Informatica® Big Data Management
10.2 HotFix 1

Hadoop Integration Guide

Informatica Big Data Management Hadoop Integration Guide
10.2 HotFix 1
August 2018

Publication Date: 2018-12-11

Table of Contents

Preface 9

Informatica Resources 9

- Informatica Network 9
- Informatica Knowledge Base 9
- Informatica Documentation 9
- Informatica Product Availability Matrixes 10
- Informatica Velocity 10
- Informatica Marketplace 10
- Informatica Global Customer Support 10

Chapter 1: Introduction to Hadoop Integration 11

Hadoop Integration Overview 11

How to Use This Guide 12

Big Data Management Component Architecture 13

- Hadoop Environment 13
- Clients and Tools 14
- Application Services 14
- Repositories 14

Integration with Other Informatica Products 15

- PowerExchange Adapters Connectivity 15
- Data Lineage 16
- Data Lakes 16
- Address Validation 17
- Data Discovery 17
- Data Replication 17
- Persistent Data Masking 17
- Data Streaming 18

Chapter 2: Before You Begin 19

Read the Release Notes 19

Verify System Requirements 19

- Verify Product Installations 19
- Verify HDFS Disk Space 20
- Verify the Hadoop Distribution 20
- Verify Port Requirements 20

Uninstall Big Data Management 21

- Uninstall for Amazon EMR, Azure HDInsight, IBM BigInsights, and MapR 21
- Uninstall for Cloudera CDH 22
- Uninstall for Hortonworks HDP 23

Prepare Directories, Users, and Permissions 23

Verify and Create Users. . . . 24
Create an Informatica Hadoop Staging Directory. . . . 24
Grant Permissions on the Hive Warehouse Directory. . . . 25
Create a Hive Staging Directory. . . . 25
Create Blaze Engine Directories. . . . 25
Create a Spark Staging Directory. . . . 26
Create a Reject File Directory. . . . 26
Configure the Data Integration Service. . . . 27
Download the Informatica Server Binaries for the Hadoop Environment. . . . 27
Configure Data Integration Service Properties. . . . 28

Chapter 3: Amazon EMR Integration Tasks. . . . 29

Amazon EMR Integration Task Flow. . . . 30
Prepare for Cluster Import from Amazon EMR. . . . 31
Configure *-site.xml Files for Amazon EMR. . . . 31
Prepare the Archive File for Amazon EMR. . . . 34
Edit the hosts File for the Blaze Engine. . . . 34
Create a Cluster Configuration. . . . 35
Importing a Cluster Configuration from a File. . . . 35
Generate Cluster Configuration Files. . . . 36
Update hadoopEnv.properties. . . . 36
Update odbc.ini. . . . 38
Download the JDBC Drivers for Sqoop Connectivity. . . . 38
Copy .jar Files for Hive Tables on S3 39
Set S3 Bucket Access Policies. . . . 39
Step 1. Identify the S3 Access Policy Elements. . . . 40
Step 2. Optionally Copy an Existing S3 Access Policy as a Template. . . . 40
Step 3. Create or Edit an S3 Access Policy. . . . 41
Configure the Developer Tool. . . . 41
Extract the Cluster Configuration Files. . . . 42
Configure developerCore.ini. . . . 42
Configure the Developer Tool for Kerberos. . . . 42
Complete Upgrade Tasks. . . . 42
Update the hadoopEnv.properties File. . . . 43
Replace the Connections. . . . 43
Complete Connection Upgrade from Version 10.0 or Later. . . . 43

Chapter 4: Azure HDInsight Integration Tasks. . . . 45

Azure HDInsight Integration Task Flow. . . . 46
Prepare for Cluster Import from Azure HDInsight. . . . 47
Configure *-site.xml Files for Azure HDInsight. . . . 47
Prepare for Direct Import from Azure HDInsight. . . . 50
Prepare the Archive File for Import from Azure HDInsight. . . . 50

Edit the hosts File for the Blaze Engine. . . . 51
Create a Cluster Configuration. . . . 51
Before You Import. . . . 52
Importing a Cluster Configuration from the Cluster. . . . 52
Importing a Cluster Configuration from a File. . . . 53
Generate Cluster Configuration Files. . . . 54
Update hadoopEnv.properties. . . . 54
Update odbc.ini. . . . 56
Download the JDBC Drivers for Sqoop Connectivity. . . . 56
Import Files for Data Decryption. . . . 57
Edit the hosts File. . . . 57
Configure the Developer Tool. . . . 57
Extract the Cluster Configuration Files. . . . 58
Configure developerCore.ini. . . . 58
Configure the Developer Tool for Kerberos. . . . 58
Complete Upgrade Tasks. . . . 58
Update the hadoopEnv.properties File. . . . 59
Replace the Connections. . . . 59
Complete Connection Upgrade from Version 10.0 or Later. . . . 59

Chapter 5: Cloudera CDH Integration Tasks. . . . 61

Cloudera CDH Integration Task Flow. . . . 62
Prepare for Cluster Import from Cloudera CDH. . . . 63
Configure *-site.xml Files for Cloudera CDH. . . . 63
Prepare for Direct Import from Cloudera CDH. . . . 66
Prepare the Archive File for Import from Cloudera CDH. . . . 66
Edit the hosts File for the Blaze Engine. . . . 67
Create a Cluster Configuration. . . . 67
Before You Import. . . . 67
Importing a Cluster Configuration from the Cluster. . . . 68
Importing a Cluster Configuration from a File. . . . 69
Generate Cluster Configuration Files. . . . 69
Update hadoopEnv.properties. . . . 70
Update odbc.ini. . . . 71
Download the JDBC Drivers for Sqoop Connectivity. . . . 72
Import Security Certificates to Clients. . . . 72
Configure the Developer Tool. . . . 73
Extract the Cluster Configuration Files. . . . 73
Configure developerCore.ini. . . . 73
Configure the Developer Tool for Kerberos. . . . 74
Complete Upgrade Tasks. . . . 74
Update the hadoopEnv.properties File. . . . 74
Replace the Connections. . . . 75

Complete Connection Upgrade from Version 10.0 or Later. 75
Complete Connection Upgrade from Version 9.6.1. 75

Chapter 6: Hortonworks HDP Integration Tasks. 77

Hortonworks HDP Integration Task Flow. 78
Prepare for Cluster Import from Hortonworks HDP. 79
Configure *-site.xml Files for Hortonworks HDP. 79
Prepare for Direct Import from Hortonworks HDP. 83
Prepare the Archive File for Import from Hortonworks HDP. 83
Edit the hosts File for the Blaze Engine. 84
Create a Cluster Configuration. 84
Before You Import. 84
Importing a Cluster Configuration from the Cluster. 84
Importing a Cluster Configuration from a File. 85
Generate Cluster Configuration Files. 86
Update hadoopEnv.properties. 87
Update odbc.ini. 88
Download the JDBC Drivers for Sqoop Connectivity. 89
Import Security Certificates to Clients. 89
Configure the Developer Tool. 90
Extract the Cluster Configuration Files. 90
Configure developerCore.ini. 90
Configure the Developer Tool for Kerberos. 90
Complete Upgrade Tasks. 91
Update the hadoopEnv.properties File. 91
Replace the Connections. 91
Complete Connection Upgrade from Version 10.0 or Later. 92
Complete Connection Upgrade from Version 9.6.1. 92

Chapter 7: MapR Integration Tasks. 94

MapR Integration Task Flow. 95
Prepare for Cluster Import from MapR. 97
Configure *-site.xml Files for MapR. 97
Prepare the Archive File for Import from MapR. 100
Edit the hosts File for the Blaze Engine. 100
Install the MapR Client and Configure Environment Path. 101
Create a Cluster Configuration. 101
Importing a Cluster Configuration from a File. 101
Generate Cluster Configuration Files. 102
Update hadoopEnv.properties. 103
Update odbc.ini. 104
Download the JDBC Drivers for Sqoop Connectivity. 105
Create a Proxy Directory for MapR. 105

Generate MapR Tickets. 105
Generate Tickets. 106
Configure the Data Integration Service. 106
Configure the Analyst Service. 107
Test the Hive Connection. 108
Get MapR Configuration Files for the Domain. 108
Configure the Developer Tool. 109
Extract the Cluster Configuration Files. 109
Configure Files to Enable the Developer Tool 109
Configure the Developer Tool for Kerberos. 110
Complete Upgrade Tasks. 110
Update the hadoopEnv.properties File. 110
Replace the Connections. 111
Complete Connection Upgrade from Version 10.0 or Later. 111
Complete Connection Upgrade from Version 9.6.1. 112

Index. 113

Preface

The *Informatica Big Data Management™ Hadoop Integration Guide* is written for the system administrator who is responsible for integrating the Informatica domain with the Hadoop environment through Informatica Big Data Management. This guide contains information for new integration and for upgrades. Readers of this guide need to have knowledge of the Hadoop environment and distributions, operating systems, and the database engines in their environment.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- 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 search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.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

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

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

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access 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. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access 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 augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

Introduction to Hadoop Integration

This chapter includes the following topics:

- Hadoop Integration Overview, 11
- How to Use This Guide, 12
- Big Data Management Component Architecture, 13
- Integration with Other Informatica Products, 15

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.

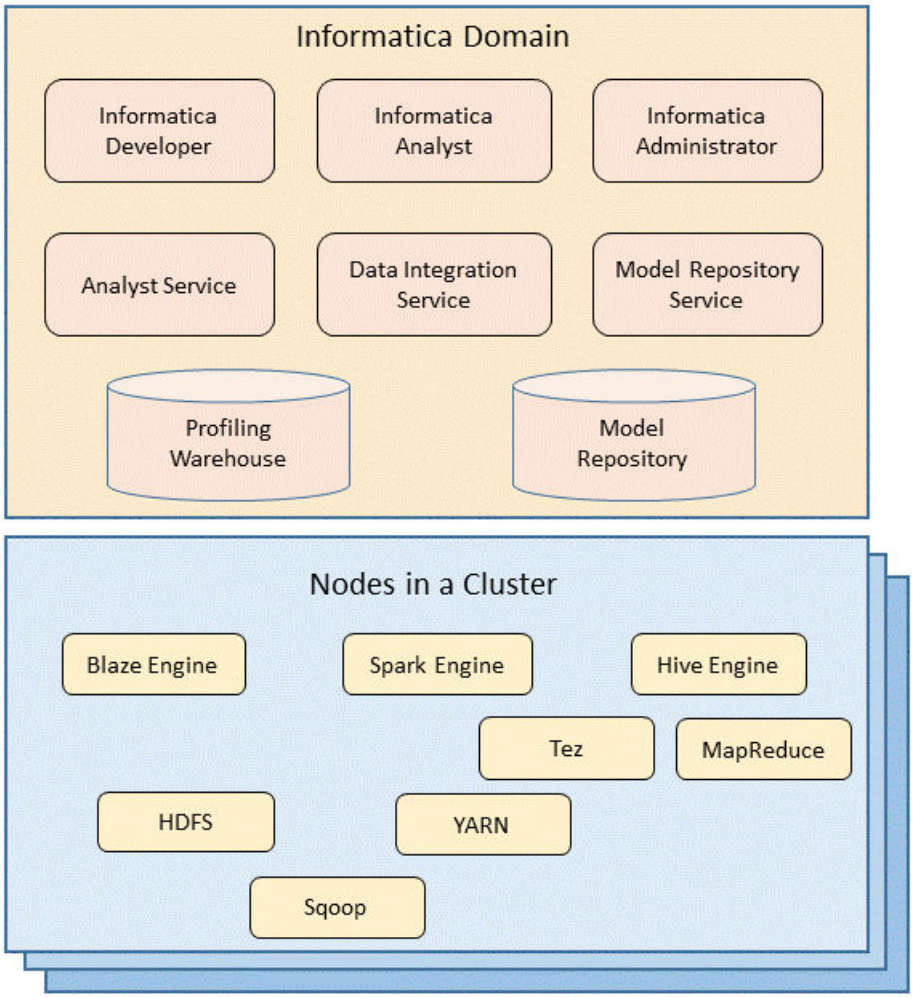
Integration tasks are required on the Hadoop cluster, the Data Integration Service machine, and the Developer tool machine. As a result, this guide contains tasks for Hadoop administrators, Informatica administrators, and Informatica mapping developers. Tasks required by the Hadoop administrator are 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 Hadoop as a data source and push job processing to the Hadoop cluster. It can also connect to HDFS, which enables high performance access to files across the cluster. It can connect to Hive, which is a data warehouse that connects to HDFS and uses SQL-like queries to run MapReduce jobs on Hadoop, or YARN, which can manage Hadoop clusters more efficiently. 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.

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.

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 Information Catalog. Perform data lineage analysis for big data sources and targets.
- Intelligent 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.
- Intelligent Streaming. Stream data as messages, and process it as it becomes available.
- Vibe Data Stream. Collect and ingest data in real time to a Kafka queue.

PowerExchange Adapters Connectivity

Based on your business needs, install and configure Informatica adapters. Use Big Data Management with Informatica adapters for access to sources and targets.

You can use the following Informatica adapters in conjunction with Big Data Management:

- PowerExchange for Amazon Redshift
- PowerExchange for Amazon S3
- PowerExchange for Greenplum
- PowerExchange for HBase
- PowerExchange for HDFS
- PowerExchange for Hive
- PowerExchange for MapR-DB
- PowerExchange for Microsoft Azure Blob Storage
- PowerExchange for Microsoft Azure SQL Data Warehouse
- PowerExchange for Netezza
- PowerExchange for Teradata Parallel Transporter API

For more information, see the PowerExchange adapter documentation.

Data Lineage

Perform data lineage analysis in Enterprise Information Catalog for big data sources and targets.

Use Enterprise Information Catalog to create a Cloudera Navigator resource to extract metadata for big data sources and targets and perform data lineage analysis on the metadata. Cloudera Navigator is a data management tool for the Hadoop platform that enables users to track data access for entities and manage metadata about the entities in a Hadoop cluster.

You can create one Cloudera Navigator resource for each Hadoop cluster that is managed by Cloudera Manager. Enterprise Information Catalog extracts metadata about entities from the cluster based on the entity type.

Enterprise Information Catalog extracts metadata for the following entity types:

- HDFS files and directories
- Hive tables, query templates, and executions
- Oozie job templates and executions
- Pig tables, scripts, and script executions
- YARN job templates and executions

Note: Enterprise Information Catalog does not extract metadata for MapReduce job templates or executions.

For more information, see the *Informatica Enterprise Information Catalog Administrator Guide*.

Data Lakes

A data lake is a shared repository of raw and enterprise data from a variety of sources. It is often built over a distributed Hadoop cluster, which provides an economical and scalable persistence and compute layer.

Intelligent Data Lake is a collaborative self-service big data discovery and preparation solution for data analysts and data scientists. It enables analysts to rapidly discover and turn raw data into insight and allows IT to ensure quality, visibility, and governance. With Intelligent Data Lake, analysts to spend more time on analysis and less time on finding and preparing data.

Hadoop makes it possible to store large volumes of structured and unstructured data from various enterprise systems within and outside the organization. Data in the lake can include raw and refined data, master data and transactional data, log files, and machine data.

Intelligent Data Lake provides the following benefits:

- Data analysts can use semantic search and smart recommendations to find and explore trusted data assets within the data lake and outside the data lake.
- Data analysts can transform, cleanse, and enrich data in the data lake using an Excel-like spreadsheet interface.
- Data analysts can publish data and share knowledge with the rest of the community and analyze the data using their choice of BI or analytic tools.
- IT and governance staff can monitor user activity related to data usage in the lake.
- IT can track data lineage to verify that data is coming from the right sources and going to the right targets.
- IT can enforce appropriate security and governance on the data lake.
- IT can operationalize the work done by data analysts into a data delivery process that can be repeated and scheduled.

Address Validation

If you have a Data Quality product license, you can push a mapping that contains data quality transformations to a Hadoop cluster. Data quality transformations can use reference data to verify that data values are accurate and correctly formatted.

When you apply a pushdown operation to a mapping that contains data quality transformations, the operation can copy the reference data that the mapping uses. The pushdown operation copies reference table data and content set data to the Hadoop cluster. After the mapping runs, the cluster deletes the reference data that the pushdown operation copied with the mapping.

Note: The pushdown operation does not copy address validation reference data. If you push a mapping that performs address validation, you must install the address validation reference data files on each DataNode that runs the mapping. The cluster does not delete the address validation reference data files after the address validation mapping runs.

Address validation mappings validate and enhance the accuracy of postal address records. You can buy address reference data files from Informatica on a subscription basis. You can download the current address reference data files from Informatica at any time during the subscription period.

For more information, see the *Informatica Reference Data Guide*.

Data Discovery

Data discovery is the process of discovering the metadata of source systems that include content, structure, patterns, and data domains. Content refers to data values, frequencies, and data types. Structure includes candidate keys, primary keys, foreign keys, and functional dependencies. The data discovery process offers advanced profiling capabilities.

Run a profile to evaluate the data structure and to verify that data columns contain the types of information you expect. You can drill down on data rows in profiled data. If the profile results reveal problems in the data, you can apply rules to fix the result set. You can create scorecards to track and measure data quality before and after you apply the rules.

You can push column profiles and the data domain discovery process to the Hadoop cluster.

For more information, see the *Informatica Data Discovery Guide*.

Data Replication

You can replicate large amounts of transactional data between heterogeneous databases and platforms with Data Replication. You might replicate data to distribute or migrate the data across your environment.

You can configure Data Replication to write transactional changes to flat files instead of to a target database. Data Replication can generate these flat files on a file system or Hadoop Distributed File System (HDFS).

For more information, see the *Informatica Data Replication User Guide*.

Persistent Data Masking

Use Test Data Management to perform persistent data masking. You can mask sensitive and confidential data in non-production systems.

You can perform data masking on data that is stored in a Hadoop cluster. Additionally, you can mask data during data ingestion in the native or Hadoop environment. Masking rules can replace, scramble, or initialize data. When you create a project, you can select masking rules for each table field that you want to mask.

When you run the project, the rule masks data in the cluster to create realistic data that you can use for development or testing purposes.

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

Data Streaming

You can subscribe to sources that stream data and process data as it becomes available. Streaming sources stream data as messages. Use Informatica Intelligent Streaming mappings to collect the streamed data, build the business logic for the data, and push the logic to a Spark engine for processing.

Intelligent Streaming is built on Informatica Big Data Management and extends the platform to provide streaming capabilities. Intelligent Streaming uses Spark Streaming to process streamed data. It uses YARN to manage the resources on a Spark cluster more efficiently and uses third-parties distributions to connect to and push job processing to a Hadoop environment.

You can create streaming mappings to stream machine, device, and social media data. You can stream data from sources such as JMS providers and Apache Kafka brokers. A streaming mapping receives data from unbounded data sources. An unbounded data source is one where data continuously flows in and there is no definite boundary. Sources stream data as events. The Spark engine receives the input data streams and divides the data into micro batches. The Spark engine processes the data and publishes data in batches.

You can also use Informatica Vibe Data Stream to collect and ingest data in real time, for example, data from sensors and machine logs to a Kafka queue. Intelligent Streaming can process the data.

For more information, see the *Informatica Intelligent Streaming User Guide* and the *Informatica Vibe Data Stream for Machine Data User Guide*.

CHAPTER 2

Before You Begin

This chapter includes the following topics:

- Read the Release Notes, 19
- Verify System Requirements, 19
- Uninstall Big Data Management, 21
- Prepare Directories, Users, and Permissions, 23
- Configure the Data Integration Service, 27

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 and a Data Integration 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 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 distribution version associate for the distribution in the Hadoop environment.

The following table lists the supported distribution versions:

Distribution	Version
Amazon EMR	5.8
Azure HDInsight	3.5.x 3.6.x
Cloudera CDH	5.10.x 5.11.x 5.12.x 5.13.x
Hortonworks HDP	2.5x 2.6x
MapR	5.2 MEP 3.0.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: IBM BigInsights, 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.

Uninstall Big Data Management

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

Uninstall for Amazon EMR, Azure HDInsight, IBM BigInsights, 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 hadoopEnv.properties file on the Data Integration Service machine contains the following entry for the services log directory:

```
#Services log directory on Processing nodes
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 hadoopEnv.properties file.

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 hadoopEnv.properties file on the Data Integration Service machine contains the following entry for the HDFS log directory:

```
#HDFS Log Directory for persisting logs
infacal.hadoop.logs.directory=/var/log/hadoop-yarn/apps/informatica
```

Ensure that value in the hadoopEnv.properties file 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 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 and the path to Hadoop integration files on the Data Integration Service machine.

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.

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. Delete files from the extracted location that are not necessary. For example, run the following command:

   ```
   rm -Rf Messages Server upgrade_utils unjar_esd.sh silentinstall.sh
   silentinstallDT.sh SilentInput_upgrade.properties
   SilentInput_upgrade_NewConfig.properties SilentInput.properties
   SilentInput_DT.properties saptrans sapsolutions properties install.sh logs
   ```
4. Extract infa_native_esd.jar. For example, run the following command:

   ```
   cd <Informatica server binaries>/source &&jar -xvf infa_native_esd.jar
   ```
5. Delete files that are not required. For example, run the following command:

   ```
   rm -Rf <Informatica server binary tar file> ./source/infa_native_esd.jar
   ```

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 needs 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.
Data Integration Service Hadoop Distribution Directory	The Hadoop distribution directory on the Data Integration Service node. Enter `<Informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>_<version>`.
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.

Custom Properties	Path to the truststore files in the Hadoop distribution directory. Enter the following name for the custom property: JVMOption. Supply the following value:
	`-Djavax.net.ssl.trustStore=<path to the truststore file on the cluster>`
	For example:
	`JVMOption=-Djavax.net.ssl.trustStore=/etc/security/serverKeys/all.jks`
	If a JVMOption custom property exists, then increment the name with an integer like JVMOption1.
	Note: The Data Integration service converts the name of custom properties to add the prefix "ExecutionContextOptions." when you recycle the service.

CHAPTER 3

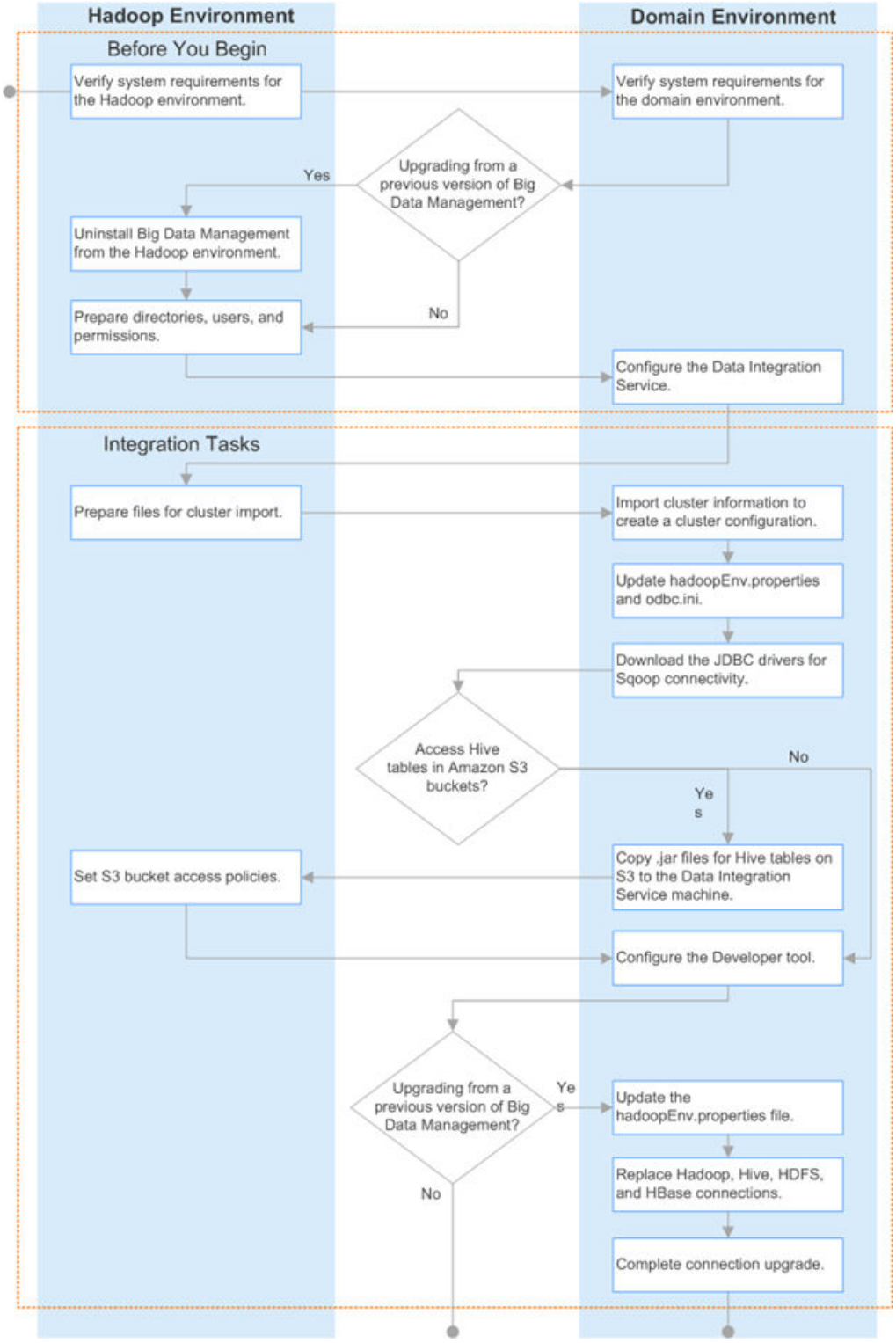
Amazon EMR Integration Tasks

This chapter includes the following topics:

- Amazon EMR Integration Task Flow, 30
- Prepare for Cluster Import from Amazon EMR, 31
- Create a Cluster Configuration, 35
- Update hadoopEnv.properties, 36
- Update odbc.ini, 38
- Download the JDBC Drivers for Sqoop Connectivity, 38
- Copy .jar Files for Hive Tables on S3 , 39
- Set S3 Bucket Access Policies, 39
- Configure the Developer Tool, 41
- Complete Upgrade Tasks, 42

Amazon EMR Integration Task Flow

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



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.

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

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. Required for Spark and Hive engines if S3 policy does not allow EMR access.

Set to your access key.

fs.s3.enableServerSideEncryption

Enables server side encryption for hive buckets. Required if the S3 bucket is encrypted.

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.

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 realm into local names within the Hadoop cluster.

Set to: RULE:[1:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$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.ZooKeeperTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for the Update Strategy transformation.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for the Update Strategy transformation.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for the Update Strategy transformation.

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. Also required for the Update Strategy transformation.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for the Update Strategy transformation.

Set to: TRUE

hive.txn.manager

Turns on transaction support. Required for the Update Strategy transformation.

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

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 engine.

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.

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.

The Developer tool requires access to the *-site.xml files for metadata browsing. After you create the cluster configuration, generate an archive file to extract on each Developer tool machine.

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.
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.

Generate Cluster Configuration Files

The Developer tool requires configuration files to access cluster metadata at design-time. Generate a cluster configuration archive file and extract it on the Developer tool machine. The archive contains .xml files based on the configuration sets in the cluster configuration.

Important: When you export the cluster configuration, you can export it with sensitive properties or without sensitive properties. When you export without sensitive properties, the sensitive properties are not included in the archive file. When you export with sensitive properties, the sensitive properties are exported with unmasked values. Sensitive properties are required only to access Hive sources on S3 in a non-EMR cluster.

1. From the **Connections** tab, expand the **Cluster Configuration** node in the Domain Navigator.
2. Select the cluster configuration that you want to export.
3. From the Actions menu, choose to export with sensitive properties only to access Hive sources on S3 in a non-EMR cluster. Otherwise, export without sensitive properties.

   The Administrator tool assigns a default name to the archive file using the name of the cluster configuration and a datetime string. For example, when the cluster configuration is named `CC1`: `CC1_2017-07-24-21-39-45.zip`
4. Accept the name or rename it, and then browse to a directory to save the file.

   The Service Manager creates a .zip archive file that contains all properties in the cluster configuration.

Give the .zip file to the mapping developers or copy the .zip file to the Developer tool machines and extract the contents to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

Update hadoopEnv.properties

Update the hadoopEnv.properties file to configure functionality such as Sqoop connectivity, environment variables, and ODBC connectivity for the Hive engine.

Open hadoopEnv.properties and back it up before you configure it. You can find the hadoopEnv.properties file in the following location: `<informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>_<version number>/infaConf`

Configure Sqoop Connectivity

Configure the following property for Sqoop connectivity:
infapdo.env.entry.hadoop_node_jdk_home

Configure the HADOOP_NODE_JDK_HOME to represent the directory from which you run the cluster services and the JDK version that the cluster nodes use. You must use JDK version 1.7 or later.

Configure the property as follows:

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=<cluster JDK home>/
jdk<version>
```

For example,

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=/usr/java/default
```

Configure Environment Variables

Add third-party environment variables and extend the existing LD_LIBRARY_PATH environment variable in the hadoopEnv.properties file. The following text shows sample entries to configure environment variables:

```
infapdo.env.entry.oracle_home=ORACLE_HOME=/databases/oracle
infapdo.env.entry.db2_home=DB2_HOME=/databases/db2
infapdo.env.entry.db2instance=DB2INSTANCE=OCA_DB2INSTANCE
infapdo.env.entry.db2codepage=DB2CODEPAGE="1208"
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
infapdo.env.entry.home=HOME=/opt/thirdparty
infapdo.env.entry.gphome_loaders=GPHOME_LOADERS=/databases/greenplum
infapdo.env.entry.pythonpath=PYTHONPATH=$GPHOME_LOADERS/bin/ext
infapdo.env.entry.nz_home=NZ_HOME=/databases/netezza
infapdo.env.entry.ld_library_path=LD_LIBRARY_PATH=$HADOOP_NODE_INFA_HOME/services/
shared/bin:$HADOOP_NODE_INFA_HOME/DataTransformation/bin:$HADOOP_NODE_HADOOP_DIST/lib/
native:$HADOOP_NODE_INFA_HOME/ODBC7.1/lib:$HADOOP_NODE_INFA_HOME/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/jre/lib/amd64/server:$HADOOP_NODE_INFA_HOME/java/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/java/jre/lib/amd64/server:/databases/oracle/lib:/
databases/db2/lib64:$LD_LIBRARY_PATH
infapdo.env.entry.path=PATH=$HADOOP_NODE_HADOOP_DIST/scripts:$HADOOP_NODE_INFA_HOME/
services/shared/bin:$HADOOP_NODE_INFA_HOME/jre/bin:$HADOOP_NODE_INFA_HOME/java/jre/bin:
$HADOOP_NODE_INFA_HOME/ODBC7.1/bin:/databases/oracle/bin:/databases/db2/bin:$PATH
#teradata
infapdo.env.entry.twb_root=TWB_ROOT=/databases/teradata/tbuild
infapdo.env.entry.manpath=MANPATH=/databases/teradata/odbc_64:/databases/teradata/odbc_64
infapdo.env.entry.nlspath=NLSPATH=/databases/teradata/odbc_64/msg/%N:/databases/
teradata/msg/%N
infapdo.env.entry.pwd=PWD=/databases/teradata/odbc_64/samples/C
```

Configure Spark Encryption

Configure the following properties to enable Spark encryption:

spark.shuffle.encryption.enabled

Enables encrypted communication when authentication is enabled.

Set the value to TRUE.

spark.authenticate

Enables authentication for the Spark service on Hadoop.

Set the value to TRUE.

spark.authenticate.enableSaslEncryption

Enables encrypted communication when SASL authentication is enabled.

Set the value to TRUE.

spark.authenticate.sasl.encryption.aes.enabled

Enables AES support when SASL authentication is enabled.

Set the value to TRUE.

Configure ODBC Connectivity for the Hive Engine

Add the following properties in the hadoopEnv.properties file to run mappings with ODBC sources and ODBC targets on the Hive engine:

infapdo.env.entry.odbchome

Specifies the ODBC home directory.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
```

infapdo.env.entry.odbcini

Specifies the path and file name of the odbc.ini file.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbcini=ODBCINI=$HADOOP_NODE_INFA_HOME/ODBC7.1/odbc.ini
```

After you update the hadoopEnv.properties file, you must also manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must 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.
2. Copy 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.

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

Copy .jar Files for Hive Tables on S3

To run mappings on Hive S3, you need to copy .jar files from the master node to the Data Integration Service machine.

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

- emrfs-hadoop-assembly-2.15.0.jar
- hadoop-common-2.7.3-amzn-1.jar
- s3-dist-cp-2.4.0.jar

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

Set S3 Bucket Access Policies

To run mappings on the Spark engine, the Hadoop administrator needs to set S3 bucket access policies to allow any user read and write access.

S3 bucket access policies allow you to control user access to S3 buckets and the actions that users can perform.

1. In the Amazon AWS interface, select the bucket that you want to set policies for.
2. Click **Add bucket policy**. Or, if the bucket already has an access policy, click **Edit bucket policy**.
3. Type the bucket policy to grant read and write access to all users. Then click **Save**.

For example,

```
{
"Version": "<date>",
"Id": "Allow",
"Statement": [
{ "Sid": "<Statement ID>", "Effect": "Allow", "Principal": "*", "Action": "s3:*",
"Resource": [ "arn:aws:s3:::<bucket name>/*", "arn:aws:s3:::<bucket name>" ] }
]
}
```

In the example, Principal is set to `"*"` to grant access to the bucket to all users.

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:

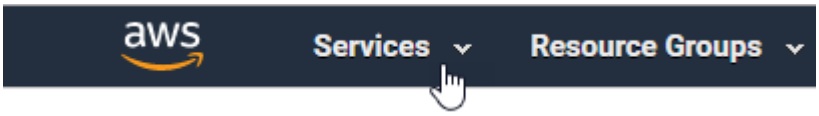
```
{
"Version": "<date>",
"Id": "Allow", "Statement": [
{ "Sid": "<Statement ID>", "Effect": "Allow",
"Principal": {
                "AWS": "arn:aws:iam::<account_2_ID>:<user>"
}
"Action":[
"s3:PutObject","s3:PutObjectAcl",
"s3:GetObject"
]
"Resource": [
"Resource": "arn:aws:s3:::<bucket_1_name>/foldername/*"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

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

When the AWS administrator selects a role for EMR 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:

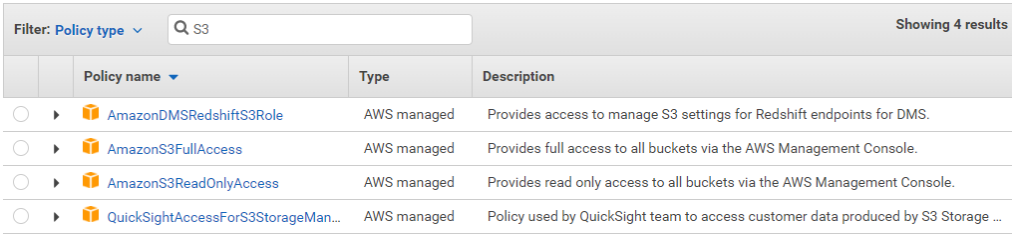
1. In the AWS console, click the **Services** menu.
   The image below shows the **Services** menu in the menu bar:



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:



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

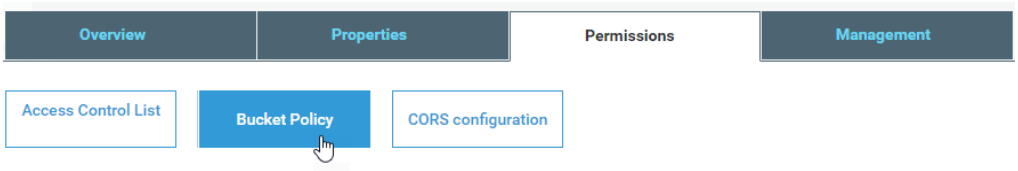
Create an S3 access policy or edit an existing policy. The AWS administrator can enter a JSON statement, based on a template. The administrator can copy and customize the S3 policy from another bucket.

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.

Extract the Cluster Configuration Files

To browse metadata on the Hadoop cluster, the Developer tool requires access to the *-site.xml files. The Informatica administrator generates an archive file that needs to be extracted on each Developer tool machine.

If the Informatica administrator did not extract the archive file to the Developer tool machine, get the file and extract it to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

When you extract the archive file, the Developer tool contains a set of *-site.xml configuration files required for Hadoop access.

Configure developerCore.ini

Edit developerCore.ini to enable communication between the Developer tool and the Hadoop cluster.

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.

Configure the Developer Tool for Kerberos

To import metadata from Hive, HBase, and complex file sources, you need to generate the Kerberos credentials file on the Developer tool machine.

1. Copy krb5.conf from `<Developer tool installation directory>/services/shared/security` to `C:/Windows`.
2. Rename krb5.conf to krb5.ini.
3. In krb5.ini, verify the value of the forwardable option to determine how to use the kinit command.
   - If `forwardable=true`, run the command with the `-f` option.
   - If `forwardable=false`, or if the option is not specified, run the command without the `-f` option.
4. To generate the Kerberos credentials file, run the kinit command from the following location: `<Developer tool installation directory> /clients/java/bin/kinit.exe`
   For example, you might run the following command: `kinit joe/domain12345@MY-REALM`

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:

Update the hadoopEnv.properties file.

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

Replace connections.

If you chose to 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 the hadoopEnv.properties File

When you run the Informatica upgrade, the installer creates a new hadoopEnv.properties file and backs up the existing configuration file.

You need to edit the version 10.2 hadoopEnv.properties file to include any manual configuration that you performed in the previous version for the corresponding distribution.

You can find the backup hadoopEnv.properties file in the following location:

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

You can find the version 10.2 hadoopEnv.properties file in the following location:

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

Replace the Connections

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

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 from Version 10.0 or Later

If you upgraded from version 10.0 or later, and *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

After you upgrade from version 10.0 or later, you need to 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

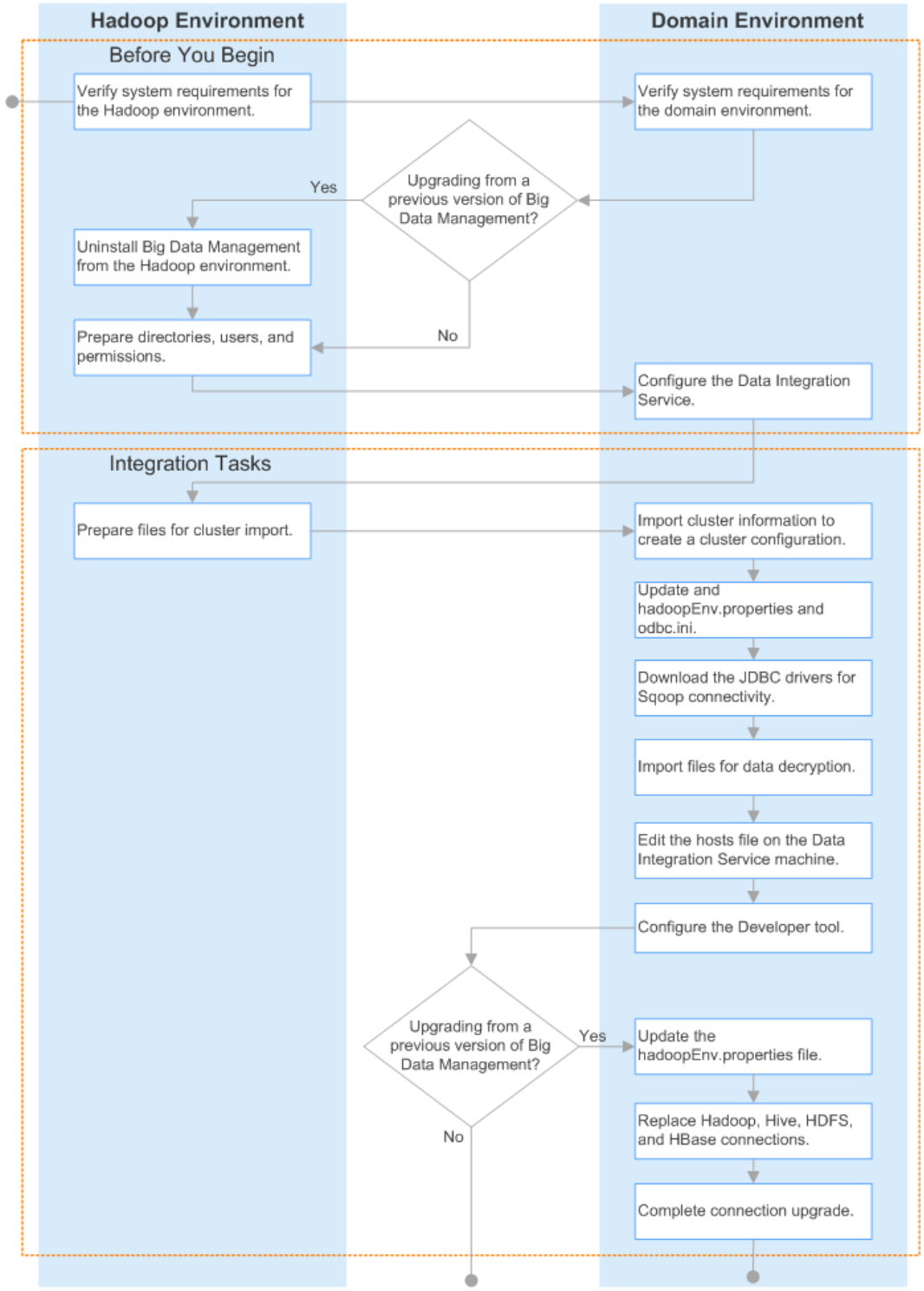
Azure HDInsight Integration Tasks

This chapter includes the following topics:

- Azure HDInsight Integration Task Flow, 46
- Prepare for Cluster Import from Azure HDInsight, 47
- Create a Cluster Configuration, 51
- Update hadoopEnv.properties, 54
- Update odbc.ini, 56
- Download the JDBC Drivers for Sqoop Connectivity, 56
- Import Files for Data Decryption, 57
- Edit the hosts File, 57
- Configure the Developer Tool, 57
- Complete Upgrade Tasks, 58

Azure HDInsight Integration Task Flow

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



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.

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:

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 realm into local names within the Hadoop cluster.

Set to: RULE:[1:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$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.ZooKeeperTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for the Update Strategy transformation.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for the Update Strategy transformation.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for the Update Strategy transformation.

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. Also required for the Update Strategy transformation.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for the Update Strategy transformation.

Set to: TRUE

hive.txn.manager

Turns on transaction support. Required for the Update Strategy transformation.

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

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 engine.

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, Spark, and Hive 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.

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.

The Developer tool requires access to the *-site.xml files for metadata browsing. After you create the cluster configuration, generate an archive file to extract on each Developer tool machine.

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 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.
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.

Property	Description
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.
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.

Generate Cluster Configuration Files

The Developer tool requires configuration files to access cluster metadata at design-time. Generate a cluster configuration archive file and extract it on the Developer tool machine. The archive contains .xml files based on the configuration sets in the cluster configuration.

Important: When you export the cluster configuration, you can export it with sensitive properties or without sensitive properties. When you export without sensitive properties, the sensitive properties are not included in the archive file. When you export with sensitive properties, the sensitive properties are exported with unmasked values. Sensitive properties are required only to access Hive sources on S3 in a non-EMR cluster.

1. From the **Connections** tab, expand the **Cluster Configuration** node in the Domain Navigator.
2. Select the cluster configuration that you want to export.
3. From the Actions menu, choose to export with sensitive properties only to access Hive sources on S3 in a non-EMR cluster. Otherwise, export without sensitive properties.

   The Administrator tool assigns a default name to the archive file using the name of the cluster configuration and a datetime string. For example, when the cluster configuration is named `CC1`: `CC1_2017-07-24-21-39-45.zip`
4. Accept the name or rename it, and then browse to a directory to save the file.

   The Service Manager creates a .zip archive file that contains all properties in the cluster configuration.

Give the .zip file to the mapping developers or copy the .zip file to the Developer tool machines and extract the contents to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

Update hadoopEnv.properties

Update the hadoopEnv.properties file to configure functionality such as Sqoop connectivity, environment variables, and ODBC connectivity for the Hive engine.

Open hadoopEnv.properties and back it up before you configure it. You can find the hadoopEnv.properties file in the following location: `<informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>_<version number>/infaConf`

Configure Sqoop Connectivity

Configure the following property for Sqoop connectivity:
infapdo.env.entry.hadoop_node_jdk_home

Configure the HADOOP_NODE_JDK_HOME to represent the directory from which you run the cluster services and the JDK version that the cluster nodes use. You must use JDK version 1.7 or later.

Configure the property as follows:

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=<cluster JDK home>/
jdk<version>
```

For example,

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=/usr/java/default
```

Configure Environment Variables

Add third-party environment variables and extend the existing LD_LIBRARY_PATH environment variable in the hadoopEnv.properties file. The following text shows sample entries to configure environment variables:

```
infapdo.env.entry.oracle_home=ORACLE_HOME=/databases/oracle
infapdo.env.entry.db2_home=DB2_HOME=/databases/db2
infapdo.env.entry.db2instance=DB2INSTANCE=OCA_DB2INSTANCE
infapdo.env.entry.db2codepage=DB2CODEPAGE="1208"
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
infapdo.env.entry.home=HOME=/opt/thirdparty
infapdo.env.entry.gphome_loaders=GPHOME_LOADERS=/databases/greenplum
infapdo.env.entry.pythonpath=PYTHONPATH=$GPHOME_LOADERS/bin/ext
infapdo.env.entry.nz_home=NZ_HOME=/databases/netezza
infapdo.env.entry.ld_library_path=LD_LIBRARY_PATH=$HADOOP_NODE_INFA_HOME/services/
shared/bin:$HADOOP_NODE_INFA_HOME/DataTransformation/bin:$HADOOP_NODE_HADOOP_DIST/lib/
native:$HADOOP_NODE_INFA_HOME/ODBC7.1/lib:$HADOOP_NODE_INFA_HOME/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/jre/lib/amd64/server:$HADOOP_NODE_INFA_HOME/java/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/java/jre/lib/amd64/server:/databases/oracle/lib:/
databases/db2/lib64:$LD_LIBRARY_PATH
infapdo.env.entry.path=PATH=$HADOOP_NODE_HADOOP_DIST/scripts:$HADOOP_NODE_INFA_HOME/
services/shared/bin:$HADOOP_NODE_INFA_HOME/jre/bin:$HADOOP_NODE_INFA_HOME/java/jre/bin:
$HADOOP_NODE_INFA_HOME/ODBC7.1/bin:/databases/oracle/bin:/databases/db2/bin:$PATH
#teradata
infapdo.env.entry.twb_root=TWB_ROOT=/databases/teradata/tbuild
infapdo.env.entry.manpath=MANPATH=/databases/teradata/odbc_64:/databases/teradata/odbc_64
infapdo.env.entry.nlspath=NLSPATH=/databases/teradata/odbc_64/msg/%N:/databases/
teradata/msg/%N
infapdo.env.entry.pwd=PWD=/databases/teradata/odbc_64/samples/C
```

Configure Spark Encryption

Configure the following properties to enable Spark encryption:

spark.shuffle.encryption.enabled

Enables encrypted communication when authentication is enabled.

Set the value to TRUE.

spark.authenticate

Enables authentication for the Spark service on Hadoop.

Set the value to TRUE.

spark.authenticate.enableSaslEncryption

Enables encrypted communication when SASL authentication is enabled.

Set the value to TRUE.

spark.authenticate.sasl.encryption.aes.enabled

Enables AES support when SASL authentication is enabled.

Set the value to TRUE.

Configure ODBC Connectivity for the Hive Engine

Add the following properties in the hadoopEnv.properties file to run mappings with ODBC sources and ODBC targets on the Hive engine:

infapdo.env.entry.odbchome

Specifies the ODBC home directory.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
```

infapdo.env.entry.odbcini

Specifies the path and file name of the odbc.ini file.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbcini=ODBCINI=$HADOOP_NODE_INFA_HOME/ODBC7.1/odbc.ini
```

After you update the hadoopEnv.properties file, you must also manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must 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.
2. Copy 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.

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

Import Files for Data Decryption

When you run a mapping on the Azure HDInsight cluster that has encrypted data, you must configure files to decrypt data on the Azure HDInsight cluster.

The Data Integration Service host machine requires the following files from the Hadoop cluster to decrypt data:

- key_decryption_cert.prv
- decrypt.sh

Perform the following steps to import the files:

1. Locate these files on the cluster head node.
2. Copy the files from the Hadoop cluster to the machine that runs the Data Integration Service. The directory structure must be the same as that of the Hadoop cluster.
3. Verify that permissions on the directory are 775.

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.

Extract the Cluster Configuration Files

To browse metadata on the Hadoop cluster, the Developer tool requires access to the *-site.xml files. The Informatica administrator generates an archive file that needs to be extracted on each Developer tool machine.

If the Informatica administrator did not extract the archive file to the Developer tool machine, get the file and extract it to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

When you extract the archive file, the Developer tool contains a set of *-site.xml configuration files required for Hadoop access.

Configure developerCore.ini

Edit developerCore.ini to enable communication between the Developer tool and the Hadoop cluster.

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.

Configure the Developer Tool for Kerberos

To import metadata from Hive, HBase, and complex file sources, you need to generate the Kerberos credentials file on the Developer tool machine.

1. Copy krb5.conf from `<Developer tool installation directory>/services/shared/security` to `C:/Windows`.
2. Rename krb5.conf to krb5.ini.
3. In krb5.ini, verify the value of the forwardable option to determine how to use the kinit command.
   - If `forwardable=true`, run the command with the `-f` option.
   - If `forwardable=false`, or if the option is not specified, run the command without the `-f` option.
4. To generate the Kerberos credentials file, run the kinit command from the following location: `<Developer tool installation directory> /clients/java/bin/kinit.exe`
   For example, you might run the following command: `kinit joe/domain12345@MY-REALM`

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:

Update the hadoopEnv.properties file.

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

Replace connections.

If you chose to 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 the hadoopEnv.properties File

When you run the Informatica upgrade, the installer creates a new hadoopEnv.properties file and backs up the existing configuration file.

You need to edit the version 10.2 hadoopEnv.properties file to include any manual configuration that you performed in the previous version for the corresponding distribution.

You can find the backup hadoopEnv.properties file in the following location:

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

You can find the version 10.2 hadoopEnv.properties file in the following location:

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

Replace the Connections

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

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 from Version 10.0 or Later

If you upgraded from version 10.0 or later, and *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

After you upgrade from version 10.0 or later, you need to 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

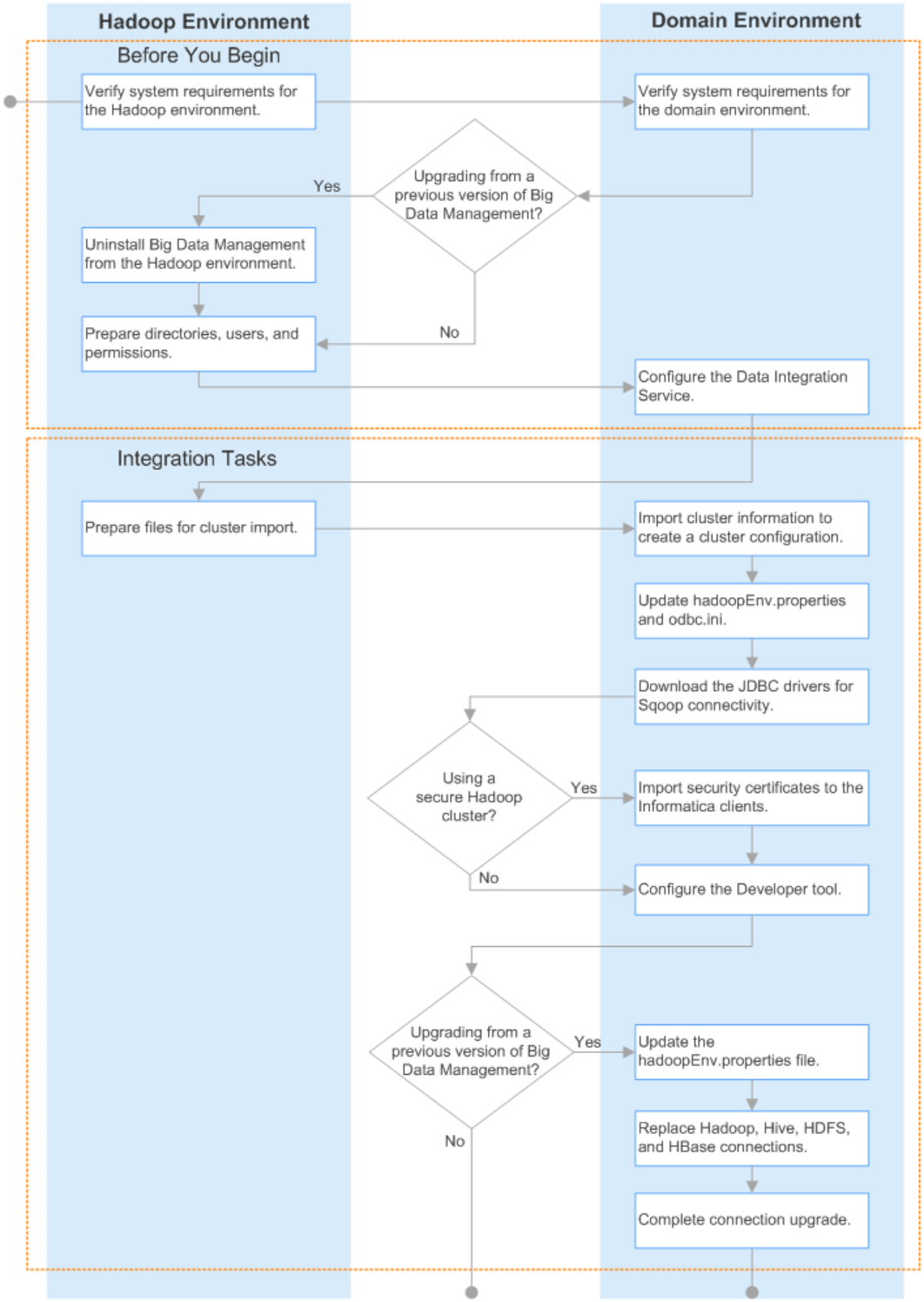
Cloudera CDH Integration Tasks

This chapter includes the following topics:

- Cloudera CDH Integration Task Flow, 62
- Prepare for Cluster Import from Cloudera CDH, 63
- Create a Cluster Configuration, 67
- Update hadoopEnv.properties, 70
- Update odbc.ini, 71
- Download the JDBC Drivers for Sqoop Connectivity, 72
- Import Security Certificates to Clients, 72
- Configure the Developer Tool, 73
- Complete Upgrade Tasks, 74

Cloudera CDH Integration Task Flow

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



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.

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 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.

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.

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 realm into local names within the Hadoop cluster.

Set to: RULE:[1:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$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

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 engine.

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.

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.

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.

The Developer tool requires access to the *-site.xml files for metadata browsing. After you create the cluster configuration, generate an archive file to extract on each Developer tool machine.

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 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.
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.
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.

Generate Cluster Configuration Files

The Developer tool requires configuration files to access cluster metadata at design-time. Generate a cluster configuration archive file and extract it on the Developer tool machine. The archive contains .xml files based on the configuration sets in the cluster configuration.

Important: When you export the cluster configuration, you can export it with sensitive properties or without sensitive properties. When you export without sensitive properties, the sensitive properties are not included in the archive file. When you export with sensitive properties, the sensitive properties are exported with unmasked values. Sensitive properties are required only to access Hive sources on S3 in a non-EMR cluster.

1. From the **Connections** tab, expand the **Cluster Configuration** node in the Domain Navigator.
2. Select the cluster configuration that you want to export.
3. From the Actions menu, choose to export with sensitive properties only to access Hive sources on S3 in a non-EMR cluster. Otherwise, export without sensitive properties.

The Administrator tool assigns a default name to the archive file using the name of the cluster configuration and a datetime string. For example, when the cluster configuration is named `CC1`: `CC1_2017-07-24-21-39-45.zip`

4. Accept the name or rename it, and then browse to a directory to save the file.

   The Service Manager creates a .zip archive file that contains all properties in the cluster configuration.

Give the .zip file to the mapping developers or copy the .zip file to the Developer tool machines and extract the contents to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

Update hadoopEnv.properties

Update the hadoopEnv.properties file to configure functionality such as Sqoop connectivity, environment variables, and ODBC connectivity for the Hive engine.

Open hadoopEnv.properties and back it up before you configure it. You can find the hadoopEnv.properties file in the following location: `<informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>_<version number>/infaConf`

Configure Sqoop Connectivity

Configure the following property for Sqoop connectivity:
infapdo.env.entry.hadoop_node_jdk_home

Configure the HADOOP_NODE_JDK_HOME to represent the directory from which you run the cluster services and the JDK version that the cluster nodes use. You must use JDK version 1.7 or later.

Configure the property as follows:

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=<cluster JDK home>/
jdk<version>
```

For example,

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=/usr/java/default
```

Configure Environment Variables

Add third-party environment variables and extend the existing LD_LIBRARY_PATH environment variable in the hadoopEnv.properties file. The following text shows sample entries to configure environment variables:

```
infapdo.env.entry.oracle_home=ORACLE_HOME=/databases/oracle
infapdo.env.entry.db2_home=DB2_HOME=/databases/db2
infapdo.env.entry.db2instance=DB2INSTANCE=OCA_DB2INSTANCE
infapdo.env.entry.db2codepage=DB2CODEPAGE="1208"
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
infapdo.env.entry.home=HOME=/opt/thirdparty
infapdo.env.entry.gphome_loaders=GPHOME_LOADERS=/databases/greenplum
infapdo.env.entry.pythonpath=PYTHONPATH=$GPHOME_LOADERS/bin/ext
infapdo.env.entry.nz_home=NZ_HOME=/databases/netezza
infapdo.env.entry.ld_library_path=LD_LIBRARY_PATH=$HADOOP_NODE_INFA_HOME/services/
shared/bin:$HADOOP_NODE_INFA_HOME/DataTransformation/bin:$HADOOP_NODE_HADOOP_DIST/lib/
native:$HADOOP_NODE_INFA_HOME/ODBC7.1/lib:$HADOOP_NODE_INFA_HOME/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/jre/lib/amd64/server:$HADOOP_NODE_INFA_HOME/java/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/java/jre/lib/amd64/server:/databases/oracle/lib:/
databases/db2/lib64:$LD_LIBRARY_PATH
infapdo.env.entry.path=PATH=$HADOOP_NODE_HADOOP_DIST/scripts:$HADOOP_NODE_INFA_HOME/
services/shared/bin:$HADOOP_NODE_INFA_HOME/jre/bin:$HADOOP_NODE_INFA_HOME/java/jre/bin:
$HADOOP_NODE_INFA_HOME/ODBC7.1/bin:/databases/oracle/bin:/databases/db2/bin:$PATH
#teradata
infapdo.env.entry.twb_root=TWB_ROOT=/databases/teradata/tbuild
```

```
infapdo.env.entry.manpath=MANPATH=/databases/teradata/odbc_64:/databases/teradata/odbc_64
infapdo.env.entry.nlspath=NLSPATH=/databases/teradata/odbc_64/msg/%N:/databases/
teradata/msg/%N
infapdo.env.entry.pwd=PWD=/databases/teradata/odbc_64/samples/C
```

Configure Spark Encryption

Configure the following properties to enable Spark encryption:

spark.shuffle.encryption.enabled

Enables encrypted communication when authentication is enabled.

Set the value to TRUE.

spark.authenticate

Enables authentication for the Spark service on Hadoop.

Set the value to TRUE.

spark.authenticate.enableSaslEncryption

Enables encrypted communication when SASL authentication is enabled.

Set the value to TRUE.

spark.authenticate.sasl.encryption.aes.enabled

Enables AES support when SASL authentication is enabled.

Set the value to TRUE.

Configure ODBC Connectivity for the Hive Engine

Add the following properties in the hadoopEnv.properties file to run mappings with ODBC sources and ODBC targets on the Hive engine:

infapdo.env.entry.odbchome

Specifies the ODBC home directory.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
```

infapdo.env.entry.odbcini

Specifies the path and file name of the odbc.ini file.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbcini=ODBCINI=$HADOOP_NODE_INFA_HOME/ODBC7.1/odbc.ini
```

After you update the hadoopEnv.properties file, you must also manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must 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.
2. Copy 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.

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

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
   ```

   - If the domain is SSL-enabled, import the certificate alias file to the following locations: `<Informatica installation directory>\clients\DeveloperClient\clients\shared\security\infa_truststore.jks`
     - The following path on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperTool\clients\shared\security\infa_truststore.jks`
     - The following path on the machine that hosts the Data Integration Service: `<Informatica installation directory>/services/shared/security/infa_truststore.jks`
   - If the domain is not SSL-enabled, import the certificate alias file to the following locations:
     - The following path on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperClient\clients\java\jre\lib\security\cacerts`
     - The following path on the machine that hosts the Data Integration Service: `<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.

Extract the Cluster Configuration Files

To browse metadata on the Hadoop cluster, the Developer tool requires access to the *-site.xml files. The Informatica administrator generates an archive file that needs to be extracted on each Developer tool machine.

If the Informatica administrator did not extract the archive file to the Developer tool machine, get the file and extract it to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

When you extract the archive file, the Developer tool contains a set of *-site.xml configuration files required for Hadoop access.

Configure developerCore.ini

Edit developerCore.ini to enable communication between the Developer tool and the Hadoop cluster.

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.

Configure the Developer Tool for Kerberos

To import metadata from Hive, HBase, and complex file sources, you need to generate the Kerberos credentials file on the Developer tool machine.

1. Copy krb5.conf from `<Developer tool installation directory>/services/shared/security` to `C:/Windows`.
2. Rename krb5.conf to krb5.ini.
3. In krb5.ini, verify the value of the forwardable option to determine how to use the kinit command.
   - If `forwardable=true`, run the command with the `-f` option.
   - If `forwardable=false`, or if the option is not specified, run the command without the `-f` option.
4. To generate the Kerberos credentials file, run the kinit command from the following location: `<Developer tool installation directory> /clients/java/bin/kinit.exe`
   For example, you might run the following command: `kinit joe/domain12345@MY-REALM`

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:

Update the hadoopEnv.properties file.

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

Replace connections.

If you chose to 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 the hadoopEnv.properties File

When you run the Informatica upgrade, the installer creates a new hadoopEnv.properties file and backs up the existing configuration file.

You need to edit the version 10.2 hadoopEnv.properties file to include any manual configuration that you performed in the previous version for the corresponding distribution.

You can find the backup hadoopEnv.properties file in the following location:

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

You can find the version 10.2 hadoopEnv.properties file in the following location:

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

Replace the Connections

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

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 from Version 10.0 or Later

If you upgraded from version 10.0 or later, and *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

After you upgrade from version 10.0 or later, you need to 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*.

Complete Connection Upgrade from Version 9.6.1

Effective in version 10.0, Big Data Management requires a Hadoop connection to run mappings on the Hadoop cluster. If you upgraded from 9.6.1 or any 9.6.1 hotfix, you must generate Hadoop connections from Hive connections that are enabled to run mappings in the Hadoop environment.

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. Generate Hadoop connections from Hive connections if the following conditions are true:

- You upgraded from 9.6.1 or any 9.6.1 hotfix.

- The Hive connections are configured to run mappings in the Hadoop environment.

Complete the following tasks to upgrade connections:

Generate Hadoop connections

Effective in version 10.0, Big Data Management requires a Hadoop connection to run mappings in the Hadoop environment. You must generate Hadoop connections from Hive connections that are enabled 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*.

CHAPTER 6

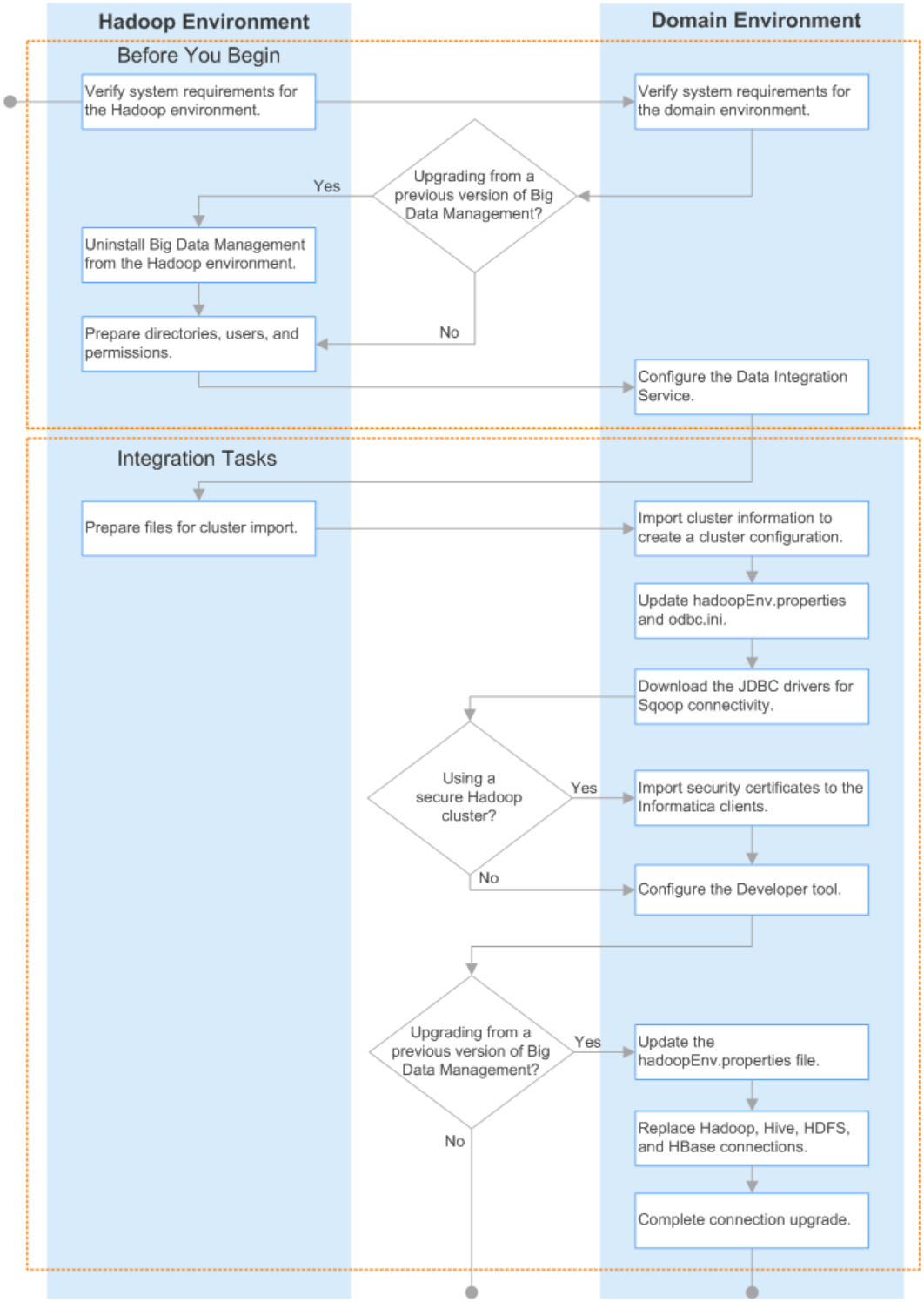
Hortonworks HDP Integration Tasks

This chapter includes the following topics:

- Hortonworks HDP Integration Task Flow, 78
- Prepare for Cluster Import from Hortonworks HDP, 79
- Create a Cluster Configuration, 84
- Update hadoopEnv.properties, 87
- Update odbc.ini, 88
- Download the JDBC Drivers for Sqoop Connectivity, 89
- Import Security Certificates to Clients, 89
- Configure the Developer Tool, 90
- Complete Upgrade Tasks, 91

Hortonworks HDP Integration Task Flow

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



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.

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 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.

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.

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 realm into local names within the Hadoop cluster.

Set to: RULE:[1:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$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.ZooKeeperTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for the Update Strategy transformation.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for the Update Strategy transformation.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for the Update Strategy transformation.

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. Also required for the Update Strategy transformation.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for the Update Strategy transformation.

Set to: TRUE

hive.txn.manager

Turns on transaction support. Required for the Update Strategy transformation.

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

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 engine.

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, Spark, and Hive 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

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>
```

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.

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.

The Developer tool requires access to the *-site.xml files for metadata browsing. After you create the cluster configuration, generate an archive file to extract on each Developer tool machine.

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 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.
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.
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.

Generate Cluster Configuration Files

The Developer tool requires configuration files to access cluster metadata at design-time. Generate a cluster configuration archive file and extract it on the Developer tool machine. The archive contains .xml files based on the configuration sets in the cluster configuration.

Important: When you export the cluster configuration, you can export it with sensitive properties or without sensitive properties. When you export without sensitive properties, the sensitive properties are not included in the archive file. When you export with sensitive properties, the sensitive properties are exported with unmasked values. Sensitive properties are required only to access Hive sources on S3 in a non-EMR cluster.

1. From the **Connections** tab, expand the **Cluster Configuration** node in the Domain Navigator.
2. Select the cluster configuration that you want to export.
3. From the Actions menu, choose to export with sensitive properties only to access Hive sources on S3 in a non-EMR cluster. Otherwise, export without sensitive properties.

   The Administrator tool assigns a default name to the archive file using the name of the cluster configuration and a datetime string. For example, when the cluster configuration is named `CC1`: `CC1_2017-07-24-21-39-45.zip`
4. Accept the name or rename it, and then browse to a directory to save the file.

   The Service Manager creates a .zip archive file that contains all properties in the cluster configuration.

Give the .zip file to the mapping developers or copy the .zip file to the Developer tool machines and extract the contents to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

Update hadoopEnv.properties

Update the hadoopEnv.properties file to configure functionality such as Sqoop connectivity, environment variables, and ODBC connectivity for the Hive engine.

Open hadoopEnv.properties and back it up before you configure it. You can find the hadoopEnv.properties file in the following location: `<informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>_<version number>/infaConf`

Configure Sqoop Connectivity

Configure the following property for Sqoop connectivity:
infapdo.env.entry.hadoop_node_jdk_home

Configure the HADOOP_NODE_JDK_HOME to represent the directory from which you run the cluster services and the JDK version that the cluster nodes use. You must use JDK version 1.7 or later.

Configure the property as follows:

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=<cluster JDK home>/
jdk<version>
```

For example,

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=/usr/java/default
```

Configure Environment Variables

Add third-party environment variables and extend the existing LD_LIBRARY_PATH environment variable in the hadoopEnv.properties file. The following text shows sample entries to configure environment variables:

```
infapdo.env.entry.oracle_home=ORACLE_HOME=/databases/oracle
infapdo.env.entry.db2_home=DB2_HOME=/databases/db2
infapdo.env.entry.db2instance=DB2INSTANCE=OCA_DB2INSTANCE
infapdo.env.entry.db2codepage=DB2CODEPAGE="1208"
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
infapdo.env.entry.home=HOME=/opt/thirdparty
infapdo.env.entry.gphome_loaders=GPHOME_LOADERS=/databases/greenplum
infapdo.env.entry.pythonpath=PYTHONPATH=$GPHOME_LOADERS/bin/ext
infapdo.env.entry.nz_home=NZ_HOME=/databases/netezza
infapdo.env.entry.ld_library_path=LD_LIBRARY_PATH=$HADOOP_NODE_INFA_HOME/services/
shared/bin:$HADOOP_NODE_INFA_HOME/DataTransformation/bin:$HADOOP_NODE_HADOOP_DIST/lib/
native:$HADOOP_NODE_INFA_HOME/ODBC7.1/lib:$HADOOP_NODE_INFA_HOME/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/jre/lib/amd64/server:$HADOOP_NODE_INFA_HOME/java/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/java/jre/lib/amd64/server:/databases/oracle/lib:/
databases/db2/lib64:$LD_LIBRARY_PATH
infapdo.env.entry.path=PATH=$HADOOP_NODE_HADOOP_DIST/scripts:$HADOOP_NODE_INFA_HOME/
services/shared/bin:$HADOOP_NODE_INFA_HOME/jre/bin:$HADOOP_NODE_INFA_HOME/java/jre/bin:
$HADOOP_NODE_INFA_HOME/ODBC7.1/bin:/databases/oracle/bin:/databases/db2/bin:$PATH
#teradata
infapdo.env.entry.twb_root=TWB_ROOT=/databases/teradata/tbuild
infapdo.env.entry.manpath=MANPATH=/databases/teradata/odbc_64:/databases/teradata/odbc_64
infapdo.env.entry.nlspath=NLSPATH=/databases/teradata/odbc_64/msg/%N:/databases/
teradata/msg/%N
infapdo.env.entry.pwd=PWD=/databases/teradata/odbc_64/samples/C
```

Configure Spark Encryption

Configure the following properties to enable Spark encryption:
spark.shuffle.encryption.enabled

Enables encrypted communication when authentication is enabled.

Set the value to TRUE.

spark.authenticate

Enables authentication for the Spark service on Hadoop.

Set the value to TRUE.

spark.authenticate.enableSaslEncryption

Enables encrypted communication when SASL authentication is enabled.

Set the value to TRUE.

spark.authenticate.sasl.encryption.aes.enabled

Enables AES support when SASL authentication is enabled.

Set the value to TRUE.

Configure ODBC Connectivity for the Hive Engine

Add the following properties in the hadoopEnv.properties file to run mappings with ODBC sources and ODBC targets on the Hive engine:

infapdo.env.entry.odbchome

Specifies the ODBC home directory.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
```

infapdo.env.entry.odbcini

Specifies the path and file name of the odbc.ini file.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbcini=ODBCINI=$HADOOP_NODE_INFA_HOME/ODBC7.1/odbc.ini
```

After you update the hadoopEnv.properties file, you must also manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must 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.
2. Copy 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.

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

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
   ```

   - If the domain is SSL-enabled, import the certificate alias file to the following locations: `<Informatica installation directory>\clients\DeveloperClient\clients\shared\security\infa_truststore.jks`
     - The following path on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperTool\clients\shared\security\infa_truststore.jks`
     - The following path on the machine that hosts the Data Integration Service: `<Informatica installation directory>/services/shared/security/infa_truststore.jks`

- If the domain is not SSL-enabled, import the certificate alias file to the following locations:
  - The following path on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperClient\clients\java\jre\lib\security\cacerts`
  - The following path on the machine that hosts the Data Integration Service: `<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.

Extract the Cluster Configuration Files

To browse metadata on the Hadoop cluster, the Developer tool requires access to the *-site.xml files. The Informatica administrator generates an archive file that needs to be extracted on each Developer tool machine.

If the Informatica administrator did not extract the archive file to the Developer tool machine, get the file and extract it to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

When you extract the archive file, the Developer tool contains a set of *-site.xml configuration files required for Hadoop access.

Configure developerCore.ini

Edit developerCore.ini to enable communication between the Developer tool and the Hadoop cluster.

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.

Configure the Developer Tool for Kerberos

To import metadata from Hive, HBase, and complex file sources, you need to generate the Kerberos credentials file on the Developer tool machine.

1. Copy krb5.conf from `<Developer tool installation directory>/services/shared/security` to `C:/Windows`.
2. Rename krb5.conf to krb5.ini.
3. In krb5.ini, verify the value of the forwardable option to determine how to use the kinit command.
   - If `forwardable=true`, run the command with the `-f` option.
   - If `forwardable=false`, or if the option is not specified, run the command without the `-f` option.

4. To generate the Kerberos credentials file, run the kinit command from the following location: `<Developer tool installation directory> /clients/java/bin/kinit.exe`
   For example, you might run the following command: `kinit joe/domain12345@MY-REALM`

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:

Update the hadoopEnv.properties file.

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

Replace connections.

If you chose to 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 the hadoopEnv.properties File

When you run the Informatica upgrade, the installer creates a new hadoopEnv.properties file and backs up the existing configuration file.

You need to edit the version 10.2 hadoopEnv.properties file to include any manual configuration that you performed in the previous version for the corresponding distribution.

You can find the backup hadoopEnv.properties file in the following location:

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

You can find the version 10.2 hadoopEnv.properties file in the following location:

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

Replace the Connections

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

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 from Version 10.0 or Later

If you upgraded from version 10.0 or later, and *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

After you upgrade from version 10.0 or later, you need to 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*.

Complete Connection Upgrade from Version 9.6.1

Effective in version 10.0, Big Data Management requires a Hadoop connection to run mappings on the Hadoop cluster. If you upgraded from 9.6.1 or any 9.6.1 hotfix, you must generate Hadoop connections from Hive connections that are enabled to run mappings in the Hadoop environment.

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. Generate Hadoop connections from Hive connections if the following conditions are true:

- You upgraded from 9.6.1 or any 9.6.1 hotfix.
- The Hive connections are configured to run mappings in the Hadoop environment.

Complete the following tasks to upgrade connections:

Generate Hadoop connections

Effective in version 10.0, Big Data Management requires a Hadoop connection to run mappings in the Hadoop environment. You must generate Hadoop connections from Hive connections that are enabled 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. The generated Hadoop connection uses the MRV2 as the engine type. If you use Tez to run mappings, you must update the engine type.
5. 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*.

CHAPTER 7

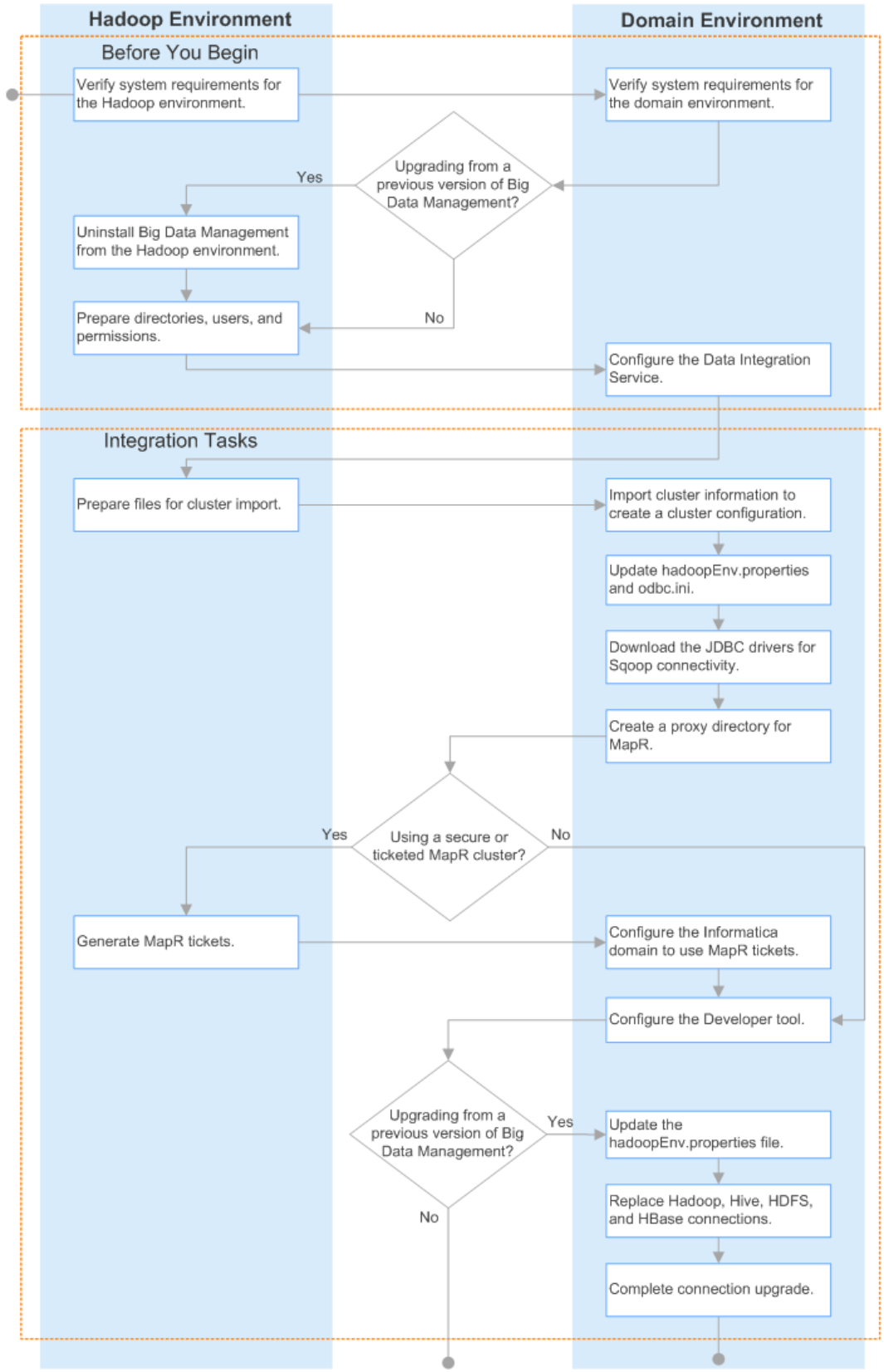
MapR Integration Tasks

This chapter includes the following topics:

- MapR Integration Task Flow, 95
- Prepare for Cluster Import from MapR, 97
- Install the MapR Client and Configure Environment Path, 101
- Create a Cluster Configuration, 101
- Update hadoopEnv.properties, 103
- Update odbc.ini, 104
- Download the JDBC Drivers for Sqoop Connectivity, 105
- Create a Proxy Directory for MapR, 105
- Generate MapR Tickets, 105
- Get MapR Configuration Files for the Domain, 108
- Configure the Developer Tool, 109
- Complete Upgrade Tasks, 110

MapR Integration Task Flow

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.
Upgrading from a previous version of Big Data Management?
Yes
Uninstall Big Data Management from the Hadoop environment.
No
Prepare directories, users, and permissions.
Configure the Data Integration Service.
Integration Tasks
Prepare files for cluster import.
Import cluster information to create a cluster configuration.
Update hadoopEnv.properties and odbc.ini.
Download the JDBC drivers for Sqoop connectivity.
Create a proxy directory for MapR.
Using a secure or ticketed MapR cluster?
Yes
No
Generate MapR tickets.
Configure the Informatica domain to use MapR tickets.
Configure the Developer tool.
Upgrading from a previous version of Big Data Management?
Yes
Update the hadoopEnv.properties file.
No
Replace Hadoop, Hive, HDFS, and HBase connections.
Complete connection upgrade.

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.

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 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.

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.

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 realm into local names within the Hadoop cluster.

Set to: RULE:[1:$1@$0](^.*@INFA-AD-REALM$)s/^(.*)@INFA-AD-REALM$/$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.ZooKeeperTokenStore

hive.compactor.initiator.on

Runs the initiator and cleaner threads on metastore instance. Required for the Update Strategy transformation.

Set to: TRUE

hive.compactor.worker.threads

The number of worker threads to run in a metastore instance. Required for the Update Strategy transformation.

Set to: 1

hive.enforce.bucketing

Enables dynamic bucketing while loading to Hive. Required for the Update Strategy transformation.

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. Also required for the Update Strategy transformation.

Set to: nonstrict

hive.support.concurrency

Enables table locking in Hive. Required for the Update Strategy transformation.

Set to: TRUE

hive.txn.manager

Turns on transaction support. Required for the Update Strategy transformation.

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

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 engine.

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.

Install the MapR Client and Configure Environment Path

To run mappings on a MapR 5.2 with MEP 3.x cluster, you must install the MapR client and then configure environment settings on the Informatica domain machine.

To install and configure the MapR client, see MapR documentation at http://doc.mapr.com/display/MapR/Setting+Up+the+Client#SettingUptheClient-client.

After you install and configure the MapR client, configure the following properties on the machines that host domain services:

Property	Description
MAPR_HOME	Home directory for the MapR client. For example: `/opt/mapr` Configure the MAPR_HOME property in the Environment Variables section of the Data Integration Service.

Create a Cluster Configuration

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

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.

The Developer tool requires access to the *-site.xml files for metadata browsing. After you create the cluster configuration, generate an archive file to extract on each Developer tool machine.

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.
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.

Generate Cluster Configuration Files

The Developer tool requires configuration files to access cluster metadata at design-time. Generate a cluster configuration archive file and extract it on the Developer tool machine. The archive contains .xml files based on the configuration sets in the cluster configuration.

Important: When you export the cluster configuration, you can export it with sensitive properties or without sensitive properties. When you export without sensitive properties, the sensitive properties are not included in the archive file. When you export with sensitive properties, the sensitive properties are exported with unmasked values. Sensitive properties are required only to access Hive sources on S3 in a non-EMR cluster.

1. From the **Connections** tab, expand the **Cluster Configuration** node in the Domain Navigator.
2. Select the cluster configuration that you want to export.
3. From the Actions menu, choose to export with sensitive properties only to access Hive sources on S3 in a non-EMR cluster. Otherwise, export without sensitive properties.

   The Administrator tool assigns a default name to the archive file using the name of the cluster configuration and a datetime string. For example, when the cluster configuration is named `CC1`: `CC1_2017-07-24-21-39-45.zip`
4. Accept the name or rename it, and then browse to a directory to save the file.

   The Service Manager creates a .zip archive file that contains all properties in the cluster configuration.

Give the .zip file to the mapping developers or copy the .zip file to the Developer tool machines and extract the contents to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

Update hadoopEnv.properties

Update the hadoopEnv.properties file to configure functionality such as Sqoop connectivity, environment variables, and ODBC connectivity for the Hive engine.

Open hadoopEnv.properties and back it up before you configure it. You can find the hadoopEnv.properties file in the following location: `<informatica installation directory>/services/shared/hadoop/<Hadoop distribution name>_<version number>/infaConf`

Configure Sqoop Connectivity

Configure the following property for Sqoop connectivity:
infapdo.env.entry.hadoop_node_jdk_home

Configure the HADOOP_NODE_JDK_HOME to represent the directory from which you run the cluster services and the JDK version that the cluster nodes use. You must use JDK version 1.7 or later.

Configure the property as follows:

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=<cluster JDK home>/
jdk<version>
```

For example,

```
infapdo.env.entry.hadoop_node_jdk_home=HADOOP_NODE_JDK_HOME=/usr/java/default
```

Configure Environment Variables

Add third-party environment variables and extend the existing LD_LIBRARY_PATH environment variable in the hadoopEnv.properties file. The following text shows sample entries to configure environment variables:

```
infapdo.env.entry.oracle_home=ORACLE_HOME=/databases/oracle
infapdo.env.entry.db2_home=DB2_HOME=/databases/db2
infapdo.env.entry.db2instance=DB2INSTANCE=OCA_DB2INSTANCE
infapdo.env.entry.db2codepage=DB2CODEPAGE="1208"
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
infapdo.env.entry.home=HOME=/opt/thirdparty
infapdo.env.entry.gphome_loaders=GPHOME_LOADERS=/databases/greenplum
infapdo.env.entry.pythonpath=PYTHONPATH=$GPHOME_LOADERS/bin/ext
infapdo.env.entry.nz_home=NZ_HOME=/databases/netezza
infapdo.env.entry.ld_library_path=LD_LIBRARY_PATH=$HADOOP_NODE_INFA_HOME/services/
shared/bin:$HADOOP_NODE_INFA_HOME/DataTransformation/bin:$HADOOP_NODE_HADOOP_DIST/lib/
native:$HADOOP_NODE_INFA_HOME/ODBC7.1/lib:$HADOOP_NODE_INFA_HOME/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/jre/lib/amd64/server:$HADOOP_NODE_INFA_HOME/java/jre/lib/
amd64:$HADOOP_NODE_INFA_HOME/java/jre/lib/amd64/server:/databases/oracle/lib:/
databases/db2/lib64:$LD_LIBRARY_PATH
infapdo.env.entry.path=PATH=$HADOOP_NODE_HADOOP_DIST/scripts:$HADOOP_NODE_INFA_HOME/
services/shared/bin:$HADOOP_NODE_INFA_HOME/jre/bin:$HADOOP_NODE_INFA_HOME/java/jre/bin:
$HADOOP_NODE_INFA_HOME/ODBC7.1/bin:/databases/oracle/bin:/databases/db2/bin:$PATH
#teradata
infapdo.env.entry.twb_root=TWB_ROOT=/databases/teradata/tbuild
infapdo.env.entry.manpath=MANPATH=/databases/teradata/odbc_64:/databases/teradata/odbc_64
infapdo.env.entry.nlspath=NLSPATH=/databases/teradata/odbc_64/msg/%N:/databases/
teradata/msg/%N
infapdo.env.entry.pwd=PWD=/databases/teradata/odbc_64/samples/C
```

Configure Spark Encryption

Configure the following properties to enable Spark encryption:
spark.shuffle.encryption.enabled

Enables encrypted communication when authentication is enabled.

Set the value to TRUE.

spark.authenticate

Enables authentication for the Spark service on Hadoop.

Set the value to TRUE.

spark.authenticate.enableSaslEncryption

Enables encrypted communication when SASL authentication is enabled.

Set the value to TRUE.

spark.authenticate.sasl.encryption.aes.enabled

Enables AES support when SASL authentication is enabled.

Set the value to TRUE.

Configure ODBC Connectivity for the Hive Engine

Add the following properties in the hadoopEnv.properties file to run mappings with ODBC sources and ODBC targets on the Hive engine:

infapdo.env.entry.odbchome

Specifies the ODBC home directory.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbchome=ODBCHOME=$HADOOP_NODE_INFA_HOME/ODBC7.1
```

infapdo.env.entry.odbcini

Specifies the path and file name of the odbc.ini file.

Set the value of the infapdo.env.entry.odbchome property to

```
infapdo.env.entry.odbcini=ODBCINI=$HADOOP_NODE_INFA_HOME/ODBC7.1/odbc.ini
```

After you update the hadoopEnv.properties file, you must also manually edit the odbc.ini file to replace the absolute driver paths with relative driver paths.

Update odbc.ini

Before you run mappings with ODBC sources and ODBC targets on the Hive engine, you must 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.
2. Copy 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.

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

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 following users:

Data Integration Service User

The Data Integration Service user requires an account on the MapR cluster and a MapR ticket on the Data Integration Service machine.

When the MapR cluster uses both Kerberos and Ticket authentication, you generate a ticket for the Data Integration Service user for each authentication system.

Developer Tool User

Create and configure an account for the Developer tool user on every node in the cluster. The Developer tool user requires an account on the MapR cluster and a MapR ticket on the machine where the Developer tool is installed.

After you generate and save MapR tickets, you perform additional steps to configure application services to communicate with the MapR cluster.

Generate Tickets

After you create a MapR user account for the Data Integration Service user and the Developer tool user, generate a MapR ticket for each user and save it to a local directory.

To generate a MapR ticket, refer to the MapR documentation.

Data Integration Service User Ticket

Generate a MapR ticket for the Data Integration Service user. Name the ticket file using the following naming convention:

```
maprticket_<user name>
```

Save the ticket file in the `/tmp` directory of the machine that runs the Data Integration Service.

When the MapR cluster is configured to enable a user to use Kerberos authentication and MapR Ticket authentication, you generate a MapR ticket file for the user for each authentication mode. Save one ticket file in /tmp. Save the other ticket file in any directory on the Data Integration Service machine, and create the environment variable, MAPR_TICKETFILE_LOCATION, in the Data Integration Service Process properties.

Developer Tool User Ticket

Generate a MapR ticket for the Developer tool user. Name the ticket file using the following naming convention:

```
maprticket_<user name>
```

Save the ticket file in the `%TEMP%` directory of the machine the runs the Developer tool.

Configure the Data Integration Service

When the MapR cluster is secured with MapR Kerberos authentication, edit Data Integration Service properties to enable communication between the Informatica domain and the cluster.

Data Integration Service Process Properties

In the Administrator tool Domain Navigator, select the Data Integration Service to configure, and then select the **Processes** tab.

In the **Custom Properties** area, define the following properties and values:

Property	Value
ExecutionContextOptions.JVMOption	`-Djava.security.krb5.conf=<Informatica installation directory>/services/shared/security/krb5.conf`
ExecutionContextOptions.JVMOption2	`-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`. For example, `-Dhadoop.login=hybrid`
ExecutionContextOptions.JVMOption7	`-Dhttps.protocols=TLSv1.2`

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	Hadoop distribution directory location on the machine that runs the Data Integration Service. For example, `<Informatica installation directory>/services/shared/hadoop/mapr_<version>` **Note:** Do not configure MAPR_HOME if the MapR client is installed on the machine that runs the Data Integration Service.
MAPR_TICKETFILE_LOCATION	Required when the MapR cluster is ticketed. Directory where an additional MapR Ticket file is stored on the machine that runs the Data Integration Service. When the MapR cluster is configured to enable a user to use Kerberos authentication and MapR Ticket authentication, generate a MapR ticket file for the user for each authentication mode. Save one ticket file in `/tmp`. Save the other ticket file in any directory on the Data Integration Service machine, and provide the location as the value for this property. For example, for a user id 1234, save a MapR ticket file named like `maprticket_1234` in `/tmp`, and save another MapR ticket file named like `maprticket_1234` in the MAPR_TICKETFILE_LOCATION. **Note:** The ticket files can have the same or different names. You must generate the MapR ticket files separately and save one to the MAPR_TICKETFILE_LOCATION.

Changes take effect when you restart the Data Integration 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_HOME	Hadoop distribution directory location on the machine that runs the Data Integration Service. For example, `<Informatica installation directory>/services/shared/hadoop/mapr_<version>`

Property	Value
MAPR_TICKETFILE_LOCATION	Directory where the MapR Ticket file is stored on the machine that runs the Analyst Service. 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.

Get MapR Configuration Files for the Domain

If MapR runs on a secure or ticketed cluster, the Data Integration Service requires some MapR configuration files.

Get the following files from the Hadoop administrator:

- mapr-clusters.conf
- mapr.login.conf
- ssl_truststore

Put the files in the following location on the Data Integration Service machine: `<Informatica installation directory>/services/shared/hadoop/mapr_<version>/conf`

Note: Do not copy the ssl_truststore if the Hadoop administrator generated an ssl_truststore file to access Amazon Redshift or Amazon S3.

Configure the Developer Tool

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

Extract the Cluster Configuration Files

To browse metadata on the Hadoop cluster, the Developer tool requires access to the *-site.xml files. The Informatica administrator generates an archive file that needs to be extracted on each Developer tool machine.

If the Informatica administrator did not extract the archive file to the Developer tool machine, get the file and extract it to the following location: `<Informatica installation directory>\clients\DeveloperClient\hadoop\<Hadoop distribution>\conf`

When you extract the archive file, the Developer tool contains a set of *-site.xml configuration files required for Hadoop access.

Configure Files to Enable the Developer Tool

Configure files on the Developer tool client machine to create, edit, and run mappings on the MapR cluster.

mapr-clusters.conf

Get the mapr-clusters.conf file from the Informatica administrator and copy it to the Developer tool machine.

The file is in the following location on the Data Integration Service machine:

```
<Informatica installation directory>/services/shared/hadoop/mapr_<version>/conf
```

Copy the file to the following location on the Developer tool machine:

```
<Informatica installation directory>\clients\DeveloperClient\hadoop\mapr_<version>
```

developerCore.ini

Edit the `developerCore.ini` file to enable communication between the Developer tool and the Hadoop cluster.

You can find developerCore.ini in the following directory: `<Informatica installation directory>\clients\DeveloperClient`

Configure the following properties:

-Djava.library.path

Path to the Java library on the machine that hosts the Developer tool.

Use the following value: `hadoop\mapr_<version>\lib\native\Win64;bin;..\DT\bin`

-DINFA_HADOOP_DIST_DIR

Path to the Hadoop distribution directory on the cluster.

Use the following value: `hadoop\<distribution>_<version>`

For example, `-DINFA_HADOOP_DIST_DIR=hadoop\mapr_5.2.0`

run.bat

Edit the `run.bat` file to enable Developer tool launch settings for MapR.

You can find `run.bat` in the following directory on the machine where the Developer tool runs: `<Informatica installation directory>\<version_number>\clients\DeveloperClient`

Edit `run.bat` to include the MAPR_HOME environment variable and the -clean settings. For example, include the following lines:

```
<Informatica installation directory>\clients\DeveloperClient\hadoop\mapr_<version>
developerCore.exe -clean
```

Configure the Developer Tool for Kerberos

To import metadata from Hive, HBase, and complex file sources, you need to generate the Kerberos credentials file on the Developer tool machine.

1. Copy krb5.conf from `<Developer tool installation directory>/services/shared/security` to `C:/Windows`.
2. Rename krb5.conf to krb5.ini.
3. In krb5.ini, verify the value of the forwardable option to determine how to use the kinit command.
   - If `forwardable=true`, run the command with the `-f` option.
   - If `forwardable=false`, or if the option is not specified, run the command without the `-f` option.
4. To generate the Kerberos credentials file, run the kinit command from the following location: `<Developer tool installation directory> /clients/java/bin/kinit.exe`
   For example, you might run the following command: `kinit joe/domain12345@MY-REALM`

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:

Update the hadoopEnv.properties file.

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

Replace connections.

If you chose to 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 the hadoopEnv.properties File

When you run the Informatica upgrade, the installer creates a new hadoopEnv.properties file and backs up the existing configuration file.

You need to edit the version 10.2 hadoopEnv.properties file to include any manual configuration that you performed in the previous version for the corresponding distribution.

You can find the backup hadoopEnv.properties file in the following location:

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

You can find the version 10.2 hadoopEnv.properties file in the following location:

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

Replace the Connections

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

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 from Version 10.0 or Later

If you upgraded from version 10.0 or later, and *you did not create connections* when you imported the cluster configuration, you need to update connection properties for Hadoop, Hive, HDFS, and HBase connections.

After you upgrade from version 10.0 or later, you need to 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*.

Complete Connection Upgrade from Version 9.6.1

Effective in version 10.0, Big Data Management requires a Hadoop connection to run mappings on the Hadoop cluster. If you upgraded from 9.6.1 or any 9.6.1 hotfix, you must generate Hadoop connections from Hive connections that are enabled to run mappings in the Hadoop environment.

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. Generate Hadoop connections from Hive connections if the following conditions are true:

- You upgraded from 9.6.1 or any 9.6.1 hotfix.
- The Hive connections are configured to run mappings in the Hadoop environment.

Complete the following tasks to upgrade connections:

Generate Hadoop connections

Effective in version 10.0, Big Data Management requires a Hadoop connection to run mappings in the Hadoop environment. You must generate Hadoop connections from Hive connections that are enabled 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*.

INDEX

A

Amazon EMR
  Hadoop administrator tasks 31
  Hadop administrator tasks 31
  Hive access 39
  integration task flow 30
  S3 bucket access policy 39
architecture
  Big Data Management 13
  Hadoop environment 13
Azure HDInsight
  Hadoop administrator tasks 47
  integration task flow 46
  hosts file requirement 57

B

big data
  application services 14
  data lineage 16
  repositories 14
Big Data Management
  integration with Informatica products 15
  Data Quality 17
Blaze engine
  create a user account 25
  port requirements 20
  directories to create 25

C

Cloudera CDH
  Hadoop administrator tasks 63
  integration task flow 62
cluster configuration
  create 35, 51, 67, 84, 101
  import from a cluster 52, 68, 84
  import from a file 35, 53, 69, 85, 101
cluster integration 11
component architecture
  clients and tools 14
configuration files
  Developer tool configuration 109
connecting to a cluster 52, 68, 84
Custom Hadoop OS Path
  configuring 28

D

Data Discovery
  description 17
Data Integration Service
  prerequisites 28
  configuration for MapR 106, 107
data lakes
  description 16
Data Quality
  reference data 17
Data Replication
  description 17
data streaming
  description 18
Developer tool
  configuration 109
disk space
  requirements 20

H

Hadoop administrator
  prerequisite tasks for Amazon EMR 31
  prerequisite tasks for Azure HDInsight 47
  prerequisite tasks for Cloudera CDH 63
  prerequisite tasks for Hortonworks HDP 79
  prerequisite tasks for MapR 97
Hadoop administrator tasks
  Amazon EMR 31
  Azure HDInsight 47
  Cloudera CDH 63
  configure *-site.files 31, 47, 63, 79, 97
  Hortonworks HDP 79
  MapR 97
Hadoop operating system
  on Data Integration Service 27
hadoopEnv.properties
  configure 36, 54, 70, 87, 103
HDFS staging directory 25
high availability
  configuration on Developer tool 109
Hive access
  for Amazon EMR 39
Hortonworks HDP
  Hadoop administrator tasks 79
  integration task flow 78
hosts file
  Azure HDInsight 57

I

Informatica adapters
  integration with Big Data Management 15

J

JDBC
  Sqoop connectivity 38, 56, 72, 89, 105

K

Kerberos authentication
  security certificate import 72, 89

M

MapR
  Hadoop administrator tasks 97
  integration task flow 95
  Data Integration Service configuration 106, 107
  tickets 105
mapr-cluster.conf
  requirements 108
mapr.login.conf
  requirements 108

O

overview 11

P

permissions
  Blaze engine user 25
ports
  Amazon EMR requirements 20
  Azure HDInsight requirements 20
  Blaze engine requirements 20
Prerequisite
  download Hadoop operating system 27
prerequisite tasks for Amazon EMR 30
prerequisite tasks for Azure HDInsight 46
prerequisite tasks for Cloudera CDH 62
prerequisite tasks for Hortonworks HDP 78
prerequisite tasks for MapR 95
prerequisites
  create directories for the Blaze engine 25
  disk space 20
  Hadoop administrator tasks. 31, 47, 63, 79, 97
  verify system requirements 19
  Data Integration Service properties 28
  uninstall 21
  verify product installations 19
process 11
product installations
  prerequisites 19

R

reject file directory
  HDFS 26

S

Sqoop
  hadoopEnv.properties 36, 54, 70, 87, 103
  JDBC drivers 38, 56, 72, 89, 105
staging directory
  HDFS 25
system requirements
  prerequisites 19

U

uninstall
  prerequisite 21
user accounts
  MapR 105