**Steps to Deploy a Model on Microsoft Azure**

1. **Create a Microsoft Azure Account**
   * Go to <https://azure.microsoft.com> and sign up.
   * Provide your personal and payment (card) details if prompted.
2. **Access the Azure Portal**
   * After account creation, go to the [Azure Portal](https://portal.azure.com).
3. **Create a Virtual Machine**
   * From the dashboard, search for **"Virtual Machines"** in the top search bar.
   * Click on **"Create" → "Azure Virtual Machine"**.
4. **Configure the Virtual Machine**
   * Choose your preferred **region**, **Ubuntu** as the OS, and a size (like Standard B1s for small apps).
   * Under **Administrator Account**, select **SSH public key** and generate a new key pair if needed.
5. **Configure Networking**
   * In the **Networking** tab, add an **inbound port rule** to allow traffic on **port 7080**.
   * For testing purposes, you can allow "All traffic" (not recommended for production environments).
6. **Download SSH Key**
   * Download the private key (.pem or .ppk) if you created a new one.
   * Save it securely in your deployment directory.
7. **Connect to the Virtual Machine**
   * Once the VM is running, go to the **"Connect"** tab.
   * Choose **"SSH"** and look for the **"Native SSH"** option
   * Copy the provided SSH connection command.
   * It will look something like this:
   * e.g : ssh -i azure\_key.pem azureuser@<your-vm-ip>
8. **Open Terminal or CMD and Connect**
   * Navigate to the folder containing the .pem file.
   * Run the SSH command to connect:

ssh -i azure\_key.pem azureuser@<your-vm-ip>

* + Type yes if prompted to trust the connection.

1. **Update the VM and Install Required Packages**

sudo apt update

sudo apt install python3

sudo apt install unzip

1. **Upload Your Deployment Files to the VM**

* Use SCP or SFTP to upload the deployment.zip file. Example using SCP:

sftp -i azure\_key.pem deployment.zip azureuser@<your-vm-ip>:/home/azureuser/

1. **Unzip and Setup Project**

unzip deployment.zip

cd deployment

ls # Confirm your files are there

pip install -r requirements.txt

If you face issues, try:

pip install -r requirements.txt --break-system-packages

1. **Ensure your Flask App Runs on Port 7080**

* In your main.py, make sure it ends with:

app.run(host="0.0.0.0", port=7080)

1. **Run Your Flask App**

python3 main.py

1. **Access Your App in a Browser**

* Use your VM’s **public IP** and **port 7080**:

http://<your-vm-ip>:7080

Example:

http://20.123.45.67:7080

**Notes**

* To edit a file on the server:

vi <filename>

* + Press i to enter **INSERT mode**
  + Make your changes
  + Press Esc, then type :wq to **save and exit**