DevOps Assignment - 1

Developing and deploying a Node.js app from Docker to Kubernetes

- Install Node.js and npm
- Install Docker
- Install Kubernetes
- Install Minikube
- Install Kubectl

Step 1:

- Make A Separate Directory And Initialize The Node Application

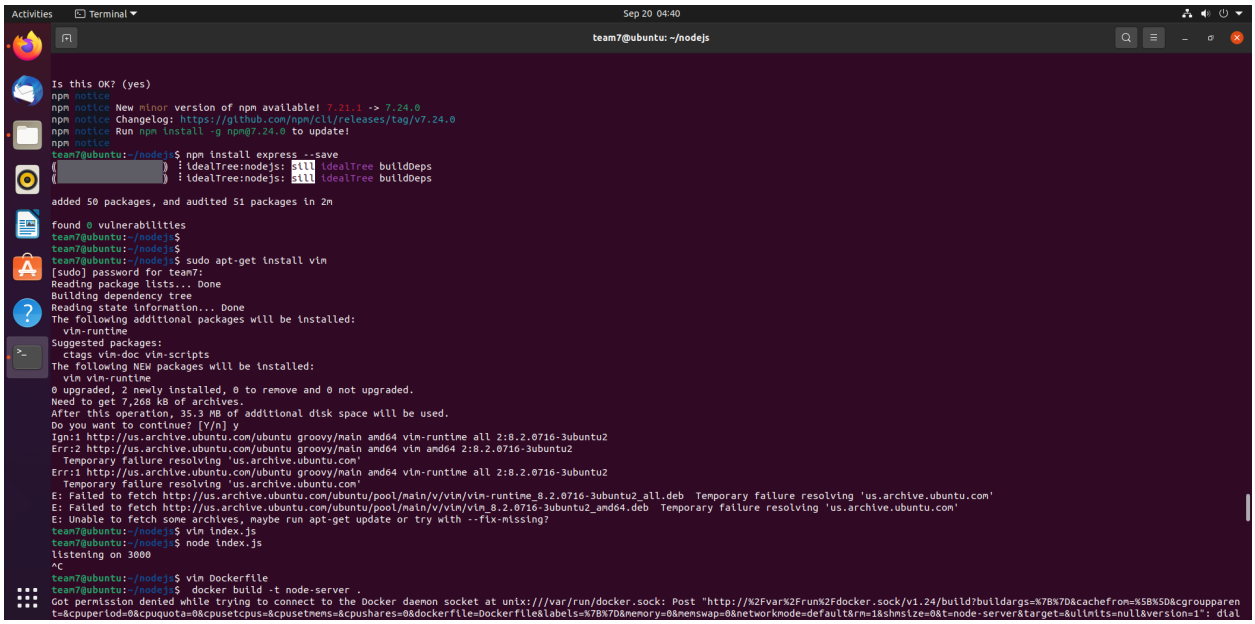
  First, we'll initialize the project with npm (Node Package Manager)

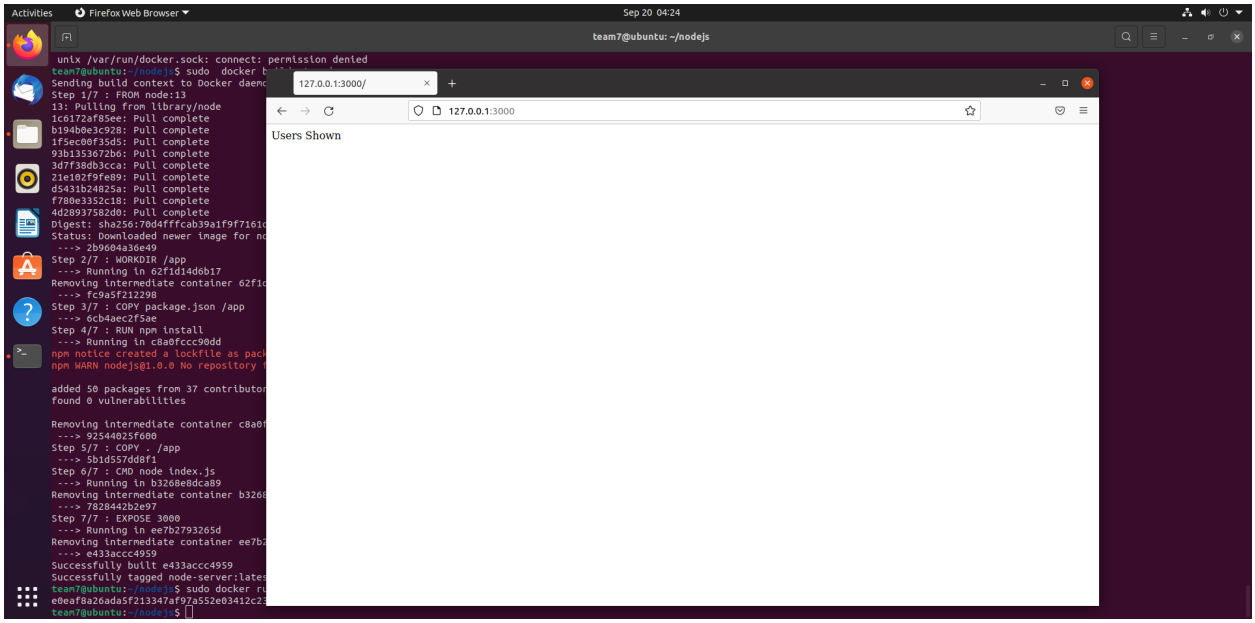
Step 2: Installing Express

Next, we'll install Express through npm (Node Package Manager). The Express framework is used to build a web application and API's:

Step 3: Make index.js File And Write Some Code

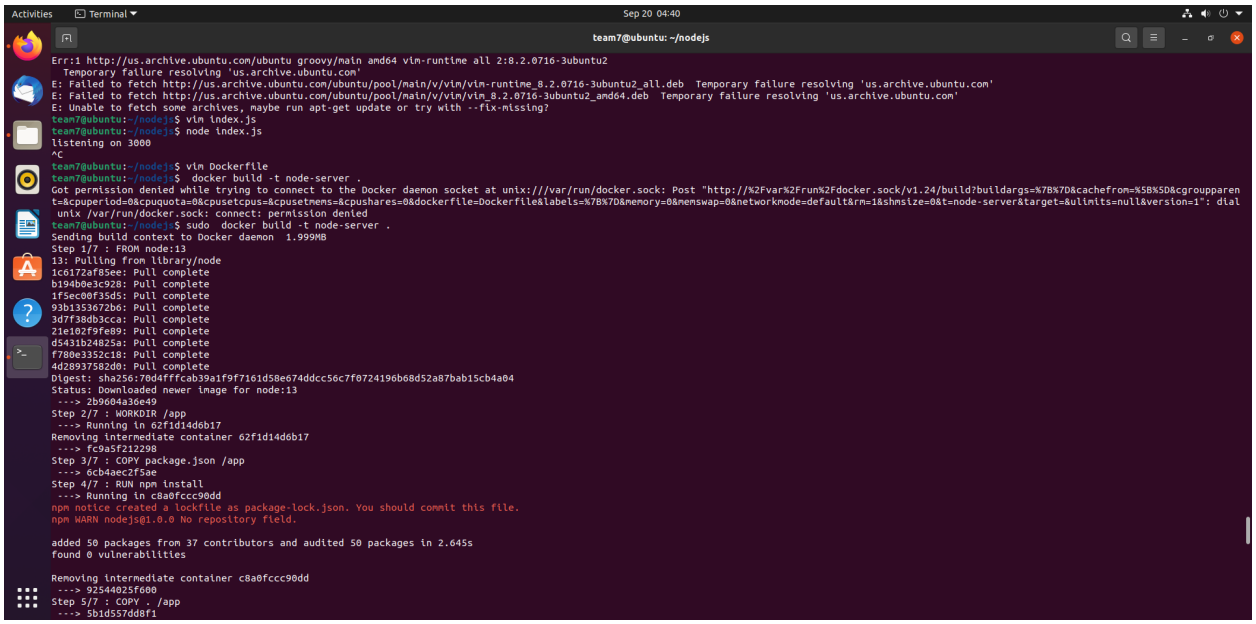
First, create a file named index.js in the root folder. Then we can write some code to test the application on the Kubernetes cluster:


Activities
Terminal
Sep 20 04:40
team7@ubuntu: ~/nodejs
Is this OK? (yes)
npm notice
npm notice New minor version of npm available! 7.21.1 -> 7.24.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.24.0
npm notice Run npm install -g npm@7.24.0 to update!
npm notice
team7@ubuntu:~/nodejs$ npm install express --save
idealTree:nodejs: sill idealTree buildDeps
idealTree:nodejs: sill idealTree buildDeps
added 50 packages, and audited 51 packages in 2m
found 0 vulnerabilities
team7@ubuntu:~/nodejs$
team7@ubuntu:~/nodejs$
team7@ubuntu:~/nodejs$ sudo apt-get install vim
[sudo] password for team7:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
vim-runtime
Suggested packages:
ctags vim-doc vim-scripts
The following NEW packages will be installed:
vim vim-runtime
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,268 kB of archives.
After this operation, 35.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Ign:1 http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vim-runtime all 2:8.2.0716-3ubuntu2
Err:2 http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vim amd64 2:8.2.0716-3ubuntu2
Temporary failure resolving 'us.archive.ubuntu.com'
Err:1 http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vim-runtime all 2:8.2.0716-3ubuntu2
Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/v/vim/vim-runtime_8.2.0716-3ubuntu2_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/v/vim/vim_8.2.0716-3ubuntu2_amd64.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
team7@ubuntu:~/nodejs$ vim index.js
team7@ubuntu:~/nodejs$ node index.js
listening on 3000
^C
team7@ubuntu:~/nodejs$ vim Dockerfile
team7@ubuntu:~/nodejs$ docker build -t node-server .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachefrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=node-server&target=&ulimits=null&version=1": dial


Activities
Firefox Web Browser
Sep 20 04:24
team7@ubuntu: ~/nodejs
unix /var/run/docker.sock: connect: permission denied
team7@ubuntu:~/nodejs$ sudo docker
Sending build context to Docker daemon
Step 1/7 : FROM node:13
13: Pulling from library/node
1c6172af85ee: Pull complete
b194b0e3c928: Pull complete
1f5ec00f35d5: Pull complete
93b1353672b6: Pull complete
3d7f38db3cca: Pull complete
21e102f9fe89: Pull complete
d5431b24825a: Pull complete
f700e3352c18: Pull complete
4d28937502d0: Pull complete
Digest: sha256:70d4fffcab39a1f9f7161d
Status: Downloaded newer image for no
---> 2b9604a36e49
Step 2/7 : WORKDIR /app
---> Running in 62f1d14d6b17
Removing intermediate container 62f1d
---> fc9a5f212298
Step 3/7 : COPY package.json /app
---> 6cb4aec2f5ae
Step 4/7 : RUN npm install
---> Running in c8a0fccc90dd
npm notice created a lockfile as pack
npm WARN nodejs@1.0.0 No repository f
added 50 packages from 37 contributor
found 0 vulnerabilities
Removing intermediate container c8a0f
---> 92544025f600
Step 5/7 : COPY . /app
---> 5b1d557dd8f1
Step 6/7 : CMD node index.js
---> Running in b3268e8dca89
Removing intermediate container b3268
---> 7828442b2e97
Step 7/7 : EXPOSE 3000
---> Running in ee7b2793265d
Removing intermediate container ee7b2
---> e433accc4959
Successfully built e433accc4959
Successfully tagged node-server:lates
team7@ubuntu:~/nodejs$ sudo docker ru
e0eaf8a26ada5f213347af97a552e03412c23
team7@ubuntu:~/nodejs$
127.0.0.1:3000/
127.0.0.1:3000
Users Shown

Step 4: Dockerizing The Node Server

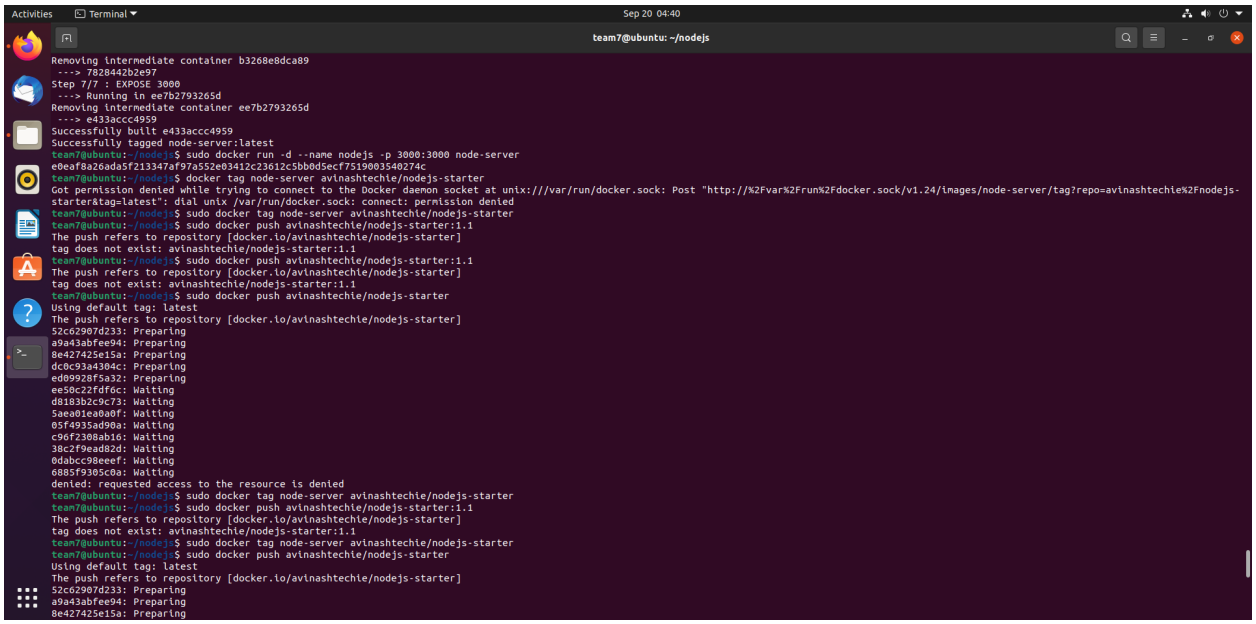
we have the code and the server is ready to deploy. But first, we have to build the image, and for that, we'll have to write the Dockerfile.



The Docker build command is used to create an image with instructions given by Docker-file. -t flag is used to tag the images with our node-server name.

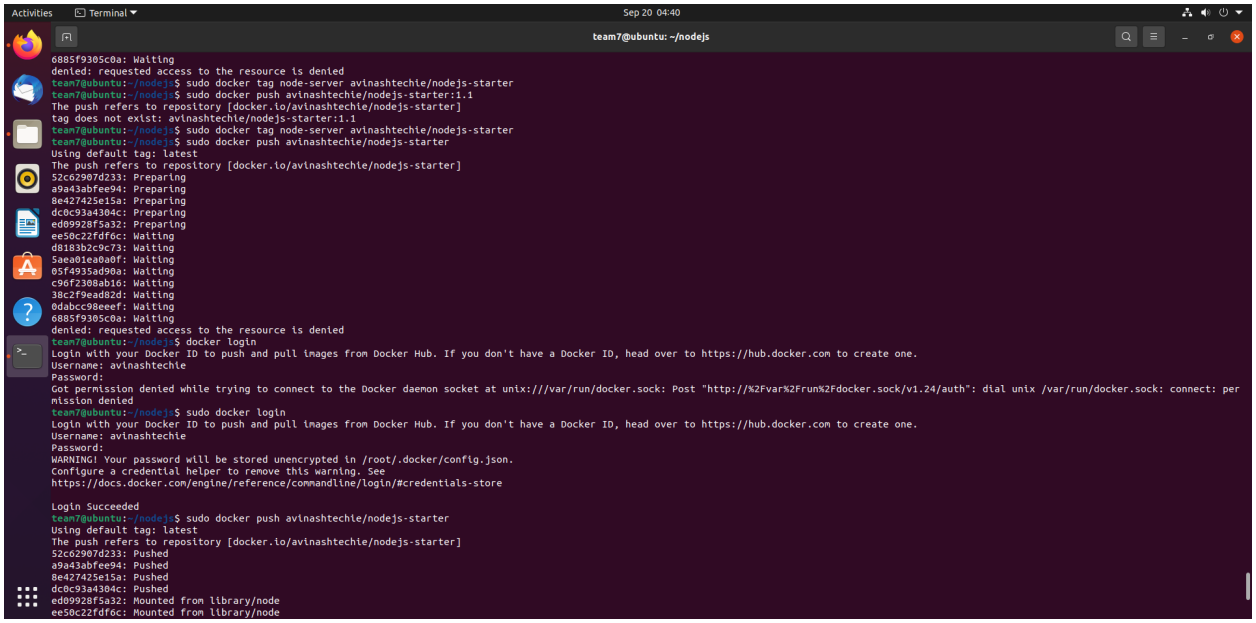
Step 5: Create And Run The Container

we'll then run the container to ensure it works as intended.



Step 6: Upload The Image To Docker Registry Docker Hub

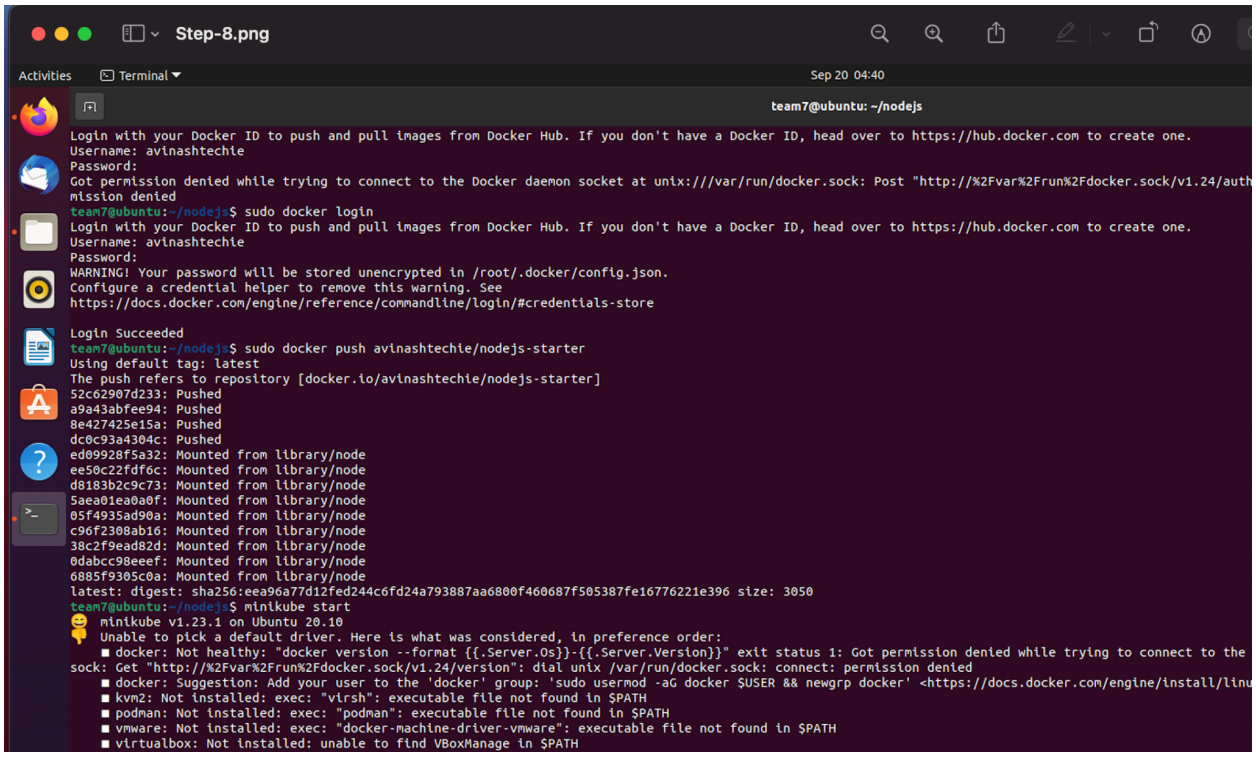
- The image registry that we're using is Docker Hub. First, your account has to be created, then create a repository with any name.



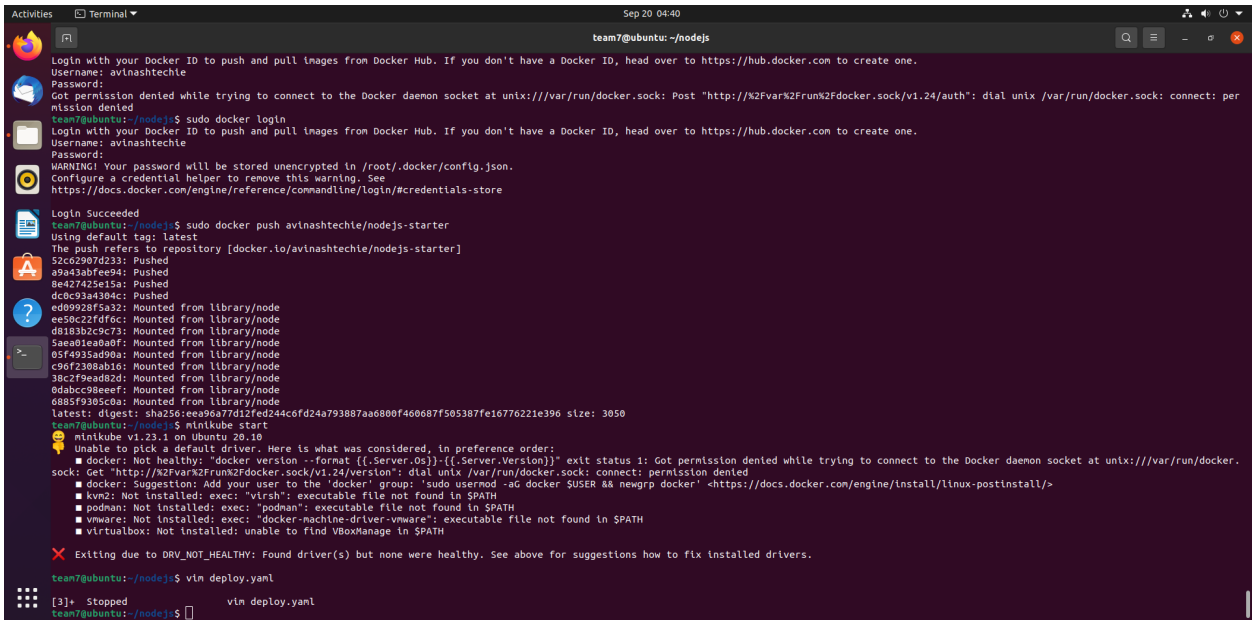
we've pushed our docker image to the registry by using a docker push

Step 7: Start The Kubernetes Cluster

We're are doing this lab on Minikube (used to run Kubernetes locally):

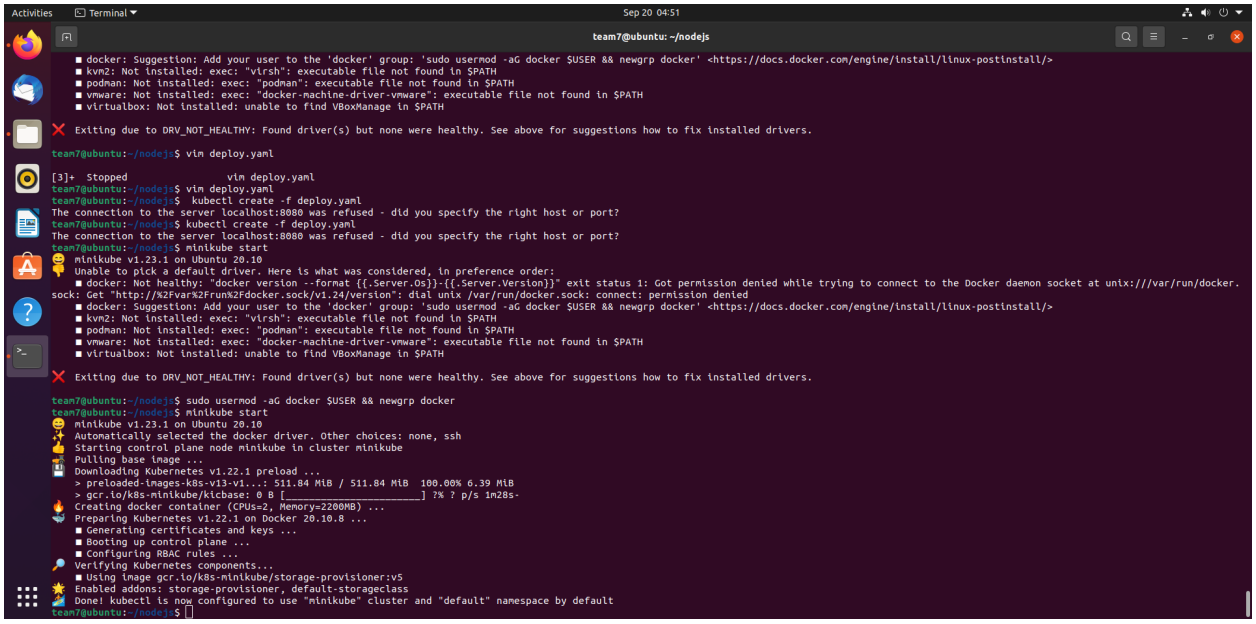


Step 8: Define YAML File To Create A Deployment In Kubernetes Cluster



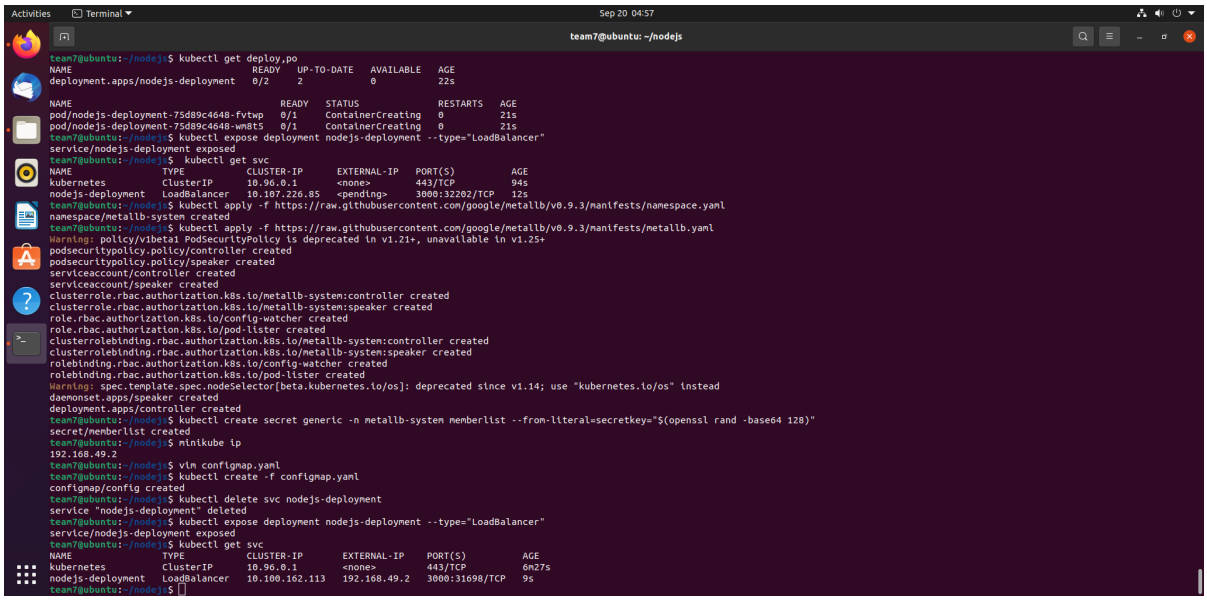
Step 9: Create Deployment In Kubernetes Cluster

As we've created the YAML file, we can go ahead and create a deployment from this YAML file.



Step 10: Expose the deployment to the internet

we're going live through Kubernetes service object



Step 11: Using MetalLB In Your Minikube Environment

we're going live through Kubernetes service object