TASK BASED ON DOCKER MASTERCLASS

#TASK2.1: *The compose should deploy two services (web and DB), and each service should deploy a container as per details below:*

For web service: --->> php:rc-apache

a. Container name must be php_web.

b. Use image php with any apache tag. Check here for more details https://hub.docker.com/_/php?tab=tags.

c. Map php_web container's port 80 with host port 6000 d. Map php_web container's /var/www/html volume with host volume /var/www/html.

For DB service:

a. Container name must be mysql_web.

b. Use image mariadb with any tag (preferably latest). Check here for more details https://hub.docker.com/_/mariadb?tab=tags.

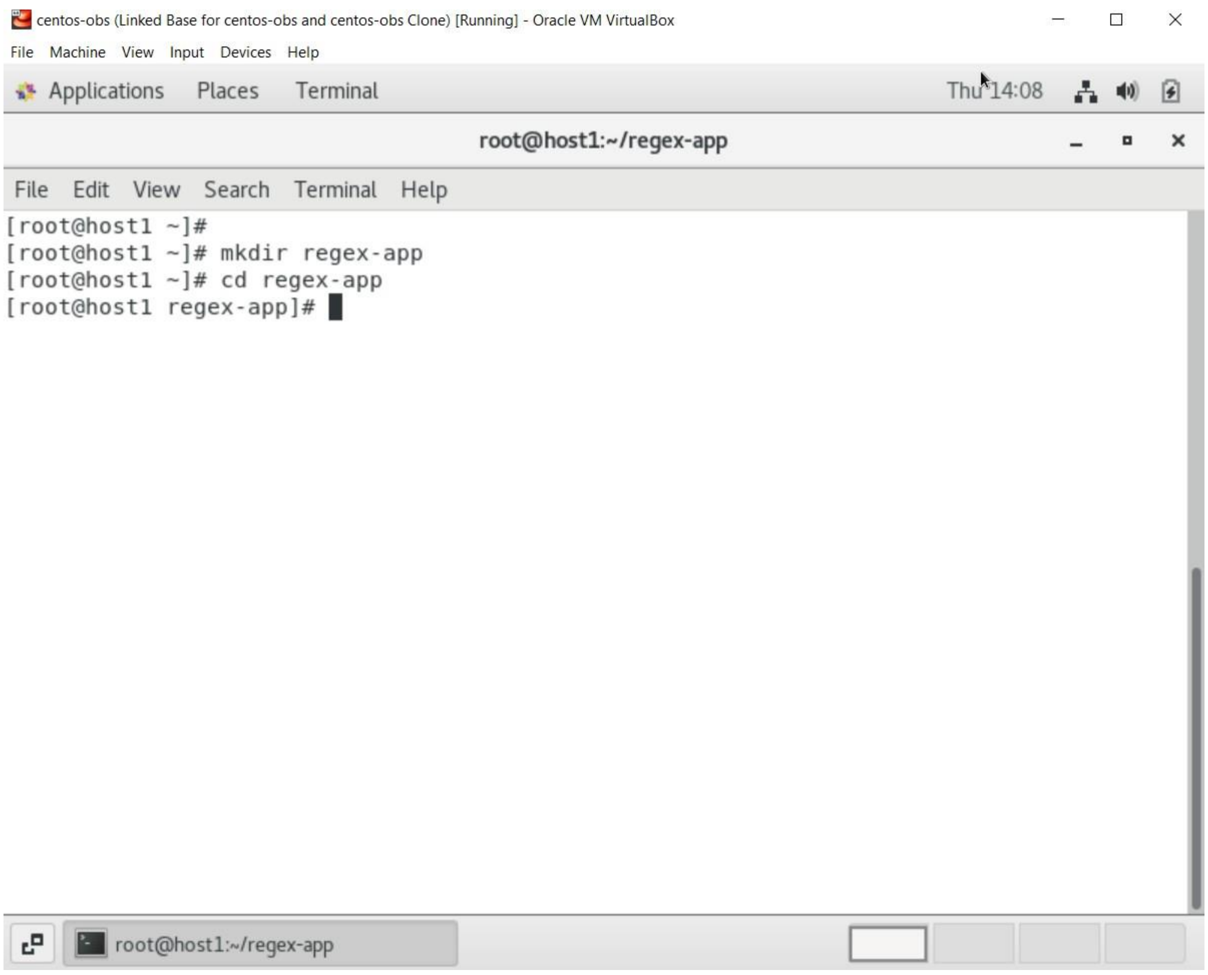
c. Map mysql_web container's port 3306 with host port 3306

d. Map mysql_web container's /var/lib/mysql volume with host volume /var/lib/mysql.

e. Set MYSQL_DATABASE=database_web and use any custom user (except root) with some complex password for DB connections. After running docker-compose up you can access the app with curl command curl <server-ip or hostname>:6000/

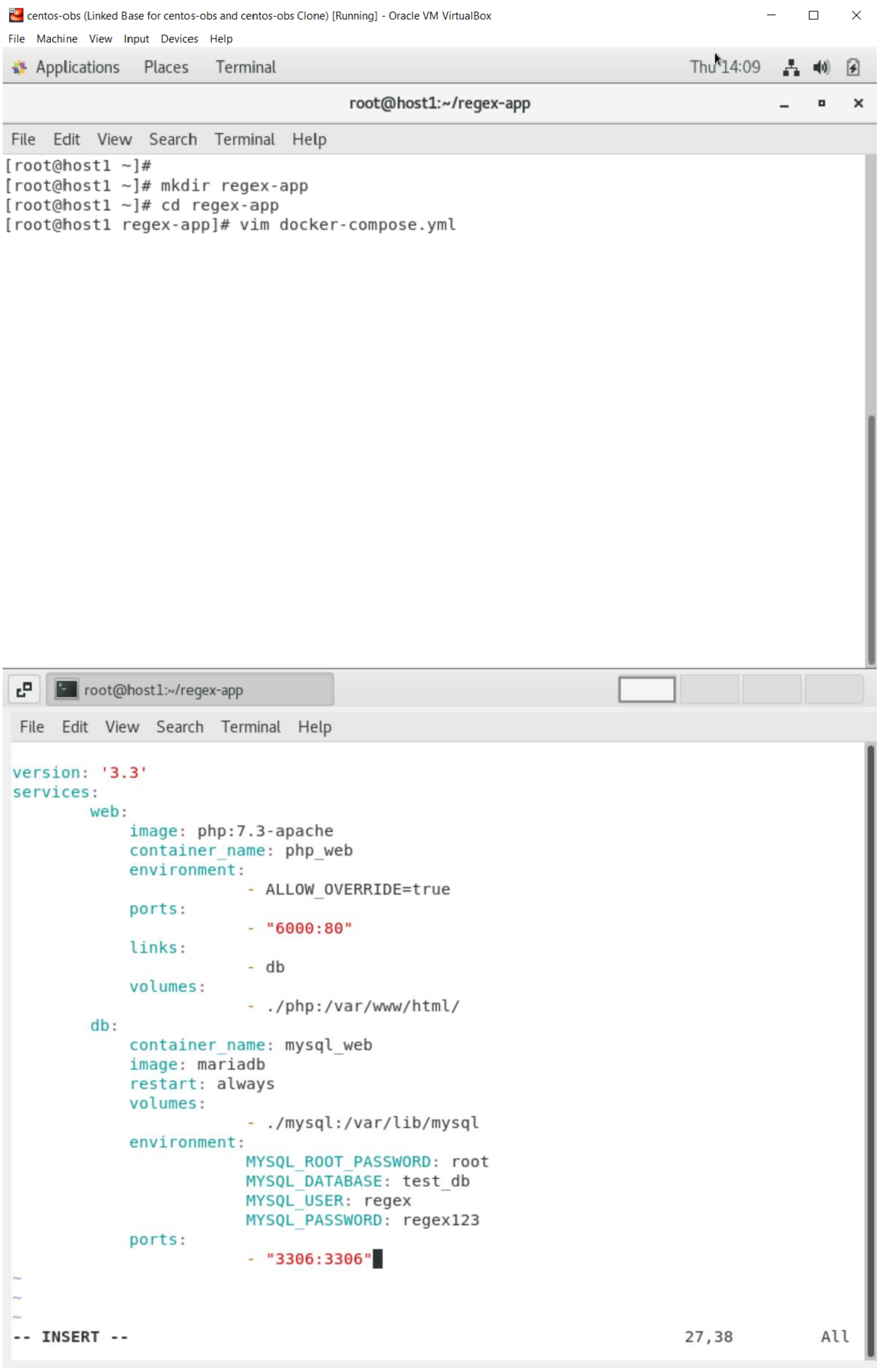
SOLUTION:

Step 1: Create a directory using "mkdir" command. Get into that directory using "cd" command


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Thu 14:08
root@host1:~/regex-app
File Edit View Search Terminal Help
root@host1:~/regex-app

```
[root@host1 ~]#
[root@host1 ~]# mkdir regex-app
[root@host1 ~]# cd regex-app
[root@host1 regex-app]#
```

Step 2: Create docker-compose.yml file using "vim" command. Edit the docker-compose file, *first section* to define will be the web portion of the stack and *next section* defines the database.

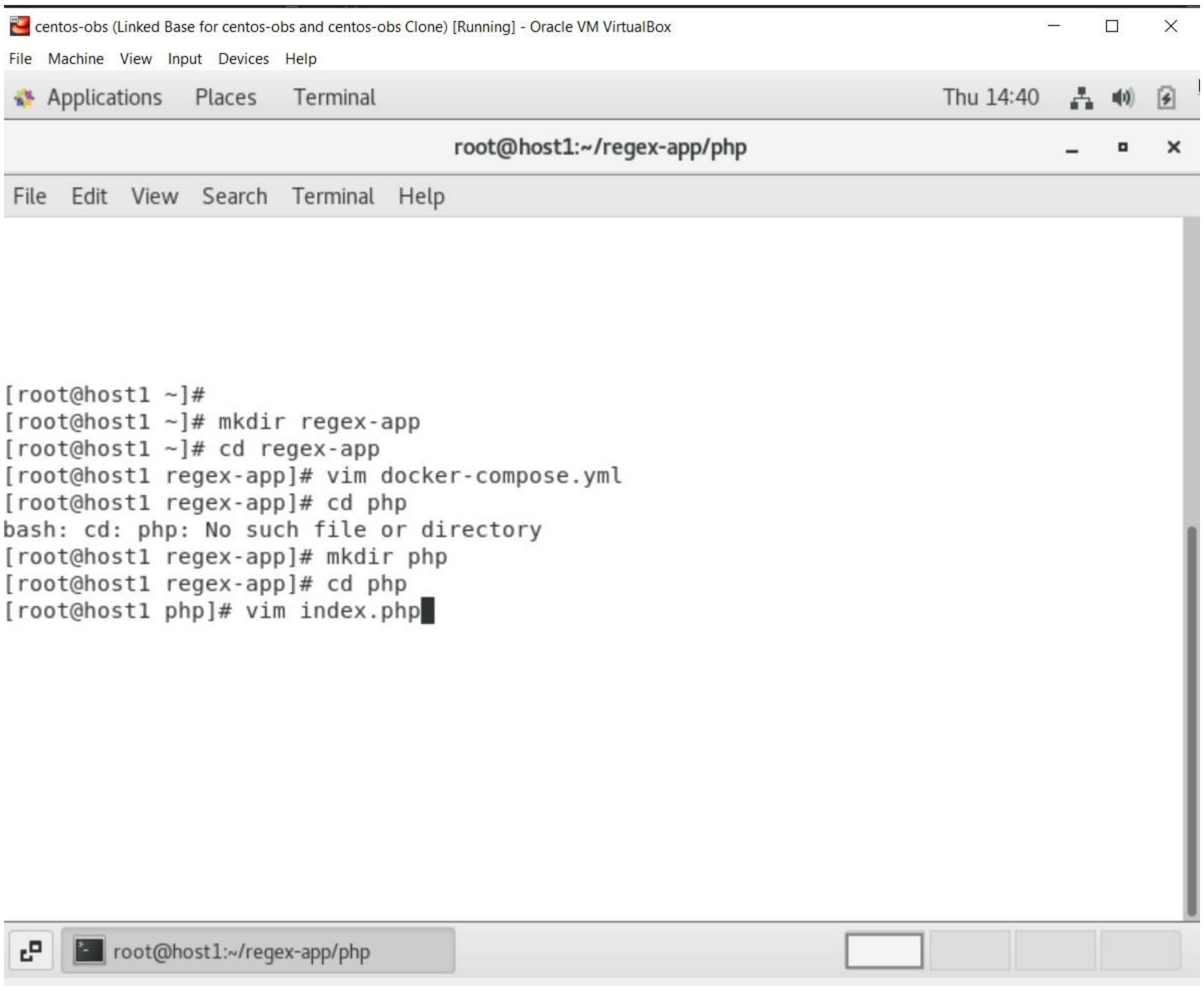

centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Thu 14:09
root@host1:~/regex-app
File Edit View Search Terminal Help

```
[root@host1 ~]#
[root@host1 ~]# mkdir regex-app
[root@host1 ~]# cd regex-app
[root@host1 regex-app]# vim docker-compose.yml
```

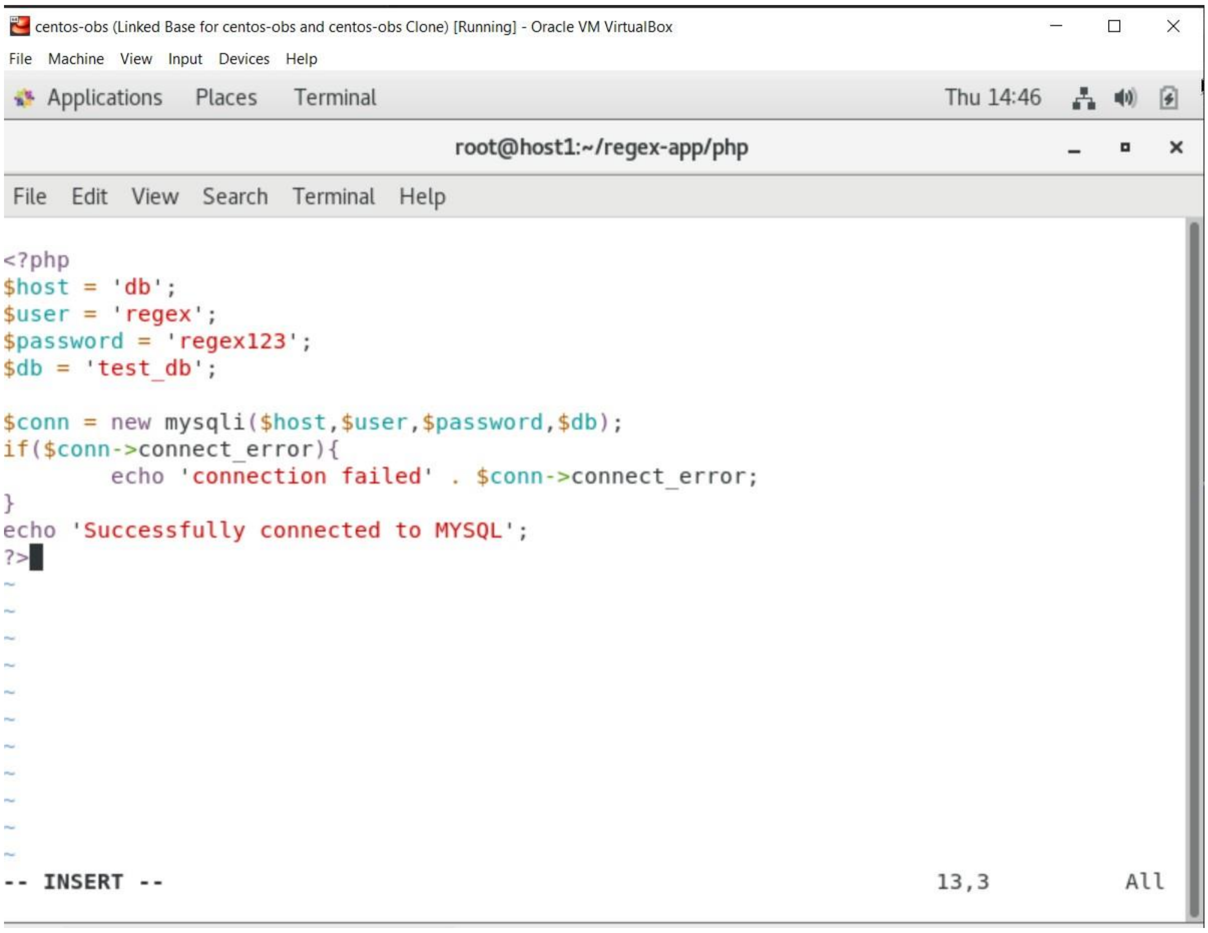
root@host1:~/regex-app
File Edit View Search Terminal Help

```
version: '3.3'
services:
        web:
            image: php:7.3-apache
            container_name: php_web
            environment:
                        - ALLOW_OVERRIDE=true
            ports:
                        - "6000:80"
            links:
                        - db
            volumes:
                        - ./php:/var/www/html/
        db:
            container_name: mysql_web
            image: mariadb
            restart: always
            volumes:
                        - ./mysql:/var/lib/mysql
            environment:
                        MYSQL_ROOT_PASSWORD: root
                        MYSQL_DATABASE: test_db
                        MYSQL_USER: regex
                        MYSQL_PASSWORD: regex123
            ports:
                        - "3306:3306"
~
~
~
-- INSERT --                                                          27,38         All
```

Step 3: Create a directory named "php" under the previous directory. Create index.php file php directory. Edit index.php and add credentials for mysql access.

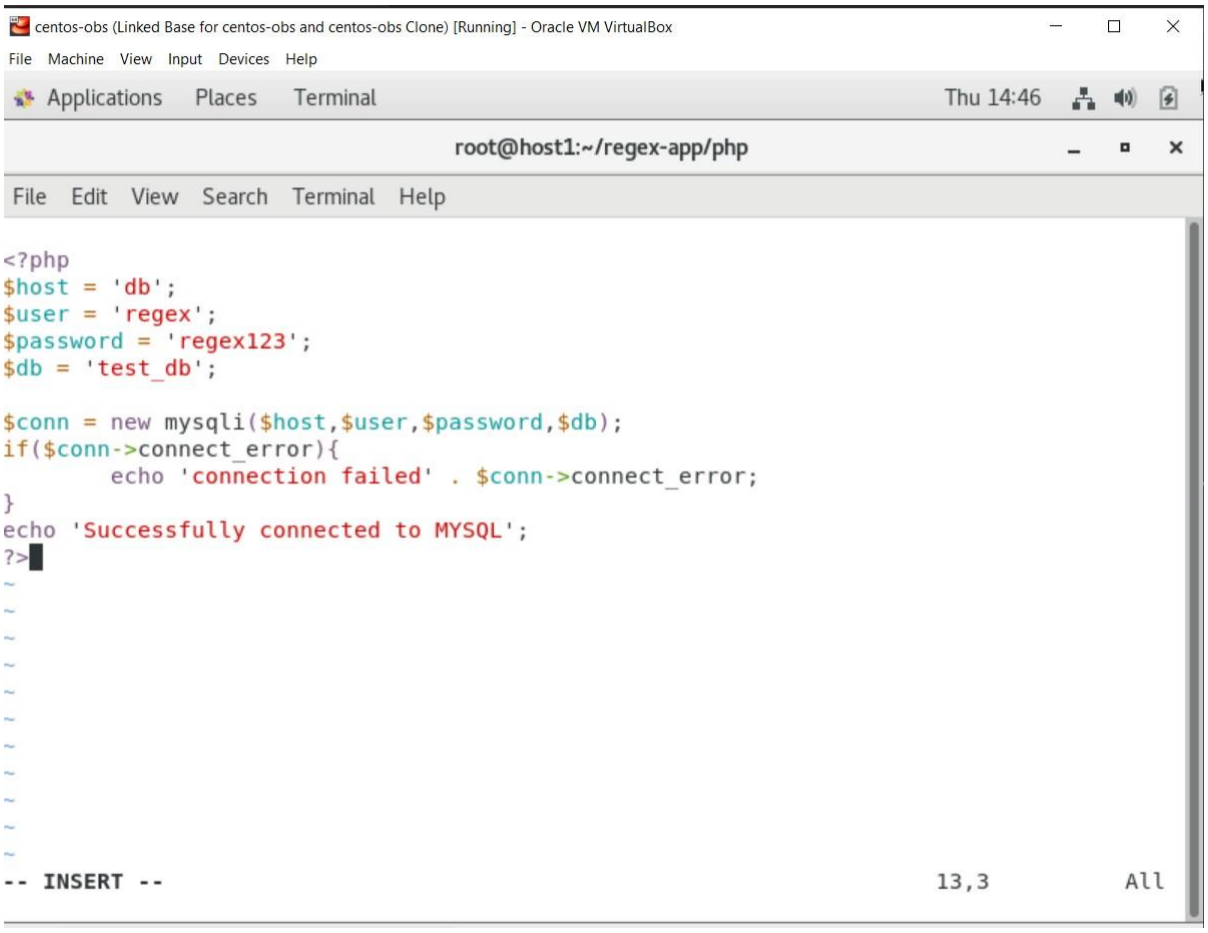

centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Thu 14:40
root@host1:~/regex-app/php
File Edit View Search Terminal Help
root@host1:~/regex-app/php

```
[root@host1 ~]#
[root@host1 ~]# mkdir regex-app
[root@host1 ~]# cd regex-app
[root@host1 regex-app]# vim docker-compose.yml
[root@host1 regex-app]# cd php
bash: cd: php: No such file or directory
[root@host1 regex-app]# mkdir php
[root@host1 regex-app]# cd php
[root@host1 php]# vim index.php
```


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Thu 14:46
root@host1:~/regex-app/php
File Edit View Search Terminal Help

```
<?php
$host = 'db';
$user = 'regex';
$password = 'regex123';
$db = 'test_db';

$conn = new mysqli($host,$user,$password,$db);
if($conn->connect_error){
        echo 'connection failed' . $conn->connect_error;
}
echo 'Successfully connected to MYSQL';
?>
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                13,3          All
```

Step 4: Create a Dockerfile using "vim" command. Edit that file put specific keywords that dictate how to build a specific image.

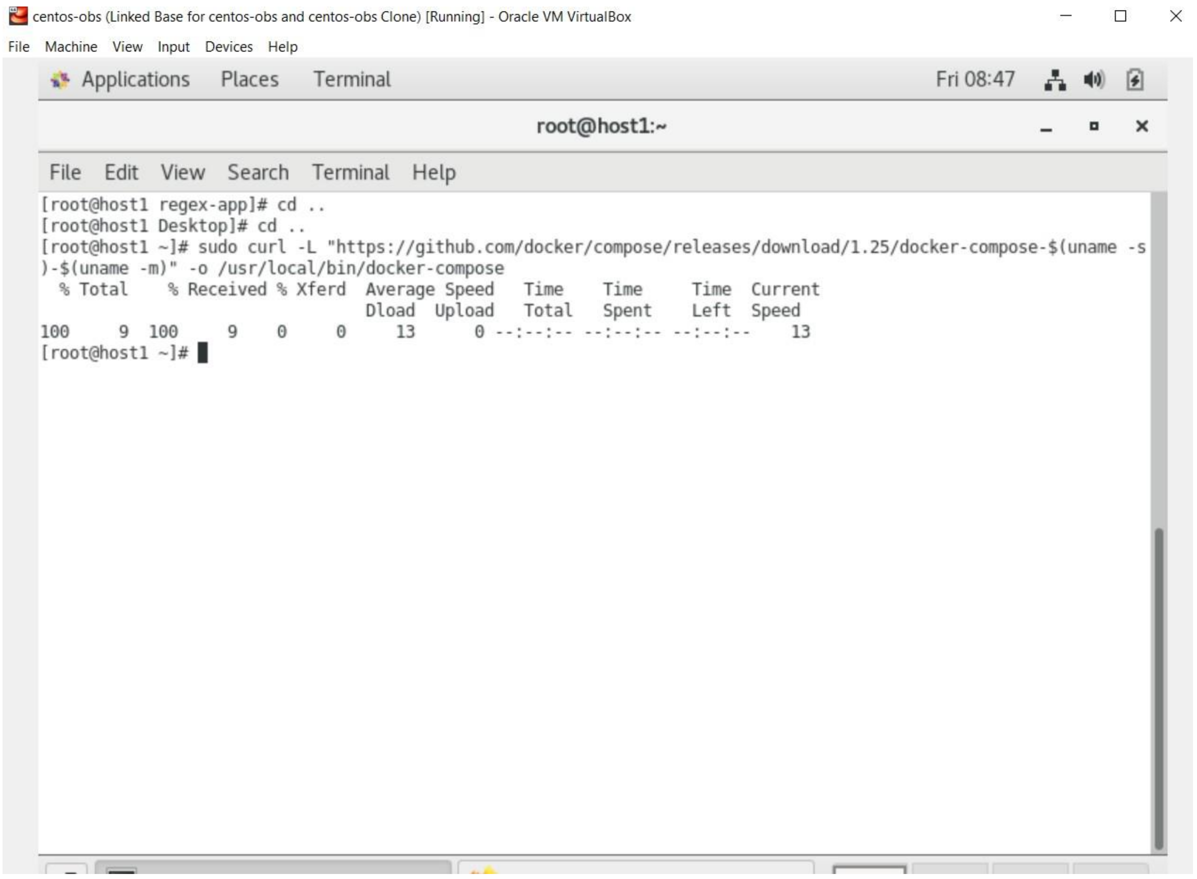

centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Thu 14:46
root@host1:~/regex-app/php
File Edit View Search Terminal Help

```
<?php
$host = 'db';
$user = 'regex';
$password = 'regex123';
$db = 'test_db';

$conn = new mysqli($host,$user,$password,$db);
if($conn->connect_error){
        echo 'connection failed' . $conn->connect_error;
}
echo 'Successfully connected to MYSQL';
?>
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                              13,3          All
```

Step 4: Create a Dockerfile using "vim" command. Edit that file put specific keywords that dictate how to build a specific image.

Step 5: Install Docker compose.

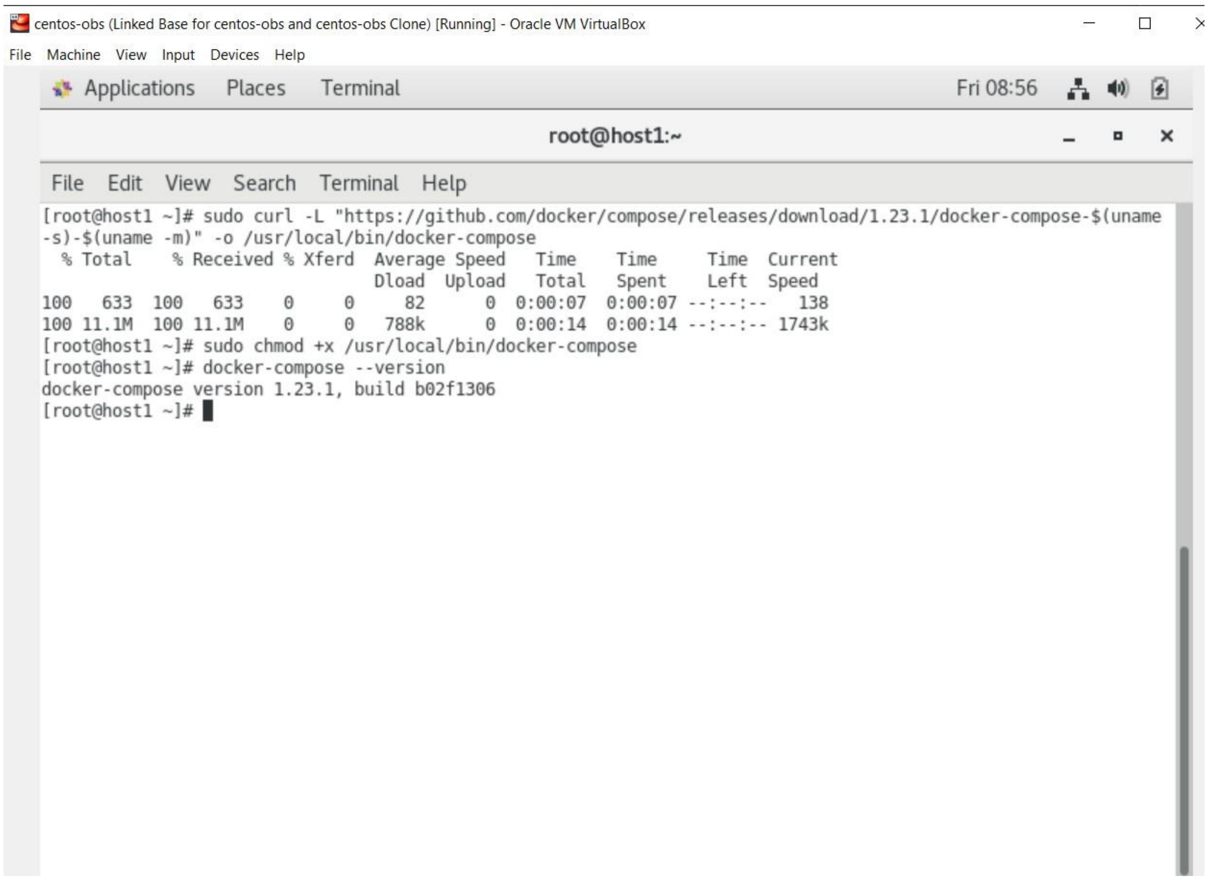
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 08:47
root@host1:~
File Edit View Search Terminal Help

```
[root@host1 regex-app]# cd ..
[root@host1 Desktop]# cd ..
[root@host1 ~]# sudo curl -L "https://github.com/docker/compose/releases/download/1.25/docker-compose-$(uname -s
)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100     9  100     9    0     0     13      0 --:--:-- --:--:-- --:--:--    13
[root@host1 ~]# 
```

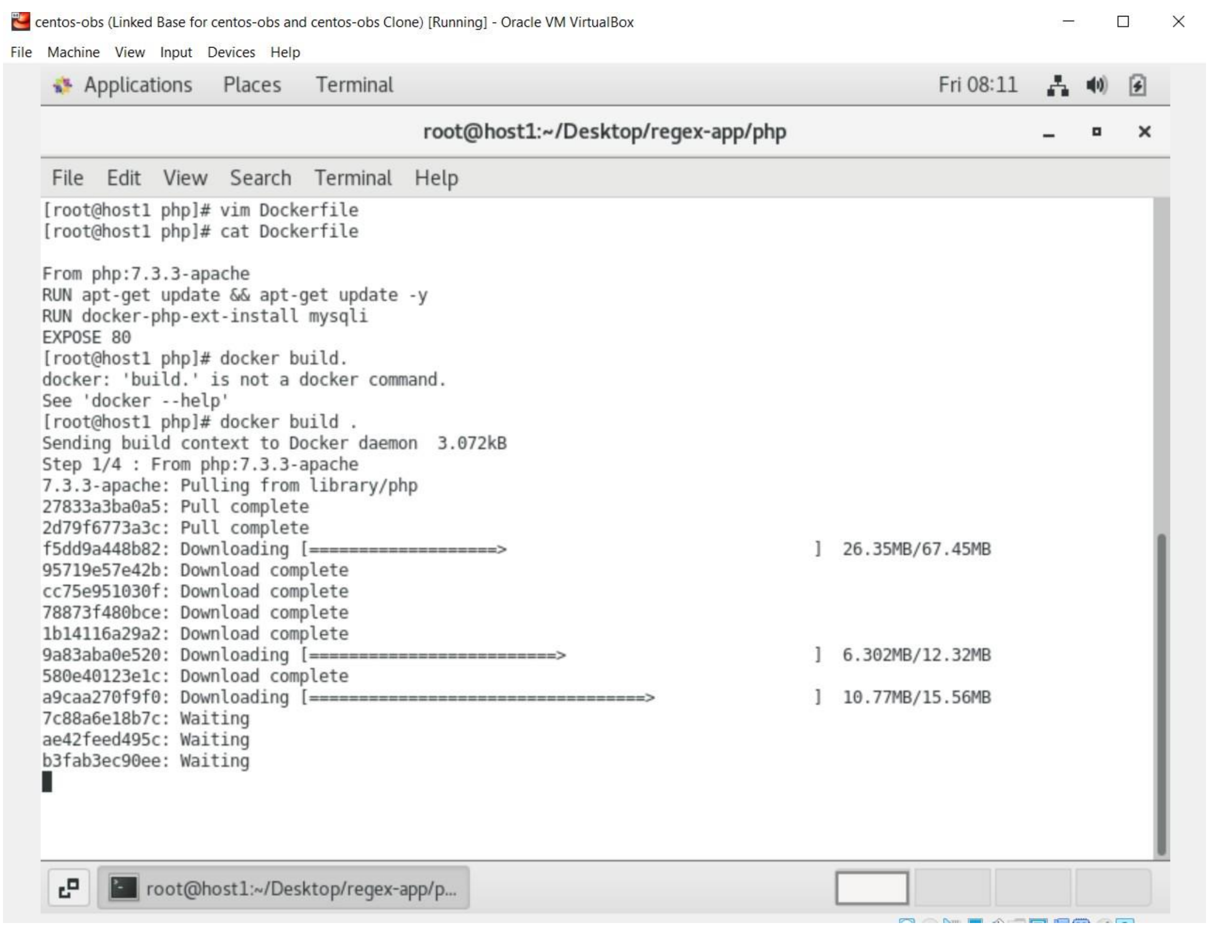
sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 08:56
root@host1:~
File Edit View Search Terminal Help

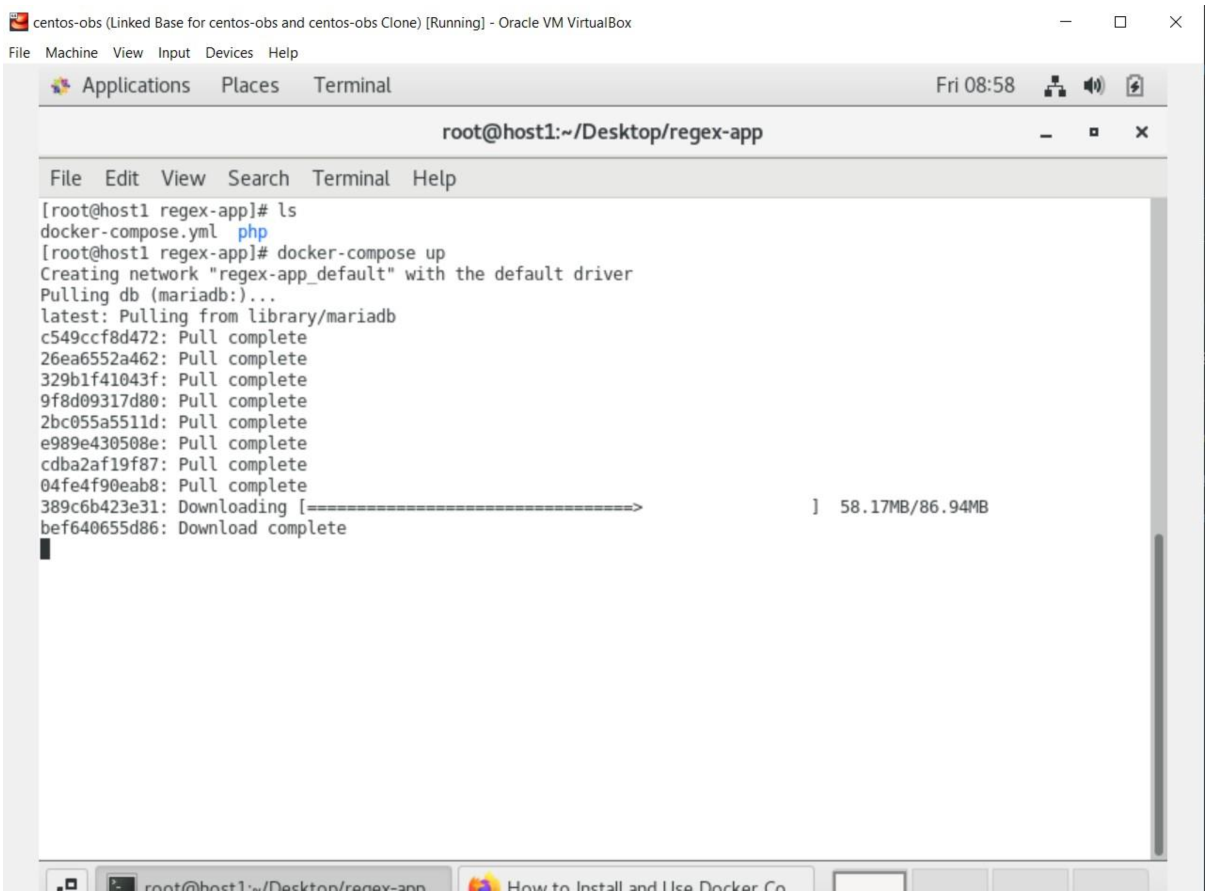
```
[root@host1 ~]# sudo curl -L "https://github.com/docker/compose/releases/download/1.23.1/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   633  100   633    0     0     82      0  0:00:07  0:00:07 --:--:--   138
100 11.1M  100 11.1M    0     0   788k      0  0:00:14  0:00:14 --:--:-- 1743k
[root@host1 ~]# sudo chmod +x /usr/local/bin/docker-compose
[root@host1 ~]# docker-compose --version
docker-compose version 1.23.1, build b02f1306
[root@host1 ~]#
```

Step 6: Build Docker Container.


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 08:11
root@host1:~/Desktop/regex-app/php
File Edit View Search Terminal Help
root@host1:~/Desktop/regex-app/p...

```
[root@host1 php]# vim Dockerfile
[root@host1 php]# cat Dockerfile

From php:7.3.3-apache
RUN apt-get update && apt-get update -y
RUN docker-php-ext-install mysqli
EXPOSE 80
[root@host1 php]# docker build.
docker: 'build.' is not a docker command.
See 'docker --help'
[root@host1 php]# docker build .
Sending build context to Docker daemon  3.072kB
Step 1/4 : From php:7.3.3-apache
7.3.3-apache: Pulling from library/php
27833a3ba0a5: Pull complete
2d79f6773a3c: Pull complete
f5dd9a448b82: Downloading [===================>                              ]  26.35MB/67.45MB
95719e57e42b: Download complete
cc75e951030f: Download complete
78873f480bce: Download complete
1b14116a29a2: Download complete
9a83aba0e520: Downloading [=========================>                        ]  6.302MB/12.32MB
580e40123e1c: Download complete
a9caa270f9f0: Downloading [==================================>               ]  10.77MB/15.56MB
7c88a6e18b7c: Waiting
ae42feed495c: Waiting
b3fab3ec90ee: Waiting
```

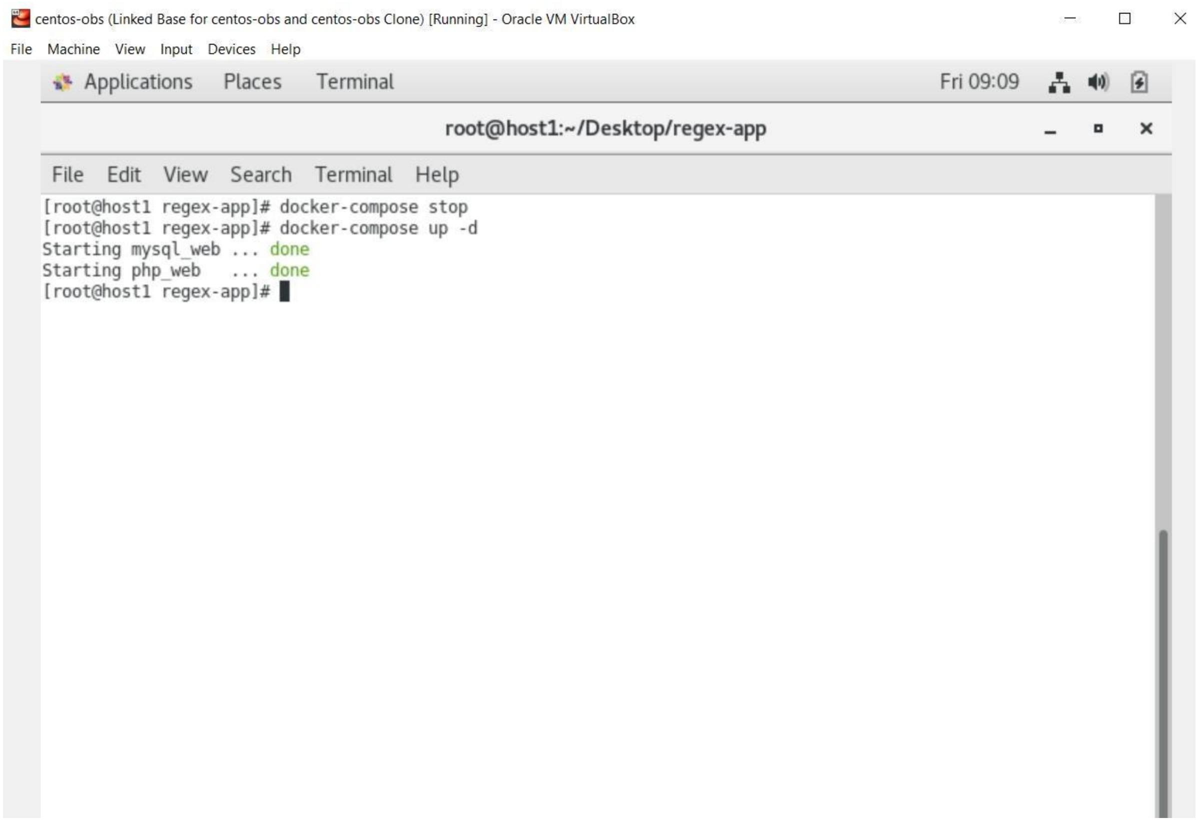
Step 7: Use "docker-compose up " command to aggregate the output of each container.


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 08:58
root@host1:~/Desktop/regex-app
File Edit View Search Terminal Help

```
[root@host1 regex-app]# ls
docker-compose.yml  php
[root@host1 regex-app]# docker-compose up
Creating network "regex-app_default" with the default driver
Pulling db (mariadb:)...
latest: Pulling from library/mariadb
c549ccf8d472: Pull complete
26ea6552a462: Pull complete
329b1f41043f: Pull complete
9f8d09317d80: Pull complete
2bc055a5511d: Pull complete
e989e430508e: Pull complete
cdba2af19f87: Pull complete
04fe4f90eab8: Pull complete
389c6b423e31: Downloading [================================>                 ]  58.17MB/86.94MB
bef640655d86: Download complete
```

Step 8:

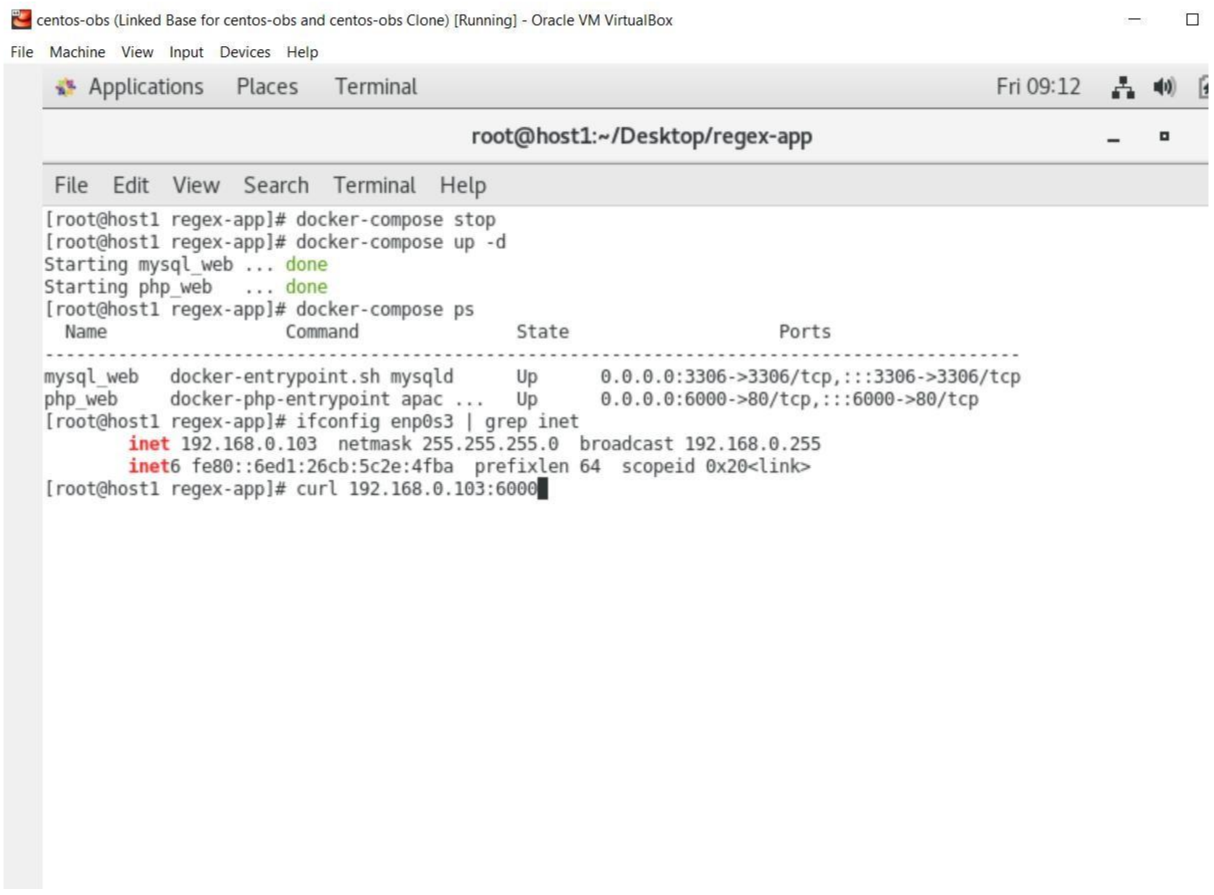
- Use "docker-compose stop" command to Stop the docker-compose.
- Use "docker-compose up -d" command for Detached mode: Run containers in the background.


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 09:09
root@host1:~/Desktop/regex-app
File Edit View Search Terminal Help

```
[root@host1 regex-app]# docker-compose stop
[root@host1 regex-app]# docker-compose up -d
Starting mysql_web ... done
Starting php_web   ... done
[root@host1 regex-app]# 
```

Step 9:

- Use "docker-compose ps" which only shows running containers.
- Use "curl 192.168.29.22:6000" server-id with 6000 port number.
- It will show output as "Successfuly Connected to MYSQL"


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 09:12
root@host1:~/Desktop/regex-app
File Edit View Search Terminal Help

```
[root@host1 regex-app]# docker-compose stop
[root@host1 regex-app]# docker-compose up -d
Starting mysql_web ... done
Starting php_web   ... done
[root@host1 regex-app]# docker-compose ps
  Name                 Command                State                    Ports
-------------------------------------------------------------------------------------------
mysql_web   docker-entrypoint.sh mysqld      Up      0.0.0.0:3306->3306/tcp,:::3306->3306/tcp
php_web     docker-php-entrypoint apac ...   Up      0.0.0.0:6000->80/tcp,:::6000->80/tcp
[root@host1 regex-app]# ifconfig enp0s3 | grep inet
        inet 192.168.0.103  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::6ed1:26cb:5c2e:4fba  prefixlen 64  scopeid 0x20<link>
[root@host1 regex-app]# curl 192.168.0.103:6000
```

#TASK2.2: *Dockerfile*

1) Webserver

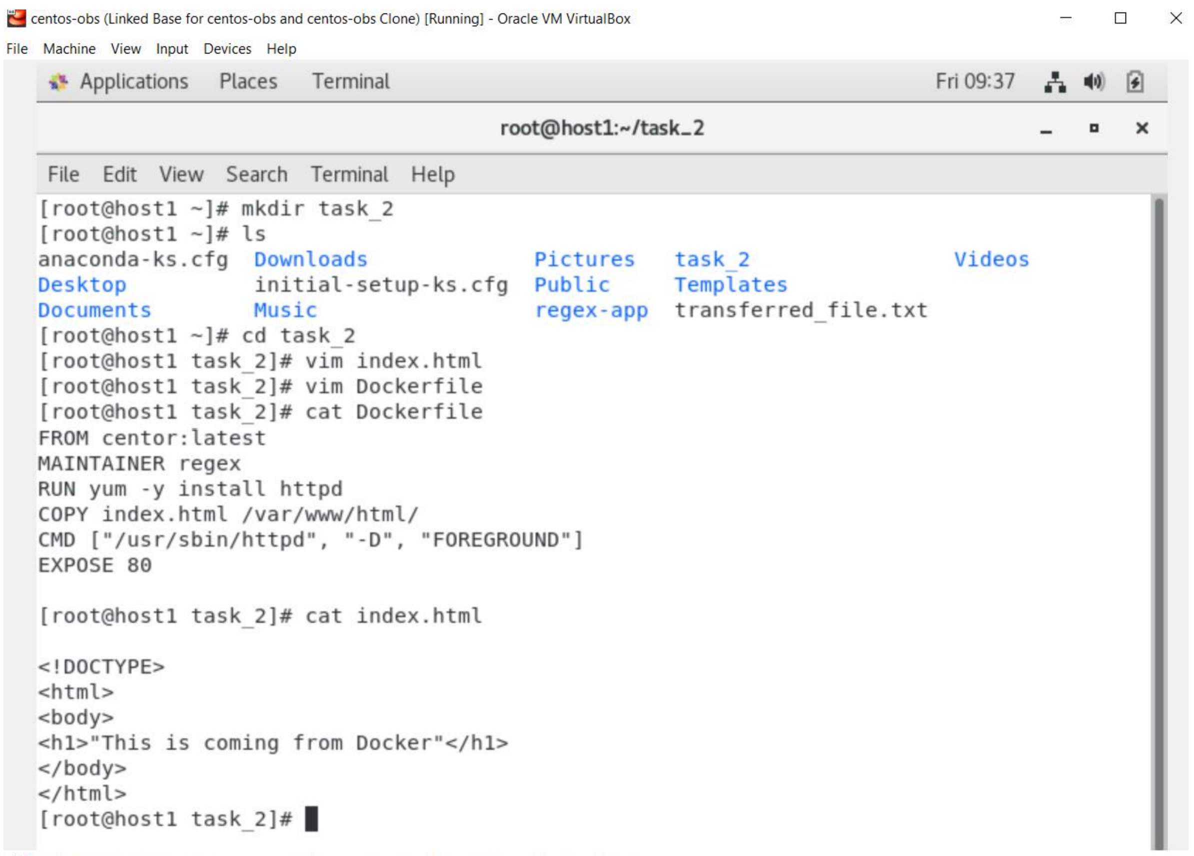
2) This is coming from Docker ---> Content

3) CentOS

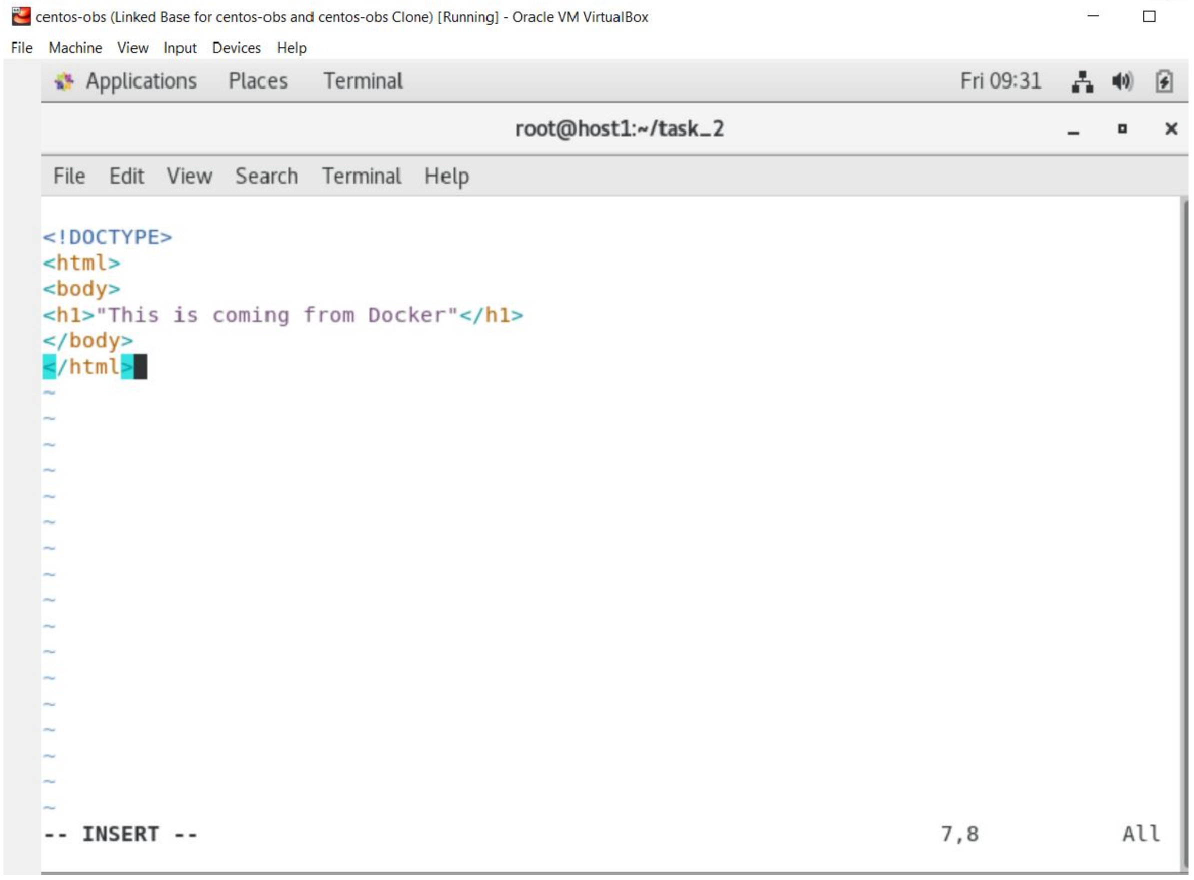
SOLUTION:

Step 1:

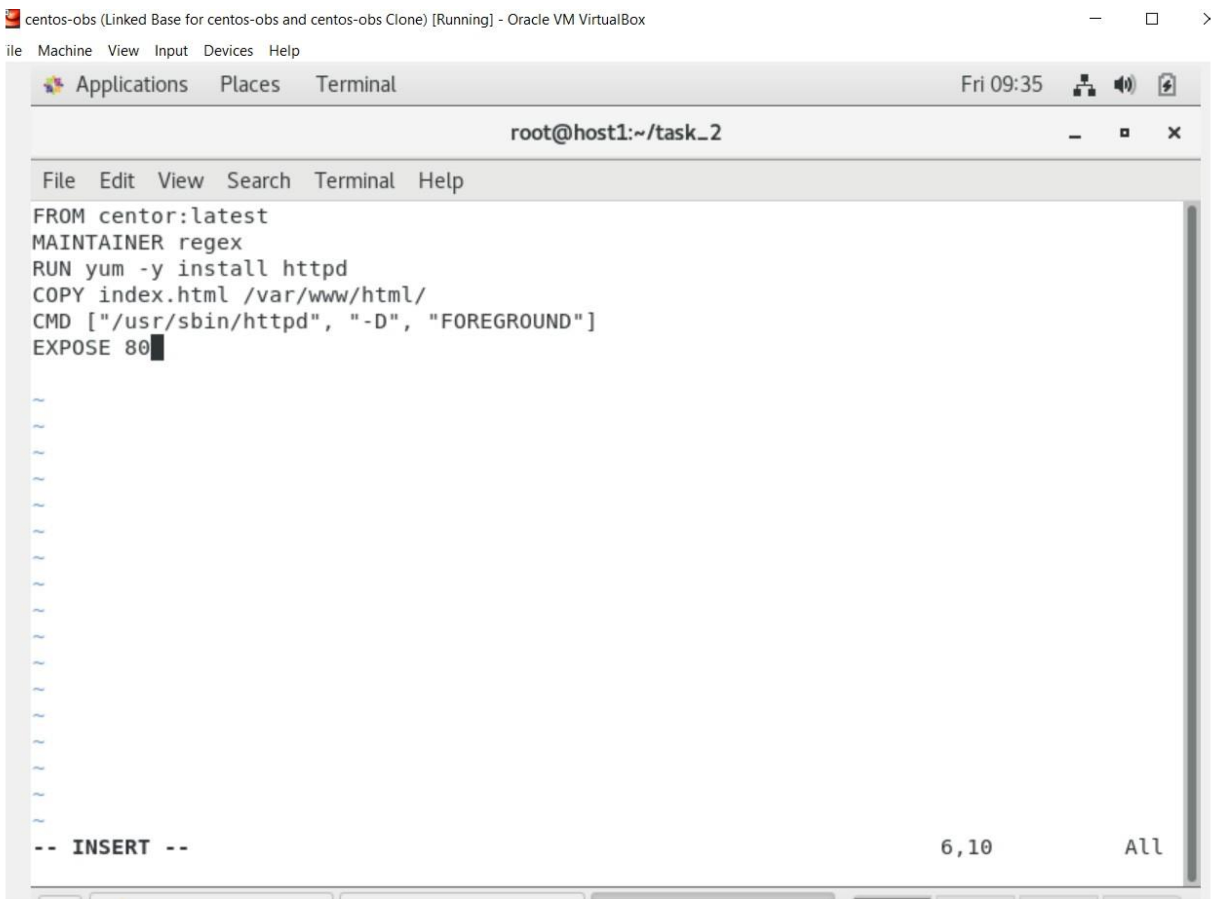
- Create a new directory
- Create index.html file with some html content "This is coming from Docker" using "vim" command.
- Create a Dockerfile which includes specific keywords that dictate how to build a specific image.


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 09:37
root@host1:~/task_2
File Edit View Search Terminal Help

```
[root@host1 ~]# mkdir task_2
[root@host1 ~]# ls
anaconda-ks.cfg  Downloads              Pictures   task_2              Videos
Desktop          initial-setup-ks.cfg   Public     Templates
Documents        Music                  regex-app  transferred_file.txt
[root@host1 ~]# cd task_2
[root@host1 task_2]# vim index.html
[root@host1 task_2]# vim Dockerfile
[root@host1 task_2]# cat Dockerfile
FROM centor:latest
MAINTAINER regex
RUN yum -y install httpd
COPY index.html /var/www/html/
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
EXPOSE 80

[root@host1 task_2]# cat index.html

<!DOCTYPE>
<html>
<body>
<h1>"This is coming from Docker"</h1>
</body>
</html>
[root@host1 task_2]#
```


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 09:31
root@host1:~/task_2
File Edit View Search Terminal Help

```

<!DOCTYPE>
<html>
<body>
<h1>"This is coming from Docker"</h1>
</body>
</html>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    7,8           All
```

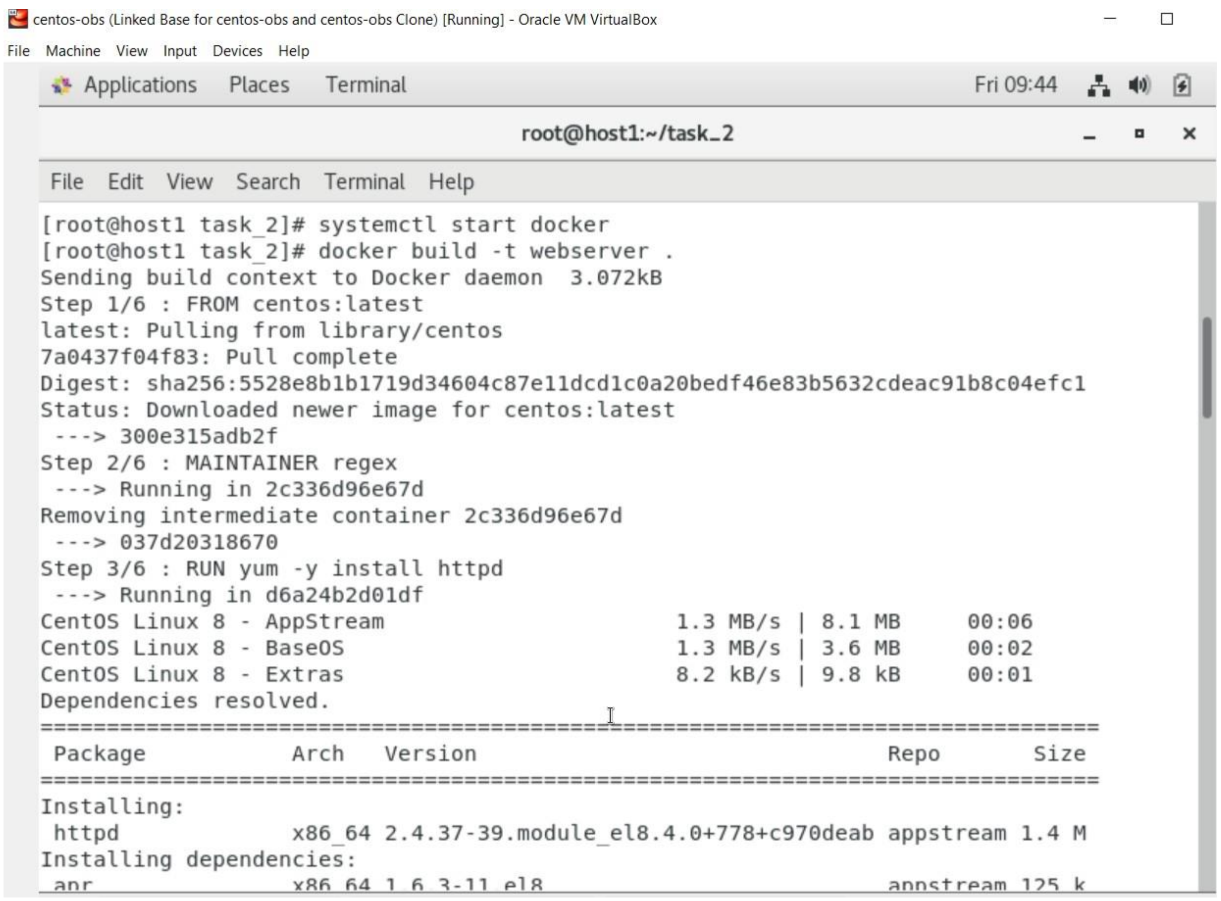

centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
ile Machine View Input Devices Help
Applications Places Terminal
Fri 09:35
root@host1:~/task_2
File Edit View Search Terminal Help

```
FROM centor:latest
MAINTAINER regex
RUN yum -y install httpd
COPY index.html /var/www/html/
CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]
EXPOSE 80

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                          6,10          All
```

Step 2:

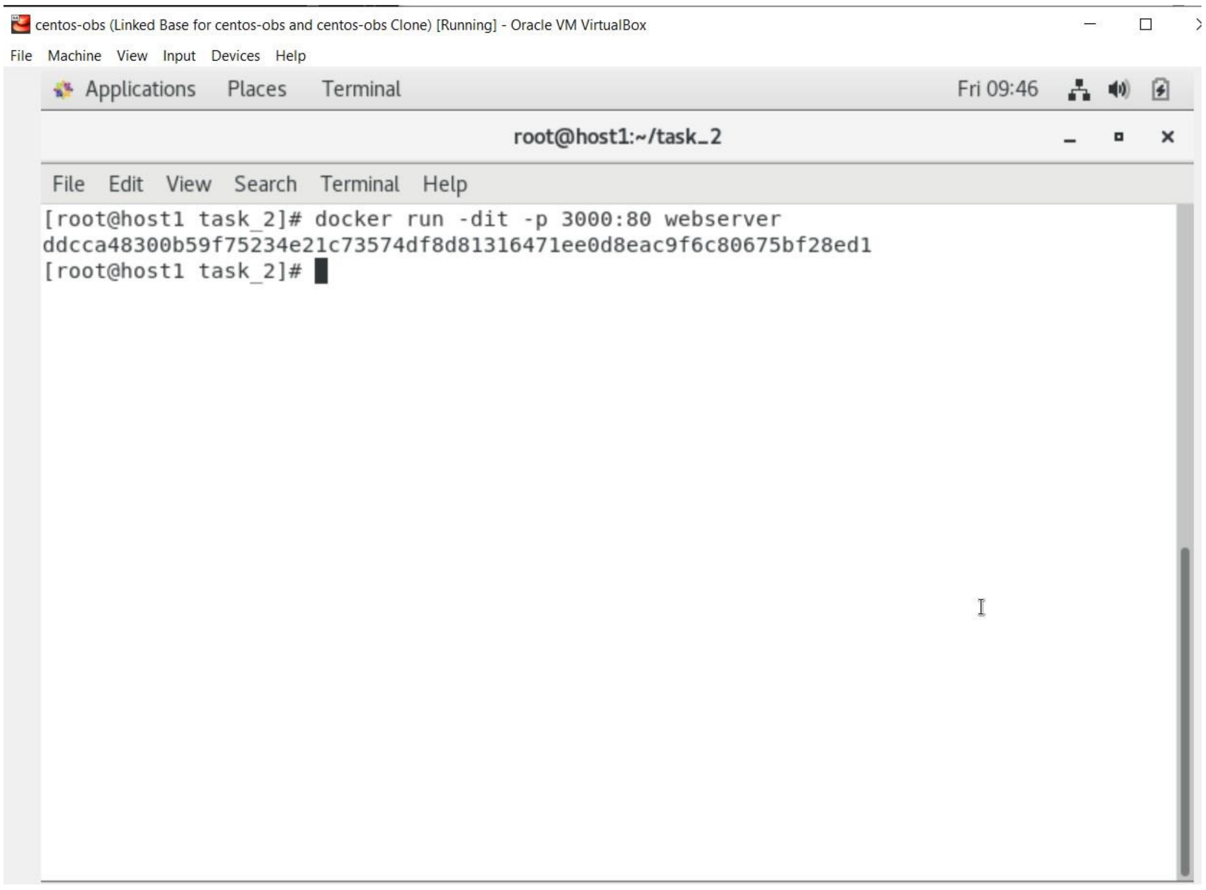
- Use "systemctl start docker" to start the Docker Service

Use "docker build -t webserver ." to create a Docker image from the definition contained in a Dockerfile


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 09:44
root@host1:~/task_2
File Edit View Search Terminal Help

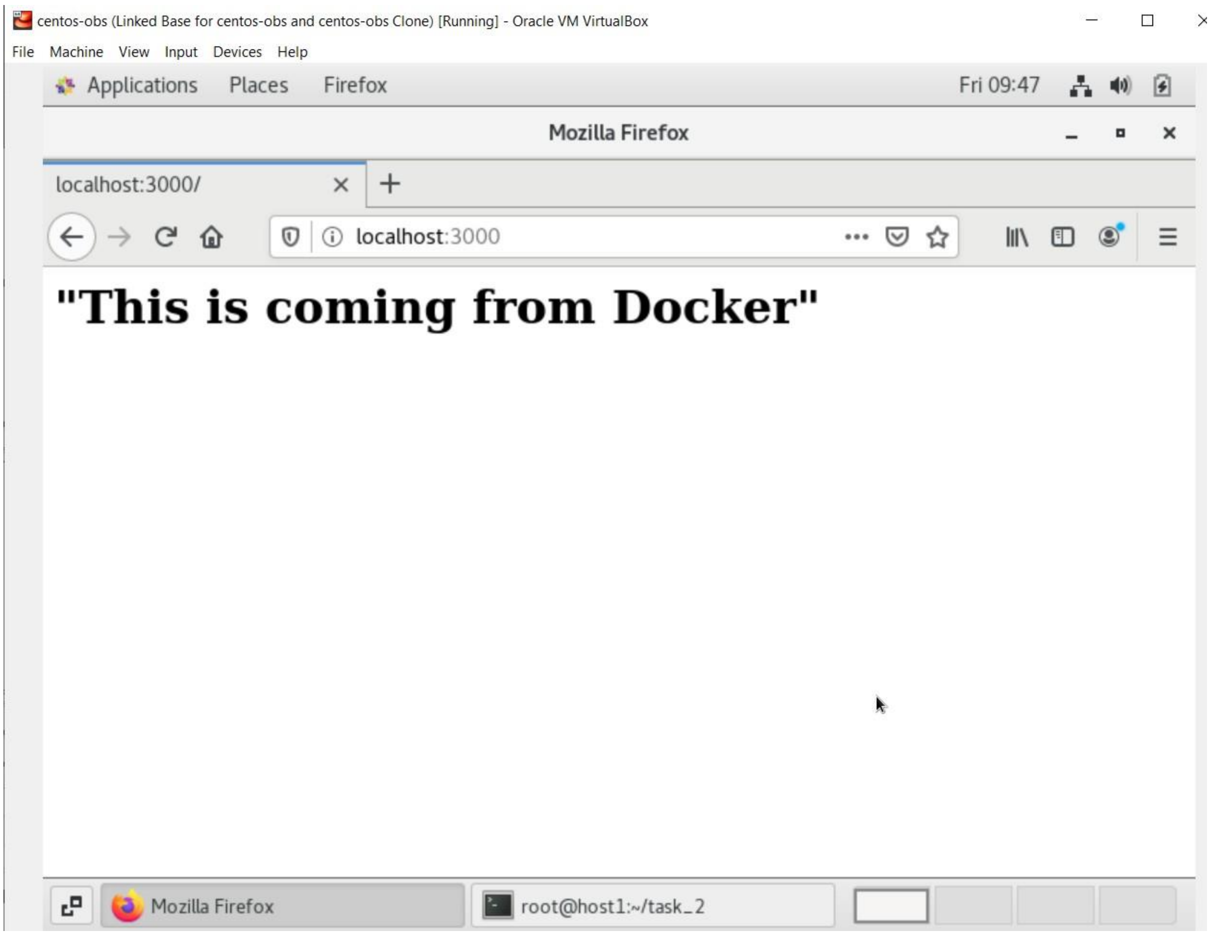
```
[root@host1 task_2]# systemctl start docker
[root@host1 task_2]# docker build -t webserver .
Sending build context to Docker daemon  3.072kB
Step 1/6 : FROM centos:latest
latest: Pulling from library/centos
7a0437f04f83: Pull complete
Digest: sha256:5528e8b1b1719d34604c87e11dcd1c0a20bedf46e83b5632cdeac91b8c04efc1
Status: Downloaded newer image for centos:latest
 ---> 300e315adb2f
Step 2/6 : MAINTAINER regex
 ---> Running in 2c336d96e67d
Removing intermediate container 2c336d96e67d
 ---> 037d20318670
Step 3/6 : RUN yum -y install httpd
 ---> Running in d6a24b2d01df
CentOS Linux 8 - AppStream                     1.3 MB/s | 8.1 MB     00:06
CentOS Linux 8 - BaseOS                        1.3 MB/s | 3.6 MB     00:02
CentOS Linux 8 - Extras                        8.2 kB/s | 9.8 kB     00:01
Dependencies resolved.
================================================================================
 Package           Arch   Version                              Repo       Size
================================================================================
Installing:
 httpd             x86_64 2.4.37-39.module_el8.4.0+778+c970deab appstream 1.4 M
Installing dependencies:
 apr               x86_64 1.6.3-11.el8                         appstream 125 k
```

Step 3: Use "docker run -dit -p 3000:80 webserver" to run the container.


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal
Fri 09:46
root@host1:~/task_2
File Edit View Search Terminal Help

```
[root@host1 task_2]# docker run -dit -p 3000:80 webserver
ddcca48300b59f75234e21c73574df8d81316471ee0d8eac9f6c80675bf28ed1
[root@host1 task_2]# 
```

Step 4:


centos-obs (Linked Base for centos-obs and centos-obs Clone) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Fri 09:47
Mozilla Firefox
localhost:3000/
localhost:3000
Mozilla Firefox
root@host1:~/task_2

"This is coming from Docker"

After that go to the browser and Type => localhost:300