**Configure a Point-to-Site connection to a VNet using certificate authentication**

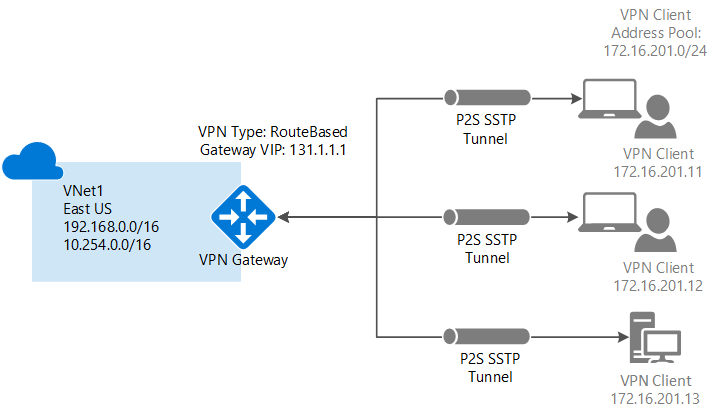
This article shows you how to create a VNet with a Point-to-Site connection in the classic deployment model using the Azure portal. This configuration uses certificates to authenticate the connecting client, either self-signed or CA issued. You can also create this configuration using a different deployment tool or deployment model by selecting a different option from the following list:



A Point-to-Site (P2S) VPN gateway lets you create a secure connection to your virtual network from an individual client computer. Point-to-Site VPN connections are useful when you want to connect to your VNet from a remote location, such when you are telecommuting from home or a conference. A P2S VPN is also a useful solution to use instead of a Site-to-Site VPN when you have only a few clients that need to connect to a VNet. A P2S VPN connection is established by starting it from the client computer.

**Important**

The classic deployment model supports Windows VPN clients only and uses the Secure Socket Tunneling Protocol (SSTP), an SSL-based VPN protocol. In order to support non-Windows VPN clients, your VNet must be created using the Resource Manager deployment model. The Resource Manager deployment model supports IKEv2 VPN in addition to SSTP. For more information, see [**About P2S connections**](https://docs.microsoft.com/en-us/azure/vpn-gateway/point-to-site-about).



Point-to-Site certificate authentication connections require the following:

* A Dynamic VPN gateway.
* The public key (.cer file) for a root certificate, which is 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.

Point-to-Site connections do not require a VPN device or an on-premises public-facing IP address. The VPN connection is created over SSTP (Secure Socket Tunneling Protocol). On the server side, we support SSTP versions 1.0, 1.1, and 1.2. The client decides which version to use. For Windows 8.1 and above, SSTP uses 1.2 by default.

For more information about Point-to-Site connections, see the [Point-to-Site FAQ](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-point-to-site-classic-azure-portal#faq) at the end of this article.

### Example settings

You can use the following values to create a test environment, or refer to these values to better understand the examples in this article:

* **Name: VNet1**
* **Address space: 192.168.0.0/16**  
  For this example, we use only one address space. You can have more than one address space for your VNet, as shown in the diagram.
* **Subnet name: FrontEnd**
* **Subnet address range: 192.168.1.0/24**
* **Subscription:** If you have more than one subscription, verify that you are using the correct one.
* **Resource Group: TestRG**
* **Location: East US**
* **Connection type: Point-to-site**
* **Client Address Space: 172.16.201.0/24**. VPN clients that connect to the VNet using this Point-to-Site connection receive an IP address from the specified pool.
* **GatewaySubnet: 192.168.200.0/24**. The Gateway subnet must use the name 'GatewaySubnet'.
* **Size:** Select the gateway SKU that you want to use.
* **Routing Type: Dynamic**

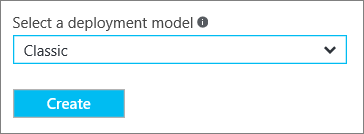
## 1. Create a virtual network and a VPN gateway

Before beginning, verify that you have an Azure subscription. If you don't already have an Azure subscription, you can activate your [MSDN subscriber benefits](https://azure.microsoft.com/pricing/member-offers/msdn-benefits-details) or sign up for a [free account](https://azure.microsoft.com/pricing/free-trial).

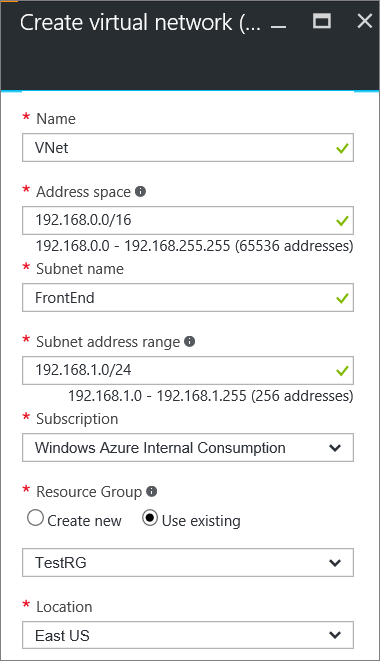
### Part 1: Create a virtual network

If you don't already have a virtual network, create one. Screenshots are provided as examples. Be sure to replace the values with your own. To create a VNet by using the Azure portal, use the following steps:

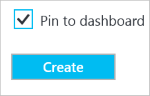
1. From a browser, navigate to the [Azure portal](http://portal.azure.com/) and, if necessary, sign in with your Azure account.
2. Click **Create a resource** > **Networking** > **Virtual Network**.
3. Near the bottom of the Virtual Network page, from the **Select a deployment model** list, select **Classic**, and then click **Create**.



1. On the **Create virtual network** page, configure the VNet settings. On this page, you add your first address space and a single subnet address range. After you finish creating the VNet, you can go back and add additional subnets and address spaces.



1. Verify that the **Subscription** is the correct one. You can change subscriptions by using the drop-down.
2. Click **Resource group** and either select an existing resource group, or create a new one by typing a name for your new resource group. If you are creating a new resource group, name the resource group according to your planned configuration values. For more information about resource groups, visit [Azure Resource Manager Overview](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview#resource-groups).
3. Next, select the **Location** settings for your VNet. The location determines where the resources that you deploy to this VNet will reside.
4. Select **Pin to dashboard** if you want to be able to find your VNet easily on the dashboard, and then click **Create**.



1. After clicking Create, a tile appears on your dashboard that will reflect the progress of your VNet. The tile changes as the VNet is being created.

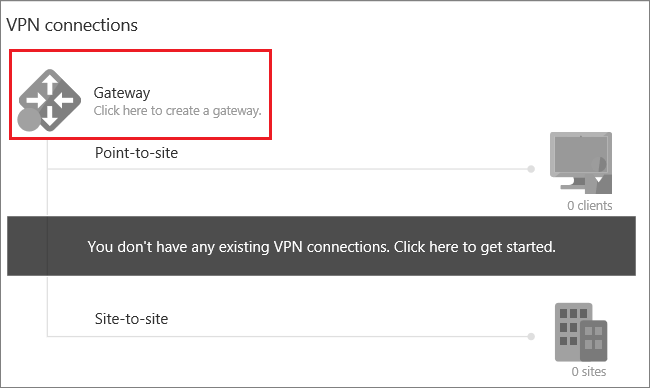


1. Once your virtual network has been created, you see **Created**.
2. Add a DNS server (optional). After you create your virtual network, you can add the IP address of a DNS server for name resolution. The DNS server IP address that you specify should be the address of a DNS server that can resolve the names for the resources in your VNet.  
   To add a DNS server, open the settings for your virtual network, click DNS servers, and add the IP address of the DNS server that you want to use.

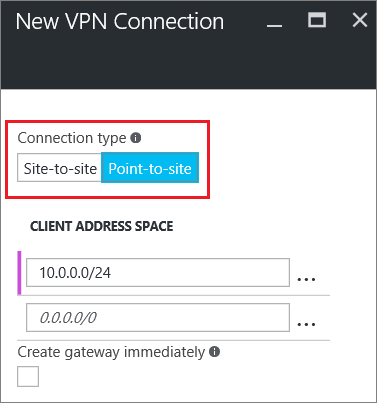
### Part 2: Create gateway subnet and a dynamic routing gateway

In this step, you create a gateway subnet and a Dynamic routing gateway. In the Azure portal for the classic deployment model, creating the gateway subnet and the gateway can be done through the same configuration pages. The gateway subnet is used for the gateway services only. Never deploy anything directly to the gateway subnet (such as VMs or other services).

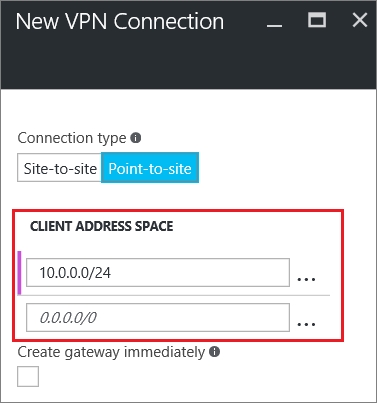
1. In the portal, navigate to the virtual network for which you want to create a gateway.
2. On the page for your virtual network, on the **Overview** page, in the VPN connections section, click **Gateway**.



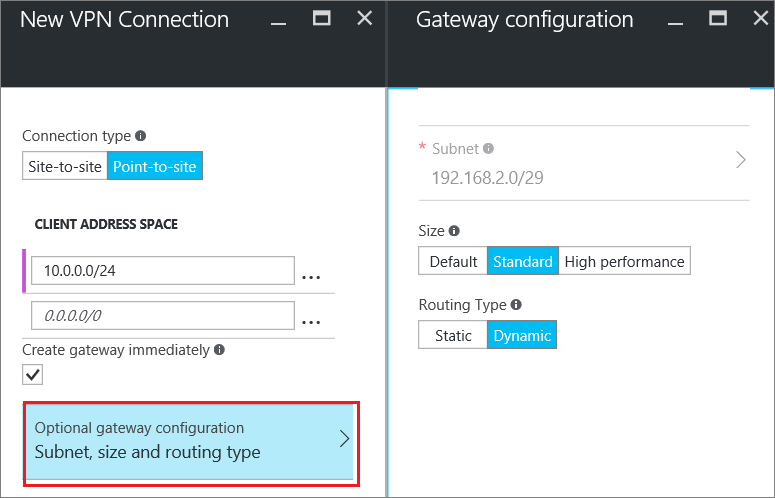
1. On the **New VPN Connection** page, select **Point-to-site**.



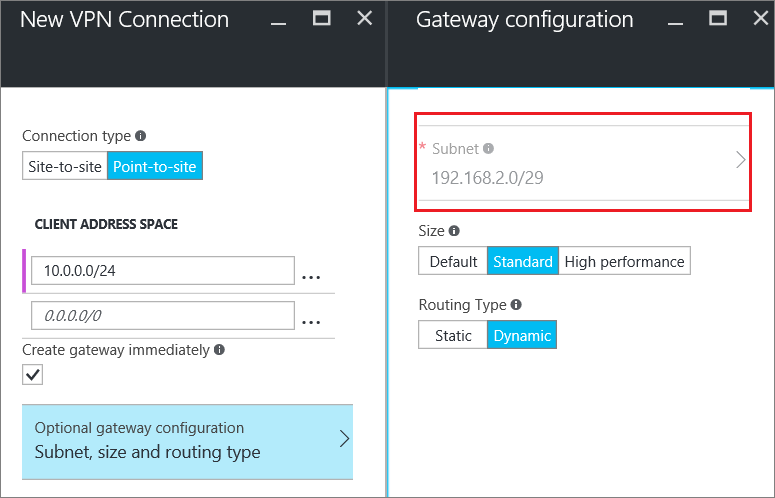
1. For **Client Address Space**, add the IP address range. This is the range from which the VPN clients receive an IP address when connecting. Use a private IP address range that does not overlap with the on-premises location that you will connect from, or with the VNet that you want to connect to. You can delete the auto-filled range, then add the private IP address range that you want to use. This example shows the auto-filled ranged. Delete it to add the value that you want.



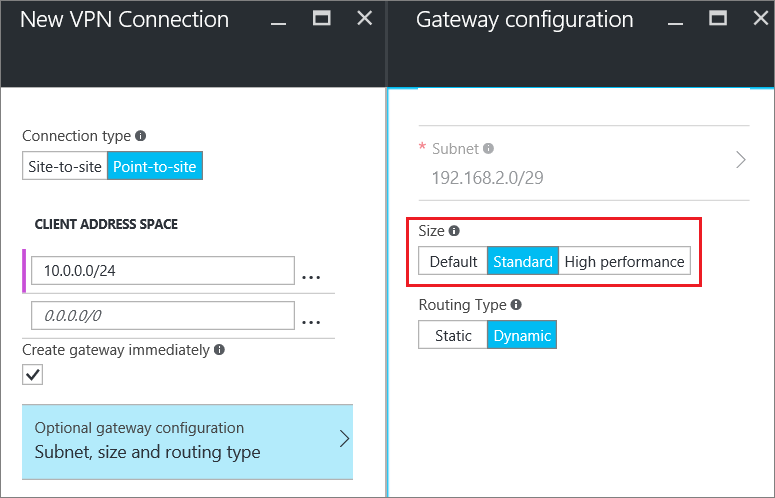
1. Select the **Create gateway immediately** checkbox. Click **Optional gateway configuration** to open the **Gateway configuration** page.



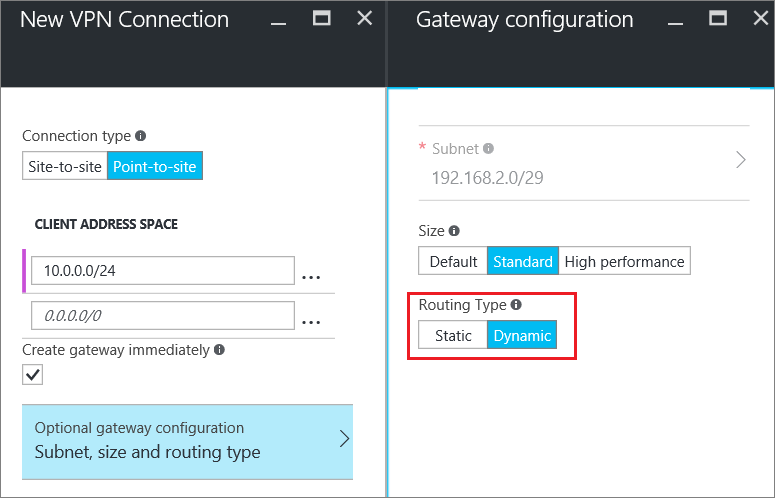
1. Click **Subnet Configure required settings** to add the **gateway subnet**. While it is possible to create a gateway subnet as small as /29, we recommend that you create a larger subnet that includes more addresses by selecting at least /28 or /27. This will allow for enough addresses to accommodate possible additional configurations that you may want in the future. When working with gateway subnets, avoid associating a network security group (NSG) to the gateway subnet. Associating a network security group to this subnet may cause your VPN gateway to stop functioning as expected.



1. Select the gateway **Size**. The size is the gateway SKU for your virtual network gateway. In the portal, the Default SKU is **Basic**. For more information about gateway SKUs, see [About VPN Gateway Settings](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-about-vpn-gateway-settings#gwsku).



1. Select the **Routing Type** for your gateway. P2S configurations require a **Dynamic** routing type. Click **OK** when you have finished configuring this page.



1. On the **New VPN Connection** page, click **OK** at the bottom of the page to begin creating your virtual network gateway. A VPN gateway can take up to 45 minutes to complete, depending on the gateway sku that you select.

## 2. Create certificates

Certificates are used by Azure to authenticate VPN clients for Point-to-Site VPNs. You upload the public key information of the root certificate to Azure. The public key is then considered 'trusted'. Client certificates must be generated from the trusted root certificate, and then installed on each client computer in the Certificates-Current User/Personal certificate store. The certificate is used to authenticate the client when it initiates a connection to the VNet.

If you use self-signed certificates, they must be created using specific parameters. You can create a self-signed certificate using the instructions for [PowerShell and Windows 10](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site), or [MakeCert](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site-makecert). It's important that you follow the steps in these instructions when working with self-signed root certificates and generating client certificates from the self-signed root certificate. Otherwise, the certificates you create will not be compatible with P2S connections and you will receive a connection error.

### Part 1: Obtain the public key (.cer) for the root certificate

You can use either a root certificate that was generated using an enterprise solution (recommended), or you can generate a self-signed certificate. After creating the root certificate, export the public certificate data (not the private key) as a Base-64 encoded X.509 .cer file and upload the public certificate data to Azure.

* **Enterprise certificate:** If you are using an enterprise solution, you can use your existing certificate chain. Obtain the .cer file for the root certificate that you want to use.
* **Self-signed root certificate:** If you aren't using an enterprise certificate solution, you need to create a self-signed root certificate. It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the certificates you create won't be compatible with P2S connections and clients receive a connection error when trying to connect. You can use Azure PowerShell, MakeCert, or OpenSSL. The steps in the provided articles generate a compatible certificate:
  + [Windows 10 PowerShell instructions](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site): These instructions require Windows 10 and PowerShell to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.
  + [MakeCert instructions](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site-makecert): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. Client certificates that are generated from the root certificate can be installed on any supported P2S client.

### Part 2: Generate a client certificate

Each client computer that connects to a VNet using Point-to-Site must have a client certificate installed. The client certificate is generated from the root certificate and installed on each client computer. If a valid client certificate is not installed and the client tries to connect to the VNet, authentication fails.

You can either generate a unique certificate for each client, or you can use the same certificate for multiple clients. The advantage to generating unique client certificates is the ability to revoke a single certificate. Otherwise, if multiple clients are using the same client certificate and you need to revoke it, you have to generate and install new certificates for all the clients that use that certificate to authenticate.

You can generate client certificates using the following methods:

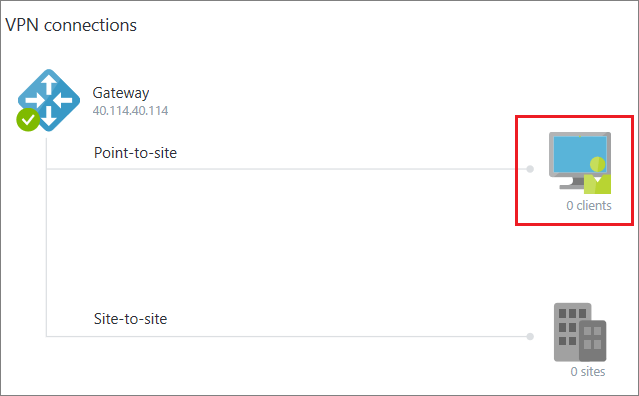
* **Enterprise certificate:**
  + If you are using an enterprise certificate solution, generate a client certificate with the common name value format 'name@yourdomain.com', rather than the 'domain name\username' format.
  + Make sure the client certificate is based on the 'User' certificate template that has 'Client Authentication' as the first item in the use list, rather than Smart Card Logon, etc. You can check the certificate by double-clicking the client certificate and viewing **Details > Enhanced Key Usage**.
* **Self-signed root certificate:** It's important that you follow the steps in one of the P2S certificate articles below. Otherwise, the client certificates you create won't be compatible with P2S connections and clients receive an error when trying to connect. The steps in either of the following articles generate a compatible client certificate:
  + [Windows 10 PowerShell instructions](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site#clientcert): These instructions require Windows 10 and PowerShell to generate certificates. The certificates that are generated can be installed on any supported P2S client.
  + [MakeCert instructions](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site-makecert): Use MakeCert if you don't have access to a Windows 10 computer to use to generate certificates. MakeCert deprecated, but you can still use MakeCert to generate certificates. The certificates that are generated can be installed on any supported P2S client.

When you generate a client certificate from a self-signed root certificate using the preceding instructions, it's automatically installed on the computer that you used to generate it. If you want to install a client certificate on another client computer, you need to export it as a .pfx, along with the entire certificate chain. This creates a .pfx file that contains the root certificate information that is required for the client to successfully authenticate. For steps to export a certificate, see [Certificates - export a client certificate](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site#clientexport).

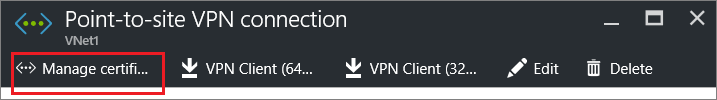
## 3. Upload the root certificate .cer file

After the gateway has been created, you can 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. You can upload additional trusted root certificate files - up to a total of 20 - later, if needed.

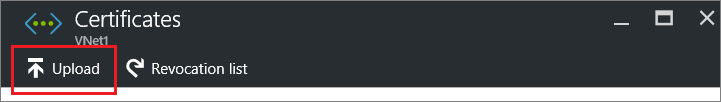
1. On the **VPN connections** section of the page for your VNet, click the **clients** graphic to open the **Point-to-site VPN connection** page.



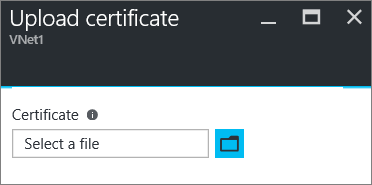
1. On the **Point-to-site connection** page, click **Manage certificates** to open the **Certificates** page.



1. On the **Certificates** page, click **Upload** to open the **Upload certificate** page.



1. Click the folder graphic to browse for the .cer file. Select the file, then click **OK**. Refresh the page to see the uploaded certificate on the **Certificates** page.



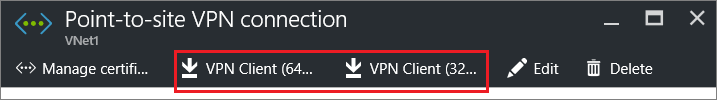
## 4. Configure the client

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. For the list of client operating systems that are supported, see the [Point-to-Site connections FAQ](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-point-to-site-classic-azure-portal#faq) at the end of this article.

### Part 1: Generate and install the VPN client configuration package

1. In the Azure portal, in the **Overview** page for your VNet, in **VPN connections**, click the client graphic to open the **Point-to-site VPN connection** page.
2. At the top of the **Point-to-site VPN connection** page, click the download package that corresponds to the client operating system on which it will be installed:
   * For 64-bit clients, select **VPN Client (64-bit)**.
   * For 32-bit clients, select **VPN Client (32-bit)**.



1. Once the packaged generates, download and install it on your client computer. If you see a SmartScreen popup, click **More info**, then **Run anyway**. You can also save the package to install on other client computers.

### Part 2: Install the client certificate

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. For more information, see [Install an exported client certificate](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site#install).

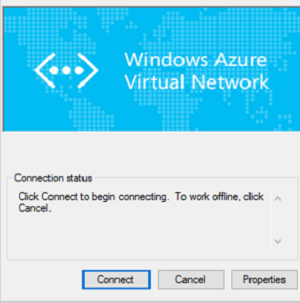
## 5. Connect to Azure

### Connect to your VNet

**Note**

You must have Administrator rights on the client computer from which you are connecting.

1. To connect to your VNet, on the client computer, navigate to VPN connections and locate the VPN connection that you created. It is named the same name as your virtual network. Click **Connect**. A pop-up message may appear that refers to using the certificate. If this happens, click **Continue** to use elevated privileges.
2. On the **Connection** status page, click **Connect** to start the connection. If you see a **Select Certificate** screen, verify that the client certificate showing is the one that you want to use to connect. If it is not, use the drop-down arrow to select the correct certificate, and then click **OK**.



1. Your connection is established.

Established connection

#### Troubleshooting P2S connections

If you are having trouble connecting, check the following items:

* If you exported a client certificate, make sure that you exported it as a .pfx file using the default value 'Include all certificates in the certification path if possible'. When you export it using this value, the root certificate information is also exported. When the certificate is installed on the client computer, the root certificate which is contained in the .pfx file is then also installed on the client computer. The client computer must have the root certificate information installed. To check, go to **Manage user certificates** and navigate to **Trusted Root Certification Authorities\Certificates**. Verify that the root certificate is listed. The root certificate must be present in order for authentication to work.
* If you are using a certificate that was issued using an Enterprise CA solution and are having trouble authenticating, check the authentication order on the client certificate. You can check the authentication list order by double-clicking the client certificate, and going to **Details > Enhanced Key Usage**. Make sure the list shows 'Client Authentication' as the first item. If not, you need to issue a client certificate based on the User template that has Client Authentication as the first item in the list.
* For additional P2S troubleshooting information, see [Troubleshoot P2S connections](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-troubleshoot-vpn-point-to-site-connection-problems).

### Verify the VPN connection

1. To verify that your VPN connection is active, from your client computer, open an elevated command prompt, and run ipconfig/all.
2. View the results. Notice that the IP address you received is one of the addresses within the Point-to-Site connectivity address range that you specified when you created your VNet. The results should be similar to this example:

Copy

PPP adapter VNet1:

Connection-specific DNS Suffix .:

Description.....................: VNet1

Physical Address................:

DHCP Enabled....................: No

Autoconfiguration Enabled.......: Yes

IPv4 Address....................: 192.168.130.2(Preferred)

Subnet Mask.....................: 255.255.255.255

Default Gateway.................:

NetBIOS over Tcpip..............: Enabled

## Connect to a virtual machine

You can connect to a VM that is deployed to your VNet by creating a Remote Desktop Connection to your VM. The best way to initially verify that you can connect to your VM is to connect by using its private IP address, rather than computer name. That way, you are testing to see if you can connect, not whether name resolution is configured properly.

1. Locate the private IP address for your VM. You can find the private IP address of a VM by either looking at the properties for the VM in the Azure portal, or by using PowerShell.
2. Verify that you are connected to your VNet using the Point-to-Site VPN connection.
3. Open Remote Desktop Connection by typing "RDP" or "Remote Desktop Connection" in the search box on the taskbar, then select Remote Desktop Connection. You can also open Remote Desktop Connection using the 'mstsc' command in PowerShell.
4. In Remote Desktop Connection, enter the private IP address of the VM. You can click "Show Options" to adjust additional settings, then connect.

### To troubleshoot an RDP connection to a VM

If you are having trouble connecting to a virtual machine over your VPN connection, there are a few things you can check. For more troubleshooting information, see [Troubleshoot Remote Desktop connections to a VM](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/troubleshoot-rdp-connection).

* Verify that your VPN connection is successful.
* Verify that you are connecting to the private IP address for the VM.
* Use 'ipconfig' to check the IPv4 address assigned to the Ethernet adapter on the computer from which you are connecting. If the IP address is within the address range of the VNet that you are connecting to, or within the address range of your VPNClientAddressPool, this is referred to as an overlapping address space. When your address space overlaps in this way, the network traffic doesn't reach Azure, it stays on the local network.
* If you can connect to the VM using the private IP address, but not the computer name, verify that you have configured DNS properly. For more information about how name resolution works for VMs, see [Name Resolution for VMs](https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-name-resolution-for-vms-and-role-instances).
* Verify that the VPN client configuration package was generated after the DNS server IP addresses were specified for the VNet. If you updated the DNS server IP addresses, generate and install a new VPN client configuration package.

## Add or remove trusted root certificates

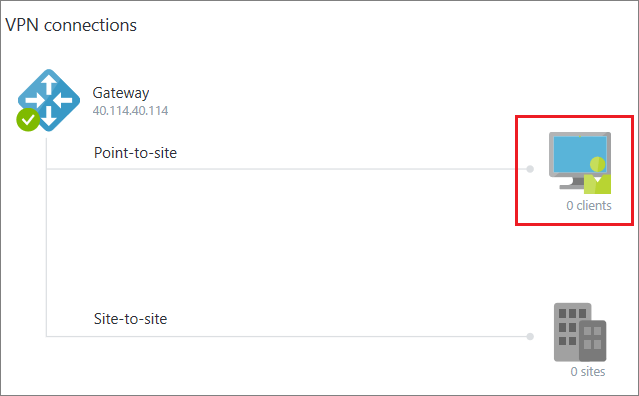
You can add and remove trusted root certificates from Azure. When you remove a root certificate, clients that have a certificate generated from that root won't be able to authenticate, and thus will not be able to connect. If you want a client to authenticate and connect, you need to install a new client certificate generated from a root certificate that is trusted (uploaded) to Azure.

### To add a trusted root certificate

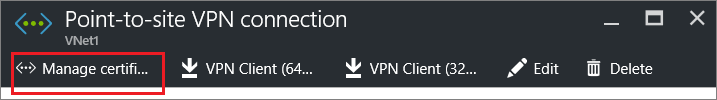
You can add up to 20 trusted root certificate .cer files to Azure. For instructions, see [Section 3 - Upload the root certificate .cer file](https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-howto-point-to-site-classic-azure-portal#upload).

### To remove a trusted root certificate

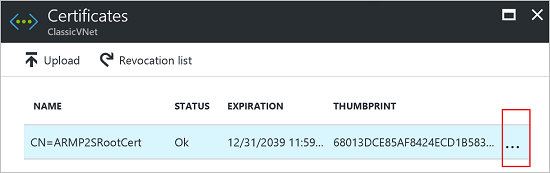
1. On the **VPN connections** section of the page for your VNet, click the **clients** graphic to open the **Point-to-site VPN connection** page.



1. On the **Point-to-site connection** page, click **Manage certificates** to open the **Certificates** page.



1. On the **Certificates** page, click the ellipsis next to the certificate that you want to remove, then click **Delete**.



## Revoke a client certificate

You can revoke client certificates. The certificate revocation list allows you to selectively deny Point-to-Site connectivity based on individual client certificates. This differs from removing a trusted root certificate. If you remove a trusted root certificate .cer from Azure, it revokes the access for all client certificates generated/signed by the revoked root certificate. Revoking a client certificate, rather than the root certificate, allows the other certificates that were generated from the root certificate to continue to be used for authentication for the Point-to-Site connection.

The common practice is to use the root certificate to manage access at team or organization levels, while using revoked client certificates for fine-grained access control on individual users.

### To revoke a client certificate

You can revoke a client certificate by adding the thumbprint to the revocation list.

1. Retrieve the client certificate thumbprint. For more information, see [How to: Retrieve the Thumbprint of a Certificate](https://msdn.microsoft.com/library/ms734695.aspx).
2. Copy the information to a text editor and remove all spaces so that it is a continuous string.
3. Navigate to the **'classic virtual network name' > Point-to-site VPN connection > Certificates** page and then click **Revocation list** to open the Revocation list page.
4. On the **Revocation list** page, click **+Add certificate** to open the **Add certificate to revocation list** page.
5. On the **Add certificate to revocation list** page, paste the certificate thumbprint as one continuous line of text, with no spaces. Click **OK** at the bottom of the page.
6. After updating has completed, the certificate can no longer be used to connect. Clients that try to connect using this certificate receive a message saying that the certificate is no longer valid.

## Point-to-Site FAQ

This FAQ applies to P2S connections using the classic deployment model.

### What client operating systems can I use with Point-to-Site?

The following client operating systems are supported:

* Windows 7 (32-bit and 64-bit)
* Windows Server 2008 R2 (64-bit only)
* Windows 8 (32-bit and 64-bit)
* Windows 8.1 (32-bit and 64-bit)
* Windows Server 2012 (64-bit only)
* Windows Server 2012 R2 (64-bit only)
* Windows 10

### Can I use any software VPN client for Point-to-Site that supports SSTP?

No. Support is limited only to the Windows operating system versions listed above.

### How many VPN client endpoints can I have in my Point-to-Site configuration?

We support up to 128 VPN clients to be able to connect to a virtual network at the same time.

### Can I use my own internal PKI root CA for Point-to-Site connectivity?

Yes. Previously, only self-signed root certificates could be used. You can still upload 20 root certificates.

### Can I traverse proxies and firewalls using Point-to-Site capability?

Yes. We use SSTP (Secure Socket Tunneling Protocol) to tunnel through firewalls. This tunnel will appear as an HTTPs connection.

### If I restart a client computer configured for Point-to-Site, will the VPN automatically reconnect?

By default, the client computer will not reestablish the VPN connection automatically.

### Does Point-to-Site support auto-reconnect and DDNS on the VPN clients?

Auto-reconnect and DDNS are currently not supported in Point-to-Site VPNs.

### Can I have Site-to-Site and Point-to-Site configurations coexist for the same virtual network?

Yes. Both these solutions will work if you have a RouteBased VPN type for your gateway. For the classic deployment model, you need a dynamic gateway. We do not support Point-to-Site for static routing VPN gateways or gateways using the -VpnType PolicyBased cmdlet.

### Can I configure a Point-to-Site client to connect to multiple virtual networks at the same time?

Yes, it is possible. But the virtual networks cannot have overlapping IP prefixes and the Point-to-Site address spaces must not overlap between the virtual networks.

### How much throughput can I expect through Site-to-Site or Point-to-Site connections?

It's difficult to maintain the exact throughput of the VPN tunnels. IPsec and SSTP are crypto-heavy VPN protocols. Throughput is also limited by the latency and bandwidth between your premises and the Internet.