**Site-to-site using Hyper-V**

**How to setup Site-to-Site using Hyper-V.**

* A site-to-site VPN connects two or more distinct networks (such as your Hyper-V-hosted on-premises infrastructure and a remote cloud or branch site) securely over the internet. In this scenario, your VPN appliance will be a Windows Server VM running on Hyper-V. Routing and Remote Access Service (RRAS) on Windows Server enables the VPN functionality. The remote end can be Azure, AWS, or another on-prem environment.

**Step 1: Prepare Your Environment**

**1.1. On-Premises (Hyper-V) Prerequisites**

* Physical host running Hyper-V, with at least one public IP available (or NAT/firewall port forwarding).
* Windows Server ISO (ideally 2016, 2019, or 2022) for your VPN appliance VM.

**1.2. Azure/Remote Site Prerequisites**

* An Azure virtual network (VNet) with GatewaySubnet created.
* A static public IP resource for the VPN gateway.
* Non-overlapping network ranges between on-prem and remote site.

**Step 2: Deploy the Windows Server VPN VM on Hyper-V**

**2.1. Create VM**

* Open Hyper-V Manager → New → Virtual Machine.
* Assign at least 2 vCPUs, 4–8GB RAM, and a 60GB disk.
* Connect two network adapters:
  + Internal: Connects to your local network.
  + External: Connects to internet-facing virtual switch.

**2.2. Install Windows Server**

* Mount the ISO, boot the VM, and follow the setup wizard.
* Assign strong admin credentials.

**Step 3: Assign IP Addresses**

**3.1. Internal NIC**

* Assign a static private IP in your LAN/subnet (e.g., 10.1.0.4).

**3.2. External NIC**

* Assign your public IP (if available) or configure NAT/port forwarding so UDP 500 & 4500 and ESP (protocol 50) reach this adapter from the public internet.

**Step 4: Install and Configure RRAS**

**4.1. Add RRAS Role**

* In Windows Server, open Server Manager → Add Roles and Features.
* Install Remote Access → Routing (and DirectAccess if available).

**4.2. Configure RRAS**

* Open Routing and Remote Access tool.
* Right-click your server, choose Configure and Enable Routing and Remote Access.
* Select Custom configuration → VPN Access and LAN Routing.
* Complete the wizard and start the RRAS service.

**Step 5: Create the Demand-dial Interface (VPN Tunnel)**

* Right-click Network Interfaces under your RRAS server → New Demand-dial Interface.
* Name: e.g., "AzureS2S".
* Connect using VPN. Select IKEv2 (or L2TP/IPsec if required by peer).
* Enter the remote gateway’s public IP (Azure Gateway IP, or remote firewall IP).

**Authentication**

* Use Pre-shared Key (input one you will also use in Azure/local peer).
* Set the dial-out credentials (not always used, but can enter placeholders).

**Add Static Route**

* In wizard, add Destination network (remotely connected subnet, e.g., 10.2.0.0/16 for Azure).

**Step 6: Edit IP Routing in RRAS**

* In IPv4 → Static Routes, ensure a route exists so all traffic destined for the remote network (e.g., 10.2.0.0/16) is sent via the "AzureS2S" demand-dial interface.

**Step 7: Configure the Remote Site**

* Create a Virtual Network Gateway (Route-based, VPN SKU).
* Create a Local Network Gateway (define your on-prem address space and your RRAS VM’s public IP).
* In Azure, set up a Site-to-Site Connection:
  + Select VPN type: IKEv2/IPsec.
  + Enter the pre-shared key (must match RRAS).
  + Use RRAS VM's public IP as the "on-premises VPN device address".

**For On-Prem-to-On-Prem:**

* On the remote firewall/VPN device, configure a tunnel with matching parameters (peer IP, pre-shared key, remote/local networks, and encryption algorithms).

**Step 8: Open Firewall Ports**

On physical/virtual firewalls and any NAT between RRAS and the internet, allow or forward to the public IP of the RRAS VM:

* UDP 500 (IKE)
* UDP 4500 (IPsec NAT Traversal)
* ESP (protocol 50)

**Step 9: Test and Verify Tunnel**

* In RRAS (Network Interfaces), right-click your demand-dial interface → Connect.
* In Azure Portal (VPN Connections), confirm status shows "Connected".
* Run ping or tracert from a VM on your LAN to a VM in the remote network.

**[I was not able to perform it practically since I was not having sufficient specs to run a windows server properly.]**