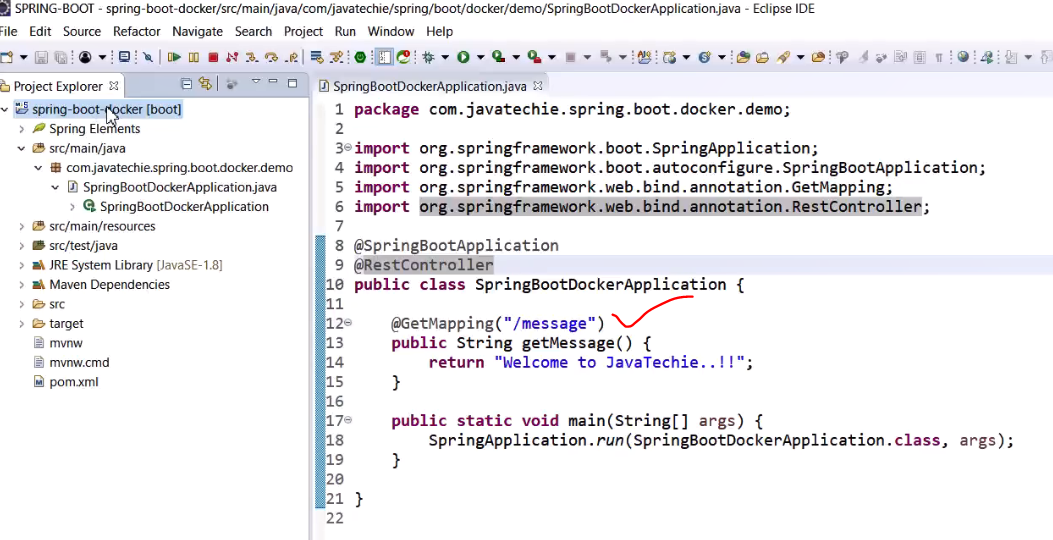
Docker Image Creation Steps

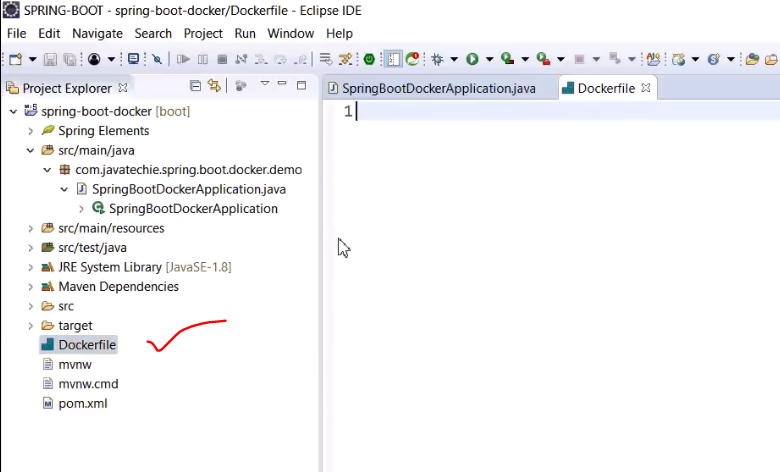
Contents

[Dockerize Spring Boot Microservice 1](#_Toc103093428)

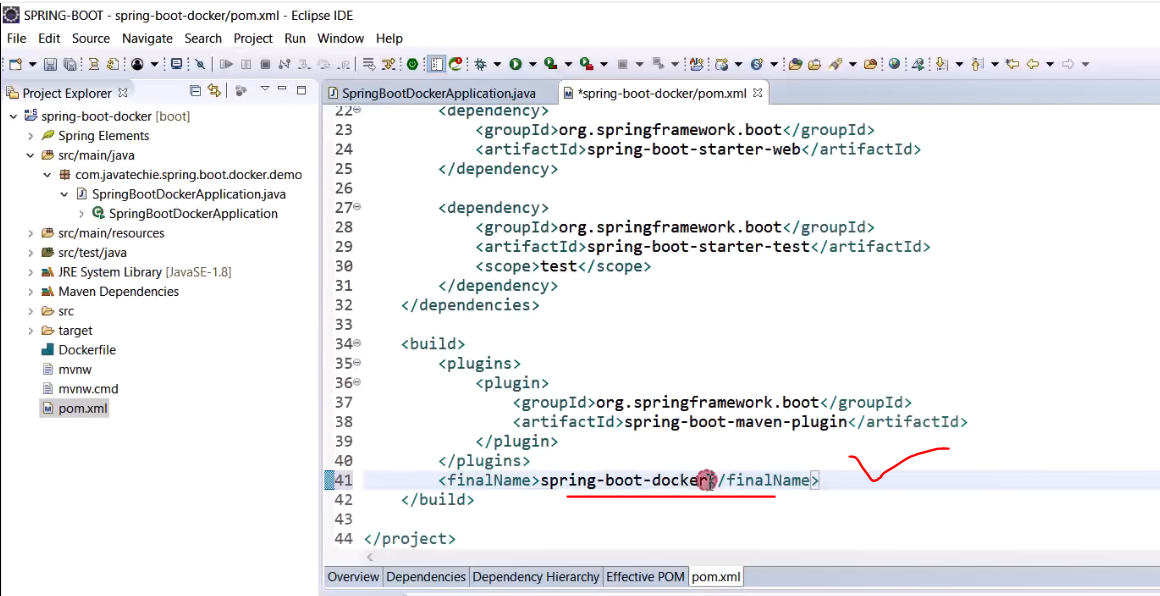
[Dockerize Angular Application 6](#_Toc103093429)

# Dockerize Spring Boot Microservice

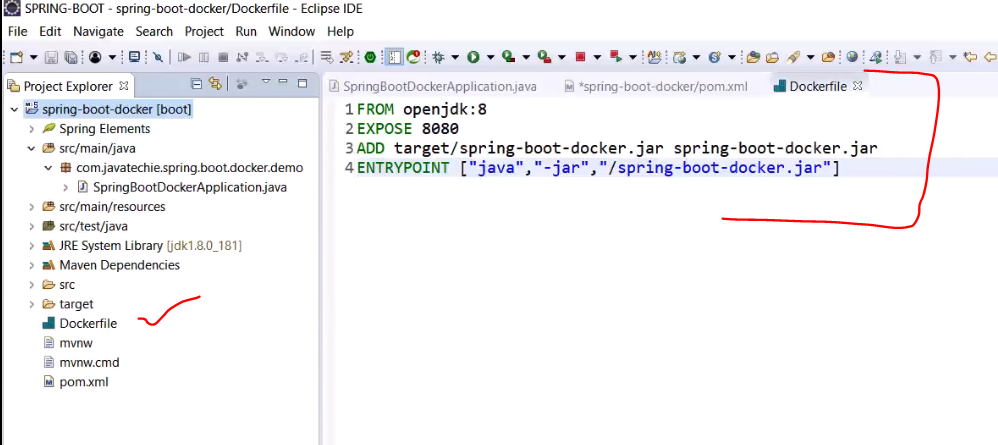




Create a new file “Dockerfile”, with D capital.



Specify the final jar name here.

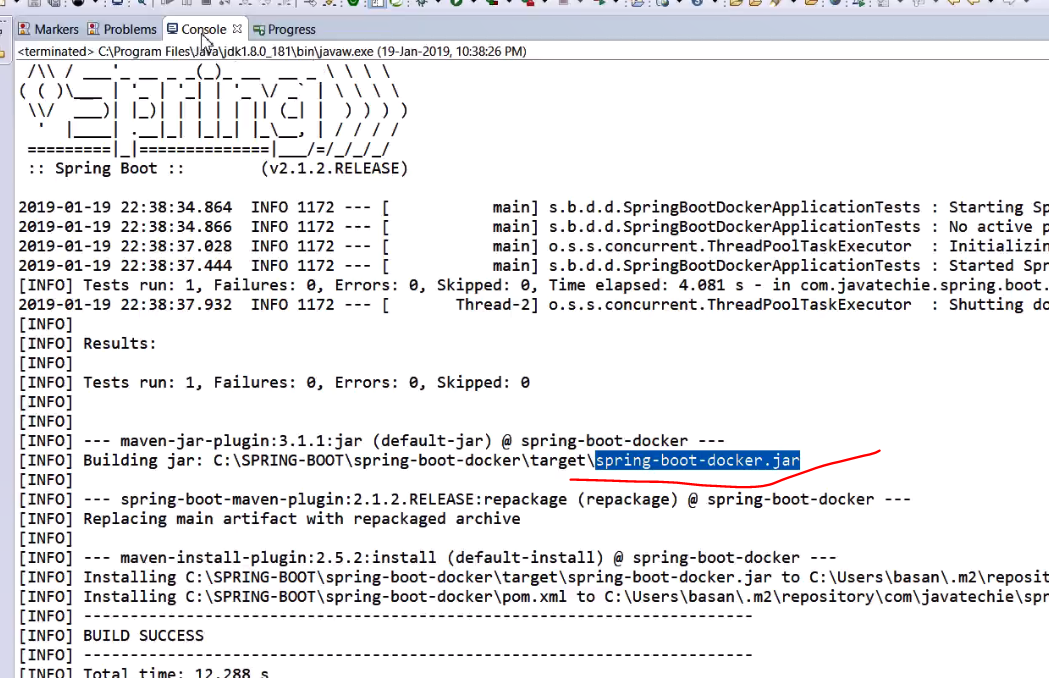


FROM – what is the project based on

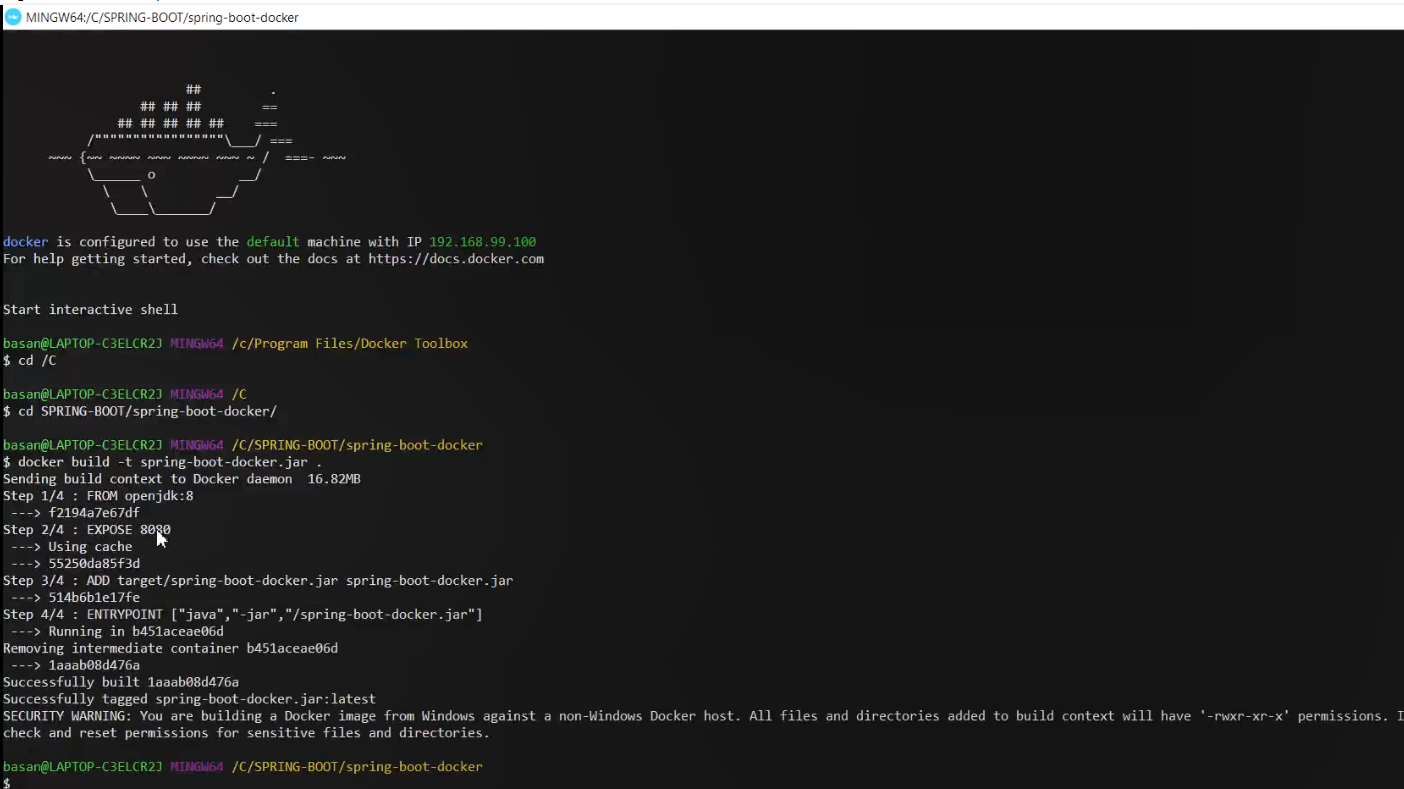
EXPOSE – the local port to expose

ADD – add the jar from the target folder after the build

ENTRYPOINT – what command needs to run to start the application



Build the jar.

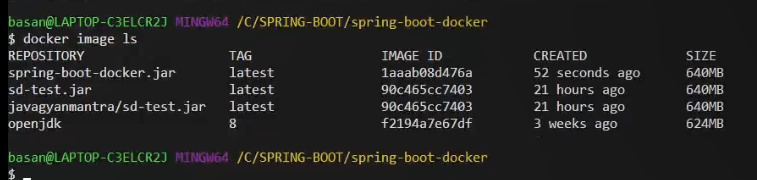


Command:

Docker build -t <project.jar> .

Here as you can observe it ran all the 4 steps one by one as mentioned in our Dockerfile.

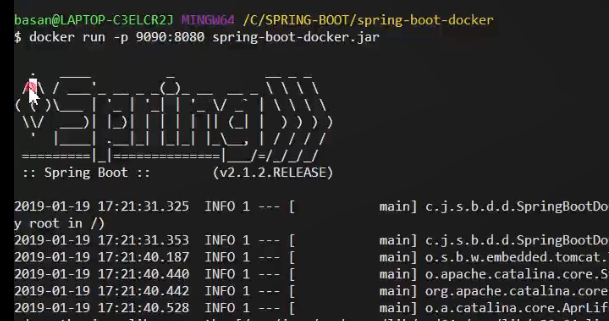
Now check the image



Command:

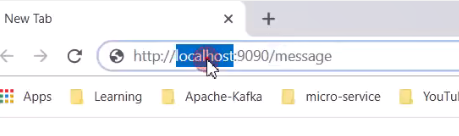
Docker image ls

Next, we will run the docker image.

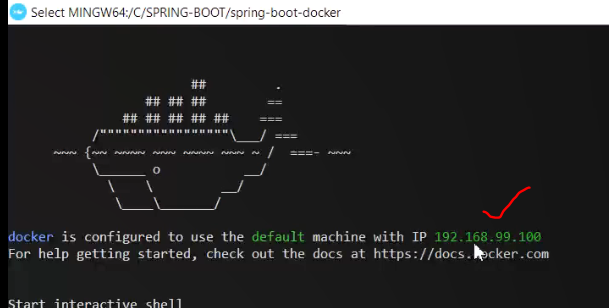


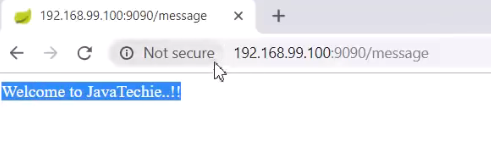
Command:

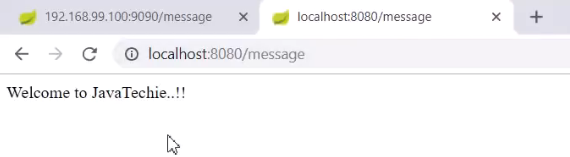
Docker run -p <container port>:<local port as mentioned in EXPOSE> <docker image>



Instead of localhost, mention the IP address.







**Note:**

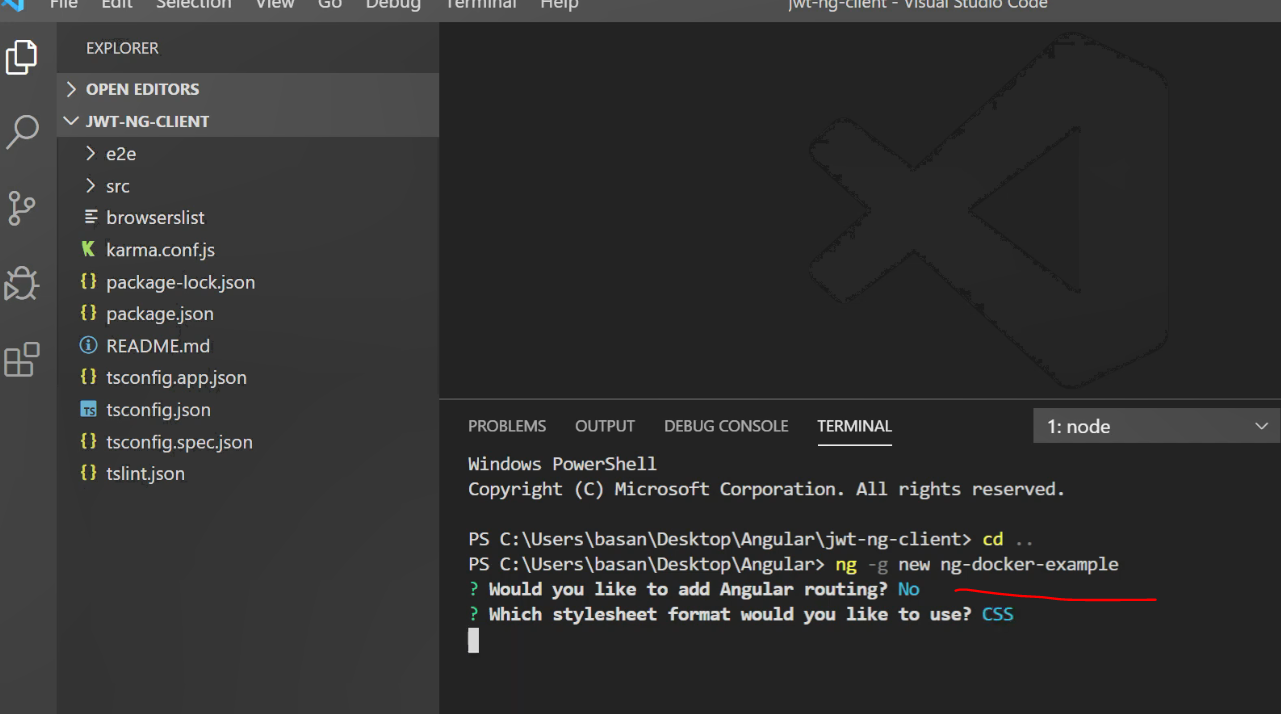
Here localhost will work.

Done!

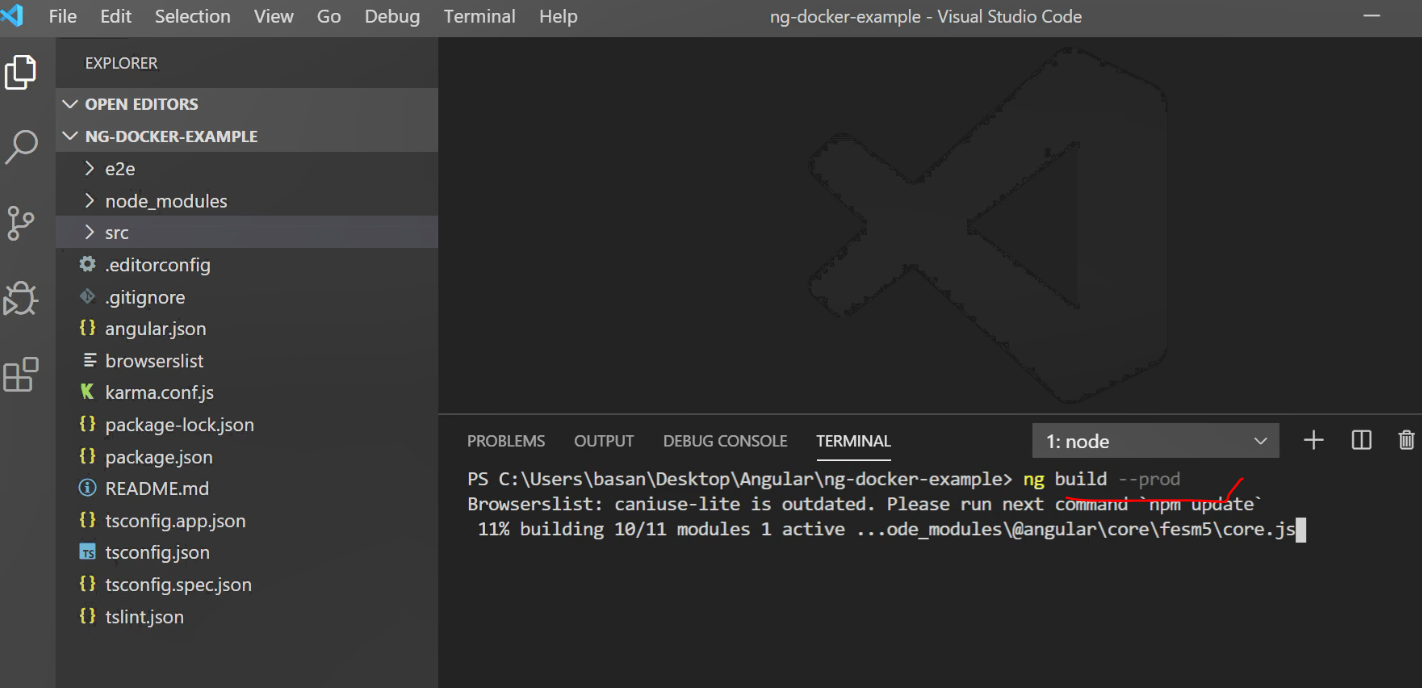
# Dockerize Angular Application

It’s the same process for all applications, just the Dockerfile will be different.

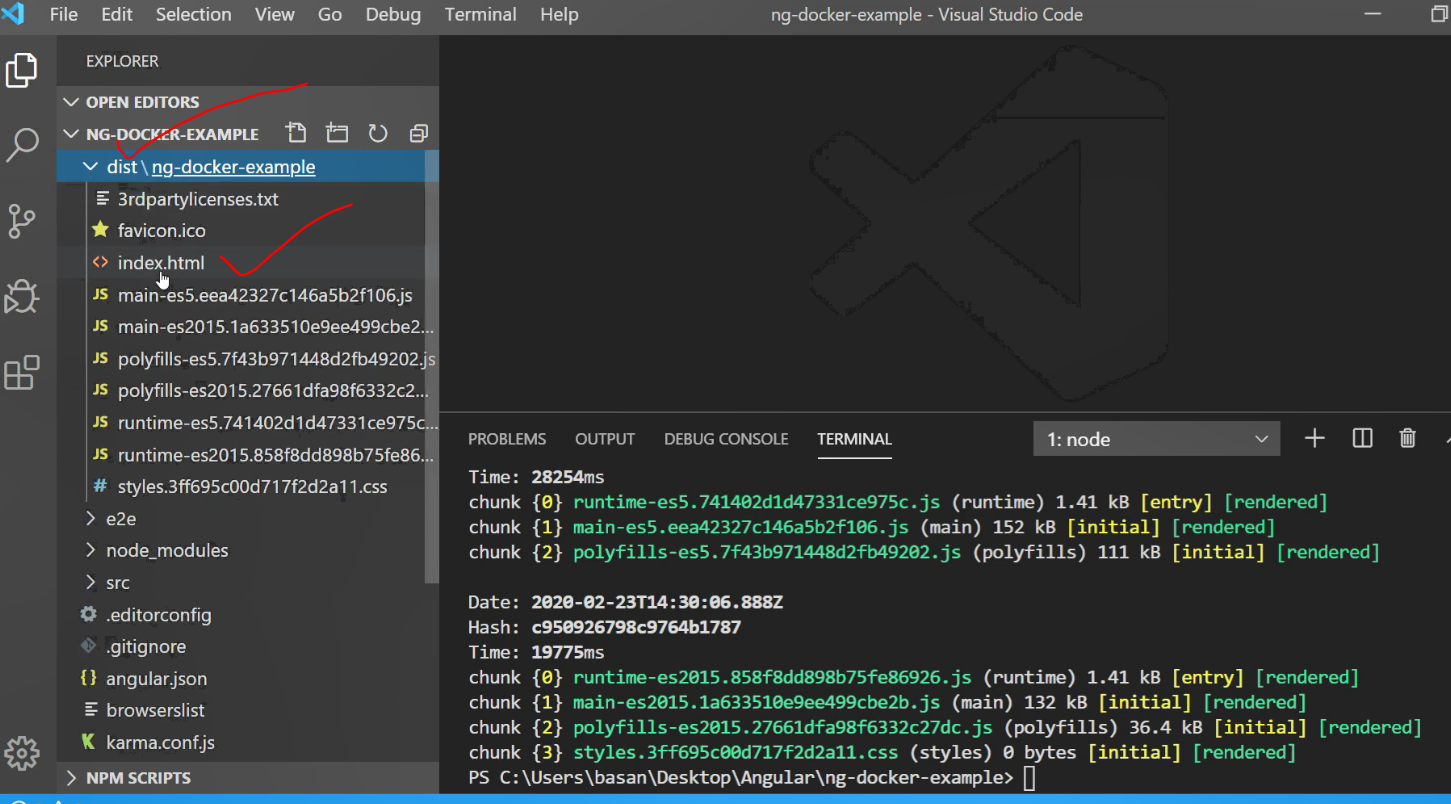
Create the project.



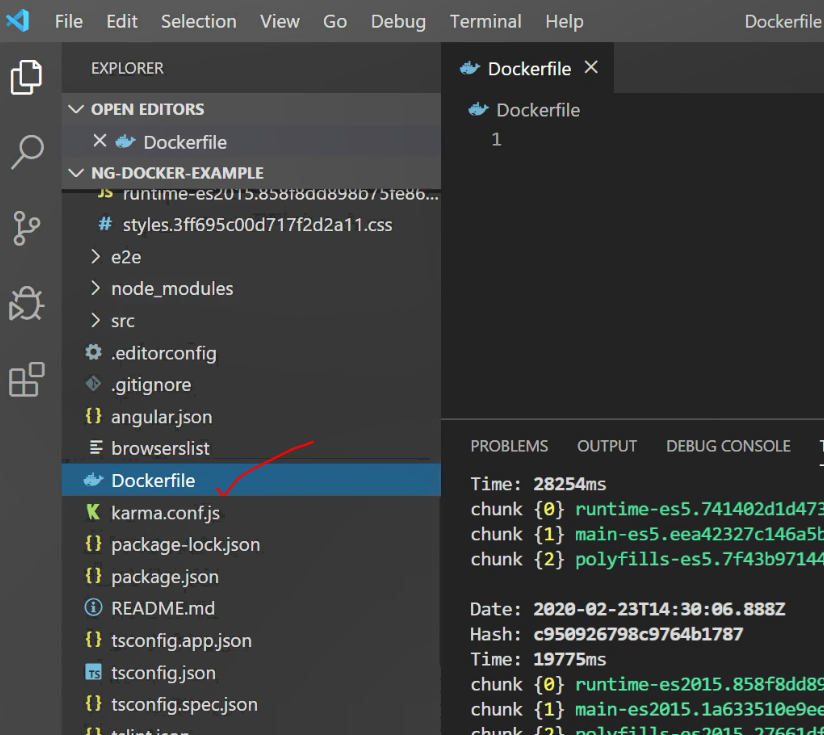
Build the project:

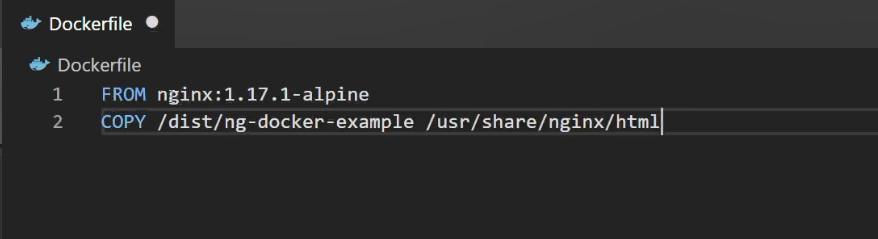


All the compiled files are here:



Create the Dockerfile in the root directory: Note: “D” should be capital.





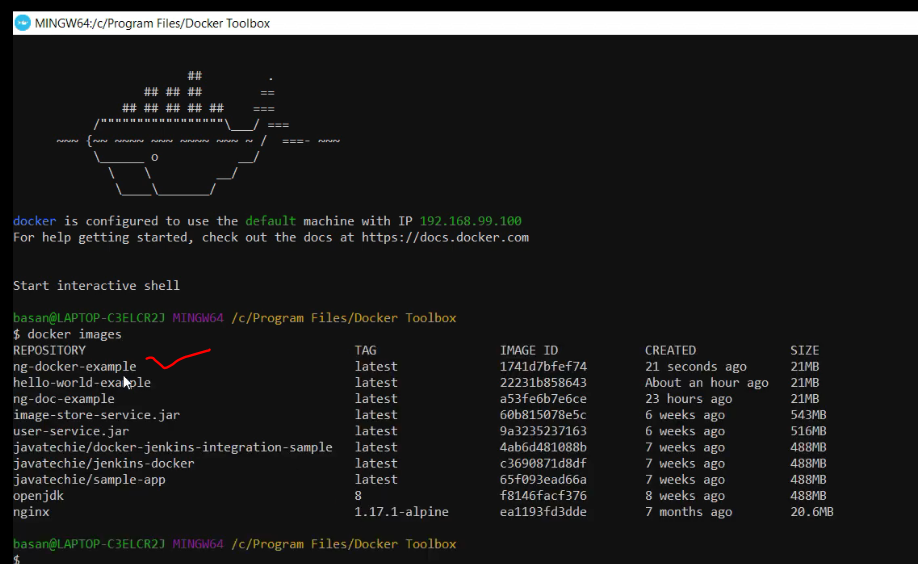
FROM – download nginx with this version

COPY – Then copy all the complied files from dist folder and copy it in the docker container

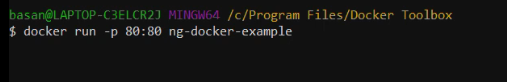


**Note:**

While building the docker image, make sure the docker terminal is up and running.

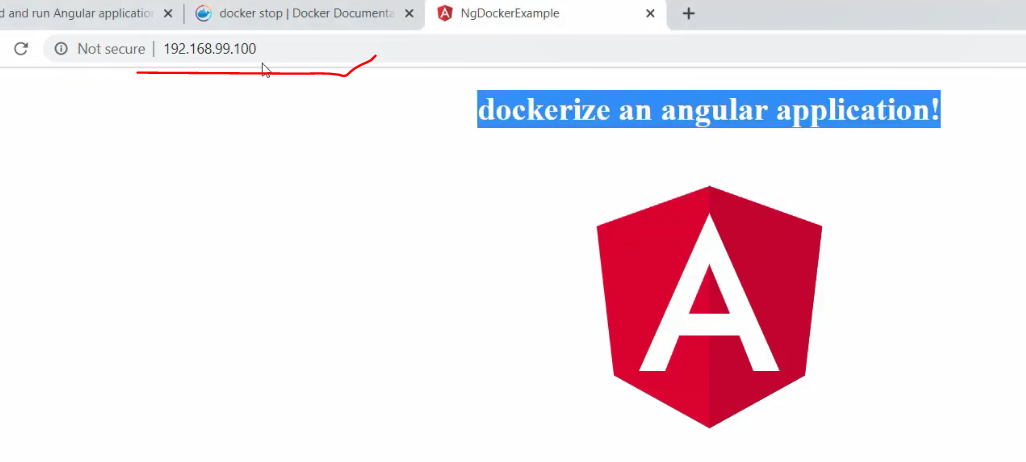


Now run the image:



Command:

Docker run -p <any port>:<any port> <image name>



Done!