Maidenhead Bridge

zscaler

Cloud Security Connector Mux (1/2/4/8) – Azure with Private Cloud Private Access

(For Azure Cloud)

Version 4.0.4

June 2023

Table of Contents

1 Introduction to Cloud Security Connectors for Zscaler.

The Cloud Security Connector (CSC) is a device that enables easy deployments of the Zscaler ZIA solution in any customer environment. There are CSC models for Virtual Platforms (VMware, Hyper-V) and Public Clouds (Azure, AWS, etc.).

The Cloud Security Connector Multiplex (CSC Mux) for Azure is a virtual machine connecting internal Azure resources to Zscaler ZIA.

The CSC Mux for Azure lets you connect securely to Zscaler ZIA up to 6.4 Gbps without hassle.

The primary purpose of the CSC family is simplicity. The CSC for Azure comes with all configurations required.

After launching the CSC Mux from the Azure Marketplace using the ARM template provided, the CSC Mux will automatically select the best Zscaler nodes and do IPsec tunnels to Primary and Secondary Zscaler Nodes.

The CSC Mux comes in 4 models:

1. CSC Mux 1 with PriCPA: 1 x IPSec to Zscaler (400 Mbps)
2. CSC Mux 2 with PriCPA: 2 x IPSec to Zscaler (800 Mbps)
3. CSC Mux 4 with PriCPA: 4 x IPsec to Zscaler (1,6 Gbps)
4. CSC Mux 8 with PriCPA: 8 x IPsec to Zscaler (3,2 Gbps)

The CSC Mux contains the perfect configuration for IPsec tunnels, firewall rules, and necessary routing tables.

All Zscaler functionalities are available, providing complete visibility of all Internet traffic.

In addition to this, the CSC Mux provides high availability changing the default route to Zscaler when configured as High Availability pair, and an easy way to manage direct bypasses to trusted sites using your public IP.

Includes Private Cloud Private Access (PriCPA) functionality that allows you to create a full mesh among the CSCs communicating your private traffic on a Zero Trust model.

Simple to install with complete management using DevOps change management tools like Amazon Systems Manager, Rundeck, Ansible, etc; and SSH.

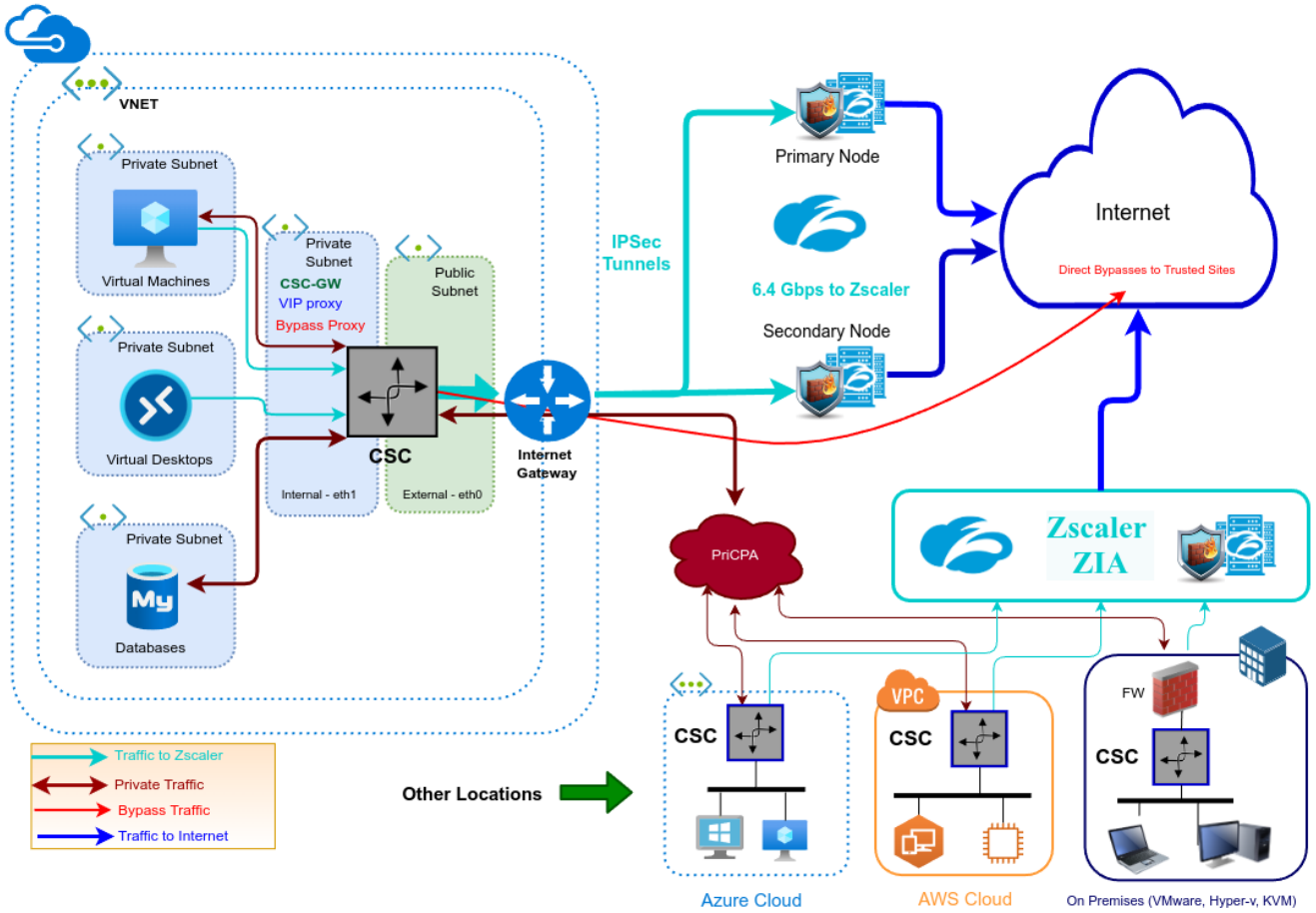
2 Key benefits of the Cloud Security Connector Mux for Azure

- No Networking knowledge is required.

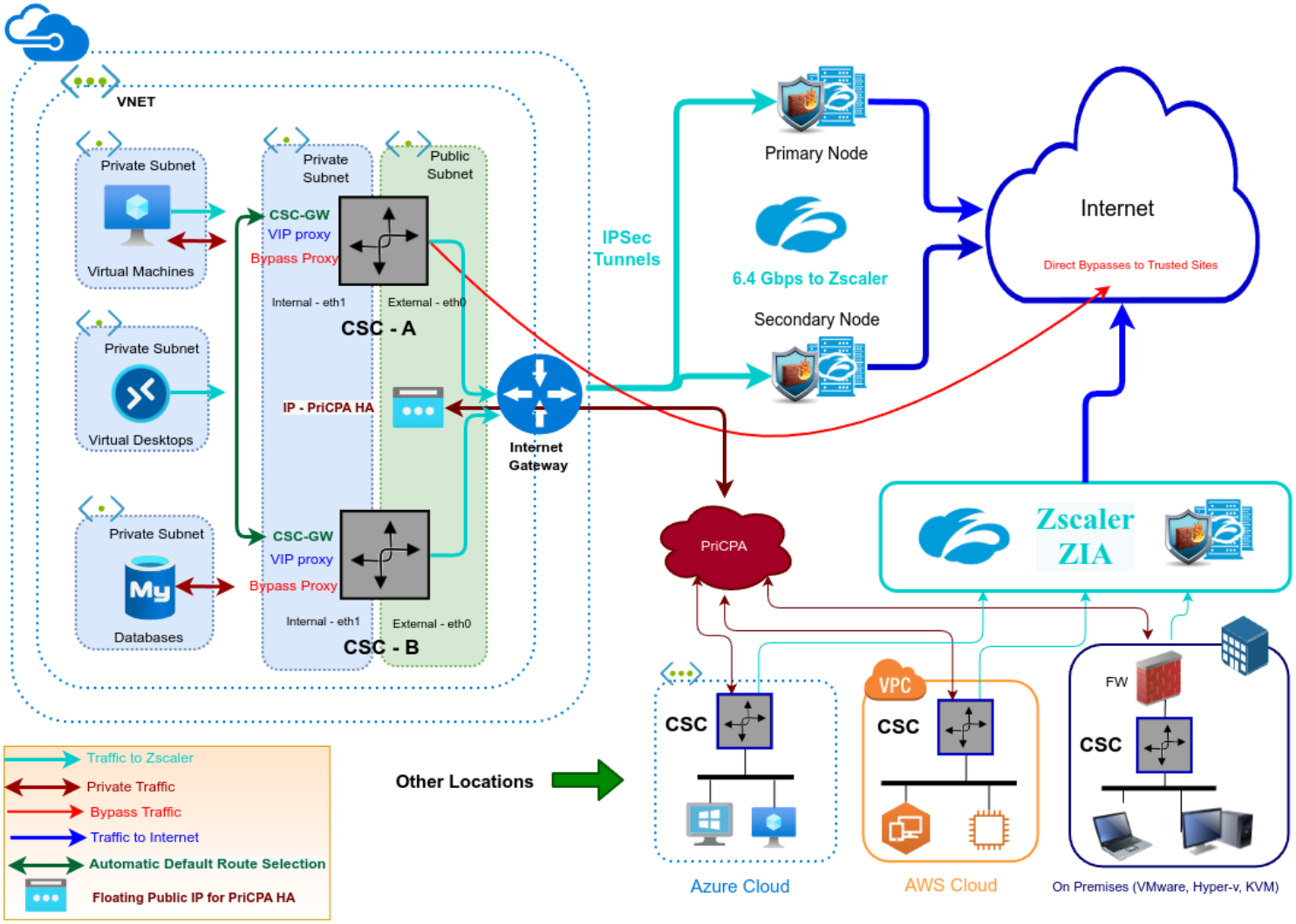
- Automated deployment from Marketplace, ARM template or your tool of choice. (i.e. Terraform)
- Enables any Location to be connected to Zscaler up to 6,4 Gbps.
- With Private Cloud Private Access you can connect all sites securely on a Zero Trust model. The CSC secures your Private Traffic between your physical and cloud locations.
- The CSC comes with the optimal values to work with Zscaler ZIA.
- Full tunnel redundancy.
- High Availability with automatic "Next-Hop" selection on multiple routes.
- All traffic steering options supported:
  - Route all traffic to Zscaler (or http/s only).
  - Use of PAC files.
  - Use of Explicit Proxy.
  - No default Route scenarios.
- Multiple options to Bypass Traffic via dedicated Public IP:
  - Layer 7 Proxy Bypass to Trusted Web Sites.
  - Layer 4 Routed Bypass: TCP, UDP and ICMP per source/destination Network and Port (UDP/TCP)
- Cloud Firewall and Cloud Web Security.
- Complete visibility of internal IPs on Zscaler Console.
- No operational burden for Administrators.
- Full hardened device.
- Multiple tools for testing and troubleshooting included: Traffic Logs. TCPDump, Speed Test, MTR (MyTraceRoute), Keepalives statuses, Etc.
- Allow the internal communication between your locations with Private Cloud Private Access.
- Management via SSH, AWS Systems Manager, Rundeck or similar. (Ansible, Salt, Etc.)

3 Network Diagrams

3.1 CSC Mux for Azure – Single deployment



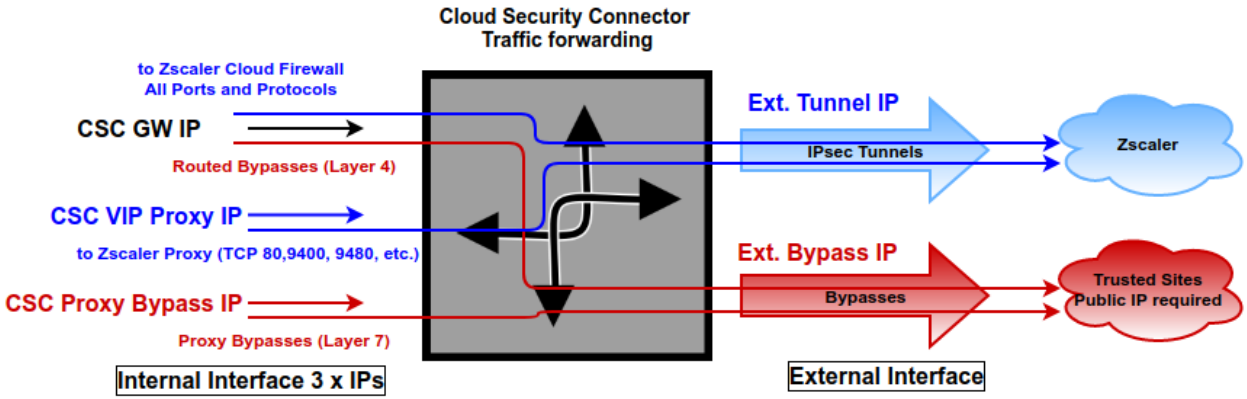
3.2 CSC Mux for Azure – High Availability Deployment



3.3 Traffic Forwarding: Routing and Proxying all together.

The most significant benefit of the Cloud Security Connector for Zscaler is that it covers all possible scenarios (routed traffic, PAC files, explicit proxy, Etc.) for any device on your organization: Laptops, Desktops, Servers, IoT devices, Virtual Desktops, Etc.

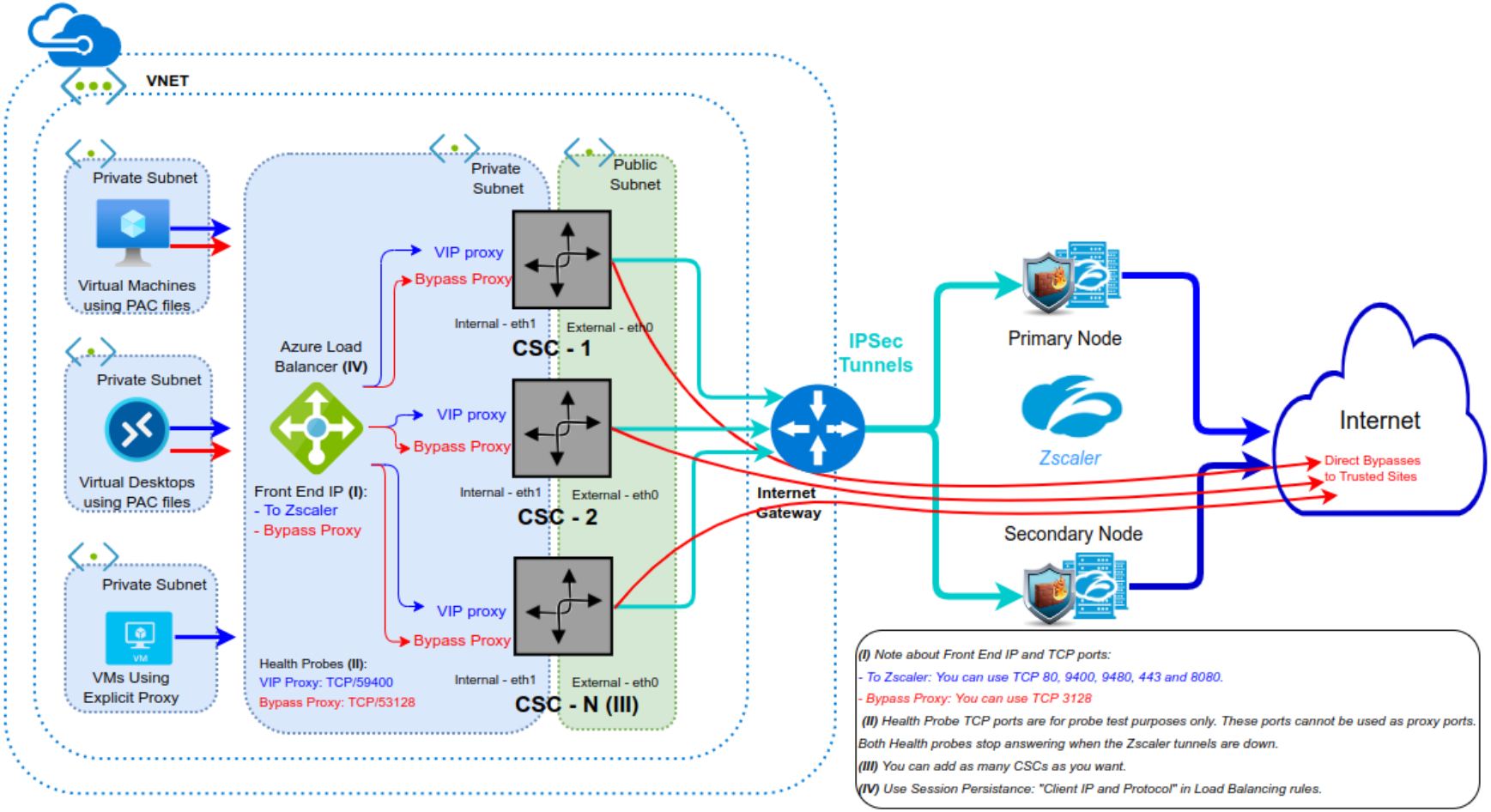
The following picture shows the CSC working will all scenarios combined.



The function of each internal IP is the follwing:

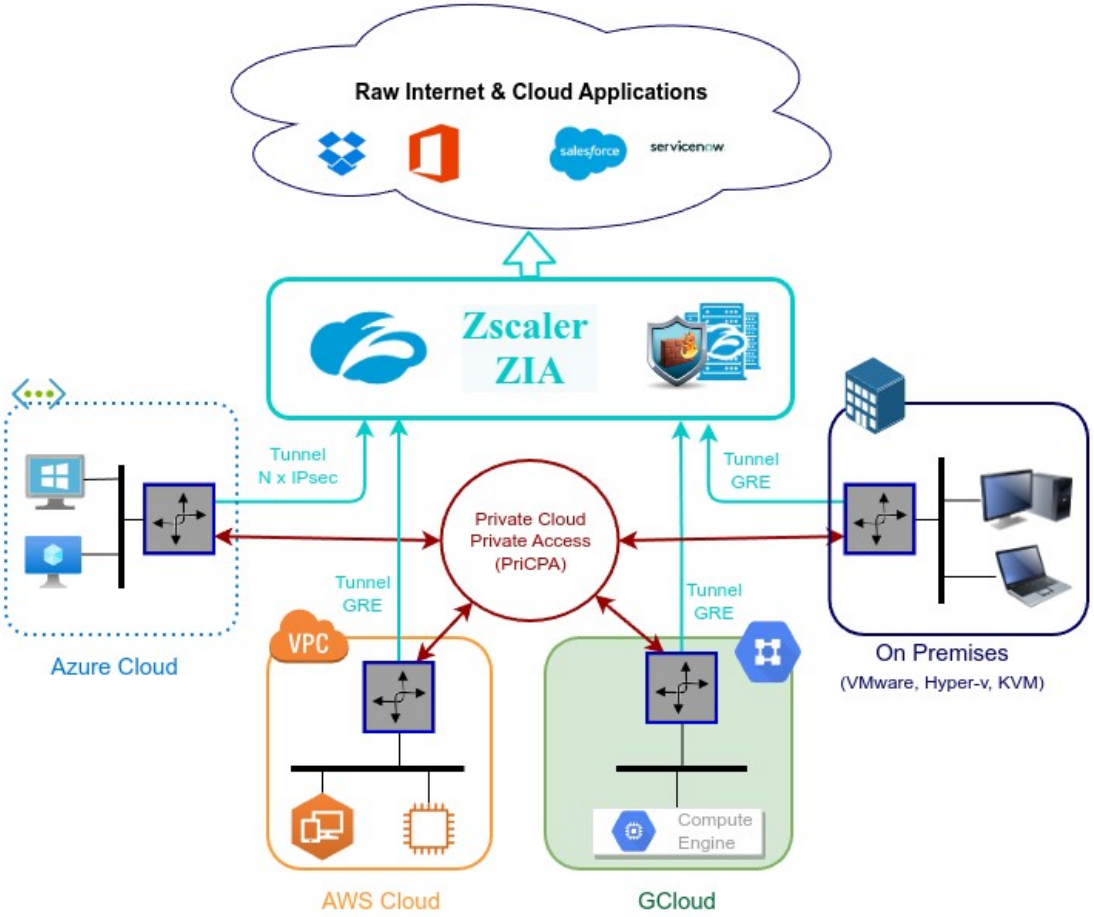
IP	Type	Function
CSC GW	Gateway	Used as Gateway for traffic to Zscaler and bypasses using "Routed Bypass" (Layer 4) functionality.
CSC Vip Proxy	Proxy	Used as Proxy for traffic to Zscaler.
CSC Proxy Bypass	Proxy	User as Proxy for bypasses using "Proxy Bypass" (Layer 7) functionality.

3.4 High Availability using Azure Load Balancer



3.5 Private Cloud Private Access (PriCPA)

With the CSCs for Zscaler, you can create your Private Cloud for connecting all your internal devices in a Zero Trust model from your physical and cloud locations.



4 Understanding the CSC Mux with PriCPA

4.1 What problem the CSC Multiplex solves?

GRE tunnels are the recommended method to forward traffic to Zscaler, but Azure cloud doesn't support GRE (Generic Router Encapsulation) packets.
From: https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-faq

What protocols can I use within VNets?

You can use TCP, UDP, and ICMP TCP/IP protocols within VNets. Unicast is supported within VNets, with the exception of Dynamic Host Configuration Protocol (DHCP) via Unicast (source port UDP/68 / destination port UDP/67) and UDP source port 65330 which is reserved for the host. Multicast, broadcast, IP-in-IP encapsulated packets, and Generic Routing Encapsulation (GRE) packets are blocked within VNets.

IPSec tunnels to Zscaler can go up to 400 Mbps only. If you need more bandwidth to Zscaler, you need to aggregate multiple tunnels to Zscaler from different Public IPs, as this article says:
From: https://help.zscaler.com/zia/configuring-ipsec-vpn-tunnel

> Zscaler IPSec tunnels support a limit of 400 Mbps for each public source IP address. If your organization wants to forward more than 400 Mbps of traffic, Zscaler recommends configuring more IPSec VPN tunnels with different public source IP addresses. For example, if your organization forwards 800 Mbps of traffic, you can configure two primary VPN tunnels and two backup VPN tunnels. If your organization forwards 1200 Mbps of traffic, you can configure three primary VPN tunnels and three backup VPN tunnels.

Suppose you want to create a setup of Gigabits per second to Zscaler with discrete elements. In that case, you will find that it is impossible in most cases or extremely expensive. You will need to implement and configure several components: Firewall, Load balancers, Routers, VPN Concentrators, Etc.

The CSC Multiplex has everything ready to work. You don't need to worry about the complexity of the solution.

The CSC Multiplex provides connectivity to all ports and protocols to Zscaler without restrictions and interacts with the Azure fabric to provide High Availability.

Your only task is to input the IPsec VPN credential.

Problem solved. You can quickly overcome the limitation of IPsec tunnels to Zscaler and reach speeds up to 6.4 Gbps or more.

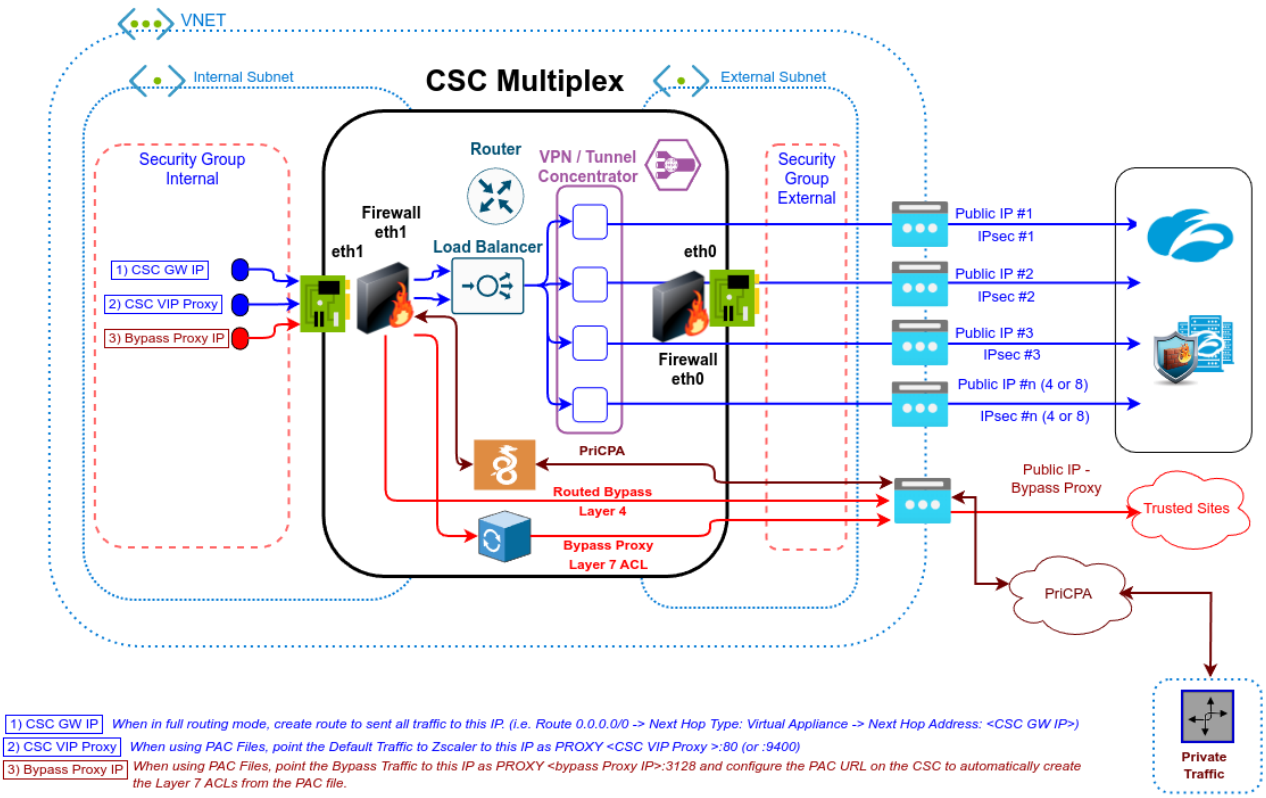
4.2 What does the CSC Multiplex do?

The CSC Multiplex does the job of multiple devices: Firewall, Load Balancer, VPN Concentrator, Router and Proxy, and there are 4 models:

1. CSC Mux 1 with PriCPA: 1 x IPSec to Zscaler (400 Mbps)
2. CSC Mux 2 with PriCPA: 2 x IPSec to Zscaler (800 Mbps)
3. CSC Mux 4 with PriCPA: 4 x IPsec to Zscaler (1,6 Gbps)
4. CSC Mux 8 with PriCPA: 8 x IPsec to Zscaler (3,2 Gbps)

***IMPORTANT**: PriCPA can reach 1 Gbps for private traffic in all models.*

The following diagram shows the internal architecture of the CSC Multiplex:



PriCPA uses Wireguard protocol. "WireGuard" and the "WireGuard" logo are registered trademarks of Jason A. Donenfeld.

4.3 The CSC Mux 1, 2, 4 and 8 in action

4.3.1 Speed Test with CSC Mux 1

```
Selection: 4

SPEED TEST
This is experimental. We are using third party tools. (Speedtest.net)
Results can be inaccurate or none. The test takes a while


Aggregated Bandwidth Download:  862.19 Mbps
```

4.3.2 Speed Test with CSC Mux 2

```
Selection: 4

SPEED TEST
This is experimental. We are using third party tools. (Speedtest.net)
Results can be inaccurate or none. The test takes a while


Aggregated Bandwidth Download:  1343.78 Mbps
```

4.3.3 Speed Test with CSC Mux 4

Speed expected: 1600 Mbps

```
Selection: 4

SPEED TEST
This is experimental. We are using third party tools. (Speedtest.net)
Results can be inaccurate or none. The test takes a while


Aggregated Bandwidth Download:  2472.33 Mbps
```

4.3.4 Speed Test with CSC Mux 8

```
Selection: 4

SPEED TEST
This is experimental. We are using third party tools. (Speedtest.net)
Results can be inaccurate or none. The test takes a while


Aggregated Bandwidth Download:  4363.72 Mbps
```

4.3.5 Iperf test using PriCPA

This Iperf test is between a CSC on Azure and another CSC on AWS. Both are located in the US East region.

PriCPA provides 1 Gbps throughput.

```
ubuntu-ds-01@ubuntu-ds-01:~$ iperf -c 10.3.200.16 -p 5500 -t 60
------------------------------------------------------------
Client connecting to 10.3.200.16, TCP port 5500
TCP window size: 1.62 MByte (default)
------------------------------------------------------------
[  3] local 10.2.3.5 port 56046 connected with 10.3.200.16 port 5500
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-60.0 sec  6.43 GBytes   921 Mbits/sec
```

```
wg0                                                                                                                bmon 4.0
Interfaces        RX bps       pps     %   TX bps       pps     %
 eth0              6.32Mb       0          1.01Gb    84.35K
 eth1            967.52Mb       0          3.02Mb     5.71K
 ztun1a            2.56Kb       0          1.66Kb         1
 ztun2a            1.32Kb       0           824b          0
 ztun3a            1.37Kb       0          1.05Kb         0
 ztun4a            3.12Kb       0          2.62Kb         2
 wg0               4.39Mb       0        979.70Mb    84.34K

     Mb                    (RX Bytes/second)                                  Mb                    (TX Bytes/second)
   5.53 ...|..|..|.||||.|.||.|..|....|.||...|.......................        981.51 ||||||||||.||||||||||||||||||||||.|||||||....................
   4.61 ||||||||||||||||||||||||||||||||||||||||||||||.................        817.92 ||||||||||||||||||||||||||||||||||||||||||||||||....................
   3.69 ||||||||||||||||||||||||||||||||||||||||||||||.................        654.34 ||||||||||||||||||||||||||||||||||||||||||||||||...................
   2.76 ||||||||||||||||||||||||||||||||||||||||||||||.................        490.75 ||||||||||||||||||||||||||||||||||||||||||||||||...................
   1.84 ||||||||||||||||||||||||||||||||||||||||||||||.................        327.17 ||||||||||||||||||||||||||||||||||||||||||||||||...................
   0.92 ||||||||||||||||||||||||||||||||||||||||||||||::::::::::::::::::        163.58 ||||||||||||||||||||||||||||||||||||||||||||||||::::::::::::::::::
        1   5   10   15   20   25   30   35   40   45   50   55   60                1   5   10   15   20   25   30   35   40   45   50   55   60
```

5 Creating the CSC Mux for Zscaler with PriCPA

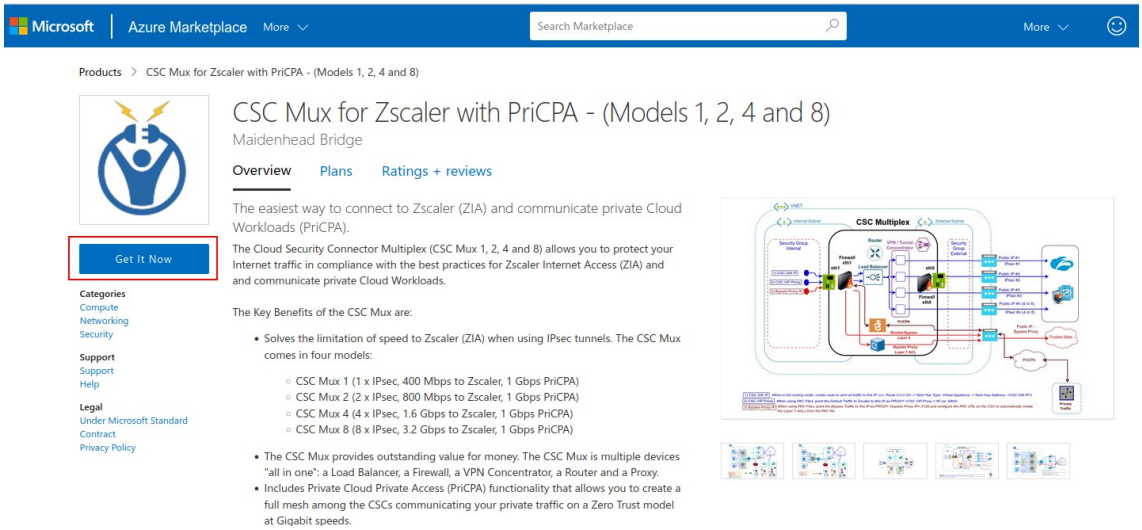
5.1 Prerequisites

Before launching the CSC Mux 1, 2, 4 or 8 for Azure, you need to have these elements ready:

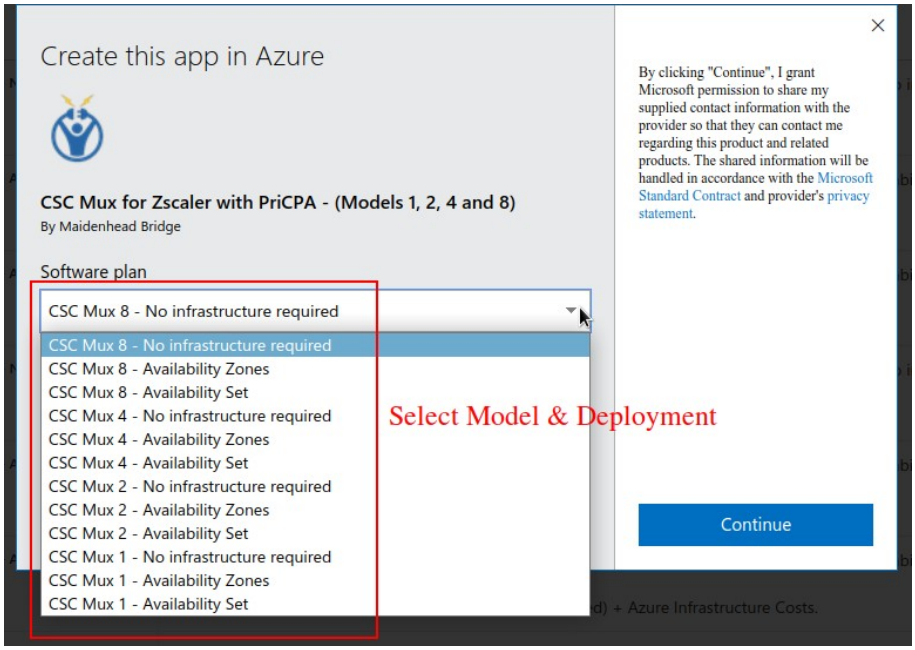
1. **(Optional) SSH Key:** If you want to access the CSC using SSH keys. If not, you can use a password during the installation.
2. **Virtual Network**
3. **External Subnet:** The External Subnet must be on the same Virtual Network as the Internal Subnet.
4. **Internal Subnet:** The Internal Subnet must be on the same Virtual Network as the External Subnet.

5.2 Launching the CSC Mux for Azure Marketplace

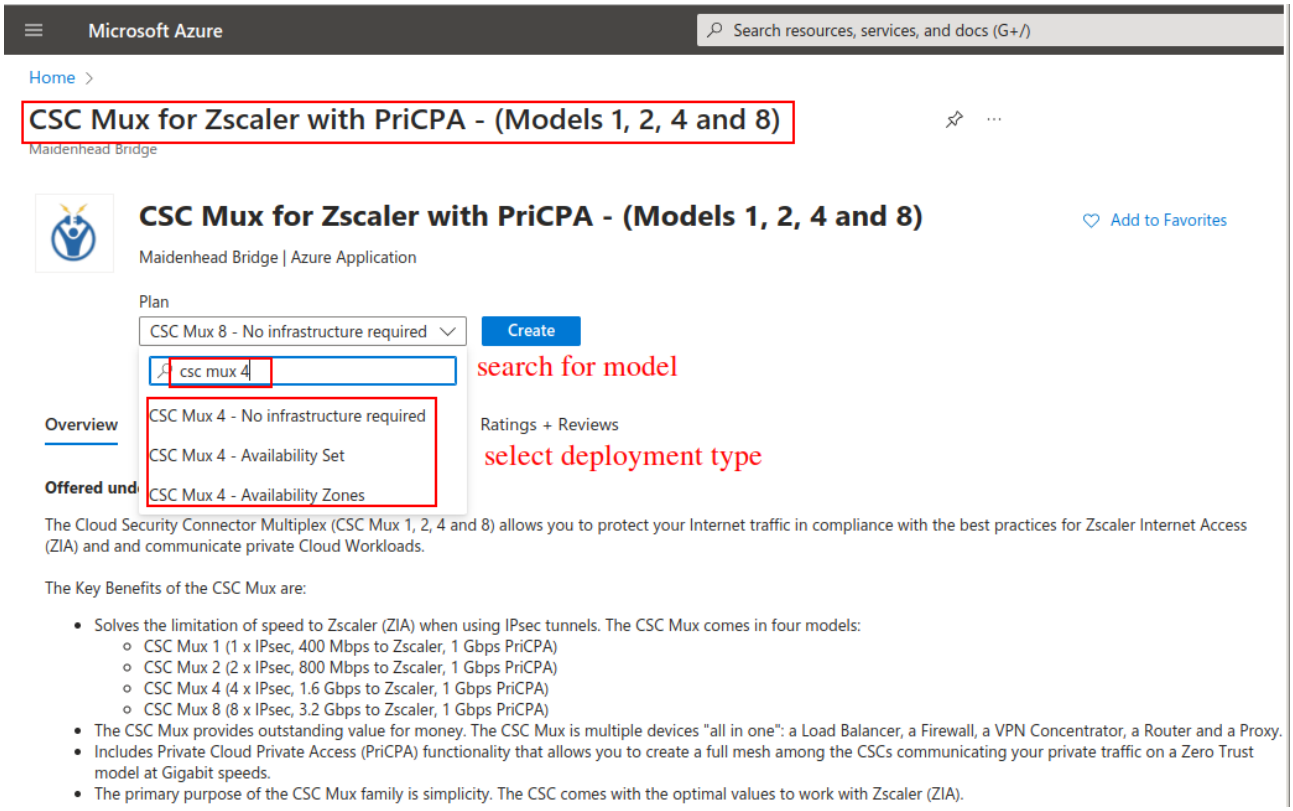
Go to Azure Marketplace, search for "Maidenhead Bridge", and select **"CSC Mux for Zscaler with PriCPA - Model (1, 2, 4 and 8)"**.
(https://azuremarketplace.microsoft.com/en-us/marketplace/apps/maidenhead-bridge.zs-csc-mux-azure-application?tab=Overview)



→ Click "Get it Now"



→ Select "Software Plan" and click "Continue". You will be redirected to your Azure Portal.



→ Please, Check the Plan and click "Create".

Home > CSC Mux for Zscaler with PriCPA - (Models 1, 2, 4 and 8) >

Create CSC Mux for Zscaler with PriCPA - (Models 1, 2, 4 and 8) ...

1 Basics | 2 Virtual Machine Settings | 3 Networking | 4 configUserData.json File | 5 Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : MHB

Resource group * : CSC-East-US
Create new

1

Instance details

Location * : East US

2

Please, check if the Location (Region) selected previously supports Availability Zones (see: https://docs.microsoft.com/en-us/azure/availability-zones/az-region).

Select Single or HA configuration * :
- ○ Deploy Single (1x) CSC
- ◉ Deploy High Availability (2x) CSCs

3

Choose the Availability Zones for each Cloud Security Connector.

4

First CSC Availabity Zone * : Zone 1

Second CSC Availabity Zone * : Zone 2

CSC_Name * : zs-csc-mux-4-az-doc

5

Admin Username : cscadmin

Authentication type * :
- ◉ Password
- ○ SSH Public Key

6

Password * : ••••••••••••••••••••

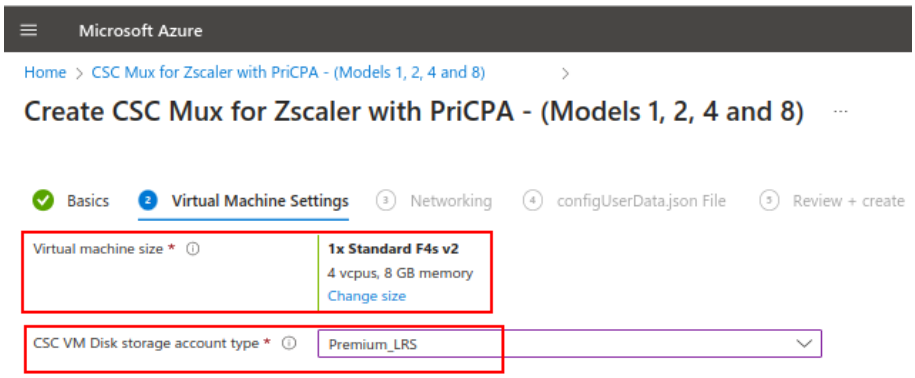
Confirm password * : ••••••••••••••••••••

< Previous | Next

Fill the values on "Basics"

1. Resource Group.
2. Location.
3. Single deployment or High Availability (2 x CSC).
4. Select Availability Zone for the first and second CSC. (Note: if the deployment is using Availability Sets, the menu will offer the corresponding options).
5. Put a name to the CSC VM. (Note: the ARM template will append a digit to the name. For example, if you deploy 2 x CSCs, the names will be <name>-1 and <name-2>)
6. For the username "cscadmin", choose to use Password or SSH key.

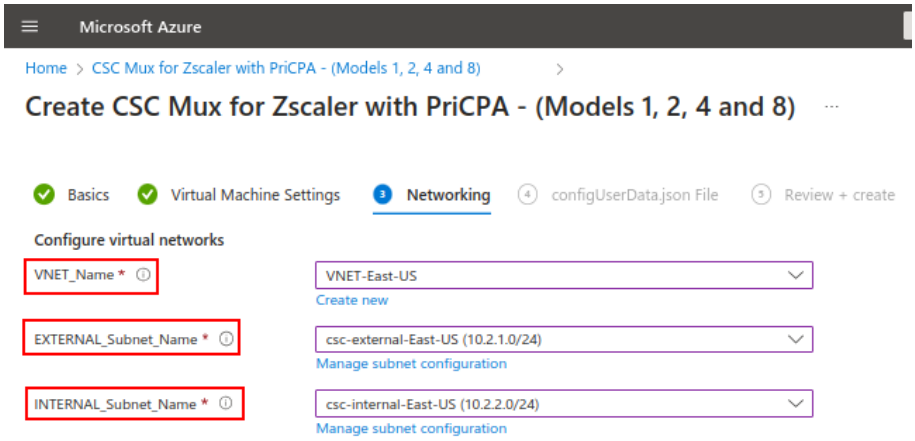
→ Click "Next".



→ Select the Virtual Machine size and Storage. We recommend using the Virtual Machine Size suggested.

Note: Microsoft created the VM Size "Standard Fx" family for Virtual Appliances. We recommend using the "Standard Fx" series for the CSC Mux 2 (Standard F2), 4 (Standard F4) and 8 (Standard F8). In the case of the CSC Mux 1, you can use Standard B1s or similar.

→ Click "Next"



-> Select the VNET, External and Internal Subnet for the CSC and click "Next".

(Optional) -> Paste configUserData.json file.

IMPORTANT: See Appendix B for format and examples of the configUserdata.json file.

Via configUserData.json file, you can pass values to parameters during the installation of the CSC. You can setup:

1. Zscaler Information: CloudName (zscloud, zscalerthree, zscalertwo; etc.), Nodes (autodiscovery or manual selection), vpnCredentials "domain". (Using "domain" the CSC automatically creates the FQDN (<vmName@domain> and Pre shared keys for IPsec.)
2. AWS SSM agent registration values.
3. DNS servers
4. Syslog servers and traffic log configuration.
5. Bypasses: Proxy Bypass PAC URL and Routed Bypass URL
6. PriCPA Local configuration values, Peers URL and Remote Management Networks.

7. SSH Restrictions via eth1 and wg0.
8. Admin Management: Enable csccli user and SSH Key.

configUserData.json (blank)

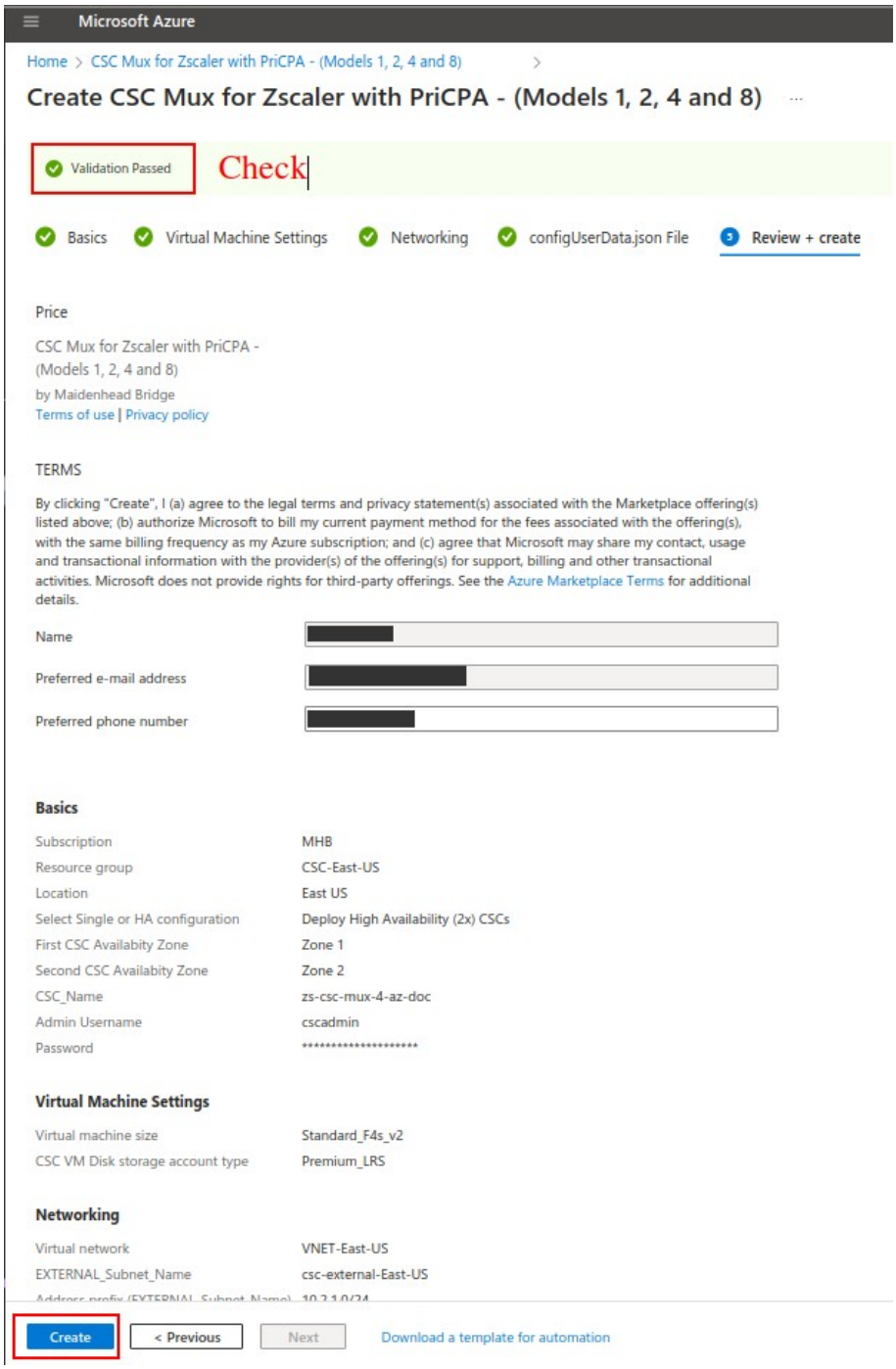
*The fields in **bold** are not configurable. So please, do not modify.*

configUserData.json

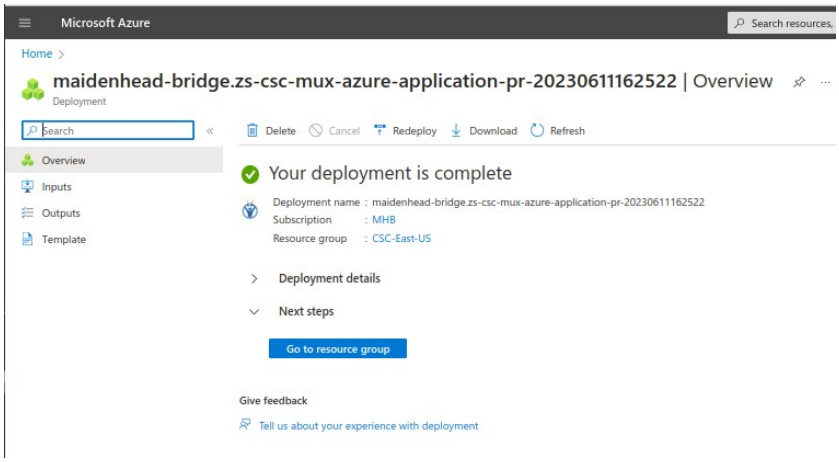
```
{
  "model": "csc-mux-zs-azure",
  "type": "configUserData",
  "version": "1.0",
  "zscalerInformation": {
    "cloudName": "",
    "vpnNodes": {
      "autoDiscovery": "yes",
      "primary": {
        "hostName": ""
      },
      "secondary": {
        "hostName": ""
      }
    },
    "vpnCredentials": {
      "domain": ""
    }
  },
  "awsSsmAgent": {
    "enable": "no",
    "activationCode": "",
    "activationId": "",
    "awsRegion": ""
  },
  "dns": {
    "useCloudDns": "yes",
    "primaryDnsIp": "",
    "secondaryDnsIp": ""
  },
  "syslog": {
    "enable": "no",
    "primaryServer": {
      "ip": "",
      "port": ""
    },
    "secondaryServer": {
      "ip": "",
      "port": ""
    },
    "trafficLogs": {
      "enable": "no"
    }
  },
  "bypasses": {
    "proxyBypass": {
      "pacUrl": ""
    },
```

```
      "routedBypass": {
        "jsonUrl": ""
      }
    },
    "priCPA": {
      "enable": "no",
      "nodeName": "",
      "location": "",
      "description": "",
      "publicUdpPort": "51280",
      "privateCirdIp": "",
      "persistentKeepAlive": "no",
      "peersJsonFileUrl": "",
      "remoteManagementNetworks": []
    },
    "sshRestrictions": {
      "eth1": {
        "enable": "no",
        "allowedNetworks": []
      },
      "wg0": {
        "enable": "no",
        "allowedNetworks": []
      }
    },
    "adminManagement": {
      "csccli": {
        "enable": "no",
        "sshPublicKey": ""
      }
    }
}
```

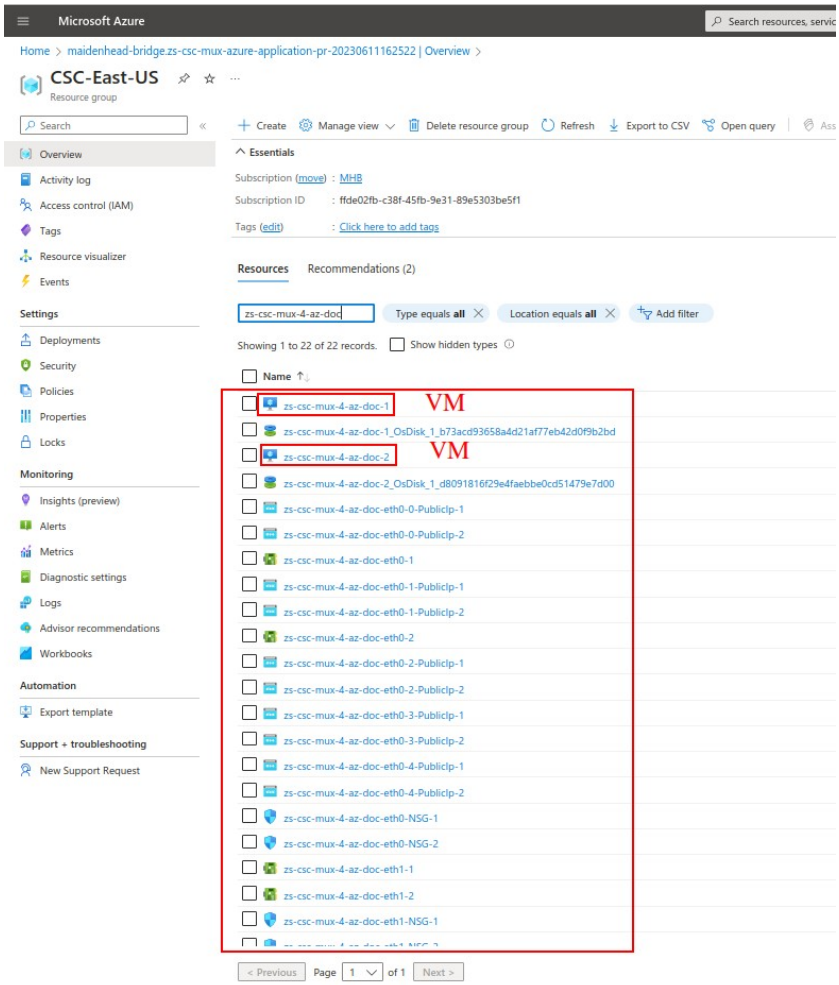
→ Click Next.



→ Check "Validation Passed" and click "Create". Wait up to "Your deployment is complete".



-> Click "Go to resource group" and you will see the components created.

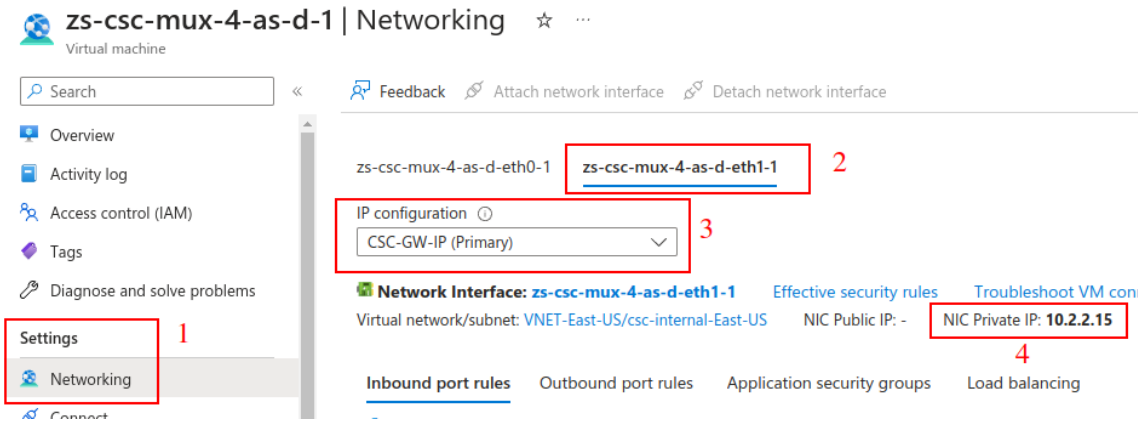


→ Done! Your CSCs Mux for Azure are deployed.

6 Accessing for first time to your CSC

6.1 SSH to the Admin Console using CSC GW IP

1. Go to your Azure Dashboard → Select the VM created → Networking → eth1 and check "NIC Private IP". (CSC-GW-IP (Primary))



2. In this example, "NIC Private IP" is: 10.2.2.15

3. From a machine inside the Virtual Network or via remotely via PriCPA, ssh the CSC using username "cscadmin" and key or password:

   ssh -i <keyname.pem> cscadmin@<eth1 Private IP>

   ssh cscadmin@<eth1 Private IP>

Important: Please, wait 2 minutes before to SSH the CSC to allow all processes to complete.

6.1.1 Initial Screen when using configUserData.json file

When passing Zscaler information via configUserData.json file, the CSC will automatically configure the Cloud, Nodes and VPN Credentials.

configUserData.json example:

```
"zscalerInformation": {
  "cloudName": "zscalerthree",
  "vpnNodes": {
    "autoDiscovery": "yes",
    "primary": {
      "hostName": ""
    },
    "secondary": {
      "hostName": ""
    }
  },
  "vpnCredentials": {
    "domain": "maidenheadbridge.com"
  }
},
```

Initial screen:

```
Maidenhead Bridge

CSC MUX 4 (1.6 Gbps) for Zscaler with PriCPA - Admin Console


Reminder: Configuration required on your Zscaler Console: VPN credentials and Location

 --> VPN Credentials creation: Go to > Administration > VPN Credentials > Add VPN Credential -> Select Authentication Type = FDQN and configure:

    User ID: zs-csc-mux-4-as-d-1@maidenheadbridge.com
    Pre-Shared Key: ztyckrj6viyAWWzkdWpl3gg0SM7QJjAm

 --> Location creation: Go to > Administration > Location > Add Location. Put your Location values and select 'VPN Credentials' created in the step before

Did you created the VPN Credentials and Location on the Zscaler console? Please, confirm.

1) Yes
2) No
Enter your choice:
```

You need to add the VPN credentials on your Zscaler console and to associate it with a Location.

6.1.2 Initial screen wihout using configUserData.json file

In this case, the initial screen presents the Wizard to configure manually the Zscaler Information.

```
*****IPsec tunnel information was never configured*****

Checking ZEN Databases...
This CSC has the latest version: 4.62


1) Configuration required on your Zscaler Console: VPN credentials and Location
 --> VPN Credentials creation: Go to > Administration > VPN Credentials > Add VPN Credential -> Select Authentication Type = FDQN and configure 'User ID' and 'Pre-Shared Key'
 --> Location creation: Go to > Administration > Location > Add Location. Put your Location values and select 'VPN Credentials' created in the step before
2) Run the Wizard on the CSC and Select Cloud, Nodes, input VPN Credentials, DNS Servers, Bypass PAC URL and Syslog Servers

Current Values Configured:
-------------------
ZSCALER INFORMATION
Zscaler Cloud:
Primary ZEN node: | Hostname: | IP:
Secondary ZEN node: | Hostname: | IP:
-------------------
CREDENTIALS INFORMATION
User ID: | Pre-Shared Key:

-------------------
Do you want to continue?

1) Yes
2) No
Enter your choice:
```

6.1.2.1 Running the initial wizard

Configuration required on your Zscaler Console: VPN credentials and Location

1. VPN Credentials creation: Go to > Administration > VPN Credentials > Add VPN Credential -> Select Authentication Type = FDQN and configure 'User ID' and 'Pre-Shared Key'
2. Location creation: Go to > Administration > Location > Add Location. Put your Location values and select 'VPN Credentials' created in the step before
3. Run the Wizard. Insert the values. Confirm and reboot.
4. Done!

Example:

1. Select Cloud and Nodes:

```
Do you want to change these values?

1) Yes
2) No
Enter your choice: 1

-----------------------------
Please, select your Cloud

1) zscalerthree
2) zscloud
3) zscalertwo
4) zscaler
5) zscalerone
6) zscalerbeta
7) zscalergov
8) Not in the list? Input Manually
9) Quit
Enter your choice: 1

-----------------------------
Please, select Manual or Auto Node Selection

1) Manual
2) Auto
3) Quit
Enter your choice: 2

-----------------------------
You have chosen the following:

Cloudname: zscalerthree
Primary node: AutoPrimary (vpn.zscalerthree.net)
Secondary Node: AutoSecondary (secondary.vpn.zscalerthree.net)
-------------------
```

2. Input VPN Credentials:

3. Confirm Vavlues

```
Please confirm this values:
-------------------
Cloudname: zscalerthree
Primary node: AutoPrimary (vpn.zscalerthree.net)
Secondary Node: AutoSecondary (secondary.vpn.zscalerthree.net)
-------------------
VPN Credentials
User ID: zs-csc-mux-4-as-d-2@maidenheadbridge.com | Pre-Shared Key: ONebnc9kXOKsH7x7n1i97StUzO2qNKHs
-------------------
Do you want to implement this values?

1) Yes
2) No
Enter your choice:
```

```
Email:zs-csc-mux-4-as-d-2@maidenheadbridge.com

Pre Shared Key:
Do you want to display the Pre Shared Key? (y/n)? n
```

4. The CSC will validate the configuration and will ask to confirm.

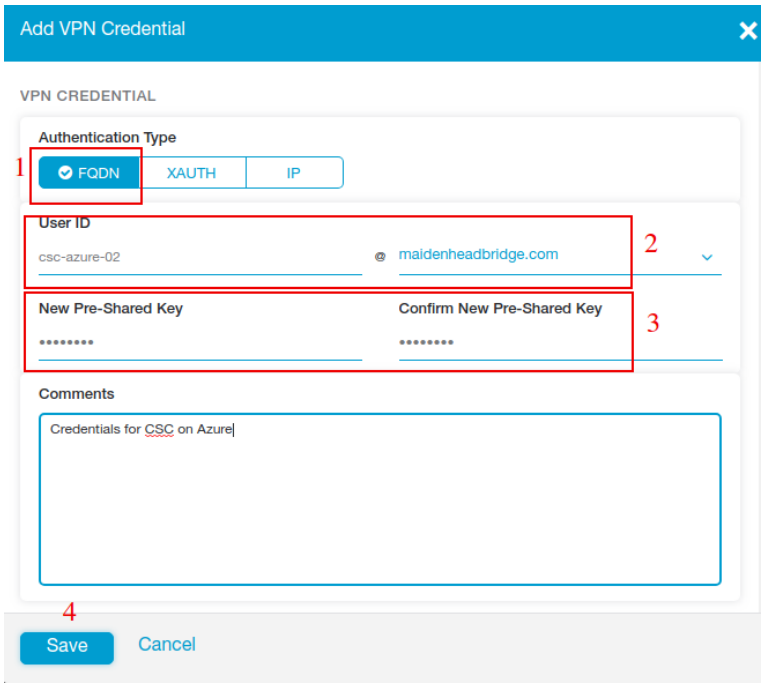
```
Validating Configuration

Your Cloud is: zscalerthree

Checking Node AutoPrimary hostname vpn.zscalerthree.net
Hostname vpn.zscalerthree.net has IP 165.225.8.35
Node AutoPrimary is Alive

Checking Node AutoSecondary hostname secondary.vpn.zscalerthree.net
Hostname secondary.vpn.zscalerthree.net has IP 165.225.38.52
Node AutoSecondary is Alive

Do you want to apply this values? (y/n)?:y

(MHB-CSC)(INFO) CSC:zs-csc-mux-4-as-d-2 connected to Zscaler Cloud: zscalerthree
(MHB-CSC)(INFO) Primary Zscaler Node using: AutoPrimary, hostname: vpn.zscalerthree.net (IP: 165.225.8.35) on CSC:zs-csc-mux-4-as-d-2
(MHB-CSC)(INFO) Secondary Zscaler Node using: AutoSecondary, hostname: secondary.vpn.zscalerthree.net (IP: 165.225.38.52) on CSC:zs-csc-mux-4-as-d-2
(MHB-CSC)(INFO) VPN Credentials using FQDN: zs-csc-mux-4-as-d-2@maidenheadbridge.com on CSC:zs-csc-mux-4-as-d-2

Rebooting after initial configuration

Connection to 10.2.2.18 closed by remote host.
Connection to 10.2.2.18 closed.
```

5. Done! The CSC is ready.

7 Zscaler console: create VPN Credentials and Location

7.1 VPN Credential creation.

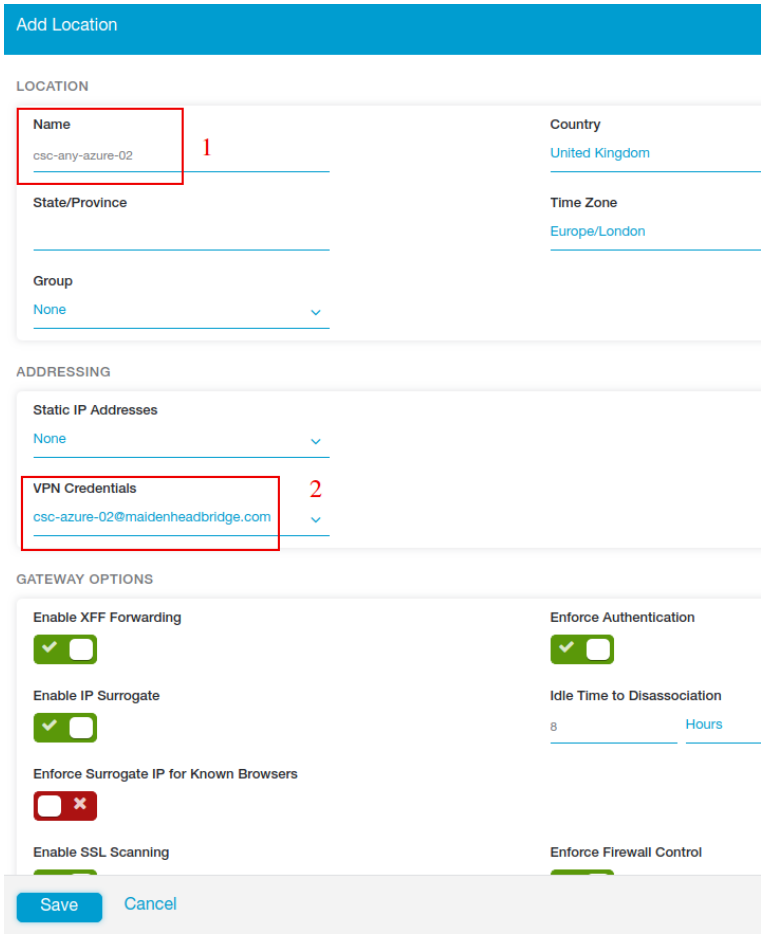
Go to > Administration > VPN Credentials > Add VPN Credential -> Select Authentication Type = FDQN and configure ‘User ID’ and ‘Pre-Shared Key’



Click “Save” and “Activation”

7.2 Create the Location on the Zscaler Console

Location creation: Go to > Administration > Location > Add Location. Put your Location values and select 'VPN Credentials' created in the step before.



Fill other values on the Location, click “Save” and “Activate”

7.3 Checking tunnel statuses on the CSC console.

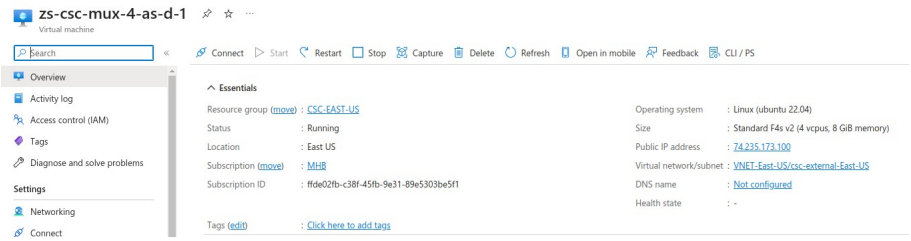
→ SSH the CSC and Run: "Show Configuration and Status". Check the Load Balancer and Tunnel information.

```
LOAD BALANCING INFORMATION
Last change: Tue 30 May 11:04:23 UTC 2023
(UP)   Ztun1 is active, using primary.
(UP)   Ztun2 is active, using primary.
(UP)   Ztun3 is active, using primary.
(UP)   Ztun4 is active, using primary.

IPSEC INFORMATION
Ztun1 connected to: AutoPrimary, IPsec uptime uptime: 6 minutes, since May 30 11:03:13 2023, Last Security Association: ESTABLISHED 6 minutes ago
Ztun2 connected to: AutoPrimary, IPsec uptime uptime: 6 minutes, since May 30 11:03:13 2023, Last Security Association: ESTABLISHED 6 minutes ago
Ztun3 connected to: AutoPrimary, IPsec uptime uptime: 6 minutes, since May 30 11:03:14 2023, Last Security Association: ESTABLISHED 6 minutes ago
Ztun4 connected to: AutoPrimary, IPsec uptime uptime: 6 minutes, since May 30 11:03:14 2023, Last Security Association: ESTABLISHED 6 minutes ago

CREDENTIALS INFORMATION
Username: zs-csc-mux-4-as-d-2@maidenheadbridge.com | PSK: Not shown. Please, read it from 'Configuration Wizards' Menu

http://ip.zscaler.com INFORMATION
Ztun1 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.68.253, via Public IP: 74.235.175.176
Ztun2 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.51.16, via Public IP: 20.163.185.99
Ztun3 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 165.225.9.19, via Public IP: 74.235.173.170
Ztun4 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.51.20, via Public IP: 20.163.185.151
```

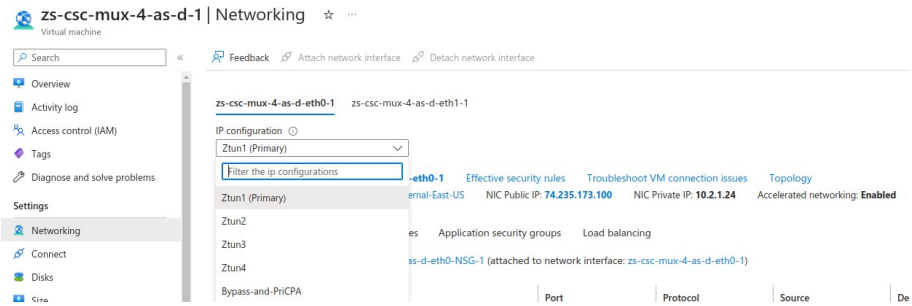
8 Resources creates by the ARM template

The following resources are created by the ARM template:

1. Virtual Machine



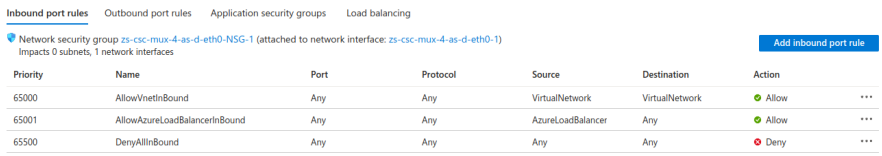
2. Interfaces External and Internal.



Depending the model, the ARM template creates 1, 2, 4 or 8 x Public IP (Ztunx) for the IPsec tunnels, and 1 x Public IP used by Bypass functionality and Private Access.

3. Security Group for External Interface. [1] [2]

   3.1. Inbound Rules



1 The CSC contains Firewall Rules on each interface that are more specific in some cases. For example, the CSC only allows reaching the configured Zscaler Nodes for IPsec traffic. Therefore, there is double protection: The Azure Security Group and the internal Firewall Rules of the CSC.

2 When using Private Access (PriCPA), the CSC automatically updates the internal FW rules and Security Groups to allow Peers to communicate with each other.

3.2. Outbound Rules

Inbound port rules | **Outbound port rules** | Application security groups | Load balancing

Network security group zs-csc-mux-4-as-d-eth0-NSG-1 (attached to network interface: zs-csc-mux-4-as-d-eth0-1)
Impacts 0 subnets, 1 network interfaces

Add outbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
4000	AllowPing	Any	ICMP	Any	Any	Allow
4010	AllowUDP500	500	UDP	Any	Any	Allow
4020	AllowUDP4500	4500	UDP	Any	Any	Allow
4030	AllowHTTP	80	TCP	Any	Any	Allow
4040	AllowHTTPS	443	TCP	Any	Any	Allow
4050	AllowPublicDNS	53	UDP	Any	Any	Allow
4060	AllowNTP	123	UDP	Any	Any	Allow
4070	DenyAllOutbound	Any	Any	Any	Any	Deny
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

4. Security Group for Internal Interface.

4.1. Inbound Rules

Inbound port rules | Outbound port rules | Application security groups | Load balancing

Network security group zs-csc-mux-4-as-d-eth1-NSG-1 (attached to network interface: zs-csc-mux-4-as-d-eth1-1)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
4000	AllowNet-10.0.0.0-8	Any	Any	10.0.0.0/8	Any	Allow
4010	AllowNet-172.16.0.0-12	Any	Any	172.16.0.0/12	Any	Allow
4020	AllowNet-192.168.0.0-16	Any	Any	192.168.0.0/16	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

4.2. Outbound Rules

Inbound port rules | **Outbound port rules** | Application security groups | Load balancing

Network security group zs-csc-mux-4-as-d-eth1-NSG-1 (attached to network interface: zs-csc-mux-4-as-d-eth1-1)
Impacts 0 subnets, 1 network interfaces

Add outbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

9 The Cloud Security Connector Admin Console:

The CSC's SSH Console simplifies administrative tasks showing what is essential to administrators for operation and troubleshooting.

When accessing the console via SSH (using the CSC GW IP), you will receive the Admin Console.

```
Maidenhead Bridge

CSC MUX 4 (1.6 Gbps) for Zscaler with PriCPA - Admin Console

VM Name : zs-csc-mux-4-as-d-1
Azure Region : eastus
Soft Version : 4.0

Please select an option by typing its number

Monitoring Tasks
1) Show Configuration and Status
2) Show Interfaces Traffic
3) Tcpdump, Traceroute and Latency Test
4) Speed Test (Experimental)

CSC Admin tasks
5) AWS SSM Agent (Register or De-Register)
6) Manage Administrators, Restrict SSH access and Radius Configuration.
7) Change Timezone

Proxy Bypass
8) View Current Proxy Bypass List
9) Configure Proxy Bypass List

Routed Bypass
10) View Current Routed Bypass List
11) Configure Routed Bypass List

System and Traffic Logs
12) View System Logs
13) Configure Syslog and Traffic Logs

Configuration Wizards
14) Configure Zscaler Nodes, VPN Credentials, DNS servers and SNMP.
15) Switch Zscaler Tunnels - Primary / Secondary.
16) High Availability configuration.

Private Cloud Private Access (PriCPA)
17) Show Configuration and Status PriCPA.
18) Configure PriCPA (Local and Peers Configuration).
19) Configure CSC Remote Management via PriCPA.

e) Exit

Selection:
```

The Main Sections are:

- **Monitoring Tasks:** To check configuraiton, statuses, real-time traffic, tcpdump, traceroute and speed..
- **CSC Admin Tasks:** To register the CSC for AWS management, manage administrators, restrict SSH, configure radius and change timezone.
- **Proxy Bypass:** View and configure Proxy Bypass (Layer 7) functionality.
- **Routed Bypass:** View and configure Routed Bypass (Layer 4) functionality
- **System and Traffic Logs:** Shows Systems logs, configure Syslog Servers and enable/disable traffic logs.
- **Configuration Wizards:** Configure Zscaler Nodes, VPN Credentials, DNS servers, SNMP, switch tunnels and configure High Availability.
- **Private Cloud Private Access (PriCPA):** Show configuration and statuses, create local coniguration, priCPA peers and add remote management networks,

9.1 Monitoring Tasks

9.1.1 Show Configuration and Status

```
Selection: 1

GENERAL INFORMATION
Name: zs-csc-mux-4-as-d-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_F4s_v2
CSC date: Sat 3 Jun 19:38:06 UTC 2023
Soft version: 4.0 | CSC Model: CSC MUX 4 (1.6 Gbps) for Zscaler with PriCPA
Azure Cloud: AzureCloud

INTERFACES INFORMATION
External: Tunnel IPs (eth0): 10.2.1.24-[25,26,27]/24 | Bypass Proxy Egress IP 10.2.1.28 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.15/24 | Network Gateway: 10.2.2.1

TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.2.2.16:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.2.2.17:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP

PUBLIC IP Address INFORMATION
IPsec tunnels Public IP: 74.235.173.100, 74.235.171.133, 74.235.171.132, 20.163.185.222
Bypass Public IP: 74.235.173.171

DNS INFORMATION
DNS Server (1): 1.1.1.1 is Alive
DNS Server (2): 8.8.8.8 is Alive

ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
Primary ZEN node: NewYorkIII | Hostname: nyc3-vpn.zscalerthree.net | IP: 165.225.38.52 is Alive
Secondary ZEN node: WashingtonDC_2 | Hostname: was1-2-vpn.zscalerthree.net | IP: 165.225.8.35 is Alive

LOAD BALANCING INFORMATION
Last change: Sat  3 Jun 19:24:54 UTC 2023
(UP)   Ztun1 is active, using primary.
(UP)   Ztun2 is active, using primary.
(UP)   Ztun3 is active, using primary.
(UP)   Ztun4 is active, using primary.

IPSEC INFORMATION
Ztun1 connected to: NewYorkIII, IPsec uptime uptime: 14 minutes, since Jun 03 19:23:43 2023, Last Security Association: ESTABLISHED 14 minutes ago
Ztun2 connected to: NewYorkIII, IPsec uptime uptime: 14 minutes, since Jun 03 19:23:43 2023, Last Security Association: ESTABLISHED 14 minutes ago
Ztun3 connected to: NewYorkIII, IPsec uptime uptime: 14 minutes, since Jun 03 19:23:43 2023, Last Security Association: ESTABLISHED 14 minutes ago
Ztun4 connected to: NewYorkIII, IPsec uptime uptime: 14 minutes, since Jun 03 19:23:43 2023, Last Security Association: ESTABLISHED 14 minutes ago

CREDENTIALS INFORMATION
Username: zs-csc-mux-4-as-d-1@maidenheadbridge.com | PSK: Not shown. Please, read it from 'Configuration Wizards' Menu

http://ip.zscaler.com INFORMATION
Ztun1 Node: New York III in the zscalerthree.net cloud. ZEN Instance IP: 165.225.39.99, via Public IP: 74.235.173.100
Ztun2 Node: New York III in the zscalerthree.net cloud. ZEN Instance IP: 165.225.39.126, via Public IP: 74.235.171.133
Ztun3 Node: New York III in the zscalerthree.net cloud. ZEN Instance IP: 165.225.221.4, via Public IP: 74.235.171.132
Ztun4 Node: New York III in the zscalerthree.net cloud. ZEN Instance IP: 165.225.39.121, via Public IP: 20.163.185.222

PROXY BYPASS
Proxy Bypass URL is: https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac
Proxy Bypass Rules configured via URL: 8
Proxy Bypass Egress Interface 10.2.1.28 can reach test page (https://ip.maidenheadbridge.com) via Public IP 74.235.173.171

ROUTED BYPASS
Routed Bypass URL is: https://mhb-csc-pac.s3.amazonaws.com/routedBypassRulesFile.json
Routed Bypass Rules configured via URL: 12
Routed Bypass URL https://mhb-csc-pac.s3.amazonaws.com/routedBypassRulesFile.json is reachable

AWS SSM AGENT
AWS SSM Agent is active (running) since Sat 2023-06-03 00:22:49 UTC; 19h ago
Registration values: {"ManagedInstanceID":"mi-055ab68d5af2fd09e","Region":"us-east-1"}

SYSLOG INFORMATION
Primary Syslog / SIEM (IP/TCP PORT): 172.19.0.5/5514 is Alive
Secondary Syslog / SIEM IP: Not configured
Traffic Logs (IP packets) are enabled.

HIGH AVAILABILITY Information
The HA service is:  active (running) since Sat 2023-06-03 00:22:49 UTC; 19h ago
Identity Type: SystemAssigned
Route to Zscaler using Next Hop: 10.2.2.18 of VM: zs-csc-mux-4-as-d-2 (the other CSC in the pair)
Current values configured are:
   Route/s (Qty)= 2
      Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
      Route 2: Zscaler-Global-GW (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
   Computer Name of other CSC in the pair: zs-csc-mux-4-as-d-2 (Resource Group=CSC-EAST-US)
Private Access Public IP= 74.235.173.101
```

9.1.1.1 GENERAL INFORMATION

This section contains general information about the instance:

```
GENERAL INFORMATION
Name: zs-csc-mux-4-as-d-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_F4s_v2
CSC date: Tue 30 May 16:26:50 UTC 2023
Soft version: 4.0 | CSC Model: CSC MUX 4 (1.6 Gbps) for Zscaler with PriCPA
Azure Cloud: AzureCloud
```

9.1.1.2 INTERFACES INFORMATION

This section contains the interfaces information:

```
INTERFACES INFORMATION
External: Tunnel IPs (eth0): 10.2.1.24-[25,26,27]/24 | Bypass Proxy Egress IP 10.2.1.28 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.15/24 | Network Gateway: 10.2.2.1
```

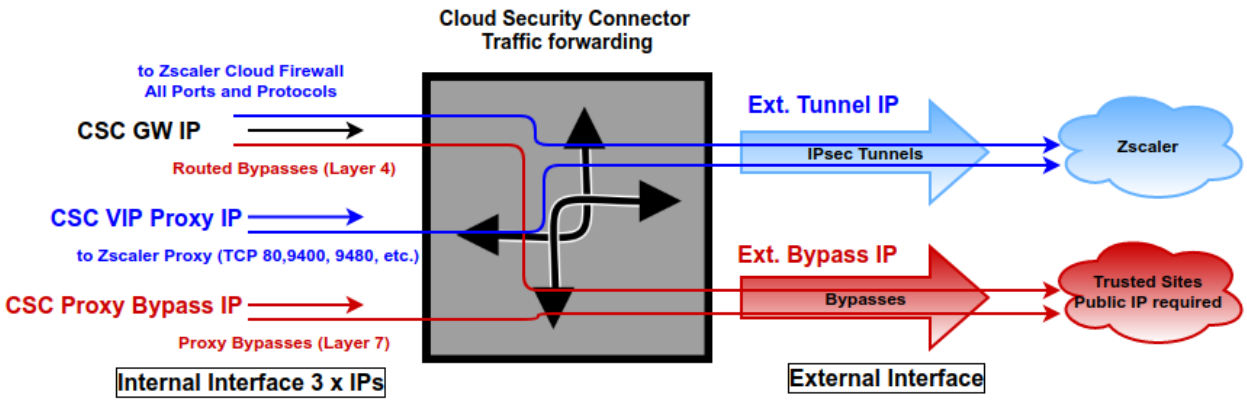
9.1.1.3 TRAFFIC REDIRECTION Options.

The section contains information about how to steer traffic to Zscaler.

```
TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.2.2.16:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.2.2.17:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP
```

The objective of the Cloud Security Connectors of Maidenhead Bridge is to provide a simple architecture, 100% proven that works when connecting to Zscaler.

Every member of the CSC family follows the principle of "three IPs" on the internal side:



- **CSC GW IP (*)**: To be used as Default Gateway for internal devices behind the CSC redirecting all ports and protocols to Zscaler when using Cloud Firewall. Traffic routed via CSC GW IP can be bypassed from Zscalerusing "Routed Bypasses" (Layer 4).
- **VIP Proxy:** This Virtual IP Proxy translates the packets directly to the Zscaler proxy. To be used when PAC files are implemented or explicit proxy.
- **Bypass Proxy IP:** The Bypass Proxy enables a simple way to do Layer 7 Bypasses to the Internet. To be used when PAC files are implemented.

(*) *On Azure Routes, the value to use as a "Next-Hop" is the CSC GW IP.*

Note:

The CSC Mux for Azure accepts the option using the Zscaler Global Proxies to send traffic to Zscaler Cloud and via the Proxy Bypass.

Your task is to route the Zscaler Global Proxies IPs via the CSC GW IP and to create a return statement on your PAC file like:

*Traffic to Zscaler → return "PROXY 185.46.212.88:**80**"; (you can use port 9400 as well)*
*Traffic via Bypass Proxy → return "PROXY 185.46.212.88:**3128**";*

List of Zscaler Global Proxies:

185.46.212.88	*185.46.212.89*	*185.46.212.90*	*185.46.212.91*
185.46.212.92	*185.46.212.93*	*185.46.212.97*	*185.46.212.98*

Important: *Please, see Chapter 10 for detailed information about traffic redirection (with examples)*

9.1.1.4 PUBLIC IP Address INFORMATION

This section shows the Public IP used to initiate the tunnels to Zscaler and the Public IP used for the Bypass Proxy functionality.

```
PUBLIC IP Address INFORMATION
IPsec tunnels Public IP: 74.235.173.100, 74.235.171.133, 74.235.171.132, 20.163.185.222
Bypass Public IP: 74.235.173.101
```

9.1.1.5 DNS INFORMATION

This section displays the DNS information. You can use the default DNS server from Azure and Google or set up your DNS servers.

```
DNS INFORMATION
Using Azure DNS (168.63.129.16) and Google DNS (8.8.8.8, 8.8.4.4)
```

9.1.1.6 ZSCALER INFORMATION

This section shows the IPsec tunnels information and if the Zscaler's nodes are reachable.

```
ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
Primary ZEN node: AutoPrimary | Hostname: vpn.zscalerthree.net | IP: 165.225.48.10 is Alive
Secondary ZEN node: AutoSecondary | Hostname: secondary.vpn.zscalerthree.net | IP: 165.225.38.52 is Alive
```

9.1.1.7 LOAD BALANCING INFORMATION

The CSC Mux has the capacity to aggregate multiple IPsec tunnels and has a Load Balancer that distributes the load evenly among each tunnel. This section shows the status of the Load Balancer of a CSC Mux 4.

```
LOAD BALANCING INFORMATION
Last change: Thu 25 May 02:08:59 UTC 2023
(UP)   Ztun1 is active, using primary.
(UP)   Ztun2 is active, using primary.
(UP)   Ztun3 is active, using primary.
(UP)   Ztun4 is active, using primary.
```

9.1.1.8 IPSEC INFORMATION

This section shows the status of each IPsec tunnel.

```
IPSEC INFORMATION
Ztun1 connected to: AutoPrimary, IPsec uptime uptime: 5 days, since May 25 02:08:01 2023, Last Security Association: ESTABLISHED 105 minutes ago
Ztun2 connected to: AutoPrimary, IPsec uptime uptime: 5 days, since May 25 02:08:01 2023, Last Security Association: ESTABLISHED 107 minutes ago
Ztun3 connected to: AutoPrimary, IPsec uptime uptime: 5 days, since May 25 02:08:01 2023, Last Security Association: ESTABLISHED 2 hours ago
Ztun4 connected to: AutoPrimary, IPsec uptime uptime: 5 days, since May 25 02:08:02 2023, Last Security Association: ESTABLISHED 2 hours ago
```

9.1.1.9 CREDENTIALS INFORMATION

This section shows the User ID in use:

```
CREDENTIALS INFORMATION
Username: zs-csc-mux-4-as-d-1@maidenheadbridge.com | PSK: Not shown. Please, read it from 'Configuration Wizards' Menu
```

9.1.1.10 http://ip.zscaler.com INFORMATION

Zscaler recommends checking the page http://ip.zscaler.com to validate that you are using Zscaler and see Zscaler Node connected, Cloud and IP address. The CSC does this test for you on each tunnel.

```
http://ip.zscaler.com INFORMATION
Ztun1 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.49.17, via Public IP: 74.235.173.100
Ztun2 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.53.25, via Public IP: 74.235.171.133
Ztun3 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.53.23, via Public IP: 74.235.171.132
Ztun4 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.49.4, via Public IP: 20.163.185.222
```

9.1.1.11 PROXY BYPASS

This sections shows the Proxy Bypass PAC URL, validates if the Proxy Bypass can access internet directly going to https://ip.maidenheadbridge.com and shows the amounts of proxy bypass rules configured.

```
PROXY BYPASS
Proxy Bypass URL is: https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac
Proxy Bypass Rules configured via URL: 8
Proxy Bypass Egress Interface 10.2.1.28 can reach test page (https://ip.maidenheadbridge.com) via Public IP 74.235.173.101
```

9.1.1.12 ROUTED BYPASS

This section shows the configuration of Routed Bypasses and check if the routed bypass URL is reachable.

```
ROUTED BYPASS
Routed Bypass URL is: https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json
Routed Bypass Rules configured via URL: 12
Routed Bypass URL https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json is reachable
```

9.1.1.13 AWS SSM AGENT

This section shows the status of the AWS SSM Agent.

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Thu 2023-05-25 01:51:37 UTC; 5 days ago
Registration values: {"ManagedInstanceID":"mi-055ab68d5af2fd09e","Region":"us-east-1"}
```

9.1.1.14 SYSLOG INFORMATION

When configured, this section will show the IP/s and TCP port of your Syslog/SIEM server.

```
SYSLOG INFORMATION
Primary Syslog / SIEM (IP/TCP PORT): 172.19.0.5/5514 is Alive
Secondary Syslog / SIEM IP: Not configured
Traffic Logs (IP packets) are enabled.
```

All CSC's logs are tagged with (MHB-CSC)(<action>). The values of <action> are:

SystemLogs:

- UP
- DOWN
- INFO
- ALERT
- ERROR

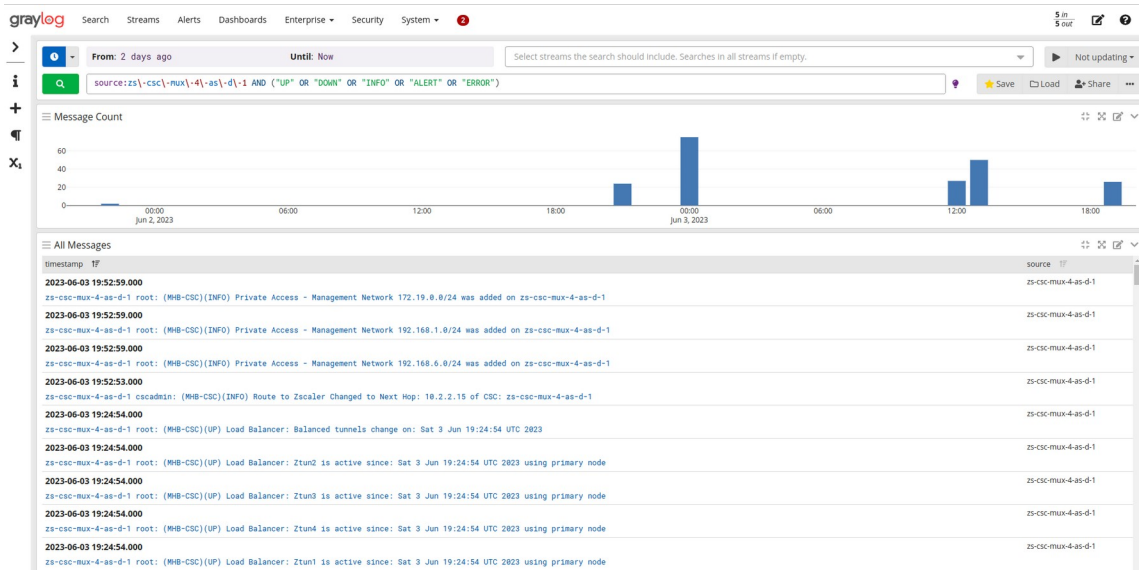
Traffic Logs:

- ALLOW
- BLOCK

9.1.1.14.1 System Logs example:

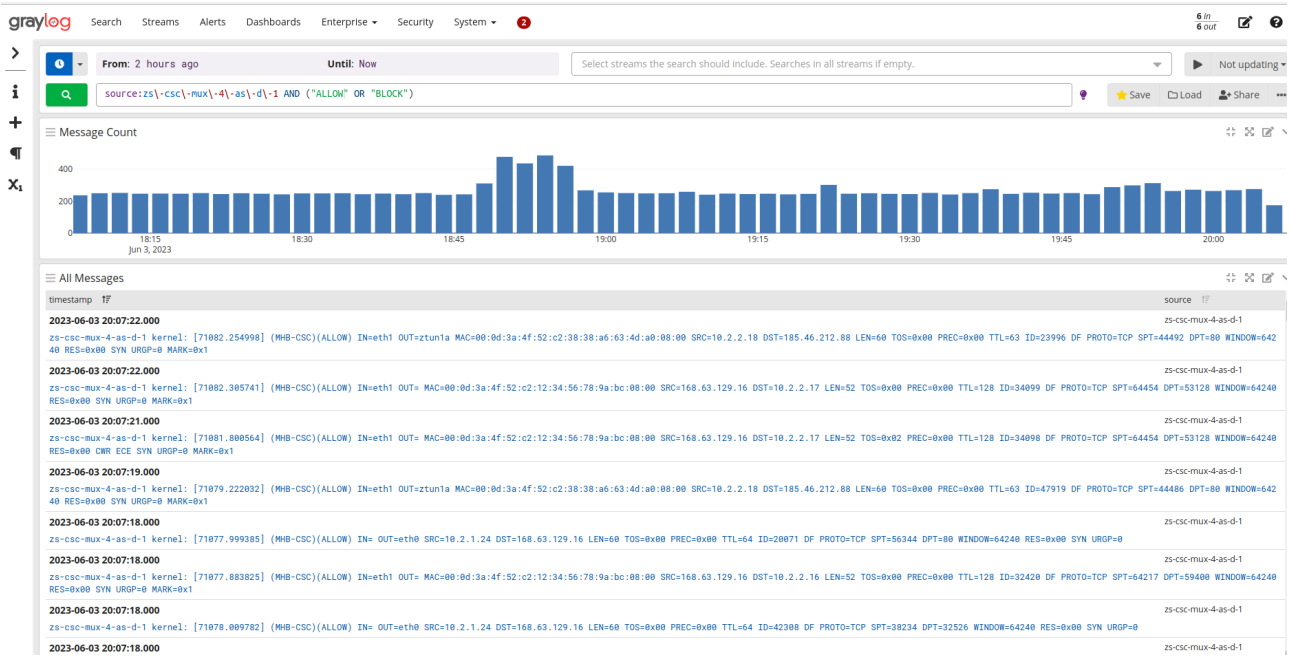
To obtain your System Logs, you can search by CSC name plus the following TAG. For example:

Using GrayLog Server: source:zs\-csc\-mux\-4\-as\-d\-1 AND ("UP" OR "DOWN" OR "INFO" OR "ALERT" OR "ERROR")



9.1.1.14.2 Traffic Logs example:

Using GrayLog Server: source:zs\-csc\-mux\-4\-as\-d\-1 AND ("ALLOW" OR "BLOCK")



9.1.1.15 HIGH AVAILABILITY Information

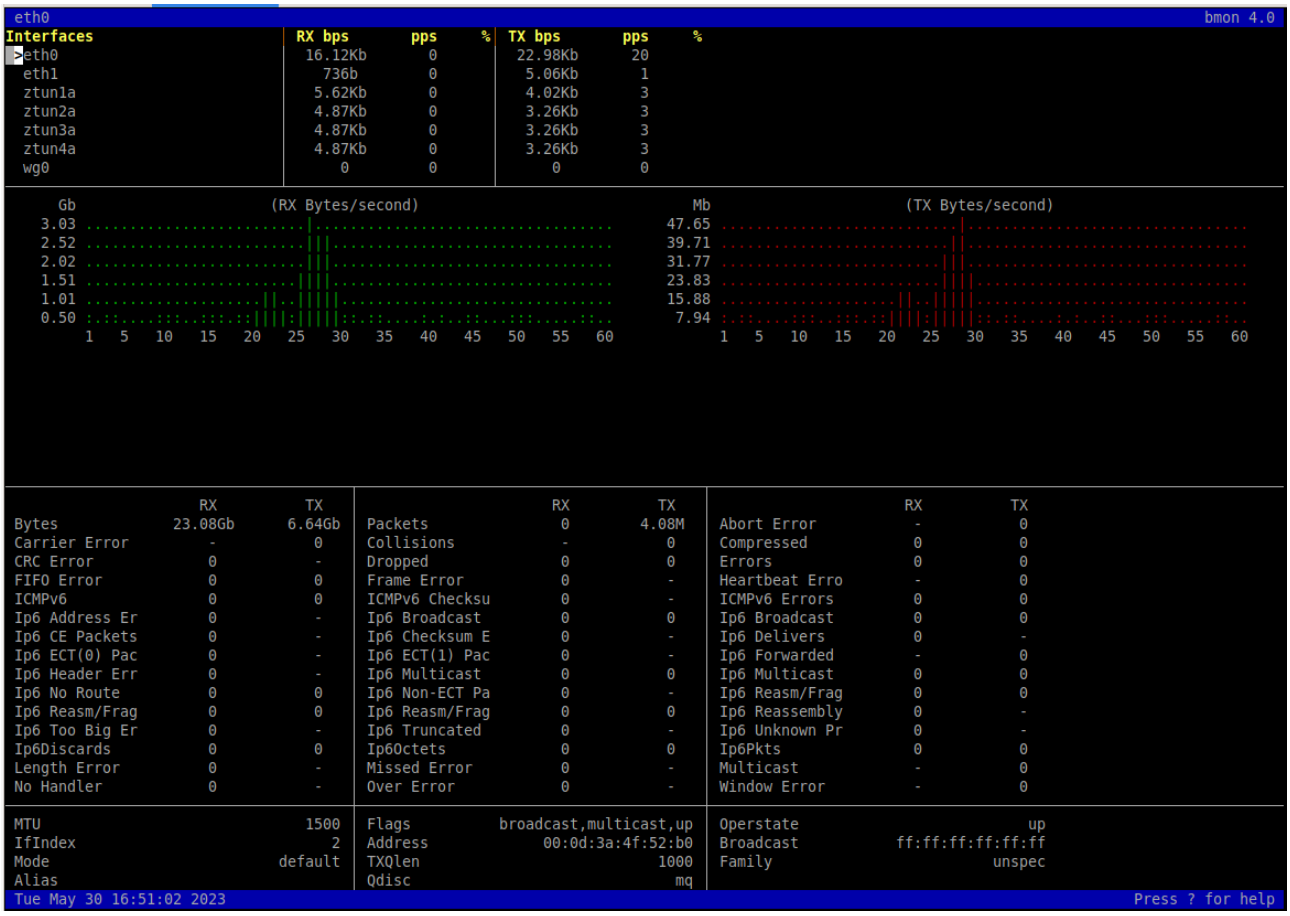
This section show all the information when the CSC Mux is configured on HA:

```
HIGH AVAILABILITY Information
The HA service is:  active (running) since Sat 2023-06-03 00:22:49 UTC; 19h ago
Identity Type: SystemAssigned
Route to Zscaler using Next Hop: 10.2.2.15 of VM: zs-csc-mux-4-as-d-1 (this CSC)
Current values configured are:
   Route/s (Qty)= 2
      Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
      Route 2: Zscaler-Global-GW (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
   Computer Name of other CSC in the pair: zs-csc-mux-4-as-d-2 (Resource Group=CSC-EAST-US)
Private Access Public IP= 74.235.173.101
```

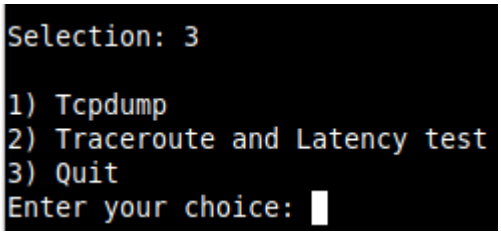
- If HA service is active.
- The Identity Type in use.
- The current “Next Hop” active for all "Route/s" configured.
- Amount of Routes configured.
- The Route names.
- Which is the VM Name of other CSC on the HA pair.
- Private Access Public IP.

9.1.2 Show Interfaces Traffic

Use this section to see the traffic in real time.



9.1.3 Tcpdump, Traceroute and Latency Test



9.1.3.1 Tcpdump

The objective of this test is to have detailed visibility of any type of traffic via any interface.

```
This menu helps to run the 'tcpdump' command on the Cloud Security Connector.
You can inspect packets per Interface, IP, Network, Protocol and Port.
After following the menu, you will see the resulting 'tcpdump' command. If you want to run more complex tcpdump commands, please log in to the CSC using 'csccli' username.

Recommendations about Interfaces:
a) Use Interface eth1 (internal CSC) to validate the traffic end-to-end between your devices. We recommend starting always checking eth1.
b) Use Interface eth0 (external CSC) to validate Bypasses, Tunnel traffic and communications between CSCs using PriCPA.
c) Use Interface PriCPA (wg0) to validate PriCPA Rules. For example, you can see the traffic for a particular remote destination arriving at eth1 (internal CSC) but not on PriCPA (wg0). If this happens, your Rule is blocking traffic
 to the remote destination, and you need to correct the Rule.
d) Use 'All Interfaces' to check the ingress interface and egress interface.

Last Command: sudo timeout 30 tcpdump -n -c 10 -i eth1 tcp port 80

Do you want to continue?

1) Yes - Repeat Last Command
2) Yes - New Command
3) No
Enter your choice:
```

You can repeat the last command or running a new command. Example running a new command:

- Select the options:

```
Enter your choice: 2

Please select the Interface.

1) Internal(eth1)
2) External(eth0)
3) priCPA(wg0)
4) All Interfaces
5) Quit
Enter your choice: 1

Please select the Host or Net or Specific Source/Destination Pair or Any.

1) Host
2) Net
3) Source/Destination IPs
4) Any
5) Quit
Enter your choice: 1
Host (IP): 10.2.9.4

Please select the Protocol (TCP/UDP/ICMP) or Any.

1) TCP
2) UDP
3) ICMP
4) Any
5) Quit
Enter your choice: 1
Please, input Port Number (1 to 65535) or '0' for Any: 22

By default, this script stops after 10 packets or 30 seconds.
These values work in most troubleshooting scenarios.
You can increase these values here up to 100 packets or 300 seconds maximum.

Do you want to change default values?

1) Yes
2) No
3) Quit
Enter your choice: 2
```

- The test will show the resulting tcpdump command and will show the traffic captured.

```
Enter your choice: 2

COMMAND: sudo timeout 30 tcpdump -n -l -c 10 -i eth1 host 10.2.9.4 and tcp port 22

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:48:12.837271 IP 10.2.2.15.22 > 10.2.9.4.33304: Flags [P.], seq 3253839517:3253839705, ack 2539124923, win 501, options [nop,nop,TS val 4053139764 ecr 3660682945], length 188
17:48:12.838167 IP 10.2.9.4.33304 > 10.2.2.15.22: Flags [.], ack 188, win 501, options [nop,nop,TS val 3660682977 ecr 4053139764], length 0
17:48:12.931384 IP 10.2.2.15.22 > 10.2.9.4.33304: Flags [P.], seq 188:544, ack 1, win 501, options [nop,nop,TS val 4053139858 ecr 3660682977], length 356
17:48:12.932277 IP 10.2.9.4.33304 > 10.2.2.15.22: Flags [.], ack 544, win 501, options [nop,nop,TS val 3660683071 ecr 4053139858], length 0
17:48:13.021197 IP 10.2.2.15.22 > 10.2.9.4.33304: Flags [P.], seq 544:876, ack 1, win 501, options [nop,nop,TS val 4053139948 ecr 3660683071], length 332
17:48:13.022134 IP 10.2.9.4.33304 > 10.2.2.15.22: Flags [.], ack 876, win 501, options [nop,nop,TS val 3660683161 ecr 4053139948], length 0
17:48:13.125393 IP 10.2.2.15.22 > 10.2.9.4.33304: Flags [P.], seq 876:1208, ack 1, win 501, options [nop,nop,TS val 4053140052 ecr 3660683161], length 332
17:48:13.126340 IP 10.2.9.4.33304 > 10.2.2.15.22: Flags [.], ack 1208, win 501, options [nop,nop,TS val 3660683265 ecr 4053140052], length 0
17:48:13.229322 IP 10.2.2.15.22 > 10.2.9.4.33304: Flags [P.], seq 1208:1540, ack 1, win 501, options [nop,nop,TS val 4053140156 ecr 3660683265], length 332
17:48:13.231098 IP 10.2.9.4.33304 > 10.2.2.15.22: Flags [.], ack 1540, win 501, options [nop,nop,TS val 3660683370 ecr 4053140156], length 0
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

9.1.3.2 Traceroute and Latency Test

This test can validate the quality of the Internet path between your location and Zscaler Nodes. You can run it with tunnels down or up. When the tunnels are up, it does a “Reverse Path” test from your active Zscaler node to your location. This test is beneficial to check if there is any packet loss at some point.

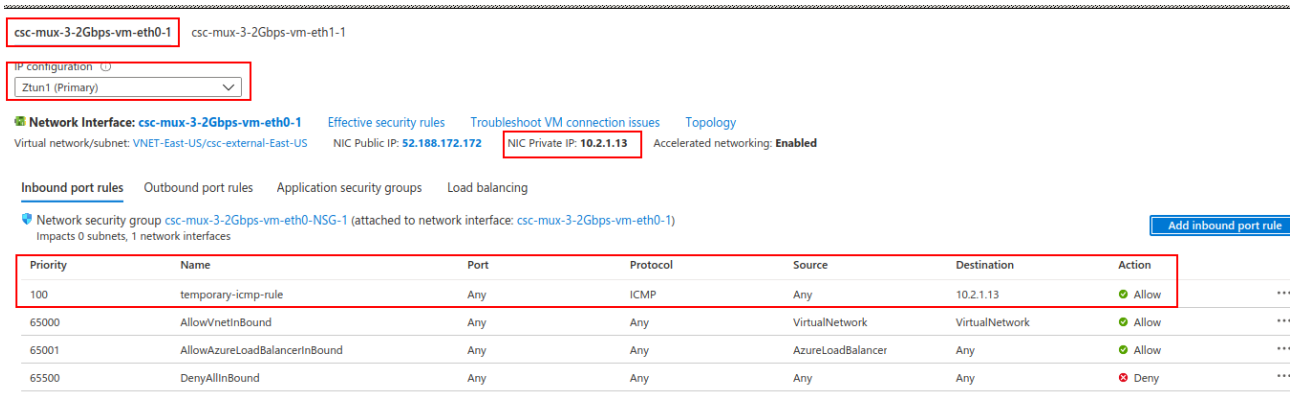
- **IMPORTANT**: It is required to allow ICMP Time exceeded (type 11) on the Inbound rule of the Security Group of eth0 to destination IP: x.x.x.x (←This IP is Ztun1)

  Without this security rule added, you will not able to see the results of middle hops.

- When the Ztun1 is UP, a Reverse Path test from the active ZEN to Ztun1 Public IP is performed

- Max Hops is equal 30. This test can take a while.

Example of temporary ICMP Rule required:



```
My TraceRoute (MTR) Test Report
This test does 10 probes DIRECT to Primary / Secondary ZEN and a Reverse test via Ztun1 to Ztun1 Public IP
Notes:
- IMPORTANT: It is required to allow ICMP Time exceeded (type 11) on the Inbound rule of the Security Group of eth0 to destination IP: 10.2.1.24
  Without this security rule added, you will not able to see the results of middle hops.
- When the Ztun1 is UP, a Reverse Path test from the active ZEN to Ztun1 Public IP is performed
- Max Hops is equal 30. This test can take a while

Testing Primary ZEN: AutoPrimary : vpn.zscalerthree.net > 165.225.48.10
Start: 2023-05-30T17:41:51+0000
HOST: zs-csc-mux-4-as-d-1                                          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  2. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  3. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  4. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  5. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  6. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  7. AS8075   ae34-0.ash-96cbe-1a.ntwk.msn.net (104.44.233.19)     0.0%    10    1.0   3.1   1.0  13.7   4.0
  8. AS???    eqix-was1-r2.zscaler9.net (206.126.237.212)          10.0%    10  299.2 291.3 280.3 309.1   8.8
  9. AS22616  165.225.254.30                                       50.0%    10  287.0 295.0 287.0 302.2   5.6
 10. AS22616  165.225.48.10                                         0.0%    10    1.4   1.7   1.4   3.2   0.5

Testing Secondary ZEN: AutoSecondary : secondary.vpn.zscalerthree.net > 165.225.38.52
Start: 2023-05-30T17:42:11+0000
HOST: zs-csc-mux-4-as-d-1                                          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  2. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  3. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  4. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  5. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  6. AS???    ???                                                  100.0    10    0.0   0.0   0.0   0.0   0.0
  7. AS8075   be-144-0.ibr03.bl7.ntwk.msn.net (104.44.32.32)       0.0%    10    7.9   7.4   6.9   8.4   0.4
  8. AS8075   be-10-0.ibr01.ewr30.ntwk.msn.net (104.44.16.8)       0.0%    10    6.9   7.0   6.8   7.6   0.2
  9. AS8075   ae25-0.ear04.ewr30.ntwk.msn.net (104.44.33.204)      0.0%    10    6.8   8.6   6.3  12.4   2.3
 10. AS8075   ae27-0.ier01.teb31.ntwk.msn.net (104.44.239.19)      0.0%    10    6.5   6.6   6.3   7.0   0.2
 11. AS22616  165.225.250.79                                        0.0%    10    6.5   6.5   6.2   6.7   0.2
 12. AS22616  165.225.38.52                                         0.0%    10    7.2   7.0   6.6   8.0   0.4


Reverse path from: AutoPrimary to your Public IP: 74.235.173.100
Start: 2023-05-30T17:42:32+0000
HOST: zs-csc-mux-4-as-d-1          Loss%   Snt   Last   Avg  Best  Wrst StDev
  1. AS???    ???                  100.0    10    0.0   0.0   0.0   0.0   0.0
  2. AS22616  136.226.49.23         0.0%    10    2.3   2.2   1.7   3.0   0.4
  3. AS22616  136.226.48.2         50.0%    10  304.3 299.4 287.1 304.3   7.0
  4. AS22616  165.225.250.62       90.0%    10    2.2   2.2   2.2   2.2   0.0
  5. AS8075   104.44.51.17          0.0%    10    4.9  11.4   2.3  64.3  18.8
  6. AS???    ???                  100.0    10    0.0   0.0   0.0   0.0   0.0
```

9.1.4 SPEED TEST

This test is experimental because we use third-party tools (speedtest.net), but it works fine in most cases. Here the result using a CSC Mux 4.

```
Selection: 4

SPEED TEST
This is experimental. We are using third party tools. (Speedtest.net)
Results can be inaccurate or none. The test takes a while


Aggregated Bandwidth Download:  2472.33 Mbps
```

> ***Note:*** *At the moment of writing this documentation, Zscaler provides 400 Mbps per IPsec tunnel. When running a single test without production traffic, it is possible to obtain aggregates speeds of more than 1.6 Gbps, but with production traffic, Zscaler will rate limit the traffic to 400 Mbps.*

9.2 CSC Admin Tasks

```
CSC Admin tasks
5) AWS SSM Agent (Register or De-Register)
6) Manage Administrators, Restrict SSH access and Radius Configuration.
7) Change Timezone
```

9.2.1 AWS SSM Agent (Register or De-Register)

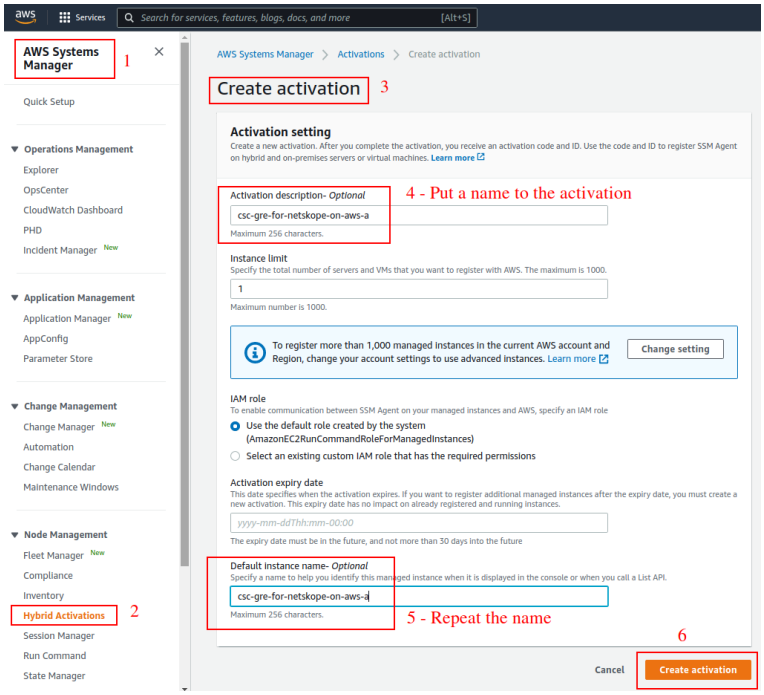
The CSC AWS has installed the AWS SSM Agent that allows you to check remotely the status of the CSC and “Run Commands” using AWS Systems Manager. You can manage all CSCs models[3] using AWS Systems Manager.

***Note**: You can learn more about “Run Commands” on Appendix B*

Steps to create a "Hybrid Activation" and "Register the CSC".

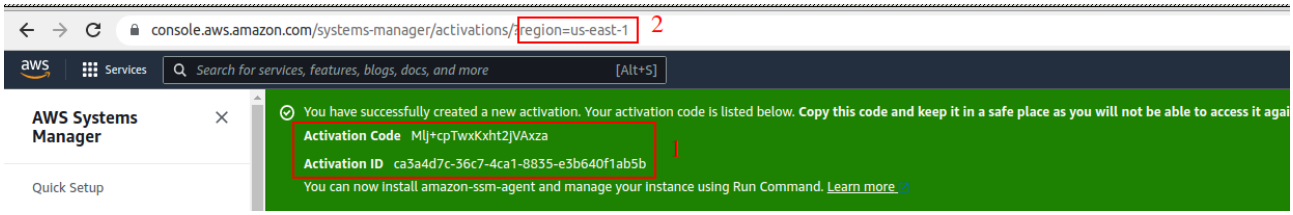
9.2.1.1 Create a "Hybrid Activation" from AWS console.

On your AWS Console, go to Services → Systems Manager → Node Management → Hybrid Activations and click "Create". Fill the values on shown below:



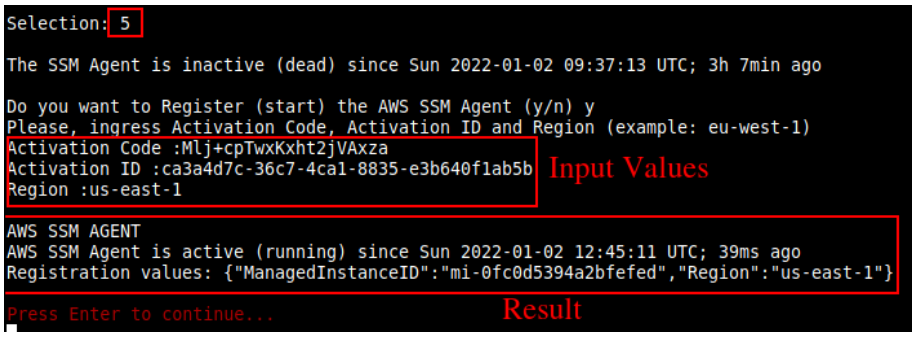
3 For Vmware, Hyper-V, KVM, Azure, Gcloud and AWS.

→ Click "Create Activation"

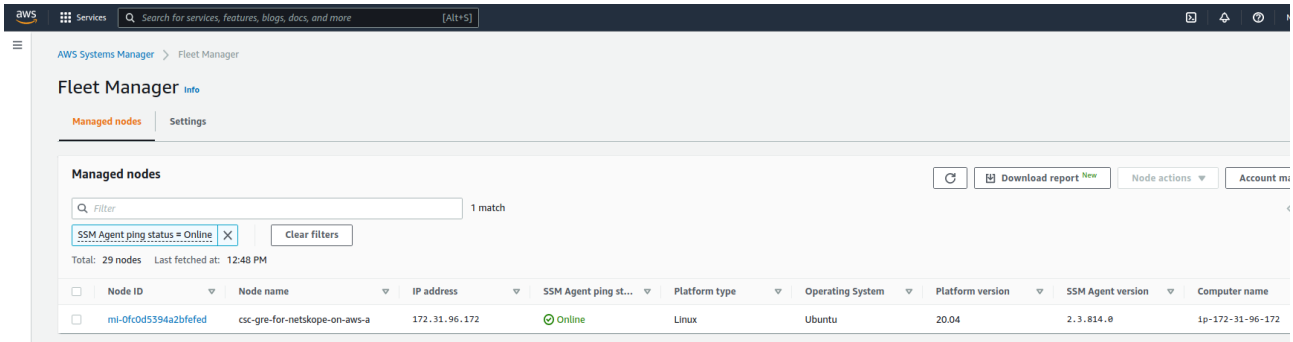


The vaules of Actication Code, Activation ID and Region are required to register the CSC. Keep this values on a safe place.

9.2.1.2 Register the CSC



9.2.1.3 View the Registered CSC on AWS Systems Manager



9.2.2 Manage Administrators, Restrict SSH access and Radius Configuration

IMPORTANT: *This section can be accessed only by the "cscadmin" user.*

```
Selection: 6

Please, select the task to do:

1) Manage Administrators: cscadmin and csccli
2) Restrict SSH Access
3) Radius Configuration
4) Quit
Enter your choice:
```

9.2.2.1 Manage Administrators: cscadmin and csccli

The CSC Mux for AZure has 2 users configured: cscadmin (for SSH Administrator Console Access), csccli (standard user, disabled by default.).

From this menu, you can edit the SSH Keys or Password.

```
Selection: 6

Please, select the Administrator: 'cscadmin' or 'csccli'

1) cscadmin
2) csccli
3) Quit
Enter your choice:
```

Note: *the user "cscadmin" cannot be disabled.*

9.2.2.1.1 "cscadmin" settings

```
Please, select the Administrator: 'cscadmin' or 'csccli'

1) cscadmin
2) csccli
3) Quit
Enter your choice: 1

Please, select the task to do:

1) Change Password
2) Manage SSH Keys
3) Quit
Enter your choice:
```

9.2.2.1.2 "csccli" settings

Note: the "csccli" user allows console access to the CSC. If you are managing the CSC using Rundeck, or Ansible, you will need to enabled the "csccli" user and to setup the SSH Key.

```
1) cscadmin
2) csccli
3) Quit
Enter your choice: 2

User 'csccli' is not enabled.

Do you want to enable user 'csccli'?

1) Yes
2) No
Enter your choice: 1

User 'csccli' was enabled via console.

Please, input a SSH Key for user 'csccli'

This Menu allows to add/delete the SSH Public keys using Nano editor.

To save, press CTRL+S and to exit Nano, press CTRL+X

Do you want to continue?

1) Edit SSH Keys
2) Quit
Enter your choice:
```

9.2.2.1.3 Managing the SSH Key of a User

You can add/remove keys for a User using "nano editor" when selecting the user from the previous menu.

9.2.2.2 Restrict SSH Access

This functionality allows adminsitrators to restrict SSH access to the CSC. You can setup restrictions for the Internal (eth1) and the PriCPA (wg0) interface. SSH to external (eth0) interface is always blocked.

IMPORTANT (1): DEFAULT VALUES.
-> Internal Interface (eth1): SSH the CSC to CSC GW IP (<IP>) is allowed from any Host or Subnet.
-> External Interface (eth0): SSH the CSC to any eth0 IP is permanently blocked and cannot be changed.
-> PriCPA Interface (wg0): SSH the CSC to wg0 IP (<IP>) is allowed from any other PriCPA node that belongs to the PriCPA Subnet. (<Subnet>/<Bitmask>)

IMPORTANT (2): If the Host or Subnet is reachable via PriCPA interface and not Internal Interface eth1, you must add these Hosts or Subnets as Management Networks on PriCPA configuration.

Example of configuration:

```
1) Manage Administrators: cscadmin and csccli
2) Restrict SSH Access
3) Radius Configuration
4) Quit
Enter your choice: 2

This wizard allows restricting the SSH access to the CSC.

IMPORTANT (1): DEFAULT VALUES.
     -> Internal Interface (eth1): SSH the CSC to CSC GW IP (10.2.2.15) is allowed from any Host or Subnet.
     -> External Interface (eth0): SSH the CSC to any eth0 IP is permanently blocked and cannot be changed.
     -> PriCPA Interface (wg0): SSH the CSC to wg0 IP (192.168.7.16) is allowed from any other PriCPA node that belongs to the PriCPA Subnet. (192.168.7.0/24)

WARNING! You can isolate this node if the configuration is wrong.
Be careful with these settings. We recommend being precise with the Host or Subnet configured here.
Subnet Prefixes less than /8 are not accepted.

IMPORTANT (2): If the Host or Subnet is reachable via PriCPA interface and not Internal Interface eth1, you must add these Hosts or Subnets as Management Networks on PriCPA configuration.

Current values configured are:

SSH to CSC GW IP (10.2.2.15) is allowed only from: 10.2.0.0/16 172.19.0.0/24 192.168.1.0/24 192.168.6.0/24
SSH to PriCPA IP (192.168.7.16) is allowed only from: 10.2.0.0/16 172.19.0.0/24 192.168.1.0/24

Do you want to change values?

1) Yes
2) No
3) Reset to Default
Enter your choice:
```

9.2.2.3 Radius Configuration

This functionality enables Radius Authentication for users accessing the Admin Console. The configuration requires the Radius Server IP and Secret. Optionally, you can add a secondary radius server as backup.

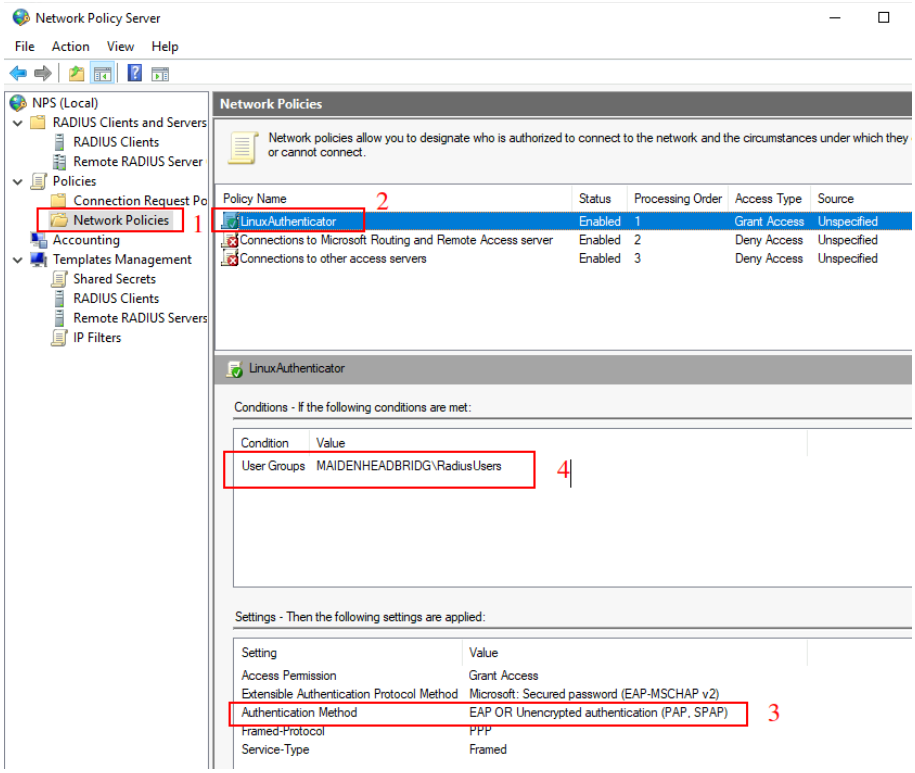
-> Configuration on the CSC: Add Radius Server and User:

```
Selection: 6

Please, select the task to do:

1) Manage Administrators: cscadmin and csccli
2) Restrict SSH Access
3) Radius Configuration
4) Quit
Enter your choice: 3

Welcome to the Radius Authentication Wizard.

This wizard will help you configure Radius Authentication to authenticate and access the CSC SSH Admin console using the radius protocol.
Values required are:
     -> Username/s. (samAccountName if using Windows).
     -> Radius Servers: IP and Shared Secret for Primary and (optional) Secondary.

IMPORTANT:
     -> The CSC uses protocol UDP and port 1812 for communications with the Radius Servers.

Radius Athentication is not currently configured. Do you want to configure Radius Authentication?

1) Yes
2) No
Enter your choice: 1

Radius Servers:

No Radius Severs are configured.

1) Configure Radius Servers.
2) Skip. Leave values as is.
Enter your choice: 1

Primary Radius Server (IP): 172.19.0.100
Primary Radius Shared Secret: 12345

(Optional) Do you want to configure a Secondary Radius Server?

1) Yes
2) No
Enter your choice: 2

No Radius Users are configured

1) ADD Radius Users.
2) Skip. Leave values as is.
Enter your choice: 1

Input Username: radius_user

Do you want to add another Username ?

1) Yes
2) No
Enter your choice: 2

Radius values to configure are:

Primary Server IP= 172.19.0.100 | Shared Secret= 12345
Secondary Server IP not configured

Radius Users:
     Radius Users Qty: 1
     Radius User: radius_user

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1

(MHB-CSC)(INFO) Primary Radius Server with IP:172.19.0.100 was added on zs-csc-mux-4-as-mkt-1
(MHB-CSC)(INFO) Radius Username radius_user was added on zs-csc-mux-4-as-mkt-1
```

-> Example Configuration Windows NPS

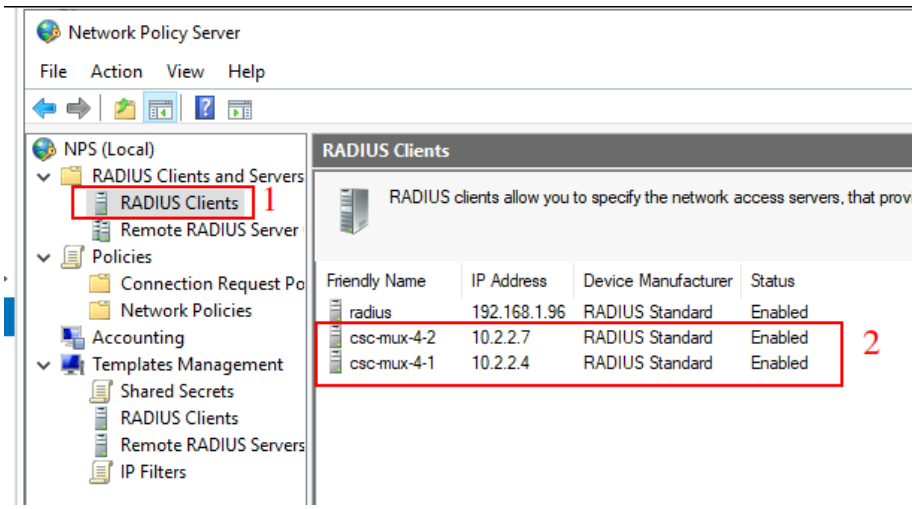
1 - Create Network Policy

In this particular case we are allowing users on the Security Group = Radius Users to authenticate using radius protocol. Please, note the Authentication method required.



2 - Add the CSC as Radius Clients:

Note: The traffic will arrive to the NPS with source IP: CSC GW IP



9.2.3 Change Timezone

Use this menu to select the timezone of the CSC.

WARNING: Some SIEM/SYSLOG software will show the logs in the past or future if the Time Zone is incorrect. In most circumstances, UTC is the best choice.

```
Selection: 7

Your current Time Zone is UTC +0000
WARNING: Some SIEM/SYSLOG software will show the logs in the past or future if the Time Zone is incorrect. In most circumstances, UTC is the best choice.

Do you want to change the Time Zone?
1) Yes
2) No
Enter your choice:
```

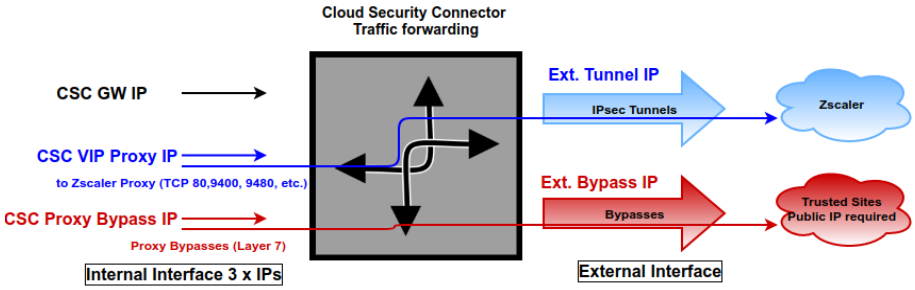
```
Configuring tzdata
Please select the geographic area in which you live. Subsequent configuration questions will narrow this down by presenting a list of cities, representing the time zones in which they are located.

Geographic area:

                                    Africa
                                    America
                                    Antarctica
                                    Australia
                                    Arctic Ocean
                                    Asia
                                    Atlantic Ocean
                                    Europe
                                    Indian Ocean
                                    Pacific Ocean
                                    System V timezones
                                    US
                                    None of the above

                    <OK>                                    <Cancel>
```

9.3 Proxy Bypass

The Proxy Bypass functionality allows doing layer 7 bypasses. This functionality works in conjunction with PAC files.

```
Proxy Bypass
8) View Current Proxy Bypass List
9) Configure Proxy Bypass List
```

9.3.1 Proxy Bypass - Traffic Flow



9.3.2 View Current Proxy Bypass List

This menu displays the current Proxy Bypass List. For example:

```
Selection: 8

This is the list of current Domains configured:

.okta.com
.oktacdn.com
.okta-emea.com
login.mydomain.com
login.microsoftonline.com
login.microsoft.com
login.windows.net
ip.maidenheadbridge.com
```

9.3.3 Configure Proxy Bypass List

This menu allows to configure the Proxy Bypass List.

```
Please, select method to configure Proxy Bypass List

1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice:
```

9.3.3.1 Auto - Proxy Bypass PAC URL

Auto–Proxy Bypass PAC URL is the recommended method to use. You need to create a "Proxy Bypass PAC file" on your Zscaler console. The CSC will read the "Proxy Bypass List" from the "Proxy Bypass PAC file" URL.

The "Proxy Bypass PAC file" URL acts is a central repository of all Layer 7 bypasses required. Moreover, if you manage the CSCs using AWS Systems Manager (or another tool), you can update all CSCs in your network doing one command.

<u>Example of Proxy Bypass PAC:</u>

```
1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 1

Please, select method to configure Proxy Bypass List

1) Configure Proxy Bypass PAC URL and/or Update Proxy Bypasses
2) See PAC Proxy Bypass Example
3) Quit
Enter your choice: 2


Instructions when using Bypass PAC:

1) Create the http://pac.<cloudname>.net/something/<pacname>.pac using the Template example on your Zscaler console.

Template example for PAC Bypass:

----------------
function FindProxyForURL(url, host) {

    var bypassproxy = "PROXY 172.30.200.228:3128";

    // ==========================================================
    // Section 3: bypassproxy via Cloud Security Connectors

    // bypassproxy via CSC Public IPs (Examples)
    // Okta Domains (for Location Rules)
    if ((shExpMatch(host, "*.okta.com")) ||
        (shExpMatch(host, "*.oktacdn.com")) ||
        (shExpMatch(host, "*.okta-emea.com")) ||
        (shExpMatch(host, "login.mydomain.com")) ||
        // O365 Domains for ConditionalAccess
        (shExpMatch(host, "login.microsoftonline.com")) ||
        (shExpMatch(host, "login.microsoft.com")) ||
        (shExpMatch(host, "login.windows.net")) ||
        // Zoho Email IP auth
        (shExpMatch(host, "*.zoho.com")) ||
        // IP test page
        (shExpMatch(host, "ip.maidenheadbridge.com"))) {
        return bypassproxy;
    }
    // ==========================================================

    return bypassproxy;
}
----------------

2) Replace with your values and Copy this information to your production PAC files

3) On your CSC, add your Bypass PAC URL like, http://pac.<cloudname>.net/something/<pacname>.pac

3) Update the Bypass List on the CSC via SSH or AWS Run Command.
```

Note 1: It is mandatory to use this function and format. Feel free to add lines but don't change the format. We recommend to start filling the first line and the last line. Use middle lines for copy/paste.

Note 2: The Bypass Proxy port is 3128

Configuring the Proxy Bypass PAC URL and Refresh the List

```
Selection: 9

Please, select method to configure Proxy Bypass List

1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 1

Please, select method to configure Proxy Bypass List

1) Configure Proxy Bypass PAC URL and/or Update Proxy Bypasses
2) See PAC Proxy Bypass Example
3) Quit
Enter your choice: 1
Proxy Bypass Configuration

Your current Proxy Bypass PAC URL is https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac

Do you want to change the Proxy Bypass PAC URL?
1) Yes
2) No
Enter your choice: 2

Do you want to refresh Proxy Bypass List?
1) Yes
2) No
Enter your choice: 1

This is your current Proxy Bypass List

.okta.com
.oktacdn.com
.okta-emea.com
login.mydomain.com
login.microsoftonline.com
login.microsoft.com
login.windows.net
ip.maidenheadbridge.com


Do you want apply changes?
1) Yes
2) No
Enter your choice: 1

(MHB-CSC)(INFO) Proxy Bypass List updated sucessfully.
```

9.3.3.2 Manual Proxy Bypass Configuration.

If you want to update manually your Proxy Bypass list, follow this steps.

1. Select Option 2)

```
1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 2

Please, read the instructions carefully:

You are going to edit the list using NANO editor

The following formats are accepted:

Full Domains:  'www.example.com'
Wildcard for subdomains: '.example.com' - This will allow all subdomains of example.com

To save, press CTRL-X and 'Yes'

Paid attention to ERROR messages if any. ERRORs must be corrected before to continue

Do you want to continue? (y/n)?
```

2. Input "y"

```
  GNU nano 4.8                    domains                    Modified
.okta.com
.oktacdn.com
.okta-emea.com
login.mydomain.com
login.microsoftonline.com
login.microsoft.com
login.windows.net
portquiz.net
manualAdded.com

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo
```

3. Add / Delete / Modify your full domains and subdomains
4. Please, CTL+X and "Yes" (and after next prompt Enter) to Save
5. The modified Bypass List will be displayed.

```
This is your current Proxy Bypass List

.okta.com
.oktacdn.com
.okta-emea.com
login.mydomain.com
login.microsoftonline.com
login.microsoft.com
login.windows.net
portquiz.net
manualAdded.com


Do you want apply changes?
1) Yes
2) No
Enter your choice:
```

6. Apply Changes Yes or No. If "1" you will receive the following message:

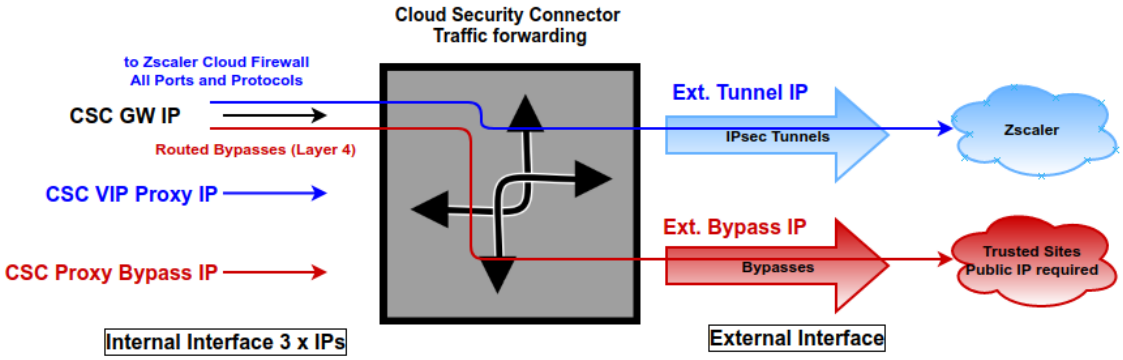
```
Do you want apply changes?
1) Yes
2) No
Enter your choice: 1

Proxy Bypass List updated sucessfully.
```

9.4 Routed Bypass

When routing all traffic via the CSC GW IP, the Routed Bypass functionality allows you to connect specific destinations (IP/Subnet) direct to the Internet using your Public IP. By default, all destinations will travel via the tunnels to Zscaler. If you want to bypass the tunnel, you need to create a Routed Bypass Rule.

```
Routed Bypass
10) View Current Routed Bypass List
11) Configure Routed Bypass List
```

9.4.1 Routed Bypass - Traffic Flow



9.4.2 View Current Routed Bypass List

You can select to view the Routed Bypass Rules in Compact format or JSON.

```
Selection: 10

Please, Select 'Compact' or 'Json' format
1) Compact
2) Json
3) Quit
Enter your choice:
```

9.4.2.1 Compact

```
Enter your choice: 1

Current Values configured are:

Index: 0, Protocol: icmp, SourceIP: 0.0.0.0/0, DestinationIP: 1.1.1.1/32, FromPort: , To Port: , Description: "Test ICMP"
Index: 1, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 20.190.128.0/18, FromPort: 443, To Port: 443, Description: "O365 Login URLs 2"
Index: 2, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 40.126.0.0/18, FromPort: 80, To Port: 80, Description: "O365 Login URLs 3"
Index: 3, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 52.47.209.216/32, FromPort: 80, To Port: 80, Description: "portquiz.net"
Index: 4, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 40.126.0.0/18, FromPort: 443, To Port: 443, Description: "O365 Login URLs 4"
Index: 5, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 13.107.64.0/18, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 1"
Index: 6, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.112.0.0/14, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 2"
Index: 7, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.120.0.0/14, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 3"
Index: 8, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.38.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 1"
Index: 9, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.36.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 2"
Index: 10, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.34.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 3"
Index: 11, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.32.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 4"
```

9.4.2.2 Json

```
Selection: 10

Please, Select 'Compact' or 'Json' format
1) Compact
2) Json
3) Quit
Enter your choice: 2

{
  "routedBypassRules": [
    {
      "description": "Test ICMP",
      "ipProtocol": "icmp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "1.1.1.1/32",
      "fromPort": "",
      "toPort": ""
    },
    {
      "description": "O365 Login URLs 2",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "443",
      "toPort": "443"
    },
    {
      "description": "O365 Login URLs 3",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "40.126.0.0/18",
      "fromPort": "80",
      "toPort": "80"
    },
```

9.4.3 Configure Routed Bypass List

There are two methods to configure the Routed Bypass List: Routed Bypass URL and Manual. The recommended method is to use Routed Bypass URL.

```
Selection: 11

Please, Select Method:
1) Routed Bypass URL
2) Manual (Paste Routed Bypass Rules JSON File)
3) Reset to Default Values
4) Quit
Enter your choice:
```

9.4.3.1 Routed Bypass URL

Routed Bypass URL is the recommended method. Create an AWS bucket or Azure Blob and place your JSON file on it. Here an example:
https://mhb-csc-pac.s3.amazonaws.com/routedBypassRulesFile.json

```
Enter your choice: 1

Please, input Routed Bypass URL
Routed Bypass URL: https://mhb-csc-pac.s3.amazonaws.com/routedBypassRulesFile.json

Do you want to refresh the Routed Bypass List?
1) Yes
2) No
Enter your choice: 1

Routed Bypass JSON file imported successfully


You can review your values before to apply. Please, Select 'Compact' or 'Json' format.
1) Compact
2) Json
3) No review is needed
Enter your choice: 1

Current Values configured are:

Index: 0, Protocol: icmp, SourceIP: 0.0.0.0/0, DestinationIP: 1.1.1.1/32, FromPort: , To Port: , Description: "Test ICMP"
Index: 1, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 20.190.128.0/18, FromPort: 443, To Port: 443, Description: "O365 Login URLs 2"
Index: 2, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 40.126.0.0/18, FromPort: 80, To Port: 80, Description: "O365 Login URLs 3"
Index: 3, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 52.47.209.216/32, FromPort: 80, To Port: 80, Description: "portquiz.net"
Index: 4, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 40.126.0.0/18, FromPort: 443, To Port: 443, Description: "O365 Login URLs 4"
Index: 5, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 13.107.64.0/18, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 1"
Index: 6, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.112.0.0/14, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 2"
Index: 7, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.120.0.0/14, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 3"
Index: 8, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.38.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 1"
Index: 9, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.36.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 2"
Index: 10, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.34.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 3"
Index: 11, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 216.239.32.21/32, FromPort: 443, To Port: 443, Description: "ip.maidenheadbridge.com 4"

Do you want to apply this values?
1) Yes
2) No
Enter your choice: 1

Routed Bypass - (Index: 0) Rule "Test ICMP" was created succesfully.
Routed Bypass - (Index: 1) Rule "O365 Login URLs 2" was created succesfully.
Routed Bypass - (Index: 2) Rule "O365 Login URLs 3" was created succesfully.
Routed Bypass - (Index: 3) Rule "portquiz.net" was created succesfully.
Routed Bypass - (Index: 4) Rule "O365 Login URLs 4" was created succesfully.
Routed Bypass - (Index: 5) Rule "Skype and Teams UDP 1" was created succesfully.
Routed Bypass - (Index: 6) Rule "Skype and Teams UDP 2" was created succesfully.
Routed Bypass - (Index: 7) Rule "Skype and Teams UDP 3" was created succesfully.
Routed Bypass - (Index: 8) Rule "ip.maidenheadbridge.com 1" was created succesfully.
Routed Bypass - (Index: 9) Rule "ip.maidenheadbridge.com 2" was created succesfully.
Routed Bypass - (Index: 10) Rule "ip.maidenheadbridge.com 3" was created succesfully.
Routed Bypass - (Index: 11) Rule "ip.maidenheadbridge.com 4" was created succesfully.

Routed Bypass - Routed Bypass List updated succesfully.
```

9.4.3.2 Manual (Paste Routed Bypass JSON file)

Another option to configure Routed Bypass Rules is to paste the JSON file using the following menu:

```
Enter your choice: 2


WARNING: Manual Configuration will remove the Bypass Routed URL if configured.

Do you want to paste the Routed Bypass Rules JSON File?
1) Yes
2) No
Enter your choice: 1

Please, paste Routed Bypass JSON File and press 'Enter' if required.

NOTE: If the json file has errors, it is possible that the script will hang. Press '}' and 'Enter' to end the operation.

Routed Bypass JSON file:
```

and paste the JSON file. The JSON file will be displayed, and if no errors are found, you can apply the changes:

```
        },
        {
            "description": "Skype and Teams UDP 3",
            "ipProtocol": "udp",
            "sourceCirdIp": "0.0.0.0/0",
            "destinationCirdIp": "52.120.0.0/14",
            "fromPort": "3478",
            "toPort": "3481"
        }
    ]
}

Routed Bypass JSON file imported successfully


You can review your values before to apply. Please, Select 'Compact' or 'Json' format.
1) Compact
2) Json
3) No review is needed
Enter your choice: 1

Current Values configured are:

Index: 0, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 20.190.128.0/18, FromPort: 80, To Port: 80, Description: "O365 Login URLs 1"
Index: 1, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 20.190.128.0/18, FromPort: 443, To Port: 443, Description: "O365 Login URLs 2"
Index: 2, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 40.126.0.0/18, FromPort: 80, To Port: 80, Description: "O365 Login URLs 3"
Index: 3, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 52.47.209.216/32, FromPort: 80, To Port: 80, Description: "portquiz.net"
Index: 4, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 40.126.0.0/18, FromPort: 443, To Port: 443, Description: "O365 Login URLs 4"
Index: 5, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 13.107.64.0/18, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 1"
Index: 6, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.112.0.0/14, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 2"
Index: 7, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.120.0.0/14, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 3"

Do you want to apply this values?
1) Yes
2) No
Enter your choice: 1

(Index: 0) Rule "O365 Login URLs 1" was created succesfully.
(Index: 1) Rule "O365 Login URLs 2" was created succesfully.
(Index: 2) Rule "O365 Login URLs 3" was created succesfully.
(Index: 3) Rule "portquiz.net" was created succesfully.
(Index: 4) Rule "O365 Login URLs 4" was created succesfully.
(Index: 5) Rule "Skype and Teams UDP 1" was created succesfully.
(Index: 6) Rule "Skype and Teams UDP 2" was created succesfully.
(Index: 7) Rule "Skype and Teams UDP 3" was created succesfully.

Routed Bypass List updated succesfully.

Press ENTER to continue
```

9.5 System and Traffic Logs

In this section you can view System Logs, configure Syslog Servers and enable/disable traffic logs.

```
System and Traffic Logs
12) View System Logs
13) Configure Syslog and Traffic Logs
```

9.5.1 View System Logs

```
Selection: 12

Please, Select 'Current Month' or 'Last 6 Months'.

1) Current Month
2) Last 6 Months
3) Quit
Enter your choice: 1

Current Month (May 2023) Logs for zs-csc-mux-4-as-d-1

May 25 01:50:33 root: (MHB-CSC)(DOWN) Load Balancer: All Ztunnels are inactive since: Thu 25 May 01:50:32 UTC 2023
May 25 01:50:35 root: (MHB-CSC)(INFO) Routed Bypass - Routed Bypass Rules JSON file integrity is OK
May 25 01:50:35 root: (MHB-CSC)(INFO) CSC: zs-csc-mux-4-as-d-1: DNS Servers using Azure (168.63.129.16) and Google (8.8.8.8, 8.8.4.4)
May 25 01:50:36 root: (MHB-CSC)(INFO) CSC: zs-csc-mux-4-as-d-1: Syslog Servers using (IP/TCP PORT): 172.19.0.5/5514
May 25 01:50:38 root: (MHB-CSC)(INFO) AWS SSM Agent is  active (running) since Thu 2023-05-25 01:50:38 UTC; 14ms ago
May 25 01:50:38 root: (MHB-CSC)(INFO) AWS SSM Agent Registration values are: {"ManagedInstanceID":"mi-055ab68d5af2fd09e","Region":"us-east-1"}
May 25 01:50:39 root: (MHB-CSC)(INFO) Proxy Bypass List updated sucessfully.
```

9.5.2 Configure Syslog and Traffic Logs

```
Selection: 13

---------------------
Syslog / SIEM Configuration

Syslog / SIEM servers are not configured.
Traffic Logs (IP packets) are disabled.

Do you want to change these values?

1) Yes
2) No
Enter your choice: 1

NOTE: The CSC always generates System Logs (Power UP, Tunnel Changes, etc.), but Traffic Logs (IP Packet information) are optional.
Enabling or Disabling Traffic Logs will require rebooting the CSC.

Traffic Logs are disabled. Do you want to enable Traffic Logs?

1) Yes
2) No
Enter your choice: 1

Primary Syslog Server (IP): 172.19.0.5
Please enter Primary Syslog TCP port: 5514

(Optional) Do you want to configure a Secondary Syslog Server?

1) Yes
2) No
Enter your choice: 2

Please confirm these values:
-------------------
Primary Syslog / SIEM (IP/TCP PORT): 172.19.0.5/5514
Traffic Logs (IP packets) are enabled.
-------------------
Do you want to implement these values?
The CSC will reboot.

1) Yes
2) No
Enter your choice: 1
(MHB-CSC)(INFO) CSC: zs-csc-mux-4-as-d-1: Syslog Servers changed via console. Using (IP/TCP PORT): 172.19.0.5/5514

(MHB-CSC)(INFO) Rebooting the CSC because of a change on Traffic Logs status (disabled to enabled).
Connection to 10.2.2.15 closed by remote host.
Connection to 10.2.2.15 closed.
```

9.6 Configuration Wizards

In this section, you can run the Configuration Wizard to change Zscaler Nodes, VPN Credentials, DNS servers, SNMP, Switch tunnels, and configure High Availability.

```
Configuration Wizards
14) Configure Zscaler Nodes, VPN Credentials, DNS servers and SNMP.
15) Switch Zscaler Tunnels - Primary / Secondary.
16) High Availability configuration.
```

9.6.1 Configure Zscaler Nodes, VPN Credentials, DNS servers and SNMP.

```
Selection: 14

Please, select what you want to configure:

1) Zscaler Nodes and VPN Credentials
2) DNS servers
3) SNMP
4) Quit
Enter your choice:
```

9.6.1.1 Zscaler Nodes and VPN Credentials

This wizard allows you to change the current values configured. It is the same wizard that runs at the initial deployment of the CSC when not using the configUserData.json file to pass User Data.

In this section we are going to show how to select Zscaler Nodes manually.

```
Enter your choice: 1

Checking ZEN Databases...
This CSC has the latest version: 4.62


1) Configuration required on your Zscaler Console: VPN credentials and Location
 --> VPN Credentials creation: Go to > Administration > VPN Credentials > Add VPN Credential -> Select Authentication Type = FDQN and configure 'User ID' and 'Pre-Shared Key'
 --> Location creation: Go to > Administration > Location > Add Location. Put your Location values and select 'VPN Credentials' created in the step before
2) Run the Wizard on the CSC and Select Cloud, Nodes, input VPN Credentials, DNS Servers, Bypass PAC URL and Syslog Servers

Current Values Configured:
-------------------
ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
Primary ZEN node: AutoPrimary | Hostname: vpn.zscalerthree.net | IP: 165.225.48.10
Secondary ZEN node: AutoSecondary | Hostname: secondary.vpn.zscalerthree.net | IP: 165.225.38.52
-------------------
CREDENTIALS INFORMATION
User ID: zs-csc-mux-4-as-d-1@maidenheadbridge.com | Pre-Shared Key: <Run the Wizard to see it>

-------------------
Do you want to continue?

1) Yes
2) No
Enter your choice:
```

Selecting the Nodes:

```
Enter your choice: 1

-----------------------------
Please, select your Cloud

1) zscalerthree
2) zscloud
3) zscalertwo
4) zscaler
5) zscalerone
6) zscalerbeta
7) zscalergov
8) Not in the list? Input Manually
9) Quit
Enter your choice: 1

-----------------------------
Please, select Manual or Auto Node Selection

1) Manual
2) Auto
3) Quit
Enter your choice: 1

-----------------------------
Please, select your Primary Node on 'zscalerthree'
Nodes marked with (-NRU) may be Not Ready for Use. Check http://ips.zscalerthree.net

 1) EMEA,Amsterdam                12) EMEA,LagosII                 23) EMEA,RouenI                  34) Americas,DallasII            45) Americas,VancouverI          56) APAC,NewDelhiI
 2) EMEA,AmsterdamII              13) EMEA,LondonIII               24) EMEA,StockholmIII            35) Americas,DenverIII           46) Americas,WashingtonDC        57) APAC,OsakaI
 3) EMEA,BrusselsII               14) EMEA,MadridIII               25) EMEA,TelAviv                 36) Americas,LosAngeles          47) Americas,WashingtonDC_2      58) APAC,SeoulI
 4) EMEA,CopenhagenII             15) EMEA,ManchesterI             26) EMEA,ViennaI                 37) Americas,MexicoCityI         48) APAC,AucklandII              59) APAC,Shanghai
 5) EMEA,DubaiI                   16) EMEA,MarseilleI              27) EMEA,WarsawII                38) Americas,MiamiIII            49) APAC,Beijing                 60) APAC,ShanghaiII
 6) EMEA,DusseldorfI              17) EMEA,MilanIII                28) EMEA,Zurich                  39) Americas,NewYorkIII          50) APAC,BeijingIII              61) APAC,SingaporeIV
 7) EMEA,FrankfurtIV              18) EMEA,MoscowIII               29) Americas,AtlantaII           40) Americas,NuevoLaredoI        51) APAC,CanberraI               62) APAC,SydneyIII
 8) EMEA,FrankfurtIV_2            19) EMEA,MunichI                 30) Americas,BostonI             41) Americas,SanFranciscoIV_2    52) APAC,ChennaiII               63) APAC,Taipei
 9) EMEA,HelsinkiI                20) EMEA,OsloIII                 31) Americas,Chicago_1           42) Americas,SaoPauloIV          53) APAC,HongKongIII             64) APAC,TokyoIV
10) EMEA,JohannesburgII           21) EMEA,ParisII                 32) Americas,Chicago_2           43) Americas,Seattle             54) APAC,MelbourneII             65) Not in the list? Input Manually
11) EMEA,JohannesburgIII          22) EMEA,ParisII_2               33) Americas,DallasI             44) Americas,TorontoIII          55) APAC,MumbaiVI                66) Quit
Enter your choice: 47

-----------------------------
Please, select your Secondary Node on 'zscalerthree'
Nodes marked with (-NRU) may be Not Ready for Use. Check http://ips.zscalerthree.net

 1) EMEA,Amsterdam                12) EMEA,LagosII                 23) EMEA,RouenI                  34) Americas,DallasII            45) Americas,VancouverI          56) APAC,NewDelhiI
 2) EMEA,AmsterdamII              13) EMEA,LondonIII               24) EMEA,StockholmIII            35) Americas,DenverIII           46) Americas,WashingtonDC        57) APAC,OsakaI
 3) EMEA,BrusselsII               14) EMEA,MadridIII               25) EMEA,TelAviv                 36) Americas,LosAngeles          47) Americas,WashingtonDC_2      58) APAC,SeoulI
 4) EMEA,CopenhagenII             15) EMEA,ManchesterI             26) EMEA,ViennaI                 37) Americas,MexicoCityI         48) APAC,AucklandII              59) APAC,Shanghai
 5) EMEA,DubaiI                   16) EMEA,MarseilleI              27) EMEA,WarsawII                38) Americas,MiamiIII            49) APAC,Beijing                 60) APAC,ShanghaiII
 6) EMEA,DusseldorfI              17) EMEA,MilanIII                28) EMEA,Zurich                  39) Americas,NewYorkIII          50) APAC,BeijingIII              61) APAC,SingaporeIV
 7) EMEA,FrankfurtIV              18) EMEA,MoscowIII               29) Americas,AtlantaII           40) Americas,NuevoLaredoI        51) APAC,CanberraI               62) APAC,SydneyIII
 8) EMEA,FrankfurtIV_2            19) EMEA,MunichI                 30) Americas,BostonI             41) Americas,SanFranciscoIV_2    52) APAC,ChennaiII               63) APAC,Taipei
 9) EMEA,HelsinkiI                20) EMEA,OsloIII                 31) Americas,Chicago_1           42) Americas,SaoPauloIV          53) APAC,HongKongIII             64) APAC,TokyoIV
10) EMEA,JohannesburgII           21) EMEA,ParisII                 32) Americas,Chicago_2           43) Americas,Seattle             54) APAC,MelbourneII             65) Not in the list? Input Manually
11) EMEA,JohannesburgIII          22) EMEA,ParisII_2               33) Americas,DallasI             44) Americas,TorontoIII          55) APAC,MumbaiVI                66) Quit
Enter your choice: 39
```

Next, you can change VPN credentials and to confirm the values.

```
-----------------------------
You have chosen the following:

Cloudname: zscalerthree
Primary node: WashingtonDC_2 (was1-2-vpn.zscalerthree.net)
Secondary Node: NewYorkIII (nyc3-vpn.zscalerthree.net)
-----------------------------

-------------------
CREDENTIALS INFORMATION

You current VPN Credentials are:

User ID: zs-csc-mux-4-as-d-1@maidenheadbridge.com | Pre-Shared Key: ztyckrj6viyAWWzkdWpl3gg0SM7QJjAm

Do you want to change these values?

1) Yes
2) No
Enter your choice: 2

Please confirm this values:
-------------------
Cloudname: zscalerthree
Primary node: WashingtonDC_2 (was1-2-vpn.zscalerthree.net)
Secondary Node: NewYorkIII (nyc3-vpn.zscalerthree.net)
-------------------
VPN Credentials
User ID: zs-csc-mux-4-as-d-1@maidenheadbridge.com | Pre-Shared Key: ztyckrj6viyAWWzkdWpl3gg0SM7QJjAm
-------------------
Do you want to implement this values?

1) Yes
2) No
Enter your choice: 1

Validating Configuration

Your Cloud is: zscalerthree

Checking Node WashingtonDC_2 hostname was1-2-vpn.zscalerthree.net
Hostname was1-2-vpn.zscalerthree.net has IP 165.225.8.35
Node WashingtonDC_2 is Alive

Checking Node NewYorkIII hostname nyc3-vpn.zscalerthree.net
Hostname nyc3-vpn.zscalerthree.net has IP 165.225.38.52
Node NewYorkIII is Alive

Do you want to apply this values? (y/n)?:y

(MHB-CSC)(INFO) CSC:zs-csc-mux-4-as-d-1 connected to Zscaler Cloud: zscalerthree
(MHB-CSC)(INFO) Primary Zscaler Node using: WashingtonDC_2, hostname: was1-2-vpn.zscalerthree.net (IP: 165.225.8.35) on CSC:zs-csc-mux-4-as-d-1
(MHB-CSC)(INFO) Secondary Zscaler Node using: NewYorkIII, hostname: nyc3-vpn.zscalerthree.net (IP: 165.225.38.52) on CSC:zs-csc-mux-4-as-d-1
(MHB-CSC)(INFO) VPN Credentials using FQDN: zs-csc-mux-4-as-d-1@maidenheadbridge.com on CSC:zs-csc-mux-4-as-d-1
```

9.6.1.2 DNS servers

```
Selection: 14

Please, select what you want to configure:

1) Zscaler Nodes and VPN Credentials
2) DNS servers
3) SNMP
4) Quit
Enter your choice: 2

You are using Azure DNS server 168.63.129.16, 8.8.8.8 and 8.8.4.4

Do you want to change the DNS servers?

1) Yes
2) No
Enter your choice: 1

Primary DNS Server (IP): 1.1.1.1
Secondary DNS Server (IP): 8.8.8.8

(MHB-CSC)(INFO) CSC: zs-csc-mux-4-as-d-1: DNS Servers changed via console. Using 1.1.1.1 and 8.8.8.8
```

9.6.1.3 SNMP

The CSC Mux uses Ubuntu Server as its OS and offers all SNMP values of a standard Ubuntu Server. The CSC Mux supports SNMP v2c or v3. No special MIBs are required.

SNMP Traps are not supported. For information about tunnels up/down and other changes, please, use Systems Logs to trigger alarms or events.

9.6.1.3.1 Configure SNMP attributes

```
Selection: 14

Please, select what you want to configure:

1) Zscaler Nodes and VPN Credentials
2) DNS servers
3) SNMP
4) Quit
Enter your choice: 3

Welcome to the SNMP Wizard.

This wizard will help you to configure SNMP Attributes (name, location, etc.), SNMP Version (v2c or v3) and Host (/32) or Subnet (IP/Subnet Prefix) allowed to access the CSC via SNMP.
The SNMP configuration is read only. Via SNMP, you can obtain all CSC Information and Statistics, but you cannot configure anything.
The CSC is based on Ubuntu OS. All SNMP values offered by Ubuntu OS by default are available. Special MIBs are not required.

SNMP is not currently configured. Do you want to configure SNMP?

1) Yes
2) No
Enter your choice: 1

Current SNMP Attributes configured are:
Name=
Location=
Description=
Contact=

Do you want to configure SNMP Attributes?

1) Configure SNMP Attributes.
2) Skip. Leave values as is.
3) Reset ALL SNMP parameters to default.
Enter your choice: 1

Please input Name for this device: zs-csc-mux-4-as-d-1
Please input Location for this device: Azure East US
Please input Description for this device: Zscaler Mux 4 on Azure East
Please input Contact for this device: support@maidenheadbridge.com
```

9.6.1.3.2 SNMP v2c configuration

SNMP version 2c requires the "read only community" and the IP or Subnet of the SNMP platform.

In this example, our SNMP server has IP: 172.19.0.8/32 and the rocommunity is "public".

```
SNMP v2c Configuration

SNMP v2c is not configured

Do you want to configure SNMP v2c ?

1) Configure SNMP v2c.
2) Skip. Leave values as is.
3) Disable SNMP v2c.
Enter your choice: 1

Please input SNMP v2c Read Only Community: public

SNMP v3 Configuration

SNMP v3 is not configured.

Do you want to configure SNMP v3 ?

1) Configure SNMP v3.
2) Skip. Leave values as is.        Skip SNMP v3
3) Disable SNMP v3.
Enter your choice: 2
```

9.6.1.3.3 SNMP Networks

The CSC blocks all SNMP request by default. You need to enable the source IPs (or Subnets) that will query the CSC using SNMP. This setting is mandatory for SNMP v2c and v3.

```
SNMP access is NOT allowed from any Host (/32) or Subnet (IP/Subnet Prefix).

Please allow SNMP acess to specific Hosts (/32) or Subnets (IP/Subnet Prefix). This is a mandatory setting.

1) Configure Networks.
2) Skip. Leave values as is.
3) Reset to Default.
Enter your choice: 1

Input Host (/32) or Subnet (IP/Subnet Prefix): 172.19.0.8/32

Do you want to add another Host (/32) or Subnet (IP/Subnet Prefix)?

1) Yes
2) No
Enter your choice: 2

SNMP values to configure are:

Name= zs-csc-mux-4-as-d-1
Location= Azure East US
Description= Zscaler Mux 4 on Azure East
Contact= support@maidenheadbridge.com

SNMP v2c:
     Read-only Community name: public

Networks:
     Networks Qty: 1
     Host or Subnet: 172.19.0.8/32
     IMPORTANT: If the Host or Subnet is reachable via PriCPA interface and not Internal Interface eth1, you must add these Host or Subnet as Management Networks on PriCPA configuration.

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1
(MHB-CSC)(INFO) SNMP Network 172.19.0.8/32 was added on zs-csc-mux-4-as-d-1
SNMP Status is:  active (running) since Thu 2023-06-01 22:42:59 UTC; 807ms ago
(MHB-CSC)(INFO) SNMP configuration was changed on zs-csc-mux-4-as-d-1
```

9.6.1.3.4 SNMP v3 configuration

SNMP attributes and Networks are standard settings of SNMP v2c and SNMP v3. This section will show the specific values required for SNMP v3.

1. Security Name (or UserName) : <string>
2. Security Level: noAuthNoPriv|authNoPriv|authPriv
3. Authentication Passphrase: <string>
4. Authentication Protocol: MD5|SHA|SHA-512|SHA-384|SHA-256|SHA-224
5. Privacy Passphrase: <string>
6. Privacy Protocol: DES|AES

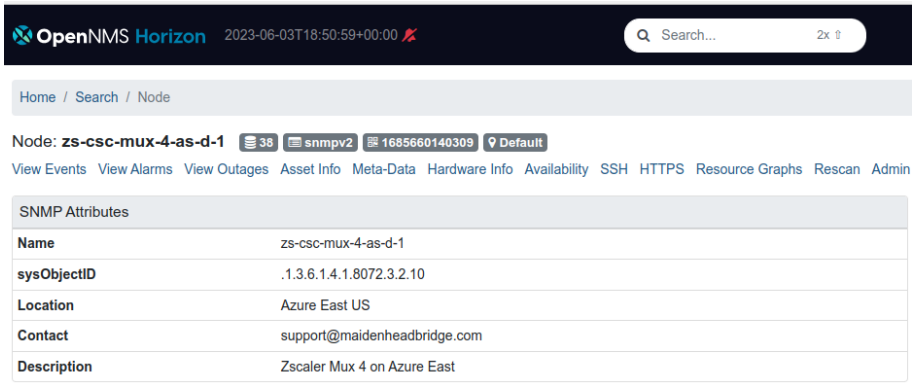
```
SNMP v2c is not configured

Do you want to configure SNMP v2c ?

1) Configure SNMP v2c.
2) Skip. Leave values as is.        Skip v2c
3) Disable SNMP v2c.
Enter your choice: 2

SNMP v3 Configuration

SNMP v3 is not configured.

Do you want to configure SNMP v3 ?

1) Configure SNMP v3.
2) Skip. Leave values as is.
3) Disable SNMP v3.
Enter your choice: 1

Please input Security Name (string): authPrivUser

Please input Security Level (noAuthNoPriv|authNoPriv|authPriv):
1) noAuthNoPriv
2) authNoPriv
3) authPriv
Enter your choice: 3

Please input Authentication Passphrase (string): mhbAuth1

Please input Authentication Protocol (MD5|SHA|SHA-512|SHA-384|SHA-256|SHA-224):
1) MD5
2) SHA
3) SHA-512
4) SHA-384
5) SHA-256
6) SHA-224
Enter your choice: 3

Please input Privacy Passphrase (string): mhbPriv1

Please input Privacy Protocol (DES|AES):
1) DES
2) AES
Enter your choice: 2

SNMP access is NOT allowed from any Host (/32) or Subnet (IP/Subnet Prefix).

Please allow SNMP acess to specific Hosts (/32) or Subnets (IP/Subnet Prefix). This is a mandatory setting.

1) Configure Networks.
2) Skip. Leave values as is.
3) Reset to Default.
Enter your choice: 1

Input Host (/32) or Subnet (IP/Subnet Prefix): 172.19.0.8/32
```

```
SNMP values to configure are:

Name= zs-csc-mux-4-as-d-2
Location= Azure East US
Description= Zscaler Mux 4 on Azure East
Contact= support@maidenheadbridge.com

SNMP v3:
     SecurityName= authPrivUser
     SecurityLevel= authPriv
     AuthPassphrase= mhbAuth1
     AuthProtocol= SHA-512
     PrivacyPassphrase= mhbPriv1
     PrivacyProtocol= AES

Networks:
     Networks Qty: 1
     Host or Subnet: 172.19.0.8/32
     IMPORTANT: If the Host or Subnet is reachable via PriCPA interface and not Internal Interface eth1, you must add these Host or Subnet as Management Networks on PriCPA configuration.

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1
(MHB-CSC)(INFO) SNMP Network 172.19.0.8/32 was added on zs-csc-mux-4-as-d-2
SNMP Status is:  active (running) since Sat 2023-06-03 07:56:40 UTC; 779ms ago
(MHB-CSC)(INFO) SNMP configuration was changed on zs-csc-mux-4-as-d-2
```

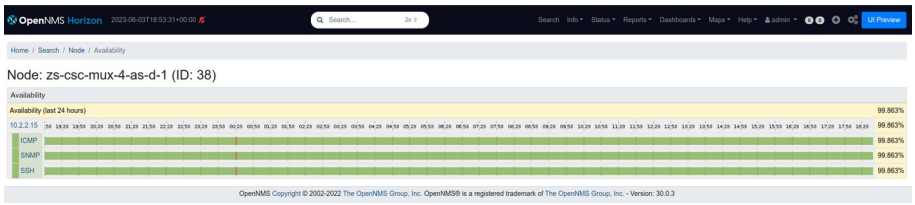
9.6.1.3.5 What can you do with SNMP?

Here some examples of monitoring the CSC Mux via SNMP, using OpenNMS.

9.6.1.3.5.1 Node Information



9.6.1.3.5.2 Node Availability



9.6.1.3.5.3 Node Interfaces (IP & SNMP)

Node Interfaces

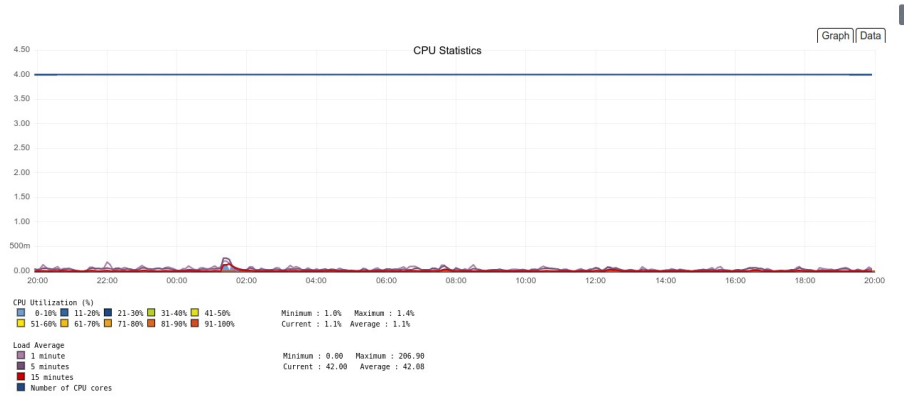
IP Interfaces | SNMP Interfaces

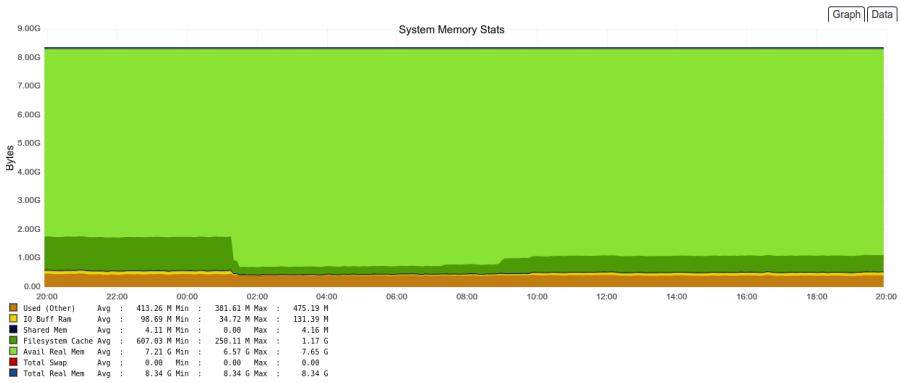
Search/Filter SNMP Interfaces

SNMP ifIndex	SNMP ifDescr	SNMP ifName	SNMP ifAlias	SNMP ifSpeed
1	lo	lo	N/A	10000000
2	eth0	eth0	N/A	N/A
3	eth1	eth1	N/A	N/A
4	eth2	eth2	N/A	N/A
5	eth3	eth3	N/A	N/A
6	zum1	zum1	N/A	N/A
8	ztun1a	ztun1a	N/A	10000000000
10	ztun2a	ztun2a	N/A	10000000000
12	ztun3a	ztun3a	N/A	10000000000
14	ztun4a	ztun4a	N/A	10000000000

First Previous 1 2 Next Last

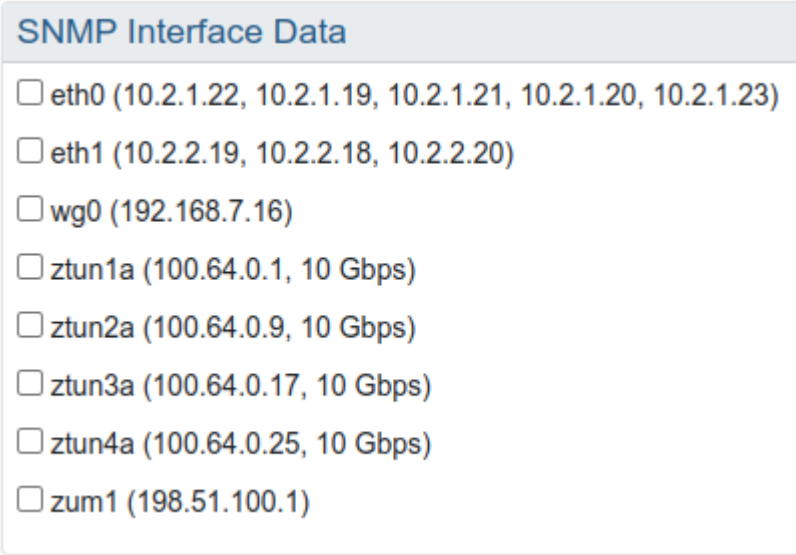
9.6.1.3.5.4 Node Statistics (CPU, Memory, etc)



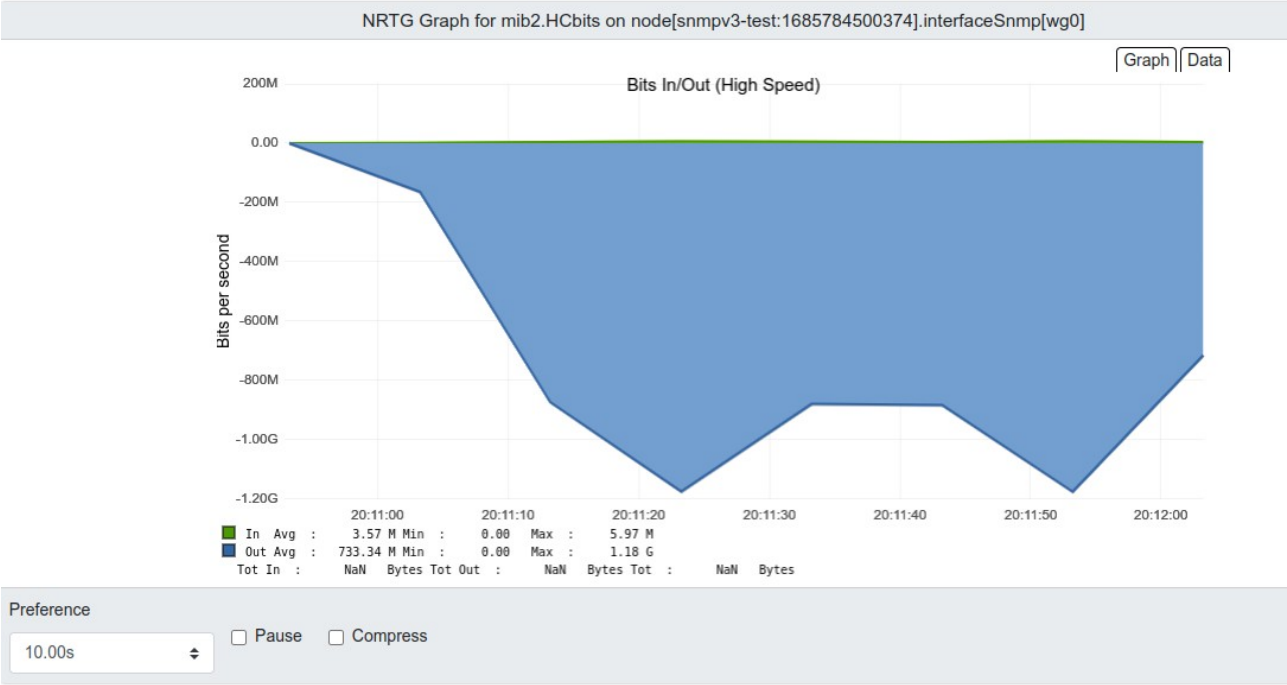


9.6.1.3.5.5 Interfaces Traffic

You can see the traffic per physical interfaces (eth0, eth1), tunnel interfaces (Ztunx), and PriCPA interface (wg0).



Example of real time traffic on PriCPA interface:



9.6.2 Switch Tunnels - Primary / Secondary.

This Wizard allows to Switch Tunnels Primary to Secondary and vice-versa.

```
Configuration Wizards
14) Configure Zscaler Nodes, VPN Credentials, DNS servers and SNMP.
15) Switch Zscaler Tunnels - Primary / Secondary.
16) High Availability configuration.
```

```
Selection: 15
--------------------
ZSCALER INFORMATION

You current Zscaler Cloud and Nodes are:

Zscaler Cloud:  zscalerthree
Primary ZEN node: WashingtonDC_2 | Hostname: was1-2-vpn.zscalerthree.net | IP: 165.225.8.35
Secondary ZEN node: NewYorkIII | Hostname: nyc3-vpn.zscalerthree.net | IP: 165.225.38.52

Do you want to switch tunnels? (y/n) y


Validating Configuration

Your Cloud is: zscalerthree

Checking Node NewYorkIII hostname nyc3-vpn.zscalerthree.net
Hostname nyc3-vpn.zscalerthree.net has IP 165.225.38.52
Node NewYorkIII is Alive

Checking Node WashingtonDC_2 hostname was1-2-vpn.zscalerthree.net
Hostname was1-2-vpn.zscalerthree.net has IP 165.225.8.35
Node WashingtonDC_2 is Alive

Do you want to apply this values? (y/n)?:y

(MHB-CSC)(INFO) CSC:zs-csc-mux-4-as-d-1 connected to Zscaler Cloud: zscalerthree
(MHB-CSC)(INFO) Primary Zscaler Node using: NewYorkIII, hostname: nyc3-vpn.zscalerthree.net (IP: 165.225.38.52) on CSC:zs-csc-mux-4-as-d-1
(MHB-CSC)(INFO) Secondary Zscaler Node using: WashingtonDC_2, hostname: was1-2-vpn.zscalerthree.net (IP: 165.225.8.35) on CSC:zs-csc-mux-4-as-d-1
(MHB-CSC)(INFO) VPN Credentials using FQDN: zs-csc-mux-4-as-d-1@maidenheadbridge.com on CSC:zs-csc-mux-4-as-d-1
```

9.6.3 High Availability configuration

When deployed as High Availability pair, the CSCs can manage the "Next-Hop" of the route/s configured.

The CSC active will assign its CSC GW IP as the "Next-Hop" of the routes configured. You can configure several routes; there is no limit.

For example, you can control the default route to the internet, 0.0.0.0/0 or any other destination via the CSCs.

```
Selection: 16

This Wizard is for High Availability scenarios when changing Next-Hop on Routes via CSC HA Pair.

---------------------
How to configure:

1) 'Deployment': Deploy a pair of CSCs with the following conditions:
   1.1) There is connectivity each other via their internal interfaces.
   1.2) (Optional, but Recommended) Deploy the CSCs on Availability Zones or Availability Sets.
2) 'Identity': On each CSC VM
   2.1) Go to 'Identity -> System Assigned' and 'Turn ON' status. (and Save).
   2.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:
       -> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>
       -> Role: Contributor, Resource Group: <Route Tables Resource Group/s>
       -> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>
3) 'Routes'
   3.1) Go to Routes (inside the Route Table) and create the Routes that the CSC HA group will control:
      -> Route name: <any name you want>
      -> Address prefix: <Subnet/Mask>
            Examples: 0.0.0.0/0 (if you want to send all traffic via Zscaler) or 185.46.212.88/32 (when using PAC files and/or Explicit Proxy)
      -> Next hop type: Virtual Appliance
      -> Next hop address: <Input CSC-GW-IP (eth1, first IP) of any CSC>
   3.2) Go to Subnets and associate the Subnet with the Route Table.
   3.3) Repeat the process if you want to add more Routes. The CSC HA functionality can manipulate multiple Routes.
4) Obtain the following values and Run the Wizard.
   4.1) Route, Route Table, Resource Group.
   4.2) Computer Name and Resource Group of each CSC.
5) This Wizard will create a JSON file. You can use this JSON file to configure the Other CSC in the pair.

How it works:

The CSCs on the HA pair will automatically select the Next-Hop for the Route/s configured.

---------------------

The HA service in NOT Active

Do you want to configure it?

1) Yes
2) No
Enter your choice:
```

Help provided:

How to configure:

1) 'Deployment': Deploy a pair of CSCs with the following conditions:

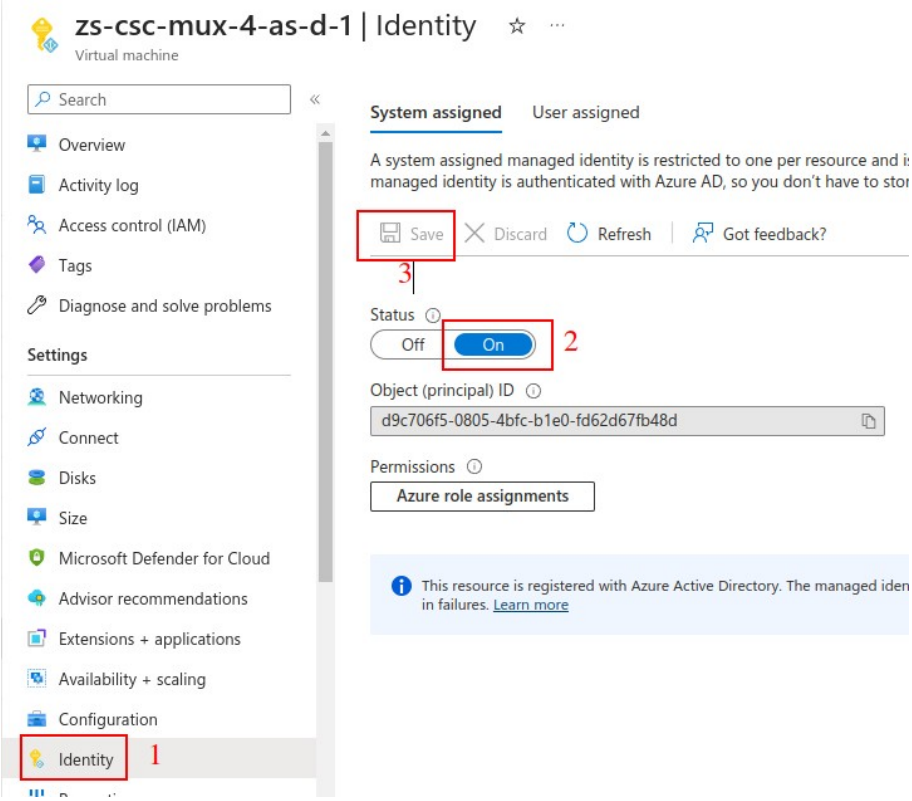
1.1) There is connectivity each other via their internal interfaces.

1.2) (Optional, but Recommended) Deploy the CSCs on Availability Zones or Availability Sets.

2) 'Identity': On each CSC VM

2.1) Go to 'Identity -> System Assigned' and 'Turn ON' status. (and Save).

Example:



Note: Repeat the same step on the other CSC on the HA Pair.

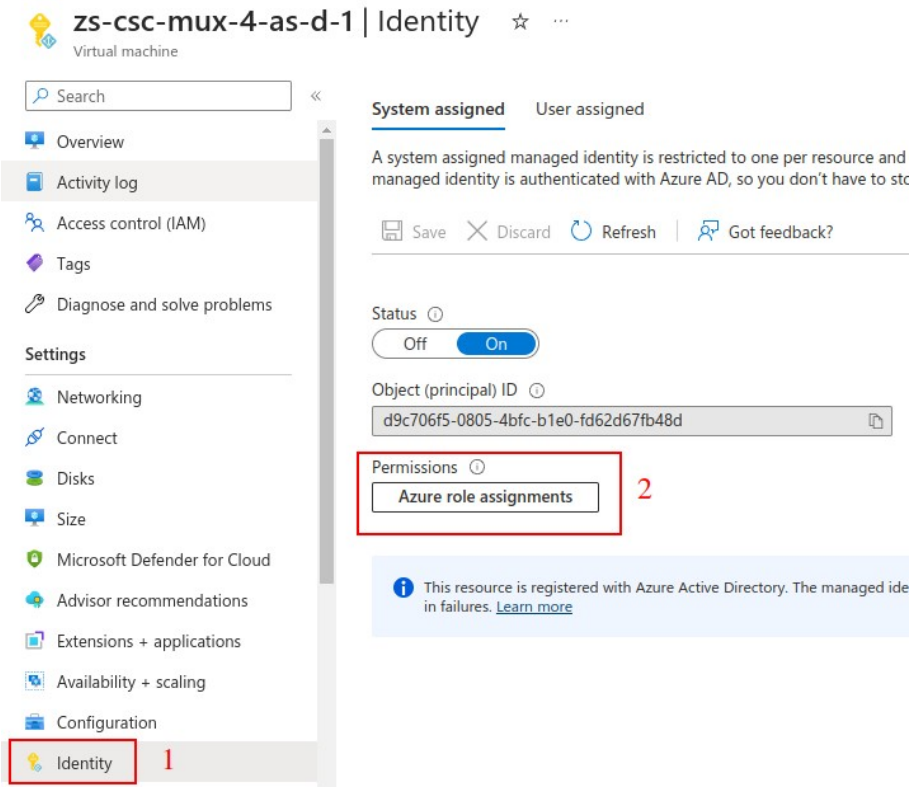
2.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:

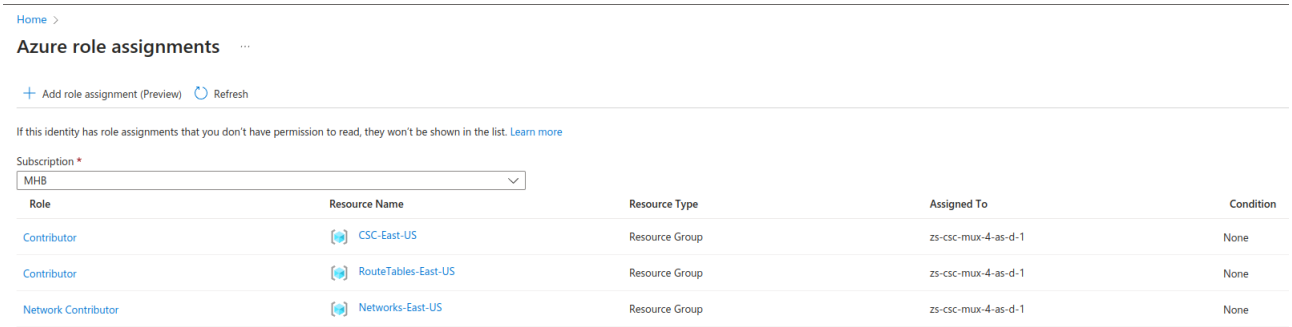
-> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>

-> Role: Contributor, Resource Group: <Route Tables Resource Group/s>

-> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>

Example:





Note: Repeat the same step on the other CSC on the HA Pair.

3) 'Routes'

3.1) Go to Routes (inside the Route Table) and create the Routes that the CSC HA group will control:

-> Route name: <any name you want>

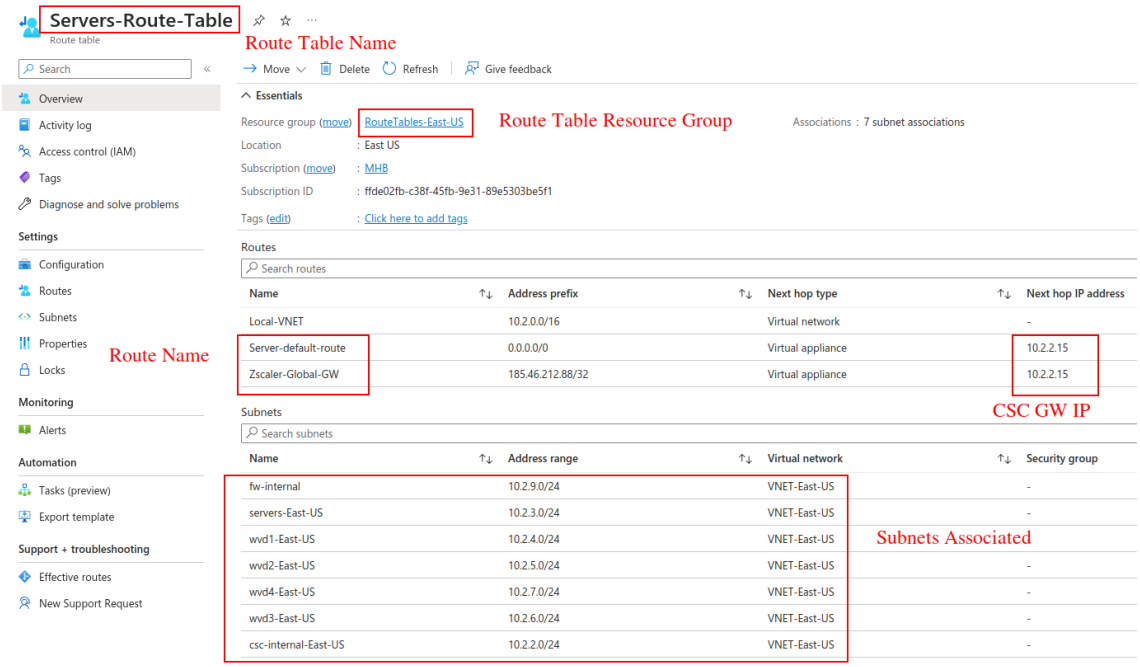
-> Address prefix: <Subnet/Mask>

Examples: 0.0.0.0/0 (if you want to send all traffic via Zscaler) or 185.46.212.88/32 (when using PAC files and/or Explicit Proxy)

-> Next hop type: Virtual Appliance

-> Next hop address: <Input CSC-GW-IP (eth1, first IP) of any CSC>

Example:



3.2) Go to Subnets and associate the Subnet with the Route Table.

3.3) Repeat the process if you want to add more Routes. The CSC HA functionality can manipulate multiple Routes.

4) Obtain the following values and Run the Wizard.

4.1) Route, Route Table, Resource Group.

4.2) Computer Name and Resource Group of each CSC.

Example:

First CSC on the HA Pair – Manual Configuration:

```
Enter your choice: 1

Please, Select 'Manual' to configure the First CSC on the HA pair or 'Json' for the Second CSC.

1) Manual
2) Json
3) Quit
Enter your choice: 1

Identity Type: SystemAssigned

Please, input the Route/s values:

Route Name= Server-default-route
Route Table= Servers-Route-Table
Resource Group= RouteTables-East-US

Do you want to add another Route? (y/n)? y

Route Name= Zscaler-Global-GW
Route Table= Servers-Route-Table
Resource Group= RouteTables-East-US

Do you want to add another Route? (y/n)? n

Please, input values of other CSC in the pair

Computer Name= zs-csc-mux-4-as-d-2
Resource Group= CSC-EAST-US

Values to configure are:

Route/s (Qty)=2
Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
Route 2: Zscaler-Global-GW (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)

Computer Name of other CSC in the pair: zs-csc-mux-4-as-d-2 (Resource Group=CSC-EAST-US)

Do you want to apply changes? (y/n)? y
```

Input:
- Route Name
- Route Table
- Resource Group

Input:
- Computer Name
- Resource Group

5) This Wizard will create a JSON file. You can use this JSON file to configure the Other CSC in the pair.

```
Do you want to apply changes? (y/n)? y

Please, copy the following values in a safe place to configure the other CSC on the High Availability Pair.


High Availability JSON file:
_____________________________________________________________

{
  "model": "csc-mux-zs-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "SystemAssigned",
    "haFirstCsc": {
      "vmName": "zs-csc-mux-4-as-d-1",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "74.235.173.101"
    },
    "haSecondCsc": {
      "vmName": "zs-csc-mux-4-as-d-2",
      "vmResourceGroup": "CSC-EAST-US",
      "haBypassPublicIp": "74.235.173.171"
    },
    "haPrivateAccessPublicIp": "74.235.173.101",
    "haRoutes": [
      {
        "routeName": "Server-default-route",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      },
      {
        "routeName": "Zscaler-Global-GW",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      }
    ]
  }
}
_____________________________________________________________

CSC HA is : active (running) since Wed 2023-05-31 22:34:09 UTC; 17ms ago
```

Example:

Second CSC on the HA Pair – (paste) JSON Configuration:

```
Please, Select 'Manual' to configure the First CSC on the HA pair or 'Json' for the Second CSC.

1) Manual
2) Json
3) Quit
Enter your choice: 2

Please, paste 'High Availability JSON file' and press 'Enter' if required.

NOTE: If the json file has errors, it is possible that the script will hang. Press '}' and 'Enter' to end the operation.

High Availability JSON file: {
  "model": "csc-mux-zs-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "SystemAssigned",
    "haFirstCsc": {
      "vmName": "zs-csc-mux-4-as-d-1",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "74.235.173.101"
    },
    "haSecondCsc": {
      "vmName": "zs-csc-mux-4-as-d-2",
      "vmResourceGroup": "CSC-EAST-US",
      "haBypassPublicIp": "74.235.173.171"
    },
    "haPrivateAccessPublicIp": "74.235.173.101",
    "haRoutes": [
      {
        "routeName": "Server-default-route",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      },
      {
        "routeName": "Zscaler-Global-GW",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      }
    ]
  }
}
(MHB-CSC)(INFO) High Availability JSON file (highAvailability.json) integrity is OK
(MHB-CSC)(INFO) High Availability: IAM Identity in use: SystemAssigned
(MHB-CSC)(INFO) High Availability: Route Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
(MHB-CSC)(INFO) High Availability: Route Zscaler-Global-GW (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
(MHB-CSC)(INFO) High Availability is  active (running) since Wed 2023-05-31 22:43:01 UTC; 14ms ago.
```

How it works:

The CSCs on the HA pair will automatically select the Next-Hop for the Route/s configured.

Check HA using Show Configuration and Status:

```
HIGH AVAILABILITY Information
The HA service is:  active (running) since Wed 2023-05-31 22:34:09 UTC; 11min ago
Identity Type: SystemAssigned
Route to Zscaler using Next Hop: 10.2.2.15 of VM: zs-csc-mux-4-as-d-1 (this CSC)
Current values configured are:
   Route/s (Qty)= 2
      Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
      Route 2: Zscaler-Global-GW (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
   Computer Name of other CSC in the pair: zs-csc-mux-4-as-d-2 (Resource Group=CSC-EAST-US)
Private Access Public IP= 74.235.173.101
```

Note: The HA wizard automatically selects the Floating IP for Private Access.

Logs generated by High Availability:

```
May 31 22:42:45 root: (MHB-CSC)(INFO) High Availability JSON file (highAvailability.json) integrity is OK
May 31 22:42:51 root: (MHB-CSC)(INFO) High Availability: IAM Identity in use: SystemAssigned
May 31 22:43:01 root: (MHB-CSC)(INFO) High Availability: Route Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
May 31 22:43:01 root: (MHB-CSC)(INFO) High Availability: Route Zscaler-Global-GW (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
May 31 22:43:01 root: (MHB-CSC)(INFO) High Availability is  active (running) since Wed 2023-05-31 22:43:01 UTC; 14ms ago.
May 31 22:44:41 cscadmin: (MHB-CSC)(INFO) Route to Zscaler using Next Hop: 10.2.2.15 of CSC: zs-csc-mux-4-as-d-1
```

10 Traffic Forwarding to Zscaler ZIA with the CSC Mux for Azure.

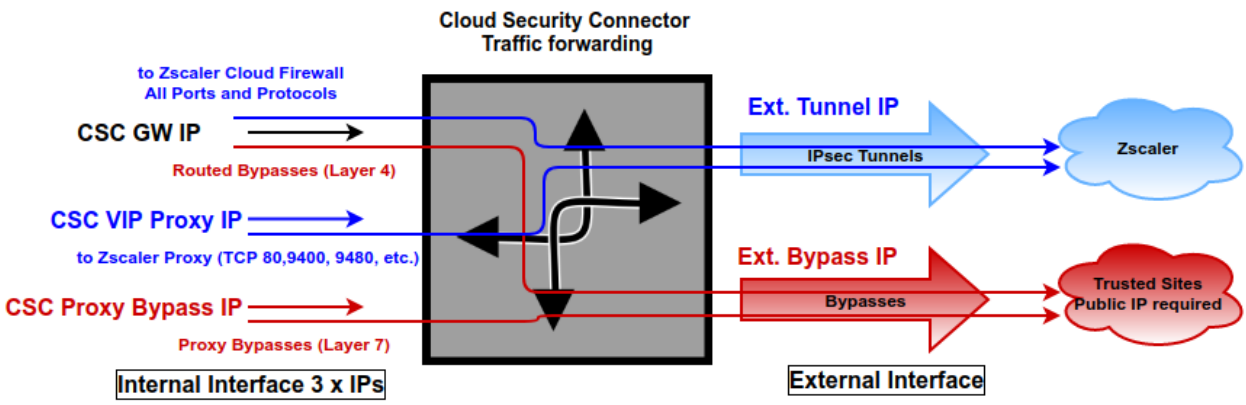
In Chapter 3 of this Administrator Guide, we showed the Network Diagrams of different scenarios of traffic forwarding.

When connecting Virtual Machines, Virtual Desktops, etc., to Zscaler using the CSC Mux, you have two options for traffic forwarding: routing and proxying.

The options are not mutually exclusive. You can use both at the same time. Moreover, when the CSC Mux is on a HA pair, you can use both simultaneously, duplicating the capacity for Web Traffic.

In both cases, it is possible to bypass traffic from the tunnel to Zscaler and send it directly via a dedicated Public IP, using the "routed" and "proxied" bypass functionalities.

As shown in a previous chapter, the CSC offers multiple options for traffic forwarding:



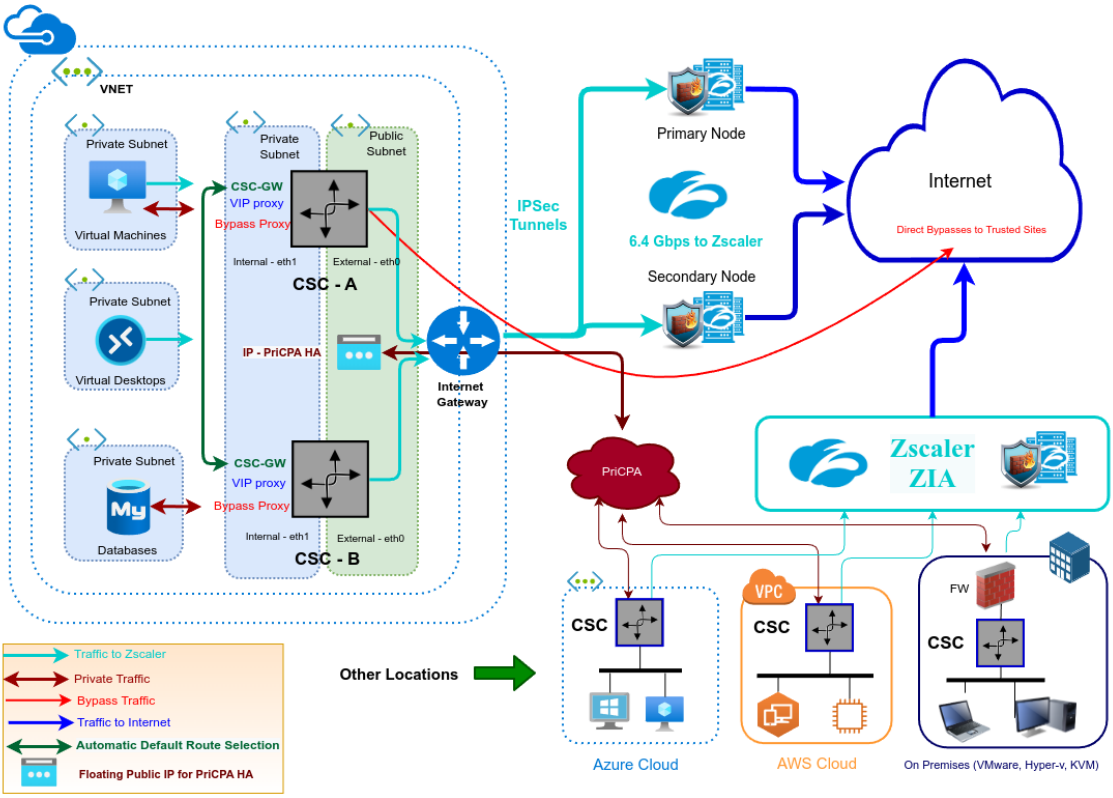
The function of each internal IP is the following:

IP	Type	Function
CSC GW	Gateway	Used as Gateway for traffic to Zscaler and bypasses using "Routed Bypass" (Layer 4) functionality.
CSC Vip Proxy	Proxy	Used as Proxy for traffic to Zscaler.
CSC Proxy Bypass	Proxy	User as Proxy for bypasses using "Proxy Bypass" (Layer 7) functionality.

This chapter will dig into more detail about the configuration required, showing examples when the CSC Mux is on High Availability as HA Pair and when using Azure Load Balancer.

10.1 CSC Mux in HA Pair

10.1.1 Network Diagram



10.1.2 Prerequisites

1. Deploy 2 x CSC Mux as HA Pair. (See section "9.6.3 High Availability configuration")
2. Create the Routed Bypasses. (See section "9.4 Routed Bypass")
3. Obtain the VIP Proxy and Bypass Proxy of each CSC running "Show Configuration and Status" from SSH console or AWS Systems Manager.
4. Created the Proxy Bypasses. (See section "9.3 Proxy Bypass")

10.1.3 Real Case Scenario: Routing, Explicit Proxy and PAC files.

Suppose a customer with the following requirements:

- Case 1): Send all ports and protocols to Zscaler and reach Trusted Sites using your Public IP (Routed Bypass - Layer 4).

- Case 2): To provide redundancy to servers that do not support PAC files. Only Explicit Proxy (Single IP) can be configured and reach Trusted Sites using your Public IP (Routed Bypass).
- Case 3): To provide the maximum bandwidth available to Virtual Desktops (or any device that can use PAC file) using both CSC simultaneously for Web Traffic and reach Trusted Sites using your Public IP. (Routed Bypass - Layer 4 or Proxy Bypass - Layer 7).

10.1.3.1 Routing and Explicit proxy: Solving Case 1 and 2

To solve requirements 1) and 2), the CSC Mux on HA pair will manage the Routes on the Routing Tables to select the best exit to Zscaler.

In the case of 1) the CSC Mux Hair Par will control de default routes Next-Hop to Zscaler for the server farm.

Default Route to Internet → 0.0.0.0/0

All traffic to Zscaler:

Routes

Search routes

Name	Address prefix	Next hop
CSC-Zscaler-Default	0.0.0.0/0	172.31.200.17

In the case of 2), we are going to use the Zscaler Global Proxy IP as Explicit Proxy on the servers.

You can use any of this values:

Zscaler Global ZEN IP addresses → 185.46.212.88/32, 185.46.212.89/32, 185.46.212.90/32, 185.46.212.91/32, 185.46.212.92/32, 185.46.212.93/32, 185.46.212.97/32, 185.46.212.98/32.

The CSC Mux HA pair will control this routes to Zscaler Global ZEN IP address.

To Zscaler Global ZENs:

Routes

Search routes

Name	Address prefix	Next hop
server-farm-1	185.46.212.88/32	172.31.200.17

The CSC on the HA Pair will manage the Next Hop this this Address Prefix: 185.46.212.88/32

IMPORTANT: The Next-Hop IP is the CSC GW IP.

10.1.3.2 Case 1, 2 and 3: Routed Bypasses - Layer 4

Routed Bypasses works in a similar way to an outbound Azure Security Group. You can create rules per Source IP, Destination IP, Protocol (UDP / TCP) and destination port.

The configuration file for Routed Bypasses is a JSON file with the following format:

```
{
  "routedBypassRules": [
    {
      "description": "O365 Login URLs 1",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "O365 Login URLs 2",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "443",
      "toPort": "443"
    },
    {
      "description": "O365 Login URLs 3",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "40.126.0.0/18",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "portquiz.net",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "52.47.209.216/32",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "O365 Login URLs 4",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "40.126.0.0/18",
      "fromPort": "443",
      "toPort": "443"
    },
    {
      "description": "Skype and Teams UDP 1",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "13.107.64.0/18",
      "fromPort": "3478",
      "toPort": "3481"
    },
    {
      "description": "Skype and Teams UDP 2",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "52.112.0.0/14",
      "fromPort": "3478",
      "toPort": "3481"
    },
    {
      "description": "Skype and Teams UDP 3",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "52.120.0.0/14",
      "fromPort": "3478",
      "toPort": "3481"
    }
  ]
}
```

To configure Routed Bypasses on the CSC, you paste the JSON file directly via the SSH console or configure an URL from where the CSC can retrieve the JSON file. You can create an object on a Blob container and configure the URL of the object on the CSC.

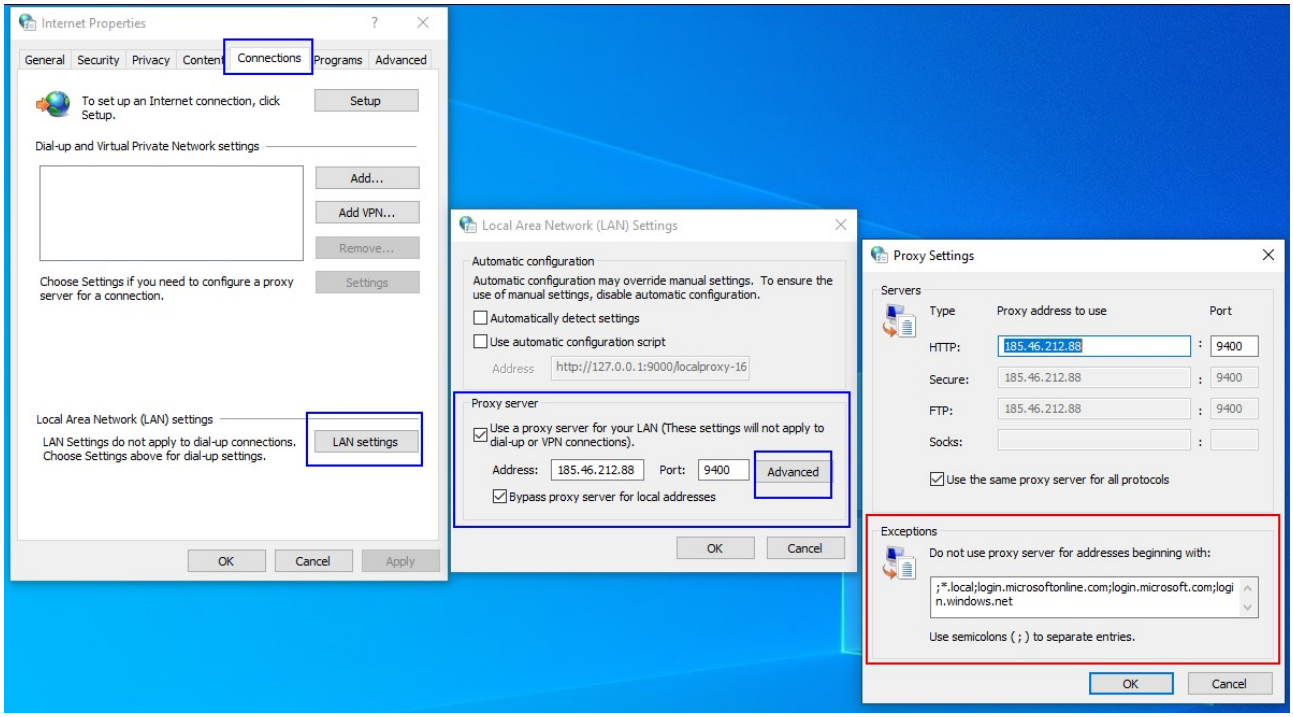
Routed Bypasses for Case 1 - Sending all ports and protocols via the CSC GW.

The rules of Routed Bypass will inspect any traffic routed by the CSC GW IP. You can bypass any TCP or UDP traffic from Zscaler, for example, O365 Authentication URLs for Conditional Access rules and MFA, and Skype UDP real-time traffic. In this case, there is nothing to configure on the internal devices.

Routed Bypasses for Case 2 - Servers with Explicit Proxy

In this case, the traffic sent via the Routed Bypass is configured on "Exceptions" (Windows) or "no_proxy" (Linux). Here is an example for each one.

Windows:



Linux (Ubuntu):

Dynamic setting:

```
export http_proxy="http://185.46.212.88:9400"
export ftp_proxy="http://185.46.212.88:9400"
export https_proxy="http://185.46.212.88:9400"
export no_proxy=localhost,127.0.0.0/8,10.0.0.0/8,172.16.0.0/12,192.168.0.0/16,*.local, login.microsoftonline.com, login.microsoft.com, login.windows.net
```

Remove Dynamic Settings:

```
unset http_proxy
unset ftp_proxy
unset https_proxy
unset no_proxy
```

Make settings permanent:

```
sudo nano /etc/environment
```

```
http_proxy="http://185.46.212.88:9400"
ftp_proxy="http://185.46.212.88:9400"
https_proxy="http://185.46.212.88:9400"
no_proxy=localhost,127.0.0.0/8,10.0.0.0/8,172.16.0.0/12,192.168.0.0/16,*.local,login.microsoftonline.com,login.microsoft.com,login.windows.net
```

Routed Bypasses for Case 3 - Devices with PAC files.

In this case, you need to create a section on the PAC file to send "DIRECT" the traffic via the Routed Bypass.

For example:

```
// Routed Bypass for O365 Login destinations: 20.190.128.0/18 and 40.126.0.0/18
if ((isInNet(host, "20.190.128.0", "255.255.192.0") ||
      isInNet(host, "40.126.0.0", "255.255.192.0"))) {
   return "DIRECT";
}
```

In the following section, we are talking about all options when using PAC files.

10.1.3.3 PAC files: Solving requirements Case 3

In Case 3, the Virtual Desktops support PAC files. Due to both CSCs on the HA pair being active simultaneously, we can duplicate the bandwidth to Zscaler, achieving maximum throughput. In addition, using the Bypass functionality of the CSCs, it is possible to reach Trusted Sites via your Public IP and not Zscaler's IPs.

The CSC allows "Routed Bypasses" (Layer 4) and "Proxy Bypasses" (Layer 7).

As shown in previous pictures, the CSC has 3 Internal IPs:

CSC GW IP	Default Gateway for Routed traffic - On the PAC file, anything with "return DIRECT" will travel via this GW.
CSC VIP Proxy	Proxy to Zscaler.
CSC Bypass Proxy	Proxy for "Proxy Bypass" traffic.

PAC File for Virtual Desktops:

The following PAC file shows how to achieve 6.4 Gbps using the CSC Mux 3.2 Gbps as HA pair.

```
function FindProxyForURL(url, host) {
  // ==========================================================
  // Section 1: Zscaler standard PAC values

  var privateIP = /^(0|10|127|192\.168|172\.1[6789]|172\.2[0-9]|172\.3[01]|169\.254|192\.88\.99)\.[0-9.]+$/;
  var resolved_ip = dnsResolve(host);

  /* Don't send non-FQDN or private IP auths to us */
  if (isPlainHostName(host) || isInNet(resolved_ip, "192.0.2.0", "255.255.255.0") || privateIP.test(resolved_ip))
    return "DIRECT";

  /* FTP goes directly */
  if (url.substring(0, 4) == "ftp:")
    return "DIRECT";

  /* test with ZPA */
  if (isInNet(resolved_ip, "100.64.0.0", "255.255.0.0"))
    return "DIRECT";

  // ==========================================================
  // Section 2: Routed Bypass: Destination IPs / Networks going "DIRECT"
  // Routed Bypass for O365 Login destinations: 20.190.128.0/18 and 40.126.0.0/18
  if ((isInNet(host, "20.190.128.0", "255.255.192.0") ||
    isInNet(host, "40.126.0.0", "255.255.192.0"))) {
  return "DIRECT";
  }

  // ==========================================================
  // Section 3: Load Balancing: 2 x CSC Mux

  // Get NIC IP address
  nicIp = myIpAddress();

  // Assigning values to "tozscaler" and "bypass"
  if (isInNet(nicIp, "0.0.0.0", "0.0.0.1")) {
    var tozscaler = "PROXY csc1vip:80; PROXY csc2vip:80";
    var bypass = "PROXY csc1bypass:3128; PROXY csc2bypass:3128";
  }

  if (isInNet(nicIp, "0.0.0.1", "0.0.0.1")) {
    var tozscaler = "PROXY csc2vip:80; PROXY csc1vip:80";
    var bypass = "PROXY csc2bypass:3128; PROXY csc1bypass:3128";
```

```
  }

  // ==========================================================
  // Section 4: Proxy Bypass via Cloud Security Connectors

  // Proxy Bypass via CSC Public IPs (Examples)
  // Okta Domains (for Location Rules)
  if ((shExpMatch(host, "*.okta.com")) ||
    (shExpMatch(host, "*.oktacdn.com")) ||
    (shExpMatch(host, "*.okta-emea.com")) ||
    // Trusted Sites
    (shExpMatch(host, "trusted.domain.com")) ||
    (shExpMatch(host, "trusted2.domain.com")) ||
    (shExpMatch(host, "*.trusted-domain.com")) ||
    // O365 Domains for ConditionalAccess
    (shExpMatch(host, "login.microsoftonline.com")) ||
    (shExpMatch(host, "login.microsoft.com")) ||
    (shExpMatch(host, "login.windows.net")) ||
    // IP / Port test page
    (shExpMatch(host, "ip.maidenheadbridge.com"))) {
    return bypass;
  }

  // ==========================================================
  // Section 5: Default Traffic

  /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
  return tozscaler;
}
```

Sections Explained:

Section 1:	Zscaler Standard PAC values to: Do not send Private IPs to Zscaler or ZPA or FTP traffic.
Section 2:	Shows Destinations IP / Networks sent DIRECT that you want to reach using your Public IP using Routed Bypasses. (L4)
Section 3:	Section 2 does the load balancing between both CSC on the HA pair. As you can see, we are reading the source IP of the VDI (nicIp = myIpAddress();), and we are load balancing by odd/even IP, using different primary/secondary cscvip and cscbypass for odd/even IPs.
Section 4:	Section 4 shows examples of URLs to bypass from Zscaler to reach the destination website with your Public IP. Examples of required URLs for OKTA (for Location rules) and O365 (for Conditional Access) are shown. One common use of these examples is not asking for MFA (Multi-factor Authentication) for VDIs.
Section 5:	Default traffic will go via Zscaler.

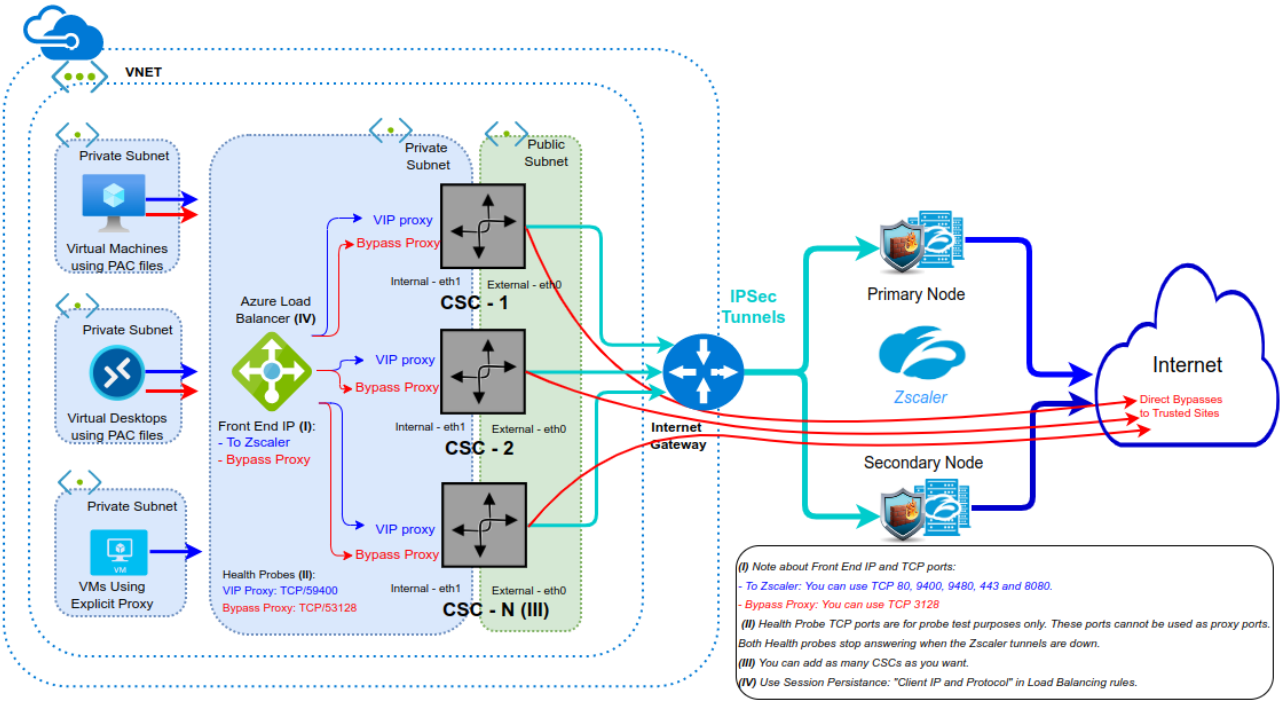
Proxy Bypass: PAC file for the CSCs

```
function FindProxyForURL(url, host) {
   // This value of bypass on the PAC file for the CSC can be any.
   // We need to assigned a value just to pass the "Validation" of the PAC on Zscaler console.
   var bypass = "PROXY 1.1.1.1:3128; PROXY 2.2.2.2:3128";

   // ==========================================================
   // Section 3: Bypass via Cloud Security Connectors

   // Bypass via CSC Public IPs (Examples)
   // Okta Domains (for Location Rules)
   if ((shExpMatch(host, "*.okta.com")) ||
      (shExpMatch(host, "*.oktacdn.com")) ||
      (shExpMatch(host, "*.okta-emea.com")) ||
      // Trusted Sites
      (shExpMatch(host, "trusted.domain.com")) ||
      (shExpMatch(host, "trusted2.domain.com")) ||
      (shExpMatch(host, "*.trusted-domain.com")) ||
      // O365 Domains for ConditionalAccess
      (shExpMatch(host, "login.microsoftonline.com")) ||
      (shExpMatch(host, "login.microsoft.com")) ||
      (shExpMatch(host, "login.windows.net")) ||
      // IP / Port test page
      (shExpMatch(host, "ip.maidenheadbridge.com"))) {
      return bypass;
      }

return bypass

}
```

10.2 CSC Mux in HA with Azure Load Balancer

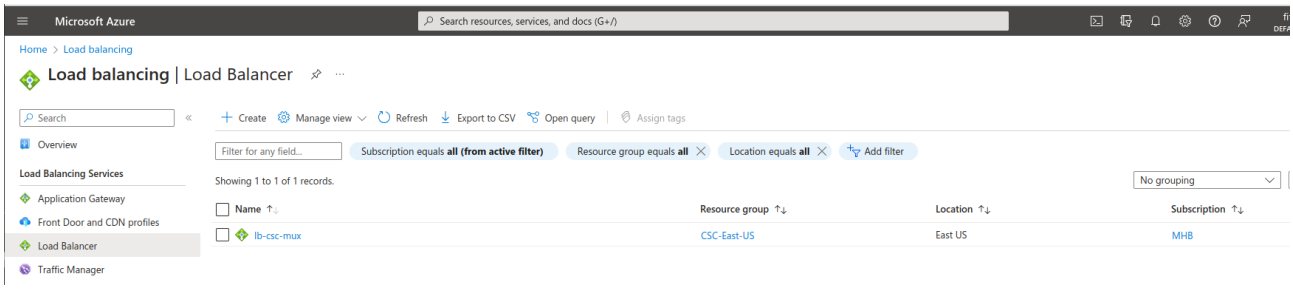
The CSC Mux can be balanced using Azure Load Balancer. In this chapter, we will present how to use the Azure Load Balancer's Front End IPs as a Proxy for traffic to Zscaler and Proxy Bypass.

10.2.1 Network Diagram



10.2.2 Azure Load Balancer configuration

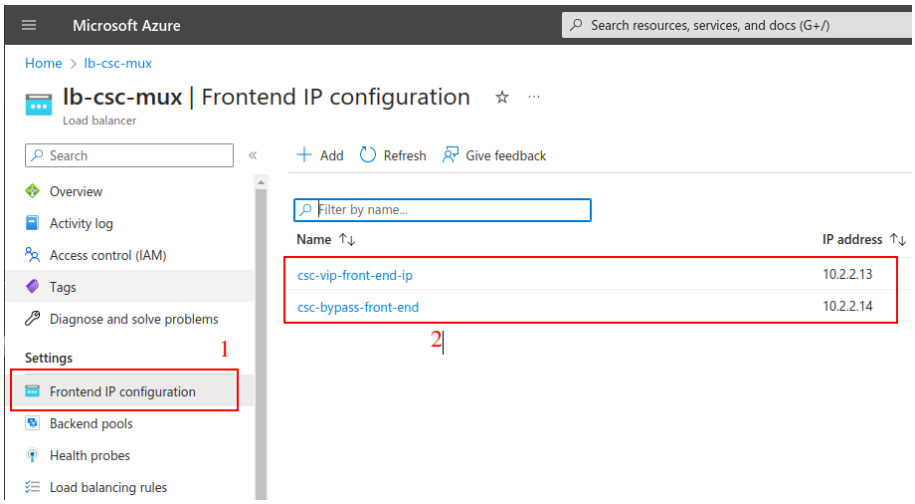
10.2.2.1 Create a Standard Load Balancer



10.2.2.2 Front End IP Configuration

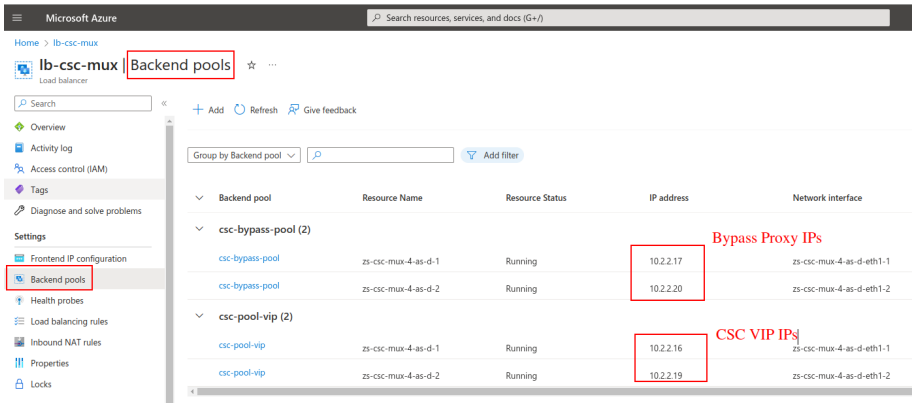
The first task is to create the Front End IP. In this example, we will create two: One for traffic "to Zscaler" and another for "Proxy Bypasses".

IMPORTANT: *The Front End IPs must be located in the Internal Subnet of the CSC.*



10.2.2.3 Backend pools

Create a Backend with the CSC VIPs and another with the CSC Bypass Proxy IPs.

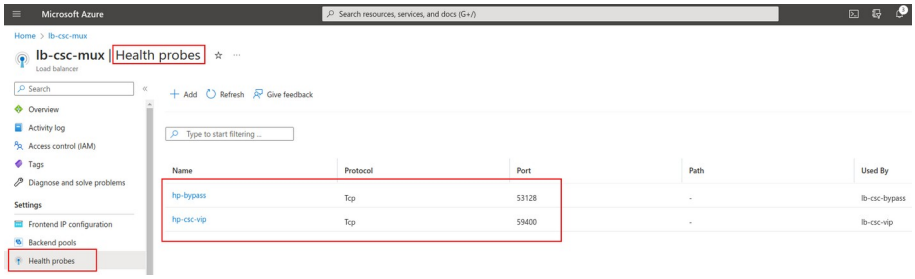


10.2.2.4 Health Probes

You need to create a Health Probe for the CSC VIPs and another for the Proxy Bypasses.

- CSC VIP Health Probe: TCP / 59400
- CSC Proxy Bypass Health Probe: TCP / 53128

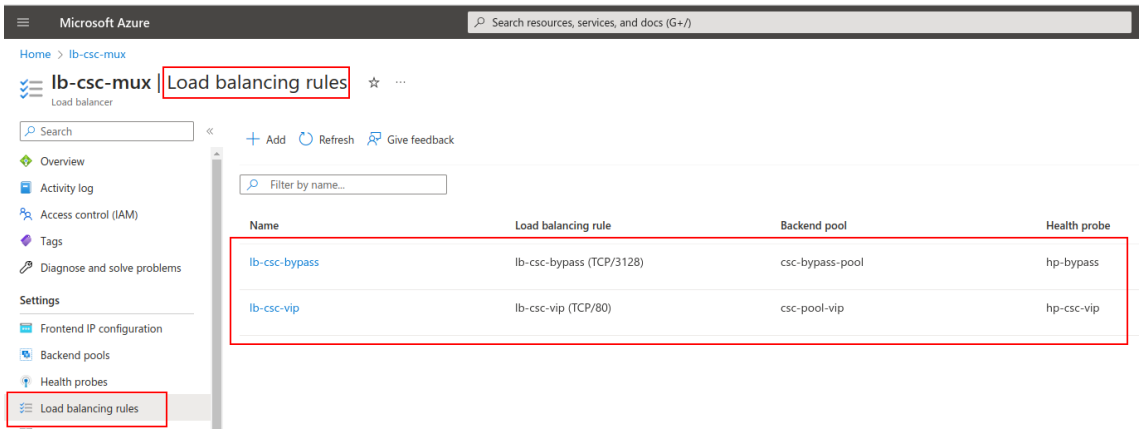
IMPORTANT: *Health Probe TCP ports are for probe test purposes only. These ports cannot be used as proxy ports. Both Health probes stop answering when the Zscaler tunnels are down.*



10.2.2.5 Load Balancing rules

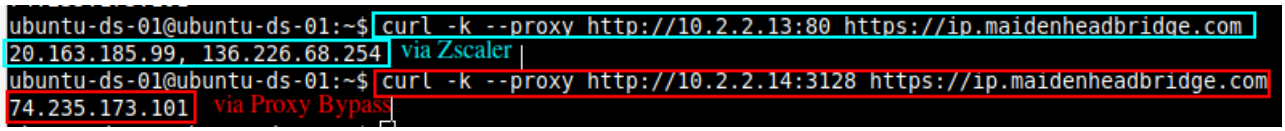
On the Load Balancing rules you need to associate the Front End, Health Probes, Backend Pools and to define the Port to use. In this case, you need to create two Load Balance Rules, one for the CSC VIP and another for the Proxy Bypass. The TCP ports allowed are:

- To Zscaler: You can use TCP 80, 9400, 9480, 443 and 8080.
- Bypass Proxy: You can use TCP 3128

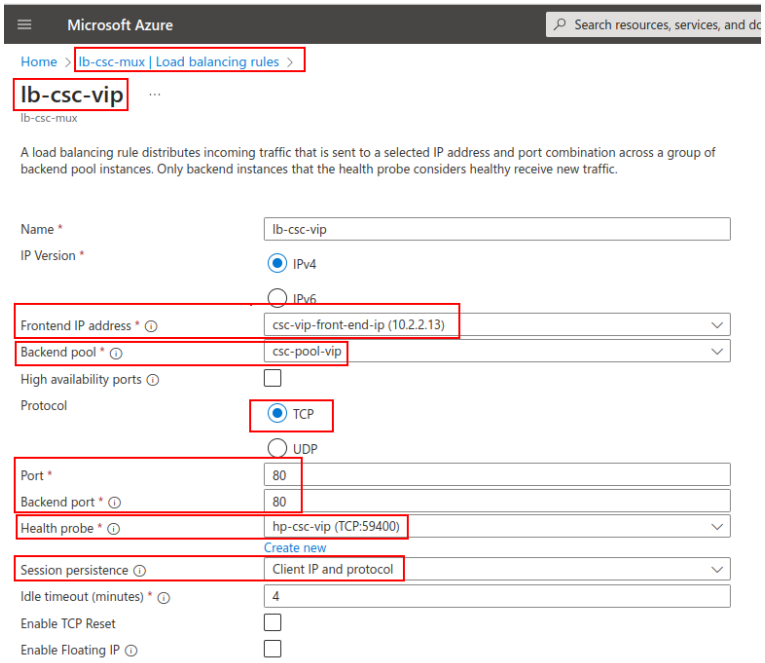


IMPORTANT: *Use Session Persistence "Client IP and Protocol"*

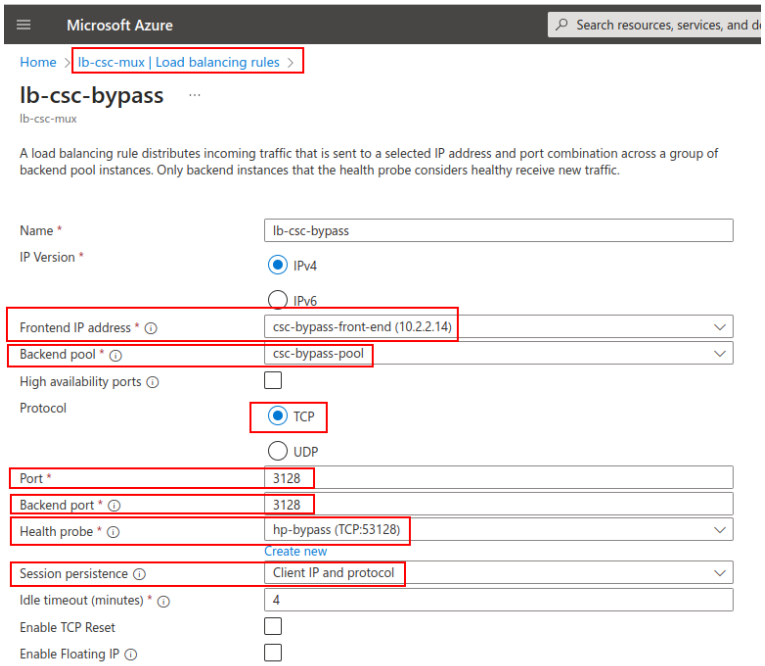
Test using CURL command:



10.2.2.5.1 Load Balancing Rule for CSC VIP in detail.



10.2.2.5.2 Load Balancing Rule for Proxy Bypass in detail.

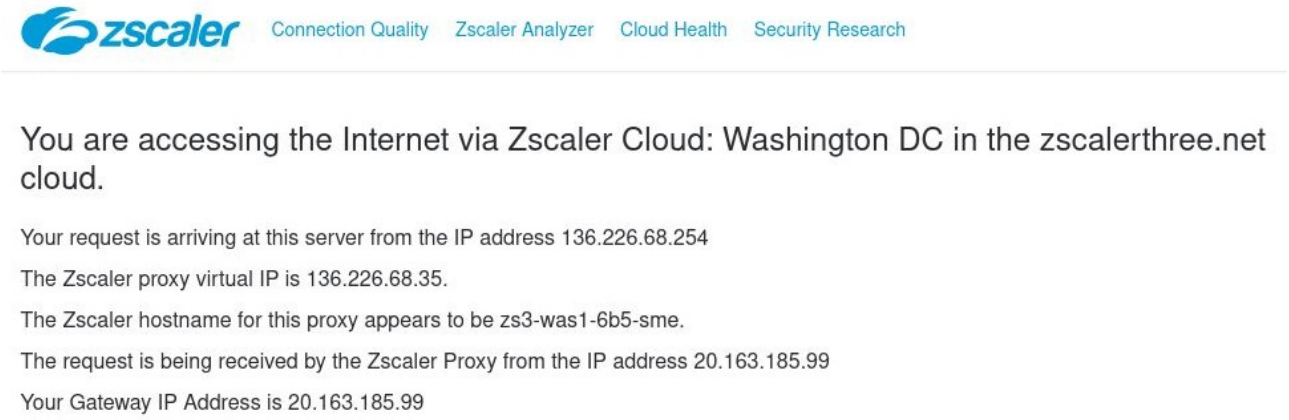


10.3 Testing traffic to Zscaler

10.3.1 ip.zscaler.com

The page ip.zscaler.com shows the Location values.

Using the browser:



Using curl command from CMD or Terminal

Proxy environment:

Command	Windows: curl -s --proxy http://**<CSC VIP>**:80 ip.zscaler.com | findstr "You" Linux: curl -s --proxy http://**<CSC VIP>**:80 ip.zscaler.com | grep "You"
Expected Result	ubuntu-ds-01@ubuntu-ds-01:~$ curl -s --proxy http://10.2.2.13:80 ip.zscaler.com | grep "You" You are accessing the Internet via Zscaler Cloud: Washington DC in the zscalerthree.net cloud. Your request is arriving at this server from the IP address 136.226.68.254 Your Gateway IP Address is 20.163.185.99

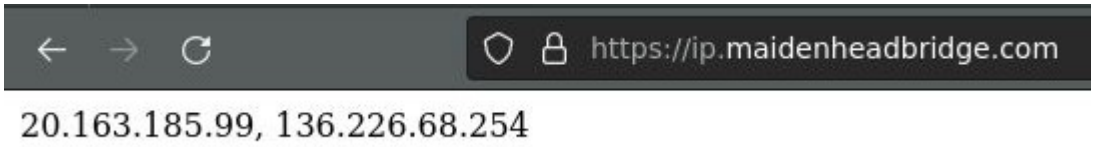
Routed environment:

Command	Windows: curl -s ip.zscaler.com | findstr "You" Linux: curl -s ip.zscaler.com | grep "You"
Expected Result	ubuntu-ds-01@ubuntu-ds-01:~$ curl -s ip.zscaler.com | grep "You" You are accessing the Internet via Zscaler Cloud: Washington DC in the zscalerthree.net cloud. Your request is arriving at this server from the IP address 136.226.69.11 Your Gateway IP Address is 20.163.185.99

10.3.2 https://ip.maidenheadbridge.com

Maidenhead Bridge provides a HTTPS page to check the IP.

Using the Browser:



Using curl command from CMD or Terminal

(Note: the switch "-k" on curl command is to avoid SSL certificate validation)

Proxy environment:

Command	curl -k --proxy http://**<CSC VIP>**:80 https://ip.maidenheadbridge.com
Expected Result	`ubuntu-ds-01@ubuntu-ds-01:~$ curl -s -k --proxy http://10.2.2.13:80 https://ip.maidenheadbridge.com` `20.163.185.99, 136.226.68.254`

Routed environment:

Command	curl -k https://ip.maidenheadbridge.com
Expected Result	`ubuntu-ds-01@ubuntu-ds-01:~$ curl -s -k https://ip.maidenheadbridge.com` `20.163.185.99, 136.226.69.5`

10.3.3 SpeedTest

The CSC contains the SpeedTest client. You can run it from the SSH console or using any Management tool (AWS Systems Manager, Rundeck, , Ansible, etc.)

```
Monitoring Tasks
1) Show Configuration and Status
2) Show Interfaces Traffic
3) Tcpdump, Traceroute and Latency Test
4) Speed Test (Experimental)
```

This test is experimental because we use third-party tools (speedtest.net), but it works fine in most cases. Here the result using a CSC Mux 4.

```
Selection: 4

SPEED TEST
This is experimental. We are using third party tools. (Speedtest.net)
Results can be inaccurate or none. The test takes a while


Aggregated Bandwidth Download:  1905.09 Mbps
```

Note: At the moment of writing this documentation, Zscaler provides 400 Mbps per IPsec tunnel. The CSC Mux 4 can aggregate 4 x IPsec tunnels (~ 1.6 Gbps total).

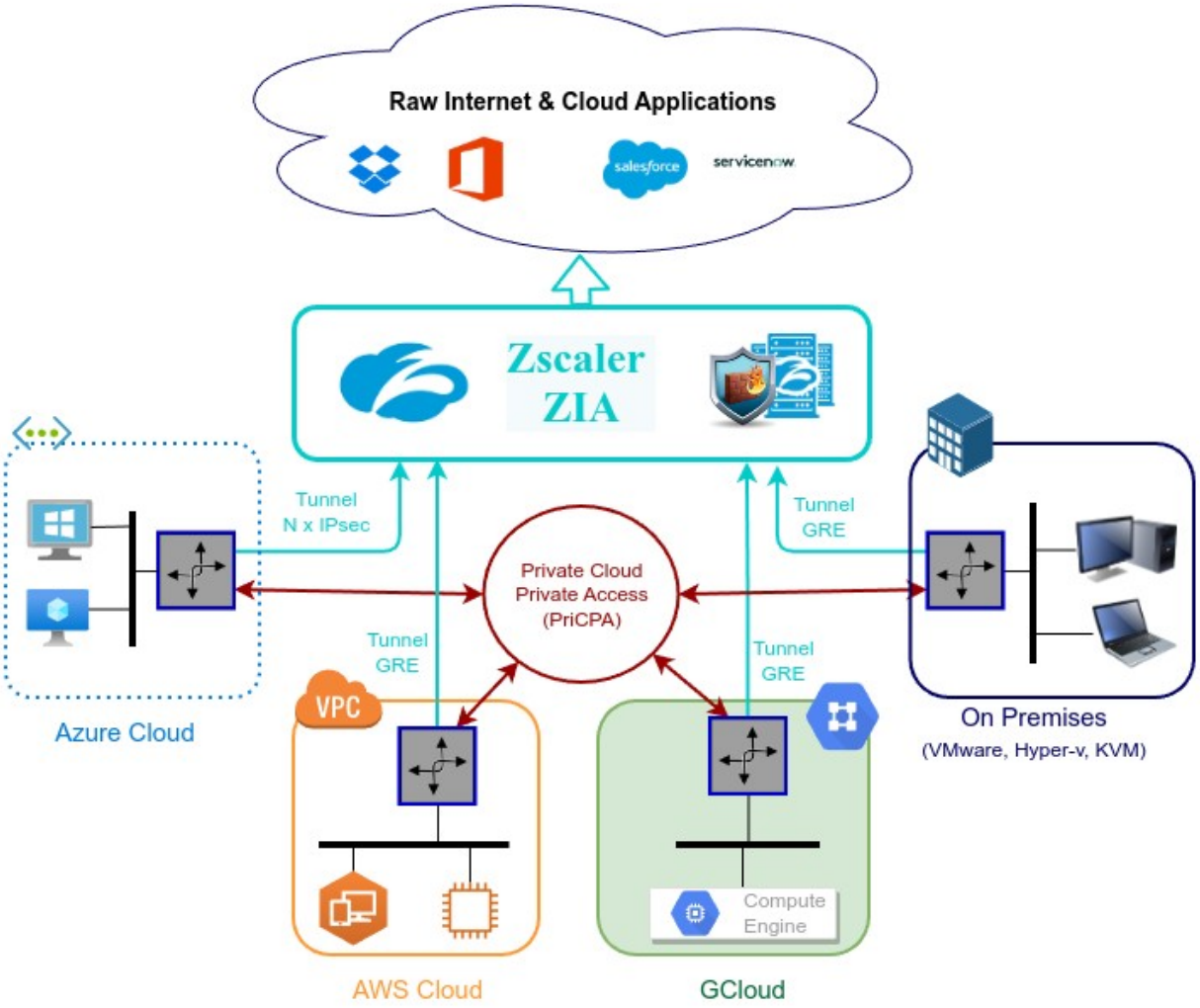
11 Private Cloud Private Access

11.1 What is Private Cloud Private Access (PriCPA)?

Private Cloud Private Access (PriCPA) is a new functionality of the Cloud Security Connector. PriCPA allows you to create a Private Cloud among all CSCs for private traffic. In a matter of minutes, you can build a full mesh encrypted topology between your locations for private traffic with Zero Trust. After making the Private Cloud, you can set up your policies to define who will talk with who inside your Private Cloud.

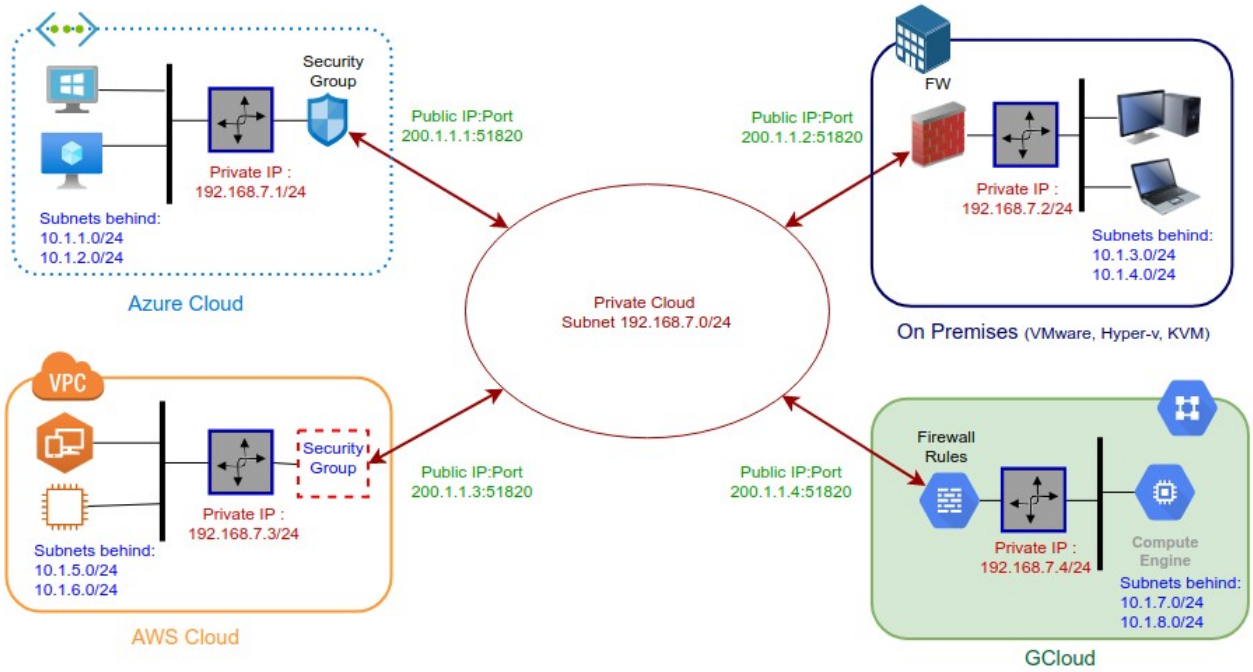
11.2 PriCPA Network Diagrams

11.2.1 High Level Network Diagram



11.2.2 Low Level Network Diagram – PriCPA only

The following network diagram shows the IP addressing for PriCPA.



Steps to design your Private Cloud:

1. Select a Subnet for your Private Cloud. The example above is 192.168.7.0/24. Due to the Subnet is /24, up to 255 CSCs can participate in this Private Cloud.
2. Assign a Cloud Private IP to each CSC. In this example, we are assigning 192.168.7.1 to 192.168.7.4
3. The Public IP to be used will be the same assigned to the Bypass of each CSC. You can choose the UDP port to use at each location. For simplicity, it is recommended to use the same port at all locations.
4. Gather the information of the private Subnets behind each CSC. This information will be required when configuring the Peers.
5. Firewall Rules (or Security Groups Rules): The CSC for Azure, AWS and Gcloud will implement the firewall rules automatically. Manual FW rules are required when the CSC is "On-Premises". The CSC provides a JSON file with the rules required.

11.3 Configuring PriCPA

The Main Menu has a section for Private Access:

```
Private Cloud Private Access (PriCPA)
17) Show Configuration and Status PriCPA.
18) Configure PriCPA (Local and Peers Configuration).
19) Configure CSC Remote Management via PriCPA.
```

The configuration of PriCPA is four simple steps:

```
Selection: 18

Private Access Configuration Wizard

_____________________________________________________________

Steps to configure Private Access:

A) Assign 'Identity' to the VM:
   A.1) Go to 'Identity -> System Assigned' and 'Turn ON' status. (and Save).
   A.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:
       -> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>
       -> Role: Contributor, Resource Group: <Route Tables Resource Group/s> (optional, but required for HA)
       -> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>
B) Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
C) (optional, if HA is enabled.) Copy Local Configuration to the other CSC in the HA pair.
D) Load Private Access Peers JSON configuration file.
_____________________________________________________________

1) Create (or change) Private Access Local Configuration
2) Load Private Access Peers JSON configuration file
3) Quit
Enter your choice:
```

1. Assign "Identity" to the CSC (and to the "other CSC" if HA is enabled).
2. Create the Local Node configuration. This step will initialize and enable Private Access on the Node. The result of this operation will show a "Token" and "Private Access Local JSON file".
3. (HA Pair only) Initialize the second Node of the HA pair using the "Token" and "Private Access Local JSON file".
4. Create and distribute the Private Access Peers JSON file to all nodes.

IMPORTANT: *We strongly recommend using software with a JSON formatter to create the Peers JSON file, like Visual Code or Notepad ++ . See Appendix C for more detail about how to install these programs and the plugins required.*

11.3.1 Create the Local configuration (First node of the HA pair or Single deployment)

11.3.1.1 Using configUserData.json file

You can pass the Local configuration parameters via configUserData.json file during the initial deployment.

Here an example:

```
    "priCPA": {
        "enable": "yes",
        "nodeName": "zs-csc-mux-4-as-d",
        "location": "Azure US East",
        "description": "CSC MUX 4 AS D",
        "publicUdpPort": "51280",
        "privateCirdIp": "192.168.7.16/24",
        "persistentKeepAlive": "no",
        "peersJsonFileUrl": "https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
        "remoteManagementNetworks": [
            "172.19.0.0/24",
            "192.168.1.0/24",
            "192.168.6.0/24"
        ]
    },
```

The CSC will read this information and create the Local configuration on the First node of the HA deployment or when it is a single deployment. As a result, the Local Configuration for PriCPA will be ready:

```
Enter your choice: 1

Identity Type: SystemAssigned

Private Access is enabled.

The current values configured are:

Please, copy the following values in a safe place to configure the other CSC on the High Availability Pair. Discard this message if you are doing a single deployment.

____________________________________________________________

Token: OEZYWE9WdHFaOFdHQlJ0eWZESndnYnVmYlFmNlkvaHQvNndPQ0sxQ2RWZz0K

Private Access Local Config JSON file:

{
  "peers": [
    {
      "nodeName": "zs-csc-mux-4-as-d",
      "location": "Azure US East",
      "description": "CSC MUX 4 AS D",
      "publicKey": "ecs6WnGUrS1Dey8FPppWk9Q8MjVj9w7+bBeaH+MXoGY=",
      "publicIpAndUdpPort": "74.235.173.101:51280",
      "privateCirdIp": "192.168.7.16/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
____________________________________________________________

Please, select next action:

1) Change configuration
2) Reset to default values
3) Restart Service
4) Quit
Enter your choice:
```

11.3.1.2 Manual Configuration

→ From Main Menu, select "18) Configure Private Access."

→ Select "1) Create (or change) Private Access Local Configuration"

```
Selection: 18

Private Access Configuration Wizard

_____________________________________________________________

Steps to configure Private Access:

A) Assign 'Identity' to the VM:
   A.1) Go to 'Identity -> System Assigned' and 'Turn ON' status. (and Save).
   A.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:
       -> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>
       -> Role: Contributor, Resource Group: <Route Tables Resource Group/s> (optional, but required for HA)
       -> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>
B) Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
C) (optional, if HA is enabled.) Copy Local Configuration to the other CSC in the HA pair.
D) Load Private Access Peers JSON configuration file.
_____________________________________________________________

1) Create (or change) Private Access Local Configuration
2) Load Private Access Peers JSON configuration file
3) Quit
Enter your choice:
```

→ Select "1) Manual Configuration" and input the values requested.

```
Do you want to enable Private Access?

1) Manual Configuration
2) Token and JSON
3) Quit
Enter your choice 1

Before continuing, you need to have the following values ready:

   - Node Name. (string)
   - (Optional) Location Name. (string)
   - (Optional) Description. (string)
   - Public IP and UDP Port. (IP:Port)
   - Private IP/Subnet of Local Interface. (IP/Subnet Prefix)

Do you want to continue?

1) Yes
2) No
Enter your choice: 1

Please, input the following values:

Node Name (string): zs-csc-mux-4-as-d
(Optional) Location Name (string): Azure US East
(Optional) Description (string): CSC MUX 4 AS D
Public IP and UDP port (IP:Port): 74.235.173.101:51280
Private IP/Subnet of Local Interface (IP/Subnet Prefix): 192.168.7.16/24

Persistent KeepAlive setting:

-> Persistent KeepAlive is required in rare cases:
    a) When the firewall of this site cannot do an outbound NAT without changing the source port.
    b) When incoming connections are not possible at all to this site.

IMPORTANT: We strongly recommend keeping the default value of 'Persistent KeepAlive = no'. Enabling 'Persistent KeepAlive' generates unnecessary traffic and consumes CPU resources.

Do you want to change default value of 'Persistent KeepAlive = no'?

1) Yes
2) No (Recommended)
Enter your choice: 2

The values to configure are:

Node Name: zs-csc-mux-4-as-d
Public IP and UDP Port: 74.235.173.101:51280
Private IP/Subnet of Local Interface: 192.168.7.16/24
Location Name: Azure US East
Description: CSC MUX 4 AS D
Persistent KeepAlive: no

Do you want to apply this values?
```

➢ Apply values

```
Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1

(MHB-CSC)(INFO) Private Access - Private Access service is enabled on zs-csc-mux-4-as-d-1.

Please, copy the following values in a safe place to configure the other CSC on the High Availability Pair. Discard this message if you are doing a single deployment.

Token: YU9WVk9zSUdkRVNnb1UzVmZNZXZISDIvU2RYSWVpdHFpOG01aTAyU04zTT0K

Private Access Local Config JSON file:

{
  "peers": [
    {
      "nodeName": "zs-csc-mux-4-as-d",
      "location": "Azure US East",
      "description": "CSC MUX 4 AS D",
      "publicKey": "4OJ7QPswdTx+mrLMbgLBube0/rw9sSunY78OkljTZ1g=",
      "publicIpAndUdpPort": "74.235.173.101:51280",
      "privateCirdIp": "192.168.7.16/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

Token

Local JSON

IMPORTANT: *The "Token" and "Private Access Local Config JSON file" will be used to create the local configuration on the second node of the HA pair. Please, keep these values in a safe place. You can use these values to reconfigure any node of the HA Pair if necessary in the future. For example, if you want to change the IPs or descriptions.*

11.3.2 Create the Local configuration (second node of HA Pair)

SSH the second node of the HA Pair and input the "Token" and "Private Access Local Config JSON file".

Go to 18) Configure Private Access. → 1) Create (or change) Private Access Local Configuration → 2) Token and JSON

```
Selection: 18

Private Access Configuration Wizard

_____________________________________________________________

Steps to configure Private Access:

A) Assign 'Identity' to the VM:
   A.1) Go to 'Identity -> System Assigned' and 'Turn ON' status. (and Save).
   A.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:
       -> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>
       -> Role: Contributor, Resource Group: <Route Tables Resource Group/s> (optional, but required for HA)
       -> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>
B) Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
C) (optional, if HA is enabled.) Copy Local Configuration to the other CSC in the HA pair.
D) Load Private Access Peers JSON configuration file.
_____________________________________________________________

1) Create (or change) Private Access Local Configuration
2) Load Private Access Peers JSON configuration file
3) Quit
Enter your choice: 1

Identity Type: SystemAssigned

Private Access is not enabled.

IMPORTANT:
   1) Use 'Manual Configuration' to generate keys and values.
   2) Use 'Token and JSON' to load previous generated values. For example, to configure second CSC on HA Pair.

Do you want to enable Private Access?

1) Manual Configuration
2) Token and JSON
3) Quit
Enter your choice: 2

Before continuing, you need to have ready the values generated on the Fist CSC on the HA Pair.:

    1 - Token (string)
    2 - Private Access Local Config JSON file. (JSON File)

Do you want to continue?

1) Yes
2) No
Enter your choice: 1
```

```
Do you want to continue?

1) Yes
2) No
Enter your choice: 1

Please, input the following values:

Token (string): YU9WVk9zSUdkRVNnb1UzVmZNZXZISDIvU2RYSWVpdHFpOG01aTAyU04zTT0K

Please, paste 'Private Access Local Config JSON file' and press 'Enter' if required.

NOTE: If the json file has errors, it is possible that the script will hang. Press '}' and 'Enter' to end the operation.

Private Access Local Config JSON file: {
  "peers": [
    {
      "nodeName": "zs-csc-mux-4-as-d",
      "location": "Azure US East",
      "description": "CSC MUX 4 AS D",
      "publicKey": "4QJ7QPswdTx+mrLMbgLBube0/rw9sSunY78OkljTZ1g=",
      "publicIpAndUdpPort": "74.235.173.101:51280",
      "privateCirdIp": "192.168.7.16/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}

Private Access Local Config JSON file imported successfully


The values to configure are:

Node Name: "zs-csc-mux-4-as-d"
Public IP and UDP Port: 74.235.173.101:51280
Private IP/Subnet of Local Interface: 192.168.7.16/24
Location Name: "Azure US East"
Description: "CSC MUX 4 AS D"
Persistent KeepAlive: no

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1

(MHB-CSC)(INFO) Private Access - Private Access service is enabled on zs-csc-mux-4-as-d-2.
```

11.3.3 Create the Private Access Peers JSON file

The Private Access Peers JSON file contains:

1. The Local configuration of each Peer.
2. The "networks" behind each Peer.
3. The "privateApps" allowed to be reached on each Peer.

Here some examples.

11.3.3.1 Full mesh Private Access Peers JSON file

Consider the following example:

We have 3 nodes and we want to allow full communication between sites for all port and protocols.

The Local Config JSON file of each node is:

ns-cgc00001

```
{
  "peers": [
    {
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
      "publicIpAndUdpPort": "200.1.1.1:51821",
      "privateCirdIp": "192.168.7.1/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

ns-cgc00002

```
{
  "peers": [
    {
      "nodeName": "ns-cgc00002",
      "description": "Node on VMware Server 2",
      "location": "Datacentre 2",
      "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
      "publicIpAndUdpPort": "200.1.1.2:51821",
      "privateCirdIp": "192.168.7.2/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

ns-cgc00003


```
{
  "peers": [
    {
      "nodeName": "ns-cgc00003",
      "description": "Node on VMware Server 3",
      "location": "Branch",
      "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```


Firstly, we need to create our "basic" Peers Configuration JSON file: It contains the Local Configuration of each Node plus the "networks" behind each node.

Basic Peers Configuration JSON file


```
{
  "peers": [
    {
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
      "publicIpAndUdpPort": "200.1.1.1:51821",
      "privateCirdIp": "192.168.7.1/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.1.1.0/24",
        "10.1.2.0/24"
      ],
      "privateApps": []
    },
    {
      "nodeName": "ns-cgc00002",
      "description": "Node on VMware Server 2",
      "location": "Datacentre 2",
      "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
      "publicIpAndUdpPort": "200.1.1.2:51821",
      "privateCirdIp": "192.168.7.2/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.2.1.0/24",
        "10.2.2.0/24"
      ],
      "privateApps": []
```

```
    },
    {
      "nodeName": "ns-cgc00003",
      "description": "Node on VMware Server 3",
      "location": "Branch",
      "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.3.1.0/24",
        "10.3.2.0/24"
      ],
      "privateApps": []
    }
  ]
}
```


In this "Basic Peers Configuration JSON file" we have:

- Green: The Local values generated at each node.
- Yellow: The Subnets behind each node
- Red: Nothing. No private Apps configured.

If you deployed this "Basic Peers Configuration JSON file" to all CSCs, you have created the Private Cloud. All Peers will be visible to each other, but no traffic between subnets will be allowed because there is no "privateApps" configured.

If we want to allowed traffic any to any between subnets, we need to add the corresponding "privateApps" to each node. For example for node: "ns-cgc00001"

ns-cgc00001


```
{
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
      "publicIpAndUdpPort": "200.1.1.1:51821",
      "privateCirdIp": "192.168.7.1/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.1.1.0/24",
        "10.1.2.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
```

```
            "sourceCirdIp": [
              "0.0.0.0/0"
            ],
            "destinationCirdIp": [
              "10.1.1.0/24",
              "10.1.2.0/24"
            ],
            "destinationSinglePorts": [
              ""
            ],
            "destinationPortRange": {
              "fromPort": "",
              "toPort": ""
            }
          }
        ]
      },
```

In this case, we added a "privateApp" that allows any source IPs (0.0.0.0/0) to reach the "networks" (10.1.1.0/24 and 10.1.2.0/24) using "all" protocols ("ipProtocol" : "all".)

Now, completing our "Peers Configuration JSON file":

Full Mesh Peers Configuration JSON file.

```
{
  "peers": [
    {
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
      "publicIpAndUdpPort": "200.1.1.1:51821",
      "privateCirdIp": "192.168.7.1/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.1.1.0/24",
        "10.1.2.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [
            "0.0.0.0/0"
          ],
          "destinationCirdIp": [
            "10.1.1.0/24",
            "10.1.2.0/24"
          ],
          "destinationSinglePorts": [
            ""
          ],
          "destinationPortRange": {
            "fromPort": "",
            "toPort": ""
          }
        }
      ]
    },
    {
      "nodeName": "ns-cgc00002",
      "description": "Node on VMware Server 2",
      "location": "Datacentre 2",
      "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
      "publicIpAndUdpPort": "200.1.1.2:51821",
      "privateCirdIp": "192.168.7.2/24",
      "persistentKeepAlive": "no",
      "networks": [
```

```
        "10.2.1.0/24",
        "10.2.2.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [
            "0.0.0.0/0"
          ],
          "destinationCirdIp": [
            "10.2.1.0/24",
            "10.2.2.0/24"
          ],
          "destinationSinglePorts": [
            ""
          ],
          "destinationPortRange": {
            "fromPort": "",
            "toPort": ""
          }
        }
      ]
    },
    {
      "nodeName": "ns-cgc00003",
      "description": "Node on VMware Server 3",
      "location": "Branch",
      "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.3.1.0/24",
        "10.3.2.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [
            "0.0.0.0/0"
          ],
          "destinationCirdIp": [
            "10.3.1.0/24",
            "10.3.2.0/24"
          ],
          "destinationSinglePorts": [
            ""
          ],
          "destinationPortRange": {
            "fromPort": "",
            "toPort": ""
          }
        }
      ]
    }
  ]
}
```


Done! Your task is to implement this JSON file on all CSCs and you will have full connectivity any to any for all protocols.

11.3.3.2 Understanding "privateApps" configuration and values

Question 1: Where to configure the "privateApps"?

Only on the node that has the "destinationCirdIp": [], that belongs to its "networks".

Example: I want to allow access to "destinationCirdIp": ["**10.1.1.50/32**"]. The rule must be created on node ns-cgc00001 that has "networks": ["**10.1.1.0/24**","10.1.2.0/24"]

Question 2 : What about the values to configure?

On "privateApps" section there are two types of values to input:

Accepts single value only -> ""

Accepts single or multiple values -> []

```
"privateApps": [
        {
          "description": "",
          "ipProtocol": "",
          "sourceCirdIp": [],
          "destinationCirdIp": [],
          "destinationSinglePorts": [],
          "destinationPortRange": {
            "fromPort": "",
            "toPort": ""
          }
        }
      ]
```

Examples:

<u>Single value (""):</u>

"description": " Intranet Servers",
"ipProtocol": "tcp",

<u>Single or Multiple values ([]):</u>

"sourceCirdIp": ["0.0.0.0/0"],

"destinationCirdIp": ["10.1.1.100/32", "10.1.2.100/32"],
"destinationSinglePorts": ["80", "443"],

The following table shows all fields and values accepted:

Field	Value Type	Values to configure	Example
"description": "",	Single	String	"description": "Intranet Server Access",
"ipProtocol": "",	Single	tcp,udp,icmp or all	"ipProtocol": "tcp",
"sourceCirdIp": [],	Single or Multiple	Networks in the range of: 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 and 0.0.0.0/0	"sourceCirdIp": ["10.2.1.0/24", "10.2.2.0/24", "10.3.1.0/24", "10.3.2.0/24"],
"destinationCirdIp": [],	Single or Multiple	Networks in the range of [4]: 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	"destinationCirdIp": ["10.1.1.100/32", "10.1.1.200/32"],
"destinationSinglePorts": [],	Single or Multiple	Single Port of the range 1 to 65535	"destinationSinglePorts": ["80", "443"],

"destinationPortRange": {
"fromPort": "",
"toPort": ""
}
	Single	Single Port of the range 1 to 65535	"destinationPortRange": {
			"fromPort": "3780",
			"toPort": "3784"
			}

IMPORTANT: *For PriCPA, 0.0.0.0/0 represent the private network segments: 10/8, 172.16/12, 192.168/16 and not the entire internet addresses.*

4 The expected value here is a value that belongs to the "network" defined behind the CSC. For example, of the network behind the CSC is 10.1.1.0/24, any destination configured must belong to 10.1.1.0/24, like 10.1.1.100/32.

11.3.3.3 Example of "privateApps" for a Windows Domain controller

The following example shows how to create rules to allow access to your Domains Controllers.

The port information was taken from this article:

https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/config-firewall-for-ad-domains-and-trusts

Example: Domain Controllers IPs: "10.2.1.100/32" and "10.2.2.100/32" on Node ns-cgc00002 of previous example

```
"privateApps": [
        {
          "description": "Domain Controllers TCP",
          "ipProtocol": "tcp",
          "sourceCirdIp": [ "0.0.0.0/0" ],
          "destinationCirdIp": [ "10.2.1.100/32", "10.2.2.100/32" ],
          "destinationSinglePorts": [ "135", "464", "389", "636", "3268", "3269", "53", "88", "445" ],
          "destinationPortRange": { "fromPort": "49152", "toPort": "65535" }
        },
        {
          "description": "Domain Controllers UDP",
          "ipProtocol": "udp",
          "sourceCirdIp": [ "0.0.0.0/0" ],
          "destinationCirdIp": [ "10.2.1.100/32", "10.2.2.100/32" ],
          "destinationSinglePorts": [ "123", "464", "389", "53", "88" ],
          "destinationPortRange": { "fromPort": "", "toPort": "" }
        },
        {
          "description": "Domain Controllers Ping",
          "ipProtocol": "icmp",
          "sourceCirdIp": [ "0.0.0.0/0" ],
          "destinationCirdIp": [ "10.2.1.100/32", "10.2.2.100/32" ],
          "destinationSinglePorts": [],
          "destinationPortRange": { "fromPort": "", "toPort": "" }
        }
      ]
```

11.3.3.4 Example of "privateApps" for Internal Web Server.

In this example, we are showing how to configure access to users on ns-cgc00001 to an Internal Web server located behind node Node ns-cgc00003.

Example: Web Server "10.3.1.200/32" on Node ns-cgc00003 of previous example. Allow access to all users behind ns-cgc00001

```
"privateApps": [
        {
          "description": "Web Server 3",
          "ipProtocol": "tcp",
          "sourceCirdIp": [ "10.1.1.0/24", "10.1.2.0/24" ],
          "destinationCirdIp": [ "10.3.1.200/32" ],
          "destinationSinglePorts": [ "80", "443" ],
          "destinationPortRange": { "fromPort": "", "toPort": "" }
        }
      ]
```

11.3.4 Load the "Private Access Peers JSON file" to the CSCs.

After the Local Configuration is done and the "Private Access Peers JSON file" is created, the next task is to distribute and apply it on each CSC.

There are three methods available:

1. URL: (Recommended) Using "Private Access Peers URL" and running the command "Refresh Private Access Peers URL" using AWS Systems Manager, Rundeck or Azure CLI commands.
2. DevOps: Distribute the JSON file on all CSC and run the command "Reload Private Access Peers URL" using AWS Systems Manager or Rundeck.
3. Manual: Copy/Paste the JSON file on each CSCs.

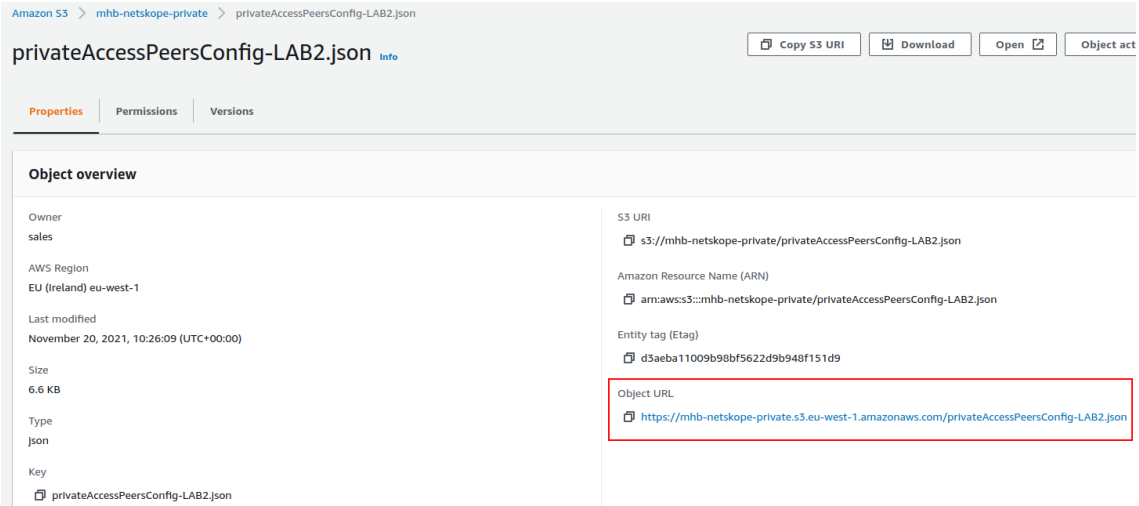
In this section we are going to explain two methods: URL and Manual Copy. The DevOps method is explained on Section12: DevOps operations.

11.3.4.1 Using "Private Access Peers URL"

This is the recommended method. The steps to configure are:

1. Place the Private Access Peers JSON file on an internal web server or an AWS bucket[5] or similar. Obtain the download URL.

Example of AWS bucket:



2. Configure the URL on each CSC.

Ssh the each CSC and go to Main Menu -> 18) Configure Private Access

5 See Appendix D to learn how to secure an AWS S3 bucket by Source IP.

```
1) Create (or change) Private Access Local Configuration
2) Load Private Access Peers JSON configuration file
3) Quit
Enter your choice: 2

Private Access is enabled.

Please, Select Method:
1) Private Access Peers URL
2) Manual (Paste Private Access Peers JSON File)
3) Reset to Default Values
4) Quit
Enter your choice: 1

*** Private Acces Peers URL is not configured ***

Do you want to configure the Private Acces Peers URL?
1) Yes
2) No
Enter your choice: 1

Please, input Private Acces Peers URL
Private Acces Peers URL: https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json

Do you want to refresh the Private Access Peers List?
1) Yes
2) No
Enter your choice: 1

Private Access Peers JSON file imported successfully


You can review your values before to apply. Please, Select 'Compact' or 'Json' format.
1) Compact
2) Json
3) No review is needed
Enter your choice: 1

Private Apps:

Index: 0, NodeName: zs-csc-mux-4-as-d, Location: Azure US East, publicIpAndUdpPort: 74.235.173.101:51280, privateCirdIp: 192.168.7.16/24, Private Apps Qty: 3
Index: 1, NodeName: ns-csc-mux-4-as, Location: Azure East US, publicIpAndUdpPort: 4.246.221.166:51820, privateCirdIp: 192.168.7.15/24, Private Apps Qty: 0
Index: 2, NodeName: pricpa-gcloud-v-0-2-a, Location: Google Cloud Europe, publicIpAndUdpPort: 35.246.67.148:51820, privateCirdIp: 192.168.7.102/24, Private Apps Qty: 2
Index: 3, NodeName: ns-csc-gre-v-1-0e, Location: AWS US, publicIpAndUdpPort: 18.213.109.84:51820, privateCirdIp: 192.168.7.37/24, Private Apps Qty: 4
Index: 4, NodeName: ns-csc-gre-aws-v-0-4, Location: vpc-10-3-0-0, publicIpAndUdpPort: 52.4.62.40:51820, privateCirdIp: 192.168.7.88/24, Private Apps Qty: 4
Index: 5, NodeName: ns-cgc00004, Location: MHB-DC-KVM, publicIpAndUdpPort: 82.68.6.74:51821, privateCirdIp: 192.168.7.11/24, Private Apps Qty: 8
Index: 6, NodeName: ns-cgc00008, Location: CSC via Smartphone, publicIpAndUdpPort: 92.40.213.105:51820, privateCirdIp: 192.168.7.8/24, Private Apps Qty: 0
Index: 7, NodeName: ns-cgc00006, Location: MHB-BH-DC, publicIpAndUdpPort: 217.155.196.81:51820, privateCirdIp: 192.168.7.20/24, Private Apps Qty: 2

Do you want to apply this values?
1) Yes
2) No
Enter your choice: 1
```

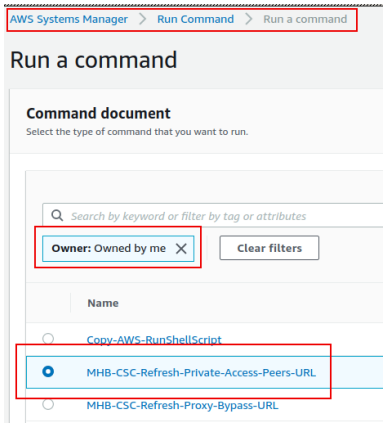
At this moment, you have the option to review the privateApps to configure in Compact or JSON format and to apply the values.

```
Do you want to apply this values?
1) Yes
2) No
Enter your choice: 1

Creating Private Apps:
(MHB-CSC)(INFO) Private Access - (Index: 0, Node: zs-csc-mux-4-as-d) Private App 'Allow all to Azure' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 0, Node: zs-csc-mux-4-as-d) Private App 'Test ICMP from Google' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 0, Node: zs-csc-mux-4-as-d) Private App 'SSH and RDP from MGMT Networks' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 2, Node: pricpa-gcloud-v-0-2-a) Private App 'Allow all to Google Cloud.' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 2, Node: pricpa-gcloud-v-0-2-a) Private App 'Management Networks' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'AWS - SSH and RDP to Remote Server' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'AWS - icmp' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf tcp' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf udp' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SSH and RDP to 10.3.200.0/24' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SNMP to 10.3.200.0/24' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf tcp' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf udp' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Intranet Server' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created succesfully. (destinationPortRange)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Domain Controllers UDP' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'ICMP to 172.19.0.133' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Syslog ICMP' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Syslog tcp port' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Syslog udp port' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Radius Server' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 7, Node: ns-cgc00006) Private App 'BH - SSH to Servers' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 7, Node: ns-cgc00006) Private App 'BH - SSH and RDP to Remote Server' not applicable to this node.

Adding Peers:
(MHB-CSC)(INFO) Private Access - Node: ns-csc-mux-4-as added successfuly.
(MHB-CSC)(INFO) Private Access - Node: pricpa-gcloud-v-0-2-a added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-csc-gre-v-1-0e added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-csc-gre-aws-v-0-4 added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-cgc00004 added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-cgc00008 added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-cgc00006 added successfuly.

Changes Security Group External:
Private Access - Inbound Port Rules 'mhb-csc-private-access-I51280' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-1'
Private Access - Outbound Port Rules 'mhb-csc-private-access-O51820, mhb-csc-private-access-O51821' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-1'

(MHB-CSC)(INFO) Private Access - Private Access Peers List updated succesfully.
```

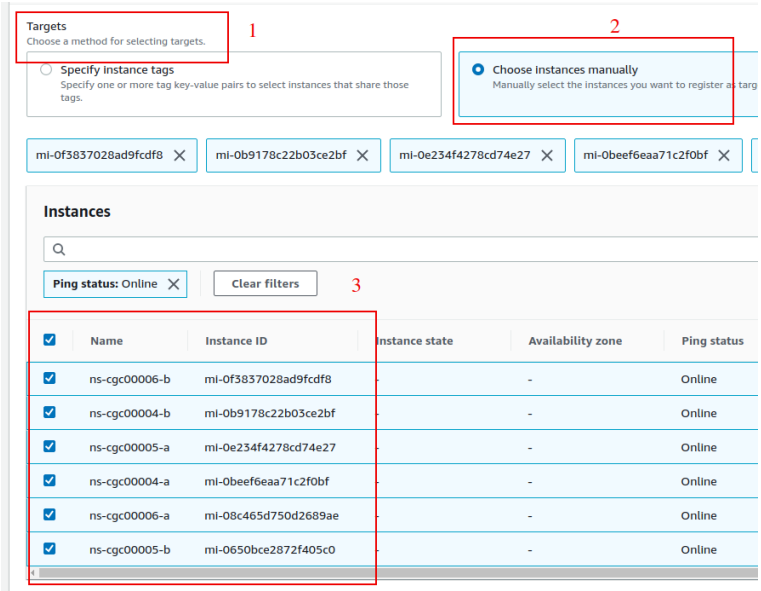
3. The next time you want to refresh the Private Access Peers JSON file, update the file, deploy it on the same location URL and Run Command: "Refresh Private Access Peers URL" using AWS SSM Agent or Rundeck.

AWS System Manager:

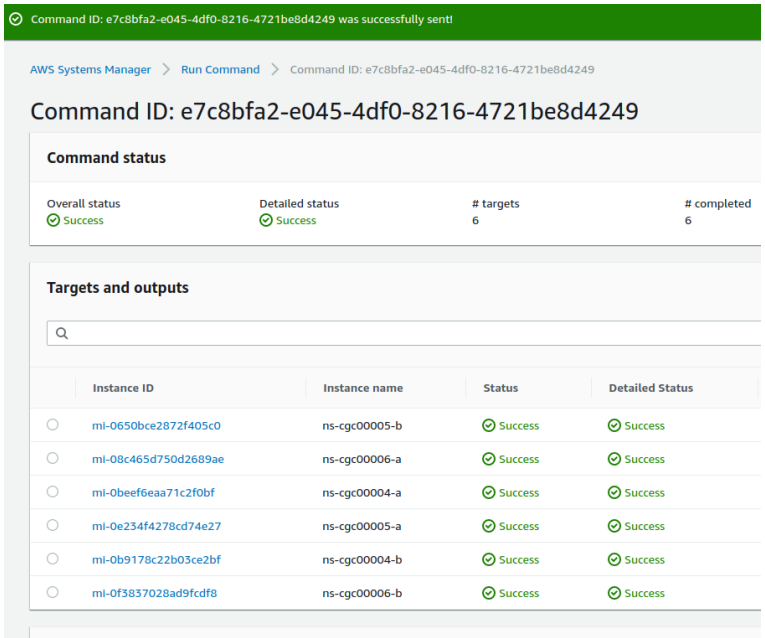
- Go to AWS Systems Manager -> Run Command -> and Select "MHB-CSC-Refresh-Private-Access-Peers-URL"



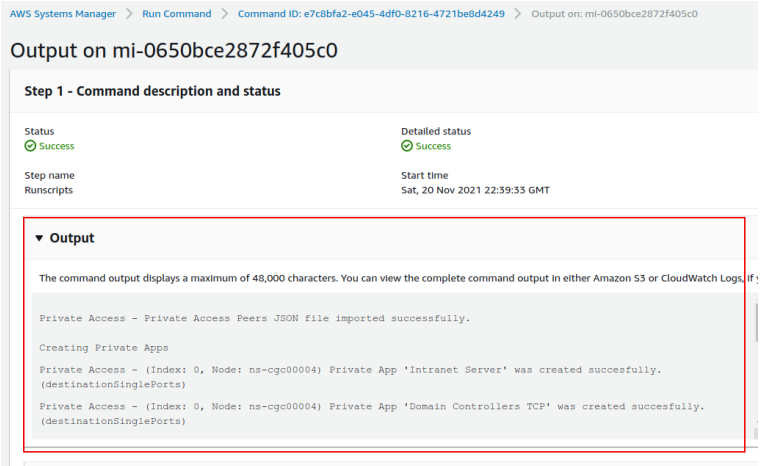
- Move down the screen and select all CSCs:



- Go to the bottom of the page and click "Run". The next page shows the status of the command on each CSC.

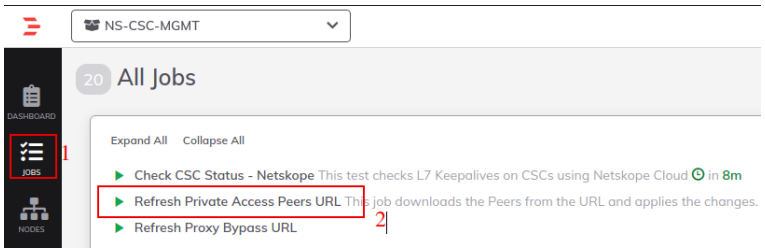


- To see the individual result, right click on the Instance ID and open it on a new TAB. Check the "Output"

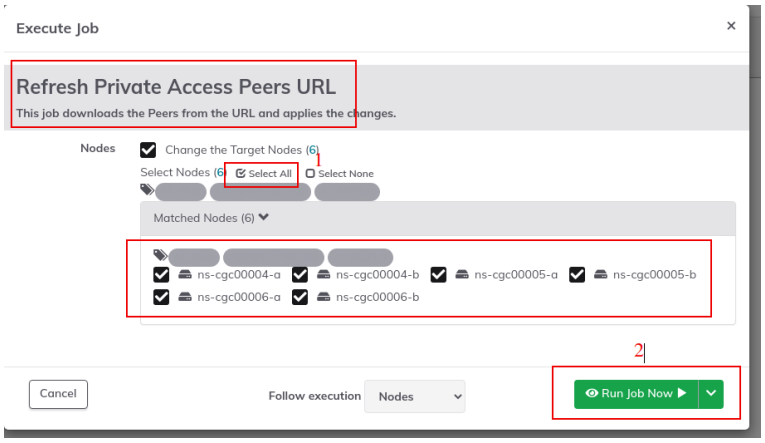


Using Rundeck

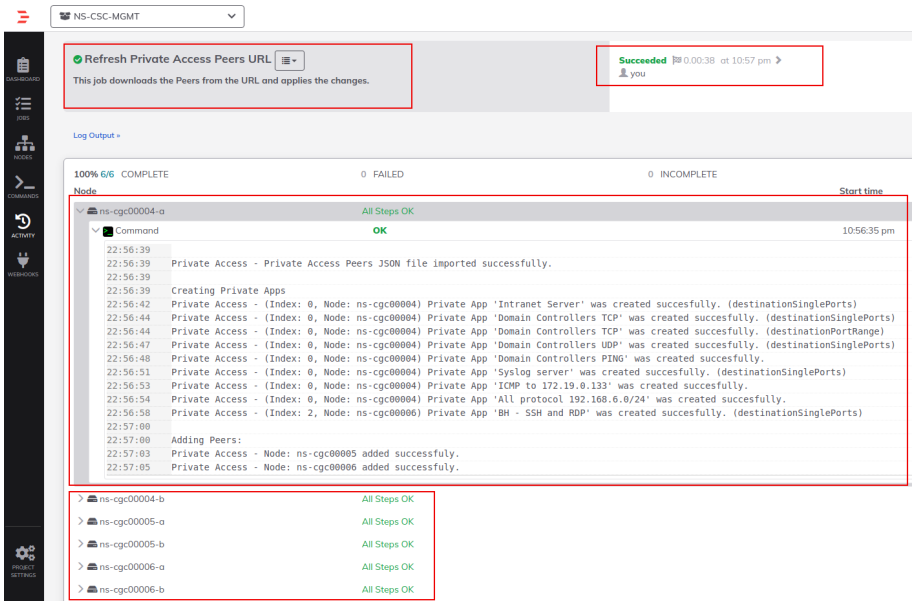
- Go to the Project <name> -> All Jobs -> Run " Refresh Private Access Peers URL"



- Select ALL nodes and click Run.



- Wait to succeeded. You can click on "command" to see the results node by node.



11.3.4.2 Manual: Copy and Paste "Private Access Peers Json file"

From Main Menu, go to 18) Configure Private Access, follow the steps below and Paste the Private Access Peers Json File:

```
1) Create (or change) Private Access Local Configuration
2) Load Private Access Peers JSON configuration file
3) Quit
Enter your choice: 2

Private Access is enabled.

Please, Select Method:
1) Private Access Peers URL
2) Manual (Paste Private Access Peers JSON File)
3) Reset to Default Values
4) Quit
Enter your choice: 2


WARNING: Manual Configuration will remove the Private Access Peers URL if configured.

Do you want to paste the Private Access Peers JSON File?
1) Yes
2) No
Enter your choice: 1

Please, paste Private Access Peers JSON File and press 'Enter' if required.

NOTE: If the json file has errors, it is possible that the script will hang. Press '}' and 'Enter' to end the operation.

Private Access Peers JSON file: {
    "peers": [
        {
            "nodeName": "zs-csc-mux-4-as-d",
            "location": "Azure US East",
            "description": "CSC MUX 4 AS D",
            "publicKey": "4QJ7QPswdTx+mrLMbgLBube0/rw9sSunY78OkljTZ1g=",
            "publicIpAndUdpPort": "74.235.173.101:51280",
            "privateCirdIp": "192.168.7.16/24",
            "persistentKeepAlive": "no",
            "networks": [
                "10.2.2.0/24",
                "10.2.3.0/24",
```

```
Private Access Peers JSON file imported successfully

You can review your values before to apply. Please, Select 'Compact' or 'Json' format.
1) Compact
2) Json
3) No review is needed
Enter your choice: 1

Private Apps:

Index: 0, NodeName: zs-csc-mux-4-as-d, Location: Azure US East, publicIpAndUdpPort: 74.235.173.101:51280, privateCirdIp: 192.168.7.16/24, Private Apps Qty: 3
Index: 1, NodeName: ns-csc-mux-4-as, Location: Azure East US, publicIpAndUdpPort: 4.246.221.166:51820, privateCirdIp: 192.168.7.15/24, Private Apps Qty: 0
Index: 2, NodeName: pricpa-gcloud-v-0-2-a, Location: Google Cloud Europe, publicIpAndUdpPort: 35.246.67.148:51820, privateCirdIp: 192.168.7.102/24, Private Apps Qty: 2
Index: 3, NodeName: ns-csc-gre-v-1-0e, Location: AWS US, publicIpAndUdpPort: 18.213.109.84:51820, privateCirdIp: 192.168.7.37/24, Private Apps Qty: 4
Index: 4, NodeName: ns-csc-gre-aws-v-0-4, Location: vpc-10-3-0-0, publicIpAndUdpPort: 52.4.62.40:51820, privateCirdIp: 192.168.7.88/24, Private Apps Qty: 4
Index: 5, NodeName: ns-cgc00004, Location: MHB-DC-KVM, publicIpAndUdpPort: 82.68.6.74:51821, privateCirdIp: 192.168.7.11/24, Private Apps Qty: 8
Index: 6, NodeName: ns-cgc00008, Location: CSC via Smartphone, publicIpAndUdpPort: 92.40.213.105:51820, privateCirdIp: 192.168.7.8/24, Private Apps Qty: 0
Index: 7, NodeName: ns-cgc00006, Location: MHB-BH-DC, publicIpAndUdpPort: 217.155.196.81:51820, privateCirdIp: 192.168.7.20/24, Private Apps Qty: 2

Do you want to apply this values?
1) Yes
2) No
Enter your choice: 1

Creating Private Apps:
(MHB-CSC)(INFO) Private Access - (Index: 0, Node: zs-csc-mux-4-as-d) Private App 'Allow all to Azure' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 0, Node: zs-csc-mux-4-as-d) Private App 'Test ICMP from Google' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 0, Node: zs-csc-mux-4-as-d) Private App 'SSH and RDP from MGMT Networks' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 2, Node: pricpa-gcloud-v-0-2-a) Private App 'Allow all to Google Cloud.' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 2, Node: pricpa-gcloud-v-0-2-a) Private App 'Management Networks' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'AWS - SSH and RDP to Remote Server' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'AWS - icmp' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf tcp' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 3, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf udp' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SSH and RDP to 10.3.200.0/24' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SNMP to 10.3.200.0/24' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf tcp' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 4, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf udp' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Intranet Server' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created succesfully. (destinationPortRange)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Domain Controllers UDP' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'ICMP to 172.19.0.133' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Syslog ICMP' was created succesfully.
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Syslog tcp port' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Syslog udp port' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 5, Node: ns-cgc00004) Private App 'Radius Server' was created succesfully. (destinationSinglePorts)
(MHB-CSC)(INFO) Private Access - (Index: 7, Node: ns-cgc00006) Private App 'BH - SSH to Servers' not applicable to this node.
(MHB-CSC)(INFO) Private Access - (Index: 7, Node: ns-cgc00006) Private App 'BH - SSH and RDP to Remote Server' not applicable to this node.

Adding Peers:
(MHB-CSC)(INFO) Private Access - Node: ns-csc-mux-4-as added successfuly.
(MHB-CSC)(INFO) Private Access - Node: pricpa-gcloud-v-0-2-a added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-csc-gre-v-1-0e added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-csc-gre-aws-v-0-4 added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-cgc00004 added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-cgc00008 added successfuly.
(MHB-CSC)(INFO) Private Access - Node: ns-cgc00006 added successfuly.

Changes Security Group External:
Private Access - Inbound Port Rules 'mhb-csc-private-access-I51280' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-2'
Private Access - Outbound Port Rules 'mhb-csc-private-access-O51820, mhb-csc-private-access-O51821' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-2'

(MHB-CSC)(INFO) Private Access - Private Access Peers List updated succesfully.
```

Done!

11.4 Show Configurations and Status Private Access.

11.4.1 Using SSH Admin console

From Main Menu, go to 17) Show Configurations and Status Private Access.

```
Private Cloud Private Access (PriCPA)
17) Show Configuration and Status PriCPA.
18) Configure PriCPA (Local and Peers Configuration).
19) Configure CSC Remote Management via PriCPA.

e) Exit

Selection: 17
```

11.4.1.1 Show Peer/s Status

In this menu you can see "All Peers Status" or by peer "Select Peer".

```
1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 1

Please, select an option:
```

1. Show All Peers Status

```
Please, select an option:

1) Show ALL Peers Status
2) Select Peer
3) Quit
Enter your choice: 1

Peer 'ns-csc-mux-4-as' (4.246.221.166:51820) -> 192.168.7.15 is Alive. Source Port OK. Using '51820'
Peer 'pricpa-gcloud-v-0-2-a' (35.246.67.148:51820) -> 192.168.7.102 is Alive. Source Port OK. Using '51820'
Peer 'ns-csc-gre-v-1-0e' (18.213.109.84:51820) -> 192.168.7.37 is Alive. Source Port OK. Using '51820'
Peer 'ns-csc-gre-aws-v-0-4' (52.4.62.40:51820) -> 192.168.7.88 is Alive. Source Port OK. Using '51820'
Peer 'ns-cgc00004' (82.68.6.74:51821) -> 192.168.7.11 is Alive. Source Port was changed. Port configured is '51821' and is using '43330'. Please review NAT rules on this node or, as the last resource, enable Persistent Keepalive o
this node.
Peer 'ns-cgc00008' (92.40.213.105:51820) -> 192.168.7.8 is not reachable. Source Port OK. Using '51820'
Peer 'ns-cgc00006' (217.155.196.81:51820) -> 192.168.7.20 is Alive. Source Port OK. Using '51820'
```

IMPORTANT: This section show is the Peer is Alive and the "Source Port" that arrives at this node from the Peer. The Source Port information is essential to validate that the NAT on the Remote Peer is correct or if the FW on the other end is changing the Source Port. Please, correct the NAT on the remote Peer if you see that the Source Port differs from the expected.

2. Select Peer

This section shows a more detailed information about the Peer.

```
Please, select an option:

1) Show ALL Peers Status
2) Select Peer
3) Quit
Enter your choice: 2

Please, select a Peer

1) "ns-csc-mux-4-as"
2) "pricpa-gcloud-v-0-2-a"
3) "ns-csc-gre-v-1-0e"
4) "ns-csc-gre-aws-v-0-4"
5) "ns-cgc00004"
6) "ns-cgc00008"
7) "ns-cgc00006"
8) Quit
Enter your choice: 4

Peer Status:
     Peer '"ns-csc-gre-aws-v-0-4"' (52.4.62.40:51820) -> 192.168.7.88 is Alive. Source Port OK. Using '51820'

Peer Counters:
      Latest Communication: Thu 1 Jun 21:00:06 UTC 2023
      Transfer: 1.2Gi received, 5.9Mi sent

Peer Configuration:
_____________________________________________

{
  "nodeName": "ns-csc-gre-aws-v-0-4",
  "location": "vpc-10-3-0-0",
  "description": "Node en US east VPC 10.3.0.0/24",
  "publicKey": "mU4StCAt4sW13xVXaMXcRZjZTuP9G9l/OSL2bsFCh2o=",
  "publicIpAndUdpPort": "52.4.62.40:51820",
  "privateCirdIp": "192.168.7.88/24",
  "persistentKeepAlive": "no",
  "networks": [
    "10.3.200.0/24"
  ],
  "privateApps": [
    {
      "description": "Allow SSH and RDP to 10.3.200.0/24",
      "ipProtocol": "tcp",
      "sourceCirdIp": [
```

11.4.1.2 Show Peers Json file (active)

This menu shows the active Private Access Peers Json file.

```
Selection: 17

Show Configuration and Status Private Access

Please, select an option:

1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 2

{
  "peers": [
    {
      "nodeName": "zs-csc-mux-4-as-d",
      "location": "Azure US East",
      "description": "CSC MUX 4 AS D",
      "publicKey": "4QJ7QPswdTx+mrLMbgLBube0/rw9sSunY78OkljTZ1g=",
      "publicIpAndUdpPort": "74.235.173.101:51280",
      "privateCirdIp": "192.168.7.16/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.2.2.0/24",
        "10.2.3.0/24",
        "10.2.4.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all to Azure",
          "ipProtocol": "all",
          "sourceCirdIp": [
            "0.0.0.0/0"
          ],
          "destinationCirdIp": [
            "10.2.2.0/24",
            "10.2.3.0/24"
          ],
          "destinationSinglePorts": [
            ""
```

11.4.1.3 Show Local Configuration

This menu shows the Local configuration of the node.

```
0) Exit

Selection: 17

Show Configuration and Status Private Access

Please, select an option:

1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 3

The Local Configuration menu shows the initial configuration of the node. Private Apps and Networks are not shown here. Check 'Show Peers Json file' to see all information.

Please, copy the following values in a safe place to configure the other CSC on the High Availability Pair. Discard this message if you are doing a single deployment.

Token: YU9WVk9zSUdkRVNnb1UzVmZNZXZISDIvU2RYSWVpdHFpOG01aTAyU04zTT0K

Private Access Local Config JSON file:

{
  "peers": [
    {
      "nodeName": "zs-csc-mux-4-as-d",
      "location": "Azure US East",
      "description": "CSC MUX 4 AS D",
      "publicKey": "4Q77QPsw4Tx+mrLMbgLBubeB/rw9sSunY78OklJTZ1g=",
      "publicIpAndUdpPort": "74.235.173.181:51280",
      "privateCirdIp": "192.168.7.16/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

11.4.1.4 Show Firewall Local Rules

This menu shows in JSON format the Rules required on the Security Group of the external interface of the CSC.

> ***Note:*** *The CSC does the refresh of the External Security Group every time you update the Peers configuration. No manual configuration is required.*

```
Please, select an option:

1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 4

This JSON File shows the required Inbound and Outbound Firewall Rules for the CSC's 'localPrivateIp'.

{
  "nodeName": "zs-csc-mux-4-as-d",
  "localPrivateIp": "10.2.1.28",
  "inboundFirewallRules": [
    {
      "localUdpPort": "51280",
      "peersPublicSourceIP": [
        "4.246.221.166",
        "35.246.67.148",
        "18.213.109.84",
        "52.4.62.40",
        "82.68.6.74",
        "92.40.213.105",
        "217.155.196.81"
      ]
    }
  ],
  "outboundFirewallRules": [
    {
      "remoteUdpPort": "51820",
      "peersPublicDestinationIP": [
        "4.246.221.166",
        "35.246.67.148",
        "18.213.109.84",
        "52.4.62.40",
        "92.40.213.105",
        "217.155.196.81"
      ]
    },
    {
      "remoteUdpPort": "51821",
      "peersPublicDestinationIP": [
        "82.68.6.74"
      ]
    }
  ]
}
```

11.4.2 Using AWS Systems Manager or Rundeck

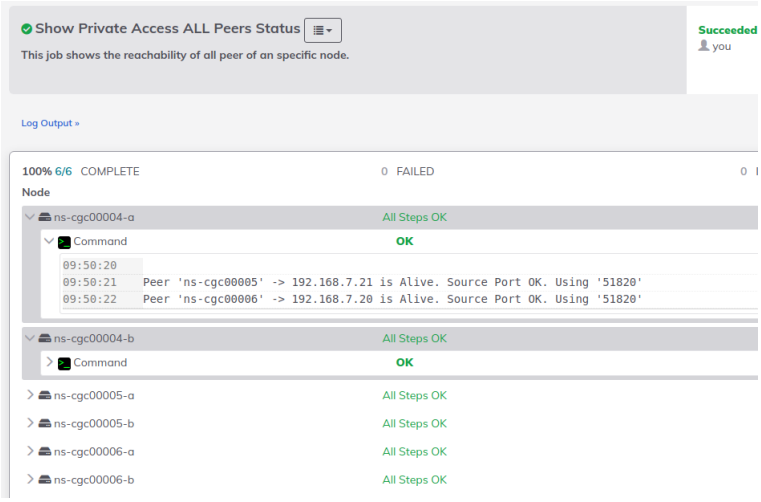
In this case, the information provided is only "Show ALL Peer Status"

11.4.2.1 AWS Systems Manager

Go to AWS Systems Manager and Run Command: "MHB-CSC-Show-Private-Access-ALL-Peers-Status" and select the Nodes. The result will show:

11.4.2.2 Rundeck

On Rundeck, run Job: "Show Private Access ALL Peers Status". Select the nodes. The output will show:



11.5 Configure CSC Remote Management via Private Access.

When the CSC is in HA pair, only the active node belongs to the Private Cloud. For this reason, if you want to reach "the Other CSC" node using SSH, you must configure Remote Management on both CSCs of the HA pair.

The configuration is via SSH Main Menu. You need to add the "Management Networks". For example, in your primary Datacentre, you have the Subnet 172.19.0.0/24, and from that Subnet, you want to reach ALL CSCs on the Private Cloud.

The configuration will be:

```
18) Configure PriCPA (Local and Peers Configuration).
19) Configure CSC Remote Management via PriCPA.

e) Exit

Selection: 19

WARNING! You can isolate this node if the configuration is wrong.
Be careful with this settings. We recommend to be precise with the Host or Subnet configured here.
Subnet Prefixes less than /17 are not accepted.

No Management Networks are configured.

Do you want to configure Management Networks?

1) Yes
2) No
3) Reset to Default
Enter your choice: 1

Input Management Network (IP/Subnet Prefix): 172.19.0.0/24

Do you want to add another Management Network?

1) Yes
2) No
Enter your choice: 2

Management Networks to configure:

Management Networks Qty = 1
Management Network= 172.19.0.0/24

Do you want to apply changes?

1) Yes
2) No
Enter your choice: 1
Private Access - Management Network 172.19.0.0/24 was added on zs-csc-mux-4-as-d-1
```

12 Remote Management

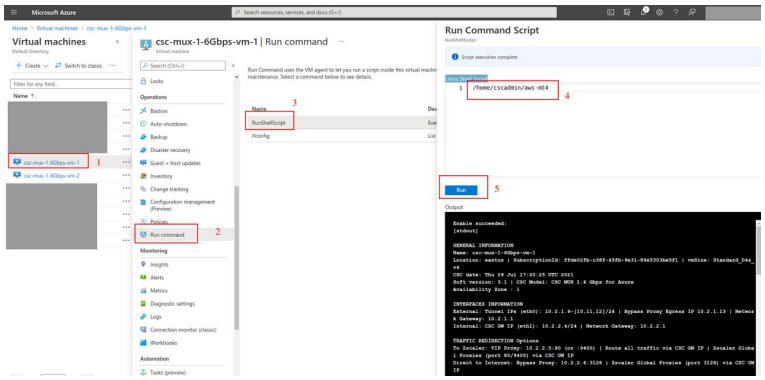
You can use several tools to Remote Manage the CSC. In this chapter, we are showing how to use Azure "Run Command", AWS Systems Manager (Fleet Manager) and Rundeck.

12.1 Azure "Run Command"

12.1.1 Using Azure Portal

Azure portal allows to "Run Command" per VM. "Run Command" is particularly useful if you want to do a quick check, not SSH the CSC. Unfortunately, it is still very buggy and sometimes doesn't work.

Instructions: Select the VM go to Run Command → RunShellScript and on "Linux Shell Script" put the command showed in the below table.



12.1.2 Using Azure CLI

The command to execute is the following:

Linux:

```
$az vm run-command invoke -g <ResourceGroup> -n <VmName> --command-id RunShellScript --scripts <CSC Command> | jq -r .value[].messag
```

Please, note that we are using the program "jq" to extract "message" information and to present it to the Linux terminal.

Example:

```
[redacted] $ az vm run-command invoke -g CSC-EAST-US -n zs-csc-mux-4-as-d-1 --command-id RunShellScript --scripts /home/cscadmin/aws-mt4 | jq -r .value[].message
Enable succeeded:
[stdout]
f1 | vmSize: Standard_F4s_v2
CSC date: Sun 4 Jun 10:27:46 UTC 2023
Soft version: 4.0 | CSC Model: CSC MUX 4 (1.6 Gbps) for Zscaler with PriCPA
Azure Cloud: AzureCloud

INTERFACES INFORMATION
External: Tunnel IPs (eth0): 10.2.1.24-[25,26,27]/24 | Bypass Proxy Egress IP 10.2.1.28 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.15/24 | Network Gateway: 10.2.2.1

TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.2.2.16:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.2.2.17:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP

PUBLIC IP Address INFORMATION
IPsec tunnels Public IP: 74.235.173.100, 74.235.171.133, 74.235.171.132, 20.163.185.222
Bypass Public IP: 74.235.173.171

DNS INFORMATION
DNS Server (1): 1.1.1.1 is Alive
DNS Server (2): 8.8.8.8 is Alive

ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
Primary ZEN node: WashingtonDC_2 | Hostname: was1-2-vpn.zscalerthree.net | IP: 165.225.8.35 is Alive
Secondary ZEN node: NewYorkIII | Hostname: nyc3-vpn.zscalerthree.net | IP: 165.225.38.52 is Alive

LOAD BALANCING INFORMATION
Last change: Sat  3 Jun 22:02:08 UTC 2023
(UP)   Ztun1 is active, using primary.
(UP)   Ztun2 is active, using primary.
(UP)   Ztun3 is active, using primary.
(UP)   Ztun4 is active, using primary.

IPSEC INFORMATION
Ztun1 connected to: WashingtonDC_2, IPsec uptime uptime: 12 hours, since Jun 03 22:00:56 2023, Last Security Association: ESTABLISHED 4 hours ago
Ztun2 connected to: WashingtonDC_2, IPsec uptime uptime: 12 hours, since Jun 03 22:00:56 2023, Last Security Association: ESTABLISHED 4 hours ago
Ztun3 connected to: WashingtonDC_2, IPsec uptime uptime: 12 hours, since Jun 03 22:00:57 2023, Last Security Association: ESTABLISHED 4 hours ago
Ztun4 connected to: WashingtonDC_2, IPsec uptime uptime: 12 hours, since Jun 03 22:00:57 2023, Last Security Association: ESTABLISHED 4 hours ago
```

12.1.3 Commands table

Test #	Descritption	CSC Command
1	MHB-CSC-ShowConfigurationAndStatus	/home/cscadmin/aws-mt4
2	MHB-CSC-SpeedTest	/home/cscadmin/aws-mt7
3	MHB-CSC-TraceRouteAndLatencyTest	/home/cscadmin/aws-mt6
4	MHB-CSC-Refresh-Proxy-Bypass-URL	/home/cscadmin/aws-bp-refresh-list
5	MHB-CSC-ShowLogCurrentMonth	/home/cscadmin/aws-l-current-month
6	MHB-CSC-Refresh-Routed-Bypass-URL	/home/cscadmin/aws-refresh-routed-bypass-url
7	MHB-CSC-ShowLogLastSixMonths	/home/cscadmin/aws-l-last-6-months
8	MHB-CSC-SwitchTunnels	/home/cscadmin/aws-tun-switch
9	MHB-CSC-Reload-High-Availability	/home/cscadmin/aws-reload-high-availability-json
10	MHB-CSC-Reload-Routed-Bypass-json	/home/cscadmin/aws-reload-routed-bypass-json
11	MHB-CSC-Refresh-Private-Access-Peers-URL	/home/cscadmin/aws-refresh-private-access-peers-url
12	MHB-CSC-Reload-Private-Access-JSON-file	/home/cscadmin/aws-reload-private-access-peers-json
13	MHB-CSC-Show-Private-Access-ALL-Peers-Status	/home/cscadmin/aws-show-private-access-all-peers-status
14	MHB-CSC-Update-Nodes-Database	/home/cscadmin/aws-node-region-update

12.2 AWS Systems Manager

The easiest and accessible way to manage the Cloud Security Connectors is to use AWS Systems Manager. AWS official documentation is available here: https://aws.amazon.com/systems-manager/. The CSC has preinstalled the SSM Agent. You need to register the CSC using "Hybrid Activations" and "Run Documents" afterwards.

With AWS Systems Manager, you can manage the CSC remotely. To do it, you need to create "Documents" in advance. "Documents" are a series of commands used by the "Run Command" functionality.
This section explains how to create the "Documents" and "Run Commands".

12.2.1 Create Documents

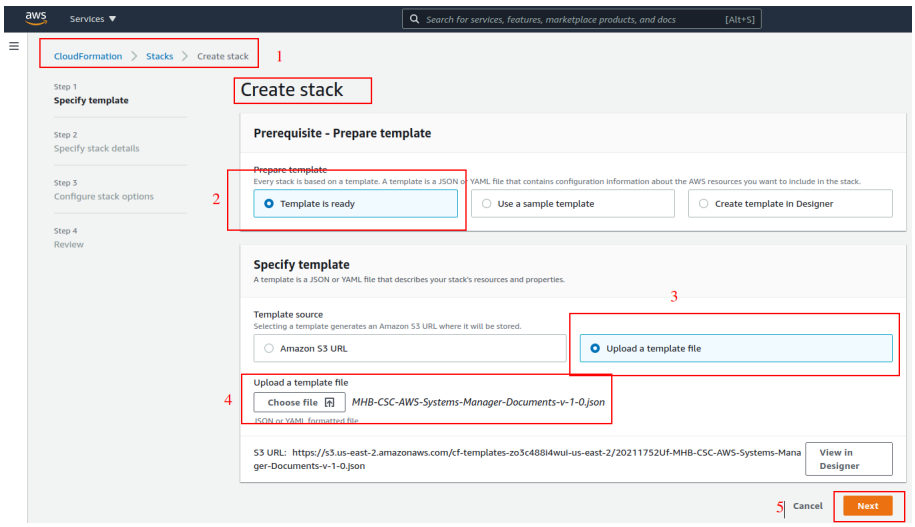
We provide a CloudFormation template to create all "Documents" in one shot.

Steps:

1. Download the CloudFormation template from:

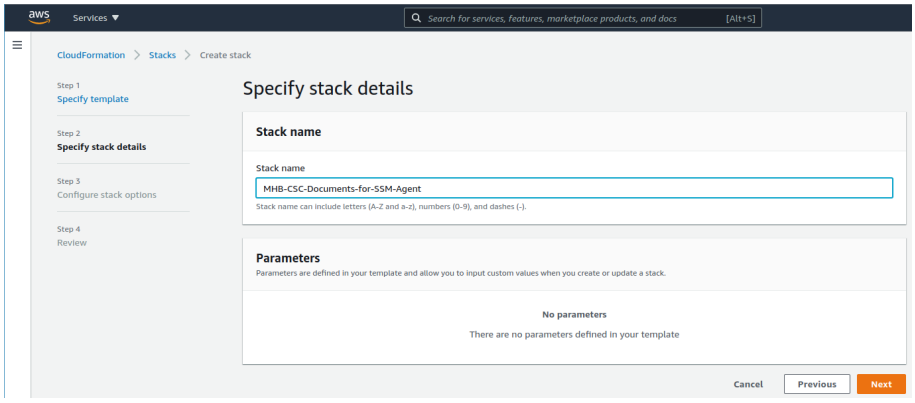
https://maidenheadbridge.freshdesk.com/support/solutions/articles/33000280930-create-documents-to-manage-the-csc-via-aws-systems-manager

2. Deploy Stack. Go to Cloudformation →Create Stack→ Upload a template file



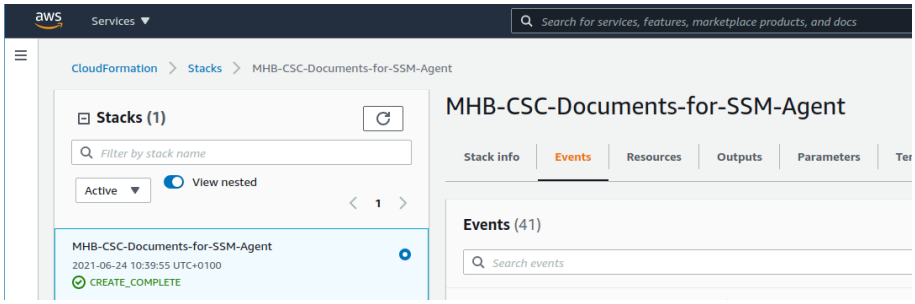
3. Click next.

4. Put the Stack Name

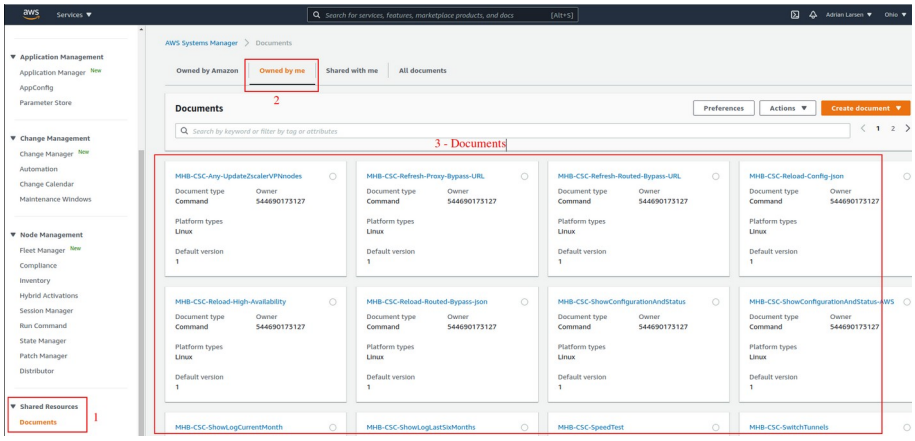


5. Click Next -> Next -> Create Stack.

6. Wait the Stack to complete.



7. Now go to Services -> Systems Manager -> and click "Documents" and choose "Owned by me"



8. Done!

12.2.2 Run Commands

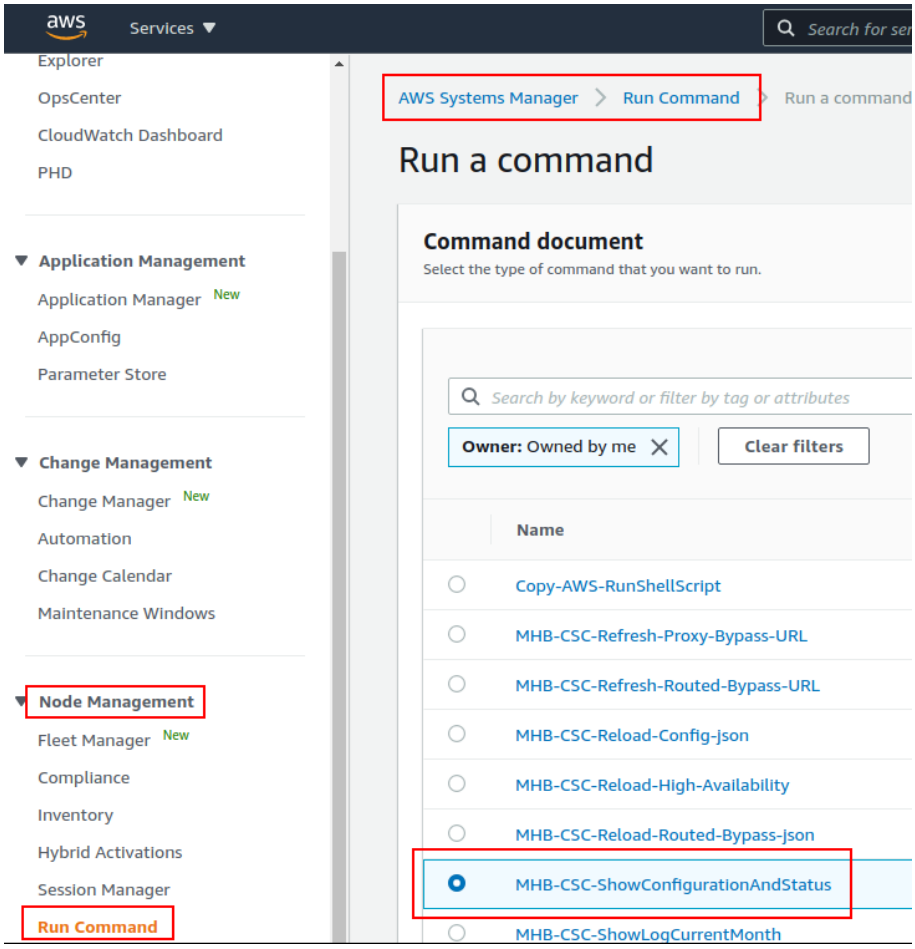
After you have created the Documents, you are ready to Run Commands on the CSC.

You can see the operation results on the "Output" section or store the results on S3 Buckets for further inspection.

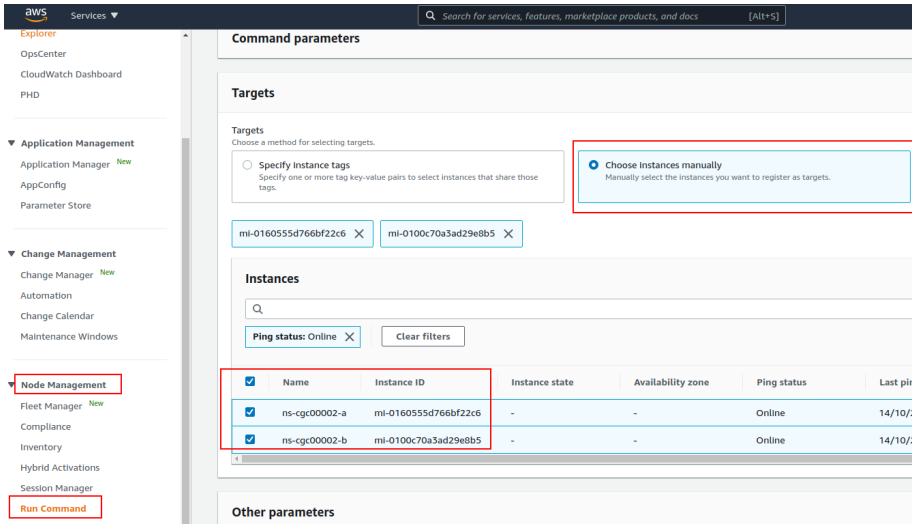
To "Run Commands", go to AWS Systems Manager → Instances & Nodes → Run Command.

Here is an example of Running: MHB-CSC-ShowConfigurationAndStatus

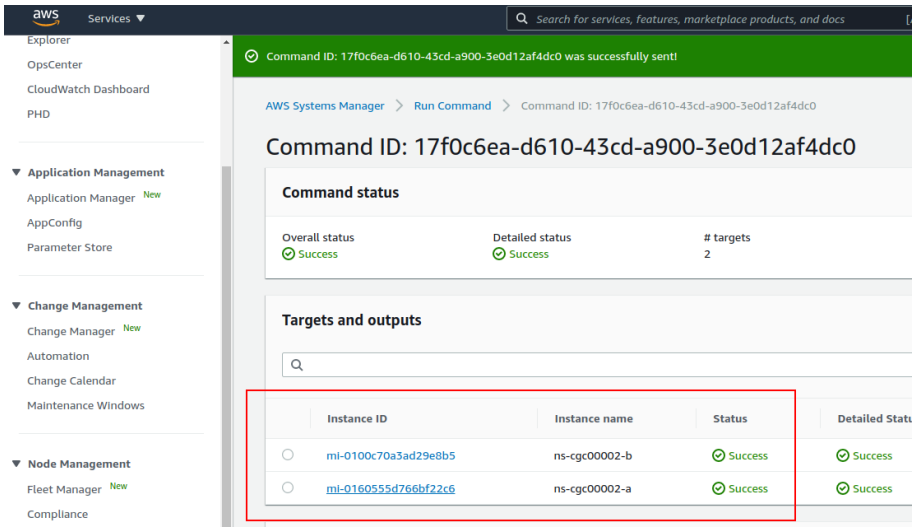
1. Run a Command
2. Select the Document created (Tip: Select "Owned by me")



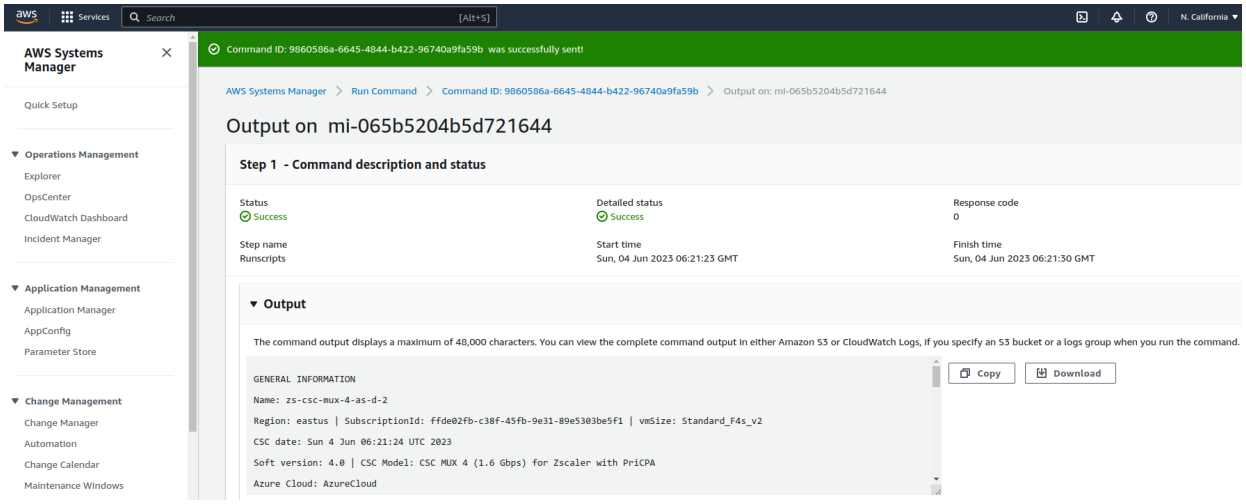
3. Scroll down and Select the Instances



4. Click "Run" . Wait for the Command Status "success"



5. Right click on Instance ID (mi-xxxx) and open in new tab. Check Output.



6. Done! (Note: You can copy the output and to display on a text editor for more visibility)

File Edit View Search Tools Documents Help

*Unsaved Document 1

```
GENERAL INFORMATION
Name: zs-csc-mux-4-as-d-2
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_F4s_v2
CSC date: Sun 4 Jun 06:21:24 UTC 2023
Soft version: 4.0 | CSC Model: CSC MUX 4 (1.6 Gbps) for Zscaler with PriCPA
Azure Cloud: AzureCloud

INTERFACES INFORMATION
External: Tunnel IPs (eth0): 10.2.1.19-[20,21,22]/24 | Bypass Proxy Egress IP 10.2.1.23 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.18/24 | Network Gateway: 10.2.2.1

TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.2.2.19:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.2.2.20:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP

PUBLIC IP Address INFORMATION
IPsec tunnels Public IP: 74.235.175.176, 20.163.185.99, 74.235.173.170, 20.163.185.151
Bypass Public IP: 74.235.173.101

DNS INFORMATION
Using Azure DNS (168.63.129.16) and Google DNS (8.8.8.8, 8.8.4.4)

ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
Primary ZEN node: AutoPrimary | Hostname: vpn.zscalerthree.net | IP: 165.225.8.35 is Alive
Secondary ZEN node: AutoSecondary | Hostname: secondary.vpn.zscalerthree.net | IP: 165.225.38.52 is Alive

LOAD BALANCING INFORMATION
Last change: Sat  3 Jun 19:54:28 UTC 2023
(UP)   Ztun1 is active, using primary.
(UP)   Ztun2 is active, using primary.
(UP)   Ztun3 is active, using primary.
(UP)   Ztun4 is active, using primary.

IPSEC INFORMATION
Ztun1 connected to: AutoPrimary, IPsec uptime uptime: 10 hours, since Jun 03 19:53:19 2023, Last Security Association: ESTABLISHED 2 hours ago
Ztun2 connected to: AutoPrimary, IPsec uptime uptime: 10 hours, since Jun 03 19:53:19 2023, Last Security Association: ESTABLISHED 2 hours ago
Ztun3 connected to: AutoPrimary, IPsec uptime uptime: 10 hours, since Jun 03 19:53:19 2023, Last Security Association: ESTABLISHED 2 hours ago
Ztun4 connected to: AutoPrimary, IPsec uptime uptime: 10 hours, since Jun 03 19:53:19 2023, Last Security Association: ESTABLISHED 2 hours ago

CREDENTIALS INFORMATION
Username: zs-csc-mux-4-as-d-2@maidenheadbridge.com | PSK: Not shown. Please, read it from 'Configuration Wizards' Menu

http://ip.zscaler.com INFORMATION
Ztun1 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.68.253, via Public IP: 74.235.175.176
Ztun2 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.69.27, via Public IP: 20.163.185.99
Ztun3 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 165.225.9.19, via Public IP: 74.235.173.170
Ztun4 Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.51.20, via Public IP: 20.163.185.151
```

12.2.3 List of Documents available for "Run Command"

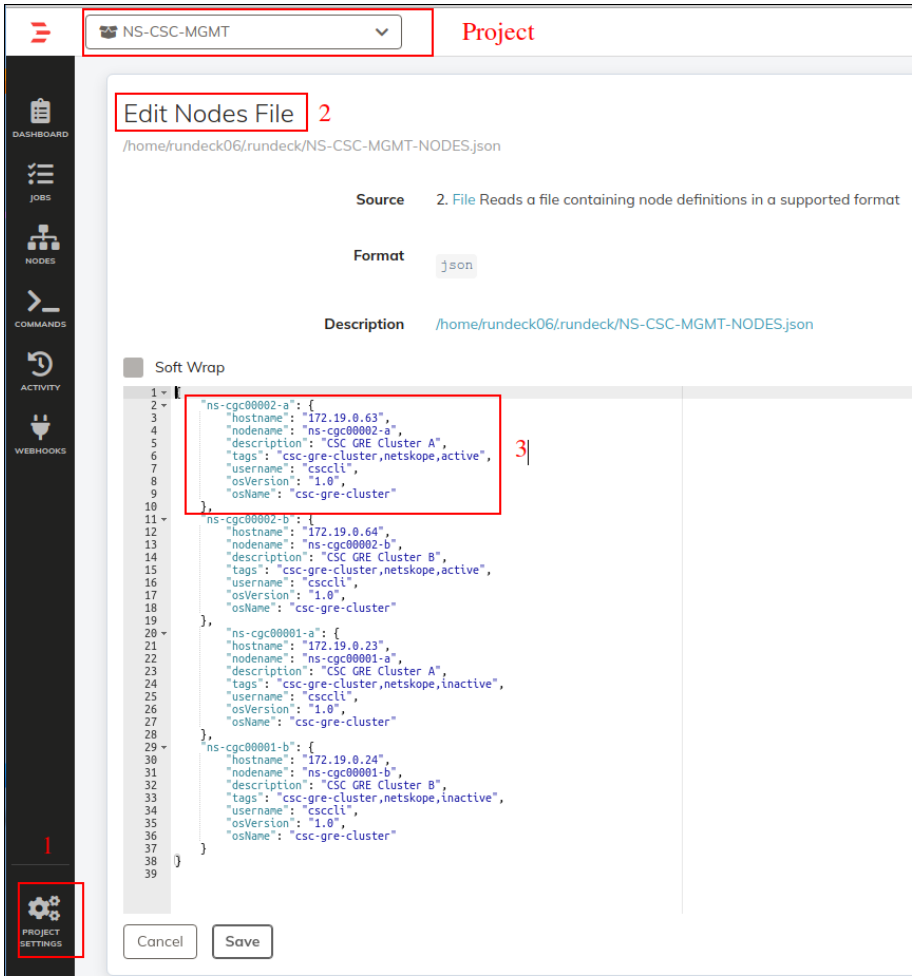
1. "MHB-CSC-ShowConfigurationAndStatus": Executes "Show Configuration and Status"
2. "MHB-CSC-SpeedTest": Performs speedtest.net on the CSC.
3. "MHB-CSC-TraceRouteAndLatencyTest": Performs MyTraceRoute test against the Primary and Secondary ZEN. It also does a Reverse Test from the tunnel active to your Public IP if the tunnel is up.
4. "MHB-CSC-Refresh-Proxy-Bypass-URL": Refresh the Proxy Bypass list using the values of the Proxy Bypass PAC file stored in the URL configured.
5. "MHB-CSC-Refresh-Routed-Bypass-URL": Refresh the Routed Bypass list using the values of the JSON file stored in the URL configured.
6. "MHB-CSC-ShowLogCurrentMonth": Shows current month logs.
7. "MHB-CSC-ShowLogLastSixMonths": Shows last six month logs.
8. "MHB-CSC-SwitchTunnels": Switch tunnels.
9. "MHB-CSC-Reload-Config-json": Reloads the values of config.json file. (not implemented on the CSC Mux for Azure)
10. "MHB-CSC-Reload-High-Availability": Reloads the values of highAvailability.json file. (for CSC on AWS, Azure and Gcloud. Not in use on CSC for Virtual Platforms.
11. "MHB-CSC-Reload-Routed-Bypass-json": Reloads the values of routedBypassRulesFile.json.
12. "MHB-CSC-Update-Nodes-Database": Updates the Zscaler Node Database.
13. "MHB - CSC - Refresh Private Access Peers URL": Refresh the Private Access Peers list using the values of the JSON file stored in the URL configured.
14. "MHB - CSC - Reload Private Access Peers JSON file": Reloads the values of privateAccessPeersConfig.json
15. "MHB - CSC - Show Private Access ALL Peers status": Show the Status of all Private Access Peers.

12.3 Rundeck

Rundeck (https://www.rundeck.com/) is an open-source software Job scheduler and Run Book Automation system for automating routine processes across development and production environments. It combines task scheduling multi-node command execution workflow orchestration and logs everything that happens.

Installation Steps:

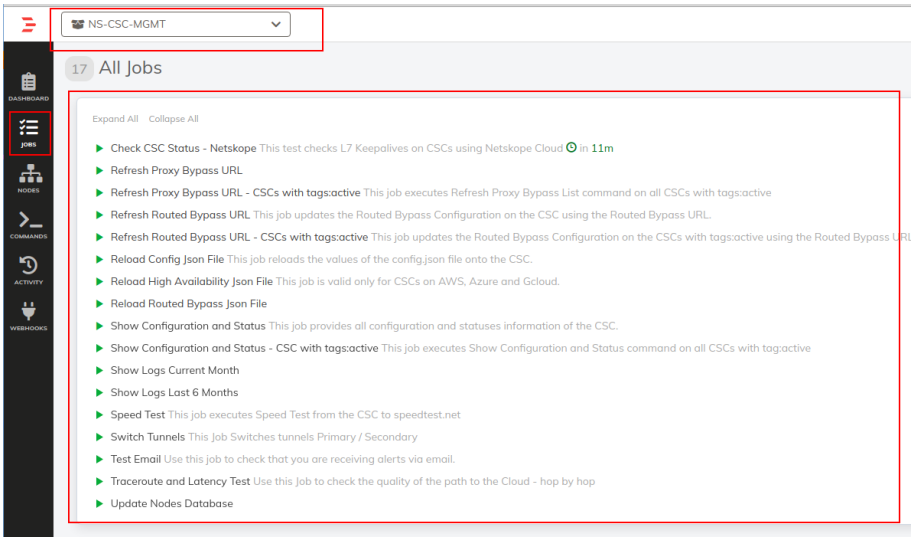
1. Install Rundeck. Intructions at: https://www.rundeck.com/open-source
2. Create a Project.
3. Enable user "csccli" and setup the SSH Public key on each CSC.
4. On the Project, setup the SSH Private and define the nodes:



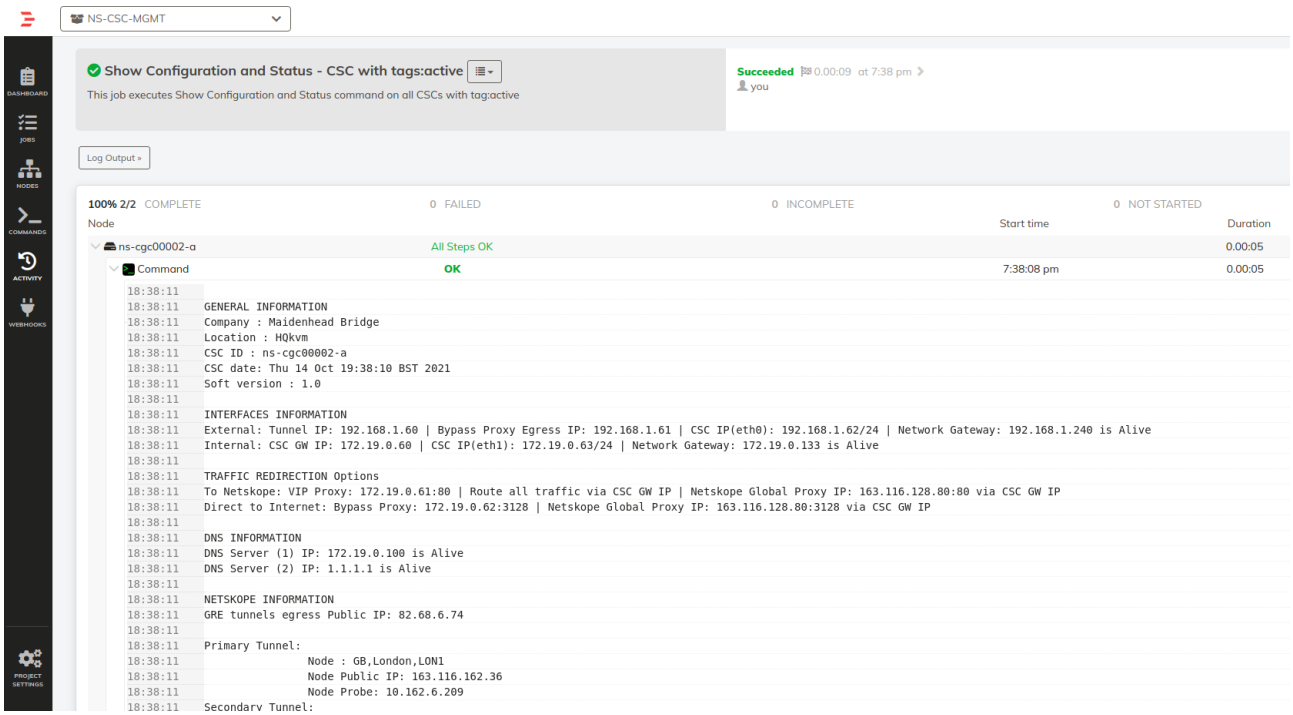
5. Create the jobs. Please, contact Support at http://support.maidenheadbridge.com for the latest Job List.

12.3.1 Jobs

The following screen shows the list of Jobs available.



12.3.2 Running job "Show Configuration and Status"



13 DevOps operations

The CSC is delivered will all configurations and is ready for production. Even so, during the life cycle of the CSC, some parametrization may be required to be changed or modified. For this reason, we provide some configuration utilities that will help with further parametrization and change management.

The CSC offers an option to do some changes using JSON config files. The operation is simple and is three steps:

1. Obtain the current JSON file from the CSC.
2. Download the modified JSON file to the CSC.
3. "Run Command" (AWS Systems Manager) of the specific "reload" document. (or use Rundeck Job or Azure Run Command)

The JSON files are available are:

1. **routedBypassRulesFile.json**: Allows administrators to manually configure Routed Bypass Rules if not using the Routed Bypass URL method.
2. **privateAccessPeersConfig.json:** Use this Json file to configure "networks" and "privateApps" on your Private Cloud.
3. **highAvailability.json**: Allows administrators to configure the CSC on HA pair.

In this chapter, we are going to explain the procedures.

13.1 routedBypassRulesFile.json

You can use this file to create Routed Bypass Rules manually instead of using the automatic method via Routed Bypass URL.

1. Obtain the current "routedBypassRulesFile.json" from the CSC, running "Run Command" (AWS-RunShellScript.). For example:

```
cat /usr/local/etc/mhb-csc/routedBypassRulesFile.json
```


```
{
  "routedBypassRules": [
    {
      "description": "O365 Login URLs 1",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "O365 Login URLs 2",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "443",
      "toPort": "443"
    },
    {
      "description": "O365 Login URLs 3",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "40.126.0.0/18",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "portquiz.net",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "52.47.209.216/32",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "O365 Login URLs 4",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "40.126.0.0/18",
      "fromPort": "443",
      "toPort": "443"
    },
    {
      "description": "Skype and Teams UDP 1",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "13.107.64.0/18",
      "fromPort": "3478",
      "toPort": "3481"
    },
    {
      "description": "Skype and Teams UDP 2",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "52.112.0.0/14",
      "fromPort": "3478",
      "toPort": "3481"
    },
    {
      "description": "Skype and Teams UDP 3",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "52.120.0.0/14",
      "fromPort": "3478",
      "toPort": "3481"
    }
  ]
}
```

2. Create a AWS bucket (or other place) and place on it the modified "routedBypassRulesFile.json" file.
3. Download the file to the CSC. Run Command "AWS-RunShellScript"

```
wget <Your bucket file URL> -O /usr/local/etc/mhb-csc/routedBypassRulesFile.json
```

4. Run Document "MHB-CSC-Reload-Routed-Bypass-json" to apply the changes.

13.2 privateAccessPeersConfig.json

You can use this file to create Private Access Peer Rules manually instead of using the automatic method via Private Access Peers URL.

1. Obtain the current "privateAccessPeersConfig.json" from the CSC, running "Run Command" (AWS-RunShellScript.). For example:

```
cat /usr/local/etc/mhb-csc/privateAccessPeersConfig.json
```


```
{
  "peers": [
    {
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
      "publicIpAndUdpPort": "200.1.1.1:51821",
      "privateCirdIp": "192.168.7.1/24",
      "persistentKeepAlive": "no",
      "networks": ["10.1.1.0/24", "10.1.2.0/24" ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [ "0.0.0.0/0" ],
          "destinationCirdIp": [ "10.1.1.0/24", "10.1.2.0/24" ],
          "destinationSinglePorts": [ "" ],
          "destinationPortRange": {  "fromPort": "", "toPort": "" }
        }
      ]
    },
    {
      "nodeName": "ns-cgc00002",
      "description": "Node on VMware Server 2",
      "location": "Datacentre 2",
      "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
      "publicIpAndUdpPort": "200.1.1.2:51821",
      "privateCirdIp": "192.168.7.2/24",
      "persistentKeepAlive": "no",
      "networks": [ "10.2.1.0/24",  "10.2.2.0/24" ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [ "0.0.0.0/0" ],
          "destinationCirdIp": [ "10.2.1.0/24", "10.2.2.0/24" ],
          "destinationSinglePorts": [ "" ],
          "destinationPortRange": {  "fromPort": "",  "toPort": "" }
        }
      ]
    },
    {
      "nodeName": "ns-cgc00003",
      "description": "Node on VMware Server 3",
      "location": "Branch",
      "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "networks": [ "10.3.1.0/24", "10.3.2.0/24" ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [ "0.0.0.0/0" ],
          "destinationCirdIp": [ "10.3.1.0/24", "10.3.2.0/24" ],
          "destinationSinglePorts": [ "" ],
          "destinationPortRange": { "fromPort": "",  "toPort": "" }
        }
      ]
    }
  ]
}
```


2. Create a AWS bucket and place on it the modified "privateAccessPeersConfig.json" file.

3. Download the file to the CSC. Run Command "AWS-RunShellScript"

```
wget <Your bucket file URL> -O /usr/local/etc/mhb-csc/privateAccessPeersConfig.json
```

4. Run Document "MHB-CSC-Reload-Private-Access-JSON-file" to apply the changes.

13.3 highAvailability.json file

You can configure High Availability via downloading the highAvailability.json file and "Run Command" using the "MHB-CSC-Reload-High-Availability" AWS SSM document.

Steps:

1. Obtain the current "highAvailability.json" from the CSC, running "Run Command" (AWS-RunShellScript.)

*The fields in **bold** are not configurable. So please, do not modify.*

```
cat /usr/local/etc/mhb-csc/highAvailability.json
```

```
{
  "model": "csc-mux-zs-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": false,
    "haIamRole": "",
    "haFirstCsc": {
      "vmName": "",
      "vmResourceGroup": "",
      "haBypassPublicIp": ""
    },
    "haSecondCsc": {
      "vmName": "",
      "vmResourceGroup": "",
      "haBypassPublicIp": ""
    },
    "haPrivateAccessPublicIp": "",
    "haRoutes": []
  }
}
```

2. Create a AWS bucket and place on it the modified "highAvailability.json" file. For example:

*The fields in **bold** are not configurable. So please, do not modify.*

```
{
  "model": "csc-mux-zs-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "SystemAssigned",
    "haFirstCsc": {
      "vmName": "zs-csc-mux-4-as-d-1",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "74.235.173.101"
    },
    "haSecondCsc": {
      "vmName": "zs-csc-mux-4-as-d-2",
      "vmResourceGroup": "CSC-EAST-US",
      "haBypassPublicIp": "74.235.173.171"
    },
    "haPrivateAccessPublicIp": "74.235.173.101",
    "haRoutes": [
      {
        "routeName": "Server-default-route",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      },
      {
        "routeName": "Zscaler-Global-GW",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      }
    ]
  }
}
```

3. Download the file to the CSC. Run Command "AWS-RunShellScript"

```
wget <Your bucket file URL> -O /usr/local/etc/mhb-csc/highAvailability.json
```

4. Apply the IAM Role to the CSC via AWS Console and Run Document "MHB-CSC-Reload-High-Availability" to apply the changes.

14 Appendixes

14.1 Appendix A: Release Notes

14.1.1 Version 4.0.4 (June 2023)

Version 4.0.4 of the CSC Mux for Azure has the following enhancements:

- Change to Standard SKU for Public IPs when deploying the CSC on Availability Set and No Infrastructure.
- The Radius timeout was adjusted to allow more time for MFA authentication.
- Azure WAAGENT agent goes direct instead via the tunnel. This change allows running "Run Commands" even if the Zscaler tunnels are down.
- Manage Administrators menu restricted to "cscadmin" user only. This change avoids a clash of configurations when multiple admins are manipulating the settings.
- Solved a problem when passing User Data (via configUserData.json file) at initial deployment with wrong values on the Zscaler nodes section. The CSC will try once and stop if the entered Zscaler node information is invalid.

14.1.2 Version 4.0 (June 2023)

Version 4.0 of the CSC Mux for Azure has the following enhancements:

- Product renaming: The Cloud Security Connectors for Azure have been renamed according to the amount of IPsec tunnels to Zscaler: CSC Mux 1 (1 x IPsec, 400 Mbps), CSC Mux 2 (2 x IPsec, 800 Mbps), CSC Mux 4 (4 x IPsec, 1,6 Gbps) and CSC Mux 8 (8 x IPsec, 3.2 Gbps). The following list shows the old and new names.
    - CSC Anywhere for Azure → CSC Mux 1 for Azure with PriCPA.
    - New ! → CSC Mux 2 for Azure with PriCPA.
    - CSC Mux 1.6 Gbps → CSC Mux 4 for Azure with PriCPA.
    - CSC Mux 3.2 Gbps → CSC Mux 8 for Azure with PriCPA.
- New! Private Cloud Private Access: PriCPA is a new functionality of the Cloud Security Connector. PriCPA allows you to create a Private Cloud among all CSCs for private traffic. In a matter of minutes, you can build a full mesh encrypted topology between your locations for private traffic with Zero Trust. After making the Private Cloud, you can set up your policies to define who will talk with who inside your Private Cloud.
- New! Traffic Logs: The CSC can send all traffic logs to a Syslog/SIEM server. The Traffic Logs provide visibility of all IP communications to Zscaler, Routed and Proxy Bypasses, PriCPA, and Local received and generated traffic. This functionality is essential to customers with a basic Zscaler Cloud Firewall license.

- New! SNMP support: The CSC Mux for Azure can be monitored via SNMP v2c and v3.
- New! Radius integration: You can access the Admin console using your username and authenticating via Radius protocol to a Radius Server.
- New! The "csccli" user can be enabled and configured via the Admin console, allowing terminal access to the CSC using SSH keys.
- New! SSH access can be restricted per Subnet or IP. It applies to the CSC's Internal (eth1) and PriCPA interface. It is not required anymore to set up external security groups.
- TCPdump functionality is provided via the Admin console for easy troubleshooting of IP traffic.
- New! Azure Load Balancer support. The CSC answers Azure LB Health probes on the CSC VIP IP (port 59400) and CSC Bypass IP (port 53128) when the tunnels to Zscaler are up. If the tunnels to Zscaler are down, the CSC stops answering the probes.
- New! Config User Data support. When launching the CSC, you can insert the configUserData.json file as VM's User Data to pass configuration parameters to the CSC, such as Zscaler Cloud, Zscaler Nodes, DNS Servers, AWS Systems Manager Credentials, Syslog configuration, Bypass (routed and proxy) configuration, PriCPA values and more.

14.1.3 Version 3.1 (July 2021)

Version 3.1 of the CSC Mux for Azure has the following enhancements:

- New! CSC Mux 1.6 Gbps (ex CSC Mux 1G). The CSC Mux with 4 x IPsec tunnels can deliver now 1.6 Gbps.
- New! CSC Mux 3.2 Gbps (ex CSC Mux 2G). The CSC Mux with 8 x IPsec tunnels can deliver now 3.2 Gbps.
- New! Routed Bypass functionality Added. The Routed Bypass allows you to bypass Zscaler for specific destinations when routing all traffic via the CSC Mux using your Public IP.

14.1.4 Version 3.0 (October 2020)

The CSC Mux for Azure was created merging two existing products: the CSC for Azure + CSC Mux for Vmware/Hyper-V.

This version contains all the features of the CSC for Azure (single) plus the following enhancements:

- The CSC Mux is using Ubuntu 20.04 as base OS
- The CSC Mux 1 Gbps can aggregate 4 x IPsec tunnels to deliver 1 Gbps to Zscaler.
- The CSC Mux 2 Gbps can aggregate 8 x IPsec tunnels to deliver 1 Gbps to Zscaler.
- Speedtest runs in parallel in all tunnels and returns; as a result, the sum of all tests.

14.2 Appendix B: configUserData.json file

14.2.1 Parameters

Via configUserData.json file, you can pass values to parameters during the installation of the CSC. You can setup:

1. Zscaler Information: CloudName (zscloud, zscalerthree, zscalertwo; etc.), Nodes (autodiscovery or manual selection), vpnCredentials "domain". (Using "domain" the CSC automatically creates the FQDN (<vmName@domain> and Pre shared keys for IPsec.)
2. AWS SSM agent registration values.
3. DNS servers
4. Syslog servers and traffic log configuration.
5. Bypasses: Proxy Bypass PAC URL and Routed Bypass URL
6. PriCPA Local configuration values, Peers URL and Remote Management Networks.
7. SSH Restrictions via eth1 and wg0.
8. Admin Management: Enable csccli user and SSH Key.

14.2.2 configUserData.json file (blank)

configUserData.json (blank)

*The fields in **bold** are not configurable. So please, do not modify.*

configUserData.json

```
{
  "model": "csc-mux-zs-azure",
  "type": "configUserData",
  "version": "1.0",
  "zscalerInformation": {
    "cloudName": "",
    "vpnNodes": {
      "autoDiscovery": "yes",
      "primary": {
        "hostName": ""
      },
      "secondary": {
        "hostName": ""
      }
    },
    "vpnCredentials": {
      "domain": ""
    }
  },
  "awsSsmAgent": {
    "enable": "no",
    "activationCode": "",
    "activationId": "",
    "awsRegion": ""
  },
  "dns": {
    "useCloudDns": "yes",
    "primaryDnsIp": "",
    "secondaryDnsIp": ""
  },
  "syslog": {
```

```
      "enable": "no",
      "primaryServer": {
        "ip": "",
        "port": ""
      },
      "secondaryServer": {
        "ip": "",
        "port": ""
      },
      "trafficLogs": {
        "enable": "no"
      }
    },
    "bypasses": {
      "proxyBypass": {
        "pacUrl": ""
      },
      "routedBypass": {
        "jsonUrl": ""
      }
    },
    "priCPA": {
      "enable": "no",
      "nodeName": "",
      "location": "",
      "description": "",
      "publicUdpPort": "51280",
      "privateCirdIp": "",
      "persistentKeepAlive": "no",
      "peersJsonFileUrl": "",
      "remoteManagementNetworks": []
    },
    "sshRestrictions": {
      "eth1": {
        "enable": "no",
        "allowedNetworks": []
      },
      "wg0": {
        "enable": "no",
        "allowedNetworks": []
      }
    },
    "adminManagement": {
      "csccli": {
        "enable": "no",
        "sshPublicKey": ""
      }
    }
  }
```

14.2.3 configUserData.json file: Example

configUserData.json


```
{
  "model": "csc-mux-zs-azure",
  "type": "configUserData",
  "version": "1.0",
  "zscalerInformation": {
    "cloudName": "zscalerthree",
    "vpnNodes": {
      "autoDiscovery": "yes",
      "primary": {
        "hostName": ""
      },
      "secondary": {
        "hostName": ""
      }
    },
    "vpnCredentials": {
      "domain": "maidenheadbridge.com"
    }
  },
  "awsSsmAgent": {
    "enable": "yes",
    "activationCode": "ic/L9H7p+floBF5vVefe",
    "activationId": "6ba05125-0bfc-42e7-9535-7f523746e752",
    "awsRegion": "us-east-1"
  },
  "dns": {
    "useCloudDns": "yes",
    "primaryDnsIp": "",
    "secondaryDnsIp": ""
  },
  "syslog": {
    "enable": "yes",
    "primaryServer": {
      "ip": "172.19.0.5",
      "port": "5514"
    },
    "secondaryServer": {
      "ip": "",
      "port": ""
    },
    "trafficLogs": {
      "enable": "yes"
    }
  },
  "bypasses": {
    "proxyBypass": {
      "pacUrl": "https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac"
    },
    "routedBypass": {
      "jsonUrl": "https://mhb-zscaler-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json"
    }
  },
  "priCPA": {
    "enable": "yes",
    "nodeName": "zs-csc-mux-4-as-d",
    "location": "Azure US East",
    "description": "CSC MUX 4 AS D",
    "publicUdpPort": "51280",
    "privateCirdIp": "192.168.7.16/24",
    "persistentKeepAlive": "no",
    "peersJsonFileUrl": "https://mhb-zscaler-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
    "remoteManagementNetworks": [
      "172.19.0.0/24",
      "192.168.1.0/24",
      "192.168.6.0/24"
    ]
  },
  "sshRestrictions": {
    "eth1": {
      "enable": "yes",
      "allowedNetworks": [
        "10.2.0.0/16",
        "172.19.0.0/24",
        "192.168.1.0/24",
        "192.168.6.0/24"
      ]
    },
    "wg0": {
      "enable": "yes",
      "allowedNetworks": [
        "10.2.0.0/16",
        "172.19.0.0/24",
```

```
          "192.168.1.0/24"
        ]
      }
    },
    "adminManagement": {
      "csccli": {
        "enable": "yes",
        "sshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDPm+99wX1/ZhtDIKWh+Uv4TrEYboLoLJlRV6NZctrrkbpq/
WuSCtY9ghL456s4jmSSaNzSSCZ5ywpa3LxmYZ60huoUvYEXBR+lj7MX+trVsifYUe6aJgjPzH8q3x2X72bS20jBQovrNoeN6DZRWWLzLZ4xyczOF+samsm6I/
O3jop68KG6+FydfxFM4DddlJrw29sMi9BJmOzA0Ejl2r3x/
Niz+PWqgbvg5Aq9+uPbJsd6t5egsBsXsKi62blv2rX5hMyZQbxpjcq7BUAc4QcxwZH76X2Y3QVkGnjKXjkFJMcLd6qzN6Su3yYqn41H8ffN2C0rSKD38fvwNDDJTkmZ93PW9mWQweNuWMvLxQTG14z0qR9
VDnIkMtxiJCZyJpUYe6RQDga0nweFIOGBO7N9fA/KzA8r/Gjl52E5KlEQQ725pQXcjHZHFjzTeiD1ZjjEBAURUx0DLldBTsO/oisGUT+pZNQAx6gmX/YxDE/le7qdjuf48aHjwH+uGL1/Q0= cscadmin"
      }
    }
  }
```


14.2.3.1 zscalerInformation


```
  "zscalerInformation": {
    "cloudName": "zscalerthree",
    "vpnNodes": {
      "autoDiscovery": "yes",
      "primary": {
        "hostName": ""
      },
      "secondary": {
        "hostName": ""
      }
    },
    "vpnCredentials": {
      "domain": "maidenheadbridge.com"
    }
  },
```


In this case, the CSC automatically discover the nearest primary and secondary node of Zscaler Three and creates the VPN Credentials. (FQDN: <vmName>@<domain>, PSK: <autogenerated>)

If you want to set up the nodes manually, select "autodiscovery": "no" and put the primary and secondary node host names. You can obtain the VPN hostname from the page: https://ips.<cloudname>.net

14.2.3.2 awsSsmAgent


```
  "awsSsmAgent": {
    "enable": "yes",
    "activationCode": "ic/L9H7p+floBF5vVefe",
    "activationId": "6ba05125-0bfc-42e7-9535-7f523746e752",
    "awsRegion": "us-east-1"
  },
```


Insert here the values for the activation of the AWS SSM agent. See section 9.2.1 for more details.

14.2.3.3 dns


```
  "dns": {
    "useCloudDns": "yes",
    "primaryDnsIp": "",
    "secondaryDnsIp": ""
  },
```


Select "UserCloudDns": "yes" if you want to use Azure DNS (primary) and Google DNS (secondary)

Select "UserCloudDns": "no" and put the values of "primaryDnsIp" and "secondaryDnsIp" if you want to use your own DNS servers.

14.2.3.4 syslog

```
"syslog": {
  "enable": "yes",
  "primaryServer": {
    "ip": "172.19.0.5",
    "port": "5514"
  },
  "secondaryServer": {
    "ip": "",
    "port": ""
  },
  "trafficLogs": {
    "enable": "yes"
  }
}
```

Configure "enable": "yes" and put a value on "primaryServer", "ip" and "port". The secondary servers is optional.

If you want the CSC to collect traffic logs, put "trafficLogs", "enable": "yes"

14.2.3.5 bypasses

```
"bypasses": {
  "proxyBypass": {
    "pacUrl": "https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac"
  },
  "routedBypass": {
    "jsonUrl": "https://mhb-zscaler-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json"
  }
},
```

In this section, you can configure the Proxy Bypass PAC URL and the Routed Bypass URL that contains the routeBypassRulesFile.json

14.2.3.6 priCPA

```
"priCPA": {
  "enable": "yes",
  "nodeName": "zs-csc-mux-4-as-d",
  "location": "Azure US East",
  "description": "CSC MUX 4 AS D",
  "publicUdpPort": "51280",
  "privateCirdIp": "192.168.7.16/24",
  "persistentKeepAlive": "no",
  "peersJsonFileUrl": "https://mhb-zscaler-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
  "remoteManagementNetworks": [
    "172.19.0.0/24",
    "192.168.1.0/24",
    "192.168.6.0/24"
  ]
},
```

In this section, you can configure the Local values for PriCPA, the Peers URL and the Remote Management Networks. See section 11 for more details.

14.2.3.7 sshRestrictions


```
"sshRestrictions": {
  "eth1": {
    "enable": "yes",
    "allowedNetworks": [
      "10.2.0.0/16",
      "172.19.0.0/24",
      "192.168.1.0/24",
      "192.168.6.0/24"
    ]
  },
  "wg0": {
    "enable": "yes",
    "allowedNetworks": [
      "10.2.0.0/16",
      "172.19.0.0/24",
      "192.168.1.0/24"
    ]
  }
},
```


In this section, you can configure from which networks you can access the CSC via SSH. You can configure when the traffic arrives from the local internal interface (eth1) or via PriCPA (wg0).

14.2.3.8 adminManagement

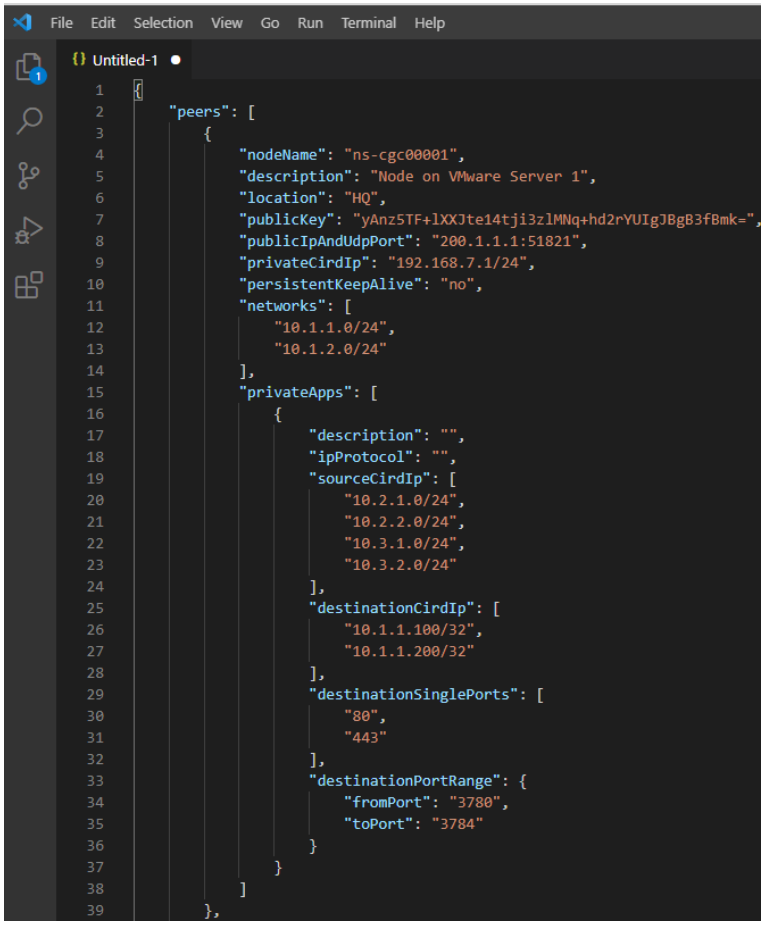

```
"adminManagement": {
  "csccli": {
    "enable": "yes",
    "sshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDPm+99wX1/ZhtDIKWh+Uv4TrEYboLoLJlRV6NZctrrkbpq/
WuSCtY9ghL456s4jmSSaNzSSCZ5ywpa3LxmYZ60huoUvYEXBR+Ij7MX+trVsifYUe6aJgjPzH8q3x2X72bS20jBQovrNoeN6DZRWWLzLZ4xyczOF+samsm6I/
O3jop68KG6+FydfxFM4DddlJrw29sMi9BJmOzA0EjI2r3x/
Niz+PWqgbvg5Aq9+uPbJsd6t5egsBsXsKi62bIv2rX5hMyZQbxpjcq7BUAc4QcxwZH76X2Y3QVkGnjKXjkFJMcLd6qzN6Su3yYqn41H8ffN2C0rSKD38fvwNDDJTkmZ93PW9mWQweNuWMvLxQTG14z0qR9
VDnIkMtxiJCZyJpUYe6RQDga0nweFIOGBO7N9fA/KzA8r/Gjl52E5KlEQQ725pQXcjHZHFjzTeiD1ZjjEBAURUx0DLldBTsO/oisGUT+pZNQAx6gmX/YxDE/le7qdjuf48aHjwH+uGL1/Q0= cscadmin"
  }
}
```


In this section, you can enable access to the terminal console using the "csccli" user. You need to add here the SSH Public Key.

14.3 Appendix C: JSON formatters (Visual Code, Notepad ++)

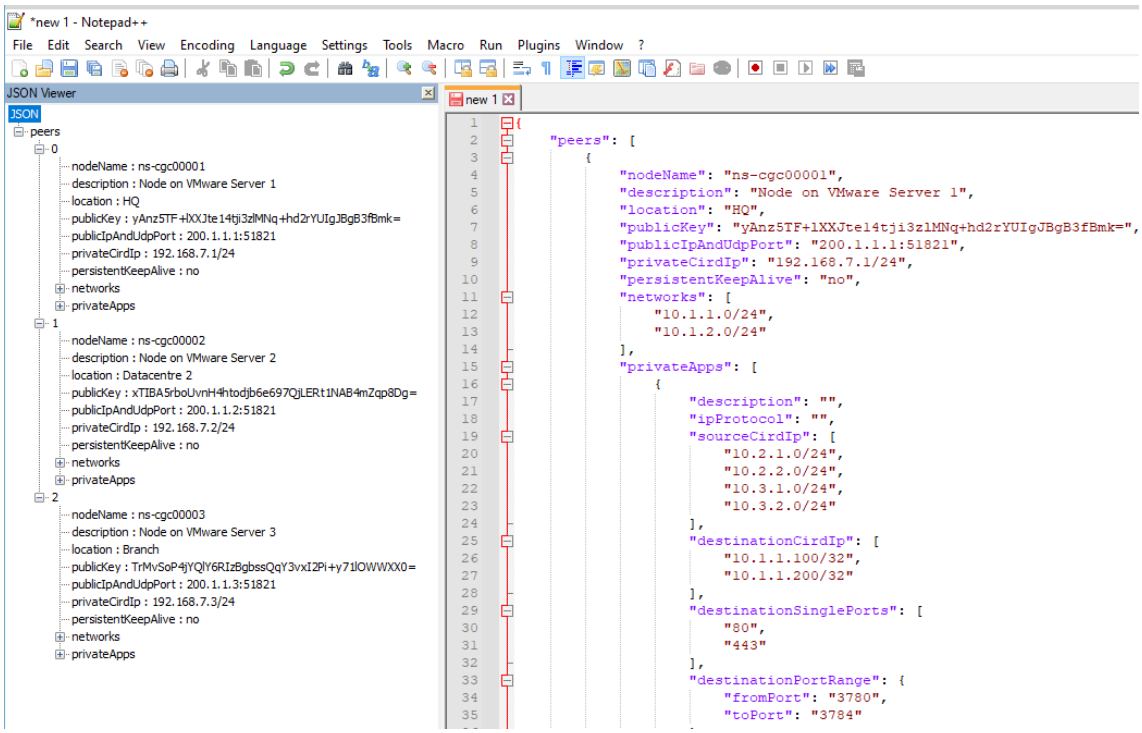
We strongly recommend using Software that can show errors on your JSON file and also can format (beautify) the file for better visibility. Below two examples.

14.3.1 Visual Code

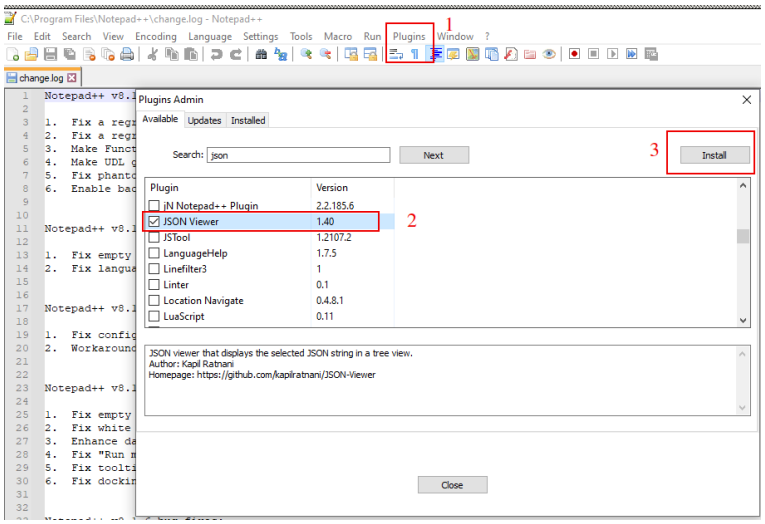


1. Download : https://code.visualstudio.com/download
2. Select your platform and install.
3. Create your JSON.
   - 3.1. Visual Code will show the errors in RED.
   - 3.2. To "Beautify" your JSON file press:
     - 3.2.1. On Windows: "Shift + Alt + F"
     - 3.2.2. On MAC: "Shift + Option + F"
     - 3.2.3. On Linux: " Ctrl + Shift + I"

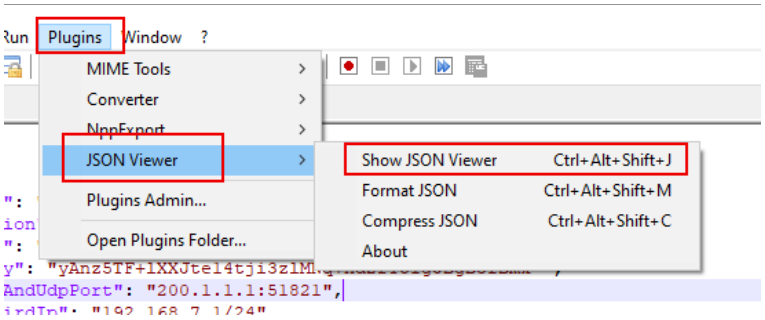
14.3.2 Notepad ++



1. Download: https://notepad-plus-plus.org/downloads/
2. Install JSON Viewer Plug in.



3. Create your JSON file.
4. To Check your JSON file go to: Plugins -> JSON Viewer -> Show JSON Viewer.

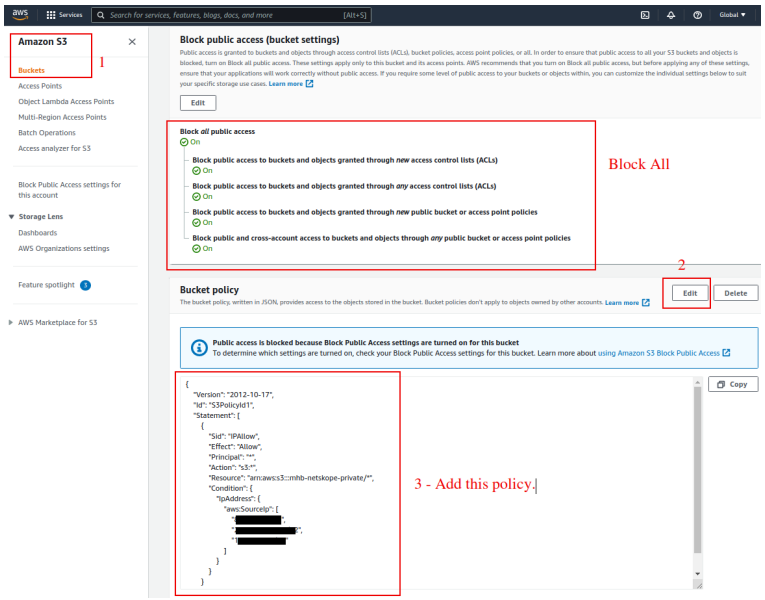


5. To format ("Beautify") your JSON go to: Plugins -> JSON Viewer -> Format JSON

14.4 Appendix D: Securing an AWS Bucket by source IP.

1. On your AWS console create a bucket with default values for permissions: "Block *all* Public Access = on"

2. On Bucket Policy, add your Public IPs in "aws:SourceIp":[]




```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::mhb-zscaler-private/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "200.1.1.1/32",
            "200.1.1.2/32",
            "200.2.0.0/24"
          ]
        }
      }
    }
  ]
}
```

3. Done!