

Реализация Spark jobs для Openstack Sahara

и использование Spark в Openstack (Liberty)

Алексиянц Александр,
Институт Системного Программирования РАН

Openstack Sahara

Цели проекта Openstack Sahara (2013-...):

1. Запуск и управление виртуальными кластерами для обработки больших данных в Openstack.
2. Поддержка основных дистрибутивов стека Apache технологий (Cloudera, HortonWorks, MapR, vanilla distros).
3. Предоставление доступа к виртуальным кластерам в режиме PaaS (из браузера, при помощи REST API, из CLI, при помощи Python SDK).

Исходные данные

На момент начала работы над проектом (феврале 2015 года) в Openstack Sahara присутствовала поддержка Spark (vanilla) со следующими ограничениями:

- Версия Apache Spark только 1.0.0 и 0.9.1.
- Для запуска Spark-job необходимо было подключаться к мастер-узлу Spark по ssh.
- Реализация разворачивания кластера Spark была сильно завязана на bash-скрипты.
- Нельзя было развернуть в пределах одного кластера Spark + что-то еще (например, Oozie)

Цели проекта

- Добавить поддержку создания кластеров Apache Spark разных версий.
- Добавить поддержку исполнения Spark-jobs через веб-интерфейс Horizon и API проекта Sahara.
- Сделать механизм запуска Spark-jobs универсальным и подходящим для любого вендора дистрибутивов.
- Сделать возможным создание кластеров Apache Spark вместе с другими полезными штуками.
- Сделать возможным использование Openstack Swift в качестве замены HDFS для кластеров Apache Spark.

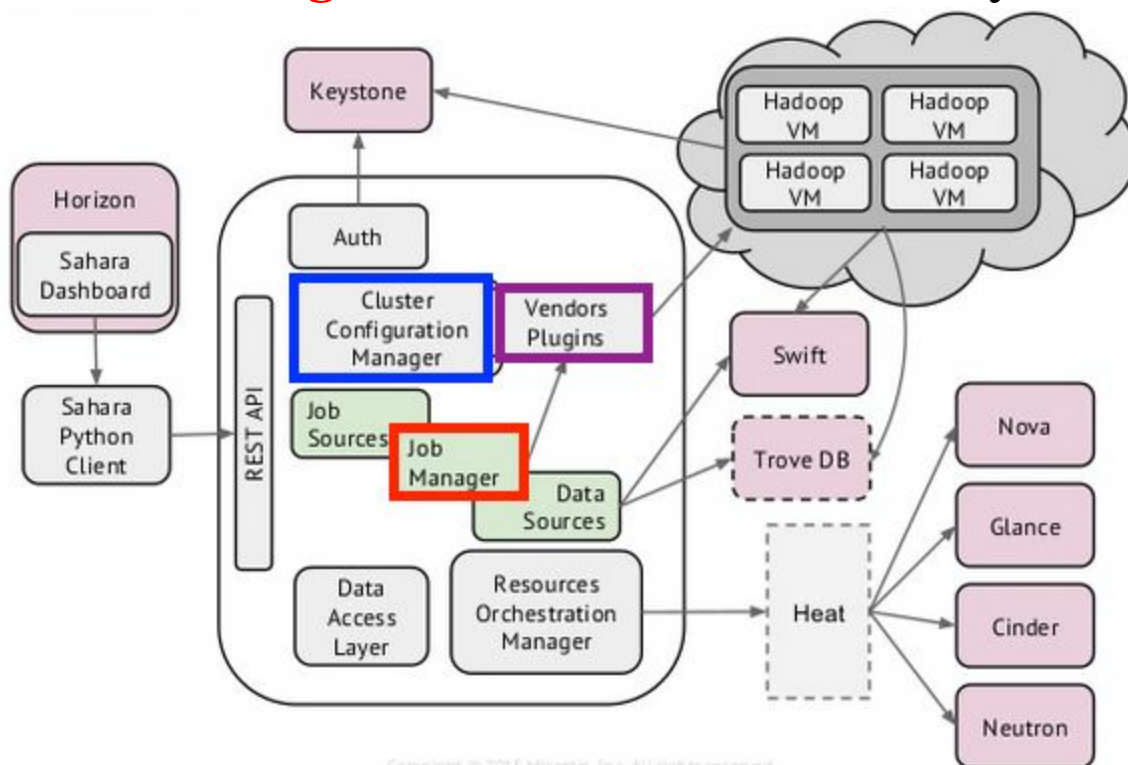
Но как можно было успеть сделать это за полгода?

На самом деле, 95% подготовительной (и самой сложной) работы было уже сделано до нас:

1. Разворачивание кластеров было реализовано командами Mirantis и Red Hat для всей остальной части Openstack Sahara.
2. Вся оркестрация систем стека технологий была уже реализована вендорами дистрибутивов (CDH, HDP, MapR).
3. Системы оркестрации были встроены в Openstack Sahara совместными усилиями вендоров дистрибутивов и команд Mirantis и Red Hat.
4. Слой совместимости между Openstack Swift и HDFS также уже был реализован вендорами.

И что же мы тогда сделали?

- Доработали **Cluster Configuration Manager** и **Vendors plugins** для возможности создавать Spark кластеры с возможностью использования Swift.
- Доработали **Job Manager** для возможности запуска Spark заданий.



Как пользоваться: Horizon

 admin ▼

admin ▼

Project ▼

Admin ▲

System ▲

Overview

Hypervisors

Host Aggregates

Instances

Volumes

Flavors

Images

Networks

Routers

Defaults

Overview

Usage Summary

Select a period of time to query its usage:

From: 2015-08-01 To: 2015-08-25 [Submit](#) The date should be in YYYY-mm-dd format.

Active Instances: 4 Active RAM: 26ГБ This Period's VCPU-Hours: 7800,00 This Period's GB-Hours: 155999,93 This Period's RAM-Hours: 15974392,60

Usage

[Download CSV Summary](#)

Project Name	VCPU	Disk ▼	RAM	VCPU Hours ⓘ	Disk GB Hours ⓘ	Memory MB Hours ⓘ
admin	12	240ГБ	24ГБ	7200,00	143999,93	14745593,17
demo	1	20ГБ	2ГБ	600,00	11999,99	1228799,43

Displaying 2 items

Как пользоваться: Horizon

 admin ▾

admin ▾

Project ^

Compute ▾

Network ▾

Object Store ▾

Orchestration ▾

Data Processing ^

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Job Binaries

Overview

Usage Summary

Select a period of time to query its usage:

From: To: [Submit](#) The date should be in YYYY-mm-dd format.

Active Instances: 4 Active RAM: 26ГБ This Period's VCPU-Hours: 7800,00 This Period's GB-Hours: 155999,93 This Period's RAM-Hours: 15974392,60

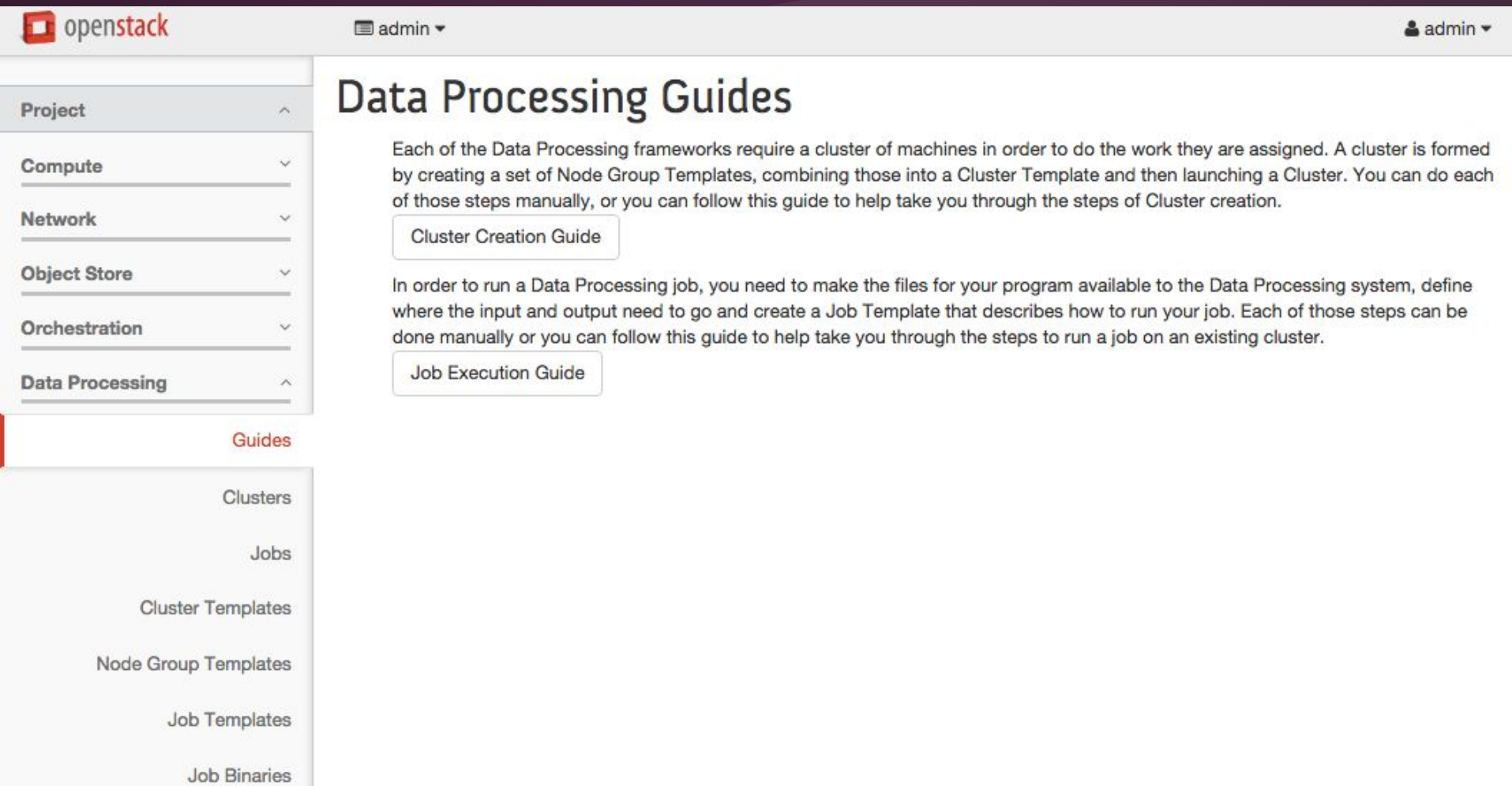
Usage

[Download CSV Summary](#)

Project Name	VCPU	Disk ▾	RAM	VCPU Hours ⓘ	Disk GB Hours ⓘ	Memory MB Hours ⓘ
admin	12	240ГБ	24ГБ	7200,00	143999,93	14745593,17
demo	1	20ГБ	2ГБ	600,00	11999,99	1228799,43

Displaying 2 items

Как пользоваться: Horizon



The screenshot displays the OpenStack Horizon web interface. At the top, the OpenStack logo is on the left, and the user 'admin' is logged in on the right. A left-hand navigation menu lists various OpenStack services: Project, Compute, Network, Object Store, Orchestration, and Data Processing. Below this menu, a secondary list of links is shown: Guides, Clusters, Jobs, Cluster Templates, Node Group Templates, Job Templates, and Job Binaries. The main content area is titled 'Data Processing Guides' and contains an introductory paragraph about cluster creation. Below the text are two buttons: 'Cluster Creation Guide' and 'Job Execution Guide'.

openstack

admin

admin

Data Processing Guides

Each of the Data Processing frameworks require a cluster of machines in order to do the work they are assigned. A cluster is formed by creating a set of Node Group Templates, combining those into a Cluster Template and then launching a Cluster. You can do each of those steps manually, or you can follow this guide to help take you through the steps of Cluster creation.

Cluster Creation Guide

In order to run a Data Processing job, you need to make the files for your program available to the Data Processing system, define where the input and output need to go and create a Job Template that describes how to run your job. Each of those steps can be done manually or you can follow this guide to help take you through the steps to run a job on an existing cluster.

Job Execution Guide

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Job Binaries

Как пользоваться: Horizon

openstack admin admin

Guided Cluster Creation

1. The first step is to determine which type of cluster you want to run. You may have several choices available depending on the configuration of your system. Click on "choose plugin" to bring up the list of data processing plugins. There you will be able to choose the data processing plugin along with the version number. Choosing this up front will allow the rest of the cluster creation steps to focus only on options that are pertinent to your desired cluster type.
[Choose plugin](#)
Current choice: **No plugin chosen**
2. Next, you need to define the different types of machines in your cluster. This is done by defining a Node Group Template for each type of machine. A very common case is where you need to have one or more machines running a "master" set of processes while another set of machines need to be running the "worker" processes. Here, you will define the Node Group Template for your "master" node(s).
[+ Create a Master Node Group Template](#)
Current choice: **No Master Node Group Template Created**
3. Repeat the Node Group Template creation process, but this time you are creating your "worker" Node Group Template.
[+ Create a Worker Node Group Template](#)
Current choice: **No Worker Node Group Template Created**

Как пользоваться: Horizon

The screenshot displays the OpenStack Horizon web interface. On the left, a sidebar contains navigation links: Project, Compute, Network, Object Store, Orchestration, and Data Processing. The main content area shows a 'Clusters' section with buttons for 'Create a Master Node Group Template' and 'Create a Worker Node Group Template'. Below these buttons, the status indicates 'Current choice: No Master Node Group Template Created' and 'Current choice: No Worker Node Group Template Created'. A modal dialog box titled 'Choose plugin and version' is centered on the screen. It features a 'Plugin Name' dropdown menu with 'Cloudera Plugin' selected, and a 'Cloudera Plugin' dropdown menu with '5.4.0' selected. A text box on the right of the dialog reads: 'Select which plugin and version that you want to use to create your cluster.' At the bottom right of the dialog are 'Cancel' and 'Select' buttons. The background interface is partially visible, showing the 'Jobs' and 'Cluster Templates' sections.

openstack admin admin

Choose plugin and version

Plugin Name *

Cloudera Plugin

Cloudera Plugin

5.4.0

Select which plugin and version that you want to use to create your cluster.

Cancel Select

depending on the
you will be able to choose
master creation steps to

Template for each type
processes while another
or your "master" node(s).

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Job Binaries

Create a Master Node Group Template

Create a Worker Node Group Template

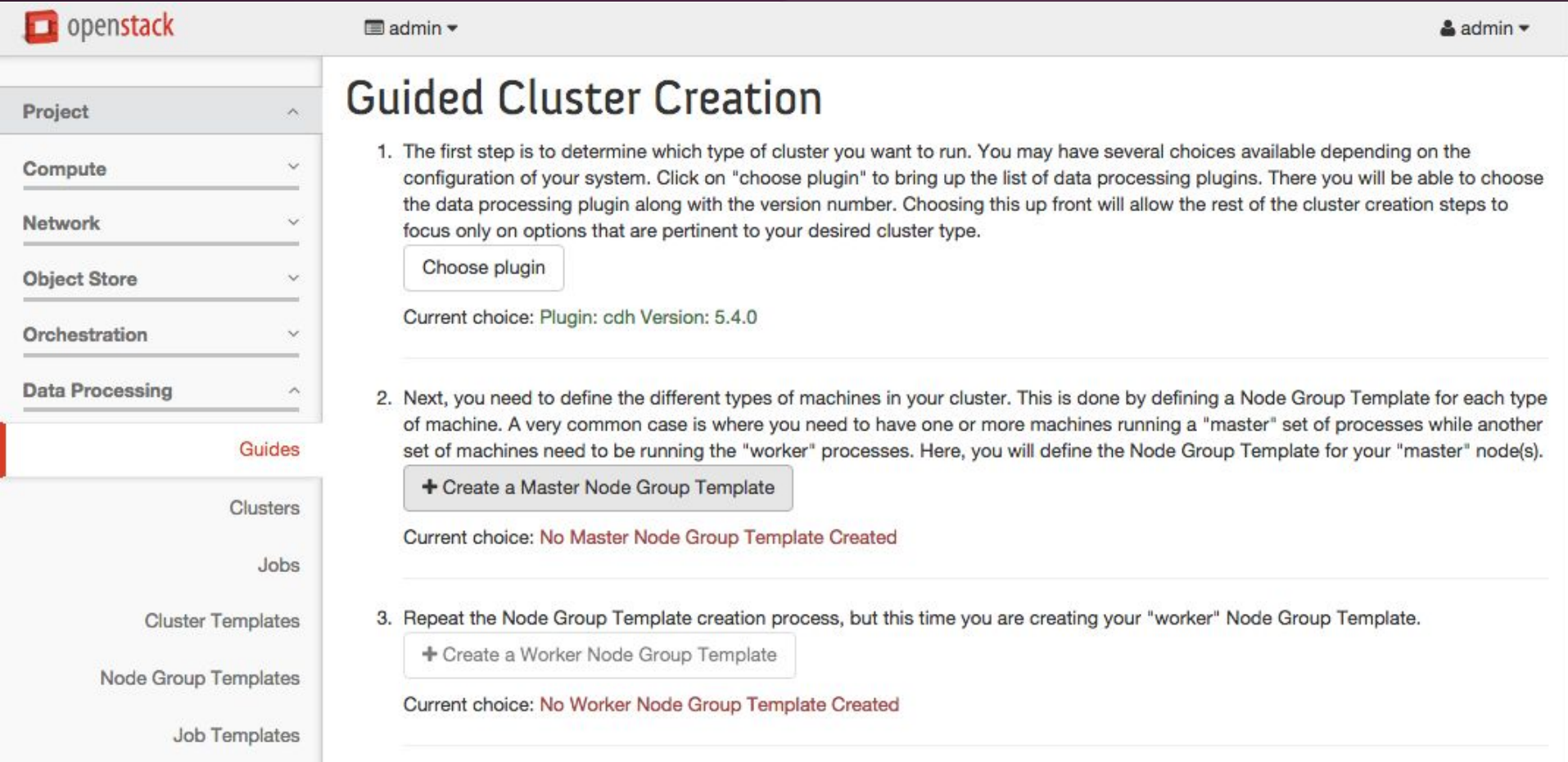
Current choice: No Master Node Group Template Created

Current choice: No Worker Node Group Template Created

3. Repeat the Node Group Template creation process, but this time you are creating your "worker" Node Group Template.

4. Now you need to set the layout of your cluster. By creating a Cluster Template, you will be choosing the number of instances of each

Как пользоваться: Horizon



openstack admin admin

Guided Cluster Creation

- The first step is to determine which type of cluster you want to run. You may have several choices available depending on the configuration of your system. Click on "choose plugin" to bring up the list of data processing plugins. There you will be able to choose the data processing plugin along with the version number. Choosing this up front will allow the rest of the cluster creation steps to focus only on options that are pertinent to your desired cluster type.
[Choose plugin](#)
Current choice: Plugin: cdh Version: 5.4.0
- Next, you need to define the different types of machines in your cluster. This is done by defining a Node Group Template for each type of machine. A very common case is where you need to have one or more machines running a "master" set of processes while another set of machines need to be running the "worker" processes. Here, you will define the Node Group Template for your "master" node(s).
[+ Create a Master Node Group Template](#)
Current choice: No Master Node Group Template Created
- Repeat the Node Group Template creation process, but this time you are creating your "worker" Node Group Template.
[+ Create a Worker Node Group Template](#)
Current choice: No Worker Node Group Template Created

Как пользоваться: Horizon

openstack admin admin

Create Node Group Template

Configure Node Group Template * Security CATALOGSERVER Parameters SENTRY Parameters

STATESTORE Parameters KMS Parameters MASTER Parameters FLUME Parameters

HUE Parameters JOBHISTORY Parameters HDFS_GATEWAY Parameters IMPALAD Parameters

REGIONSERVER Parameters DATANODE Parameters OOZIE Parameters

HIVEMETASTORE Parameters SQOOP Parameters RESOURCEMANAGER Parameters

SECONDARYNAMENODE Parameters **NAMENODE Parameters** SOLR Parameters

KS_INDEXER Parameters JOURNALNODE Parameters SPARK_ON_YARN Parameters

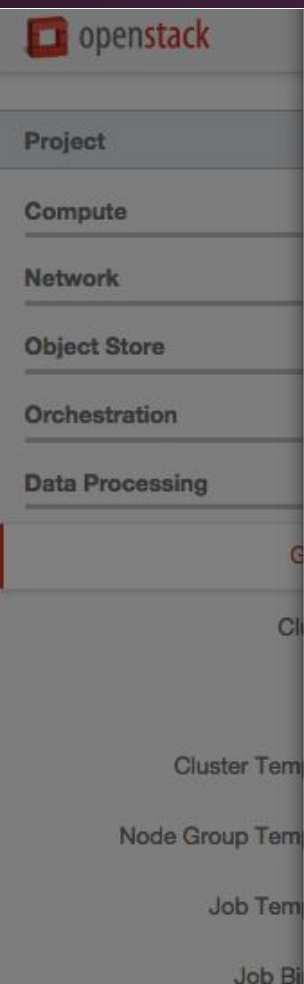
HIVESERVER Parameters ZOOKEEPER Parameters YARN_GATEWAY Parameters

NODEMANAGER Parameters WEBHCAT Parameters

Template Name *

This Node Group Template will be created for:
Plugin: cdh

Как пользоваться: Horizon



Template Name *

cloudera-spark54

Description

OpenStack Flavor *

m1.large

Availability Zone ?

nova

Storage location * ?

Ephemeral Drive

Floating IP Pool

ext-net

☐ Proxy Gateway ?

Processes * ?

☐ HADOOP CATALOGBROWSER

This Node Group Template will be created for:

Plugin: cdh

Version: 5.4.0

The Node Group Template object specifies the processes that will be launched on each instance. Check one or more processes. When processes are selected, you may set **node** scoped configurations on corresponding tabs.

You must choose a flavor to determine the size (VCPUs, memory and storage) of all launched VMs.

Data Processing provides different storage location options. You may choose Ephemeral Drive or a Cinder Volume to be attached to instances.

admin

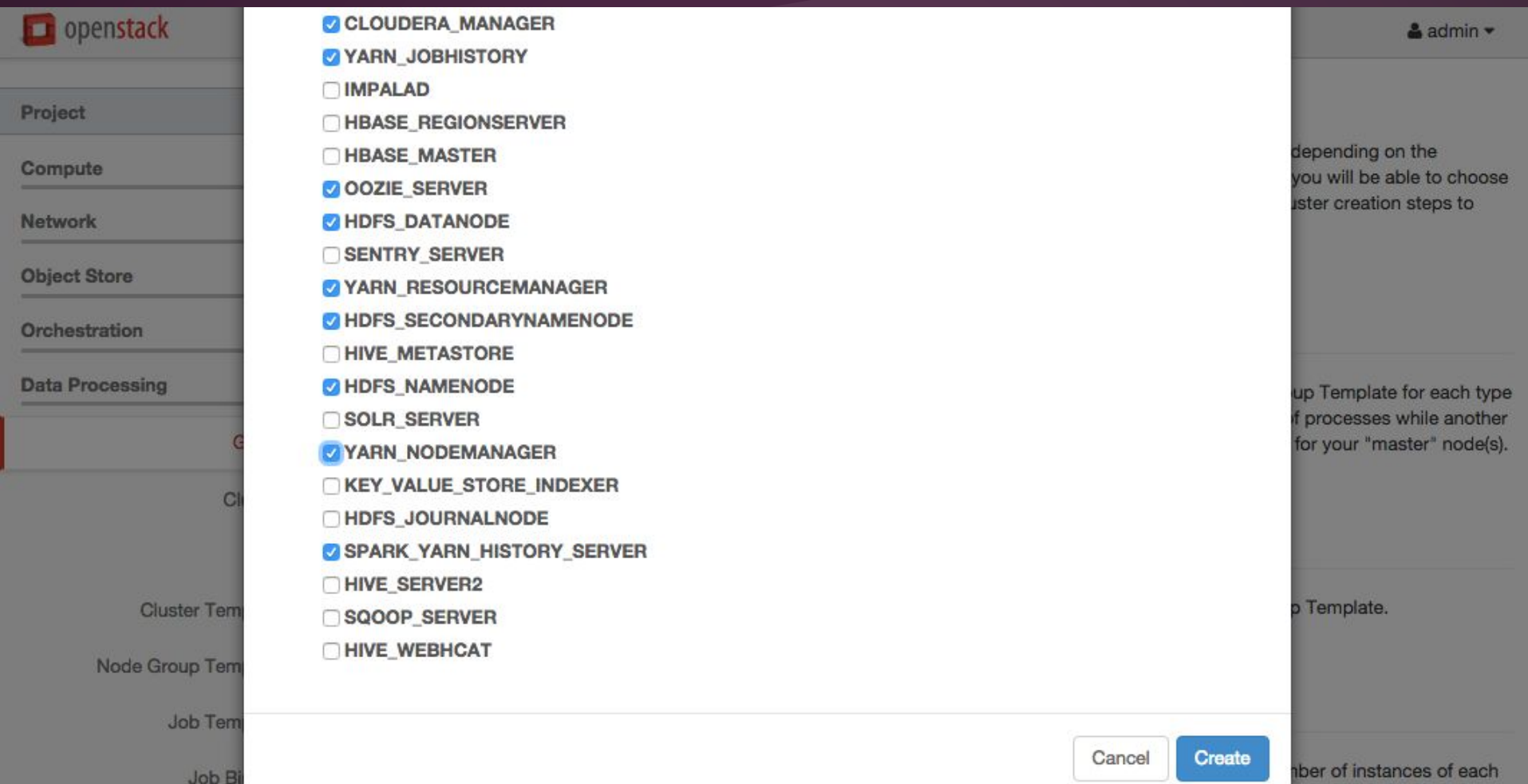
depending on the
you will be able to choose
cluster creation steps to

up Template for each type
if processes while another
for your "master" node(s).

p Template.

number of instances of each

Как пользоваться: Horizon

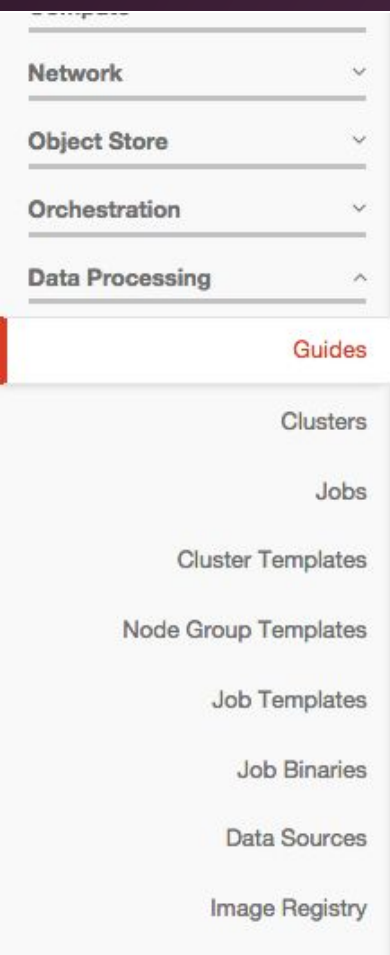


The screenshot displays the OpenStack Horizon web interface. On the left, a sidebar menu shows navigation options: Project, Compute, Network, Object Store, Orchestration, and Data Processing (which is currently selected). The main content area is titled 'Data Processing' and contains a list of services with checkboxes for selection. The services listed are:

- ☒ CLOUDERA_MANAGER
- ☒ YARN_JOBHISTORY
- ☐ IMPALAD
- ☐ HBASE_REGIONSERVER
- ☐ HBASE_MASTER
- ☒ OOZIE_SERVER
- ☒ HDFS_DATANODE
- ☐ SENTRY_SERVER
- ☒ YARN_RESOURCEMANAGER
- ☒ HDFS_SECONDARYNAMENODE
- ☐ HIVE_METASTORE
- ☒ HDFS_NAMENODE
- ☐ SOLR_SERVER
- ☒ YARN_NODEMANAGER
- ☐ KEY_VALUE_STORE_INDEXER
- ☐ HDFS_JOURNALNODE
- ☒ SPARK_YARN_HISTORY_SERVER
- ☐ HIVE_SERVER2
- ☐ SQOOP_SERVER
- ☐ HIVE_WEBHCAT

At the bottom right of the main content area, there are two buttons: 'Cancel' and 'Create'. The right side of the image shows a partial view of the 'Create' button and some text from the next slide.

Как пользоваться: Horizon



configuration of your system. Click on "choose plugin" to bring up the list of data processing plugins. Here you will be able to choose the data processing plugin along with the version number. Choosing this up front will allow the rest of the cluster creation steps to focus only on options that are pertinent to your desired cluster type.

Choose plugin

Current choice: Plugin: cdh Version: 5.4.0

2. Next, you need to define the different types of machines in your cluster. This is done by defining a Node Group Template for each type of machine. A very common case is where you need to have one or more machines running a "master" set of processes while another set of machines need to be running the "worker" processes. Here, you will define the Node Group Template for your "master" node(s).

+ Create a Master Node Group Template

Current choice: Master Node Group Template: cloudera-spark54

3. Repeat the Node Group Template creation process, but this time you are creating your "worker" Node Group Template.

+ Create a Worker Node Group Template

Current choice: No Worker Node Group Template Created

4. Now you need to set the layout of your cluster. By creating a Cluster Template, you will be choosing the number of instances of each Node Group Template that will appear in your cluster. Additionally, you will have a chance to set any cluster-specific configuration items in the additional tabs on the create Cluster Template form.

+ Create a Cluster Template

Current choice: No Cluster Template Created

Как пользоваться: Horizon

Network

Object Store

Orchestration

Data Processing

Cluster Template

Node Group Template

Job Template

Job Blueprint

Data Source

Image Repository

NODEMANAGER Parameters

WEBHCAT Parameters

Template Name *

cloudera-spark54slave

Description

OpenStack Flavor *

m1.large

Availability Zone ?

nova

Storage location * ?

Ephemeral Drive

Floating IP Pool

ext-net

☐ Proxy Gateway ?

This Node Group Template will be created for:
Plugin: cdh
Version: 5.4.0

The Node Group Template object specifies the processes that will be launched on each instance. Check one or more processes. When processes are selected, you may set **node** scoped configurations on corresponding tabs.

You must choose a flavor to determine the size (VCPUs, memory and storage) of all launched VMs.

Data Processing provides different storage location options. You may choose Ephemeral Drive or a Cinder Volume to be attached to instances.

you will be able to choose cluster creation steps to

up Template for each type of processes while another for your "master" node(s).

o Template.

number of instances of each specific configuration

Как пользоваться: Horizon

The screenshot shows the Horizon web interface for creating a cluster template. On the left, a sidebar lists various components: Network, Object Store, Orchestration, and Data Processing. The main area displays a list of services with checkboxes. The following table represents the visible data in this list:

Service	Selected
☐ YARN_JOBHISTORY	No
☐ IMPALAD	No
☐ HBASE_REGIONSERVER	No
☐ HBASE_MASTER	No
☐ OOZIE_SERVER	No
☒ HDFS_DATANODE	Yes
☐ SENTRY_SERVER	No
☐ YARN_RESOURCEMANAGER	No
☐ HDFS_SECONDARYNAMENODE	No
☐ HIVE_METASTORE	No
☐ HDFS_NAMENODE	No
☐ SOLR_SERVER	No
☒ YARN_NODEMANAGER	Yes
☐ KEY_VALUE_STORE_INDEXER	No
☐ HDFS_JOURNALNODE	No
☐ SPARK_YARN_HISTORY_SERVER	No
☐ HIVE_SERVER2	No
☐ SQOOP_SERVER	No
☐ HIVE_WEBHCAT	No

At the bottom right of the modal, there are two buttons: "Cancel" and "Create".

Как пользоваться: Horizon

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Job Binaries

Data Sources

Image Registry

Plugins

Admin

Identity

of machine. A very common case is where you need to have one or more machines running a "master" set of processes while another set of machines need to be running the "worker" processes. Here, you will define the Node Group Template for your "master" node(s).

+ Create a Master Node Group Template

Current choice: Master Node Group Template: cloudera-spark54

3. Repeat the Node Group Template creation process, but this time you are creating your "worker" Node Group Template.

+ Create a Worker Node Group Template

Current choice: Worker Node Group Template: cloudera-spark54slave

4. Now you need to set the layout of your cluster. By creating a Cluster Template, you will be choosing the number of instances of each Node Group Template that will appear in your cluster. Additionally, you will have a chance to set any cluster-specific configuration items in the additional tabs on the create Cluster Template form.

+ Create a Cluster Template

Current choice: No Cluster Template Created

5. You are now ready to launch your cluster. When you click on the link below, you will need to give your cluster a name, choose the Cluster Template to use and choose which image to use to build your instances. After you click on "Create", your instances will begin to spawn. Your cluster should be operational in a few minutes.

+ Launch a Cluster

Reset Cluster Creation Guide

Как пользоваться: Horizon

Create Cluster Template

Details *

Node Groups *

General Parameters

SENTRY Parameters

KMS Parameters

FLUME Parameters

HUE Parameters

YARN Parameters

OOZIE Parameters

SQOOP Parameters

SOLR Parameters

HBASE Parameters

HDFS Parameters

KS_INDEXER Parameters

SPARK_ON_YARN Parameters

ZOOKEEPER Parameters

HIVE Parameters

IMPALA Parameters

Template Name *

cl54

Description

This Cluster Template will be created for:

Plugin: cdh

Version: 5.4.0

The Cluster Template object should specify Node Group Templates that will be used to build a Cluster. You can add Node Groups using Node Group Templates on a "Node Groups" tab.

You may set **cluster** scoped configurations on

Как пользоваться: Horizon

Create Cluster Template

Details * Node Groups * General Parameters SENTRY Parameters KMS Parameters
FLUME Parameters HUE Parameters YARN Parameters OOZIE Parameters
SQOOP Parameters SOLR Parameters HBASE Parameters HDFS Parameters
KS_INDEXER Parameters SPARK_ON_YARN Parameters ZOOKEEPER Parameters
HIVE Parameters IMPALA Parameters

Select a Node Group Template to add:

Select

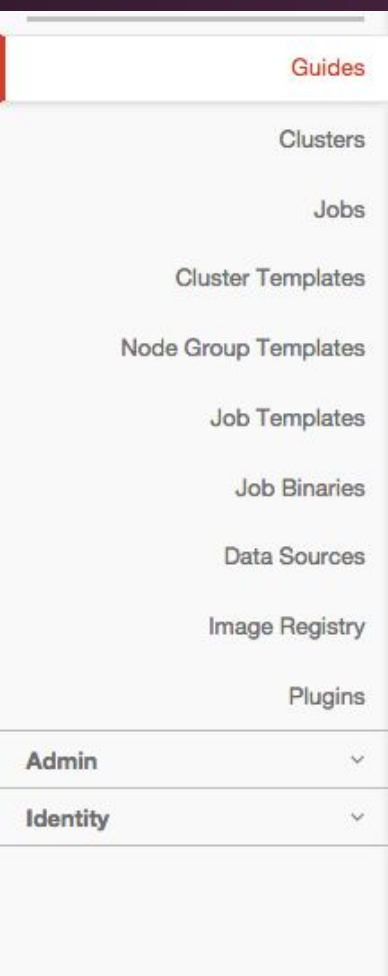


Group Name	Template	Count	
cloudera-spark54	cloudera-spark54	1	- + Remove
cloudera-spark54slave	cloudera-spark54slave	3	- + Remove

Cancel

Create

Как пользоваться: Horizon



of machine. A very common case is where you need to have one or more machines running a "master" set of processes while another set of machines need to be running the "worker" processes. Here, you will define the Node Group Template for your "master" node(s).

[+ Create a Master Node Group Template](#)

Current choice: Master Node Group Template: cloudera-spark54

3. Repeat the Node Group Template creation process, but this time you are creating your "worker" Node Group Template.

[+ Create a Worker Node Group Template](#)

Current choice: Worker Node Group Template: cloudera-spark54slave

4. Now you need to set the layout of your cluster. By creating a Cluster Template, you will be choosing the number of instances of each Node Group Template that will appear in your cluster. Additionally, you will have a chance to set any cluster-specific configuration items in the additional tabs on the create Cluster Template form.

[+ Create a Cluster Template](#)

Current choice: Worker Node Group Template: cl54

5. You are now ready to launch your cluster. When you click on the link below, you will need to give your cluster a name, choose the Cluster Template to use and choose which image to use to build your instances. After you click on "Create", your instances will begin to spawn. Your cluster should be operational in a few minutes.

[+ Launch a Cluster](#)

[Reset Cluster Creation Guide](#)

Как пользоваться: Horizon

Launch Cluster

Configure Cluster *

Cluster Name *
democluster

Description

Cluster Template *
cl54

Base Image *
ubuntu_sahara_cloudera_5_4_0

Keypair ⓘ
al_indigo

Neutron Management Network *
ext-net

This Cluster will be started with:
Plugin: cdh
Version: 5.4.0

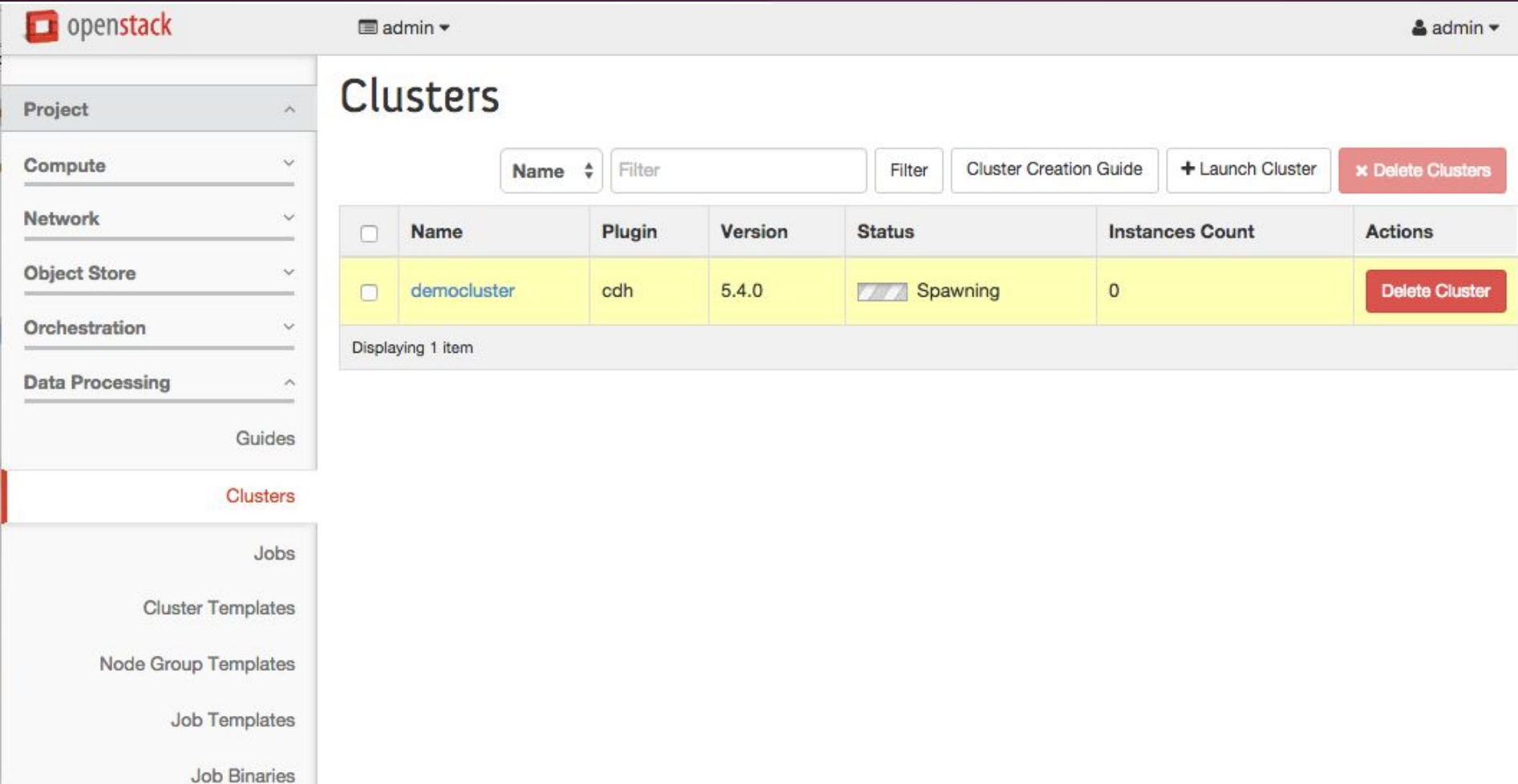
Cluster can be launched using existing Cluster Templates.

The Cluster object should specify OpenStack Image to boot instances for Cluster.

User has to choose a keypair to have access to clusters instances.

Cancel Launch

Как пользоваться: Horizon



The screenshot displays the OpenStack Horizon web interface for managing clusters. The top navigation bar includes the OpenStack logo, a user menu for 'admin', and a 'Clusters' link in the sidebar. The main content area is titled 'Clusters' and features a table with the following data:

	Name	Plugin	Version	Status	Instances Count	Actions
☐	democluster	cdh	5.4.0	Spawning	0	<button>Delete Cluster</button>

Below the table, it states 'Displaying 1 item'. Above the table, there are controls for filtering by name, a 'Filter' button, a 'Cluster Creation Guide' link, a '+ Launch Cluster' button, and a 'x Delete Clusters' button.

Как пользоваться: Horizon

The screenshot shows the OpenStack Horizon interface. The top header includes the OpenStack logo, a user menu for 'admin', and a user icon for 'admin'. The left sidebar contains a navigation menu with the following items: Project, Compute, Network, Object Store, Orchestration, and Data Processing. Below these are links for Guides and Clusters. The 'Jobs' section is highlighted in red. The main content area is titled 'Jobs' and features a table with the following columns: ID, Job Template, Cluster, Status, and Actions. A single job is listed with ID 7a9a1ad8-4ded-4d3e-bc26-bc60535bc581, Job Template spark-wordcount-template-swift, Cluster Not available, and Status Succeeded. The Actions column for this job contains a 'Delete Job' button. Above the table, there are controls for Status, Filter, and a 'Delete Jobs' button. Below the table, it says 'Displaying 1 item'.

openstack admin admin

Jobs

Status Filter Filter Job Guide x Delete Jobs

☐	ID	Job Template	Cluster	Status	Actions
☐	7a9a1ad8-4ded-4d3e-bc26-bc60535bc581	spark-wordcount-template-swift	Not available	Succeeded	Delete Job

Displaying 1 item

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Job Binaries

Как пользоваться: Horizon

openstack admin admin

Guided Job Execution

1. First, select which type of job that you want to run. This choice will determine which other steps are required

[+ Select type](#)

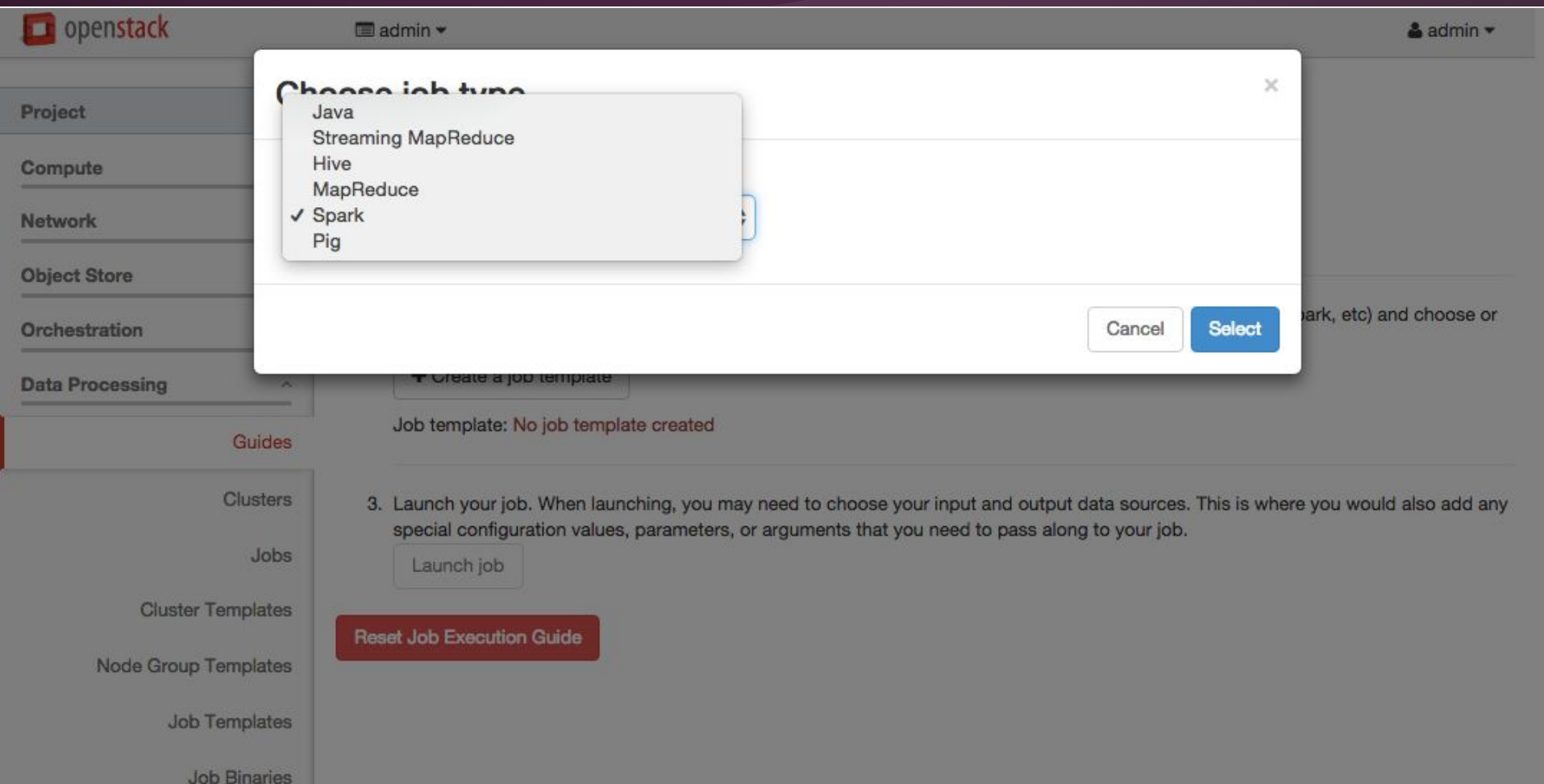
Current type: No type chosen

[Reset Job Execution Guide](#)

Guides

- Clusters
- Jobs
- Cluster Templates
- Node Group Templates
- Job Templates
- Job Binaries

Как пользоваться: Horizon



The screenshot displays the OpenStack Horizon web interface. On the left, a sidebar menu lists various sections: Project, Compute, Network, Object Store, Orchestration, and Data Processing. The 'Data Processing' section is currently selected. A modal dialog box titled 'Choose job type' is open in the center, featuring a list of job types: Java, Streaming MapReduce, Hive, MapReduce, Spark (which is selected with a checkmark), and Pig. At the bottom of the dialog are 'Cancel' and 'Select' buttons. In the background, the main content area shows a 'Job template: No job template created' message, a 'Launch job' button, and a 'Reset Job Execution Guide' button. A step 3 instruction is visible: '3. Launch your job. When launching, you may need to choose your input and output data sources. This is where you would also add any special configuration values, parameters, or arguments that you need to pass along to your job.'

Как пользоваться: Horizon

openstack admin admin

Guided Job Execution

1. First, select which type of job that you want to run. This choice will determine which other steps are required.
[+ Select type](#)
Current type: Spark
2. Define your Job Template. This is where you choose the type of job that you want to run (Pig, Java Action, Spark, etc) and choose or upload the files necessary to run it. The inputs and outputs will be defined later.
[+ Create a job template](#)
Job template: No job template created
3. Launch your job. When launching, you may need to choose your input and output data sources. This is where you would also add any special configuration values, parameters, or arguments that you need to pass along to your job.
[Launch job](#)

[Reset Job Execution Guide](#)

Project ^

Compute ^

Network ^

Object Store ^

Orchestration ^

Data Processing ^

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Как пользоваться: Horizon

The screenshot shows the 'Create Job Template' interface in the OpenStack Horizon dashboard. On the left is a sidebar with navigation links: Project, Compute, Network, Object Store, Orchestration, and Data Processing. The main content area has a title 'Create Job Template' and a close button. Below the title are two tabs: 'Create Job Template *' (active) and 'Libs'. The form contains the following fields:

- Name ***: A text input field containing 'sparkwc'.
- Job Type ***: A dropdown menu with 'Spark' selected.
- Choose a main binary ?**: A dropdown menu with 'sparkwc' selected and a '+' button to add more binaries.
- Description**: A large text area for an optional description.

On the right side of the form, there is a help box with the following text:

Create a job template with a specified name.

Select the type of your job:

- Pig
- Hive
- Spark
- MapReduce
- Java Action

Choose or create your main binary. Additional libraries can be added from the "Libs" tab.

For Spark jobs, only a main is required, "libs" are optional.

For MapReduce or Java Action jobs, "mains" are not applicable. You are required to add one or more "libs" for these jobs.

You may also enter an optional description for your job template.

At the top right of the dashboard, the user is logged in as 'admin'.

Как пользоваться: Horizon

openstack admin admin

Guided Job Execution

1. First, select which type of job that you want to run. This choice will determine which other steps are required.
[+ Select type](#)
Current type: Spark
2. Define your Job Template. This is where you choose the type of job that you want to run (Pig, Java Action, Spark, etc) and choose or upload the files necessary to run it. The inputs and outputs will be defined later.
[+ Create a job template](#)
Job template: sparkwc
3. Launch your job. When launching, you may need to choose your input and output data sources. This is where you would also add any special configuration values, parameters, or arguments that you need to pass along to your job.
[Launch job](#)

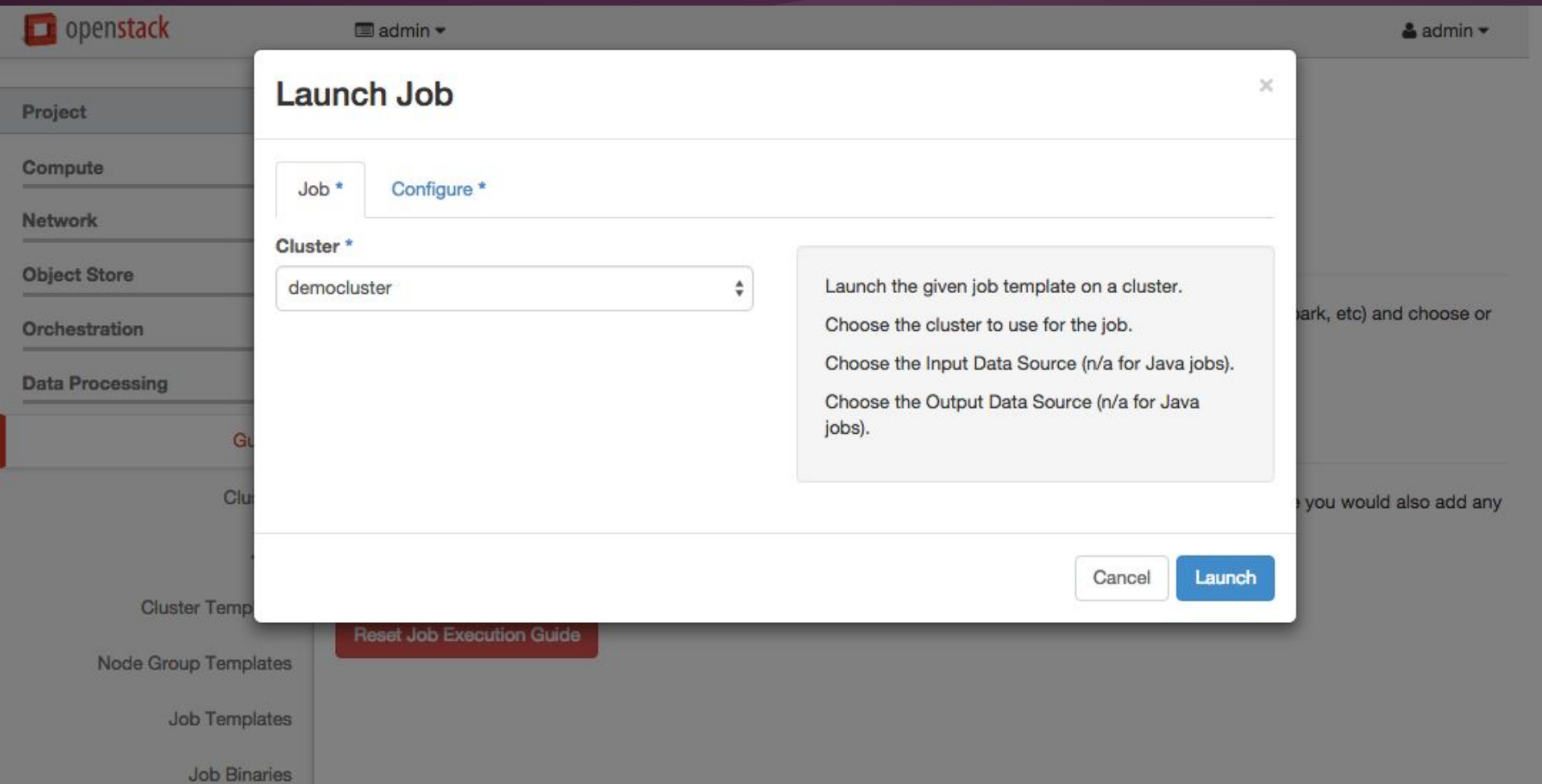
[Reset Job Execution Guide](#)

Project ^
Compute ^
Network ^
Object Store ^
Orchestration ^
Data Processing ^

Guides

Clusters
Jobs
Cluster Templates
Node Group Templates
Job Templates

Как пользоваться: Horizon



The screenshot shows the OpenStack Horizon interface with a 'Launch Job' dialog box open. The dialog has two tabs: 'Job *' (selected) and 'Configure *'. Under the 'Job *' tab, there is a 'Cluster *' dropdown menu with 'democluster' selected. To the right of the dropdown is a text box containing instructions: 'Launch the given job template on a cluster. Choose the cluster to use for the job. Choose the Input Data Source (n/a for Java jobs). Choose the Output Data Source (n/a for Java jobs)'. At the bottom right of the dialog are 'Cancel' and 'Launch' buttons. The background shows the Horizon sidebar with links to Project, Compute, Network, Object Store, Orchestration, and Data Processing, and the top header with the OpenStack logo and user 'admin'.

openstack admin admin

Launch Job

Job * Configure *

Cluster *

democluster

Launch the given job template on a cluster.
Choose the cluster to use for the job.
Choose the Input Data Source (n/a for Java jobs).
Choose the Output Data Source (n/a for Java jobs).

Cancel Launch

Reset Job Execution Guide

Node Group Templates
Job Templates
Job Binaries

Как пользоваться: Horizon

The screenshot displays the OpenStack Horizon interface. On the left is a sidebar with navigation links: Project, Compute, Network, Object Store, Orchestration, Data Processing, Guides, Clusters, Jobs, Cluster Templates, Node Group Templates, Job Templates, Job Binaries, Data Sources, Image Registry, Plugins, Admin, and Identity. The main content area is titled 'Guided J' and shows a 'Configure' tab for a job template. The 'Main Class' is set to 'com.cloudera.sparkwordcount.SparkWordCount'. Below this is a 'Java Opts' field. The 'Configuration' section contains a table with three entries:

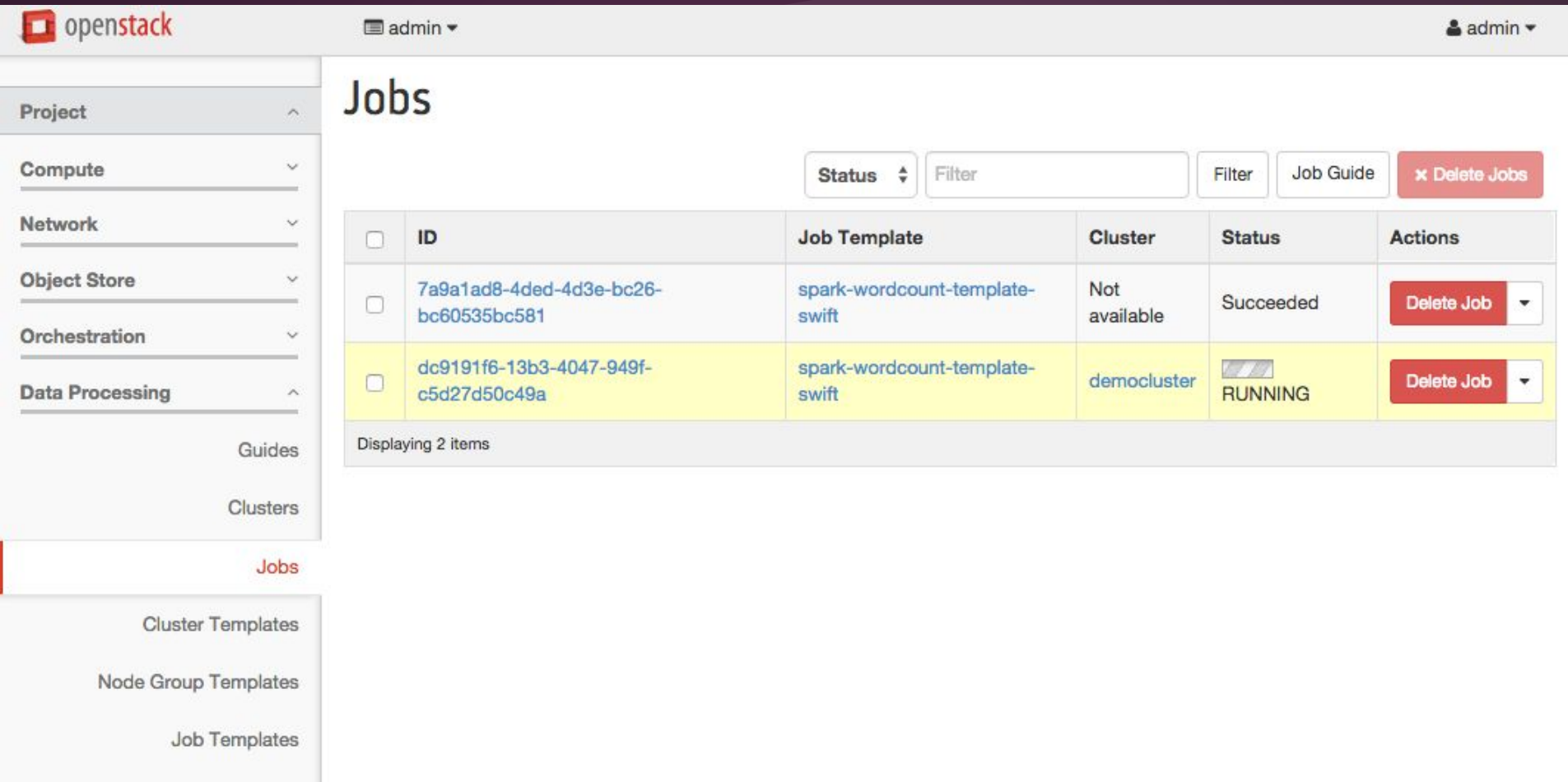
Name	Value	Action
fs.swift.service.sahara.username	admin	Remove
fs.swift.service.sahara.password	*****	Remove
edp.spark.adapt_for_swift	True	Remove

Below the configuration table is an 'Add' button. The 'Arguments' section has a 'Value' field with two entries:

Value	Action
swift://vg-cont/input	Remove
2	Remove

At the bottom of the configuration panel is another 'Add' button. At the very bottom of the main content area are 'Cancel' and 'Launch' buttons.

Как пользоваться: Horizon



The screenshot displays the OpenStack Horizon web interface. The top navigation bar includes the OpenStack logo, a user menu for 'admin', and a 'Jobs' section header. The left sidebar contains a navigation menu with categories like Project, Compute, Network, Object Store, and Data Processing, along with links to Guides, Clusters, Jobs, Cluster Templates, Node Group Templates, and Job Templates. The main content area shows a table of jobs with columns for ID, Job Template, Cluster, Status, and Actions. Two jobs are listed: one with ID '7a9a1ad8-4ded-4d3e-bc26-bc60535bc581' in a 'Succeeded' state, and another with ID 'dc9191f6-13b3-4047-949f-c5d27d50c49a' in a 'RUNNING' state. The 'RUNNING' job is highlighted in yellow. A 'Delete Job' button is visible for each job.

openstack admin admin

Jobs

Status Filter Filter Job Guide x Delete Jobs

☐	ID	Job Template	Cluster	Status	Actions
☐	7a9a1ad8-4ded-4d3e-bc26-bc60535bc581	spark-wordcount-template-swift	Not available	Succeeded	Delete Job
☐	dc9191f6-13b3-4047-949f-c5d27d50c49a	spark-wordcount-template-swift	democluster	 RUNNING	Delete Job

Displaying 2 items

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Как пользоваться: Horizon

openstack admin admin

Jobs

Status Filter Filter Job Guide x Delete Jobs

☐	ID	Job Template	Cluster	Status	Actions
☐	7a9a1ad8-4ded-4d3e-bc26-bc60535bc581	spark-wordcount-template-swift	Not available	Succeeded	Delete Job
☐	dc9191f6-13b3-4047-949f-c5d27d50c49a	spark-wordcount-template-swift	democluster	Succeeded	Delete Job

Displaying 2 items

Guides

Clusters

Jobs

Cluster Templates

Node Group Templates

Job Templates

Как пользоваться: CLI

Сначала подготовьте свою систему:

1. Сделайте отдельный `virtualenv` для Python и активируйте его.
2. Установите в него `pip`.
3. Склонируйте проект <https://github.com/openstack/python-saharaclient>
4. Зайдите в папку с проектом.
5. `pip install -r requirements.txt`
6. `python setup.py install`

Теперь у вас есть консольная программа `sahara` и необходимый SDK, в случае если вы захотите пользоваться API

Как пользоваться: CLI

Теперь создадим кластер.

Для начала создадим текстовые файлы для создания шаблонов master и slave узлов.

```
{
  "plugin_name": "cdh",
  "hadoop_version": "5.3.0",
  "name": "cloudera-master-cli",
  "flavor_id": "4",
  "node_processes": [
    "CLLOUDERA_MANAGER",
    "YARN_JOBHISTORY",
    "OOZIE_SERVER",
    "HDFS_DATANODE",
    "YARN_RESOURCEMANAGER",
    "HDFS_SECONDARYNAMENODE",
    "HDFS_NAMENODE",
    "SPARK_YARN_HISTORY_SERVER",
    "YARN_NODEMANAGER"
  ],
  "floating_ip_pool": "c75ddbac-ed68-491c-a53a-6b36b0490dcf"
}
```

```
{
  "plugin_name": "cdh",
  "hadoop_version": "5.3.0",
  "name": "cloudera-slave-cli",
  "node_processes": [
    "HDFS_DATANODE",
    "YARN_NODEMANAGER"
  ],
  "flavor_id": "4",
  "floating_ip_pool": "c75ddbac-ed68-491c-a53a-6b36b0490dcf"
}
```


Как пользоваться: CLI

Так создаются шаблоны групп узлов при помощи этих файлов:

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-  
password <your_password> --os-tenant-name admin node-group-template-create --json  
slave.json
```

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-  
password <your_password> --os-tenant-name admin node-group-template-create --json  
master.json
```

В качестве результата вернутся две таблицы с параметрами созданных шаблонов:

Property	Value
volume_local_to_instance	False
volumes_availability_zone	None
availability_zone	None
volume_mount_prefix	/volumes/disk
plugin_name	cdh
floating_ip_pool	c75ddbdc-ed68-491c-a53a-6b36b0490dcf
is_default	False
volumes_size	0
use_autoconfig	True
is_proxy_gateway	False
volumes_per_node	0
is_public	False
hadoop_version	5.3.0
id	be1e3f80-2f59-4a19-85ac-76a2f8a3733c
security_groups	None
name	cloudera-slave-cli
tenant_id	9a5893f7db304ea682fdc878cb63180
created_at	2015-08-28T13:07:31
volume_type	None
is_protected	False
node_configs	{}
flavor_id	4
node_processes	HDFS_DATANODE, YARN_NODEMANAGER
auto_security_group	False

Как пользоваться: CLI

Из этих таблиц мы берем id и создаем шаблон для создания кластера:

```
{
  "name": "cloudera-cli",
  "plugin_name": "cdh",
  "hadoop_version": "5.3.0",
  "node_groups": [
    {
      "count": 3,
      "node_group_template_id": "be1e3f80-2f59-4a19-85ac-76a2f8a3733c",
      "name": "slave"
    },
    {
      "count": 1,
      "node_group_template_id": "01a12393-126d-4ae0-b0c5-dc20275f8863",
      "name": "master"
    }
  ],
  "neutron_management_network": "e1503cae-7a1f-412d-8b46-04c8037c90aa",
  "cluster_configs": {}
}
```

И посылаем запрос на создание шаблона:

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-
password <your_password> --os-tenant-name admin cluster-template-create --json cluster-
template.json
```

Как пользоваться: CLI

Нам снова вернут таблицу с получившимися параметрами, но она слишком большая, чтобы ее показать. Запоминаем id и создаем еще один json-файл:

```
{  
  "plugin_name": "cdh",  
  "hadoop_version": "5.3.0",  
  "cluster_template_id": "4a7fce83-ef57-4983-bddb-59d193696d40",  
  "default_image_id": "eb4c1bb5-3fce-440d-896b-ebbaa9d6425d",  
  "user_keypair_id": "alexander",  
  "name": "doc-cluster",  
  "neutron_management_network": "e1503cae-7a1f-412d-8b46-04c8037c90aa",  
  "cluster_configs": {}  
}
```

Все, теперь можно запускать:

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-password  
<your_password> --os-tenant-name admin cluster-create --json cluster.json
```

Как пользоваться: CLI

Теперь нам нужно создать Spark-job шаблон и запустить его на получившемся кластере.

Сначала заливаем бинарный файл с нашей программой:

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-password <your_password> --os-tenant-name admin job-binary-data-create --file ~/spark-for-cloudera.jar
```

Нам вернется id файла. Теперь скажем сахаре, что мы будем использовать файл как бинарный файл для запуска (Binary file):

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-password <your_password> --os-tenant-name admin job-binary-create --name spark-hdfs-cli --url internal-db://2e99072e-b1cf-4528-8ff5-bdd1cd598fd1
```

Как пользоваться: CLI

Осталось создать шаблон job:

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-password <your_password> --os-tenant-name admin job-template-create --name spark-hdfs-template-cli --type Spark --main e1d655ac-69c3-457a-bc04-ac8d2ce9327a
```

И наконец-то запустить его:

```
sahara --os-username <username> --os-auth-url http://10.10.10.121:5000/v2.0 --os-password <your_password> --os-tenant-name admin job-create --job-template 0dd96e6f-9345-4354-b5b8-8419e544a584 --cluster 1e15be71-0b0b-4fc2-a693-c9801719e087 --config edp.java.main_class=com.cloudera.sparkwordcount.SparkWordCount --config edp.spark.adapt_for_swift=True --config fs.swift.service.sahara.password="<swift_password>" --config fs.swift.service.sahara.username="<swift_username>" --arg swift://vg-cont/input --arg 2
```

Ура! Задание запущено

И как все это попробовать?

Варианты таковы:

1. Поставить OpenStack версии Liberty
2. Поставить себе devstack (будьте очень осторожны и устанавливайте его себе только на полностью чистые системы)
3. Настройте себе на существующем Openstack Sahara из репозитория на github (крайне не рекомендуется). Кроме того, вам понадобится обновить Heat и Horizon, чтобы все заработало.

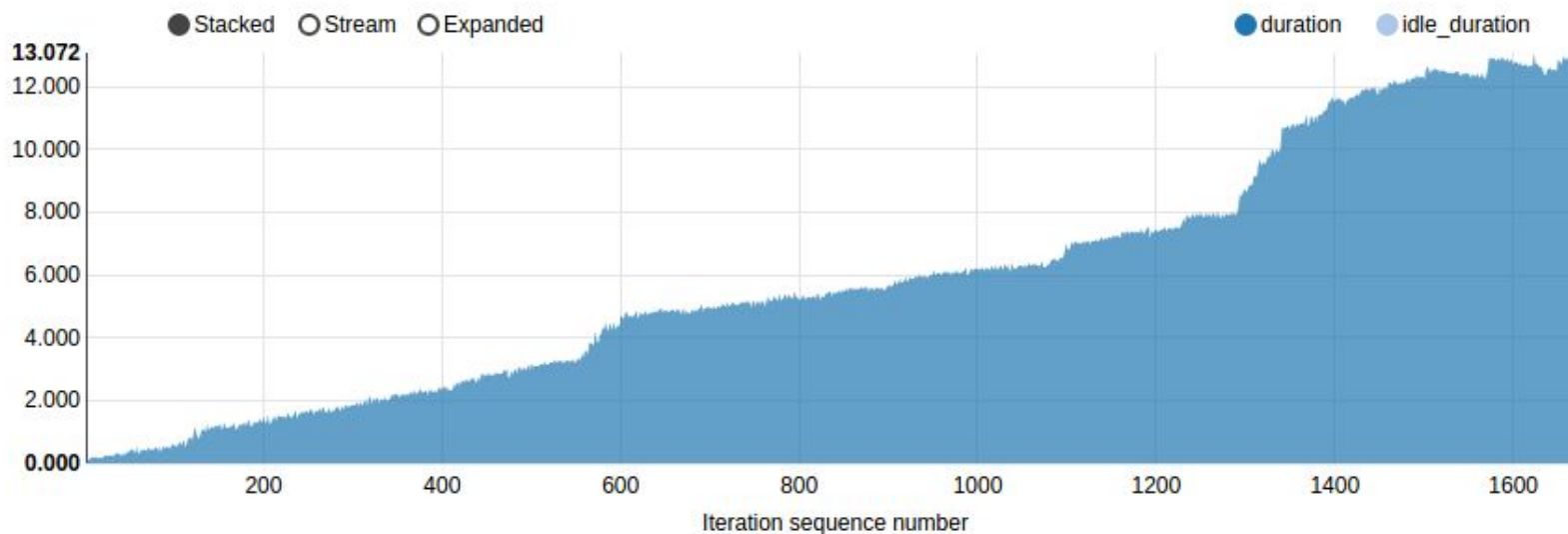
OpenStack Keystone

- Keystone - сервис OpenStack, ответственный за аутентификацию и авторизацию пользователей и сервисов
- Все коммуникации происходят через Keystone
- Плохо справляется с нагрузками

Проблемы с Keystone

Деградация системы при низких нагрузках

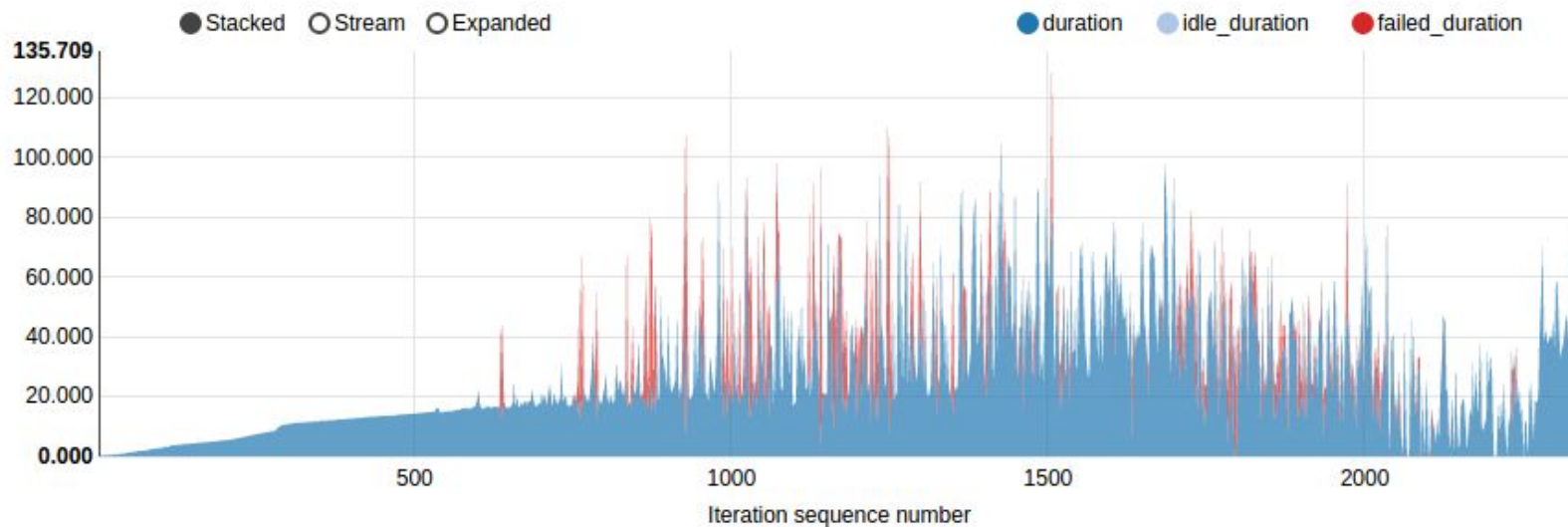
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
total	0.101	5.454	12.347	12.647	13.439	5.913	100.0%	1680



Проблемы с Keystone

Ещё большая деградация

Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
total	0.084	21.058	74.05	92.079	168.745	30.407	87.4%	2340



В проекте Sahara участвовали

Александр Алексиянц (ИСП РАН)

Олег Борисенко (ИСП РАН)

Виталий Гриднев (Mirantis)

Сергей Лукьянов (Mirantis)

Также независимо от нас примерно в то же время появилась реализация для дистрибутива MapR:
Артем Осадчий (MapR)

мы уже установили контакт и приведем реализацию от MapR к общему виду.

*Спасибо за
внимание!*