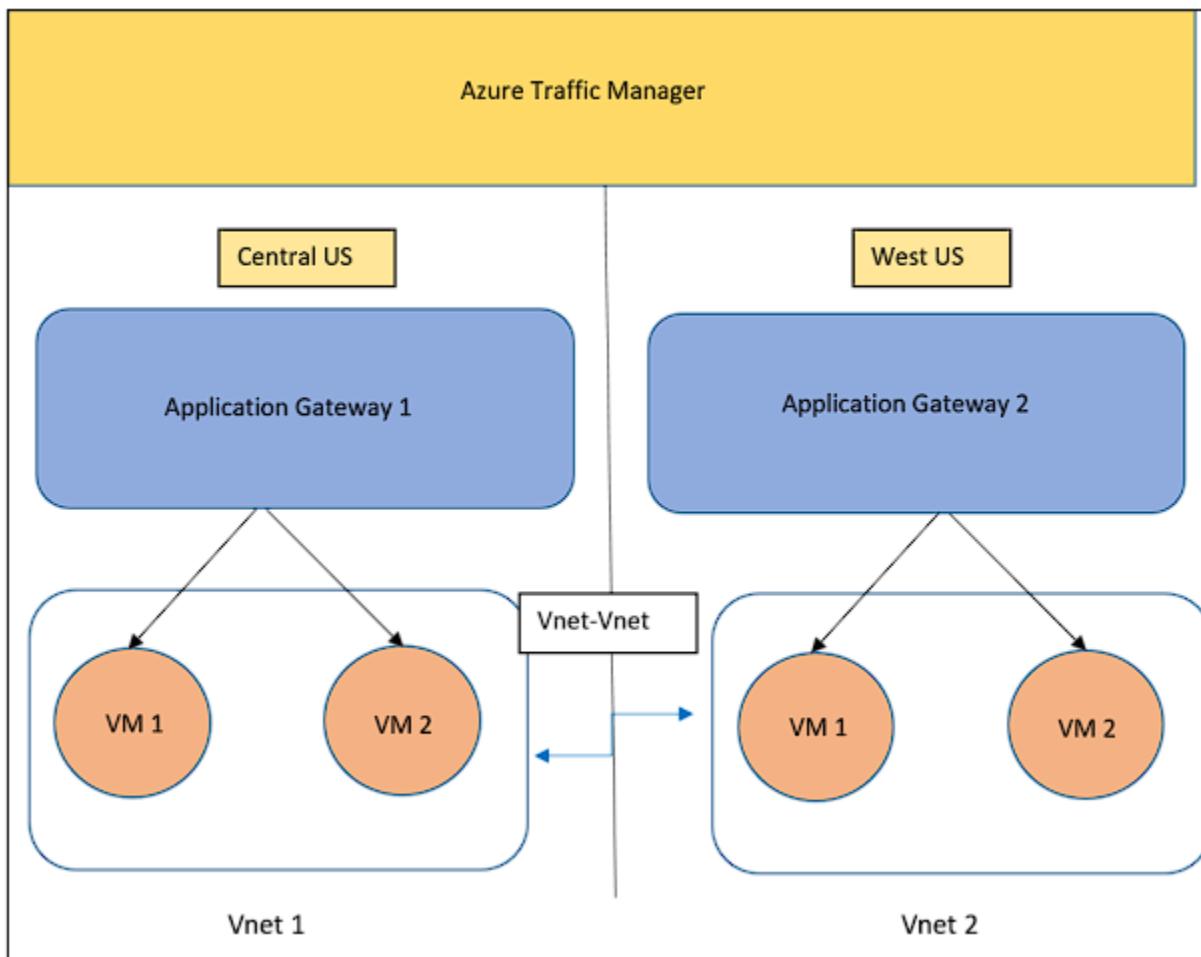


AZURE PROJECT

You work as an Azure professional for XYZ Corp. 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 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 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 the regions should have VMs inside VNets.
2. Clone the github repo <https://github.com/hshar94/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 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 the 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.

Let us create a VM first. While creating VM II us create a new resource group for this project as well. Select on create new under resource group.

The screenshot shows the 'Create a virtual machine' wizard in Microsoft Azure. The 'Basics' tab is selected. In the 'Project details' section, the 'Subscription' dropdown is set to 'Free Trial'. The 'Resource group' dropdown is set to '(New) Resource group' with 'Create new' selected. The 'Virtual machine name' field contains 'VM1'. The 'Region' dropdown is set to '(US) East US'. Under 'Availability options', the 'Availability zone' dropdown is set to 'Zones 1' with a note: 'You can now select multiple zones. Selecting multiple zones will create one VM per zone.' The 'Security type' dropdown is set to 'Trusted launch virtual machines'. The 'Image' dropdown is set to 'Ubuntu Server 18.04 LTS - x64 Gen2 (free services eligible)'. The 'VM architecture' dropdown is set to 'x64'. At the bottom, there are 'Review + create' and 'Next : Disks >' buttons.

Name your RG and VM. While selecting image for VM go with Ubuntu 18.04 because the scripts for the project are supported for this version only.

This screenshot shows the 'Create a virtual machine' wizard in Microsoft Azure, identical to the previous one but with a different resource group name. The 'Resource group' dropdown is set to '(New) ForProject' with 'Create new' selected. All other settings (Virtual machine name, Region, Availability options, Security type, Image, and VM architecture) are the same as the first screenshot. The 'Review + create' and 'Next : Disks >' buttons are also present at the bottom.

Now select password as authentication method and provide credentials to be set. Also allow all http(80), https(443) and ssh(22) ports.

Now simultaneously let us do the networking part aswell. Let us create a VN and subnets needed for this project. Goto networking section from options on left above from your vm creation screen and click on create new option under Virtual Network.

Here we will name our Virtual Network and create a subnet for Application Gateway1.

The screenshot shows the Azure portal interface for creating a virtual machine. On the left, the 'Create a virtual machine' wizard is open, with the 'Networking' tab selected. Under 'Virtual network', the dropdown shows '(new) forProject-1-vnet'. Under 'Subnet', it shows '(new) default (10.9.0.0/24)'. Under 'Public IP', it shows '(new) forProject-1-ip'. In the 'Address space' section of the adjacent 'Create virtual network' dialog, a new subnet 'subnet-for-AG1' is being created with the address range '10.9.1.0/24'. The 'OK' button is highlighted at the bottom right of the dialog.

Then change subnet in network interface to recently created subnet and then review and create.

The screenshot shows the 'Create a virtual machine' wizard with the 'Review + create' step selected. The 'Networking' section shows the subnet has been changed to '(new) forProject-AG1 (10.9.1.0/24)'. The 'Validation passed' message is displayed at the top. The 'Create' button is highlighted at the bottom left.

Now let us create our next VM. Follow the same steps. WE will use the same resource group that we created previously for this whole project.
We will use same Region, VN and subnet for this VM.

The screenshot shows the Microsoft Azure portal's Virtual Machines list. It displays two entries: 'vm1' and 'vm2'. Both are of type 'Virtual machine' under 'Subscription equals all' and 'Resource group equals all'. They are located in 'East US' and have an 'Operating system' of 'Linux'. The size is 'Standard_B1s' and they have one public IP address. The status is 'Running'.

Name	Type	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disk
vm1	Virtual machine	Free Trial	for-project	East US	Running	Linux	Standard_B1s	20.23.76.148	1
vm2	Virtual machine	Free Trial	FOR_PROJECT	East US	Running	Linux	Standard_B1s	40.114.4.201	1

Now let us create another 2 VM's in another region i.e west US as mentioned.

The screenshot shows the 'Create a virtual machine' wizard in the 'Basics' step. The user has selected 'Subscription' as 'Free Trial' and 'Resource group' as 'for-project'. Under 'Instance details', the 'Virtual machine name' is 'vm3', 'Region' is '(US) West US 3', and 'Image' is 'Ubuntu Server 18.04 LTS - x64 Gen2 (free services eligible)'. The 'VM architecture' is set to 'x64'. The 'Run with Azure Spot discount' checkbox is unchecked. At the bottom, there are 'Review + create' and 'Next : Disks >' buttons.

Now provide password as authentication method and set credentials to login.

The screenshot shows the 'Create a virtual machine' wizard in the 'Configuration' step. The user has selected 'Size' as 'Standard_B1s - 1 vcpu, 1 GiB memory (€596.23/month) (free services eligible...)'. Under 'Administrator account', 'Authentication type' is set to 'Password' with 'MyUser' as the 'Username' and '*****' as both 'Password' and 'Confirm password'. In the 'Inbound port rules' section, 'Public inbound ports' is set to 'Allow selected ports' and 'Select inbound ports' is set to 'HTTP (80), HTTPS (443), SSH (22)'. A warning message at the bottom states: 'This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.'

Now get to networking section and create new Virtual Network and new subnet for AG2 like before.

Now assign the new subnet and review and create.

Let us create VM4 now. Follow the same steps as before. RG, VN, Subnet, region all are same as previous third VM.

4 VM's has been created.

A screenshot of the Microsoft Azure portal showing the 'Virtual machines' list. There are four entries:

Name	Type	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address	Disk
vm1	Virtual machine	Free Trial	for-project	East US	Running	Linux	Standard_B1s	20.25.76.148	1
vm2	Virtual machine	Free Trial	for-project	East US	Running	Linux	Standard_B1s	40.114.4.201	1
vm3	Virtual machine	Free Trial	for-project	West US 3	Running	Linux	Standard_B1s	20.106.94.197	1
vm4	Virtual machine	Free Trial	FOR-PROJECT	West US 3	Creating	Linux	Standard_B1s	20.38.170.72	1

Now we move forward for creating storage account and container.
Go to storage accounts and click on add icon to create new one.

A screenshot of the Microsoft Azure portal showing the 'Storage accounts' list. There is one entry:

Name	Type	Kind	Resource group	Location	Subscription
csg100320027d724f0	Storage account	StorageV2	cloud-shell-storage-centralindia	Central India	Free Trial

Select RG that we created for this project. Name the storage account and let other fields be default. REview and create.

A screenshot of the 'Create a storage account' wizard in the Microsoft Azure portal, currently on the 'Basics' step.

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription: Free Trial
Resource group: for-project

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name: forproject1
Region: (US) East US
Performance: Standard: Recommended for most scenarios (general-purpose v2 account)
Redundancy: Geo-redundant storage (GRS)
Make read access to data available in the event of regional unavailability.

Review < Previous Next : Advanced > Give feedback

It is created.

A screenshot of the 'Your deployment is complete' summary page in the Microsoft Azure portal.

Deployment status: Your deployment is complete

Deployment name: forproject1_169120665964
Subscription: Free Trial
Resource group: for-project

Start time: 4/11/2023, 3:21:03 PM
Correlation ID: 3265db98-950d-4e9b-922a-74d94e56cd52

Deployment details:

Next steps:

Go to resource | Give feedback | Tell us about your experience with deployment

Cost Management: Get notified to stay within your budget and prevent unexpected charges on your bill. Set up cost alerts >

Microsoft Defender for Cloud: Secure your appz and infrastructure

Next step is to create a container named “upload”. Go to the storage account you created, Under Data Storage subsection click on container and click on add icon to create new.

The screenshot shows the 'Containers' section of the Azure Storage blade for the 'forproject1' storage account. A single container named 'Logs' is listed. The table columns include Name, Last modified, Public access level, and Lease state. The 'Logs' container was last modified on 4/11/2023, 3:21:38 PM, has a Private public access level, and is in an Available lease state. The 'Containers' tab is selected in the left sidebar.

Name it and make sure to make it public.

The screenshot shows the 'New container' dialog box. In the 'Name' field, 'upload' is entered. In the 'Public access level' dropdown, 'Container (anonymous read access for containers and blob...)' is selected. A warning message at the bottom states: 'All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request, but cannot enumerate containers within the storage account.' The 'Advanced' section is collapsed.

**Now under Data Management, find the Static Website option, click on it and enable it.
Let us create an error page first. So name it error.html and save.**

The screenshot shows the 'Static website' configuration dialog. The 'Static website' toggle switch is set to 'Enabled'. The 'Index document name' field contains 'error.html'. The 'Error document path' field is empty. The 'Tables' tab is selected in the left sidebar.

After saving you will get endpoints and \$web option. Click on it.

The screenshot shows the 'Static website' configuration dialog after saving. It displays the primary endpoint as 'https://forproject1.z13.web.core.windows.net/' and the secondary endpoint as 'https://forproject1-secondary.z13.web.core.windows.net/'. The 'Tables' tab is selected in the left sidebar.

**After clicking you'll have upload sign. We need to upload error page here.
To get the error html file which is to be uploaded just get it from github link provided.**

The screenshot shows the Microsoft Azure Storage Blob service interface. On the left, there's a sidebar with options like Overview, Diagnose and solve problems, Access Control (IAM), Settings, Shared access tokens, Access policy, Properties, Metadata, and Editor (preview). The main area shows a table with columns Name, Modified, Access tier, and Archive status. One row is listed: Name is 'error.html', Modified is '4/11/2023, 3:59:35 PM', Access tier is 'Hot (Inferred)', and Archive status is 'No results'. At the top, there are buttons for Upload, Change access level, Refresh, Delete, Change tier, Acquire lease, Break lease, View snapshots, Create snapshot, and a search bar. To the right, a modal window titled 'Upload blob' has a large dashed area for dragging files. Below it are checkboxes for 'Overwrite if files already exist' and 'Advanced' settings. At the bottom are 'Upload' and 'Give feedback' buttons.

Upload the error.html you got from github here.

This screenshot is similar to the previous one but shows the result of a successful upload. The message 'Successfully uploaded 1 blob(s)' is displayed at the top of the 'Upload blob' modal. The rest of the interface is identical to the first screenshot.

Go to storage account and copy the endpoint that we got.

The screenshot shows the Microsoft Azure Storage blade. On the left, there are sections for Tables, Networking, Azure CDN, Access keys, Shared access signature, Encryption, Microsoft Defender for Cloud, Data management, Redundancy, Data protection, and Object replication. The main area is titled 'Static website' and shows 'Enabled'. It lists the 'Primary endpoint' as 'https://forproject1z13.web.core.windows.net/' and the 'Secondary endpoint' as 'https://forproject1secondary.z13.web.core.windows.net/'. There are also fields for 'Index document name' (set to 'error.html') and 'Error document path'. A 'Copy to clipboard' button is located next to the primary endpoint URL.

Paste it in browser.

The screenshot shows a web browser window displaying the static page content. The address bar shows the URL 'https://forproject1z13.web.core.windows.net/'. The page itself displays the text 'Application Gateway errors!' and 'This is a static page'.

Now let us create a Application Gateway.
Search for application gateway click on it and create new.

Name it. Select East US region and choose the VN that we created. We will need another subnet for this so click on subnet config and subnet selection option.

Fill the details and create a new subnet.

The screenshot shows the Azure portal interface for creating a new subnet. On the left, there's a sidebar with various network-related options like Subnets, Bastion, DDoS protection, Firewall, Microsoft Defender for Cloud, Network manager, DNS servers, Peering, Service endpoints, Private endpoints, Properties, Locks, Monitoring, Alerts, Metrics, Diagnostic settings, Logs, and Connection monitor (classic). The main area shows a table of existing subnets:

Name	IPv4	IPv6	Available IPs	Delegated to
default	10.0.0.0/24	-	251	-
subnet2-ag1	10.0.1.0/24	-	249	-

To the right, a modal window titled 'Add subnet' is open, allowing the creation of a new subnet. It includes fields for Name (set to 'forAG1'), Subnet address range (set to '10.0.2.0/24'), and other optional settings like NAT gateway, Network security group, Route table, and Service endpoints.

Select this newly made subnet in Application Gateway creation.

This screenshot shows the 'Configure virtual network' step in the Application Gateway creation wizard. It displays the selected virtual network 'vm1-vnet' and the newly created subnet 'forAG1 (10.0.2.0/24)'. Below these, there are buttons for 'Previous' and 'Next : Frontends >'.

Now click on next.
Here just app a public ip name and go next for backends.

This screenshot shows the 'Frontends' step in the Application Gateway creation wizard. It shows the selected 'Basic' configuration type. A modal window titled 'Add a public IP' is open, prompting for a name ('AGip1'), SKU ('Standard'), and assignment type ('Static'). The 'Availability zone' field is set to 'None'. At the bottom of the modal are 'OK' and 'Cancel' buttons.

This screenshot shows the 'Frontends' step in the Application Gateway creation wizard, continuing from the previous step. The 'Public IP address' dropdown now includes the newly added public IP 'AGip1'. The rest of the interface remains the same, showing the 'Basic' configuration type and the 'OK' and 'Cancel' buttons at the bottom.

Here click on create Backend pools.
Here add both the VM that we created one by one.

The screenshots show the 'Create application gateway' wizard in the 'Backends' step. In the first screenshot, there are no targets. In the second, a target 'vm1155' is added to 'Pool1'. In the third, a target 'vm2389' is added to 'Pool2'.

Click on next and now you will be on configuration page. Click on add icon to create routing rules. Here in forbidden and bad gateway paste the endpoints we got from static website creation page in storage account followed by /error.html.

The screenshot shows the 'Create application gateway' wizard in the 'Configuration' step. It displays the 'Frontends' and 'Routing rules' sections. A 'Routing rules' dialog is open, showing the configuration for a new rule named 'rule1'. The dialog includes fields for Listener name (name1), Frontend IP (Public), Protocol (HTTP), Port (80), and Additional settings (Listener type: Basic, Error page url: https://forproject1.z13.web.core.windows.net/error.html, Bad gateway - 502: https://forproject1.z13.web.core.windows.net/error.html, Forbidden - 403: https://forproject1.z13.web.core.windows.net/error.html).

**Now we have to add Backend Targets.
Name them. In pool select any one.**

The screenshot shows the 'Create application gateway' configuration page in Microsoft Azure. The 'Configuration' tab is selected. On the left, there are sections for 'Frontends' (with a 'Public (new) AG1ip1' entry), 'Routing rules' (with a '+ Add a routing rule' button), and 'Backends'. The main area is titled 'Add a routing rule' and contains fields for 'Rule name' (set to 'rule1'), 'Priority' (set to '1'), 'Listener' (selected 'Backend targets'), 'Target type' (selected 'Backend pool'), 'Backend target' (selected 'Pool2'), and 'Backend settings' (with a note about the value must not be empty). Below this is a 'Path-based routing' section with a table for adding multiple targets. At the bottom are 'Previous' and 'Next : Tags >' buttons, and 'Add' and 'Cancel' buttons.

**In backend settings create new name it default and add to routing rule page.
Next on the same page you will get option “Add multiple targets to create a path-based rule” click on it.**

The screenshot shows the 'Create application gateway' configuration page in Microsoft Azure. The 'Configuration' tab is selected. On the left, there are sections for 'Frontends' (with a 'Public (new) AG1ip1' entry), 'Routing rules' (with a '+ Add a routing rule' button), and 'Backends'. The main area is titled 'Add Backend setting' and contains fields for 'Backend settings name' (set to 'default'), 'Backend protocol' (selected 'HTTP'), 'Backend port' (set to '80'), and 'Additional settings' (including 'Cookie-based affinity' and 'Connection draining' options). Below this is a 'Host name' section with a note about overriding host headers. At the bottom are 'Previous' and 'Next : Tags >' buttons, and 'Add' and 'Cancel' buttons.

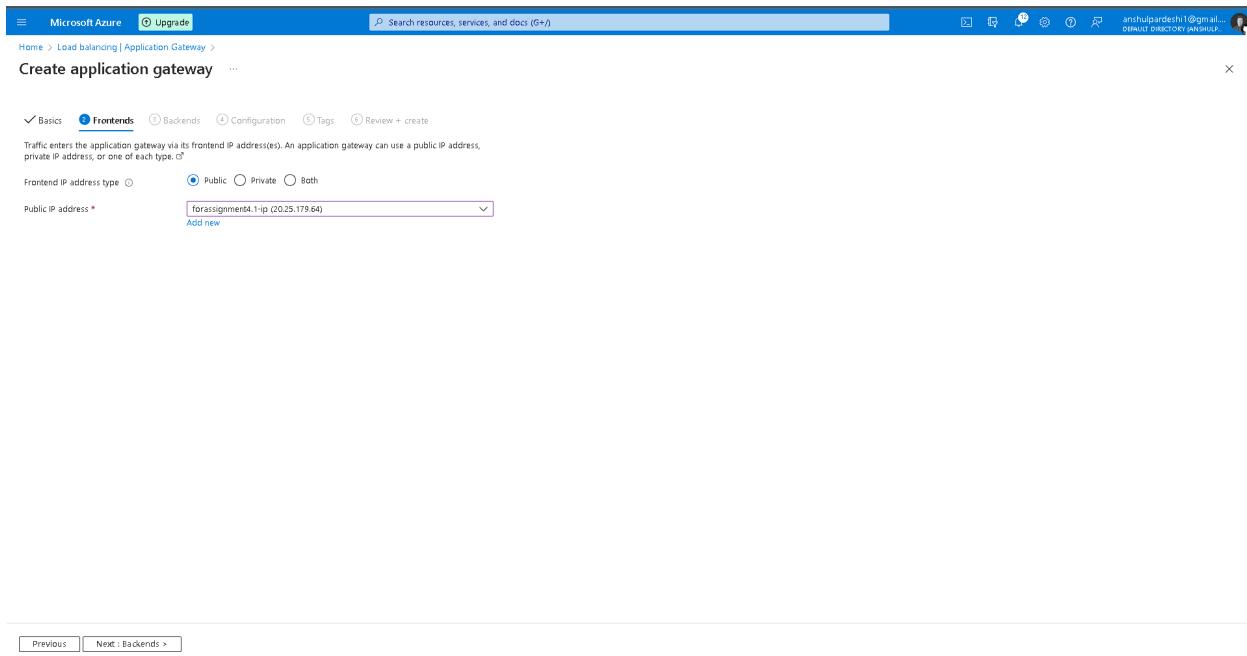
Fill the details as shown and click on add.

Click on next and review and create.

Let us create Application gateway 2.

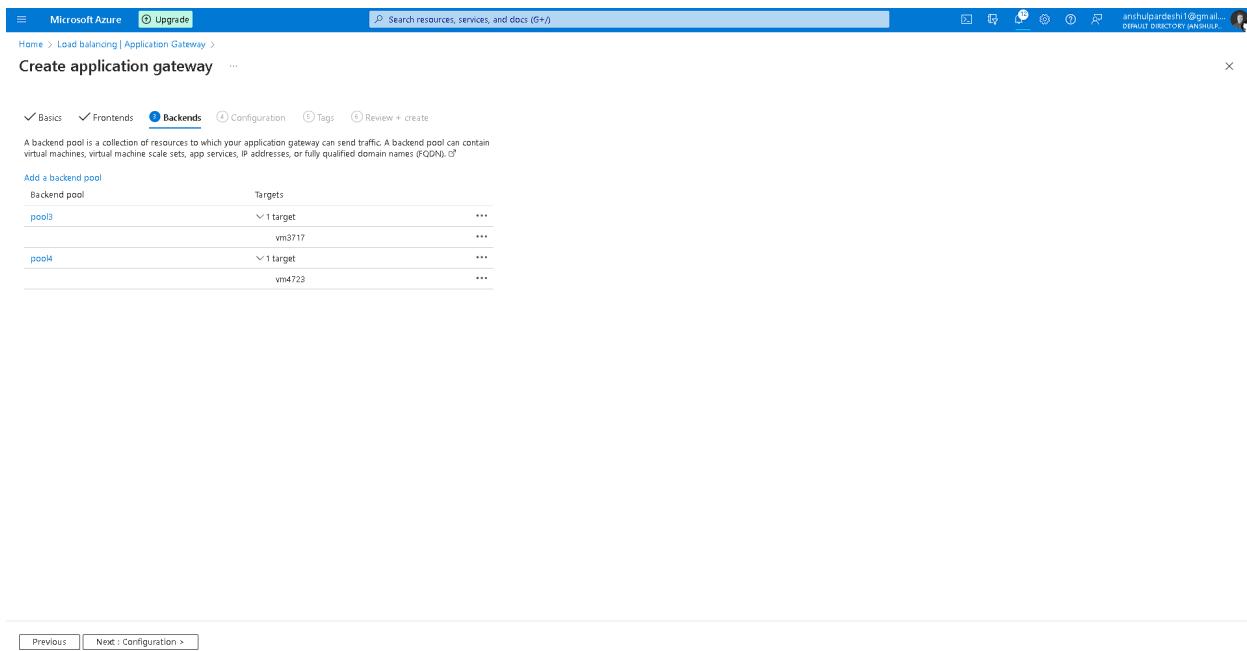
Like before create new subnet for AG2 and select it while creation. Also this time region is us west 3.

Click on next and assign a ip address to it.



The screenshot shows the Microsoft Azure portal interface for creating an Application Gateway. The top navigation bar includes 'Microsoft Azure' and 'Upgrade' buttons, a search bar, and user information. Below the navigation is a breadcrumb trail: 'Home > Load balancing | Application Gateway > Create application gateway'. The main content area has tabs for 'Basics', 'Frontends' (which is selected), 'Backends', 'Configuration', 'Tags', and 'Review + create'. Under 'Frontends', it says '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.' A radio button for 'Public' is selected. A dropdown menu shows 'forassignment64.1-ip (20.25.179.64)' with a checkmark. Buttons for 'Add new' and 'Next: Backends >' are at the bottom.

**Click on next and now let us add backend pools.
Like before create 2 pools and add each VM to each pool.**



The screenshot shows the Microsoft Azure portal interface for creating an Application Gateway. The top navigation bar includes 'Microsoft Azure' and 'Upgrade' buttons, a search bar, and user information. Below the navigation is a breadcrumb trail: 'Home > Load balancing | Application Gateway > Create application gateway'. The main content area has tabs for 'Basics', 'Frontends', 'Backends' (selected), 'Configuration', 'Tags', and 'Review + create'. Under 'Backends', it says '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).'. A link 'Add a backend pool' is present. A table titled 'Backend pool' lists two entries: 'pool3' and 'pool4'. Each entry has a 'Targets' column with a dropdown menu showing 'vm3711' and 'vm4723' respectively. Buttons for 'Previous' and 'Next: Configuration >' are at the bottom.

**Click on next and add routing rules.
We give same badgateway and forbidden url as endpoints followed by error.html**

Microsoft Azure Upgrade 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 you haven't already, or edit previ...

Frontends

+ Add a frontend IP Public for assignment4.1-ip (20.25.179.64) ...

Routing rules

+ Add a routing rule

Rule name * rule2

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.^(f)

Listener name * name2

Protocol * Public

Frontend IP * Protocol HTTP HTTPS

Port * 80

Additional settings

Listener type Basic Multi site

Error page url Yes No

Bad gateway - 502 http://forproject1213web.core.windows.net/error.html

Forbidden - 403 https://forproject1213web.core.windows.net/error.html

Previous Next : Tags > Add Cancel

Let us give backend targets. Like before.

Then click on add multiple targets to create path based rules.

Click on add then review and create.

Microsoft Azure Upgrade 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 you haven't already, or edit previ...

Frontends

+ Add a frontend IP Public for assignment4.1-ip (20.25.179.64) ...

Routing rules

+ Add a routing rule

Rule name * rule2

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.^(f)

Target type Backend pool Redirection

Backend target * pool4

Add new default

Backend settings * default

Add new

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.^(f)

Path	Target name	Backend setting name	Backend pool
/upload	upload	default	pool3

Add multiple targets to create a path-based rule

Previous Next : Tags > Add Cancel

Now let us ssh into VM2 using putty.
Copy the public ip of VM2 and paste it in PUTTY.

The screenshot shows the Microsoft Azure portal interface. On the left, the 'Overview' section for 'vm2' is displayed, including details like Resource group (move), Status (Running), Location (East US), Subscription ID, and Tags. In the center, a 'Putty Configuration' window is open, showing session settings for connecting to '40.114.4.201' on port 22 via SSH. The right side of the screen shows the VM2 details again, including its IP address (40.114.4.201) and network information.

Provide the credentials and login.

```

MyUser@vm2: ~
Login as: MyUser
MyUser@40.114.4.201's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1105-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Maintenance: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Apr 11 12:06:03 UTC 2023

System load: 0.0      Processes:          98
Usage of /: 4.8% of 28.89GB Users logged in: 0
Memory usage: 24%    IP address for eth0: 10.0.1.5
Swap usage: 0K

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/suse/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro static

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as an administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

MyUser@vm2: ~

```

Update it using: sudo apt-get update

```

MyUser@vm2: ~
Reading package lists... Done
MyUser@vm2: ~

```

The terminal session shows the command 'sudo apt-get update' being run, followed by a large amount of output listing various packages and their download progress from the Ubuntu archive.

We need to copy git link and clone to VM so get to the github repo and copy the link.

The screenshot shows a GitHub repository page for 'hsharp/app-project'. The 'Code' tab is selected, showing the repository's structure with files like 'templates', 'README.md', and 'app.py'. Below the code area, the 'Clone' button is highlighted, and the URL 'https://github.com/hsharp/app-project.git' is displayed.

Now use command: git clone <https://github.com/hshar94/azproject.git>

```
MyUser@MyVm2:~$ git clone https://github.com/hshar94/aspxproject.git
Cloning into 'aspxproject'...
remote: Enumerating objects: 147, done.
remote: Counting objects: 100 (39/39), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 147 (delta 121, reused 26 (delta 10), pack-reused 114
Receiving objects: 100% (147/147), 31.26 KiB / 10.42 MiB/s, done.
Resolving deltas: 100% (60/60), done.
MyUser@MyVm2:~$
```

**We need to run the shell script file. So first get to the file location by using:
cd azproject**

**Then use command to run shell script:
./vm2.sh**

Follow the same procedure for VM4.

The screenshot shows the Microsoft Azure portal interface. On the left, a sidebar lists various services: Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Networking, Connect, Disks, Size, Microsoft Defender for Cloud, Advisor recommendations, Extensions + applications, Continuous delivery, Availability + scaling, and Configuration. The main area displays a virtual machine named "FOR-PROJECT".

FOR-PROJECT

Essentials

Resource group (move) : FOR-PROJECT

Status: Running

Location: West US 3

Subscription (move)

Subscription ID: 2038170-73

Tags (edit)

Properties

Computer name: FOR-PROJECT

Health state: Healthy

Operating system: Ubuntu 18.04 LTS

Publisher: Canonical

Offer: Ubuntu Server 18.04 LTS

Plan: Standard

VM generation: Gen 2

VM architecture: x64

PUTTY Configuration

Category: Session

Basic options for your PuTTY session

Specify the destination you want to connect to:

Host Name (or IP address): 20.38.170.73

Port: 22

Connection type: SSH (radio button selected)

Appearance: Load, save or delete a stored session

Behaviour: Saved Sessions

Translation: Default Settings

Keyboard: Load

Mouse: Save

Serial: Delete

DOSBox: Close window on exit: Always (radio button selected)

Connection: Data, SSH, Serial, Telnet, Rlogin, SUPDUP

Networking

Public IP address: 20.38.170.72 (Network interface vm4723)

Public IP address (IPv6): -

Private IP address: 10.1.1.5

Private IP address (IPv6): -

Virtual network/subnet: vm2-vnet/subnet4-AG2

DNS name: Configure

Size

Operating system: Linux (Ubuntu 18.04)

Size: Standard_B2s, 1 GB memory

Public IP address: 20.38.170.72

Virtual network/subnet: vm2-vnet/subnet4-AG2

DNS name: Not configured

JSON View

Now ssh to Vm1.

The screenshot shows the Microsoft Azure portal interface. On the left, a sidebar lists resources: vm1, vm2, vm3, and vm4. Below them are sections for Tags, Diagnose and solve problems, Settings, Networking, Connect, Disks, Size, Microsoft Defender for Cloud, Advisor recommendations, Extensions + applications, Continuous delivery, Availability + scaling, and Configuration. The main area displays 'vm1' details: Status (Standby), Location (West US), Subscription (myVMSubscription), Subscription ID, Tags (edit), Properties, and Virtual machine (Computer name: vm1, Health state: OK, Operating system: Windows Server 2019 Datacenter, Publisher: Microsoft Corporation, Offer: Windows Server 2019 Datacenter, Plan: Standard, VM generation: Gen2). To the right, a 'Putty Configuration' window is open, showing session details: Host Name (or IP address): 20.25.76.148, Port: 22, Connection type: SSH, and Session settings. At the bottom of the Putty window are 'Open' and 'Cancel' buttons. The status bar at the bottom right shows 'Size: Standard Copied' and '1.6 GB memory'. The IP address '20.25.76.148' is highlighted with a red box.

Enter the credentials.

```
[MyUser@ml-azproject ~]
# Enter as: MyUser
# MyUser@ml-azproject:~$ password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1105-azure x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Apr 11 12:43:24 UTC 2023

System load: 0.0          Processes:      96
Usage of /: 4.6% of 28.09GB  Users logged in:      0
Memory usage: 1004MiB   IP address for eth0: 10.0.1.4
Swap usage: 0KiB

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/azure/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See man sudo_root for details.

MyUser@ml-azproject:~$
```

**Update using: sudo apt-get update -y
Then clone repo using: git clone <link>
cd azproject
.vm1.sh**

Also upgrade the pip to avoid further errors using: sudo -H pip3 install --upgrade pip

```
[MyUser@ml-azproject ~]
MyUser@ml-azproject:~$ git clone https://github.com/hshash94/azproject.git
Cloning into 'azproject'...
remote: Enumerating objects: 147, done.
remote: Counting objects: 100, done.
remote: Compressing objects: 100 (23/23), done.
remote: Total 147 (delta 12), reused 28 (delta 10), pack-reused 114
Receiving objects: 100 (147/147), 31.26 KiB | 10.42 MiB/s, done.
Resolving deltas: 100 (100/100), done.
MyUser@ml-azproject:~$ cd azproject
MyUser@ml-azproject:~/azproject$ ./vm1.sh

Reading state information... Done
python3 is already the newest version (3.6.7-1-18.04).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3-ubuntu1.18.04.8).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
DeprecationWarning: Using or importing the '*' wildcard * because normal site-packages is not writeable
Collecting azure-storage[~/azproject]$ ./vm1.sh
  Downloading azure-storage-0.37.0.zip (4.3 kB)
  Preparing metadata (setup.py) ... error
  error: Command errored out with exit status 1:
  command: /usr/bin/python3 -c "import setuptools, tokenize; __file__='/tmp/pip-install-sq034h13/azure-storage_d9d4c1848a42499ca5f509c93eed4ac/setup.py'; exec(compile(open(__file__).read().replace(b'\r\n', b'\n'), __file__, 'exec'))" egg_info --egg-base /tmp/pip-pip-egg-info-4mdns62
      cwd: /tmp/pip-install-sq034h13/azure-storage_d9d4c1848a42499ca5f509c93eed4ac/
  Complete output (19 lines):
  Traceback (most recent call last):
    File "/usr/lib/python3.6/distutils/command/egg_info.py", line 1, in <module>
      from setuptools import setup
    File "/usr/lib/python3.6/distutils/_msvccompiler.py", line 1, in <module>
      raise RuntimeError(_error_message)
RuntimeError:
Starting with v0.37.0, the 'azure-storage' meta-package is deprecated and cannot be installed anymore.
Please install the service specific packages prefixed by 'azure' needed for your application.
The complete list of available packages can be found at:
https://aka.ms/azshy/python/all

Here's a non-exhaustive list of common packages:
- [azure-storage-blob](https://pypi.org/project/azure-storage-blob/) : Blob storage client
- [azure-storage-file-share](https://pypi.org/project/azure-storage-file-share/) : Storage file share client
- [azure-storage-file-datalake](https://pypi.org/project/azure-storage-file-datalake/) : ADLS Gen2 client
- [azure-storage-queue](https://pypi.org/project/azure-storage-queue/) : Queue storage client
  ...
  -----
WARNING: Command errored out with exit status 1:
command: /usr/bin/python3 -c "import setuptools, tokenize; __file__='/tmp/pip-install-sq034h13/azure-storage-0.37.0.zip#sha256=8c7b0e38e7385172013aa39dbc8d661145c029ede6515a467f299e020584bcfa' (from https://pypi.org/simple/azure-storage/)
  command errored out with exit status 1:
  python setup.py egg_info --egg-base /tmp/pip-pip-egg-info-4mdns62
  Complete output (19 lines):
  Traceback (most recent call last):
    File "/usr/lib/python3.6/distutils/command/egg_info.py", line 1, in <module>
      from setuptools import setup
    File "/usr/lib/python3.6/distutils/_msvccompiler.py", line 1, in <module>
      raise RuntimeError(_error_message)
RuntimeError:
Starting with v0.37.0, the 'azure-storage' meta-package is deprecated and cannot be installed anymore.
Please install the service specific packages prefixed by 'azure' needed for your application.
The complete list of available packages can be found at:
https://aka.ms/azshy/python/all

Here's a non-exhaustive list of common packages:
- [azure-storage-blob](https://pypi.org/project/azure-storage-blob/) : Blob storage client
- [azure-storage-file-share](https://pypi.org/project/azure-storage-file-share/) : Storage file share client
- [azure-storage-file-datalake](https://pypi.org/project/azure-storage-file-datalake/) : ADLS Gen2 client
- [azure-storage-queue](https://pypi.org/project/azure-storage-queue/) : Queue storage client
  ...
  -----
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from azure.storage) (2.18.4)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from azure.storage) (2.18.4)
Collecting python-dateutil
  Using cached python-dateutil-2.8.2-py3.py3-none-any.whl (247 kB)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from azure.storage) (2.1.4)
Requirement already satisfied: six<1.5 in /usr/lib/python3/dist-packages (from python-dateutil->azure.storage) (1.11.0)
Installing collected packages: python-dateutil, azure-askpy, azure-common, azure-storage
Successfully installed python-dateutil-2.8.2 cryptography-2.1.4 python-dateutil-2.8.2
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: configparser in /home/MyUser/.local/lib/python3.6/site-packages (5.2.0)
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: idna<2.1,>=2.0 in /usr/lib/python3/dist-packages (from python-dateutil->azure.storage) (2.1.1)
Requirement already satisfied: werkzeug<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from flask) (2.0.3)
Requirement already satisfied: Jinja2<3.0,>=2.10 in /home/MyUser/.local/lib/python3.6/site-packages (from flask) (2.1.3)
Requirement already satisfied: MarkupSafe<2.1,>=2.0.1 in /home/MyUser/.local/lib/python3.6/site-packages (from flask) (2.0.1)
Requirement already satisfied: importlib-metadata in /home/MyUser/.local/lib/python3.6/site-packages (from click<7.1.2->flask) (4.8.3)
Requirement already satisfied: MarkupSafe<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from Jinja2<3.0->flask) (2.0.1)
Requirement already satisfied: databases in /home/MyUser/.local/lib/python3.6/site-packages (from Werkzeug<2.0->flask)
Requirement already satisfied: importlib-metadata in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata<7.1.3->flask) (3.6.0)
Requirement already satisfied: typing-extensions<3.2.4 in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata<7.1.3->flask) (4.1.1)
MyUser@ml-azproject:~$
```

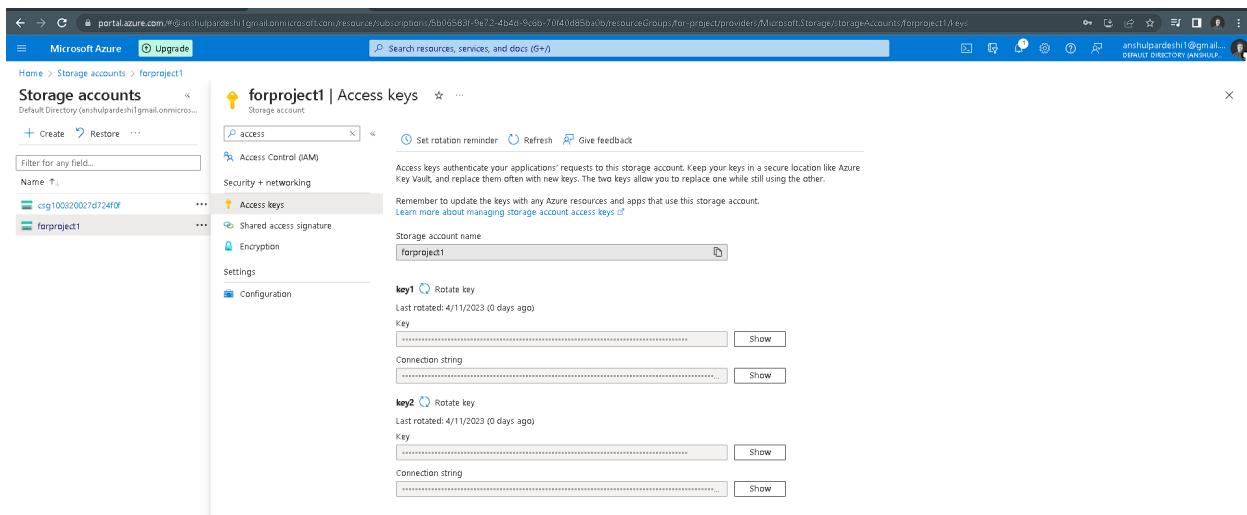
**Now we need to change the config file:
sudo nano config.py**



```
[MyUser@vm1:~/azproject]
GNU nano 2.9.3

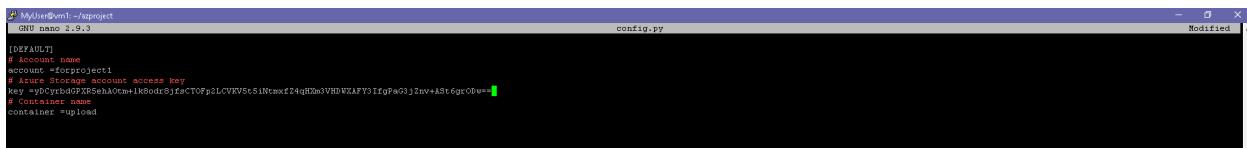
[DEFAULT]
# Account name
account = "storage3121"
# Azure Storage account access key
key = "trXwMuY09Td+4fAc2cooS19C4tfBxmxgeF1oqyaU4slhr?TLTC059cu83aXK0t6yXOTQBnMh2ARvus="
# Container name
container = "upload"
```

To get storage name and access keys, go to your storage account. Under security+networking click on Access keys option.



The screenshot shows the Microsoft Azure Storage accounts interface. A storage account named 'forproject1' is selected. In the 'Access keys' section, two keys are listed: 'key1' and 'key2'. Each key has a 'Show' button next to its corresponding connection string. The connection strings are long, redacted strings starting with 'connectionString='.

Copy and paste the required fields. Then save and exit



```
[MyUser@vm1:~/azproject]
GNU nano 2.9.3

[DEFAULT]
# Account name
account = "forproject1"
# Azure Storage account access key
key = "yDCytg0gXSehA0tm=ik8Sod8jfsCTOpzLCVVKVSt5iNrmxF24qH2e3VHDWXAFT3IfgPaGj3Znv+ASt6grODv=" [REDACTED]
# Container name
container = "upload"
```

And now run: sudo python3 app.py



```
Requirement already satisfied: typing-extensions>=3.9.4 in /home/myuser/.local/lib/python3.8/site-packages (from importlib-metadata>=4.11.1->1.5.1)
MyUser@vm1:~/azproject$ sudo python3 app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://10.0.2.5:49160 (Press CTRL+C to quit)
10.0.2.5 - - [11/Apr/2023 13:17:19] "GET / HTTP/1.1" 200 -
10.0.2.5 - - [11/Apr/2023 13:17:19] "GET / HTTP/1.1" 200 -
10.0.2.7 - - [11/Apr/2023 13:17:48] "GET / HTTP/1.1" 200 -
10.0.2.8 - - [11/Apr/2023 13:17:49] "GET / HTTP/1.1" 200 -
```

Follow exactly the same procedure for VM3.

```

MACHINE:~$ cd /vagrant
MACHINE:~/vagrant$ pip install -r requirements.txt
Hit:1 file:///var/lib/apt/lists/ubuntu.com_ubuntu_bionic-security_InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (9.0.1-2.3-ubuntu1.18.04.8).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
Requirement already satisfied: python3-distutils in /home/MyUser/.local/lib/python3.6/site-packages (0.3.6)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1))
Requirement already satisfied: azure-nspkg<=2.0.0 in /home/MyUser/.local/lib/python3.6/site-packages (from -r requirements.txt (line 1))
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1))
Requirement already satisfied: python-dateutil in /home/MyUser/.local/lib/python3.6/site-packages (from -r requirements.txt (line 1))
Requirement already satisfied: six<1.5 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1))
Requirement already satisfied: werkzeug<2.0 because normal site-packages is not writable
Requirement already satisfied: contextvars in /home/MyUser/.local/lib/python3.6/site-packages (5.2.0)
Defaulting to user installation because normal site-packages is not writable
Requirement already satisfied: flask in /home/MyUser/.local/lib/python3.6/site-packages (2.0.3)
Requirement already satisfied: click<7.1.2 in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: itsdangerous<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: Jinja2<3.0 in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: Werkzeug<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: importlib-metadata in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: click<7.1.2->flask in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: dateutil in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.3)
Requirement already satisfied: typing-extensions<3.6.4 in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata->click<7.1.2->flask) (4.1.1)
Requirement already satisfied: ciso8601<0.5 in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata->click<7.1.2->flask) (0.6.0)
MyUser:~/vagrant$ ./as-project
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
You are generating WSGI server instead.
* Debug mode: off
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on https://10.1.1.4:80/ (Press CTRL+C to quit)
10.1.2.4 = { [11/Apr/2023 13:35:54] "GET / HTTP/1.1" 200 -
10.1.2.5 = { [11/Apr/2023 13:35:54] "GET / HTTP/1.1" 200 -

```

Now let us Create a Traffic Manager. Click on + create icon.

Fill in the details as follow and create.

Once traffic manager is created we need to connect it to application gateway.
Let us add labels to IP of application gateway to avoid further confusion.
Just go to application gateway. Click on frontend public ip

Add a DNS label.

Similarly, follow the same steps for another application gateway.

**Now go to Traffic Manager and Endpoints.
And click on +.**

Name	Status	Monitor status	Type	Location
ep1	Enabled	Checking endpoint	Azure endpoint	East US
ep2	Enabled	Checking endpoint	Azure endpoint	West US 3

Add both the ip as endpoints separately.

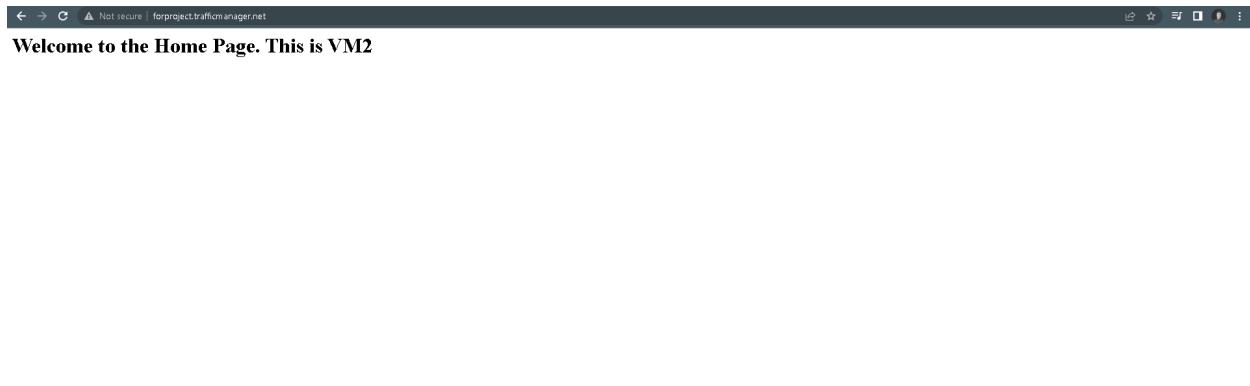
Name	Status	Monitor status	Type	Location
ep1	Enabled	Checking endpoint	Azure endpoint	East US
ep2	Enabled	Checking endpoint	Azure endpoint	West US 3
ep3	Enabled	Checking endpoint	Azure endpoint	West US 3

**Now copy our Traffic Manager DNS name. Paste it and add /upload, /error, as well.
You will see home page upload page and error page.**

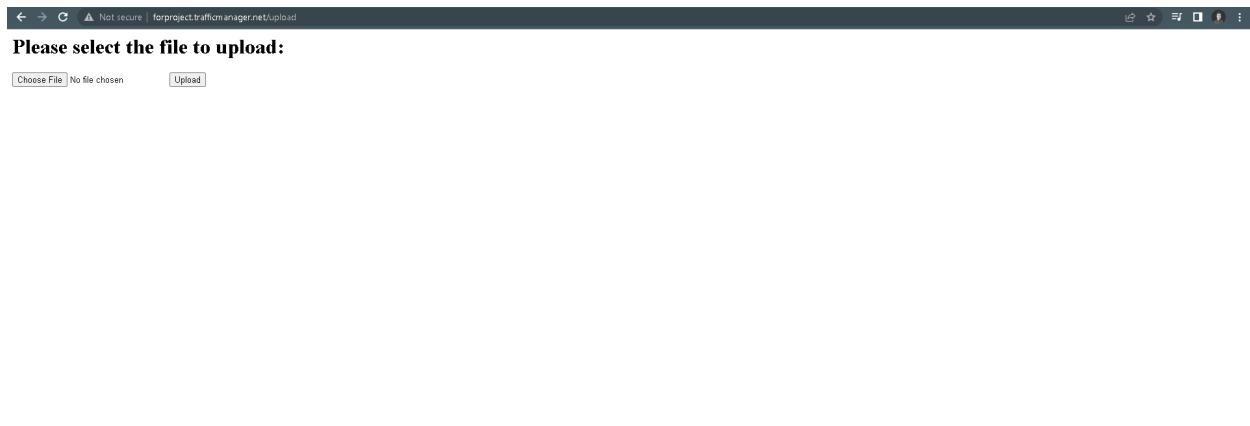
DNS name : http://forproject.trafficmanager.net

Name	Status	Monitor status	Type	Location
ep1	Enabled	Online	Azure endpoint	East US
ep2	Enabled	Online	Azure endpoint	West US 3

Home Page



Upload Page



Error Page

