How-To Library

Informatica

# Install Data Engineering Integration on Kubernetes with the Container Utility (10.4.0 - 10.4.1)

# Abstract

Informatica provides the Informatica container utility to install the Informatica domain quickly. This article describes how to install Data Engineering Integration from the Docker image through the Informatica container utility on Kubernetes.

# Supported Versions

- Data Engineering Integration 10.4.0 - 10.4.1

# Table of Contents

# Overview

With a web-based interface in the Informatica container utility, you can easily create standard docker images and create a container using docker.

Docker is an open source platform that provides an isolated environment called containers to run the applications. Docker allows independent containers to run within a single Linux instance. A docker image is an executable package that can run an application, a code, run-time files, environment variables, or configuration files. A container is a run-time instance of an image.

You can configure Data Engineering Integration on Kubernetes to optimize resource management and to enable load balancing for the Informatica domain within the containerized environment. You can also provide horizontal scaling of pods based on the load provided on the domain. The containers are portable across cloud and operating system distribution for Amazon EKS.

Run the Informatica container utility to build or run the docker image to configure the domain. You can build the docker image with the base operating system and Informatica product binaries. With the Informatica container utility, you can create containers using this docker image to configure the Informatica domain.

When you run the docker image, you can install Informatica products with the Informatica container utility and you can create or join a domain.

With the Informatica container utility, you can build or run the docker image with the silent installer. CI/CD or continuous integration and continuous delivery, is a practice that automates the integration and delivery operations in a CI/CD pipeline. With the CI/CD pipeline, you can quickly deploy, test, and deliver the docker image to the production environment. You can generate the docker image from the CI/CD content. The CI/CD content uses the same input values that you used in the Informatica container utility to either build or run the docker image. You can manually replace the sample silent input properties with the CI/CD content before you run the silent installation.

You can create the application services when you run the docker image, or you can create them in the Administrator tool.

You can also use the prepackaged domain image to configure the domain and to install the Informatica products.

# Container Utility Process Checklist

Perform the following tasks associated with the installation:

- ❑ Plan for all installation components within the domain, such as nodes and services.
- ❑ Complete prerequisites:
  - Verify system requirements.
  - Configure docker engine.
  - Set up keystore and truststore files.
  - Extract the Informatica container utility and access the Informatica installer .tar file.

- Verify the license key.
- Set up repository databases.
- Prepare for the cluster configuration.
- Prepare for Kubernetes configuration.
- Generate CI/CD content.

❑ Build the Informatica docker image with base operating system and Informatica binaries.

❑ Run the docker image to configure the Informatica domain.

❑ Deploy the docker image quickly or download and run the default image.

❑ Build or run the docker image in silent mode with CI/CD content.

❑ Complete the post-requisites:

- Complete the domain configuration.
- Create the application services.
- Install the Developer tool.

# Before You Begin

Before you install the Informatica container utility, verify that the machine meets the pre-installation requirements for the Informatica product installation.

## *Verify System Requirements*

Verify that your environment meets the minimum system requirements for the installation process:

- Disk space of 56 GB in the docker build directory for Data Engineering Integration.
- Disk space of 50 GB in the current working directory for Data Engineering Integration.
- Disk space of 27 GB in the docker configuration directory for Data Engineering Integration.
- Supported image name or one of the following image names for the subscribed base operating system: `registry.access.redhat.com/rhel7` or `centos:7`.
- Register and subscribe the machine where you build the docker images.
- Temporary disk space and permissions
- Patch requirements
- Sizing Requirements

## *Set Up the Keystore and Truststore Files*

When you install the Informatica services, you can configure secure communication for the domain and set up a secure connection to Informatica Administrator (the Administrator tool). If you configure the security options, you must set up the keystore and truststore files.

## Contact Informatica Global Customer Support

When you initially contact Informatica Global Customer Support for Informatica installation access, the Informatica Global Customer Support team initiates a shipping request. The shipping team sends the Informatica installation link and license information to you.

You can contact Informatica Global Customer Support online for access to the following files:

- Informatica installer .tar file
- License key
- Informatica client installation

### Access to .tar File

You require access to both the Informatica container utility and the Informatica installer .tar file to install the product.

Download and extract the Informatica container utility installer files from the download center for Data Engineering Integration location on Informatica Marketplace.

After you extract the informatica_container_utility_linux-x64.zip file, contact the support team to access the Informatica installer .tar file. You can then place the Informatica installer tar file on the machine where you build the image.

To gather inputs for the silent input properties files, launch the Informatica container utility and click **Generate silent properties for CI/CD**. You can then change the format of the properties file to UNIX before you run the silent installation.

### Verify the License Key

Before you install the software, verify that you have the license key available for the Informatica product installer.

The product is provided as a Bring Your Own License (BYOL). Copy the license key file to a directory accessible to the user account that installs the product. You can use an existing Informatica license or contact Informatica Global Customer Support if you do not have a license key or if you have an incremental license key.

The volume directory is the directory where the Informatica license key and certificates are stored. You must have read, write, and execute permissions on the volume directory.

**Note:** You do not need a license for the Informatica container utility.

### Access the Informatica Client Installation

Ensure that the Informatica Global Customer Support team has provided you with access to the Informatica client installation.

Before you install the Informatica clients, verify that the machine meets the minimum system and third-party software requirements. If the machine where you install the Informatica clients is not configured correctly, the installation can fail.

## Set Up Databases

Informatica components store metadata in relational database repositories. When you set up the databases, you will need to allow for disk space and create the databases with parameters required by the product. You will also create user accounts and install the database clients.

Set up databases for the following repositories:

**Domain configuration repository**

The domain stores configuration and user information in a domain configuration repository. You can create the domain configuration repository on an Oracle, PostgreSQL, or Microsoft SQL Server database.

**Model repository**

The Model repository stores information about the metadata for the data objects and mappings in a relational database. If you want to generate monitoring statistics, you must create a dedicated monitoring Model repository to store run-time monitoring statistics. You can create the Model repository and the monitoring Model repository on an Oracle, PostgreSQL, or Microsoft SQL Server database.

Create either an on-premises database or a container database for each repository. When you run the docker image, you will provide the database information. Create the container database to share the system resources for disk space, kernels, and operating system libraries within the containerized application.

## Domain Configuration Repository

The domain stores configuration and user information in a domain configuration repository.

You can create the domain configuration repository in one of the following databases.

**Oracle**

Complete the following tasks to prepare the Oracle database on-premises or in the docker container:

- Create the database with therequired parameters, allowing for 200 MB of disk space.
- Set up database user accounts.
- Install compatible versions of the Oracle client and Oracle database server. You must also install the same version of the Oracle client on all machines that require it.

**PostgreSQL**

Complete the following tasks to prepare the PostgreSQL database on-premises or in the docker container:

- Create the database with the required parameters, allowing for 200 MB of disk space.
- Set up database user accounts.

You do not need to install the PostgreSQL client software, as the installer bundles it with the image.

**Microsoft SQL Server**

Complete the following tasks to prepare the Microsoft SQL Server database on-premises or in the docker container:

- Create the database with the required parameters, allowing for 200 MB of disk space.
- Set up database user accounts.
- Download the Microsoft SQL Server client and install it on all machines that require it.

## Model Repository

Create a Model repository to store information about the metadata for the data objects and mappings in a relational database. If you want to generate monitoring statistics, you must create a dedicated monitoring Model repository to store run-time monitoring statistics.

You can create the Model repository and the monitoring Model repository in one of the following databases:

**Oracle**

Complete the following tasks to prepare an Oracle database for the Model repository and for the monitoring Model repository or to create a container database:

- Create the database with the required parameters, allowing for 200 MB of disk space.
- Set up database user accounts.
- Install compatible versions of the Oracle client and Oracle database server. You must also install the same version of the Oracle client on all machines that require it.

**PostgreSQL**

Complete the following tasks to prepare a PostgreSQL database for the Model repository and for the monitoring Model repository or to create a container database:

- Create the database with the required parameters, allowing for 200 MB of disk space.
- Set up database user accounts.

You do not need to install the PostgreSQL client software, as the installer bundles it with the image.

**Microsoft SQL Server**

Complete the following tasks to prepare a Microsoft SQL Server database for the Model repository and for the monitoring Model repository or to create a container database:

- Create the database with the required parameters, allowing for 200 MB of disk space.
- Set up database user accounts.
- Download the Microsoft SQL Server client and install it on all machines that require it.

## *Create the Cluster Configuration*

When you run the docker image, you can choose create the cluster configuration required to connect to the Hadoop cluster. The installer imports property values from *-site.xml files required to run mappings in the Hadoop environment.

You can choose to create the cluster configuration in one of the following ways:

**Import directly from the cluster**

The Hadoop administrator can provide you with cluster authentication information to connect to the cluster for the import process.

The following table describes the properties that you need to provide when you run the docker image:

| Property | Description |
|---|---|
| Host | The host name or IP address of the cluster manager. |
| Port | Port of the cluster manager. |
| User ID | Cluster user name. |
| Password | Password for the cluster 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 installer imports information based on the default cluster. |

**Import through an archive file**

The Hadoop administrator can provide you an archive file that contains properties from *- site.xml files on the cluster.

The *-site.xml files that you package in the archive file depend on the Hadoop distribution:

- Amazon EMR
- Azure HDInsight
- Cloudera CDH
- Cloudera CDP
- Hortonworks HDP
- MapR

**Note:** If you are importing from Amazon EMR or MapR, you can import only from an archive file.

### *Verify Kubernetes Configuration*

Complete the following tasks on the machine where you run the Informatica container utility:

1. Install kubectl.
2. Ensure that the Kubernetes cluster that you set up a Kubernetes master node and one or more worker nodes.
3. Ensure that the machine where the container utility runs can communicate with the Kubernetes cluster through the kubectl cluster-info command. The command displays information about the master and services that run in the cluster.

For more information, see the Kubernetes documentation.

### *Generate CI/CD Content for Silent Input Properties File*

CI/CD is a process of continuous integration, continuous delivery, and continuous deployment. With the CI/CD process available when you create or build the docker image in the Informatica Container Utility, you can generate property values to run the silent installer. Before you can run the silent installer, click **Generate silent properties for CI/CD** from the Informatica container utility. Use the generated CI/CD values to replace the contents of the silent input properties file. You must convert the properties file to the UNIX format before you run the silent installer.

When you run the CI/CD process, you can copy all the user input values entered to build or run the docker image from the Informatica container utility from a single panel. The generated CI/CD contents are in the supported format of the silent install property files. Manually save the CI/CD contents into the silent input properties file located in the silent installation directory. Convert the properties file to the UNIX format before you run the silent installation.

## Build the Informatica Docker Image

Build a docker image for Informatica products. After you build the docker image, the Informatica container utility uses the docker image to run on containers. Ensure that the Informatica docker image is stored in the local host or docker registry before you install Informatica products in the docker with the Informatica container utility.

### *Step 1. Choose the Deployment Type*

On the **Deployment type** page, you must select the deployment type and host instance to build the docker image.

1. Launch the Informatica container utility with the `sh startup.sh` command. By default, the utility uses port 12386.

   To specify the startup command with a different port number, enter the values in the following format: `sh startup.sh <required port number>`
2. To build the Informatica Docker image, select **Build the docker image**.

   The **Deployment type** page appears.

3. On the **Deployment type** page, enter the information for the deployment type.
   - Select Data Engineering Integration for the product.
   - Choose to host on Amazon Web Services (AWS), Microsoft Azure, or on-premises.

   The build image uses Docker as the default deployment type.
4. Optionally, you can choose to provide secure authentication to build the image or deploy the containers on the remote host. If you do not want to set the secure authentication for the containers, go to step 6.
5. If you enable secure authentication, set the authentication type to **Password** or **Key**.
   - For password type authentication, enter the machine IP address or host name, user name, password, and port to connect to the host machine.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
6. Click **Next**.

   The **Build the docker image** page appears.

## *Step 2. Build the Docker Image*

On the **Build Docker Image** page, enter information for the tar file location and the images, or accept the default values to build the docker image.

1. Specify the base operating system path to build the docker image.
   - To build the docker image on CentOS, enter a value or select the default value of `centos:7`.
   - To build the docker image on RHEL, enter a value or select the default value of `registry.access.redhat.com/rhel7:7.8`.
2. Enter the Informatica installer tar file path on the machine where you build the image.

   The Informatica container utility uses the tar utility to extract the installer server files to a directory on your machine where you build the image.
3. Specify the docker base image name.

   Default is `informatica1041:1.0`, where `1.0` is the tag name.
4. Enter the docker build directory.

   Specify the machine host directory to build the docker image.

   Default is `/user/home`.
5. Optionally, you can choose to generate the silent properties for CI/CD.

   When you generate silent properties for CI/CD from the container utility, you can also copy all the installation options and the values specified in the container utility and later paste the contents into the property file for the silent installer.
6. Click **Build**.

## *Step 3. Verify the Informatica Docker Image*

You can verify that the Informatica image is present in the host specified while building the docker image.

To verify that the Informatica docker image exists, enter the docker images command from the command prompt. Ensure that the values for the tag, image ID, created date, and size information appears for the Informatica docker image.

The following sample displays the result of the docker image command:

```
[root@master abc]#docker images
REPOSITORY                    TAG         IMAGE ID      CREATED       SIZE
informatica1041pcrspostgre    1.0         f9923fb2cfa0  20 hours ago  17.7 GB
```

# Run the Informatica Docker Image to Create a Domain

You can install Informatica products by running the docker image. Run the docker image to create nodes in the Informatica domain.

The first time that you run the docker image, choose to create a domain. When you create a domain, the node that you create becomes a gateway node in the domain. The gateway node contains a Service Manager that manages all the domain operations.

## *Step 1. Choose the Deployment Type*

On the **Deployment type** page, you must select the deployment type and host instance to run the docker image.

1. Launch the Informatica container utility with the `sh startup.sh` command. Default port is 12386.

   To specify the startup command with a different port number, enter the values in the following format:

   ```
   sh startup.sh <required port number>
   ```

2. To run the Informatica Docker image, select **Run the docker image**.

   The **Deployment Type** page appears.

3. Enter the information for the deployment type.
   - Select Data Engineering Integration for the product.
   - Choose to host on Amazon Web Services (AWS), Microsoft Azure, or on-premises.
   - Select the Kubernetes deployment type.
4. Optionally, you can choose to provide secure authentication to deploy the containers on the remote host. If you do not want to set the secure authentication for the containers, go to step 6.
5. If you enable secure authentication, set the authentication type to **Password** or **Key**.
   - For password type authentication, enter the machine IP address or host name, user name, password, and port to connect to the host machine.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
6. Click **Next**.

   The **Kubernetes Auto Scaling** panel appears.

## *Step 2. Provide Docker and Kubernetes Auto Scaling Information*

On the **Kubernetes Auto Scaling** page, provide the required information to connect to the docker image and pod.

1. In the **Docker and Kubernetes information** section, enter the docker image name, pod name, path and file name of the license key, and the volume directory. Enter the existing directory to create tmpfs volume. Ensure that you specify different file paths for the volume directory and the license key directory.
2. Optionally, choose to set Kubernetes autoscaling.

   The Horizontal Pod Autoscaler scales the number of pods in a deployment or replica set based on the CPU utilization mentioned for the CPU load factor.

Kubernetes creates horizontally scalable worker nodes that are added to the grid with the Data Integration Service process enabled.

3. If you selected Kubernetes autoscaling, enter the CPU utilization and worker node information to autoscale:
   - **CPU load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and only when the Kubernetes worker node reaches the value set for the CPU percentage metrics. Default is 80 percent.
   - **Maximum number of Informatica worker nodes**. The maximum number of worker nodes to create in the domain. Default is 3.

   The minimum number of worker node in the domain is 1.
4. Optionally, choose to store data outside of the containers when you select the persistent volume.

   If you selected persistent volume, you cannot join the domain.
5. If you selected persistent volume, specify the persistent volume claim name.
6. Optionally, choose to expose additional ports in the container, and enter them as comma separated values.
7. Enter the network name.

   You can update the **Domain selection** section on the same page.

## *Step 3. Provide the Domain Connection Information*

On the **Kubernetes Auto Scaling** page, provide information to create a domain, connection details for Informatica Administrator, and whether to secure communication to the domain or not.

1. In the **Domain selection** section, choose to create a domain.
2. Optionally, choose to enable secure communication for services in the Informatica domain.

   By default, if you enable secure communication for the domain, the installer sets up an HTTPS connection. You can also create a domain configuration repository on a secure database.
3. Specify the connection details for Informatica Administrator.
   a. If you disabled secure communication for the domain, you can specify whether to set up a secure HTTPS connection for Informatica Administrator.
   b. If you enabled secure connection for the domain or if you enabled HTTPS connection, enter the HTTPS port number and the keystore file information. Choose to use a default keystore or a custom keystore file.
      - If you use the default keystore, the installer creates a self-signed keystore file named Default.keystore in the following location: `<Informatica installation directory>/tomcat/conf/`
      - If you use a custom keystore, specify the location and password.
4. Specify whether to use the default Informatica SSL certificates or to use your SSL certificates to secure domain communication.
5. If you provide the SSL certificates, copy the keystore and truststore files to a directory in the persistent volume, and then specify the location and passwords of the copied files. The directory must contain a keystore file named infa_keystore.jks or infa_keystore.pem and a truststore file named infa_truststore.jks or infa_truststore.pem.

   The persistent volume is mounted at `/mnt` in the pod.
6. Click **Next**.

   The **Domain database** page appears.

## Step 4. Configure the Domain Configuration Repository

On the **Domain database** page, provide the domain configuration repository database information, JDBC connection information, and specify information for the domain and gateway node.

1. Select the database to use for the domain configuration repository.
2. Optionally, specify if you created a database in the container. If you created a database in the container, complete the container configuration for the domain configuration repository.
3. Enter the database user account and password.
4. Specify whether the database is secure.

   If you select the secure database option, provide inputs for the qualified path to the database truststore file, and truststore password.
5. Enter the parameters for the database.
   a. If you select Microsoft SQL Server or PostgreSQL, enter the schema name for the database. If you select Oracle, the installer creates the tables in the default schema.
   b. To enter the JDBC connection information, you can use either the JDBC URL or the JDBC connection string.

      To use the JDBC URL information, select **JDBC URL**, and specify the following JDBC URL properties:

      - **Database address.** Host name and port number for the database.
      - **Database service name.** Oracle service name, the database name for Microsoft SQL Server, or the database name for PostgreSQL.

      Optionally, choose to include additional JDBC parameters.
   c. To use a custom JDBC connection string, select **JDBC Connection String**.

      You can use the following syntax in the JDBC connection string to connect to a secure database:
      **Oracle**

      ```
      jdbc:Informatica:oracle://<host name>:<port number>;ServiceName=<service
      name>;EncryptionMethod=SSL;HostNameInCertificate=<database host
      name>;ValidateServerCertificate=<true or false>
      ```

      **Microsoft SQL Server**

      ```
      jdbc:Informatica:sqlserver://<host name>:<port
      number>;SelectMethod=cursor;DatabaseName=<database
      name>;EncryptionMethod=SSL;HostNameInCertificate=<database host
      name>;ValidateServerCertificate=<true or false>
      ```

      **PostgreSQL**

      ```
      jdbc:Informatica:postgresql://<host name>:<port number>;DatabaseName=<database
      name>;EncryptionMethod=SSL;HostNameInCertificate=<database host
      name>;ValidateServerCertificate=<true or false>
      ```
6. In the **Domain and node configuration** section, enter the keyword and directory for the encryption key for the Informatica domain.

   - **Encryption key directory.** Directory in which to store the encryption key for the Informatica domain. The encryption key is created in the following directory: `<Informatica installation directory>/isp/config/keys`.
   - **Keyword.** Keyword to use to create a custom encryption key to secure sensitive data in the domain. The keyword cannot contain spaces; must be 8 to 20 characters long; and must include at least one number and one uppercase and lowercase letter.

     **Note:** If you selected persistent volume, the keyword option is not displayed.

7. Specify the domain name, node name, domain port number, domain user name, and password. When you enter the information for the domain and the gateway node that you want to create, consider the following guidelines:
   - The default domain name is Domain. The domain name must not exceed 128 characters and must be 7-bit ASCII only. The name cannot contain a space or any of the following characters:
     ` % * + ; " ? , < > \ /
   - The domain port number must be in the range of 30000 to 32657.
   - Informatica container utility exposes the ports from the range <domain port> - <domain port > + 110 ports from the container to the docker host, where the initial port number is the domain port number. Enter the service ports within the supported service port range.
   - The password for the domain administrator must be more than 2 characters and must not exceed 128 characters.
8. You can choose to configure application services or run the docker image.

   **Note:** If you choose to create the supported application services, you must also provide the related database information as part of the Informatica container utility tasks. You can also choose to set the cluster configuration.

## *Step 5. Configure the Model Repository Service*

On the **Model Repository Service** page, you can configure the Model repository database properties.

1. In the **Model Repository Service information** section, enter the database type, user ID, and user password. Specify the database name or database service name based on the database associated with Model repository.
2. Optionally, specify if you created a database in the container. If you created a database in the container, complete the container configuration for the Model repository.
3. Specify whether the database is secure.

   If you select the secure database option, you need to provide inputs for the qualified path to the database truststore file, truststore password, and specify the secure JDBC Parameters. By default, the value for the secure JDBC parameters is

   ```
   EncryptionMethod=SSL;HostNameInCertificate=;ValidateServerCertificate=false;
   ```
4. Enter the parameters for the database.
   a. If you select Microsoft SQL Server or PostgreSQL, enter the schema name for the database. If you select Oracle, the installer creates the tables in the default schema.
   b. To enter the JDBC connection information, you can use either the JDBC URL or the JDBC connection string.

      To use the JDBC URL information, select **JDBC URL**. To enter the connection information using the JDBC URL information, specify the following JDBC URL properties:
      - **Database address.** Host name and port number for the database.
      - **Database service name.** Oracle service name, the database name for Microsoft SQL Server, or the database name for PostgreSQL.

      Optionally, chose to include additional JDBC parameters.

c. To use a custom JDBC connection string, select **JDBC connection string**.

You can use the following syntax in the JDBC connection string to connect to a secure database:
**Oracle**

```
jdbc:Informatica:oracle://<host name>:<port number>;ServiceName=<service name>
```

**Microsoft SQL Server**

```
jdbc:Informatica:sqlserver://<host name>:<port
number>;SelectMethod=cursor;DatabaseName=<database name>
```

**PostgreSQL**

```
jdbc:Informatica:postgresql://<host name>:<port number>;DatabaseName=<database name>
```

5. Click next to configure additional services, or run the docker image.

## *Step 6. Configure the Data Integration Service*

After you configure the Model Repository database, you can configure the service parameters for the application service.

1. In the **Data Integration Service information** section, you can specify the connection details for the Data Integration Service:
   a. Choose if you want to enable secure communication for the Data Integration Service with the HTTP, HTTPS, or HTTP&HTTPS connection options and specify the port number for the Data Integration Service.
   b. If you enabled secure connection, enter the HTTP port number and the keystore file information. Choose to use a default keystore or a custom keystore file.
      - If you use the default keystore, the installer creates a self-signed keystore file named Default.keystore in the following location: `<Informatica installation directory>/tomcat/conf/`
      - If you use a custom keystore, specify the location and password.

      **Note:** If you enabled HTTPS connection, you can enter the HTTPS port number and do not need to specify the keystore information.
2. Specify whether to use the default Informatica SSL certificates or to use your SSL certificates to secure the communication to the Data Integration Service.
3. If you provide the SSL certificate, specify the location and passwords of the keystore and truststore files. The directories must contain a keystore file named infa_keystore.jks or infa_keystore.pem and a truststore file named infa_truststore.jks or infa_truststore.pem.
4. Click next to configure additional services, or run the docker image.

## *Step 7. Configure the Monitoring Model Repository Service*

On the **Monitoring Model Repository Service** page, you can configure the Monitoring Model repository database properties.

1. In the **Monitoring Model Repository Service information** section, enter the database type, user ID, and user password. Specify the database name or database service name based on the database associated with monitoring Model Repository database.
2. Optionally, specify if you created a database in the container. If you created a database in the container, complete the container configuration for the monitoring Model repository.
3. Specify whether the database is secure.

If you select the secure database option, you need to provide inputs for the qualified path to the database truststore file, truststore password, and specify the secure JDBC Parameters. By default, the value for the secure JDBC parameters is

```
EncryptionMethod=SSL;HostNameInCertificate=;ValidateServerCertificate=false;
```

4. Enter the parameters for the database.
   a. If you select Microsoft SQL Server or PostgreSQL, enter the schema name for the database. If you select Oracle, the installer creates the tables in the default schema.
   b. To enter the JDBC connection information, you can use either the JDBC URL or the JDBC connection string.

      To use the JDBC URL information, select **JDBC URL**. To enter the connection information using the JDBC URL information, specify the following JDBC URL properties:

      - **Database address.** Host name and port number for the database.
      - **Database service name.** Oracle service name, the database name for Microsoft SQL Server, or the database name for PostgreSQL.

      Optionally, chose to include additional JDBC parameters.
   c. To use a custom JDBC connection string, select **JDBC connection string**.

      You can use the following syntax in the JDBC connection string to connect to a secure database:
      **Oracle**

      ```
      jdbc:Informatica:oracle://<host name>:<port number>;ServiceName=<service name>
      ```

      **Microsoft SQL Server**

      ```
      jdbc:Informatica:sqlserver://<host name>:<port
      number>;SelectMethod=cursor;DatabaseName=<database name>
      ```

      **PostgreSQL**

      ```
      jdbc:Informatica:postgresql://<host name>:<port number>;DatabaseName=<database name>
      ```
5. Click next to configure the cluster configuration, or run the docker image.

## *Step 8. Configure the Cluster*

Optionally, on the **Cluster configuration** page, you can specify the cluster configuration for the Hadoop environment.

1. In the **Cluster configuration information** section, choose to enter the name of the cluster configuration to create.
2. Specify the Hadoop distribution for the cluster from the drop down.
3. Select the method to import the configuration properties from the Hadoop cluster to create the cluster configuration. You can import the properties from an archive file or directly from the cluster.
   - If you import from the archive file, enter the archive file. Optionally, specify the cluster name.

     **Note:** If you select MapR or Amazon EMR cluster, you must import the cluster configuration properties from an archive file.
   - If you import the properties directly from the cluster, specify the host name or IP address of the cluster manager. You also need to provide the cluster manager port, user name, password, and the 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.
4. Optionally, choose to create the Hadoop, HDFS, Hive, and HBase connections.

   The installer appends the connection type to the cluster connection name to create each connection name.

5. Optionally, you can choose to generate the silent properties for CI/CD.

   When you generate silent properties for CI/CD from the container utility, you can also copy all the installation options and the values specified in the container utility and later paste the contents into the property file for the silent installer.

6. Click **Run the docker image**.

7. To verify that the container is present in the host specified, run the docker ps -a command from the command prompt.

   Ensure that you can see the the container ID, image, names, command, created date, and status of each container appears.
   The following sample displays the result of the docker ps -a command:

```
[root@irlcmg08 source]#docker ps -a
CONTAINER ID PORTS  IMAGE NAMES                          COMMAND
CREATED     STATUS
566c418d0972        informaticaltd/pcdqservices:1041pc  "/Installer/launcher..." 2 days
ago  Up 2 days
```

8. To verify all the docker images, run the docker images -a command in the command line.

   Ensure that you can see the repository name, tags, and the size information for the images.

After the docker image run completes, you can view the post-installation summary on the Informatica container utility. You can view the installation log files to get more information about the tasks performed by the utility inside the container. To view the log inside the container, run the docker exec -ti <container name> /bin/bash command. You can navigate to the `/home/Informatica` directory to see the installer logs. When the installation is complete, the Informatica_<Version>_Services_<timestamp>.log file is created in the installation directory specified inside the container.

# Run the Informatica Docker Image to Join a Domain

After you create the domain, you can run the Informatica container utility on any machine to join the domain. When you join the domain, you need to create a node on the machine and then add the node to the domain.

## *Step 1. Choose the Deployment Type*

On the **Deployment type** page, you must select the deployment type and host instance to run the docker image.

1. Launch the Informatica container utility with the `sh startup.sh` command. Default port is 12386.

   To specify the startup command with a different port number, enter the values in the following format:

   `sh startup.sh <required port number>`

2. To run the Informatica Docker image, select **Run the docker image** in the Informatica container utility.

   The **Deployment type** page appears.

3. Enter the information for the deployment type selection.
   - Select Data Engineering Integration for the product.
   - Choose to host on Amazon Web Services (AWS), Microsoft Azure, or on-premises.
   - Select the Kubernetes deployment type.

4. Optionally, you can choose to provide secure authentication to deploy the containers on the remote host. If you do not want to set the secure authentication for the containers, go to step 6.

5. If you enable secure authentication, set the authentication type to **Password** or **Key**.
   - For password type authentication, enter the machine IP address or host name, user name, password, and port to connect to the host machine.

- For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.

6. Click **Next**.

   The **Kubernetes Auto Scaling** panel appears.

## *Step 2. Provide Kubernetes Auto Scaling Information*

On the **Kubernetes Auto Scaling** page, provide the required information to connect to the docker image and pod.

1. In the **Docker and Kubernetes information** section, enter the docker image name, pod name, path and file name of the license key, and the volume directory. Enter the existing directory to create tmpfs volume.

   Ensure that you enter different values for the volume directory and the license key directory. You must also ensure that the value you enter for the volume directory matches the volume directory path for the domain.

2. Optionally, choose to set Kubernetes autoscaling.

   The Horizontal Pod Autoscaler scales the number of pods in a deployment or replica set based on the CPU utilization mentioned for the CPU load factor.

   Kubernetes creates horizontally scalable worker nodes that are added to the grid with the Data Integration Service process enabled.

3. If you selected Kubernetes autoscaling, you must enter the CPU utilization and worker node information to autoscale:

   - **CPU load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and only when the Kubernetes worker node reaches the value set for the CPU percentage metrics. Default is 80 percent.
   - **Maximum number of Informatica worker nodes**. Maximum number of worker nodes to create in the domain. Default is 3.

4. Optionally, choose to expose additional ports in the container, and enter them as comma separated values.
5. Enter the network name.

   You can update the **Domain selection** section on the same page.

## *Step 3. Provide the Domain Connection Information*

On the **Kubernetes Auto Scaling** page, provide information to join a domain, connection details for Informatica Administrator, and whether to secure communication to the domain or not.

1. In the **Domain Selection** section, choose to join a domain.

   The installer creates a node on the container where you install. You can specify the domain to join.

2. Optionally, choose to enable secure communication for services in the Informatica domain.

   By default, if you enable secure communication for the domain, the installer sets up an HTTPS connection. You can also create a domain configuration repository on a secure database.

3. Specify the node type that you want to create. If you have previously enabled Kubernetes autoscaling in the **Docker Information** section, the created node is a gateway. This is because you can only enable Kubernetes autoscale on worker nodes. You can also enable a secure HTTPS connection to the Informatica Administrator.
4. Specify the connection details for Informatica Administrator.

   a. If you disabled secure communication for the domain, you can specify whether to set up a secure HTTPS connection for Informatica Administrator.

b. If you enabled secure connection for the domain or if you enabled HTTPS connection, enter the HTTPS port number and the keystore file information. Choose to use a default keystore or a custom keystore file.

- If you use the default keystore, the installer creates a self-signed keystore file named Default.keystore in the following location: `<Informatica installation directory>/tomcat/conf/`
- If you use a custom keystore, specify the location and password.

5. Specify whether to use the default Informatica SSL certificates or to use your SSL certificates to secure domain communication.
6. If you provide the SSL certificates, copy the keystore and truststore files to a directory in the persistent volume, and then specify the location and passwords of the copied files. The directory must contain a keystore file named infa_keystore.jks or infa_keystore.pem and a truststore file named infa_truststore.jks or infa_truststore.pem.

   The persistent volume is mounted at `/mnt` in the pod.
7. Click **Next**.

   The **Domain Configuration** page appears.

## *Step 4. Configure the Domain*

On the **Domain Configuration** page, provide the domain, node, gateway node, and gateway container information.

1. In the **Domain Configuration** section, specify the domain name, gateway container name, gateway node port, domain user name, and password.
2. Optionally, choose to join an Informatica domain created through the container database.
3. In the **Domain Configuration** section, enter the key and directory for the encryption key for the Informatica domain.
   - **Select the encryption key.** Specify the custom site key to secure sensitive data in the domain. Before you can enter the site key path, copy the file from the encryption key directory within the container to the host with the following command: docker cp <container name>: <encryption key directory>/siteKey <dst path in host>
   - **Encryption key directory.** Directory in which to store the encryption key for the Informatica domain. The encryption key is created in the following directory: `<Informatica installation directory>/isp/config/keys`.
4. Under the **Domain and Node Configuration** section, enter the node host name, node name, and node port number for the current node. The node host name is the container name of the node.
5. Click **Deploy**.

   The docker image runs.
6. To verify that the container is present in the host specified, run the docker ps -a command from the command prompt.

   Ensure that you can see the container ID, image, names, command, created date, and status of each container appears.
   The following sample displays the result of the docker ps -a command:

```
[root@irlcmg08 source]#docker ps -a
CONTAINER ID PORTS  IMAGE NAMES                          COMMAND
CREATED     STATUS
566c418d0972        informaticaltd/pcdqservices:1041pc  "/Installer/launcher..." 2 days
ago  Up 2 days
```

7. To verify the docker images, run the docker images -a command in the command line.

   Ensure that you can see the repository name, tags, and the size information for the images.

After the docker image run completes, you can view the post-installation summary. You can view the installation log files to get more information about the tasks performed by the utility. The installation is complete when the Informatica_<Version>_Services_<timestamp>.log file is created in the installation directory.

# Complete the Post-Install Tasks

After you run the Informatica container utility, perform the post-installation tasks.

## *Complete the Domain Configuration*

To complete the domain configuration after you install the Informatica services, perform the following tasks:

- Verify locale settings and code page
- Configure Informatica environment variables
- Configure the library path environment variables
- Configure locale environment variables

## *Complete the Domain Integration*

Complete the following tasks:

**Update the hosts file to access the Administrator tool through the browser.**

To access the Administrator tool from the clients or to communicate to the Administrator tool from the node port, enter the node external IP and pod name in the hosts file:

1. To get the node name for the pod where you created the domain, enter the following command with the pod name:
   `kubectl get pod <pod name> -o wide`

   The output displays the node name on which the pod runs. Note the node name of the pod.
2. To get the node's IP address, enter the following command with the node name:
   `kubectl get node <node name> -o wide`

   The output lists the node and the corresponding external IP. Note the external IP address of the node that has the pod running.
3. On the Windows machine, add the following entry to the hosts file in the `C:\Windows\System32\drivers\etc\hosts` path, and save the file:
   `<Node external IP> <tab or space> <pod name>`

   For example: `37.100.167.23 infaserver`

**Log in to Informatica Administrator.**

To connect to the Administrator tool, get the node IP on which the infaserver runs and the NodePort of the pod. To access the Administrator tool in a browser, enter the pod name followed by the NodePort number in the following format:

`http://<pod name>:<node port>`

For example: `http://new-infaserver:32005`

Default Administrator tool node port number is 32008.

**Create the application services if you did not create it during deployment.**

If you did not create the Model Repository Service and the Data Integration Service when you created a domain, use the service creation wizard in the Administrator tool to create them. You must also create the Metadata Access Service so that you can import metadata from the Hadoop environment.

Create the following services:

- Model Repository Service. To generate monitoring statistics, you must create a dedicated Model Repository Service for monitoring.
- Data Integration Service
- Metadata Access Service

**Complete the integration of the domain with the non-native environment.**

For information about how to integrate the domain with the non-native environment, see the Integration Guide.

### *Install the Developer tool*

You can copy the Developer tool installation binaries from the Akamai link that you received when you contacted Informatica Global Customer Support for the Informatica installation tar file. Copy the files to your installation directory and install the Developer tool.

You can install the client to create data objects, create and run mappings, and create virtual databases. To install the client, perform the following tasks:

**Before you install the client.**

Before you install the Informatica client, verify that the minimum installation requirements are met. If the machine where you install the Informatica client is not configured correctly, the installation can fail.

**Install the client.**

Use the Informatica client installer to install the Developer tool.

**Install languages.**

To view languages other than the system locale and to work with repositories that use a UTF-8 code page, install additional languages on Windows for use with the Informatica clients.

**Configure the client for a secure domain.**

When you enable secure communication within the domain, you also secure connections between the domain and Informatica client applications. Based on the truststore files used, you might need to specify the location and password for the truststore files in environment variables on each client host.

**Start the Developer tool.**

The first time you start the Developer tool, you add the domain and connect to a Model repository. To connect to the node, you can get the host name and port number from node present on the Administrator tool.

## Starting and Stopping the Informatica Services

Run infaservice.sh to start and stop the Informatica daemon.

You can start the daemon with the infaservice.sh startup command. To stop the daemon, enter the infaservice.sh shutdown command. By default, infaservice.sh is installed in the following directory:

```
<Informatica installation directory>/tomcat/bin
```

**Note:** If you use a softlink to specify the location of infaservice.sh, set the INFA_HOME environment variable to the location of the Informatica installation directory.

# Troubleshooting Kubernetes

**I tried to build the docker image, but the build image fails.**

If build image fails during installer tar file extraction or before the docker build image starts, verify that adequate disk space is available in the current working directory. Data Engineering Integration requires 50 GB disk space in the current working directory.

If the build image fails at the step `Copy/Binaries/home/Informatica` in the log file, then there is not enough disk space in the docker configuration directory. To find the docker configuration directory, run the docker info command. Data Engineering Integration requires 27 GB disk space in the docker configuration directory.

You can also free up some space from the docker configuration directory when you delete a dangling image. When you create the new build of the image but do not specify a new name, you create a dangling image and the old image becomes the dangling image. Those old images remain untagged and display "<none>" as the name when you run the docker images command.

To delete the dangling images, complete the following steps:

1. To list all dangling images and to get the image ID, run the command: docker images -f dangling=true
2. To delete each dangling image, run the command: docker image rm -f <image id>

If the build image fails with the following error, you need to manually pull the docker image:

```
Sending build context to Docker daemon 25.75GB
Step 1/30: FROM registry.access.redhat.com/rhel7:7.7
Trying to pull repository registry.access.redhat.com/rhel7 ...
Installation Status:ERROR
```

To fix the issue, you can manually run the following command:

```
docker pull registry.access.redhat.com/rhel7:7.7
```

If the build image fails with the following error, the docker build script has failed to access the repository to download dependencies:

```
This system is not registered with an entitlement server.
```

To register and subscribe the machine where you build the docker image:

1. Run the yum repolist all command to see all the repositories.
2. Use the subscription-manager to register with the following command: subscription-manager repos --enable.
3. To enable custom repositories, run the command: yum-config-manager --enable

If the build image displays the authenticationrequired error when you run the docker pull command to pull the image from RHEL, the authentication with the docker authentication might have expired. Run the docker logout command and then pull the docker image. If the issue persists, try to pull the docker image from registry.redhat.io instead of registry.access.redhat.com. If any issue remains with RHEL, you could try to use Centos7 for proof of concept cases.

**I cannot access Administrator tool from the browser for Kubernetes.**

Check the IP address from where you access the Administrator tool:

For on-premises, verify that you specify the internal IP of the node on which the pod is running to access the Administrator tool.

For AWS or Azure, verify that you specify the external IP of the node on which the pod is running to access the Administrator tool.

For Azure, you will not have the external IP of the services by default. The service type of pod must be LoadBalancer instead of NodePort. It is from the LoadBalancer service type that you fetch the external IP.

If you want to verify that the external IP in the Administrator tool for Azure and AWS, run kubectl get nodes -o wide command followed by the kubectl get svc -o wide command.

**How do I restart an already running pod when it goes down?**

When you have a pod that uses the persistent volume, a .yaml file gets generated under the image name folder inside the appconnTemp folder of the user home directory . Run the yaml file manually with the following command to restart the pod with persistent volume: kubectl create -f <file name>.yaml

**How do I to connect to different sources in pods?**

In case of multiple DSN, you can manually update the ODBC.ini file.

# Appendix A: Sample Silent Properties for CI/CD for Build Image

The following example shows the contents of the file:

```
DOWNLOAD_AND_RUN=0
HubUserName=
IMAGE_NAME=informatica1040:1.0
INSTALL_EDP=0
CLUSTER_TYPE=2
HubPassword=
OS_NAME=registry.access.redhat.com/rhel7:7.8
INFA_REPO=0
PRIVATE_HUB=1
INSTALL_OPTION=1
AMBARI_FILE_LOC=
INSTALL_DPM=0
SCANNER_FILE_LOC=
DPM_TAR_FILE_LOC=
productType=PC
TAR_FILE_LOC=/home/
```

# Appendix B: Sample Silent Properties for CI/CD for Run Image

The following example shows the contents of the file:

```
ENABLE_KERBEROS=0
DIS_KEYTAB_FILELOC=
MRS_DB_CUSTOM_STRING=jdbc:informatica:oracle://host_name:port_no;ServiceName=
MRS_KEYTAB_FILELOC=
CMS_KEYSTORE_FILE=
DPS_KEYSTORE_PSSWD=
CMS_DATA_ACCESS_CONNECT_STRING=
OS_NAME=centos:centos7
HDFS_SERVICE_PRINCIPAL=
CREATE_CONNECTION=true
ADVANCE_PORT_CONFIG=0
PRIVATE_HUB=1
EDP_KERBEROS_KEYTAB_FILE=/etc/merge.keytab
EDP_DIS_CUSTOM_SELECTION=false
KSTORE_PSSWD=
EDP_DIS_KEYSTORE_PASSWD=
CLUSTER_PASSWORD=
CMS_DB_UNAME=CPI_CMS
WORKING_DIR=/home
DPS_TRUSTSTORE_DIR=
SPN_SHARE_LEVEL=
YARN_RESOURCE_MANAGER_HTTP_URI=
MAX_PORT=
EDC_KDC_HOST_IP=
DOMAIN_USER=Administrator
LOAD_DATA_DOMAIN=0
CONFIRM_HADOOP_AMBARI_PASSWD=
HADOOP_KEYSTORE_FILE=
DPS_HTTPS_PORT=
```

```
MRS_SERVICE_NAME_EDP=Model_Repository_Service_EDP
TRUSTSTORE_MRS_DB_FILE=
PWH_DB_PASSWD=CPI_PWH
YARN_QUEUE_NAME=
CMS_DB_CUSTOM_STRING=
PWH_DB_ADDRESS=insrh74dsg015.informatica.com:1521
PCRS_DB_HOST=
PWH_DATA_ACCESS_CONNECT_STRING=
EDP_DIS_KEYSTORE_DIR=
IS_CLUSTER_SSL_ENABLE=
EDP_MRS_SSL_DEFAULT_STRING=
HADOOP_IMPERSONATION_USER=
CATALOGUE_SERVICE_KEYSTORE_PASSWD=
EDP_SERVICE_NAME=Enterprise_Data_Preparation_Service
IMPORT_METHOD=2
USE_DB_CONTAINER=false
DPS_CONFIGURE_SERVICES=true
HDFS_SERVICE_NAME_HA=
CMS_SERVICE_NAME=CMS
EDP_PROTOCOL_TYPE=
TRUSTSTORE_DB_FILE=
YARN_RESOURCE_MANAGER_SCHEDULER_URI=
DPS_DB_HOST=
SERVICE_PROVIDER_ID=
TRUSTSTORE_MRS_DB_PASSWD=
PWH_DB_CUSTOM_STRING_SELECTION=0
DB_CUSTOM_STRING=jdbc:informatica:oracle://host_name:port_no;ServiceName=
CATALOGUE_SERVICE_KEYSTORE_FILE=
DIS_CUSTOM_SELECTION=false
INSTALLATION_ENVIRONMENT=Sandbox
INSTALL_TYPE=0
KSTORE_FILE_LOCATION=
LICENSE_KEY_LOC=/root/license.key
EDP_MRS_DB_CUSTOM_STRING_SELECTION=0
TRUSTSTORE_DB_PASSWD=
AMBARI_FILE_LOC=
MRS_DB_TYPE=Oracle
HADOOP_NODES=
AC_PORT=
MONITORING_SSL_DEFAULT_STRING=EncryptionMethod=SSL;HostNameInCertificate=;ValidateServerCertific
ate=false;
USER_REALM_NAME=
HADOOP_TLS_HTTPS_PORT=
KERBEROS_KEYTAB_FILE=/etc/merge.keytab
DIS_HTTPS_PORT=18095
INSTALL_DPM=0
CMS_DB_CUSTOM_STRING_SELECTION=0
PWH_DB_TYPE=Oracle
PCRS_DB_PSSWD=
DPS_CUSTOM_HTTPS_ENABLED=false
DPM_TAR_FILE_LOC=
NODE_TRUSTSTORE_DIR=
CUSTOM_HTTPS_ENABLED=0
LOAD_TYPE=low
USER_INSTALL_DIR=/home/Informatica
PCRS_DB_TYPE=Oracle
DPS_PROTOCOL_TYPE=
HADOOP_TRUSTSTORE_FILE=
DPM_LICENSE_KEY_LOC=/data2/EDC_Resources/DataPrivacyManagement.key
PWH_SQLSERVER_SCHEMA_NAME=
NEW_HADOOP_AMBARI_PASSWD=
DB_CONTAINER_NAME=
EDP_CUSTOM_HTTPS_ENABLED=false
CMS_KEYSTORE_PASSWD=
DPS_DB_USER=
HADOOP_GATEWAY_HOST=
SECURITY_DOMAIN_NAME=
MONITORING_KEYTAB_FILELOC=
MRS_DB_PASSWD=
DPS_DB_TRUSTSTORE_PASSWD=
EDP_TRUSTSTORE_PSSWD=
```

```
KRB5_FILE_LOCATION=/etc/krb5.conf
CMS_ADVANCE_JDBC_PARAM=
EDP_DIS_HTTPS_PORT=
VOLUME_DIRECTORY=/root/swathi/shared
MRS_ADVANCE_JDBC_PARAM=
EDP_DPS_SERVICE_NAME=Interactive_Data_Preparation_Service
AC_SHUTDWN_PORT=
CREATE_PCRS=false
SSL_ENABLED=false
DIS_SERVICE_NAME_EDP=Data_Integration_Service_EDP
LOCAL_STORAGE_DIR=/home/toolprod/satish/temp
CLOUD_SUPPORT_ENABLE=0
IS_SERVICE_NAME=
HTTPS_PORT=
CATALOGE_SERVICE_KEYTAB_LOCATION=
HDFS_HOST_NAME=
MONITORING_DB_ADDRESS=
DEFAULT_HTTPS_ENABLED=1
CLUSTER_HADOOP_DISTRIBUTION_TYPE=
EDL_KRB5_FILE_LOCATION=/etc/krb5.conf
DB_PASSWD=admin
EDP_MRS_DB_UNAME=
SCANNER_FILE_LOC=
SSH_HOST_NAMES=
NODE_KEYSTORE_PASSWD=
DPS_KEYSTORE_DIR=
CMS_DB_TYPE=Oracle
INFA_CONTAINER_NAME=infa
EDP_MRS_DB_ADDRESS=HostName:PortNumber
SAML_TRUSTSTORE_PASSWD=
DPS_HTTP_PORT=
DIS_PROTOCOL_TYPE=http
HDFS_SYSTEM_DIR=/datalake/EDL
EDP_MRS_SERVICE_NAME=Model_Repository_Service_EDP
HADOOP_CUSTOM_SELECTION=false
DPS_KDC_HOST_IP=
MIN_PORT=
CREATE_LAKE_SERVICES=0
SAML_TRUSTSTORE_ALIAS=informatica llc
EDP_DIS_SERVICE_NAME=Data_Integration_Service_EDP
CLUSTER_HADOOP_DISTRIBUTION_URL=
KEYTAB_LOCATION=
IHS_ADMINISTRATOR_PASSWORD=
DPS_CLUSTER_TRUSTSTORE_FILE_LOC=
SILENT_FILE_LOC=/root/appconTemp/informaticaltd/dei/SilentInput.properties
ENABLE_EDP_SERVICE=true
EDP_KERBEROS_PRINCIPAL=pam_user@PLATFORMKRB.COM
DOWNLOAD_AND_RUN=0
DB_SSL_ENABLED=false
SECURITY_MODE=Kerberos
EDP_MRS_DB_CUSTOM_STRING=
MRS_DB_SSL_ENABLED=false
DPS_DB_TRUSTSTORE_FILE=
EDP_DIS_TRUSTSTORE_PASSWD=
SAML_TRUSTSTORE_DIR=
CATALOGUE_SERVICE_NAME=catalog_service
CATALOGUE_SERVICE_SOLR_KEYSTORE_FILE=
CMS_SERVICE_NAME_ASSOCIATED=CMS
KEY_SRC_LOCATION=
TRUSTSTORE_MONITORING_DB_PASSWD=
CATALOGUE_SERVICE_PORT=8180
DISTRIBUTION_TYPE=cdh
NODE_TRUSTSTORE_PASSWD=
IDP_URL=info@informatica.com
HubUserName=
IHS_ADMINISTRATOR_PRINCIPAL=
CLUSTER_CONFIG_NAME=
CLUSTER_NAME=
DPS_NODE_NAME=Node01
MONITORING_DB_SSL_ENABLED=false
TRUSTSTORE_MONITORING_DB_FILE=
```

```
INSTALL_EDP=0
HubPassword=
ENABLE_USAGE_COLLECTION=1
UPGRADE_WITHOUT_PC=1
CONTAINER_DN=
INFA_REPO=0
LOCAL_SYSTEM_DIR=/home/toolprod/satish/temp
CREATE_DOMAIN=1
SQLSERVER_SCHEMA_NAME=
PWH_DB_SERVICENAME=SSL12CR2.informatica.com
CLUSTER_PORT=
CMS_HTTP_PORT=9050
SERVICE_CLUSTER_NAME=pam_user
HADOOP_KEYSTORE_PASSWD=
DPS_SSL_DEFAULT_STRING=
CATALOGUE_SERVICE_KEYSTORE_ALIAS=
IS_CLUSTER_HA_ENABLE=
DPS_DB_TYPE=
CMS_CUSTOM_SELECTION=false
EDP_TRUSTSTORE_MRS_DB_PASSWD=oracle4u
DOMAIN_PSSWD=admin
EDP_HTTPS_PORT=
MONITORING_DB_SERVICENAME=
DB_ADDRESS=oracle
DIS_SERVICE_NAME=Data_Integration_Service
HTTPS_ENABLED=0
SERVES_AS_GATEWAY=0
EDP_MRS_SQLSERVER_SCHEMA_NAME=
DOMAIN_HOST_NAME=infa
MONITORING_SQLSERVER_SCHEMA_NAME=
PCRS_DB_PORT=1521
NODE_KEYSTORE_DIR=
SERVER_PORT=
DOCKER_NETWORK_NAME=
OVERRIDE_HADOOP_AMBARI_PSD=false
KDC_CERT_FILE_LOC=
DIS_HTTP_PORT=
EDP_MRS_ADVANCE_JDBC_PARAM=
MRS_DB2_TABLESPACE=
EDP_DIS_PROTOCOL_TYPE=
DPS_DB_PORT=
EDP_MRS_DB2_TABLESPACE=
AUTH_MODE=Kerberos
NODE_NAME=node01
CLUSTER_HADOOP_DISTRIBUTION_URL_PASSWD=
DB2_TABLESPACE=
KERBEROS_CONF_FILE_LOC=/etc/krb5.conf
ENABLE_CUSTOM_SERVICE_PROVIDER_ID=0
KERBEROS_DOMAIN_PSSWD=
INSTALL_LDM=0
CMS_DB_SERVICENAME=SSL12CR2.informatica.com
MONITORING_DB_CUSTOM_STRING=
CMS_SQLSERVER_SCHEMA_NAME=
EDP_TRUSTSTORE_MRS_DB_FILE=/home/toolprod/INFA_Automation/newssldb/keystore_orassl.jks
HDFS_PRINCIPAL_NAME=nn/_HOST@PLATFORMKRB.COM
KEY_DEST_LOCATION=/home/Informatica/isp/config/keys
MRS_SERVICE_NAME=Model_Repository_Service
DB_UNAME=admin
ENABLE_CMS_SERVICE=true
CLUSTER_TYPE=2
EDP_DIS_TRUSTSTORE_DIR=
SERVICE_REALM_NAME=
KDC_KEYTAB_LOCATION=/etc/merge.keytab
SAML_AUTHENTICATION=false
MONITORING_DB_UNAME=
DIS_TRUSTSTORE_DIR=
CLUSTER_CONFIG_VALIDATION=0
HADOOP_GATEWAY_PORT=8080
HOST_SSH_USER=root
MRS_SQLSERVER_SCHEMA_NAME=
DOMAIN_PORT=6005
```

```
DB_SERVICENAME=ora
RESUME_INSTALLATION=false
EDP_TRUSTSTORE_DIR=
DPS_DB_CUSTOM_STRING_SELECTION=false
PWH_ADVANCE_JDBC_PARAM=
EDP_MRS_DB_SERVICENAME=DBServiceName
ENABLE_DB_SSL=false
DPS_DB_SCHEMA=
CONFIGURE_SERVICES=1
MONITORING_ADVANCE_JDBC_PARAM=
EDP_HTTP_PORT=
MRS_SSL_DEFAULT_STRING=EncryptionMethod=SSL;HostNameInCertificate=;ValidateServerCertificate=fal
se;
TUNING_SIZE=sandbox
DPS_TRUSTSTORE_PSSWD=
KDC_HOST=
DOMAIN_CNFRM_PSSWD=admin
JOIN_NODE_NAME=
DOMAIN_NAME=Domain
PCRS_DBTYPE_AZURE=0
CATALOGUE_SERVICE_SOLR_KEYSTORE_PASSWD=
DB_TYPE=Oracle
DIS_TRUSTSTORE_PASSWD=
KERBEROS_SECURITY_DOMAIN_NAME=
PASSWORD_COMPLEXITY=false
MRS_DB_SERVICENAME=
TAR_FILE_LOC=
ARCHIVE_FILE=
PCRS_SERVICE_NAME=
INFA_SERVICES_INSTALLED=false
DIS_KEYSTORE_PASSWD=
DPS_CLUSTER_TRUSTSTORE_FILE_PASSWD=
YARN_SERVICE_PRINCIPAL=
CMS_HTTPS_PORT=
CATALOGUE_SERVICE_TLS_HTTPS_PORT=
EDP_MRS_DB_TYPE=
KDC_TYPE=mit-kdc/active-directory
MRS_DB_ADDRESS=
MRS_DB_CUSTOM_STRING_SELECTION=0
CMS_PROTOCOL_TYPE=http
PWH_DB_CUSTOM_STRING=
IHS_ADMINISTRATOR_SERVER_HOST=
CATALOGUE_SERVICE_CUSTOM_SELECTION=false
MONITORING_DB_PASSWD=
DPS_DIS_SERVICE_NAME=Data_Integration_Service_EDP
JOIN_DOMAIN=0
TLS_CUSTOM_SELECTION=false
HIVE_LOCALSTORAGE_FORMAT=DefaultFormat
PORT_RANGE=6005-6115
ENABLE_DPS_SERVICE=true
IMAGE_NAME=informaticaltd/dei:1040-rc336.update
CATALOGUE_SERVICE_NAME_ASSOCIATED=catalog_service
TUNE_APPLICATION_SERVICES=true
EDP_MRS_DB_PASSWD=
DPS_MRS_SERVICE_NAME=Model_Repository_Service_EDP
ENABLE_BIGDATA_JOB_RECOVERY=
DB_CUSTOM_STRING_SELECTION=0
EDP_NODE_NAME=Node01
MONITORING_DB2_TABLESPACE=
CREATE_MONITORING_STATS=0
CLUSTER_NAME_CCO=
CREATE_SERVICES=0
IHS_REALM=
EDP_KEYSTORE_DIR=
MONITORING_SERVICE_NAME=
EIC_SERVICES_INSTALLED=false
DPS_SERVICE_NAME=Interactive_Data_Preparation_Service
KDC_DOMAIN_NAME=PLATFORMKRB.COM
ZOOKEEPER_URI=
JOIN_DOMAIN_PORT=
CLUSTER_USERID=
```

```
DPS_DB_PSSWD=
EDP_MRS_DB_SSL_ENABLED=
LDAP_URL=
YARN_RESOURCE_MANAGER_URI=
EDP_DIS_HTTP_PORT=
MRS_DB_UNAME=admin
DPS_DB_SERVICENAME=
PCRS_DB_SERVICE_NAME=
HADOOP_KEYSTORE_ALIAS=
INSTALL_OPTION=2
MONITORING_DB_TYPE=Oracle
CREATE_CCO_CONNECTION=0
IS_CLUSTER_SECURE=false
DIS_KEYSTORE_DIR=
HDFS_LOCATION=/datalake/DPS
CATALOG_ENABLE_EMAIL_SERVICE=false
JOIN_HOST_NAME=
HADOOP_SERVICE_PORT=8160
PWH_DB2_TABLESPACE=
EXTERNAL_CLUSTER_TRUSTSTORE_FILE_LOC=
RUN_CLUSTER_PREVALIDATION=0
CLUSTER_HOST=
HADOOP_SERVICE_NAME=hadoop_service
PWH_DB_UNAME=CPI_PWH
productType=PC
CMS_DB_PASSWD=CPI_CMS
DOMAINMACHINE_PASSWORDLESS_SSH_ENABLED=0
PASS_PHRASE_PASSWD=key
SLIDER_MAX_PORT=
TRUSTED_CONNECTION=
CMS_DB2_TABLESPACE=
ASSOCIATE_PROFILE_CONNECTION=0
CMS_DB_ADDRESS=insrh74dsg015.informatica.com:1521
PCRS_DB_USER=
DPS_DB_CUSTOM_STRING=
CREATE_EDP_SERVICES=0
ADVANCE_JDBC_PARAM=
CLUSTER_HADOOP_DISTRIBUTION_URL_USER=
HISTORY_SERVER_HTTP_URI=
EDP_KEYSTORE_PSSWD=
HADOOP_LOG_DIR=
MONITORING_DB_CUSTOM_STRING_SELECTION=0
DOMAIN_KEYSTORE_ALIAS=
TOMCAT_PORT=
EXTERNAL_CLUSTER_TRUSTSTORE_FILE_PASSWD=
GATEWAY_USERNAME=root
SLIDER_MIN_PORT=
```

# Author

**Sujitha Alexander**