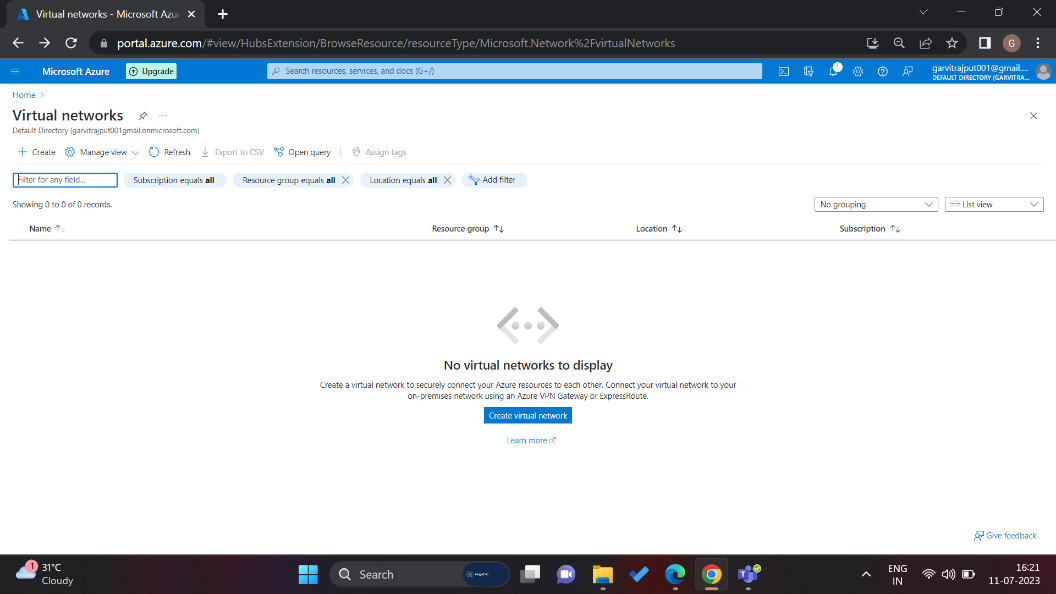
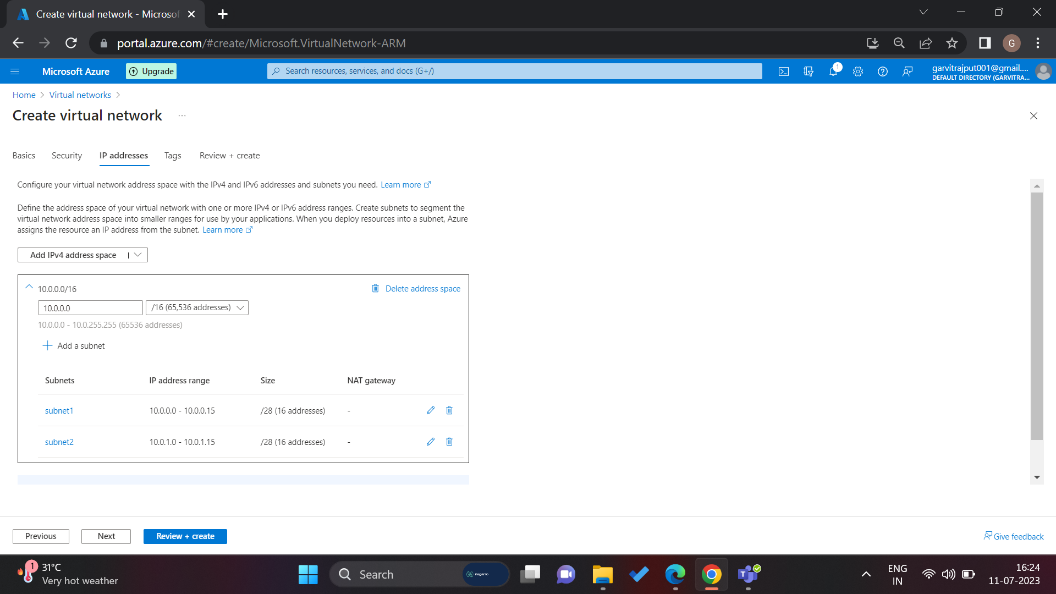
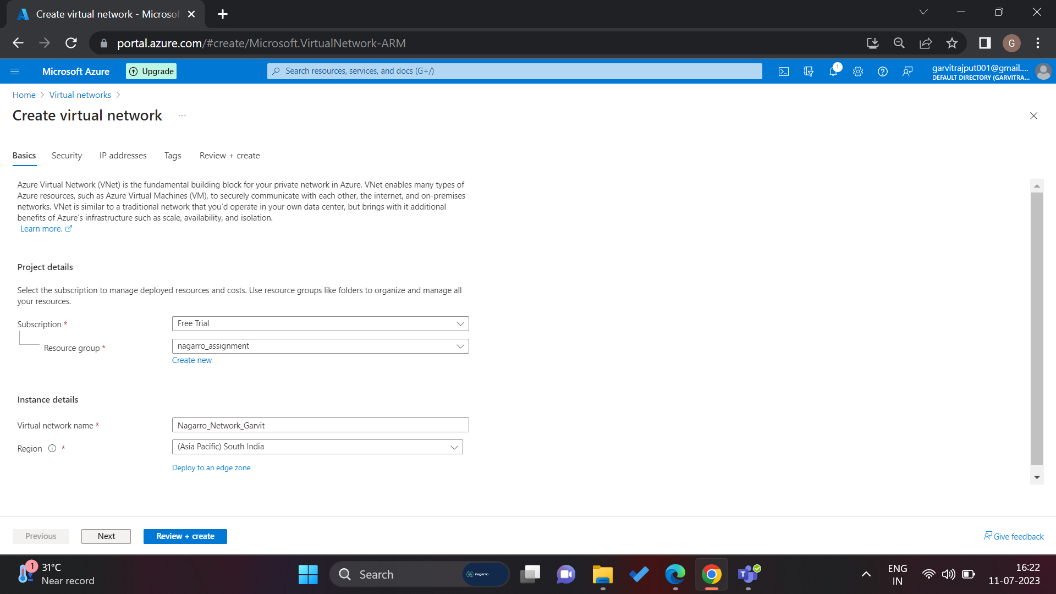
**Step 1:** Create Virtual Network



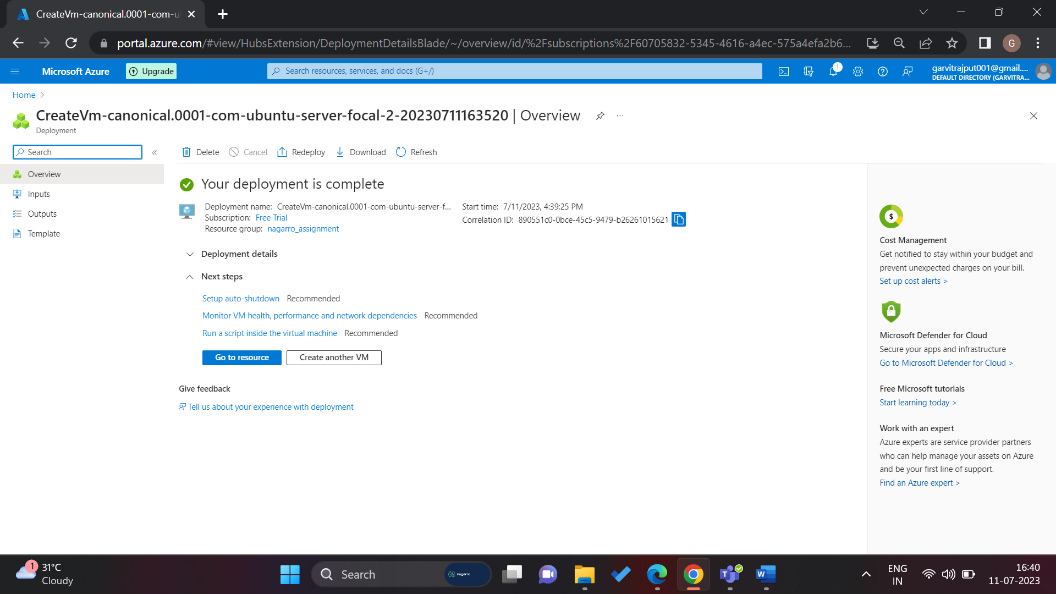
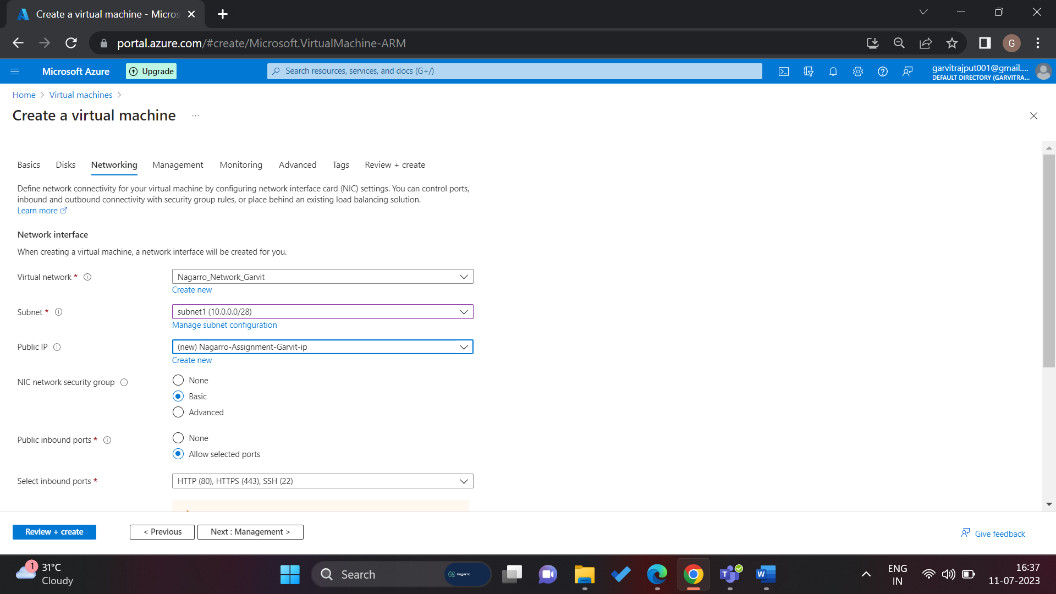
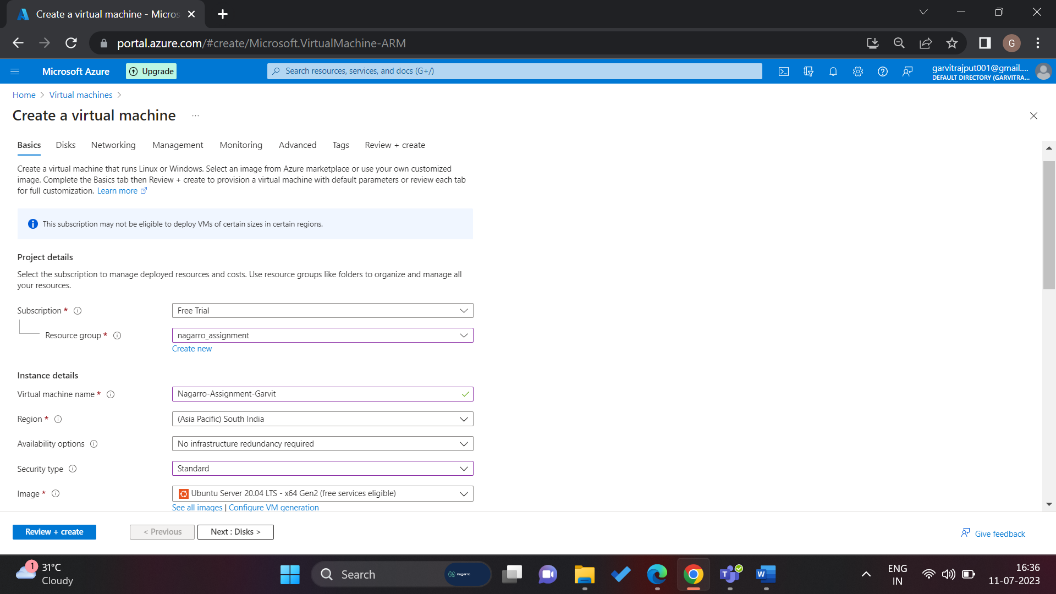


A Virtual Network is created here *“Nagarro\_Network\_Garvit”.*

It contains two subnets having 16 IP addresses in each.

*{subnet1, subnet2}*

**Step 2:** Creating a virtual machine in one of the subnets created above.



Here a ubuntu virtual machine inside one of the subnets is being created *“Nagarro\_Assignment\_Garvit”*

Deployment is complete.

Virtual Machine is ready to use now.

**Step 3:** Connect this VM through terminal using the “.pem” key downloaded and the public IP address using the command:

ssh -i <path-to-pem-key> azureuser@ip-address

Here, I’m connecting to my virtual machine using the given command.

“**ssh -i nagarro\_assignment.pem azureuser@20.219.101.34**”

Connected to the VM.

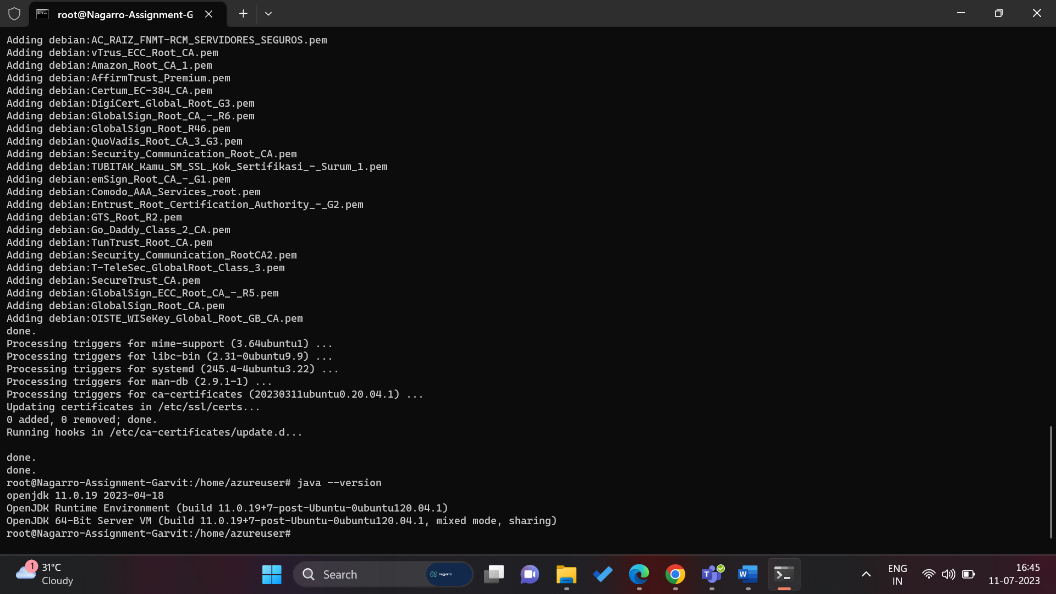
Now we will install required packages to run spring boot application i.e. compatible JDK.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



I have installed compatible JDK Version.

**Step 4:** Create MySQL Database Server in Azure and copy the connection strings.

MySQL Database Server is created:

“nagarrogarvit”

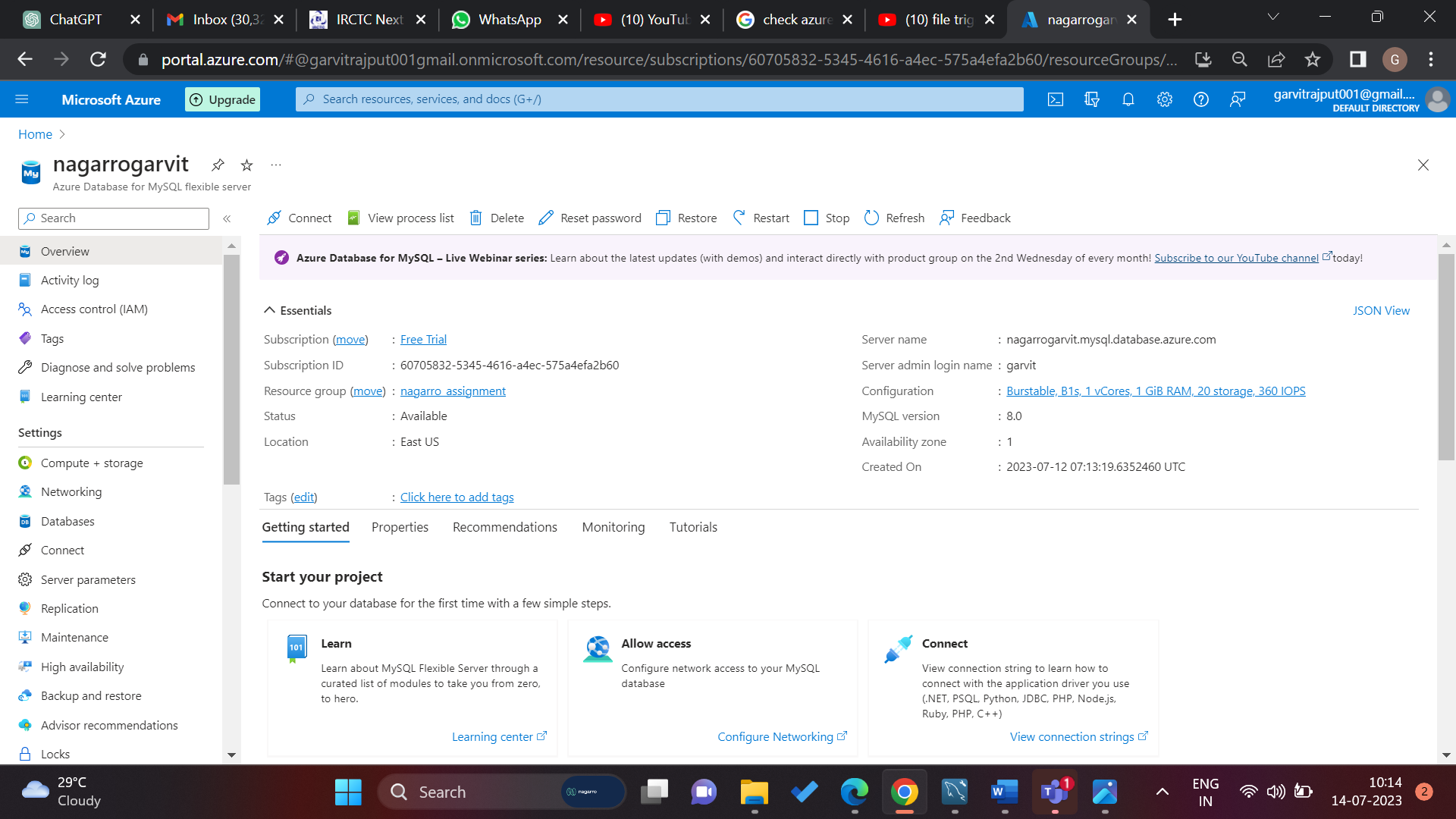
Now to connect to this server using MySQL Workbench we need 1. Hostname

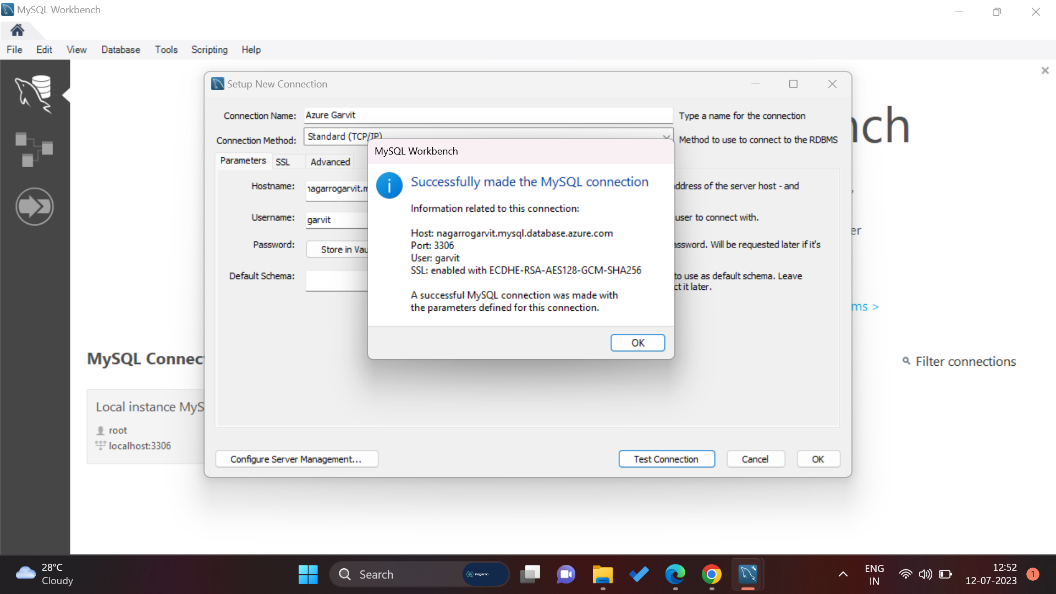
2. Username

3. Password

After providing correct credentials.

We have connected to the cloud MySQL server successfully.



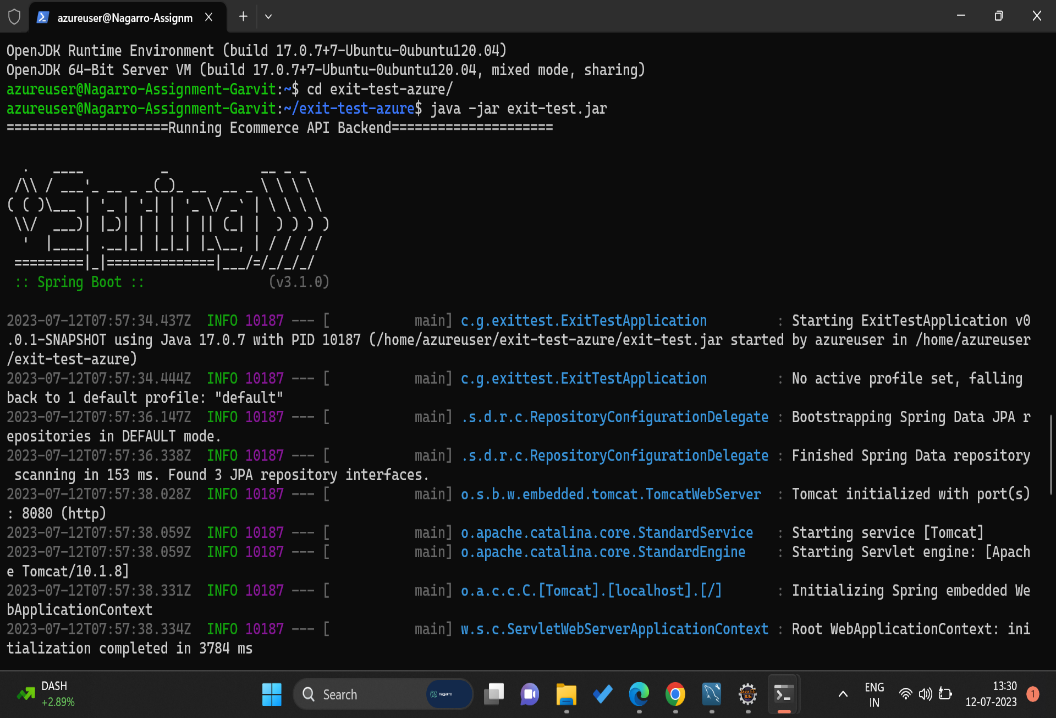


**Step 5:** Now we must specify the database server details in our spring boot application which we want to deploy on the VM and create and executable jar file, now this jar file should be uploaded to the VM either through git clone or any FTP server.

I used GitHub for this.

After cloning it into the VM, cd to the location of this jar file and hit the command:

java -jar {jarFileName.jar}



Running the command

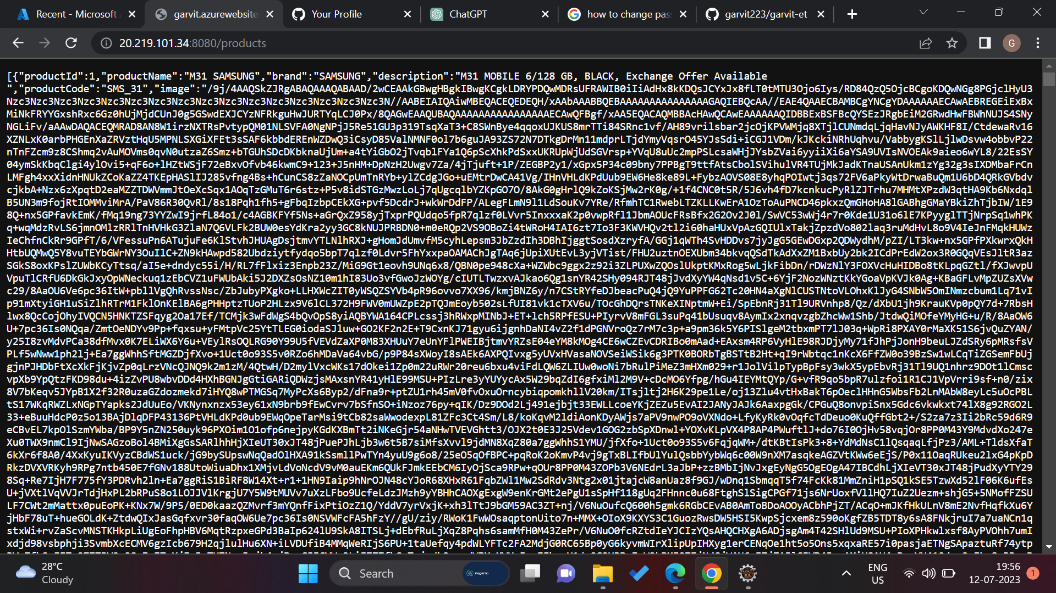
“java -jar exit-test.jar”

It is running successfully on the defined port [default port: 8080] on embedded tomcat server.

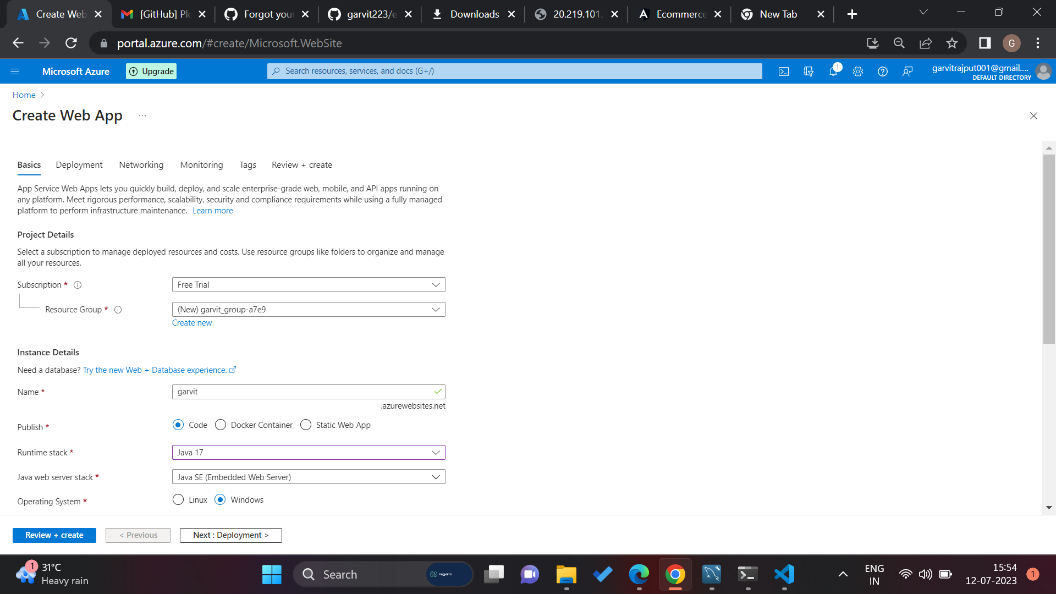
API Running on

**20.219.101.34:8080**

VM is working fine.



**Step 6:** Now we have to deploy the same application on Azure App Service.



Creating Azure App Service

Name: garvit.azurewebsites.net

Java 17

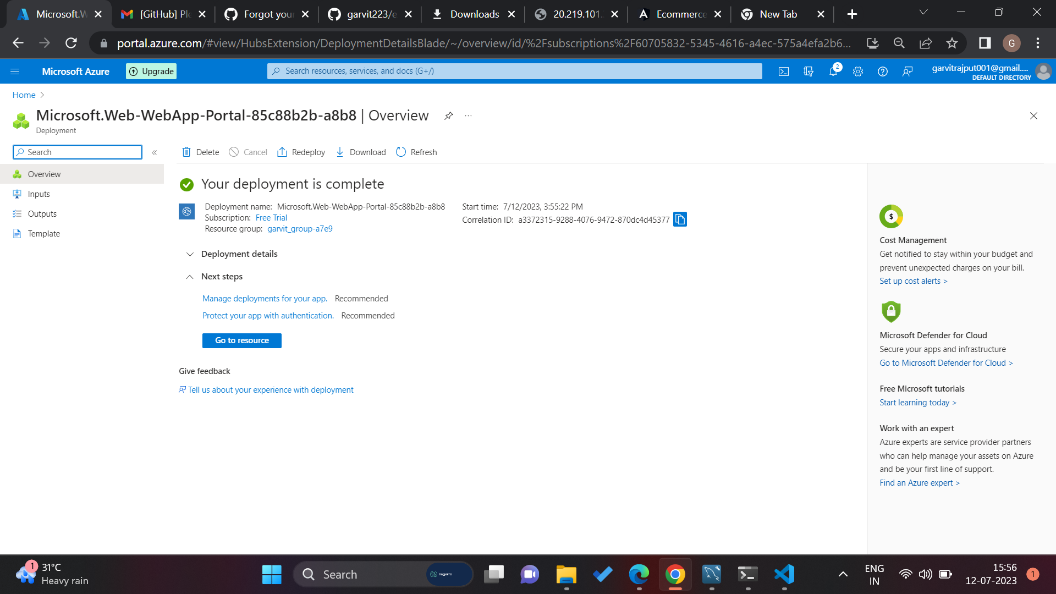
Web App Created Successfully.

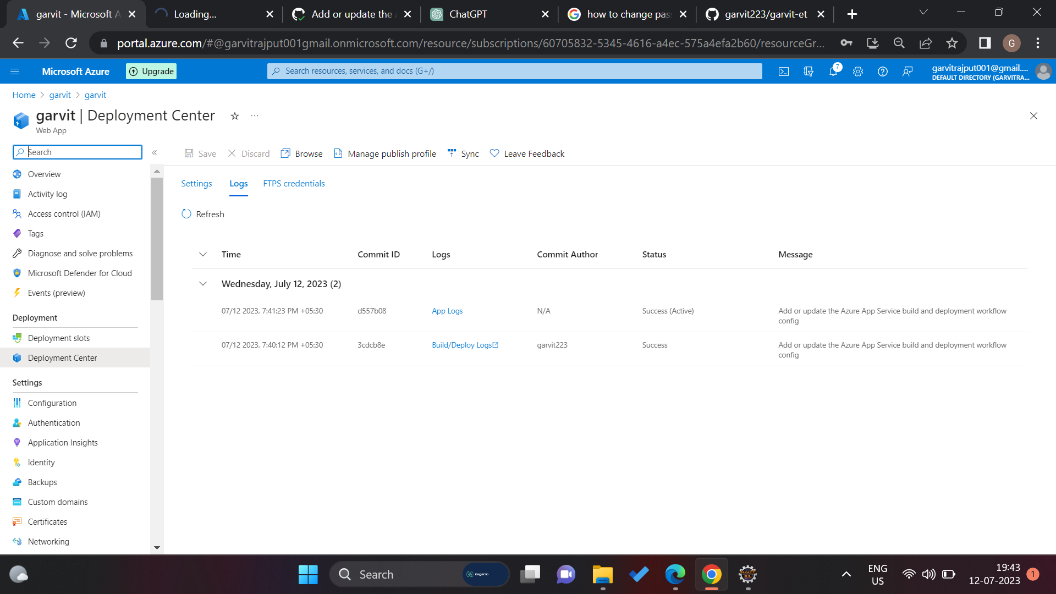
Now we have to connect it to GitHub Repository.

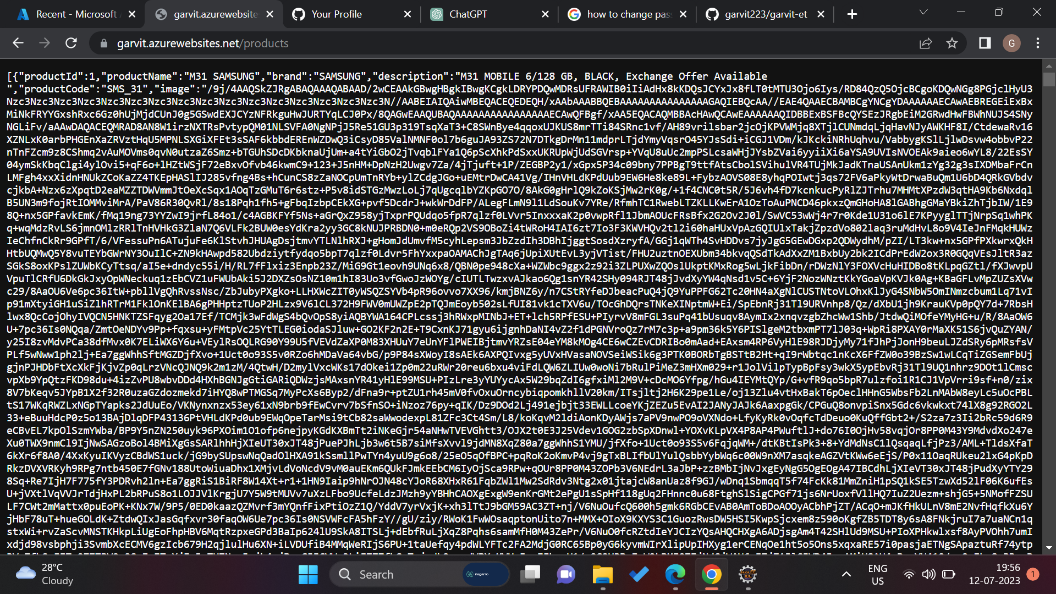
Connected to GitHub Repository, it shows **Success Status,** means it has successfully build the project and it is ready to run

Project is running successfully.

It is returning the API Data for products.

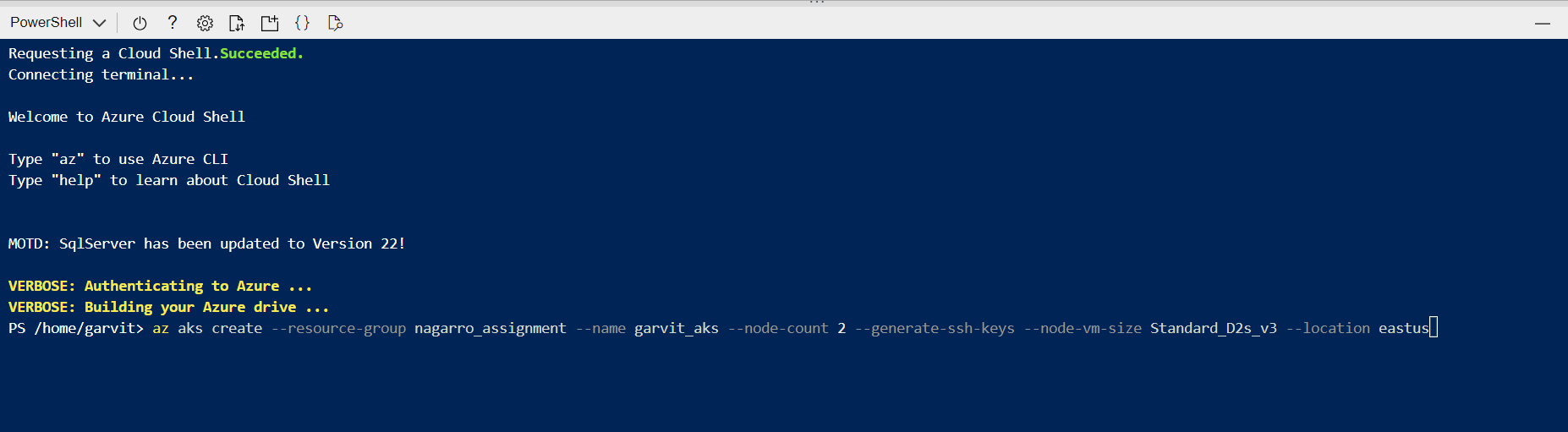






**Step 7:** Creating Azure Kubernetes Cluster using the command:

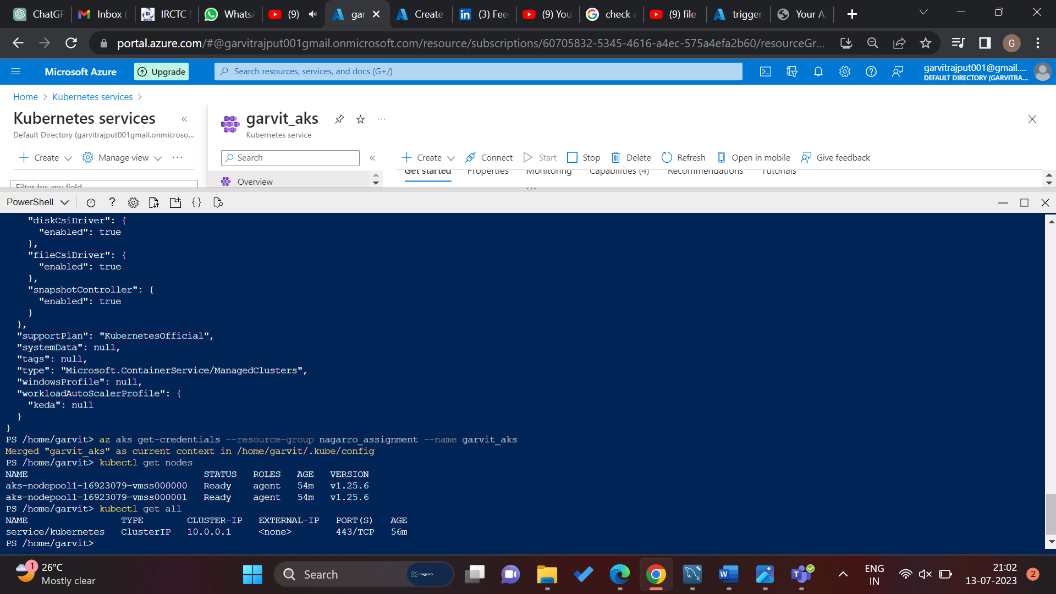
PS /home/garvit> az aks create --resource-group nagarro\_assignment --name garvit\_aks --node-count 2 --generate-ssh-keys --node-vm-size Standard\_D2s\_v3 --location eastus

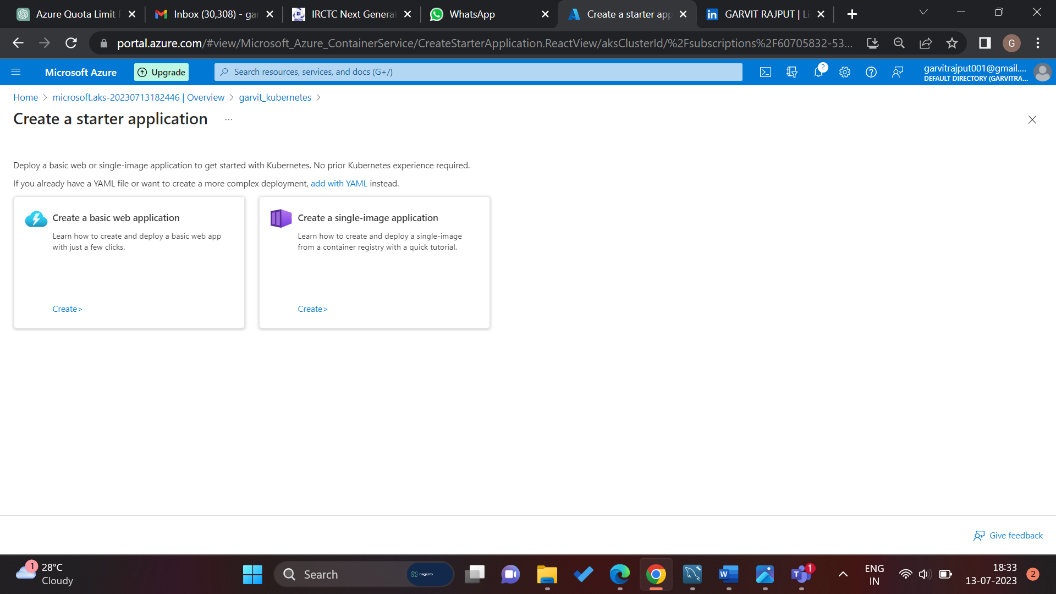


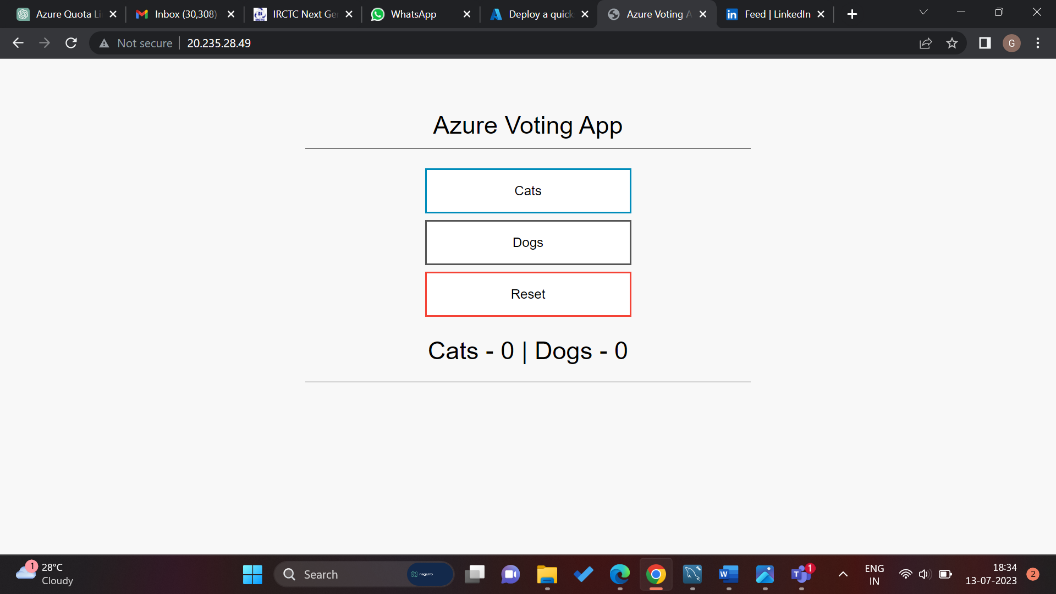
Creating an AKS Cluster using the given command.

Now, we will deploy the service to this aks cluster.

Here we will create a basic web application using AKS Cluster



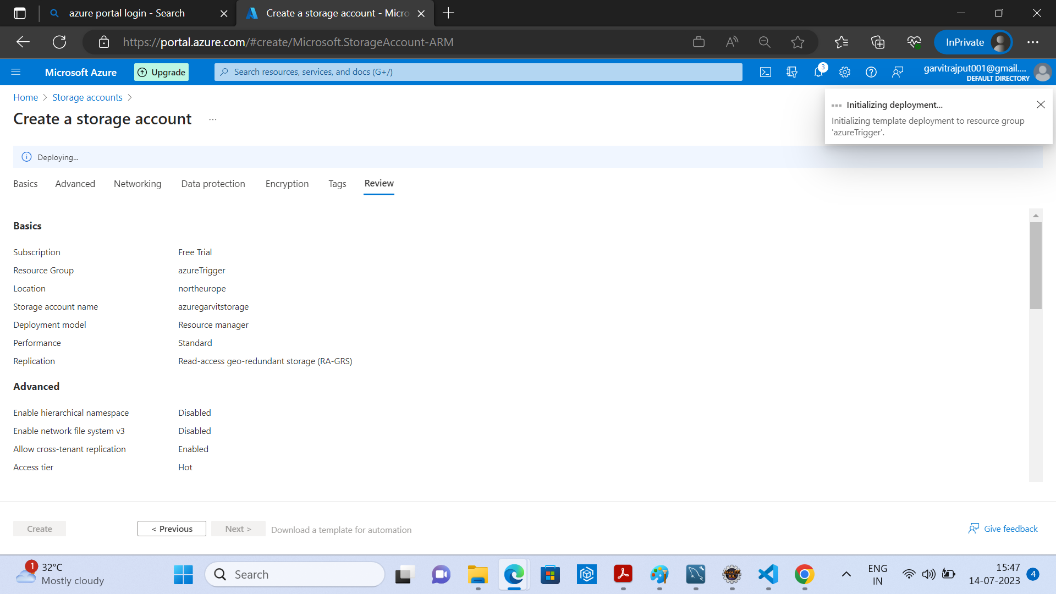




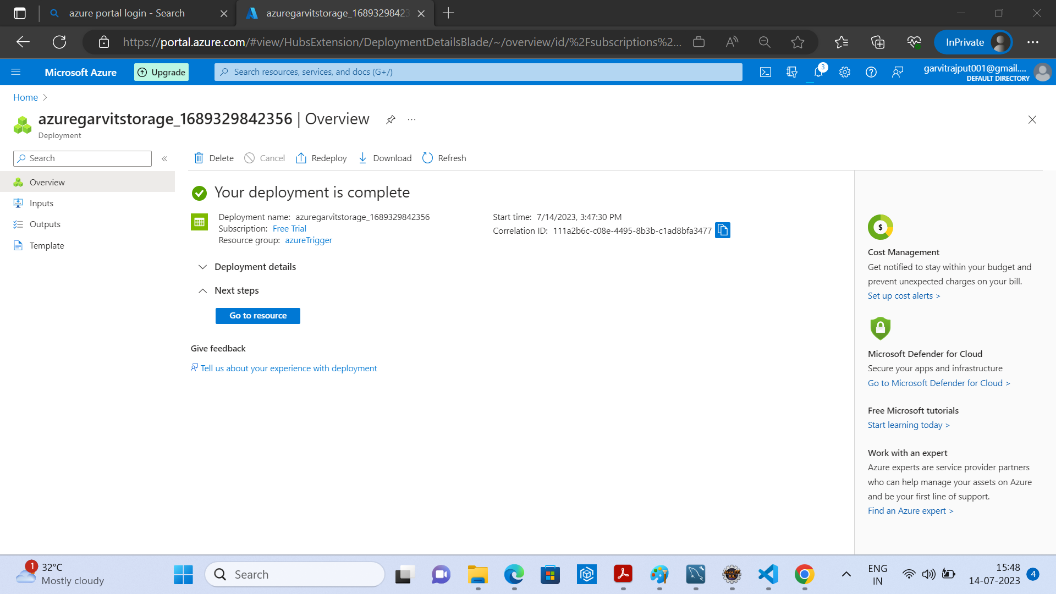
This demo **Azure Voting App** is created, and it is accessible through internet.

*\*Due to the limitations of nodes, I have only one node has been created and deployed only one service there.*

**Step 8:** Creating a function through Azure Function App, which gets triggered as soon as we upload a file to BLOB storage service.



Create a storage account, where we will upload the files to test the function.



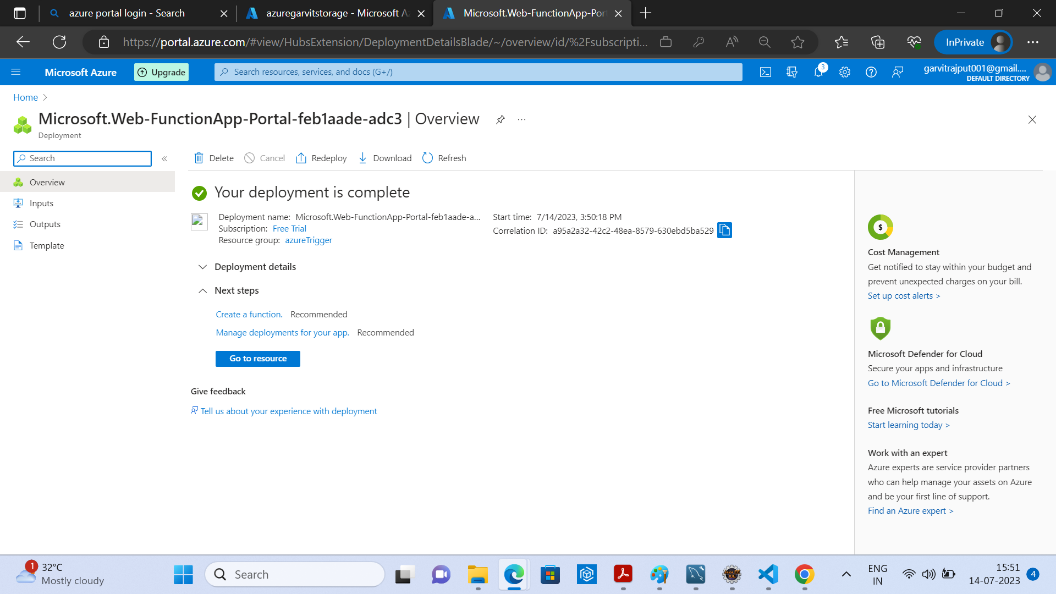
Storage account created, now proceed to creation of function.

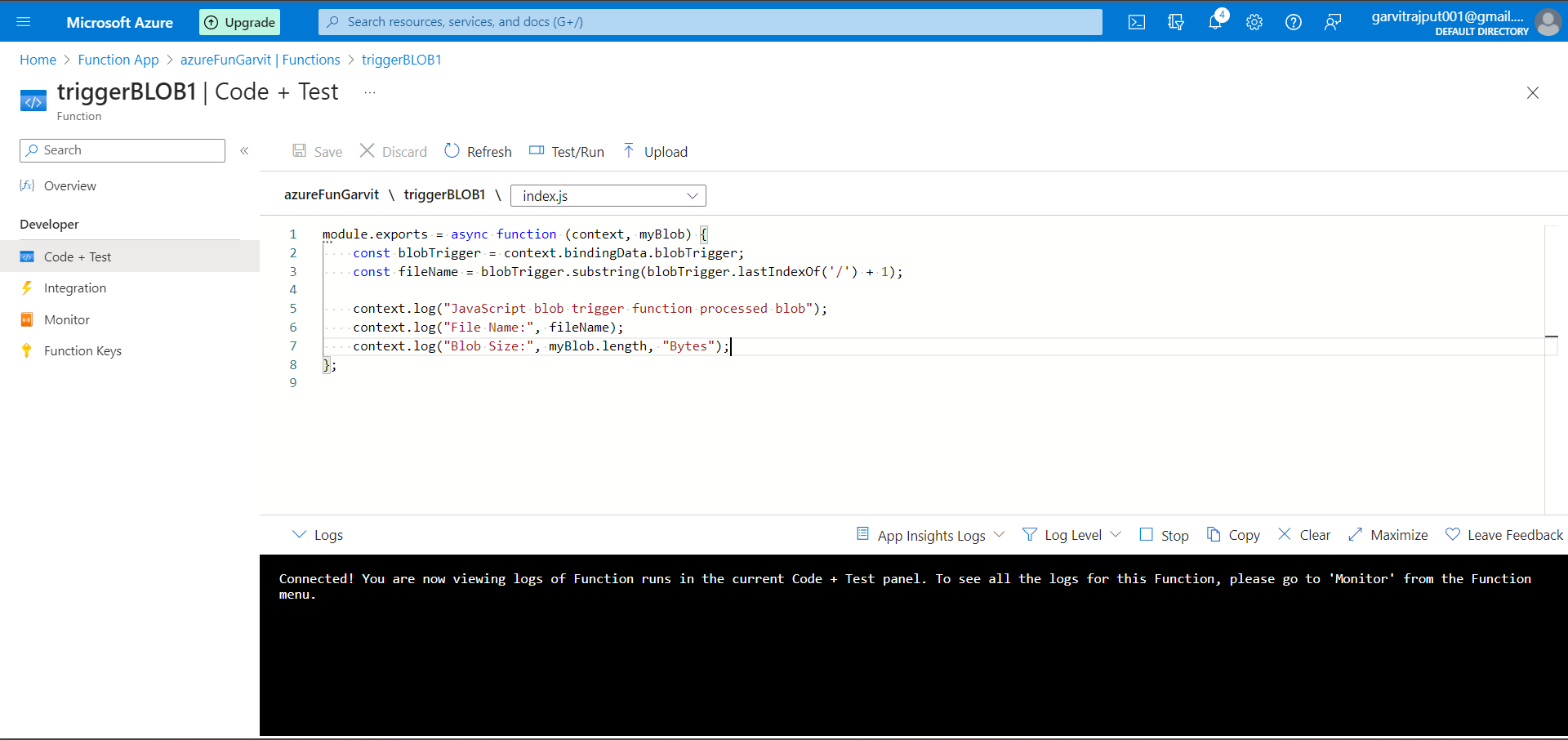
Creating function app here with Node.js configuration in North Europe Region

Function app is created and is now we will create a function that triggers.

A screenshot of a computer

Description automatically generated





I have created a function using Node.js, which prints the file name on the console.

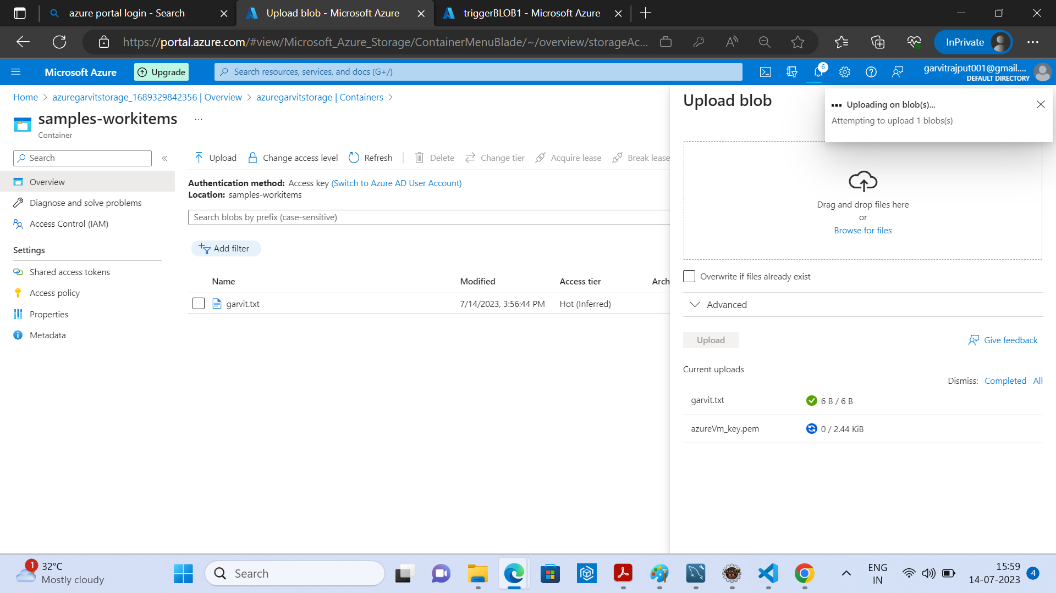
Save this code.

Here I have uploaded two files,

1. Garvit.txt

2. azureVm\_key.pem

The function gets triggered and prints the filename which were uploaded.





**Component Description:**

**Resource group:** A logical container in Azure used for organizing and managing related Azure resources.

**Virtual Network:** A network topology in Azure that provides isolated communication between resources.

**Subnet:** A subdivision of a virtual network that allows further segmentation and organization of resources.

**VM (Virtual Machine):** A virtualized computing instance in Azure that runs an operating system and applications.

**Azure App Service:** A platform in Azure for hosting and managing web applications, supporting multiple programming languages and frameworks.

**AKS (Azure Kubernetes Service):** A managed container orchestration service in Azure that simplifies deployment and management of containerized applications using Kubernetes.

**Storage Account:** A storage service in Azure for storing data objects such as files, disks, and backups.

**BLOB (Binary Large Object):** A type of object in Azure Storage Account used for storing unstructured data like images, videos, and documents.

**Azure Function:** A serverless compute service in Azure that allows the execution of small code snippets, or functions, in response to events or triggers.