
Cloud & Virtualization Class

Lab 2 · Virtual Networks

Adam Lahbib · M. Sofiene Barka · Mohamed Rafraf

3/1/2023 @ INSAT

Contents

1	Introduction	4
1.1	Lab Objectives	4
2	Lab Walkthrough	5
2.1	Task 1	5
2.1.1	1. Uploading definition files to Azure Cloud Shell	5
2.1.2	2. Creating a resource group in North Europe	6
2.1.3	3. Creating 3 VMs, each in a virtual network using the definition files	6
2.1.4	4. vnet00 Overview	7
2.1.5	5. Making a peering link between vnet00 and vnet01 and vice versa	7
2.1.6	6. Configure peering link between vnet00 and vnet02 and vice versa using Cloud Shell	9
2.1.7	7. Configure peering link between vnet01 and vnet02 and vice versa using Cloud Shell	10
2.1.8	8+9+10. Testing vm00 Connectivity to vm01 and vm02	11
2.1.9	11. Testing vm01 Connectivity to vm02	12
2.1.10	12. Cleaning up	13
2.2	Task 2	14
2.2.1	1. Uploading second definition files to Azure Cloud Shell	14
2.2.2	2. Creating a resource group in North Europe	15
2.2.3	3. Creating 3 VMs, each in a virtual network using the definition files	15
2.2.4	4. Installing Network Watcher extension on the VMs	15
2.2.5	5+6. Kept records of vnet01 and vnet02 resource IDs!	17
2.2.6	7+8. Configure peering link between vnet00 and vnet01 then vnet00 and vnet02 using Azure Portal	17
2.2.7	9. Testing transitivity of vm00 to vm01 using the Network Watcher	19
2.2.8	10. Testing transitivity of vm00 to vm02 using the Network Watcher	20
2.2.9	11. Testing transitivity of vm01 to vm02 using the Network Watcher	21
2.2.10	12. Configuring routing between vnet01 and vnet02 by enabling IP forwarding	21
2.2.11	13. Installing remote access Windows server role	22
2.2.12	14. Installing routing role service	23

2.2.13	15+16. Create user-defined routes on the virtual networks.	26
2.2.14	17. Create a route for vnet01 to vnet02	27
2.2.15	18. Associate the route table with the vnet01 subnet0	28
2.2.16	19. Create a route table for vnet02	30
2.2.17	20. Create a route for vnet02 to vnet01	31
2.2.18	21. Associate the route table with the vnet02 subnet0	32
2.2.19	22. Network Watcher test	33
2.2.20	23. Cleaning Up	33
3	Conclusion	35

1 Introduction

This lab will guide us to create our virtual machines and virtual networks based on definition files (template + parameters). We will test the connectivity between VMs and the transitivity of virtual network peering. We will also use the Network Watcher to test the transitivity of virtual network peering.

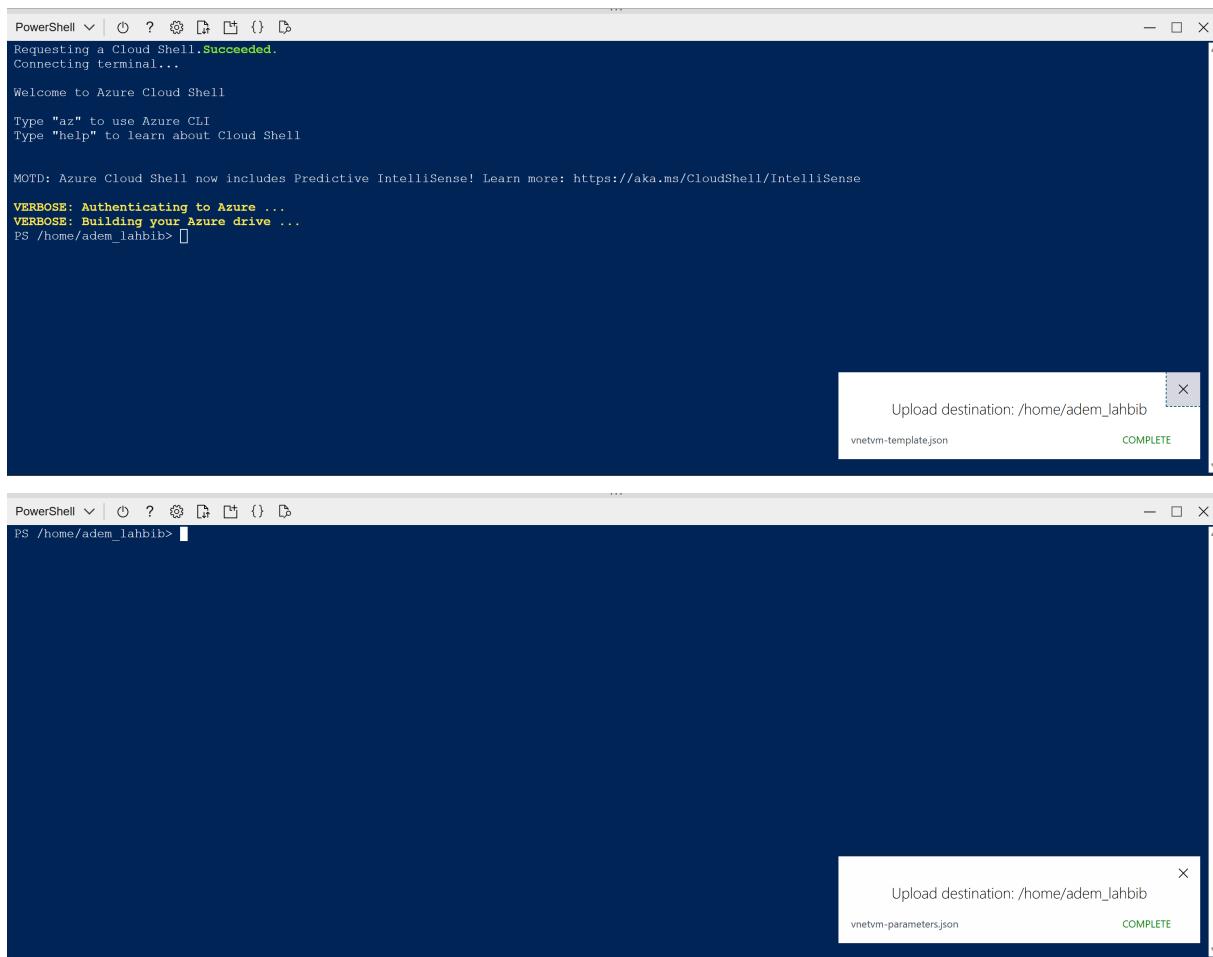
1.1 Lab Objectives

- Create virtual machines and virtual networks based on definition files (template + parameters).
- Test the connectivity between VMs.
- Familiarity with Azure Cloud Shell as well as the Azure Portal.

2 Lab Walkthrough

2.1 Task 1

2.1.1 1. Uploading definition files to Azure Cloud Shell



The screenshot shows two separate instances of the Azure Cloud Shell interface. Both instances are running PowerShell and show a terminal session.

The top instance shows the following output:

```
PowerShell | ⚡ ? ⚡ 🔍 ⚡ () 🔍
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/adem_lahbib> []
```

A modal dialog box is open in the top instance, indicating the upload of a file named `vnetvm-template.json` to the destination `/home/adem_lahbib`. The status is **COMPLETE**.

The bottom instance shows the following output:

```
PowerShell | ⚡ ? ⚡ 🔍 ⚡ () 🔍
PS /home/adem_lahbib> []
```

A modal dialog box is open in the bottom instance, indicating the upload of a file named `vnetvm-parameters.json` to the destination `/home/adem_lahbib`. The status is **COMPLETE**.

2.1.2 2. Creating a resource group in North Europe

2.1.3 3. Creating 3 VMs, each in a virtual network using the definition files

2.1.4 4. vnet00 Overview

The screenshot shows the Microsoft Azure portal interface for the 'vnet00' virtual network. The left sidebar lists other virtual networks: 'vnet00' (selected), 'vnet01', and 'vnet02'. The main panel displays the 'Overview' tab for 'vnet00'. Key details shown include:

- Resource group:** tp2-rg1
- Address space:** 10.50.0.0/22
- Location:** North Europe
- Subscription:** Adam
- Virtual network ID:** 97cd8887-e707-4ecb-8c73-4eef77a13527
- DNS servers:** Azure provided DNS service
- Flow timeout:** Configure
- BGP community string:** Configure
- Virtual network ID:** 145bd0af-9a5b-4742-90cd-f814ad8e08b3

The 'Capabilities' section lists four features with status 'Not configured':

- DDoS protection
- Azure Firewall
- Peering
- Microsoft Defender for Cloud

2.1.5 5. Making a peering link between vnet00 and vnet01 and vice versa

A peering link is a connection between two virtual networks that allows them to communicate with each other. It is a logical connection between two virtual networks that is established by the user. It is not a physical connection. It is a connection that is established between two virtual networks that are in the same region. It is not a connection that is established between two virtual networks that are in different regions.

The image shows two screenshots of the Microsoft Azure portal interface.

Top Screenshot: Add peering - Microsoft Azure

This screenshot shows the 'Add peering' blade for creating a virtual network peering between two virtual networks. The 'Peering link name' is set to 'vnet00_to_vnet01'. Under 'Traffic to remote virtual network', the 'Allow (default)' option is selected. Under 'Traffic forwarded from remote virtual network', the 'Block traffic that originates from outside the remote virtual network' option is selected. Under 'Virtual network gateway or Route Server', the 'None (default)' option is selected. Under 'Remote virtual network', the 'Peering link name' is set to 'vnet01_to_vnet00'. Under 'Virtual network deployment model', the 'Resource manager' option is selected. At the bottom, there is a blue 'Add' button.

Bottom Screenshot: vnet00 - Microsoft Azure

This screenshot shows the 'vnet00 | Peerings' blade, which lists the existing peering connection. The table shows one entry:

Name	Peer	Gateway transit
vnet00_to_vnet01	vnet01	Disabled

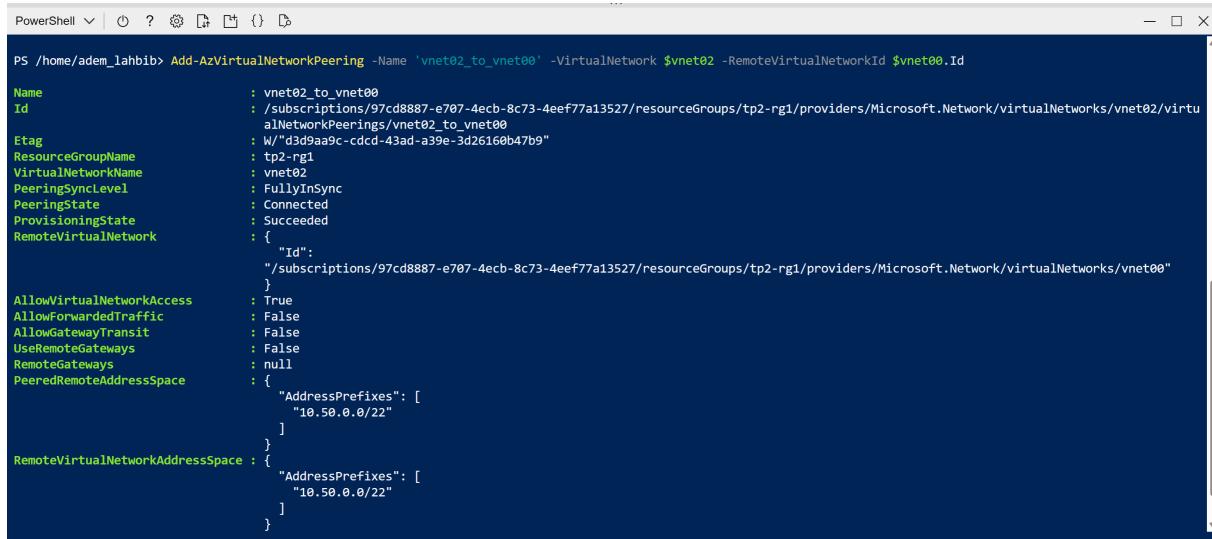
The screenshot shows the Azure portal interface for managing virtual networks. The left sidebar is expanded to show 'Peering' under 'vnet01'. The main area displays a table of peering connections. One connection is listed: 'vnet01_to_vnet00' with a status of 'Connected' and 'Peer' 'vnet00'. The 'Peering' option in the sidebar is highlighted.

2.1.6 6. Configure peering link between vnet00 and vnet02 and vice versa using Cloud Shell

```

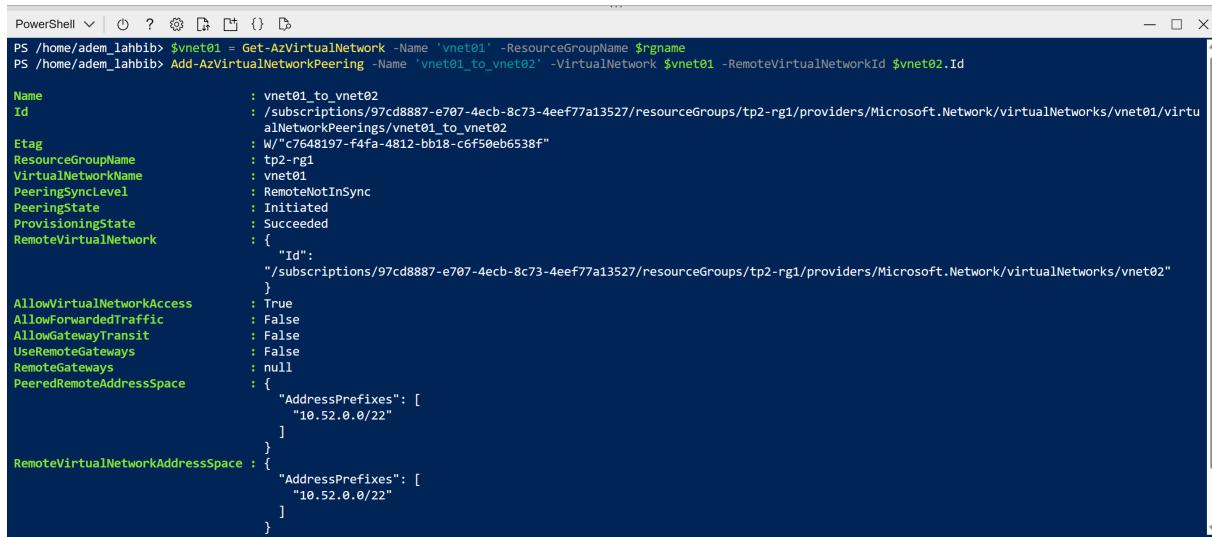
PowerShell PS /home/adem_lahbib> $vnet00 = Get-AzVirtualNetwork -Name 'vnet00' -ResourceGroupName $rgname
PS /home/adem_lahbib> $vnet02 = Get-AzVirtualNetwork -Name 'vnet02' -ResourceGroupName $rgname
PS /home/adem_lahbib> Add-AzVirtualNetworkPeering -Name 'vnet00_to_vnet02' -VirtualNetworkId $vnet02.Id
Name          : vnet00_to_vnet02
Id           : /subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet00/virtua
lNetworkPeerings/vnet00_to_vnet02
Etag         : W/"f19e99ea-ea2a-47a5-9a39-0071e73b9f98"
ResourceGroupName : tp2-rg1
VirtualNetworkName : vnet00
PeeringSyncLevel : RemoteNotInSync
PeeringState    : Initiated
ProvisioningState : Succeeded
RemoteVirtualNetwork : {
  "Id": "/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet02"
}
AllowVirtualNetworkAccess : True
AllowForwardedTraffic : False
AllowGatewayTransit : False
UserRemoteGateways : False
RemoteGateways : null
PeeredRemoteAddressSpace : {
  "AddressPrefixes": [
    "10.52.0.0/22"
  ]
}
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.52.0.0/22"
  ]
}

```



```
PowerShell PS /home/adem_lahbib> Add-AzVirtualNetworkPeering -Name 'vnet02_to_vnet00' -VirtualNetwork $vnet02 -RemoteVirtualNetworkId $vnet00.Id
Name          : vnet02_to_vnet00
Id           : /subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet02/virtu
               alNetworkPeerings/vnet02_to_vnet00
Etag         : W/"3d9a9a5c-cddc-43ad-a39e-3d26160b47b9"
ResourceGroupName : tp2-rg1
VirtualNetworkName : vnet02
PeeringSyncLevel : FullyInSync
PeeringState    : Connected
ProvisioningState : Succeeded
RemoteVirtualNetwork : {
  "Id": "/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet00"
}
AllowVirtualNetworkAccess : True
AllowForwardedTraffic : False
AllowGatewayTransit : False
UseRemoteGateways : False
RemoteGateways : null
PeeredRemoteAddressSpace : {
  "AddressPrefixes": [
    "10.50.0.0/22"
  ]
}
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.50.0.0/22"
  ]
}
```

2.1.7 7. Configure peering link between vnet01 and vnet02 and vice versa using Cloud Shell



```
PowerShell PS /home/adem_lahbib> $vnet01 = Get-AzVirtualNetwork -Name 'vnet01' -ResourceGroupName $rgname
PS /home/adem_lahbib> Add-AzVirtualNetworkPeering -Name 'vnet01_to_vnet02' -VirtualNetwork $vnet01 -RemoteVirtualNetworkId $vnet02.Id
Name          : vnet01_to_vnet02
Id           : /subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet01/virtu
               alNetworkPeerings/vnet01_to_vnet02
Etag         : W/"7648197-f4fa-4812-bb18-c6f50eb6538f"
ResourceGroupName : tp2-rg1
VirtualNetworkName : vnet01
PeeringSyncLevel : RemoteNotInSync
PeeringState    : Initiated
ProvisioningState : Succeeded
RemoteVirtualNetwork : {
  "Id": "/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet02"
}
AllowVirtualNetworkAccess : True
AllowForwardedTraffic : False
AllowGatewayTransit : False
UseRemoteGateways : False
RemoteGateways : null
PeeredRemoteAddressSpace : {
  "AddressPrefixes": [
    "10.52.0.0/22"
  ]
}
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.52.0.0/22"
  ]
}
```

```

PS /home/adem_lahbib> Add-AzVirtualNetworkPeering -Name 'vnet02_to_vnet01' -VirtualNetwork $vnet02 -RemoteVirtualNetworkId $vnet01.Id

Name          : vnet02_to_vnet01
Id           : /subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet02/virtu
               alNetworkPeerings/vnet02_to_vnet01
Etag         : W/"478a8ce-7be-4435-a1f7-841a85ace67"
ResourceGroupName : tp2-rg1
VirtualNetworkName : vnet02
PeeringSyncLevel : FullyInSync
PeeringState    : Connected
ProvisioningState : Succeeded
RemoteVirtualNetwork : {
  "Id": "/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg1/providers/Microsoft.Network/virtualNetworks/vnet01"
}
AllowVirtualNetworkAccess : True
AllowForwardedTraffic : False
AllowGatewayTransit : False
UseRemoteGateways : False
RemoteGateways : null
PeeredRemoteAddressSpace : {
  "AddressPrefixes": [
    "10.51.0.0/22"
  ]
}
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.51.0.0/22"
  ]
}

```

2.1.8 8+9+10. Testing vm00 Connectivity to vm01 and vm02

`Test-NetConnection -ComputerName 10.51.0.4 -Port 3389 -InformationLevel 'Detailed'` will test the connection to the vm01 from vm00.

```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

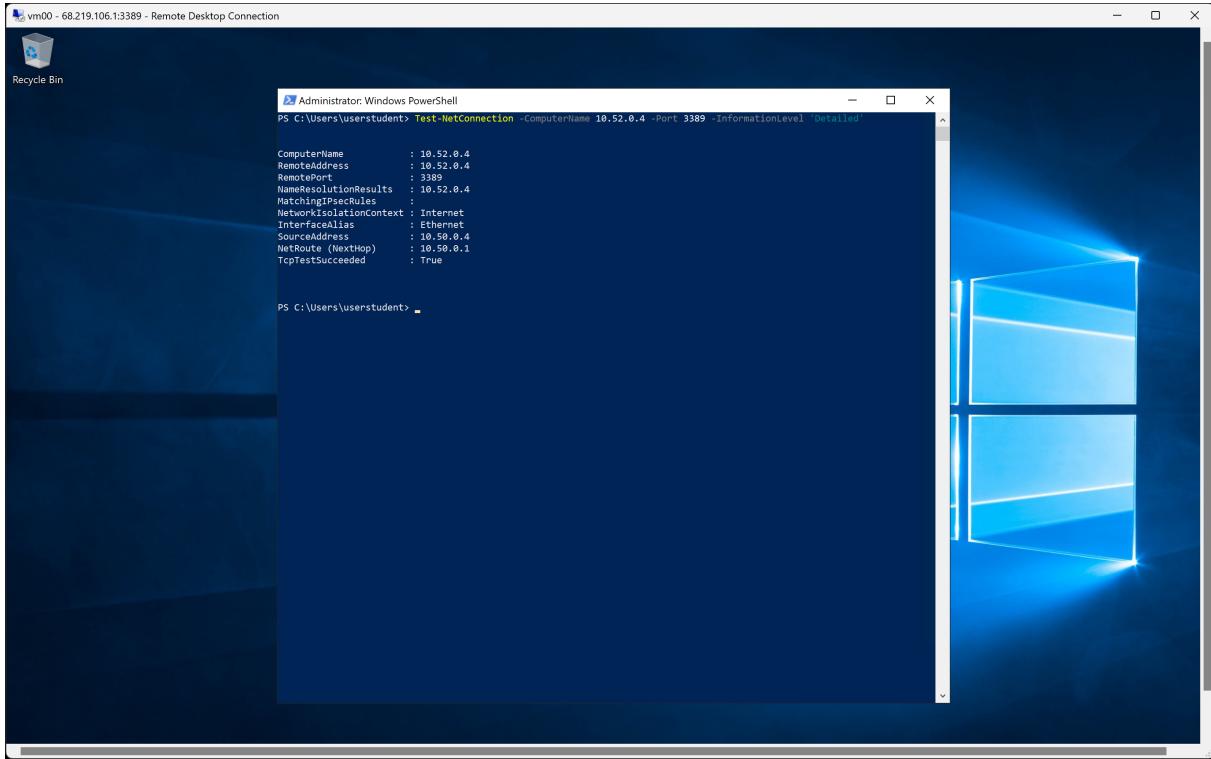
PS C:\Users\userstudent> Test-NetConnection -ComputerName 10.51.0.4 -Port 3389 -InformationLevel 'detailed'
>>>

ComputerName      : 10.51.0.4
RemoteAddress     : 10.51.0.4
RemotePort        : 3389
NameResolutionResults : 10.51.0.4
MatchingIPsecRules : 
NetworkIsolationContext : Internet
InterfaceAlias   : Ethernet
SourceAddress    : 10.50.0.4
NetRoute (NextHop) : 10.50.0.1
TcpTestSucceeded : True

PS C:\Users\userstudent>

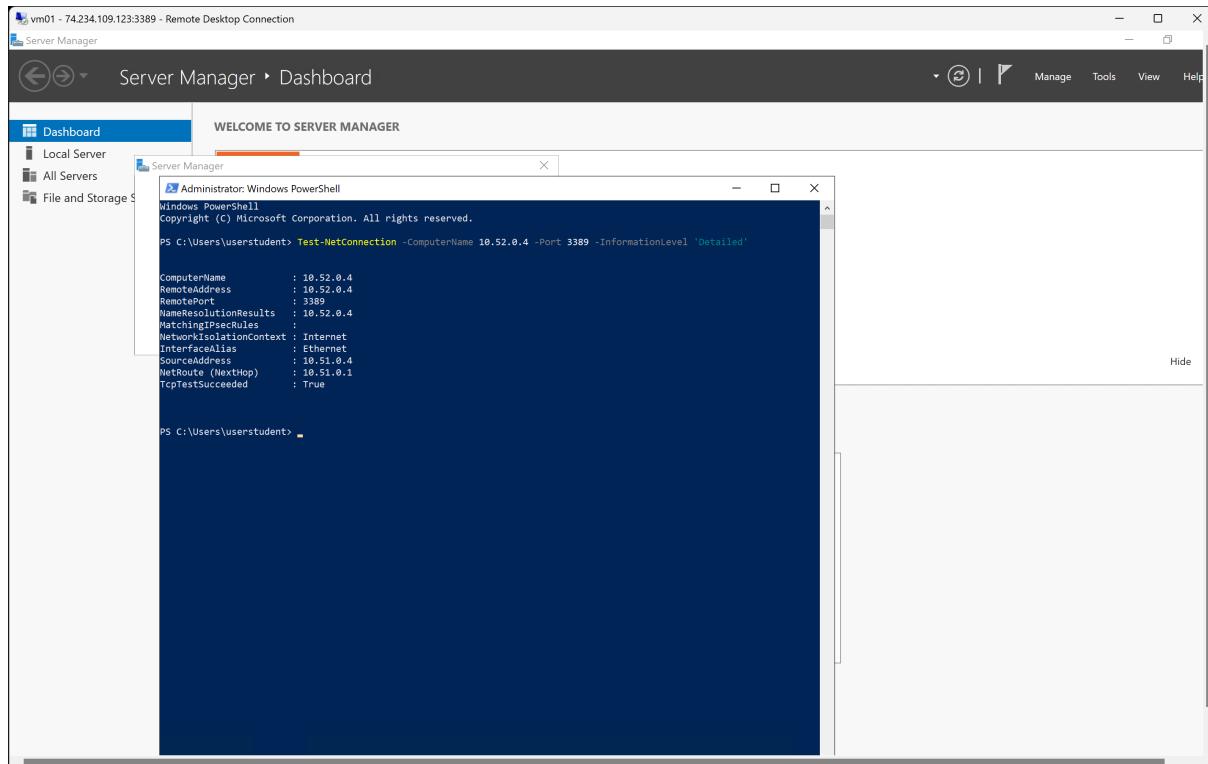
```

`Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel 'Detailed'` will test the connection to the vm02 from vm00.



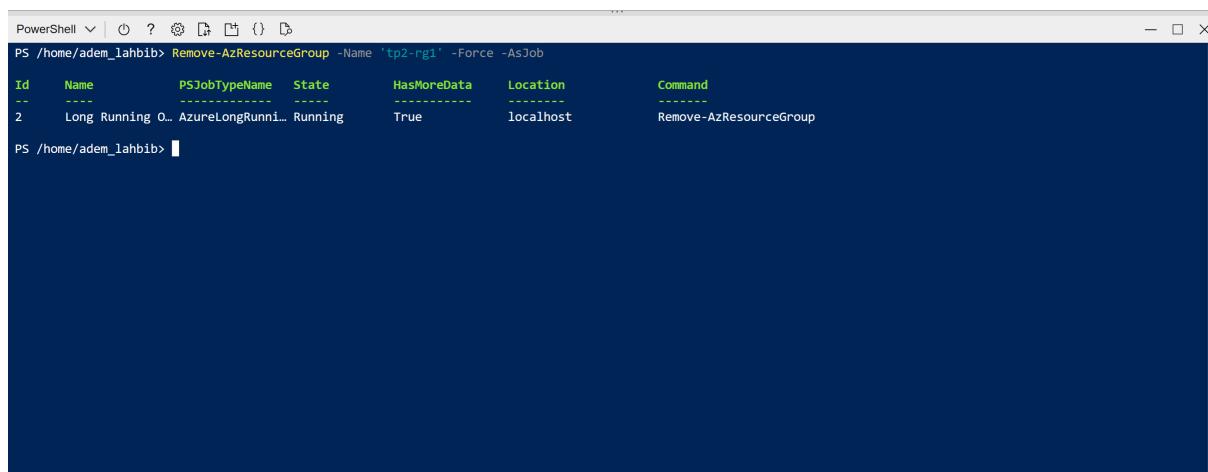
2.1.9 11. Testing vm01 Connectivity to vm02

`Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel 'Detailed'` will test the connection to the vm02 from vm01.



2.1.10 12. Cleaning up

`Remove-AzResourceGroup -Name 'tp2-rg1' -Force -AsJob` will delete the resource group tp2-rg1.



The screenshot shows the Microsoft Azure Storage accounts page. A specific storage account, "csb1003200231c147b4", is selected. On the right, a modal dialog titled "Delete storage account" is open, confirming the deletion of the selected account and its contents. The dialog lists the resource group as "cloud-shell-storage-westeurope", location as "West Europe", and subscription ID as "97cd8887-e070-4ecb-8c73-4ef77a13527". It also shows the number of instances for various resources: Containers, File shares, Tables, and Queues, all currently at zero. At the bottom, there is a text input field with the account name "csb1003200231c147b4" and two buttons: "Delete" and "Cancel".

2.2 Task 2

2.2.1 1. Uploading second definition files to Azure Cloud Shell

2.2.2 2. Creating a resource group in North Europe

2.2.3 3. Creating 3 VMs, each in a virtual network using the definition files

2.2.4 4. Installing Network Watcher extension on the VMs

```
foreach ($vmName in $vmNames) {  
    Set-AzVMExtension -ResourceGroupName $rgName -Location $location -VMName $vmName -Name 'networkWatcherAgent' -Publisher 'Microsoft.Azure.NetworkWatcher'
```

```
-Type 'NetworkWatcherAgentWindows'  
-TypeHandlerVersion '1.4'  
}  
}
```

Will install the Network Watcher extension on the VMs. The Network Watcher extension is a virtual machine extension that enables Network Watcher to monitor the network.

```
PowerShell ✓ | ⌂ ? ⌂ ⌂ ⌂ {} ⌂

    adminUsername   String           "userstudent"
    adminPassword  SecureString     null

Outputs      :
DeploymentLogLevel :

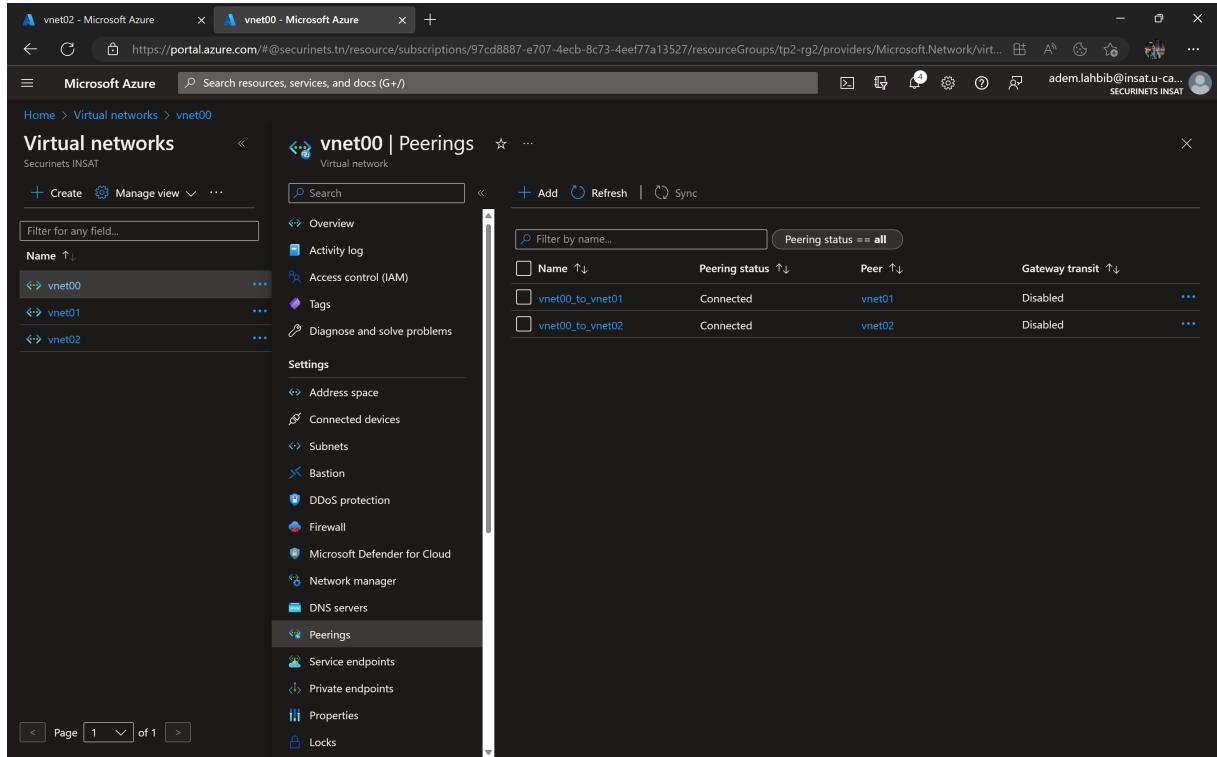

PS /home/adem_lahbib>
PS /home/adem_lahbib> $location = (Get-AzResourceGroup -ResourceGroupName $rgName).location
PS /home/adem_lahbib> $vmNames = (Get-AzVM -ResourceGroupName $rgName).Name
PS /home/adem_lahbib> foreach ($vmName in $vmNames) {
>>> Set-AzVmExtension `
>>> -ResourceGroupName $rgName ` 
>>> -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

PS /home/adem_lahbib>
```

2.2.5 5+6. Kept records of vnet01 and vnet02 resource IDs!

2.2.6 7+8. Configure peering link between vnet00 and vnet01 then vnet00 and vnet02 using Azure Portal



The screenshot shows the Azure Portal interface for managing virtual networks. The left sidebar lists 'Virtual networks' with three entries: vnet00, vnet01, and vnet02. The main content area is titled 'vnet00 | Peerings' and displays a table of existing peering connections:

Name	Peering status	Peer	Gateway transit
vnet00_to_vnet01	Connected	vnet01	Disabled
vnet00_to_vnet02	Connected	vnet02	Disabled

The 'Peering' option in the sidebar is highlighted.

The screenshot shows two separate Azure portal windows side-by-side.

Top Window (vnet01):

- Title:** vnet01 - Microsoft Azure | vnet01 - Microsoft Azure
- URL:** https://portal.azure.com/#@securinets.tn/resource/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg2/providers/Microsoft.Network/virtualNetworks/vnet01/peering
- Blade:** vnet01 | Peerings
- Table:**

Name	Peering status	Peer	Gateway transit
vnet01_to_vnet00	Connected	vnet00	Disabled

Bottom Window (vnet02):

- Title:** vnet02 - Microsoft Azure | vnet02 - Microsoft Azure
- URL:** https://portal.azure.com/#@securinets.tn/resource/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg2/providers/Microsoft.Network/virtualNetworks/vnet02/peering
- Blade:** vnet02 | Peerings
- Table:**

Name	Peering status	Peer	Gateway transit
vnet02_to_vnet00	Connected	vnet00	Disabled

2.2.7 9. Testing transitivity of vm00 to vm01 using the Network Watcher

To test the transitivity of virtual network peering, we will use the Network Watcher. The Network Watcher is a service that provides network monitoring and diagnostic capabilities for Azure virtual networks.

Transitivity is the ability of a virtual network peering to propagate the peering to other virtual networks that are connected to it. For example, if vnet01 is connected to vnet00 and vnet02 is connected to vnet01, then vnet02 will be able to communicate with vnet00.

The screenshot shows the 'Network Watcher | Connection troubleshoot' page in the Microsoft Azure portal. The configuration is as follows:

- Subscription:** Adam
- Resource group:** tp2-rg2
- Source type:** Virtual machine
- Virtual machine:** vm00
- Destination:** Select a virtual machine (radio button selected)
- URI, FQDN or IP address:** 10.61.0.4
- Probe Settings:**
 - Protocol:** TCP (selected)
 - Destination port:** 3389
- Status:** Reachable

Reachable!

2.2.8 10. Testing transitivity of vm00 to vm02 using the Network Watcher

The screenshot shows the Microsoft Azure Network Watcher Connection troubleshoot interface. The configuration is as follows:

- SOURCE:** Subscription: Adam, Resource group: tp2-rg2, Source type: Virtual machine, Virtual machine: vm00.
- DESTINATION:** URI, FQDN or IP address: 10.62.0.4.
- PROBE SETTINGS:** Protocol: TCP, Destination port: 3389.
- STATUS:** Reachable.

Reachable!

2.2.9 11. Testing transitivity of vm01 to vm02 using the Network Watcher

The screenshot shows the Azure Network Watcher Connection troubleshoot page. The left sidebar lists various monitoring and diagnostic tools. The main form is set up for a connection test between two virtual machines:

- Source:** Subscription: Adam, Resource group: tp2-rg2, Source type: Virtual machine, Virtual machine: vm01.
- Destination:** URI, FQDN or IP address: 10.62.0.4.
- Probe Settings:** Protocol: TCP, Destination port: 3389.
- Status:** Unreachable.

Not Reachable! This is due to vm02 being in different region than vm01; therefore, the peering link between vnet01 and vnet02 is not transitive.

2.2.10 12. Configuring routing between vnet01 and vnet02 by enabling IP forwarding

Routing is the process of forwarding packets from one network to another. It is the process of selecting a path for traffic in a network or between or across multiple networks.

IP forwarding means that the router forwards the packet to the next hop based on the destination IP address. It is a process in which a router forwards packets to the next network based on the destination IP address.

The screenshot shows the Azure portal interface for managing network interfaces. The left sidebar navigation includes Home, Virtual machines, vm00 | Networking, nic0, Overview, Activity log, Access control (IAM), Tags, Settings (IP configurations selected), DNS servers, Network security group, Properties, Locks, Monitoring (Insights, Alerts, Metrics, Diagnostic settings), Automation (Tasks (preview), Export template), and Help.

The main content area displays the IP configurations for the network interface nic0. It shows the following settings:

- IP forwarding settings: IP forwarding is Enabled.
- Virtual network: vnet00.
- IP configurations: Subnet is subnet0. A table lists one IP configuration:

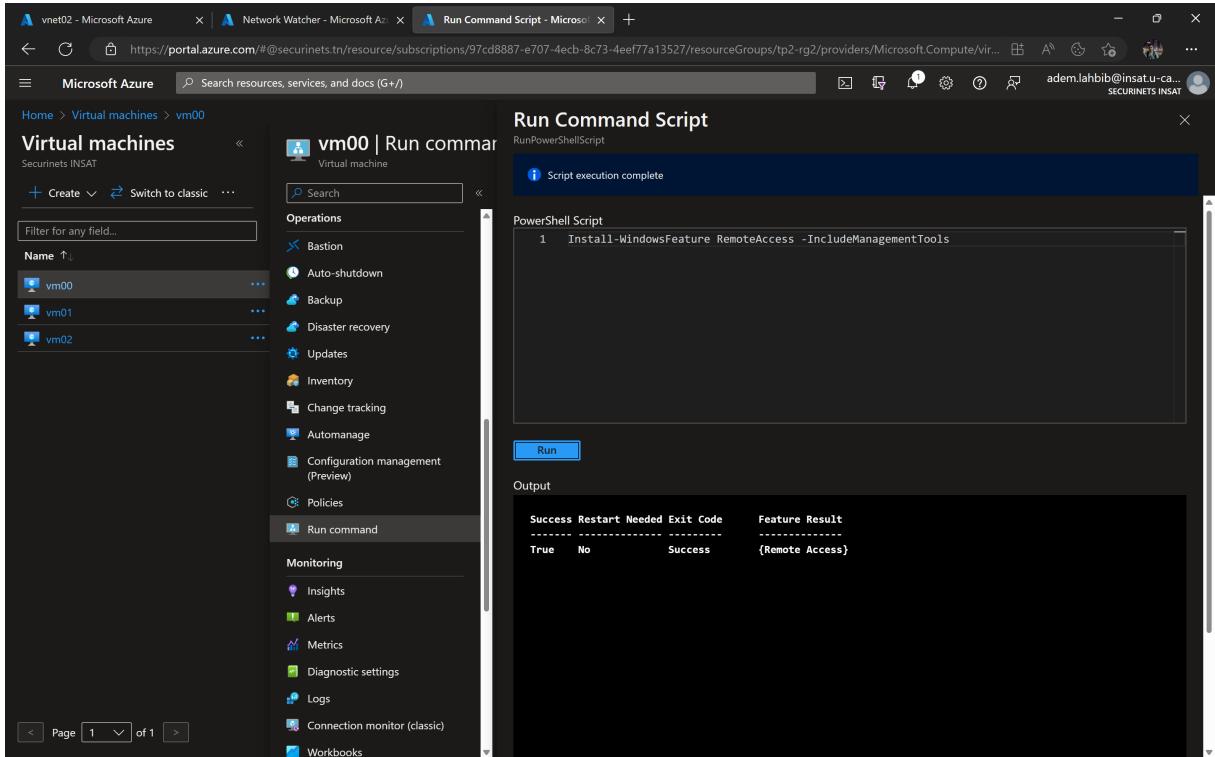
Name	IP Version	Type	Private IP address	Public IP address
ipconfig1	IPv4	Primary	10.60.0.4 (Dynamic)	-

A success message at the top right indicates: "Successfully saved IP address changes for network interface 'nic0'." The URL in the browser is https://portal.azure.com/#@securinets.tn/resource/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg2/providers/Microsoft.Network/networkInterfaces/nic0.

2.2.11 13. Installing remote access Windows server role

Remote Access is a Windows Server role that provides remote access to the network. It is a collection of services that allow users to connect to the network from a remote location.

`Install-WindowsFeature RemoteAccess -IncludeManagementTools` will install the Routing and Remote Access Service (RRAS) and the Routing and Remote Access Management Tools.



2.2.12 14. Installing routing role service

Routing role service is a Windows Server role that provides routing services. It is a collection of services that allow users to connect to the network from a remote location.

`Install-WindowsFeature -Name Routing -IncludeManagementTools -IncludeAllSubFeatures`
will install the Routing and Remote Access Service (RRAS) and the Routing and Remote Access Management Tools.

Run Command Script

RunPowerShellScript

```
1 Install-WindowsFeature -Name Routing -IncludeManagementTools -IncludeAllSubFeature
```

Run

Output

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

`Install-WindowsFeature -Name "RSAT-RemoteAccess-Powershell"` will install the Remote Access Management Tools.

Run Command Script

RunPowerShellScript

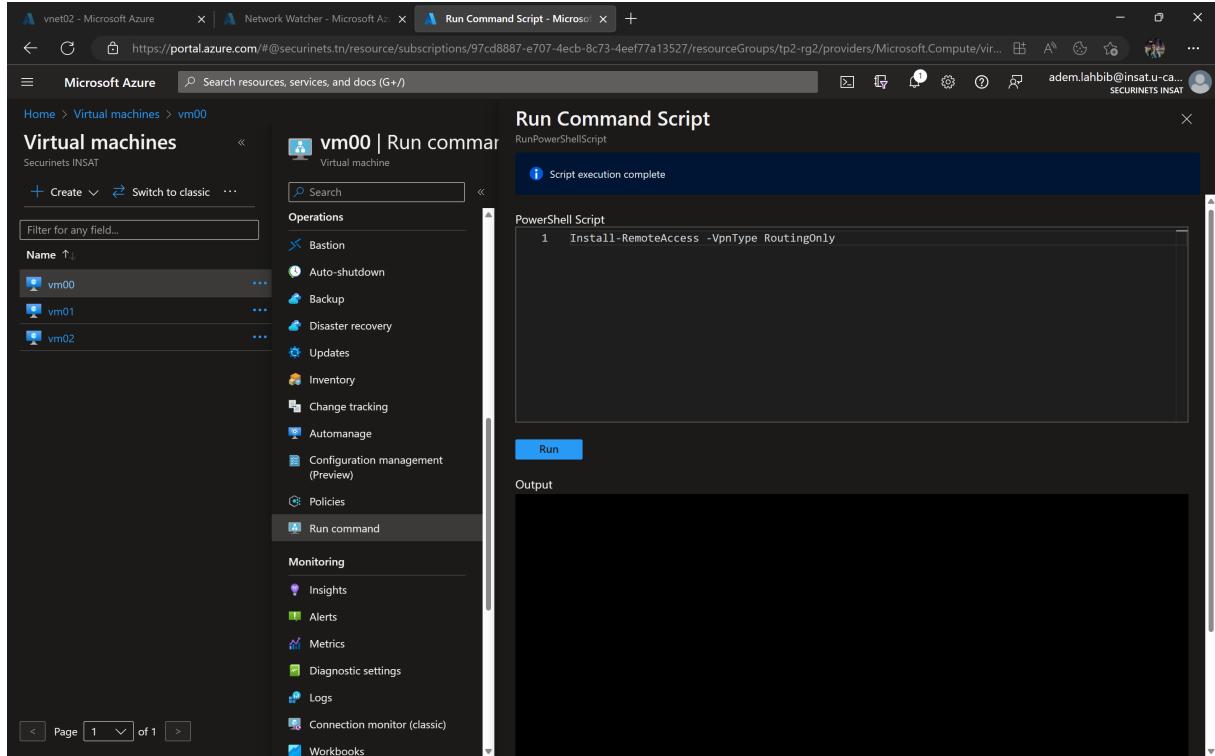
```
1 Install-WindowsFeature -Name "RSAT-RemoteAccess-Powershell"
```

Run

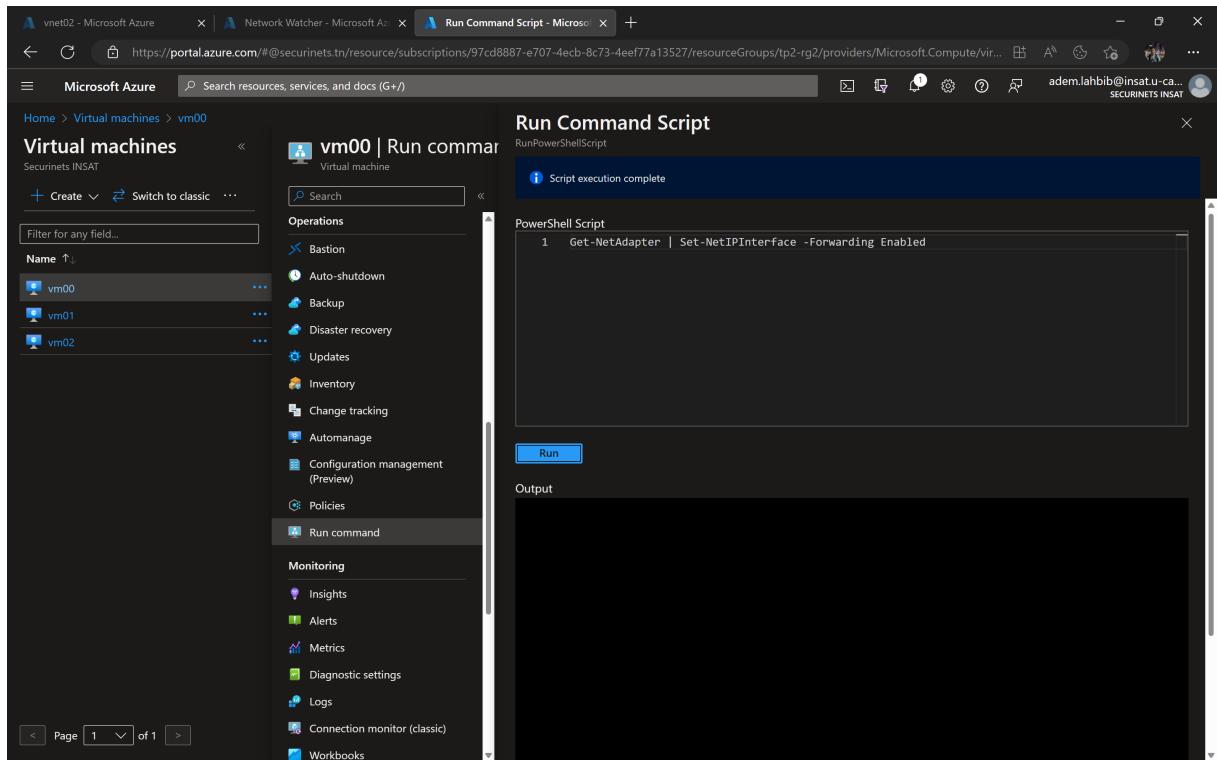
Output

Success	Restart Needed	Exit Code	Feature Result
True	No	Success	NoChangeNeeded {}

`Install-RemoteAccess -VpnType RoutingOnly` will install the Routing and Remote Access Service (RRAS) and the Routing and Remote Access Management Tools.



`Get-NetAdapter | Set-NetIPInterface -Forwarding Enabled` will enable IP forwarding.



2.2.13 15+16. Create user-defined routes on the virtual networks.

Route tables are used to route traffic between virtual networks. A route table contains a list of rules, called routes, that are used to determine where the network traffic is directed.

User-defined routes are routes that you create and manage. You can create user-defined routes to control the traffic that is routed between virtual networks.

A screenshot of the Microsoft Azure portal showing the 'route12 - Microsoft Azure' blade. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Routes, Subnets, Properties, Locks, Monitoring, Alerts, Tasks (preview), Export template, Help, Effective routes, and New Support Request. The main content area is titled 'route12' and 'Route table'. It includes sections for 'Essentials' (Resource group: tp2-rg2, Location: North Europe, Subscription: Adam, Subscription ID: 97cd8887-e707-4ecb-8c73-4eef77a13527, Tags: Click here to add tags) and 'Associations' (0 subnet associations). Below these are two tables: 'Routes' and 'Subnets'. The 'Routes' table has columns: Name, Address prefix, Next hop type, and Next hop IP address. The 'Subnets' table has columns: Name, Address range, Virtual network, and Security group. Both tables show 'No results.'

2.2.14 17. Create a route for vnet01 to vnet02

This route will allow traffic from vnet01 to vnet02.

The screenshot shows the Microsoft Azure portal interface. The left sidebar is titled 'vnet02 - Microsoft Azure' and contains a navigation tree with 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings' (selected), 'Configuration', 'Routes' (selected), 'Subnets', 'Properties', 'Locks', 'Monitoring' (with 'Alerts'), 'Automation' (with 'Tasks (preview)' and 'Export template'), 'Help' (with 'Effective routes' and 'New Support Request'), and 'Network Watcher - Microsoft Azure'. The main content area is titled 'Add route' for 'route12'. It shows a progress bar: '... Adding route' and 'Adding route 'route-vnet1-to-vnet2' to route table 'route12'...'. The form fields are: 'Route name' (set to 'route-vnet1-to-vnet2'), 'Destination address prefix' (set to 'IP Addresses'), 'Destination IP addresses/CIDR ranges' (set to '10.62.0.0/22'), 'Next hop type' (set to 'Virtual appliance'), and 'Next hop address' (set to '10.60.0.4'). A note at the bottom says: 'Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.' A blue 'Add' button is at the bottom right.

2.2.15 18. Associate the route table with the vnet01 subnet0

We need to associate the route table with the subnet in order for the route to take effect. Subnet0 is the subnet that is connected to vnet01.

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings' (with 'Configuration' selected), 'Routes', 'Subnets' (selected), 'Properties', 'Locks', 'Monitoring' (with 'Alerts' selected), 'Automation' (with 'Tasks (preview)' and 'Export template' options), 'Help' (with 'Effective routes' and 'New Support Request' options), and 'OK' at the bottom right of the dialog.

The main content area displays the 'Associate subnet' dialog. It shows the 'route12' route table and lists 'Subnets' under the 'Associate' tab. A search bar for 'Search subnets' is present. The 'Virtual network' dropdown is set to 'vnet01 (tp2-rg2)' and the 'Subnet' dropdown is set to 'subnet0'. A status message at the top right says 'Saving route table for subnet' and 'Saving route table for subnet 'subnet0'...'. The URL in the browser is https://portal.azure.com/#@securinets.tn/resource/subscriptions/97cd8887-e707-4ecb-8c73-4eef77a13527/resourceGroups/tp2-rg2/providers/Microsoft.Network/routeTables/route12.

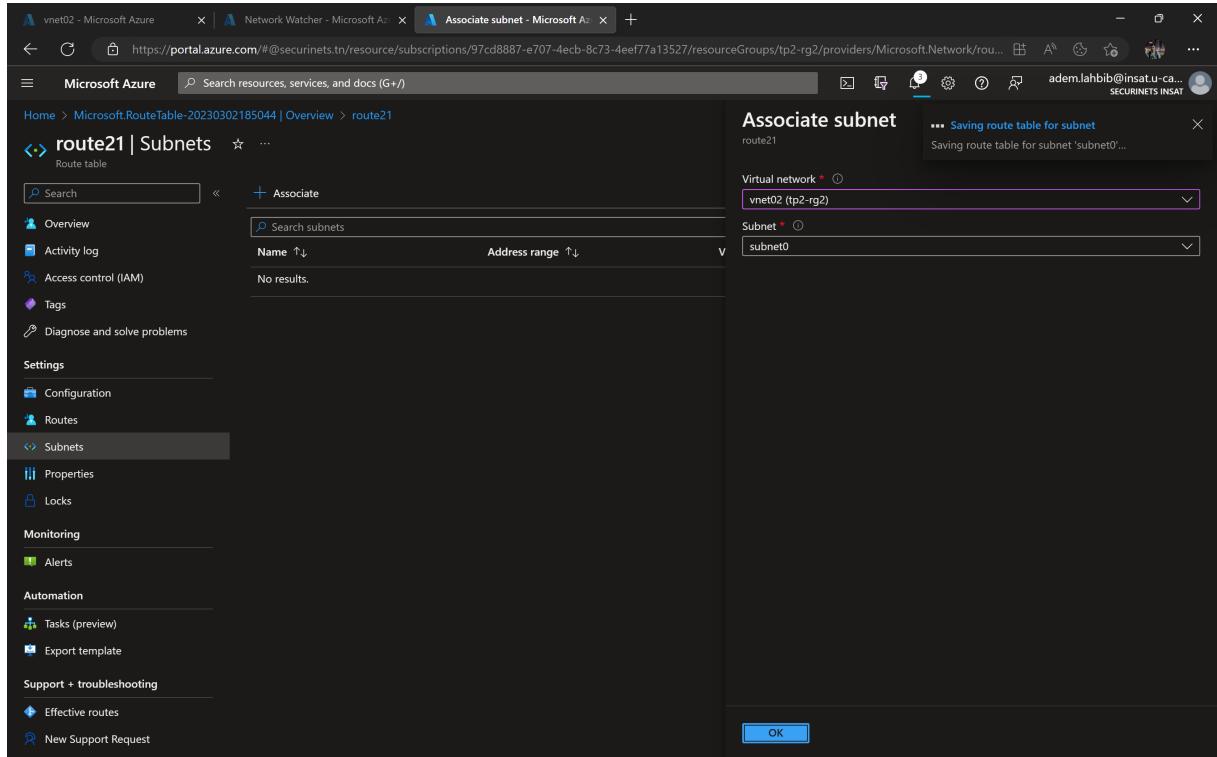
2.2.16 19. Create a route table for vnet02

The screenshot shows the Microsoft Azure portal interface with three tabs open: 'vnet02 - Microsoft Azure', 'Network Watcher - Microsoft Azure', and 'route21 - Microsoft Azure'. The 'route21' tab is active and displays the 'Overview' page for a route table named 'route21'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Configuration, Routes, Subnets, Properties, Locks, Alerts, Tasks (preview), Export template, Effective routes, and New Support Request. The main content area shows the 'Essentials' section with details: Resource group (move) : tp2-rg2, Location : North Europe, Subscription (move) : Adam, Subscription ID : 97cd8887-e707-4ecb-8c73-4eeff77a13527, and Tags (edit) : Click here to add tags. It also shows 'Associations : 0 subnet associations'. Below this is the 'Routes' section, which is currently empty (No results). The 'Subnets' section is also empty (No results).

2.2.17 20. Create a route for vnet02 to vnet01

The screenshot shows the Microsoft Azure portal interface. The left sidebar is collapsed, and the main area displays the 'route21 | Routes' page under the 'Route table' section. The 'Routes' tab is selected. On the right, a modal window titled 'Add route' is open, showing the configuration for a new route. The 'Route name' field contains 'route-vnet2-to-vnet1'. The 'Destination address prefix' field is set to 'IP Addresses' with the value '10.61.0.0/22'. The 'Next hop type' is set to 'Virtual appliance' with the value '10.60.0.4'. A note at the bottom of the modal states: 'Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.' At the bottom right of the modal is a blue 'Add' button.

2.2.18 21. Associate the route table with the vnet02 subnet0



2.2.19 22. Network Watcher test

The screenshot shows the 'Network Watcher | Connection troubleshoot' page in the Microsoft Azure portal. The left sidebar lists various monitoring and diagnostic tools. The main form is set up for a connection test between a source virtual machine ('vm01') in the 'tp2-rg2' resource group and a destination URI ('10.62.0.4'). The probe settings are configured for TCP port 3389. The status at the bottom indicates the connection is 'Reachable'.

We've got reachable this time, this is because we've enabled IP forwarding and created user-defined routes for the virtual networks.

2.2.20 23. Cleaning Up

```
PowerShell PS /home/adem_lahbib> Remove-AzResourceGroup -Name 'tp2-rg2' -Force -AsJob
PS /home/adem_lahbib>
```

A PowerShell session window showing the command to remove a resource group named 'tp2-rg2'. The command is run with the '-Force' and '-AsJob' parameters. A table below the command shows the job details:

ID	Name	PSJobTypeName	State	HasMoreData	Location	Command		
2	Long	Running	0...	AzureLongRunni...	Running	True	localhost	Remove-AzResourceGroup

The screenshot shows the Microsoft Azure portal interface. On the left, there is a list of resources under 'Storage accounts' for the 'Securinets INSAT' subscription. One resource, 'csb1003200231c147b4', is selected and has a delete icon next to it. On the right, a modal window titled 'Delete Resources' is open, asking 'Do you want to delete all the selected resources?'. A warning message states: 'Warning! Deleting the selected storage accounts is irreversible. This will permanently delete the selected storage accounts and their contents. This action cannot be undone. Do you want to continue?'. Below the warning, a 'Confirm delete' input field contains the value 'yes'. At the bottom of the modal, there are 'Delete' and 'Cancel' buttons.

3 Conclusion

In this lab, we've learned how to configure virtual network peering and how to configure routing between virtual networks. We've also learned how to use the Network Watcher to test the transitivity of virtual network peering.

These are very important concepts to understand when working with Azure virtual networks.

Lab by Pr. Imen Mami