How-To Library

Informatica

Install Data Engineering Integration 10.5 on Kubernetes with Informatica Deployment Manager

Abstract

Informatica Deployment Manager provides a quick and easy way to install and manage the Informatica domain. This article describes how to install Data Engineering Integration on Kubernetes from the Docker image using Informatica Deployment Manager. This article also describes how you can use Informatica Deployment Manager to manage an existing Data Engineering Integration deployment on Kubernetes.

Supported Versions

- Data Engineering Integration 10.5

Table of Contents

Overview

Use the web-based interface in Informatica Deployment Manager to quickly create standard Docker images and 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 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.

Note: Informatica Deployment Manager supports only Azure Kubernetes Service as the Kubernetes platform.

You can use Informatica Deployment Manager to build the Docker image. You can build the Docker image with the default base or custom operating system and Informatica product binaries. With Informatica Deployment Manager, you can create containers using this Docker image to configure the Informatica domain.

When you run the Docker image, you can install Informatica products and you can create a domain.

With Informatica Deployment Manager, you can build the Docker image with the silent installer. If you want to run the silent installer, you can choose to generate the silent input properties when you go through the wizard steps, and then copy the properties directly to the silent install properties file. You can create the application services when you run the Docker image.

Informatica Deployment Manager 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 Informatica Deployment Manager and access the Informatica installer .tar file.
  - Verify the license key.
  - Set up repository databases.
  - Prepare for the cluster configuration.
  - Prepare for Kubernetes configuration.
  - Automate the generation of the silent input property file.
- ❑ Build the Informatica Docker image with base operating system and Informatica binaries.
- ❑ Run the Docker image to configure the Informatica domain.
- ❑ Build the Docker image in silent mode.
- ❑ Complete the post-requisites:
  - Complete the domain configuration.
  - Create the application services.
  - Install the Developer tool.

Before You Begin

Before you install Informatica Deployment Manager, 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/rhel7:7.8` 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 Deployment Manager and the Informatica installer.tar file to install the product.

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

After you extract the `informatica_1050_deployment_manager.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.

Note: The name of the folder that contains the Informatica Deployment Manager installer files must not contain any space.

To gather inputs for the silent input properties files, launch Informatica Deployment Manager and click **Generate silent properties**. 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.

Note: You do not need a license for Informatica Deployment Manager.

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. Informatica certifies that you can create the Model repository and the monitoring Model repository on a Microsoft Azure SQL database. You can also create the Model repository and the monitoring Model repository on an Oracle, PostgreSQL, and Microsoft SQL Server database if you can ensure proper connectivity between the database and the Kubernetes cluster.

You can either create an on-premises database or use a managed database for each repository. When you run the Docker image, you will provide the database information.

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:

Microsoft Azure SQL

The Microsoft Azure SQL database is certified by Informatica to create the Model repository. Complete the following tasks to prepare the Microsoft Azure SQL 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.

Oracle

You can use an Oracle database if you can ensure proper connectivity between the database and the Kubernetes cluster. 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

You can use a PostgreSQL database if you can ensure proper connectivity between the database and the Kubernetes cluster. 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

You can use a Microsoft SQL Server database if you can ensure proper connectivity between the database and the Kubernetes cluster. 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.

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 Informatica Deployment Manager:

1. Install kubectl.
2. 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 Content for Silent Input Properties File

When you create or build the Docker image in Informatica Deployment Manager, you can generate the property values to run the silent installer. Before you can run the silent installer, click **Generate silent properties** in Informatica Deployment Manager. Use the generated 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.

In Informatica Deployment Manager, you can use the option to generate the silent properties to copy all the user input values entered to build or run the Docker image from a single panel. The generated contents are in the supported format of the silent install property files. Manually save the generated 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, Informatica Deployment Manager uses the Docker image to run the containers. Ensure that the Informatica Docker image is stored in the local host or Docker registry before you install Informatica products in Docker with Informatica Deployment Manager.

Step 1. Choose the Deployment Type

On the **Deployment type** page, you must select the Informatica product to deploy. Additionally, you can choose to configure secure authentication to build the Docker image on a remote host.

1. Launch Informatica Deployment Manager. You can choose to launch Informatica Deployment Manager in a secure way using SSL. If you choose to use SSL, you can use the default Informatica SSL certificate or your SSL certificate.
   - To launch without SSL, run `sh startup.sh nonssl`.
   - To launch with the default SSL certificate, run `sh startup.sh`.
   - To launch with your SSL certificate, run `sh startup.sh SSL <certificate location> <certificate password>`. The password must be in single quotes.

   Note: By default, Informatica Deployment Manager uses four ports that are available within the port range of 12100 to 12200. However, you can configure Informatica Deployment Manager to use the available ports from a custom port range. To specify a custom port range, you must replace the port numbers of the default port range in the `startup.sh` file with the port numbers of the custom port range. If you are using Informatica Deployment Manager on Windows platform, you must run the `dos2unix` command to convert the modified `startup.sh` file to Unix format.
2. To build the Informatica Docker image, select **Build image**.

   The **Deployment type** page appears.
3. On the **Deployment type** page, select **Data Engineering Integration** for the product.

   The build image uses Docker as the default deployment type.
4. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to build the image on a remote host. If you do not want to set the secure authentication, 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.
     By default, the SSH connection uses port 22.
   - 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 **Configure Docker image** page appears.

Step 2. Configure the Docker Image

On the **Configure Docker image** page, enter information for the tar file location and the images.

1. Specify the base operating system to build the Docker image.

- To build the Docker image on CentOS, select `centos:7`.
- To build the Docker image on RHEL, select `registry.access.redhat.com/rhel7:7.8`.
- To build the Docker image on a custom operating system, select `Custom`, and then enter the name of the operating system.

2. Enter the Informatica installer tar file path on the machine where you want to build the image.

   The Informatica Deployment Manager uses the tar utility to extract the installer server files to a directory on the machine.

3. Specify the name of the Docker base image.

   Default is `informatica1041:1.0`, where `1.0` is the tag name.

4. Enter the machine working directory to build the Docker image..

   Default is `/root`.

5. Click **Next**.

   The **Configure domain services** page appears.

Step 3. Configure the Domain Services

On the **Configure domain services** page, you must specify the domain services that you want to include in the Docker image.

1. Select the type of image to build.
   - To build the image with the Informatica services, select **Domain**. You can specify this option for installation of the domain and the core services that support the domain.
   - To build the image with all the services, select **Services**. You can specify this option to include the Model Repository Service, Data Integration Service, Email Service, and other additional services in the image.
   - To build the image with specific domain services, select the services that you want to install. When you select specific services instead of including all the services, the image consumes lesser disk space compared to the installation of all the services.

     Note: You can select **Other services** to install the additional services only after you select all other domain services for installation.

2. Optionally, you can choose to generate the silent properties for silent installation.

   When you generate the silent properties from Informatica Deployment Manager, you can also copy all the installation options and the values specified in the deployment manager and later paste them in the property file for the silent installer.

3. Click **Build**.

   The process of building the image starts.

After the Docker image is built, you can view the status and name of the image, along with the tasks performed by the installer in the log summary.

Step 4. 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
informatica1050pcrspostgre    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 Informatica product and version to deploy. Additionally, you can choose to configure secure authentication to run the Docker image and deploy the product on a remote host.

1. Launch Informatica Deployment Manager. You can choose to launch Informatica Deployment Manager in a secure way using SSL. If you choose to use SSL, you can use the default Informatica SSL certificate or your SSL certificate.
   - To launch without SSL, run `sh startup.sh nonssl`.
   - To launch with the default SSL certificate, run `sh startup.sh`.
   - To launch with your SSL certificate, run `sh startup.sh SSL <certificate location> <certificate password>`. The password must be in single quotes.

   Note: By default, Informatica Deployment Manager uses four ports that are available within the port range of 12100 to 12200. However, you can configure Informatica Deployment Manager to use the available ports from a custom port range. To specify a custom port range, you must replace the port numbers of the default port range in the `startup.sh` file with the port numbers of the custom port range. If you are using Informatica Deployment Manager on Windows platform, you must run the `dos2unix` command to convert the modified `startup.sh` file to Unix format.
2. To run the Informatica Docker image, select **Run image**.

   The **Deployment type** page appears.
3. On the **Deployment type** page, select **Data Engineering Integration** for the product.
4. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to deploy the containers on a remote host. If you do not want to set the secure authentication, 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.
     By default, the SSH connection uses port 22.
   - 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 **Domain details** page appears.

Step 2. Provide Docker and Domain Configuration Repository Information

In the **Domain and Docker details** section of the **Domain details** page, provide the domain configuration repository database information and the required information to connect to the Docker image.

1. Specify the Docker image name, Informatica license key file, and path of the working directory.
2. Enter the name, user name, and password of the Docker server that contains the container registry.
3. Enter the name and resource group of the cluster storage account to be used as shared storage for the Informatica domain.
4. Select the database to use for the domain configuration repository, and then enter the database user account, password, and the JDBC connection string to connect to the database.

   You can use the following syntax in the JDBC connection string:
   Oracle

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

   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>
   ```

Step 3. Configure the Domain

In the **Domain configuration** section of the **Domain details** page, provide the provide the Informatica domain details, connections details for Informatica Administrator, and whether to secure communication to the domain or not. You can also specify the application services that you want to deploy with the domain.

1. If an Informatica domain is already deployed on the host machine and if you want to replace the deployed domain with the new domain, choose to overwrite the existing domain.
2. Specify the name, user name, password, port number, and gateway node count of the domain.

   The default values are as follows:

   - **Domain name**: domain
   - **User name**: Administrator
   - **Port**: 8075
   - **Gateway node count**: 1
3. Optionally, specify the range of dynamic port numbers that can be used by the supported application services. You can also choose to add host alias names, set environment variables, and expose additional ports for the domain.
4. If you choose to add host alias names, specify the names and IP addresses of the hosts.
   a. Click **Add host names**.

      The **Host Alias Names** window appears.

b. Click **New**.

   The **New Host Alias** window appears.

c. Enter the IP address and host names, and click **OK**.

   The IP address and host names appear on the **Host Alias Names** window.

d. Repeat steps b and c for each host lias name that you want to add for the domain.

e. Click **OK**.

5. If you choose to set additional environment variables, specify the name and value of the variables.

   a. Click **Add variables**.

      The **Environment variables** window appears.

   b. Click **New**.

      The **New environment variable** window appears.

   c. Enter the name and value of the variable that you want to set for the domain, and click **OK**.

      The variable name and value appears on the **Environment variables** window.

   d. Repeat steps b and c for each variable that you want to set.

   e. Click **OK**.

6. If you choose to expose additional ports, specify the name and number of the ports to expose.

   a. Click **Add ports**.

      The **Additional ports** window appears.

   b. Click **New**.

      The **New port** window appears.

   c. Enter the name and number of the port that you want to expose in the domain, and click **OK**.

      The port name and value appears on the **Additional ports** window.

   d. Repeat steps b and c for each port that you want to expose.

   e. Click **OK**.

7. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources available for the container.

8. By default, secure communication is enabled for the Informatica domain. However, you can choose to disable it. If secure communication is enabled for the domain, specify whether to use the default Informatica SSL certificates or to use your SSL certificates to secure domain communication.

9. If you choose to provide the SSL certificates, specify the keystore and truststore files and their passwords.

10. By default, secure HTTPS connection is enabled for Informatica Administrator. However, you can choose to disable it. If HTTPS connection is enabled for Informatica Administrator, 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 keystore file and its password.

11. Optionally, choose to create the supported application services while deploying the Docker image. However, you can run the image without creating the supported application services. Perform one of the following steps based on whether you want to create the application services while you configure the domain or later:

    - To create the application services, select the services to create, and click **Next**.
      The pages to specify the database and connection information for the services appear.

- To deploy Data Engineering Integration without the supported application services, click **Deploy**. The Docker image runs. After the Docker image run completes, you can view the post-installation summary. You can view the installation log to get more information about the tasks performed by the installer.

Step 4. Configure the Model Repository Service

On the **Model Repository Service** page, you can provide the connectivity information for the Model Repository Service and the database information for the Model repository.

1. Enter the name of the Model Repository Service. Default value is mrs.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```

   - You cannot change the name of the service after it is created.
2. Enter the name, user name, and password of the Informatica domain. The default values for domain name and user name are domain and Administrator, respectively.
3. Enter the working directory of the service.
4. Enter the image name and HTTP port of the node on which the Model Repository Service needs to run. The default HTTP port is 8075.
5. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you include the existing variables, ports, or host alias names, respectively, in the list. This prevents the existing domain configurations from getting affected.
6. Specify the type, user ID, and user password of the Model repository database.
7. Enter the parameters for the database.
   a. If you select Microsoft SQL Server, you can choose to enter the schema name for the database. If you select Oracle or PostgreSQL, 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, 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>
```

PostgreSQL

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

Microsoft SQL Server

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

8. 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;
   ```

9. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
10. Click **Next** to configure additional services, or run the Docker image.

Step 5. Configure the monitoring Model Repository Service

On the **Monitoring Model Repository Service** page, you can provide the connectivity information for the monitoring Model Repository Service and the database information for the monitoring Model repository.

1. Enter the name of the monitoring Model Repository Service. Default value is mmrs.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```

   - You cannot change the name of the service after it is created.
2. Enter the name, user name, and password of the Informatica domain. The default values for domain name and user name are domain and Administrator, respectively.
3. Enter the working directory of the service.
4. Enter the image name and HTTP port of the node on which the monitoring Model Repository Service needs to run. The default HTTP port is 8075.
5. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you include the existing variables, ports, or host alias names, respectively, in the list. This prevents the existing domain configurations from getting affected.
6. Specify the type, user ID, and user password of the monitoring Model repository database.
7. Enter the parameters for the database.

   a. If you select Microsoft SQL Server or PostgreSQL, you can choose to 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, 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>
```

PostgreSQL

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

Microsoft SQL Server

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

8. 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;`

9. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
10. Click **Next** to configure additional services, or run the Docker image.

Step 6. Configure the Data Integration Service

On the **Data Integration Service** page, you can configure the service parameters for the Data Integration Service.

1. Enter the name of the Data Integration Service. Default value is dis.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```

   - You cannot change the name of the service after it is created.

2. Specify the domain name, working directory, and worker node count for the application service. The default value for domain name is domain.
3. Enter the image name and HTTP port of the node on which the Data Integration Service needs to run. The default HTTP port is 8075.
4. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you include the existing variables, ports, or host alias names, respectively, in the list. This prevents the existing domain configurations from getting affected.

5. Enter the name of the Model Repository Service associated with the Data Integration Service. Also, enter the user name and password of the user account that has permissions to manage the Model Repository Service.
6. Optionally, choose to enable 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.
7. If you choose to enable Kubernetes autoscaling, enter the worker node, CPU utilization, and memory utilization information to autoscale:
   - **Minimum number of nodes**. The minimum number of worker nodes to create in the domain. Default is 2.
   - **Maximum number of nodes**. The maximum number of worker nodes to create in the domain. Default is 4.
   - **CPU load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the CPU percentage metrics. Default is 75 percent.
   - **Memory load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the memory percentage metrics. Default is 80 percent.
8. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
9. Click **Next** to configure additional services, or run the Docker image.

Step 7. Configure the Email Service

On the **Email Service** page, you can configure the service parameters for the Email Service.

1. Enter the name of the Informatica domain.
2. Enter the image name and HTTP port of the node on which the Email Service needs to run.
3. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you include the existing variables, ports, or host alias names, respectively, in the list. This prevents the existing domain configurations from getting affected.
4. Enter the name of the Model Repository Service associated with the Email Service. Also, enter the user name and password of the user account that has permissions to manage the Model Repository Service.
5. Enter the working directory of the service.
6. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
7. Click **Deploy**.

   The Docker image runs.

Step 8. Verify the Informatica Container and Image

You can verify whether the Informatica Docker container is present in the host specified and whether the correct Docker images are being deployed.

1. 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:1050pc  "/Installer/launcher..." 2 days
ago  Up 2 days
```

2. 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 installer.

Configure the Docker Image in Silent Mode

You can build or run the Docker image in silent mode after updating the required properties files with the generated silent properties from Informatica Deployment Manager.

To run the installer in silent mode, complete the following tasks:

1. Run Informatica Deployment Manager and choose to generate the silent properties.
2. Configure the silent input properties file and specify the installation options.

   When you generate silent properties from Informatica Deployment Manager, you can copy all the installation options and the values specified in Informatica Deployment Manager. You can then paste the copied contents into the property file for the silent installer.
3. Save the properties file in the UNIX format to the file path that contains the silent installer.
4. Run the silent installer with the silent installation properties file.

If you want to run Informatica Deployment Manager for additional images, you must save each property file before every run of the silent installer.

Running the Silent Installer

Manually copy the generated silent properties when you build or run the Docker image from Informatica Deployment Manager into the supported silent properties file. The customized `SilentInput_BuildImage.properties` or `SilentInput.properties` file has different content based on whether you run the silent installer to build or run the Docker image. After you configure the properties file, open a command prompt to start the silent installation.

1. Run Informatica Deployment Manager and choose to generate the silent properties.
2. Copy the generated silent properties from Informatica Deployment Manager into the supported silent properties file when you build or run the Docker image.
3. From the terminal, go to the root of the directory that contains the installation files. For build image, navigate to the following directory: `<working directory>/appconTemp/<image name>`. For run image, navigate to the following directory: `<user home>/appconTemp/<image name>`.
4. Verify that the directory contains the customized `SilentInput_BuildImage.properties` or `SilentInput.properties` file based on whether you build or run the docker image.
5. From the terminal, run the `dos2unix` command to convert the silent input properties file to the Unix format.
6. Create a log directory: `mkdir logs`
7. Run sh silentInstall.sh >> logs/<debug configure>.log.

When the silent installation is complete, you can view the log file in the working directory.

Complete the Post-Install Tasks

After you deploy Data Engineering Integration, 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.

Start and Stop 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.

About Managing Existing Deployments

In addition to deploying Data Engineering Integration, you can also manage existing deployments of Data Engineering Integration using Informatica Deployment Manager.

You can manage and update the domains, application services, and nodes associated with existing Data Engineering Integration deployments running on Kubernetes. If Informatica releases an Emergency Bug Fix (EBF) that is applicable for the deployed product, you can also apply the EBF using Informatica Deployment Manager.

Manage Domains

You can update, shut down, and restart Data Engineering Integration domains using Informatica Deployment Manager. You can also download the logs from the nodes in a domain.

Update a Domain

You can update the gateway count, node image, and pod resources of a domain. You can also set additional environment variables and expose additional ports for the domain.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Domain** panel, click **Update domain**.

   The **Update domain** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to update the domain on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Update domain** section, enter the domain name, domain working directory, and the name, user name, and password of the Docker server that hosts the domain.
6. Optionally, specify the gateway node count and node image name. You can also choose to set additional environment variables and expose additional ports for the domain.
7. If you choose to set additional environment variables, specify the name and value of the variables.
   a. Click **Set environment variables**.

      The **Environment variables** window appears.
   b. Click **New**.

      The **New environment variable** window appears.
   c. Enter the name and value of the variable that you want to set for the domain, and click **OK**.

      The variable name and value appears on the **Environment variables** window.

d. Repeat steps b and c for each variable that you want to set. You must also repeat steps b and c for each environment variable that is already set for the domain. This prevents the existing variables from getting deleted.

e. Click **OK**.

8. If you choose to expose additional ports, specify the name and number of the ports to expose.

   a. Click **Add ports**.

      The **Additional ports** window appears.

   b. Click **New**.

      The **New port** window appears.

   c. Enter the name and number of the port that you want to expose in the domain, and click **OK**.

      The port name and value appears on the **Additional ports** window.

   d. Repeat steps b and c for each port that you want to expose. You must also repeat steps b and c for each port that is already exposed for the domain. This prevents the existing ports from getting closed in the domain.

   e. Click **OK**.

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

   By default, if you enable secure communication for the domain, an HTTPS connection is set up for Informatica Administrator.

10. 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, select the keystore file and its password.

11. If you enabled secure connection for the domain, specify whether to use the default Informatica SSL certificates or to use your SSL certificates to secure domain communication.

12. If you choose to provide the SSL certificates, specify the keystore and truststore files and their passwords.

13. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources available for the container.

14. Click **Update Domain**.

    The domain update starts. You can view whether the domain is updated in the log summary.

Shut Down a Domain

To run administrative tasks on a domain, you can shut down the domain. When you shut down a domain, the Service Manager on the master gateway node stops all application services and Informatica services in the domain. Any service processes running on nodes in the domain are also aborted.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.

2. On the **Domain** panel, click **Shut down domain**.

The **Shut down domain** page appears.

3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to shut down the domain on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Domain information** section, enter the name and working directory of the domain to shut down.
6. Choose whether to stop the application services, Informatica services, and service processes running on domain nodes or to allow them to complete their operation before the domain is shut down.
7. Click **Shut down domain**.

   The Service Manager on the master gateway node stops the application services, Informatica services, and service processes on each node in the domain and shuts down the domain. You can view whether the domain is shut down in the log summary.

Restart a Domain

You can restart a domain running on Kubernetes using Informatica Deployment Manager. When you restart a domain, the application services, Informatica services, and service processes on the gateway and worker nodes in the domain are restarted.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. In the **Domain** panel, click **Restart domain**.

   The **Restart domain** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to restart the domain on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Domain information** section, enter the name and working directory of the domain to shut down.
6. Choose whether to abort the application services, Informatica services, and service processes running on domain nodes or to allow them to complete their operation before the domain is restarted.
7. Click **Restart domain**.

   The domain and the application services, Informatica services, and service processes running on its nodes are restarted. You can view whether the domain is restarted in the log summary.

Download Logs from a Domain Node

You can use Informatica Deployment Manager to download the logs form the nodes in an existing domain.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Domain** panel, click **Download logs**.

   The **Download logs** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to download the logs from a node on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Domain information** section, enter the domain name and the working directory of the node.
6. Click **Download logs**.

The `<domain_name>_INFAK8s_Logs` folder is created in the working directory of Kubernetes executor and the logs of the node are downloaded to the newly created folder. Additionally, the node initiator, node manager, and node terminator log files from all pods are downloaded to the `<domain_name>_K8s/<pod_name>` directory.
The `<domain_name>_INFAK8s_Logs` folder contains the following files:

- The `k8s_objects.desc` file that contains the description of all Kubernetes resources except the Secrets.
- The domain log files of the shared volume. The domain log files are stored in the `LogService` folder within the `<domain_name>_INFAK8s_Logs` folder.
- The logs in the `SystemLog` directory of all pods.

Manage Application Services

You can create, update, and remove the application services in Data Engineering Integration domains using Informatica Deployment Manager.

Create a Service

You can create an application service in an existing domain.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Service** panel, click **Create service**.

   The **Create service** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to create the service on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.

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

5. In the **Service details** section, enter the name, user name, and password of the domain.
6. On the **Service** drop-down, select the service to create.

   The fields to provide the configuration details of the service appear.
7. Depending on the service that you selected, provide the details to configure the service on the same page.

Configure the Model Repository Service

In the **Model Repository Service information** section of the **Create service** page, you can provide the connectivity information for the Model Repository Service and the database information for the Model repository.

1. Enter the name of the Model Repository Service. Default value is mrs.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character @. It must also not contain spaces or the following special characters:

     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( )] [

   - You cannot change the name of the service after it is created.
2. Enter the name, user name, and password of the Informatica domain. The default value for domain name is domain and the default values for user name is Administrator.
3. Enter the working directory of the service.
4. Enter the image name and HTTP port of the node on which the Model Repository Service needs to run. The default HTTP port is 8075.
5. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.
6. Specify the type, user ID, and user password of the Model repository database.
7. Enter the parameters for the database.
   a. If you select Microsoft SQL Server, you can choose to enter the schema name for the database. If you select Oracle or PostgreSQL, 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, 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>
```

PostgreSQL

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

Microsoft SQL Server

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

8. 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;
   ```

9. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
10. Click **Create service**.

The service is created. You can view the status of the service along with the tasks performed by the installer in the log summary.

Configure the monitoring Model Repository Service

In the **Monitoring Model Repository Service information** section of the **Create service** page, you can provide the connectivity information for the monitoring Model Repository Service and the database information for the monitoring Model repository.

1. Enter the name of the monitoring Model Repository Service. Default value is mmrs.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```

   - You cannot change the name of the service after it is created.
2. Enter the name, user name, and password of the Informatica domain. The default values for domain name is domain and the default value for user name is Administrator.
3. Enter the working directory of the service.
4. Enter the image name and HTTP port of the node on which the monitoring Model Repository Service needs to run. The default HTTP port is 8075.
5. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.
6. Specify the type, user ID, and user password of the monitoring Model repository database.

7. Enter the parameters for the database.
   a. If you select Microsoft SQL Server or PostgreSQL, you can choose to 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, 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>
      ```

      PostgreSQL

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

      Microsoft SQL Server

      ```
      jdbc:Informatica:sqlserver://<host name>:<port
      number>;SelectMethod=cursor;DatabaseName=<database name>
      ```
8. 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;
   ```
9. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
10. Click **Create service**.

The service is created. You can view the status of the service along with the tasks performed by the installer in the log summary.

Configure the Data Integration Service

In the **Data Integration Service information** section of the **Create service** page, you can configure the service parameters for the Data Integration Service.

1. Enter the name of the Data Integration Service. Default value is dis.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```
   - You cannot change the name of the service after it is created.
2. Specify the domain name, working directory, and worker node count for the application service. The default value for domain name is domain.

3. Enter the image name and HTTP port of the node on which the Data Integration Service needs to run. The default HTTP port is 8075.
4. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.
5. Enter the name of the Model Repository Service associated with the Data Integration Service. Also, enter the user name and password of the user account that has permissions to manage the Model Repository Service.
6. Optionally, choose to enable 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.
7. If you choose to enable Kubernetes autoscaling, enter the worker node, CPU utilization, and memory utilization information to autoscale:
   - **Minimum number of nodes**. The minimum number of worker nodes to create in the domain. Default is 2.
   - **Maximum number of nodes**. The maximum number of worker nodes to create in the domain. Default is 4.
   - **CPU load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the CPU percentage metrics. Default is 75 percent.
   - **Memory load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the memory percentage metrics. Default is 80 percent.
8. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
9. Click **Create service**.

The service is created. You can view the status of the service along with the tasks performed by the installer in the log summary.

Configure the PowerCenter Repository Service

In the **PowerCenter Repository Service information** section of the **Create service** page, you can provide the connectivity information for the PowerCenter Repository Service and the database information for the PowerCenter repository.

1. Enter the name of the PowerCenter Repository Service and its working directory and domain. The default name of the service is pcrs.

   Note:
   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```
   - You cannot change the name of the service after it is created.
2. Enter the image name and HTTP port of the node on which the PowerCenter Repository Service needs to run.
3. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.

4. Select the code page for the service. If you choose to specify a custom code page, enter the name of the page.
5. Select the database to use for the PowerCenter repository, and then enter the database user account, password, and the JDBC connection string to connect to the database.

   You can use the following syntax in the JDBC connection string:

   Oracle

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

   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>
   ```

6. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
7. Click **Create service**.

The service is created. You can view the status of the service along with the tasks performed by the installer in the log summary.

Configure the PowerCenter Integration Service

In the **PowerCenter Integration Service information** section of the **Create service** page, you can configure the service parameters for the PowerCenter Integration Service.

1. Enter the name of the PowerCenter Integration Service. Default value is pcis.

   Note:

   - The name of the service must be unique within the domain. It must not exceed 128 characters or begin the special character `@`. It must also not contain spaces or the following special characters:

     ```
     ` ~ % ^ * + = { } \ ; : ' " / ? . , < > | ! ( ) ] [
     ```

   - You cannot change the name of the service after it is created.
2. Enter the image name and HTTP port of the node on which the PowerCenter Integration Service needs to run.
3. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.
4. Specify the working directory, code page ID, worker node count, and domain name for the application service.

5. Enter the name of the PowerCenter Repository Service associated with the PowerCenter Integration Service. Also, enter the user name and password of the user account that has permissions to manage the PowerCenter Repository Service.
6. Optionally, choose to enable 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 PowerCenter Integration Service process enabled.
7. If you choose to enable Kubernetes autoscaling, enter the worker node, CPU utilization, and memory utilization information to autoscale:
   - **Minimum number of nodes**. The minimum number of worker nodes to create in the domain. Default is 2.
   - **Maximum number of nodes**. The maximum number of worker nodes to create in the domain. Default is 4.
   - **CPU load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the CPU percentage metrics. Default is 75 percent.
   - **Memory load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the memory percentage metrics. Default is 80 percent.
8. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
9. Click **Create service**.

The service is created. You can view the status of the service along with the tasks performed by the installer in the log summary.

Configure the Email Service

In the **Email Service information** section of the **Create service** page, you can configure the service parameters for the Email Service.

1. Enter the name of the Informatica domain.
2. Enter the image name and HTTP port of the node on which the Email Service needs to run.
3. Optionally, set additional environment variables, expose additional ports, and add additional host alias names for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.
4. Enter the name of the Model Repository Service associated with the Email Service. Also, enter the user name and password of the user account that has permissions to manage the Model Repository Service.
5. Enter the working directory of the service.
6. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.
7. Click **Create service**.

The service is created. You can view the status of the service along with the tasks performed by the installer in the log summary.

Update a Service

You can update the node image, worker node count, and pod resources of an application service in a domain. You can also set additional environment variables and expose additional ports for the service.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.

2. On the **Service** panel, click **Update service**.

   The **Update service** page appears.

3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to update the service on a remote host. If you do not want to set the secure authentication, go to step 5.

4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.

5. In the **Update Service** section, enter the names of the domain and the service to update.

6. Enter the working directory of the service.

7. On the Service drop-down, select the application service to update.

8. Specify the number of worker nodes and the name of Docker image deployed on the nodes.

   Note: You can update the worker node count only for the Data Integration Service and PowerCenter Integration Service.

9. Optionally, set additional environment variables and expose additional ports for the service. If you choose to update the PowerCenter Repository Service and if the service is running on a Microsoft SQL Server database, you must set the variable ODBCINI to `/home/Informatica/K8sVolume/pcrs/odbc.ini` even if you do not set any other additional environment variable for the service.

   Note: If you choose to set additional environment variables, expose additional ports, or add additional host alias names for the service, ensure that you also include the existing values in the list. This prevents the existing domain configurations from getting affected.

10. Optonally, if you choose to update the Data Integration Service or PowerCenter Integration Service, you can enable 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.

11. If you choose to enable Kubernetes autoscaling, enter the worker node, CPU utilization, and memory utilization information to autoscale:

    - **Minimum number of nodes**. The minimum number of worker nodes to create in the domain. Default is 2.
    - **Maximum number of nodes**. The maximum number of worker nodes to create in the domain. Default is 4.
    - **CPU load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the CPU percentage metrics. Default is 75 percent.
    - **Memory load factor**. Kubernetes autoscales the Informatica worker nodes after monitoring the Kubernetes worker node and when the Kubernetes worker node reaches the value set for the memory percentage metrics. Default is 80 percent.

12. Optionally, choose to configure the pod resources, and then update the request and maximum limit values of CPU and memory resources allocated for the service.

13. Click **Update service**.

The service is updated. You can view the status of the service along with the tasks performed by the installer in the log summary.

Remove a Service

You can remove an application service node from an existing domain. When you remove a service, you stop the application service and abort all the associated processes and computations running on the node.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Service** panel, click **Remove service**.

   The **Remove service** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to remove the service from a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Remove Service** section, enter the names of the domain and the service to remove.
6. Enter the working directory of the service.
7. Choose whether to abort the service and its processes running on the nodes or to allow them to complete their operation before the service is stopped an removed from the domain.
8. Click **Remove Service**.

The application service is removed from the domain. You can view whether the service has been removed in the log summary.

Manage Nodes

Informatica Deployment Manager enables you to manage the nodes in existing Data Engineering Integration domains. You can run commands on the nodes and update the node startup commands. You can also shut down and restart the nodes in a domain.

Run a Command on a Node

You can run commands on a gateway or worker node in a domain. After you run a command, you can verify the output of the command.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Node** panel, click **Run command on a node**.

   The **Run command** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to run the command on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
  By default, the SSH connection uses port 22.
- For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.

5. In the **Domain details** section, enter the domain name and the name of the node on which you want to run the command.
6. Click **Run command**.

   The **Run command** window appears.
7. Enter the command and click **Run**.

   The command runs and its output appears in the **Run command** window.

Update the Node Startup Commands

You can update the commands that are automatically executed at the startup of the nodes in an existing domain. You can update the pre-startup or post-startup commands for the nodes. The updated commands run when the pod is restarted. After the pod is successfully restarted, the pre-startup commands run before the node startup and the post-startup commands run after the node startup.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Node** panel, click **Update node startup commands**.

   The **Update node startup commands** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to update the startup commands on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Node Startup Commands** section, enter the name and working directory of the domain for which you want to update the node startup commands.
6. Specify the updated commands.
   a. Click **Add commands**.

      The **Node Startup Commands** window appears.
   b. Click **New**.

      The **New Command** window appears.
   c. Enter the command and the name of the node.

      The command must contain its full executable path. You can enter multiple commands and nodes names as comma-separated values. You can also enter a regular expression followed by an asterisk (*) to imply all occurrences of the regular expression.
   d. On the **Command info type** drop-down, select whether it is a pre-startup or post-startup command.
   e. Choose whether the command must be executed only once before or after the startup.

f. Click **OK**.

The **New Command** window appears, displaying the command and its details.

g. Repeat step b through f for each command to update.

h. Click **OK**.

7. Click **Update**.

The startup command is updated for the node.

Shut Down Nodes

To run administrative tasks, you can shut down a set of nodes in an existing domain. When you shut down a node, you stop Informatica services and abort all application service processes and computations running on the node.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

The **Manage domain** window appears.

2. On the **Node** panel, click **Shut down nodes**.

The **Shut down nodes** page appears.

3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to shut down the nodes on a remote host. If you do not want to set the secure authentication, go to step 5.

4. 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.
  By default, the SSH connection uses port 22.
- For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.

5. In the **Domain details** section, enter the name and working directory of the domain.

6. Specify the nodes to shut down.

a. Click **Add nodes**.

The **Nodes** window appears.

b. Click **New** and enter the name of the node to shut down.

c. Click **OK**.

The name of the specified node appears on the **Nodes** window.

d. Repeat steps b and c for each node that you want to shut down.

e. Click **OK**.

7. Choose whether to stop the application services, Informatica services, and service processes running on the nodes or to allow them to complete their operation before the nodes are shut down.

8. Click **Shut down nodes**.

The application services, Informatica services, and service processes are stopped on each node and the nodes are shut down. You can view whether the nodes are shut down in the log summary.

Restart Nodes

You can restart a set of a nodes in an existing domain using Informatica Deployment Manager. When you restart a node, the application services, Informatica services, and service processes running on the node are restarted.

1. On the Informatica Deployment Manager home page, select **Manage Domain**.

   The **Manage domain** window appears.
2. On the **Node** panel, click **Restart nodes**.

   The **Restart nodes** page appears.
3. Optionally, you can choose to configure Secure Shell (SSH) protocol for secure authentication to restart the nodes on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **Domain details** section, enter the name and working directory of the domain.
6. Specify the nodes to shut down.
   a. Click **Node list**.

      The **Node list** window appears.
   b. Click **New** and enter the name of the node to restart.
   c. Click **OK**.

      The name of the specified node appears on the **Nodes** window.
   d. Repeat steps b and c for each node that you want to restart.
   e. Click **OK**.
7. Choose whether to abort the application services, Informatica services, and service processes running on the nodes or to allow them to complete their operation before the nodes are restarted.
8. Click **Restart nodes**.

   The nodes and the application services, Informatica services, and service processes running on the nodes are restarted. You can view whether the nodes are restarted in the log summary.

Manage Emergency Bug Fixes (EBFs)

Informatica releases Emergency Bug Fixes (EBFs) to provide fixes for critical issues that were found in previous releases. You can use the web-based interface in Informatica Deployment Manager to apply the EBFs on a Data Engineering Integration domain running on Kubernetes. You can also view the list of all EBFs that apply to the domain.

View the List of EBFs

You can use Informatica Deployment Manager to view the list of EBFs that apply to a Data Engineering Integration domain running on Kubernetes. The list includes the EBFs that are already applied on the domain.

1. On the Informatica Deployment Manager home page, select **Manage Emergency Bug Fix (EBF)**.

   The **EBF** page appears.
2. Optionally, choose to configure Secure Shell (SSH) protocol for secure authentication to deploy the EBF on a remote host. If you do not want to set the secure authentication, go to step 4.

3. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
4. In the **EBF details** section, choose to get the list of EBF.
5. Enter the name and working directory of the domain.
6. Click **Get EBF List**.

   The list of EBFs applied on the domain appears.

Apply an EBF

You can use Informatica Deployment Manager to apply an EBF on a Data Engineering Integration domain running on Kubernetes. When you apply an EBF, the pods are restarted and binaries of all previously applied EBFs are replaced with binaries of the EBF that you apply. Therefore, while applying an EBF, you must also apply all the EBFs that are already applied on the domain. This action avoids overriding the previously applied EBFs.

1. In the `EBFs` folder of the shared volume, create a folder for the EBF that you want to apply and extract the .tar file of the EBF in the folder. Ensure that the name of the folder matches the name of the EBF.
2. On the Informatica Deployment Manager home page, select **Manage Emergency Bug Fix (EBF)**.

   The **EBF** page appears.
3. Optionally, choose to configure Secure Shell (SSH) protocol for secure authentication to deploy the EBF on a remote host. If you do not want to set the secure authentication, go to step 5.
4. 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.
     By default, the SSH connection uses port 22.
   - For key type authentication, enter the machine IP address or host name, user name, path of the host authentication key, and port number.
5. In the **EBF details** section, choose to apply EBF.
6. Enter the domain name and the IDs of the EBFs to apply. The IDs must include the ID of the EBF that you want to apply and the IDs of all the EBFs that were previously applied on the domain. Ensure that you enter the IDs as comma-separated values.

   To obtain the IDs of the previously applied EBFs, view the list of EBFson page 35 applied on the domain.
7. Click **Apply EBF**.

   You can view the status of the EBF in the log summary.

Author

Manish Goswami