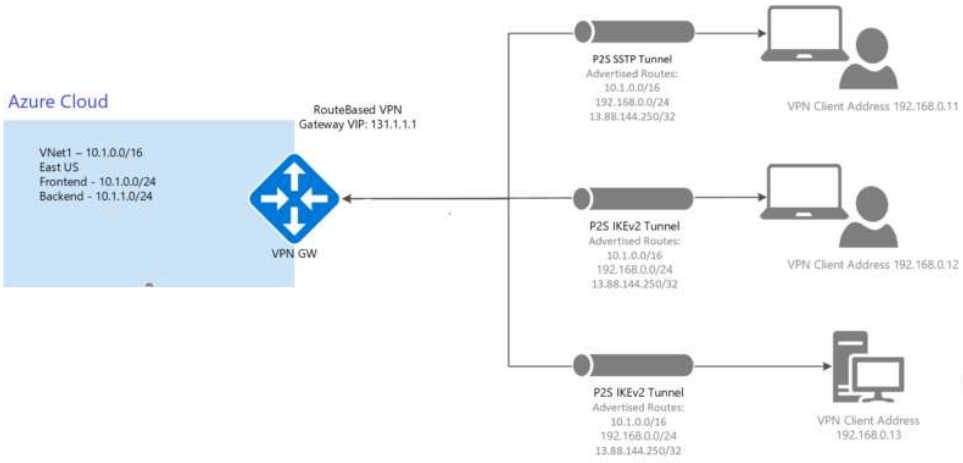
Overview on Azure Point to Site VPN

Azure Point-to-Site (P2S) VPN gateway connection lets you connect your individual client machine to Azure Network. A P2S connections established by starting it from client machine.



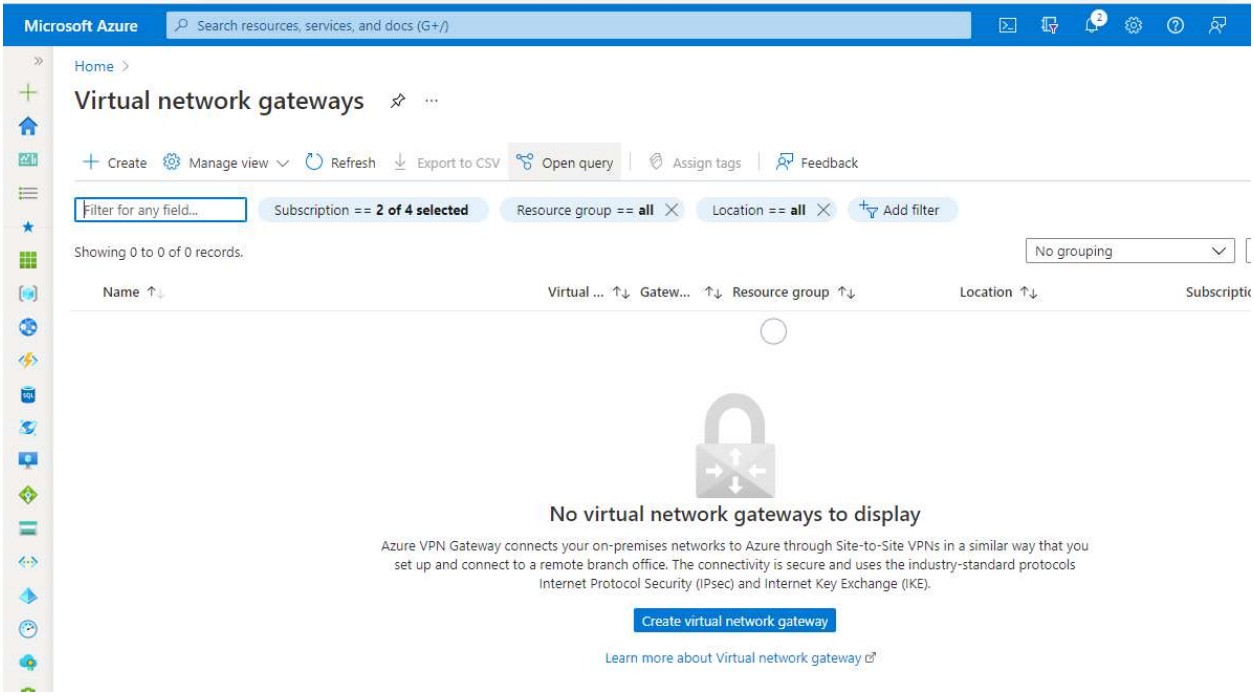
This solution helps in such scenario where end users are not part of corporate network and want to connect to Azure resources. Also, this can be used in scenario where number of users are limited so it's feasible to use rather than configuring Site to Site (S2S) VPN tunnel.

Prerequisite for P2S VPN

- Azure Subscription
- On-Prem IP Range (to avoid IP conflict issues)

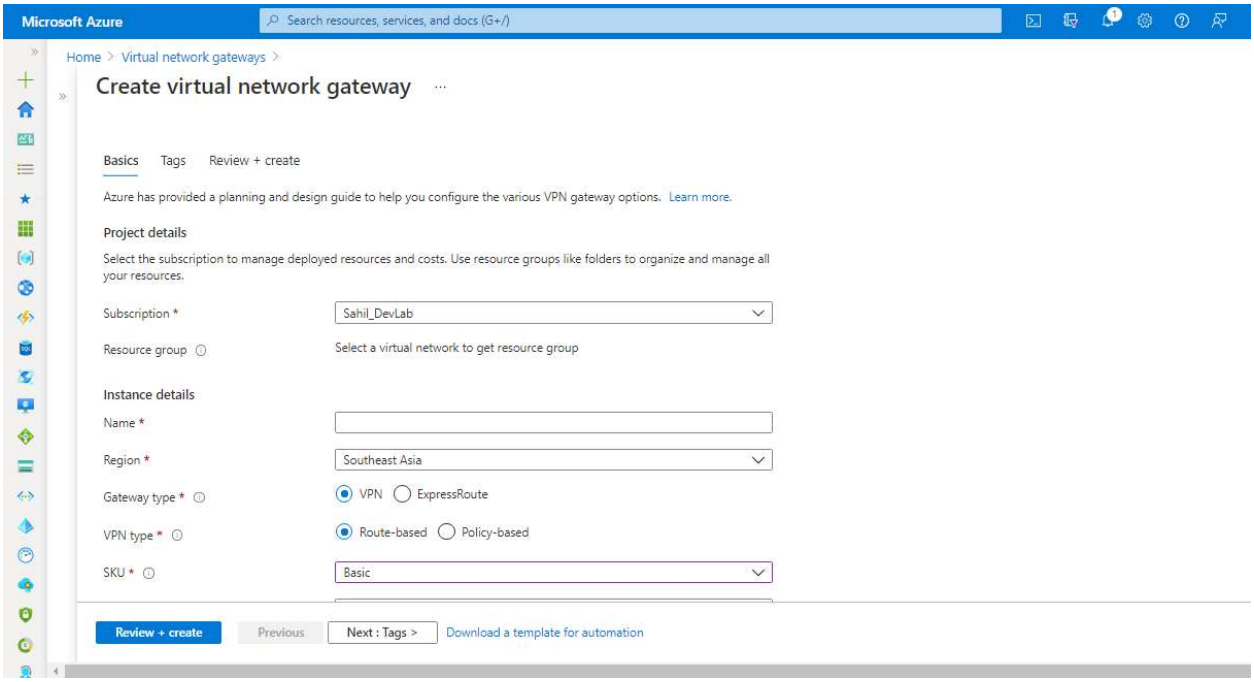
Step-1: Create Virtual Network Gateway on Azure (VPN)

Login to Azure Portal and search for Virtual Network Gateway.

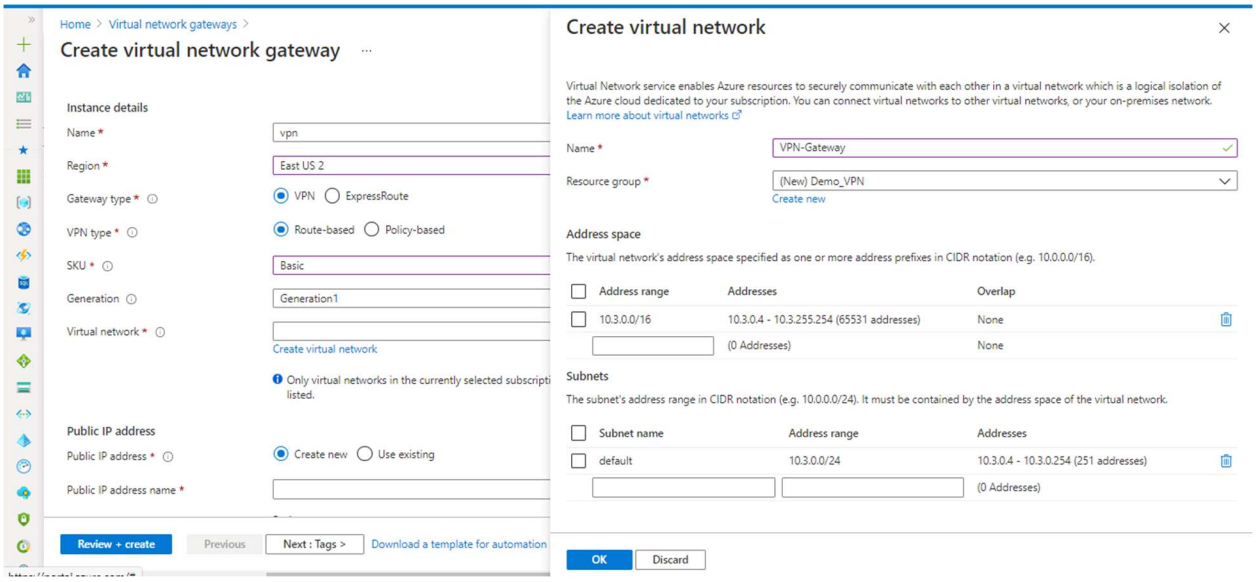


Click on create and follow the wizard.

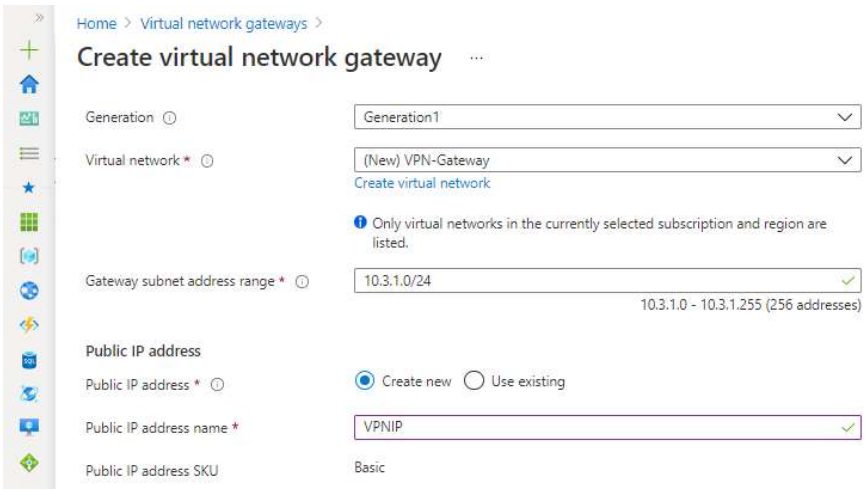
- **Subscription**: - Select your organization subscription (Subscription is logical ID assigned under your tenant)
- **Resource Group**: - Select resource group where you want to deploy this service (Resource Group is logical grouping of your resources)
- **Name**: - Name for your Virtual Network Gateway.
- **Region**: - Its geographical location where your datacenter is located. For testing purpose, you can choose East US region as its cheapest than others.
- **Gateway Type**: - VPN as we are using it for P2S.
- **VPN Type**: - Choose Route Based VPN.
- **SKU**: - For testing purpose select Basic SKU (Stock Keeping Unit). SKU will as per requirement Note: - Basic SKU only supports Windows Machines for P2S.
- **Generation**: - Select generation 2 latest.
- **Virtual Network**: - Azure Network which you want to talk with On-Prem Network.
- **Public IP address name**: Its allows you to give name for your Public IP . In Basic SKU we are using BASIC Public IP Address.



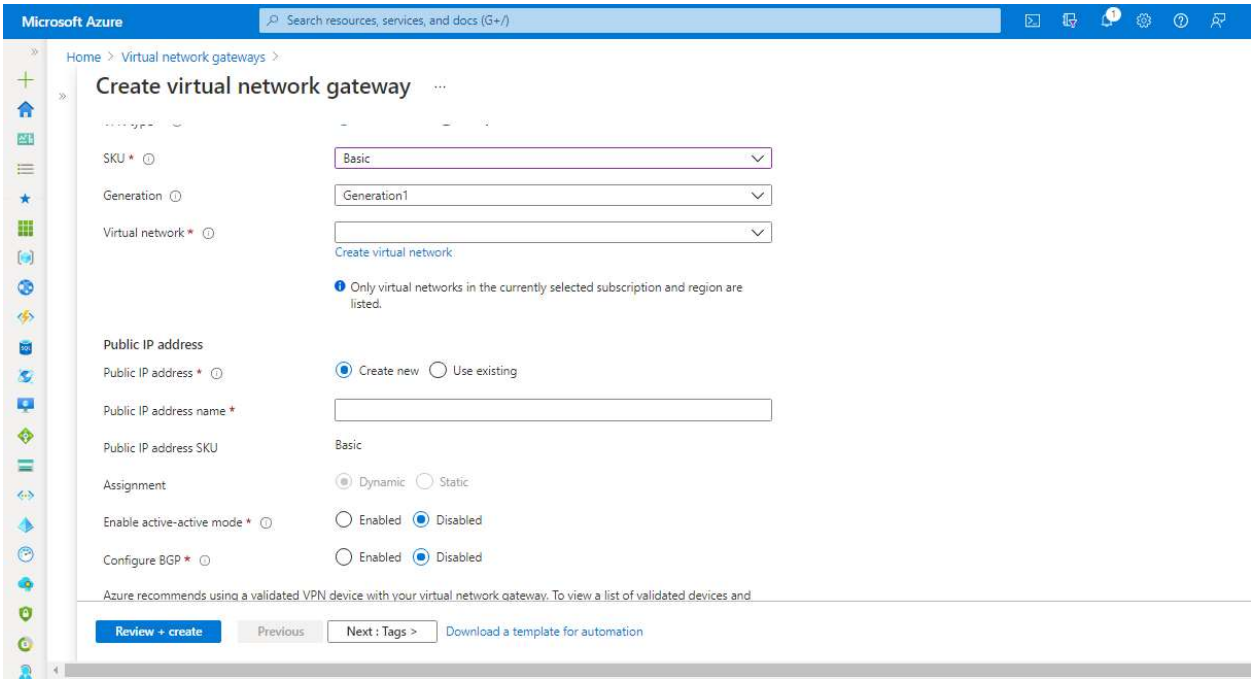
We can create Virtual Network while creating Virtual Network Gateway as per below image.



And after once you click on OK your gateway subnet would be automatically get create. or you can go to the Virtual Network resource and manually create Gateway Subnet as per the requirement.



Next, click on next option Tag your resources and review configuration and hit on create button.

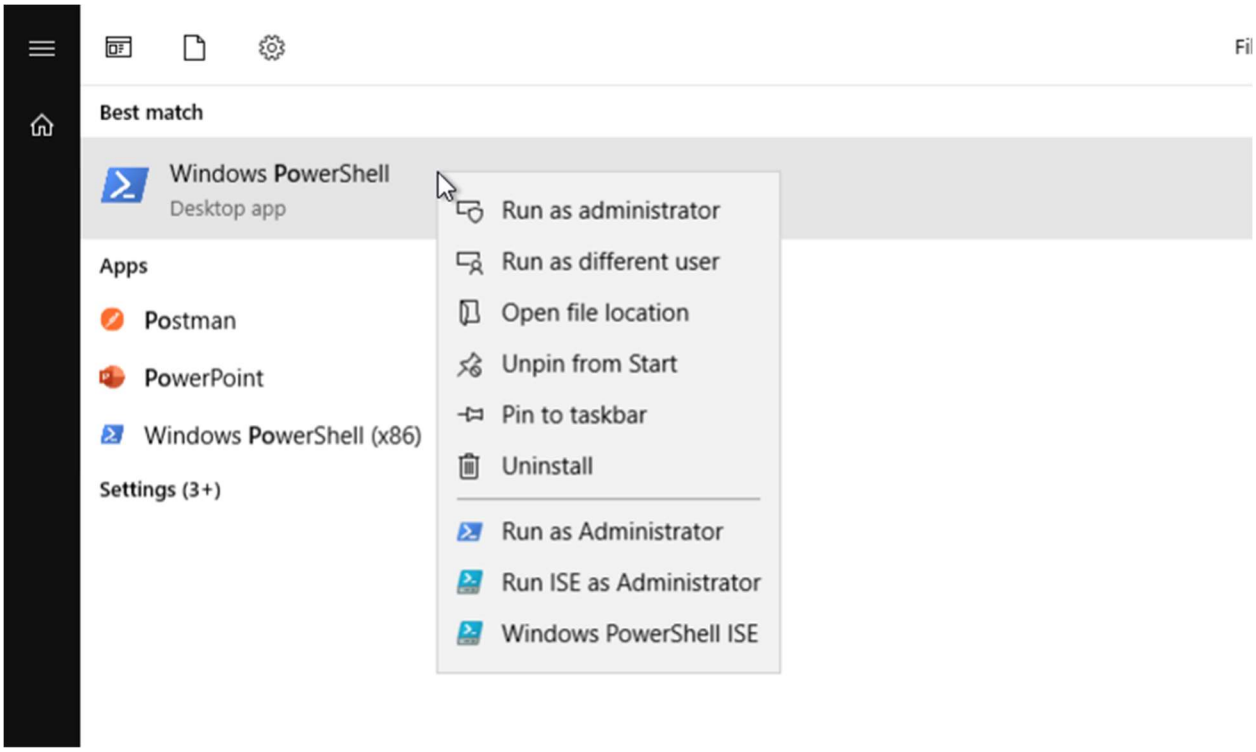


It will take approximately **35 min** to deploy your Virtual Network Gateway.

Step-2: Create Self-Signed certificate for P2S Connection

In this tutorial we are using Basic SKU for Virtual Network Gateway. Basic SKU supports only certificate-based authentication for P2S Connection. For that lets create Self-sign certificate on local system using PowerShell.

Open **Windows PowerShell** with administrative privileges as shown below:



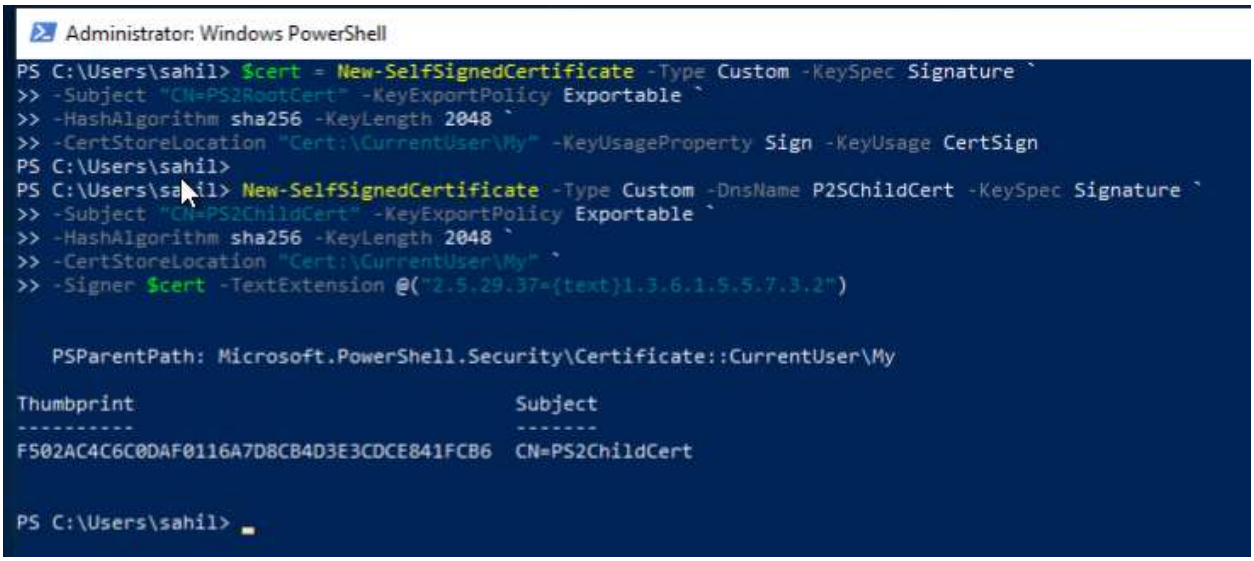
Execute the below command on the PowerShell:

This New-SelsignedCertificate module in windows PowerShell which help us to create Self-Signed certificates locally:

```
C:\Users\sahil> $cert = New-SelfSignedCertificate -Type Custom -KeySpec Signature
-Subject "CN=PS2RootCert" -KeyExportPolicy Exportable `
-HashAlgorithm sha256 -KeyLength 2048 `
-CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign -KeyUsage CertSign

C:\Users\sahil> New-SelfSignedCertificate -Type Custom -DnsName P2SChildCert -KeySpec
Signature `
-Subject "CN=PS2ChildCert" -KeyExportPolicy Exportable `
-HashAlgorithm sha256 -KeyLength 2048 `
-CertStoreLocation "Cert:\CurrentUser\My" `
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

Sample Output:



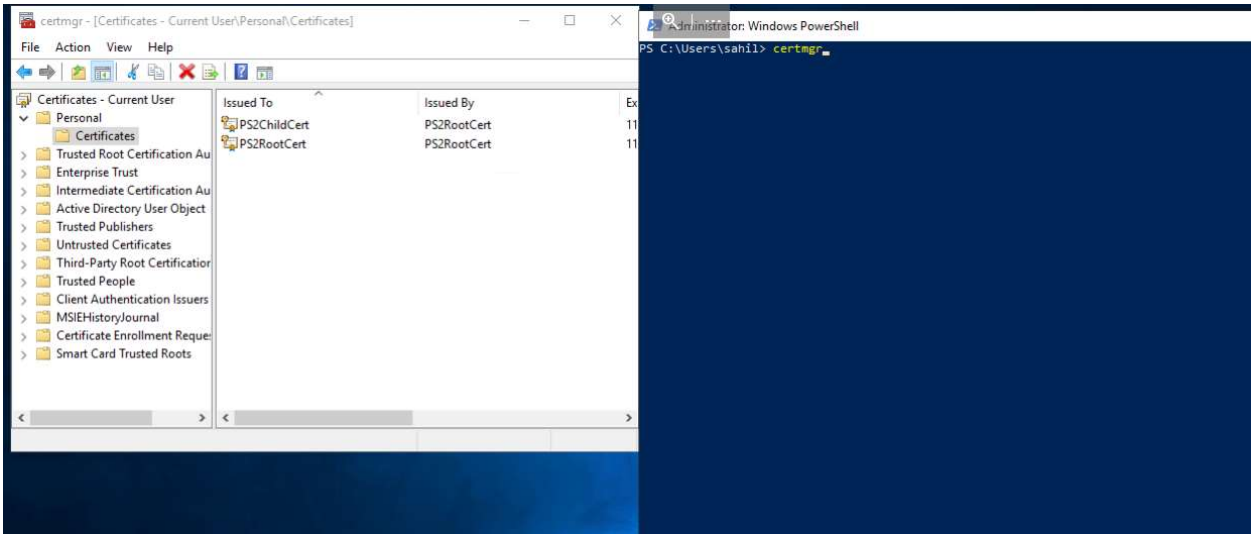
`

Here we are using **PS2RootCert** Name for our Public Key and **PS2ChildCert** for our private key you can rename name of certificate as per requirement. Here we are creating Root and Child certificate which will act as Public and Private key. Will upload Public Key to P2S VPN configuration and child certificate will share with end users for installing on local system. So, they can connect securely to Azure VPN.

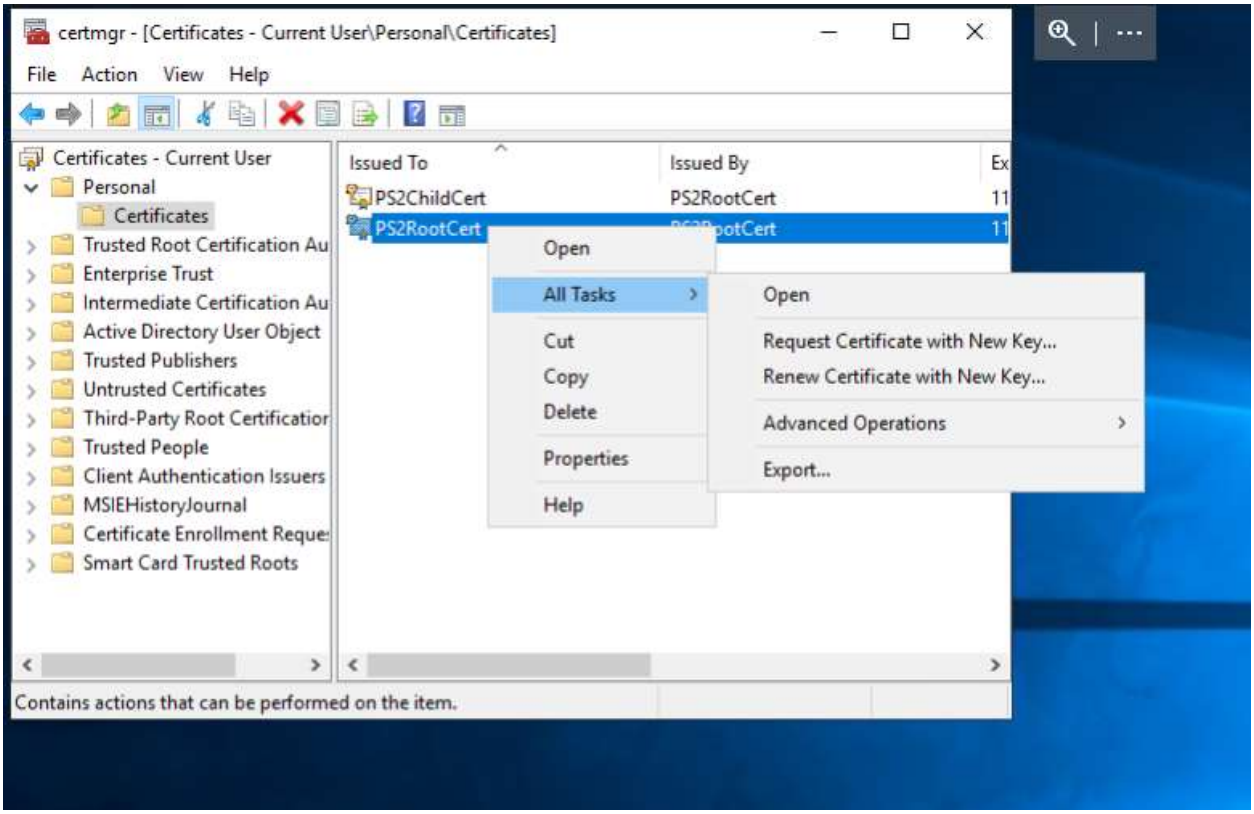
Now, Certificate has been created and stored that is automatically installed in “Certificates-CurrentUser\Personal\Certificates”.

Step-3: Export Root and Child certificates

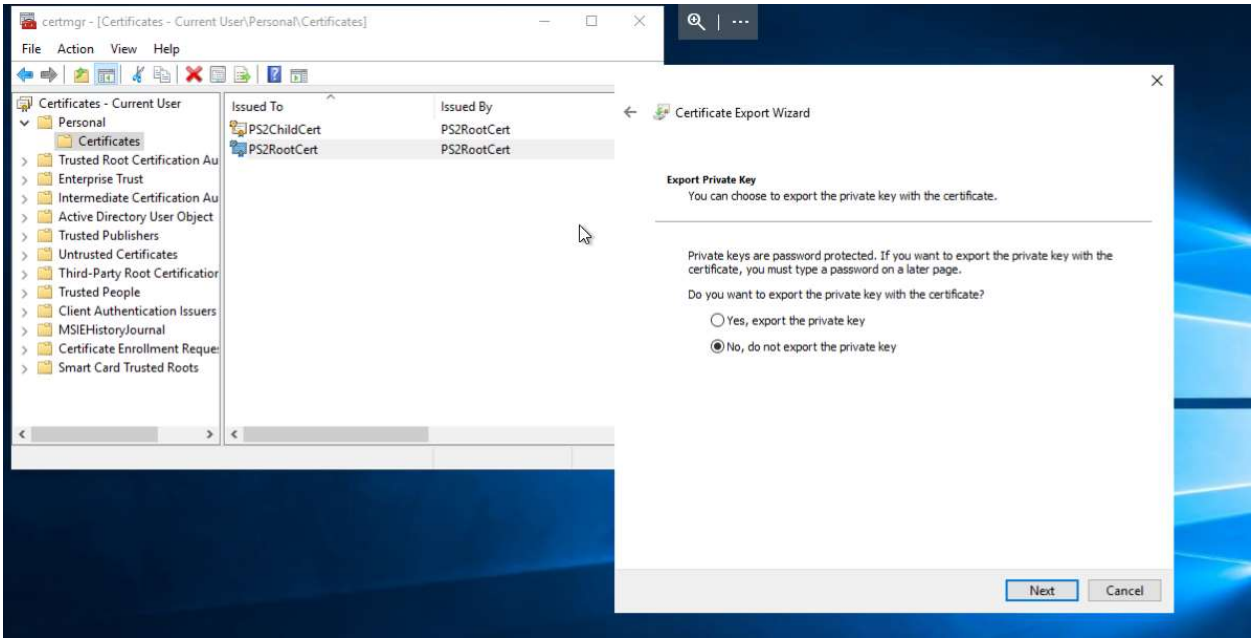
Now, Open Certificate manager to export Root and Child Certificates. Follow below step in PowerShell to open **Certmanager**.



Once PS2RootCert has been created and is visible in certmgr right-click on it. **Click All Tasks**, and then **click Export**. This opens the Certificate **Export Wizard**.

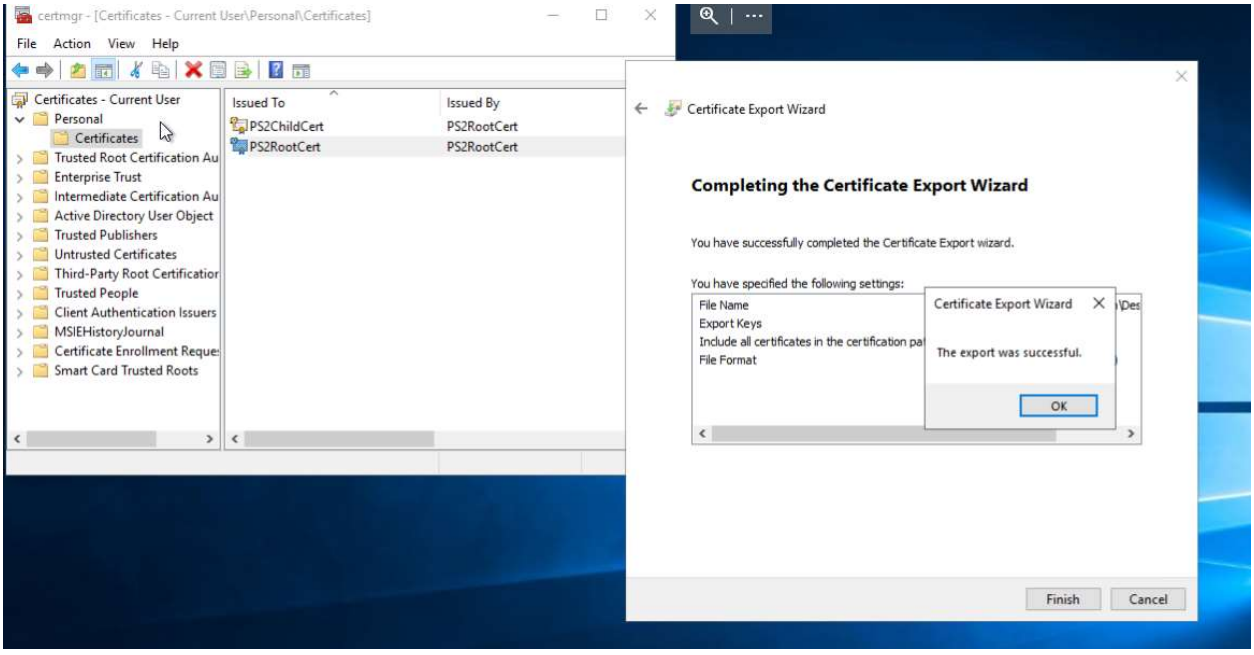


In the Wizard, click **Next** Select **No, do not export the private key** and then click **Next** On the Export File Format page, select **Base-64 encoded X.509 (.CER)**., and then click **Next.**

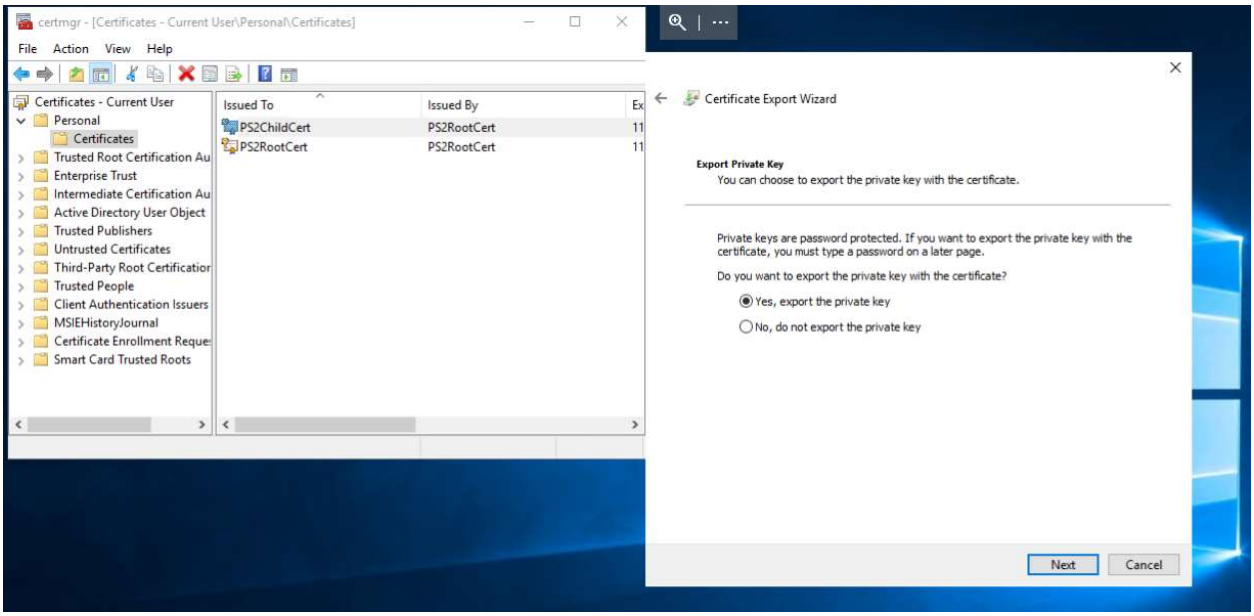


Select **Base-64 encoded X.509 (.CER).,** and then click **Next, f**or File to Export, browse to the location to which you want to export the certificate.

For **File name**, name the certificate file “PS2RootCert”. Then, click **Next** Click **Finish** to export the certificate.



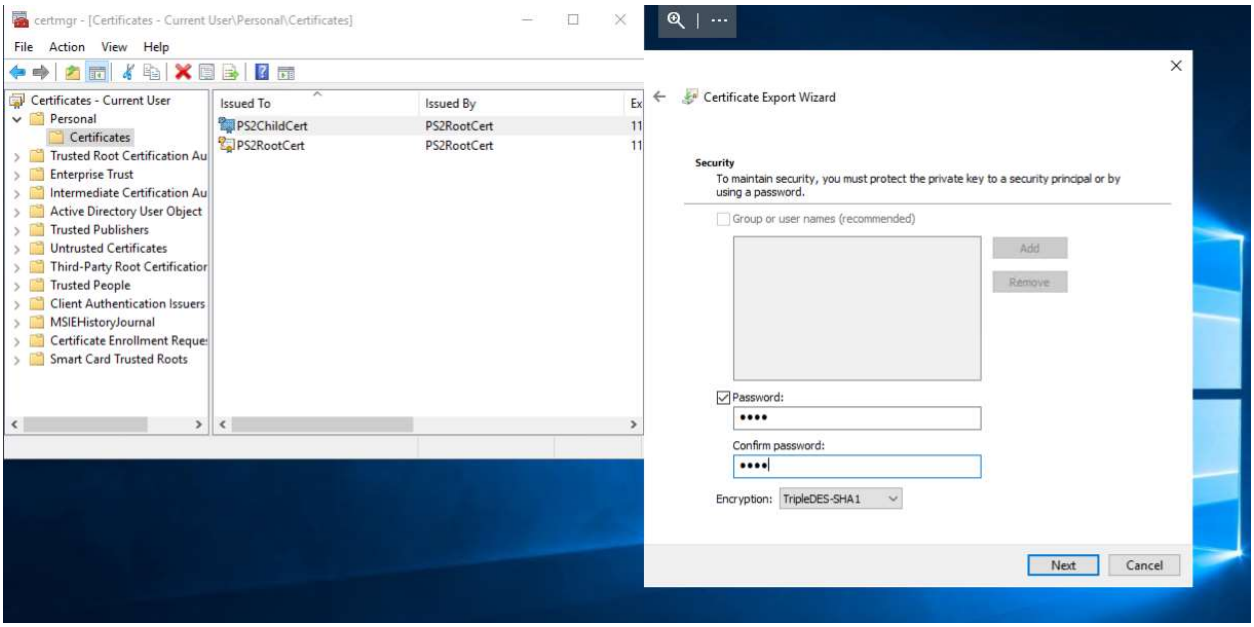
Now export the **PS2ChildCert** which has slightly different steps to follow than above.



In certmgr **right-click on PS2ChildCert**. Click All Tasks, and then **click Export**. This opens the Certificate Export Wizard.

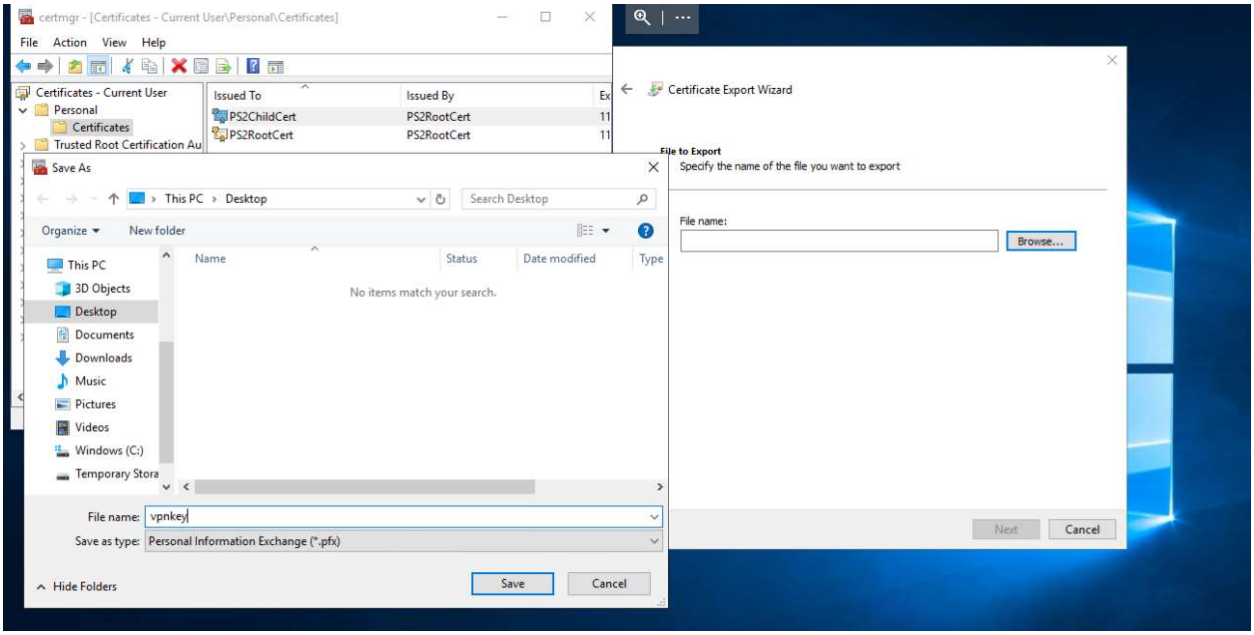
In the Certificate Export Wizard, click **Next** to continue Select **Yes, export the private key**, and then click **Next** On the **Export File Format** page, leave the defaults selected. Make sure that include all certificates in the certification path if possible is selected.

This setting additionally exports the root certificate information that is required for successful client authentication. Without it, client authentication fails because the client doesn't have the trusted root certificate.

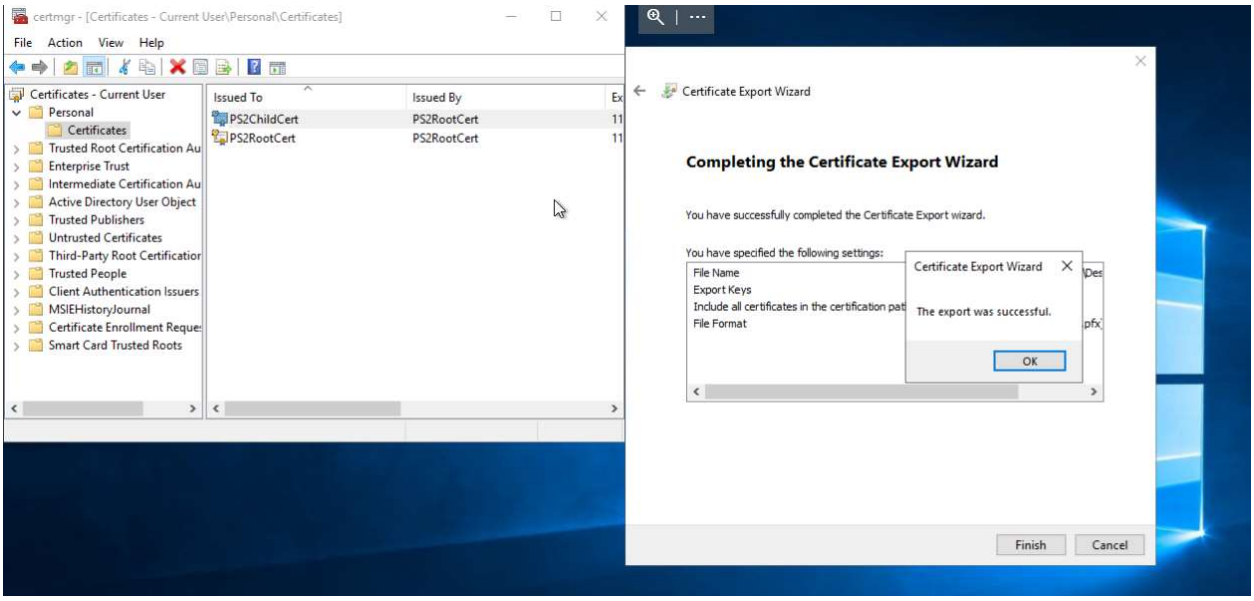


Then, click **Next** on the **Security page**, you must protect the private key. If you select to use a password, make sure to record or remember the password that you set for this certificate.

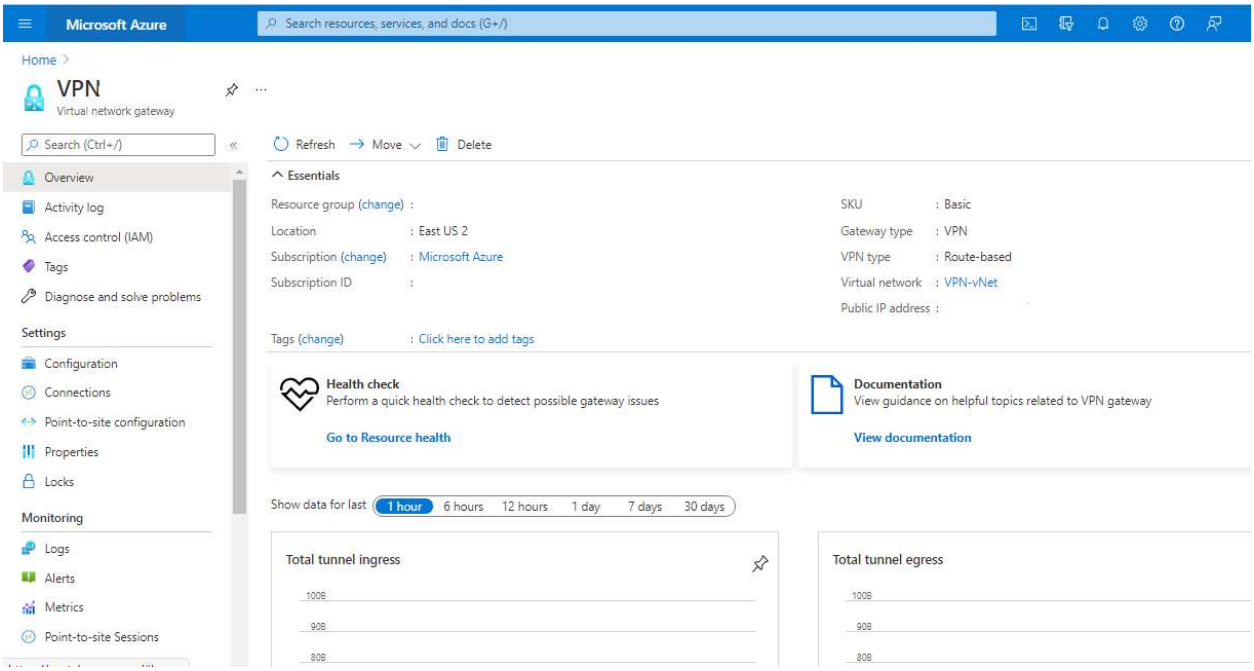
Then, click Next on the **File to Export**, browse to the location to which you want to export the certificate. For File name, name the certificate "PS2ChildCert"



Click **Next**. Click **Finish** to export the certificate.

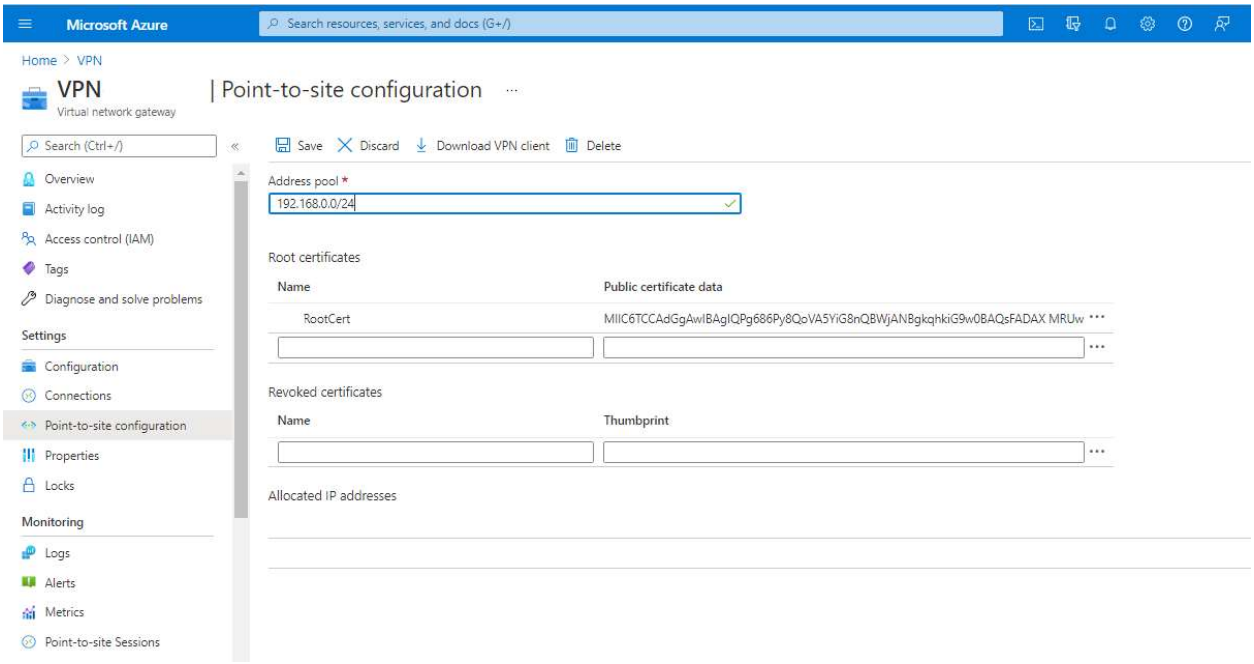


After deployment you will be able to see Virtual Network Gateway under your resource group. Now, we have successfully created Virtual Network gateway. Let's configure P2S connection.



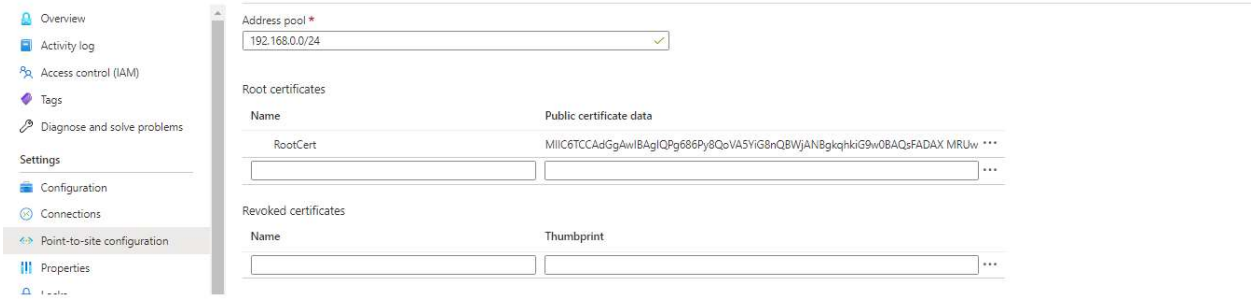
Step-4: Configure Azure Point to Site VPN

Let's start configuring P2S VPN connections. Go to the Virtual Network Gateway and click on Point to Site configuration.



Here, we are getting option to upload our root certificate. Now let's open the PSRootCert using notepad and copy only following section only.

Paste the certificate data into the Public Certificate Data field. Name the certificate "PS2RootCert", or if that name is already in use name the certificate "PS2RootCert_YourUserName", and Save. You can add up to 20 trusted root certificates.



Select Save at the top of the page to save all the configuration settings.

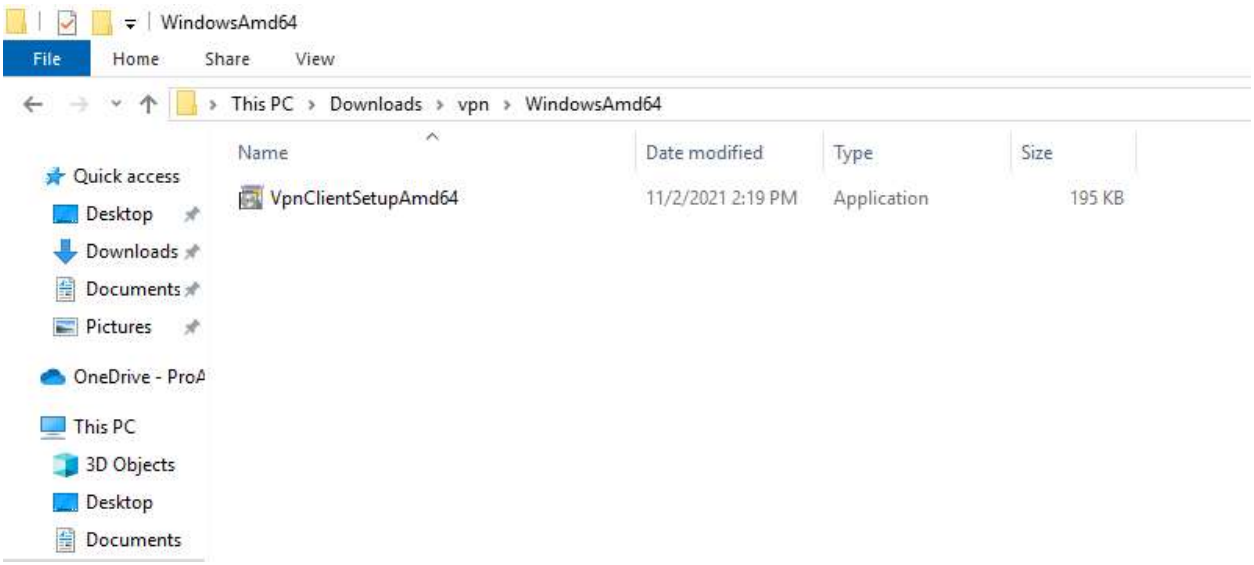
It will take few minutes to save the configuration and once its saved. You will get option to Download VPN Client.

P2S Configuration is done from Azure Portal. Now, Let's move to client machine to install VPN client and check connectivity.

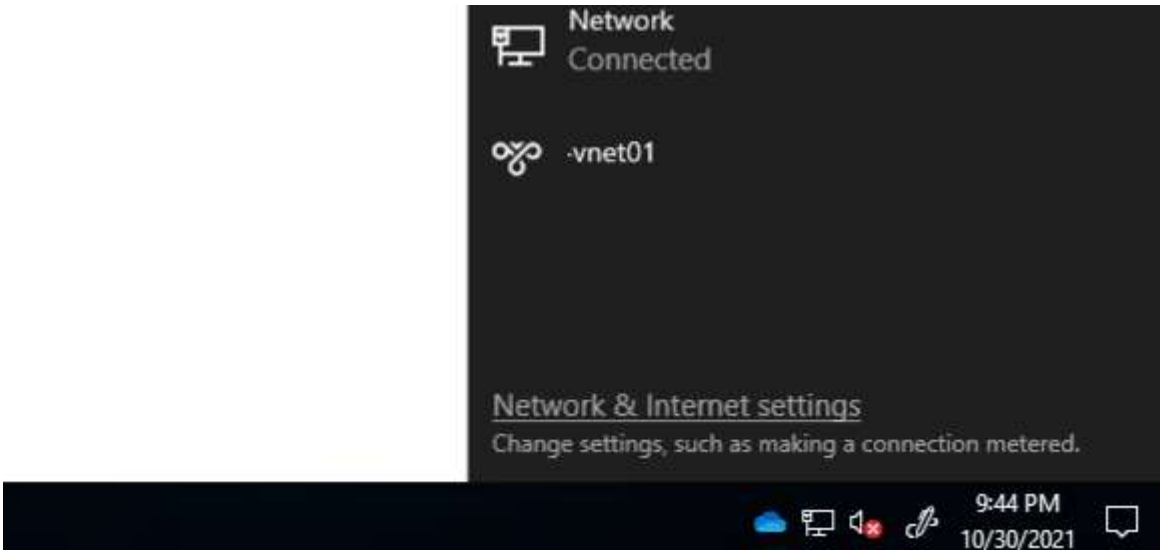
Step-5: P2S VPN Installation on Client Machine

After saving P2S on azure. One zip would download on your system. Unzip that file and you may see these three options select folder as per your OS like if you're using 32Bit OS User X86 or using 64bit use Amd64. And install exe file on your system.

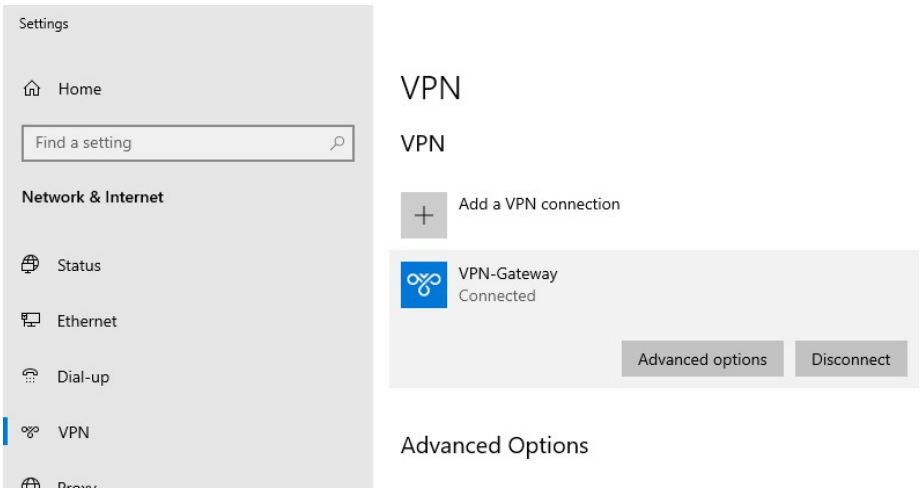
Now, remember we had exported on child file. Double click on that file install it as a local user on your system.



After installing that exe, you may see VPN option under Network Setting.



You can connect using VPN Button. And now you can connect with your Azure Infrastructure.



Now, you can connect to your Azure Network. for checking its working or not lets check we got IP form VPN Scope or not.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.2237]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sahil>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . : 40atk552wj3uhajjyfy4wr0vth.rx.internal.cloudapp.net
   Link-local IPv6 Address . . . . . : fe80::21ab:bf9:f0e8:a488%16
   IPv4 Address. . . . . . . . . . . : 10.0.0.4
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1

PPP adapter VPN-Gateway:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.0.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
   Default Gateway . . . . . . . . . :

C:\Users\sahil>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=198ms TTL=128
Reply from 192.168.0.1: bytes=32 time=196ms TTL=128
Reply from 192.168.0.1: bytes=32 time=196ms TTL=128
Reply from 192.168.0.1: bytes=32 time=265ms TTL=128

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 196ms, Maximum = 265ms, Average = 213ms

C:\Users\sahil>
```