



Red Hat OpenStack Platform 15

OpenStack Data Processing

Red Hat OpenStack Platform での native クラスターの手動でのプロビジョニングと
スケーリング

Red Hat OpenStack Platform 15 OpenStack Data Processing

Red Hat OpenStack Platform での native クラスターの手動でのプロビジョニングとスケーリング

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[®] 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

OpenStack Data Processing 機能により、大規模なデータセットの処理を簡単にプロビジョニングおよびスケールリングできます。本書では、Data Processing 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、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ルリングなど、OpenStack Data Processing ワークフロー全体について説明します。本リリースの 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、それらでジョブを実行するために必要なテンプレートの作成に役立ちます。本書の目的は、OpenStack Data Processing ワークフローをより詳細に確認できるようにすることです。そのため、Guides タブ機能を使用せずにテンプレートの

作成およびコンポーネント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の基本知識が必要です。さらに、ユーザーは選択した Hadoop プラグインの特定の点に精通している必要があります。

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 MAKEFORM 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. 入力および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. 起動ジョブを削除または再起動	21
付録A インスタンスの局所性の有効化	23

第1章 概要

OpenStack Data Processing サービス(sahara)は、easy クラスタ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強力なインターフェースを提供します。その後、このようなクラスタは、通常は大規模なデータセットを処理するのに、リソース集約型のジョブを実行するために使用できます。OpenStack のコンポーネントとして、OpenStack Data Processing は OpenStack のエコシステムに完全に統合されます。たとえば、ユーザーは OpenStack ダッシュボードでクラスタを構成することで、クラスタを設定し、それらでジョブを起動し、実行するすべての方法により、個々のデータ処理ワークフロー全体を管理できます。

Hadoop 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 で非推奨となり、バージョン 16 で削除される予定です。

OpenStack Data Processing は、各 Hadoop ディストリビューションの特定クラスタ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（およびその拡張で使用されるプラグイン）によって異なります。本リリース(Red Hat OpenStack Platform 15)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)
 - [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)
 - [バージョン 5.13.x](#) (デフォルトは 5.13.0)
- Hortonworks Data Platform
 - [バージョン 2.3](#)
 - [バージョン 2.4](#)
- [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、クラスタを起動し、それらでジョブを実行するために必要なテンプレートの作成に役立ちます。ただし、ovn イメージやジョブバイナリーなどの OpenStack Data Processing を使用するために必要なコンポーネントを引き続き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 を使用してオーバークラウドに OpenStack Data Processing をデプロイしていることを前提としています。また、director のインストールユーザーの作成と同様に、**stack** という名前の **director** の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 のインストールと使用方法](#)』）。

director は、デプロイメント中に **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、`/usr/share/openstack-tripleo-heat-templates/` とともに、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 コンポーネントが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にするには、`/usr/share/openstack-tripleo-heat-templates/deployment/sahara` 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。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、以下の内容を含む `/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: ../../deployment/sahara/sahara-api-container-puppet.yaml
  OS::TripleO::Services::SaharaEngine: ../../deployment/sahara/sahara-engine-container-puppet.yaml
```

これにより、デフォルトの null 操作リソースが上書きされ、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**局所性**を有効にして、クラスターで同じノードにローカルのボリュームを強制的に使用することも可能です。これを実行すると、クラスターのパフォーマンスを向上する可能性があります。関連する information については、[付録A インスタンスの局所性の有効化](#)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。

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に設定されたクラスターテンプレートを使用して、ovn クラスターをプロビジョニングし、スケーリングします。これらのインスタンスは、Hadoop クラスターを構成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/binaries を実行することができます。

OpenStack Data Processing を使用する予定の場合には、すでに dn 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について理解しておく必要があります。そのため、本セクションに記載す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。

- Hadoop 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す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のバイナリーおよびライブラリー（Object Storage サービスにお好ましい）。



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、「オブジェクトの [アップロード](#)」を参照してください。

さらに、ジョブを実行するのに必要な計算リソースの一般的な概念もあるはずです。これにより、必要なノード種別（およびそれぞれの数）を判断しやすくなります。

以下の高レベルのワークフローでは、OpenStack Data Processing サービスを設定して使用し、クラスターを起動し、そ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makeform イメージの作成](#))に必要なプラグインコンポーネントが含まれるイメージを作成します。これは Hadoop イメージです。
このイメージの作成手順は、選択した Hadoop プラグインによって異なります。
2. 以下のコンポーネントを OpenStack Data Processing サービスに登録します。
 - Hadoop イメージ
 - データソース（入力データと出力データが移動される場所）
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、特に注目すべき、任意のノードに対して、多くの便利な Hadoop 固有の設定を定義します。
 - ノードグループが使用すべきで、そのプラグインとバージョンは何ですか。
 - ノードで実行するプロセス。
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に定義します。
 - **ノードグループの構成:** 実際には、各ノードグループのノード数がクラスターを構成する必要があるノードの数です。
 - **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。

5. Hadoop クラスターを起動し（クラスターテンプレートを使用）、クラスターでジョブを実行します（データソースで登録された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（タイプのノードを追加または削除）。
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、それらを Hadoop クラスターで起動します。ジョブは、登録されたデータソースの処理に使用するジョブのバイナリー、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、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 MAKEFORM イメージの作成

本リリースでは、以下のプラグインがサポートされます。

- Cloudera(CDH)
 - 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)
 - バージョン 5.13.x (デフォルトは 5.13.0)
- Hortonworks Data Platform
 - バージョン 2.3
 - バージョン 2.4
- MapR 5.2

これらのプラグインに基づいて 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

イメージを準備する前に、システムが Red Hat サブスクリプション情報を使用するように設定して、システムが Hadoop イメージの作成に必要なツールをダウンロードできるようにします。これを行うには、(root で) 端末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** の値など、サブスクリプションに関する重要な情報を表示します。**Pool 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の**手動設定の詳細は**、「[Subscription Manager の使用および設定](#)」を参照してください。

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

```
# dnf 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。イメージを以下のチャンネルに登録する必要があります。

- 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。

```
# dnf 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 チャンネルのみを含める必要があります。

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 native 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](#) の場合)、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。 [バージョン 5.13.x](#) の場合は 5.13.0 を指定します。
- **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、プラグインで必要なソフトウェアを追加して、指定され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。以下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、ovn クラスタを起動し、使用するの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持つ 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。改善点を報告する場合は、以下のように行います。

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 > Data Processing > Image Registry** を選択します。
3. イメージの **登録** をクリックし、イメージ 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. OpenStack Data Processing サービスが設定を適用し、各インスタンス/ノードでプロセスを管理するために使用するユーザー名を入力します。Red Hat Enterprise Linux ([4章 makeform イメージの作成](#)で使用した) が提供する公式イメージでは、この目的のためにこの目的のために設定されたユーザー名は cloud-user です。
5. デフォルトでは、OpenStack Data Processing サービスは、プラグインおよび Version ドロップダウンメニューに必要なプラグイン およびバージョンタグを追加します。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、登録されたイメージを区別またはグループ化することも可能です。Add custom tag ボタンを使用してタグを追加します。タグは Description フィールドの下に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. 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ます。

5.1. 入力および出力データソースの登録

イメージを登録したら、データ入力ソースと出力先を登録します。どちらのオブジェクトとして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。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。



注記

データオブジェクトは別の Hadoop 互換の分散ファイルシステムから直接登録することもできます (例: HDFS)。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、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。データソースの名前を Name フィールドに入力します。
3. Description フィールドを使用してデータソースを記述します (任意)。
4. データソースのタイプおよび URL を選択します。この手順は、ソースのロケーション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 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 Hadoop Distributed File System(HDFS)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。ここでは、以下ようになります。
 - 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。
- データが S3 オブジェクトストアにある場合は、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 URL を `s3://bucket/path/to/object` の形式で入力します。
 - クラスター設定またはジョブ実行設定で以下のパラメーターがまだ設定されていない場合は、それらをここに設定する必要があります。
 - S3 アクセスキー
 - S3 シークレットキー
 - S3 エンドポイントは、プロトコルのない S3 サービスの URL です。
 - ブール値である必要がある SSL を使用します。
 - パスのバケットは、仮想ホストまたはパス URL を示します。また、ブール値である必要があります。

5. 完了をクリックします。データソースが Data Sources テーブルで利用可能になるはずです。

ジョブに必要な各データ入出力オブジェクトについて以下の手順を実行します。

第6章 ノードグループテンプレートの設定

Hadoop によるデータ処理には、クラスターでジョブを実行する必要があります。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）で構成されます。OpenStack Data Processing では、各クラスターに必要な異なるタイプ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、ノードグループの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、Alertmanager 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 プラグインの名前およびバージョンを選択します。
4. **Create** 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. **Description** フィールドを使用して、作成するノードグループの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適したものである必要があります。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**: このオプションでは、Block Storage サービスが各ノードに複数のボリュームを作成するように設定できます。ノードごとの volumes を使用してボリューム数を設定し、ボリュームサイズ(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 オプションが推奨されます。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** 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。Floating IP についての詳しい情報は、「[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、**Auto Security Group** 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リストから適切な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べき Hadoop プロセスを選択します。これには、Processs 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 テーブルに表示されます。

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、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。この表は、各テンプレートのノードで実行する Hadoop プロセスも一覧表示します。

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(5章 [必要なコンポーネントの登録](#))を登録し、必要なノードタイプ(6章 [ノードグループテンプレートの設定](#))を設定した後、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構成（各ノードタイプの数）利用可能なノードタイプは、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記載されています。
- クラスターによって使用される 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回のみ起動する必要があるプロセスの一覧です。
- 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 > Data Processing > Cluster Templates の順に選択します。
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 プラグインの名前およびバージョンを選択します。
3. Create 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. 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. Using anti-affinity groups for list リストから、ホストで複数回起動すべきでない各プロセスのチェックボックスにチェックを入れます。
7. ノードグループタブをクリックします。ここから、クラスター内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 を使用してテンプレートを追加します。ドロップダウンメニューでテンプレートを選択します（詳細は 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。クラスターテンプレートで使用する各ノードグループテンプレートに対して、これを行います。
 - c. 各ノードグループエントリーについて、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。
各ノードグループエントリーの名前を入力することも可能です（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、ほとんどのパラメーターにもコンテキストヘルプがあります。これらのパラメーターについての詳細は、各コンポーネントのサポートされる関数に関する、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に問題がなければ、Create をクリックします。テンプレートが Cluster Templates テーブル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することもできます。

また、Cluster Templates テーブル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 の順に選択します。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）を取得したら、クラスターを起動できます。これを行うには、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。

- Project > 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 をクリックした後、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 Hadoop プラグインの名前および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. Launch Cluster 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. Project > Data Processing > Clusters の表で Launch Cluster ウィザードを開くと、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。先に選択した dn 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必要のある 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 makeform 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. 必要な場合は、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、新規キーペアを作成することもできます。キーペアはクラスターを起動するには必要ありませんが、クラスターノードにログインするには（例：SSH を介して）必要になります。
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、「[一般的な管理タスク](#)」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 の順に選択します。

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、再ソースのニーズに合った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タイプ（または番号）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 の順に選択します。このページからクラスターをスケーリングまたは削除することができます。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 の表で、スケーリングするクラスターを選択します。次に、そのクラスターの 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スケーリングします。

- 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、ドロップダウンメニューを追加します。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。次に、追加するタイプのノード数を設定できます。
- 既存のノードグループに対してノードを追加または削除するには、ノードグループの行の+ 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 フィールドに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、アクション ドロップダウンメニューから Delete Cluster を選択します。また、複数のクラスターを削除して、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 (Pig、Hive、または MapReduce など)、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、これを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、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に似ています。Object Storage サービスから直接登録できます。Object Storage サービスにオブジェクトをアップロードする方法は、「[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、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直接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 の順に選択します。
2. Create Job Binary 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、オブジェクトを選択する際に使用されます。オブジェクトに特定の名前またはエクステンション (.jar など) が必要な場合は、ここに追加します。
4. 説明 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 (オプション)。
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、ストレージタイプ 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。
 - URL フィールドで、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 swift://CONTAINER/OBJECT として指定します。
 - 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 および Password フィールドに必要な認証情報を指定します。
 - b. オブジェクトが S3 ジョブバイナリーである場合は、ストレージタイプ ドロップダウンメニューから S3 を選択します。以下のようになります。
 - S3 URL を s3://bucket/path/to/object の形式で入力します。accesskey、secretkey、および endpoint も入力する必要があります。エンドポイントは、http または https プロトコルを含む S3 サービスの URL です。
 - c. それ以外の場合は、ストレージタイプ ドロップダウンメニューから Internal database を選択します。次に、Internal binary ドロップダウンメニューを使用して、以下のいずれかを実行します。
 - 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
 - Dashboard に直接スクリプトを入力する (スクリプトの作成)、または

- ローカルファイルシステムから直接バイナリー、ライブラリー、またはスクリプトをアップロードします（新しいファイルのアップロード）。
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ジョブ Binaries テーブルで利用可能になるはずです。

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、以下の手順を実施します。

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. ジョブタイプ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、サポートされているジョブタイプに関する、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には、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。
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、ライブラリーを追加します。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、Choose 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、Chosen ライブラリーリストに追加する必要があります。ジョブバイナリーに必要なすべてのライブラリーでこの手順を繰り返します。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。ジョブがジョブテーブルで利用可能になるはずです。

第10章 ジョブの起動

ジョブを作成したら、それを起動して OpenStack Data Processing サービスに登録されているデータを処理することができます（「[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（「[既存クラスターでのジョブの起動](#)」）または完全に新しい1（「[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. 既存クラスターでのジョブの起動

Dashboard で既存のクラスターの一覧を表示するには、Project > Data Processing > Clusters を選択します。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。

既存の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。ジョブの表には、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の詳細については、「[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、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 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。必要な場合は、ここで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の+をクリックします。これを実行すると、Create Data Source ウィザードが開きます。詳細は、「[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、ジョブを実行するクラスターを選択します。
5. このジョブの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 のいずれかで Add をクリックし、特別な名前/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 > データ処理 > ジョブ を選択します。起動したジョブを再起動または削除する方法は、「[起動ジョブを削除または再起動](#)」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の作成後に、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。ジョブの表には、利用可能なすべてのジョブが表示されます。新規ジョブの作成の詳細については、「[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、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Launch On New Clusterを選択します。
3. プラグイン 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 プラグインの名前および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必要のある 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makeform 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. 必要な場合は、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、新規キーペアを作成することもできます。キーペアはクラスターを起動するには必要ありませんが、クラスターノードにログインするには（例：SSHを介して）必要になります。
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、「[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、ジョブ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. 必要な場合は、ここで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の+をクリックします。これを実行すると、Create Data Source ウィザードが開きます。詳細は、「[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. このジョブの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 のいずれかで Add をクリックし、特別な名前/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>データ処理>ジョブを選択します。起動したジョブを再起動または削除する方法は、「[起動ジョブを削除または再起動](#)」を参照してください。

10.3. 起動ジョブを削除または再起動

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。また、起動した複数のジョブを削除するには、チェックボックスを選択して Delete job executions 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、Actions ドロップダウンメニューから Relaunch on Existing Cluster 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、新規クラスターでジョブ実行を完全に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 New Cluster で Relaunch 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 インスタンスの局所性の有効化

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、大幅に改善されます。特に `volume_local_to_instance` プロパティを `True` に設定すると、クラスターが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。

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、`InstanceLocality` を `/etc/cinder/cinder.conf` で有効なフィルタースケジューラーの一覧に含める必要があります。

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

`director` を使用してこれを設定するには、以下の内容を記述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、「[ボリュームの複数のバックエンドへの割り当て方法（ストレージガイド）](#)」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。

`InstanceLocality` フィルターを有効にした後に、クラスターで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るようになりました。改善点を報告する場合は、以下のように行います。

1. コマンドラインで、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、`settings.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      |
...

```

これにより、`volume_local_to_instance` プロパティを `True` に設定します。