Introduction to Cloud Computing (DV1566)

Laboratory 1: Practice with containers

Group – 18

Ananya Reddivari.
Harika Kosuru.
Monica Gattupalli.
Sri Sai Ganesh Satyadeva Naidu Totakura.
Syam Kumar Vemana.

Question-1

We choose Bulletin Board application. A Bulletin Board is a platform that is used to provide information to the users on the internet, this program provides information by posting messages or advertising information. This program acts as a client-server application.

Web application server's functionality is to show user's list of events as shown in Figure-1, these events are user specified. This web application server is a primary component in Bulletin Board application. We interact with this program over internet through a web browser.

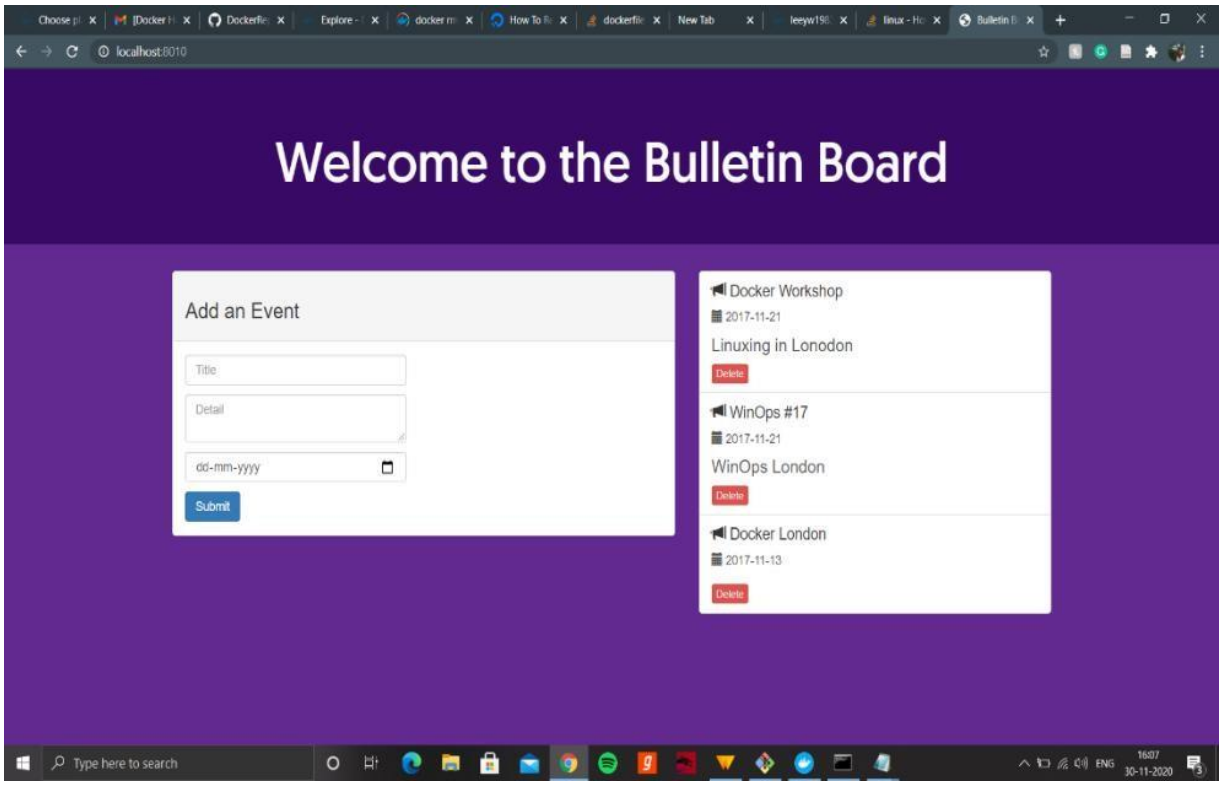


Figure-1: Web application server.

Question-2

We took the existing bulletin board called abasim/bulletinboard for building Docker image.

Docker file: -

Official image is used as a parent image.

FROM node:current-slim

Initiating working directory.

WORKDIR /usr/src/app

Copying to current location from host

COPY package.json .

Inside image filesystem command is executed.

RUN npm install

For describing, port the container is listening at runtime, we added metadata.

EXPOSE 8010

Executed within the container.

CMD ["npm", "start"]

Copying all our app's source code to image filesystem from host.

COPY . .

docker pull abasim/bulletinboard:1.0

This pull command pulls the image of abasim/bulletinboard:1.0 from Docker hub.

docker tag abasim/bulletinboard:1.0 satyadeva:1.0

This tag command tags the bulletin board in satyadeva repository.

docker push satyadeva:1.0

This command pushes the bulletin board into Docker desktop which is in satyadeva repository.

docker run -it -d -p 8010:5000 2dad8dfd3305

This run command runs the docker image in the official port 8010 where -d indicates background.

Networking Method: -

For containers to communicate each other and for also accessing the application, we used Bridge networking method.

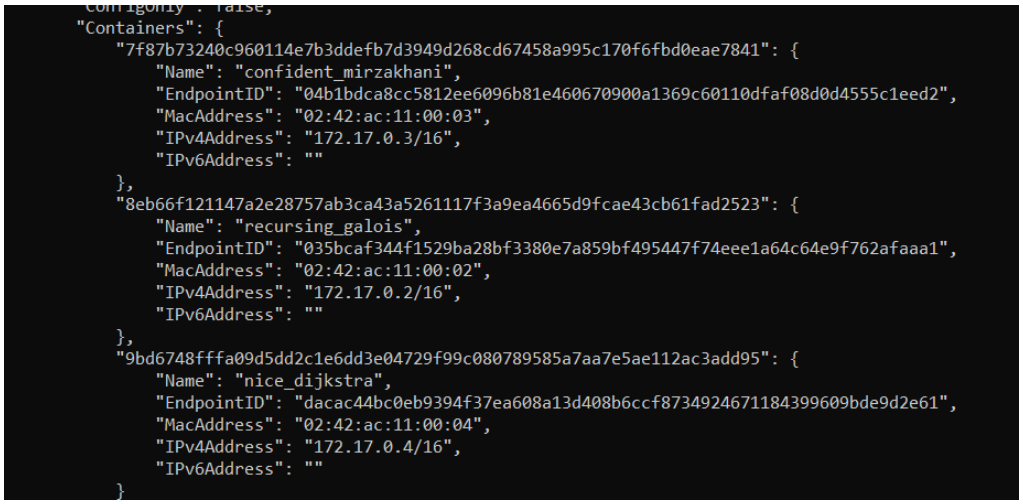
docker network ls

ls: - This command provides list of all networks along with network id and driver.

```
C:\Users\satya>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
08b15ae9848f        bridge              bridge              local
a9f5b39b979b        host                host                local
8a99ea03b71f        none                null                local
```

docker network inspect bridge

This command inspects whether there is a network assigned to every container or not, if there is a network encountered then it shows the respective ip address of that container.



docker run -d -p 8010:5000 -b my_vol:/usr/src/app test

This command creates a volume to the image in our port number and the volume name is “my_vol”.

docker volume ls

This command provides list of all volumes.

```
C:\Users\satya>docker volume ls
DRIVER              VOLUME NAME
local               my_bullentinboard
local               my_vol
```

Question-3

Docker-compose.yml: -

Docker-compose.yml was a yml file that was developed, we have added container deployment by keeping in view of two replicas. In case of an on-failure state we had set the restart policy, had also designated memory for resource use. We used port numbers 8010:5000 for communication.

```
services:
    web:
      deploy:
      replicas:2
        resources:
            limits:
            memory:30M
        max_attempts:3
        restart_policy:
        condition:on-failure delay:5s
      image:
      test/bulletinboard
      ports:
      -'8010'
      -'5000'
    version:'3.8'
```

Commands used to run and terminate services are: -

docker-compose up -d

docker-compose down

"-scale" is used to scale services manually.

docker-compose up -d –scale web=2

Docker Swarm:

For creating a swarm manager node, we used below command.

docker swarm init –advertise-addr <ip address>

For creating the workers node, we used below command.

docker swarm join –token <token id>

To create and scale services of application, we used below command.

docker service create –replicas 2 -p 8010:5000 test/bulletinboard

Motivation:

While setting-up initial process, we find it quite challenging due to lack of previous knowledge on Docker containers and images. As I said before we have zero information on Docker containers and images so we started to see some online tutorials from scratch, done some browsing about information on related topics, by doing so, we gained some knowledge and got some clarity on tasks to perform in lab-1. From then we effortlessly managed to go until Bind mount but from Bind mount, we started to face some difficulties on yml file and docker swarm. We got many errors during several stages, we have been patient and cleared all the errors by taking our time. All our groupmates contributed well enough and because of our teamwork we managed to complete this lab-1 within dedicated deadline.