

Cloud Native Unica V12.1.4 Implementation Guide for Oracle WebLogic Server



Contents

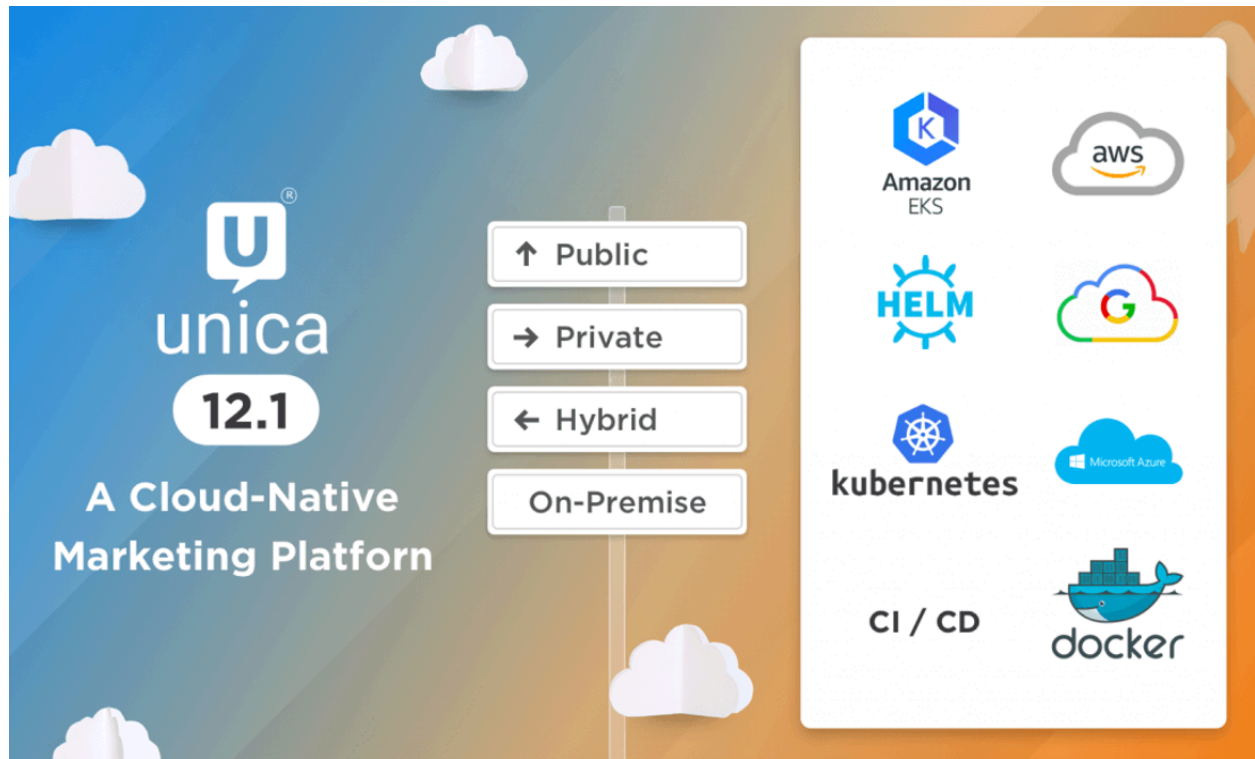
Chapter 1. Cloud Native Unica Overview.....	1
Features and Benefits.....	1
Chapter 2. Docker, Kubernetes, and Helm.....	3
Docker overview.....	3
Kubernetes overview.....	3
Helm overview.....	4
Helm charts overview.....	4
Chapter 3. ELK Overview.....	5
Chapter 4. Pre-installation configurations.....	7
Avoiding timeout issues.....	7
Application server setup.....	8
Database setup.....	8
Listener Database client setup.....	11
Configuring WebLogic for Cloud Native Unica.....	11
Setting up Cloud Native Unica environment.....	12
Cloud Native Unica setup on SSL.....	15
Chapter 5. Cloud Native Unica Helm release architecture.....	16
Chapter 6. Helm chart configuration.....	17
Chapter 7. Installation and verifying the installation.....	19
Installation.....	19
Verifying the chart.....	19
Log files.....	20
Campaign Log Files.....	20

Chapter 8. Post installation configurations.....	21
Configurations for Campaign.....	21
Configuring multi-partitions for Campaign.....	21
Configurations for Director.....	23
Configurations for Interact.....	23
Configurations for Platform.....	24
Chapter 9. Migration of on-premises applications to Cloud Native Unica.....	27
Migration prerequisites.....	27
common-configMap configurations.....	30
JVM option configurations.....	31
Performing the migration.....	31
Configuring Unica Campaign post migration.....	32
Configuring Unica Interact post migration.....	32
Configuring Unica Platform post migration.....	32
Chapter 10. Cloud Native Unica upgrade.....	34
Custom listener scripts and Cloud Native Unica container OS upgrade.....	34
Chapter 11. Scaling Unica containers.....	36
Scaling Listener containers.....	36
Load balancing.....	38
Listener integration.....	38
Scaling Interact containers.....	39
Scaling Journey engine containers.....	40
Chapter 12. Using Red Hat OpenShift.....	42
Security Context Constraints for Unica on Red Hat OpenShift.....	43
Chapter 13. Deployment monitoring.....	46

Deploying the dashboard user interface.....	46
Chapter 14. Product utilities.....	47
Unica Campaign.....	47
Unica Plan.....	48
Unica Platform.....	48
Chapter 15. Using secret to avoid passwords in plain text.....	49
Chapter 16. Using AWS Secrets and Configuration Provider with Kubernetes Secret Store CSI Driver.....	51
AWS Secret Manager Implementation.....	52
Prerequisite Software for AWS Secret Manager.....	52
Prerequisite Configurations for AWS Secret Manager.....	53
Implementing AWS Secret Manager.....	54
Chapter 17. Enabling Multicast using Weave-Net CNI plugin on AWS EKS cluster.....	59
Chapter 18. FAQs and troubleshooting.....	61
Frequently Asked Questions.....	61
Question 1.....	61
Question 2.....	61
Question 3.....	62
Question 4.....	62
Question 5.....	63
Question 6.....	63
Question 7.....	64
Troubleshooting Issues.....	64
Question 1.....	64
Question 2.....	65

Question 3.....	65
Chapter 19. Uninstalling the chart.....	66
Chapter 20. Appendix: Description of Helm chart parameters.....	67
Common configurations.....	67
Audience Central configurations.....	75
Campaign configurations.....	79
Centralized Offer Management configurations.....	86
Collaborate configurations.....	88
Contact Central configurations.....	91
Content Integration configurations.....	94
Director configurations.....	96
Insights Reports configurations.....	98
Interact configurations.....	99
Journey configurations.....	114
Journey web configurations.....	115
Kafka configurations.....	120
Plan configurations.....	121
Platform configurations.....	125
Segment Central configurations.....	130
Sub-chart configuration in Helm charts.....	137
values.yaml driven configurations.....	137

Chapter 1. Cloud Native Unica Overview



Features and Benefits

The features and benefits of Cloud Native Unica are as follows:

- **Why Unica Cloud-Native?**

We are amidst an information technology revolution driven by the cloud. The emergence of containers, with their fast startup, standardized application packaging, and isolation model, is further contributing to efficiency and agility.

Along with the benefits of Docker like reducing hardware resource costs, ease-of-use, portability, scalability, and modularity of deployments, Kubernetes provides container-orchestration capabilities for automating application deployment (rollout and rollbacks), workload scaling, and high availability. Helm charts leverage Kubernetes packages to streamline the installation and management of applications deployed on Kubernetes.

- **Faster release pace**

Unica Docker Images will be rolled out for each release and/or fix pack. Time to market has become the key differentiator between the most innovative organizations and their lagging competition. With this deployment approach, Unica can build and ship more value to its customers. With this deployment approach, it is very easy for customers to consume new releases.

- **Better CX**

HCL Unica shipd Docker Images with new features faster and keep iterating continuously. An extensive set of APIs in the Unica Products would enable integrations with enterprise data stores. Overall, Cloud-native applications allow you to improve your customer experiences.

- **Ease of Application management**

Cloud-native also has many options to make infrastructure management effortless. With helm charts, application management, monitoring, deployments will be easy, automated, and configuration driven.

- **Reduced cost through containerization**

Containers make it easy to manage and secure applications independently of the infrastructure that supports them. The industry is now consolidating around Kubernetes for the management of these containers at scale. Alongside Kubernetes, there are a host of powerful cloud-native tools and standardization of infrastructure and tooling. This, along with open-source technologies, drives down costs. Pricing flexibility models are all enabled with a cloud-native deployment approach.

- **Building Reliable systems**

With modern cloud-native approaches like Kubernetes in the cloud, you can more easily build applications to be fault-tolerant with resiliency and self-healing built-in. Because of this design, even when failures happen, you can easily isolate the impact of the incident, so it doesn't take down the entire application. Instead of servers and monolithic applications, cloud-native microservices helps you achieve higher uptime and thus further improve the user experience.

- **Deploy Anywhere**

Unica solutions can be deployed on any cloud. Unica can be deployed on Managed Kubernetes Clusters like EKS, GKE, AKS, etc.

Chapter 2. Docker, Kubernetes, and Helm

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

The following topics provide an overview of the mentioned technologies:

- For an overview on Docker, see [Docker overview \(on page 3\)](#).
- For an overview on Kubernetes, see [Kubernetes overview \(on page 3\)](#).
- For an overview on Helm, see [Helm overview \(on page 4\)](#).
- For an overview on Helm Chart, see [Helm charts overview \(on page 4\)](#).

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.

Chapter 3. ELK Overview

The ELK Stack is a collection of three open-source products: Elasticsearch, Logstash, and Kibana.

The features of the ELK stack are as follows:

- Centralized logging and monitoring to identify problems with containers, or applications, hosted inside the pod
- Visualization tools to represent data using a graph or a chart
- Host ELK as a standalone system or a Docker container or a Kubernetes pod

Filebeat

Filebeat collects and ships log files, and is also the most commonly used beat. You can install Filebeat on almost any operating system, or also as a Docker container. It includes internal modules for platforms like Apache, MySQL, Docker, MariaDB, Kafka, and many more.

Filebeat is very efficient and it displays this in how it handles backpressure. If Logstash is busy, Filebeat slows down its read rate and picks up the beat once the slowdown is completed.

For Unica, Filebeat is deployed as a sidecar container inside Platform pod, and Platform and Filebeat, as a sidecar, runs as a container inside a pod. The Filebeat has a read-only access to persistent volume. It reads the logs from the logs folder, specified in configmap for all Unica applications, to keep sending logs to ELK.

metricbeat

Metricbeat collects and reports various system-level metrics for various systems and platforms. Metricbeat also supports internal modules for collecting statistics from specific platforms.

For Unica, metricbeat is deployed as a daemonset in `kube-system` namespace to collect metrics from the metric-server in OpenShift. In Kubernetes, it connects to both the Kubernetes API-server and the metric server.

Fluentd

It is a smart metrics and log shipper. For Unica, Fluentd is deployed as a daemonset in custom namespace to collect metrics from Kubernetes.

Advantages of using ELK

Logging , keeping historical logs, or monitoring the logs is a real challenge in containerized applications. If you destroy a container, everything is lost, including the logs. The advantages of using ELK are as follows:

- Maintains and keeps all the data, and makes this activity easy, even if the cluster, pod, or node is destroyed.
- Allows searching of all the logs in a single place.
- Helps find issues in multiple servers, or pods, by connecting logs during a specific time frame.

Chapter 4. Pre-installation configurations

Before installing or upgrading to Cloud Native Unica, 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 *Cloud Native Unica Support Matrix Guide*.
- 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 \(on page 8\)](#).
 - the application server is setup. For more information, see [Application server setup \(on page 8\)](#)
 - if WebLogic is configured for Cloud Native Unica. For more information, see [Configuring WebLogic for Cloud Native Unica \(on page 11\)](#).
- Ensure that you have installed Helm. For more information, see [Helm documentation](#).

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

Cloud Native Unica supports Apache Tomcat®, Red Hat® JBoss® Enterprise Application Platform (EAP), and Oracle® WebLogic Server application servers.

Install Oracle WebLogic Server on the shared filesystem.



Note:

For WebLogic server, edit the `setDomainEnv` script in the `bin` directory, within the WebLogic domain directory, to add the following Java options to `JAVA_VENDOR`:

```
-Dfile.encoding=UTF-8
```

If you are deploying on a non-production setup, add

```
-DENABLE_NON_PROD_MODE=true
```

If you are deploying on a production setup, the Java option,

```
-DENABLE_NON_PROD_MODE=true
```

, must be removed or set to `false`.

After saving the changes, restart the WebLogic server.

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

In case of Managed Kubernetes Clusters on Cloud, the system data and the customer data must reside on Cloud.

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 WebLogic, within the cluster or external DB, complete the following steps.

1. Download Cloud Native Unica images and Helm Chart.
2. 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.

For auto-installation of database client on listener pod or container, complete the following steps:



Note: The commands and filenames are mentioned specific to Oracle database. Provide appropriate values based on the the database you use.

1. Place the Oracle client installer, named `linuxamd64_12102_client.zip`, inside the `/tmp` folder.
2. To extract the installer file, run the unzip command.

A new folder, named `client` is created in the location `/tmp`.

3. Run the following command:

```
cp /tmp/client/response/client_install.rsp /tmp/oracle_client.rsp
```

4. Access the `oracle_client.rsp` file and make the following changes in the file:

```
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/home/oracle/oraInventory
ORACLE_HOME=/home/oracle/app/oracle/product/12.1.0/client_1
ORACLE_BASE=/home/oracle/app/oracle
oracle.install.client.installType=Administrator
```

5. Run the following commands:

```
cd /tmp
```

```
mkdir linuxamd64_12102_client
```

```
mv client linuxamd64_12102_client
```

```
tar -cvf Oracle_client.tar linuxamd64_12102_client oracle_client.rsp
```

```
gzip Oracle_client.tar
```

```
mv Oracle_client.tar.gz oracle_client.rsp /docker/unica
```

6. In the `/docker/unica/` location, create a file named `oracle.sh` and add the following content in the file:

```
yum install -y libaio
/
tmp/Oracle_client_install/linuxamd64_12102_client/client/runInstaller
-silent -ignoreSysPrereqs
-responseFile /tmp/Oracle_client_install/oracle_client.rsp
```

Listener Database client setup

To establish an ODBC connection to the database, the Campaign listener requires a database client.

If you do not have a database client, you must install it. For a seamless installation of the database client, perform the following steps:

1. Place the database client installer at the mount location (NFS).
2. Configure the location of the database client installer in the `campaign-configMap.yaml` file. For more information, see [Campaign configurations \(on page 79\)](#).

Configuring WebLogic for Cloud Native Unica

Your system should have WebLogic installed to use it with Cloud Native Unica. Cloud Native Unica uses the utilities from WebLogic to create a domain for the required Unica component.

To use WebLogic with Cloud Native Unica, complete the following steps:

1. For the **JAVA_HOME_WEBLOGIC** parameter, WebLogic requires Oracle JDK. The value of this setting is the HOME location of the JDK used for the existing WebLogic installation. For example, `/docker/unica/jdk18_oracle`.
2. For the **WLS_HOME_DIR** parameter, provide the home directory of the WebLogic server installation. For example, `/docker/unica/oracle_products/middleware/wlse`.
3. For the **WLS_DOMAIN_LOCATION** parameter, provide the fully qualified path of the directory under which the domains for the products will be created. For example, `/docker/unica/wlsdomains`.



CAUTION: The creation of a new pod creates a directory for the domain of that Pod. If you discard a pod, the directory is not deleted automatically. You should delete such directories as they consume a lot of disk space.

Setting up Cloud Native Unica environment

You must set up the Cloud Native Unica environment before implementing it. The chart that you download uses Helm as a package manager for Kubernetes. The chart is a preconfigured application resource and it deploys Unica suite on a specified Kubernetes cluster. Extract the chart **ZIP** file to a location in the cloud VM, where you plan to deploy Unica. For reference purposes, this chart contains a placeholder for the database. Unica does not own the database and is not responsible for database management. If required, set a containerized database (the charts and subcharts folders are for reference) as a subchart to the Unica Chart. You can use scripts to automate the restoration of database on a container.

The prerequisites for running a Helm chart are as follows:

- Download the required Docker images from Flex Net Operations (FNO).
- To import the downloaded Docker images for all the products, run the following command:

```
docker load -i product_image_name.tar
```

- To verify if all products images are loaded and available for use, run the following command:

```
docker images
```

- To tag the images appropriately, run the following command:

```
docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]
```

- To push the images to the docker registry, run the following command:

```
docker push TARGET_IMAGE[:TAG]
```

- Open the **values.yaml** file, which is placed inside the **Unica** folder, and edit:
 - the Docker images name in the **repository** section
 - the tag numbers in the **tag** section

See the following code snippet for reference:

```

image:
  repository:
    init: TARGET_IMAGE
    platform: TARGET_IMAGE
  tag:
    init: TAG
    platform: TAG

```

- Configure the database in one of the following ways:
 - **Database within Kubernetes cluster** - Set the database as a subchart to Unica helm chart. Unica will not own or manage the database chart.
 - **Pointing to an external database** - Configure the database to reside on the same subnet as the worker nodes to ensure good performance.

To set up Cloud Native Unica environment, complete the following steps:

1. Update chart configurations:

- a. Update or customize database and application server details in the **configMap** files for each products. For more information on **configMap** files, see [Helm chart configuration \(on page 17\)](#). An example for updating or customizing the **campaign-configMap.yaml** is as follows:

```

CAMPAIGN_DATABASE_HOST:
  "{{{ .Release.Name }}}-unica-suite-database"
CAMPAIGN_DATABASE_PORT: "1521"
CAMPAIGN_DATABASE_NAME: "xe"
CAMPAIGN_DATABASE_USERNAME: "campuser"
CAMPAIGN_DATABASE_PASSWORD: "unica"
CAMPAIGN_DS_INITIAL_SIZE: "1"
CAMPAIGN_DS_MIN_IDLE: "1"
CAMPAIGN_DS_MAX_IDLE: "15"
CAMPAIGN_DS_MAX_TOTAL: "80"

```

```
CAMPAIGN_DS_STATEMENT_CACHE_SIZE: "300"
```

For more information on configurations related to `values.yaml` file, see [values.yaml driven configurations \(on page 137\)](#).

2. Update persistence volume:

- a. Based on the persistent volume of your choice, update the following files:

```
- unica/extra-configs/local-pv.yaml
  - unica/templates/pvc.yaml
```

3. Perform an upgrade:

- a. You can use one of the following methods to upgrade:
 - Upgrade from on-premises to Cloud Native (for example, Unica version 9.1.2 to version 12.1.0)
 - Upgrade from earlier Cloud Native version to new version (for example, Unica version 12.0 to version 12.1)
- b. Before the upgrade, ensure that you have backed up the file system and the Database.
- c. Place the file system on the mount point and configure the **BASE_FOLDER** parameter in the `common-configMap.yaml` file to point to the file system location.
- d. Also, update the database details in the `common-configMap.yaml` file. For example, refer the following code snippet:

```
DATABASE_EXPORT_DIR: "/DBBACKUP/"
BASE_FOLDER: "OLDINSTALL/IBMUnica_86"
SOURCE_SCHEMA: "camp86"
TARGET_SCHEMA: "camp86"
SOURCE_SCHEMA_RT: "camp86"
TARGET_SCHEMA_RT: "camp86"
SOURCE_SCHEMA_PROD: "intpr86"
```

```
TARGET_SCHEMA_PROD: "intpr86"  
SOURCE_SCHEMA_LRN: "intl86"  
TARGET_SCHEMA_LRN: "intl86"  
SOURCE_SCHEMA_RUN: "intrt86"  
TARGET_SCHEMA_RUN: "intrt86"
```

- e. In case of managed Kubernetes clusters, change the value of the **storageClassNames** parameter in the `values.yaml` file.



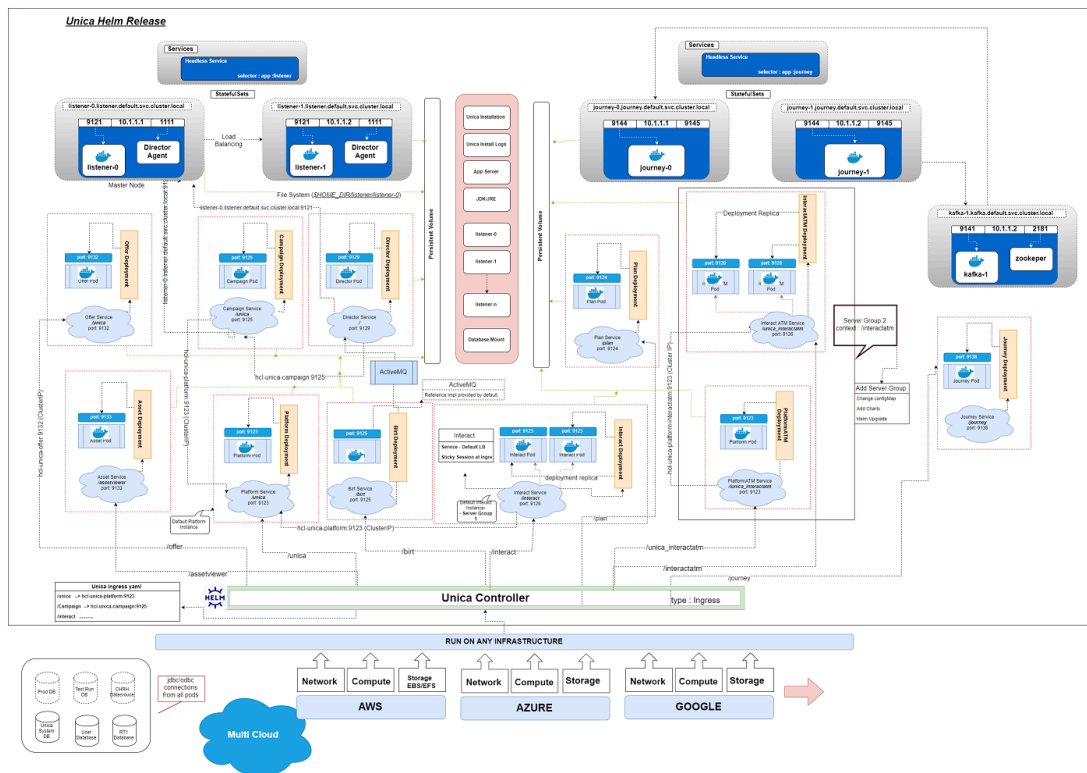
Note: Active MQ Image or Chart, provided by Unica, is for reference purposes only. Unica does not own or is not responsible for Active MQ Deployments.

Cloud Native Unica setup on SSL

You can configure SSL on Cloud Native Unica setup at the ingress level.

A provision exists to create a secret with a CERT file. For additional details, see [nginx-ingress documentation](#) for TLS configuration.

The following is a diagrammatic representation of the Cloud Native Unica Helm release architecture:



[Click here](#), to access HTML help, for a better resolution of the image.

Chapter 6. Helm chart configuration

Before you start the installation or upgrade of Cloud Native Unica, you should configure the appropriate configMap YAML files.

To access the configMap YAML files, navigate to `/unica/templates/` in the Unica charts folder. Open one of the following files and modify the parameters in that file:

- `common-configMap.yaml`. For more information, see [Common configurations \(on page 67\)](#).
- `audiencecentral-configMap.yaml`. For more information, see [Audience Central configurations \(on page 75\)](#).
- `campaign-configMap.yaml`. For more information, see [Campaign configurations \(on page 79\)](#).
- `offer-configMap.yaml`. For more information, see [Centralized Offer Management configurations \(on page 86\)](#).
- `collaborate-configMap.yaml`. For more information, see [Collaborate configurations \(on page 88\)](#).
- `contactcentral-configMap.yaml`. For more information, see [Contact Central configurations \(on page 91\)](#).
- `assetpicker-configMap.yaml`. For more information, see [Content Integration configurations \(on page 94\)](#).
- `director-configMap.yaml`. For more information, see [Director configurations \(on page 96\)](#).
- `birt-configMap.yaml`. For more information, see [Insights Reports configurations \(on page 98\)](#).
- `interact-configMap.yaml`. For more information, see [Interact configurations \(on page 99\)](#).
- `journey-configMap.yaml`. For more information, see [Journey configurations \(on page 114\)](#).
- `journeyweb-configMap.yaml`. For more information, see [Journey web configurations \(on page 115\)](#).

- `kafka-configMap.yaml`. For more information, see [Kafka configurations \(on page 120\)](#).
- `plan-configMap.yaml`. For more information, see [Plan configurations \(on page 121\)](#).
- `platform-configMap.yaml`. For more information, see [Platform configurations \(on page 125\)](#).
- `segmentcentral-configMap.yaml`. For more information, see [Segment Central configurations \(on page 130\)](#).

Chapter 7. Installation and verifying the installation

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

- [Installation \(on page 19\)](#)
- [Verifying the chart \(on page 19\)](#)
- [Log files \(on page 20\)](#)

Installation

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

- Ensure that `configMaps` in the helm chart are correctly configured.
- 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`

After installation, add the installation related parameters in the `commom-configMap.yaml` file for version 12.1.4.

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
```

Chapter 8. Post installation configurations

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

- [Configurations for Campaign \(on page 21\)](#)
- [Configurations for Director \(on page 23\)](#)
- [Configurations for Interact \(on page 23\)](#)
- [Configurations for Platform \(on page 24\)](#)

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.

Configuring multi-partitions for Campaign

For Unica Campaign, you can configure the application within the partitions where you have configured an instance of Campaign.

Application users, within each partition, can access the Campaign functions, data, and customer tables that are configured for Campaign in the same partition.

Multiple partitions are useful for setting up a strong security between groups of users, because each partition has its own set of Campaign system tables.

You must not create multiple partitions if groups of users have to share data with each other.

Each partition has its own set of configuration settings. You can customize Campaign for each group of users. However, all partitions share the same installation binaries.

With the same binaries for all partitions, you can minimize the installation and upgrade efforts for multiple partitions.

The utility to create multi-partition is available in the `$HOME_DIR/Platform/tools/bin` location.

Provide values for the following parameters in the Campaign chart:

- **PARTITIONS** - Name of the partition you want to configure. In case of multiple partitions specify partition name separated by a semi-colon. For example `partition2;partition3`.
- **SOURCE_PARTITION** - The name of the source partition to be replicated.
- **DEST_PARTITION** - The name of the destination partition to be created.
- **PARTITION_USER** - Specifies the user name of the admin user for the replicated partition. The name must be unique within the instance of Unica Platform.
- **PARTITION_GROUP** - Specifies the name of the Platform admin group that the utility creates. The name must be unique within the instance of Unica Platform.
- **CAMPAIGN_PARTITION2_DATABASE_HOST** - Host system details of the system hosting the Campaign Partition2 database.
- **CAMPAIGN_PARTITION2_DATABASE_PORT** - Port number of the Campaign Partition2 database.
- **CAMPAIGN_PARTITION2_DATABASE_NAME** - Name of the Campaign Partition2 database.
- **CAMPAIGN_PARTITION2_DATABASE_USERNAME** - Username to access the Campaign Partition2 database.
- **CAMPAIGN_PARTITION2_DATABASE_PASSWORD** - Password to access the Campaign Partition2 database.
- **CAMPAIGN_PARTITION2_DS_INITIAL_SIZE** - The initial size of the Campaign Partition2 datasource connection pool.
- **CAMPAIGN_PARTITION2_DS_MIN_IDLE** - The minimum number of idle connections (not connected to a database) in the Campaign Partition2 datasource connection pool.

- **CAMPAIGN_PARTITION2_DS_MAX_IDLE** - The maximum number of idle connections (not connected to a database) in the Campaign Partition2 datasource connection pool.
- **CAMPAIGN_PARTITION2_DS_MAX_TOTAL** - The maximum number of connections that the Campaign Partition2 datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused.
- **CAMPAIGN_PARTITION2_DS_STATEMENT_CACHE_SIZE** - Maximum number of statements that can be cached in the Campaign Partition2 datasource. Statement caching improves performance by caching executable statements that are used repeatedly.
- **CAMPAIGN_PARTITION2_JNDI_NAME** - JNDI name for Campaign Partition2.
- **CAMPAIGN_PARTITION2_POOL_NAME** - Pool name for Campaign Partition2.

The syntax to generate a partition is:

```
./multiPartition.sh >> output.out
```

After running the utility, restart the Platform and Campaign pod. After restarting the pods, login with `platform_admin`.

You can login with **PARTITION_USER** and the partition name you specify is used as the password for the `admin` user

Configurations for Director

ActiveMQ image is for reference or for tests. Unica does not own ActiveMQ. You can plug in your own ActiveMQ image in the helm chart.

To configure Director, complete 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 Unica Platform.
2. Select **Settings > Configuration**.
3. On the left pane, select **Unica Platform > Security > API management > Unica 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	<code>/authentication/login</code>
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	<code>/usr/partitions/*</code>
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	<code>/policy/partitions/*</code>
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	<code>/datasource/config</code>
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	<code>/datasource</code>
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	<code>/authentication/v1/login</code>
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	<code>/rest/v1/*</code>
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	<code>/rest/v2/*</code>
Block API access	Disabled
Secure API access over HTTPS	Disabled
Require authentication for API access	Enabled

Chapter 9. Migration of on-premises applications to Cloud Native Unica

You can migrate an on-premise version of Unica to the Cloud Native version. The Cloud Native version will be deployed on the application server.

Migration prerequisites

The prerequisites for the migration are as follows:

Basic prerequisites

- 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 Interact, 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.

SQL Server-specific prerequisites

If your database is SQL Server database, before executing the migration, manually execute the following SQL files on the Platform database:

- **If the Base version is 9.1.x**
 - `HOME_DIR/Platform/db/db/upgrade91to911/ManagerSchema_DB_Type_911upg.sql` `HOME_DIR/Platform/db/upgrade912to10/ManagerSchema_SqlServer_10upg.sql`;
 - `HOME_DIR/Platform/db/upgrade912to10/SqlServer_QRTZ_Scheduler_10_upgrade_Script.sql`;
`HOME_DIR/Platform/db/upgrade10001to10002/ManagerSchema_SqlServer_10002upg.sql`;

- HOME_DIR/Platform/db/upgrade10002to101/
ManagerSchema_SqlServer_101upg.sql; HOME_DIR/Platform/db/
upgrade101to11/ManagerSchema_SqlServer_11upg.sql;
- HOME_DIR/Platform/db/upgrade1101to111/
ManagerSchema_SqlServer_111upg.sql; HOME_DIR/Platform/db/
upgrade111to11102/ManagerSchema_SqlServer_11102upg.sql;
- HOME_DIR/Platform/db/upgrade11102to12/
ManagerSchema_SqlServer_12upg.sql

• **If the Base version is 9.1.2**

- HOME_DIR/Platform/db/upgrade912to10/
ManagerSchema_SqlServer_10upg.sql;
HOME_DIR/Platform/db/upgrade912to10/
SqlServer_QRTZ_Scheduler_10_upgrade_Script.sql;
- HOME_DIR/Platform/db/upgrade10001to10002/
ManagerSchema_SqlServer_10002upg.sql; HOME_DIR/Platform/db/
upgrade10002to101/ManagerSchema_SqlServer_101upg.sql;
- HOME_DIR/Platform/db/upgrade101to11/
ManagerSchema_SqlServer_11upg.sql; HOME_DIR/Platform/db/
upgrade1101to111/ManagerSchema_SqlServer_111upg.sql;
- HOME_DIR/Platform/db/upgrade111to11102/
ManagerSchema_SqlServer_11102upg.sql; HOME_DIR/Platform/db/
upgrade11102to12/ManagerSchema_SqlServer_12upg.sql

• **If the Base version is 10.0.0**

- HOME_DIR/Platform/db/upgrade10001to10002/
ManagerSchema_SqlServer_10002upg.sql; HOME_DIR/Platform/db/
upgrade10002to101/ManagerSchema_SqlServer_101upg.sql;
- HOME_DIR/Platform/db/upgrade101to11/
ManagerSchema_SqlServer_11upg.sql; HOME_DIR/Platform/db/
upgrade1101to111/ManagerSchema_SqlServer_111upg.sql;
- HOME_DIR/Platform/db/upgrade111to11102/
ManagerSchema_SqlServer_11102upg.sql; HOME_DIR/Platform/db/
upgrade11102to12/ManagerSchema_SqlServer_12upg.sql

- **If the Base version is 10.1.0**

- HOME_DIR/Platform/db/upgrade101to11/
ManagerSchema_SqlServer_11upg.sql; HOME_DIR/Platform/db/
upgrade1101to111/ManagerSchema_SqlServer_111upg.sql;
- HOME_DIR/Platform/db/upgrade111to11102/
ManagerSchema_SqlServer_11102upg.sql; HOME_DIR/Platform/db/
upgrade11102to12/ManagerSchema_SqlServer_12upg.sql

- **If the Base version is 11.0.0**

- HOME_DIR/Platform/db/upgrade101to11/
ManagerSchema_SqlServer_11upg.sql; HOME_DIR/Platform/db/
upgrade1101to111/ManagerSchema_SqlServer_111upg.sql;
- HOME_DIR/Platform/db/upgrade111to11102/
ManagerSchema_SqlServer_11102upg.sql; HOME_DIR/Platform/db/
upgrade11102to12/ManagerSchema_SqlServer_12upg.sql

- **If the Base version is 11.0.0**

- HOME_DIR/Platform/db/upgrade1101to111/
ManagerSchema_SqlServer_111upg.sql; HOME_DIR/Platform/db/
upgrade111to11102/ManagerSchema_SqlServer_11102upg.sql;
- HOME_DIR/Platform/db/upgrade11102to12/
ManagerSchema_SqlServer_12upg.sql

- **If the Base version is 11.1.0**

- HOME_DIR/Platform/db/upgrade111to11102/
ManagerSchema_SqlServer_11102upg.sql; HOME_DIR/Platform/db/
upgrade11102to12/ManagerSchema_SqlServer_12upg.sql

- **If the Base version is 11.1.0**

- HOME_DIR/Platform/db/upgrade111to11102/
ManagerSchema_SqlServer_11102upg.sql; HOME_DIR/Platform/db/
upgrade11102to12/ManagerSchema_SqlServer_12upg.sql

- **If the Base version is 12.0.0**

- HOME_DIR/Platform/db/upgrade11102to12/
ManagerSchema_SqlServer_12upg.sql

common-configMap configurations

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

Table 1. 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	"DBO"
DB_DRIVER_CLASS	com.microsoft.sqlserver.jdbc- .SQLServerDriver
AC_VERSION	"12.1.x"
ACI_UNICODE	"No"
CONFIGURE_ON_ERROR_PROMPT	"Yes"
LOCALE	"en_US"
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+To12.1
LISTENER_HOST_NAME	{{ .Release.Name }}-omnix-uni- ca-listener

Table 1. Configurable Parameters to perform an Upgrade (continued)

Parameter Name	Example Value
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	intl86
TARGET_SCHEMA_LRN	intl86
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

JVM option configurations

Add the JVM option `-DFAST_UPGRADE_VERSION=<BASE_VERSION>`. For example:

```
JAVA_OPTIONS="${JAVA_OPTIONS} -DFAST_UPGRADE_VERSION=8.6.x.
```

Performing the migration

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

1. To perform the migration, 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 migration logs from the mount location.

Configuring Unica Campaign post migration

To configure Unica Campaign post migration, complete the following steps:

Update the parameter **internalServerURL** to point to your Campaign pod.

For example, <http://hcl-unica-campaign:9125/Campaign>.

Configuring Unica Interact post migration

To configure Unica Interact post migration, complete the following steps:

1. Back up the current configurations.
2. Navigate to **Affinium > Campaign > partitions > partition1 > Interact > serverGroups**.
3. In Unica configuration, delete the old `serverGroup` and retain only the Interact `serverGroup`.
4. Define Interact as the `serverGroup` for the following configurations:
 - **flowchart** configuration within **Affinium > Campaign > partitions > partition1 > Interact**
 - **simulator** configuration within **Affinium > Campaign > partitions > partition1 > Interact**
5. Update the Interact design schema by replacing the old `serverGroup` name with a new name. Execute the following commands:
 - `update uaci_deployment set servergroupname='interact';`
 - `update uaci_ICTOSVRGROUP set servergroupname='interact';`
 - `update uaci_OfferMappingSG set servergroupname='interact';`

Configuring Unica Platform post migration

To configure Unica Platform post migration, complete the following steps:

1. The Unica 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 Platform, 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 10. Cloud Native Unica upgrade

To upgrade an earlier version of Cloud Native Unica to a newer version, complete the following steps:

1. Unica support team 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.
2. Download the required version image and push it to the Docker registry.
3. Update the image URLs in the helm charts.
4. Back up the Database and the file system before you start the upgrade.
5. Run the following helm upgrade command:

```
helm upgrade hcl unica -f ./unica/values-local.yaml --set  
service.hostname=kubernetes.nonprod.hclnpn.com --set  
service.applicationDomain='nonprod.hclnpn.com' --set  
ingress.enabled=true
```

6. Add upgrade related parameters in the `common-configMap.yaml` file when upgrading to version 12.1.4. In case you are upgrading from version 12.1.0.4 to version 12.1.4, there are two more parameters that you must add in the `common-configMap.yaml`.
7. Edit the helm chart `platform-deployment.yaml`. In the file, replace `args:`
`["chmod 755 /docker/unica && ./entrypoint.sh"]` with the following entry:
`args: ["chmod 755 /docker/unica && echo 'find /opt/
generate_datasource_snippet.sh -type f -print0 | xargs -0 sed -i \"s/
export DB_URL=\\$/#export DB_URL=/g\"' > /docker/unica/centos_patch.sh &&
chmod 777 /docker/unica/centos_patch.sh && ./entrypoint.sh"]`

Custom listener scripts and Cloud Native Unica container OS upgrade

Unica container OS upgrade from CentOS 8 to RHEL Universal Base Image (UBI) v8.5 may cause listener pod custom scripts to fail (applies to Cloud Native Unica versions 12.1.2 / 12.1.3).

1. Modify custom scripts as per RHEL UBI OS. Example: Database Client Installation Script on listener pod.
2. Centos8 OS was updated to RHEL UBI 8.5 because of CentOS 8 end of life.
3. Also, RHEL UBI containers are less vulnerable to security threats because of the frequent fixing and release cycle.

Chapter 11. 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 \(on page 36\)](#).
- For details related to scaling Interact containers, see [Scaling Interact containers \(on page 39\)](#).

Horizontal scaling of 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.



Note: Autoscaling of Unica containers is not supported.

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 \(on page 38\)](#)

Cluster mode

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

Also perform the following listener-related scaling activities:

- [Load balancing \(on page 38\)](#)
- [Listener integration \(on page 38\)](#)

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 > unicaACLlistener**
- **Settings > Configuration > Campaign > unicaACOLlistener**

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 &
```

Additionally, if your application server is WebLogic, the DB hostname should be a fully qualified domain name or else the Kubernetes service name will not work.

Scaling Journey engine containers

In Kubernetes, Journey engine are defined as StatefulSets.

Each Pod in a StatefulSet derives its hostname from:

- the name of the StatefulSet, and
- the ordinal of the Pod.

The service manages the Pod domain and the format is as follows:

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

Example:

The Journey engine pod is registered in the following format:

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

In the Helm chart, you can configure these pods in the `journey-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 Journey Engine shared home is `$HOME_DIR/Journey/Engine`.

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

Example:

```
$HOME_DIR/Journey/journey-0
```

In a Journey engine, by default, clustering is enabled. As soon as the engine starts, it creates `journey-0`, which is a copy of the engine folder. As you keep scaling the Journey engine, it creates folders named `journey-0`, `journey-1`, `journey-2`,..`journey-n`.

The logs for each pod will also be generated as `journey-0`, `journey-1`, `journey-2`,..`journey-n`.

Chapter 12. Using Red Hat OpenShift

You can use OpenShift to develop and run containerized applications. OpenShift allows applications, and the data centers that support them, to expand from just a few machines and applications to thousands of machines that serve millions of clients.

For detailed information related to Red Hat OpenShift Container Platform, see [OpenShift Container Platform documentation](#).

The benefits of using OpenShift Container Platform are as follows:

- Does not require separate charts as the OpenShift charts are customized, or updated, charts when compared to Kubernetes charts.
- Easy to manage and monitor using the OpenShift console.

To configure the changes required for Unica, complete the following steps:

1. Place the following items on a location that is accessible from the listener pod:
 - `unixodbc`
 - `libltdl.so.7`
 - `libltdl.so.7.30`
 - `mariadb driver` (must be installed and then copied to the required location)

Update the same in `campaign-configmap.yaml` file:

```
export ODBCINI=<driver-path>/etc/odbc.ini
export ODBCINST=<driver-path>/etc/odbcinst.ini
export ODBCSYSINI=<driver-path>/odbc1/etc
```



Note: `<driver-path>` is the path where you have copied the driver. For example, `/docker/unica/odbc1`.

2. In the `configmap.yaml` file, update the namespace for listener domain name.
3. Based on your setup, you can:

- update the `PVC.yaml` file before using it.
- avoid the `PVC.yaml` file.

Security Context Constraints for Unica on Red Hat OpenShift

For any Security Context Constraint (SCC), perform the following steps:

1. If `AllowPrivilegedContainer` is enabled (set to `TRUE`) or not enabled, set it to `FALSE`.
2. Do not assign root access to the users specified in the `deployment.yaml` file.
3. For pods that do not have a `gid` (group ID), perform the following configuration:

```
securityContext:
  runAsUser: 1000610000
```

The configuration ensures that the start user of the pods is `1000610000`. The `1000610000` user cannot switch to the `root` user or change the `root` user password.

4. For the Oracle client, in the listener pod, create a user for a valid group and perform the following configurations:

```
securityContext as :
  securityContext:
    runAsUser: 1000
    runAsGroup: 1001

oracle:x:1000:1000:~/home/oracle:/bin/bash
dba:x:1001:oracle
1000=oracle and 1001 = dba group
```

The configuration ensures that the Oracle user also cannot switch to the `root` user or change the `root` user password.

5. For the SCC (`anyuid`), configure the following values:

```
allowHostDirVolumePlugin: false
allowHostIPC: false
```



```

allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities: null
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups:
- system:cluster-admins
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: anyuid provides all features of the
restricted SCC
    but allows users to run with any UID and any GID.
    release.openshift.io/create-only: "true"
  creationTimestamp: "2020-08-24T17:55:03Z"
  generation: 6
  name: anyuid
  resourceVersion: "23505934"

  selfLink: /
apis/security.openshift.io/v1/securitycontextconstraints/anyuid
  uid: 43877aab-c522-4ca9-9575-e8b212749e29
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:
- MKNOD
runAsUser:

```

```
    type: RunAsAny
  seLinuxContext:
    type: MustRunAs
  supplementalGroups:
    type: RunAsAny
  users:
    - system:serviceaccount:unica:default
  volumes:
    - configMap
    - downwardAPI
    - emptyDir
    - persistentVolumeClaim
    - projected
    - secret
```

6. For the listener pod, remove all `chmod` or `su`.
7. In the listener `rc.unicaac`, remove the root user `check` and change it to `oracle`.
8. In the Journey configmap, update the namespace from `default` to `unica`.

Chapter 13. Deployment monitoring

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

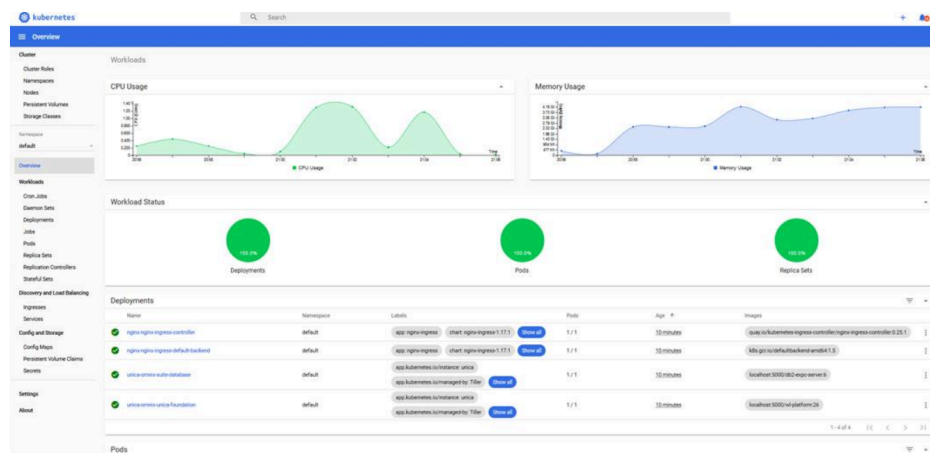
Use the Kubernetes Dashboard to:

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

You can also use the 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 14. 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 2. Unica products and their assigned pods for running the utilities

Unica Product Name	Pod Name	List of Utilities
Unica Campaign	Listener	For Campaign utilities, see Unica Campaign (on page 47) .
Unica Platform	Platform	For Platform utilities, see Unica Platform (on page 48) .
Unica Plan	Plan	For Plan utilities, see Unica Plan (on page 48) .

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 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`

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`

Chapter 15. Using secret to avoid passwords in plain text

To use a secret to avoid using passwords in plain text, complete the following steps:

1. On a Linux virtual machine, run the following command:

```
echo -n 'unica*03' | base64
```

You will see the following output: "dw5pY2EqMDM="

2. Create a YAML file (example `unicadbSecret.yaml`) and in the YAML file add the following parameters:

```
apiVersion: v1
kind: Secret
metadata:
  name: unica-db-token
type: Opaque
data:
  PLATFORM_DATABASE_PASSWORD: "dw5pY2EqMDM="
```

3. To use the password in Unica Platform, update the Platform deployment, and wherever envFrom exists, add the the following code:

```
envFrom:
- secretRef:
  name: unica-db-token
- configMapRef:
```

4. Either comment or delete the parameter **PLATFORM_DATABASE_PASSWORD**:
unica*03 from the `platform-configMap.yaml` file.



Note:



- The same `unicadbSecret.yaml` can be used for multiple Unica product database passwords. Repeat *Step 3* and *Step 4* for each products deployment and their respectiv `configmap.yaml` file. For example, in case of Unica Plan, with **PLAN_DATABASE_PASSWORD**: unica*03, add the following lines of code

```
apiVersion: v1
kind: Secret
metadata:
  name: unica-db-token
  type: Opaque
data:
  PLATFORM_DATABASE_PASSWORD: "dw5pY2EqMDM="
  PLAN_DATABASE_PASSWORD: "dw5pY2EqMDM="
```

In this case, update the Plan deployment and `configmap.yaml` file.

- Limit the secret size to 1 MB. If the secret size is more than 1 MB, split it into multiple tokens.

Chapter 16. Using AWS Secrets and Configuration Provider with Kubernetes Secret Store CSI Driver

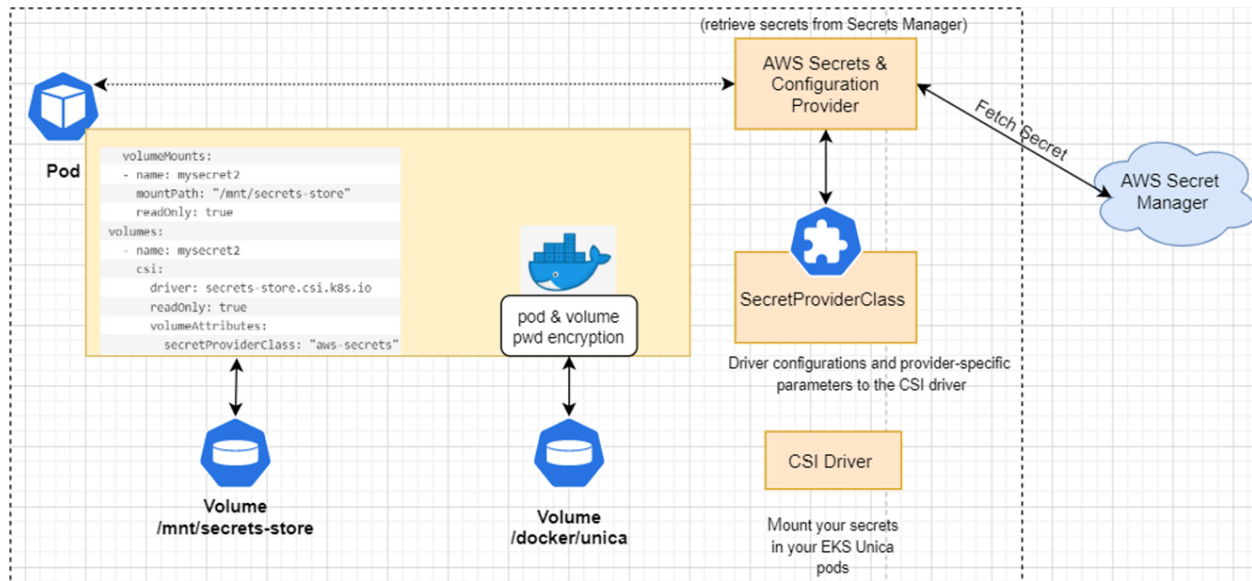
AWS Secrets Manager securely retrieves secrets from the AWS Secrets Manager for Amazon Elastic Kubernetes Service (Amazon EKS) Kubernetes pods.

AWS Secrets and Config Provider (ASCP) contains an easy-to-use plugin that provides secrets to applications that operate on Amazon EKS. The plugin supports industry-standard Kubernetes Secrets Store and Container Storage Interface (CSI) driver.

The benefits of ASCP are as follows:

- Provides compatibility for legacy Kubernetes workloads that fetched secrets through the file system or etcd.
- Securely store and manage your secrets in Secrets Manager.
- Retrieve secrets, using applications that run on Kubernetes, without writing a custom code.
- Use AWS Identity and Access Management (IAM) and resource policies on your secret to limit and restrict access to specific Kubernetes pods inside a cluster to tightly control secrets accessible by the pods.

AWS Secrets Manager Working Concept with Unica



AWS Secret Manager Implementation

To implement AWS Secret Manager, ensure that the prerequisites are met and the configurations are executed.

For more details, see the following topics:

- [Prerequisite Software for AWS Secret Manager \(on page 52\)](#)
- [Prerequisite Configurations for AWS Secret Manager \(on page 53\)](#)
- [Implementing AWS Secret Manager \(on page 54\)](#)

Prerequisite Software for AWS Secret Manager

The prerequisite software requirement for AWS Secret Manager are as follows:

- An AWS account
- AWS Command Line Interface installed
- `kubectl` installed
- Helm installed
- `eksctl` installed
- An existing EKS cluster

Prerequisite Configurations for AWS Secret Manager

Before implementing AWS Secret Manager, make the following configurations:

- An IAM policy, with permissions to retrieve secrets from Secret Manager.
- Your secret stored in Secrets Manager, for example `platsecret`, `campsecret`, and `plansecret` with keys `PLATFORM_DATABASE_PASSWORD`, `CAMPAIGN_DATABASE_PASSWORD`, and `PLAN_DATABASE_PASSWORD`:
 - keys should match the `configMap` entries
 - encryption key value must be `aws/secretsmanager`
- A user or `iamserviceaccount` that can modify your Kubernetes cluster.
- To the Docker registry, push the new set of images.
- To use the new image tags, update the `values.yaml` file.
- In the Helm chart, comment out the following `_PASSWORD` parameters from the `configMap.yaml` files:
 - `CAMPAIGN_DATABASE_PASSWORD`
 - `PLAN_DATABASE_PASSWORD`
 - `PLATFORM_DATABASE_PASSWORD`
- In the `common-configMap.yaml` file, add the following parameter:

```
TOMCAT_FACTORY: "com.unica.manager.tomcat.utils.TomcatDSFactory"
```

- Create secrets for the following Unica products with the corresponding names:

Unica Product	Secret name
Unica Campaign	<code>campsecret</code>
Unica Plan	<code>plansecret</code>
Unica Platform	<code>platsecret</code>

- Update the secret arn, secret name, and key in the following files (see the example for reference):

- `values.yaml`

```
awssecrets:
  secrets:
    iamserviceaccountforawssecretmanager: "suite-unica11"
    kubernetessecret: "application-api-key"
  platsecret:
    platformdatabasepassword: "PLATFORM_DATABASE_PASSWORD"
    secretsmanagerarnplat: "arn:aws:secretsmanager:ap-south-1:385481138434:secret:platsecret1-7dBQks"
```

- `deployment.yaml`

```
- name: {{ .Values.image.awssecrets.platsecret.platformdatabasepassword }}
  valueFrom:
    secretKeyRef:
      name: {{ .Values.image.awssecrets.secrets.kubernetessecret }}
      key: {{ .Values.image.awssecrets.platsecret.platformdatabasepassword }}
```

Implementing AWS Secret Manager

To implement AWS Secret Manager on your setup, complete the following steps:

1. Using a command line interface, restrict access to your pods using IAM roles for service accounts. Alternatively, you can also restrict access using a console.
2. To turn on Open ID Connect (OIDC), run the following `eksctl` command:

```
eksctl utils associate-iam-oidc-provider --region=<REGION>
--cluster=<CLUSTERNAME> --approve
```



Note:

- You must run the earlier mentioned command only once.
 - In the command, mentioned earlier, replace `<REGION>` and `<CLUSTERNAME>` with relevant and appropriate values.
3. For retrieving secrets from AWS Secret Manager, create a policy by running the following command:

```
aws iam create-policy --policy-name <my-policy> --policy-document
file://policy
```

A sample policy file follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [x`
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:ListSecretVersionIds"
      ],
      "Resource":
        "arn:aws:secretsmanager:ap-south-1:385481138434:secret:*"
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:ListSecrets",
      "Resource": "*"
    }
  ]
}
```

4. Create a service account role to associate the policy (created in *Step 2*) with your service account. To create a service account, run the following command:

```
eksctl create iamserviceaccount --name <SERVICE_ACCOUNT_NAME>
--namespace <NAMESPACE> --cluster <CLUSTERNAME> --attach-policy-arn
<IAM_policy_ARN> --approve --override-existing-serviceaccounts
```



Note: In the command, mentioned earlier, replace <NAMESPACE>, <CLUSTERNAME>, <IAM_policy_ARN>, and <SERVICE_ACCOUNT_NAME> with relevant and appropriate values.

5. To install the Kubernetes secrets store CSI driver, using helm with `syncSecret.enabled=true`, run the following commands:

a. Run the following command:

```
helm repo add secrets-store-csi-driver  
https://kubernetes-sigs.github.io/secrets-store-csi-driver/charts
```

b. If you do not require a periodical pull of updated secrets, initialize the driver by running the following command:

```
helm install csi-secrets-store  
secrets-store-csi-driver/secrets-store-csi-driver --set  
syncSecret.enabled=true --namespace kube-system
```

c. If you want to turn on automated rotation for the driver, using the rotation reconciler feature which is currently in alpha, run the following command:

```
helm -n kube-system install csi-secrets-store  
secrets-store-csi-driver/secrets-store-csi-driver --set  
enableSecretRotation=true --set rotationPollInterval=3600s
```



Note: You can adjust the rotation intervals, as per your requirements, to find an appropriate balance between API call cost consideration and rotation frequency

6. To install the ASCP, run the following command:

```
kubectl apply -f  
https://raw.githubusercontent.com/aws/secrets-store-csi-driver-provider-aws/main/deployment/aws-provider-installer.yaml
```

7. Create the custom resource **SecretProviderClass** and deploy it to sync with AWS secret with Kubernetes. For details, access the `spc.yaml` inside the Unica helm chart.

```
5 spec:
6   provider: aws
7   secretObjects:
8     - secretName: {{ .Values.image.awssecrets.secrets.kubernetessecret }} # the k8s secret name
9     type: Opaque
10    data:
11      - objectName: key1 # reference the corresponding parameter
12        key: {{ .Values.image.awssecrets.platsecret.platformdatabasepassword }}
13  parameters:
14    objects: |
15      - objectName: {{ .Values.image.awssecrets.platsecret.secretsmanagerarnplat }}
16        jmesPath:
17          - path: {{ .Values.image.awssecrets.platsecret.platformdatabasepassword }}
18            objectAlias: key1
```

8. Configure and deploy the pods to mount the volumes based on the configured secrets.

```
volumeMounts:
- name: volume-mount
  mountPath: /docker/unica
- name: api-secret
  mountPath: "/mnt/secrets-store"
  readOnly: true
```

```
1 spec:
2   volumes:
3     - name: volume-mount
4       {{- if .Values.persistence.enabled }}
5       persistentVolumeClaim:
6         {{- if .Values.persistence.existingClaim }}
7         claimName: {{ .Values.persistence.existingClaim }}
8         {{- else }}
9         claimName: {{ include "unica.fullname" . }}
10        {{- end }}
11      {{- else }}
12      emptyDir: {}
13      {{- end }}
14    - name: api-secret
15      csi:
16        driver: secrets-store.csi.k8s.io
17        readOnly: true
18        volumeAttributes:
19          secretProviderClass: "aws-secrets"
```

9. In the `rbac.yaml` file, assign the **ClusterRoleBinding** permissions to the `iamservice` account, created in *Step 3*, for internal Kubernetes communication.

```
36 subjects:
37   - name: {{ include "unica.fullname" . }}
38     namespace: {{ .Release.Namespace | quote }}
39     kind: ServiceAccount
40   - name: {{ .Values.image.awssecrets.secrets.iamserviceaccountforawssecretmanager }}
41     namespace: {{ .Release.Namespace | quote }}
42     kind: ServiceAccount
43 {{- end -}}
```

Chapter 17. Enabling Multicast using Weave-Net CNI plugin on AWS EKS cluster

You can enable multicasting on AWS EKS cluster only for Kubernetes versions 1.21 or above,

- Create a role on AWS having the necessary privileges for creating AWS clusters (example: `AWS_EKS_CLUSTER_ROLE`).
- Create a minimum of two subnets within the VPC. You must create the cluster within this VPC.

To enable multicasting on AWS EKS cluster using Weave-Net CNI plugin, complete the following steps:

1. Use the AWS CLI and create an EKS cluster without any node group.



Note: Multicasting will not work if you create clusters using AWS web console.

Sample Command:

```
aws eks create-cluster --region <region-name> --name  
<cluster-name> --kubernetes-version 1.21 --role-arn  
<full-arn-of-the-role> --resources-vpc-config  
subnetIds=<subnet-id1>,<subnet-id2>,...<subnet-idn>
```

2. Run the following command to delete the aws-node default daemon-set:

```
kubectl delete ds aws-node -n kube-system
```

This disables the default vpc-cni plugin.

3. Confirm if your security group allows TCP port 6783 and UDP ports 6783 and 6784.
If your security group does not allow these ports, add the necessary firewall rules to your security groups to allow these ports.
4. Run the following command to delete the kube-proxy ds:


```
kubectl delete ds kube-proxy -n kube-system
```

5. Run the following command to create an add-on for Kube-proxy:

```
aws eks create-addon --cluster-name <your-cluster-name> --addon-name  
kube-proxy --resolve-conflicts OVERWRITE
```

This will add the latest kube-proxy add-on to the cluster, based on the Kubernetes cluster version.

6. Run the following command to apply weave-net daemoset:

```
kubectl apply -f  
"https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version |  
base64 | tr -d '\n')"
```

7. Verify the Daemon sets on cluster. There should be two daemon sets for Weave and correspondingly two Kube-proxy daemon sets.
8. Add the node group to the Cluster and wait till the nodes are created and all the required nodes are ready.
9. Deploy the Unica product and verify the Multicasting.

Chapter 18. FAQs and troubleshooting

This section covers the frequently asked questions and troubleshooting issues.

To view the list of FAQs, see [Frequently Asked Questions \(on page 61\)](#)

For information related to Troubleshooting, see [Troubleshooting Issues \(on page 64\)](#)

Frequently Asked Questions

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

The list of FAQs are as follows:

- [Question 1 \(on page 61\)](#)
- [Question 2 \(on page 61\)](#)
- [Question 3 \(on page 62\)](#)
- [Question 4 \(on page 62\)](#)
- [Question 5 \(on page 63\)](#)
- [Question 6 \(on page 63\)](#)
- [Question 7 \(on page 64\)](#)

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-premises 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 exists in the provided path.

Question 3

Why has Cloud Native Unica installed `JRE9` and `JDK8` on my system?

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

- Provide the path of `JRE9` for the parameter `DOCKER_JAVA_HOME`. Cloud Native Unica uses `JRE9` for installation tasks.
- Provide the path of `JDK8` for the parameter `JAVA_HOME`. The products of Unica uses `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 Cloud Native Unica, 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 `/AC00ptAdmin.sh` work?

For `/AC00ptAdmin.sh` to work, update the following parameters in the `/AC00ptAdmin.sh` file:

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

Use the `-async` option while running `AC00ptAdmin` utility on docker environments.

Using the `-async` utility triggers the desired operation on an Optimize session in the background before exiting.

Example: `./AC00ptAdmin.sh -u "user_name" -p "password" -sn "OptimizeSessionName" -async`



Note: Not using `-async` may trigger an Optimize session run, but the polling, related to the session run progress, will fail.

Question 7

How are the Security Vulnerabilities are fixed?

Answer

- Unica fixes security vulnerabilities with the every new release.
- Upon request, interim fixes are also provided with new set of docker images.

Troubleshooting Issues

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

The list of troubleshooting issues are as follows:

- [Question 1 \(on page 64\)](#)
- [Question 2 \(on page 65\)](#)
- [Question 3 \(on page 65\)](#)

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 WebLogic, complete the following steps:

1. Save your work and confirm that all users have logged off.
2. Locate the running docker container using the command `kubectl get pods`.
3. Access the container using the command `kubectl exec -it <name of the container> bash`.
4. Go to the bin directory of the domain as specified in the `configMap.yaml` file for the **WLS_DOMAIN_LOCATION** parameter. For more information, see [Configuring WebLogic for Cloud Native Unica \(on page 11\)](#).
5. Stop the domain by running the command `stopWebLogic.sh`.

6. To restart the domain, run the command `startWebLogic.sh` in the background.
7. To exit the docker container, press CTRL+D.

Question 2

Cannot select supported locales for Plan.

When installing Plan using Cloud Native Unica, 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 19. 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.

Chapter 20. Appendix: Description of Helm chart parameters

The following topics contain description of the parameters present in the `configMap` **YAML** files:

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 <code>INSTALL</code> or <code>UP-GRADE</code> .
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.

Table 3. Data Parameters (continued)

Parameter name	Parameter description
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: • Platform – PLT • Campaign – CMP • Optimize – OPT • Director – DIR • Plan – PLN • Interact – INT • Centralized Offer Management - OFFER • Insights Reports - BIRT If you want to install all products you should provide the value as follows: PLT_CMP_INT_PLN_OPT_DIR If your database is MariaDB, Director will not work on MariaDB. In this case, you must provide the following value: PLT_CMP_INT_PLN_OPT
SERVER_TYPE	The application server installed.

Table 3. Data Parameters (continued)

Parameter name	Parameter description
IS_UNICODE	Set <code>TRUE</code> if Unica is installed to support Unicode. Set <code>FALSE</code> if Unica is installed without support for Unicode
PROTOCOL	The protocol used. For example, HTTP or HTTPS.
UPGRADE_FROM_TO	11.1+To12.1
AC_VERSION	"12.1.x"
ACI_UNICODE	"No"
CONFIGURE_ON_ERROR_PROMPT	"Yes"

Table 4. Miscellaneous Parameters

Parameter name	Parameter description
SOURCE_SCHEMA	"CAMP86"
TARGET_SCHEMA	"DBO"
DB_DRIVER_CLASS	com.microsoft.sqlserver.jdbc-.SQLServerDriver
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.
DB_PLAN_PORT	The database port number of the Plan system.
DB_PLAN_HOST_NAME	The database host name of the Plan system.

Table 4. Miscellaneous Parameters (continued)

Parameter name	Parameter description
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, <code>oracle</code> .
DB_TYPE_UTILS	The name of the database utilities used in the system. For example, <code>oracle</code> .
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 <code>jdbc:oracle:thin</code> .
DIALECT	The Hibernate dialect. Each database has a different dialect. For example, the Oracle database dialect is <code>org.hibernate.dialect.Oracle10gDialect</code> .
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 <code>jdbc:oracle:thin</code> .
JDBC_DRIVER_JAR_LOCATION	The location of the <code>JDBC</code> driver JAR file.
DB_DRIVER_JAR	The location of the database driver JAR file.

Table 4. Miscellaneous Parameters (continued)

Parameter name	Parameter description
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 <code>modules/jdbcmodule/main/module.xml</code> (name of the JDBC jar).
MDB_ENCODING	The encoding format used for MariaDB.
MDB_COLLATION	Valid values are <code>utf8_general_ci</code> and <code>utf8_unicode_ci</code> .
MAX_CONNECTIONS	The maximum concurrent connections supported.

If the JDBC URL contains additional properties, please use the parameters mentioned in the [Table 5: JDBC Parameters \(on page 71\)](#) table using the format provided in the following example:

```
jdbc:sqlserver://localhost;databaseName=AdventureWorks;MultiSubnetFailover=true;
```

Table 5. JDBC Parameters

Parameter name	Parameter description
JDBC_URL_PROD	JDBC URL of the Prod datasource of Interact. You can provide custom JDBC URL with JDBC properties.

Table 5. JDBC Parameters (continued)

Parameter name	Parameter description
JDBC_URL_TEST	JDBC URL of the Prod datasource of Test.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_LRN	JDBC URL of the Prod datasource of learning.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_INT05	JDBC URL of the Prod datasource of Interact.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_INT	JDBC URL of the Prod datasource of Interact runtime.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_PLATFORM	JDBC URL of the Prod datasource of platform.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_CAMPAIGN	JDBC URL of the Prod datasource of Campaign.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_PLAN	JDBC URL of the Prod datasource of Plan.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_CONTACTCENTRAL	JDBC URL of the Prod datasource of Contact Central.You can provide custom JDBC URL with JDBC properties.

Table 5. JDBC Parameters (continued)

Parameter name	Parameter description
JDBC_URL_JOURNEY	JDBC URL of the Prod datasource of Journey.You can provide custom JDBC URL with JDBC properties.
JDBC_URL_JOURNEYREPORT	JDBC URL of the Prod datasource of Journey Report.You can provide custom JDBC URL with JDBC properties.

Table 6. Parameters when Installing 12.1.4 or Upgrading to 12.1.4

Add the following parameters when installing Cloud Native Unica 12.1.4 or upgrading to Cloud Native Unica 12.1.4

Parameter name	Parameter description
Details	removeAbandoned is a Flag to remove abandoned connections if they exceed the removeAbandonedTimeout.
testOnBorrow	Indicates whether objects are validated before being borrowed from the pool. For an efficient validation, if objects fail validation, they are dropped from the pool and the system attempts to borrow another object.
PLATFORM_DATA_SOURCE_PARAMETERS	removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'
PLAN_DATA_SOURCE_PARAMETERS	removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'

Table 6. Parameters when Installing 12.1.4 or Upgrading to 12.1.4

Add the following parameters when installing Cloud Native Unica 12.1.4 or upgrading to Cloud Native Unica 12.1.4

(continued)

Parameter name	Parameter description
JOURNEYWEB_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>
CAMPAIGN_DATA_SOURCE_PARAMETERS	
INTERACT_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>
INTERACT_PROD_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>
INTERACT_TEST_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>
INTERACT_LEARNING_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>
INTERACT_CHRH_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>
COLLABORATE_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>

Table 6. Parameters when Installing 12.1.4 or Upgrading to 12.1.4

Add the following parameters when installing Cloud Native Unica 12.1.4 or upgrading to Cloud Native Unica 12.1.4

(continued)

Parameter name	Parameter description
CONTACTCENTRAL_DATA_SOURCE_PARAMETERS	<code>removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'</code>

Table 7. Parameters when Upgrading from 12.1.0.4 to 12.1.4

Add the following parameters only when upgrading Cloud Native Unica from version 12.1.0.4 to version 12.1.4

Parameter name	Parameter description
INSTALL_COMMAND1	<code>cp /docker/source/unica/Platform/tools/bin/manager-persistence-StandAlone.xml /docker/unica/Platform/tools/bin/</code>
INSTALL_COMMAND2	<code>cp /docker/source/unica/Platform/tools/bin/manager-persistence-StandAlone.xml /docker/unica/install/</code>

Audience Central configurations

To configure Audience Central for Cloud Native Unica, make the necessary modifications to the `audiencecentral-configMap.yaml` file.

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

Table 8. Common parameters for Audience Central

Parameter name	Parameter description
AUDIENCECENTRAL_PRODUCT_NAME	Audiencecentral
AUDIENCE_CENTRAL_WAR_NAME	AudienceCentral.war
AUDIENCECENTRAL_APPLICATION_NAME	audiencecentral
AUDIENCECENTRAL_DOMAIN_USERNAME	root
AUDIENCECENTRAL_DOMAIN_PASSWORD	unica*03

Table 9. Application Server-related parameters for Audience Central

Parameter name	Parameter description
AUDIENCECENTRAL_HOST_NAME	{{ .Release.Name }}-unica-audience-central
AUDIENCECENTRAL_MANAGEMENT_PORT	9065
AUDIENCECENTRAL_MANAGEMENT_HTTPS_PORT	9994
AUDIENCECENTRAL_AJP_PORT	8009
AUDIENCECENTRAL_HTTP_PORT	9139
AUDIENCECENTRAL_HTTPS_PORT	9445
AUDIENCECENTRAL_RECOVERY_ENV_PORT	4713
AUDIENCECENTRAL_STATUS_MANAGER_PORT	4714

Table 9. Application Server-related parameters for Audience Central (continued)

Parameter name	Parameter description
AUDIENCECENTRAL_MIN_HEAP	1024m
AUDIENCECENTRAL_MAX_HEAP	2048m
AUDIENCECENTRAL_URL	{{ include ip.protocol . }}:// {{ .Values.service.hostname }}/Au- dienceCentral
AUDIENCECENTRAL_INTERNAL_URL	http://{{ .Release.Name }}-uni- ca-audiencecentral:9139/Audience- Central
PRODUCT_OPTS_AUDIENCECENTRAL	-DAUDIENCE_CENTRAL_HOME=/dock- er/unica/AudienceCentral/ -DEN- ABLE_NON_PROD_MODE=true

Table 10. Database-related parameters for Audience Central

Parameter name	Parameter description
AUDIENCECENTRAL_USER_JNDI_NAME	{{ .Values.audiencecentralData- .audiencecentralConfigMapData.AU- DIENCECENTRAL_USER_JNDI_NAME }}
AUDIENCECENTRAL_USER_POOL_NAME	{{ .Values.audiencecentralData- .audiencecentralConfigMapData.AU- DIENCECENTRAL_USER_POOL_NAME }}
AUDIENCECENTRAL_USER_DATABASE_- HOST	{{ .Values.audiencecentralData- .audiencecentralConfigMapData.AU- DIENCECENTRAL_USER_DATABASE_HOST }}
AUDIENCECENTRAL_USER_DATABASE_- PORT	{{ .Values.audiencecentralData- .audiencecentralConfigMapData.AU- DIENCECENTRAL_USER_DATABASE_PORT }}

Table 10. Database-related parameters for Audience Central (continued)

Parameter name	Parameter description
AUDIENCECENTRAL_USER_DATABASE_NAME	{{ .Values.audiencecentralData.audiencecentralConfigMapData.AUDIENCECENTRAL_USER_DATABASE_NAME }}
AUDIENCECENTRAL_USER_DATABASE_USERNAME	{{ .Values.audiencecentralData.audiencecentralConfigMapData.AUDIENCECENTRAL_USER_DATABASE_USERNAME }}
AUDIENCECENTRAL_USER_DATABASE_PASSWORD	{{ .Values.audiencecentralData.audiencecentralConfigMapData.AUDIENCECENTRAL_USER_DATABASE_PASSWORD }}
AUDIENCECENTRAL_USER_DS_INITIAL_SIZE	{{ .Values.audiencecentralData.audiencecentralDSMData.AUDIENCECENTRAL_USER_DS_INITIAL_SIZE }}
AUDIENCECENTRAL_USER_DS_MIN_IDLE	{{ .Values.audiencecentralData.audiencecentralDSMData.AUDIENCECENTRAL_USER_DS_MIN_IDLE }}
AUDIENCECENTRAL_USER_DS_MAX_IDLE	{{ .Values.audiencecentralData.audiencecentralDSMData.AUDIENCECENTRAL_USER_DS_MAX_IDLE }}
AUDIENCECENTRAL_USER_DS_MAX_TOTAL	{{ .Values.audiencecentralData.audiencecentralDSMData.AUDIENCECENTRAL_USER_DS_MAX_TOTAL }}
AUDIENCECENTRAL_USER_DS_STATEMENT_CACHE_SIZE	{{ .Values.audiencecentralData.audiencecentralDSMData.AUDIENCECENTRAL_USER_DS_STATEMENT_CACHE_SIZE }}

Table 10. Database-related parameters for Audience Central (continued)

Parameter name	Parameter description
AUDIENCECENTRAL_USER_DATA_SOURCE_PARAMETERS	{{ .Values.audiencecentralData.audiencecentralDSMData.AUDIENCECENTRAL_USER_DATA_SOURCE_PARAMETERS }}

Campaign configurations

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

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

Table 11. 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.
USER_DATABASES	Helps in setting up user database. Plug in installations scripts for a seamless start-up of an instance. For example, a scaled listener instance.
USER_ORA_HOST_NAME	The host name of the Oracle user.

Table 12. 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 data-source connection pool.
CAMPAIGN_DS_MIN_IDLE	The minimum number of idle connections (not connected to a database) in 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 data-

Table 12. Database-related parameters for Campaign (continued)

Parameter name	Parameter description
	source. 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.
ORACLE_CLIENT_INSTALL_COMMAND	Command to install the Oracle 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 12. Database-related parameters for Campaign (continued)

Parameter name	Parameter description
MARIADB_CLIENT_INSTALL_COMMAND	Command to install the MariaDB client on the listener pod.
MARIADB_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.
SQLSERVER_CLIENT_INSTALL_SCRIPT	The path that contains the <code>sqlserv-er.sh</code> file.
USER_DB2_PORT	The port number to access the DB2 database.
USER_DB2_DB_NAME	The name of the DB2 database user.
USER_DB2_DB_USER	The username of the DB2 database user.
USER_DB2_DB_USER_PASSWORD	The password for the DB2 database user.
ASM_User_For_DB2_Credentials	The <code>asm_admin</code> credentials for DB2 data-source.
ASM_User_NZ_Data_Source_Name	The <code>asm_admin</code> user configured for the NZ datasource.
ASM_User_For_SQLSERVER_Credentials	The <code>asm_admin</code> credentials for SQL Server datasource.
ASM_User_SQLSERVER_Data_Source_Name	The <code>asm_admin</code> user configured for the SQL Server datasource.
ASM_User_DB2_Data_Source_Name	The <code>asm_admin</code> user configured for the DB2 datasource.
ASM_User_ORA_Data_Source_Name	The <code>asm_admin</code> user configured for the Oracle datasource.

Table 12. Database-related parameters for Campaign (continued)

Parameter name	Parameter description
ASM_User_For_ORA_Credentials	The <code>asm_admin</code> credentials for the Oracle datasource.
USER_ORA_DB_USERNAME	The username of the Oracle database user.
USER_ORA_DB_USER_PASSWORD	The password of the Oracle database user.
USER_ORA_PORT	The port number of the of the configured database user.
USER_ORA_SID	The <code>SID</code> details of the Oracle user.
ASM_User_NZ_Data_Source_Name	The <code>asm_admin</code> user configured for the NZ datasource.
ASM_User_For_NZ_Credentials	The <code>asm_admin</code> credentials for the NZ datasource.
USER_NZ_DB_USERNAME	The username of the NZ database user.
USER_NZ_DB_USER_PASSWORD	The password for the NZ database user.
USER_NZ_HOST_NAME	The host name of the NZ database user.
USER_NZ_PORT	The port number to access the NZ database.
USER_NZ_DB_NAME	The database name of the NZ database user.
USER_MARIA_HOST_NAME	The host name of the MariaDB database user.
USER_SQLSERVER_DB_NAME	The database name of the SQL Server database user.

Table 12. Database-related parameters for Campaign (continued)

Parameter name	Parameter description
USER_SQLSERVER_HOST_NAME	The host name of the SQL Server database user.
USER_SQLSERVER_PORT	The port number to access the SQL Server database.
USER_SQLSERVER_NAME	The host name of the SQL Server database user.
USER_SQLSERVER_USER	The username of the SQL Server database user.
CAMPAIGN_DSN_NAME	The dsname value of the respective database.
ORACLE_ODBC_DRIVER	The path or the location of the Oracle ODBC driver on your system.
DB_TEMPLATE	The name of the database template used. This is used for configuring ODBC connection in Oracle.

Table 13. 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.

Table 13. Application Server-related parameters for Campaign (continued)

Parameter name	Parameter description
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 14. 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.
LISTENER_TYPE	Specify the type of Listener.
CLUSTER_DOMAIN	Define the cluster domain. For example, <code>listener.default.svc.cluster-local</code> .
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.

Table 14. Listener-related parameters for Campaign (continued)

Parameter name	Parameter description
CAMPPORT	The deployment port for Campaign.
CLUSTER_DEPLOYMENT	Set TRUE if clustered deployment is supported or FALSE if clustered deployment is not supported.
ORACLE_CLIENT_SETUP_FILE_EXTRACT_COMMAND	The command to extract the Oracle tar/gz client setup file
DB2_CLIENT_SETUP_FILE_EXTRACT_COMMAND	The command to extract the DB2 tar/gz client setup file

Centralized Offer Management configurations

To configure Centralized Offer Management for Cloud Native Unica, make the necessary modifications to the **offer-configMap.yaml** file.

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

Table 15. Application server-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.
COM_MANAGEMENT_HTTPS_PORT	The management HTTPS port number for the Centralized Offer Management system.

Table 15. Application server-related parameters of Centralized Offer Management (continued)

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.
COM_MIN_HEAP	The minimum heap size allocated for Centralized Offer Management.
COM_MAX_HEAP	The maximum heap size allocated for Centralized Offer Management.

Table 16. Common parameters of Centralized Offer Management

Parameter name	Parameter description
PRODUCT_OPTS_COM	Product specific options for Centralized Offer Management.
COM_PRODUCT_NAME	The name assigned for Centralized Offer Management.
CENTRALIZED_OFFERS_-WAR_NAME	The name of the WAR file.
COM_APPLICATION_NAME	The name of the main application. For example, Unica.
COM_DOMAIN_USERNAME	The domain username for Centralized Offer Management.
COM_DOMAIN_PASSWORD	The domain password for Centralized Offer Management.

Collaborate configurations

To configure the Collaborate for Cloud Native Unica, make the necessary modifications to the `collaborate-configMap.yaml` file.

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

Table 17. Common parameters of Collaborate configuration

Parameter name	Parameter description
COLLABORATE_HOST	The name of the Collaborate host system.
COLLABORATE_PORT	The port number of the Collaborate host system.
COLLABORATE_JNDI_NAME	JNDI name for Collaborate.

Table 17. Common parameters of Collaborate configuration (continued)

Parameter name	Parameter description
COLLABORATE_POOL_NAME	Pool name for Collaborate.
PRODUCT_OPTS_COLLABORATE	Product-specific options for Collaborate.
COLLABORATE_PRODUCT_NAME	The name assigned for Collaborate.
COLLABORATE_WAR_NAME	The name of the WAR file.
COLLABORATE_APPLICATION_NAME	The name of the main application. For example, Unica.
COLLABORATE_DOMAIN_USERNAME	The domain username for Collaborate.
COLLABORATE_DOMAIN_PASSWORD	The domain password for Collaborate.

Table 18. Database parameters of Collaborate configuration

Parameter name	Parameter description
COLLABORATE_DATA_BASE_HOST	Host system details of the system hosting the Collaborate database.
COLLABORATE_DATA_BASE_PORT	Port number of the Collaborate database.
COLLABORATE_DATA_BASE_USERNAME	Username to access the Collaborate database.
COLLABORATE_DATA_BASE_PASSWORD	Password to access the Collaborate database.
COLLABORATE_DATA_BASE_NAME	Name of the Collaborate database.

Table 18. Database parameters of Collaborate configuration (continued)

Parameter name	Parameter description
COLLABORATE_DS_-INITIAL_SIZE	The initial size of the Collaborate datasource connection pool.
COLLABORATE_DS_MIN_IDLE	The minimum number of idle connections (not connected to a database) in the Collaborate datasource connection pool.
COLLABORATE_DS_MAX_IDLE	The maximum number of idle connections (not connected to a database) in the Collaborate datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool.
COLLABORATE_DS_MAX_TOTAL	The maximum number of connections that the Collaborate datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused.
COLLABORATE_DS_STATEMENT_CACHE_SIZE	Maximum number of statements that can be cached in the Collaborate datasource. Statement caching improves performance by caching executable statements that are used repeatedly.

Table 19. Application server parameters of Collaborate configuration

Parameter name	Parameter description
COLLABORATE_URL	The URL to access Collaborate.
COLLABORATE_HOST_NAME	The system host name of Collaborate.
COLLABORATE_MANAGEMENT_PORT	The management port number for the Collaborate system.
COLLABORATE_MANAGEMENT_HTTPS_PORT	The management HTTPS port number for the Collaborate system.

Table 19. Application server parameters of Collaborate configuration (continued)

Parameter name	Parameter description
COLLABORATE_AJP_PORT	The AJP port number for the Collaborate system.
COLLABORATE_HTTP_PORT	The HTTP port number for the Collaborate system.
COLLABORATE_HTTPS_PORT	The HTTPS port number for the Collaborate system.
COLLABORATE_RECOVERY_ENV_PORT	The recovery environment port number of the Collaborate system.
COLLABORATE_STATUS_MANAGER_PORT	The status manager port number of the Collaborate system.
COLLABORATE_MIN_HEAP	The maximum heap size allocated for Collaborate.
COLLABORATE_MAX_HEAP	The maximum heap size allocated for Collaborate.

Contact Central configurations

To configure Contact Central for Cloud Native Unica, make the necessary modifications to the `contactcentral-configMap.yaml` file.

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

Table 20. Common Contact Central parameters

Parameter name	Parameter description
CONTACTCENTRAL_JNDI_NAME	JNDI name for Contact Central.
CONTACTCENTRAL_POOL_NAME	Pool name for Contact Central.
CONTACTCENTRAL_URL	The URL to access Contact Central.

Table 20. Common Contact Central parameters (continued)

Parameter name	Parameter description
CONTACTCENTRAL_INTERNAL_URL	The internal URL to access/link Contact Central from other applications.
PRODUCT_OPTS_CONTACTCENTRAL	Product specific options for Contact Central.
CONTACTCENTRAL_PRODUCT_NAME	The name assigned for Contact Central.
CONTACT_CENTRAL_WAR_NAME	The name of the WAR file.
CONTACTCENTRAL_APPLICATION_NAME	The name of the main application. For example, Unica.
CONTACTCENTRAL_DOMAIN_USERNAME	The domain username for Contact Central.
CONTACTCENTRAL_DOMAIN_PASSWORD	The domain password for Contact Central.

Table 21. Database-related parameters for Contact Central

Parameter name	Parameter description
CONTACTCENTRAL_DATABASE_HOST	Host system details of the system hosting the Contact Central database.
CONTACTCENTRAL_DATABASE_PORT	Port number of the Contact Central database.
CONTACTCENTRAL_DATABASE_USERNAME	Username to access the Contact Central database.
CONTACTCENTRAL_DATABASE_PASSWORD	Password to access the Contact Central database.
CONTACTCENTRAL_DATABASE_NAME	Name of the Contact Central database.

Table 21. Database-related parameters for Contact Central (continued)

Parameter name	Parameter description
CONTACTCENTRAL_DS_INITIAL_SIZE	The initial size of the Contact Central datasource connection pool.
CONTACTCENTRAL_DS_MIN_IDLE	The minimum number of idle connections (not connected to a database) in the Contact Central datasource connection pool.
CONTACTCENTRAL_DS_MAX_IDLE	The maximum number of idle connections (not connected to a database) in the Contact Central datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool.
CONTACTCENTRAL_DS_MAX_TOTAL	The maximum number of connections that the Contact Central datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused.
CONTACTCENTRAL_DS_STATEMENT_CACHE_SIZE	Maximum number of statements that can be cached in the Contact Central datasource. Statement caching improves performance by caching executable statements that are used repeatedly.

Table 22. Application Server-related parameters for Contact Central

Parameter name	Parameter description
CONTACTCENTRAL_HOST_NAME	The system host name of Contact Central.

Table 22. Application Server-related parameters for Contact Central (continued)

Parameter name	Parameter description
CONTACTCENTRAL_MANAGEMENT_PORT	The management port number for the Contact Central system.
CONTACTCENTRAL_MANAGEMENT_HTTPS_PORT	The management HTTPS port number for the Contact Central system.
CONTACTCENTRAL_AJP_PORT	The AJP port number for the Contact Central system.
CONTACTCENTRAL_HTTP_PORT	The HTTP port number for the Contact Central system.
CONTACTCENTRAL_HTTPS_PORT	The HTTPS port number for the Contact Central system.
CONTACTCENTRAL_RECOVERY_ENV_PORT	The recovery environment port number of the Contact Central system.
CONTACTCENTRAL_STATUS_MANAGER_PORT	The status manager port number of the Contact Central system.
CONTACTCENTRAL_MIN_HEAP	The maximum heap size allocated for Contact Central.
CONTACTCENTRAL_MAX_HEAP	The maximum heap size allocated for Contact Central.

Content Integration configurations

To configure Content Integration for Cloud Native Unica, 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 23. Application server-related parameters of Content Integration

Parameter name	Parameter description
ASSET_HOST_NAME	The system host name of Content Integration.
ASSET_MANAGEMENT_PORT	The management port number for the Content Integration system.
ASSET_MANAGEMENT_HTTPS_PORT	The management HTTPS port number for the Content Integration system.
ASSET_AJP_PORT	The AJP port number for the Content Integration system.
ASSET_HTTP_PORT	The HTTP port number for the Content Integration system.
ASSET_HTTPS_PORT	The HTTPS port number for the Content Integration system.
ASSET_RECOVERY_ENV_PORT	The recovery environment port number of the Content Integration system.
ASSET_STATUS_MANAGER_PORT	The status manager port number of the Content Integration system.
PRODUCT_OPTS_ASSET	Product specific options for Content Integration.
ASSET_PRODUCT_NAME	The name assigned ofr the Content Integration
ASSET_WAR_NAME	The name of the WAR file.
ASSET_APPLICATION_NAME	The name of the main application. For example, Unica.
ASSET_DOMAIN_USERNAME	The domain username for Content Integration.

Table 23. Application server-related parameters of Content Integration (continued)

Parameter name	Parameter description
ASSET_DOMAIN_PASSWORD	The domain password for Content Integration.

Director configurations

To configure Director for Cloud Native Unica, 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 24. Common parameters of Director

Parameter name	Parameter description
activemq_enableEvents	Valid values are Yes or No.
activemq_url	Active MQ URL. For example, <code>tcp://unica-omnix-unica-activemq:61616</code> .
Data_Source_For_ActiveMQ_message_broker_credentials	Data source for ACTIVEMQ. For example, <code>ACTIVEMQ_CRED_DS</code> .
data_sources_for_activemq	Platform username.
activemq_queueName	Flowchart information. For example, <code>campaign</code> .

Table 25. Configuration parameters of Director

Parameter name	Parameter description
director_http_port	Director server port. The default port is 9128.

Table 25. Configuration parameters of Director (continued)

Parameter name	Parameter description
director_file_down	The download path used to store the downloaded log files from the Campaign server. For example, <code>/docker/unica/Director/Server/Downloads</code> .
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 26. 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.
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.

Insights Reports configurations

To configure Insights Reports reports for Cloud Native Unica, 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:

Update the following configurations:

- configurations at Affinium | Plan | umoConfiguration | reports.
- `reportsAnalysisSectionHome` -> Plan/Affinium Plan
- `reportsAnalysisTabHome` -> Plan/Affinium Plan - Object Specific Reports

After updating the configurations, restart the pods for Plan and Insights Reports.

Table 27. Common Insights Reports parameters

Parameter name	Parameter description
INSIGHTS_PRODUCT_NAME	The name assigned for Insights Reports.
INSIGHTS_WAR_NAME	The name of the WAR file.
INSIGHTS_APPLICATION_NAME	The name of the main application. For example, Unica.
INSIGHTS_DOMAIN_USERNAME	The domain username for Insights Reports.
INSIGHTS_DOMAIN_PASSWORD	The domain password for Insights Reports.
PRODUCT_OPTS_INSIGHTS	Product specific options for Insights Reports.

Table 28. Application server-related Insights Reports parameters

Parameter name	Parameter description
INSIGHTS_HOST_NAME	The system host name of Insights Reports.
INSIGHTS_MANAGEMENT_PORT	The management port number for the Insights Reports system.
INSIGHTS_MANAGEMENT_HTTPS_PORT	The management HTTPS port number for the Insights Reports system.
INSIGHTS_AJP_PORT	The AJP port number for the Insights Reports system.
INSIGHTS_HTTP_PORT	The HTTP port number for the Insights Reports system.
INSIGHTS_HTTPS_PORT	The HTTPS port number for the Insights Reports system.
INSIGHTS_RECOVERY_ENV_PORT	The recovery environment port number of the Insights Reports system.
INSIGHTS_STATUS_MANAGER_PORT	The status manager port number of the Insights Reports system.
INSIGHTS_MIN_HEAP	The minimum heap size allocated for Insights Reports.
INSIGHTS_MAX_HEAP	The maximum heap size allocated for Insights Reports.

Interact configurations

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

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

Table 29. 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, 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.

Table 29. Common parameters for Interact (continued)

Parameter name	Parameter description
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.
INTERACT_PRODUCT_NAME	The name assigned for Interact.
INTERACT_WAR_NAME	The name of the WAR file.
INTERACT_APPLICATION_NAME	The name of the main application. For example, Unica.
INTERACT_DOMAIN_USERNAME	The domain username for Interact.
INTERACT_DOMAIN_PASSWORD	The domain password for Interact.

Table 30. Platform server-related parameters of Interact

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.

Table 30. Platform server-related parameters of Interact (continued)

Parameter name	Parameter description
INTERACTATM_PLATFORM_DATABASE_USER-NAME	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 exceed the con-

Table 30. Platform server-related parameters of Interact (continued)

Parameter name	Parameter description
	figured 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 31. 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.

Table 31. Server group-related database parameters of Interact (continued)

Parameter name	Parameter description
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 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.

Table 31. Server group-related database parameters of Interact (continued)

Parameter name	Parameter description
INTERACTWEB_DATABASE_NAME	Name of the Interact Web database.
INTERACTWEB_DS_INITIAL_SIZE	The initial size of the Interact Web data-source connection pool.
INTERACTWEB_DS_MIN_IDLE	The minimum number of idle connections (not connected to a database) in the Interact Web datasource connection pool.
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 data-source. 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.

Table 31. Server group-related database parameters of Interact (continued)

Parameter name	Parameter description
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.
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 per-

Table 31. Server group-related database parameters of Interact (continued)

Parameter name	Parameter description
	formance by caching executable statements that are used repeatedly.

Table 32. 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.
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

Table 32. Server-related database parameters of Interact (continued)

Parameter name	Parameter description
	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.
INTERACT_PROD_DSN_NAME	The dbname of the respective database.
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.

Table 32. Server-related database parameters of Interact (continued)

Parameter name	Parameter description
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 by caching executable statements that are used repeatedly.
INTERACT_TEST_DSN_NAME	The <code>dbname</code> of the respective database.
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.

Table 32. Server-related database parameters of Interact (continued)

Parameter name	Parameter description
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 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.

Table 32. Server-related database parameters of Interact (continued)

Parameter name	Parameter description
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.
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 data-

Table 32. Server-related database parameters of Interact (continued)

Parameter name	Parameter description
	source. 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 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 ex-

Table 32. Server-related database parameters of Interact (continued)

Parameter name	Parameter description
	ceed 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 data-source. Statement caching improves performance by caching executable statements that are used repeatedly.

Table 33. 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.
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.

Table 33. Application server-related parameters of Interact (continued)

Parameter name	Parameter description
INT_MIN_HEAP	The maximum heap size allocated for Interact.
INT_MAX_HEAP	The maximum heap size allocated for Interact.

Journey configurations

To configure the Journey server for Cloud Native Unica, make the necessary modifications to the `journey-configMap.yaml` file.

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

Table 34. Parameters of Journey

Parameter name	Parameter description
JOURNEY_HOST_NAME	The system host name of Journey.
JOURNEY_MANAGEMENT_PORT	The management port number for the Journey system.
JOURNEY_MANAGEMENT_HTTPS_PORT	The management <code>HTTPS</code> port number for the Journey system.
JOURNEY_AJP_PORT	The <code>AJP</code> port number for the Journey system.
JOURNEY_HTTP_PORT	The <code>HTTP</code> port number for the Journey system.
JOURNEY_HTTPS_PORT	The <code>HTTPS</code> port number for the Journey system.
JOURNEY_RECOVERY_ENV_PORT	The recovery environment port number of the Journey system.

Table 34. Parameters of Journey (continued)

Parameter name	Parameter description
JOURNEY_STATUS_MANAGER_PORT	The status manager port number of the Journey system.
JOURNEY_MIN_HEAP	The maximum heap size allocated for Journey. For example, 1024m.
JOURNEY_MAX_HEAP	The maximum heap size allocated for Journey. For example, 6614m.
DB_TYPE_JOURNEY	The name of the database used by the Journey system. For example, Oracle.
DB_DRIVER_CLASS_JOURNEY	The class name of the Journey Database drivers. For example <code>oracle.jdbc.OracleDriver</code> .
JOURNEYREPORT_DB_NAME	The database name of the server hosting the Journey Reports. For example, <code>journeyuser</code> .

Journey web configurations

To configure the Journey web server for Journey, make the necessary modifications to the `journeyweb-configMap.yaml` file.

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

Table 35. Common parameters of Journey web configuration

Parameter name	Parameter description
JOURNEYWEB_JNDI_NAME	JNDI name for Journey web.
JOURNEYWEB_POOL_NAME	Pool name for Journey web.
PRODUCT_OPTS_PLATFORM	Product-specific options for Journey web.

Table 35. Common parameters of Journey web configuration (continued)

Parameter name	Parameter description
JOURNEYWEB_PRODUCT_NAME	The name assigned for Journey web.
JOURNEYWEB_WAR_NAME	The name of the <code>WAR</code> file.
JOURNEYWEB_APPLICATION_NAME	The name of the main application. For example, Unica.
JOURNEYWEB_DOMAIN_USERNAME	The domain username for Journey web.
JOURNEYWEB_DOMAIN_PASSWORD	The domain password for Journey web.

Table 36. Database parameters of Journey web configuration

Parameter name	Parameter description
JOURNEYWEB_DATABASE_HOST	Host system details of the system hosting the Journey web database.
JOURNEYWEB_DATABASE_PORT	Port number of the Journey web database.
JOURNEYWEB_DATABASE_USERNAME	Username to access the Journey web database.
JOURNEYWEB_DATABASE_PASSWORD	Password to access the Journey web database.
JOURNEYWEB_DATABASE_NAME	Name of the Journey web database.
JOURNEYWEB_DS_INITIAL_SIZE	The initial size of the Journey web datasource connection pool.

Table 36. Database parameters of Journey web configuration (continued)

Parameter name	Parameter description
JOURNEYWEB_DS_MIN_IDLE	The minimum number of idle connections (not connected to a database) in the Journey web datasource connection pool.
JOURNEYWEB_DS_MAX_IDLE	The maximum number of idle connections (not connected to a database) in the Journey web datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool.
JOURNEYWEB_DS_MAX_TOTAL	The maximum number of connections that the Journey web datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused.
JOURNEYWEB_DS_STATEMENT_CACHE_SIZE	Maximum number of statements that can be cached in the Journey web datasource. Statement caching improves performance by caching executable statements that are used repeatedly.

Table 37. Application server parameters of Journey web configuration

Parameter name	Parameter description
JOURNEYWEB_URL	The URL to access Journey web.
JOURNEYWEB_HOST_NAME	The system host name of Journey web.
JOURNEYWEB_MANAGEMENT_PORT	The management port number for the Journey web system.
JOURNEYWEB_MANAGEMENT_HTTPS_PORT	The management HTTPS port number for the Journey web system.
JOURNEYWEB_AJP_PORT	The AJP port number for the Journey web system.
JOURNEYWEB_HTTP_PORT	The HTTP port number for the Journey web system.

Table 37. Application server parameters of Journey web configuration (continued)

Parameter name	Parameter description
JOURNEYWEB_HTTPS_PORT	The HTTPS port number for the Journey web system.
JOURNEYWEB_RECOVERY_- ENV_PORT	The recovery environment port number of the Journey web system.
JOURNEYWEB_STATUS_- MANAGER_PORT	The status manager port number of the Journey web system.
JOURNEYWEB_MIN_HEAP	The maximum heap size allocated for Journey web.
JOURNEYWEB_MAX_HEAP	The maximum heap size allocated for Journey web.

Table 38. Other parameters of Journey web configuration

Parameter name	Parameter description
JOURNEYWEB_IP_FINDER_- LIST	
JOURNEYWEB_MULTI- CAST_GROUP	
JOURNEYWEB_MULTI- CAST_PORT	
JOURNEYWEB_MULTI- CASE_ENABLED	
JOURNEYWEB_DEFAULT_- DATA_REGION_MAX_SIZE	
JOURNEYWEB_GOAL_MAX_- SIZE_ALLOWED	

Table 39. Configuration of Journey report parameters

Parameter name	Parameter description
JOURNEYREPORT_DATA-BASE_NAME	Name of the Journey report database.
JOURNEY_REPORT_DATA-BASE_USERNAME	Username to access the Journey report database.
JOURNEY_REPORT_DATA-BASE_PASSWORD	Password to access the Journey report database.
JOURNEYREPORT_DS-INITIAL_SIZE	The initial size of the Journey report datasource connection pool.
JOURNEYREPORT_DS_MIN-IDLE	The minimum number of idle connections (not connected to a database) in the Journey report datasource connection pool.
JOURNEYREPORT_DS-MAX_IDLE	The maximum number of idle connections (not connected to a database) in the Journey report datasource connection pool. Any idle connections, which exceeds the configured value, will be removed from the pool.
JOURNEYREPORT_DS-MAX_TOTAL	The maximum number of connections that the Journey report datasource can hold. If the number of connection requests exceed the configured value, the connection will be refused.
JOURNEYREPORT_DS-STATEMENT_CACHE_SIZE	Maximum number of statements that can be cached in the Journey report datasource. Statement caching improves performance by caching executable statements that are used repeatedly.
JOURNEYREPORT_JNDI-NAME	JNDI name for Journey report.

Table 39. Configuration of Journey report parameters (continued)

Parameter name	Parameter description
JOURNEYREPORT_POOL_NAME	Pool name for Journey report.
JOURNEYREPORT_DB_NAME	The database name of the server hosting the Journey Reports. For example, <code>journeyuser</code> .

Kafka configurations

To configure the Kafka server for Journey, make the necessary modifications to the `kafka-configMap.yaml` file.

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

Table 40. Database-parameters of Kafka configuration

Parameter name	Parameter description
JOURNEY_DATABASE_HOST	Host system details of the system hosting the Journey database.
JOURNEY_DATABASE_PORT	Port number of the Journey database.
JOURNEY_DATABASE_USERNAME	Username to access the Journey database.
JOURNEY_DATABASE_PASSWORD	Password to access the Journey database.
JOURNEY_DATABASE_NAME	Name of the Journey database.

Table 41. Common parameters of Kafka configuration

Parameter name	Parameter description
KAFKA_SERVER	The details of the system hosting the Kafka server.

Table 41. Common parameters of Kafka configuration (continued)

Parameter name	Parameter description
KAFKA_HOST_NAME	The host name of the Kafka server.
KAFKA_PORT	The port number to access the Kafka server.
JOURNEY_HOST_NAME	The host name of the Journey server.
JOURNEY_PORT	The port number to access the Journey server.
ZOOKEEPER_PORT	

Plan configurations

To configure Plan for Cloud Native Unica, make the necessary modifications to the **plan-configMap.yaml** file.

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

Table 42. 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.
PLAN_PRODUCT_NAME	The name assigned for Plan.
PLAN_WAR_NAME	The name of the WAR file.

Table 42. Common parameters of Plan (continued)

Parameter name	Parameter description
PLAN_APPLICATION_NAME	The name of the main application. For example, Unica.
PLAN_DOMAIN_USERNAME	The domain username for Plan.
PLAN_DOMAIN_PASSWORD	The domain password for Plan.

Table 43. 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.
PLAN_RECOVERY_ENV_PORT	The recovery environment port number of the Plan system.

Table 43. Application server-related parameters of Plan (continued)

Parameter name	Parameter description
PLAN_STATUS_MANAGER_PORT	The status manager port number of the Plan system.
PLAN_URL	The URL to access Plan.

Table 44. 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	The database name for Plan.
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 host-

Table 44. Database-related parameters for Plan (continued)

Parameter name	Parameter description
	ing the Plan database.
PLAN_DATABASE_PORT	Port number of the Plan database.
PLAN_DATABASE_NAME	Name of the Plan database.
PLAN_DATABASE_USERNAME	Plan
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 config-

Table 44. Database-related parameters for Plan (continued)

Parameter name	Parameter description
	ured value, will be removed from the pool.
PLAN_DS_MAX_TOTAL	The maximum number of connections that the Plan data-source 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 datasource. Statement caching improves performance by caching executable statements that are used repeatedly.

Platform configurations

To configure Platform for Cloud Native Unica, 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 45. 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	Unica
PRODUCT_OPTS_PLATFORM	Product specific options for Platform.
FORCE_INIT_WEBLOGIC	Set whether you want to force initialize WebLogic. TRUE to activate for initialization and FALSE to deactivate force initialization.
JAVA_HOME_WEBLOGIC	Location of Java Home on your system.
PLATFORM_PRODUCT_NAME	The name assigned for Platform.
PLATFORM_WAR_NAME	The name of the WAR file.
PLATFORM_APPLICATION_NAME	The name of the main application. For example, Unica.
PLATFORM_DOMAIN_USERNAME	The domain username for Platform.

Table 45. Common parameters of Platform (continued)

Parameter name	Parameter description
PLATFORM_DOMAIN_PASSWORD	The domain password for Platform.
REPLACE_ADMIN_USR_NAME	
REPLACE_ADMIN_USR_PASSWORD	

Table 46. 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.
DB_PLAT	The database name for Platform.
PLATFORM_DS_INITIAL_SIZE	The initial size of the Platform datasource connection pool.
PLATFORM_DS_MIN_IDLE	The minimum number of idle connections (not con-

Table 46. Database-related parameters of Platform (continued)

Parameter name	Parameter description
	nected 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 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 47. Application server-related parameters of Platform

Parameter name	Parameter description
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.
PLAT_MIN_HEAP	The minimum heap size allocated for Platform.
PLAT_MAX_HEAP	The maximum heap size allocated for Platform.

Table 48. Apache Tomcat-specific parameters

Parameter name	Parameter description
TOMCAT_INSTALLER_TARGZ	The name of the Apache Tomcat installer TARGZ file.
TOMCAT_INSTALLER_UNZIP_DIR-NAME	The location to extract the Apache Tomcat installer TARGZ file.
TOMCAT_INSTALL_LOCATION	The location to install Apache Tomcat.
FORCE_INIT_TOMCAT	Set whether you want to force initialize Apache Tomcat. TRUE to activate for initialization and FALSE to deactivate force initialization.
TOMCAT_SHUTDOWN_PORT	The TCP/IP port number of the Apache Tomcat server waiting for a shutdown command.
TOMCAT_MAX_EXECUTOR_THREADS	The maximum number of threads (based on the maxThreads property of Apache Tomcat) used for HTTP connections.
TOMCAT_MIN_EXECUTOR_THREADS	The minimum number of threads (based on the minSpareThreads property of Apache Tomcat) that is always present in the thread pool.
TOMCAT_REDIRECT_PORT	The redirect port number (redirectPort property) of the Apache Tomcat server handling SSL connections.

Segment Central configurations

To configure Segment Central for Cloud Native Unica, make the necessary modifications to the `segmentcentral-configMap.yaml` file.

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

Table 49. Common parameters for Segment Central

Parameter name	Parameter description
SEGMENTCENTRAL_PRODUCT_NAME	Segmentcentral
SEGMENT_CENTRAL_WAR_NAME	SegmentCentral.war
SEGMENTCENTRAL_APPLICATION_NAME	segmentcentral
SEGMENTCENTRAL_DOMAIN_USERNAME	root
SEGMENTCENTRAL_DOMAIN_PASSWORD	unica*03
SEGMENTATIONENGINE_PRODUCT_NAME	Segmentationengine
SEGMENTATION_ENGINE_WAR_NAME	SegmentationEngine.war
SEGMENTATIONENGINE_APPLICATION_NAME	segmentationengine
SEGMENTATIONENGINE_DOMAIN_USERNAME	root
SEGMENTATIONENGINE_DOMAIN_PASSWORD	unica*03

Table 50. Application Server-related parameters for Segment Central

Parameter name	Parameter description
SEGMENTCENTRAL_HOST_NAME	{{ .Release.Name }}-unica-segment-central

Table 50. Application Server-related parameters for Segment Central (continued)

Parameter name	Parameter description
SEGMENTCENTRAL_MANAGEMENT_PORT	9066
SEGMENTCENTRAL_MANAGEMENT_HTTPS_PORT	9995
SEGMENTCENTRAL_AJP_PORT	8010
SEGMENTCENTRAL_HTTP_PORT	9140
SEGMENTCENTRAL_HTTPS_PORT	9446
SEGMENTCENTRAL_RECOVERY_ENV_PORT	4714
SEGMENTCENTRAL_STATUS_MANAGER_PORT	4715
SEGMENTCENTRAL_MIN_HEAP	1024m
SEGMENTCENTRAL_MAX_HEAP	2048m
SEGMENTCENTRAL_URL	<code>{{ include ip.protocol . }}://{{ .Values.service.hostname }}/SegmentCentral</code>
SEGMENTCENTRAL_INTERNAL_URL	<code>http://{{ .Release.Name }}-unica-segmentcentral:9140/SegmentCentral</code>
PRODUCT_OPTS_SEGMENTCENTRAL	<code>-DSEGMENT_CENTRAL_HOME=/docker/unica/SegmentCentral/ -DENABLE_NON_PROD_MODE=true</code>

Table 51. Application Server-related parameters for Segment Central Engine

Parameter name	Parameter description
SEGMENTATIONENGINE_HOST_NAME	{{ .Release.Name }}-unica-segmentationengine
SEGMENTATIONENGINE_MANAGEMENT_PORT	9067
SEGMENTATIONENGINE_MANAGEMENT_HTTPS_PORT	9996
SEGMENTATIONENGINE_AJP_PORT	8011
SEGMENTATIONENGINE_HTTP_PORT	9141
SEGMENTATIONENGINE_HTTPS_PORT	9447
SEGMENTATIONENGINE_RECOVERY_ENV_PORT	4715
SEGMENTATIONENGINE_STATUS_MANAGER_PORT	4716
SEGMENTATIONENGINE_MIN_HEAP	1024m
SEGMENTATIONENGINE_MAX_HEAP	2048m
SEGMENTATIONENGINE_INTERNAL_URL	http://{{ .Release.Name }}-unica-segmentationengine:9141/SegmentationEngine
PRODUCT_OPTS_SEGMENTATIONENGINE	-DSEGMENT_CENTRAL_HOME=/docker/unica/SegmentCentral/ -DENABLE_NON_PROD_MODE=true

Table 52. Database-related parameters for Segment Central

Parameter name	Parameter description
SEGMENTCENTRAL_USER_JNDI_NAME	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_JNDI_NAME }}
SEGMENTCENTRAL_USER_POOL_NAME	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_POOL_NAME }}
SEGMENTCENTRAL_USER_DATABASE_HOST	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_DATABASE_HOST }}
SEGMENTCENTRAL_USER_DATABASE_PORT	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_DATABASE_PORT }}
SEGMENTCENTRAL_USER_DATABASE_NAME	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_DATABASE_NAME }}
SEGMENTCENTRAL_USER_DATABASE_USERNAME	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_DATABASE_USERNAME }}
SEGMENTCENTRAL_USER_DATABASE_PASSWORD	{{ .Values.segmentcentralData.segmentcentralConfigMapData.SEGMENTCENTRAL_USER_DATABASE_PASSWORD }}
SEGMENTCENTRAL_USER_DS_INITIAL_SIZE	{{ .Values.segmentcentralData.segmentcentralDSMData.SEGMENTCENTRAL_USER_DS_INITIAL_SIZE }}

Table 52. Database-related parameters for Segment Central (continued)

Parameter name	Parameter description
SEGMENTCENTRAL_USER_DS_MIN_IDLE	{{ .Values.segmentcentralData.segmentcentralDSMData.SEGMENTCENTRAL_USER_DS_MIN_IDLE }}
SEGMENTCENTRAL_USER_DS_MAX_IDLE	{{ .Values.segmentcentralData.segmentcentralDSMData.SEGMENTCENTRAL_USER_DS_MAX_IDLE }}
SEGMENTCENTRAL_USER_DS_MAX_TOTAL	{{ .Values.segmentcentralData.segmentcentralDSMData.SEGMENTCENTRAL_USER_DS_MAX_TOTAL }}
SEGMENTCENTRAL_USER_DS_STATEMENT_CACHE_SIZE	{{ .Values.segmentcentralData.segmentcentralDSMData.SEGMENTCENTRAL_USER_DS_STATEMENT_CACHE_SIZE }}
SEGMENTCENTRAL_USER_DATA_SOURCE_PARAMETERS	{{ .Values.segmentcentralData.segmentcentralDSMData.SEGMENTCENTRAL_USER_DATA_SOURCE_PARAMETERS }}

Table 53. Database-related parameters for Segment Central Engine

Parameter name	Parameter description
SEGMENTATIONENGINE_USER_POOL_NAME	{{ .Values.segmentationengineData.segmentationengineConfigMapData.SEGMENTATIONENGINE_USER_POOL_NAME }}
SEGMENTATIONENGINE_USER_DATABASE_HOST	{{ .Values.segmentationengineData.segmentationengineConfigMapData.SEGMENTATIONENGINE_USER_DATABASE_HOST }}

Table 53. Database-related parameters for Segment Central Engine (continued)

Parameter name	Parameter description
SEGMENTATIONENGINE_USER_DATA-BASE_PORT	{{ .Values.segmentationengineData-.segmentationengineConfigMapData-.SEGMENTATIONENGINE_USER_DATABASE_-PORT }}
SEGMENTATIONENGINE_USER_DATA-BASE_NAME	{{ .Values.segmentationengineData-.segmentationengineConfigMapData-.SEGMENTATIONENGINE_USER_DATABASE_-NAME }}
SEGMENTATIONENGINE_USER_DATA-BASE_USERNAME	{{ .Values.segmentationengineData-.segmentationengineConfigMapData-.SEGMENTATIONENGINE_USER_DATABASE_-USERNAME }}
SEGMENTATIONENGINE_USER_DATA-BASE_PASSWORD	{{ .Values.segmentationengineData-.segmentationengineConfigMapData-.SEGMENTATIONENGINE_USER_DATABASE_-PASSWORD }}
SEGMENTATIONENGINE_USER_DS-INITIAL_SIZE	{{ .Values.segmentationengineData-.segmentationengineDSMData.SEGMENTATIONENGINE_USER_DS_INITIAL_SIZE }}
SEGMENTATIONENGINE_USER_DS-MIN_IDLE	{{ .Values.segmentationengineData-.segmentationengineDSMData.SEGMENTATIONENGINE_USER_DS_MIN_IDLE }}
SEGMENTATIONENGINE_USER_DS-MAX_IDLE	{{ .Values.segmentationengineData-.segmentationengineDSMData.SEGMENTATIONENGINE_USER_DS_MAX_IDLE }}

Table 53. Database-related parameters for Segment Central Engine (continued)

Parameter name	Parameter description
SEGMENTATIONENGINE_USER_DS_MAX_TOTAL	{{ .Values.segmentationengineData.segmentationengineDSMData.SEGMENTATIONENGINE_USER_DS_MAX_TOTAL }}
SEGMENTATIONENGINE_USER_DS_STATEMENT_CACHE_SIZE	{{ .Values.segmentationengineData.segmentationengineDSMData.SEGMENTATIONENGINE_USER_DS_STATEMENT_CACHE_SIZE }}
SEGMENTATIONENGINE_USER_DATA_SOURCE_PARAMETERS	{{ .Values.segmentationengineData.segmentationengineDSMData.SEGMENTATIONENGINE_USER_DATA_SOURCE_PARAMETERS }}

Sub-chart configuration in Helm charts

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



Note: Cloud Native Unica does not own the database.

values.yaml driven configurations

1. Database and Data source memory related parameters can be configured in `values.yaml` and Product `configmap.yaml` files as shown in the following examples:

- **Values file example:**

```
campaignData:
  campaignConfigMapData:
    CAMPAIGN_DATABASE_HOST: "hcl-unica-suite-database"
```

```
CAMPAIGN_DATABASE_PORT: "9088"
CAMPAIGN_DATABASE_NAME: "campuser:ONEDB_SERVER=onedb"
CAMPAIGN_DATABASE_USERNAME: "onedbsa"
CAMPAIGN_DATABASE_PASSWORD: "onedb4ever"
CAMPAIGN_DSN_NAME: "campuser"
campaignDSMData:
  CAMPAIGN_DS_INITIAL_SIZE: "1"
  CAMPAIGN_DS_MIN_IDLE: "1"
  CAMPAIGN_DS_MAX_IDLE: "1"
  CAMPAIGN_DS_MAX_TOTAL: "80"
  CAMPAIGN_DS_STATEMENT_CACHE_SIZE: "180"
  CAMPAIGN_DATA_SOURCE_PARAMETERS:
    "removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'"
```

- **Reflected in Campaign configmap file:**

```
CAMPAIGN_DATABASE_HOST:
  "{{ .Values.campaignData.campaignConfigMapData.CAMPAIGN_DATABASE_HOST }}"
CAMPAIGN_DATABASE_PORT:
  "{{ .Values.campaignData.campaignConfigMapData.CAMPAIGN_DATABASE_PORT }}"
CAMPAIGN_DATABASE_NAME:
  "{{ .Values.campaignData.campaignConfigMapData.CAMPAIGN_DATABASE_NAME }}"
CAMPAIGN_DATABASE_USERNAME:
  "{{ .Values.campaignData.campaignConfigMapData.CAMPAIGN_DATABASE_USERNAME }}"
CAMPAIGN_DATABASE_PASSWORD:
  "{{ .Values.campaignData.campaignConfigMapData.CAMPAIGN_DATABASE_PASSWORD }}"
```

```

    CAMPAIGN_DSN_NAME:
    "{{ .Values.campaignData.campaignConfigMapData.CAMPAIGN_DSN_NAME
    }}"

    CAMPAIGN_DS_INITIAL_SIZE:
    "{{ .Values.campaignData.campaignDSMData.CAMPAIGN_DS_INITIAL_SIZ
    E }}"

    CAMPAIGN_DS_MIN_IDLE:
    "{{ .Values.campaignData.campaignDSMData.CAMPAIGN_DS_MIN_IDLE }}"

    CAMPAIGN_DS_MAX_IDLE:
    "{{ .Values.campaignData.campaignDSMData.CAMPAIGN_DS_MAX_IDLE }}"

    CAMPAIGN_DS_MAX_TOTAL:
    "{{ .Values.campaignData.campaignDSMData.CAMPAIGN_DS_MAX_TOTAL }}"

    CAMPAIGN_DS_STATEMENT_CACHE_SIZE:
    "{{ .Values.campaignData.campaignDSMData.CAMPAIGN_DS_STATEMENT_C
    ACHESIZE }}"

```

2. Parameters categorization: Each product is separately defined by a tag in the **values.yaml** file and subtag is provided for Data type: database or data source memory.

- **Example: Data type categorisation (Database and Data source memory)**

```

campaignData:
  campaignConfigMapData:
    CAMPAIGN_DATABASE_HOST: "hcl-unica-suite-database"
    CAMPAIGN_DATABASE_PORT: "9088"
    CAMPAIGN_DATABASE_NAME: "campuser:ONEDB_SERVER=onedb"
    CAMPAIGN_DATABASE_USERNAME: "onedbsa"
    CAMPAIGN_DATABASE_PASSWORD: "onedb4ever"
    CAMPAIGN_DSN_NAME: "campuser"

```



```
campaignDSMData:
  CAMPAIGN_DS_INITIAL_SIZE: "1"
  CAMPAIGN_DS_MIN_IDLE: "1"
  CAMPAIGN_DS_MAX_IDLE: "1"
  CAMPAIGN_DS_MAX_TOTAL: "80"
  CAMPAIGN_DS_STATEMENT_CACHE_SIZE: "180"
  CAMPAIGN_DATA_SOURCE_PARAMETERS:
    "removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'"
```

- **Example: Product wise categorisation**

```
collaborateData:
  collaborateConfigMapData:
    COLLABORATE_DATABASE_HOST: "hcl-unica-suite-database"
    COLLABORATE_DATABASE_PORT: "9088"
    COLLABORATE_DATABASE_NAME: "platuser"
    COLLABORATE_DATABASE_USERNAME: "informix"
    COLLABORATE_DATABASE_PASSWORD: "in4mix"

  collaboratedDSMData:
    COLLABORATE_DS_INITIAL_SIZE: "1"
    COLLABORATE_DS_MIN_IDLE: "1"
    COLLABORATE_DS_MAX_IDLE: "1"
    COLLABORATE_DS_MAX_TOTAL: "80"
    COLLABORATE_DS_STATEMENT_CACHE_SIZE: "180"
    COLLABORATE_DATA_SOURCE_PARAMETERS:
      "removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'"

  contactcentralData:
    contactcentralConfigMapData:
      CONTACTCENTRAL_DATABASE_HOST: "hcl-unica-suite-database"
      CONTACTCENTRAL_DATABASE_PORT: "9088"
```

```
CONTACTCENTRAL_DATABASE_USERNAME: "onedbsa"
CONTACTCENTRAL_DATABASE_PASSWORD: "onedb4ever"
CONTACTCENTRAL_DATABASE_NAME: "platuser:ONEDB_SERVER=onedb"
contactcentralDSMData:
  CONTACTCENTRAL_DS_INITIAL_SIZE: "1"
  CONTACTCENTRAL_DS_MIN_IDLE: "1"
  CONTACTCENTRAL_DS_MAX_IDLE: "1"
  CONTACTCENTRAL_DS_MAX_TOTAL: "80"
  CONTACTCENTRAL_DS_STATEMENT_CACHE_SIZE: "180"
  CONTACTCENTRAL_DATA_SOURCE_PARAMETERS:
    "removeAbandonedTimeout='300';removeAbandoned='true';testOnBorrow='true'"
```

3. JAVA_HOME is set in `values.yaml`.

- JAVA_HOME, DOCKER_HOME, JRE_HOME, DIRECTOR_JAVA_HOME parameters have same value assigned in common configmap files.

```
DOCKER_JAVA_HOME: "{{
  .Values.commonConfigMapMiscData.JAVA_HOME }}"
DIRECTOR_JAVA_HOME:
  "{{ .Values.commonConfigMapMiscData.JAVA_HOME }}"
JRE_HOME: "{{ .Values.commonConfigMapMiscData.JAVA_HOME }}"
JavaHOME: "{{ .Values.commonConfigMapMiscData.JAVA_HOME }}"
```

- JAVA_HOME: `docker/unica/jre` is defined in values file and used in above `common_configmap.yaml` file.

4. Ingress updates with version 1.22:

- For k8s 1.22 version API updates have been made to RBAC and INGRESS files also ingress structure is changed as per K8S update.
- **Example old ingress:**

```
rules:
  - host: {{ .Values.service.hostname }}
    http:
      paths:
```

```
- path: /Insights
  backend:
    serviceName: {{ include "unica.fullname" . }}-insights
    servicePort: {{ .Values.service.port.insights }}
```

• **Example new ingress:**

```
rules:
- host: {{ .Values.service.hostname }}
  http:
    paths:
    - path: /Insights
      pathType: Prefix
      backend:
        service:
          name: {{ include "unica.fullname" . }}-insights
          port:
            number: {{ .Values.service.port.insights }}
```