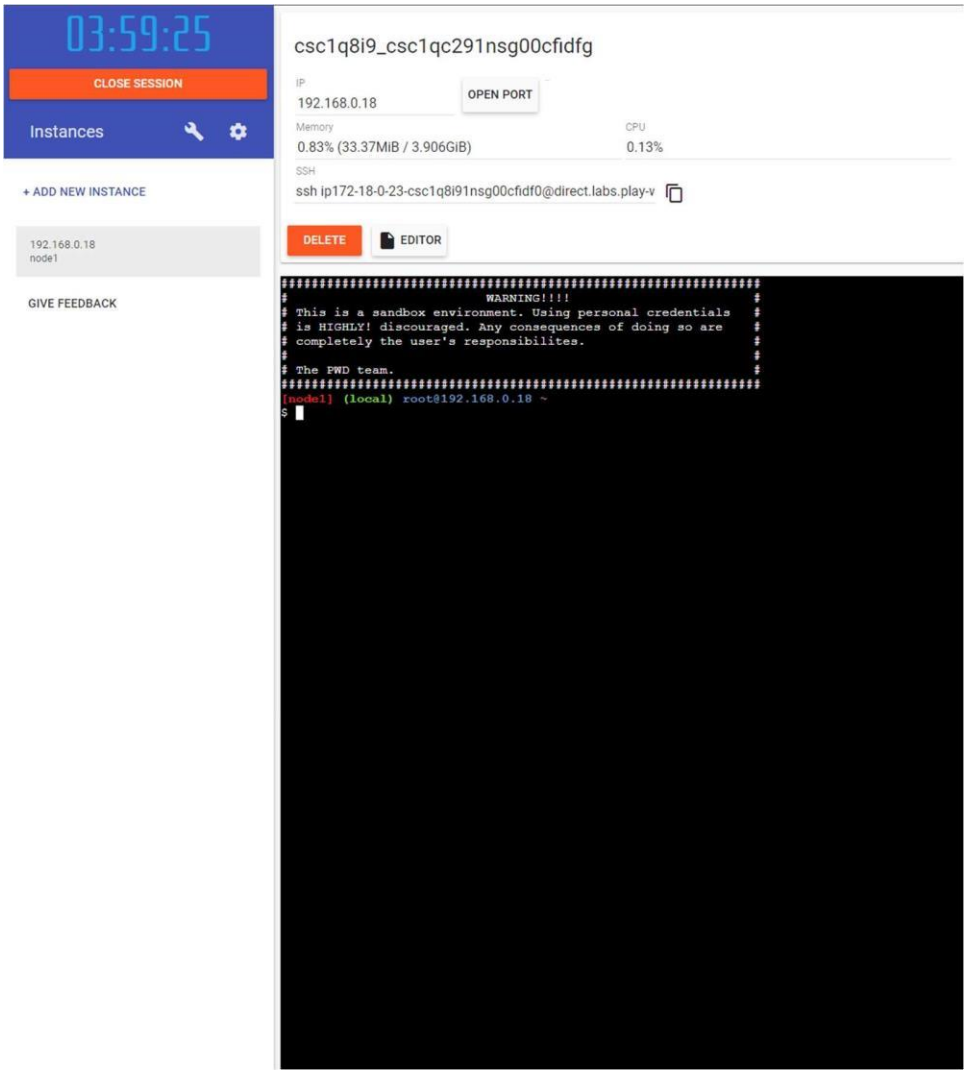
SVKM'S NMIMS (Deemed-to-be University)
MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT AND ENGINEERING
NAVI MUMBAI CAMPUS

Roll Number: A053	Name: Nishant Baruah
Class: B. Tech CE (Sem VII)	Batch: B2
Date of Experiment: 22-10-24	Date of Submission: 05-10-24
Subject: Cloud Computing	Practical No: 9

AIM:

Working with Docker Containers to run a web server

Task:


03:59:25
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.18
node1
GIVE FEEDBACK
csc1q8i9_csc1qc291nsg00cfidfg
IP
192.168.0.18
OPEN PORT
Memory
0.83% (33.37MiB / 3.906GiB)
CPU
0.13%
SSH
ssh ip172-18-0-23-csc1q8i91nsg00cfidf0@direct.labs.play-v
DELETE
EDITOR
WARNING!!!!
This is a sandbox environment. Using personal credentials
is HIGHLY! discouraged. Any consequences of doing so are
completely the user's responsibilites.
The PWD team.
[node1] (local) root@192.168.0.18 ~
$

```
[node1] (local) root@192.168.0.18 ~
$ mkdir project
[node1] (local) root@192.168.0.18 ~
$ cd project
[node1] (local) root@192.168.0.18 ~/project
$ touch index.html
[node1] (local) root@192.168.0.18 ~/project
$ vi index.html
[node1] (local) root@192.168.0.18 ~/project
$ touch Dockerfile
[node1] (local) root@192.168.0.18 ~/project
$ vi Dockerfile
[node1] (local) root@192.168.0.18 ~/project
$ vi Dockerfile
[node1] (local) root@192.168.0.18 ~/project
$ docker build . -t my-frontend
[+] Building 9.2s (7/7) FINISHED                                                docker:default
 => [internal] load build definition from Dockerfile                                      0.0s
 => => transferring dockerfile: 184B                                                      0.0s
 => [internal] load metadata for public.ecr.aws/nginx/nginx:1.26                          0.7s
 => [internal] load .dockerignore                                                         0.0s
 => => transferring context: 2B                                                           0.0s
 => [internal] load build context                                                         0.0s
 => => transferring context: 155B                                                         0.0s
 => [1/2] FROM public.ecr.aws/nginx/nginx:1.26@sha256:d950b46ff6d7c8700af095c8169d        8.2s
 => => resolve public.ecr.aws/nginx/nginx:1.26@sha256:d950b46ff6d7c8700af095c8169d        0.0s
 => => sha256:9018b2392d0db980b6c5d48fcae193183777fbe09950582cee80 1.78kB / 1.78kB        0.0s
 => => sha256:122ce9f0cbb4dfe43ffdb473f28715920b333fdb1a24276feb91 7.45kB / 7.45kB        0.0s
 => => sha256:a480a496ba95a197d587aa1d9e0f545ca7dbd40495a4715342 29.13MB / 29.13MB        0.4s
 => => sha256:f7e45c747637f0e61388543469410f0aec22f7f73e826bc568 41.88MB / 41.88MB        0.6s
 => => sha256:692a61bd1d675ef8df19e2ba27f6d4f033e5a608e73e88c9a26dca34 629B / 629B        0.1s
 => => sha256:d950b46ff6d7c8700af095c8169d9522a829656b00621f5b3f5156e8 772B / 772B        0.0s
 => => sha256:8992a25329a60603e9b1fdef8109e5366555eef8df387834f3847e6a 955B / 955B        0.2s
 => => sha256:f8eff2f530ec1da2a7304e45e8f31a67c31036974030ed6b5069 1.21kB / 1.21kB        0.3s
 => => sha256:eec32f85414d90fb81744a5304e6aaaae309038002cba7c848638e9a 394B / 394B        0.3s
 => => sha256:7a37000823d1e36c2b29d63a18b7514465b64352768dfe70ac3b 1.40kB / 1.40kB        0.4s
 => => extracting sha256:a480a496ba95a197d587aa1d9e0f545ca7dbd40495a4715342228db62        4.1s
 => => extracting sha256:f7e45c747637f0e61388543469410f0aec22f7f73e826bc568ba7d3dd        2.8s
 => => extracting sha256:692a61bd1d675ef8df19e2ba27f6d4f033e5a608e73e88c9a26dca342        0.0s
 => => extracting sha256:8992a25329a60603e9b1fdef8109e5366555eef8df387834f3847e6ab        0.0s
 => => extracting sha256:eec32f85414d90fb81744a5304e6aaaae309038002cba7c848638e9a4        0.0s
 => => extracting sha256:f8eff2f530ec1da2a7304e45e8f31a67c31036974030ed6b506964002        0.0s
 => => extracting sha256:7a37000823d1e36c2b29d63a18b7514465b64352768dfe70ac3b2c41d        0.0s
 => [2/2] COPY --chown=nginx index.html /usr/share/nginx/html/index.html                  0.1s
 => exporting to image                                                                    0.0s
 => => exporting layers                                                                   0.0s
 => => writing image sha256:7fcb5f2cdbb86c0946ba1115c99175789206f7a754e691a96bd120        0.0s
 => => naming to docker.io/library/my-frontend                                            0.0s
[node1] (local) root@192.168.0.18 ~/project
$
```

```
[node1] (local) root@192.168.0.18 ~/project
$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
my-frontend   latest    7fcb5f2cdbb8   27 seconds ago   188MB
[node1] (local) root@192.168.0.18 ~/project
$ docker run -d --name frontend-container -p 80:80 my-frontend
633c0ade8fecf707a0308764f7b73816e3f6359406b801dbbdade5429146944e
[node1] (local) root@192.168.0.18 ~/project
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORT
S              NAMES
633c0ade8fec   my-frontend    "/docker-entrypoint.…"   6 seconds ago   Up 5 seconds   0.0.
0.0:80->80/tcp   frontend-container
[node1] (local) root@192.168.0.18 ~/project
$
```

Hello, I am Reyansh

This is my first container project

Conclusion

In this experiment, I learned how to set up a html web page over Nginx web server using Docker containers.