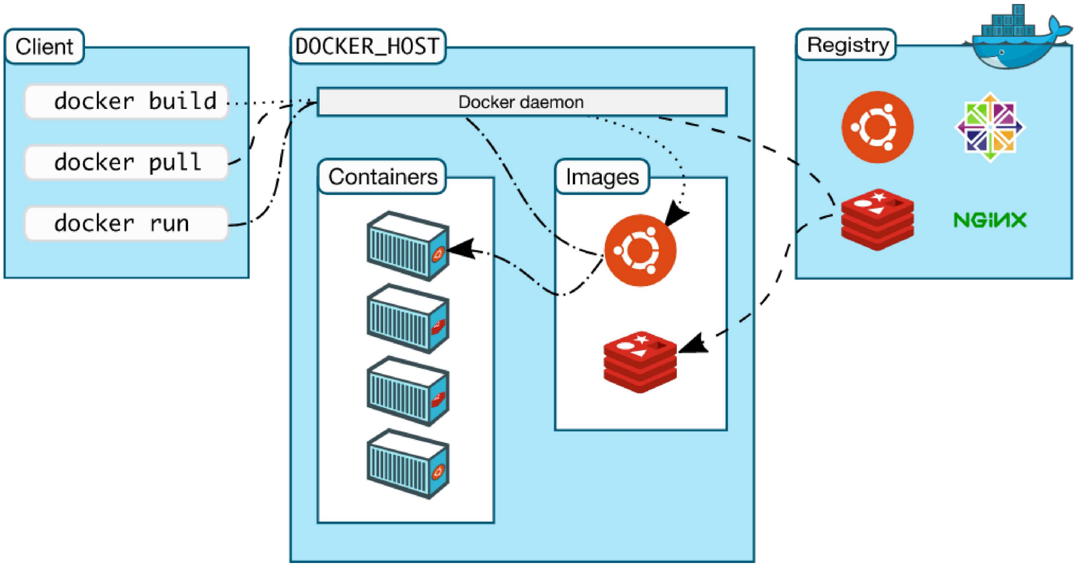
Experiment No : 09

Aim : To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.

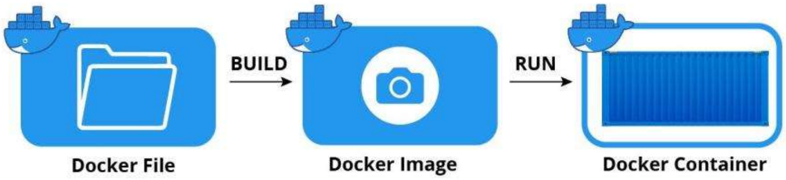
Theory :

Docker is a Linux-based, open-source containerization platform that developers use to build, run, and package applications for deployment using containers. Unlike virtual machines, Docker containers offer: OS-level abstraction with optimum resource utilization. Interoperability. Efficient build and test.

A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template. Docker images also act as the starting point when using Docker. An image is comparable to a snapshot in virtual machine (VM) environments.



Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container.



Base Image - In simple terms, a base image is an empty first layer, which allows you to build your Docker images from scratch. Base images give you full control over the contents of images, but are generally intended for more advanced Docker users.

Docker images have multiple layers, each one originates from the previous layer but is different from it. The layers speed up Docker builds while increasing reusability and decreasing disk use. Image layers are also read-only files. Once a container is created, a writable layer is added on top of the unchangeable images, allowing a user to make changes.

Prerequisite :

- Install Docker
- Create a Web App

Procedure :

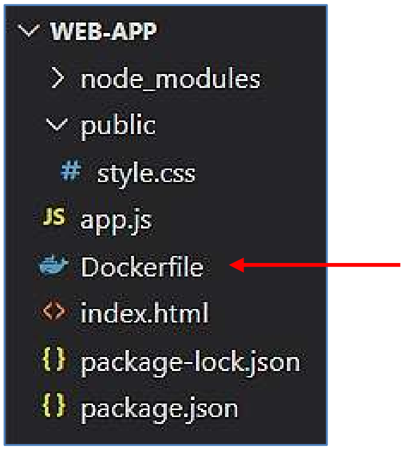
1. **Start Docker Deamon**

```
sudo systemctl start docker
```

OR

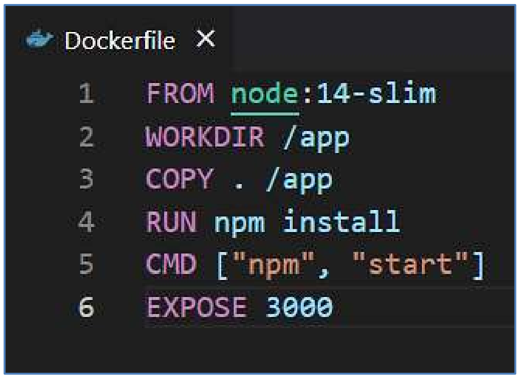
```
sudo service docker start
```

2. **Create a file named *`Dockerfile`* in the same folder as the file *`package.json`* with the following contents.**



```
WEB-APP
  node_modules
  public
    style.css
  app.js
  Dockerfile
  index.html
  package-lock.json
  package.json
```

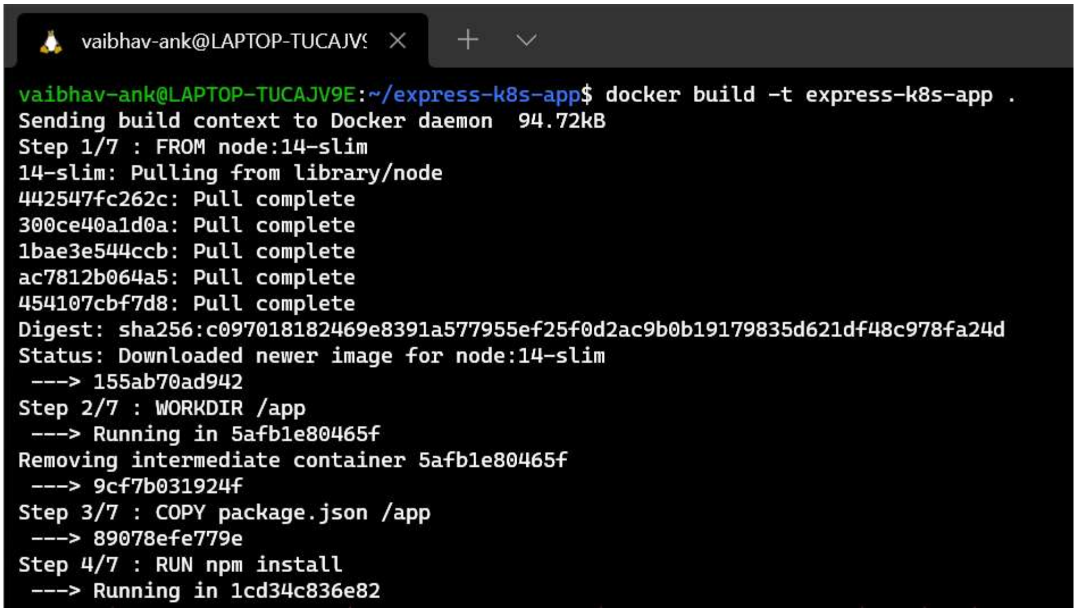
3. **In *`Dockerfile`* write the commands you want to run when your Docker image will run.** (Below code is for a NodeJS Web App)



```
FROM node:14-slim
WORKDIR /app
COPY . /app
RUN npm install
CMD ["npm", "start"]
EXPOSE 3000
```

4. **In Terminal, build your docker image using following command.**

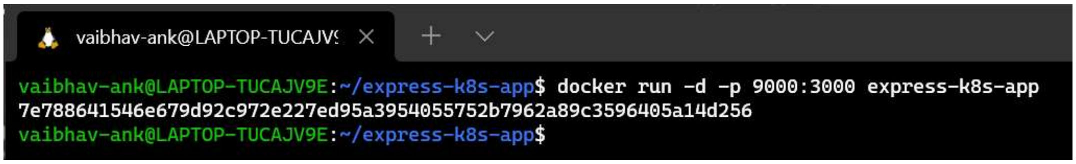
```
docker build -t docker_image_name path_to_app_folder
```



```
vaibhav-ank@LAPTOP-TUCAJV9E:~/express-k8s-app$ docker build -t express-k8s-app .
Sending build context to Docker daemon  94.72kB
Step 1/7 : FROM node:14-slim
14-slim: Pulling from library/node
442547fc262c: Pull complete
300ce40a1d0a: Pull complete
1bae3e544ccb: Pull complete
ac7812b064a5: Pull complete
454107cbf7d8: Pull complete
Digest: sha256:c097018182469e8391a577955ef25f0d2ac9b0b19179835d621df48c978fa24d
Status: Downloaded newer image for node:14-slim
 ---> 155ab70ad942
Step 2/7 : WORKDIR /app
 ---> Running in 5afb1e80465f
Removing intermediate container 5afb1e80465f
 ---> 9cf7b031924f
Step 3/7 : COPY package.json /app
 ---> 89078efe779e
Step 4/7 : RUN npm install
 ---> Running in 1cd34c836e82
```

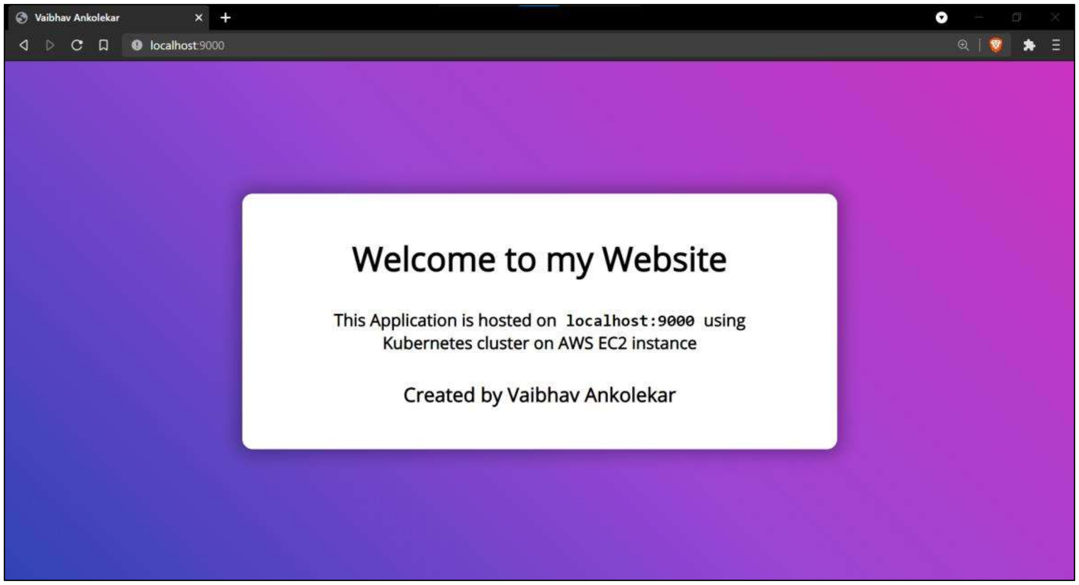
5. **Run the docker in your machine using below command.**

```
docker run -d -p <external-port>:<internal-port> image-name
```



```
vaibhav-ank@LAPTOP-TUCAJV9E:~/express-k8s-app$ docker run -d -p 9000:3000 express-k8s-app
7e788641546e679d92c972e227ed95a3954055752b7962a89c3596405a14d256
vaibhav-ank@LAPTOP-TUCAJV9E:~/express-k8s-app$
```

6. **Open Browser and go to** ***`http://<ip-address>:<external-port>`***



Welcome to my Website

This Application is hosted on `localhost:9000` using Kubernetes cluster on AWS EC2 instance

Created by Vaibhav Ankolekar

Conclusion :

The Docker goal is to ease the creation, deploy and the delivery of an application using the so-called Containers. The Docker Containers allow the developer/sysadmin to bundle an application with all needed components (libraries and other resources) and to deliver it as an independent and single package. It makes deployment very easy for a developer to deploy an App using less commands. Hence, we have successfully built and ran a Web App using Docker.