Oracle® Communications Digital Business Experience

Solution Deployment Guide

Release 26.4
G47938-01
May 2026

ORACLE

Oracle Communications Digital Business Experience Solution Deployment Guide, Release 26.4

G47938-01

Contents

4 Performing Post-Deployment Tasks

5 Troubleshooting

6 Downloading and Deploying the Reference Solution

About This Content

This guide provides an overview of the ways to deploy the Oracle Communications Digital Business Experience Solution. This guide also describes the system requirements and procedures for deploying the solution and its components.

Audience

This document is intended for cloud operations administrators and other personnel who are responsible for deploying, configuring, managing, and maintaining the Oracle Communications Digital Business Experience solution.

1

Solution Overview and Architecture

The chapter provides an overview of Oracle Communications Digital Business Experience deployed in a cloud native environment.

You can deploy the solution on Oracle Cloud Infrastructure (OCI) and in other cloud native environments. When deployed on OCI, the cloud native solution uses OCI Container Engine for Kubernetes (OKE).

About the Solution

Digital Business Experience is a pre-integrated, end-to-end, digital business support system (BSS) for managing experiences and revenue at every stage of your journey. The solution enables you to:

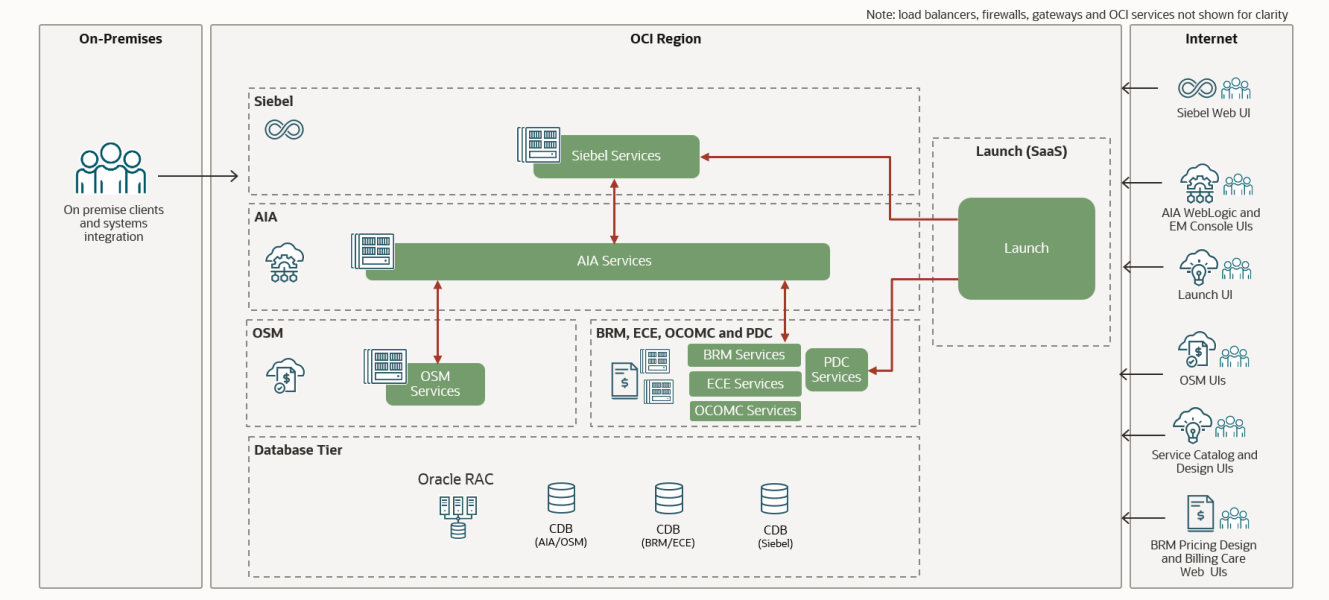
- Use Launch to intuitively design multi-dimensional offers faster with an intelligence-enabled, GUI-based enterprise product catalog.
- Use Siebel CRM to capture customer and partner orders across assisted and unassisted channels.
- Use Oracle Communications Order and Service Management (OSM) to capture, validate, and deliver orders across channels quickly and to dynamically orchestrate the order fulfillment of both subscriber orders and service orders.
- Use Oracle Communications Billing and Revenue Management (BRM) with Oracle Communications Elastic Charging Engine (ECE) and Oracle Communications Offline Mediation Controller to mediate, charge and bill for multi-generational communications services (2G-5G mobile, fixed, satellite) at scale.
- Use Siebel CRM to provide intelligent, personalized, and proactive subscriber care across traditional, digital, assisted, and unassisted channels.

Oracle Application Integration Architecture (Oracle AIA) is the solution's integration framework that provides pre-built integrations and process flows between Siebel, OSM, and BRM using standard integration patterns, business processes, orchestration logic, and common objects and services to connect Oracle applications.

Digital Business Experience Architecture

Figure 1-1 illustrates a functional deployment view of the solution within a single OCI region, showing the use of Oracle RAC in data tier with container databases for AIA, OSM, BRM, ECE, and Siebel CRM.

Figure 1-1 Digital Business Experience Functional Architecture



> **Note**
>
> The above diagram is a functional representation of the Digital Business Experience architecture and does not include infrastructure components such as load balancers, firewalls, gateways and OCI services for clarity.

In the above conceptual reference architecture, the solution is deployed on Oracle Cloud Infrastructure Container Engine for Kubernetes (OKE). The solution applications have dedicated clusters.

You can use this architectural blueprint as a starting point for an end-to-end Cash to Care communications application solution that leverages the benefits of OCI. Your actual deployment architecture depends on your needs and may differ from this architecture.

> **Note**
>
> While the conceptual reference architecture above shows the Digital Business Experience Solution deployed on OCI, you can deploy the solution on Oracle Cloud Native Environment (OCNE), Cloud Native Computing Foundation (CNCF), and other cloud environments as well.

The Oracle RAC database is privately accessible from the Kubernetes worker nodes using the Oracle Base Database Service or the Exadata Cloud Service. This architecture depicts a typical approach to distributing the solution applications across Container Databases (CDBs). For more information about the latest supported database versions, refer to the compatibility matrix of each application.

Oracle Data Guard or Active Data Guard can be used for replication to standby databases.

A bastion host can be configured in a public subnet to allow access to the solution's worker nodes from your network. For example, this can be done using SSH. The web clients for

Siebel, BRM and OSM and external integrations connect to load balancers through the Internet gateway. Additional security rule enforcement can be provided by Oracle Cloud Infrastructure Web Application Firewall (WAF) for Internet traffic.

You can use an ingress controller behind an external load balancer to expose solution services outside of the Kubernetes cluster and use that to communicate with the solution application components. The ingress controller monitors the ingress objects and acts on the configuration embedded in these objects to expose HTTP and T3 services to the external network. The load balancer provides a highly reliable, single-point access into the services exposed by the Kubernetes cluster.

Database backups can use the Object Storage Service, and block storage is required for the worker node OS boot volumes and root filesystems. OKE Persistent Volumes (PVs) can use NFS-based persistence for shared storage. Block volumes can be used for all PVs that do not require shared storage.

> **Note**
>
> You should share an external public load balancer with multiple backends across the solution component applications.

Considerations for Deploying the Solution

You need to consider the following for deploying the solution:

- Detailed deployment architecture and performance: Actual production deployment architectures and solution sizing will vary depending on many factors, which should be discussed with Oracle or your implementation partner prior to and during the deployment project design phase.
- Availability and resiliency: For simplicity, this architecture depicts a single availability domain. Advanced Kubernetes features (such as pod anti-affinity) enable deployments on OCI regions with multiple availability domains to maximize availability. In addition, deployments can be split across regions for geographic redundancy and disaster recovery scenarios. In such models, data replication across RAC instances can be provided using Active Data Guard. The above reference architecture does not consider individual application availability scenarios and these must be factored into your production deployment architecture.

2

Solution Requirements

This chapter describes the requirements for deploying the Digital Business Experience Solution.

Prerequisites for Deploying the Solution Components

Refer to the Digital Business Experience Compatibility Matrix for information about the required solution components.

For information on the prerequisites for your deployment of the solution components, refer to the following:

- For Oracle Communications Launch Cloud Service: Launch Overview in *Launch Cloud Service User's Guide*
- For Oracle Communications Billing and Revenue Management Cloud Native: About Configuring and Deploying Your BRM Cloud Native Environment in *BRM Cloud Native Installation Guide*
- For Oracle Analytics Server (OAS): Before You Begin in *Oracle Analytics Server Documentation*
- For Oracle Communications Service Catalog and Design (SCD): Planning and Validating Your Cloud Environment in *SCD Solution Designer Installation Guide*
- For Oracle Communications Order and Service Management (OSM): Planning and Validating Your Cloud Native Environment in *Cloud Native Deployment Guide*
- For Oracle Communications Order-to-Activate (O2A): Generating the OSM Cloud Native Artifacts for the Order-to-Activate Solution in *Order-to-Activate Solution Guide*
- For Oracle Siebel CRM: Requirements for Installing and Configuring Siebel CRM in *Siebel Installation Guide*
- For Oracle Data Integrator (ODI): Preparing to Install and Configure Oracle Data Integrator in *Oracle Data Integrator Installation Guide*
- For Oracle Application Integration Architecture (AIA): Prerequisites for Your AIA Cloud Native Deployment in *AIA Cloud Native Deployment Guide*

3

Deploying and Validating the Solution Components

This chapter provides information about deploying and validating the solution components.

Assumptions for Deploying the Solution

Before deploying the solution, ensure that:

- You understand cloud native technologies and deployment tasks.
- You have access to all of the appropriate systems.
- You know how to operate and administer the operating systems (for example, Linux) you will use in the solution.
- You understand the resources (for example, Oracle Cloud Infrastructure (OCI)) and applications (for example, the database) that will be included in the solution.
- All of the prerequisites for the component applications have been met.

See the following topics for details about the activities you perform to deploy and validate the solution components:

- See Preparing the Cluster for more information.
- See Deploying the Solution Components for more information.
- See Validating the Solution Components for more information.

Preparing the Cluster

Before deploying the Digital Business Experience solution, you must prepare the cluster components. Information is provided about key infrastructure elements, including a Virtual Cloud Network (VCN), a Public Bastion for secure access, a Kubernetes Cluster using Oracle Kubernetes Engine (OKE), and an Oracle Base Database Service. Each component serves a distinct purpose and together they form a robust cloud environment suitable for a range of enterprise applications.

Deploy the following key infrastructure elements in the sequence provided below:

- **VCN**: Helps establishing a secure, isolated, and customizable virtual network, which serves as the foundation for deploying various cloud resources. VCN provides full control over the network architecture, including IP address ranges, subnets, route tables, and security lists. See Deploying the Virtual Cloud Network for more information.
- **Public Bastion**: Allows secure access to resources within the VCN without exposing those resources to the public internet. The Public Bastion acts as a gateway for administrators and authorized users to manage resources in private subnets, enhancing the security of the environment. See Deploying the Public Bastion for more information.
- **OKE**: Provides a scalable platform for deploying, managing, and automating the operations of containerized applications, ensuring high availability and fault tolerance. See Deploying Using an Oracle Kubernetes Engine Environment for more information.

- **Oracle Base Database Service**: Provides reliable, high-performance data management, with configurations optimized for the specific needs of the application workloads. See Deploying the Oracle Base Database Service for more information.
- **Kubernetes cluster using Oracle Cloud Native Environment (Oracle CNE)**: The Kubernetes cluster is set up using the Oracle CNE Command Line Interface (CLI) using the `libvirt` provider. Oracle CNE designed for running cloud native applications at scale, providing a standardized environment for managing microservices, containers, and workloads across various cloud providers. Its benefits include improved agility, scalability, and cost efficiency by automating infrastructure management and simplifying deployment and orchestration of containerized applications. See Deploying Using an Oracle CNE for more information.
- **Cloud Native Computing Foundation (CNCF)**: CNCF is an open-source software foundation that promotes the adoption of cloud-native computing. It is a subsidiary of the Linux Foundation. CNCF hosts and supports projects, such as Kubernetes, Prometheus, and Envoy, which are essential components of many modern cloud-native architectures. See Deploying Using a Cloud Native Computing Foundation Environment for more information.

Deploying the Virtual Cloud Network

Perform the following tasks to deploy the VCN:

- Creating a VCN
- Creating a Service Gateway, NAT Gateway, and Internet Gateway in the VCN
- Creating Public and Private Route Tables in the VCN
- Creating Public, Private, and LB Subnets in the VCN
- Modifying the Security List in the VPN

Creating a VCN

Creating the Virtual Cloud Network (VCN) helps you to define the IP address space and create the VCN.

To create a VCN within Oracle Cloud Infrastructure (OCI):

1. Log in to Oracle Cloud Infrastructure (https://www.oracle.com/cloud/) using your credentials.
   The OCI home page opens.
2. From the **Navigation** pane, select **Networking**, and then click **Virtual cloud networks**.
   The **Create a Virtual Cloud Network** page opens. Provide the following details:
   - In the **Name** field, enter the desired VCN name. For example, `dbe-vcn`.
   - In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
   - In the **CIDR Block** field, enter the Classless Inter-Domain Routing (CIDR) in the **0.0.0.0/0** format. For example, `10.0.0.0/28`.
3. Click **Create VCN**.

Creating a Service Gateway, NAT Gateway, and Internet Gateway in the VCN

Creating Subnets helps you to set up public and private subnets within the VCN, ensuring they are logically separated for different types of resources.

To create a Service Gateway:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**, then select **Virtual cloud networks**, and then select your VCN.
3. On the **Resources** pane, select **Service Gateways**, and then click **Create Service Gateway**.
4. In the **Create Service Gateway** page, provide the following details:
   - In the **Name** text box, enter the service gateway name. For example, `serviceGW`.
   - In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
   - In the **Services** text box, enter the required services. For example, `All IAD Services in Oracle Services Network`.
   - Click **Create Service Gateway**.

To create a Network Address Translation (NAT) Gateway:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**, then select **Virtual cloud networks**, and then select your VCN.
3. On the **Resources** pane, select **NAT Gateways**, and then click **Create NAT Gateway**.
4. In the **Create NAT Gateway** page, provide the following details:
   - In the **Name** text box, enter the NAT gateway name. For example, `NATGW`.
   - In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
   - Select the **Ephemeral Public IP Address** option.
   - Click **Create NAT Gateway**.

To create an Internet Gateway:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**, then select **Virtual cloud networks**, and then select your VCN.
3. On the **Resources** pane, select **Internet Gateways**, and then click **Create Internet Gateway**.
4. In the **Create Internet Gateway** page, provide the following details:
   - In the **Name** text box, enter the internet gateway name. For example, `InternetGW`.
   - In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
   - Click **Create Internet Gateway**.

Creating Public and Private Route Tables in the VCN

Configuring route tables helps you to set up route tables for network traffic management.

To create a public route table:

1. Log in to OCI.

2. From the **Navigation** pane, select **Networking**, then select **Virtual cloud networks**, and then select your VCN.
3. On the **Resources** pane, select **Route Tables**, and then click **Create Route Table**.
4. In the **Create Route Table** page, provide the following details:
   - In the **Name** text box, enter the public route table name. For example, `rt_publicsubnet`.
   - In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
   - Click **Create Route Table**.
5. To attach an internet gateway to the created public route table:
   - From the **Navigation** pane, select **Networking**.
   - Select **Virtual cloud networks**, then select your VCN.
   - On the **Resources** pane, select your public route table, and then click the **Add Route Rules** button.
     The **Add Route Rules** page opens.
   - From the **Target Type** drop-down list, select the **Internet Gateway** option.
     The **Destination CIDR Block** and **Target Internet Gateway in Compartment** fields are auto-populated.
   - Click **Add Route Rules**.

To create a private route table:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**.
3. Select **Virtual cloud networks**, then select your VCN.
4. On the **Resources** pane, select **Route Tables**, and then click the **Create Route Table** button.
5. In the **Create Route Table** page, provide the following details:
   - In the **Name** text box, enter the public route table name. For example, `rt_privatesubnet`.
   - In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
   - Click **Create Route Table**.
6. To attach a NAT gateway to the created private route table:
   - From the **Navigation** pane, select **Networking**.
   - Select **Virtual cloud networks**, then select your VCN.
   - On the **Resources** pane, select your private route table, and then click the **Add Route Rules** button.
     The **Add Route Rules** page opens.
   - From the **Target Type** drop-down list, select the **NAT Gateway** option.
     The **Destination CIDR Block** and **Target NAT Gateway in Compartment** fields are autopopulated.
   - Click **Add Route Rules**.
7. To attach a service gateway to the created private route table:

- From the **Navigation** pane, select **Networking**.
- Select **Virtual cloud networks**, then select your VCN.
- On the **Resources** pane, select your private route table, and then click the **Add Route Rules** button.
  The **Add Route Rules** page opens.
- From the **Target Type** drop-down list, select the **Service Gateway** option.
  The **Destination Service** and **Target Service Gateway in Compartment** fields are autopopulated.
- Click **Add Route Rules**.

Creating Public, Private, and LB Subnets in the VCN

To create a public subnet:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**.
3. Select **Virtual cloud networks**, then select your VCN.
4. On the **Resources** pane, select **Subnets**, and then click the **Create Subnet** button.
   The **Create Subnet** page opens.
5. In the **Name** text box, enter the public subnet name. For example, `Public subnet`.
6. In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
7. In the **Subnet Type** drop-down list, select **Regional (Recommended)**.
8. In the **IPv4 CIDR Block** text box, enter the CIDR in the **0.0.0.0/0** format. For example, `10.0.0.0/28`.
9. From the **Route Table Compartment** drop-down list, select public route table you created.
10. In the **Subnet Access** drop-down list, select **Public Subnet**.
11. Under the **DNS Resolution** field, select **Use DNS hostnames in this Subnet**.
    The **DNS Label** and **DNS Domain Name** fields are autopopulated.
12. Click **Create Subnet**.

To create a private subnet:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**.
3. Select **Virtual cloud networks**, then select your VCN.
4. On the **Resources** pane, select **Subnets**, and then click **Create Subnet**.
   The **Create Subnet** page opens.
5. In the **Name** text box, enter the private subnet name. For example, `Private subnet`.
6. In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
7. In the **Subnet Type** drop-down list, select **Regional (Recommended)**.
8. In the **IPv4 CIDR Block** text box, enter the CIDR in the **0.0.0.0/0** format. For example, `10.0.0.0/24`.
9. In the **Route Table Compartment** drop-down list, select the created private route table.

10. In the **Subnet Access** drop-down list, select **Private Subnet**.
11. Under the **DNS Resolution** field, select **Use DNS hostnames in this Subnet**. The **DNS Label** and **DNS Domain Name** fields are autopopulated.
12. Click **Create Subnet**.

To create a private Load Balancer subnet:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**.
3. Select **Virtual cloud networks**, then select your VCN.
4. On the **Resources** pane, select **Subnets**, and then click the **Create Subnet** button. The **Create Subnet** page opens.
5. In the **Name** text box, enter the private subnet name. For example, `Private LB subnet`.
6. In the **Create In Compartment** text box, enter the compartment name. For example, `demo`.
7. In the **Subnet Type** drop-down list, select **Regional (Recommended)**.
8. In the **IPv4 CIDR Block** text box, enter the CIDR in the **0.0.0.0/0** format. For example, `10.0.2.0/28`.
9. In the **Route Table Compartment** drop-down list, select the created private route table.
10. In the **Subnet Access** drop-down list, select **Private Subnet**.
11. Under the **DNS Resolution** field, select the **Use DNS hostnames in this Subnet**. The **DNS Label** and **DNS Domain Name** fields are autopopulated.
12. Click **Create Subnet**.

Modifying the Security List in the VPN

You can add, modify, or terminate the ingress and egress rules of your VCN.

To modify the Security List of your VCN:

1. Log in to OCI.
2. From the **Navigation** pane, select **Networking**, then select **Virtual cloud networks**, then select your VCN, and then from the **Resources** pane, select **Security List Details**. The **Default Security List** page of your VCN opens.
3. To add an Ingress rule:
   - On the **Resources** pane, select **Ingress Rules**.
   - Click **Add Ingress Rules**.
   - Provide the required details and then click **Save**.
4. To add an Egress rule:
   - On the **Resources** pane, select **Egress Rules**.
   - Click **Add Egress Rules**.
   - Provide the required details and then click **Save**.
5. To modify the security list details, select any one rule from the ingress or egress rules list, and then click **Edit**.

6. Modify the required details, and then click **Save**.

Deploying the Public Bastion

To deploy the public bastion:

1. Ensure that you are connected to the OCNA VPN before starting this procedure.
2. Log in to OCI.
3. From the **Navigation** pane, select **Compute**, then select **Instances**, and then click **Create instance**.
   The **Create compute instance** page opens, provide the following details:
   - **Name**. For example, `Public-bastion`.
   - **Image and Shape**
   - **VCN**
   - **Boot volume**
   - **SSH public keys**

   > **Note**
   >
   > You must create your own public keys.
4. Click **Create**. The public bastion is created and a public IP is generated.

You can verify the public bastion access using the SSH key.

To verify the public bastion:

1. Open Git Bash on your local machine.
2. Run the following command:

   ```
   ssh -i /path/to/private-key opc@bastion-public-ip
   ```

   Replace *`/path/to/private-key`* with the path to your SSH private key.

   Replace *`bastion-public-ip`* with the bastion's public IP address.

If correct values are provided in the above command, you will get access to the public bastion.

Deploying the Oracle Base Database Service

To deploy the Oracle Base database service:

1. Log in to OCI.
2. From the **Navigation** pane, select **Oracle Database**, then select **Oracle Base Database**, then select **DB Systems**, and then click the **Create DB System** button.
   A **Create DB System** page opens.
3. Click DB system information, and provide the following details:
   a. From the **Select a compartment** drop-down list, select the appropriate compartment.
   b. In the **Name your DB System** text box, enter the desired name. For example, `dbe-database`.

c. From the **Select an availability domain** list, select an appropriate domain.

d. In the **Configure shape** pane, attach the shape details.

e. From the **Configure storage** pane, select the appropriate storage management software type.

f. In the **Configure the DB System** field, enter the host name.

g. Click **Save**.

4. Click **Database information**, and configure the administrator credentials as follows:

a. In the **Database name** text box, enter the desired name. For example, `DB0822`.

b. Click the **Change database image** button in the **Database image** field, browse to the appropriate image, and upload the image.

c. In the **Create administrator credentials** pane, provide the following details:

- In the **Password** text box, enter the desired password.

> **Note**
>
> The **Username** field will be auto-populated and is a read-only field.

- In the **Confirm Password** text box, enter the password again.
- Select **Use the administrator password for the TDE wallet**.

d. From the **Configure database backups** pane, select **Enable automatic backups**.

5. Click **Create DB System**. The Oracle Base Database is created.
The **Database system information** page opens.

6. From the **Resources** pane, click **Nodes**, and make a note of the Private IP address.

Verifying and Accessing the Database

To verify and access the Oracle Base Database:

1. Open a command terminal window.
2. Log in to public bastion using SSH. See Deploying the Public Bastion for more information.
3. Run the following command inside the public bastion:

```
ssh opc@Private_IP_Address
```

Replace *`Private_IP_Address`* with the Oracle Database private IP address generated after it was created.

4. Run the following commands to verify the access to Oracle Database:

```
sudo su -oracle
sqlplus / as sysdba
```

Creating a Pluggable Database within the Oracle Base Database

To create a pluggable database (PDB):

1. Log in to OCI.

2. From the **Navigation** pane, select **Oracle Database**, then select **Oracle Base Database Service**, then from the **Compartment** drop-down list, select your compartment, and then select your **Oracle Database System**.
3. From the **Resources** pane, click **Databases**, and then select the existing database from the table.
4. From the **Resources** pane, click **Pluggable Databases**, and then click **Create pluggable database**.
5. Enter values for the following fields:
   - **PDB name**
   - **PDB admin password**

     > **Note**
     >
     > You must always refer to the OCI vault for the password.

   - **Confirm PDB admin password**
   - **TDE wallet password of database**
6. Select **Unlock the PDB admin account**.
7. Click **Create pluggable database**.

Deploying the Kubernetes Cluster

Learn about deploying the Kubernetes cluster in your Oracle Communications Digital Business Experience solution.

You can deploy the Kubernetes cluster using any one of the following environment types:

- Deploying Using an Oracle Kubernetes Engine Environment
- Deploying Using an Oracle CNE
- Deploying Using a Cloud Native Computing Foundation Environment

Deploying Using an Oracle Kubernetes Engine Environment

This section provides detailed instructions about creating a Kubernetes cluster using Oracle Kubernetes Engine (OKE), configuring the node pools, setting up the Kubernetes access, and verifying the access.

To deploy the Kubernetes cluster using OKE:

1. Log in to OCI.
2. From the **Navigation** pane, select **Developer Services**, and then select **Kubernetes Cluster**.
3. Click the **Create cluster** button.
   The **Create cluster (custom)** page opens. Provide the following details:
   a. In the **Name** text box, enter the desired cluster name. For example, `dbe-cluster`.
   b. In the **Compartment** text box, enter the name of the container. For example, `demo`.
   c. In the **Kubernetes version** text box, enter the **Kubernetes** version. For example, `v1.30.1`.

d. From the **Network type** field, select your VCN.

e. In the **VCN in Compartment** text box, enter your VCN name. For example, `dbe-vcn`.

f. From the **Kubernetes service LB subnets in compartment** field, select the LB subnet created in your VCN.

g. From the **Kubernetes API endpoint subnet in compartment** list, select the private subnet created in your VCN.
The node pool details will be auto-populated in the **Node pools** section.

h. Click the **Review** link from the left pane to review the details.

4. Click **Create cluster**.
The Kubernetes cluster is created.

Verifying and Accessing the Cluster

Before verifying the access to the cluster:

- Download OCI CLI version 2.24.0 or later. See Installing the CLI for more information.
- Configure CLI on your local machine. See Configuring the CLI for more information.
- Navigate to the cluster details in OCI, click **Access Cluster**, and then select the **Local Access** option.

To verify the access to the cluster:

1. Open OCI CLI on your local machine.
2. Log in to the public bastion using SSH.
3. Navigate to the cluster details in OCI, then click **Access Cluster**, and then copy the commands from the **Access Your Cluster** page.
4. Run the commands in OCI CLI in the sequence mentioned in the **Access Your Cluster** page to download and configure the `kubectl` file.
5. Run the following commands in OCI CLI to verify the cluster access:

```
$kubectl get ns
$kubectl get nodes
```

Deploying Using an Oracle CNE

To deploy a Kubernetes cluster using Oracle CNE:

1. Log in to OCI.
2. From the **Navigation** pane, select **Compute**, and then select **Instances**.
A **Create Instance** page opens.
3. Click **Create Instance**.
4. From the **Compartment** drop-down list, select the appropriate compartment for your instance. For example, **operations-staging**.
5. In the **Instance Name** field, enter the display name of your instance.
6. Under the **Image and Shape** field, provide the following details:

a. In the **Image** field, browse and upload an appropriate image for your instance. For example, Oracle Linux, Ubuntu, or a custom image.

b. In the **Shape** field, select an appropriate shape. For example, **VM.Standard.E2.1.Micro**.

7. Under the **Configure Networking** field, provide the following details:
   a. From the **VCN** drop-down list, select the VCN you created.
   b. From the **Subnet** drop-down list, select the subnet within the VCN selected in the above step.
8. Select either **Public IP address** or **Private IP address** for the instance.
9. Click **Create Instance**.
10. After the instance is created, see Oracle Cloud Native Environment Quick Start Guide in Oracle Cloud Native Environment Quick Start for Release 2 and follow the instructions in the sequence mentioned in this guide to complete the deployment of the Kubernetes cluster using OCNE.

Deploying Using a Cloud Native Computing Foundation Environment

This section provides detailed instructions about deploying a Kubernetes cluster using a Cloud Native Computing Foundation (CNCF) environment, which involves creating a control plane and creating a worker node.

Prerequisites for Deploying a CNCF Environment

- Ensure that the system has at least four CPUs and 64 GB of memory.
- You must be on an Oracle Linux 8 operating system.
- You must disable the swap option on all nodes.
- You must have a good internet connection to download the packages.

Creating a Control Plane

To create a control plane:

1. Log in to the virtual machine (VM) control plane as a user with sudo access and run the following script on Oracle Linux 8 to configure a Kubernetes control plane:

```
#!/bin/bash
################# Adapted for Oracle Linux 8 ################
# This script configures a Kubernetes control plane on Oracle Linux 8.

# Set Kubernetes and crictl versions
export VER="v1.31.1"
export K8S_VER="1.31"
export K8S_PKG="v1.31"

# Check if the script has been run before
FILE=/k8scp_run
if [ -f "$FILE" ]; then
    echo "WARNING!"
    echo "$FILE exists. Script has already been run on control plane."
    echo
    exit 1
else
    echo "$FILE does not exist. Running script."
fi
```

```
# Prevent script from running twice
sudo touch /k8scp_run

# Update the system
sudo dnf update -y

# Install necessary software
sudo dnf install -y curl vim git wget gnupg2 socat \
    yum-utils device-mapper-persistent-data lvm2

# Add the Kubernetes repo
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/${K8S_PKG}/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/${K8S_PKG}/rpm/repodata/
repomd.xml.key
#exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF

# Install Kubernetes components
sudo dnf install -y kubelet kubeadm kubectl
sudo dnf versionlock add kubelet kubeadm kubectl

# Ensure Kubelet is running
sudo systemctl enable --now kubelet

# Disable swap
sudo swapoff -a
sudo sed -i '/swap/d' /etc/fstab

# Load necessary kernel modules
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter

# Update networking settings
cat <<EOF | sudo tee /etc/sysctl.d/kubernetes.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sudo sysctl --system

# Install containerd
sudo dnf config-manager --add-repo https://download.docker.com/linux/
centos/docker-ce.repo
sudo dnf install -y containerd.io

# Configure containerd
```

```
sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml
sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/
containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd

# Install and configure crictl
wget https://github.com/kubernetes-sigs/cri-tools/releases/download/${VER}/
crictl-${VER}-linux-amd64.tar.gz
tar zxvf crictl-${VER}-linux-amd64.tar.gz
sudo mv crictl /usr/local/bin
sudo crictl config --set \
runtime-endpoint=unix:///run/containerd/containerd.sock \
--set image-endpoint=unix:///run/containerd/containerd.sock

# Initialize the Kubernetes cluster
sudo kubeadm init --pod-network-cidr=10.244.0.0/16 | sudo tee /var/log/
kubeinit.log

# Configure kubectl for the current user
mkdir -p $HOME/.kube
sudo cp -f /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

# Install Cilium CLI
export CILIUM_CLI_VERSION=$(curl -s https://raw.githubusercontent.com/
cilium/cilium-cli/master/stable.txt)
export CLI_ARCH=amd64
if [ "$(uname -m)" = "aarch64" ]; then CLI_ARCH=arm64; fi
curl -L --fail --remote-name-all https://github.com/cilium/cilium-cli/
releases/download/${CILIUM_CLI_VERSION}/cilium-linux-$
{CLI_ARCH}.tar.gz{,.sha256sum}
sha256sum --check cilium-linux-${CLI_ARCH}.tar.gz.sha256sum
sudo tar xzvfC cilium-linux-${CLI_ARCH}.tar.gz /usr/local/bin
rm cilium-linux-${CLI_ARCH}.tar.gz{,.sha256sum}
cilium install --set ipam.mode=cluster-pool --set
ipam.operator.clusterPoolIPv4PodCIDRList=10.244.0.0/16 --set
ipam.operator.clusterPoolIPv4MaskSize=24

#Disable firewall
sudo systemctl stop firewalld
sudo systemctl disable firewalld
sudo systemctl mask --now firewalld

# Install Helm
wget https://get.helm.sh/helm-v3.16.2-linux-amd64.tar.gz
tar -xf helm-v3.16.2-linux-amd64.tar.gz
sudo mv linux-amd64/helm /usr/local/bin/

# Output the state of the cluster
kubectl get node

# release storage
sudo /usr/libexec/oci-growfs -y
```

```
echo "Setup complete. Proceed to the next step."
```

2. Check if the status of the control plane node is **Ready**. The following is a sample command and its output:

```
kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
vanillak8sre-cp    Ready    control-plane   78m   v1.31.4
```

3. Run the following command to get the kubejoin details:

```
kubeadm token create --print-join-command
```

Creating a Worker Node

To create a worker node:

1. Log in to the VM worker node and run the following script on Oracle Linux 8 to set up a Kubernetes worker node:

> **Note**
>
> You can also add multiple worker nodes based on your environment's requirement.

```
#!/bin/bash
################# Adapted for Oracle Linux 8 ################
# This script sets up a Kubernetes worker node on Oracle Linux 8.


export VER="v1.31.1"
export K8S_VER="1.31"
export K8S_PKG="v1.31"

# Check if the script has been run before. Exit if it has.
FILE=/k8scp_run
if [ -f "$FILE" ]; then
    echo "WARNING!"
    echo "$FILE exists. Script has already been run."
    echo "Do not run on the control plane. Run on a worker node."
    echo
    exit 1
else
    echo "$FILE does not exist. Running script."
fi

# Prevent script from being run multiple times
sudo touch /k8scp_run

# Update the system
sudo dnf update -y

# Install necessary software
```

```
sudo dnf install -y curl vim git wget gnupg2 socat \
    yum-utils device-mapper-persistent-data lvm2

# Add the Kubernetes repo
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/${K8S_PKG}/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/${K8S_PKG}/rpm/repodata/
repomd.xml.key
#exclude=kubelet kubeadm kubectl cri-tools kubernetes-cni
EOF

# Install Kubernetes components and lock their versions
sudo dnf install -y kubelet kubeadm kubectl
sudo dnf versionlock add kubelet kubeadm kubectl

# Ensure Kubelet is running
sudo systemctl enable --now kubelet

# Disable swap
sudo swapoff -a
sudo sed -i '/swap/d' /etc/fstab

# Load necessary kernel modules
cat <<EOF | sudo tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF
sudo modprobe overlay
sudo modprobe br_netfilter

# Update networking settings
cat <<EOF | sudo tee /etc/sysctl.d/kubernetes.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF
sudo sysctl --system

# Install containerd
sudo dnf config-manager --add-repo https://download.docker.com/linux/
centos/docker-ce.repo
sudo dnf install -y containerd.io

# Configure containerd
sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml
sudo sed -i 's/SystemdCgroup = false/SystemdCgroup = true/' /etc/
containerd/config.toml
sudo systemctl restart containerd
sudo systemctl enable containerd

# Install and configure crictl
```

```
wget https://github.com/kubernetes-sigs/cri-tools/releases/download/${VER}/
crictl-${VER}-linux-amd64.tar.gz
tar zxvf crictl-${VER}-linux-amd64.tar.gz
sudo mv crictl /usr/local/bin
sudo crictl config --set \
runtime-endpoint=unix:///run/containerd/containerd.sock \
--set image-endpoint=unix:///run/containerd/containerd.sock

#Disable firewall
sudo systemctl stop firewalld
sudo systemctl disable firewalld
sudo systemctl mask --now firewalld


# release storage
sudo /usr/libexec/oci-growfs -y


# Instructions for joining the worker node to the cluster
sleep 3
echo
echo
echo '**************************'
echo
echo "Continue to the next step"
echo
echo "Use sudo and copy the kubeadm join command from"
echo "the control plane node."
echo
echo '**************************'
echo
echo
```

2. Run the kubejoin command and check the node status. The following is a sample kubejoin command and its status:

```
kubeadm join 10.0.5.248:6443 --token ukztj1.sgd165r61xknz2qs --discovery-
token-ca-cert-hash
sha256:7ab8dff2197aaa6c0701c35132006ab0934e95e8e259611b17d4710285dbfbc9
kubectl get nodes
NAME              STATUS   ROLES           AGE   VERSION
abc8sre-cp        Ready    control-plane   78m   v1.31.4
abc8sre-wn1       Ready    <none>          67m   v1.31.4
```

Deploying the Solution Components

To deploy the Digital Business Experience solution:

1. Request an environment for the solution. Contact Oracle Support for assistance.
2. Prepare the cluster for the solution. See Preparing the Cluster for information.
3. Deploy Launch Cloud Service and CX Industries Framework. See *Oracle Communications Launch Implementation Guide* for detailed instructions.

4. Deploy BRM. See *Oracle Communications Billing and Revenue Management Cloud Native Deployment Guide* for detailed instructions.
5. Deploy PDC. See *Oracle Communications Billing and Revenue Management PDC Installation Guide* for detailed instructions about deploying PDC.
6. Deploy ECE. See *Oracle Communications Billing and Revenue Management ECE Installation Guide* for detailed instructions about deploying ECE.
7. Deploy OCOMC. See *Oracle Communications Offline Mediation Controller Cloud Native Installation and Administration Guide* for detailed instructions.
8. Deploy Oracle Analytics Server. See Installing and Configuring Oracle Analytics Server for detailed instructions.
9. Deploy Service Catalog and Design Studio. See *Oracle Communications Service Catalog and Design Studio Installation Guide* for detailed instructions.
10. Deploy OSM. See *Oracle Communications Order and Service Management Cloud Native Deployment Guide* for detailed instructions.
    - Deploy Order to Activate (O2A). See *Order and Service Management Cartridges for Application Integration Architecture Cloud Native Deployment Guide* for detailed instructions.
11. Deploy Siebel CRM. See Developing and Deploying Siebel CRM for detailed instructions.
12. Deploy Oracle Data Integrator. See Oracle Fusion Middleware Installing and Configuring Oracle Data Integrator Guide for detailed instructions.
13. Deploy AIA. See *Oracle Communications Application Integration Architecture Cloud Native Deployment Guide* for detailed instructions.

Validating the Solution Components

You must validate your solution after deploying it to ensure the proper functioning of the system.

To validate the deployment of the solution:

1. Verify the password expiration of the cloud native applications: See Verifying the Password Expiration.
2. Validate the public certificates: See Validating the Public Certificates.
3. Validate the Launch and CX Industries Framework (CXIF) deployment: See Validating the Connection in *Oracle Communications Launch Cloud Service Integration Guide* and follow the procedure for validating the Launch and CXIF connection.
4. Validate the SCD deployment: See Validating the Solution Designer Instance in *Oracle Communications Service Catalog and Design Solution Designer Installation Guide* and follow the procedure for validating the SCD deployment.
5. Validate the AIA deployment: See Validating the AIA Cloud Native Deployment in *Oracle Communications Application Integration Architecture Cloud Native Deployment Guide* and follow the procedure for validating the AIA deployment.

> **Note**
>
> If any of the above validation fails, stop the validation process, and contact Oracle Support.

Verifying the Password Expiration

To verify the password expiration of the Digital Business Experience cloud native applications, from the Oracle Database:

1. Run the following command to access the sql client:
   `sqlplus / as sysdba`

   The sample output is as follows:

```
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 11 11:46:04 2024
Version 19.24.0.0.0

Copyright (c) 1982, 2024, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c EE High Perf Release 19.0.0.0.0 - Production
Version 19.24.0.0.0

SQL>
```

2. Run the following command to access the PDB:
   `alter session set container=BRMCN15;`
3. Run the following command to check the expiry of a database user:
   `select username, account_status, EXPIRY_DATE from dba_users;`
4. Run the following command to set a database user to no expiration:

```
SQL> alter profile DEFAULT limit PASSWORD_REUSE_TIME unlimited;

SQL> alter profile DEFAULT limit PASSWORD_LIFE_TIME  unlimited;
```

Validating the Public Certificates

This section provides information about validating various public certificates for the Digital Business Experience environment.

Validating the CXIF Certificate

To validate the CXIF certificate, run the following command:

```
openssl s_client -showcerts -connect rododsiebel.jetpen.com:31000
```

The following is a sample output:

```
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = R11
verify return:1
depth=0 CN = rododsiebel.jetpen.com
verify return:1
---
```

Validating the Siebel and PDC Certificates

Before validating the Siebel and PDC certificates, you must identify the Siebel and PDCRSM API endpoint URLs. For example, Siebel API Endpoint URL for Variant 1 is: `https://rododsiebel.jetpen.com:32401`

To validate the Siebel and PDC certificates, run the following command on a jump host with openssl:

```
echo -n Q | openssl s_client -connect rododsiebel.jetpen.com:32401 | openssl x509
-noout -dates ---
```

The following is a sample output:

```
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = R11
verify return:1
depth=0 CN = rododsiebel.jetpen.com
verify return:1
DONE
notBefore=Oct 24 21:39:58 2024 GMT
notAfter=Jan 22 21:39:57 2025 GMT
```

4

Performing Post-Deployment Tasks

This chapter describes the tasks you perform after deploying the Digital Business Experience solution.

Integrating Launch and CXIF

This section describes the tasks you perform to integrate Launch and CX Industries Framework (CXIF) applications with the Digital Business Experience solution.

See Integrate Launch with Digital Business Experience in *Oracle Communications Launch Cloud Service Implementation Guide* for information about Launch integration with Digital Business Experience solution and other applications.

Prerequisites

Before integrating Launch and CXIF, you must:

- Request a Launch environment. Contact Oracle Support for assistance.
- Request a CXIF environment. Contact Oracle Support for assistance.
- Request Launch and CXIF integration. Contact Oracle Support for assistance.

Integrating Launch and CXIF with Siebel CRM and PDC

After the prerequisites are completed, perform the following:

- Integrate Launch and CXIF with Siebel CRM. See Launch Cloud Service Siebel CRM Integration in *Oracle Communications Launch Cloud Service Integration Guide* for more information.
- Integrate Launch and CXIF with Pricing Design Center (PDC) a part of the Oracle Communications Billing and Revenue Management (BRM) suite. See Launch Cloud Service PDC (BRM) Integration in *Oracle Communications Launch Cloud Service Integration Guide* for more information.

Integrating Siebel CRM on Containers with AIA Cloud Native

This section provides instructions for integrating Siebel CRM on Containers with Application Integration Architecture (AIA) cloud native.

To integrate Siebel CRM on Containers with AIA cloud native:

1. Get the following JAR files:
    - From Siebel containers, get **siebel.jar** and **SiebelJI_enu.jar**.
    - From the Oracle WebLogic Server container, get **wlthint3client.jar**.

2. Log in to Siebel eCommunication Web UI as SADMIN user and update the JAVA64 profile parameters to include the JAR files you copied in the previous step:

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

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

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 with the following contents:

> **Note**
>
> - Ensure that the AIACN t3 channel URL and user name and password are set with the correct values.
> - In the below example, Siebel and AIA are in the same cluster and different namespace.

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
java.naming.provider.url=t3://
soa_cluster_service_name.soa_namespace.svc.cluster.local:soa_service_cluste
r_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 Enterprise Server (SES) pod.
7. Configure Siebel Web Service in the Siebel database.
   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_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/services/default/AccountBalanceSiebelCommsReqABCS/
AccountBalanceSiebelCommsReqABCS_ep' WHERE
NAME='_soap_AccountBalanceSiebelCommsReqABCS_AccountBalanceSiebelCommsRe
qABCS';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://
aiacn_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/services/default/AdjustmentSiebelCommsReqABCS/
AdjustmentSiebelCommsReqABCS_ep' WHERE
NAME='AdjustmentSiebelCommsReqABCSPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://
aiacn_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/services/default/InvoiceSiebelCommsReqABCS/
InvoiceSiebelCommsReqABCS_ep' WHERE
NAME='_soap_InvoiceSiebelCommsReqABCS_InvoiceSiebelCommsReqABCS';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://
aiacn_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/services/default/PaymentSiebelCommsReqABCS/
PaymentSiebelCommsReqABCS_ep' WHERE
NAME='PaymentSiebelCommsReqABCSPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://
aiacn_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/services/default/UnbilledUsageSiebelCommsReqABCS/
UnbilledUsageSiebelCommsReqABCS_ep' WHERE
NAME='_soap_UnbilledUsageSiebelCommsReqABCS_UnbilledUsageSiebelCommsReqA
BCS';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://
aiacn_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/services/default/SyncCustomerSiebelEventAggregator/Client'
WHERE NAME='SyncCustomerSiebelEventAggregatorPort';
UPDATE S_WS_PORT SET PORT_ADDRESS='http://
aiacn_clusterService_name.aiacn_kubernetes_name:aiacn_clusterService_por
t/soa-infra/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 the Siebel database and run the SQL script with the database user name and the corresponding password.

8. Configure the Siebel repository in the Siebel database:

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 the Siebel database and run the SQL script with the database user name and the corresponding password.

9. Configure EAI File Transfer Folder:

a. Connect to the SES pod of the Siebel CRM on Containers instance.

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

> **Note**
>
> Ensure that the `/siebel/mde/siebsrvr/temp/OrderBackup` folder is created in the Siebel SES container. If not, then create the folder and proceed with below commands.

```
[aiacn_pod-0:/siebel/mde]#srvrmgr /g cgw-aiacn-0.ses-aiacn.siebel-
cn.svc.cluster.local:2320 /e aiacn /u username /p password


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 you have verified that it is running, follow Step 5 and Step 6.

10. (Optional) Import products into Siebel CRM on Containers by using the Siebel eCommunication application. Refer to Siebel CRM documentation for instructions.
11. Import Siebel CRM on Containers SSL/TLS security certificates and configure AIA cloud native with the certificates. See Installing SSL Certificates in *Oracle Application Integration Architecture Cloud Native Deployment Guide* for more details.
    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/***domain_name***/bin/setDomainEnv.sh** with custom trust:
       i. Connect to any managed server pod.
       ii. Open the **/u01/oracle/user_projects/domains/***domain_name***/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* **-Djavax.net.ssl.trustStorePassword=***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 Connection details, Business Unit ID, and PRICELIST.
    a. Get the latest **AIAConfigurationProperties.xml** and **PRICELIST.dvm** files:
        i. In the Enterprise Manager Console for AIA, navigate to the **MDS Configuration** section and export the zip file.
        ii. Copy the following files from the zip: **soa/configuration/default/AIAConfigurationProperties.xml** and **apps/AIAMetaData/dvm/PRICELIST.dvm**. For more information on MDS operations, refer to Managing the Metadata Repository in *Administering Oracle Fusion Middleware*.
    b. Update **Siebel connection** details:
        i. In the **AIAConfigurationProperties.xml** file, update all the values of the XML tag **SEBL_01.EndpointURI** with correct details if required.
    c. Update the **Business Unit ID** details:
        i. Log in to Siebel and get Siebel Business Unit ID by navigating to **Organizations**. Navigate to **About Record**, and copy the Row Number.
        ii. In the **AIAConfigurationProperties.xml** file, update all the values of the XML tag **Siebel.SEBL_01.BusinessUnit**, with the copied Default Organization Row Number.
    d. Update Pricelist details:
        i. **Pricelists**:
            i. Log in to Siebel and create or confirm pricelists as required on Siebel. Copy the Row Number of the required pricelist. For more information, refer to Siebel Price List in the *Application Services Interface Reference Guide*.
            ii. In the **PRICELIST.dvm** file, update or add the values for **SEBL_01 column** of **row** with the copied Row Number and with the other required values.

                For more information, refer to Working with the PRICELIST DVM in the *Order to Cash Implementation Guide* .
            iii. Repeat steps 1 and 2 for each pricelist.
        ii. **Default Pricelist**:
            i. In the **AIAConfigurationProperties.xml** file, update all the values of XML tag **Siebel.SEBL_01.PriceList.ID** with Row Number from Siebel for default pricelist.

> **Note**
>
> Mention the default pricelist details in the **AIAConfigurationProperties.xml** file. In case of multi-pricelist configuration, do not mention the default pricelist section in the **PRICELIST.dvm** file.

e. Update AIA MDS with the updated **PRICELIST.dvm** and the **AIAConfigurationProperties.xml** files. For more information about updating files in AIA MDS, refer to Updating Files in AIA MDS in the *Application Integration Architecture Cloud Native Deployment Guide*.

f. Restart AIA cloud native using 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.

15. Update Maximum Number of Records in Siebel:

    a. Create and activate a workspace:

        i. Log in to Siebel Web Tools.

        ii. Click the **Workspaces** icon in the top menu bar.
        Ensure the **Main** workspace is selected.

        iii. Click **Create**.

        iv. Enter the required **Workspace Name** and **Comments**.

        v. Click **Save**.

        vi. Confirm that the new workspace is active.
        The workspace name appears in the top menu bar instead of **Main**.

        vii. Click **X**.

    b. Update maximum number of records:

        i. Navigate to **Integration Object** and query `CMU Request Billed Usage IO`.
        If the integration object is hidden,

            - Click the pencil icon to view the hidden menus.
            - Select the integration object.
            - Click **Save**.

        ii. Navigate to **Integration Component Field**, then select **Integration Component Field User Props**, and query `Maximum*`.

        iii. Update the **PREDEFAULT** value present under **Integration Component Field User Props** to 30.

        iv. Click **Save**.

    c. Submit and deliver the workspace:

        i. Navigate back to Workspaces.
        Ensure your workspace is selected.

        ii. Click **Version**, enter the comments, and verify if the status is changed to **Checkpointed**.

        iii. Click **Submit** and verify if the status is changed to **Submitted for delivery**.

        iv. Click **Deliver** and verify if the status is changed to **Delivered**.

v. After the workspace is delivered, the system switches to the **Main** workspace, and your changes will be visible in the **Main** workspace.

d. Validation:

i. Log out of Siebel Web Tools.

ii. Log in again.

iii. Verify that your changes are visible in the workspace.

e. Restart Siebel to ensure your changes are visible.

16. Update SWI Trouble Ticket Service in Business Service Access:

a. Add the business service:

i. Log in to Siebel application using your credentials.

ii. Navigate to **Site Map**, select **Administration - Application**, and then click **Business Service Access**.

iii. In the **Business Service Access** view, click the **Add (+)** button.

iv. In the new record, enter the business service name (for example, SWI Trouble Ticket Service).

v. Click **Save**.

b. Assign Responsibility:

i. In the same view, locate the newly added business service (for example, SWI Trouble Ticket Service).

ii. Update the Responsibility details as required.

iii. Click **Save**.

> **Note**
>
> After adding the business service and assigning the responsibilities, you must clear cache in Business Service and Responsibility applets.

17. Enable Dynamic Pricing:

a. Log in to Siebel application using your credentials.

b. Navigate to **Site Map**, and then select **Administration - Pricing**.

c. In the **Price Lists** view, click the **Settings** icon.

d. From the menu, select **Enable Dynamic Pricing**.

e. Click the **Settings** icon again, and then click **Save**.

Integrating BRM Cloud Native with AIA 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 service>.<brm ns>.svc.cluster.local
```

2. Deploy the BRM JCA Adapter. See Deploying the BRM JCA Adapter in *Oracle Communications Application Integration Architecture Cloud Native Deployment Guide* for more information.
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. See Testing JCA Resource Adapter Configuration and BRM Connectivity in *Oracle Communications BRM JCA Resource Adapter Guide* for more information.
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. You can set this by using the helm chart or by editing the Kubernetes config manager entry of cm `cm-pin-conf-config`.
   b. For EAI (**eai-java-server** container in the cm pod), ensure that payload for **Infranet.properties** is set to **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 0.0.0.0, which points to EAI. The DB entries mapping can be found in dm-oracle pin.conf. You can set this by using the helm chart or by editing the Kubernetes config manager entry of the EAI `eai-java-server-infranet-properties-config` and `eai-java-server-payload`.

      > **Note**
      >
      > Bounce the CM pod after the change to reflect the changes.
      > `Publisher DB=\"0.0.0.0`

   c. In the **dm-oracle-aq-event-map-config** config file, uncomment ALL in aq_event_map.
   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. Run helm command from the BRM Helm charts path to publish Prod Sync data.

```
cd $BRM_CNTK/artifacts
$ helm upgrade --namespace <brm_namespace> brm-cn-apps helm-charts/ --
values profiles/deploy-oci.yaml --values profiles/pdc-publish.yaml
```

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

5. Enable Sequential Cycle Discounting Using Utility:
   a. Access the Pin Job Executor (PJE) pod from the `dbe_monetization` namespace.
   b. Navigate to the configuration directory `cd /oms/sys/data/config/`.
   c. Open the configuration file using the following command:

```
vi bus_params_billing.xml
```

   d. Locate and enable the `SequentialCycleDiscounting` parameter as follows:

```
<SequentialCycleDiscounting>enabled</SequentialCycleDiscounting>
```

   e. Save and close the file.
   f. Run the following command to load the updated XML into the BRM database:

```
pin_bus_params -v bus_params_billing.xml
```

   g. Verify that the command completes without errors.
   h. Restart the CM pod to apply the changes.

Integrating OSM Cloud Native with AIA 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 Administration Console for AIA cloud native.
     a. Log in to AIA cloud native WebLogic Administration Console.
     b. Navigate to the Servers section in the Domain Structure pane and then select a managed server (for example, select **soa_server1**).
     c. In Protocols, go to the Channels tab, 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 as ***soa-cluster-service-name.namespace*.svc.cluster.local**
     e. Ensure that **Tunneling Enabled** is selected.
     f. For each managed server, repeat steps **b.** to **e.**

- If 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 restart automatically, run the corresponding scripts to manually 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 to the hostname defined earlier. In addition, ensure that **HTTP Enabled for This Protocol** is selected.

2. Deploy the Order-to-Activate cartridge into the OSM cloud native instance. See Creating Order-to-Activate Credentials and Accounts and Deploying the Sample Cartridge for instructions.
3. Set the AIA cloud native SAF t3 value which was 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_servicep
ort
    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. Restart the OSM instance to deploy the O2A cartridge with the AIA cloud native configuration. See Restarting the Instance in *Oracle Communications Order and Service Management Cloud Native Deployment Guide*.
5. Log in to the WebLogic Administration 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 displayed 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.

> **Note**
>
> You must set a username and password for the WebLogic Admin Console for AIA cloud native.

9. Restart the AIA cloud native domain services by using the domain-lifecycle scripts.
10. In the WebLogic Administration Console for the AIA domain, 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, and 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.

Integrating ODI with AIA Cloud Native

This section describes tasks to be performed to integrate Oracle Data Integrator (ODI) with Oracle Communications Application Integration Architecture (AIA) cloud native.

Prerequisites and Assumptions

Ensure the following:

- The Master Repository and the Work Repository are created in ODI.
- ODI is installed with the **J2EE Enterprise Agent**template and has the **encode.sh** script in **$ODI_HOME/bin** directory.
- The ODI agent is available and the agent's WSDL is reachable inside the AIA cloud native installation pods.
- Collections is installed properly.. Within the admin pod, verify that the following are present in the DVM path **/u01/shared/aia-comms/commsOracleHome/comms_home/source/soainfra/apps/AIAMetaData/dvm**:
  - CURRENCY_CODE.dvm
  - COLLECTION_STATUS.dvm
  - COLLECTION_ACTIONNAME.dvm
  - COLLECTION_PRIORITY.dvm
  - COLLECTION_SUBSTATUS.dvm
- Confirm the URL for composite definition for **SyncCollectionHeaderInfoBRMCommsReqImpl** composite is correct.

Perform the following tasks before integrating ODI with AIA:

- Mount AIA PV on the machine where ODI is installed. Mount the PV at **/u01/shared/**.
- Configure a unified collections action view and a unified collections scenario view in single-schema or multischema environment to be queried by Oracle AIA. See Configuring Views for Oracle Application Infrastructure Architecture for detailed instructions about configuring views for Oracle AIA.

Integrating ODI with AIA

To integrate ODI with AIA cloud native:

1. Update the email details in the **/u01/shared/aia-comms/commsOracleHome/comms_home/source/soainfra/apps/config/AIAConfigurationProperties.xml** file as required. Update the properties for the below **CollectionsParameters** module file:
   - **FromMailAddress**
   - **ToMailAddress**
   - **MailServer**
   - **MailSMTPPort**
2. Encode the **AIA XREF** password:

To encode the XREF password, run the following command and note the encoded password:

```
$ODI_HOME/bin/encode.sh -INSTANCE=OdiInstanceName AIA_XREF_DB_Password
```

Update the `$AIA_PV/aia-comms/commsOracleHome/comms_home/source/soainfra/apps/services/industry/Communications/BulkDataProcess/BRMToSiebel/Collections/ODI/Oracle/V1/ODI_Master_Repository/CONN_AIADS.xml` file with the encoded password.

3. Encode the **BRM** password:
   To encode the BRM password, run the following command and note the encoded password:

   ```
   $ODI_HOME/bin/encode.sh -INSTANCE=OdiInstanceName
   BRM_Database_Advanced_Queuing_user_password
   ```

   Update the `$AIA_PV/aia-comms/commsOracleHome/comms_home/source/soainfra/apps/services/industry/Communications/BulkDataProcess/BRMToSiebel/Collections/ODI/Oracle/V1/ODI_Master_Repository/CONN_OracleBRMDS.xml` file with the encoded password.

4. Encode the **Siebel** password:
   To encode the Siebel password, run the following command:

   ```
   $ODI_HOME/bin/encode.sh -INSTANCE=OdiInstanceName
   Siebel_database_user_password
   ```

   Update the `$AIA_PV/aia-comms/commsOracleHome/comms_home/source/soainfra/apps/services/industry/Communications/BulkDataProcess/BRMToSiebel/ Collections/ODI/Oracle/V1/ODI_Master_Repository` file with the encoded password.

5. Import the AIA connections to ODI Studio. If the AABC installation is successful, then the **importAIAODIConnections.sh** script will be generated in the **AIA_PV** directory. Run the following command to import:

   ```
   sh importAIAODIConnections.sh \
       -a AIA_PV_Path \
       -o ODI_DOMAIN_HOME_path \
       -i ODI Instance Name. Example: OracleDIAgent \
       -w ODI Workrep name. Example: WORKREP
   ```

Verifying the Integration

To verify the Siebel CRM on Containers and BRM integration deployment:

1. Log in to ODI Studio with ODI user credentials.
2. Verify and confirm that the **AIA Collections Project** is imported into the **Designer** section.
3. Verify and confirm the connection details under **Topology** - **Physical Architecture** - **Oracle and XML**.
   - Oracle - AIA DS
   - Oracle - Oracle BRM DS
   - Oracle - Siebel DS

- XML - AIA Config Properties DS
- XML - Collection Action Name
- XML - Collection Priority DS
- XML - Collection Status Mapping
- XML - Collection Substatus model
- XML - Currency DS

Applying Tuning Parameters on BRM and AIA

Apply the following tuning parameters to improve performance, scalability, and resource utilization during high-volume operations.

Updating the Front-End and Back-End Configuration in BRM

To update the front-end and back-end configuration in Oracle Communications Billing and Revenue Management (BRM):

1. Navigate to the BRM Helm chart directory and open the `values.yaml` file.
2. Under `dm-oracle/config`, update the following parameters:

   ```
   totalFrontEnds: 8
   totalBackEnds: 64
   connectionsPerFrontEnd: 16
   totalTransBackEnds: 52
   ```

3. In the same `values.yaml` file, increment the `restart_count` value by 1.
4. Run a Helm upgrade for BRM.
   The `dm-oracle` pod restarts automatically after the upgrade.

Validation

1. Access the `dm-oracle` pod using the following command:

   ```
   kubectl exec -it <dm-oracle-pod> -- /bin/bash
   ```

2. Navigate to the `/oms/sys/dm_oracle` directory.
3. Verify the updated configuration by running the following commands:

   ```
   Check connectionsPerFrontEnd: echo $DM_MAX_PER_FE
       Check totalFrontEnds: echo $DM_NO_FRONT_ENDS
       Check totalBackEnds: echo $DM_NO_BACK_ENDS
       Check totalTransBackEnds: echo $DM_NO_TRANS_BE_MAX
   ```

4. Confirm that the output values match the configuration in the `values.yaml` file.

Verifying and Updating the Java Heap Space in AIA Deployment

To verify and update the Java heap space in Oracle Communications Application Integration Architecture (AIA) deployment:

1. Run the following commands to verify the current Java heap configuration for AIA pods:

```
kubectl exec -n <aia_namespace> <soa_server1_pod> -- printenv | grep
USER_MEM_ARGS
kubectl exec -n <aia_namespace> <soa_server2_pod> -- printenv | grep
USER_MEM_ARGS
kubectl exec -n <aia_namespace> <soa_adminserver_pod> -- printenv | grep
USER_MEM_ARGS
```

2. Review the output. If the heap size is low (for example, -Xms256m -Xmx1024m), increase it by editing the `USER_MEM_ARGS` domain of Java heap.
3. Before making changes, create a backup of the domain resource:

```
kubectl get domain -n <namespace> <domain_name> -o yaml > <domain_name>-
backup.yaml
```

4. Update Java heap size:
   a. Edit the domain configuration:

```
kubectl edit domain -n <namespace> <domain_name>
```

   b. Locate `USER_MEM_ARGS` under `spec.serverPod.env` and update the heap values. For example:

```
serverPod:
  env:
  - name: USER_MEM_ARGS
   value: "-Djava.security.egd=file:/dev/./urandom -Xms4g -Xmx8g"
```

   c. Adjust `-Xms` (initial heap) and `-Xmx` (maximum heap) based on the system capacity and requirements.
5. Restart the domain to apply the updated configuration.
6. After the restart, run the verification commands (see Step 1) again to confirm that the new heap settings are applied correctly.

Testing the Solution Deployment

To test the end-to-end deployment of the solution, do the following:

- Testing the Solution using the Smoke Test Validation
- Testing the Integrations using Functional Testing (Smoke Test)

Testing the Solution using the Smoke Test Validation

The smoke test validation includes validating your access to UIs of all the applications integrated with the Digital Business Experience solution and validating if the load balancer is working properly.

Checking the URLs of the Deployed Applications

To check if all the deployed application UIs are accessible:

1. For each of the URLs in the email you received, open the application URLs one-by-one and log in using your credentials.
2. Verify if you can access all the URLs.

> **Note**
>
> If you have trouble logging in to any of the application URLs, contact Oracle Support.

Validating the Access to Load Balancer

To validate if the load balancer is working fine:

1. Open Postman on the environment you wish to test the load balancer from.
2. Click **New**, and then click **Request**.
3. From the **Request Type** drop-down list, select **GET**.
   The GET method page opens.
4. In the **GET URL** field, enter the URL for the public DNS name of your load balancer.
   Example load balancer URL for PDC:

   `https://public_load_balancer_dns_name:31200/pdc/config`

   Example load balancer URL for Siebel:

   `https://<public_load_balancer_dns_name:32200/siebel/config`
5. Under the **Authorization** tab, set the header for authorization.
6. Click **Send**.
7. Validate the response.
   - If the status code is **200 OK**, the load balancer is working fine.
   - If the status code is **503 Service Unavailable**, there might be a back-end issue.

Testing the Integrations using Functional Testing (Smoke Test)

This section provides information about the functional testing of the application integrations by manually creating a smoke test offer.

Prerequisites

- You must create Siebel and PDC destinations in Launch.
- The Lookup code for `/service/telco/gsm/telephony` must be added in Launch and Service Event must be added in PDC.

To create a Siebel destination in Launch:

1. Log in to the Launch application using your credentials.
2. Navigate to **Product Management**, select **Enterprise Catalog**, then select **Administration**, then select **Manage Lifecycle Status**, then select **Destinations**, and then click **Create**.
3. Provide the values to the following fields:

- **Domain Name**: Siebel
- **Domain Instance**: siebel-test
- **Publish Sequence**: 1
- Turn on the **Internal** toggle.

4. Click **Save**.

To create a PDC destination in Launch:

1. Log in to the Launch application using your credentials.
2. Navigate to **Product Management**, select **Enterprise Catalog**, then select **Administration**, then select **Manage Lifecycle Status**, then select **Destinations**, and then click **Create**.
3. Provide the values to the following fields:
   - **Domain Name**: PDC
   - **Domain Instance**: pdc-test
   - **Publish Sequence**: 2
   - Turn on the **Internal** toggle.
4. Click **Save**.

After Siebel and PDC destinations are created:

1. From the Launch UI, navigate to **Manage Lifecycle Status**, and then click **Save as New Version**.
2. Set the Lifecycle Status to **Ready to publish** for both the destinations.
3. Click **Activate**.

Creating a Smoke Test Offer in Launch (Design Time)

> **Note**
>
> The procedure mentioned below is for creating a smoke test offer in Launch manually.

To create a smoke test offer in Launch:

1. Verify that the Siebel and PDC destinations are created in Launch:
   a. Log in to the Launch application using your credentials.
   b. Navigate to **Administration**, and then click **Manage** available on the **Lifecycle status** card.
   The **Entity lifecycle** page opens.
   c. Click **View** on the **Destinations** card.
   d. Ensure that Siebel and PDC destinations are configured with appropriate details.
   e. Click **Back (<)** to go back to the **Entity lifecycle** page.
   f. Click **Manage** available on the **Lifecycle status** card.
   The **Lifecycle Status** page opens.
   g. Confirm that the **Lifecycle Status** column displays **Ready to publish** for **Destinations (2)**.

2. Create the smoke test offer in Launch:
   a. From the Launch home page, navigate to **Administration**, and then click **Create** available on the **Initiatives** card.
   The **Initiative create** page opens.
   b. Create an initiative by providing values for the **Name** and **ID** fields under the **Identifying information** tab. For example, the value can be **SmokeTest_initiative**.
   c. Create a Balance Element (as Balance Element is a seeded data in Launch) for USD in Launch using the following API details:
      - API Endpoint: `POST {{host_url}}/crmRestApi/atcProductCatalog/11.13.18.05/productCatalogReferenceManagement/v1/balanceElement`
      - host_url: FA URL of Launch
      - Authorization: Basic Auth with Launch credentials
      - Use the following sample payload as the body of the POST request:

```
{
        "id": "USDCurrency",
        "name": "USD Currency",
        "version": "1.0",
        "lifecycleStatus": "In design",
        "@type": "BalanceElementOracle",
        "validFor": {
            "startDateTime": "2023-09-29T03:50:48.000Z"
        },
        "project": {
            "id": "SmokeTest_initiative",
            "version": "1.0",
            "@referredType": "ProjectOracle"
        },
        "consumptionRule": "LST",
        "balanceElementType": "CURRENCY",
        "code": "USD",
        "numericCode": 840,
        "symbol": "$",
        "roundingMethod": "CALC",
        "decimalPlaces": "2"
    }
```

   d. After the Balance Element is created, from the Launch home page, navigate to **Administration**, and then click **Manage** available on the **Pricing strategies** card.
   e. Click **Create Pricelist**.
   f. In the **Name** field, enter `NA Pricelist`.

> **Note**
>
> You must enter the name as NA Pricelist only as it is the default pricelist in the AIA DVM configuration.

   g. From the **Price List Type** drop-down list, select **USD**.

h. Navigate to **Workbench**, select **Specifications**, and then select **Create Service Specification**.

i. To create a service specification, enter any value in the **Name** (for example, SmokeTest_SS) and **ID** fields.

j. Navigate back to the **Specifications** page, and click **Create Product Specification**.

> **Note**
>
> You must create VoIP PS and CME VoIP PS product specifications in the offer as they are part of the O2A cartridge by default.

k. Create the VoIP PS as follows:

   i. In the **Name** field, enter `VoIP PS`.

   ii. In the **Other information** section, from the **Initiative** drop-down list, select your initiative (for example, **SmokeTest_initiative**).

   iii. Click **Save**.

l. Create the CME VoIP PS as follows:

   i. In the **Name** field, enter `CME VoIP PS`.

   ii. From the **Parent Product Specification** drop-down list, select **VoIP PS**.

   iii. In the **Other information** section, from the **Initiative** drop-down list, select your initiative (for example, **SmokeTest_initiative**).

   iv. In the **Other information** section, in the **Service Specification** field, select your service specification (for example, **SmokeTest_SS**).

   v. Click **Save**.

m. After the product specifications are created:

   i. Navigate back to the **Specifications** page.

   ii. Under the **Action** column for each product specification, click **...**, then select **Change Status**, and then click **Design Complete**.

> **Note**
>
> This step is mandatory for the specifications to be available to use in Offers.

n. From the Launch home page, click the **Offerings** tab and click **Create Simple Offering**.

o. Enter the general information as follows:

- **Name**: Any name (for example, SmokeTest_Offer)
- **ID**: Any ID (for example, SmokeTest_Offer)
- **Offering Type**: Service
- **Product Specification**: CME VoIP PS
- **Service Specification**: SmokeTest_SS
- **Billing Type**: Subscription

- **Billing Service Type**: /service/telco/gsm/telephony
- Enable the **Service Instance** flag (to make it a Simple Service Bundle)
- **Fulfillment Item Code**: VoIP PS
- Enable **Orderable**, **Track as Asset**, **Shippable**, and **Bill on Purchase**

p. Click the **Pricing** tab, and from the **Add Fee** drop-down list, select **Recurring Fee**.

q. In the **Price plan** page, provide the following information:

- **Name**: Any name (for example, SmokeTest_Offer_POP)
- **ID**: Any ID
- From the **Price Lists** drop-down list, select **NA Pricelist**.
- From the **Recurring Fee Type** drop-down list, select **Cycle**.
- From the **Period** drop-down list, select **Monthly**.
- Click the **Edit** icon available on the **Price and effective period** section, and add the **Price** (for example, $1.00).

r. Click **Update** to update the price in the offer.

s. Click **Create** to create the smoke test offer.

3. Publish the initiative containing the smoke test offer:

a. From the Launch home page, navigate to **Administration**, and then click **View** available on the **Initiatives** card.

b. Click **...** available under **Actions** of your initiative, select **Change status to**, and then click **Ready to publish**.

c. After the status is updated to **Ready to publish**, click **...**, and then click **Publish**.

d. After the **Publish** action is initiated, two tasks are created to sync the data to Siebel and PDC as per the initiative configuration.

e. After the publish is successfully completed, the two tasks will move to **Success** status and the status of the initiative changes to **In Test**.

Validating the Offer in PDC

To validate the smoke test offer data in PDC:

1. Log in to the Oracle Communications Pricing Design Center (PDC) application using your credentials.
2. From the PDC home page, click **Search Pricing** available on the bottom left of the navigation pane.
3. Click the **Advanced** link.
   The **Advanced Search** page opens.
4. In the **Advanced Search** page:

a. From the **Pricing Component** drop-down list, select **Charge Offer**.

b. From the **Name** drop-down list, select **Starts with**, and in the text box enter the offer name (for example, SmokeTest_Offer).

c. Click **Search**.

5. From the results list, click the name of the test offer.
   The details of the offer are displayed.

6. Validate the details.
7. Check if **em-gateway** and **pricingupdater** are up and running in the environment using the following command:
   `"kubectl get pods -n <namespace>" | grep -e em -e pricingupdater`

Validating the Offer in Siebel

To validate the smoke test offer data in Siebel:

1. Log in to Siebel using your credentials.
2. Click the **Site Map** icon and then select Siebel UI.
3. Search for **Administration - Product**.
4. From the results list, click the offer name (for example, SmokeTest_Offer).
5. Validate if the **Service Instance** check box is selected.
6. Validate the details of the offer.

Validating the Offer in BRM

To validate the smoke test offer data in BRM:

1. Ensure that SQL Developer is installed in your system.
2. Connect with the Oracle Communications Billing and Revenue Management (BRM) DB by clicking **New Collection** and following steps:
   a. Name: BRM PIN DB:
   b. username: pin
   c. password: (password given in file)
   d. hostname: localhost:1521
   e. port: 1521
   f. service name: (Service name given in the Excel file)
3. Run the following query to get the product details:
   `select * from product_t where name ='<product_name>';`
4. Run the following query to get the price of the product and validate it with Siebel and PDC:

```
select SCALED_AMOUNT from RATE_BAL_IMPACTS_T where OBJ_ID0 IN (
select POID_ID0 from RATE_T where RATE_PLAN_OBJ_ID0 IN
( select POID_ID0 from RATE_PLAN_T where PRODUCT_OBJ_ID0 IN
(select POID_ID0 from pin.product_t where name like 'product-name'))))
```

> **Note**
>
> The product name is the smoke test offer name you have created in Launch.

Validating the AIA XREFs

To validate the smoke test offer data in AIA:

1. Ensure that SQL Developer is installed in your system.

2. Connect with the Oracle Communications Application Integration Architecture (AIA) DB by clicking **New Collection** and following steps:
   a. Name: AIA XREF:
   b. username: AIAFP_XREF
   c. password: (password given in file)
   d. hostname: localhost:1521
   e. port: 1521
   f. service name: (Service name given in the Excel file)
3. Run the following query by replacing <BRM POID> with the POID you get from BRM DB:

```
SELECT
    BRM_01,
    COMMON,
    SEBL_01
FROM
    XREF_ITEM_ITEMID
WHERE
    BRM_01 LIKE '0.0.0.1 /product <BRM POID> 0%'

UNION

SELECT
    t1.BRM_01,
    t1.COMMON,
    t1.SEBL_01
FROM
    XREF_ITEM_ITEMID t1
    INNER JOIN XREF_PRICELINETYPE_ID t2 ON t1.COMMON = t2.ITEM_ID_COMMON
WHERE
    t2.BRM_01 LIKE '0.0.0.1 /product <BRM POID> 0%'
```

   Check that SEBL_01, COMMON, and BRM_01 are not null, and validate if the SEBL_01 from the Siebel Administration UI is same.
4. Run the following query to get the ITEM_ID_COMMON, and all four fields should not be null.:

```
SELECT
    BRM_01,
    COMMON,
    SEBL_01,
    ITEM_ID_COMMON
FROM
    XREF_PRICELINE_ID
WHERE
    BRM_01 LIKE '0.0.0.1 /product <BRM POID> 0%'
```

5. Run the following query to get the priceline type:

```
SELECT
    BRM_01,
    COMMON,
```

```
    SEBL_01,
    ITEM_ID_COMMON
FROM
    XREF_PRICELINETYPE_ID
WHERE
    BRM_01 LIKE '0.0.0.1 /product <BRM POID> 0%'
```

6. Run the following query to get the Siebel product reference:

```
SELECT
    t1.ITEM_ID_COMMON,
    t1.LINEPRICETYPECODE
FROM
    XREF_SIEBELPRODUCTEV_118 t1
    INNER JOIN XREF_PRICELINETYPE_ID t2 ON t1.LINEPRICETYPECODE = t2.COMMON
WHERE
    t2.BRM_01 LIKE '0.0.0.1 /product <BRM POID> 0%'
```

Deleting the Initiative From Launch

> **Note**
>
> You must delete the smoke test initiative after successful completion of all the validations.

To delete the smoke test initiative from Launch:

1. Log in to the Launch application using your credentials.
2. Navigate to **Administration**, and then click **View** available on the **Initiative** card.
3. From the results list, click **...** against your smoke test initiative name (for example, SmokeTest_initiative), then select **Change status to**, and then click **In design**.
4. After the status of the initiative changes to **In design**, click **...**, and then select **Delete**.

Creating and Submitting the Order From Siebel (Run Time)

To create and submit an order for the smoke test offer created in Launch:

1. Log in to the Siebel application using your credentials.
   The Siebel home screen opens.
2. In the **Account** field, enter the account name (for example, smoke_test), and then click **Add & Go**.
3. In the **Accounts** page, provide the following information:
   a. From the **Account Type** drop-down list, select **Residential**.
   b. Click the **Address** icon, select the appropriate address from the **Account Addresses** page.
   c. From the **Pricelist** drop-down list, select **NA Pricelist**.
4. From the **Account Summary** drop-down list, select **Contacts**, and then click the **++** icon.
5. In the **First Name** and **Last Name** fields, enter any name (for example, smoke_test).

6. Click the **Settings** icon and select **Save**.
7. In the **Billing Profile** section, click **+**, and provide the following information:
   - **Payment Method**: Bill Me
   - **Bill Type**: Summary
   - **Frequency**: Monthly
   - **Bill Media**: Email
   - Click the **Settings** icon and select **Save**.
8. In the **Orders** section, click the **Settings** icon, and then select **New Record**.
9. Click the link in the **Order #** column.
   The **Order Details** page opens.
10. In the **Due** field, click the **Calendar** icon, and select the date same as the **Order Date** field.
11. In the **Line Items** section, enter the name of the published offer from Launch, validate the price for that service, and add service ID for service bundles.
12. Click **Submit**.
13. After validating the order in Oracle Communications Application Integration Architecture (AIA), refresh the page, and verify if the order status is changed to **Completed**.

Validating the Order in AIA

To validate the order data in AIA:

1. Log in to the Oracle Communications Application Integration Architecture (AIA) application using your credentials.
2. Click the **aia-fp** icon, expand **SOA**, and then select **soa-infra(soa_server1)**.
3. Click the **Flow Instances** tab, click **Recent Instances** and verify composite state [running or completed or failed].
4. For the successful orders, the **Flow State** column displays **Completed**.

Validating the Order in OSM

To validate the order data in OSM:

1. Log in to the Oracle Communications Order and Service Management (OSM) application using your credentials.
2. Click the **Query** tab.
3. In the **Order Number** field, enter the order number generated in Siebel.
4. From the **Order State** drop-down list, select **All**.
5. From the **Failure** drop-down list, select **All**, and then click **Search**.
6. The system displays both COM_ and SOM_ instances as **Completed**.

Validating the Order in BRM

To validate the order data in BRM:

1. Log in to the Oracle Communications Billing and Revenue Management (BRM) application using your credentials.

2. From the **Search** drop-down list, select **Accounts**, and then click **Search**.
3. From the results list, select your account, and then click **Open**.
   The Account Details page opens
4. Verify the account details, such as Services, First Name, Last Name, and so on are created correctly in BRM.
5. Click **Assets** tab, verify the products and services, and validate the Service status is **Active**.
6. Navigate to the **Bills** tab, and verify the total amount due.
7. Click **Bill Now** to generate the bill in BRM.
8. Click **OK**.

Validating the Account in ECE Cache

Check if the prices are in the Oracle Communications Elastic Charging Engine (ECE) cache:

1. Run the following command to check the pod ID for PDC and ECE pods in namespace.
   `kubectl get pods -n brmcn15 | grep -e pdc -e upd`
2. Run the following command to log in to the pod:
   `kubectl exec -it ecs-0 -n brmcn15 -- bash`
3. Go to temp and run the following command to run the shell script:
   `[eceuser@ecs-0 temp]$ sh /home/charging/opt/ECE/oceceserver/bin/query.sh`
4. Query for the ChargeOffering using the query `CohQL> Select value() from ChargeOffering`

   > **Note**
   > - The above query ensures the offer is present and externalId is populated.
   > - For detailed information, run the `query.sh` script in ECE pod and query for ChargeOffering, Status, and so on.

5. `Select value() from Customer where customerId='<cust_id_from_brm_acc_url>'`
   This ensures the account is present in the ECE cache.

Validating the Siebel Order Completion

To validate the Siebel order completion:

1. Log in to the Siebel application using your credentials.
2. Open your account from the **Accounts** page.
3. From the **Installed Assets** section, verify that all assets included in the order are purchased.
4. From the **Billing Items** section, verify that all items included in the order are billed.
5. From the **Orders** section, click the link from the **Order #** column.
   The **Order Details** page opens.
6. Verify that the order status is **Complete**.
7. Verify the First Name and Last Name in Siebel and Billing Care.

8. From the **Billing Profile** section, click the **Name** link, and validate the bill details in Siebel UI.

5

Troubleshooting

This chapter provides information about issues that you may face while deploying the Digital Business Experience solution.

Troubleshooting for the Solution Components

This section provides troubleshooting instructions for the components of the Digital Business Experience solution.

To troubleshoot the deployment for the solution components, refer to the following:

- See Troubleshooting for detailed instructions about troubleshooting Launch and CX Industries Framework applications.
- See Troubleshooting Your BRM Cloud Native Deployment for detailed instructions about troubleshooting BRM.
- See Troubleshooting the ECE Installation for detailed instructions about troubleshooting ECE.
- See Troubleshooting OAP for detailed instructions about troubleshooting OAP.
- See Troubleshooting Issues with the Scripts for detailed instructions about troubleshooting SCD.
- See Debugging and Troubleshooting for detailed instructions about troubleshooting OSM.
- See Troubleshooting Order-to-Activate Cartridges for detailed instructions about troubleshooting O2A.
- See Troubleshooting Installation and Configuration for Siebel CRM for detailed instructions about troubleshooting Siebel.
- See Troubleshooting Oracle Data Integrator for detailed instructions about troubleshooting ODI.
- See Troubleshooting Issues for detailed instructions about troubleshooting AIA.

6

Downloading and Deploying the Reference Solution

This chapter provides information about downloading and deploying the Digital Business Experience Reference Solution.

Before you start to download and deploy the reference solution:

1. Learn about the reference solution. See About the Reference Solution in *Oracle Communications Digital Business Experience Concepts* for more details.
2. Learn about the reference product models and seed data available in the reference solution. See Reference Product Models and Seed Data in *Oracle Communications Digital Business Experience Concept to Market Guide* for more details.
3. Learn about the order to cash reference library, which consists of a few sample preconfigured order-to-cash end-to-end features. See Using the Order to Cash Reference Library in *Oracle Communications Digital Business Experience Order to Cash Implementation Guide* for more details.
4. Download and install Oracle Communications Solution Test Automation Platform (STAP) in your Digital Business Experience environment. See Installing STAP for more details.

   > **Note**
   >
   > The reference solution is packaged with STAP, so you must install STAP before you can deploy the reference solution package.

5. Set up STAP Design Experience (STAP DE). See Setting Up The STAP Design Experience for more information.

Downloading the Reference Solution Package

You must download the packages in the following order:

- Reference Solution package, which contains scripts for:
  - Prerequisite and Seed data: Preconfigured Launch, Siebel, BRM, and PDC data.
  - STAP helper scripts: These scripts help you to publish various scenarios to STAP.
  - Design-time product models: Includes all the business-to-business and business-to-customer product models.
  - Smoke test suite: Includes a smoke test scenario for your Digital Business Experience environment.
  - Feature extensions: These are Siebel and BRM custom extensions available under the `B2BSolutionPack` folder of the `oc-dbe-rs-<version>.zip` file. To apply these extensions to your Digital Business Experience environment, follow the Standard Operating Procedures (SOPs) documented under the `B2BSolutionPack` folder. The `B2BSolutionPack` folder also contains details about the automation test cases covered for various features. To run the automation test cases using STAP, see the

TestPlan_Execution_SOP SOP available at `oc-dbe-rs-<version>/Test Automation/TestPlan_Execution_SOP.pdf`.

- Order-to-cash test library (hosted with STAP), which consists of the end-to-end run-time test catalog.

To download the reference solution package:

1. Download the reference solution pack from the Oracle software downloads website, located at:

   https://www.oracle.com/downloads/applications/communications/dbe-downloads.html

2. Run the following commands from the STAP environment to unzip the reference solution pack:

   ```
   unzip oc-dbe-rs-<version>.zip
   ```

   ```
   unzip -d oc-dbe-rs-<version>/B2CSolutionPack/ oc-stap-otc-testlib-<version>.zip
   ```

   > **Note**
   >
   > You can delete the zip file after unzipping it.

3. Edit the `stap.properties` file:

   ```
   cd oc-dbe-rs-<version>/B2CSolutionPack/
   vi stap.properties
   Replace:
   export JAVA_HOME="/home/java/jdk-xx.x.x" with the correct path to JDK
   export STAP_HOME="/home/STAP-DE"  with the correct path to STAP-DE jar
   For example:
   export JAVA_HOME="/home/user/java/jdk-21.0.6"
   export STAP_HOME="/home/user/STAP-DE"
   ```

4. Edit the config files for Publish-Automation.
   Set up the **Tdaas** and **TES** environment files to point them to the STAP deployment to which you want to publish them:

   ```
   /oc-dbe-rs-<version>/B2CSolutionPack/Publish-Automation/config/environments/TdaasEnvironment.properties
   /oc-dbe-rs-<version>/B2CSolutionPack/Publish-Automation/config/environments/TaaS-TES-environment.properties
   ```

Prerequisites

The prerequisites for installing the reference solution are:

- Ensure that the `ENABLE_PRODUCT_ATTRIBUTE_VALIDATION` property is **disabled** in Oracle Communications Order to Activate (O2A) application.

- Ensure that Oracle Communications Order and Service Management (OSM) image is extended to include UIM and TOM emulators if actual UIM and TOM applications are not available in the Digital Business Experience stack.
- Verify all the REST API end points for various applications and SSH keys, ensuring proper connectivity.
- Ensure that you enabled the REST calls for BCWS and TMF. See Configuring REST Services in *Oracle Communications Billing and Revenue Management Cloud Native Deployment Guide* for detailed instructions about configuring the rest services in BRM and PDC.
- Edit the config files for pre-automation.

  ```
  oc-dbe-rs-<version>/B2CSolutionPack/Pre-Automation/config/publish/
  tdaasEnvironment.properties
  oc-dbe-rs-<version>/B2CSolutionPack/Pre-Automation/config/publish/
  environment.properties
  oc-dbe-rs-<version>/B2CSolutionPack/Pre-Automation/config/publish/
  persistence-volume-environment.properties
  oc-dbe-rs-<version>/B2CSolutionPack/Pre-Automation/config/publish/publish-
  automation.properties
  oc-dbe-rs-<version>/B2CSolutionPack/Pre-Automation/config/environments/* --
  > all files
  oc-dbe-rs-<version>/B2CSolutionPack/path.properties
  ```

- Run the following command to update the scenario files:

  ```
  sh replace_path.sh
  ```

- Run the `publish_workspace.sh` script for pre-automation.

  ```
  sh publish_workspace.sh -w Pre-Automation
  ```

- Run the `run_scenarios.sh` script for pre-automation.

  ```
  sh run_scenarios.sh -w Pre-Automation
  ```

- Run the Seed Data (SD) Automation:
  - Edit the config files for SD Automation as follows:

    ```
    oc-dbe-rs-<version>/B2CSolutionPack/SD-Automation/config/publish/
    tdaasEnvironment.properties
    oc-dbe-rs-<version>/B2CSolutionPack/SD-Automation/config/publish/
    environment.properties
    oc-dbe-rs-<version>/B2CSolutionPack/SD-Automation/config/publish/
    persistence-volume-environment.properties
    oc-dbe-rs-<version>/B2CSolutionPack/SD-Automation/config/publish/
    publish-automation.properties
    oc-dbe-rs-<version>/B2CSolutionPack/SD-Automation/config/environments/*
    --> all files
    oc-dbe-rs-<version>/B2CSolutionPack/token.properties
    oc-dbe-rs-<version>/B2CSolutionPack/SD-Automation/scenarios/5.AIA/data/
    config.properties
    ```

- Run the following command to check if the token refresh is working fine:

  ```
  sh refresh_token.sh
  ```

- Run the following commands for SD Automation:

  ```
  sh publish_workspace.sh -w SD-Automation


  sh run_scenarios.sh -w SD-Automation
  ```

- Verify that the seed data has been created correctly in Launch, PDC, and BRM. See About the Seed Data in *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* for more information.
- Check if the destinations for PDC and Siebel are created in Launch.
- Verify if the pdc-test destination has destination exclusion rule to avoid publishing bundles to PDC as it is not supported by Digital Business Experience.
- Ensure that the **Lifecycle Status** for the destinations in the Launch UI is **Ready to publish**.

Deploying the Reference Solution Package

To deploy the reference solution package:

1. Run the Design-Time (DT) automation:
   a. Edit the config files for DT automation:

      ```
      oc-dbe-rs-<version>/B2CSolutionPack/DT-Automation/config/publish/
      tdaasEnvironment.properties
      oc-dbe-rs-<version>/B2CSolutionPack/DT-Automation/config/publish/
      environment.properties
      oc-dbe-rs-<version>/B2CSolutionPack/DT-Automation/config/publish/
      persistence-volume-environment.properties
      oc-dbe-rs-<version>/B2CSolutionPack/DT-Automation/config/publish/
      publish-automation.properties
      oc-dbe-rs-<version>/B2CSolutionPack/DT-Automation/config/environments/*
      --> all files
      ```

   b. Update the `token.properties` file with the newly generated token, then run the following command to update the scripts:

      ```
      sh refresh_token.sh
      ```

   c. Run the following commands for DT automation:

      ```
      sh publish_workspace.sh -w DT-Automation
      ```

   d. Run the script in the following order to publish all the product models required for DT automation:

      ```
      sh run_dt_automation.sh MobileProductModels
      sh run_dt_automation.sh BroadbandProductModels
      sh run_dt_automation.sh DigitalTVProductModels
      ```

```
sh run_dt_automation.sh HomePhoneProductModels
sh run_dt_automation.sh FamilySharePlan
sh run_dt_automation.sh QuadPlayProductModels
sh run_dt_automation.sh TriplePlayProductModels
sh run_dt_automation.sh DualPlayProductModels
sh run_dt_automation.sh AddPromotionBundles
```

> **Note**
>
> You must follow the publish order mentioned above because models in DT automation are interdependent.

2. Run the Run-Time (RT) automation:
   a. Edit the config files for RT automation.

```
oc-dbe-rs-<version>/B2CSolutionPack/RT-Automation/config/publish/
tdaasEnvironment.properties
oc-dbe-rs-<version>/B2CSolutionPack/RT-Automation/config/publish/
environment.properties
oc-dbe-rs-<version>/B2CSolutionPack/RT-Automation/config/publish/
persistence-volume-environment.properties
oc-dbe-rs-<version>/B2CSolutionPack/RT-Automation/config/publish/
publish-automation.properties
oc-dbe-rs-<version>/B2CSolutionPack/RT-Automation/config/environments/*
--> all files
```

   b. Run the following command for RT automation:

```
sh publish_workspace.sh -w RT-Automation
```

After the RT automation is published, you can run either full or partial automation based on product models as follows:

- To run all the run-time scenarios, run the following command:

```
sh run_rt_automation.sh --all
```

  or

- To run bundled sample orders (functional smoke test) scenario, run the following command:

```
sh run_rt_automation.sh --scenario_name/subfolder_name
For example: sh run_rt_automation.sh --MobileProductModels/
Supremo5GUnlimited/SampleOrder
```

  or

- To run a specific run-time scenario, you can run the script with the corresponding path to the scenario. You can see the list of scenarios by passing the `--list` option to the script:

```
sh run_rt_automation.sh --list
For example:
sh run_rt_automation.sh --BroadbandProductModels/SupremoBroadbandBasic/
```

```
SalesOrderCreation
sh run_rt_automation.sh --BroadbandProductModels/SupremoBroadbandGigabit/
CancelOrderBeforePONR
```

Troubleshooting the Reference Solution

This section provides troubleshooting information for problems while deploying the reference solution.

Problem: BRM Seed Data Fails to Load

If BRM data fails to load:

- Load the following files from the BRM scenario data folder in the cm pod:
  - `customservices_proximity.podl`
  - `customservices_OTT_VOIP_dtv_dmusic.podl`

  See BRM PDC Seed Data in *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* for a list of supported BRM Services, Events, and ServiceEvent maps.
- Sync the new events and services by restarting the cm, dm, and syncpdc pods.

Problem: PDC Seed Data Fails to Import

If PDC data fails to import, create the ServiceEvent map manually from the PDC UI. See BRM PDC Seed Data in *Oracle Communications Digital Business Experience Concept to Market Implementation Guide* for a list of supported PDC Services, Events, and ServiceEvent maps.

Problem: Reimporting Service Event Map Files for PDC

This section provides a solution for running and reimporting Service Event Map files (attributeSpecMap.xml) multiple times in Oracle Communications Pricing Design Center (PDC) environments.

Issue: You need to import the Service Event Map file (`attributeSpecMap.xml`) multiple times during configuration, either in a new or an existing PDC environment.

Scenario 1: New PDC Environment
You have a newly installed PDC environment and need to import the `attributeSpecMap.xml` file multiple times.

Resolution

- Import the Automated ASM file (`attributeSpecMap.xml`) as needed.
- Use the `-ow` (Overwrite) and `-ignoreID` command-line options to avoid import conflicts:
  - `-ow`: Overwrites matching components in the PDC database.
  - `-ignoreID`: Ignores internal IDs to prevent ID conflicts.

Import Command Example

To reimport the same ASM file, use the following Helm upgrade command configuration:

```
ocpdc:
  configEnv:
    importExport:
      IE_Operation: "import"
      IE_Component: "config"
      IE_File_OR_Dir_Name: attributeSpecMap.xml
      extraCmdLineArgs: "-ow -ignoreID"
      IE_LogLevel: WARNING
      IE_PersistLog: failed
      persistIELogs: "enabled/all"
```

> **Note**
>
> No additional steps are required in a fresh installation; multiple imports will not cause issues when using these options.

Scenario 2: Existing PDC Environment
You are working with an existing PDC environment where the Service Event Map has already been loaded, and you need to load additional or updated events for the same service.

Workaround

1. Export the already loaded map.
   - Export the current Service Event Map from the existing PDC system (for example, `E_ASM.xml`).
2. Compare with the new map.
   a. Obtain the new attribute specification map file to be imported (for example, `N_ASM.xml`).
   b. Manually compare `E_ASM.xml` and `N_ASM.xml` to identify differences or new entries.
3. Create a merged map.
   - Merge the required changes (new or updated entries) from `N_ASM.xml` into `E_ASM.xml` to create a combined file (for example, `E_ASM_merged.xml`).
4. Reimport the merged file.
   - Use `-ow -ignoreID` options while importing to overwrite existing entries and ignore internal IDs.

Import Command Example

- `E_ASM.xml`: Contains the existing event map (for example, for service TelcoGsmTelephony). The following is a sample `E_ASM.xml` file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><cim:ConfigObjects
xmlns:cim="http://xmlns.oracle.com/communications/platform/model/
Config"><attributeSpecMaps xmlns:cim="http://xmlns.oracle.com/
communications/platform/model/Config">
<name>ConvergentVoice_ASM</name>
<internalId>4fe9a625-839e-4135-a6cd-9818feb75ce6</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
```

```
<eventRUMSpec>
<name>EventBillingCycleRolloverMonthly_ERS</name>
<internalId>df0080b0-7e52-4bb7-a533-df60b02040dd</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventBillingCycleRolloverMonthly</eventSpecName>
<rumSpec>
<rumExpression>
<numericRumExpression>
<value>1.0</value>
</numericRumExpression>
</rumExpression>
<rumName>Occurrence</rumName>
</rumSpec>
</eventRUMSpec>
<eventRUMSpec>
<name>EventBillingProductFeeCycleCycle_forward_monthly_ERS</name>
<internalId>43f53235-57db-455e-9033-a8ec33e15a58</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventBillingProductFeeCycleCycle_forward_monthly</
eventSpecName>
<rumSpec>
<rumExpression>
<numericRumExpression>
<value>1.0</value>
</numericRumExpression>
</rumExpression>
<rumName>Occurrence</rumName>
</rumSpec>
</eventRUMSpec>
<eventRUMSpec>
<name>EventDelayedSessionTelcoGsm_ERS</name>
<internalId>ccdf11ed-aac6-45eb-9487-ee77043a3643</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventDelayedSessionTelcoGsm</eventSpecName>
<rumSpec>
<rumExpression>
<binaryRUMExpression>
<leftOperand>
<eventFieldExpression>
<fieldName>EventDelayedSessionTelcoGsm.END_T</fieldName>
</eventFieldExpression>
</leftOperand>
<rightOperand>
<eventFieldExpression>
<fieldName>EventDelayedSessionTelcoGsm.START_T</fieldName>
</eventFieldExpression>
</rightOperand>
<operation>SUBTRACT</operation>
</binaryRUMExpression>
</rumExpression>
<rumName>Duration</rumName>
</rumSpec>
</eventRUMSpec>
```

```
<productSpecName>TelcoGsmTelephony</productSpecName>
</attributeSpecMaps></cim:ConfigObjects>
```

- `N_ASM.xml`: Contains new or modified events (for example, added Purchase Fee event, updated RUM for "Event delayed session telco GSM"). The following is a sample `N_ASM.xml` file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><cim:ConfigObjects
xmlns:cim="http://xmlns.oracle.com/communications/platform/model/
Config">            <attributeSpecMaps xmlns:cim="http://xmlns.oracle.com/
communications/platform/model/Config">
   <name>ConvergentVoice_ASM</name>
   <internalId>4fe9a625-839e-4135-a6cd-9818feb75ce6</internalId>
   <priceListName>Default</priceListName>
   <obsolete>false</obsolete>
   <eventRUMSpec>
      <name>EventDelayedSessionTelcoGsm_ERS</name>
      <internalId>ccdf11ed-aac6-45eb-9487-ee77043a3643</internalId>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <eventSpecName>EventDelayedSessionTelcoGsm</eventSpecName>
      <rumSpec>
         <rumExpression>
            <binaryRUMExpression>
               <leftOperand>
                  <eventFieldExpression>
                     <fieldName>EventDelayedSessionTelcoGsm.END_T</
fieldName>
                  </eventFieldExpression>
               </leftOperand>
               <rightOperand>
                  <eventFieldExpression>
                     <fieldName>EventDelayedSessionTelcoGsm.START_T</
fieldName>
                  </eventFieldExpression>
               </rightOperand>
               <operation>SUBTRACT</operation>
            </binaryRUMExpression>
         </rumExpression>
         <rumName>Duration_M</rumName>
      </rumSpec>
   </eventRUMSpec>
   <eventRUMSpec>
      <name>EventBillingProductFeePurchase_ERS</name>
      <internalId>d090943b-1287-48db-a3e3-5f971a1753cf</internalId>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <eventSpecName>EventBillingProductFeePurchase</eventSpecName>
      <rumSpec>
         <rumExpression>
            <numericRumExpression>
               <value>1.0</value>
            </numericRumExpression>
         </rumExpression>
         <rumName>Occurrence</rumName>
      </rumSpec>
```

```
   </eventRUMSpec>
<productSpecName>TelcoGsmTelephony</productSpecName>
</attributeSpecMaps></cim:ConfigObjects>
```

- `E_ASM_merged.xml`: Manually merged file containing all required events and updates. The following is a sample `E_ASM_merged.xml` file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><cim:ConfigObjects
xmlns:cim="http://xmlns.oracle.com/communications/platform/model/
Config"><attributeSpecMaps xmlns:cim="http://xmlns.oracle.com/
communications/platform/model/Config">
<name>ConvergentVoice_ASM</name>
<internalId>4fe9a625-839e-4135-a6cd-9818feb75ce6</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventRUMSpec>
<name>EventBillingCycleRolloverMonthly_ERS</name>
<internalId>df0080b0-7e52-4bb7-a533-df60b02040dd</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventBillingCycleRolloverMonthly</eventSpecName>
<rumSpec>
<rumExpression>
<numericRumExpression>
<value>1.0</value>
</numericRumExpression>
</rumExpression>
<rumName>Occurrence</rumName>
</rumSpec>
</eventRUMSpec>
<eventRUMSpec>
<name>EventBillingProductFeeCycleCycle_forward_monthly_ERS</name>
<internalId>43f53235-57db-455e-9033-a8ec33e15a58</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventBillingProductFeeCycleCycle_forward_monthly</
eventSpecName>
<rumSpec>
<rumExpression>
<numericRumExpression>
<value>1.0</value>
</numericRumExpression>
</rumExpression>
<rumName>Occurrence</rumName>
</rumSpec>
</eventRUMSpec>
<eventRUMSpec>
<name>EventDelayedSessionTelcoGsm_ERS</name>
<internalId>ccdf11ed-aac6-45eb-9487-ee77043a3643</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventDelayedSessionTelcoGsm</eventSpecName>
<rumSpec>
<rumExpression>
<binaryRUMExpression>
<leftOperand>
```

```
<eventFieldExpression>
<fieldName>EventDelayedSessionTelcoGsm.END_T</fieldName>
</eventFieldExpression>
</leftOperand>
<rightOperand>
<eventFieldExpression>
<fieldName>EventDelayedSessionTelcoGsm.START_T</fieldName>
</eventFieldExpression>
</rightOperand>
<operation>SUBTRACT</operation>
</binaryRUMExpression>
</rumExpression>
<rumName>Duration</rumName>
</rumSpec>
<rumSpec>
<rumExpression>
<binaryRUMExpression>
<leftOperand>
<eventFieldExpression>
<fieldName>EventDelayedSessionTelcoGsm.END_T</fieldName>
</eventFieldExpression>
</leftOperand>
<rightOperand>
<eventFieldExpression>
<fieldName>EventDelayedSessionTelcoGsm.START_T</fieldName>
</eventFieldExpression>
</rightOperand>
<operation>SUBTRACT</operation>
</binaryRUMExpression>
</rumExpression>
<rumName>Duration_M</rumName>
</rumSpec>
</eventRUMSpec>
<eventRUMSpec>
<name>EventBillingProductFeePurchase_ERS</name>
<internalId>d090943b-1287-48db-a3e3-5f971a1753cf</internalId>
<priceListName>Default</priceListName>
<obsolete>false</obsolete>
<eventSpecName>EventBillingProductFeePurchase</eventSpecName>
<rumSpec>
<rumExpression>
<numericRumExpression>
<value>1.0</value>
</numericRumExpression>
</rumExpression>
<rumName>Occurrence</rumName>
</rumSpec>
</eventRUMSpec>
<productSpecName>TelcoGsmTelephony</productSpecName>
</attributeSpecMaps></cim:ConfigObjects>
```

To reimport the merged file, configure the Helm upgrade as follows:

```
ocpdc:
  configEnv:
    importExport:
```

```
IE_Operation: "import"
IE_Component: "config"
IE_File_OR_Dir_Name: E_ASM_merged.xml
extraCmdLineArgs: "-ow -ignoreID"
IE_LogLevel: WARNING
IE_PersistLog: failed
persistIELogs: "enabled/all"
```

Note

- Always create backups of configuration files before performing merges or imports.
- Manual merging is required to ensure that the Reference Solution test cases run without issues, as well as retain any current service event maps that are already available.

Problem: Pre Seed Model Fails to Import or Publish During SD Automation

If pre seed model fails to import or publish, verify if the pre seed model is available under the **Initiatives** tab in the Launch UI. If not, rerun the import or publish jobs accordingly.

Problem: AIA DVM Script Fails to Run During SD Automation

If the AIA DVM script (`dvm_update.sh`) fails to run, check the `config.properties` file available in the `AIA_FILE_TARGET_DIR` folder from `path.properties`, and rerun the `dvm_update.sh` script manually from the bastion host from the `AIA_FILE_TARGET_DIR` folder.

Problem: restartDomain.sh Script Fails to Run During SD Automation

If the AIA pods are not starting or the `restartDomain.sh` script fails to run, rerun the `restartDomain.sh` script from the `AIA_FILE_TARGET_DIR` folder and check the AIA WebLogic console to start `soa_server1` and `soa_server2` properly.

Problem: Product Models Fail to Import or Publish During DT Automation

If product models fail to import or publish, verify if the product model is available under **Initiatives** tab in the Launch UI. Check if there are any failed import or publish jobs and rerun those jobs from the Launch UI in the correct order.

For information about the correct order, see the "Prerequisite Setup for Product Models" table from the "Reference Product Models and Seed Data" chapter in *Oracle Communications Digital Business Experience Concept to Market Implementation Guide*.

Problem: Sales Order Status Verification Failed in Siebel During RT Automation

Issue: When you submit a sales order from Siebel, it fails intermittently due to AIA or BRM-related issues.

The failure occurs at the validation step with the following validation timeout error:

```
Timeout - endAfter reached: test condition not satisfied within the wait
duration
```

To verify the intermittent issues using the AIA Enterprise Manager (EM) console:

1. Login to the EM Console using your credentials.
2. Navigate to the **soa-infra** server.
3. Search for the order and look for instances corresponding to the order number.
4. Identify failures and click **Flow ID** for failed instances to view detailed errors. Common error messages include failure of composite `ProcessFulfillmentOrderBillingOSMCFSCommsJMSConsumer [1.0]` with different errors.

   Sample error message:

   ```
   [Exception in oneway execution]Unexpected exception in one-way operation
   "processBilling" on reference
   "ProcessFulfillmentOrderBillingBRMCommsProvABCSImpl".Possible Fix:Check
   whether the reference service is properly configured and running or look
   at exception for analyzing the reason or contact Oracle Support Services.
   Cause: ORABPEL-02199

   JTA transaction is not in active state.
   The transaction became inactive when executing activity "" for instance
   "342,494", bpel engine can not proceed further without an active
   transaction. please debug the invoked subsystem on why the transaction is
   not in active status. the transaction status is "ROLLEDBACK".
   The reason was The execution of this instance "342494" for process
   "ProcessFulfillmentOrderBillingBRMCommsProvABCSImplMasterProcess" is
   supposed to be in an active jta transaction, the current transaction
   status is "ROLLEDBACK", the underlying exception is
   "javax.transaction.xa.XAException: Internal error: XAResource 'eis/BRM' is
   unavailable" .
   Consult the system administrator regarding this error.
   ```

Solution:

- Rerun the failed scenario using the following command:

  ```
  sh run_rt_automation.sh <scenario_path>
  ```

- If the issue still exists, create a service ticket for more assistance.

Problem: Bill Generation Failed in BRM During RT Automation

Issue: Bill generation fails in Oracle Communications Billing Revenue Management (BRM) due to intermittent environment issues.

The failure occurs in the `generate-bill.brm.case` file with a 504 response as follows:

```
"{"extension":null,"errorCode":10013,"errorMessage":"The operation has timed
out!","isValidationError":false,"paramInfo":[]}"
```

Solution:

- Rerun the failed scenario using the following command:

  ```
  sh run_rt_automation.sh <scenario_path>
  ```

- If the issue still exists, create a service ticket for further assistance.

Problem: Issue While Generating Triggers in Siebel During RT Automation

Issue: If you do not configure triggers correctly in Siebel, the account synchronization requests fail to reach BRM.

The failure occurs in the following scenarios:

- Updating first name and last name after the sales order is created for Supremo TV Basic.
- Updating address details after the sales order is created for Supremo TV Basic.
- Suspending services and terminating an account.

Solution:

- Verify that **Enable Account Status Sync** is configured correctly.
- If the issue exists, check the Siebel server logs for errors and work accordingly based on the error.
- If the issue still persists, create a service ticket for further assistance.

Problem: Usage Validation Issues During RT Automation

Issue: If the required usage consumption configuration is not configured correctly in Oracle Communications Elastic Charging Engine (ECE) and Oracle Communications Offline Mediation Controller, the usage scenarios fail.

The failure occurs in all the following zone-based pricing and special rating usage scenarios:

```
MobileProductModels/Supremo5GLite/ZoneBasedPricingUsage/ItalyAndSpain
MobileProductModels/Supremo5GPremium/ZoneBasedPricingUsage/GreeceAndROW
MobileProductModels/Supremo5GUnlimited/ZoneBasedPricingUsage/Mexico&Germany
HomePhoneProductModels/SupremoStarterHomePhone/ZoneBasedPricingUsage/
GreeceAndGermany
HomePhoneProductModels/SupremoStarterHomePhone/ZoneBasedPricingUsage/
ItalyIndiaAndLocal
HomePhoneProductModels/SupremoStarterHomePhone/SpecialRating/
ModifySpecialRatingListAndRunUsage/DeleteAndAddSpecialRatingNumbers
HomePhoneProductModels/SupremoStarterHomePhone/SpecialRating/
NonSpecialRatingNumbersUsage
HomePhoneProductModels/SupremoStarterHomePhone/SpecialRating/
SpecialRatingNumbersUsage
HomePhoneProductModels/SupremoPremiumHomePhone/ZoneBasedPricingUsage/
SpainAndMexico
HomePhoneProductModels/SupremoPremiumHomePhone/SpecialRating/
ModifySpecialRatingListAndRunUsage/AddNewSpecialRatingNumbers
HomePhoneProductModels/SupremoPremiumHomePhone/SpecialRating/
ModifySpecialRatingListAndRunUsage/DeleteSpecialRatingNumbers
HomePhoneProductModels/SupremoPremiumHomePhone/SpecialRating/
SpecialRatingNumbersUsage
```

Solution:

- Verify if all accounts are available in ECE.
  If accounts are missing, validate the BRM to AIA integration and ensure account synchronization is correctly configured.
- Check EPDC pod logs for any builder-related failures, which may occur due to improper synchronization between ECE and Offline Mediation Controller.
  Restart the ECE and Offline Mediation Controller services using the below command:

  ```
  kubectl rollout restart sts ecs -n <Monetisation_namespace>
  ```

- Ensure all required configurations are correctly set up in Offline Mediation Controller as per the deployment and integration guidelines. For more information on the deployment and integration guidelines, see *Oracle Communications Offline Mediation Controller Cloud Native Installation and Administration Guide*.
- If the issue still exists, create a service ticket for further assistance.