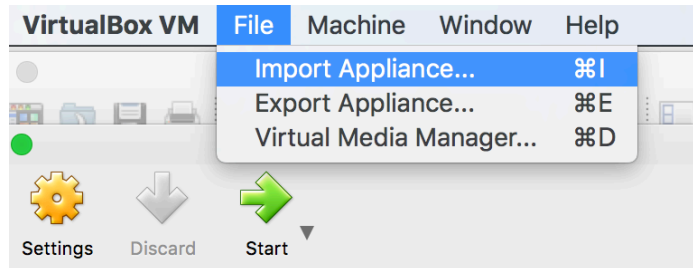


TECLO Docker Workshop

Prerequisite-Labs

- Check your laptop is connected to the internet
- Import Oracle Linux Container Engine ova file



- Start VM and log with next credentials

root
oracle

- Run command

```
# ip addr show
```

- check the ip virtualbox is assigning to the vm in the nic enp0s8
- Open a ssh session from your laptop to the oracle linux container engine vm in port 22

Ej.

```
# ssh root@192.168.56.102
```

- Proceed to install Oracle Linux containers (docker engine) from ssh
- Ping www.yahoo.com from the virtual machine console to verify the oracle linux container engine vm has internet access



LINUX

Docker Installation on Oracle Linux

```
# yum install docker-engine -y  
# systemctl enable docker  
# systemctl start docker
```

Familiarize with basic commands

```
# docker version  
# docker info  
# docker images  
# docker ps -l
```

DEPLOY AN APPLICATION SERVER

Download Oracle Linux Docker Image from Oracle Cloud

```
# docker pull oraclelinux  
# docker images
```

Start the Oracle Linux Container

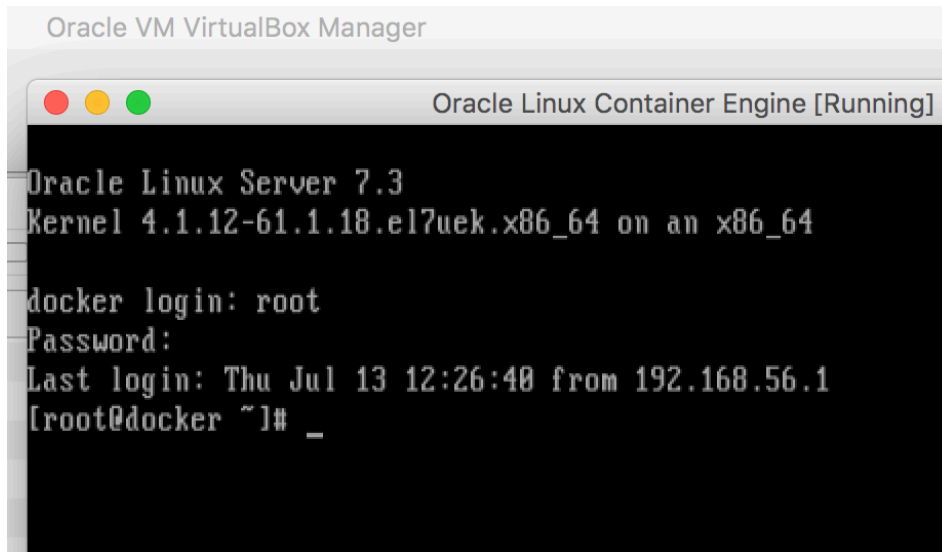
```
# docker run -it --name teclo oraclelinux bash
```

This line of command will create a container and will get you "inside" the container, you should see something like this.

A screenshot of a terminal window with a dark background. The first line shows the command `[[root@docker ~]# docker run -it --name att oraclelinux bash`. The second line shows the prompt `[root@73ee66330e24 /]#`, which is circled in black. To the right of the prompt, the text "You are inside the container" is displayed.

Container Location

If you want to find out where the container is located go to the **Oracle Linux container engine** vm, open the virtual machine console inside **virtualbox**

A screenshot of the Oracle VM VirtualBox Manager interface. The main window is titled 'Oracle VM VirtualBox Manager' and contains a sub-window titled 'Oracle Linux Container Engine [Running]'. The sub-window displays a terminal window with the following text: 'Oracle Linux Server 7.3', 'Kernel 4.1.12-61.1.18.el7uek.x86_64 on an x86_64', 'docker login: root', 'Password:', 'Last login: Thu Jul 13 12:26:40 from 192.168.56.1', and '[root@docker ~]# _'. The terminal window has a black background and white text.

```
Oracle VM VirtualBox Manager
Oracle Linux Container Engine [Running]
Oracle Linux Server 7.3
Kernel 4.1.12-61.1.18.el7uek.x86_64 on an x86_64
docker login: root
Password:
Last login: Thu Jul 13 12:26:40 from 192.168.56.1
[root@docker ~]# _
```

Type next command.

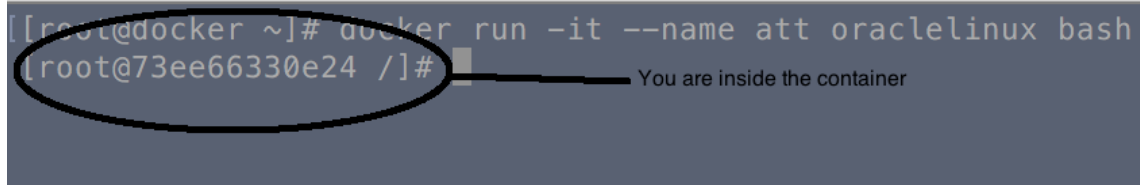
```
# cd /var/lib/docker/containers
# ls
```

Observe container listed.

Install Web Server packages

back to the container secure shell terminal session from your laptop client proceed to install a web server.

```
[[root@docker ~]# docker run -it --name att oraclelinux bash  
[root@73ee66330e24 /]#
```



```
# yum install -y httpd perl && yum clean all
```

Create content in Web Server

```
# echo "ejemplo web container teclo" > /var/www/html/index.html
```

Save Container changes

```
# exit  
# docker commit 73ee66330e24 teclo-web-server
```

73ee66330e24 (Container ID)

Check that the saved container appears listed as a new container image.

```
# docker images
```

Save as Tar File

```
# docker save -o teclo-web-server.tar teclo-web-server
```

stop & remove all exited containers

```
# docker stop $(docker ps -a -q)  
# docker rm $(docker ps -a -q)  
# docker ps -l
```



LINUX

Clone 3 Web Server Containers from Tar File

```
# docker rmi teclo-web-server
# docker images
# docker load -i teclo-web-server.tar
# docker images
```

Run the Container and observe the speed of deployment

```
# docker run -d --name web-server-1 -p 8080:80 teclo-web-server
/usr/sbin/httpd -D FOREGROUND
```

<http://192.168.56.102:8080>

Run another container instance from that image

```
# docker run -d --name web-server-2 -p 8081:80 teclo-web-server
/usr/sbin/httpd -D FOREGROUND
```

<http://192.168.56.102:8081>

And Another One

```
# docker run -d --name web-server-3 -p 8082:80 teclo-web-server
/usr/sbin/httpd -D FOREGROUND
```

<http://192.168.56.102:8082>

Stop and remove everything (images & containers)

```
# docker stop $(docker ps -a -q)
# docker rm $(docker ps -a -q)
# docker ps -l
# docker rmi teclo-web-server
# docker rmi oraclelinux
```

Check that every image was removed

```
# docker images
```



LINUX

CONTAINER AUTOMATION WITH DOCKERFILES

Create 3 web server containers automatically layer by layer from a dockerfile

```
# cd /dockerfiles/
```

Now watch all the previous deployment steps executed at once by means of a dockerfile

```
# docker build -t teclo-dockerfile -f httpd .  
# docker images
```

Now run again the web servers instances

```
# docker run -d --name teclo-from-dockerfile-1 -p 8080:80 teclo-  
dockerfile /usr/sbin/httpd -D FOREGROUND
```

<http://192.168.56.102:8080>

```
# docker run -d --name teclo-from-dockerfile-2 -p 8081:80 teclo-  
dockerfile /usr/sbin/httpd -D FOREGROUND
```

<http://192.168.56.102:8081>

```
# docker run -d --name teclo-from-dockerfile-3 -p 8082:80 teclo-  
dockerfile /usr/sbin/httpd -D FOREGROUND
```

<http://192.168.56.102:8082>



LINUX

Glassfish Dockerfile

```
# docker build -t teclo-dockerfile-glassfish -f glassfish .
```

Wait a few seconds until prompt is showed and list the latest images

```
# docker images
```

Run the glassfish container instance

```
# docker run -d -p 4848:4848 -p 8083:8083 -p 8181:8181 teclo-dockerfile-glassfish
```

Access the application server

<https://192.168.56.102:4848/>

admin
glassfish



LINUX

Now let's prepare the system to remove vulnerabilities from workloads with zero downtime.

open ksplICE inspector web and follow procedure

<http://www.ksplICE.com/inspector>

Install KsplICE in docker host

```
# yum install wget -y
# wget -N https://www.ksplICE.com/uptrack/install-uptrack
# sh install-uptrack dff244b8340a1ea09d46ba38a0346455b57d8241b3935e0db6e2c27a2fdeb9
```

Deploying WEBLOGIC from a Tar File

There is a previously load weblogic tar file loaded to avoid 3gb download

```
# cd
# docker load -i weblogic.tar
```

wait a few seconds, then

```
# docker images
```

Run Weblogic container instances

```
# docker run -d -p 49163:7001 -p 49164:7002 -p 49165:5556 att-weblogic:latest
# docker run -d -p 49166:7001 -p 49167:7002 -p 49168:5556 att-weblogic:latest
# docker run -d -p 49169:7001 -p 49170:7002 -p 49171:5556 att-weblogic:latest
```

Login to the weblogic instances

<http://192.168.56.102:49163/console>

<http://192.168.56.102:49166/console>

<http://192.168.56.102:49169/console>

User: weblogic
Pass: welcome1



LINUX

REMOVE VULNERABILITIES WITH ZERO DOWNTIME (KSPLICE)

First review the current kernel version

```
# uname -r
```

Now take action and remove the vulnerabilities

```
# /usr/sbin/uptrack-upgrade -y
```

Observe how the kernel was updated to the latest stable version with no need of reboots

```
# uname -r
```

Rollback vulnerabilities, bugs and errata with zero downtime

```
# /usr/sbin/uptrack-remove --all -y
```

From Zero To Hero with **Oracle Linux Containers and KSplice**

Thank You i i