Informatica® Big Data Management
10.2.1

Hadoop Integration Guide

Informatica Big Data Management Hadoop Integration Guide
10.2.1
April 2018

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2022-04-29

Table of Contents

Chapter 3: Amazon EMR Integration Tasks. 28

Chapter 4: Azure HDInsight Integration Tasks. 46

##

Preface

The *Informatica Big Data Management™ Integration Guide* is written for the system administrator who is responsible for integrating the native environment of the Informatica domain with a non-native environment, such as Hadoop or Databricks. This guide contains instructions to integrate the Informatica and non-native environments.

Integration tasks are required on the Hadoop cluster, the Data Integration Service machine, and the Developer tool machine. As a result, this guide contains tasks for administrators of the non-native environments, Informatica administrators, and Informatica mapping developers. Tasks required by the Hadoop or Databricks administrator are directed to the administrator.

Use this guide for new integrations and for upgrades. The instructions follow the same task flow. Tasks required for upgrade indicate that they are for upgrade.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at https://search.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Introduction to Hadoop Integration

This chapter includes the following topics:

Hadoop Integration Overview

You can integrate the Informatica domain with the Hadoop cluster through Big Data Management.

The Data Integration Service automatically installs the Hadoop binaries to integrate the Informatica domain with the Hadoop environment. The integration requires Informatica connection objects and cluster configurations. A cluster configuration is a domain object that contains configuration parameters that you import from the Hadoop cluster. You then associate the cluster configuration with connections to access the Hadoop environment.

Perform the following tasks to integrate the Informatica domain with the Hadoop environment:

1. Install or upgrade to the current Informatica version.
2. Perform pre-import tasks, such as verifying system requirements and user permissions.
3. Import the cluster configuration into the domain. The cluster configuration contains properties from the *-site.xml files on the cluster.
4. Create a Hadoop connection and other connections to run mappings within the Hadoop environment.
5. Perform post-import tasks specific to the Hadoop distribution that you integrate with.

When you run a mapping, the Data Integration Service checks for the binary files on the cluster. If they do not exist or if they are not synchronized, the Data Integration Service prepares the files for transfer. It transfers the files to the distributed cache through the Informatica Hadoop staging directory on HDFS. By default, the staging directory is /tmp. This transfer process replaces the requirement to install distribution packages on the Hadoop cluster.

How to Use This Guide

This guide contains instructions to integrate the Informatica and Hadoop environments.

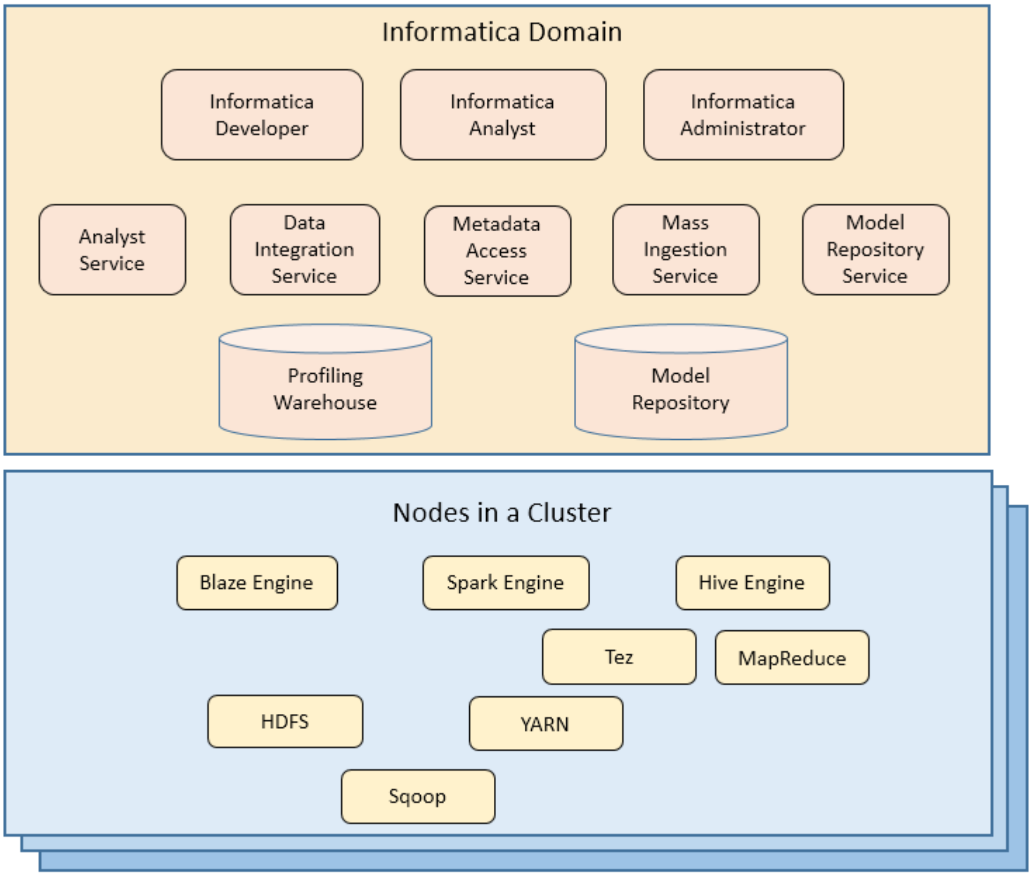
Integration tasks are required on the Hadoop cluster, the Data Integration Service machine, and the Developer tool machine. As a result, this guide contains tasks for Hadoop administrators, Informatica administrators, and Informatica mapping developers. Tasks required by the Hadoop administrator are directed to the Hadoop administrator.

Use this guide for new integrations and for upgrades. The instructions follow the same task flow. Tasks required for upgrade indicate that they are for upgrade.

Big Data Management Component Architecture

The Big Data Management components include client tools, application services, repositories, and third-party tools that Big Data Management uses for a big data project. The specific components involved depend on the task you perform.

The following image shows the components of Big Data Management:



Hadoop Environment

Big Data Management can connect to clusters that run different Hadoop distributions. Hadoop is an open-source software framework that enables distributed processing of large data sets across clusters of machines. You might also need to use third-party software clients to set up and manage your Hadoop cluster.

Big Data Management can connect to the supported data source in the Hadoop environment, such as HDFS, HBase, or Hive, and push job processing to the Hadoop cluster. To enable high performance access to files across the cluster, you can connect to an HDFS source. You can also connect to a Hive source, which is a data warehouse that connects to HDFS.

It can also connect to NoSQL databases such as HBase, which is a database comprising key-value pairs on Hadoop that performs operations in real-time. The Data Integration Service pushes mapping and profiling jobs to the Blaze, Spark, or Hive engine in the Hadoop environment.

Big Data Management supports more than one version of some Hadoop distributions. By default, the cluster configuration wizard populates the latest supported version.

Clients and Tools

Based on your product license, you can use multiple Informatica tools and clients to manage big data projects.

Use the following tools to manage big data projects:

Informatica Administrator

Monitor the status of profile, mapping, and MDM Big Data Relationship Management jobs on the Monitoring tab of the Administrator tool. The Monitoring tab of the Administrator tool is called the Monitoring tool. You can also design a Vibe Data Stream workflow in the Administrator tool.

Informatica Analyst

Create and run profiles on big data sources, and create mapping specifications to collaborate on projects and define business logic that populates a big data target with data.

Informatica Developer

Create and run profiles against big data sources, and run mappings and workflows on the Hadoop cluster from the Developer tool.

Application Services

Big Data Management uses application services in the Informatica domain to process data.

Big Data Management uses the following application services:

Analyst Service

The Analyst Service runs the Analyst tool in the Informatica domain. The Analyst Service manages the connections between service components and the users that have access to the Analyst tool.

Data Integration Service

The Data Integration Service can process mappings in the native environment or push the mapping for processing to the Hadoop cluster in the Hadoop environment. The Data Integration Service also retrieves metadata from the Model repository when you run a Developer tool mapping or workflow. The Analyst tool and Developer tool connect to the Data Integration Service to run profile jobs and store profile results in the profiling warehouse.

Mass Ingestion Service

The Mass Ingestion Service manages and validates mass ingestion specifications that you create in the Mass Ingestion tool. The Mass Ingestion Service deploys specifications to the Data Integration Service. When a specification runs, the Mass Ingestion Service generates ingestion statistics.

Metadata Access Service

The Metadata Access Service is a user-managed service that allows the Developer tool to access Hadoop connection information to import and preview metadata. The Metadata Access Service contains information about the Service Principal Name (SPN) and keytab information if the Hadoop cluster uses Kerberos authentication. You can create one or more Metadata Access Services on a node. Based on your license, the Metadata Access Service can be highly available. Informatica recommends to create a separate Metadata Access Service instance for each Hadoop distribution. If you use a common Metadata Access Service instance for different Hadoop distributions, you might face exceptions.

HBase, HDFS, Hive, and MapR-DB connections use the Metadata Access Service when you import an object from a Hadoop cluster. Create and configure a Metadata Access Service before you create HBase, HDFS, Hive, and MapR-DB connections.

Model Repository Service

The Model Repository Service manages the Model repository. The Model Repository Service connects to the Model repository when you run a mapping, mapping specification, profile, or workflow.

Repositories

Big Data Management uses repositories and other databases to store data related to connections, source metadata, data domains, data profiling, data masking, and data lineage. Big Data Management uses application services in the Informatica domain to access data in repositories.

Big Data Management uses the following databases:

Model repository

The Model repository stores profiles, data domains, mapping, and workflows that you manage in the Developer tool. The Model repository also stores profiles, data domains, and mapping specifications that you manage in the Analyst tool.

Profiling warehouse

The Data Integration Service runs profiles and stores profile results in the profiling warehouse.

Integration with Other Informatica Products

To expand functionality and to process data more efficiently, you can use Big Data Management in conjunction with other Informatica products.

Big Data Management integrates with the following Informatica products:

- PowerExchange adapters. Connect to data sources through adapters.
- Enterprise Data Catalog. Perform data lineage analysis for big data sources and targets.
- Enterprise Data Lake. Discover raw data and publish it in a lake as a Hive table.
- Data Quality. Perform address validation and data discovery.
- Data Replication. Replicate change data to a Hadoop Distributed File System (HDFS).

- Data Transformation. Process complex file sources from the Hadoop environment.
- Big Data Streaming. Stream data as messages, and process it as it becomes available.
- Edge Data Streaming. Collect and ingest data in real time to a Kafka queue.

CHAPTER 2

Before You Begin

This chapter includes the following topics:

Read the Release Notes

Read the Release Notes for updates to the installation and upgrade process. You can also find information about known and fixed limitations for the release.

Verify System Requirements

Verify that your environment meets the minimum system requirements for the installation process, disk space requirements, port availability, and third-party software.

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:
https://network.informatica.com/community/informatica-network/product-availability-matrices

Verify Product Installations

Before you begin the Big Data Management integration between the domain and Hadoop environments, verify that Informatica and third-party products are installed.

You must install the following products:

Informatica domain and clients

Install and configure the Informatica domain and the Developer tool. The Informatica domain must have a Model Repository Service, a Data Integration Service, and a Metadata Access Service.

Hadoop File System and MapReduce

The Hadoop installation must include a Hive data warehouse with a non-embedded database for the Hive metastore. Verify that Hadoop is installed with Hadoop File System (HDFS) and MapReduce on each node. Install Hadoop in a single node environment or in a cluster. For more information, see the Apache website: http://hadoop.apache.org.

Database client software

Install the database client software to perform database read and write operations in native mode. Informatica requires the client software to run MapReduce or Tez jobs on the Hive engine. For example, install the Oracle client to connect to an Oracle database.

Verify HDFS Disk Space

When the Data Integration Service integrates the domain with the Hadoop cluster, it uploads the Informatica binaries onto the HDFS.

Verify with the Hadoop administrator that the distributed cache has at least 1.5 GB of free disk space.

Verify the Hadoop Distribution

Verify the version of the Hadoop distribution in the Hadoop environment.

The following table lists the supported distribution versions:

Distribution	Version
Amazon EMR	5.10, 5.14
Azure HDInsight	3.6.x
Cloudera CDH	5.11, 5.12, 5.13, 5.14, 5.15
Hortonworks HDP	2.5.x, 2.6.x
MapR	6.x MEP 5.0.x

Verify Port Requirements

Open a range of ports to enable the Informatica domain to communicate with the Hadoop cluster and the distribution engine.

To ensure access to ports, the network administrator needs to complete additional tasks in the following situations:

- The Hadoop cluster is behind a firewall. Work with the network administrator to open a range of ports that a distribution engine uses.
- The Hadoop environment uses Azure HDInsight. Work with the network administrator to enable VPN between the Informatica domain and the Azure cloud network.

The following table lists the ports to open:

Port	Description
7180	Cluster management web app for Cloudera. Required for Cloudera only.
8020	NameNode RPC. Required for all supported distributions except MapR.
8032	ResourceManager. Required for all distributions.
8080	Cluster management web app. Used by distributions that use Ambari to manage the cluster: HDinsight, Hortonworks.
8088	Resource Manager web app. Required for all distributions.
8443	MapR control system. Required for MapR only.
9080	Blaze monitoring console. Required for all distributions if you run mappings using Blaze.
9083	Hive metastore. Required for all distributions.
12300 to 12600	Default port range for the Blaze distribution engine. A port range is required for all distributions if you run mappings using Blaze.
19888	YARN JobHistory server webapp. Optional for all distributions.
50070	HDFS Namenode HTTP. Required for all distributions.

Note: The network administrators must ensure that the port used by the Metadata Access Service is accessible from the cluster nodes.

Spark Engine Monitoring Port

Spark engine monitoring requires the cluster nodes to communicate with the Data Integration Service over a socket. The Data Integration Service picks the socket port randomly from the port range configured for the domain. You can view the port range in the advanced properties of the primary node. By default, the minimum port number is 12000 and the maximum port number is 13000. The network administrators must ensure that the port range is accessible from the cluster nodes to the Data Integration Service. If the administrators cannot provide a port range access, you can configure the Data Integration Service to use a fixed port with the SparkMonitoringPort custom property. The network administrator must ensure that the configured port is accessible from the cluster nodes to the Data Integration Service.

Uninstall Big Data Management

If you are upgrading Big Data Management from a version earlier than 10.2 and have a previous version of Big Data Management installed on the Hadoop environment, Informatica recommends that you uninstall the previous version.

Perform this task in the following situation:
- You upgraded from a version earlier than 10.2.

Uninstall for Amazon EMR, Azure HDInsight, and MapR

Complete the following prerequisite tasks before you uninstall Big Data Management:

1. Verify that the Big Data Management administrator can run `sudo` commands.
2. If you are uninstalling Big Data Management in a cluster environment, configure the root user to use a passwordless Secure Shell (SSH) connection between the machine where you want to run the Big Data Management uninstall and all of the nodes where Big Data Management is installed.
3. If you are uninstalling Big Data Management in a cluster environment using the `HadoopDataNodes` file, verify that the `HadoopDataNodes` file contains the IP addresses or machine host names of each of the nodes in the Hadoop cluster from which you want to uninstall Big Data Management. The `HadoopDataNodes` file is located on the node from where you want to launch the Big Data Management installation. You must add one IP address or machine host name of the nodes in the Hadoop cluster for each line in the file.

Complete the following tasks to perform the uninstallation:

1. Log in to the machine as root user. The machine you log in to depends on the Big Data Management environment and uninstallation method.
   - To uninstall in a single node environment, log in to the machine on which Big Data Management is installed.
   - To uninstall in a cluster environment using the HADOOP_HOME environment variable, log in to the primary name node.
   - To uninstall in a cluster environment using the `HadoopDataNodes` file, log in to any node.
2. Run the following command to start the uninstallation in console mode:
   ```
   bash InformaticaHadoopInstall.sh
   sh InformaticaHadoopInstall.sh
   ./InformaticaHadoopInstall.sh
   ```
3. Press **y** to accept the Big Data Management terms of agreement.
4. Press **Enter**.
5. Select **3** to uninstall Big Data Management.
6. Press **Enter**.
7. Select the uninstallation option, depending on the Big Data Management environment:
   - Select **1** to uninstall Big Data Management from a single node environment.
   - Select **2** to uninstall Big Data Management from a cluster environment.
8. Press **Enter**.
9. If you are uninstalling Big Data Management in a cluster environment, select the uninstallation option, depending on the uninstallation method:
   - Select **1** to uninstall Big Data Management from the primary name node.
   - Select **2** to uninstall Big Data Management using the `HadoopDataNodes` file.
10. Press **Enter**.
11. If you are uninstalling Big Data Management from a cluster environment from the primary name node, type the absolute path for the Hadoop installation directory. Start the path with a slash.

The uninstaller deletes all of the Big Data Management binary files from the following directory: `/<Big Data Management installation directory>/Informatica`

In a cluster environment, the uninstaller deletes the binary files from all nodes within the Hadoop cluster.

Uninstall for Cloudera CDH

Uninstall Big Data Management on Cloudera from the Cloudera Manager.

1. In Cloudera Manager, browse to **Hosts** > **Parcels** > **Informatica**.
2. Select **Deactivate**.

   Cloudera Manager stops the Informatica Big Data Management instance.
3. Select **Remove**.

   The cluster uninstalls Informatica Big Data Management.

Uninstall for Hortonworks HDP

To uninstall the stack deployment of Big Data Management, you use the Ambari configuration manager to stop and deregister the Big Data Management service, and then perform manual removal of Informatica files from the cluster.

1. In the Ambari configuration manager, select **INFORMATICA BDM** from the list of services.
2. Click the **Service Actions** dropdown menu and select **Delete Service**.
3. To confirm that you want to delete Informatica Big Data Management, perform the following steps:
   a. In the **Delete Service** dialog box, click **Delete**.
   b. In the Confirm Delete dialog box, type `delete` and then click **Delete**.
   c. When the deletion process is complete, click **OK**.

   Ambari stops the Big Data Management service and deletes it from the listing of available services. To fully delete Big Data Management from the cluster, continue with the next steps.
4. In a command window, delete the `INFORMATICABDM` folder from the following directory on the name node of the cluster: `/var/lib/ambari-server/resources/stacks/<Hadoop distribution>/<Hadoop version>/services/`
5. Delete the `INFORMATICABDM` folder from the following location on all cluster nodes where it was installed: `/var/lib/ambari-agent/cache/stacks/<Hadoop distribution>/<Hadoop version>/services`
6. Perform the following steps to remove RPM binary files:
   a. Run the following command to determine the name of the RPM binary archive:

      ```
      rpm -qa |grep Informatica
      ```
   b. Run the following command to remove RPM binary files:

      ```
      rpm -ev <output_from_above_command>
      ```

      For example:

      ```
      rpm -ev InformaticaHadoop-10.1.1-1.x86_64
      ```
7. Repeat the previous step to remove RPM binary files from each cluster node.
8. Delete the following directory, if it exists, from the name node and each client node: `/opt/Informatica/`.
9. Repeat the last step on each cluster node where Big Data Management was installed.
10. On the name node, restart the Ambari server.

Prepare Directories, Users, and Permissions

The Data Integration Service needs access to the Hadoop environment for integration and staging.

Prepare the following directories, users, and permissions:

- Informatica Hadoop staging directory
- Hive warehouse directory
- Hive staging directory
- Blaze engine directories
- Spark engine staging directory
- Reject file directory

Verify and Create Users

The Data Integration Service requires different users to access the Hadoop environment.

Create or verify the following users on each node in the Hadoop cluster:

Hadoop impersonation user

Verify that every node on the cluster has an impersonation user that can be used in a Hadoop connection. Create one if it does not exist. The Data Integration Service impersonates this user to run jobs in the Hadoop environment. If the MapR distribution uses Ticket or Kerberos authentication, the name must match the system user that starts the Informatica daemon and the gid of the user must match the gid of the MapR user.

Service principal name (SPN) for the Data Integration Service

If the cluster uses Kerberos authentication, verify that the SPN corresponding to the cluster keytab file matches the name of the system user that starts the Informatica daemon.

Hadoop staging user

Optionally, create an HDFS user that performs operations on the Hadoop staging directory. If you do not create a staging user, the Data Integration Service uses the operating system user that starts the Informatica daemon.

Blaze user

Optionally, create an operating system user account that the Blaze engine uses to write to staging and log directories. If you do not create a Blaze user, the Data Integration Service uses the Hadoop impersonation user.

Operating system profile user

If operating system profiles are configured for the Data Integration Service, the Data Integration Service runs jobs with permissions of the operating system user that you define in the profile. You can choose to use the operating system profile user instead of the Hadoop impersonation users to run jobs in a Hadoop environment. To use an operating system profile user, you must create a user on each node in the cluster that matches the name on the Data Integration Service machine.

The Data Integration Service also uses the following user:

Mapping impersonation user

A mapping impersonation user is valid for the native run time environment. Use mapping impersonation to impersonate the Data Integration Service user that connects to Hive, HBase, or HDFS sources and targets that use Kerberos authentication. Configure functionality in the Data Integration Service and the mapping properties. The mapping impersonation user uses the following format: <Hadoop service name>/<host name>@<Kerberos realm>

Create an Informatica Hadoop Staging Directory

Optionally, create a directory on HDFS that the Data Integration Service uses to stage the Informatica binary archive files.

By default, the Data Integration Service writes the files to the HDFS directory `/tmp`.

Grant permission to the Hadoop staging user. If you did not create a Hadoop staging user, the Data Integration Services uses the operating system user that starts the Informatica daemon.

Grant Permissions on the Hive Warehouse Directory

Grant access to the absolute HDFS file path of the default database for the hive warehouse.

Grant read and write permissions on the Hive warehouse directory. You can find the location of the warehouse directory in the hive.metastore.warehouse.dir property of the hive-site.xml file. For example, the default might be `/user/hive/warehouse` or `/apps/hive/warehouse`.

Grant permission to the Hadoop impersonation user. Optionally, you can assign -777 permissions on the directory.

Create a Hive Staging Directory

The Blaze, Spark, and Hive engines require access to the Hive staging directory. You can use the default directory, or you can create a directory on HDFS. For example, if you create a directory, you might run the following command:

```
hadoop fs -mkdir /staging
```

Note: If you create a staging directory, update the yarn.app.mapreduce.am.staging-dir property in the mapred-site.xml file.

If you use the default directory or create a directory, you must grant execute permission to the Hadoop impersonation user and the mapping impersonation users.

Create Blaze Engine Directories

Create a blaze user account and directories required by the Blaze engine.

Complete the following tasks to prepare the Hadoop cluster for the Blaze engine:

Create a home directory for the blaze user.

If you created a blaze user, create home directory for the blaze user. For example,

```
hdfs hadoop fs -mkdir /user/blaze
hdfs hadoop fs -chown blaze:blaze /user/blaze
```

If you did not create a blaze user, the Hadoop impersonation user is the default user.

Optionally, create a local services log directory.

By default, the Blaze engine writes the service logs to the YARN distributed cache. For example, run the following command:

```
mkdir -p /opt/informatica/blazeLogs
```

The Blaze configuration advanced properties in the Hadoop connection contains the following custom property for the services log directory:

```
infagrid.node.local.root.log.dir=$HADOOP_NODE_INFA_HOME/blazeLogs
```

$HADOOP_NODE_INFA_HOME gets set to the YARN distributed cache. If you create a directory, you must update the value of the custom property in the Hadoop connection.

Create an aggregated HDFS log directory.

Create a log directory on HDFS to contain aggregated logs for local services. For example:

```
hadoop fs -mkdir -p /var/log/hadoop-yarn/apps/informatica
```

The Blaze configuration advanced properties in the Hadoop connection contains the following custom property for the HDFS log directory:

```
infacal.hadoop.logs.directory=/var/log/hadoop-yarn/apps/informatica
```

Ensure that value of the custom property in the Hadoop connection matches the directory that you created.

Optionally, create a Blaze staging directory.

You can write the logs to the Informatica Hadoop staging directory, or you can create a Blaze staging directory. If you do not want to use the default location, create a staging directory on the HDFS. For example:

```
hadoop fs -mkdir -p /blaze/workdir
```

Note: If you do not create a staging directory, clear the Blaze staging directory property value in the Hadoop connection and the Data Integration Service uses the HDFS directory `/tmp/blaze_<user name>`.

Grant permissions on the local services log directory, aggregated HDFS log directory, and the staging directory.

Grant permission to the following users:

- Blaze user
- Hadoop impersonation user
- Mapping impersonation users

If the blaze user does not have permission, the Blaze engine uses a different user, based on the cluster security and the mapping impersonation configuration.

Create a Spark Staging Directory

When the Spark engine runs job, it stores temporary files in a staging directory.

Optionally, create a staging directory on HDFS for the Spark engine. For example:

```
hadoop fs -mkdir -p /spark/staging
```

If you want to write the logs to the Informatica Hadoop staging directory, you do not need to create a Spark staging directory. By default, the Data Integration Service uses the HDFS directory `/tmp/spark_<user name>`.

Grant permission to the following users:

- Hadoop impersonation user
- SPN of the Data Integration Service
- Mapping impersonation users

Optionally, you can assign -777 permissions on the directory.

Create a Reject File Directory

You can choose to store reject files on HDFS for the Blaze, Spark, and Hive engines.

Reject files can be very large, and you can choose to write them to HDFS instead of the Data Integration Service machine. You can configure the Hadoop connection object to write to the reject file directory.

Grant permission to the following users:

- Blaze user
- Hadoop impersonation user
- Mapping impersonation users

If the blaze user does not have permission, the Blaze engine uses a different user, based on the cluster security and the mapping impersonation configuration.

Configure the Metadata Access Service

Configure the Metadata Access Service to integrate with the Hadoop environment.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded Big Data Management.

The following table describes the Metadata Access Service properties that you need to configure:

Property	Description
Hadoop Kerberos Service Principal Name	Service Principal Name (SPN) of the Metadata Access Service to connect to a Hadoop cluster that uses Kerberos authentication.
Hadoop Kerberos Keytab	The file path to the Kerberos keytab file on the machine on which the Metadata Access Service runs.
Use logged in user as impersonation user	Required if the Hadoop cluster uses Kerberos authentication. If enabled, the Metadata Access Service uses the impersonation user to access the Hadoop environment. Default is false.

Configure the Data Integration Service

Configure the Data Integration Service to integrate with the Hadoop environment.

Perform the following pre-integration tasks:

1. Download Informatica Hadoop binaries to the Data Integration Service machine if the operating systems of the Hadoop environment and the Data Integration Service are different.
2. Configure the Data Integration Service properties, such as the Hadoop staging directory, Hadoop Kerberos service principal name, and the path to the Kerberos keytab file.

Download the Informatica Server Binaries for the Hadoop Environment

If the domain and the Hadoop environments use different supported operating systems, you must configure the Data Integration Service to be compatible with the Hadoop environment. To run a mapping, the local path to the Informatica server binaries must be compatible with the Hadoop operating system.

The Data Integration Service can synchronize the following operating systems: SUSE and Redhat

The Data Integration Service machine must include the Informatica server binaries that are compatible with the Hadoop cluster operating system. The Data Integration Service uses the operating system binaries to integrate the domain with the Hadoop cluster.

You must run the installer to extract the installation binaries into custom Hadoop OS path and then exit the installer.

1. Create a directory on the Data Integration Service host machine to store the Informatica server binaries associated with the Hadoop operating system.
   If the Data Integration Service runs on a grid, Informatica recommends extracting the files to a location that is shared by all services on the grid. If the location is not shared, you must extract the files to all Data Integration Service machines that run on the grid.

   The directory names in the path must not contain spaces or the following special characters: @ | * $ # ! % ( ) { } [ ]
2. Download and extract the Informatica server binaries from the Informatica download site. For example,
   ```
   tar -xvf <Informatica server binary tar file>
   ```
3. Run the installer to extract the installation binaries into the custom OS path.
   Perform the following steps to run the installer:
   - Run the `sh Server/install.bin -DINSTALL_MODE=CONSOLE -DINSTALL_TYPE=0` file.
   - Press **Y** to continue the installation.
   - Press **1** to install Informatica Big Data Suite Products.
   - Press **3** to run the installer.
   - Press **2** to accept the terms and conditions.
   - Press **2** to continue the installation for big data products only.
   - Press **2** to configure the Informatica domain to run on a network with Kerberos authentication.
   - Enter the path and file name of the Informatica license key and press an option to tune the services.
   - Enter the custom Hadoop OS path.
   - Type **Quit** to quit the installation.
4. Set the custom Hadoop OS path in the Data Integration Service and then restart the service
5. Optionally, you can delete files that are not required. For example, run the following command:
   ```
   rm -Rf <Informatica server binary file> ./source/*.7z
   ```

Note: If you subsequently install an Informatica EBF, you must also install it in the path of the Informatica server binaries associated with the Hadoop environment.

Configure Data Integration Service Properties

The Data Integration Service contains properties that integrate the domain with the Hadoop cluster.

The following table describes the Data Integration Service properties that you need to configure:

Property	Description
Hadoop Staging Directory	The HDFS directory where the Data Integration Service pushes Informatica Hadoop binaries and stores temporary files during processing. Default is `/tmp`.
Hadoop Staging User	The HDFS user that performs operations on the Hadoop staging directory. The user requires write permissions on Hadoop staging directory. Default is the operating system user that starts the Informatica daemon.
Custom Hadoop OS Path	The local path to the Informatica server binaries compatible with the Hadoop operating system. Required when the Hadoop cluster and the Data Integration Service are on different supported operating systems. The Data Integration Service uses the binaries in this directory to integrate the domain with the Hadoop cluster. The Data Integration Service can synchronize the following operating systems: - SUSE and Redhat Include the source directory in the path. For example, `<Informatica server binaries>/source`. Changes take effect after you recycle the Data Integration Service. **Note:** When you install an Informatica EBF, you must also install it in this directory.
Hadoop Kerberos Service Principal Name	Service Principal Name (SPN) of the Data Integration Service to connect to a Hadoop cluster that uses Kerberos authentication. Not required for the MapR distribution.
Hadoop Kerberos Keytab	The file path to the Kerberos keytab file on the machine on which the Data Integration Service runs. Not required for the MapR distribution.
JDK Home Directory	The JDK installation directory on the machine that runs the Data Integration Service. Changes take effect after you recycle the Data Integration Service. The JDK version that the Data Integration Service uses must be compatible with the JRE version on the cluster. Required to run Sqoop mappings or mass ingestion specifications that use a Sqoop connection on the Spark engine, or to process a Java transformation on the Spark engine. Default is blank.
Custom Properties	Properties that are unique to specific environments. You can configure run-time properties for the Hadoop environment in the Data Integration Service, the Hadoop connection, and in the mapping. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Hadoop connection or in the mapping. The Data Integration Service processes property overrides based on the following priorities: 1. Mapping custom properties set using infacmd ms runMapping with the `-cp` option 2. Mapping run-time properties for the Hadoop environment 3. Hadoop connection advanced properties for run-time engines 4. Hadoop connection advanced general properties, environment variables, and classpaths 5. Data Integration Service custom properties

Configure Access to Secure Hadoop Clusters

If the Hadoop cluster uses Kerberos authentication or SSL/TLS, you must configure the Informatica domain to access the cluster. If the cluster uses transparent encryption, you must configure the Key Management Server (KMS) for Informatica user access.

Depending on the security implementation on the cluster, you must perform the following tasks:

Cluster uses Kerberos authentication.

You must configure the Kerberos configuration file on the Data Integration Service machine to match the Kerberos realm properties of the Hadoop cluster. Verify that the Hadoop Kerberos properties are configured in the Data Integration Service and the Metadata Access Service.

Cluster uses SSL/TLS.

You must import security certificates to the Data Integration Service and the Metadata Access Service machines.

Cluster uses transparent encryption.

If the transparent encryption uses Cloudera Java KMS, Cloudera Navigator KMS, or Apache Ranger KMS, you must configure the KMS for Informatica user access.

For more information, see the *Informatica Big Data Management Administrator Guide*.

CHAPTER 3

Amazon EMR Integration Tasks

This chapter includes the following topics:

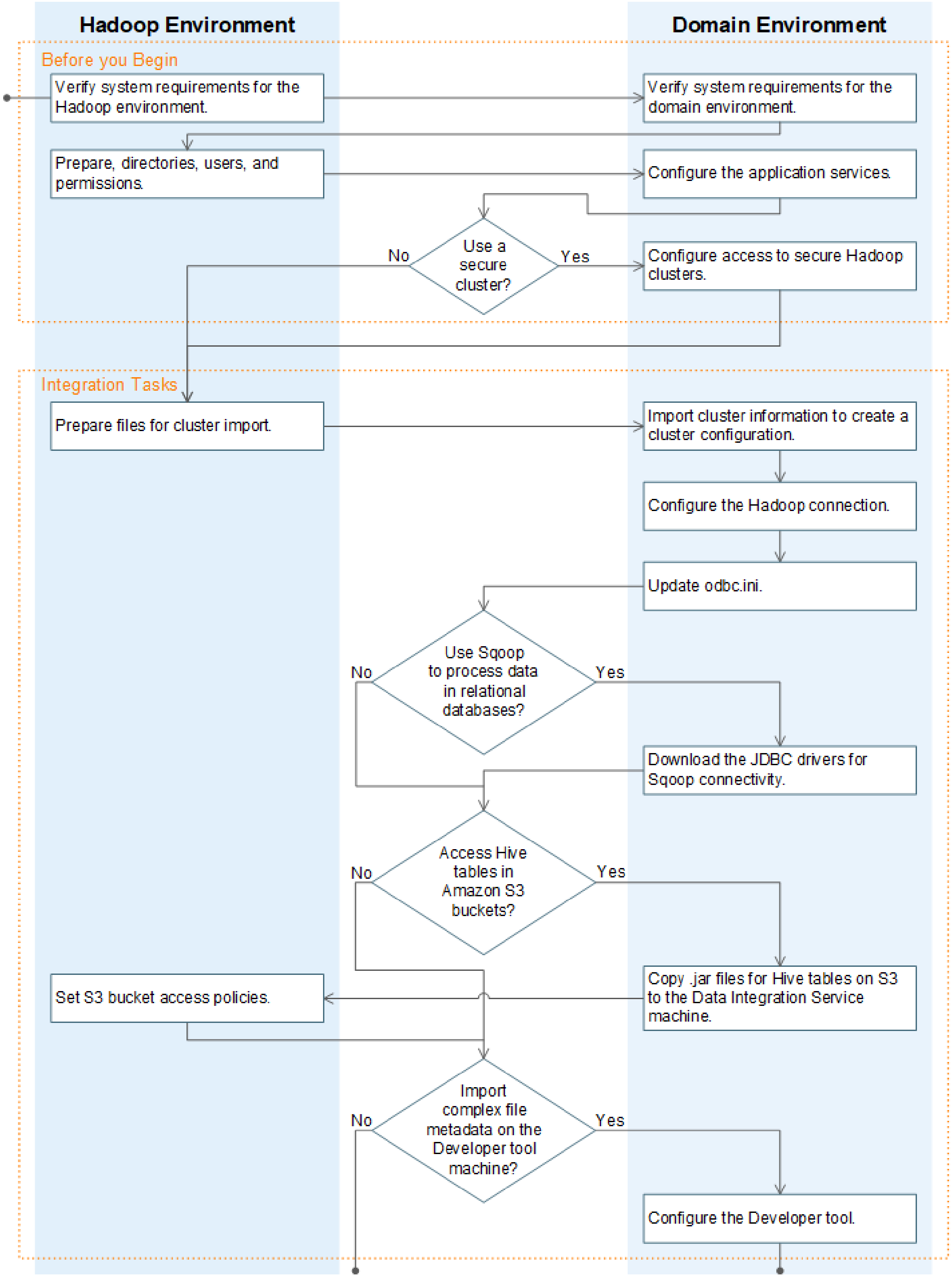
Amazon EMR Task Flows

Depending on whether you want to integrate or upgrade Big Data Management in an Amazon EMR environment, you can use the following flow charts to perform the tasks:

- Integrate the Informatica domain with Amazon EMR for the first time.
- Upgrade Big Data Management from 10.2 to 10.2.1 in an Amazon EMR environment.
- Upgrade Big Data Management from a version earlier than 10.2 to 10.2.1 in an Amazon EMR environment.

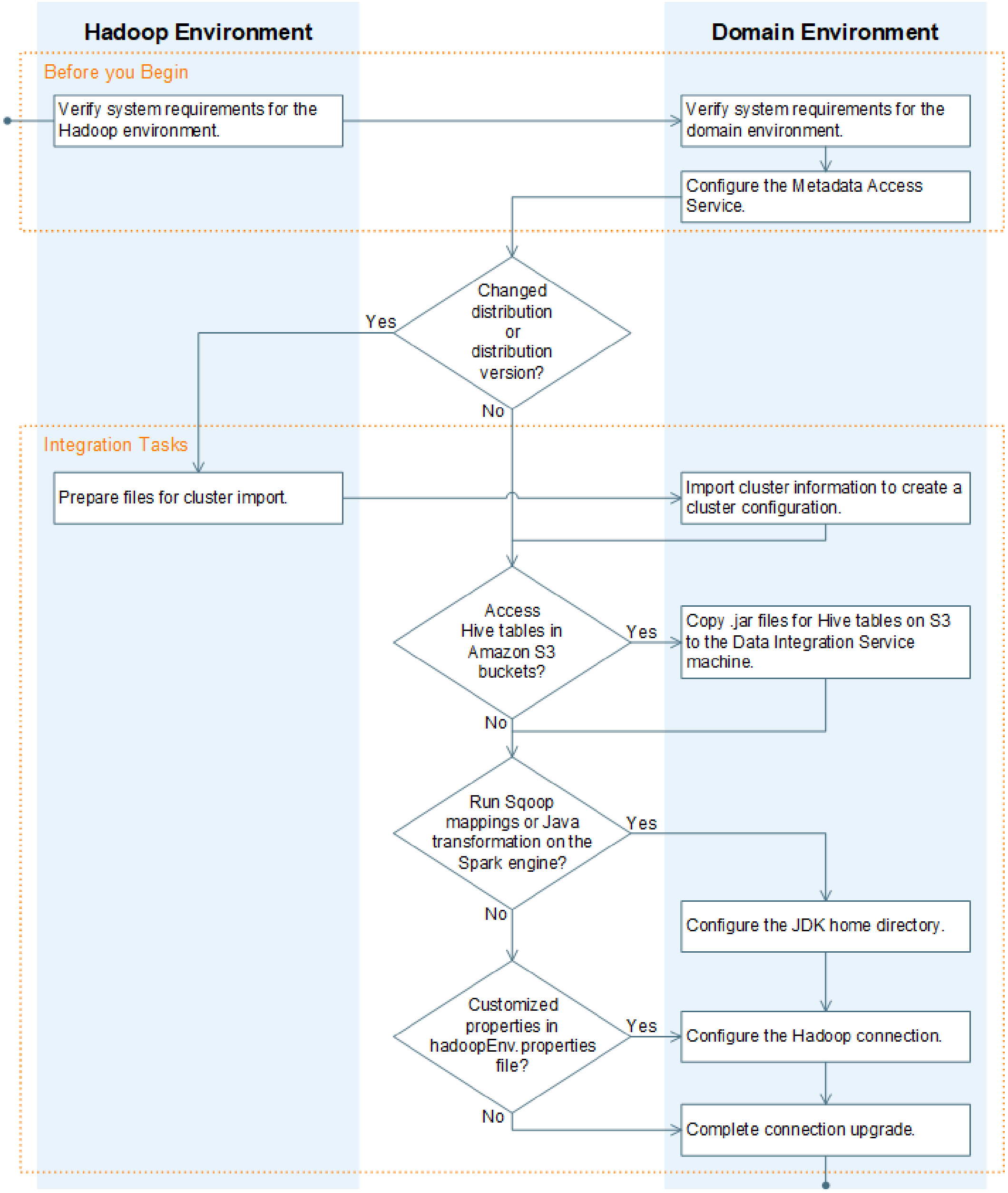
Task Flow to Integrate with Amazon EMR

The following diagram shows the task flow to integrate the Informatica domain with Amazon EMR:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare, directories, users, and permissions.
Configure the application services.
Use a secure cluster?
No
Yes
Configure access to secure Hadoop clusters.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure the Hadoop connection.
Update odbc.ini.
Use Sqoop to process data in relational databases?
No
Yes
Download the JDBC drivers for Sqoop connectivity.
Access Hive tables in Amazon S3 buckets?
No
Yes
Set S3 bucket access policies.
Copy .jar files for Hive tables on S3 to the Data Integration Service machine.
Import complex file metadata on the Developer tool machine?
No
Yes
Configure the Developer tool.

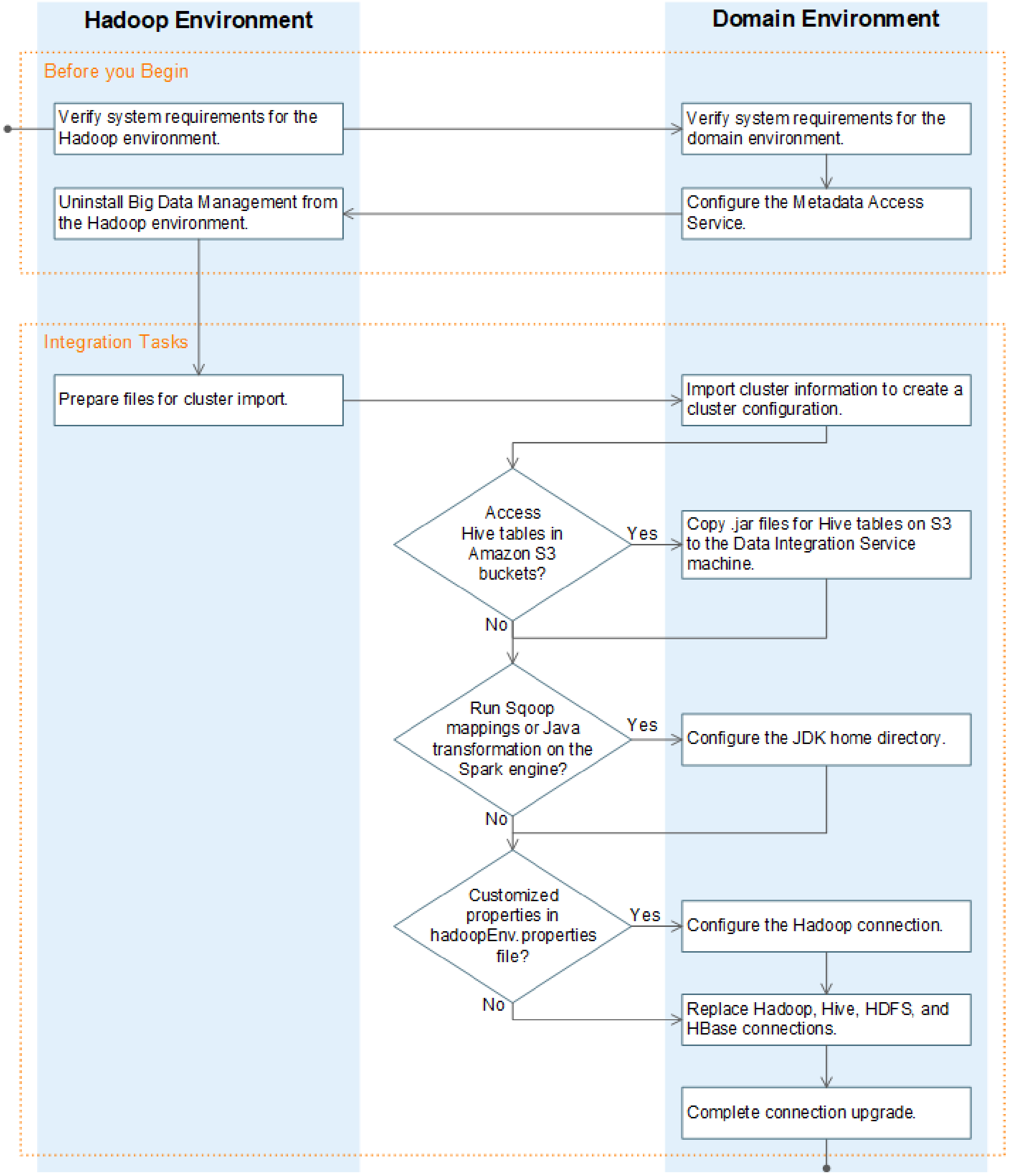
Task Flow to Upgrade from Version 10.2

The following diagram shows the task flow to upgrade Big Data Management 10.2 for Amazon EMR:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Changed distribution or distribution version?
Yes
No
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Access Hive tables in Amazon S3 buckets?
Yes
Copy .jar files for Hive tables on S3 to the Data Integration Service machine.
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
No
Configure the JDK home directory.
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Complete connection upgrade.

Task Flow to Upgrade from a Version Earlier than 10.2

The following diagram shows the task flow to upgrade Big Data Management from a version earlier than 10.2 for Amazon EMR:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Uninstall Big Data Management from the Hadoop environment.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Access Hive tables in Amazon S3 buckets?
Yes
Copy .jar files for Hive tables on S3 to the Data Integration Service machine.
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Replace Hadoop, Hive, HDFS, and HBase connections.
Complete connection upgrade.

Prepare for Cluster Import from Amazon EMR

Before the Informatica administrator can import cluster information to create a cluster configuration in the Informatica domain, the Hadoop administrator must perform some preliminary tasks.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded from a version earlier than 10.2. - You upgraded from 10.2 and changed the distribution or distribution version.

Complete the following tasks to prepare the cluster before the Informatica administrator creates the cluster configuration:

1. Verify property values in *-site.xml files that Big Data Management needs to run mappings in the Hadoop environment.
2. Prepare the archive file to import into the domain.

Note: You cannot import cluster information directly from the Amazon EMR cluster into the Informatica domain.

Configure *-site.xml Files for Amazon EMR

The Hadoop administrator needs to configure *-site.xml file properties before the Informatica administrator imports cluster information into the domain.

core-site.xml

Configure the following properties in the core-site.xml file:

fs.s3.awsAccessKeyID

The ID for the run-time engine to connect to the Amazon S3 file system. Required for the Blaze engine and for the Spark engine if the Data Integration if S3 policy does not allow EMR access.

Note: If the Data Integration Service is deployed on an EC2 instance and the IAM roles and policies allow access to S3 and other resources, this property is not required. If the Data Integration Service is deployed on-premises, then you can choose to configure the value for this property in the cluster configuration on the Data Integration Service after you import the cluster configuration. Configuring the AccessKeyID value on the cluster configuration is more secure than configuring it in core-site.xml on the cluster.

Set to your access ID.

fs.s3.awsSecretAccessKey

The access key for the Blaze and Spark engines to connect to the Amazon S3 file system. Required for the Blaze engine and for the Spark engine if the Data Integration if S3 policy does not allow EMR access.

Note: If the Data Integration Service is deployed on an EC2 instance and the IAM roles and policies allow access to S3 and other resources, this property is not required. If the Data Integration Service is deployed on-premises, then you can choose to configure the value for this property in the cluster configuration on the Data Integration Service after you import the cluster configuration. Configuring the AccessKeyID value on the cluster configuration is more secure than configuring it in core-site.xml on the cluster.

Set to your access key.

fs.s3.enableServerSideEncryption

Enables server side encryption for hive buckets. Required if the S3 bucket is encrypted. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to: TRUE

fs.s3a.server-side-encryption-algorithm

The server-side encryption algorithm for S3. Required if the S3 bucket is encrypted using an algorithm. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to the encryption algorithm used.

fs.s3a.endpoint

URL of the entry point for the web service. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

For example:

```
<property>
   <name>fs.s3a.endpoint</name>
   <value>s3-us-west-1.amazonaws.com</value>
</property>
```

fs.s3a.bucket.BUCKET_NAME.server-side-encryption.key

Server-side encryption key for the S3 bucket. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

For example:

```
<property>
   <name>fs.s3a.bucket.BUCKET_NAME.server-side-encryption.key</name>
   <value>arn:aws:kms:us-west-1*******/value>
   <source>core-site.xml</source>
</property>
```

where BUCKET_NAME is the name of the S3 bucket.

hadoop.proxyuser.<proxy user>.groups

Defines the groups that the proxy user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.<proxy user>.hosts

Defines the host machines that a user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

hadoop.proxyuser.yarn.groups

Comma-separated list of groups that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.yarn.hosts

Comma-separated list of hosts that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

hadoop.security.auth_to_local

Translates the principal names from the Active Directory and MIT realm into local names within the Hadoop cluster. Based on the Hadoop cluster used, you can set multiple rules.

Set to: RULE:[1:$1@$0](^.*@YOUR.REALM)s/^(.*)@YOUR.REALM\.COM$/$1/g

Set to: RULE:[2:$1@$0](^.*@YOUR.REALM\.$)s/^(.*)@YOUR.REALM\.COM$/$1/g

io.compression.codecs

Enables compression on temporary staging tables.

Set to a comma-separated list of compression codec classes on the cluster.

hbase-site.xml

Configure the following properties in the hbase-site.xml file:

hbase.use.dynamic.jars

Enables metadata import and test connection from the Developer tool. Required for an HDInsight cluster that uses ADLS storage or an Amazon EMR 5.8 cluster that uses HBase resources in S3 storage.

Set to: false

zookeeper.znode.parent

Identifies HBase master and region servers.

Set to the relative path to the znode directory of HBase.

hive-site.xml

Configure the following properties in the hive-site.xml file:

hive.cluster.delegation.token.store.class

The token store implementation. Required for HiveServer2 high availability and load balancing.

Set to: org.apache.hadoop.hive.thrift.DBTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: 1

hive.conf.hidden.list

Comma-separated list of hidden configuration properties.

Set to:
javax.jdo.option.ConnectionPassword,hive.server2.keystore.password,fs.s3n.awsAccessKeyId,fs.s3n.awsSecretAccessKey,fs.s3a.access.key,fs.s3a.secret.key,fs.s3a.proxy.password

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.exec.dynamic.partition

Enables dynamic partitioned tables for Hive tables. Applicable for Hive versions 0.9 and earlier.

Set to: TRUE

hive.exec.dynamic.partition.mode

Allows all partitions to be dynamic. Required for the Update Strategy transformation in a mapping that writes to a Hive target. Also required if you use Sqoop and define a DDL query to create or replace a partitioned Hive target at run time.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.txn.manager

Turns on transaction support. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

hive.zookeeper.quorum

Comma-separated list of ZooKeeper server host:ports in a cluster. Required for HiveServer2 high availability.

For example:

```
<property>
<name>hive.zookeeper.quorum</name>
<value>ip-10-21-70-162.us-west-1.compute.internal</value>
</property>
```

kms-site.xml

Configure the following properties in the kms-site.xml file:

hadoop.kms.authentication.kerberos.name.rules

Translates the principal names from the Active Directory and MIT realm into local names within the Hadoop cluster. Based on the Hadoop cluster used, you can set multiple rules.

Set to: RULE:[1:$1@$0](^.*@YOUR.REALM\.COM$)s/^(.*)@YOUR.REALM\.COM$/$1/g

Set to: RULE:[2:$1@$0](^.*@YOUR.REALM\.COM$)s/^(.*)@YOUR.REALM\.COM$/$1/g

mapred-site.xml

Configure the following properties in the mapred-site.xml file:

mapreduce.framework.name

The run-time framework to run MapReduce jobs. Values can be local, classic, or yarn. Required for Sqoop.

Set to: yarn

yarn.app.mapreduce.am.staging-dir

The HDFS staging directory used while submitting jobs.

Set to the staging directory path.

yarn-site.xml

Configure the following properties in the yarn-site.xml file:

yarn.application.classpath

Required for dynamic resource allocation.

"Add spark_shuffle.jar to the class path". The .jar file must contain the class "org.apache.network.yarn.YarnShuffleService."

yarn.nodemanager.resource.memory-mb

The maximum RAM available for each container. Set the maximum memory on the cluster to increase resource memory available to the Blaze engine.

Set to 16 GB if value is less than 16 GB.

yarn.nodemanager.resource.cpu-vcores

The number of virtual cores for each container. Required for Blaze engine resource allocation.

Set to 10 if the value is less than 10.

yarn.scheduler.minimum-allocation-mb

The minimum RAM available for each container. Required for Blaze engine resource allocation.

Set to 6 GB if the value is less than 6 GB.

yarn.nodemanager.vmem-check-enabled

Disables virtual memory limits for containers. Required for the Blaze and Spark engines.

Set to: FALSE

yarn.nodemanager.aux-services

Required for dynamic resource allocation for the Spark engine.

Add an entry for "spark_shuffle."

yarn.nodemanager.aux-services.spark_shuffle.class

Required for dynamic resource allocation for the Spark engine.

Set to: org.apache.spark.network.yarn.YarnShuffleService

yarn.resourcemanager.scheduler.class

Defines the YARN scheduler that the Data Integration Service uses to assign resources.

Set to: org.apache.hadoop.yarn.server.resourcemanager.scheduler

yarn.node-labels.enabled

Enables node labeling.

Set to: TRUE

yarn.node-labels.fs-store.root-dir

The HDFS location to update node label dynamically.

Set to: <hdfs://[Node name]:[Port]/[Path to store]/[Node labels]/>

Prepare the Archive File for Amazon EMR

After you verify property values in the *-site.xml files, create a .zip or a .tar file that the Informatica administrator can use to import the cluster configuration into the domain.

Create an archive file that contains the following files from the cluster:

- core-site.xml
- hbase-site.xml. Required only if you access HBase sources and targets.
- hdfs-site.xml
- hive-site.xml
- mapred-site.xml or tez-site.xml. Include the mapred-site.xml file or the tez-site.xml file based on the Hive execution type used on the Hadoop cluster.
- yarn-site.xml

Note: To import from Amazon EMR, the Informatica administrator must use an archive file.

Edit the hosts File for the Blaze Engine

To run the Blaze engine on every node in the cluster, verify that the /etc/hosts file on every node has entries for all other nodes.

Each node in the cluster requires an entry for the IP address and the fully qualified domain name (FQDN) of all other nodes. For example,

```
127.0.0.1 localhost node1.node.com
208.164.186.1 node1.node.com node1
208.164.186.2 node2.node.com node2
208.164.186.3 node3.node.com node3
```

Changes take effect after you restart the network.

Create a Cluster Configuration

After the Hadoop administrator prepares the cluster for import, the Informatica administrator must create a cluster configuration.

Perform this task in the following situations:

- You are integrating for the first time.
- You upgraded from a version earlier than 10.2.
- You upgraded from 10.2 and changed the distribution or distribution version.

A cluster configuration is an object in the domain that contains configuration information about the Hadoop cluster. The cluster configuration enables the Data Integration Service to push mapping logic to the Hadoop environment. Import configuration properties from the Hadoop cluster to create a cluster configuration.

The import process imports values from *-site.xml files into configuration sets based on the individual *-site.xml files. When you perform the import, the cluster configuration wizard can create Hadoop, HBase, HDFS, and Hive connection to access the Hadoop environment. If you choose to create the connections, the wizard also associates the cluster configuration with the connections.

Note: If you imported the cluster configuration when you installed Enterprise Data Lake with the Informatica domain, you can create the cluster configuration again or refresh the cluster configuration.

For more information about the cluster configuration, see the *Big Data Management Administrator Guide*.

Importing a Cluster Configuration from a File

You can import properties from an archive file to create a cluster configuration.

Before you import from the cluster, you must get the archive file from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. This is the latest version of the Hadoop distribution that Big Data Management supports. When the cluster version differs from the default version, the cluster configuration wizard populates the cluster configuration Hadoop distribution property with the most recent supported version relative to the cluster version. For example, suppose Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the wizard populates the version with 5.10. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.
Method to import the cluster configuration	Choose **Import from file** to import properties from an archive file.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

4. Click **Browse** to select a file. Select the file and click **Open**.
5. Click **Next** and verify the cluster configuration information on the summary page.

Configure the Hadoop Connection

Hadoop connections contain default values for properties such as cluster environment variables, cluster path variables, and advanced properties. Based on the cluster environment and the functionality that you use, you can add to the default values or change the default values of these properties.

For a list of Hadoop connection properties to configure, see "Configuring Hadoop Connection Properties" on page 161.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must configure the ODBCHOME and ODBCINI cluster environment variables in the Hadoop connection. You must then manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

By default, the odbc.ini file contains absolute driver paths. To run ODBC mappings on the Hive engine, you must edit the odbc.ini file and replace the absolute driver paths with relative driver paths.

You can access the odbc.ini file from the following directory on the machine that runs the Data Integration Service:

```
$INFA_HOME/ODBC7.1/
```

Replace the absolute driver paths with relative driver paths. For instance, if you use the DataDirect Greenplum Wire Protocol driver, by default, the odbc.ini file contains the following driver entries:

```
[Greenplum Wire Protocol]
Driver=/data/opt/cloudera/parcels/INFORMATICA/ODBC7.1/lib/DWgplm27.so
Description=DataDirect 7.1 Greenplum Wire Protocol
```

Update the driver entries as follows to replace the absolute driver path with a relative driver path:

```
[Greenplum Wire Protocol
Driver=./lib/DWgplm27.so
Description=DataDirect 7.1 Greenplum Wire Protocol
```

Changes take effect after you recycle the Data Integration Service.

Download the JDBC Drivers for Sqoop Connectivity

To configure Sqoop connectivity for relational databases, you must download JDBC driver .jar files.

1. Download any Type 4 JDBC driver that the database vendor recommends for Sqoop connectivity.

   Note: The DataDirect JDBC drivers that Informatica ships are not licensed for Sqoop connectivity.

2. To optimize the Sqoop mapping performance on the Spark engine while writing data to an HDFS complex file target of the Parquet format, download the following .jar files:
   - `parquet-hadoop-bundle-1.6.0.jar` from https://mvnrepository.com/artifact/com.twitter/parquet-hadoop-bundle
   - `parquet-avro-1.6.0.jar` from https://mvnrepository.com/artifact/com.twitter/parquet-avro

- `parquet-column-1.5.0.jar` from https://mvnrepository.com/artifact/org.apache.parquet/parquet-column

3. Copy all of the .jar files to the following directory on the machine where the Data Integration Service runs:

   ```
   <Informatica installation directory>\externaljdbcjars
   ```

   At run time, the Data Integration Service copies the .jar files to the Hadoop distribution cache so that the .jar files are accessible to all nodes in the cluster.

Configure the Files for Hive Tables on S3

To run mappings with Hive sources or targets on S3, you need to configure the files from the master node to the Data Integration Service machine.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded Big Data Management and changed the distribution version.

You can perform one of the following steps to configure the files:

Copy the .jar file

Get .jar files from the Hadoop administrator. The following files are on the master node in the Hadoop cluster:

- For integration with EMR 5.10, copy `emrfs-hadoop-assembly-2.20.0.jar`.
- For integration with EMR 5.14, copy `emrfs-hadoop-assembly-2.23.0.jar`

Copy the .jar files to the following directory on each Data Integration Service machine: `/<Informatica installation directory>/services/shared/hadoop/EMR_<version number>/lib`.

For integration with EMR 5.14, also copy `emrfs-hadoop-assembly-2.23.0.jar` to the following path: `/<Informatica installation directory>/services/shared/hadoop/EMR_<version number>/extras/hive-auxjars`.

Note: If you have upgraded from EMR 5.10 to EMR 5.14, the part of the filepath that includes `EMR_<version number>` remains `EMR_5.10`.

Create a file

Create a `~/.aws/config` on the Data Integration Service machine. The file must contain AWS location.

For example,

```
[default] region=us-west-2
```

Create an environment variable

Create `AWS_CONFIG_FILE` environment variable on the Data Integration Service machine. Set the value to `<EMR_5.10>/conf/aws.default`

Setting S3 Access Policies

The AWS administrator must set S3 access policies to grant users the required access to S3 resources.

S3 access policies allow control of user access to S3 resources and the actions that users can perform. The AWS administrator uses policies to control access and actions for specific users and resources, depending on the use case that mappings and workflows require.

AWS uses a JSON statement for S3 access policies. To set the S3 access policy, determine the principal, actions, and resources to define, then create or edit an existing S3 access policy JSON statement.

For more information about Amazon S3 access policies, see AWS documentation.

Step 1. Identify the S3 Access Policy Elements

Identify the principal, actions, and resources to insert in the access policy.

The following table describes the tags to set in the access policy:

Tag	Description
Principal	The user, service, or account that receives permissions that are defined in a policy. Assign the owner of the S3 bucket resources as the principal. **Note:** The S3 bucket owner and the owner of resources within the bucket can be different.
Action	The activity that the principal has permission to perform. In the sample, the Action tag lists two put actions and one get action. You must specify both get and put actions to grant read and write access to the S3 resource.
Resource	The S3 bucket, or folder within a bucket. Include only resources in the same bucket.

Sample S3 Policy JSON Statement

The following JSON statement contains the basic elements of an S3 bucket access policy:

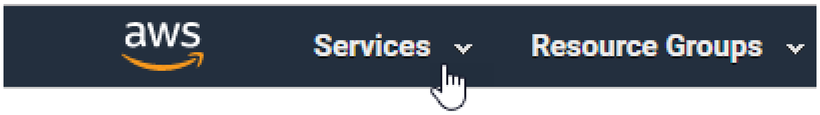
```
{
"Version": "<date>",
"Id": "Allow", "Statement": [
{ "Sid": "<Statement ID>", "Effect": "Allow",
"Principal": {
                "AWS": "arn:aws:iam::<account_2_ID>:<user>"
}
"Action":[
"s3:PutObject","s3:PutObjectAcl",
"s3:GetObject"
]
"Resource": [
"Resource": "arn:aws:s3:::<bucket_1_name>/foldername/*"
]
}
```

Step 2. Optionally Copy an Existing S3 Access Policy as a Template

When the AWS administrator selects a role for cluster users, the AWS console generates a default access policy. After the AWS console generates the default policy, you can copy it and customize it to grant access to specific resources to specific users.

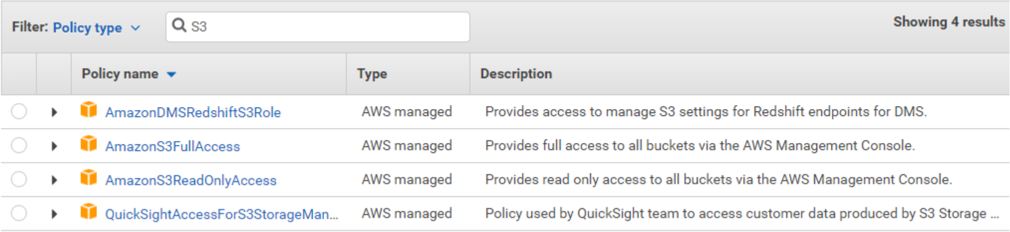
Complete the following steps to copy an existing S3 access policy:

1. In the AWS console, click the **Services** menu.
   The image below shows the **Services** menu in the menu bar:



2. Type "IAM" in the search bar and press Enter.
   The **Welcome to Identity and Access Management** screen opens.
3. In the menu on the left, select **Policies**.
   The console displays a list of existing policies.
4. Type "S3" in the search bar and press Enter.
   The console displays a list of existing S3 access policies.

   The image below shows an example of a list of S3 access policies:



Filter: Policy type S3 Showing 4 results

Policy name	Type	Description
AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings for Redshift endpoints for DMS.
AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.
AmazonS3ReadOnlyAccess	AWS managed	Provides read only access to all buckets via the AWS Management Console.
QuickSightAccessForS3StorageMan...	AWS managed	Policy used by QuickSight team to access customer data produced by S3 Storage ...

5. Click the name of the policy that you want to copy.
   The policy opens in a read-only window.
6. Highlight and copy the policy statement.

After you copy the JSON statement, you can edit it in a text editor or in the bucket policy editor.

Step 3. Create or Edit an S3 Access Policy

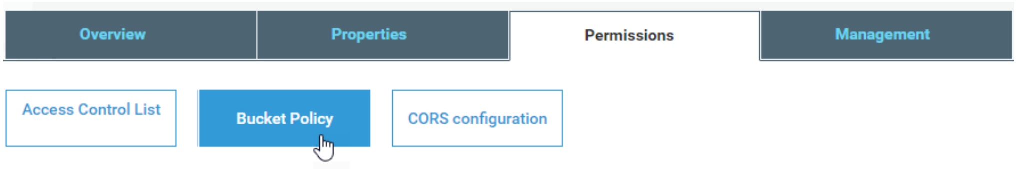
Create an S3 access policy or edit an existing policy. The AWS administrator can enter a JSON statement, based on a template. The administrator can copy and customize the S3 policy from another bucket.

1. In the AWS console, click the **Services** menu.
2. In the **Storage** section, choose **S3**.

   The AWS console displays a list of existing buckets.
3. Use the **search box** to find the bucket you want to set a policy for, and select the bucket from the results.
4. Click the **Permissions** tab, then click **Bucket Policy**.

   The **Bucket Policy Editor** opens.

   The image below shows the **Bucket Policy** button:



5. Type the bucket access policy, or edit the existing policy, and click **Save**.

   AWS applies the access policy to the bucket.

Configure the Developer Tool

To access the Hadoop environment from the Developer tool, the mapping developers must perform tasks on each Developer tool machine.

Configure developerCore.ini

Edit `developerCore.ini` to successfully import local complex files available on the Developer tool machine.

When you import a complex file, such as Avro or Parquet, the imported object includes metadata associated with the distribution in the Hadoop environment. If the file resides on the Developer tool machine, the import process picks up the distribution information from the `developerCore.ini` file. You must edit the `developerCore.ini` file to point to the distribution directory on the Developer tool machine.

You can find `developerCore.ini` in the following directory:

```
<Informatica installation directory>\clients\DeveloperClient
```

Add the following property:

```
-DINFA_HADOOP_DIST_DIR=hadoop\<distribution>_<version>
```

The change takes effect when you restart the Developer tool.

Complete Upgrade Tasks

If you upgraded the Informatica platform, you need to perform some additional tasks within the Informatica domain.

Based on the version that you upgraded from, perform the following tasks:

Configure the JDK home directory.

To run Sqoop mappings or process a Java transformation on the Spark engine, you must configure the JDK Home Directory in the Data Integration Service properties.

Configure the Hadoop connection.

The Hadoop connection contains additional properties. You need to manually update it to include customized configuration in the hadoopEnv.properties file from previous versions.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections. The tasks to complete connection upgrade depends on the Informatica version that you upgraded.

Configure the JDK Home Directory

To leverage Sqoop or to process a Java transformation on the Spark engine, you must install the Java Development Kit (JDK) on the machine that runs the Data Integration Service. Then, you must configure the **JDK Home Directory** property for the Data Integration Service.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

Configure the following property under the Data Integration Service execution options in Informatica Administrator:

JDK Home Directory

Required to run Sqoop mappings or mass ingestion specifications that use a Sqoop connection on the Spark engine, or to process a Java transformation on the Spark engine.

The JDK version that the Data Integration Service uses must be compatible with the JRE version on the cluster.

Set the property to the JDK installation directory on the machine that runs the Data Integration Service. Changes take effect after you recycle the Data Integration Service.

Configure the Hadoop Connection

To use properties that you customized in the hadoopEnv.properties file, you must configure the Hadoop connection properties such as cluster environment variables, cluster path variables, and advanced properties.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

When you run the Informatica upgrade, the installer backs up the existing hadoopEnv.properties file. You can find the backup hadoopEnv.properties file in the following location:

```
<Previous Informatica installation directory>/services/shared/hadoop/<Hadoop distribution
name>_<version>/infaConf
```

Edit the Hadoop connection in the Administrator tool or the Developer tool to include any properties that you manually configured in the hadoopEnv.properties file. The Hadoop connection contains default values for properties such as cluster environment and path variables and advanced properties. You can update the default values to match the properties in the hadoopEnv.properties file.

Replace the Connections with New Connections

If you created connections when you imported the cluster configuration, you need to replace connections in mappings with the new connections.

Perform this task in the following situation:
- You upgraded from a version earlier than 10.2.

The method that you use to replace connections in mappings depends on the type of connection.

Hadoop connection

Run the following commands to replace the connections:

- `infacmd dis replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that are deployed in applications.
- `infacmd mrs replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that you run from the Developer tool.

For information about the infacmd commands, see the *Informatica Command Reference*.

Hive, HDFS, and HBase connections

You must replace the connections manually.

Complete Connection Upgrade

If *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

Perform the following tasks to update the connections:

Update changed properties

Review connections that you created in a previous release to update the values for connection properties. For example, if you added nodes to the cluster or if you updated the distribution version, you might need to verify host names, URIs, or port numbers for some of the properties.

Associate the cluster configuration

The Hadoop, Hive, HDFS, and HBase connections must be associated with a cluster configuration. Complete the following tasks:

1. Run `infacmd isp listConnections` to identify the connections that you need to upgrade. Use `-ct` to list connections of a particular type.
2. Run `infacmd isp UpdateConnection` to associate the cluster configuration with the connection. Use `-cn` to name the connection and `-o clusterConfigID` to associate the cluster configuration with the connection.

For more information about infacmd, see the *Informatica Command Reference*.

CHAPTER 4

Azure HDInsight Integration Tasks

This chapter includes the following topics:

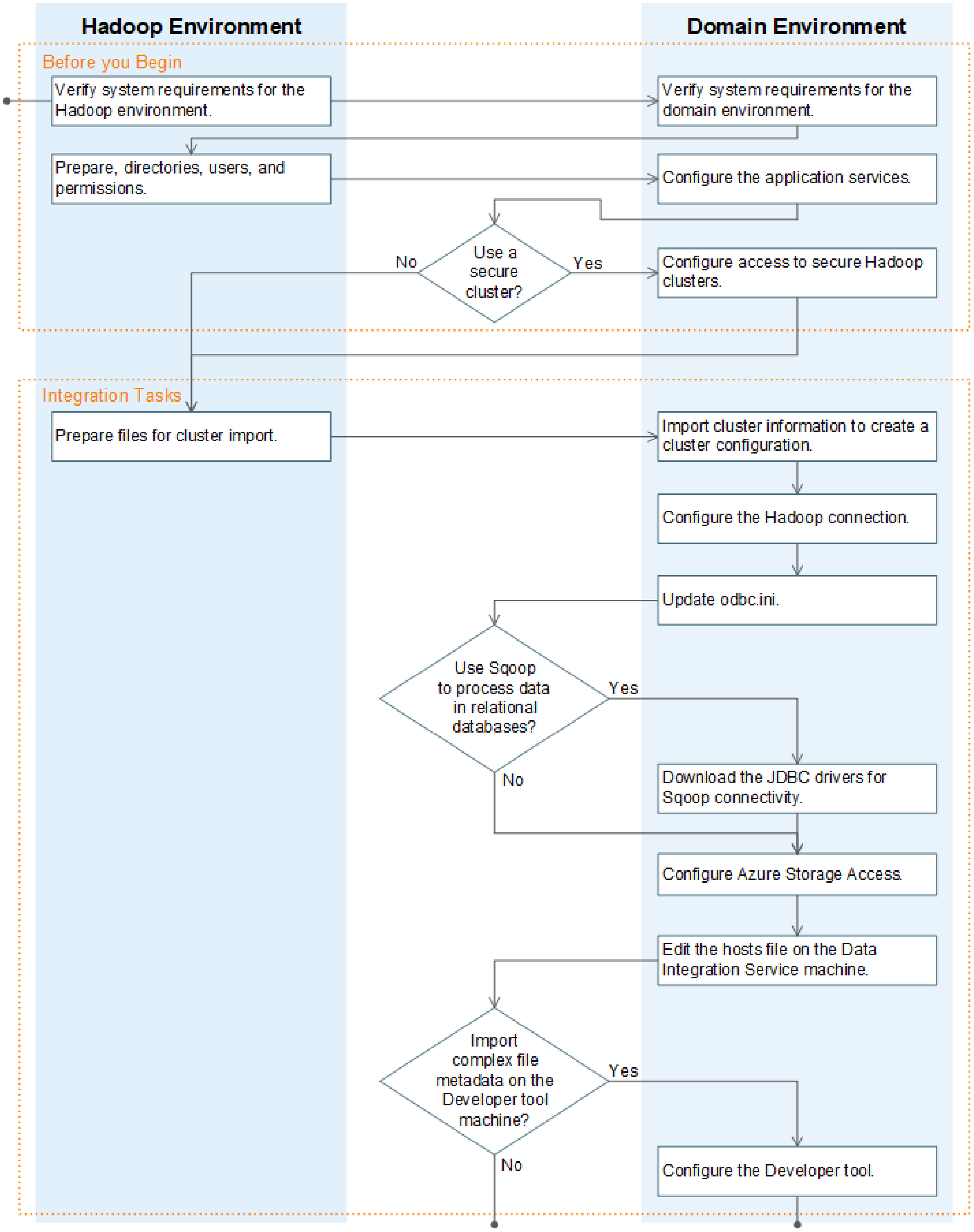
Azure HDInsight Task Flows

Depending on whether you want to integrate or upgrade Big Data Management in an Azure HDInsight environment, you can use the following flow charts to perform the tasks:

- Integrate the Informatica domain with Azure HDInsight for the first time.
- Upgrade Big Data Management from version 10.2 to 10.2.1 in an Azure HDInsight environment.
- Upgrade Big Data Management from a version earlier than 10.2 to 10.2.1 in an Azure HDInsight environment.

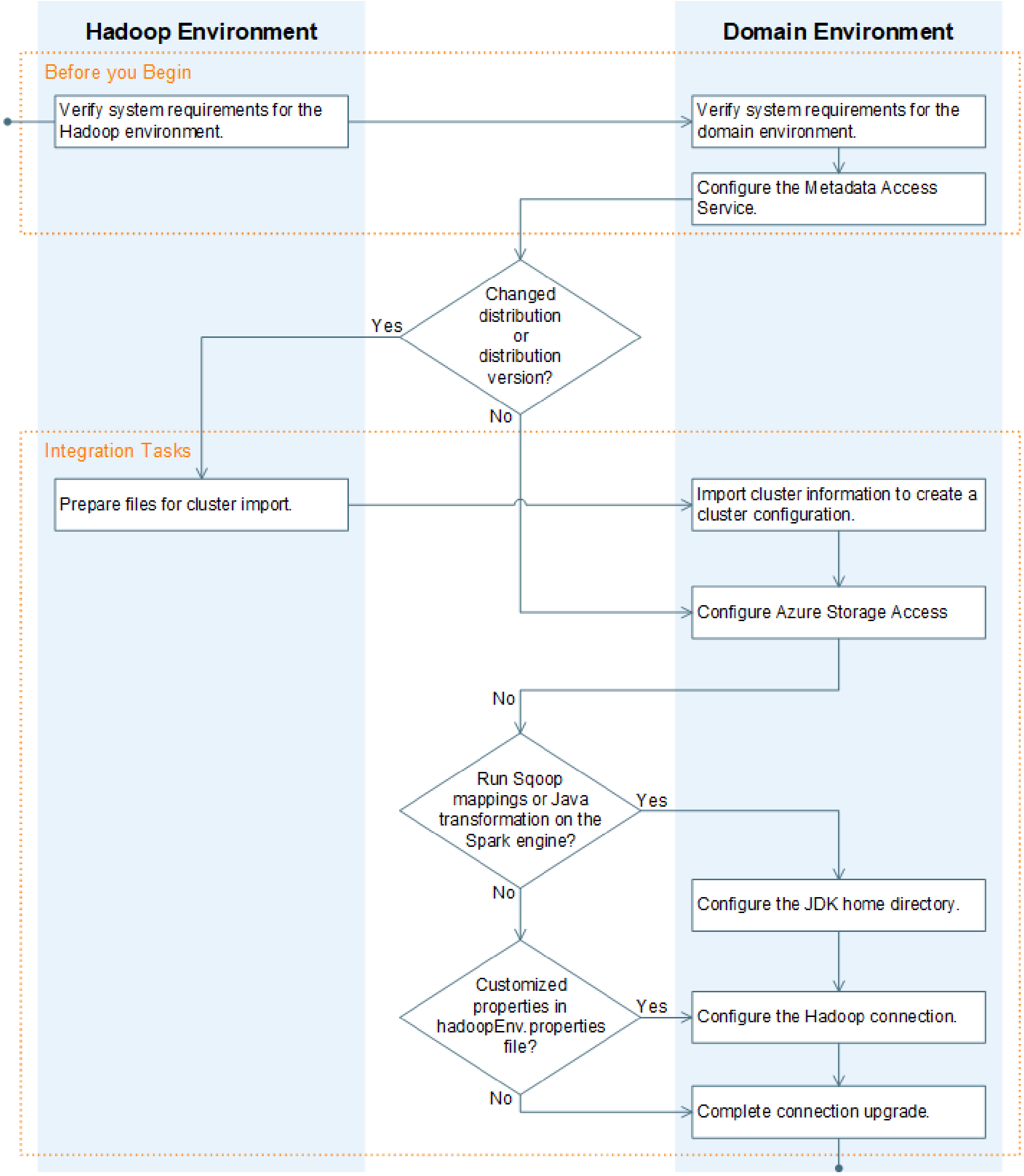
Task Flow to Integrate with Azure HDInsight

The following diagram shows the task flow to integrate the Informatica domain with Azure HDInsight:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare, directories, users, and permissions.
Configure the application services.
Use a secure cluster?
No
Yes
Configure access to secure Hadoop clusters.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure the Hadoop connection.
Update odbc.ini.
Use Sqoop to process data in relational databases?
Yes
No
Download the JDBC drivers for Sqoop connectivity.
Configure Azure Storage Access.
Edit the hosts file on the Data Integration Service machine.
Import complex file metadata on the Developer tool machine?
Yes
No
Configure the Developer tool.

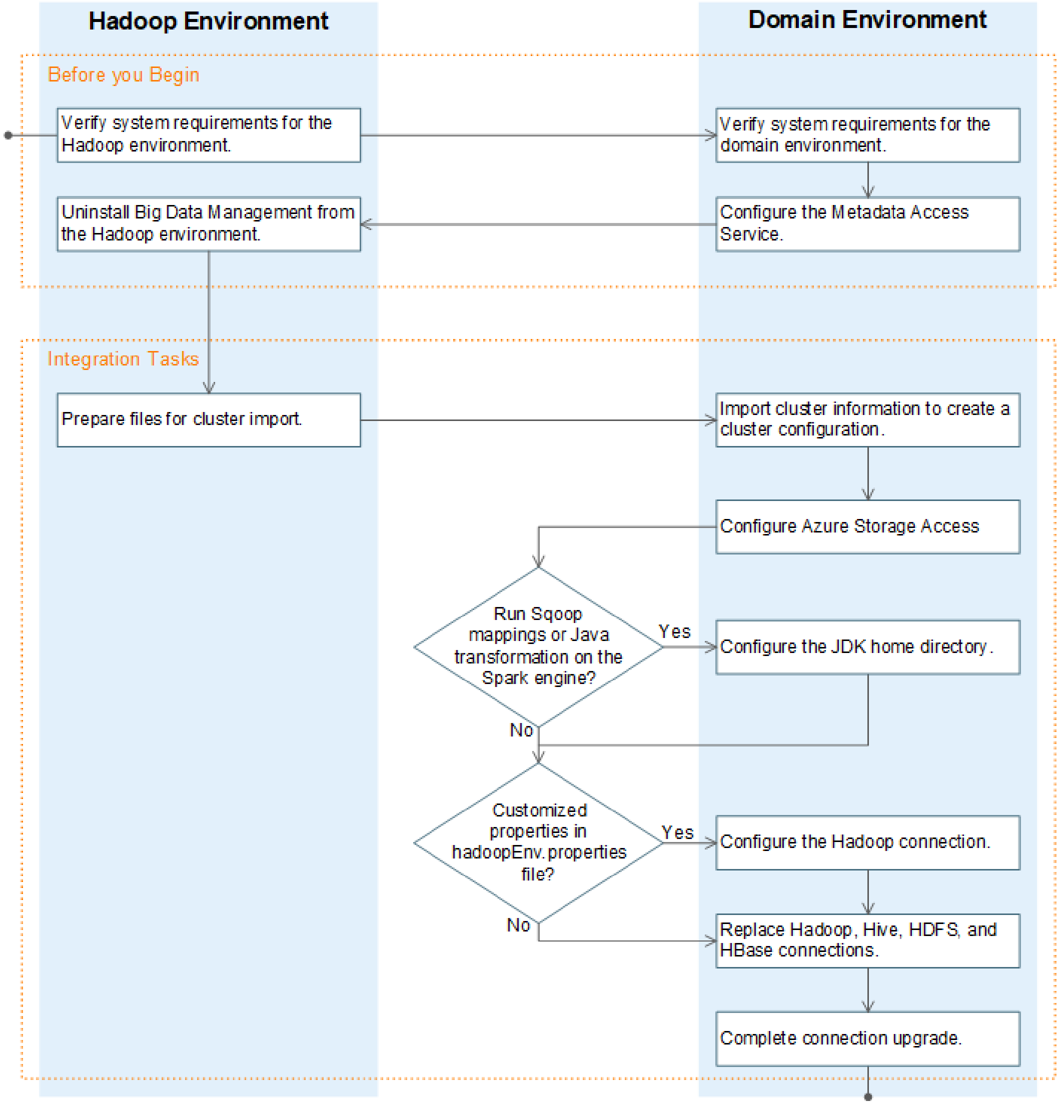
Task Flow to Upgrade from Version 10.2

The following diagram shows the task flow to upgrade Big Data Management 10.2 for Azure HDInsight:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Changed distribution or distribution version?
Yes
No
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure Azure Storage Access
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
No
Configure the JDK home directory.
Customized properties in hadoopEnv.properties file?
Yes
No
Configure the Hadoop connection.
Complete connection upgrade.

Task Flow to Upgrade from a Version Earlier than 10.2

The following diagram shows the task flow to upgrade Big Data Management from a version earlier than 10.2 for Azure HDInsight:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Uninstall Big Data Management from the Hadoop environment.
Configure the Metadata Access Service.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure Azure Storage Access
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Replace Hadoop, Hive, HDFS, and HBase connections.
Complete connection upgrade.

Prepare for Cluster Import from Azure HDInsight

Before the Informatica administrator can import cluster information to create a cluster configuration in the Informatica domain, the Hadoop administrator must perform some preliminary tasks.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded from a version earlier than 10.2. - You upgraded from 10.2 and changed the distribution or distribution version.

Complete the following tasks to prepare the cluster before the Informatica administrator creates the cluster configuration:

1. Verify that the VPN is enabled between the Informatica domain and the Azure HDInsight cloud network.
2. Verify property values in *-site.xml files that Big Data Management needs to run mappings in the Hadoop environment.
3. Provide information to the Informatica administrator that is required to import cluster information into the domain. Depending on the method of import, perform one of the following tasks:
   - To import directly from the cluster, give the Informatica administrator cluster authentication information to connect to the cluster.
   - To import from an archive file, export cluster information and provide an archive file to the Informatica administrator.

Configure *-site.xml Files for Azure HDInsight

The Hadoop administrator needs to configure *-site.xml file properties before the Informatica administrator imports cluster information into the domain.

core-site.xml

Configure the following properties in the core-site.xml file:

fs.azure.account.key.<youraccount>.blob.core.windows.net

Required for Azure HDInsight cluster that uses WASB storage. The storage account access key required to access the storage.

You can contact the HDInsight cluster administrator to get the storage account key associated with the HDInsight cluster. If you are unable to contact the administrator, perform the following steps to decrypt the encrypted storage account key:

1. Copy the value of the `fs.azure.account.key.<youraccount>.blob.core.windows.net` property.

```
<property>
<name>fs.azure.account.key.<youraccount>.blob.core.windows.net5</name>
<value>STORAGE ACCOUNT KEY</value>
</property>
```

2. Decrypt the storage account key. Run the `decrypt.sh` specified in the `fs.azure.shellkeyprovider.script` property along with the encrypted value you copied in the previous step.

```
<property>

<name>fs.azure.shellkeyprovider.script</name>
```

```
<value>/usr/lib/hdinsight-common/scripts/decrypt.sh</value>
</property>
```

3. Copy the decrypted value and update the value of `fs.azure.account.key.youraccount.blob.core.windows.net` property in the cluster configuration core-site.xml.

dfs.adls.oauth2.client.id

Required for Azure HDInsight cluster that uses ADLS storage. The application ID associated with the Service Principal required to authorize the service principal and access the storage.

To find the application ID for a service principal, in the Azure Portal, click **Azure Active Directory** > **App registrations** > **Service Principal Display Name**.

dfs.adls.oauth2.refresh.url

Required for Azure HDInsight cluster that uses ADLS storage. The OAuth 2.0 token endpoint required to authorize the service principal and access the storage.

To find the refresh URL OAuth 2.0 endpoint, in the Azure portal, click **Azure Active Directory** > **App registrations** > **Endpoints**.

dfs.adls.oauth2.credential

Required for Azure HDInsight cluster that uses ADLS storage. The password required to authorize the service principal and access the storage.

To find the password for a service principal, in the Azure portal, click **Azure Active Directory** > **App registrations** > **Service Principal Display Name** > **Settings** > **Keys**.

hadoop.proxyuser.<proxy user>.groups

Defines the groups that the proxy user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.<proxy user>.hosts

Defines the host machines that a user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

hadoop.proxyuser.yarn.groups

Comma-separated list of groups that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.yarn.hosts

Comma-separated list of hosts that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

io.compression.codecs

Enables compression on temporary staging tables.

Set to a comma-separated list of compression codec classes on the cluster.

hadoop.security.auth_to_local

Translates the principal names from the Active Directory and MIT realm into local names within the Hadoop cluster. Based on the Hadoop cluster used, you can set multiple rules.

Set to: RULE:[1:$1@$0](^.*@YOUR.REALM)s/^(.*)@YOUR.REALM\.COM$/$1/g

Set to: RULE:[2:$1@$0](^.*@YOUR.REALM\.$)s/^(.*)@YOUR.REALM\.COM$/$1/g

hbase-site.xml

Configure the following properties in the hbase-site.xml file:

hbase.use.dynamic.jars

Enables metadata import and test connection from the Developer tool. Required for an HDInsight cluster that uses ADLS storage or an Amazon EMR 5.8 cluster that uses HBase resources in S3 storage.

Set to: false

zookeeper.znode.parent

Identifies HBase master and region servers.

Set to the relative path to the znode directory of HBase.

hive-site.xml

Configure the following properties in the hive-site.xml file:

hive.cluster.delegation.token.store.class

The token store implementation. Required for HiveServer2 high availability and load balancing.

Set to: org.apache.hadoop.hive.thrift.DBTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.exec.dynamic.partition

Enables dynamic partitioned tables for Hive tables. Applicable for Hive versions 0.9 and earlier.

Set to: TRUE

hive.exec.dynamic.partition.mode

Allows all partitions to be dynamic. Required for the Update Strategy transformation in a mapping that writes to a Hive target. Also required if you use Sqoop and define a DDL query to create or replace a partitioned Hive target at run time.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.server2.support.dynamic.service.discovery

Enables HiveServer2 dynamic service discovery. Required for HiveServer2 high availability.

Set to: TRUE

hive.server2.zookeeper.namespace

The value of the ZooKeeper namespace in the JDBC connection string. Required for HiveServer2 high availability.

Set to: `jdbc:hive2://<zookeeper_ensemble>/default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2`

hive.txn.manager

Turns on transaction support. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

hive.zookeeper.quorum

Comma-separated list of ZooKeeper server host:ports in a cluster. The value of the ZooKeeper ensemble in the JDBC connection string. Required for HiveServer2 high availability.

Set to: `jdbc:hive2://<zookeeper_ensemble>/default;serviceDiscoveryMode=zooKeeper;`

mapred-site.xml

Configure the following properties in the mapred-site.xml file:

mapreduce.framework.name

The run-time framework to run MapReduce jobs. Values can be local, classic, or yarn. Required for Sqoop.

Set to: yarn

yarn.app.mapreduce.am.staging-dir

The HDFS staging directory used while submitting jobs.

Set to the staging directory path.

yarn-site.xml

Configure the following properties in the yarn-site.xml file:

yarn.application.classpath

Required for dynamic resource allocation.

"Add spark_shuffle.jar to the class path". The .jar file must contain the class "org.apache.network.yarn.YarnShuffleService."

yarn.nodemanager.resource.memory-mb

The maximum RAM available for each container. Set the maximum memory on the cluster to increase resource memory available to the Blaze engine.

Set to 16 GB if value is less than 16 GB.

yarn.nodemanager.resource.cpu-vcores

The number of virtual cores for each container. Required for Blaze engine resource allocation.

Set to 10 if the value is less than 10.

yarn.scheduler.minimum-allocation-mb

The minimum RAM available for each container. Required for Blaze engine resource allocation.

Set to 6 GB if the value is less than 6 GB.

yarn.nodemanager.vmem-check-enabled

Disables virtual memory limits for containers. Required for the Blaze and Spark engines.

Set to: FALSE

yarn.nodemanager.aux-services

Required for dynamic resource allocation for the Spark engine.

Add an entry for "spark_shuffle."

yarn.nodemanager.aux-services.spark_shuffle.class

Required for dynamic resource allocation for the Spark engine.

Set to: org.apache.spark.network.yarn.YarnShuffleService

yarn.resourcemanager.scheduler.class

Defines the YARN scheduler that the Data Integration Service uses to assign resources.

Set to: org.apache.hadoop.yarn.server.resourcemanager.scheduler

yarn.node-labels.enabled

Enables node labeling.

Set to: TRUE

yarn.node-labels.fs-store.root-dir

The HDFS location to update node label dynamically.

Set to: <hdfs://[Node name]:[Port]/[Path to store]/[Node labels]/>

tez-site.xml

Configure the following properties in the tez-site.xml file:

tez.runtime.io.sort.mb

The sort buffer memory. Required when the output needs to be sorted for Blaze and Spark engines.

Set value to 270 MB.

Prepare for Direct Import from Azure HDInsight

If you plan to provide direct access to the Informatica administrator to import cluster information, provide the required connection information.

The following table describes the information that you need to provide to the Informatica administrator to create the cluster configuration directly from the cluster:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster.

Prepare the Archive File for Import from Azure HDInsight

When you prepare the archive file for cluster configuration import from HDInsight, include all required *-site.xml files and edit the file manually after you create it.

Create a .zip or .tar file that contains the following *-site.xml files:

- core-site.xml
- hbase-site.xml. Required only to access HBase sources and targets.
- hdfs-site.xml
- hive-site.xml
- mapred-site.xml or tez-site.xml. Include the mapred-site.xml file or the tez-site.xml file based on the Hive execution type used on the Hadoop cluster.
- yarn-site.xml

After you create the archive file, edit the Hortonworks Data Platform (HDP) version string wherever it appears in the archive file. Search for the string `${hdp.version}` and replace all instances with the HDP version that HDInsight includes in the Hadoop distribution.

For example, the edited tez.task.launch.cluster-default.cmd-opts property value looks similar to the following:

```
<property>
<name>tez.task.launch.cluster-default.cmd-opts</name>
<value>-server -Djava.net.preferIPv4Stack=true -Dhdp.version=2.6.0.2-76</value>
</property>
```

Edit the hosts File for the Blaze Engine

To run the Blaze engine on every node in the cluster, verify that the /etc/hosts file on every node has entries for all other nodes.

Each node in the cluster requires an entry for the IP address and the fully qualified domain name (FQDN) of all other nodes. For example,

```
127.0.0.1 localhost node1.node.com
208.164.186.1 node1.node.com node1
208.164.186.2 node2.node.com node2
208.164.186.3 node3.node.com node3
```

Changes take effect after you restart the network.

Create a Cluster Configuration

After the Hadoop administrator prepares the cluster for import, the Informatica administrator must create a cluster configuration.

Perform this task in the following situations:

- You are integrating for the first time.
- You upgraded from a version earlier than 10.2.
- You upgraded from 10.2 and changed the distribution or distribution version.

A cluster configuration is an object in the domain that contains configuration information about the Hadoop cluster. The cluster configuration enables the Data Integration Service to push mapping logic to the Hadoop environment. Import configuration properties from the Hadoop cluster to create a cluster configuration.

The import process imports values from *-site.xml files into configuration sets based on the individual *-site.xml files. When you perform the import, the cluster configuration wizard can create Hadoop, HBase, HDFS, and Hive connection to access the Hadoop environment. If you choose to create the connections, the wizard also associates the cluster configuration with the connections.

Note: If you imported the cluster configuration when you installed Enterprise Data Lake with the Informatica domain, you can create the cluster configuration again or refresh the cluster configuration.

For more information about the cluster configuration, see the *Big Data Management Administrator Guide*.

Before You Import

Before you can import the cluster configuration, you must get information from the Hadoop administrator based on the method of import.

If you import directly from the cluster, contact the Hadoop administrator to get cluster connection information. If you import from a file, get an archive file of exported cluster information.

Importing a Cluster Configuration from the Cluster

When you import the cluster configuration directly from the cluster, you provide information to connect to the cluster.

Get cluster connection information from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following General properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. The default version is the latest version of the Hadoop distribution that Big Data Management supports. **Note:** When the cluster version differs from the default version and Informatica supports more than one version, the cluster configuration import process populates the property with the most recent supported version. For example, consider the case where Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the cluster configuration import process populates this property with 5.10, because 5.10 is the most recent supported version before 5.12. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.
Method to import the cluster configuration	Choose **Import from cluster**.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

The cluster properties appear.

4. Configure the following properties:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster.

5. Click **Next** and verify the cluster configuration information on the summary page.

Importing a Cluster Configuration from a File

You can import properties from an archive file to create a cluster configuration.

Before you import from the cluster, you must get the archive file from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. This is the latest version of the Hadoop distribution that Big Data Management supports. When the cluster version differs from the default version, the cluster configuration wizard populates the cluster configuration Hadoop distribution property with the most recent supported version relative to the cluster version. For example, suppose Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the wizard populates the version with 5.10. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.

Property	Description
Method to import the cluster configuration	Choose **Import from file** to import properties from an archive file.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

4. Click **Browse** to select a file. Select the file and click **Open**.
5. Click **Next** and verify the cluster configuration information on the summary page.

Configure the Hadoop Connection

Hadoop connections contain default values for properties such as cluster environment variables, cluster path variables, and advanced properties. Based on the cluster environment and the functionality that you use, you can add to the default values or change the default values of these properties.

For a list of Hadoop connection properties to configure, see "Configuring Hadoop Connection Properties" on page 161.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must configure the ODBCHOME and ODBCINI cluster environment variables in the Hadoop connection. You must then manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

By default, the odbc.ini file contains absolute driver paths. To run ODBC mappings on the Hive engine, you must edit the odbc.ini file and replace the absolute driver paths with relative driver paths.

You can access the odbc.ini file from the following directory on the machine that runs the Data Integration Service:

```
$INFA_HOME/ODBC7.1/
```

Replace the absolute driver paths with relative driver paths. For instance, if you use the DataDirect Greenplum Wire Protocol driver, by default, the odbc.ini file contains the following driver entries:

```
[Greenplum Wire Protocol]
Driver=/data/opt/cloudera/parcels/INFORMATICA/ODBC7.1/lib/DWgplm27.so
Description=DataDirect 7.1 Greenplum Wire Protocol
```

Update the driver entries as follows to replace the absolute driver path with a relative driver path:

```
[Greenplum Wire Protocol
Driver=./lib/DWgplm27.so
Description=DataDirect 7.1 Greenplum Wire Protocol
```

Changes take effect after you recycle the Data Integration Service.

Download the JDBC Drivers for Sqoop Connectivity

To configure Sqoop connectivity for relational databases, you must download JDBC driver .jar files.

1. Download any Type 4 JDBC driver that the database vendor recommends for Sqoop connectivity.

   Note: The DataDirect JDBC drivers that Informatica ships are not licensed for Sqoop connectivity.

2. To optimize the Sqoop mapping performance on the Spark engine while writing data to an HDFS complex file target of the Parquet format, download the following .jar files:

   - `parquet-hadoop-bundle-1.6.0.jar` from https://mvnrepository.com/artifact/com.twitter/parquet-hadoop-bundle
   - `parquet-avro-1.6.0.jar` from https://mvnrepository.com/artifact/com.twitter/parquet-avro
   - `parquet-column-1.5.0.jar` from https://mvnrepository.com/artifact/org.apache.parquet/parquet-column

3. Copy all of the .jar files to the following directory on the machine where the Data Integration Service runs:

   ```
   <Informatica installation directory>\externaljdbcjars
   ```

   At run time, the Data Integration Service copies the .jar files to the Hadoop distribution cache so that the .jar files are accessible to all nodes in the cluster.

Edit the hosts File

To ensure that Informatica can access the HDInsight cluster, edit the `/etc/hosts` file on the machine that hosts the Data Integration Service to add the following information:

- Enter the IP address, DNS name, and DNS short name for each data node on the cluster. Use `headnodehost` to identify the host as the cluster headnode host.
  For example:

  ```
  10.75.169.19 hn0-rndhdi.grg2yxlb0aouniiuvfp3bet13d.ix.internal.cloudapp.net
  headnodehost
  ```

- If the HDInsight cluster is integrated with ADLS storage, you also need to enter the IP addresses and DNS names for the hosts listed in the cluster property fs.azure.datalake.token.provider.service.urls.
  For example:

  ```
  1.2.3.67  gw1-ltsa.1320suh5npyudotcgaz0izgnhe.gx.internal.cloudapp.net
  1.2.3.68  gw0-ltsa.1320suh5npyudotcgaz0izgnhe.gx.internal.cloudapp.net
  ```

  Note: To get the IP addresses, run a telnet command from the cluster host using each host name found in the fs.azure.datalake.token.provider.service.urls property.

Configure the Developer Tool

To access the Hadoop environment from the Developer tool, the mapping developers must perform tasks on each Developer tool machine.

Configure developerCore.ini

Edit `developerCore.ini` to successfully import local complex files available on the Developer tool machine.

When you import a complex file, such as Avro or Parquet, the imported object includes metadata associated with the distribution in the Hadoop environment. If the file resides on the Developer tool machine, the import process picks up the distribution information from the `developerCore.ini` file. You must edit the `developerCore.ini` file to point to the distribution directory on the Developer tool machine.

You can find `developerCore.ini` in the following directory:

```
<Informatica installation directory>\clients\DeveloperClient
```

Add the following property:

```
-DINFA_HADOOP_DIST_DIR=hadoop\<distribution>_<version>
```

The change takes effect when you restart the Developer tool.

Complete Upgrade Tasks

If you upgraded the Informatica platform, you need to perform some additional tasks within the Informatica domain.

Based on the version that you upgraded from, perform the following tasks:

Configure the JDK home directory.

To run Sqoop mappings or process a Java transformation on the Spark engine, you must configure the JDK Home Directory in the Data Integration Service properties.

Configure the Hadoop connection.

The Hadoop connection contains additional properties. You need to manually update it to include customized configuration in the hadoopEnv.properties file from previous versions.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections. The tasks to complete connection upgrade depends on the Informatica version that you upgraded.

Configure the JDK Home Directory

To leverage Sqoop or to process a Java transformation on the Spark engine, you must install the Java Development Kit (JDK) on the machine that runs the Data Integration Service. Then, you must configure the **JDK Home Directory** property for the Data Integration Service.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

Configure the following property under the Data Integration Service execution options in Informatica Administrator:

JDK Home Directory

Required to run Sqoop mappings or mass ingestion specifications that use a Sqoop connection on the Spark engine, or to process a Java transformation on the Spark engine.

The JDK version that the Data Integration Service uses must be compatible with the JRE version on the cluster.

Set the property to the JDK installation directory on the machine that runs the Data Integration Service. Changes take effect after you recycle the Data Integration Service.

Configure the Hadoop Connection

To use properties that you customized in the hadoopEnv.properties file, you must configure the Hadoop connection properties such as cluster environment variables, cluster path variables, and advanced properties.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

When you run the Informatica upgrade, the installer backs up the existing hadoopEnv.properties file. You can find the backup hadoopEnv.properties file in the following location:

```
<Previous Informatica installation directory>/services/shared/hadoop/<Hadoop distribution
name>_<version>/infaConf
```

Edit the Hadoop connection in the Administrator tool or the Developer tool to include any properties that you manually configured in the hadoopEnv.properties file. The Hadoop connection contains default values for properties such as cluster environment and path variables and advanced properties. You can update the default values to match the properties in the hadoopEnv.properties file.

Replace the Connections with New Connections

If you created connections when you imported the cluster configuration, you need to replace connections in mappings with the new connections.

Perform this task in the following situation:
- You upgraded from a version earlier than 10.2.

The method that you use to replace connections in mappings depends on the type of connection.

Hadoop connection

Run the following commands to replace the connections:

- `infacmd dis replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that are deployed in applications.
- `infacmd mrs replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that you run from the Developer tool.

For information about the infacmd commands, see the *Informatica Command Reference*.

Hive, HDFS, and HBase connections

You must replace the connections manually.

Complete Connection Upgrade

If *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

Perform the following tasks to update the connections:

Update changed properties

Review connections that you created in a previous release to update the values for connection properties. For example, if you added nodes to the cluster or if you updated the distribution version, you might need to verify host names, URIs, or port numbers for some of the properties.

Associate the cluster configuration

The Hadoop, Hive, HDFS, and HBase connections must be associated with a cluster configuration. Complete the following tasks:

1. Run `infacmd isp listConnections` to identify the connections that you need to upgrade. Use `-ct` to list connections of a particular type.
2. Run `infacmd isp UpdateConnection` to associate the cluster configuration with the connection. Use `-cn` to name the connection and `-o clusterConfigID` to associate the cluster configuration with the connection.

For more information about infacmd, see the *Informatica Command Reference*.

CHAPTER 5

Cloudera CDH Integration Tasks

This chapter includes the following topics:

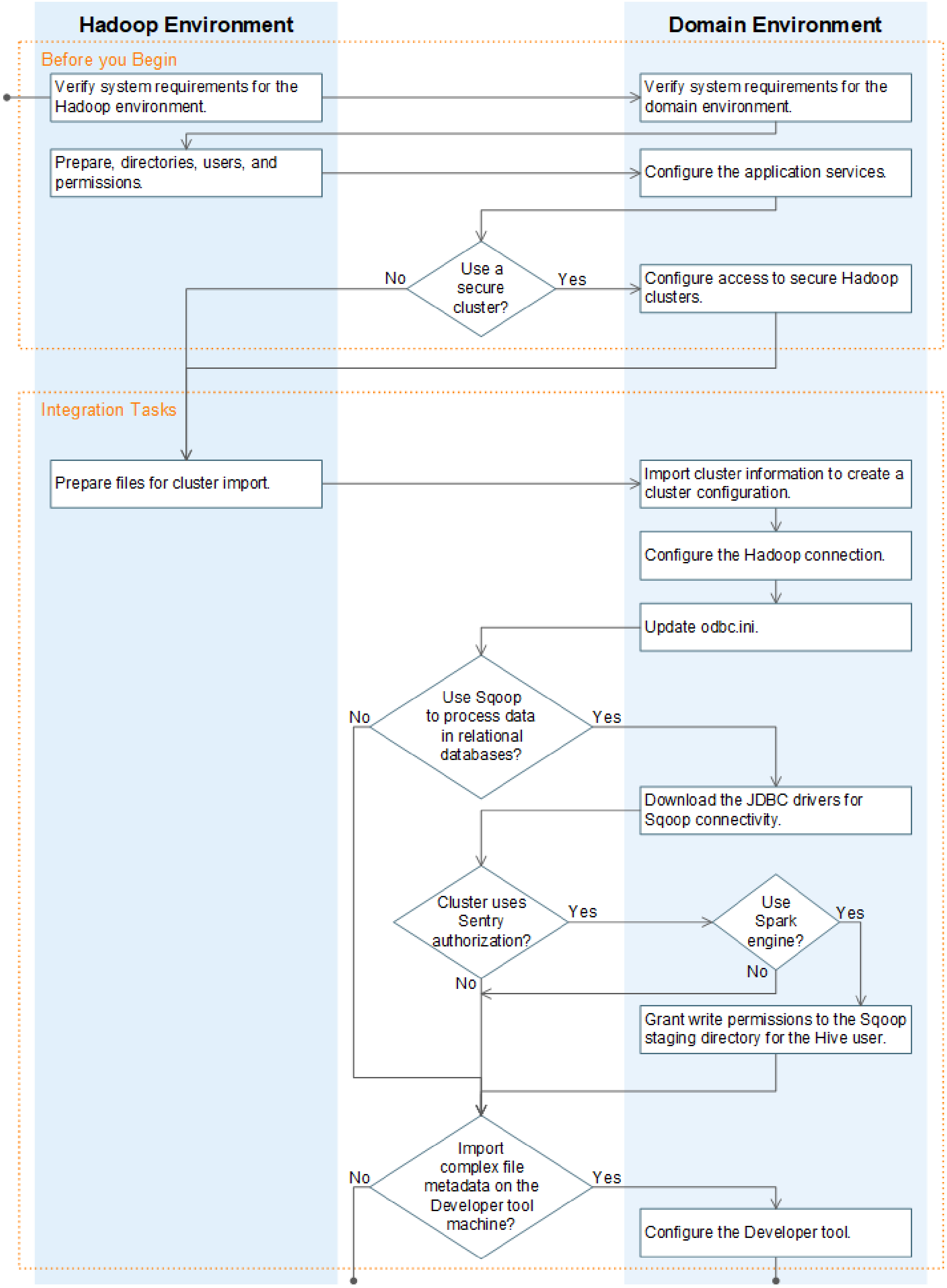
Cloudera CDH Task Flows

Depending on whether you want to integrate or upgrade Big Data Management in a Cloudera CDH environment, you can use the flow charts to perform the following tasks:

- Integrate the Informatica domain with Cloudera CDH for the first time.
- Upgrade from version 10.2.1.
- Upgrade from version 10.2.
- Upgrade from a version earlier than 10.2.

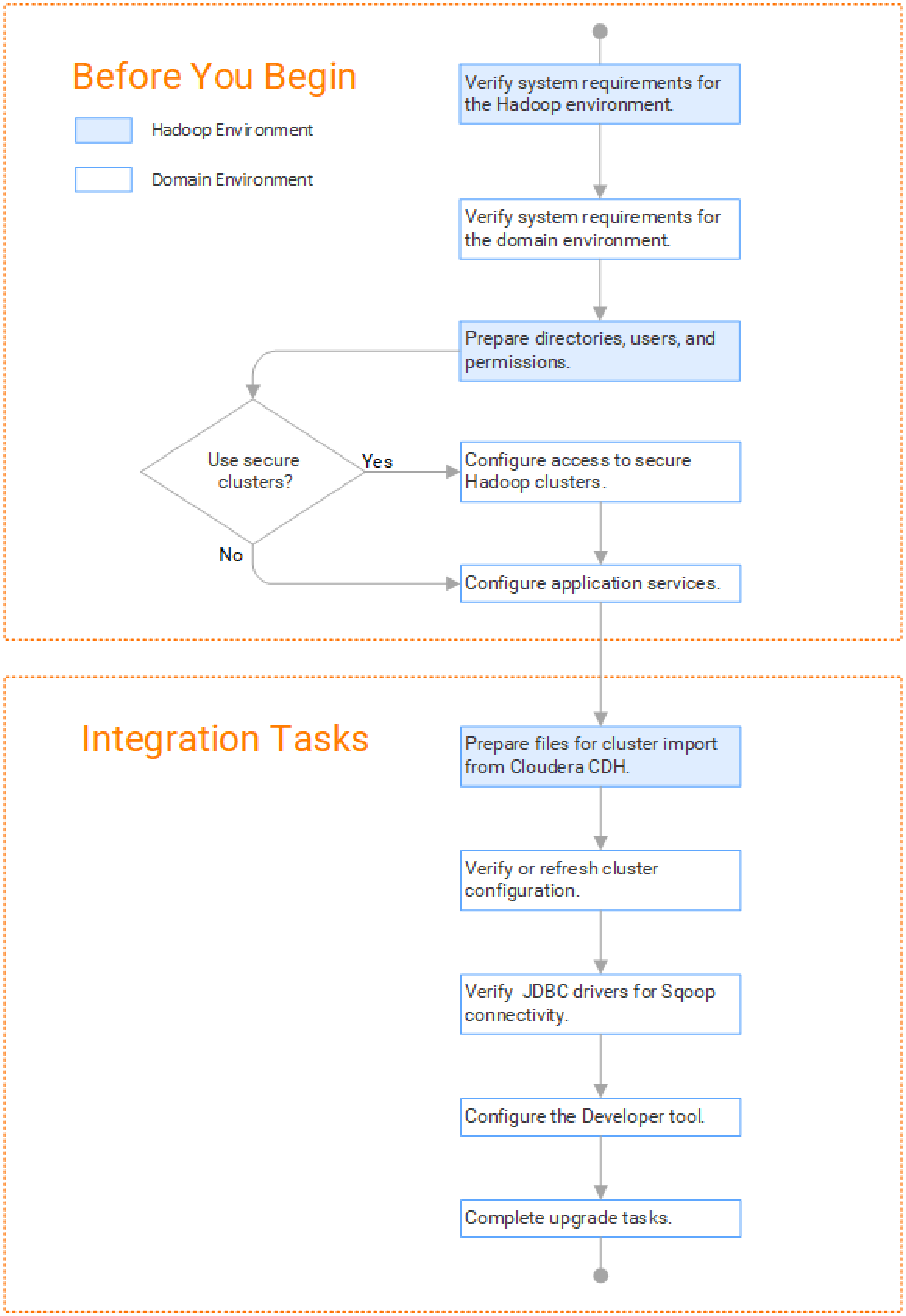
Task Flow to Integrate with Cloudera CDH

The following diagram shows the task flow to integrate the Informatica domain with Cloudera CDH:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare, directories, users, and permissions.
Configure the application services.
Use a secure cluster?
No
Yes
Configure access to secure Hadoop clusters.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure the Hadoop connection.
Update odbc.ini.
Use Sqoop to process data in relational databases?
No
Yes
Download the JDBC drivers for Sqoop connectivity.
Cluster uses Sentry authorization?
Yes
No
Use Spark engine?
Yes
No
Grant write permissions to the Sqoop staging directory for the Hive user.
Import complex file metadata on the Developer tool machine?
No
Yes
Configure the Developer tool.

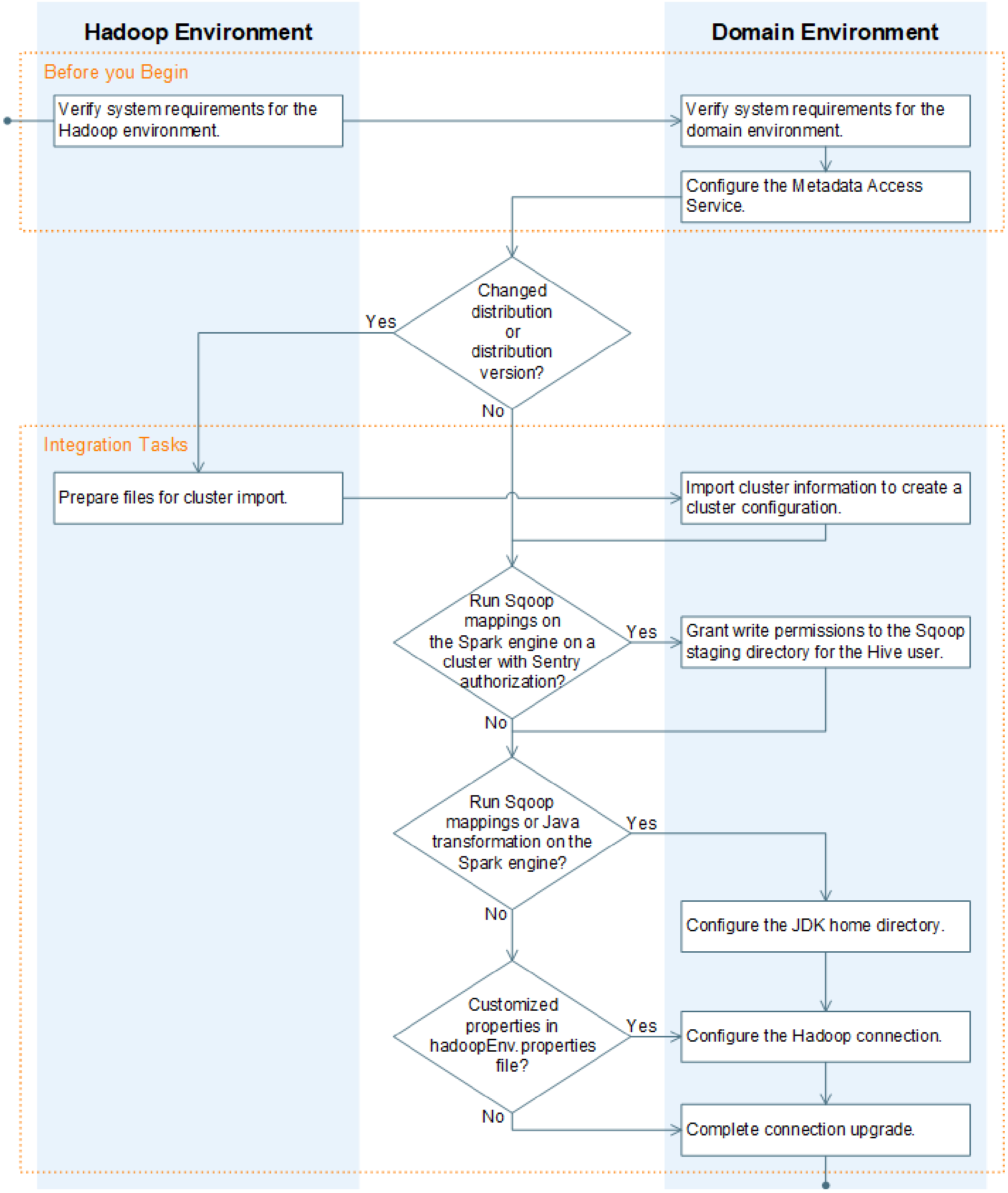
Task Flow to Upgrade from Version 10.2.1

The following diagram shows the task flow to upgrade Big Data Management 10.2.1 for Cloudera CDH:


Before You Begin
Hadoop Environment
Domain Environment
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare directories, users, and permissions.
Use secure clusters?
Yes
No
Configure access to secure Hadoop clusters.
Configure application services.
Integration Tasks
Prepare files for cluster import from Cloudera CDH.
Verify or refresh cluster configuration.
Verify JDBC drivers for Sqoop connectivity.
Configure the Developer tool.
Complete upgrade tasks.

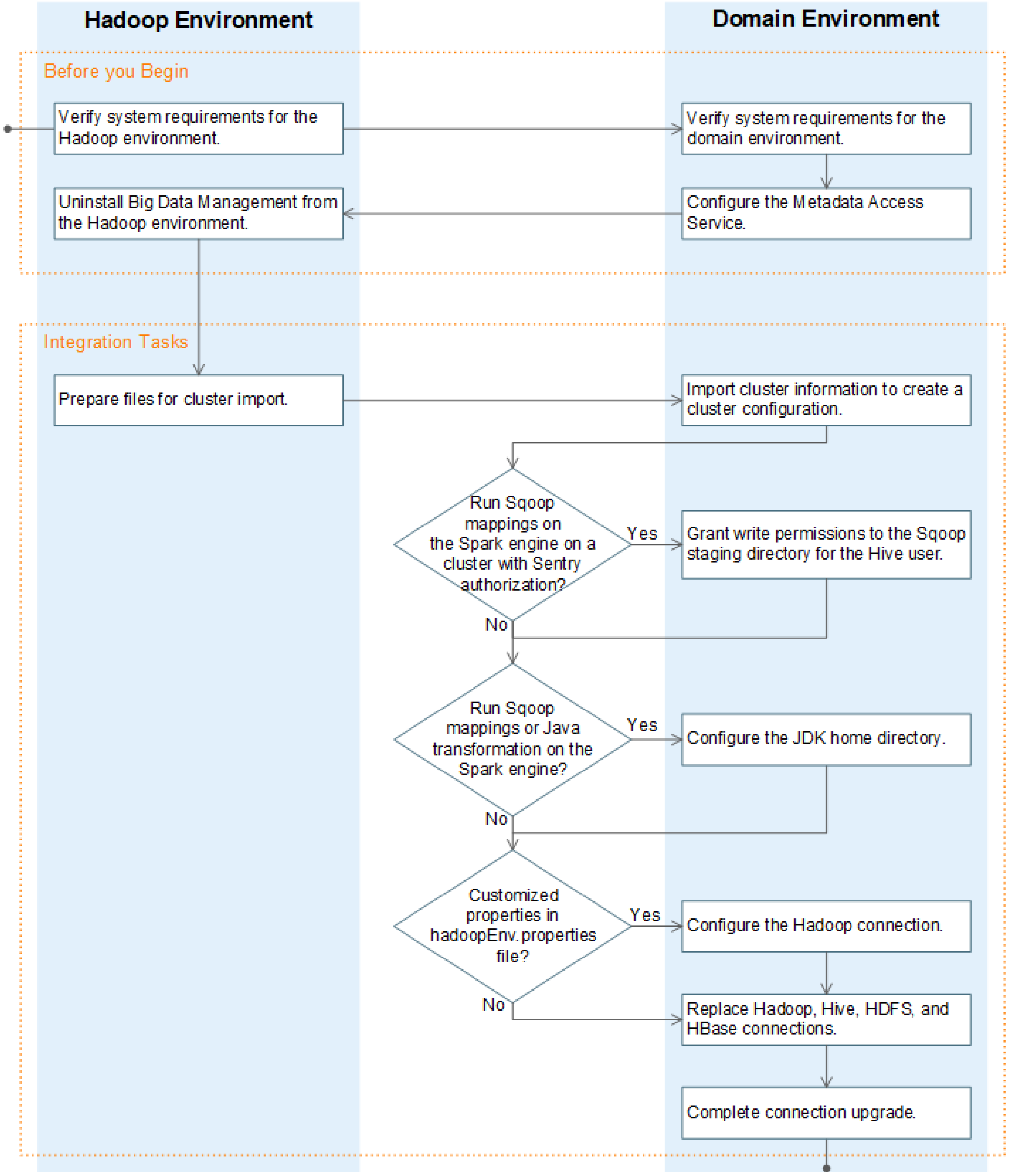
Task Flow to Upgrade from Version 10.2

The following diagram shows the task flow to upgrade Big Data Management 10.2 for Cloudera CDH:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Changed distribution or distribution version?
Yes
No
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Run Sqoop mappings on the Spark engine on a cluster with Sentry authorization?
Yes
Grant write permissions to the Sqoop staging directory for the Hive user.
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Complete connection upgrade.

Task Flow to Upgrade from a Version Earlier than 10.2

The following diagram shows the task flow to upgrade Big Data Management from a version earlier than 10.2 for Cloudera CDH:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Uninstall Big Data Management from the Hadoop environment.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Run Sqoop mappings on the Spark engine on a cluster with Sentry authorization?
Yes
Grant write permissions to the Sqoop staging directory for the Hive user.
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Replace Hadoop, Hive, HDFS, and HBase connections.
Complete connection upgrade.

Prepare for Cluster Import from Cloudera CDH

Before the Informatica administrator can import cluster information to create a cluster configuration in the Informatica domain, the Hadoop administrator must perform some preliminary tasks.

Perform this task in the following situations:

- You are integrating for the first time.
- You upgraded Big Data Management.

Note: If you are upgrading from a previous version, verify the properties and suggested values, as Big Data Management might require additional properties or different values for existing properties.

Complete the following tasks to prepare the cluster before the Informatica administrator creates the cluster configuration:

1. Verify property values in *-site.xml files that Big Data Management needs to run mappings in the Hadoop environment.
2. Provide information to the Informatica administrator that is required to import cluster information into the domain. Depending on the method of import, perform one of the following tasks:
    - To import directly from the cluster, give the Informatica administrator cluster authentication information to connect to the cluster.
    - To import from an archive file, export cluster information and provide an archive file to the Big Data Management administrator.

Configure *-site.xml Files for Cloudera CDH

The Hadoop administrator needs to configure *-site.xml file properties and restart impacted services before the Informatica administrator imports cluster information into the domain.

core-site.xml

Configure the following properties in the core-site.xml file:

fs.s3.enableServerSideEncryption

Enables server side encryption for hive buckets. Required if the S3 bucket is encrypted. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to: TRUE

fs.s3a.access.key

The ID for the Blaze and Spark engines to connect to the Amazon S3 file system.

Set to your access key.

fs.s3a.secret.key

The password for the Blaze and Spark engines to connect to the Amazon S3 file system

Set to your access ID.

fs.s3a.server-side-encryption-algorithm

The server-side encryption algorithm for S3. Required if the S3 bucket is encrypted using an algorithm. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to the encryption algorithm used.

hadoop.proxyuser.<proxy user>.groups

Defines the groups that the proxy user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.<proxy user>.hosts

Defines the host machines that a user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

io.compression.codecs

Enables compression on temporary staging tables.

Set to a comma-separated list of compression codec classes on the cluster.

hadoop.security.auth_to_local

Translates the principal names from the Active Directory and MIT realm into local names within the Hadoop cluster. Based on the Hadoop cluster used, you can set multiple rules.

Set to: RULE:[1:$1@$0](^.*@YOUR.REALM)s/^(.*)@YOUR.REALM\.COM$/$1/g

Set to: RULE:[2:$1@$0](^.*@YOUR.REALM\.$)s/^(.*)@YOUR.REALM\.COM$/$1/g

hbase-site.xml

Configure the following properties in the hbase-site.xml file:

zookeeper.znode.parent

Identifies HBase master and region servers.

Set to the relative path to the znode directory of HBase.

hdfs-site.xml

Configure the following properties in the hdfs-site.xml file:

dfs.encryption.key.provider.uri

The KeyProvider used to interact with encryption keys when reading and writing to an encryption zone. Required if sources or targets reside in the HDFS encrypted zone on Java KeyStore KMS-enabled Cloudera CDH cluster or a Ranger KMS-enabled Hortonworks HDP cluster.

Set to: kmf://http@xx11.xyz.com:16000/kms

hive-site.xml

Configure the following properties in the hive-site.xml file:

hive.cluster.delegation.token.store.class

Applies only to Cloudera CDH cluster if HiveServer2 uses Apache Zookeeper for high availability and load balancing. The token store implementation.

Set to: org.apache.hadoop.hive.thrift.ZooKeeperTokenStore

hive.cluster.delegation.token.store.class

The token store implementation. Required for HiveServer2 high availability and load balancing.

Set to: org.apache.hadoop.hive.thrift.DBTokenStore

hive.exec.dynamic.partition

Enables dynamic partitioned tables for Hive tables. Applicable for Hive versions 0.9 and earlier.

Set to: TRUE

hive.exec.dynamic.partition.mode

Allows all partitions to be dynamic. Required if you use Sqoop and define a DDL query to create or replace a partitioned Hive target at run time.

Set to: nonstrict

hiveserver2_load_balancer

Enables high availability for multiple HiveServer2 hosts.

Set to: `jdbc:hive2://<HiveServer2 Load Balancer>:<HiveServer2 Port>/default;principal=hive/<HiveServer2 load Balancer>@<REALM>`

mapred-site.xml

Configure the following properties in the mapred-site.xml file:

mapreduce.application.classpath

A comma-separated list of CLASSPATH entries for MapReduce applications. Required for Sqoop.

Include the entries: $HADOOP_MAPRED_HOME/*,$HADOOP_MAPRED_HOME/lib/*,$MR2_CLASSPATH, $CDH_MR2_HOME

mapreduce.framework.name

The run-time framework to run MapReduce jobs. Values can be local, classic, or yarn. Required for Sqoop.

Set to: yarn

mapreduce.jobhistory.address

Location of the MapReduce JobHistory Server. The default port is 10020. Required for Sqoop.

Set to: <MapReduce JobHistory Server>:<port>

mapreduce.jobhistory.intermediate-done-dir

Directory where MapReduce jobs write history files. Required for Sqoop.

Set to: /mr-history/tmp

mapreduce.jobhistory.done-dir

Directory where the MapReduce JobHistory Server manages history files. Required for Sqoop.

Set to: /mr-history/done

mapreduce.jobhistory.principal

The Service Principal Name for the MapReduce JobHistory Server. Required for Sqoop.

Set to: mapred/_HOST@YOUR-REALM

mapreduce.jobhistory.webapp.address

Web address of the MapReduce JobHistory Server. The default value is 19888. Required for Sqoop.

Set to: <host>:<port>

yarn.app.mapreduce.am.staging-dir

The HDFS staging directory used while submitting jobs.

Set to the staging directory path.

yarn-site.xml

Configure the following properties in the yarn-site.xml file:

yarn.application.classpath

Required for dynamic resource allocation.

"Add spark_shuffle.jar to the class path". The .jar file must contain the class "org.apache.network.yarn.YarnShuffleService."

yarn.nodemanager.resource.memory-mb

The maximum RAM available for each container. Set the maximum memory on the cluster to increase resource memory available to the Blaze engine.

Set to 16 GB if value is less than 16 GB.

yarn.nodemanager.resource.cpu-vcores

The number of virtual cores for each container. Required for Blaze engine resource allocation.

Set to 10 if the value is less than 10.

yarn.scheduler.minimum-allocation-mb

The minimum RAM available for each container. Required for Blaze engine resource allocation.

Set to 6 GB if the value is less than 6 GB.

yarn.nodemanager.vmem-check-enabled

Disables virtual memory limits for containers. Required for the Blaze and Spark engines.

Set to: FALSE

yarn.nodemanager.aux-services

Required for dynamic resource allocation for the Spark engine.

Add an entry for "spark_shuffle."

yarn.nodemanager.aux-services.spark_shuffle.class

Required for dynamic resource allocation for the Spark engine.

Set to: org.apache.spark.network.yarn.YarnShuffleService

yarn.resourcemanager.scheduler.class

Defines the YARN scheduler that the Data Integration Service uses to assign resources.

Set to: org.apache.hadoop.yarn.server.resourcemanager.scheduler

yarn.node-labels.enabled

Enables node labeling.

Set to: TRUE

yarn.node-labels.fs-store.root-dir

The HDFS location to update node label dynamically.

Set to: <hdfs://[Node name]:[Port]/[Path to store]/[Node labels]/>

Prepare for Direct Import from Cloudera CDH

If you plan to provide direct access to the Informatica administrator to import cluster information, provide the required connection information.

The following table describes the information that you need to provide to the Informatica administrator to create the cluster configuration directly from the cluster:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster. To find the correct Cloudera cluster name when you have multiple clusters, perform the following steps: 1. Log in to Cloudera Manager adding the following string to the URL: /api/v8/clusters 2. Provide the Informatica Administrator the cluster property name that appears in the browser tab.

Prepare the Archive File for Import from Cloudera CDH

If you plan to provide an archive file for the Informatica administrator, ensure that you include all required site-*.xml files.

Create a .zip or .tar file that contains the following *-site.xml files:

- core-site.xml
- hbase-site.xml. Required only for access to HBase sources and targets.
- hdfs-site.xml
- hive-site.xml
- mapred-site.xml
- yarn-site.xml

Give the Informatica administrator access to the archive file to import the cluster information into the domain.

Create a Cluster Configuration

After the Hadoop administrator prepares the cluster for import, the Informatica administrator must create a cluster configuration.

Perform this task in the following situations:

- You are integrating for the first time.
- You upgraded from a version earlier than 10.2.
- You upgraded from 10.2 and changed the distribution or distribution version.

A cluster configuration is an object in the domain that contains configuration information about the Hadoop cluster. The cluster configuration enables the Data Integration Service to push mapping logic to the Hadoop environment. Import configuration properties from the Hadoop cluster to create a cluster configuration.

The import process imports values from *-site.xml files into configuration sets based on the individual *-site.xml files. When you perform the import, the cluster configuration wizard can create Hadoop, HBase, HDFS, and Hive connection to access the Hadoop environment. If you choose to create the connections, the wizard also associates the cluster configuration with the connections.

Note: If you imported the cluster configuration when you installed Enterprise Data Lake with the Informatica domain, you can create the cluster configuration again or refresh the cluster configuration.

For more information about the cluster configuration, see the *Big Data Management Administrator Guide*.

Before You Import

Before you can import the cluster configuration, you must get information from the Hadoop administrator based on the method of import.

If you import directly from the cluster, contact the Hadoop administrator to get cluster connection information. If you import from a file, get an archive file of exported cluster information.

Importing a Cluster Configuration from the Cluster

When you import the cluster configuration directly from the cluster, you provide information to connect to the cluster.

Get cluster connection information from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.

3. Configure the following General properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. The default version is the latest version of the Hadoop distribution that Big Data Management supports. **Note:** When the cluster version differs from the default version and Informatica supports more than one version, the cluster configuration import process populates the property with the most recent supported version. For example, consider the case where Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the cluster configuration import process populates this property with 5.10, because 5.10 is the most recent supported version before 5.12. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.
Method to import the cluster configuration	Choose **Import from cluster**.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

The cluster properties appear.

4. Configure the following properties:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.

Property	Description
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster.

5. Click **Next** and verify the cluster configuration information on the summary page.

Importing a Cluster Configuration from a File

You can import properties from an archive file to create a cluster configuration.

Before you import from the cluster, you must get the archive file from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. This is the latest version of the Hadoop distribution that Big Data Management supports. When the cluster version differs from the default version, the cluster configuration wizard populates the cluster configuration Hadoop distribution property with the most recent supported version relative to the cluster version. For example, suppose Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the wizard populates the version with 5.10. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.

Property	Description
Method to import the cluster configuration	Choose **Import from file** to import properties from an archive file.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

4. Click **Browse** to select a file. Select the file and click **Open**.
5. Click **Next** and verify the cluster configuration information on the summary page.

Verify or Refresh the Cluster Configuration

You might need to refresh the cluster configuration or update the distribution version in the cluster configuration when you upgrade.

Verify the Cluster Configuration

The cluster configuration contains a property for the distribution version. The verification task depends on the version you upgraded:

Upgrade from 10.2

If you upgraded from 10.2 and you changed the distribution version, you need to verify the distribution version in the General properties of the cluster configuration.

Upgrade from 10.2.1

Effective in version 10.2.1, Informatica assigns a default version to each Hadoop distribution type. If you configure the cluster configuration to use the default version, the upgrade process upgrades to the assigned default version if the version changes. If you have not upgraded your Hadoop distribution to Informatica's default version, you need to update the distribution version property.

For example, suppose the assigned default Hadoop distribution version for 10.2.1 is *n*, and for 10.2.2 is *n +1*. If the cluster configuration uses the default supported Hadoop version of *n*, the upgraded cluster configuration uses the default version of *n+1*. If you have not upgraded the distribution in the Hadoop environment you need to change the cluster configuration to use version *n*.

If you configure the cluster configuration to use a distribution version that is not the default version, you need to update the distribution version property in the following circumstances:

- Informatica dropped support for the distribution version.
- You changed the distribution version.

Refresh the Cluster Configuration

If you updated any of the *-site.xml files noted in the topic to prepare for cluster import, you need to refresh the cluster configuration in the Administrator tool.

Verify JDBC Drivers for Sqoop Connectivity

Verify that you have the JDBC drivers to access JDBC-compliant databases in the Hadoop environment. You might need separate drivers for metadata import and for run-time processing.

You download drivers based on design-time and run-time requirements:

- **Design-time.** To import metadata, you can use the DataDirect drivers packaged with the Informatica installer if they are available. If they are not available, use any Type 4 JDBC driver that the database vendor recommends.
- **Run-time.** To run mappings, use any Type 4 JDBC driver that the database vendor recommends. Some distributions support other drivers to use Sqoop connectors. You cannot use the DataDirect drivers for run-time processing.

Verify Design-time Drivers

Use the DataDirect JDBC drivers packaged with the Informatica installer to import metadata from JDBC-compliant databases. If the DataDirect JDBC drivers are not available for a specific JDBC-compliant database, download the Type 4 JDBC driver associated with that database.

Copy the JDBC driver .jar files to the following location on the Developer tool machine:

```
<Informatica installation directory>\clients\externaljdbcjars
```

Verify Run-time Drivers

Verify run-time drivers for mappings that access JDBC-compliant databases in the Hadoop environment. Use any Type 4 JDBC driver that the database vendor recommends.

1. Download Type 4 JDBC drivers associated with the JCBC-compliant databases that you want to access.
2. To use Sqoop TDCH Cloudera Connector Powered by Teradata, perform the following tasks:
   - Download all .jar files in the Cloudera Connector Powered by Teradata package from the following location: http://www.cloudera.com/downloads.html. The package has the following naming convention: `sqoop-connector-teradata-<version>.tar`
   - Download `terajdbc4.jar` and `tdgssconfig.jar` from the following location: http://downloads.teradata.com/download/connectivity/jdbc-driver
3. To optimize the Sqoop mapping performance on the Spark engine while writing data to an HDFS complex file target of the Parquet format, download the following .jar files:
   - `parquet-hadoop-bundle-1.6.0.jar` from http://central.maven.org/maven2/com/twitter/parquet-avro/1.6.0/
   - `parquet-avro-1.6.0.jar` from http://central.maven.org/maven2/com/twitter/parquet-hadoop-bundle/1.6.0/

- `parquet-column-1.5.0.jar` from http://central.maven.org/maven2/com/twitter/parquet-column/1.5.0/

4. Copy all of the .jar files to the following directory on the machine where the Data Integration Service runs:

   ```
   <Informatica installation directory>\externaljdbcjars
   ```

   Changes take effect after you recycle the Data Integration Service. At run time, the Data Integration Service copies the .jar files to the Hadoop distribution cache so that the .jar files are accessible to all nodes in the cluster.

Import Security Certificates to Clients

When you use custom, special, or self-signed security certificates to secure the Hadoop cluster, Informatica clients that connect to the cluster require these certificates to be present in the client machine truststore.

To connect to the Hadoop cluster to develop a mapping, the Developer tool requires security certificate aliases on the machine that hosts the Developer tool. To run a mapping, the machine that hosts the Data Integration Service requires these same certificate alias files.

Perform the following steps from the Developer tool host machine, and then repeat them from the Data Integration Service host machine:

1. Run the following command to export the certificates from the cluster:

   ```
   keytool -export -alias <alias name> -keystore <custom.truststore file location> -
   file <exported certificate file location> -storepass <password>
   ```

   For example,

   ```
   keytool -export -alias <alias name> -keystore ~/custom.truststore -file ~/
   exported.cer
   ```

   The command produces a certificate file.

2. Choose to import security certificates to an SSL-enabled domain or a domain that is not SSL-enabled using the following command:

   ```
   keytool -import -trustcacerts -alias <alias name> -file <exported certificate file
   location> -keystore <java cacerts location> -storepass <password>
   ```

   For example,

   ```
   keytool -import -alias <alias name> -file ~/exported.cer -keystore <Informatica
   installation directory>/java/jre/lib/security/cacerts
   ```

Configure the Developer Tool

To access the Hadoop environment from the Developer tool, the mapping developers must perform tasks on each Developer tool machine.

Configure developerCore.ini

Edit `developerCore.ini` to successfully import local complex files available on the Developer tool machine.

When you import a complex file, such as Avro or Parquet, the imported object includes metadata associated with the distribution in the Hadoop environment. If the file resides on the Developer tool machine, the import process picks up the distribution information from the `developerCore.ini` file. You must edit the `developerCore.ini` file to point to the distribution directory on the Developer tool machine.

You can find `developerCore.ini` in the following directory:

```
<Informatica installation directory>\clients\DeveloperClient
```

Add the following property:

```
-DINFA_HADOOP_DIST_DIR=hadoop\<distribution>_<version>
```

The change takes effect when you restart the Developer tool.

Complete Upgrade Tasks

If you upgraded the Informatica platform, you need to perform some additional tasks within the Informatica domain.

Based on the version that you upgraded from, perform the following tasks:

Configure the JDK home directory.

To run Sqoop mappings or process a Java transformation on the Spark engine, you must configure the JDK Home Directory in the Data Integration Service properties.

Configure the Hadoop connection.

The Hadoop connection contains additional properties. You need to manually update it to include customized configuration in the hadoopEnv.properties file from previous versions.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections. The tasks to complete connection upgrade depends on the Informatica version that you upgraded.

Update Connections

You might need to update connections based on the version you are upgrading from.

Consider the following types of updates that you might need to make:

Configure the Hadoop connection.

Configure the Hadoop connection to incorporate properties from the hadoopEnv.properties file.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections.

Configure the Hadoop Connection

To use properties that you customized in the hadoopEnv.properties file, you must configure the Hadoop connection properties such as cluster environment variables, cluster path variables, and advanced properties.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

When you run the Informatica upgrade, the installer backs up the existing hadoopEnv.properties file. You can find the backup hadoopEnv.properties file in the following location:

```
<Previous Informatica installation directory>/services/shared/hadoop/<Hadoop distribution
name>_<version>/infaConf
```

Edit the Hadoop connection in the Administrator tool or the Developer tool to include any properties that you manually configured in the hadoopEnv.properties file. The Hadoop connection contains default values for properties such as cluster environment and path variables and advanced properties. You can update the default values to match the properties in the hadoopEnv.properties file.

Replace the Connections with New Connections

If you created connections when you imported the cluster configuration, you need to replace connections in mappings with the new connections.

Perform this task in the following situation:
- You upgraded from a version earlier than 10.2.

The method that you use to replace connections in mappings depends on the type of connection.

Hadoop connection

Run the following commands to replace the connections:

- `infacmd dis replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that are deployed in applications.
- `infacmd mrs replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that you run from the Developer tool.

For information about the infacmd commands, see the *Informatica Command Reference*.

Hive, HDFS, and HBase connections

You must replace the connections manually.

Complete Connection Upgrade

If *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

Perform the following tasks to update the connections:

Update changed properties

Review connections that you created in a previous release to update the values for connection properties. For example, if you added nodes to the cluster or if you updated the distribution version, you might need to verify host names, URIs, or port numbers for some of the properties.

Associate the cluster configuration

The Hadoop, Hive, HDFS, and HBase connections must be associated with a cluster configuration. Complete the following tasks:

1. Run `infacmd isp listConnections` to identify the connections that you need to upgrade. Use `-ct` to list connections of a particular type.
2. Run `infacmd isp UpdateConnection` to associate the cluster configuration with the connection. Use `-cn` to name the connection and `-o clusterConfigID` to associate the cluster configuration with the connection.

For more information about infacmd, see the *Informatica Command Reference*.

Update Streaming Objects

Big Data Streaming uses Spark Structured Streaming to process data instead of Spark Streaming. To support Spark Structured Streaming, some header ports are added to the data objects, and support to some of the data objects and transformations are deferred to a future release. The behavior of some of the data objects is also updated.

After you upgrade, the existing streaming mappings become invalid because of the unavailable header ports, the unsupported transformations or data objects, and the behavior change of some data objects.

To use an existing mapping, perform the following tasks:

- Re-create the physical data objects. After you re-create the physical data objects, the data objects get the required header ports, such as timestamp, partitionID, or key based on the data object.
- Re-create the Normalizer transformation. After you re-create the Normalizer transformation, you can change or specify the type configuration for the complex port.
- Update the streaming mapping. If the mapping contains Kafka target, Aggregator transformation, Joiner transformation, or Normalizer transformation replace the data object or transformation, and then update the mapping because of the changed behavior of these transformations and data objects.
- Verify the deferred data object types. If the streaming mapping contains unsupported transformations or data objects, contact Informatica Global Customer Support.

Re-create the Physical Data Objects

When you re-create the physical data objects, the physical data objects get the header ports and some properties are not available for some data objects. Update the existing mapping with the newly created physical data objects.

1. Go to the existing mapping, select the data object from the mapping.
2. Click the **Properties** tab. On the **Column Projection** tab, click **Edit Schema**.
3. Note the schema information from the **Edit Schema** dialog box.
4. Note the parameters information from the **Parameters** tab.
5. Create new physical data objects.

After you re-create the data objects, the physical data objects get the required header ports. The Microsoft Azure does not support the following properties and are not available for Azure Event Hubs data objects:

- Consumer Properties
- Partition Count

Update the Streaming Mappings

After you re-create the data object, replace the existing data objects with the re-created data objects. If the mapping contains Normaliser Transformation, Aggregator transformation, or Joiner transformation, update the mapping because of the changed behavior of these transformations and data objects.

Aggregator and Joiner Transformation Updates

An Aggregator transformation must be downstream from a Joiner transformation. A Window transformation must be directly upstream from both Aggregator and Joiner transformations. Previously, you could use an Aggregator transformation anywhere in the streaming mapping.

If a mapping contains an Aggregator transformation upstream from a Joiner transformation, move the Aggregator transformation downstream from a Joiner transformation. Add a Window transformation directly upstream from both Aggregator and Joiner transformations.

Transformation Updates

If a streaming mapping contains a transformation that uses a complex data types, you must manually set the property to an appropriate type configuration.

Verify the Deferred Data Object Types

After you upgrade, the streaming mappings might contain some transformations and data objects that are deferred.

The following table lists the data object types to which the support is deferred to a future release:

Object Type	Object
Source	JMS MapR Streams
Target	MapR Streams
Transformation	Data Masking Joiner Rank Sorter

If you want to continue using the mappings that contain deferred data objects or transformations, you must contact Informatica Global Customer Support.

CHAPTER 6

Hortonworks HDP Integration Tasks

This chapter includes the following topics:

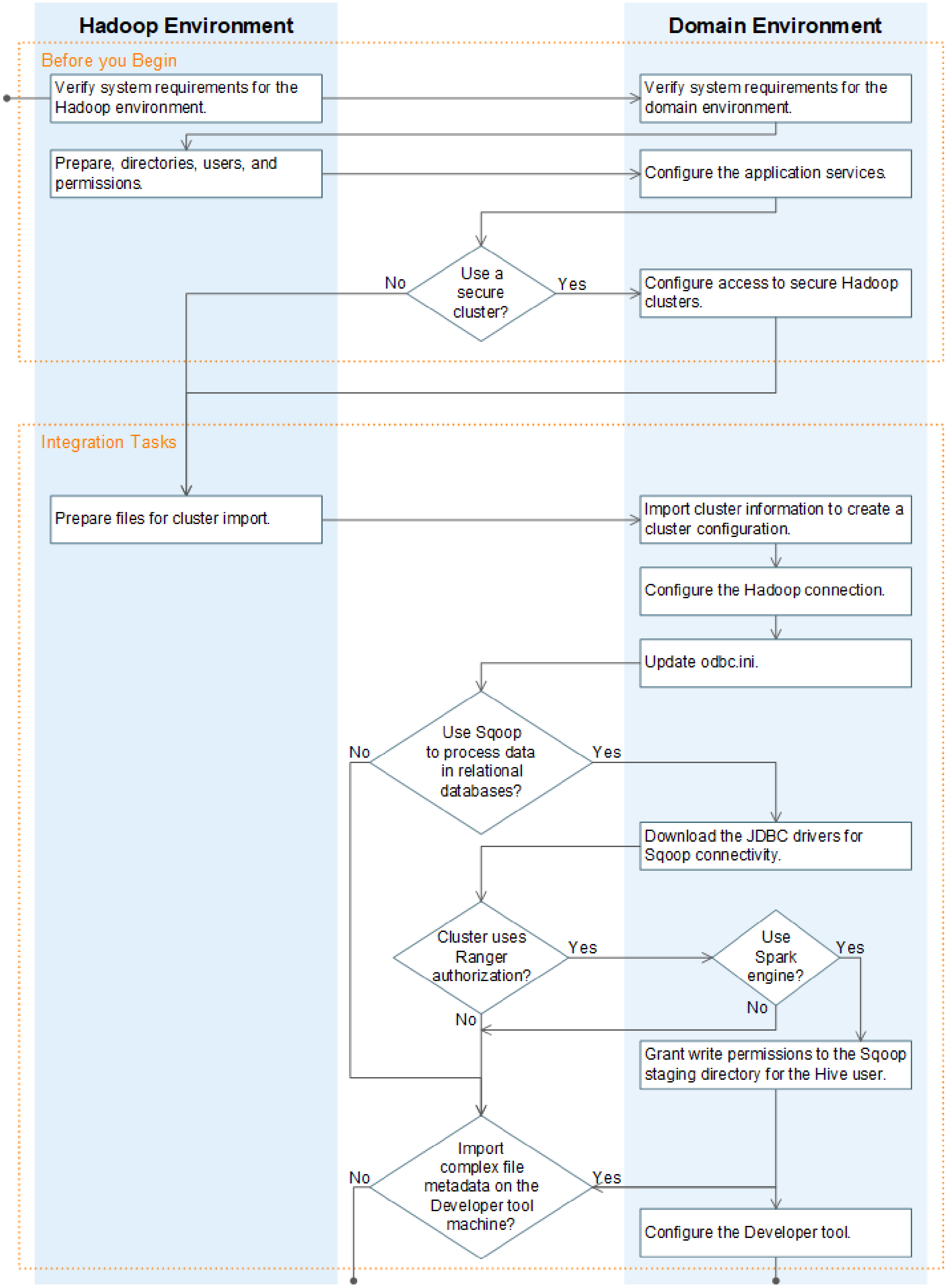
Hortonworks HDP Task Flows

Depending on whether you want to integrate or upgrade Big Data Management in a Hortonworks HDP environment, you can use the flow charts to perform the following tasks:

- Integrate the Informatica domain with Hortonworks HDP for the first time.
- Upgrade from version 10.2.1.
- Upgrade from version 10.2.
- Upgrade from a version earlier than 10.2.

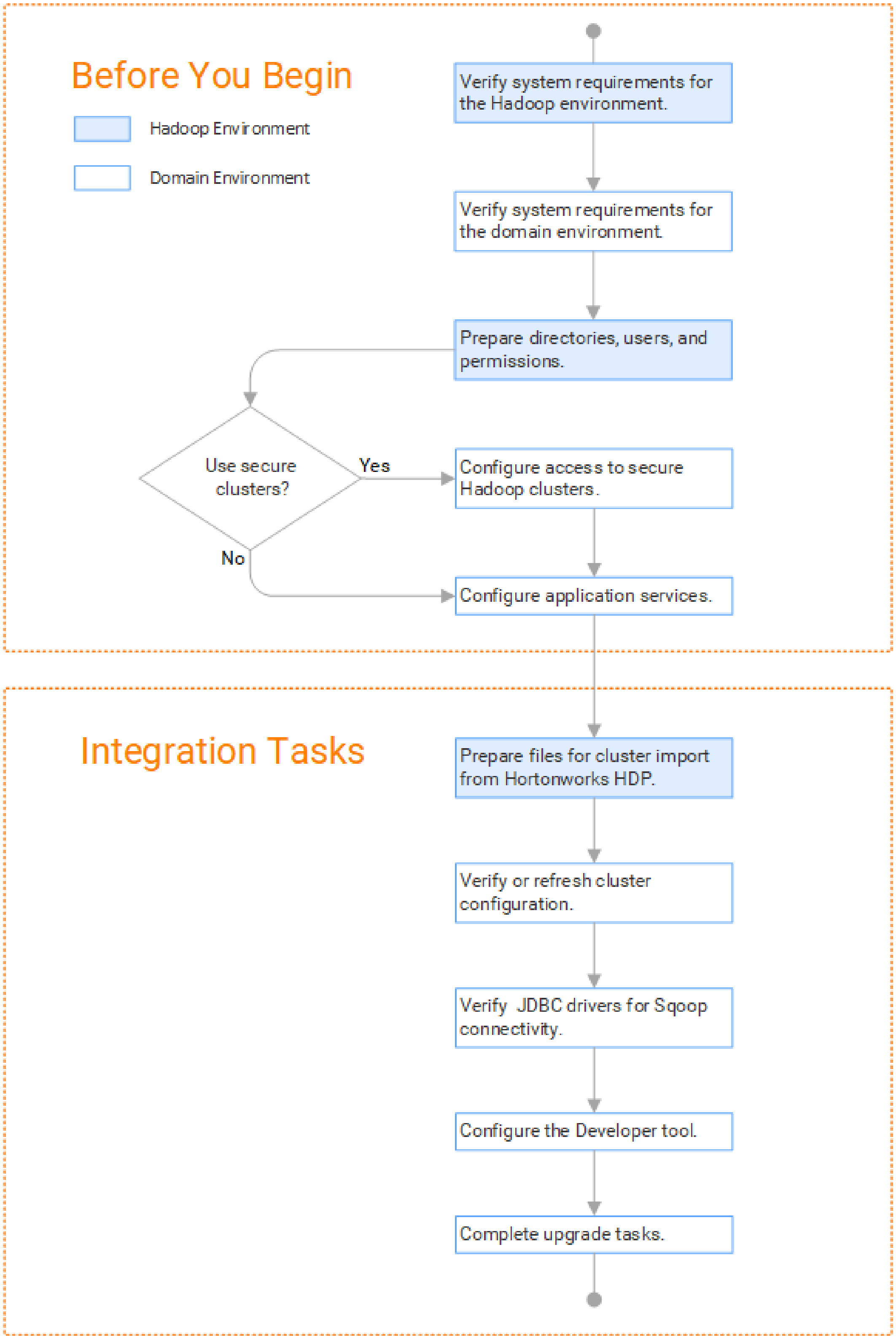
Task Flow to Integrate with Hortonworks HDP

The following diagram shows the task flow to integrate the Informatica domain with Hortonworks HDP:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare, directories, users, and permissions.
Configure the application services.
Use a secure cluster?
No
Yes
Configure access to secure Hadoop clusters.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure the Hadoop connection.
Update odbc.ini.
Use Sqoop to process data in relational databases?
No
Yes
Download the JDBC drivers for Sqoop connectivity.
Cluster uses Ranger authorization?
Yes
No
Use Spark engine?
Yes
No
Grant write permissions to the Sqoop staging directory for the Hive user.
Import complex file metadata on the Developer tool machine?
No
Yes
Configure the Developer tool.

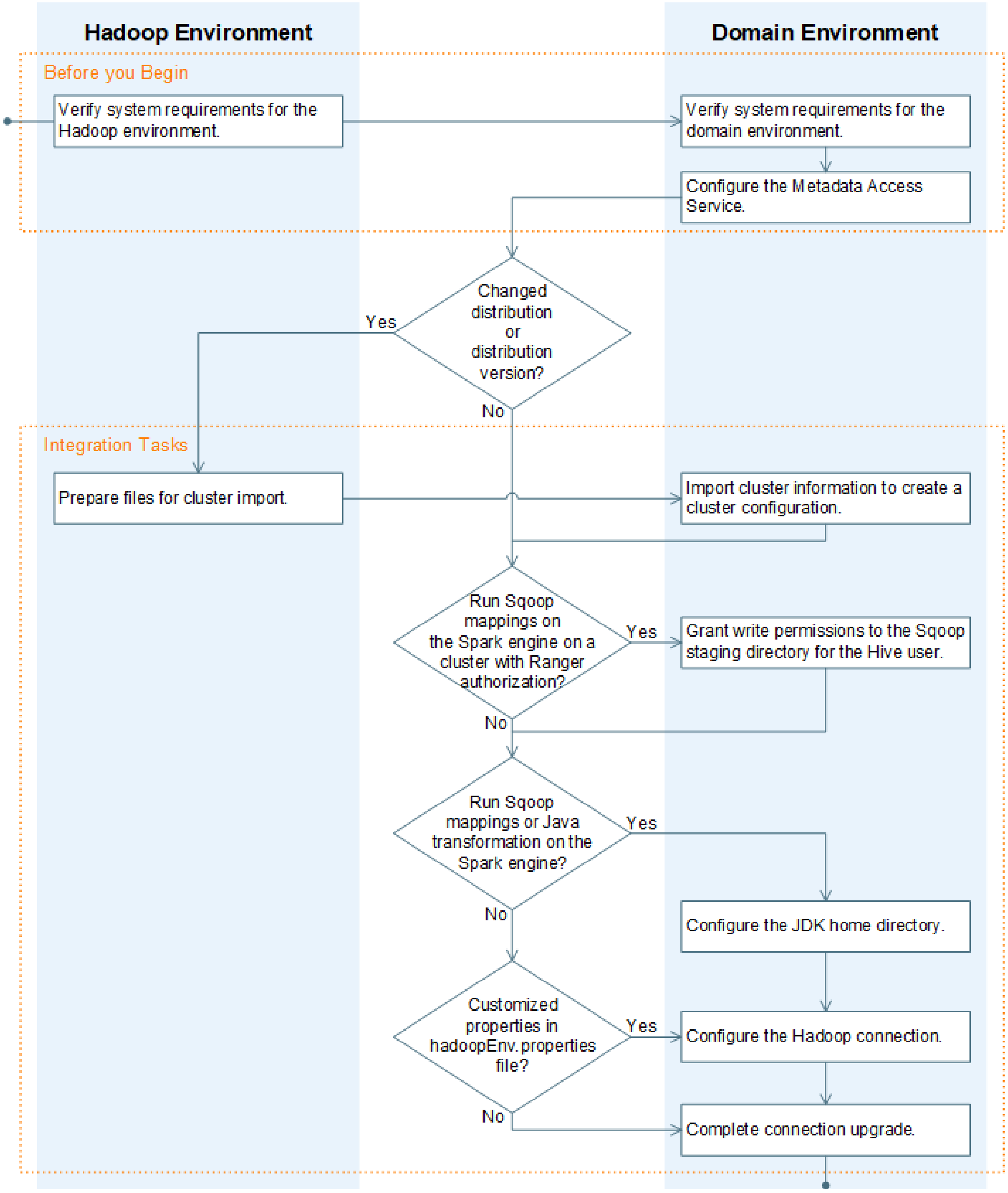
Task Flow to Upgrade from Version 10.2.1

The following diagram shows the task flow to upgrade Big Data Management from version 10.2.1 for Hortonworks HDP:


Before You Begin
Hadoop Environment
Domain Environment
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare directories, users, and permissions.
Use secure clusters?
Yes
No
Configure access to secure Hadoop clusters.
Configure application services.
Integration Tasks
Prepare files for cluster import from Hortonworks HDP.
Verify or refresh cluster configuration.
Verify JDBC drivers for Sqoop connectivity.
Configure the Developer tool.
Complete upgrade tasks.

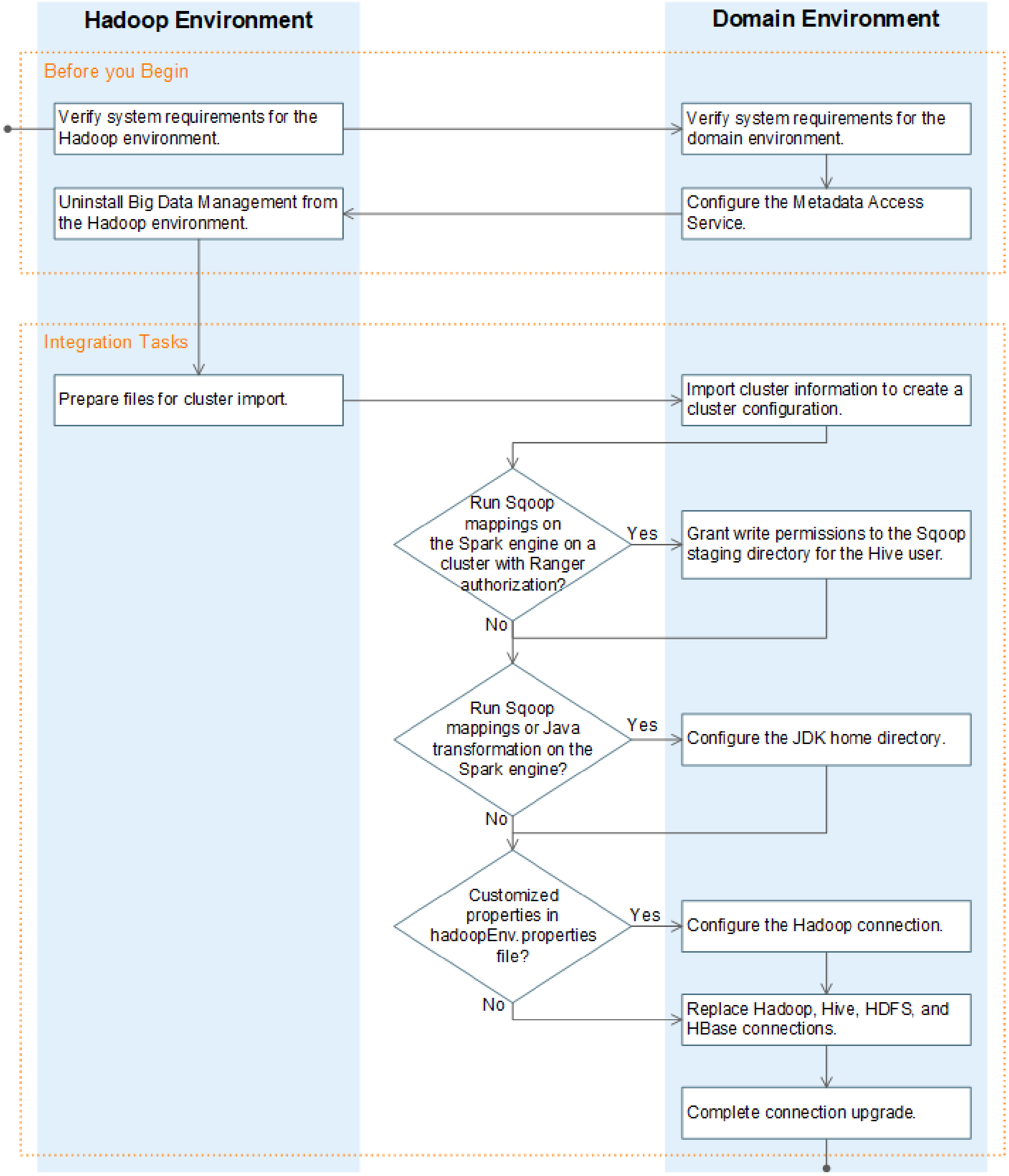
Task Flow to Upgrade from Version 10.2

The following diagram shows the task flow to upgrade Big Data Management 10.2 for Hortonworks HDP:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Changed distribution or distribution version?
Yes
No
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Run Sqoop mappings on the Spark engine on a cluster with Ranger authorization?
Yes
Grant write permissions to the Sqoop staging directory for the Hive user.
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Complete connection upgrade.

Task Flow to Upgrade from a Version Earlier than 10.2

The following diagram shows the task flow to upgrade Big Data Management from a version earlier than 10.2 for Hortonworks HDP:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Uninstall Big Data Management from the Hadoop environment.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Run Sqoop mappings on the Spark engine on a cluster with Ranger authorization?
Yes
Grant write permissions to the Sqoop staging directory for the Hive user.
No
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Replace Hadoop, Hive, HDFS, and HBase connections.
Complete connection upgrade.

Prepare for Cluster Import from Hortonworks HDP

Before the Informatica administrator can import cluster information to create a cluster configuration in the Informatica domain, the Hadoop administrator must perform some preliminary tasks.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded Big Data Management.

Note: If you are upgrading from a previous version, verify the properties and suggested values, as Big Data Management might require additional properties or different values for existing properties.

Complete the following tasks to prepare the cluster before the Informatica administrator creates the cluster configuration:

1. Verify property values in *-site.xml files that Big Data Management needs to run mappings in the Hadoop environment.
2. Provide information to the Informatica administrator that is required to import cluster information into the domain. Depending on the method of import, perform one of the following tasks:
   - To import directly from the cluster, give the Informatica administrator cluster authentication information to connect to the cluster.
   - To import from an archive file, export cluster information and provide an archive file to the Big Data Management administrator.

Configure *-site.xml Files for Hortonworks HDP

The Hadoop administrator needs to configure *-site.xml file properties and restart impacted services before the Informatica administrator imports cluster information into the domain.

core-site.xml

Configure the following properties in the core-site.xml file:

fs.s3.enableServerSideEncryption

Enables server side encryption for hive buckets. Required if the S3 bucket is encrypted. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to: TRUE

fs.s3a.access.key

The ID for the Blaze and Spark engines to connect to the Amazon S3 file system.

Set to your access key.

fs.s3a.secret.key

The password for the Blaze and Spark engines to connect to the Amazon S3 file system

Set to your access ID.

fs.s3a.server-side-encryption-algorithm

The server-side encryption algorithm for S3. Required if the S3 bucket is encrypted using an algorithm. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to the encryption algorithm used.

hadoop.proxyuser.<proxy user>.groups

Defines the groups that the proxy user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.<proxy user>.hosts

Defines the host machines that a user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

hadoop.proxyuser.yarn.groups

Comma-separated list of groups that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.yarn.hosts

Comma-separated list of hosts that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

hadoop.security.auth_to_local

Translates the principal names from the Active Directory and MIT realm into local names within the Hadoop cluster. Based on the Hadoop cluster used, you can set multiple rules.

Set to: RULE:[1:$1@$0](^.*@YOUR.REALM)s/^(.*)@YOUR.REALM\.COM$/$1/g

Set to: RULE:[2:$1@$0](^.*@YOUR.REALM\.$)s/^(.*)@YOUR.REALM\.COM$/$1/g

hbase-site.xml

Configure the following properties in the hbase-site.xml file:

zookeeper.znode.parent

Identifies HBase master and region servers.

Set to the relative path to the znode directory of HBase.

hdfs-site.xml

Configure the following properties in the hdfs-site.xml file:

dfs.encryption.key.provider.uri

The KeyProvider used to interact with encryption keys when reading and writing to an encryption zone. Required if sources or targets reside in the HDFS encrypted zone on Java KeyStore KMS-enabled Cloudera CDH cluster or a Ranger KMS-enabled Hortonworks HDP cluster.

Set to: kmf://http@xx11.xyz.com:16000/kms

hive-site.xml

Configure the following properties in the hive-site.xml file:

hive.cluster.delegation.token.store.class

The token store implementation. Required for HiveServer2 high availability and load balancing.

Set to: org.apache.hadoop.hive.thrift.DBTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

io.compression.codecs

Enables compression on temporary staging tables.

Set to a comma-separated list of compression codec classes on the cluster.

hive.exec.dynamic.partition.mode

Allows all partitions to be dynamic. Required for the Update Strategy transformation in a mapping that writes to a Hive target. Also required if you use Sqoop and define a DDL query to create or replace a partitioned Hive target at run time.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.server2.support.dynamic.service.discovery

Enables HiveServer2 dynamic service discovery. Required for HiveServer2 high availability.

Set to: TRUE

hive.server2.zookeeper.namespace

The value of the ZooKeeper namespace in the JDBC connection string. Required for HiveServer2 high availability.

Set to: `jdbc:hive2://<zookeeper_ensemble>/default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2`

hive.txn.manager

Turns on transaction support. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

hive.zookeeper.quorum

Comma-separated list of ZooKeeper server host:ports in a cluster. The value of the ZooKeeper ensemble in the JDBC connection string. Required for HiveServer2 high availability.

Set to: `jdbc:hive2://<zookeeper_ensemble>/default;serviceDiscoveryMode=zooKeeper;`

mapred-site.xml

Configure the following properties in the mapred-site.xml file:

mapreduce.framework.name

The run-time framework to run MapReduce jobs. Values can be local, classic, or yarn. Required for Sqoop.

Set to: yarn

yarn.app.mapreduce.am.staging-dir

The HDFS staging directory used while submitting jobs.

Set to the staging directory path.

yarn-site.xml

Configure the following properties in the yarn-site.xml file:

yarn.application.classpath

Required for dynamic resource allocation.

"Add spark_shuffle.jar to the class path". The .jar file must contain the class "org.apache.network.yarn.YarnShuffleService."

yarn.nodemanager.resource.memory-mb

The maximum RAM available for each container. Set the maximum memory on the cluster to increase resource memory available to the Blaze engine.

Set to 16 GB if value is less than 16 GB.

yarn.nodemanager.resource.cpu-vcores

The number of virtual cores for each container. Required for Blaze engine resource allocation.

Set to 10 if the value is less than 10.

yarn.scheduler.minimum-allocation-mb

The minimum RAM available for each container. Required for Blaze engine resource allocation.

Set to 6 GB if the value is less than 6 GB.

yarn.nodemanager.vmem-check-enabled

Disables virtual memory limits for containers. Required for the Blaze and Spark engines.

Set to: FALSE

yarn.nodemanager.aux-services

Required for dynamic resource allocation for the Spark engine.

Add an entry for "spark_shuffle."

yarn.nodemanager.aux-services.spark_shuffle.class

Required for dynamic resource allocation for the Spark engine.

Set to: org.apache.spark.network.yarn.YarnShuffleService

yarn.resourcemanager.scheduler.class

Defines the YARN scheduler that the Data Integration Service uses to assign resources.

Set to: org.apache.hadoop.yarn.server.resourcemanager.scheduler

yarn.node-labels.enabled

Enables node labeling.

Set to: TRUE

yarn.node-labels.fs-store.root-dir

The HDFS location to update node label dynamically.

Set to: <hdfs://[Node name]:[Port]/[Path to store]/[Node labels]/>

tez-site.xml

Configure the following properties in the tez-site.xml file:

tez.runtime.io.sort.mb

The sort buffer memory. Required when the output needs to be sorted for Blaze and Spark engines.

Set value to 270 MB.

Prepare for Direct Import from Hortonworks HDP

If you plan to provide direct access to the Informatica administrator to import cluster information, provide the required connection information.

The following table describes the information that you need to provide to the Informatica administrator to create the cluster configuration directly from the cluster:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster.

Prepare the Archive File for Import from Hortonworks HDP

When you prepare the archive file for cluster configuration import from Hortonworks, include all required *-site.xml files and edit the file manually after you create it.

The Hortonworks cluster configuration archive file must have the following contents:

- core-site.xml
- hbase-site.xml. hbase-site.xml is required only if you access HBase sources and targets.
- hdfs-site.xml
- hive-site.xml

- mapred-site.xml or tez-site.xml. Include the mapred-site.xml file or the tez-site.xml file based on the Hive execution type used on the Hadoop cluster.
- yarn-site.xml

Update the Archive File

After you create the archive file, edit the Hortonworks Data Platform (HDP) version string wherever it appears in the archive file. Search for the string `${hdp.version}` and replace all instances with the HDP version that Hortonworks includes in the Hadoop distribution.

For example, the edited tez.lib.uris property looks similar to the following:

```
<property>
<name>tez.lib.uris</name>
<value>/hdp/apps/2.5.0.0-1245/tez/tez.tar.gz</value>
</property>
```

Create a Cluster Configuration

After the Hadoop administrator prepares the cluster for import, the Informatica administrator must create a cluster configuration.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded from a version earlier than 10.2. - You upgraded from 10.2 and changed the distribution or distribution version.

A cluster configuration is an object in the domain that contains configuration information about the Hadoop cluster. The cluster configuration enables the Data Integration Service to push mapping logic to the Hadoop environment. Import configuration properties from the Hadoop cluster to create a cluster configuration.

The import process imports values from *-site.xml files into configuration sets based on the individual *-site.xml files. When you perform the import, the cluster configuration wizard can create Hadoop, HBase, HDFS, and Hive connection to access the Hadoop environment. If you choose to create the connections, the wizard also associates the cluster configuration with the connections.

Note: If you imported the cluster configuration when you installed Enterprise Data Lake with the Informatica domain, you can create the cluster configuration again or refresh the cluster configuration.

For more information about the cluster configuration, see the *Big Data Management Administrator Guide*.

Before You Import

Before you can import the cluster configuration, you must get information from the Hadoop administrator based on the method of import.

If you import directly from the cluster, contact the Hadoop administrator to get cluster connection information. If you import from a file, get an archive file of exported cluster information.

Importing a Cluster Configuration from the Cluster

When you import the cluster configuration directly from the cluster, you provide information to connect to the cluster.

Get cluster connection information from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following General properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. The default version is the latest version of the Hadoop distribution that Big Data Management supports. **Note:** When the cluster version differs from the default version and Informatica supports more than one version, the cluster configuration import process populates the property with the most recent supported version. For example, consider the case where Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the cluster configuration import process populates this property with 5.10, because 5.10 is the most recent supported version before 5.12. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.
Method to import the cluster configuration	Choose **Import from cluster**.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

   The cluster properties appear.

4. Configure the following properties:

Property	Description
Host	IP address of the cluster manager.
Port	Port of the cluster manager.
User ID	Cluster user ID.
Password	Password for the user.
Cluster name	Name of the cluster. Use the display name if the cluster manager manages multiple clusters. If you do not provide a cluster name, the wizard imports information based on the default cluster.

5. Click **Next** and verify the cluster configuration information on the summary page.

Importing a Cluster Configuration from a File

You can import properties from an archive file to create a cluster configuration.

Before you import from the cluster, you must get the archive file from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. This is the latest version of the Hadoop distribution that Big Data Management supports. When the cluster version differs from the default version, the cluster configuration wizard populates the cluster configuration Hadoop distribution property with the most recent supported version relative to the cluster version. For example, suppose Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the wizard populates the version with 5.10. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.

Property	Description
Method to import the cluster configuration	Choose **Import from file** to import properties from an archive file.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

4. Click **Browse** to select a file. Select the file and click **Open**.
5. Click **Next** and verify the cluster configuration information on the summary page.

Verify or Refresh the Cluster Configuration

You might need to refresh the cluster configuration or update the distribution version in the cluster configuration when you upgrade.

Verify the Cluster Configuration

The cluster configuration contains a property for the distribution version. The verification task depends on the version you upgraded:

Upgrade from 10.2

If you upgraded from 10.2 and you changed the distribution version, you need to verify the distribution version in the General properties of the cluster configuration.

Upgrade from 10.2.1

Effective in version 10.2.1, Informatica assigns a default version to each Hadoop distribution type. If you configure the cluster configuration to use the default version, the upgrade process upgrades to the assigned default version if the version changes. If you have not upgraded your Hadoop distribution to Informatica's default version, you need to update the distribution version property.

For example, suppose the assigned default Hadoop distribution version for 10.2.1 is *n*, and for 10.2.2 is *n +1*. If the cluster configuration uses the default supported Hadoop version of *n*, the upgraded cluster configuration uses the default version of *n+1*. If you have not upgraded the distribution in the Hadoop environment you need to change the cluster configuration to use version *n*.

If you configure the cluster configuration to use a distribution version that is not the default version, you need to update the distribution version property in the following circumstances:

- Informatica dropped support for the distribution version.
- You changed the distribution version.

Refresh the Cluster Configuration

If you updated any of the *-site.xml files noted in the topic to prepare for cluster import, you need to refresh the cluster configuration in the Administrator tool.

Verify JDBC Drivers for Sqoop Connectivity

Verify that you have the JDBC drivers to access JDBC-compliant databases in the Hadoop environment. You might need separate drivers for metadata import and for run-time processing.

You download drivers based on design-time and run-time requirements:

- **Design-time.** To import metadata, you can use the DataDirect drivers packaged with the Informatica installer if they are available. If they are not available, use any Type 4 JDBC driver that the database vendor recommends.
- **Run-time.** To run mappings, use any Type 4 JDBC driver that the database vendor recommends. Some distributions support other drivers to use Sqoop connectors. You cannot use the DataDirect drivers for run-time processing.

Verify Design-time Drivers

Use the DataDirect JDBC drivers packaged with the Informatica installer to import metadata from JDBC-compliant databases. If the DataDirect JDBC drivers are not available for a specific JDBC-compliant database, download the Type 4 JDBC driver associated with that database.

Copy the JDBC driver .jar files to the following location on the Developer tool machine:

```
<Informatica installation directory>\clients\externaljdbcjars
```

Verify Run-time Drivers

Verify run-time drivers for mappings that access JDBC-compliant databases in the Hadoop environment. Use any Type 4 JDBC driver that the database vendor recommends.

1. Download Type 4 JDBC drivers associated with the JCBC-compliant databases that you want to access.
2. To use Sqoop TDCH Hortonworks Connector for Teradata, perform the following task:

   Download all .jar files in the Hortonworks Connector for Teradata package from the following location : http://hortonworks.com/downloads/#addons

   The package has the following naming convention: `hdp-connector-for-teradata-<version>-distro.tar.gz`
3. To optimize the Sqoop mapping performance on the Spark engine while writing data to an HDFS complex file target of the Parquet format, download the following .jar files:
   - `parquet-hadoop-bundle-1.6.0.jar` from http://central.maven.org/maven2/com/twitter/parquet-avro/1.6.0/
   - `parquet-avro-1.6.0.jar` from http://central.maven.org/maven2/com/twitter/parquet-hadoop-bundle/1.6.0/
   - `parquet-column-1.5.0.jar` from http://central.maven.org/maven2/com/twitter/parquet-column/1.5.0/

4. Copy all of the .jar files to the following directory on the machine where the Data Integration Service runs:

   ```
   <Informatica installation directory>\externaljdbcjars
   ```

   Changes take effect after you recycle the Data Integration Service. At run time, the Data Integration Service copies the .jar files to the Hadoop distribution cache so that the .jar files are accessible to all nodes in the cluster.

Import Security Certificates to Clients

When you use custom, special, or self-signed security certificates to secure the Hadoop cluster, Informatica clients that connect to the cluster require these certificates to be present in the client machine truststore.

To connect to the Hadoop cluster to develop a mapping, the Developer tool requires security certificate aliases on the machine that hosts the Developer tool. To run a mapping, the machine that hosts the Data Integration Service requires these same certificate alias files.

Perform the following steps from the Developer tool host machine, and then repeat them from the Data Integration Service host machine:

1. Run the following command to export the certificates from the cluster:

   ```
   keytool -export -alias <alias name> -keystore <custom.truststore file location> -
   file <exported certificate file location> -storepass <password>
   ```

   For example,

   ```
   keytool -export -alias <alias name> -keystore ~/custom.truststore -file ~/
   exported.cer
   ```

   The command produces a certificate file.

2. Choose to import security certificates to an SSL-enabled domain or a domain that is not SSL-enabled using the following command:

   ```
   keytool -import -trustcacerts -alias <alias name> -file <exported certificate file
   location> -keystore <java cacerts location> -storepass <password>
   ```

   For example,

   ```
   keytool -import -alias <alias name> -file ~/exported.cer -keystore <Informatica
   installation directory>/java/jre/lib/security/cacerts
   ```

Configure the Developer Tool

To access the Hadoop environment from the Developer tool, the mapping developers must perform tasks on each Developer tool machine.

Configure developerCore.ini

Edit `developerCore.ini` to successfully import local complex files available on the Developer tool machine.

When you import a complex file, such as Avro or Parquet, the imported object includes metadata associated with the distribution in the Hadoop environment. If the file resides on the Developer tool machine, the import process picks up the distribution information from the `developerCore.ini` file. You must edit the `developerCore.ini` file to point to the distribution directory on the Developer tool machine.

You can find `developerCore.ini` in the following directory:

```
<Informatica installation directory>\clients\DeveloperClient
```

Add the following property:

```
-DINFA_HADOOP_DIST_DIR=hadoop\<distribution>_<version>
```

The change takes effect when you restart the Developer tool.

Complete Upgrade Tasks

If you upgraded the Informatica platform, you need to perform some additional tasks within the Informatica domain.

Based on the version that you upgraded from, perform the following tasks:

Configure the JDK home directory.

To run Sqoop mappings or process a Java transformation on the Spark engine, you must configure the JDK Home Directory in the Data Integration Service properties.

Configure the Hadoop connection.

The Hadoop connection contains additional properties. You need to manually update it to include customized configuration in the hadoopEnv.properties file from previous versions.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections. The tasks to complete connection upgrade depends on the Informatica version that you upgraded.

Update Connections

You might need to update connections based on the version you are upgrading from.

Consider the following types of updates that you might need to make:

Configure the Hadoop connection.

Configure the Hadoop connection to incorporate properties from the hadoopEnv.properties file.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections.

Configure the Hadoop Connection

To use properties that you customized in the hadoopEnv.properties file, you must configure the Hadoop connection properties such as cluster environment variables, cluster path variables, and advanced properties.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

When you run the Informatica upgrade, the installer backs up the existing hadoopEnv.properties file. You can find the backup hadoopEnv.properties file in the following location:

```
<Previous Informatica installation directory>/services/shared/hadoop/<Hadoop distribution
name>_<version>/infaConf
```

Edit the Hadoop connection in the Administrator tool or the Developer tool to include any properties that you manually configured in the hadoopEnv.properties file. The Hadoop connection contains default values for properties such as cluster environment and path variables and advanced properties. You can update the default values to match the properties in the hadoopEnv.properties file.

Replace the Connections with New Connections

If you created connections when you imported the cluster configuration, you need to replace connections in mappings with the new connections.

Perform this task in the following situation:
- You upgraded from a version earlier than 10.2.

The method that you use to replace connections in mappings depends on the type of connection.

Hadoop connection

Run the following commands to replace the connections:

- `infacmd dis replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that are deployed in applications.
- `infacmd mrs replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that you run from the Developer tool.

For information about the infacmd commands, see the *Informatica Command Reference*.

Hive, HDFS, and HBase connections

You must replace the connections manually.

Complete Connection Upgrade

If *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

Perform the following tasks to update the connections:

Update changed properties

Review connections that you created in a previous release to update the values for connection properties. For example, if you added nodes to the cluster or if you updated the distribution version, you might need to verify host names, URIs, or port numbers for some of the properties.

Associate the cluster configuration

The Hadoop, Hive, HDFS, and HBase connections must be associated with a cluster configuration. Complete the following tasks:

1. Run `infacmd isp listConnections` to identify the connections that you need to upgrade. Use `-ct` to list connections of a particular type.
2. Run `infacmd isp UpdateConnection` to associate the cluster configuration with the connection. Use `-cn` to name the connection and `-o clusterConfigID` to associate the cluster configuration with the connection.

For more information about infacmd, see the *Informatica Command Reference*.

Update Streaming Objects

Big Data Streaming uses Spark Structured Streaming to process data instead of Spark Streaming. To support Spark Structured Streaming, some header ports are added to the data objects, and support to some of the data objects and transformations are deferred to a future release. The behavior of some of the data objects is also updated.

After you upgrade, the existing streaming mappings become invalid because of the unavailable header ports, the unsupported transformations or data objects, and the behavior change of some data objects.

To use an existing mapping, perform the following tasks:

- Re-create the physical data objects. After you re-create the physical data objects, the data objects get the required header ports, such as timestamp, partitionID, or key based on the data object.
- Re-create the Normalizer transformation. After you re-create the Normalizer transformation, you can change or specify the type configuration for the complex port.
- Update the streaming mapping. If the mapping contains Kafka target, Aggregator transformation, Joiner transformation, or Normalizer transformation replace the data object or transformation, and then update the mapping because of the changed behavior of these transformations and data objects.
- Verify the deferred data object types. If the streaming mapping contains unsupported transformations or data objects, contact Informatica Global Customer Support.

Re-create the Physical Data Objects

When you re-create the physical data objects, the physical data objects get the header ports and some properties are not available for some data objects. Update the existing mapping with the newly created physical data objects.

1. Go to the existing mapping, select the data object from the mapping.
2. Click the **Properties** tab. On the **Column Projection** tab, click **Edit Schema**.
3. Note the schema information from the **Edit Schema** dialog box.
4. Note the parameters information from the **Parameters** tab.
5. Create new physical data objects.

After you re-create the data objects, the physical data objects get the required header ports. The Microsoft Azure does not support the following properties and are not available for Azure Event Hubs data objects:

- Consumer Properties
- Partition Count

Update the Streaming Mappings

After you re-create the data object, replace the existing data objects with the re-created data objects. If the mapping contains Normaliser Transformation, Aggregator transformation, or Joiner transformation, update the mapping because of the changed behavior of these transformations and data objects.

Aggregator and Joiner Transformation Updates

An Aggregator transformation must be downstream from a Joiner transformation. A Window transformation must be directly upstream from both Aggregator and Joiner transformations. Previously, you could use an Aggregator transformation anywhere in the streaming mapping.

If a mapping contains an Aggregator transformation upstream from a Joiner transformation, move the Aggregator transformation downstream from a Joiner transformation. Add a Window transformation directly upstream from both Aggregator and Joiner transformations.

Transformation Updates

If a streaming mapping contains a transformation that uses a complex data types, you must manually set the property to an appropriate type configuration.

Verify the Deferred Data Object Types

After you upgrade, the streaming mappings might contain some transformations and data objects that are deferred.

The following table lists the data object types to which the support is deferred to a future release:

Object Type	Object
Source	JMS MapR Streams
Target	MapR Streams
Transformation	Data Masking Joiner Rank Sorter

If you want to continue using the mappings that contain deferred data objects or transformations, you must contact Informatica Global Customer Support.

CHAPTER 7

MapR Integration Tasks

This chapter includes the following topics:

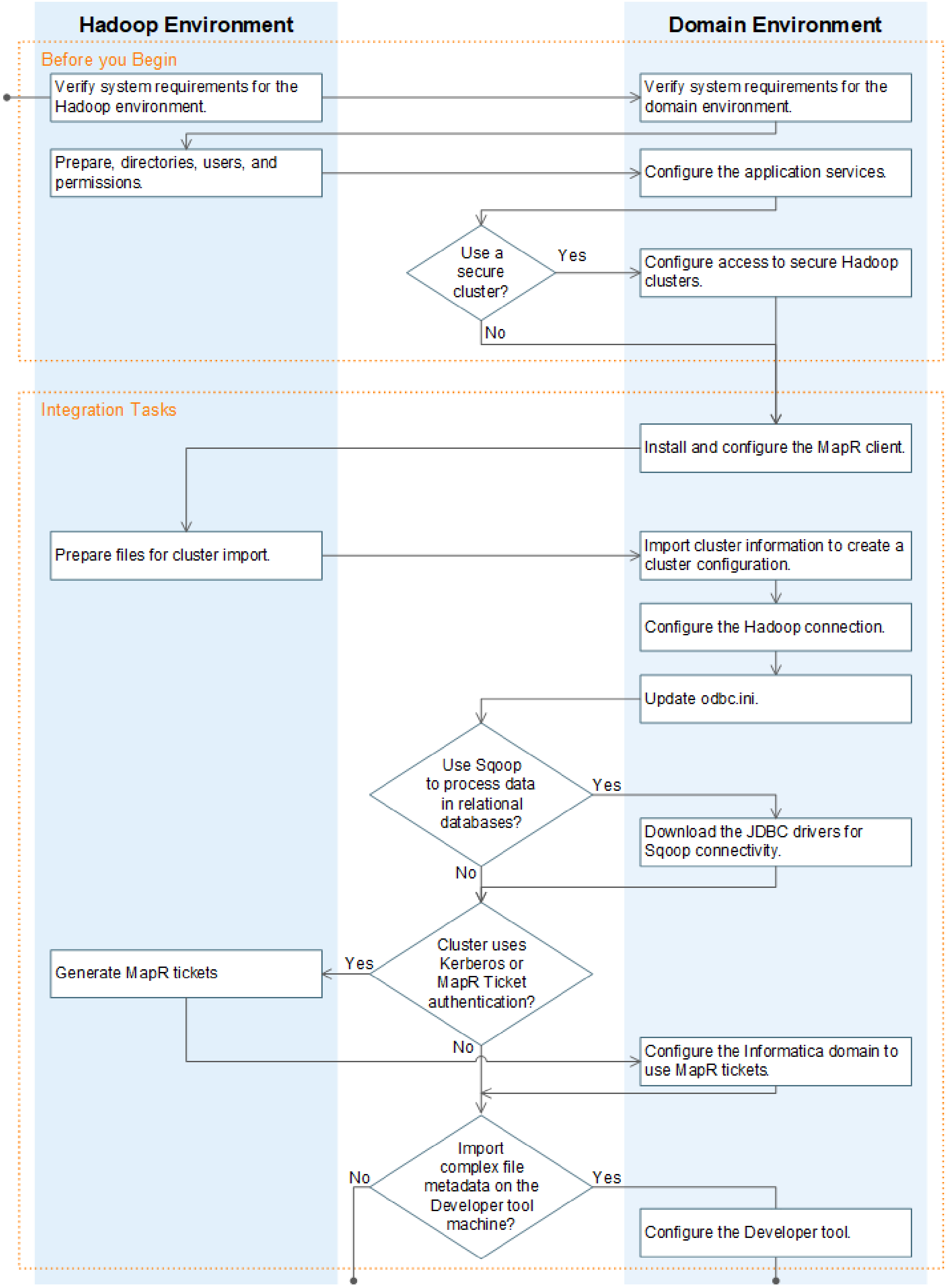
MapR Task Flows

Depending on whether you want to integrate or upgrade Big Data Management in a MapR environment, you can use the flow charts to perform the following tasks:

- Integrate the Informatica domain with MapR for the first time.
- Upgrade from version 10.2.1.
- Upgrade from version 10.2.
- Upgrade from a version earlier than 10.2.

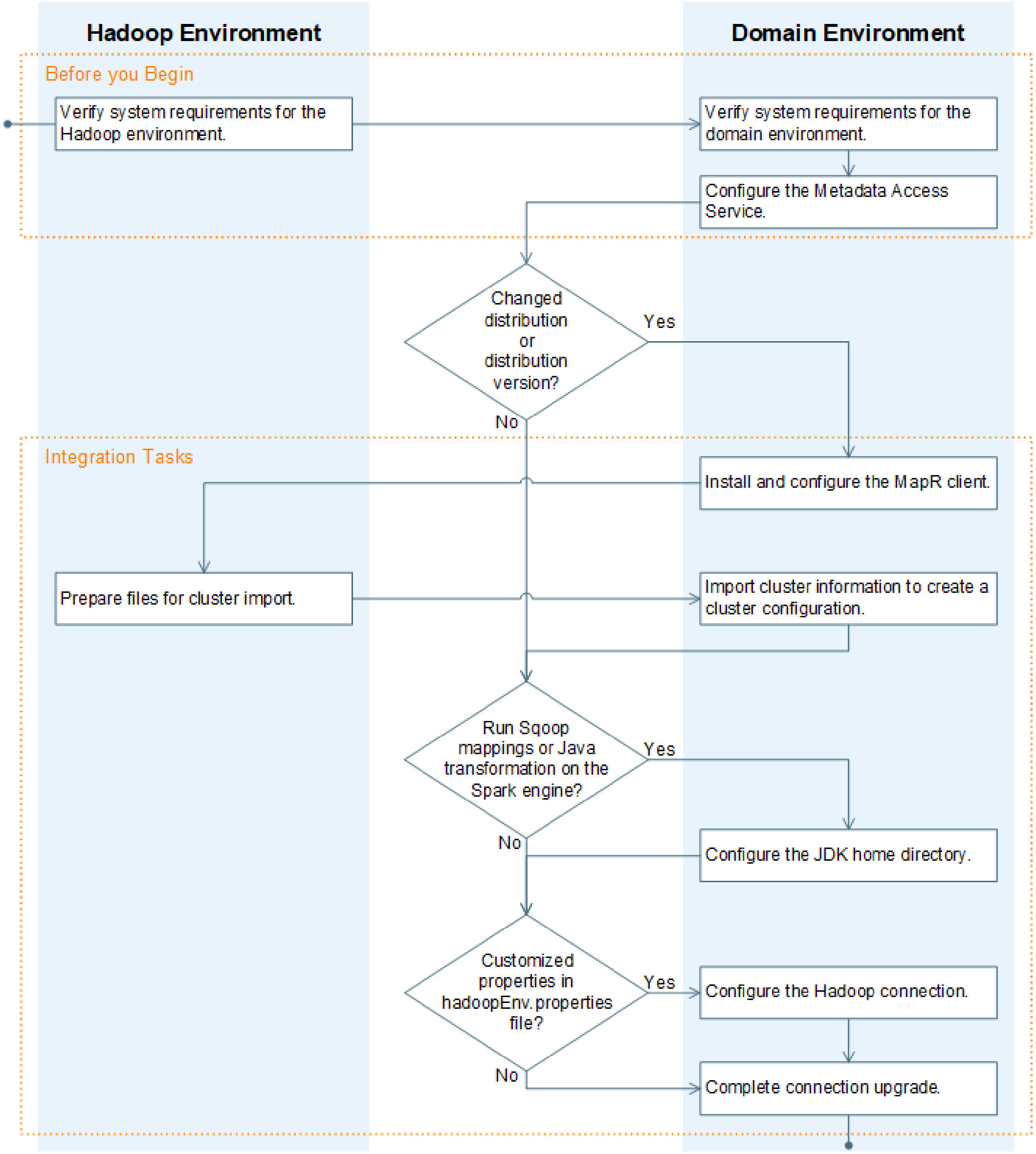
Task Flow to Integrate with MapR

The following diagram shows the task flow to integrate the Informatica domain with MapR:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Prepare, directories, users, and permissions.
Configure the application services.
Use a secure cluster?
Yes
Configure access to secure Hadoop clusters.
No
Integration Tasks
Install and configure the MapR client.
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Configure the Hadoop connection.
Update odbc.ini.
Use Sqoop to process data in relational databases?
Yes
Download the JDBC drivers for Sqoop connectivity.
No
Generate MapR tickets
Yes
Cluster uses Kerberos or MapR Ticket authentication?
No
Configure the Informatica domain to use MapR tickets.
No
Import complex file metadata on the Developer tool machine?
Yes
Configure the Developer tool.

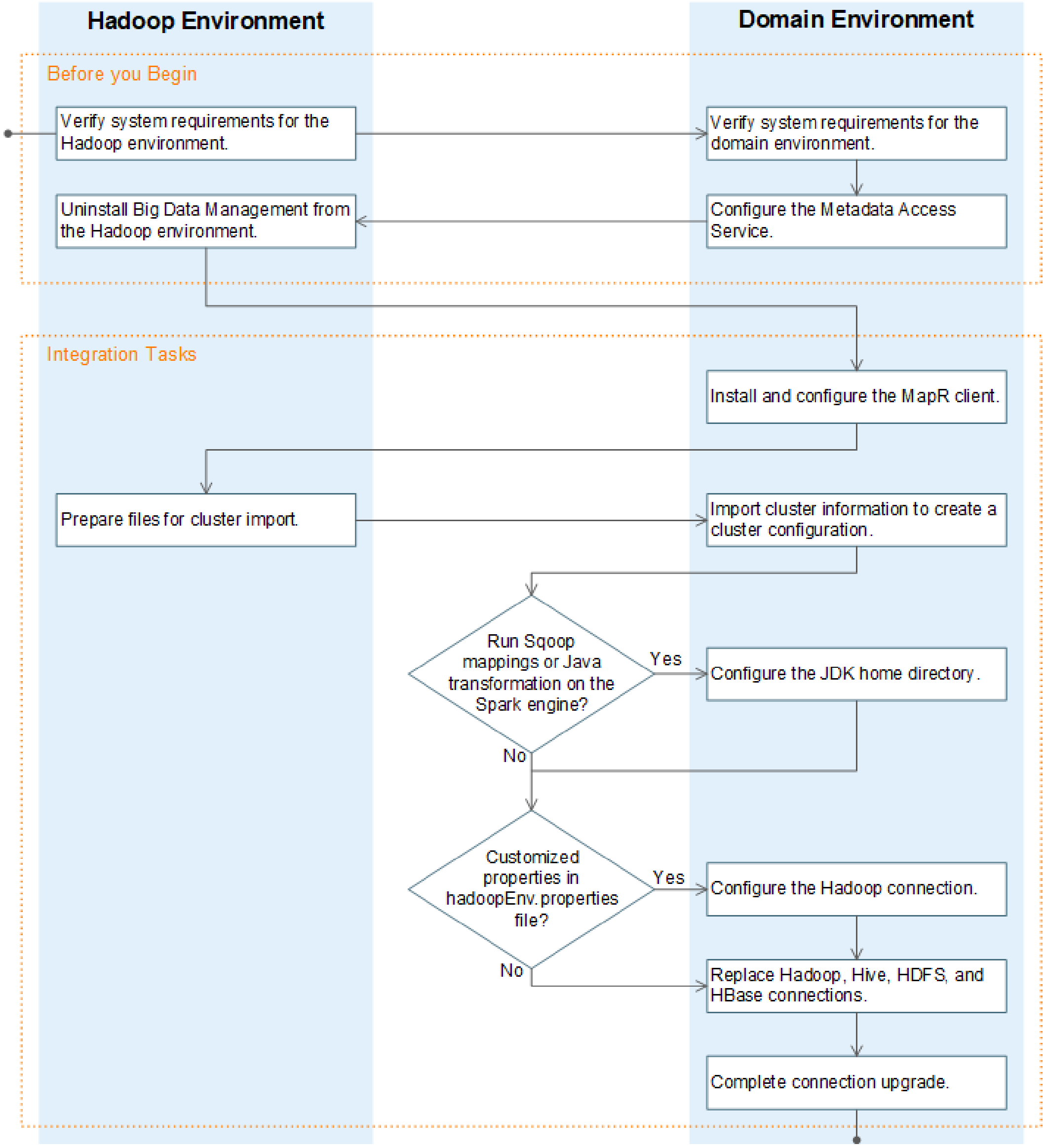
Task Flow to Upgrade from Version 10.2

The following diagram shows the task flow to upgrade Big Data Management 10.2 for MapR:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Changed distribution or distribution version?
Yes
No
Integration Tasks
Install and configure the MapR client.
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
No
Configure the JDK home directory.
Customized properties in hadoopEnv.properties file?
Yes
No
Configure the Hadoop connection.
Complete connection upgrade.

Task Flow to Upgrade from a Version Earlier than 10.2

The following diagram shows the task flow to upgrade Big Data Management from a version earlier than 10.2 for MapR:


Hadoop Environment
Domain Environment
Before you Begin
Verify system requirements for the Hadoop environment.
Verify system requirements for the domain environment.
Configure the Metadata Access Service.
Uninstall Big Data Management from the Hadoop environment.
Integration Tasks
Install and configure the MapR client.
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Run Sqoop mappings or Java transformation on the Spark engine?
Yes
Configure the JDK home directory.
No
Customized properties in hadoopEnv.properties file?
Yes
Configure the Hadoop connection.
No
Replace Hadoop, Hive, HDFS, and HBase connections.
Complete connection upgrade.

Install and Configure the MapR Client

To enable communication between the Informatica domain and the MapR cluster, install and configure the MapR client on the application service machines. The MapR client version on the MapR cluster and the application service machines must match.

You install the MapR client on the Data Integration Service, Metadata Access Service, and Analyst Service machines in the following directory:

```
/opt/mapr
```

For instructions about installing and configuring the MapR client, refer to the MapR documentation at https://mapr.com/docs/60/AdvancedInstallation/SettingUptheClient-install-mapr-client.html.

Prepare for Cluster Import from MapR

Before the Informatica administrator can import cluster information to create a cluster configuration in the Informatica domain, the Hadoop administrator must perform some preliminary tasks.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded Big Data Management.

Note: If you are upgrading from a previous version, verify the properties and suggested values, as Big Data Management might require additional properties or different values for existing properties.

Complete the following tasks to prepare the cluster before the Informatica administrator creates the cluster configuration:

1. Verify property values in *-site.xml files that Big Data Management needs to run mappings in the Hadoop environment.
2. Prepare the archive file to import into the domain.

Note: You cannot import cluster information directly from the MapR cluster into the Informatica domain.

Configure *-site.xml Files for MapR

The Hadoop administrator needs to configure *-site.xml file properties and restart impacted services before the Informatica administrator imports cluster information into the domain.

core.site.xml

Configure the following properties in the core-site.xml file:

fs.s3.enableServerSideEncryption

Enables server side encryption for hive buckets. Required if the S3 bucket is encrypted. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to: TRUE

fs.s3a.access.key

The ID for the Blaze and Spark engines to connect to the Amazon S3 file system.

Set to your access key.

fs.s3a.secret.key

The password for the Blaze and Spark engines to connect to the Amazon S3 file system

Set to your access ID.

fs.s3a.server-side-encryption-algorithm

The server-side encryption algorithm for S3. Required if the S3 bucket is encrypted using an algorithm. Required for EMR 5.14 integration if the S3 bucket is encrypted with SSE-KMS.

Set to the encryption algorithm used.

hadoop.proxyuser.<proxy user>.groups

Defines the groups that the proxy user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.<proxy user>.hosts

Defines the host machines that a user account can impersonate. On a secure cluster the <proxy user> is the Service Principal Name that corresponds to the cluster keytab file. On a non-secure cluster, the <proxy user> is the system user that runs the Informatica daemon.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

hadoop.proxyuser.yarn.groups

Comma-separated list of groups that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to group names of impersonation users separated by commas. If less security is preferred, use the wildcard " * " to allow impersonation from any group.

hadoop.proxyuser.yarn.hosts

Comma-separated list of hosts that you want to allow the YARN user to impersonate on a non-secure cluster.

Set to a single host name or IP address, or set to a comma-separated list. If less security is preferred, use the wildcard " * " to allow impersonation from any host.

io.compression.codecs

Enables compression on temporary staging tables.

Set to a comma-separated list of compression codec classes on the cluster.

hadoop.security.auth_to_local

Translates the principal names from the Active Directory and MIT realm into local names within the Hadoop cluster. Based on the Hadoop cluster used, you can set multiple rules.

Set to: RULE:[1:$1@$0](^.*@YOUR.REALM)s/^(.*)@YOUR.REALM\.COM$/$1/g

Set to: RULE:[2:$1@$0](^.*@YOUR.REALM\.$)s/^(.*)@YOUR.REALM\.COM$/$1/g

hbase-site.xml

Configure the following properties in the hbase-site.xml file:

zookeeper.znode.parent

Identifies HBase master and region servers.

Set to the relative path to the znode directory of HBase.

hive-site.xml

Configure the following properties in the hive-site.xml file:

hive.cluster.delegation.token.store.class

The token store implementation. Required for HiveServer2 high availability and load balancing.

Set to: org.apache.hadoop.hive.thrift.DBTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.exec.dynamic.partition

Enables dynamic partitioned tables for Hive tables. Applicable for Hive versions 0.9 and earlier.

Set to: TRUE

hive.exec.dynamic.partition.mode

Allows all partitions to be dynamic. Required for the Update Strategy transformation in a mapping that writes to a Hive target. Also required if you use Sqoop and define a DDL query to create or replace a partitioned Hive target at run time.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: TRUE

hive.server2.support.dynamic.service.discovery

Enables HiveServer2 dynamic service discovery. Required for HiveServer2 high availability.

Set to: TRUE

hive.server2.zookeeper.namespace

The value of the ZooKeeper namespace in the JDBC connection string. Required for HiveServer2 high availability.

Set to: `jdbc:hive2://<zookeeper_ensemble>/default;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2`

hive.txn.manager

Turns on transaction support. Required for an Update Strategy transformation in a mapping that writes to a Hive target.

Set to: org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

hive.zookeeper.quorum

Comma-separated list of ZooKeeper server host:ports in a cluster. The value of the ZooKeeper ensemble in the JDBC connection string. Required for HiveServer2 high availability.

Set to: `jdbc:hive2://<zookeeper_ensemble>/default;serviceDiscoveryMode=zooKeeper;`

mapred-site.xml

Configure the following properties in the mapred-site.xml file:

mapreduce.framework.name

The run-time framework to run MapReduce jobs. Values can be local, classic, or yarn. Required for Sqoop.

Set to: yarn

mapreduce.jobhistory.address

Location of the MapReduce JobHistory Server. The default port is 10020. Required for Sqoop.

Set to: <MapReduce JobHistory Server>:<port>

yarn.app.mapreduce.am.staging-dir

The HDFS staging directory used while submitting jobs.

Set to the staging directory path.

yarn-site.xml

Configure the following properties in the yarn-site.xml file:

yarn.application.classpath

Required for dynamic resource allocation.

"Add spark_shuffle.jar to the class path". The .jar file must contain the class "org.apache.network.yarn.YarnShuffleService."

yarn.nodemanager.resource.memory-mb

The maximum RAM available for each container. Set the maximum memory on the cluster to increase resource memory available to the Blaze engine.

Set to 16 GB if value is less than 16 GB.

yarn.nodemanager.resource.cpu-vcores

The number of virtual cores for each container. Required for Blaze engine resource allocation.

Set to 10 if the value is less than 10.

yarn.scheduler.minimum-allocation-mb

The minimum RAM available for each container. Required for Blaze engine resource allocation.

Set to 6 GB if the value is less than 6 GB.

yarn.nodemanager.vmem-check-enabled

Disables virtual memory limits for containers. Required for the Blaze and Spark engines.

Set to: FALSE

yarn.nodemanager.aux-services

Required for dynamic resource allocation for the Spark engine.

Add an entry for "spark_shuffle."

yarn.nodemanager.aux-services.spark_shuffle.class

Required for dynamic resource allocation for the Spark engine.

Set to: org.apache.spark.network.yarn.YarnShuffleService

yarn.resourcemanager.scheduler.class

Defines the YARN scheduler that the Data Integration Service uses to assign resources.

Set to: org.apache.hadoop.yarn.server.resourcemanager.scheduler

yarn.node-labels.enabled

Enables node labeling.

Set to: TRUE

yarn.node-labels.fs-store.root-dir

The HDFS location to update node label dynamically.

Set to: <hdfs://[Node name]:[Port]/[Path to store]/[Node labels]/>

Prepare the Archive File for Import from MapR

After you verify property values in the *-site.xml files, create a .zip or a .tar file that the Informatica administrator can use to import the cluster configuration into the domain.

Create an archive file that contains the following files from the cluster:

- core-site.xml
- hbase-site.xml. Required only if you access HBase sources and targets.
- hive-site.xml
- mapred-site.xml
- yarn-site.xml

Note: To import from MapR, the Informatica administrator must use an archive file.

Edit the hosts File for the Blaze Engine

To run the Blaze engine on every node in the cluster, verify that the /etc/hosts file on every node has entries for all other nodes.

Each node in the cluster requires an entry for the IP address and the fully qualified domain name (FQDN) of all other nodes. For example,

```
127.0.0.1 localhost node1.node.com
208.164.186.1 node1.node.com node1
208.164.186.2 node2.node.com node2
208.164.186.3 node3.node.com node3
```

Changes take effect after you restart the network.

Create a Cluster Configuration

After the Hadoop administrator prepares the cluster for import, the Informatica administrator must create a cluster configuration.

Perform this task in the following situations:
- You are integrating for the first time. - You upgraded from a version earlier than 10.2. - You upgraded from 10.2 and changed the distribution or distribution version.

A cluster configuration is an object in the domain that contains configuration information about the Hadoop cluster. The cluster configuration enables the Data Integration Service to push mapping logic to the Hadoop environment. Import configuration properties from the Hadoop cluster to create a cluster configuration.

The import process imports values from *-site.xml files into configuration sets based on the individual *-site.xml files. When you perform the import, the cluster configuration wizard can create Hadoop, HBase, HDFS, and Hive connection to access the Hadoop environment. If you choose to create the connections, the wizard also associates the cluster configuration with the connections.

Note: If you imported the cluster configuration when you installed Enterprise Data Lake with the Informatica domain, you can create the cluster configuration again or refresh the cluster configuration.

For more information about the cluster configuration, see the *Big Data Management Administrator Guide*.

Importing a Cluster Configuration from a File

You can import properties from an archive file to create a cluster configuration.

Before you import from the cluster, you must get the archive file from the Hadoop administrator.

1. From the **Connections** tab, click the **ClusterConfigurations** node in the Domain Navigator.
2. From the Actions menu, select **New** > **Cluster Configuration**.

   The **Cluster Configuration** wizard opens.
3. Configure the following properties:

Property	Description
Cluster configuration name	Name of the cluster configuration.
Description	Optional description of the cluster configuration.
Distribution type	The cluster Hadoop distribution type.

Property	Description
Distribution version	Version of the Hadoop distribution. Each distribution type has a default version. This is the latest version of the Hadoop distribution that Big Data Management supports. When the cluster version differs from the default version, the cluster configuration wizard populates the cluster configuration Hadoop distribution property with the most recent supported version relative to the cluster version. For example, suppose Informatica supports versions 5.10 and 5.13, and the cluster version is 5.12. In this case, the wizard populates the version with 5.10. You can edit the property to choose any supported version. Restart the Data Integration Service for the changes to take effect.
Method to import the cluster configuration	Choose **Import from file** to import properties from an archive file.
Create connections	Choose to create Hadoop, HDFS, Hive, and HBase connections. If you choose to create connections, the Cluster Configuration wizard associates the cluster configuration with each connection that it creates. If you do not choose to create connections, you must manually create them and associate the cluster configuration with them. **Important:** When the wizard creates the Hive connection, it populates the Metadata Connection String and the Data Access Connection String properties with the value from the hive.metastore.uris property. If the Hive metastore and HiveServer2 are running on different nodes, you must update the Metadata Connection String to point to the HiveServer2 host.

4. Click **Browse** to select a file. Select the file and click **Open**.
5. Click **Next** and verify the cluster configuration information on the summary page.

Configure the Hadoop Connection

Hadoop connections contain default values for properties such as cluster environment variables, cluster path variables, and advanced properties. Based on the cluster environment and the functionality that you use, you can add to the default values or change the default values of these properties.

For a list of Hadoop connection properties to configure, see “Configuring Hadoop Connection Properties” on page 161.

Download the JDBC Drivers for Sqoop Connectivity

To configure Sqoop connectivity for relational databases, you must download JDBC driver .jar files.

1. Download any Type 4 JDBC driver that the database vendor recommends for Sqoop connectivity.

   Note: The DataDirect JDBC drivers that Informatica ships are not licensed for Sqoop connectivity.

2. If you want to use MapR Connector for Teradata, perform the following steps:
   1. Download the `sqoop-connector-tdch-1.1-mapr-1707.jar` file from the following URL:

      https://repository.mapr.com/nexus/content/repositories/mapr-public/org/apache/sqoop/connector/sqoop-connector-tdch/1.1-mapr-1707/
   2. Download the MapR Connector for Teradata .jar file from the Teradata website.
   3. Download the `terajdbc4.jar` file and `tdgssconfig.jar` file from the following URL:

      http://downloads.teradata.com/download/connectivity/jdbc-driver
3. To optimize the Sqoop mapping performance on the Spark engine while writing data to an HDFS complex file target of the Parquet format, download the following .jar files:
   - `parquet-hadoop-bundle-1.6.0.jar` from https://mvnrepository.com/artifact/com.twitter/parquet-hadoop-bundle
   - `parquet-avro-1.6.0.jar` from https://mvnrepository.com/artifact/com.twitter/parquet-avro
   - `parquet-column-1.5.0.jar` from https://mvnrepository.com/artifact/org.apache.parquet/parquet-column
4. Copy all of the .jar files to the following directory on the machine where the Data Integration Service runs:

   ```
   <Informatica installation directory>\externaljdbcjars
   ```

   At run time, the Data Integration Service copies the .jar files to the Hadoop distribution cache so that the .jar files are accessible to all nodes in the cluster.

Create a Proxy Directory for MapR

If the Hadoop cluster runs on MapR, you must create a proxy directory for the user who will impersonate other users.

Verify the following requirements for the proxy user:

- Create a user or verify that a user exists on every Data Integration Service machine and on every node in the Hadoop cluster.
- Verify that the uid and the gid of the user matches in both environments.
- Verify that a directory exists for the user on the cluster. For example, `/opt/mapr/conf/proxy/<user name>`

Generate MapR Tickets

To run mappings on a MapR cluster that uses Kerberos or MapR Ticket authentication with information in Hive tables, generate a MapR ticket for the Data Integration Service user.

The Data Integration Service user requires an account on the MapR cluster and a MapR ticket on the application service machines that require access to MapR. When the MapR cluster uses both Kerberos and Ticket authentication, you generate a single ticket for the Data Integration Service user for both authentication systems.

After you generate and save MapR tickets, you perform additional steps to configure the Data Integration Service, the Metadata Access Service, and the Analyst Service to communicate with the MapR cluster.

Generate Tickets

After you create a MapR user account for the Data Integration Service user, name the ticket file using the following naming convention:

```
maprticket_<user name>
```

For example, for a user ID 1234, a MapR ticket file named maprticket_1234 is generated.

Save the ticket on the machines that host the Data Integration Service, the Metadata Access Service, and the Analyst Service. The Data Integration Service and the Analyst Service access the ticket at run time. The Metadata Access Service access the ticket for the Developer tool at design time.

By default, the services access the ticket in the /tmp directory. If you save the ticket to any other location, you must configure the MAPR_TICKETFILE_LOCATION environment variable in the service properties.

Configure the Data Integration Service

When the MapR cluster is secured with Kerberos or MapR Ticket authentication, edit Data Integration Service properties to enable communication between the Informatica domain and the cluster.

In the Administrator tool Domain Navigator, select the Data Integration Service to configure, and then select the **Processes** tab.

In the **Environment Variables** area, configure the following property to define the Kerberos authentication protocol:

Property	Value
JAVA_OPTS	`-Dhadoop.login=<MAPR_ECOSYSTEM_LOGIN_OPTS> -Dhttps.protocols=TLSv1.2` where <MAPR_ECOSYSTEM_LOGIN_OPTS> is the value of the MAPR_ECOSYSTEM_LOGIN_OPTS property in the file `/opt/mapr/conf/env.sh`.
MAPR_HOME	MapR client directory on the machine that runs the Data Integration Service. For example, `opt/mapr` Required if you want to fetch a MapR Streams data object.
MAPR_TICKETFILE_LOCATION	Required when the MapR cluster uses Kerberos or MapR Ticket authentication. Location of the MapR ticket file if you saved it to a directory other than /tmp. For example: `/export/home/username1/Keytabs_and_krb5conf/Tickets/project1/maprticket_30103`

Changes take effect when you restart the Data Integration Service.

Configure the Metadata Access Service

When the MapR cluster is secured with MapR Kerberos or ticketed authentication, edit Metadata Access Service properties to enable communication between the Developer tool and the cluster.

In the Administrator tool Domain Navigator, select the Metadata Access Service to configure, and then select the **Processes** tab.

In the **Environment Variables** area, configure the following property to define the Kerberos authentication protocol:

Property	Value
JAVA_OPTS	`-Dhadoop.login=<MAPR_ECOSYSTEM_LOGIN_OPTS> -Dhttps.protocols=TLSv1.2` where <MAPR_ECOSYSTEM_LOGIN_OPTS> is the value of the MAPR_ECOSYSTEM_LOGIN_OPTS property in the file `/opt/mapr/conf/env.sh`.
MAPR_TICKETFILE_LOCATION	Required when the MapR cluster uses Kerberos or MapR Ticket authentication. Location of the MapR ticket file if you saved it to a directory other than /tmp. For example, `/export/home/username1/Keytabs_and_krb5conf/Tickets/project1/maprticket_30103`

Changes take effect when you restart the Metadata Access Service.

Configure the Analyst Service

If you use the Analyst tool to profile data in Hive data objects, configure properties on the Analyst Service to enable communication between the Analyst tool and the cluster, including testing of the Hive connection.

In the Administrator tool Domain Navigator, select the Analyst Service to configure, then select the **Processes** tab.

In the **Environment Variables** area, configure the following property to define the Kerberos authentication protocol:

Property	Value
JAVA_OPTS	`-Dhadoop.login=hybrid -Dhttps.protocols=TLSv1.2`
MAPR_TICKETFILE_LOCATION	Required when the MapR cluster uses Kerberos or MapR Ticket authentication. Location of the MapR ticket file if you saved it to a directory other than /tmp. For example, `/export/home/username1/Keytabs_and_krb5conf/Tickets/project1/maprticket_30103`
LD_LIBRARY_PATH	The location of Hadoop libraries. For example, `<Informatica installation directory>/java/jre/lib:<Informatica installation directory>/services/shared/bin:<Informatica installation directory>/server/bin:<Informatica installation directory>/services/shared/hadoop/<MapR location>/lib/native/Linux-amd64-64`

Changes take effect when you restart the Analyst Service.

Test the Hive Connection

After you configure users for MapR Ticket or Kerberos authentication on MapR clusters, you can test the Hive connection.

To test the Hive connection, or perform a metadata fetch task, use the following format for the connection string if the cluster is Kerberos-enabled:

```
jdbc:hive2://<hostname>:10000/default;principal=<SPN>
```

For example,

```
jdbc:hive2://myServer2:10000/default;principal=mapr/myServer2@clustername
```

Note: When the mapping performs a metadata fetch of a complex file object, the user whose maprticket is present at %TEMP% on the Windows machine must have read permission on the HDFS directory to list the files inside it and perform the import action. The metadata fetch operation ignores privileges of the user who is listed in the HDFS connection definition.

Configure the Developer Tool

To access the Hadoop environment from the Developer tool, the mapping developers must perform tasks on each Developer tool machine.

Configure developerCore.ini

Edit `developerCore.ini` to successfully import local complex files available on the Developer tool machine.

When you import a complex file, such as Avro or Parquet, the imported object includes metadata associated with the distribution in the Hadoop environment. If the file resides on the Developer tool machine, the import process picks up the distribution information from the `developerCore.ini` file. You must edit the `developerCore.ini` file to point to the distribution directory on the Developer tool machine.

You can find `developerCore.ini` in the following directory: `<Informatica installation directory>\clients\DeveloperClient`

Add the following property:

```
-DINFA_HADOOP_DIST_DIR=hadoop\<distribution>_<version>
```

For example, `-DINFA_HADOOP_DIST_DIR=hadoop\mapr_5.2.0`

Complete Upgrade Tasks

If you upgraded the Informatica platform, you need to perform some additional tasks within the Informatica domain.

Based on the version that you upgraded from, perform the following tasks:

Configure the JDK home directory.

To run Sqoop mappings or process a Java transformation on the Spark engine, you must configure the JDK Home Directory in the Data Integration Service properties.

Configure the Hadoop connection.

The Hadoop connection contains additional properties. You need to manually update it to include customized configuration in the hadoopEnv.properties file from previous versions.

Replace connections.

If you chose the option to create connections when you ran the **Cluster Configuration** wizard, you need to replace connections in mappings with the new connections.

Complete connection upgrades.

If you did not create connections when you created the cluster configuration, you need to update the connections. The tasks to complete connection upgrade depends on the Informatica version that you upgraded.

Configure the JDK Home Directory

To leverage Sqoop or to process a Java transformation on the Spark engine, you must install the Java Development Kit (JDK) on the machine that runs the Data Integration Service. Then, you must configure the **JDK Home Directory** property for the Data Integration Service.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

Configure the following property under the Data Integration Service execution options in Informatica Administrator:

JDK Home Directory

Required to run Sqoop mappings or mass ingestion specifications that use a Sqoop connection on the Spark engine, or to process a Java transformation on the Spark engine.

The JDK version that the Data Integration Service uses must be compatible with the JRE version on the cluster.

Set the property to the JDK installation directory on the machine that runs the Data Integration Service. Changes take effect after you recycle the Data Integration Service.

Configure the Hadoop Connection

To use properties that you customized in the hadoopEnv.properties file, you must configure the Hadoop connection properties such as cluster environment variables, cluster path variables, and advanced properties.

Perform this task in the following situation:
- You upgraded Big Data Management from any previous version.

When you run the Informatica upgrade, the installer backs up the existing hadoopEnv.properties file. You can find the backup hadoopEnv.properties file in the following location:

```
<Previous Informatica installation directory>/services/shared/hadoop/<Hadoop distribution
name>_<version>/infaConf
```

Edit the Hadoop connection in the Administrator tool or the Developer tool to include any properties that you manually configured in the hadoopEnv.properties file. The Hadoop connection contains default values for properties such as cluster environment and path variables and advanced properties. You can update the default values to match the properties in the hadoopEnv.properties file.

Replace the Connections with New Connections

If you created connections when you imported the cluster configuration, you need to replace connections in mappings with the new connections.

Perform this task in the following situation:
- You upgraded from a version earlier than 10.2.

The method that you use to replace connections in mappings depends on the type of connection.

Hadoop connection

Run the following commands to replace the connections:

- `infacmd dis replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that are deployed in applications.
- `infacmd mrs replaceMappingHadoopRuntimeConnections`. Replaces connections associated with mappings that you run from the Developer tool.

For information about the infacmd commands, see the *Informatica Command Reference*.

Hive, HDFS, and HBase connections

You must replace the connections manually.

Complete Connection Upgrade

If *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

Perform the following tasks to update the connections:

Update changed properties

Review connections that you created in a previous release to update the values for connection properties. For example, if you added nodes to the cluster or if you updated the distribution version, you might need to verify host names, URIs, or port numbers for some of the properties.

Associate the cluster configuration

The Hadoop, Hive, HDFS, and HBase connections must be associated with a cluster configuration. Complete the following tasks:

1. Run `infacmd isp listConnections` to identify the connections that you need to upgrade. Use `-ct` to list connections of a particular type.
2. Run `infacmd isp UpdateConnection` to associate the cluster configuration with the connection. Use `-cn` to name the connection and `-o clusterConfigID` to associate the cluster configuration with the connection.

For more information about infacmd, see the *Informatica Command Reference*.

Replace Hive Run-time Connections with Hadoop Connections

Big Data Management requires a Hadoop connection to run mappings on the Hadoop cluster. If you used Hive connections to run mappings on the Hadoop cluster, you must generate Hadoop connections from the Hive connections.

The upgrade process generates a connection name for the Hadoop connection and replaces the connection name in the mappings. It does not create the physical connection object. When the upgrade is complete, you must run a command to generate the connection. You generate Hadoop connections from Hive connections that are configured to run mappings in the Hadoop environment.

Complete the following tasks to upgrade connections:

Generate Hadoop connections

You must generate Hadoop connections from Hive connections that are configured to run mappings in the Hadoop environment.

1. Run `infacmd isp generateHadoopConnectionFromHiveConnection` to generate a Hadoop connection from a Hive connection that is configured to run in the Hadoop environment. The command names the connection as follows: "Autogen_<Hive connection name>." If the connection name exceeds the 128 character limit, the command fails.
2. If the command fails, complete the following tasks:
    a. Rename the connection to meet the character limit and run the command again.
    b. Run `infacmd dis replaceMappingHadoopRuntimeConnections` to replace connections associated with mappings that are deployed in applications.
    c. Run `infacmd mrs replaceMappingHadoopRuntimeConnections` to replace connections associated with mappings that you run from the Developer tool.
3. If the Hive connection was parameterized, you must update the connection names in the parameter file. Verify that the Hive sources, Hive targets, and the Hive engine parameters are updated with the correct connection name.
4. If any properties changed in the cluster, such as host names, URIs, or port numbers, you must update the properties in the connections.

Associate the cluster configuration

The Hadoop, Hive, HDFS, and HBase connections must be associated with a cluster configuration. Complete the following tasks:

1. Run `infacmd isp listConnections` to identify the connections that you need to upgrade. Use `-ct` to list connections of a particular type.
2. Run `infacmd isp UpdateConnection` to associate the cluster configuration with the connection. Use `-cn` to name the connection and `-o clusterConfigID` to associate the cluster configuration with the connection.

For information about the infacmd commands, see the *Informatica Command Reference*.

APPENDIX A

Connections

This appendix includes the following topics:

Connections

Create a connection to access non-native environments, Hadoop and Databricks. If you access HBase, HDFS, or Hive sources or targets in the Hadoop environment, you must also create those connections. You can create the connections using the Developer tool, Administrator tool, and infacmd.

You can create the following types of connections:

Hadoop connection

Create a Hadoop connection to run mappings in the Hadoop environment.

HBase connection

Create an HBase connection to access HBase. The HBase connection is a NoSQL connection.

HDFS connection

Create an HDFS connection to read data from or write data to the HDFS file system on a Hadoop cluster.

Hive connection

Create a Hive connection to access Hive as a source or target. You can access Hive as a source if the mapping is enabled for the native or Hadoop environment. You can access Hive as a target if the mapping runs on the Blaze engine.

JDBC connection

Create a JDBC connection and configure Sqoop properties in the connection to import and export relational data through Sqoop.

Databricks connection

Create a Databricks connection to run mappings in the Databricks environment.

Note: For information about creating connections to other sources or targets such as social media web sites or Teradata, see the respective PowerExchange adapter user guide for information.

Cloud Provisioning Configuration

The cloud provisioning configuration establishes a relationship between the Create Cluster task and the cluster connection that the workflows use to run mapping tasks. The Create Cluster task must include a reference to the cloud provisioning configuration. In turn, the cloud provisioning configuration points to the cluster connection that you create for use by the cluster workflow.

The properties to populate depend on the Hadoop distribution you choose to build a cluster on. Choose one of the following connection types:

- AWS Cloud Provisioning. Connects to an Amazon EMR cluster on Amazon Web Services.
- Azure Cloud Provisioning. Connects to an HDInsight cluster on the Azure platform.
- Databricks Cloud Provisioning. Connects to a Databricks cluster on the Azure Databricks platform.

AWS Cloud Provisioning Configuration Properties

The properties in the AWS cloud provisioning configuration enable the Data Integration Service to contact and create resources on the AWS cloud platform.

General Properties

The following table describes cloud provisioning configuration general properties:

Property	Description
Name	Name of the cloud provisioning configuration.
ID	ID of the cloud provisioning configuration. Default: Same as the cloud provisioning configuration name.
Description.	Optional. Description of the cloud provisioning configuration.
AWS Access Key ID	Optional. ID of the AWS access key, which AWS uses to control REST or HTTP query protocol requests to AWS service APIs. If you do not specify a value, Informatica attempts to follow the Default Credential Provider Chain.
AWS Secret Access Key	Secret component of the AWS access key. Required if you specify the AWS Access Key ID.
Region	Region in which to create the cluster. This must be the region in which the VPC is running. Use AWS region values. For a list of acceptable values, see AWS documentation. **Note:** The region where you want to create the cluster can be different from the region in which the Informatica domain is installed.

Permissions

The following table describes cloud provisioning configuration permissions properties:

Property	Description
EMR Role	Name of the service role for the EMR cluster that you create. The role must have sufficient permissions to create a cluster, access S3 resources, and run jobs on the cluster. When the AWS administrator creates this role, they select the "EMR" role. This contains the default AmazonElasticMapReduceRole policy. You can edit the services in this policy.
EC2 Instance Profile	Name of the EC2 instance profile role that controls permissions on processes that run on the cluster. When the AWS administrator creates this role, they select the "EMR Role for EC2" role. This includes S3 access by default.
Auto Scaling Role	Required if you configure auto-scaling for the EMR cluster. This role is created when the AWS administrator configures auto-scaling on any cluster in the VPC. Default: When you leave this field blank, it is equivalent to setting the Auto Scaling role to "Proceed without role" when the AWS administrator creates a cluster in the AWS console.

EC2 Configuration

The following table describes cloud provisioning configuration EC2 configuration properties:

Property	Description
EC2 Key Pair	EC2 key pair to enable communication with the EMR cluster master node. Optional. This credential enables you to log into the cluster. Configure this property if you intend the cluster to be non-ephemeral.
EC2 Subnet	ID of the subnet on the VPC in which to create the cluster. Use the subnet ID of the EC2 instance where the cluster runs.
Master Security Group	Optional. ID of the security group for the cluster master node. Acts as a virtual firewall to control inbound and outbound traffic to cluster nodes. Security groups are created when the AWS administrator creates and configures a cluster in a VPC. In the AWS console, the property is equivalent to ElasticMapReduce-master. You can use existing security groups, or the AWS administrator might create dedicated security groups for the ephemeral cluster. If you do not specify a value, the cluster applies the default security group for the VPC.
Additional Master Security Groups	Optional. IDs of additional security groups to attach to the cluster master node. Use a comma-separated list of security group IDs.
Core and Task Security Group	Optional. ID of the security group for the cluster core and task nodes. When the AWS administrator creates and configures a cluster In the AWS console, the property is equivalent to the ElasticMapReduce-slave security group If you do not specify a value, the cluster applies the default security group for the VPC.
Additional Core and Task Security Groups	Optional. IDs of additional security groups to attach to cluster core and task nodes. Use a comma-separated list of security group IDs.
Service Access Security Group	EMR managed security group for service access. Required when you provision an EMR cluster in a private subnet.

Azure Cloud Provisioning Configuration Properties

The properties in the Azure cloud provisioning configuration enable the Data Integration Service to contact and create resources on the Azure cloud platform.

Authentication Details

The following table describes authentication properties to configure:

Property	Description
Name	Name of the cloud provisioning configuration.
ID	ID of the cloud provisioning configuration. Default: Same as the cloud provisioning configuration name.
Description	Optional. Description of the cloud provisioning configuration.

Property	Description
Subscription ID	ID of the Azure account to use in the cluster creation process.
Tenant ID	A GUID string associated with the Azure Active Directory.
Client ID	A GUID string that is the same as the Application ID associated with the Service Principal. The Service Principal must be assigned to a role that has permission to create resources in the subscription that you identified in the Subscription ID property.
Client Secret	An octet string that provides a key associated with the client ID.

Storage Account Details

Choose to configure access to one of the following storage types:

- Azure Data Lake Storage (ADLS). See Azure documentation.
- An Azure Storage Account, known as general or blob storage. See Azure documentation.

The following table describes the information you need to configure Azure Data Lake Storage (ADLS) with the HDInsight cluster:

Property	Description
Azure Data Lake Store Name	Name of the ADLS storage to access. The ADLS storage and the cluster to create must reside in the same region.
Data Lake Service Principal Client ID	A credential that enables programmatic access to ADLS storage. Enables the Informatica domain to communicate with ADLS and run commands and mappings on the HDInsight cluster. The service principal is an Azure user that meets the following requirements: - Permissions to access required directories in ADLS storage. - Certificate-based authentication for ADLS storage. - Key-based authentication for ADLS storage.
Data Lake Service Principal Certificate Contents	The Base64 encoded text of the public certificate used with the service principal. Leave this property blank when you create the cloud provisioning configuration. After you save the cloud provisioning configuration, log in to the VM where the Informatica domain is installed and run infacmd ccps updateADLSCertificate to populate this property.
Data Lake Service Principal Certificate Password	Private key for the service principal. This private key must be associated with the service principal certificate.
Data Lake Service Principal Client Secret	An octet string that provides a key associated with the service principal.
Data Lake Service Principal OAUTH Token Endpoint	Endpoint for OAUTH token based authentication.

The following table describes the information you need to configure Azure General Storage, also known as blob storage, with the HDInsight cluster:

Property	Description
Azure Storage Account Name	Name of the storage account to access. Get the value from the Storage Accounts node in the Azure web console. The storage and the cluster to create must reside in the same region.
Azure Storage Account Key	A key to authenticate access to the storage account. To get the value from the Azure web console, select the storage account, then Access Keys. The console displays the account keys.

Cluster Deployment Details

The following table describes the cluster deployment properties that you configure:

Property	Description
Resource Group	Resource group in which to create the cluster. A resource group is a logical set of Azure resources.
Virtual Network Resource Group	Optional. Resource group to which the virtual network belongs. If you do not specify a resource group, the Data Integration Service assumes that the virtual network is a member of the same resource group as the cluster.
Virtual Network	Name of the virtual network or vnet where you want to create the cluster. Specify a vnet that resides in the resource group that you specified in the Virtual Network Resource Group property. The vnet must be in the same region as the region in which to create the cluster.
Subnet Name	Subnet in which to create the cluster. The subnet must be a part of the vnet that you designated in the previous property. Each vnet can have one or more subnets. The Azure administrator can choose an existing subnet or create one for the cluster.

External Hive Metastore Details

You can specify the properties to enable the cluster to connect to a Hive metastore database that is external to the cluster.

You can use an external relational database like MySQL or Amazon RDS as the Hive metastore database. The external database must be on the same cloud platform as the cluster to create.

If you do not specify an existing external database in this dialog box, the cluster creates its own database on the cluster. This database is terminated when the cluster is terminated.

The following table describes the Hive metastore database properties that you configure:

Property	Description
Database Name	Name of the Hive metastore database.
Database Server Name	Server on which the database resides. **Note:** The database server name on the Azure web console commonly includes the suffix `database.windows.net`. For example: `server123xyz.database.windows.net`. You can specify the database server name without the suffix and Informatica will automatically append the suffix. For example, you can specify `server123xyz`.
Database User Name	User name of the account for the domain to use to access the database.
Database Password	Password for the user account.

Databricks Cloud Provisioning Configuration Properties

The properties in the Databricks cloud provisioning configuration enable the Data Integration Service to contact and create resources on the Databricks cloud platform.

The following table describes the Databricks cloud provisioning configuration properties:

Property	Description
Name	Name of the cloud provisioning configuration.
ID	The cluster ID of the Databricks cluster.
Description	Optional description of the cloud provisioning configuration.
Databricks domain	Domain name of the Databricks deployment.
Databricks token ID	The token ID created within Databricks required for authentication. **Note:** If the token has an expiration date, verify that you get a new token from the Databricks administrator before it expires.

Amazon Redshift Connection Properties

When you set up an Amazon Redshift connection, you must configure the connection properties.

The following table describes the Amazon Redshift connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select Amazon Redshift in the Database.

The **Details** tab contains the connection attributes of the Amazon Redshift connection. The following table describes the connection attributes:

Property	Description
Username	User name of the Amazon Redshift account.
Password	Password for the Amazon Redshift account.
Schema	Optional. Amazon Redshift schema name. Do not specify the schema name if you want to use multiple schema. The Data Object wizard displays all the user-defined schemas available for the Amazon Redshift objects. Default is public.
AWS Access Key ID	Amazon S3 bucket access key ID. **Note:** Required if you do not use AWS Identity and Access Management (IAM) authentication.
AWS Secret Access Key	Amazon S3 bucket secret access key ID. **Note:** Required if you do not use AWS Identity and Access Management (IAM) authentication.
Master Symmetric Key	Optional. Provide a 256-bit AES encryption key in the Base64 format when you enable client-side encryption. You can generate a key using a third-party tool. If you specify a value, ensure that you specify the encryption type as client side encryption in the advanced target properties.

Property	Description
Customer Master Key ID	Optional. Specify the customer master key ID or alias name generated by AWS Key Management Service (AWS KMS). You must generate the customer master key corresponding to the region where Amazon S3 bucket resides. You can specify any of the following values: **Customer generated customer master key** Enables client-side or server-side encryption. **Default customer master key** Enables client-side or server-side encryption. Only the administrator user of the account can use the default customer master key ID to enable client-side encryption. **Note:** You can use customer master key ID when you run a mapping in the native environment or on the Spark engine.
Cluster Node Type	Node type of the Amazon Redshift cluster. You can select the following options: - ds1.xlarge - ds1.8xlarge - dc1.large - dc1.8xlarge - ds2.xlarge - ds2.8xlarge For more information about nodes in the cluster, see the Amazon Redshift documentation.
Number of Nodes in Cluster	Number of nodes in the Amazon Redshift cluster. For more information about nodes in the cluster, see the Amazon Redshift documentation.
JDBC URL	Amazon Redshift connection URL.

Note: If you upgrade the mappings created in earlier versions, you must select the relevant schema in the connection property. Else, the mappings fail when you run them on current version.

Amazon S3 Connection Properties

When you set up an Amazon S3 connection, you must configure the connection properties.

The following table describes the Amazon S3 connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.

Property	Description
Location	The domain where you want to create the connection.
Type	The Amazon S3 connection type.
Access Key	The access key ID for access to Amazon account resources. **Note:** Required if you do not use AWS Identity and Access Management (IAM) authentication.
Secret Key	The secret access key for access to Amazon account resources. The secret key is associated with the access key and uniquely identifies the account. **Note:** Required if you do not use AWS Identity and Access Management (IAM) authentication.
Folder Path	The complete path to Amazon S3 objects. The path must include the bucket name and any folder name. Do not use a slash at the end of the folder path. For example, `<bucket name>/<my folder name>`.
Master Symmetric Key	Optional. Provide a 256-bit AES encryption key in the Base64 format when you enable client-side encryption. You can generate a master symmetric key using a third-party tool.
Customer Master Key ID	Optional. Specify the customer master key ID or alias name generated by AWS Key Management Service (AWS KMS). You must generate the customer master key for the same region where Amazon S3 bucket reside. You can specify any of the following values: **Customer generated customer master key** Enables client-side or server-side encryption. **Default customer master key** Enables client-side or server-side encryption. Only the administrator user of the account can use the default customer master key ID to enable client-side encryption. **Note:** Applicable when you run a mapping in the native environment or on the Spark engine.
Region Name	Select the AWS region in which the bucket you want to access resides. Select one of the following regions: - Asia Pacific (Mumbai) - Asia Pacific (Seoul) - Asia Pacific (Singapore) - Asia Pacific (Sydney) - Asia Pacific (Tokyo) - AWS GovCloud (US) - Canada (Central) - China (Beijing) - China (Ningxia) - EU (Ireland) - EU (Frankfurt) - EU (London) - EU (Paris) - South America (Sao Paulo) - US East (Ohio) - US East (N. Virginia) - US West (N. California) - US West (Oregon) Default is US East (N. Virginia).

Cassandra Connection Properties

When you set up a Cassandra connection, you must configure the connection properties.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Cassandra connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select **Cassandra**.
Host Name	Host name or IP address of the Cassandra server.
Port	Cassandra server port number. Default is 9042.
User Name	User name to access the Cassandra server.
Password	Password corresponding to the user name to access the Cassandra server.
Default Keyspace	Name of the Cassandra keyspace to use by default.
SQL Identifier Character	Type of character that the database uses to enclose delimited identifiers in SQL or CQL queries. The available characters depend on the database type. Select **None** if the database uses regular identifiers. When the Data Integration Service generates SQL or CQL queries, the service does not place delimited characters around any identifiers. Select a character if the database uses delimited identifiers. When the Data Integration Service generates SQL or CQL queries, the service encloses delimited identifiers within this character.
Additional Connection Properties	Enter one or more JDBC connection parameters in the following format: `<param1>=<value>;<param2>=<value>;<param3>=<value>` PowerExchange for Cassandra JDBC supports the following JDBC connection parameters: - BinaryColumnLength - DecimalColumnScale - EnableCaseSensitive - EnableNullInsert - EnablePaging - RowsPerPage - StringColumnLength - VTTableNameSeparator

Property	Description
SSL Mode	Not applicable for PowerExchange for Cassandra JDBC. Select **disabled**.
SSL Truststore Path	Not applicable for PowerExchange for Cassandra JDBC.
SSL Truststore Password	Not applicable for PowerExchange for Cassandra JDBC.
SSL Keystore Path	Not applicable for PowerExchange for Cassandra JDBC.
SSL Keystore Password	Not applicable for PowerExchange for Cassandra JDBC.

Databricks Connection Properties

Use the Databricks connection to run mappings on a Databricks cluster.

A Databricks connection is a cluster type connection. You can create and manage a Databricks connection in the Administrator tool or the Developer tool. You can use infacmd to create a Databricks connection. Configure properties in the Databricks connection to enable communication between the Data Integration Service and the Databricks cluster.

The following table describes the general connection properties for the Databricks connection:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Connection Type	Choose Databricks.
Cluster Configuration	Name of the cluster configuration associated with the Databricks environment. Required if you do not configure the cloud provisioning configuration.
Cloud Provisioning Configuration	Name of the cloud provisioning configuration associated with a Databricks cloud platform. Required if you do not configure the cluster configuration.

Property	Description
Staging Directory	The directory where the Databricks Spark engine stages run-time files. If you specify a directory that does not exist, the Data Integration Service creates it at run time. If you do not provide a directory path, the run-time staging files are written to */<cluster staging directory>/DATABRICKS*.
Advanced Properties	List of advanced properties that are unique to the Databricks environment. You can configure run-time properties for the Databricks environment in the Data Integration Service and in the Databricks connection. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Databricks connection. The Data Integration Service processes property overrides based on the following priorities: 1. Databricks connection advanced properties 2. Data Integration Service custom properties **Note:** Informatica does not recommend changing these property values before you consult with third-party documentation, Informatica documentation, or Informatica Global Customer Support. If you change a value without knowledge of the property, you might experience performance degradation or other unexpected results.

Advanced Properties

Configure the following properties in the **Advanced Properties** of the Databricks configuration section:

infaspark.json.parser.mode

Specifies the parser how to handle corrupt JSON records. You can set the value to one of the following modes:

- DROPMALFORMED. The parser ignores all corrupted records. Default mode.
- PERMISSIVE. The parser accepts non-standard fields as nulls in corrupted records.
- FAILFAST. The parser generates an exception when it encounters a corrupted record and the Spark application goes down.

infaspark.json.parser.multiLine

Specifies whether the parser can read a multiline record in a JSON file. You can set the value to true or false. Default is false. Applies only to non-native distributions that use Spark version 2.2.x and above.

infaspark.flatfile.writer.nullValue

When the Databricks Spark engine writes to a target, it converts null values to empty strings (" "). For example, 12, AB,"",23p09udj.

The Databricks Spark engine can write the empty strings to string columns, but when it tries to write an empty string to a non-string column, the mapping fails with a type mismatch.

To allow the Databricks Spark engine to convert the empty strings back to null values and write to the target, configure the following advanced property in the Databricks Spark connection:

```
infaspark.flatfile.writer.nullValue=true
```

Google Analytics Connection Properties

When you set up a Google Analytics connection, you must configure the connection properties.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Google Analytics connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select **Google Analytics**.
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
APIVersion	API that PowerExchange for Google Analytics uses to read from Google Analytics reports. Select **Core Reporting API v3**. **Note:** PowerExchange for Google Analytics does not support Analytics Reporting API v4.

Google BigQuery Connection Properties

When you set up a Google BigQuery connection, you must configure the connection properties.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Google BigQuery connection properties:

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account in Google BigQuery.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account in Google BigQuery.

Property	Description
Connection mode	The mode that you want to use to read data from or write data to Google BigQuery. Select one of the following connection modes: - Simple. Flattens each field within the Record data type field as a separate field in the mapping. - Hybrid. Displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the mapping. - Complex. Displays all the columns in the Google BigQuery table as a single field of the String data type in the mapping. Default is Simple.
Schema Definition File Path	Specifies a directory on the client machine where the PowerCenter Integration ServiceData Integration Service must create a JSON file with the sample schema of the Google BigQuery table. The JSON file name is the same as the Google BigQuery table name. Alternatively, you can specify a storage path in Google Cloud Storage where the PowerCenter Integration ServiceData Integration Service must create a JSON file with the sample schema of the Google BigQuery table. You can download the JSON file from the specified storage path in Google Cloud Storage to a local machine.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account in Google BigQuery. If you have created multiple projects with the same service account, enter the ID of the project that contains the dataset that you want to connect to.
Storage Path	This property applies when you read or write large volumes of data. Path in Google Cloud Storage where the PowerCenter Integration ServiceData Integration Service creates a local stage file to store the data temporarily. You can either enter the bucket name or the bucket name and folder name. For example, enter `gs://<bucket_name>` or `gs://<bucket_name>/<folder_name>`

Google Cloud Spanner Connection Properties

When you set up a Google Cloud Spanner connection, you must configure the connection properties.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Google Cloud Spanner connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.

Property	Description
Location	The domain where you want to create the connection.
Type	The connection type. Select Google Cloud Spanner.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the bucket that you want to connect to.
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
Instance ID	Name of the instance that you created in Google Cloud Spanner.

Google Cloud Storage Connection Properties

When you set up a Google Cloud Storage connection, you must configure the connection properties.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Google Cloud Storage connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select **Google Cloud Storage**.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the bucket that you want to connect to.

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.

Hadoop Connection Properties

Use the Hadoop connection to configure mappings to run on a Hadoop cluster. A Hadoop connection is a cluster type connection. You can create and manage a Hadoop connection in the Administrator tool or the Developer tool. You can use infacmd to create a Hadoop connection. Hadoop connection properties are case sensitive unless otherwise noted.

Hadoop Cluster Properties

Configure properties in the Hadoop connection to enable communication between the Data Integration Service and the Hadoop cluster.

The following table describes the general connection properties for the Hadoop connection:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. Enter a string that you can use to identify the connection. The description cannot exceed 4,000 characters.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment. Required if you do not configure the Cloud Provisioning Configuration.
Cloud Provisioning Configuration	Name of the cloud provisioning configuration associated with a cloud platform such as Amazon AWS or Microsoft Azure. Required if you do not configure the Cluster Configuration.

Property	Description
Cluster Environment Variables*	Environment variables that the Hadoop cluster uses. For example, the variable ORACLE_HOME represents the directory where the Oracle database client software is installed. You can configure run-time properties for the Hadoop environment in the Data Integration Service, the Hadoop connection, and in the mapping. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Hadoop connection or in the mapping. The Data Integration Service processes property overrides based on the following priorities: 1. Mapping custom properties set using infacmd ms runMapping with the `-cp` option 2. Mapping run-time properties for the Hadoop environment 3. Hadoop connection advanced properties for run-time engines 4. Hadoop connection advanced general properties, environment variables, and classpaths 5. Data Integration Service custom properties
Cluster Library Path*	The path for shared libraries on the cluster. The $DEFAULT_CLUSTER_LIBRARY_PATH variable contains a list of default directories.
Cluster Classpath*	The classpath to access the Hadoop jar files and the required libraries. The $DEFAULT_CLUSTER_CLASSPATH variable contains a list of paths to the default jar files and libraries. You can configure run-time properties for the Hadoop environment in the Data Integration Service, the Hadoop connection, and in the mapping. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Hadoop connection or in the mapping. The Data Integration Service processes property overrides based on the following priorities: 1. Mapping custom properties set using infacmd ms runMapping with the `-cp` option 2. Mapping run-time properties for the Hadoop environment 3. Hadoop connection advanced properties for run-time engines 4. Hadoop connection advanced general properties, environment variables, and classpaths 5. Data Integration Service custom properties
Cluster Executable Path*	The path for executable files on the cluster. The $DEFAULT_CLUSTER_EXEC_PATH variable contains a list of paths to the default executable files.
* Informatica does not recommend changing these property values before you consult with third-party documentation, Informatica documentation, or Informatica Global Customer Support. If you change a value without knowledge of the property, you might experience performance degradation or other unexpected results.	

Common Properties

The following table describes the common connection properties that you configure for the Hadoop connection:

Property	Description
Impersonation User Name	Required if the Hadoop cluster uses Kerberos authentication. Hadoop impersonation user. The user name that the Data Integration Service impersonates to run mappings in the Hadoop environment. The Data Integration Service runs mappings based on the user that is configured. Refer the following order to determine which user the Data Integration Services uses to run mappings: 1. Operating system profile user. The mapping runs with the operating system profile user if the profile user is configured. If there is no operating system profile user, the mapping runs with the Hadoop impersonation user. 2. Hadoop impersonation user. The mapping runs with the Hadoop impersonation user if the operating system profile user is not configured. If the Hadoop impersonation user is not configured, the Data Integration Service runs mappings with the Data Integration Service user. 3. Informatica services user. The mapping runs with the operating user that starts the Informatica daemon if the operating system profile user and the Hadoop impersonation user are not configured.
Temporary Table Compression Codec	Hadoop compression library for a compression codec class name. **Note:** The Spark engine does not support compression settings for temporary tables. When you run mappings on the Spark engine, the Spark engine stores temporary tables in an uncompressed file format.
Codec Class Name	Codec class name that enables data compression and improves performance on temporary staging tables.
Hive Staging Database Name	Namespace for Hive staging tables. Use the name `default` for tables that do not have a specified database name. If you do not configure a namespace, the Data Integration Service uses the Hive database name in the Hive target connection to create staging tables. When you run a mapping in the native environment to write data to Hive, you must configure the Hive staging database name in the Hive connection. The Data Integration Service ignores the value you configure in the Hadoop connection.
Advanced Properties	List of advanced properties that are unique to the Hadoop environment. The properties are common to the Blaze and Spark engines. The advanced properties include a list of default properties. You can configure run-time properties for the Hadoop environment in the Data Integration Service, the Hadoop connection, and in the mapping. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Hadoop connection or in the mapping. The Data Integration Service processes property overrides based on the following priorities: 1. Mapping custom properties set using infacmd ms runMapping with the `-cp` option 2. Mapping run-time properties for the Hadoop environment 3. Hadoop connection advanced properties for run-time engines 4. Hadoop connection advanced general properties, environment variables, and classpaths 5. Data Integration Service custom properties **Note:** Informatica does not recommend changing these property values before you consult with third-party documentation, Informatica documentation, or Informatica Global Customer Support. If you change a value without knowledge of the property, you might experience performance degradation or other unexpected results.

Reject Directory Properties

The following table describes the connection properties that you configure to the Hadoop Reject Directory.

Property	Description
Write Reject Files to Hadoop	If you use the Blaze engine to run mappings, select the check box to specify a location to move reject files. If checked, the Data Integration Service moves the reject files to the HDFS location listed in the property, Reject File Directory. By default, the Data Integration Service stores the reject files based on the RejectDir system parameter.
Reject File Directory	The directory for Hadoop mapping files on HDFS when you run mappings.

Hive Pushdown Configuration

Note: Effective in version 10.2.2, Informatica dropped support for the Hive engine. Do not configure the pushdown properties related to the Hive engine.

Blaze Configuration

The following table describes the connection properties that you configure for the Blaze engine:

Property	Description
Blaze Staging Directory	The HDFS file path of the directory that the Blaze engine uses to store temporary files. Verify that the directory exists. The YARN user, Blaze engine user, and mapping impersonation user must have write permission on this directory. Default is `/blaze/workdir`. If you clear this property, the staging files are written to the Hadoop staging directory `/tmp/blaze_<user name>`.
Blaze User Name	The owner of the Blaze service and Blaze service logs. When the Hadoop cluster uses Kerberos authentication, the default user is the Data Integration Service SPN user. When the Hadoop cluster does not use Kerberos authentication and the Blaze user is not configured, the default user is the Data Integration Service user.
Minimum Port	The minimum value for the port number range for the Blaze engine. Default is 12300.
Maximum Port	The maximum value for the port number range for the Blaze engine. Default is 12600.
YARN Queue Name	The YARN scheduler queue name used by the Blaze engine that specifies available resources on a cluster.
Blaze Job Monitor Address	The host name and port number for the Blaze Job Monitor. Use the following format: `<hostname>:<port>` Where - <hostname> is the host name or IP address of the Blaze Job Monitor server. - <port> is the port on which the Blaze Job Monitor listens for remote procedure calls (RPC). For example, enter: `myhostname:9080`

Property	Description
Blaze YARN Node Label	Node label that determines the node on the Hadoop cluster where the Blaze engine runs. If you do not specify a node label, the Blaze engine runs on the nodes in the default partition. If the Hadoop cluster supports logical operators for node labels, you can specify a list of node labels. To list the node labels, use the operators `&&` (AND), `||` (OR), and `!` (NOT).
Advanced Properties	List of advanced properties that are unique to the Blaze engine. The advanced properties include a list of default properties. You can configure run-time properties for the Hadoop environment in the Data Integration Service, the Hadoop connection, and in the mapping. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Hadoop connection or in the mapping. The Data Integration Service processes property overrides based on the following priorities: 1. Mapping custom properties set using infacmd ms runMapping with the `-cp` option 2. Mapping run-time properties for the Hadoop environment 3. Hadoop connection advanced properties for run-time engines 4. Hadoop connection advanced general properties, environment variables, and classpaths 5. Data Integration Service custom properties **Note:** Informatica does not recommend changing these property values before you consult with third-party documentation, Informatica documentation, or Informatica Global Customer Support. If you change a value without knowledge of the property, you might experience performance degradation or other unexpected results.

Spark Configuration

The following table describes the connection properties that you configure for the Spark engine:

Property	Description
Spark Staging Directory	The HDFS file path of the directory that the Spark engine uses to store temporary files for running jobs. The YARN user, Data Integration Service user, and mapping impersonation user must have write permission on this directory. If you do not specify a file path, by default, the temporary files are written to the Hadoop staging directory `/tmp/SPARK_<user name>`. When you run Sqoop jobs on the Spark engine, the Data Integration Service creates a Sqoop staging directory within the Spark staging directory to store temporary files: `<Spark staging directory>/sqoop_staging`
Spark Event Log Directory	Optional. The HDFS file path of the directory that the Spark engine uses to log events.

Property	Description
YARN Queue Name	The YARN scheduler queue name used by the Spark engine that specifies available resources on a cluster. The name is case sensitive.
Advanced Properties	List of advanced properties that are unique to the Spark engine. The advanced properties include a list of default properties. You can configure run-time properties for the Hadoop environment in the Data Integration Service, the Hadoop connection, and in the mapping. You can override a property configured at a high level by setting the value at a lower level. For example, if you configure a property in the Data Integration Service custom properties, you can override it in the Hadoop connection or in the mapping. The Data Integration Service processes property overrides based on the following priorities: 1. Mapping custom properties set using infacmd ms runMapping with the `-cp` option 2. Mapping run-time properties for the Hadoop environment 3. Hadoop connection advanced properties for run-time engines 4. Hadoop connection advanced general properties, environment variables, and classpaths 5. Data Integration Service custom properties **Note:** Informatica does not recommend changing these property values before you consult with third-party documentation, Informatica documentation, or Informatica Global Customer Support. If you change a value without knowledge of the property, you might experience performance degradation or other unexpected results.

HDFS Connection Properties

Use a Hadoop File System (HDFS) connection to access data in the Hadoop cluster. The HDFS connection is a file system type connection. You can create and manage an HDFS connection in the Administrator tool, Analyst tool, or the Developer tool. HDFS connection properties are case sensitive unless otherwise noted.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes HDFS connection properties:

Property	Description
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 765 characters.
Location	The domain where you want to create the connection. Not valid for the Analyst tool.
Type	The connection type. Default is Hadoop File System.

Property	Description
User Name	User name to access HDFS.
NameNode URI	The URI to access the storage system. You can find the value for `fs.defaultFS` in the `core-site.xml` configuration set of the cluster configuration. **Note:** If you create connections when you import the cluster configuration, the NameNode URI property is populated by default, and it is updated each time you refresh the cluster configuration. If you manually set this property or override the value, the refresh operation does not update this property.

Accessing Multiple Storage Types

Use the NameNode URI property in the connection parameters to connect to various storage types. The following table lists the storage type and the NameNode URI format for the storage type:

Storage	NameNode URI Format
HDFS	`hdfs://<namenode>:<port>` where: - `<namenode>` is the host name or IP address of the NameNode. - `<port>` is the port that the NameNode listens for remote procedure calls (RPC). `hdfs://<nameservice>` in case of NameNode high availability.
MapR-FS	`maprfs:///`

WASB in HDInsight	`wasb://<container_name>@<account_name>.blob.core.windows.net/<path>`
	where:
	- `<container_name>` identifies a specific Azure Storage Blob container.
	Note: `<container_name>` is optional.
	- `<account_name>` identifies the Azure Storage Blob object.
	Example:
	`wasb://infabdmoffering1storage.blob.core.windows.net/infabdmoffering1cluster/mr-history`

ADLS in HDInsight	`adl://home`

When you create a cluster configuration from an Azure HDInsight cluster, the cluster configuration uses either ADLS or WASB as the primary storage. You cannot create a cluster configuration with ADLS or WASB as the secondary storage. You can edit the NameNode URI property in the HDFS connection to connect to a local HDFS location.

HBase Connection Properties

Use an HBase connection to access HBase. The HBase connection is a NoSQL connection. You can create and manage an HBase connection in the Administrator tool or the Developer tool. HBase connection properties are case sensitive unless otherwise noted.

The following table describes HBase connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select HBase.
Database Type	Type of database that you want to connect to. Select **HBase** to create a connection for an HBase table.

HBase Connection Properties for MapR-DB

Use an HBase connection to connect to a MapR-DB table. The HBase connection is a NoSQL connection. You can create and manage an HBase connection in the Administrator tool or the Developer tool. HBase connection properties are case sensitive unless otherwise noted.

The following table describes the HBase connection properties for MapR-DB:

Property	Description
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.

Property	Description
Description	Description of the connection. The description cannot exceed 4,000 characters.
Location	Domain where you want to create the connection.
Type	Connection type. Select **HBase**.
Database Type	Type of database that you want to connect to. Select **MapR-DB** to create a connection for a MapR-DB table.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.
MapR-DB Database Path	Database path that contains the MapR-DB table that you want to connect to. Enter a valid MapR cluster path. When you create an HBase data object for MapR-DB, you can browse only tables that exist in the MapR-DB path that you specify in the **Database Path** field. You cannot access tables that are available in sub-directories in the specified path. For example, if you specify the path as `/user/customers/`, you can access the tables in the `customers` directory. However, if the `customers` directory contains a sub-directory named `regions`, you cannot access the tables in the following directory: `/user/customers/regions`

Hive Connection Properties

Use the Hive connection to access Hive data. A Hive connection is a database type connection. You can create and manage a Hive connection in the Administrator tool, Analyst tool, or the Developer tool. Hive connection properties are case sensitive unless otherwise noted.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes Hive connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4000 characters.

Property	Description
Location	The domain where you want to create the connection. Not valid for the Analyst tool.
Type	The connection type. Select Hive.
LDAP username	LDAP user name of the user that the Data Integration Service impersonates to run mappings on a Hadoop cluster. The user name depends on the JDBC connection string that you specify in the Metadata Connection String or Data Access Connection String for the native environment. If the Hadoop cluster uses Kerberos authentication, the principal name for the JDBC connection string and the user name must be the same. Otherwise, the user name depends on the behavior of the JDBC driver. With Hive JDBC driver, you can specify a user name in many ways and the user name can become a part of the JDBC URL. If the Hadoop cluster does not use Kerberos authentication, the user name depends on the behavior of the JDBC driver. If you do not specify a user name, the Hadoop cluster authenticates jobs based on the following criteria: - The Hadoop cluster does not use Kerberos authentication. It authenticates jobs based on the operating system profile user name of the machine that runs the Data Integration Service. - The Hadoop cluster uses Kerberos authentication. It authenticates jobs based on the SPN of the Data Integration Service. LDAP username will be ignored.
Password	Password for the LDAP username.
Environment SQL	SQL commands to set the Hadoop environment. In native environment type, the Data Integration Service executes the environment SQL each time it creates a connection to a Hive metastore. If you use the Hive connection to run profiles on a Hadoop cluster, the Data Integration Service executes the environment SQL at the beginning of each Hive session. The following rules and guidelines apply to the usage of environment SQL in both connection modes: - Use the environment SQL to specify Hive queries. - Use the environment SQL to set the classpath for Hive user-defined functions and then use environment SQL or PreSQL to specify the Hive user-defined functions. You cannot use PreSQL in the data object properties to specify the classpath. If you use Hive user-defined functions, you must copy the .jar files to the following directory: `<Informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>/extras/hive-auxjars` - You can use environment SQL to define Hadoop or Hive parameters that you want to use in the PreSQL commands or in custom queries. - If you use multiple values for the Environment SQL property, ensure that there is no space between the values.
SQL Identifier Character	The type of character used to identify special characters and reserved SQL keywords, such as WHERE. The Data Integration Service places the selected character around special characters and reserved SQL keywords. The Data Integration Service also uses this character for the **Support mixed-case identifiers** property.

Properties to Access Hive as Source or Target

The following table describes the connection properties that you configure to access Hive as a source or target:

Property	Description
JDBC Driver Class Name	Name of the Hive JDBC driver class. If you leave this option blank, the Developer tool uses the default Apache Hive JDBC driver shipped with the distribution. If the default Apache Hive JDBC driver does not fit your requirements, you can override the Apache Hive JDBC driver with a third-party Hive JDBC driver by specifying the driver class name.
Metadata Connection String	The JDBC connection URI used to access the metadata from the Hadoop server. You can use PowerExchange for Hive to communicate with a HiveServer service or HiveServer2 service. To connect to HiveServer, specify the connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database name to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer2, use the connection string format that Apache Hive implements for that specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation. For user impersonation, you must add `hive.server2.proxy.user=<xyz>` to the JDBC connection URI. If you do not configure user impersonation, the current user's credentials are used connect to the HiveServer2. If the Hadoop cluster uses SSL or TLS authentication, you must add `ssl=true` to the JDBC connection URI. For example: `jdbc:hive2://<hostname>:<port>/<db>;ssl=true` If you use self-signed certificate for SSL or TLS authentication, ensure that the certificate file is available on the client machine and the Data Integration Service machine. For more information, see the *Informatica Big Data Management Integration Guide*.
Bypass Hive JDBC Server	JDBC driver mode. Select the check box to use the embedded JDBC driver mode. To use the JDBC embedded mode, perform the following tasks: - Verify that Hive client and Informatica services are installed on the same machine. - Configure the Hive connection properties to run mappings on a Hadoop cluster. If you choose the non-embedded mode, you must configure the Data Access Connection String. Informatica recommends that you use the JDBC embedded mode.
Fine Grained Authorization	When you select the option to observe fine grained authorization in a Hive source, the mapping observes the following: - Row and column level restrictions. Applies to Hadoop clusters where Sentry or Ranger security modes are enabled. - Data masking rules. Applies to masking rules set on columns containing sensitive data by Dynamic Data Masking. If you do not select the option, the Blaze and Spark engines ignore the restrictions and masking rules, and results include restricted or sensitive data.

Property	Description
Data Access Connection String	The connection string to access data from the Hadoop data store. To connect to HiveServer, specify the non-embedded JDBC mode connection string in the following format: `jdbc:hive2://<hostname>:<port>/<db>` Where - <hostname> is name or IP address of the machine on which HiveServer2 runs. - <port> is the port number on which HiveServer2 listens. - <db> is the database to which you want to connect. If you do not provide the database name, the Data Integration Service uses the default database details. To connect to HiveServer2, use the connection string format that Apache Hive implements for the specific Hadoop Distribution. For more information about Apache Hive connection string formats, see the Apache Hive documentation. For user impersonation, you must add `hive.server2.proxy.user=<xyz>` to the JDBC connection URI. If you do not configure user impersonation, the current user's credentials are used connect to the HiveServer2. If the Hadoop cluster uses SSL or TLS authentication, you must add `ssl=true` to the JDBC connection URI. For example: `jdbc:hive2://<hostname>:<port>/<db>;ssl=true` If you use self-signed certificate for SSL or TLS authentication, ensure that the certificate file is available on the client machine and the Data Integration Service machine. For more information, see the *Informatica Big Data Management Integration Guide*.
Hive Staging Directory on HDFS	HDFS directory for Hive staging tables. You must grant execute permission to the Hadoop impersonation user and the mapping impersonation users. This option is applicable and required when you write data to a Hive target in the native environment.
Hive Staging Database Name	Namespace for Hive staging tables. Use the name `default` for tables that do not have a specified database name. This option is applicable when you run a mapping in the native environment to write data to a Hive target. If you run the mapping on the Blaze or Spark engine, you do not need to configure the Hive staging database name in the Hive connection. The Data Integration Service uses the value that you configure in the Hadoop connection.

JDBC Connection Properties

You can use a JDBC connection to access tables in a database. You can create and manage a JDBC connection in the Administrator tool, the Developer tool, or the Analyst tool.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes JDBC connection properties:

Property	Description
Database Type	The database type.
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: `~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /`
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 765 characters.
User Name	The database user name. If you configure Sqoop, Sqoop uses the user name that you configure in this field. If you configure the --username argument in a JDBC connection or mapping, Sqoop ignores the argument.
Password	The password for the database user name. If you configure Sqoop, Sqoop uses the password that you configure in this field. If you configure the --password argument in a JDBC connection or mapping, Sqoop ignores the argument.
JDBC Driver Class Name	Name of the JDBC driver class. The following list provides the driver class name that you can enter for the applicable database type: - DataDirect JDBC driver class name for Oracle: `com.informatica.jdbc.oracle.OracleDriver` - DataDirect JDBC driver class name for IBM DB2: `com.informatica.jdbc.db2.DB2Driver` - DataDirect JDBC driver class name for Microsoft SQL Server: `com.informatica.jdbc.sqlserver.SQLServerDriver` - DataDirect JDBC driver class name for Sybase ASE: `com.informatica.jdbc.sybase.SybaseDriver` - DataDirect JDBC driver class name for Informix: `com.informatica.jdbc.informix.InformixDriver` - DataDirect JDBC driver class name for MySQL: `com.informatica.jdbc.mysql.MySQLDriver` For more information about which driver class to use with specific databases, see the vendor documentation.

Property	Description
Connection String	Connection string to connect to the database. Use the following connection string: `jdbc:<subprotocol>:<subname>` The following list provides sample connection strings that you can enter for the applicable database type: - Connection string for DataDirect Oracle JDBC driver: `jdbc:informatica:oracle://<host>:<port>;SID=<value>` - Connection string for Oracle JDBC driver: `jdbc:oracle:thin:@//<host>:<port>:<SID>` - Connection string for DataDirect IBM DB2 JDBC driver: `jdbc:informatica:db2://<host>:<port>;DatabaseName=<value>` - Connection string for IBM DB2 JDBC driver: `jdbc:db2://<host>:<port>/<database_name>` - Connection string for DataDirect Microsoft SQL Server JDBC driver: `jdbc:informatica:sqlserver://<host>;DatabaseName=<value>` - Connection string for Microsoft SQL Server JDBC driver: `jdbc:sqlserver://<host>;DatabaseName=<value>` - Connection string for Netezza JDBC driver: `jdbc:netezza://<host>:<port>/<database_name>` - Connection string for Pivotal Greenplum driver: `jdbc:pivotal:greenplum://<host>:<port>;/database_name=<value>` - Connection string for Postgres Greenplum driver: `jdbc:postgressql://<host>:<port>/<database_name>` - Connection string for Teradata JDBC driver: `jdbc:teradata://<host>/database_name=<value>,tmode=<value>,charset=<value>` For more information about the connection string to use with specific drivers, see the vendor documentation.
Environment SQL	Optional. Enter SQL commands to set the database environment when you connect to the database. The Data Integration Service executes the connection environment SQL each time it connects to the database. **Note:** If you enable Sqoop, Sqoop ignores this property.
Transaction SQL	Optional. Enter SQL commands to set the database environment when you connect to the database. The Data Integration Service executes the transaction environment SQL at the beginning of each transaction. **Note:** If you enable Sqoop, Sqoop ignores this property.
SQL Identifier Character	Type of character that the database uses to enclose delimited identifiers in SQL queries. The available characters depend on the database type. Select (None) if the database uses regular identifiers. When the Data Integration Service generates SQL queries, the service does not place delimited characters around any identifiers. Select a character if the database uses delimited identifiers. When the Data Integration Service generates SQL queries, the service encloses delimited identifiers within this character. **Note:** If you enable Sqoop, Sqoop ignores this property.
Support Mixed-case Identifiers	Enable if the database uses case-sensitive identifiers. When enabled, the Data Integration Service encloses all identifiers within the character selected for the **SQL Identifier Character** property. When the **SQL Identifier Character** property is set to none, the **Support Mixed-case Identifiers** property is disabled. **Note:** If you enable Sqoop, Sqoop honors this property when you generate and execute a DDL script to create or replace a target at run time. In all other scenarios, Sqoop ignores this property.

Property	Description
Use Sqoop Connector	Enables Sqoop connectivity for the data object that uses the JDBC connection. The Data Integration Service runs the mapping in the Hadoop run-time environment through Sqoop. You can configure Sqoop connectivity for relational data objects, customized data objects, and logical data objects that are based on a JDBC-compliant database. Select **Sqoop v1.x** to enable Sqoop connectivity. Default is **None**.

Sqoop Arguments	Enter the arguments that Sqoop must use to connect to the database. Separate multiple arguments with a space.
	To run the mapping on the Blaze engine with the Teradata Connector for Hadoop (TDCH) specialized connectors for Sqoop, you must define the TDCH connection factory class in the Sqoop arguments. The connection factory class varies based on the TDCH Sqoop Connector that you want to use.
	- To use Cloudera Connector Powered by Teradata, configure the following Sqoop argument:
	`-Dsqoop.connection.factories=com.cloudera.connector.teradata.TeradataManagerFactory`
	- To use Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop), configure the following Sqoop argument:
	`-Dsqoop.connection.factories=org.apache.sqoop.teradata.TeradataManagerFactory`
	To run the mapping on the Spark engine, you do not need to define the TDCH connection factory class in the Sqoop arguments. The Data Integration Service invokes the Cloudera Connector Powered by Teradata and Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop) by default.
	Note: To run the mapping with a generic JDBC connector instead of the specialized Cloudera or Hortonworks connector, you must define the --driver and --connection-manager Sqoop arguments in the JDBC connection. If you define the --driver and --connection-manager arguments in the Read or Write transformation of the mapping, Sqoop ignores the arguments.
	If you do not enter Sqoop arguments, the Data Integration Service constructs the Sqoop command based on the JDBC connection properties.

Sqoop Connection-Level Arguments

In the JDBC connection, you can define the arguments that Sqoop must use to connect to the database. The Data Integration Service merges the arguments that you specify with the default command that it constructs based on the JDBC connection properties. The arguments that you specify take precedence over the JDBC connection properties.

If you want to use the same driver to import metadata and run the mapping, and do not want to specify any additional Sqoop arguments, select **Sqoop v1.x** from the **Use Sqoop Version** list and leave the **Sqoop Arguments** field empty in the JDBC connection. The Data Integration Service constructs the Sqoop command based on the JDBC connection properties that you specify.

However, if you want to use a different driver for run-time tasks or specify additional run-time Sqoop arguments, select **Sqoop v1.x** from the **Use Sqoop Version** list and specify the arguments in the **Sqoop Arguments** field.

You can configure the following Sqoop arguments in the JDBC connection:

driver

Defines the JDBC driver class that Sqoop must use to connect to the database.

Use the following syntax:

```
--driver <JDBC driver class>
```

For example, use the following syntax depending on the database type that you want to connect to:

- Aurora: `--driver com.mysql.jdbc.Driver`
- Greenplum: `--driver org.postgresql.Driver`
- IBM DB2: `--driver com.ibm.db2.jcc.DB2Driver`
- IBM DB2 z/OS: `--driver com.ibm.db2.jcc.DB2Driver`
- Microsoft SQL Server: `--driver com.microsoft.sqlserver.jdbc.SQLServerDriver`
- Netezza: `--driver org.netezza.Driver`
- Oracle: `--driver oracle.jdbc.driver.OracleDriver`
- Teradata: `--driver com.teradata.jdbc.TeraDriver`

connect

Defines the JDBC connection string that Sqoop must use to connect to the database. The JDBC connection string must be based on the driver that you define in the driver argument.

Use the following syntax:

```
--connect <JDBC connection string>
```

For example, use the following syntax depending on the database type that you want to connect to:

- Aurora: `--connect "jdbc:mysql://<host_name>:<port>/<schema_name>"`
- Greenplum: `--connect jdbc:postgresql://<host_name>:<port>/<database_name>`
- IBM DB2: `--connect jdbc:db2://<host_name>:<port>/<database_name>`
- IBM DB2 z/OS: `--connect jdbc:db2://<host_name>:<port>/<database_name>`
- Microsoft SQL Server: `--connect jdbc:sqlserver://<host_name>:<port or named_instance>;databaseName=<database_name>`
- Netezza: `--connect "jdbc:netezza://<database_server_name>:<port>/<database_name>;schema=<schema_name>"`
- Oracle: `--connect jdbc:oracle:thin:@<database_host_name>:<database_port>:<database_SID>`
- Teradata: `--connect jdbc:teradata://<host_name>/database=<database_name>`

connection-manager

Defines the connection manager class name that Sqoop must use to connect to the database.

Use the following syntax:

```
--connection-manager <connection manager class name>
```

For example, use the following syntax to use the generic JDBC manager class name:

```
--connection-manager org.apache.sqoop.manager.GenericJdbcManager
```

direct

When you read data from or write data to Oracle, you can configure the direct argument to enable Sqoop to use OraOop. OraOop is a specialized Sqoop plug-in for Oracle that uses native protocols to connect to the Oracle database. When you configure OraOop, the performance improves.

You can configure OraOop when you run Sqoop mappings on the Spark engine.

Use the following syntax:

```
--direct
```

When you use OraOop, you must use the following syntax to specify multiple arguments:

```
-D<argument=value> -D<argument=value>
```

Note: If you specify multiple arguments and include a space character between -D and the argument name-value pair, Sqoop considers only the first argument and ignores the remaining arguments.

If you do not direct the job to a specific queue, the Spark engine uses the default queue.

-Dsqoop.connection.factories

To run the mapping on the Blaze engine with the Teradata Connector for Hadoop (TDCH) specialized connectors for Sqoop, you must configure the -Dsqoop.connection.factories argument. Use the argument to define the TDCH connection factory class that Sqoop must use. The connection factory class varies based on the TDCH Sqoop Connector that you want to use.

- To use Cloudera Connector Powered by Teradata, configure the -Dsqoop.connection.factories argument as follows:
  `-Dsqoop.connection.factories=com.cloudera.connector.teradata.TeradataManagerFactory`
- To use Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop), configure the -Dsqoop.connection.factories argument as follows:
  `-Dsqoop.connection.factories=org.apache.sqoop.teradata.TeradataManagerFactory`

Note: To run the mapping on the Spark engine, you do not need to configure the -Dsqoop.connection.factories argument. The Data Integration Service invokes Cloudera Connector Powered by Teradata and Hortonworks Connector for Teradata (powered by the Teradata Connector for Hadoop) by default.

--infaoptimize

Use this argument to disable the performance optimization of Sqoop pass-through mappings on the Spark engine.

When you run a Sqoop pass-through mapping on the Spark engine, the Data Integration Service optimizes mapping performance in the following scenarios:

- You read data from a Sqoop source and write data to a Hive target that uses the Text format.
- You read data from a Sqoop source and write data to an HDFS target that uses the Flat, Avro, or Parquet format.

If you want to disable the performance optimization, set the --infaoptimize argument to false. For example, if you see data type issues after you run an optimized Sqoop mapping, you can disable the performance optimization.

Use the following syntax:

```
--infaoptimize false
```

For a complete list of the Sqoop arguments that you can configure, see the Sqoop documentation.

Microsoft Azure Blob Storage Connection Properties

Use a Microsoft Azure SQL Blob Storage connection to access a Microsoft Azure Blob Storage.

Note: The order of the connection properties might vary depending on the tool where you view them.

You can create and manage a Microsoft Azure Blob Storage connection in the Administrator tool or the Developer tool. The following table describes the Microsoft Azure Blob Storage connection properties:

Property	Description
Name	Name of the Microsoft Azure Blob Storage connection.
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Description of the connection.
Location	The domain where you want to create the connection.
Type	Type of connection. Select AzureBlob.

The **Connection Details** tab contains the connection attributes of the Microsoft Azure Blob Storage connection. The following table describes the connection attributes:

Property	Description
Account Name	Name of the Microsoft Azure Storage account.
Account Key	Microsoft Azure Storage access key.
Container Name	The root container or sub-folders with the absolute path.
Endpoint Suffix	Type of Microsoft Azure end-points. You can select any of the following end-points: - `core.windows.net`: Default - `core.usgovcloudapi.net`: To select the US government Microsoft Azure end-points - `core.chinacloudapi.cn`: Not applicable

Microsoft Azure Cosmos DB SQL API Connection Properties

Use a Microsoft Azure Cosmos DB connection to connect to the Cosmos DB database. When you create a Microsoft Azure Cosmos DB connection, you enter information for metadata and data access.

The following table describes the Microsoft Azure Cosmos DB connection properties:

Property	Description
Name	Name of the Cosmos DB connection.
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Description of the connection. The description cannot exceed 765 characters.
Location	The project or folder in the Model repository where you want to store the Cosmos DB connection.
Type	Select Microsoft Azure Cosmos DB SQL API.
Cosmos DB URI	The URI of Microsoft Azure Cosmos DB account.
Key	The primary and secondary key to which provides you complete administrative access to the resources within Microsoft Azure Cosmos DB account.
Database	Name of the database that contains the collections from which you want to read or write JSON documents.

Note: You can find the Cosmos DB URI and Key values in the **Keys** settings on Azure portal. Contact your Azure administrator for more details.

Microsoft Azure Data Lake Store Connection Properties

Use a Microsoft Azure Data Lake Store connection to access a Microsoft Azure Data Lake Store.

Note: The order of the connection properties might vary depending on the tool where you view them.

You can create and manage a Microsoft Azure SQL Data Warehouse connection in the Administrator tool or the Developer tool. The following table describes the Microsoft Azure Data Lake Store connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select Microsoft Azure Data Lake Store.

The following table describes the properties for metadata access:

Property	Description
ADLS Account Name	The name of the Microsoft Azure Data Lake Store.
ClientID	The ID of your application to complete the OAuth Authentication in the Active Directory.
Client Secret	The client secret key to complete the OAuth Authentication in the Active Directory.
Directory	The Microsoft Azure Data Lake Store directory that you use to read data or write data. The default is root directory.
AuthEndpoint	The OAuth 2.0 token endpoint from where access code is generated based on based on the Client ID and Client secret is completed.

For more information about creating a client ID, client secret, and auth end point, contact the Azure administrator or see Microsoft Azure Data Lake Store documentation.

Microsoft Azure SQL Data Warehouse Connection Properties

Use a Microsoft Azure SQL Data Warehouse connection to access a Microsoft Azure SQL Data Warehouse.

Note: The order of the connection properties might vary depending on the tool where you view them.

You can create and manage a Microsoft Azure SQL Data Warehouse connection in the Administrator tool or the Developer tool. The following table describes the Microsoft Azure SQL Data Warehouse connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select Microsoft Azure SQL Data Warehouse.

The following table describes the properties for metadata access:

Property	Description
Azure DW JDBC URL	Microsoft Azure Data Warehouse JDBC connection string. For example, you can enter the following connection string: jdbc:sqlserver:// <Server>.database.windows.net: 1433;database=<Database>. The Administrator can download the URL from Microsoft Azure portal.
Azure DW JDBC Username	User name to connect to the Microsoft Azure SQL Data Warehouse account. You must have permission to read, write, and truncate data in Microsoft Azure SQL Data Warehouse.
Azure DW JDBC Password	Password to connect to the Microsoft Azure SQL Data Warehouse account.
Azure DW Schema Name	Name of the schema in Microsoft Azure SQL Data Warehouse.
Azure Blob Account Name	Name of the Microsoft Azure Storage account to stage the files.
Azure Blob Account Key	The key that authenticates the access to the Blob storage account.
Blob End-point	Type of Microsoft Azure end-points. You can select any of the following end-points: - `core.windows.net`: Default - `core.usgovcloudapi.net`: To select the US government Microsoft Azure end-points - `core.chinacloudapi.cn`: Not applicable You can configure the US government Microsoft Azure end-points when a mapping runs in the native environment and on the Spark engine.

Snowflake Connection Properties

When you set up a Snowflake connection, you must configure the connection properties.

Note: The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Snowflake connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~ ` ! $ % ^ & * ( ) - + = { [}] | \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select SnowFlake.
Username	The user name to connect to the Snowflake account.
Password	The password to connect to the Snowflake account.
Account	The name of the Snowflake account.
Warehouse	The Snowflake warehouse name.
Role	The Snowflake role assigned to the user.

Additional JDBC URL Parameters	Enter one or more JDBC connection parameters in the following format:
	`<param1>=<value>&<param2>=<value>&<param3>=<value>....`
	For example:
	`user=jon&warehouse=mywh&db=mydb&schema=public`
	To access Snowflake through Okta SSO authentication, enter the web-based IdP implementing SAML 2.0 protocol in the following format:
	`authenticator=https://<Your_Okta_Account_Name>.okta.com`
	Note: Microsoft ADFS is not supported.
	For more information about configuring Okta authentication, see the following website: https://docs.snowflake.net/manuals/user-guide/admin-security-fed-auth-configure-snowflake.html#configuring-snowflake-to-use-federated-authentication

Creating a Connection to Access Sources or Targets

Create connections before you import data objects, preview data, and profile data.

1. Within the Administrator tool click **Manage** > **Connections**.
2. Select **Actions** > **New** > **Connection**.
3. Select the type of connection that you want to create:
   - To select an HBase connection, select **NoSQL** > **HBase**.
   - To select an HDFS connection, select **File Systems** > **Hadoop File System**.
   - To select a Hive connection, select **Database** > **Hive**.
   - To select a JDBC connection, select **Database** > **JDBC**.
4. Click **OK**.
5. Enter a connection name, ID, and optional description.
6. Configure the connection properties. For a Hive connection, you must choose the **Access Hive as a source or target** option to use Hive as a source or a target.
7. Click **Test Connection** to verify the connection.
8. Click **Finish**.

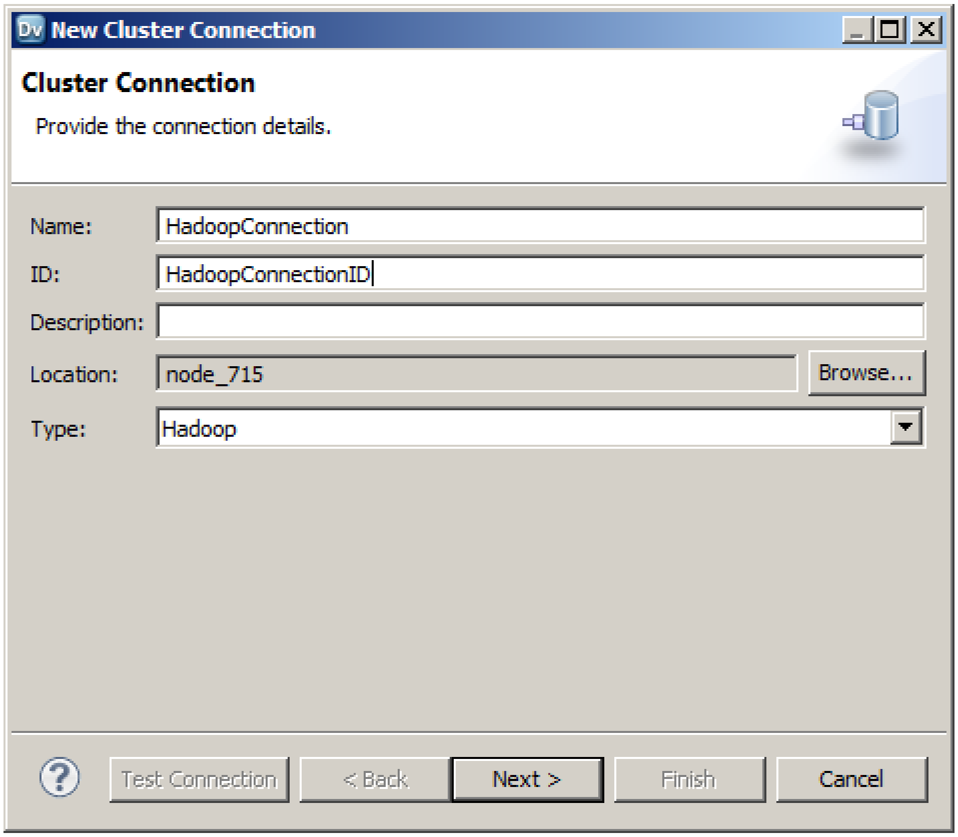
Creating a Hadoop Connection

Create a Hadoop connection before you run a mapping in the Hadoop environment.

1. Click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain in the **Available Connections** list.
4. Select the **Cluster** connection type in the **Available Connections** list and click **Add**.

   The **New Cluster Connection** dialog box appears.

5. Enter the general properties for the connection.



Cluster Connection

Provide the connection details.

Name: HadoopConnection

ID: HadoopConnectionID

Description:

Location: node_715

Type: Hadoop

6. Click **Next**.
7. Enter the Hadoop cluster properties, common properties, and the reject directory properties.
8. Click **Next**.
9. Click **Next**.

   Effective in version 10.2.2, Informatica dropped support for the Hive engine. Do not enter Hive configuration properties.
10. Enter configuration properties for the Blaze engine and click **Next**.
11. Enter configuration properties for the Spark engine and click **Finish**.

Configuring Hadoop Connection Properties

When you create a Hadoop connection, default values are assigned to cluster environment variables, cluster path properties, and advanced properties. You can add or edit values for these properties. You can also reset to default values.

You can configure the following Hadoop connection properties based on the cluster environment and functionality that you use:

- Cluster Environment Variables
- Cluster Library Path

- Common Advanced Properties
- Blaze Engine Advanced Properties
- Spark Engine Advanced Properties

Note: Informatica does not recommend changing these property values before you consult with third-party documentation, Informatica documentation, or Informatica Global Customer Support. If you change a value without knowledge of the property, you might experience performance degradation or other unexpected results.

To reset to default values, delete the property values. For example, if you delete the values of an edited Cluster Library Path property, the value resets to the default $DEFAULT_CLUSTER_LIBRARY_PATH.

Cluster Environment Variables

Cluster Environment Variables property lists the environment variables that the cluster uses. Each environment variable contains a name and a value. You can add environment variables or edit environment variables.

To edit the property in the text box, use the following format with &: to separate each name-value pair:

```
<name1>=<value1>[&:<name2>=<value2>…&:<nameN>=<valueN>]
```

Configure the following environment variables in the **Cluster Environment Variables** property:

HADOOP_NODE_JDK_HOME

Represents the directory from which you run the cluster services and the JDK version that the cluster nodes use. Required to run the Java transformation in the Hadoop environment and Sqoop mappings on the Blaze engine. Default is /usr/java/default. The JDK version that the Data Integration Service uses must be compatible with the JDK version on the cluster.

Set to <cluster JDK home>/jdk<version>.

For example, `HADOOP_NODE_JDK_HOME=<cluster JDK home>/jdk<version>`.

Cluster Library Path

Cluster Library Path property is a list of path variables for shared libraries on the cluster. You can add or edit library path variables.

To edit the property in the text box, use the following format with : to separate each path variable:

```
<variable1>[:<variable2>…:<variableN]
```

Configure the library path variables in the **Cluster Library Path** property.

Common Advanced Properties

Common advanced properties are a list of advanced or custom properties that are unique to the Hadoop environment. The properties are common to the Blaze and Spark engines. Each property contains a name and a value. You can add or edit advanced properties.

To edit the property in the text box, use the following format with &: to separate each name-value pair:

```
<name1>=<value1>[&:<name2>=<value2>…&:<nameN>=<valueN>]
```

Configure the following property in the **Advanced Properties** of the common properties section:

infapdo.java.opts

List of Java options to customize the Java run-time environment. The property contains default values.

If mappings in a MapR environment contain a Consolidation transformation or a Match transformation, change the following value:

- -Xmx512M. Specifies the maximum size for the Java virtual memory. Default is 512 MB. Increase the value to at least 700 MB.
  For example, `infapdo.java.opts=-Xmx700M`

Blaze Engine Advanced Properties

Blaze advanced properties are a list of advanced or custom properties that are unique to the Blaze engine. Each property contains a name and a value. You can add or edit advanced properties.

To edit the property in the text box, use the following format with &: to separate each name-value pair:

```
<name1>=<value1>[&:<name2>=<value2>…&:<nameN>=<valueN>]
```

Configure the following properties in the **Advanced Properties** of the Blaze configuration section:

infagrid.cadi.namespace

Namespace for the Data Integration Service to use. Required to set up multiple Blaze instances.

Set to <unique namespace>.

For example, `infagrid.cadi.namespace=TestUser1_namespace`

infagrid.blaze.console.jsfport

JSF port for the Blaze engine console. Use a port number that no other cluster processes use. Required to set up multiple Blaze instances.

Set to <unique JSF port value>.

For example, `infagrid.blaze.console.jsfport=9090`

infagrid.blaze.console.httpport

HTTP port for the Blaze engine console. Use a port number that no other cluster processes use. Required to set up multiple Blaze instances.

Set to <unique HTTP port value>.

For example, `infagrid.blaze.console.httpport=9091`

infagrid.node.local.root.log.dir

Path for the Blaze service logs. Default is /tmp/infa/logs/blaze. Required to set up multiple Blaze instances.

Set to <local Blaze services log directory>.

For example, `infagrid.node.local.root.log.dir=<directory path>`

infacal.hadoop.logs.directory

Path in HDFS for the persistent Blaze logs. Default is /var/log/hadoop-yarn/apps/informatica. Required to set up multiple Blaze instances.

Set to <persistent log directory path>.

For example, `infacal.hadoop.logs.directory=<directory path>`

infagrid.node.hadoop.local.root.log.dir

Path in the Hadoop connection for the service log directory.

Set to <service log directory path>.

For example, `infagrid.node.local.root.log.dir=$HADOOP_NODE_INFA_HOME/blazeLogs`

Spark Advanced Properties

Spark advanced properties are a list of advanced or custom properties that are unique to the Spark engine. Each property contains a name and a value. You can add or edit advanced properties. Each property contains a name and a value. You can add or edit advanced properties.

Configure the following properties in the **Advanced Properties** of the Spark configuration section:

To edit the property in the text box, use the following format with &: to separate each name-value pair:

```
<name1>=<value1>[&:<name2>=<value2>…&:<nameN>=<valueN>]
```

spark.authenticate

Enables authentication for the Spark service on Hadoop. Required for Spark encryption.

Set to TRUE.

For example, `spark.authenticate=TRUE`

spark.authenticate.enableSaslEncryption

Enables encrypted communication when SASL authentication is enabled. Required if Spark encryption uses SASL authentication.

Set to TRUE.

For example, `spark.authenticate.enableSaslEncryption=TRUE`

spark.executor.cores

Indicates the number of cores that each executor process uses to run tasklets on the Spark engine.

Set to: `spark.executor.cores=1`

spark.executor.instances

Indicates the number of instances that each executor process uses to run tasklets on the Spark engine.

Set to: `spark.executor.instances=1`

spark.executor.memory

Indicates the amount of memory that each executor process uses to run tasklets on the Spark engine.

Set to: `spark.executor.memory=3G`

infaspark.driver.cluster.mode.extraJavaOptions

List of extra Java options for the Spark driver that runs inside the cluster. Required for streaming mappings to read from or write to a Kafka cluster that uses Kerberos authentication.

For example, set to:

```
infaspark.driver.cluster.mode.extraJavaOptions=
-Djava.security.egd=file:/dev/./urandom
-XX:MaxMetaspaceSize=256M -Djavax.security.auth.useSubjectCredsOnly=true
-Djava.security.krb5.conf=/<path to keytab file>/krb5.conf
-Djava.security.auth.login.config=<path to jaas config>/kafka_client_jaas.config
```

To configure the property for a specific user, you can include the following lines of code:

```
infaspark.driver.cluster.mode.extraJavaOptions =
-Djava.security.egd=file:/dev/./urandom
-XX:MaxMetaspaceSize=256M -XX:+UseG1GC -XX:MaxGCPauseMillis=500
-Djava.security.krb5.conf=/etc/krb5.conf
```

infaspark.executor.extraJavaOptions

List of extra Java options for the Spark executor. Required for streaming mappings to read from or write to a Kafka cluster that uses Kerberos authentication.

For example, set to:

```
infaspark.executor.extraJavaOptions=
-Djava.security.egd=file:/dev/./urandom
-XX:MaxMetaspaceSize=256M -Djavax.security.auth.useSubjectCredsOnly=true
-Djava.security.krb5.conf=/<path to krb5.conf file>/krb5.conf
-Djava.security.auth.login.config=/<path to jAAS config>/kafka_client_jaas.config
```

To configure the property for a specific user, you can include the following lines of code:

```
infaspark.executor.extraJavaOptions =
-Djava.security.egd=file:/dev/./urandom
-XX:MaxMetaspaceSize=256M -XX:+UseG1GC -XX:MaxGCPauseMillis=500
-Djava.security.krb5.conf=/etc/krb5.conf
```

infaspark.flatfile.writer.nullValue

When the Databricks Spark engine writes to a target, it converts null values to empty strings (" "). For example, 12, AB,"",23p09udj.

The Databricks Spark engine can write the empty strings to string columns, but when it tries to write an empty string to a non-string column, the mapping fails with a type mismatch.

To allow the Databricks Spark engine to convert the empty strings back to null values and write to the target, configure the following advanced property in the Databricks Spark connection:

```
infaspark.flatfile.writer.nullValue=true
```

spark.hadoop.validateOutputSpecs

Validates if the HBase table exists. Required for streaming mappings to write to a HBase target in an Amazon EMR cluster. Set the value to false.

infaspark.json.parser.mode

Specifies the parser how to handle corrupt JSON records. You can set the value to one of the following modes:

- DROPMALFORMED. The parser ignores all corrupted records. Default mode.
- PERMISSIVE. The parser accepts non-standard fields as nulls in corrupted records.
- FAILFAST. The parser generates an exception when it encounters a corrupted record and the Spark application goes down.

infaspark.json.parser.multiLine

Specifies whether the parser can read a multiline record in a JSON file. You can set the value to true or false. Default is false. Applies only to non-native distributions that use Spark version 2.2.x and above.

infaspark.pythontx.exec

Required to run a Python transformation on the Spark engine for Big Data Management. The location of the Python executable binary on the worker nodes in the Hadoop cluster.

For example, set to:

```
infaspark.pythontx.exec=/usr/bin/python3.4
```

If you use the installation of Python on the Data Integration Service machine, set the value to the Python executable binary in the Informatica installation directory on the Data Integration Service machine.

For example, set to:

```
infaspark.pythontx.exec=INFA_HOME/services/shared/spark/python/lib/python3.4
```

infaspark.pythontx.executorEnv.PYTHONHOME

Required to run a Python transformation on the Spark engine for Big Data Management and Big Data Streaming. The location of the Python installation directory on the worker nodes in the Hadoop cluster.

If the Python installation directory on the worker nodes is in a directory such as `usr/lib/python`, set the property to the following value:

```
infaspark.pythontx.executorEnv.PYTHONHOME=usr/lib/python
```

If you use the installation of Python on the Data Integration Service machine, use the location of the Python installation directory on the Data Integration Service machine.

For example, set the property to the following value:

```
infaspark.pythontx.executorEnv.PYTHONHOME=
INFA_HOME/services/shared/spark/python/
```

infaspark.pythontx.executorEnv.LD_PRELOAD

Required to run a Python transformation on the Spark engine for Big Data Streaming. The location of the Python shared library in the Python installation folder on the Data Integration Service machine.

For example, set to:

```
infaspark.pythontx.executorEnv.LD_PRELOAD=
INFA_HOME/services/shared/spark/python/lib/libpython3.6m.so
```

infaspark.pythontx.submit.lib.JEP_HOME

Required to run a Python transformation on the Spark engine for Big Data Streaming. The location of the Jep package in the Python installation folder on the Data Integration Service machine.

For example, set to:

```
infaspark.pythontx.submit.lib.JEP_HOME=
INFA_HOME/services/shared/spark/python/lib/python3.6/site-packages/jep/
```

spark.shuffle.encryption.enabled

Enables encrypted communication when authentication is enabled. Required for Spark encryption.

Set to TRUE.

For example, `spark.shuffle.encryption.enabled=TRUE`

spark.scheduler.maxRegisteredResourcesWaitingTime

The number of milliseconds to wait for resources to register before scheduling a task. Default is 30000. Decrease the value to reduce delays before starting the Spark job execution. Required to improve performance for mappings on the Spark engine.

Set to 15000.

For example, `spark.scheduler.maxRegisteredResourcesWaitingTime=15000`

spark.scheduler.minRegisteredResourcesRatio

The minimum ratio of registered resources to acquire before task scheduling begins. Default is 0.8. Decrease the value to reduce any delay before starting the Spark job execution. Required to improve performance for mappings on the Spark engine.

Set to: 0.5

For example, `spark.scheduler.minRegisteredResourcesRatio=0.5`

INDEX