Compte Rendu de Travaux Pratiques

Compte Rendu - Travaux Pratiques En Cloud & Virtualisation

Filière : Réseaux Informatiques & Télécommunications
Niveau : 4$^{\text{ème}}$ Année

Sujet :

TP2 : Virtual Networks

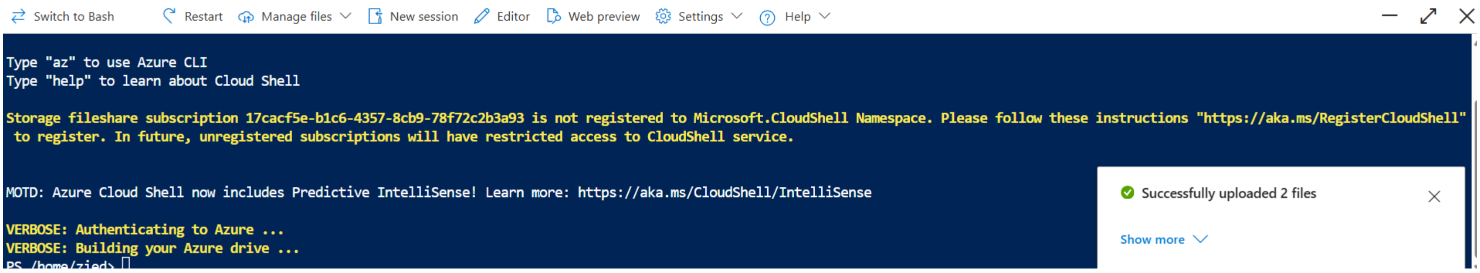
Réalisé par :

Zied KHARRAT
Nidhal JABNOUNI
Yassine BELARBI

Année Universitaire : 2024-25

TASK 01

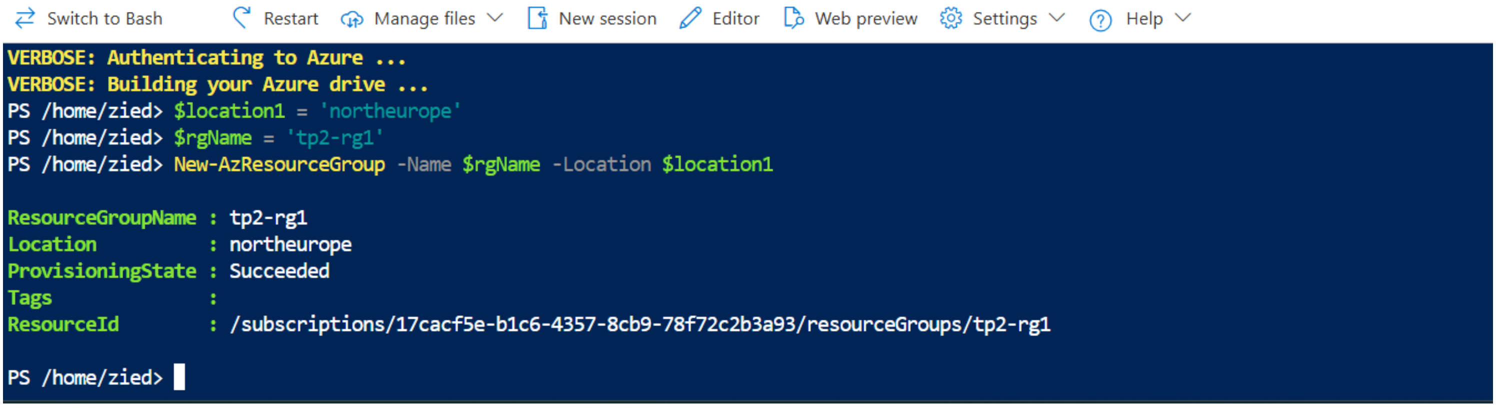
1. We have successfully uploaded the files. The first file is an ARM template that deploys Windows VMs in multiple Azure regions, sets up networking, and allows RDP access. The deploymentParameters.json file provides input values for the template. **These 2 files make configuration easier to manage and ensure consistency across all deployments (IaC).**



```
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Storage fileshare subscription 17cacf5e-b1c6-4357-8cb9-78f72c2b3a93 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell"
 to register. In future, unregistered subscriptions will have restricted access to CloudShell service.

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/zied>
```

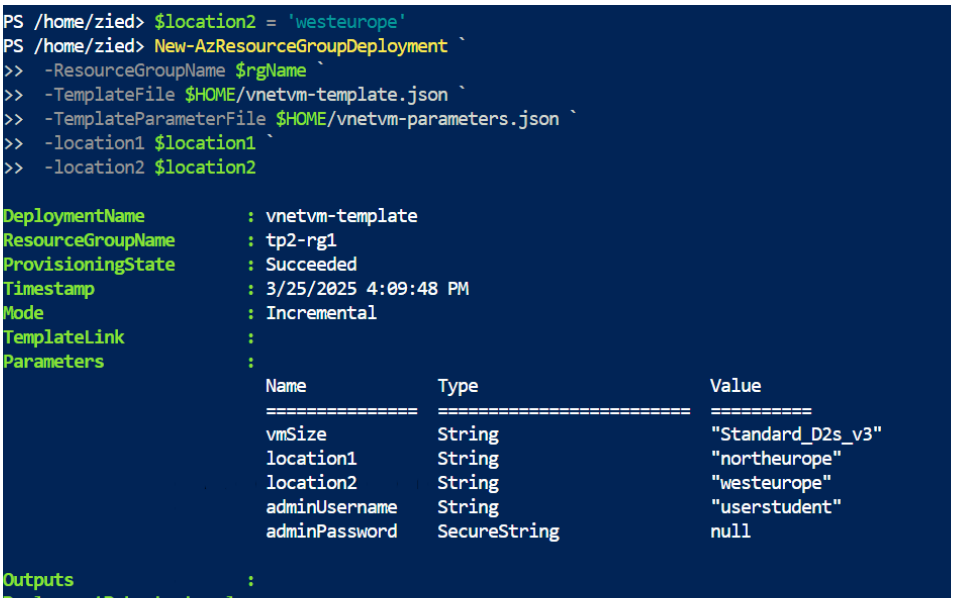
Successfully uploaded 2 files

2. We have successfully created the RG using the location and name variables we defined previously.



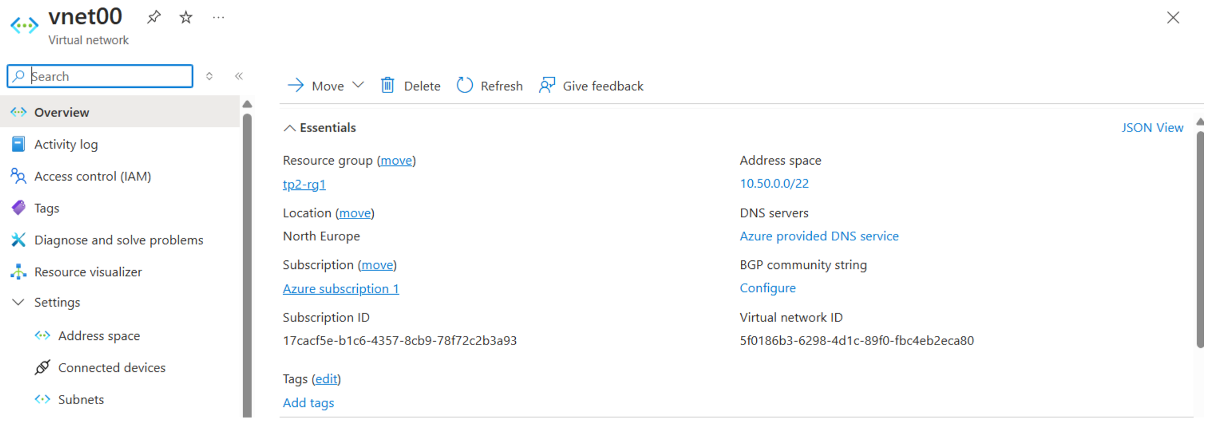
```
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/zied> $location1 = 'northeurope'
PS /home/zied> $rgName = 'tp2-rg1'
PS /home/zied> New-AzResourceGroup -Name $rgName -Location $location1

ResourceGroupName : tp2-rg1
Location          : northeurope
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/17cacf5e-b1c6-4357-8cb9-78f72c2b3a93/resourceGroups/tp2-rg1

PS /home/zied>
```

3. We have successfully deployed tbe 3 VMs to 2 different regions, **ensuring disaster recovery.**



```
PS /home/zied> $location2 = 'westeurope'
PS /home/zied> New-AzResourceGroupDeployment `
>>  -ResourceGroupName $rgName `
>>  -TemplateFile $HOME/vnetvm-template.json `
>>  -TemplateParameterFile $HOME/vnetvm-parameters.json `
>>  -location1 $location1 `
>>  -location2 $location2

DeploymentName          : vnetvm-template
ResourceGroupName       : tp2-rg1
ProvisioningState       : Succeeded
Timestamp               : 3/25/2025 4:09:48 PM
Mode                    : Incremental
TemplateLink            :
Parameters              :
                          Name             Type                       Value
                          ===============  =========================  ==========
                          vmSize           String                     "Standard_D2s_v3"
                          location1        String                     "northeurope"
                          location2        String                     "westeurope"
                          adminUsername    String                     "userstudent"
                          adminPassword    SecureString               null

Outputs                 :
```

4. We have found vnet00, this virtual network was deployed in the NE region as part of our prervious ARM template.



vnet00
Virtual network

Search

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource visualizer
- Settings
  - Address space
  - Connected devices
  - Subnets

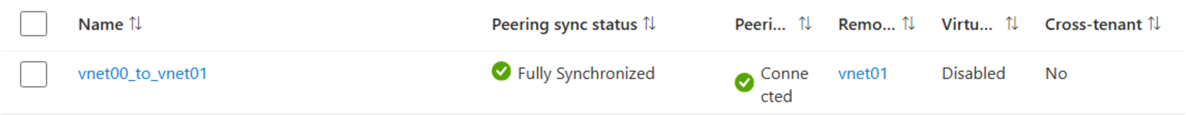
Move | Delete | Refresh | Give feedback

Essentials | JSON View

Resource group (move) tp2-rg1	Address space 10.50.0.0/22
Location (move) North Europe	DNS servers Azure provided DNS service
Subscription (move) Azure subscription 1	BGP community string Configure
Subscription ID 17cacf5e-b1c6-4357-8cb9-78f72c2b3a93	Virtual network ID 5f0186b3-6298-4d1c-89f0-fbc4eb2eca80

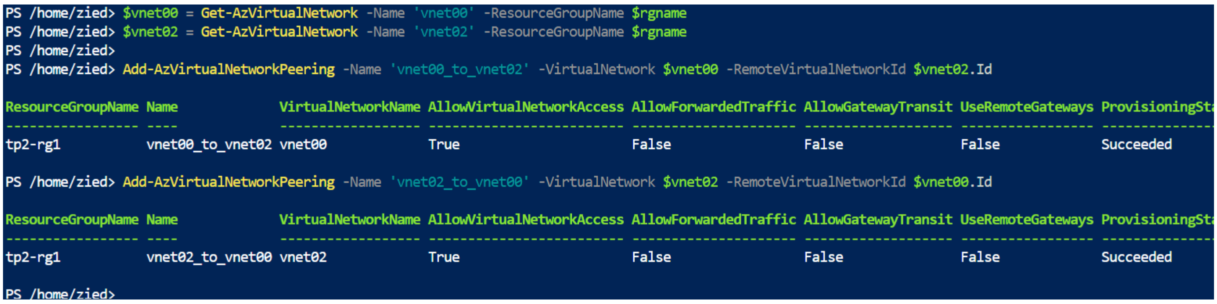
Tags (edit)
Add tags

5. We have created the peering between vnet00 and vnet01. This VNet peering connects vnet00 and vnet01, allowing direct communication between resources in both networks while blocking traffic forwarded from external sources. No virtual network gateway is used, meaning no VPN or ExpressRoute connection is involved.



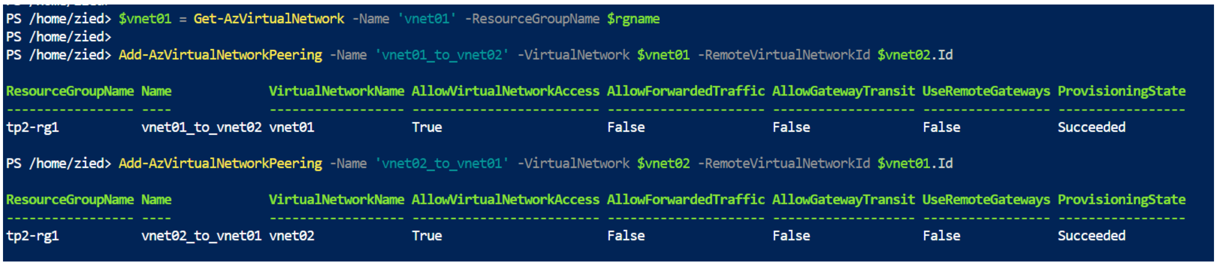
□	Name ⇅	Peering sync status ⇅	Peeri... ⇅	Remo... ⇅	Virtu... ⇅	Cross-tenant ⇅
□	vnet00_to_vnet01	✓ Fully Synchronized	✓ Connected	vnet01	Disabled	No

6. This PowerShell script establishes a VNet peering between vnet00 and vnet02, enabling direct communication between them without requiring a gateway while maintaining their separate network policies.



```
PS /home/zied> $vnet00 = Get-AzVirtualNetwork -Name 'vnet00' -ResourceGroupName $rgname
PS /home/zied> $vnet02 = Get-AzVirtualNetwork -Name 'vnet02' -ResourceGroupName $rgname
PS /home/zied>
PS /home/zied> Add-AzVirtualNetworkPeering -Name 'vnet00_to_vnet02' -VirtualNetwork $vnet00 -RemoteVirtualNetworkId $vnet02.Id

ResourceGroupName Name             VirtualNetworkName AllowVirtualNetworkAccess AllowForwardedTraffic AllowGatewayTransit UseRemoteGateways ProvisioningSta
----------------- ----             ------------------ ------------------------- --------------------- ------------------- ----------------- ---------------
tp2-rg1           vnet00_to_vnet02 vnet00             True                      False                 False               False             Succeeded

PS /home/zied> Add-AzVirtualNetworkPeering -Name 'vnet02_to_vnet00' -VirtualNetwork $vnet02 -RemoteVirtualNetworkId $vnet00.Id

ResourceGroupName Name             VirtualNetworkName AllowVirtualNetworkAccess AllowForwardedTraffic AllowGatewayTransit UseRemoteGateways ProvisioningSta
----------------- ----             ------------------ ------------------------- --------------------- ------------------- ----------------- ---------------
tp2-rg1           vnet02_to_vnet00 vnet02             True                      False                 False               False             Succeeded

PS /home/zied>
```

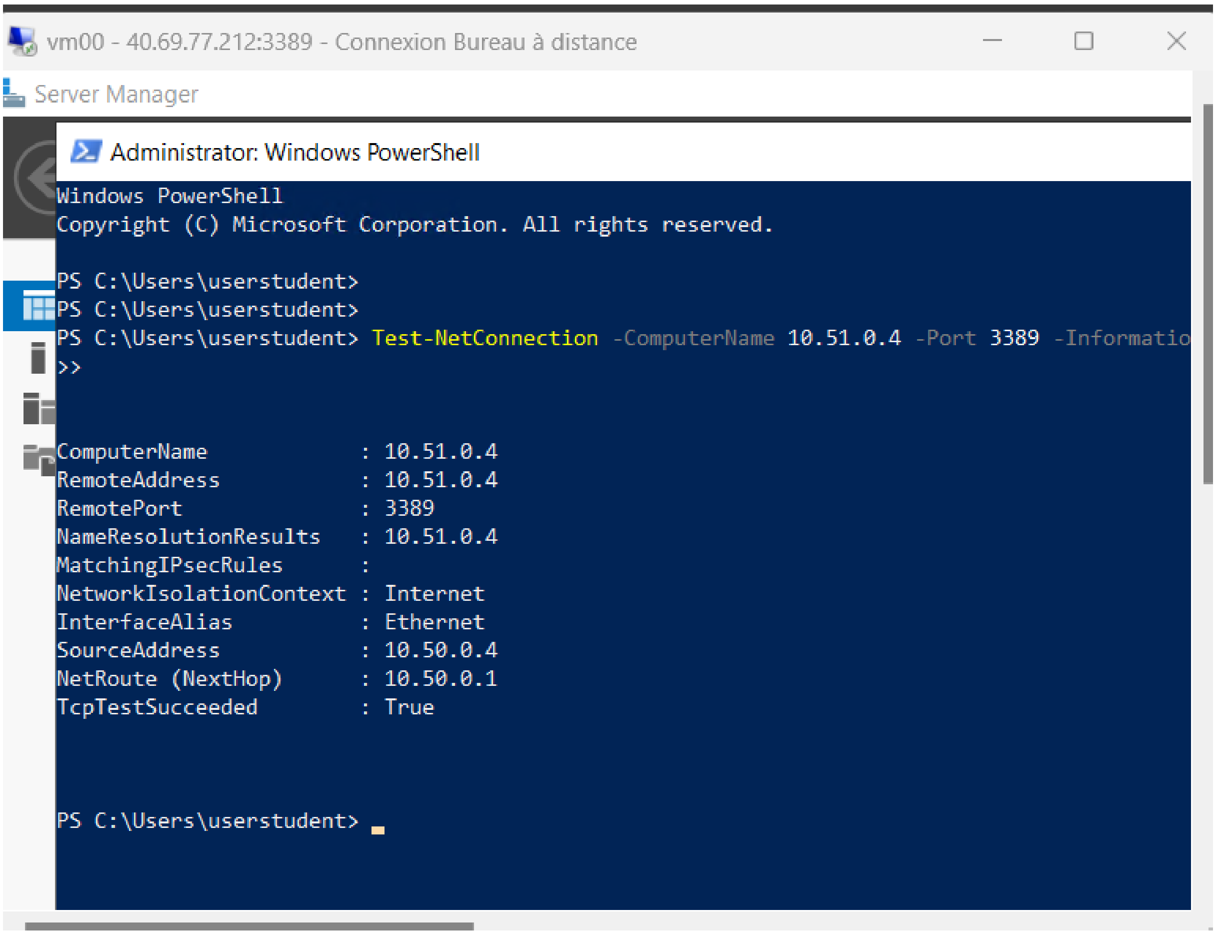
7. Similarly, this script configures a VNet peering between vnet01 and vnet02, allowing seamless connectivity between resources in both networks while keeping traffic isolation rules in place.



```
PS /home/zied> $vnet01 = Get-AzVirtualNetwork -Name 'vnet01' -ResourceGroupName $rgname
PS /home/zied>
PS /home/zied> Add-AzVirtualNetworkPeering -Name 'vnet01_to_vnet02' -VirtualNetwork $vnet01 -RemoteVirtualNetworkId $vnet02.Id

ResourceGroupName Name             VirtualNetworkName AllowVirtualNetworkAccess AllowForwardedTraffic AllowGatewayTransit UseRemoteGateways ProvisioningState
----------------- ----             ------------------ ------------------------- --------------------- ------------------- ----------------- -----------------
tp2-rg1           vnet01_to_vnet02 vnet01             True                      False                 False               False             Succeeded

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

ResourceGroupName Name             VirtualNetworkName AllowVirtualNetworkAccess AllowForwardedTraffic AllowGatewayTransit UseRemoteGateways ProvisioningState
----------------- ----             ------------------ ------------------------- --------------------- ------------------- ----------------- -----------------
tp2-rg1           vnet02_to_vnet01 vnet02             True                      False                 False               False             Succeeded
```

8. This step verifies that the VNet peering is working by connecting to vm00 via RDP, ensuring that network communication is possible.

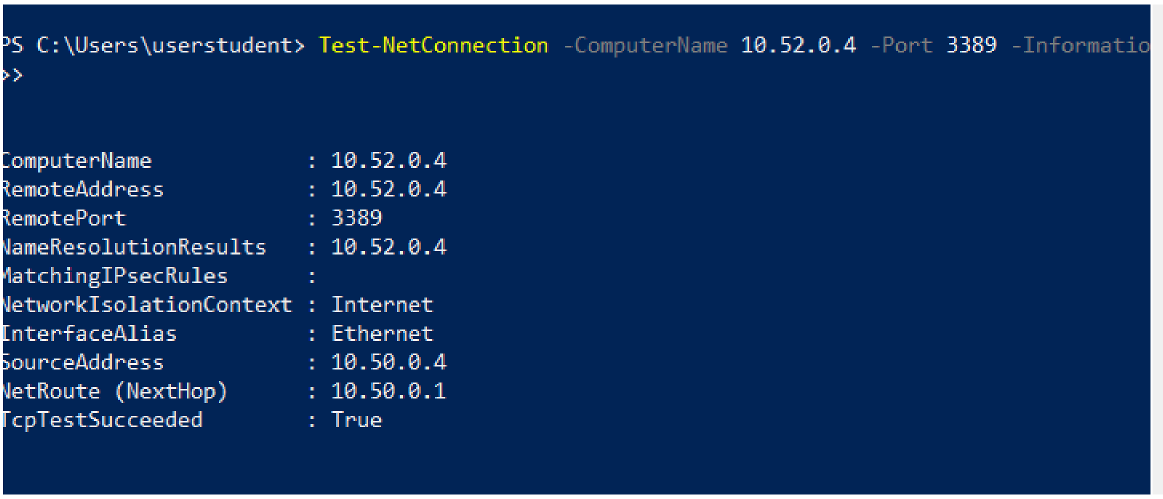
9. Running the Test-NetConnection command checks if vm00 can reach vm01 over TCP port 3389 (RDP), **confirming that the peering allows remote desktop connectivity between the VMs.**



vm00 - 40.69.77.212:3389 - Connexion Bureau à distance
Server Manager
Administrator: Windows PowerShell

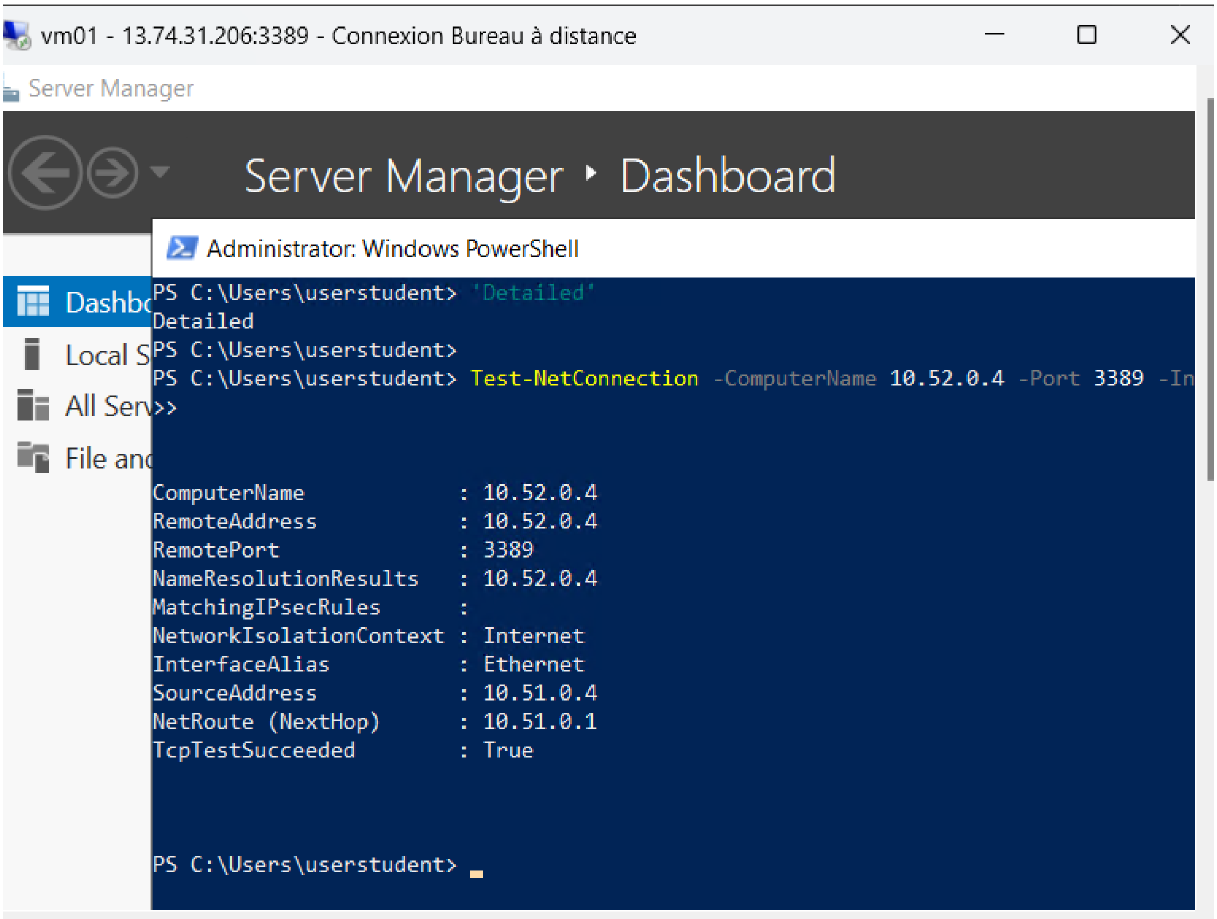
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\userstudent>
PS C:\Users\userstudent>
PS C:\Users\userstudent> Test-NetConnection -ComputerName 10.51.0.4 -Port 3389 -Informatio
>>


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>
```

10. vm00 can also connect to vm02 (test successful), confirming that the peering allows remote desktop connectivity between the VMs.



```
PS C:\Users\userstudent> Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -Informatio
>>


ComputerName           : 10.52.0.4
RemoteAddress          : 10.52.0.4
RemotePort             : 3389
NameResolutionResults  : 10.52.0.4
MatchingIPsecRules     :
NetworkIsolationContext : Internet
InterfaceAlias         : Ethernet
SourceAddress          : 10.50.0.4
NetRoute (NextHop)     : 10.50.0.1
TcpTestSucceeded       : True
```

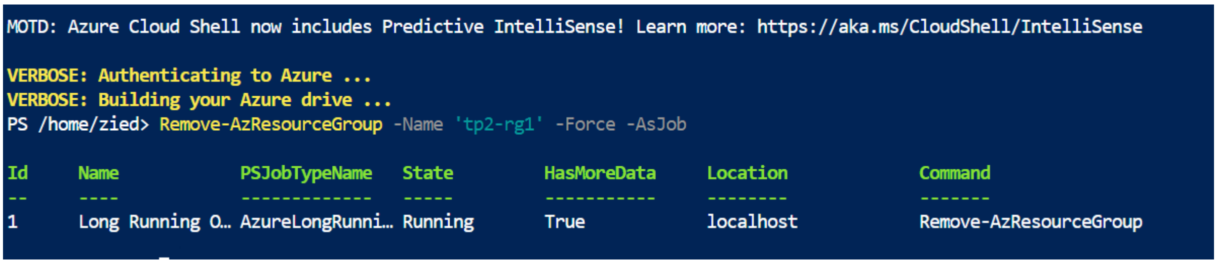
11. vm01 can connect to vm02 (test successful) confirming that the peering allows remote desktop connectivity between the VMs.



vm01 - 13.74.31.206:3389 - Connexion Bureau à distance
Server Manager
Server Manager ‣ Dashboard
Administrator: Windows PowerShell
Dashbo
Local S
All Serv
File and

```
PS C:\Users\userstudent> 'Detailed'
Detailed
PS C:\Users\userstudent>
PS C:\Users\userstudent> Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -In
>>


ComputerName            : 10.52.0.4
RemoteAddress           : 10.52.0.4
RemotePort              : 3389
NameResolutionResults   : 10.52.0.4
MatchingIPsecRules      :
NetworkIsolationContext : Internet
InterfaceAlias          : Ethernet
SourceAddress           : 10.51.0.4
NetRoute (NextHop)      : 10.51.0.1
TcpTestSucceeded        : True



PS C:\Users\userstudent> _
```

12. We successfully deleted the resource group.



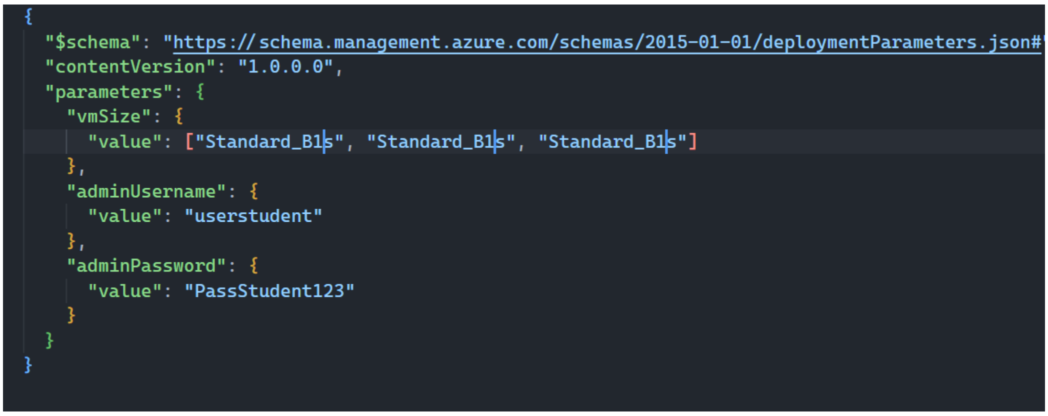
```
MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/zied> Remove-AzResourceGroup -Name 'tp2-rg1' -Force -AsJob

Id     Name            PSJobTypeName   State         HasMoreData     Location             Command
--     ----            -------------   -----         -----------     --------             -------
1      Long Running O… AzureLongRunni… Running       True            localhost            Remove-AzResourceGroup
```

Task 1 Conclusion:

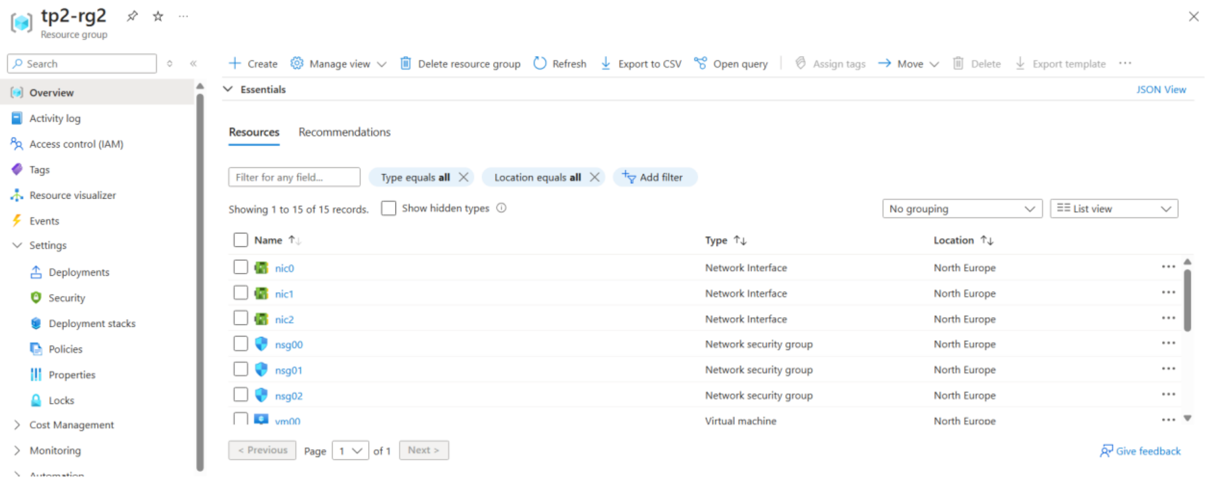
During this task, we created three virtual machines in separate virtual networks across two Azure regions and established network peerings between them. Using PowerShell commands and ARM templates, **we configured the networks and tested connectivity** between the virtual machines using their private IP addresses. This exercise provided **practical experience in setting up Azure virtual networks,** implementing network peering, and verifying inter-network communication.

TASK 02

1/2/3. We managed to deploy the resource group correctly; however, we had to change the VMs' sizes to **Standard_B1s** to comply with the quota limits (we initially got an error saying that we exceeded the quota of 4 cores defined by the free plan). This reduced the VMs' core count from **2 to 1** and RAM from **4GB to 1GB**.



```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmSize": {
      "value": ["Standard_B1s", "Standard_B1s", "Standard_B1s"]
    },
    "adminUsername": {
      "value": "userstudent"
    },
    "adminPassword": {
      "value": "PassStudent123"
    }
  }
}
```

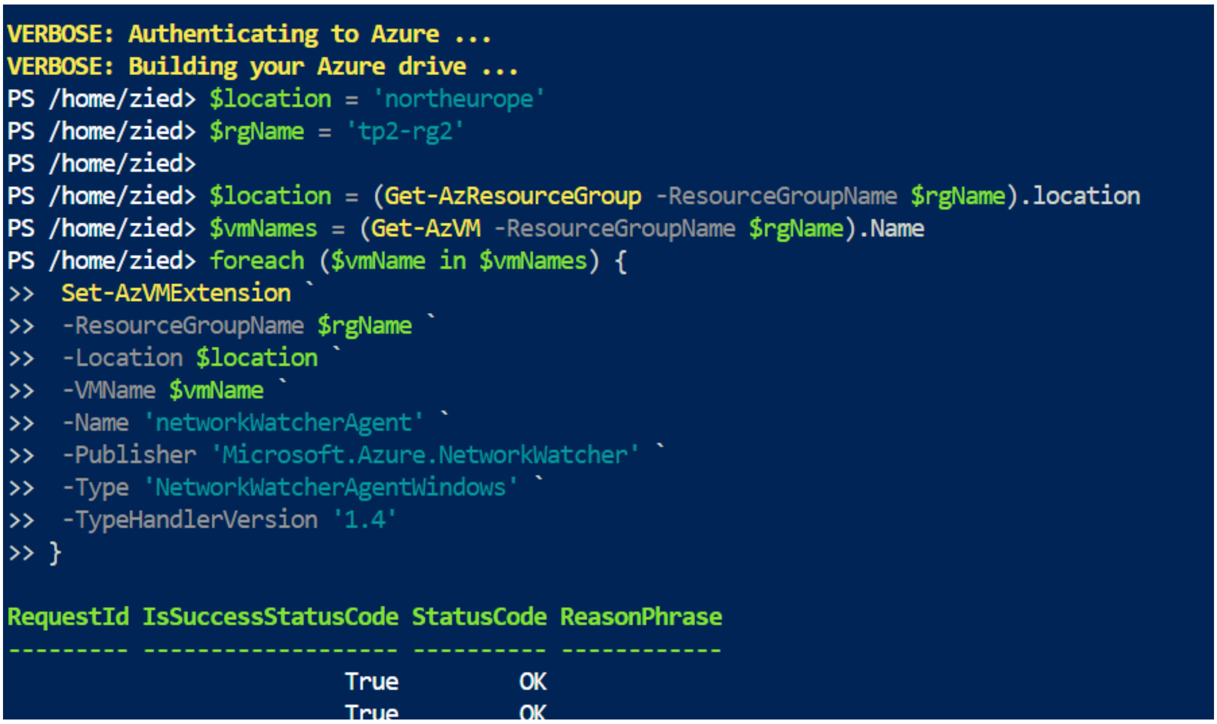


Showing 1 to 15 of 15 records. □ Show hidden types

□ Name	Type	Location
□ nic0	Network Interface	North Europe
□ nic1	Network Interface	North Europe
□ nic2	Network Interface	North Europe
□ nsg00	Network security group	North Europe
□ nsg01	Network security group	North Europe
□ nsg02	Network security group	North Europe
□ vm00	Virtual machine	North Europe

Page 1 of 1

4. We successfully added the **Network Watcher extension** to all VMs by looping through them and installing the **NetworkWatcherAgent**, which enables network monitoring, traffic analytics, and diagnostics for troubleshooting connectivity issues.

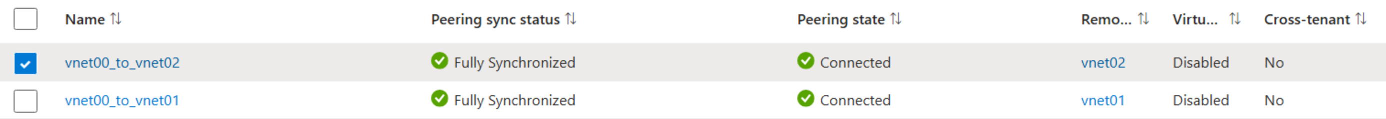


```
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/zied> $location = 'northeurope'
PS /home/zied> $rgName = 'tp2-rg2'
PS /home/zied>
PS /home/zied> $location = (Get-AzResourceGroup -ResourceGroupName $rgName).location
PS /home/zied> $vmNames = (Get-AzVM -ResourceGroupName $rgName).Name
PS /home/zied> foreach ($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
                         True         OK
```

5. The value is 600aece1-d932-4a8d-9ed1-badb37a63fe5

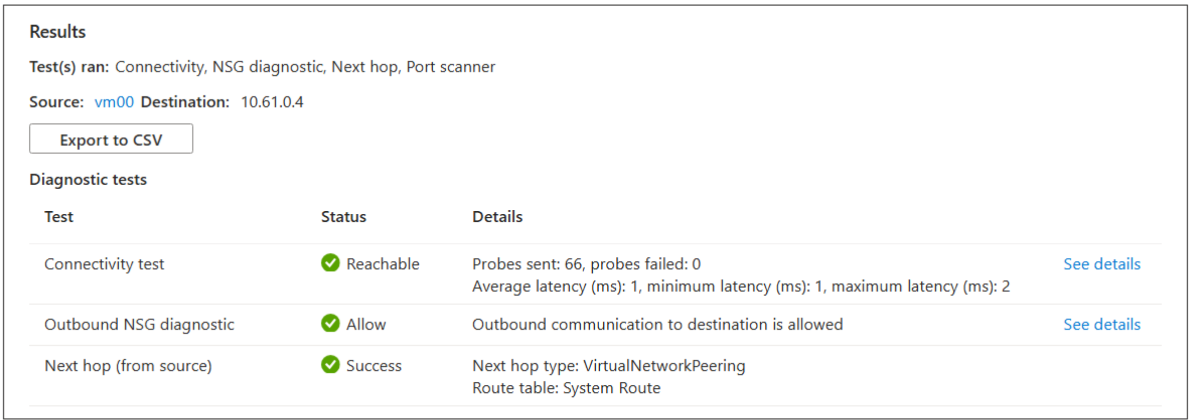
6. The value is 02e9d5a1-1587-49a9-a44a-0c0e41b7d6dd

7/8. We successfully established **virtual network peering** between **vnet00, vnet01, and vnet02**, allowing seamless communication between these networks while enforcing specific traffic rules. **Task 7** connects **vnet00 to vnet01**, enabling bidirectional traffic while blocking forwarded traffic from external sources. **Task 8** links **vnet00 to vnet02**, following the same rules, ensuring controlled connectivity between networks without requiring a VPN or public internet, since we do not require a VN gateway.



□	Name ↑↓	Peering sync status ↑↓	Peering state ↑↓	Remo... ↑↓	Virtu... ↑↓	Cross-tenant ↑↓
☑	vnet00_to_vnet02	✓ Fully Synchronized	✓ Connected	vnet02	Disabled	No
□	vnet00_to_vnet01	✓ Fully Synchronized	✓ Connected	vnet01	Disabled	No

9/10. The successful test confirms that **vm00 can directly reach both vm01 (10.61.0.4) and vm02 (10.62.0.4) over TCP port 3389**, indicating that the virtual network peering configurations are correctly set up and allowing communication as expected. **However, this does not confirm transitive peering unless an explicit route or additional peering connections enable indirect communication between VNets.**



Results

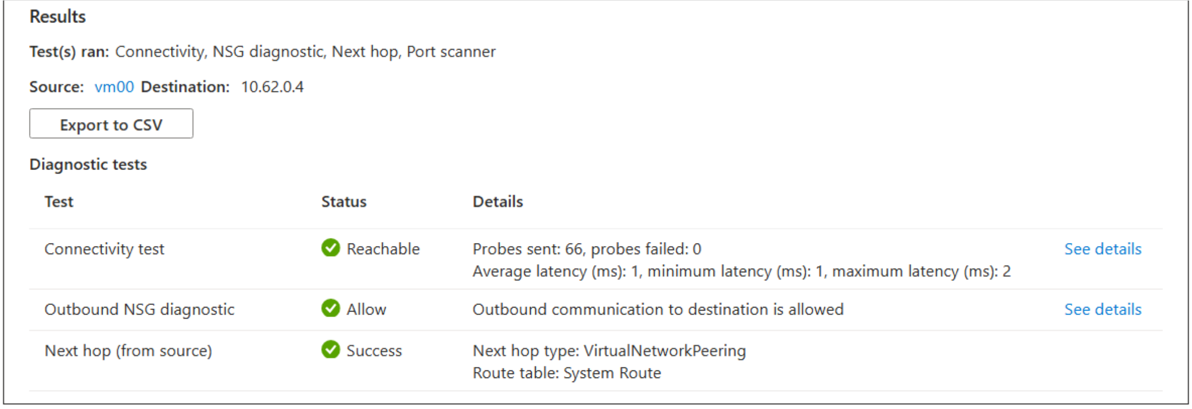
Test(s) ran: Connectivity, NSG diagnostic, Next hop, Port scanner

Source: vm00 **Destination:** 10.61.0.4

Export to CSV

Diagnostic tests

Test	Status	Details	
Connectivity test	✓ Reachable	Probes sent: 66, probes failed: 0 Average latency (ms): 1, minimum latency (ms): 1, maximum latency (ms): 2	See details
Outbound NSG diagnostic	✓ Allow	Outbound communication to destination is allowed	See details
Next hop (from source)	✓ Success	Next hop type: VirtualNetworkPeering Route table: System Route	



Results

Test(s) ran: Connectivity, NSG diagnostic, Next hop, Port scanner

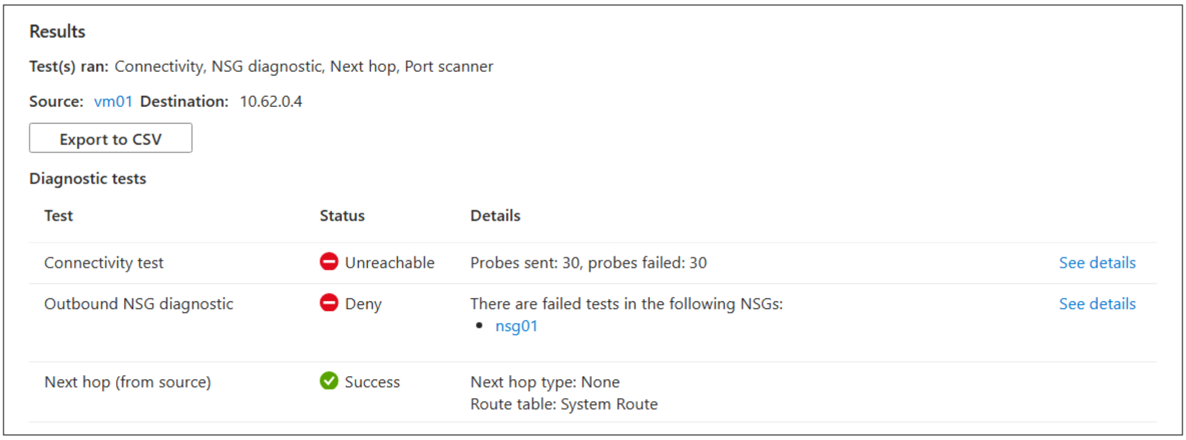
Source: vm00 **Destination:** 10.62.0.4

Export to CSV

Diagnostic tests

Test	Status	Details	
Connectivity test	✓ Reachable	Probes sent: 66, probes failed: 0 Average latency (ms): 1, minimum latency (ms): 1, maximum latency (ms): 2	See details
Outbound NSG diagnostic	✓ Allow	Outbound communication to destination is allowed	See details
Next hop (from source)	✓ Success	Next hop type: VirtualNetworkPeering Route table: System Route	

11. The connectivity test returned unreachable because the two virtual networks are not peered together (peering is not transitive, as was previously stated).



Results

Test(s) ran: Connectivity, NSG diagnostic, Next hop, Port scanner

Source: vm01 **Destination:** 10.62.0.4

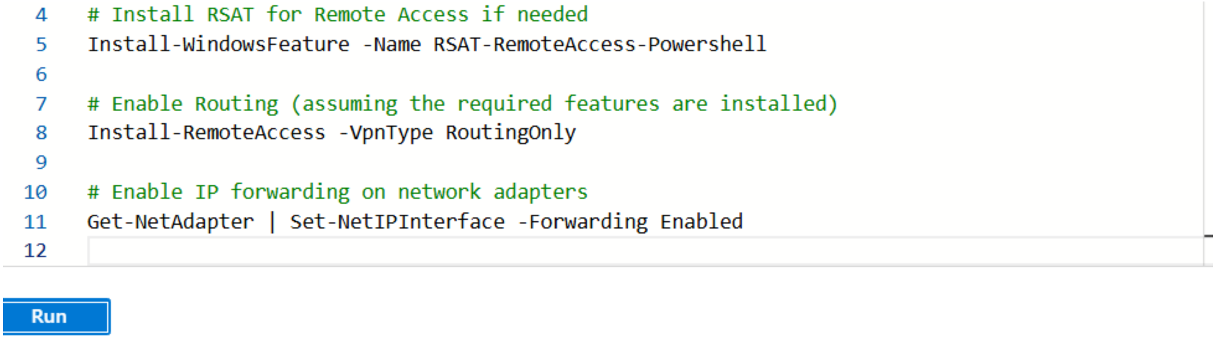
Export to CSV

Diagnostic tests

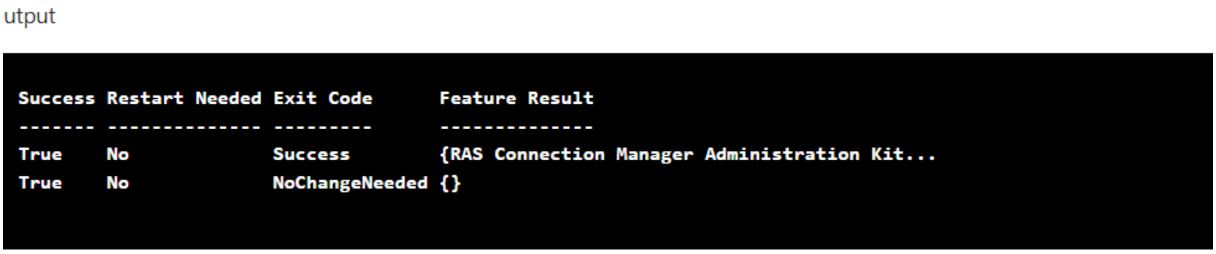
Test	Status	Details	
Connectivity test	⛔ Unreachable	Probes sent: 30, probes failed: 30	See details
Outbound NSG diagnostic	⛔ Deny	There are failed tests in the following NSGs: • nsg01	See details
Next hop (from source)	✓ Success	Next hop type: None Route table: System Route	

12/13. Enabling **IP forwarding** on vm00 allows it to route traffic between different networks, which is essential for acting as a network appliance (e.g., a router or firewall). Together, these steps ensure vm00 can effectively forward packets between VNets.

14. We successfully installed RSAT and enabled routing/IP forwarding, this means **that vm00 is now configured to route network traffic between virtual networks vnet01 and vnet02, acting as a gateway.**



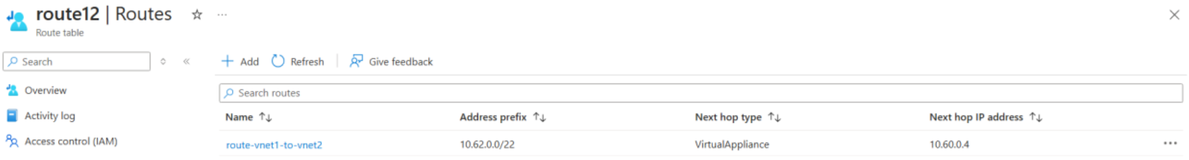
```
# Install RSAT for Remote Access if needed
Install-WindowsFeature -Name RSAT-RemoteAccess-Powershell

# Enable Routing (assuming the required features are installed)
Install-RemoteAccess -VpnType RoutingOnly

# Enable IP forwarding on network adapters
Get-NetAdapter | Set-NetIPInterface -Forwarding Enabled

```



utput

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

15/16/17. We successfully created a routing table (route12) and defined a custom route (route-vnet1-to-vnet2) to direct traffic from vnet1 to vnet2 through a virtual appliance at 10.60.0.4. This ensures controlled traffic flow between networks without relying on default system routes.

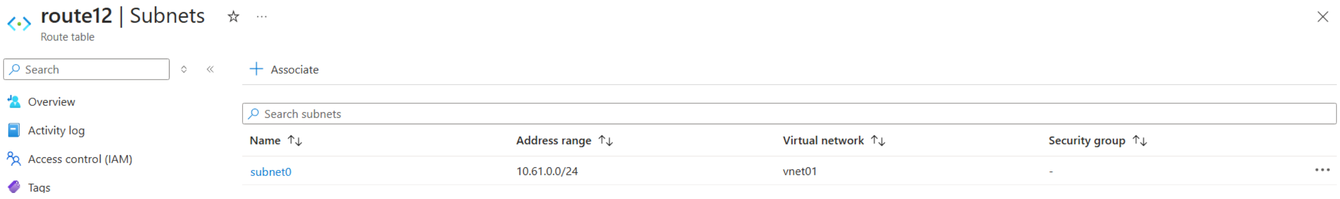


route12 | Routes

Route table

Name	Address prefix	Next hop type	Next hop IP address
route-vnet1-to-vnet2	10.62.0.0/22	VirtualAppliance	10.60.0.4

18. We successfully associated the route12 route table with subnet0 in vnet01, ensuring that all traffic from this subnet follows the custom routing rules defined in route12, such as directing traffic to vnet2 via the virtual appliance.

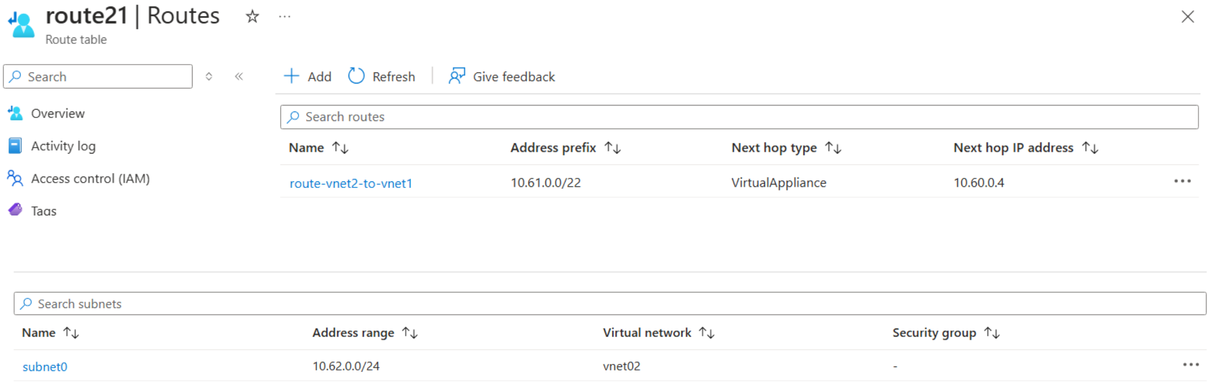


route12 | Subnets

Route table

Name	Address range	Virtual network	Security group
subnet0	10.61.0.0/24	vnet01	-

19/20/21. We did the same for route 21 and vnet02 with subnet0, ensuring that all traffic from this subnet follows the custom routing rules defined in route21, such as directing traffic to vnet2 via the virtual appliance.



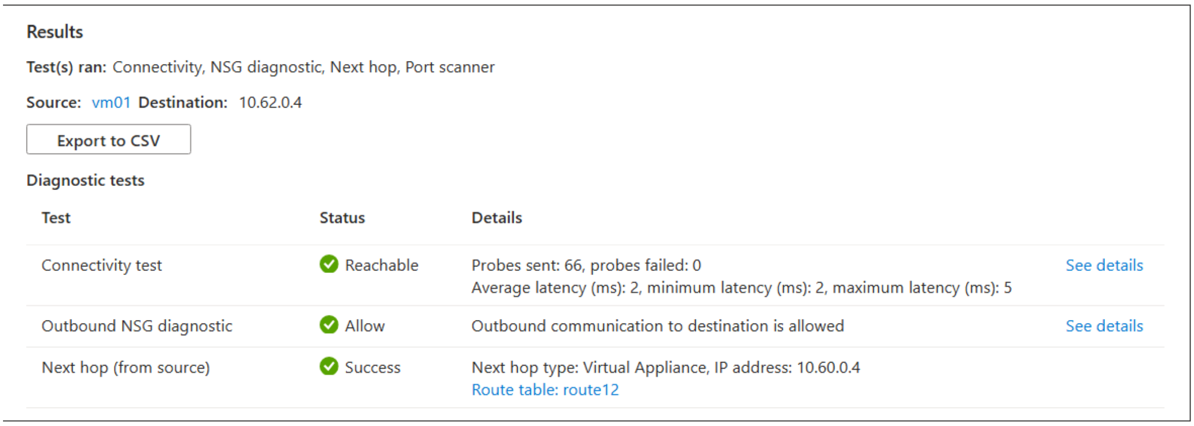
route21 | Routes

Route table

Name	Address prefix	Next hop type	Next hop IP address
route-vnet2-to-vnet1	10.61.0.0/22	VirtualAppliance	10.60.0.4

Name	Address range	Virtual network	Security group
subnet0	10.62.0.0/24	vnet02	-

22. We successfully verified network connectivity from vm01 to 10.62.0.4 over TCP port 3389, confirming that the configured routing and peering settings allow seamless communication between the virtual networks.



Results

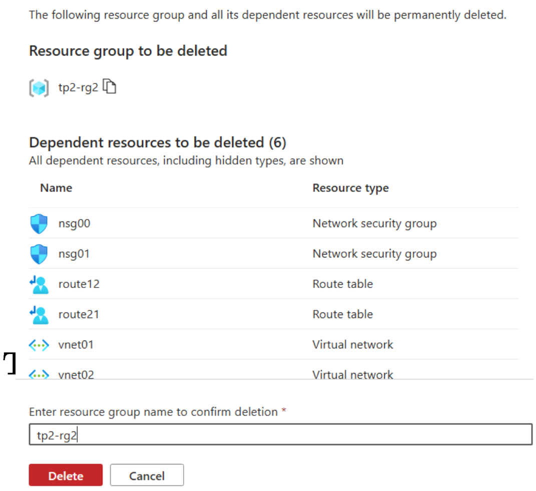
Test(s) ran: Connectivity, NSG diagnostic, Next hop, Port scanner

Source: vm01 **Destination:** 10.62.0.4

Diagnostic tests

Test	Status	Details	
Connectivity test	✓ Reachable	Probes sent: 66, probes failed: 0 Average latency (ms): 2, minimum latency (ms): 2, maximum latency (ms): 5	See details
Outbound NSG diagnostic	✓ Allow	Outbound communication to destination is allowed	See details
Next hop (from source)	✓ Success	Next hop type: Virtual Appliance, IP address: 10.60.0.4 Route table: route12	

23. Finally, we cleaned up by deleting the resource group tp2-rg2.



The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

tp2-rg2

Dependent resources to be deleted (6)

All dependent resources, including hidden types, are shown

Name	Resource type
nsg00	Network security group
nsg01	Network security group
route12	Route table
route21	Route table
vnet01	Virtual network
vnet02	Virtual network

Enter resource group name to confirm deletion *

tp2-rg2

In this task, we deployed three virtual machines in separate virtual networks within the same Azure region and configured network peering between them. We used the Network Watcher tool to **test connectivity** and verified that direct communication was possible only between peered networks. **To enable transitive connectivity, we configured routing by enabling IP forwarding, installing routing services, and defining custom route tables**. This task provided hands-on experience in setting up virtual network peering, troubleshooting connectivity, and implementing user-defined routing in Azure.

Overall Conclusion:

Throughout this lab, we worked with Azure virtual networks, virtual machines, and network peering configurations. Task 1 focused on deploying virtual machines across different Azure regions and establishing network peering, while Task 2 expanded on this by implementing routing for transitive connectivity. By completing these exercises, we gained practical knowledge of deploying cloud infrastructure, managing network connectivity, and troubleshooting network communication issues in Azure.

Interpretation:

This lab provided hands-on exposure to essential cloud infrastructure concepts within Azure, such as setting up and managing virtual networks and virtual machines. The exercises not only helped in understanding how to deploy resources across different regions but also highlighted the importance of network connectivity through peering and routing. By troubleshooting network issues, the lab reinforced the need for effective configuration and monitoring in cloud environments, offering practical skills that are crucial for cloud infrastructure management and network optimization.