Use Case for Docker - To get a refresher

Docker allows you to "containerize" a service/installation and deploy them without the complexity of setup

There are many pre-configured containers on the docker hub

An image is file you create with all of the resources of your application

A Container is a running instance of the image

If you pull an image such as a database, you have to create a network to attach it to
Run this command to create your network
docker network create mongo-network

To view your network(s)
docker network ls

docker network rm my-network

To view your images
Docker images

To view logs for any image (to see what the status and what it is doing)
docker logs < container ID-guid of the running container>

For the last set of logs
docker logs <guid of the running container> | tail (tail only works on UNIX)

To view running process/instances
docker ps

To run an image
docker run <image>

To see a list of containers
docker container ls

To stop a container
docker stop <container id>

To delete a container
docker rm <container id>

To delete an image
docker rmi <image id>

To build your own image (At the end of the day, this is what you want to do below)
First Create a file in project, the file must be called
Dockerfile

Below is the contents of a file in a sample Golang web app
FROM golang:1.17

WORKDIR /go/src/app
COPY . .

RUN mkdir /cmd
RUN go build -o /cmd/app /go/src/app/main.go

ENTRYPOINT ["/cmd/app"]

Save the file in your project directory
Issue the and
docker build -t my-app:1.0 .
The :1.0 is the version
The . means current directory

After you run the command, you can go see the app in docker repo by typing at the command prompt
docker images

To run the image
docker run my-app:1.0

To run the image against a port (to be able to browse)
docker run -p 3000:3000 my-app:1.0

Use Case (building my Rock Paper Scissors app)
docker build -t mygoapprockpaper:1.0 .

After I built the image
Docker images

lioneljones@lionels-MacBook-Air ~ % **docker images**
REPOSITORY TAG IMAGE ID CREATED SIZE
mygoapprockpaper 1.0 7aaa03f6baa5 42 seconds ago 813MB
lioneljones@lionels-MacBook-Air ~ %

Now that I have an image, I am going to **containerize** it
docker run -p 8085:8085 mygoapprockpaper:1.0

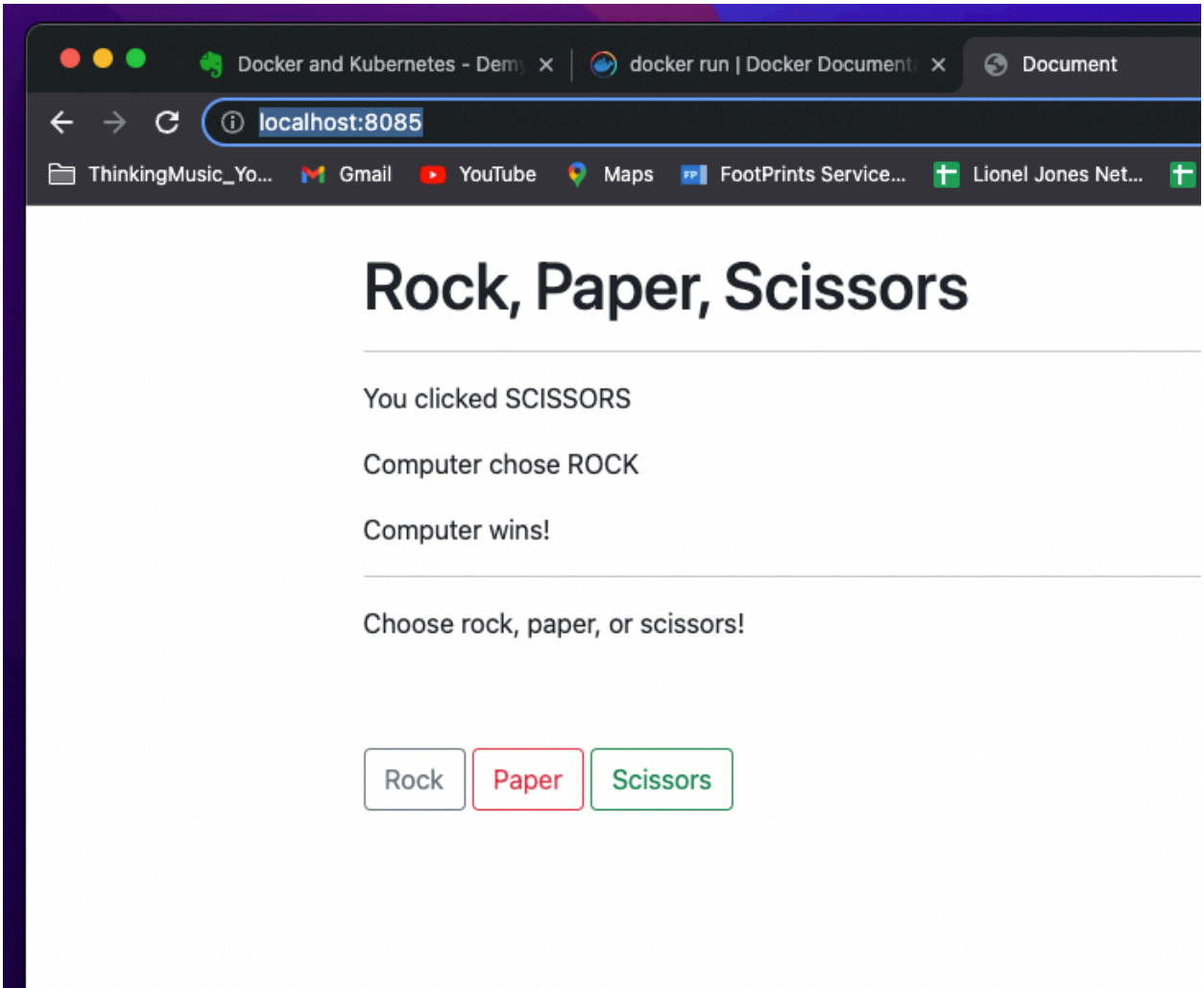
lioneljones@lionels-MacBook-Air ~ % **docker run -p 8085:8085 mygoapprockpaper:1.0**
2022/03/23 01:43:42 Starting web server on port 8085

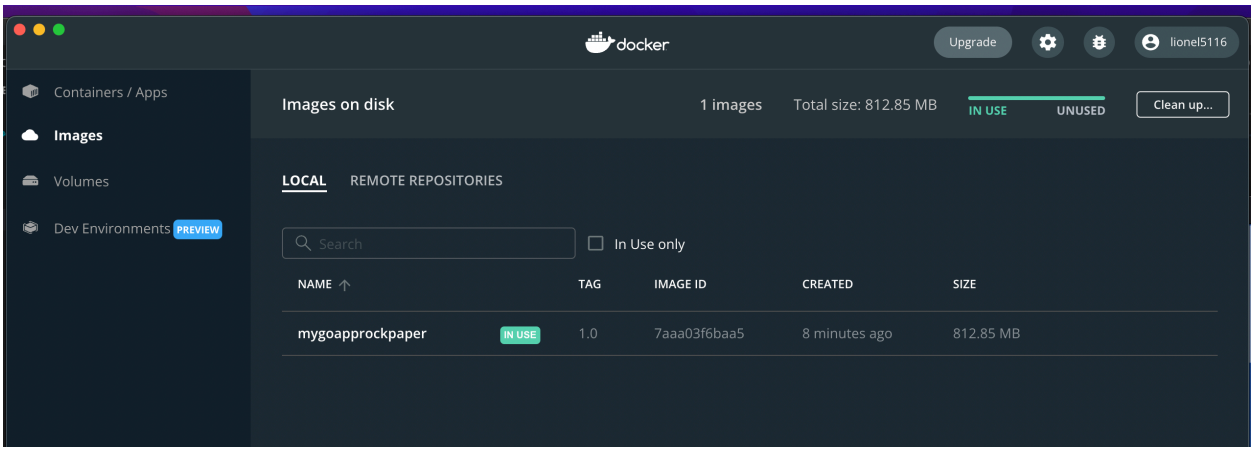
I ran the commands below to view statuses

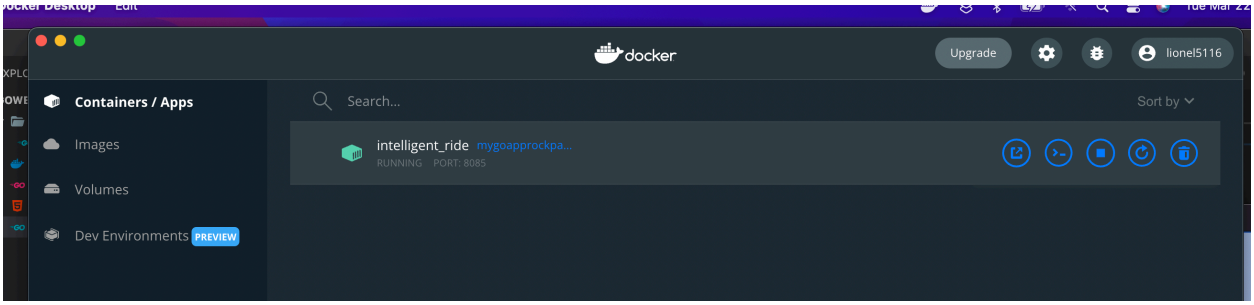
```
Last login: Tue Mar 22 20:04:23 on ttys003
lioneljones@lionels-MacBook-Air ~ % docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
mygoapprockpaper    1.0       7aaa03f6baa5   9 minutes ago    813MB

lioneljones@lionels-MacBook-Air ~ % docker ps
CONTAINER ID   IMAGE                  COMMAND      CREATED              STATUS              PORTS
NAMES
0e3775d495d5   mygoapprockpaper:1.0   "/cmd/app"   About a minute ago   Up About a
minute   0.0.0.0:8085->8085/tcp   intelligent_ride

lioneljones@lionels-MacBook-Air ~ % docker container ls
CONTAINER ID   IMAGE                  COMMAND      CREATED              STATUS              PORTS
NAMES
0e3775d495d5   mygoapprockpaper:1.0   "/cmd/app"   About a minute ago   Up About a
minute   0.0.0.0:8085->8085/tcp   intelligent_ride
lioneljones@lionels-MacBook-Air ~ %
```

As you can see above, after I did a docker run ..., it created a container for my image to run on and the port
Was forwarded to 8085

I enter the URL
http://localhost:8085/
And the site comes up

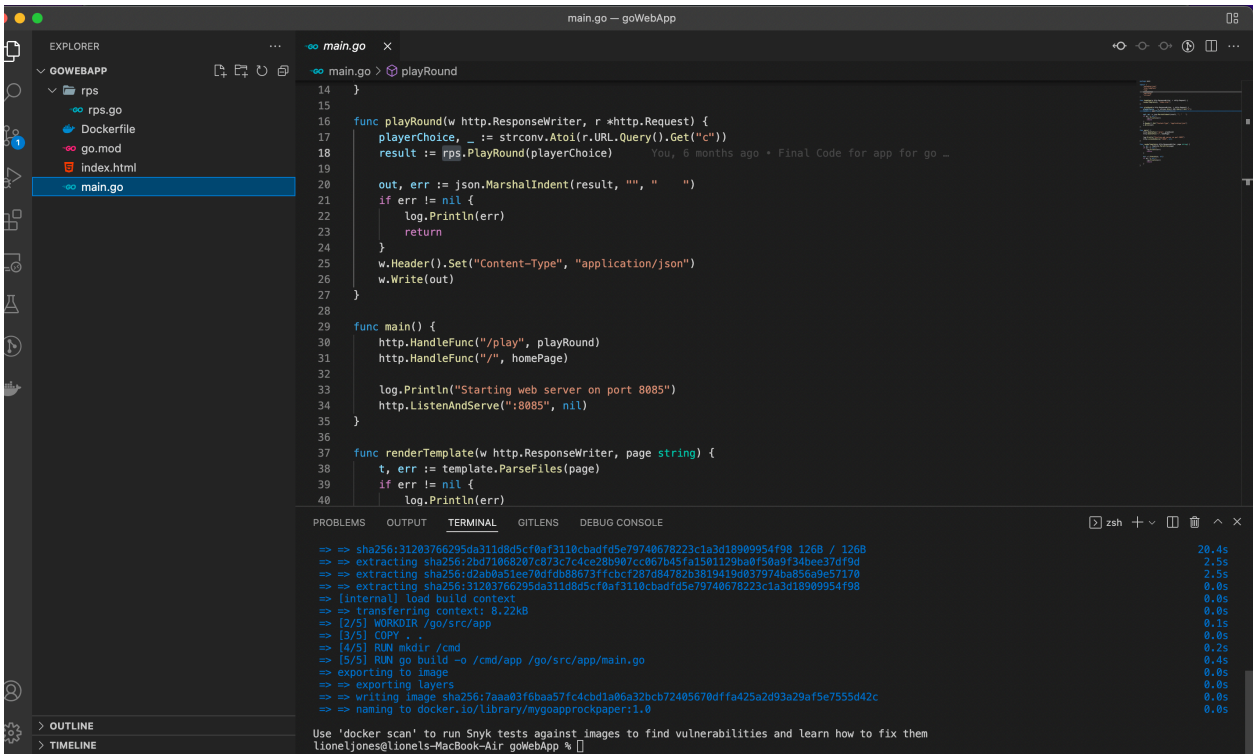






You can view the GUI above to see the statuses of your image/container

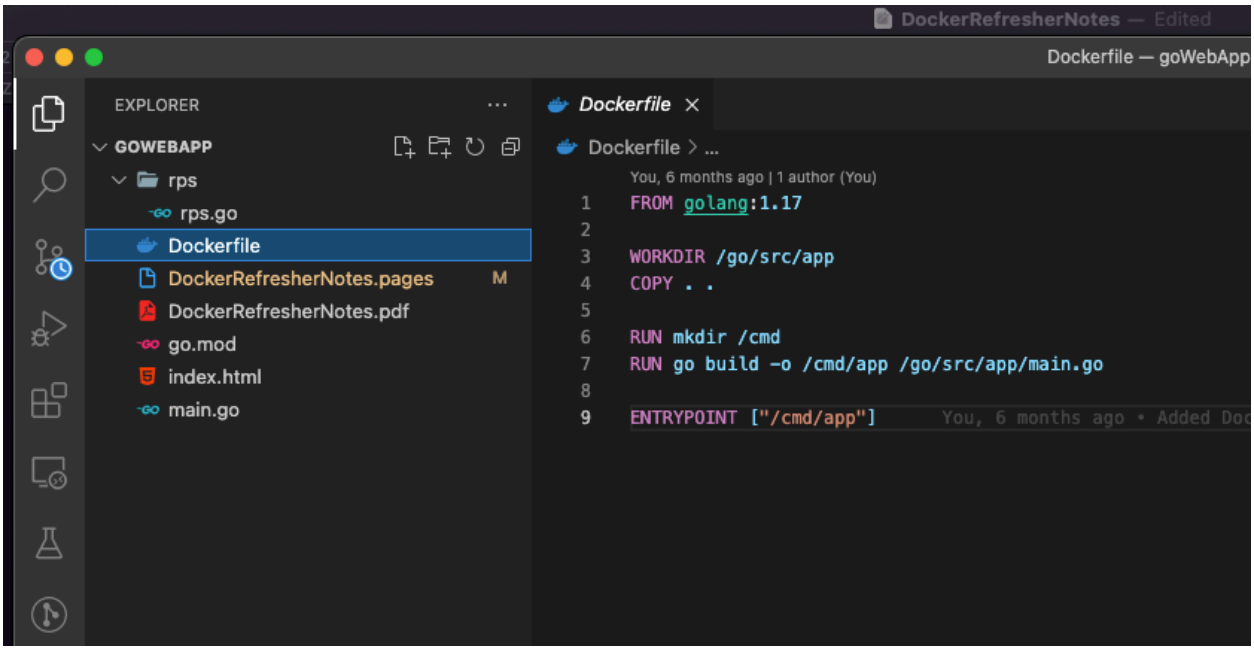
So in a nutshell:
I created a sample Go Project (From Udemy Course)
I created a docker file



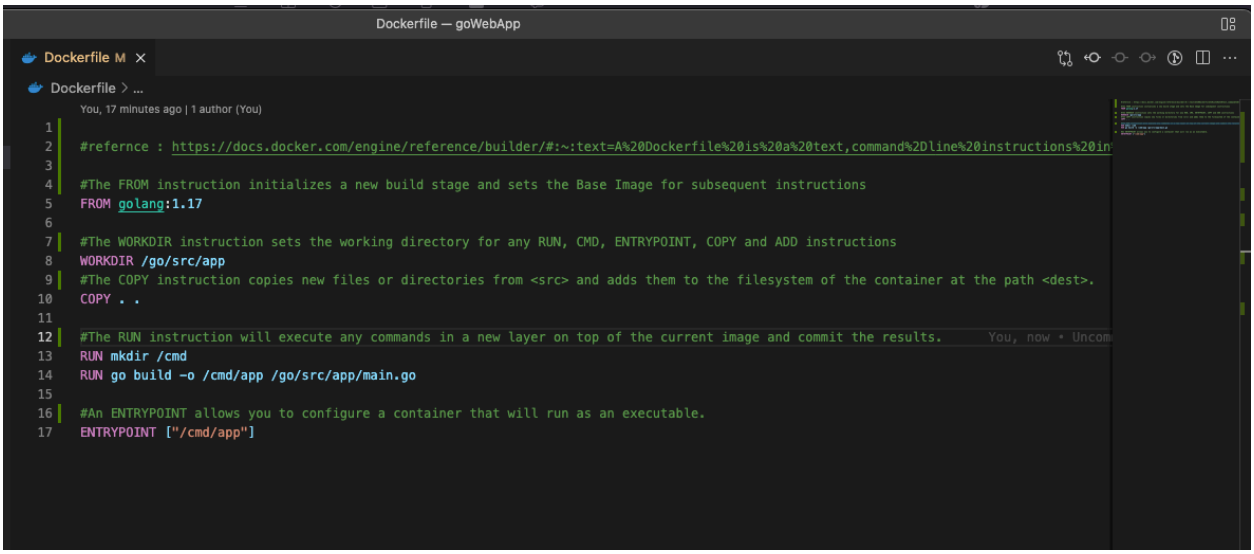
I run docker command to create an image
I ran a command to run the image in a container
I was able to browse the application through the docker image
Project Name:
GoWebApp

Explanation of the Docker File

The explanation of the file below:
The WORKDIR:
This is the directory that your build files will be copied to after your project is built
The COPY ..
This means copy all of the build files to the working directory
The RUN
This runs a command - mkdir /cmd
RUN go ..
This issues the go run command to build the project, -o output the build files to the working directory you specified earlier in the file, and the build file is main.go



https://docs.docker.com/engine/reference/builder/#:~:text=A%20Dockerfile%20is%20a%20text,command%2Dline%20instructions%20in%20succession.
From the Docs (the link above)


Dockerfile — goWebApp
Dockerfile M
Dockerfile > ...
You, 17 minutes ago | 1 author (You)
#refernce : https://docs.docker.com/engine/reference/builder/#:~:text=A%20Dockerfile%20is%20a%20text,command%2Dline%20instructions%20in
#The FROM instruction initializes a new build stage and sets the Base Image for subsequent instructions
FROM golang:1.17
#The WORKDIR instruction sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions
WORKDIR /go/src/app
#The COPY instruction copies new files or directories from <src> and adds them to the filesystem of the container at the path <dest>.
COPY . .
#The RUN instruction will execute any commands in a new layer on top of the current image and commit the results.
You, now • Uncom
RUN mkdir /cmd
RUN go build -o /cmd/app /go/src/app/main.go
#An ENTRYPOINT allows you to configure a container that will run as an executable.
ENTRYPOINT ["/cmd/app"]