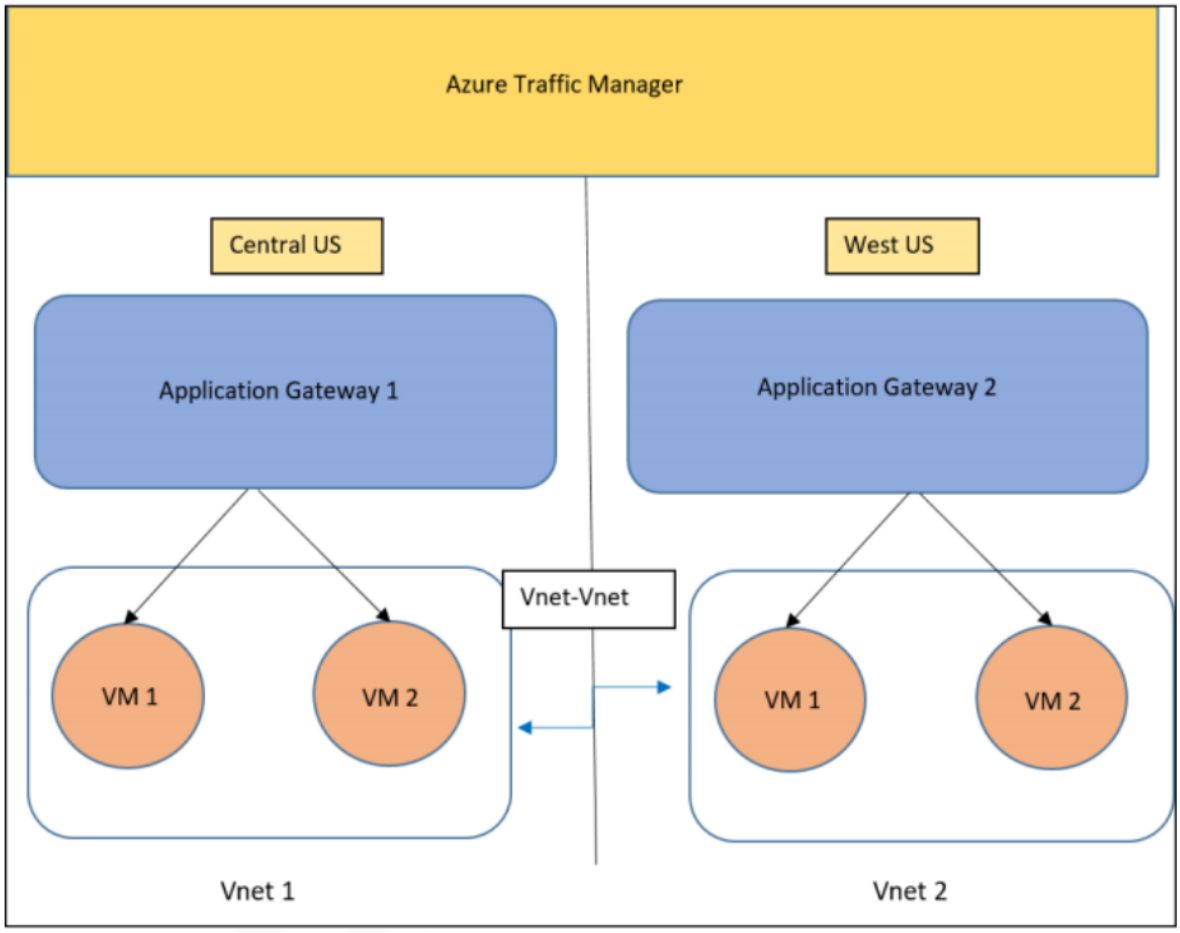
Azure administrator project-

Project description

You work as an Azure professional for a Corporation. You are assigned the task of implementing the below architecture for the company's website.



There are three web pages to be deployed:

1. The home page is the default page (VM2)
2. The upload page is where you can upload the files to your Azure Blob Storage (VM1)
3. The error page for 403 and 502 errors

Application Gateway has to be configured in the following manner:

1. Example.com should be pointed to the home page
2. Example.com/upload should be pointed to the upload page

3. Application Gateway's error pages should be pointed to error.html which should be hosted as a static website in Azure Containers. The error.html file is present in the GitHub repository

The term 'Example' here refers to the Traffic Manager's domain name.
The client wants you to deploy them in the Central US and the West US regions such that the traffic is distributed optimally between both regions.

Storage Account has to be configured in the following manner:

1. You need to host your error.html as a static website here, and then point the application gateway's 403 and 502 errors to it.
2. Create a container named upload, this will be used by your code to upload the files.

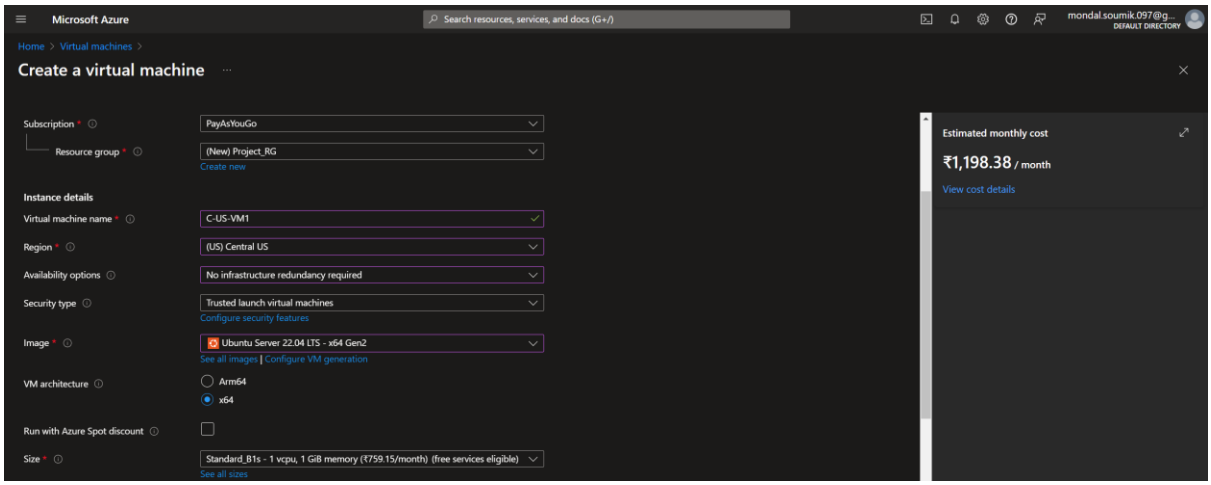
Technical specifications for the deployments are as follows:

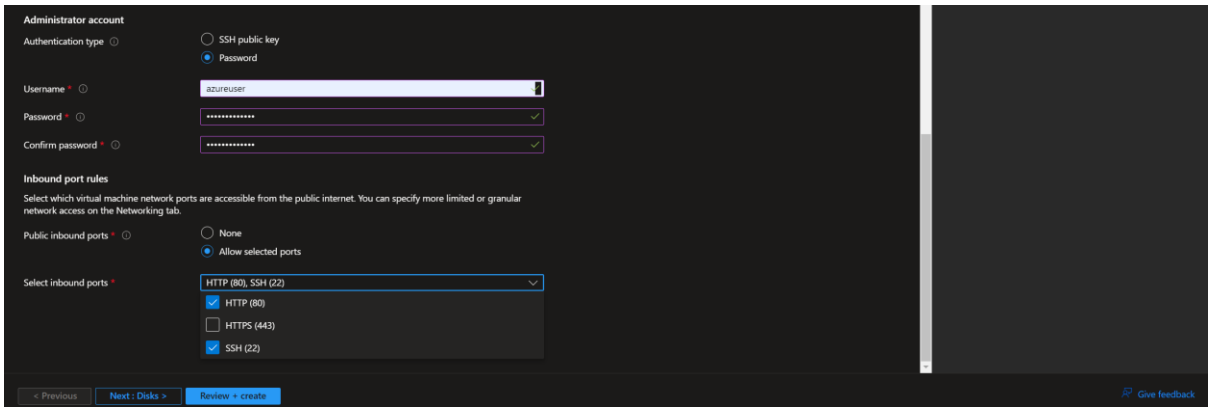
1. Deployments in both regions should have VMs inside VNets.
2. Clone the GitHub repo https://github.com/azcloudberg/azproject to all the VMs.
3. On VM1, please run vm1.sh this will deploy the upload page, on VM2 please run VM2.sh, this will install the home page.
4. For running the scripts, please run the following command inside the GitHub directory from the terminal.

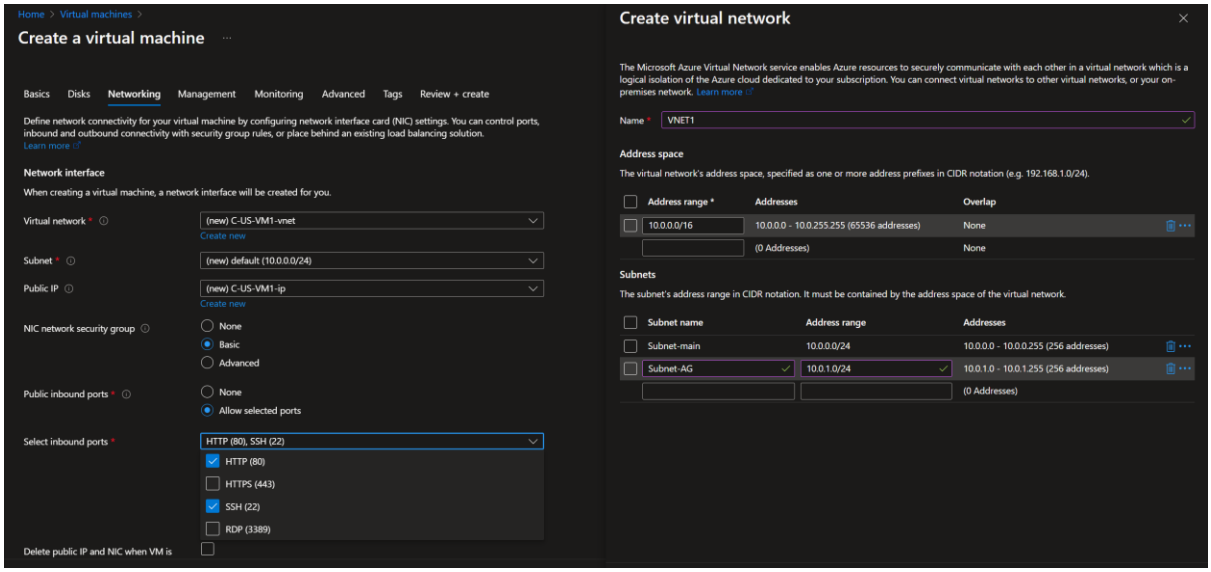
VM1: ./vm1.sh
VM2: ./vm2.sh

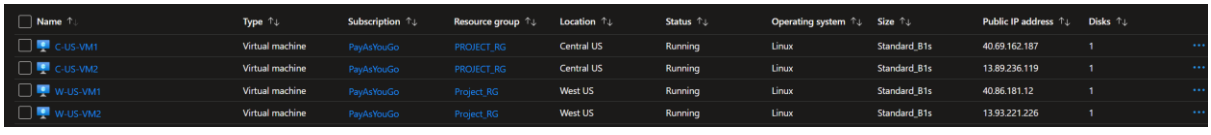
5. After running the scripts, please edit the config.py file on VM1, and enter the details related to your storage account where the files will be uploaded.
6. Once done, please run the following command: **sudo python3 app.py**
7. Both regions should be connected to each other using VNet-VNet Peering.
8. Finally, your Traffic Manager should be pointing to the application gateway of both the regions.

Step 1: - Created 2 virtual machines in Central US and 2 virtual machines in West US region.
Attached Central US VMs with VNET1 and West US with VNET2 Virtual Networks.





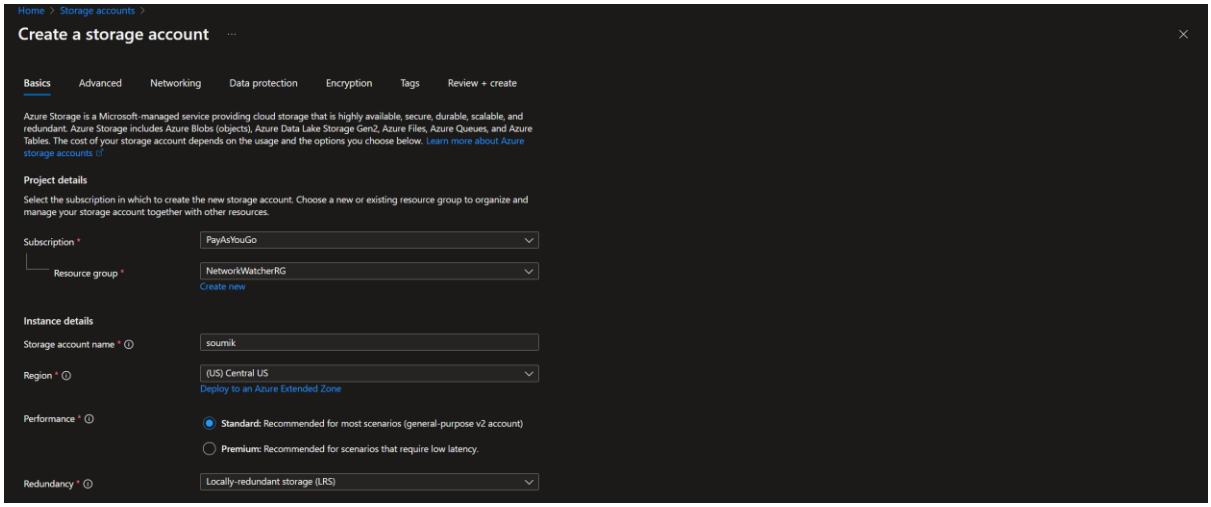


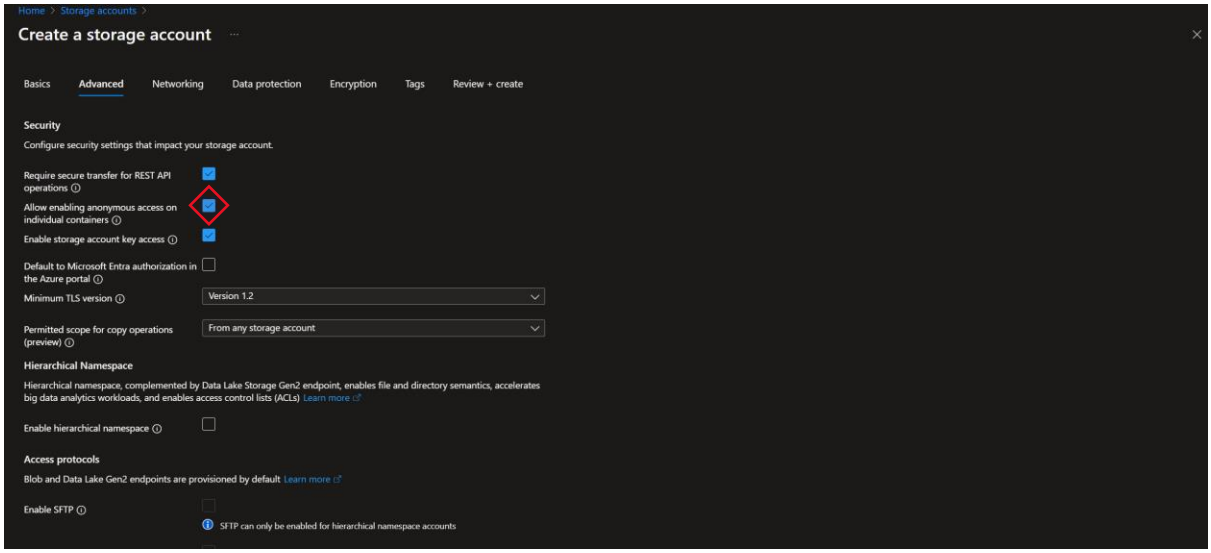


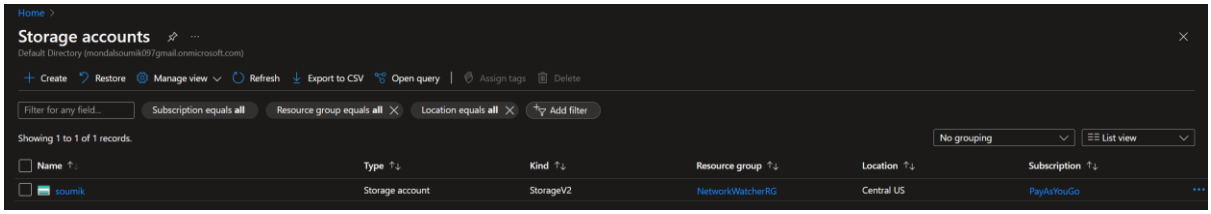
Name	Type	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disks
C-US-VM1	Virtual machine	PayAsYouGo	PROJECT_RG	Central US	Running	Linux	Standard_B1s	40.69.162.187	1
C-US-VM2	Virtual machine	PayAsYouGo	PROJECT_RG	Central US	Running	Linux	Standard_B1s	13.89.236.119	1
W-US-VM1	Virtual machine	PayAsYouGo	Project_RG	West US	Running	Linux	Standard_B1s	40.86.181.12	1
W-US-VM2	Virtual machine	PayAsYouGo	Project_RG	West US	Running	Linux	Standard_B1s	13.93.221.226	1

Using the same procedure created VNET2, subnet-main, and subnet-AG

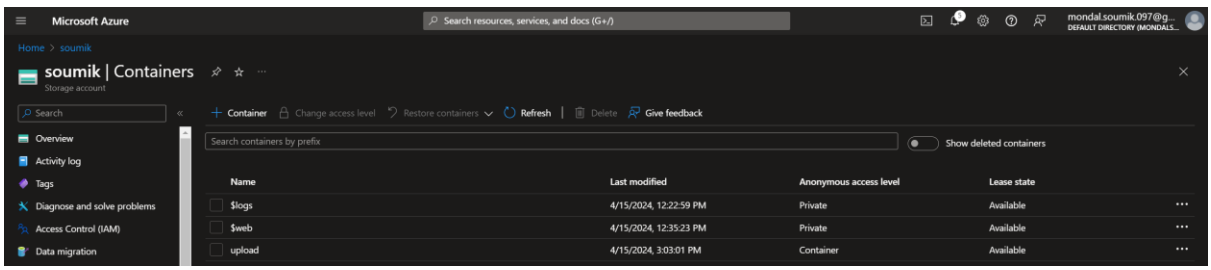
Creating a storage account for storing the data



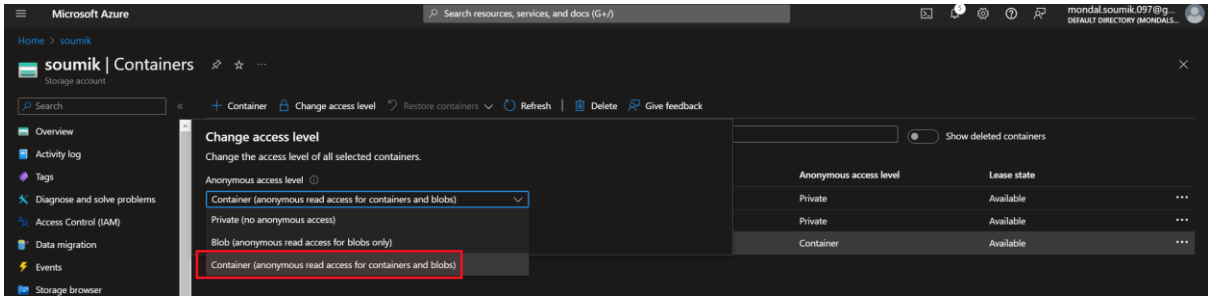


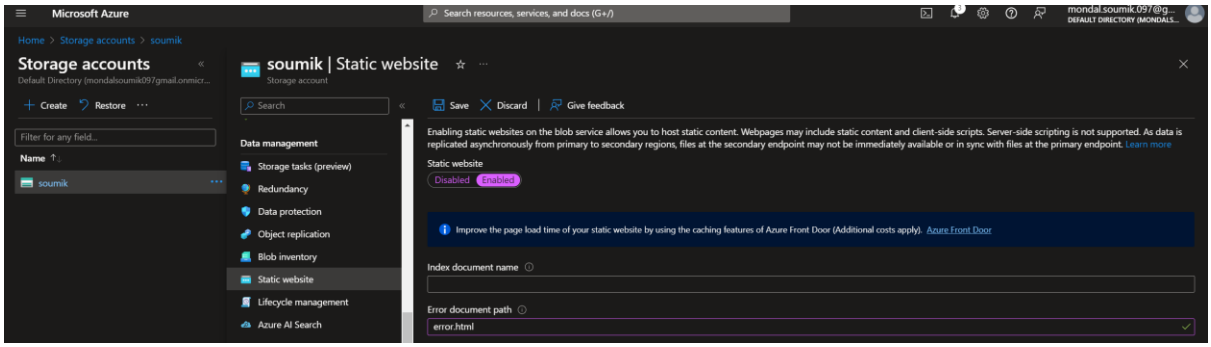


Successfully create a storage account

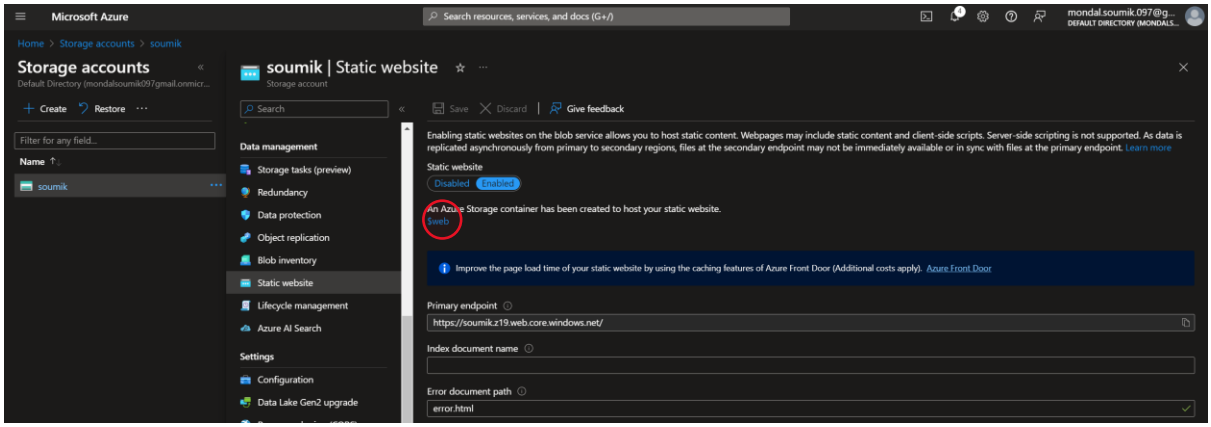


Created a container naming “upload”.





Creating a static website to access the website outside world

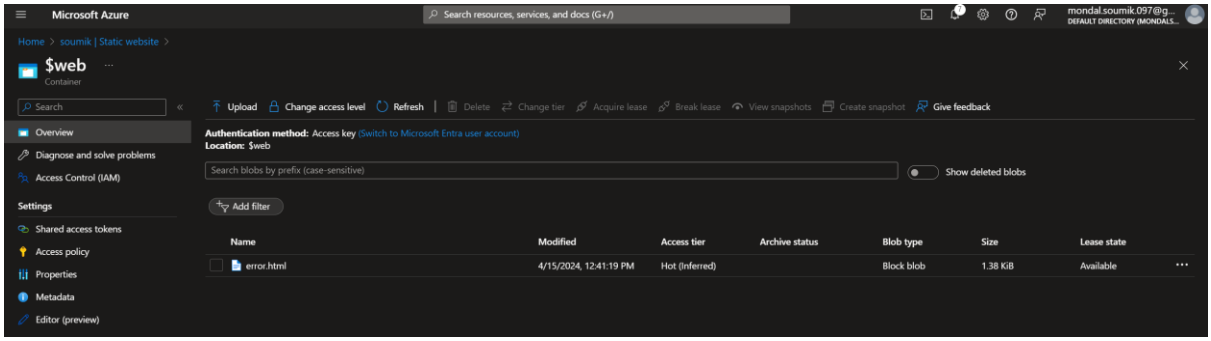


Now creating and uploading an error.html page-

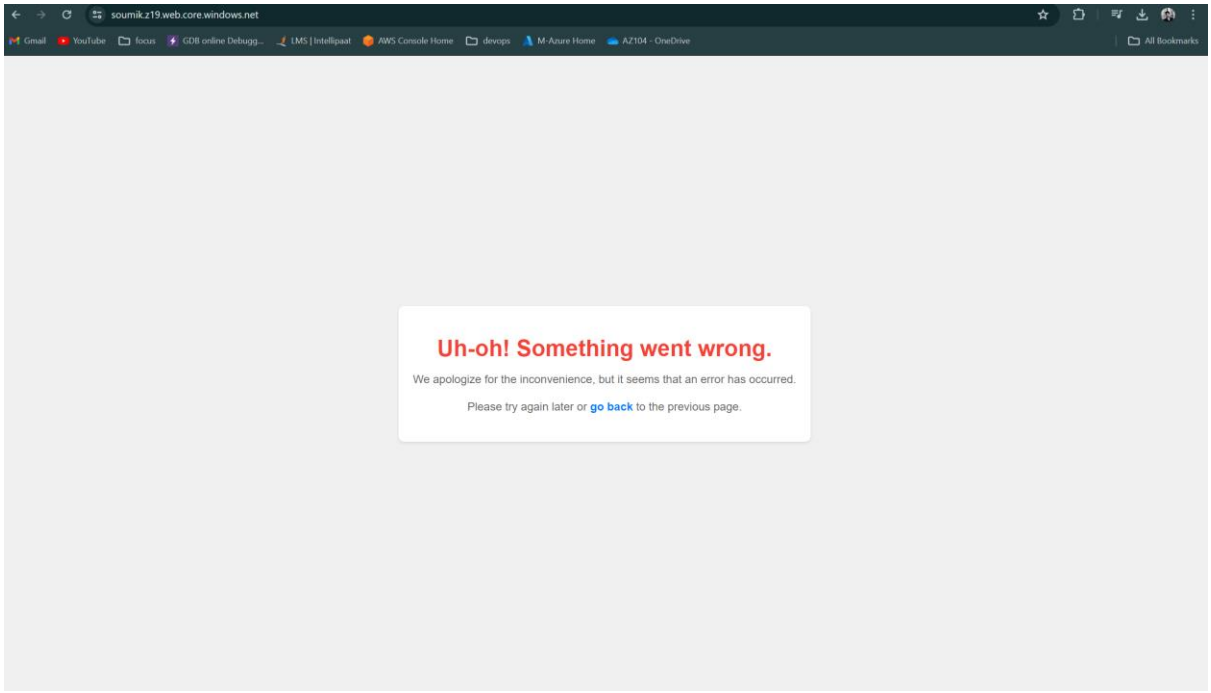
error.html-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Error Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
```

```
            align-items: center;
            height: 100vh;
            text-align: center;
        }
        .error-container {
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 2px 4px rgba(0,0,0,0.1);
        }
        h1 {
            color: #f44336;
            margin-bottom: 10px;
        }
        p {
            color: #666;
            margin-bottom: 20px;
        }
        a {
            color: #007bff;
            text-decoration: none;
            font-weight: bold;
        }
        a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
    <div class="error-container">
        <h1>Uh-oh! Something went wrong.</h1>
        <p>We apologize for the inconvenience, but it seems that an error has
occurred.</p>
        <p>Please try again later or <a href="javascript:history.back()">go
back</a> to the previous page.</p>
    </div>
</body>
</html>
```

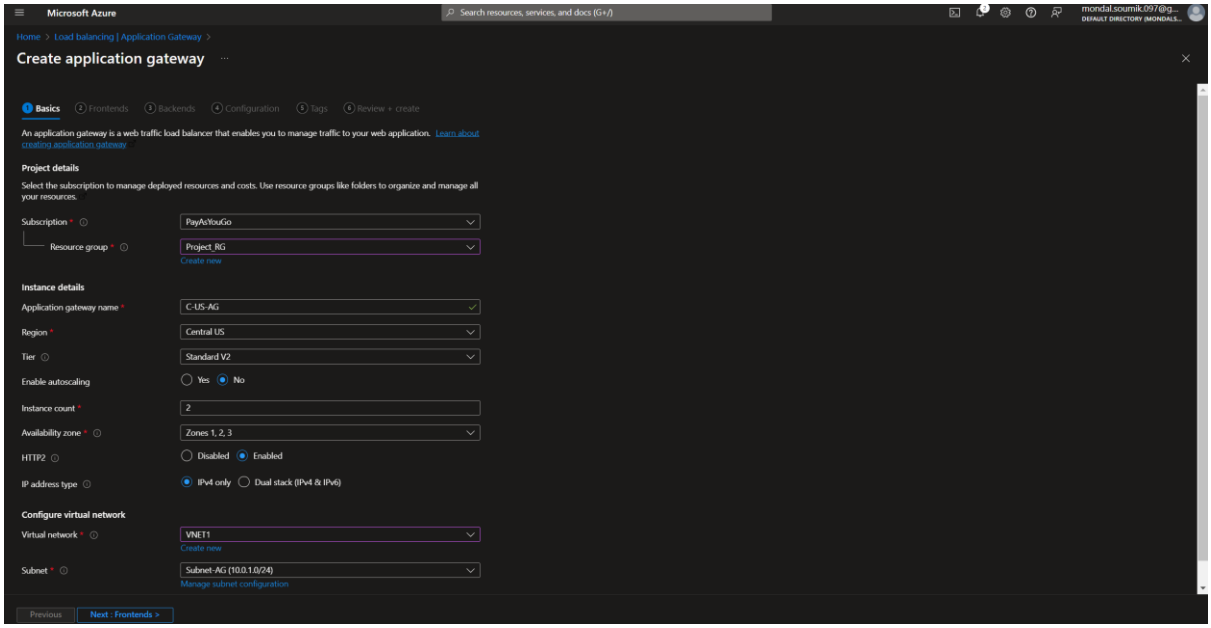


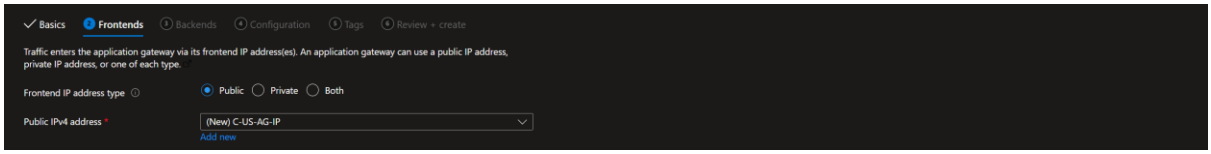
Uploaded the error.html page in $web path

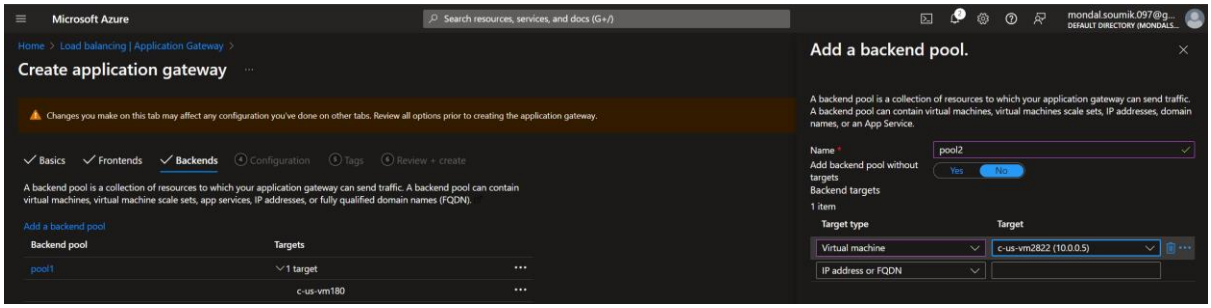


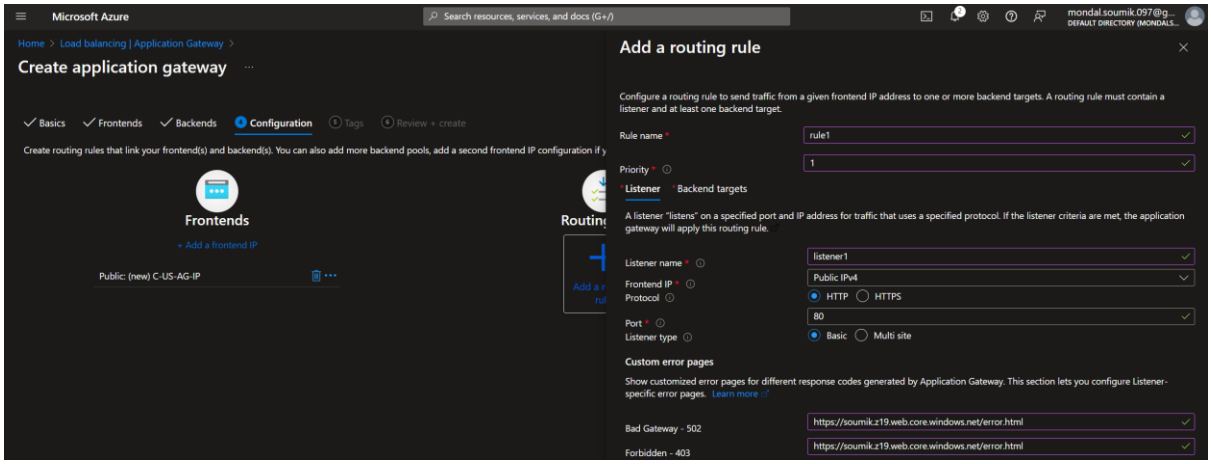
Successfully running the error.html page

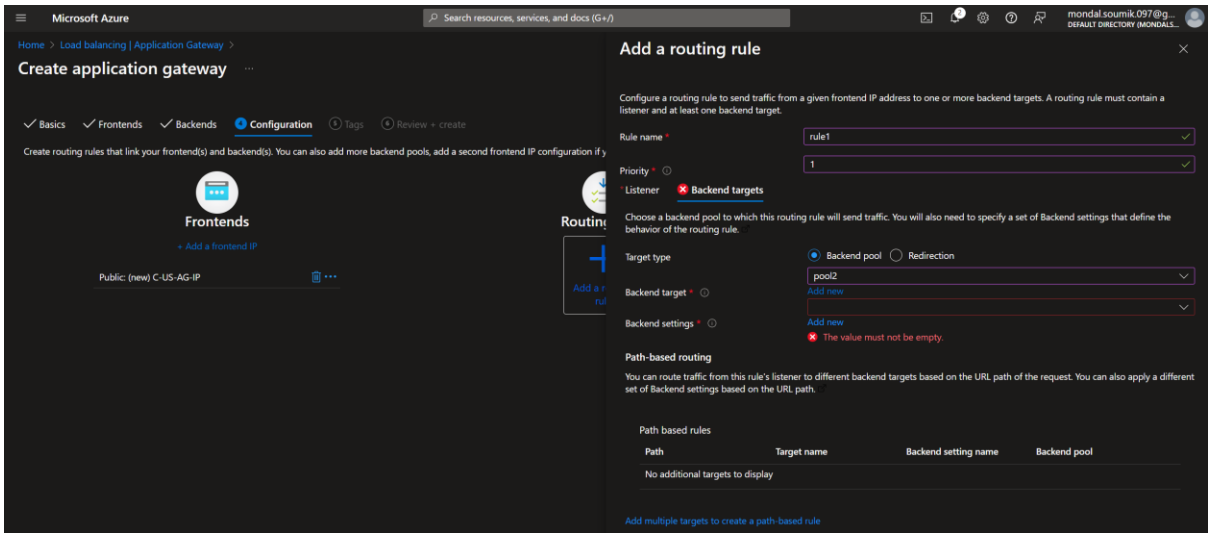
Now, let's create an application gateway for path based routing in Central US region-

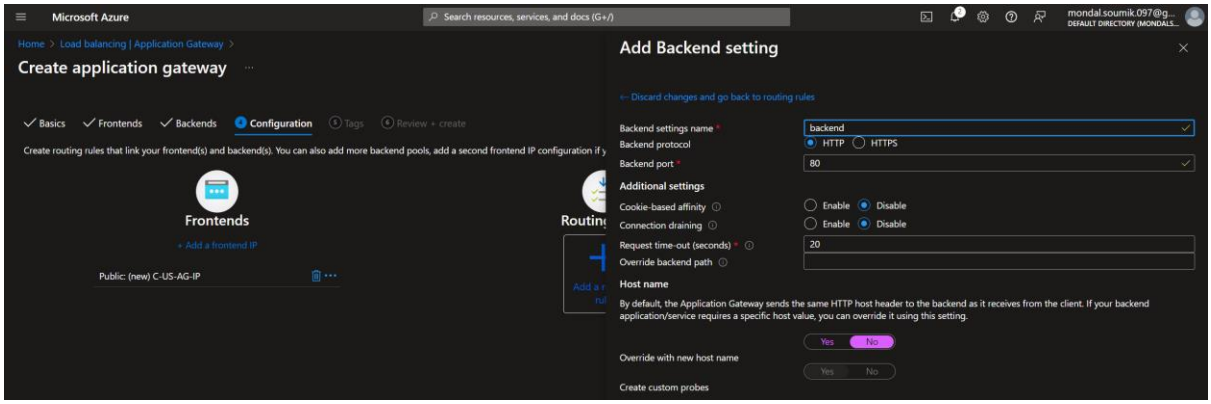


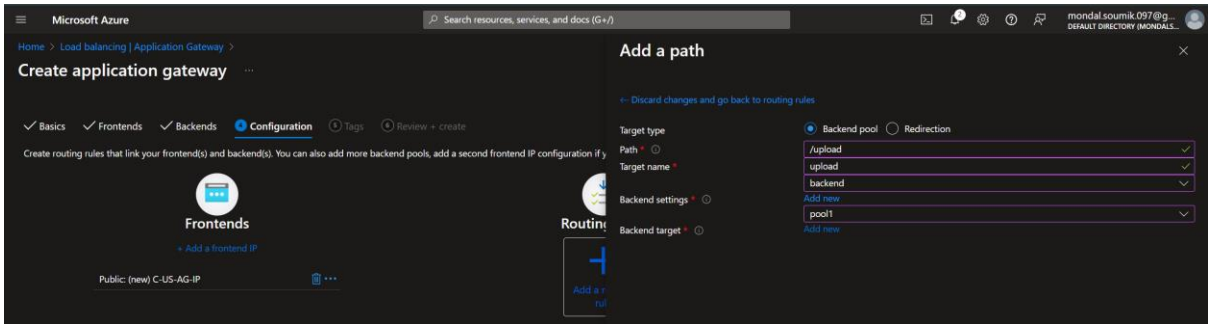

Basics
Frontends
Backends
Configuration
Tags
Review + create
Traffic enters the application gateway via its frontend IP address(es). An application gateway can use a public IP address, private IP address, or one of each type.
Frontend IP address type
Public
Private
Both
Public IPv4 address *
(New) C-US-AG-IP
Add new


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Load balancing | Application Gateway >
Create application gateway
Changes you make on this tab may affect any configuration you've done on other tabs. Review all options prior to creating the application gateway.
Basics
Frontends
Backends
Configuration
Tags
Review + create
A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).
Add a backend pool
Backend pool
Targets
pool1
1 target
c-us-vm180
Add a backend pool.
A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.
Name *
pool2
Add backend pool without targets
Yes
No
Backend targets
1 item
Target type
Target
Virtual machine
c-us-vm2822 (10.0.0.5)
IP address or FQDN

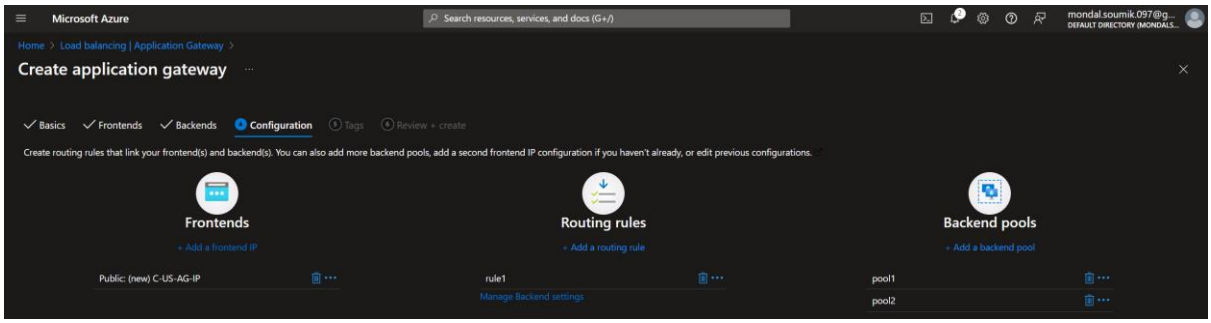

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Load balancing | Application Gateway >
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration if y
Frontends
+ Add a frontend IP
Public: (new) C-US-AG-IP
Add a routing rule
Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.
Rule name *
rule1
Priority *
1
Listener
Backend targets
A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.
Listener name *
listener1
Frontend IP *
Public IPv4
Protocol
HTTP
HTTPS
Port *
80
Listener type
Basic
Multi site
Custom error pages
Show customized error pages for different response codes generated by Application Gateway. This section lets you configure Listener-specific error pages. Learn more
Bad Gateway - 502
https://soumik.z19.web.core.windows.net/error.html
Forbidden - 403
https://soumik.z19.web.core.windows.net/error.html


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Load balancing | Application Gateway >
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration if y
Frontends
+ Add a frontend IP
Public: (new) C-US-AG-IP
Add a routing rule
Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.
Rule name *
rule1
Priority *
1
Listener
Backend targets
Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of Backend settings that define the behavior of the routing rule.
Target type
Backend pool
Redirection
Backend target *
pool2
Add new
Backend settings *
Add new
The value must not be empty.
Path-based routing
You can route traffic from this rule's listener to different backend targets based on the URL path of the request. You can also apply a different set of Backend settings based on the URL path.
Path based rules
Path
Target name
Backend setting name
Backend pool
No additional targets to display
Add multiple targets to create a path-based rule

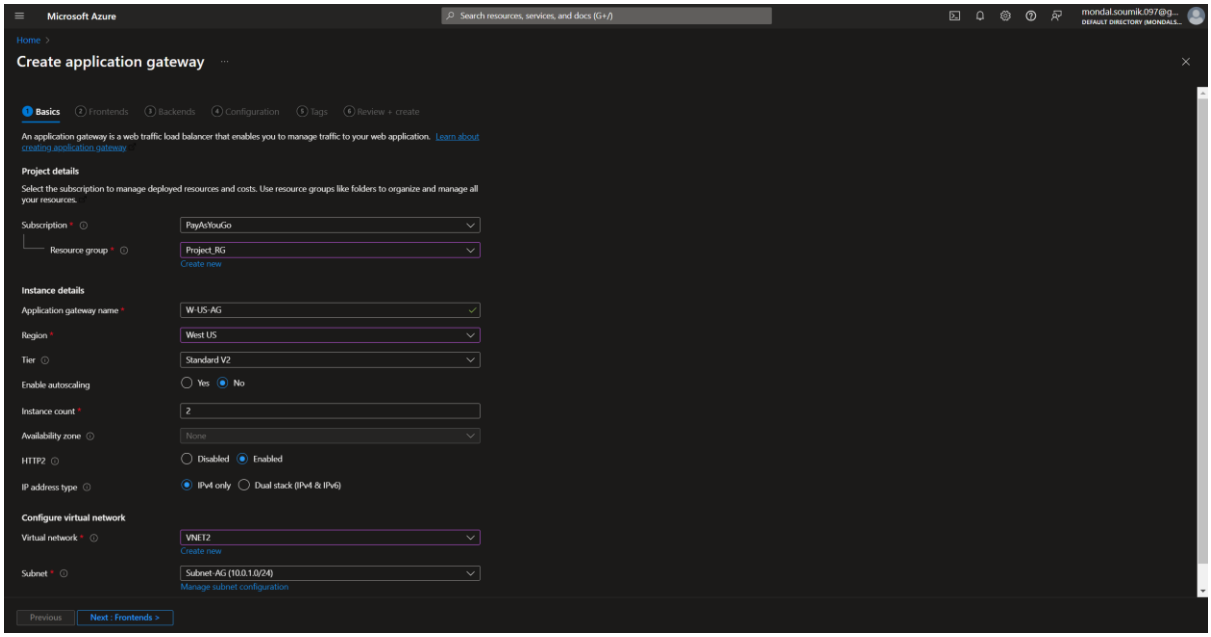


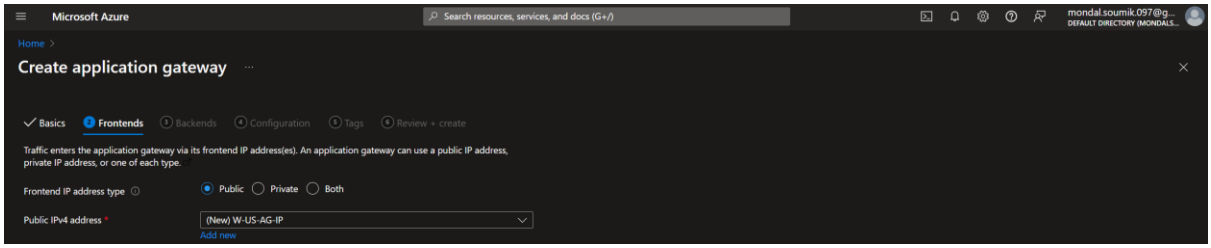


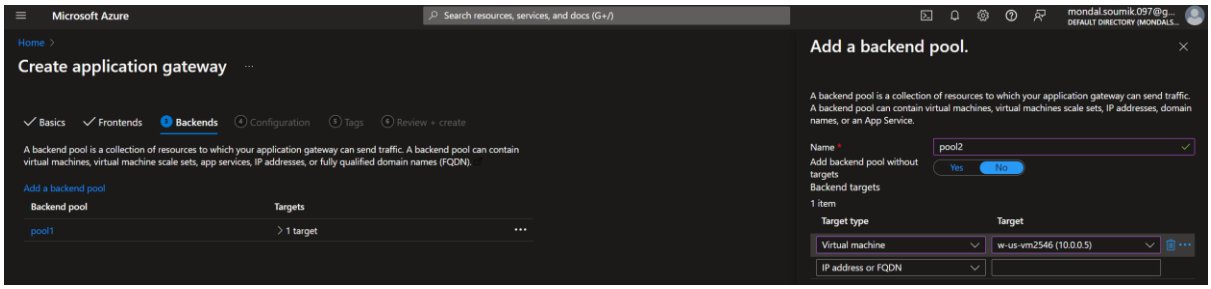
pointing the path on '/upload' for path based routing

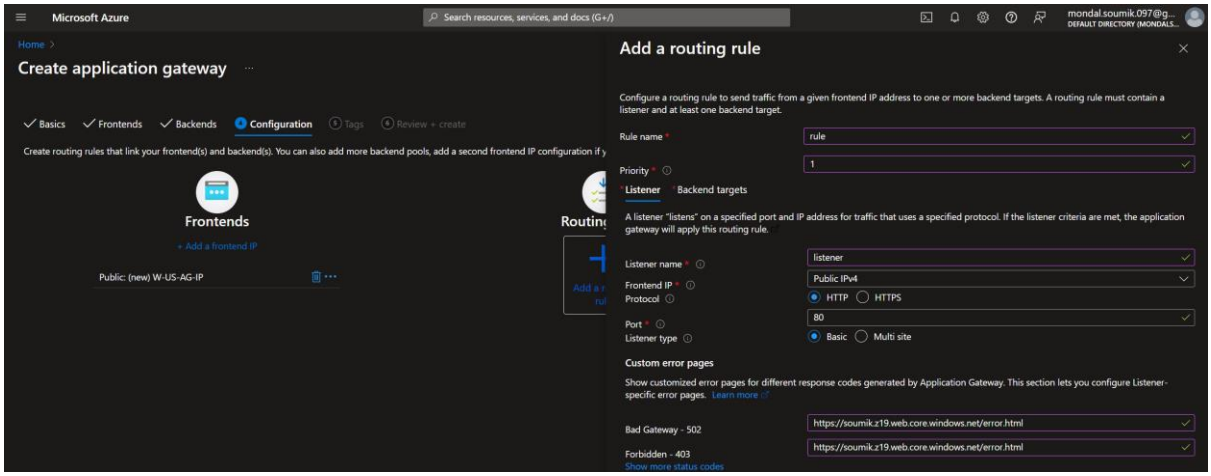


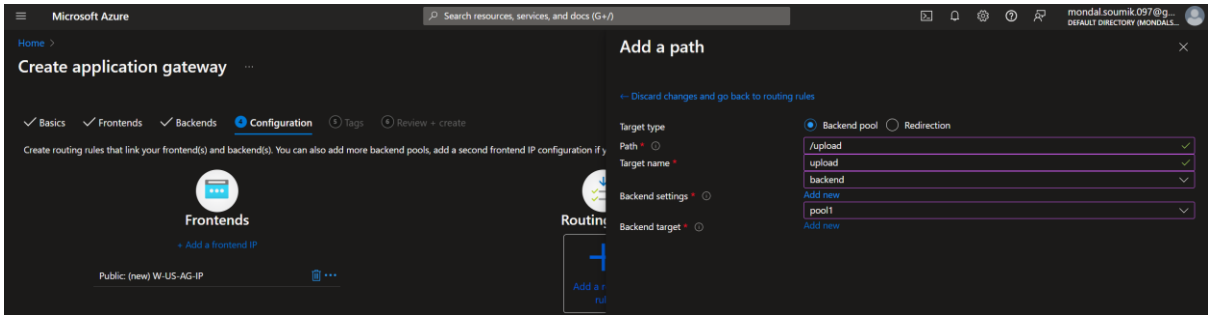
Next we will create an application gateway for path based routing in West US region-

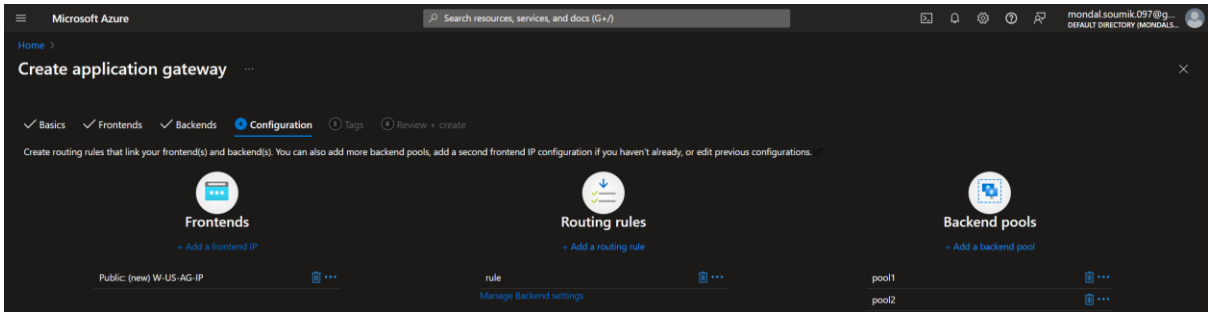


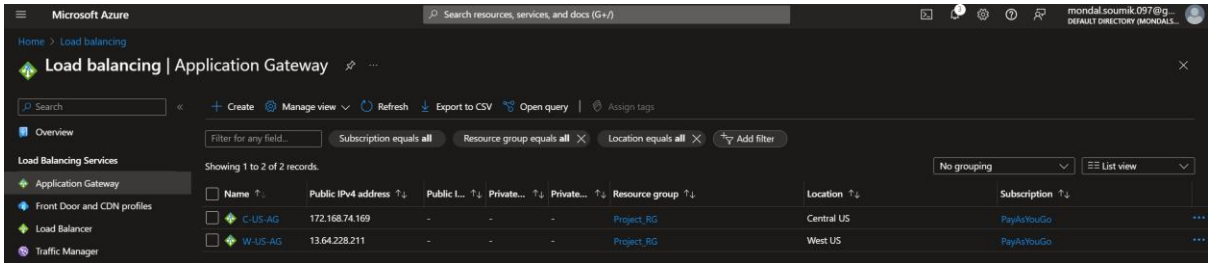

Microsoft Azure
Search resources, services, and docs (G+/)
mondal.soumik.097@g...
DEFAULT DIRECTORY (MONDALS...
Home
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
Traffic enters the application gateway via its frontend IP address(es). An application gateway can use a public IP address, private IP address, or one of each type.
Frontend IP address type
Public
Private
Both
Public IPv4 address *
(New) W-US-AG-IP
Add new


Microsoft Azure
Search resources, services, and docs (G+/)
mondal.soumik.097@g...
DEFAULT DIRECTORY (MONDALS...
Home
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).
Add a backend pool
Backend pool
Targets
pool1
1 target
Add a backend pool.
A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.
Name *
pool2
Add backend pool without targets
Yes
No
Backend targets
1 item
Target type
Target
Virtual machine
w-us-vm2546 (10.0.0.5)
IP address or FQDN

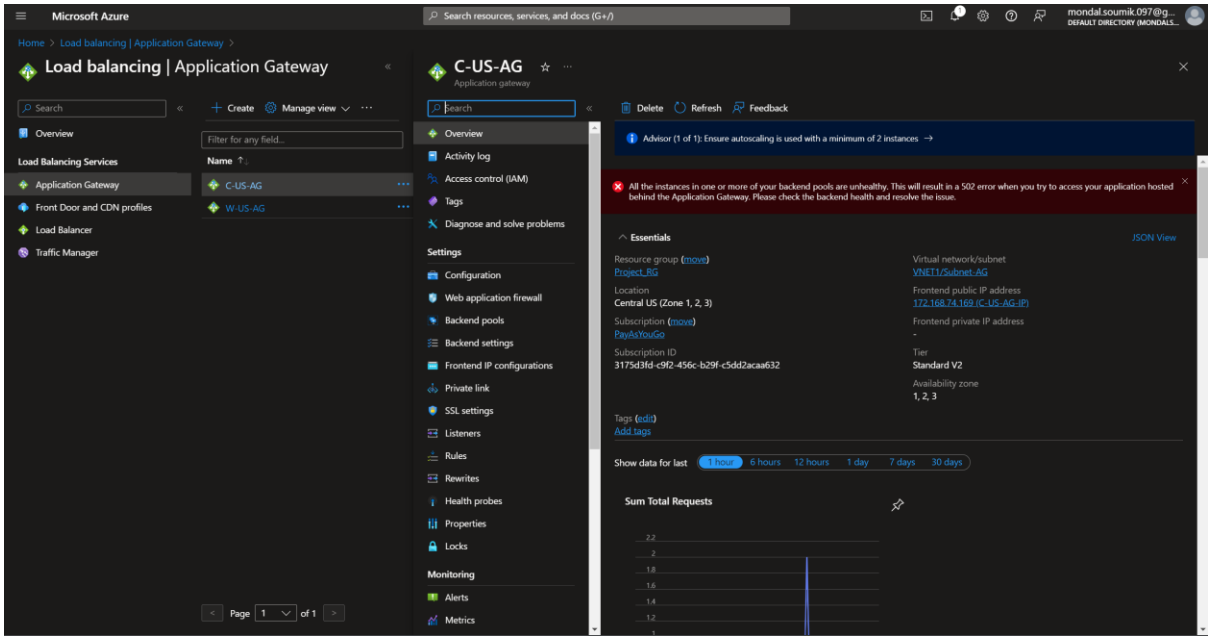

Microsoft Azure
Search resources, services, and docs (G+/)
mondal.soumik.097@g...
DEFAULT DIRECTORY (MONDALS...
Home
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration if y
Frontends
+ Add a frontend IP
Public: (new) W-US-AG-IP
Add a routing rule
Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.
Rule name *
rule
Priority *
1
Listener
Backend targets
A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.
Listener name *
listener
Frontend IP *
Public IPv4
Protocol
HTTP
HTTPS
Port *
80
Listener type
Basic
Multi site
Custom error pages
Show customized error pages for different response codes generated by Application Gateway. This section lets you configure Listener-specific error pages. Learn more
Bad Gateway - 502
https://soumik.z19.web.core.windows.net/error.html
Forbidden - 403
https://soumik.z19.web.core.windows.net/error.html
Show more status codes

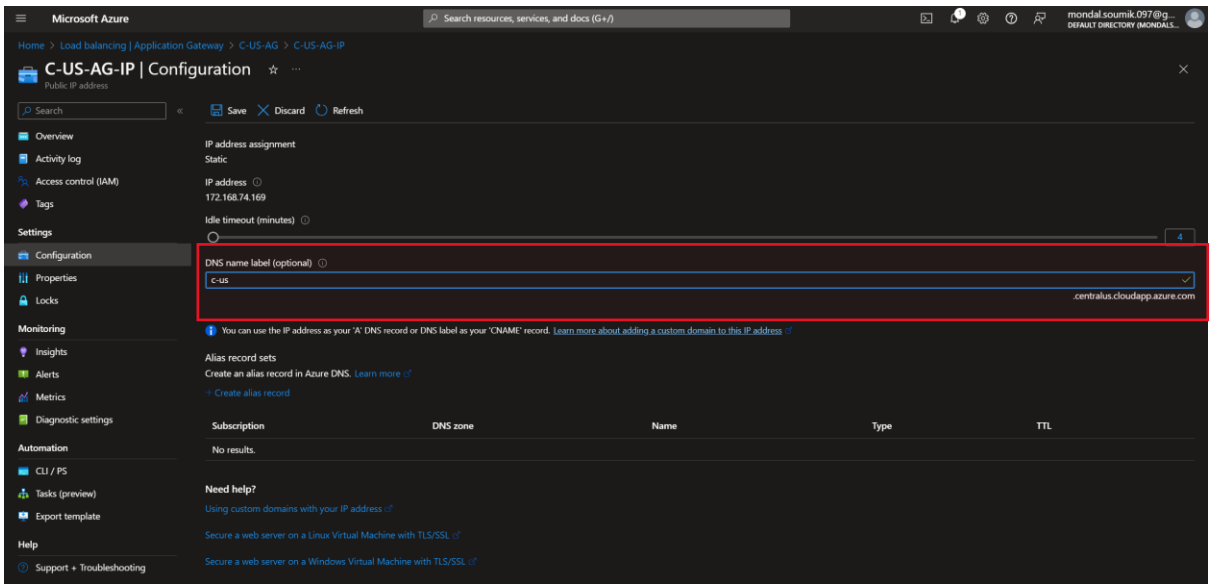

Microsoft Azure
Search resources, services, and docs (G+/)
mondal.soumik.097@g...
DEFAULT DIRECTORY (MONDALS...
Home
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration if y
Frontends
+ Add a frontend IP
Public: (new) W-US-AG-IP
Add a path
← Discard changes and go back to routing rules
Target type
Backend pool
Redirection
Path *
/upload
Target name *
upload
Backend settings *
backend
Add new
Backend target *
pool1
Add new

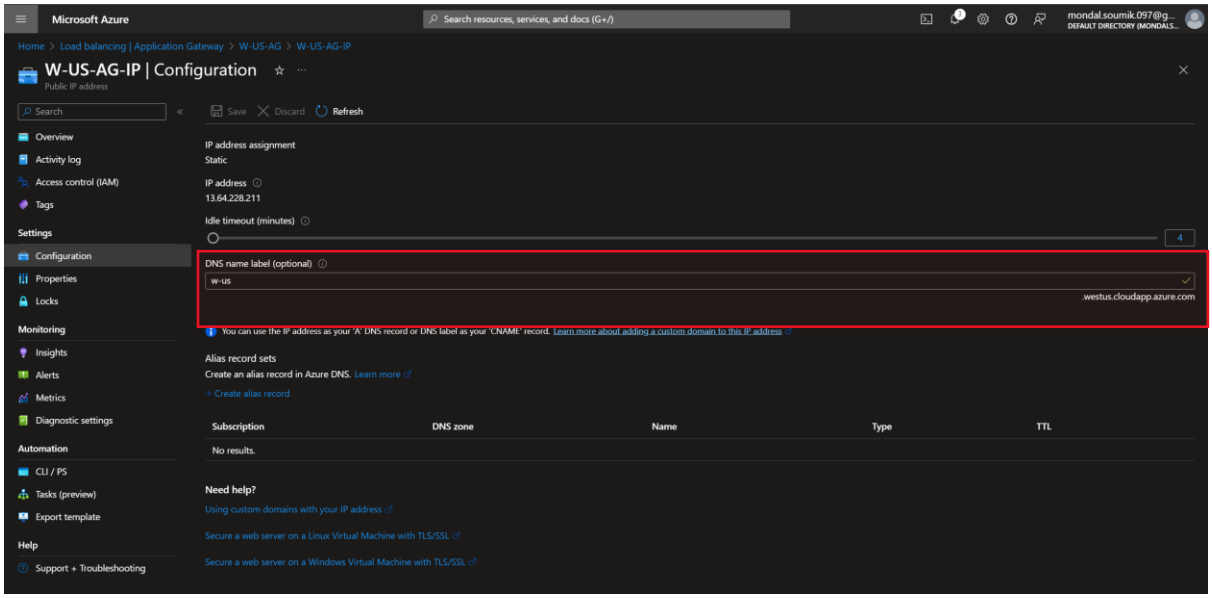

Microsoft Azure
Search resources, services, and docs (G+/)
mondal.soumik.097@g...
DEFAULT DIRECTORY (MONDALS...
Home
Create application gateway
Basics
Frontends
Backends
Configuration
Tags
Review + create
Create routing rules that link your frontend(s) and backend(s). You can also add more backend pools, add a second frontend IP configuration if you haven't already, or edit previous configurations.
Frontends
+ Add a frontend IP
Public: (new) W-US-AG-IP
Routing rules
+ Add a routing rule
rule
Manage Backend settings
Backend pools
+ Add a backend pool
pool1
pool2



Successfully created Application Gateway in the West US and Central US region.

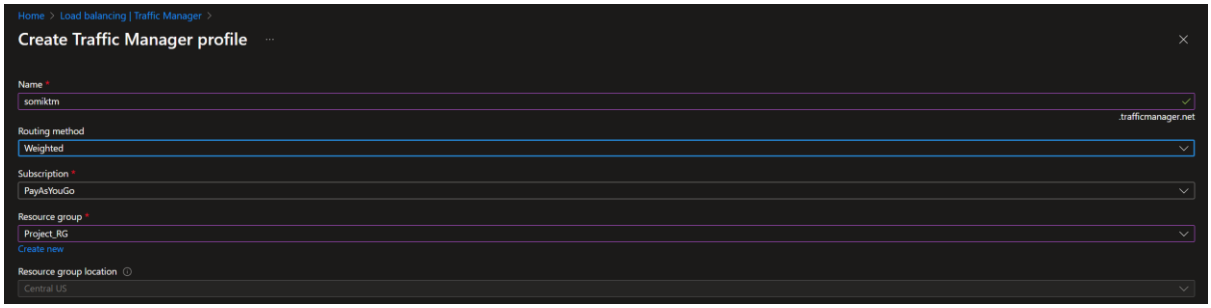


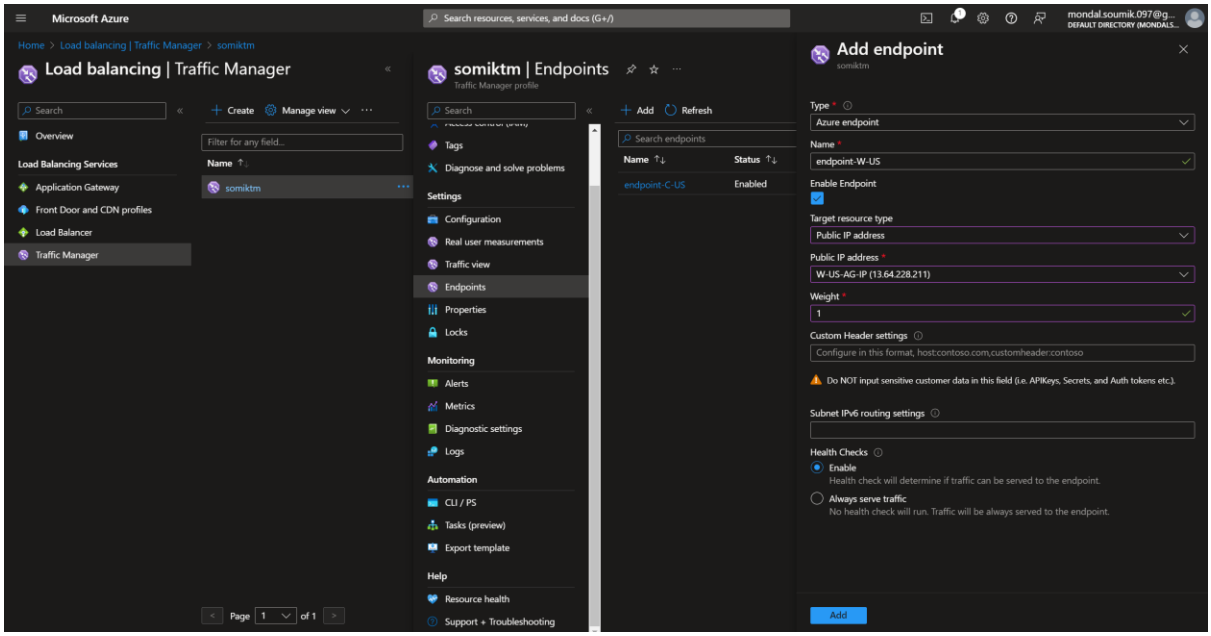




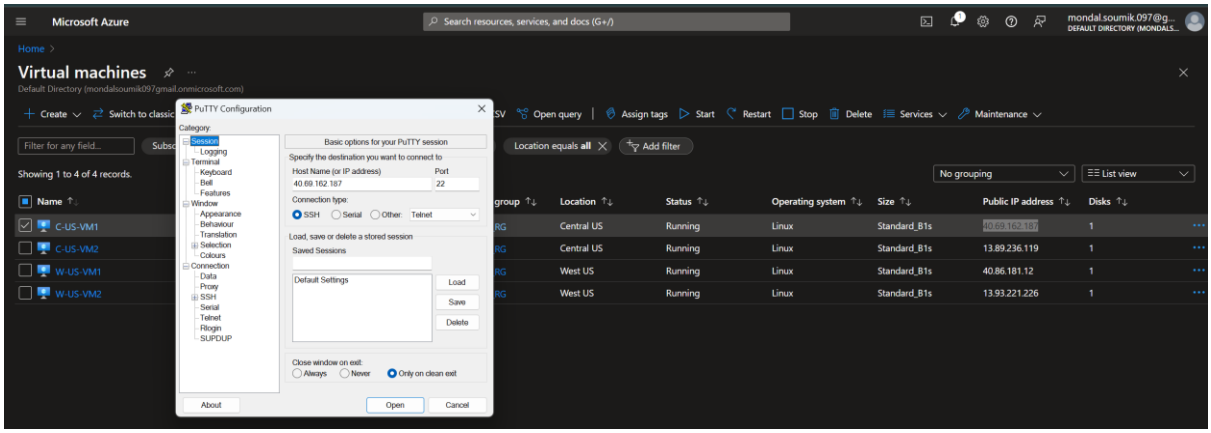
Creating “DNS name” for traffic manages because Traffic manager route traffic on DNS names

Creating Traffic manager that will route the traffic on the region level

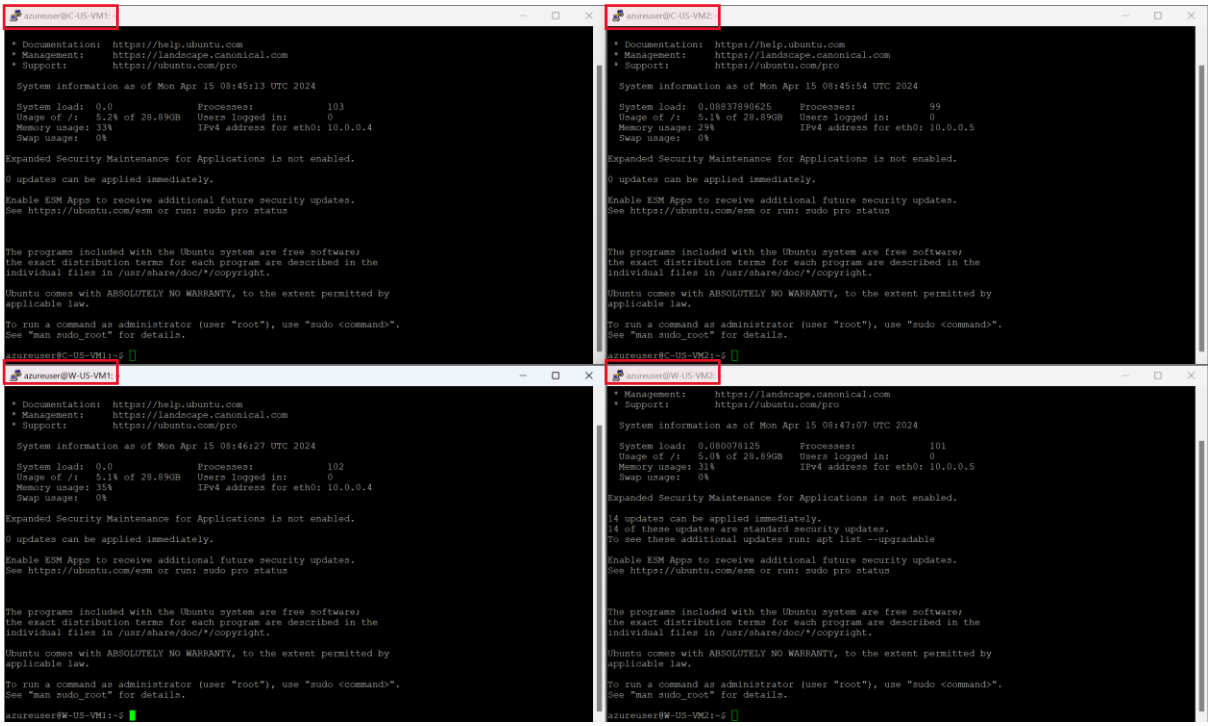


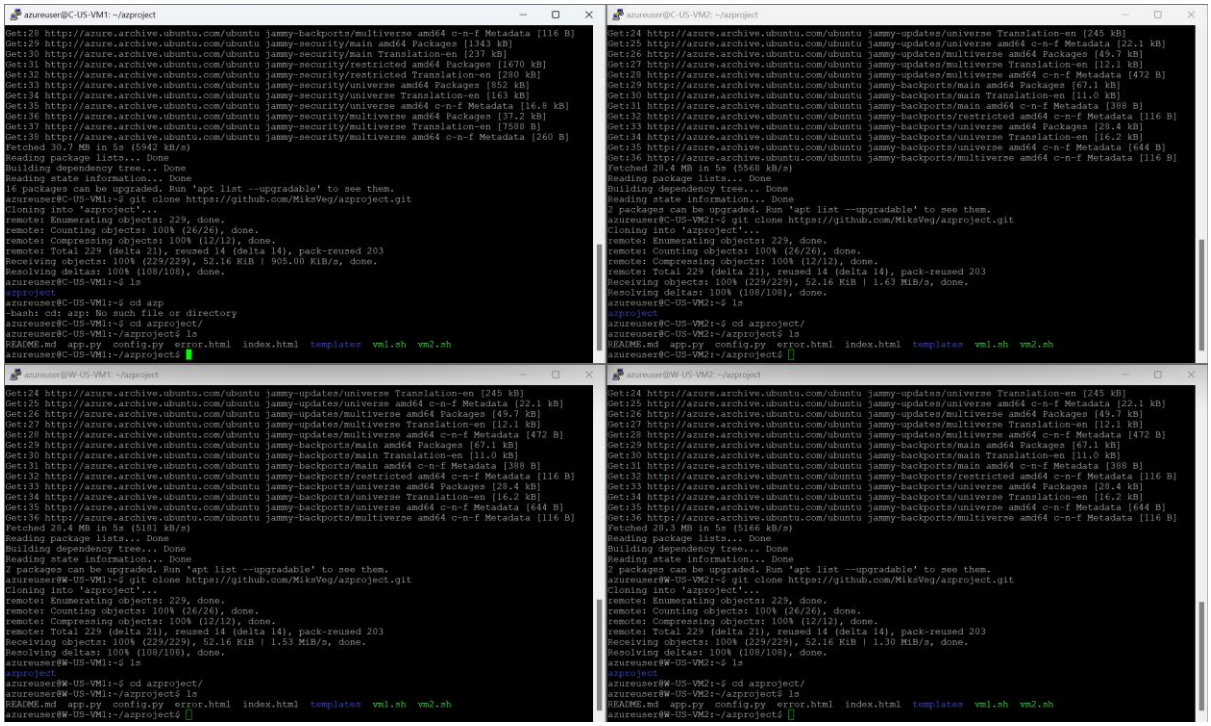


Now let’s configure the webpage.

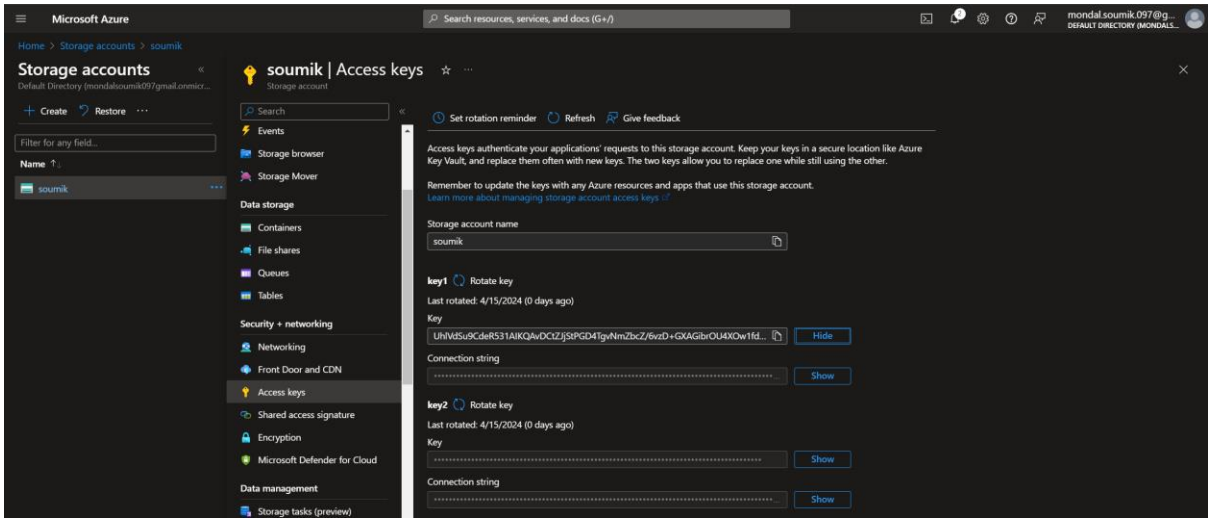


SSH to the Virtual machines

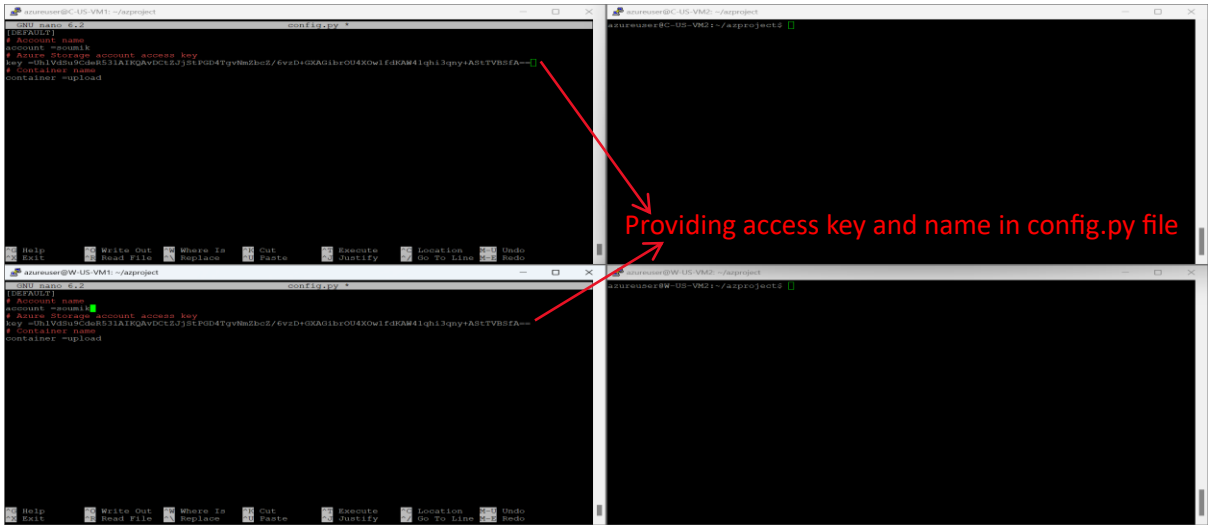


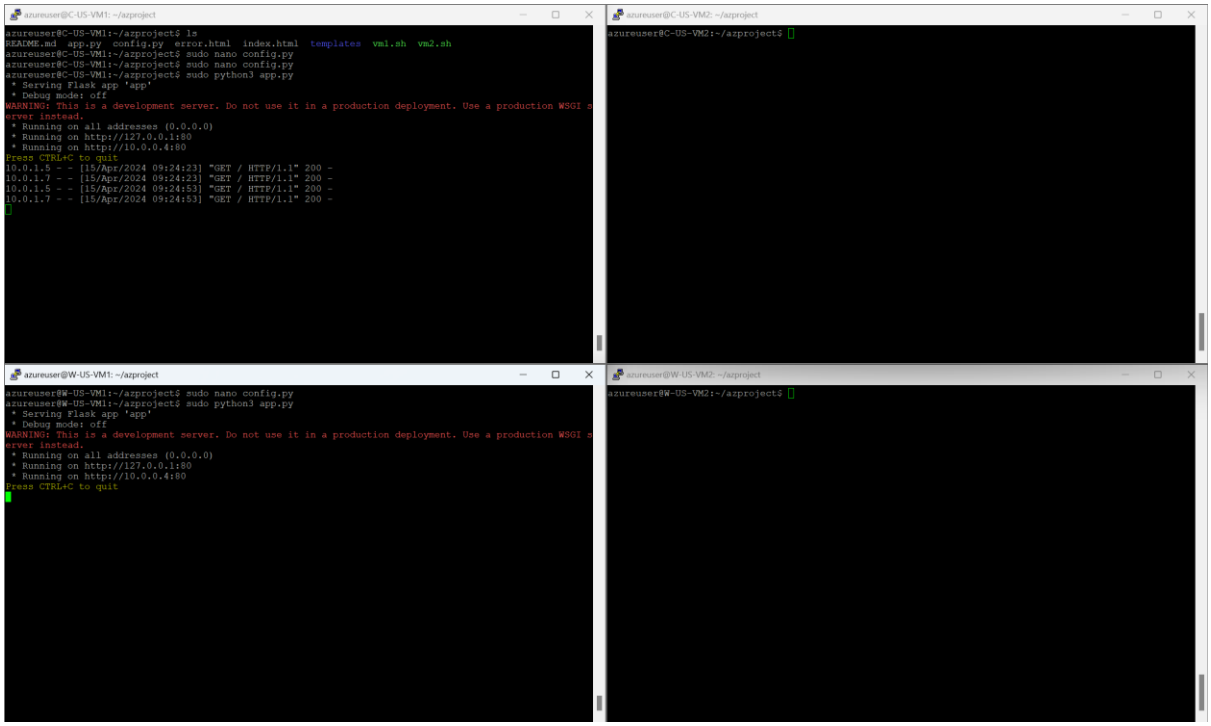


As per the project description running **./vm1.sh** on VM1 and **./vm2.sh** on VM2

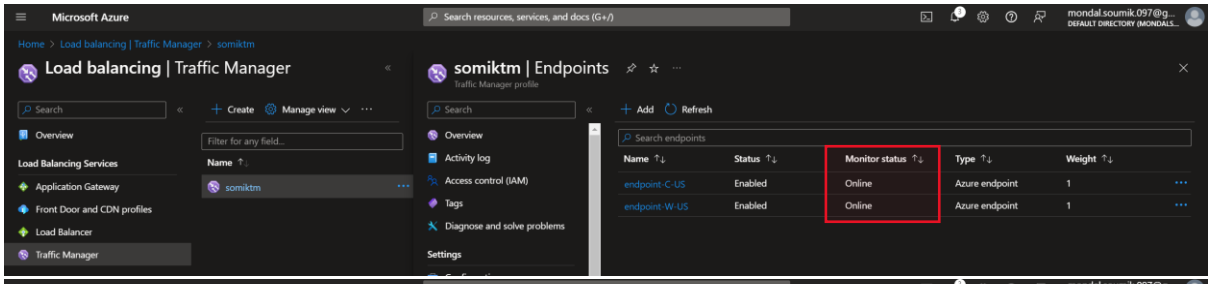


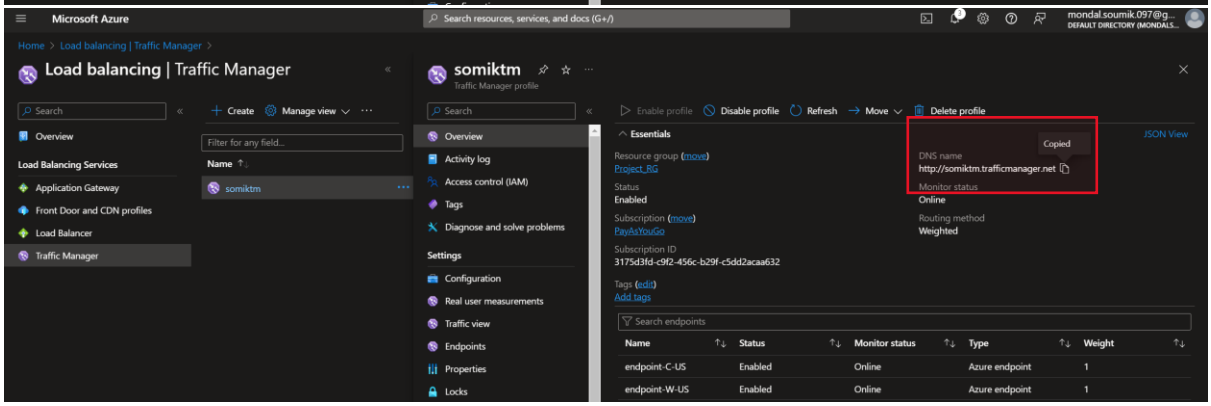
Generating access key in Storage account



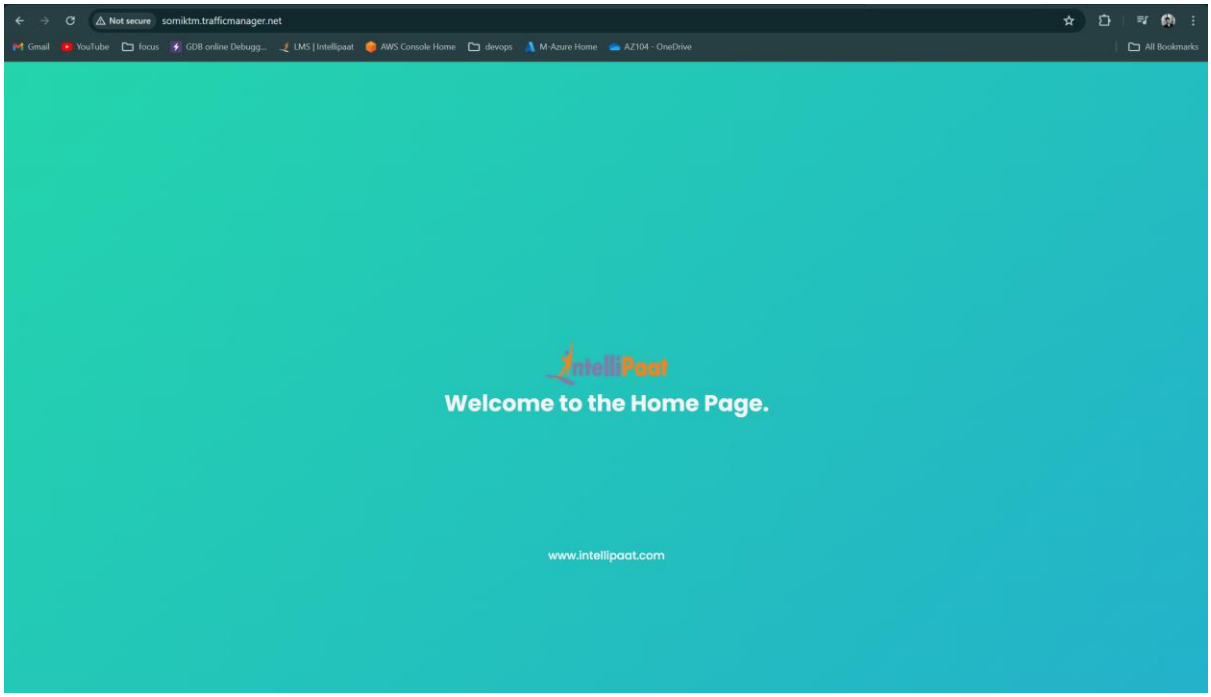


Now executing the python script on VM1 – sudo python3 app.py

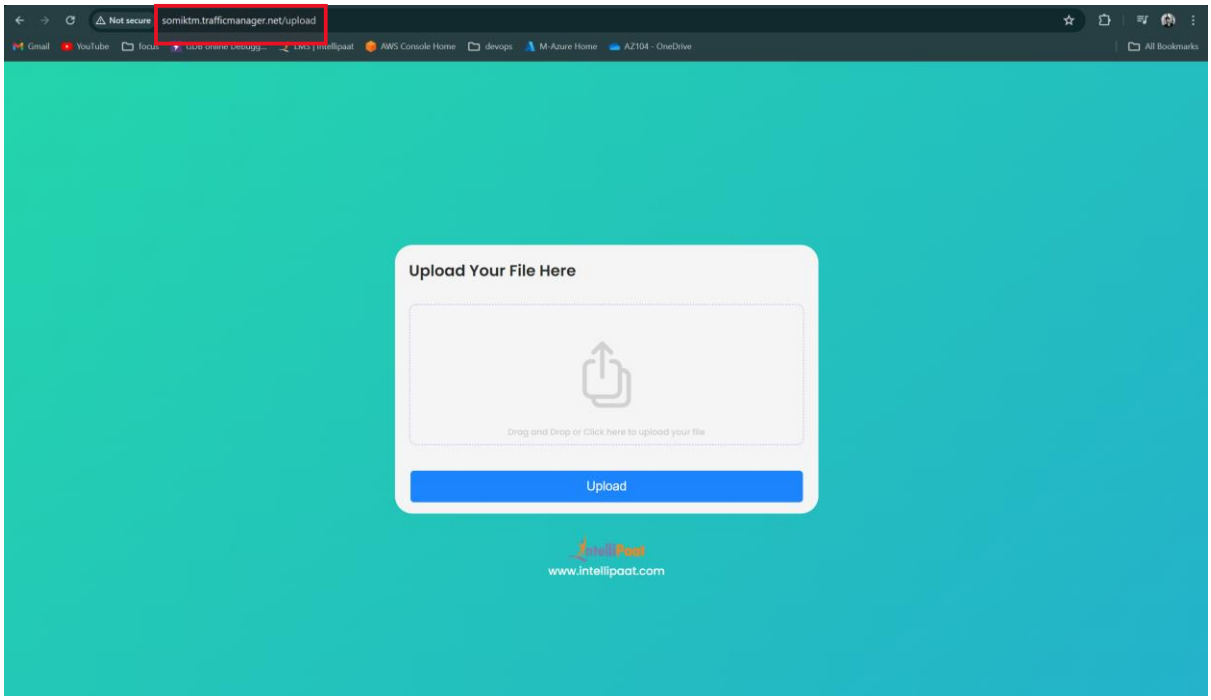




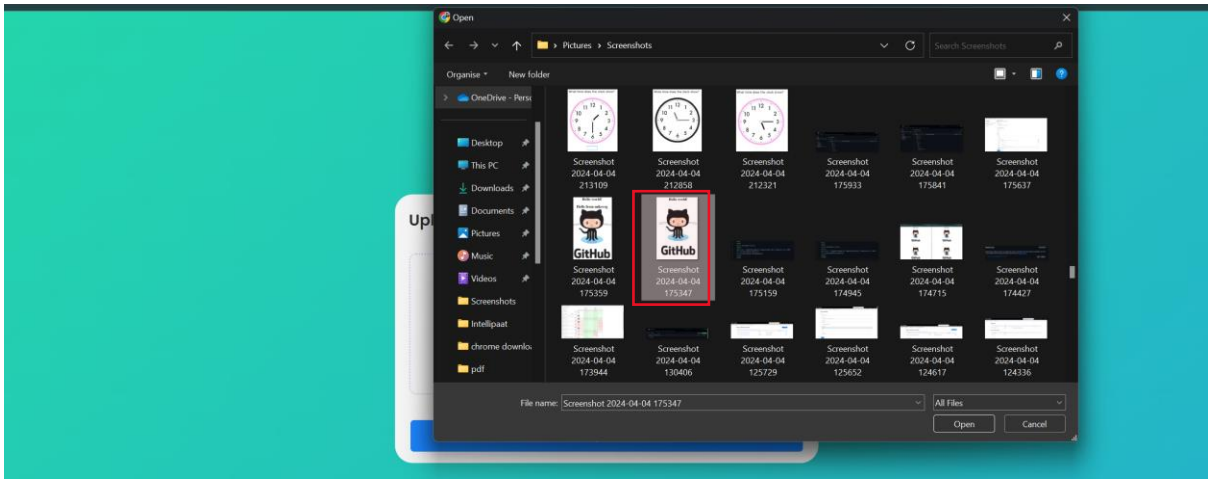
Now copy the DNS name and hit on the browser.

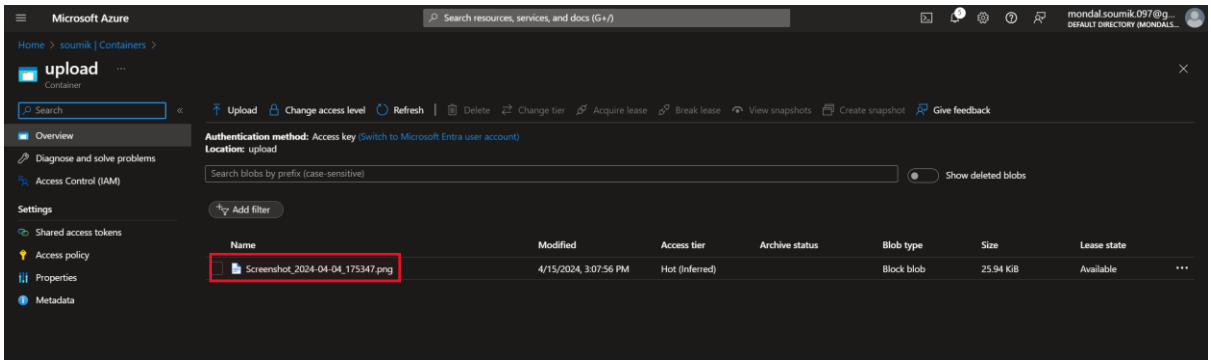


This is the home page we get



<DNSname>/upload page – To upload anything and the uploaded contents will go in the storage account -> containers





Successfully uploaded and the content stored into the container

----------**end of the project**----------