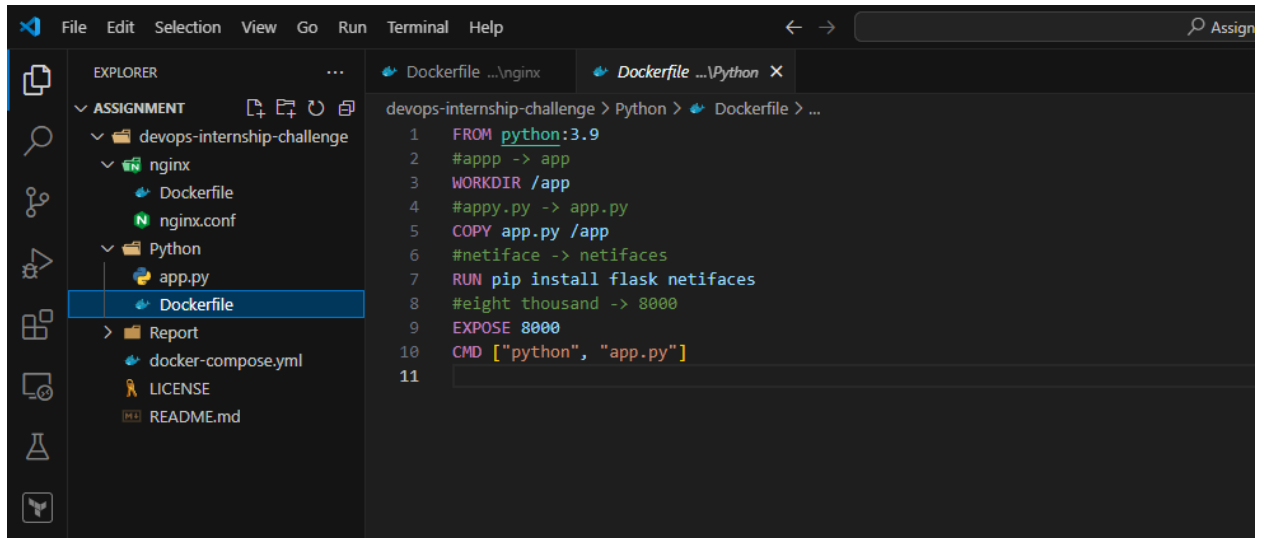


Assignment:

devops-qoala-assignment-Aditya-Singh-21ucc011

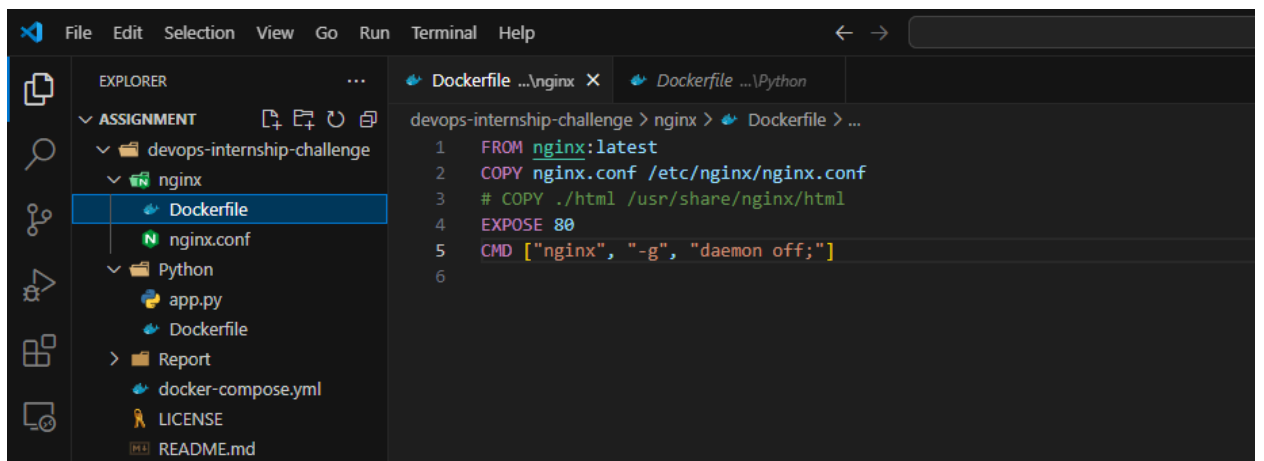
Approach to the Problem:

1. First we **debug the dockerfile** of both **nginx** and **Python**



This screenshot shows the Visual Studio Code interface with the Explorer panel on the left and the Dockerfile editor on the right. The Explorer panel shows the project structure under 'devops-internship-challenge', including 'nginx' and 'Python' folders. The 'Python' folder is selected, and its 'Dockerfile' is open in the editor. The Dockerfile content is as follows:

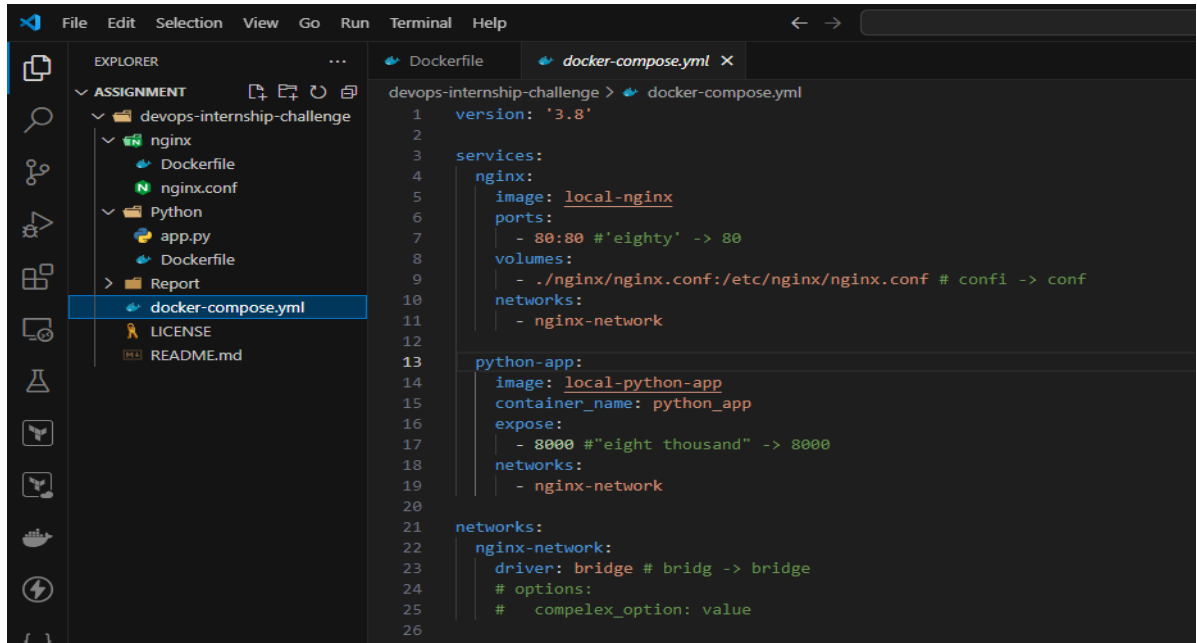
```
devops-internship-challenge > Python > Dockerfile > ...
1 FROM python:3.9
2 #app -> app
3 WORKDIR /app
4 #app.py -> app.py
5 COPY app.py /app
6 #netifaces -> netifaces
7 RUN pip install flask netifaces
8 #eight thousand -> 8000
9 EXPOSE 8000
10 CMD ["python", "app.py"]
11
```



This screenshot shows the Visual Studio Code interface with the Explorer panel on the left and the Dockerfile editor on the right. The Explorer panel shows the project structure under 'devops-internship-challenge', including 'nginx' and 'Python' folders. The 'nginx' folder is selected, and its 'Dockerfile' is open in the editor. The Dockerfile content is as follows:

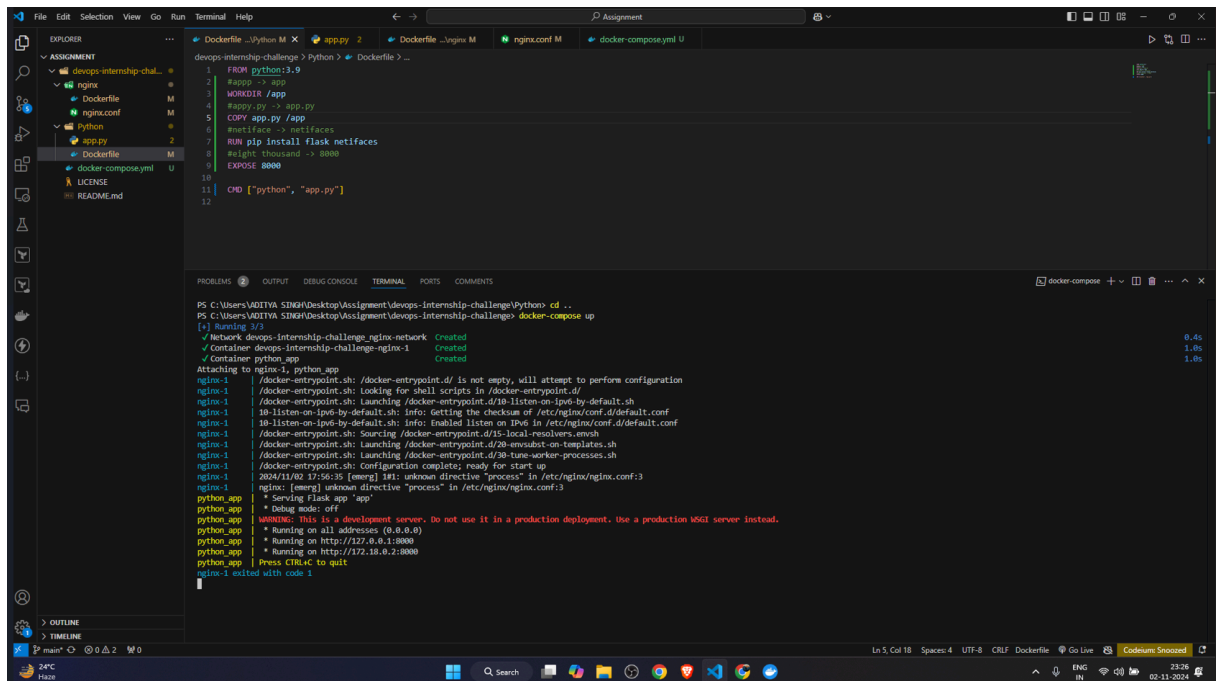
```
devops-internship-challenge > nginx > Dockerfile > ...
1 FROM nginx:latest
2 COPY nginx.conf /etc/nginx/nginx.conf
3 # COPY ./html /usr/share/nginx/html
4 EXPOSE 80
5 CMD ["nginx", "-g", "daemon off;"]
6
```

-
2. We then debug the compose file and then changed the extension from **.yaml** to **.yml**



```
1 version: '3.8'
2
3 services:
4   nginx:
5     image: local-nginx
6     ports:
7       - 80:80 # 'eighty' -> 80
8     volumes:
9       - ./nginx/nginx.conf:/etc/nginx/nginx.conf # confi -> conf
10    networks:
11      - nginx-network
12
13   python-app:
14     image: local-python-app
15     container_name: python_app
16     expose:
17       - 8000 # "eight thousand" -> 8000
18     networks:
19       - nginx-network
20
21 networks:
22   nginx-network:
23     driver: bridge # bridg -> bridge
24     # options:
25     #   complex_option: value
26
```

-
-
3. We first build the docker images of the nginx and the run the command **docker-compose up** to run the application locally on port: 80



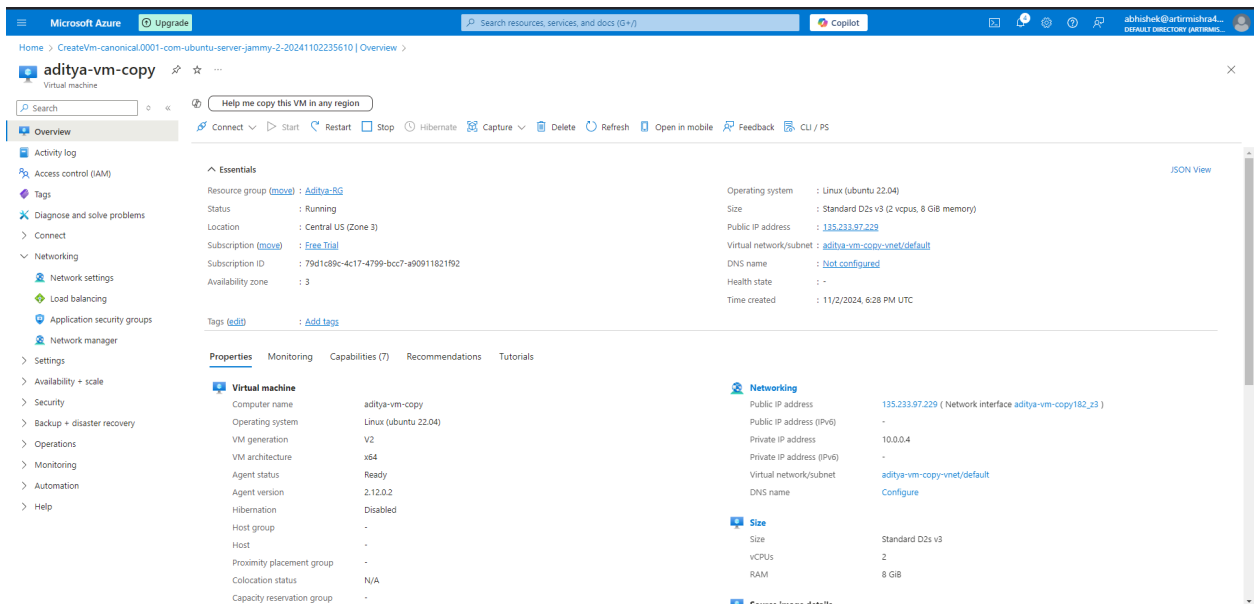
```
PS C:\Users\ADITYA SINGH\Desktop\Assignment\devops-interhip-challenge> docker-compose up
[*] Running 1/3
✓ Network devops-interhip-challenge-nginx-network Created
✓ Container devops-interhip-challenge-nginx-1 Created
✓ Container python-app Created
Attaching to nginx-1, python-app
nginx-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx-1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
nginx-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
nginx-1 | 2024/11/02 17:56:35 [emerg] 181: unknown directive "process" in /etc/nginx/nginx.conf:3
nginx-1 | * Serving Flask app
python-app | * Serving Flask app
python-app | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
python-app | * Running on all addresses (0.0.0.0)
python-app | * Running on http://172.18.0.1:8000
python-app | * Running on http://172.18.0.2:8000
python-app | Press CTRL+C to quit
nginx-1 exited with code 1
```

4. Attaching the Screenshot of the locally running application



5. Now for the Bonus Points:

Let's **deploy** it on Azure Cloud after creating a **Virtual Machine**



6. Now we will access the SSH on cmd by the following command
ssh [user_name]@[Public-IP]:

```
aditya-vm@aditya-vm-copy: x + -
C:\Users\ADITYA SINGH>ssh aditya-vm@135.233.97.229
aditya-vm@135.233.97.229's password:
Permission denied, please try again.
aditya-vm@135.233.97.229's password:
Permission denied, please try again.
aditya-vm@135.233.97.229's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-1025-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Nov  2 18:38:33 UTC 2024

System load:  0.0               Processes:    111
Usage of /:   5.2% of 28.89GB   Users logged in:  0
Memory usage: 3%               IPv4 address for eth0: 10.0.0.4
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

aditya-vm@aditya-vm-copy:~$
```

7. Now we follow the same set of instructions on Virtual Machine

a. Install Docker

b. Install Docker Compose

```
aditya-vm@aditya-vm-copy: X
Setting up docker-buildx-plugin (0.17.1-1-ubuntu.22.04-jammy) ...
Setting up containerd.io (1.7.22-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.29.7-1-ubuntu.22.04-jammy) ...
Setting up libltdl7:amd64 (2.4.6-1build2) ...
Setting up docker-ce-cli (5:27.3.1-1-ubuntu.22.04-jammy) ...
Setting up libltdl-dev:amd64 (2.4.6-1build2) ...
Setting up pigz (2.6-1) ...
Setting up docker-ce-rootless-extras (5:27.3.1-1-ubuntu.22.04-jammy) ...
Setting up slirp4netns (1.8.1-2) ...
Setting up docker-ce (5:27.3.1-1-ubuntu.22.04-jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Scanning processes...
Scanning Linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
aditya-vm@aditya-vm-copy:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-11-02 18:41:15 UTC; 6s ago
     Triggers: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 3600 (dockerd)
     Tasks: 9
    Memory: 23.3M
       CPU: 38ms
   CGroup: /system.slice/docker.service
           └─3600 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 02 18:41:14 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:14.311368636Z" level=info msg="detected 127.0.0.53 nameserver, assuming system-resolved, so using resolv.conf: /run/systemd/resolve/resolv
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.042786895Z" level=info msg="Loading containers: start."
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.485569394Z" level=info msg="Loading containers: done."
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.485569394Z" level=warning msg="Not using native diff for overlay2, this may cause degraded performance for building images: kernel has c
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.485297212Z" level=warning msg="WARNING: bridge-nf-call-iptables is disabled"
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.485312412Z" level=warning msg="WARNING: bridge-nf-call-ip6tables is disabled"
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.485337012Z" level=info msg="Docker daemon" commit=aca97b containerd=graphotter=false storage=driver=overlay2 version=27.3.1
Nov 02 18:41:15 aditya-vm-copy dockerd[3600]: time="2024-11-02T18:41:15.504879990Z" level=info msg="Daemon has completed initialization"
Nov 02 18:41:15 aditya-vm-copy systemd[1]: Started Docker Application Container Engine.
[lines 1-22/22 (END)]
```

8. Now we git clone the repo and the **problem** I faced was that I was **not able to access the Private repo**, So for that I created a repo which was Public and then **deployed it on V.M**

```
root@aditya-vm-copy: /home
root@aditya-vm-copy:~# cd home
root@aditya-vm-copy:/home# ls
aditya-vm
root@aditya-vm-copy:/home# cd aditya-vm/
root@aditya-vm-copy:/home/aditya-vm# ls
assignment
root@aditya-vm-copy:/home/aditya-vm# cd assignment/
root@aditya-vm-copy:/home/aditya-vm/assignment# ls
LICENSE Python README.md docker-compose.yml nginx
root@aditya-vm-copy:/home/aditya-vm/assignment# cd Python/
root@aditya-vm-copy:/home/aditya-vm/assignment/Python# ls
Dockerfile app.py
root@aditya-vm-copy:/home/aditya-vm/assignment/Python# docker build -t local-python-app .
[+] Building 34.4s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring Dockerfile: 238B
=> [internal] load metadata for docker.io/library/python:3.9
=> [internal] load dockerignore
=> transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.9@sha256:ed8b9dd4e9f89c11f4b4db85a55f8c9f9e22796a2984a9380b15f627d9914095
=> resolve docker.io/library/python:3.9@sha256:ed8b9dd4e9f89c11f4b4db85a55f8c9f9e22796a2984a9380b15f627d9914095
=> sha256:3de9e023d8992f2f4ad674068f8ad8a7c0f49992936e1c276f84f0809a15a62 6.36MB / 6.36MB
=> sha256:709e013d8ef037a5721108a08117a3b4aef152a8e1c11d8d19171803 69.50MB / 69.50MB
=> sha256:da802df85c96baca9d39869f9e2cbb3dc8uud627f413bfbb2f2c3d6e55988 24.05MB / 24.05MB
=> sha256:7aad5892c3b7a86566b10bf19d08322b1961204271a158c9eac1ed1b 64.39MB / 64.39MB
=> sha256:e0b0daee4f06c11f4b4db85a55f8c9f9e22796a2984a9380b15f627d9914095 19.35MB / 19.35MB
=> sha256:980816fef8a8a957c9ac8b04502951bdca10c771576d529431cc38c5c61b4 2.32MB / 2.32MB
=> sha256:ad1c7fc337f5c36fc207b5b5f8a8fbc6803771895769532fc3240b06b15343 211.27MB / 211.27MB
=> sha256:da4b115a421399a047666a321263977f31e779480dc8d8d8f9bfb35f92d4 6.10MB / 6.10MB
=> extracting sha256:7d98d813d4f6297a5772108b4a881378343a8f1b2db6c1214060191718056
=> sha256:cccccc1cbfb3a839f853b1dda5a0fb46a5b0a4fd39fb38caf526778824 19.80MB / 19.80MB
=> sha256:3d4c31f992b20a72702988a430bcbcd0bea170e1f13bf8859f64d9171 250B / 250B
=> extracting sha256:da802df85c96baca9d39869f9e2cbb3dc8uud627f413bfbb2f2c3d6e55988
=> extracting sha256:7aad5892c3b7a86566b10bf19d08322b1961204271a158c9eac1ed1b
=> extracting sha256:4eb0115a421399a047666a321263977f31e779480dc8d8d8f9bfb35f92d4
=> extracting sha256:cccccc1cbfb3a839f853b1dda5a0fb46a5b0a4fd39fb38caf526778824
=> extracting sha256:3d4c31f992b20a72702988a430bcbcd0bea170e1f13bf8859f64d9171
[internal] load build context
=> transferring context: 1.21MB
=> [2/4] WORKDIR /app
=> [2/4] COPY app.py /app
=> [4/4] RUN pip install flask netifaces
=> exporting to image
=> writing image sha256:e781ba3a58ff402c8990b3ca939a77f57208516de31ee998ea58c991898ea0
=> pushing to docker.io/library/local-python-app
root@aditya-vm-copy:/home/aditya-vm/assignment/Python# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
local-python-app latest e781ba3a58ff 21 seconds ago 1.01GB
root@aditya-vm-copy:/home/aditya-vm/assignment/Python#
```

9. Docker Images created of both the **nginx** and **Python** app

```
Waiting for image sha256:193ef685bf35...
root@aditya-vm-copy: /home/aditya-vm/assignment/nginx# cd ..
root@aditya-vm-copy: /home/aditya-vm/assignment# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
local-nginx          latest             193ef685bf35       15 seconds ago     192MB
local-python-app     latest            e781ba3a58ff       About a minute ago  1.01GB
root@aditya-vm-copy: /home/aditya-vm/assignment#
```

10. We deployed it on **Virtual Machine** and generated the **Public IP** and Hit it directly on the server.

The screenshot displays the Microsoft Azure portal interface for a virtual machine named 'aditya-vm-copy'. The left sidebar shows navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Connect, Networking, Network settings, Load balancing, Application security groups, Network manager, Settings, Availability + scale, Security, Backup + disaster recovery, Operations, Monitoring, Automation, and Help. The main content area is divided into 'Essentials' and 'Properties' sections. The 'Essentials' section provides a quick overview of the VM's status (Running), location (Central US), and various links for management. The 'Properties' section is further divided into 'Virtual machine' and 'Networking' details. The 'Virtual machine' section lists attributes such as Computer name, Operating system, VM generation, VM architecture, Agent status, Agent version, Hibernation, Host group, Host, Proximity placement group, Colocation status, and Capacity reservation group. The 'Networking' section shows the Public IP address (135.233.97.229), Private IP address (10.0.0.4), and the Virtual network/subnet (aditya-vm-copy-vnet/default).

Virtual machine	
Computer name	aditya-vm-copy
Operating system	Linux (ubuntu 22.04)
VM generation	V2
VM architecture	x64
Agent status	Ready
Agent version	2.12.0.2
Hibernation	Disabled
Host group	-
Host	-
Proximity placement group	-
Colocation status	N/A
Capacity reservation group	-

Networking	
Public IP address	135.233.97.229 (Network interface aditya-vm-copy182_23)
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	aditya-vm-copy-vnet/default
DNS name	Configure

11. Final Result:



12. Attaching the screenshot of the **commands** used for the process:

```
root@aditya-vm-copy: /home/
└─ Container python_app Created
Attaching to nginx-1, python_app
nginx-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
nginx-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
nginx-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
nginx-1 | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
nginx-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolves.envsh
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
nginx-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
nginx-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
python_app | * Serving Flask app 'app'
python_app | * Debug mode: off
python_app | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
python_app | * Running on all addresses (0.0.0.0)
python_app | * Running on http://172.18.0.1:8080
python_app | * Running on http://172.18.0.3:8080
python_app | Press CTRL+C to quit
python_app | 172.18.0.2 - - [02/Nov/2024 19:00:48] "GET / HTTP/1.0" 200 -
python_app | 172.18.0.2 - - [02/Nov/2024 19:00:48 +0000] "GET / HTTP/1.1" 200 385 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36"
python_app | 172.18.0.2 - - [02/Nov/2024 19:00:49] "GET /favicon.ico HTTP/1.0" 404 -
python_app | 172.18.0.2 - - [02/Nov/2024 19:00:49] "GET /favicon.ico HTTP/1.1" 404 297 "http://135.233.97.229/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/130.0.0.0 Safari/537.36"
Gracefully stopping... (press Ctrl+C again to force)
[+] Stopping 2/2
└─ Container assignment-nginx-1 Stopped 0.3s
└─ Container python_app Stopped 3.7s
canceled
root@aditya-vm-copy: /home/aditya-vm/assignment# history
1 ls
2 cd ..
3 pwd
4 ls
5 cd home
6 ls
7 cd aditya-vm/
8 ls
9 cd assignment/
10 ls
11 cd Python/
12 ls
13 docker build -t local-python-app .
14 docker images
15 cd ..
16 ls
17 cd nginx/
18 cd ..
19 docker images
20 docker-compose up
21 history
root@aditya-vm-copy: /home/aditya-vm/assignment#
```