Experiment-7

AIM:- Improvise the docker image quality using devops.

(Note:-Before do first follow below steps. Here we will create image with tag and also we minimize the image size changing version or using multi staging)

Step 1:-Connecting AWS Instance Ubuntu using Mobaxterm

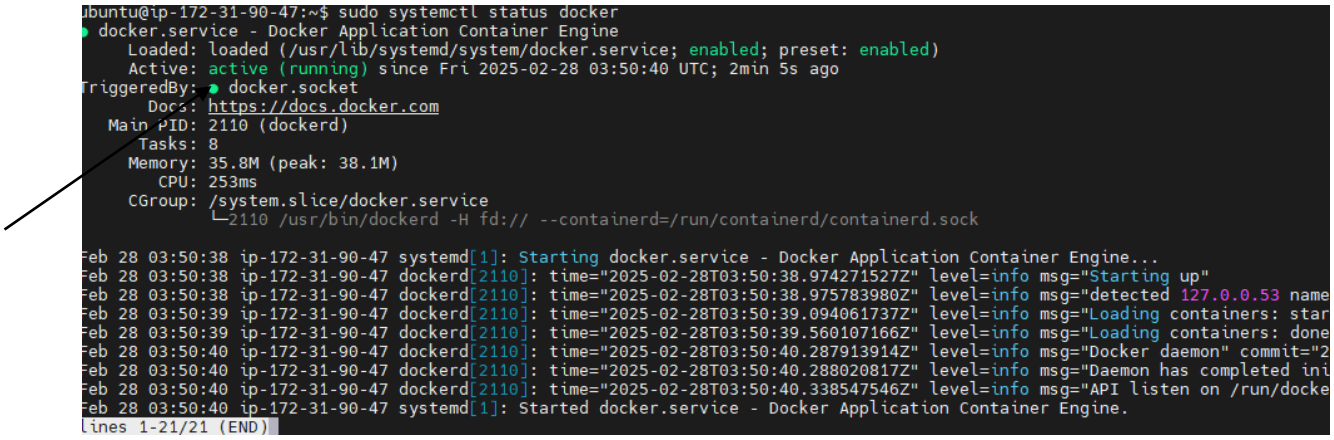
1. Login AWS(Amazon Web Services) Account
2. Lunch Instance name Docker
3. Connect to Ubuntu or mobaxterm

(**Note:**-Follow this url for docker file and application—https://github.com/devisar/devopslab)

Step 2:-Create Docker Hub Account and create repository in Docker Hub

Step 3:-Install Docker and Check Status and Start Docker in mobaxterm

1. sudo apt update -y
2. sudo apt install docker.io –y
3. sudo systemctl status docker(come outside use command ctl+z)



Above status command is docker running means no problem if not run use command below to run

4. sudo systemctl start docker

Step 4:- Grant Access

Why we give grant access means

A easy way to verify your Docker installation is by running the below command

docker run hello-world

If the output says:

```
ubuntu@ip-172-31-90-47:~$ docker run hello-world
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permiss
ion denied.
See 'docker run --help'.
ubuntu@ip-172-31-90-47:~$
```

This can mean two things,

1. Docker deamon is not running.(start docker using “sudo systemctl start docker”)
2. Your user does not have access to run docker commands.

Grant Access to your user to run docker commands

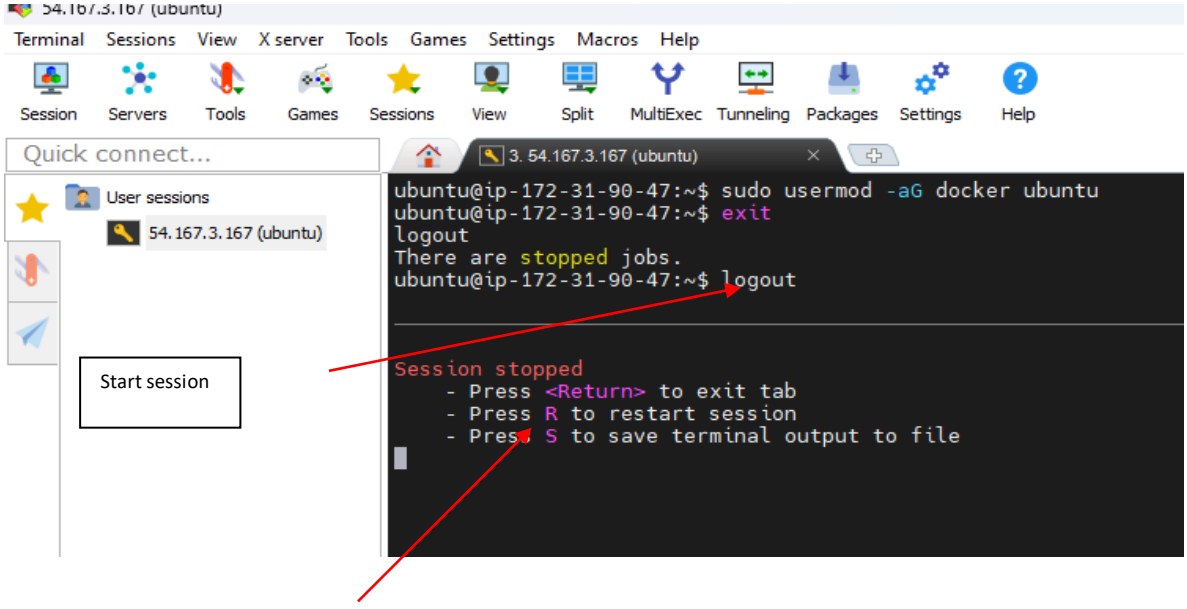
1. sudo usermod -aG docker ubuntu

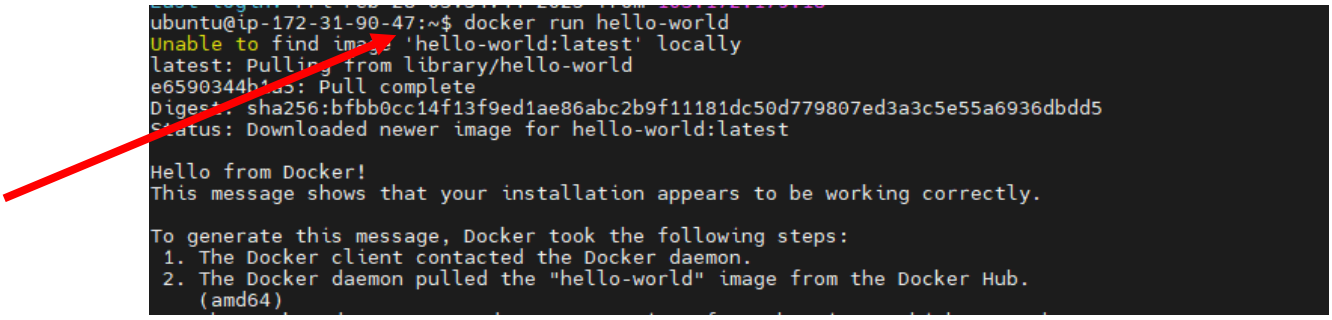
In the above command ubuntu is the name of the user, you can change the username appropriately.

NOTE: : You need to logout and login back for the changes to be reflected.

2. Logout purpose use commands exit or logout and Press R to restart session.

   Again run command “docker run hello-world”





Step 5:-Creating application and Docker file

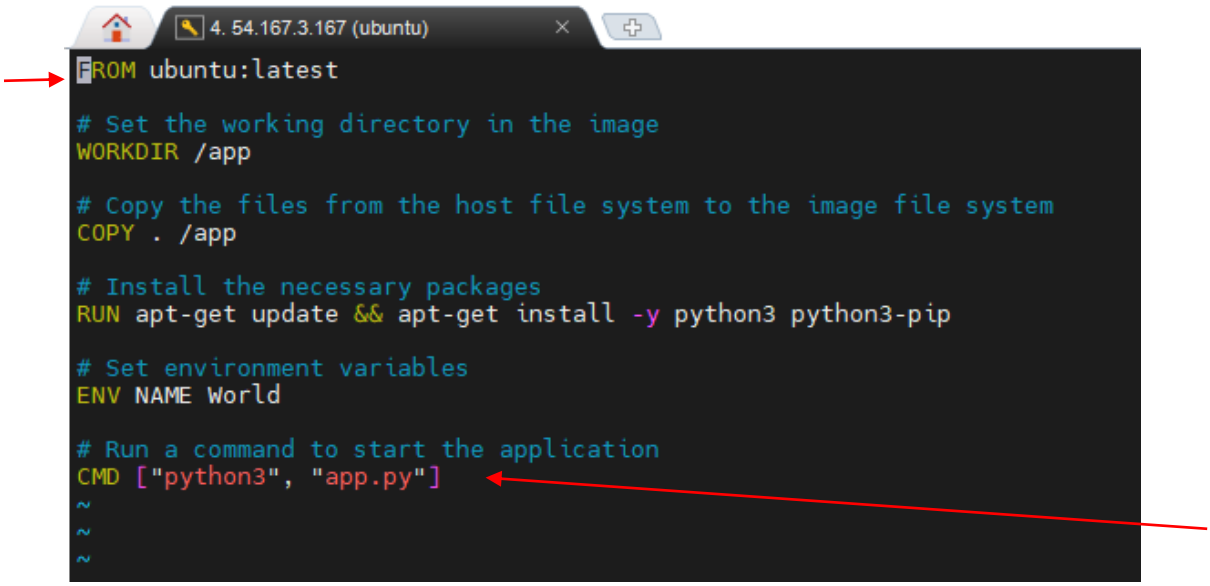
Commands:-

1. Mkdir docker1
2. cd docker1
3. vim app.py

   print("hello world")

4. cat app.py
5. vim Dockerfile

(below pic commands write lab record) typing command vim before click "i" for insert data after completion Docker file commands save before click esc use :wq!



Below image for understanding purpose

```
ubuntu@ip-172-31-90-47:~$ mkdir docker1
ubuntu@ip-172-31-90-47:~$ cd docker1
ubuntu@ip-172-31-90-47:~/docker1$ vim app.py
ubuntu@ip-172-31-90-47:~/docker1$ cat app.py
print("hello world")
ubuntu@ip-172-31-90-47:~/docker1$ vim Dockerfile
ubuntu@ip-172-31-90-47:~/docker1$ 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"]

ubuntu@ip-172-31-90-47:~/docker1$
```

se

Step 6:- Build and Check Docker image

Syntax:- docker build –t dockerhub_username/repositoryname:tag .

1. docker build –t anu1308/dockerimage:latest .
2. docker images

Below images for understanding purpose

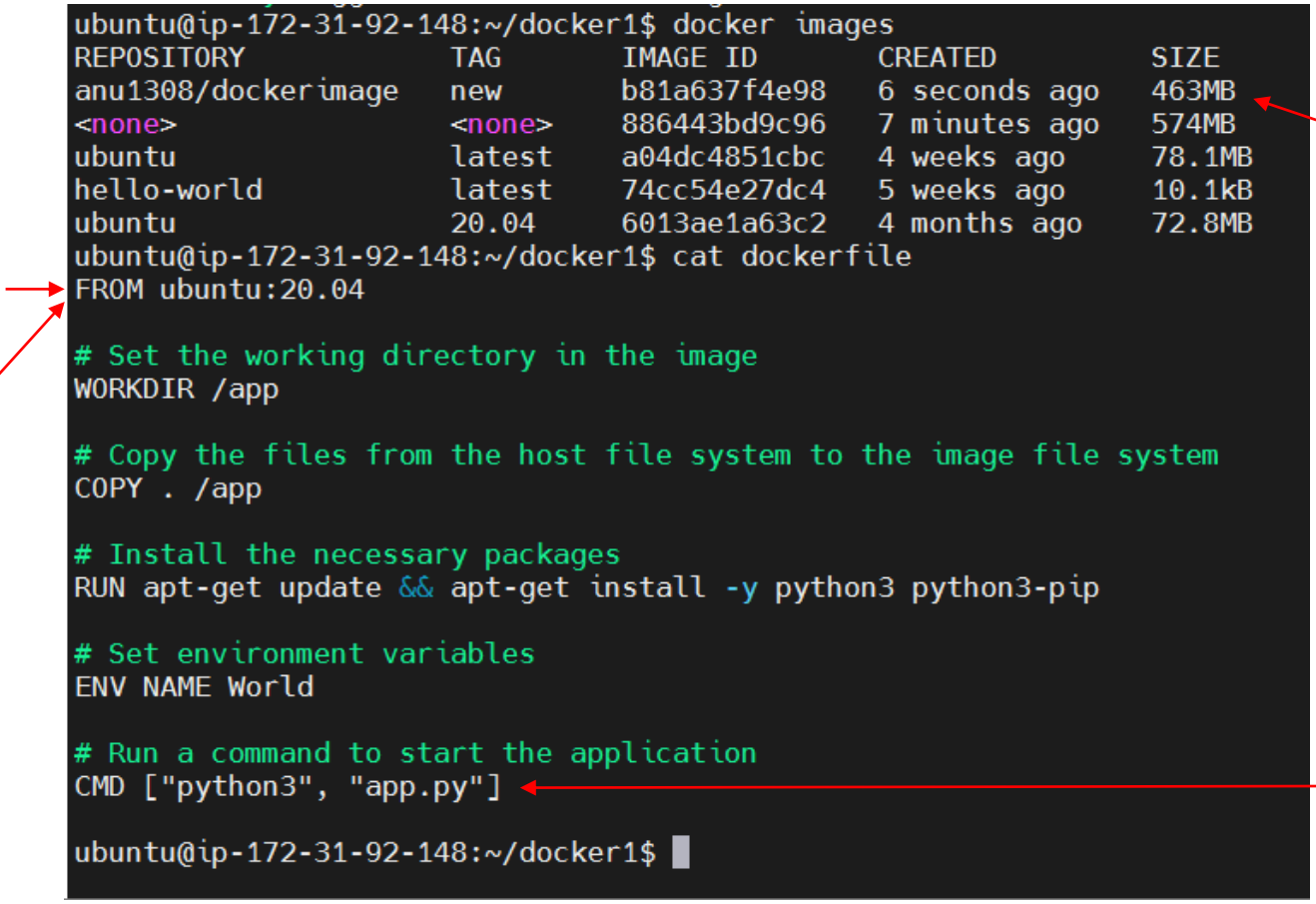
```
ubuntu@ip-172-31-90-47:~/docker1$ docker build -t anu1308/dockerimage:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/6 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
5a7813e071bf: Pull complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
 ---> a04dc4851cbc
Step 2/6 : WORKDIR /app
 ---> Running in 490762c2b766
 ---> Removed intermediate container 490762c2b766
 ---> f19d0296889a
Step 3/6 : COPY . /app
 ---> 05e2e2564f3a
Step 4/6 : RUN apt-get update && apt-get install -y python3 python3-pip
 ---> Running in 0cad8878ec6b
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [842 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [24.2 kB]
Successfully tagged anu1308/dockerimage:latest
ubuntu@ip-172-31-90-47:~/docker1$ docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
anu1308/dockerimage   latest    cdbf156a1eda   56 seconds ago   574MB
ubuntu                latest    a04dc4851cbc   4 weeks ago      78.1MB
hello-world           latest    74cc54e27dc4   5 weeks ago      10.1kB
ubuntu@ip-172-31-90-47:~/docker1$
```

Step 7:-Changing Version to minimize image quality

Instead of using ubuntu:latest, consider lighter base images like python:3.9-slim or python:3.9-alpine or specific versions of ubuntu like ubuntu:20.04 for smaller,stable and secure images.

(Note:-Write from to cmd all in you are lab record)



Step 8:-Using Multi stages to minimizing image quality

Multi-stage builds allow you to separate the build environment from the production environment.This reduces the final image size.

- **Steps:**
  - Create a **build stage** where you compile or install dependencies.
  - Copy only the necessary files to the **final stage**, leaving behind build tools and unnecessary files.

  (Note:-Write from to cmd all in you are lab record)

```
Successfully tagged anu1308/dockerimage:new
ubuntu@ip-172-31-92-148:~/docker1$ docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
anu1308/dockerimage     new       395136d147e2   9 seconds ago    78.1MB
<none>                  <none>    7142b374fde9   10 seconds ago   574MB
<none>                  <none>    b81a637f4e98   30 minutes ago   463MB
<none>                  <none>    886443bd9c96   38 minutes ago   574MB
ubuntu                  24.04     a04dc4851cbc   5 weeks ago      78.1MB
ubuntu                  latest    a04dc4851cbc   5 weeks ago      78.1MB
hello-world             latest    74cc54e27dc4   5 weeks ago      10.1kB
ubuntu                  20.04     6013ae1a63c2   4 months ago     72.8MB
ubuntu@ip-172-31-92-148:~/docker1$ cat dockerfile
# Build Stage
FROM ubuntu:24.04 AS builder

# Set the working directory in the build stage
WORKDIR /app

# Copy only the necessary files to the build stage
COPY . /app

# Install build dependencies (python and pip)
RUN apt-get update && apt-get install -y python3 python3-pip

# Install any Python dependencies if needed (e.g., from requirements.txt)
# RUN pip3 install -r requirements.txt   # Uncomment if you have a requirements.txt

# Production Stage
FROM ubuntu:latest

# Set the working directory in the production stage
WORKDIR /app

# Copy only the necessary files from the build stage (avoiding build dependencies)
COPY --from=builder /app /app

# Set environment variables
ENV NAME World

# Run the application
CMD ["python3", "app.py"]
ubuntu@ip-172-31-92-148:~/docker1$
```

Step 9:-Creating Container and check container

Create command:-

docker run –it anu1308/dockerimage:new

Check Container command:-

docker ps -a

```
ubuntu@ip-172-31-92-148:~/docker1$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS     NAMES
a9026b81571c   hello-world   "/hello"   24 minutes ago   Exited (0) 24 minutes ago             silly_proskuriakova
ubuntu@ip-172-31-92-148:~/docker1$ docker run -it anu1308/dockerimage:new
Hello world
ubuntu@ip-172-31-92-148:~/docker1$ docker ps -a
CONTAINER ID   IMAGE                    COMMAND          CREATED          STATUS                      PORTS     NAMES
a1f2de3f317d   anu1308/dockerimage:new  "python3 app.py" 6 seconds ago    Exited (0) 4 seconds ago              confident_hawking
a9026b81571c   hello-world              "/hello"         25 minutes ago   Exited (0) 25 minutes ago             silly_proskuriakova
ubuntu@ip-172-31-92-148:~/docker1$
```

Output:-

```
ubuntu@ip-172-31-92-148:~/docker1$ 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"]

ubuntu@ip-172-31-92-148:~/docker1$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    74cc54e27dc4   5 weeks ago    10.1kB
ubuntu@ip-172-31-92-148:~/docker1$ docker build -t anu1308/dockerimage:new .
DEPRECATED: The legacy builder is deprecated and will be removed in a future rel
ease.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/6 : FROM ubuntu:latest
latest: Pulling from library/ubuntu
```

Image tag

```
ubuntu@ip-172-31-92-148:~/docker1$ vim dockerfile
ubuntu@ip-172-31-92-148:~/docker1$ docker images
REPOSITORY            TAG       IMAGE ID       CREATED         SIZE
anu1308/dockerimage   new       886443bd9c96   6 minutes ago   574MB
ubuntu                latest    a04dc4851cbc   4 weeks ago     78.1MB
hello-world           latest    74cc54e27dc4   5 weeks ago     10.1kB
```

Fig.1.Before Build no image is created and After Build Image created

```
ubuntu@ip-172-31-92-148:~/docker1$ docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
anu1308/dockerimage   new       b81a637f4e98   6 seconds ago    463MB
<none>                <none>    886443bd9c96   7 minutes ago    574MB
ubuntu                latest    a04dc4851cbc   4 weeks ago      78.1MB
hello-world           latest    74cc54e27dc4   5 weeks ago      10.1kB
ubuntu                20.04     6013ae1a63c2   4 months ago     72.8MB
ubuntu@ip-172-31-92-148:~/docker1$ cat dockerfile
FROM ubuntu:20.04

# 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"]

ubuntu@ip-172-31-92-148:~/docker1$
```

Fig.2.After changing version latest to 20.04 Image size minimize 574 to 463

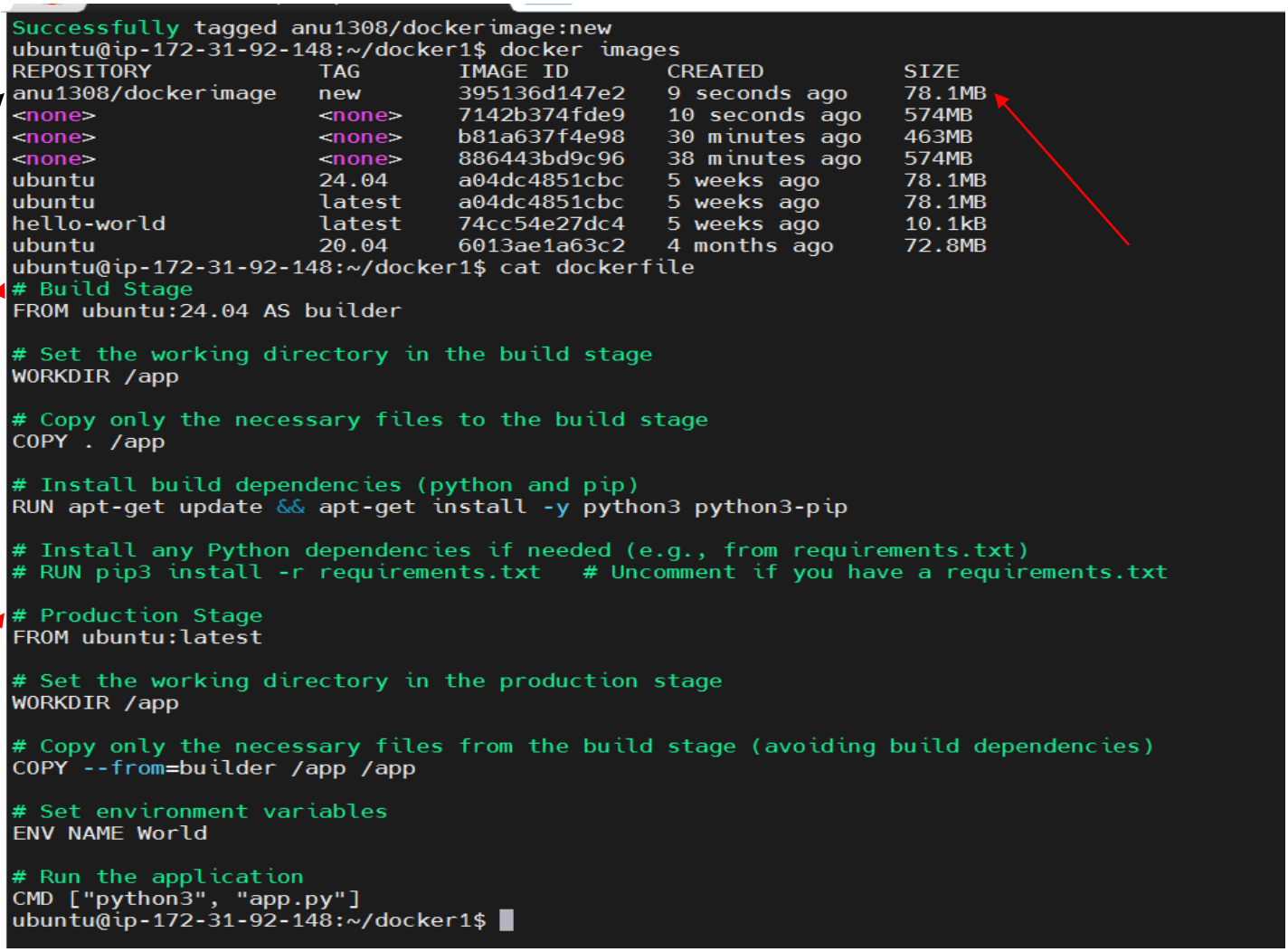


Fig.3.using Multi Stage to minimize image size 463 to 78.1 MB