Lheidar

Containerisation (Docker)

https://hub.docker.com/
(Repository of all Dockers)

Task 1 – Install Docker on your System

wget -qO- https://get.docker.com | sh
#runs command to install docker

docker run -p 8080:8080 -p 50000:50000 jenkins

docker run --name some-ghost -p 8079:2368 -d ghost

docker run --name my_solr -d -p 8078:8983 -t solr

docker rm<container>

docker rm 2fa456ae399c5431fa08700801b80d25e8481bc3ba685c6190c3423e8cd8fb53

docker kill <process name> e.g2fa456ae399c

TCP Rule to open/configure external ports for containers within firewall

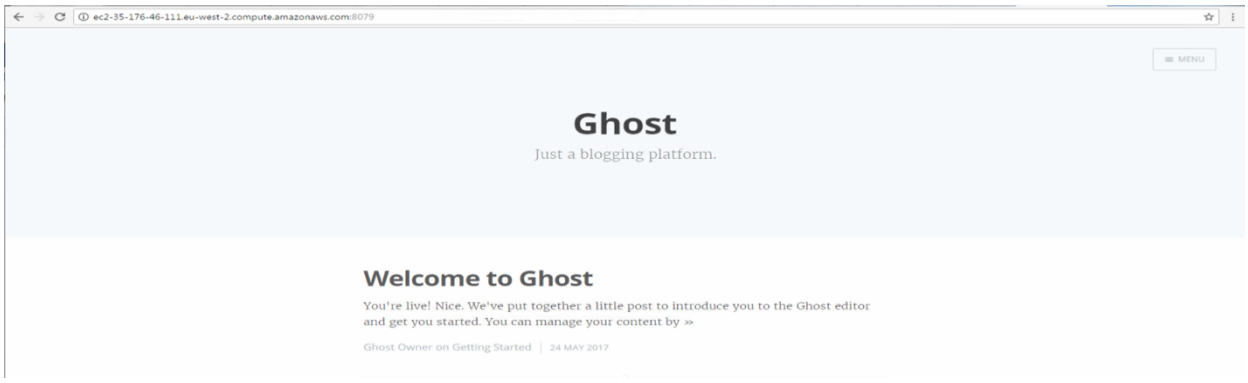
Ghost: 8079
Jenkins: 8080
Solr: 8078

Description | Inbound | Outbound | Tags

Edit

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom TCP Rule	TCP	8080	::/0
Custom TCP Rule	TCP	8078	0.0.0.0/0
Custom TCP Rule	TCP	8078	::/0
All traffic	All	All	sg-631c860a (default)
SSH	TCP	22	0.0.0.0/0
SSH	TCP	22	::/0
Custom TCP Rule	TCP	8079	0.0.0.0/0
Custom TCP Rule	TCP	8079	::/0

http://ec2-35-176-46-111.eu-west-2.compute.amazonaws.com:8079/

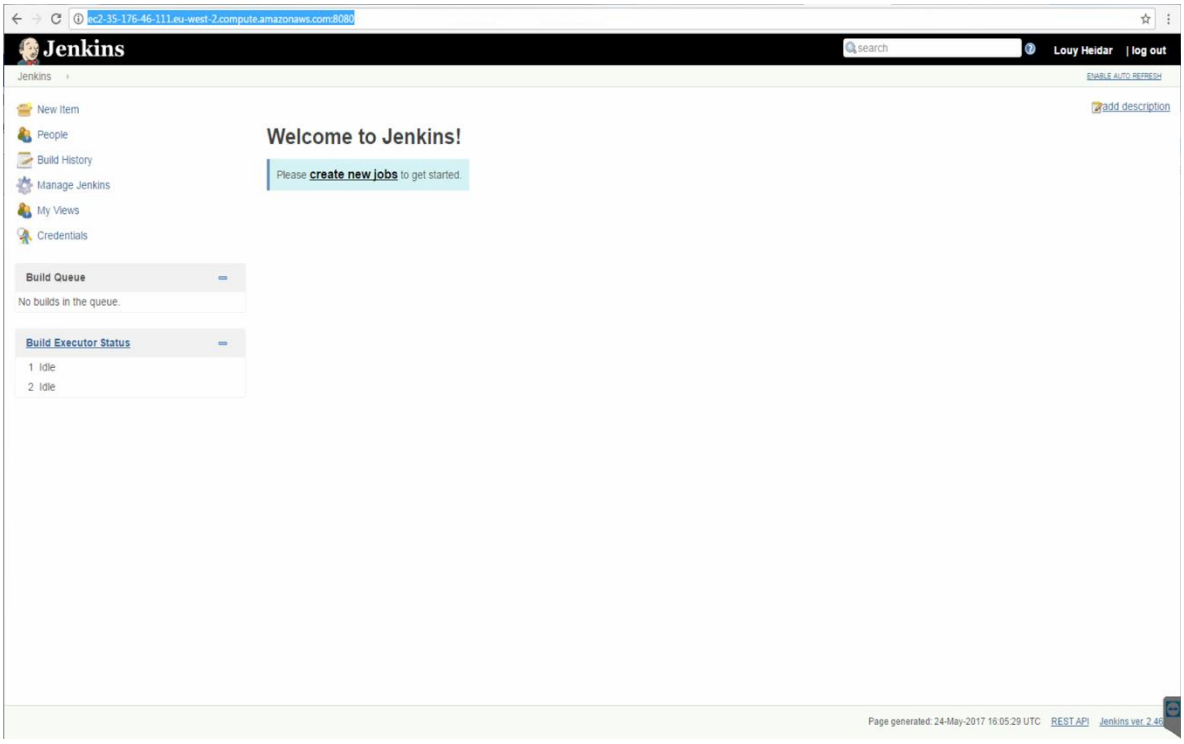


Lheidar

Task 2 Installed Jenkins Container

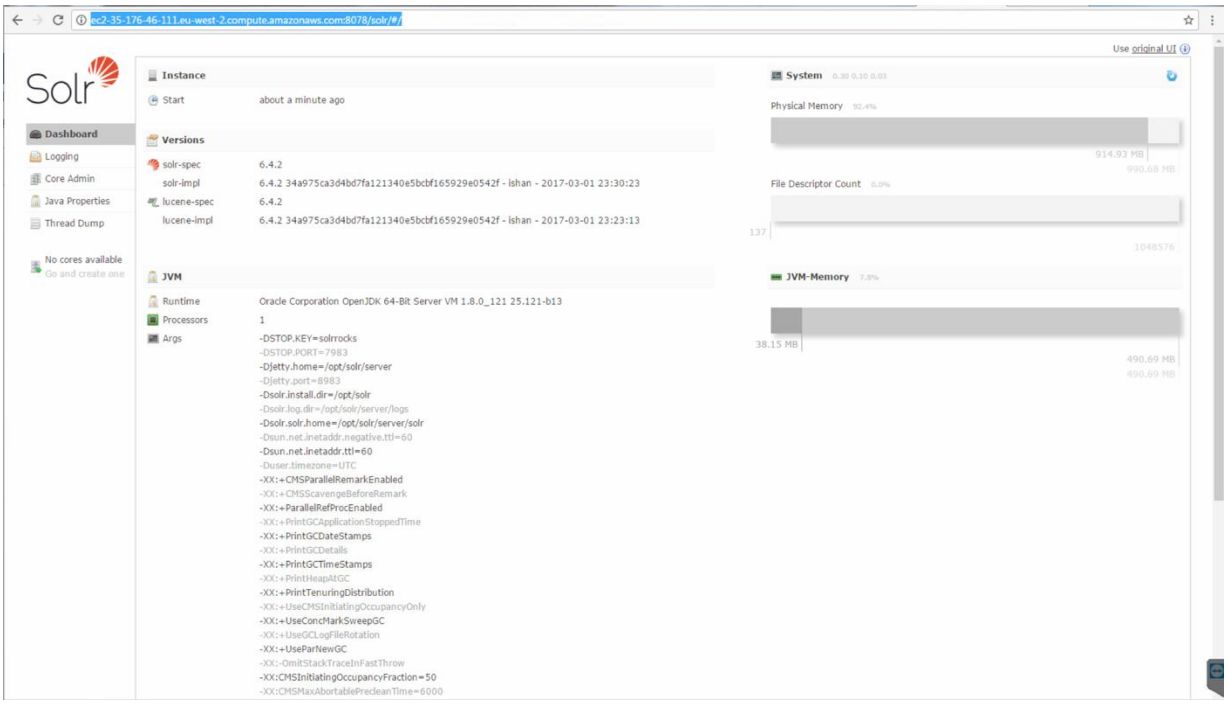
docker run -p 8080:8080 -p 50000:50000 jenkins

http://ec2-35-176-46-111.eu-west-2.compute.amazonaws.com:8080/



http://ec2-35-176-46-111.eu-west-2.compute.amazonaws.com:8078/

$ docker run --name my_solr -d -p 8078:8983 -t solr



Lheidar

Task 3 – Creating a Docker Environment

Starting with base image from Dockerfile

Docker build commands:

$docker build -t ubuntu:latest

-t is the Docker image tag

```
ubuntu@ip-172-31-10-242: ~/images
[1]+  Stopped                 nano Dockerfile
ubuntu@ip-172-31-10-242:~/images$ nano Dockerfile
ubuntu@ip-172-31-10-242:~/images$ nano Dockerfile
ubuntu@ip-172-31-10-242:~/images$ docker build -t ubuntu:latest .
Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
b6f892c0043b: Pull complete
55010f332b04: Pull complete
2955fb827c94: Pull complete
3deef3fcbd30: Pull complete
cf9722e506aa: Pull complete
Digest: sha256:382452f82a8bbd34443b2c727650af46aced0f94a44463c62a9848133ecb1aa8
Status: Downloaded newer image for ubuntu:latest
 ---> ebcd9d4fca80
Step 2/2 : MAINTAINER Louy Heidar (Louy.Heidar@QA.com)
 ---> Running in 5d29c8be6663
 ---> b9836216a8f8
Removing intermediate container 5d29c8be6663
Successfully built b9836216a8f8
Successfully tagged ubuntu:latest
ubuntu@ip-172-31-10-242:~/images$
```

List all images
$ docker images

Basic Principles:
A running image is a container

```
ubuntu@ip-172-31-10-242: ~/images
Step 2/2 : MAINTAINER Louy Heidar (Louy.Heidar@QA.com)
 ---> Running in 5d29c8be6663
 ---> b9836216a8f8
Removing intermediate container 5d29c8be6663
Successfully built b9836216a8f8
Successfully tagged ubuntu:latest
ubuntu@ip-172-31-10-242:~/images$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
ubuntu              latest              b9836216a8f8        6 minutes ago
118MB
ghost               latest              142165321ade        7 days ago
334MB
jenkins             latest              681ef98a247f        8 days ago
704MB
ubuntu              <none>              ebcd9d4fca80        9 days ago
118MB
kibana              latest              1dfc1de100f1        2 weeks ago
397MB
solr                latest              1a1515505797        4 weeks ago
514MB
ubuntu@ip-172-31-10-242:~/images$
```

Lheidar

#Lists the available Docker images
sudo docker ps

#interacts with the terminal inside ubuntu container
sudo docker run –ti ubuntu bash

Keyboard Shortcut:

Ctrl PQ

#switching to the AWS control terminal

Dockers file for installing Java:

```
ubuntu@ip-172-31-10-242: ~/images
  GNU nano 2.5.3                    File: Dockerfile

FROM ubuntu:14.04

#Always update your running system
RUN sudo apt-get update -y

#You may or may not need to run these commands
RUN sudo apt-get install -y wget
RUN sudo apt-get install -y tar

#installs the libraries needed to run the GUI
RUN sudo apt-get install -y libgtk2.0
RUN sudo apt-get install -y mesa-utils
RUN sudo apt-get install -y libXtst6

#RUN sudo apt-get install -y openjdk-7-jre
#RUN java -version

#Now install the Java Compiler
#RUN sudo apt-get install -y openjdk-7-jdk
#RUN javac -version
#Add java from file and install
WORKDIR /opt

ADD files /opt

RUN sudo tar zxvf /opt/java.tar.gz

RUN sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_74/bin/java 100
RUN sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_74/bin/javac 100

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   ^Y Prev Page   M-\ First Line
^X Exit       ^R Read File   ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line ^V Next Page   M-/ Last Line
```

Java Version Build:

$ java -version

```
root@86ca48ed7e37: /
root@1e14528eab4c:/# java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
root@1e14528eab4c:/# clear
root@1e14528eab4c:/# ubuntu@ip-172-31-10-242:~/images$ sudo docker run -ti ubuntu bash
root@86ca48ed7e37:/# java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
root@86ca48ed7e37:/#
```

Lheidar

Useful Commands

docker pull mysql
rm -rflampp

Deletes all files and folders contained in the directory.

Encountered Issue:
Many containers have created with <none> tag, research required for reason

```
ubuntu@ip-172-31-10-242: ~/images/mysqldata
-pw -d mysql:tag^C
root@bbb6ba71cfc7:/datadir# pwd
/datadir
root@bbb6ba71cfc7:/datadir# docker run --name some-mysql -v /datadir:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=my-secret
-pw -d mysql:tagubuntu@ip-172-31-10-242:~/images/mysqldata$ sudo dockers ps
sudo: dockers: command not found
ubuntu@ip-172-31-10-242:~/images/mysqldata$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
                    NAMES
bbb6ba71cfc7        ubuntu              "bash"                   9 minutes ago       Up 9 minutes
                    affectionate_almeida
a6cd2801eb9c        ubuntu              "bash"                   13 minutes ago      Up 13 minutes
                    reverent_yonath
86ca48ed7e37        ubuntu              "bash"                   About an hour ago   Up About an hour
                    eager_turing
1e14528eab4c        ubuntu              "bash"                   About an hour ago   Up About an hour
                    affectionate_banach
fee4bb009a95        ubuntu              "bash"                   About an hour ago   Up About an hour
                    zen_hugle
1b2952f2c1d5        ubuntu              "bash"                   About an hour ago   Up About an hour
                    romantic_goldwasser
469ad08aa3b0        ubuntu              "bash"                   2 hours ago         Up 2 hours
                    relaxed_davinci
15a13358146a        ubuntu              "bash"                   2 hours ago         Up 2 hours
                    wonderful_jackson
df5b72a1ffc3        solr                "docker-entrypoint..."   20 hours ago        Up 20 hours         0.0.0.0:807
8->8983/tcp   my_solr
d1a30c3f0172        ghost               "docker-entrypoint..."   24 hours ago        Up 24 hours         0.0.0.0:807
9->2368/tcp   some-ghost
ubuntu@ip-172-31-10-242:~/images/mysqldata$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              48c0691af282        About an hour ago   667MB
<none>              <none>              abddcc51619a        3 hours ago         870MB
ubuntu              latest              e3a062e1c46f        3 hours ago         867MB
<none>              <none>              6f147ecb2b41        3 hours ago         118MB
ghost               latest              142165321ade        7 days ago          334MB
jenkins             latest              681ef98a247f        8 days ago          704MB
openjdk             latest              ab0ecda9094c        8 days ago          610MB
ubuntu              <none>              ebcd9d4fca80        10 days ago         118MB
ubuntu              14.04               2ff3b426bbaa        10 days ago         188MB
mysql               latest              e799c7f9ae9c        2 weeks ago         407MB
kibana              latest              1dfc1de100f1        2 weeks ago         397MB
solr                latest              1a1515505797        4 weeks ago         514MB
ubuntu@ip-172-31-10-242:~/images/mysqldata$
```

Work Around:

<none>:<none> are known as intermediate tags

dockerrmi $(docker images -f "dangling=true" -q)

The above command can be used to clean up dangling images since Docker doesn't have an automatic garbage collection system
#Remove containers by image ID
dockerrmi<image-id>

Lheidar

#Display all containers
$ docker images

```
^Cubuntu@ip-172-31-10-242:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              e3a062e1c46f        20 hours ago        867MB
ghost               latest              142165321ade        8 days ago          334MB
jenkins             latest              681ef98a247f        9 days ago          704MB
ubuntu              14.04               2ff3b426bbaa        10 days ago         188MB
mysql               latest              e799c7f9ae9c        2 weeks ago         407MB
kibana              latest              1dfc1de100f1        2 weeks ago         397MB
solr                latest              1a1515505797        4 weeks ago         514MB
```

dockerps
#Display all running containers

dockerps–a

Task 4 – Createown linked container
Approach1 – to be continued

#Created new directory
Sudomkdirmysqldata

Reason: To create a data directory on the host system (outside the container) and to mount this to a directory visible from inside the container.

Run the container for my MySQL mounts new volume (–v). Also sets and the directory locations specified in addition MySQL database properties such as root password for the associated container.

docker run --name some-mysql -v /home/ubuntu/images/mysqldata:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=testpassword -d mysql:latest

$ dockerps–a
#Shows only running container by default

```
ubuntu@ip-172-31-10-242: ~/images/mysqldata
ubuntu@ip-172-31-10-242:~/images/mysqldata$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
                    PORTS               NAMES
8ee506c6e2a9        mysql:latest        "docker-entrypoint..."   About a minute ago  Up About a m
ute                 3306/tcp            some-mysql
edfc3a7e16cf        ubuntu              "bash"                   25 minutes ago      Up 25 minute
                                        musing_shaw
f8195ec9830c        jenkins             "/bin/tini -- /usr..."   37 minutes ago      Exited (130)
2 minutes ago                           hopeful_ritchie
8f9e68f3ac1e        jenkins             "/bin/tini -- /usr..."   40 minutes ago      Exited (130)
8 minutes ago                           gifted_williams
ad76077f9630        jenkins             "/bin/tini -- /usr..."   About an hour ago   Exited (130)
bout an hour ago                        silly_heyrovsky
d2f65bb28a8f        jenkins             "/bin/tini -- /usr..."   About an hour ago   Exited (130)
bout an hour ago                        romantic_edison
be8b17bb230f        ubuntu              "bash"                   About an hour ago   Up About an
ur                                      mystifying_kilby
ubuntu@ip-172-31-10-242:~/images/mysqldata$
```

Mysql:latest conainer has been created and running in background

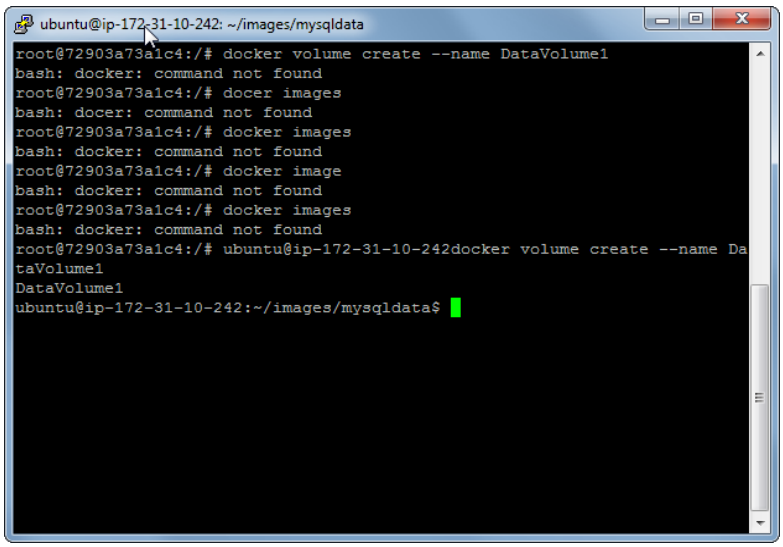
Lheidar

Creating File to be shared between containers

sudo bash -c 'echo "hello" > 'Example1.txt'

Approach 2:
Share Data between Docker Container by creating an Independent Volume

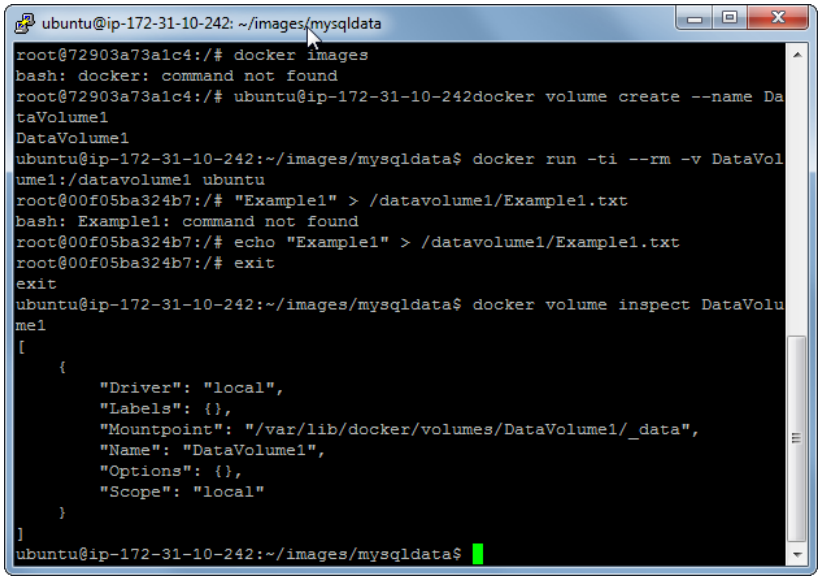
1) Command to add a volume named DataVolume1:
docker volume create --name DataVolume1



Volume name is displayed, showing that command was successful.

Command:
docker run -ti --rm -v DataVolume1:/datavolume1 ubuntu

2) Verifying that the volume is present on the system with docker volume inspect



Lheidar

3) Running a new container and attaching DataVolume1

Commands:

suubuntu

docker run --name DataVolume2 --rm -ti -v DataVolume1:/datavolume1 ubuntu

docker ps

```
ubuntu@ip-172-31-10-242: ~/images/mysqldata
ddocker: command not found
ubuntu@ip-172-31-10-242:~/images/mysqldata$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
      STATUS              PORTS               NAMES
1ced4a85fd48        ubuntu              "/bin/bash"         50 seconds ago
      Up 50 seconds                           DataVolume2
72903a73a1c4        ubuntu              "/bin/bash"         38 minutes ago
      Up 38 minutes                           elegant_golick
ubuntu@ip-172-31-10-242:~/images/mysqldata$
```

Listing the processes with a different process ID for each container
Starting a new container and attach DataVolume1:

```
root@1e6b20e4d0f6: /
1ced4a85fd48        ubuntu              "/bin/bash"         18 minutes ago      Up 18 minutes
                         DataVolume2
72903a73a1c4        ubuntu              "/bin/bash"         About an hour ago   Up About an hou
r                        elegant_golick
ubuntu@ip-172-31-10-242:/root$ cat /datavolume1/Example1.txt
cat: /datavolume1/Example1.txt: No such file or directory
ubuntu@ip-172-31-10-242:/root$ docker run --rm -ti -v DataVolume1:/datavolume1 ubuntu
root@1e6b20e4d0f6:/# cat /datavolume1/Example1.txt
Example1
root@1e6b20e4d0f6:/# cat /datavolume1/Example1.txt
```

The above steps demonstrate a created new volume, attaching it to a container, and then verifying its persistence via the create new Example1 file.

Approach 3:
Creating own linked container using a shared folder location

1) Creating directory to store the data in host volume i.e location which can be accessed by both via php and mysql containers.

sudo mkdir sharedvolume

```
ubuntu@ip-172-31-9-56:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-9-56:~$ sudo mkdir sharedvolume
ubuntu@ip-172-31-9-56:~$ ls
sharedvolume
```

Lheidar

2) Creating and running the mysql container with the associated path to the shared volume location which sets the mysql root password and the version number for associated container.

sudo docker run –name mysql –v /home/ubuntu/sharedvolume:/var/lib/shared –e MYSQL_ROOT_PASSWORD=testpass –d mysql:latest

```
ubuntu@ip-172-31-9-56:~$ sudo docker run --name mysql2 -v /home/ubuntu/sharedvol
ume:/var/lib/shared -e MYSQL_ROOT_PASSWORD=testpass -d mysql:latest
5b326be092075c0c9d3ec11b02439f17898516d5c5398735aabeb2c6fcd5d670
ubuntu@ip-172-31-9-56:~$ pwd
/home/ubuntu
```

3) 3) Runs/checks the current processes for created containers

Sudo docker ps

```
root@c08e774c49a7:/var/lib# ubuntu@ip-172-31-9-56:~/sharedvolume$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
     STATUS              PORTS               NAMES
5b326be09207        mysql:latest        "docker-entrypoint..."   13 minutes ago
     Up 13 minutes       3306/tcp            mysql2
c08e774c49a7        php:latest          "docker-php-entryp..."   35 minutes ago
     Up 35 minutes                           sleepy_brattain
5897fb9222da        mysql:latest        "docker-entrypoint..."   38 minutes ago
     Up 38 minutes       3306/tcp            myupdatedsql
5695f6b3ee5e        mysql:latest        "docker-entrypoint..."   44 minutes ago
     Up 44 minutes       3306/tcp            some-mysql
b1ec04a68140        php                 "docker-php-entryp..."   About an hour a
go   Up About an hour                        jovial_booth
278cf18d1f7c        ubuntu              "/bin/bash"              About an hour a
go   Up About an hour                        frosty_goldwasser
```

Executing the interactive terminal inside a container using exec and containerID (c08e774c49a7)

```
ubuntu@ip-172-31-9-56:~/sharedvolume$ sudo docker exec -ti c08e774c49a7 bash
root@c08e774c49a7:/# ls
bin   datavolume1  etc   lib    media  opt   root  sbin  sys  usr
boot  dev          home  lib64  mnt    proc  run   srv   tmp  var
```

4) Running the php container and mounting a directory to new directory in php file path

```
ubuntu@ip-172-31-9-56:~/sharedvolume$ sudo docker run --name phpupdatev2 -ti -v /home/ubuntu/sharedvolume:/var/lib/shared  php:
ash
root@a140bbfa4af0:/# cd var lib/shared
root@a140bbfa4af0:/var# ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp
root@a140bbfa4af0:/var# cd lib
root@a140bbfa4af0:/var/lib# ls
apt  dpkg  initscripts  insserv  misc  pam  shared  systemd  update-rc.d  urandom
root@a140bbfa4af0:/var/lib# cd shared
root@a140bbfa4af0:/var/lib/shared# ls
Sharedfile.txt
```

Lheidar

Task 5 – Create your own docker-compose file

Commands used:

```
# Install docker
wget -qO- https://get.docker.com | sh

#Creating the .yml file to execute the containers as separate processes
sudo touch docker-compose.yml

#Running the docker compose file
sudo docker-compose up -d
```

Problem encountered:

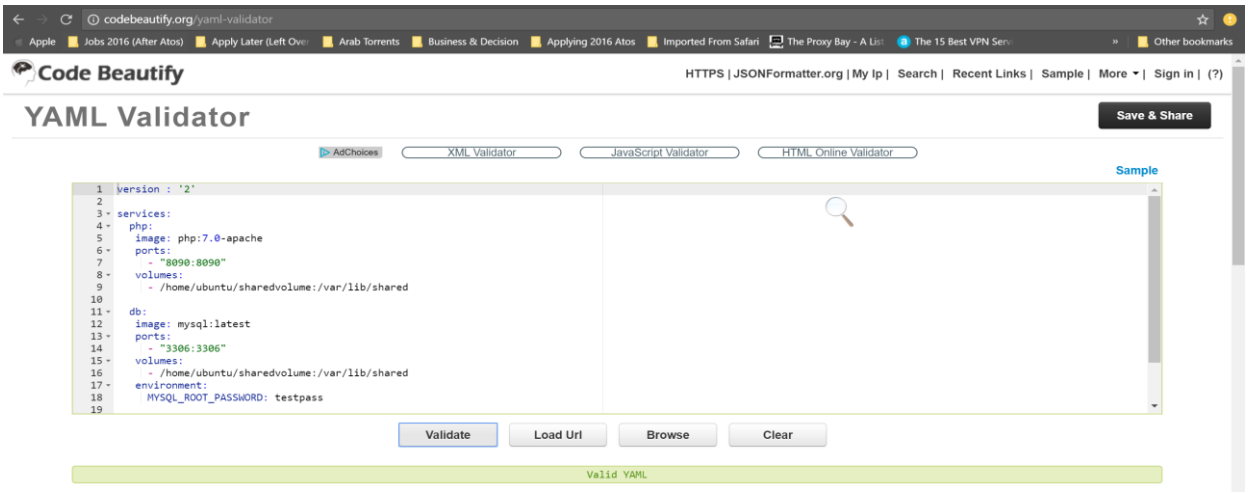
PHP container was not running and displaying as a separate process after executing the docker-compose.yml file. Only the mysql container displayed a process ID.

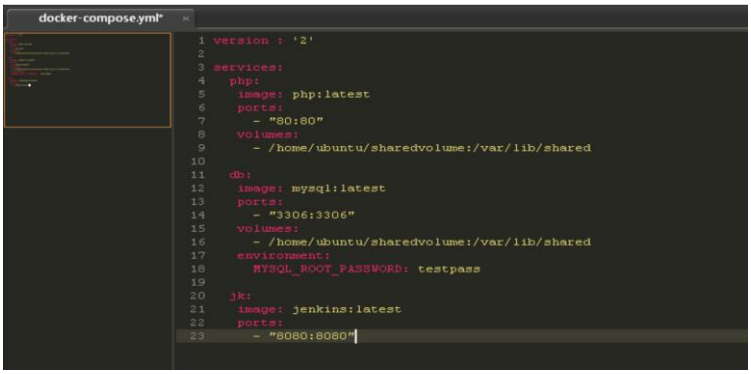
Troubleshooting & Solution:

1) Added another separate docker container (Jenkins) to the 'docker-compose.yml' file to test and see if that container was running successfully and displayed a separate process.

Result: Jenkins container successfully displayed (Process ID: b37283936262)

2) Online YML Validator was used to troubleshoot any code related parsing issues. Correct spacing and indentation was very critical, any other format would result in parsing errors and unsuccessful executions





3) Sublime text (Text editor) used to troubleshoot ensure that spaces have been removes and file is formatted to parse and successfully run

Lheidar

4) docker-compose.yml

Php:7.0-apache
mysql:latest
Jenkins:latest

Above Containers set to run once the .yml file is excuted.

#Running the docker compose file
sudo docker-compose up -d

```
ubuntu@ip-172-31-9-56: ~/sharedvolume
  GNU nano 2.5.3              File: docker-compose.yml

version : '2'

services:
  php:
   image: php:7.0-apache
   ports:
     - "8090:8090"
   volumes:
     - /home/ubuntu/sharedvolume:/var/lib/shared

  db:
   image: mysql:latest
   ports:
     - "3306:3306"
   volumes:
     - /home/ubuntu/sharedvolume:/var/lib/shared
   environment:
     MYSQL_ROOT_PASSWORD: testpass

  jk:
   image: jenkins:latest
   ports:
                                  [ Read 23 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

#List all the containers as separate processes
#All 3 containers are running successfully

sudo docker ps

```
ubuntu@ip-172-31-9-56:~/sharedvolume$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
4e984dfdb65e        php:7.0-apache      "docker-php-entryp..."   About a minute ago
b37283936262        jenkins:latest      "/bin/tini -- /usr..."   40 minutes ago
1bd8e9a10e3d        mysql:latest        "docker-entrypoint..."   About an hour ago
```