What is docker????
→ docker solves very great thing that our code will run on any machine jut create an image and juts use it no problem no developer problem and all that stuffs having here…

Docker run-it <image_name>--> let say create an container and pull this image into this container for using anddocker command and jut ue the containerhere
<if the image is not having then it will download form the hub.docker.com and fetch the particular dependencies and then it will run here >

Container→it is basically to run images..
(the container dont share the data between the container)

Docker container ls→all the running containers are visible here after typing this here
Docker container ls -a→ it will gives us all the container running and all here

Docker start→it will start truely newly container here
Docker stop→it will stop the container

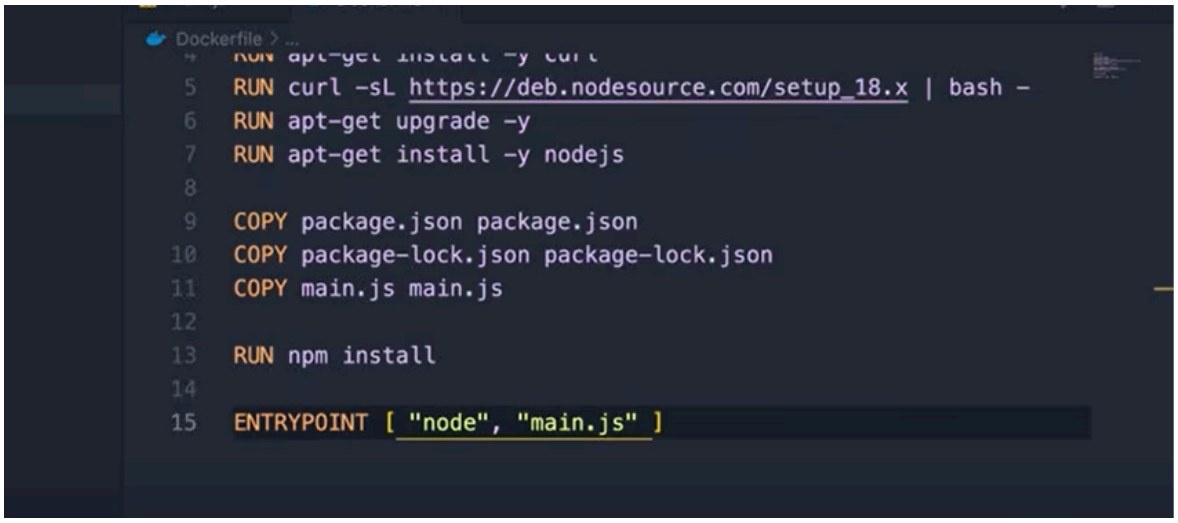
Docker exec -it <container> <command>--->it will do interactive your terminal with the docker terminal here(it islike the isolated container here)

Docker images->it will shows us all the images that are having

Port mapping -> it is for exposing our port outside the container
Docker run -it -p 1025:1025 <external-port>:<internal-port>
→it is exposing containers port 1025 to external port of the system 1025 here

How to pass environment variables into the environments variables here
Docker run -it -e key=value by this way here

How to build an image actually so for that we have to
1)first create Dockerfile into the project-folder
2)



```
RUN apt-get install -y curl
RUN curl -sL https://deb.nodesource.com/setup_18.x | bash -
RUN apt-get upgrade -y
RUN apt-get install -y nodejs

COPY package.json package.json
COPY package-lock.json package-lock.json
COPY main.js main.js

RUN npm install

ENTRYPOINT [ "node", "main.js" ]
```

3)Here are the configurations we have to set-up for the docker here to use it correctly ..
4)then we have to type an command
5)docker build <image-name> . (here dot is basically referencing to the Dockerfile into the given folder<basically-path-of-Dockerfile>)

Now to see what i actually have into the folder structure inside it then we can do this like…
Here is the command for that
→docker exec -it <docker-container-id> bash
->now we can see all the files that are in the docker now..

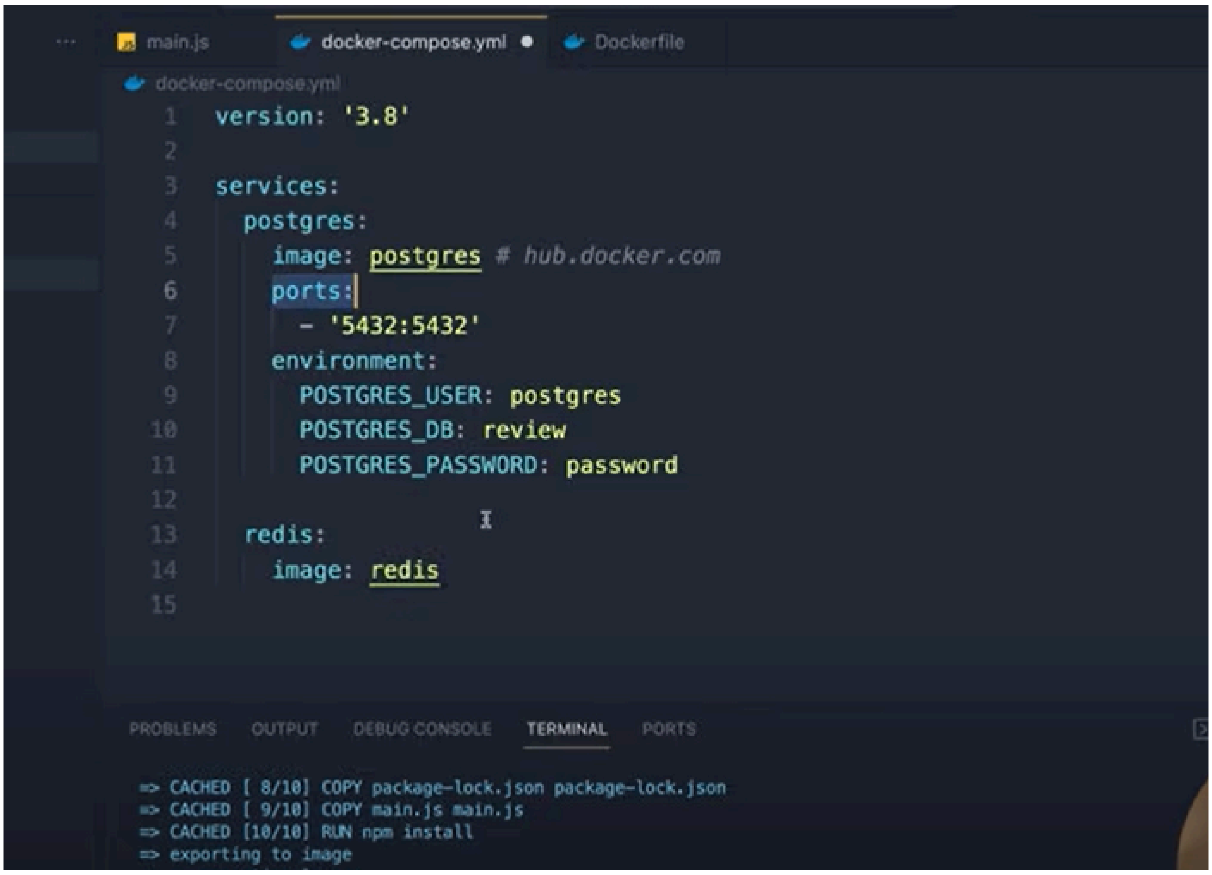
→remember when we have to build any image we have to first take the base image and then we have to work onto that image to build further images there….

→docker is basically caching the things so we have to be aware of how we are doing and to have to put common things at the top only there…

Task
→task is to upload local image into the globally os any one can use
→1) create docker repository and it will provide you the image name
—2)then you have to simply first login into the docker
3)then you have to push that image with the same name
4)now your image will be fully publicly accessed now
5) so now any onecan use and port match and al easily there

Task
→task is like when you have multiple images you want to use into your system then how you can actually use that ???

→1)create- docker-compose.yml <create this file here and then define which services and which ports you have to actually use and which port you want to actually expose here >



```yaml
version: '3.8'

services:
  postgres:
    image: postgres # hub.docker.com
    ports:
      - '5432:5432'
    environment:
      POSTGRES_USER: postgres
      POSTGRES_DB: review
      POSTGRES_PASSWORD: password

  redis:
    image: redis

```

```
=> CACHED [ 8/10] COPY package-lock.json package-lock.json
=> CACHED [ 9/10] COPY main.js main.js
=> CACHED [10/10] RUN npm install
=> exporting to image
```

→3) you have to do up this docker-compose file here
→4)command is here
—>5)docker compose up <this command will helpful to compose and up all the services and now helpful to use that serviceshere >

—>6) now you want to remove and dont want to use the services now then you just have to use this command here <docker compose down> <it will down all the docker services here that is actually running here>

—>7) now you want to run this services as an backend task or services then you just have to use this command <docker compose up -d><detach mode><all services that are useful here running as an backend here >

Docker Networking
→there are many type of network here
→ to communicate with the network
→when we dont select any thing it will take bridge network by default here

Now which command to czech which docker associated with which network basically here…
→docker inspect network <image-name>

Bridge→so bridge is basically connecting our system to the network to actually communicate through
→ the command is for that is
→docker run –it <image-name> –d (it means drive name) bridge

Now to check to howmany type of network basically there
→docker network ls->to czech for the network here..

To give container to the particular network command is
→docker run –it <image-name> –network==<name-of-the-network>

Bridge network vs host network
→bridge network is basically connecting our local machine with the docker network machine basically

→when itis host we dont have to set any port or like something because it is directly having connected to our system actually…

How can we give a container to none network
→ **you have to** type this command basically
→dockerrun –it <image-name> –network=none
→so we cant do nay network operation now here
Ex… ping google.com
–>it will gives us bad address request here

Now to create custom network
→docker network create -d bridge youtube
→ so it is basically having and create you tube network here

Ex.. of using custom network here

1)first create an network
Docker network create -d bridge youtube
Docker network ls
Docker run -it –network=<network-name-which-you-wnat-to-give> –name <container-name> <image-name>

->Docker run -it –network=youtube –name tony_stark ubuntu
->created first container here

2)create second container here to communicate
->docker run -it –network=youtube –name dr_strange ubuntu

3)now we can run any container here and then basically communicate to this two container with each other here

4)we can also do

Ping tony_stark (to go into the dr_strange and after typing this we can actually communicate here we can see that here)

Now we have to mount our data into that particular container then we can do this
→**here is the command for that..**
—>docker run -it -v <your-system-machine-path>:<your-container-machine-path> <container-name>
→so by this command we can actually mount their your data/folder any kind of thing there..

Now what is the mean of mounting
→ so mounting is basically we have to mount ur system data/folder anything to our system
→we dont have to do anything then
→whatever work we are running on our containerit will automatically mount to there location means any file or any kind of folder will mount there by automatically we just dont wnat to do anything there…

We can also create docker volume by default that we have to actually want there ..

Now what i docker cache
→**Docker Cache is basically just means by ordering of configurations that we need**

What is .dockerignore
→it is basically like the .gitignore
→when we done push/pull anything into the github whatever files we don't want to ignore of files to not upload/upload that we are basically define there
→**COPY….**
→**after this command type into the Dockerfile we are just ignoring the file which we dont want to copy there..**

What is docker multi-stage build basically…
→**Docker multi-stage builds allow you to break down the Docker image creation process into multiple stages within a single Dockerfile. Each stage acts as a separate build environment, enabling you to include only the necessary components for your application's runtime in the final image, resulting in smaller, more efficient images**