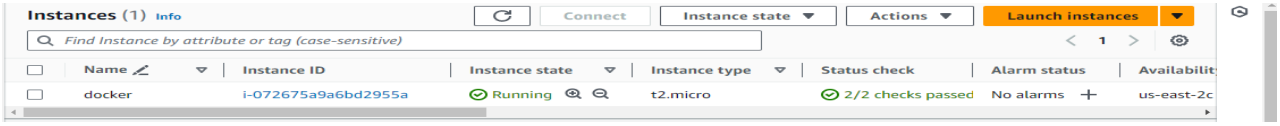
Docker

Step - 1 : Launch 1 instance in AWS EC2.



Step -2 : connect EC2 instance, which we have created.

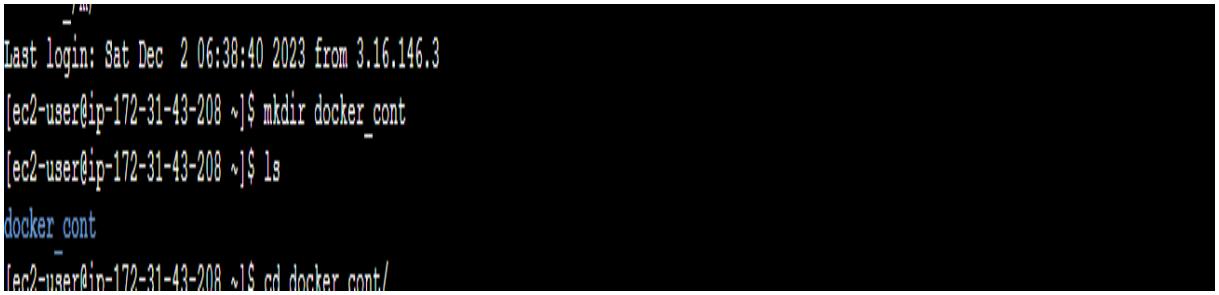
Step -3 : Download Docker.

To install : Docker

```
sudo yum update
sudo yum install docker -y
sudo systemctl enable docker
sudo systemctl start docker
sudo systemctl status docker
```

- After install, we should enable Docker, and start Docker, we can check the Docker status also, by using above commands.

- I created 1 directory, inside I'm going to build my Dockerfile, and run my image, and finally generating container.



Step - 4 : I have taken one simple python program

```
Dockerfile  app.py
[ec2-user@ip-172-31-43-208 Docker_practice]$ nano app.py
[ec2-user@ip-172-31-43-208 Docker_practice]$ cat app.py
print("Docker container is running")
[ec2-user@ip-172-31-43-208 Docker_practice]$
```

Step -5 : Created Dockerfile and added some configurations.

```
[ec2-user@ip-172-31-43-208 Docker_practice]$ cat app.py
print("Docker container is running")
[ec2-user@ip-172-31-43-208 Docker_practice]$ cat Dockerfile
FROM ubuntu:latest

# Set the working directory in the image
WORKDIR /app

# Copy the files from the host file system to the image file system
COPY . /app

# Install the necessary packages
RUN apt-get update && apt-get install -y python3 python3-pip

# Set environment variables
ENV NAME World

# Run a command to start the application
CMD ["python3", "app.py"]
[ec2-user@ip-172-31-43-208 Docker_practice]$
```

- No images, and no containers present before building a Dockerfile.

```
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]# docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
[root@ip-172-31-43-208 Docker_practice]# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
```

Step - 6 : **Build Dockerfile**

- If I build Dockerfile it will create a Image, after build Image will stored in DockerD.

To build : Dockerfile (cmd)

Docker build -t python .

- -t -- tagging image name.
- . (dot) -- current directory.

- After build, result will be like this.

```
[ec2-user@ip-172-31-43-208 Docker_practice]$ sudo docker build -t python .
[+] Building 36.7s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 460B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [1/4] FROM docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936225685bb
 => => resolve docker.io/library/ubuntu:latest@sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936225685bb
 => => sha256:8eab65df33a6de2844c9aefd19efe8ddb87b7df5e9185a4ab73af936225685bb 1.13kB / 1.13kB
 => => sha256:149d67e29f765f4db62aa52161009e99e389544e25a8f43c8c89d4a445a7ca37 424B / 424B
 => => sha256:b6548eacb0639263e9d8abfee48f8ac8b327102a05335b67572f715c580a968e 2.30kB / 2.30kB
 => => sha256:5e8117c0bd28aecad06f7e76d4d3b64734d59c1a0a44541d18060cd8fba30c50 29.55MB / 29.55MB
 => => extracting sha256:5e8117c0bd28aecad06f7e76d4d3b64734d59c1a0a44541d18060cd8fba30c50
 => [internal] load build context
 => => transferring context: 34.00kB
 => [2/4] WORKDIR /app
 => [3/4] COPY . /app
 => [4/4] RUN apt-get update && apt-get install -y python3 python3-pip
 => exporting to image
 => => exporting layers
 => => writing image sha256:4092bad6a95699058adbfc4192f0038c179d456f6d286cc8ebbc839925601eb7
 => => naming to docker.io/library/python
```

Images has been Successfully Created using by building Dockerfile.

To check : image is created or not. (cmd)

```
docker images
```

```
[root@ip-172-31-43-208 Docker_practice]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
python       latest    4092bad6a956   24 minutes ago   472MB
```

Step - 7 : **Run Image.**

- If I run image, container will create, and the python program will run, output will shown in below.

To run : image (cmd)

Docker run -it python

- run - To run a image.
- -it - Used together to make the container run in interactive mode.

To check : docker container (cmd)

```
docker ps -a
```

```
[root@ip-172-31-43-208 Docker_practice]# docker ps -a
CONTAINER ID   IMAGE     COMMAND           CREATED              STATUS                          PORTS     NAMES
a94eb55046bd   python    "python3 app.py"  About a minute ago   Exited (0) About a minute ago             gallant_diffie
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
[root@ip-172-31-43-208 Docker_practice]#
```

- After running the docker image we receive the docker container
- Output is shown below:-

```
[root@ip-172-31-43-208 Docker_practice]# docker run -it python
Docker container is running
```

Dockerfile ➔ Dockerimage ➔ Dockercontainer