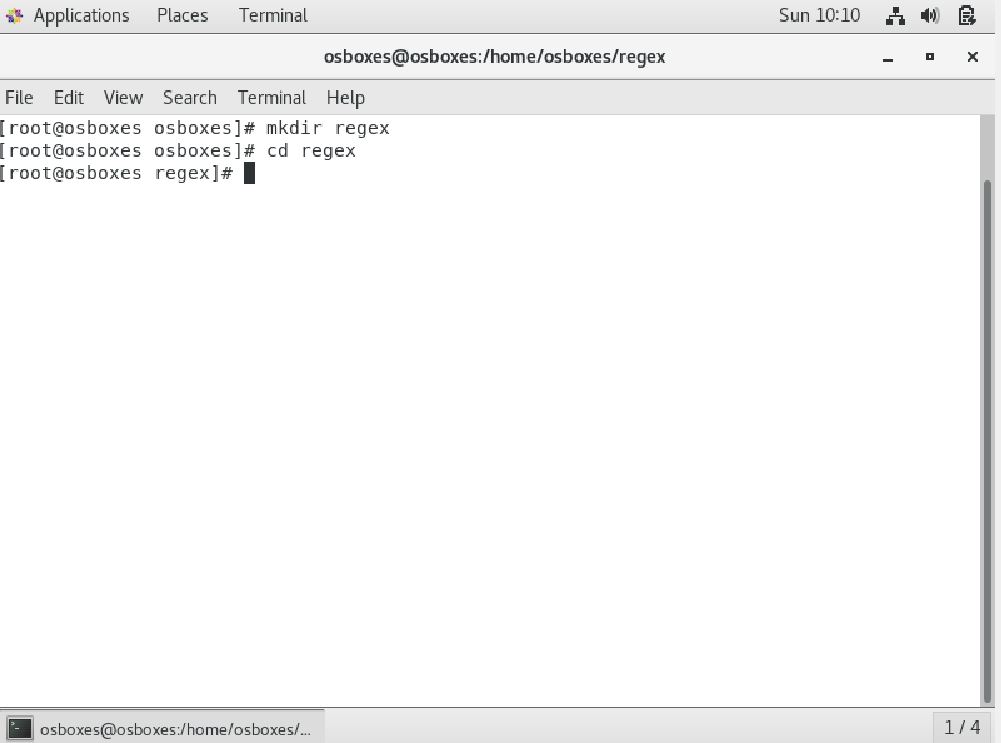
**Task Based on Docker Sessions:**  
  
**TASK2.1:**

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/ \_\_\_\_\_\_\_\_\_\_\_\_  
  
**TASK2.2: Dockerfile**

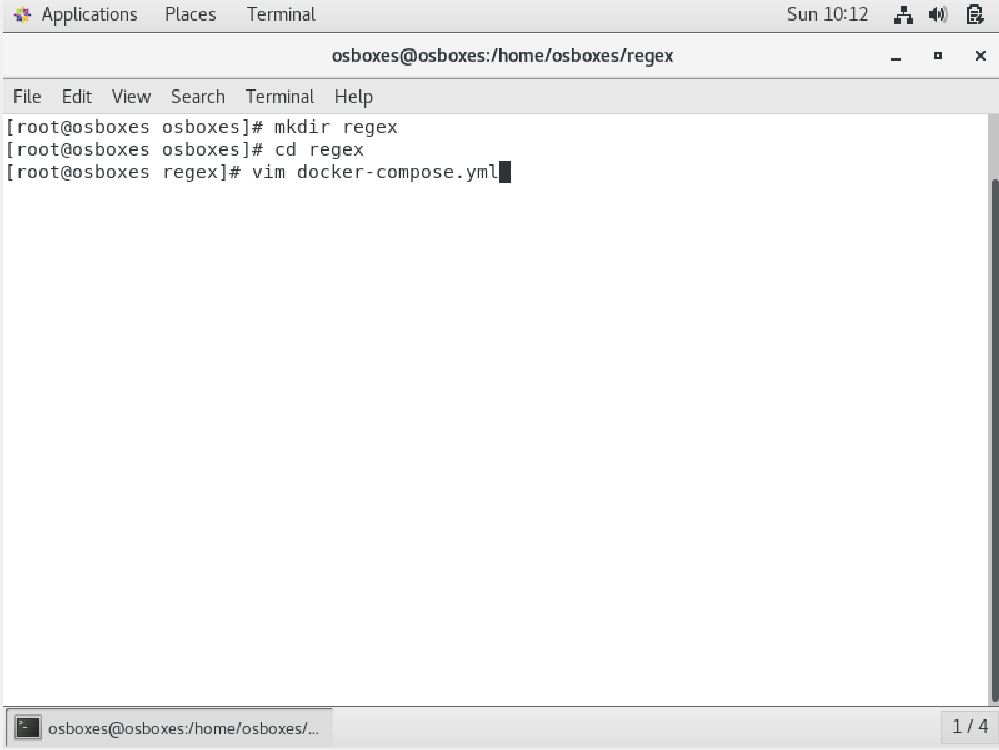
1) Webserver  
2) This is coming from Docker ---> Content  
3) CentOS

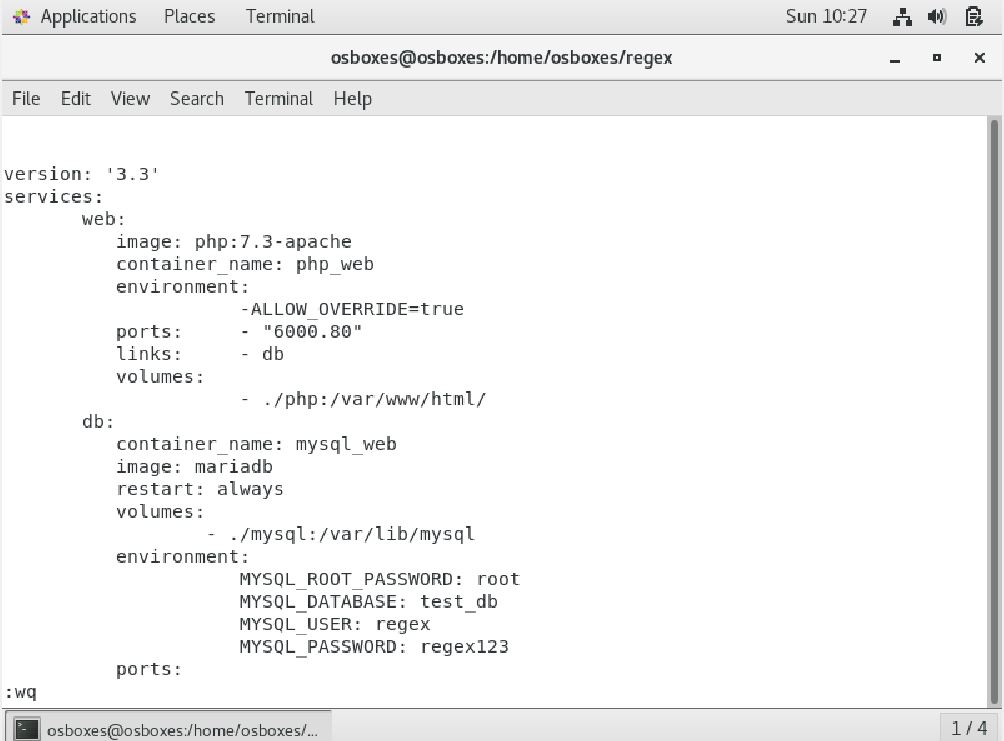
Task-1:

* Create a directory using "mkdir" command. Get into that directory using "cd" command.

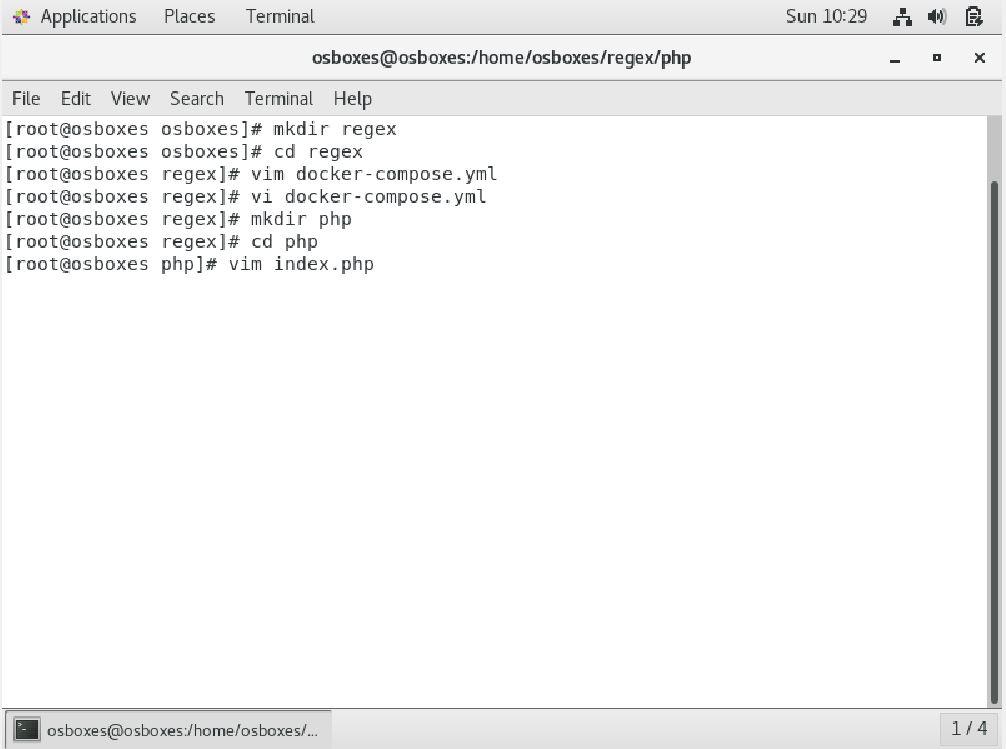


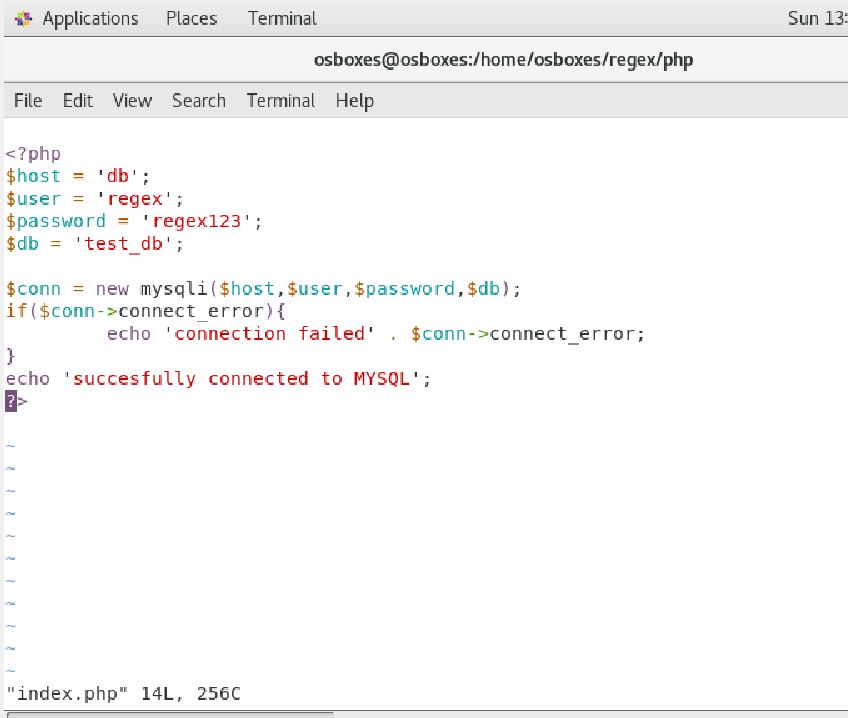
* Create docker-compose.yml file using "vim" command. Edit the docker-compose file.



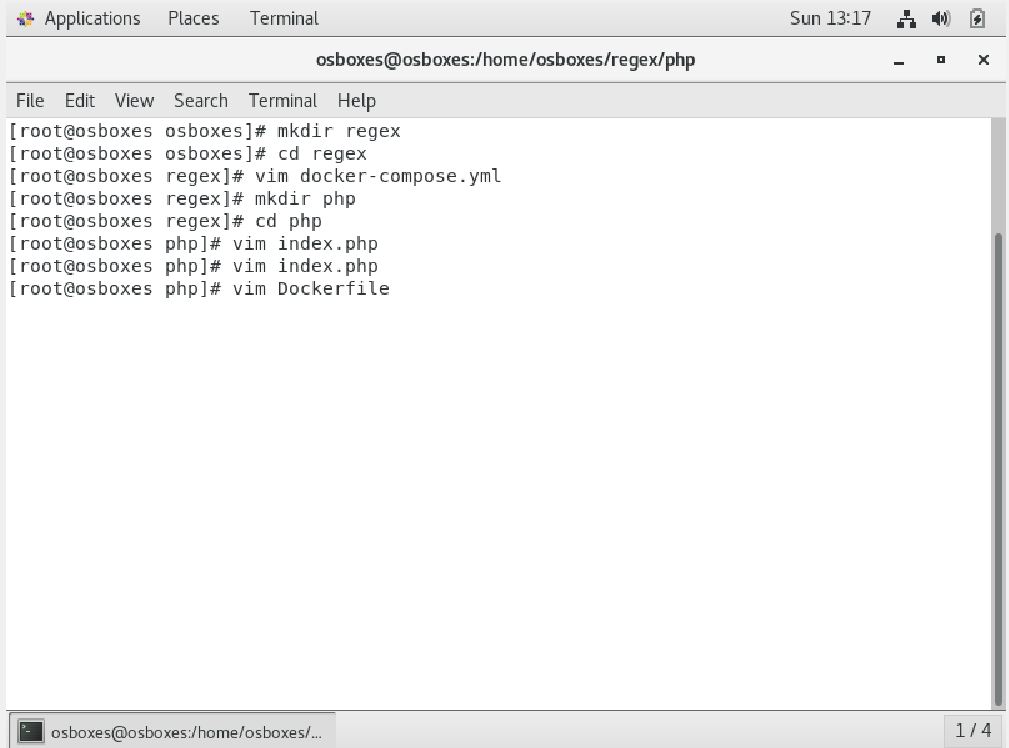


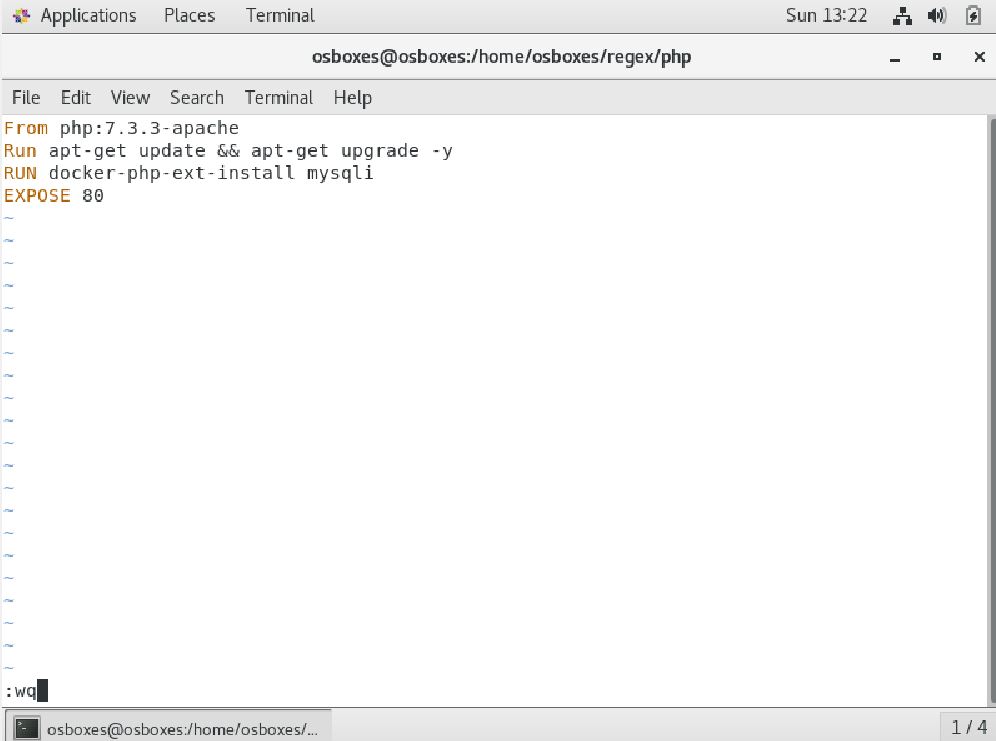
* Create a directory named "php" under the previous directory. Create index.php file php directory. Edit index.php and add credentials for mysql access.



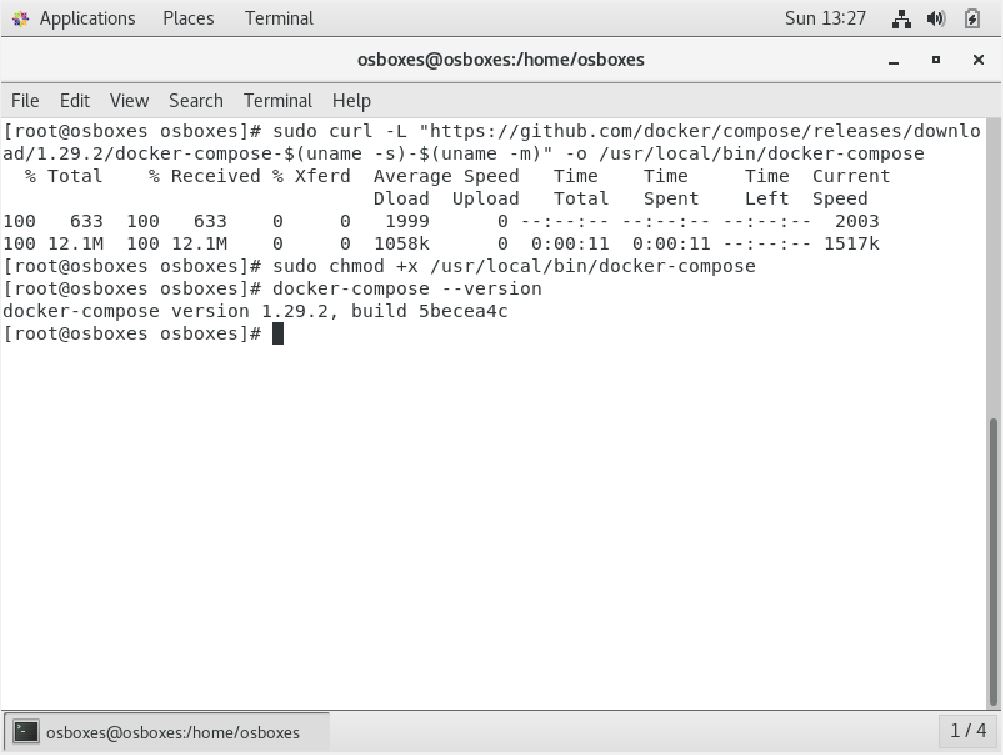


* Create a Dockerfile using "vim" command. Edit that file put specific keywords that dictate how to build a specific image.

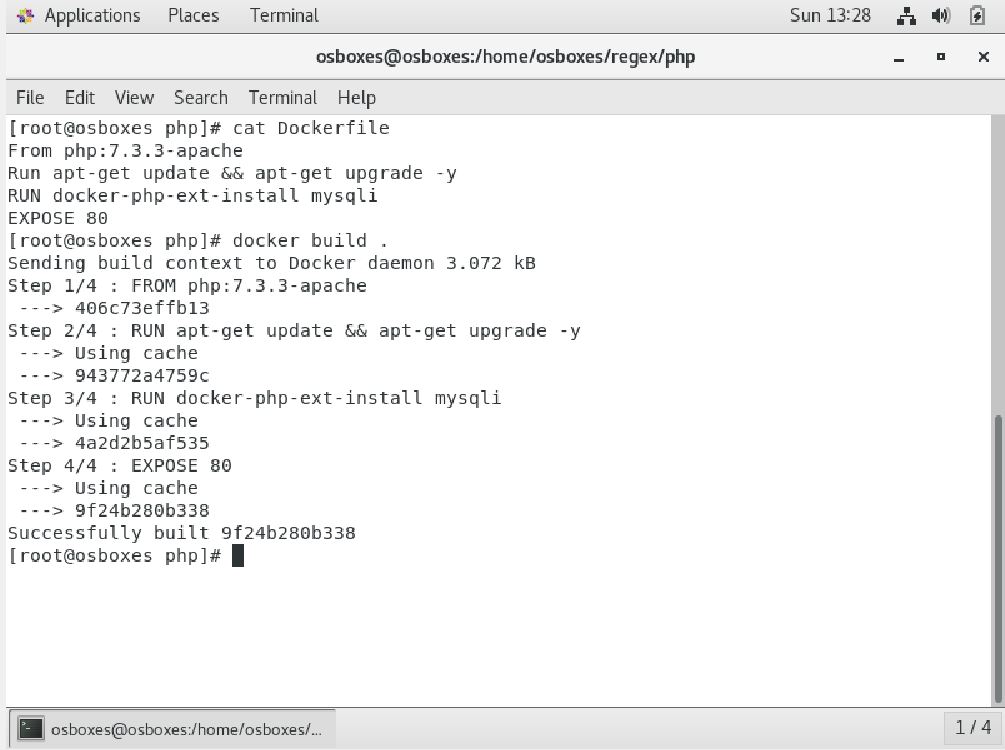




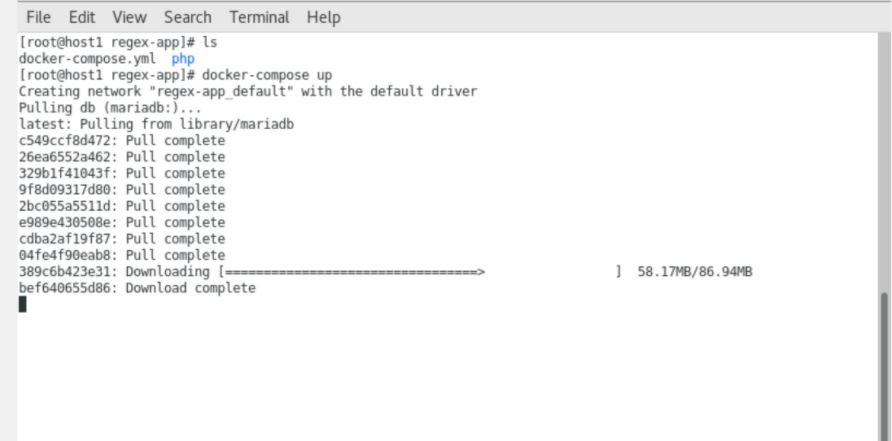
* Install Docker compose



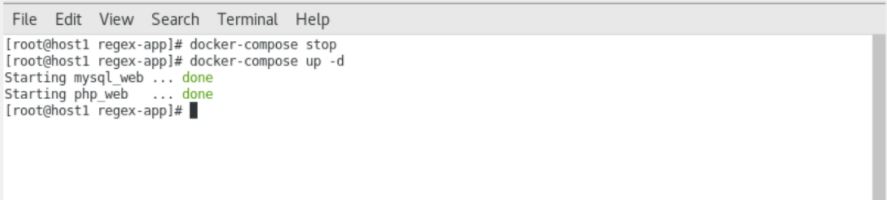
* Build Docker Container.



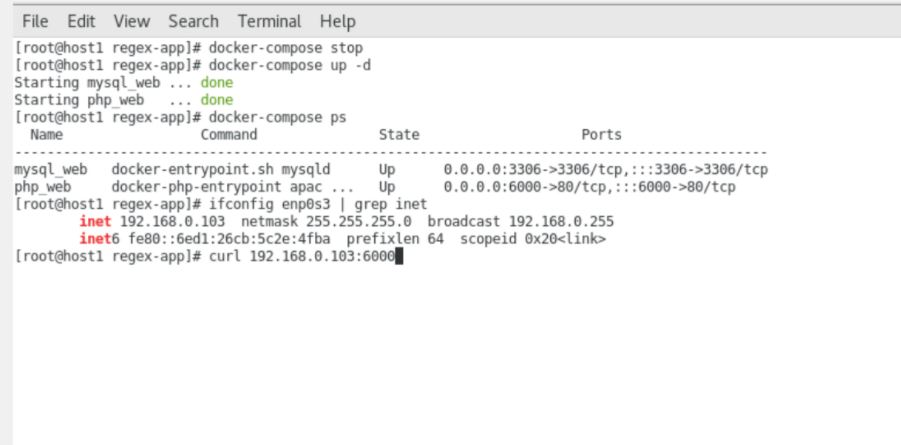
* Use "docker-compose up " command to aggregate the output of each container.



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



* 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".

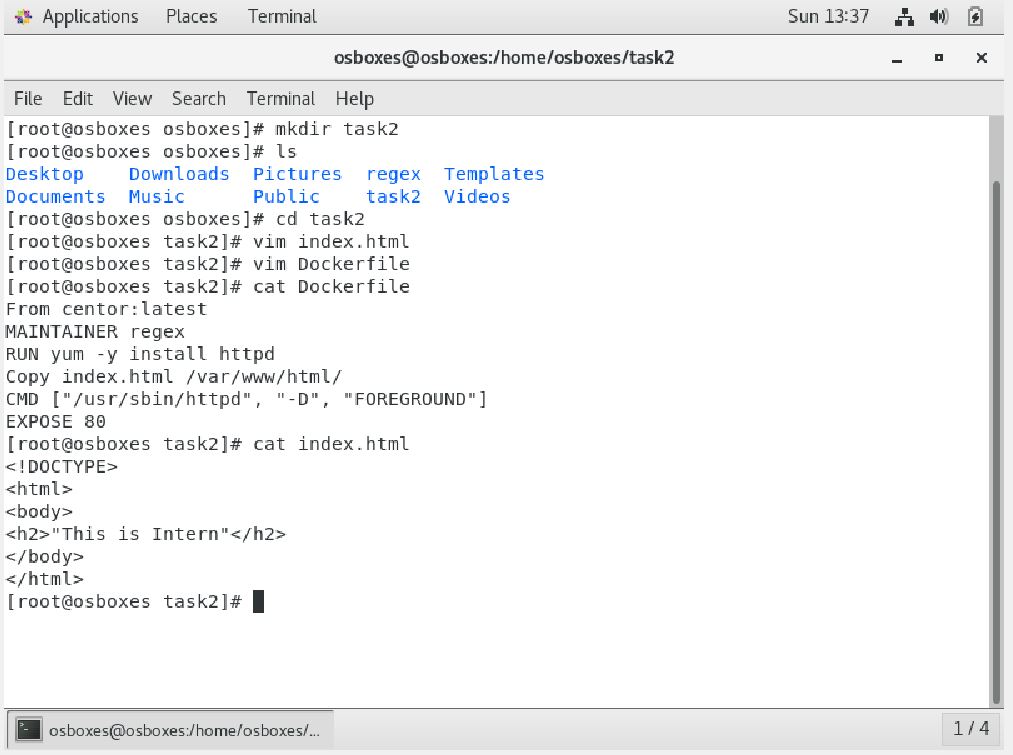


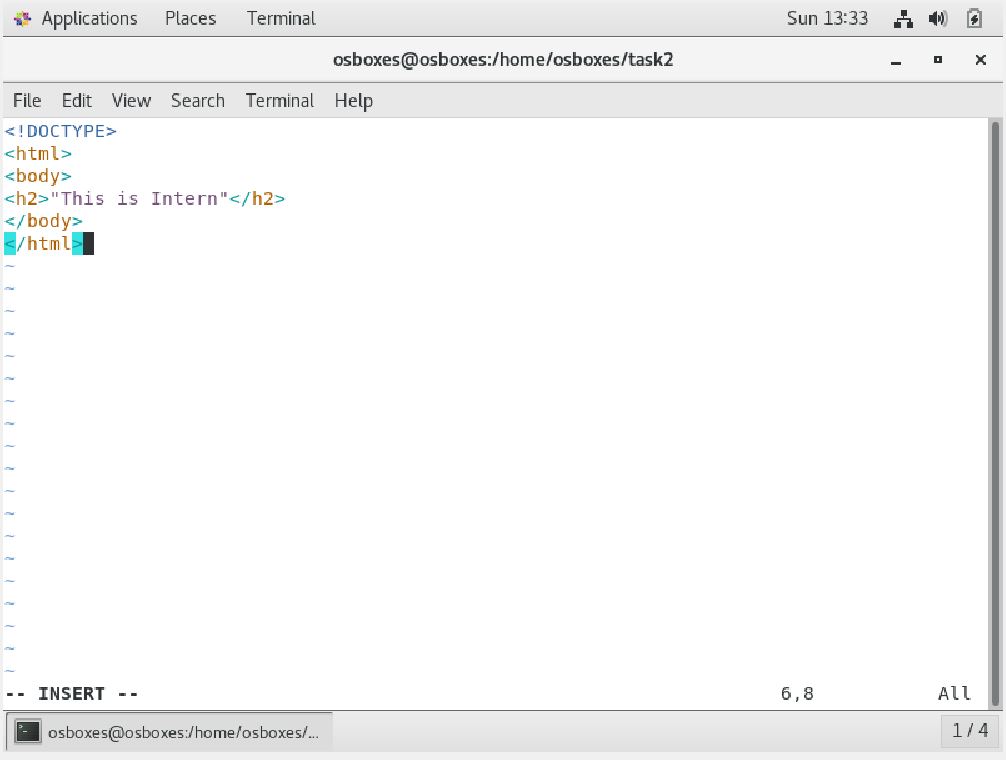
Task-2:

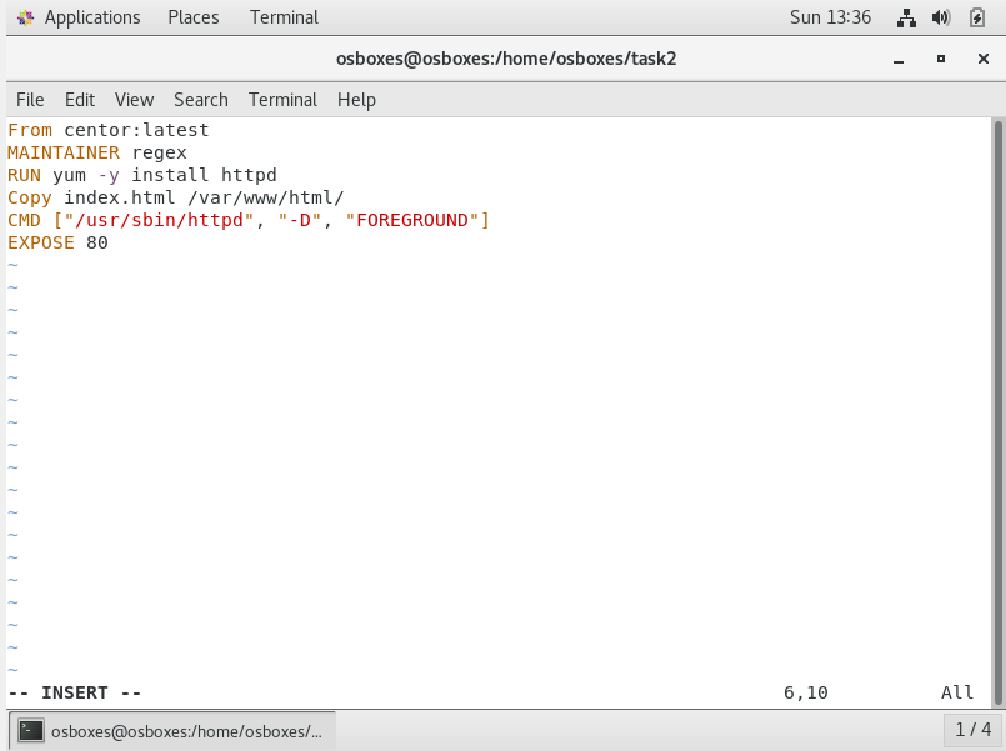
* Webserver
* This is coming from Docker ---> Content
* CentOS

Steps:

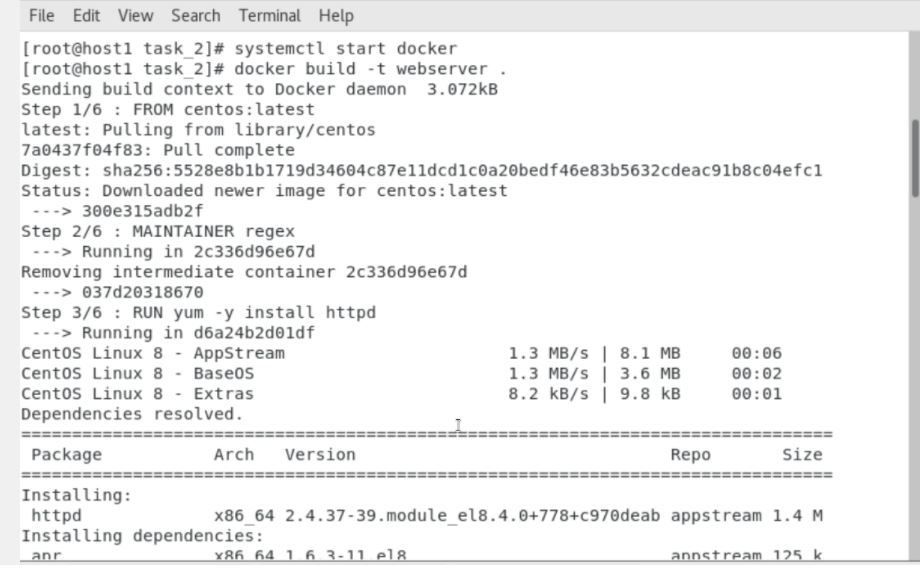
* Create a new directory
* Create index.html file with some html content "This is coming from docker" using "vim" command.
* Create a Dockerfile which includes specific keywords that dictate how to build a specific image.







* 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.



* Use**“**docker run -dit -p 3000:80 webserver” to run the container.
* After that go to the browser and Type => localhost:3000



These are the tasks about DOCKER.