



Red Hat OpenStack Platform 17.1

OpenStack Dashboard を使用したクラウドリソースの管理

OpenStack Dashboard GUI の表示および設定

Red Hat OpenStack Platform 17.1 OpenStack Dashboard を使用したクラウドリソースの管理

OpenStack Dashboard GUI の表示および設定

OpenStack Team
rhos-docs@redhat.com

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

このガイドでは、Red Hat OpenStack Platform Dashboard のユーザーインターフェイスで利用可能なオプションについて説明します。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 RED HAT OPENSTACK PLATFORM DASHBOARD サービス (HORIZON)	5
1.1. ADMIN タブ	5
1.2. PROJECT タブ	6
1.3. IDENTITY タブ	8
第2章 DASHBOARD のカスタマイズ	9
2.1. HORIZON コンテナイメージの取得	9
2.2. RCUE テーマの取得	9
2.3. RCUE をベースにした自分専用テーマの作成	10
2.4. 自分専用テーマを有効にして DASHBOARD をカスタマイズするためのファイルの作成	10
2.5. 変更した HORIZON イメージの生成	11
2.6. 変更したコンテナイメージのオーバークラウドでの使用	12
2.7. PUPPET パラメーターの編集	12
2.8. SHARED FILE SYSTEMS サービスの共有プロトコルのカスタマイズ	13
2.9. DASHBOARD をカスタマイズしたオーバークラウドのデプロイ	14

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。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Jira でドキュメントのフィードバックを提供する

ドキュメントに関するフィードバックを提供するには、[Create Issue](#) フォームを使用します。Red Hat OpenStack Platform Jira プロジェクトで Jira Issue が作成され、フィードバックの進行状況を追跡できます。

1. Jira にログインしていることを確認してください。Jira アカウントをお持ちでない場合は、アカウントを作成してフィードバックを送信してください。
2. [Create Issue](#) をクリックして、**Create Issue** ページを開きます。
3. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
4. **Create** をクリックします。

第1章 RED HAT OPENSTACK PLATFORM DASHBOARD サービス (HORIZON)

Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) は、RHOSP サービスの管理に使用できる Web ベースのグラフィカルユーザーインターフェイスです。

ブラウザで Dashboard にアクセスするには、Dashboard Service がインストール済みで、かつその Dashboard のホスト名または IP アドレス、およびログインパスワードが必要です。Dashboard の URL は、以下のような形式です。

`http://HOSTNAME/dashboard/`

1.1. ADMIN タブ

Admin タブでは、使用状況の確認や、インスタンス、ボリューム、フレーバー、イメージ、プロジェクト、ユーザー、サービス、クォータを管理できます。



注記

Admin タブは、管理ユーザーとしてログインしている場合にメインウィンドウに表示されます。

以下のオプションは Admin タブから利用できます。

表1.1 システムパネル

パラメーター名	説明
Overview	基本的なレポートを表示します。
Resource Usage	以下のタブを使用して、それぞれの使用状況を表示します。 <ul style="list-style-type: none"> ● Usage Report: 使用状況のレポートを表示します。 ● Stats: 全リソースの統計情報を表示します。
Hypervisors	ハイパーバイザーの概要を表示します。
Host Aggregates	ホストアグリゲートを表示、作成、編集します。アベイラビリティゾーンのリストを表示します。
Instances	全プロジェクトではなく、一部のプロジェクトのユーザーに属する実行中のインスタンスを表示、一時停止、再開、休止、移行、ソフトリブート/ハードリブート、削除します。また、インスタンスのログを表示したり、コンソールを使用してインスタンスにアクセスしたりします。
Volumes	ボリュームおよびボリュームタイプを表示、作成、編集、削除します。

パラメーター名	説明
Flavors	フレーバーを表示、作成、編集、削除したり、その追加スペックを表示したりします。フレーバーとは、Red Hat OpenStack Platform (RHOSP) 内の仮想ハードウェアのテンプレートです。
Images	カスタムイメージを表示、作成、削除し、イメージのプロパティを編集します。
Networks	ネットワークを表示、作成、削除し、ネットワークのプロパティを編集します。
Routers	ルーターを表示、作成、削除し、ルーターのプロパティを編集します。
Floating IP	すべてのプロジェクトについて、割り当てられた Floating IP アドレスを表示します。
Defaults	環境内のリソースのデフォルトクォータ (上限) を表示、編集します。
Metadata Definitions	メタデータ定義の名前空間をインポート、表示、編集して、メタデータ定義を特定のリソースタイプと関連付けます。
System Information	次のタブがあります。 <ul style="list-style-type: none"> ● Services: サービス一覧を表示します。 ● Compute Services: すべてのコンピュートサービスを一覧表示します。 ● Network Agents: ネットワークエージェントを表示します。 ● Block Storage Services: すべての Block Storage サービスを一覧表示します。 ● Orchestration Services: すべての Orchestration サービスを一覧表示します。

1.1.1. 割り当てられた Floating IP アドレスの表示

Floating IP パネルを使用して、割り当てられた Floating IP アドレスの一覧を表示できます。**nova list --all-projects** コマンドを使用して、コマンドラインから同じ情報にアクセスできます。

1.2. PROJECT タブ

Project タブでは、プロジェクトリソースを表示および管理できます。**Identity > Projects** でプロジェクトを有効な状態に設定し、対象プロジェクトのリソースを表示、管理します。

以下のオプションは **Project** タブから利用できます。

表1.2 Compute タブ

パラメーター名	説明
Overview	プロジェクトのレポートを表示します。
Instances	インスタンスの表示、起動、停止、一時停止、リブートや、インスタンスからのスナップショットの作成、コンソールからインスタンスへの接続を実行します。
Volumes	以下のタブを使用して、以下に示すタスクを完了します。 <ul style="list-style-type: none"> ● Volumes: ボリュームを表示、作成、編集、削除します。 ● Volume Snapshots: ボリュームのスナップショットを表示、作成、編集、削除します。
Images	プロジェクトユーザーが作成したイメージ、インスタンスのスナップショット、ボリュームのスナップショット、および公開されているイメージを表示します。イメージを作成、編集、削除したり、イメージやスナップショットからインスタンスを起動します。
Access & Security	以下のタブを使用して、以下に示すタスクを完了します。 <ul style="list-style-type: none"> ● Security Groups: セキュリティーグループおよびセキュリティグループのルールを表示、作成、編集、削除します。 ● Key Pairs: キーペアを表示、作成、編集、インポート、削除します。 ● Floating IP: IP アドレスをプロジェクトに割り当てたり、プロジェクトから割り当てを解除したりします。 ● API Access: API エンドポイントの表示、OpenStack RC ファイルのダウンロード、EC2 認証情報のダウンロード、現在のプロジェクトユーザーの認証情報の表示を行います。

表1.3 Network タブ

パラメーター名	説明
Network Topology	対話型のネットワークトポロジーを表示します。
Networks	パブリックネットワークおよびプライベートネットワークとサブネットを作成、管理します。
Routers	ルーターを作成、管理します。

パラメーター名	説明
Trunks	トランクを作成、管理します。OpenStack Networking (neutron) で trunk 拡張機能を有効にする必要があります。

表1.4 Object Store タブ

パラメーター名	説明
Containers	ストレージコンテナを作成および管理します。コンテナとは、データ用のストレージコンパートメントで、データの整理に使用できます。Linux のファイルディレクトリーと概念が似ていますが、ネストできません。

表1.5 Orchestration タブ

パラメーター名	説明
Stacks	OpenStack ネイティブの REST API と CloudFormation 互換の Query API の両方を介し、テンプレートを使用して、複数の複合クラウドプラットフォームをオーケストレーションします。

1.3. IDENTITY タブ

Identity タブでは、プロジェクトおよびユーザーを表示、管理できます。

以下のオプションは Identity タブから利用できます。

- **Projects:** プロジェクトの表示、作成、編集、削除、プロジェクト使用状況の表示、プロジェクトメンバーとしてのユーザーの追加や削除、クォータの変更、有効なプロジェクトの設定を行います。
- **Users:** ユーザーの表示、作成、編集、無効化、削除とユーザーパスワードの変更を行います。Users タブは、管理ユーザーとしてログインしている場合に利用できます。

Red Hat OpenStack Platform Dashboard を使用したクラウド管理について、詳細は以下のガイドを参照してください。

- [インスタンスの作成と管理](#)
- [イメージの作成および管理](#)
- [ネットワークガイド](#)
- [ユーザーおよびアイデンティティ管理ガイド](#)

第2章 DASHBOARD のカスタマイズ

Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) では、horizon コンテナ内に保管されるデフォルトのテーマ (RCUE) が使用されます。独自のテーマをコンテナイメージに追加し、特定のパラメーターをカスタマイズして Dashboard の以下の要素の外観と操作感を変更できます。

- ロゴ
- サイトの色
- スタイルシート
- HTML タイトル
- サイトのブランディングリンク
- ヘルプの URL



注記

変更後の RHOSP コンテナイメージが [Red Hat コンテナのサポートポリシー](#) に準拠していなければサポートは継続されません。

2.1. HORIZON コンテナイメージの取得

horizon コンテナイメージのコピーを取得するには、イメージをアンダークラウドまたは podman を実行する別のクライアントシステムのいずれかにプルします。

手順

- horizon コンテナイメージをプルします。

```
$ sudo podman pull registry.redhat.io/rhosp-rhel8/openstack-horizon:17.1
```

イメージ変更のベースとして、このイメージを使用できます。

2.2. RCUE テーマの取得

デフォルトでは、horizon コンテナイメージは Red Hat ブランドの RCUE テーマを使用します。コンテナイメージからテーマのコピーを抽出して、自分専用のテーマのベースとして使用できます。

手順

1. 自分専用テーマのディレクトリーを作成します。

```
$ mkdir ~/horizon-themes  
$ cd ~/horizon-themes
```

2. null ループを実行するコンテナを起動します。たとえば、以下のコマンドを実行します。

```
$ sudo podman run --rm -d --name horizon-temp registry.redhat.io/rhosp-rhel8/openstack-horizon /usr/bin/sleep infinity
```

- RCUE テーマをコンテナからローカルディレクトリーにコピーします。

```
$ sudo podman cp horizon-temp:/usr/share/openstack-  
dashboard/openstack_dashboard/themes/rcue .
```

- 次のコマンドで、コンテナを終了します。

```
$ sudo podman kill horizon-temp
```

結果: これで、RCUE テーマのローカルコピーができました。

2.3. RCUE をベースにした自分専用テーマの作成

RCUE をベースとして使用するには、RCUE テーマのディレクトリー rcue 全体を新しい場所にコピーします。以下の手順では、名前の例として **mytheme** を使用しています。

手順

- テーマをコピーします。

```
$ cp -r rcue mytheme
```

テーマの色、グラフィックス、フォント、およびその他の要素を変更するには、mytheme 内のファイルを編集します。このテーマを編集する場合は、パス、ファイル、ディレクトリーを含む rcue のすべてのインスタンスを確認し、それらを新しい mytheme の名前に変更してください。

2.4. 自分専用テーマを有効にして DASHBOARD をカスタマイズするためのファイルの作成

dashboard コンテナ内の自分専用テーマを有効にするには、**AVAILABLE_THEMES** パラメーターを上書きするファイルを作成する必要があります。

手順

- horizon-themes** ディレクトリー内に **_12_mytheme_theme.py** という名前の新たなファイルを作成し、以下の内容を追加します。

```
AVAILABLE_THEMES = [('mytheme', 'My Custom Theme', 'themes/mytheme')]
```

ファイル名の **12** により、このファイルが **11** を使用する RCUE ファイルの後に読み込まれ、**AVAILABLE_THEMES** パラメーターが上書きされます。

- オプション: **_12_mytheme_theme.py** ファイルで、カスタムパラメーターを設定することもできます。以下の例を参考にしてください。

SITE_BRANDING

ブラウザーウィンドウの上部に表示される HTML タイトルを設定します。

```
SITE_BRANDING = "Example, Inc. Cloud"
```

SITE_BRANDING_LINK

デフォルトでは、テーマのロゴのハイパーリンクは `horizon:user_home` にリダイレクトします。

```
SITE_BRANDING_LINK = "http://example.com"
```

2.5. 変更した HORIZON イメージの生成

カスタムテーマの作成が完了したら、自分専用テーマを使用する新たなコンテナイメージを作成できます。

手順

1. 以下の例に示すように、**Dockerfile** を使用して、元の **horizon** イメージをベースにコンテナイメージを生成します。

```
FROM registry.redhat.io/rhosp-rhel8/openstack-horizon
MAINTAINER Acme
LABEL name="rhosp-rhel8/openstack-horizon-mytheme" vendor="Acme" version="0"
release="1"
COPY mytheme /usr/share/openstack-dashboard/openstack_dashboard/themes/mytheme
COPY _12_mytheme_theme.py /etc/openstack-
dashboard/local_settings.d/_12_mytheme_theme.py
RUN sudo chown apache:apache /etc/openstack-
dashboard/local_settings.d/_12_mytheme_theme.py
```

2. このファイルを **Dockerfile** という名前で **horizon-themes** ディレクトリーに保存します。
3. Dockerfile を使用して新しいイメージを生成します。

```
$ sudo podman build . -t "172.24.10.10:8787/rhosp-rhel8/openstack-horizon:0-5" --log-level
debug
```

-t オプションで、生成されるイメージに名前およびタグを付けます。このオプションには、以下の構文を使用します。

```
[LOCATION]/[NAME]:[TAG]
```

LOCATION

最終的にオーバークラウドがイメージをプルするために通常使用するコンテナレジストリーの場所です。ここでは、このイメージをアンダークラウドのコンテナレジストリーにプッシュするので、この項目をアンダークラウドの IP アドレスおよびポートに設定します。

名前

一貫性を保つため、通常は元のコンテナイメージと同じ名前で、その後はユーザー独自のテーマの名前になります。ここでは、**rhosp-rhel8/openstack-horizon-mytheme** になります。

TAG

イメージのタグです。Red Hat では、このタグのベースとして **version** および **release** ラベルを使用します。このイメージの新しいバージョンを生成する場合は、**release** の数字を増やします (例: **0-2**)。

4. イメージをアンダークラウドのコンテナレジストリーにプッシュします。

```
$ sudo openstack tripleo container image push --local 172.24.10.10:8787/rhosp-
rhel8/openstack-horizon:0-5
```

5. イメージがローカルレジストリーにアップロードされたことを確認します。

```
[stack@director horizon-themes]$ curl http://172.24.10.10:8787/v2/_catalog | jq
.repositories[] | grep -i hori
"rhosp-rhel8/openstack-horizon"
[stack@director horizon-themes]$

[stack@director ~]$ sudo openstack tripleo container image list | grep hor
| docker://director.ctlplane.localdomain:8787/rhosp-rhel8/openstack-horizon:16.0-84
| docker://director.ctlplane.localdomain:8787/rhosp-rhel8/openstack-horizon:0-5
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
,Uploaded
[stack@director ~]$
```



重要

Red Hat OpenStack Platform を更新またはアップグレードする場合は、新しい **horizon** イメージにテーマを再度適用し、変更したイメージの新しいバージョンをアンダークラウドにプッシュする必要があります。

2.6. 変更したコンテナイメージのオーバークラウドでの使用

変更したコンテナイメージをオーバークラウドのデプロイメントに使用するには、コンテナイメージの場所のリストが含まれる環境ファイルを編集します。通常、この環境ファイルの名前は **overcloud-images.yaml** です。

手順

1. **DockerHorizonConfigImage** および **DockerHorizonImage** パラメーターを編集して、変更したコンテナイメージを指すようにします。

```
parameter_defaults:
  ...
  ContainerHorizonConfigImage: 192.168.24.1:8787/rhosp-rhel8/openstack-horizon-
mytheme:0-1
  ContainerHorizonImage: 192.168.24.1:8787/rhosp-rhel8/openstack-horizon-mytheme:0-1
  ...
```

2. この新しいバージョンの **overcloud-images.yaml** ファイルを保存します。

2.7. PUPPET パラメーターの編集

director には、環境ファイルを使用して変更できる Dashboard のパラメーターセットが用意されています。

手順

- **ExtraConfig** パラメーターを使用して Puppet hieradata を設定します。たとえば、デフォルトのヘルプ URL は [Red Hat OpenStack Platform の製品ドキュメント](#) を指しています。この URL を変更するには、以下の環境ファイルの内容を使用して、URL を置き換えます。

```
parameter_defaults:
  ExtraConfig:
    horizon::help_url: "http://openstack.example.com"
```

関連情報

- [ダッシュボードパラメーター](#)

2.8. SHARED FILE SYSTEMS サービスの共有プロトコルのカスタマイズ

カスタム環境ファイルの **ExtraConfig** パラメーターを使用して、OpenStack Dashboard (horizon) の Manila UI プラグインで使用可能な共有プロトコルのリストをカスタマイズできます。**ExtraConfig** パラメーターを変更して、Shared File Systems サービス (manila) のダッシュボードオプションをカスタマイズするために使用できる puppet パラメーターを統合します。

手順

1. アンダークラウドホストに **stack** ユーザーとしてログインします。
2. **stackrc** アンダークラウド認証情報ファイルを入手します。

```
$ source ~/stackrc
```

3. YAML 環境ファイルを作成するか開いて、ダッシュボードでクラウドユーザーが利用できるようにする共有プロトコルを設定します。

例

```
$ vi /home/stack/templates/<share_protocols>.yaml
```

- **<share_protocols>** をファイルの名前に置き換えます。
4. Shared File Systems サービスのオプションをカスタマイズするために使用できる puppet パラメーターを設定します。

```
parameter_defaults:
  ControllerExtraConfig:
    horizon::dashboards::manila::manila_options:
```



注記

カスタムロールを使用する場合は、**ControllerExtraConfig** パラメーターの代わりに **[role_name]ExtraConfig** パラメーターを使用します。**[role_name]** を、カスタムロールの名前に置き換えます。

5. 使用するプロトコル (CephFS プロトコルなど) を使用して、**enabled_share_protocols** のリストを設定します。

```
parameter_defaults:
```

```
ControllerExtraConfig:  
  horizon::dashboards::manila::manila_options:  
    enabled_share_protocols: ['CEPHFS']
```

6. **<share_protocols>.yaml** ファイルを他の環境ファイルとともにスタックに追加し、オーバークラウドをデプロイします。

```
(undercloud)$ openstack overcloud deploy --templates \  
-e [your environment files] \  
-e /home/stack/templates/<share_protocols>.yaml
```

2.9. DASHBOARD をカスタマイズしたオーバークラウドのデプロイ

手順

- Dashboard をカスタマイズしたオーバークラウドをデプロイするには、**openstack overcloud deploy** コマンドで以下の環境ファイルを指定します。
 - 変更したコンテナイメージの場所を定義した環境ファイル
 - 追加の dashboard 変更を定義した環境ファイル
 - ユーザーのオーバークラウド設定に関連するその他すべての環境ファイル

```
$ openstack overcloud deploy --templates \  
-e /home/stack/templates/overcloud-images.yaml \  
-e /home/stack/templates/help_url.yaml \  
[OTHER OPTIONS]
```