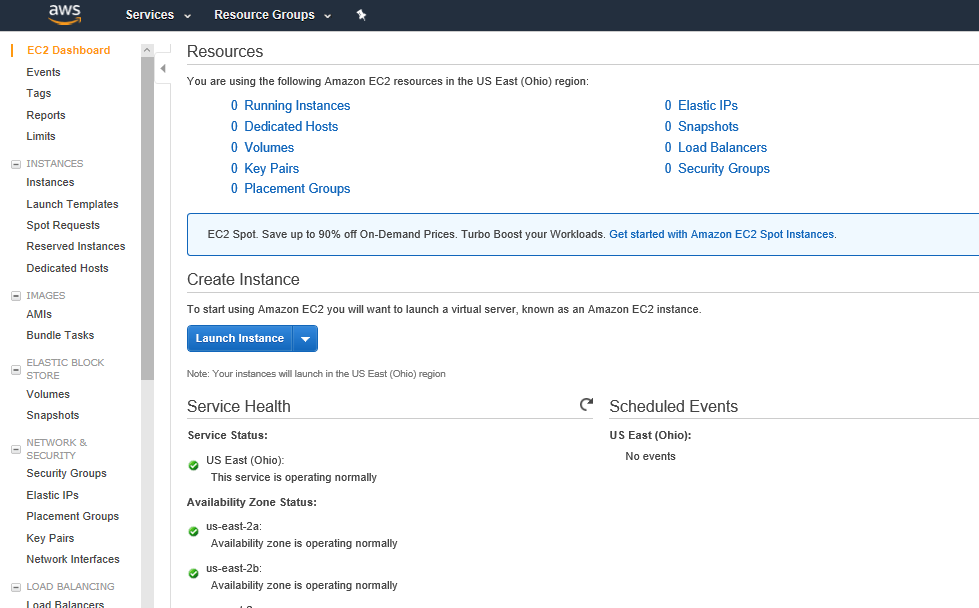
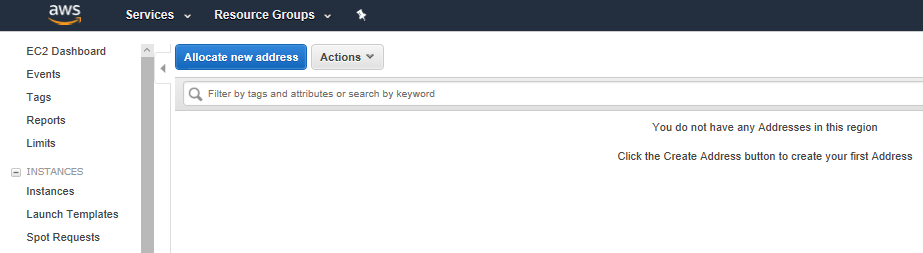
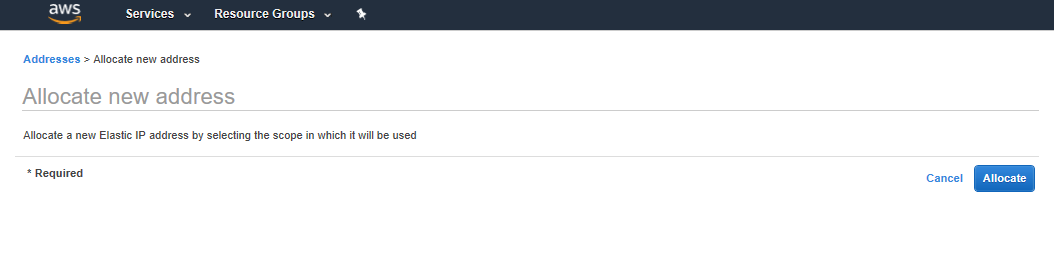
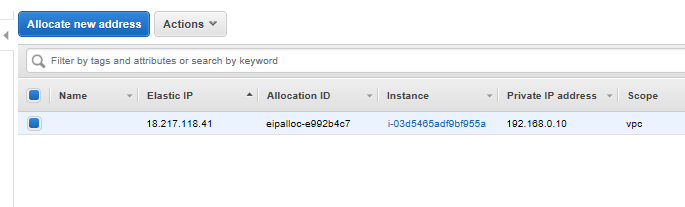
# **Connect AWS and Azure environments with a VPN tunnel using Windows RRAS Server**

**Create an Elastic IP**

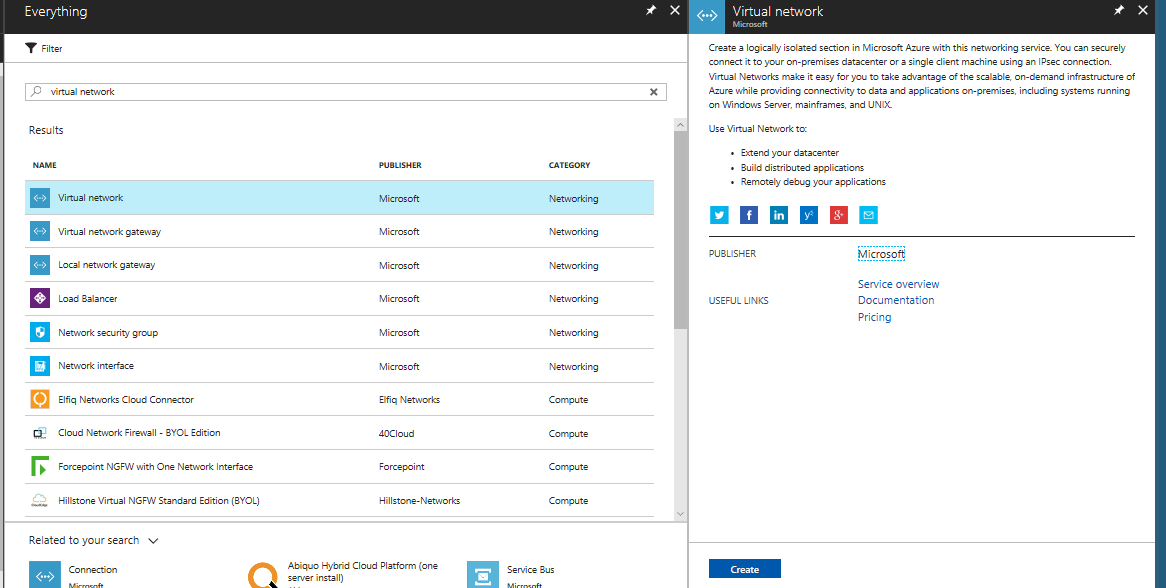


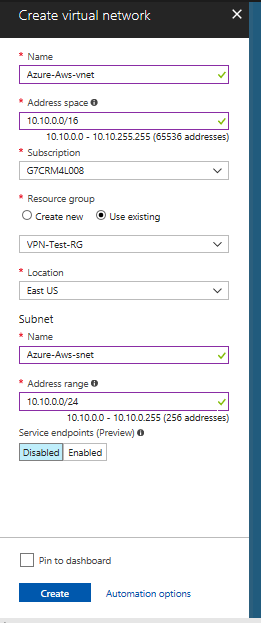


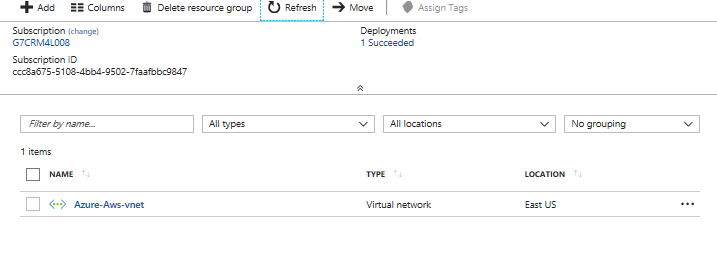




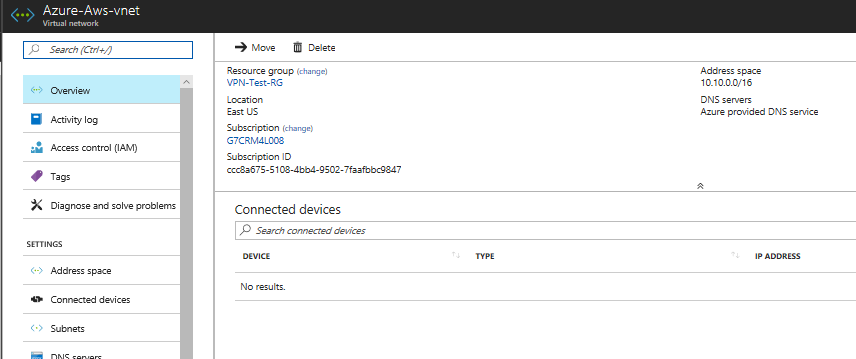
**Create Virtual network**

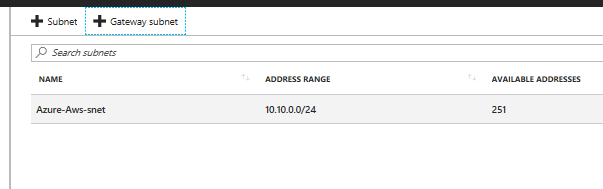


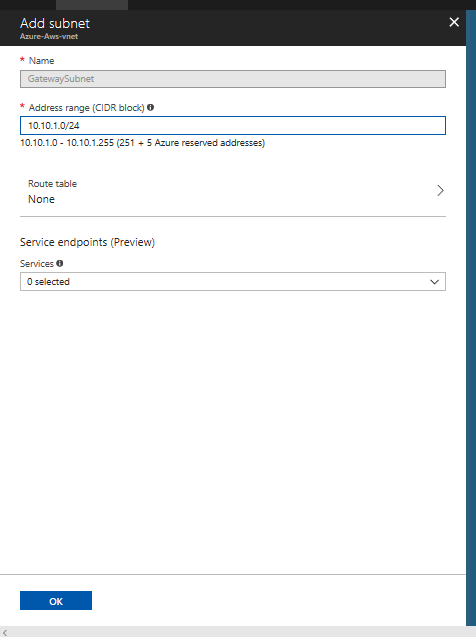




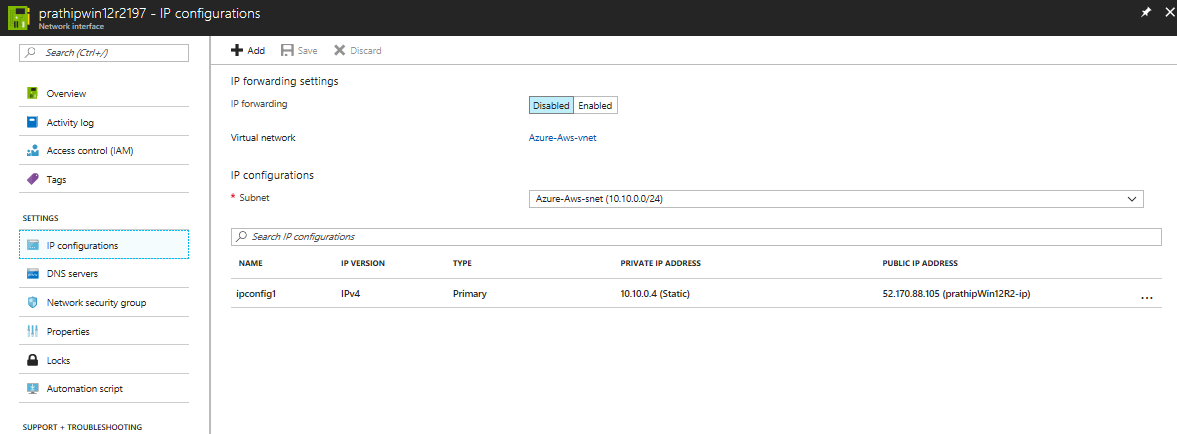
**Create Gateway Subnet**



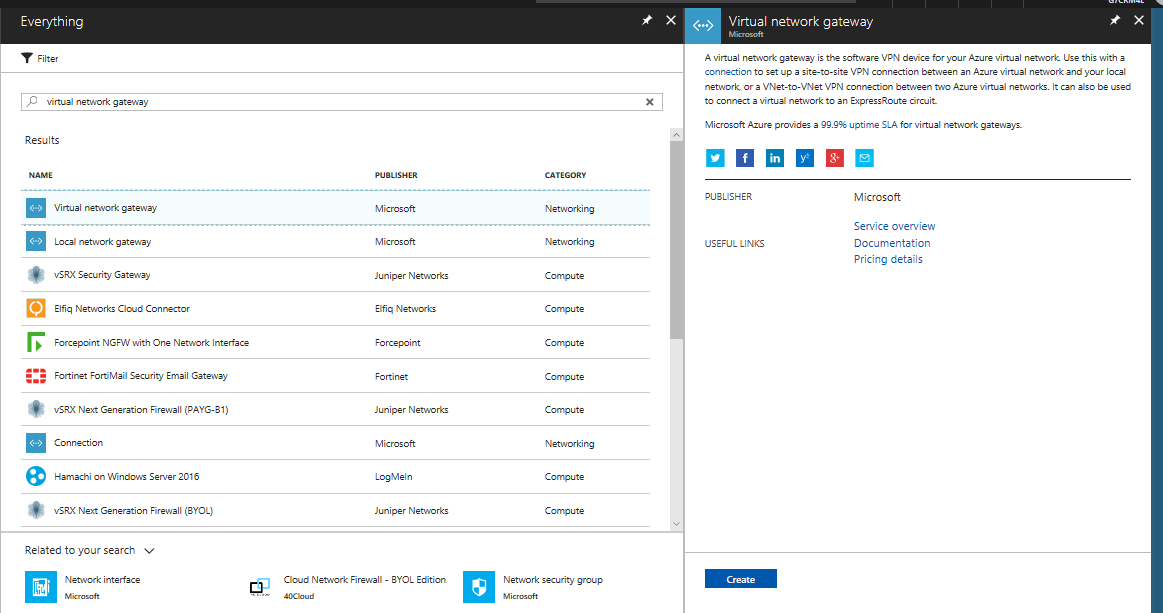


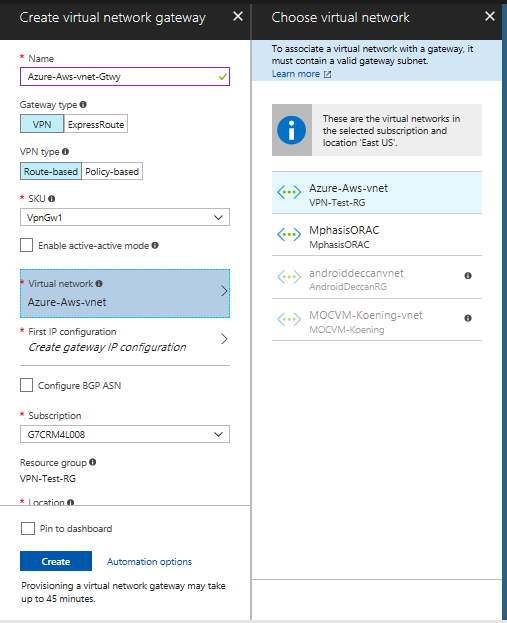


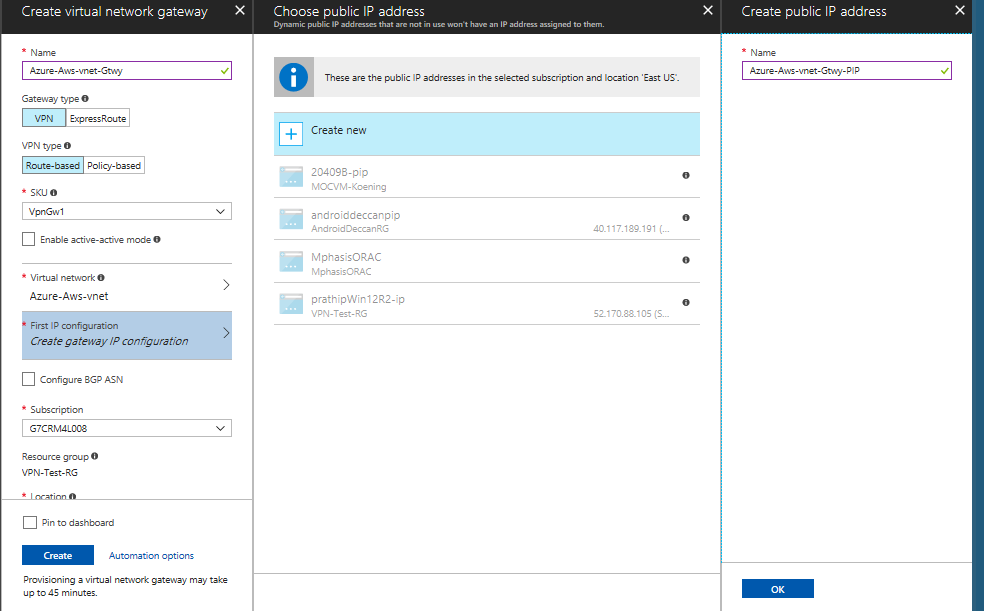
**Make the VM’s private IP as static**

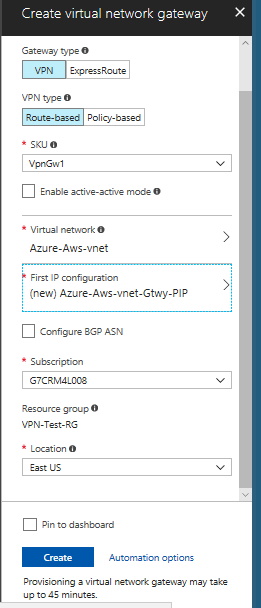


**Create Virtual Network Gateway**

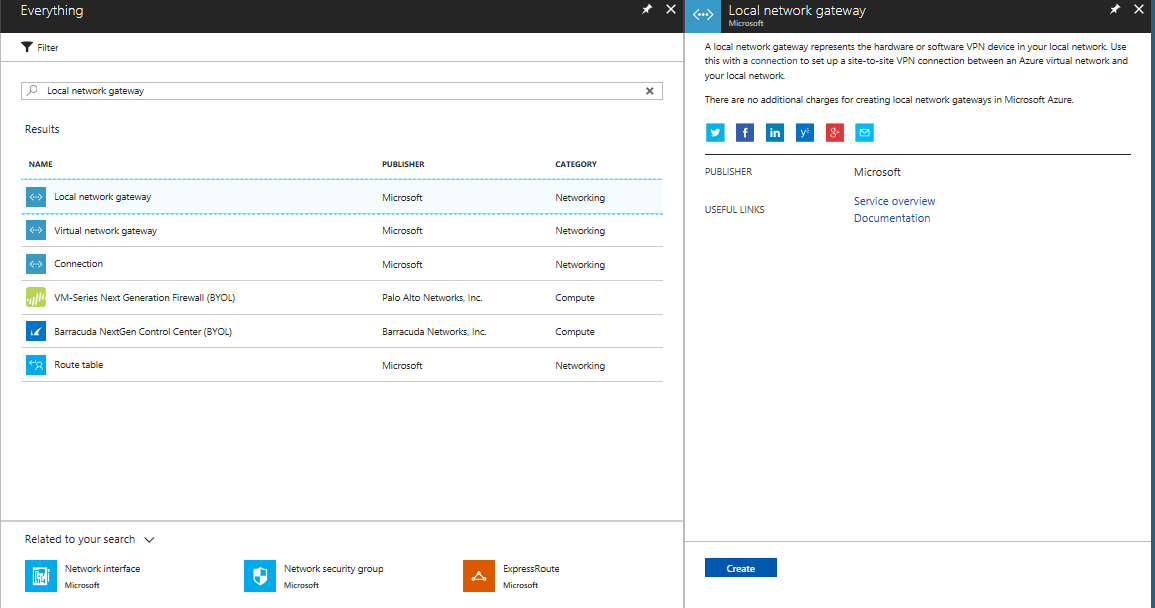


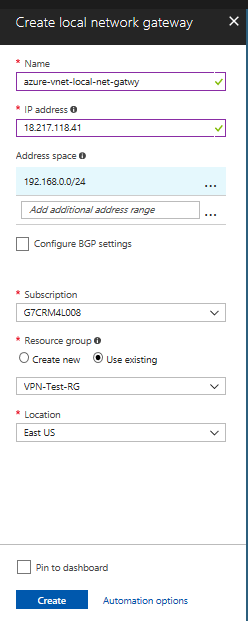






**Create Local Network Gateway**



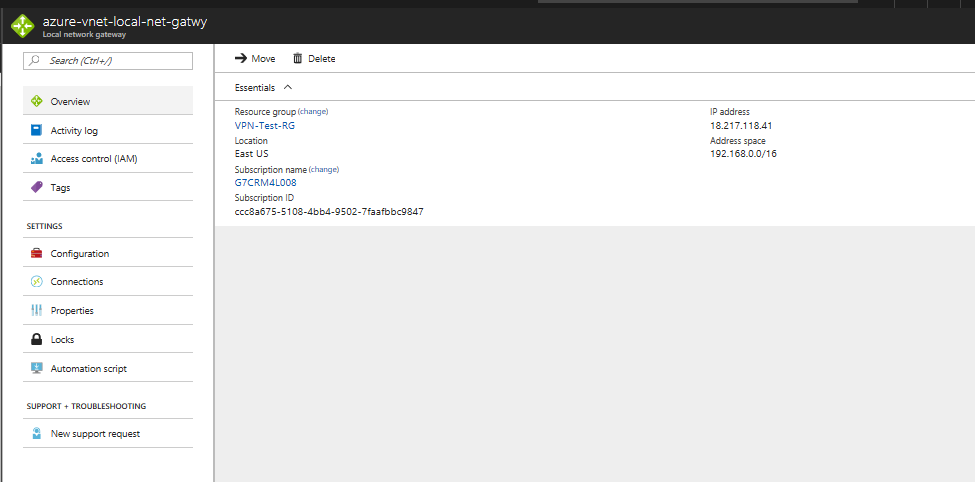


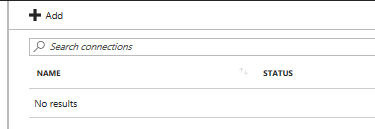


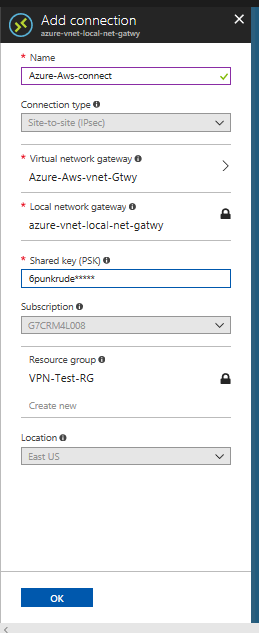
AWS VPC CIDR ADDR

AWS Elastic IP

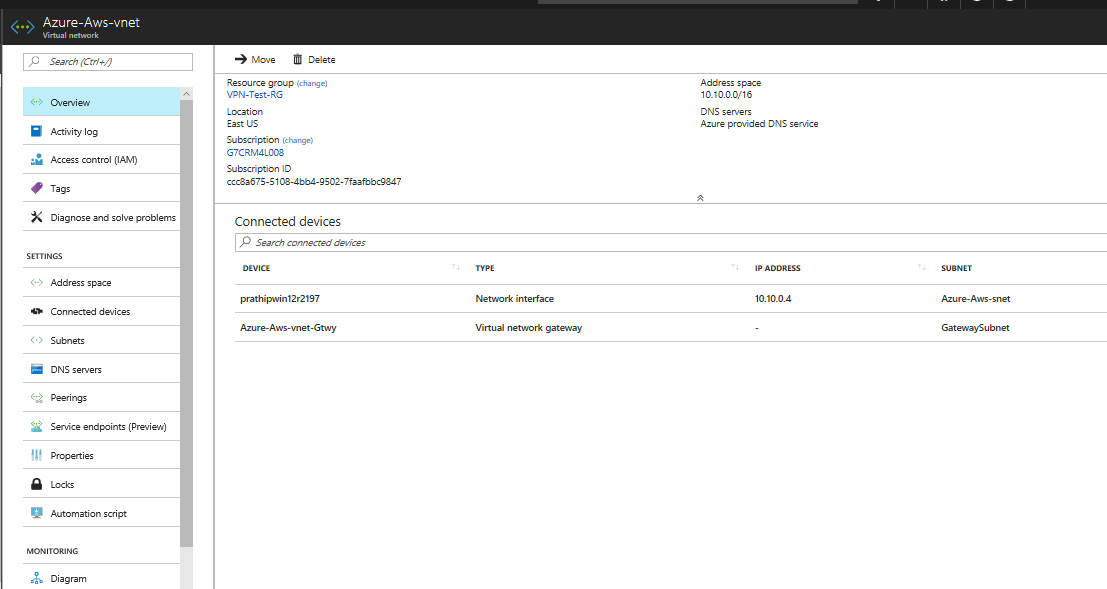
**Add a Connection in Local Network Gateway**

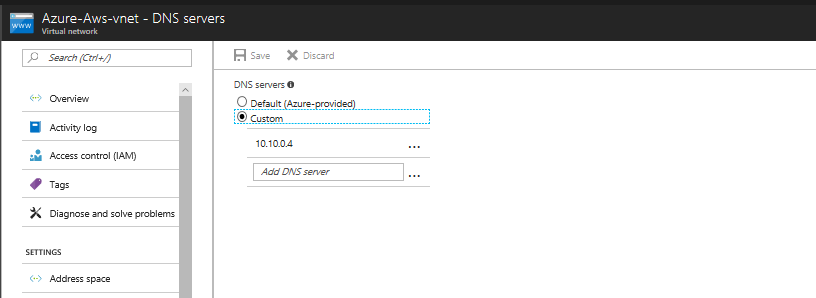






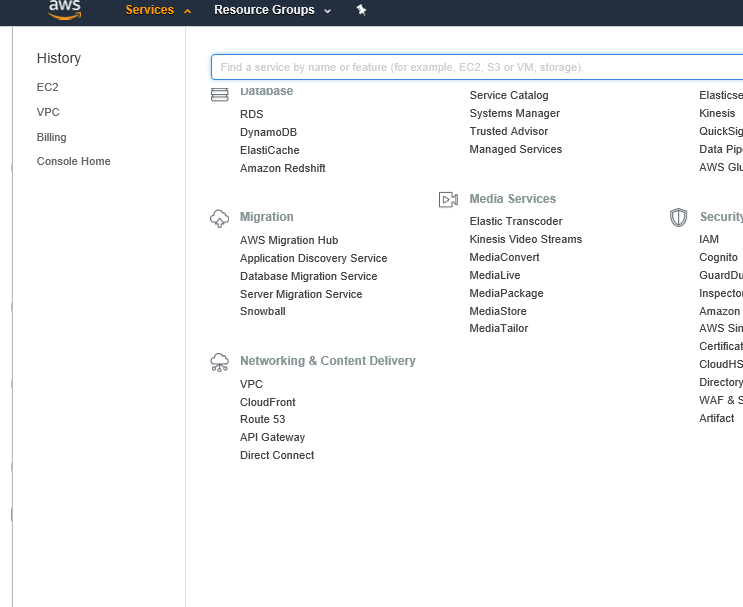
**Update the Custom DNS in the VNET**



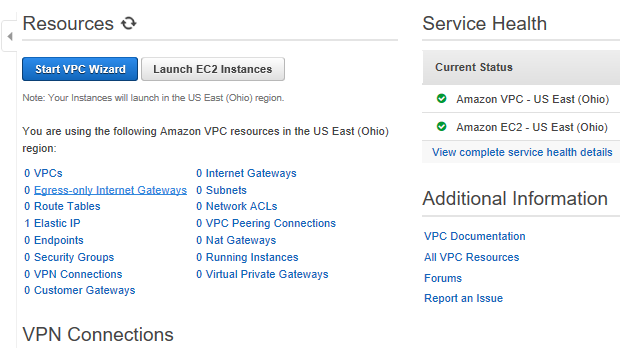


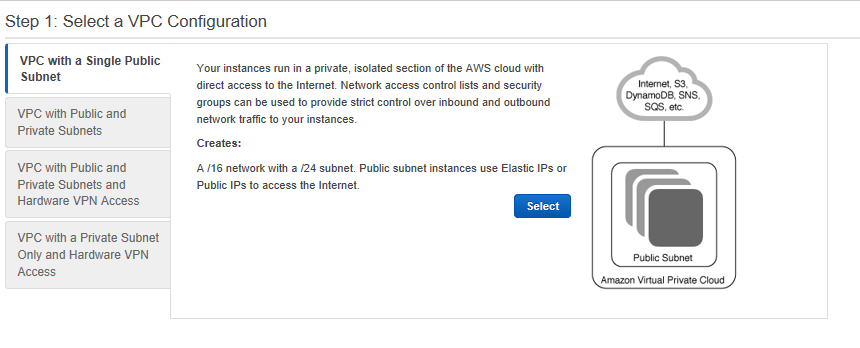
Azure VM’s private IP

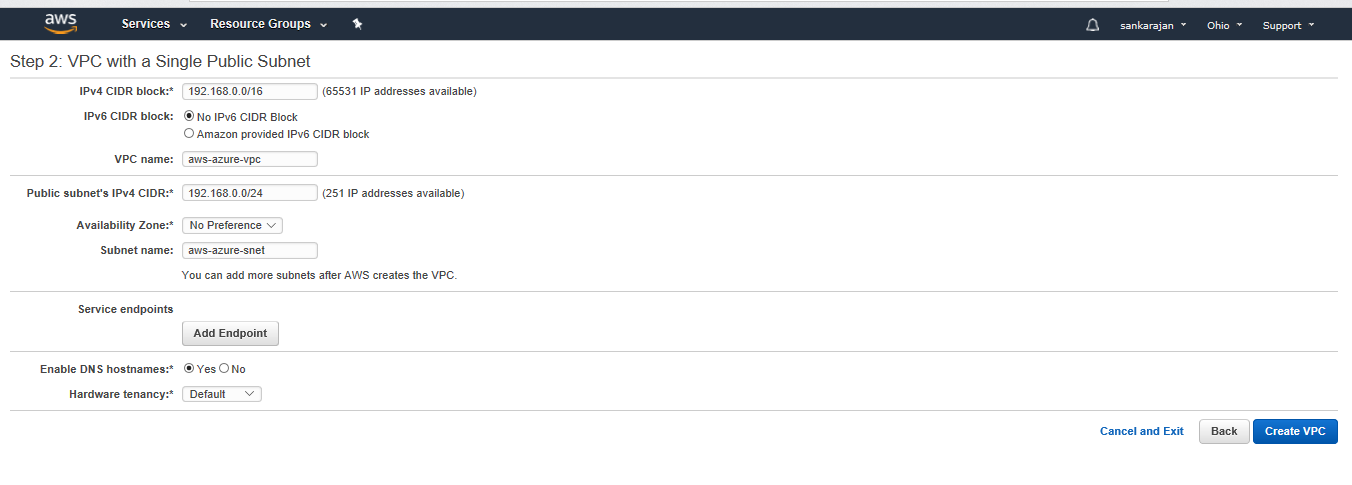
**Open Azure Portal and navigate to VPC**



**Create VPC and Subnet**

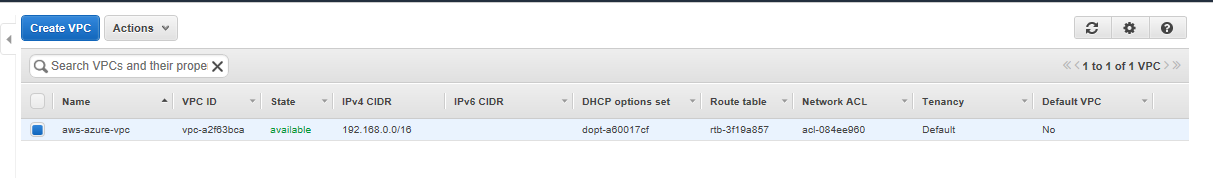


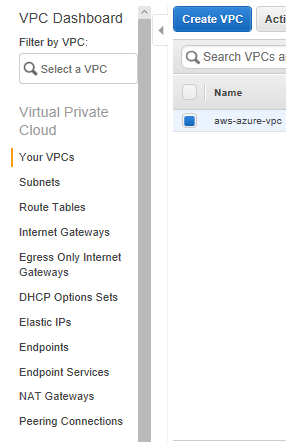




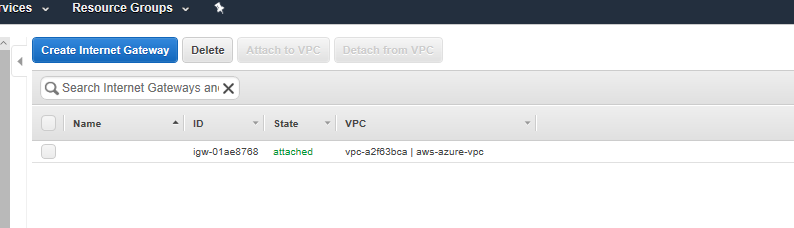
Create Subnet

Create VPC

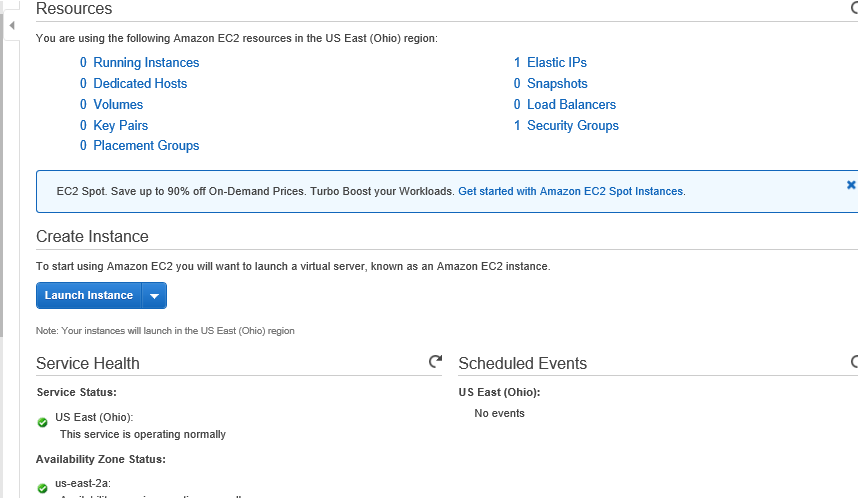


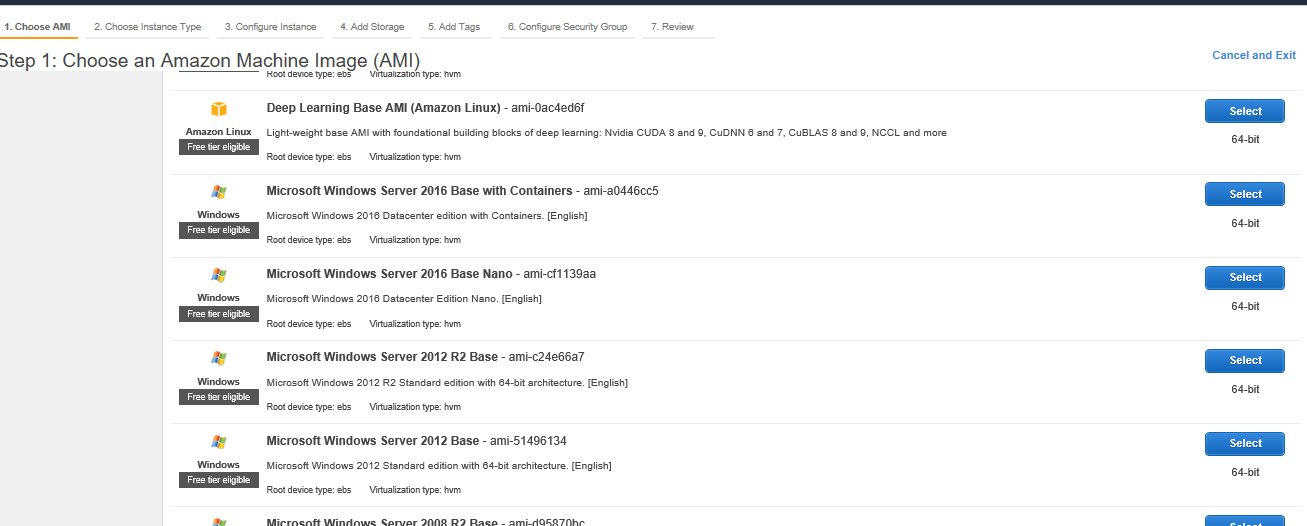


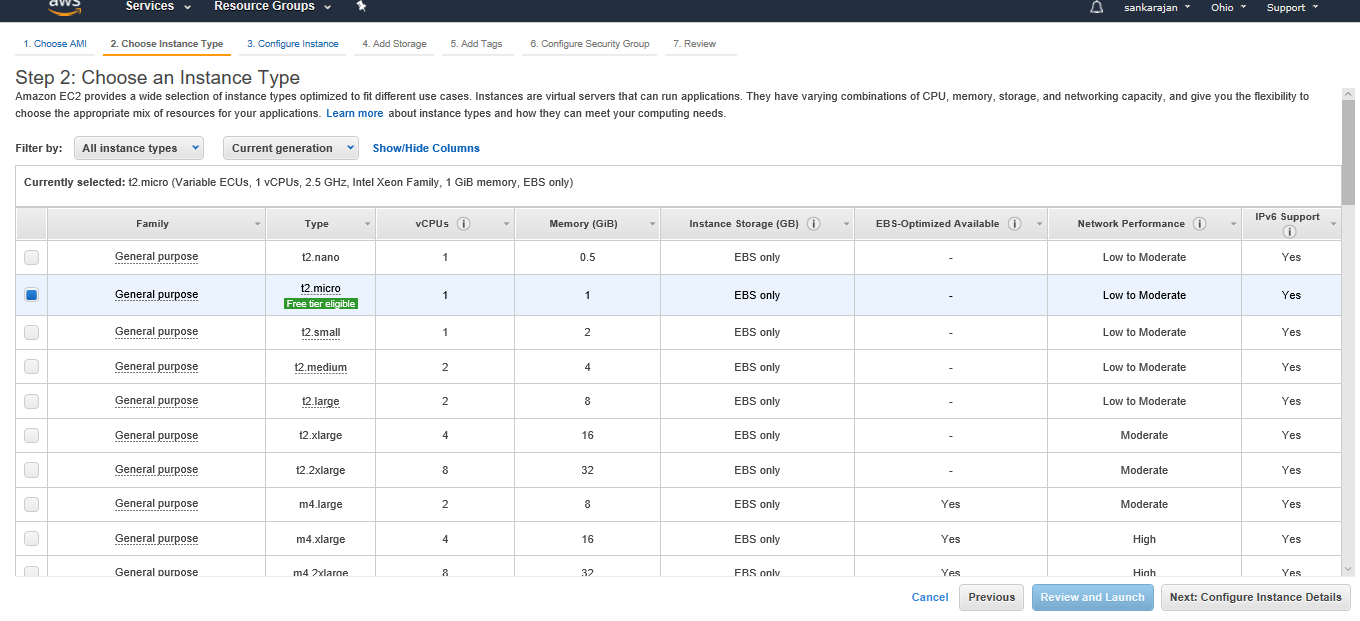
**Make sure that the Internet Gateway has been created and connected to VPC**

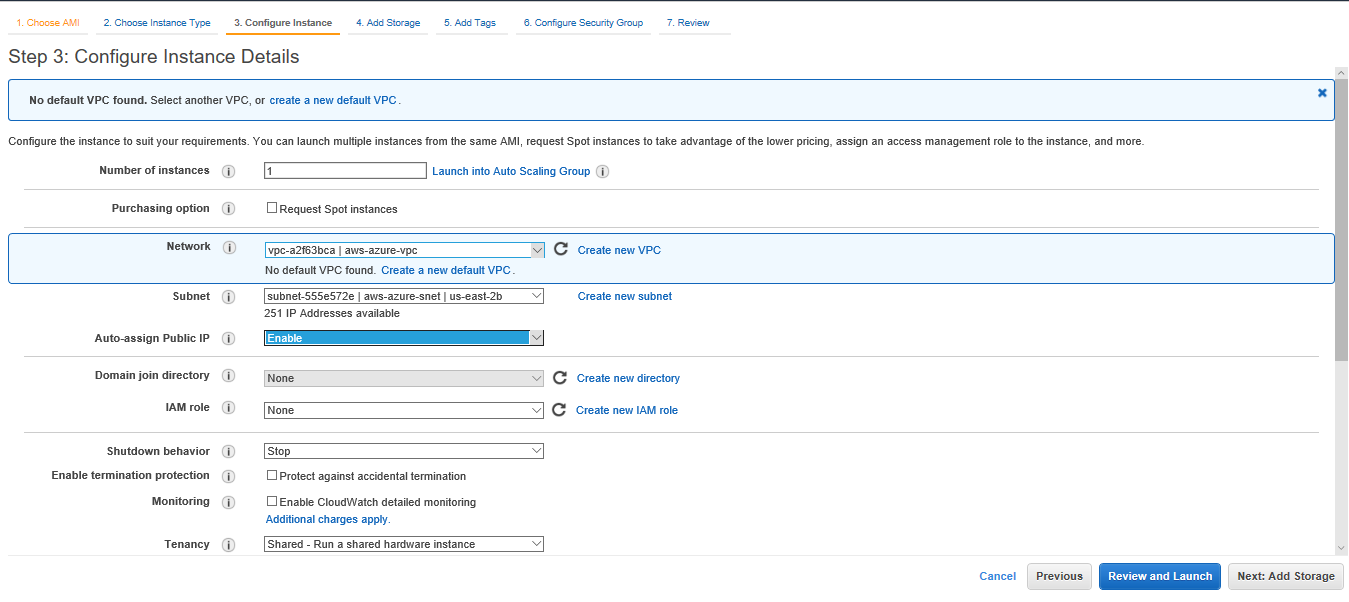


**Create a VM in AWS**







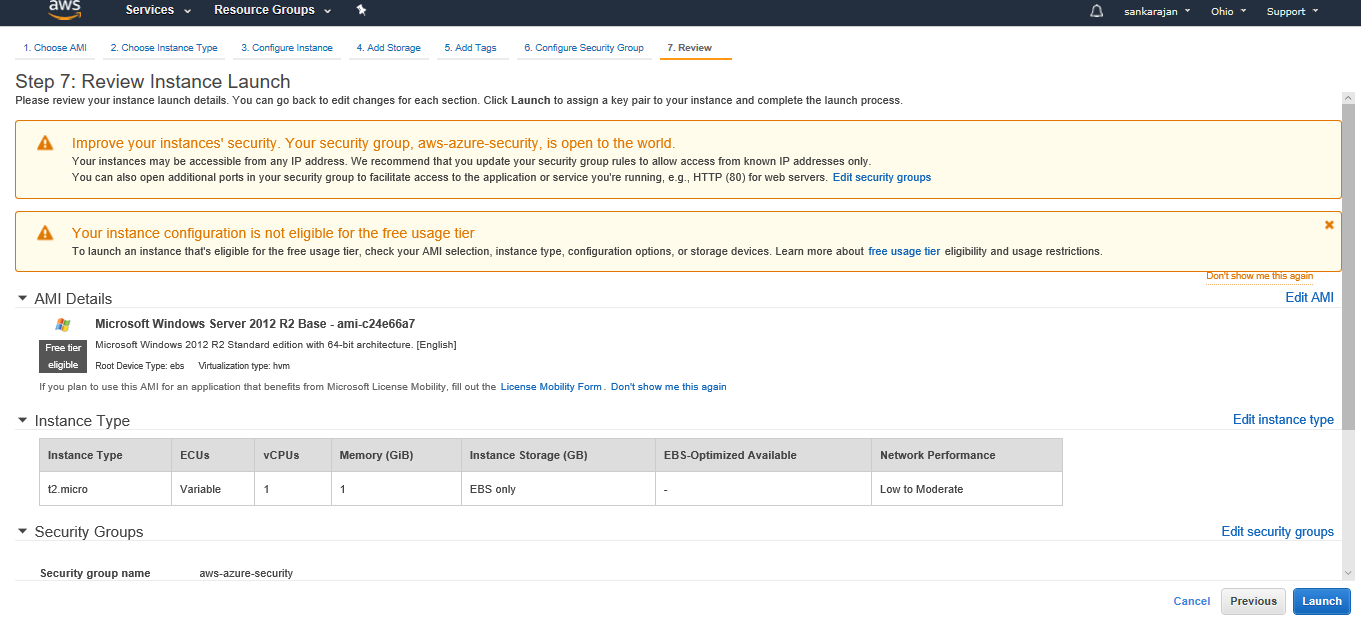


Associate the existing VPC

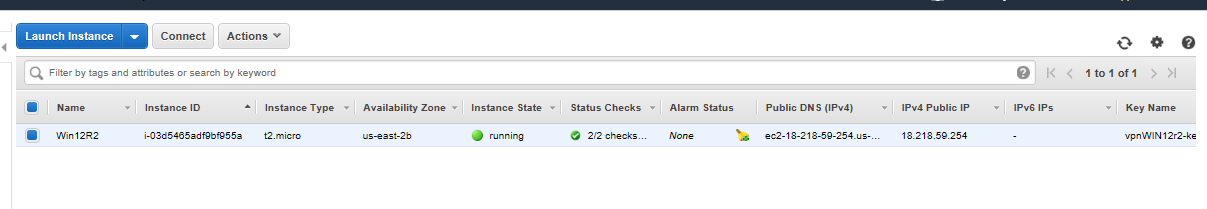
And Subnet that have been created

&

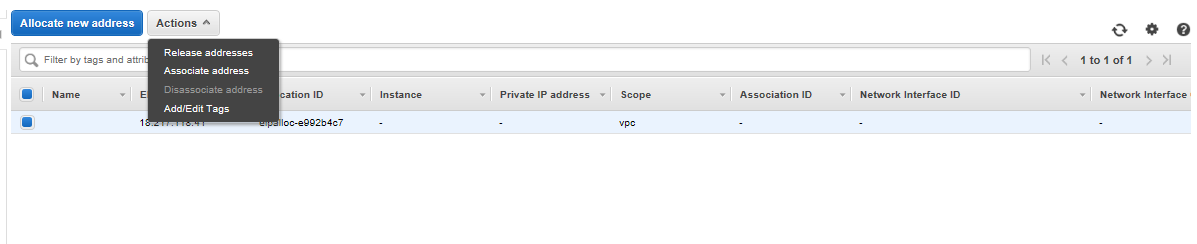
And enable Auto-assign Public IP

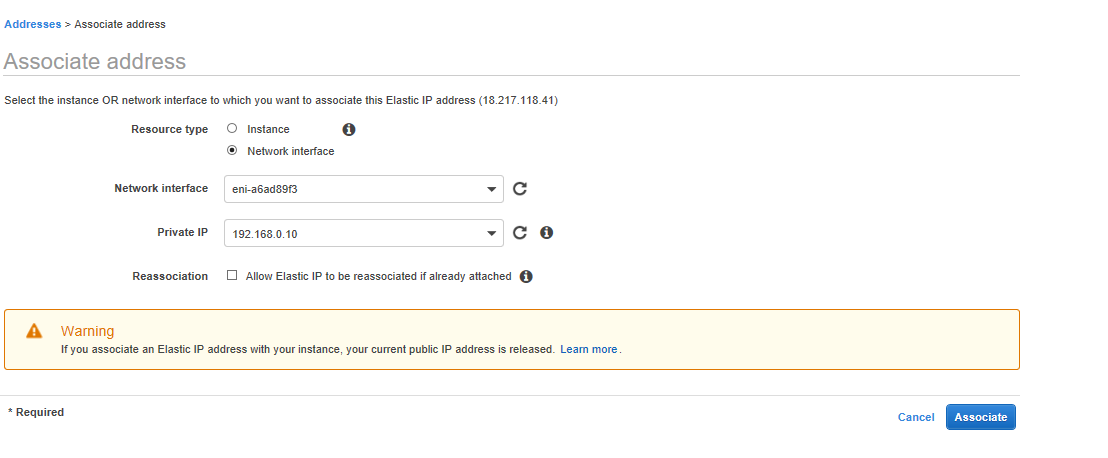


Launch the Instance

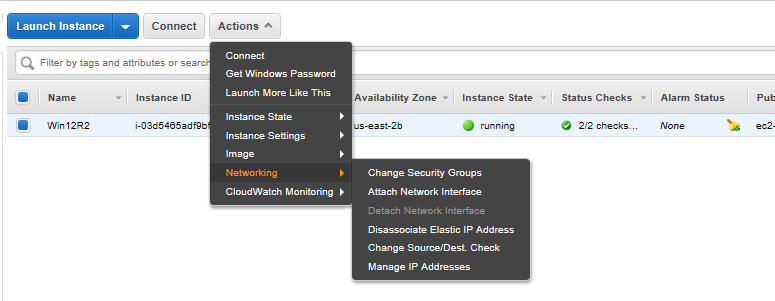


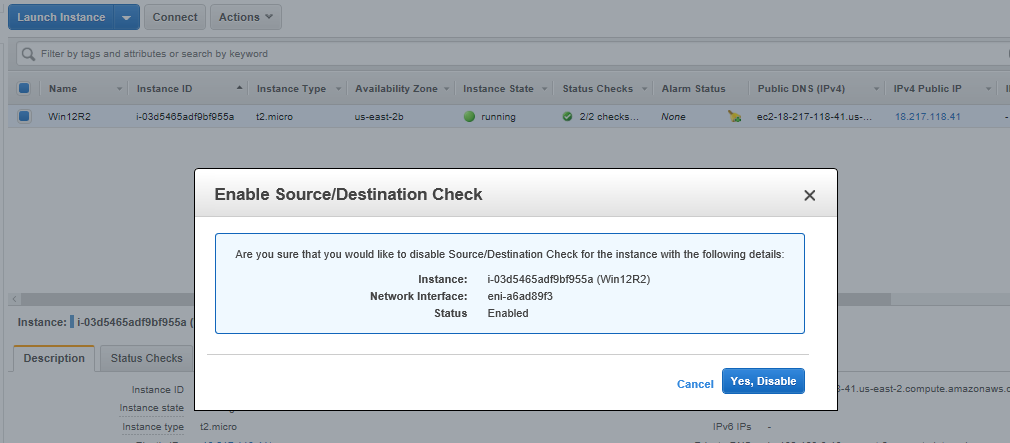
**Associate the Elastic IP to the VM’s Network Interface**





**Disable Source/Destination Check in the VM**





**Configure RRAS Server in the AWS VM**

Open AWS windows VM 🡪Open Powershell ISE and Run as Administrator

Copy the Below script and Change the Highlighted area

# Windows Azure Virtual Network

# This configuration template applies to Microsoft RRAS running on Windows Server 2012 R2.

# It configures an IPSec VPN tunnel connecting your on-premise VPN device with the Azure gateway.

# !!! Please notice that we have the following restrictions in our support for RRAS:

# !!! 1. Only IKEv2 is currently supported

# !!! 2. Only route-based VPN configuration is supported.

# !!! 3. Admin priveleges are required in order to run this script

Function Invoke-WindowsApi(

[string] $dllName,

[Type] $returnType,

[string] $methodName,

[Type[]] $parameterTypes,

[Object[]] $parameters

)

{

## Begin to build the dynamic assembly

$domain = [AppDomain]::CurrentDomain

$name = New-Object Reflection.AssemblyName 'PInvokeAssembly'

$assembly = $domain.DefineDynamicAssembly($name, 'Run')

$module = $assembly.DefineDynamicModule('PInvokeModule')

$type = $module.DefineType('PInvokeType', "Public,BeforeFieldInit")

$inputParameters = @()

for($counter = 1; $counter -le $parameterTypes.Length; $counter++)

{

$inputParameters += $parameters[$counter - 1]

}

$method = $type.DefineMethod($methodName, 'Public,HideBySig,Static,PinvokeImpl',$returnType, $parameterTypes)

## Apply the P/Invoke constructor

$ctor = [Runtime.InteropServices.DllImportAttribute].GetConstructor([string])

$attr = New-Object Reflection.Emit.CustomAttributeBuilder $ctor, $dllName

$method.SetCustomAttribute($attr)

## Create the temporary type, and invoke the method.

$realType = $type.CreateType()

$ret = $realType.InvokeMember($methodName, 'Public,Static,InvokeMethod', $null,

$null, $inputParameters)

return $ret

}

Function Set-PrivateProfileString(

$file,

$category,

$key,

$value)

{

## Prepare the parameter types and parameter values for the Invoke-WindowsApi script

$parameterTypes = [string], [string], [string], [string]

$parameters = [string] $category, [string] $key, [string] $value, [string] $file

## Invoke the API

[void] (Invoke-WindowsApi "kernel32.dll" ([UInt32]) "WritePrivateProfileString"

$parameterTypes $parameters)

}

# Install RRAS role

Import-Module ServerManager

Install-WindowsFeature RemoteAccess -IncludeManagementTools

Add-WindowsFeature -name Routing -IncludeManagementTools

# !!! NOTE: A reboot of the machine might be required here after which the script can be executed again.

# Install S2S VPN

Import-Module RemoteAccess

if ((Get-RemoteAccess).VpnS2SStatus -ne "Installed")

{

Install-RemoteAccess -VpnType VpnS2S

}

# Add and configure S2S VPN interface

Add-VpnS2SInterface -Protocol IKEv2 -AuthenticationMethod PSKOnly -NumberOfTries 3

-ResponderAuthenticationMethod PSKOnly -Name <Public IP of Azure gateway>

-Destination <Public IP of Azure gateway>

-IPv4Subnet @("<Address space of subnet in Azure>:100")

-SharedSecret <Shared Key>

Set-VpnServerIPsecConfiguration -EncryptionType MaximumEncryption

Set-VpnS2Sinterface -Name <Public IP of Azure gateway> -InitiateConfigPayload $false

-Force

# Set S2S VPN connection to be persistent by editing the router.pbk file (required admin priveleges)

Set-PrivateProfileString $env:windir\System32\ras\router.pbk “<Public IP of Azure gateway>” "IdleDisconnectSeconds" "0"

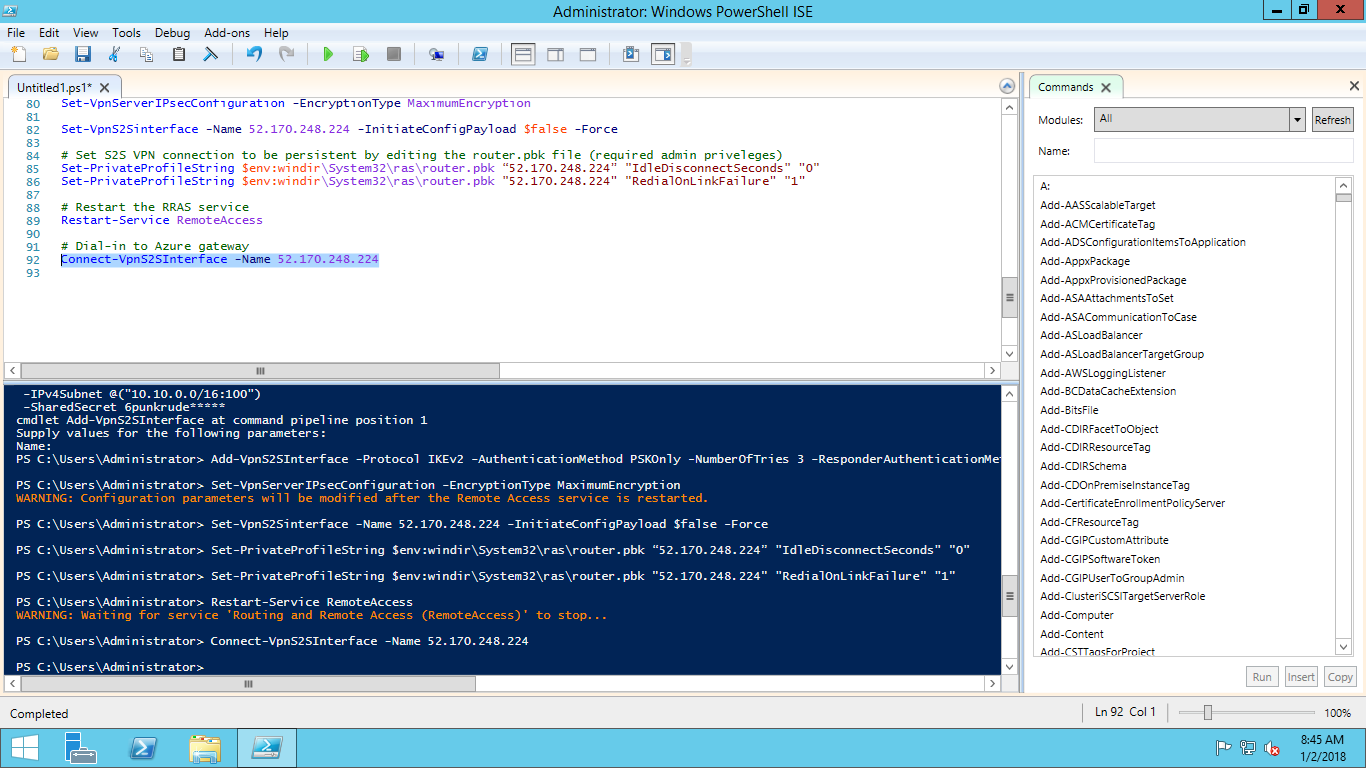
Set-PrivateProfileString $env:windir\System32\ras\router.pbk "<Public IP of Azure gateway>" "RedialOnLinkFailure" "1"

# Restart the RRAS service

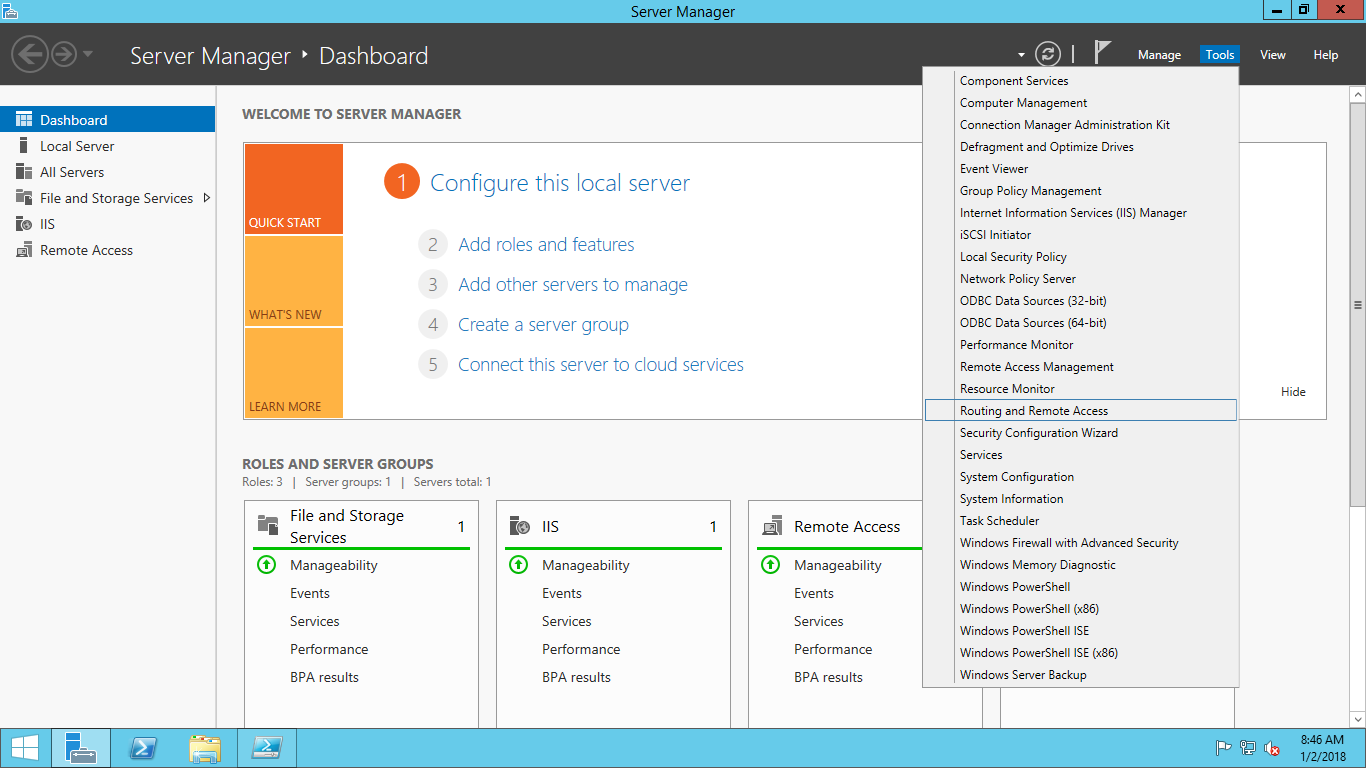
Restart-Service RemoteAccess

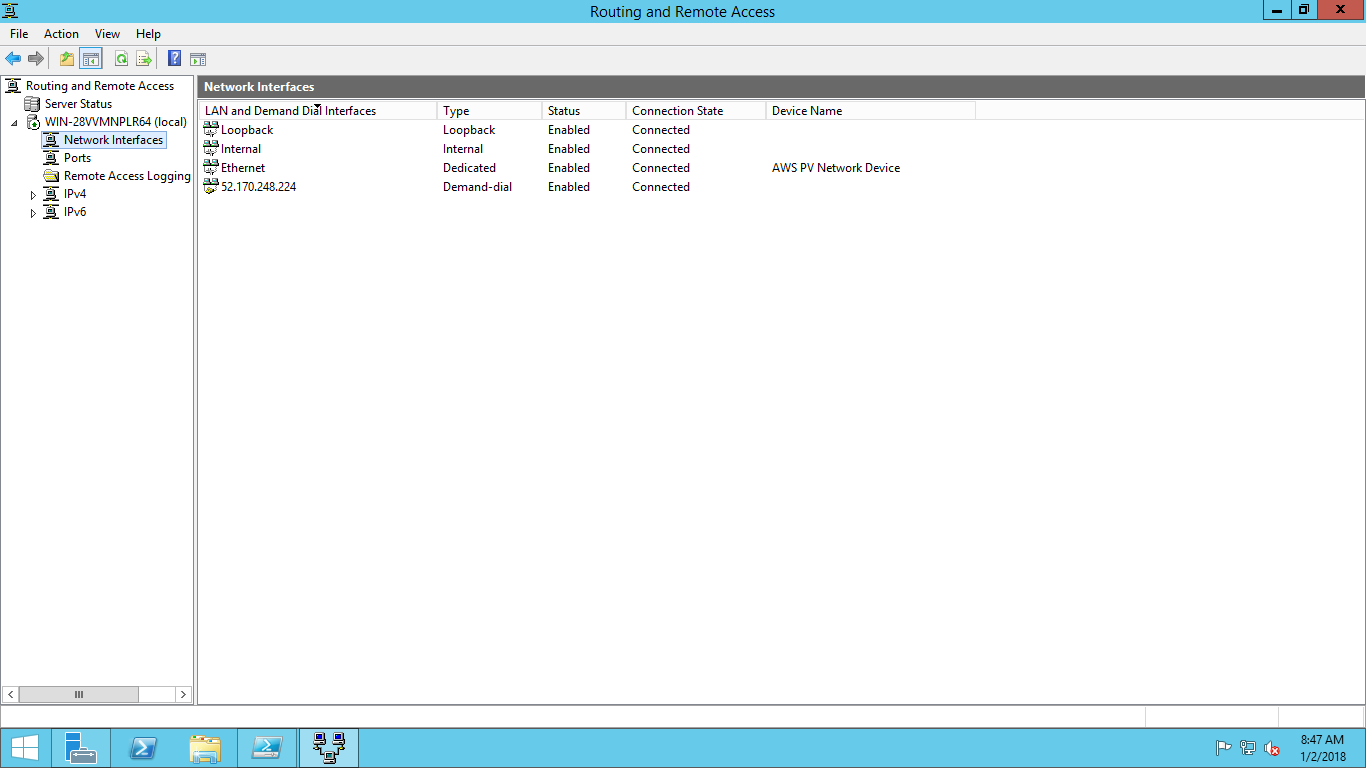
# Dial-in to Azure gateway

Connect-VpnS2SInterface -Name <Public IP of Azure gateway>



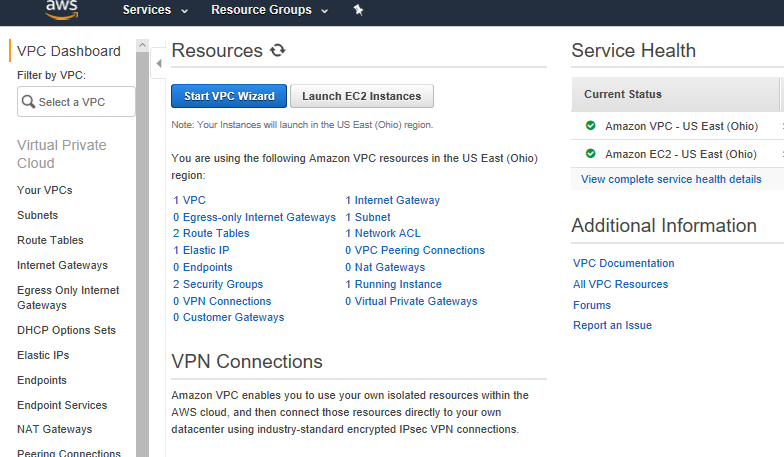
**Verify RRAS in Server Manager**



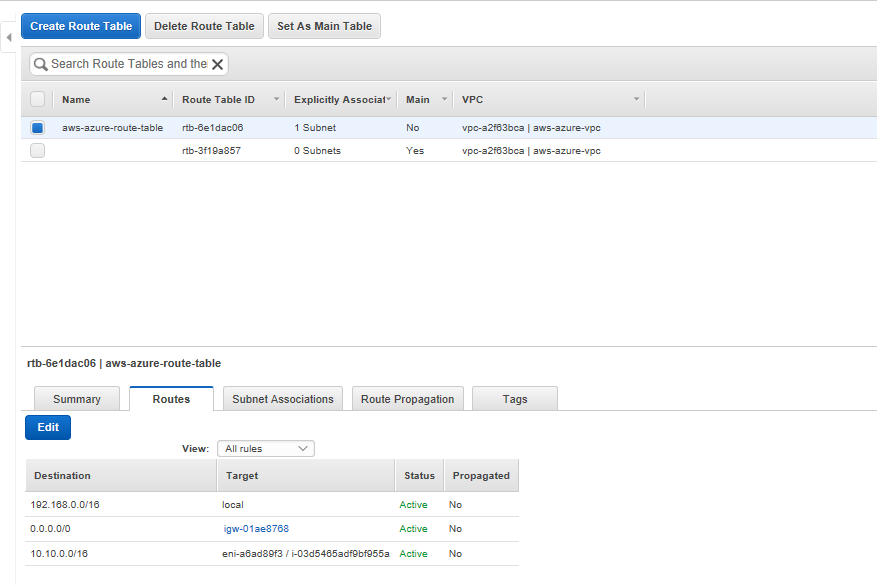


Make sure the Status is “Enabled” and Connection state is “Connected”

**Configure Route Tables**



**Create the Route Table**

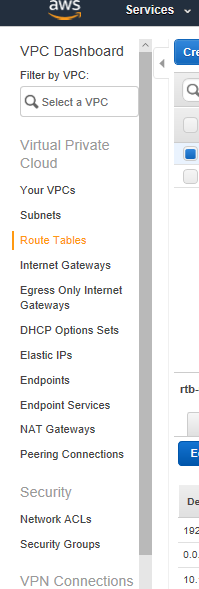


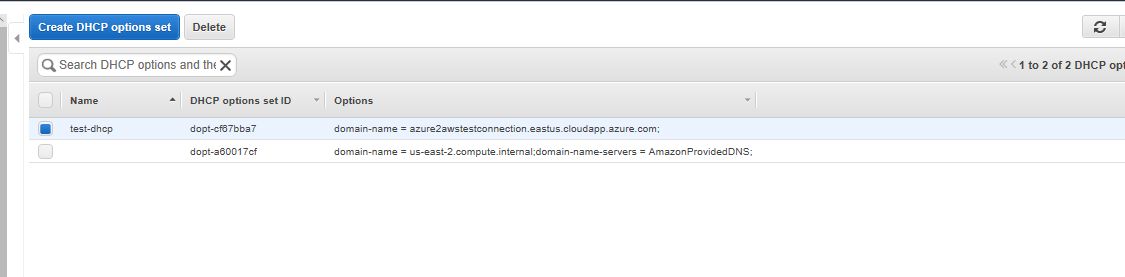
Click on edit and do the following

0.0.0.0/0 will be associated with Internet gateway

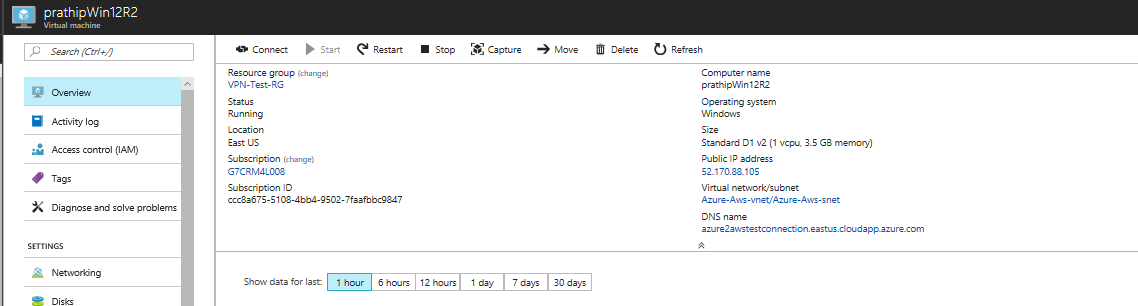
10.10.0.0/16 (Azure Vnet address space) will be associated with Network Interface of the VM

**Create DHCP Options Sets**





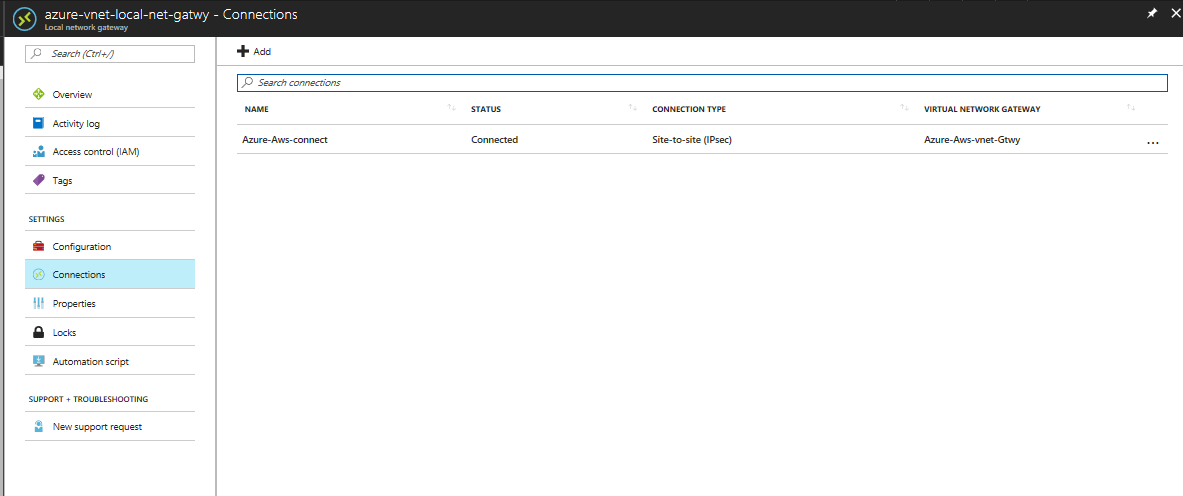
Assign Azure VM’s DNS Name as the Domain name while creating DHCP Option Sets



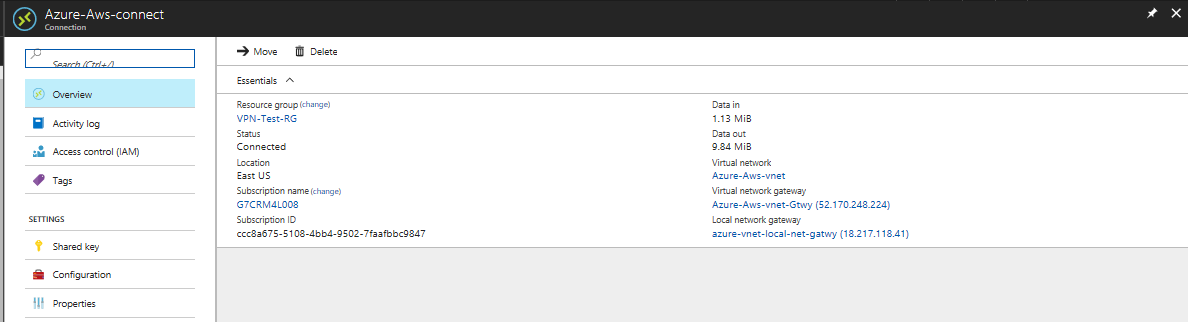
Dns Name of Azure VM

**Check the VPN Connection Status**

**Local Network Gateway**



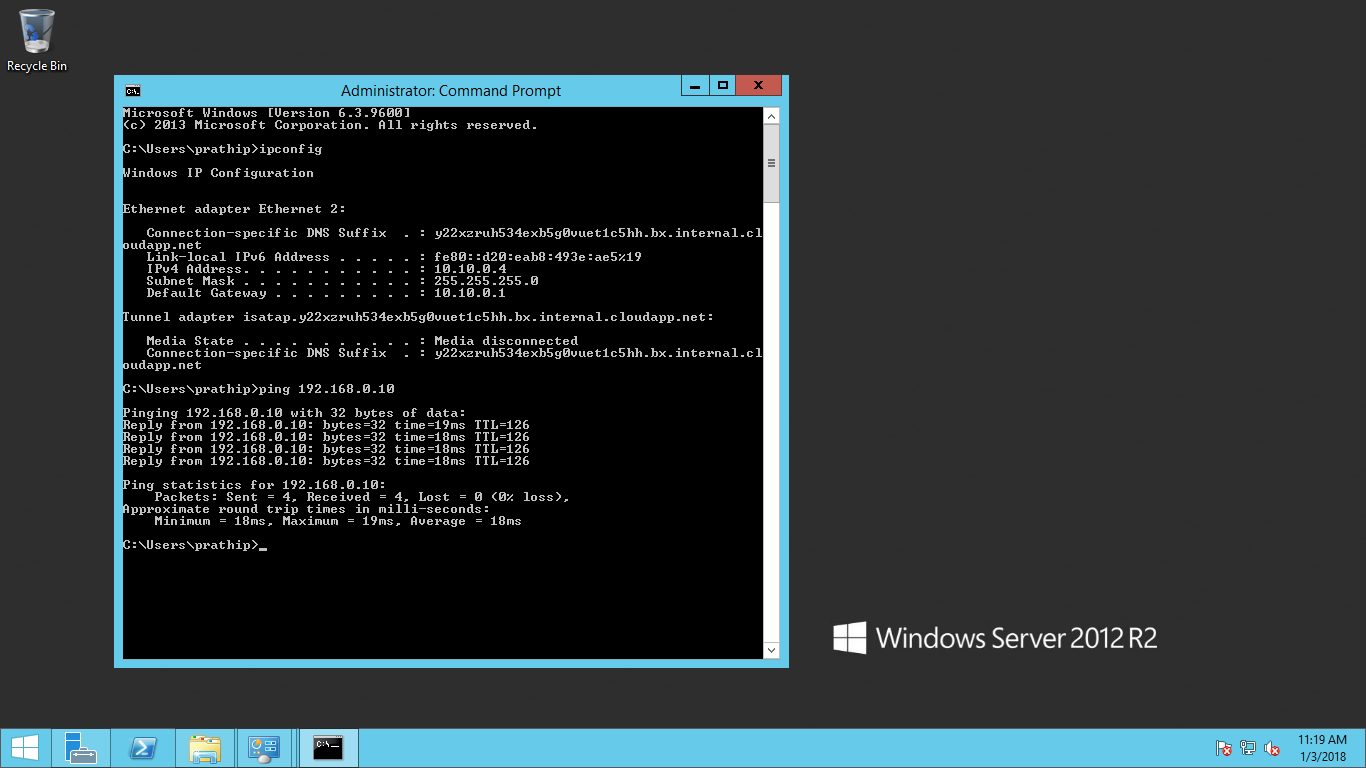
**VPN Connection**



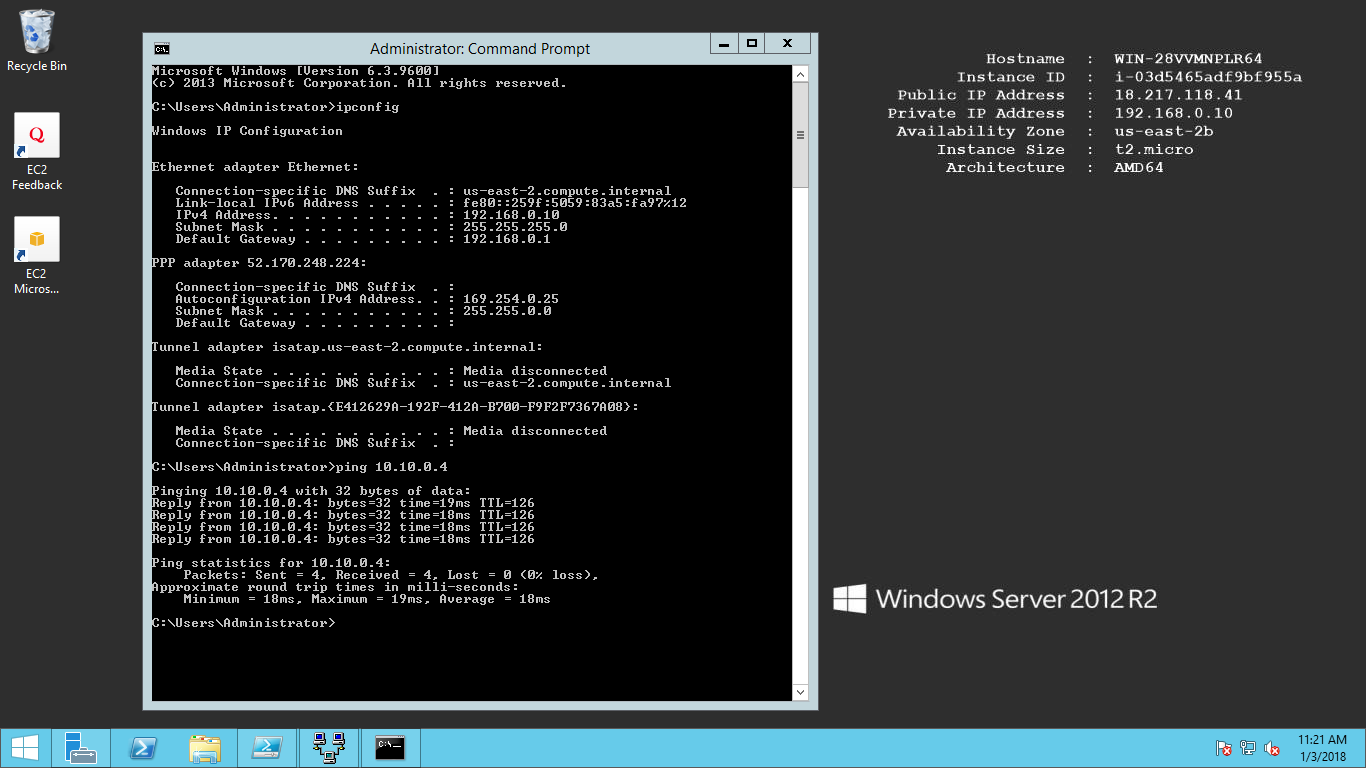
AWS VM Private IP : 192.168.0.10

Azure VM Private IP : 10.10.0.4

**Test Connectivity from Azure VM**



**Test Connectivity from AWS VM**



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*😊😊😊😊😊😊😊😊\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*