Lab 06 - Implement Traffic Management

Task 1: Provision the lab environment

In this task, we will deploy four virtual machines into the same Azure region. The first two will reside in a hub virtual network, while each of the remaining two will reside in a separate spoke virtual network.

Making sure I've uploaded the files \Allfiles\Labs\06\az104-06-vms-loop-template.json and \Allfiles\Labs\06\az104-06-vms-loop-parameters.json into the Cloud Shell home directory:

```
PS /home/andrijana> ls -l
total 36
-rw-r--r-- 1 andrijana andrijana  447 Mar 23 14:38 az104-02a-customRoleDefinition.json
-rw-r--r-- 1 andrijana andrijana  373 Mar 23 22:38 az104-05-vnetvm-loop-parameters.json
-rw-r--r-- 1 andrijana andrijana 7842 Mar 23 22:37 az104-05-vnetvm-loop-template.json
-rw-r--r-- 1 andrijana andrijana  388 Mar 23 23:27 az104-06-vms-loop-parameters.json
-rw-r--r-- 1 andrijana andrijana 9867 Mar 23 23:27 az104-06-vms-loop-template.json
lrwxrwxrwx 1 andrijana andrijana   22 Mar 23 21:32 clouddrive -> /usr/csuser/clouddrive
drwxr-xr-x 3 andrijana andrijana 4096 Mar 20 22:09 Microsoft
```

Creating the first resource group:

```
PS /home/andrijana>  $location = 'eastus'
PS /home/andrijana>  $rgName = 'az104-06-rg1'
PS /home/andrijana>  New-AzResourceGroup -Name $rgName -Location $location

ResourceGroupName : az104-06-rg1
Location          : eastus
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-06-rg1
```

From the Cloud Shell pane, run the following to create the three virtual networks and four Azure VMs into them by using the template and parameter files you uploaded:

```
New-AzResourceGroupDeployment `
   -ResourceGroupName $rgName `
   -TemplateFile $HOME/az104-06-vms-loop-template.json `
```

```
   -TemplateParameterFile $HOME/az104-06-vms-loop-parameters.json
```

```
PS /home/andrijana> New-AzResourceGroupDeployment `
>>    -ResourceGroupName $rgName `
>>    -TemplateFile $HOME/az104-06-vms-loop-template.json `
>>    -TemplateParameterFile $HOME/az104-06-vms-loop-parameters.json

DeploymentName          : az104-06-vms-loop-template
ResourceGroupName       : az104-06-rg1
ProvisioningState       : Succeeded
Timestamp               : 3/23/2023 11:34:27 PM
Mode                    : Incremental
TemplateLink            :
Parameters              :
                          Name             Type                       Value
                          ===============  =========================  ==========
                          vmSize           String                     "Standard_D2s_v3"
                          vmName           String                     "az104-06-vm"
                          vmCount          Int                        4
                          adminUsername    String                     "Student"
                          adminPassword    SecureString               null

Outputs                 :
DeploymentDebugLogLevel :
```

Terminal container button

Next, we install the Network Watcher extension on the Azure VMs deployed in the previous step by running the following command in PowerShell:

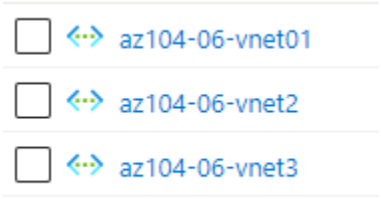
```
PS /home/andrijana> $rgName = 'az104-06-rg1'
PS /home/andrijana> $location = (Get-AzResourceGroup -ResourceGroupName $rgName).location
PS /home/andrijana> $vmNames = (Get-AzVM -ResourceGroupName $rgName).Name
PS /home/andrijana>
PS /home/andrijana> foreach ($vmName in $vmNames) {
>>   Set-AzVMExtension `
>>    -ResourceGroupName $rgName `
>>    -Location $location `
>>    -VMName $vmName `
>>    -Name 'networkWatcherAgent' `
>>    -Publisher 'Microsoft.Azure.NetworkWatcher' `
>>    -Type 'NetworkWatcherAgentWindows' `
>>    -TypeHandlerVersion '1.4'
>> }

RequestId IsSuccessStatusCode StatusCode ReasonPhrase
--------- ------------------- ---------- ------------
                         True         OK OK
                         True         OK OK
                         True         OK OK
                         True         OK OK
```

Task 2: Configure the hub and spoke network topology

In this task, we will configure local peering between the virtual networks you deployed in the previous tasks in order to create a hub and spoke network topology.

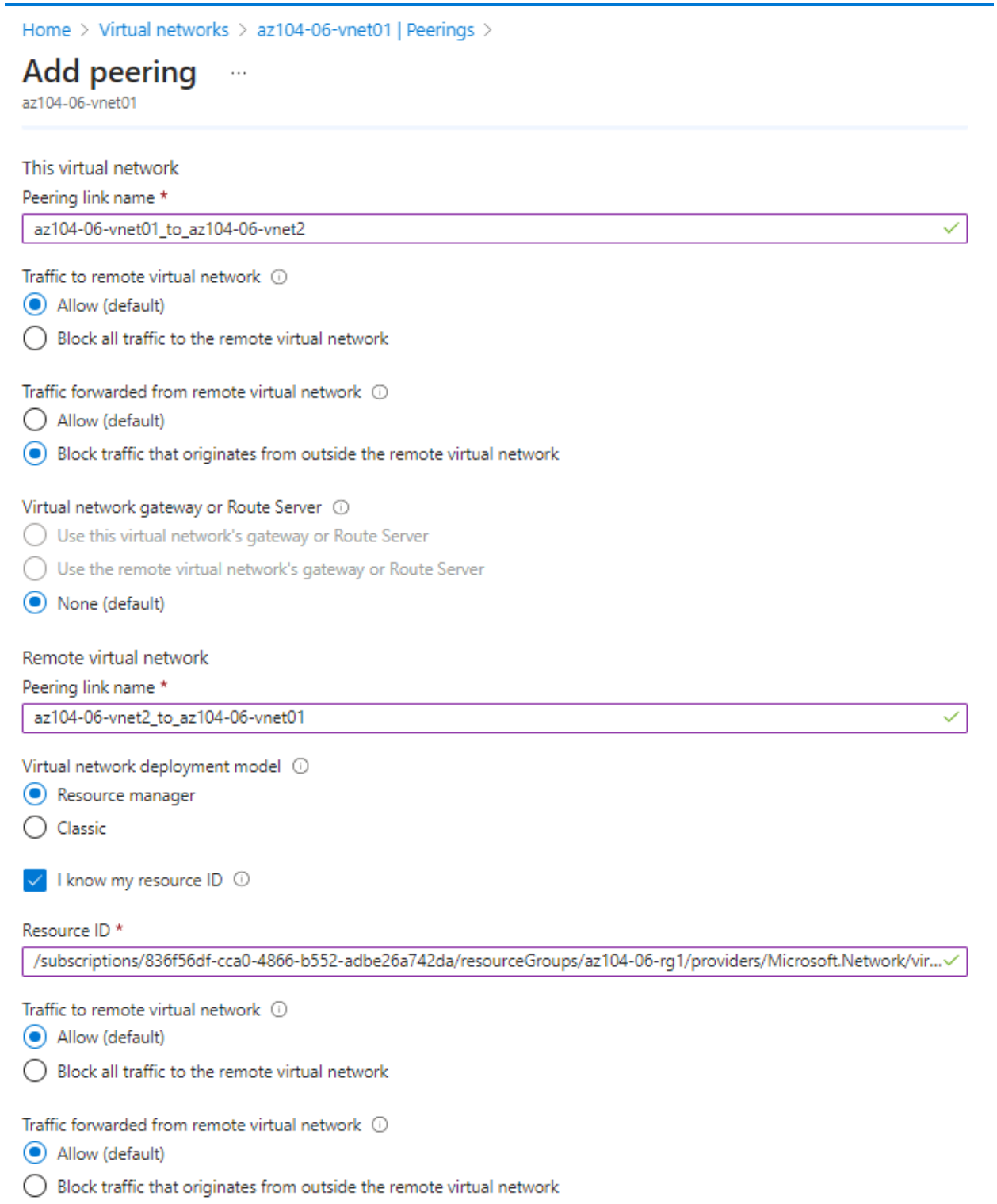
We've created 3 Virtual Networks:



Record the Resource ID of the second virtual network:

/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-06-rg1/providers/Microsoft.Network/virtualNetworks/az104-06-vnet2

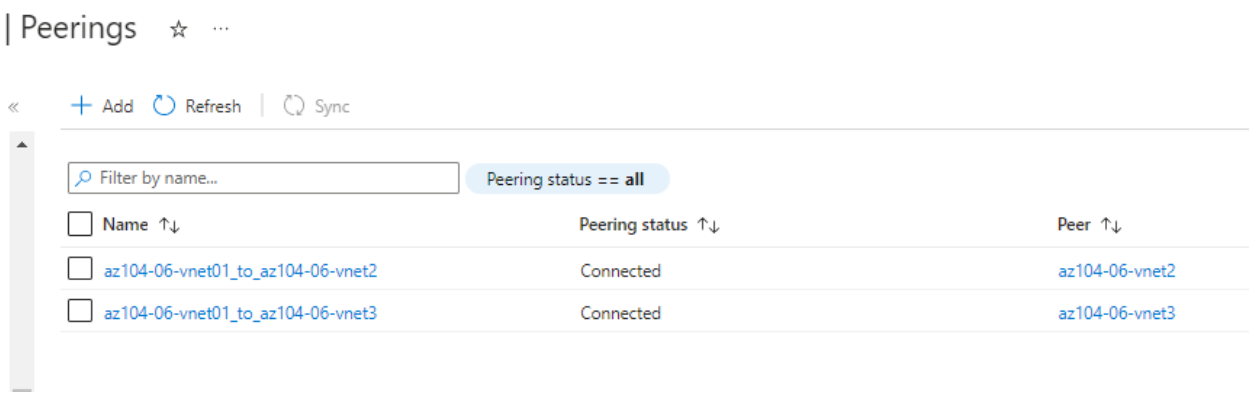
Next, we add Peerings to vnet01:



This step establishes two local peerings - one from az104-06-vnet01 to **az104-06-vnet2** and the other from **az104-06-vnet**2 to **az104-06-vnet01**.

We record the Resource ID of the Virtual Network 3:

/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-06-rg1/providers/Microsoft.Network/virtualNetworks/az104-06-vnet3



This step establishes two local peerings - one from az104-06-vnet01 to az104-06-vnet3 and the other from az104-06-vnet3 to az104-06-vnet01. This completes setting up the hub and spoke topology (with two spoke virtual networks).

Task 3: Test transitivity of virtual network peering

In this task, we will test transitivity of virtual network peering by using Network Watcher.

The transitivity or clustering coefficient of a network is a measure of the tendency of the nodes to cluster together. High transitivity means that the network contains communities or groups of nodes that are densely connected internally.

Network Watcher provides you the ability to diagnose your most common VPN Gateway and Connections issues. Allowing you, not only, to identify the issue but also to use the detailed logs created to help further investigate.

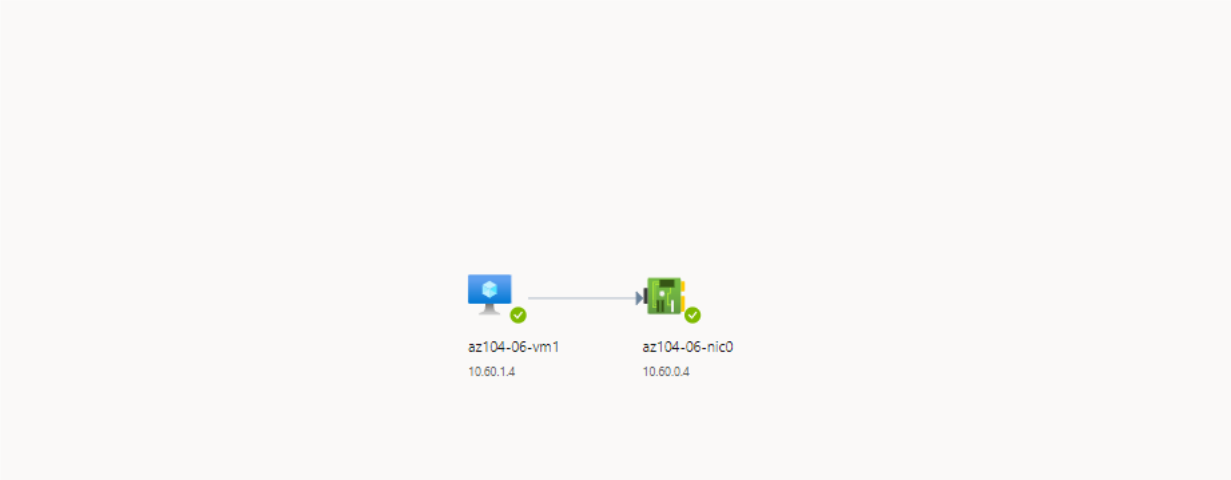
We initialize a Connection troubleshooting from **az104-06-vm1** to 10.60.0.4, which belongs to **az104-06-vm0**. We run the Diagnostics test:

Test	Status	Details	Suggestions
Connectivity Test	Fail	Probes Sent: 0 ,Probes Failed: 0	-
NSG Outbound (from source)	Success	Outbound communication from source is allowed	None
Next Hop (from source)	Success	Next Hop Type: VirtualNetwork Route Table Id: System Route	None

Hop by hop details

Name	Status	IP address	Next hop	RTT	Errors
az104-06-vm1	Info	10.60.1.4	10.60.0.4	-	-
az104-06-nic0	Info	10.60.0.4	-	-	-

Topology view



In the topology view, we can see the virtual networks are peered with each other.

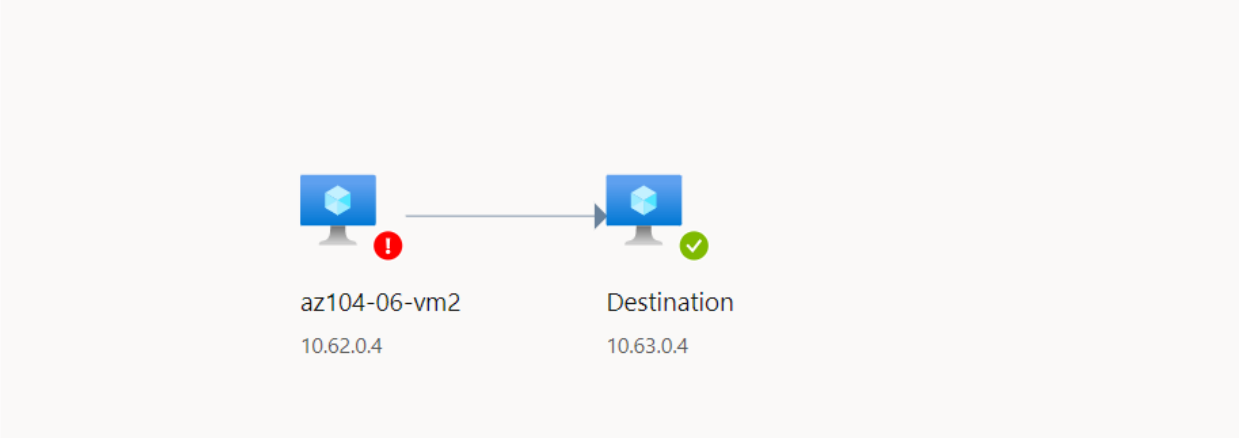
We initialize a Connection troubleshooting from **az104-06-vm2** to 10.63.0.4, which belongs to **az104-06-vm3**. We run the Diagnostics test:

Test	Status	Details	Suggestions
Connectivity Test	Fail	Probes Sent: 0 ,Probes Failed: 0	-
NSG Outbound (from source)	Fail	There are failed tests in the following NSGs: • az104-06-nsg2	Go to VM > Update the networking rule | Read docs
Next Hop (from source)	Success	Next Hop Type: None Route Table Id: System Route	None

Hop by hop details

Name	Status	IP address	Next hop	RTT	Errors
az104-06-vm2	Fail	10.62.0.4	10.63.0.4	-	Routes for the destination are missing in the virtual network gateway.
Destination (10.63.0.4)	Info	10.63.0.4	-	-	-

Topology view



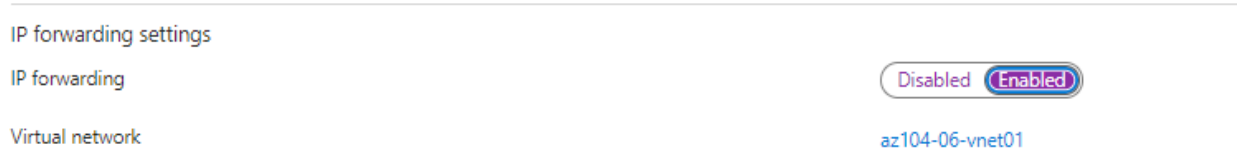
The test fails because the two virtual networks are not peered with each other. Virtual peering is not transitive, and this is an expected behavior when running the diagnostics test. Note that the status is Fail.

10.62.0.4 represents the private IP address of az104-06-vm2

Task 4: Configure routing in the hub and spoke topology

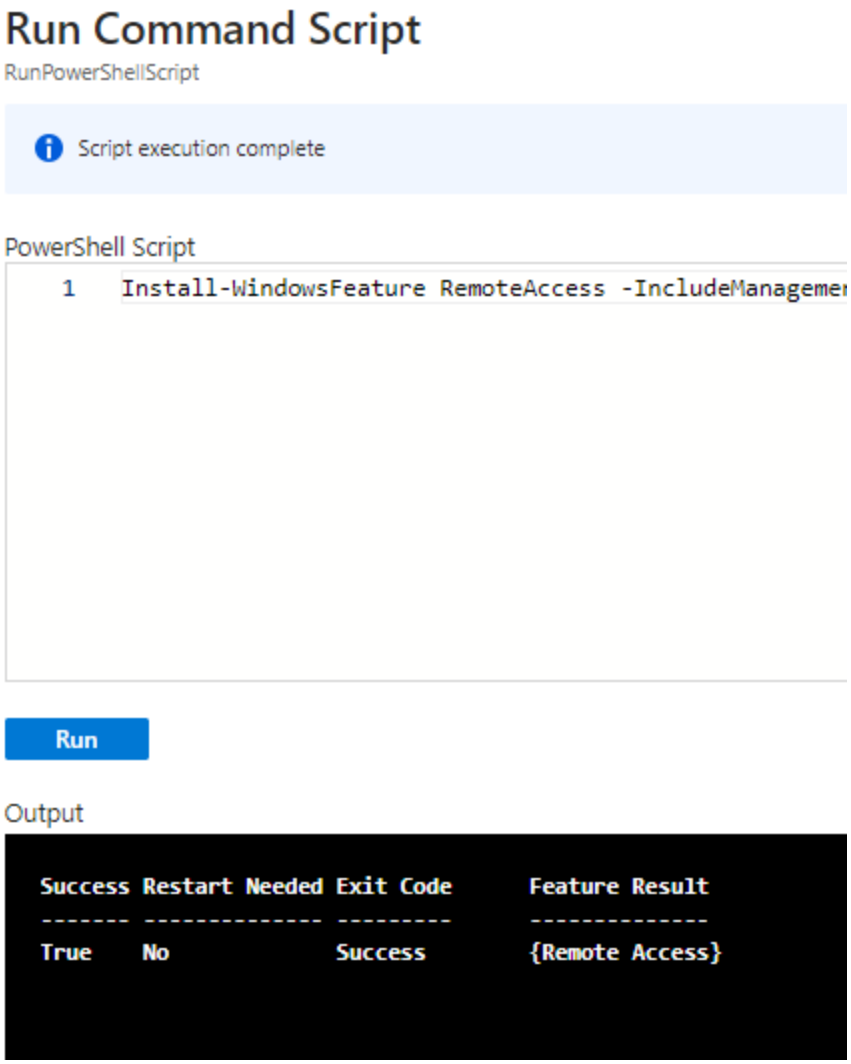
In this task, we will configure and test routing between the two spoke virtual networks by enabling IP forwarding on the network interface of the az104-06-vm0 virtual machine, enabling routing within its operating system, and configuring user-defined routes on the spoke virtual network.

Set IP forwarding to Enabled:



Next, we need to configure an operating system for the **az104-06-vm0** Virtual Machine by running the following command in Overview -> Run Command -> RunPowerShellScript:

```
Install-WindowsFeature RemoteAccess -IncludeManagementTools
```



We need to configure the operating system because this virtual machine will support routing.

Install the routing role service:

Run Command Script

RunPowerShellScript

Script execution complete

PowerShell Script

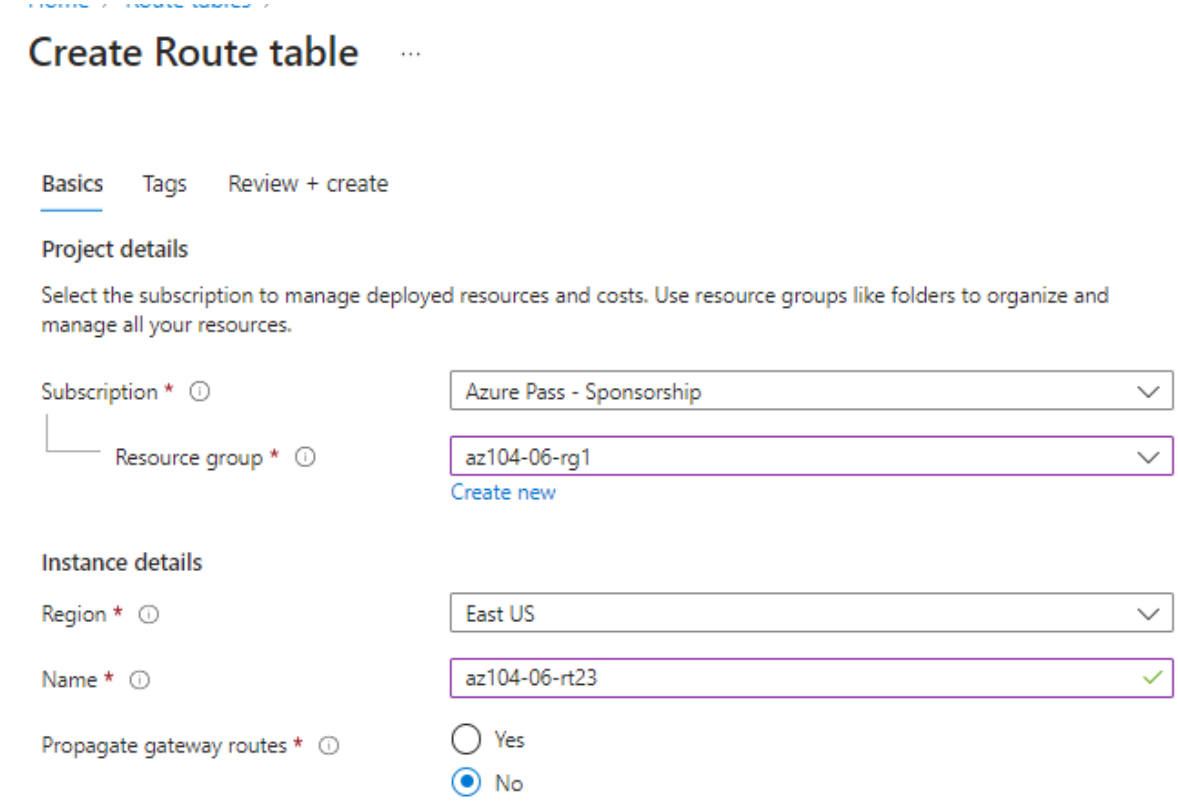
```
1  Install-WindowsFeature -Name Routing -IncludeManagementTools -IncludeAllSubFe
2
3  Install-WindowsFeature -Name "RSAT-RemoteAccess-Powershell"
4
5  Install-RemoteAccess -VpnType RoutingOnly
6
7  Get-NetAdapter | Set-NetIPInterface -Forwarding Enabled
```

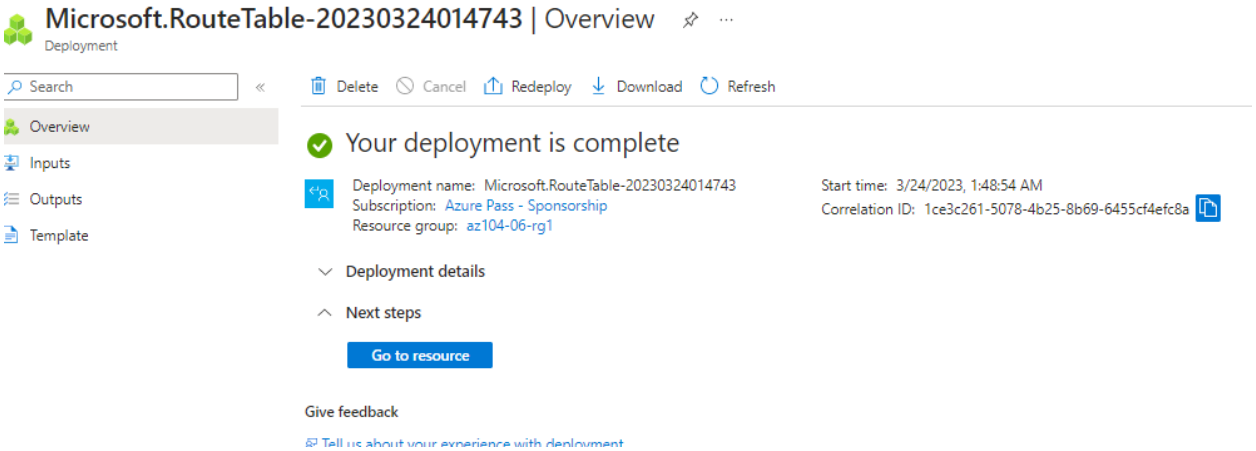
Run

Output

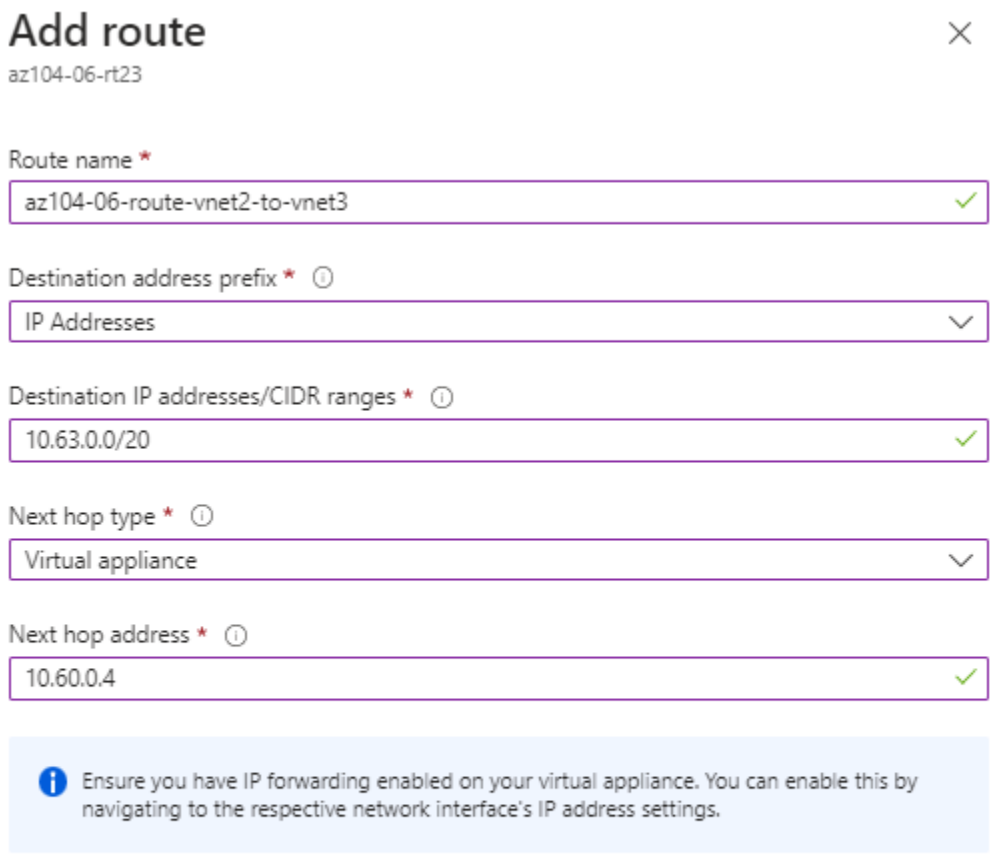
```
Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {RAS Connection Manager Administration Kit...
True    No             NoChangeNeeded {}
```

Create route tables:

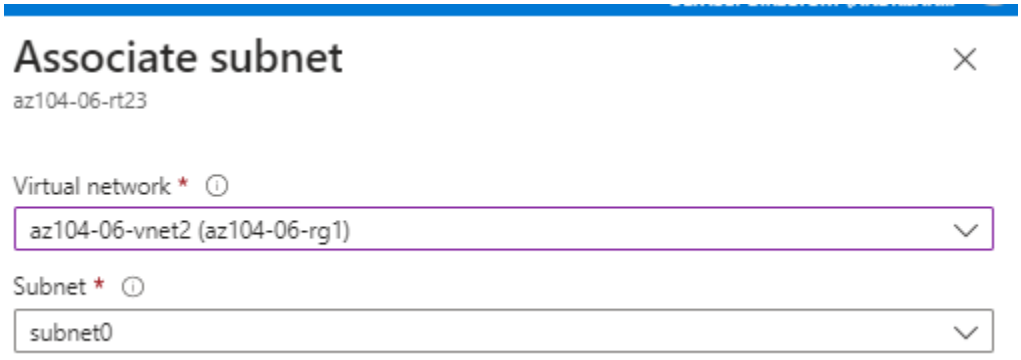




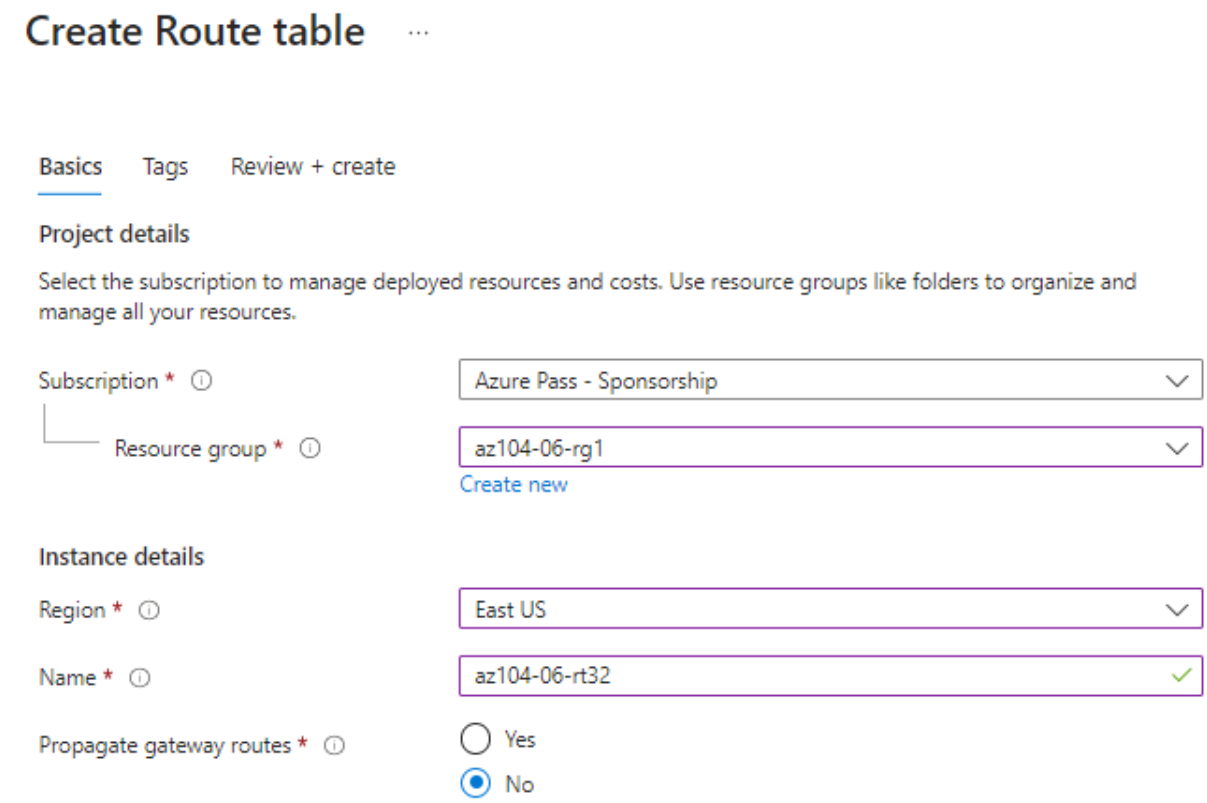
After the deployment is finished, we click Go to resource where we add a route with the following settings:

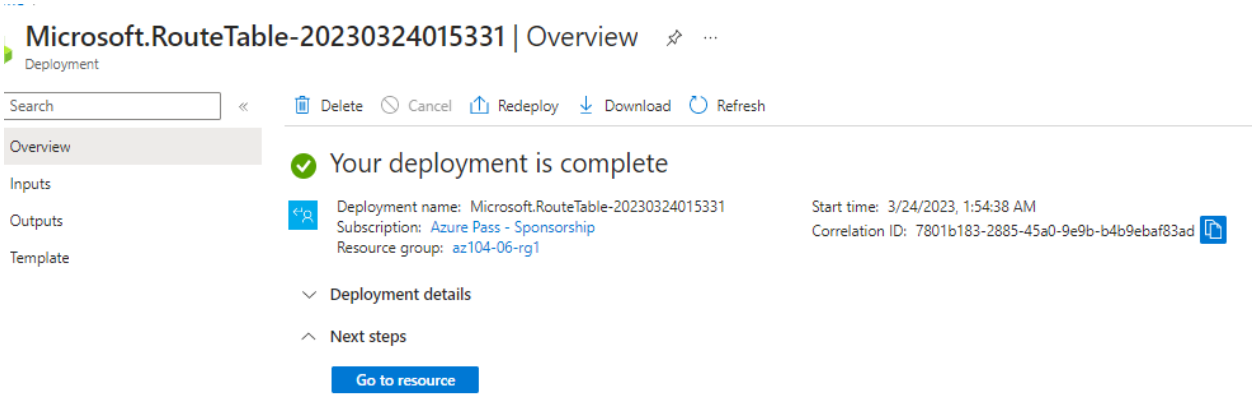


Then, we add an associate subnet:

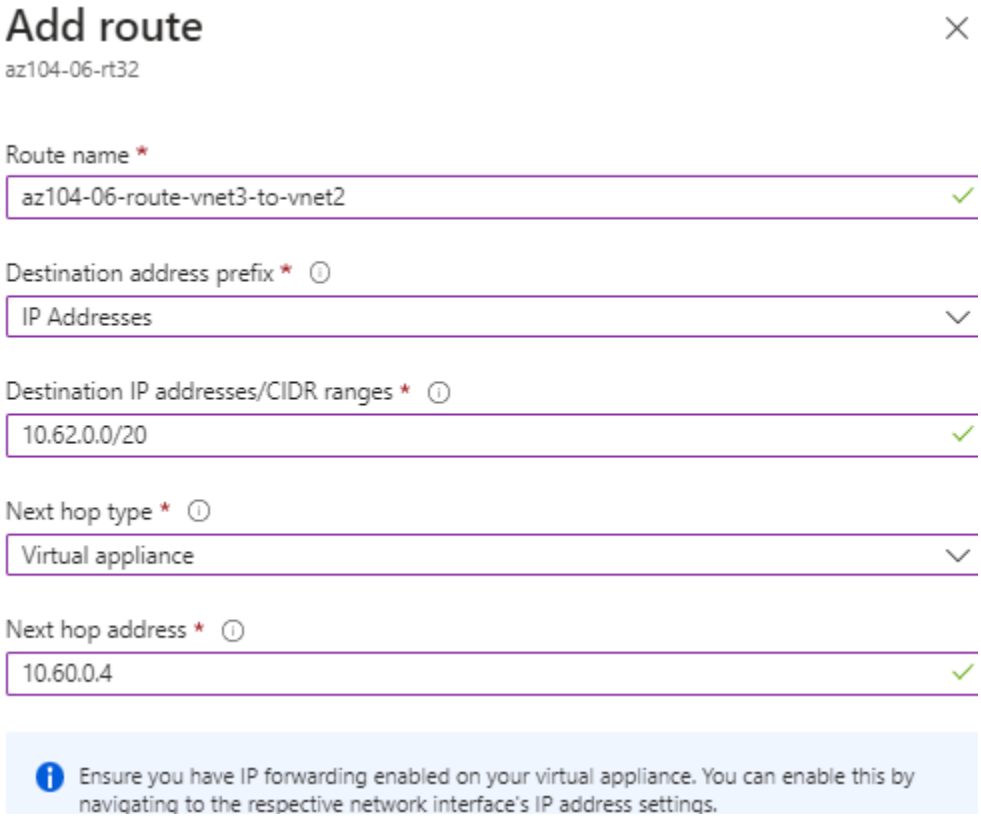


We created another route table:

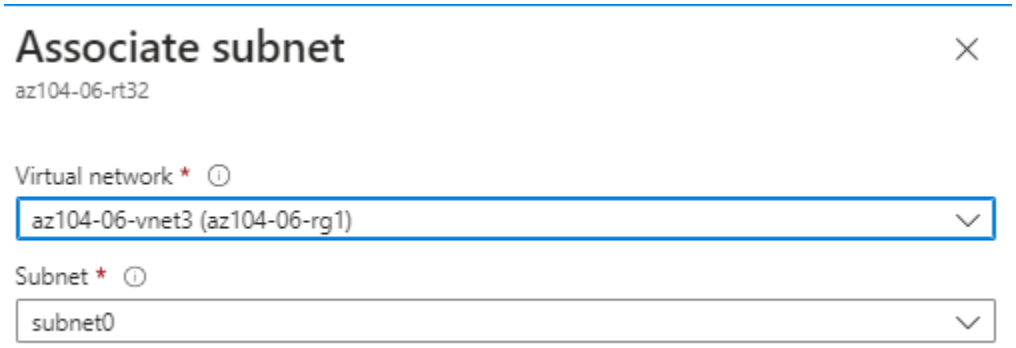




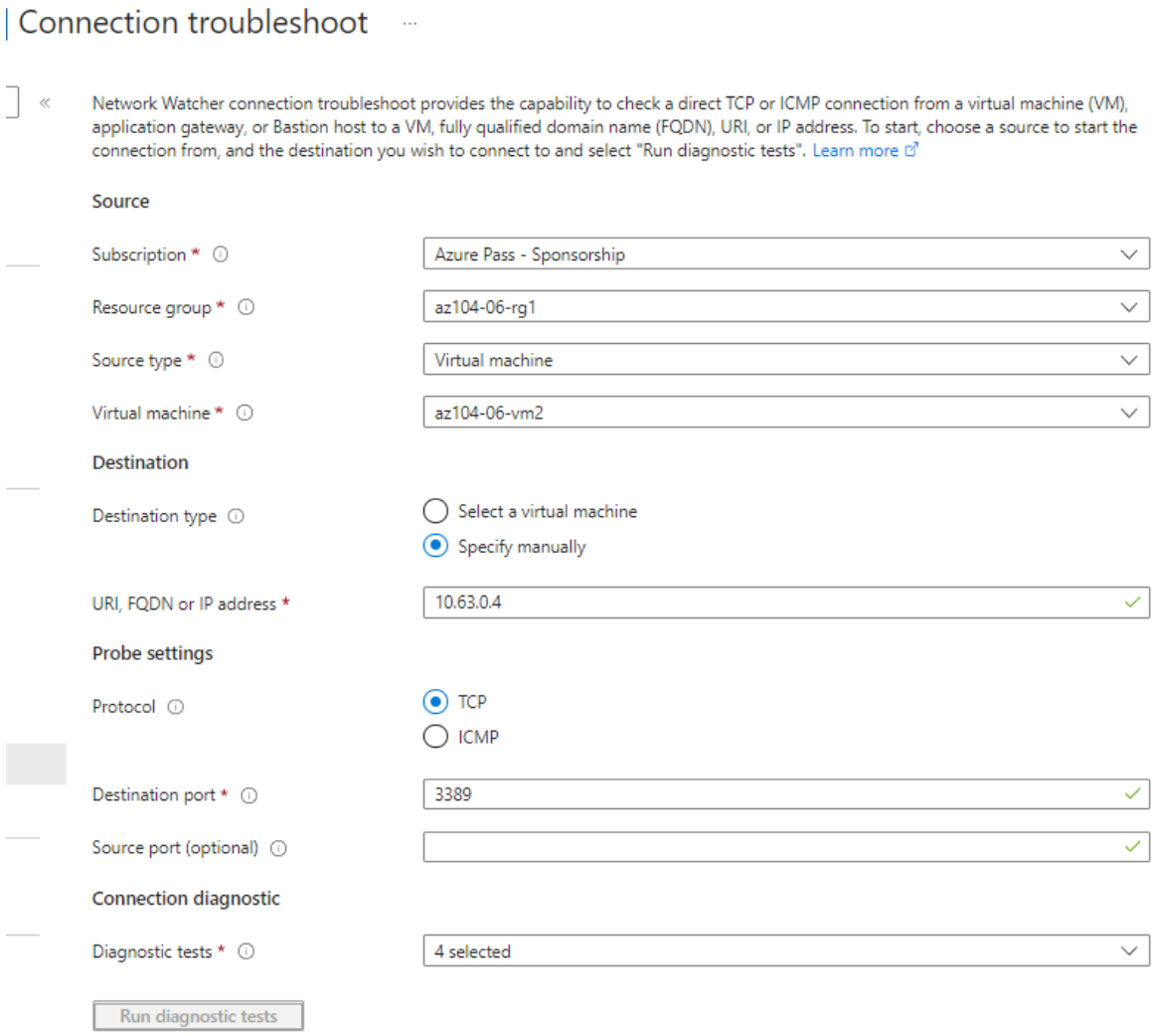
Then, we create another route from the third virtual machine to the first virtual machine:



And we add an associate subnet:



We run diagnostics test with the following settings:



So, it's basically a test from the third virtual machine to the fourth one.

Click Check and wait until results of the connectivity check are returned. Verify that the status is Reachable. Review the network path and note that the traffic was routed via 10.60.0.4, assigned to the az104-06-nic0 network adapter. If status is Unreachable, we should stop and then start az104-06-vm0.

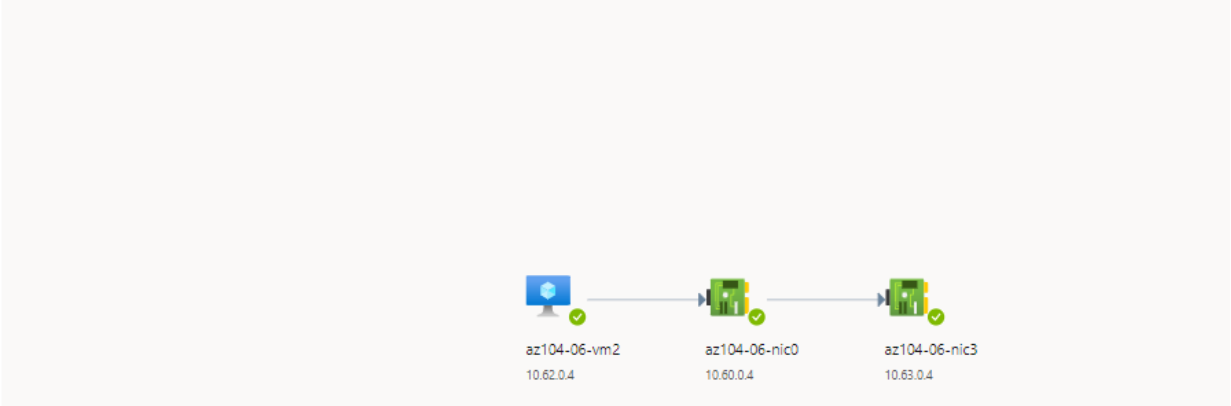
Diagnostic tests

Test	Status	Details	Suggestions
Connectivity Test	Fail	Probes Sent: 0 ,Probes Failed: 0	-
NSG Outbound (from source)	Success	Outbound communication from source is allowed	None
Next Hop (from source)	Success	Next Hop Type: VirtualAppliance Next Hop IP: 10.60.0.4	None

Hop by hop details

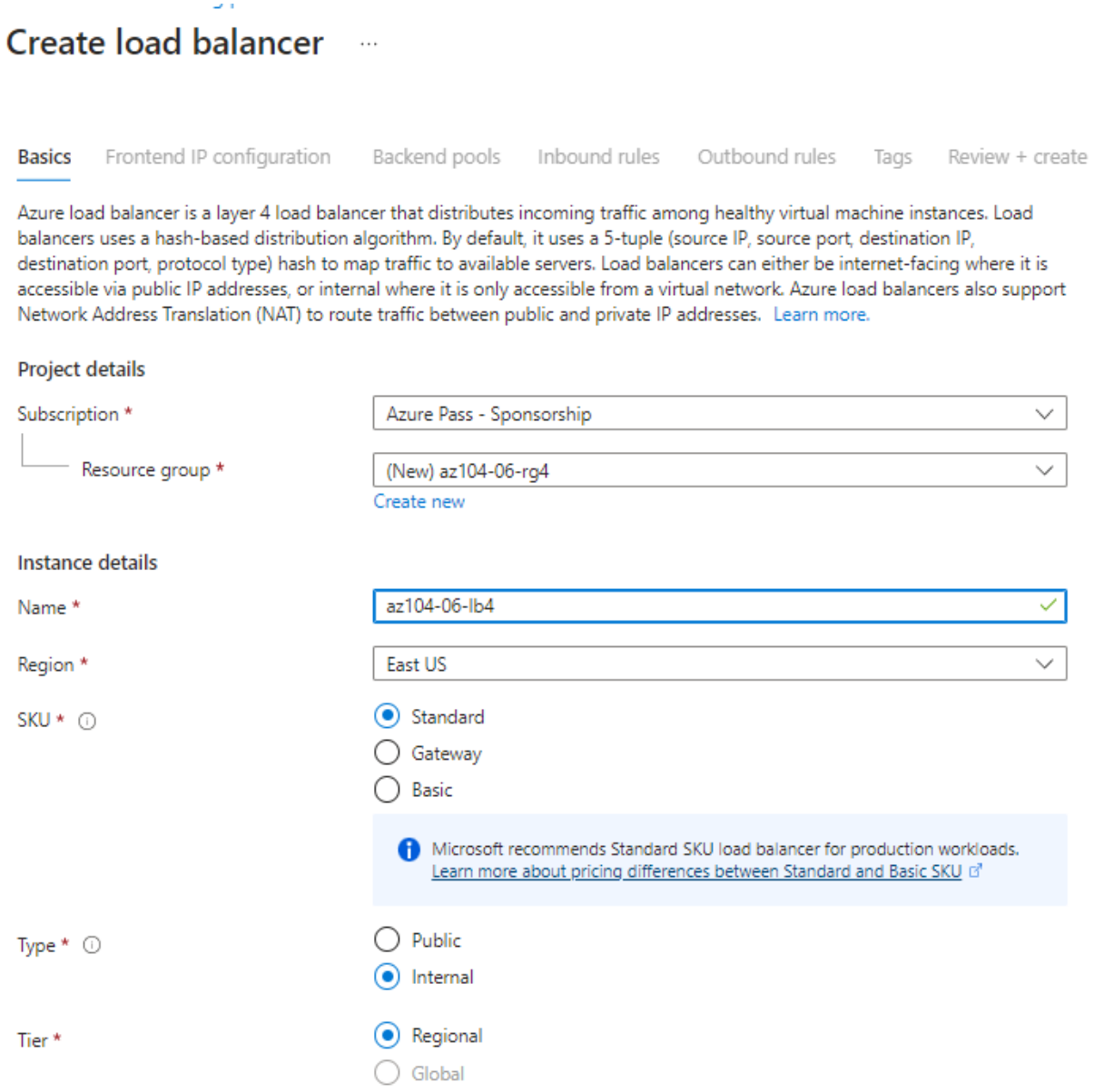
Name	Status	IP address	Next hop	RTT	Errors
az104-06-vm2	Info	10.62.0.4	10.60.0.4	-	-
az104-06-nic0	Info	10.60.0.4	10.63.0.4	-	-
az104-06-nic3	Info	10.63.0.4	-	-	-

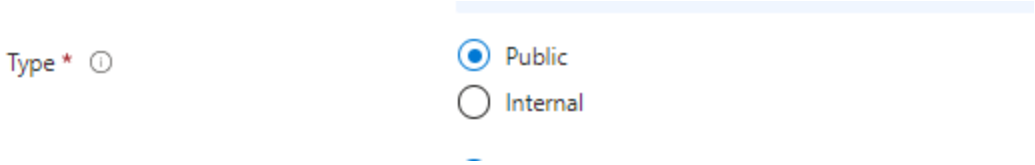
Topology view



The connection is established.

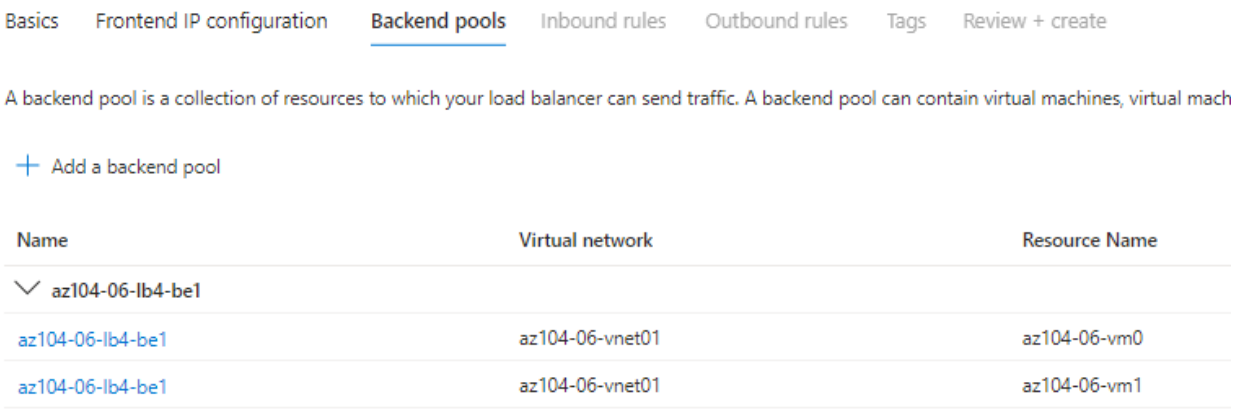
Task 5: Implement Azure Load Balancer



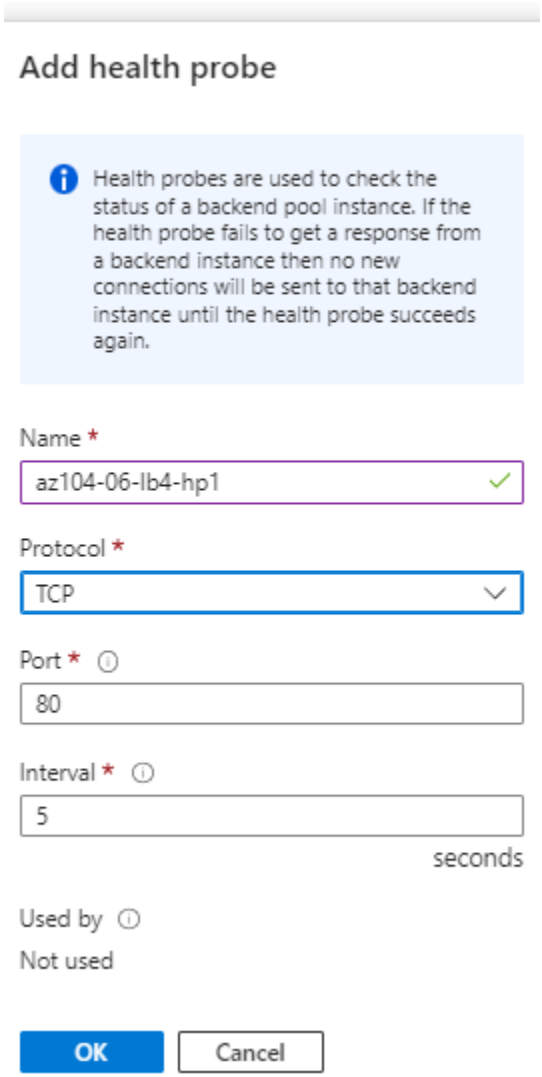


Type should be Public (I've set this as internal previously - my mistake, and got some errors)

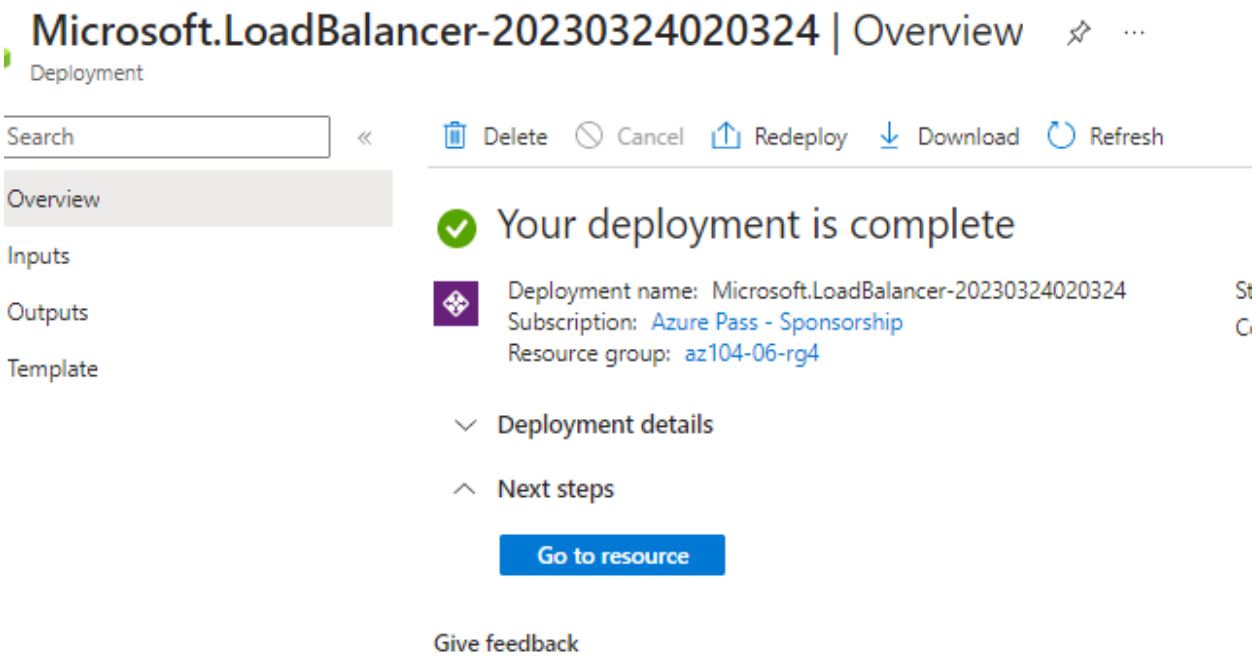
We add the backend pools:



Adding a health probe:

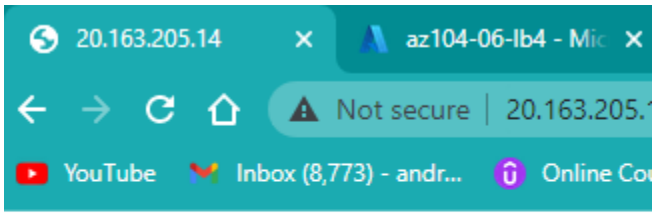


We create the Load Balancer:

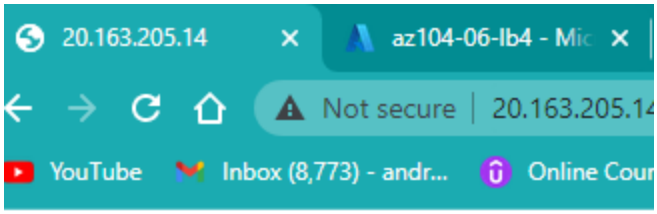


Select Frontend IP configuration from the Load Balancer resource page. Copy the IP address.

Then we open another browser tab and navigate to the IP address. Verify that the browser window displays the message Hello World from az104-06-vm0 or Hello World from az104-06-vm1.



Hello World from az104-06-vm1



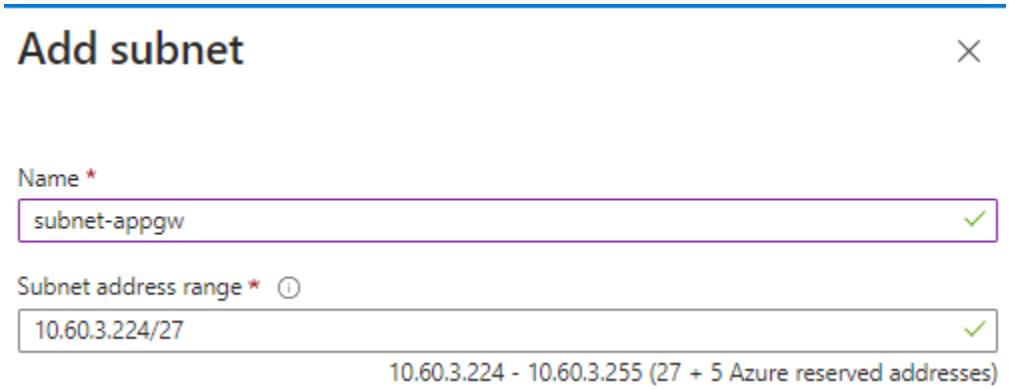
This demonstrates the load balancer rotating through the virtual machines.

Task 6: Implement Azure Application Gateway

In this task, we will implement an Azure Application Gateway in front of the two Azure virtual machines in the spoke virtual networks.

Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications. Traditional load balancers operate at the transport layer (OSI layer 4 - TCP and UDP) and route traffic based on source IP address and port, to a destination IP address and port.

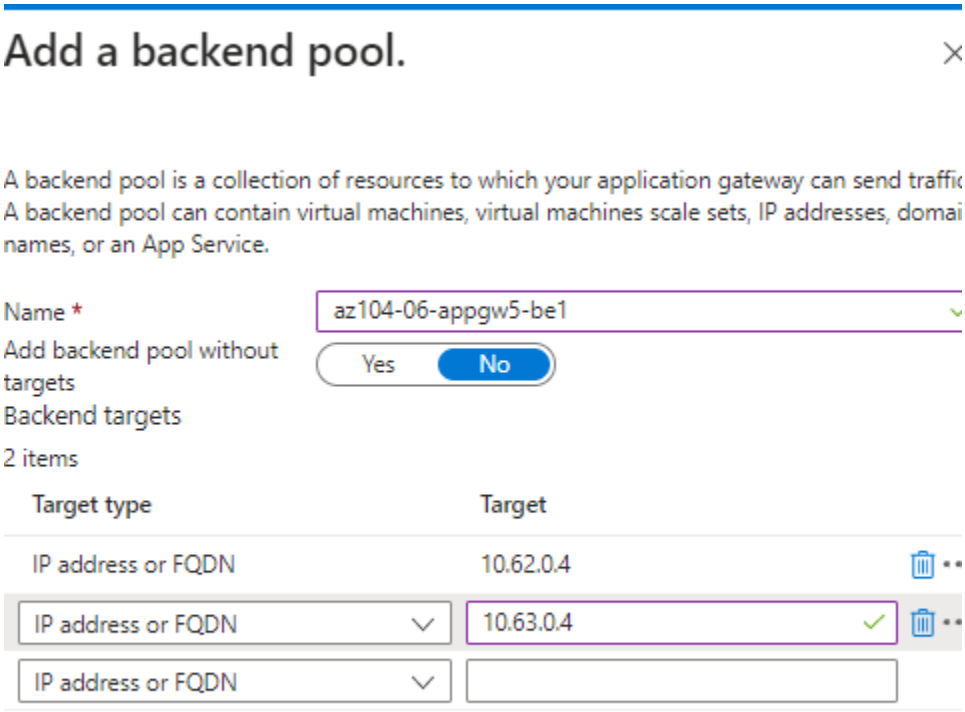
Add a subnet to az104-06-vnet01:



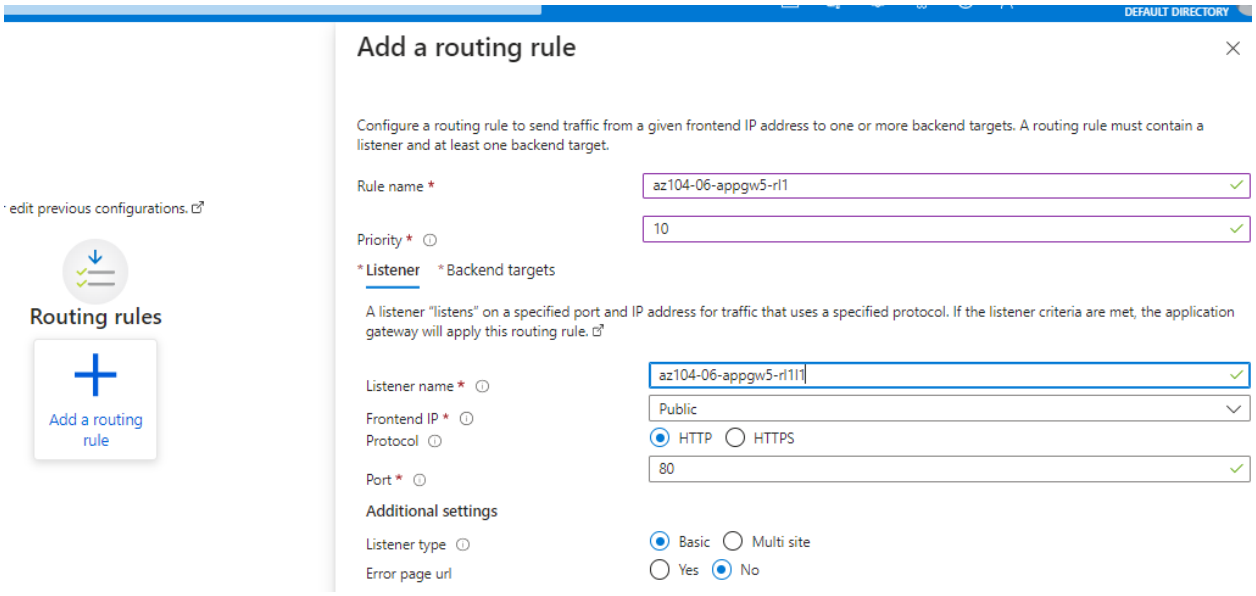
This subnet will be used by the Azure Application Gateway instances, which you will deploy later in this task. The Application Gateway requires a dedicated subnet of /27 or larger size.

Then, we add a backend pool.

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).



And we add a routing rule so we can send traffic from the given IP address to one or more backend targets:



As the final step, we deploy the Application Gateway.