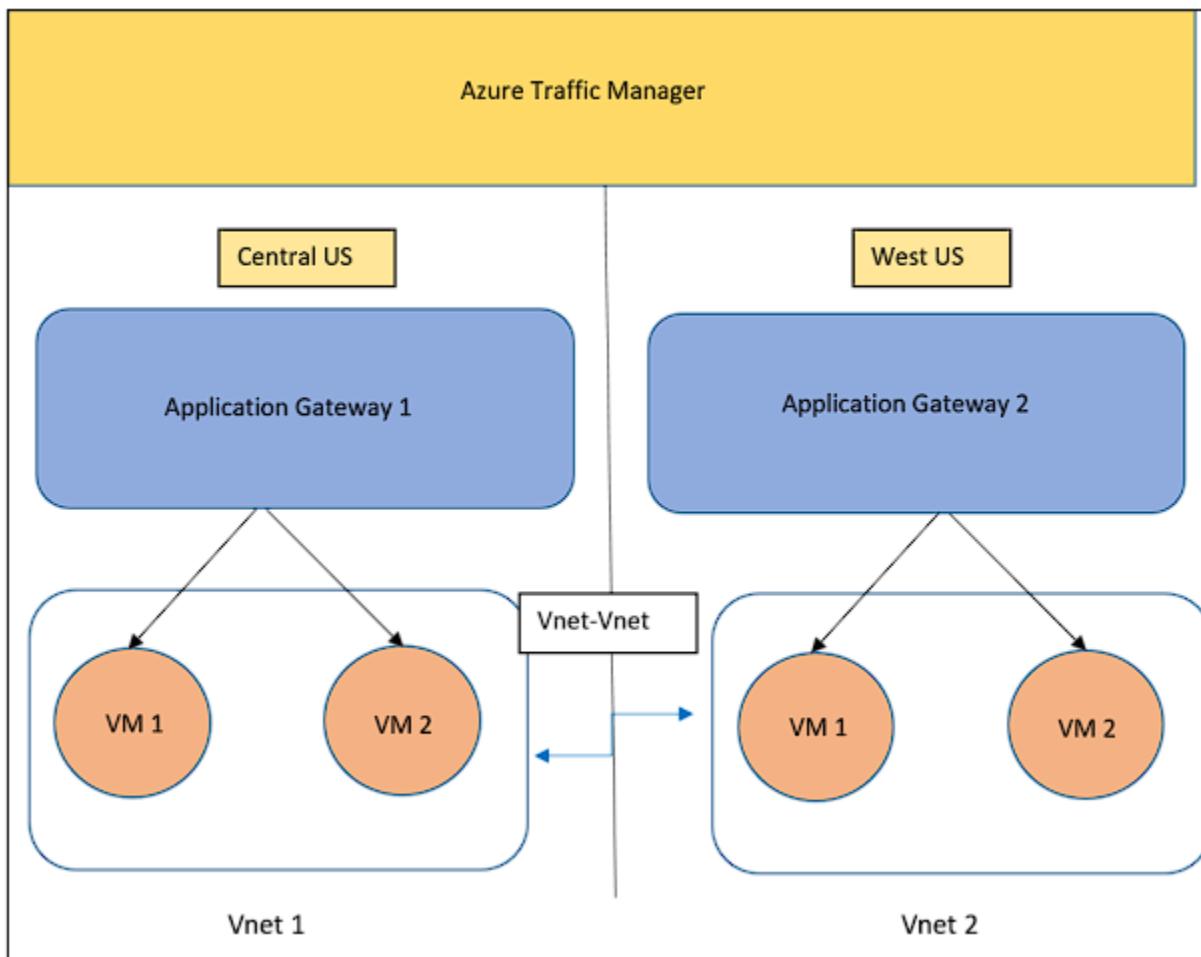


AZURE PROJECT

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' is set to 'Free Trial' and the 'Resource group' is set to '(New) Resource group' with 'Create new' selected. The 'Instance details' section includes fields for 'Virtual machine name' (left blank), 'Region' (set to '(US) East US'), 'Availability options' (set to 'No infrastructure redundancy required'), 'Availability zone' (set to 'Zones 1'), 'Security type' (set to 'Trusted launch virtual machines'), 'Image' (set to 'Ubuntu Server 18.04 LTS - x64 Gen2 (free services eligible)'), and 'VM architecture' (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 now shows '(New) ForProject' with 'Create new' selected. All other settings and configurations remain the same as in the first screenshot.

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 network. On the left, the 'Create a virtual machine' wizard is open, showing the 'Networking' tab. In the 'Virtual network' dropdown, '(new) forProject-1-vnet' is selected. In the 'Subnet' dropdown, '(new) default (10.9.0.0/24)' is selected. Under 'Public IP', '(new) forProject-1-ip' is selected. Under 'NIC network security group', 'Basic' is chosen. Under 'Public inbound ports', 'Allow selected ports' is selected, with 'HTTP (80), HTTPS (443), SSH (22)' listed. A warning message 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.' Below these settings, there are checkboxes for 'Delete public IP and NIC when VM is deleted' and 'Enable accelerated networking'. The 'Load balancing' section is visible at the bottom. On the right, a separate window titled 'Create virtual network' shows the configuration of the subnet 'forProject-1-vnet'. It lists an 'Address space' with '10.9.0.0/16' and an 'Overlap' section. A 'Subnets' table shows two entries: 'subnet1' with address range '10.9.0.0/24' and 'Addresses' '10.9.0 - 10.9.0.255 (256 addresses)', and 'subnet-for-AG1' with address range '10.9.1.0/24' and 'Addresses' '10.9.1.0 - 10.9.1.255 (256 addresses)'.

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 'Networking' tab selected. The 'Virtual network' dropdown shows '(new) forProject-1-vnet'. The 'Subnet' dropdown now shows '(new) forProject-AG1 (10.9.1.0/24)', indicating the recent change. The 'Public IP' dropdown shows '(new) forProject-1-ip'. The 'Networking' section includes fields for 'Public inbound ports' (SSH, HTTP, HTTPS) and 'Azure Spot' (No). The 'Management' section includes fields for 'Microsoft Defender for Cloud' (None), 'System assigned managed identity' (Off), 'Login with Azure AD' (Off), 'Auto-shutdown' (Off), 'Enable hotpatch' (Off), and 'Patch orchestration options' (Image Default). The 'Monitoring' section includes a 'Alerts' field set to 'Off'. At the bottom, there are 'Create', 'Previous', 'Next: Management >', and 'Download a template for automation' buttons. A 'Validation passed' message is displayed at the top of the main form area.

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' for both.

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 step 1: Basics. It asks for basic configuration details like instance name, region, and image. The instance name is 'vm3', region is '(US) West US 3', and the image is 'Ubuntu Server 18.04 LTS - x64 Gen2 (free services eligible)'. Other settings include VM architecture (x64 selected), and no spot discount.

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

The screenshot shows the 'Create a virtual machine' wizard in step 2: Set administrator account. It specifies the administrator account type as 'Password', with 'MyUser' as the username and a complex password. It also configures inbound port rules to allow selected ports (HTTP, HTTPS, SSH).

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

The Microsoft Azure Virtual Network service enables Azure resources to securely communicate with each other in a virtual network which is a logical isolation of the Azure cloud dedicated to your subscription. You can connect virtual networks to other virtual networks, or your on-premises network. [Learn more ↗](#)

Name: vm2-vnet

Address space

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

Address range	Addresses	Overlap
10.1.0.0/16	10.1.0.0 - 10.1.255.255 (65536 addresses)	None
	(0 Addresses)	None

Subnets

The subnet's address range in CIDR notation. It must be contained by the address space of the virtual network.

Subnet name	Address range	Addresses
subnet3	10.1.0.0/24	10.1.0.0 - 10.1.0.255 (256 addresses)
subnet4-AG2	10.1.1.0/24	10.1.1.0 - 10.1.1.255 (256 addresses)
	(0 Addresses)	

Networking interface

When creating a virtual machine, a network interface will be created for you.

Virtual network: (new) vm2-vnet
Subnet: (new) subnet4-AG2 (10.1.1.0/24)
Public IP: (new) vm2-ip

NIC network security group: Basic
Public inbound ports: Allow selected ports
Select inbound ports: HTTP (80), HTTPS (443), SSH (22)

Warning: This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced control in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Delete public IP and NIC when VM is deleted:
Enable accelerated networking: The selected VM size does not support accelerated networking.

Load balancing

Review + create < Previous Next : Management > OK Discard

Now assign the new subnet and review and create.

Validation passed

Information about the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Free Trial
Resource group	for-project
Virtual machine name	vm3
Region	West US 3
Availability options	No infrastructure redundancy required
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	Yes
Image	Ubuntu Server 18.04 LTS - Gen2
VM architecture	x64
Size	Standard B1s (1 vcpu, 1 GiB memory)
Authentication type	Password
Username	MyUser
Public inbound ports	SSH, HTTP, HTTPS
Azure Spot	No

Disk

OS disk size	Default size (30 GiB)
OS disk type	Premium SSD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No

Networking

Subnet: subnet4-AG2

Create < Previous Next > Download a template for automation Give feedback

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	20.25.6.148	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
csg10032007d7240f	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 deployment 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:

- Geo-redundant storage (GRS)
- Make read access to data available in the event of regional unavailability.

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 Microsoft Azure Storage Container list for the 'forproject1' storage account. The 'Containers' section is selected in the left sidebar. A single container named 'Logs' is listed in the main pane. The container was last modified on 4/11/2023, 3:21:38 PM, has a private public access level, and is available. The 'Name' column header is highlighted.

Name it and make sure to make it public.

The screenshot shows the 'New container' dialog in the Microsoft Azure Storage Container list. A new container named 'upload' is being created. The 'Public access level' dropdown is set to 'Container (anonymous read access for containers and blob...)'. 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 in the Microsoft Azure Storage Data management section. The 'Static website' toggle is set to 'Enabled'. The 'Index document name' field contains 'error.html'. The 'Error document path' field is empty. The 'Advanced' section is collapsed.

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

The screenshot shows the 'Static website' configuration dialog again. 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 'Advanced' section is collapsed.

**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'. To the right, a large window titled 'Upload blob' has a central area for dragging files or browsing for files. Below it is a checkbox for overwriting existing files and an 'Advanced' section. At the bottom are 'Upload' and 'Give feedback' buttons.

Upload the error.html you got from github here.

This screenshot shows the same Azure Storage interface after the upload. The 'error.html' file is now listed in the table under the 'Name' column. The 'Modified' column shows the same timestamp as before. The 'Access tier' and 'Archive status' columns remain the same. The 'Upload blob' window on the right now displays a success message: 'Successfully uploaded blob(s)' and 'Successfully uploaded 1 blob(s)'.

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

The screenshot shows the Microsoft Azure Storage Accounts settings page. On the left, there's a sidebar with options like 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 shows a table with columns Primary endpoint, Secondary endpoint, Index document name, and Error document path. The Primary endpoint is set to 'https://forproject1z13.web.core.windows.net/'. There are buttons for 'Save' and 'Discard', and a 'Copy to clipboard' button next to the endpoint URL.

Paste it in browser.

The screenshot shows a web browser window with the address bar containing 'forproject1z13.web.core.windows.net'. The page content says 'Application Gateway errors!' and 'This is a static page'. This indicates that the static website has been successfully deployed.

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

No application gateways to display

Azure Application Gateway gives you application-level routing and load balancing services that let you build a scalable and highly-available web front end in Azure. You control the size of the gateway and scale your deployment based on your needs.

[Create application gateway](#)

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.

Basics

Frontends Backends Configuration Tags Review + create

An application gateway is a web traffic load balancer that enables you to manage traffic to your web application. [Learn more about application gateway](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription: Free Trial [Create new](#)

Resource group: [for-project](#) [Create new](#)

Instance details

Application gateway name: ForProject

Region: East US

Tier: Standard V2

Enable autoscaling: Yes No

Minimum instance count: 0

Maximum instance count: 10

Availability zone: None

HTTP2: Disabled Enabled

Configure virtual network

Virtual network: [vm1-vnet](#) [Create new](#)

Subnet: [subnet2-ag1 \(10.0.1.0/24\)](#) [Manage subnet configuration](#)

Previous **Next : Frontends >**

<http://portal.azure.com/#>

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, Firewall, and Network manager. The main area shows a table of existing subnets: 'default' (10.0.0.0/24) and 'subnet2-ag1' (10.0.1.0/24). A modal window titled 'Add subnet' is open, prompting for a name ('forAG1'), subnet address range ('10.0.2.0/24'), and other configuration details like NAT gateway and service endpoints. Buttons for 'Save' and 'Cancel' are at the bottom of the modal.

Select this newly made subnet in Application Gateway creation.

This screenshot shows the 'Configure virtual network' step in the application gateway creation process. It displays the selected virtual network 'vm1-vnet' and the newly created subnet 'forAG1 (10.0.2.0/24)'. Navigation buttons for 'Previous' and 'Next : Frontends >' are visible at the bottom.

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

This screenshot shows the 'Create application gateway' wizard at the 'Frontends' step. It shows the selection of a public IP address named 'AGip1'. The 'Basic' tab is selected, and the 'Frontend IP address type' is set to 'Public'. The 'Public IP address' dropdown shows '(New) AGip1'. Navigation buttons for 'Previous' and 'Next : Backends >' are visible 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, two backend pools are created: 'Pool1' with target 'vm1155' and 'Pool2' with target 'vm2339'. In the third, both pools are listed.

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 'Frontends' and 'Routing rules'. A 'Routing rules' card shows a plus icon to 'Add a routing rule'. An 'Add a routing rule' modal is open, showing a rule for listener 'name1' on port 80, protocol HTTP, with two endpoints: 'https://forproject1.z13.web.core.windows.net/error.html' and 'https://forproject1.z13.web.core.windows.net/error.html'.

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

**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.**

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', 'Upgrade', and a search bar. 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.**

This screenshot continues the 'Create application gateway' wizard. The 'Backends' tab is selected. 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. Below, there's a table titled 'Backend pool' with two entries: 'pool3' and 'pool4'. Each entry has a 'Targets' column with a single item: 'vm3711' under pool3 and 'vm4723' under pool4. Each target row has three dots (...). Navigation buttons '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**

**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.**

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 Status: Running, Location: East US, and Subscription ID: Sb0653cf-9e72-4b4d-9cbb-70f4d85ba0b. A 'Tags' section is also present. On the right, a 'Putty Configuration' window is open, showing session details for '40.114.4.201'. The session type is set to SSH, and the port is 22. The operating system is listed as 'Linux (Ubuntu 18.04)'. Below the session details, there are sections for 'Session', 'Connection', and 'Default Settings'.

Provide the credentials and login.

```

MyUser@vm2: ~
MyUser@40.114.4.201: password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-105~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 /: 4.4% of 28.89GB  Users logged in: 0
Memory usage: 24%      IP address for eth0: 10.0.1.5
Swap usage: 0B

* 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: ~
MyUser@vm2: ~$ sudo apt-get update -y
Hit:1 http://archive.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://archive.archive.ubuntu.com/ubuntu bionic-updates InRelease [89.7 kB]
Get:3 http://archive.archive.ubuntu.com/ubuntu bionic-backports InRelease [89.3 kB]
Get:4 http://archive.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://archive.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [9370 kB]
Get:6 http://archive.archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [405 kB]
Get:7 http://archive.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:8 http://archive.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:9 http://archive.archive.ubuntu.com/ubuntu bionic/main amd64 Packages [294 kB]
Get:10 http://archive.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [295 kB]
Get:11 http://archive.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [1200 kB]
Get:12 http://archive.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [178 kB]
Get:13 http://archive.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [145 kB]
Get:14 http://archive.archive.ubuntu.com/ubuntu bionic-updates/universe i386 Packages [25.6 kB]
Get:15 http://archive.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6088 kB]
Get:16 http://archive.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [14.5 kB]
Get:17 http://archive.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [10.1 kB]
Get:18 http://archive.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [8668 kB]
Get:19 http://archive.archive.ubuntu.com/ubuntu bionic-security/main amd64 Packages [102 kB]
Get:20 http://archive.archive.ubuntu.com/ubuntu bionic-security/main Translation-en [452 kB]
Get:21 http://archive.archive.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [1169 kB]
Get:22 http://archive.archive.ubuntu.com/ubuntu bionic-security/restricted Translation-en [1169 kB]
Get:23 http://archive.archive.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [173 kB]
Get:24 http://archive.archive.ubuntu.com/ubuntu bionic-security/universe Translation-en [302 kB]
Get:25 http://archive.archive.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [19.8 kB]
Get:26 http://archive.archive.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [3926 kB]
Reading package lists... Done
MyUser@vm2: ~
  
```

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 'hsharp4/app-project'. The repository has 0 stars and 1 watching. The 'About' section indicates no description, website, or topics provided. The 'Clone' button is visible, with options for HTTPS, SSH, and GitHub CLI. The HTTPS URL is listed as <https://github.com/hsharp4/app-project.git>.

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 resources: vm1, vm2, vm3, vm4, and a new VM creation option. The main area displays a VM named 'FOR-PROJECT' under the 'Essentials' tab. The VM details show it's running on Linux (Ubuntu 18.04), has a public IP of 20.38.170.72, and is part of the 'vm2-vnet/subnet4-AG2' network. It has 1 GB of memory and is not configured with a DNS name. The 'Networking' tab indicates a private IP of 10.1.1.5 and a virtual network subnet of 'vm2-vnet/subnet4-AG2'. The 'Size' tab shows the current configuration. Below the VM details, a terminal window is open with the command 'git clone https://github.com/bshar94/asproject.git' being run. The terminal output shows the cloning process, including file counts and sizes, and a warning about a missing directory. The bottom navigation bar includes 'Capabilities', 'Recommendations', and 'Tutorials'.

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' with tabs for Status, Subscription (move), Subscription ID, Tags (edit), Properties, and Virtual machine. The Properties tab is selected, showing fields for Computer name, Health state, Operating system, Publisher, Offer, Plan, and VM generation. To the right, a 'Putty Configuration' window is open, showing session details like Host Name (or IP address) as 20.25.76.148 and Port as 22. Connection type is set to SSH. At the bottom of the Putty window are buttons for Open and Cancel. On the far right, a summary table provides details about the VM: Size (Standard D1 v2, 1 vCPU, 1.75 GB memory), Public IP address (20.25.76.148), Virtual network/subnet (vm1-vnet/subnet2-sg1), and DNS name (Not configured). Below this, a 'Networking' section shows public and private IP addresses, and a 'Size' section indicates the VM's configuration.

Enter the credentials.

```
MyUser@MyVm1: ~
$ login as: MyUser
$ MyUser@MyVm1:~$ password
Welcome to Ubuntu 10.04.6 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 28.89GB Users logged in:  0
Memory usage: 22k          IP address for eth0: 10.0.1.4
Swap usage:  0B

 * 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@MyVm1: ~
```

```
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

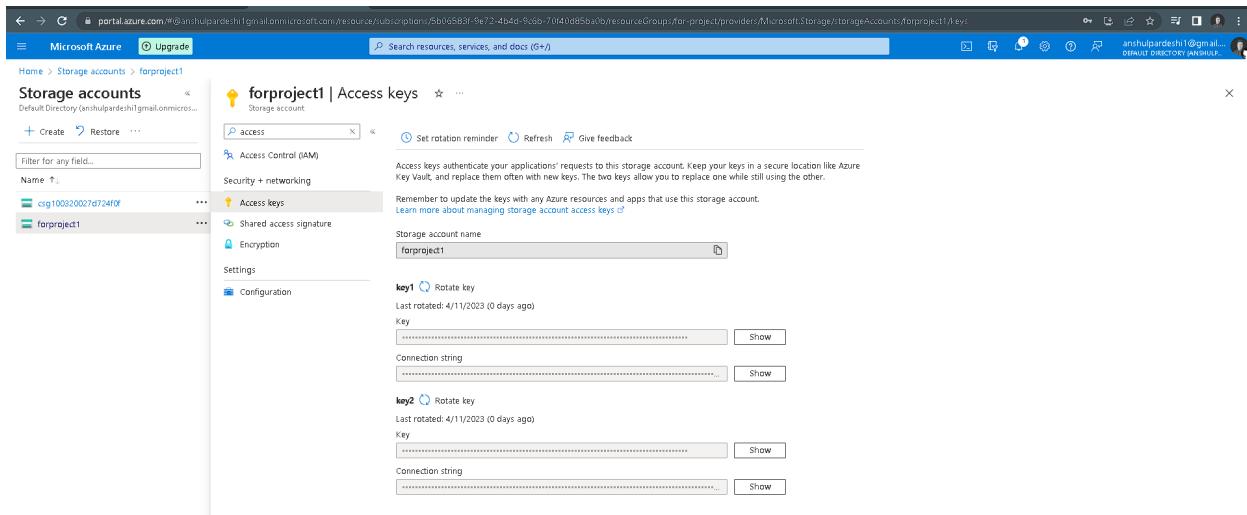
**Now we need to change the config file:
sudo nano config.py**



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

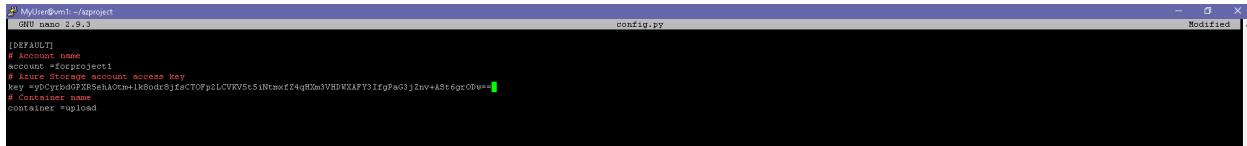
[DEFAULT]
# Account name
account = "storage3121"
# Azure Storage account access key
key = "trXwMuY097d+4fAc2cooS19C4tBxwsgF1oqyU4zlh?TLTC059cu83aXK0t6yXOTQBnMh2ARvus="
# Container name
container = "upload"
```

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



The screenshot shows the Azure portal interface for managing storage accounts. In the left sidebar, under 'Storage accounts', 'forproject1' is selected. On the right, the 'Access keys' blade is open for this account. It displays two sets of keys: 'key1' and 'key2'. Each key includes a 'Rotate key' button, a 'Last rotated' timestamp (4/11/2023), and a 'Key' field containing a long hex string. Below each key is a 'Show' button and a 'Connection string' field with a 'Show' button.

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 = "yDcytbgXSehA0tm=ik8Sod8jfsCTOp2LCVVKVSt51NrmxF24qH2e3VHDWXAFT3IfgPaGjJZnv+ASt6grCDv=4"
# 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://10.0.2.5:5000 (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.

```

Hit:1 http://archive.ubuntu.com/ubuntu bionic-security InRelease
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).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
Requirement already satisfied: python3-stdlib in /home/MyUser/.local/lib/python3.6/site-packages (0.36.0)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: azure-nspkg<=2.0.0 in /home/MyUser/.local/lib/python3.6/site-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: python-dateutil in /home/MyUser/.local/lib/python3.6/site-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: six<1.5 in /usr/lib/python3/dist-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: werkzeug<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: click<7.1.2 in /home/MyUser/.local/lib/python3.6/site-packages (from python3-pip==9.0.1-2.3-ubuntu1.18.04)
Requirement already satisfied: itsdangerous<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.1)
Requirement already satisfied: Jinja2<3.0 in /home/MyUser/.local/lib/python3.6/site-packages (from flask==2.0.1)
Requirement already satisfied: Werkzeug<2.0 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) (4.0.3)
Requirement already satisfied: itsdangerous<2.0 in /home/MyUser/.local/lib/python3.6/site-packages (from Jinja2==2.11.2->flask) (2.0.1)
Requirement already satisfied: dateclasses in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata==4.0.3->click==7.1.2->flask) (3.6.1)
Requirement already satisfied: typing-extensions<3.6.4 in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata==4.0.3->click==7.1.2->flask) (3.6.1)
Requirement already satisfied: cipp>0.5 in /home/MyUser/.local/lib/python3.6/site-packages (from importlib-metadata==4.0.3->click==7.1.2->flask) (0.6.0)
MyUser@My-Asus:~/as-project$ sudo python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
You can set the environment variable to "production" for better security.
* Debug mode: off
* Running on all addresses.
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.

Name *	<input type="text" value="ForProject"/>
Routing method	<input type="text" value="Performance"/>
Subscription *	<input type="text" value="Free Trial"/>
Resource group *	<input type="text" value="for-project"/>
Resource group location	<input type="text" value="East US"/>

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

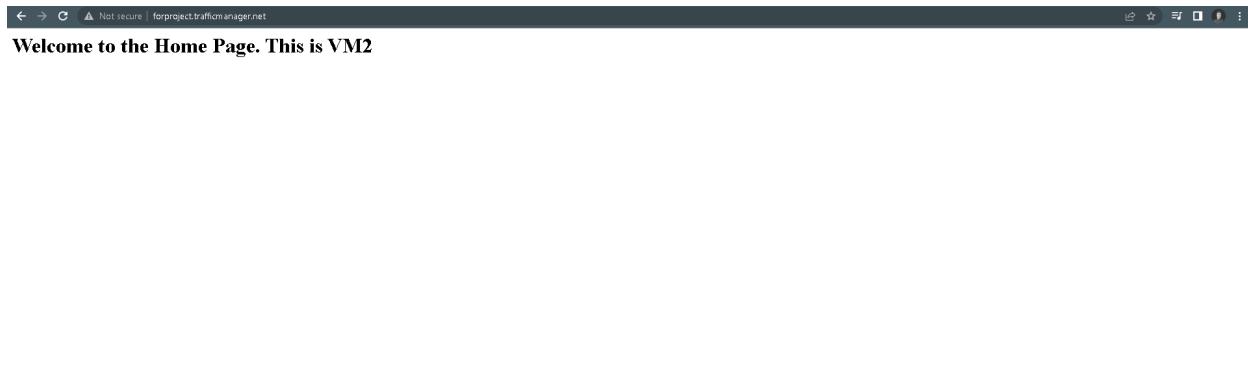
Add both the ip as endpoints separately.

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

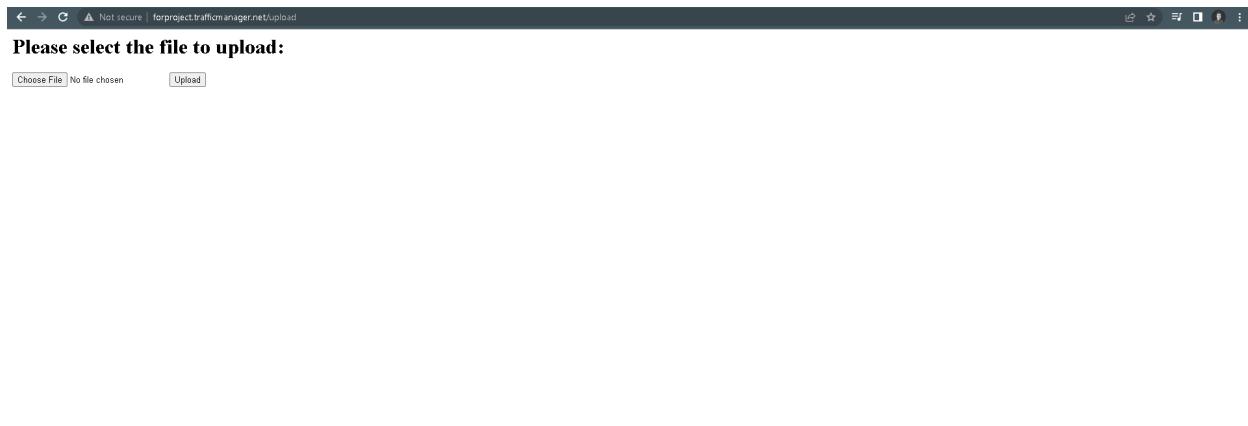
**Now copy our Traffic Manager DNS name. Paste it and add /upload, /error, as well.
You will see home page upload page and error page.**

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

