AZURE VPN & 3E-636L3 SITE-TO-SITE TEST NOTES

May 20, 2020

Ultra Intelligence & Communications

12410 Milestone Center Drive, Germantown, MD 20876
Tel 800-449-3384 Fax 301-515-1027
www.Ultra-3eTI.com

Table of Contents

List of Figures 3
List of Tables 3
Intended Reader 4
1. 3e-636L3 Product Overview 4
2. Azure Gateway Specific Settings for 3e-636L3 4
2.1. Site-to-Site Connection 4
2.1.1. Create Local Network Gateway 5
2.1.2. Create IPsec Policy 5
2.1.3. Create New S2S Connection 5
3. Configure 3e-636L3 to Connect to Virtual Network Gateway 5
3.1. Minimum Configuration 6
3.1.1. Black Side Network: Uplink 6
3.2. Site-to-Site VPN Configuration 7
4. Test VPN Connection 8
Appendix A: 3e-636L3 Supported Crypto Algorithms 9
Appendix B: PowerShell Script to Set up Azure VPN Gateway 11
Appendix C: Script to Remove & Clean Up Azure VPN Gateway 15

List of Figures

Figure 1 3e-636L3 Typical Use Cases 4
Figure 2 Accessing UUT 6
Figure 3 UUT Uplink IP Setting 7
Figure 4 Minimum Network Diagram 8

List of Tables

Table 1 Encryption Algorithms Supported by 636L3 9
Table 2 Integrity Algorithms Supported by 636L3 9
Table 3 Diffie Hellman Groups Supported by 636L3 10

Intended Reader

This writeup is intended for Azure lab to test IPsec Site-to-Site interoperability between Azure VPN gateway and Ultra-3eTI 3e-636L3 product.

Attached in the Appendix B is a complete Azure PowerShell script that was used in Ultra-3eTI in-house test. It can be used as a reference.

1. 3e-636L3 Product Site-to-Site VPN Overview

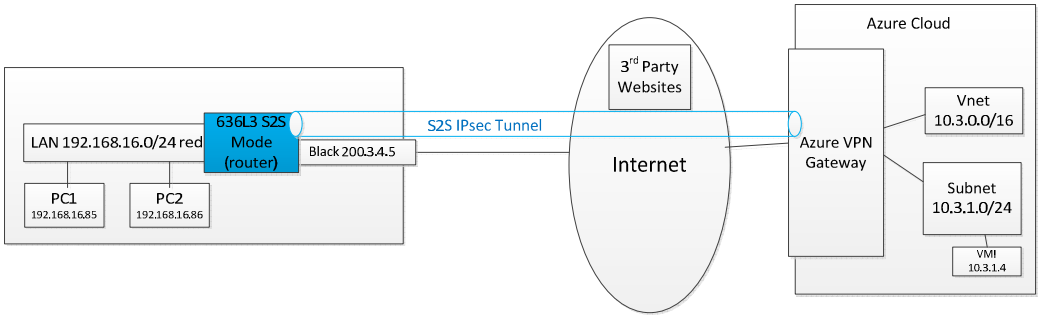


Figure 1 3e-636L3 Typical Use Cases

3e-636L3 can be used in small office as on-premise VPN router to create Site-to-Site IPsec tunnel to Azure VPN gateway where major IT infrastructure is. Traffic targeting to vnet on cloud is forwarded in IPsec tunnel. All other traffic to 3rd party websites goes as usual.

Only IKEv2, route-based VPN is supported on 3e-636L3. BGP protocol is not supported.

2. Azure Gateway Specific Settings for 3e-636L3

- Create “virtual network” with one subnet as usual
- Create “virtual machine” in subnet as usual, for access test purpose
- Create “virtual network gateway” as usual

2.1. Site-to-Site Connection

In the following PowerShell script, the specific configuration items 3e-636L3 needs are highlighted in **bold comment**.

```
$RGn          = “VPN-Resource-Group”
$LOCn         = “USGov Virginia”
$GWn          = "Gov-VPN-GW3"
```

2.1.1. Create Local Network Gateway

```
$LocalGWn   = "On-Premise-VPN1"
#Use your own IP during test
$OnPremisePublicIP = "71.163.239.155"
#3e-636L3 default LAN (red side) network is 192.168.16.0/24
$OnPremiseSubnets = "192.168.16.0/24"

$localgw = New-AzLocalNetworkGateway `
            -Name $LocalGWn   `
            -ResourceGroupName $RGn `
            -Location $LOCn `
            -GatewayIpAddress $OnPremisePublicIP `
            -AddressPrefix $OnPremiseSubnets
```

2.1.2. Create IPsec Policy

```
#Refer to Appendix A for 3e-636L3 supported crypto policy.
$ipsecPolicy = New-AzIpsecPolicy `
               -IpsecEncryption "GCMAES256" `
               -IpsecIntegrity "GCMAES256" `
               -IkeEncryption "AES256" `
               -IkeIntegrity "SHA256" `
               -DhGroup "ECP384" `
               -PfsGroup "None"
# Note: PFSgroup must be "None". Otherwise, rekey would always fail.
```

2.1.3. Create New S2S Connection

```
$S2SCONn    = "On-Premise-VPN1-connection"
#3e-636L3 requires minimum 16 characters in PSK
$PSK        = "12345678901234567890"

$vnetgw = Get-AzVirtualNetworkGateway `
               -Name $GWn `
               -ResourceGroupName $RGn

$S2SCONn   = New-AzVirtualNetworkGatewayConnection `
               -Name $LocalGWn `
               -ResourceGroupName $RGn `
               -Location $LOCn `
               -VirtualNetworkGateway1 $vnetgw `
               -LocalNetworkGateway2 $localgw `
               -ConnectionType "IPsec" `
               -SharedKey $PSK `
               -IpsecPolicies $ipsecPolicy
```

3. Configure 3e-636L3 to Connect to Virtual Network Gateway

Plug in management PC on 3e-636L3 ethernet port labeled “Local MGMT”. Configure PC network adapter to have IP address 192.168.15.2

Access 3e-636-L3 by URL https://192.168.15.1

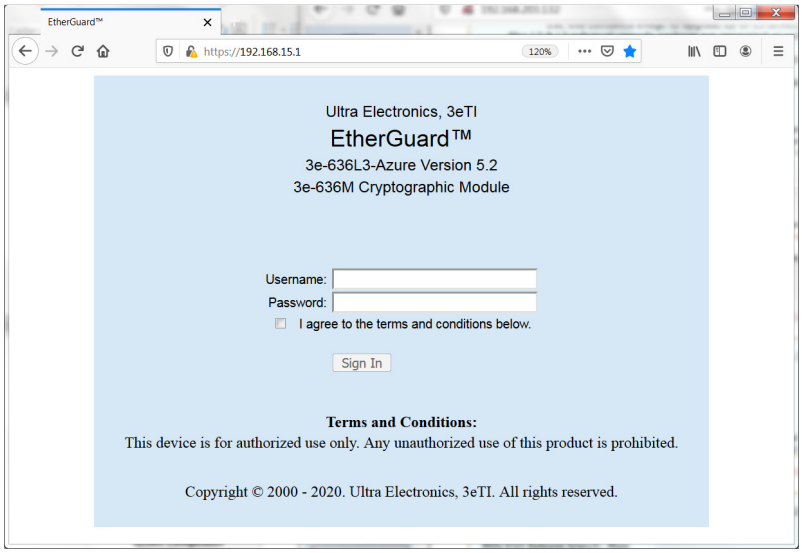


Figure 2 Accessing UUT

Device shipped for test is pre-configured with username “CryptoOfficer” and password “CryptoFIPS1” without quote sign.

Login to the device and click “*Continue*” button.

3.1. Minimum Configuration

3.1.1. Black Side Network: Uplink

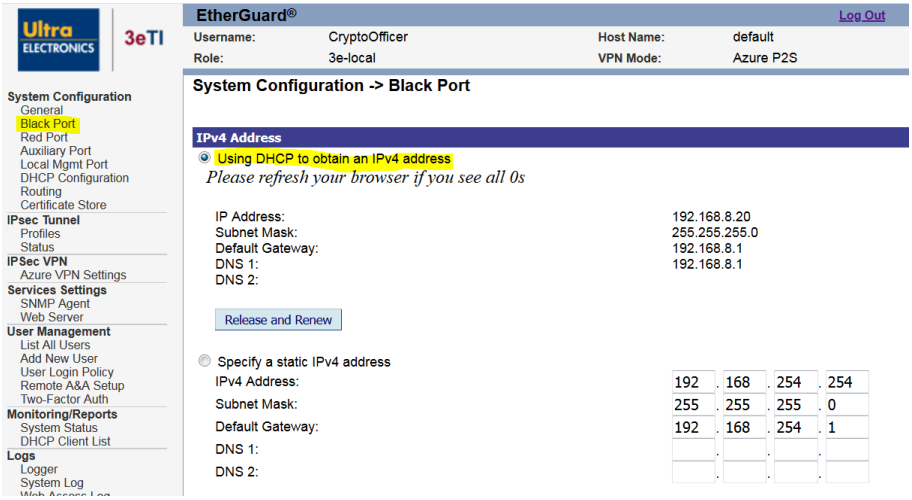
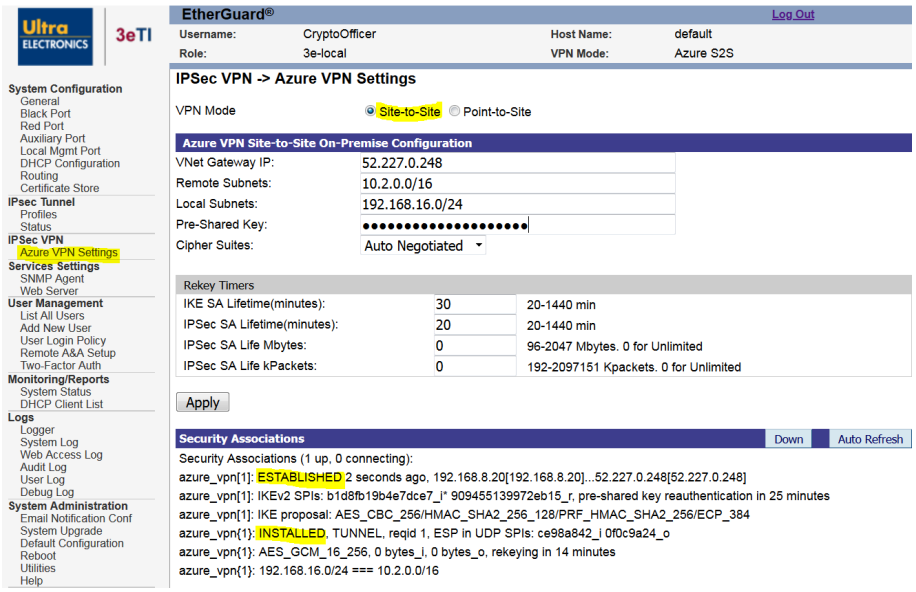


Figure 3 UUT Uplink IP Setting

The uplink port labeled “ENCRPYT/POE” is pre-configured in DHCP mode. Click “Black Port” link on left menu to verify it gets IP address successfully or you may set static IP as needed.

The downlink port labeled “UNENCRYPT/POE” is pre-configured with IP 192.168.16.1 and DHCP server to allocate IP (192.168.16.0/24) for LAN attached on this port.

3.2. Site-to-Site VPN Configuration



Click “**Azure VPN Settings**” link on left menu.

Click “**Site-to-Site**” radio option.

VNet Gateway IP: Find “virtual network gateway” IP from Azure portal or PowerShell

Remote Subnets: Fill in all subnets reachable (and want to reach) through this connection to cloud. Multiple subnets are delimited by “,”. No white space is allowed. The example only shows the VNet on Azure VPN Gateway.

Local Subnets: This “,” separated string tells Azure VPN gateway all subnets reachable through this 3e-636L3 device. No white space is allowed. The example only shows one LAN on red port labeled UNENCRYPT/POE”.

Pre-Shared Key: Value must be a character string between 16 and 64 characters. A character string may be composed of any combination of upper and lower case letters, number and the following special characters !, @, #, $, %, ^, &, *, (,).

Cipher Suite: Leave as default "***Auto negotiated***" which would use full list of supported algorithms (Appendix A) to negotiate with Azure. Note: If you use other option, make sure the IPsec Policy on Azure side matches.

The bottom section of this UI page shows the IPsec connection status. The example shows successful IPsec connection. You may click "***Auto Refresh***" button to see the statistics automatically updated every 2 seconds.

4. Test VPN Connection

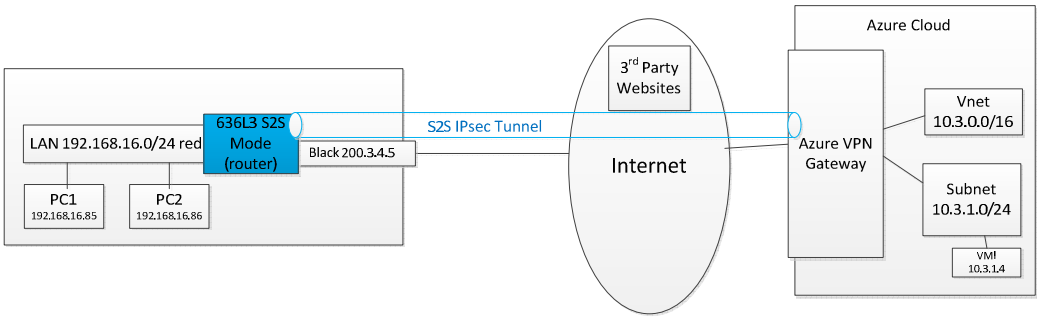


Figure 4 Minimum Network Diagram

To test VPN connection, plug a PC on red port labeled "UNENCRYPTED/POE" on 3e-636L3 device.

Ping virtual machine on virtual network on cloud.

e.g. in Figure 4,

To test S2S IPsec connectivity, ping from PC1 or PC2 to VM1 (10.3.1.4)

Appendix A: 3e-636L3 Supported Crypto Algorithms

3e-636L3 only supports the algorithms approved by FIPS/CC/DoDIN-APL.

Encryption Algorithms

Keyword	Description	IANA	AEAD	IKE	ESP
aes128	128 bit AES-CBC	12	N	Y	Y
aes256	256 bit AES-CBC				
aes256gcm64	256 bit AES-GCM with 64 bit ICV	18	Y	N	Y
aes192gcm64	192 bit AES-GCM with 64 bit ICV				
aes128gcm64	128 bit AES-GCM with 64 bit ICV				
aes256gcm96	256 bit AES-GCM with 96 bit ICV	19			
aes192gcm96	192 bit AES-GCM with 96 bit ICV				
aes128gcm96	128 bit AES-GCM with 96 bit ICV				
aes256gcm128	256 bit AES-GCM with 128 bit ICV	20			
aes192gcm128	192 bit AES-GCM with 128 bit ICV				
aes128gcm128	128 bit AES-GCM with 128 bit ICV				

Table 1 Encryption Algorithms Supported by 636L3

Integrity Algorithms

Keyword	Description	IANA	IKE	ESP
sha1	HMAC SHA1	2	Y	Y
sha256	HMAC SHA2_256_128	12		
sha384	HMAC SHA2_384_192	13		
sha512	HMAC SHA2_512_256	14		

Table 2 Integrity Algorithms Supported by 636L3

Diffie Hellman Groups

DH Groups used in IKE only, not in ESP (ie. PFS group must be none)

Keyword	DH Group	Modulus	Note
mod2048	14	2048 bits	regular group
ecp384	20	384 bits	NIST elliptic curve group
ecp521	21	521 bits	

Table 3 Diffie Hellman Groups Supported by 636L3

3e-636L3 List

```
  ike=aes256-aes128-sha512-sha384-sha256-sha1-ecp384-ecp256-modp2048!
  esp=aes256gcm128-aes256gcm96-aes256gcm64-aes256-aes192gcm128-aes192gcm96-
aes192gcm64-aes128gcm128-aes128gcm96-aes128gcm64-aes128-sha512-sha384-sha256-
sha1!
```

Appendix B: PowerShell Script to Set up Azure VPN Gateway

#define variable names. Change values for variables $XXX to fit your deployment need
#postfix "n" means this variable is name string (vs. object)

```
$RGn         = "VPN-Resource-Group"
$TAGn        = "VPN"
$LOCn        = "USGov Virginia"
$VNETn       = "vnet-10-3-0-0-16"
$VNetPrefix = "10.3.0.0/16"
$FESUBNETn   = "FrontEnd"
$FEPrefix    = "10.3.0.0/24"
$BESUBNETn   = "Backend"
$BEPrefix    = "10.3.1.0/24"
$VMhostname = "PC-in-Vnet3"
$VMintfname = "vm3intf"
$DFTuser     = "chaoxing"
```

#Note: Virtual Network Gateway can only be created in subnet with name 'GatewaySubnet'.

```
$GWSUBNETn   = "GatewaySubnet"
$GwPrefix    = "10.3.255.0/27"
$GWn         = "Gov-VPN-GW3"
$GWIPn       = "Gov-VPN-GW3-IP"
$GWIPCONFn   = "GW3ipconf"
$LocalGWn    = "On-Premise-VPN1"
$OnPremisePublicIP = "71.163.239.155"
```

#Note: PowerShell only allows one subnet per connection if it's enclosed by "".

```
$OnPremiseSubnets = "192.168.2.0/24"
$S2SCONn     = "On-Premise-VPN1-connection"
$PSK         = "12345678901234567890"

#ssh rsa key pair here is from X:\System_Test\LabSetup\azure-vm-sshkeys\
$sshPublicKey = "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDFuyJzL39nY7OHHAEXeAdBfVDsQvFO41g2cEfMSGA6zTRka
WncD0JIhOhw3sm4OxCqlSu9jfXRryWFoLH7LsN1NCd0k0Cu7nEot+1HXH3Kw4msczu2ePigd7550I
YQb4ErzyLN/oIoH/tCtca+iCXZlCfhRUrDpC44xWAD5niknsbAOqINgS6D3yeKSMS03v0s/BKkUG9
5wYd/lC29NrHcRn0hSh5mNga55cGSh4sxqAZwW+SAtr7u7si4CQW1wybVNEgb5A5MMoFUW0gMdHnG
gj8eMv6EEXf21fY/kf0kgzpkOGgapQj5aYvjaxSlErbRfTEObQkc5yL/cTxSxGe9
chaoxing.lin@ultra-3eti.com"
```

#Create Virtual Network

```
$fesub = New-AzVirtualNetworkSubnetConfig `
            -Name $FESUBNETn `
            -AddressPrefix $FEPrefix
$besub = New-AzVirtualNetworkSubnetConfig `
            -Name $BESubnetn `
            -AddressPrefix $BEPrefix
$gwsub = New-AzVirtualNetworkSubnetConfig `
            -Name $GWSUBNETn `
            -AddressPrefix $GwPrefix
$vnet    = New-AzVirtualNetwork `
            -Name $VNETn `
            -ResourceGroupName $RGn `
```

```
            -Location $LOCn `
            -AddressPrefix $VNetPrefix `
            -Subnet $fesub,$besub,$gwsub
```

#Create a virtual machine in BACKEND subnet, to test accessing PC on cloud

```
# Define a credential object
$securePassword = ConvertTo-SecureString ' ' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential ($DFTuser,
$securePassword)

$vnet = get-AzVirtualNetwork `
        -ResourceGroupName $RGn `
        -Name $VNETn

$subnetconfig = Get-AzVirtualNetworkSubnetConfig `
                -Name $BESUBNETn `
                -VirtualNetwork $vnet

$nic = New-AzNetworkInterface `
        -Name $VMintfname `
        -ResourceGroupName $RGn `
        -Location $LOCn `
        -Subnet $subnetconfig

# Create a virtual machine configuration
$vmConfig = New-AzVMConfig `
  -VMName $VMhostname `
  -VMSize "Standard_B1s" | `
Set-AzVMOperatingSystem `
  -Linux `
  -ComputerName $VMhostname `
  -Credential $cred `
  -DisablePasswordAuthentication | `
Set-AzVMSourceImage `
  -PublisherName "Canonical" `
  -Offer "UbuntuServer" `
  -Skus "18.04-LTS" `
  -Version "latest" | `
Add-AzVMNetworkInterface `
  -Id $nic.Id


# Configure the SSH key
Add-AzVMSshPublicKey `
  -VM $vmconfig `
  -KeyData $sshPublicKey `
  -Path "/home/$DFTuser/.ssh/authorized_keys"

New-AzVM `
  -ResourceGroupName $RGn `
  -Location $LOCn -VM $vmConfig
```

#Request a public IP, for VPN gateway

```
$gwpip     = New-AzPublicIpAddress `
             -Name $GWIPn    `
             -ResourceGroupName $RGn `
             -Location $LOCn `
             -AllocationMethod Dynamic

$subnet    = Get-AzVirtualNetworkSubnetConfig `
             -Name $GWSUBNETn `
             -VirtualNetwork $vnet

$gwipconf = New-AzVirtualNetworkGatewayIpConfig `
             -Name $GWIPCONFn `
             -Subnet $subnet `
             -PublicIpAddress $gwpip
```

#Verify: get public ip instance

```
$gwpip     = Get-AzPublicIpAddress `
             -Name $GWIPn    `
             -ResourceGroupName $RGn
```

#Create Virtual Network Gateway with point-to-site VPN. This can take up to 45 minutes! Just WAIT

```
$Gateway = New-AzVirtualNetworkGateway `
             -Name $GWn    `
             -ResourceGroupName $RGn `
             -Location $LOCn `
             -IpConfigurations $gwipconf `
             -GatewayType Vpn `
             -VpnType RouteBased `
             -GatewaySku VpnGw1

#Powershell session may expire, re-assign variables and go on
$RGn         = "VPN-Resource-Group"
$TAGn        = "VPN"
$LOCn        = "USGov Virginia"
$VNETn       = "vnet-10-3-0-0-16"
$VNetPrefix = "10.3.0.0/16"
$FESUBNETn  = "FrontEnd"
$FEPrefix    = "10.3.0.0/24"
$BESUBNETn  = "Backend"
$BEPrefix    = "10.3.1.0/24"
$VMhostname = "PC-in-Vnet3"
$VMintfname = "vm3intf"
$DFTuser     = "chaoxing"
```

#Note: Virtual Network Gateway can only be created in subnet with name 'GatewaySubnet'.

```
$GWSUBNETn  = "GatewaySubnet"
$GwPrefix    = "10.3.255.0/27"
$GWn         = "Gov-VPN-GW3"
$GWIPn       = "Gov-VPN-GW3-IP"
$GWIPCONFn  = "GW3ipconf"
$LocalGWn    = "On-Premise-VPN1"
$OnPremisePublicIP = "71.163.239.155"
```

#Note: PowerShell only allows one subnet per connection if it's enclosed by "".

```
$OnPremiseSubnets = "192.168.2.0/24"
```

```
$S2SCONn     = "On-Premise-VPN1-connection"
$PSK         = "12345678901234567890"

$Gateway = Get-AzVirtualNetworkGateway `
            -ResourceGroupName $RGn `
            -Name $GWn
```

#Create Site-to-Site VPN

```
$ipsecPolicy = New-AzIpsecPolicy `
               -IpsecEncryption GCMAES256 `
               -IpsecIntegrity GCMAES256 `
               -IkeEncryption AES256 `
               -IkeIntegrity SHA256 `
               -DhGroup "ECP384" `
               -PfsGroup "None"

$localgw = New-AzLocalNetworkGateway `
               -Name $LocalGWn   `
               -ResourceGroupName $RGn `
               -Location $LOCn `
               -GatewayIpAddress $OnPremisePublicIP `
               -AddressPrefix $OnPremiseSubnets

$connection = New-AzVirtualNetworkGatewayConnection `
               -Name $S2SCONn    `
               -ResourceGroupName $RGn `
               -Location $LOCn `
               -VirtualNetworkGateway1 $Gateway `
               -LocalNetworkGateway2 $localgw `
               -ConnectionType "IPsec" `
               -SharedKey $PSK `
               -IpsecPolicies $ipsecPolicy
```

Appendix C: Script to Remove & Clean Up Azure VPN Gateway

#This script is to delete all persisted resources created by script in **Appendix B**. This is so that we can run script in **Appendix B** for demo.

#define variable names
#postfix "n" means this variable is name string (vs. object)

```
$RGn = "VPN-Resource-Group"
$LOCn = "USGov Virginia"
$VNETn = "vnet-10-3-0-0-16"
$GWn       = "Gov-VPN-GW3"
$GWIPn     = "Gov-VPN-GW3-IP"
$LocalGWn  = "On-Premise-VPN1"
$S2SCONn   = "On-Premise-VPN1-connection"
$VMhostname = "PC-in-Vnet3"
$VMintfname = "vm3intf"

Remove-AzVirtualNetworkGatewayConnection `
               -Name $S2SCONn   `
               -ResourceGroupName $RGn `
               -Force
Remove-AzLocalNetworkGateway `
               -Name $LocalGWn   `
               -ResourceGroupName $RGn `
               -Force
#Like create AzVgw, the remove- would take a long time, too.
Remove-AzVirtualNetworkGateway `
            -Name $GWn   `
            -ResourceGroupName $RGn `
            -Force
Remove-AzPublicIpAddress `
            -Name $GWIPn   `
            -ResourceGroupName $RGn `
            -Force
Remove-AzVM `
        -ResourceGroupName $RGn `
        -Name $VMhostname `
        -Force
Remove-AzNetworkInterface `
        -ResourceGroupName $RGn `
        -Name $VMintfname `
        -Force
Remove-AzVirtualNetwork `
        -ResourceGroupName $RGn `
        -Name $VNETn `
        -Force
```