



# Red Hat OpenStack Platform 13

## OpenStack データ処理

Red Hat OpenStack Platform での Hadoop クラスターの手動プロビジョニングおよびスケーリング



## Red Hat OpenStack Platform 13 OpenStack データ処理

---

Red Hat OpenStack Platform での Hadoop クラスターの手動プロビジョニングおよびスケーリング

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/OpenStack\_Data\_Processing.ent file |.

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

OpenStack Data Processing 機能により、Hadoop クラスターを簡単にプロビジョニングおよびスケーリングして、大規模なデータセットを処理できます。本書では、OpenStack Data Processing ワークフロー全体について説明します。このワークフローには、データ処理要件（イメージ、入力データ、ジョブバイナリー）の登録、クラスターのプロビジョニングに使用するテンプレートの設定、これらのクラスターでのデータの処理、必要に応じてこれらのクラスターのスケーリングが含まれます。このリリースの OpenStack Data Processing には、Guides タブが含まれています。このタブ機能ウィザードは、クラスターを起動し、それらに対してジョブを実行するために必要なテンプレートを作成するのに役立ちます。本書の目的は、OpenStack Data Processing ワークフロー

の詳細を確認することです。そのため、Guidesタブ機能を使用せずにテンプレートの作成とコンポーネントの登録について説明します。OpenStack Data Processing 機能を使用するには、Hadoop フレームワーク内のデータ処理に関する基本的な知識が必要です。さらに、ユーザーは選択したHadoop プラグインの内容を理解する必要もあります。

---

## 目次

第1章 概要 .....	3
第2章 インストールシステム .....	4
第3章 ワークフロー .....	5
第4章 HOPA イメージの作成 .....	7
第5章 必要なコンポーネントの登録 .....	10
5.1. 入出力データソースの登録 .....	10
第6章 ノードグループテンプレートの設定 .....	12
第7章 クラスターテンプレートの設定 .....	14
第8章 クラスターの起動 .....	16
8.1. クラスターのスケールリングまたは削除 .....	16
第9章 ジョブの設定 .....	18
9.1. ジョブバイナリー、スクリプト、またはライブラリーの登録 .....	18
9.2. ジョブテンプレートの作成 .....	19
第10章 ジョブの起動 .....	20
10.1. 既存クラスターでのジョブの起動 .....	20
10.2. 新規クラスターでのジョブの起動 .....	20
10.3. 起動したジョブの削除または再起動 .....	21
付録A インスタンスのローカリティーの有効化 .....	23



## 第1章 概要

OpenStack Data Processing サービス(sahara)は、Hadoop クラスターを簡単にプロビジョニングおよびスケールアップするための堅牢なインターフェイスを提供します。このようなクラスターは、通常は大規模なデータセットを処理するために、リソース集約型ジョブを実行するために使用できます。

OpenStack コンポーネントとして、OpenStack Data Processing は OpenStack エコシステムに完全に統合されています。たとえば、ユーザーは、クラスターを設定して、Hadoop データ処理ワークフロー全体を、それらでジョブを起動して実行する方法で管理できます。

Hopa の詳細は、<http://hadoop.apache.org/> を参照してください。



### 注記

Data Processing サービス(sahara)は OpenStack Platform バージョン 15 で非推奨になり、バージョン 16 で削除される予定です。

OpenStack Data Processing は、Hadoop ディストリビューションの特定のクラスターをプロビジョニングするために異なるプラグインを使用します。クラスターの設定に使用できる Hadoop パラメーターは、Hadoop ディストリビューション（また、拡張、使用するプラグイン）によって異なります。本リリース(Red Hat OpenStack Platform 13)の時点で、OpenStack Data Processing は以下のプラグインをサポートします。

- Cloudera CDH 5
  - [バージョン 5.7.x](#) (デフォルトは 5.7.0)
  - [バージョン 5.9.x](#) (デフォルトは 5.9.0)
  - [バージョン 5.11.x](#) (デフォルトは 5.11.0)
- Hortonworks Data Platform 2.4
  - [バージョン: 8.29](#)
  - [バージョン: 8.29](#)
- [MapR 5.2](#)

OpenStack Data Processing には、**Guides** タブも含まれます。このタブ機能ウィザードは、クラスターを起動し、それらに対してジョブを実行するために必要なテンプレートを作成するのに役立ちます。ただし、Hadoop イメージやジョブバイナリーなどの OpenStack Data Processing の使用に必要なコンポーネントを登録する必要があります。そのため、**Guides** 機能を使用する場合は、最初に [5章 必要なコンポーネントの登録](#) を読むことが推奨されます。



## 第2章 インストールシステム

本項では、director を使用して OpenStack Data Processing をオーバークラウドにデプロイすることを前提としています。また、director のインストールと使用方法ガイドの stack ユーザー の作成と同じ director インストール [ユーザーを作成](#) していることを前提とします。

director は [環境ファイル](#) を使用して、デプロイメント中にオーバークラウドを設定します。これらの環境ファイルは、`/usr/share/openstack-tripleo-heat-templates/` に、各サービスのオーケストレーションに使用する Puppet スクリプトおよび heat テンプレートと共に保存されます。

デフォルトでは、OpenStack Data Processing サービスはオーバークラウドで無効になっています。これは、OpenStack Data Processing コンポーネントは、メインのオーバークラウド環境ファイル (`/usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-puppet.yaml`) で null 操作 (`OS::Heat::None`) として登録されるためです。

```
OS::TripleO::Services::SaharaApi: OS::Heat::None
OS::TripleO::Services::SaharaEngine: OS::Heat::None
```

これらのサービスを有効にするには、`/usr/share/openstack-tripleo-heat-templates/puppet/services` ディレクトリー内の対応する Heat テンプレートにこれらのリソースをリンクする環境ファイルを追加します。一部のサービスには、`environments` ディレクトリー内に事前定義済みの環境ファイルがあります。たとえば、Data Processing サービス(`sahara`)は、以下を含む `/usr/share/openstack-tripleo-heat-templates/environments/services/sahara.yaml` ファイルを使用します。

```
resource_registry:
  OS::TripleO::Services::SaharaApi: ../../docker/services/sahara-api.yaml
  OS::TripleO::Services::SaharaEngine: ../../docker/services/sahara-engine.yaml
```

これにより、デフォルトの null 操作のリソースが上書きされ、これらのサービスが有効になります。`openstack overcloud deploy` コマンドの実行時に、この環境ファイルを指定します。

```
$ openstack overcloud deploy --templates -e \
  /usr/share/openstack-tripleo-heat-templates/environments/services/sahara.yaml
```



### 注記

インスタンスのローカリティーを有効にして、クラスターが同じノードに対してローカルのボリュームを使用するように強制することもできます。これを行うと、クラスターのパフォーマンスが向上します。関連情報は、[付録A インスタンスのローカリティーの有効化](#) を参照してください。



### 重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合には、予定外の変更がオーバークラウドに加えられないように、ここで `-e` オプションを使用して環境ファイルを再度渡します。詳細は、director のインストールと使用方法ガイドの [オーバークラウド環境の変更](#) を参照してください。

## 第3章 ワークフロー

OpenStack Data Processing は、特別に設計されたインスタンスを定義する事前設定されたクラスターテンプレートを使用して、Hadoop クラスターをプロビジョニングし、スケールリングします。これらのインスタンスは、Hadoop クラスターを設定する個々のノードを形成します。次に、これらの Hadoop クラスターを使用して、データを処理するジョブ/バイナリーを実行できます。

OpenStack Data Processing を使用する場合は、Hadoop フレームワーク内で機能するために必要なコンポーネントをすでに理解している必要があります。そのため、本セクションで説明している一般的なワークフローでは、以下のコンポーネントがすでに準備されていることを前提としています。

- Hado イメージ。特に、Hadoop データ処理プラグインを含む Red Hat Enterprise Linux イメージ。サポートされるプラグインの一覧は、[1章概要](#)を参照してください。
- 処理する入力データ。Object Storage サービスにアップロードされることが推奨されます。
- 入力データの処理に使用するジョブバイナリーおよびライブラリー。Object Storage サービスにアップロードされることが推奨されます。



### 注記

Object Storage サービスにコンテンツをアップロードする方法の詳細は、[Uploading an Object](#) を参照してください。

また、ジョブの実行に必要な計算リソースの一般的な概念も必要です。これにより、必要なノードのタイプ（および各ノード数）を判断しやすくなります。

以下のワークフローの概要では、OpenStack Data Processing サービスを設定して使用し、クラスターを起動し、これらのクラスターでジョブを実行する方法を説明します。

1. OpenStack Data Processing ([4章Hopa イメージの作成](#))に必要なプラグインコンポーネントを含むイメージを作成します。これは、Hadoop イメージになります。  
このイメージの作成手順は、選択した Hive プラグインによって異なります。
2. OpenStack Data Processing サービスに以下のコンポーネントを登録します。
  - Hadoop イメージ
  - データソース（入力データおよび出力データの場所）
3. ノードグループテンプレートを作成します。各テンプレートは、指定ノードに対して多くの便利な Hadoop 固有の設定を定義します。特に、以下のようになります。
  - ノードグループが使用するのは、Hadoop プラグインとバージョンは何ですか？
  - ノードで実行されるプロセス
4. クラスターテンプレートを作成またはアップロードします。クラスターテンプレートは、以下のような内容を定義します。
  - **ノードグループの設定:** つまり、各ノードグループのノード数。
  - **クラスタースコープのHadoop設定:** 各Hadoopコンポーネント(HIVE、AMBARI、HDFS など)に設定する必要がある特定のパラメーターです。
5. (クラスターテンプレートを使用して) Hado クラスターを起動し、クラスターでジョブを実

行します（つまり、データソースで登録済みのジョブバイナリーを実行します）。必要に応じて、クラスターをスケーリングすることもできます（任意のタイプのノードを追加または削除します）。

6. ジョブバイナリー、スクリプト、またはライブラリーを OpenStack Data Processing Service に登録し、ジョブを作成し、Hadoop クラスターで起動します。ジョブは、登録したデータソースを処理するために使用するジョブバイナリー、スクリプト、またはライブラリーを定義します。

次の数セクションでは、各ワークフローの手順について詳しく説明します。

## 第4章 HOPA イメージの作成

本リリースでは、以下のプラグインをサポートします。

- Cloudera CDH 5
  - バージョン 5.7.x (デフォルトは 5.7.0)
  - バージョン 5.9.x (デフォルトは 5.9.0)
  - バージョン 5.11.x (デフォルトは 5.11.0)
- Hortonworks Data Platform 2.4
  - バージョン: 8.29
  - バージョン: 8.29
- MapR 5.2

これらのプラグインのいずれかに基づいて Hadoop イメージを作成するには、まず Red Hat Enterprise Linux 7 イメージをダウンロードする必要があります。以下のリンクから取得できます (Red Hat サブスクリプションが必要です)。

[https://access.redhat.com/downloads/content/69/ver=/rhel---7/7.5/x86\\_64/product-software](https://access.redhat.com/downloads/content/69/ver=/rhel---7/7.5/x86_64/product-software)

イメージを準備する前に、システムが、Hadoop イメージを作成するために必要なツールをダウンロードできるように、Red Hat サブスクリプションの詳細を使用するようにシステムを設定します。これを行うには、(root として)ターミナルから以下のコマンドを入力します。

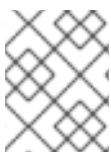
```
# export REG_USER=USERNAME
# export REG_PASSWORD=PASSWORD
# export REG_POOL_ID=POOLID
```

詳細は以下のようになります。

- **username** および **PASSWORD** は、Red Hat サブスクリプションの認証情報です。
- **POOLID** は Red Hat サブスクリプションのプール ID です。サブスクリプションのプール ID を確認するには、以下のコマンドを入力します。

```
# subscription-manager list --available
```

このコマンドは、**Pool ID** 値など、サブスクリプションに関する重要な情報を表示します。**プール ID** には Red Hat OpenStack Platform チャンネルが含まれている必要があります。



### 注記

Red Hat サブスクリプションを手動で設定する方法の詳細は、[Using and Configuring Red Hat Subscription Manager](#) を参照してください。

**openstack-sahara-image-pack** パッケージをシステムにインストールします。

```
# yum install openstack-sahara-image-pack -y
```

Red Hat サブスクリプションを使用するように Red Hat Enterprise Linux 7 イメージを設定します。イメージは以下のチャンネルに登録されている必要があります。

- rhel-7-server-rpms
- rhel-7-server-optional-rpms

これには以下のコマンドを実行します。

```
# yum install libguestfs-tools -y
```

以下のコマンドは root 以外のユーザーで実行する必要があります。

```
$ virt-customize -a IMAGE \  
  --sm-register \  
  --sm-credentials USERNAME:password: _PASSWORD_ \  
  --sm-attach pool: _POOLID_ \  
  --run-command 'subscription-manager repos --disable=* --enable rhel-7-server-rpms --enable rhel-7-server-optional-rpms'
```

詳細は以下のようになります。

- **IMAGE** は Red Hat Enterprise Linux 7 イメージの名前です。
- **username** および **PASSWORD** は、Red Hat サブスクリプションの認証情報です。
- **POOLID** は Red Hat サブスクリプションのプール ID です。このプールには、基本的な Red Hat Enterprise Linux チャンネルのみが含まれている必要があります。

カスタマイズプロセスを開始するには、Red Hat Enterprise Linux イメージと同じディレクトリーから以下のコマンドを実行します。

```
$ sahara-image-pack --image IMAGE PLUGIN VERSION
```

**PLUGIN** は、Hadoop イメージが使用するプラグインに置き換えます。

以下の値は **PLUGIN** および **VERSION** に使用できます。

- **CDH** ([Cloudera](#)用)
  - 5.7.0 ([バージョン 5.7.x](#)用)、5.9.0 ([バージョン 5.9.x](#)の場合)、または 5.11.0 ([バージョン 5.11.x](#)の場合)をバージョン番号として指定します。
- **ambari** ([Hortonworks Data Platform](#))
  - 2.4 をバージョン番号として指定します。Ambari 2.4 イメージは、Hortonworks Data Platform 2.3 と 2.4 の両方に使用できます。
- **MapR** ([MapR](#))
  - 5.2.0 ([MapR 5.2](#))をバージョン番号として指定します。

openstack-sahara-image-pack コマンドは、プラグインに必要なソフトウェアを追加して、指定したベースイメージを変更します。

サブスクリプションを削除し、SELinux コンテキストを修正して、イメージの最終処理を行います。次のコマンドを実行します。

```
$ virt-customize -a IMAGE --sm-unregister --selinux-relabel
```

## 第5章 必要なコンポーネントの登録

OpenStack Data Processing には、Hadoop クラスターを起動し、使用するために必要な要素を含む、Hadoop イメージが必要です。具体的には、Red Hat OpenStack Platform には、必要なデータ処理プラグインを備えた Red Hat Enterprise Linux を含むイメージが必要です。

実行するジョブに適したHadoopイメージを取得したら、それを OpenStack Data Processing サービスに登録します。これを行うには、以下を行います。

1. イメージを Image サービスにアップロードします。手順については、イメージの [アップロードを参照してください](#)。
2. イメージのアップロード後に、ダッシュボードで プロジェクト > データ処理 > イメージレジストリー を選択します。
3. Register Image をクリックして、Image ドロップダウンメニューから、Hadoop イメージを選択します。
4. 各インスタンス/ノードで設定を適用し、プロセスを管理するために OpenStack Data Processing サービスが使用するユーザー名を入力します。Red Hat Enterprise Linux が提供する公式イメージ( [4章 Hopa イメージの作成](#) で使用)でこの目的のユーザー名は cloud-user です。
5. デフォルトでは、OpenStack Data Processing サービスは、必要なプラグインおよびバージョンタグを プラグイン およびバージョンメニューに追加します。タグの選択が正しいことを確認してから、Add plugin tags をクリックして追加します。OpenStack Data Processing サービスでは、カスタムタグを使用して、登録したイメージを区別またはグループ化することもできます。Add custom tag ボタンを使用してタグを追加します。タグは Description フィールドのボックスに表示されます。  
カスタムタグを削除するには、その名前の横にある x をクリックします。
6. Done をクリックします。イメージが Image Registry テーブルに表示されます。

### 5.1. 入出力データソースの登録

イメージを登録したら、データ入力ソースおよび出力先を登録します。Object Storage サービスからオブジェクトとして登録できます。そのため、両方を最初にオブジェクトとしてアップロードする必要があります。手順については、[Upload an Object](#) を参照してください。



#### 注記

別の Hadoop 互換の分散ファイルシステム(HDFS など)からデータオブジェクトを直接登録することもできます。選択した分散ファイルシステムにデータをアップロードする方法は、そのドキュメントを参照してください。

1. Dashboard で プロジェクト > Data Processing > Data Sources の順に選択します。
2. Create Data Source をクリックします。Name フィールドにデータソースの名前を入力します。
3. Description フィールドを使用してデータソースを記述します (オプション)。
4. データソースのタイプおよび URL を選択します。これを行う手順は、ソースの場所によって異なります。

- データが Object Storage サービスにある場合は、Data Source Type ドロップダウンメニューから Swift を選択します。以下のようになります。
    - a. URL フィールドに `swift://CONTAINER/OBJECT` としてデータソースのコンテナおよびオブジェクト名を指定します。
    - b. データソースにログインが必要な場合は、Source username フィールドおよび Source password フィールドに必要な認証情報を指定します。
  - データが HDFS (Hadoop Distributed File System)にある場合は、Data Source Type ドロップダウンメニューから対応するソースを選択します。次に、URL フィールドにデータソースの URL を `hdfs://HDFSHOST:PORT/OBJECTPATH` として入力します。
    - HDFSHOST は、HDFS ホストのホスト名です。
    - PORT はデータソースにアクセスできるポートです。
    - OBJECTPATH は、HDFSHOST のデータソースへの利用可能なパスです。
5. Done をクリックします。データソースはデータソーステーブルで利用できるようになります。

この手順は、ジョブに必要なデータ入力/出力オブジェクトごとに実行します。



## 第6章 ノードグループテンプレートの設定

Hopa によるデータ処理には、クラスター上でジョブの実行ジョブが含まれます。各クラスターは、特別に設定されたノード（またはインスタンス）で設定されます。OpenStack Data Processing では、各クラスターに必要なさまざまな種類のノードを定義するテンプレートを設定できます。これらのテンプレートは、ノードグループテンプレートと呼ばれます。

ノードグループのテンプレートは、Hadoop クラスターの各ノードに必要な設定を定義します。以下の手順では、ノードグループテンプレートを設定する方法を説明します。

1. Dashboard で **プロジェクト > データ処理 > ノードグループテンプレート** を選択します。
2. **Create Template** をクリックします。
3. **Plugin Name** および **Version** ドロップダウンメニューを使用して、ノードが使用する Hadoop プラグインの名前とバージョンを選択します。
4. **Create** をクリックします。
5. **Template Name** フィールドにテンプレートの名前を入力します。
6. **Description** フィールドを使用して、作成するノードグループテンプレートを記述します（オプション）。
7. **OpenStack Flavor** ドロップダウンメニューから、ノードが使用するフレーバーを選択します。フレーバーは、ノードのコンピューティング、メモリー、およびストレージのニーズに対応する必要があります。フレーバーについての詳しい情報は、フレーバーの [管理](#) を参照してください。
8. **Storage location** ドロップダウンメニューから、ノードのストレージの場所を選択します。
  - **一時ドライブ**: OpenStack Data Processing サービスは、ノードに必要な一時ストレージを生成します。
  - **Cinder ボリューム**: このオプションを使用して、Block Storage サービスを設定して、各ノードにボリュームセットを作成することができます。Volumes per node フィールドを使用してボリュームの数を設定し、Volumes size (GB) フィールドを使用して各ボリュームのサイズを指定します。  
ほとんどのデプロイメントでは、Cinder Volume オプションを推奨します。一時ドライブのデータは、ホストの整合性に依存するため、ホストの障害に対して脆弱です。
9. **Floating IP プール** のドロップダウンメニューを使用して、ノードが Floating IP を使用するかどうかを設定します。また、ある場合はこれを設定します。Floating IP の詳細は、[IP Addressing の設定](#) を参照してください。
10. 次に、ノードのセキュリティーを設定します。ノードグループのセキュリティーグループを作成するには、Auto Security Group チェックボックスを選択します。また、セキュリティーグループリストから該当するチェックボックスを選択して、既存のセキュリティーグループでノードを起動することもできます。  
セキュリティーグループの詳細は、[プロジェクトセキュリティー管理](#) を参照してください。
11. 最後に、ノードグループで起動する Hive プロセスを選択します。これを行うには、プロセスリストから選択した各プロセスのチェックボックスを選択します。各プロセスの詳細については、選択した Hadoop プラグインのドキュメントを参照してください。
12. **Create** をクリックします。テンプレートは Node Group Template の表に表示されます。

Node Group Templateの表には、利用可能なすべてのノードグループテンプレート、それぞれのHadoopプラグイン、およびバージョンが記載されています。この表には、各テンプレートのノードでどのHadoopプロセスが実行されるかも表示されます。

## 第7章 クラスターテンプレートの設定

必要なコンポーネントを登録し(5章 [必要なコンポーネントの登録](#))し、必要なノードの種類を設定したら(6章 [ノードグループテンプレートの設定](#))、クラスターテンプレートを設定できるようになりました。クラスターテンプレートは、クラスターの以下の設定を定義します。

- クラスターのノード設定（各ノード種別の数など）。利用可能なノードタイプは、Project > Data Processing > Node Group Templates に一覧表示されます。
- クラスターが使用する、Hadoop プラグインおよびバージョン。
- 単一ホスト(非アフィニティー)で1回だけ起動する必要があるプロセスの一覧。
- Hopa サービスのクラスターレベルのパラメーター。

クラスターテンプレートを作成および設定するには、以下を実行します。

1. Dashboard で プロジェクト > データ処理 > クラスターテンプレート を選択します。
2. Plugin Name および Version ドロップダウンメニューを使用して、クラスターが使用する Hadoop プラグインの名前とバージョンを選択します。
3. Create をクリックします。
4. Template Name フィールドにテンプレートの名前を入力します。
5. Description フィールドを使用して、作成するクラスターテンプレートを記述します（オプション）。
6. Use anti-affinity groups for list から、ホストで複数回起動してはならない各プロセスのボックスにチェックを入れます。
7. Node Groups タブをクリックします。ここから、各ノードで実行する各ノード種別のノード数を設定します。これを実行するには、以下を行います。
  - a. Select a Node Group Template to add: ドロップダウンメニューを使用してテンプレートを選択します（関連情報は [6章 ノードグループテンプレートの設定](#) を参照してください）。
  - b. + をクリックして、テンプレートに基づいてノードグループエントリーを作成します。これは、クラスターテンプレートで使用するノードグループテンプレートごとに行います。
  - c. 各ノードグループエントリーに、クラスター内で起動するノードタイプのインスタンス数を設定します。  
各ノードのグループエントリーの名前も入力できます（オプション）。
8. 必要に応じて、コンポーネントで、Hadoop クラスターを設定します。各コンポーネント（例：ZOOKEEPER、AMBARI、HUE）には独自のパラメータータブがあります。各タブには関連パラメーターが含まれており、そのほとんどは簡単なコンテキストヘルプがあります。これらのパラメーターの詳細は、各コンポーネントでサポートされている機能について、選択した Hadoop プラグインのドキュメントを参照してください。
9. クラスターテンプレートの設定が適切になったら、Create をクリックします。テンプレートは Cluster Templates の表に表示されます。

または、Upload Template ボタンを使用して、ローカルファイルシステムから既存のクラスターテンプレートをアップロードすることもできます。

Cluster Templates の表から既存のテンプレートを使用してクラスターを起動することもできます。これには、クラスターテンプレートの Actions ドロップダウンメニューから Launch Cluster を選択します。起動したクラスターを表示するには、Project > Data Processing > Clusters を選択します。クラスターの起動に関する詳細は、[8章 クラスターの起動](#) を参照してください。

## 第8章 クラスターの起動

クラスターテンプレート([7章 クラスターテンプレートの設定](#)を参照)したら、クラスターを起動できます。これには、Launch Cluster ウィザードを開きます。このウィザードを開く方法は複数あります。

- プロジェクト > Data Processing > Cluster Templates の順に選択します。そこから、テンプレートの Action ドロップダウンメニューから Launch Cluster を選択します。
- Project > Data Processing > Clusters の順に選択します。そこから、Launch Cluster をクリックします。次に、Plugin Name および Hadoop Version ドロップダウンメニューを使用して、クラスターが使用する HQL プラグインの名前とバージョンを選択します。

その後、Create をクリックします。これにより、Launch Cluster ウィザードが開きます。

1. クラスターの 起動 ウィザードで、Cluster Name フィールドにクラスターの名前を入力します。
2. Description フィールドを使用して、起動するクラスターを記述します (オプション)。
3. Cluster の表で Launch Cluster ウィザードを Project > Data Processing > Clusters の表で開いた場合は、Cluster Template ドロップダウンメニューを使用してクラスターテンプレートを選択できます。事前に選択した Hadoop プラグインおよびバージョンと互換性のあるテンプレートのみを選択できます。
4. Base Image ドロップダウンメニューから、クラスターが使用する Hadoop イメージを選択します。Hadoop イメージの作成および登録の詳細は、[4章 Hopa イメージの作成](#)および[5章 必要なコンポーネントの登録](#)を参照してください。
5. 必要に応じて、Keypair ドロップダウンメニューからキーペアを選択します。このメニューの横にある + をクリックして、新しいキーペアを作成することもできます。クラスターを起動するのにキーペアは必要ありませんが、クラスターノードにログインする必要があります (例: SSH を使用)。  
キーペアの詳細は [キーペアの管理](#) を参照してください。
6. Neutron Management Network ドロップダウンメニューから、クラスターが使用するネットワークを選択します。OpenStack でのネットワークの追加および管理に関する詳細は、[Common Administrative Tasks](#) を参照してください。
7. Create をクリックしてクラスターを起動します。

起動したクラスターを表示するには、Project > Data Processing > Clusters を選択します。

### 8.1. クラスターのスケーリングまたは削除

OpenStack Data Processing により、リソースのニーズに合うように既存のクラスターを簡単にスケーリングできます。スケーリングにより、実行中のクラスターから任意のタイプ (または数字) のノードを追加または削除できます。

起動したクラスターを表示するには、Project > Data Processing > Clusters を選択します。このページからクラスターをスケーリングまたは削除できます。クラスターの起動方法は、[8章 クラスターの起動](#)を参照してください。

クラスターをスケーリングするには、以下を実行します。

1. Clusters テーブルで、スケーリングするクラスターを選択します。次に、クラスターの Action ドロップダウンメニューから Scale Cluster を選択します。必要に応じてクラスターを縮小します。
  - 新規ノードタイプをクラスターに追加するには、Select a Node Group Template からそのテンプレートを選択して、ドロップダウンメニューから追加します。次に、+ ボタンをクリックしてノードタイプをクラスターに追加します。続いて、追加する必要があるタイプのノード数を設定できます。
  - 既存ノードグループからノードを追加または削除するには、ノードグループの行にある + ボタンまたは - ボタンを使用します。または、ノードグループの Count フィールドでノード数を設定することもできます。
2. Scale をクリックします。

クラスターを削除するには、Action ドロップダウンメニューから Delete Cluster を選択します。チェックボックスをオンにして、クラスターの削除 ボタンをクリックして、複数のクラスターを削除することもできます。

## 第9章 ジョブの設定

OpenStack Data Processing サービスでは、ジョブは実際のデータ処理タスクを定義します。各ジョブは、ジョブタイプ（例：Pig、Hive、MapReduce）、バイナリー、スクリプト、ライブラリーを指定します。ジョブは、OpenStack Data Processing サービスに登録されているバイナリー、スクリプト、またはライブラリーのみを使用できます。

ジョブの作成後に、クラスターで起動し、入力データソースに対して実行できます。ジョブバイナリーなどの入出力データソースも、最初に OpenStack Data Processing サービスに登録する必要があります（「[入出力データソースの登録](#)」を参照してください）。

### 9.1. ジョブバイナリー、スクリプト、またはライブラリーの登録

ジョブバイナリー、スクリプト、およびライブラリーを登録するプロセスは、イメージおよびデータソースの登録と似ています。Object Storage サービスから直接登録できます。オブジェクトを Object Storage サービスにアップロードする方法については、[Upload an Object](#)を参照してください。または、ローカルファイルシステムから直接 OpenStack Data Processing サービスにバイナリーおよびライブラリーをアップロードすることもできます。

1. Dashboard で **プロジェクト > Data Processing > Job Binaries** を選択します。
2. **ジョブバイナリーの作成** をクリックします。
3. オブジェクトの名前を入力します（スクリプト、バイナリー、またはライブラリー）。この名前はオブジェクトの選択時に使用されます。オブジェクトに特定の名前またはエクステンションが必要な場合は（.jar など）、ここに追加します。
4. **Description** フィールドを使用して、登録するスクリプト、バイナリー、またはライブラリーを記述します（任意）。
5. ストレージタイプに応じてオブジェクトを設定します。
  - a. オブジェクトが Object Storage サービスで利用可能な場合は、**Storage type** ドロップダウンメニューから **Swift** を選択します。以下ようになります。
    - URL フィールドに `swift://CONTAINER/OBJECT` として、スクリプト、バイナリー、またはライブラリーのコンテナおよびオブジェクト名を指定します。
    - スクリプト、バイナリー、またはライブラリーでログインが必要な場合は、**Username** フィールドおよび **Password** フィールドに必要な認証情報を指定します。
  - b. それ以外の場合は、**Storage type** ドロップダウンメニューから **Internal database** を選択します。次に、**Internal バイナリー** ドロップダウンメニューを使用して、以下のいずれかを行います。
    - OpenStack Data Processing サービスから、利用可能なバイナリー、ライブラリー、またはスクリプトを選択します。
    - ダッシュボードに直接スクリプトを入力する（スクリプトの作成）、または
    - バイナリー、ライブラリー、またはスクリプトをローカルファイルシステムから直接アップロードします（新しいファイルをアップロードします）。
6. **Create** をクリックします。バイナリー、ライブラリー、またはスクリプトが **Job Binaries** テーブルで利用できるようになりました。

## 9.2. ジョブテンプレートの作成

必要なバイナリー、スクリプト、およびライブラリーを OpenStack Data Processing に登録したら、以下の手順を実行します。

1. Dashboard で プロジェクト > データ処理 > ジョブテンプレート を選択します。
2. ジョブテンプレートの作成 をクリックします。
3. Name フィールドにジョブの名前を入力します。
4. Job Type ドロップダウンメニューから正しいタイプを選択します。ジョブタイプの詳細は、選択したプラグインのドキュメントを参照してください。
5. Choose a main binary ドロップダウンメニューから、このジョブに使用するバイナリーを選択します。このメニューのオプションは、OpenStack Data Processing サービスに登録されているジョブバイナリーとスクリプトと共に入力されます。詳細は、[「ジョブバイナリー、スクリプト、またはライブラリーの登録」](#) を参照してください。
6. Description フィールドを使用して、作成するジョブを記述します（オプション）。
7. 指定したジョブバイナリーにライブラリーが必要な場合は、追加します。これを行うには、Libs タブをクリックして、Choose libraries ドロップダウンメニューからライブラリーを選択します。次に、Chosen ライブラリー一覧にライブラリーを追加するには、Chosen をクリックしてジョブに追加します。ジョブバイナリーで必要なすべてのライブラリーに対してこれを繰り返します。バイナリーと同様に、Choose libraries ドロップダウンメニューのオプションには、OpenStack Data Processing サービスに登録されているライブラリーが入力されます。詳細は、[「ジョブバイナリー、スクリプト、またはライブラリーの登録」](#) を参照してください。
8. Create をクリックします。ジョブはジョブテーブルで利用できるようになります。



## 第10章 ジョブの起動

ジョブを作成したら、ジョブを起動して、OpenStack Data Processing サービスに登録されているデータを処理できます(「[入出力データソースの登録](#)」を参照)。ジョブにはクラスターが必要です。既存のクラスター(「[既存クラスターでのジョブの起動](#)」)でジョブを起動するか、完全に新しいクラスター(「[新規クラスターでのジョブの起動](#)」)でジョブを起動できます。



### 注記

ジョブを起動するには、入力データソースと出力データの宛先を指定する必要があります。どちらのオブジェクトも、最初に OpenStack Data Processing サービスに登録する必要があります。詳細は、「[入出力データソースの登録](#)」を参照してください。

### 10.1. 既存クラスターでのジョブの起動

Dashboard で既存クラスターの一覧を表示するには、Project > Data Processing > Clusters の順に選択します。クラスターの起動方法は、[8章 クラスターの起動](#)を参照してください。

既存クラスターでジョブを起動するには、以下を実行します。

1. Dashboard で [プロジェクト > データ処理 > ジョブ](#) を選択します。ジョブテーブルには、利用可能なすべてのジョブテンプレートが表示されます。新規ジョブテンプレートの作成に関する詳細は、「[ジョブテンプレートの作成](#)」を参照してください。
2. 使用するジョブテンプレートを選択してから、ジョブテンプレートの Actions ドロップダウンメニューから Launch On Existing Cluster を選択します。
3. Launch Job ウィザードで、Input ドロップダウンメニューから入力データソースを選択します。次に、Output ドロップダウンメニューから出力先を選択します。必要に応じて、ここから入力データソースまたは出力先を登録することもできます。これを行うには、Input または Output ドロップダウンメニューで + をクリックします。これにより、Create Data Source ウィザードが開きます。詳細は、「[入出力データソースの登録](#)」を参照してください。
4. Cluster ドロップダウンメニューから、ジョブを実行するクラスターを選択します。
5. このジョブに特別なジョブプロパティを設定する必要がある場合は、Configure タブをクリックします。そこから、Configuration または Parameters のどちらかの Add をクリックして、特別な名前と値のペアを指定します。このタブでは、複数の名前と値のペアを指定できません。サポートされるジョブプロパティの詳細については、選択した Hadoop プラグインのドキュメントを参照してください。
6. Launch をクリックします。

起動したジョブのステータスを表示するには、プロジェクト > Data Processing > Jobs を選択します。起動済みのジョブの再起動または削除方法については、「[起動したジョブの削除または再起動](#)」を参照してください。

### 10.2. 新規クラスターでのジョブの起動

ジョブテンプレートを作成したら、これを使用して完全に新しいクラスターでジョブを起動することもできます。これにより、ジョブの完了後にクラスターを自動的に強制終了するオプションが提供されます。

1. Dashboardでプロジェクト>データ処理>ジョブを選択します。ジョブテーブルには利用可能なすべてのジョブが表示されます。新規ジョブの作成に関する詳細は、「[ジョブテンプレートの作成](#)」を参照してください。
2. 使用するジョブテンプレートを選択してから、ジョブテンプレートの Actions ドロップダウンメニューから Launch On New Cluster を選択します。
3. プラグインの Name および Version ドロップダウンメニューを使用して、ジョブが使用する Hadoop プラグインの名前とバージョンを選択します。
4. Create をクリックします。
5. Cluster Name フィールドにクラスターの名前を入力します。
6. Base Image ドロップダウンメニューから、クラスターが使用する Hadoop イメージを選択します。Hadoop イメージの作成および登録の詳細は、[4章 Hopsa イメージの作成](#)および[5章 必要なコンポーネントの登録](#)を参照してください。
7. 必要に応じて、Keypair ドロップダウンメニューからキーペアを選択します。このメニューの横にある + をクリックして、新しいキーペアを作成することもできます。クラスターを起動するのにキーペアは必要ありませんが、クラスターノードにログインする必要があります（例：SSH を使用）。  
キーペアの詳細は [キーペアの管理](#) を参照してください。
8. Neutron Management Network ドロップダウンメニューから、クラスターが使用するネットワークを選択します。OpenStack でのネットワークの追加および管理に関する詳細は、[Common Administrative Tasks](#) を参照してください。
9. デフォルトでは、OpenStack Data Processing サービスは、ジョブが完了するとすぐにクラスターを削除します。これが発生しないようにするには、ジョブの終了後に Persist cluster を選択します。
10. 次に、Job タブをクリックします。そこから、Input ドロップダウンメニューから入力データソースを選択します。次に、Output ドロップダウンメニューから出力先を選択します。
11. 必要に応じて、ここから入力データソースまたは出力先を登録することもできます。これを行うには、Input または Output ドロップダウンメニューで + をクリックします。これにより、Create Data Source ウィザードが開きます。詳細は、「[入出力データソースの登録](#)」を参照してください。
12. このジョブに特別なジョブプロパティを設定する必要がある場合は、Configure タブをクリックします。そこから、Configuration または Parameters のどちらかの Add をクリックして、特別な名前と値のペアを指定します。このタブでは、複数の名前と値のペアを指定できます。  
サポートされるジョブプロパティの詳細については、選択した Hadoop プラグインのドキュメントを参照してください。
13. Launch をクリックします。

起動したジョブのステータスを表示するには、プロジェクト>Data Processing>Jobsを選択します。起動済みのジョブの再起動または削除方法については、「[起動したジョブの削除または再起動](#)」を参照してください。

### 10.3. 起動したジョブの削除または再起動

起動したジョブのステータスを表示するには、プロジェクト > Data Processing > Jobsを選択します。ここから、ジョブを削除または再起動できます。

起動したジョブを削除するには、アクションドロップダウンメニューからジョブ実行の削除を選択します。チェックボックスをオンにし、Delete job executionsボタンをクリックして、起動した複数のジョブを削除することもできます。

既存のクラスターでジョブを再起動するには、アクションドロップダウンメニューから既存のクラスターで再起動を選択します。続行する方法は、「[既存クラスターでのジョブの起動](#)」を参照してください。

または、完全に新規クラスターでジョブ実行を再起動することもできます。これを行うには、ActionドロップダウンメニューからRelaunch on New Clusterを選択します。続行する方法は、「[新規クラスターでのジョブの起動](#)」を参照してください。

## 付録A インスタンスのローカリティの有効化

クラスターのパフォーマンスは、同じノードにローカルのボリュームのみを使用する場合に大幅に改善できます。クラスターに対して、ノードグループテンプレートを介してこのようなボリュームのみを使用するように強制できます。具体的には、`volume_local_to_instance` プロパティを `True` に設定することで行います。

このプロパティを使用する前に、Block Storage サービスで対応するフィルターを有効にする必要があります。そのためには、`/etc/cinder/cinder.conf` の有効なフィルタースケジューラーの一覧に `InstanceLocalityFilter` を含める必要があります。

```
scheduler_default_filters =
AvailabilityZoneFilter,CapacityFilter,CapabilitiesFilter,InstanceLocalityFilter
```

`director` を使用してこれを設定するには、以下を含む環境ファイルをデプロイメントに追加します。

```
parameter_defaults:
  ControllerExtraConfig: # 1
  cinder::config::cinder_config:
    DEFAULT/scheduler_default_filters:
      value: 'AvailabilityZoneFilter,CapacityFilter,CapabilitiesFilter,InstanceLocalityFilter'
```

- 1 **ControllerExtraConfig:** フックとそのネストされているセクションを、既存の環境ファイルの `parameter_defaults:` セクションに追加することもできます。



### 注記

フィルタースケジューラーの詳細は、Storage Guide の [Configure Volumes are Allocated to Multiple Back Ends](#) を参照してください。

OpenStack Data Processing が有効なオーバークラウドのデプロイに関する詳細は、[2 章 インストールシステム](#) を参照してください。

`InstanceLocalityFilter` を有効にしたら、クラスターが同じノードに対してローカルのボリュームを使用するように強制できるようになりました。これを行うには、以下を行います。

1. コマンドラインから、利用可能なすべてのノードグループテンプレートの名前を一覧表示します。

```
$ sahara node-group-template-list
+-----+-----+
| name   | id           | ...
+-----+-----+
| mytemplate | 3d6b4b7c-bca7-4f3a-a6ae-621a31ab7a75 ...
+-----+-----+
```

2. テンプレートのプロパティを表示します。

```
$ sahara node-group-template-show --name mytemplate
+-----+-----+
| Property           | Value      |
+-----+-----+
| volume_local_to_instance | False     |
```

```

| volumes_availability_zone | None      |
| description                |          |
| availability_zone         |          |
| volume_mount_prefix      | /volumes/disk |
| updated_at                | None     |
...

```

ここで、`volume_local_to_instance` は `False` に設定されます。

- 以下の文字列を含む `setting.json` という名前のファイルを作成します。

```
{"volume_local_to_instance": true}
```

- ノードグループテンプレートを `setting.json` の文字列で更新します。

```
$ sahara node-group-template-update --json setting.json --id 3d6b4b7c-bca7-4f3a-
a6ae-621a31ab7a75
```

```

+-----+-----+
| Property          | Value      |
+-----+-----+
| volume_local_to_instance | True      |
| volumes_availability_zone | None     |
| description        |          |
| availability_zone   |          |
| volume_mount_prefix | /volumes/disk |
| updated_at         | None     |
...

```

これを行うと、`volume_local_to_instance` プロパティを `True` に設定します。