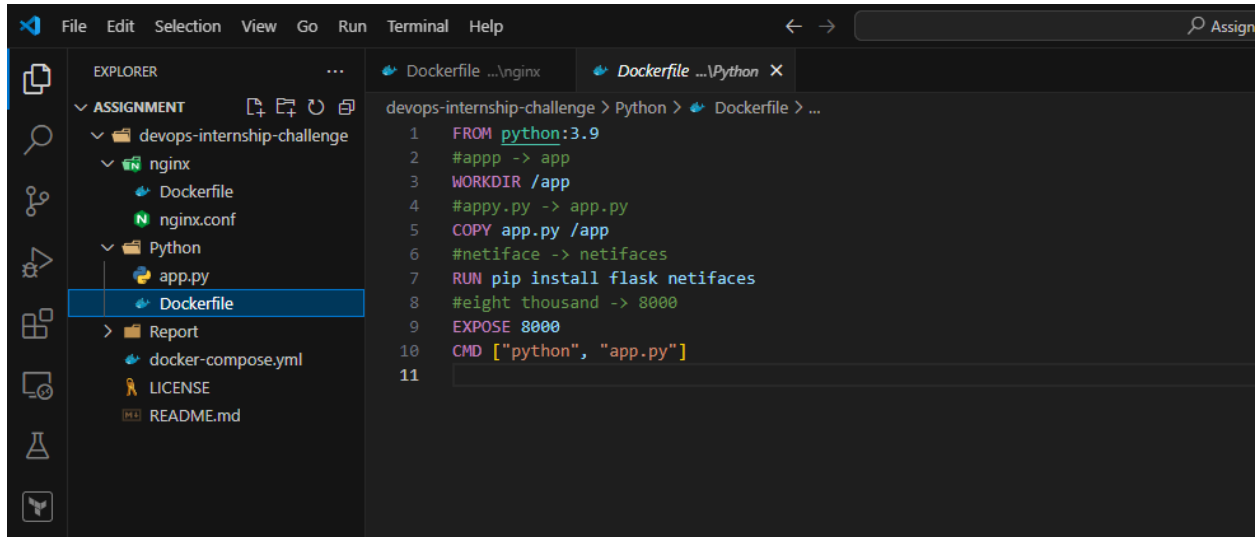


## Assignment:

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

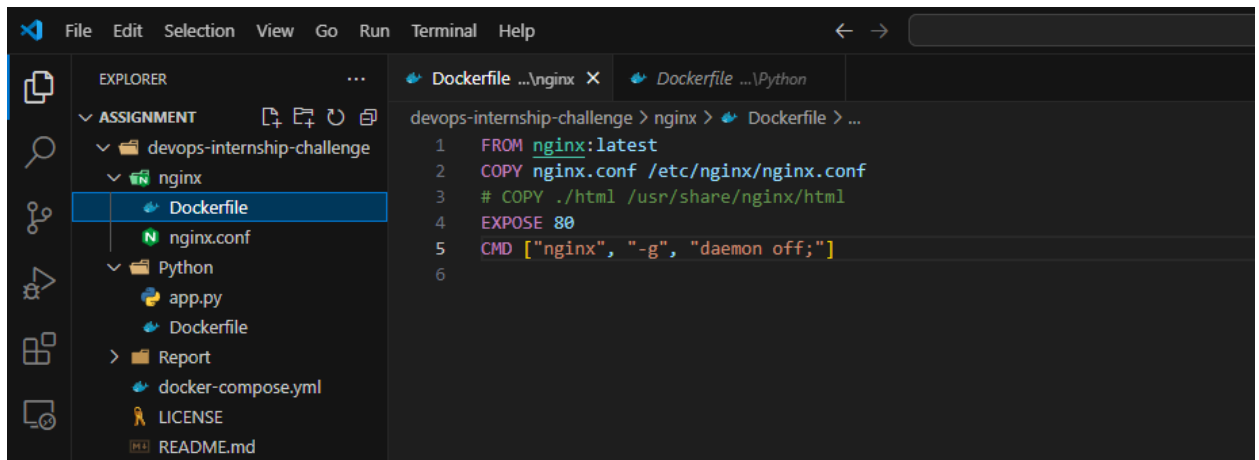
### Approach to the Problem:

1. First we **debug the dockerfile** of **nginx**, **Python** and **docker.compose.yml** file



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', 'python', and 'Report' folders. The 'python' folder is selected, and its 'Dockerfile' is open in the editor. The Dockerfile content is as follows:

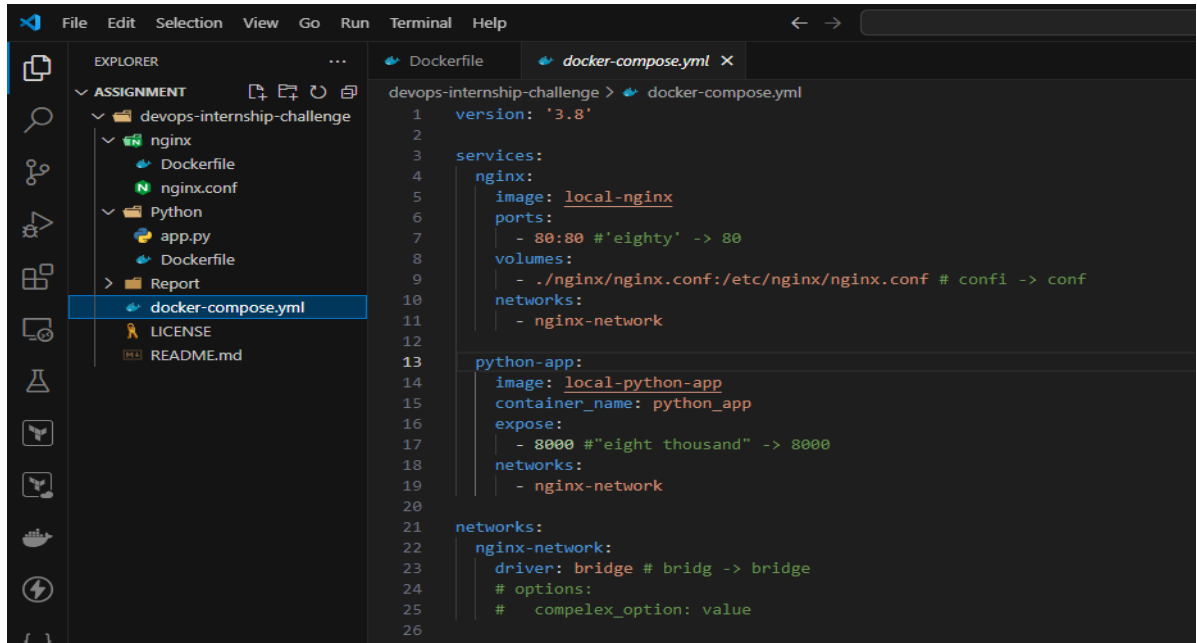
```
1 FROM python:3.9
2 #app -> app
3 WORKDIR /app
4 #app.py -> app.py
5 COPY app.py /app
6 #netiface -> netifaces
7 RUN pip install flask netifaces
8 #eight thousand -> 8000
9 EXPOSE 8000
10 CMD ["python", "app.py"]
11
```



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', 'python', and 'Report' folders. The 'nginx' folder is selected, and its 'Dockerfile' is open in the editor. The Dockerfile content is as follows:

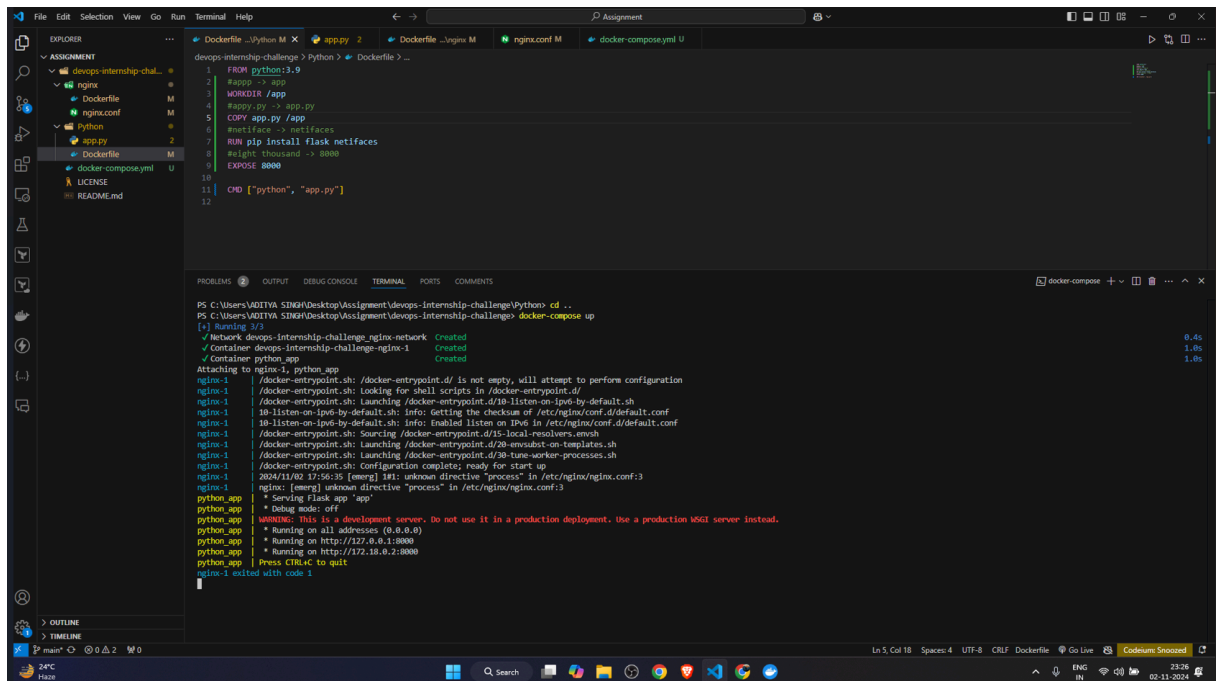
```
1 FROM nginx:latest
2 COPY nginx.conf /etc/nginx/nginx.conf
3 # COPY ./html /usr/share/nginx/html
4 EXPOSE 80
5 CMD ["nginx", "-g", "daemon off;"]
6
```

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



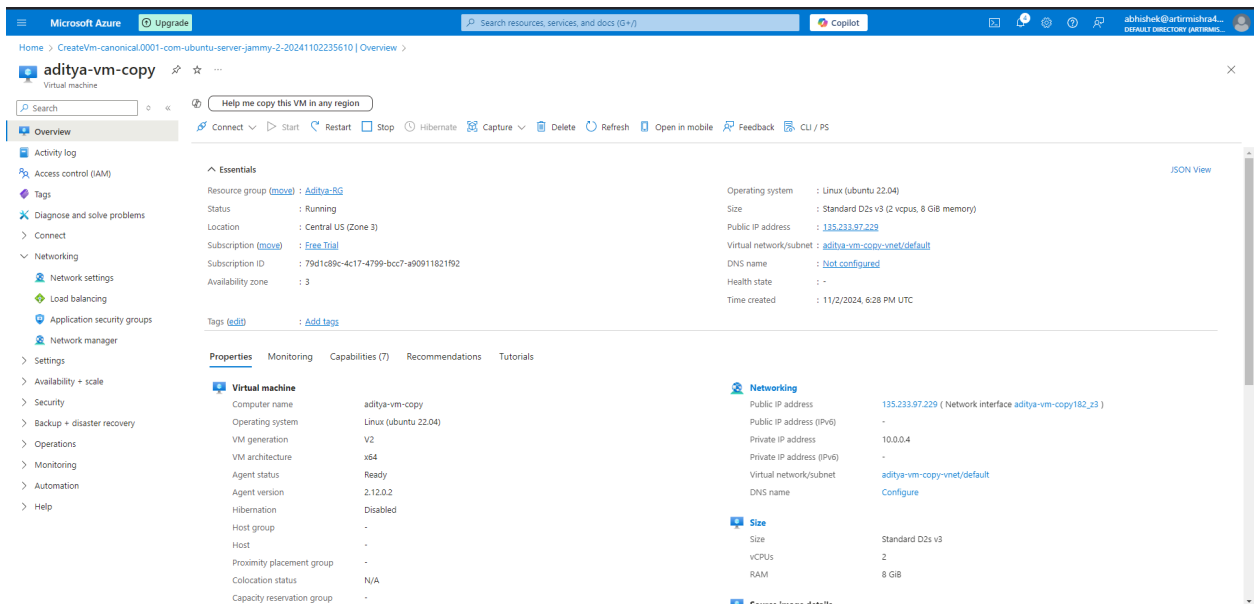
```
PS C:\Users\ADITYA SINGH\Desktop\Assignment\devops-internship-challenge> docker-compose up
[*] Running 1/3
✓ Network devops-internship-challenge/nginx-network Created
✓ Container devops-internship-challenge/nginx-1 Created
✓ Container python_app Created
Attaching to nginx-1, python_app
nginx-1 | /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: ~$ ssh aditya-vm@135.233.97.229
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.

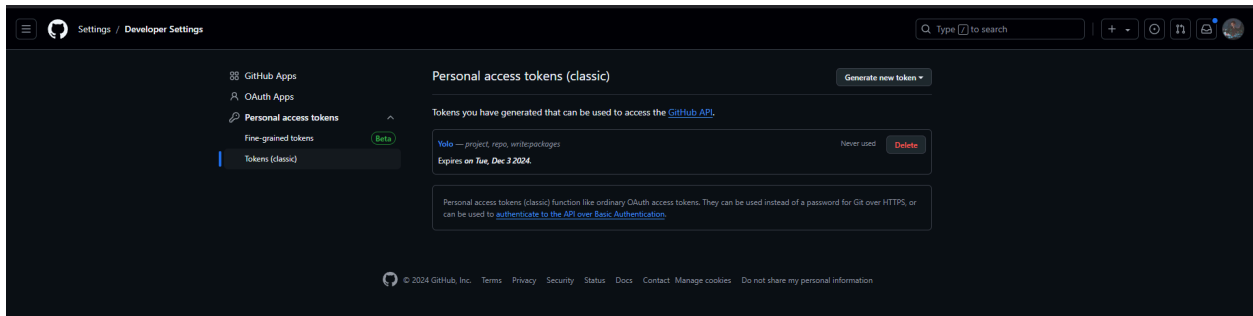
aditya-vm@aditya-vm-copy:~$
```

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

- Install Docker
- Install Docker Compose

```
root@aditya-vm-copy:/home# 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:ed8b9dd4e9f89c11f4b4d88a55f8c9f8e22796a2984a9380b15f627d9914895
=> resolve docker.io/library/python:3.9@sha256:ed8b9dd4e9f89c11f4b4d88a55f8c9f8e22796a2984a9380b15f627d9914895
=> sha256:2de9023d8992f28ade7a08fa08e04fca469922938e12778f840859a33a62 6.38kB / 6.38kB
=> sha256:7d98d813d84f6297a57721088a0881378343a8f1b2db66c121486819171885b 49.56kB / 49.56kB
=> sha256:da882d85c968baeca9d39869f9e2cbb3dc8b0d4e27f413bfbb2f2c3d6855988 24.85kB / 24.85kB
=> sha256:7aade5992c397f8566810bef20a08382019612482c1a158c98a6c1d1b 64.39kB / 64.39kB
=> sha256:8e8b504e49f88c11f4b4d88a55f8c9f8e22796a2984a9380b15f627d9914895 10.35kB / 10.35kB
=> sha256:988816fef8aa8a957c9ac8bca4892951bdca186c77f1578d529431cc38c5c61b4 2.32kB / 2.32kB
=> sha256:ad1c7fc397f95c89fc2678058f6a0f6cc4883971d85768532fc20d894b153d3 211.27kB / 211.27kB
=> sha256:8a8a115a021399a647666a3212b3977f31d779a80dcad8d8f9bbfb35f92e4 6.10kB / 6.10kB
=> extracting sha256:7d98d813d84f6297a57721088a0881378343a8f1b2db66c121486819171885b
=> sha256:cccc6c1c0bf8a8339708381dda5a0fba6a5b0aaf4d10fb3dcafc520778024 19.60kB / 19.60kB
=> sha256:63ca18f8933247f92980a5105bcbdc2ba1c70e1513f8d8859f8f091071 280B / 280B
=> extracting sha256:da882d85c968baeca9d39869f9e2cbb3dc8b0d4e27f413bfbb2f2c3d6855988
=> extracting sha256:7aade5992c397f8566810bef20a08382019612482c1a158c98a6c1d1b
=> extracting sha256:ad1c7fc397f95c89fc2678058f6a0f6cc4883971d85768532fc20d894b153d3
=> extracting sha256:4ab08115a021399a647666a3212b3977f31d779a80dcad8d8f9bbfb35f92e4
=> extracting sha256:cccc6c1c0bf8a8339708381dda5a0fba6a5b0aaf4d10fb3dcafc520778024
=> extracting sha256:63ca18f8933247f92980a5105bcbdc2ba1c70e1513f8d8859f8f091071
=> [internal] load build context
=> transferring context: 1.21kB
=> [2/4] WORKDIR /app
=> [3/4] COPY app.py /app
=> [4/4] RUN pip install flask netifaces
=> exporting to image
=> exporting layers
=> writing image sha256:e781ba3a58ff602c8990b3ca8939a7f7f757206516de31ee998ea50c091698ea9
=> using image for library/cache/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#
```

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 token from GitHub to access the Private Repositories.



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

```
> docker image ls
=> naming to docker.io/library/local-nginx
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 but we did not create the Virtual Machine directly from the portal, we created it using **Terraform(IaC)** tool.

### Steps:

1. Read the documentation of Azure Virtual Machine Creation
2. Used Terraform for the same purpose.
3. terraform init -> terraform plan -> terraform apply

```
resource "azurerm_virtual_machine" "main" {
  name                        = "${var.prefix}-vm"
  location                  = azurerm_resource_group.example.location
  resource_group_name       = azurerm_resource_group.example.name
  network_interface_ids     = [azurerm_network_interface.main.id]
  vm_size                   = "Standard_DS1_v2"

  # Uncomment this line to delete the OS disk automatically when deleting the VM
  # delete_os_disk_on_termination = true

  # Uncomment this line to delete the data disks automatically when deleting the VM
  # delete_data_disks_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "0001-com-ubuntu-server-jammy"
    sku       = "22_04-lts"
    version   = "latest"
  }
  storage_os_disk {
    name              = "myosdisk1"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
  os_profile {
    computer_name     = "hostname"
    admin_username    = "testadmin"
    admin_password    = "Password1234!"
  }
  os_profile_linux_config {
    disable_password_authentication = false
  }
  tags = {
    environment = "staging"
  }
}
```

Virtual Machine:

Microsoft Azure

Upgrade

Search resources, services, and docs (0+)

Copilot

abhishek@artimishra4...  
DEFAULT DIRECTORY (ARTIMISHRA4...)

Home > CreateVm-canonical.0001-com-ubuntu-server-jammy-2-20241102235610 | Overview >

aditya-vm-copy

Virtual machine

Search

Help me copy this VM in any region

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

Help

Connect

Start

Restart

Stop

Hibernate

Capture

Delete

Refresh

Open in mobile

Feedback

CLI / PS

Essentials

Resource group (move)

Status

Location

Subscription (move)

Subscription ID

Availability zone

Tags (edit)

aditya-rg

Running

Central US (Zone 3)

Free Trial

79d1c89c-4c17-4799-bcc7-a90911821f92

3

Add tags

Operating system

Size

Public IP address

Virtual network/subnet

DNS name

Health state

Time created

Linux (ubuntu 22.04)

Standard D2s v3 (2 vcpus, 8 GiB memory)

135.233.97.229

aditya-vm-copy-vnet/default

Not configured

-

11/2/2024, 6:28 PM UTC

Properties

Monitoring

Capabilities (7)

Recommendations

Tutorials

Virtual machine

Computer name

Operating system

VM generation

VM architecture

Agent status

Agent version

Hibernation

Host group

Host

Proximity placement group

Colocation status

Capacity reservation group

aditya-vm-copy

Linux (ubuntu 22.04)

V2

x64

Ready

2.12.0.2

Disabled

-

-

-

N/A

-

Networking

Public IP address

Private IP address

Private IP address (IPv6)

Virtual network/subnet

DNS name

135.233.97.229 ( Network interface aditya-vm-copy182\_23 )

-

10.0.0.4

-

aditya-vm-copy-vnet/default

Configure

Size

Size

vCPUs

RAM

Standard D2s v3

2

8 GiB

Customize linux details

11. Final Result:

135.233.97.229

VPN

Update

All bookmarks

IP Address: 172.18.0.2

MAC Address: 00:00:00:00:00:00

Username: Guest

Timestamp: 2024-11-02 19:00:48

Assignment completed successfully!

## 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-env.sh
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 -
nginx-1 | 172.18.0.2 - - [02/Nov/2024 19:00:48 +0000] "GET / HTTP/1.1" 200 305 "-" "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 -
nginx-1 | 172.18.0.2 - - [02/Nov/2024 19:00:49 +0000] "GET /favicon.ico HTTP/1.1" 404 207 "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
Container python_app Stopped
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#
```

-----Thank You-----