IMPORTANT COMMANDS

Example SSH credentials:

Private IP: 172.31.27.49

Public IP: 44.206.37.188

Username: sankar

sudo Password: s@nkaR37

using these credentials go to the vscode download extention remote ssh. Then add a config file like below

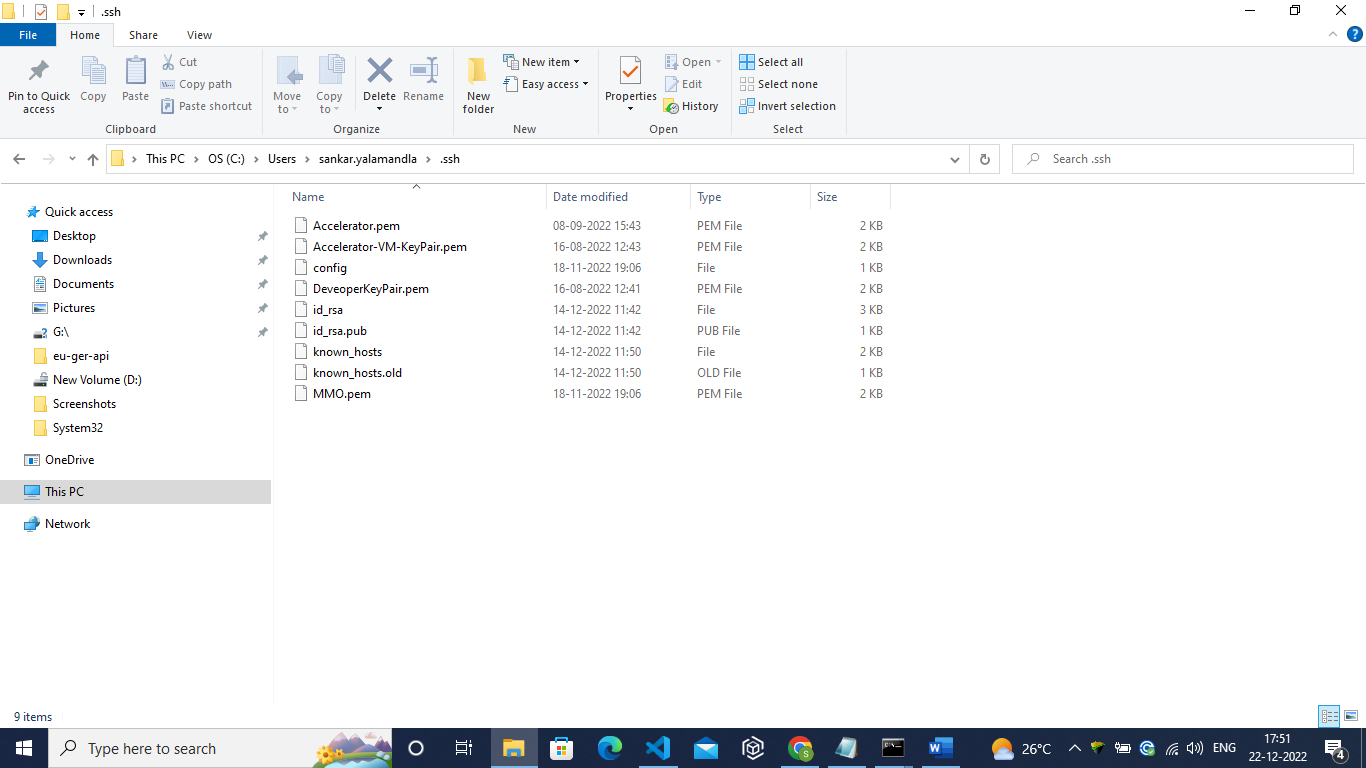
Host MMO\_server

    HostName 44.206.37.188

    User sankar

    IdentityFile ~/.ssh/MMO.pem

Here we need a .pem file or we can convert ppk file to pem using putty. And place that pem file inside .ssh folder



Now we connect in vs code to this remote

Docker Commands

docker-compose build -m 6g frontend

docker-compose down

docker-compose up ====> to up server

sudo docker-compose up -d ====> To run app in background

docker ps ====> running containers

docker stop <container id>

sudo docker system prune –volumes =🡺 remove previous containers

sudo docker system prune -af

sudo docker build --tag testdocker:0.0.1 .

sudo docker run -i -t -d -w /app -p 0.0.0.0:5000:5000/tcp -p 0.0.0.0:3000:3000/tcp --name sankar\_docker testdocker:0.0.1

df -h ==🡺 to check the linux size

<https://dev.to/oneofthedevs/docker-angular-nginx-37e4>

docker build -t price\_backend -f ./Dockerfile .

docker run -p 4300:80 -d price\_frontend

MOST IMPORTANT FRONTEND DOCKER FILE

1. First build app in dev mode then in DockerFile write

### STAGE 1:BUILD ###

# Defining a node image to be used as giving it an alias of "build"

# Which version of Node image to use depends on project dependencies

# This is needed to build and compile our code

# while generating the docker image

FROM node:16.19.0 AS build

# Create a Virtual directory inside the docker image

WORKDIR /dist/src/app

# Copy files to virtual directory

# COPY package.json package-lock.json ./

# Run command in Virtual directory

RUN npm cache clean --force

# RUN npm install -g @angular/cli@latest

# Copy files from local machine to virtual directory in docker image

COPY . .

# RUN npm install --force

# RUN npm run watch

### STAGE 2:RUN ###

# Defining nginx image to be used

FROM nginx:latest AS ngi

# Copying compiled code and nginx config to different folder

# NOTE: This path may change according to your project's output folder

COPY --from=build /dist/src/app/dist/mars-wrigley /usr/share/nginx/html

COPY /nginx.conf  /etc/nginx/conf.d/default.conf

# Exposing a port, here it means that inside the container

# the app will be using Port 80 while running

EXPOSE 80

In .dockerignore

.git

.editorconfig

/.vscode/\*

/node\_modules

/e2e

/docs

.gitignore

\*.zip

In nginx.conf

server {

  listen 80;

  sendfile on;

  default\_type application/octet-stream;

  gzip on;

  gzip\_http\_version 1.1;

  gzip\_disable      "MSIE [1-6]\.";

  gzip\_min\_length   256;

  gzip\_vary         on;

  gzip\_proxied      expired no-cache no-store private auth;

  gzip\_types        text/plain text/css application/json application/javascript application/x-javascript text/xml application/xml application/xml+rss text/javascript;

  gzip\_comp\_level   9;

  root /usr/share/nginx/html;

  location / {

    try\_files $uri $uri/ /index.html =404;

  }

}

example docker-compose.yml

version: '3'

services:

frontend:

image: frontend:latest

container\_name: frontend

build:

context: ./frontend

args:

- SERVER\_NAME=https://accelerators.tigeranalytics.com

- API\_ENDPOINT=http://backend:5000/

restart: always

networks:

- tpo-network

ports:

- 81:80

depends\_on:

- backend

backend:

image: backend:latest

build:

context: ./backend

container\_name: backend

restart: always

networks:

tpo-network:

aliases:

- backend

ports:

- 5000:5000

networks:

tpo-network:

Sudo Commands

sudo su -i

sudo mv <old-name> <new-name> 🡺 rename

sudo cp <source> <destination> 🡺 copy

sudo systemctl restart apache2.service 🡺 restart

sudo systemctl status apache2.service 🡺 status

sudo rm -rf <Folder name>

sudo git init

ps -ef | grep runserver 🡺 to check server

kill -9 <id> 🡺 kill the id

nohup python manage.py runserver 0.0.0.0:9500 & 🡺 example nohup to django

sudo apt install python3.9

sudo apt-get install -y python3-pip

Nodejs install in ec2

curl -fsSL https://deb.nodesource.com/setup\_14.x | sudo -E bash -

sudo apt-get install -y nodejs

Angular build

ng build --aot=false --build-optimizer=false

<https://phoenixnap.com/kb/upgrade-python>

lsb\_release -a 🡺 linux version

* SSH setup url

[Setup ssh](https://www.golinuxcloud.com/set-up-visual-studio-code-remote-ssh-github/%23:~:text=Set%20up%20Visual%20Studio%20Code%20Remote%20SSH%20with,Visual%20Studio%20Code%20remote%20SSH%20with%20GitHub%20)

* increase Watchers for react or angular

echo fs.inotify.max\_user\_watches=524288 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p

* if you face when running npm install

npm config set legacy-peer-deps true

* conda

conda activate <envname>

conda deactivate

conda env list

<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

amit/MMo/source/nohup.out

EXAMPLE FOR CHANGING PERMISSION IN Linux

777 used to give all permissions to user like read, write,

sudo chmod 777 <filename>

To change all files in folder:

sudo chmod -R 777 <path>

To change git permission in folder:

sudo chmod -R 777 .git/

To change permission to readonly:

sudo chmod 0444 <filename>

ls -lha =🡺 check the permission of that folder

<https://www.geeksforgeeks.org/chmod-command-linux/>

To remove staged files

git rm --cached <file>;