



# Red Hat OpenStack Platform 11 OpenStack Data Processing

---

Red Hat OpenStack Platform における Hadoop クラスターの手動による  
プロビジョニングとサイズ調整

OpenStack Team



## Red Hat OpenStack Platform における Hadoop クラスターの手動による プロビジョニングとサイズ調整

OpenStack Team  
rhos-docs@redhat.com

## 法律上の通知

Copyright © 2017 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, 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 Software Collections 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

OpenStack Data Processing 機能は、Hadoop クラスターを容易にプロビジョニングおよびサイズ調整して、大量のデータセットを処理することができます。本ガイドでは、Data Processing の必須コンポーネント (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、クラスター上でのデータ処理、必要に応じたクラスターのサイズ調整などを含む OpenStack Data Processing の全ワークフローを順を追って説明します。本リリースの OpenStack Data Processing には、ガイド タブが含まれています。このタブは、クラスターを起動してそこでジョブを実行するのに必要なテンプレートの作成に役立つウィザードの機能を提供しますが、本ガイドでは、OpenStack Data Processing ワークフローをさらに深く考察し、ガイド タブ機能を使用しない場合のテンプレート作成およびコンポーネント登録の手順を順を追って説明することを目的としています。OpenStack Data Processing 機能には、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。また、ユーザーは、選択した Hadoop プラグインに十分精通している必要があります。本リリースでは、Cloudera 5.5.0 のプラグインをサポートしています。

---

目次

1. 概要 .....	2
2. インストール .....	2
3. ワークフロー .....	3
4. HADOOP イメージの作成 .....	4
5. 必要なコンポーネントの登録 .....	6
5.1. 入力および出力データソースの登録 .....	6
6. ノードグループテンプレートの設定 .....	7
7. クラスターテンプレートの設定 .....	8
8. クラスターの起動 .....	9
8.1. クラスターのサイズ調整または削除 .....	10
9. ジョブの設定 .....	11
9.1. ジョブバイナリー、スクリプト、ライブラリーの登録 .....	11
9.2. ジョブテンプレートの作成 .....	12
10. ジョブの起動 .....	12
10.1. 既存のクラスター上でのジョブの起動 .....	13
10.2. 新規クラスター上でのジョブの起動 .....	13
10.3. 起動したジョブの削除または再起動 .....	14
A. インスタンスのローカリティーの有効化 .....	15

## 1. 概要

OpenStack Data Processing は、OpenStack Sahara プロジェクトにより提供される機能で、Hadoop クラスターを容易にプロビジョニングおよびサイズ調整するための堅牢なインターフェースを実装します。Hadoop クラスターは、通常、大量のデータセットの処理など、リソースを集中的に使用するジョブの実行に利用することができます。OpenStack Data Processing は、OpenStack コンポーネントとして OpenStack エコシステムに完全に統合されています。たとえば、クラスターの設定からそのクラスター上でのジョブの起動/実行に至るまで、OpenStack Dashboard を使用して、Hadoop データ処理のワークフロー全体を管理することができます。

Hadoop に関する詳しい情報は、<http://hadoop.apache.org/> を参照してください。

OpenStack Data Processing は異なるプラグインを使用して、各 Hadoop ディストリビューションの固有のクラスターをプロビジョニングします。クラスターの設定の際に利用可能な Hadoop パラメーターは、Hadoop ディストリビューション (およびプラグインが使用する拡張機能) により異なります。本リリース (Red Hat OpenStack Platform 11) の時点では、OpenStack Data Processing は以下のプラグインをサポートします。

- ✦ [Cloudera \(CDH\) 5.5.0](#)
- ✦ [Hortonworks Data Platform](#)
  - [バージョン 2.3](#)
  - [バージョン 2.4](#)
- ✦ [MapR 5.1](#)

OpenStack Data Processing の本リリースには、**ガイド** タブも含まれています。このタブは、クラスターを起動してそこでジョブを実行するのに必要なテンプレートの作成に役立つウィザードの機能を提供します。ただし、OpenStack Data Processing を使用するのに必要なコンポーネント (例: Hadoop イメージ、ジョブバイナリーなど) を登録する必要もあるので、**ガイド** 機能を使用する場合には、最初に「[必要なコンポーネントの登録](#)」の項を一読しておくことをお勧めします。

## 2. インストール

本項では、director を使用したオーバークラウドへの OpenStack Data Processing のデプロイを想定しています。また、**stack** と呼ばれる **director** をインストールするユーザーを作成することも前提です。stack というユーザーは、『[director のインストールと使用方法](#)』の「[director のインストールユーザーの作成](#)」で作成したものと同じです。

director はデプロイメント時に **環境ファイル** を使用して、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、各サービスのオーケストレーションに使用する Heat テンプレートと Puppet スクリプトと一緒に `/usr/share/openstack-tripleo-heat-templates/` に保存されます。

オーバークラウド上の OpenStack Data Processing サービスは、デフォルトでは無効になっています。これは、OpenStack Data Processing コンポーネントが主要なオーバークラウドの環境ファイル (`/usr/share/openstack-tripleo-heat-templates/overcloud-resource-registry-puppet.yaml`) で null 操作 (`OS::Heat::None`) として登録されているためです。

```
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化するには、`/usr/share/openstack-tripleo-heat-templates/puppet/services` ディレクトリーの適切な 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。サービスによっては、`environments` ディレクトリーに事前定義された環境ファイルが含まれます。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: ../../puppet/services/sahara-api.yaml
  OS::TripleO::Services::SaharaEngine: ../../puppet/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、以下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

**instance locality** を有効化して、同じノードにローカルなボリュームをクラスターが使用するよう強制することも可能です。このように設定すると、クラスターのパフォーマンスを向上させることができます。関連情報については、「[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、不要な変更がオーバークラウドに加えられないように、`-e` オプションを使用して環境ファイルをここで再度渡します。詳しい情報は、『[director のインストールと使用方法](#)』の「[オーバークラウド環境の変更](#)」を参照してください。

## 3. ワークフロー

OpenStack Data Processing は、Hadoop クラスターのプロビジョニングおよびサイズ調整に、事前設定済みのテンプレートを使用して、特別に設計されたインスタンスを定義します。これらのインスタンスにより、個別ノードが形成され、Hadoop クラスターが構成されます。この 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することができます。

OpenStack Data Processing を使用する場合は、Hadoop フレームワーク内で作業するのに必要とされるコンポーネントに精通している必要があります。このように、本セクションに記載の一般的なワークフローは、以下のコンポーネントの準備が整っていることを前提としています。

- ※ Hadoop イメージ。特に Hadoop データ処理プラグインが含まれる Red Hat Enterprise Linux イメージ。サポートされるプラグインの一覧は、「[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ていることが望ましい)。
- ※ 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ていることが望ましい)。



## 注記

Object Storage サービスへコンテンツをアップロードする詳しい方法は、「[オブジェクトのアップロード](#)」を参照してください。

また、ジョブの実行に必要なコンピュートリソースに関する一般知識も必要です。このような知識は、必要なノードタイプ (各タイプに必要とされる個数) を判断するのに役立ちます。

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

1. OpenStack Data Processing に必要なプラグインのコンポーネントが含まれるイメージを作成します (「[Hadoop イメージの作成](#)」)。これが今後使用する Hadoop イメージです。

このイメージの作成手順は、選択した Hadoop プラグインにより左右されます。

2. OpenStack Data Processing サービスに必要な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 (例: 入力データおよび出力データの出力先)

3. ノードグループのテンプレートを作成します。各テンプレートは、任意のノードに Hadoop 固有の役立つ設定を多数定義します。特に以下の設定が重要となります。

- ※ ノードグループが使用する必要のある Hadoop プラグインとバージョン

- ※ ノード上で実行すべきプロセス

4. クラスターの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、以下の項目をはじめとするさまざまな設定を定義します。

- ※ **ノードグループの構成** クラスターを構成する各ノードグループ内のノード数

- ※ **クラスタースコープの Hadoop 設定** 各 Hadoop コンポーネントに設定する必要のある固有のパラメーター (HIVE、AMBARI、HDFS など)

5. クラスターテンプレートを使用して Hadoop クラスターを起動し、クラスター上でジョブを実行します (データソース上で登録済みのジョブバイナリーを実行します)。また、必要に応じて、クラスターのサイズを調節することも可能です (タイプを問わずノードを追加または削除することが可能です)。

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 サービスに登録して、ジョブを作成し、Hadoop クラスターでジョブを起動します。ジョブにより、どのジョブバイナリー、スクリプトまたはライブラリーを使用して、登録済みのデータソースを処理するかが定義されます。

以下のセクションでは、各ワークフローの手順を詳しく説明します。

## 4. HADOOP イメージの作成

今回のリリースでは、以下のプラグインがサポートされます。

- ※ [Cloudera \(CDH\) 5.5.0](#)

- ※ Hortonworks Data Platform



- バージョン 2.3

- バージョン 2.4

- ※ MapR 5.1

これらのプラグインのいずれかをベースにして、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 のサブスクリプションが必要)。

[https://access.redhat.com/downloads/content/69/ver=/rhel---7/7.2/x86\\_64/product-software](https://access.redhat.com/downloads/content/69/ver=/rhel---7/7.2/x86_64/product-software)

イメージのダウンロード後には、システムが Red Hat サブスクリプションの詳細を使用できるように設定します。これにより、ダウンロードした Red Hat Enterprise Linux イメージから Hadoop イメージを作成するのに必要なツールをさらにダウンロードすることができるようになります。ターミナルから (root として) 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
```

上記の設定で、

- ※ **USERNAME** と **PASSWORD** は Red Hat サブスクリプションの認証情報に置き換えてください。
- ※ **POOLID** には Red Hat サブスクリプションのプール ID を指定します。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
```

このコマンドにより、**プール ID** など、お使いのシステムのサブスクリプションの重要な情報が表示されます。



#### 注記

Red Hat のサブスクリプションを手動で設定する詳しい方法については、『[Using and Configuring Red Hat Subscription Manager](#)』を参照してください。

次に、**sahara-image-elements** パッケージをインストールします。

```
# yum install sahara-image-elements -y
```

最後に、Hadoop イメージを作成します。Hadoop イメージの作成には、Red Hat Enterprise Linux イメージと同じディレクトリーに移動して、以下のコマンドを実行します。

```
# sahara-image-create -p PLUGIN
```

**PLUGIN** は Hadoop イメージの使用するプラグインに置き換えます。以下の値がサポートされています。

- ※ **cloudera** ([Cloudera \(CDH\) 5.5.0](#))
- ※ **mapr** ([MapR 5.1](#))
- ※ **ambari** (Hortonworks Data Platform)

- OpenStack Data Processing は、Hortonworks Data Platform プラグインの 2 つのバージョンをサポートします。コマンドを実行する際に、**-v 2.3** ([バージョン 2.3](#)) または **-v 2.4** ([バージョン 2.4](#)) を追加するようにしてください。

**sahara-image-create** コマンドは、プラグインの名前を含むイメージを作成します (例 **cloudera** の場合は **sahara-rhel-7-2-cdh-5-5.qcow2** です)。

## 5. 必要なコンポーネントの登録

OpenStack Data Processing は、Hadoop クラスターの起動および使用に必要な要素が含まれる Hadoop イメージが必要です。特に、Red Hat OpenStack Platform には、Red Hat Enterprise Linux と必要なデータ処理プラグインが含まれるイメージが必要です。

実行するジョブに適した Hadoop イメージが用意できたら、OpenStack Data Processing サービスにそのイメージを登録します。登録の方法は以下のとおりです。

1. イメージを Image サービスにアップロードします。Image サービスへのアップロードの方法は、「[イメージのアップロード](#)」を参照してください。
2. イメージをアップロードした後は、Dashboard 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. **イメージの登録** をクリックして、イメージのドロップダウンメニューから Hadoop のイメージを選択します。
4. 設定の適用および各インスタンス/ノード上のプロセス管理の際に、OpenStack Data Processing サービスが使用すべきユーザー名を入力します。「[Hadoop イメージの作成](#)」で使用した、Red Hat Enterprise Linux が提供する正式なイメージでは、上記の目的で使用されるユーザー名は **cloud-user** です。
5. デフォルトでは、OpenStack Data Processing サービスにより必要なプラグインとバージョンのタグが **プラグイン** と **バージョン** のドロップダウンメニューに追加されます。選択されているタグが正しいことを確認してから **プラグインタグの追加** をクリックしてそれらのタグを追加します。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ができます。タグを追加するには、説明のフィールドの下欄に表示されている **カスタムタグの追加** ボタンを使用します。

カスタムタグを削除するには、削除するタグの名前の横にある **x** をクリックします。

6. **完了** をクリックします。イメージが**イメージレジストリー**の表に表示されるはずです。

### 5.1. 入力および出力データソースの登録

イメージの登録の後には、データの入力元および出力先を登録します。これらはいずれも、Object Storage サービスからオブジェクトとして登録することができます。そのためには、最初に入力元も出力先もオブジェクトとしてアップロードする必要があります。手順は、「[オブジェクトのアップロード](#)」を参照してください。

#### 注記

Hadoop と互換性のある別の分散ファイルシステム (例: HDFS) からデータオブジェクトを直接登録することも可能です。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、対象のファイルシステム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 > データ処理 > データソース** を選択します。
2. **データソースの作成** をクリックします。 **名前** のフィールドにデータソースの名前を入力します。
3. **説明** のフィールドで、データソースの説明を入力します (任意)。
4. データソースの種別と URL を選択します。この手順は、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には、**データソース種別** のドロップダウンメニューから **Swift** を選択して、以下のステップを実行します。

※ **URL** フィールドで、データソースのコンテナおよびオブジェクト名を **swift://CONTAINER/OBJECT** として指定します。

※ データソースにログインが必要な場合は、**データソースのユーザー名** および **データソースのパスワード** フィールドに必要な認証情報を入力します。

※ データが Hadoop Distributed File System (HDFS) に保存されている場合には、データソース種別のドロップダウンメニューから対象の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** の形式で入力します。ここでは、

- **HDFSHOST** は、HDFS ホストのホスト名に置き換えます。
- **PORT** は、データソースがアクセス可能なポートに置き換えます。
- **OBJECTPATH** は **HDFSHOST** のデータソースへの利用可能なパスに置き換えます。

5. **完了** をクリックします。これで、このデータソースは、**データソース** の表から利用できるようになったはずです。

対象のジョブに必要なデータの入力/出力オブジェクトそれぞれに対して、上記の手順を実行してください。

## 6. ノードグループテンプレートの設定

Hadoop を使用したデータ処理では、クラスター上でジョブが実行されます。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) により構成されます。OpenStack Data Processing では、テンプレートを使用して各クラスターに必要な各種ノードを定義できます。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。以下の手順は、ノードグループテンプレートの設定方法を説明します。

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. **テンプレートの作成** をクリックします。
3. **プラグイン名** と **バージョン** のドロップダウンメニューを使用して、ノードで使う Hadoop プラグインの名前とバージョンを選択します。
4. **作成** をクリックします。
5. **テンプレート名** フィールドに、テンプレートの名前を入力します。
6. 作成するノードグループテンプレートの説明を追加するには、**説明** フィールドを使用します (任意)。

7. **OpenStack フレーバー** ドロップダウンメニューから、ノードが使用する必要のあるフレーバーを選択します。このフレーバーは、ノードのコンピューティング、メモリー、ストレージ要件に適している必要があります。フレーバーに関する詳細情報は、「[フレーバーの管理](#)」を参照してください。
  8. **ストレージの場所** の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、対象のノードが必要とする一時ストレージを生成します。
    - ✧ **Cinder ボリューム**: このオプションでは、Block Storage サービスがノードごとに指定のボリューム数を作成するように設定することができます。ボリューム数を指定するには「ノードあたりのボリューム数」フィールドを、各ボリュームのサイズを指定するには「ボリュームサイズ (GB)」フィールドを使用してください。

一時ドライブ上のデータは、ホストの整合性に左右されるので、ホストに問題があった場合に影響を受けやすいため、大半のデプロイメントには、**Cinder ボリューム** オプションを推奨します。
  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 を使用すべきかどうか、使用する場合にはどれを使用するのかを設定します。Floating IP に関する詳しい情報は、「[IP アドレスの設定](#)」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、**自動セキュリティーグループ** チェックボックスを選択します。**セキュリティーグループ** の一覧の該当するチェックボックスにチェックを入れると、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に関する詳しい説明は、「[プロジェクトの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必要がある Hadoop プロセスを選択します。これには、**プロセス** の一覧から選択する各プロセスのチェックボックスにチェックを入れます。各プロセスの詳細情報は、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. **作成** をクリックします。テンプレートが**ノードグループテンプレート**の表に表示されるはずです。

**ノードグループテンプレート**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、対応する Hadoop プラグイン、およびバージョンが一覧表示されます。またこの表には、各テンプレートのノード上で実行される Hadoop プロセスも表示されます。

## 7. 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て（「[必要なコンポーネントの登録](#)」）、必要なノードの種別を設定した後は（「[ノードグループテンプレートの設定](#)」）、クラスターのテンプレートを設定することができます。クラスターのテンプレートは、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構成 (各ノード種別をいくつ使用して構成されているか)。利用可能なノード種別は **プロジェクト > データ処理 > ノードグループテンプレート** に一覧表示されます。
- ✧ クラスターで使用する Hadoop プラグインおよびバージョン
- ✧ 単一ホストで1度しか起動すべきでないプロセス一覧 (**anti-affinity**)
- ✧ Hadoop 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の作成/設定は、以下の手順に従って行います。

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. **プラグイン名** と **バージョン** のドロップダウンメニューを使用して、クラスターで使う Hadoop プラグインの名前とバージョンを選択します。
  3. **作成** をクリックします。
  4. **テンプレート名** フィールドに、テンプレートの名前を入力します。
  5. 作成するクラスターテンプレートの説明を追加するには、**説明** フィールドを使用します (任意)。
  6. **anti-affinity グループ** を使用するプロセス一覧から、1 台のホスト上で 2 回以上起動すべきでないプロセスのチェックボックスをそれぞれ選択します。
  7. ノードグループのタブをクリックします。ここから、クラスターで実行する必要があるノードの数をノード種別ごとに設定します。これは、以下のステップに従って行います。
    - a. **追加するノードグループテンプレート** を選択してくださいの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 (この手順に関する説明は、「[「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、テンプレートをベースにしてノードグループのエントリーを作成します。クラスターテンプレートに使用するノードグループテンプレートごとに、この作業を実行してください。
    - c. ノードグループのエントリーごとに、そのノード種別のインスタンスをクラスター内でいつ起動する必要があるか設定します。
- また、ノードグループエントリーごとに名前を入力することもできます (任意)。
8. 必要に応じて、コンポーネントごとに Hadoop クラスターの設定を行います。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、関連のパラメーターが含まれており、そのパラメーターの多くに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しい情報は、選択した Hadoop プラグインのマニュアルで、各コンポーネントのサポートする機能についての説明を参照してください。
  9. 対象のクラスターテンプレートの設定に問題がなければ、**作成** をクリックします。テンプレートは、クラスターテンプレートの表に表示されるはずです。

または、**テンプレートのアップロード** 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します。

クラスターテンプレートの表にある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可能です。これには、クラスターテンプレートの **アクション** のドロップダウンメニューから **クラスターの起動** を選択します。起動したクラスターを表示するには **プロジェクト > データ処理 > クラスター** を選択します。クラスターの起動に関する詳しい情報は、「[クラスターの起動](#)」を参照してください。

## 8. クラスターの起動

クラスターテンプレートの設定 ([「クラスターテンプレートの設定」](#)) が完了したら、クラスターを起動することができます。クラスターを起動するには、**クラスターの起動** ウィザードを開きます。このウィザードは、以下のいずれかの方法で開くことができます。

- ※ **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、次にテンプレートの **アクション** のドロップダウンメニューから **クラスターの起動** を選択します。



- ※ プロジェクト > データ処理 > クラスターを選択します。そこから、**クラスターの起動**をクリックし、**プラグイン名**と **Hadoopバージョン** のドロップダウンメニューを使用して、クラスターで使う Hadoop プラグインの名前とバージョンを選択します。

次に **作成** をクリックします。クラスターの**起動**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、**クラスター名** フィールドにクラスターの名前を入力します。
2. 起動するクラスターの説明を追加するには、**説明** フィールドを使用します (任意)。
3. プロジェクト > データ処理 > クラスターの **クラスター** の表から **クラスターの起動** ウィザードを開いた場合は、**クラスターテンプレート** の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** を選択することができます。上記の手順で選択した 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することができます。
4. ベースイメージのドロップダウンメニューから、クラスターが使用すべき Hadoop イメージを選択します。Hadoop イメージの作成/登録に関する詳細は「[Hadoop イメージの作成](#)」および「[必要なコンポーネントの登録](#)」を参照してください。
5. 必要であれば、**キーペア** 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、新規キーペアを作成することも可能です。クラスターの起動には、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際に必要です (例: SSH を使用する場合など)。

キーペアに関する情報は、「[キーペアの管理](#)」を参照してください。

6. **Neutron 管理ネットワーク** ドロップダウンメニューからクラスターが使用すべき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「[一般的な管理タスク](#)」を参照してください。
7. **作成** 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 **プロジェクト > データ処理 > クラスター** を選択します。

## 8.1. クラスターのサイズ調整または削除

OpenStack Data Processing では、リソースのニーズに合わせて既存のクラスターのサイズを調整することができます。サイズを調節することで、実行中のクラスターからどの種別のノードを (いくつでも) 追加または削除することができます。

起動したクラスターを表示するには **プロジェクト > データ処理 > クラスター** を選択します。このページからクラスターのサイズ調整も削除も可能です。クラスターの起動方法については、「[クラスターの起動](#)」を参照してください。

クラスターのサイズ調整の方法:

1. **クラスター** の表で、サイズ調整するクラスターを選択します。次に、そのクラスターの**アクション** ドロップダウンメニューから **クラスターのサイズ調整** を選択します。必要に応じてクラスターをサイズ調整します。
  - ※ 新しいノード種別をクラスターに追加するには、**追加するノードグループテンプレート** を選択してくださいのドロップダウンメニューからテンプレートを選択します。次に、**+** ボタンをクリックして、クラスターにノード種別を追加し、その種別のノードをいくつ追加するかを設定することができます。

- ※ 既存のノードグループにノードを追加したり、ノードグループからノードを削除したりするには、ノードグループの行の + または - ボタンを使用します。または、単に、ノードグループの数 フィールドにノード数を設定することもできます。

## 2. サイズ調整 をクリックします。

クラスターを削除するには **アクション** ドロップダウンメニューから **クラスターの削除** を選択します。削除するクラスターのチェックボックスを選択して **クラスターの削除** ボタンをクリックすることで、複数のクラスターを削除することも可能です。

# 9. ジョブの設定

OpenStack Data Processing サービスでは、ジョブにより実際のデータ処理タスクが定義されます。各ジョブは、ジョブの種別 (例: Pig、Hive または MapReduce)、バイナリー、スクリプト、ライブラリーを指定します。ただし、OpenStack Data Processing サービスに登録されたバイナリー、スクリプトまたはライブラリーしか使用することができません。

ジョブの作成後には、クラスター上でジョブを起動して、入力データソースに対してジョブを実行します。ジョブバイナリーなどの入力/出力データソースは、まず OpenStack Processing サービスに登録しておく必要があります (「[入力および出力データソースの登録](#)」を参照)。

## 9.1. ジョブバイナリー、スクリプト、ライブラリーの登録

ジョブバイナリー、スクリプト、ライブラリーの登録プロセスは、イメージおよびデータソースの登録方法に似ています。Object Storage サービスから直接登録することができます。Object Storage サービスへのオブジェクトのアップロードの方法は、「[オブジェクトのアップロード](#)」を参照してください。または、バイナリーおよびライブラリーを、ローカルファイルシステムから直接 OpenStack Data Processing サービスにアップロードすることも可能です。

1. Dashboard で **プロジェクト > データ処理 > ジョブバイナリー** 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 (例: スクリプト、バイナリー、またはライブラリーなどの名前)。この名前は、オブジェクトの選択時に使用します。オブジェクトが特定の名前または拡張子 (例: **.jar**) を必要とする場合には、ここに追加します。
4. 登録するスクリプト、バイナリー、ライブラリーの説明を追加するには、**説明** フィールドを使用します (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 サービスにある場合は、ストレージ種別のドロップダウンメニューから **Swift** を選択して、以下のステップを実行します。
    - ※ URL のフィールドで、スクリプト、バイナリー、ライブラリーのコンテナとオブジェクト名を **swift://CONTAINER/OBJECT** として指定します。
    - ※ お使いのスクリプト、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、**ユーザー名** および **パスワード** フィールドに必要な認証情報を入力します。
  - b. 必要でない場合は、「ストレージ種別」のドロップダウンメニューから「内部データベース」を選択して、次に「内部バイナリー」ドロップダウンメニューを使用して以下の操作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。
  - ※ Dashboard に直接スクリプトを入力します (スクリプトを作成)。
  - ※ バイナリー、ライブラリー、またはスクリプトを直接ローカルのファイルシステムにアップロードします (新しいファイルをアップロード)。
6. **作成** 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ジョブバイナリーの表で利用できるようになります。

## 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、ライブラリーが OpenStack Data Processing に登録されたら、以下の手順を実行してください。

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. **名前** フィールドにジョブの名前を入力します。
4. **ジョブ種別** のドロップダウンメニューから正しい種別を選択します。ジョブ種別に関する詳しい情報は、選択したプラグインのドキュメント内のサポートされるジョブ種別を参照してください。
5. **メインバイナリー** を選択してくださいのドロップダウンメニューから、このジョブに使用すべき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 サービスで登録したジョブバイナリーおよびスクリプトで生成されます。詳しくは、「[ジョブバイナリー、スクリプト、ライブラリーの登録](#)」を参照してください。
6. 作成するジョブの説明を追加するには、**説明** フィールドを使用します (任意)。
7. 指定したジョブバイナリーがライブラリーを必要とする場合には、ライブラリーを追加してください。これには、**ライブラリー** タブをクリックして、**ライブラリーの選択** 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、**選択** をクリックして、ライブラリーをジョブに追加します。ライブラリーは **ライブラリーの選択** 一覧に含まれているはずです。ジョブバイナリーで必要なライブラリーすべてに対して、この作業を繰り返してください。詳しい説明は、「[ジョブバイナリー、スクリプト、ライブラリーの登録](#)」を参照してください。
8. **作成** 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ジョブバイナリーの表で利用できるようになります。

## 10. ジョブの起動

ジョブの作成後には、ジョブを起動して OpenStack Data Processing サービスで登録したデータを処理します (「[入力および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 (existing cluster (「[既存のクラスター上でのジョブの起動](#)」)) または全く新しいクラスター (「[新規クラスター上でのジョブの起動](#)」) でジョブを起動することができます。

### 注記

ジョブを起動するには、入力データソースおよび出力データの宛先を指定する必要があります。最初に、両オブジェクトを OpenStack Data Processing サービスに登録しておく必要があります。詳しい情報は、「[入力および出力データソースの登録](#)」を参照してください。



## 10.1. 既存のクラスター上でのジョブの起動

Dashboard で既存のクラスターの一覧を表示するには、**プロジェクト > データ処理 > クラスター**を選択します。クラスターの起動方法に関する情報は、「[クラスターの起動](#)」を参照してください。

既存のクラスターでジョブを起動する方法:

1. Dashboard で、**プロジェクト > データ処理 > ジョブ**を選択します。ジョブの表には、利用可能なジョブテンプレートがすべて表示されます。新規ジョブテンプレートの作成方法「[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、そのジョブテンプレートの **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起動**を選択します。
3. **ジョブの起動** ウィザードで、**入力** 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、**出力** ドロップダウンメニューから出力先を選択します。

必要な場合には、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可能です。そのためには、**入力** または **出力** のドロップダウンメニューで+をクリックします。これで、**データソースの作成** ウィザードが開きます。詳しい説明は、「[入力および出力データソースの登録](#)」を参照してください。

4. **クラスター** ドロップダウンメニューから、ジョブの実行先の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ーを設定する必要がある場合には、**設定** タブをクリックします。そのタブで、**設定** または **パラメーター** のいずれかの下にある **追加** をクリックして、特別な名前/値のペアを指定します。このタブから、名前/値のペアを複数指定することができます。

サポートされるジョブプロパティーに関する詳しい情報は、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. **起動** 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 > データ処理 > ジョブ**を選択します。起動したジョブの再起動または削除の方法は、「[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. 新規クラスター上でのジョブの起動

ジョブテンプレートの作成後には、そのテンプレートを使用して新しいクラスターでジョブを起動することも可能です。新規クラスターで実行することで、ジョブの完了後に、クラスターを自動で終了することも可能になります。

1. Dashboard で、**プロジェクト > データ処理 > ジョブ**を選択します。ジョブの表には、利用可能なジョブがすべて表示されます。新規ジョブの作成方法についての詳しい説明は、「[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、そのジョブテンプレートの **アクション** ドロップダウンメニューから **新規のクラスターで起動**を選択します。
3. プラグインの **名前** と **バージョン** のドロップダウンメニューを使用して、ジョブで使う Hadoop プラグインの名前とバージョンを選択します。
4. **作成** をクリックします。
5. **クラスター名** フィールドに、クラスターの名前を入力します。

6. ベースイメージのドロップダウンメニューから、クラスターが使用すべき Hadoop イメージを選択します。Hadoop イメージの作成/登録に関する詳細は「[Hadoop イメージの作成](#)」および「[必要なコンポーネントの登録](#)」を参照してください。
7. 必要であれば、キーペア 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、新規キーペアを作成することも可能です。クラスターの起動には、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際に必要です (例: SSH を使用する場合があります)。

キーペアに関する情報は、「[キーペアの管理](#)」を参照してください。

8. **Neutron 管理ネットワーク** ドロップダウンメニューからクラスターが使用すべき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「[一般的な管理タスク](#)」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、ジョブの完了後にはクラスターを削除します。削除されないようにするには、**ジョブ終了後もクラスターを保持する** チェックボックスにチェックを入れます。
10. 次に、**ジョブ** タブをクリックして、**入力** 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た後に、**出力** ドロップダウンメニューから出力先を選択します。
11. 必要な場合には、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可能です。そのためには、**入力** または **出力** のドロップダウンメニューで + をクリックします。これで、**データソースの作成** ウィザードが開きます。詳しい説明は、「[入力および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ーを設定する必要がある場合には、**設定** タブをクリックします。そのタブで、**設定** または **パラメーター** のいずれかの下にある **追加** をクリックして、特別な名前/値のペアを指定します。このタブから、名前/値のペアを複数指定することができます。

サポートされるジョブプロパティーに関する詳しい情報は、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. **起動** 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 > データ処理 > ジョブ** を選択します。起動したジョブの再起動または削除の方法は、「[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 > データ処理 > ジョブ** を選択します。ここで、ジョブの削除または再起動を行います。

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。起動したジョブを複数、削除するには、削除するジョブのチェックボックスを選択して **クラスターの削除** ボタンをクリックします。

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。その後の操作手順については、「[既存のクラスター上でのジョブの起動](#)」を参照してください。

または、全く別の新規クラスターで、ジョブ実行を再起動することもできます。これには、**アクション** ドロップダウンメニューから **新規のクラスターで再起動** を選択します。その後の設定方法については、「[新規クラスター上でのジョブの起動](#)」を参照してください。

## A. インスタンスのローカリティの有効化

クラスターが、同じノードにローカルなボリュームのみを使用すると、そのクラスターのパフォーマンスを大幅に向上させることができます。ノードグループテンプレートを使用して **volume\_local\_to\_instance** プロパティを **True** に設定することによって、クラスターがそのようなボリュームのみを使用するように強制することができます。

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** で有効化されたフィルタースケジューラーの一覧に **InstanceLocality** が含まれていなければなりません。

```
scheduler_default_filters =
AvailabilityZoneFilter,CapacityFilter,CapabilitiesFilter,InstanceLocality
```

これを **director** で設定するには、以下の内容を記載した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,InstanceLocality'
```

1

**ControllerExtraConfig:** フックとそのネストされた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可能です。

### 注記

フィルタースケジューラーに関する情報は、「[ボリュームを複数のバックエンドに割り当てる方法の設定](#)」(『[ストレージガイド](#)』)を参照してください。

OpenStack Data Processing を有効化したオーバークラウドのデプロイについての詳しい情報は、「[インストール](#)」を参照してください。

**InstanceLocality** フィルターを有効化した後には、クラスターが同じノードにローカルのボリュームを使用するように強制できます。そのためには、以下のステップを実行します。

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確認します。

```
$ sahara node-group-template-show --name mytemplate
```

Property	Value
volume_local_to_instance	<b>False</b>
volumes_availability_zone	None
description	
availability_zone	
volume_mount_prefix	/volumes/disk
updated_at	None
...	

ここでは、**volume\_local\_to\_instance** が **False** に設定されています。

3. **setting.json** という名前のファイルを作成して、以下の文字列を記述します。

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

4. **setting.json** の文字列を適用して、ノードグループテンプレートを更新します。

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

Property	Value
volume_local_to_instance	<b>True</b>
volumes_availability_zone	None
description	
availability_zone	
volume_mount_prefix	/volumes/disk
updated_at	None
...	

この操作により、**volume\_local\_to\_instance** プロパティが **True** に設定されます。