**Implementing Virtual Networking with DNS resolution**

**Project Architecture:**

Diagram

Description automatically generated

**Azure Services Used:**

Resource Groups, Virtual Machines, PowerShell, ARM Templates, Network Security Groups, Virtual Networks, Subnets, Private DNS Zone and DNS Zones.

**Detailed Steps:**

Step 1: Created a Virtual Network (az104-04-vnet1) and added this Virtual Network to the Resource Group (az104-04-rg1). The Virtual Network and the Resource group were deployed in the East US region.

Graphical user interface, text, application, email

Description automatically generated

Step 2: Now, in the IP addresses section of the Virtual Network, I have added the IPv4 type address space and booked an address space of 10.40.0.0/20-10.40.15.255 (4096 addresses).

Graphical user interface, text, application

Description automatically generated

Step 3: Added a new subnet (default) to the VNET – the address space for the default subnet is 10.40.0.0/24.

Graphical user interface, application, Word

Description automatically generated

Step 4: In this step, I have deployed an ARM Template using PowerShell.

(The templates and parameters JSON files are in the GitHub repository.)

Deployed the Template using the following Powershell Commands:

PS /home/harshit> $rgName = 'az104-04-rg1'

PS /home/harshit> New-AzResourceGroupDeployment `

>> -ResourceGroup $rgName `

>> -TemplateFile $HOME/az104-04-vms-loop-template.json `

>> -TemplateParameterFile $HOME/az104-04-vms-loop-parameters.json

In the image below, the Template has been deployed successfully.

Text

Description automatically generated

Step 5: The nic0 of the VM0 had Dynamic IP using DHCP. In this step, I have changed the dynamic assignment to static IP and have assigned a Public IP (Basic SKU and Dynamic Assignment).

Graphical user interface, text, application, email

Description automatically generated

Step 6: As mentioned in the previous step, Assigned the nic0 of the VM0 with Public IP (Basic SKU and Dynamic Assignment) and made the internal IP static. Made Internal IP static so, if the VM restarts or reboots, it does not lose its internal IP.

Graphical user interface, application, Word

Description automatically generated

Step 7: Now, changed the internal IP assignment for nic1 of the VM1 to static and assigned a Public IP (Basic SKU and Dynamic Assignment) to the nic1.

Graphical user interface, application, Word

Description automatically generated

Step 8: In this step, I have tried to login into VM0 through an RDP connection. This failed as the Security Rules block incoming traffic to the VMs by default. So, I have to allow incoming traffic to the VMs, create Network Security Group Rules, and then associate the rules to the Network Interface Cards of the VMs to connect using RDP on port 3389.

Graphical user interface, application

Description automatically generated

Step 9: Created a new NSG rule in the East US region and in a Resource Group (az104-04-rg1) with the following rules to allow any incoming traffic to the VM, using the RDP service which will be on port 3389.

AllowRDPInbound NSG Rule:

Source – Any

Source Port Ranges – Any

Destination – Any

Service – RDP

Destination Port – 3389

Action – Allow

Priority – 300

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application

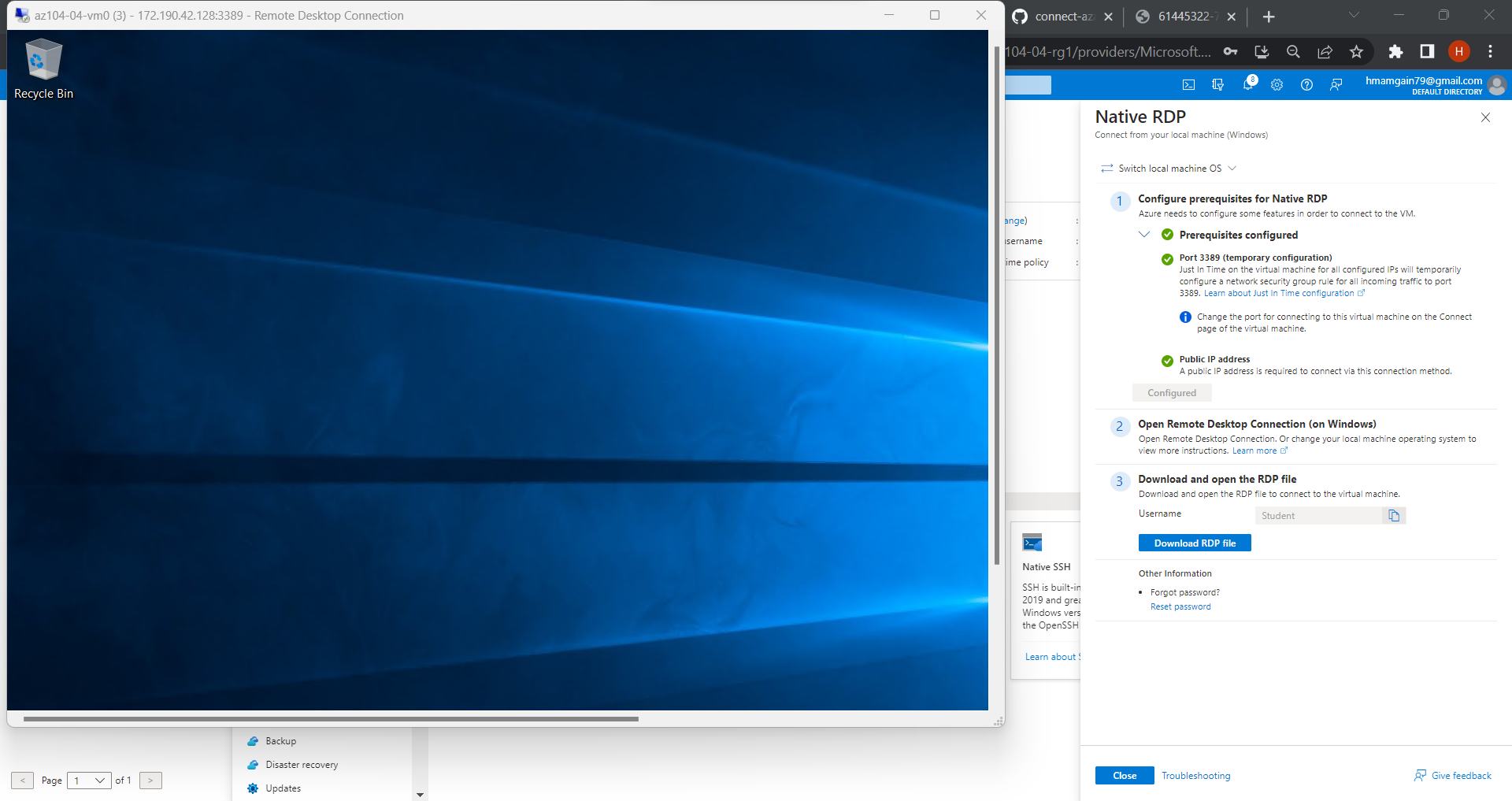
Description automatically generated

Step 10: Now I have associated both the nic with the Security Rule.

Graphical user interface, text, application, email

Description automatically generated

Step 11: After the previous step, I was successfully logged into the VM.



Step 12: Now, the VMs are accessible, and I was able to log in. Here comes the need for DNS resolution as it is very hard for us to remember the IP of all the VMs. In this step, I will configure Private DNS resolution for the internal name resolution. With this, the machines that are connected internally on the Virtual Network (az104-04-vnet1) will be able to communicate with each other.

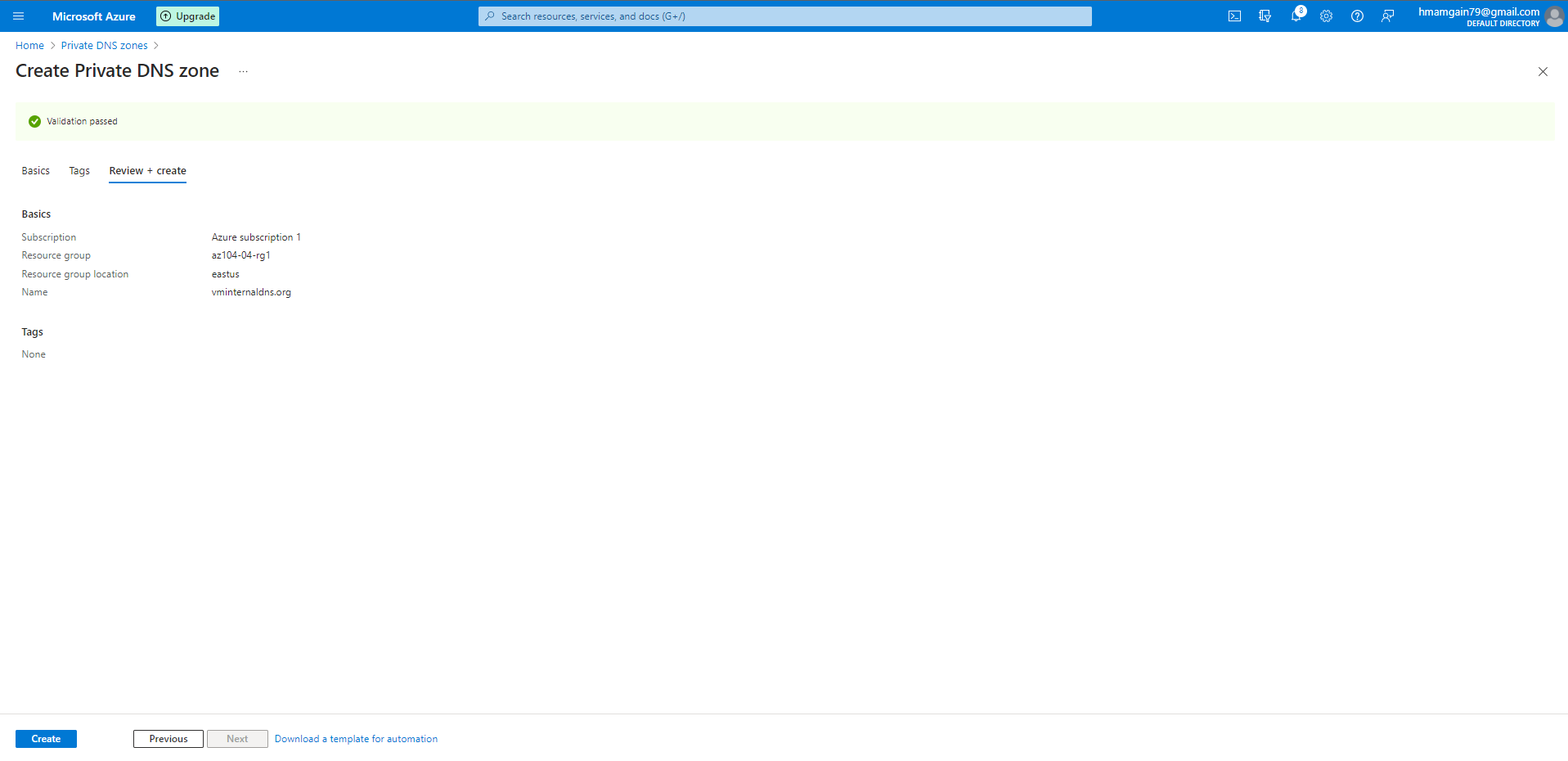
For internal name resolution, I have created a Private DNS Zone. With the following details.

Subscription - Azure subscription 1

Resource group - az104-04-rg1

Resource group location- eastus

Name - vminternaldns.org



Step 13: In this step, I have created a Virtual Network Link. The Virtual network link in a private DNS zone enables resolving domain names from the virtual network (az104-04-vnet1) without the need for public internet connectivity. With this configuration, the virtual machines and other resources within your virtual network can resolve domain names within the private DNS zone.

Graphical user interface, application, Teams

Description automatically generated

Step 14: I have tried to ping VM1 using the ‘nslookup’ command using the name of the VM instead of the IP. Here, I have used the name of the VM and the internal Domain name. (VM\_Name.Domain\_Name)

C:\Users\Student>nslookup az104-04-vm1.vminternaldns.org

Server: UnKnown

Address: 168.63.129.16

Non-authoritative answer:

Name: az104-04-vm1.vminternaldns.org

Address: 10.40.1.4

C:\Users\Student>nslookup az104-04-vm0.vminternaldns.org

Server: UnKnown

Address: 168.63.129.16

Non-authoritative answer:

Name: az104-04-vm0.vminternaldns.org

Address: 10.40.0.4

Text

Description automatically generated

Step 15: The Private DNS Zone will only resolve internally and not over the Internet. For this I have configured a DNS Zone, with this I will be able to resolve the VM Globally over the internet.

Creating a DNS Zone, using this service. With the following details:

Resource group - az104-04-rg1

Resource group location - eastus

Name - vmexternal.org

Graphical user interface, application, Word

Description automatically generated

Step 16: After creating the DNS Zone, I have created an ‘A Record’ set for VM0.

A record set is a collection of DNS records that share the same name and type. Record sets are used to manage DNS records for a domain name or subdomain within a DNS zone.

TTL – ‘ Time to Live’ This determines how long the DNS record will be cached by DNS servers.

Record Type – A

TTL – 1 hour

Public IP - 172.190.42.128

A picture containing graphical user interface

Description automatically generated

Step 17: An ‘A Record’ set for VM1.

Record Type – A

TTL – 1 hour

Public IP - 172.190.43.246

Text

Description automatically generated with low confidence

Step 18: Public DNS records Have been created for the VMs.

Graphical user interface

Description automatically generated

Step 19: I can resolve the VMs over the internet using the following commands on the Cloud Shell, I am able to do this because of the Four Highly available Nameservers that are configured and maintained by Azure.

Nameservers are a critical component of the Domain Name System (DNS) that translate domain names into IP addresses. In simple terms, nameservers are servers that are responsible for storing information about a domain name and its associated IP addresses.

PS /home/harshit> nslookup az104-04-vm1.vmexternal.org ns1-37.azure-dns.com.

Server: ns1-37.azure-dns.com.

Address: 150.171.10.37#53

Name: az104-04-vm1.vmexternal.org

Address: 172.190.43.246

PS /home/harshit> nslookup az104-04-vm0.vmexternal.org ns1-37.azure-dns.com.

Server: ns1-37.azure-dns.com.

Address: 150.171.10.37#53

Name: az104-04-vm0.vmexternal.org

Address: 172.190.42.128

4 Nameservers available \_

ns1-37.azure-dns.com.

ns2-37.azure-dns.net.

ns3-37.azure-dns.org.

ns4-37.azure-dns.info.

Graphical user interface

Description automatically generated