# Unica Docker V12.0
# Implementation Guide for JBOSS

# Contents

# Chapter 1. Docker, Kubernetes, and Helm

In this release, we have implemented Unica using Docker, Kubernetes, and Helm.

The following topics provide an overview of the mentioned technologies:

- For an overview on Docker, see Docker overview *(on page 1)*.
- For an overview on Kubernetes, see Kubernetes overview *(on page 2)*.
- For an overview on Helm, see Helm overview *(on page 2)*.
- For an overview on Helm Chart, see Helm charts overview *(on page 3)*.

## Docker overview

Docker is an open source software that makes it easy to create, deploy, and manage virtualized applications using containers.

You can use containers to package applications with the necessary components to run the applications, like libraries and other dependencies. Because a container has all the components requires for its execution, it is not dependent on other containers and can run in an isolated manner.

The software that hosts the containers is called a Docker Engine. The Docker Engine creates containers on top of an operating system and automates application deployment on the container.

Docker-based architecture also offers standardization. With standardization of service infrastructure, every team member can work on a production parity environment.

For more information on Docker, its prerequisites, and the minimum system requirements, see https://docs.docker.com/.

# Kubernetes overview

Kubernetes is an open source container orchestration system that provides a platform to automate deployment, scaling, and management of application containers across clusters of hosts.

If you have a cluster containing groups of hosts running Linux containers, you can use Kubernetes to manage the clusters efficiently. These clusters can span hosts across public, private, or hybrid clouds. This makes Kubernetes an ideal platform for hosting cloud-native applications that require rapid scaling.

For more information on Kubernetes, its prerequisites, and the minimum system requirements, see https://kubernetes.io/docs/home/.

# Helm overview

Helm is a package manager for Kubernetes that can be used by developers and operators to easily package, configure, and deploy applications and services onto Kubernetes clusters

Helm can:

- install and upgrade software
- automatically install software dependencies
- configure software deployments
- fetch software packages from repositories

Helm packages are called charts and they contain a few YAML files and templates that are rendered into the Kubernetes manifest files.

For more information on Helm, its prerequisites, and the minimum system requirements, see https://helm.sh/.

## Helm charts overview

Helm Charts are packaging formats. A chart is a collection of files that describe a related set of Kubernetes resources.

You can use a single chart to deploy something simple, like a memcached pod, or something complex, like a full web application stack with HTTP servers, databases, caches, etc.

Charts are created as files laid out in a specific directory tree, and you can package charts into versioned archives for deployment.

**Note:** We will roll out the Helm Charts after you specify the offering related details and requirements. Please contact Unica support team to get a Helm chart.

# Chapter 2. Unica Docker support matrix

Before installation or upgrade, please ensure that the systems meet the recommended requirements.

- To check the recommended hardware requirements, see Minimum hardware requirements *(on page 4)*.
- To check the recommended software requirements, see Software requirements *(on page 4)*.
- To download the Unica Docker image, see Download Unica Docker image *(on page 6)*.

## Minimum hardware requirements

The minimum hardware requirements for Unica Docker are as follows:

| | |
|---|---|
| **Memory** | 64 GB RAM |
| **Storage** | 150 GB Hard disk drive |
| **CPU** | 8 CPU machines |

## Software requirements

The recommended software requirements for Unica Docker are as follows:

**Table 1. Recommended Software Requirements**

| Software type | Recommended requirement |
|---|---|
| Operating System | HCL Unica Docker supports: |

| Software type | Recommended requirement |
|---|---|
| | • CentOS version 7<br>• RHEL version 7.3 |
| Database | HCL Unica Docker supports:<br>• Oracle<br>• DB2 |
| Application Server | HCL Unica Docker supports:<br>• JBOSS EAP 7.1.0<br>• Apache Tomcat |
| Other Software | • Apache ActiveMQ 5.15.8<br>• Docker Enterprise version 19.xx.x<br>• Latest Kubernetes version<br>• Helm version 2 |

**Note:**

- Deployments that use managed Kubernetes clusters are not certified with Unica 12.0.
- Managed Kubernetes clusters require modifications in the chart. Unica will provide sample charts for use and reference.

# Download Unica Docker image

Download the required image from Flex Net Operations (FNO). By default, the chart uses Unica Docker image.

**Note:**

- Although Unica Docker images are CentOS-based or RHEL-based, it may work on any Unix-based operating systems. HCL Software will provide support only for CentOS and RHEL operating system. HCL will not provide support for any other operating systems.
- Although Oracle drivers are bundled with Unica Docker images, HCL Software will not provide support for any issues related to third-party software or third-party drivers bundled with Unica Docker images.
- We will roll out the Helm Charts after you specify the offering related details and requirements. Please contact Unica support team to get a Helm chart.

# Chapter 3. Pre-installation configurations

Before installing or upgrading to Unica Docker, you should complete some configurations.

The list of pre-installation or pre-upgrade configurations are as follows:

- Configure the resources for containers. For more information, see .Minimum required resources for containers *(on page 7)*
- Ensure that you have installed Docker Enterprise version 19.xx.x. For more information, see Docker documentation.
- Ensure that you have installed Kubernetes. For more information, see Kubernetes documentation.
- Verify if:
  - you have configured a Kubernetes cluster.
  - the Kubernetes environment has the appropriate image enforcement policy to allow access to the required repositories.
  - the database is setup. For more information, see Database setup *(on page 9)*.
  - the application server is setup. For more information, see Application server setup *(on page 9)*
  - Place the `jboss-eap-7.1.0.zip` file in the shared filesystem.
- Ensure that you have installed Helm. For more information, see Helm documentation.
- Ensure that JBOSS is configured for Docker. For more information, see Configuring JBOSS for Docker *(on page 10)*.

## Minimum required resources for containers

The following table lists the minimum resources required by each container.

**Table 2. Resources Required by Each Container**

| Resource Name | Resource Value |
|---|---|
| CPU | 2000m (2000 millicpu per container). |

| Resource Name | Resource Value |
|---|---|
| Memory | 2048 MB. |
| Storage | Requires a persistent volume.<br>If the containers fail and restart, the installation and the database data persist. However, setups and JDBC drivers are not packaged with the solution and they need to be placed at the mount point. |

**Note:**

- The configuration will impact the minimum hardware requirements. It is recommended that you use a configuration that is slightly higher than the minimum requirements. If a container does not get the required resources, it might hang.
- **Docker Image Security Vulnerabilities**: For Operating System package vulnerabilities, like NSS, SQLite, etc., run the yum updates within the containers.

# Avoiding timeout issues

To avoid timeout issues, perform the following steps.

1. Access the path `/home/unica/helm/<chart-name>/omnix-unica/`.
2. Open the file `values.yaml`.
3. Add the following lines of code in the annotations section within ingress.

```
nginx.ingress.kubernetes.io/proxy-connect-timeout: "30"
nginx.ingress.kubernetes.io/proxy-read-timeout: "1800"
nginx.ingress.kubernetes.io/proxy-send-timeout: "1800"
nginx.ingress.kubernetes.io/proxy-body-size: 50m
ingress.kubernetes.io/proxy-body-size: 50m
```

4. Save the changes.

# Application server setup

Unica Docker supports Apache Tomcat and JBOSS application servers.

To setup the JBOSS application server, place the JBOSS EAP `ZIP` file on the mount location and configure the path in the Helm chart.

# Database setup

You need to set up the database before you begin installation.

You can setup the database in one of the following ways.

- Use your database Docker image
- Connect to an external database system

If your database resides in an external system, the configuration of the following parameters, in Unica Helm chart, is mandatory.

- Database Users
- Tablespace Users
- Operating System Users

The database can reside within Kubernetes cluster. If the database resides within the Kubernetes cluster, use any available database image, and edit the Unica Helm chart. Ensure that user creation is complete before the Unica solution starts.

For example, to use JBoss, within the cluster or external DB, complete the following steps.

1. Download Unica Docker images and Helm Chart.
2. Add the installable `JBoss` and `JDBC Drivers` to the mount location.
3. Create `Databases` and `Users` and enter those details in the Unica Helm Chart.

If you set the Database as a sub-chart in Unica chart, you can completely automate data import using Shell scripts. For import, data should be available on the Database container mount point. You can also place the data after the container starts. Ensure that Database configuration and user creation activity is completed before running the Unica chart.

# Configuring JBOSS for Docker

To use JBOSS with Unica Docker, complete the following steps.

1. Open the file `common-configMap.yaml`. To locate the file, access the `JBOSSOracle/unica/templates/` location.
2. For the **_JBOSS_ZIP_LOCATION** parameter, provide the folder name, residing within the `HOME` folder, containing the JBOSS installation `ZIP` file. For example, `/docker/unica/JBossZip/JBOSS.Zip`.
3. For the **_JBOSS_ZIP_NAME_** parameter, provide the name of the JBOSS installer `ZIP` file. For example, `jboss-eap-7.1.0.zip`.
4. For the **_DEST_JBOSS_UNZIP_LOCATION_** parameter, provide the absolute directory location where you want to install JBOSS. For example, if you want to install JBOSS inside the container, provide the value `/opt`. If you want to install JBOSS in a mapped shared folder, provide the value `/docker/unica`.
5. For the **_DEST_UNZIP_FOLDER_** parameter, provide a folder name that contains the unzipped contents of the JBOSS installer `ZIP` file. For example, if the `ZIP` file is `jboss-eap-7.1.0.zip` and the folder within the `ZIP` file is `jBoss710`, provide the value `jBoss710`.

Completing the earlier mentioned configuration will automatically install JBOSS and the required Unica component.

# Chapter 4. Helm chart configuration

Before you start the installation or upgrade of Unica Docker, you should configure a few configMap YAML files.

**Note:** We will roll out the Helm Charts after you specify the offering related details and requirements. Please contact Unica support team to get a Helm chart.

The list of configMap YAML files, which you should configure, are as follows:

- `common-configMap.yaml`. For more information, see Common configurations *(on page 11)*.
- `assetpicker-configMap.yaml`. For more information, see Asset Picker configurations *(on page 15)*.
- `birt-configMap.yaml`. For more information, see Birt Reports configurations *(on page 16)*.
- `campaign-configMap.yaml`. For more information, see Campaign configurations *(on page 18)*.
- `interact-configMap.yaml`. For more information, see Interact configurations *(on page 25)*.
- `offer-configMap.yaml`. For more information, see Offer configurations *(on page 39)*.
- `plan-configMap.yaml`. For more information, see Plan configurations *(on page 40)*.
- `platform-configMap.yaml`. For more information, see Platform configurations *(on page 44)*.

## Common configurations

To configure the common configurations, make the necessary modifications to the `common-configMap.yaml` file.

To access the `common-configMap.yaml` file, navigate to `/unica/templates/` in the Unica charts folder. Open the file and make modifications to the following parameters:

**Table 3. Data Parameters**

| Parameter name | Parameter description |
|---|---|
| **WAIT_TIME** | Idle wait time in minutes. |
| **VERSION** | Version number of Unica. |
| **HOME_DIR** | Home directory of docker. |
| **JAVA_HOME** | The location of Java Development Kit on the system. |
| **CERTIFICATE_IMPORT_DIR** | The location of the Unica certificates. |
| **TYPE** | Specify if it is a new installation or an upgrade. Valid values are `INSTALL` or `UPGRADE`. |
| **APPLICATION_DOMAIN** | The application domain. |
| **HOST** | Host ID of the Docker host. |
| **HOST_NAME** | Host name of the Docker host. |
| **DEFAULT_LOCALE** | The default locale to be used. |
| **DOCKER_JAVA_HOME** | The path of the Docker Java Home. |
| **DIRECTOR_JAVA_HOME** | The path of JDK1.8. |
| **JRE_HOME** | The path of the Docker Java Runtime Environment. |
| **MODE** | Specify the products that you will install on the Docker environment. The abbreviated values for each product are as follows:<br>• Platform – PLT<br>• Campaign – CMP |

| Parameter name | Parameter description |
|---|---|
| | • Optimize – OPT<br>• Director – DIR<br>• Plan – PLN<br>• Interact – INT<br>• Offer - OFFER<br>• BIRT - BIRT<br><br>If you want to install all products you should provide the value as follows:<br>`PLT_CMP_INT_PLN_OPT_DIR`<br>If your database is MariaDB, Director will not work on MariaDB. In this case, you must provide the following value:<br>`PLT_CMP_INT_PLN_OPT` |
| **SERVER_TYPE** | The application server installed. |
| **IS_UNICODE** | Set `TRUE` if Unica is installed to support Unicode. Set `FALSE` if Unica is installed without support for Unicode |
| **PROTOCOL** | The protocol used. For example, `HTTP` or `HTTPS`. |

**Table 4. Miscellaneous Parameters**

| Parameter name | Parameter description |
|---|---|
| **DB_HOST_NAME** | The host name of the database system. |
| **DB_PORT** | The port number of the database system. |
| **DB_PLAN_HOST** | The host details of the database in the Plan system. |

| Parameter name | Parameter description |
|---|---|
| **DB_PLAN_PORT** | The database port number of the Plan system. |
| **DB_PLAN_HOST_NAME** | The database host name of the Plan system. |
| **DB_DRIVER** | The database driver file name. |
| **DB_ROOT_USER** | The database root username. |
| **DB_ROOT_PASSWORD** | The database root password. |
| **WLS_DB_USER_NAME** | WebLogic database username. |
| **WLS_DB_PASSWORD** | WebLogic database password. |
| **DB_TYPE** | The name of the database used in the system. For example, `Oracle`. |
| **DB_TYPE_UTILS** | The name of the database utilities used in the system. For example, `Oracle`. |
| **REPLACE_CONNECTION_URL_PREFIX** | The prefix used when forming a URL to the database. Each database has a different prefix. For example, the Oracle database prefix is `jdbc:oracle:thin`. |
| **DIALECT** | The Hibernate dialect. Each database has a different dialect. For example, the Oracle database dialect is `org.hibernate.dialect.Oracle10gDialect`. |
| **DB_DRIVER_CLASS** | The class name of the database drivers. |
| **REPLACE_CONNECTION_URL_PREFIX** | The prefix used when forming a URL to the database. Each database has a different prefix. For example, the Oracle database prefix is `jdbc:oracle:thin`. |

| Parameter name | Parameter description |
|---|---|
| **JDBC_DRIVER_JAR_LOCATION** | The location of the JDBC driver `JAR` file. |
| **DB_DRIVER_JAR** | The location of the database driver `JAR` file. |
| **MYSQL_ROOT_PASSWORD** | The root password for MYSQL. |
| **ORACLE_OWNER** | Oracle owner details. |
| **ORACLE_SID** | Oracle SID details. |
| **REPLACE_JDBC_DRIVER_JAR** | Name of the JDBC driver jar file. This name is also used in replacements in `modules/jdbcmodule/main/module.xml` (name of the JDBC jar). |

# Asset Picker configurations

To configure Asset Picker for Docker, make the necessary modifications to the `assetpicker-configMap.yaml` file.

To access the `assetpicker-configMap.yaml` file, navigate to `/unica/templates/` in the Unica charts folder. Open the file and make modifications to the following parameters:

**Table 5. Asset Picker parameters for JBOSS**

| Parameter name | Parameter description |
|---|---|
| **ASSET_HOST_NAME** | The system host name of Asset Picker. |
| **ASSET_MANAGEMENT_PORT** | The management port number for the Asset Picker system. |
| **ASSET_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the Asset Picker system. |

| Parameter name | Parameter description |
|---|---|
| **ASSET_AJP_PORT** | The AJP port number for the Asset Picker system. |
| **ASSET_HTTP_PORT** | The HTTP port number for the Asset Picker system. |
| **ASSET_HTTPS_PORT** | The HTTPS port number for the Asset Picker system. |
| **ASSET_RECOVERY_ENV_PORT** | The recovery environment port number of the Asset Picker system. |
| **ASSET_STATUS_MANAGER_PORT** | The status manager port number of the Asset Picker system. |
| **PRODUCT_OPTS_ASSET** | Product specific options for Asset Picker. |

# Birt Reports configurations

To configure BIRT reports for Docker, make the necessary modifications to the `birt-configMap.yaml` file.

To access the `birt-configMap.yaml` file, navigate to `/unica/templates/` in the Unica charts folder. Open the file and make modifications to the following parameters:

**Table 6. Common BIRT parameters**

| Parameter name | Parameter description |
|---|---|
| **BIRT_PRODUCT_NAME** | The name assigned for BIRT reports. |

| Parameter name | Parameter description |
|---|---|
| **BIRT_WAR_NAME** | The name of the WAR file. |
| **BIRT_APPLICATION_NAME** | The name of the main application. For example, `Unica`. |
| **BIRT_DOMAIN_USERNAME** | The domain username for BIRT reports. |
| **BIRT_DOMAIN_PASSWORD** | The domain password for BIRT reports. |
| **PRODUCT_OPTS_BIRT** | Product specific options for BIRT reports. |

**Table 7. BIRT parameters for application server**

| Parameter name | Parameter description |
|---|---|
| **BIRT_HOST_NAME** | The system host name of BIRT. |
| **BIRT_MANAGEMENT_PORT** | The management port number for the BIRT system. |
| **BIRT_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the BIRT system. |
| **BIRT_AJP_PORT** | The AJP port number for the BIRT system. |
| **BIRT_HTTP_PORT** | The HTTP port number for the BIRT system. |
| **BIRT_HTTPS_PORT** | The HTTPS port number for the BIRT system. |
| **BIRT_RECOVERY_ENV_PORT** | The recovery environment port number of the BIRT system. |

| Parameter name | Parameter description |
|---|---|
| **BIRT_STATUS_MANAGER_PORT** | The status manager port number of the BIRT system. |
| **BIRT_MIN_HEAP** | The minimum heap size allocated for BIRT. |
| **BIRT_MAX_HEAP** | The maximum heap size allocated for BIRT. |

# Campaign configurations

To configure Campaign for Docker, make the necessary modifications to the `campaign-configMap.yaml` file.

To access the `campaign-configMap.yaml` file, navigate to `/unica/templates/` in the `JBOSSOracle` charts folder. Open the file and make modifications to the following parameters:

**Table 8. Common Campaign parameters**

| Parameter name | Parameter description |
|---|---|
| **CAMPAIGN_JNDI_NAME** | JNDI name for Campaign. |
| **CAMPAIGN_POOL_NAME** | Pool name for Campaign. |
| **PRODUCT_OPTS_CAMPAIGN** | Product specific options for Campaign. |
| **TERM** | The database host name. |

| Parameter name | Parameter description |
|---|---|
| **USER_DATABASES** | Helps in setting up user database. Plug in installations scripts for a seamless startup of an instance. For example, a scaled listener instance. |
| **USER_ORA_HOST_NAME** | The host name of the Oracle user. |

**Table 9. Database-related parameters for Campaign**

| Parameter name | Parameter description |
|---|---|
| **CAMPAIGN_DATABASE_HOST** | Host system details of the system hosting the Campaign database. |
| **CAMPAIGN_DATABASE_PORT** | Port number of the Campaign database. |
| **CAMPAIGN_DATABASE_NAME** | Username to access the Campaign database. |
| **CAMPAIGN_DATABASE_USERNAME** | Password to access the Campaign database. |
| **CAMPAIGN_DATABASE_PASSWORD** | Name of the Campaign database. |
| **CAMPAIGN_DS_INITIAL_SIZE** | The initial size of the Campaign datasource connection pool. |
| **CAMPAIGN_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in |

| Parameter name | Parameter description |
| --- | --- |
| | the Campaign datasource connection pool. |
| **CAMPAIGN_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Campaign datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **CAMPAIGN_DS_MAX_TOTAL** | The maximum number of connections that the Campaign datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **CAMPAIGN_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Campaign datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |
| **ORACLE_CLIENT_SETUP_FILE** | Path of the `tar/gz` file of client. |
| **ORACLE_CLIENT_RESPONSE_FILE** | Path of response file to install client. |

| Parameter name | Parameter description |
|---|---|
| **ORACLE_CLIENT_INSTALL_COMMAND** | Command to install the DB2 client on the listener pod. |
| **ORACLE_CLIENT_INSTALL_SCRIPT** | Path of the test scripts to install client on listener pod. You can write the set of command in this file to install the client and it is executed on the listener pod. |
| **ORACLE_HOME** | Path of the oracle home. |
| **NLS_LANG** | American_America.UTF8 |
| **PATH** | Define the PATH variable |
| **SQLPATH** | Define the SQLPATH variable |
| **TNS_ADMIN** | Path of the Oracle admin folder. |
| **LD_LIB_PATH** | Path to the required shared libraries in the environment configuration script, `setenv.sh`, for Campaign.. |
| **SETENV_COMMAND1** | Setting the variables for `setenv.sh` in the listener you can provide the command. |
| **SETENV_COMMAND2** | Setting the variables for`setenv.sh` in the listener you can provide the command. |

**Table 10. Application Server-related parameters for Campaign**

| Parameter name | Parameter description |
|---|---|
| **CAMPAIGN_URL** | The URL to access Campaign |
| **CAMP_HOST_NAME** | The system host name of Campaign. |
| **CAMP_MANAGEMENT_PORT** | The management port number for the Campaign system. |
| **CAMP_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the Campaign system. |
| **CAMP_AJP_PORT** | The AJP port number for the Campaign system. |
| **CAMP_HTTP_PORT** | The HTTP port number for the Campaign system. |
| **CAMP_HTTPS_PORT** | The HTTPS port number for the Campaign system. |
| **CAMP_RECOVERY_ENV_PORT** | The recovery environment port number of the Campaign system. |
| **CAMP_STATUS_MANAGER_PORT** | The status manager port number of the Campaign system. |

**Table 11. Listener-related parameters for Campaign**

| Parameter name | Parameter description |
|---|---|
| **LISTENER_HOST_NAME** | The hostname of the Listener. |
| **LISTENER_PORT** | The port number of the Listener. |

| Parameter name | Parameter description |
|---|---|
| **LISTENER_TYPE** | Specify the type of Listener. |
| **CLUSTER_DOMAIN** | Define the cluster domain. For example, `listener.default.svc.cluster.local`. |
| **SSL_FOR_PORT2** | SSL server port 2. |
| **SERVER_PORT2** | Server port 2. |
| **MASTER_LISTENER_PRIORITY** | Define the Listener priority. |
| **LOAD_BALANCE_WEIGHT** | The load balance weight of the Listener. |
| **CAMP_HOSTNAME** | The host name of the Campaign system. |
| **CAMPPORT** | The deployment port for Campaign. |
| **CLUSTER_DEPLOYMENT** | Set TRUE if clustered deployment is supported or FALSE if clustered deployment is not supported. |

# Director configurations

To configure Director for Docker, make the necessary modifications to the `director-configMap.yaml` file.

To access the `director-configMap.yaml` file, navigate to `/unica/templates/` in the Unica charts folder. Open the file and make modifications to the following parameters:

**Table 12. Common parameters of Director**

| Parameter name | Parameter description |
|---|---|
| **activemq_enableEvents** | Valid values are `Yes` or `No`. |
| **activemq_url** | Active MQ URL. For example, `tcp://unica-omnix-unica-activemq:61616`. |

| Parameter name | Parameter description |
|---|---|
| **Data_Source_For_ActiveMQ_message_broker_credentials** | Data source for ACTIVEMQ. For example, `ACTIVEMQ_CRED_DS`. |
| **data_sources_for_activemq** | Platform username. |
| **activemq_queueName** | Flowchart information. For example, `campaign`. |

**Table 13. Configuration parameters of Director**

| Parameter name | Parameter description |
|---|---|
| **director_http_port** | Director server port. The default port is `9128`. |
| **director_file_down** | The download path used to store the downloaded log files from the Campaign server. For example, `/docker/unica/Director/Server/Downloads`. |
| **director_show_sql** | Valid values are `TRUE` or `FALSE`. |
| **director_accesstoken_validityseconds** | Director application session timed out token. For example, `10800` seconds. |
| **director_listener_profile_data_days** | Campaign listener CPU and Memory consumption data retention to `7` Days. |

**Table 14. Database-related parameters of Director**

| Parameter name | Parameter description |
|---|---|
| **director_db_name** | Director DB name. |
| **director_datasource_username** | Director database name or username. |
| **director_datasource_password** | Director database password. |
| **director_db_host_ip** | Director database machine host IP address. |

| Parameter name | Parameter description |
|---|---|
| **director_host_name** | Director database machine host name. |
| **director_db_port** | Director database machine port number. |
| **director_datasource_driverClassName** | Database driver class name. |
| **director_jpa_hibernate** | Database driver dialect name. |
| **director_ddl_auto** | Director database mode like create, update, or validate. |
| **director_db_url** | Director database URL. |

# Interact configurations

To configure Interact for Docker, make the necessary modifications to the `interact-configMap.yaml` file.

To access the `interact-configMap.yaml` file, navigate to `/unica/templates/` in the `JBOSSOracle` charts folder. Open the file and make modifications to the following parameters:

**Table 15. Common parameters for Interact**

| Parameter name | Parameter description |
|---|---|
| **CONTEXT_ROOTS** | To enable multiple server groups in Interact. Ensure that the context root and deployment name are in sync. If you change the deployment name, remember to change the context root as well. For example, if server groups are named atm, callcenter, and web, define the deployment and services with similar names like interactatm, |

| Parameter name | Parameter description |
|---|---|
| | interactcallcenter, and interactweb and ensure that the **CONTEXT_ROOT** parameter contains the following values: `INTERACTATM;INTERACTCALLCENTER;INTERACTWEB`. |
| **INTERACT_PROD_JNDI_NAME** | JNDI name for Interact production. |
| **INTERACT_PROD_POOL_NAME** | Pool name for Interact production. |
| **INTERACT_TEST_JNDI_NAME** | JNDI name for Interact test. |
| **INTERACT_TEST_POOL_NAME** | Pool name for Interact test. |
| **INTERACT_LEARNING_JNDI_NAME** | JNDI name for Interact learning. |
| **INTERACT_LEARNING_POOL_NAME** | Pool name for Interact learning. |
| **INTERACT_CHRH_JNDI_NAME** | JNDI name for Interact CHRH. |
| **INTERACT_CHRH_POOL_NAME** | Pool name for Interact CHRH. |
| **INTERACT05_JNDI_NAME** | JNDI name for Interact05. |
| **INTERACT05_POOL_NAME** | Pool name for Interact05. |
| **INTERACTATM_JNDI_NAME** | JNDI name for Interact ATM. |
| **INTERACTATM_POOL_NAME** | Pool name for Interact ATM. |
| **INTERACTCALLCNTR_JNDI_NAME** | JNDI name for Interact Call Center. |
| **INTERACTCALLCNTR_POOL_NAME** | Pool name for Interact Call Center. |
| **INTERACTWEB_JNDI_NAME** | JNDI name for Interact Web. |
| **INTERACTWEB_POOL_NAME** | Pool name for Interact Web. |
| **PRODUCT_OPTS_INTERACT** | Product specific options for Interact. |
| **TERM** | The database host name. |

**Table 16. Platform-related Interact server parameters**

| Parameter name | Parameter description |
|---|---|
| **INTERACTATM_PLATFORM_DATABASE_HOST** | Host system details of the system hosting the Platform-Interact ATM database. |
| **INTERACTATM_PLATFORM_DATABASE_PORT** | Port number of the Platform-Interact ATM database. |
| **INTERACTATM_PLATFORM_DATABASE_USERNAME** | Username to access the Platform-Interact ATM database. |
| **INTERACTATM_PLATFORM_DATABASE_PASSWORD** | Password to access the Platform-Interact ATM database. |
| **INTERACTATM_PLATFORM_DATABASE_NAME** | Name of the Interact Platform-Interact database. |
| **INTERACTATM_PLATFORM_DS_INITIAL_SIZE** | The initial size of the Platform-Interact ATM datasource connection pool. |
| **INTERACTATM_PLATFORM_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Platform-Interact ATM datasource connection pool. |
| **INTERACTATM_PLATFORM_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Platform-Interact ATM datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACTATM_PLATFORM_DS_MAX_TOTAL** | The maximum number of connections that the Platform-Interact ATM datasource can hold. If the number of connection requests |

| Parameter name | Parameter description |
|---|---|
| | exceed the configured value, the connection will be refused. |
| **INTERACTATM_PLATFORM_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Platform-Interact ATM datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |

**Table 17. Server group-related database parameters of Interact**

| Parameter name | Parameter description |
|---|---|
| **INTERACTATM_DATABASE_HOST** | Host system details of the system hosting the Interact ATM database. |
| **INTERACTATM_DATABASE_PORT** | Port number of the Interact ATM database. |
| **INTERACTATM_DATABASE_USERNAME** | Username to access the Interact ATM database. |
| **INTERACTATM_DATABASE_PASSWORD** | Password to access the Interact ATM database. |
| **INTERACTATM_DATABASE_NAME** | Name of the Interact ATM database. |
| **INTERACTATM_DS_INITIAL_SIZE** | The initial size of the Interact ATM datasource connection pool. |
| **INTERACTATM_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact ATM datasource connection pool. |
| **INTERACTATM_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact ATM datasource connection pool. Any idle connections, which exceeds the |

| Parameter name | Parameter description |
|---|---|
| | configured value, will be removed from the pool. |
| **INTERACTATM_DS_MAX_TOTAL** | The maximum number of connections that the Interact ATM datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACTATM_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact ATM datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |
| **INTERACTATM_DATABASE_HOST** | Host system details of the system hosting the Interact ATM database. |
| **INTERACTATM_DATABASE_PORT** | Port number of the Interact ATM database. |
| **INTERACTATM_DATABASE_USERNAME** | Username to access the Interact ATM database. |
| **INTERACTATM_DATABASE_PASSWORD** | Password to access the Interact ATM database. |
| **INTERACTATM_DATABASE_NAME** | Name of the Interact ATM database. |
| **INTERACTATM_DS_INITIAL_SIZE** | The initial size of the Interact ATM datasource connection pool. |
| **INTERACTATM_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact ATM datasource connection pool. |
| **INTERACTATM_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the |

| Parameter name | Parameter description |
|---|---|
| | Interact ATM datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACTATM_DS_MAX_TOTAL** | The maximum number of connections that the Interact ATM datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACTATM_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact ATM datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |
| **INTERACTWEB_DATABASE_HOST** | Host system details of the system hosting the Interact Web database. |
| **INTERACTWEB_DATABASE_PORT** | Port number of the Interact Web database. |
| **INTERACTWEB_DATABASE_USERNAME** | Username to access the Interact Web database. |
| **INTERACTWEB_DATABASE_PASSWORD** | Password to access the Interact Web database. |
| **INTERACTWEB_DATABASE_NAME** | Name of the Interact Web database. |
| **INTERACTWEB_DS_INITIAL_SIZE** | The initial size of the Interact Web datasource connection pool. |
| **INTERACTWEB_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact Web datasource connection pool. |

| Parameter name | Parameter description |
|---|---|
| **INTERACTWEB_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact Web datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACTWEB_DS_MAX_TOTAL** | The maximum number of connections that the Interact Web datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACTWEB_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact Web datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |
| **INTERACTCALLCNTR_DATABASE_HOST** | Host system details of the system hosting the Interact Call Center database. |
| **INTERACTCALLCNTR_DATABASE_PORT** | Port number of the Interact Call Center database. |
| **INTERACTCALLCNTR_DATABASE_USERNAME** | Username to access the Interact Call Center database. |
| **INTERACTCALLCNTR_DATABASE_PASSWORD** | Password to access the Interact Call Center database. |
| **INTERACTCALLCNTR_DATABASE_NAME** | Name of the Interact Call Center database. |
| **INTERACTCALLCNTR_DS_INITIAL_SIZE** | The initial size of the Interact Call Center datasource connection pool. |

| Parameter name | Parameter description |
|---|---|
| **INTERACTCALLCNTR_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact Call Center datasource connection pool. |
| **INTERACTCALLCNTR_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact Call Center datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACTCALLCNTR_DS_MAX_TOTAL** | The maximum number of connections that the Interact Call Center datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACTCALLCNTR_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact Call Center datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |

**Table 18. Server-related database parameters of Interact**

| Parameter name | Parameter description |
|---|---|
| **INTERACT_PROD_DATABASE_HOST** | Host system details of the system hosting the Interact Production database. |
| **INTERACT_PROD_DATABASE_PORT** | Port number of the Interact Production database. |

| Parameter name | Parameter description |
|---|---|
| **INTERACT_PROD_DATABASE_NAME** | Username to access the Interact Production database. |
| **INTERACT_PROD_DATABASE_USERNAME** | Password to access the Interact Production database. |
| **INTERACT_PROD_DATABASE_PASSWORD** | Name of the Interact Production database. |
| **INTERACT_PROD_DS_INITIAL_SIZE** | The initial size of the Interact Production datasource connection pool. |
| **INTERACT_PROD_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact Production datasource connection pool. |
| **INTERACT_PROD_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact Production datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACT_PROD_DS_MAX_TOTAL** | The maximum number of connections that the Interact Production datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACT_PROD_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact Production datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |

| Parameter name | Parameter description |
|---|---|
| **INTERACT_TEST_DATABASE_HOST** | Host system details of the system hosting the Interact Test database. |
| **INTERACT_TEST_DATABASE_PORT** | Port number of the Interact Test database. |
| **INTERACT_TEST_DATABASE_NAME** | Username to access the Interact Test database. |
| **INTERACT_TEST_DATABASE_USERNAME** | Password to access the Interact Test database. |
| **INTERACT_TEST_DATABASE_PASSWORD** | Name of the Interact Test database. |
| **INTERACT_TEST_DS_INITIAL_SIZE** | The initial size of the Interact Test datasource connection pool. |
| **INTERACT_TEST_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact Test datasource connection pool. |
| **INTERACT_TEST_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact Test datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACT_TEST_DS_MAX_TOTAL** | The maximum number of connections that the Interact Test datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACT_TEST_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact Test datasource. Statement caching improves performance |

| Parameter name | Parameter description |
|---|---|
| | by caching executable statements that are used repeatedly. |
| **INTERACT_LEARNING_DATABASE_HOST** | Host system details of the system hosting the Interact Learning database. |
| **INTERACT_LEARNING_DATABASE_PORT** | Port number of the Interact Learning database. |
| **INTERACT_LEARNING_DATABASE_NAME** | Username to access the Interact Learning database. |
| **INTERACT_LEARNING_DATABASE_USERNAME** | Password to access the Interact Learning database. |
| **INTERACT_LEARNING_DATABASE_PASSWORD** | Name of the Interact Learning database. |
| **INTERACT_LEARNING_DS_INITIAL_SIZE** | The initial size of the Interact Learning datasource connection pool. |
| **INTERACT_LEARNING_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact Learning datasource connection pool. |
| **INTERACT_LEARNING_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact Learning datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **INTERACT_LEARNING_DS_MAX_TOTAL** | The maximum number of connections that the Interact Learning datasource can hold. If the number of connection requests exceed |

| Parameter name | Parameter description |
|---|---|
| | the configured value, the connection will be refused. |
| **INTERACT_LEARNING_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact Learning datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |
| **INTERACT_CHRH_DATABASE_HOST** | Host system details of the system hosting the Interact CHRH database. |
| **INTERACT_CHRH_DATABASE_PORT** | Port number of the Interact CHRH database. |
| **INTERACT_CHRH_DATABASE_NAME** | Username to access the Interact CHRH database. |
| **INTERACT_CHRH_DATABASE_USERNAME** | Password to access the Interact CHRH database. |
| **INTERACT_CHRH_DATABASE_PASSWORD** | Name of the Interact CHRH database. |
| **INTERACT_CHRH_DS_INITIAL_SIZE** | The initial size of the Interact CHRH datasource connection pool. |
| **INTERACT_CHRH_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact CHRH datasource connection pool. |
| **INTERACT_CHRH_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact CHRH datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |

| Parameter name | Parameter description |
|---|---|
| **INTERACT_CHRH_DS_MAX_TOTAL** | The maximum number of connections that the Interact CHRH datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACT_CHRH_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact CHRH datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |
| **INTERACT05_DATABASE_HOST** | Host system details of the system hosting the Interact05 database. |
| **INTERACT05_DATABASE_PORT** | Port number of the Interact05 database. |
| **INTERACT05_DATABASE_NAME** | Username to access the Interact05 database. |
| **INTERACT05_DATABASE_USERNAME** | Password to access the Interact05 database. |
| **INTERACT05_DATABASE_PASSWORD** | Name of the Interact05 database. |
| **INTERACT05_DS_INITIAL_SIZE** | The initial size of the Interact 05 datasource connection pool. |
| **INTERACT05_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Interact 05 datasource connection pool. |
| **INTERACT05_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Interact 05 datasource connection pool. Any idle connections, which exceeds the |

| Parameter name | Parameter description |
|---|---|
| | configured value, will be removed from the pool. |
| **INTERACT05_DS_MAX_TOTAL** | The maximum number of connections that the Interact 05 datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **INTERACT05_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Interact 05 datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |

**Table 19. Application server-related parameters of Interact**

| Parameter name | Parameter description |
|---|---|
| **INT_HOST_NAME** | The system host name of Interact. |
| **INT_MANAGEMENT_PORT** | The management port number for the Interact system. |
| **INT_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the Interact system. |
| **INT_AJP_PORT** | The AJP port number for the Interact system. |
| **INT_HTTP_PORT** | The HTTP port number for the Interact system. |
| **INT_HTTPS_PORT** | The HTTPS port number for the Interact system. |

| Parameter name | Parameter description |
|---|---|
| **INT_RECOVERY_ENV_PORT** | The recovery environment port number of the Interact system. |
| **INT_STATUS_MANAGER_PORT** | The status manager port number of the Interact system. |

# Offer configurations

To configure Centralized Offer Management for Docker, make the necessary modifications to the `offer-configMap.yaml` file.

To access the `offer-configMap.yaml` file, navigate to `/unica/templates/` in the `JBOSSOracle` charts folder. Open the file and make modifications to the following parameters:

**Table 20. JBOSS-related parameters of Centralized Offer Management**

| Parameter name | Parameter description |
|---|---|
| **COM_HOST_NAME** | The system host name of Centralized Offer Management. |
| **COM_MANAGEMENT_PORT** | The management port number for the Centralized Offer Management system. |
| **COM_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the Centralized Offer Management system. |

| Parameter name | Parameter description |
|---|---|
| **COM_AJP_PORT** | The AJP port number for the Centralized Offer Management system. |
| **COM_HTTP_PORT** | The HTTP port number for the Centralized Offer Management system. |
| **COM_HTTPS_PORT** | The HTTPS port number for the Centralized Offer Management system. |
| **COM_RECOVERY_ENV_PORT** | The recovery environment port number of the Centralized Offer Management system. |
| **COM_STATUS_MANAGER_PORT** | The status manager port number of the Centralized Offer Management system. |
| **PRODUCT_OPTS_COM** | Product specific options for Centralized Offer Management. |

# Plan configurations

To configure Plan for Docker, make the necessary modifications to the `plan-configMap.yaml` file.

To access the `plan-configMap.yaml` file, navigate to `/unica/templates/` in the `JBOSSOracle` charts folder. Open the file and make modifications to the following parameters:

**Table 21. Common parameters of Plan**

| Parameter name | Parameter description |
|---|---|
| **PLAN_HOME** | The home directory for the Plan system. |
| **PRODUCT_OPTS_PLAN** | Product specific options for Plan. |
| **PLAN_JNDI_NAME** | JNDI name for Plan. |
| **PLAN_POOL_NAME** | Pool name for Plan. |

**Table 22. Application server-related parameters of Plan**

| Parameter name | Parameter description |
|---|---|
| **PLAN_HOST_NAME** | The system host name of Plan. |
| **PLAN_MANAGEMENT_PORT** | The management port number for the Plan system. |
| **PLAN_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the Plan system. |
| **PLAN_AJP_PORT** | The AJP port number for the Plan system. |
| **PLAN_HTTP_PORT** | The HTTP port number for the Plan system. |
| **PLAN_HTTPS_PORT** | The HTTPS port number for the Plan system. |

| Parameter name | Parameter description |
|---|---|
| **PLAN_RECOVERY_ENV_PORT** | The recovery environment port number of the Plan system. |
| **PLAN_STATUS_MANAGER_PORT** | The status manager port number of the Plan system. |
| **PLAN_URL** | The minimum heap size allocated for Plan. |

**Table 23. Database-related parameters for Plan**

| Parameter name | Parameter description |
|---|---|
| **PLAN_PORT** | The port number to access the Plan system. |
| **PLAN_HOST** | The host name of the Plan system. |
| **DB_PLAN_HOST** | The host details of the database in the Plan system. |
| **DB_PLAN_PORT** | The database port number of the Plan system. |
| **DB_PLAN_HOST_NAME** | Host name of the system hosting the Plan database. |
| **PLAN_DATABASE_HOST** | Host system details of the system hosting the Plan database. |
| **PLAN_DATABASE_PORT** | Port number of the Plan database. |
| **PLAN_DATABASE_NAME** | Name of the Plan database. |

| Parameter name | Parameter description |
|---|---|
| **PLAN_DATABASE_USERNAME** | Username to access the Plan database. |
| **PLAN_DATABASE_PASSWORD** | Password to access the Plan database. |
| **PLAN_DS_INITIAL_SIZE** | The initial size of the Plan datasource connection pool. |
| **PLAN_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Plan datasource connection pool. |
| **PLAN_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Plan datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **PLAN_DS_MAX_TOTAL** | The maximum number of connections that the Plan datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused. |
| **PLAN_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Plan |

| Parameter name | Parameter description |
|---|---|
| | datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |

# Platform configurations

To configure Platform for Docker, make the necessary modifications to the `platform-configMap.yaml` file.

To access the `platform-configMap.yaml` file, navigate to `/unica/templates/` in the Unica charts folder. Open the file and make modifications to the following parameters:

**Table 24. Common parameters of Platform**

| Parameter name | Parameter description |
|---|---|
| **PLATFORM_JNDI_NAME** | JNDI name for Platform. |
| **PLATFORM_POOL_NAME** | Pool name for Platform. |
| **PRODUCT_OPTS_BASE** | Base options for all products of Unica. |
| **PRODUCT_OPTS_PLATFORM** | Product specific options for Platform. |
| **UNICA_PLATFORM_HOME** | The home directory for the Platform product. |
| **REPLACE_VALID_CONNECTION_CHECKER** | Common replacements in the file `standalone/configuration/standalone.xml`. |
| **REPLACE_DATASOURCE_CLASS** | Common replacements in the file `standalone/configuration/standalone.xml`. |
| **TERM** | The database host name. |

**Table 25. Database-related parameters of Platform**

| Parameter name | Parameter description |
|---|---|
| **PLATFORM_DATABASE_HOST** | Host system details of the system hosting the Platform database. |
| **PLATFORM_DATABASE_PORT** | Port number of the Platform database. |
| **PLATFORM_DATABASE_USERNAME** | Username to access the Platform database. |
| **PLATFORM_DATABASE_PASSWORD** | Password to access the Platform database. |
| **PLATFORM_DATABASE_NAME** | Name of the Platform database. |
| **PLATFORM_DS_INITIAL_SIZE** | The initial size of the Platform datasource connection pool. |
| **PLATFORM_DS_MIN_IDLE** | The minimum number of idle connections (not connected to a database) in the Platform datasource connection pool. |
| **PLATFORM_DS_MAX_IDLE** | The maximum number of idle connections (not connected to a database) in the Platform datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool. |
| **PLATFORM_DS_MAX_TOTAL** | The maximum number of connections that the Platform datasource can hold. If the number of connection requests exceed the |

| Parameter name | Parameter description |
|---|---|
| | configured value, the connection will be refused. |
| **PLATFORM_DS_STATEMENT_CACHE_SIZE** | Maximum number of statements that can be cached in the Platform datasource. Statement caching improves performance by caching executable statements that are used repeatedly. |

**Table 26. Application server-related parameters of Platform**

| Parameter name | Parameter description |
|---|---|
| **ADMIN_USR_NAME** | The administrator user name. |
| **ADMIN_USR_PASSWORD** | The password of the administrator user. |
| **JBOSS_ZIP_LOCATION** | The location of the JBOSS ZIP file. |
| **JBOSS_ZIP_NAME** | The name of the JBOSS ZIP file. |
| **DEST_JBOSS_UNZIP_LOCATION** | The location to unzip the JBOSS ZIP file. |
| **DEST_UNZIP_FOLDER** | The folder into which the JBOSS ZIP file should be unzipped. |
| **REPLACE_JDBC_DRIVER_JAR** | Name of the JDBC driver jar file. This name is also used in replacements in `modules/jdbcmodule/main/module.xml` (name of the `JDBC jar`). |
| **FORCE_INIT_JBOSS** | Set TRUE to force initialize JBOSS or FALSE to avoid force initialization. |
| **JVM_MIN_HEAP** | The minimum heap size for JVM. |
| **JVM_MAX_HEAP** | The maximum heap size for JVM. |

| Parameter name | Parameter description |
|---|---|
| **JVM_MIN_METASPACE** | The minimum meta space for JVM. |
| **JVM_MAX_METASPACE** | The maximum meta space for JVM. |
| **MANAGER_URL** | The URL to access Manager. |
| **PLAT_HOST_NAME** | The system host name of Platform. |
| **PLAT_MANAGEMENT_PORT** | The management port number for the Platform system. |
| **PLAT_MANAGEMENT_HTTPS_PORT** | The management HTTPS port number for the Platform system. |
| **PLAT_AJP_PORT** | The AJP port number for the Platform system. |
| **PLAT_HTTP_PORT** | The HTTP port number for the Platform system. |
| **PLAT_HTTPS_PORT** | The HTTPS port number for the Platform system. |
| **PLAT_RECOVERY_ENV_PORT** | The recovery environment port number of the Platform system. |
| **PLAT_STATUS_MANAGER_PORT** | The status manager port number of the Platform system. |

# Sub-chart configuration in Helm charts

To run a DB container as a sub-chart, the database must reside within the cluster. Sub-charts have their own configMap for configurations.

**Note:** Unica Docker does not own the database.

# Chapter 5. Installation and verifying the installation

The following topics provide information related to installation and verification of installation.



## Installation

You can install Unica using helm charts. Override the following Helm chart values using `--set name=value`.

**Note:** We will roll out the Helm Charts after you specify the offering related details and requirements. Please contact Unica support team to get a Helm chart.

- Ensure that configMaps in the helm chart are correctly configured. For more information, see Helm chart configuration *(on page 11)*.
- Verify all the configurations and ensure that the mount location does not have any Unica-related installation files.

1. `kubectl apply -f ./omnix-unica/extra-configs/local-pv.yaml`
2. `helm install --name nginx stable/nginx-ingress -f ./omnix-unica/extra-configs/nginx-conf.yaml`
3. `helm install --name unica -f ./omnix-unica/values-local.yaml omnix-unica --set service.hostname=kubernetes.nonprod.hclpnp.com --set service.applicationDomain='nonprod.hclpnp.com' --set ingress.enabled=true`

# Verifying the chart

Follow the instructions after the completion of Helm installation for chart verification. The chart generates an output for all the resources it creates.

1. To confirm if a chart has generated output for all the resources, run the following command:

   ```
   helm ls
   ```

2. To view the installed helm release, run the following command:

   ```
   helm status unica
   ```

3. To view the Unica Kubernetes pods, run the following command:

   ```
   kubectl get pods
   ```

# Log files

Confirm if the required containers are up and running. Upon confirmation, check the logs for all the running services.

- **Installation log files**:

  The installation log files are placed in the logs folder at the mount point. For example, `$HOME_DIR/logs`.
- **Product log files**:

  Log files are persisted out of the containers at the mount location. The log files for the products are placed in their respective install location folders. For example, if the product is Campaign and the mount location is `/docker/unica`, the Campaign log files will be available within the `/docker/unica/Campaign/logs/` location.

## Campaign Log Files

To enable the ETL, Engage, and UBX logs within the Campaign logs folder, provide the absolute path in the `$CAMPAIGN_HOME/conf/campaign_log4j.properties` file.

**Example**

```
log4j.appender.ETL.File=/docker/unica/Campaign/logs/ETL.log
log4j.appender.ENGAGE_ETL.File=/docker/unica/Campaign/logs/EngageETL.log
log4j.appender.UBX.File=/docker/unica/Campaign/logs/UBX.log
```

## Interact Log Files

For Interact, the log files are present in the following path:

**Table 27. Log File Location in Application Servers**

| Stack Name | Location |
|---|---|
| JbossOracle | `/opt/jboss-eap-7.1/jboss-eap-7.1/bin/` |

# Chapter 6. Post installation configurations

The following topics contain details about post installation configurations related to the products of Unica.



## Configurations for Campaign

To add user database in Campaign, complete the following steps:

1. Connect to the Listener pod.
2. Add the user database.
3. In the application, navigate to **Campaign > Configuration**.
4. Add an entry for Datasources.

**Note:** Unica Docker does not support multipartitioning on Campaign.

## Configurations for Director

To configure Director post installation, perform the following step:

Update the **_DIR_HOME_** in the `Campaign/bin/setenv.sh` location with the actual path.

# Configurations for Interact

For Gateway configurations to work, perform the following step.

1. Add the required JAR files and the configuration files to the mount location.
2. On JMX console, use the CentOS desktop and the VNC viewer to view the individual pod consoles. Enable port forwarding on different ports.

# Configurations for Platform

For Director and Campaign History tab, you should configure the Platform settings.

To configure Platform settings, complete the following steps:

1. Log in to Marketing Software.
2. Select **Settings > Configuration**.
3. On the left pane, select **IBM Marketing Platform > Security > API management > IBM Marketing Platform**.
4. On the left pane, select **Authentication** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/authentication/login` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Enabled |
| **Require authentication for API access** | Disabled |

5. On the left pane, select **User** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/usr/partitions/*` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Enabled |

6. On the left pane, select **Policy** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/policy/partitions/*` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Enabled |

7. On the left pane, select **Configurations** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/datasource/config` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Enabled |

8. On the left pane, select **Datasource** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/datasource` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Enabled |

9. On the left pane, select **Login** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/authentication/v1/login` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Disabled |

10. On the left pane, select **Unica Campaign > Campaign REST API Filter** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/rest/v1/*` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Enabled |

11. On the left pane, select **Unica Campaign > Campaign REST API V2 Filter** and in the right pane click **Edit settings**. The value for the fields should be:

| Field name | Value |
|---|---|
| **API URI** | `/rest/v2/*` |
| **Block API access** | Disabled |
| **Secure API access over HTTPS** | Disabled |
| **Require authentication for API access** | Enabled |

# Chapter 7. Upgrading on-premise applications to Docker

You can upgrade an on-premise version of Unica to the Docker version. The Docker will be deployed on the application server.

The prerequisites for the upgrade are as follows:

- Take a backup of your existing database.
- Copy the file system of the previous version to the mount location.
- Provide appropriate values the database parameters of all Unica components. For more information, see Helm chart configuration *(on page 11)*.
- For Interact fast upgrade, the schema name in the target setup should be the same as the one in the base setup.
- Manually map the tables and restart the Campaign Pod.

In the `common-configMap.yaml` file, provide values for the following fields.

**Table 28. Configurable Parameters to perform an Upgrade**

| Parameter Name | Example Value |
|---|---|
| **BASE_FOLDER** | `"OLDINSTALL/HCLUnica_86"` |
| **FROM** | `"8.6.0"` |
| **TO** | `"12.0.0"` |
| **SOURCE_SCHEMA** | `"CAMP86"` |
| **TARGET_SCHEMA** | `"CAMP86"` |
| **AC_VERSION** | `"8.6.x"` |
| **ACI_UNICODE** | `"y"` |
| **CONFIGURE_ON_ERROR_PROMPT** | `"n"` |
| **LOCALE** | `"en_US"` |

| Parameter Name | Example Value |
|---|---|
| **TYPE** | UPGRADE |
| **DATABASE_EXPORT_DIR** | /DBBACKUP/ |
| **ISEXTERNALDB** | false |
| **DB_IMPORT_WAIT_TIME** | 1050 |
| **DB_PRE_IMPORT_WAIT_TIME** | 1050 |
| **IS_UNICODE** | false |
| **UPGRADE_FROM_TO** | 11.1+To12.0 |
| **LISTENER_HOST_NAME** | {{ .Release.Name }}-omnix-unica-listener |
| **SOURCE_SCHEMA_RT** | camp86 |
| **TARGET_SCHEMA_RT** | camp86 |
| **DB_DRIVER_CLASS_RT** | com.ibm.db2.jcc.DB2Driver |
| **SOURCE_SCHEMA_PROD** | intpr86 |
| **TARGET_SCHEMA_PROD** | intpr86 |
| **DB_DRIVER_CLASS_PROD** | com.ibm.db2.jcc.DB2Driver |
| **SOURCE_SCHEMA_LRN** | intlr86 |
| **TARGET_SCHEMA_LRN** | intlr86 |
| **DB_DRIVER_CLASS_LRN** | com.ibm.db2.jcc.DB2Driver |
| **SOURCE_SCHEMA_RUN** | intrt86 |
| **TARGET_SCHEMA_RUN** | intrt86 |
| **DB_DRIVER_CLASS_RUN** | com.ibm.db2.jcc.DB2Driver |

Add the JVM Option `-DFAST_UPGRADE_VERSION=<BASE_VERSION>`. For example: `JAVA_OPTIONS="${JAVA_OPTIONS} -DFAST_UPGRADE_VERSION=8.6.x`.

Supported versions in the fast upgrade are as follows:

| | |
|---|---|
| `8.6.x` | -DFAST_UPGRADE_VERSION=8.6.x |
| `9.0.x` | -DFAST_UPGRADE_VERSION=9.0.x |
| `9.1.x.x (9.1.x, 9.1.1.x and 9.1.2.x)` | -DFAST_UPGRADE_VERSION=9.1.x |
| `10.0.x` | -DFAST_UPGRADE_VERSION=10.0.x |
| `10.1.x` | -DFAST_UPGRADE_VERSION=10.1.x |
| `11.0.x` | -DFAST_UPGRADE_VERSION=11.0.x |
| `11.0.1` | -DFAST_UPGRADE_VERSION=11.0.1.x |

The mount location should contain the old version of the Unica file system. Unica Docker containers will manage the database upgrade and the file system updates.

1. To perform the upgrade, run the following command.
   helm install --name unica omnix-unica --set service.hostname=<kubernetes.nonprod.hclpnp.com --set service.applicationDomain='nonprod.hclpnp.com' --set ingress.enabled=true
2. Access the upgrade logs from the mount location.

# Configuring Marketing Platform post upgrade

For a fast upgrade of Marketing Platform, complete the following steps:

1. Marketing Platform application URL will point to the old base environment. Change the navigation URL using the SQL script from the Platform system database.

2. Manually change the URL of the start page, which appears when you log in to Unica Marketing Software products, from the `USM_PERSONALIZATION` table.

3. Copy the following properties files from the source environment to the destination environment. Ensure that all the URLs mentioned in the files are also updated to the destination environment.
    - `Platform_Admin_URL.properties`
    - `Platform_Admin_View_Priv.properties`
    - `Platform_Admin_URL.properties`
    - `Platform_Admin_Scheduler_Scripts.properties`
    - `Platform_Admin_Scheduler_API.properties`

# Chapter 8. Scaling Unica containers

Scaling a deployment ensures creation and scheduling of new Pods. Scaling increases the number of Pods to the new required state. Kubernetes also supports autoscaling of Pods.

For Multicast, perform the configurations on Kubernetes host to support it. For example, weave supports multicast and can be configured for multicast support.

The following topics provide information on scaling the containers of Unica:

- For details related to scaling Listener containers, see Scaling Listener containers *(on page 60)*.
- For details related to scaling Interact containers, see Scaling Interact containers *(on page 63)*.

## Scaling Listener containers

Listeners are defined as StatefulSets in Kubernetes. Each Pod in a StatefulSet derives its hostname from the name of the StatefulSet and the ordinal of the Pod.

The Pod domain is managed by the service and it takes the following form:

```
$(service name).$(namespace).svc.cluster.local.
```

For example, the listener pod entry is registered as follows:

```
listener-0.listener.default.svc.cluster.local
```

These can be configured in the Helm chart in the `campaign-configMap.yaml` file.

Like a Deployment, a StatefulSet manages the Pods that are based on identical container specifications. Unlike a Deployment, a StatefulSet maintains a sticky identity for each of their Pods.

The location of Campaign shared home is `$HOME_DIR/Campaign`.

For the scaled instances of StatefulSet, `listener-0`, `listener-1`, `listener-2`,..`listener-n`, each instance has a file system mapped on the mount location. For example, `$HOME_DIR/listener/listener-0`.

### Ordered scale up and scale down

1. Ordered and graceful deployment and scaling.
   If you want to scale up the Listener pod, run the following command:

   ```
   kubectl scale StatefulSets listener --replicas=2
   ```

2. First instance gets deleted in the end.
   If you want to scale down the Listener pod, run the following command:

   ```
   kubectl scale StatefulSets listener --replicas=1
   ```

### Listener-Optimize merge

1. Single scalable deployment in Kubernetes.
2. Configuration and license driven `config.xml`.
3. Listener integration *(on page 62)*

### Cluster mode

1. To enable scaling, by default, cluster mode must be `TRUE`.

Also perform the following listener-related scaling activities:

- Load balancing *(on page 62)*
- Listener integration *(on page 62)*

## Load balancing

For load balancing, there is a single listener that executes commands related to Campaign flowchart and Optimize sessions. In comparison to Campaign flowchart, an Optimize session requires a significantly better hardware configuration, which exceeds the minimum recommendation, for successful execution.

This newly introduced single listener helps the master listener to decide the node on which it should send the execution of the flowcharts or sessions, considering the `loadBalanceWeight`. We recommend that you avoid executing Optimize sessions on a node, configured to execute campaign flowcharts. Similarly, we recommend that you avoid setting up a node with a significantly higher configuration of hardware for executing flowcharts. Using the new flag, the master listener can utilize the available resources in an appropriate way.

Choose an appropriate `listenerType` during installation based on the hardware, or configuration, or your requirements.

## Listener integration

Prior to Unica 12.0 release, Campaign and Optimize were separate products. Users having both Campaign and Optimize had to run separate listeners. The Campaign listener `unica_aclsnr` to run flowcharts and Optimize listener `unica_acolsnr` to run the Optimize session.

### Campaign-Optimize merged scenario

With text-based license for v12, the listener image expects a license file at mount point.

If both listener host name txt (`listener-0.txt` …) and `opt.instance` file exist, it will create only the Optimize listener. If listener host name txt contains the first listener, it creates the listener as `LISTENER_TYPE 3`, which means it is for both Campaign and Optimize, otherwise it creates the listener as `LISTENER_TYPE 2` indicating that it is only for Optimize.

If the listener host name txt, `listener-0.txt` and so on, exists and the `opt.instance` file does not exist, it creates the listener as `LISTENER_TYPE 3`, which indicates that it is for both Campaign and Optimize.

### Listener types

- **CAMPAIGN_ONLY (TYPE 1)** - This listener can handle commands for Campaign or flowchart only.
- **OPTIMIZE_ONLY (TYPE 2)** - This listener can handle commands for Optimize session only.
- **ALL ((TYPE 3)**- This listener can handle commands for Campaign or Flowchart or Optimize session.

The Type option is available in the following locations:

- **Settings > Configuration > Campaign > unicaACListener**
- **Settings > Configuration > Campaign > unicaACOListener**

## Scaling Interact containers

Each existing Interact machine runs a Kubernetes Interact deployment. If you have set the **hostNetwork** to `TRUE`, the existing network, which already supports multicast, can be used as it is without changing any settings. You can also use the existing load balancers over the Kubernetes Interact deployments.

To scale Interact pods for multiple server groups, refactor the helm chart to add services and deployments per server group. Each Server Group should point to a different Platform Instance. For example, if there are three RT server groups, there will be three Platform instances (three services and three deployments for Platform and Interact).

The **CONTEXT_ROOTS** variable, in the **interact_configMap.yaml** file drives:

- the context roots for Interact and Platform.
- PLT and RT database details per server group.

If you want to scale pods for a server group, run the following command:

```
kubectl scale deployment hcl-unica-interact --replicas=2
```

If the Interact POD crashes, or if you manually delete the pod, manually delete an entry from the configuration using the following command:

```
./configTool.sh -d -p 'Affinium|Campaign|partitions|partition1|Interact
|serverGroups|interactatm|instanceURLs|$1' -o "
```

In the earlier command $1 refers to the Interact POD name that crashed or was manually deleted.

## Monitoring the scaled instances

**Note:** Ensure that VNC viewer exists on the host machine to monitor instances.

You can perform JMX monitoring for each of the scaled instances using port forwarding.

For POD1, run the following command:

```
kubectl port-forward --address 0.0.0.0 pod/unica-omnix-unica-
interact-84d7b47f59-d2rsl 9998:9998 &
```

For POD2, run the following command:

```
kubectl port-forward --address 0.0.0.0 pod/unica-omnix-unica-
interact-84d7b47f59-d2rsl 9999:9998 &
```

# Chapter 9. Deployment monitoring

The Kubernetes Dashboard is a web-based user interface to monitor deployments.

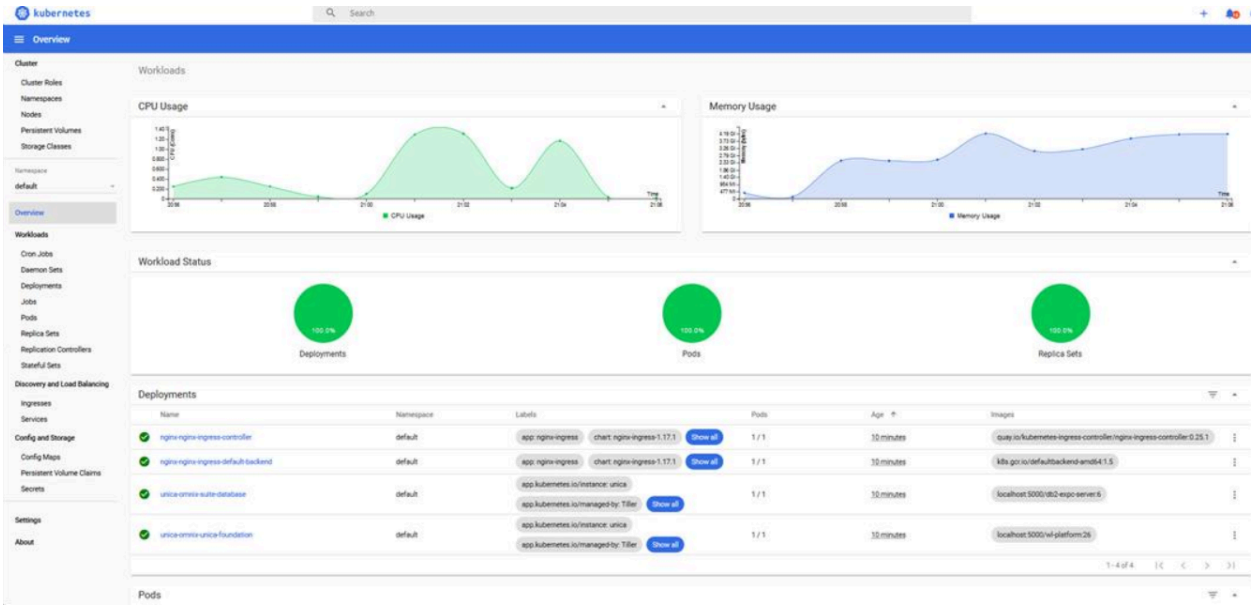
Use the Kubernetes Dashboard to:

- deploy containerized applications to a Kubernetes cluster
- troubleshoot your containerized applications
- managing cluster resources

You can also the use Dashboard to get an overview of the applications running on your cluster, as well as for creating or modifying individual Kubernetes resources.

The Dashboard also provides information on the state of Kubernetes resources in your cluster and on any errors that may have occurred.

Figure 1. Kubernetes dashboard




# Deploying the dashboard user interface

The Dashboard user interface is not deployed by default.

To deploy the Dashboard user interface, run the following command.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/
v2.0.0-beta4/aio/deploy/recommended.yaml
```

# Chapter 10. Product utilities

You can execute all the utilities of the Unica products in their assigned pods.

The following table lists the Unica products and their assigned pods for running the product-specific utilities.

**Table 29. Unica products and their assigned pods for running the utilities**

| Unica Product Name | Pod Name | List of utilities |
|---|---|---|
| Unica Campaign | `Listener` | For the list of Campaign utilities, see Unica Campaign *(on page* |
| Unica Platform | `Platform` | For the list of Platform utilities, see Unica Platform *(on page 68* |
| Unica Plan | `Plan` | For the list of Plan utilities, see Unica Plan *(on page 68)*. |

## Unica Campaign

You can access all the utilities of Campaign from the location `CAMPAIGN_HOME/bin`. Access the location and run the required utility. For more information on utilities, see Unica Campaign Administrator's Guide.

Following is the list of utilities available in Campaign.

- Campaign advanced search utility (`advSrchUtil`)
- Campaign advanced search agent (`advSrchAgent`)
- Campaign listener shutdown utility (`svrstop`)
- Campaign Server Manager (`unica_svradm`)
- Campaign session utility (`unica_acsesutil`)
- Campaign cleanup utility
- Campaign report generation utility (`unica_acgenrpt`)

# Unica Platform

You can access all the utilities of Platform from the location `Platform/tools/bin`. Access the location and run the required utility. For more information on utilities, see Unica Platform Administrator's Guide.

Following is the list of utilities available in Platform.

- `alertConfigTool`
- `configTool`
- `datafilteringScriptTool`
- `encryptPasswords`
- `encryptTomcatDBPasswords`
- `partitionTool`
- `populated`
- `restoreAccess`

# Unica Plan

You can access all the utilities of Plan from the location `MarketingOperations/tools/bin`. Access the location and run the required utility. For more information on the utilities, see Unica Plan Installation Guide.

Following is the list of utilities available in Plan.

- `umodbsetup`
- `configTool`

# Chapter 11. FAQs and troubleshooting

This section covers the frequently asked questions and troubleshooting issues.

To view the list of FAQs, see Frequently Asked Questions (on page 69)

For information related to Troubleshooting, see Troubleshooting Issues (on page 71)

## Frequently Asked Questions

This topic contains the list of FAQs related to Unica Docker release.

The list of FAQs are as follows:



### Question 1

How do I configure Campaign Docker image to support non-ASCII data?

To configure non-ASCII data support for the Campaign Docker image, execute the same steps used for configuring non-ASCII data support on On-premise Campaign. For more details, see the topic **Non-ASCII data in Campaign** in the Unica Campaign Administrator's Guide.

### Question 2

How to install products on locations other than default location mentioned in the `common-configMap.yaml` file?

To install products on location other than the default location configured in the `common-configMap.yaml` file, complete the following steps.

1. Mount the directory.
2. Open the `common-configMap.yaml` file and update the default path to the required path.
3. Ensure that the JDBCDrivers folder and the `JBOSS.zip` file exists in the provided path.

## Question 3

Why has Unica Docker installed JRE9 and JDK8 on my system?

Unica Docker is bundled with JRE9 and JRE8. In the `common-configMap.yaml` file:

- Provide the path of JRE9 for the parameter **DOCKER_JAVA_HOME**. Unica Docker uses JRE9 for installation tasks.
- Provide the path of JDK8 for the parameter **JAVA_HOME**. The products of Unica use JDK8.

## Question 4

Should the passwords in the jdbc.properties file be encrypted?

Yes. The passwords in the jdbc.properties file should be encrypted. Configure the passwords using the helm commands similar to configuring the host name. You do not have to store the passwords anywhere for reuse. Once you configure the passwords, it will be set in the application.

For Unica Docker, the `jdbc.properties` file is available in the following locations:

- `/Interact/PatternStateETL/bin/jdbc.properties`
- `/Interact/tools/bin/jdbc.properties`
- `/ContactOptimization/install/jdbc.properties`
- `/Platform/tools/bin/jdbc.properties`
- `/install/jdbc.properties`

- `/Campaign/bin/jdbc.properties`
- `/Campaign/eMessage/conf/jdbc.properties`
- `/Campaign/install/jdbc.properties`

### Question 5

List the default JDBC drivers provided with the Listener container.

On the Listener container, the JDBC drivers exist in the following path: `Docker_Home/JdbcDrivers/`. The list of default JDBC drivers available with the Listener container are as follows:

- `db2jcc4.jar`
- `mariadb-java-client-2.4.1.jar`
- `ojdbc8_docker.jar`

### Question 6

What should I do to make `/ACOOptAdmin.sh` work?

For `/ACOOptAdmin.sh` to work, update the following parameters in the `/ACOOptAdmin.sh` file:

- `JAVA_HOME`
- `OPTIMIZE_HOME`
- `JDBCDRIVER_CLASSPATH`

## Troubleshooting Issues

This topic contains the list of Troubleshooting issues related to Unica Docker release.

The list of FAQs are as follows:

## Question 1

Stopping and Restarting an Application Server

Sometimes, you might have to stop and restart the application server. For example, if you have modified some settings and these modified settings require restarting the application server.

Before stopping and restarting JBOSS, complete the following steps.

a. Save your work and confirm that all users have logged off.
b. Locate the running docker container using the command kubectl get pods.
c. Access the container using the command kubectl exec -it <name of the container> bash.
d. Locate the running process using the command ps -ef.
e. Kill the process using the command kill -9. This stops the JBOSS server.
f. **Note:** Always start the server in the background. If you do not start the server in the background, you cannot access the command prompt till the server starts. If the server takes too long to start, press CTRL+C to terminate the JBOSS server.

   To restart the server, access the bin directory of JBOSS and start the server by running the command standalone.sh in the background.
g. To exit the docker container, press CTRL+D.

## Question 2

Cannot select supported locales for Marketing Operations.

When installing Marketing Operations using Docker, you cannot select specific supported locales from the available list of supported locales. The system will automatically accept all available locales as the supported locales.

## Question 3

ActiveMQ URL does not work.

The `ActiveMQ` URL `http://unica-omnix-unica-activemq:8161/admin/queues.jsp`, which provides information about the flowchartInfo-campaign events count, will not work. This is a Known Issue and will be fixed in the next release.

# Chapter 12. Uninstalling the chart

1. To uninstall or delete the my-release deployment, run the following command.

   ```
   helm delete --purge <releasename>
   ```

2. Delete the persistent volumes.

3. Delete the file systems.

If required, clean the persisted data of the database.