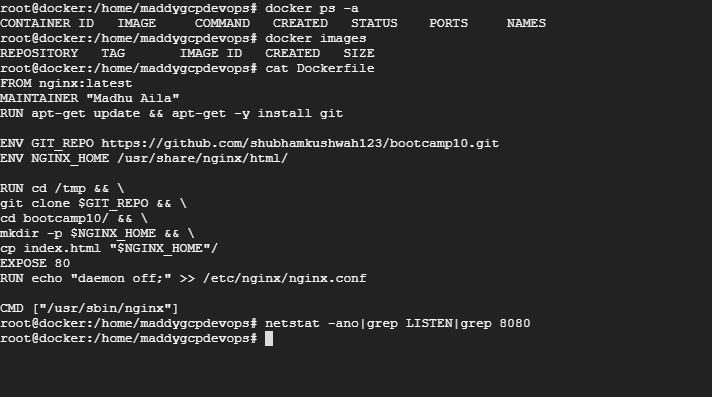
# **Docker**

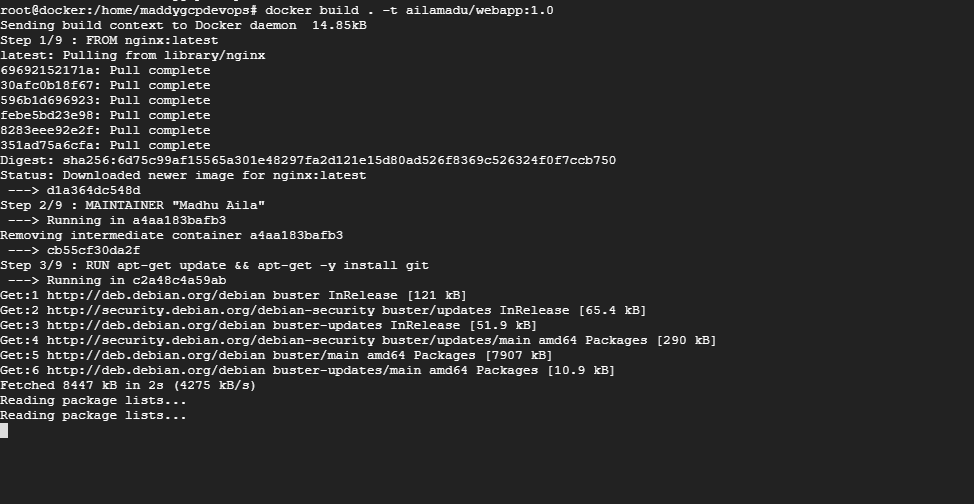
1. **Write a docker file to create a docker image for this application using nginx as the base image.**

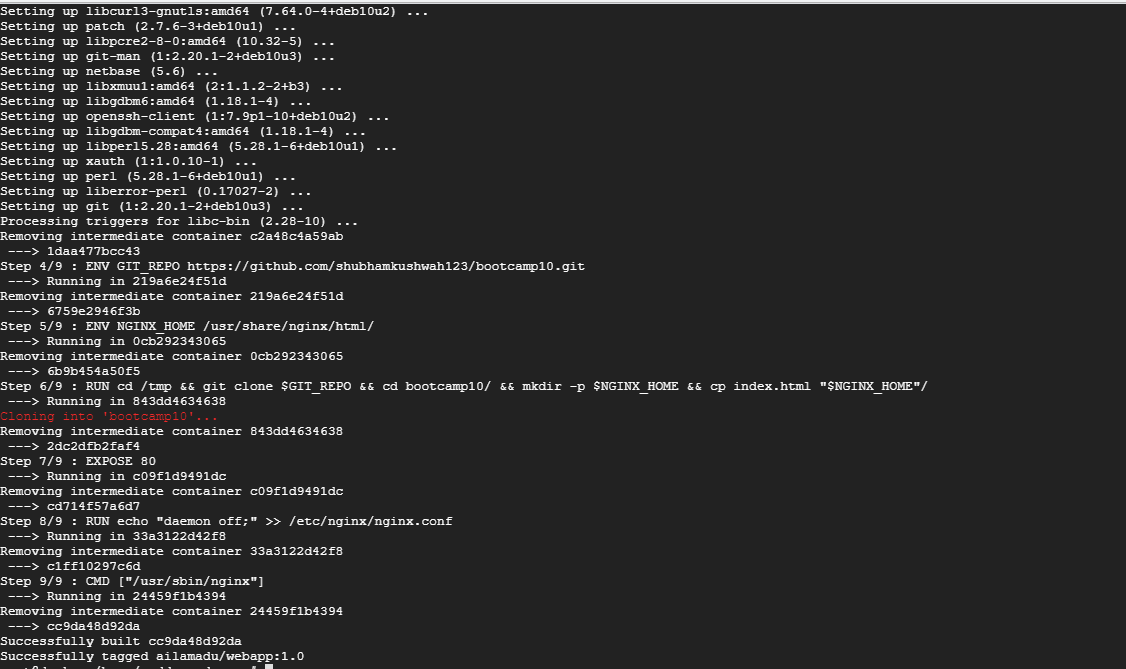
* I have checked whether any docker is running already since I have used this machine before.
* I have written a docker file which does below:
  + Uses the latest nginx image from dockerhub
  + Updated the repo and installed the git as I would like to clone the repo provided as part of the build.
  + For the above said purpose, I have declared the GIT\_REPO environment variable. Along with that, provided the NGINX\_HOME path where the index.html should be copied.
  + I have cloned the repo and copied the index.html to the NGINX\_HOME
  + Exposed 80 as nginx runs in 80 port.
  + I have turned off nginx being run as daemon
  + Finally started the nginx in the foreground so that I can view the logs with docker logs command.
* I have checked whether any process is running on 8080 just in case as I am planning to run the nginx in 8080.

**Note: I am doing these in machine called “docker”**



* Now, I have executed the docker build with tag.

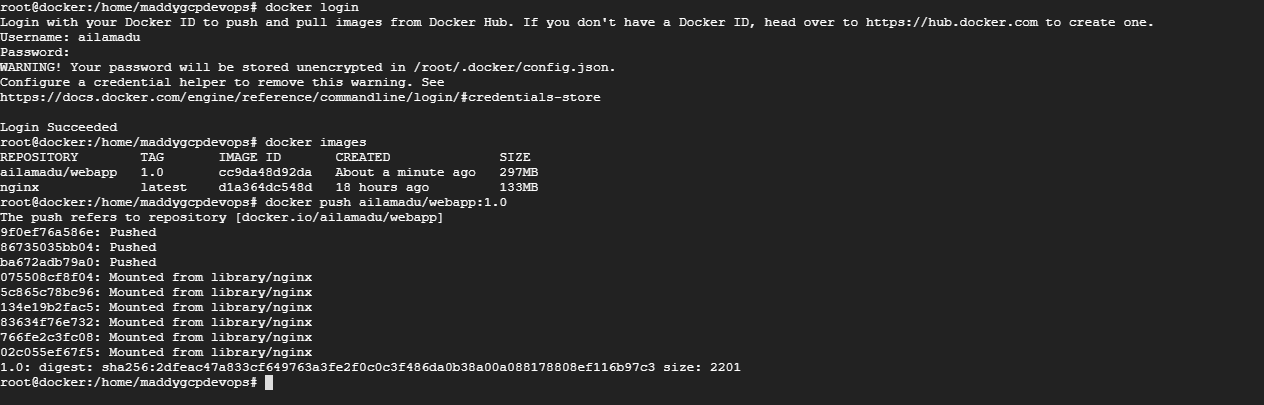




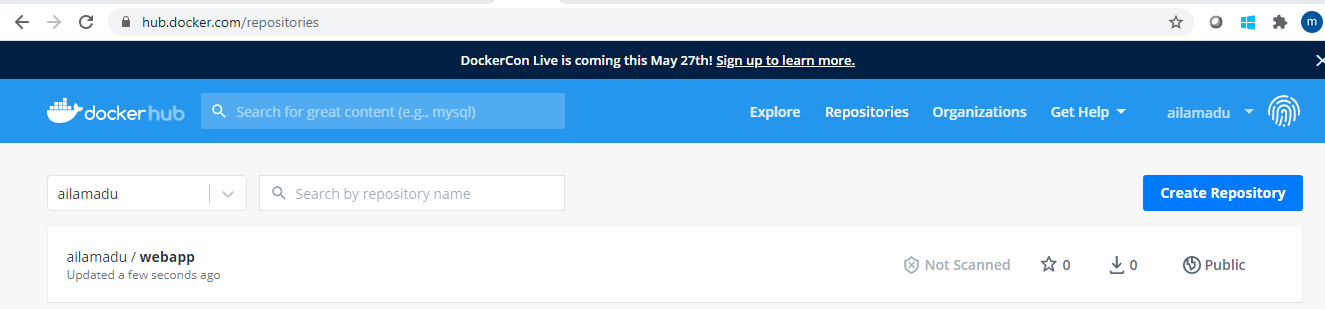
* In the above screenshot, build is completed successfully.

1. **Push the docker image to the Docker Hub.**

* Now, I have logged into the docker with my credentials and pushed the image to docker hub.

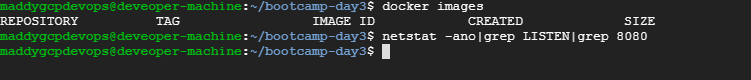


* Pushed image can be viewed in the dockerhub. PFB the screenshot.

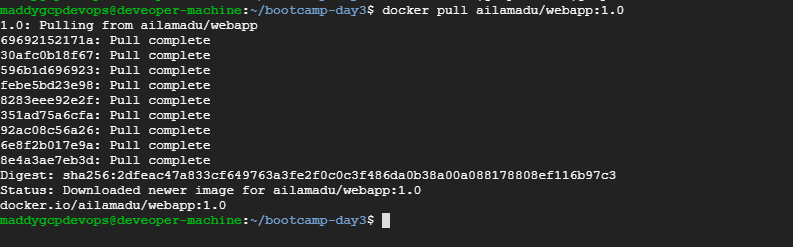


1. **Pull and Run the docker containers on any remote instance.**

* As it was requested to do the pull in the remote machine, I have used “developer-machine” . Different one from the one where the image was created.
* I have verified that there are no containers are running.
* 8080 port which is free to use.



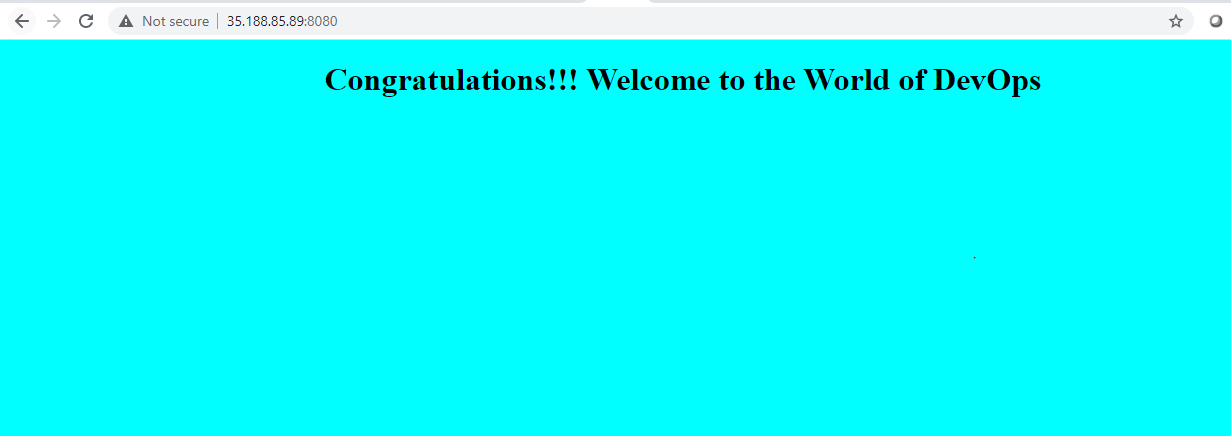
* I have pulled the image



* I have ran the docker run with 8080 as my port.



* Docker container is started to run and I am able to see the webpage.



# **Docker-compose**

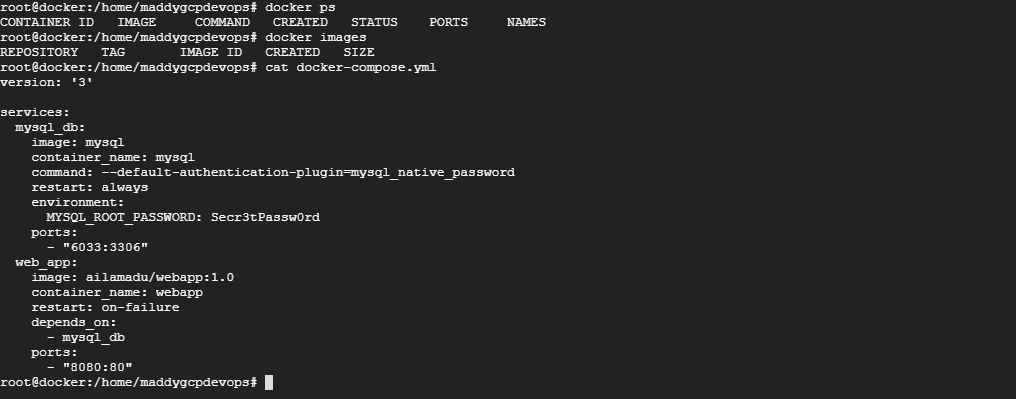
1. **Write a docker-compose.yml file to run the above application container along with a mysql database container together.** 
   * I have written the docker-compose.yml :
     + I will have 2 services one is mysql\_db and web\_app.

**Mysql\_db**

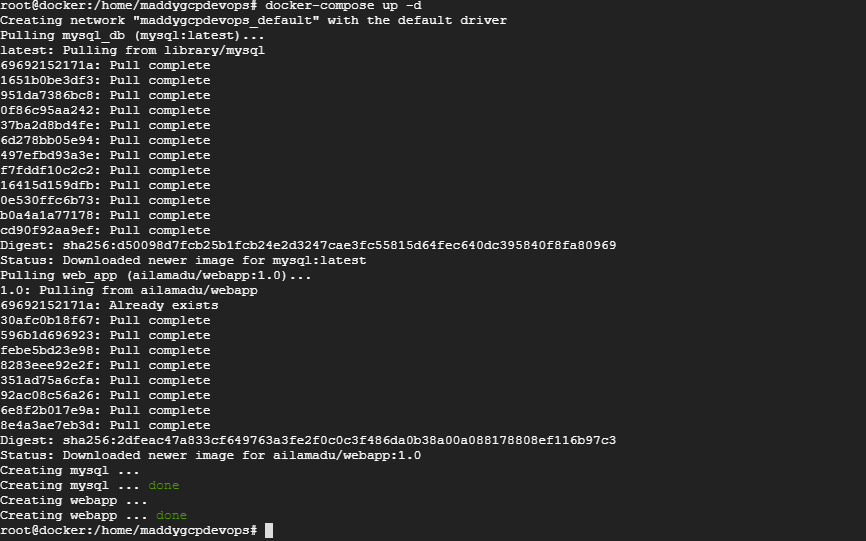
* + - I have used latest mysql and set the password and changed the authentication mode as per the dockerhub documentation
    - Used the port as 6033
    - Restart will be always

**Web\_app**

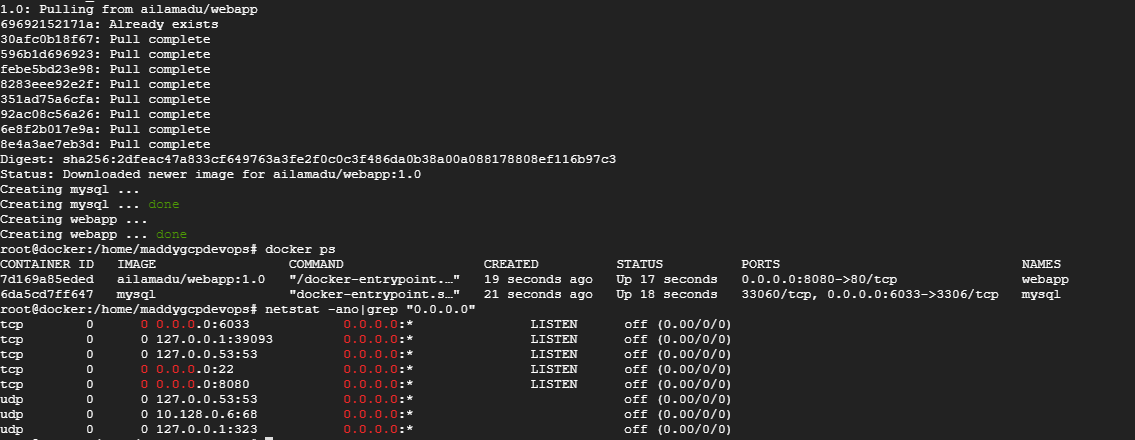
* + - I have used the pushed image in the previous exercise ailamadu/webapp:1.0
    - Added mysql\_db as a dependent though we have not done any tight integration
    - Port as 8080



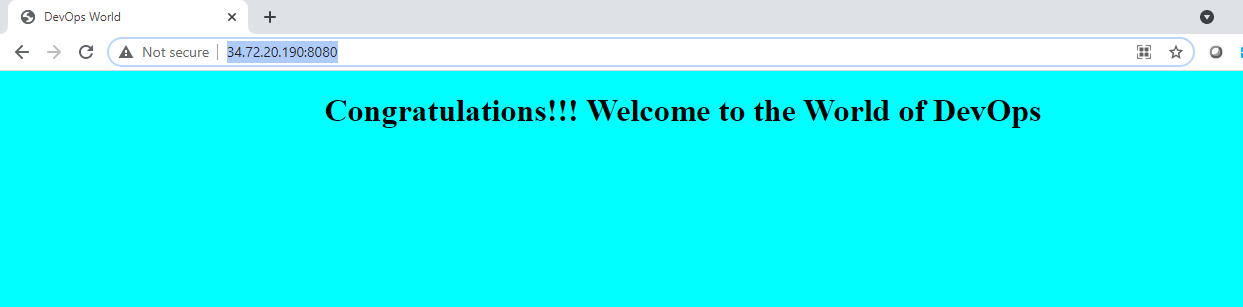
* I have executed docker-compose up in detached mode.



* Docker compose is successful and I was able to see 2 containers running as defined.
* Verified the ports are listening.



* After ensuing the ports are listening, I am able to see the web page as well. **Please note this screenshot is in different server from the previous exercise and note the ip is changed.**



1. **Connect to the mysql database container in the -it mode and create a table named test having id, name and age as attributes.**

* I have connected to docker container with exec -it
* Connected to mysql with password provided in the yml file.
* I have listed the databases
* Switched to mysql db
* Created a table as per the instructions.
* Listed the tables and ensured that test table is created.

