Devops Practical 4

A) Set up a virtual machine on a cloud provider or locally (e.g., using VMWare or Azure).

B) Configure networking concepts like virtual networks (Vnet), IP addresses, and ports.

C) Deploy the Flask/FastAPI app manually on the VM.

1) Download & install VMware Workstation Player (Windows)

1. Go to the VMware website and download **VMware Workstation Player** for Windows.
2. Run the installer (.exe) as Administrator. Accept defaults, click Next → Install.
3. Reboot if prompted.

2) Get the guest OS ISO

- Windows: download an official Windows ISO (Microsoft).
  Save ISO to a folder like C:\ISOs.

3) Create the VM (GUI steps — Player)

1. Open **VMware Workstation Player**.
2. Click **Create a New Virtual Machine**.
3. Choose **Installer disc image file (iso)** and browse to the ISO you downloaded.
4. Choose the guest OS type and version (VMware often detects it automatically).
5. Enter a **Name** and **Location** for the VM files.
6. Specify disk capacity (suggest **40–60 GB** for desktop OS) and choose **Store virtual disk as a single file** (simple).
7. Click **Customize Hardware…** and set:
   - **Memory (RAM):** 4096 MB (4 GB) for light desktop; 8192 MB (8 GB) for Windows 10/11 if you can spare it.
   - **Processors:** 1–2 cores (2 if your CPU has 4+ real cores).
   - **Network Adapter:** NAT (default) or Bridged (explained below).
   - **CD/DVD:** set to the ISO file if not already.
   - **USB Controller / Sound Card:** leave enabled unless you don't need them.
   - **Display:** enable 3D acceleration if you plan to run a GUI desktop.
8. Click **Close** → **Finish**.

4) Power on the VM & install the OS

1. Select the VM → click **Play virtual machine**.
2. The VM boots from the ISO — follow the normal OS installer steps (language, keyboard, username, partitioning).
   - When installer asks about disk choose to use the entire virtual disk (this only affects the virtual disk file).
3. After installation finishes, the VM may reboot. If it boots back into the installer, go to **Player → Removable Devices → CD/DVD (IDE) → Settings** and **disconnect** the ISO, then reboot the VM.

5) Install VMware Tools

Windows guest:

- In the VM window menu: **Player → Manage → Install VMware Tools** (or in Pro: **VM → Install VMware Tools**).
- In Windows, the VMware Tools installer auto-runs — follow the wizard and reboot when prompted.

6) Enable & use shared folders, clipboard, drag & drop

- **Shared folders:** In Player, open **Player → Manage → Virtual Machine Settings → Options → Shared Folders** → Add a host folder and set **Always enabled**.
  - In Windows guest the shared folder appears as a network drive or under **This PC**.
  - In Linux guests, after installing open-vm-tools-desktop, shared folders usually show up under /mnt/hgfs or appear in the File Manager.
- **Shared clipboard & drag & drop:** Player supports copy/paste and drag/drop if VMware Tools is installed. In Pro you can control these under VM **Settings → Options**.

7) Snapshots & backups

- **Workstation Pro**: use **VM → Snapshot → Take Snapshot…** before risky changes (updates, installs). Quickly revert if needed.
- **Workstation Player**: snapshots aren't available — to back up manually, shut down the VM and copy the VM folder (.vmx, .vmdk) to a safe place.

b)

Networking options

- **NAT (default):** VM shares host IP for outbound internet — easy and safe. Good for most beginners.
- **Bridged:** VM appears as a device on your LAN with its own IP — good for server testing or remote access.
- **Host-only:** VM ↔ host only (no internet) — useful for isolated test networks.
  Set these in **Player → Manage → Virtual Machine Settings → Network Adapter**.

C)

Step 1: Log in to your VM

Open VMware/VirtualBox → Start your Ubuntu VM → log in with your username & password.
Once logged in, open the **Terminal** (black screen icon).

Step 2: Update your system

Run these commands inside the VM terminal:

```
sudo apt update && sudo apt upgrade -y
```

Step 3: Install Python and pip

Ubuntu usually comes with Python3, but let's make sure:

```
python3 --version
pip3 --version
```

Step 4: Create a project folder

Inside your home directory:

```
mkdir flask_app
cd flask_app
```

Step 6: Install Flask

```
pip install flask
```

Step 7: Write a simple Flask app

Create a file:

```
nano app.py
```

Paste this in (beginner "Hello, world" app):

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "Hello, this is my first Flask app on a VM!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Save with **CTRL+O**, press **Enter**, then exit with **CTRL+X**.

Step 8: Run your Flask app

In the terminal:

```
python app.py
```

You'll see something like:

```
* Running on http://0.0.0.0:5000
```

Step 9: Access the app

- Inside the VM: open Firefox/Chrome → go to **http://127.0.0.1:5000**
  You should see: *Hello, this is my first Flask app on a VM!*
- From your **host computer** (Windows):
  1. Shut down the app (CTRL+C in terminal).
  2. Check your VM's IP address:
  3. ip addr show
     Look for something like inet 192.168.x.x.
  4. Start the app again:
  5. python app.py
  6. On your host machine, open a browser and go to:
     http://192.168.x.x:5000