POINT TO SITE

A Point-to-Site (P2S) VPN connection in Azure allows individual client computers to establish a secure connection to an Azure Virtual Network (VNet) over the internet. This is particularly useful for remote workers, telecommuters, or anyone needing to connect to Azure resources from a remote location without requiring a dedicated VPN device on their premises (as with a Site-to-Site VPN).

Step 1.1: Create an Azure Virtual Network (VNet) and Subnet :

- In the search bar, type "Virtual Network" and select it and Click + Create.
  - Resource Group: Create a new resource group (e.g., RG-OpenVPN).
  - Name: VNet-VPN
  - IPv4 address space: Define your VNet's IP address space (e.g., 10.0.0.0/16).
  - + Add subnet: Add a subnet for your resources (e.g., VM-Subnet, 10.0.1.0/24).

Step 1.2: Deploy an Azure Linux VM : This will be your OpenVPN server.

- Virtual machine name: OpenVPN-Server
- Image: Select Ubuntu Server 24.04 LTS - Gen2.
- Azure Spot instance: No
- Size: Click "See all sizes" and select B1s.
- Public inbound ports: Select Allow selected ports and Choose SSH (22).
  - We will add the OpenVPN port later via NSG.
- NIC network security group: Select Basic.

Step 1.3: Configure Network Security Group (NSG) for OpenVPN Traffic :

We only opened SSH port 22. We need to open the OpenVPN port.

1. Navigate to your VM: In the Azure portal, search for and select your newly created VM (e.g., OpenVPN-Server).
2. Networking: In the left-hand menu, under Settings, select Networking.
    - Click Add inbound port rule.
    - Source: Any, Source port ranges: *, Destination: Any
    - Destination port ranges: 1194 (This is the default OpenVPN UDP port)
    - Protocol: UDP, Action: Allow, Priority: 110
    - Name: Allow_OpenVPN_UDP

Step 1.4: SSH into the VM and Install OpenVPN Server :

1. Connect via SSH:
2. Download and Run the OpenVPN Install Script:
    - wget https://raw.githubusercontent.com/Nyr/openvpn-install/master/openvpn-install.sh
    - chmod +x openvpn-install.sh
        - This command makes the downloaded script executable.
    - sudo ./openvpn-install.sh
3. scp -i ~/Downloads/Internship/week\ 8/OpenVPN-Server_key.pem azureuser@98.70.98.105:/home/azureuser/myclient.ovpn .
4. You can now close the SSH session to your VM (type exit in the VM's terminal).

Step 1.5: Configure IP Forwarding on the OpenVPN VM :

This step enables your OpenVPN server VM to forward network traffic between your VPN clients and your Azure Virtual Network.

1. Connect to your OpenVPN VM via SSH
2. Edit the sysctl.conf file:
    - Once connected to the VM's terminal, run this command:
        - sudo nano /etc/sysctl.conf
3. Enable IP Forwarding:
    - Scroll down the file and find the line: #net.ipv4.ip_forward=1
    - Remove the # character at the beginning of this line.
4. Save and Exit nano:
    - Press Ctrl+O (Write Out), Enter to save, Press Ctrl+X to exit the editor.
5. Apply the Changes:
    - In the VM's terminal, run this command to immediately apply the IP forwarding setting: sudo sysctl -p

Step 1.6: Configure Routing on the OpenVPN Server :

This step tells the OpenVPN server to instruct connected VPN clients how to reach your Azure VNet's private IP range (10.0.1.0/24).

1. Edit the OpenVPN Server Configuration File:
    - Still connected, run: sudo nano /etc/openvpn/server/server.conf
    - Find : push "redirect-gateway def1 bypass-dhcp". Add the following line exactly as shown: push "route 10.0.1.0 255.255.255.0"
    - Press Ctrl+O (Write Out), Enter to save, Press Ctrl+X to exit the editor.
2. Restart the OpenVPN Service: In the VM's terminal, run:
    - sudo systemctl restart openvpn-server@server.service

Step 1.7: Configure Routing in Azure (Azure Portal) :

This crucial step ensures that traffic originating from your Azure VNet knows how to find its way back to your VPN clients.

1. Identify your OpenVPN VM's Private IP Address:
    - Navigate to your OpenVPN-Server VM and note down the Private IP address. For example it's 10.0.1.4.
2. Create a Route Table:
    - Search "Route tables", select it and Click + Create.
    - Fill in the Basics tab:
        - Resource Group: Your resource group (e.g., RG-OpenVPN).
        - Name: RT-OpenVPN-Clients (or a name you prefer).
3. Add a Route to the Route Table:
    - Navigate to RT-OpenVPN-Clients, select Routes and Click + Add.
        - Route name: Route_to_OpenVPN_Clients.
        - Address prefix destination: This is the IP range OpenVPN assigns to clients. The default is 10.8.0.0/24. Use this value.
        - Next hop type: Select Virtual appliance.
        - Next hop address: Enter the Private IP address of your OpenVPN VM.
4. Associate Route Table to Subnets:
    - Still on your Route Table (RT-OpenVPN-Clients), in the left menu, under Settings, select Subnets.
    - Click + Associate.
    - Virtual network: Select your VNet (e.g., VNet-VPN).
    - Subnet: Select your only subnet, which is 10.0.1.0/24.
    - Click OK.

Step 1.8: Install OpenVPN Client and Connect (for Ubuntu Client) :

This step involves installing the OpenVPN client on your local Ubuntu machine and configuring it using your downloaded .ovpn file.

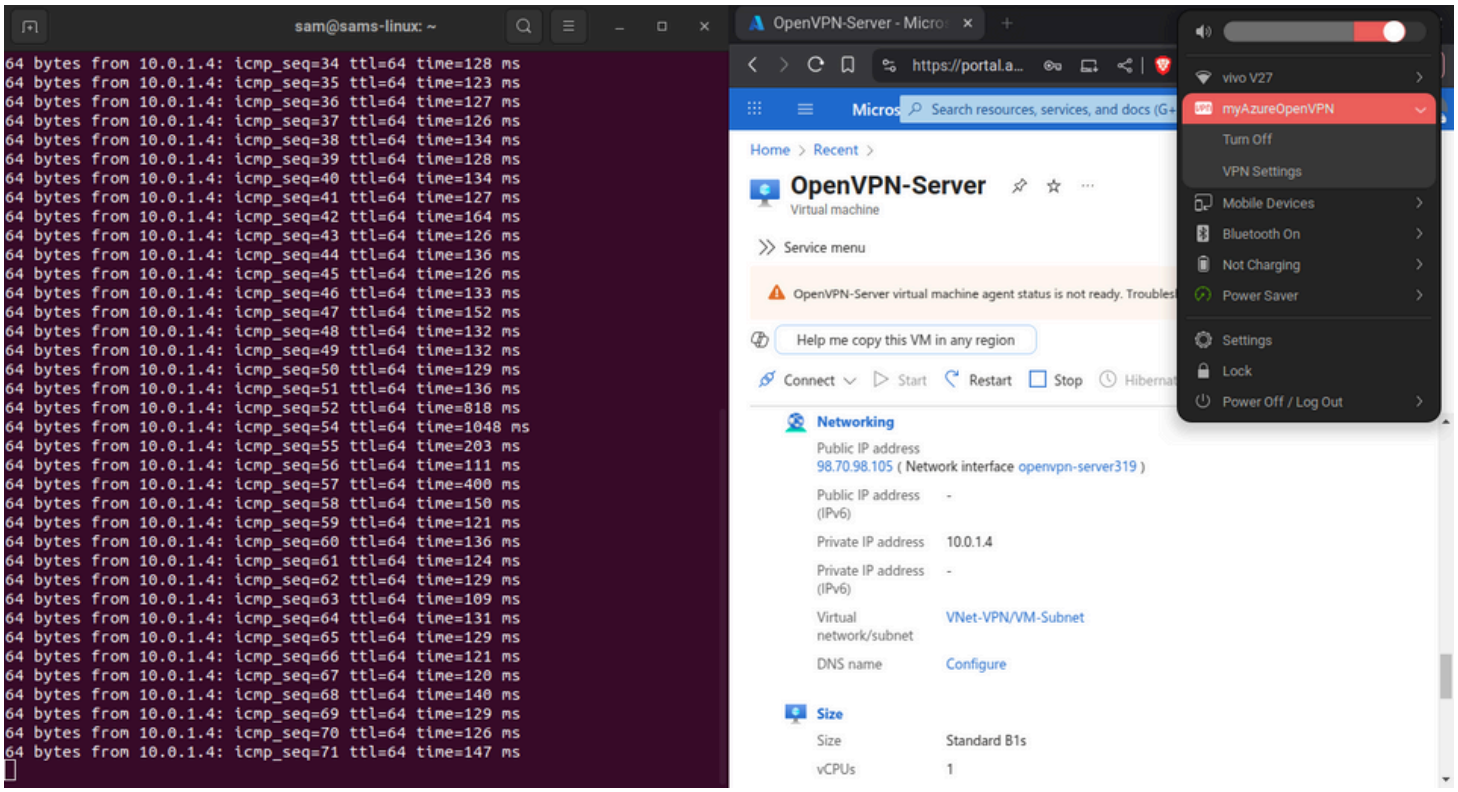
Using Network Manager:

1. Install Required Packages:
    - Open a terminal on your local Ubuntu machine, Run:
    - sudo apt install openvpn network-manager-openvpn network-manager-openvpn-gnome -y
    - Restart Network Manager: sudo systemctl restart NetworkManager
2. Import the .ovpn Profile:
    - Click the Network icon (Wi-Fi/Ethernet) in your Ubuntu system tray.
    - Click "VPN Off" or "Settings" (gear icon) to open Network settings.
    - Go to the "VPN" section, Click the + (plus) button.
    - Select "Import from file...", myclient.ovpn file, Click "Open".
    - A new window will appear with pre-filled settings.
    - You can change the "Connection name" to something like "Azure OpenVPN."
    - Connect to the VPN:
    - Go back to the Network icon in your system tray.
    - Click on your newly added VPN connection (e.g., "Azure OpenVPN").
    - Click "Connect".
    - The network icon should change, and you should see a notification confirming the VPN connection is established.

Step 1.9: Verify Connectivity :

This is the final check to confirm that your VPN tunnel is functional and you can reach resources in your Azure VNet via their private IPs.

1. Open a Terminal on your Local Ubuntu Machine:
    - Ensure your OpenVPN client shows "Connected." If you used the command-line client (sudo openvpn --config myclient.ovpn), open a new terminal window.
2. Test by Pinging your OpenVPN Server's Private IP:
    - Run the ping command to test connectivity to the private IP address of your OpenVPN server VM. This confirms basic routing within the VNet through the VPN.
    - ping 10.0.1.4 Expected Outcome: You should see ping replies.



CONCLUSION

- **Virtual Machine Preparation:** Setting up an Ubuntu VM on the Azure Free Tier and ensuring necessary network security group (NSG) rules were in place.

- **OpenVPN Server Installation and Configuration:** Utilizing an automated script to install OpenVPN, generate certificates, and configure server-side routing for seamless traffic flow. Crucially, we enabled IP forwarding on the VM and pushed routes to inform VPN clients how to reach the Azure VNet.

- **Azure Networking Integration:** We highlighted the importance of creating and associating an Azure Route Table to explicitly direct return traffic from Azure resources back to the VPN clients, ensuring full bidirectional communication.

- **Client Setup and Verification:** Guiding the installation of the OpenVPN client on an Ubuntu machine, importing the generated profile, and verifying connectivity through basic ping and ssh tests to resources within the Azure VNet.

Submitted by:

Sambit Kumar Panda

References:

https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-overview
https://openvpn.net/
https://github.com/Nyr/openvpn-install
https://learn.microsoft.com/en-us/azure/cost-management-billing/cost-management-billing-overview