

# Lab 05 - Implement Intersite Connectivity

## Task 1: Provision the lab environment

We upload the files `\Allfiles\Labs\05\az104-05-vnetvm-loop-template.json` and `\Allfiles\Labs\05\az104-05-vnetvm-loop-parameters.json` into the Cloud Shell home directory

```
-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
```

Upon listing the available locations, I've chosen 'eastus' and 'westus2' as the locations for my lab provisioning. However, 'westus2' wasn't available for deploying the virtual networks, so eventually, I had to change it to 'westus'.

```
PS /home/andrijana> $location1 = 'eastus'
PS /home/andrijana> $location2 = 'westus2'
PS /home/andrijana> $rgName = 'az104-05-rg1'
PS /home/andrijana> New-AzResourceGroup -Name $rgName -Location $location1
```

```
ResourceGroupName : az104-05-rg1
Location           : eastus
ProvisioningState  : Succeeded
Tags               :
ResourceId         : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1
```

```
New-AzResourceGroupDeployment: 10:45:21 PM - Error: Code=InvalidTemplateDeployment; Message=The template deployment 'az104-05-vnetvm-loop-template' is not valid according to the validation procedure. The tracking id is '06508cef-4df3-4e43-b0d3-c6f88ca71303'. See inner errors for details.
New-AzResourceGroupDeployment: 10:45:21 PM - Error: Code=ShowNotAvailable; Message=The requested VM size for resource 'Following SKUs have failed for Capacity Restrictions: Standard_E2s_v3' is currently not available in location 'westus2'. Please try another size or deploy to a different zone. See https://aka.ms/azureskumetaavailable for details.
New-AzResourceGroupDeployment: The deployment validation failed
```

Successful creation of the Virtual Networks:

```
PS /home/andrijana> $location2 = 'westus'
PS /home/andrijana> New-AzResourceGroupDeployment `
>> -ResourceGroupName $rgName `
>> -TemplateFile $HOME/az104-05-vnetvm-loop-template.json `
>> -TemplateParameterFile $HOME/az104-05-vnetvm-loop-parameters.json `
>> -location1 $location1 `
>> -location2 $location2

DeploymentName      : az104-05-vnetvm-loop-template
ResourceGroupName  : az104-05-rg1
ProvisioningState   : Succeeded
Timestamp          : 3/23/2023 10:46:55 PM
Mode               : Incremental
TemplateLink       :
Parameters         :
                    Name                Type                Value
                    =====
vmSize             String              "Standard_D2s_v3"
location1          String              "eastus"
location2          String              "westus"
adminUsername      String              "Student"
adminPassword      SecureString        null

Outputs            :
DeploymentDebugLogLevel :
```

Task 2: Configure local and global virtual network peering

Verifying the location of each virtual network. The first two networks are in East US, while the third one is in West US.

## Virtual networks

Default Directory (andrijanasharkoskaoutlook.onmicrosoft.com)

+ Create Manage view Refresh Export to CSV Open query Assign tags

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 1 to 3 of 3 records.

<input type="checkbox"/> Name ↑↓	Resource group ↑↓	Location ↑↓
<input type="checkbox"/> az104-05-vnet0	az104-05-rg1	East US
<input type="checkbox"/> az104-05-vnet1	az104-05-rg1	East US
<input type="checkbox"/> az104-05-vnet2	az104-05-rg1	West US

The networks were not displaying in the Peering section, so I had to run the following code to establish the connection:

```
PS /home/andrijana> $rgName = 'az104-05-rg1'
PS /home/andrijana> $vnet0 = Get-AzVirtualNetwork -Name 'az104-05-vnet0' -ResourceGroupName $rgName
PS /home/andrijana> $vnet1 = Get-AzVirtualNetwork -Name 'az104-05-vnet1' -ResourceGroupName $rgName
PS /home/andrijana> Add-AzVirtualNetworkPeering -Name 'az104-05-vnet0_to_az104-05-vnet1' -VirtualNetwork $vnet0 -RemoteVirtualNetworkId $vnet1.Id

Name                               : az104-05-vnet0_to_az104-05-vnet1
Id                                 : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet0/virtualNetworkPeerings/
Etag                               : W/"a632123b-75cf-42d0-86d0-f27f022fb195"
ResourceGroupName                  : az104-05-rg1
VirtualNetworkName                 : az104-05-vnet0
PeeringSyncLevel                   : RemoteNotInSync
PeeringState                       : Initiated
ProvisioningState                  : Succeeded
RemoteVirtualNetwork               : {
  "Id": "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet1"
}
AllowVirtualNetworkAccess          : True
AllowForwardedTraffic              : False
AllowGatewayTransit                : False
UseRemoteGateways                 : False
RemoteGateways                    : null
PeeredRemoteAddressSpace           : {
  "AddressPrefixes": [

```

```
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.51.0.0/22"
  ]
}

PS /home/andrijana>
PS /home/andrijana> Add-AzVirtualNetworkPeering -Name 'az104-05-vnet1_to_az104-05-vnet0' -VirtualNetwork $vnet1 -RemoteVirtualNetworkId $vnet0.Id

Name                               : az104-05-vnet1_to_az104-05-vnet0
Id                                 : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet1/virtualNetworkPeerings/az104-05-vnet1_t
Etag                               : W/"93f8e9da-4219-4d97-9260-bb8343d57445"
ResourceGroupName                  : az104-05-rg1
VirtualNetworkName                 : az104-05-vnet1
PeeringSyncLevel                   : FullyInSync
PeeringState                       : Connected
ProvisioningState                  : Succeeded
RemoteVirtualNetwork               : {
  "Id": "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet0"
}
AllowVirtualNetworkAccess          : True
AllowForwardedTraffic              : False
AllowGatewayTransit                : False
UseRemoteGateways                 : False
RemoteGateways                    : null
PeeredRemoteAddressSpace           : {
  "AddressPrefixes": [
    "10.50.0.0/22"

```

I had to repeat the same step in the PowerShell for adding a peering from az104-05-vnet2 to az104-05-vnet0:

```
PS /home/andriana> $rgName = 'az104-05-rg1'
PS /home/andriana> $vnet0 = Get-AzVirtualNetwork -Name 'az104-05-vnet0' -ResourceGroupName $rgName
PS /home/andriana> $vnet2 = Get-AzVirtualNetwork -Name 'az104-05-vnet2' -ResourceGroupName $rgName
PS /home/andriana> Add-AzVirtualNetworkPeering -Name 'az104-05-vnet0_to_az104-05-vnet2' -VirtualNetwork $vnet0 -RemoteVirtualNetworkId $vnet2.Id

Name                               : az104-05-vnet0_to_az104-05-vnet2
Id                                 : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet0/virtualNetworkPeerings/az104-05-vnet0_to_az104-05-vnet2
Etag                               : W/"2f6d6acc-7acc-451b-be24-c2843f2bbbaf"
ResourceGroupName                  : az104-05-rg1
VirtualNetworkName                 : az104-05-vnet0
PeeringSyncLevel                   : RemoteNotInSync
PeeringState                       : Initiated
ProvisioningState                  : Succeeded
RemoteVirtualNetwork               : {
  "Id": "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet2"
}
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/andriana> Add-AzVirtualNetworkPeering -Name 'az104-05-vnet2_to_az104-05-vnet0' -VirtualNetwork $vnet2 -RemoteVirtualNetworkId $vnet0.Id
Name                               : az104-05-vnet2_to_az104-05-vnet0
Id                                 : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet2/virtualNetworkPeerings/az104-05-vnet2_to_az104-05-vnet0
Etag                               : W/"346f3815-f8a9-4770-a029-f94618c17f4a"
ResourceGroupName                  : az104-05-rg1
VirtualNetworkName                 : az104-05-vnet2
PeeringSyncLevel                   : FullyInSync
PeeringState                       : Connected
ProvisioningState                  : Succeeded
RemoteVirtualNetwork               : {
  "Id": "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet0"
}
AllowVirtualNetworkAccess          : True
AllowForwardedTraffic              : False
AllowGatewayTransit                : False
UseRemoteGateways                 : False
RemoteGateways                    : null
PeeredRemoteAddressSpace           : {
  "AddressPrefixes": [
    "10.50.0.0/22"
  ]
}
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.50.0.0/22"
  ]
}
```

Since no virtual networks are available for the peering between virtual network 1 and virtual network 2, the same steps from above are repeated:

### Add peering ...

az104-05-vnet1

☐ Use this virtual network's gateway or Route Server

☐ Use the remote virtual network's gateway or Route Server

☒ None (default)

Remote virtual network

Peering link name \*

Virtual network deployment model ⓘ


☒ Resource manager

☐ Classic

☐ I know my resource ID ⓘ

Subscription \* ⓘ

Azure Pass - Sponsorship

 No virtual network in this subscription

Virtual network \*

Traffic to remote virtual network ⓘ

☒ Allow (default)

☐ Block all traffic to the remote virtual network

```
PS /home/andriana> $rgName = 'az104-05-rg1'
PS /home/andriana> $vnet1 = Get-AzVirtualNetwork -Name 'az104-05-vnet1' -ResourceGroupName $rgName
PS /home/andriana> $vnet2 = Get-AzVirtualNetwork -Name 'az104-05-vnet2' -ResourceGroupName $rgName
PS /home/andriana> Add-AzVirtualNetworkPeering -Name 'az104-05-vnet1_to_az104-05-vnet2' -VirtualNetwork $vnet1 -RemoteVirtualNetworkId $vnet2.Id

Name                               : az104-05-vnet1_to_az104-05-vnet2
Id                                 : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet1/virtualNetworkPeerings/az104-05-vnet1_to_az104-05-vnet2
Etag                               : W/"d88c2c6f-5fd6-42e3-a3f7-c6867b4dfdae"
ResourceGroupName                  : az104-05-rg1
VirtualNetworkName                 : az104-05-vnet1
PeeringSyncLevel                   : RemoteNotInSync
PeeringState                       : Initiated
ProvisioningState                   : Succeeded
RemoteVirtualNetwork               : {
  "Id": "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet2"
}
AllowVirtualNetworkAccess          : True
AllowForwardedTraffic              : False
AllowGatewayTransit                : False
UseRemoteGateways                  : False
RemoteGateways                     : null
PeeredRemoteAddressSpace           : {
}
```


```

PS /home/andriana> Add-AzVirtualNetworkPeering -Name 'az104-05-vnet2_to_az104-05-vnet1' -VirtualNetwork $vnet2 -RemoteVirtualNetworkId $vnet1.Id
PS /home/andriana>
Name                                : az104-05-vnet2_to_az104-05-vnet1
Id                                  : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet2/virtualNetworkPeerings/az104-05-vnet2_to_az104-05-vnet1
Tag                                  : w/a722a8d2-7345-4ae8-aded-276886cf9e2f
ResourceGroupName                   : az104-05-rg1
VirtualNetworkName                  : az104-05-vnet2
PeeringSyncLevel                    : FullyInSync
PeeringState                        : Connected
ProvisioningState                   : Succeeded
RemoteVirtualNetwork                : {
  "Id": "/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-05-rg1/providers/Microsoft.Network/virtualNetworks/az104-05-vnet1"
}
AllowVirtualNetworkAccess           : True
AllowForwardedTraffic               : False
AllowGatewayTransit                 : False
UseRemoteGateways                   : False
RemoteGateways                     : null
PeerRemoteAddressSpace              : {
  "AddressPrefixes": [
    "10.51.0.0/22"
  ]
}
RemoteVirtualNetworkAddressSpace : {
  "AddressPrefixes": [
    "10.51.0.0/22"
  ]
}

```

### Task 3: Test intersite connectivity

Checking connectivity for the Virtual Network 0 through RDP:


**az104-05-vm0** | Connect
 ☆
...

Virtual machine

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Networking


**Connect**

Windows Admin Center

Disks

Size

Microsoft Defender for Cloud

 To improve security, enable just-in-time access on

RDP SSH Bastion

**Connect with RDP**

✓

Suggested method for connecting

To connect to your virtual machine via RDP, select an RDP file.

IP address \*

Public IP address (20.169.136.76)

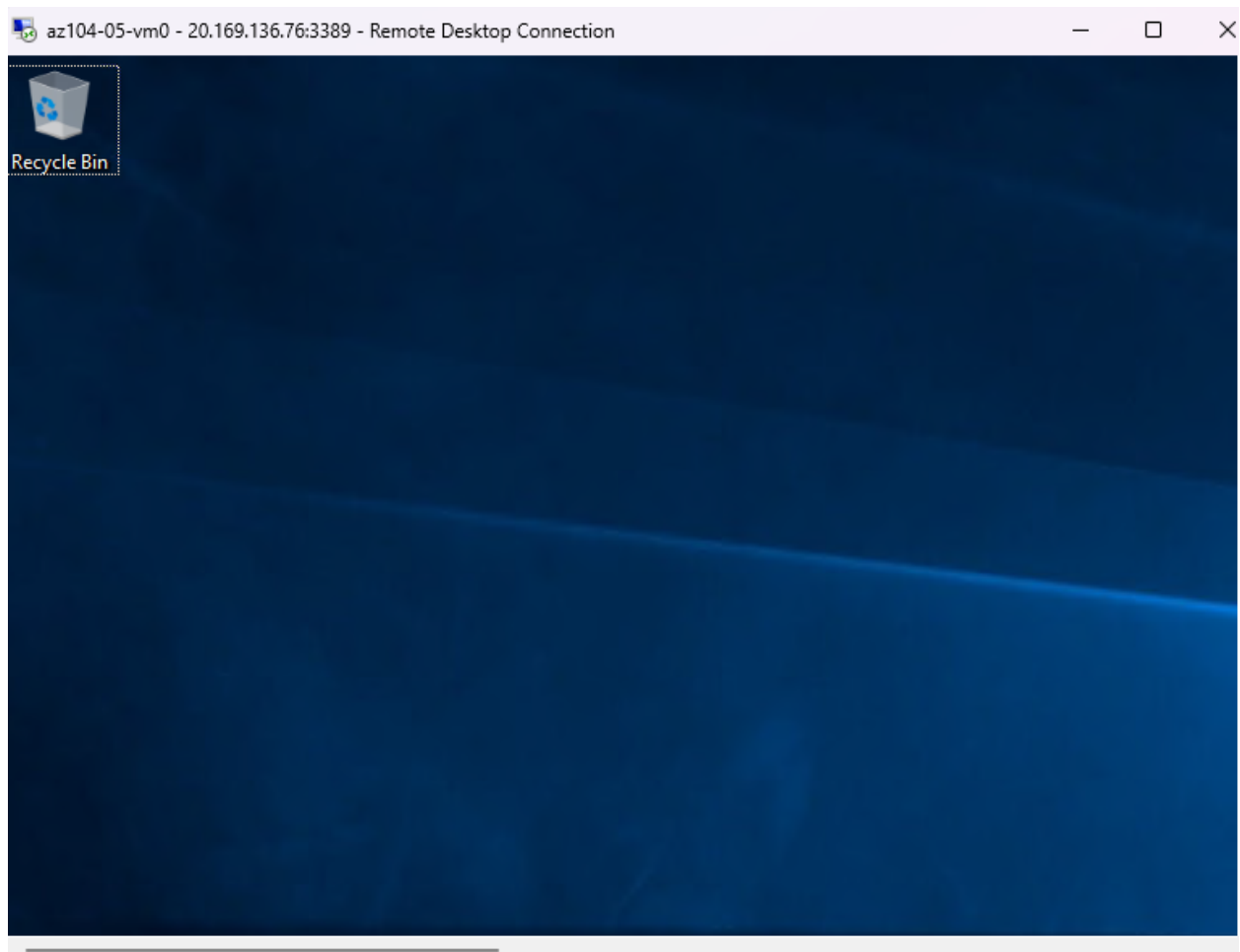
Port number \*

3389

Download RDP File

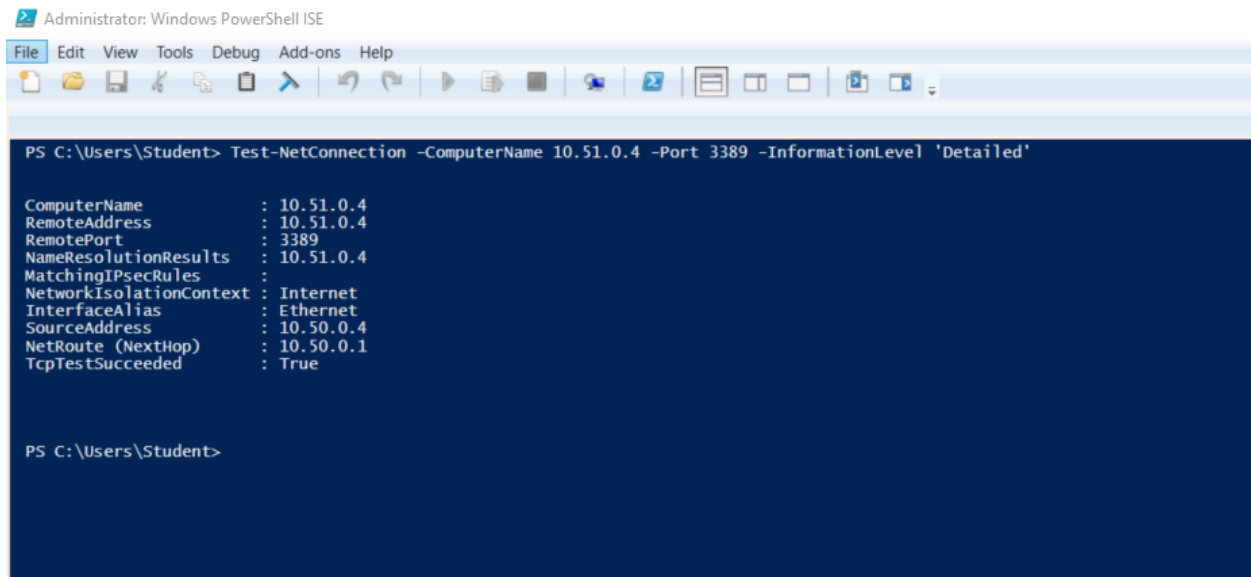
Can't connect?

When prompted, sign in by using the Student username and the password from your parameters file.



We are testing the connectivity to az104-05-vm1 via PowerShell (run as administrator):

```
Test-NetConnection -ComputerName 10.51.0.4 -Port 3389 -InformationLevel  
'Detailed'
```



The screenshot shows the Windows PowerShell ISE interface. The title bar reads 'Administrator: Windows PowerShell ISE'. The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The command prompt shows the execution of the `Test-NetConnection -ComputerName 10.51.0.4 -Port 3389 -InformationLevel 'Detailed'` command. The output displays detailed network connectivity information, including the remote address, port, and interface details.

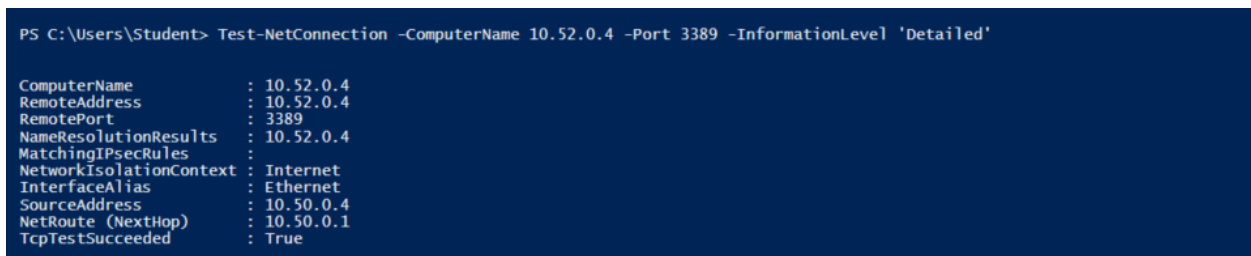
```
PS C:\Users\Student> Test-NetConnection -ComputerName 10.51.0.4 -Port 3389 -InformationLevel 'Detailed'
```

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\Student>
```

We are testing the connectivity to az104-05-vm2 via PowerShell (run as administrator):

```
Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel  
'Detailed'
```



The screenshot shows the Windows PowerShell ISE interface. The title bar reads 'Administrator: Windows PowerShell ISE'. The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The command prompt shows the execution of the `Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel 'Detailed'` command. The output displays detailed network connectivity information, including the remote address, port, and interface details.

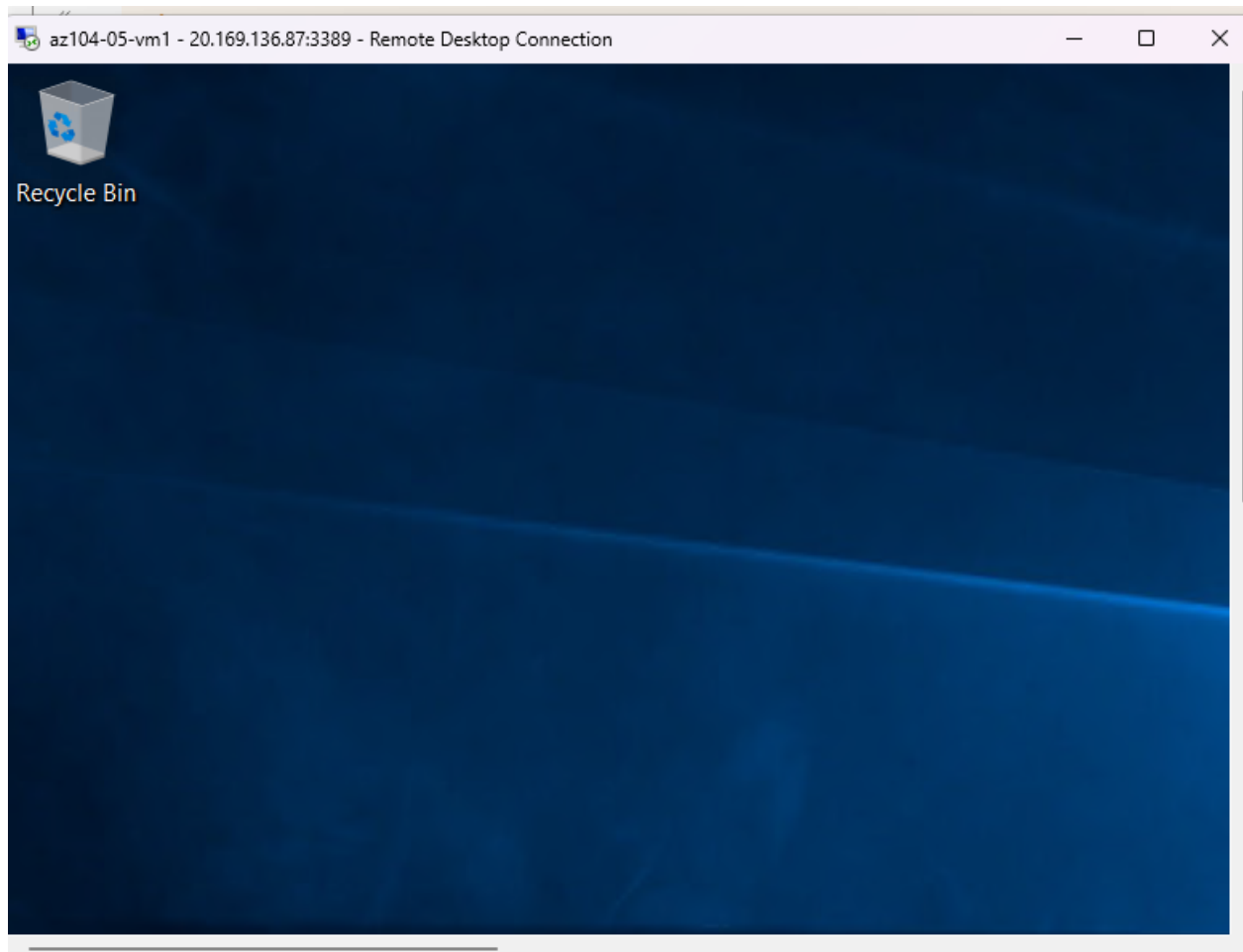
```
PS C:\Users\Student> Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel 'Detailed'
```

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

Upon concluding the connection was successful, we navigate back to the default directory in the Azure Portal.

In the next step, we check the connectivity for the Virtual Network 1 through RDP:





We are checking connectivity to az104-05-vm2 via PowerShell (run as administrator).

In the Windows PowerShell console window, run the following to test connectivity to az104-05-vm2 (which has the private IP address of 10.52.0.4) over TCP port 3389:

```
Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel  
'Detailed'
```

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

PS C:\Users\Student> Test-NetConnection -ComputerName 10.52.0.4 -Port 3389 -InformationLevel 'Detailed'

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\Student> █
```

After we've ensured the interconnectivity among the three virtual networks is established, it's time to clean up the resources.