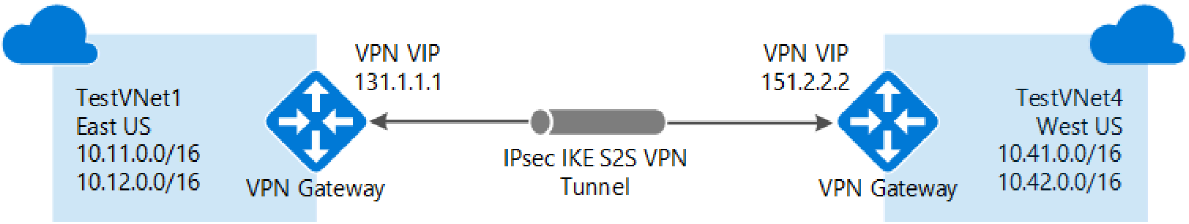
VNET to VNET connectivity using PowerShell (with in same subscription)

Note: ***When you use the Azure porta****l* to connect virtual networks, the ***VNets must be in the same subscription***. If your **virtual networks are in different subscriptions**, you can still connect them by **using the PowerShell**.



VNet peering

It's also possible to connect VNets without using a VPN gateway. If your VNets are in the same region, you may want to consider connecting them by using VNet peering.

VNET to VNET connections

Connecting a virtual network to another virtual network (VNet-to-VNet) is similar to connecting a VNet to an on-premises site location. Both connectivity types use an Azure VPN gateway to provide a secure tunnel using IPsec/IKE. The VNets you connect can be in different regions, or in different subscriptions.

Why to connect VNET to VNET?

- **Cross region geo-redundancy**
  You can set up your own geo-replication or synchronization with secure connectivity without going over Internet-facing endpoints.
- **Geo-presence**
  With Azure Traffic Manager and Load Balancer, you can set up highly available workload with geo-redundancy across multiple Azure regions. One important example is to set up SQL Always On with Availability Groups spreading across multiple Azure regions.
- **Regional multi-tier applications with isolation or administrative boundary**
  Within the same region, you can set up multi-tier applications with multiple virtual networks connected together due to isolation or administrative requirements.

Steps to Create VNET to VNET Connection using PowerShell

We can set up VNET to VNET connections using PowerShell. PowerShell allows us to connect VNETs that reside in the same subscription and VNETs that resides in different subscriptions. Using Azure Portal, we can connect two VNETs that resides in same subscription only.

Method 1: VNets that reside in the same subscription

1) Create a resource group with the following configurations

Name: HybridCloudResGroup
Location: SouthEast Asia
Subscription: Your active Subscription

```
#Login to Azure Account
Login-AzureRmAccount

#Select default subscription
Set-AzureRmContext -SubscriptionName "<your subscription>"

#Create a new Resource group
$resourceGroup="HybridCloudResGroup"
$location="Southeast Asia"
New-AzureRmResourceGroup -Name $resourceGroup `
                         -Location $location
```

2) We need to create two Virtual networks and their subnets. Also, we need to set up two Virtual network gateways to connect the virtual networks together.
The following table shows the configurations list used to create the virtual networks and virtual network gateways.

VNET Name	FirstVNET	
Resource Group	HybridCloudResGrp	
Location	Southeast Asia	
Address space	10.11.0.0/16	
Subnet 1		
Name	FrontEnd	
Address range	10.11.0.0/24	
Subnet 2		
Name	BackEnd	
Address range	10.12.0.0/24	
Gateway subnet		
Name	Gateway Subnet (default)	
Address range	10.13.0.0/27	

Create the FirstVNET using the following PowerShell command.

```
#FirstVNET creation
$resourceGroupName="HybridCloudResGroup"

$firstVNETName="FirstVNET"
$firstVNETLocation="Southeast Asia"
$firstVNETAddressSpace="10.11.0.0/16"
$frontEndSubnetName ="FrontEnd"
$backEndSubnetName = "BackEnd"
$fnGatewaySubnetName="GatewaySubnet"
$frontEndSubnetIPrange ="10.11.1.0/24"
$backEndSubnetIPrange ="10.11.2.0/24"
$fnGatewaySubnetIPrange ="10.11.3.0/27"
```

```
#Create Subnet configurations for FirstVNET
$frontEndSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
$frontEndSubnetName -AddressPrefix $frontEndSubnetIPrange
$backEndSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
$backEndSubnetName -AddressPrefix $backEndSubnetIPrange
$firstVNETGatewaySubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
$fnGatewaySubnetName -AddressPrefix $fnGatewaySubnetIPrange

#Create FirstVNET
New-AzureRmVirtualNetwork -Name $firstVNETName `
      -ResourceGroupName $resourceGroup `
      -Location $firstVNETLocation `
      -AddressPrefix $firstVNETAddressSpace `
      -Subnet $frontEndSubnet,$backEndSubnet,$firstVNETGatewaySubnet
```

3) Now you need to create SecondVNET, use the following configurations to create second VNET

VNET Name	SecondVNET	
Resource Group	HybridCloudResGrp	
Location	East US	
Address space	10.50.0.0/16	
Subnet 1		
Name	Dev	
Address range	10.50.1.0/24	
Subnet 2		
Name	Test	
Address range	10.50.1.0/24	
Gateway subnet		
Name	Gateway Subnet (default)	
Address range	10.50.1.0/27	

Use the following Powershell script to create SecondVNET

```
#SecondVNET creation
$resourceGroupName="HybridCloudResGroup"
$secondVNETName="SecondVNET"
$secondVNETLocation="East US"
$secondVNETAddressSpace="10.50.0.0/16"
$devSubnetName ="Dev"
$testSubnetName =  "Test"
$secGatewaySubnetName="GatewaySubnet"
$devSubnetIPrange ="10.50.1.0/24"
$testSubnetIPrange ="10.50.2.0/24"
$secGatewaySubnetIPrange ="10.50.3.0/27"

#Create Subnet configurations for SecondVNET
$devSubnet=New-AzureRmVirtualNetworkSubnetConfig `
            -Name $devSubnetName `
```

```
               -AddressPrefix $devSubnetIPrange

   $testSubnet=New-AzureRmVirtualNetworkSubnetConfig `
               -Name $testSubnetName `
               -AddressPrefix $testSubnetIPrange

   $secondVNETGatewaySubnet=New-AzureRmVirtualNetworkSubnetConfig `
               -Name $secGatewaySubnetName `
               -AddressPrefix $secGatewaySubnetIPrange

   #Create SecondVNET
   New-AzureRmVirtualNetwork -Name $secondVNETName `
               -ResourceGroupName $resourceGroup `
               -Location $secondVNETLocation `
               -AddressPrefix $secondVNETAddressSpace `
               -Subnet $devSubnet,$testSubnet,$secondVNETGatewaySubnet
```

4) Create the Gateway for the FirstVNET, it may take 45 minute to create the gateway.

```
#Create a Public IP for Virtual Network Gateway of FirstVNET
$resourceGroupName="HybridCloudResGroup"
$firstVNETName="FirstVNET"
$firstVNETLocation="Southeast Asia"

$fnGatewayPublicIPName="FirstGatewayPublicIP"
$fnGatewayIPConfigName="FirstGatewayIPConfig"
$fnGatewayName="FirstVNETGateway"

$fnGatewayPublicIP = New-AzureRmPublicIpAddress `
                  -Name $fnGatewayPublicIPName `
                  -ResourceGroupName $resourceGroupName `
                  -Location $firstVNETLocation `
                  -AllocationMethod Dynamic

$firstVNET = Get-AzureRmVirtualNetwork `
                  -Name $firstVNETName `
                  -ResourceGroupName $resourceGroupName

$fnGatewaySubnet = Get-AzureRmVirtualNetworkSubnetConfig `
                  -Name $fnGatewaySubnetName `
                  -VirtualNetwork $firstVNET

$fnGatewayIPConfig = New-AzureRmVirtualNetworkGatewayIpConfig `
                  -Name $fnGatewayIPConfigName `
                  -Subnet $fnGatewaySubnet `
                  -PublicIpAddress $fnGatewayPublicIP

New-AzureRmVirtualNetworkGateway -Name $fnGatewayName `
                  -ResourceGroupName $resourceGroupName `
                  -Location $firstVNETLocation `
                  -IpConfigurations $fnGatewayIPConfig `
                  -GatewayType Vpn `
                  -VpnType RouteBased `
                  -GatewaySku Standard
```

5) Create the Gateway for the SecondVNET, it may take 45 minute to create the gateway.

```
#Create a Public IP for Virtual Network Gateway of SecondVNET
$resourceGroupName="HybridCloudResGroup"
$secondVNETName="SecondVNET"
$secondVNETLocation="East US"
$secGatewaySubnetName="GatewaySubnet"

$secGatewayPublicIPName="SecondGatewayPublicIP"
$secGatewayIPConfigName="SecondGatewayIPConfig"
$secGatewayName="SecondVNETGateway"

$secGatewayPublicIP = New-AzureRmPublicIpAddress `
               -Name $secGatewayPublicIPName `
               -ResourceGroupName $resourceGroupName `
               -Location $secondVNETLocation `
               -AllocationMethod Dynamic

$secondVNET = Get-AzureRmVirtualNetwork -Name $secondVNETName `
               -ResourceGroupName $resourceGroupName

$secGatewaySubnet = Get-AzureRmVirtualNetworkSubnetConfig `
               -Name $secGatewaySubnetName `
               -VirtualNetwork $secondVNET


$secGatewayIPConfig = New-AzureRmVirtualNetworkGatewayIpConfig `
               -Name $secGatewayIPConfigName `
               -Subnet $secGatewaySubnet `
               -PublicIpAddress $secGatewayPublicIP

New-AzureRmVirtualNetworkGateway -Name $secGatewayName `
               -ResourceGroupName $resourceGroupName `
               -Location $secondVNETLocation `
               -IpConfigurations $secGatewayIPConfig `
               -GatewayType Vpn `
               -VpnType RouteBased `
               -GatewaySku Standard
```

6) Finally, we need to connect both the VNETs together using the gateways created above. You can execute the following scripts to create the connection.

```
Login-AzureRmAccount
#Create connection between two gateways

$resourceGroupName="HybridCloudResGroup"
$fnGatewayName="FirstVNETGateway"
$secGatewayName="SecondVNETGateway"
$firstVNETLocation="Southeast Asia"
$secondVNETLocation="East US"
$fnToSec="FirstVNETtoSecondVNET"
$secToFn="SecondVNETtoFirstVNET"

#Get the gateways references
$fnVNETGateway = Get-AzureRmVirtualNetworkGateway -Name $fnGatewayName `
               -ResourceGroupName $resourceGroupName
```

```
$secVNETGateway = Get-AzureRmVirtualNetworkGateway -Name $secGatewayName
               -ResourceGroupName $resourceGroupName

#Connecting FirstVNET to SecondVNET

New-AzureRmVirtualNetworkGatewayConnection -Name $fnToSec `
                            -ResourceGroupName $resourceGroupName `
                            -VirtualNetworkGateway1 $fnVNETGateway `
                            -VirtualNetworkGateway2 $secVNETGateway `
                            -Location $firstVNETLocation `
                            -ConnectionType Vnet2Vnet `
                            -SharedKey 'sample1234'


New-AzureRmVirtualNetworkGatewayConnection -Name $secToFn `
                            -ResourceGroupName $resourceGroupName `
                            -VirtualNetworkGateway1 $secVNETGateway `
                            -VirtualNetworkGateway2 $fnVNETGateway `
                            -Location $secondVNETLocation `
                            -ConnectionType Vnet2Vnet `
                            -SharedKey 'sample1234'
```

Test the VNET-to-VNET Connectivity

To test the connectivity, we can create two virtual machines in each VNETs and access one from another. To do so we need to create a Virtual Machine in the '**FrontEnd'** subnet of the '**FirstVNET'** and another Virtual Machine in the '**Dev'** Subnet of the '**SecondVNET'**.

Create a VM in FrontEnd Subnet of FirstVNET

Basic	
Name	WebServerVM
VM Disk type	HDD
User name	[your username]
Password/Confirm Password	[Your password]
Subscription	[Your Active subscription]
Resource group	HybridCloudResourceGroup [You created above]
Location	Southeast Asia
Size	
VM Size	A2 Basic
Settings	
Storage: Use managed disks	No
Storage account	Choose existing storage account or Create New
Network	
Virtual Network	FirstVNET
Subnet	FrontEnd
Public IP	Create new public IP [Name: WebServerVMPublicIP]
NSG	None
Extensions	No extensions

High Availability	Availability set:None
Boot diagnostics	Disabled
Guest OS Diagnostics	Disabled

1) Use the following PowerShell command to create the ‘**WebServerVM’** in **FrontEnd** subsnet of the ‘**FirstVNET**’.

```
#create a new VM in FrontEnd subnet of FirstVNET
$locationName ="Southeast Asia"
$resourceGroupName="HybridCloudResGroup"

#create storage account
$storageAccName = "firstvnetvmstorage"
$storageAcc = New-AzureRmStorageAccount `
                     -ResourceGroupName $resourceGroupName `
                     -Name $storageAccName `
                     -Type "Standard_LRS" `
                     -Location $locationName

#Get the FrontEnd subnet from FirstVNET
$subnetName = "FrontEnd"
$fnVNETName="FirstVNET"

$firstVNET=Get-AzureRmVirtualNetwork -Name $fnVNETName `
                     -ResourceGroupName $resourceGroupName

$frontEndSubnet = Get-AzureRmVirtualNetworkSubnetConfig `
                     -Name $subnetName `
                     -VirtualNetwork $firstVNET

#create public IP address and network interface
$webServerPublicIPName = "WebServerPublicIP"
$webServerPublicIP = New-AzureRmPublicIpAddress `
                     -Name $webServerPublicIPName `
                     -ResourceGroupName $resourceGroupName `
                     -Location $locationName `
                     -AllocationMethod Dynamic

$nicName = "WebServerNIC"
$nic = New-AzureRmNetworkInterface -Name $nicName `
                     -ResourceGroupName $resourceGroupName `
                     -Location $locationName `
                     -SubnetId $frontEndSubnet.Id `
                     -PublicIpAddressId $webServerPublicIP.Id

#Create a virtual machine
#Run the command to set the administrator account name and password for
the virtual machine.
$cred = Get-Credential -Message "Type the name and password of the local
administrator account."
$vmName = "WebServerVM"
$vm = New-AzureRmVMConfig -VMName $vmName `
                     -VMSize "Standard_A1"

#run the commands to define the operating system to use.
```

```
$compName = "MSVM-COMPUTER"
$vm = Set-AzureRmVMOperatingSystem -VM $vm `
                     -Windows -ComputerName $compName `
                     -Credential $cred `
                     -ProvisionVMAgent -EnableAutoUpdate

#Run the command to define the image to use to provision the virtual
machine.
$vm = Set-AzureRmVMSourceImage -VM $vm `
                     -PublisherName MicrosoftWindowsServer `
                     -Offer WindowsServer `
                     -Skus 2012-R2-Datacenter `
                     -Version "latest"

#add the network interface created to the virtual machine configuration.
$vm = Add-AzureRmVMNetworkInterface -VM $vm `
                     -Id $nic.Id

$blobPath = "vhds/WindowsR2DC.vhd"
$osDiskUri = $storageAcc.PrimaryEndpoints.Blob.ToString() + $blobPath

$diskName = "windowsvmosdisk"
$vm = Set-AzureRmVMOSDisk -VM $vm `
                     -Name $diskName `
                     -VhdUri $osDiskUri `
                     -CreateOption fromImage

New-AzureRmVM -ResourceGroupName $resourceGroupName `
                     -Location $locationName `
                     -VM $vm
```

Create a VM in Dev Subnet of SecondVNET

Create a new VM in the second VNET using the following configurations. You can follow the steps used to create **WebServerVM** in the **FrontEnd** subnet of **FirstVNET**.

Basic	
Name	DevServerVM
VM Disk type	HDD
User name	[your username]
Password/Confirm Password	[Your password]
Subscription	[Your Active subscription]
Resource group	HybridCloudResourceGroup [You created above]
Location	East US
Size	
VM Size	A2 Basic
Settings	
Storage: Use managed disks	No
Storage account	Choose existing storage account or Create New
Network	
Virtual Network	SecondVNET
Subnet	Dev
Public IP	Create new public IP [Name: DevServerVMPublicIP]
NSG	None

Extensions	No extensions
High Availability	Availability set:None
Boot diagnostics	Disabled
Guest OS Diagnostics	Disabled

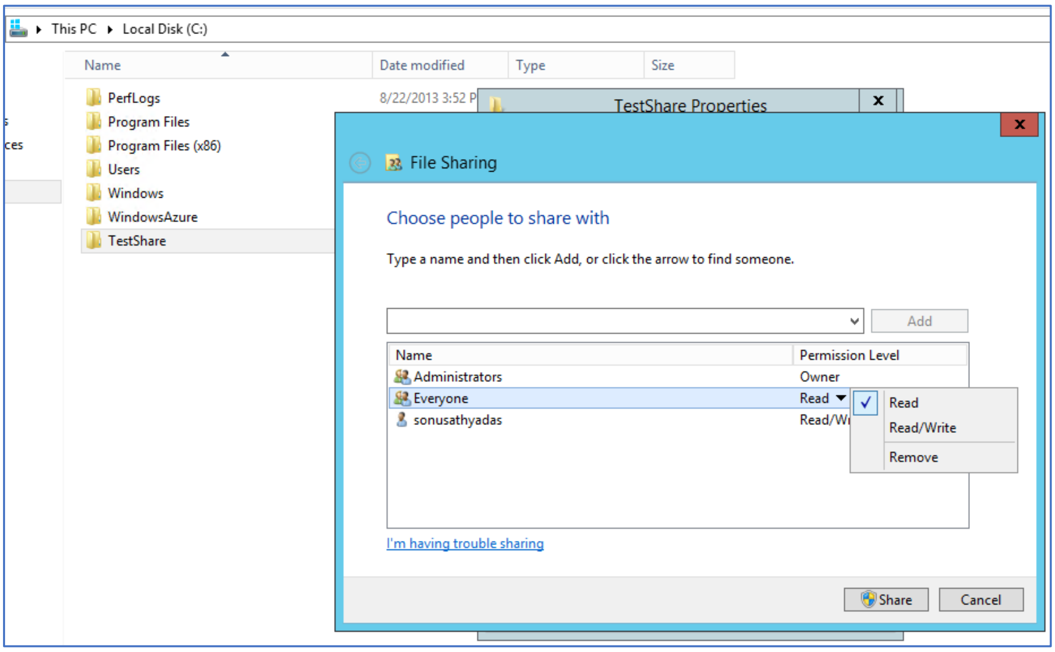
Use the following PowerShell Script to create the VM.

```
#create a new VM in Dev Subnet of SecondVNET
$locationName ="East US"
$resourceGroupName="HybridCloudResGroup"

#create storage account
$storageAccName = "secondvnetvmstorage"
$storageAcc = New-AzureRmStorageAccount `
                        -ResourceGroupName $resourceGroupName `
                        -Name $storageAccName `
                        -Type "Standard_LRS" `
                        -Location $locationName

#Get the FrontEnd subnet from FirstVNET
$subnetName = "Dev"
$secVNETName="SecondVNET"

$secondVNET=Get-AzureRmVirtualNetwork -Name $secVNETName `
                        -ResourceGroupName $resourceGroupName

$devSubnet = Get-AzureRmVirtualNetworkSubnetConfig `
                        -Name $subnetName `
                        -VirtualNetwork $secondVNET

#create public IP address and network interface
$devServerPublicIPName = "DevServerPublicIP"
$devServerPublicIP = New-AzureRmPublicIpAddress `
                        -Name $devServerPublicIPName `
                        -ResourceGroupName $resourceGroupName `
                        -Location $locationName `
                        -AllocationMethod Dynamic

$nicName = "DevServerNIC"
$nic = New-AzureRmNetworkInterface -Name $nicName `
                        -ResourceGroupName $resourceGroupName `
                        -Location $locationName `
                        -SubnetId $devSubnet.Id `
                        -PublicIpAddressId $devServerPublicIP.Id

#Create a virtual machine
#Run the command to set the administrator account name and password
for the virtual machine.
$cred = Get-Credential -Message "Type the name and password of the
local administrator account."
$vmName = "DevServerVM"
$vm = New-AzureRmVMConfig -VMName $vmName `
                        -VMSize "Standard_A1"

#run the commands to define the operating system to use.
```

```
$compName = "DEV-COMPUTER"
$vm = Set-AzureRmVMOperatingSystem -VM $vm `
                        -Windows -ComputerName $compName `
                        -Credential $cred `
                        -ProvisionVMAgent -EnableAutoUpdate

#Run the command to define the image to use to provision the virtual
machine.
$vm = Set-AzureRmVMSourceImage -VM $vm `
                        -PublisherName MicrosoftWindowsServer `
                        -Offer WindowsServer `
                        -Skus 2012-R2-Datacenter `
                        -Version "latest"

#add the network interface created to the virtual machine
configuration.
$vm = Add-AzureRmVMNetworkInterface -VM $vm `
                        -Id $nic.Id

$blobPath = "vhds/WindowsR2DC.vhd"
$osDiskUri = $storageAcc.PrimaryEndpoints.Blob.ToString() + $blobPath

$diskName = "windowsvmosdisk"
$vm = Set-AzureRmVMOSDisk -VM $vm `
                        -Name $diskName `
                        -VhdUri $osDiskUri `
                        -CreateOption fromImage

New-AzureRmVM -ResourceGroupName $resourceGroupName `
                        -Location $locationName `
                        -VM $vm
```

2) Connect to both VMs using Remote connection (RDP).

3) Create a shared folder in the C:\ drive of the ‘**DevServerVM**’.



This PC ▸ Local Disk (C:)

Name	Date modified	Type	Size
PerfLogs	8/22/2013 3:52 P		
Program Files			
Program Files (x86)			
Users			
Windows			
WindowsAzure			
TestShare			

TestShare Properties

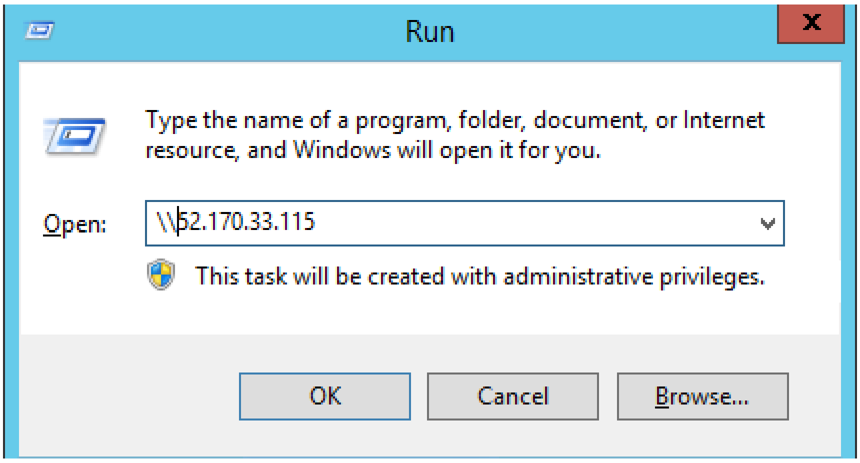
File Sharing

Choose people to share with

Type a name and then click Add, or click the arrow to find someone.

Name	Permission Level
Administrators	Owner
Everyone	Read
sonusathyadas	Read/Wr

4) Open **WebServerVM** and Open Run command and type the public IP of the **DevServerVM**.

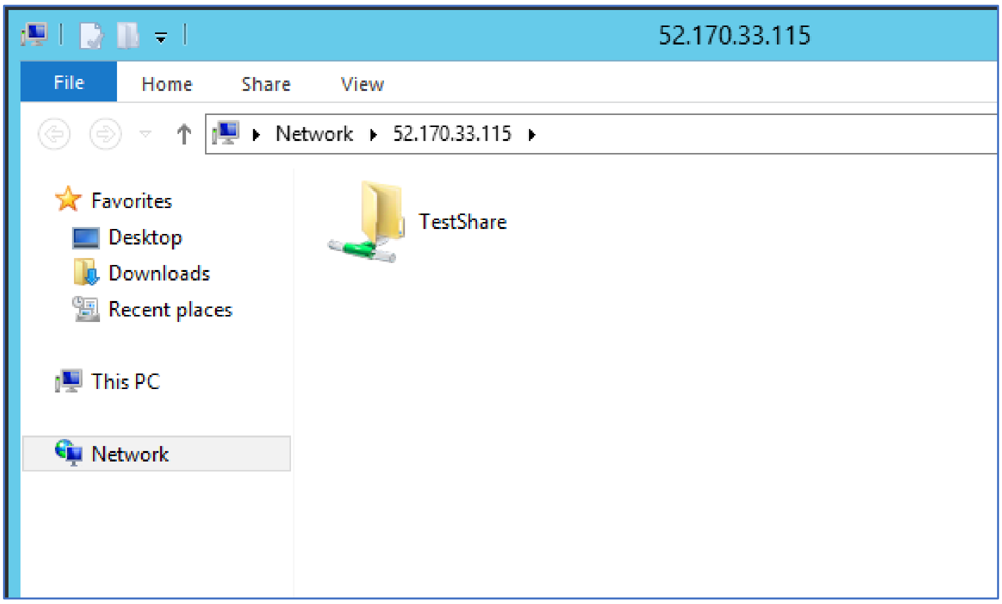


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: \\52.170.33.115

This task will be created with administrative privileges.

5) It opens the shared folder of the DevServerVM



6) You have completed the workshop successfully, Thanks