



# **BUILDING ORACLE STACKS WITH DOCKER COMPOSE**

HANDS-ON LAB

Collaborate 2018

SENIOR CLOUD AND DEVOPS SOLUTION SPECIALIST

VERTICAL TRAIL LLC

# THE INNOVATION ACCELERATORS

BRINGING YOUR BUSINESS



Cloud-Enabled  
Innovation



Data-Driven  
Solutions



High-Value  
Insights

USING OUR PROVEN DELIVERY FRAMEWORK.



Accelerate innovation. Streamline development. Drive revenue.  
Get on the Trail.

**CONTACT:**

**Eric Mader** | Senior Cloud and DevOps Solution Specialist | [emader@verticaltrail.com](mailto:emader@verticaltrail.com)

**Rob Koury** | Managing Partner | [rkoury@verticaltrail.com](mailto:rkoury@verticaltrail.com)

Phone: **847-778-1837**

## CONTENTS

<b>Prerequisites</b>	<b>2</b>
<b>Part 1: Setup and Image Downloads</b>	<b>2</b>
<b>Part 2: Build a Stack using Docker Utilities</b>	<b>7</b>
<b>Part 3: Build a Stack using Docker Compose</b>	<b>12</b>

## PREREQUISITES

This lab requires the following:

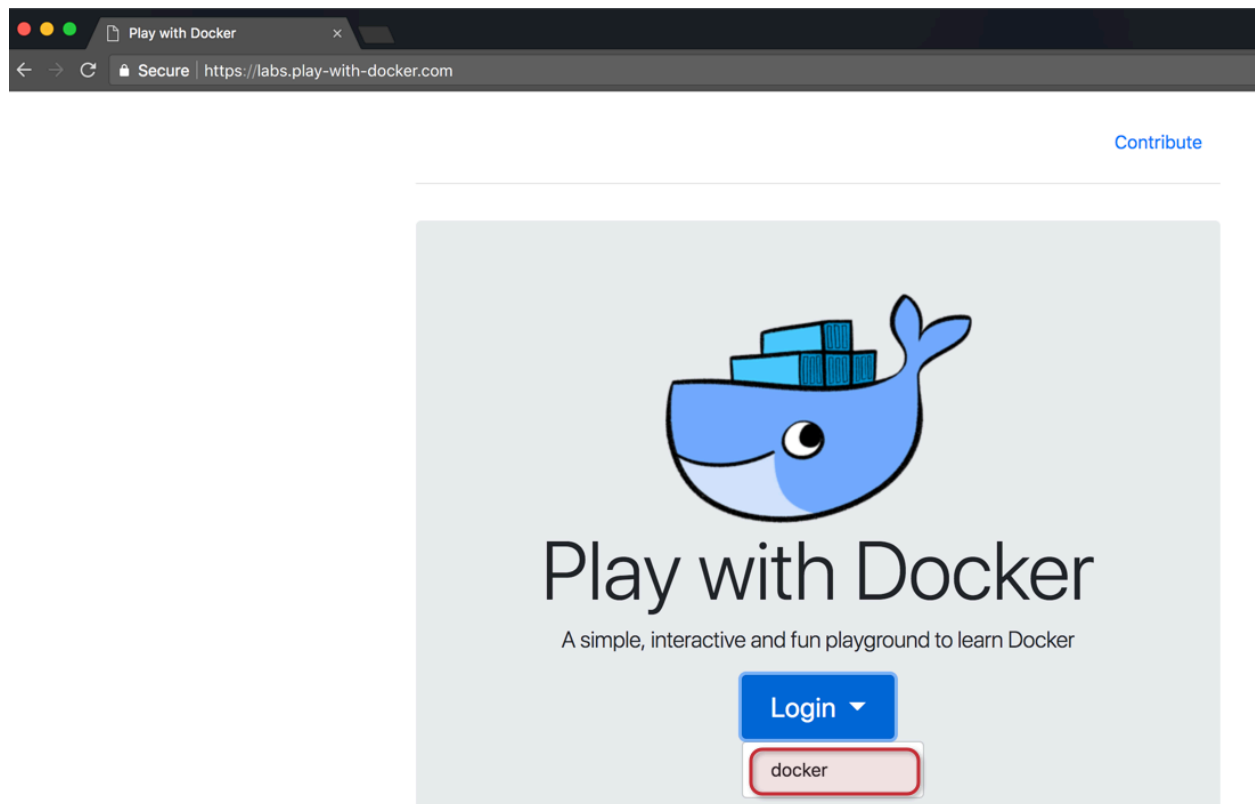
- Free login Docker ID: <https://cloud.docker.com/>
- Free Oracle login account: <https://profile.oracle.com/myprofile/account/create-account.jspx>
- Laptop or tablet with a web browser capable of running Play with Docker VM sessions: <https://labs.play-with-docker.com/> (most modern, up-to-date browsers will do the trick)

## PART 1: SETUP AND IMAGE DOWNLOADS

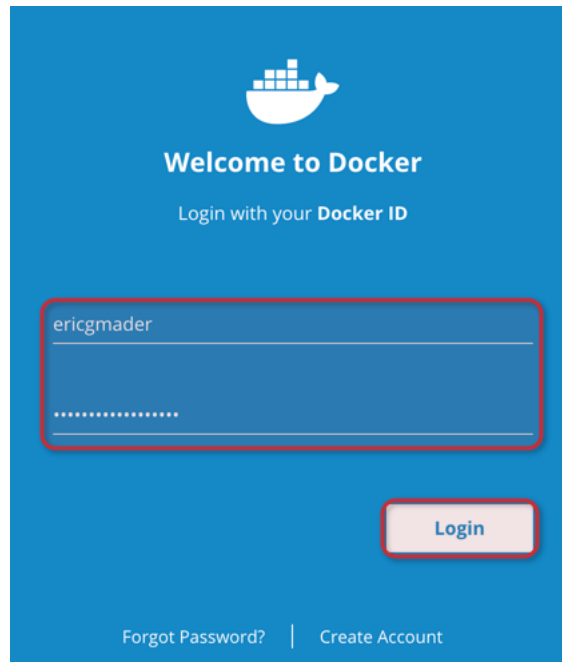
Docker Labs provides a web-based, lightweight laboratory environment called Play with Docker. This platform is completely free, and will work with most modern, up-to-date browsers. **Please note that VMs are automatically deleted and unrecoverable after 4 hours, so do not store any critical information or lab notes directly on the Play with Docker instances.**

Alternatively: The Play with Docker VMs include all of the necessary software pre-installed. The lab will also work on any system with Docker and Docker Compose version 17.12 or later. However, the system might not meet the minimum specifications to run the Docker containers, as has been validated with the Play with Docker VMs.

1. In a web browser, navigate to the Play with Docker site via <https://labs.play-with-docker.com/>
2. On the home page, navigate to **Login > docker**:



3. Login using a Docker ID and password.

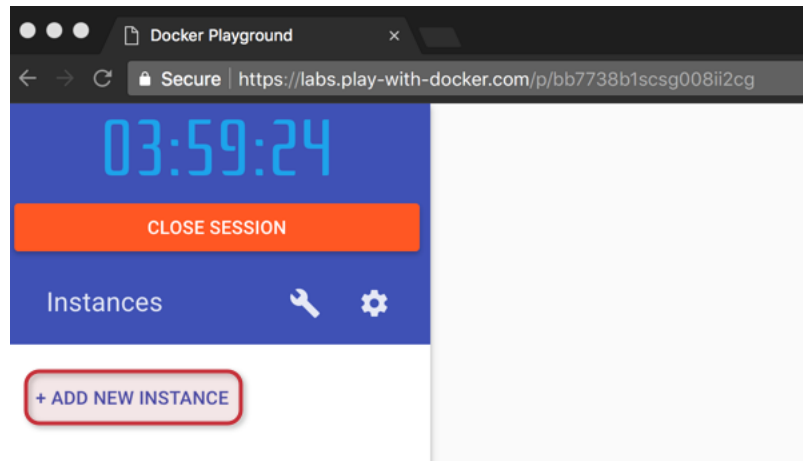


4. Click **Start**:

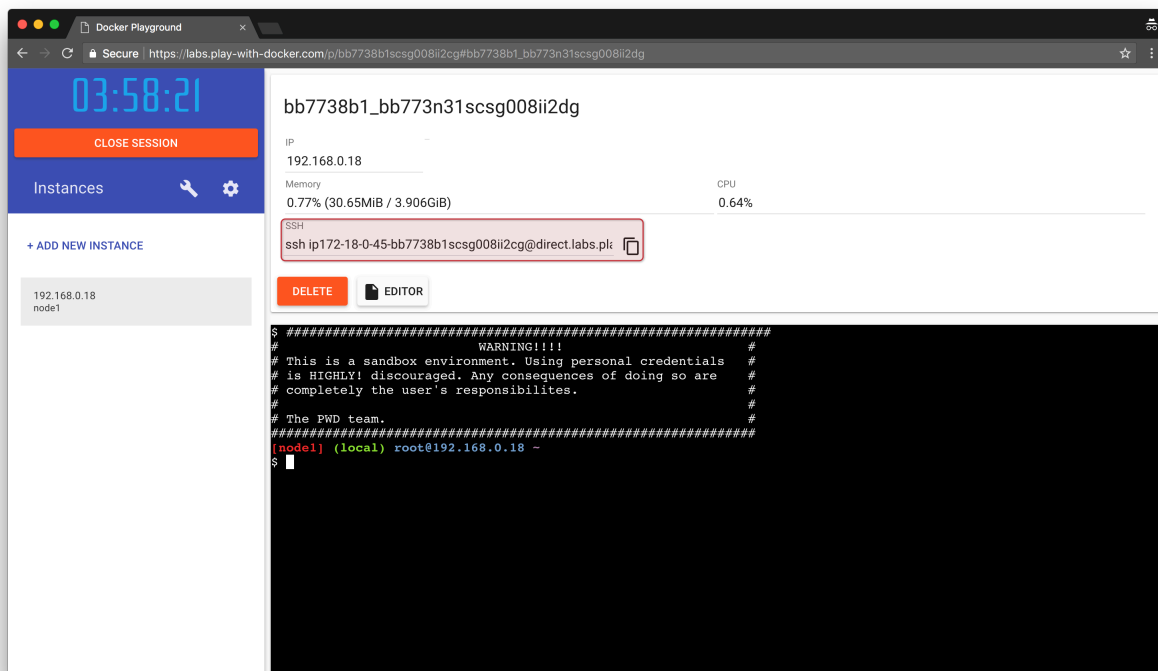


5. In the lab window, click **Add New Instance**:

## Building Oracle Stacks with Docker Compose – Hands-on Lab - 4



- The console for the new instance will appear. Note that the 4-hour counter has started in the upper left of the screen. Also, if desired, the instance can be connected to via an SSH client such as PuTTY.



7. The instance is already logged into the Docker Hub, which stores the MySQL container. Download the latest MySQL container instance using the **pull** option of the docker command:

```
$ docker pull mysql:latest

latest: Pulling from library/mysql
2a72cbf407d6: Pull complete
38680a9b47a8: Pull complete
4c732aa0eb1b: Pull complete
c5317a34eddd: Pull complete
f92be680366c: Pull complete
e8ecd8bec5ab: Pull complete
2a650284a6a8: Pull complete
5b5108d08c6d: Pull complete
beaff1261757: Pull complete
c1a55c6375b5: Pull complete
8181cde51c65: Pull complete
Digest: sha256:691c55aabb3c4e3b89b953dd2f022f7ea845e5443954767d321d5f5fa394e28c
Status: Downloaded newer image for mysql:latest
```

8. Due to license restrictions regarding the distribution of the WebLogic installation media, the WebLogic 12.2.1.3 container must be downloaded directly from the Oracle Container Registry. Use the **login** option of the docker command to login to the Oracle Container Registry with valid Oracle credentials:

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

Username: ORACLE USERNAME
Password: ORACLE PASSWORD
Login Succeeded
```

9. After logging into the registry, download the 12.2.1.3 version of the WebLogic container using the **pull** option of the docker command (**NOTE:** this image is about 1.1 GB and may take several minutes to download):

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

12.2.1.3: Pulling from middleware/weblogic
8dc3ae811b63: Pull complete
4efd7b91b3ef: Pull complete
9f8c6a66986c: Pull complete
9e565d6ad1d3: Pull complete
c46907c1dea7: Pull complete
e378102878a1: Pull complete
bc0642354402: Pull complete
Digest: sha256:23d9048f6eb73bd16f02a63bf78427bc6c1cb9cb3eba149e6da1173ae1e9bf1d
Status: Downloaded newer image for container-
registry.oracle.com/middleware/weblogic:12.2.1.3
```

10. The Oracle GitHub repository contains scripts for building Docker images from original installation media for many Oracle products, including Database, Web Tier, and GoldenGate. The repository also contains several sample Dockerfiles and Docker Compose artifacts. To simplify the scripting involved, this lab will use a sample Dockerfile from the GitHub repository that builds a Docker image containing a simple WebLogic domain.

Clone the Oracle GitHub repository and change to the WebLogic directory containing the domain sample:

```
$ git clone https://github.com/oracle/docker-images.git
Cloning into 'docker-images'...
remote: Counting objects: 7811, done.
remote: Compressing objects: 100% (90/90), done.
remote: Total 7811 (delta 40), reused 84 (delta 33), pack-reused 7688
Receiving objects: 100% (7811/7811), 9.73 MiB | 20.08 MiB/s, done.
Resolving deltas: 100% (4308/4308), done.
$ cd docker-images/OracleWebLogic/samples/12213-domain
```

11. In a text editor such as vi, edit the file name Dockerfile. replace the following line:

```
FROM oracle/weblogic:12.2.1.3-developer
```

With the image downloaded from the Oracle Container Registry:

```
FROM container-registry.oracle.com/middleware/weblogic:12.2.1.3
```

12. Use the **build** option of the docker command to create a Docker container image containing a simple domain, named *12213-domain*:

```
$ docker build -t 12213-domain .
*snip*
Successfully tagged 12213-domain:latest
```

13. The Vertical Trail GitHub repository contains a sample domain properties file, plus a PDF version of this lab. Change back to the home directory and clone this repository:

```
$ git clone https://github.com/VerticalTrail/Docker-Compose-Oracle-HOL.git
```

The environment is now ready to build an Oracle stack containing a WebLogic environment with a MySQL database!



## PART 2: BUILD A STACK USING DOCKER UTILITIES

To validate the Docker images, use the basic Docker utilities to build an initial Oracle stack.

14. Create a user-defined network specifically for the Docker containers. This isolates the network traffic from other Docker instances that may be running on the machine, and provides a network channel for the Docker containers to communicate. Use the **network create** option of the docker command to create this network:

```
$ docker network create --driver bridge oracle-net
```

15. Use the **network inspect** option of the docker command to validate the network configuration. Note that the specific configuration is automatically generated and may vary from what is listed below:

```
$ docker network inspect oracle-net

[
  {
    "Name": "oracle-net",
    "Id": "7c27b637ab5a04e9472d2422e773cee460b6d57491fd5996257ab01d6cf2abb4",
    "Created": "2018-02-13T01:54:21.53557946Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.21.0.0/16",
          "Gateway": "172.21.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

16. Create a folder in the home directory of the Docker host to store the MySQL database data. This will allow the container to be destroyed and re-created while persisting the data in the MySQL instance.

```
$ cd /root
$ mkdir mysql_data
$ chmod a+rwX -R mysql_data
```

17. Create a MySQL container instance attached to the new *oracle-net* network. This container uses the `mysql:latest` image downloaded from the Docker Hub, exposes TCP port 3306 to the Docker host machine, and mounts the `mysql_data` folder to the MySQL data storage directory within the container:

```
$ docker run -d --name mysql --hostname mysql --network oracle-net -p 3306:3306 \
-v /root/mysql_data:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=Abcd1234 mysql:latest
```

18. Watch the container startup process using the **logs** option of the docker command. When finished, use Ctrl+C to return to the command prompt:

```
$ docker logs -f mysql

*snip*

2018-02-13T02:10:15.649676Z 0 [Note] Event Scheduler: Loaded 0 events
2018-02-13T02:10:15.652763Z 0 [Note] mysqld: ready for connections.
Version: '5.7.21'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community
Server (GPL)
```

19. Create a folder in the home directory of the Docker host to store the WebLogic domain. This will allow the container to be destroyed and re-created while persisting the domain configuration.

```
$ mkdir domain_data
$ chmod a+rwX -R domain_data
```

20. Create a WebLogic Administration Server instance that is also attached to the *oracle-net* network. This container uses the `12213-domain` image generated in Part 1, exposes the WebLogic administration TCP port 7001 to the Docker host machine, and mounts the `domain_data` folder to the WebLogic domain directory within the container:

```
$ docker run -d --name wlsadmin --hostname wlsadmin --network oracle-net \
-p 7001:7001 --env-file ./domain.properties -e ADMIN_PASSWORD=Abcd1234 \
-v /root/domain_data:/u01/oracle/user_projects 12213-domain
```

21. Watch the container startup process using the **logs** option of the docker command. When finished, use Ctrl+C to return to the command prompt:

```
$ docker logs -f wlsadmin
```

\*snip\*

```
<BEA-002613> <Channel "Default[1]" is now listening on 127.0.0.1:7001 for protocols
iiop, t3, ldap, snmp, http.>
<Feb 13, 2018 2:21:29,846 AM GMT> <Notice> <Server> <BEA-002613> <Channel "Default" is
now listening on 172.21.0.3:7001 for protocols iiop, t3, ldap, snmp, http.>
<Feb 13, 2018 2:21:29,866 AM GMT> <Notice> <WebLogicServer> <BEA-000360> <The server
started in RUNNING mode.>
<Feb 13, 2018 2:21:29,903 AM GMT> <Notice> <WebLogicServer> <BEA-000365> <Server state
changed to RUNNING.>
```

22. Create a WebLogic Managed Server instance that is also attached to the *oracle-net* network. This container uses the 12213-domain image generated in Part 1, exposes the WebLogic managed server TCP port 8001 to the Docker host machine, and mounts the domain\_data folder to the WebLogic domain directory within the container. It also calls a shell script within the container that creates a managed server and joins it to the *wlsadmin* instance:

```
$ docker run -d --name wlsms --hostname wlsms --network oracle-net -p 8001:8001 \
--env-file ./domain.properties -e ADMIN_PASSWORD=Abcd1234 -e MS_NAME=wlsms \
-v /root/domain_data:/u01/oracle/user_projects 12213-domain createServer.sh
```

23. Watch the container startup process using the **logs** option of the docker command. When finished, use Ctrl+C to return to the command prompt:

```
$ docker logs -f wlsms
```

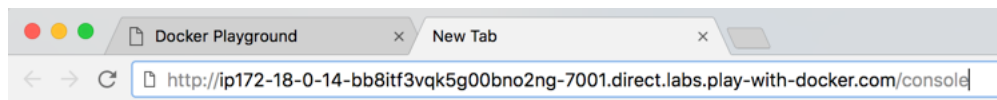
\*snip\*

```
<Feb 13, 2018 2:46:51,446 AM GMT> <Notice> <Server> <BEA-002613> <Channel "Default" is
now listening on 172.21.0.4:8001 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp,
http.>
<Feb 13, 2018 2:46:51,452 AM GMT> <Notice> <Server> <BEA-002613> <Channel "Default" is
now listening on 172.21.0.4:8001 for protocols iiop, t3, CLUSTER-BROADCAST, ldap, snmp,
http.>
<Feb 13, 2018 2:46:51,934 AM GMT> <Notice> <WebLogicServer> <BEA-000360> <The server
started in RUNNING mode.>
<Feb 13, 2018 2:46:52,067 AM GMT> <Notice> <WebLogicServer> <BEA-000365> <Server state
changed to RUNNING.>
```

24. At the top of the lab window, right-click the URL for port 7001 and click **Copy Link Address** (or the browser's equivalent):



25. Open a new browser tab and paste the link address in the navigation bar. Add the '/console' suffix:

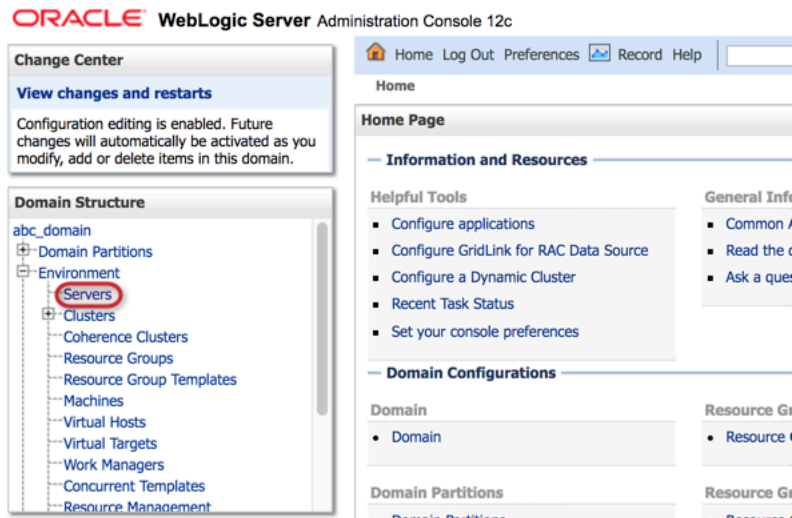


26. Login to the WebLogic Administration Console using the username **weblogic** and the password **Abcd1234**:

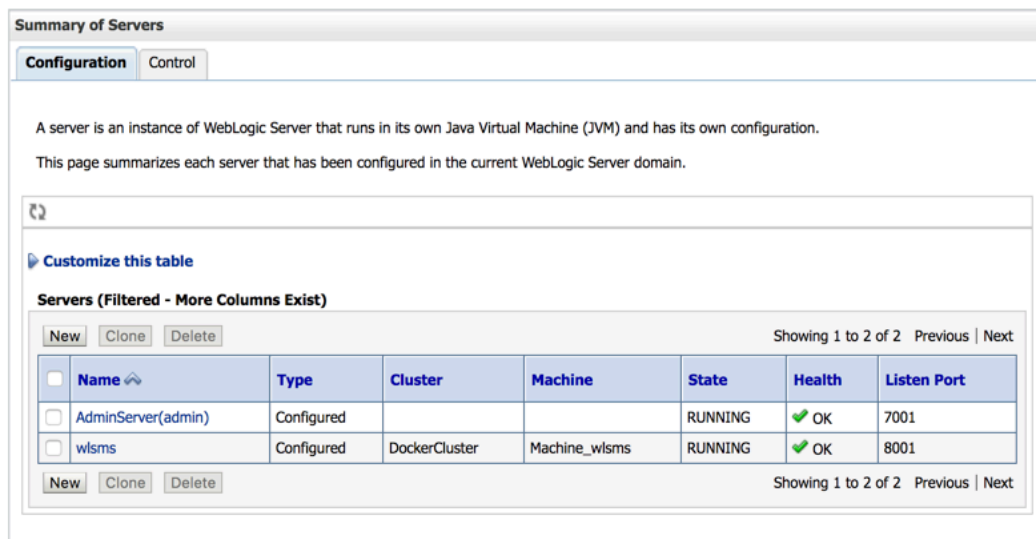


27. In the left Domain Structure navigation pane, click **Environment > Servers**:

## Building Oracle Stacks with Docker Compose – Hands-on Lab - 11



28. Both the AdminServer and the managed server instance (MS1) should be listed on the Summary of Servers page:



29. A simple Oracle stack is now running and ready for an application deployment! Stop the running containers using the **stop** option of the docker command:

```
$ docker stop wlsms wlsadmin mysql
```

30. In order to reuse the container names, remove the container instances:

```
$ docker rm wlsms wlsadmin mysql
```

31. Also delete the directories storing the persistent data for the Docker containers:

```
$ rm -r domain_data
```

```
$ rm -r mysql_data
```

## PART 3: BUILD A STACK USING DOCKER COMPOSE

The Docker images have now been validated by manually creating an Oracle stack using three Docker images built individually. Docker Compose can be used to define an entire Oracle stack based on multiple Docker containers that are spun up simultaneously.

32. A Docker Compose stack is defined in YML. In the /root directory, create a file **docker-compose.yml** with the following content:

**NOTE:** The stack is defined in Version 2 of the Docker Compose YML, in order to maintain compatibility with the Oracle Cloud Container Service. Version 3 is the latest version, which contains minor syntax changes.

```
version: '2'
services:
  mysql:
    image: 'mysql:latest'
    environment:
      - MYSQL_ROOT_PASSWORD=Abcd1234
  wlsadmin:
    image: 'wls12213-domain'
    environment:
      - ADMIN_PASSWORD=Abcd1234
      - DOMAIN_NAME=c181v_domain
      - ADMIN_PORT=7001
      - ADMIN_HOST=wlsadmin
      - CLUSTER_NAME=DockerCluster
      - PRODUCTION_MODE=dev
      - ADMIN_USERNAME=weblogic
      - ADMIN_NAME=AdminServer
      - MS_PORT=8001
      - DEBUG_FLAG=false
    ports:
      - '7001:7001/tcp'
    volumes:
      - 'domain_data:/root/domain_data'
    links:
      - mysql
  wlsms:
    image: 'wls12213-domain'
    command: createServer.sh
    environment:
      - ADMIN_PASSWORD=Abcd1234
      - DOMAIN_NAME=c181v_domain
      - ADMIN_PORT=7001
      - ADMIN_HOST=wlsadmin
      - CLUSTER_NAME=DockerCluster
```

```
- PRODUCTION_MODE=dev
- ADMIN_USERNAME=weblogic
- ADMIN_NAME=AdminServer
- MS_NAME=MS1
- MS_PORT=8001
- DEBUG_FLAG=false
ports:
  - '8001:8001/tcp'
volumes:
  - 'domain_data:/root/domain_data'
links:
  - wlsadmin

volumes:
  domain_data:
  mysql_data:
```

33. The entire Oracle stack can now be started using a single command:

```
$ docker-compose up
```

To validate, perform the steps outlined in the previous section, starting at Step 25.

To shut down the entire stack, issue a Ctrl+C at the command prompt. The MySQL database and the WebLogic domain home are both preserved in the local file system in the /root directory.