

## AZURE ASSIGNMENT

1. Create a virtual network with 2 subnets. Each subnet should have 16 Ips only.

Created Virtual Network and 2 Subnets under resource group MyRG in Free-Trial subscription

The screenshot shows the Microsoft Azure portal interface. The main content area displays the 'MyVnet | Subnets' page. The page includes a search bar, a list of subnets, and a table of subnet details. The table has columns for Name, IPv4, IPv6, Available IPs, and Delegation. Two subnets are listed: 'subnet1' and 'subnet2'. 'subnet1' has an IPv4 address of 10.0.1.0/28 and 10 available IPs. 'subnet2' has an IPv4 address of 10.0.2.0/28 and 11 available IPs. The left sidebar shows the 'Virtual networks' section with 'MyVnet' selected. The top navigation bar shows the user is logged in as 'akansha.khandelwal@nagarro.com'.

Name	IPv4	IPv6	Available IPs	Delegation
subnet1	10.0.1.0/28	-	10	-
subnet2	10.0.2.0/28	-	11	-

2. Inside one of the subnets, create a VM and deploy an application code inside it and it should leverage the database on the cloud (any existing application created by you before)

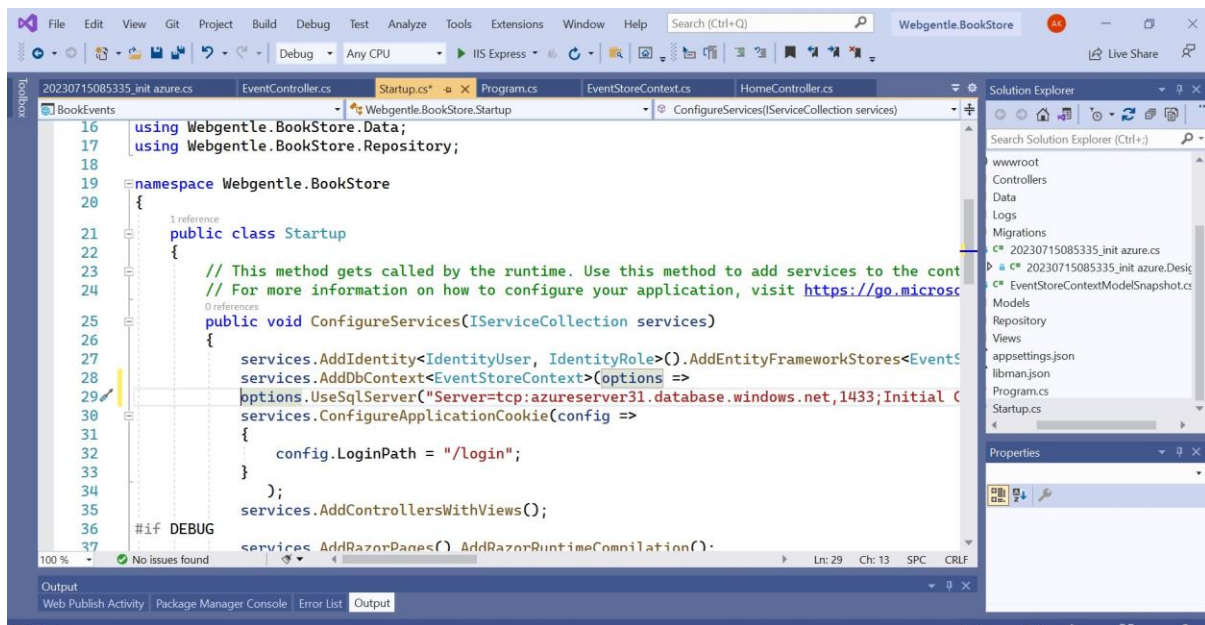
### Step 1 -Create VM

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, an 'Upgrade' button, and a search bar. The main content area is titled 'Virtual machines' and shows a list of VMs with 'MyVM' selected. The 'MyVM' details page is displayed, showing the 'Overview' tab. The 'Essentials' section lists key information: Resource group (MyRG), Operating system (Windows (Windows Server 2022 Datacenter...)), Status (Running), Location (East US), Subscription (Free Trial), Subscription ID (836a1537-3388-4d3b-aad2-ff2d22d199f1), Virtual network/subnet (MyVnet/subnet1), DNS name (Not configured), and Health state (-). The 'Properties' tab is also visible at the bottom.

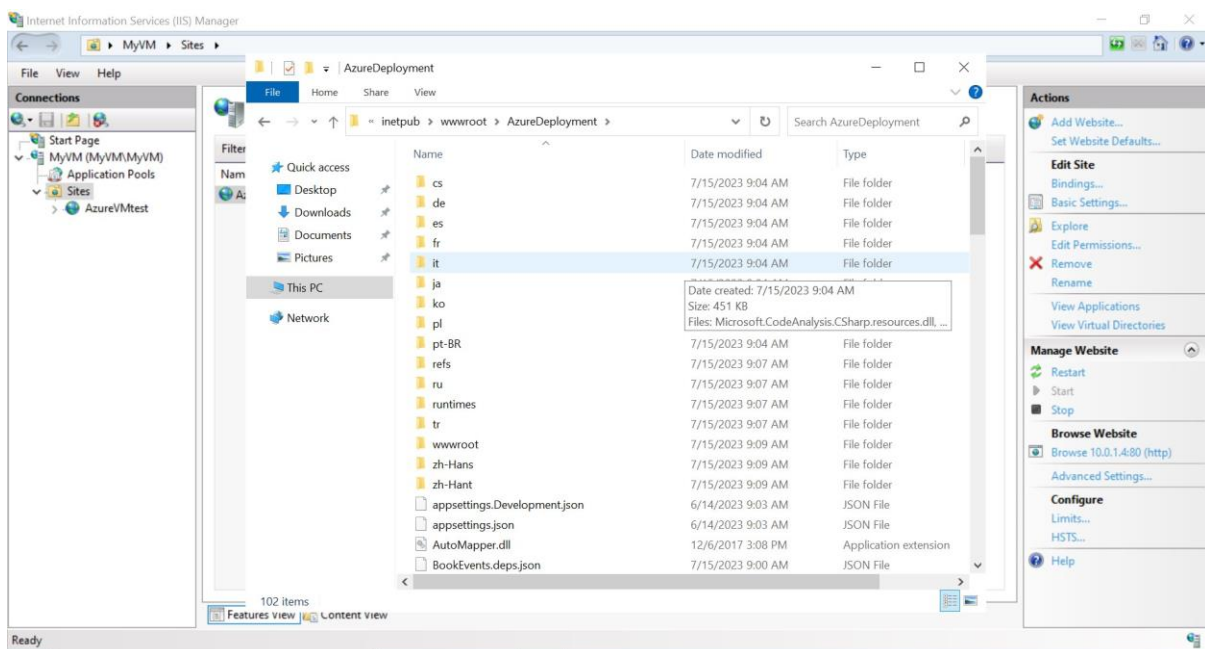
### Step-2 Setting up SQL Database

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo, an 'Upgrade' button, and a search bar. The main content area is titled 'SQL databases' and shows a list of databases with 'AzureDb (azureserver31/AzureDb)' selected. The 'AzureDb (azureserver31/AzureDb)' details page is displayed, showing the 'Overview' tab. The 'Essentials' section lists key information: Resource group (MyRG), Server name (azureserver31.database.windows.net), Status (Online), Location (East US), Subscription (Free Trial), Subscription ID (836a1537-3388-4d3b-aad2-ff2d22d199f1), Elastic pool (No elastic pool), Connection strings (Show database connection strings), Pricing tier (Basic), and Earliest restore point (2023-07-15 09:25 UTC). The 'Getting started' tab is also visible at the bottom.

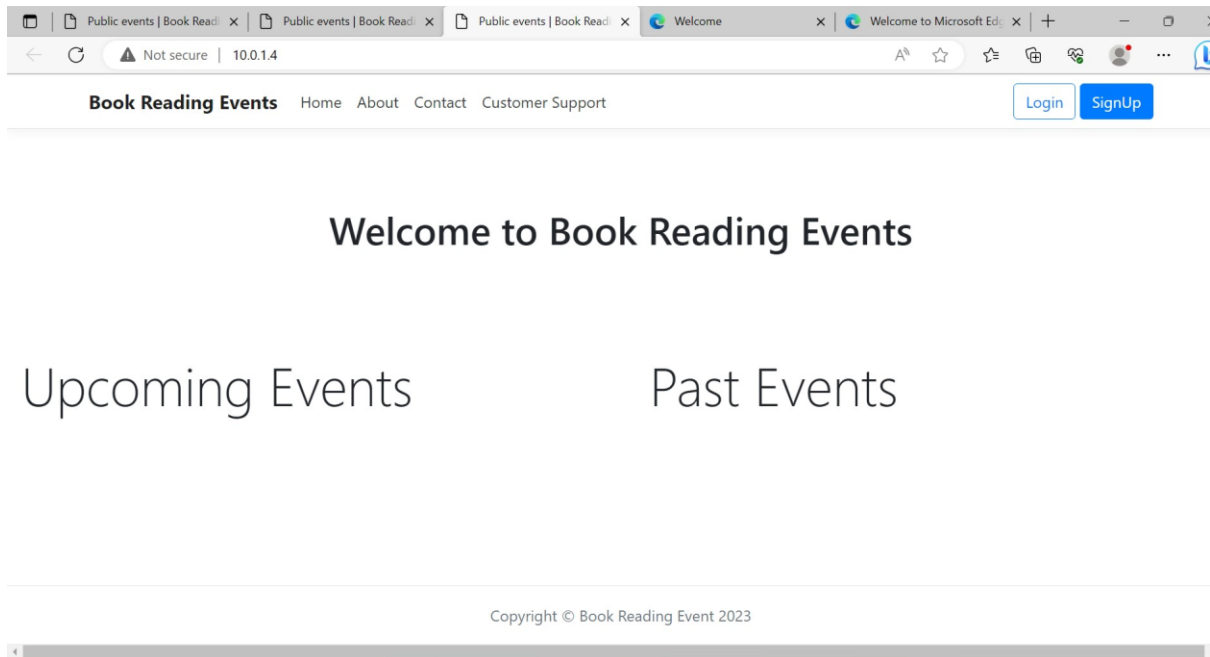
### Step-3 Setting up Connection String for Azure SQL database



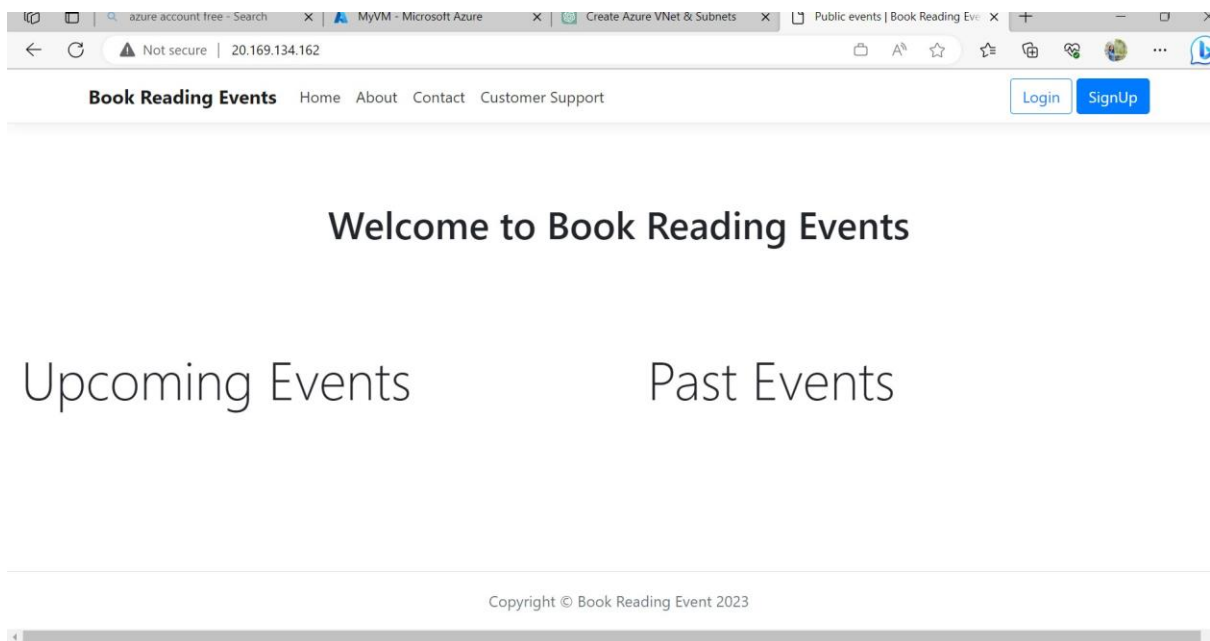
### Step- 4 Setting up Environment on VM for Deployment



## Step-5 Code Deployed on Virtual Machine

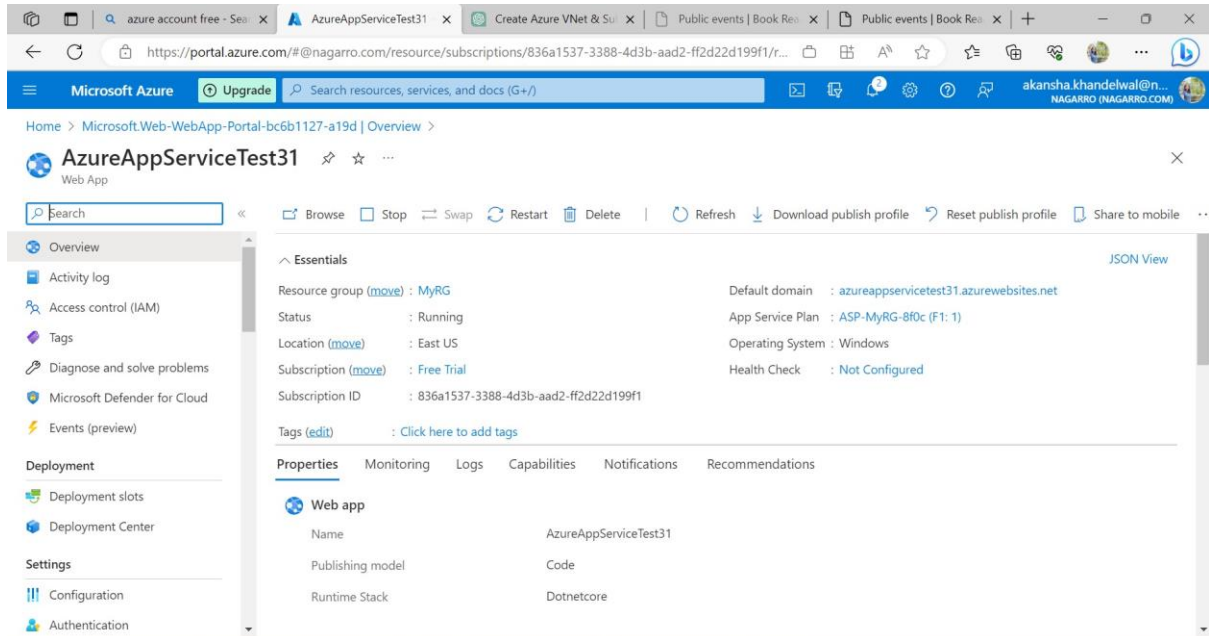


## Step- 6 VM Deployed Code Testing on Local machine

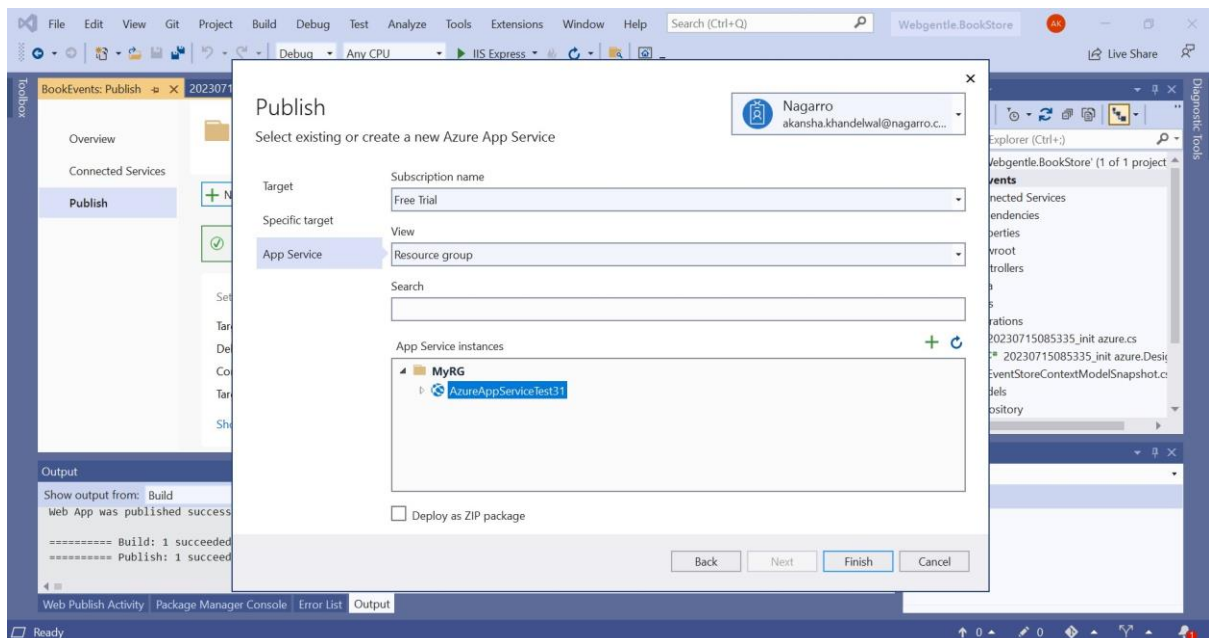


3. Deploy the same application to Azure App Service. It should also leverage the database on the cloud.

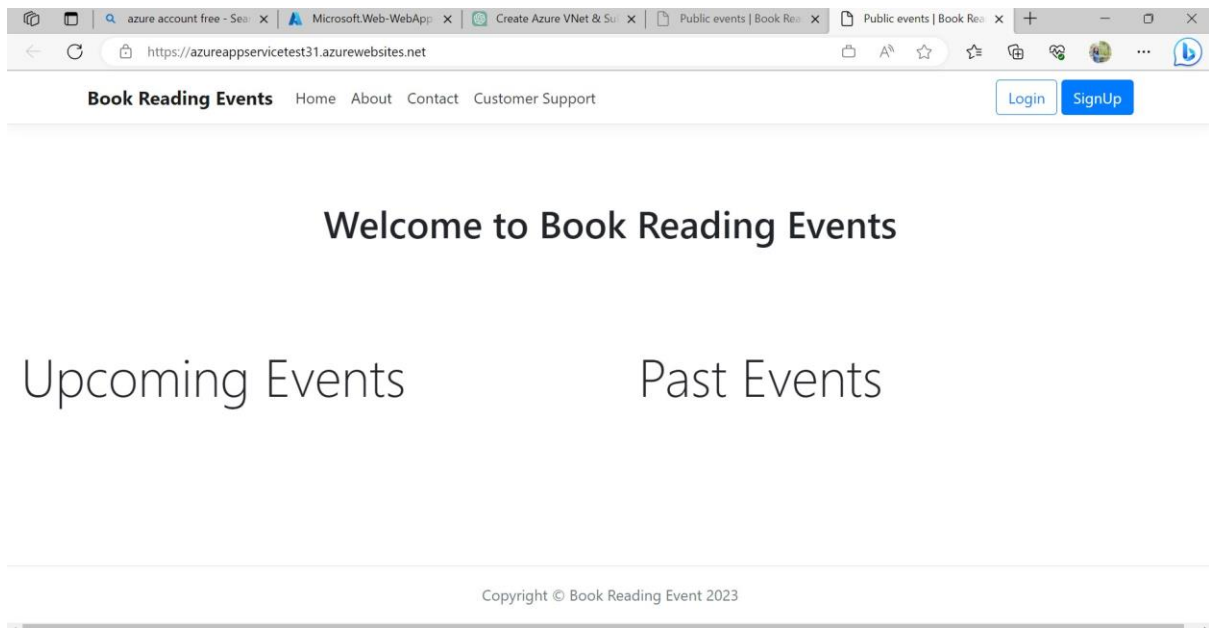
### Step -1 Creating App Service



### Step-2 Publishing the code to AppService

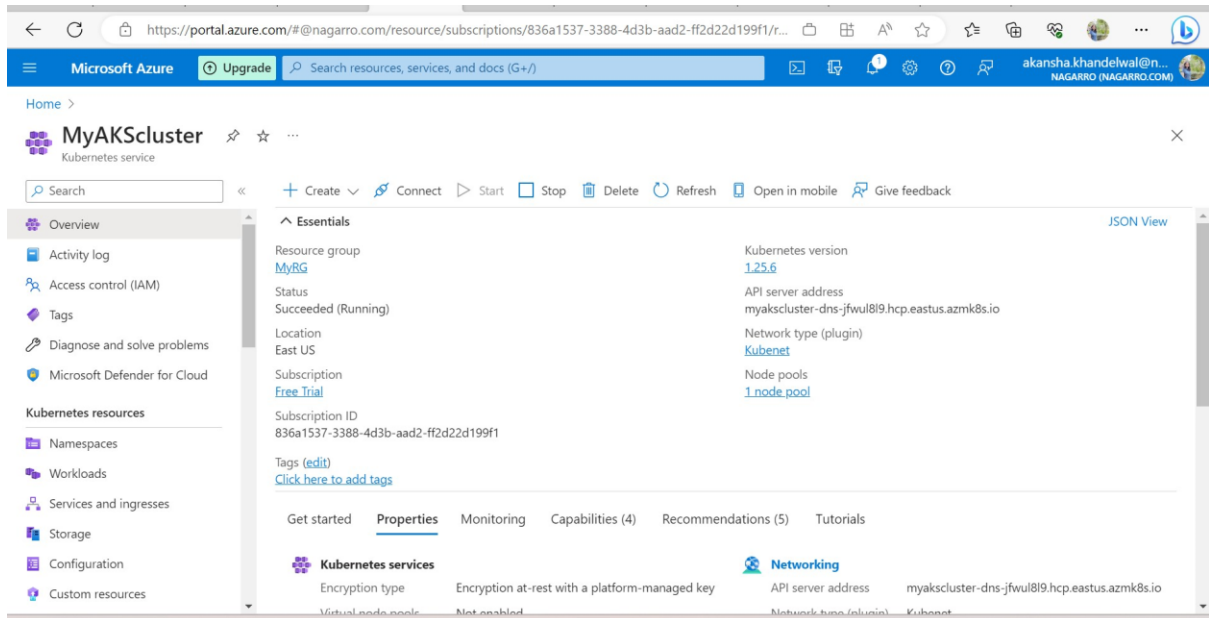


### Step-3 Code deployed on Azure App service

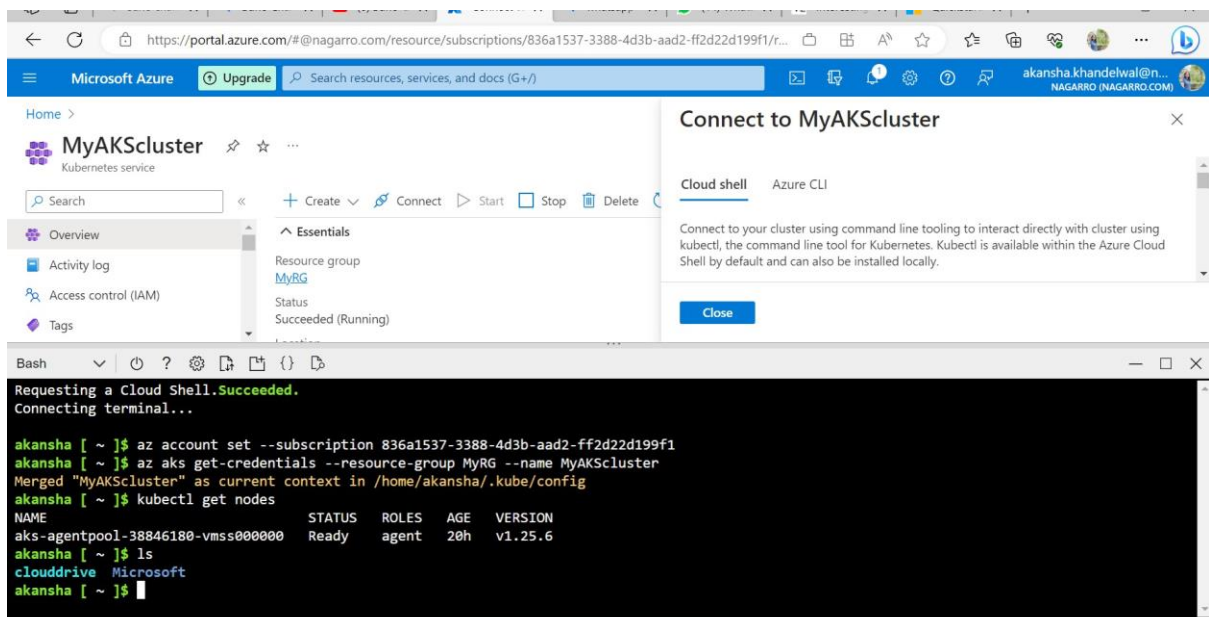


4. Create the AKS cluster and deploy services on it. Services should be accessible from the internet.

### Step-1 Created AKS cluster



### Step-2 Opening cloud-bash and checking nodes





### Step-3 Created yaml file and deployed Code on AKS cluster

The screenshot shows the Azure portal interface for a Kubernetes service named "MyAKScluster". The left sidebar contains navigation options: Overview, Activity log, Access control (IAM), and Tags. The main area displays the cluster's status as "Succeeded (Running)". A "Connect to MyAKScluster" dialog box is open, offering "Cloud shell" and "Azure CLI" options. Below the dialog, a terminal window shows the following commands and output:

```
Merged "MyAKScluster" as current context in /home/akansha/.kube/config
akansha [ ~ ]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
aks-agentpool-38846180-vmss000000 Ready     agent    20h    v1.25.6
akansha [ ~ ]$ ls
clouddrive  Microsoft
akansha [ ~ ]$ vi App.yml
akansha [ ~ ]$ kubectl apply -f App.yml
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
akansha [ ~ ]$
```

### Step-4 Checking services related to deployed code

The screenshot shows the Azure portal interface for the "MyAKScluster" page. The left sidebar contains navigation options: Overview, Activity log, Access control (IAM), and Tags. The main area displays the cluster's status as "Succeeded (Running)". A terminal window shows the following commands and output:

```
deployment.apps/azure-vote-front created
service/azure-vote-front created
akansha [ ~ ]$ kubectl get svc
NAME                                TYPE                CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
azure-vote-back                    ClusterIP           10.0.57.129   <none>         6379/TCP         2m54s
azure-vote-front                    LoadBalancer       10.0.109.19   52.149.255.155 80:30183/TCP     2m54s
kubernetes                         ClusterIP           10.0.0.1      <none>         443/TCP          20h
akansha [ ~ ]$ ls
App.yml  clouddrive  Microsoft
akansha [ ~ ]$ kubectl get all
NAME                                READY    STATUS    RESTARTS    AGE
pod/azure-vote-back-84ff4ffd4b-hqf5h 1/1      Running   0           7m17s
pod/azure-vote-front-579fd797d7-g6d44 1/1      Running   0           7m16s

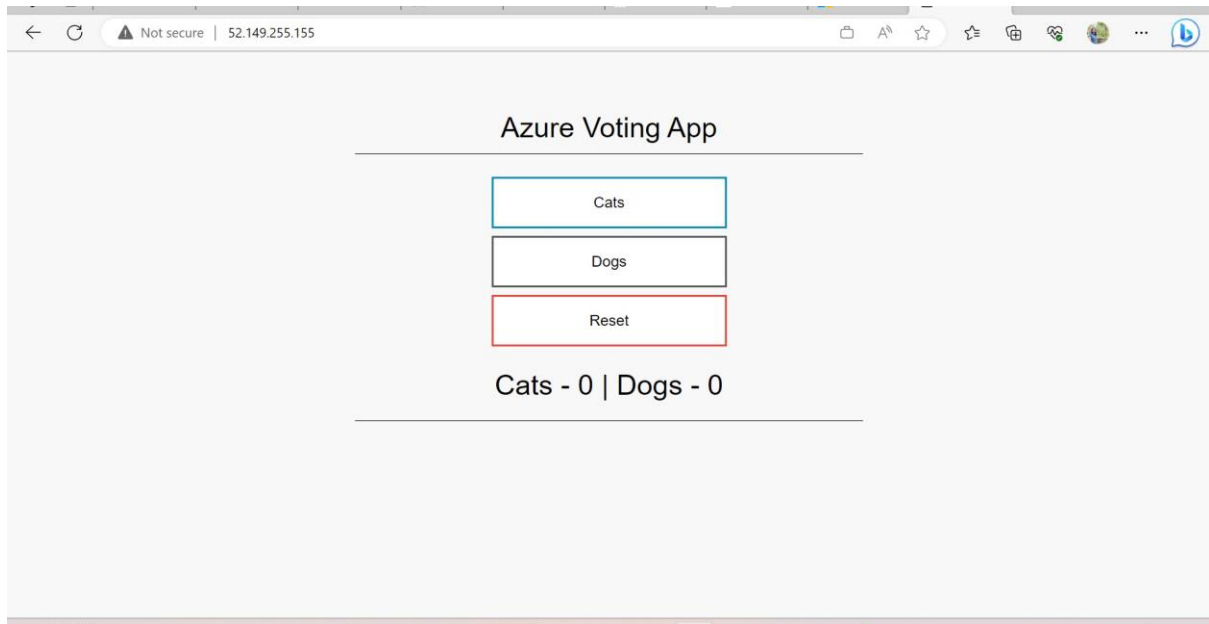
NAME                                TYPE                CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/azure-vote-back              ClusterIP           10.0.57.129   <none>         6379/TCP         7m17s
service/azure-vote-front              LoadBalancer       10.0.109.19   52.149.255.155 80:30183/TCP     7m17s
service/kubernetes                   ClusterIP           10.0.0.1      <none>         443/TCP          20h

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/azure-vote-back      1/1      1             1            7m18s
deployment.apps/azure-vote-front    1/1      1             1            7m17s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/azure-vote-back-84ff4ffd4b 1          1          1        7m18s
replicaset.apps/azure-vote-front-579fd797d7 1          1          1        7m17s
akansha [ ~ ]$
```

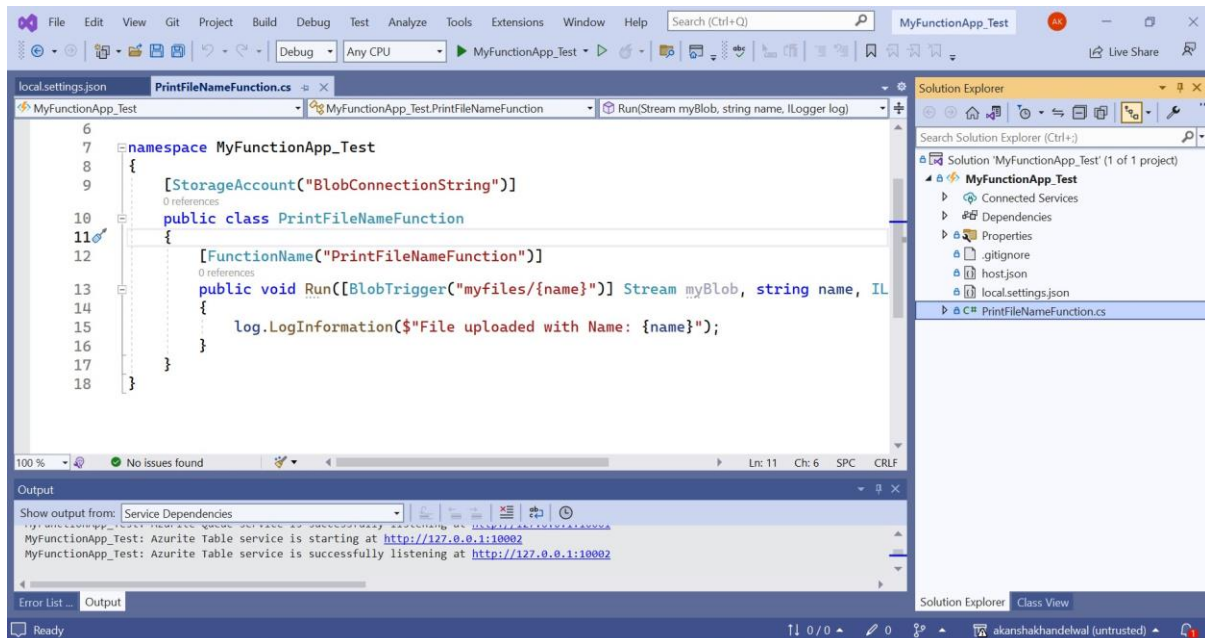


## Step-5 Deployed code checking through external-IP

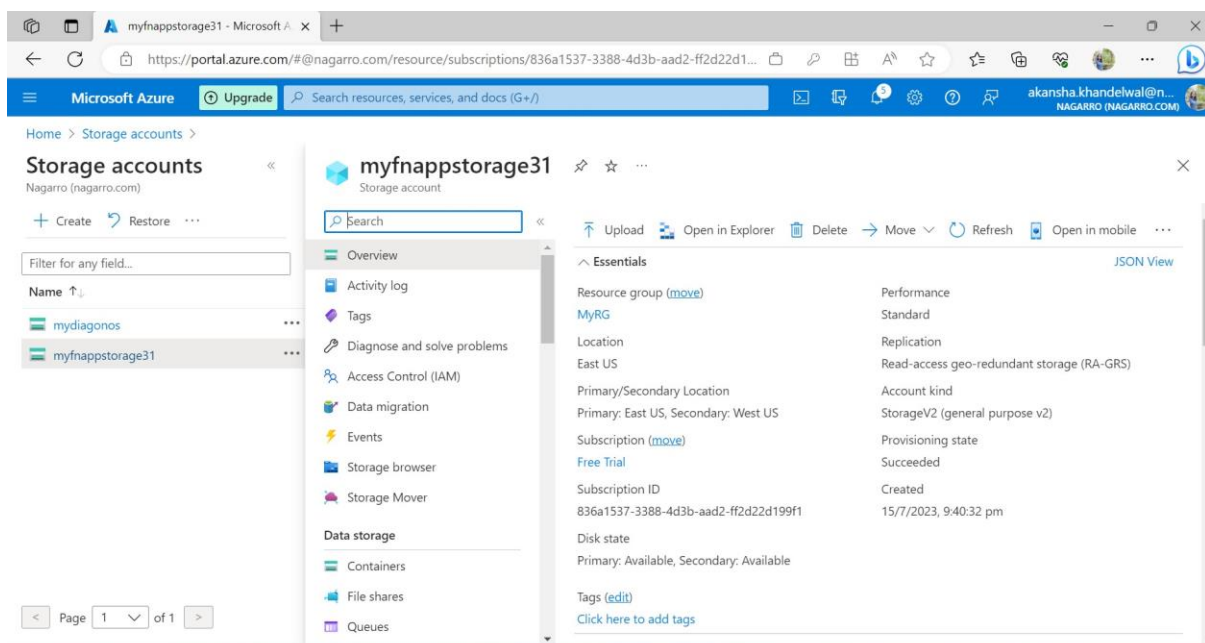


5. Create an Azure function that should trigger as soon as you upload a file in the blob storage. Function should be able to print the name of the file uploaded in the function.

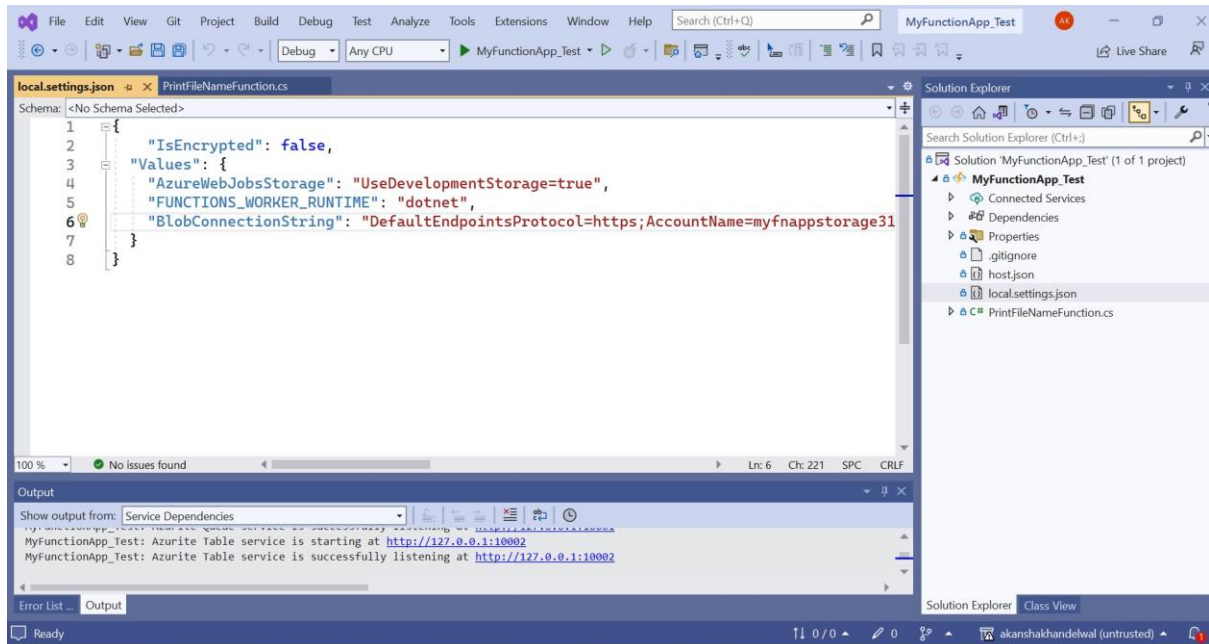
### Step-1 Creating Azure Function in Visual studio 2022



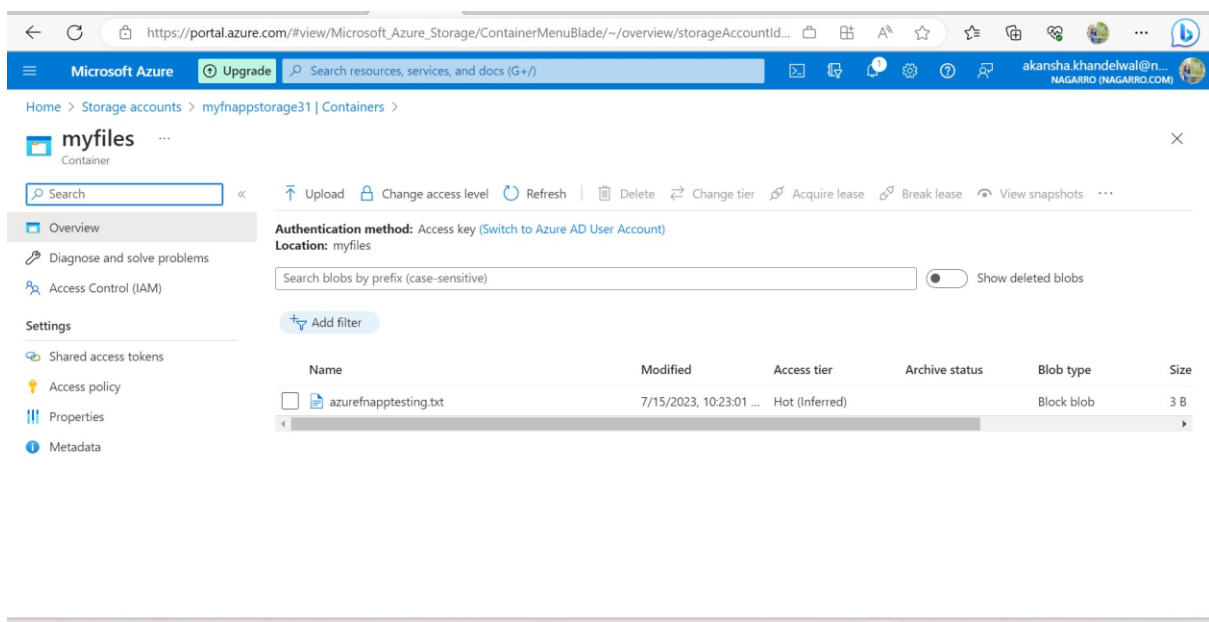
### Step-2 Created Storage Account



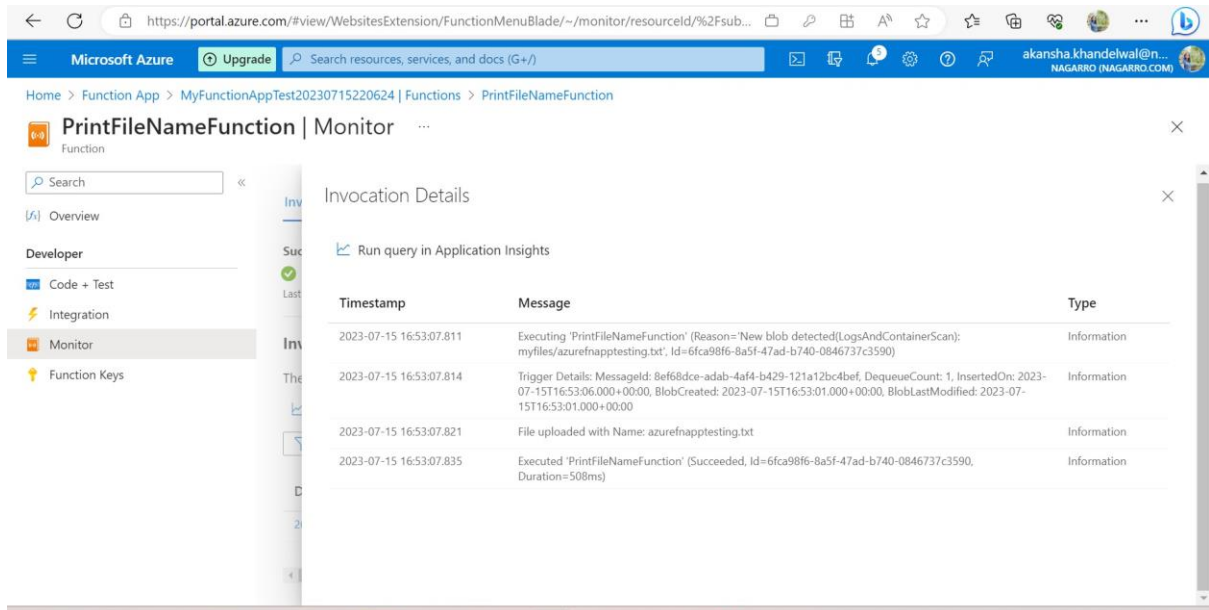
### Step-3 Setting up Blob Connection String of Storage Account



### Step-4 Created Container in Storage account and uploaded file



## Step-5 Blob Trigger Created while uploading file



The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, an 'Upgrade' button, and a search bar. The breadcrumb trail indicates the path: Home > Function App > MyFunctionAppTest20230715220624 | Functions > PrintFileNameFunction. The main heading is 'PrintFileNameFunction | Monitor'. On the left, there is a sidebar with a search bar and navigation links: Overview, Developer, Code + Test, Integration, Monitor (selected), and Function Keys. The main content area is titled 'Invocation Details' and includes a link to 'Run query in Application Insights'. Below this is a table with four columns: Timestamp, Message, and Type. The table contains four rows of invocation data.

Timestamp	Message	Type
2023-07-15 16:53:07.811	Executing 'PrintFileNameFunction' (Reason='New blob detected(LogsAndContainerScan); myfiles/azurefnapptesting.txt', Id=6fca98f6-8a5f-47ad-b740-0846737c3590)	Information
2023-07-15 16:53:07.814	Trigger Details: MessageId: 8ef68dce-adab-4a44-b429-121a12bc4bef, DequeueCount: 1, InsertedOn: 2023-07-15T16:53:06.000+00:00, BlobCreated: 2023-07-15T16:53:01.000+00:00, BlobLastModified: 2023-07-15T16:53:01.000+00:00	Information
2023-07-15 16:53:07.821	File uploaded with Name: azurefnapptesting.txt	Information
2023-07-15 16:53:07.835	Executed 'PrintFileNameFunction' (Succeeded, Id=6fca98f6-8a5f-47ad-b740-0846737c3590, Duration=508ms)	Information