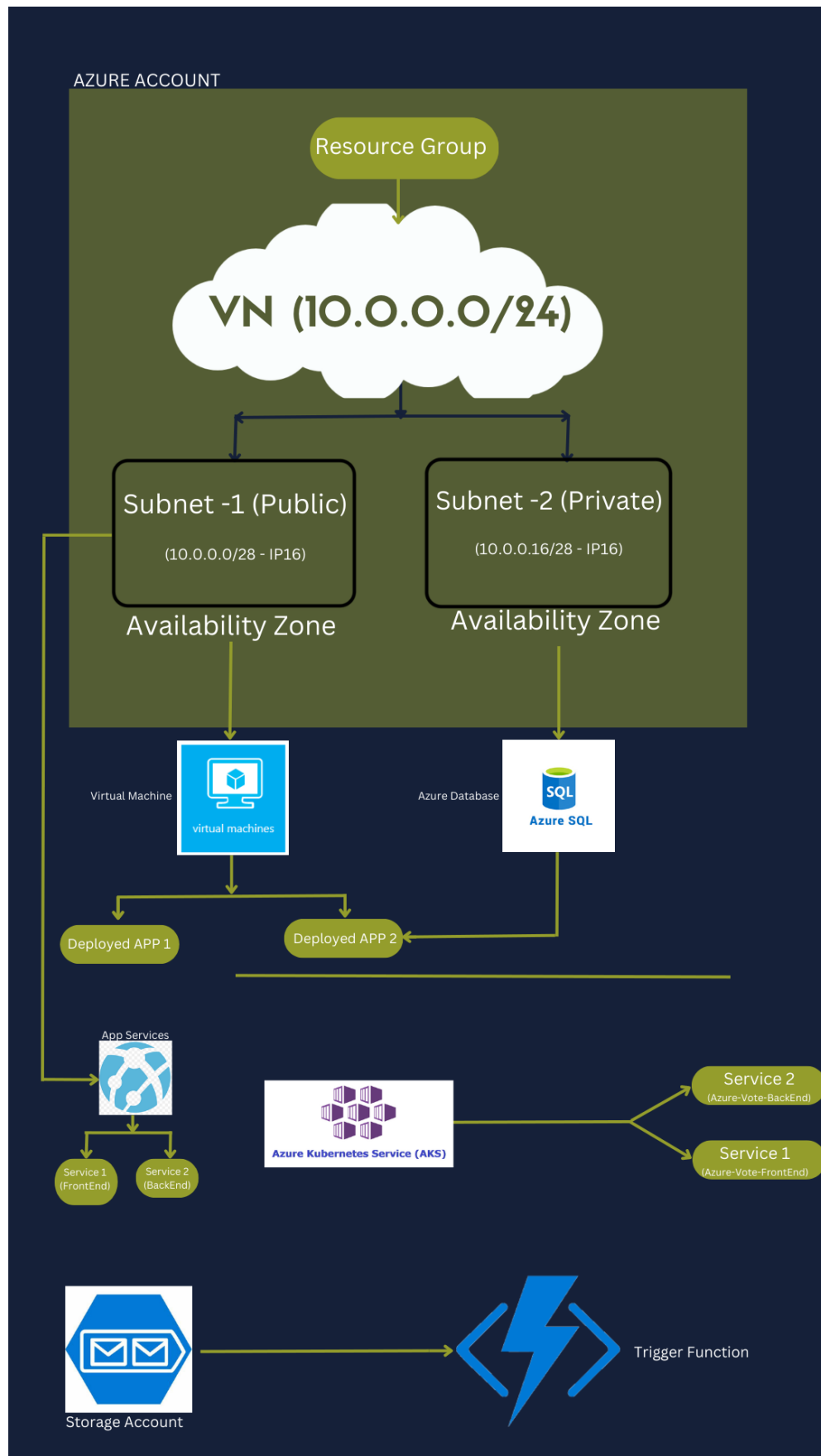


Azure Assignment

Architecture Diagram



Azure Virtual Network:

Azure Virtual Network (VNet) is a fundamental building block in Azure networking. It is a logically isolated network infrastructure provided by Azure that allows you to securely connect and control your Azure resources.

Azure Subnet:

A subnet, short for subnetwork, is a range of IP addresses within the address space of a VNet. It helps in further segmenting the network and allows you to organize and isolate resources within the VNet.

Virtual Machine:

In Azure, a virtual machine (VM) is a computing resource that runs an operating system and applications just like a physical computer. It provides a scalable and flexible way to deploy and manage your workloads in the cloud. Azure virtual machines are commonly referred to as Azure VMs.

AKS Cluster:

AKS (Azure Kubernetes Service) is a managed container orchestration service provided by Azure. It simplifies the deployment, management, and scaling of containerized applications using Kubernetes.

Azure App service:

Azure App Service is a fully managed platform-as-a-service (PaaS) offering from Azure that allows you to build, deploy, and scale web applications, mobile backends, and RESTful APIs quickly and easily.

Azure Blob Storage:

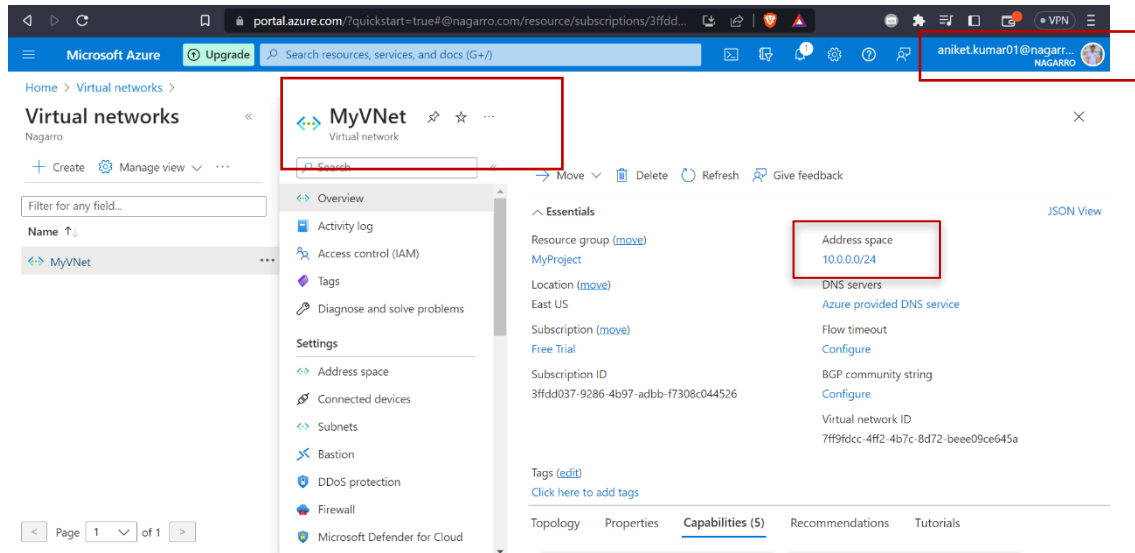
Azure Blob storage is a cloud-based object storage service that allows you to store and retrieve large amounts of unstructured data, such as text files, images, videos, and binary files.

Azure Function App:

Azure Function app is a serverless computer service that allows you to run small pieces of code, called functions, in the cloud.

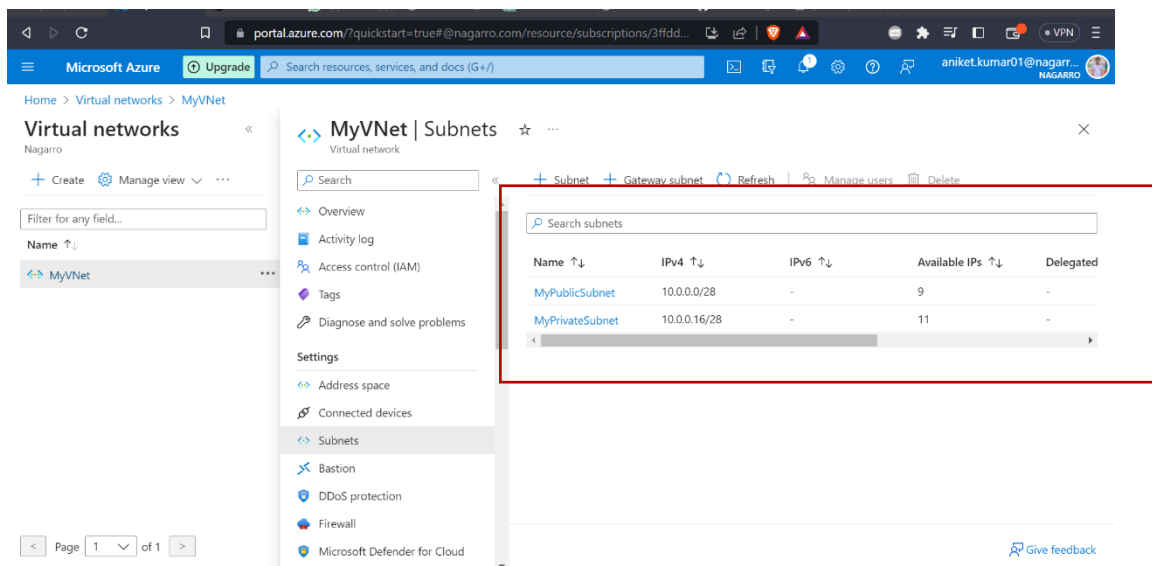
Project Screenshots

1. Virtual Network screenshots:

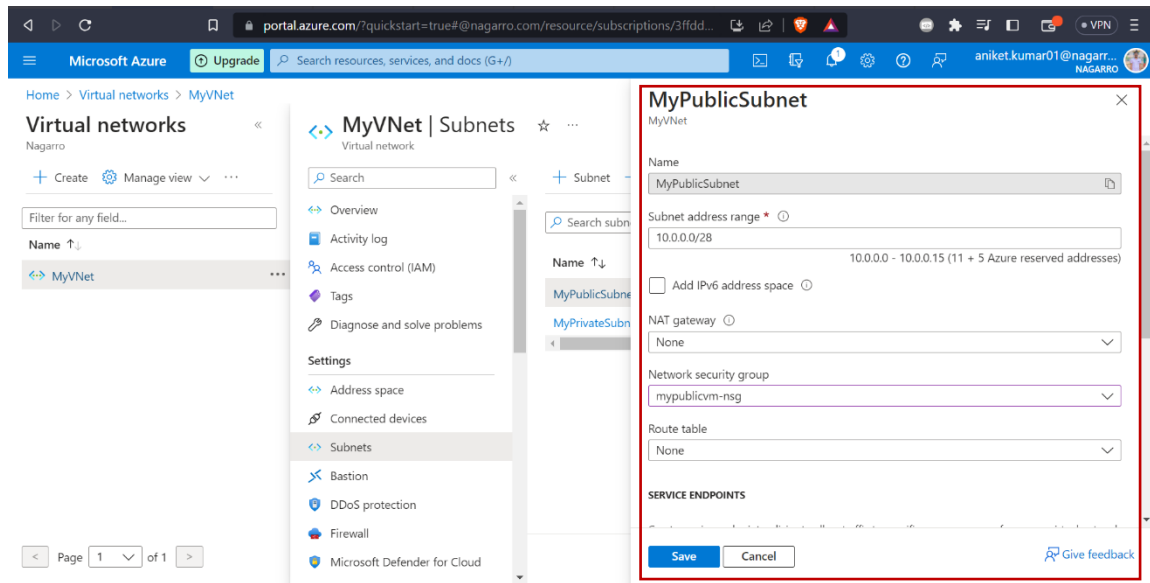


The screenshot is of Vnet that I have created.

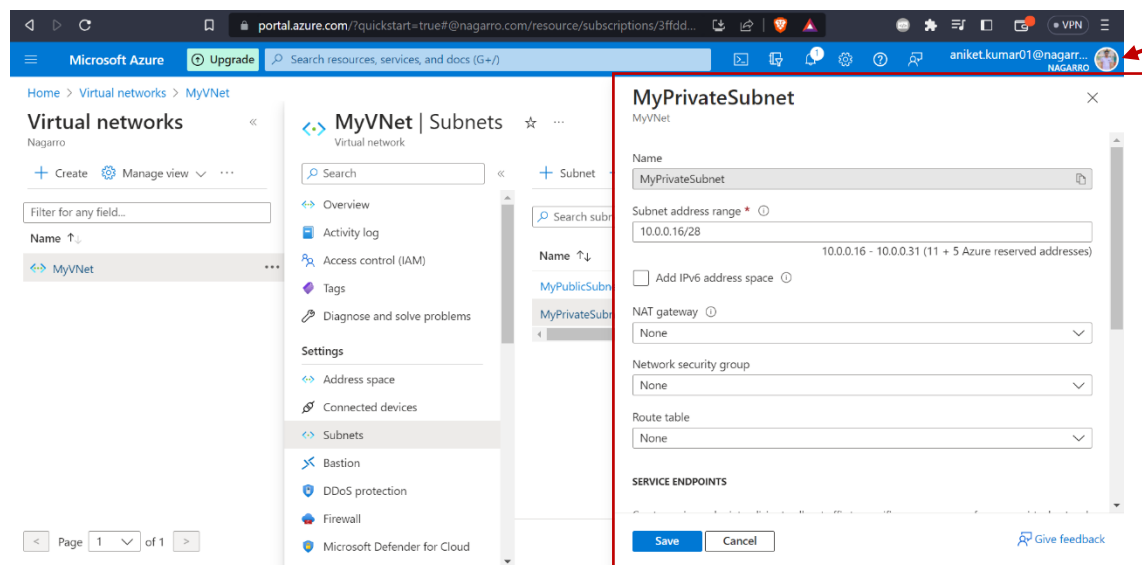
2. Subnets:



I created two subnets, Private and Public subnets. The private subnet is for databases and the public subnet is for deploying applications.

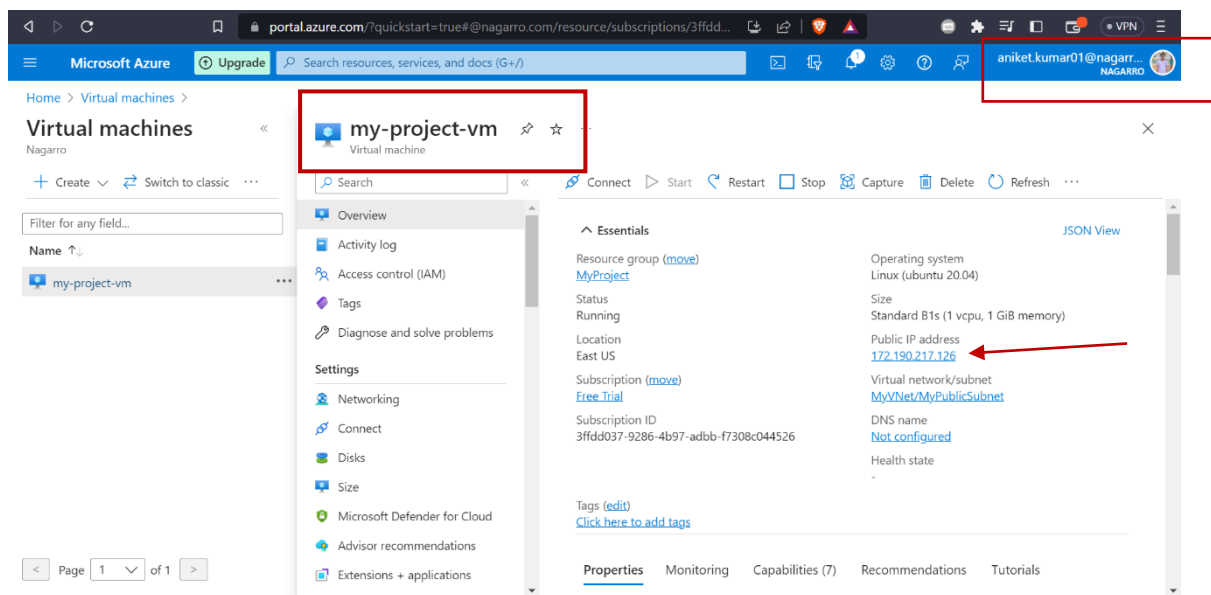


The screenshot is of a public subnet.

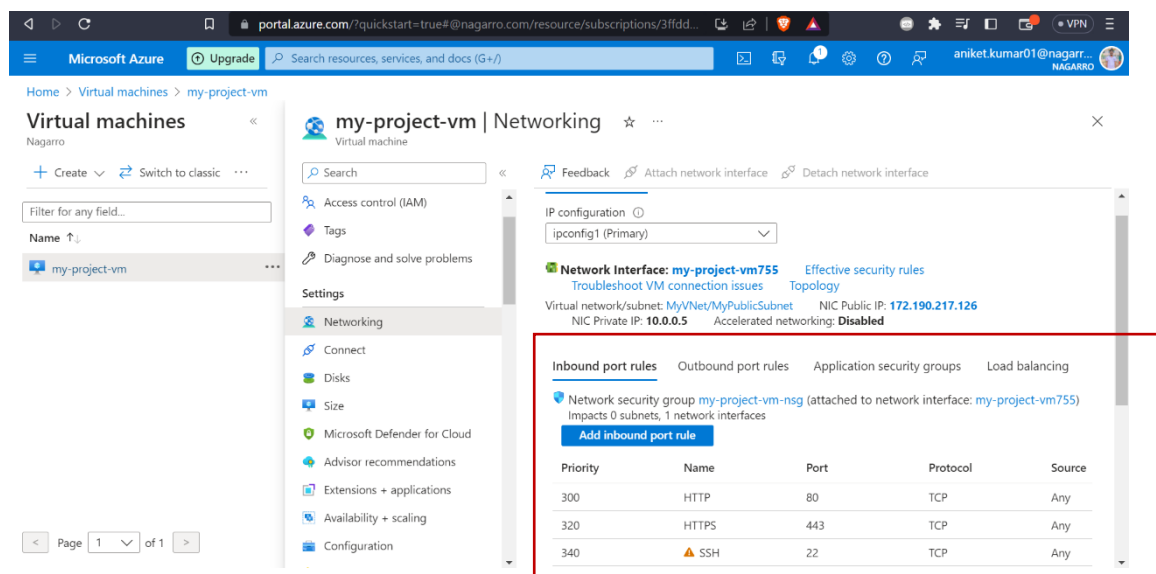


This screenshot describes a private subnet.

Virtual Machine Screenshots:



This screenshot describes the virtual machine I created. Here We can see that I created this virtual machine in my custom vnet and in public subnet.



This screenshot is of inbound rules of Virtual machine.

Databases screenshots:

The screenshot shows the Azure portal interface for the 'myspringdb' Azure Database for MySQL flexible server. The 'Essentials' section is highlighted with a red box, displaying the following details:

- Subscription (move): [Free Trial](#)
- Subscription ID: 3fdd037-9286-4b97-adbb-f7308c044526
- Resource group (move): [MyProject](#)
- Status: Available
- Location: East US
- Created On: 2023-07-11 08:20:55.6699929 UTC

Other visible details include the Server name (myspringdb.mysql.database.azure.com), Server admin login name (aniket), Configuration (Burstable, B1s, 1 vCores, 1 GiB RAM, 20 storag...), MySQL version (8.0), Availability zone (2), and Tags (edit).

I have connected this database with my backend application.

The screenshot shows the Azure portal interface for the 'myspringdb' Azure Database for MySQL flexible server, specifically the 'Networking' section. The 'Networking' section is highlighted with a red box, displaying the following details:

- Connectivity method: Private access (VNet Integration)
- Virtual network: [MyVNet/MyPrivateSubnet](#)
- Private DNS zone: [myspringdb.private.mysql.database.azure.com](#)

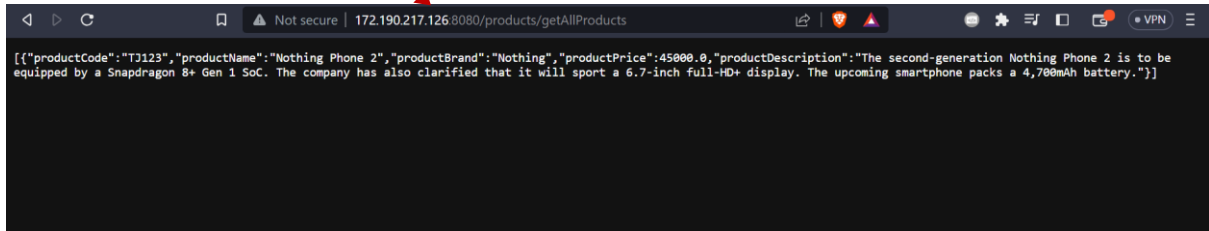
Other visible details include the Compute + storage section (Pricing tier: Burstable, Compute size: Standard_B1s (1 vCore, 1 GiB memory, 400 max iops), Storage: 20 GiB, IOPS: 360, Storage Auto-growth: Enabled) and the Backup section (Backup Type: Locally redundant, Retention period: 7, Earliest restore point: 2023-07-11 14:08:00.9637062 UTC).

I have created this database in a private subnet of my custom virtual network.

Public ip of VM

App working on Virtual machine Screenshots:

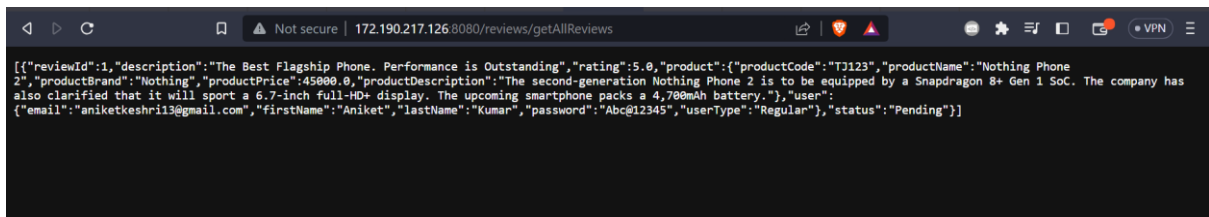
Backend:



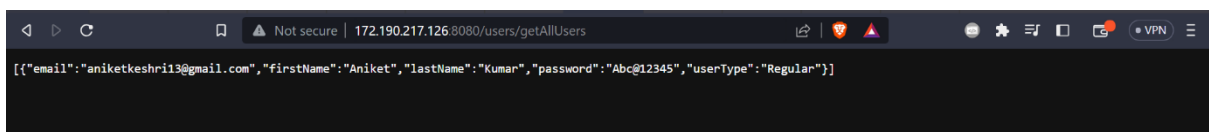
The screenshot shows all the product list in the database.



This screenshot shows working of Backend application on the public Ip of Virtual machine.



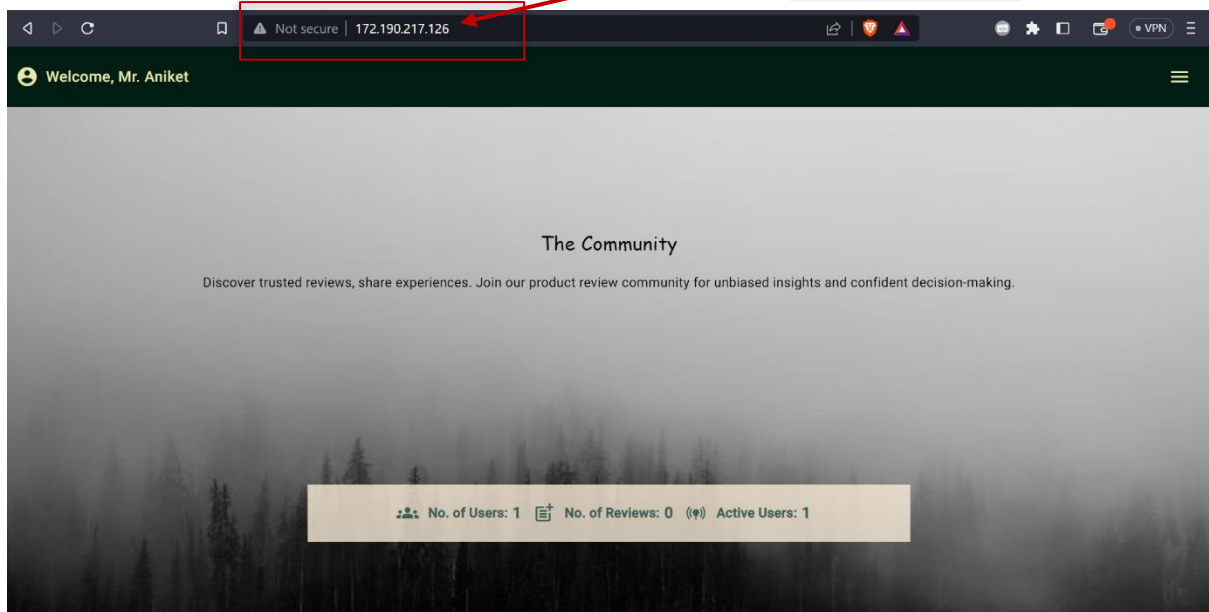
This screenshot shows all the list of review present in the database.



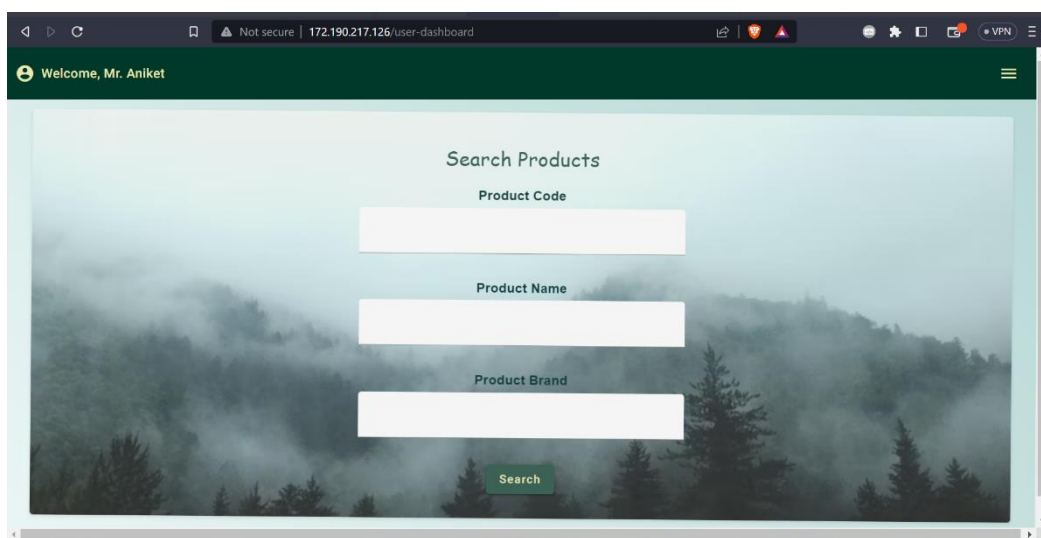
This screenshot shows the registered user.

Frontend Running on Virtual Machine:

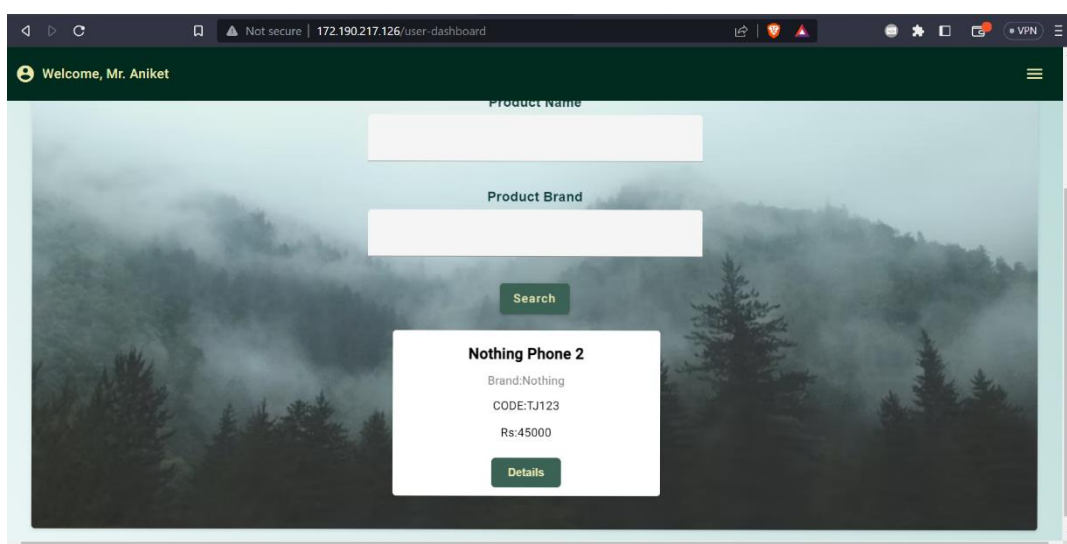
Public ip of vm



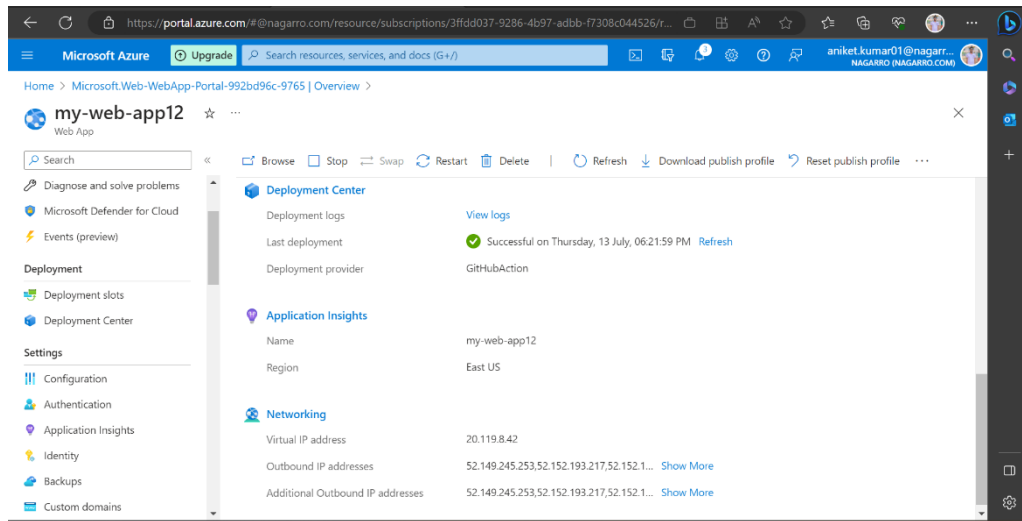
This is the home page.



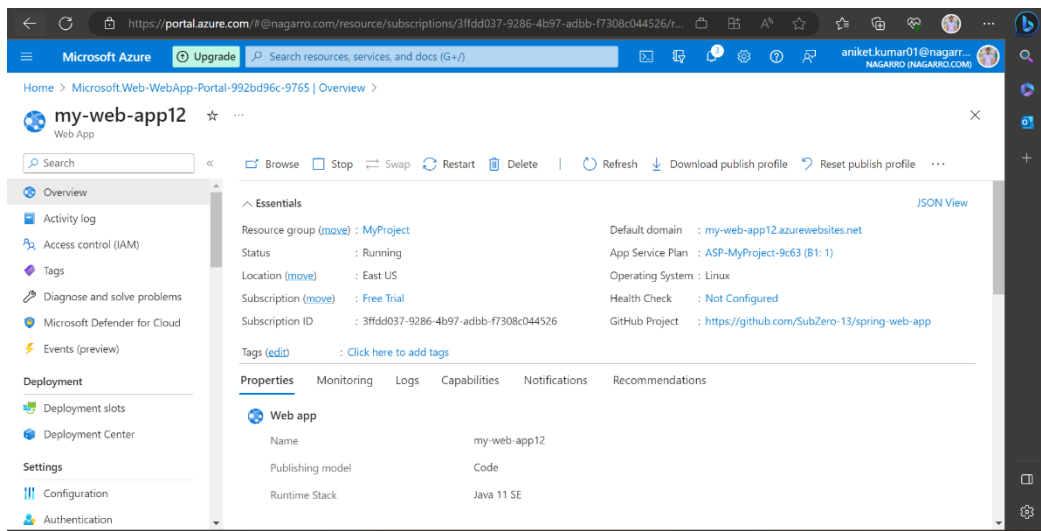
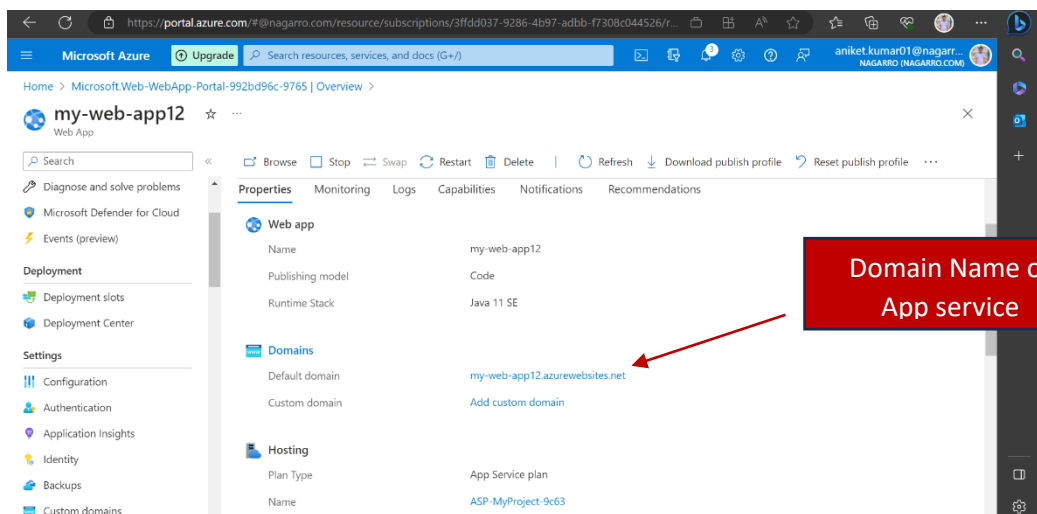
This is the user dashboard page.



App service screenshots:



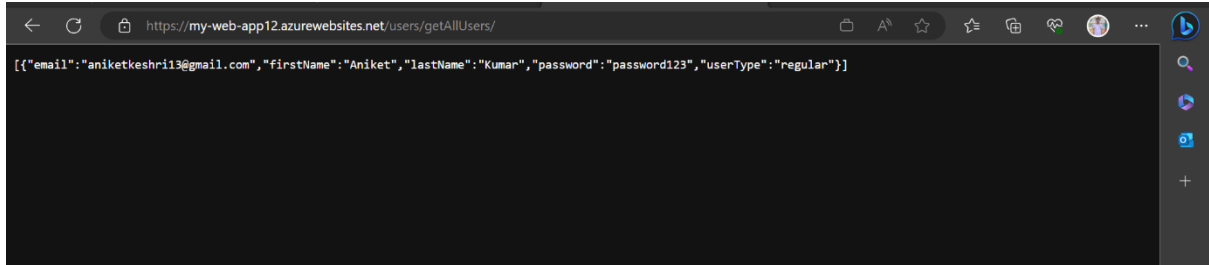
The screenshot shows the deployed backend on App service



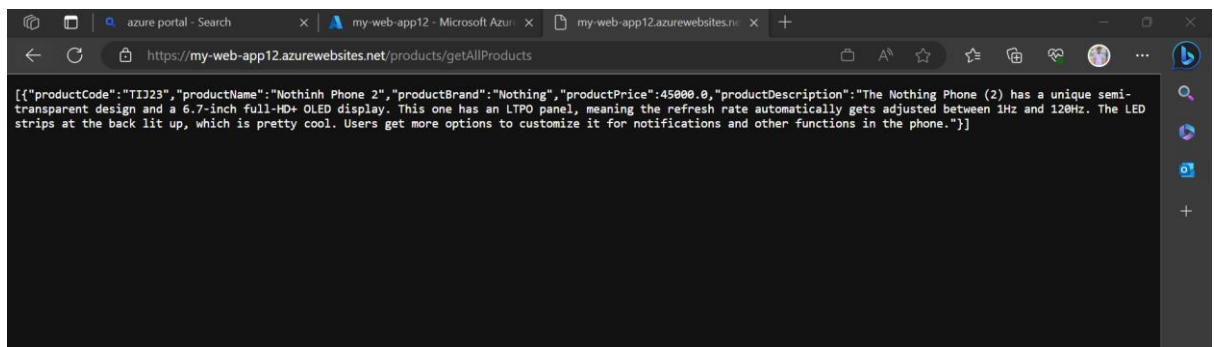
App Running on App service:

Domain name of App service

Backend:

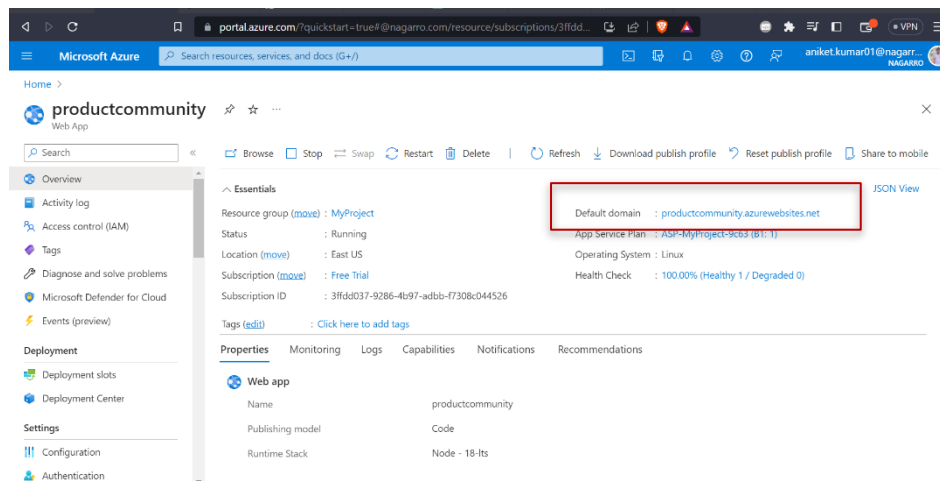


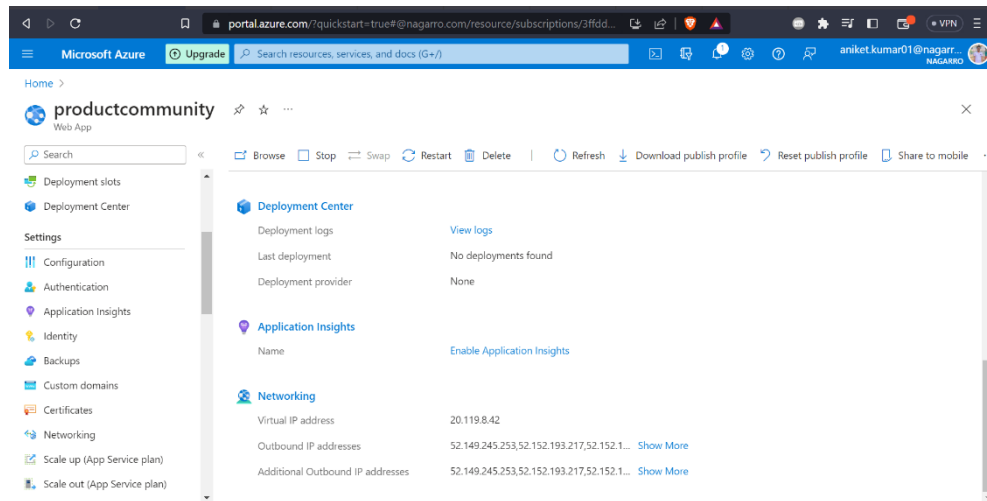
```
[[{"email": "aniketkeshri13@gmail.com", "firstName": "Aniket", "lastName": "Kumar", "password": "password123", "userType": "regular"}]]
```



```
[[{"productCode": "TI123", "productName": "Nothinh Phone 2", "productBrand": "Nothing", "productPrice": 45800.0, "productDescription": "The Nothing Phone (2) has a unique semi-transparent design and a 6.7-inch full-HD+ OLED display. This one has an LTPO panel, meaning the refresh rate automatically gets adjusted between 1Hz and 120Hz. The LED strips at the back lit up, which is pretty cool. Users get more options to customize it for notifications and other functions in the phone."}]]
```

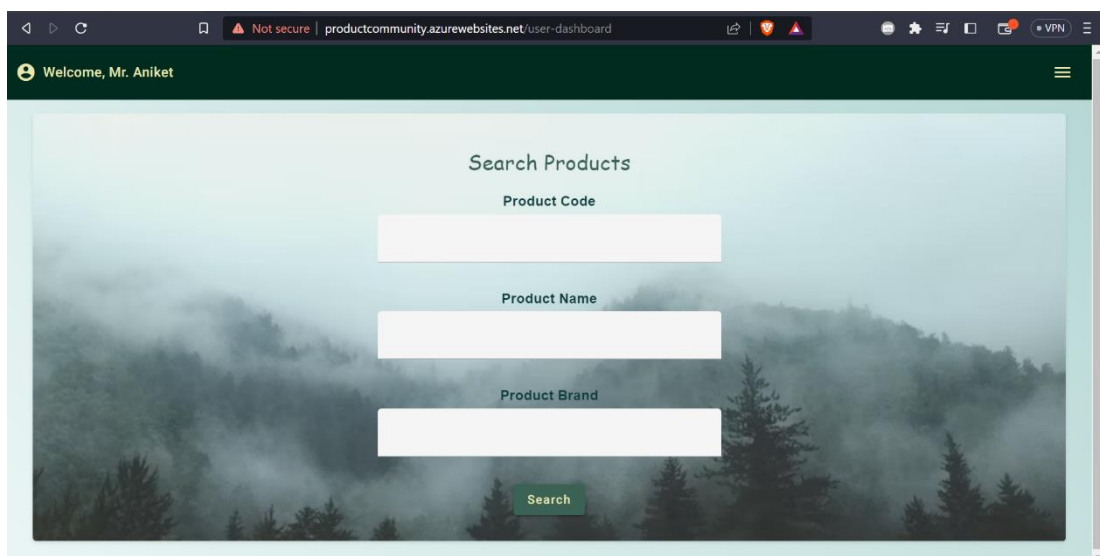
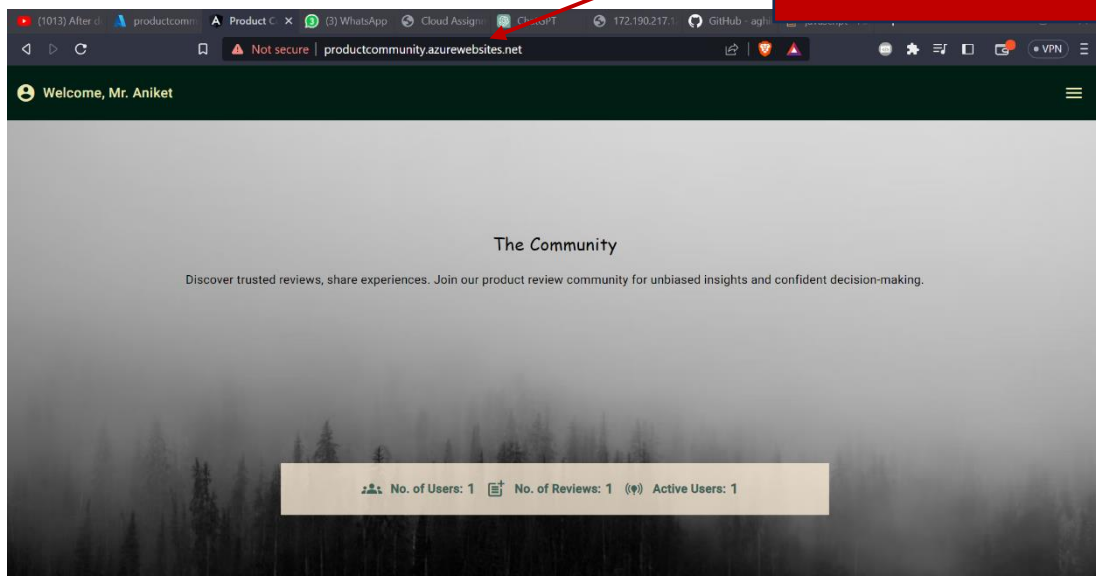
Frontend app on App service:

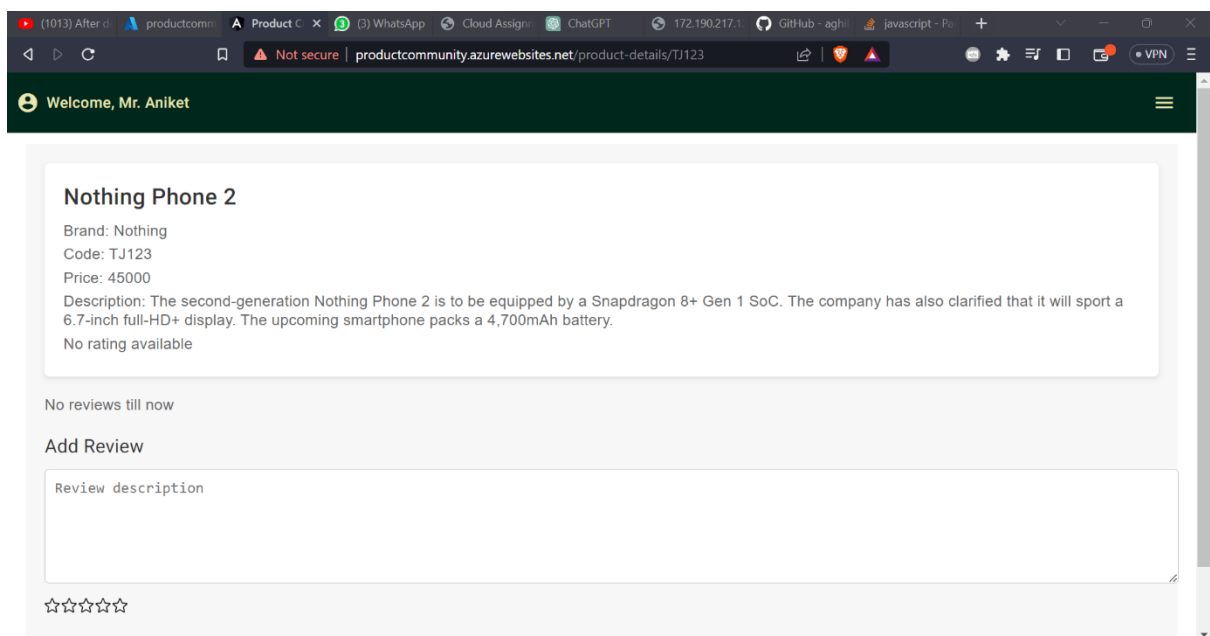
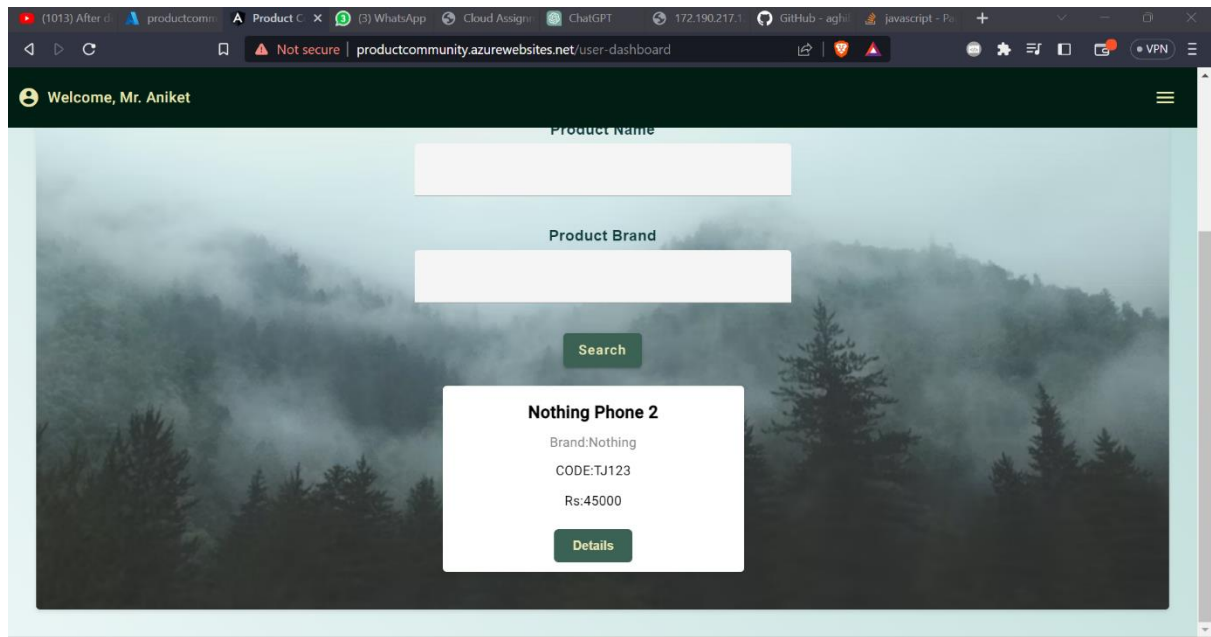




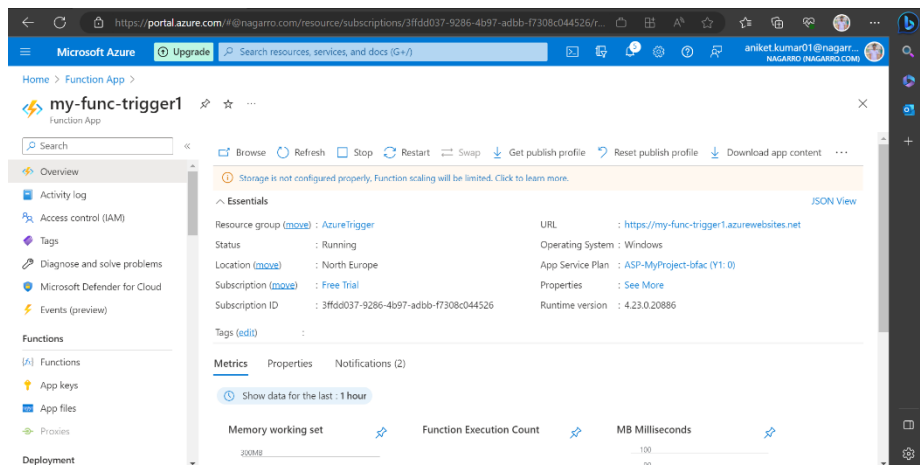
Frontend app running on App service:

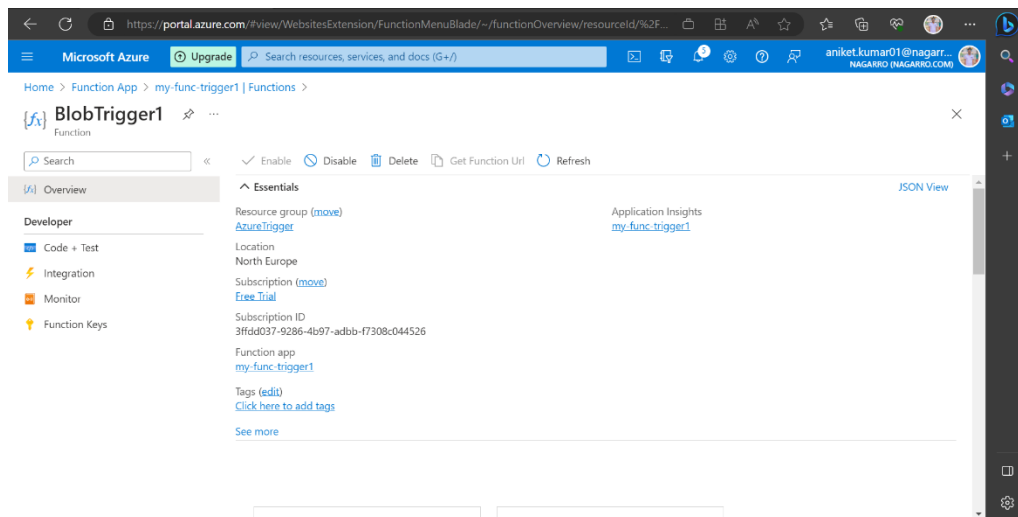
Domain name of frontend app service





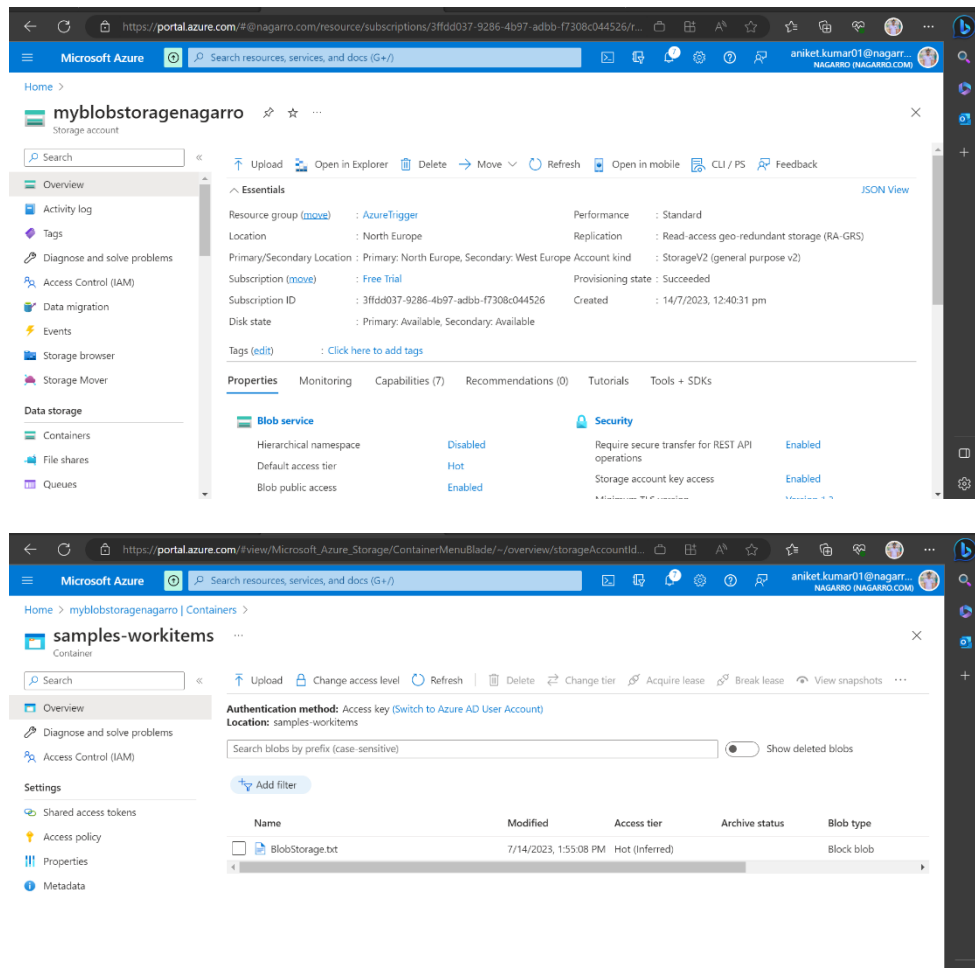
Azure Function:



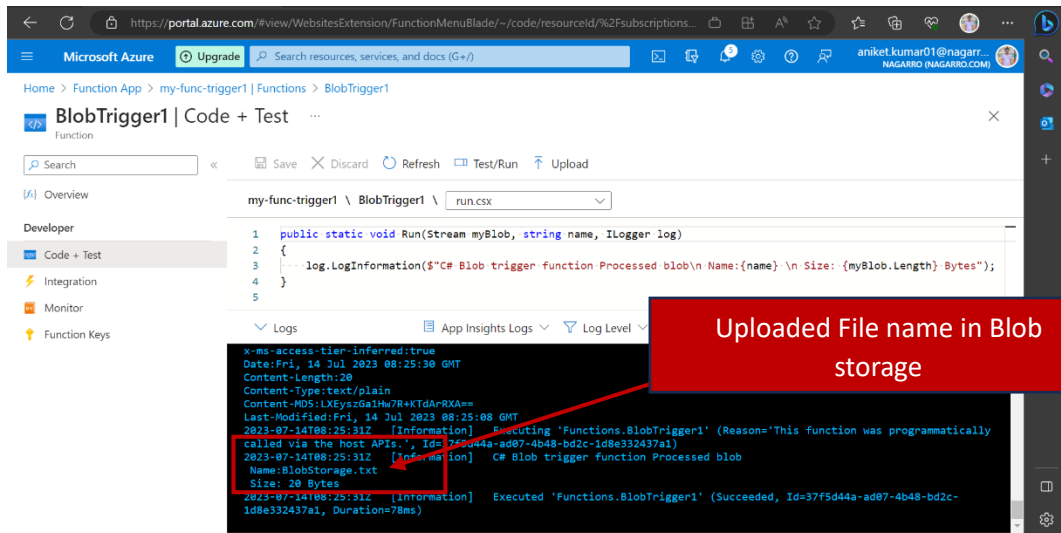


These screenshot shows the created azure function

Blob Storage:

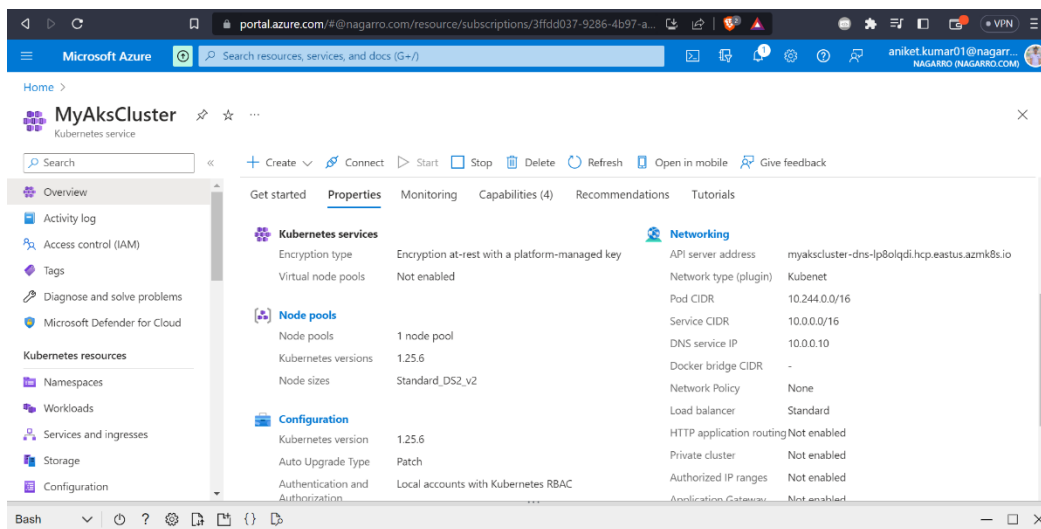


These screenshots show created Blob storage where I uploaded the file.

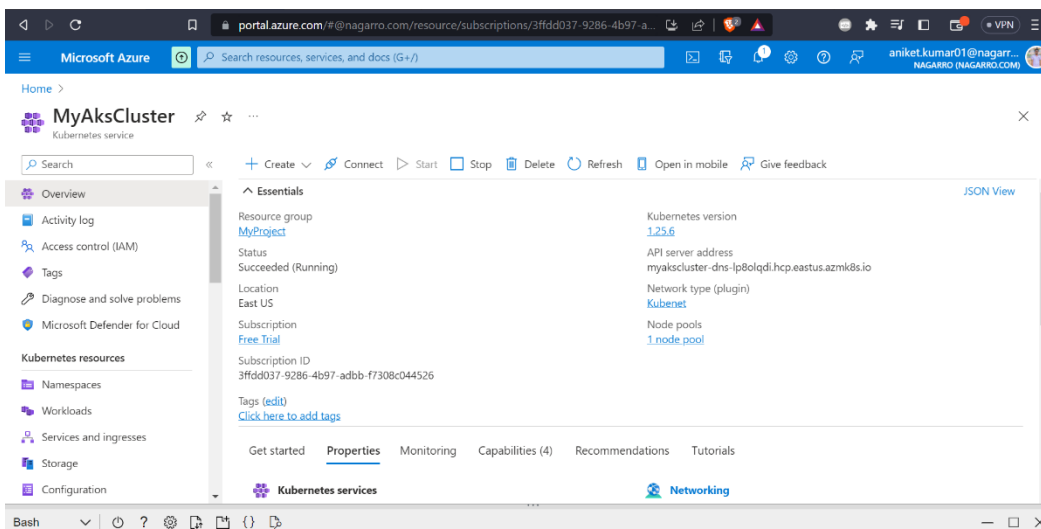


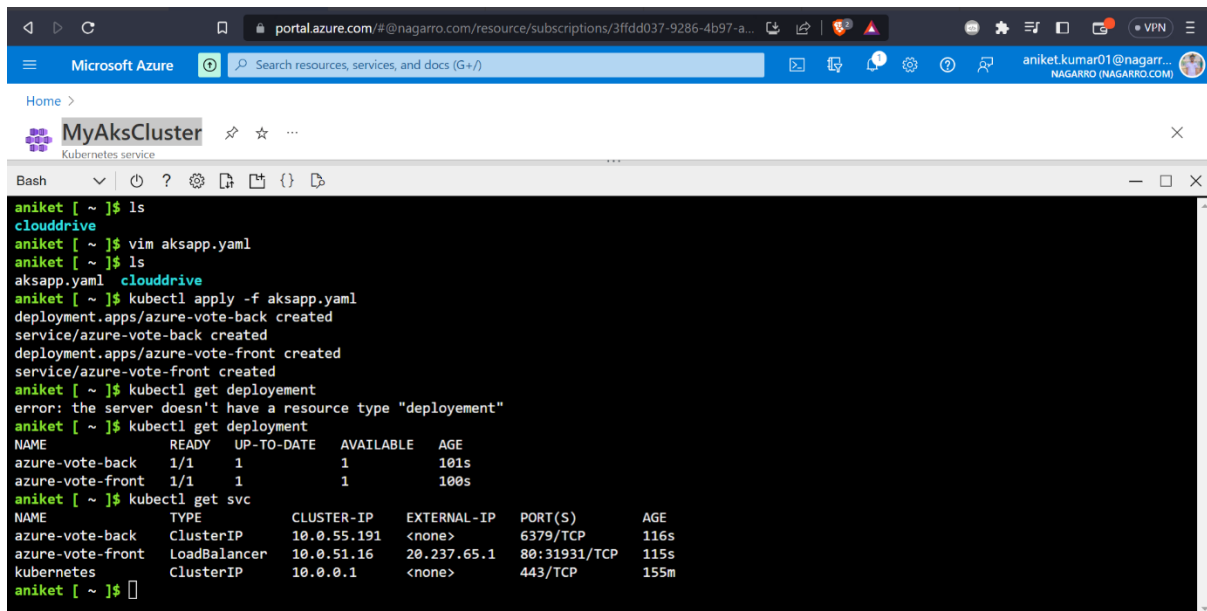
In this screenshot we can see the name of uploaded txt file

AKS Cluster:



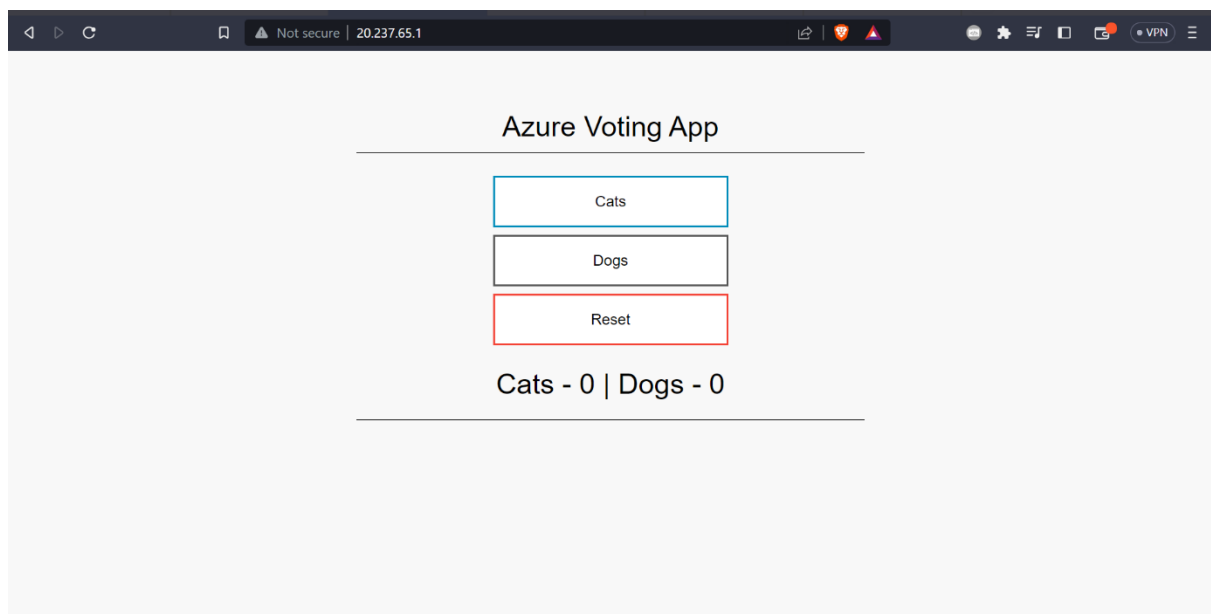
This screenshot shows the created aks cluster.





```
aniket [ ~ ]$ ls
clouddrive
aniket [ ~ ]$ vim aksapp.yaml
aniket [ ~ ]$ ls
aksapp.yaml clouddrive
aniket [ ~ ]$ kubectyl apply -f aksapp.yaml
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
aniket [ ~ ]$ kubectyl get deployment
error: the server doesn't have a resource type "deployment"
aniket [ ~ ]$ kubectyl get deployment
NAME          READY  UP-TO-DATE  AVAILABLE  AGE
azure-vote-back 1/1      1            1          101s
azure-vote-front 1/1      1            1          100s
aniket [ ~ ]$ kubectyl get svc
NAME          TYPE          CLUSTER-IP  EXTERNAL-IP  PORT(S)          AGE
azure-vote-back ClusterIP      10.0.55.191  <none>        6379/TCP         116s
azure-vote-front LoadBalancer  10.0.51.16   20.237.65.1  80:31931/TCP    115s
kubernetes    ClusterIP      10.0.0.1     <none>        443/TCP          155m
aniket [ ~ ]$
```

This screenshot shows the process of How We can Deploy our services.



This screenshot shows the deployed Service on AKS Cluster

