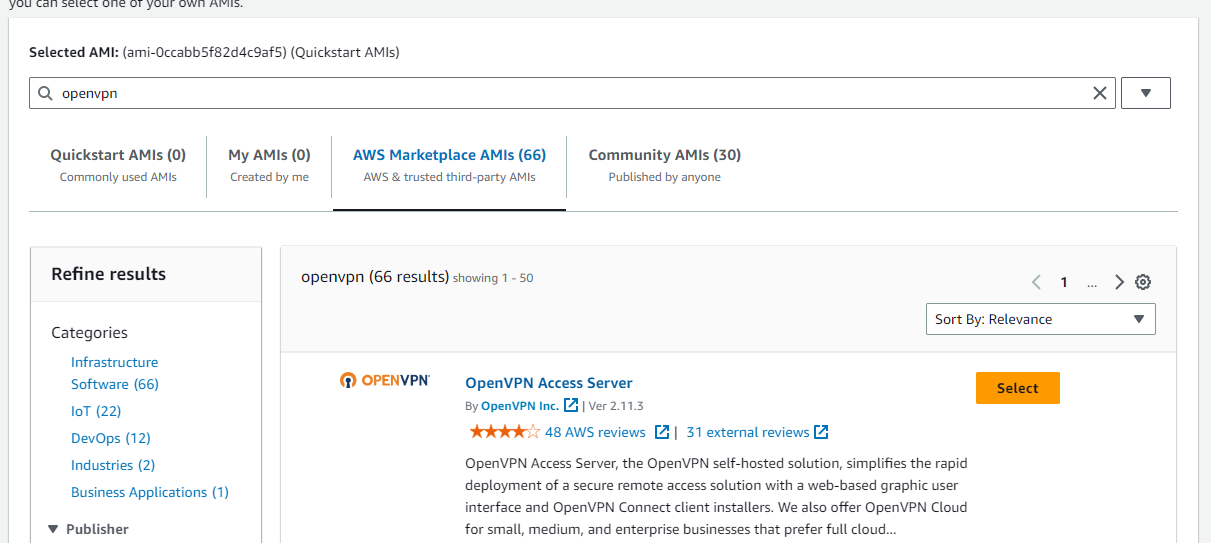
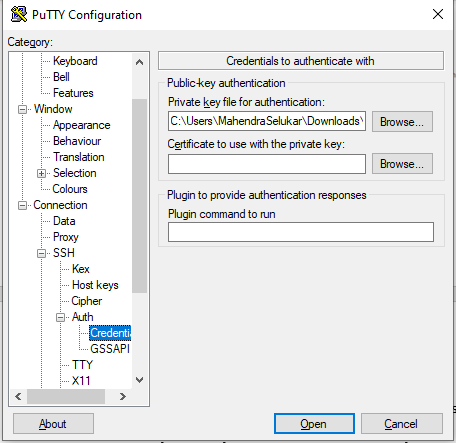
**AWS VPN Connection Using Transit Gateway**

A computer screen shot of a network

Description automatically generated

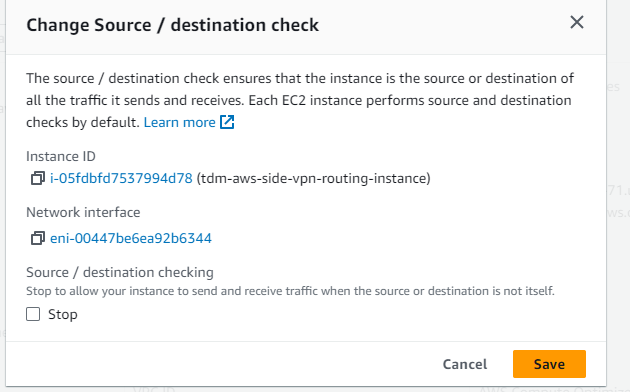
1. In AWS Create a new VPC with CIDR as 10.0.0.0/16 and s3 endpoint. This VPC will act as AWS side and will communicate with on-premises data center through transit gateway.
2. Create internet gateway and attach to aws new vpc.
3. Create gateway endpoint for s3 service.
4. Create interface endpoint for glue, databrew, vpc-lattice service.
5. Setting Up Customer Gateway Using OpenVPN Server
   * In AWS, set up EC2 instance with OpenVPN access server Ami image. 
   * Instance type as t2micro, create a new key pair and download .pem/.ppk file.Use the newly created key pair, configure Network settings with default vpc.A screenshot of a computer

     Description automatically generated
   * Create New Security group as per suggested rules and launch instance.
   * Since public ip address will be used for login, allocate new Elastic ip address for this instance. Click Elastic IP’s 🡪Allocate Elastic IP address 🡪Allocate
   * Select newly create EIP 🡪Actions 🡪Associate Elastic IP address 🡪Select instance and then associate.
   * Copy elastic ip address of openvpn server and ssh to it using .ppk /.pem file. On windows, open putty, enter elastic ip address, expand Connection 🡪SSH🡪Auth🡪Credentials🡪Browse .ppk file and click open button 
   * Login to server with root user as **openvpnas.** Configure the instance with default options. The admin username is **openvpn** and password needs to set during configuration.
   * In browser, launch openvpn access server using elastic ip address https://<elastic ip address>:943/admin/; Enter username as openvpn and password set during configuration. Make sure login is successful.
   * Expand Configuration 🡪Network Settings🡪 Interface and IP Address / Admin Web Server 🡪Yes for eth0. Save settings and update running server.
   * Click on VPN Settings 🡪Routing 🡪Enter CIDR addresses of both vpc’s(default on which openvpn server is running and new aws vpc). Save settings and update running server.A white background with black text

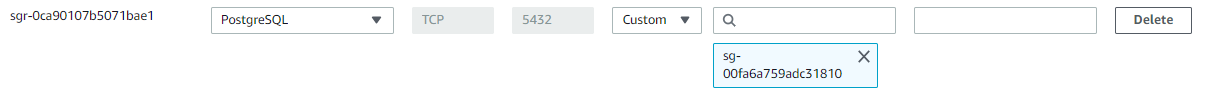
     Description automatically generated
   * Click User Management 🡪User Permission. Create a new user and click More settings. Set the password.
   * Under access control change setting then save and update running server A screenshot of a computer

     Description automatically generated
   * Go to details page of openvpn access server ec2 instance, click on security tab and check the inbound/outbound rules are as per below list A screenshot of a computer

     Description automatically generatedA screenshot of a computer

     Description automatically generated
   * Create new EC2 on t2 micro instance with new aws side vpc, security group and any subnet. Once the instance is in running state, click on Actions 🡪Networking 🡪Change source/destination check 🡪uncheck stop option 
   * Make sure to check inbound/outbound rules of security group as per below entries A screenshot of a computer

     Description automatically generatedA screenshot of a computer

     Description automatically generatedA screenshot of a computer

     Description automatically generated

A screenshot of a computer

Description automatically generated

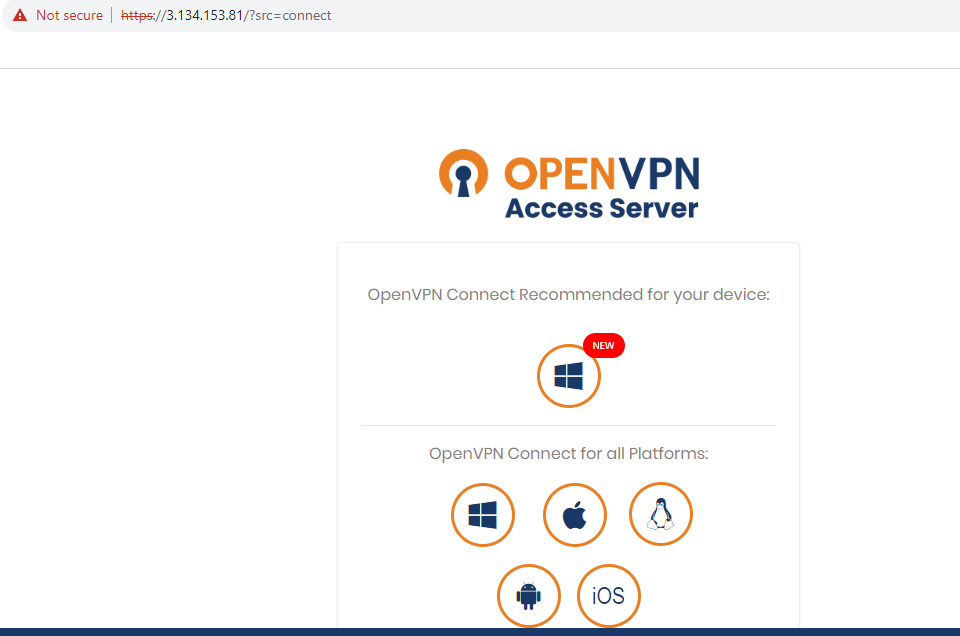
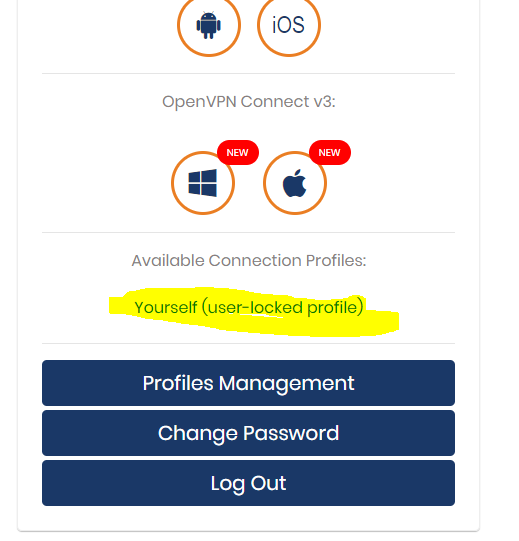
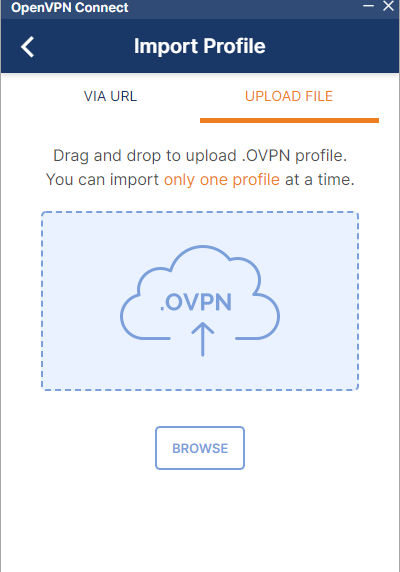
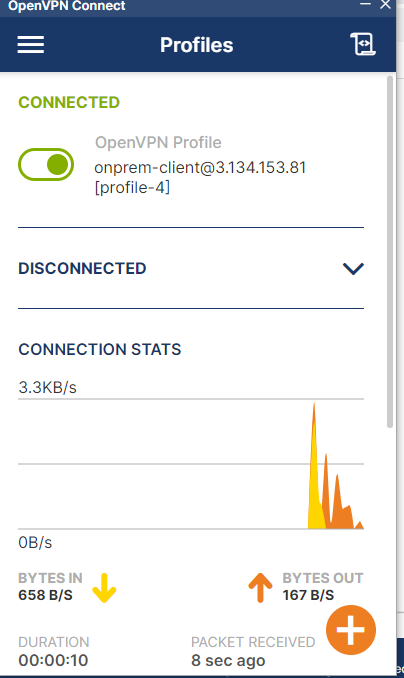
* Create a Transit gateway.
* Create transit gateway attachments for both default and new vpc. A screenshot of a computer

  Description automatically generated
* Navigate to Transit gateway route table, select the one associated with newly created attachments, click on Routes 🡪Create static route.
* Get the Dynamic IP address from Configuration 🡪VPN settings of openvpn access server admin page and use it while creating static route. For this address select attachment id of openvpn server. A screenshot of a route

  Description automatically generatedA screenshot of a computer

  Description automatically generated
* Navigate to route table of a subnet associated with openvpn access server and make sure to have routes as per below table A screenshot of a computer

  Description automatically generated
* Routes are Internet gateway for all traffic, transit gateway for aws new vpc CIDR(vpc on which aws resources will be running), Network interface for openvpn server with dynamic ip and local traffic to default vpc CIDR (172.31.0.0/16)
* Navigate to route table of a subnet associated with aws side ec2 instance and make sure to have routes as per below table A screenshot of a computer

  Description automatically generated
* The first route is s3 endpoint, then internet gateway for all traffic, local traffic to VPC CIDR, then transit gateways to both new and default vpc’s cidr.
* From onpremise data center, login to user portal of openvpn access server UI, https://< elastic ip address of openvpn>/?src=connect
* Enter credentials for the user created earlier through admin role and hit sign in button.
* Download OpenVPN client as per your system 
* From Available Connection profiles, click Yourself 
* This will download vpn certificate for the user we created in openvpn access server portal.
* Launch OpenVPN client, and import profile using certificate downloaded 
* Enter credentials and connect. Make sure vpn is successfully connected. 
* Open terminal and check the ip address, it will be in new format as assigned by openvpn access server something like **172.27.232.2**
* SSH to EC2 instance running on aws new vpc using public ip address and ping on-premise data centre ip address; **ping** **172.27.232.2**. This should return proper response.
* SSH to OpenVPN access server and again **ping** **172.27.232.2**. There must be a response.
* We can create one testing EC2 instance on same subnet on which OpenVPN access server is running to check bi-directional traffic i.e., aws to on-premise and vice-versa.
* From on premise data centre, ping to private ip addresses of both testing EC2 instance running on default vpc or vpc of OpenVPN server and aws side new vpc EC2 instance. There must be response from clients. This validates successful VPN connection between AWS and on-prem datacentre.