



Red Hat Update Infrastructure 4

Red Hat Update Infrastructure の設定および管理

Red Hat Update Infrastructure 4 の設定および修正

Red Hat Update Infrastructure 4 Red Hat Update Infrastructure の設定および管理

Red Hat Update Infrastructure 4 の設定および修正

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat Update Infrastructure 4 (RHUI 4) を設定して管理する方法を説明します。

目次

多様性を受け入れるオープンソースの強化	4
第1章 RED HAT UPDATE INFRASTRUCTURE 4 の概要	5
1.1. インストールオプション	5
1.2. RHUI 4 コンポーネント	8
1.3. コンテンツプロバイダーのタイプ	12
1.4. コンポーネントの通信	12
1.5. 管理者パスワードの変更	13
1.6. 関連情報	14
第2章 リポジトリの管理	15
2.1. 利用可能なチャンネル	15
2.2. RHUI 4 で現在管理しているリポジトリのリスト表示	15
2.3. リポジトリの詳細情報の表示	16
2.4. リポジトリステータスファイルの生成	17
2.5. 新しい RED HAT コンテンツリポジトリの追加	19
2.6. 入力ファイルを使用した新しい RED HAT コンテンツリポジトリの追加	21
2.7. カスタムリポジトリの新規作成 (RPM コンテンツのみ)	22
2.8. RHUI 4 からのリポジトリ削除	24
2.9. カスタムリポジトリへのコンテンツのアップロード (RPM コンテンツのみ)	24
2.10. 遠隔地の WEB サイトからのコンテンツのアップロード (RPM コンテンツのみ)	25
2.11. パッケージグループメタデータのカスタムリポジトリへのインポート	26
2.12. カスタムリポジトリからのコンテンツの削除 (カスタム RPM コンテンツのみ)	27
2.13. リポジトリ内のパッケージのリスト表示 (RPM コンテンツのみ)	28
第3章 エンタイトルメント証明書とクライアント設定 RPM の作成	30
3.1. RED HAT UPDATE INFRASTRUCTURE MANAGEMENT TOOL でのクライアントエンタイトルメント証明書の作成	30
3.2. CLI でのクライアントエンタイトルメント証明書の作成	31
3.3. クライアントエンタイトルメント証明書が FUTURE 暗号化ポリシーに準拠しているかどうかの確認	32
3.4. CLI を使用したクライアント設定 RPM のリポジトリ ID 接頭辞の変更	32
3.5. RED HAT UPDATE INFRASTRUCTURE MANAGEMENT TOOL でのクライアント設定 RPM の作成	33
3.6. CLI でのクライアント設定 RPM の作成	33
第4章 RED HAT エンタイトルメント証明書の管理	35
4.1. RED HAT UPDATE APPLIANCE 証明書	35
4.2. コンテンツ配信サーバー証明書	35
4.3. クライアント証明書	35
第5章 SYNCHRONIZATION の状況確認とスケジューリング	38
5.1. リポジトリ同期の概要表示	38
5.2. 実行同期の表示	38
5.3. 前回のリポジトリ同期の詳細を表示	39
5.4. 個別リポジトリの即時同期	39
5.5. アクティブな同期タスクのキャンセル	40
5.6. 待機中の同期タスクのキャンセル	41
5.7. リポジトリの自動公開ステータスの表示と変更	42
5.8. リポジトリのワークフローの表示と進行	43
5.9. リポジトリのファイルシステムへのエクスポート	44
第6章 コンテンツデリバリーサーバーの管理	45
6.1. RHUI 4 で管理されているすべての既知の CDS インスタンスのリストアップ	45
6.2. CDS の新規登録について	45

6.3. CDS への設定の再インストールおよび再適用	47
6.4. レガシー CA を受け入れる CDS の設定	48
6.5. レガシー CA の受け入れ停止の CDS 設定	48
6.6. CDS の登録解除	49
第7章 HAPROXY ロードバランサーインスタンスの管理	50
7.1. RHUI 4 で管理されているすべての HAPROXY ロードバランサーインスタンスのリストアップ	50
7.2. HAPROXY ロードバランサーの新規登録について	50
7.3. HAPROXY ロードバランサーの再インストールと設定の再適用について	52
7.4. HAPROXY ロードバランサーの登録解除	53
第8章 コンテナの管理	54
8.1. RED HAT UPDATE INFRASTRUCTURE のコンテナについて	54
8.2. RED HAT UPDATE INFRASTRUCTURE へのコンテナの追加	54
8.3. コンテナリポジトリの同期	56
8.4. コンテナクライアント設定の生成	56
8.5. クライアントへのコンテナ設定 RPM のインストール	57
8.6. クライアントでの PODMAN PULL コマンドのテスト	58
第9章 設定ファイル、終了コード、およびログファイル	59
第10章 RHUI 4 コマンドの使用	62
10.1. RHUI 4 メニューオプションの使用	62
10.2. RHUI 4 CLI オプションの使用	65
第11章 認定クラウドおよびサービスプロバイダー認定のワークフロー	71
11.1. 関連情報	71
第12章 RED HAT UPDATE INFRASTRUCTURE のバックアップと復元	72
12.1. RED HAT UPDATE APPLIANCE のバックアップ	72
12.2. RED HAT UPDATE APPLIANCE の復元	73
12.3. コンテンツ配信サーバーのバックアップ	74
12.4. コンテンツ配信サーバーの復元	75
12.5. HAPROXY サーバーのバックアップ	76
12.6. HAPROXY サーバーの復元	76
第13章 RHUI 4 の一般的な問題を解決	77

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 RED HAT UPDATE INFRASTRUCTURE 4 の概要

Red Hat Update Infrastructure 4 (Red Hat Update Infrastructure 4) は、スケーラビリティが高く、冗長なフレームワークで、リポジトリとコンテンツを管理できます。また、クラウドプロバイダーは、Red Hat Enterprise Linux (RHEL) インスタンスにコンテンツおよび更新を配信できます。アップストリームの Pulp プロジェクトに基づいて、RHUI を使用すると、クラウドプロバイダーは Red Hat がホストするリポジトリコンテンツをローカルにミラーリングし、独自のコンテンツでカスタムリポジトリを作成し、負荷分散されたコンテンツ配信システムを通じてエンドユーザーの大規模なグループがそれらのリポジトリを利用できるようになります。

システム管理者は、Red Hat Update Appliance (RHUA)、コンテンツ配信サーバー (CDS)、リポジトリ、共有ストレージ、および負荷分散をインストールして設定することで、[Red Hat Certified Cloud and Service Provider プログラム](#) に参加するためのインフラストラクチャーを準備できます。

RHUI の設定は、以下のタスクで設定されます。

- Red Hat リポジトリの作成および同期
- クライアントエンタイトルメント証明書およびクライアント設定 RPM の作成
- RHUI サーバーのクライアントプロファイルの作成

対象となるのは、経験豊富な RHEL システム管理者です。RHEL のスキルに限りがあるシステム管理者は、Red Hat 認定クラウドプロバイダーアーキテクチャーサービスを提供する場合に、Red Hat コンサルティングを利用すること検討してください。

RHUI の設定、管理、および更新について、以下のトピックでご紹介します。

- RHUI コンポーネント
- コンテンツプロバイダーのタイプ
- コンポーネントの管理に使用するコマンドラインインターフェイス (CLI)
- ユーティリティコマンド
- 証明書の管理
- コンテンツ管理

1.1. インストールオプション

以下の表は、さまざまな Red Hat Update Infrastructure 4 コンポーネントを示しています。

表1.1 Red Hat Update Infrastructure のコンポーネントおよび機能

コンポーネント	略語	機能	代替方法
Red Hat Update Appliance	RHUA	Red Hat コンテンツ配信ネットワークから新しいパッケージをダウンロードして、各 CDS ノードにコピーする	なし

コンポーネント	略語	機能	代替方法
コンテンツ配信サーバー	CDS	クライアントが更新パッケージ用に接続する yum リポジトリを提供する	なし
HAProxy	なし	CDS ノード全体での負荷分散を提供する	既存の負荷分散ソリューション
共有ストレージ	なし	共有ストレージを提供する	既存のストレージソリューション

以下の表は、インストールタスクを実行する方法を説明します。

表1.2 Red Hat Update Infrastructure のインストールタスク

インストールタスク	実行する場所
RHEL 8 のインストール	RHUA、CDS、および HAProxy
システムのサブスクリプション	RHUA、CDS、および HAProxy
RHUI サブスクリプションのアタッチ	RHUA、CDS、および HAProxy
更新の適用	RHUA、CDS、および HAProxy
rhui-installer のインストール	RHUA
rhui-installer の実行	RHUA

1.1.1. オプション 1:完全インストール

- RHUA
- 共有ストレージを持つ 2 つ以上の CDS ノード
- 1 つ以上の HAProxy ロードバランサー

1.1.2. オプション 2:既存のストレージソリューションを使用したインストール

- RHUA
- 既存のストレージソリューションを備えた 2 つ以上の CDS ノード
- 1 つ以上の HAProxy ロードバランサー

1.1.3. オプション 3:既存のロードバランサーソリューションを使用したインストール

- RHUA

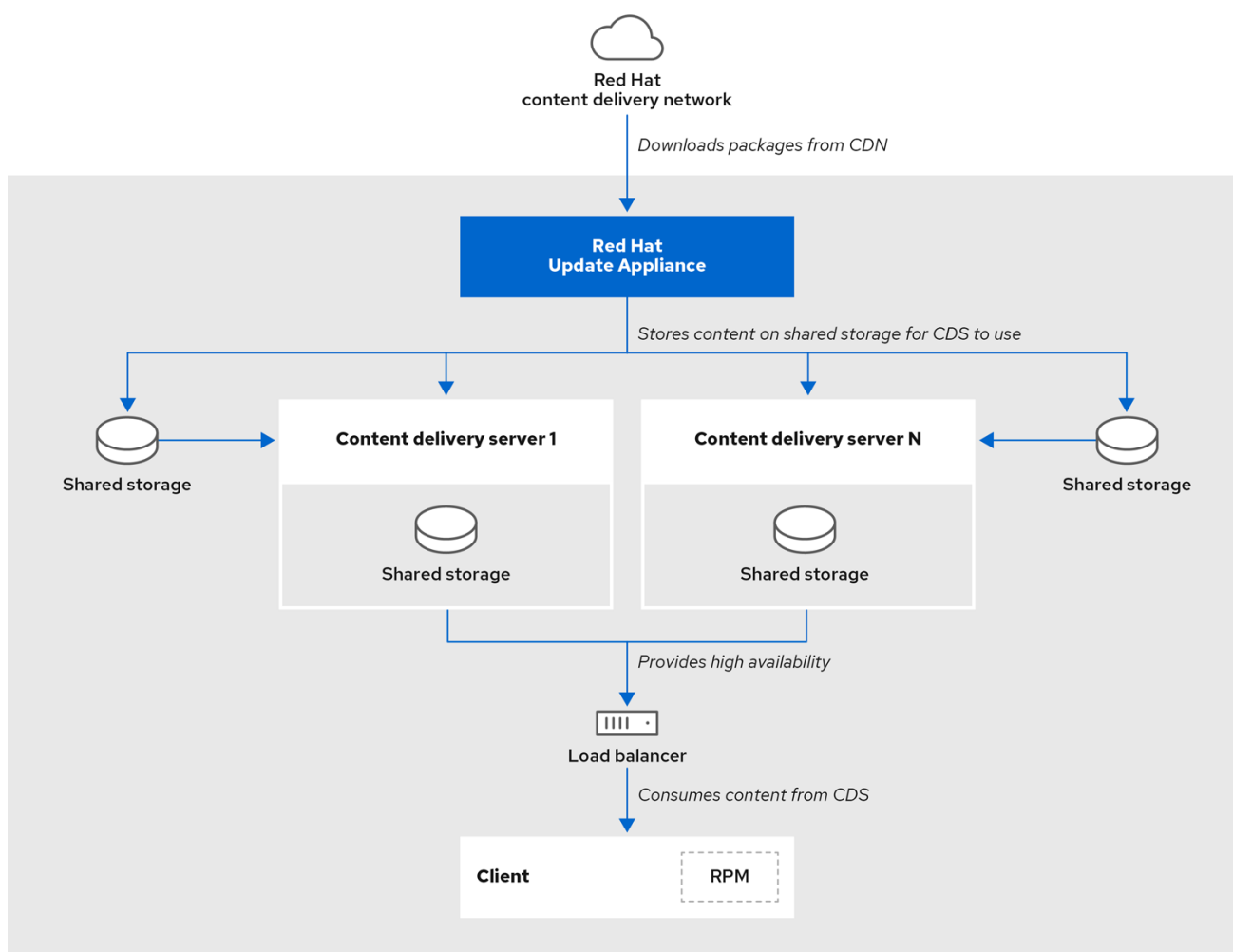
- 共有ストレージを持つ2つ以上の CDS ノード
- 既存のロードバランサー

1.1.4. オプション 4: 既存のストレージおよびロードバランサーのソリューションを使用したインストール

- RHUA
- 既存の共有ストレージを持つ2つ以上の CDS ノード
- 既存のロードバランサー

以下は、さまざまな Red Hat Update Infrastructure 4 コンポーネントがどのように対話するかについての俯瞰図です。

図1.1 Red Hat Update Infrastructure 4 の概要



172_RHUI.1121



注記

RHUA を **--type rhui** としてサブスクライブし、RHUI をインストールするには、Red Hat 認定クラウドおよびサービスプロバイダーサブスクリプションが必要です。適切なコンテンツ証明書も必要です。

RHUA ノードおよび CDS ノードを別の **x86_64** サーバー (ベアメタルまたは仮想マシン) にインストールします。RHUI に接続するサーバーおよびネットワークがすべて Red Hat Subscription Management サービスにアクセスできるようにします。

1.2. RHUI 4 コンポーネント

RHUI の各コンポーネントが他のコンポーネントとどのように相互作用するかを理解することで、システム管理者としての仕事が少し楽になります。

1.2.1. Red Hat Update Appliance

RHUI インストールごとに RHUA がありますが、多くのクラウド環境では、リージョンまたはデータセンターごとに RHUI インストールがあります。たとえば、Amazon の EC2 クラウドは複数のリージョンで設定されています。すべてのリージョンでは、独自の RHUA ノードを持つ個別の RHUI が設定されています。

RHUA では、以下の作業を行うことができます。

- Red Hat コンテンツ配信ネットワーク (CDN) から新規パッケージをダウンロードします。RHUA は Red Hat に接続する唯一の RHUI コンポーネントであり、RHUA の同期スケジュールを設定できます。
- 共有ネットワークストレージに新しいパッケージをコピーします。
- RHUI インストールの健全性を検証し、結果を RHUA にあるファイルに書き込みます。監視ソリューションは、このファイルを使用して RHUI インストールの健全性を判断します。
- CLI ツールを使用して RHUI インストールの健全性について人間が判読できるビューを提供します。

RHUI は主に `/etc/rhui/rhui-tools.conf` と `/etc/rhui/rhui-subscription-sync.conf` の設定ファイルを使用します。

`/etc/rhui/rhui-tools.conf` 設定ファイルには、RHUA で使用される一般的なオプション (証明書のデフォルトのファイルの場所、Red Hat CDN 同期のデフォルト設定パラメーターなど) が含まれます。通常、このファイルを編集する必要はありません。

Red Hat Update Infrastructure Management Tool は、ユーザー入力の値に基づいて `/etc/rhui/rhui-subscription-sync.conf` 設定ファイルを生成します。これには、特定のリージョンで RHUA の実行を促進するすべての情報が含まれます。設定例としては、RHUA 上にパッケージのダウンロード先があります。

RHUA は複数のサービスを使用して、配信を簡単にするためにコンテンツを同期し、整理し、配布します。

RHUA サービス

Pulp

サポート対象のサービスの管理を監視するサービスで、ユーザーが操作できるユーザーインターフェイスを提供します。

PostgreSQL

現在同期しているリポジトリ、パッケージ、およびその他の重要なメタデータを追跡するために使用される PostgreSQL データベースです。

1.2.2. コンテンツ配信サーバー

CDS ノードは、クライアントが更新したコンテンツ用に接続するリポジトリを提供します。CDS は 1 つしかありません。RHUI はフェイルオーバー機能を備えたロードバランサーを提供するため、複数の CDS ノードを使用することが推奨されます。

エンドユーザーの RHEL システムへの CDS ノードホストコンテンツシステムの数はありませんが、CDS はラウンドロビン形式の負荷分散方式 (A、B、C、A、B、C) で動作し、エンドユーザーシステムにコンテンツを提供します。CDS は HTTP を使用して、httpd ベースの **yum** リポジトリを介してコンテンツをエンドユーザーシステムにホストします。

設定時に、パッケージが同期される CDS ディレクトリーを指定します。RHUA と同様に、唯一の要件は、CDS にディレクトリーをマウントすることです。必要なデバイスを割り当てる際の最善のアクションは、クラウドプロバイダーによって決定されます。Red Hat Update Infrastructure Management Tool 設定 RPM は、パッケージディレクトリーを NGINX 設定にリンクして提供します。

現在、RHUI は以下の共有ストレージソリューションをサポートします。

NFS

NFS が使用される場合、**rhui-installer** は RHUA で NFS 共有を設定し、コンテンツと NFS 共有をマウントする CDS ノードのディレクトリーを保存できます。以下の **rhui-installer** オプションは、このような設定を制御します。

- **--remote-fs-mountpoint** は、リモートファイルシステム共有をマウントする必要があるファイルシステムの場所です (デフォルト: **/var/lib/rhui/remote_share**)
- **--remote-fs-server** は、共有ファイルシステムが使用するリモートマウントポイントです。たとえば、**nfs.example.com:/path/to/share** (デフォルト: **nfs.example.com:/export**) です。

CephFS

CephFS を使用する場合は、CephFS を個別に設定し、RHUI をマウントポイントとして使用する必要があります。以下の **rhui-installer** オプションは、このような設定を制御します。

- **--remote-fs-server** は、共有ファイルシステムが使用するリモートマウントポイントです。たとえば、**ceph.example.com:/path/to/share** (デフォルト: **ceph.example.com:/export**) です。



注記

本書には、Ceph 共有ストレージを設定して設定する手順は含まれません。Ceph 関連のタスクについては、システム管理者に問い合わせるか、Ceph ドキュメントを参照してください。

これらのデフォルト値が使用される場合、RHUA の **/export** ディレクトリーと各 CDS の **/var/lib/rhui/remote_share** ディレクトリーは同じになります。

想定される使い方は、RHUA とすべての CDS ノードで、NFS などの 1 つの共有ネットワークファイルシステムを使用することです。クラウドプロバイダーは、RHUA がパッケージを書き込んで各 CDS が読み取る共有ストレージの形式を使用できます。



注記

ストレージソリューションでは、RHUA ノードおよび CDS ノードにマウントするための NFS または CephFS エンドポイントを提供する必要があります。ローカルストレージが実装されている場合、クラスターが機能するために共有ストレージが必要になります。RHUA にローカルストレージを提供する場合は、**rhua.example.com:/path/to/nfs/share** エンドポイントが設定された NFS サーバーとして機能するように RHUA を設定します。

RHUI ノードには Ceph 共有ファイルシステムのストレージを設定しないでください。CephFS は、独立した専用マシンに設定する必要があります。

各 CDS で行われる唯一の標準以外のロジックは、エンタイトルメント証明書チェックです。このチェックにより、クライアントが **yum** リポジトリに対して要求を行うのが、クラウドプロバイダーがこれらのリポジトリにアクセスする権限を付与するようになります。このチェックでは、以下の条件を確認します。

- エンタイトルメント証明書がクラウドプロバイダーの認証局 (CA) 証明書により署名されている。また、この検証を容易にするため、CA 証明書はその設定の一部として CDS にインストールされている。
- 要求された URI が、クライアントのエンタイトルメント証明書にあるエンタイトルメントと一致する。

CA の検証に失敗すると、クライアントには SSL エラーが表示されます。詳細については、**/var/log/nginx/** の CDS ノードの NGINX ログを参照してください。

```
[root@cds01 ~]# ls -l /var/log/nginx/
access.log
error.log
unicorn-auth.log
unicorn-content_manager.log
unicorn-mirror.log
ssl-access.log----
```



重要

NGINX 設定は、CDS のインストール時に **/etc/nginx/conf.d/ssl.conf** ファイルによって処理されます。

複数のクライアントがリポジトリに対して更新に問題がある場合は、RHUI で問題があることを示す可能性があります。詳細は、[Yum generates 'Errno 14 HTTP Error 401: Authorization Required' while accessing RHUI CDS](#) を参照してください。

1.2.3. HAProxy ロードバランサーの追加

複数の CDS が使用される場合は、負荷分散ソリューションを準備して、クライアント HTTPS 要求をすべてのサーバーに分散する必要があります。RHUI には HAProxy が同梱されていますが、インストール時に使用する負荷分散ソリューション (クラウドプロバイダーからのソリューションなど) を選択するのはユーザー次第です。HAProxy が使用されている場合は、挿入するノードの数を決定する必要もあります。

クライアントは CDS に直接移動するように設定されていません。リポジトリファイルは、RHUI ロードバランサーである HAProxy を参照するように設定されます。HAProxy は特に高可用性環境に適した TCP/HTTP リバースプロキシです。HAProxy は以下の作業を行います。

- 静的に割り当てられた Cookie (クッキー) に応じて HTTP リクエストをルーティングします。
- HTTP Cookie を使用してサーバー永続性を維持しながら、複数のサーバーに負荷を分散します。
- メインサーバーが失敗した場合にバックアップサーバーに切り替えます。
- サービス監視専用の特別なポートへの接続を許可します。
- 既存の接続を破損することなく接続の受け入れを停止します。
- 両方向で HTTP ヘッダーを追加、変更、および削除します。
- 特定のパターンに一致するリクエストをブロックします。
- アプリケーション Cookie に応じて、正しいアプリケーションサーバーへのクライアント接続を永続化します。
- HTML ページとして、アプリケーションから傍受された URI からの認証ユーザーに詳細なステータスを報告します。



注記

既存のロードバランサーを使用している場合は、ポート 443 が **cds-lb-hostname** のロードバランサーに設定され、クラスター内のすべての CDS がロードバランサーのプールにあることを確認します。

正確な設定は、使用する特定のロードバランサーソフトウェアによって異なります。ロードバランサーの設定方法を理解するには、一般的な HAProxy 設定から取られる以下の設定を参照してください。

```
[root@rhui4proxy ~]# cat /etc/haproxy/haproxy.cfg
# This file managed by Puppet
global
  chroot /var/lib/haproxy
  daemon
  group haproxy
  log 10.10.153.149 local0
  maxconn 4000
  pidfile /run/haproxy.pid
  stats socket /var/lib/haproxy/stats
  user haproxy

defaults
  log global
  maxconn 8000
  option redispatch
  retries 3
  stats enable
  timeout http-request 10s
  timeout queue 1m
  timeout connect 10s
  timeout client 1m
  timeout server 1m
  timeout check 10s

listen https00
```

```
bind 10.10.153.149:443
balance roundrobin
option tcplog
option tcp-check
server cds01.example.com cds01.example.com:443 check
server cds02.example.com cds02.example.com:443 check
```

クライアントが接続に失敗した場合、`/var/log/nginx/`の下にある CDS 上の **nginx** ログを確認し、CDS への要求の到達を確認することが重要です。リクエストが CDS に届かない場合は、DNS や一般的なネットワーク接続に問題がある可能性があります。

1.2.4. リポジトリおよびコンテンツ

リポジトリは、ソフトウェアパッケージ (RPM) のストレージの場所です。RHEL は、**yum** コマンドを使用して、RPM の検索、ダウンロード、インストール、および設定を行います。RPM には、アプリケーションの実行に必要なすべての依存関係が含まれます。RPM は、リポジトリのソフトウェアの更新もダウンロードします。

RHEL は、リソース管理に制御グループ (cgroups) などのコアテクノロジーを使用します。プロセスの分離には名前空間を使用します。セキュリティには SELinux を使用します。これにより、セキュアなマルチテナンシーを有効にし、セキュリティエクスプロイトの可能性を低減します。この技術を使用すると、セキュリティを向上させながら、アプリケーションの迅速なデプロイメント、簡単なテスト、メンテナンス、およびトラブルシューティングが可能になります。

RHUI に関連するコンテンツは、RHUA ノードおよび CDS ノードで使用する Red Hat CDN からダウンロードしたソフトウェア (RPM など) です。RPM は、特定のアプリケーションおよびツールの実行に必要なファイルを提供します。クライアントには、rpm パッケージが提供する SSL コンテンツ証明書と鍵のセットによるアクセスが付与されます。これは、生成された **yum** リポジトリファイルのセットも提供します。

関連情報

- [What Channels Can Be Delivered at Red Hat's Certified Cloud & Service Provider \(CCSP\) Partners?](#)

1.3. コンテンツプロバイダーのタイプ

クラウドコンピューティング環境には、3つのタイプがあります。

- パブリッククラウド
- プライベートクラウド
- ハイブリッドクラウド

本ガイドでは、パブリッククラウドおよびプライベートクラウドに焦点を当てています。パブリッククラウド、プライベートクラウド、ハイブリッドクラウドの使用の影響を理解していることを前提としています。

1.4. コンポーネントの通信

すべての RHUI コンポーネントは、ポート 443 での HTTPS 通信プロトコルを使用します。

表1.3 Red Hat Update Infrastructure の通信プロトコル

ソース	宛先	プロトコル	目的
Red Hat Update Appliance	Red Hat コンテンツ配信ネットワーク	HTTPS	Red Hat からパッケージをダウンロードする。
ロードバランサー	コンテンツ配信サーバー	HTTPS	クライアントの yum を転送する。
クライアント	ロードバランサー	HTTPS	クライアントの yum で CDS からパッケージをダウンロードする時に使用する。
コンテンツ配信サーバー	Red Hat Update Appliance	HTTPS	コンテンツに関する情報を Pulp API に要求する場合がある。

RHUI ノードでは、相互に通信するために以下のネットワークアクセスが必要です。



注記

ネットワークポートを開放して、使用する予定のノードだけにネットワークアクセスが制限されていることを確認してください。

表1.4 Red Hat Update Infrastructure ネットワークアクセス

ノード	ポート	アクセス
RHUA	443	RHUA、CDS01、CDS02、... CDSn
HAProxy	443	クライアントの仮想マシン

1.5. 管理者パスワードの変更

rhui-installer は、RHUI の初期ログインパスワードを設定します。また、**/etc/rhui/rhui-subscription-sync.conf** ファイルでも記述されます。**--rhui-manager-password** オプションを使用して初期パスワードを上書きできます。

初期パスワードを後で変更する場合は、**rhui-manager** ツールまたは **rhui-installer** から変更できます。**rhui-installer --help** コマンドを実行して **rhui-installer** オプションの完全なリストを表示します。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **u** を押して、**manage RHUI users** を選択します。
3. **User Manager** 画面で、**p** を押して **change admin's password (followed by logout)** を選択します。

```
-= User Manager =-
```

```
p change admin's password (followed by logout)
```

```
rhui (users) => p
```

```
Warning: After password change you will be logged out.
```

```
Use ctrl-c to cancel password change.
```

```
New Password:
```

4. 新しいパスワードを入力し、再度入力すると変更が確定します。

```
New Password:
```

```
Re-enter Password:
```

```
[localhost] env PULP_SETTINGS=/etc/pulp/settings.py /usr/bin/pulpcore-manager reset-admin-password -p *****
```

検証

1. 管理者パスワードを変更すると、次のようなメッセージが表示されます。

```
Password successfully updated. For security reasons you have been logged out.
```

```
[root@ip-10-141-150-145 ~]#
```

1.6. 関連情報

- [Red Hat Cloud Access リファレンスガイド](#)
- [Red Hat Enterprise Linux 8](#)
- [ストレージデバイスの管理](#)
- [HAProxy](#)
- [Pulp プロジェクト](#)

第2章 リポジトリの管理

2.1. 利用可能なチャンネル

CCSP (Certified Cloud and Service Provider) パートナーは、サービス経由で配信されるチャンネルとパッケージを制御します。さまざまな OS バージョンで利用可能なチャンネルに関する最新の情報は、[Red Hat Enterprise Linux repositories that can be delivered through Red Hat's Certified Cloud and Service Provider \(CCSP\) partners via RHUI](#) を参照してください。

RHEL 8 で使用できるリポジトリは以下の通りです。

- Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (RPMs)
- Red Hat Enterprise Linux 8 for x86_64 - BaseOS from RHUI (RPMs)
- Red Hat Enterprise Linux 8 for x86_64 - High Availability (RPMs) from RHUI
- Red Hat Enterprise Linux 8 for x86_64 - Resilient Storage (RPMs) from RHUI
- Red Hat Enterprise Linux 8 for x86_64 - Supplementary (RPMs) from RHUI
- Red Hat CodeReady Linux Builder for RHEL 8 x86_64 (RPMs) from RHUI

必要なチャンネルがない場合は、CCSP にお問い合わせください。

関連情報

- [Red Hat エコシステムカタログ](#)

2.2. RHUI 4 で現在管理しているリポジトリのリスト表示

リポジトリは、Linux ディストリビューションのダウンロード可能なソフトウェアが含まれるサーバーノードです。**yum** を使用して、リポジトリから RHUA ノードおよび CDS ノードへの RPM の検索、インストール、制御を行います。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で **l** を押して、**list repositories currently managed by the RHUI** を選択します。

```
...
```

```
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.0)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.1)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.2)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.3)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.4)
```

```

Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.0)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.1)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.2)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.3)
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.4)
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8)
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.0)
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.1)
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.2)
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.3)
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.4)

```

...

2.3. リポジトリの詳細情報の表示

Repository Management 画面では、特定のリポジトリに関する情報を表示できます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で、**i** を押します。

```
Enter value (1-1631) to toggle selection, 'c' to confirm selections, or '?' for more commands:
```

4. リポジトリ名の横にある値を入力してリポジトリを選択します。製品の選択を確認する前に、一度に1つのリポジトリ選択を入力します。
5. **c** を押して確認します。

```

Name:          Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Debug RPMs) from
RHUI (8.4)
ID:            rhel-8-for-aarch64-appstream-debug-rhui-rpms-8.4
Type:          Red Hat
Version:       0
Relative Path: content/dist/rhel8/rhui/8.4/aarch64/appstream/debug
GPG Check:     Yes
Custom GPG Keys: (None)
Red Hat GPG Key: Yes
Content Unit Count:
Last Sync:     2021-11-15 15:56:06
Next Sync:     2021-11-15 22:00:00

Name:          Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI
(8.4)
ID:            rhel-8-for-aarch64-appstream-rhui-rpms-8.4
Type:          Red Hat
Version:       0

```

```

Relative Path:  content/dist/rhel8/rhui/8.4/aarch64/appstream/os
GPG Check:     Yes
Custom GPG Keys:  (None)
Red Hat GPG Key:  Yes
Content Unit Count:
Last Sync:      2021-11-15 19:50:20
Next Sync:      2021-11-16 01:55:00

Name:          Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from
RHUI (8.4)
ID:            rhel-8-for-aarch64-appstream-source-rhui-rpms-8.4
Type:          Red Hat
Version:       0
Relative Path:  content/dist/rhel8/rhui/8.4/aarch64/appstream/source/SRPMS
GPG Check:     Yes
Custom GPG Keys:  (None)
Red Hat GPG Key:  Yes
Content Unit Count:
Last Sync:      2021-11-15 15:56:51
Next Sync:      2021-11-15 22:00:00

```

検証

1. 選択した内容についても同様の出力が表示されます。

2.4. リポジトリステータスファイルの生成

すべての RHUI リポジトリのステータスを表示し、いくつかの追加情報を提供する、機械判読可能な JSON ファイルを生成できます。これは、リポジトリのステータスを受動的に監視する場合などに便利です。

2.4.1. RHUI リポジトリのステータスファイルの生成

rhui-manager コマンドを使用すると、各リポジトリのステータスを機械判読可能な形式で取得できます。

手順

- RHUA ノードで、以下のコマンドを実行します。

```
rhui-manager --non-interactive status --repo_json <output file>
```

すべてのカスタムリポジトリおよび Red Hat リポジトリのディクショナリーのリストを含む JSON ファイルが生成されます。利用可能なディクショナリーのリストは、「[リポジトリステータス JSON ファイル内のディクショナリーキーのリスト](#)」を参照してください。

2.4.2. リポジトリステータス JSON ファイル内のディクショナリーキーのリスト

各 RHUI リポジトリのステータスを取得するコマンドを実行すると、機械判読可能な JSON ファイルが作成されます。JSON ファイルには、リポジトリごとに1つのディクショナリーを含むディクショナリーのリストが含まれています。

カスタムリポジトリのディクショナリーキーのリスト

表2.1 カスタムリポジトリーのディクショナリーキーのリスト

キー	説明
base_path	リポジトリーのパス。
description	リポジトリーの名前。
group	リポジトリーが属するグループ。これは常に文字列、 custom に設定されます。
id	リポジトリー ID。
name	リポジトリーの名前。リポジトリー ID と同じです。

Red Hat リポジトリーのディクショナリーキーのリスト

表2.2 Red Hat リポジトリーのディクショナリーキーのリスト

キー	説明
base_path	リポジトリーのパス。
description	リポジトリーの名前。
group	リポジトリーが属するグループ。これは常に文字列 redhat に設定されます。
id	リポジトリー ID。
last_sync_date	リポジトリーが最後に同期された日時。リポジトリーが一度も同期されていない場合、値は null になります。
last_sync_exception	リポジトリーの同期に失敗した場合に発生する例外。リポジトリーが正しく同期された場合、値は null になります。
last_sync_result	同期タスクの結果。 値は以下のようになります。 <ul style="list-style-type: none"> ● completed: リポジトリーが正しく同期されている場合。 ● null: リポジトリーが一度も同期されていない場合。 ● failed: 同期に失敗した場合。 ● running: 同期タスクが現在実行中の場合。

キー	説明
last_sync_traceback	リポジトリの同期に失敗した場合に記録されたトレースバック。リポジトリが正しく同期された場合、または一度も同期されなかった場合、値は null になります。
metadata_available	リポジトリでメタデータが利用可能かどうかを示すブール値。
name	リポジトリの名前。リポジトリ ID と同じです。
next_sync_date	次にスケジュールされているリポジトリの同期の日時。同期タスクが現在実行中の場合、値は running になります。
repo_published	このリポジトリが RHUI で公開されているかどうかを示すブール値。デフォルトでは、RHUI はリポジトリを自動的に公開するように設定されている点に注意してください。

2.5. 新しい RED HAT コンテンツリポジトリの追加

アップストリームの Pulp プロジェクトに基づいて、RHUI を使用すると、クラウドプロバイダーは Red Hat がホストするリポジトリコンテンツをローカルにミラーリングし、独自のコンテンツでカスタムリポジトリを作成し、負荷分散されたコンテンツ配信システムを通じてエンドユーザーがそれらのリポジトリを利用できるようになります。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で **a** を押して、**add a new Red Hat content repository** を選択します。
4. Red Hat Update Infrastructure Management Tool がエンタイトルメントのあるリポジトリを判断するのを待ちます。これには数分の時間がかかる場合があります。

```
rhui (repo) => a
```

```
Loading latest entitled products from Red Hat...
... listings loaded
Determining undeployed products...
... product list calculated
```

5. Red Hat Update Infrastructure Management Tool は、選択方法を求めるプロンプトを表示します。

■

```

Import Repositories:
  1 - All in Certificate
  2 - By Product
  3 - By Repository
Enter value (1-3) or 'b' to abort:

```

6. **2** を押して、**By Product** メソッドを選択します。
7. プロンプトでリポジトリの数を入力して、追加するリポジトリを選択します。また、リポジトリの範囲を選択することもできます (例: **1 ~ 5**)。

```

Enter value (1-620) to toggle selection, 'c' to confirm selections, or '?' for more commands:

```

8. 追加するリポジトリがすべて確認されるまで続行します。
9. リポジトリの選択が完了したら、**c** を押します。Red Hat Update Infrastructure Management Tool は、デプロイするリポジトリを表示し、確認を求めるプロンプトを表示します。

```

The following products will be deployed:
  Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI
  Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI
Proceed? (y/n)

```

10. **y** を押して続行します。メッセージは、それぞれのデプロイメントの成功を示します。

```

Importing Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.4)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.3)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.2)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.1)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8.0)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8)...

Importing Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI (8.4)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI (8.3)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI (8.2)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI (8.1)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI (8.0)...
  Importing product repository Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI (8)...

Content will not be downloaded to the newly imported repositories
until the next sync is run.

```


検証

1. **Repository Management** 画面で **I** を押し、正しいリポジトリがインストールされていることを確認します。

関連情報

- [Red Hat Enterprise Linux repositories that can be delivered through Red Hat's Certified Cloud and Service Provider \(CCSP\) partners via RHUI](#)

2.6. 入力ファイルを使用した新しい RED HAT コンテンツリポジトリの追加

Red Hat Update Infrastructure 4.2 以降では、設定された YAML 入力ファイルを使用してカスタムリポジトリを追加できます。YAML ファイルのテンプレート例は、RHUA ノードの `/usr/share/rhui-tools/examples/repo_add_by_file.yaml` ディレクトリーにあります。

この機能はコマンドラインインターフェイス (CLI) でのみ利用できます。

前提条件

- RHUA ノードへの root のアクセス権がある。

手順

1. RHUA ノードで、以下の形式で YAML 入力ファイルを作成します。

```
# cat /my_home/example.yaml
---
name: Example_YAML_File
repo_ids:
  - rhel-8-for-x86_64-baseos-eus-rhui-rpms-8.1
  - rhel-8-for-x86_64-baseos-eus-rhui-rpms-8.2
  - rhel-8-for-x86_64-baseos-eus-rhui-rpms-8.4
  - rhel-8-for-x86_64-baseos-eus-rhui-rpms-8.6
---
```

2. **rhui-manager** ユーティリティを使用して、入力ファイルに一覧表示されているリポジトリを追加します。

```
# rhui-manager repo add_by_file --file /my_home/example.yaml --sync_now
The name of the repos being added: Example_YAML_File
Loading latest entitled products from Red Hat...
... listings loaded
Successfully added Red Hat Enterprise Linux 8 for x86_64 - BaseOS - Extended Update
Support from RHUI (RPMs) (8.1) (Yum)
Successfully added Red Hat Enterprise Linux 8 for x86_64 - BaseOS - Extended Update
Support from RHUI (RPMs) (8.2) (Yum)
Successfully added Red Hat Enterprise Linux 8 for x86_64 - BaseOS - Extended Update
Support from RHUI (RPMs) (8.4) (Yum)
Successfully added Red Hat Enterprise Linux 8 for x86_64 - BaseOS - Extended Update
Support from RHUI (RPMs) (8.6) (Yum)
... successfully scheduled for the next available timeslot.
```

```
... successfully scheduled for the next available timeslot.
... successfully scheduled for the next available timeslot.
... successfully scheduled for the next available timeslot.
```

検証

- CLI で、以下のコマンドを使用してインストールされたリポジトリの一覧を表示し、正しいリポジトリがインストールされているかどうかを確認します。

```
# rhui-manager repo list
```

- RHUI 管理ツールのリポジトリ管理画面で、**l** を押して、インストールされているすべてのリポジトリを一覧表示し、正しいリポジトリがインストールされているかどうかを確認します。

2.7. カスタムリポジトリの新規作成 (RPM コンテンツのみ)

更新されたクライアント設定パッケージまたは他の Red Hat 以外のソフトウェアを RHUI クライアントに配信するために使用できるカスタムリポジトリを作成できます。64 ビット RHUI サーバー (例: **client-rhui-x86_64**) の保護されたリポジトリは、RHUI クライアントに新しい Red Hat 以外のパッケージ (更新済みのクライアント設定パッケージなど) を配布するための優先手段になります。

Red Hat コンテンツリポジトリと同様、保護されているすべてのカスタムリポジトリが、プロセッサアーキテクチャーでのみ異なる (**i386** と **AMD64**) 場合は、**\$basearch** yum 変数を使用して、エンタイトルメント証明書内の単一のエンタイトルメントに統合されます。

証明書に問題がある場合には、RHUI サーバーの保護されていないリポジトリを、更新された RPM を RHUI クライアントに配布するためのフォールバック方法として使用できます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で **c** を押して、**create a new custom repository (RPM content only)** を選択します。
4. リポジトリの一意の ID を入力します。英数字、_(アンダースコア)、および-(ハイフン)のみが許可されます。一意の ID でスペースを使用することはできません。たとえば、**repo1**、**repo_1**、および **repo-1** は有効なエントリになります。

```
Unique ID for the custom repository (alphanumerics, _, and - only):
```

5. リポジトリの表示名を入力します。この名前は、Red Hat Update Infrastructure Management Tool 内でリポジトリを識別するのに使用されます。

```
Display name for the custom repository [repo_1]:
```

6. リポジトリをホストするパスを指定します。このパスは、RHUI がホストする全リポジトリで一意でなければなりません。たとえば、このステップのパスを **some/unique/name** として指定すると、リポジトリは **//server/pulp/repos/some/unique/name** に配置されます。

Unique path at which the repository will be served [repo_1]:

7. リポジトリメタデータに使用されるチェックサムタイプとして **sha256** を選択します。



注記

RHEL 6、RHEL 7、または RHEL 8 のカスタムリポジトリを作成する場合は、**sha256** を使用します。RHEL 5 クライアントのリポジトリを作成する場合は **sha1** を使用します。

Algorithm to use when calculating the checksum values for repository metadata:

1 - sha256 (default)

2 - sha1 (RHEL 5)

Enter value (1-2) or 'b' to abort:

8. 新規リポジトリを保護するかどうかを選択します。この質問に **no** と回答すると、すべてのクライアントがリポジトリにアクセスできます。 **yes** と回答すると、適切なエンタイトルメント証明書を持つクライアントのみがリポジトリにアクセスできます。



警告

名前が示すように、保護されていないリポジトリのコンテンツは、クライアントのエンタイトルメント証明書を必要とせずに要求するシステムで利用できます。保護されていないリポジトリを使用してコンテンツを配信する場合は、更新されたクライアント設定 RPM などのコンテンツを配布する場合は注意してください。これにより、保護されたリポジトリへのアクセスが提供されます。

9. 以下のような質問に **yes** または **no** と答えます。

Should the repository require clients to perform a GPG check and verify packages are signed by a GPG key? (y/n)

Will the repository be used to host any Red Hat GPG signed content? (y/n)

Will the repository be used to host any custom GPG signed content? (y/n)

Enter the absolute path to the public key of the GPG key pair:

Would you like to enter another public key? (y/n)

Enter the absolute path to the public key of the GPG key pair:

Would you like to enter another public key? (y/n)

10. 新しいリポジトリの詳細が表示されます。プロンプトが表示されたら **y** を押して、情報を確認し、リポジトリを作成します。

検証

1. **Repository Management** 画面で **l** を押し、正しいリポジトリがインストールされていることを確認します。

関連情報

- [Red Hat Enterprise Linux repositories that can be delivered through Red Hat's Certified Cloud and Service Provider \(CCSP\) partners via RHUI](#)

2.8. RHUI 4 からのリポジトリ削除

Red Hat Update Infrastructure Management Tool が Red Hat リポジトリを削除すると、RHUA および適用可能なすべての CDS ノードからリポジトリが削除されます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で **d** を押して、Red Hat リポジトリを削除します。RHUI が現在管理しているリポジトリのリストが表示されます。
4. プロンプトでリポジトリの数を入力して、削除するリポジトリを選択します。リポジトリの数を入力すると、そのリポジトリ名の横にチェックマークが付けられます。また、リポジトリの範囲を選択することもできます (例: **1 ~ 5**)。
5. 削除するリポジトリがすべて確認されるまで続行します。
6. プロンプトで **c** を押して確認します。



注記

リポジトリの削除後、削除されたリポジトリを参照するクライアント設定 RPM は **yum** では使用できません。

2.9. カスタムリポジトリへのコンテンツのアップロード (RPM コンテンツのみ)

複数のパッケージを同時にアップロードし、一度に複数のリポジトリにアップロードできます。パッケージは RHUA に即時にアップロードされますが、CDS ノードが次回同期するまで CDS ノードでは使用できません。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で、**u** を押します。

```
Select the repositories to upload the package into:
- 1: test
```

4. 値 (1-1) を入力すると、選択が切り替わります。
5. **c** を押して選択を確認します。
6. アップロードするパッケージの場所を入力してください。場所が RPM の場合は、ファイルがアップロードされます。場所がディレクトリの場合は、そのディレクトリ内のすべての RPM がアップロードされます。

```
/home/localuser/bear-4.1-1.noarch.rpm
```

```
The following RPMs will be uploaded:
bear-4.1-1.noarch.rpm
```

7. **y** を押して続行するか、**n** を押してキャンセルします。

```
Uploaded /home/ec2-user/bear-4.1-1.noarch.rpm
```

検証

1. コンテンツのアップロードに成功した場合も同様のメッセージが表示されます。

```
Uploaded /home/ec2-user/bear-4.1-1.noarch.rpm
Total of 1 packages uploaded.
1 RPM(s) associated to repo test.
```

2.10. 遠隔地の WEB サイトからのコンテンツのアップロード (RPM コンテンツのみ)

リモートサーバーに保存されているパッケージを、手動でダウンロードせずにアップロードすることができます。パッケージは、HTTP、HTTPS、または FTP でアクセスする必要があります。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を押して、**リポジトリの管理** を選択します。
3. **Repository Management** 画面で、**ur** を押します。

```
Select the repositories to upload the package into:
- 1: test
```

4. 値 (1-1) を入力すると、選択が切り替わります。

5. **c** を押して選択を確認します。

```
### WARNING ### WARNING ### WARNING ### WARNING ### WARNING ###
WARNING ###
#                               #
# Content retrieved from non-Red Hat arbitrary places can contain #
# unsupported or malicious software. Proceed at your own risk. #
#                               #
#####
```

6. アップロードするパッケージのリモート URL を入力します。場所が RPM の場合は、ファイルがアップロードされます。場所が Web ページの場合、そのページからリンクされているすべての RPM がアップロードされます。

```
https://www.google.com/url?
q=https://repos.fedorapeople.org/pulp/pulp/demo_repos/zoo/bear-4.1-
1.noarch.rpm&sa=D&source=docs&ust=1637381734614000&usg=AOvVaw2LGPli3y-
CXIVAQJtbtKRw
Retrieving https://www.google.com/url?
q=https://repos.fedorapeople.org/pulp/pulp/demo_repos/zoo/bear-4.1-
1.noarch.rpm&sa=D&source=docs&ust=1637381734614000&usg=AOvVaw2LGPli3y-
CXIVAQJtbtKRw
Found 1 RPMs to download
Retrieving https://repos.fedorapeople.org/pulp/pulp/demo_repos/zoo/bear-4.1-1.noarch.rpm

The following RPMs will be uploaded:
bear-4.1-1.noarch.rpm
```

7. **y** を押して続行するか、**n** を押してキャンセルします。

```
Uploaded /tmp/rhui.318mfp81.tmp/bear-4.1-1.noarch.rpm
```

検証

1. コンテンツのアップロードに成功した場合も同様のメッセージが表示されます。

```
Uploaded /tmp/rhui.318mfp81.tmp/bear-4.1-1.noarch.rpm
Total of 1 packages uploaded.
1 RPM(s) associated to repo test.
```

2.11. パッケージグループメタデータのカスタムリポジトリへのインポート

RHUI ユーザーがパッケージグループまたは言語パックを表示およびインストールできるようにするには、**comps.xml** または **comps.xml.gz** ファイルをカスタムリポジトリにインポートできます。

この機能はコマンドラインインターフェイスでのみ使用できます。

前提条件

- 有効な **comps.xml** または **comps.xml.gz** ファイルがあることを確認する。

- RHUA ノードへの root アクセスがある。

手順

- RHUA ノードで、**rhui-manager** ユーティリティを使用して、**comps** ファイルからカスタムリポジトリにデータをインポートします。

```
# rhui-manager repo add_comps --repo_id Example_Custom_Repo --comps
/home/Example-Comps.xml
```

検証

- カスタムリポジトリを使用するクライアントシステムで以下を行います。
 - a. リポジトリデータを更新します。

```
# yum clean metadata
```

- b. リポジトリデータを一覧表示し、**comps** ファイルが更新されていることを確認します。

```
# yum grouplist
```

2.12. カスタムリポジトリからのコンテンツの削除 (カスタム RPM コンテンツのみ)

RHUI の Text User Interface (TUI) を使用して、カスタムリポジトリからパッケージを削除できます。

コマンドラインインターフェイス (CLI) コマンドについては、[「RHUI 4 CLI オプションの使用」](#) を参照してください。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **r** を入力して、**manage repositories** を選択します。
3. **Repository Management** 画面で **r** を入力し、**リポジトリから削除するパッケージを選択しま (カスタム RPM コンテンツのみ)**。

```
-- Repository Management --
```

```
l list repositories currently managed by the RHUI
i display detailed information on a repository
a add a new Red Hat content repository
ac add a new Red Hat container
c create a new custom repository (RPM content only)
d delete a repository from the RHUI
u upload content to a custom repository (RPM content only)
```

```

ur upload content from a remote web site (RPM content only)
p list packages in a repository (RPM content only)
r select packages to remove from a repository (Custom RPM content only)

```

- 値を入力してリポジトリを選択します。

```

Choose a repository to delete packages from:
1 - Test-RPM-1
2 - Test-RPM-2

```

- 値を入力して削除するパッケージを選択します。

```

Select the packages to remove:
- 1: example-package-1.noarch.rpm
- 2: example-package-2.noarch.rpm

```

- c**を入力して選択を確定します。

```

The following packages will be removed:
example-package-1.noarch.rpm

```

- y**を入力して続行するか、**n**を入力してキャンセルします。

```

Removed example-package-1.noarch.rpm

```

2.13. リポジトリ内のパッケージのリスト表示 (RPM コンテンツのみ)

Red Hat Update Infrastructure Management Tool 内のリポジトリのリストを表示する際に、パッケージ数が 100 未満のリポジトリのみがそのコンテンツを表示します。100 を超えるパッケージでは、パッケージ数のみが表示されます。

手順

- Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

- r**を押して、**リポジトリの管理**を選択します。
- Repository Management**画面で、**p**を押します。
- 表示するリポジトリの数を選択します。Red Hat Update Infrastructure Management Tool により、結果をフィルタリングするかどうかを尋ねられます。行を空白のままにして、フィルターなしで結果を確認します。

```
Enter value (1-1631) or 'b' to abort: 1
```

```
Enter the first few characters (case insensitive) of an RPM to filter the results
(blank line for no filter):
```

```
Only filtered results that contain less than 100 packages will have their
contents displayed. Results with more than 100 packages will display
a package count only.
```



```
Packages:  
bear-4.1-1.noarch.rpm
```

検証

1. 3種類あるメッセージの1つが表示されます。

```
Packages:  
bear-4.1-1.noarch.rpm
```

```
Package Count: 8001
```

```
No packages in the repository.
```

第3章 エンタイトルメント証明書とクライアント設定 RPM の作成

RHUI はエンタイトルメント証明書を使用して、リポジトリで要求を行うクライアントが、クラウドプロバイダーによって、これらのリポジトリへのアクセスが許可されているようにします。エンタイトルメント証明書は、クラウドプロバイダーの CA 証明書 (CA) で署名されている必要があります。CA 証明書が設定の一部として CDS にインストールされている。

3.1. RED HAT UPDATE INFRASTRUCTURE MANAGEMENT TOOL でのクライアントエンタイトルメント証明書の作成

Red Hat は、元のエンタイトルメント証明書を発行すると、要求したリポジトリへのアクセスが許可されます。クライアントエンタイトルメント証明書を作成すると、クライアントを細分化して、それぞれに個別の証明書を作成する方法を決定します。その後、各証明書を使用して個別の RPM を作成できます。

前提条件

- エンタイトルメント証明書が、クラウドプロバイダーの CA 証明書で署名されている。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **e** を押して **create entitlement certificates and client configuration RPMs** を選択します。
3. **e** を押して、**generate an entitlement certificate** を選択します。
4. プロンプトでリポジトリの番号を入力して、エンタイトルメント証明書に含めるリポジトリを選択します。リポジトリ数を入力すると、リポジトリ名前の横に x が付けられます。追加するすべてのリポジトリがチェックされるまで続行します。



重要

単一の RHEL バージョンのリポジトリのみを単一のエンタイトルメントに含めます。複数の RHEL バージョンのリポジトリを追加すると、使用できない **yum** 設定ファイルが発生します。

5. プロンプトで **c** を押して確認します。
6. 証明書の名前を入力します。この名前は、Red Hat Update Infrastructure Management Tool 内で証明書を特定し、証明書およびキーファイルの名前を生成するのに役立ちます。

```
Name of the certificate. This will be used as the name of the certificate file (name.crt) and its associated private key (name.key). Choose something that will help identify the products contained with it.
```

7. 証明書を保存するパスを入力します。フィールドを空白のままにして、現在の作業ディレクトリに保存します。
8. 証明書が有効な日数を入力します。365 日の場合はフィールドを空白のままにしておきます。証明書に含まれるリポジトリの詳細が表示されます。

Repositories to be included in the entitlement certificate:

Red Hat Repositories

Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Debug RPMs) from RHUI

Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI

Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI

Proceed? (y/n)

9. プロンプトが表示されたら **y** を押して、情報を確認し、エンタイトルメント証明書を作成します。

検証

1. エンタイトルメント証明書が作成されると、同様のメッセージが表示されます。

```
.....+++++
....+++++
Entitlement certificate created at ./rhel8-for-rhui4.crt
-----
```

3.2. CLI でのクライアントエンタイトルメント証明書の作成

Red Hat は、元のエンタイトルメント証明書を発行すると、要求したリポジトリへのアクセスが許可されます。クライアントエンタイトルメント証明書を作成すると、クライアントを細分化して、それぞれに個別の証明書を作成する方法を決定します。その後、各証明書を使用して個別の RPM を作成できます。

前提条件

- エンタイトルメント証明書が、クラウドプロバイダーの CA 証明書で署名されている。

手順

1. 以下のコマンドを使用して、CLI からエンタイトルメント証明書を作成します。

```
# rhui-manager client cert --repo_label rhel-8-for-x86_64-appstream-eus-rhui-source-rpms --
name rhuiclientexample --days 365 --dir /root/clientcert
.....+++++
.....+++++
Entitlement certificate created at /root/clientcert/rhuiclientexample.crt
```



注記

ID ではなく、Red Hat リポジトリのラベルを使用します。すべてのラベルのリストを表示するには、**rhui-manager client labels** コマンドを実行します。保護されたカスタムリポジトリを証明書に含める場合は、代わりにリポジトリの ID を使用します。

検証

1. エンタイトルメント証明書の作成に成功した場合も同様のメッセージが表示されます。

```
Entitlement certificate created at /root/clientcert/rhuiclientexample.crt
```

3.3. クライアントエンタイトルメント証明書が **FUTURE** 暗号化ポリシーに準拠しているかどうかの確認

クライアントのエンタイトルメント証明書を確認することで、RHUI のインスタンスが準拠している暗号化ポリシーを確認できます。

- RHUI バージョン 3.1 から 4.0 に生成される証明書は、**FIPS** および **DEFAULT** 暗号化ポリシーに準拠しています。
- RHUI バージョン 4.1 以降によって生成された証明書は、**FIPS**、**DEFAULT**、および **FUTURE** の暗号化ポリシーに準拠しています。

前提条件

- クライアントエンタイトルメント証明書の場所を把握している。
デフォルトの場所は `/etc/pki/rhui/product/content.crt` です。

手順

1. クライアント RPM、または RPM がインストールされているマシンで、クライアントエンタイトルメント証明書が保存されるパスを指定して以下のコマンドを実行します。

```
# openssl x509 -noout -text -in /etc/pki/rhui/product/content.crt | grep bit
```

2. RSA 鍵の長さを確認します。
 - 長さが 2048 ビットの場合には、クライアントエンタイトルメント証明書は **FUTURE** ポリシーに準拠していません。
 - 長さが 4096 ビットの場合には、クライアントエンタイトルメント証明書は **FUTURE** ポリシーに準拠しています。

関連情報

- [Red Hat Update Infrastructure Management Tool でのクライアントエンタイトルメント証明書の作成](#)
- [CLI でのクライアントエンタイトルメント証明書の作成](#)

3.4. CLI を使用したクライアント設定 RPM のリポジトリ ID 接頭辞の変更

RPM の作成時に、カスタムリポジトリ ID 接頭辞を設定するか、完全に削除できます。デフォルトでは、接頭辞は **rhui-** です。

手順

- RHUA ノードで、RHUI インストーラーコマンドを使用して接頭辞を設定または削除します。
 - カスタム接頭辞を設定します。

```
rhui-installer --rerun --client-repo-prefix CUSTOM_PREFIX
```

- 接頭辞の代わりに2つの引用符を使用して、接頭辞を完全に削除します。

```
rhui-installer --rerun --client-repo-prefix ""
```

3.5. RED HAT UPDATE INFRASTRUCTURE MANAGEMENT TOOL でのクライアント設定 RPM の作成

Red Hat は、元のエンタイトルメント証明書を発行すると、要求したリポジトリへのアクセスが許可されます。クライアントエンタイトルメント証明書を作成する場合、クライアントを細分化し、クライアントごとに個別の証明書を作成する方法を決定する必要があります。次に、各証明書を使用して、適切なゲストイメージへのインストール用に個別の RPM を作成できます。

この手順では、RHUI 管理ツールで RPM を作成します。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **e** を押して **create entitlement certificates and client configuration RPMs** を選択します。
3. **Client Entitlement Management** 画面で **c** を押し、**create a client configuration RPM from an entitlement certificate** を選択します。
4. ローカルディレクトリーの完全なパスを入力し、設定ファイルを保存します。

```
Full path to local directory in which the client configuration files generated by this tool should be stored (if this directory does not exist, it will be created):
```

5. RPM の名前を入力します。
6. 設定 RPM のバージョンを入力します。デフォルトのバージョンは 2.0 です。
7. 設定 RPM のリリースを入力します。デフォルトのバージョンは 1 です。
8. 特定のチャンネルにアクセスするためにクライアントを認証するエンタイトルメント証明書への完全パスを入力します。
9. エンタイトルメント証明書の秘密鍵への完全パスを入力します。
10. クライアント設定に含める保護されていないカスタムリポジトリを選択します。
11. **c** を押して選択するか、その他のコマンドには **?** を確定します。

検証

1. RPM が正常に作成されても、同様のメッセージが表示されます。

```
Successfully created client configuration RPM.
Location: /tmp/clientrpmtest-2.0/build/RPMS/noarch/clientrpmtest-2.0-1.noarch.rpm
```

3.6. CLI でのクライアント設定 RPM の作成

Red Hat は、元のエンタイトルメント証明書を発行すると、要求したリポジトリへのアクセスが許可されます。クライアントエンタイトルメント証明書を作成する場合、クライアントを細分化し、クライアントごとに個別の証明書を作成する方法を決定する必要があります。次に、各証明書を使用して、適切なゲストイメージへのインストール用に個別の RPM を作成できます。

この手順では、CLI で RPM を作成します。

手順

1. 以下のコマンドを使用して、RHUI CLI で RPM を作成します。

```
# rhui-manager client rpm --entitlement_cert /root/clientcert/rhuiclientexample.crt --
private_key /root/clientcert/rhuiclientexample.key --rpm_name clientrpmtest --dir /tmp --
unprotected_repos unprotected_repo1
Successfully created client configuration RPM.
Location: /tmp/clientrpmtest-2.0/build/RPMS/noarch/clientrpmtest-2.0-1.noarch.rpm
```



注記

CLI を使用して、RHUI リポジトリで使用するプロキシサーバーの URL を指定するか、**_none_** (アンダースコアを含む) を使用してクライアントマシンのグローバル **yum** 設定を上書きすることもできます。プロキシを指定するには、**-proxy** パラメーターを使用します。

検証

1. クライアント設定 RPM の作成に成功した場合も同様のメッセージが表示されます。

```
Successfully created client configuration RPM.
Location: /tmp/clientrpmtest-2.0/build/RPMS/noarch/clientrpmtest-2.0-1.noarch.rpm
```

第4章 RED HAT エンタイトルメント証明書の管理

4.1. RED HAT UPDATE APPLIANCE 証明書

RHUI の RHUA は以下の証明書およびキーを使用します。

- コンテンツ証明書および秘密鍵
- エンタイトルメント証明書および秘密鍵
- SSL 証明書および秘密鍵
- クラウドプロバイダーの CA 証明書

RHUA は、コンテンツ証明書とエンタイトルメント証明書で設定されます。RHUA はコンテンツ証明書を使用して Red Hat CDN に接続します。また、Red Hat CA 証明書を使用して Red Hat CDN への接続を確認します。RHUA は Red Hat CDN に接続する唯一のコンポーネントであるため、この証明書がデプロイされた唯一の RHUI コンポーネントになります。複数の RHUI インストールが同じコンテンツ証明書を使用できることに注意してください。たとえば、Amazon EC2 クラウドは、(リージョンごとに) RHUI インストールを複数実行しますが、各 RHUI インストールは同じコンテンツ証明書を使用します。

クライアントはエンタイトルメント証明書を使用して RHUI のパッケージへのアクセスを許可します。環境ヘルスチェックを実行するには、RHUA は各 CDS に対して **yum** 要求を試みます。成功するには、**yum** リクエストに有効なエンタイトルメント証明書を指定する必要があります。

4.2. コンテンツ配信サーバー証明書

RHUI の各 CDS ノードは以下の証明書およびキーを使用します。

- SSL 証明書および秘密鍵
- クラウドプロバイダーの CA 証明書

CDS に必要な唯一の証明書は SSL 証明書で、クライアントと CDS 間の HTTPS 通信を許可します。SSL 証明書は特定のホスト名にスコープ設定されているため、各 CDS ノードには一意の SSL 証明書が必要になります。CDS への接続時に SSL エラーが発生する場合は、証明書のコモンネームが、インストールされている CDS の完全修飾ドメイン名 (FQDN) に設定されていることを確認してください。

CA 証明書は、**yum** 要求の一部としてクライアントが送信したエンタイトルメント証明書がクラウドプロバイダーにより署名されたことを確認するために使用されます。これにより、不正なインスタンスが RHUI 内で不正に使用するための独自のエンタイトルメント証明書を生成することを回避できます。

4.3. クライアント証明書

RHUI の各クライアントは、エンタイトルメント証明書と秘密鍵、およびクラウドプロバイダーの CA 証明書を使用します。

エンタイトルメント証明書とその秘密鍵により、CDS からクライアントへの情報の暗号化が有効になります。各クライアントは、CDS に接続する際にエンタイトルメント証明書を使用して、パッケージをダウンロードするパーミッションがあることを証明します。すべてのクライアントは単一のエンタイトルメント証明書を使用します。

クラウドプロバイダーの CA 証明書は、接続時に CDS の SSL 証明書を検証するために使用されます。これにより、悪意のあるインスタンスが CDS を偽装して、悪意のある可能性のあるパッケージをクライアントに導入することがなくなります。

CA 証明書は、エンタイトルメント証明書ではなく、SSL 証明書を検証します。CDS ノードの逆は true です。SSL 証明書と秘密鍵は、クライアントから CDS へのデータを暗号化するために使用されます。CDS 上にある CA 証明書は、CDS ノードがクライアントによって送信されたエンタイトルメント証明書を信頼する必要があることを確認します。

4.3.1. 証明書の対象製品をリスト表示

Entitlements Manager 画面は、現在の Red Hat コンテンツ証明書でエンタイトルメントの付いた製品のリストを表示し、新しい証明書をアップロードするのに使用されます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **n** を押して **manage Red Hat entitlement certificates** を選択します。
3. **Entitlements Manager** 画面で **l** を押して、現在のコンテンツ証明書に関するデータをリスト表示します。

```
rhui (entitlements) => l
```

```
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Debug RPMs) from RHUI
Expiration: 02-27-2022 Certificate: c885597492374720bb5d398c3f65d1ed.pem
```

```
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI
Expiration: 02-27-2022 Certificate: c885597492374720bb5d398c3f65d1ed.pem
```

```
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI
Expiration: 02-27-2022 Certificate: c885597492374720bb5d398c3f65d1ed.pem
```

```
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI
Expiration: 02-27-2022 Certificate: c885597492374720bb5d398c3f65d1ed.pem
```

```
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (RPMs) from RHUI
Expiration: 02-27-2022 Certificate: c885597492374720bb5d398c3f65d1ed.pem
```

```
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Source RPMs) from RHUI
Expiration: 02-27-2022 Certificate: c885597492374720bb5d398c3f65d1ed.pem
```

検証

1. 現在の Red Hat コンテンツ証明書に含まれるエンタイトルされた製品のリストが表示されます。

4.3.2. カスタムリポジトリエンタイトルメントのリスト表示

Entitlements Manager 画面を使用して、カスタムリポジトリのエンタイトルメントをリスト表示できます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **n** を押して **manage Red Hat entitlement certificates** を選択します。
3. **Entitlements Manager** 画面で **c** を押して、カスタムリポジトリエンタイトルメントに関するデータをリスト表示します。

```
rhui (entitlements) => c
```

```
Custom Repository Entitlements
```

```
For each entitlement URL listed, the corresponding repositories that are  
configured with that entitlement are listed.
```

```
/protected/$basearch/os
```

```
Name: Repo 1
```

```
URL: protected/i386/os
```

```
Name: Repo 2
```

```
URL: protected/x86_64/os
```

第5章 SYNCHRONIZATION の状況確認とスケジューリング

リポジトリは、ソフトウェアパッケージ (RPM) のストレージの場所です。RHEL は、**yum** コマンドを使用して、RPM の検索、ダウンロード、インストール、および設定を行います。RPM には、アプリケーションの実行に必要なすべての依存関係が含まれます。RPM は、リポジトリのソフトウェアの更新もダウンロードします。

Red Hat コンテンツの初回同期は時間によって異なります。できるだけ早くリポジトリの同期を開始する場合は、CLI で **rhui-manager repo sync_all** を実行して、Red Hat Update Infrastructure 4 のすべてのリポジトリを同期できます。

5.1. リポジトリ同期の概要表示

Synchronization Status 画面では、特定のリポジトリに関する情報を表示することができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面から、**dr** を押します。

```
-- Repository Summary Synchronization Status --
Last Refreshed: 02:01:22
(updated every 5 seconds, ctrl+c to exit)

Last Sync          Last Result
-----
Red Hat Enterprise Linux 8 for ARM 64 - BaseOS (Debug RPMs) from RHUI (8)
  Never              None
....
....
Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (Debug RPMs) (8.2)
  2021-07-29 17:45:41      Running
  Associating Content: 11001 (97%)
  Downloading Artifacts: 7376
```

5.2. 実行同期の表示

Synchronization Status 画面では、実行同期タスクの状況を確認することができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。

3. **Synchronization Status** 画面から **rr** を押します。

```

Last Refreshed: 02:06:46
(updated every 5 seconds, ctrl+c to exit)

Current Sync          Result
-----
Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (Debug RPMs) (8.2)
  2021-07-29 17:45:41      Running
Associating Content: 11001 (97%)
Downloading Artifacts: 7376

```

5.3. 前回のリポジトリ同期の詳細を表示

Synchronization Status 画面では、前回のリポジトリ同期の詳細を確認することができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面から **vr** を押します。
4. 詳細を確認するリポジトリの番号を入力します。

```
Enter value (1-66) or 'b' to abort:
```

検証

1. 選択したリポジトリが同期されていない場合も同様のメッセージが表示されます。

```
Repo: Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (Debug RPMs) (8.2)
No syncs have been completed for this repository.
```

5.4. 個別リポジトリの即時同期

コンテンツの初期同期には、通常 10 分から 20 分かかります。できるだけ早くリポジトリの同期を開始する場合は、CLI で **rhui-manager repo sync_all** を実行して、Red Hat Update Infrastructure 4 のすべてのリポジトリを同期できます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面から **sr** を押します。

Select one or more repositories to schedule to be synchronized before its scheduled time. The sync will happen as soon as possible depending on other tasks that may be executing in the RHUI. Sync requests for repositories with tasks in running or pending state will be ignored.

Last Result	Next Sync	Repository

- リポジトリ名の横にある値を入力してリポジトリを選択します。製品の選択を確認する前に、一度に1つのリポジトリ選択を入力します。

```
x 714: Error    2021-11-17 20:30:00  Red Hat Enterprise Linux 8 for ARM 64 - AppStream
(RPMs) from RHUI (8.4)
```

- c** を押して確認します。

```
The following repositories will be scheduled for synchronization:
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.4)
Proceed? (y/n) y
```

- y** を押して続行します。

```
Scheduling sync for Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI
(8.4)...
... successfully scheduled for the next available timeslot.
```



注記

このメッセージは、選択したリポジトリのタスクが実行中であるかどうかを示します。 **Ignoring sync request for Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (Debug RPMs) (8.2) as the repo is currently reserved by a running task.**

5.5. アクティブな同期タスクのキャンセル

ほとんどの環境では、スケジュールされているリポジトリを同期します。アクティブな同期タスクをキャンセルする必要がある状況が発生する場合があります。

前提条件

- 既存のリポジトリがある。
- アクティブな同期タスクがある。

手順

- Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

- s** を押して、**同期のステータスとスケジューリング** を選択します。
- Synchronization Status** 画面から **ca** を押して、**cancel active sync tasks** を選択します。

4. 取り消すタスクまたはタスクの値を入力します。

```
Select one or more repositories for which you want to cancel their active tasks.
- 1: Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (Debug RPMs) (8.2)
Enter value (1-1) to toggle selection, 'c' to confirm selections, or '?' for more commands:
```

5. **c** を押して選択を確認します。
6. **y** を押して、1つまたは複数の同期タスクをキャンセルします。

```
The active tasks will be canceled for the following repositories:
Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (Debug RPMs) (8.2)
Proceed? (y/n)
```

検証

1. アクティブな同期タスクをキャンセルすると、この例と同様のメッセージが表示されます。

```
Canceling active task for repo Red Hat Enterprise Linux 8 for x86_64 - AppStream from
RHUI (Debug RPMs) (8.2) ...
... done
```

5.6. 待機中の同期タスクのキャンセル

ほとんどの環境では、スケジュールされているリポジトリを同期します。保留中の同期タスクをキャンセルする必要がある状況が発生する可能性があります。

前提条件

- 既存のリポジトリがある。
- 同期タスクがスケジュールされている。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面で **cw** を押し、**cancel waiting sync tasks** を選択します。
4. 取り消すタスクまたはタスクの値を入力します。

```
Select one or more repositories for which you want to cancel their pending tasks.
- 1: Single Sign-On 7.4 for RHEL 8 x86_64 (Source RPMs) from RHUI
Enter value (1-1) to toggle selection, 'c' to confirm selections, or '?' for more commands: 1
```

5. **c** を押して選択を確認します。

```
Select one or more repositories for which you want to cancel their pending tasks.
x 1: Single Sign-On 7.4 for RHEL 8 x86_64 (Source RPMs) from RHUI
Enter value (1-1) to toggle selection, 'c' to confirm selections, or '?' for more commands: c
```

6. **y** を押して続行します。

```
The pending tasks will be canceled for the following repositories:
Single Sign-On 7.4 for RHEL 8 x86_64 (Source RPMs) from RHUI
Proceed? (y/n) y
```

検証

1. キャンセルに成功した場合も同様のメッセージが表示されます。

```
Canceling pending task for repo Single Sign-On 7.4 for RHEL 8 x86_64 (Source RPMs) from
RHUI ...
... done
```

2. 保留中の同期タスクがない場合は、以下のメッセージが表示されます。

```
There are no repositories with pending sync related tasks.
```

5.7. リポジトリの自動公開ステータスの表示と変更

Synchronization Status 画面では、リポジトリの自動公開ステータスを確認変更することができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面から、**ap** を押します。

```
rhui (sync) => ap
```

```
Select one or more repositories to toggle the auto-publish status.
The operation will be executed as soon as possible depending on other tasks
that may be executing in the RHUI.
```

```
      Status | Repository
-----|-----
```

```
Select one or more repositories:
```

```
Custom Repositories
```

```
Red Hat Repositories: yum
```

```
- 713:  AUTO Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI
(8.3)
```

- 714: AUTO Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.4)
- 719: AUTO Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.3)
- 720: AUTO Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8.4)

4. 値 (**1-1631**) を入力すると選択が切り替わり、**c** 選択が確定し、さらにコマンドを入力する場合は **?** を選択します。

```
The following repositories will have their auto-publish status changed:
Red Hat Repositories
yum
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8)
```

5. **c** を押して選択を確認します。
6. **y** を押して続行します。

検証

1. 選択して確定したときにも同様のメッセージが表示されます。

```
Scheduling a task to turn off auto-publish status of repository Red Hat Enterprise Linux 8 for
ARM 64 - AppStream (RPMs) from RHUI (8)
```

関連情報

- [Red Hat Enterprise Linux repositories that can be delivered through Red Hat's Certified Cloud and Service Provider \(CCSP\) partners via RHUI](#)

5.8. リポジトリのワークフローの表示と進行

Synchronization Status 画面を使用して、リポジトリのワークフローを確認したり変更したりすることができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面から **wf** を押します。
4. 値 (**1-1631**) を入力すると選択が切り替わり、**c** 選択が確定し、さらにコマンドを入力する場合は **?** を選択します。

```
The following repositories will be scheduled for workflow push:
Red Hat Repositories
yum
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (RPMs) from RHUI (8.4)
```

5. **y** を押して続行します。

検証

1. スケジューリングに成功した場合も同様のメッセージが表示されます。

```
Scheduling a task for generating metadata version 0 for repo Red Hat Enterprise Linux 8 for
ARM 64 - AppStream (RPMs) from RHUI (8.4) ...
... task scheduled.
```

5.9. リポジトリのファイルシステムへのエクスポート

Synchronization Status 画面を使用して、リポジトリをファイルシステムにエクスポートすることができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **s** を押して、**同期のステータスとスケジューリング** を選択します。
3. **Synchronization Status** 画面から、**ex** を押します。
4. 値を入力すると、選択が切り替わります。
5. **c** を押して選択を確認します。

```
The following repositories will be exported:
Red Hat Repositories
yum
Red Hat Enterprise Linux 8 for ARM 64 - AppStream (Source RPMs) from RHUI (8)
```

6. **y** を押して続行します。

検証

1. リポジトリがファイルシステムにエクスポートされている場合も、同様のメッセージが表示されます。

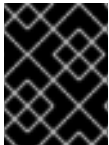
```
[1/1] Exporting version 1 of the repo Red Hat Enterprise Linux 8 for ARM 64 - AppStream
(Source RPMs) from RHUI (8).
```


第6章 コンテンツデリバリーサーバーの管理

CD ノードは、コンテンツ配信ネットワーク (CDN) の主なコンポーネントで、クライアントに高可用性を提供します。地理的に分散した方法でサーバーを実行すると、応答時間も改善します。

Content Delivery Server (CDS) Management画面では、CDS ノードのリスト表示、追加、削除、再インストールを行うことができます。

必要なデバイスを割り当てる際の最善のアクションは、クラウドプロバイダーによって決定されます。Red Hat Update Infrastructure Management Tool 設定 RPM は、パッケージディレクトリーを NGINX 設定にリンクして提供します。



重要

NGINX 設定は、CDS のインストール時に `/etc/nginx/conf.d/ssl.conf` ファイルによって処理されます。

複数のクライアントがリポジトリーに対して更新に問題がある場合は、RHUI で問題があることを示す可能性があります。

6.1. RHUI 4 で管理されているすべての既知の CDS インスタンスのリストアップ

Content Delivery Server (CDS) Management画面を使用して、Red Hat Update Infrastructure 4 が管理するすべての CDS ノードをリストアップすることができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **c** を押して、**manage content delivery servers (CDS)** を選択します。
3. **Content Delivery Server (CDS) Management**画面で **l** を押して、Red Hat Update Infrastructure 4 が管理する既知の CDS ノードをすべてリストアップします。

```
Hostname:      <cds1.example.com>
SSH Username:  <cloud-user>
SSH Private Key:  /<cloud-user>/.ssh/id_rsa_rhua
```

6.2. CDS の新規登録について

Red Hat Update Infrastructure Management Tool は、RHUI 内で CDS を設定するためのいくつかのオプションを提供します。

前提条件

- CDS ノードで **sshd** が稼働しており、**ポート 443** が開いていることを確認する。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **c** を押して、**manage content delivery servers (CDS)** を選択します。
3. **Content Delivery Server (CDS) Management** 画面で **a** を押して、新しい CDS インスタンスを追加します。
4. 追加する CDS のホスト名を入力します。

```
Hostname of the CDS instance to register:
cds1.example.com
```

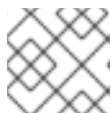
5. CDS への SSH アクセスがあり、**sudo** 権限を持つユーザー名を入力します。

```
Username with SSH access to <cds1.example.com> and sudo privileges:
<cloud-user>
```

6. CDS にログインするために SSH 秘密鍵への絶対パスを入力し、**Enter** を押します。

```
Absolute path to an SSH private key to log into <cds1.example.com> as <cloud-user>:
/home/<cloud-user>/.ssh/id_rsa_rhua
```

7. **オプション:** カスタム SSL 証明書を使用する場合は、カスタム SSL 証明書、SSL キー、および SSL crt ファイルへの絶対パスを入力します。



注記

SSL 証明書を指定しない場合は、自動的に生成されます。

```
Optional absolute path to user supplied SSL key file:
/home/<cloud-user>/custom_ssl.key
```

```
Optional absolute path to user supplied SSL crt file:
/home/<cloud-user>/custom_ssl.crt
```

```
.....
The following CDS has been successfully added:
```

```
Hostname:      <cds1.example.com>
SSH Username:  <cloud-user>
SSH Private Key: /home/<cloud-user>/.ssh/id_rsa_rhua
```

```
The CDS will now be configured:
```

```
.....
The CDS was successfully configured.
```

8. コンテンツ配信サーバーを追加できない場合は、ファイアウォールルールが RHUA と CDS 間のアクセスを許可していることを確認します。
9. **mount** コマンドを実行して、共有ストレージが読み取り/書き込みとしてマウントされているかどうかを確認します。

```
[root@rhua ~]# mount | grep rhui
```

```
nfs.example.com:/export on /var/lib/rhui/remote_share type nfs4
(rw,relatime,vers=4.2,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,re
trans=2,sec=sys,clientaddr=10.8.41.163,local_lock=none,addr=10.8.41.163)
```

10. 設定に成功すると、残りのすべての CDS ノードでこの手順を繰り返します。

6.3. CDS への設定の再インストールおよび再適用

CDS の設定を再インストールして再適用する必要がある場合に、状況が発生する可能性があります。Red Hat Update Infrastructure Management Tool では、このタスクを簡単に実行できます。

前提条件

- CDS が少なくとも1つインストールされている。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **c** を押して、**manage content delivery servers (CDS)** を選択します。
3. **Content Delivery Server (CDS) Management** 画面で、**r** を押して、**reinstall and reapply configuration to an existing CDS instance** を選択します。Red Hat Update Infrastructure Management Tool は、すべての再インストールおよび再設定タスクを自動的に実行します。
4. 再インストールする CDS を選択します。

```
1 -
Hostname:      <cds1.example.com>
SSH Username:  <cloud-user>
SSH Private Key:  /<cloud-user>/.ssh/id_rsa_rhua
```

5. 値または **b** を入力して中断します。1:

```
Checking that the RHUA services are reachable from the instance...
Done.
```

```
Installing and configuring the CDS...
```

```
PLAY [Registering a CDS instance] *****
```

```
PLAY RECAP *****
```

```
cloud-user@cds1.example.com : ok=24  changed=10  unreachable=0  failed=0
skipped=2  rescued=0  ignored=0
```

```
Done.
```

検証

1. コードの出力を表示して、CDS が正常に再インストールおよび再設定されたことを確認します。

```
Ensuring that instance ports are reachable ...
Done.
```

6.4. レガシー CA を受け入れる CDS の設定

デフォルトでは、コンテンツ配信サーバー (CDS) ノードは、RHUI システムに現在設定されている認証局 (CA) が署名したエンタイトルメント証明書だけを受け入れます。ただし、メインの CA を変更する場合や、CA 証明書の有効期限が切れた場合に、クライアントが動作を継続できるように、以前に作成した CA を受け入れる必要がある場合があります。

この手順では、CDS ノードに CA 証明書をインストールして、RHUI でレガシー CA をサポートする手順を説明します。

前提条件

- RHUI の最新バージョンを実行していることを確認する。



注記

以前のバージョンの RHUI をインストールしている場合は、**rhui-manager** で CDS ノードを再インストールする必要があります。

手順

1. CDS ノードで、`/etc/pki/rhui/legacy` ディレクトリーがない場合は作成します。

```
# mkdir /etc/pki/rhui/legacy
```

2. レガシー CA 証明書をディレクトリーに保存します。

検証

- CDS ノードは、CA 証明書をディレクトリーに保存するとすぐにレガシー CA の受け入れを開始します。

6.5. レガシー CA の受け入れ停止の CDS 設定

コンテンツ配信サーバー (CDS) ノードがレガシー認証局 (CA) を受け入れないように制限するには、それぞれの CA 証明書を削除します。

前提条件

- クライアントが対象の CA を使用していない。

手順

1. CDS ノードで、`/etc/pki/rhui/legacy/` ディレクトリーに移動します。

```
# cd /etc/pki/rhui/legacy/
```

2. オプション:既存の CA 証明書をバックアップします。
3. 制限する CA に対応する CA 証明書を削除します。

```
# rm example-legacy.crt
```

検証

- CDS ノードは、CA 証明書の削除直後にレガシー CA の受け入れを停止します。

6.6. CDS の登録解除

使用しない CDS インスタンスを登録解除 (削除) することができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **c** を押して、**manage content delivery servers (CDS)** を選択します。
3. **Content Delivery Server (CDS) Management**画面で **d** を押して CDS インスタンスを削除します。
4. 削除する CDS のホスト名を入力します。

```
Hostname of the CDS instance to unregister:  
cds1.example.com
```

第7章 HAPROXY ロードバランサーインスタンスの管理

複数の CDS が使用される場合は、負荷分散ソリューションを準備して、クライアント HTTPS 要求をすべてのサーバーに分散する必要があります。Red Hat Update Infrastructure 4 には HAProxy が同梱されていますが、インストール時に使用する負荷分散ソリューション (クラウドプロバイダーからのソリューションなど) を選択するのはユーザー次第です。HAProxy が使用されている場合は、挿入するノードの数を決定する必要もあります。

7.1. RHUI 4 で管理されているすべての HAPROXY ロードバランサーインスタンスのリストアップ

Load-balancer (HAProxy) Management 画面では、RHUI 4 が管理している既知の HAProxy ロードバランサーインスタンスをすべて表示することができます。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **I** を押して、**manage HAProxy load-balancer instances** を選択します。
3. **Load-balancer (HAProxy) Management** 画面で **I** を押すと、RHUI が管理するロードバランサーインスタンスがリスト表示されます。

```
Hostname:      <haproxy1.example.com>
SSH Username:  <cloud-user>
SSH Private Key:  /<cloud-user>/.ssh/id_rsa_rhua
```

7.2. HAPROXY ロードバランサーの新規登録について

Red Hat Update Infrastructure 4 は DNS を使用して CDN に到達します。ほとんどの場合、インスタンスは、クラウドのインフラストラクチャーの一部としてホストされる適切な DNS サーバーと通信するように事前設定する必要があります。独自の DNS サーバーを実行するか、クライアント DNS 設定を更新すると、**yum Could not contact any CDS load balancers** のようなエラーが表示される可能性があります。このような場合は、DNS サーバーがリクエストのためにクラウドの DNS サーバーに転送していること、または DNS クライアントが名前解決のためにクラウドの DNS サーバーにフォールバックするように設定されていることを確認してください。

複数の HAProxy ノードを使用するには、すべての HAProxy ノードの IP アドレスを解決する **rhui-installer** の実行時に **--cds-lb-hostname** パラメーターの値として使用されるホスト名のラウンドロビン DNS エントリ (このガイドでは **cds.example.com**) が必要です。[DNS Round Robin の設定方法](#) により、ラウンドロビン DNS を設定する方法があります。Red Hat Update Infrastructure 4 のコンテキストでは、これらは HAProxy ノードの IP アドレスで、**rhui-installer** を呼び出す際に、**--cds-lb-hostname** で指定したホスト名にマップされます。

前提条件

1. HAProxy ロードバランサーノードで **sshd** が稼働しており、**port 443** が開いていることを確認する。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **I** を押して、**manage HAProxy load-balancer instances** を選択します。
3. **Load-balancer (HAProxy) Management** 画面で **a** を押して、新しいロードバランサーインスタンスを追加します。
4. 新しいロードバランサーのホスト名を入力します。

```
Hostname of the HAProxy Load-balancer instance to register:
<haproxy1.example.com>
```

5. ロードバランサーへの SSH アクセスがあり、`sudo` 権限を持つユーザー名を入力します。

```
Username with SSH access to cds.example.com and sudo privileges:
<cloud-user>
```

6. ロードバランサーインスタンスにログインするために SSH 秘密鍵に絶対的な部分を入力し、**Enter** を押します。

```
Absolute path to an SSH private key to log into cds.example.com as <cloud-user>:
/<cloud-user>/ssh/id_rsa_rhua
```

7. **オプション**:ユーザー提供の HAProxy 設定ファイルへのオプションの絶対パスを入力し、**Enter** を押します。
カスタム設定ファイルへのパスを指定しない場合は、代わりにデフォルトのファイル `/usr/share/rhui-tools/templates/haproxy.cfg` が使用されます。

```
Optional absolute path to user supplied HAProxy config file:
```

```
.....
The following load-balancer has been successfully added:
```

```
Hostname:      <haproxy1.example.com>
SSH Username:  <cloud-user>
SSH Private Key: /<cloud-user>/ssh/id_rsa_rhua
```

```
The load-balancer will now be configured:
```

8. ロードバランサーの追加に失敗した場合、ファイアウォールのルールで RHUA とロードバランサー間のアクセスが許可されているか確認してください。
9. 設定に成功すると、残りのロードバランサーインスタンスについてこの手順を繰り返します。

検証

- 以下のメッセージが表示されます。

```
The HAProxy Load-balancer was successfully configured.
```

関連情報

- [HAProxy の設定](#)

7.3. HAPROXY ロードバランサーの再インストールと設定の再適用について

HAProxy ロードバランサーの設定を再インストールして再適用する必要がある場合に、状況が発生する可能性があります。Red Hat Update Infrastructure Management Tool では、このタスクを簡単に実行できます。

前提条件

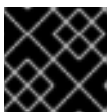
- HAProxy ロードバランサーノードで **sshd** が稼働しており、**port 443** が開いていることを確認する。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **I** を押して、**manage HAProxy load-balancer instances** を選択します。
3. **Load-balancer (HAProxy) Management** 画面で **r** を押すと、ロードバランサーのインスタンスに設定を再インストールして再適用します。



重要

復元に含まれるファイルは、現在の属性を保持することが重要です。

Red Hat Update Infrastructure Management Tool は、すべての再インストールおよび再設定タスクを自動的に実行します。

1. 再インストールするロードバランサーを選択します。

```
1 -
Hostname:      <haproxy1.example.com>
SSH Username:  <cloud-user>
SSH Private Key:  /<cloud-user>/.ssh/id_rsa_rhua
```

2. 値または **b** を入力して中断します。1:

```
Installing and configuring the HAProxy Load-balancer...

PLAY [Registering a load balancer instance] *****

PLAY RECAP *****
cloud-user@haproxy1.example.com : ok=8  changed=3  unreachable=0  failed=0
skipped=0  rescued=0  ignored=0

Done.
```


検証

1. コードの出力を表示して、ロードバランサー が正常に再インストールおよび再設定されたことを確認します。

```
Ensuring that HAProxy is available...  
Done.
```

7.4. HAPROXY ロードバランサーの登録解除

使用しない HAProxy ロードバランサーのインスタンスを登録解除 (削除) することができます。

前提条件

- HAProxy ロードバランサーノードで **sshd** が稼働しており、**port 443** が開いていることを確認する。

手順

1. Red Hat Update Infrastructure Management Tool のホーム画面に移動します。

```
[root@rhua ~]# rhui-manager
```

2. **l** を押して、**manage HAProxy load-balancer instances** を選択します。
3. **Load-balancer (HAProxy) Management** 画面で **d** を押してロードバランサーインスタンスを削除します。
4. 削除するロードバランサーのホスト名を入力してください。

```
Hostname of the load-balancer instance to unregister:  
<haproxy1.example.com>
```

第8章 コンテナの管理

RHUI を使用して、Linux コンテナ内のアプリケーションのデプロイメントを自動化できます。コンテナを使用すると、次の利点があります。

- 必要なストレージとメモリー内の領域が仮想マシンに比べて少ない:コンテナはアプリケーションの実行に必要なものだけを保持するため、コンテナを使用すると、オペレーティングシステム全体を含む仮想マシンよりも保存と共有が効率的になります。
- パフォーマンスを向上:完全に別個のオペレーティングシステムを実行していないため、コンテナは通常、新しい仮想マシン全体のオーバーヘッドを実行するアプリケーションよりも高速に動作します。
- セキュア:通常、コンテナには独自のネットワークインターフェイス、ファイルシステム、およびメモリーがあるため、そのコンテナで実行されているアプリケーションは、ホストコンピュータ上の他のアクティビティから分離して保護できます。
- 柔軟:アプリケーションのランタイム要件がコンテナ内のアプリケーションに組み込まれているため、コンテナは複数の環境で実行できます。



注記

コンテナのサポートは、デフォルトでは有効になっていません。**--container-support-enabled** フラグを設定して、サポートを手動で有効にする必要があります。

8.1. RED HAT UPDATE INFRASTRUCTURE のコンテナについて

コンテナはアプリケーションのサンドボックスです。各コンテナは、必要な設定データを保持するイメージに基づいています。イメージからコンテナを起動すると、書き込み可能な階層がこのイメージの上部に追加されます。コンテナをコミットするたびに、変更を保存する新しいイメージ層が追加されます。

イメージは、変更されることのない読み取り専用レイヤーです。変更はすべて最上位の書き込み可能なレイヤーで行われ、新規イメージを作成することによってのみ保存できます。各イメージは、複数の親イメージに依存します。

プラットフォームイメージは、親のないイメージです。プラットフォームイメージは、コンテナ化されたアプリケーションの実行に必要なランタイム環境、パッケージ、およびユーティリティを定義します。プラットフォームイメージは読み取り専用であるため、その上部にスタックされたコピーイメージに変更が反映されます。

8.2. RED HAT UPDATE INFRASTRUCTURE へのコンテナの追加

rhui-manager ツールを使用して、Repository Management セクションを使用してコンテナを追加できます。

手順

1. RHUA ノードで、コンテナのサポートを有効にし、これをすべての CDS ノードに適用します。

```
# rhui-installer --rerun --container-support-enabled True
# rhui-manager --noninteractive cds reinstall --all
```

2. オプション:`/etc/rhui/rhui-tools.conf` ファイルを編集し、`[container]` セクションの次の行を削除して、RHUI 設定でコンテナレジストリー認証情報を設定します。
RHUI 4.1.1 以降をクリーンインストールしている場合、最後の数行には、podman 固有のオプションと便利なコメントが含まれる `[container]` セクションが含まれます。以前のバージョンの RHUI から更新した場合、このセクションは `etc/rhui/rhui-tools.conf.rpmnew` ファイルの最後にあり、`rhui-tools.conf` ファイルにコピーできます。

```
[container]
...
registry_username: your_RH_login
registry_password: your_RH_password
```



注記

通常、`registry.redhat.io` とは異なるレジストリーから同期する場合は、それに応じて `registry_url` オプションと `registry_auth` オプションの値も変更します。

3. RHUA ノードで、`rhui-manager` を実行します。

```
# rhui-manager
```

4. `r` を押して、Repository Management 画面にアクセスします。

```
-= Red Hat Update Infrastructure Management Tool =-

-= Repository Management =-

I list repositories currently managed by the RHUI
i display detailed information on a repository
a add a new Red Hat content repository
ac add a new Red Hat container
c create a new custom repository (RPM content only)
d delete a repository from the RHUI
u upload content to a custom repository (RPM content only)
ur upload content from a remote web site (RPM content only)
p list packages in a repository (RPM content only)

Connected: rhua.example.com
```

5. `ac` を押して、新しい Red Hat コンテナを追加します。

```
rhui (repo) => ac Specify URL of registry [https://registry.redhat.io]:
```

6. 追加するコンテナがデフォルト以外のレジストリーに存在する場合は、レジストリーの URL を入力します。デフォルトのレジストリーを使用するには、何も入力せずに **Enter** キーを押します。
7. レジストリーにコンテナの名前を入力します。

```
jboss-eap-6/eap64-openshift
```

8. コンテナの一意的 ID を入力します。

rhui-manager は、スラッシュとドットをアンダースコアに置き換えることにより、コンテナの名前をレジストリーから Pulp で使用できる形式に変換します。Enter キーを押すか、任意の名前を入力することで、このように変換された名前を使用できます。

9. コンテナの表示名を入力します。

```
jboss-eap-6_eap64-openshift
```

10. オプション:プロンプトが表示されたら、RHUI 設定でログインとパスワードを設定します。
11. 表示された概要を確認します。

```
The following container will be added:  
Registry URL: http://registry.redhat.io  
Container Id: jboss-eap-6_eap64-openshift  
Display Name: jboss-eap-6_eap64-openshift  
Upstream Container Name: jboss-eap-6/eap64-openshift  
Proceed? (y/n)
```

12. **y** を押して続行し、コンテナを追加します。

```
y  
Successfully added container jboss-eap-6_eap64-openshift
```

8.3. コンテナリポジトリーの同期

コンテナを Red Hat Update Infrastructure に追加したら、**rhui-manager** ツールを使用してコンテナを同期できます。

手順

1. RHUA ノードで、**rhui-manager** を実行します。

```
# rhui-manager
```

2. **s** を押して、**synchronization status and scheduling** 画面にアクセスします。
3. **sr** を押して、個々のリポジトリーを即時に同期します。
4. 同期するリポジトリーの数を入力します。
5. **c** を押して選択を確認します。
6. リポジトリーを確認し、**y** を押して同期するか、**n** を押してキャンセルします。

```
The following repositories will be scheduled for synchronization: jboss-eap-6_eap64-  
openshift  
Proceed? (y/n) y  
Scheduling sync for jboss-eap-6_eap64-openshift...  
... successfully scheduled for the next available timeslot.
```

8.4. コンテナクライアント設定の生成

RHUI クライアントは、クライアント設定を使用して RHUI からコンテナをプルできます。RPM にはロードバランサーの証明書が含まれており、これを使用して、ロードバランサーをコンテナレジストリーに追加し、コンテナ設定を変更できます。

手順

1. RHUA ノードで、**rhui-manager** を実行します。

```
# rhui-manager
```

2. **e** を押して、**entitlement certificates and client configuration RPMs** 画面にアクセスします。
3. **d** を押して、**create a container client configuration RPM** (コンテナクライアント設定 RPM の作成) を実行します。
4. 設定ファイルを保存するローカルディレクトリーの完全パスを入力します。

```
/root/
```

5. RPM の名前を入力します。

```
containertest
```

6. 設定 RPM のバージョン番号を入力します。デフォルトは **2.0** です。
7. 設定 RPM のリリース番号を入力します。デフォルトは **1** です。
8. 証明書が有効な日数を入力します。デフォルトは **365** です。

```
Successfully created client configuration RPM.
```

```
Location: /root/containertest-2.0/build/RPMS/noarch/containertest-2.0-1.noarch.rpm
```

8.5. クライアントへのコンテナ設定 RPM のインストール

コンテナ設定 RPM を生成したら、それをローカルマシンにインポートすることでクライアントにインストールできます。

手順

1. RPM を RHUA ノードからローカルマシンに取得します。

```
# root@rhua.example.com:/root/containertest-2.0/build/RPMS/noarch/containertest-2.0-1.noarch.rpm .
```

2. RPM をローカルマシンからクライアントに転送します。

```
# scp containertest-2.0-1.noarch.rpm root@cli01.example.com:.
```

3. クライアントに切り替え、RPM をインストールします。

```
[root@cli01 ~]# yum install containertest-2.0-1.noarch.rpm
```

8.6. クライアントでの PODMAN PULL コマンドのテスト

podman pull コマンドを使用して、コンテナのコンテンツを確認できます。

手順

1. **podman pull** コマンドを実行します。

```
[root@cli01 ~]# podman pull jboss-eap-6_eap64-openshift

Resolving "jboss-eap-6_eap64-openshift" using unqualified-search registries
(/etc/containers/registries.conf)
Trying to pull cds.example.com/jboss-eap-6_eap64-openshift:latest...
Getting image source signatures
Copying blob b0e0b761a531 done
Copying blob aa23ac04e287 done
Copying blob 0d30ea1353f9 done
Copying config 3d0728c907 done
Writing manifest to image destination
Storing signatures
3d0728c907d55d9faedc4d19de003f21e2a1ebdf3533b3d670a4e2f77c6b35d2
```

2. **podman pull** コマンドが失敗した場合は、**rhui-manager** のステータスを確認します。同期がまだ実行されていない可能性があり、同期するまで待機する必要があります。

```
Resolving "jboss-eap-6_eap64-openshift" using unqualified-search registries
(/etc/containers/registries.conf)
Trying to pull cds.example.com/jboss-eap-6_eap64-openshift:latest...
Error: initializing source docker://cds.example.com/jboss-eap-6_eap64-
openshift:latest: reading manifest latest in cds.example.com/jboss-eap-6_eap64-
openshift: manifest unknown: Manifest not found.
```

第9章 設定ファイル、終了コード、およびログファイル

以下の設定ファイル、RHUI Manager 終了コード、およびログファイルが Red Hat Update Infrastructure 4 で使用されます。

設定ファイル

表9.1 設定ファイル

コンポーネント	ファイルまたはディレクトリー	使用方法
Red Hat Update Appliance	<code>/etc/pulp/*</code>	Pulp 設定ファイル
	<code>/etc/rhui/rhui-tools.conf</code>	rhui-manager 設定ファイル
	<code>/etc/pki/rhui/*</code>	Red Hat Update Infrastructure の証明書
	<code>/root/.rhui/answers.yaml</code>	RHUA 設定での使用
	<code>/etc/rhui/rhui-subscription-sync.conf</code>	サブスクリプション同期スクリプトの設定
コンテンツ配信サーバー	<code>/etc/pki/rhui/certs/</code>	CDS の証明書
HAProxy	<code>/etc/haproxy/haproxy.cfg</code>	HAProxy 設定ファイル

RHUI Manager 終了コード

RHUI Manager は、以下のコードを使用して、`rhui-manager status` コマンドと `rhui-manager CLI` コマンドを実行した結果を示します。

表9.2 RHUI Manager 終了コード

ステータスコード	説明
0	成功
1	一般エラーまたはリポジトリ同期エラー
2	CDS での SSL 証明書エラー
32	エンタイトルメント CA 証明書または SSL 証明書の有効期限に関する警告
64	エンタイトルメント CA 証明書または SSL 証明書の有効期限エラー
128	1つ以上の RHUI サービスが RHUA、CDS、または HAProxy ノードで実行されていません。

ステータスコード	説明
239	リポジトリが存在しないため、削除できませんでした。
240	必要なリソースに問題がありました。たとえば、有効なリポジトリが見つからなかったため、クライアント設定 RPM を構築できませんでした。
241	<p>不明なリポジトリが指定されたため、同期タスクをスケジュールできませんでした。</p> <p>トラブルシューティングするには、以下を実行します。</p> <p>* スペルを確認してください * 最初にリポジトリを追加してください * Pulp の問題のログを確認してください</p>
242	Pulp の問題のため、カスタムリポジトリを作成できませんでした。詳細は、メッセージとログを確認してください。
243	Red Hat リポジトリの一部はすでに RHUI に存在し、一部はエンタイトルメントで使用できないため、Red Hat リポジトリを追加できませんでした。
244	カスタムリポジトリは RHUI にすでに存在するため、作成できませんでした。
245	Red Hat リポジトリは RHUI にすでに存在しているため、追加できませんでした。
246	Red Hat リポジトリはエンタイトルメントで使用できないため、追加できませんでした。スペルを確認するか、コマンド <code>rm -f /var/cache/rhui/*</code> を使用してリポジトリマッピングキャッシュを削除し、再試行してください。
247	Pulp の問題のため、Red Hat リポジトリを追加できませんでした。詳細は、メッセージとログを確認してください。
248	1つ以上の Red Hat リポジトリが RHUI 4 にすでに存在しているため、RHUI 3 から RHUI 4 への移行は停止されました。リポジトリを削除するか、 <code>--force</code> フラグを使用する必要があります。
249	RHUI 設定 <code>/etc/rhui/rhui-tools.conf</code> が無効です。詳細は、メッセージを確認してください。
250	エンタイトルメント証明書は、書き込み可能ではありません。
251	エンタイトルメント証明書の有効期限が切れています。
252	RHUI リポジトリが含まれていないため、エンタイトルメント証明書は無効です。

ステータスコード	説明
253	エンタイトルメント証明書ファイルは、有効な証明書ではありません。
254	コマンドラインエラー:ネットワークの問題により、RHUI CLI を実行できませんでした。
255	引数エラー:必須の引数が指定されていませんでした。

ログファイル

表9.3 ログファイル

コンポーネント	ファイルまたはディレクトリー	使用方法
Red Hat Update Appliance	/root/.rhui/rhui.log* (rhui-manager が root として実行されていることを前提)	Red Hat Update Infrastructure Management Tool のログ
	/var/log/messages	現在書き込み中のファイル以外に、4つの最新バージョンの保持
	/var/log/nginx/*	NGINX のログ
	/var/log/messages	Qpid ログ
	/var/log/rhui-subscription-sync.log	サブスクリプションの同期ログ
コンテンツ配信サーバー	/var/log/nginx/*	NGINX のログ
	/nginx/ssl-access.log	ロギングの例: Successful yum repolist : CDS ノードがこのクライアントのアクティビティーについてログに記録する
クライアント	RHEL7 以前のバージョンでは、 /var/log/yum.log	yum コマンドログ
クライアント	RHEL 8 以降のバージョンの /var/log/dnf.log	dnf コマンドログ
	/var/log/messages	クライアント syslog
	yum/dnf コマンドのコンソール出力	

第10章 RHUI 4 コマンドの使用

RHUpdate からアクセスできる Red Hat Update Infrastructure Management Tool 内の画面には、さまざまな Red Hat Update Infrastructure 4 コンポーネントを設定および更新できるメニューオプションがあります。

10.1. RHUI 4 メニューオプションの使用

Red Hat Update Infrastructure Management Tool は、RHUI 4 を設定管理するための対話型コマンドを提供します。

表10.1 Red Hat Update Infrastructure Management Tool のメニューおよびコマンド

スクリーン	Screen コマンド	メニューオプション	メニューオプション コマンド
manage repositories	r		
Repository Management		list repositories currently managed by the RHUI	l
		display detailed information on a repository	i
		add a new Red Hat content repository	a
		create a new custom repository (RPM content only)	c
		delete a repository from the RHUI	d
		upload content to a custom repository (RPM content only)	u
		upload content from a remote web site (RPM content only)	ur
		list packages in a repository (RPM content only)	p
manage content delivery servers (CDS)	c		

スクリーン	Screen コマンド	メニューオプション	メニューオプション コマンド
Content Delivery Server (CDS) Management		list all known CDS instances managed by the RHUI	l
		register (add) a new CDS instance	a
		reinstall and reapply configuration to an existing CDS instance	r
		unregister (delete) a CDS instance from the RHUI	d
manage HAProxy load-balancer instances	l		
Load-balancer (HAProxy) Management		list all known HAProxy Load-balancer instances managed by the RHUI	l
		register (add) a new HAProxy Load-balancer instance	a
		reinstall and reapply configuration to an existing HAProxy Load-balancer instance	r
		unregister (delete) a HAProxy Load-balancer instance from the RHUI	d
synchronization status and scheduling	s		
Synchronization Status		display repo sync summary	dr
		display running syncs	rr
		view the details of the last repository sync	vr

スクリーン	Screen コマンド	メニューオプション	メニューオプション コマンド
		sync an individual repository immediately	sr
		cancel active sync tasks	ca
		cancel waiting sync tasks	cw
		view/change repository auto-publish status	ap
		view/advance repository workflow	wf
		export repo to filesystem	ex
create entitlement certificates and client configuration RPMs	e		
Client Entitlement Management		generate an entitlement certificate	e
		create a client configuration RPM from an entitlement certificate	c
manage Red Hat entitlement certificates	n		
Entitlements Manager		list Red Hat content certificate entitlements	l
		list custom repository entitlements	c
		upload a new or updated Red Hat content certificate	u
manage RHUI users	u		
User Manager		change admin's password (followed by logout)	p

10.2. RHUI 4 CLI オプションの使用

Red Hat Update Infrastructure 4 の管理タスクの大半は、インストール中に行われます。インストール後は、Red Hat CDN から定期的に更新パッケージを取得し、クライアントでそのパッケージを自動的に利用できるようにします。

Red Hat Update Infrastructure Management Tool (**rhui-manager** で実行する) と呼ばれるコマンドラインインターフェイスは、インストールを容易にします。このツールは、RHUI の各コンポーネントに必要な設定要素の対話式プロンプトを提供します。RHUA、CDS、およびロードバランサー。このツールは、Red Hat が提供するコンテンツ証明書を取得して、Red Hat CDN に接続するときに使用し、クライアントが RHUI に接続するために使用する内部のクラウド固有の証明書を生成する手段も提供します。Red Hat Update Infrastructure Management Tool を使用すると、クラウドプロバイダーはクライアント RHEL インスタンスにインストールされるクライアント設定バンドルを生成することができます。このバンドルにより、クライアントは RHUI インストールから更新を取得できるようになります。

Red Hat Update Infrastructure Management Tool は対話型シェルを使用しますが、一部の機能はシェルプロンプトからも実行できます。Red Hat Update Infrastructure Management Tool は、主に 7 つのコマンドを使用します。各コマンドのサブコマンドでは、サブコマンドが **-h** および **--help** 以外のオプションを想定している場合は、オプションのリストが提供されます。

すべてのオプションおよびコマンドを表示します。

```
[root@ip-10-141-150-145 ~]# rhui-manager --help
Usage: rhui-manager [options]

OPTIONS
  -h/--help  show this help message and exit
  --debug    enables debug logging
  --noninteractive prevents console input, used for scripting
  --config   absolute path to the configuration file; defaults to /etc/rhui/rhui-tools.conf
  --server   location of the RHUA server (overrides the config file)
  --username if specified, previously saved authentication credentials are ignored and this username
is used to login
  --password used in conjunction with --username
  --logout  logout from the active session

COMMANDS
  cert      : Red Hat content certificate management
  packages  : package manipulation on repositories
  repo      : repository listing and manipulation
  cds       : CDS listing and manipulation
  migrate   : Migrate from {RHUI3}
  haproxy   : Load balancer listing and manipulation
  status    : RHUI status and health information
  client    : Red Hat client management
```

10.2.1. cert

```
Red Hat content certificate management
  upload   : uploads a new content certificate
  info     : display information about the current content certificate
```

```
# rhui-manager cert upload
upload: uploads a new content certificate
```

--cert - full path to the new content certificate (required)
 --key - full path to the new content certificate's key

10.2.2. packages

package manipulation on repositories

list : lists all packages in a repository
 remove : removes a package from a custom repository
 upload : uploads a package or directory of packages to a custom repository
 remote : uploads RPM content from a remote URL to a custom repository

list: lists all packages in a repository

--repo_id - id of the repository to list packages for (required)

remove: removes a package from a custom repository

--repo_id - id of the custom repository to remove a package from (required)
 --package - name of the package to be removed (required)
 --vr - if specified, only the supplied version-release of the package will be removed
 --force - don't ask for confirmation

upload: uploads a package or directory of packages to a custom repository

--repo_id - id of the custom repository where the packages will be uploaded (required)
 --packages - path to an .rpm file or directory of RPMs that will be uploaded (required)

remote: uploads RPM content from a remote URL to a custom repository

--repo_id - id of the custom repository where the packages will be uploaded (required)
 --url - remote URL of the package or a web page that will be scraped for RPM content (required)

10.2.3. repo

repository listing and manipulation

list : lists all repositories in the RHUI
 info : displays information on an individual repo
 add : add a Red Hat repository to the RHUA
 add_by_repo: add Red Hat repositories to the RHUA via repo ID
 add_by_file: add Red Hat repositories to the RHUA using an input file
 add_errata: associate errata metadata with a repository
 delete : delete a repository
 sync : sync a repository
 export : export a repository to the filesystem
 enable_sync: enable scheduled synchronization of a repository
 disable_sync: disable scheduled synchronization of a repository
 sync_all : sync all repositories
 metadata : ensure metadata is generated for the latest version of repositories
 enable_autopublish: enable automatic publishing of a new repository version
 disable_autopublish: disable automatic publishing of a new repository version
 create_custom: create a custom repository
 unused : list of products available but not synced to the RHUA

info: displays information on an individual repo

--repo_id - identifies the repository to display (required)

add: add a Red Hat repository to the RHUA
--product_name - product to add the RHUA (required)

add_by_repo: add Red Hat repositories to the RHUA via repo ID
--repo_ids - repo IDs to add, comma-separated (required)
--sync-now - Use to sync any repos that are added (optional)

add_by_file: add Red Hat repositories to the RHUA using an input file
--file - file containing repo IDs to add, one per line (required)
--sync_now - Use to sync any repos that are added (optional)

add_errata: associate errata metadata with a repository
--repo_id - repo ID to associate the metadata with (required)
--updateinfo - updateinfo file to be applied (required)

delete: delete a repository
--repo_id - identifies the repository to delete (required)

sync: sync a repository
--repo_id - identifies the repository to sync (required)

export: export a repository to the filesystem
--repo_id - identifies the repository to export (required)

metadata : ensure metadata is generated for the latest version of repositories
--repo_id - explicit repo ID to generate metadata for

enable_sync: enable scheduled synchronization of a repository
--repo_id - identifies the repository to enable scheduled synchronization for (required)
--verbose - if present, info on last/next synchronization tasks will be displayed

disable_sync: disable scheduled synchronization of a repository
--repo_id - identifies the repository to disable scheduled synchronization for (required)
--verbose - if present, info on last/next synchronization tasks will be displayed

enable_autopublish: enable automatic publishing of a new repository version
--repo_id - identifies the repository to enable automatic publishing for (required)

disable_autopublish: disable automatic publishing of a new repository version
--repo_id - identifies the repository to disable automatic publishing for (required)

create_custom: create a custom repository
--repo_id - identifies the repository to add (required)
--path - path to the content being served by CDS; defaults to repo_id
--display_name - display name for the custom repository
--entitlement - path used in the entitlement certificate; may use yum variable substitutions
--legacy_md - if present, the repo will use SHA1, otherwise default value is used (SHA256)
--redhat_content - repository will host Red Hat GPG signed content

--protected - make the content protected by entitlement certificate
 --gpg_public_keys - comma separated list of public keys used to sign the served content; the filenames must not contain comma

unused: list all unused Red Hat repositories
 Loading latest entitled products from Red Hat...
 ... listings loaded
 Available Repositories

10.2.4. cds

CDS listing and manipulation

list : lists all cds instances in the RHUI
 add : register a cds instance to the RHUI
 reinstall : reinstalls an already registered cds instance
 delete : unregisters cds instances from the RHUI

add: register a cds instance to the RHUI

--hostname - The hostname of the instance to add. (required)
 --ssh_user - Username with SSH access to the instance and sudo privileges. (required)
 --keyfile_path - Absolute path to an SSH private key to use with the given user. (required)
 --hostfile - Absolute path to a known_hosts file to use to determine the identity of the instance; if this is not provided and the instance hostkey is not in the system-wide known_hosts file, this command will fail.
 --user_supplied_ssl_key - Optional absolute path to the user supplied SSL key file.
 --user_supplied_ssl_cert - Optional absolute path to the user supplied SSL crt file.
 --force - Add the system even if the hostname is already registered.
 --unsafe - Proceed even if the instance host key is not in the known_hosts file. This is not secure!

reinstall: reinstalls an already registered cds instance

--hostname - The hostname of the instance to reinstall on; this instance must be registered already.
 --all - Reinstall all the registered instances.

delete: unregisters cds instances from the RHUI

--force - Delete the system, even if it is the last of its kind.
 --hostnames - Comma-separated list of hostnames to delete (unregister) from RHUI. (required)

10.2.5. migrate

Migrate from {RHUI3}

--hostname : the {RHUI3} hostname
 --password : <your_password>
 --keyfile_path : ~/.ssh/id_rsa_rhua*

10.2.6. haproxy

Load balancer listing and manipulation

list : lists all haproxy instances in the RHUI
 add : register a haproxy instance to the RHUI

reinstall : reinstalls an already registered haproxy instance
 delete : unregisters haproxy instances from the RHUI

add: register a haproxy instance to the RHUI

--hostname - The hostname of the instance to add. (required)
 --ssh_user - Username with SSH access to the instance and sudo privileges. (required)
 --keyfile_path - Absolute path to an SSH private key to use with the given user. (required)
 --hostfile - Absolute path to a known_hosts file to use to determine the identity of the instance; if this is not provided and the instance hostkey is not in the system-wide known_hosts file, this command will fail.
 --config - Optional absolute path to a user supplied HAProxy config file.
 --force - Add the system even if the hostname is already registered.
 --unsafe - Proceed even if the instance host key is not in the known_hosts file. This is not secure!

reinstall: reinstalls an already registered haproxy instance

--hostname - The hostname of the instance to reinstall on; this instance must be registered already.
 --all - Reinstall all the registered instances.

delete: unregisters haproxy instances from the RHUI

--force - Delete the system, even if it is the last of its kind.
 --hostnames - Comma-separated list of hostnames to delete (unregister) from RHUI. (required)

10.2.7. status

status: RHUI status and health information

--code - if specified, only a numeric code for the result will be displayed
 --repo_json - Name of the JSON file for a repo status

10.2.8. client

Red Hat client management

labels : list the labels required for client certificate creation
 cert : create a content certificate for a rhui client
 rpm : create a client config rpm
 content_source: create an alternate source config rpm
 acs_config: output a JSON representation of the alternate source config

cert: create a content certificate for a rhui client

--repo_label - identifies the repositories to add. Comma delimited string of repo labels (required)
 --name - identifies the certificate name (required)
 --days - number of days cert will be valid (required)
 --dir - directory where the certificate will be stored (required)

rpm: create a client config rpm

--private_key - entitlement private key
 --entitlement_cert - entitlement certificate
 --rpm_version - version number of the client config rpm
 --rpm_release - release of rpm package. Default is 1
 --rpm_name - name of the client config rpm (required)
 --dir - directory where the rpm will be created (required)
 --unprotected_repos - comma-separated list of unprotected repos to include
 --cert - generate certificate also before building client config rpm if given

- ca_cert - full path to the certificate authority of CDS servers
- repo_label - identifies the repositories to add. Comma delimited string of repo labels
- name - identifies the certificate name if it is different from rpm name
- days - number of days cert will be valid
- proxy - url/string in case proxy option is necessary in yum repo file

content_source: create an alternate source config rpm

- private_key - entitlement private key
- entitlement_cert - entitlement certificate
- rpm_version - version number of the client config rpm
- rpm_name - name of the client config rpm (required)
- dir - directory where the rpm will be created (required)
- unprotected_repos - comma-separated list of unprotected repos to include
- cert - generate certificate also before building client config rpm if given
- ca_cert - full path to the certificate authority of CDS servers
- repo_label - identifies the repositories to add. Comma delimited string of repo labels
- name - identifies the certificate name if it is different from rpm name
- days - number of days cert will be valid

acs_config: output a JSON representation of the alternate source config

- dir - directory where the JSON representation will be stored (required)
- private_key - entitlement private key
- entitlement_cert - entitlement certificate
- cert - generate certificate based on the the repos supplied via --repo_label
- ssl_ca_cert - full path to the certificate authority of CDS servers (defaults to ssl_ca_cert specified in rhui-tools.conf)
- repo_label - identifies the repositories to add. Comma delimited string of repo labels
- days - number of days cert will be valid if new cert is generated

第11章 認定クラウドおよびサービスプロバイダー認定のワークフロー

認定クラウドプロバイダー契約では、作成したテナントインスタンスのイメージ(テンプレート)を Red Hat が認定し、エンドカスタマーが完全にサポートされる設定を維持する必要があります。

Red Hat Enterprise Linux 用のイメージを認証するには2つの方法があります。推奨される方法は、Certified Cloud and Service Provider (CCSP) イメージ認定ワークフローを使用することです。

認定が Red Hat によって確認されると、合格/不合格が割り当てられ、この認定内容について [Red Hat Ecosystem Catalog](#) のパブリックの Red Hat 認定 Web サイトに掲載されます。

11.1. 関連情報

- [Red Hat Certified Cloud and Service Provider Certification Workflow Guide](#)
- [Product Documentation for Red Hat Certified Cloud and Service Provider Certification 7.34](#)

第12章 RED HAT UPDATE INFRASTRUCTURE のバックアップと復元

Red Hat Update Infrastructure (RHUI) サーバーをインストールして設定したら、それらをバックアップすることを推奨します。RHUI のバックアップは、問題が発生した場合や RHUI を正しく設定していない場合に役立ちます。このような場合、RHUI を復元することで以前の作業設定に戻すことができます。

RHUI を正常にバックアップするには、すべての RHUA、CDS、および HAProxy ノードをバックアップする必要があります。

12.1. RED HAT UPDATE APPLIANCE のバックアップ

Red Hat Update Appliance をバックアップするには、関連するすべてのファイルとストレージをバックアップする必要があります。



注記

RHUA をバックアップするには、関連するサービスを停止する必要があります。ただし、クライアントはコンテンツ配信サーバー (CDS) にのみ接続されているため、サービスを停止しても、クライアントインスタンスがパッケージの更新またはインストールを無効にすることはありません。したがって、自動モニタリングソリューションを導入している場合、バックアッププロセス中にモニタリングが失敗する可能性があります。

手順

1. **pulp-server** サービスを停止します。

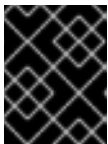
```
# systemctl stop pulpcore-api pulpcore-content pulpcore-worker*
```

2. サービスが停止したかどうかを確認します。

```
# systemctl status pulpcore-api pulpcore-content pulpcore-worker*
```

3. 以下のファイルをバックアップします。

```
# cp -a <source_files_path> <destination_files_path>
```



重要

ファイルをバックアップするときに、ファイルが現在の属性を保持していることを確認してください。

ファイルのリスト:

- /etc/pki/rhui/*
- /etc/pulp/*
- /etc/rhui/*
- /etc/rhui/rhui-tools.conf

- /etc/nginx/*
- /root/.rhui/*
- /var/log/rhui/*
- /var/log/rhui-subscription-sync.log*
- オプション: /var/lib/rhui/*



注記

このディレクトリーをバックアップすると、ダウンロードされたすべてのコンテンツがバックアップされるため、データ量が多くなる可能性があります。

4. 生成されたクライアントエンタイトルメント証明書とクライアント設定 RPM をバックアップします。
5. RHUI サービスを再起動します。

```
# rhui-services-restart
```

12.2. RED HAT UPDATE APPLIANCE の復元

RHUA を復元するには、新しい RHUA ノードを作成し、関連するファイルをバックアップしたバージョンに置き換える必要があります。

手順

1. 新しい RHUA ノードを作成します。詳細は、[RHUA ノードの設定](#) を参照してください。
2. **pulp-server** サービスを停止します。

```
# systemctl stop pulpcore-api pulpcore-content pulpcore-worker*
```

3. サービスが停止したかどうかを確認します。

```
# systemctl status pulpcore-api pulpcore-content pulpcore-worker*
```

4. 以下のファイルを復元します。

```
# cp -a <source_files_path> <destination_files_path>
```



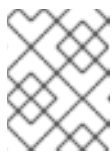
重要

ファイルを復元するときに、ファイルが現在の属性を保持していることを確認してください。

ファイルのリスト:

- /etc/pki/rhui/*

- /etc/pulp/*
- /etc/rhui/*
- /etc/rhui/rhui-tools.conf
- /etc/nginx/*
- /root/.rhui/*
- /var/log/rhui/*
- /var/log/rhui-subscription-sync.log*
- オプション: /var/lib/rhui/*



注記

このディレクトリーを復元すると、ダウンロードされたすべてのコンテンツが復元されるため、データ量が多くなる可能性があります。

5. 生成されたクライアントエンタイトルメント証明書とクライアント設定 RPM を復元します。
6. RHUI サービスを再起動します。

```
# rhui-services-restart
```

12.3. コンテンツ配信サーバーのバックアップ

CDS をバックアップするには、関連するすべてのファイルとストレージをバックアップする必要があります。



注記

サービスが完全に失われないようにするために、一度に1つの CDS ノードをバックアップしてください。クライアントは、実行中の他の CDS ノードに自動的に切り替わります。

手順

1. **nginx** サービスを停止します。

```
# systemctl stop nginx
```

2. **nginx** サービスが停止していることを確認します。

```
# systemctl status nginx
```

3. 以下のファイルをバックアップします。

```
# cp -a <source_files_path> <destination_files_path>
```

**重要**

ファイルをバックアップするときに、ファイルが現在の属性を保持していることを確認してください。

ファイルのリスト:

- /etc/nginx/*
 - /var/log/nginx/*
 - /etc/pki/rhui/*
4. RHUI サービスを再起動します。

```
# rhui-services-restart
```

12.4. コンテンツ配信サーバーの復元

コンテンツ配信サーバーを復元するには、新しい CDS ノードを作成し、関連するファイルをバックアップしたバージョンに置き換える必要があります。

手順

1. 新しい CDS ノードを作成します。詳細は、[CDS ノードの設定](#) を参照してください。
2. **nginx** サービスを停止します。

```
# systemctl stop nginx
```

3. **nginx** サービスが停止していることを確認します。

```
# systemctl status nginx
```

4. 以下のファイルを復元します。

```
# cp -a <source_files_path> <destination_files_path>
```

**重要**

ファイルを復元するときに、ファイルが現在の属性を保持していることを確認してください。

ファイルのリスト:

- /etc/nginx/*
 - /var/log/nginx/*
 - /etc/pki/rhui/*
5. RHUI サービスを再起動します。

```
# rhui-services-restart
```

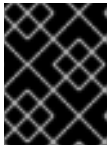
12.5. HAPROXY サーバーのバックアップ

HAProxy サーバーをバックアップするには、関連するすべてのファイルとストレージをバックアップする必要があります。

手順

1. `/etc/haproxy/haproxy.cfg` ファイルをバックアップします。

```
# cp -a <source_files_path> <destination_files_path>
```



重要

ファイルをバックアップするときに、ファイルが現在の属性を保持していることを確認してください。

12.6. HAPROXY サーバーの復元

HAProxy サーバーを復元するには、新しい HAProxy ノードを作成し、関連するファイルをバックアップしたバージョンに置き換える必要があります。

手順

1. 新しい HAProxy ノードを作成します。詳細は、[HAProxy ノードの設定](#) を参照してください。
2. `/etc/haproxy/haproxy.cfg` ファイルを復元します。

```
# cp -a <source_files_path> <destination_files_path>
```



重要

ファイルを復元するときに、ファイルが現在の属性を保持していることを確認してください。

第13章 RHUI 4 の一般的な問題を解決

以下の表は、Red Hat Update Infrastructure の既知の問題を示しています。これらの問題が発生した場合は、Bugzilla を通じて問題を報告してください。

表13.1 Red Hat Update Infrastructure の一般的な問題

イベント	既知の問題の説明	推奨事項
インストールおよび設定	RHUA と CDS の間でコミュニケーションの問題が発生しています。	<p>RHUA および CDS に完全修飾ドメイン名 (FQDN) が設定されており、解決可能であることを確認します。</p> <p>Bug 726420 - Quick note on proxy URL で説明されているように HTTP プロキシを適切に設定します。</p>
同期	リポジトリを Red Hat と同期することができません。	<p>RHUI SKU がアカウントにあることを確認します。</p> <p>適切なコンテンツ証明書が RHUA に読み込まれていることを確認します。</p> <p>一時的な CDN の問題を検索します。</p> <p>環境内の HTTP プロキシを確認し、While syncing repositories to RHUA, it fails with "RepoError: Cannot retrieve repository metadata (repomd.xml) for repository. Please verify its path and try again" when proxy used エラーが発生しないことを確認してください。</p> <p>RHUA が CDS に同期できません。これは、qpuid 証明書の有効期限が切れていることが原因です。詳細は、CDS sync fails with error "ssl3 alert certificate expired" due to expired qpuid CA certificates on RHUI を参照してください。</p>

イベント	既知の問題の説明	推奨事項
<p>Red Hat Update Appliance/コンテンツ配信ネットワーク通信</p>	<p>Red Hat Update Appliance がコンテンツ配信ネットワークと通信していません。</p>	<p>/etc/pki/rhui/redhat (.pem ファイル) のコンテンツ証明書を使用して RHUA と CDN との間の接続とアクセスをテストします。</p> <pre># cd /etc/pki/rhui/redhatwget --certificate=8a85f98146a087b80146afacb3362499.pem --ca-certificate=/etc/rhsm/ca/redhat-uep.pem</pre> <p>https://cdn.redhat.com/content/dist/rhel/rhui/server/6/6Server/x86_64/os/repodata/repomd.xml</p> <p>curl (1) の man ページからの引用です。NSS PEM PKCS#11 モジュール (lib-nsspem.so) が利用可能な場合は、PEM ファイルが読み込まれることがあります。現在のディレクトリーからファイルを使用する場合は、ニックネームとの混同を避けるために先頭に接頭辞 ./ を付けます。</p> <p>各 CDS では、/etc/pki/pulp/content のエンタイトルメント証明書を使用して、# curl --cert ./rhui-ec2-20120619.pem を使用して RHUA コンテンツの可用性をテストすることができます。</p> <p>RHUA でホストされるリポジトリーの URL は常に https://fqdn/pulp/content で始まります。次の方法で残りの URL は除外できます。/var/lib/rhui/remote_share/symlinks/pulp/content/ の RHUA のファイルパスを確認します。1.6 で終わる OID にはパスが含まれているため、openssl コマンドを使用してコンテンツ証明書を直接確認します。</p>

イベント	既知の問題の説明	推奨事項
クライアントとコンテンツ配信サーバーの通信	<p>curl を使用して、コンテンツ配信サーバーノードとのクライアント通信を検証することもできます。</p>	<pre># curl --cert /etc/pki/entitlement/product/content.crt --key /etc/pki/entitlement/key.pem https://ip-10-4-58-34.ec2.internal/pulp/repos/content/dist/rhel/rhui/server/6/6Server/x86_64 /rhui/2.1/os/repodata/repomd.xml -k <?xml version="1.0" encoding="UTF-8"?> <repomd xmlns="http://linux.duke.edu/metadata/repo" xmlns:rpm="http://linux.duke.edu/metadata/rpm"> <revision>1339940325</revision> <data type="other_db"> <location href="repodata/4f86b0ae203bba90d22a8363120c66ed6f37da81- other.sqlite.bz2"/> <checksum type="sha">4f86b0ae203bba90d22a8363120c66ed6f37da81</checksum> <timestamp>1339940328.43</timestamp></pre>
クライアント/HAProxy 通信	<p>HAProxy ノードがすべてダウンしています。クライアントが RHUI リポジトリへのアクセスを失いました。</p>	<p>新規の HAProxy ノードを追加および設定します。何らかの理由でこれを実行できない場合は、DNS 設定を一時的に変更して、主なロードバランサーのホスト名 (本ガイドの <code>cds.example.com</code>) がいずれかの CDS ノードの IP アドレスに解決するようにします。これにより、クライアントは利用できない HAProxy ノードを回避し、CDS と直接通信できます。</p>