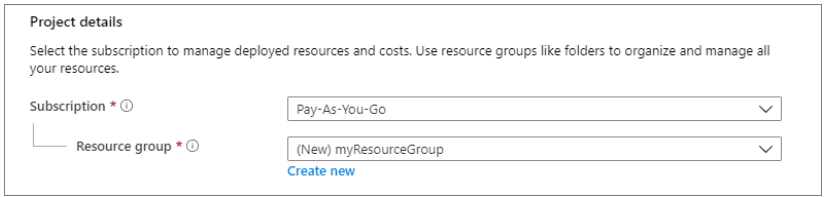
**Web App Deployment Using Linux VM in Azure**

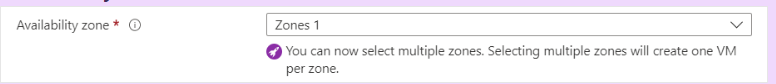
**Step 1-** Select the subscription and create a resource group.

****

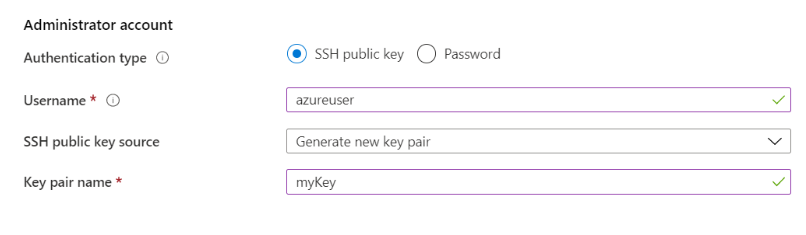
**Step 2-** Give a name to the VM of your choice, select the image option, choose the size (you might get an option of choosing a zone a to create VM so you can do that as well)

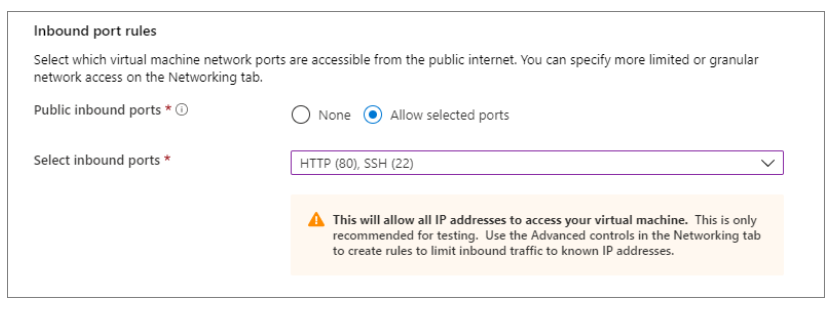
A screenshot of a computer

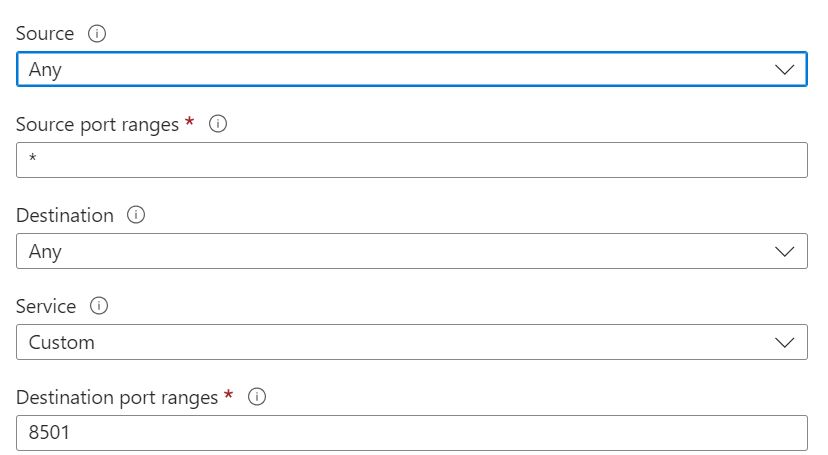
Description automatically generated



**Step 3-** Under the Administrator account, select SSH public key, enter “azureuser” for the Username, leave the default of “Generate new key pair” for the SSH public key source, and enter a name for the Key pair name**.**



**Step 4-** Under Inbound port rules > Public inbound ports, choose Allow selected ports and then select SSH (22) and HTTP (80) from the drop-down. Here we can choose an option based on security. We can specify specific ports for the app to run.  


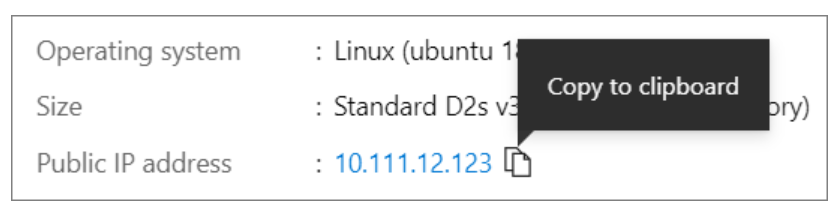
**For example-** Go to the network security group and change the Inbound rules like we have opened port 8501 here.

**Step 5-** Leave the remaining defaults and then select the Review + Create button at the bottom of the page.

**Step 6-** On the Create a Virtual Machine page, you can see the details about the VM you are about to create. When you are ready, select Create.

**Step** **7-** When the Generate new key pair window opens, select Download private key and create the resource. Your key file will be downloaded as myKey.pem. Ensure you know where the .pem file was downloaded; you will need the path to it in the next step.

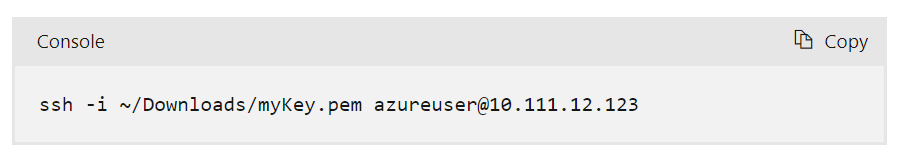
**Step 8-** When the deployment is finished, select "Go to resource," then copy the public IP address of your new VM to your clipboard.



**Create SSH connection with VM**

**Step 9-** If you are on a Windows machine, open a PowerShell prompt.

**Step 10-** Open an SSH connection to your virtual machine at your prompt. Replace the IP address with the one from your VM (from the clipboard) and replace the path to the .pem with the path to where the key file was downloaded.

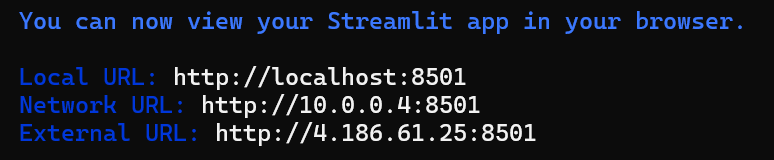


**Step 11-** Now, upload your zipped folder in the VM and unzip it.

**Step 12-** Pip install the requirements.txt

**Step 13-** Run you streamlit code file using streamlit run <name>.py

**Step 14-** The External URL will be the link to your deployed app.

****

**Note:- The above demonstration is for the Portal deployment. We can perform the same thing using CLI and PowerShell also. And we can use Windows as well instead of Linux.**

**Reference link for Azure VM Deployment:** [**https://learn.microsoft.com/en-us/azure/virtual-machines/linux/quick-create-cli**](https://learn.microsoft.com/en-us/azure/virtual-machines/linux/quick-create-cli)

**Deployment Using App Services in Azure**

**Step 1**- Create an App Service Plan in Azure.

a) Log in to the Azure portal, navigate to the App Service Plans section, and click on the “+ Create” button. Provide a name, and select the appropriate subscription, resource group, and operating system for your plan. Choose a pricing tier that fits your needs and select the desired region.

b) Click on the “Review + Create” button to review your App Service Plan settings, then click on the “Create” button to finalize and create the App Service Plan.

A screenshot of a service plan

Description automatically generated

**A screenshot of a computer

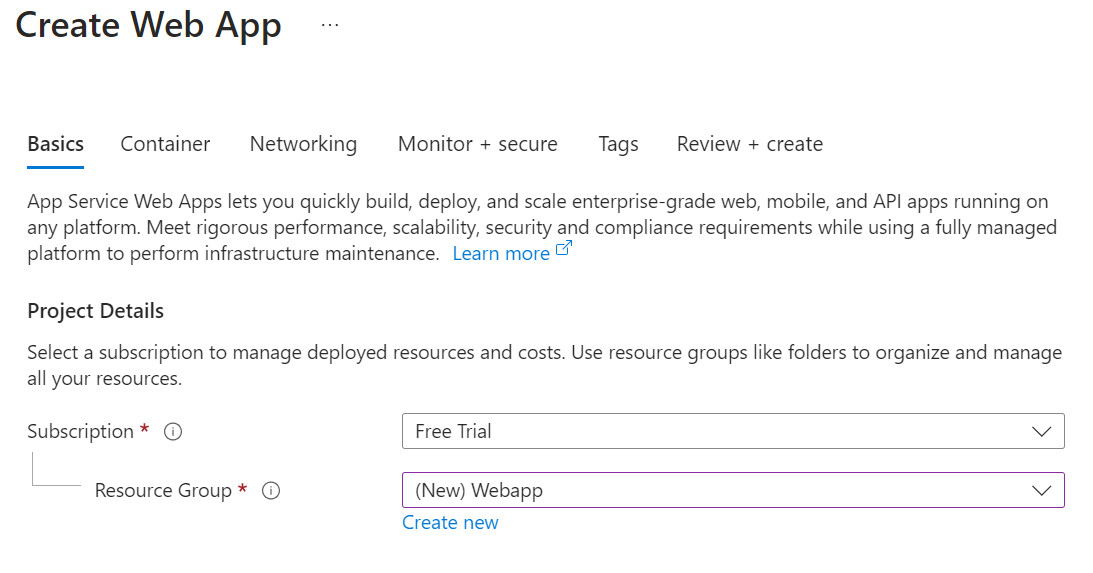
Description automatically generated**

**Step 2-** App services section

a) Once the App Service Plan is created, navigate to the App Services section in the Azure portal and click on the “Create” button to start creating a new App Service.

b) Provide a name, select the subscription, resource group, and the App Service Plan you created in step 1. Chose runtime stack based on the requirement, and here we are choosing python. Click "Next" with the default values.

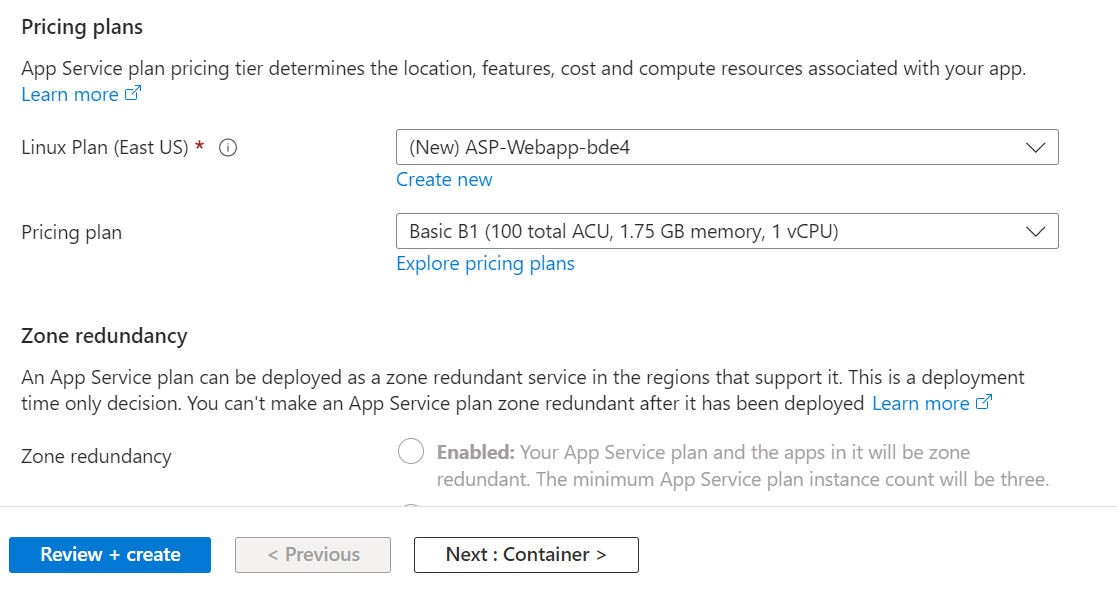
c) In the “Review + Create” tab, review the settings and then click on the “Create” button to finalize and create the App Service. Refer to the screenshots below for guidance.



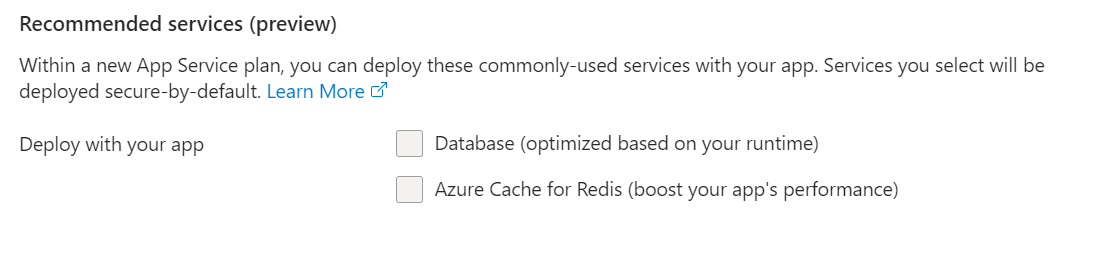
Chose how you want to publish the Container, Code or app and choose pricing plan (shown below)

A screenshot of a computer

Description automatically generated



Azure has a couple of recommendations, if you need click from here and click on Next.

**Note:** If you are using a container you must select your preferred source for container image and other settings. With container the step 3 below will be different instead of Deployment it will be container.

**Step 3-** Enable if you want continuous deployment.

**A screenshot of a computer

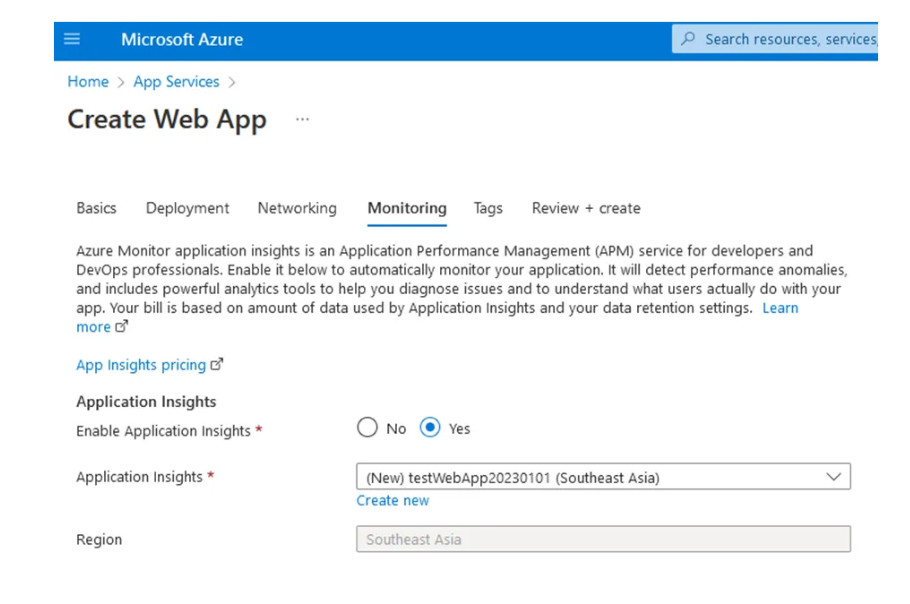
Description automatically generated**

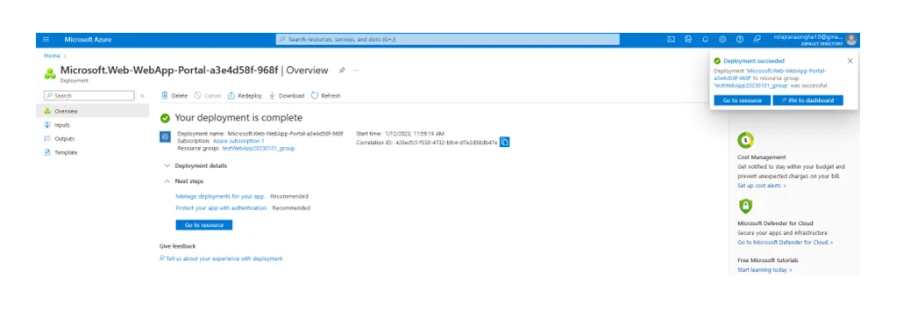
**Step 4-** Networking (Use default settings unless you have a specific requirement)

**A screenshot of a computer

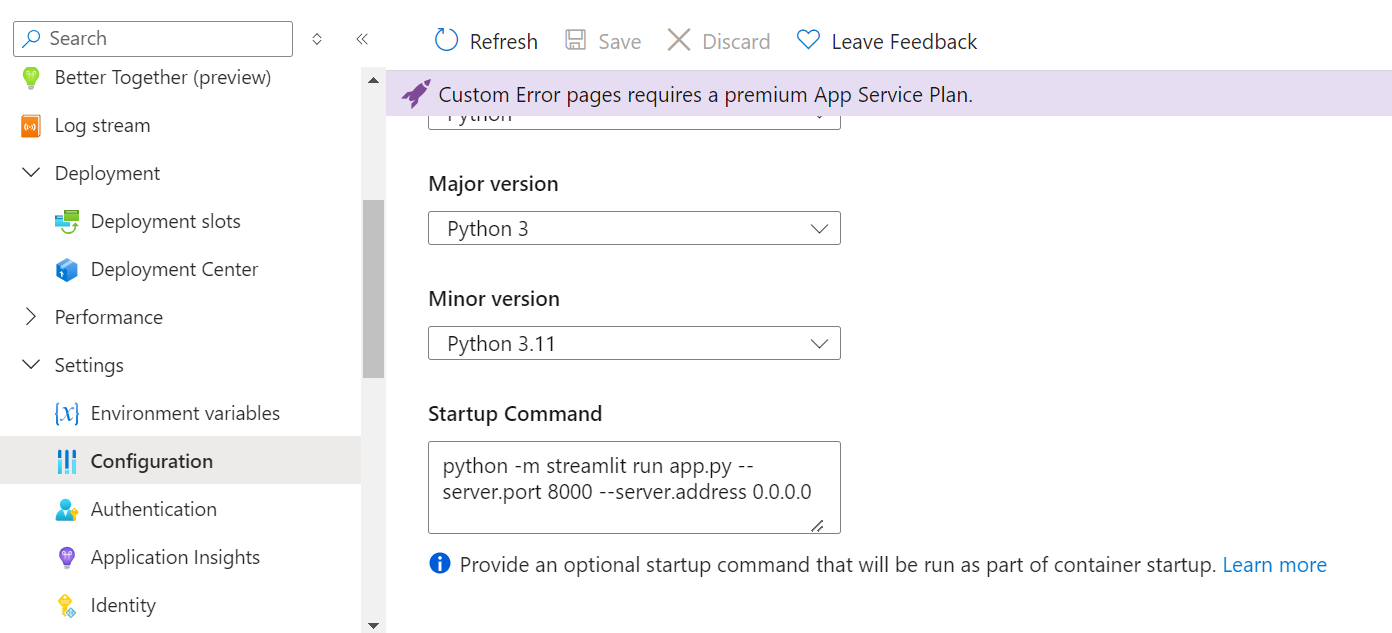
Description automatically generated**

**Step 5-** Monitoring (Use default settings unless you have a specific requirement)

****

**Step 6-** Review+ Create, it will start the deployment, and once the deployment is done follow the next steps below. ****

Step 7- Go to the app service you created>settings>configuration, select major,minor version, and give a startup command.(In our case it’s a streamlit app)



**Step 8-** Enable/Disable platform settings (Auth Publishing credentials, FTP state, Debugging and incoming client certificates, etc) and save.

**Step 9-**

In VS Code, open your folder with all the dependencies file and the .py notebook of your web app.

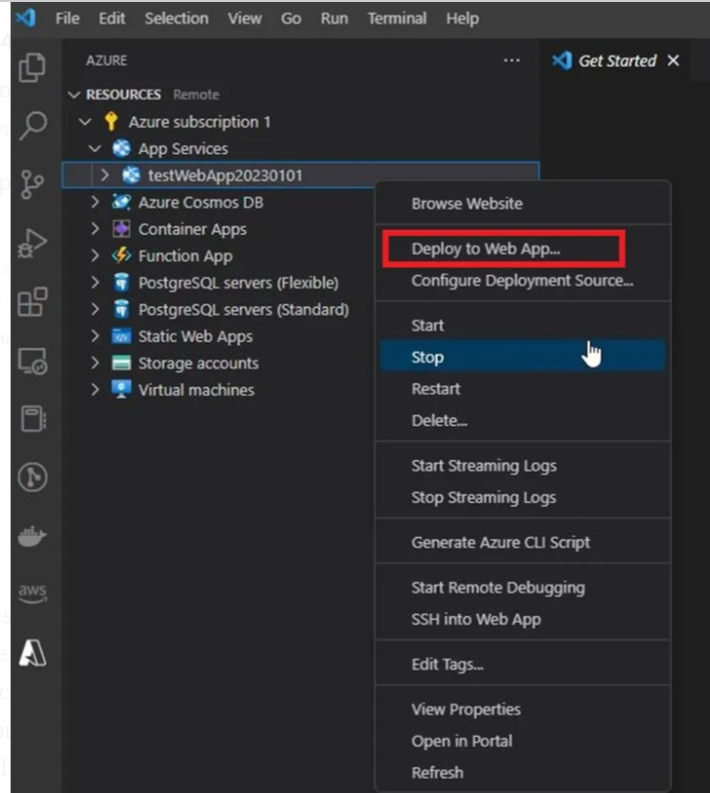
**Step 10**- Install the Azure App Service extension and Azure account in Visual Studio Code by searching for “Azure” in the Extensions marketplace.

1. Go to extensions search Azure Account install it.
2. Search Azure App services and install it.

A screenshot of a computer

Description automatically generated

**Step 11-** Right-click on the project in the Explorer pane and select “Deploy to Web App”.



**Step 12-** Test the application by navigating to the publicly available endpoint provided after the deployment is complete. Open your browser and go to the endpoint to verify that your application is running correctly.

**Reference link for Azure AppService Deployment:** [**https://medium.com/@nirajranasinghe/hosting-your-first-web-application-on-azure-app-service-a-comprehensive-introduction-to-setting-up-90008c84e400**](https://medium.com/@nirajranasinghe/hosting-your-first-web-application-on-azure-app-service-a-comprehensive-introduction-to-setting-up-90008c84e400)

**Web App deployment using Azure CLI**

**Step 1-** Download and Install Azure CLI:

- Download Azure CLI from the official Microsoft website

- Install and launch it

**Step 2 -** Open Windows Command Prompt

**Step 3-** Start Azure CLI: azure-az

**Step 4-** Login to Azure (with MFA): az login --use-device-code , if MFA is not there az login will work as well.

**Step 5-** List Available Locations: az account list-locations --output table

**Step 6-** Create a Resource Group: az group create --name MyNewResourceGroup --location <location name>

**Step 7-** List Ubuntu Server Images:

az vm image list --publisher Canonical --offer UbuntuServer --all --output table

**Step 8-** Create a VM:

az vm create --resource-group MyNewResourceGroup --name MyUbuntuVM --image Canonical:UbuntuServer:18.04-LTS:latest --size Standard\_B1ls --admin-username azureuser --generate-ssh-keys --public-ip-sku Standard --location centralindia

**Step 9-** Get the VM's Public IP:

az vm show --resource-group MyNewResourceGroup --name MyUbuntuVM --show-details --query publicIps --output tsv

**Step 10-** Copy the IP address

**Step 11-** Connect via SSH:

ssh azureuser@<public-ip-address>

**Step 12-** Upload Zip File to VM (from local terminal):

rsync -avz -e ssh /path/to/your/local/zipped\_folder.zip azureuser@<public-ip-address>:~

**Step 13-** Verify Upload in SSH: ls ~

**Step 14-** Install Unzip:

sudo apt install unzip

**Step 15-** Unzip the Folder:

unzip -l your\_zipped\_folder.zip

**Step 16-** Navigate to Unzipped Directory:

- Use `pwd` to check current directory  
 - Use `ls` to list folders in the directory

**Step 17-** Update Package List (recommended):

sudo apt update

**Step 18-** Install Requirements:

- If pip is not available: sudo apt install python-pip

- Then: pip install -r requirements.txt

**Step 19-** For Basic/Free Versions (if facing errors):

a.) Create a virtual environment:

b.) sudo apt install python3-venv

c.) python3 -m venv streamlit\_env

d.) source streamlit\_env/bin/activate

e.) pip install --upgrade pip

f.) pip install --no-cache-dir streamlit

**Step 20-** Run App in Background:

nohup streamlit run your\_app.py &

Note: Opening a Custom Port

If you encounter a network error, you may need to open a custom port (e.g., 8502):



az network nsg rule create --resource-group MyNewResourceGroup --nsg-name YourNSGName --name AllowStreamlit --protocol tcp --priority 1001 --destination-port-range 8502 --access Allow