1. How to set up a **Point-to-Site (P2S)** VPN in Azure using the easiest authentication method (certificate-based).
2. How to set up a **Site-to-Site (S2S)** VPN in Azure using a simulated on-prem device via Hyper-V and RRAS (Routing and Remote Access), which is the simplest supported method.

Point-to-Site (P2S) VPN Setup in Azure

A **Point-to-Site (P2S) VPN** lets individual client PCs securely connect to an Azure virtual network. We'll configure certificate-based P2S authentication using the Azure portal. This requires a route-based VPN gateway in Azure, plus a root/client certificate pair for authentication. Below are the detailed steps.

Prerequisites

- **Azure Subscription and Resources:** Ensure you have an Azure-for-Students subscription (or similar) and a Virtual Network (VNet) with a dedicated *GatewaySubnet* (e.g. /27). Create a **Virtual Network Gateway** (VPN gateway) of type **Vpn** (route-based). The Basic SKU does *not* support IKEv2 (certificate auth), so use at least a **VpnGw1** or higher.
- **Certificates:** On a Windows machine (Windows 10/11 or Windows Server 2016+), generate a *self-signed root certificate* and a *client certificate* using PowerShell. For example, run in an elevated PowerShell:

```
# Create root certificate

$root = New-SelfSignedCertificate -Type Custom -Subject "CN=P2SRootCert"
`

  -KeySpec Signature -KeyExportPolicy Exportable -KeyUsage CertSign `

  -KeyLength 2048 -HashAlgorithm sha256 -NotAfter (Get-Date).AddYears(2)
`

  -CertStoreLocation "Cert:\CurrentUser\My"



# Create client certificate signed by the root
```

```
$client = New-SelfSignedCertificate -Type Custom -Subject
"CN=P2SClientCert" `
  -DnsName P2SClient -KeySpec Signature -KeyExportPolicy Exportable `
  -KeyLength 2048 -HashAlgorithm sha256 -NotAfter (Get-Date).AddYears(2)
`
  -CertStoreLocation "Cert:\CurrentUser\My" -Signer $root `
  -TextExtension @('2.5.29.37={text}1.3.6.1.5.5.7.3.2')
```

- **Export Certificates:** Use the Certificates MMC or certmgr.msc to export the **root certificate** (CN=P2SRootCert) as a **Base-64 encoded X.509 (.CER)** file (export *without* private key). This .cer file's text (Base-64 content) will be pasted into Azure. Also export the **client certificate** with private key (as PFX) for later installation on your client PC.

Configure VPN Gateway (P2S) in Azure

1. **Open P2S Configuration:** In the Azure portal, navigate to your Virtual Network Gateway resource. In the left menu, click **Point-to-site configuration** and then **Configure now**.
2. **Address Pool:** Enter an IP address pool (IPv4) to assign VPN clients (e.g. 172.16.201.0/24). This range must not overlap with your VNet. It should be a contiguous subnet (mask ≥ 29 bits).
3. **Tunnel & Auth Type:** For **Tunnel type**, select **IKEv2** (you can also include OpenVPN(SSL) if desired). For **Authentication type**, select **Azure certificate** (certificate-based auth). This ensures the VPN gateway will trust certificates issued by your root.
4. **Upload Root Certificate:** In the *Root certificates* section, click **Add**. Give the root a name (e.g. P2SRootCert) and paste the Base64 text from your exported .cer file into the *Public certificate data* field. Be sure to copy the entire certificate (including ——BEGIN CERTIFICATE—— and footer) as a single block of text.

5. **Save Configuration:** Click **Save** at the top of the page to apply the settings. Azure will update the gateway's P2S configuration (this can take a few minutes). Once saved, the P2S gateway is configured to trust client certs issued by your root.
6. **Download VPN Client:** Still on the Point-to-site configuration page, click **Download VPN client**. This generates a client configuration package (ZIP) containing scripts and config files tailored to your P2S gateway settings. Wait for the ZIP to finish preparing, then save it.

Configure VPN Client and Connect

1. **Install Client Certificate:** On your local Windows client PC, install the client certificate (the PFX you exported). Double-click the .pfx file, follow the wizard, and include the private key in **Current User\Personal\Certificates**.
2. **Install VPN Client Software:** Ensure the **Azure VPN Client** application is installed on the client (Windows 10/11). Download it from the Microsoft Store or [Azure VPN Client download page] if needed.
3. **Import VPN Profile:** Unzip the downloaded VPN client package. Inside, locate the AzureVPN folder and find azurevpnconfig.xml (for OpenVPN) or azurevpnconfig_cert.xml. In the Azure VPN Client app, click **+ Add** > **Import** and select this XML file. This configures the VPN connection profile.
4. **Connect:** In the Azure VPN Client, select the imported profile and click **Connect**. The client will use the installed certificate to authenticate.
5. **Verify Connection:** After connecting, verify the VPN is active:
    - In Windows, run ipconfig /all and look for a new VPN adapter with an IP from the address pool.
    - In the Azure portal, go to your Virtual Network Gateway, select **Connections**, and check that the P2S connection shows status **Connected**.

- Test network access: for example, ping the private IP of an Azure VM in the VNet or RDP to it. This confirms traffic is flowing through the VPN.

Verification

To confirm the P2S VPN works end-to-end, you can:

- Check the **Connection** status in the Azure portal (virtual network gateway → Connections) shows "Connected".
- On the client PC, verify the VPN adapter's IP and DNS settings.
- Ping or access resources in the Azure VNet (e.g. ping 10.x.x.x or RDP to an Azure VM). Successful replies indicate the tunnel is up.

Sources: Microsoft documentation on P2S VPN configuration and client setup provides detailed steps for certificate-based P2S in Azure.

Site-to-Site (S2S) VPN Setup Using Hyper-V and Azure

A **Site-to-Site (S2S) VPN** connects an entire on-premises network to an Azure virtual network. In this scenario, we simulate the on-prem network by running a Windows Server VM on Hyper-V with Routing and Remote Access Service (RRAS) as the VPN "device." Below are prerequisites and steps to configure RRAS on Hyper-V, create the Azure VPN gateway, and establish an IPSec connection.

Prerequisites

- **Azure Virtual Network:** An Azure VNet (e.g. 10.1.0.0/16) with at least one subnet, *plus* a **GatewaySubnet** (for example 10.1.255.0/27).
- **Azure VPN Gateway:** A **Virtual Network Gateway** (VPN type, route-based) deployed in that VNet, with an associated Public IP address. (Creating a VPN gateway can take ~30–45 minutes.)
- **Azure Local Network Gateway:** An Azure **Local Network Gateway** object representing the on-prem network. This is configured with your on-

premises VPN device's public IP (or FQDN) and the on-premises address prefixes (e.g. 192.168.0.0/24).

- **Windows Server VM (On-Premises):** On your Hyper-V host, create a Windows Server VM (2016/2019/2022/2025) with **two network interfaces**:
    - **External NIC:** Connected to your physical (or virtual) switch/NAT with Internet access. Assign it a static IP (from your home/office router) or use DHCP if fixed.
    - **Internal NIC:** Connected to a private internal network (e.g. a Hyper-V internal switch). Assign it a static IP (e.g. 192.168.0.1/24) to represent your local LAN.

The Windows VM will act as the RRAS VPN router. It should *not* be domain-joined for simplicity.

- **Network Configuration:** Your router/firewall must forward UDP ports **500 and 4500** to the Windows RRAS server's external IP, as S2S VPN uses IKE (UDP 500) and NAT-T (UDP 4500). You also need a public IP for the Azure gateway (created with the VPN gateway).
- **Firewall Rules:** Ensure the Windows VM's firewall allows ICMP Echo (ping) and the RRAS service to receive VPN traffic. You can enable inbound ICMPv4 via PowerShell: Enable-NetFirewallRule -DisplayName "Virtual Machine Monitoring (Echo Request - ICMPv4-In)".

Azure Configuration

1. **Create Local Network Gateway:** In the Azure portal, go to **Local network gateways** and **Add** a new gateway. Enter a name (e.g. OnPrem-LocalGW), specify your Hyper-V/ISP's public IP (external IP of RRAS), and enter the on-prem address space (e.g. 192.168.0.0/24). Click **Create**.
2. **Create VPN Connection:** On your Virtual Network Gateway page, select **Connections** > **Add**. Set:
    - **Name:** any descriptive name (e.g. VNet1-to-OnPrem).
    - **Connection type:** *Site-to-site (IPSec)*.

- **Virtual network gateway:** (pre-filled with your VNet gateway).
- **Local network gateway:** select the one you created above.
- **Shared key:** Enter a strong pre-shared key (PSK). *This key must match exactly on both Azure and the RRAS server*.
  Click **OK** to create the connection. Azure will configure IPSec (this also takes a few minutes).

Configure Windows Server (RRAS) on Hyper-V VM

1. **Install RRAS Role:** Log in to the Windows VM. Open an elevated PowerShell or Server Manager and install the Remote Access and RRAS role:
2. Install-WindowsFeature -Name RemoteAccess, DirectAccess-VPN, Routing -IncludeManagementTools -Verbose

This installs the RRAS role and management tools. After installation, launch **Routing and Remote Access** (rrasmgmt.msc).

3. **Enable RRAS:** Right-click your server in the RRAS console and choose **Configure and Enable Routing and Remote Access**. In the wizard, select **"Secure connection between two private networks"** (the site-to-site template). Click Next through prompts (leave Demand-Dial = Yes and IP assignment = Automatic). Finish the wizard to start the RRAS service.
4. **Configure Demand-Dial Interface (VPN):** A Demand-Dial Interface wizard will pop up:
   - **Interface Name:** Enter a name for the connection (e.g. AzureConnection).
   - **Connection Type:** Choose **Connect using virtual private networking (VPN)**.
   - **VPN Type:** Select **IKEv2** (Azure supports IKEv2 for Windows RRAS). Click Next.

- **Destination Address:** Enter the *Azure VPN gateway's public IP address*. (You can find it on the Azure VPN gateway's Overview page.) Click Next.
- **Protocols and Security:** Keep defaults (Route IP packets on this interface). Click Next.
- **Static Routes:** Click **Add**. Enter the Azure VNet's address prefix (e.g. 10.1.0.0) and mask length (16), set Metric = 10. Click OK, then Next. This tells RRAS to route Azure-netbound traffic into the VPN interface.
- **Credentials:** Leave demand-dial credentials blank (we'll use PSK on the next screen). Click Finish.

The AzureConnection interface now appears (initially Disconnected).

5. **Edit Interface Security:** Double-click the new interface (AzureConnection). On the **Options** tab, set "Redial attempts" to 3 (for resilience).
   On the **Security** tab, select **"Use preshared key for authentication"** and enter the same PSK you used in Azure (e.g. YourStrongPSK). Click OK.
   On the **IPv4** tab of the interface properties, click **"New Static Route"**. Again enter the Azure VNet prefix (10.1.0.0/16) and mask 255.255.0.0, and check **"Use this route to initiate demand-dial connections"**. Click OK. This duplicate static route ensures RRAS initiates the VPN when needed.

Establish and Verify the VPN

1. **Initiate the Connection:** In the RRAS console under **Network Interfaces**, right-click AzureConnection and choose **Connect**. If your Azure gateway is ready, RRAS will establish IKEv2/IPSec tunnels.
2. **Verify on RRAS:** The status of AzureConnection should change to **Connected** under Interfaces. You can also right-click it and view **Status** to see packets sent/received.
3. **Verify on Azure:** In the Azure portal, under your Virtual Network Gateway → **Connections**, the new connection should report **Connected**.

4. **Test Connectivity:**
    - **On-Prem → Azure:** From the Windows VM, open a command prompt and ping a VM inside the Azure VNet (use its private IP, e.g. ping 10.1.0.4). Successful replies confirm the VPN is passing traffic.
    - **Azure → On-Prem:** From an Azure VM, ping the internal IP of your Windows VM (e.g. ping 192.168.0.1). If RRAS is routing correctly, you should see replies.

Cost-Saving Tips

- **Gateway Billing:** Note that Azure VPN Gateway is billed **hourly while provisioned**, regardless of usage. There is no "stop" command for gateways. To save cost on an Azure for Students budget, consider **deleting the gateway** when not in use and recreating it later (an admittedly time-consuming process), or use a low-cost VPN appliance VM as an alternative. Always plan your test sessions to avoid idle billing.

Sources: This guide follows Azure's S2S VPN documentation and examples, along with community references on using Windows RRAS as a VPN device. The cited materials include official Azure tutorials for S2S configuration and expert walkthroughs. Each step above corresponds to these sources for accuracy and completeness.