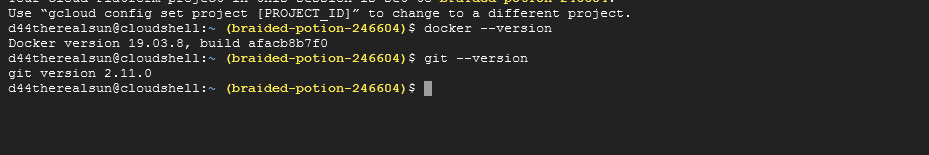
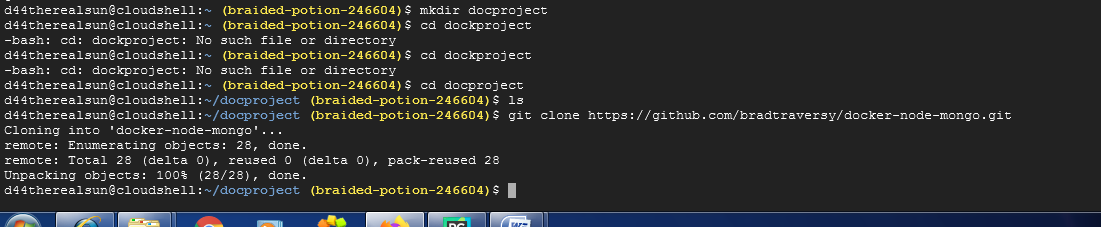
**Docker 201 Project case study Solution**

**We will be creating two tier Node application and will be containerizing the same.**

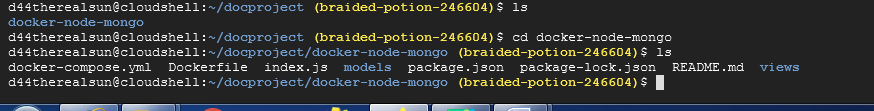
Check whether git and docker are installed already.

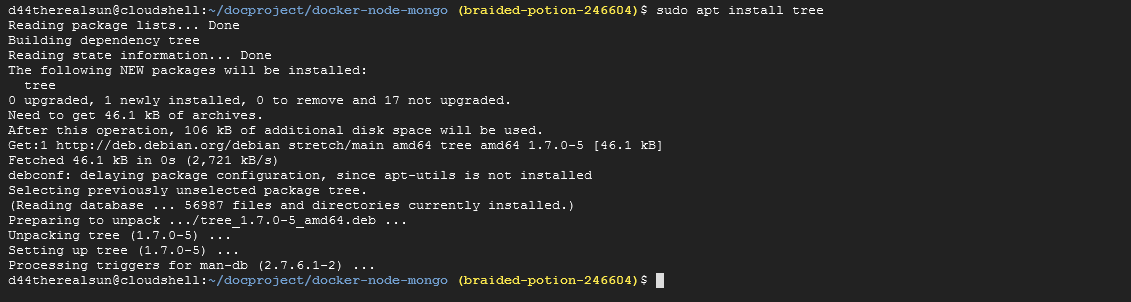


Now we will cloning 1 node js application code from internet.

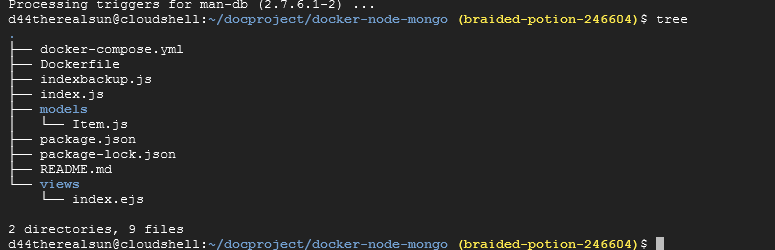


We could see the project contents downloaded in our directory

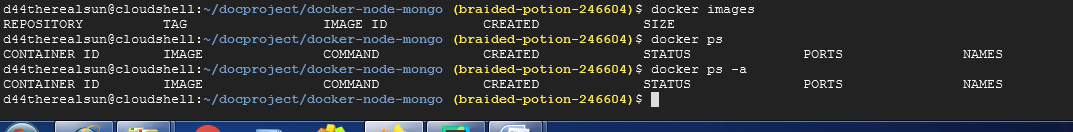




Tree command will help up to see the structure of the codes

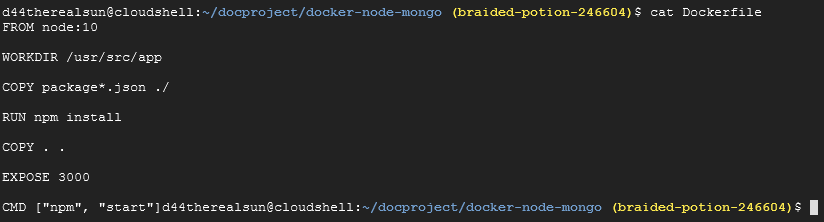


Try to execute some docker commands

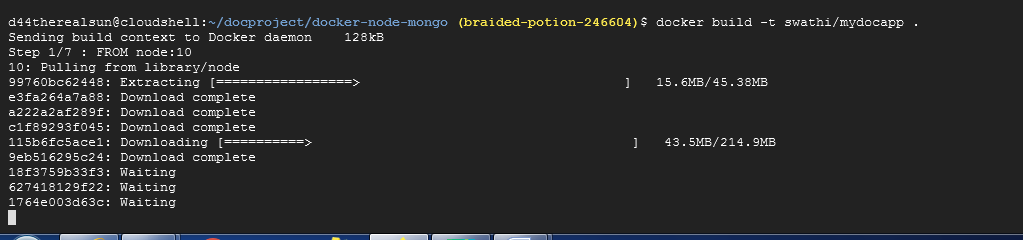


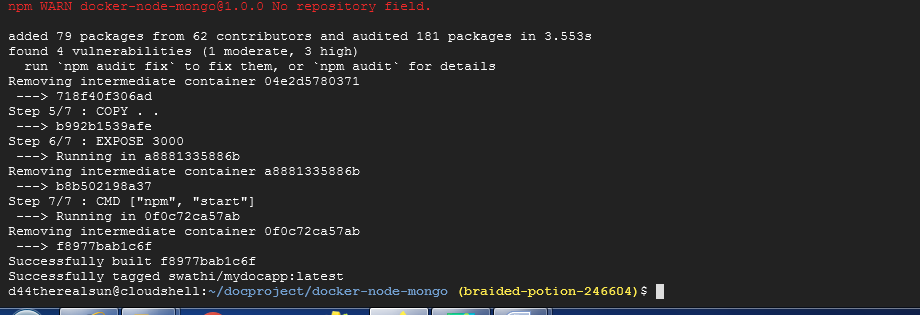
Now we will be creating the docker image for our application.

For that we need docker file to be created.

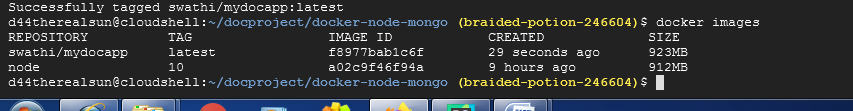


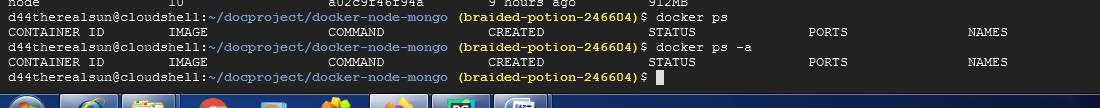
Now we will be executing the commands for building the image of our nodejs application





Now we could see the docker images of our application got created as follows.





We can also run this image of the application which is build by specifying the port numbers in the run command which is called containerization

Command for conternization of the docker image

docker run –d --name anydesiredname -p localmachineport:containerport –d image name

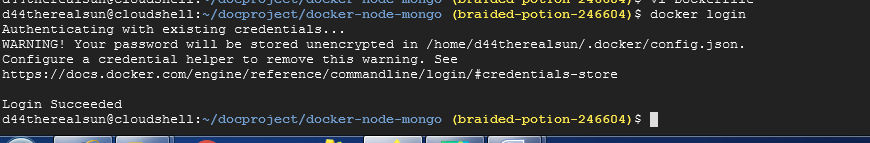
docker run –d --name web –p 2718:3000 –d swathi/mydocapp

Now we will be pushing our docker image to docker hub as follows.

Docker hub is a repository where we will be storing the images of our applications,so that the images of our application is centrally accessible over internet.

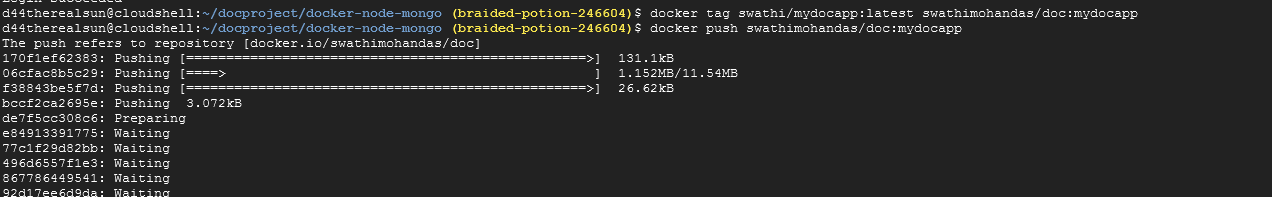
Before that we will be logging into dockerhub account through our cmd and we will be providing the credentials

since mine is already configured hence it got connected.

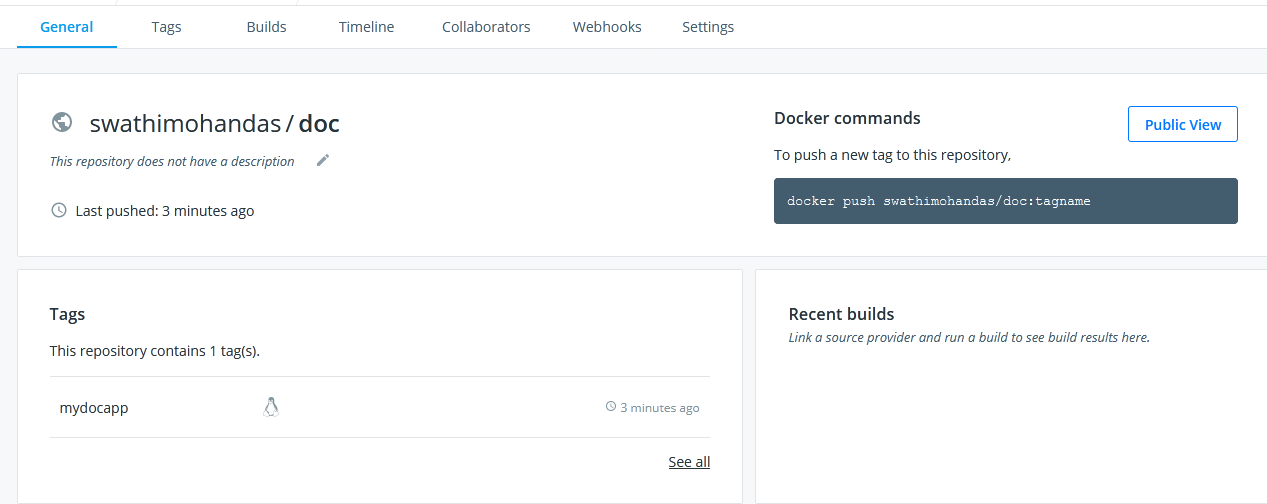


After the login is succesful we need to push the docker image to the dockerhub,so that the image created is centrally accessible from anywhere on the internet.

First we will be tagging our application image and then we will be pushing it into the dockerhub repository

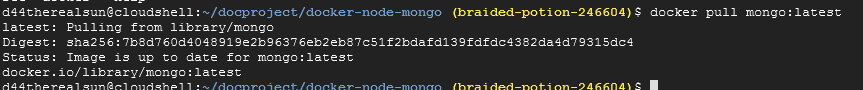


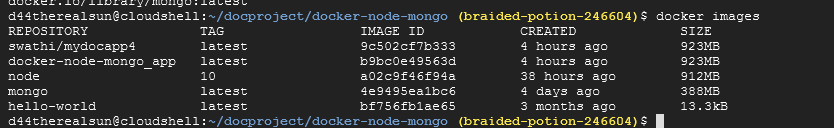
We could see the image pushed into docker hub as follows.



Now we will be running the conternerized version of the mongodb as follows

Firstlty we will pull the mongo image from dockerhub,even if it is not pulled we can run it it will pull automatically from the dockerhub

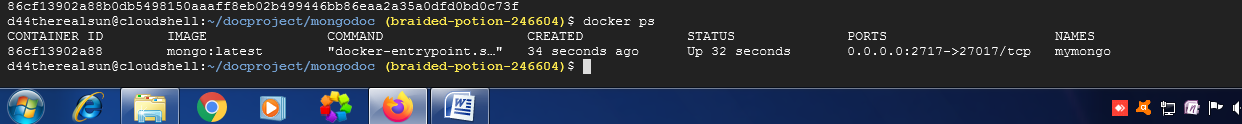




Now we will be containerizing the image and also we will be attaching volumes for the same as follows.

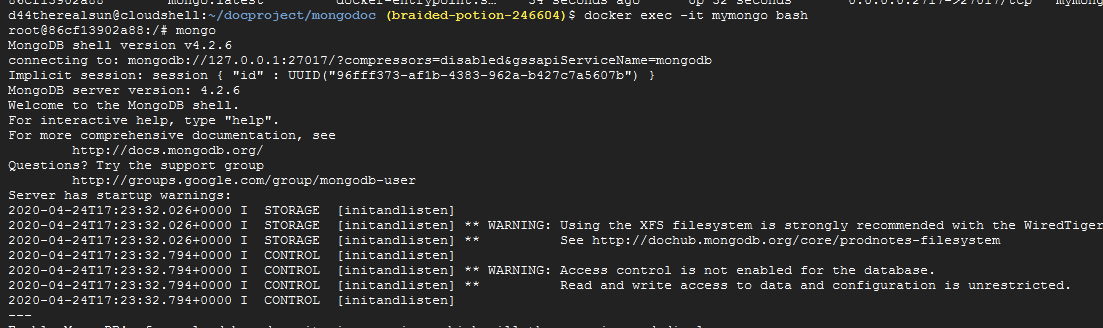
So that the data is stored in the volumes attached and even if the container is deleted the data will be retained

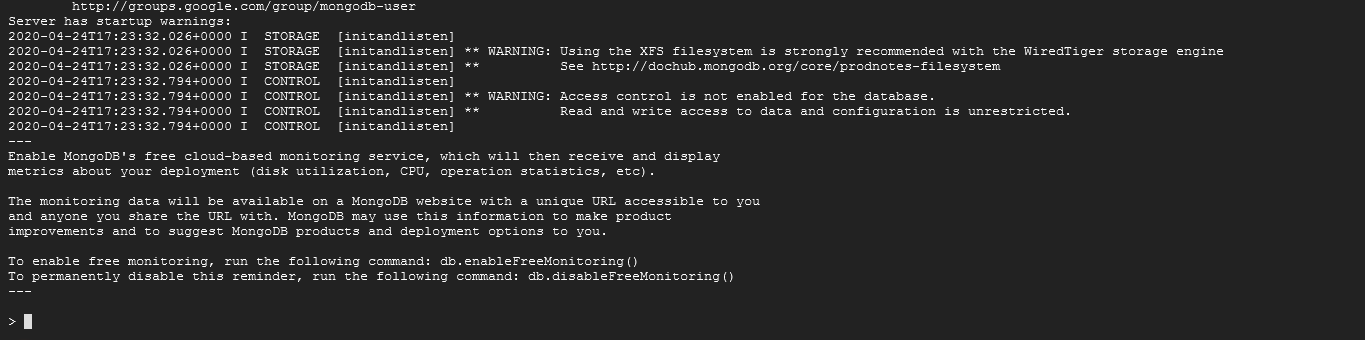


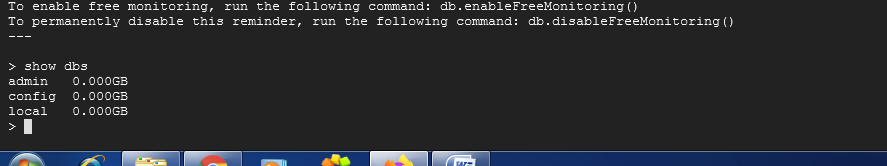


Now if we want to go inside the container then execute below command as follows.

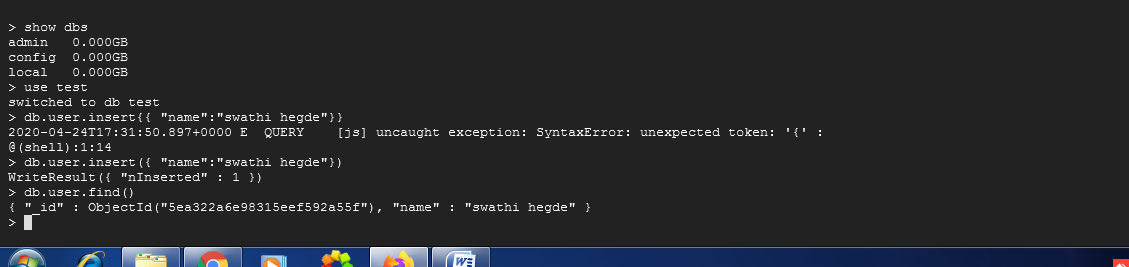
And then enter mongo



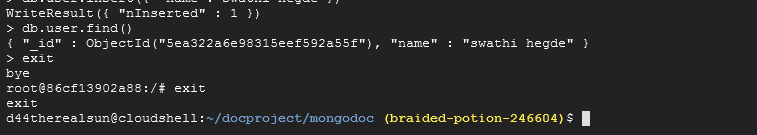




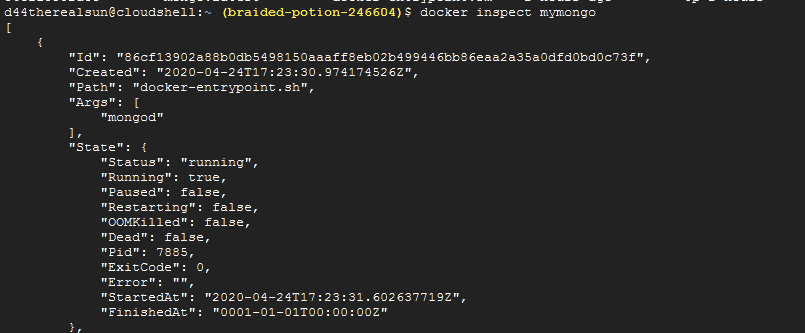
Now we will be creating new db as follows and will add the new user



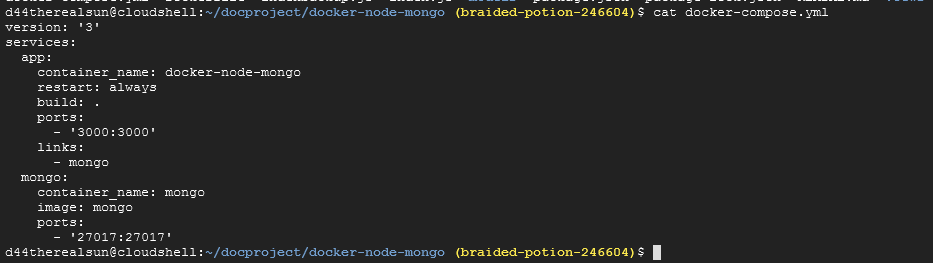
Now exit from the shell



We can inspect the container by the command as follows

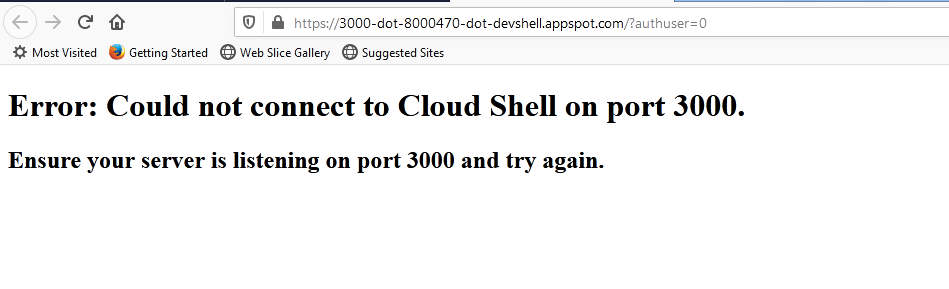


Now we will be writing the script to spin up both node js application container and mongo db container using docker-compose.yml as follows and will be linking both the containers as follows

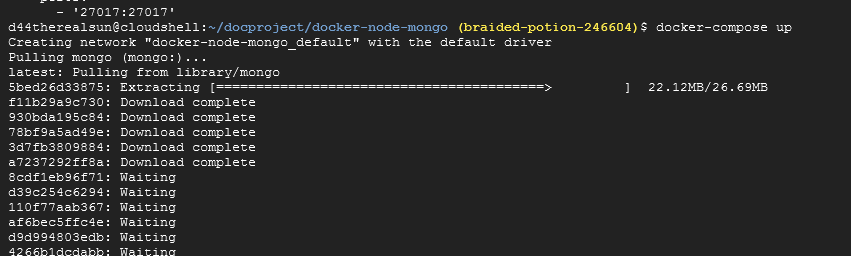


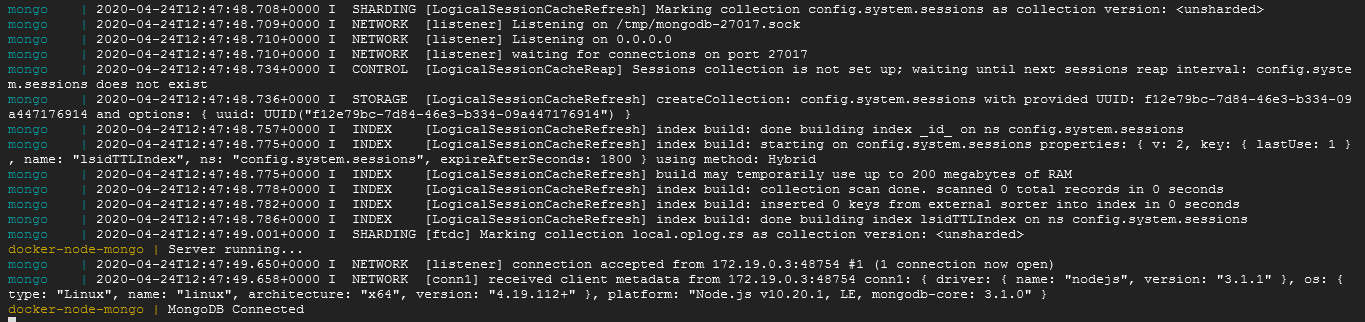
Before running lets check in the browser at port 3000 if anything is already running or not.

We can also check it thru the command prompt



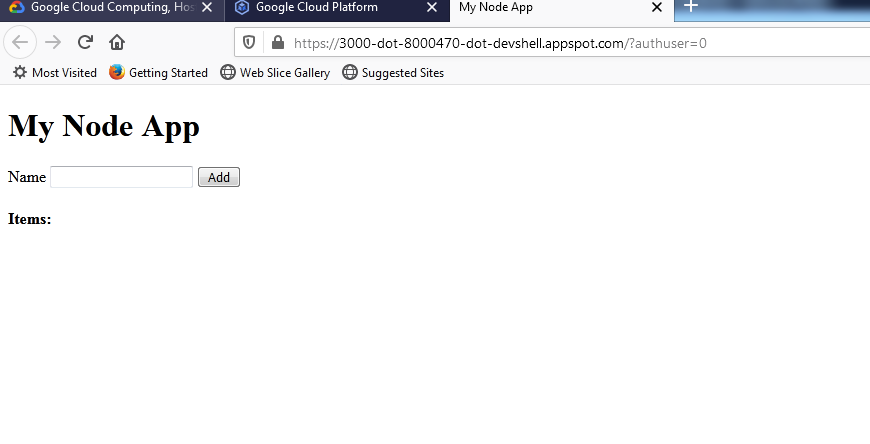
Now we will be bringing up the docker-compose file





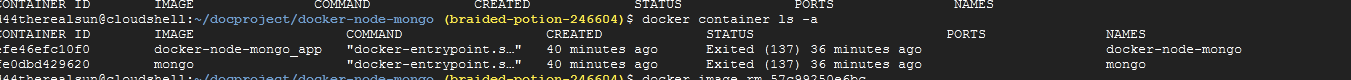
To run the container in background run as docker-compose up -d

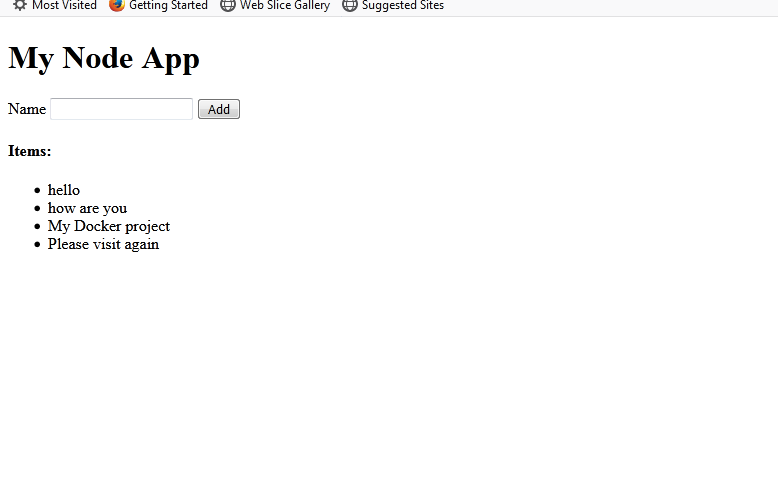
Now just check the browser on port 3000



We could see the application running

And whatever we will be enter the items will get stored in the mongo db.



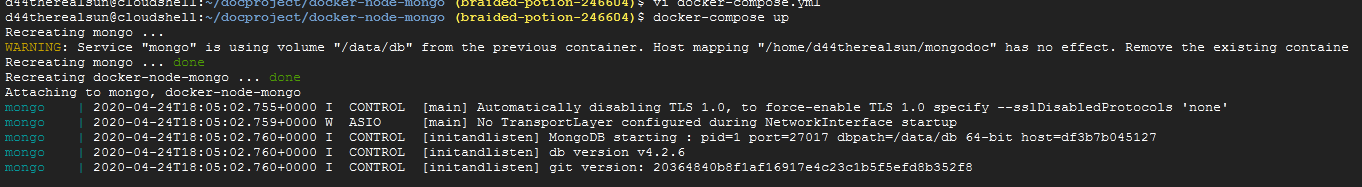


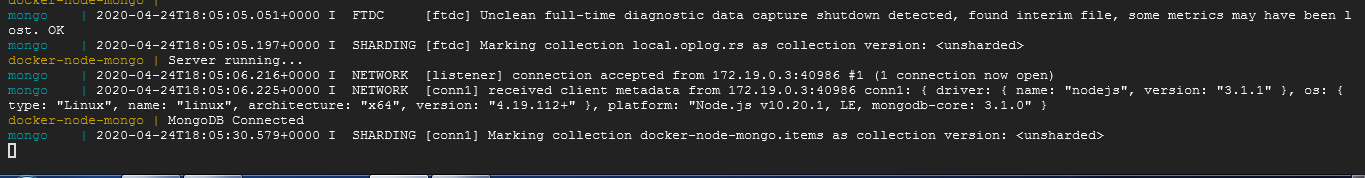
We didn’t attach volume in our previous docker compose file so the data inserted while the container was running will be saved within the container,No other containers can access those data.Also if we delete our container the data will also be lost.

So we will be attaching the volumes so that the data will be stored even if the container is deleted and also other containers can access the data.

Now we will be attaching the same volume which we had attached while running the mongo container individually. (i.e in mongodoc directory)







So in this way we can attach the volumes.