How-To Library

Informatica

How to Create Cloudera Altus Clusters with a Cluster Workflow on Big Data Management

Abstract

You can implement Cloudera Altus clusters hosted on Amazon Web Services (AWS) with Big Data Management 10.2.1. Create a workflow with Command tasks that run scripts and Mapping tasks that run mappings.

Supported Versions

- Informatica Big Data Management 10.2.1

Table of Contents

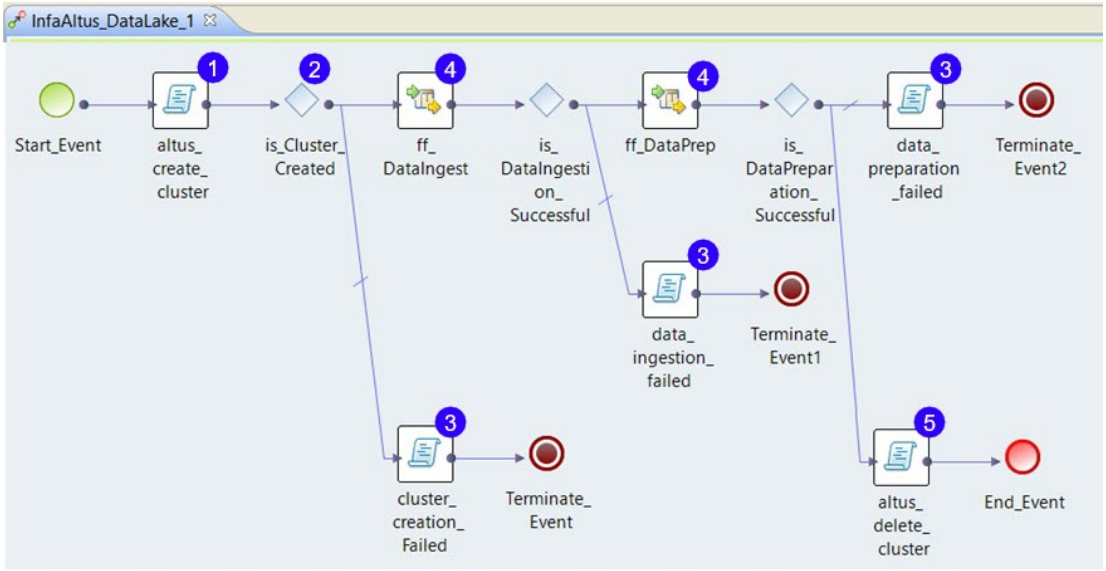
Overview

You can use a workflow to create a Cloudera Altus cluster on Amazon Web Services (AWS) and run Mapping and other tasks.

When you include a Command task to terminate and delete the cluster and the resources it used in the cloud platform, the cluster is called an ephemeral cluster. In an ephemeral cluster strategy, cloud clusters are created, exist for the time it takes for jobs to complete, and then cease to exist when they are brought down.

Create a workflow to implement the strategy. The workflow contains Command tasks that use scripts that spin up and configure a Cloudera Altus cluster on AWS, and Mapping tasks that run mappings. When the mapping runs are complete, an additional Command task can then terminate the cluster, so that you use cluster resources only when you need them.

The following image shows a typical workflow to create an Altus cluster, run mappings, and terminate the cluster:



The workflow contains Command and Mapping tasks that run scripts, as well as gateway events that evaluate the success or failure of tasks. The following descriptions explain elements in the image:

1. The script creates the cluster and configures connections between the cluster and the Informatica domain.

   The Command task contains the altusCreateCluster.sh script. The script uses Cloudera APIs to create an Altus cluster on AWS EC2 instances. The script contains references to the Informatica domain on AWS and to properties files that you edit and upload to AWS.

   The workflow has several gateway events. When the previous task fails, the workflow goes on to a command task that terminates the workflow.
2. If a Command task fails, another Command task terminates the cluster.
3. The Command task contains a script that deletes connections and terminates the cluster.
4. The workflow runs Mapping tasks, which run mappings on the cluster.
5. You can include a Command task to terminate and delete the cluster and connections and configuration files that are no longer necessary. You might want to do this to conserve cloud resource costs.

   If you do not include a task to terminate and delete the cluster, it continues to run.

Prerequisites

Before you begin, verify the following prerequisites:

- You have an AWS account with a Virtual Private Cloud (VPC) where you have permissions to create resources.

  Read the Cloudera documentation at the following URL for AWS account requirements and settings: https://www.cloudera.com/documentation/altus/topics/altaws_adm_administration.html#aws_account_requirements
- You have installed Big Data Management and the Informatica domain on an Amazon EC2 instance in the VPC.

Download the Script Archive File

Download the script archive file Altus_Script_Files.zip. The archive contains script and properties files to edit with property values for the cluster to create, and other files to enable script files to run.

The Altus_Script_Files.zip archive contains the following files:

Properties files to edit:

- arguments.properties
- tags.list

Script files to edit:

- altusClusterCreate.sh
- altusClusterDelete.sh

Python files that the altusClusterCreate.sh and altusClusterDelete.sh files use:

- infatag.py
- pyUtils.py

Note: It is not necessary to edit the Python files. Simply upload them with the edited files in Step 7.

Download the script archive from the Informatica Network at the following URL: XXXXXXXXXXXX

Create the Workflow

Create a workflow with Command tasks to create an Altus cluster and Mapping tasks to run mappings on the cluster.

Complete the following steps:

Step 1. asdfasdf

Step 2. oiqewrfq

Step 3. 319i3rf

Step 1. Create the Altus Environment on AWS

Create a Cloudera Altus environment in the same VPC as the Informatica domain.

Create the Altus environment with appropriate permissions and resources. For more information, see Cloudera documentation.

As part of the setup, install the Altus client in the VPC. For more information, see Cloudera documentation.

Step 2. Install Software on the Informatica Domain Instance

Install the following software utilities and packages on the Amazon EC2 instance that hosts the domain:

- AWS Command Line Interface (CLI) for Linux

  For more information, see AWS documentation.
- AWS Software Development Kit (SDK) for Python (Boto3)

  For more information, see AWS documentation.
- PIP python manager utility

  For more information, see AWS documentation.

Step 3. Get AWS Account Information

Get the following information from the AWS administrator:

Property	Description
Access Key ID	The access key ID allows the user to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services. For more information, see AWS documentation.

Step 4. Create Properties Files

Create properties files based on templates. The properties files supply values to workflow Command tasks to create and configure the Altus cluster and connections with the Informatica domain.

Create the following two files from the templates in the Altus_Script_Files.zip archive.

arguments.properties

The arguments.properties file contains properties that the Create Cluster script task uses to create the Altus cluster. Supply a value for each property in the Informatica and Altus sections.

tags.list

The tags.list file allows you to optionally tag the EC2 instances of the Altus cluster. When you tag the EC2 resources, you can easily identify them in resource logs and dashboards. Tags must follow naming conventions. For more information, see AWS documentation.

Step 5. Create Parameters and Configure Mappings

Create parameters for the cluster name and the Hadoop connection. The Create Cluster task uses when creating the cluster and configuring the connection between the domain and the cluster.

Create Mapping Parameters

Parameters enable mappings to inherit and reuse values for mapping properties

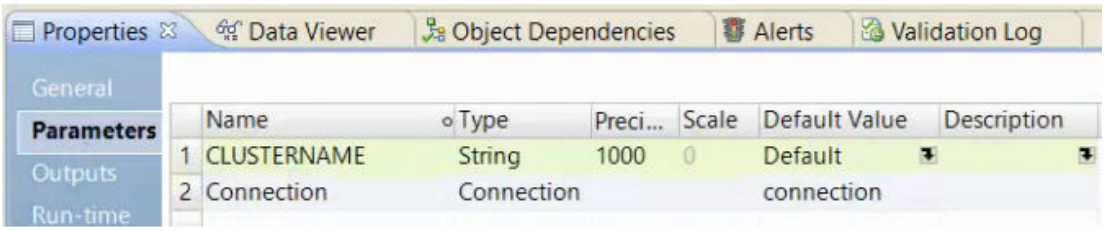
You can open a mapping and create parameters for that mapping and other mappings to use.

To create parameters, complete the following steps:

1. Open a mapping in the Developer tool.
2. In the Properties tab, create two new parameters with the following properties:

Parameter Name	Type	Value
CLUSTERNAME	String	Default
Connection	Connection	connection

The following image shows the configured parameters:



Configure Mapping Run-Time Properties

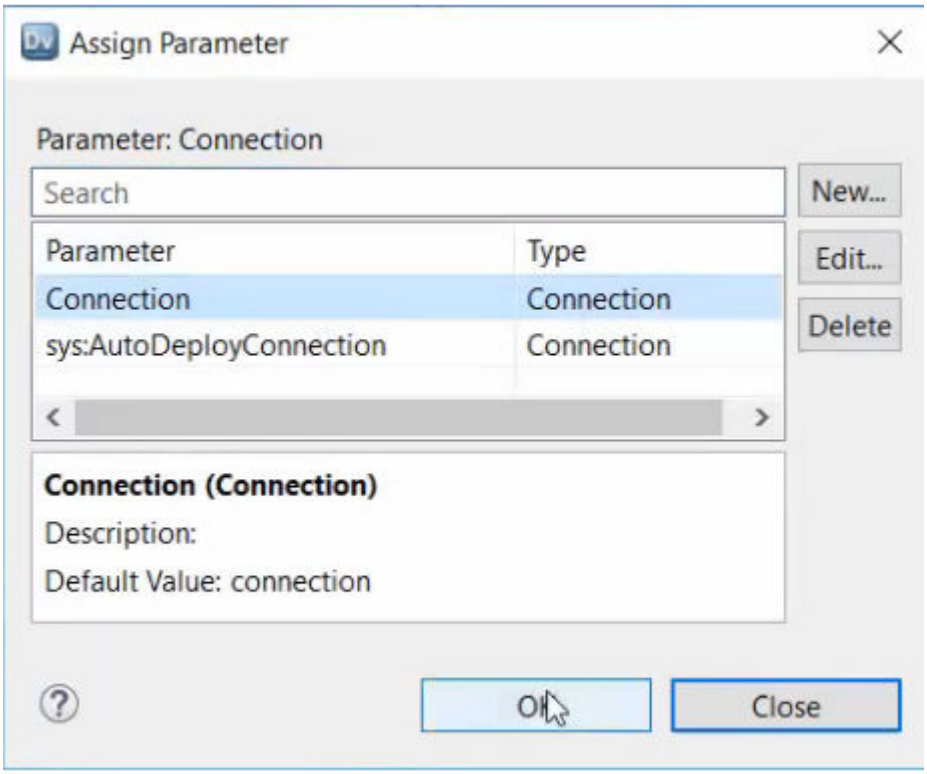
For each mapping that you want to run in the workflow, complete the following steps:

1. In the mapping Run-Time Properties tab, set the following run-time engines in the Validation Environment section:
   - Deselect Native.
   - Select Spark.

   Cloudera Altus clusters support only mappings that use the Spark run-time engine.
2. In the Execution Environment section, complete the following steps to assign parameters to the mapping:
   a. Click **Hadoop** > **Connection** and choose **Assign Parameter** from the drop down list.

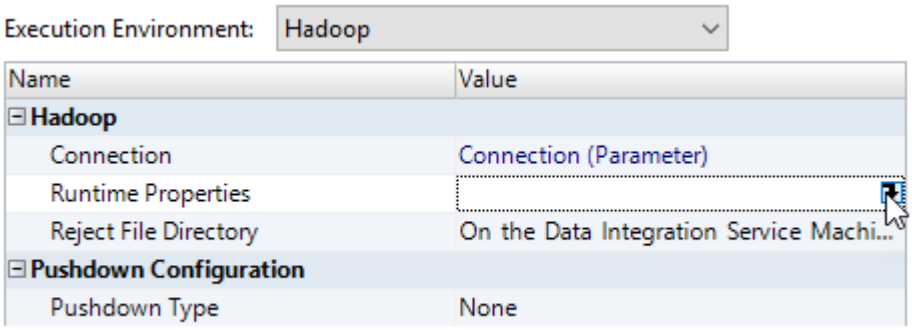
      The **Assign Parameter** dialog box opens.
   b. Select the Connection parameter that you created and click **OK**.

      The following image shows how the Connection parameter appears in the dialog box:



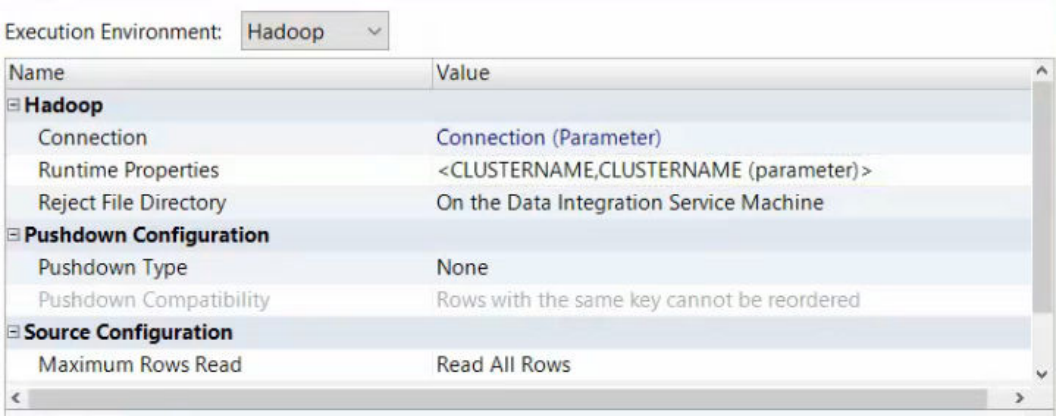
3. Click in the Hadoop Runtime Properties value pane to display the Execution Parameters dialog box.

   The following image shows the selection icon to click:



4. In the **Execution Parameters** dialog box, complete the following steps to create a new parameter:
   a. Click **New**.
   b. Name the new parameter CLUSTERNAME.
   c. Click the selection arrow on the right side of the value pane to display the **Parameters** list.
   d. Click **Assign parameter**.
   e. Select the CLUSTERNAME parameter that you created in step 4b and click **OK**.

   The following image shows the configured run-time mapping properties:



5. Save the mapping.

Step 6. Create the Workflow Task Script Files

Create script files for workflow tasks to use to create the Altus cluster and to perform other workflow tasks.

Workflow Command tasks use scripts that are contained in files that you upload to the AWS cloud platform. Base the script files on templates in the Altus_Script_Files.zip archive.

Create script files for the following Command tasks:

- Edit the the altusClusterCreate.sh script for the Create Cluster Command task.

- Optional gateway events. You can create scripts to handle events that gateway events trigger. The following examples describe Command tasks that appear in the sample workflow in the Overview:
  - Cluster Creation Failed. If you create a gateway task to detect cluster creation failure, this script terminates the workflow.
  - Data Ingestion Failed. If you created a gateway task to detect data ingestion failure, this script terminates the workflow. Note that the cluster has been successfully created and is running, unless you include a cluster termination section in this script.
  - Data Preparation Failed. If you created a gateway task to detect data preparation failure, this script terminates the workflow. Note that the cluster has been successfully created and is running, unless you include a cluster termination section in this script.
- Optional Delete Cluster Command task. This task runs the altusClusterDelete.sh script to terminate and delete the cluster and the connections that it used. If you do not include a Delete Cluster task, the cluster continues to run after the workflow completes tasks.

Step 7. Upload Scripts and Properties Files to the EC2 Instance

The cluster workflow uses scripts and properties files to create and configure the Altus cluster and its resources. Upload the following files to the EC2 instance that hosts the domain:

- The Create Cluster script altusClusterCreate.sh.
- Scripts to handle gateway events, if you created gateway events in the workflow.
- The Delete Cluster script altusClusterDelete.sh.
- Python scripts (infatag.py and pyUtils.py) that the Create Cluster and Delete Cluster scripts use.
- The arguments.properties file that you configured in Step 4.
- Optional: The tags.list file that you configured in Step 4.

Step 8. Create the Workflow, Configure Parameters, and Add Command Tasks

Create the workflow, create parameters for workflow tasks to use. Add Command tasks that run scripts.

When you create workflow parameters, any Command task and its script can use these parameters.

1. Create a workflow in the Developer tool.
2. Click on the Parameters tab of the Workflow properties and create the following parameters:

   clusterName

   Name of the Altus cluster to create.

   nodeCount

   Number of EC2 nodes in the cluster. Create at least 2 nodes.

   instanceType

   EC2 instance type for cluster nodes. Recommended type: m4.xlarge.

   For more information about the EC2 node types you can use, see Altus documentation.

   cdhVersion

   Cloudera version to use with the cluster. Specify CDH511 or later.

   altusEnvironment

   Name of the Altus environment that you created in Step 1.

serviceType

Run-time engine types that can run mappings. Set the value to MULTI to enable the cluster to run jobs using the Spark and Hive engines.

logFile

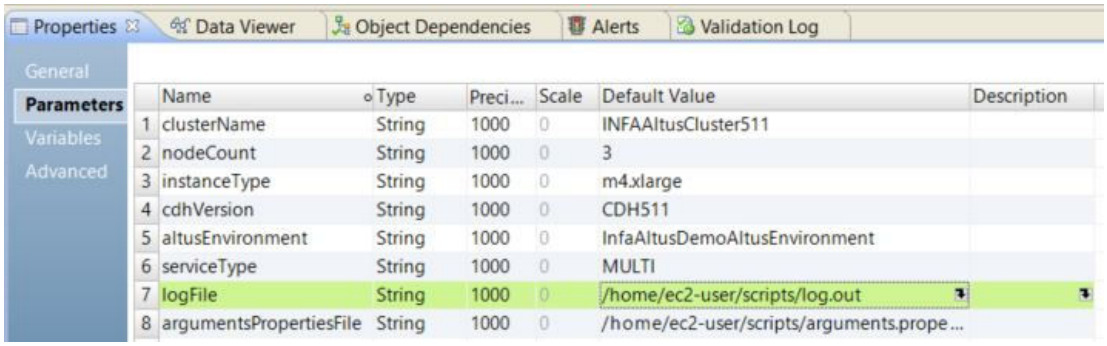
Location of the log file to create for the cluster creation process.

argumentsPropertiesFile

Location of the arguments.properties file that you uploaded in Step 7.

3. Add Command tasks that run the scripts that you uploaded in Step 7. See the Overview of this article for a sample design for the workflow.

The following image shows sample configured parameters:



	Name	Type	Preci...	Scale	Default Value	Description
1	clusterName	String	1000	0	INFAAltusCluster511	
2	nodeCount	String	1000	0	3	
3	instanceType	String	1000	0	m4.xlarge	
4	cdhVersion	String	1000	0	CDH511	
5	altusEnvironment	String	1000	0	InfaAltusDemoAltusEnvironment	
6	serviceType	String	1000	0	MULTI	
7	logFile	String	1000	0	/home/ec2-user/scripts/log.out	
8	argumentsPropertiesFile	String	1000	0	/home/ec2-user/scripts/arguments.prope...	

Step 9. Add Mapping Tasks in the Workflow

The workflow uses Mapping tasks to run mappings.

You can use mappings that you prepared in Step 5 of this article.

To add Mapping tasks, complete the following steps:

1. Drag a Mapping task from the task list to the workflow editor.

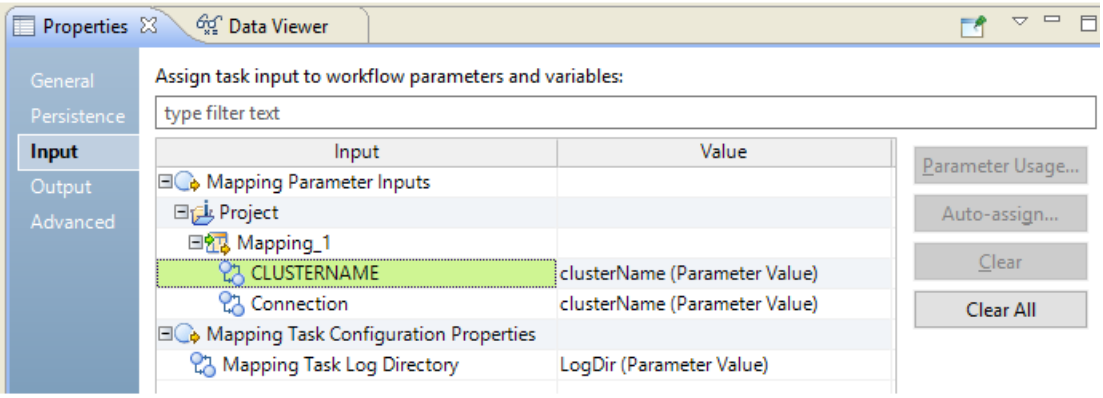
   The Mapping Task dialog box opens.
2. Name the Mapping task.
3. Select one of the mappings that you configured in Step 5 to run with the Mapping task.

   Click **Browse** next to the Mapping property, select a mapping, and click **Finish**.
4. Select the **Input** properties tab and assign the following properties:

Input	Value
CLUSTERNAME	clusterName (Parameter Value)
Connection	clusterName (Parameter Value)

5. Save the Mapping task.

The following image shows the configured Input properties:



Step 10. Set Gateway Events in the Workflow

Gateway events enable the workflow to detect and handle process failures. When a failure occurs, the workflow branches to one of the Command tasks that you created to handle the failure.

Create gateway events in the workflow for each of the gateway event scripts that you created and uploaded.

Step 11: Add a Command Task to Delete the Cluster

Create a Command task if you want to delete the cluster at the end of the workflow.

Populate the Command task with the following command to enable the task to run the deleteAltusCluster.sh script:

```
sh /home/etc-user/scripts/altusClusterDelete.sh -c ${par:ClusterName} -f $
{par:arguments.properties} -o ${par:logFile}
```

The script terminates the cluster, deletes the Hadoop connection and the cluster configuration, then deletes the cluster.

Run the Workflow and Monitor Workflow Logs

Deploy the workflow to the Data Integration Service, which runs it automatically.

The workflow runs and executes the tasks that you configured.

Use the Administrator tool to monitor workflow tasks. You can also monitor jobs on the Cloudera Altus web console.

Monitor Command task execution in the logs at the location that you specified in the Create Cluster command task. Monitor Mapping task execution in the Mapping task log.

For more information about deploying, running, and monitoring workflows, see the *Developer Workflow Guide*.

Author

Mark Pritchard
Principal Technical Writer