

Aditya Kumar Singh
Username - CT_CSI_DV_4150
Email - adityaaks868@gmail.com

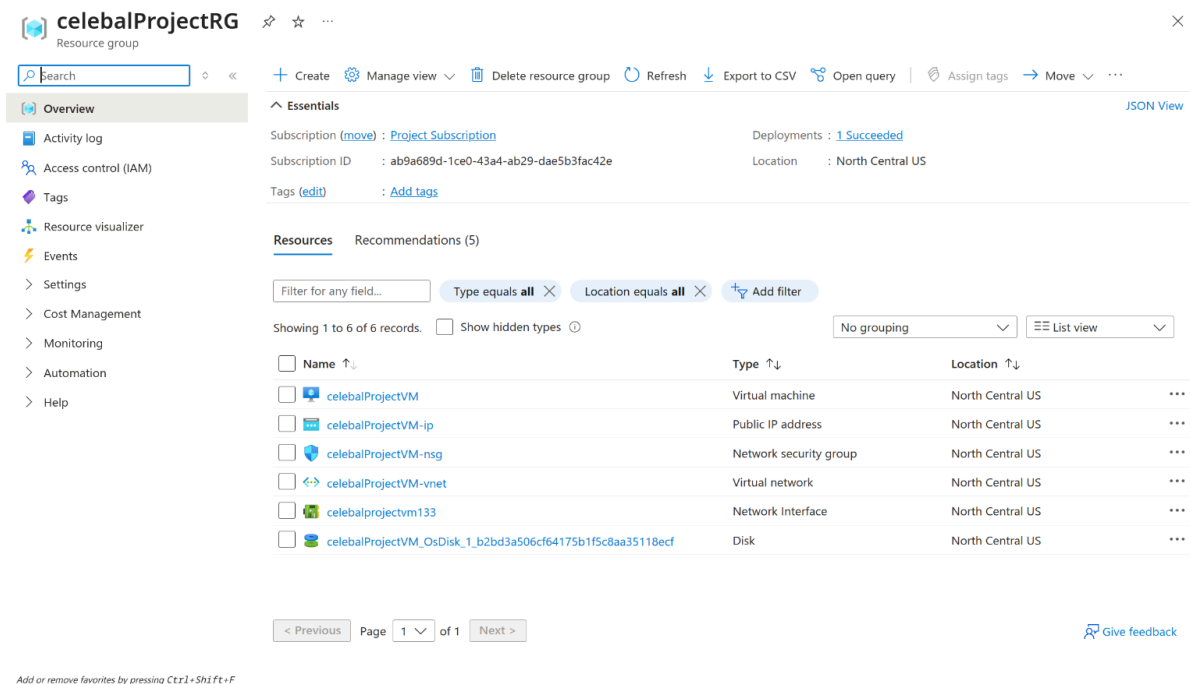
This document provides a comprehensive walkthrough of the steps I followed while working on this Docker-based deployment project. It includes the used commands along with relevant screenshots highlighting the GUI components and interactions performed throughout the process. These visuals are intended to give a clear understanding of the overall workflow, from setup to deployment.

In addition to this document, a separate file containing all the command-line interface (CLI) interactions has been prepared and will be uploaded to the project's GitHub repository. This file will allow the evaluator to verify the exact commands used and their corresponding outputs.

Direct link to the deployed application - <http://172.183.242.237/>

WORKFLOW

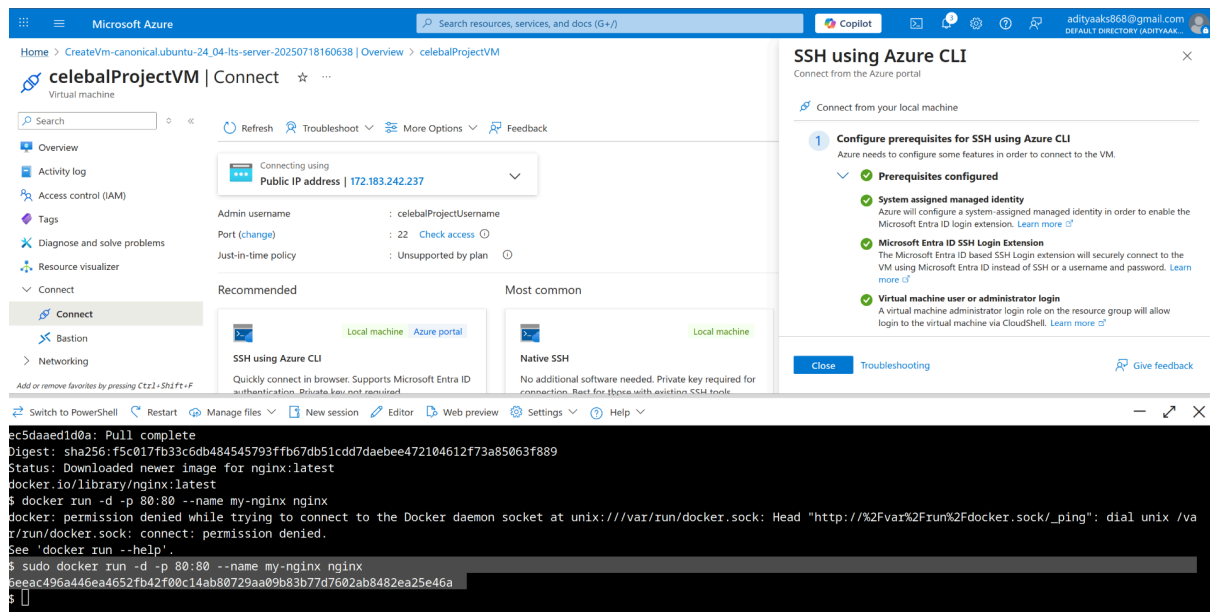
STEP 1 - Created a Resource Group named celebalProjectRG



The screenshot shows the Azure portal interface for the resource group 'celebalProjectRG'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Cost Management, Monitoring, Automation, and Help. The main content area displays the 'Overview' tab, which includes a search bar, action buttons (Create, Manage view, Delete resource group, Refresh, Export to CSV, Open query, Assign tags, Move), and a 'JSON View' link. Below this, the 'Essentials' section shows the subscription ID and location. The 'Resources' section displays a table of resources with columns for Name, Type, and Location. The table lists six resources, all located in North Central US.

Name	Type	Location
celebalProjectVM	Virtual machine	North Central US
celebalProjectVM-ip	Public IP address	North Central US
celebalProjectVM-nsg	Network security group	North Central US
celebalProjectVM-vnet	Virtual network	North Central US
celebalprojectvm133	Network Interface	North Central US
celebalProjectVM_OsDisk_1_b2bd3a506cf64175b1f5c8aa35118ecf	Disk	North Central US

STEP 2 - Created a VM named celebalProjectVM in the resource group celebalProjectRG. Allowed ports SSH(20) and HTTP(80)



STEP 3 - Connected to celebalProjectVM Virtual Machine via SSH

STEP 4 - Installed docker on the virtual machine using the Azure CLI

Commands used:

```
sudo apt install docker.io -y
sudo systemctl start docker
sudo systemctl enable docker
docker --version
```

STEP 5 - Pulled a docker image of nginx.

Command used:

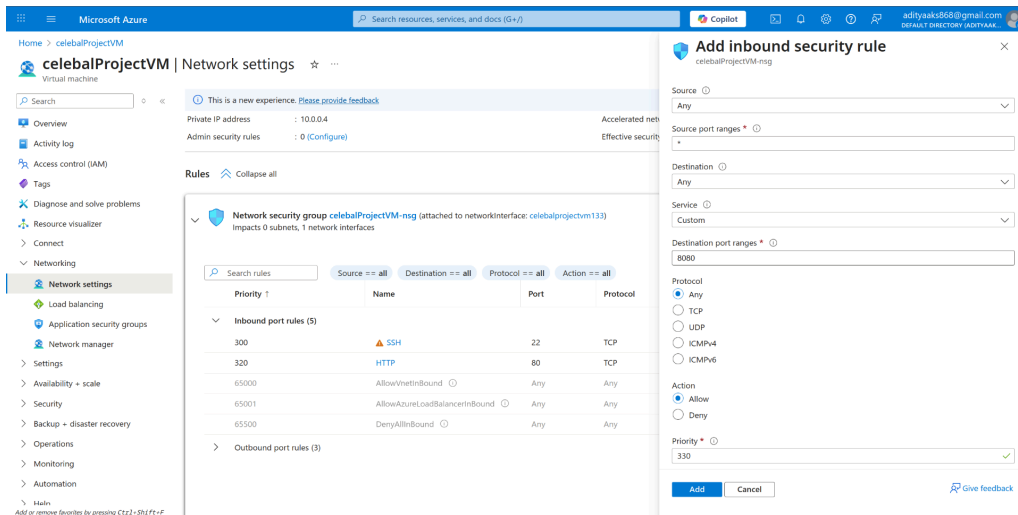
```
docker pull nginx
```

STEP 6 - Launched the container and exposed it on port 80. Made it accessible for everyone

Command used:

```
docker run -d -p 80:80 --name my-nginx nginx
```

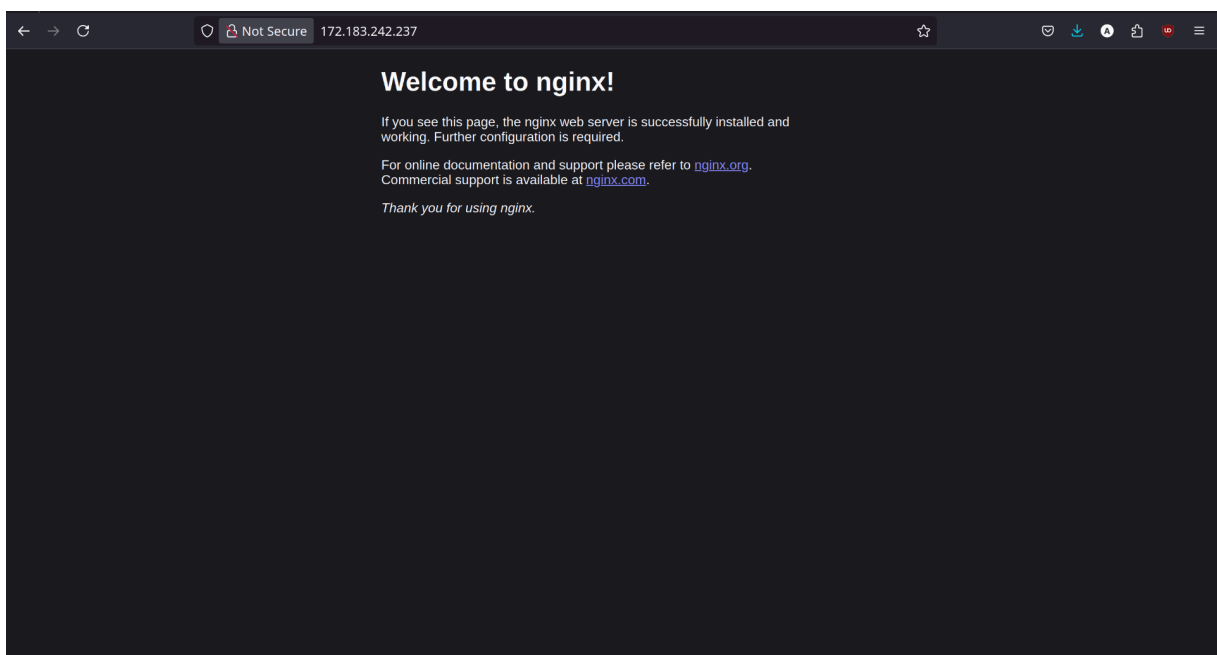
STEP 7 - Configured Azure networking for celebalProjectVM virtual machine



Added a new inbound port for 3000 (Node.js)

Inbound port rules (6)							
300	SSH	22	TCP	Any	Any	Allow	
320	HTTP	80	TCP	Any	Any	Allow	
330	AllowAnyCustom3000Inbound	3000	TCP	Any	Any	Allow	
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow	
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow	
65500	DenyAllInBound	Any	Any	Any	Any	Deny	
Outbound port rules (3)							

STEP 9 - Tested the deployed application through my browser using `http://<Virtual Machine IP>` which in my case was <http://172.183.242.237/>



STEP 10 - Monitoring and management of containers

See all running containers

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6eeac496a446	nginx	"/docker-entrypoint..."	4 days ago	Up 4 days	0.0.0.0:80->80/tcp, :::80->80/tcp	my-nginx

Scaling containers

```
$ sudo docker run -d -p 8080:80 --name nginx2 nginx  
f54159a1fea16eb900c98bdf3982a73a843adb2df1d6c7a1a72bc20ae5973e23
```

Updating Container

```
$ sudo docker pull nginx:latest  
latest: Pulling from library/nginx  
59e22667830b: Pull complete  
140da4f89dcb: Pull complete  
96e47e70491e: Pull complete  
2ef442a3816e: Pull complete  
4b1e45a9989f: Pull complete  
1d9f51194194: Pull complete  
f30ffbee4c54: Pull complete  
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb  
Status: Downloaded newer image for nginx:latest  
docker.io/library/nginx:latest  
$ sudo docker stop my-nginx  
my-nginx  
$ sudo docker rm my-nginx  
my-nginx  
$ sudo docker run -d -p 80:80 --name my-nginx nginx:latest  
3b862e47fa74d89b20a126092e4a42fdc59dd60daf37909536cc8360b4a0aad2
```