Oracle WebLogic Server on Docker Containers

ORACLE WHITE PAPER | APRIL 2020

Table of Contents

Oracle WebLogic Server on Docker

Oracle WebLogic Server (WLS) is certified to run on Docker and CRI-O containers. On GitHub, we have Dockerfiles and supporting scripts to build images of Oracle WebLogic Server; these images are an extension of existing Oracle Linux images. You can use these WLS Docker images or create your own.

Docker is a platform that lets you build, package, ship, and run distributed applications. Using Docker, you package your applications and dependent libraries or files, into a Docker image. Docker images are portable artifacts that can be distributed across Linux environments. You can use images that have been distributed to instantiate containers where applications can run in isolation from other applications, running in other containers on the same host operating system.

Oracle WebLogic Server versions 12.1.3, 12.2.1.3.0, 12.2.1.4.0, and 14.1.1.0.0 are certified to run in Docker and CRI-O containers. You can use combinations of Oracle WebLogic Server, JDK, Linux, and Docker versions when building your Docker images. For details on the most current Oracle WebLogic Server supported configurations and support statement, see the Oracle Fusion Middleware Certification Pages.

The provided Dockerfiles and scripts let you create clustered and non-clustered WLS domain configurations. Each server running in the resulting domain configuration, runs in its Docker container, and can communicate with other servers. Other configurations and approaches are possible; this paper will guide you to create these configurations.

Oracle WebLogic Server Docker Images

The Dockerfiles and supporting scripts on GitHub build Oracle WebLogic Server Docker images. These images are extensions of existing Oracle Linux image 7.0 slim, with Server JRE 7, 8, or JDK 11, and the Oracle WebLogic Server 12c (12.1.3, 12.2.1.3, 12.2.1.4) and 14c (14.1.1.0) installations.

Three types of images are posted:

1. An Oracle WebLogic Server Generic image.
2. An Oracle WebLogic Server Developer image.
3. An Oracle WebLogic Server Slim image.

The WebLogic Server generic image is built with the “generic installer” and includes all Oracle WebLogic Server and Oracle Coherence software, including examples, and is supported for all development and production purposes. The WebLogic Server developer image is created with the “quick installer” and is intended for development purposes. It includes all Oracle WebLogic

Server and Oracle Coherence runtime software but excludes examples and localized WebLogic Server Administration Console help files. The WebLogic Server slim image is created with the “slim installer” and is for development and production usage of Docker or CRI-O images and containers in Kubernetes, when WLS Console monitoring and configuration is not required. It includes all Oracle WebLogic Server and Oracle Coherence server runtime software, but excludes examples, the WebLogic Server Administration Console, WebLogic clients, Maven plug-ins, and Java DB.

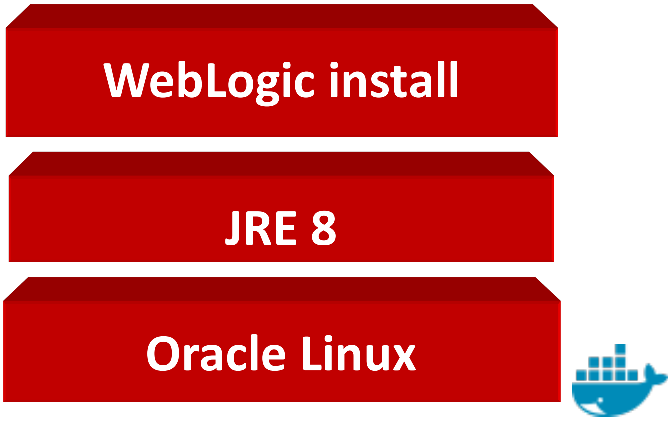

Fig 1. Oracle WebLogic Server Docker Image

Custom Built Oracle WebLogic Server Images

You can create your own Oracle WebLogic Server Docker images. As examples to help you to get started, we have posted Dockerfiles and scripts on GitHub Oracle WebLogic Server Dockerfiles.

What are the prerequisites to build custom WLS images?

1. An Oracle Linux base image.
2. Dockerfiles and scripts from GitHub.
3. The Oracle WebLogic Server Generic, Quick, or Slim installer.
4. The corresponding Server JRE or JDK.

The WebLogic Server images can be created and patched using the WebLogic Image Tool. For more information, see the Image Tool GitHub project.

Dockerfiles and Scripts on GitHub

Dockerfiles

The Dockerfile to create an Oracle WebLogic Server install image, does the following:

1. Extends the Server JRE image built with the Oracle Linux Slim base image.
2. Installs WLS using the Generic, Quick, or Slim installer, in silent mode.

After creating your Oracle WebLogic Server install images, you can extend them to have a base Oracle WebLogic Server domain configured. For more information, see the samples in the Oracle Docker GitHub project.

The Dockerfile to create an Oracle WebLogic Server domain image, does the following:

1. Extends the Oracle WebLogic Server install image.
2. Configures a WLS domain by calling Oracle WebLogic Server Scripting Tool (WLST) scripts or the WebLogic Deploy Tooling.

The domain has one Administration Server, a WebLogic cluster with multiple Managed Servers, and resources deployed to the cluster, such as data sources and JMS.

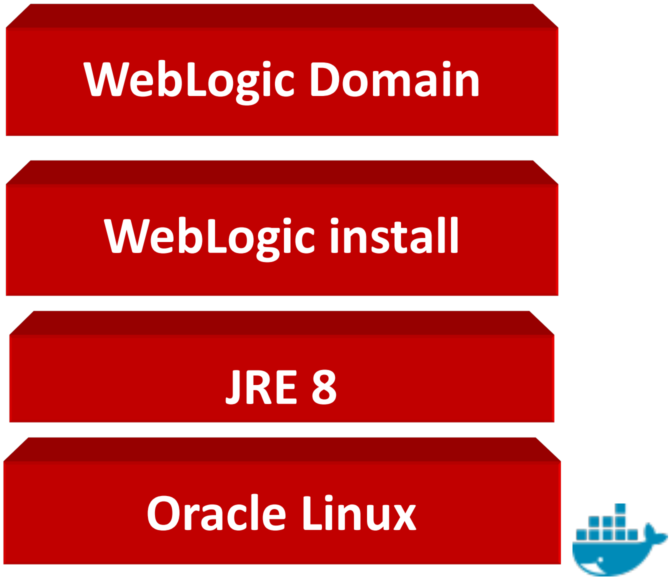

Fig 2. Oracle WebLogic Server Domain Image

Using the Oracle WebLogic Server domain image, you can create two types of containers:

- An Administration Server container with a single Oracle WebLogic Server Administration Server.
- A Managed Server container.

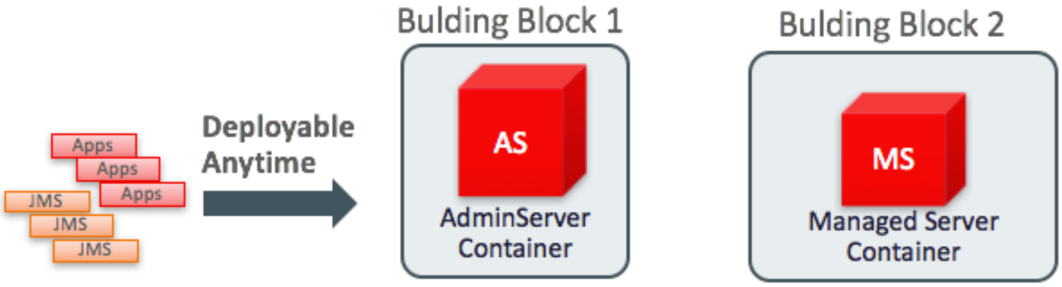

Fig 3. Types of Oracle WebLogic Server Containers

Scripts

The scripts help you create the Oracle WebLogic Server image and serve as examples of extending the image, by configuring an Oracle WebLogic Server domain.

Clustering Oracle WebLogic Server on Docker Containers

Docker containers let you create clustered and non-clustered WLS domain configurations. Each server in the domain, runs in its own Docker container and can communicate with other servers on the same host.

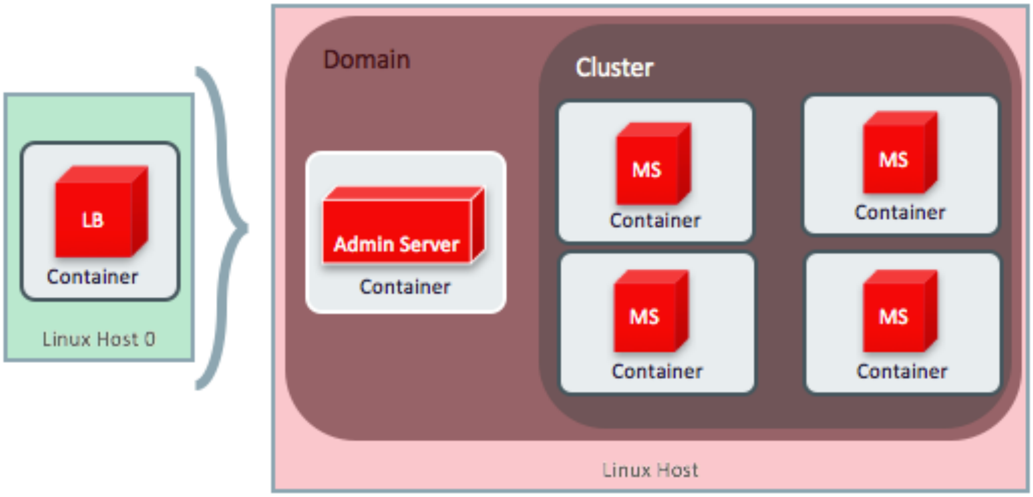

Fig 4. Clustering Oracle WebLogic Server on Docker Containers across Single Host

The advantages of this topology are:

- Good for traditional deployments.
- Easy to deploy containers from WLS domain images.
- Easy to scale clusters up and down.
- Good for developers.
- No need to install or configure anything on the host except for Docker binaries.

The containers can be all on a single physical or virtual server Linux host or on multiple physical or virtual server Linux hosts, as in figure 5.

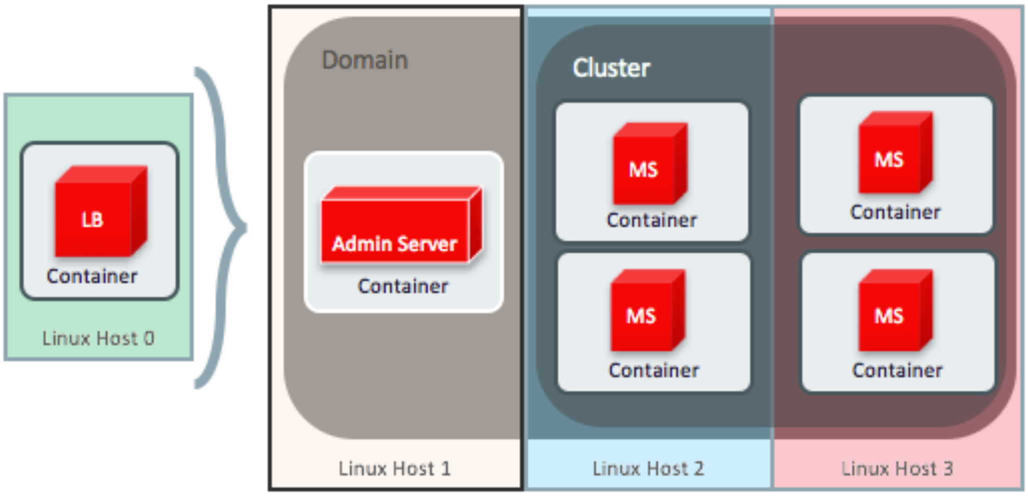

Fig 5. Containerized Oracle WebLogic Server Applications on a Multi Host

The advantages of this topology are:

- The "Docker-way" for containerized applications and services.
- Containers are easily reproducible.
- Each container is an instance of the same WLS domain.

A topology aligned with the "Docker-way" for containerized applications and services, consists of a container running an Administration Server only containing all the resources, shared libraries, and deployments. The Docker image includes all pre-defined domain resources, applications, and shared libraries deployed, with no Managed Servers or clusters configured, as in figure 6.

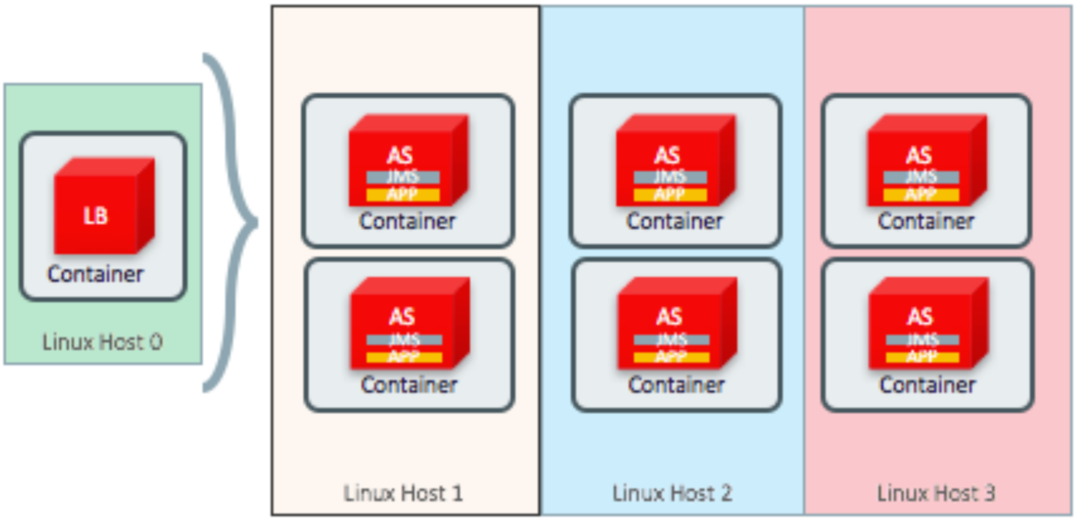

Fig 6. Single Server Domain with Applications and Resources on a Multi Host

How to Build and Run

On GitHub you will find the Dockerfiles and supporting scripts to build an Oracle WebLogic Server install image and to extend this image to create an Oracle WebLogic Server domain image. To build your Oracle WebLogic Server images and start your containers, first download the entire directory structure.

Dockerfiles

There are three Dockerfiles for Oracle WebLogic Server 12.2.1.4.0, located under the /OracleWebLogic/dockerfiles/12.2.1.4 subdirectory, one each to build: a WLS 'developer' image, a WLS 'generic' image, and a WLS ‘slim’ image.

Dockerfile.developer

Dockerfile.generic

Dockerfile.slim

Scripts

Under the /OracleWebLogic/dockerfiles /12.2.1.4/container-scripts subdirectory, there are supporting scripts which, together with the Dockerfiles mentioned above, are necessary to build an Oracle WebLogic Server image and configure a single server domain.

Script Name	Functionality
buildDockerImage.sh	Builds the WLS image using the WLS installation Dockerfile instructions.
container-scripts/createAndStartEmptyDomain.sh	Takes credentials from the properties file, creates the boot.properties file, invokes create-wls-domain.py to run WLST to create the domain configuration, and starts the Administration Server.
container-scripts/create-wls-domain.py	Configures a base domain with one Administration Server.
properties/domain.properties	Contains the Administration Server credentials.

Building an Oracle WebLogic Server Image

The WebLogic image extends the Oracle Server JRE 8 image. For access to the Dockerfiles and scripts mentioned in this paper, clone the Docker repository to your local machine:

```
$ git clone https://github.com/oracle/docker-images.git
```

Build the image by using the Dockerfile in ../../../OracleJava/java8 or pull a pre-built image from the Oracle Container Registry (OCR). First time users, follow these directions.
The commands to pull the image from OCR:

```
$ docker login container-registry.oracle.com

$ docker pull container-registry.oracle.com/java/serverjre:8

$docker tag container-registry.oracle.com/java/serverjre:8 oracle/serverjre:8
```

First, decide which installation type you want to use, either generic, quick, or slim installer, and then download the required Oracle WebLogic Server installer to the dockerfiles/12.2.1.4 folder. From the dockerfiles folder, run the buildDockerImage.sh script as root:

```
$ sudo sh buildDockerImage.sh -h
```

Usage: buildDockerImage.sh [-d|-g|-m]

Parameters:

-d: creates image based on 'developer' distribution

-g: creates image based on 'generic' distribution

-m: creates image based on 'slim' distribution

***Note**: The resulting image will NOT have a pre-configured domain. We provide separate Dockerfile and supporting scripts to extend the Oracle WebLogic Server install image and create a WLS domain image.*

Samples for Oracle WebLogic Server Domain Creation

To show you how to create a domain from a custom Dockerfile to extend the Oracle WebLogic Server install image, we provide a few samples for Oracle WebLogic Server. We support different topologies where the WebLogic Server domain home is stored either in the image or on a host volume. You can create the domain home using WLST or the WebLogic Deploy Tooling (WDT). For more information about WebLogic Deploy Tooling, see the WebLogic Deploy Tooling GitHub Project.

Sample Domain for Oracle WebLogic Server

The Dockerfile, in OracleWebLogic/samples/12213-domain, will create an image by extending oracle/weblogic:12.2.1.3-dev (from the Developer distribution). It will configure a WebLogic Server domain on a host volume. The WebLogic Server domain will have an Administration Server with a WebLogic cluster (configured or dynamic) containing one or more Managed Servers.

***NOTE**: This sample can be used to create and run a WebLogic Server 12.2.1.4 or 14.1.1.0 domain. Edit the Dockerfile and change the “FROM” clause to point to the image you built in the steps above, e.g. “FROM oracle/weblogic:12.2.1.4-dev”.*

Under the subdirectory, /OracleWebLogic/samples/12213-domain and /OracleWebLogic/samples /12213-domain/container-scripts, you will find supporting scripts which, together with the Dockerfiles, are necessary to build an Oracle WebLogic Server domain image, configure a domain, and run the Administration Server and Managed Servers in a cluster.

Script Name	Functionality
build.sh	Builds the WLS domain image called “12213-weblogic-domain-in-volume”.
run-admin-server.sh	Takes credentials from the properties file, creates the boot.properties file, invokes create-wls-domain.py to run WLST to create the domain configuration, and starts the Administration Server.

run-managed-server.sh	Configures a base domain with one Administration Server.

properties/domain.properties	Contains the following values, which lets you customize the domain.
	DOMAIN_NAME=myDomain
	ADMIN_PORT=7001
	ADMIN_NAME=myadmin
	ADMIN_HOST=AdminContainer
	MANAGED_SERVER_PORT=8001
	MANAGED_SERVER_NAME_BASE=MS
	CONFIGURED_MANAGED_SERVER_COUNT=2
	PRODUCTION_MODE_ENABLED=true
	CLUSTER_NAME=cluster1
	CLUSTER_TYPE=DYNAMIC
	DOMAIN_HOST_VOLUME=/Users/host/temp

properties/domain_security.properties	WebLogic Administration Server credentials
container-scripts/setEnv.sh	Sets the parameters above to run the servers.
container-scripts/createWLSDomain.sh	Takes credentials from the domain_security.properties file, creates the boot.properties file, and invokes create-wls-domain.py to run WLST to create the domain configuration.
container-scripts/create-wls-domain.py	Configures a domain with one Administration Server, a cluster (configured or dynamic), and Managed Servers in the cluster.
container-scripts/startAdminServer.sh	Starts an Administration Server.
container-scripts/startManagedServer.sh	Starts a specific Managed Server in the cluster.

Building a sample Docker Image of Oracle WebLogic Server Domain

To create an Oracle WebLogic Server image with a domain configured, follow these steps:

If you have not already done so, clone the Docker GitHub repository to your local machine. You will need the Dockerfiles and scripts to the sample files mentioned in this paper:

```
$ git clone https://github.com/oracle/docker-images.git
```

Make sure you have the oracle/weblogic:12.2.1.3-dev image built following the instructions above or pull a pre-built image from the Oracle Container Registry (OCR). First time users, follow these directions.

The commands to pull the image from OCR:

```
$ docker login container-registry.oracle.com

$ docker pull container-registry.oracle.com/middleware/weblogic:12.2.1.3-dev

$docker tag container-registry.oracle.com/middleware/weblogic:12.2.1.3-dev
oracle/weblogic:12.2.1.3-dev
```

From OracleWebLogic/samples/12213-domain, run:

```
$build.sh
```

or

```
$ docker build -f Dockerfile -t 12213-weblogic-domain-in-volume .
```

To make sure that you now have this image in place, run:

```
$sudo docker images
```

Running the Oracle WebLogic Server Administration Server Container

When you run the Oracle WebLogic Server domain image to start your container, an Administration Server will start running in the container by default. The default Administration Server name is “AdminServer”, the default port number is 7001, and the default Administration Server container name is “wlsadmin”.

***NOTE**: When running more than one domain in the same, single host you must change the Docker port number where the server is running and the container name.*

To create the WebLogic Server domain on a host volume, you can simply call the **docker run** command; “12213-weblogic-domain-in-volume” is the Oracle WebLogic Server domain image tag. The sample Dockerfiles define **container-scripts/createWLSDomain.sh** as the default CMD (command).

Enter your WebLogic Server credentials in the **properties/domain_security.properties** file and define the properties of your domain in **properties/domain.properties,** as described in the table above.

This sample creates the domain home on a host volume. In the docker run command, you will map the host volume with the property *-v ${domainhostvol}:/u01/oracle/user_projects/domains.* The directory *${ domainhostvol} is defined in the file* **properties/domain.properties** DOMAIN_HOST_VOLUME.

To start the Administration Server, run:

```
$run-admin-server.sh

or

$ container-scripts/setEnv.sh

$ sudo docker run -d -p 9001:7001 -p 9002:9002 --name ${adminhost}     \
--hostname ${adminhost} -v ${scriptDir}/properties:/u01/oracle/properties  \
-v ${domainhostvol}:/u01/oracle/user_projects/domains                     \
${ENV_ARG} 12213-weblogic-domain-in-volume
```

To obtain the IP address of the Administration Server container, run:

```
$ sudo docker inspect --format '{{ .NetworkSettings.IPAddress }}' ${adminhost}
```

Sample return value: **xxx.xx.x.xx**

Now you can access the WebLogic Server Administration Console at http://**xxx.xx.x.xx**:7001/console.

NOTE*: If you have several Oracle WebLogic Server domains running on the same host (several Administration Servers), change the –name (name of the Administration Server container) and the –p (Docker port to which the Administration Server port is mapped).*

Running Oracle WebLogic Server Managed Server Containers

When the domain is created by invoking **container-scripts/createWLSDomain.sh**, a cluster with a defined number of Managed Servers (CONFIGURED_MANAGED_SERVER_COUNT) are configured in the cluster. Each Managed Server in the domain has a unique name, calculated by using a base name (MANAGED_SERVER_NAME_BASE) followed by a number. All Managed Servers listen on port 8001.

Managed Server containers communicate to the Administration Server container by linking (--link command) using the Administration Server container name *${adminhost}*. The Administration Server container name defaults to “wlsadmin”. When there is more than one domain running on the same host, the Administration Server container name needs to be unique. Change the Administration Server container name by using the –name *${adminhost}* parameter and matching the name given in the –link command of each Managed Server container.

To start the Managed Server, run the script:

```
$run-managed-server.sh <managed-server-name> <maped-port>

or

$ container-scripts/setEnv.sh

$ sudo docker run -d -p ${maped-port}:${CUSTOM_MANAGED_SERVER_PORT}                    \
-v ${scriptDir}/properties:/u01/oracle/properties ${ENV_ARG} --volumes-from {adminhost} \
--link ${adminhost}:${adminhost} --name ${managed-server-name}                          \
12213-weblogic-domain-in-volume /u01/oracle/container-scripts/startManagedServer.sh
```

NOTE*: You must assign a new, unique Docker listen port (-p) when you create an additional Managed Server container on a host OS where a Managed Server container is already running. This prevents multiple Managed Servers running on the same host OS from listening on the same listen port. For example, Managed Server1 will map the port using the property –p 9810:8100, Managed Server2 will map the port using the property –p 9820:8100.*

Additional Samples

The preceding sample configures a WebLogic Server domain on a shared host volume. There are two samples that show you how to create a WebLogic domain home inside of an image. The sample, OracleWebLogic/samples/12213-domain-home-in-image, creates a domain home inside the image using WLST and the sample, OracleWebLogic/samples/12213-domain-home-in-image-wdt, creates a domain home inside the image using WDT. For more information about the WebLogic Deploy Tooling, see the WebLogic Deploy Tooling GitHub Project.

Considerations for Running Oracle WebLogic Server with Docker

- When a Docker container is restarted, its IP address changes, and Oracle WebLogic Servers running in the Docker container will now have a new address. Applications, as well as other servers that were communicating with the server before the container restart, will be unable to communicate. A solution to the IP address change after a container restart, is to configure a DNS server on Docker and configure WLS domains to use DNS names.
- Oracle WebLogic Server configuration, server logs, file stores, and such, are all kept in the container file system. When a Docker container is destroyed, you will lose your entire file system. There are two alternatives to this:
  - Use the host file system to store the container’s local file system.
  - Maintain a “data-only” container to store your domain file system.

  To minimize the dependency on the file system, we recommend:
  - Keep your stores, such as TLog and JMS stores, in the database.

  - If you do XA transactions, use “XATransactions without TLog Write”; this minimizes writing to the TLog.

- Clustered Oracle WebLogic Servers must communicate between themselves and with the Administration Server. Docker containers running on different host machines do not have the necessary visibility and access to communicate directly with other containers. Consequently, the use of Oracle WebLogic Server configurations that span multiple hosts’ operating systems is not supported at this time. A possible alternative configuration is to run your entire Oracle WebLogic Server domain on a single host.
- Oracle recommends patching WebLogic Server images with the latest Patch Set Updates (PSU). These patches contain important security fixes required to secure your deployments in production. There are two ways to patch the WebLogic Server images:
  - The Oracle WebLogic Image Tool helps simplify and automate the creation and patching of Docker images for WebLogic Server. This tool lets you create a new Linux based image, with installations of a JDK and WebLogic Server, and optionally, configure a WebLogic domain with your applications, apply WebLogic Server patches, or update an existing image. You will find documentation and samples in the Image Tool GitHub repository.
  - You can also patch your WebLogic images manually by creating Dockerfiles which apply the patches to the Oracle WebLogic Server generic, developer, or slim installation images. Refer to the sample in the Docker GitHub repository OracleWebLogic/samples/12213-patch.
- Security concerns have been raised regarding Docker and Linux containers:
  - One area of concern is whether it is possible to isolate code running in separate containers. There are no known issues at this time impacting the ability to run Oracle WebLogic Server in such an environment.
  - Another area of concern is the source of Docker images. You should obtain Docker images only from trusted sources and be aware of the frequency of updates and the nature of the controls on Docker Hub.
  - You should stay current with Docker and Linux technology and remain aware of security issues that are raised in each.
  - Docker containers default network mode of “Bridge Networking” does not support multicast. Docker containers “Host Networking” supports multicast but provides less isolation because it uses the host networking stack. We recommend the use of unicast as the Oracle WebLogic Server clustering protocol when running in Docker containers.

Conclusion

Docker technology offers the promise of simplifying operations and reducing cost due to the portable characteristics of its artifacts and ease of distribution across Linux environments. Oracle has responded to the growing interest of our customers by certifying Oracle WebLogic Server to run in Docker containers, and by providing images, Dockerfiles, and scripts that support the creation of Oracle WebLogic Server configurations running in Docker containers. We hope these are useful and will improve the scope of our support for Docker environments over time.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

- blogs.oracle.com/oracle
- facebook.com/oracle
- twitter.com/oracle
- oracle.com

Hardware and Software, Engineered to Work Together

Oracle WebLogic Server on Docker Containers
April 2020
Monica Riccelli WebLogic Server PM

Oracle is committed to developing practices and products that help protect the environment

CONNECT WITH US