Oracle® Application Integration Architecture

Cloud Native Deployment Guide

Release 12.3.1
F79221-02
February 2024

Oracle Application Integration Architecture Cloud Native Deployment Guide, Release 12.3.1

F79221-02

Copyright © 2023, 2024, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

1 Overview of the AIA Cloud Native Deployment

2 Prerequisites for Your AIA Cloud Native Deployment

3 Creating AIA Cloud Native Images

4 Deploying AIA Cloud Native

5 Performing Post-deployment Tasks

6 Managing Your Cloud Native Deployment

7 Configuring Parameters in Helm Charts

8 Securing Your AIA Cloud Native Deployment

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-RODOD 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:

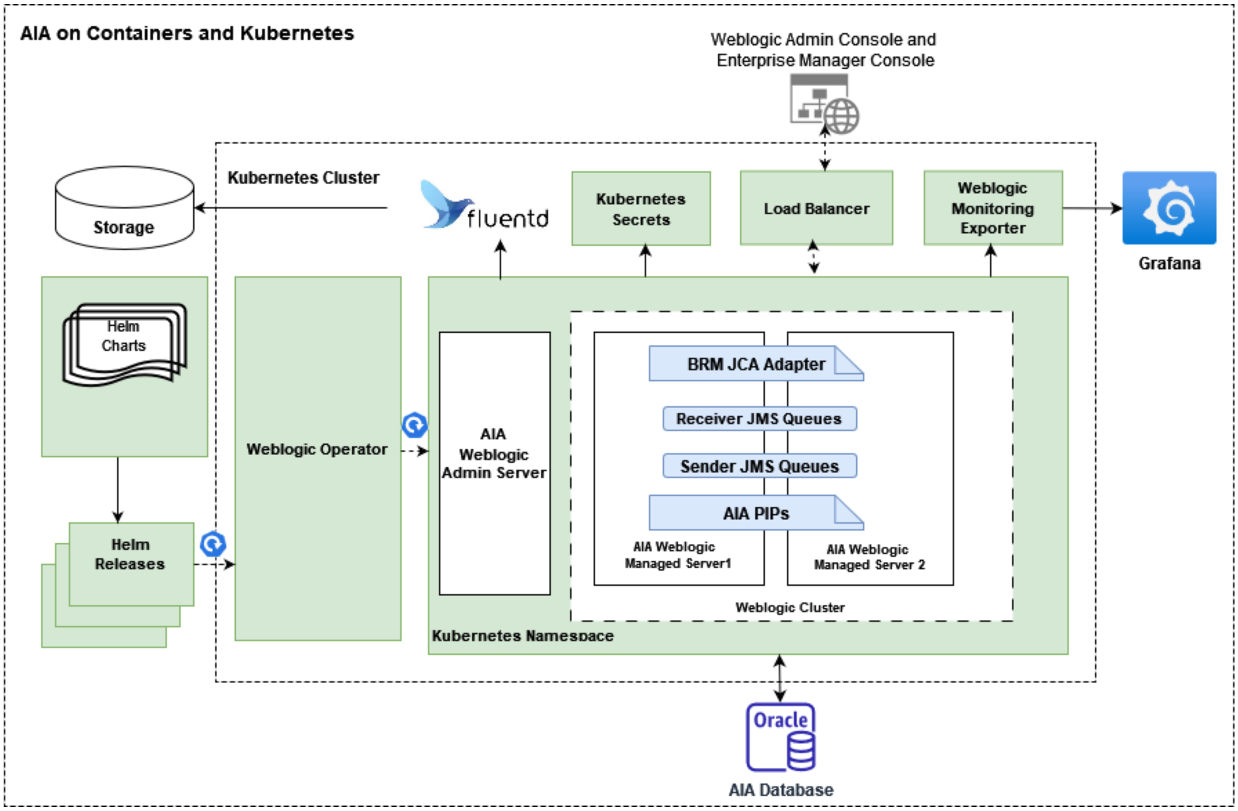
- Siebel CRM Cloud Native 23.03
- Oracle Communications Order and Service Management (OSM) Cloud Native 7.4.1 Patch 10
- Oracle Communications Billing and Revenue Management (BRM) Cloud Native 12 Patch Set 8

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 on Containers and Kubernetes



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 2023 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.

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/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 12.3.1 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-12.3.1.0.0.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 4.0.4, use branch `release/23.1.2` in the git clone command.

```
$ cd $AIA_DIR
$ git clone https://github.com/oracle/fmw-kubernetes.git -b release/
23.1.2
$ export WORKDIR=$AIA_DIR/fmw-kubernetes/OracleSOASuite/kubernetes
```

Refer to the Oracle SOA Suite documentation available at: https://oracle.github.io/fmw-kubernetes/23.1.2/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_12214_TEMPLATES section
##
## Ensure to include ',' in the SOA_12214_TEMPLATES ##
## Find below sample ##
    SOA_12214_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/23.1.2/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 -f Dockerfile -t aia-comms:12.3.1.0.0 .
```

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/*.

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/23.1.2/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/23.1.2/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/23.1.2/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/23.1.2/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/23.1.2/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/23.1.2/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/23.1.2/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/23.1.2/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/23.1.2/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 GutHub 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/23.1.2/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/23.1.2/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.

Deploying the AIA PIPs

This section provides instructions for deploying the AIA PIPs.

Configuring AIA PIP Credentials

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

- 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
```

Deploying the BRM JCA Adapter

Deploy the BRM JCA Adapter by downloading Patch 35353279 from My Oracle Support and apply it.

For instructions, see "Deploying and Configuring JCA Resource Adapter on Oracle WebLogic Server" in the *BRM JCA Adapter* guide.

Configuring and Deploying the AIA PIPs

The AIA PIPs Helm chart requires the following:

- aia-comms-pv-pvc helm chart
- AIA PIP credentials
- BRM JCA adapter deployed

> **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 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/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 the AIA PIPs:

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-aiapip \
       aia-comms-chart/charts/aia-comms-deploy-aiapip/ \
       --namespace namespace \
       --values ./aia-comms-chart/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 there are any errors encountered during the deployment, delete and recreate the domains before redeploying the PIPs. For instructions about deletion, see "Deleting the AIA Cloud Native Instance".

5

Performing Post-deployment Tasks

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

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 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-osm \
     aia-comms-chart/charts/aia-comms-certs/ \
     --namespace namespace \
     --values ./aia-comms-chart/values.yaml

# Restart domain
```

Integrating Applications with AIA Cloud Native

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

- Siebel CRM Cloud Native 23.03
- Billing and Revenue Management (BRM) Cloud Native 12 Patch Set 8
- Order and Service Management Cloud Native 7.4.1 Patch 10

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

Integrating Siebel Cloud Native

This section provides instructions for integrating Siebel cloud native with AIA cloud native.

To integrate Siebel cloud native with AIA cloud native:

1. Get the following JAR files:
   - From Siebel containers, get **siebel.jar** and **SiebelJI_enu.jar**.
   - From the WebLogic container, get **wlthint3client.jar**.
2. Log in to Siebel eCommunication Web UI as SADMIN user and update the JAVA64 profile parameters to include the three JAR files:

   > **Note:**
   >
   > Ensure that the path **/sfs/aiacn/jms** is a persistent store so that the files are retained after the pod restarts.

   ```
   /sfs/aiacn/jms:/sfs/aiacn/jms/Siebel.jar:/sfs/aiacn/jms/
   SiebelJI_enu.jar:/sfs/aiacn/jms/wlthint3client.jar:.
   ```

3. Relocate the JAR files:
   a. Connect to the Siebel SES pod.
   b. Create a folder with the same name as what is listed in step 2 (/sfs/aiacn/jms), where /sfs is a shared persistent folder, and then copy the 3 JAR files into the folder.
4. In the same folder, create the **jndi.properties** file and copy the following text into it:

> **Note:**
>
> Ensure that the AIACN t3 channel URL and user name and password are set with the correct values.

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
java.naming.provider.url=t3://
soa_cluster_service_name.soa_namespace.svc.cluster.local:soa_service
_cluster_port
java.naming.security.principal=soa_console_username
java.naming.security.credentials=soa_console_password
```

5. Copy the three JAR files into the Apache TOMCAT **/siebel/mde/applicationcontainer/lib** folder.
6. Restart Apache TOMCAT server inside Siebel SES pod.
7. Configure Siebel Web Service in Siebel DB.
    a. Create the "update_siebel_ws.sql" SQL script with the following content:

```
UPDATE S_WS_PORT SET PORT_ADDRESS='jms://jms/aia/
AIA_SALESORDERJMSQUEUE@jms/aia/COMMS_SUBMITORDER_CONSUMER',
PORT_TRANSPORT='JMS' WHERE NAME='SWISubmitOrderPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='jms://jms/aia/
AIA_SALESORDERJMSQUEUE@jms/aia/COMMS_SUBMITORDER_CONSUMER',
PORT_TRANSPORT='JMS' WHERE NAME='SWISubmitOrder_o2cPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='jms://jms/aia/
AIA_SALESORDERJMSQUEUE@jms/aia/COMMS_SUBMITORDER_CONSUMER',
PORT_TRANSPORT='JMS' WHERE NAME='SWISubmitQuote_o2cPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='jms://jms/aia/
AIA_SPECIALRATINGJMSQ@jms/aia/COMMS_SPECIALRATINGLIST_CONSUMER',
PORT_TRANSPORT='JMS' WHERE NAME='SWISpecialRatingListPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='jms://jms/aia/
AIA_CMUREQADJIOJMSQUEUE@jms/aia/COMMS_ADJUSTMENT_CONSUMER',
PORT_TRANSPORT='JMS' WHERE NAME='SWICreateAdjustmentPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/
soa-infra2/services/default/AccountBalanceSiebelCommsReqABCS/
AccountBalanceSiebelCommsReqABCS_ep' WHERE
NAME='_soap_AccountBalanceSiebelCommsReqABCS_AccountBalanceSiebel
CommsReqABCS';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/
soa-infra2/services/default/AdjustmentSiebelCommsReqABCS/
AdjustmentSiebelCommsReqABCS_ep' WHERE
NAME='AdjustmentSiebelCommsReqABCSPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/
soa-infra2/services/default/InvoiceSiebelCommsReqABCS/
InvoiceSiebelCommsReqABCS_ep' WHERE
NAME='_soap_InvoiceSiebelCommsReqABCS_InvoiceSiebelCommsReqABCS';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/
soa-infra2/services/default/PaymentSiebelCommsReqABCS/
PaymentSiebelCommsReqABCS_ep' WHERE
NAME='PaymentSiebelCommsReqABCSPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/
```

```
soa-infra2/services/default/UnbilledUsageSiebelCommsReqABCS/
UnbilledUsageSiebelCommsReqABCS_ep' WHERE
NAME='_soap_UnbilledUsageSiebelCommsReqABCS_UnbilledUsageSiebelCommsRe
qABCS';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/soa-
infra2/services/default/SyncCustomerSiebelEventAggregator/Client'
WHERE NAME='SyncCustomerSiebelEventAggregatorPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://aiacn_hostname:port/soa-
infra2/services/default/UpdateCreditAlertSiebelCommsReqABCSImpl/
UpdateCreditAlertSiebelCommsReqABCSImpl' WHERE
NAME='UpdateCreditAlertSiebelCommsReqABCSImplServicePort';
SET ESCAPE ON;
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/eai/enu/start.swe?
SWEExtSource=SecureWebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='SWIOrderUpsert';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/eai/enu/start.swe?
SWEExtSource=SecureWebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='SWI Product Attribute Import';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/eai/enu/start.swe?
SWEExtSource=SecureWebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='SWI Product Class Import';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/eai/enu/start.swe?
SWEExtSource=SecureWebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='SWIProductImport';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/eai/enu/start.swe?
SWEExtSource=SecureWebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='SWIPromotionImport';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/eai/enu/start.swe?
SWEExtSource=SecureWebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='SWIUpsertQuote';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/app/eai/enu?
SWEExtSource=WebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='COMMSCancelOrderPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/app/eai/enu?
SWEExtSource=WebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='COMMSCustomServicesPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/app/eai/enu?
SWEExtSource=WebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='COMMSOrderUpsert';
UPDATE S_WS_PORT SET PORT_ADDRESS='https://
Siebel_hostname:Siebel_port/siebel/app/eai/enu?
SWEExtSource=WebService\&SWEExtCmd=Execute\&WSSOAP=1' WHERE
NAME='COMMSSubmitBillingOrder';
commit;
quit;
```

   b. Connect to Siebel DB and run the SQL script with DB user name and the corresponding password.

8. Configure the Siebel repository in Siebel DB:

   a. Create the "update_siebel_repository.sql" SQL script with the following contents:

```
UPDATE S_SYS_PREF SET VAL='TRUE' WHERE SYS_PREF_CD='Enable AIA
Comms';
UPDATE S_SYS_PREF SET VAL='TRUE' WHERE SYS_PREF_CD='Enable AIA
MDM';
UPDATE S_SYS_PREF SET VAL='TRUE' WHERE SYS_PREF_CD='Enable AIA
Testing';
UPDATE S_SYS_PREF SET VAL='FALSE' WHERE SYS_PREF_CD='Enable AIA
Utility';
UPDATE S_SYS_PREF SET VAL='No' WHERE SYS_PREF_CD='Enable
Promotion Group';
UPDATE S_SYS_PREF SET VAL='/siebel/mde/siebsrvr/temp/
OrderBackup/' WHERE SYS_PREF_CD='AIA Order Backup Path';
UPDATE S_SYS_PREF SET VAL='Yes' WHERE SYS_PREF_CD='Enable
Promotion Group';
UPDATE S_SYS_PREF SET VAL='Y' WHERE SYS_PREF_CD='Promotion Group
Compatibility';
commit;
quit;
```

   b. Connect to Siebel DB and run the SQL script with DB user name and the corresponding password.

9. Configure EAI File Transfer Folder:

   a. Connect to the SES pod of the Siebel cloud native instance.

   b. Run the following two Siebel commands respectively and set "EAIFileTransportFolders" with the created "OrderBackup" sub-folder's full path as follows:

```
[aiacn_pod-0:/siebel/mde]#srvrmgr /g cgw-aiacn-0.ses-
aiacn.siebel-cn.svc.cluster.local:2320 /e aiacn /u username /p
password /c "change ent param EAIFileTransportFolders=/
siebel/mde/siebsrvr/temp/OrderBackup"


srvrmgr> change ent param EAIFileTransportFolders=/siebel/mde/
siebsrvr/temp/OrderBackup


srvrmgr> change ent param EAIFileTransportFolders=/siebel/mde/
siebsrvr/temp/OrderBackup for server aiacn_pod-0
```

   c. Restart the SES service using the kubectl command:

```
kubectl -n siebel_namespace, delete pod
Siebel_Enterprise_Server_pod_name-0
```

   d. After the pod is recreated and verified that it is running, do steps 5 and 6.

10. (Optional) Import products into Siebel cloud native by using the Siebel eCommunication application. Refer to Siebel CRM documentation for instructions.
11. Import Siebel cloud native SSL/TLS security certificates and configure AIA cloud native with the certificates. See "Installing SSL Certificates".
    a. Validate that keystore custom identity and custom trust are created successfully. To do this, log in to the Enterprise Manager Console for AIA and navigate to the Keystore section.
    b. Validate that Siebel trust certificate is available in the custom trust keystore in the Keystore section.
    c. Log in to the Weblogic Console of AIA cloud native. For each managed server, in the Keystore section, ensure the following:
       - **kss://system**/*custom_identity_keystorename* is displayed for Custom Identity Store.
       - **kss://system**/*custom_trust_keystorename* is displayed for Custom Trust Store.
    d. Validate **/u01/oracle/user_projects/domains/soainfra/bin/setDomainEnv.sh** with custom trust:
       i. Connect to any managed server pod.
       ii. Open the **/u01/oracle/user_projects/domains/soainfra/bin/setDomainEnv.sh** using the vi tool.
       iii. Validate that **-Djavax.net.ssl.trustStore=kss://system**/*custom_trust_keystorename* **-Djavax.net.ssl.trustStoreType=kss -Djavax.net.ssl.keyStorePassword**=*password*" is configured in EXTRA_JAVA_PROPERTIES.
12. Configure Siebel credentials in the Enterprise Console of AIA cloud native.
    a. Navigate to the Credentials section and edit **participatingapplications.siebel.server.eai.password** to specify the username and password.
    b. Repeat step a. for **participatingapplications.siebel.server.db.password** if you want to change the Siebel DB credentials.
    c. Restart the AIA cloud native services by using the domain-lifecycle scripts.
13. Configure AIA cloud native MetaData with Siebel Business Unit ID, and PRICELIST.
    a. Log in to Siebel and get Siebel Business Unit Id by navigating to Organizations. Navigate to About Record, copy Row #, and update the **AIAConfigurationProperties.xml** file with it.
    b. If not already available, create a pricelist.
    c. Search for the pricelist you created.
    d. Update **AIAConfigurationProperties.xml** and **PRICELIST.dvm** for Default Pricelist.
    e. Repeat steps from **b.** to **d.** for Business Pricelist and Consumer Pricelist.
    f. Search for the Pricelist and get Row # if available. Navigate to All Price Lists, select the row that contains NA Pricelist, and copy the Row # from About Record.
    g. Update **AIAConfigurationProperties.xml** and **PRICELIST.dvm** for Default Pricelist.
    h. Repeat steps from **f.** and **g.** for Business Pricelist and Consumer Pricelist.

i. Log in to Enterprise Manager console for AIA and update Siebel end points, Price List IDs, and Business Unit IDs:

    i. Get the latest **AIAConfigurationProperties.xml** and **PRICElIST.dvm** files.

    ii. In Enterprise Manager console for AIA, navigate to the **MDS Configuration** section and export the zip file.

    iii. Go to **soa/configuration/default** folder of your zip and copy the **AIAConfigurationProperties.xml**.

    iv. Go to **apps/AIAMetaData/dvm** and copy **PRICELIST.dvm**.

    v. In the **AIAConfigurationProperties.xml** file, replace the Siebel URL and then save and close the file.

    vi. Update MDS with the updated **AIAConfigurationProperties.xml** file and restart AIA cloud native using domain-lifecycle scripts.

j. In the **PRICElIST.dvm** file, do the following:

    i. Update the value of <cell> element, which is immediately after DEFAULT Pricelist, with the copied Row # of NA Pricelist.

    ii. Change the value of <cell> element, which is immediately after Consumer Pricelist, with the copied Row # of Consumer Pricelist.

    iii. Change the value of <cell> element, which is immediately after Business Pricelist, with the copied Row # of Business Pricelist.

    iv. Save and close the file.

    v. Open **AIAConfigurationProperties.xml** and search for XML tags property by the name **Siebel.SEBL_01.PriceList.ID**.

    vi. Update all the values of **Siebel.SEBL_01.PriceList.ID** with the copied NA Pricelist Row #.

    vii. Save and close the file.

k. Update the Business Unit ID:

    i. In the **AIAConfigurationProperties.xml** file, search for XML tags property by the name **Siebel.SEBL_01.BusinessUnit**.

    ii. Update all the values of **Siebel.SEBL_01.BusinessUnit** with the copied **Default Organization Row #**.

    iii. Save and close the file.

l. Update MDS with the updated **PRICELIST.dvm** and **AIAConfigurationProperties.xml** files.

    i. Copy the **PRICELIST.dvm** and **AIAConfigurationProperties.xml** files to a shared folder of AIA cloud native.

    ii. Create a file with the name **UpdateMetaDataDP.xml** and add the following content:

```
<?xml version="1.0" standalone="yes"?>
<DeploymentPlan component="Metadata" version="3.0">
        <Configurations>
                <UpdateMetadataFile wlserver="fp"
mdslocation="soa/configuration/default/">
```

```
                                <fileset
dir="absolute_path_to_share_folder_where_the_files_are_copied"
                                <include
name="AIAConfigurationProperties.xml" />
                        </fileset>
                </UpdateMetadataFile>
                <UpdateMetadataFile wlserver="fp" mdslocation="/
apps/AIAMetaData/dvm">
                        <fileset
dir="absolute_path_to_share_folder_where_the_files_are_copied"
                                <include name="PRICELIST.dvm" />
                        </fileset>
                </UpdateMetadataFile>
        </Configurations>
</DeploymentPlan>
```

iii. Connect to any SOA or AIA cloud native managed server pod and run following command:

```
$ cd /u01/aiacn/comms_home/bin
$ source ./commsenv.sh
$ ant -f $MW_HOME/soa/aiafp/Install/AID/AIAInstallDriver.xml -
DDeploymentPlan=absolute_path_to_file_no_variables_please/
UpdateMetaDataDP.xml -DPropertiesFile=$DOMAIN_LOCATION/soa/aia/bin/
AIAInstallProperties.xml
```

iv. Restart AIACN domain services by running the domain-lifecycle scripts.

14. Enable the eai_enu application configuration using the Siebel Management Console. Refer to the Siebel Management Console documentation for instructions.

Integrating BRM Cloud Native

This section provides instructions for integrating BRM cloud native with AIA cloud native.

To integrate BRM cloud native with AIA cloud native:

1. Validate the BRM cloud native CM parameter by running the following command:

   > **Note:**
   >
   > Ensure that the BRM CM service is configured with the dnsName of the cluster, so that the CM service can be connected using the dnsName in the cluster.

   ```
   kubectl -n brmcn_namespace get deployment/cm -o yaml
   ```

   A sample output is as follows:

   ```
   - name: CM_DNS_NAME
     value: dns:cm.brmcn-ps8
   ```

2. Deploy the BRM JCA Adapter by downloading Patch 35353279 from My Oracle Support and apply it.

   - In the extracted files, in the **Oracle_BRMJCA_Adapter_RAR/META-INF** directory, update the value for the ConnectionString parameter with the same value of CM_DNS_NAME of the CM service of BRM cloud native.
   - In the **ra.xml** and **weblogic-ra.xml** files, set the properties appropriately. For details, see "Deploying and Configuring JCA Resource Adapter on Oracle WebLogic Server" in the *BRM JCA Resource Adapter* guide. In addition to the parameters listed in the *BRM JCA Resource Adapter* guide, set the following parameters with appropriate values for a single cluster. The values shown are samples.
     - **ConnectionString** : `"ip cm.brmcn-ps8 11960"`
     - **FailoverConnectionString** : `"root.0.0.0.1:`*`password`*`@cm.brmcn-ps8:11960"`
     - **UserName** : `"root.0.0.0.1"`
     - **Password** : *`password`*
     - **SslWalletLocation** : `"/u01/oracle/user_projects/domains/soainfra/servers/AdminServer/tmp/_WL_user/OracleBRMJCA15Adapter/wernpi/wallet"`
       Note: 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"`
   - Download **cwallet.sso** file from the CM pod of the BRM cloud native instance and re-package **ra.xml**, **weblogic-ra.xml** and **cwallet.sso** into **OracleBRMJCA15Adapter.rar**.
   - Deploy **OracleBRMJCA15Adapter.rar** into AIA cloud native Weblogic cluster using the Weblogic Admin console and ensure that it is in the "Active" state.

3. (Optional) Validate the connection between AIA cloud native and BRM cloud native by deploying the BRM JCA Adapter test client (Web application) and sending a test request with the test client Web UI. For more details, see "Testing JCA Resource Adapter Configuration and BRM Connectivity" in the *BRM JCA Resource Adapter* guide.

4. Enable notification and Product Sync in BRM cloud native.

   a. Ensure that the **fm_publish enable_publish** parameter is set to **1** in the CM pin.conf.

   b. Ensure that eai-java-server in cm pod Infranet.properties file has the same payload as payloadconfig_crm_sync.xml. This payload contains the required events (ProductInfoChange and DiscountInfo change) for generating the XML for EAI. Ensure that the DB is 0.0.9.7, which points to EAI. The DB entries mapping can be found in dm-oracle pin.conf.

   c. Ensure that in the dm-aq config file, ALL queues are enabled.

d. Ensure that notifications are enabled for the following:

```
/event/notification/price/products/modify
/event/notification/price/discounts/modify
/event/notification/price/sponsorships/modify
/event/customer/status
/event/notification/amt/AccountInfoChange
```

e. Restart the **cm** and **dm-oracle**, **dm-aq** and **dm-ifw-sync** pods of the BRM cloud native instance.

Integrating OSM Cloud Native

This section provides instructions for integrating OSM cloud native with AIA cloud native.

To integrate OSM cloud native with AIA cloud native:

1. Create a t3 channel in the AIA cloud native instance:
   - For a single cluster scenario, where in AIA cloud native, Siebel CRM cloud native, BRM cloud native, and OSM cloud native are deployed in the same cluster, create a t3 channel in the WebLogic Admin Console for AIA cloud native.
     a. Log in to AIA cloud native Weblogic Console.
     b. Navigate to the Servers section in the Domain Structure pane and then select a managed server (for example, select **soa_server1**).
     c. Navigate to the Channels tab in Protocols and create a new channel (for example, T3Channel).
     d. Specify the following:
        – Listen Address as **soainfra-cluster-soa-cluster.*namespace*.svc.cluster.local**.
        – External Listen Port.
     e. Ensure that 'Tunneling Enabled' is selected.
     f. Repeat steps **b.** to **e.** for managed server **soa_server2**.
   - For multiple clusters, where in AIA cloud native and OSM cloud native are deployed in different clusters, do the following:
     a. Connect to the AIA cloud native cluster and create an ingressroute that includes route rules for all common names and the respective ports.

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: aia-ingress
  namespace: namespace
spec:
  entryPoints:
  - web
  routes:
  - kind: Rule
    match: Host(`soa.domain_name.namespace.aia.org`)
    services:
```

```
  - name: soa_cluster_service_name
    port: soa_ms_port
    sticky:
      cookie:
        httpOnly: true
 - kind: Rule
  match: Host(`t3.domain_name.namespace.aia.org`)
  services:
  - name: soa_cluster_service_name
    port: soa_cluster_service_port
    sticky:
      cookie:
        httpOnly: true
 - kind: Rule
  match: Host(`admin.domain_name.namespace.aia.org`)
  services:
  - name: soa_admin_server_service_name
    port: soa_admin_server_port
    sticky:
      cookie:
        httpOnly: true
```

b. Apply the yaml file to create the ingressroute:

```
kubectl apply -f aia-ingress.yaml
```

c. Edit the AIA cloud native domain configuration to specify the following settings for integrating with the OSM cloud native instance:

> **Note:**
>
> If the previous node is cordoned, deleted, or repaved, update the values for `hostAliases` and change the IP addresses to new IP addresses of a working node.

```
spec:
.........
.........
 serverPod:
.........
.........
  hostAliases:
  - hostnames:
   - t3.instance.project.osm.org
   - instance.project.osm.org
   - admin.instance.project.osm.org
   ip: osm_node_IP_address
  - hostnames:
   - soa.soainfra.soa_namespace.aia.org
   - t3.domain_name.namespace.aia.org
   - admin.domain_name.namespace.aia.org
   ip: soa_node_IP_address
```

d. Edit the OSM cloud native domain configuration to specify the same settings as described in step 1.b for integrating with the AIA cloud native instance.

> **Note:**
>
> Ensure that the AIA cloud native and OSM cloud native instance pods restart automatically after steps b and c. If they do not, run the corresponding scripts to restart the AIA and OSM instances.

e. Create a t3 channel in the WebLogic Admin Console for AIA cloud native, ensuring that the External Listen Address is set as the hostname defined earlier. In addition, ensure that the **HTTP Enabled for This Protocol** option is selected.

2. Deploy the O2A cartridge into the OSM cloud native instance. For instructions, see *OSM Cloud Native Deployment Guide for Oracle Application Integration Architecture Cartridges* and "Deploying the Sample Cartridge" in *OSM Cloud Native Deployment Guide*.

3. Set the AIA cloud native SAF t3 value, created earlier, in the OSM project specification file as follows:

   - For single cluster, specify the following:

```
safConnectionConfig:
  - name: O2A_SAFImportedDestinations
    t3Url: t3://
soa_cluster_servicename.namespace.svc.cluster.local:soa_cluster_servic
eport
    secretName: osm_project_instance__saf_credentials_aia_secret_name
```

> **Note:**
>
> *osm_project_instance__saf_credentials_aia_secret_name* is the secret you created while setting up and deploying OSM.

   - For multiple clusters, specify the following:

```
safConnectionConfig:
  - name: O2A_SAFImportedDestinations
    t3Url: http://t3_hostname_in_ingressroute:t3_port_created
    secretName: osm_project_instance__saf_credentials_aia_secret_name
```

4. Run the corresponding scripts to deploy the O2A cartridge with the AIA cloud native configuration:

```
[opc@cn-deployment o2a_cartridge]$ ./shutdown_osm.sh ; ./
deploy_o2a_par.sh ; ./start_osm.sh
```

5. Log in to the Weblogic Admin Console for OSM cloud native and verify that the SAF setting is configured with the expected t3 Url value.

6. Log in to the OSM Task Web client and verify that the O2A cartridge is deployed.

7. In WebLogic Admin Console for OSM cloud native, copy the t3 URL dispalyed for **T3ClustChannel** for single cluster or the value of **T3Channel (HTTP)** for multi-cluster channel.
8. In the WebLogic Admin console for AIA cloud native, navigate to the JMS Modules page, for the **OSM** and **SOM** AIAJMSModules, set the URL test fields with the copied t3 URLs.
9. Restart the AIA cloud native domain services by using the domain-lifecycle scripts.
10. In WebLogic Admin Console, navigate to the Store and Forward Agents page. In the Remote Endpoints tab for OSM_SAFAgent, ensure that for each Remote Endpoint, the t3 URL is displayed for single-cluster environment.
11. Log in to Oracle Enterprise Manager Fusion Middleware Control for AIA cloud native, add the singleton property. See the *AIA Installation Guide* for instructions.
12. Configure the AIA queues to support JMS Priority. See the *AIA Installation Guide* for instructions.
13. Add the No Authentication security policy to the Product class service. See the *AIA Installation Guide* for instructions. Ensure that you select "QueryProductClassAndAttributesSCECommsReqABCSImpl" in the Service and References region.
14. Ensure that the JAVA_OPTIONS parameter `-Dweblogic.rjvm.allowUnknownHost=true` is added into AIA cloud native domain setting.
15. In Oracle Enterprise Manager Fusion Middleware Control, for SOA Infrastructure, set the **Callback Server URL** and **Server URL** common properties with the SOA infrastructure URL.

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/22.4.2/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/22.4.2/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 mostly 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".

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 logs from the following deployment pods:
   >
   > - /u01/shared/runconfig.log
   > - /u01/shared/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.
3. Test the order flow to check connectivity between Siebel, OSM, and BRM.

6

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

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-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/23.1.2/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/*
   ```

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/23.1.2/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/23.1.2/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/.

Upgrading Your AIA Cloud Native Deployment

To upgrade your AIA cloud native deployment, perform the procedures described in the *Oracle Fusion Middleware on Kubernetes* documentation at: https://oracle.github.io/fmw-kubernetes/23.1.2/soa-domains/patch_and_upgrade/.

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-deploy-aiapip 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:

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

3. Identify and resolve issues, if any.
4. 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/*
   ```

5. Uninstall the helm chart.

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

6. Reinstall the helm chart with updated parameters, if any.

Redeploying the AIA PIPs Helm Chart

The aia-comms-deploy-aiapip AIA helm chart requires the following:

- aia-comms-pv-pvc helm chart
- AIA PIP credentials
- BRM JCA Adapter deployment

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-deploy-aiapip-job
   ```

2. Get logs of the pod:

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

3. Identify and resolve issues, if any:
   - The logs might contain the "ERROR - Deployment found." error message. This error occurs if reinstallation of the aia-comms-deploy-aiapip helm chart is done over existing PV data.
   - The job pod might throw an error before reaching the "Configuring AIAPIPs deployment..." stage. If this happens, do not delete Oracle SOA Suite Domain or redeploy the domain. Resolve the error and perform steps 4 and 7.
   - If stage "Configuring AIAPIPs deployment..." or "Deploying AIAPIPs..." is already reached, the RCU schemas might have been updated by the time the error occurred. Therefore, overwriting of the AIA PV and the SOA PV data from the backup is not recommended since it may not solve the issue. Continue with the steps from 4 to 7 to resolve the issue.

4. Uninstall the helm chart:

   ```
   $ helm uninstall aia-comms-deploy-aiapip -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. You also do not need to delete the namespace for Oracle SOASuite Domain.
6. Redeploy the domain. See "Deploying AIA Cloud Native".
7. Install the aia-comms-deploy-aiapip helm chart.

Oracle SOASuite 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 SOASuite 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-deploy-aiapip** 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 6-1.

Table 6-1 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.

7

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 7-1 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 7-1 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 7-1 (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 7-2 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 7-2 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 7-3 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 7-3 Parameters for AIA PIPs

Parameter	Description
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.
Order2Cash_Siebel	Select true if want to install PIPs for Siebel.
SPLIT_XREF	Create Split XRef schema? Default is true.
Order2Cash_brm	Select true if want to install PIPs for BRM.

Table 7-3 (Cont.) Parameters for AIA PIPs

Parameter	Description
Order2Cash_osm	Select true if want to install PIPs for OSM.
DOMAIN_LOCATION	Weblogic Domain location for SOA
OSM_VERSION	OSM Version. For example, 7.4.1.0.0.
OSM_PROV_WL_JMS_QUEUE_ACCESS_PORT	OSM Provisioning Weblogic JMS Queue port
OSM_CFS_WL_JMS_QUEUE_ACCESS_HOST	OSM CFS Weblogic JMS Queue Host
OSM_CFS_WL_JMS_QUEUE_ACCESS_PORT	OSM CFS Weblogic JMS Queue port
OSM_PROV_WL_JMS_QUEUE_ACCESS_HOST	OSM Provisioning Weblogic JMS Queue Host
BRM_PRIMARY_CM_HOST	BRM CM Host Name
BRM_VERSION	BRM Server Version. For example, 12.2.0.0.
BRM_DB_HOST	BRM Database Host Name
BRM_AQ_DB_SID	BRM AQ Database System ID
BRM_PRIMARY_CM_PORT	BRM CM Port Number
BRM_AQ_QUEUE	BRM AQ Queue Name. For example, AQ_QUEUE.
BRM_DB_PORT	BRM Database Port Number
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*TNS_ADMIN : jdbc:oracle:thin:@*db_name*? TNS_ADMIN=*tns_admin_location*
SPM_PROXY_PORT	Siebel Session Pool Manager Port Number. Can be empty if one is not available.
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	System role name to create XRef Schema. Example: SYSDBA.
XREF_SCHEMA_JDBC_URL	XRef Schema JDBC URL. This should be same as SOA_DB_JDBC_URL.
SBL_VERSION	Siebel Server Version. Example: 21.2.0.0.
SBL_HOST	Siebel Host Name
SBL_PROTOCOL	Protocol to connect Siebel Server. Example: http://
SBL_LANG	Siebel local language. Example: enu
SBL_ENTERPRISE_SERVER_NAME	Siebel Enterprise Server Name. Example: SBA_82
SBL_PORT	Siebel Port, if siebel is running on Kubernetes, you can use Tomcat's exposed port number.
SBL_DB_HOST	Siebel Database Host Name
SBL_DB_PORT	Siebel Database Port Number

Table 7-3 (Cont.) Parameters for AIA PIPs

Parameter	Description
SBL_DB_SID	Siebel Database SID/Service Name
SBL_DB_JDBC_URL	Siebel 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* TNS_ADMIN : jdbc:oracle:thin:@*db_name*?TNS_ADMIN=*tns_admin_location*
SOA_WL_MS_PORT	SOA Managed Server Cluster port. 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	SOA Admin server port. This will be the port number of the SOA admin server's service inside the Kubernetes SAO namespace.
SOA_WL_DOMAIN_NAME	Unique name that is used to identify this particular domain where AIA needs to be installed.
SOA_WL_MS_NAME	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	SOA Database JDBC URL JDBC URL is used to connect and configure corresponding 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	SOA managed server host name. 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	SOA Admin server host name. This will be the service name of admin server inside Kubernetes SOA namespace.

Parameters for AIA Certificates

Table 7-4 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 7-4 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.

Table 7-4 (Cont.) Parameters for AIA Certificates

Parameter	Description
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 7-5 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 7-5 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

Table 7-5 (Cont.) Parameters for AIA PIPs Credentials

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

8

Securing Your AIA Cloud Native Deployment

This chapter describes security considerations for your AIA cloud native deployment.

Oracle AIA offers its cloud native deployment on Kubernetes 1.25. Based on the variety of customizations and plugins you have for your Kubernetes platform, you need to consider all possible security risks and have a mitigation plan in place.

General Security Considerations

Consider the following general security guidelines:

- While the **values.yaml** file of the Helm charts can be stored in versioning systems, it is recommended that you do not use it to save sensitive information such as application credentials. Instead, use Kubernetes secrets.
- Use the sample scripts provided with the cloud native toolkit for creating secrets to maintain credentials for various applications such as OSM, Siebel, BRM, SOA, AIA and RCU.
- Use the sample scripts for secrets and store them in a vault that has strong encryption.
- Secure your Kubernetes secrets by using strong encryption, instead of a default base64 encryption.
- Use Kubernetes RBAC on minimum privileges policy and restrict kubectl get, list, and watch privileges for secrets, pods, logs, and services.
- Use Kubernetes RBAC on minimum privileges policy and restrict resource access to pods such as secrets and network.
- Consider Kubernetes general security guidelines. For details, see Kubernetes documentation available at: https://kubernetes.io/docs/setup/best-practices/enforcing-pod-security-standards/.

Also refer to the *AIA Security Guide* for other security considerations.