Oracle WebLogic Server on Docker Containers

ORACLE WHITE PAPER |JUNE 2015

Table of Contents

Oracle WebLogic Server on Docker

We are announcing that Oracle WebLogic Server (WLS) is now certified to run on Docker containers. As part of this certification, we are releasing Dockerfiles and supporting scripts on GitHub to build images of the Oracle WebLogic Server. These images are built as an extension of existing Oracle Linux images. You can use these Oracle WebLogic Server Docker images or create your own.

Docker is a platform that enables users to build, package, ship and run distributed applications. Docker users package up their applications, and any dependent libraries or files, into a Docker image. Docker images are portable artifacts that can be distributed across Linux environments. Images that have been distributed can be used to instantiate containers where applications can run in isolation from other applications running in other containers on the same host operating system.

The table below describes the certification provided for various Oracle WebLogic Server versions. You can use these combinations of Oracle WebLogic Server, JDK, Linux and Docker versions when building your Docker images.

Oracle WebLogic Server Version	JDK Version	HOST OS	Kernel Version	Docker Version
12.1.3.0.0	7/8	Oracle Linux 6 UL 5+	Unbreakable Enterprise Kernel Release 3 (3.8.13)+	1.3.3+
12.1.3.0.0	7/8	Oracle Linux 7 UL 0+	Unbreakable Enterprise Kernel Release 3 (3.8.13)+ Or Red Hat Compatible Kernel (3.10)+	1.3.3+
12.1.3.0.0	7/8	Red Hat Enterprise Linux 7+	Red Hat Enterprise Linux Kernel (3.10)+	1.3.3+

For additional details on the most current Oracle WebLogic Server supported configurations and support statement please refer to Oracle Fusion Middleware Certification Pages.

These Dockerfiles and scripts we have provided enable users to create clustered and non-clustered Oracle WebLogic Server domain configurations, including both development and production running on a single host operating system or VMs. Each server running in the resulting domain configurations runs in its Docker container, and is capable of communicating as required with other servers. Other configurations and approaches are possible; this paper will guide you to create these configurations.

Oracle WebLogic Server Docker Images

We are releasing Dockerfiles and supporting scripts to build Oracle WebLogic Server Docker images on GitHub. These images are built as an extension of existing Oracle Linux image 7.0, with JDK 7, and the Oracle WebLogic Server 12c (12.1.3) installations.

Two type of images are posted,

1. An Oracle WebLogic Server image created with the generic installer.
2. An Oracle WebLogic Server image created with the ZIP installer.

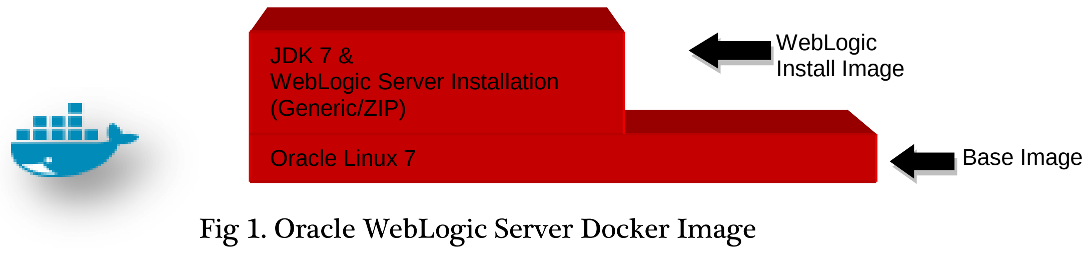

Fig 1. Oracle WebLogic Server Docker Image

Custom Built Oracle WebLogic Server Images

You can create your own Oracle WebLogic Server Docker images. To help you with this, we have posted Dockerfiles and scripts on GitHub Oracle WebLogic Server Docker files as examples for you to get started.

What are the prerequisites to build the custom WLS images?

1. Oracle Linux base image.
2. Dockerfiles and scripts from GitHub.
3. Oracle WebLogic Server Generic installer or ZIP installer.
4. Corresponding JDK.

Dockerfiles and Scripts on GitHub

Dockerfile

The Dockerfile to create an Oracle WebLogic Server install image performs the following functions:

1. Extend the Oracle Linux base image
2. Installs the JDK
3. Installs WLS using the (WLS generic/ZIP installers) in silent mode

After creating your Oracle WebLogic Server install images, you can extend them to have a base Oracle WebLogic Server domain configured.

The Dockerfile to create an Oracle WebLogic Server domain image performs the following functions:

- Extend the Oracle WebLogic Server install image.
- Configure a WLS domain by calling Oracle WebLogic Server Scripting Tool (WLST) scripts. The domain has one Admin server, JMS server, Data Source, enables JPA 2.1 and JAX-RS 2.0.

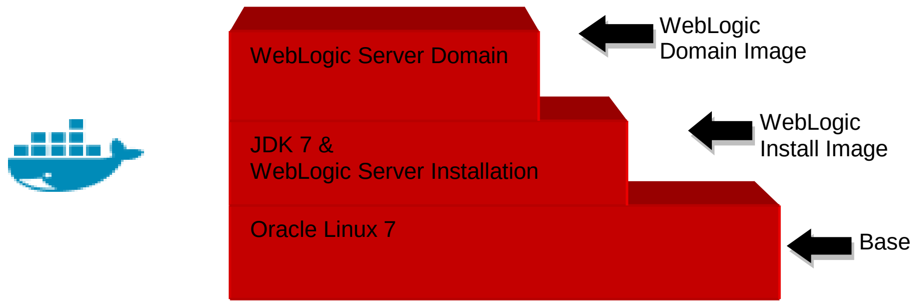

Fig 2. Oracle WebLogic Server Domain Image

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

1. Admin Server container with a single Oracle WebLogic Server Admin Server.
2. Managed Server container with a Node Manager, which adds itself as a machine to the Admin Server and a Managed Server.

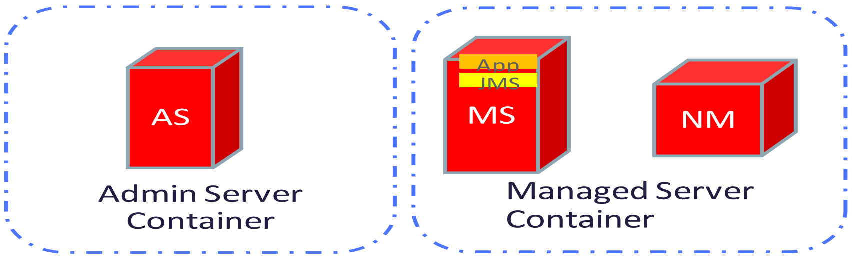

Fig 3. Types of Oracle WebLogic Server Containers

Scripts

The scripts aid in the creation of the Oracle WebLogic Server image and serve as examples to extend the Oracle WebLogic Server image with the configuration of an Oracle WebLogic Server domain.

Clustering Oracle WebLogic Server on Docker Containers

Oracle WebLogic Server has a Machine concept, which is an operational system with an agent, the Node Manager. This resource allows Oracle WebLogic Admin Server to create and assign Managed Servers of an underlying domain in order to expand an environment of servers for different applications and resources, and also to define a Cluster. By using Machines from containers, you can easily create a Dynamic Cluster by simply firing new Managed Server containers. With WLST, your cluster can scale in and out.

These Docker containers enable users to create clustered and non-clustered Oracle WebLogic Server domain configurations. Each server running in the domain runs in its own Docker container and is capable of communicating as required with other servers on the same host.

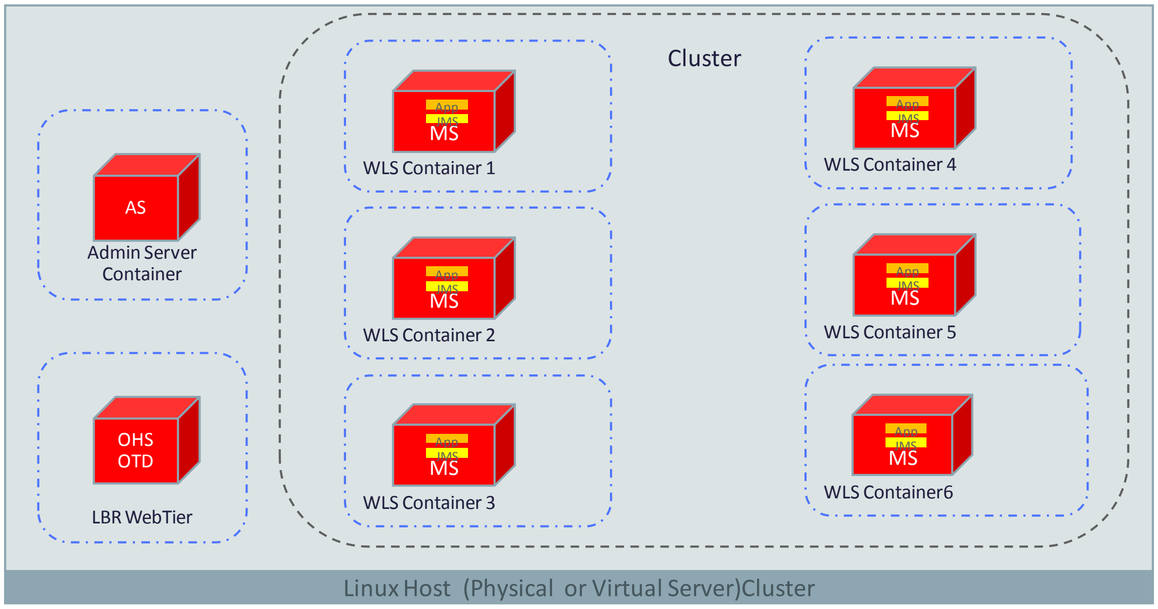

Fig 4. Clustering Oracle WebLogic Server on Docker Containers across Single Host

The advantages of this topology

- Good for traditional-like deployments.
- Easy to deploy containers from Oracle WebLogic Server domain images.
- Easy to scale up and down the cluster.
- Good for developers.

- No need to install or configure anything on host except for Docker binaries.

A topology which is in line with the "Docker-way" for containerized applications and services consists of a container designed to run only an administration server containing all resources, shared libraries and deployments. The Docker image includes all domain resources pre-defined, applications and shared libraries deployed upfront, and no Managed servers or clusters configured.

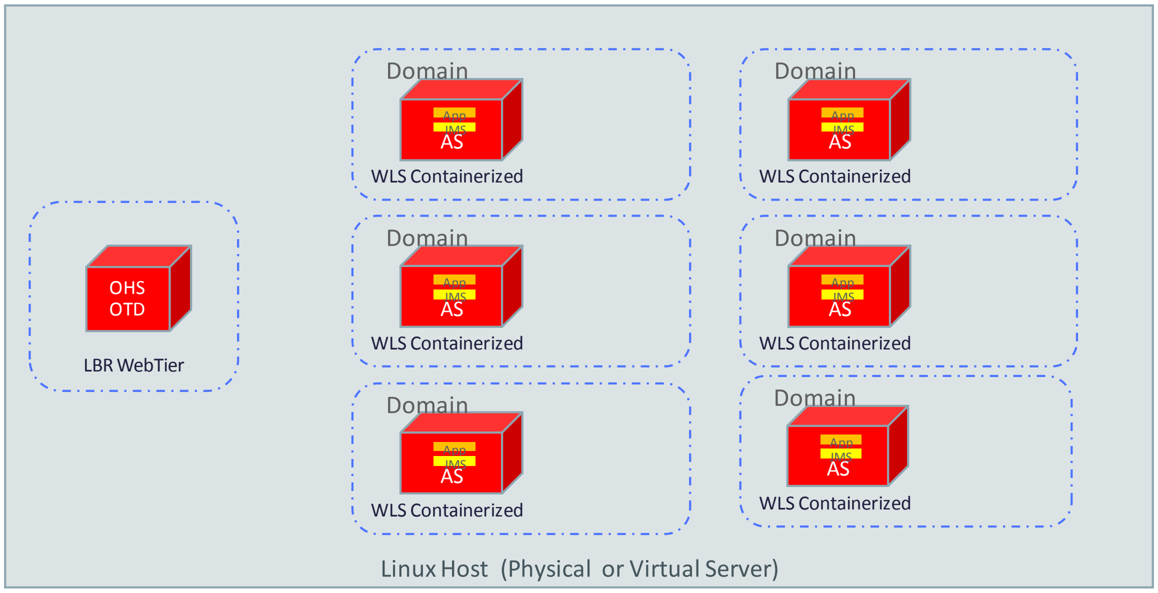

Fig 5. Containerized Oracle WebLogic Server Applications on a Single Host

The advantages of this topology

- The "Docker-way" for containerized applications and services.
- Containers are easily repeatable
- Each container is an instance of the same Oracle WebLogic Server domain.

The containers can all be on a single physical or virtual server Linux host as in figure 5 or on multiple physical or virtual server Linux hosts as in figure 6.

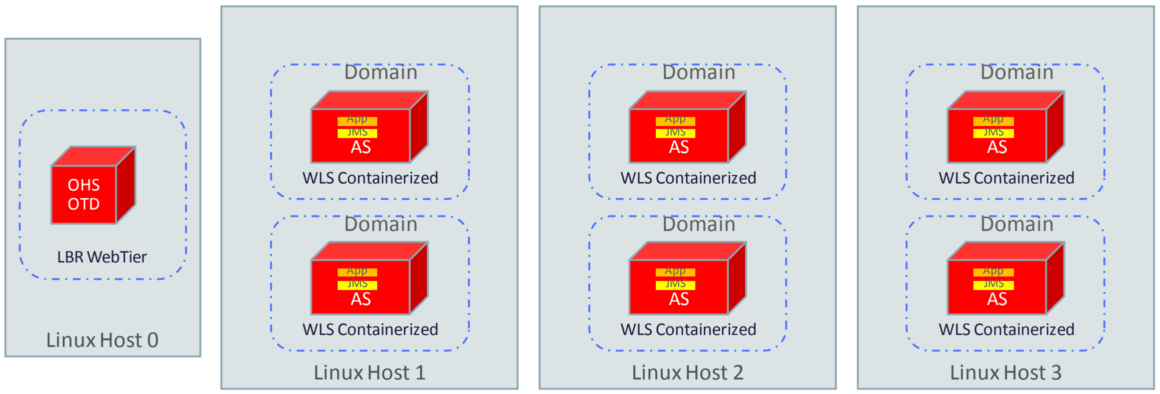

Fig 6. Containerized Oracle WebLogic Server Applications on Multiple Hosts

A different topology is a single Oracle WebLogic Server domain running on a single container on a single Linux host communicating to an Oracle WebLogic Server on a remote host and a Database.

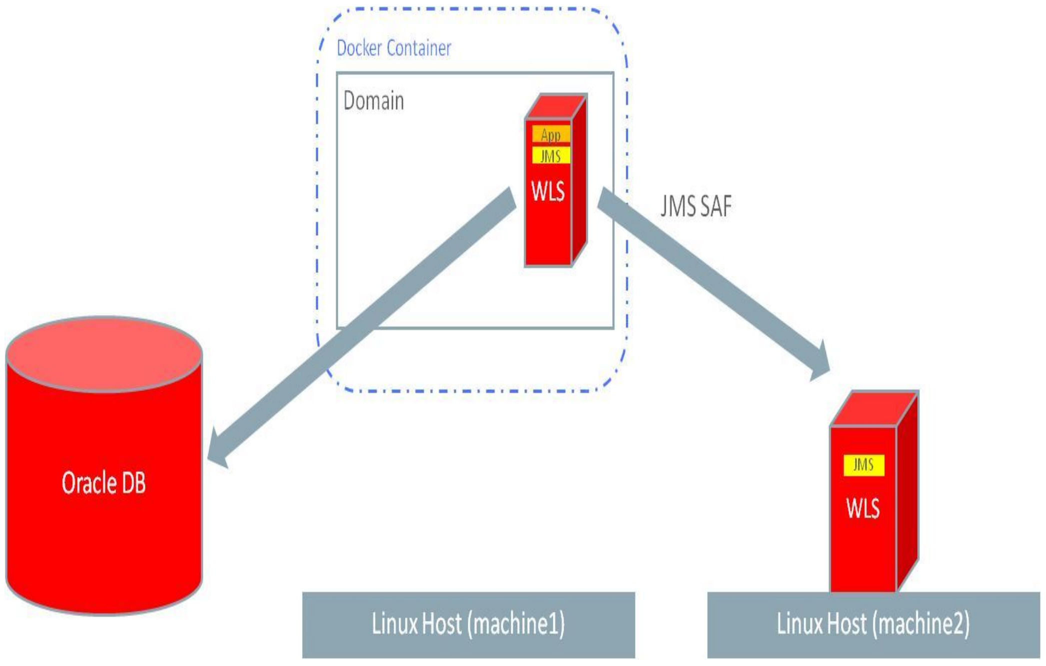

Fig 7. Single Oracle WebLogic Server domain in a Docker Container on a Single Host

How to Build and Run

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

Dockerfiles

Two Dockerfiles for Oracle WebLogic Server 12c (12.1.3) under the /OracleWebLogic/dockerfiles/12.1.3 subdirectory, one Dockerfile to build an Oracle WebLogic Server 'developer' install image and a second Dockerfile to build an Oracle WebLogic Server 'generic' install image.

Dockerfile.developer

Dockerfile.generic

Scripts

Under the subdirectories /OracleWebLogic/dockerfiles and OracleWebLogic/samples/12c-domain/container-scripts you will find supporting scripts which together with the Dockerfiles above are necessary to build an Oracle WebLogic Server image or to configure a domain at the time that the container is started.

buildDockerImage.sh -builds the WLS image using the WLS installation Dockerfile instructions.

createMachine.sh – starts a Node Manager in the container and calls addMachine.sh to start Node Manager and add the Node Manager machine.

createServer.sh – starts a Node Manager in the container and calls add-server.py to configure a Managed Server in the machine create by add-machine.py.

create-wls-domain.py – WLST script configures a base domain with one Admin server, JMS server, JSP, Data Source.

add-machine.sh –set environment and call add-machine.py

add-machine.py – WLST scripts to create a machine using the Managed Server container name.

add-server.py – WLST scripts to create a Managed Server.

rm_containers.sh – removes all running containers

clean-up-docker.sh - removes all ghost containers and all ghost images

commEnv.sh - enable JPA 2.1 support

Building a Oracle WebLogic Server Image

First decide which installation type you want to use either generic or ZIP installer, download the required Oracle WebLogic Server installer and JDK in the dockerfiles/12.1.3 folder. Go into the dockerfiles folder and run the buildDockerImage.sh script as root.

$ sudo sh buildDockerImage.sh -h

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

Parameters:

-d: creates image based on 'developer' distribution

-g: creates image based on 'generic' distribution

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

Samples for Oracle WebLogic Server Domain Creation

To give users an idea on 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 12c for the Developer distribution and Generic distribution; check the folder samples/12c-domain.

Sample Domain for Oracle WebLogic Server 12c

This Dockerfile will create an image by extending oracle/weblogic:12.1.3-dev (from the Developer distribution). It will configure a base_domain with the following settings:

- JPA 2.1 enabled
- JAX-RS 2.0 shared library deployed
- Admin Username: weblogic
- Admin Password: welcome1
- Oracle Linux Username: oracle
- Oracle Linux Password: welcome1
- Oracle WebLogic Server Domain Name: base_domain

Write your own Oracle WebLogic Server domain with WLST

The best way to create your own, or extend domains is by using WLST. The WLST script used to create domains in the Dockerfile is create-wls-domain.py. This script by default adds JMS resources and a few other settings. You may want to tune this script with your own setup to create Data Sources and Connection pools, Security Realms, deploy artifacts, and so on. You can also extend images and override the existing domain, or create a new one with WLST.

Building a sample Docker Image of Oracle WebLogic Server Domain

To try a sample of an Oracle WebLogic Server image with a domain configured, follow the steps below:

Make sure you have oracle/weblogic:12.1.3-dev image built. If not go into dockerfiles and call

```
$sudo sh buildDockerImage.sh [-d|-g]
```

Go to folder samples/12c-domain

Run the following command:

$sudo docker build -t samplewls:12.1.3 .

Make sure you now have this image in place with

$sudo docker images

Running Oracle WebLogic Server Admin Server Container

When you use the Oracle WebLogic Server domain image to start your container an Admin Server will start running in the container by default. The default Admin Server name is "AdminServer," the default port configuration is 8001 and the default Admin Server container name is "wlsadmin". When running more than one domain in the same single host you must change the Admin Server name, port and container name.

To start the Oracle WebLogic Admin Server, you can simply call **docker run -d samplewls:12.1.3** command, "samplewls:12.1.3" is the Oracle WebLogic Server domain image tag. The samples Dockerfiles define startWebLogic.sh as the default CMD (command).

$ sudo docker run -d --name=wlsadmin samplewls:12.1.3

To obtain the IPAddress of the Admin Server Container run the command

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

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

Now you can access the Admin Server Web Console at http:// **xxx.xx.x.xx**:8001/console.

Note that If you have several Oracle WebLogic Server domains running on the same host (several Admin Servers), change the –name (name of the Admin Server container) and the –p (port of the Admin Server)

Running Oracle WebLogic Server Managed Server Container

Managed Server containers have a Node Manager and a Managed Server running in it. These Managed Server containers communicate to an Admin Server Container by linking (--link command) using the Admin Server container name. The Admin Server container name defaults to "wlsadmin" . When there are more than one domain running on the same host then the Admin Server container name needs to be unique, you need to change the Admin Server container name by using the –name parameter and matching the name given in the –link command of each Managed Server container.

There are 3 different ways to start a Managed Server Container:

Start Node Manager (Manually):

$ sudo docker run -d --link wlsadmin:wlsadmin <image-name> startNodeManager.sh

Start Node Manager and Create a Machine Automatically:

$ sudo docker run -d --link wlsadmin:wlsadmin <image-name> createMachine.sh

Start Node Manager, Create a Machine, and Create a Managed Server Automatically

$ sudo docker run -d --link wlsadmin:wlsadmin <image-name> createServer.sh

Parameters you can use:

```
$ sudo docker run -d –link wlsadmin:wlsadmin \
    -p <NM Port>:5556 –p <MS Port>:<MS Port> \
    –name=<Container name> \
    -e MS_HOST=<Host address where Managed Server container runs> \
    -e MS_PORT=<Managed Server port> \
    -e NM_HOST=<Host address where Managed Server container runs> \
    -e NM_PORT=<Node Manager Port (should match the port in the –p)> \
    <image name> \
    <createMachine.sh, startNodeManager.sh, createServer.sh>
```

The scripts have a list of variables that must be properly configured,

Variable	Meaning
ADMIN_USERNAME	username of the AdminServer 'weblogic' user. Default: weblogic
ADMIN_PASSWORD	password of ADMIN_USERNAME. Defaults to value passed during Dockerfile build. ('welcome1' in samples)
ADMIN_URL	t3 URL of the AdminServer. Default: t3://wlsadmin:8001
CONTAINER_NAME	name of the Machine to be created. Default: node manager_ + hash of the container
NM_HOST	IP address where Node Manager can be reached. Default: IP address of the container
NM_PORT	Port of Node Manager. Default: 5556

MS_HOST	IP address where Managed Server can be reached. Default: IP address of the container
MS_PORT	Port of Managed Server. Default: 7001

Example:

If you want to run "Single-Host" configuration on a remote server, you must expose ports and addresses of Admin Server, Managed Servers, and Node Manager.

The following command will run a Managed Server container.

```
$ sudo docker run -d –link wlsadmin:wlsadmin -p 5556:5556 --name="wlsnm0" -e
NM_HOST="xx.xxx.xx.xxx" -e NM_PORT="5556" samplewls:12.1.3 createMachine.sh


$ sudo docker run -d --link wlsadmin:wlsadmin   -p 7003:7003
-e MS_HOST=xx.xxx.xx.xxx -e MS_PORT=7003 samplewls:12.1.3 createServer.sh

$ sudo docker run -d --link wlsadmin:wlsadmin   -p 7002:7002
-e MS_HOST= xx.xxx.xx.xxx  -e MS_PORT=7002 samplewls:12.1.3 createServer.sh
```

Note that you must assign a new, unique listen port 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.

If you used the createServer.sh command

1. Now access the Admin Server Console at http://<admin-container-ip>:8001 /console
2. Go to Environment > Machines and you should now have a Machine registered
3. Your Node Manager and Managed Server should also be configured.
4. Do to Environment > Machines > Servers click on the Control tab and start your server

Oracle WebLogic Server Docker Container Communicating With Servers on a Remote Host

Another possible topology is to run a single Admin Server container communicating with Oracle WebLogic Server running on a remote host. Use --add-host so that container assumes the IP address of host where it is running instead of the local container IP address.

```
$ sudo docker run -d -p 8001:8001 --net=host \
    --add-host=hostname:<host ip address where container is running> \
    --name wlsadmin samplewls:12.1.3
```

For this configuration to work the following configurations are necessary:

- The listen address of the AdminServer in the Docker container has to be configured.
- The listen address of the AdminServer in the remote host has to be configured.
- The client must use the hosts IP addresses to get the initial context for JNDI lookup.

Additional Considerations Running Oracle WebLogic Server with Docker

- When a Docker container is restarted its IPAddresses changes, Oracle WebLogic Servers running in the Docker container will now have a new address. Applications as well as others servers that were communicating with the server before the container restart, will be unable to communicate. Configuring a DNS server on Docker and configure WLS domains to use DNS names is a solution to the IPAddress change after a container restart.
- Oracle WebLogic Server configuration, server logs, file stores etc 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 the "XATransactions without TLog Write", this minimizes the writing to the TLog.
- Clustered Oracle WebLogic Servers must communicate between themselves and with the Admin Server. Docker containers running on different host machines do not have the necessary visibility and access to communicate directly with other containers. Consequently use of Oracle WebLogic Server configurations that span

multiple hosts operating systems is not supported at this time. Possible alternative configuration is to run your entire Oracle WebLogic Server domain on a single host.

- To patch or upgrade your Oracle WebLogic Server images created with the Oracle WebLogic Server generic installation image, follow the following steps:
  1. Upgrade/patch by extending the Oracle WebLogic Server install Docker image.
  2. Use the Docker **cp** (copy) command to copy your domain folder to destination either the host or a "data-only" container
  3. Remove Container
  4. Run new container from extended image (with upgrade/patch).
  5. Use the Docker **cp** (copy) command to copy back your domain folder to the upgraded container
- 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 from each other. There are no known issues impacting the ability to run Oracle WebLogic Server in such an environment at this time.
  - Another area of concern with respect to security is the source of Docker images. One should only obtain Docker images from trusted sources and one needs to be aware of the frequency of updates and the nature of the controls on Docker Hub.
  - Customers 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 since 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 seek to 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
July 2015
Monica Riccelli WebLogic Server PM
Bruno Borges Technical Advisor

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