Azure VPN – (Virtual Private Network)

## **What is VPN (Virtual Private Network)?**

A VPN gateway is a type of virtual network gateway that sends encrypted traffic across a public connection. To send network traffic between your Azure virtual network and your on-premises site, you must create a virtual network gateway for your virtual network.

## **Why VPN is required?**

No organizations host all of their workloads in a public cloud like Microsoft Azure, most of the organizations have their own on-premises environments and probably it would decades to move all the applications to cloud, in such scenarios there has to be secured way of communication between public cloud like Azure and on-premises environments, which allows data exchange between cloud and on-premises data centers or private cloud like azure stack, this is called as hybrid cloud. If you’re building a hybrid cloud, you probably want to have network connectivity between the two clouds and that means a VPN. Microsoft Azure uses a Virtual Network Gateway to provide this connectivity.

Following are few scenarios of hybrid cloud

* A company uses a public development platform that sends data to a private cloud or a data center–based application.
* A company leverages a number of SaaS (Software as a Service) applications and moves data between private or data center resources.
* A business process is designed as a service so that it can connect with environments as though they were a single environment.

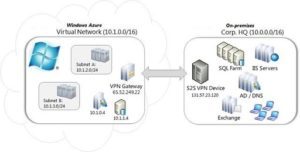
## **Different ways to create VPN Gateways**

To connect on-premises sites securely to a virtual network, three are different ways to do so.

The option can depend on various considerations, such as:

* What kind of throughput does your solution require?
* Do you want to communicate over the public Internet via secure VPN, or over a private connection?
* Do you have a public IP address available to use?
* Are you planning to use a VPN device? If so, is it compatible?
* Are you connecting just a few computers, or do you want a persistent connection for your site?
* What type of VPN gateway is required for the solution you want to create?
* Which gateway SKU should you use?

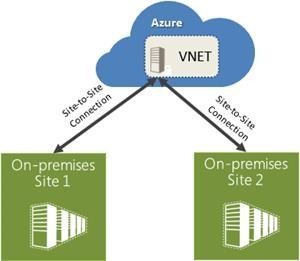
**Site-to-Site**  
A Site-to-Site (S2S) VPN gateway connection is a connection over IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has a public IP address assigned to it and is not located behind a NAT (Network Address Translation). S2S connections can be used for cross-premises and hybrid configurations.



Site to Site VPN

**Multi-Site**

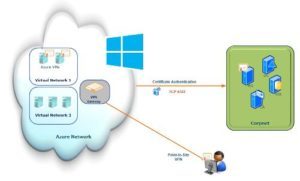
This type of connection is a variation of the Site-to-Site connection. You create more than one VPN connection from your virtual network gateway, typically connecting to multiple on-premises sites. When working with multiple connections, you must use a RouteBased VPN type (known as a dynamic gateway when working with classic VNets). Because each virtual network can only have one VPN gateway, all connections through the gateway share the available bandwidth. This is often called a “multi-site” connection.



Multisite VPN

**Point-to-Site (VPN over SSTP)**

A Point-to-Site (P2S) VPN gateway connection allows you to create a secure connection to your virtual network from an individual client computer. P2S is a VPN connection over SSTP (Secure Socket Tunneling Protocol). P2S connections do not require a VPN device or a public-facing IP address to work. You establish the VPN connection by starting it from the client computer. This solution is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a VNet. P2S connections can be used with S2S connections through the same VPN gateway, as long as all the configuration requirements for both connections are compatible.



Point to Site VPN

**VNet-to-VNet connections (IPsec/IKE VPN tunnel)**

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 a VPN gateway to provide a secure tunnel using IPsec/IKE. You can even combine VNet-to-VNet communication with multi-site connection configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

The VNets you connect can be:

* in the same or different regions
* in the same or different subscriptions
* in the same or different deployment models

VNet to VNet

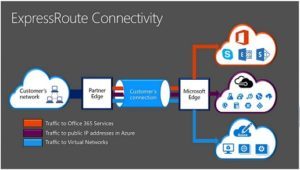
VNet to VNet

**ExpressRoute (dedicated private connection)**

Microsoft Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity provider. With ExpressRoute, you can establish connections to Microsoft cloud services, such as Microsoft Azure, Office 365, and CRM Online. Connectivity can be from any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a co-location facility.

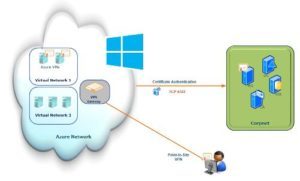
ExpressRoute connections do not go over the public Internet. This allows ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the Internet.

An ExpressRoute connection does not use a VPN gateway, although it does use a virtual network gateway as part of its required configuration. In an ExpressRoute connection, the virtual network gateway is configured with the gateway type ‘ExpressRoute’, rather than ‘Vpn’.



ExpressRoute

## **Point-to-Site (VPN over SSTP)**

Point to Site VPN

A Point-to-Site (P2S) VPN gateway connection allows you to create a secure connection to your virtual network from an individual client computer. P2S is a VPN connection over SSTP (Secure Socket Tunneling Protocol). P2S connections do not require a VPN device or a public-facing IP address to work. You establish the VPN connection by starting it from the client computer. This solution is useful when you want to connect to your VNet from a remote location, such as from home or a conference, or when you only have a few clients that need to connect to a VNet. P2S connections can be used with S2S connections through the same VPN gateway, as long as all the configuration requirements for both connections are compatible.

P2S connections require the following:

* A RouteBased VPN gateway.
* The public key (.cer file) for a root certificate, uploaded to Azure. This is considered a trusted certificate and is used for authentication.
* A client certificate generated from the root certificate, and installed on each client computer that will connect. This certificate is used for client authentication.
* A VPN client configuration package must be generated and installed on every client computer that connects. The client configuration package configures the native VPN client that is already on the operating system with the necessary information to connect to the VNet.

VPN gateways can be configured either through ARM Portal or Azure PowerShell.

This article follow Azure PowerShell to configure P2S connection.

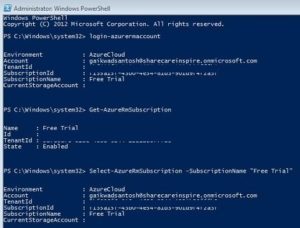
## **Step 1: Login and prepare variables**

1. Login

Login-AzureRmAccount

Get-AzureRmSubscription

Select-AzureRmSubscription -SubscriptionName "Name of subscription"

login

2. Prepare Variables for VNET

$VNetName  = "VNet1"

$FESubName = "FrontEnd"

$BESubName = "Backend"

$GWSubName = "GatewaySubnet"

$VNetPrefix1 = "192.168.0.0/16"

$VNetPrefix2 = "10.254.0.0/16"

$FESubPrefix = "192.168.1.0/24"

$BESubPrefix = "10.254.1.0/24"

$GWSubPrefix = "192.168.200.0/26"

$VPNClientAddressPool = "172.16.201.0/24"

$RG = "TestResourceG"

$Location = "West India"

$DNS = "8.8.8.8"

$GWName = "VNet1GW"

$GWIPName = "VNet1GWPIP"

$GWIPconfName = "gwipconf"

## **Step 2: Configure VNet**

1. Create new resource group

New-AzureRmResourceGroup -Name $RG -Location $Location

2. Create the subnet configurations for the virtual network, naming them FrontEnd, BackEnd, and GatewaySubnet

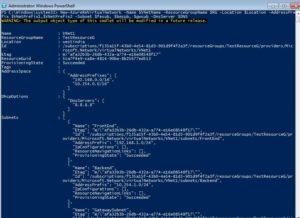
$fesub = New-AzureRmVirtualNetworkSubnetConfig -Name $FESubName -AddressPrefix $FESubPrefix

$besub = New-AzureRmVirtualNetworkSubnetConfig -Name $BESubName -AddressPrefix $BESubPrefix

$gwsub = New-AzureRmVirtualNetworkSubnetConfig -Name $GWSubName -AddressPrefix $GWSubPrefix

3. Create Virtual network

New-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG -Location $Location -AddressPrefix $VNetPrefix1,$VNetPrefix2 -Subnet $fesub, $besub, $gwsub -DnsServer $DNS

Create New Virtual Network

4. Read created virtual network in a variable

$vnet = Get-AzureRmVirtualNetwork -Name $VNetName -ResourceGroupName $RG

$subnet = Get-AzureRmVirtualNetworkSubnetConfig -Name "GatewaySubnet" -VirtualNetwork $vnet

5. Create Public IP address and set it to network gateway

$pip = New-AzureRmPublicIpAddress -Name $GWIPName -ResourceGroupName $RG -Location $Location -AllocationMethod Dynamic

$ipconf = New-AzureRmVirtualNetworkGatewayIpConfig -Name $GWIPconfName -Subnet $subnet -PublicIpAddress $pip

## **Step 3: Create Certificate using MakeCert**

Makecert is deprecated option, but as I am using windows 7, I had to go for it as PoewerShell module is not available for in windows 7.

1. Download and install [MakeCert](https://msdn.microsoft.com/library/windows/desktop/aa386968(v=vs.85).aspx).

2. Makecert by default got installed at location  “C:\Program Files\Microsoft SDKs\Windows\v7.1\Bin” on my machine, path might differ on your machine, so check during installation.

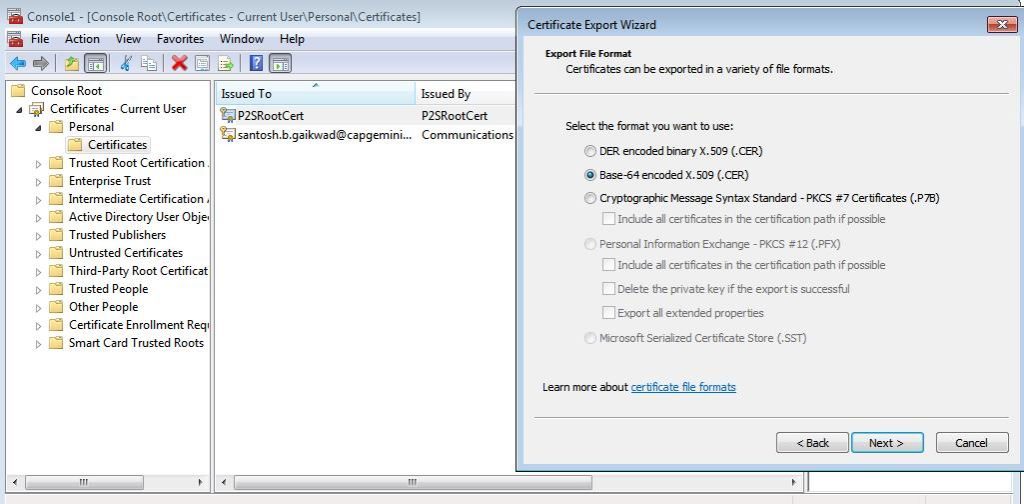
3. Run Makecert command to create certificate

makecert -sky exchange -r -n "CN=P2SRootCert" -pe -a sha1 -len 2048 -ss My "P2SRootCert.cer"

MakeCertificateMakeCertificate

4. Export public key and upload in Azure  
Open management console of certificates for current user, select certificates under personal folder and select “P2SRootCert”, right click and select export option. Select “No, do not export the private key”.

Save file as “P2SRootCer.cer” and store on local machine,  this file contains the public key of certificate, which needs to be uploaded to azure in further steps.

Export Public Key

5. Generate a client certificate  
Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. You generate a client certificate from the self-signed root certificate, and then export and install the client certificate. If the client certificate is not installed, authentication fails.

makecert -n "CN=P2SClientCert" -pe -sky exchange -m 96 -ss My -in "P2SRootCert" -is my -a sha1

Export client certificate to install on the client machines, Select “Yes, to export the private key”, Select “Include all certificates in the certification path if possible”

Export Client Certificate

6. Install an exported client certificate on client machine.  
If you want to create a P2S connection from a client computer other than the one you used to generate the client certificates, you need to install a client certificate. When installing a client certificate, you need the password that was created when the client certificate was exported. Typically, this is just a matter of double-clicking the certificate and installing it.

7. Upload server (root) certificate to Azure  
Prepare to upload the .cer file (which contains the public key information) for a trusted root certificate to Azure. You do not upload the private key for the root certificate to Azure. Once a .cer file is uploaded, Azure can use it to authenticate clients that have installed a client certificate generated from the trusted root certificate. Run following PowerShell.

$P2SRootCertName = "P2SRootCert.cer"

$filePathForCert = "D:\Santosh\Personal\Azure\Articles\P2SRootCert.cer"

$cert = new-object System.Security.Cryptography.X509Certificates.X509Certificate2($filePathForCert)

$CertBase64 = [system.convert]::ToBase64String($cert.RawData)

$p2srootcert = New-AzureRmVpnClientRootCertificate -Name $P2SRootCertName -PublicCertData $CertBase64

## **Step 4 – Create the VPN gateway**

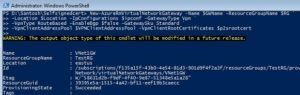
Configure and create the virtual network gateway for your Virtual Network. The -GatewayTypemust be **Vpn** and the -VpnType must be **RouteBased**.

New-AzureRmVirtualNetworkGateway -Name $GWName -ResourceGroupName $RG `

-Location $Location -IpConfigurations $ipconf -GatewayType Vpn `

-VpnType RouteBased -EnableBgp $false -GatewaySku Standard `

-VpnClientAddressPool $VPNClientAddressPool -VpnClientRootCertificates $p2srootcert

Create New Azure Virtual Network Gateway

## **Step 5 – Download the VPN client configuration package**

To connect to a VNet using a Point-to-Site VPN, each client must install a package to configure the native Windows VPN client. The configuration package configures the native Windows VPN client with the settings necessary to connect to the virtual network

You can use the same VPN client configuration package on each client computer, as long as the version matches the architecture for the client.

1. After the gateway has been created, you can generate and download the client configuration package. This example downloads the package for 32-bit clients. If you want to download the 64-bit client, replace ‘x86’ with ‘ Amd64’. You can also download the VPN client by using the Azure portal.

Get-AzureRmVpnClientPackage -ResourceGroupName $RG `

-VirtualNetworkGatewayName $GWName -ProcessorArchitecture x86

Download VPN Client PackageDownload VPN Client Package

2. Copy and paste the link that is returned to a web browser to download the package.

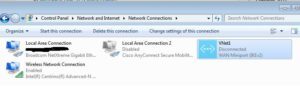
**JUST FOR REFERENCE,** In my case link is as follows, you would get different URL

https://mdsbrketwprodsn1prod.blob.core.windows.net/cmakexe/12fd7c84-4fcb-4bac-b5a5-2986ef94a5f3/x86/12fd7c84-4fcb-4bac-b5a5-2986ef94a5f3.exe?sv=2015-04-05&sr=b&sig=akCRd7TyTTEeUerJKdx9IE8K1BH%2BKghG%2BxhRYuVDcbo%3D&st=2017-06-07T09%3A30%3A50Z&se=2017-06-07T10%3A30%3A50Z&sp=r&fileExtension=.exe

You would be prompted to install VPN client, select Yes.

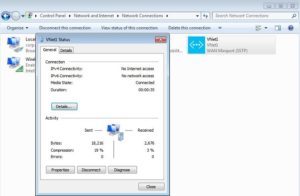
Install VPN Client

VPN get installed on client machine.

List of networks on client machine

## **Step 6 – Connect to Azure**

Right client VNet1 and click on connect, so that client machine gets connected to the P2S VPN.

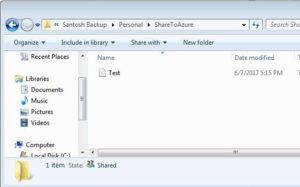
VPN Connected

You can run ipconfig /all command to check the IP details of machine.

IP of Client Machine

You can see that your machine is part of the Azure VNET now and it has got the IP address in the range of VNET of Azure.

Create a folder on client machine and share to everyone to check if shared folder can be accessed from Azure VM.

Share a folder on client machine

Run a PowerShell script to check the list of VMs available in Azure and get the IPs for each VM, and initiate mstsc command to open remote desktop connection.

$VMs = Get-AzureRmVM

$Nics = Get-AzureRmNetworkInterface | Where VirtualMachine -ne $null

foreach($Nic in $Nics)

{

$VM = $VMs | Where-Object -Property Id -eq $Nic.VirtualMachine.Id

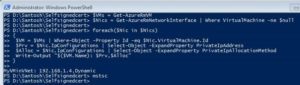
$Prv = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAddress

$Alloc = $Nic.IpConfigurations | Select-Object -ExpandProperty PrivateIpAllocationMethod

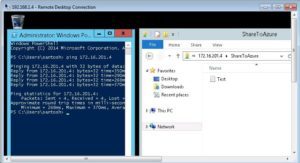
Write-Output "$($VM.Name): $Prv,$Alloc"

}

mstsc

Get the IPs of Azure VM and initiate Remote Desktop Connection

Log in to azure VM and ping local machine IP,  It gives back reply, open shared a folder of client machine and provide client credentials to connect to shared folder.

Login to Azure VM and Ping client machine and open shared folder of client machine