**Azure Administrator Capstone Project AZ-104**

**Requirement:**

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

A diagram of a computer network

Description automatically generated with medium confidence

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

**Solution:**

1. Create a virtual network in Central US region which will host 2 VMs and an application gateway1.
2. Create 2 subnets in this virtual network, one for deploying VMs and another for application gateway.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create another Virtual network in West US region
2. Create 2 subnets that will have VMs and application gateway2

A screenshot of a computer

Description automatically generated

1. Create 4 virtual machines, 2 in VNET1 and the other 2 in VNET2

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Create VM2 with the same configuration.

A screenshot of a computer

Description automatically generated

1. Create VM3 and VM4 in West US region inside VNET2.

A screenshot of a computer

Description automatically generated

1. Create VM4 with the same configuration.

A screenshot of a computer

Description automatically generated

1. Create a new storage account as below.

A screenshot of a computer

Description automatically generated

1. 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. Download error.html and save it locally.
2. Go to the newly created storage account and select static website and enable it.

A screenshot of a computer

Description automatically generated

1. Go to container $web and upload the error.html file and verify the error page using primary endpoint

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create 2 application gateways for VNet1 and VNet2

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Create another application gateway for VNET2

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Configure DNS name for the application gateways.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create a traffic manager profile to distribute traffic equally

A screenshot of a computer

Description automatically generated

1. Create 2 endpoints for centralUS and WestUS as below.

A screenshot of a computer

Description automatically generated

1. Create another endpoint for westUS

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Launch all 4 Virtual machines and update the machines using sudo apt-get update

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Clone the git repository in all the VMs – git clone <https://github.com/azcloudberg/azproject.git>

A computer screen shot of a black background

Description automatically generated

1. Go inside azproject folder and run ./vm1.sh and ./vm2.sh on all 4 machines respectively

A computer screen shot of a program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

1. Now edit the config.py file and enter details such as account name and key details from the storage account access keys section.

A black screen with a black background

Description automatically generated

1. Do the same on VM3 (this is where upload page is configured)

A black screen with white text

Description automatically generated

1. Create a container named upload on the storage account.

A screenshot of a computer

Description automatically generated

1. Run the following command – sudo python3 app.py on VM1 and VM3

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

1. Go to traffic manager profile and copy DNS name and paste it in browser.

A screenshot of a computer

Description automatically generated

1. Create VNet-to-VNet peering. Go to centralus-vnet and choose peering option

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

A screenshot of a chat

Description automatically generated

1. Verify the peering connection using ping command from VM1 to VM3

A screenshot of a computer screen

Description automatically generateds

1. Copy the DNS name of traffic manager profile followed by /upload and paste in browser.

A screenshot of a computer

Description automatically generated

1. Verify the file uploaded in storage account.

A screenshot of a computer

Description automatically generated

1. Also verified the error page after the application is stopped.

A screenshot of a computer

Description automatically generated