自動シャットダウンを無効にするには:

1. Cobblerメニューで矢印キーを使用して `pxe-default-profile` を選択し、タイマーが切れる前にTabキーを押します。
2. カーネルブートパラメータ `spacewalk-finally=running` を追加します。そのためには、統合されたエディタを使用します。その後、Enterキーを押してブートを続行します。
3. ユーザ名を `root`、パスワードを `linux` にしてシェルに入力し、デバッグを続行します。



重複プロファイル

技術的な制約により、新しいベアメタルシステムを以前に検出されたシステムと高い信頼性で区別することはできません。 したがって、ベアメタルシステムの電源を複数回オンにしないことをお勧めします。この操作を実行するとプロファイルの重複が発生します。

25.3. サポート終了製品のブートストラップリポジトリのトラブルシューティング

サポートされている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。 製品がサポート終了になり、サポートされなくなったが、製品の使用を継続する場合、ブートストラップリポジトリを手動で作成する必要があります。

ブートストラップリポジトリの詳細については、[Client-configuration > Bootstrap-repository](#)を参照してください。

プロシージャ: サポート終了製品のブートストラップリポジトリの作成

1. Uyuniサーバのコマンドプロンプトで、`root`として、`--force` オプションを指定して使用可能なサポート対象外のブートストラップリポジトリの一覧を表示してください。たとえば下記のようになります:

```
mgr-create-bootstrap-repo --list --force
1. SLE-11-SP4-x86_64
2. SLE-12-SP2-x86_64
3. SLE-12-SP3-x86_64
```

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

```
mgr-create-bootstrap-repo --create SLE-12-SP2-x86_64 --force
```

ブートストラップリポジトリを手動で作成しない場合、必要な製品およびブートストラップリポジトリ

でLTSSが使用できるかどうかを確認できます。

25.4. クライアントが複製したSaltクライアントのトラブルシューティング

ハイパーバイザ複製ユーティリティを使用していて、複製したSaltクライアントを登録しようとすると、次のエラーが発生します。

残念ながら、このシステムは見つかりませんでした。

新しい複製システムのマシンIDが既存の登録済みシステムのマシンIDと同じことが原因です。 マシンIDを手動で調整してエラーに対処すると、複製したシステムを正常に登録できます。

詳細および手順については、[Administration > Troubleshooting](#)を参照してください。

25.5. 破損したリポジトリのトラブルシューティング

リポジトリメタデータファイルの情報が破損したり、古くなったりする可能性があります。 これにより、クライアントの更新で問題が発生する可能性があります。 これは、ファイルを削除して再生成することで修正できます。 新しいリポジトリデータファイルを使用すると、更新が期待どおりに動作するはずです。

プロシージャ: 破損したリポジトリデータの解決

1. `/var/cache/rhn/repoadata/<channel-label>-updates-x86_64` からすべてのファイルを削除します。 チャンネルラベルがわからない場合は、Uyuni Web UIでソフトウェア > チャンネル > チャンネルラベルに移動して検索できます。
2. コマンドラインからファイルを再生成します。

```
spacecmd softwarechannel_regenerateyumcache <channel-label>-updates-x86_64
```

25.6. パッケージが競合するカスタムチャンネルのトラブルシューティング

パッケージが競合するカスタムチャンネルを設定する場合、ブートストラップリポジトリの作成などの機能が未定義の動作を引き起こし、クライアントの登録に失敗する可能性があります。

たとえば、バージョン番号がより新しい競合するパッケージがブートストラップリポジトリに含まれる可能があります。このようなパッケージ(たとえば、`python3-zmq` や `zeromq`)により、ブートストラップリポジトリの作成が破損したり、クライアントのブートストラップ中に問題が発生する可能性があります。

カスタムチャンネル(たとえば、EPELチャンネル)が親ベンダチャンネルの下に追加されると、パッケージの競合に関する問題を直接解決できません。 これを解決する方法は、カスタムチャンネルをベンダチャンネルから分離する方法です。 カスタムチャンネルを別のツリーで作成する必要があります。 カスタムチャンネル

を子として配信する必要がある場合は、このような環境をコンテンツライフサイクル管理(CLM)を使用して作成できます。 CLMプロジェクトのソースは別のツリーからそこに追加できます。 このようなアプローチを使用すると、カスタムチャンネルは構築された環境内で親の下に維持されます。 ただし、ベンダチャンネルツリーはカスタムチャンネルとブートストラップリポジトリなしで維持されます。 その後、クライアントの登録は正しく機能します。

競合するパッケージ(salt、 zeromqなど)を持つカスタムチャンネルを子チャンネルとして作成する場合は、次のステップに従うことで問題を回避できます。

プロシージャ: カスタムチャンネルで競合するパッケージを回避する

1. カスタムチャンネルを親チャンネルから子チャンネルとして削除します。 詳細については、[administration:custom-channels.pdf](#)を参照してください。
2. 別のツリーでカスタムチャンネルを作成します。 詳細については、[administration:custom-channels.pdf](#)を参照してください。
3. コンテンツライフサイクル管理(CLM)内で子チャンネルとしてカスタムチャンネルを取得するには、次の手順に従います。
 - UyuniのWeb UIで、Content Lifecycleに移動し、[Create Project]をクリックします。 Name と Label を入力します。
 - ソースをプロジェクトに割り当てます。 必要なベンダチャンネルとカスタムチャンネルを使用します。 // (CentOS8を使用した共有例)
 - プロジェクトに環境を追加します。 // CentOS8を使用した例
 - 環境を構築するには、[ビルド]ボタンをクリックします。これにより、アクティベーションキーに関連付けて、クライアントのブートストラップに使用できるベンダチャンネルとカスタムチャンネルを備えた環境が作成されます。
4. 重要なメモ: CLMプロジェクトでは、問題のあるパッケージや競合するパッケージを除外するフィルタを追加することをお勧めします。これを追加しないと、より新しいバージョン番号の競合するパッケージがクライアントの更新中にインストールされます。 フィルタリングの詳細については[administration:content-lifecycle-examples.pdf](#)を参照してください。
5. 最新のパッチをCLM環境(ベンダチャンネルおよびカスタムチャンネルを使用)に取得するには、プロジェクトの[ビルド]ボタンをクリックします。これは、環境を再構築するために必要です。
 - CLM の詳細については、Administration > Content-lifecycle を参照してください。
 - 別の回避策については、https://www.suse.com/releasenotes/x86_64/SUSE-MANAGER/4.2/index.html#_epel_and_salt_packagesを参照してください。

25.7. FQDNS grainの無効化のトラブルシューティング

FQDNS grainは、システムのすべての完全修飾DNSサービスのリストを返します。 この情報の収集は、通常、高速プロセスですが、DNS設定が間違っていると、長時間かかる可能性があります。 場合によっては、クライアントが無応答またはクラッシュする場合があります。

この問題を回避するには、Saltフラグを使用してFQDNS grainを無効にできます。 grainを無効にした場合、ネットワークモジュールを使用して、FQDNSサービスを提供できます。 この場合、クライアントが無応答になるリスクはありません。



- この操作は、古いSaltクライアントにのみ適用されます。 最近Saltクライアントを登録した場合、FQDN grainはデフォルトで無効になっています。

Uyuniサーバのコマンドプロンプトで、次のコマンドを使用してFQDN grainを無効にします。

```
salt '*' state.sls util.mgr_disable_fqdns_grain
```

このコマンドを実行すると、各クライアントが再起動され、サーバが処理する必要があるSaltイベントが生成されます。 クライアント数が多い場合、バッチモードでコマンドを実行できます。

```
salt --batch-size 50 '*' state.sls util.mgr_disable_fqdns_grain
```

バッチコマンドの実行完了を待機します。 **Ctrl**+**C**でプロセスを中断しないでください。

25.8. ディスク容量のトラブルシューティング

ディスク容量が不足すると、Uyuniデータベースとファイル構造に重大な影響を及ぼす可能性があり、ほとんどの場合、回復できません。 Uyuniでは特定のディレクトリの空き容量を監視し、設定可能なアラートを用意しています。 スペース管理の詳細については、Administration > Space-managementを参照してください。

未使用的ソフトウェアチャンネルを削除することで、ディスク容量を回復できます。 ベンダチャンネルを削除する方法については、Administration > Channel-managementを参照してください。 また、カスタムチャンネルを削除する方法については、Administration > Custom-channelsを参照してください。

カスタムチャンネルが同期される頻度を確認することもできます。 カスタムチャンネル同期の処理方法については、[administration:custom-channels.pdf](#)を参照してください。

未使用的アクティベーションキー、コンテンツライフサイクルプロジェクト、およびクライアント登録をクリーンアップすることで、ディスク容量を回復することもできます。 また、冗長なデータベースエントリを削除することもできます。

プロシージャ: 冗長なデータベースエントリの解決

1. **spacewalk-data-fsck** コマンドを使用して、冗長なデータベースエントリを一覧にします。
2. **spacewalk-data-fsck --remove** コマンドを使用して削除します。

25.9. ファイアウォールのトラブルシューティング

送信トラフィックをブロックするファイアウォールを使用している場合は、**REJECT** または **DROP** のいずれかのネットワーク要求を実行できます。 '**DROP**' に設定されている場合、SUSE Customer Centerとの同期がタイムアウトする可能性があります。

これは、同期プロセスがSUSE Customer Centerだけではなく、SUSE以外のクライアント用のパッケージを

提供するサードパーティリポジトリにアクセスする必要があるために発生します。 Uyuniサーバがこれらのリポジトリに到達して有効であることを確認しようとすると、ファイアウォールは要求をドロップし、同期はタイムアウトするまで応答を待ち続けます。

これが発生する場合、同期が失敗するまで長時間かかり、SUSE以外の製品が製品リストに表示されません。

この問題はさまざまな方法で修正できます。

最も簡単な方法は、SUSE以外のリポジトリで必要なURLへのアクセスを許可するようにファイアウォールを設定することです。これにより、同期プロセスがURLに到達し、正常に完了することができます。

外部トラフィックを許可できない場合は、Uyuniからの **REJECT (DROP ではない)** 要求を行うようにファイアウォールを設定してください。これにより、サードパーティURLへの要求が拒否されるため、同期はタイムアウトではなく早期に失敗し、製品はリストに表示されません。

ファイアウォールへの設定アクセス権がない場合は、代わりにUyuniサーバに別のファイアウォールを設定することを検討してください。

25.10. WAN接続を介したUyuniサーバとプロキシ間の長い同期時間に関するトラブルシューティング

WebUIで、あるいは配布またはシステム設定へのAPIコールを介して実行される変更によって、UyuniサーバからUyuniプロキシシステムにファイルを転送するために、**cobbler sync**コマンドが必要になる場合があります。これを実現するために、cobblerは **/etc/cobbler/settings** で指定されたプロキシのリストを使用します。

cobbler syncは、その設計上、変更されたファイルや最近追加されたファイルのみを同期することはできません。

代わりに、**cobbler sync**実行すると、**/etc/cobbler/settings** で設定された指定のすべてのプロキシに **/srv/tftpboot** ディレクトリの完全同期がトリガされます。また、関連するシステム間のWAN接続の遅延によっても影響を受けます。

/var/log/cobbler/ のログによると、同期のプロセスが完了するまでにかなりの時間がかかる場合があります。

たとえば、次の日時に開始したとします。

```
Thu Jun  3 14:47:35 2021 - DEBUG | running python triggers from
/var/lib/cobbler/triggers/task/sync/pre/*
Thu Jun  3 14:47:35 2021 - DEBUG | running shell triggers from
/var/lib/cobbler/triggers/task/sync/pre/*
```

そして、次の日時に終了したとします。

```
Thu Jun 3 15:18:49 2021 - DEBUG | running shell triggers from
/var/lib/cobbler/triggers/task-sync/post/*
Thu Jun 3 15:18:49 2021 - DEBUG | shell triggers finished successfully
```

転送量は約1.8GBでした。転送には30分ほどかかりました。

比較すると、**/srv/tftboot**と同じサイズの大きな単一ファイルのコピーは、数分以内に完了します。

Uyuniサーバとプロキシ間でファイルをコピーするために**rsync**ベースのアプローチに切り替えると、転送時間と待機時間を短縮できる場合があります。

このタスクを実行するためのスクリプト

は、https://suse.my.salesforce.com/sfc/p/1i000000gLOd/a/1i000000ll5B/B2AmvIJN2_JsAyjTQzCVP_x5ioVgd0bYN9X9NpMugS8でダウンロードできます。

このスクリプトはコマンドラインオプションを受け入れません。スクリプトを実行する前に、手動で編集し、**SUMAHOSTNAME SUMAIP**、および**SUMAPROXY1**変数を正しく設定して、スクリプトが正しく機能するようにする必要があります。



スクリプトの個々の調整に利用可能なサポートはありません。スクリプトと内部のコメントは、プロセスの概要と考慮すべきステップを提供することを目的としています。さらにサポートが必要な場合は、SUSEコンサルティングにお問い合わせください。

スクリプトを使用した提案されるアプローチは、次の環境で役立ちます。

- ・ SUSE Manager ProxyシステムがWAN接続を介して接続されている。
- ・ **/srv/tftboot**に多数のディストリビューション用ファイルおよびクライアントPXEブートファイル(合計数千ファイル)が含まれている。
- ・ **/etc/cobbler/settings**の任意のプロキシは無効になっているが、それ以外の場合、Uyuniは引き続きプロキシとコンテンツを同期する。

```
#proxies:
# - "sumaproxy.sumaproxy.test"
# - "sumaproxy2.sumaproxy.test"
```

プロシージャ: 新しい同期速度の分析

1. Uyuniと関連するシステム間のTCPトラフィックのダンプを取得します。

- SUSE Managerサーバの場合:

```
tcpdump -i ethX -s 200 host <ip-address-of-susemanagerproxy> and
not ssh
```

- SUSE Managerプロキシの場合:

```
tcpdump -i ethX -s 200 host <ip-address-of-susemanager> and not
ssh
```

- これにより、分析を実行するのに十分な200のパッケージサイズのみがキャプチャされます。
- プロキシと通信するためにUyuniが使用する各ネットワークインターフェイスにehtXを調整します。
- 最後に、さらにパッケージ数を削減するため、ssh通信はキャプチャされません。

2. cobbler syncを開始します。

- 同期を強化するため、最初にcobbler jsonキャッシュファイルを削除してから、cobbler syncを発行します。

```
rm /var/lib/cobbler/pxe_cache.json
cobbler sync
```

3. cobblerが終了したら、TCPdumpsを停止します。

4. Wiresharkを使用してTCPdumpsを開き、[Statistics] (統計) 計 Conversations (対話)]に移動して、ダンプが分析されるのを待ちます。

5. TCPタブに切り替えます。このタブに表示される数は、SUSE ManagerとSUSE Manager Proxy間でキャプチャされた合計対話数を示しています。

6. [Duration] (期間) 列を探します。

- 昇順にソートして開始し、ファイルの転送にかかった最小時間を確認します。
- 降順にソートして続行し、カーネルやinitrdの転送など、大きなファイルの最大値を確認します。



ポート4505と4506はSalt通信に使用されるため無視してください。

TCPdumpsの分析では、Uyuniサーバからプロキシへの、サイズが約1800バイトの小さなファイルの転送に約0.3秒かかったことを示しています。

大きなファイルは多くありませんでしたが、小さなファイルの数が多いため、転送されるファイルごとに新しいTCP接続が作成され、確立された接続数が多くなりました。

したがって、最小転送時間と必要な接続数(たとえば、約5000)がわかれば、転送時間全体の概算推定時間が得られます($5000 * 0.3 / 60 = 25$ 分)。

25.11. 無効なクライアントのトラブルシューティング

Taskomaticジョブは、クライアントが接続されていることを確認するため、クライアントに定期的にpingを送信します。 クライアントが24時間以上Taskomaticのチェックインに応答しない場合は、無効であるとみなされます。 Web UIで無効なクライアントのリストを表示するには、[システム > システム一覧 > 無効](#)に移動します。

クライアントはさまざまな理由で無効になる可能性があります。

- ・ クライアントにUyuniサービスへのエンタイトルメントが付与されていない。 クライアントが180日間(6ヶ月間)エンタイトルメントを付与されないままの場合、クライアントが削除される。
- ・ 従来のクライアントで、**rhnsd** サービスが無効化されている。
- ・ クライアントがHTTPS接続を許可しないファイアウォールの背後にある。
- ・ クライアントが誤って設定されたプロキシの背後にある。
- ・ クライアントが異なるUyuniサーバと通信しているか、接続が正しく設定されていない。
- ・ クライアントがUyuniサーバと通信できるネットワーク内にない。
- ・ ファイアウォールがクライアントとUyuniサーバ間のトラフィックをブロックしている。
- ・ Taskomaticが正しく設定されていない。

サーバへのクライアント接続の詳細については、[Client-configuration > Contact-methods-intro](#)を参照してください。

ports 設定の詳細については、[Installation-and-upgrade > Ports](#) を参照してください。

ファイアウォールのトラブルシューティングの詳細については、[Administration > Troubleshooting](#)を参照してください。

25.12. サーバ間同期のトラブルシューティング

サーバ間同期では、キャッシングを使用してISSマスターとスレーブを管理します。 これらのキャッシングには、無効なエントリを作成するバグが発生する可能性があります。 この場合、キャッシングがまだ無効なエントリを使用しているため、バグを解決するバージョンに更新した後でもバグが表示される可能性があります。 新しいバージョンのISSにアップグレードしても問題が解決しない場合は、すべてのキャッシングをクリアして、問題の原因となる古いエントリがないことを確認します。

キャッシングエラーにより、さまざまなエラーで同期が失敗する可能性がありますが、エラーメッセージは通常、次のような内容をレポートします。

```
consider removing satellite-sync cache at /var/cache/rhn/satsync/* and
re-run satellite-sync with same options.
```

これを解決するには、ISSマスターとISSスレーブでキャッシングを削除して、同期が正常に完了するようにします。

プロシージャ: ISSキャッシュエラーの解決

- ISSマスターのコマンドプロンプトで、rootとして、マスターのキャッシュファイルを削除します。

```
rm -rf /var/cache/rhn/xml-*
```

- サービスを再起動します。

```
rcaapache2 restart
```

- ISSマスターのコマンドプロンプトで、rootとして、スレーブのキャッシュファイルを削除します。

```
rm -rf /var/cache/rhn/satsync/*
```

- サービスを再起動します。

```
rcaapache2 restart
```

25.13. ローカル発行者証明書のトラブルシューティング

一部の古いブートストラップスクリプトは、ローカル証明書へのリンクを間違った場所に作成します。これにより、zypperはローカルの発行者証明書に関する **Unrecognized error** を返します。/etc/ssl/certs/ディレクトリをチェックすることにより、ローカル発行者証明書へのリンクが正しく作成されていることを確認できます。この問題が発生した場合は、zypperが期待どおりに動作するようにブートストラップスクリプトを更新することを検討する必要があります。

25.14. ログインタイムアウトのトラブルシューティング

デフォルトでは、Uyuni Web UIはユーザに30分後に再度ログインするように要求します。環境によっては、ログインタイムアウトの値を調整したい場合があります。

値を調整するには、**rhn.conf** と **web.xml** の両方で変更する必要があります。/etc/rhn/rhn.conf で秒単位の値、**web.xml** で分単位の値を設定してください。この2つの値は同じ時間に等しくなる必要があります。

たとえば、タイムアウト値を1時間に変更するには、**rhn.conf** の値を3600秒に設定し、**web.xml** の値を60分に設定します。

プロシージャ: Web UIログインタイムアウト値の調整

- サービスを停止します。

```
spacewalk-service stop
```

2. `/etc/rhn/rhn.conf`を開いて、次の行を追加または編集して、秒単位の新しいタイムアウト値を含めます。

```
web.session_database_lifetime = <Timeout_Value_in_Seconds>
```

3. ファイルを保存して閉じます。
4. `/srv/tomcat/webapps/rhn/WEB-INF/web.xml`を開いて、次の行を追加または編集して、分単位の新しいタイムアウト値を含めます。

```
<session-timeout>Timeout_Value_in_Minutes</session-timeout>
```

5. ファイルを保存して閉じます。
6. サービスを再開します。

```
spacewalk-service start
```

25.15. メール設定のトラブルシューティング

安全なメール通信にするため、認証を有効にし、ユーザ名とパスワードを定義して、`/etc/rhn/rhn.conf`で`SSL`または`STARTTLS`を有効にすることができます。

```
java.smtp_port = integer (default: 25)
java.smtp_auth = true/false (default: false)
java.smtp_ssl = true/false (default: false)
java.smtp_starttls = true/false (default: false)
java.smtp_user = string (default: null)
java.smtp_pass = string (default: null)
```

25.16. noexecで/tmpをマウントする場合のトラブルシューティング

Saltはクライアントのファイルシステム上の`/tmp`からリモートコマンドを実行します。したがって、`noexec`オプションを指定して`/tmp`をマウントしないでください。この問題を解決する別の方法は、一時ディレクトリパスをSaltサービスに指定された`TMPDIR`環境変数で上書きして、`noexec`オプションが設定されていないディレクトリを指すようにすることです。Salt Bundleを使用する場合はsystemdドロップイン設定ファイル`/etc/systemd/system/venv-salt-minion.service.d/10-TMPDIR.conf`を使用し、クライアントで`salt-minion`を使用する場合は`/etc/systemd/system/salt-minion.service.d/10-TMPDIR.conf`を使用する

ことをお勧めします。 ドロップイン設定ファイルの内容の例を次に示します。

```
[Service]
Environment=TMPDIR=/var/tmp
```

25.17. noexecで/var/tmpをマウントする場合のトラブルシューティング

Salt SSHは `/var/tmp` を使用して Salt Bundle を配備し、バンドルされた Python を使用して クライアント上で Salt コマンドを実行します。したがって、`noexec` オプションを使用して `/var/tmp` をマウントしないでください。 It is not possible to bootstrap the clients, `noexec` オプションを使用してマウントした `/var/tmp` がある 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。これは、ブートストラッププロセスが Salt SSH を 使用して クライアントに接続するためです。

25.18. 十分なディスク容量がない場合のトラブルシューティング

移行を開始する前に、使用できるディスク容量を確認してください。
別々の XFS ファイルシステムで `/var/spacewalk` と `/var/lib/pgsql` を探すことをお勧めします。

別々のファイルシステムを設定している場合、`/etc/fstab` を編集し、`/var/lib/pgsql` サブボリュームを削除します。サーバを再起動して変更を取得します。

アップグレードの問題の詳細については、移行ログファイルを確認してください。ログファイルは、アップグレードしているシステムの `/var/log/rhn/migration.log` にあります。

25.19. 通知のトラブルシューティング

通知メッセージのデフォルトの有効期間は30日です。その期間が過ぎると、メッセージは読み込みステータスに関係なくデータベースから削除されます。この値を変更するには、`/etc/rhn/rhn.conf` で次の行を追加または編集します。

```
java.notifications_lifetime = 30
```

通知タイプを有効または無効にするには、`/etc/rhn/rhn.conf` で次のように行を追加または編集します。

```
java.notifications_type_disabled =
OnboardingFailed,ChannelSyncFailed,ChannelSyncFinished
```

デフォルト設定と設定オプションについては、`usr/share/rhn/config-defaults/rhn_java.conf` テンプレートファイルを参照してください。

25.20. OSADとjabberdのトラブルシューティング

場合によっては、jabberが開くことができる最大ファイル数は、接続されているOSADクライアントの数よりも少ないことがあります。

この場合、OSADクライアントはSUSE Managerサーバに接続できず、jabberdはポート5222での応答に時間がかかりすぎます。



この修正は、OSADを使用して接続されている8192台を超えるクライアントがある場合にのみ必要です。この場合は、代わりにSaltクライアントの使用を検討することをお勧めします。大規模インストールのチューニングの詳細については、[Specialized-guides > Salt](#)を参照してください。

jabberdローカル設定ファイルを編集して、jabberに使用可能なファイル数を増やすことができます。デフォルトで、ファイルは `/etc/systemd/system/jabberd.service.d/override.conf` にあります。

プロシージャ: 最大ファイル数の調整

1. コマンドプロンプトで、rootとして、編集用にローカル設定ファイルを開きます。

```
systemctl edit jabberd
```

2. 次のセクションを追加または編集します。

```
[Service]
LimitNOFILE=<soft_limit>:<hard_limit>
```

選択した値は環境によって異なります。たとえば、9500のクライアントがある場合は、ソフト値を100増やして9600に、ハード値を1000増やして10500にします。

```
[Unit]
LimitNOFILE=
LimitNOFILE=9600:10500
```

3. ファイルを保存し、エディタを終了します。



systemctlファイルのデフォルトのエディタはvimです。ファイルを保存して終了するには、`Esc`を押して、[通常]モードに入り、`:wq`と入力し`Enter`を押します。

`/etc/jabberd/c2s.xml` の`max_fds`パラメータも更新してください。例: `<max_fds>10500</max_fds>`

ソフトファイルの制限は、1つのプロセスで開かれているファイルの最大数です。Uyuniでは、最も消費量の

多いプロセスは `c2s` であり、これはクライアントごとに接続を開きます。 ここでは、`c2s` が正しく動作するために必要な非接続ファイルに対応するために、100個のファイルが追加されます。 ハードリミットは、jabberに属するすべてのプロセスに適用され、ルータの `c2s` プロセスと `sm` プロセスからの開かれたファイルも考慮されます。

25.21. パッケージの不整合のトラブルシューティング

クライアント上のパッケージがロックされている場合、Uyuniサーバは適用可能なパッチのセットを正しく判断できない場合があります。 この場合、パッケージの更新はWeb UIで使用できますが、クライアントには表示されず、クライアントを更新しようとすると失敗します。 パッケージのロックと除外リストをチェックして、クライアント上でパッケージがロックされているか除外されているかを判断します。

クライアント上で、パッケージのロックと除外リストをチェックして、パッケージがロックされているか、除外されているかを判断します。

- ・ 拡張サポートプラットフォームでは、`/etc/yum.conf` をチェックして、`exclude=` を検索します。
- ・ SUSE Linux EnterpriseおよびopenSUSEでは、`zypper locks` コマンドを使用します。

25.22. プロキシ経由のリポジトリの問題のトラブルシューティング

場合によっては、プロキシの `squid` キャッシュが破損することがあります。 これが発生すると、プロキシに接続されているクライアント上のパッケージまたはリポジトリのメタデータの取得に失敗し、さまざまなエラーメッセージが表示されます。

`squid` キャッシュのクリーニングは、通常のプロキシとコンテナプロキシでは異なる方法で実行されます。

通常のプロキシの場合は、プロキシマシンで、次のプロシージャに従います。

```
systemctl stop squid
rm -rf /var/cache/squid
systemctl start squid
```

`podman` で実行されているコンテナプロキシの場合は、ホストマシンで、次のプロシージャに従います。

```
systemctl stop uyuni-proxy-pod
podman volume rm uyuni-proxy-squid-cache
systemctl start uyuni-proxy-pod
```

25.23. grainを開始イベントに渡す場合のトラブルシューティング

Saltクライアントは、起動するたびに、`machine_id` grainをUyuniに渡します。 Uyuniは、このgrainを使用して、クライアントが登録されたかどうかを判定します。 このプロセスでは、同期Saltコールが必要です。

同期Saltコールは、その他のプロセスをブロックするため、多数のクライアントを同時に起動する場合、大幅な遅延が発生する可能性があります。

この問題を克服するために、別々のSaltコールを回避するための新しい機能がSaltに導入されました。

この機能を使用するには、この機能をサポートしているクライアントのクライアント設定に設定パラメータを追加できます。

このプロセスを簡単にするには、`mgr_start_event_grains.sls` ヘルパー Salt の状態を使用します。



この操作は、登録済みのクライアントにのみ適用されます。最近Saltクライアントを登録した場合、この設定パラメータはデフォルトで追加されています。

Uyuniサーバのコマンドプロンプトで、次のコマンドを使用して `start_event_grains` 設定ヘルパーを有効にします。

```
salt '*' state.sls util.mgr_start_event_grains
```

このコマンドを実行すると、必要な設定がクライアントの設定ファイルに追加され、クライアントを再起動したときに適用されます。クライアント数が多い場合、バッチモードでコマンドを実行できます。

```
salt --batch-size 50 '*' state.sls mgr_start_event_grains
```

25.24. プロキシの接続およびFQDNのトラブルシューティング

プロキシ経由で接続されているクライアントがWeb UIに表示されることがあります、そのようなクライアントがプロキシ経由で接続されていることは示されません。完全修飾ドメイン名(FQDN)を使用して接続していない場合、Uyuniでプロキシが認識されないと、この動作が発生することがあります。

この動作を修正するには、プロキシのクライアント設定ファイルでgrainとして追加のFQDNを指定します。

```
grains:
  susemanager:
    custom_fqdns:
      - name.one
      - name.two
```

25.25. クローンクライアントの登録のトラブルシューティング

Uyuniを使用して仮想マシンを管理している場合は、仮想マシンのクローンを作成すると役立つ場合があります。クローンとは、既存のディスクの正確なコピーであるプライマリディスクを使用する仮想マシンのことです。

仮想マシンのクローンを作成すると時間を大幅に節約できますが、ディスク上の識別情報が重複しているために問題が発生する可能性があります。

すでに登録されているクライアントがある場合は、そのクライアントのクローンを作成してから、クローンを登録しようとすると、おそらく、Uyuniでそれらを2つの別々のクライアントとして登録する必要があります。ただし、元のクライアントとクローンのマシンIDが同じ場合、Uyuniは両方のクライアントを1つのシステムとして登録し、既存のクライアントデータはクローンのデータで上書きされます。

これは、Uyuniが2つの別のクライアントとして認識できるように、クローンのマシンIDを変更することで解決できます。



このプロセージャの各ステップはクローンクライアントで実行されます。このプロセージャではUyuniに登録されたままである、元のクライアントを操作しません。

プロセージャ: 複製されたSaltクライアントでの重複するマシンIDを解決する

1. クローンマシンで、ホスト名とIPアドレスを変更します。 `/etc/hosts` に加えられた変更と正しいホストエントリが含まれていることを確認します。
2. systemdをサポートするディストリビューションの場合: マシンに同じマシンIDがある場合は、rootとして、複製された各クライアントのファイルを削除し、再作成します。

```
rm /etc/machine-id
rm /var/lib/dbus/machine-id
rm /var/lib/zypp/AnonymousUniqueId
dbus-uuidgen --ensure
systemd-machine-id-setup
```

3. systemdをサポートしないディストリビューションの場合: rootとして、dbusからマシンIDを生成します。

```
rm /var/lib/dbus/machine-id
rm /var/lib/zypp/AnonymousUniqueId
dbus-uuidgen --ensure
```

4. クライアントがまだ同じSaltクライアントIDを持っている場合は、各クライアントの `minion_id` ファイルを削除します(FQDNはクライアントの再起動時に再生成されるときに使用されます)。 Salt Minionクライアントの場合:

```
rm /etc/salt/minion_id
rm -rf /etc/salt/pki
```

Salt Bundleクライアントの場合:

```
rm /etc/venv-salt-minion/minion_id
rm -rf /etc/venv-salt-minion/pki
```

5. オンボーディングページから受諾されたキーを削除し、Uyuniからシステムプロファイルを削除して、次のコマンドを使用してクライアントを再起動します。

```
service salt-minion restart
```

6. クライアントを再登録します。クライアントは異なる **/etc/machine-id** を持つようになり、[システムの概要] ページに正しく表示されるはずです。

プロシージャ: 複製された従来のクライアントでの重複するマシンIDを解決する

1. クローンマシンで、ホスト名とIPアドレスを変更します。 **/etc/hosts** に加えられた変更と正しいホストエントリが含まれていることを確認します。
2. 次のコマンドを使用して、**rhnsd** デーモンを停止します。Red Hat Enterprise Linux Server 6およびSUSE Linux Enterprise 11の場合:

```
/etc/init.d/rhnsd stop
```

または、より新しいsystemdベースのシステムの場合:

```
service rhnsd stop
```

3. 次のコマンドを使用して **osad** を停止します。

```
/etc/init.d/osad stop
```

または

```
service osad stop
```

または

```
rcosad stop
```

4. **osad** 認証設定ファイルとシステムIDを削除します。

```
rm -f /etc/sysconfig/rhn/{osad-auth.conf,systemid}
```

5. マシンIDを含むファイルを削除します。

- SLES 12:

```
rm /etc/machine-id
rm /var/lib/dbus/machine-id
dbus-uuidgen --ensure
systemd-machine-id-setup
```

- SLES 11:

```
suse_register -E
```

6. 資格情報ファイルを削除します。

- SLESクライアント:

```
rm -f /etc/zypp/credentials.d/{SCCcredentials,NCCcredentials}
```

- Red Hat Enterprise Linuxクライアント:

```
rm -f /etc/NCCcredentials
```

7. ブートストラップスクリプトを再実行します。 複製元のシステムを上書きせずにUyuniに複製されたシステムが表示されるはずです。

25.26. Web UIからの登録が失敗し、エラーが表示されない場合のトラブルシューティング

Web UIからの初期登録では、すべてのSaltクライアントがSalt SSHを使用しています。

その性質上、Salt SSHクライアントはサーバにエラーを報告しません。

ただし、Salt SSHクライアントは、エラーを検査できるログを `/var/log/salt-ssh.log` にローカルに保存します。

25.27. 従来のクライアントを削除した後、Salt minionとして登録する場合のトラブルシューティング

これは有効なシナリオではありません。通常、クライアントを削除せずに、従来のクライアントをSalt minionに移行します。 Saltは、従来のクライアントがあることを自動的に検出し、必要な変更を自動的に行います。ただし、何らかの方法で従来のクライアントを削除し、それをSalt minionとして再度登録する場合は、Salt minionとして登録する前に、クライアントで次のステップを実行する必要があります。

1. 次のファイルを削除します。

```
/etc/sysconfig/rhn/systemid
```

2. 次のパッケージを削除します。

```
zypp-plugin-spacewalk
```

25.28. 従来のRed Hatクライアントの登録のトラブルシューティング

従来のRed Hat Enterprise Linux 7およびSLES Expanded Support 7クライアントでは、Uyuniで正常に動作するためにいくつかの署名されていないパッケージが必要なため、これらのクライアントタイプのカスタムチャンネルでは通常、**gpgcheck** フラグを設定解除します。ただし、**rhnplugin.conf** ファイルはこの設定を上書きし、GPGチェックを有効にします。

これは、従来のRed Hat Enterprise Linux 7およびSLES Expanded Support 7クライアントが登録されている場合、カスタムチャンネルで**gpgcheck** が無効になっている場合でも、クライアントはカスタムチャンネルから署名されていないパッケージをインストールしないことを意味します。

この問題を解決するには、**/etc/yum/pluginconf.d/rhnplugin.conf** ファイルを編集し、GPGチェックを無効にして、署名されていないパッケージのインストールを有効にし、予期されるようにクライアントが動作できるようにします。

25.29. Red Hat CDNチャンネルと複数の証明書のトラブルシューティング

Red Hatコンテンツデリバリネットワーク(CDN)チャンネルは複数の証明書を提供することができますが、Uyuni Web UIは単一の証明書しかインポートできません。 CDNがUyuni Web UIで認識されている証明書とは異なる証明書を提示した場合、証明書が正確であっても検証が失敗し、リポジトリへのアクセスパミッションが拒否されます。 次のようなエラーメッセージが生じます。

```
[error] ([エラー])
Repository '<repo_name>' is invalid. (リポジトリ'<repo_name>'は無効です。)
<repo.pem> Valid metadata not found at specified
URL (有効なメタデータが指定されているURLで見つかりませんでした)
History: (履歴:)
- [] Error trying to read from '<repo.pem>' (
'<repo.pem>'からの読み込み時にエラーが発生しました)
- Permission to access '<repo.pem>' denied. (
'<repo.pem>'へのアクセスが拒否されました。)
Please check if the URIs defined for this repository are pointing to a
valid repository. (このリポジトリ用に定義されている
URIが有効なリポジトリを指しているかどうかを確認してください。
Skipping repository '<repo_name>' because of the above
error. (上記エラーのため、リポジトリ'<repo_name>'をスキップします。)
Could not refresh the repositories because of
errors. (エラーが発生したため、リポジトリを更新できませんでした。)
HH:MM:SS RepoMDError: Cannot access repository. Maybe repository GPG keys
are not imported (HH:MM:SS RepoMDError:
リポジトリにアクセスできません。リポジトリGPGキーがインポートされていない可能性があります)
```

この問題を解決するには、すべての有効な証明書を1つの `.pem` ファイルにマージし、Uyuniで使用する証明書を作成します。

手順: 複数のRed Hat CDN証明書の解決

1. Red Hat クライアントのコマンドプロンプトで、rootとして、`/etc/pki/entitlement/` からすべての現行の証明書を単一の `rh-cert.pem` ファイルに収集します。

```
cat 866705146090697087.pem 3539668047766796506.pem redhat-entitlement-
authority.pem > rh-cert.pem
```

2. `/etc/pki/entitlement/` からすべての現行のキーを単一の `rh-key.pem` ファイルに収集します。

```
cat 866705146090697087-key.pem 3539668047766796506-key.pem > rh-
key.pem
```

次に、Client-configuration > Clients-rh-cdnの手順に従って、新しい証明書をUyuniサーバにインポートできます。

25.30. Uyuniサーバの名前変更のトラブルシューティング

Uyuniサーバのホスト名をローカルで変更する場合は、Uyuniインストールが適切に機能しなくなります。これは、変更がデータベースで行われていないため、変更がクライアントとプロキシに伝播されないためです。

Uyuniサーバのホスト名を変更する必要がある場合は、**spacewalk-hostname-rename** スクリプトを使用して変更できます。このスクリプトはPostgreSQLデータベースの設定とUyuniの内部構造を更新します。

spacewalk-hostname-rename スクリプトは **spacewalk-utils** パッケージの一部です。

このスクリプトの唯一の必須パラメータはUyuniサーバの新たに設定されたIPアドレスです。

プロシージャ: Uyuniサーバの名前変更

- システムレベルのサーバのネットワーク設定を、DNSサーバでローカルおよびリモートで変更します。逆引き名前解決のための設定を指定する必要があります。ネットワーク設定の変更は、他のシステムの名前変更と同じ方法で実行されます。
- Uyuniサーバを再起動して、新しいネットワーク構成を使用し、ホスト名が変更されていることを確認します。
- サーバのパブリックIPアドレスを使用して、**spacewalk-hostname-rename** スクリプトを実行します。サーバが新しいホスト名を使用していない場合、スクリプトは失敗します。このスクリプトがすべてのSalt minionのpillarデータを更新することに注意してください。実行にかかる時間は登録済みSaltシステムの数によって異なります。
- クライアントを再設定して、環境に新しいホスト名とIPアドレスを認識させるようにします。Salt minion設定ファイルの `/etc/salt/minion` で、新しいSaltマスター(Uyuniサーバ)の名前を指定する必要があります。

```
master: <new_hostname>
```

- これが変更されたら、**salt-minion** プロセスを再起動します。

```
systemctl restart salt-minion
```

- ホスト名がminion設定に完全に伝播されるようにするには、highstateを適用します。highstateを適用すると、リポジトリURLのホスト名が更新されます。

従来のクライアントには、変更される必要がある `/etc/sysconfig/rhn/up2date` 設定ファイルがあります。再アクティベーションキーを使用すると、従来のクライアント(存在する場合)を再登録できます。 詳細については、[Client-configuration > Registration-cli](#)を参照してください。



プロキシからPXEブートを使用する場合は、プロキシの設定を確認する必要があります。プロキシで、`configure-tftpsync.sh` セットアップスクリプトを実行して、要求される情報を入力します。 詳細については、[Installation-and-upgrade > Proxy-setup](#)を参照してください。

25.31. xref:troubleshooting/tshoot-retrying-setup-target-system.adoc[ターゲットシステムの設定を再試行する場合のトラブルシューティング]

25.32. RPC接続タイムアウトのトラブルシューティング

RPC接続は、ネットワークが低速になったり、ネットワークリンクがダウンしたりするためにタイムアウトすることがあります。その結果、パッケージのダウンロードやバッチジョブがハングしたり予想よりも時間がかかります。設定ファイルを編集することで、RPC接続にかかる最大時間を調整できます。ただし、これではネットワークの問題は解決されず、プロセスがハングするのではなく失敗します。

プロシージャ: RPC接続タイムアウトの解決

- Uyuniサーバで、`/etc/rhn/rhn.conf` ファイルを開いて、最大タイムアウト値(秒単位)を設定します。

```
server.timeout = `number`
```

- Uyuniプロキシで、`/etc/rhn/rhn.conf` ファイルを開いて、最大タイムアウト値(秒単位)を設定します。

```
proxy.timeout = `number`
```

- zypperを使用するSUSE Linux Enterprise Serverクライアントで、`/etc/zypp/zypp.conf` ファイルを開いて、最大タイムアウト値(秒単位)を設定します。

```
## Valid values: [0,3600]
## Default value: 180
download.transfer_timeout = 180
```

- yumを使用するRed Hat Enterprise Linuxクライアントで、`/etc/yum.conf` ファイルを開いて、最大タイムアウト値(秒単位)を設定します。

```
timeout = `number`
```



RPCタイムアウトを`180`秒未満に制限する場合は、完全に正常な操作が中断されるリスクがあります。

25.33. ダウンと表示されるSaltクライアントとDNS設定のトラブルシューティング

Saltクライアントが実行されている場合でも、パッケージの更新や状態の適用などのアクションは、次のメッセージで失敗としてマークされる可能性があります。

Minionがダウンしているか、接続できませんでした。

この場合、アクションのスケジュールを変更してみてください。スケジュールの変更が成功した場合、問題の原因はDNS設定の誤りである可能性があります。

Saltクライアントが再起動されたとき、またはグレインが更新された場合、クライアントはFQDNグレインを計算し、グレインが続行されるまで応答しません。Uyuniサーバでスケジュールされたアクションが実行される場合、Uyuniサーバは、実際のアクションの前にクライアントに対して `test.ping` を実行して、クライアントが実際に実行され、アクションをトリガできることを確認します。

デフォルトでは、Uyuniサーバは `test.ping` コマンドからの応答を取得するために5秒間待機します。5秒以内に応答が受信されない場合、アクションは失敗に設定され、クライアントがダウンしているか、接続できなかったというメッセージが表示されます。

これを修正するには、クライアントのDNS解決を修正して、クライアントがFQDNの解決中に5秒間スタックしないようにします。

これができない場合は、Uyuniサーバ上の `/etc/rhn/rhn.conf` ファイルの `java.salt_presence_ping_timeout` の値を4より大きい値に増やしてみてください。

例:

```
java.salt_presence_ping_timeout = 6
```

その後、次のコマンドを使用して `spacewalk-services` を再起動します。

```
spacewalk-services restart
```



この値を大きくすると、minionに到達できないのかminionが応答しないのかをUyuniサーバが確認するのに時間がかかり、Uyuniサーバの全体的な速度が低下したり応答しなくなったりします。

25.34. Saltboot Formulaのトラブルシューティング

計算されたパーティションサイズの値に問題があるため、saltboot formulaをSLE 11 SP3クライアントで作成すると、次のようなエラーで失敗することがあります。

```

ID: disk1_partitioned
Function: saltboot.partitioned
    Name: disk1
    Result: false
Comment: An exception occurred in this state: Traceback (most recent
call last):
  File "/usr/lib/python2.6/site-packages/salt/state.py", line 1767, in
call
      **cdata['kwargs'])
  File "/usr/lib/python2.6/site-packages/salt/loader.py", line 1705, in
wrapper
      return f(*args, **kwargs)
  File "/var/cache/salt/minion/extmods/states/saltboot.py", line 393, in
disk_partitioned
      existing = __salt__['partition.list'](device, unit='MiB')
  File "/usr/lib/python2.6/site-packages/salt/modules/parted.py", line
177, in list_
      'Problem encountered while parsing output from parted')
CommandExecutionError: Problem encountered while parsing output from
parted

```

この問題は、オペレーティングシステムを含むパーティションのサイズを手動で設定することにより解決できます。 サイズが正しく設定されている場合、式の作成は予期されるように機能します。

プロシージャ: Saltboot Formulaでパーティションサイズを手動で設定する

1. Uyuni Web UIで、**システム** > **システムグループ**に移動し、エラーの原因となっているSLE 11 SP3クライアントを含む **[Hardware Type Group]** (ハードウェアタイプグループ) を選択します。 **[Formulas]** タブで、 **[Saltboot]** サブタブに移動します。
2. オペレーティングシステムを含むパーティションを見つけ、 **[Partition Size]** (パーティションサイズ) フィールドに、適切なサイズ(MiB単位)を入力します。
3. **[Save Formula]** (Formulaの保存) をクリックし、highstateを適用して、変更を保存します。

25.35. スキーマのアップグレードが失敗する場合のトラブルシューティング

スキーマのアップグレードに失敗すると、データベースのバージョン確認およびその他すべてのspacewalkサービスが開始されません。 詳細および続行する方法のヒントについては **spacewalk-service start** を実行してください。

バージョン確認を直接実行することもできます。

```
systemctl status uyuni-check-database.service
```

または

```
journalctl -u uyuni-check-database.service
```

一般的な **spacewalk-service** コマンドを実行しない場合、これらのコマンドを実行するとデバッグ情報が出力されます。

25.36. 同期のトラブルシューティング

同期はさまざまな理由で失敗する可能性があります。接続問題に関する詳細情報を取得するには、次のコマンドを実行します。

```
export URLGRABBER_DEBUG=DEBUG
spacewalk-repo-sync -c <channelname> <options> > /var/log/spacewalk-repo-
sync-$(date +%F-%R).log 2>&1
```

/var/log/zypper.log にあるZypperによって作成されたログを確認することもできます。

GPGキーの不一致

UyuniはサードパーティのGPGキーを自動的に信頼しません。パッケージ同期が失敗する場合は、信頼されていないGPGキーが原因である可能性があります。この場合は、**/var/log/rhn/reposync**を開いて、次のようなエラーを検索することで確認できます。

```
[ '/usr/bin/spacewalk-repo-sync', '--channel', 'sle-12-sp1-ga-desktop-
nvidia-driver-x86_64', '--type', 'yum', '--non-interactive']
RepoMDError: Cannot access repository. Maybe repository GPG keys are
not imported
```

この問題を解決するには、UyuniにGPGキーをインポートする必要があります。GPGキーのインポートの詳細については、Administration > **Repo-metadata**を参照してください。

spacewalk-repo-sync からのGPGキーの削除

リポジトリのGPGキーが **spacewalk-repo-sync** を使用して手動でインポートされ、このキーが不要になった場合(たとえば、キーが侵害された場合や、テスト目的専用で使用された場合)、次のコマンドを使用して、**spacewalk-repo-sync** によって使用されるzypper RPMデータベースから削除できます。

```
rpm --dbpath=/var/lib/spacewalk/reposync/root/var/lib/rpm/ -e gpg-
pubkey-*
```

ここで、**gpg-pubkey-*** は削除されるGPGキーの名前です。

GPGキーの更新

GPGキーを更新する場合は、まず古いキーを削除してから、新しいキーを生成してインポートします。

チェックサムの不一致

チェックサムが失敗すると、`/var/log/rhn/reposync/*.log` ログファイルに次のようなエラーが表示される場合があります。

```
Repo Sync Errors: (50, u'checksums did not match
326a904c2fdb7a0e20033c87fc84ebba6b24d937 vs
afd8c60d7908b2b0e2d95ad0b333920aea9892eb', 'Invalid information
uploaded
to the server')
The package microcode_ctl-1.17-102.57.62.1.x86_64 which is referenced
by
patch microcode_ctl-8413 was not found in the database. This patch has
been skipped.
```

-Y オプションを使用して、コマンドラインプロンプトから同期を実行することで、このエラーを解決できます。

```
spacewalk-repo-sync --channel <channelname> -Y
```

このオプションはローカルにキャッシュされたチェックサムに依存するのではなく、同期前にリポジトリデータを検証します。

接続タイムアウト

ダウンロードが次のエラーでタイムアウトする場合:

```
28, 'Operation too slow. Less than 1000 bytes/sec transferred the last
300 seconds
```

このエラーは、`/etc/rhn/rhn.conf` で `reposync_timeout` と `reposync_minrate` の設定値を指定することで解決できます。デフォルトでは、300秒で1秒あたり1000バイト未満の転送が行われると、ダウンロードが中止されます。`reposync_minrate` で1秒あたりのバイト数を調整でき、`reposync_timeout` で待機する秒数を調整できます。

25.37. Taskomaticのトラブルシューティング

リポジトリメタデータの再生成は比較的集中的なプロセスであるため、Taskomaticが完了するまでに数分かかる可能性があります。また、Taskomaticがクラッシュした場合、リポジトリメタデータの再生成が中断される可能性があります。

Taskomatic がまだ実行されている場合、またはプロセスがクラッシュした場合は、Web UIでパッケージ更新が使用可能に見える場合がありますが、クライアントに表示されず、クライアントを更新しようとすると失敗します。この場合、`zypper ref` コマンドを使用すると、次のようなエラーが表示されます。

```
Valid metadata not found at specified URL
```

これを修正するには、Taskomaticがまだリポジトリメタデータを生成中であるか、またはクラッシュが発生した可能性があるかどうかを判断します。 クライアントの更新が正しく実行されるように、メタデータの再生成が完了するまで待つか、クラッシュ後にTaskomaticを再起動します。

プロシージャ: Taskomaticの問題の解決

1. Uyuniサーバで、`/var/log/rhn/rhn_taskomatic_daemon.log` ファイルを確認して、メタデータ再生成プロセスがまだ実行されているか、クラッシュが発生したかどうかを判断します。
2. taskomaticを再起動します。

```
service taskomatic restart
```

3. Taskomaticログファイルで、次のような開始および終了行を検索することで、メタデータ再生成に関するセクションを特定できます。

```
<YYYY-DD-MM> <HH:MM:SS>,174 [Thread-584] INFO
com.redhat.rhn.taskomatic.task.repolmd.RepositoryWriter - Generating
new repository metadata for channel 'cloned-2018-q1-sles12-sp3-
updates-x86_64'(sha256) 550 packages, 140 errata
```

```
...
```

```
<YYYY-DD-MM> <HH:MM:SS>,704 [Thread-584] INFO
com.redhat.rhn.taskomatic.task.repolmd.RepositoryWriter - Repository
metadata generation for 'cloned-2018-q1-sles12-sp3-updates-x86_64'
finished in 4 seconds
```

25.38. Web UIの読み込みが失敗する場合のトラブルシューティング

移行後、Web UIが読み込まれない場合があります。 新しいシステムのホスト名およびIPアドレスが古いシステムと同じ場合、通常、このエラーはブラウザのキャッシュが原因です。 この重複によって一部のブラウザが混乱する可能性があります。

この問題は、キャッシュをクリアしてページを再読み込みすると解決します。 ほとんどのブラウザでは、この操作は、`Ctrl+Shift+F5` を押すことで実行できます。

Chapter 26. 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{ldquo}GNU Free Documentation License{rdquo}.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.