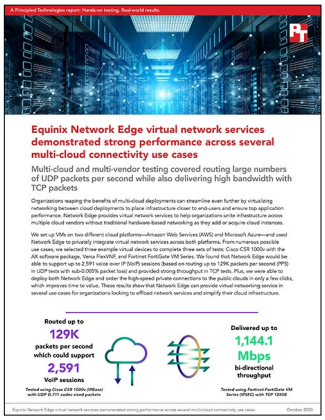

The science behind the report:

Equinix Network Edge virtual network services demonstrated strong performance across several multi-cloud connectivity use cases

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report Equinix Network Edge virtual network services demonstrated strong performance across several multi-cloud connectivity use cases.

We concluded our hands-on testing on August 14, 2020. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on June 26, 2020 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

Table 4: Complete network performance statistics we gathered using various codecs and security protocols while testing Network Edge with the Cisco CSR 1000v virtual router.

	Byte size and protocol (bi-directional)	Bandwidth (Mbps)	PPS	Average latency (ms)	Jitter (ms)	Packet loss (%)
Cisco CSR 1000v (IPBase)	UDP G.711 218B	226.0	129,990	2.7160	0.1190	0.004050%
	TCP 1350B	970.9				
	TCP IMIX	930.4				
Cisco CSR 1000v (IPSEC)	UDP G.711 218B	129.0	73,994	5.0980	0.2935	0.003355%
	TCP 1350B	931.0				
	TCP IMIX	902.6				

Table 5: Complete network performance statistics we gathered using various codecs and security protocols while testing Network Edge with the Versa FlexVNF SD-WAN device.

	Byte size and protocol (bi-directional)	Bandwidth (Mbps)	PPS	Average latency (ms)	Jitter (ms)	Packet loss (%)
Versa FlexVNF (IPBase)	UDP G.711 218B	83.8	47,997	2.3540	0.0150	0.000600%
	TCP 1350B	1,941.3				
	TCP IMIX	1,953.6				
Versa FlexVNF (IPSEC)	UDP G.711 218B	59.2	33,998	4.2030	0.1350	0.000875%
	TCP 1350B	1,104.5				
	TCP IMIX	974.4				

Table 6: Complete network performance statistics we gathered using various codecs and security protocols while testing Network Edge with the Fortinet FortiGate VM Series device.

	Byte size and protocol (bi-directional)	Bandwidth (Mbps)	PPS	Average latency (ms)	Jitter (ms)	Packet loss (%)
Fortinet FortiGate VM Series (IPBase)	UDP G.711 218B	62.8	35,998	2.0540	0.1475	0.000260%
	TCP 1350B	1,921.0				
	TCP IMIX	1,940.5				
Fortinet FortiGate VM Series (IPSEC)	UDP G.711 218B	87.2	49,994	5.8370	0.1870	0.002000%
	TCP 1350B	1,144.1				
	TCP IMIX	964.0				

System configuration information

Table 7: Detailed information for the VMs we tested.

Cloud	Instance	Processor	# of cores	Memory (GB)
Amazon Web Services	t3.xlarge	Intel® Xeon® Platinum 8175M @2.5GHz	4	16
Azure	D4as_v4	AMD EPYC 7452 @ 2.35GHz	4	16

IPSEC over Direct Connect network topology

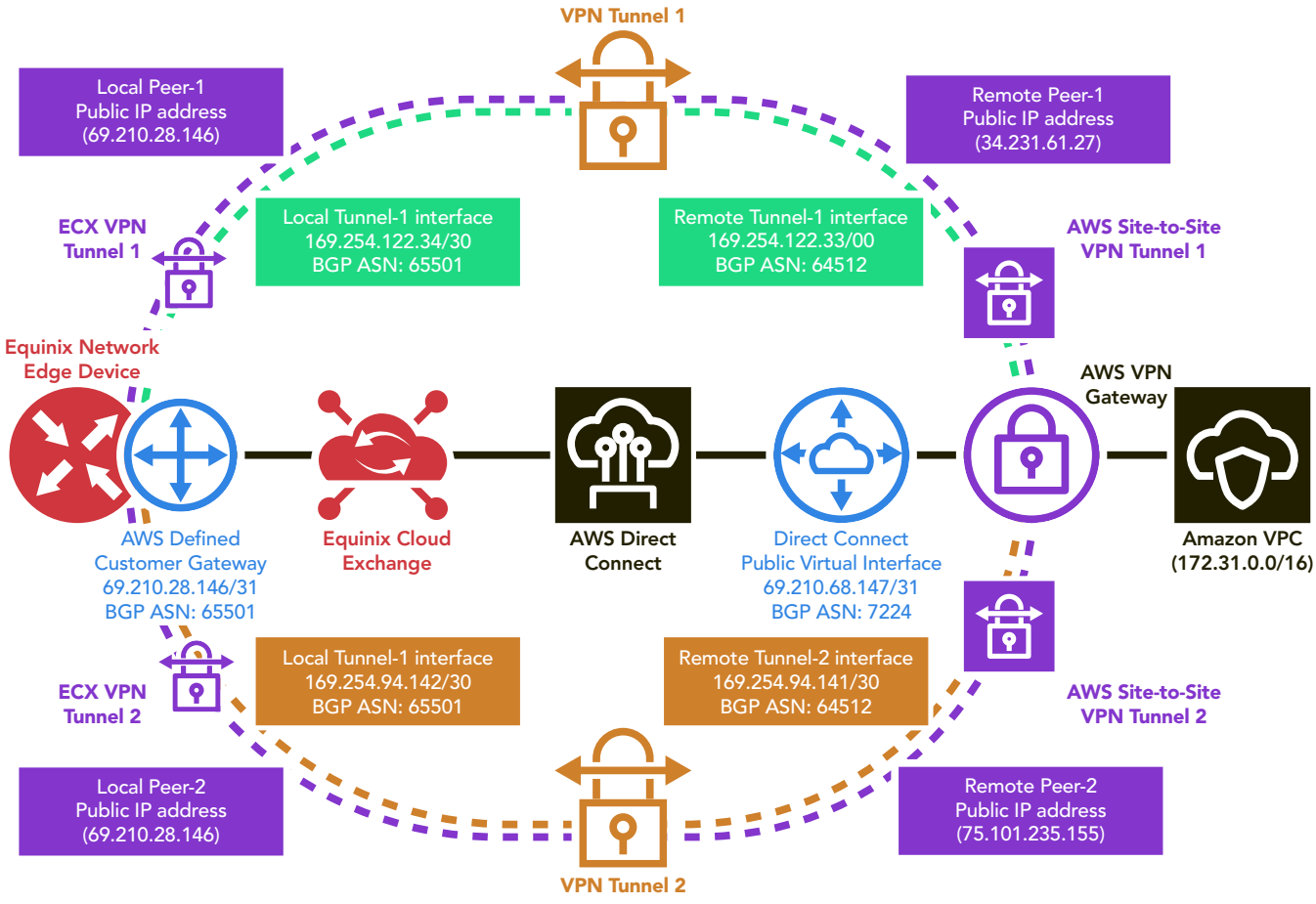

Figure 1: A diagram of the virtual and physical paths we set up for testing.

How we tested

Microsoft Azure Setup

Creating ExpressRoute circuit

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Networking and select ExpressRoute circuits
4. Click Add.
5. Select the Subscription and Resource Group associated with the project.
6. Under Instance Details, select the Region and provide a name for the ExpressRoute circuit, then click Next: Configuration.
7. Configure the following settings:

   - Port Type = Provider
   - Create new or import from classic = Create new
   - Provider = Equinix
   - Peering location = Washington DC
   - Bandwidth = 1Gbps
   - SKU = Standard
   - Billing model = Metered
   - Allow Classic operations = No

8. Click Next: Tags.
9. Add appropriate project tags, then click Next: Review + create.
10. If validation is successful, click Create.
11. Once complete, make note of the Service Key.

Creating connection in ECX Fabric

1. Log into ECX Fabric.
2. Click Connections, then click Create Connection.
3. Under Frequent Connection, under the Microsoft Azure icon, click Select.
4. In the Azure Express Route area, select Create Connection.
5. Click Create a Connection to Microsoft ExpressRoute.
6. Under Primary Origin, select Ashburn, then select your switch.
7. Under Destination, select Ashburn, then click Next.
8. Add a unique name for the Primary Connection Information. (Example: PT-Azure-1)
9. Add a unique name for the Secondary Connection Information. (Example: PT-Azure-2)
10. Add the Service Key for the ExpressRoute circuit.
11. Under Application Details, select Private.
12. Leave the Purchase Order Number blank.
13. Click Next.
14. Verify setup, add notification emails, and click Submit your Order.
15. Click Connections, and click View Connections.
16. Select either of the Azure connections.
17. Under the Primary BFP Information, use the following information.

    - Local ASN = 65501
    - Local IP Address = 172.16.254.1/30
    - Remote ASN = 12076
    - Remote IP address = 172.16.254.2
    - BGP Authentication Key = Leave blank

18. Under the Secondary BFP Information, use the following information:

    - Local ASN = 65501
    - Local IP Address = 172.16.254.5/30
    - Remote ASN = 12076
    - Remote IP address = 172.16.254.6
    - BGP Authentication Key = Leave blank

19. Click Sync BGP Peering.

Creating ExpressRoute circuit Peering

1. In Microsoft Azure, navigate to the ExpressRoute circuit that you just created.
2. Click Peerings, then click Azure Private.
3. In the Private peering setup, add the following:

    - Peer ASN = 65501
    - Primary Subnet = 172.16.254.0/30
    - Secondary Subnet = 172.16.254.4/30
    - VLAN ID = 1000
    - Shared key = Leave blank

4. Click Save.

Verifying provisioning

1. Log into ECX Fabric.
2. Click Connections, and click View Connections.
3. Click either of the Azure connections.
4. Verify the Status and Provider Status show Provisioned for both Primary and Secondary connections (this may take several minutes).

Creating Network Security Groups

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Networking, and select Network Security Groups.
4. Click Add to add a new group.
5. Select the Subscription and Resource Group associated with the project.
6. Name the Network security group <Project-Name>Public, assign the region, and click Next: Tags.
7. Add the appropriate tags, and click Next: Review + create.
8. If validation is successful, click Create. No additional changes will be needed for the public security group.
9. Click Microsoft Azure.
10. Under Azure services, click More services.
11. Under Categories, click Networking, and select Network Security Groups.
12. Click Add to add a new group.
13. Select the Subscription and Resource Group associated with the project.
14. Name the Network security group <Project-Name>Private (Example: PT-Private), assign the region, and click Next: Tags
15. Add the appropriate tags, and click Next: Review + create.
16. If validation is successful, click Create.
17. Click Go to Resource.
18. Under Settings, select Inbound security rules.
19. Click Add then verify the following:

    - Source = Any
    - Source port ranges = *
    - Destination = Any
    - Destination port ranges = *
    - Protocol = Any
    - Action = Allow
    - Priority = 100
    - Name = AllowAllInbound

20. Click Add.
21. Under Settings, select Outbound security rules.
22. Click Add, then verify the following:

    - Source = Any
    - Source port ranges = *
    - Destination = Any
    - Destination port ranges = *
    - Protocol = Any
    - Action = Allow
    - Priority = 100
    - Name = AllowAllOutbound

23. Click Add.

Creating Virtual Network

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Networking, and select Virtual Networks.
4. Click Add.
5. Select the Subscription and Resource Group associated with the project.
6. Name the virtual network and assign the region. (Example: PT-VNET1, US East US)
7. Click Next: IP Addresses
8. Choose the IP address space you wish to use.
9. Select the default subnet, and click Next: Security.
10. Verify all settings are disabled, and click Next: Tags.
11. Add the appropriate tags, and click Next: Review + create.
12. If validation is successful, click Create.

Creating Virtual Network Gateway

Note: This is where you set your gateway to ExpressRoute

1. Log into Microsoft Azure,
2. In the search window, type `Virtual Network Gateways` and press enter.
3. Click Add.
4. On the configuration screen add the following:

    - Subscription = Your project billing subscription.
    - Name = Name your VNG. (Example: PT-VNG)
    - Region = Choose the region you're in (Example: US East US)
    - Gateway type = ExpressRoute
    - SKU = Standard
    - Virtual Network = The Virtual network you created above. (Example: PT-Vnet)
    - Public IP Address = Create New
    - Public IP address name = Choose a name (Example: PT-VNG-IP)

5. Click Next: Tags.
6. Add appropriate tags, and click Next: Review + create.
7. If validation is successful, click Create.

Creating ExpressRoute circuit connection

1. In Microsoft Azure, navigate to the ExpressRoute circuit.
2. Click Connections, then click Add.
3. On the first screen, name the connection. Example: "PT-Connection"
4. Click Next: Settings.
5. On the Settings tab, add the following:
   - Virtual network gateway = PT-VNG
   - Redeem authorization = leave unchecked
   - Rounding weight = 0
6. Click Next: Tags.
7. Add appropriate tags, and click Next: Review + create.
8. If validation is successful, click Create.

Creating Network interfaces

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Networking, and select Network interfaces.
4. Click Add.
5. On the Basics tab, enter the following:
   - Subscription = Your Subscription
   - Resource Group = Your Resource Group
   - Name = Name of interface (Example: Svr1-1, Svr2-1, etc.)
   - Region = Region you're configuring for (Example: US East US)
   - Virtual Network = PT-VNET1
   - Subnet = Default
   - Private IP address assignment = Static
   - Private IP address = IP address that is in your VNET.
   - Network Security Group = PT-Private
   - Private IP address (IPv6) = Unchecked
6. Click Next: Tags.
7. Add appropriate tags, then click Next: Review + create.
8. If validation Is successful, click Create.
9. Log into Microsoft Azure Cloud Shell.
10. Enter the following command to enable Accelerated Networking:

```
az network nic update --name Svr1-1 --resource-group <resource group> --accelerated-networking true
```

11. Repeat 13 times to create 13 Network interfaces (one per test VM and one for the Log Server).

Creating Azure VMs

Creating test virtual machines

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Compute, and select Virtual Machines
4. Click Add, then select Virtual Machine.
5. In the Basics tab, add/modify the following:

   - Subscription = Your subscription
   - Resource group = PT
   - Virtual machine name = AZU01
   - Region = (US) East US
   - Availability options = No infrastructure redundancy required
   - Image = Windows Server 2016 Datacenter - Gen1
   - Azure Spot instance = No
   - Size = Standard_D4as_v4 (4 vcpus, 16GiB memory)
   - Username = <Username>
   - Password = <Password>
   - Public inbound ports = Allow selected ports
   - Select inbound ports = Click HTTP, HTTPS, SSH, and RDP.
   - Already have a Windows Server license? = No

6. Click Next: Disks.
7. In the Disks tab, add/modify the following:

   - OS disk type = Premium SSD
   - Encryption type = Default
   - Click Next: Networking

8. In the Networking tab, add/modify the following:

   - Virtual network = PT-VNET1
   - Subnet = Default
   - Public IP = (new) AZU01-ip
   - NIC network security group = Advanced
   - Configure network security group = PT-Private
   - Accelerated networking = On
   - Place this virtual machine behind an existing load balancing solution? = No

9. Click Next: Management.

   - In the Management tab, add/modify the following:
   - Boot diagnostics = On
   - OS guest diagnostics = Off
   - System assigned managed identity = Off
   - Enable auto-shutdown = Off
   - Enable backup = Off
   - Click Next: Advanced

10. In the Advanced tab, add/modify the following:

    - VM generation = Gen 1

11. Click Next: Tags.
12. Add appropriate tags, then click Next: Review + create.
13. If validation is successful, click Create.
14. Repeat 12 times to create 12 VMs.

Creating log server virtual machine

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Compute, and select Virtual Machines.
4. Click Add, and select Virtual Machine.
5. In the Basics tab, add/modify the following:

   - Subscription = Your subscription
   - Resource group = PT
   - Virtual machine name = AZULogServer
   - Region = (US) East US
   - Availability options = No infrastructure redundancy required
   - Image = Windows Server 2016 Datacenter - Gen1
   - Azure Spot instance = No
   - Size = Standard_B2s (4 vcpus, 16GiB memory)
   - Username = <Username>
   - Password = <Password>
   - Public inbound ports = Allow selected ports
   - Select inbound ports = Click HTTP, HTTPS, SSH, and RDP.
   - Already have a Windows Server license? = No

6. Click Next: Disks.
7. In the Disks tab, add/modify the following:

   - OS disk type = Standard HDD
   - Encryption type = Default

8. Click Next: Networking.
9. In the Networking tab, add/modify the following:

   - Virtual network = PT-VNET1
   - Subnet = Default
   - Public IP = (new) AZULogServer-ip
   - NIC network security group = Advanced
   - Configure network security group = PT-Private
   - Accelerated networking = Off
   - Place this virtual machine behind an existing load balancing solution? = No

10. Click Next: Management.
11. In the Management tab, add/modify the following:

    - Boot diagnostics = On
    - OS guest diagnostics = Off
    - System assigned managed identity = Off
    - Enable auto-shutdown = Off
    - Enable backup = Off

12. Click Next: Advanced.
13. In the Advanced tab, add/modify the following:

    - VM generation = Gen 1

14. Click Next: Tags.
15. Add appropriate tags, then click Next: Review + create.
16. If validation is successful, click Create.

Adding private network adapters to the servers

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Compute, and select Virtual Machines.
4. Select all 13 servers, and click Stop.
5. Verify the status of all 13 VMs are listed as "Stopped (deallocated."
6. Click the first test server to bring up its properties.
7. Click Networking.
8. Click Attach network interface.
9. In the pop-up menu, select the network interface associated with this VM, and click OK.
10. Once the interface is added, click it, and verify the Accelerated networking setting is Enabled.
11. Repeat for the remaining test systems and the log server.

Configuring all systems

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Compute, and select Virtual Machines.
4. Select all 13 servers, and click Start.
5. Log into the first server.
6. Launch a command prompt.
7. At the command prompt, enter the following command to disable the Firewall:

   ```
   netsh advfirewall set allprofiles state off
   ```

8. To remove password expiration, execute this command:

   ```
   net accounts /maxpwage:unlimited
   ```

9. To run Windows Update, execute this command:

   ```
   sconfig
   ```

10. Select option 6) Download and Install Updates, enter `A` for Search for (A)ll updates then enter `A` again.
11. Select Yes if prompted to restart.
12. Repeat until there are no more updates available.
13. To disable Windows Update, execute this command:

    ```
    sconfig
    ```

14. Select option 5) Windows Update Settings, enter `M` for to set updates to manual.
15. Exit sconfig by selecting option 15) Exit to Command Line.
16. Repeat on the remaining systems.
17. Edit C:\Windows\System32\Drivers\etc\hosts file on all systems with the private IP address of all systems.

Installing and configuring test software

1. Download iperf2: https://iperf.fr/iperf-download.php
2. Upload installation files to all test systems.
3. Log into all 12 test systems.
4. Create a folder called C:\iperf and install software there.

Setting up the Network Edge Device

Deploying Cisco CSR 1000v Network Edge Device

1. Log in to the Cloud Exchange Fabric portal at https://ecxfabric.equinix.com/dashboard.
2. In the top menu, select Network Edge→Create Virtual Device.
3. Select A Single Device without High Availability, and click Begin Creating Edge Device(s) at the bottom of the page.
4. Select Cisco CSR 1000V, and click Continue.
5. Select Ashburn as the device location, and use the pull-down menu to select the billing account. Click Next: Device Details.
6. Select Subscription as the license type.
    a. Enter PT-Cisco as the device name and PT as the prefix.
    b. Enter your email address for Device Status Notifications.
    c. Select AX as the Software Package and 16.09.05 as the Version.
    d. Select 1 Gbps as the Throughput.
    e. Click Next: Additional Services.
7. Under Add Access IP addresses, check the box for Primary Device.
    a. Under Add Users, check the box for Primary Device. Your username will automatically populate. Check the box beside your username.
    b. Enter your public IP address in the following format: aaa.bbb.ccc.ddd/32.
        i. Add additional access IP addresses as needed.
        ii. You can find your public address by searching "What's my IP" in a web browser.
8. Click Next: Review.
9. On the Review page, under Terms & Conditions, select Vendor Terms and review. Click Order Terms, and review. Once you've reviewed, click the box beside "I have read and understand these terms." and click Accept. Click Create Edge Device.

Deploying Versa SD-WAN Network Edge Device

1. Log in to the Cloud Exchange Fabric portal at https://ecxfabric.equinix.com/dashboard.
2. In the top menu, select Network Edge→Create Virtual Device.
3. Select A Single Device without High Availability, and click Begin Creating Edge Device(s) at the bottom of the page.
4. Select Versa Networks FlexVNF, and click Continue.
5. Review the instructions, and click Create SD-WAN device.
6. Select Ashburn as the device location, and use the pull-down menu to select the billing account. Click Next: Device Details.
7. Under Device Details:
    a. Under Licensing, fill in the following information:
        i. For LocalID, use the email address associated with your VERSA configuration.
        ii. For RemoteID, use the email address associated with your VERSA configuration.
        iii. For Serial Number, enter an Alphanumeric code.
        iv. Enter the IP address of the first SD-WAN controller.
        v. Enter the IP address of the second SD-WAN controller.
    b. Under Edge Device Details, provide a name for the device.
    c. Enter your email address under Device Status Notifications.
    d. Under Software Package and Version, fill in the following information:
        i. Select FlexVNF-4.
        ii. Select 16.1R2S8.
8. Click Next: Additional Services.
9. Under Add Access IP addresses, add the IP addresses for the two SD-WAN controllers you specified in the previous screen, and click Next: Review
10. On the Review page, under Terms & Conditions, click Order Terms, and review. Once you've reviewed, click the box beside "I have read and understand these terms." And click Accept. Click Create Edge Device.
11. Additional configuration was required on the VERSA Director. Equinix performed this part of the configuration on our behalf.

Deploying Fortinet FortiGate VM Series Network Edge Device

1. Log in to the Cloud Exchange Fabric portal at https://ecxfabric.equinix.com/dashboard.
2. In the top menu, select Network Edge→Create Virtual Device.
3. Select A Single Device without High Availability, and click Begin Creating Edge Device(s) at the bottom of the page.
4. Select Fortinet FortiGate VM Series, and click Continue.
5. Select Ashburn as the device location, and use the pull-down menu to select the billing account. Click Next: Device Details.
6. Under Device Details:
    a. Select Subscription as the license type.
    b. Enter PT-Fortinet as the device name and PT as the prefix.
    c. Enter your email address for Device Status Notifications.
    d. Select VM04/04V (UTM) as the Software Package and 6.0.7 as the Version.
    e. Select 1 Gbps as the Throughput.
    f. Click Next: Additional Services.
7. Under Additional Services:
    a. Under Add Users, check the box for Primary Device. Your username will automatically populate. Check the box beside your username.
    b. Under Add Access IP Addresses, enter your public IP address in the following format: aaa.bbb.ccc.ddd/32.
        i. Add additional access IP addresses as needed.
        ii. You can find your public address by searching "What's my IP" in a web browser.
8. Click Next: Review.
9. On the Review page, under Terms & Conditions, select Vendor Terms and review. Click Order Terms, and review. Once you've reviewed, click the box beside "I have read and understand these terms." and click Accept. Click Create Edge Device.

Setting up Amazon Web Services

Creating AWS Direct Connect circuit

1. Open a new browser tab and log into https://console.aws.amazon.com with your AWS credentials.
2. In the AWS console, select your credentials in the upper left, and select My Account.
3. Select and copy your Account ID number.
4. Switch to the Equinix Cloud Exchange Fabric browser page, and select Network Edge→View Virtual Devices.
5. Click the device you want to use for this connection.
6. Click Create Connection.
7. Under Amazon Web Services, click Select.
8. Under AWS Direct Connect - High Capacity, click Create Connection.
9. Click Create a Connection to Amazon Web Services.
10. Select the location of your Equinix Provider and the newly created device as the point of Origin, and select Ashburn, or your Equinix Direct Connect partner location as the Destination. Click Next.
11. Under Connection Details:
    a. Enter the name for this connection (We used PT-Cisco, -Versa, or -Fortinet depending on the target device).
    b. Paste or input your AWS Account ID (copied in step 3 above) into the AWS Account ID field.
    c. Select 1Gbps as the Connection Speed.
    d. Click Next.
12. Click Submit your order.
13. Click Accept hosted connection on AWS in the green box on the confirmation page.
14. In the AWS Management Console, locate and click Direct Connect.
15. In the connections page, click the connection showing the state "ordering."
16. In the upper-right corner of the screen, click Accept, and click Confirm.

Option 1: Creating a private virtual interface in AWS

1. In the AWS Management console, locate and click VPC.
2. Expand the menu for Virtual Private Network, and select Virtual Private Gateways.
3. Click Create Virtual Private Gateway.
4. Under Create Virtual Private Gateway:
    a. Provide a name tag.
    b. Leave the Amazon default ASN.
    c. Click create Virtual Private Gateway.
5. Select the Virtual private gateway you just created, and in the Actions menu at the top of the screen, select Attach to VPC.
6. Use the pull-down menu to select the VPC your VM instances use, and click Yes, Attach.
7. In the AWS Direct Connection side menu, click Virtual Interfaces.
8. Click Create virtual interface. On the next page:
    a. Select Private for non-IPSEC connections.
    b. Provide a name for the virtual interface.
    c. Under Connection, use the pull-down menu to select the connection you just created.
    d. Select Virtual Private Gateway for Gateway type and use the pull-down menu under Virtual private gateway for your VPC.
    e. Enter the VLAN number for this connection (found in the Equinix Cloud Exchange Fabric management console, in device details).
    f. Enter the BGP ASN number you want to use on the Equinix Edge Device. (We used 65501.)
    g. Expand the Additional settings menu item, and provide a BGP session password.
    h. Click Create virtual interface.
9. Click the newly created virtual interface.
10. Record the following information:
    a. General Configuration:
        i. Amazon side ASN
11. Peerings:
    a. BGP ASN
    b. BGP authentication key
    c. Your router peer IP
    d. Amazon router peer IP

Option 2: Creating a public virtual interface in AWS for IPSEC connections

1. In the AWS Management console, locate and click VPC.
2. Expand the menu for Virtual Private Network, and select Virtual Private Gateways.
3. Click Create Virtual Private Gateway.
4. Under Create Virtual Private Gateway:
    a. Provide a name tag.
    b. Leave the Amazon default ASN.
    c. Click create Virtual Private Gateway.
5. Select the Virtual private gateway you just created, and in the Actions menu at the top of the screen, select Attach to VPC.
6. Use the pull-down menu to select the VPC your VM instances use, and click Yes, Attach.
7. Under the Virtual Private Network (VPN) menu, click Customer Gateways.
8. Click Create Customer Gateway. On the next page:
    a. Enter the name of the gateway device.
    b. Change the routing to Dynamic.
    c. Enter the BGP ASN of the Network Edge device. (We used 65501.)
    d. Enter a public IP address that you own. You will use this IP address on the Network Edge device.
        i. Note: This IP address must be a public address that you own. Amazon will verify you own this address space before allowing any connections that leverage it to proceed.
    e. Click Create Customer Gateway.
9. In the upper-left corner of the screen, select the Services Menu, and select Direct Connect.
10. In the AWS Direct Connect side menu, click Virtual Interfaces.

11. Click Create virtual interface. On the next page:
    a. Select public virtual interface for IPSEC connections.
    b. Provide a name for the virtual interface.
    c. Under Connection, use the pull-down menu to select the connection you just created.
    d. Enter the VLAN number for this connection (found in the Equinix Cloud Exchange Fabric management console, in device details).
    e. Enter the BGP ASN number you want to use on the Equinix Edge Device. (We used 65501).
    f. Enter the network edge router IP address. This is public IP addresses you used to configure the Customer Gateway you created in the Virtual Private Network section of the VPC configuration in AWS Management Console.
    g. Enter the AWS router peer IP address. This is a second public IP address in the same network scope as the Customer Gateway you created in the Virtual Private Network section of the VPC configuration in AWS Management Console.
    h. Add at least one IP CIDR block you want advertised to AWS. We used the same IP CIDR used for the Customer Gateway.
    i. Expand Additional Settings and provide a BGP session password.
    j. Click Create virtual interface.
12. Click the newly created virtual interface.
13. Record the following information:
    a. General Configuration:
        i. Amazon side ASN
    b. Peerings:
        i. BGP ASN
        ii. BGP authentication key
        iii. Your router peer IP
        iv. Amazon router peer IP

Configuring Network Edge Device Peering

1. In the Equinix Cloud Exchange Fabric interface, click Network Edge→View Virtual Devices.
2. Click the Cisco router you created in previous steps.
3. Click the Connections tab in the middle of the page.
4. Under Virtual Connections, click the Amazon Web Services connection.
5. Scroll down to the bottom of the page. Under Primary BGP Information, populate the following fields using the information from AWS Direct Connect:
    a. Local ASN (we used 65501).
    b. Local IP address (this is the router peer IP addressed from AWS Direct Connect).
    c. Remote ASN (this is the Amazon side ASN from the AWS Direct Connect).
    d. Remote IP address (This is the Amazon router peer IP from AWS Direct Connect).
    e. BGP Authentication key (the password you provided for use by AWS Direct Connect).
    f. Click Accept.
6. When the BGP connection has been established and is up, VPC private (Option 1) or Amazon public (Option 2) IP addresses will be discovered for routing to other connections on the Network Edge device.

Configuring site-to-site VPN connections

1. In the AWS Management Console, click the VPC service.
2. Expand the Virtual Private Network (VPN) section and, click Site-to-Site VPN Connections.
3. Click Create VPN Connection. On the next page, configure the following:
    a. Provide a Name.
    b. Use the pull-down menu for Virtual Private Gateway and select the VPG associated with your VPC.
    c. Use the pull-down menu for Customer Gateway ID, and select the gateway you created for this connection.
    d. Leave the remaining options set to default, and click Create VPN Connection.
4. The new VPN connection will be created. Select the VPN and click the Tunnel Details tab in the bottom panel of the page.
5. Click Download Configuration.
6. Select the vendor and platform to receive the router configuration file. Use the information in this file to configure the Network Edge device by using SSH to login to the server and configure the IPSEC components, or to perform the configuration in the Versa Director or the FortiGate web UI.

Creating network interfaces

1. Click Microsoft Azure.
2. Under Azure services, click More services.
3. Under Categories, click Networking, and select Network interfaces.
4. Click Add.
5. On the Basics tab, enter the following:
   - Subscription = Your Subscription
   - Resource Group = Your Resource Group
   - Name = Name of interface (Example: Svr1-1, Svr2-1, etc.)
   - Region = Region you're configuring for (Example: US East US)
   - Virtual Network = PT-VNET1
   - Subnet = Default
   - Private IP address assignment = Static
   - Private IP address = IP address that is in your VNET.
   - Network Security Group = PT-Private
   - Private IP address (IPv6) = Unchecked
6. Click Next: Tags.
7. Add appropriate tags, then click Next: Review + create.
8. If validation Is successful, click Create.
9. Log into Microsoft Azure Cloud Shell.
10. Enter the following command to enable Accelerated Networking:
    ```
    az network nic update --name Svr1-1 --resource-group <resource group> --accelerated-networking true
    ```
11. Repeat 13 times to create 13 Network interfaces (one per test VM and one for the log server).

Creating AWS VMs

Creating test virtual machines

1. Under All Services→Compute, click EC2.
2. Click Launch Instance, then Launch Instance in the drop-down menu.
3. In the search window, type `Windows 2016`
4. Next to Microsoft Windows Server 2016 Base, click Select.
5. Select the size of t3.xlarge, then click Next: Configure Instance Details.
6. On the Step 3: Configure Instance Details tap, choose the following:
   - Number of instances = 1
   - Purchasing options = Unchecked
   - Network = vpc-057ed46046f86c32e
   - Subnet = subnet-0a13b525a6162e1ee | Default in us-east-1b
   - Auto-assign Public IP = Enabled
   - Placement Group = Unchecked
   - Capacity Reservation = Open
   - Domain join directory = No directory
   - IAM role = None
   - CPU options = Unchecked
   - Shutdown behavior = Stop
   - Enable termination protection = Unchecked
   - Monitoring = Unchecked
   - Tenancy = Shared - Run a shared hardware instance
   - Elastic Graphics = Unchecked
   - T2/T3 Unlimited = Checked
7. Click Next:Add Storage.
8. In the Add Storage tab, add/modify the following:
   - Size = 50GiB

9. Click Next: Add Tags.
10. Click Add Tags, add appropriate tags, and click Next: Configure Security Group.
11. On the Step 6: Configure Security Group tab, select the following:
    - Assign a security group = Select an existing security group
    - Security Group ID = sg-0b7df00caac639d26
12. Click Review and Launch.
13. Click Launch.
14. The first time, create a new key pair by selecting the drop-down menu and selecting the Create a new key pair.
15. Name the new key pair, and click Download Key Pair.
16. On subsequent installations, select Choose an existing key pair, and select the appropriate key pair.
17. Click the box to acknowledge, then click Launch Instances.
18. Click View Instances to verify the instance is initializing.
19. Repeat 12 times to create 12 VMs.

Creating log server virtual machine

1. Under All Services→Compute, click EC2.
2. Click Launch Instance, then Launch Instance in the drop-down menu.
3. In the search window, type `Windows 2016`
4. Next to Microsoft Windows Server 2016 Base, click Select.
5. Select the size of t2.medium, then click Next: Configure Instance Details.
6. On the Step 3: Configure Instance Details tap, choose the following:
    - Number of instances = 1
    - Purchasing options = Unchecked
    - Network = vpc-057ed46046f86c32e
    - Subnet = subnet-0a13b525a6162e1ee | Default in us-east-1b
    - Auto-assign Public IP = Enabled
    - Placement Group = Unchecked
    - Capacity Reservation = Open
    - Domain join directory = No directory
    - IAM role = None
    - CPU options = Unchecked
    - Shutdown behavior = Stop
    - Enable termination protection = Unchecked
    - Monitoring = Unchecked
    - Tenancy = Shared - Run a shared hardware instance
    - Elastic Graphics = Unchecked
    - T2/T3 Unlimited = Checked
7. Click Next:Add Storage.
8. In the Add Storage tab, add/modify the following:
    - Size = 50GiB
9. Click Next: Add Tags.
10. Click Add Tags, add appropriate tags, and click Next: Configure Security Group.
11. On the Step 6: Configure Security Group tab, select the following:
    - Assign a security group = Select an existing security group
    - Security Group ID = sg-0b7df00caac639d26
12. Click Review and Launch.
13. Click Launch.
14. Select Choose an existing key pair, and select the appropriate key pair.
15. Click the box to acknowledge, then click Launch Instances.
16. Click View Instances to verify the instance is initializing.
17. Repeat 12 times to create 12 VMs.

Adding private network adapters to the servers.

1. Under All Services→Compute, click EC2.
2. Under the left menu, Select Network & Security→Network Interfaces.
3. Click Create Network Interfaces.
4. Add the following:
   - Description = Name of host it will be assigned to.
   - Subnet = subnet-0a13b525a6162e1ee
   - IPv4 Private IP = Select the Custom radial button.
   - IPv4 Address = IP Address available for the device.
   - Elastic Fabric Adapter = Unchecked.
   - Add Tag = Tag with project name.
   - Security Groups = Select the private security group.
5. Click Create.
6. Repeat for the remaining test systems and the log server.

Configuring all systems

1. Under All Services→Compute, click EC2.
2. Click Running instances.
3. Select all images.
4. Select all 13 servers, click Action, click Instance state, and click Start.
5. Log into the first server.
6. Launch a command prompt
7. At the command prompt, enter the following command to disable the Firewall:

   ```
   netsh advfirewall set allprofiles state off
   ```

8. To remove password expiration, execute this command:

   ```
   net accounts /maxpwage:unlimited
   ```

9. To run Windows Update, execute this command:

   ```
   sconfig
   ```

10. Select option 6) Download and Install Updates, enter `A` for Search for (A)ll updates then enter `A` again.
11. Select Yes if prompted to restart.
12. Repeat until there are no more updates available.
13. To disable Windows Update, execute this command:

    ```
    sconfig
    ```

14. Select option 5) Windows Update Settings, enter `M` for to set updates to manual.
15. Exit sconfig by selecting option 15) Exit to Command Line.
16. Repeat on the remaining systems.
17. Edit C:\Windows\System32\Drivers\etc\hosts file on all systems with the private IP address of all systems.

Installing and configuring test software

1. Download iperf2 from https://iperf.fr/iperf-download.php
2. Upload installation files to all test systems.
3. Log into all 12 test systems.
4. Create a folder called C:\iperf and install software there.

Running the tests

Running single-instance UDP tests

We ran the single-instance UDP tests by logging into a single Azure instance and a single AWS instance. The AWS instance acted as the server, and the Azure instance acted as the client. The scripts were located in the C:\iperf folder. Single-run test results are in C:\Results. We repeated the tests until we performed three continuous successful runs. Tests ran with 218B packets to simulate standard codecs for voice.

1. Log into AWS and select EC2 instances.
2. Right-click AWS01 (instance i-07c0cf07f0c76161b), select Networking, and select Detach Network Interface.
3. Select "eth1…AWS01-1" adapter, and click Detach.
4. Start the instance.
5. Once instance is up and a public IP has been assigned, Right-click AWS01 (instance i-07c0cf07f0c76161b), select Networking, and select Attach Network Interface.
6. Select the "eni-0f9912cbed1079f40 (AWS01-1)" adapter, and click Attach.
7. Log into Azure, and select Virtual Machines.
8. Select the box next to the AZU01 server, and click Start.
9. Log into both systems using Remote Desktop.
10. On the AWS system, navigate to C:\iperf\
11. Execute the command `1x-start-218b-server.bat` This starts the server and begins logging to the C:\Results folder.

```
Run Script: C:\iperf\1x-218B-server.bat > C:\Results\iperf-1x-218B-Azure-AWS-Server-5000.output.txt
Test Script: C:\iperf\iperf.exe -s -u -i 1 -B AWS01-1 -p 5000 -w 4M -l 218.0B -f m -e
```

12. On the Azure system, navigate to C:\iperf\
13. Edit the 1x-218B-client.bat file. Change the "-w NNNNNpps" section to the number of packets per second to run. Save file.
14. Execute the command `1x-start-218b-client.bat` This starts the server and begins logging to the C:\Results folder.

```
Run Script: C:\iperf\1x-218B-client.bat > C:\Results\UDP-iperf-1x-218B-Azure-AWS-client-5000.output.
txt
Test Script: C:\iperf\iperf.exe -c AWS01-1 -u -P 1 -i 1 -B AZU01-1 -p 5000 -w 4M -l 218.0B -f m -b
61000pps -t 310 -d -e
```

15. Once test completes, verify the logs on both the AWS and Azure systems in C:\Results have not had more than 0.005% dropped packets.
16. Rename logs with Pass/Fail and packet size in the filename.
17. Close all outstanding CMD prompt windows.
18. Adjust the "-w NNNNNpps" variable and repeat steps 10 to 17 until the results are clean (less than 0.005% drop rate bidirectional) and three passing runs with the same packets-per-second rate occur.

Running the TCP tests

We performed TCP testing from a jump server located on the PT campus. TCP testing ran 24 systems, 12 in AWS acting as servers, and 12 in Azure acting as clients. There was also an additional log server in AWS and one in Azure. Several scripts ran remotely to achieve the 12 simultaneous streams needed to test the TCP load. Scripts on the Jump Server are located in the C:\Tools\PT_Run_Scripts\Official_IMIX-Scripts folder. This testing ran using 12 simultaneous streams of 1350B.

1. Log into AWS and select EC2 instances.
2. Right-click AWS01 to AWS12, then on AWSLogServer select Networking, and select Detach Network Interface.
3. Select the adapter with no public IP associated, then click Detach.
4. Select the box next to all AWS01 to AWS12 servers, along with AWSLogServer, and click Start.
5. Once instances are up and a public IP has been assigned, Right-click AWS01to AWS12 and the AWSLogServer, select Networking, and select Attach Network Interface.
6. Select the adapter that matches the machine's name, and click Attach.
7. Log into Azure, and select Virtual Machines.
8. Select the box next to all AZU01 to AZU12 servers, along with AZULogServer, and click Start.
9. Log into the jump system.
10. Edit the C:\Windows\System32\drivers\etc\hosts file on the jump system with the public IP addresses of the servers.
11. Execute the `stop-Cleanup-Systems.bat` script to make sure that all systems are clean.
12. Execute the `Run-TCP-Tests.bat` script.
13. Upon completion of testing, execute the `Copy-Logs.bat` script to copy the log files from the test systems to the log systems.
14. Log into the log systems for each cloud environment. Verify that the data is complete and accurate.
15. Rename log files to reflect which run iterations they represent.
16. Log results to the spreadsheet.
17. On the jump system, execute the `Stop-Cleanup-Servers.bat` script to make sure that all systems are clean.
18. Repeat steps 12 to 17 until you complete three successful runs.

Running the IMIX TCP tests

We performed the IMIX TCP testing from a jump server located on the PT campus. IMIX testing ran 24 systems, 12 in AWS acting as servers, and 12 in Azure acting as clients. There was also an additional log server in AWS and one in Azure. Several scripts ran remotely to achieve the 12 simultaneous streams needed to test the IMIX load. Scripts on the jump system were located in the C:\Tools\PT_Run_Scripts\Official_IMIX-Scripts folder. This testing ran using 12 simultaneous streams consisting of (7) at 64B, (4) at 512B, and (1) at 1350B. Due to inconsistent performance within AWS cloud at 1500B, we ran the final stream at 1350B.

1. Log into AWS and select EC2 instances.
2. Right-click AWS01 to AWS12, then on AWSLogServer select Networking, and select Detach Network Interface.
3. Select the adapter with no public IP associated, and click Detach.
4. Select the box next to all AWS01 to AWS12 servers, along with AWSLogServer, and click Start.
5. Once instances are up and a public IP has been assigned, right-click AWS01to AWS12 and the AWSLogServer, select Networking, and select Attach Network Interface.
6. Select the adapter that matches the machine's name, and click Attach.
7. Log into Azure, and select Virtual Machines.
8. Select the box next to all AZU01 to AZU12 servers, along with AZULogServer, and click Start.
9. Log into the jump system.
10. Edit the C:\Windows\System32\drivers\etc\hosts file on the jump system with the public IP addresses of the servers.
11. Execute the `Stop-Cleanup-Systems.bat` script to make sure that all systems are clean.
12. Execute the `Run-IMIX-Tests.bat script`.
13. Upon completion of testing, execute the `Copy-Logs.bat` script to copy the log files from the test systems to the log systems.
14. Log into the log systems for each cloud environment. Verify that the data is complete and accurate.
15. Rename log files to reflect which run iterations they represent.
16. Log results to the spreadsheet.
17. On the jump system, execute the `Stop-Cleanup-Servers.bat` script to make sure that all systems are clean.
18. Repeat steps 12 to 17 until you complete three successful runs.

Read the report at http://facts.pt/mV8n7Ye ▶

This project was commissioned by Equinix.

Principled Technologies®

Facts matter.®