Oracle® Application Integration Architecture

Cloud Native Deployment Guide

Release 12.3.3
G21083-01
May 2025

ORACLE

Oracle Application Integration Architecture Cloud Native Deployment Guide, Release 12.3.3

G21083-01

Contents

Preface

This document describes how to install and administer Oracle Application Integration Architecture Cloud Native Deployment.

Audience

This document is intended for DevOps administrators and those involved in installing and maintaining Oracle Application Integration Architecture Cloud Native Deployment.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

Overview of the AIA Cloud Native Deployment

This chapter provides an overview of Oracle Communications Application Integration Architecture (AIA) deployed in a cloud native environment using container images and a Kubernetes cluster.

About AIA and the AIA Cloud Native Deployment

AIA for Communications is an Oracle Communications Solutions Integration framework that includes pre-built integrations using standard integration patterns, business processes, orchestration logic, and common objectives and services that enable seamless interaction with Oracle Applications.

The packaged integrations provide business and functional flows that map to key business processes in the domain of operations support system and business support systems for a communication service provider.

AIA for Communications includes the following set of pre-built and packaged integrations that can be either licensed as a complete suite or selectively based on your requirements:

- Order to Cash for Siebel CRM Cloud Native
- Order to Cash for Oracle Communication Order and Service Management
- Order to Cash for Oracle Communications Billing and Revenue Management

Overview of AIA for Communications Cloud Native Deployment

AIA in a cloud native architecture increases operational efficiency by improving hardware utilization and scaling real-time business events to capture revenue.

The AIA cloud native deployment option combines the features and extensibility of AIA with the agility and efficiency of cloud infrastructure with DevOps aligned.

The key features of AIA cloud native are:

- Container images (Docker, CRI-O), orchestrated in Kubernetes - production support for AIA-DBE Deployment on Kubernetes
- Install and manage using Helm charts
- Docker files and scripts for development and testing
- Lifecycle management using WebLogic Kubernetes Operator

The integrations interface across the following applications to automate order-to-cash-to care business processes:

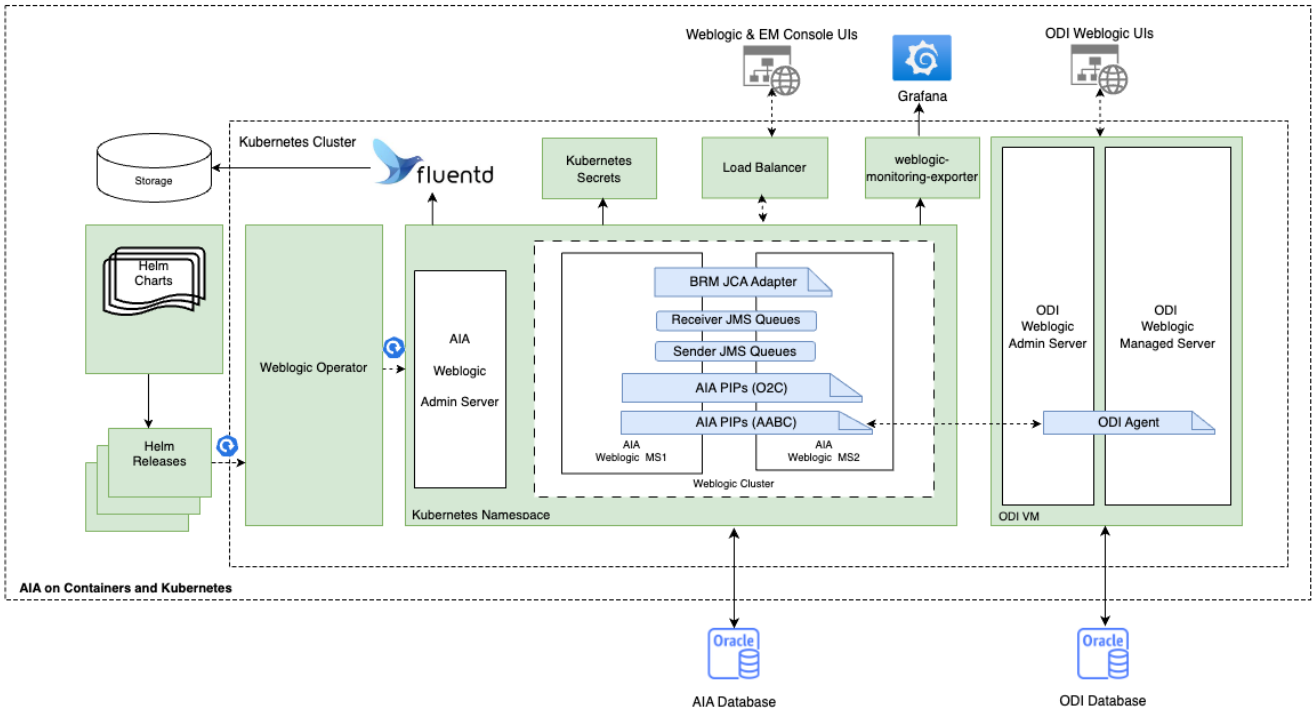
- Siebel CRM Cloud Native
- Oracle Communications Order and Service Management (OSM) Cloud Native
- Oracle Communications Billing and Revenue Management (BRM) Cloud Native

See *AIA Compatibility Matrix* for information about the recommended and supported versions of applications.

AIA Cloud Native Architecture

The following diagram illustrates AIA on containers and Kubernetes.

Figure 1-1 AIA Cloud Native Deployment Architecture



About the AIA Cloud Native Toolkit

The AIA cloud native toolkit is an archive file that contains the default configuration files, scripts for generating an AIA image and for deploying AIA in a cloud native environment.

The toolkit contains the following artifacts:

- Helm charts and configuration override values yaml files for configuring and managing AIA Order-to-Cash PIP deployment and SOA.
- Scripts for creating and managing secrets for AIA's edge systems, XRef, SOA and RCU.
- Scripts for generating AIA images.
- Scripts for restarting managed servers.
- Scripts for deploying AIA in a cloud native environment.
- Scripts for managing the life cycle of an AIA cloud native instance.

Table 1-1 AIA Cloud Native Toolkit Artifacts

Artifact	Artifact Type	Description
AIA Docker Image	Build	This is the AIA image that you build using the scripts and the Dockerfile provided with the toolkit. **Note**: Dockerfile is the docker image configuration file to be passed into the docker build command.
SOA Cloud Native Docker Image	Build	This is the SOA cloud native CPU image that is later than January 2025 that you pull before building the AIA image.
Secrets	Deployment	These are Kubernetes secrets used to create credentials and SSL certificates for systems.
Helm charts	Deployment	These are the charts used for other configuration parameters and sizing. The domain is static, as recommended by SOA. Hence, the maximum number of servers cannot be changed after the deployment, but they can be shut down and brought up.
Build scripts	Build	These are scripts used at build-time.
Deployment scripts	Deployment	These are scripts used to deploy the AIA artifacts, custom artifacts, and other artifacts.
PV/PVC	Deployment	This is the domain on Persistent Volume (PV) model (as recommended by SOA) that AIA cloud native uses. A persistent volume (PV) is a piece of storage in the Kubernetes cluster, while a persistent volume claim (PVC) is a request for storage.
Config Map	Deployment	This contains the deployable custom artifacts such as custom composites. The deployment job pods are configured to use these artifacts.
Scripts	Deployment	These are the scripts for restarting the managed servers (WebLogic Operator deployed) in the cloud native environment. You update and run the python script, which invokes WKO-based shell scripts to restart the servers.

About Helm Charts and Overrides

You use Helm charts for the following tasks:

- For managing Kubernetes
- For defining, updating, deploying, and managing versions
- For managing release history
- For customizing values and templates

The AIA cloud native toolkit contains the following charts:

- **aia-comms-pv-pvc**: This chart creates AIA PV/PVCs and service accounts.
- **aia-comms-deploy-aiapip**: This chart deploys AIA Foundation Pack and AIA Pre-integrated Packs (PIPs) in SOA and WebLogic servers.
- **aia-comms-certs**: This chart manages SSL trust certificates of Siebel and OSM in SOA and WebLogic servers.
- **aia-comms-update-mds**: This utility chart updates files in MDS.

- **aia-comms-install-brm-adapter**: This utility chart installs BRM adapter.
- **aia-comms-diagnostic**: This utility chart provides diagnostic report of SOA composite status and deployment.

The charts pick the customized values defined in the custom **values.yaml** configuration file and map them to the respective Kubernetes yaml files defined in the chart`s **/template** directory. Each chart can be installed and managed through helm commands.

For details about sizing, see "Oracle SOA cluster sizing recommendations".

About Helm Overrides

The specification files are consumed in a hierarchical fashion. If a value is found in multiple specification files (layers), the one further up the hierarchy takes precedence. This allows the instance specification to have the final control over its configuration by being able to override a value that is prescribed in either the shape or project specifications. This also allows Oracle to define sealed, base configuration, while still providing you the control over the values used for any specific AIA instance.

The main chart is **aia-comms-chart** within which there are multiple sub charts. The values in the **values.yaml** file are global and accessible to the sub charts.

The instance specification remains the final authority on any values that are found in multiple specification files.

2

Prerequisites for Your AIA Cloud Native Deployment

In preparation for the Oracle AIA cloud native deployment, you must set up and validate pre-requisite software. This chapter provides information about planning and setting up the environment for the AIA cloud native deployment. For details about the recommended and supported versions of required and supported software, see *AIA Compatibility Matrix*.

Downloading SOA Cloud Native

Download the WebLogic Kubernetes Operator and prepare and deploy Oracle SOA Suite domains.

For detailed instructions, see the installation guide for Oracle Fusion Middleware on Kubernetes for Oracle SOA Suite at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/.

Downloading and Installing the AIA Cloud Native Toolkit

To download and install the AIA cloud native toolkit:

1. Create an AIA Home directory:

   ```
   mkdir $HOME/AIA_Home
   ```

2. Run the following command:

   ```
   export AIA_DIR=$HOME/AIA_Home
   ```

3. Go to the Oracle software delivery website (https://edelivery.oracle.com/).
4. Search for and then download the **Oracle Communications Application Integration Architecture 12.3.3 Cloud Native Toolkit** .zip file to **$AIA_DIR** and extract it.
5. Navigate to the home directory and unzip the PIPs by running the following commands:

   ```
   $ cd $AIA_DIR/oc-cn-aia-comms
   $ unzip -j aiapip-12.3.3.0.0.zip comms_home_installer_generic.jar
   ```

Preparing Your Environment

Perform the following steps to prepare your environment for deploying AIA cloud native:

1. Set up the code repository to deploy Oracle SOA Suite domains by running the following commands:

> **Note:**
>
> For each Weblogic Kubernetes Operator (WKO) version, the branches for fmw scripts may change. Confirm the branch for the WKO version before running git clone. For WKO version 4.2.0, use branch `release/24.2.2` in the git clone command.

```
$ cd $AIA_DIR
$ git clone https://github.com/oracle/fmw-kubernetes.git -b release/24.2.2
$ export WORKDIR=$AIA_DIR/fmw-kubernetes/OracleSOASuite/kubernetes
```

Refer to the Oracle SOA Suite documentation available at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#set-up-the-code-repository-to-deploy-oracle-soa-suite-domains.

2. Update and save the **createSOADomain.py** Fusion Middleware script for jar details:

```
$ vi $WORKDIR/create-soa-domain/domain-home-on-pv/common/createSOADomain.py

## Make sure '@@ORACLE_HOME@@/soa/common/templates/wls/
oracle.soa.fp_template.jar' is added to SOA_TEMPLATES section ##
## Ensure to include ',' in the SOA_TEMPLATES ##
## Find below sample ##
    SOA_TEMPLATES = {
        'extensionTemplates' : [
            '@@ORACLE_HOME@@/soa/common/templates/wls/
oracle.soa.refconfig_template.jar',
            '@@ORACLE_HOME@@/oracle_common/common/templates/wls/
oracle.ess.basic_template.jar',
            '@@ORACLE_HOME@@/em/common/templates/wls/
oracle.em_ess_template.jar',
            '@@ORACLE_HOME@@/soa/common/templates/wls/
oracle.soa.fp_template.jar'
        ],
```

3
Creating AIA Cloud Native Images

This chapter describes how to create AIA cloud native images.

Prerequisites for Creating AIA Cloud Native Images

The prerequisites for creating AIA cloud native images are:

- Ensure that the AIA cloud native toolkit has been downloaded. For instructions, see "Downloading and Installing the AIA Cloud Native Toolkit".
- Ensure that Oracle SOA suite docker image is available. For details, see the documentation about Oracle Fusion Middleware on Kubernetes on Oracle GitHub at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#obtain-the-oracle-soa-suite-docker-image.

Configuring an AIA Cloud Native Image

To configure an AIA cloud native image:

1. Update the AIA dockerfile for SOA suite image tag:

```
$ cd $AIA_DIR/oc-cn-aia-comms
$ vi Dockerfile

# Update the base soasuite_cpu image tags as per your requirements and save
#
# Search for and update the following lines for image tags
#
# FROM container-registry.oracle.com/middleware/soasuite_cpu:tag as builder
#
# FROM container-registry.oracle.com/middleware/soasuite_cpu:tag
```

2. Update the domain lifecycle scripts:

> **Note:**
>
> By default, the AIA cloud native toolkit contains domain lifecycle scripts for WKO 4.0.4. This step is not required for WKO 4.0.4.

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/common
$ cp $WORKDIR/domain-lifecycle/helper.sh .
$ cp $WORKDIR/domain-lifecycle/startCluster.sh .
$ cp $WORKDIR/domain-lifecycle/stopCluster.sh .
$ cp $WORKDIR/domain-lifecycle/startServer.sh .
$ cp $WORKDIR/domain-lifecycle/stopServer.sh .
```

Creating an AIA Cloud Native Image

To create an AIA cloud native image, run the Docker build command using the updated Docker file:

```
$ cd $AIA_DIR/oc-cn-aia-comms
$ docker build --no cache -f Dockerfile -t aia-comms:12.3.3.0.0 .
```

Confirm that the AIA image is created and available:

```
$ docker images | grep aia-comms
```

This Dockerfile installs *kubectl* inside the AIA image at location */u01/oracle/*.

> **Note:**
>
> Update the *kubectl* version in the Dockerfile to the current version available at the time of deployment.

4

Deploying AIA Cloud Native

This chapter provides information about deploying AIA cloud native. AIA cloud native deployment leverages the SOA cloud native deployment scripts and uses Helm charts for configuration management.

The configuration details for AIA cloud native are the same as the configuration of standard deployments such as connection details, credentials, and XRef details of edge applications.

Along with AIA configuration details, SOA configuration details, such as RCU schema, domain parameters and other configurations are also managed by Helm charts. Helm charts are organized in a hierarchy and a master chart holds the references to several child charts for better management.

Deploying SOA Cloud Native

Deploy SOA cloud native as per the instructions provided in the Oracle SOA Suite documentation.

Deploying SOA cloud native has the following tasks:

- **Preparing the environment**
  For instructions, refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/.

  Perform all required steps and as per your requirement. For example, the following are some steps you perform based on your requirements:

  – Installing WebLogic Operator:

    1. Get dependent images. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#get-dependent-images.
    2. Install the WebLogic Kubernetes Operator. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#install-the-weblogic-kubernetes-operator.

  – Preparing the environment for Oracle SOA Suite domains:

    1. Create a namespace. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#create-a-namespace-for-an-oracle-soa-suite-domain.
    2. Create a persistent storage. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#create-a-persistent-storage-for-an-oracle-soa-suite-domain.
    3. Create a Kubernetes secret with domain credentials. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#create-a-kubernetes-secret-with-domain-credentials.
    4. Create a Kubernetes secret with the RCU credentials. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#create-a-kubernetes-secret-with-the-rcu-credentials.

5. Configure access to your database. Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#configure-access-to-your-database.
6. Run the Repository Creation Utility to set up your database schemas (create schemas).

> **Note:**
>
> Use the AIA cloud native image with the utility (command).

Refer to the information at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/prepare-your-environment/#run-the-repository-creation-utility-to-set-up-your-database-schemas.

- **Creating the SOA domain**

> **Note:**
>
> Use the AIA cloud native image for deploying the SOA domain. Also, ensure that you include the `-Dweblogic.rjvm.allowUnknownHost=true` parameter. For more information, refer to the WKO documentation on Oracle GitHub at: https://oracle.github.io/weblogic-kubernetes-operator/managing-domains/accessing-the-domain/external-clients/#enabling-unknown-host-access.

Refer to the instructions at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/installguide/create-soa-domains/.

- **Configuring a load balancer**
  Configure a load balancer as per your requirement. Refer to the instructions at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/adminguide/configure-load-balancer/.

Creating PV-PVCs

You can create AIA PV-PVCs by using the **values.yaml** files.

To create AIA PV-PVCs:

1. Update the following **values.yaml** files as per your requirement:
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml**. See "Global Parameters" for details about the parameters.
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-pv-pvc/values.yaml**. See "Parameters for AIA PV-PVC" for details about the parameters.
2. Run the following command to install the Helm chart:

```
$ cd $AIA_DIR/oc-cn-aia-comms/helm-charts/
$ helm install aia-comms-pv-pvc \
     aia-comms-chart/charts/aia-comms-pv-pvc/ \
     --namespace namespace \
     --values ./aia-comms-chart/values.yaml
```

This creates the following:

- Kubernetes PV PVC
- Kubernetes job **aia-comms-create-home-job** to populate AIA PV
- Service Account with name *domain_name*-cluster-kubectl
- ClusterRole of name *domain_name*-access-pod-cluster-role
- RoleBinding of name *domain_name*-access-pod-role-binding

3. Wait till the **aia-comms-create-home-job** job completes.

```
$ kubectl wait --for=condition=complete job.batch/aia-comms-create-home-
job -n namespace
```

Alternatively, this can also be confirmed by checking the status of the Kubernetes job or pod by running the following commands:

```
$ kubectl get pods -n namespace
$ kubectl get jobs -n namespace
```

Updating the Oracle SOA Suite Domain

To update the Oracle SOA Suite domain with the AIA PV:

1. Stop the Oracle SOA Suite domain:

```
$ cd $AIA_DIR/fmw-kubernetes/OracleSOASuite/kubernetes/domain-lifecycle/
$ sh stopDomain.sh -d domain -n namespace -v
```

(Optional. This does not add extra time.) To get more information on the shutdown process, you can run:

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/common
$ sh waitForCluster.sh -a shutdown -d domain -n namespace
```

Confirm all servers are down.

2. Do any one of the following to update the AIA PV with domain:
   - Run the following AIA cloud native scripts:

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts
# Remove any existing domain.yaml file from the folder
$ rm domain.yaml
$ sh register-AIAPV.sh \
    -n namespace \
    -d domain_name \
    -c cluster_name \
    -w SOA_PVC_name \
    -p AIA_PVC_name \
    -f $WORKDIR/create-soa-domain/domain-home-on-pv/path_to_output-
directory/weblogic-domains/domainUID/domain.yaml
```

This script does the following:

- Appends the volumes and volumeMounts with the AIA PVC details to the Oracle SOA Suite domain
- Runs the `kubectl apply` operation
- Waits till all the managed servers are available

- Update manually:

```
$ cd $WORKDIR/create-soa-domain/domain-home-on-pv/
$ vi path_to_output-directory/weblogic-domains/domainUID/domain.yaml
# Take a backup of YAML file before making any changes.
#
# Append volumes in volumes section
#    - name: aia-comms-shared-storage-volume
#      persistentVolumeClaim:
#        claimName: AIA_PVC_name

# Append volumeMounts in volumeMounts section
# Do not change the mount path here.
#    - mountPath: /u01/shared
#      name: aia-comms-shared-storage-volume

# Apply the new changes
$ kubectl apply -f path_to_output-directory/weblogic-domains/domainUID/
domain.yaml
```

Wait till all the managed servers are up.

5

Configuring and Deploying Pre-built Integrations

This chapter provides instructions for configuring and deploying the AIA pre-built integrations.

Configuring AIA PIP Credentials

Deployment of AIA PIPs requires Kubernetes secrets for OSM, BRM, Siebel, ODI, Xref, and SOA. Do not create a Kubernetes secret if you are not configuring OSM, Siebel, ODI or BRM with AIA.

To configure AIA PIP credentials, navigate to the **$AIA_DIR/oc-cn-aia-comms/scripts** folder and then run the **create-aiapip-credentials.sh** script with updated values for each component. See "Parameters for AIA PIPs Credentials" for details about the parameters.

- (Optional) For OSM, run the script with the following parameters:

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/
$ sh create-aiapip-credentials.sh \
    -c osm \
    -n namespace \
    -d domain_name \
    -OSM_CFS_WL_JMS_QUEUE_ACCESS_USER username \
    -OSM_CFS_WL_JMS_QUEUE_ACCESS_PSWD password \
    -OSM_PROV_WL_JMS_QUEUE_ACCESS_USER username \
    -OSM_PROV_WL_JMS_QUEUE_ACCESS_PSWD  password \
    -OSM_CFS_ADMIN_USER username \
    -OSM_CFS_ADMIN_PSWD password \
    -OSM_PROV_ADMIN_USER username \
    -OSM_PROV_ADMIN_PSWD password
```

- (Optional) For BRM, run the script with the following parameters:

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/
$ sh create-aiapip-credentials.sh \
    -c brm \
    -n namespace \
    -d domain_name \
    -BRM_AQ_USER username \
    -BRM_AQ_PSWD password
```

- (Optional) For Siebel, run the script with the following parameters:

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/
$ sh create-aiapip-credentials.sh \
    -c siebel \
    -n namespace \
    -d domain_name \
```

```
    -SBL_EAI_USER username \
    -SBL_EAI_PSWD password \
    -SBL_DB_USER username \
    -SBL_DB_PSWD password
```

- (Optional) For ODI, run the script with the following parameters:

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/
$ sh create-aiapip-credentials.sh \
    -c odi \
    -n namespace \
    -d domain_name \
    -ODI_USER username \
    -ODI_PASSWORD password \
    -ODI_DB_USER username \
    -ODI_DB_PSWD password
```

- For Xref, run the script with the following parameters:

> **Note:**
>
> Ensure that you specify a working DB SYS username and password for the Xref scripts.

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/
$ sh create-aiapip-credentials.sh \
    -c xref \
    -n namespace \
    -d domain_name \
    -XREF_SCHEMA_NAME schema_name \
    -XREF_SCHEMA_PASSWORD schema_password \
    -XREF_SCHEMA_SYS_USER sys_user \
    -XREF_SCHEMA_SYS_PASSWORD sys_password
```

- For SOA, run the script with the following parameters:

> **Note:**
>
> Ensure that you specify a working DB SYS username and password for the SOA scripts.

```
$ cd $AIA_DIR/oc-cn-aia-comms/scripts/
$ sh create-aiapip-credentials.sh \
    -c soa \
    -n namespace \
    -d domain_name \
    -SOA_WL_ADMIN_USER username \
    -SOA_WL_ADMIN_PASSWORD password \
    -SOA_DB_USER username \
    -SOA_DB_PASSWORD password \
```

```
-SOA_DB_SYS_USER username \
-SOA_DB_SYS_PASSWORD password
```

Configuring the BRM JCA Adapter

> **Note:**
>
> For more information, see Integrating with JCA Resource Adapter in *Oracle Communications Billing and Revenue Management (BRM) Cloud Native System Administrator's Guide.*

To configure BRM JCA Adapter:

1. Update the **ra.xml** and **weblogic-ra.xml** files in the META-INF directory.
   For more information, see "Deploying and Configuring JCA Resource Adapter on Oracle WebLogic Server" in the *BRM JCA Resource Adapter Guide*.
2. Set the following parameters with actual values for a single cluster as shown in the following samples:
   - **ConnectionString:** `ip cm.brmcn 11960`
   - **FailoverConnectionString:** `root.0.0.0.1:password@cm.brmcn:11960`
   - **UserName:** `root.0.0.0.1`
   - **Password:** `password`
   - **SslWalletLocation:** `/u01/shared/brm/brm_adapter/wallet`

   > **Note:**
   >
   > This is the location inside SOA server pods. Ensure that the path in SslWalletLocation is correct and exists on PV with correct wallet files downloaded and saved from the CM pods of BRM Cloud Native.

   In a multi-cluster scenario, set the values of **`ConnectionString`** and **`FailoverConnectionString`** as shown in the following samples:
   - **ConnectionString:** `ip 192.0.2.1 31773`
   - **FailoverConnectionString:** `root.0.0.0.1:password@192.0.2.1:31773,root.0.0.0.1:password@192.0.2.1:31773`
3. Download wallet files from the CM pod of the BRM cloud native instance and re-package **ra.xml**, **weblogic-ra.xml**, and wallet files into **OracleBRMJCA15Adapter.rar**.
4. Upload the wallet files to location mentioned in **SslWalletLocation**.

Deploying the BRM JCA Adapter

You can install BRM JCA Adapter using either WebLogic Administration Console or AIA Helm charts.

Deploying BRM JCA Adapter using WebLogic Administration Console

To deploy BRM JCA Adapter using WebLogic Admin Console:

1. Log in to WebLogic Admin Console.
2. Click **Lock and Edit**.
3. Go to **Deployments** and then to **Install**.
4. Select the archive file.
   - If the archive file is present on the host machine, select Upload your files.
   - If the archive file is present on pod or on PV, select Path from the menu.
5. In the **Choose installation type and scope** screen, do the following:
   - Choose **Install this deployment as an application**.
   - Choose **Scope** and then choose **Global**.
   - Click **Next**.
6. In the **Select Deployment targets** screen, do the following:
   - In the Servers section, select **AdminServer**.
   - In the Clusters section, select **All servers in the cluster**.
   - Click **Next**.
7. In the **Optional Settings** screen, do the following:
   - Update the name in the **General** field to OracleBRMJCA15Adapter.
   - In the **source accessibility** section, use the defaults defined by the deployment's targets.
   - In the **plan source accessibility** section, use the same accessibility plan as the application.
   - Click **Next**.
8. In the **Review your choices and click Finish** screen, do the following:
   - In the **additional configuration** section, select **Yes, take me to the deployment's configuration screen**.
   - Click **Finish**.
9. Click **Activate Changes**.
10. Go to **Deployments**, select **OracleBRMJCA15Adapter**, and then select **control** and do the following:
    - If State is prepared and not active, select the checkbox for **OracleBRMJCA15Adapter** and then go to **Start** and select **Servicing all requests**.
    - Confirm state is active.

Deploying BRM JCA Adapter Using AIA Helm Charts

The AIA Helm chart for deploying BRM JCA Adapter requires the following:

- AIA PV
- **plan.xml**

To deploy BRM JCA Adapter:

1. Update the following **values.yaml** files as per your requirement:

- **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml**. See "Global Parameters" for details about the parameters.
- **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-install-brm-adapter/values.yaml**.
  The following table lists the parameters in the **values.yaml** file.

Table 5-1 Parameters in the values.yaml file

Parameter	Description	Default
undeployExistingAdapter	To undeploy existing BRM JCA Adapter with name **OracleBRMJCA15Adapter**, set this to **true**. If set to **false**, jobs will exit after checking if any OracleBRMJCA15Adapter deployment exists.	true
deploymentPath	Path where BRM JCA Adapter archive file is present under the BRM directory,**/u01/shared/brm/brm_adapter**.	OracleBRMJCA15Adapter.rar
deployUsingPlan	To deploy using plan.xml, set this to **true**.	false
planPath	If **deployUsingPlan** is set to **true**, uncomment this parameter and provide the path where **plan.xml** exists under the BRM directory,**/u01/shared/brm/brm_adapter**.	Not applicable

2. Upload the files to AIA PV:
   - Upload the **OracleBRMJCA15Adapter.rar** archive file:

     ```
     kubectl cp OracleBRMJCA15Adapter.rar SOA_Admin_Pod:/u01/shared/brm/
     brm_adapter/ -n SOA_Namespace
     ```

   - (Optional) Upload the **plan.xml** file:

     ```
     # kubectl cp plan.xml SOA_Admin_Pod:/u01/shared/brm/brm_adapter/ -n
     SOA_Namespace
     ```

3. Deploy BRM JCA Adapter by using the **values.yaml** file:

   ```
   cd $AIA_DIR/oc-cn-aia-comms/helm-charts/

   helm install aia-comms-install-brm-adapter \
        aia-comms-chart/charts/aia-comms-install-brm-adapter/ \
        --namespace namespace \
        --values ./aia-comms-chart/values.yaml \
        --values ./aia-comms-chart/charts/aia-comms-install-brm-adapter/
   values.yaml
   ```

   This chart restarts the managed servers as required during the installation.

4. Wait till **aia-comms-install-brm-adapter-job** completes before proceeding to the next step.

Configuring Pre-built Integrations

You can configure the following Oracle Communications pre-built integration options:

- Order to Cash
- Agent-Assisted Billing Care

Run the aia-comms-config helm chart to configure pre-build integrations.

This chart requires the following:

- aia-comms-pv-pvc helm chart
- BRM JCA adapter deployed
- All secret credentials
- ODI installed (For AABC, ODI Agent URL should be accessible inside the Kubernetes pods)

> **Note:**
>
> This chart requires a minimum Java heap size of 4 GB, upto a maximum of 8 GB on the Kubernetes node.
> If this configuration is not available in the Kubernetes cluster, update the templates YAML file as follows:
>
> ```
> $ vi $AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-deploy-aiapip/charts/aia-comms-config/templates/aia-comms-execute-runconfig-job.yaml
>
> # Update the value for USER_MEM_ARGS environment variable
> # "-Djava.security.egd=file:/dev/./urandom -Xmsinitial_heap_size -Xmxmax_heap_size "
> # -Xmsinitial_heap_size is the initial size of the heap.
> # -Xmxmax_heap_size is the maximum size of the heap.
> ```

To configure pre-built integrations:

1. Update the following **values.yaml** files as per your requirement:
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml**. See "Global Parameters" for details about the parameters.
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-deploy-aiapip/values.yaml**. See "Parameters for AIA PIPs" for details about the parameters.
2. Run the following command to install the Helm chart:

   > **Note:**
   >
   > This chart requires all of the configured managed servers up and running.

   ```
   $ cd $AIA_DIR/oc-cn-aia-comms/helm-charts/
   $ helm install aia-comms-config \
        aia-comms-chart/charts/aia-comms-deploy-aiapip/charts/aia-comms-config/ \
   ```

```
--namespace namespace \
--values ./aia-comms-chart/values.yaml \
--values ./aia-comms-chart/charts/aia-comms-deploy-aiapip/values.yaml
```

This chart restarts the managed servers as required during the installation.

3. Wait till **aia-comms-execute-runconfig-job** completes before proceeding to the next step.

   Depending on the configuration, this job may take hour(s) for completion.

If you encounter any errors during the deployment, refer to the "Redeploying Pre-built Integrations Configurations Helm Chart" section in "Troubleshooting Issues".

Deploying Pre-built Integrations

After configuring the pre-built integrations, you must deploy them to the SOA server by using the deployment helm chart **aia-comms-pips**:

This chart requires the following:

- **aia-comms-pv-pvc** helm chart
- BRM JCA adapter deployed
- All secret credentials
- ODI installed (For AABC, the ODI Agent URL should be accessible inside the Kubernetes pods.)
- **aia-comms-config** chart

> **Note:**
>
> This chart requires a minimum Java heap size of 4 GB, up to a maximum of 8 GB on the Kubernetes node.
> If this configuration is not available in Kubernetes cluster, update the templates YAML file as follows:
>
> ```
> $ vi $AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-deploy-aiapip/charts/aia-comms-pip/templates/aia-comms-deploy-aiapip-job.yaml
>
> # Update the value for USER_MEM_ARGS environment variable
> # "-Djava.security.egd=file:/dev/./urandom -Xmsinitial_heap_size -Xmxmax_heap_size "
> # -Xmsinitial_heap_size is the initial size of the heap.
> # -Xmxmax_heap_size is the max size of the heap.
> ```

To deploy pre-built integrations:

1. Update the following **values.yaml** files as per your requirement:
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml**. See "Global Parameters" for details about the parameters.

- **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-deploy-aiapip/values.yaml**. See "Parameters for AIA PIPs" for details about the parameters.

2. Run the following command to install the Helm chart:

> **Note:**
>
> This chart requires all of the configured managed servers up and running.

```
$ cd $AIA_DIR/oc-cn-aia-comms/helm-charts/
$ helm install aia-comms-deploy-pip \
     aia-comms-chart/charts/aia-comms-deploy-aiapip/charts/aia-comms-pip/ \
     --namespace namespace \
     --values ./aia-comms-chart/values.yaml \
     --values ./aia-comms-chart/charts/aia-comms-deploy-aiapip/values.yaml
```

   This chart restarts the managed servers as required during the installation.

3. Wait till **aia-comms-deploy-aiapip-job** completes before proceeding to the next step.

   Depending on the configuration, this job may take hour(s) for completion.

To validate PIPs deployment, see "Validating the AIA Cloud Native Deployment".

If you encounter any errors during the deployment, refer to the "Redeploying Pre-built Integrations Helm Chart" section in "Troubleshooting Issues".

6

Performing Post-deployment Tasks

This chapter describes the tasks you perform after deploying AIA cloud native.

Integrating Applications with AIA Cloud Native

This section describes procedures for integrating the following cloud native applications with AIA cloud native:

> **Note:**
>
> Before proceeding with integrating these applications, ensure that all web user interfaces of these applications are available for integration.

- Siebel CRM Cloud Native. See the discussion about Integrating Siebel CRM on Containers with AIA Cloud Native in *Oracle Communications Digital Business Experience Solution Deployment Guide*.
- Billing and Revenue Management (BRM) Cloud Native. See the discussion about Integrating BRM Cloud Native with AIA Cloud Native in *Oracle Communications Digital Business Experience Solution Deployment Guide*.
- Order and Service Management Cloud Native. See the discussion about Integrating OSM Cloud Native with AIA Cloud Native in *Oracle Communications Digital Business Experience Solution Deployment Guide*.
- AABC Cloud Native. See the discussion aboutIntegrating ODI with AIA Cloud Native in *Oracle Communications Digital Business Experience Solution Deployment Guide*.

See *Application Integration Architecture Compatibility Matrix* for supported and recommended versions of these applications.

Deploying Custom Components

This section describes how to deploy custom components.

You can deploy the following:

- **SOA Adapter customizations**. For instructions, refer to the Oracle Fusion Middleware on Kubernetes documentation at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/adminguide/persisting-soa-adapters-customizations/.
- **Custom SOA composites**. For instructions, refer to the Oracle Fusion Middleware on Kubernetes documentation at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/adminguide/deploying-composites/.
- **Custom AIA artifacts**. See "Deploying Custom AIA Artifacts" for instructions.

Deploying Custom AIA Artifacts

The deployment of the artifacts is done by AIA Installation Driver (AID). AID takes the deployment plan and the **AIAInstallProperties.xml** file as input. Based on the tags specified in the deployment plan, AID configures and deploys the artifacts onto the server.

AID supports the following deployment plans:

- Main Deployment Plan
- Supplementary Deployment Plan
- Custom Deployment Plan

Main Deployment Plan is auto-generated by the Deployment Plan Generator. Whereas, Supplementary Deployment Plan and Custom Deployment Plan are handcoded. Support to add custom deployment tags to the main deployment plan is available through Pre-Install and Post-Install sections in the Deployment plan. However, the problem with using these sections is that the deployment plan may not be upgrade-safe. To mitigate the issue, supplementary and custom deployment plans are introduced. The supplementary deployment plan is used by the internal Pre-Built Integration development team. Use custom deployment plan to meet the requirement of non-native artifact deployment plan.

The running sequence of deployment plans followed by AID is as follows:

1. Main Deployment Plan
2. (Optional) Supplementary Deployment Plan
3. (Optional) Custom Deployment Plan

> **Note:**
>
> To facilitate durability across upgrades and patch updates, place the custom modified files in a directory path different from AIA-shipped *PIP_Name* **DP.xml** and *PIP_Name* **SupplementaryDP.xml**.

The following sections show the deployment commands for various deployment scenarios.

Deploying AIA Shipped Native Artifacts and Non-native Artifacts

This scenario does not involve any customizations. The following command takes the main deployment plan and the supplementary deployment plan which are shipped with the Pre-Built Integration installer as input.

Run this command inside the domain admin pod:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml \
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/PIP_nameDP.xml  \
-DSupplementaryDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameSupplementaryDP.xml  \
-DDeploymentPolicyFile=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameConditionalPolicy.xml
```

Deploying Modified AIA-shipped Artifacts

This section describes how to deploy modified AIA-shipped native and non-native artifacts.

Deploying Modified Native Artifacts and Original Non-native Artifacts

For modified native artifacts scenario, re-harvest the modified artifacts and regenerate the deployment plan, and name it as *PIP_Name* **CustomDP.xml**. Pass this as the main deployment plan, instead of the shipped deployment plan.

Run the following AID command inside the domain admin pod:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml \
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameCustomDP.xml  \
-DSupplementaryDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameSupplementaryDP.xml  \
-DDeploymentPolicyFile=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameConditionalPolicy.xml
```

Deploying Original Native Artifacts and Modified Non-native Artifacts

For the original native artifacts and modified non-native artifacts scenario, copy the contents of the shipped supplementary DP to a new file and name it as *PIP_Name* **CustomSupplementaryDP.xml**. Modify this file with the customizations. This is passed as the supplementary deployment plan, instead of the shipped supplementary DP.

Run the following AID command inside the domain admin pod:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml \
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/PIP_nameDP.xml  \
-DSupplementaryDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameCustomSupplementaryDP.xml  \
-DDeploymentPolicyFile=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameConditionalPolicy.xml
```

Deploying New or Custom Built Artifacts

This section discusses how to deploy newly added native and non-native artifacts.

Deploying Newly-added Native Artifacts and Original Non-native Artifacts

If you are introducing new native artifacts, harvest the new artifacts and regenerate the deployment plan for the new artifacts along with the shipped ones, and name it *PIP_Name* **CustomDP.xml**. Pass this as the main deployment plan, instead of the shipped deployment plan.

Run the following AID command inside the domain admin pod:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml \
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameCustomDP.xml  \
-DSupplementaryDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameSupplementaryDP.xml  \
-DDeploymentPolicyFile=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameConditionalPolicy.xml \
-l $COMMS_HOME/pips/PIP_name/DeploymentPlans/PIPDeploymentPlanName.log
```

Deploying Newly Added Non-native Artifacts

For new non-native artifacts scenario, add customizations to *PIP_Name* **CustomDP.xml**, which is an empty deployment plan shipped with the Pre-Built Integration. This custom plan is in the same location as the main plan. Pass this as Custom Deployment Plan to AID.

Run the following AID command inside the domain admin pod:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml \
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/PIP_nameDP.xml  \
-DSupplementaryDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameSupplementaryDP.xml  \
-DCustomDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameCustomDP.xml> \
-DDeploymentPolicyFile=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameConditionalPolicy.xml
```

The **AIAInstallProperties.xml** file contain the details of the AIA environment and is located at: **$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/AIAInstallProperties.xml**.

Undeploying Services

The undeployment plan is generated at the same location as the deployment plan with the name *PIP_Name* **UndeployDP.xml**. The undeployment plan is generated only for native artifacts modified through the Project Lifecycle Workbench. This contains undeploy tasks for all the services deployed and the configurations done as part of the Deployment Plan. The undeployment plan is run using the AID.

The undeployment command is similar to the deployment plan command except for the input argument and an additional argument "Uninstall". You run this command inside the domain admin pod.

For example, if you have used the following command to deploy modified native artifacts:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml \
```

```
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/PIP_nameDP.xml  \
-DSupplementaryDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameSupplementaryDP.xml  \
-DDeploymentPolicyFile=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameConditionalPolicy.xml
```

Then, the undeployment command would be:

```
source $COMMS_AIA_HOME/comms_home/bin/commsenv.sh
ant -f $SOA_HOME/aiafp/Install/AID/AIAInstallDriver.xml \
-DPropertiesFile=$VOLUME_DIR/domains/$DOMAIN_NAME/soa/aia/bin/
AIAInstallProperties.xml Uninstall \
-DDeploymentPlan=$COMMS_HOME/pips/PIP_name/DeploymentPlans/
PIP_nameUndeployDP.xml
```

However, for non-native artifacts, generate the undeployment plan manually:

1. Copy the supplementary deployment plan and name it as *PIP_Name* **UndeploySupplementaryDP.xml** or *PIP_Name* **UndeployCustomSupplementaryDP.xml**, depending on the supplementary deployment plan name.
2. In the new deployment plan, change the action attributes of all the tasks from "deploy" to "undeploy" or from "create" to "delete".

Installing SSL Certificates

To install SSL certificates:

> **Note:**
>
> This procedure automates the process of importing certificates to kss-based keystore. If you wish to use jks-based keystore, set it up manually.

1. Create keystore credentials by running the following commands:

   > **Note:**
   >
   > Ensure that certificates are available in the **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-certs/certificates** directory and that proper management policies are used to store the private keys.

```
cd $AIA_DIR/oc-cn-aia-comms/scripts/
./create-keystore-secret.sh -h
usage: ./create-keystore-secret.sh -k keystore -c custTrust -i
custIdentity [-d domainUID] [-n namespace] [-h]
  -k  pasword for  KeyStore, must be specified.
  -c  password for  Cust Trust, must be specified.
  -i  password for Cust Identity , must be specified.
  -d domainUID, optional. The default value is soainfra. If specified, the
secret will be labeled with the domainUID unless the given value is an
```

```
empty string.
  -n namespace, optional. Use the soans namespace if not specified
  -h Help
```

2. Run the following command, which creates a Kubernetes secret:

```
kubectl -n soa_namespace create secret generic secret_name --from-
file=certificates
```

3. Update values in the following files:
   - **oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml**
   - **oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-certs/values.yaml**

   The values to be updated are:
   - **name**: The Keystore name on which the operation is to be performed. For example, custTrust or CustIdentity.
   - **identityKeyStoreName**: The Keystore name of the Identity Keystore. For example, custIdentity.
   - **trustKeyStoreName**: The Keystore name of Custom Trust Keystore. For example, custTrust.
   - **type**: The type of the certificate which is to be imported, updated, and deleted. For example, TrustedCertificate.
   - For each Siebel certificate and OSM certificate to be imported, specify the following:
     - **fileName**: The certificate filename which is to be imported. This should be available in the **aia-comms-chart/charts/aia-comms-certs/certificates** directory.
     - **alias**: The alias name.
     - **operation**: The supported operation types are: **import**, **delete**, and **update**. Leave this value empty or commented in case no operation is required to be performed on either OSM certificate or Siebel certificate.

   > **Note:**
   >
   > The helm chart execution supports adding, updating, and deleting the certificate. The mandatory fields are:
   > - **import**: alias, fileName, operation
   > - **delete**: alias, operation
   > - **update**: alias, fileName (the file name of the new certificate to be updated), operation

4. After the **aia-comms-ssl-certificate-job** completes, restart the domain to import, update, or delete the OSM certificate, Siebel certificate, or both certificates as specified in the values.yaml file of the chart.

```
cd $AIA_DIR/oc-cn-aia-comms/helm-charts
 helm install aia-comms-certs\
    aia-comms-chart/charts/aia-comms-certs/ \
    --namespace namespace \
```

```
--values ./aia-comms-chart/values.yaml \
--values ./aia-comms-chart/charts/aia-comms-certs/values.yaml
```

5. Restart the domain.

Updating Files in AIA MDS

To update multiple files at once in **AIA MDS**:

1. Update the following **values.yaml** files as per your requirement:
   $AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml

   $AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-update-mds/values.yaml

   See Global Parameters for details about the parameters.

Table 6-1 Parameters in the values.yaml

Parameter	Description	Example

fileList	List of files with their full paths in MDS.
	Note: This is the destination path where files need to be uploaded in MDS and not a source path where files are present in AIA PV → /u01/shared/mds.
		`soa/configuration/default/AIAConfigurationProperties.xml`
		
		`apps/AIAMetaData/config/AIAInstallProperties.xml`

2. Upload the files to AIA PV:

```
kubectl cp File_to_be_Uploaded_in_MDS SOA_Admin_Pod:/u01/shared/mds/-n
SOA_Namespace
```

3. Run the following command:

```
$ cd $AIA_DIR/oc-cn-aia-comms/helm-charts/
$ helm install aia-comms-update-mds \
     aia-comms-chart/charts/aia-comms-update-mds/ \
     --namespace namespace \
     --values ./aia-comms-chart/values.yaml \
     --values ./aia-comms-chart/charts/aia-comms-update-mds/values.yaml
```

4. Wait till the **aia-comms-update-mds-job** completes.
5. Restart the domain.
   This validates the path in **MDS**.

Validating the AIA Cloud Native Deployment

To validate your AIA cloud native deployment:

1. Verify the logs:

a. Run the following command:

```
kubectl exec -it admin_server -n namepspace -- bash

cd $ORACLE_HOME/user_projects/domains/domain_name
```

b. Verify there are no errors in the log files.

> **Note:**
>
> If you want to verify the PIP deployment log, refer to the below log files within the admin server pod:
>
> - /u01/shared/logs/runconfig.log
> - /u01/shared/logs/setupo2c.log
>
> As these logs contain sensitive information, delete them once the validation is complete.

2. Verify the deployment of composites for Siebel CRM, OSM, and BRM. See "Verifying Composite Deployment" in the *AIA Installation Guide* for details. To generate a diagnostic report, see "Generating a Diagnostic Report."
3. Test the order flow to check connectivity between Siebel, OSM, and BRM.

7

Managing Your Cloud Native Deployment

This chapter describes the tasks you perform to manage your AIA cloud native deployment.

Scaling the AIA Application Cluster

To scale the AIA application cluster, perform the procedures described in the following sections in the WebLogic Kubernetes Operator user's guide:

- Scaling: https://oracle.github.io/weblogic-kubernetes-operator/managing-domains/domain-lifecycle/scaling/
- Scripts: https://oracle.github.io/weblogic-kubernetes-operator/managing-domains/domain-lifecycle/scripts/

Configuring Dynamic Autoscaling

You can configure your AIA cloud native deployment with dynamic autoscaling. For dynamic autoscaling, a minimum of WKO version 4.0.*x* is required, along with Horizontal pod Autoscaler configured with threshold for the resource metrics.

For information on how to configure dynamic autoscaling, see WKO documentation at: https://oracle.github.io/weblogic-kubernetes-operator/managing-domains/domain-lifecycle/scaling/#kubernetes-horizontal-pod-autoscaler-hpa

Restarting the AIA Cloud Native Instance

To restart (rolling restart) your AIA cloud native instance, perform the procedures described in the following sections in the WebLogic Kubernetes Operator user's guide:

- Restarting: https://oracle.github.io/weblogic-kubernetes-operator/4.0/managing-domains/domain-lifecycle/restarting/
- Scripts: https://oracle.github.io/weblogic-kubernetes-operator/4.0/managing-domains/domain-lifecycle/scripts/

Deleting the AIA Cloud Native Instance

To delete the AIA cloud native instance:

1. Get the details of AIA and SOA PV paths:

```
$ kubectl describe pv soainfra-domain-pv
$ kubectl describe pv aia-comms-shared-pv
```

2. Uninstall the helm charts:

```
helm uninstall aia-comms-pv-pvc -n namespace
helm uninstall aia-comms-config -n namespace
```

```
helm uninstall aia-comms-install-brm-adapter -n namespace
helm uninstall aia-comms-deploy-aiapip -n namespace
helm uninstall aia-comms-certs-osm -n namespace
helm uninstall aia-comms-certs-siebel -n namespace
```

3. Delete the Kubernetes Network resources, which AIA cloud native creates as part of AIA PV creation:
   - Get the details of the resources:

     ```
     $ kubectl get serviceAccount -n namespace | grep cluster-kubectl
     $ kubectl get clusterrole | grep access-pod-cluster-role
     $ kubectl get rolebinding -n namespace | grep access-pod-role-binding
     ```

   - Delete the resources:

     ```
     $ kubectl delete serviceAccount service-account-name -n namespace
     $ kubectl delete clusterrole cluster-role-name
     $ kubectl delete rolebinding role-binding-name -n namespace
     ```

4. Uninstall the domain and drop the RCU schema. For instructions, see: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/cleanup-domain-setup/.
5. Clean up the persistent volume data. To remove the AIA PV that is generated during AIA deployment, using appropriate privileges, delete the contents of the storage attached to the domain home persistent volume manually.
   For example, to delete the persistent volume of type host_path, run:

   ```
   $ rm -rf /export/shared/*
   ```

6. Clean up ODI Studio.

Monitoring the AIA Cloud Native Domain and Publishing Logs

You can monitor your AIA cloud native deployment using Grafana and OpenSearch and publish logs to Elasticsearch and Kibana.

For enabling metrics, JKS-based keystores support both types of options for deploying Weblogic Monitoring Exporter:

- Deployment of WME war file to Weblogic console. For details, see "https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/adminguide/monitoring-soa-domains/#set-up-monitoring."
- Deploying WME as a sidecar. For details, see "https://github.com/oracle/weblogic-monitoring-exporter#use-the-monitoring-exporter-with-weblogic-kubernetes-operator"

> **Note:**
>
> KSS-based keystores supports deploying WME as a sidecar only.

Refer to the Oracle Fusion Middleware on Kubernetes documentation for information about using Grafana for monitoring your deployment at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/adminguide/monitoring-soa-domains/#set-up-monitoring.

For information about using Elasticsearch and Kibana, see the WKO documentation at: https://oracle.github.io/weblogic-kubernetes-operator/4.0/samples/elastic-stack/.

Patching and Upgrading Your AIA Cloud Native Deployment

To patch and upgrade your AIA cloud native deployment from prior releases of AIA cloud native, perform the procedures described in the *Oracle Fusion Middleware on Kubernetes* documentation at: https://oracle.github.io/fmw-kubernetes/24.2.2/soa-domains/patch_and_upgrade/.

> **Note:**
>
> Upload the **AIAInstallProperties.xml** file to MDS after patching the new image and then restart the AIA domain. See "Restarting the AIA Cloud Native Instance".

Troubleshooting Issues

This section describes how to troubleshoot common issues with your AIA cloud native deployment.

Redeploying the AIA PV PVC Helm Chart

You can redeploy the aia-comms-pv-pvc helm chart if the aia-comms-config helm chart is not installed.

The aia-comms-pv-pvc helm chart creates the following:

- Kubernetes PV PVC
- Kubernetes job aia-comms-create-home-job to populate AIA PV
- Service Account with name domain_name-cluster-kubectl
- ClusterRole of name domain_name-access-pod-cluster-role
- RoleBinding of name domain_name-access-pod-role-binding

The Kubernetes job aia-comms-create-home-job pods would be in the Pending state till the required PV PVC is created.

To redeploy the PV PVC helm chart:

1. Find the pod name for a given job:

   ```
   $ kubectl get pods -n namespace | grep aia-comms-create-home-job
   ```

2. Get logs of the pod and identify issues:

   ```
   $ kubectl logs job_pod_name -n namespace
   ```

3. Clean up the AIA Persistent Volume data if there is any. To get details of storage path, run:

   ```
   $ kubectl describe pv AIA_PV_name
   ```

To remove the AIA PV that is generated during AIA deployment, using appropriate privileges, delete the contents of the storage attached to the domain home persistent volume manually. For example, to delete the persistent volume of type host_path, run:

```
$ rm -rf /export/shared/*
```

4. Uninstall the helm chart.

```
$ helm uninstall aia-comms-pv-pvc -n namespace
```

5. Reinstall the helm chart with updated parameters, if any. See "Creating PV-PVCs"

Redeploying the BRM JCA Adapter Helm Chart

The AIA Helm chart for deploying BRM JCA Adapter requires the following:

- AIA PV
- **OracleBRMJCA15Adapter.rar**
- **plan.xml**

To redeploy this chart:

1. Find the pod name for a given job:

```
$ kubectl get pods -n namespace | grep aia-comms-install-brm-adapter-job
```

2. Get logs of the pod and identify issues:

```
$ kubectl logs job_pod_name -n namespace
```

3. Uninstall the helm chart.

```
$ helm uninstall aia-comms-install-brm-adapter namespace
```

4. Reinstall the helm chart with the parameter `undeployExistingAdapter` set to `true`. See "Deploying the BRM JCA Adapter".

Redeploying Pre-built Integrations Configurations Helm Chart

The aia-comms-config helm chart requires the following:

- aia-comms-pv-pvc helm chart
- aia-comms-install-brm-adapter helm chart (BRM Adapter deployment)
- all secret credentials
- ODI installation (in case of AABC installation)

This helm chart updates the AIA PV and SOA PV data as per configuration. Hence, this chart does not support the re-run of helm installation of this chart.

To reinstall this chart:

1. Find the pod name for a given job:

```
$ kubectl get pods -n namespace | grep aia-comms-execute-runconfig-job
```

2. Get logs of the pod:

```
$ kubectl logs job_pod_name -n namespace
```

3. Identify and resolve issues, if any:
   - `ERROR - Deployment found`: Confirm that this is not a rerun of the aia-comms-config helm chart. The chart will not allow the rerun of this helm chart.
   - `User failed to be authenticated`: Confirm the username credentials are mentioned in the AIA PIPs SOA credentials.
   - `Error connecting to server please check to see if the server exists`: Confirm that the username credentials are mentioned in the SOA credentials. Confirm all the servers are accessible.
   - `Configuration error in executing runConfig.sh`: Confirm the values entered in all **values.yaml** files that are used in the installation.
4. Uninstall the helm chart:

```
$ helm uninstall aia-comms-config -n namespace
```

5. Delete Oracle SOASuite Domain. See "Deleting the AIA Cloud Native Instance". For this case, you do not need to delete namespaces or helm charts for weblogic-operator and load balancer.
6. Redeploy the domain. See "Deploying AIA Cloud Native".
7. Install the aia-comms-config helm chart.

Redeploying Pre-built Integrations Helm Chart

The aia-comms-pips helm chart requires the following:

- aia-comms-pv-pvc helm chart
- aia-comms-install-brm-adapter helm chart (BRM Adapter deployment)
- all secret credentials
- aia-comms-config helm chart
- ODI installation (in case of AABC installation)

To reinstall this chart:

1. Find the pod name for a given job:

```
$ kubectl get pods -n namespace | grep aia-comms-deploy-aiapip-job
```

2. Get logs of the pod:

```
$ kubectl logs job_pod_name -n namespace
```

3. Identify and resolve issues, if any:
   - `Configuration error in executing runConfig.sh`: Confirm that the aia-comms-config helm chart has been installed.
   - `ERROR: runconfig log file does not exist.`: Confirm aia-comms-config helm chart is installed before running this helm chart.

- `Error connecting to server please check to see if the server exists`: Confirm that the username credentials are mentioned in the AIA PIPs credentials.
- `Error in deployment`: Review the helm chart logs for deployment errors. If there are no errors related to aia-comms-config helm chart, resolve the errors and perform steps 4 and 7. If the errors are related to aia-comms-config helm chart, continue with the steps from 4 to 7 to resolve the issues.

4. Uninstall the helm chart:

```
$ helm uninstall aia-comms-pips -n namespace
```

5. Delete Oracle SOA Suite Domain. See "Deleting the AIA Cloud Native Instance". For this case, you do not need to delete namespaces or helm charts for weblogic-operator and load balancer.
6. Redeploy the domain. See "Deploying AIA Cloud Native".
7. Install the aia-comms-pips helm chart.

Oracle SOA Suite Domain fails with "folder already exists" error

This error occurs when a domain folder already exists even before deploying Oracle SOA Suite domain. Generally, this error occurs during the redeployment of domain when domain PV is not cleaned up properly.

To resolve this issue, clean up the Oracle SOA Suite PV properly and redeploy.

Clean a previous deployment

To clean a previous deployment, perform the steps described in "Deleting the AIA Cloud Native Instance".

AIA Helm chart jobs are in the "imagePullBackOff" state

This is a Kubernetes error which occurs in case the Docker image is present on the worker node. The current version of AIA cloud native toolkit does not include **imagePullSecret** support in its template yaml files. For this error, for the **aia-comms-config**, **aia-comms-pips** helm chart, you do not need to follow the procedure of redeployment of the AIA helm chart. Once the image is available on a given worker node, pods execution continues.

To solve this issue, follow any redeployment option described in Table 7-1.

Table 7-1 Redeployment Options

Option	Redeployment Steps
Manually pull the image on the worker node using `docker pull image_name`	Once the image is available on a given worker node, pod execution continues on its own. You do not need to uninstall the helm chart.
Add `imagePullSecret` in the templates yaml files of the respective AIA helm chart.	Uninstall the helm chart and install it again after updating the yaml file.
Restrict Kubernetes cluster to deploy on certain worker nodes only, where the image is present until the installation of AIA cloud native.	Uninstall the helm chart and install it again after the Kubernetes configuration is done.

Switching the Launch Flag After Deployment

To enable integration with Launch, you configure a parameter in the AIA configuration properties file.

> **Note:**
>
> Perform this task only if you have missed enabling intergration with Launch during the installation of AIA. Do not perform this procedure for switching in a production environment.

To switch the launch flag post-deployment:

1. Get the latest **AIAConfigurationProperties.xml** file:
   a. In Enterprise Manager Console for AIA, navigate to the **MDS Configuration** section and export the zip file.
   b. Copy the **soa/configuration/default/AIAConfigurationProperties.xml** file from the exported zip file.
   See https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/asadm/managing-metadata-repository.html for more informnation on MDS.
2. In the `LaunchIntegrationParameters` module, switch the **O2C.isLaunchIntegrationEnabled** property value to either **true** or **false**.
3. Update AIA MDS with the updated **AIAConfigurationProperties.xml** file.
   See Updating Files in AIA MDS for more information.

Generating a Diagnostic Report

To generate a diagnostic report about your AIA cloud native instance, by using the Helm charts:

1. Update the following **values.yaml** files as per your requirement:
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml**. See "Global Parameters" for details about the parameters.
   - **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-diagnostic/values.yaml**

Table 7-2 Parameters in the values.yaml

Parameter	Description	Default
managedServerPort	Port number for Managed Server in the SOA cluster.	8001
clusterServiceName	SOA Domain cluster service name.	soainfra-cluster-soa-cluster

2. Run the following command to install the Helm chart:

```
cd $AIA_DIR/oc-cn-aia-comms/helm-charts/

helm install aia-comms-diagnostic \
        aia-comms-chart/charts/aia-comms-diagnostic/ \
        --values ./aia-comms-chart/values.yaml \
        --values ./aia-comms-chart/charts/aia-comms-diagnostic/values.yaml
\
        --namespace namespace
```

3. The reports are generated in the **/u01/shared/logs** folder in AIA PV.

8
Configuring Parameters in Helm Charts

This chapter lists and describes the parameters in the Helm charts.

The AIA cloud native toolkit contains the following charts:

- **aia-comms-pv-pvc**: This chart creates AIA PV/PVCs and service accounts.
- **aia-comms-deploy-aiapip**: This chart deploys AIA Foundation Pack and AIA Pre-integrated Packs (PIPs) in SOA and WebLogic servers.
- **aia-comms-certs**: This chart manages SSL trust certificates of Siebel and OSM in SOA and WebLogic servers.

The charts pick the customized values defined in the custom **values.yaml** configuration file and map them to the respective Kubernetes yaml files defined in the chart`s **/template** directory. Each chart can be installed and managed through helm commands.

Global Parameters

Table 8-1 lists the parameters that you configure globally. These are available in the **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/values.yaml** file.

Table 8-1 Global Parameters

Parameter	Description
image	AIA Docker image.
imagePullPolicy	AIA Docker image pull policy. Valid values are IfNotPresent, Always, Never.
containerPort	Kubernetes container port for the kubernetes job to be triggered.
namespace	Kubernetes namespace in which domain is created.
domainUID	Unique ID that is used to identify this particular domain where AIA needs to be installed.
domainName	Unique name that is used to identify this particular domain where AIA needs to be installed.
weblogicCredentialsSecretName	Name of the Kubernetes secret for the Administration Server's user name and password.
persistentVolumeClaimName	Name of the persistent volume claim created to host the domain home.
domainPVMountPath	Mount path of the domain persistent volume.
soaClusterName	Name of the SOA WebLogic Server cluster instance generated for the domain.
adminServerName	Name of the Administration Server.
adminServerNameSvc	Name of the Administration Server Base Service. This can be found in create-domain-job.yaml of output folder specified during domain creation.
adminPort	Port number for the Administration Server inside the Kubernetes cluster.
managedServerNameSvc	Name of the Managed Server Base Service name. This can be found in create-domain-job.yaml of output folder specified during domain creation.

Table 8-1 (Cont.) Global Parameters

Parameter	Description
sharedPersistentVolumeClaimName	Name of the persistent volume claim to be created to host the AIA home.

Parameters for AIA PV-PVC

Table 8-2 lists the parameters that you configure for AIA PV-PVC. These parameters are available in the **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-pv-pvc/values.yaml** file.

Table 8-2 Parameters for AIA PV-PVC

Parameter	Description
createPV	To create AIA PV PVC, specify **true**.
persistentVolumeName	Name of the persistent volume to be created to host the AIA home.
storageClassName	Kubernetes storage class name of the persistent volume claim to be created to host the AIA home.
aiacommsStorageType	Type of storage. Legal values are **NFS** and **HOST_PATH**. If using **NFS**, **aiacommsStorageNFSServer** must be specified.
aiacommsStoragePath	Physical path of the storage for the PV. When **aiacommsStorageType** is set to **HOST_PATH**, this value should be set the to path to the domain storage on the Kubernetes host. When **aiacommsStorageType** is set to **NFS**, then **aiacommsStorageNFSServer** should be set to the IP address or name of the DNS server, and this value should be set to the exported path on that server. Note that the path where the domain is mounted in the WebLogic containers is not affected by this setting; that is determined when you create your domain.
aiacommsStorageReclaimPolicy	Kubernetes PVC policy for the persistent storage. Valid values are: **Retain**, **Delete**, and **Recycle**.
aiacommsStorageSize	Total storage allocated for the AIA PVC.

Parameters for AIA PIPs

Table 8-3 lists the parameters that you configure for AIA PIPs. These parameters are available in the **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-deploy-aiapip/values.yaml** file.

Table 8-3 Parameters for AIA PIPs

Parameter	Description
Order2Cash_Siebel	Type true if you want to install PIPs for Siebel.
SPLIT_XREF	Creates Split XRef schema. The default is `true`.
Order2Cash_brm	Type true if you want to install PIPs for BRM.
Order2Cash_osm	Type true if you want to install PIPs for OSM.
AABC_PIP	Type true if you want to install PIPs for AABC.
DOMAIN_LOCATION	Weblogic Domain location for SOA pods.

Table 8-3 (Cont.) Parameters for AIA PIPs

Parameter	Description
isLaunchIntegrationEnabled	Type true if Launch is integrated. Otherwise, type false.
OSM_VERSION	The version of OSM to which you are connecting. This value must be 10 characters or less. For example: 7.5.0.0.9.
OSM_PROV_WL_JMS_QUEUE_ACCESS_PORT	The port assigned to the WebLogic Server host used for accessing inbound JMS queues for OSM SOM. For example: 7080.
OSM_CFS_WL_JMS_QUEUE_ACCESS_HOST	The WebLogic Server host used for accessing inbound JMS queues for OSM COM.
OSM_CFS_WL_JMS_QUEUE_ACCESS_PORT	The port assigned to the WebLogic Server host used for accessing inbound JMS queues for OSM COM. For example: 7080.
OSM_PROV_WL_JMS_QUEUE_ACCESS_HOST	The WebLogic Server host used for accessing inbound JMS queues for OSM SOM.
BRM_PRIMARY_CM_HOST	The IP address or DNS name of the BRM server's primary Connection Manager.
BRM_VERSION	The version of BRM to which you are connecting. This value must be 10 characters or less. For example, 15.1.0.0.0.
BRM_DB_HOST	The IP address or DNS name of the Oracle Database Advanced Queuing database instance where the BRM Synchronization Queue Manager Data Manager (DM_AQ) is configured.
BRM_AQ_DB_SID	The Oracle Database Advanced Queuing database system ID.
BRM_PRIMARY_CM_PORT	The port assigned to the BRM server's primary Connection Manager. For example: 11960.
BRM_AQ_QUEUE	The name of the queue configured for DM_AQ. For example: AQ_QUEUE.
BRM_DB_PORT	The port assigned to the Oracle Database Advanced Queuing database instance. For example: 1521.
BRM_DB_JDBC_URL	BRM Database JDBC URL. JDBC URL is used to connect and configure the corresponding database. Examples are: SID : jdbc:oracle:thin:@*host*:*port*:*sid* Service Name : jdbc:oracle:thin:@//*host*:*port*/*service_name*.
SPM_PROXY_PORT	Siebel Session Pool Manager Port Number. Can be empty if one is not available. For example: 1521.
SPM_PROXY_HOST	Siebel Session Pool Manager Host. Can be empty if one is not available.
XREF_TEMP_TABLESPACE	An existing temp tablespace name, which can be used to create XRef schema. Tablespace must exist before installing AIA cloud native.
XREF_DEFAULT_TABLESPACE	An existing users tablespace name, which can be used to create XRef schema. Tablespace must exist before installing AIA cloud native.
XREF_SCHEMA_SYS_ROLE	The role of the SOA database administrator. Example: SYSDBA.
XREF_SCHEMA_JDBC_URL	The URL to create the cross reference schema. The cross reference schema will be created in the database specified in the JDBC URL. Oracle recommends that you create the cross reference schema in the same database as the SOA database. The JDBC URL can be provided as per the SOA database configuration. If a different database is selected, specify the corresponding JDBC URL. Examples are: SID: jdbc:oracle:thin:@*host*:*port*:*sid* Service Name : jdbc:oracle:thin:@//*host*:*port*/*service_name*.
SBL_VERSION	The version of Siebel CRM to which you are connecting. This value cannot be longer than 10 characters. For example: 21.2.0.0.
SBL_HOST	The DNS name of the Siebel CRM host.

Table 8-3 (Cont.) Parameters for AIA PIPs

Parameter	Description
SBL_PROTOCOL	The internet protocol used to connect to the Siebel CRM server. For example: http://
SBL_LANG	The language used by the Siebel application. For example: enu.
SBL_ENTERPRISE_SERVER_NAME	The name of the Siebel Enterprise Server on which Siebel CRM is installed. For example: Siebel.
SBL_PORT	The port assigned to Siebel CRM. If Siebel is running on Kubernetes, you can use Tomcat's exposed port number. For example: 4432.
SBL_DB_HOST	The IP address or DNS name of the Siebel CRM database host.
SBL_DB_PORT	The port assigned to the Siebel CRM database. For example: 1521.
SBL_DB_SID	The Siebel CRM database system ID. For example: orcl.
SBL_DB_JDBC_URL	The URL to connect to the Siebel database. JDBC URL is used to connect and configure the corresponding database. Examples are: SID : jdbc:oracle:thin:@*host*:*port*:*sid* Service Name : jdbc:oracle:thin:@//*host*:*port*/*service_name*.
SOA_WL_MS_PORT	The port assigned to the SOA managed server or cluster proxy. This will be the port number of the SOA cluster service inside Kubernetes SOA namespace.
SOA_DB_SYS_ROLE	SOA Database System role name. Example: SYSDBA.
SOA_WL_ADMIN_PORT	The port assigned to the administration server. This will be the port number of the SOA admin server's service inside the Kubernetes SAO namespace.
SOA_WL_DOMAIN_NAME	The name of your SOA domain.
SOA_WL_MS_NAME	The name of the primary SOA managed server or the name of the SOA cluster. For single node WebLogic Server, use the WebLogic managed server's name. For cluster based WebLogic Server configuration, use the name of the SOA WebLogic Server cluster instance generated for the domain.
SOA_DB_JDBC_URL	The URL to connect to the SOA database. Example: SID : jdbc:oracle:thin:@*host*:*port*:*sid*Service Name : jdbc:oracle:thin:@//*host*:*port*/*service_name* TNS_ADMIN : jdbc:oracle:thin:@*db_name*?TNS_ADMIN=*tns_admin_location*.
SOA_WL_MS_HOST	The host of the SOA managed server or the proxy URL for the cluster. This will be SOA cluster service name inside Kubernates SOA namespace. The value can be derived from *domain_name*-cluster-soa-cluster.
SOA_WL_ADMIN_HOST	The host of the administration server for your SOA domain. This will be the service name of admin server inside Kubernetes SOA namespace.
ODI_AABC_AGENT_PORT	The port assigned to the agent. For example: 20910.
ODI_AABC_REPID	The repository ID of the Oracle Data Integrator work repository for integration artifacts.
ODI_AABC_REPNAME	The name of the Oracle Data Integrator work repository for integration artifacts. For example: WORKREP.
ODI_AABC_AGENT_APP_NAME	The application name for the standalone or Java EE agent. For example: oraclediagent Note: The application name for a standalone agent is always oraclediagent.

Table 8-3 (Cont.) Parameters for AIA PIPs

Parameter	Description
ODI_DB_HOST	The DNS name of the master repository database host. For example: `odim.example.com`
ODI_DB_SID	The master repository database system ID. For example: oracle.
ODI_DB_PORT	The port assigned to the master repository database. For example: 1521.
ODI_HOST	ODI host URL as per ODI Agents WSDL URL (only IP section is required).

Parameters for AIA Certificates

Table 8-4 lists the parameters that you configure for AIA Certificates. These parameters are available in the **$AIA_DIR/oc-cn-aia-comms/helm-charts/aia-comms-chart/charts/aia-comms-certs/values.yaml** file.

Table 8-4 Parameters for AIA Certificates

Parameter	Description
certificate.name	Name of the keystore where the certificate operation is to be performed. This value can be either the identityKeyStoreName or trustKeyStoreName.
certificate.identityKeyStoreName	Name of the identity KeyStore.
certificate.trustKeyStoreName	Name of the Trusted keyStore.
certificate.type	The type of certificate to be imported: TrustedCertificate.
wlsServerHome	The root directory of your WebLogic installation. If this is not changed during image creation, then root directory will be used as the default.
certificateSecretName	Specify the Kubernetes secret name used to create secrets of certificates to be imported.
seibel.alias	Specify the Siebel certificate details to be imported, deleted, or updated. The certificate alias name.
seibel.fileName	The file name placed in certificate directory to be imported.
seibel.operation	Supported operation types are import, delete, and update. Leave the operation field empty or commented in case no operation is required to be performed.
osm.alias	Specify the osm certificate details to be imported, deleted, or updated. The certificate alias name.
osm.fileName	Filename placed in certificate directory to be imported.
osm.operation	Supported operation types are import, delete, and update. Leave the operation field empty or commented in case no operation is required to be performed.

Parameters for AIA PIPs Credentials

Table 8-5 lists the parameters that you configure for AIA PIPs credentials. Deployment of AIA PIPs requires Kubernetes secrets for OSM, BRM, Siebel, Xref, and SOA. Do not create a Kubernetes secret for a particular component, if any of the components among OSM, Siebel, or BRM is not being deployed.

Table 8-5 Parameters for AIA PIPs Credentials

Component	Parameter	Description
OSM	OSM_CFS_WL_JMS_QUEUE_ACCESS_USER	OSM CFS Weblogic JMS Queue User Name
OSM	OSM_CFS_WL_JMS_QUEUE_ACCESS_PSWD	OSM CFS Weblogic JMS Queue Password
OSM	OSM_PROV_WL_JMS_QUEUE_ACCESS_USER	OSM Provisioning Weblogic JMS Queue User Name
OSM	OSM_PROV_WL_JMS_QUEUE_ACCESS_PSWD	OSM Provisioning Weblogic JMS Queue Password
OSM	OSM_CFS_ADMIN_USER	OSM CFS Admin User Name
OSM	OSM_CFS_ADMIN_PSWD	OSM CFS Admin Password
OSM	OSM_PROV_ADMIN_USER	OSM Provisioning Admin User Name
OSM	OSM_PROV_ADMIN_PSWD	OSM Provisioning Admin Password
BRM	BRM_AQ_USER	BRM AQ Queue User Name
BRM	BRM_AQ_PSWD	BRM AQ Queue Password
Siebel	SBL_EAI_USER	Siebel EAI Server User Name
Siebel	SBL_EAI_PSWD	Siebel EAI Server Password
Siebel	SBL_DB_USER	Siebel Database User Name
Siebel	SBL_DB_PSWD	Siebel Database Password
Xref	XREF_SCHEMA_NAME	Given Schema name will be used to create XRef
Xref	XREF_SCHEMA_PASSWORD	Given password will be used to create Xref
Xref	XREF_SCHEMA_SYS_USER	SOA Database SYS user name
Xref	XREF_SCHEMA_SYS_PASSWORD	SOA Database SYS password
SOA	SOA_WL_ADMIN_USER	SOA Weblogic server admin user name
SOA	SOA_WL_ADMIN_PASSWORD	SOA Weblogic server admin password
SOA	SOA_DB_USER	SOA DB user name in the format: *RCU_Prefix*_**SOAINFRA**
SOA	SOA_DB_PASSWORD	SOA DB password
SOA	SOA_DB_SYS_USER	SOA DB SYS admin user name
SOA	SOA_DB_SYS_PASSWORD	SOA DB SYS admin user password
ODI	ODI_USER	The Oracle Data Integrator administrator's user name. For example: SUPERVISOR
ODI	ODI_PASSWORD	The password for the Oracle Data Integrator administration user.
ODI	ODI_DB_USER	The master repository database user. For example: ODI_REPO
ODI	ODI_DB_PSWD	The master repository database password.

9
Securing Your AIA Cloud Native Deployment

See *AIA Security Guide* for security considerations for your AIA cloud native deployment.