



Migrating WebLogic Server to Oracle Cloud Infrastructure

Migrating a WebLogic Server domain and applications from on-premises to Oracle Cloud Infrastructure

March 16, 2020 | Version 1.00
Copyright © 2020, Oracle and/or its affiliates



1 PURPOSE STATEMENT

This migration playbook includes an overview of proven topologies and methods to reduce the effort of moving WebLogic Server workloads to the cloud so you can realize the benefits of using cloud infrastructure. It is intended solely to help you assess the business benefits of migrating existing WebLogic applications to Oracle Cloud Infrastructure and to plan your IT projects.

2 DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.



TABLE OF CONTENTS

1	Purpose Statement	1
2	Disclaimer	1
3	Pre-requisites	3
4	Terminology	3
5	Before you start	3
6	Process overview	3
7	Assessing Requirements	4
7.1	Assess the type of domain to be migrated	4
7.2	Assess the number of Managed Servers and nodes to provision	4
7.3	Assess what data sources need to be migrated.	5
7.4	Assess other WebLogic Server services	5
8	Setting up the environment on Oracle Cloud Infrastructure.	5
8.1	Create a Compartment	5
8.2	Create the Virtual Cloud Network (VCN)	5
8.2.1	Create a VCN	6
8.3	Create a Database Subnet	6
8.3.1	Create a Security list for the database subnet	6
8.3.2	Create the private database subnet	6
8.4	Create an Internet Gateway	7
8.5	Create a key store	7
9	Provision the database(s) on OCI	7
10	Migrate on-premises Database(s)	7
11	Deploying the WebLogic Server infrastructure with the OCI Marketplace image	8
12	Database Connection string for DB VMs	9
13	Accessing the WebLogic Admin console	9
13.1	WebLogic Server deployed in a public subnet:	9
13.2	WebLogic Server deployed in a private subnet	10
14	Migrating a domain	10
14.1	Migrate resources and applications to the domain provisioned on OCI	11
14.1.1	Recreate the source domain resources	11
14.1.2	Deploy the application manually	11
14.1.3	Deploy Apps from JDeveloper (or Eclipse)	12
14.1.4	Migrate a domain to the Marketplace image domain with WebLogic Deploy Tooling	16
14.2	Update existing tools to create and deploy a custom domain in the provisioned infrastructure.	21
14.2.1	Extract the domain metadata	21
15	Scaling managed servers	22
15.1	Scaling the number of WebLogic Managed Servers by scaling the number of nodes	22
15.2	Scaling by adding Managed Servers on the provisioned nodes	22
16	Other Documentation	22



3 PRE-REQUISITES

- This guide assumes you have access to an Oracle Cloud Infrastructure account, with proper permissions to provision and setup resources.
- This guide also requires the use of the Cloud Shell, alternatively you can use the OCI CLI See the [quickstart guide](#) and make sure you upload the public key generated by the setup process to the OCI web console (Top right Profile icon -> User Settings -> API Keys -> Add Public Key).

4 TERMINOLOGY

- A WebLogic Server (WLS) Domain is a compartment concept that includes Managed Servers (Java Virtual Machines) and Resources (Database connections, libraries, etc.).
- Managed Servers are either standalone or organized in clusters, and run on Machines (underlying provisioned machine, whether VM or Bare Metal).
- Applications are deployed in a domain and assigned a Target (Managed Servers or cluster). Similarly, resources are made available on specific Targets, and one needs to make sure that resources needed by an application are available on the application Target.
- Java Required Files (JRF) is a set of components (not part of the base WLS installation) providing functionalities for Oracle Business applications and frameworks based on Oracle Fusion Middleware (ADF, SOA, etc.)

5 BEFORE YOU START

There are many ways a WebLogic Domain and applications can be deployed on a given infrastructure. This guide aims at covering the common scenarios to migrate a WLS Domain to Oracle Cloud Infrastructure.

The WebLogic Server stack (available on the Oracle Cloud Infrastructure Marketplace) provides a way to quickly provision a WebLogic Server installation on multiple machines (VMs), and optionally setup a load balancer targeting the resulting topology. A default domain configured to run a single Managed Server per VM is created and ready for applications to be deployed.

The guide covers 2 common migration scenarios based on this WLS installation:

- Re-deploying on-premises applications and resources into the newly created domain on OCI, which can be performed using the following methods:
 - Manually, through the WebLogic Admin Console
 - Using JDeveloper deployment tools
 - Using WebLogic Deploy Tooling (WLDT)
- Update your existing WLS domain creation and modification tooling used to deploy domains on-premises (such as WebLogic Scripts, or WebLogic Deploy Tooling model files), and re-target them at the newly provisioned Oracle Cloud infrastructure.

Select the method that best suit the way you currently deploy WLS domains and applications.

6 PROCESS OVERVIEW

- Assessing Requirements: assess the type of domain to be migrated, number of Managed Servers to provision, and data sources to be migrated
- Set up the environment on Oracle Cloud Infrastructure: setup a compartment, key store, virtual network and subnets as needed
- Provision the database(s) on OCI
- Migrate on-premises Database(s) to the OCI Database(s)
- Deploy the WebLogic Server installation with the OCI Marketplace image
- Deploy resources and applications on the newly provisioned infrastructure



7 ASSESSING REQUIREMENTS

Below are the key requirements to assess in order to successfully deploy the right infrastructure.

7.1 Assess the type of domain to be migrated

If you intend to deploy your apps onto the WLS domain generated by the Marketplace stack, it is critical to provision a WebLogic domain matching the on-premises domain type.

The marketplace WLS stack provisions a non-JRF domain if the Database selection field is set to 'No Database'.

If any database type is selected, a JRF domain is provisioned.

If the domain you migrate from is not a JRF domain, but includes one or more databases as data sources, the Database field should be set to **'No Database'** at the time of provisioning the WebLogic installation. Data sources will be deployed in the WLS domain at a later step.

If a JRF domain is required (Fusion Middleware applications such as ADF, SOA, etc. or Java Cloud Service domains), then a database needs to be provided for the JRF components to use as storage. This is typically a different database than the one used as the data source for the applications running in the domain.

If the applications deployed on the WebLogic domain are built on the Fusion Middleware framework, then the domain type is JRF, while standard Java EE applications are typically deployed on a non-JRF domain.

A look at the resources in the source domain may further help figure if the domain to migrate is JRF or not.

A JRF domain will contain the following data sources:

<input type="checkbox"/>	LocalSvcTblDataSource	Generic	jdbc/LocalSvcTblDataSource
<input type="checkbox"/>	mds-owsm	Generic	jdbc/mds/owsm
<input type="checkbox"/>	opss-audit-DBDS	Generic	jdbc/AuditAppendDataSource
<input type="checkbox"/>	opss-audit-viewDS	Generic	jdbc/AuditViewDataSource
<input type="checkbox"/>	opss-data-source	Generic	jdbc/OpssDataSource
<input type="checkbox"/>	WLSSchemaDataSource	Generic	jdbc/WLSSchemaDataSource

Figure 1: JRF domain data sources under domain -> services -> data sources

7.2 Assess the number of Managed Servers and nodes to provision

On-premises installations of WebLogic Server may provision several Managed Servers per Machine (node). In a cloud environment where cost is linear with CPU count, deploying a single managed server per VM sized for the expected load, ensures better redundancy and increased availability. The WLS stack deployed through the Marketplace provisions one Managed Server per node. The number of nodes selected during provisioning should match the number of Managed Servers on-premises, and the Shape selected should be sized for the workload expected of a single Managed Server.

Note:

If you intend on updating your existing deployment tooling that creates the domain and your on-premises deployment assumes multiple Managed Servers per Machine, you may choose to provision only as many nodes as Machines in your deployment. You need to choose a Shape that can support the expected load of the multiple Managed Servers in order to reproduce the same topology at the expense of the extra resilience and increased availability.

If you do so, extra load balancer backends will need to be added manually to the existing load balancer once the stack is deployed in order to direct traffic to the additional ports used by the additional managed servers.



7.3 Assess what data sources need to be migrated.

The on-premises WebLogic domain likely contains data sources that are used by the applications deployed on the domain. Not accounting for the JRF specific data sources listed in figure 1 above (if the domain is a JRF domain), identify the data sources that need to be migrated. These are the databases that will need to be provisioned and migrated in further steps.

7.4 Assess other WebLogic Server services

There are several other services that may be used in your WebLogic deployment. Assess which need to be migrated. If any of the services has external dependencies deployed on-premises, you will need to configure access to those from OCI or plan a migration of the services to OCI outside of the WebLogic migration.

8 SETTING UP THE ENVIRONMENT ON ORACLE CLOUD INFRASTRUCTURE.

Some basic elements need to be provisioned before deploying the WLS marketplace image if a database is to be connected:

- A compartment to host the WLS domain and associated data sources.
- When deploying a JRF domain, a Virtual Cloud Network (VCN), along with a private subnet for databases needs to be provisioned. The Marketplace stack can deploy the other subnets for the WLS VM instances and load balancers.
- For a non-JRF domain (where a database is not required at the time of provisioning) the Marketplace stack can create the VCN and all subnets. A database subnet for application databases can be added to the created VCN after provisioning the stack.
- Proper Security Lists and Routing Tables need to be defined on the database subnet to allow access from the other subnets.
- A key store to host a key that will be used to encrypt credentials is required to avoid leaking credentials through the terraform logs.

The steps to create the key store and encrypt the WebLogic admin password (and JRF database password if needed) are described in the WebLogic marketplace 'Getting Started' guide.

8.1 Create a Compartment

- In Governance & Administration -> Identity -> Compartments
- Click Create Compartment.
- Make sure to select the proper parent compartment where the new compartment should live.

8.2 Create the Virtual Cloud Network (VCN)

If the domain to provision is a non-JRF domain:

- Without an application database, the Marketplace stack can provision the VCN and subnets making this step is optional. Skip to Section 11.
- If one or more data sources will be deployed in the non-JRF domain, a database private subnet is needed. The database private subnet can be provisioned after the WLS on OCI stack has been deployed, as a subnet of the VCN provisioned by the stack. Follow Section 8.3, Section 9 and Section 10 after you complete Section 11
- To provision this database subnet prior to deploying the stack, the VCN needs to be provisioned manually, and used as the Existing VCN when provisioning the WLS stack. In this case, follow Section 8.2.1 through Section 11

If the domain to provision is a JRF domain:

- A database needs to be provisioned to deploy the WLS stack. Therefore, the VCN and database subnet need to be created to provision the database system, prior to deploying the Marketplace stack, and the



VCN needs to be selected as Existing VCN when provisioning the WLS stack. Follow Section 8.2.1 through Section 11

8.2.1 Create a VCN

- In the OCI web console, Go to Networking -> Create Virtual Cloud Network
- If you will be migrating databases and require connecting from on-premises to this VCN via VPN or Fast Connect, make sure the VCN CIDR range on OCI does not overlap with the on-premises VCN CIDR range.

8.3 Create a Database Subnet

8.3.1 Create a Security list for the database subnet

Create a Security list with an Ingress Rules to allow TCP 1521 from the VCN CIDR block (source CIDR)

- In the VCN created at the previous step, select **Security Lists**
- Click **Create Security List**
- Name the Security List to be recognizable as the database subnet security list, such as *domain-db-security-list*
- Under **Allow Additional Ingress Rule**, Click **Additional Ingress Rule**
- Set the **Source CIDR** to be the VCN CIDR block
- Set the **Destination Port** to 1521

The screenshot shows the 'Create Security List' dialog in the OCI console. The 'NAME' field contains 'jrf-db-security-list'. The 'CREATE IN COMPARTMENT' dropdown is set to 'WLSJRF'. The 'Allow Rules for Ingress' section is expanded, showing 'Ingress Rule 1' with the following configuration: 'Allows TCP traffic 1521', 'STATELESS' checkbox is unchecked, 'SOURCE TYPE' is 'CIDR', 'SOURCE CIDR' is '10.0.0.0/16', 'IP PROTOCOL' is 'TCP', 'SOURCE PORT RANGE' is 'All', and 'DESTINATION PORT RANGE' is '1521'.

Figure 2 Create a Security List

- Click **Create Security List**

8.3.2 Create the private database subnet

- In the VCN details page, go to **Subnets**
- Click **Create Subnet**
- Name the subnet some recognizable name, such as *domain-db-subnet*
- Select **Regional Subnet**
- Set the **CIDR block** to be a subset of the VCN CIDR block (for example 10.0.5.0/24 if the VCNCIDR block is 10.0.0.0/16)



- Select Default route table
- Select Default DHCP options
- Under **Security Lists**, select the *domain-db-security-list* created earlier
- Click **Create Subnet**

8.4 Create an Internet Gateway

- In the VCN details page go to **Internet Gateways**
- Click **Create an Internet Gateway**
- Name the Internet Gateway
- Click **Create Gateway**
- Then go to **Route Tables**
- Click the **Default Route Table for the VCN**
- Click **Add Route Rules**
- Select **Internet Gateway**
- Set **Destination CIDR block** to be 0.0.0.0/0 (all sources)
- Select the Internet Gateway created at the previous step as the **Target Internet Gateway**
- Click **Add Route Rules**
This is needed for the terraform deployment stack to be able to check deployment status.

8.5 Create a key store

Key store creation is covered as part of the WLS marketplace stack Quick Start Guide in Section 11 and can be performed at that point.

9 PROVISION THE DATABASE(S) ON OCI

One Database System needs to be provisioned as the JRF datastore if the domain is a JRF domain. Application datastores can be provisioned as separate Pluggable Databases (PDBs) within the Container Database (CDB) of a single Database System, or one can create separate Database Systems for Application datastores and for JRF domain datastore.

For each data source used as resources for the applications to be deployed on the domain, a database (PDB) needs to be provisioned on OCI, whether in a single Database System or separate ones.

For DB on VMs, both Oracle Grid Infrastructure (which supports multi-node DBs) or Logical Volume Manager are supported.

Select a shape suitable for the load required of the database engine.

Select the VCN and database subnet provisioned in previous steps.

Note:

In order to use database migration tools such as Zero Downtime Migration (ZDM), the target database SYS password must be the same as the SYS password of the source database to be migrated.

10 MIGRATE ON-PREMISES DATABASE(S)

Migrate databases used as data sources for the application. Migration strategy highly depends on downtime requirements: in order to achieve zero-downtime, the target database needs to be provisioned and schemas need to be created so that the domain and apps can be deployed successfully, however data source sync and final switch over may only be performed as a final step.

You can use the [ZDM \(Zero Downtime Migration\) tool](#) to migrate:

In order to use ZDM, you will need:



- An Oracle Linux 7 or higher instance to install and run ZDM, which is able to reach both the source and target databases, as well as the Object Storage service endpoint.
- A bucket in the OCI compartment to use as the target for DB backup. (Create a bucket, and note the bucket name)
- The full User ID associated with the bucket. Check your Profile for the value to use. It may be in the form of <email> or <IDCSserviceName>/<email> if the IDCS service is used to manage you OCI account.
- An Auth Token for the bucket access:
Go to **Profile** (top right) -> **User Settings** -> **Auth Tokens**
and generate a token if you do not already have one.
- The **Object Storage Namespace** (under **Administration** -> **Tenancy Details** -> **Object Storage Settings**) which must be appended at the end of the storage endpoint url in the ZDM configuration file. The **Object Storage Namespace** may be your tenancy name, or some randomly generated name depending on how the tenancy was created.
- Make sure source and target databases use the same SYS password.

Note:

Only Oracle Grid Infrastructure-based databases on VMs are supported by ZDM at this time.

RMAN can also be used to migrate if downtime is acceptable.

Use RMAN backup and restore to backup the source database to OCI Object Storage and restore to the destination database on OCI.

11 DEPLOYING THE WEBLOGIC SERVER INFRASTRUCTURE WITH THE OCI MARKETPLACE IMAGE

To deploy WebLogic Server infrastructure, follow the guide below and note:

- The OCI tenancy compartment where the deployment is planned must have enough availability of the requested VM shape in the targeted Availability Domain.
Check for shape availability under **Governance** -> **Limits, Quotas and Usage** in the target Availability Domain and Compartment.
- The WLS on OCI deployment supports only the following shapes:
VM.Standard2.x, VM.Standard.E2.x, BM.Standard2.x, BM.Standard.E2.x
- The WebLogic admin password supplied must comply with the WLS password requirements (at least 8 characters, with at least one number or symbol) before encoding and encryption (see step 4 in the guide below for the encryption process)
- If a JRF domain is desired, select the database type under the Database section, and provide the database information to use as JRF storage target.
- If a non-JRF domain is desired, make sure to choose **No Database**, even if the applications to be deployed will be connecting to one or more databases: these data sources will be configured at a later step in the WLS domain.
- Oracle WebLogic Server 12c must be specified as the version for a JRF-enabled domain if you intend to use an Oracle Autonomous Transaction Processing database.

The [Get Started with Oracle WebLogic Server for Oracle Cloud Infrastructure](#) tutorial describes the 5 steps:

- Create a Dynamic Group to include all WLS instances
- Create a Policy for the Dynamic Group to provide access to the KMS service to the WLS instances
- Create a Vault and Key to encrypt the passwords when deploying the stack
- Deploy the stack



Important note:

Once the stack is deployed, the Resource Manager lets users change input variables for the Terraform, however changing variables and running Terraform Action 'Apply' **will destroy the entire environment and re-create a new one**, which is not typical Terraform behavior. See Section 13 about Scaling for more details.

12 DATABASE CONNECTION STRING FOR DB VMS

For all methods of migration, you will need to obtain the database connection string to use.

To get the DB connection string, go to **Database -> Database System**

Under the **Databases** list

- Click the database name to get to the database details page
- Then click **DB Connection**
- Choose the **Easy Connect** connection string, showing the `cdb_root`
It is of the form: `<hostname>:<port>/<cdb_root_name>.<subdomain>`

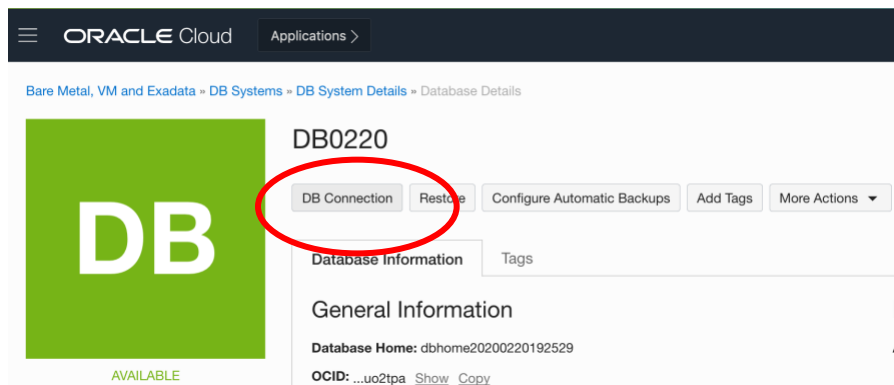


Figure 3 Database Easy Connect

Note that this provides the JDBC connection string to the Container Database (CDB), while you may need to point to the Pluggable Database (PDB) within the CDB when updating connection strings.

Note: If migrating to an Autonomous Transaction Processing (ATP) Database, a few more steps are necessary in order to connect WebLogic to ATP.

See the following blog post:

<https://blogs.oracle.com/weblogicserver/atp-database-use-with-weblogic-server-v2>

which includes WLST scripts, or this post

<https://weblogic.cafe/posts/atp-datasource/> to setup the DB connection manually.

13 ACCESSING THE WEBLOGIC ADMIN CONSOLE

13.1 WebLogic Server deployed in a public subnet:

- Once the stack has been deployed without errors, lookup the admin server public IP under **Solutions and Platform -> Resources Manager -> Stacks**
- Make sure to select the Compartment where the stack was deployed and click the Stack.
- The most recent job is the current state.



- Click the job and then the **Outputs** menu on the left. It will list the **WebLogic Server Administration Console URL**
- Copy/Paste this URL in your browser, and login to the WebLogic Admin console using the admin username and password you entered during provisioning.

13.2 WebLogic Server deployed in a private subnet

When deployed in a private subnet, the WLS admin console is not reachable directly. However, a bastion host with a public IP is created to access the WLS admin server through SSH tunneling.

- Once the stack has been deployed without errors, lookup the admin server public IP under **Solutions and Platform -> Resources Manager -> Stacks**
- Make sure to select the Compartment where the stack was deployed and click the Stack.
- The most recent job is the current state.
- Click the job and then the **Outputs** menu on the left. It will list the **Bastion Host IP address**.
- Go to **Compute -> Instances** and find the server with the suffix **-wls-0**, which hosts the WLS admin server
- Click the compute instance to get to the details page
- Get the **Private IP** of the compute instance.

Create an SSH tunnel to reach the WLS admin server with the following command:

```
export BASTION_IP=<BASTION_HOST_IP>
export WLSADMIN_HOST=<ADMIN_SERVER_PRIVATE_IP>
# SSH tunnel
ssh -M -S ssh-socket -fnNT -L 7001:${WLSADMIN_HOST}:7001 opc@${BASTION_IP} cat -
# you can check the connection was successful with
ssh -S ssh-socket -O check opc@${BASTION_IP}
```

The admin server will be reachable at <http://localhost:7001/console>

To terminate the tunnel when done, use:

```
ssh -S ssh-socket -O exit opc@${BASTION_IP}
```

14 MIGRATING A DOMAIN

Once the WebLogic Server stack is provisioned in the Oracle Cloud Infrastructure, multiple options are available to migrate a domain.

You can either

- Migrate resources and application from the on-premises domain to the new domain (see section 14.1)

Or

- Update existing deployment scripts (WLST or other) to re-create a domain and re-deploy resources and applications (see section 14.2)

Select the most appropriate method for your use case



14.1 Migrate resources and applications to the domain provisioned on OCI

This method consists in migrating on-premises resources and applications to the new domain created on OCI. The domain created will have the topology defined during creation, with one managed server per machine.

- This can be achieved with the following methods:
 - Re-deploy resources and applications manually through the WebLogic Admin Console. This method is best suited when moving with few applications. Follow Steps 14.1.1 and 14.1.2
 - Re-deploy resources through the WebLogic Admin Console, and re-deploy applications from the JDeveloper IDE. This method is best suited when moving few applications, for developers using JDeveloper. Follow steps 14.1.1 and 14.1.3
 - Re-deploy resources and applications using WebLogic Deploy Tooling `discoverDomain` and `updateDomain` commands. This method is best suited when moving several applications and resources. It consists in more steps but moves the whole domain at once, as opposed to one application at a time. Follow step 14.1.4

To deploy either manually through the WLS admin console or via JDeveloper, resources first need to be re-created manually.

14.1.1 Recreate the source domain resources

Use this step if you are deploying the resources and applications manually through the WLS admin console or using JDeveloper.

- Recreate the data sources under **Service** -> **Data Sources**
- Click **New** and follow the prompts to create a new Data Source.
- Make sure the **JDNI Name** matches the JDNI name of the data source in the source domain in the format **jdbc.<ConnectionNameDS>**, and provide the *hostname*, *port* and *database name*, based on your PDB name and the *subdomain* name obtained from the database connection string.
- The database name should be of the form: **<your_pdb_name>.<subdomain>** where **<your_pdb_name>** is the name of the PDB for the database you migrated over to OCI.
- The *hostname* is the hostname from the DB Connection string.
- The *port* is the port from the DB connection string, which defaults to 1521
- Use the default **Driver Class Name**: `oracle.jdbc.OracleDriver`
- In properties, enter the required information:
 - `user=<DB User>`
 - `Password=<DB password for the user>`
- Click **Next**
- Click **Test connection**
- Click **Next**
- For Targets, select all servers or the cluster where applications that need the resource will be deployed.

If you are deploying with JDeveloper, skip to section 14.1.3

14.1.2 Deploy the application manually

Use this step if you are deploying the resources and applications manually through the WLS admin console.



In the WebLogic Server admin console

- Click **Lock and Edit** to allow editing
- Go to **Deployments**
- Click **Install** -> **Upload your file** -> **Browse** (browse to local file to upload) then **Upload**
- Click **Next** -> Select **Install as Application**
- Click **Next**
- For Targets, select all managed server except admin servers or the cluster with all servers.
- Check **Copy this application onto every target for me**
- Click **Finish**
- Click **Activate Changes**
- Then go to **Deployments** -> **Control** tab
- Select the applications and click **Start** -> **Start Servicing all requests**
- Click **Yes**
- You should be able to reach your applications at the IP of the WLS load balancer, by their respective path
- You can find the IP of the load balancer in the OCI web console:
- Under **Solutions and Platforms** -> **Resources Manager** -> **Stack**
- Select the compartment where the Marketplace image was deployed
- Click the stack, then go to **Jobs** and select the latest job that deployed the stack (there should only be one job unless you used other terraform Actions).
- Under **Outputs**, find the **WebLogic Server Load Balancer IP**
- Your app should be running at `<Load_Balancer_IP> /<application_root_path>`
- If you are not sure what the `application_root_path` is for the app, go to
- **Deployments**, click the **Control** tab to see only applications
- Click the application, then go to **Testing** and expand the tree to see the URL path where the application is running (although it will show the local FQDN for the hostname, which needs to be substituted for the load balancer IP)

When you are through these steps, you are done with migration. See Section 16. for information on scaling the WebLogic deployment.

14.1.3 Deploy Apps from JDeveloper (or Eclipse)

Use this step if you are deploying the resources and applications manually through JDeveloper. You must have followed previous step 13.1.1 to re-create resources manually in order for the following to work.

For JDeveloper or Eclipse to connect with the WLS admin server, the admin server `ListenAddress` needs to be set to the private IP of the hosting VM.

14.1.3.1 Update the ListenAddress of the admin server

- Find the private IP of the admin server host:
- In the OCI web console, go to **Compute** -> **Instances**
- Make sure you are in the compartment where you deployed the stack from the Marketplace.
- Click the compute instance with suffix `-wls-0`
- Under **Primary VNIC Information**, take note of the **Private IP Address**.
- In the WebLogic admin console, go to **Environment**->**Server** and click the admin server name to get to the admin server configuration
- Click the **Lock and Edit** button
- Update the **ListenAddress** with the private IP of the VM recorded at the previous step.
- Click **Save** at the bottom

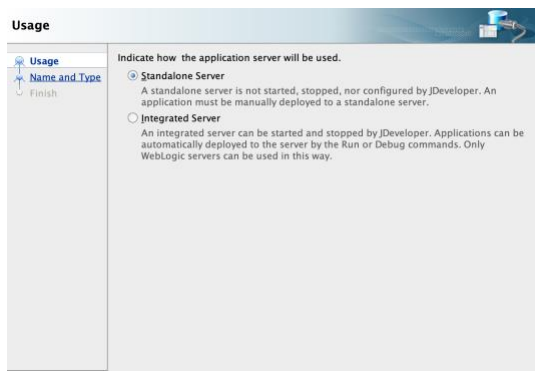


- Then click **Activate Changes**
The changes were activated but the prompt suggests the server needs to be restarted, however it cannot be restarted from the console and need to be restarted manually.
- Restart the WLS admin server manually

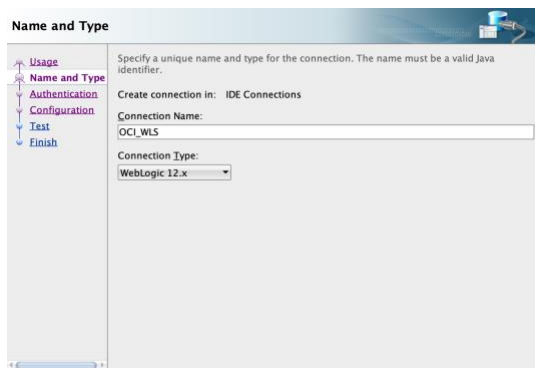
```
# ssh to the admining server
ADMIN_SERVER_PUBLIC_IP=
ssh opc@<ADMIN_SERVER_PUBLIC_IP>
# switch to oracle user
sudo su -oracle
# stop the weblogic admin server
$DOMAIN_HOME/bin/stopWebLogic.sh
# restart the weblogic admin server and send to background
$DOMAIN_HOME/bin/startWebLogic.sh &!
# &! Sends the command to the background and disowns it,
# so it survives terminal exit.
```

14.1.3.2 Create a new Application Server in JDeveloper

- Connect to the WebLogic server remotely to deploy applications by creating a new Application Server in your local JDeveloper:
- **Window -> Application Servers -> right click -> New Application Server**
- Follow prompt for a **StandAlone** Server, and with the admin server login information.
- **Choose StandAlone server**

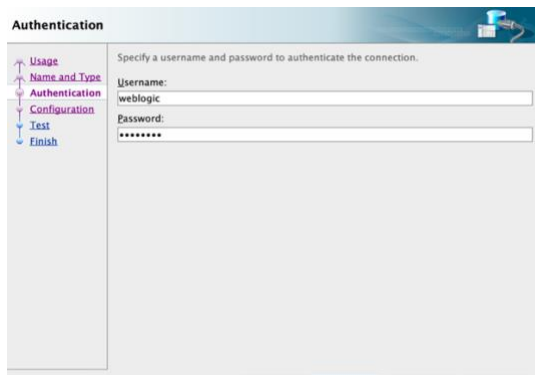


- **Name the server connection**

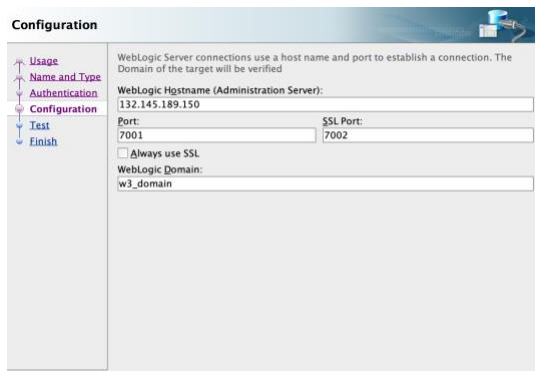


- **Provide credentials to the admin server**

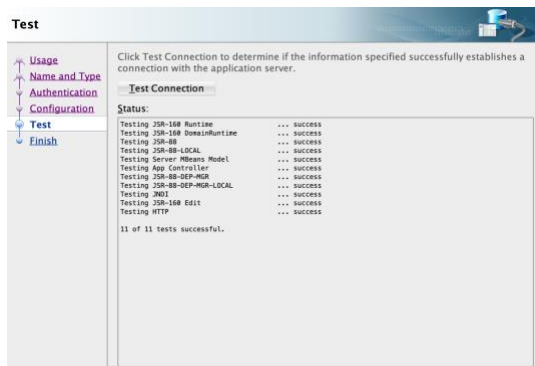




- Provide the Public IP of the Weblogic admin server, port and name of the domain that is deployed (<prefix>_domain)



- Click **Test Connection** to verify it can connect.

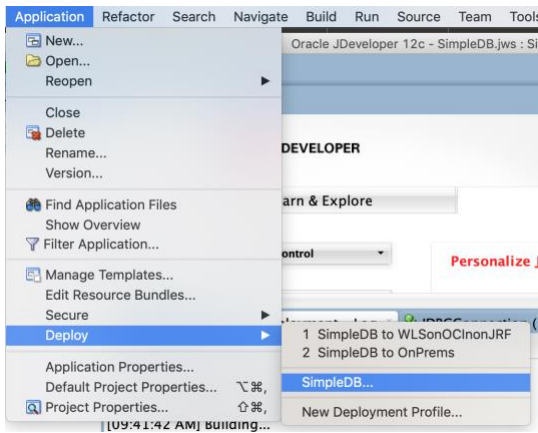


Once the Application Server has been defined in JDeveloper, deploy the application using the deployment config.

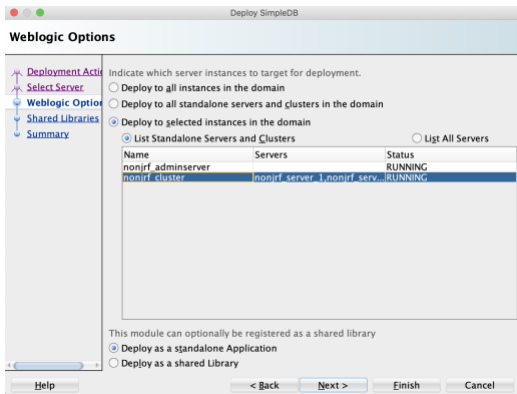
14.1.3.3 Deploy the application with JDeveloper

- Go to Application -> Deploy. Select an existing deployment profile to create a new deployment.

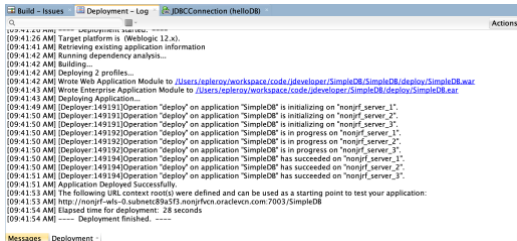




- Select **Application Server**, then select the newly created Application server and follow prompts to choose the target (choose the specific server or cluster)

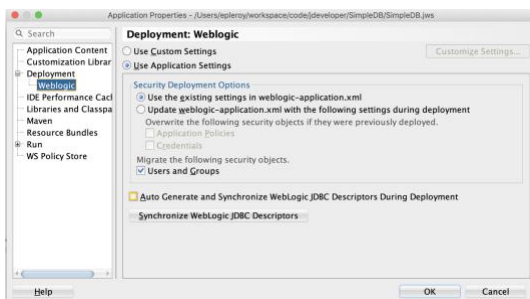


- Follow deployment logs in the Deployment Log window to verify deployment worked as expected.



- Most deployment failures are due to missing libraries.
- Also make sure that in the Deployment profile (under **Application -> Properties -> Deployment Profile -> Weblogic**), the security settings are set to **Use Existing settings in weblogic-application.xml**, and that the option to **Auto-Generate and Synchronize WebLogic JDBC descriptors** is **unchecked**.

See below:



14.1.4 Migrate a domain to the Marketplace image domain with WebLogic Deploy Tooling

This method consists of discovering the source domain with WebLogic Deploy Tooling (WLDT), updating the model file to map to the topology of the domain newly provisioned through the marketplace image, and updating the marketplace target domain with the resources and applications extracted from the source domain.

For this method to be effective, the proper domain type (JRF or non-JRF) must have been provisioned at the previous step.

This method takes advantage of the provisioned domain (including the defined topology) and re-deploys resources and applications on that domain. This involves updating or adding data sources to the domain resources with proper hostnames, ports and credentials, as well as re-targeting the applications and libraries to the new managed server or cluster names.

This method allows easy integration in your development and deployment flow if the flow assumes a pre-provisioned domain as is provided by this deployment.

14.1.4.1 Before you start

- The domain name created on the WLS on OCI image is the prefix you supplied during provisioning with *_domain*, for example *test_domain*
- Managed server names follow the pattern: *<prefix>_server_<index>* for example *test_server_1*, *test_server_2*,... *test_server_n* where *n* is the number of nodes provisioned.
- The Admin server is named: *<prefix>_adminserver*
- If a WebLogic Enterprise Edition was provisioned, a cluster named *<prefix>_cluster* is provisioned that includes all the managed servers provisioned.

Using WebLogic Deploy Tooling (WLDT), update the source domain model file to re-target the various applications and resources to the new domain's target servers, and update database connections to match the newly provisioned and migrated OCI databases.

14.1.4.2 Install WLDT on the source domain

As the proper WebLogic user (typically *oracle*) on the source domain host, with properly configured `JAVA_HOME`

```
# install weblogic-deploy-tooling
# Check for the latest version at:
# https://github.com/oracle/weblogic-deploy-tooling/releases/
WLSDT_VERSION=1.7.0
echo $JAVA_HOME
# download and unzip the package
wget -qO- https://github.com/oracle/weblogic-deploy-tooling/releases/download/weblogic-deploy-tooling- $\{WLSDT\_VERSION\}$ /weblogic-deploy.zip | $JAVA_HOME/bin/jar xv
# make the scripts executable
chmod +x weblogic-deploy/bin/*.sh
```



14.1.4.3 Run the WebLogic Deploy Tooling discoverDomain command to discover the source domain

```
# path to the parent of the wlserver folder (for example /u01/app/oracle/middleware/)
MW_HOME=
# path to the domain folder (i.e. <domain root>/<domain_name>)
DOMAIN_HOME=
# for a JRF domain
DOMAIN_TYPE=JRF
# for a non-JRF domain
DOMAIN_TYPE=WLS

weblogic-deploy/bin/discoverDomain.sh \
  -oracle_home $MW_HOME \
  -domain_home $DOMAIN_HOME \
  -archive_file source.zip \
  -model_file source.yaml \
  -variable_file source.properties \
  -domain_type $DOMAIN_TYPE
```

Note:

The **-domain_type** selection is critical during discovery. It defaults to WLS, so it can be omitted for a WLS domain.

However when **-domain_type** is **JRF**, the discovery command filters out all the JRF components, which are re-provisioned by the marketplace image in the case of a JRF domain.

The name 'source' for the .yaml,.zip and .properties files is arbitrary, feel free to change, however this guide will use this name in further steps.

14.1.4.4 Move the source domain files to the target

Move the files to the admin server on the target environment.

```
SOURCE_IP=<ip of the source WLS server>
ADMIN_SERVER_IP=<IP of the admin server on OCI>
scp <source_user>@${SOURCE_IP}:~/source.* opc@${ADMIN_SERVER_IP}:~/

# change ownership of the files from opc to oracle
# and move the file to the oracle user home
ssh opc@${ADMIN_SERVER_IP}
sudo chown oracle:oracle source.*
sudo mv source.* /home/oracle/
# switch to the oracle user
sudo su - oracle
```



14.1.4.5 Install Weblogic Deploy Tooling on the WLS admin server on OCI

```
# As the oracle user, install weblogic-deploy-tooling
WLSDT_VERSION=1.7.0
wget -qO- https://github.com/oracle/weblogic-deploy-tooling/releases/download/weblogic-deploy-tooling-${WLSDT_VERSION}/weblogic-deploy.zip | $JAVA_HOME/bin/jar xv

# make the scripts executable
chmod +x weblogic-deploy/bin/*.sh
```

14.1.4.6 Edit the source domain model file to match the target domain topology

The *source.yaml* file is the model file, which describes the domain, its topology (servers, clusters etc...), resources and application deployments.

The *source.zip* file contains the application files.

The *source.properties* file contains the placeholders for credentials that cannot be simply ported over.

Since the domain is already provisioned, with a topology defined with managed server names based on the domain name prefix, the migration consists in updating the target domain with only the resources definitions and applications, adjusting resources to target the new resources, and updating targets to match the new environment's server names.

Note:

When editing the YAML file, make sure to use SPACES and not TABS or WLDT parsing will fail.

YAML is a language where indentation represents nesting, so when removing a top-level key, you should remove everything under the key that has higher indentation up to the next key at the same level.

14.1.4.6.1 Edit the yaml file to keep only the sections resources and appDeployments

- Remove the entire **domainInfo** section
- Remove the entire **topology** section
- You should be left with a **resources** top-level key and an **appDeployments** top-level key
- Under **resources** -> **JDBCSystemResource** and then for each application under **appDeployment**:
- If you launched the **WebLogic Server Enterprise Edition**: edit the **Target** key to match the *cluster* name on OCI:
Cluster name is *<domain_prefix>_cluster*, for example *ee_cluster*
- If you launched the **WebLogic Server Standard Edition**: edit the **Target** key to match the *list of managed server names* on OCI that the application should target:
Each managed server name follows the pattern: *<domain_prefix>_server_<id>*
for example *'se_server_0,se_server_1,se_server_3'*
(make sure the list is separated by commas and surrounded by single or double quotes)

14.1.4.6.2 Edit the URLs of the data sources under each JDBCDriverParams key

- The **URL** should follow the pattern:
- *'jdbc:oracle:thin:@//<hostname>:<port>/<your_pdb_name>.<subdomain_name>'*
Where *hostname*, *port* and *subdomain_name* are from the database connection string, and *your_pdb_name* is the name of the PDB hosting your database. Refer to section 12.1 for where to get the connection string.

Note:

If the database system was created with more than one node, the connection string will be of the form:
<hostname>-scan:<port>/<cdb_root_name>.<db_subnet_domain_name>



While if created with one node, it will be:

```
<hostname>:<port>/<cdb_root_name>.<db_subnet_domain_name>
```

Either one can be used if the database consists only of one node, however if there is more than one node, the -scan name must be used to insure proper discovery. In order to scale seamlessly in the future, it is recommended to use the -scan name.

Make sure to use the proper PDB name for your database hosted under the cdb_root_name.

14.1.4.6.3 Add the StagingMode key if missing

For each application under appDeployments, make sure the StagingMode key is set to 'stage', or add it if it is not present. StagingMode: 'stage' will automatically deploy the files on the other targeted managed servers, or managed servers of the cluster. Without StagingMode: 'stage', you will need to manually deploy the file to each server

Example source.yaml file

```
resources:
  JDBCSystemResource:
    JDBCConnection:
      Target: DefaultServer # ← target server/cluster name to update
      JdbcResource:
        DatasourceType: GENERIC
        JDBCConnectionPoolParams:
          TestTableName: SQL ISVALID
        JDBCDataSourceParams:
          GlobalTransactionsProtocol: OnePhaseCommit
          JNDIName: jdbc.JDBCConnectionDS
        JDBCDriverParams:
          URL: 'jdbc:oracle:thin:@//localhost:1521/ORCLCDB' # ← JDBC URL to
update
          PasswordEncrypted: '@@PROP:JDBC.JDBCConnection.PasswordEncrypted@@'
          DriverName: oracle.jdbc.OracleDriver
          Properties:
            user:
              Value: 'C##DATALOAD'
appDeployments:
  Application:
    webApp:
      SourcePath: wlsdeploy/applications/webApp.war
      ModuleType: war
      Target: DefaultServer # ← target server/cluster name to update
      StagingMode: 'stage' # ← make sure this key is set for each app
    helloDB:
      SourcePath: wlsdeploy/applications/helloDB.ear
      ModuleType: ear
      Target: DefaultServer# ← target server/cluster name to update
      StagingMode: 'stage' # ← make sure this key is set for each app
```

Note:



If application files are under the ORACLE_HOME folder structure, they will not be picked up by WebLogic Deploy Tooling, and will need to be moved manually. Alternatively, they can be packaged in a zip file with the folder structure like **wlsdeploy/applications/<app.ear>** and the SourcePath can be changed to the same path so WebLogic Deploy Tooling will deploy in the default folder on the target.

14.1.4.6.4 Edit the variable file

Edit the *source.properties* file and update the JDBC connection password(s) to use with the database(s) that were migrated.

If you first moved the files locally to edit them, upload the updated files on the WLS admin server on OCI, make them owned by the oracle user, and move them to the oracle user home

If you moved the files to the target directly as suggested, you can skip this step.

```
ADMIN_IP=
# copy the files over
scp source.* opc@${ADMIN_IP}:~/
# make them owned by the oracle user and move them to the oracle user home
ssh opc@${ADMIN_IP} 'sudo chown oracle:oracle source.* && sudo mv source.* /home/oracle/'

# SSH to the admin server on OCI
ssh opc@${ADMIN_IP}
# switch to the oracle user
sudo su - oracle
```

14.1.4.7 Run the wlsdeploy-deploy-tooling updateDomain script to update the domain

```
# make sure the needed variables are set:
echo $DOMAIN_HOME
echo $MW_HOME

# run WebLogic Deploy Tooling updateDomain script in online mode
wlsdeploy-deploy/bin/updateDomain.sh \
-oracle_home $MW_HOME \
-domain_home $DOMAIN_HOME \
-model_file source.yaml \
-variable_file source.properties \
-archive_file source.zip \
-admin_url t3://$(hostname -i):7001
```

Despite the prompt at the end of this command, the server should not have to be restarted.

Notes:

- The **-domain_type** is not used here, as it is an update only. Using the **-domain_type JRF** option even for a JRF domain causes update issues with WLDT for online updates.
- The **-admin_url** requires the **t3** protocol with the admin server local private IP as extracted by the `hostname -i` command. `Http` or `https` through the public IP will not work.



- Providing the **-admin_url** in this command ensure the domain is updated 'online'. Not all domains can be updated online, but unless topology is modified (which is not the case here), it should work for most domains.
- Doing the update 'online' allows for deploying the applications and resources without the need to restart the servers.
If you need to do the update offline, remove the **-admin_url** option, and then stop and restart WebLogic server using the stopWebLogic.sh and then startWebLogic.sh scripts.

Use **startWeblogic.sh &!** to send the command to the background and disown it, so it survives terminal exit.
- It will be necessary to stop and restart the server **twice** for the StagingMode: 'stage' flag to take effect if the update was done offline. The first restart will update the configuration, and then StagingMode deployment will require a restart to take effect.

14.2 Update existing tools to create and deploy a custom domain in the provisioned infrastructure.

This method consists in extracting the relevant domain information from the provisioned infrastructure (machine hostnames and ports), to update existing deployment tooling used to create domains and deploy applications and target the newly provisioned infrastructure.

It is more adapted to a development / deployment flow where the tooling includes domain creation and update as part of the flow.

14.2.1 Extract the domain metadata

Once the stack is provisioned, ssh to the admin server VM and extract metadata

```
# SSH to the admin server VM
ssh opc@${ADMIN_SERVER_IP}
# extract metadata information using the meta data service endpoint
curl http://169.254.169.254/opc/v1/instance/metadata/

# specific meta data can be extracted with the key name
# useful keys:
curl http://169.254.169.254/opc/v1/instance/metadata/wls_admin_server_name
curl http://169.254.169.254/opc/v1/instance/metadata/wls_admin_port
curl http://169.254.169.254/opc/v1/instance/metadata/wls_ms_ssl_port
curl http://169.254.169.254/opc/v1/instance/metadata/service_name
curl http://169.254.169.254/opc/v1/instance/metadata/wls_domain_name
curl http://169.254.169.254/opc/v1/instance/metadata/wls_admin_host
curl http://169.254.169.254/opc/v1/instance/metadata/wls_admin_ssl_port
curl http://169.254.169.254/opc/v1/instance/metadata/wls_machine_name
curl http://169.254.169.254/opc/v1/instance/metadata/total_vm_count
curl http://169.254.169.254/opc/v1/instance/metadata/wls_cluster_name
curl http://169.254.169.254/opc/v1/instance/metadata/wls_ms_server_name
curl http://169.254.169.254/opc/v1/instance/metadata/host_index
curl http://169.254.169.254/opc/v1/instance/metadata/wls_nm_port
```

Refer to document on the [meta data service](#) for more details.




```
# info about VNICS
curl http://169.254.169.254/opc/v1/vnics/
```

Metadata can also be extracted from the State of the terraform script. If you integrate the terraform deployment script as part of your deployment tooling, this method is likely the most effective.

Scripts and helpers are available in the folder `/opt/scripts` on each VM, which can be used to extract the metadata and create a new domain

15 SCALING MANAGED SERVERS

15.1 Scaling the number of WebLogic Managed Servers by scaling the number of nodes

Once the stack is deployed via the Marketplace, the terraform script used to deploy the stack is accessible through **Solutions and Platform** -> **Stacks** -> stack -> **Download**.

To scale the number of VMs deployed, go to **Variables**, then **Edit Variables**, where the **WebLogic Server Node Count** can be updated up or down.

Important note:

*Whether adjusting number of nodes on the local Terraform script or within the Resource Manager, **the entire deployment will be destroyed and recreated**, and applications and resources will need to be re-deployed in the domain. Because of this behavior, Public and private IP addresses will change.*

15.2 Scaling by adding Managed Servers on the provisioned nodes

By default, the stack deploys 1 WebLogic managed server per VM, plus the WebLogic admin server on the first one.

To add managed servers on the already provisioned VMs, and scale WLS managed servers without scaling the number of nodes, create the new managed servers in the WLS Admin Console, or through scripting, making sure to select ports that are available.

Then add backends to the load balancer, selecting the VMs and port combination that were added.

Go to **Networking** -> **Load Balancers** -> Click the load balancer `<prefix>_lb` -> **Backend Sets** -> Click the backend set `<prefix>_lb_backendset` -> **Backends** -> **Add backends**

Select the VMs where new managed servers were created, and set the port these managed servers use, then click Add.

To scale the VM capacity vertically to accommodate for more Managed Servers, you can also change the shape of a VM:

- Go to **Compute** -> **Instances** and select the VM to scale
- Click the VM to see the details page
- Click **Actions** -> **Change Shape**

This requires a reboot.

16 OTHER DOCUMENTATION

[Oracle WebLogic Server Documentation](#)

[Oracle WebLogic Server on Oracle Infrastructure Reference Architecture](#)

[Oracle WebLogic Server on OCI Quick Start Guide](#)



CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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

Migrating WebLogic Server
August, 2020

