Add the code from local to azure repo. And then create Dockerfile ( naming convention ) should be maintained.

For creating dockerfile, we need to understand the manual steps from building to compile the source code. We are actual converting the manual steps of building, compiling the code and packaging it with the help of docker container.   
  
  
we are working with nodejs application.   
  
base image docker hub : <https://hub.docker.com/search?type=image>  
  
base image docker hub for nodejs : <https://hub.docker.com/_/node>

**Docker File**  
  
FROM node:18-alpine AS installer

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

RUN npm run build

FROM nginx:latest AS deployer

COPY --from=installer /app/build /usr/share/nginx/html

Node:18-alpine ( alpine as os, this is light weight linux based os, as it doesn’t take lot of memory, space and docker file would be less so mostly get used )

We are using multi stage file.

As installer is 1st stage.

Workdir /app means changing the directory inside the container. ( like cd /app )

Copy is doing the copy function of all directory started with package in the current working directory which is ( /app )

Copy .. means copying all the files generated by npm install like library, dependencies to working directory

From ngnix:latest as deployer means using the 2nd stage. Using another image ngix to run the applications

As when we are running npm install, it is creating lot no of files which is not required for container and it is taking more of space and memory. Thus, it will make our container slow and docker image size would be huge. So that’s why we are dividing it into multiple stages. So that we we will use the stage installer, we are directly copying the files which is required as part of the build. And excluding the unnecessary files.

So we are directly copying from installer stage to ngnix working directory.

Npm run will create the /app/build directory and store the compile code in this directory. We are copying this to ngnix directory to serve it.