Maidenhead Bridge

zscaler

Cloud Security Connector GRE - AWS with Private Cloud Private Access

(For Amazon Web Services - AWS)

Version 4.0.3

February 2024

Table of Contents

1 Introduction to Cloud Security Connectors for Zscaler.

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

The CSC's GRE for AWS lets you connect securely to Zscaler ZIA up to 1 Gbps [1] without hassle.

The CSC for AWS comes with all the required configurations and works with the Zscaler API. After launching the CSC from the AWS Marketplace using the CloudFormation template provided, it will automatically select the best ZEN nodes, create the GRE tunnels, and make the Location on your Zscaler console.

All Zscaler ZIA functionalities are available. Internal IPs are completely visible on the Zscaler console GUI.

Includes Private Cloud Private Access 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 from AWS Systems Manager, Rundeck (or similar, like Ansible, Salt, Etc.) and SSH.

2 Key benefits of the Cloud Security Connector GRE for AWS

- No Networking knowledge is required.
- Enables any AWS VPC to be connected to Zscaler ZIA up to 3 Gbps.
- Easy to create and deploy: Automated deployment using CloudFormation template and Zscaler API.
- With Private Cloud Private Access (PriCPA) 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.
- All traffic forwarding options supported:
    - Route all traffic to Zscaler (or http/s only).
    - Use of PAC files.

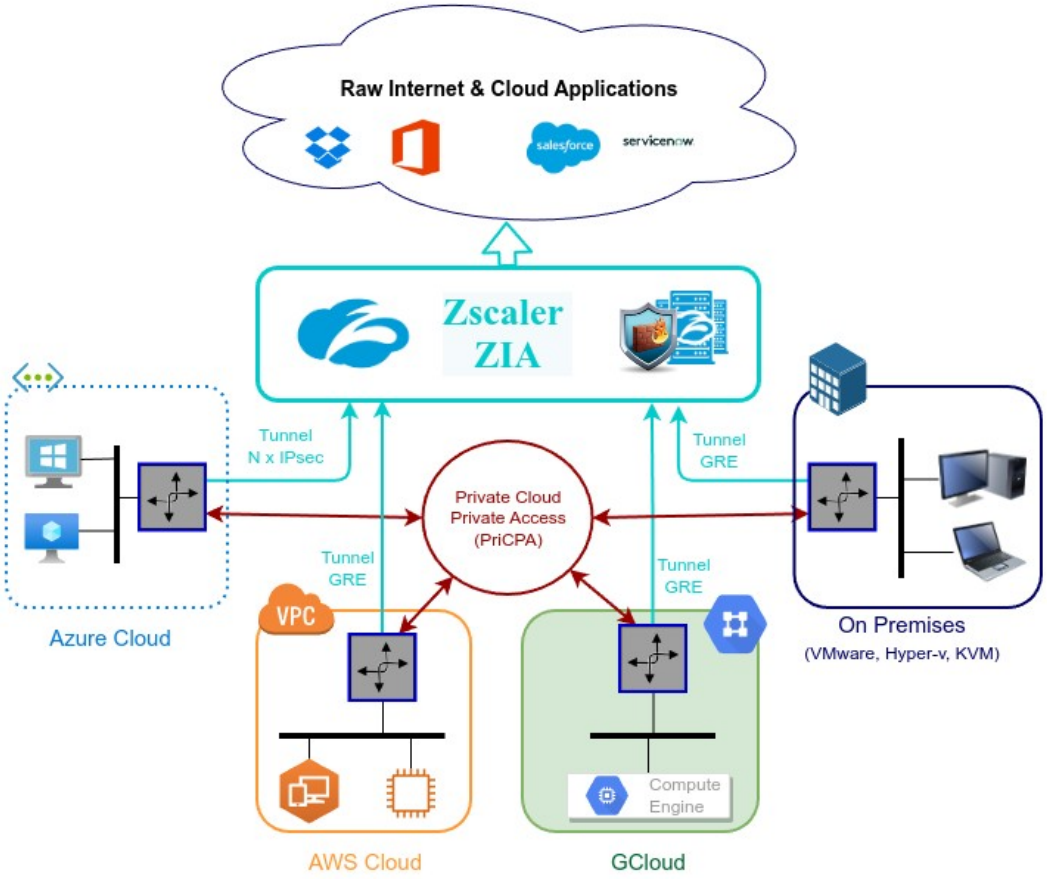
1 Zscaler guarantees 1 Gbps when using GRE tunnels, but the speed can be up to 3 Gbps, depending on the internet path.

- Use of Explicit Proxy.
- No default Route scenarios.

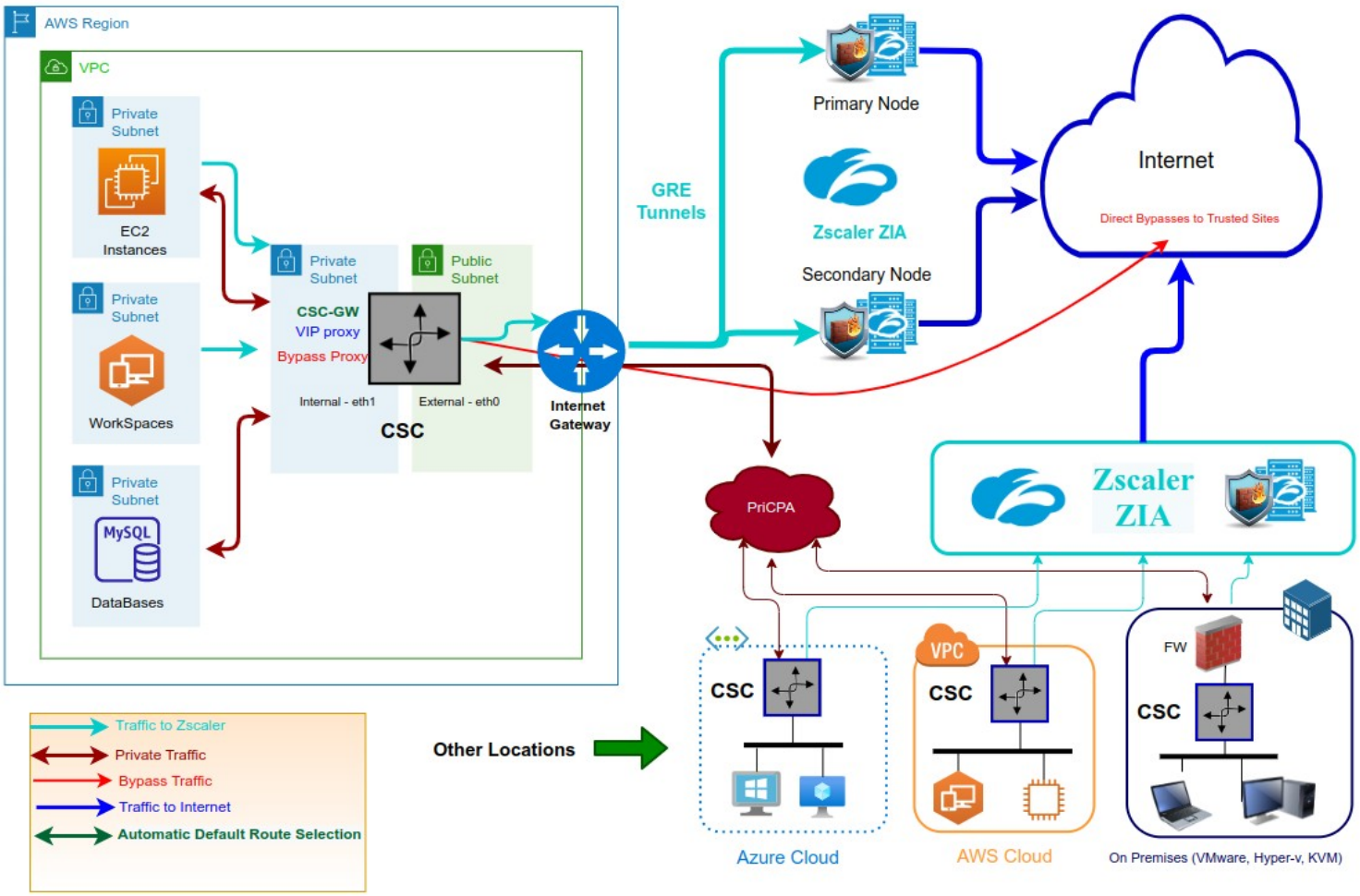
- Multiple options to Bypass Traffic via dedicated Public IP:
  - Layer 7 Proxy Bypass to Trusted Web Sites.
  - Layer 4 Routed Bypass: TCP, UDP and ICMP per source/destination Network and Port (UDP/TCP)
- New! Full Proxy mode for devices with Explicit Proxy settings (i.e. Linux hosts), enabling communications to Zscaler (Location IP based), direct domain Bypass (ie. .domain.com) and communication with internal systems.
- Zscaler 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.
- Management via SSH, AWS Systems Manager, Rundeck or similar. (Ansible, Salt, Etc.)
- It runs on a cheap AWS instance: t2, t3a and t3 instances.

3 Network Diagrams

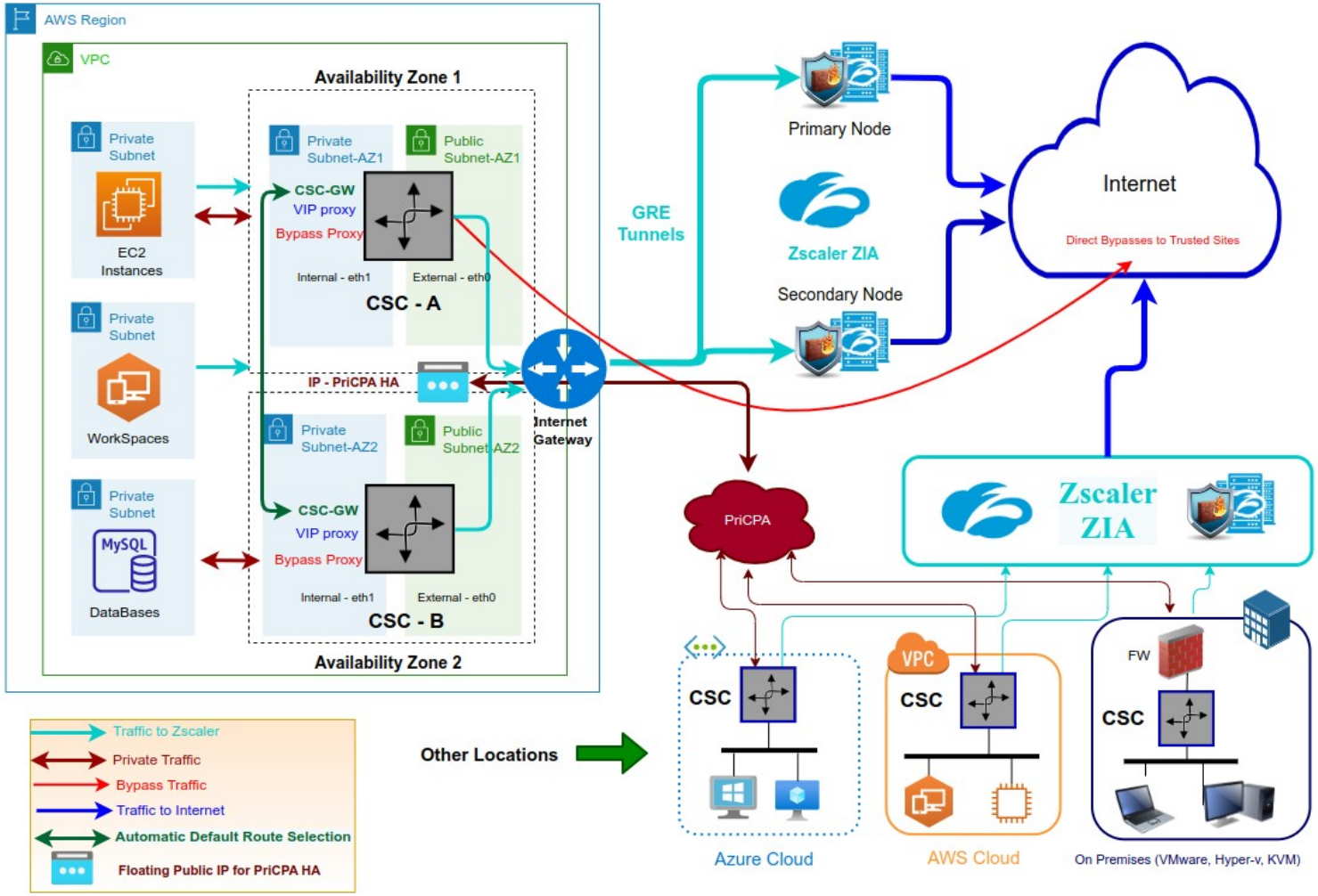
3.1 Cloud Security Connectors (CSC) for Zscaler with PriCPA.



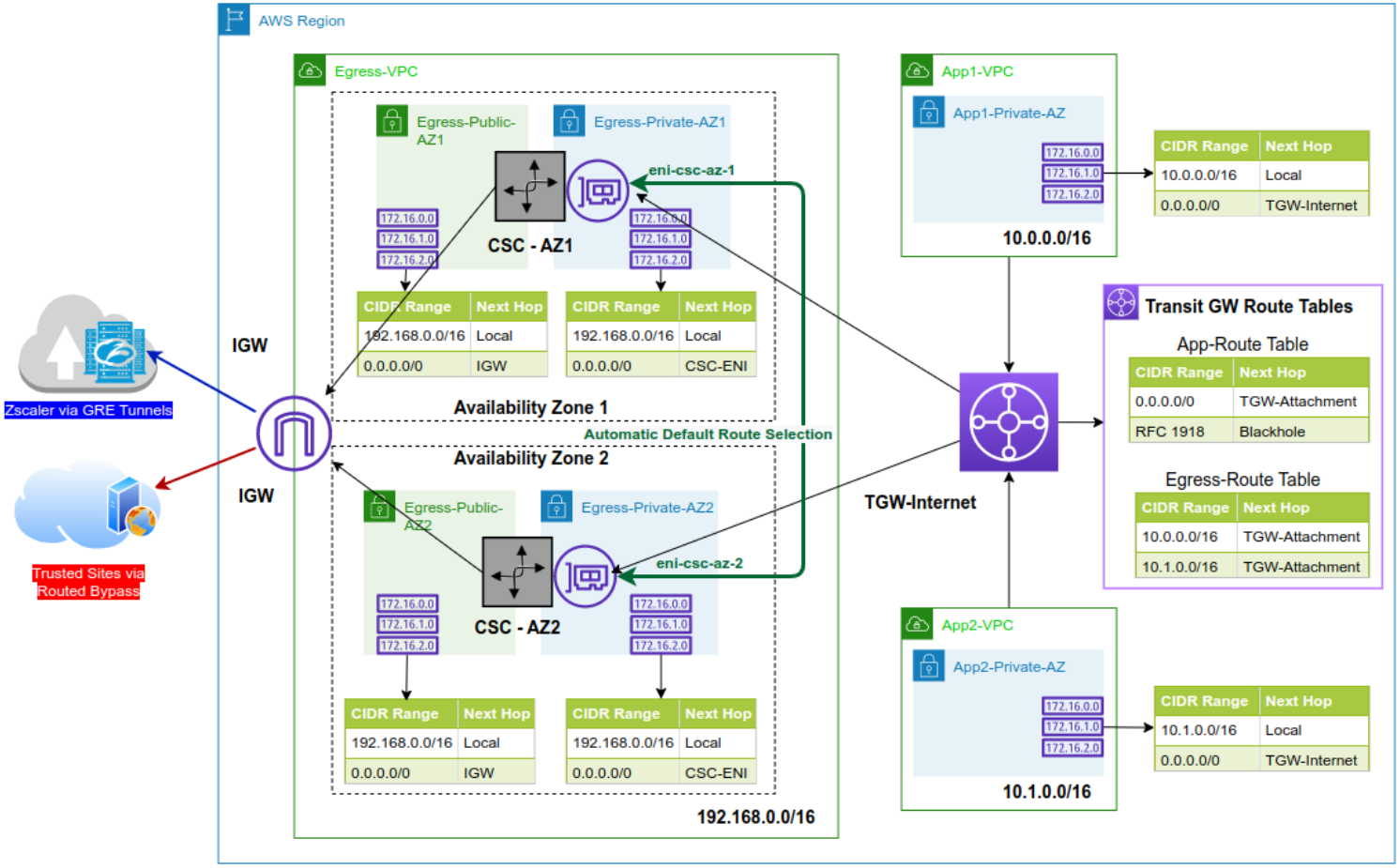
3.2 CSC GRE for AWS - Single Deployment



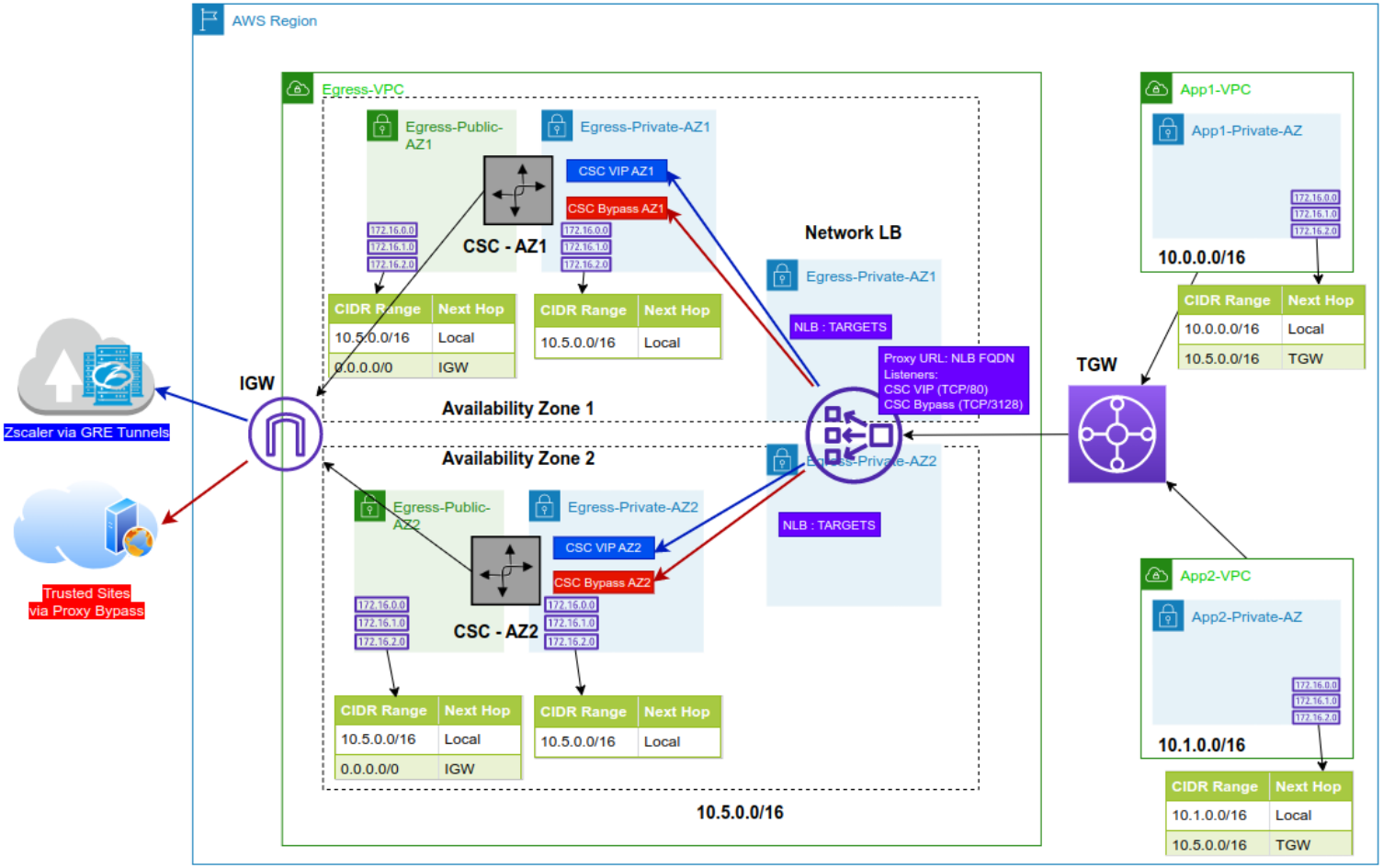
3.3 CSC GRE for AWS - High Availability Deployment



3.4 CSC GRE for AWS - Single Exit to the Internet with automatic routing

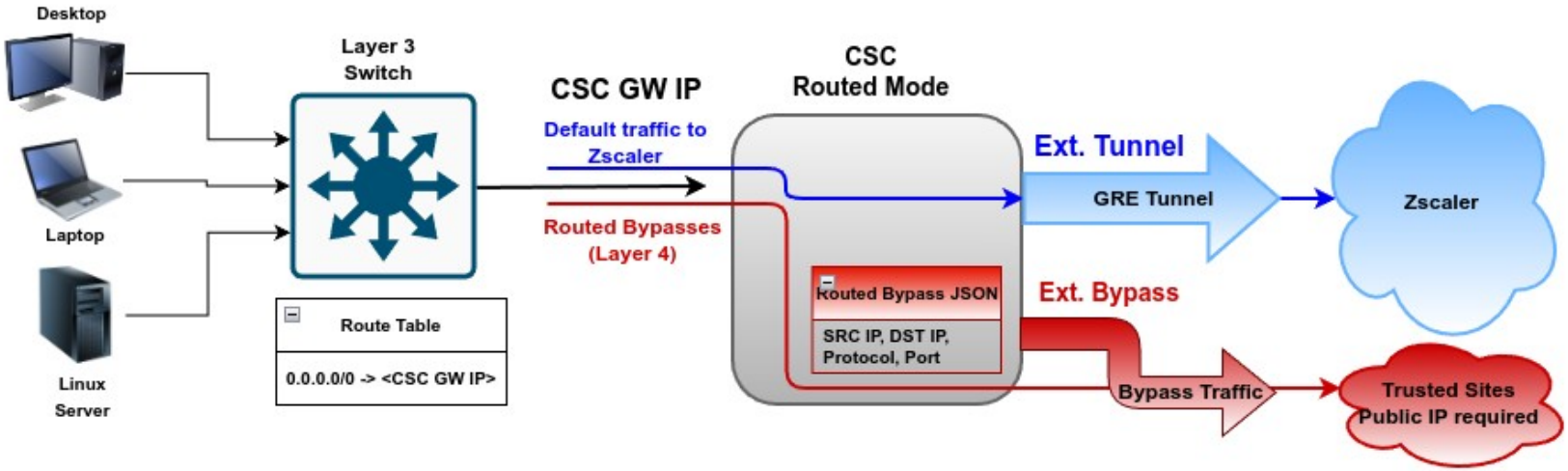


3.5 CSC GRE for AWS - Single Exit to the Internet with Network Load Balancer



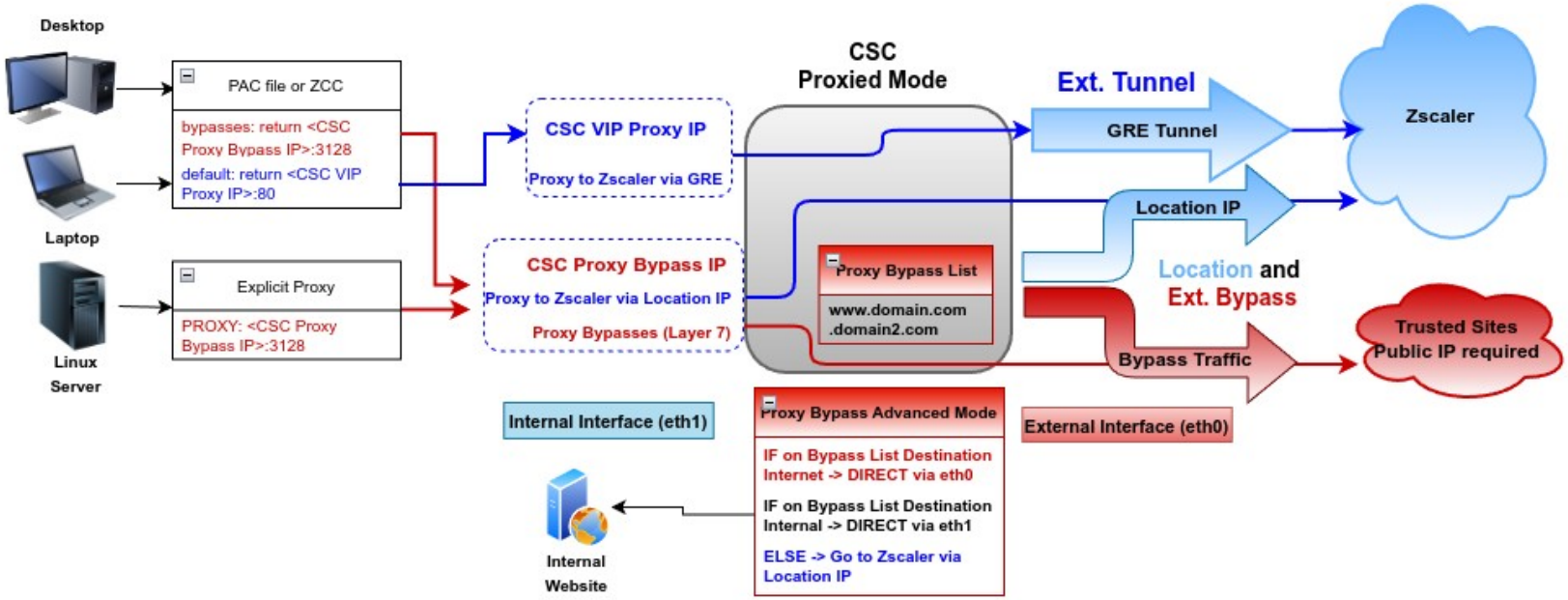
3.6 Traffic Forwarding (I): Routed Mode.

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 in your organization: Laptops, Desktops, Servers, IoT devices, Virtual Desktops, Linux servers, Etc.



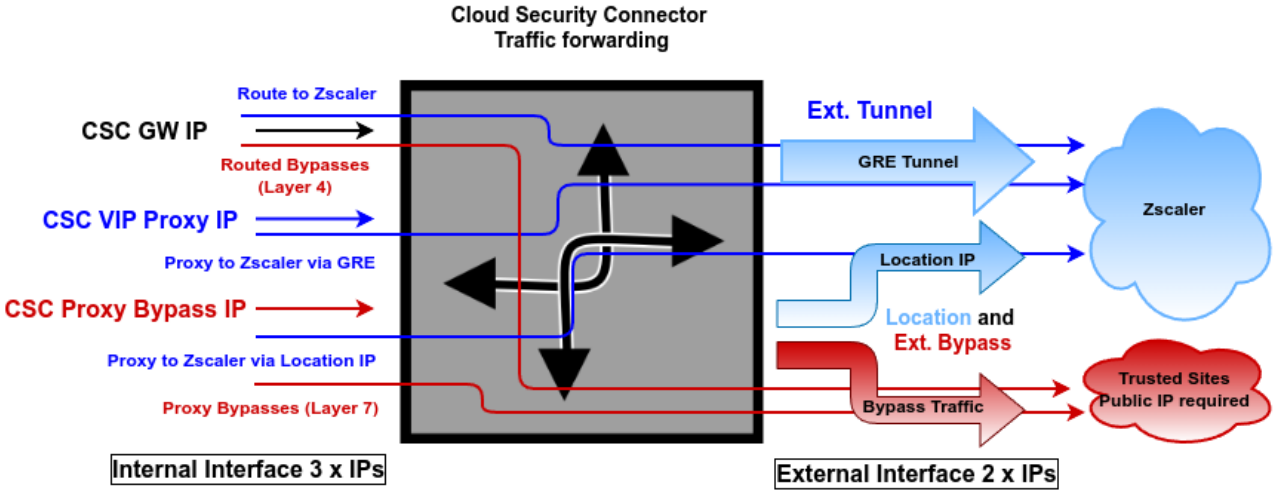
3.7 Traffic Forwarding (II): Proxied Mode.

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 in your organization: Laptops, Desktops, Servers, IoT devices, Virtual Desktops, Linux servers, Etc.



3.8 Traffic Forwarding (III): 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 in your organization: Laptops, Desktops, Servers, IoT devices, Virtual Desktops, Linux servers, Etc.

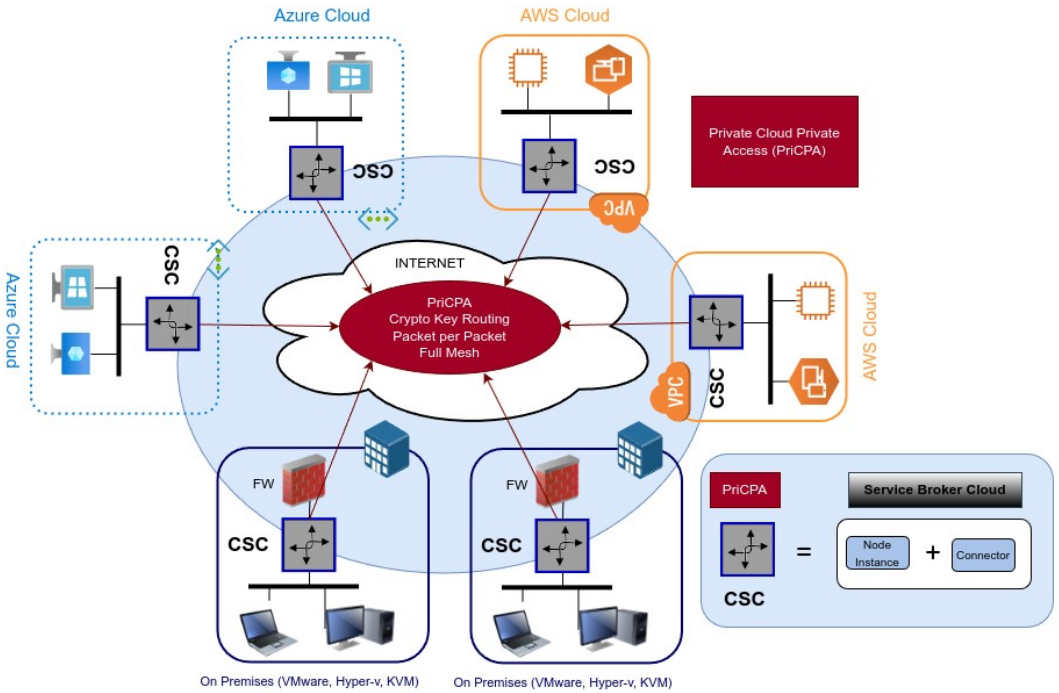


The function of each internal IP is the follwing:

IP	Type	Function
CSC GW	Gateway	It is used as a Gateway when routing traffic to Zscaler and bypasses using the "Routed Bypass" (Layer 4) functionality. (eni-xxyy on AWS)
CSC Vip Proxy	Proxy	It is used as a Proxy for traffic to Zscaler via the GRE tunnel. (1 Gbps up to 3 Gbps)
CSC Proxy Bypass	Proxy	**Standard Mode:** It is used as a Proxy for bypasses using "Proxy Bypass" (Layer 7) functionality. **Advanced Mode:** Same as Standard Mode, but all traffic not in the bypass list is sent to Zscaler (via Location IP, 300 Mbps). Advanced Mode is recommended for devices or apps supporting Explicit Proxy Settings but not PAC files—for example, Linux Servers. Additionally, it is possible to reach internal corporate sites.

3.9 Private Cloud Private Access (PriCPA)

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



Benefits:

- 80% savings compared with VPN Gateways or Service Broker Cloud.
- Any to Any Communications. (Crypto Key Routing)
- All traffic is encrypted.
- All protocols are supported.
- Scalable: Multiple sites and multiple PriCPA clouds.
- Zero Trust.
- Networking as a code.

Benefits:

- DevOps automated deployment.
- 2 Steps configuration: Onboard the Node to PriCPA and Deploy Policies.
- Traffic visibility End to End.
- Traffic Logs.
- Blocks Lateral movement.
- High Performance: 1 Gbps traffic per CSC.

4 Deploying the Cloud Security Connector (CSC)

There are two ways to deploy the CSC: **Basic** and **Advanced mode** (Zscaler API integration). When using Advanced Mode, the configuration of Static IP, GRE tunnel and Location on the Zscaler console is created automatically.

IMPORTANT NOTE: *Unless you will deploy several CSCs, the Basic mode is more than enough and is not needed to use the Zscaler API (Advanced mode). See Appendix C for more information about Advanced mode configuration.*

4.1 Basic Mode deployment

4.1.1 Prerequisites

Before to launch the CSC you need to have this elements ready:

1. **SSH Key.** (you can use any ssh key already in use or to create one specific for the CSC)
2. **VPC ID**
3. **External Subnet:** The External Subnet must be on the same VPC and Availability Zone than the Internal Subnet.
4. **Internal Subnet:** The Internal Subnet must be on the same VPC and Availability Zone than the External Subnet.
5. **Optional:** configUserData.json file. You can pass configuration (DNS servers, Syslog servers, PriCPA values, etc.) during the launch of the CSC by pasting the configUserData.json file on the field "User Data" of the Cloudformation template. Please see Appendix B for detailed information about the configUserData.json file.

4.1.2 Prerequisites EXAMPLE:

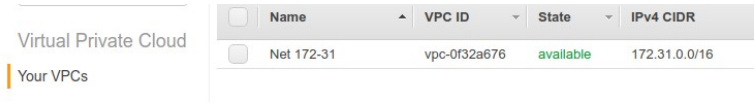
Following an EXAMPLE of prerequisites and how to obtain it.

a) Go to your EC2 Dashboard to get the Key Pairs or to create new ones.

1 – SSH Keys: us-east-key

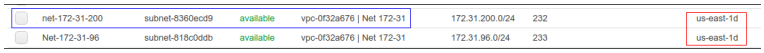
b) Go to your VPC Dashboard, to obtain VPC ID, and Subnets.

2 – VPC ID: vpc-of32a676



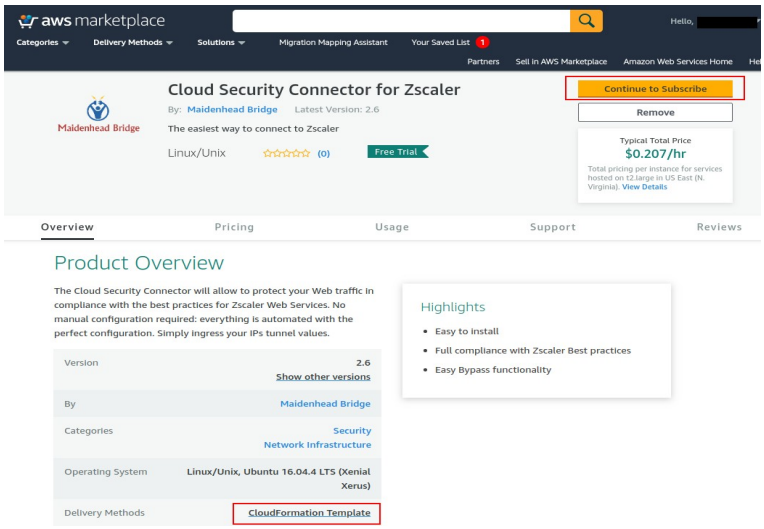
3 – External Subnet: subnet-818c0ddb (Note: Availability Zone us-east-1d and VPC ID vpc-of32a676)

4- Internal Subnet: subnet-8360ecd9 (Note: Availability Zone us-east-1d and VPC ID vpc-of32a676)



4.1.3 Launching the CSC from AWS Market

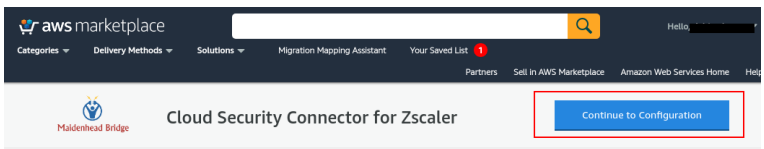
1. Go to the Cloud Security Connector for Zscaler product page at the AWS Market:



Please, note at the bottom that the Fulfilment Method is CloudFormation Template.

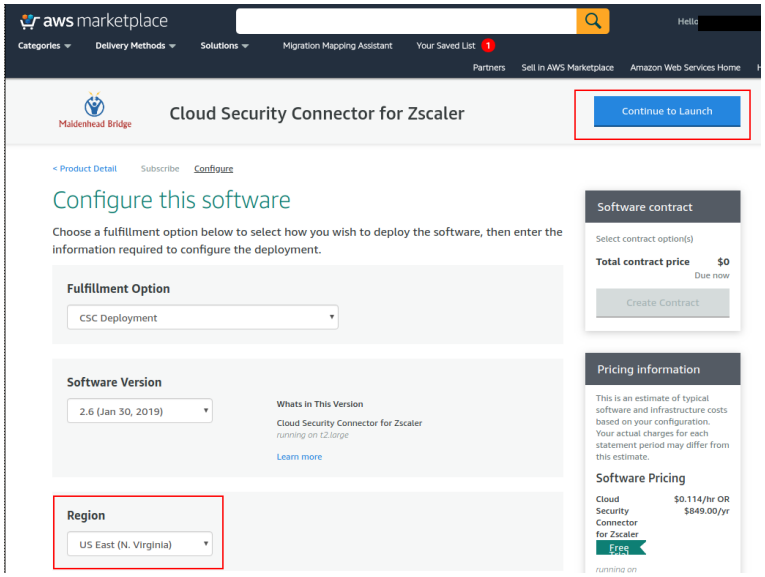
→ Click ***"Continue to Subscribe"***

2. You will be asked to accept the EULA (at the first time), then Continue..



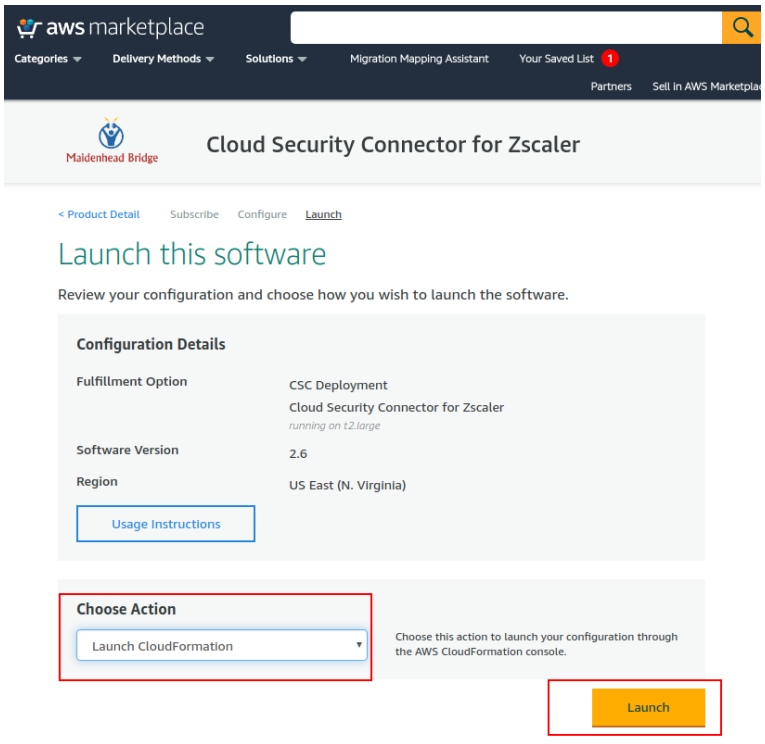
→ Click ***"Continue to Configuration"***

3. Select “Region”



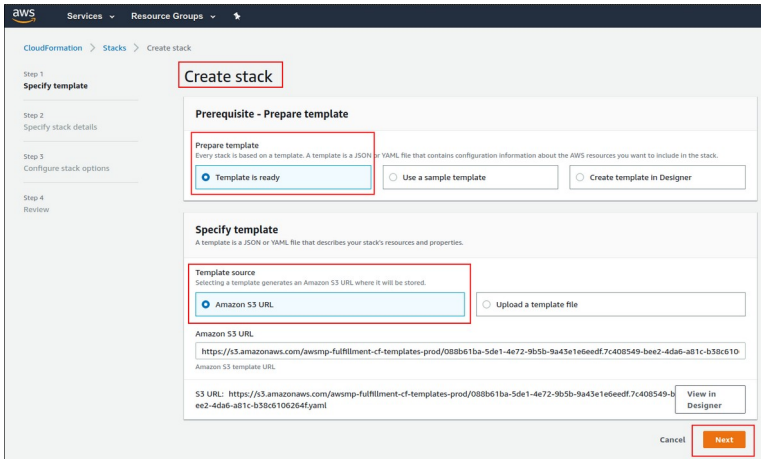
→ Click *“Continue to Launch”*

4. Choose Action: “Launch CloudFormation”



→ Click *“Launch”*

5. At this point, the “Create Stack” screen will appear.



→ Click ***"Next"***

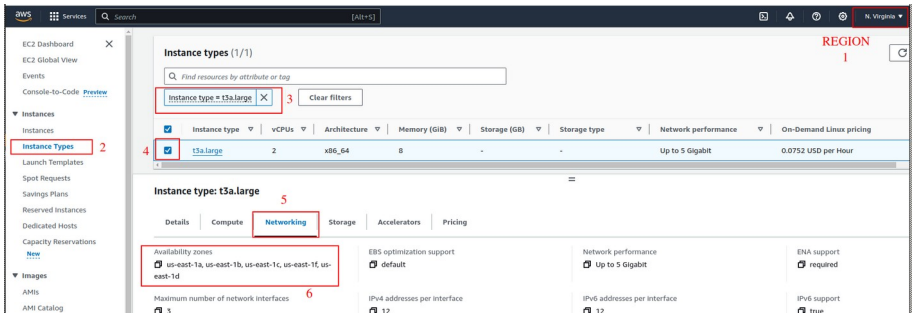
6. Specify Details. Please insert here your values:

- Stack Name
- VPC
- External Subnet
- Internal Subnet
- Name [of the instance] *(we recommend to use the same name for the stack and the instance for easy visualization)*
- AWS Instance Type: t3a.large (default). (*)
- Key Name
- (optional) User Data (paste configUserData.json file. See Appendix A)

(*) The following table shows the recommended instances. The information is an extract from:

https://aws.amazon.com/ec2/instance-types/ and https://aws.amazon.com/ec2/pricing/on-demand/

Please, check if the instance is available in the Availability Zone of your Region. Example:

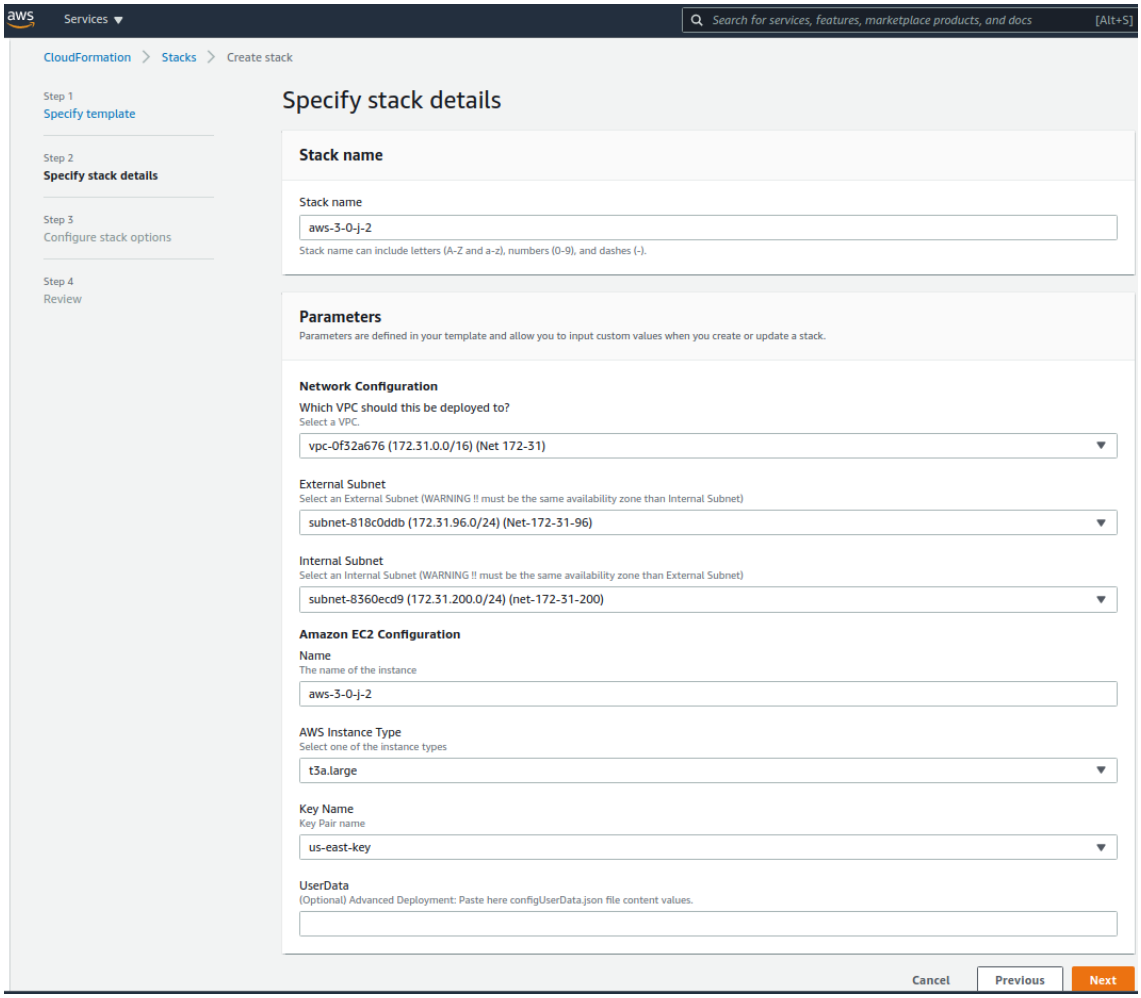


Instance	Vcpus	Memory	Bandwidth
t3a.small	2	2	Up to 5 Gigabit
t3.small	2	2	Up to 5 Gigabit
t2.small	1	2	Low to Moderate
t3a.medium	2	4	Up to 5 Gigabit
t3.medium	2	4	Up to 5 Gigabit
t2.medium	2	4	Low to Moderate
t3a.large	2	8	Up to 5 Gigabit
t3.large	2	8	Up to 5 Gigabit
m5a.large	2	8	up to 10 Gbps
t2.large	2	8	Low to Moderate
m5.large	2	8	up to 10 Gbps
m5n.large	2	8	up to 25 Gbps
m5zn.large	2	8	up to 25 Gbps
m5a.xlarge	4	16	up to 10 Gbps
m5.xlarge	4	16	up to 10 Gbps
m5n.xlarge	4	16	up to 25 Gbps
m5zn.xlarge	4	16	up to 25 Gbps

The table is ordered by price, where t3a.small is the cheapest and m5zn.xlarge is the more expensive. Some recommendations:

- Use t3a.small or t3.small when the traffic required is less than 1 Gbps and the Proxy Bypass is not in use.
- Use t3a.medium or t3.medium when the traffic required is is less than 1 Gbps and the Proxy Bypass functionality is needed.
- Use any instance in Green in all other cases.
- Avoid using t2 instances if possible because of bandwidth constraints.

Here the Screenshot using the values of point 4.1.1 Prerequisites EXAMPLE: (please, use here your own values)

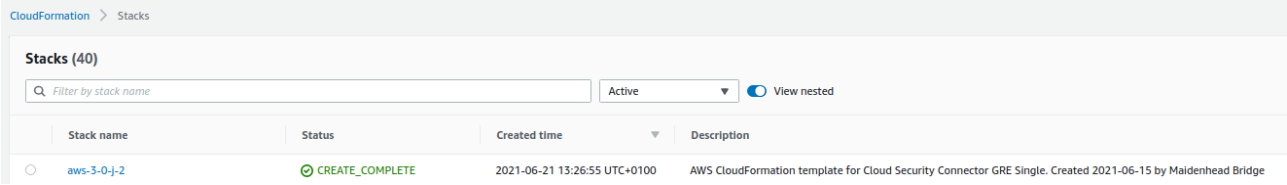


→ Click ***"Next"***

→ "Options Section": Click ***"Next"***

→ "Review": Click ***"Create Stack"***

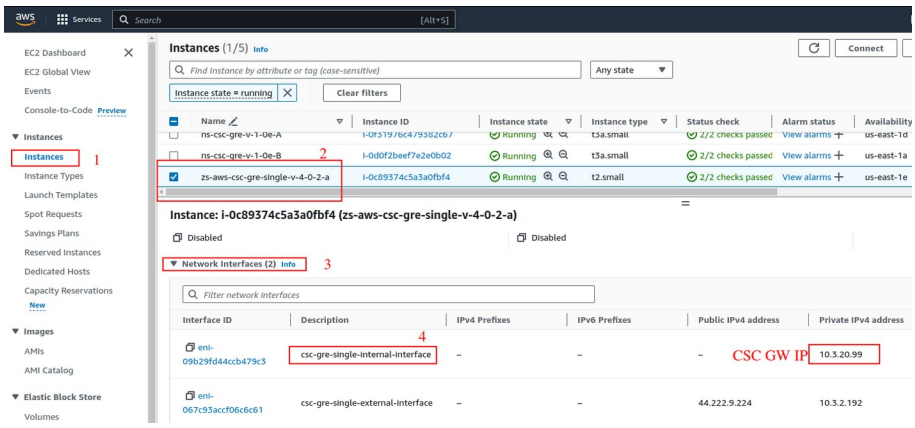
The Stack will show “status” CREATE_IN_PROGRESS, and after a while:



Done! Your CSC is deployed.

4.1.4 Accessing for first time to your CSC

1. Go to your EC2 Dashboard → Instances and select the CSC created. Go to "Networking" and scroll down.

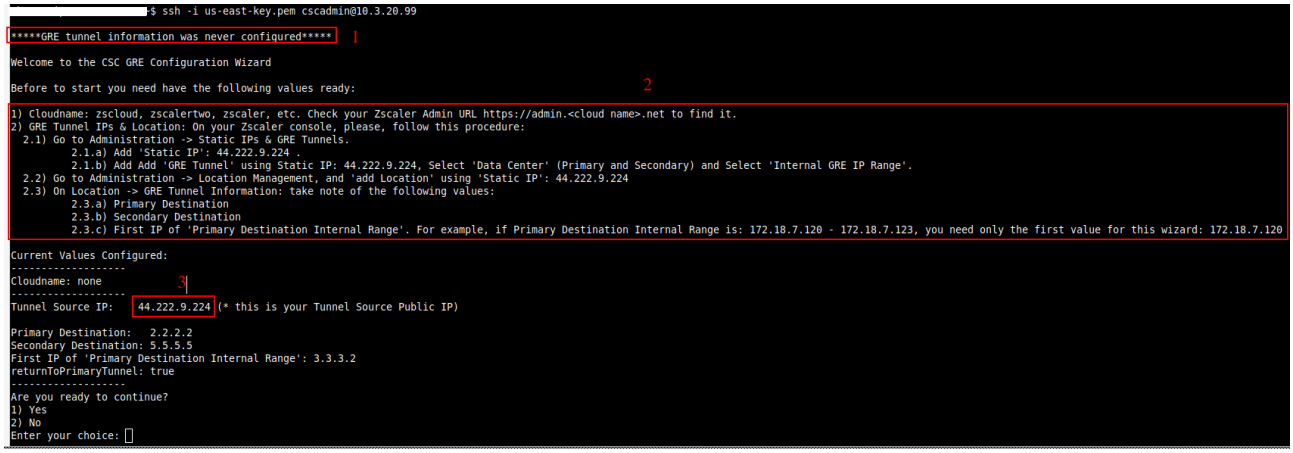


2. Find the "csc-gre-single-internal-interface" and take a look at the first Private IP address (CSC GW IP). This example is: 10.3.20.99

3. From a machine inside the VPC, ssh the CSC using the Key, like:

   ssh -i <keyname.pem> cscadmin@<CSC GW IP>

In our example, the value is $ ssh -i us-east-key.pem cscadmin@10.3.20.99



4. Your CSC is ready for the initial configuration. Just follow the instructions of the Configuration Wizard.

4.1.5 Initial Wizard Configuration

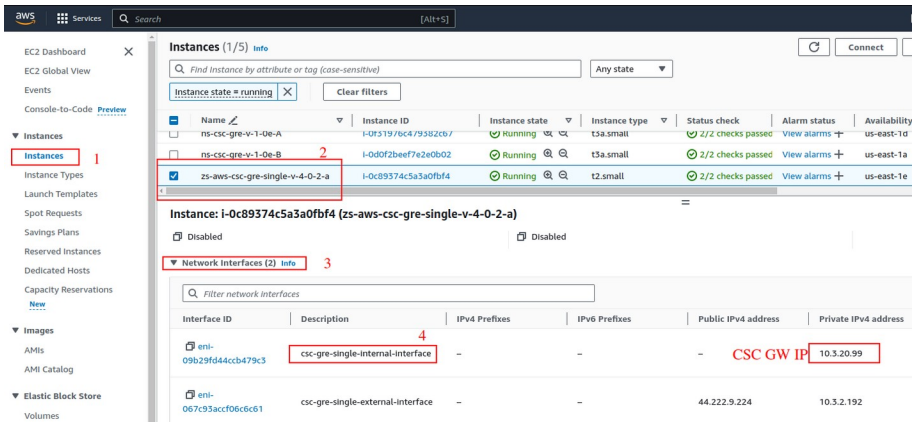
Please follow these instructions to run the initial configuration of the CSC GRE for AWS:

4.1.5.1 Short Version

1. On your Zscaler console create the "Static IP" (using "Tunnel Source Public IP") , "GRE Tunnel" and "Location".
2. Run the Wizard. Insert the values. Confirm and reboot.
3. Done!

4.1.5.2 Long Version (with Example)

In this example, after the CSC was launched, the values of my CSC are:



The internal IP (eth1) is 10.3.20.99. The initial wizard appears when SSHing the CSC.

In this example:

Key Name: us-east-key.pem

Username: cscadmin (use always "cscadmin")

CSC IP: 10.3.20.99

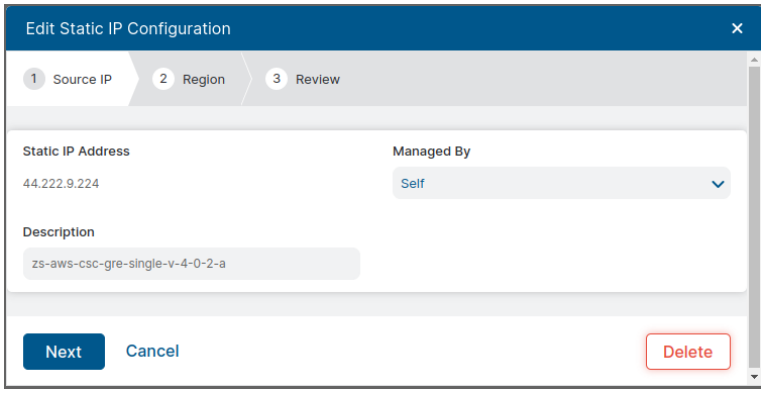
```
$ ssh -i  us-east-key.pem  cscadmin@10.3.20.99
```

```
[redacted]$ ssh -i us-east-key.pem cscadmin@10.3.20.99

*****GRE tunnel information was never configured*****   1

Welcome to the CSC GRE Configuration Wizard

Before to start you need have the following values ready:                                   2

1) Cloudname: zscloud, zscalertwo, zscaler, etc. Check your Zscaler Admin URL https://admin.<cloud name>.net to find it.
2) GRE Tunnel IPs & Location: On your Zscaler console, please, follow this procedure:
  2.1) Go to Administration -> Static IPs & GRE Tunnels.
          2.1.a) Add 'Static IP': 44.222.9.224 .
          2.1.b) Add Add 'GRE Tunnel' using Static IP: 44.222.9.224, Select 'Data Center' (Primary and Secondary) and Select 'Internal GRE IP Range'.
  2.2) Go to Administration -> Location Management, and 'add Location' using 'Static IP': 44.222.9.224
  2.3) On Location -> GRE Tunnel Information: take note of the following values:
          2.3.a) Primary Destination
          2.3.b) Secondary Destination
          2.3.c) First IP of 'Primary Destination Internal Range'. For example, if Primary Destination Internal Range is: 172.18.7.120 - 172.18.7.123, you need only the first value for this wizard: 172.18.7.120

Current Values Configured:
-------------------
Cloudname: none                3
-------------------
Tunnel Source IP:   44.222.9.224 (* this is your Tunnel Source Public IP)

Primary Destination:   2.2.2.2
Secondary Destination: 5.5.5.5
First IP of 'Primary Destination Internal Range': 3.3.3.2
returnToPrimaryTunnel: true
-------------------
Are you ready to continue?
1) Yes
2) No
Enter your choice:
```

As you can see in this example, the Tunnel Source Public IP is: 44.222.9.224

4.1.5.2.1 Create "Static IP"

From your Zscaler console, go to Administration → Static IPs & GRE Tunnels → Static IP → Click "Add Static IP Configuration"

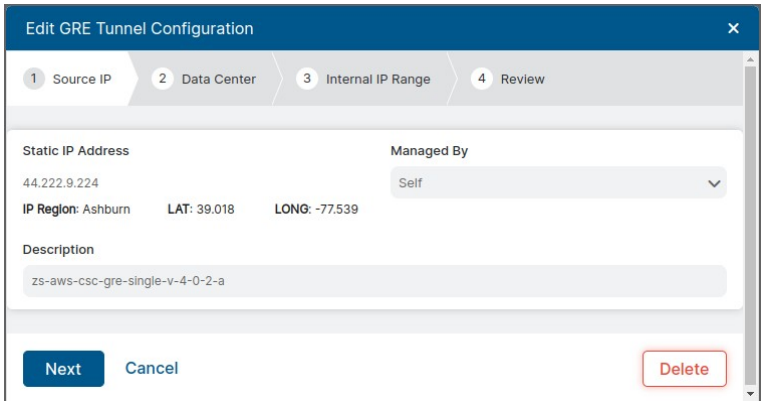


Click "Next", "Next" and "Save". Activate the changes.

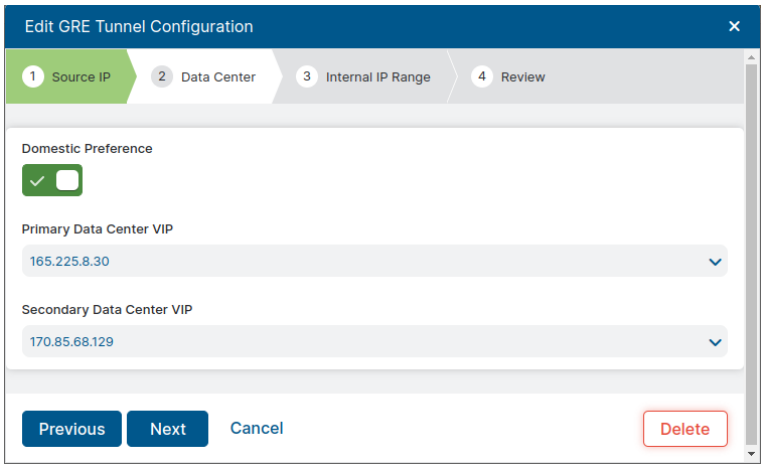
4.1.5.2.2 Create "GRE Tunnel"

From your Zscaler console, go to Administration → Static IPs & GRE Tunnels → GRE Tunnels → Add GRE Tunnel.

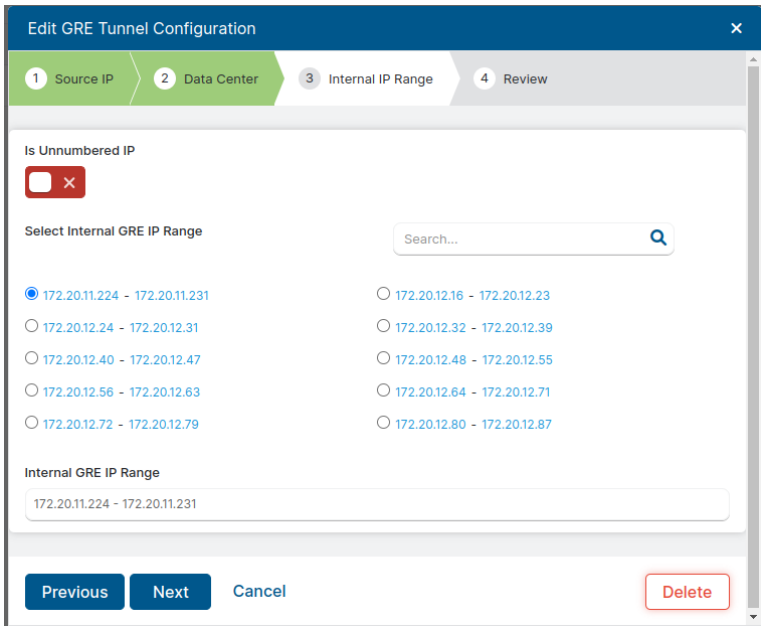
Select the "Static IP" created in the previous step.



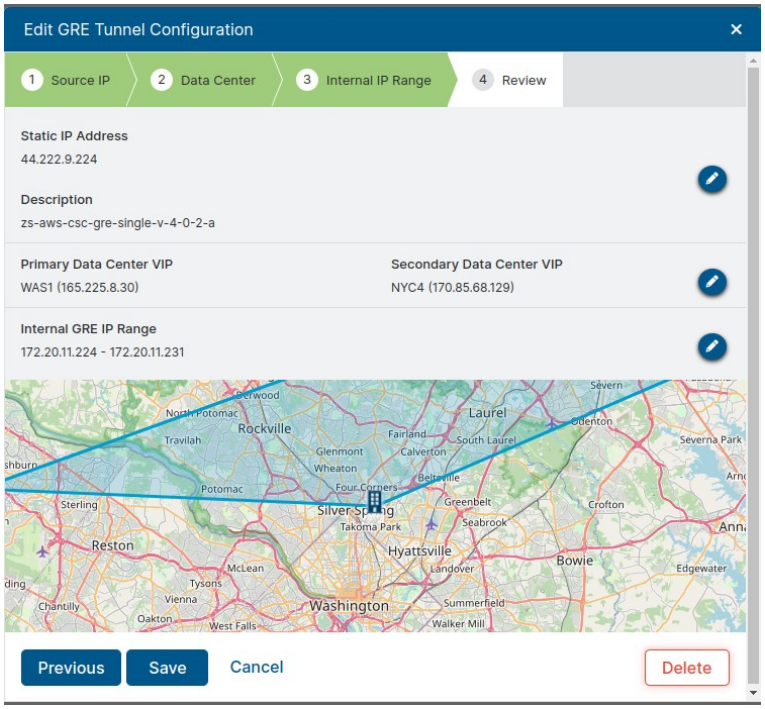
Click "Next" and select "Domestic Preference" if you prefer.



Click "Next" and "Select Internal GRE IP Range",



Click "Next"

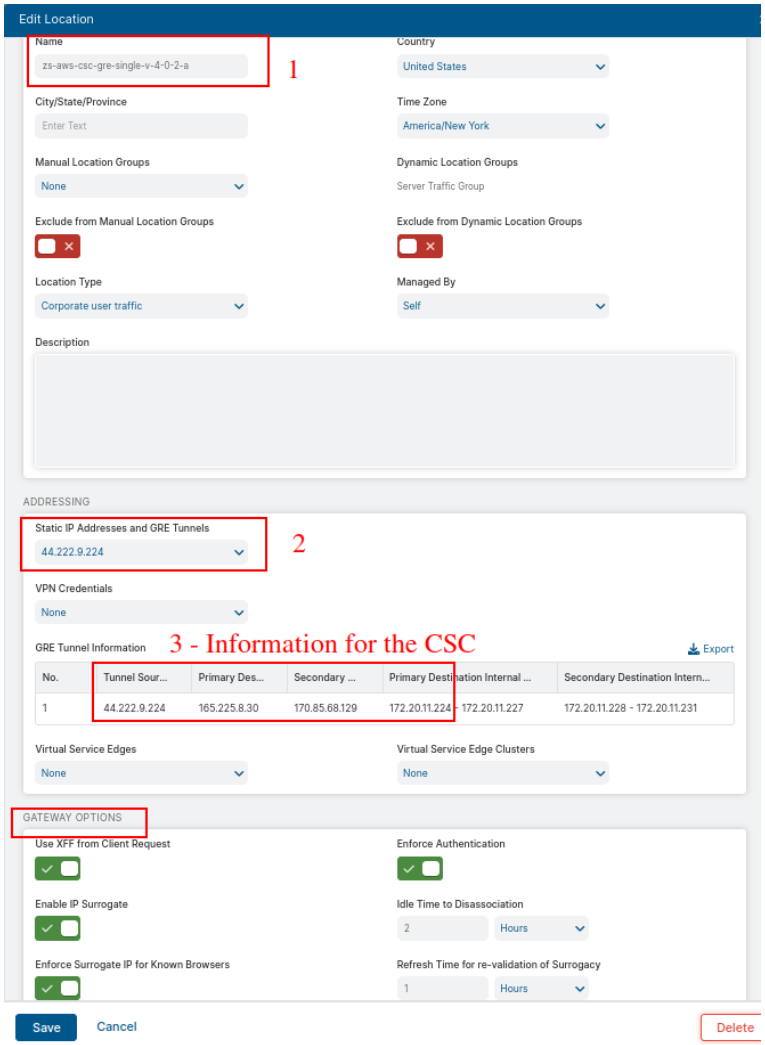


and "Save". Activate the Changes.

4.1.5.2.3 Create the Location

From your Zscaler console, go to Administration → Location Management → Add Location.

Here is the Location of this example:



Click "Save" and Activate Changes.

4.1.5.2.4 Run the Configuration Wizard

1. Select your cloud

```
-------------------
Are you ready to continue?
1) Yes
2) No
Enter your choice: 1
-------------------
Cloud Configuration

Your current Cloud is: none

Do you want to change the Cloud Name?
1) Yes
2) No
Enter your choice: 1

Please select or input your Cloud Name
1) zscalerthree
2) zscloud
3) zscalertwo
4) zscaler
5) zscalerone
6) zscalerbeta
7) zscalergov
8) Not in the list? Ingress Manually
9) Quit
Enter your choice: 1
-------------------
```

2. Enter the GRE tunnel values obtained after the Location creation: Primary Destination, Secondary Destination and First IP of "Primary Destination Internal Range."

Tunnel Sour...	Primary Des...	Secondary ...	Primary Desti
44.222.9.224	165.225.8.30	170.85.68.129	172.20.11.224

```
-------------------
GRE tunnels Configuration

Your current GRE tunnels configuration is:

Tunnel Source IP:    44.222.9.224

Primary Destination:    2.2.2.2
Secondary Destination:  5.5.5.5
First IP of 'Primary Destination Internal Range': 3.3.3.2
returnToPrimaryTunnel: true

Do you want to change the GRE tunnels configuration?
1) Yes
2) No
Enter your choice: 1

Please, Insert the GRE values:

Primary Destination (IP): 165.225.8.30
Secondary Destination (IP): 170.85.68.129
First IP of 'Primary Destination Internal Range': 172.20.11.224
```

3. Select "ReturnToPrimary" true or false.

```
'returnToPrimaryTunnel' variable:
Please select 'true' if you want the CSC to return to Primary tunnel (after 10 min of stability) when using Secondary tunnel.
Select 'false' if you want to remain using Secondary Tunnel and not to return to Primary.(Secondary will be nominated as 'new' Primary)

1) true
2) false
Enter your choice: 1
```

4. Finally, check and confirm the values:

```
Please confirm these values:
-------------------
Cloudname:  zscalerthree
-------------------
GRE tunnels IP values:

Tunnel Source IP (IP):  44.222.9.224

Primary Destination:   165.225.8.30
Secondary Destination: 170.85.68.129
First IP of 'Primary Destination Internal Range': 172.20.11.224
returnToPrimaryTunnel: true
-------------------
Do you want to implement these values? (The CSC will reboot)
1) Yes
2) No
Enter your choice:
```

Done! After the reboot, the CSC is ready for Production.

5 Accessing the CSC first time

SSH the CSC GW IP, and you will receive the Admin Console Menu.

5.1 Admin Console

```
Maidenhead Bridge

Cloud Security Connector GRE AWS for Zscaler - Admin Console

EC2 Instance ID :i-0c89374c5a3a0fbf4
AWS Availability Zone : us-east-1e
CSC Hostname : ip-10-3-2-192
Soft Version : 4.0.3

Please select an option by typing its number

Monitoring Tasks
1) Show Configuration and Status
2) Show Interfaces Traffic
3) Tcpdump, Traceroute/Latency Test and NetScanner
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) Configure DNS, SNMP, NTP and Timezone

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

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

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

Configuration Wizards
14) Configure Zscaler Nodes and GRE values
15) Switch Zscaler Tunnels - Primary / Secondary
16) High Availability Configuration

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

e) Exit
```

Check the tunnel status selecting "Show Configuration and Status"

5.2 Show Configuration and Status

```
GENERAL INFORMATION
CSC Hostname : ip-10-3-2-192
Availability Zone: us-east-1e
EC2 Instance id: i-0c89374c5a3a0fbf4 | Instance Type: t2.small | ami-id: ami-067bd9f666359655e
External Interface (eth0) Subnet-id: subnet-09fa17be9c948d4a9 | Interface-id: eni-067c93accf06c6c61 | Security-Group-id: sg-0650cdeb6403e915b
Internal Interface (eth1) Subnet-id: subnet-012afe90f3ee719ad | Interface-id: eni-09b29fd44ccb479c3 | Security-Group-id: sg-0b4c310462330c184
CSC date: Wed 7 Feb 14:13:43 UTC 2024
Soft version : 4.0.3

INTERFACES INFORMATION
External: Tunnel IP (eth0): 10.3.2.192/24 | Bypass Proxy Egress IP: 10.3.2.109 | Network Gateway: 10.3.2.1 is Alive
Internal: CSC GW IP (eth1): 10.3.20.99/24 | Network Gateway: 10.3.20.1 is Alive

TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.3.20.179:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.3.20.197:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP

ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 44.222.9.224
Bypass Proxy Public IP: 18.214.102.160

DNS INFORMATION
DNS Server (1) IP: 1.1.1.1 is Alive
DNS Server (2) IP: 1.0.0.1 is Alive

ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
GRE tunnels egress Public IP: 44.222.9.224
Primary Tunnel:
                ZEN Public IP: 165.225.8.30
                Tunnel IPs (local/zen): 172.20.11.225 / 172.20.11.226
Secondary Tunnel:
                ZEN Public IP: 170.85.68.129
                Tunnel IPs (local/zen): 172.20.11.229 / 172.20.11.230

TUNNEL STATUS
Primary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
Secondary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
returnToPrimaryTunnel: true

Tunnel Status: Primary tunnel is active since: Tue 6 Feb 11:11:59 UTC 2024

http://ip.zscaler.com INFORMATION
Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.69.13, via Public IP: 44.222.9.224

PROXY BYPASS
Proxy Bypass Mode: standard
Default Traffic Behaviour: Block
Proxy Bypass PAC URL is: https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac
Proxy Bypass Rules configured via URL: 8
Proxy Bypass Egress Interface 10.3.2.109 can reach test page (https://ip.maidenheadbridge.com) via Public IP 18.214.102.160

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 Tue 2024-02-06 11:11:12 UTC; 1 day 3h ago
Registration values: {"ManagedInstanceID":"mi-0c43f00e92e17735b","Region":"us-east-1"}

SYSLOG INFORMATION
Primary Syslog / SIEM (IP/TCP PORT): 10.63.1.10/5514 is Not reachable (ping failure)
Secondary Syslog / SIEM IP: Not configured
Traffic Logs (IP packets) are enabled.

HIGH AVAILABILITY Information
The HA service in NOT Active
```

6 Traffic forwarding to Zscaler ZIA and Bypasses.

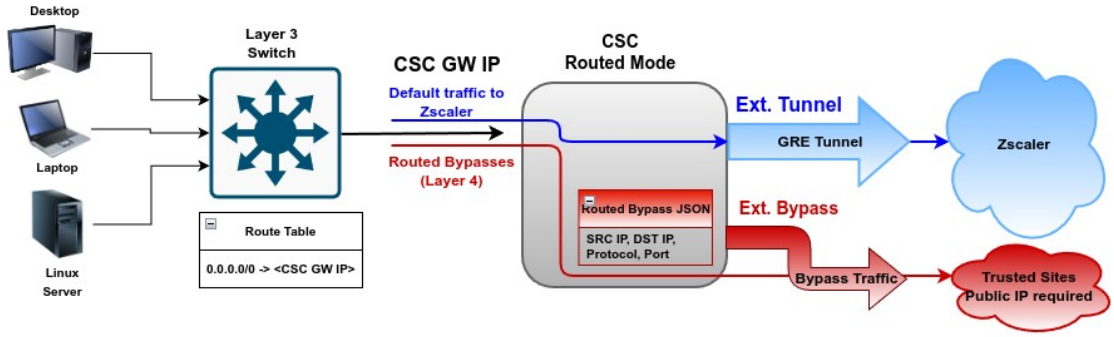
In Chapter 3 of this Administrator Guide, we showed the Network Diagrams of different scenarios of traffic forwarding and bypass traffic. In this chapter, we are going to dig into more detail about the configuration required.

We are going to analyse the following scenarios scenarios:

1. Routing all traffic via the Cloud Security Connector.
2. Using PAC files and/or Zscaler Client Connector
3. Using Explicit Proxy Settings
4. Special Cases: Using "Zscaler Global ZEN" and using proxy port tcp/8080.

6.1 Routing all traffic via the Cloud Security Connector

Network Diagram:



Setup:

This scenario is very simple to setup. The only task is to setup the default route to the internet (0.0.0.0/0) via the CSC GW IP. (CSC's eni-xxyy on AWS)

Traffic to Zscaler:

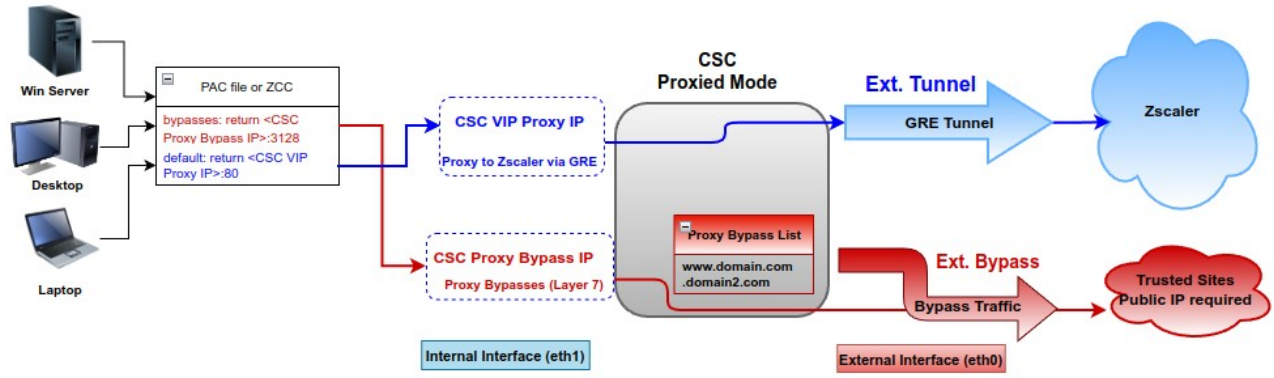
By default, all traffic will be via the GRE tunnel to Zscaler ZIA, and you can enable all Zscaler functionalities: Zscaler Cloud Firewall, Secure Web Gateway, etc.

Routed Bypass traffic:

You can bypass traffic by combining Source / Destination IP, Protocol (TCP, UDP, ICMP) and Port. Common destinations to bypass are Teams/Skype UDP real-time traffic and Windows Login destinations for conditional access rules. (See Routed Bypass Configuration in the specific section of this guide.)

6.2 Devices using PAC files or Zscaler Client Connector

Network Diagram:



Setup:

Devices with PAC Files: Distribute the PAC file URL via GPO.

PAC Example:

```
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: Variables (CSC VIP: 10.2.2.13, CSC Bypass: 10.2.2.14)
  var tozscaler = "PROXY 10.2.2.13:80";
  var bypassproxy = "PROXY 10.2.2.14:3128";

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

  // Bypass via CSC Public IPs (Examples)
  // Okta conditional access
  if ((shExpMatch(host, "*.okta.com")) ||
    (shExpMatch(host, "*.oktacdn.com")) ||
    (shExpMatch(host, "*.okta-emea.com")) ||
    (shExpMatch(host, "login.myOktaDomain.com")) ||
    // O365 Domains for ConditionalAccess
    (shExpMatch(host, "login.microsoftonline.com")) ||
    (shExpMatch(host, "login.microsoft.com")) ||
    (shExpMatch(host, "login.windows.net")) ||
    // My Trusted Sites
    (shExpMatch(host, "*.trustedSite-1.com")) ||
    (shExpMatch(host, "www.trustedSite-2.com")) ||
    // IP Test Page
    (shExpMatch(host, "ip.maidenheadbridge.com"))) {
    return bypassproxy;
  }
  // ==========================================================
  // Section 4: Default Traffic
  /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
  return tozscaler;
}
```

Devices with Zscaler Client Connector (ZCC):

1) Configure "Forwarding Profile" (Tunnel & Local Proxy) with a PAC file with the bypasses and point it to the CSC Proxy Bypass IP (return <CSC Proxy Bypass IP:3128>.

PAC for ZCC "Forwarding Profile"

Same PAC than for Devices, but changing the variable "tozscaler"

```
// ==========================================================
 // Section 2: Variables (CSC Bypass: 10.2.2.14)
 var tozscaler =  "PROXY ${ZAPP_LOCAL_PROXY}"
 var bypassproxy = "PROXY 10.2.2.14:3128";

 // ==========================================================
```

2) Configure "APP profile" with the "Forwarding Profile" and create a "Custom PAC" pointing the ZCC tunnel to the CSC VIP Proxy IP. (return <CSC VIP Proxy IP>:80 or 9400).

"Custom PAC" for ZCC "APP Profile"

Same than PAC for Devices, but removing Bypasses - Section 3, variable "bypassproxy", and adding public Zscaler Nodes to the variable "tozscaler" (OFF Corporate network condition)

```
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: Variables (CSC VIP: 10.2.2.13, off Corporate Network use Zscaler Public Nodes)
  var tozscaler = "PROXY 10.2.2.13:80; PROXY ${GATEWAY}:80; PROXY ${SECONDARY_GATEWAY}:80; DIRECT";

  // ==========================================================
  // Section 4: Default Traffic
  /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
  return tozscaler;
}
```

Traffic to Zscaler:

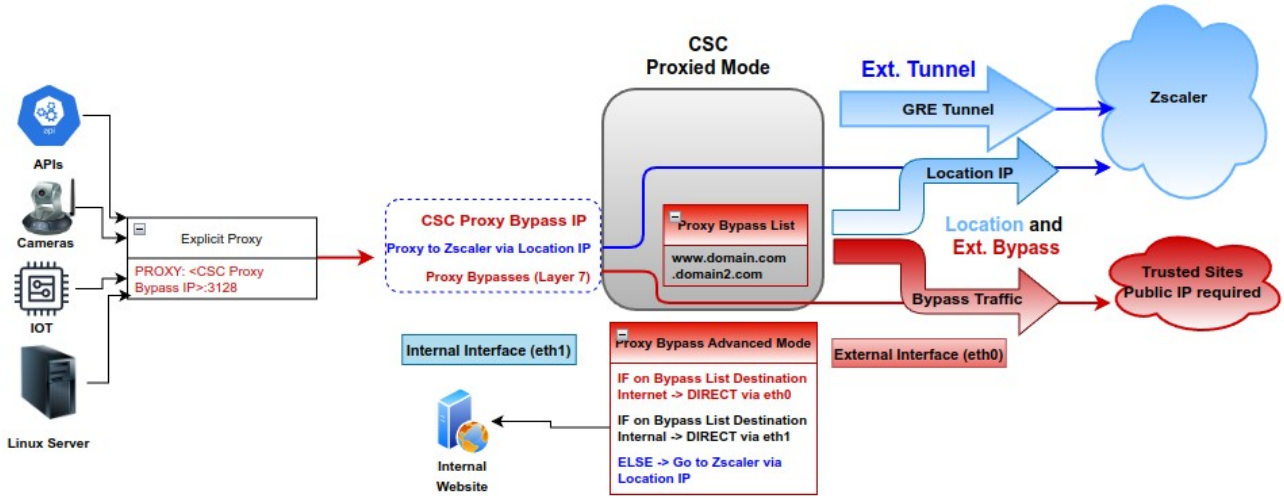
Devices with PAC Files: The default traffic will go via the CSC VIP.
Devices with Zscaler Client Connector (ZCC): The ZCC tunnel points to the CSC VIP (On Corporate Network). When the user is OFF Corporate Network, the tunnel will connect to the Zscaler Public Node.

Proxied Bypass traffic:

Devices with PAC Files: The host domains configured on the proxy bypass list will hit the CSC Proxy Bypass IP, and the CSC will allow them to go directly to the Internet. Note that you need to configure the PAC URL on the CSC. (See Proxied Bypass Configuration in the specific section of this guide.)
Devices with Zscaler Client Connector (ZCC): Same than previous.

6.3 Devices using Explicit Proxy Settings

Network Diagram:



Setup:

On the CSC, you need to enable: "Proxy Bypass Advanced Mode" and create a "Location IP" on your Zscaler console using the public IP that is nating the 2nd external IP of the CSC (External Bypass & PriCPA; see FW rules section). On the "Location IP" enable "Use XFF from Client Request" for full visibility of Internal IPs. Also, you need to set the CSC's bypass list with the domains you want to send directly to the Internet and, if needed, internal domains.

The configuration of the CSC is via JSON file. You can host the JSON file and setup the URL on the CSC, or you can paste the JSON file on the CSC.

Advanced Mode JSON file

```
{
  "model": "csc-gre-zs-vm",
  "type": "proxyBypassAdvanced",
  "version": "1.0",
  "help": ".domain.com matches domain.com and any subdomain of <>.domain.com. Do not use asterisk '*'",
  "proxyBypassRules": {
    "internalSites": [
      ".domainInternal.com",
      "fqdn-internal.com"
    ],
    "externalSites": [
      ".externalDomain.com",
      "fqdn-external.com",
      "ip.maidenheadbridge.com",
      "ipinfo.io"
    ]
  }
}
```

On your devices, you need to setup the explicit Proxy for HTTP and HTTPS traffic. For example, in a Linux Server is:

Settings Variables for http, https and no_proxy[2]
export[3] http_proxy=http://<CSC Proxy Bypass IP>:3128 export https_proxy=http://<CSC Proxy Bypass IP>:3128 export no_proxy= <your local domains>[4]

Traffic to Zscaler:

By default, the CSC will send all destination domains **not** in the bypass list to Zscaler.

Proxied Bypass traffic:

Domains in the bypass list will be routed externally or internally according the DNS resolution.

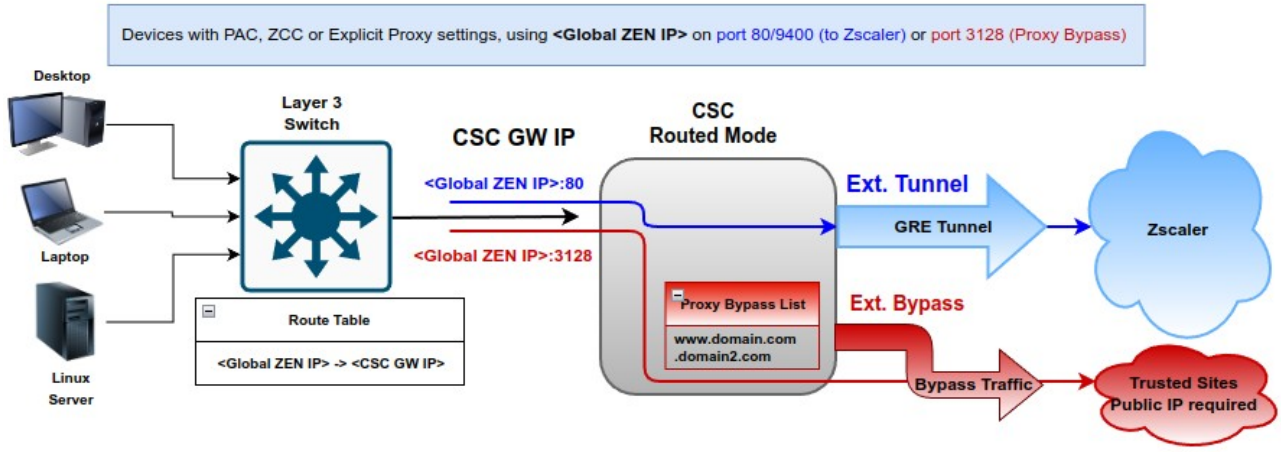
2 Add this lines to "/etc/environment" to make this changes permanent.
3 Use command $unset <variable name> to clear the values.
4 IF "no_proxy" variable is blank, we sure to add your internal domains on the CSC bypass list.

6.4 Special cases:

6.4.1 Using "Global ZEN IP Addresses" as Proxy IP

The CSC can intercept the "Global ZEN IP Addresses" when this destination is routed via the CSC GW IP. If the destination port is 80/9400, the traffic will travel via the GRE tunnel to Zscaler. If the destination port is 3128, the traffic will be sent to the Proxy Bypass.

This method is commonly used in "no default route to the internet" scenarios.



Global ZEN IP Addresses (8)

Zscaler has configured several Global, or Ghost, ZIA Public Service Edges (formerly Zscaler Enforcement Nodes or ZENs) across its clouds. These Public Service Edge addresses do not listen for traffic but are dummy addresses that every Public Service Edge knows about. They can be useful when working in no default route environments. To learn more, see Implementing Zscaler in No Default Route Environments.

Global Zen IP Addresses			Copy IPs
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

6.4.2 Using TCP port 8080.

Zscaler (ZIA) Public Service Edges accept web requests on ports 80, 443, 9400, 9480, and 9443 **but not in port 8080.**

The CSC provides support for port tcp/8080. You can use proxy: <CSC VIP Proxy>:8080 or <Global Zen IP Address>:8080 and the CSC will convert to a port accepted by Zscaler ZIA.

If you have hardcoded or configured your proxy settings with port 8080, the CSC is the solution to the above mentioned problem.

7 Testing traffic to Zscaler and Bypass

The following test is using a Windows PC, with the following PAC file:

Test PAC file

```
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: Variables (CSC VIP: 172.19.0.61, CSC Bypass: 172.19.0.62)
   var tozscaler = "PROXY 172.19.0.61:80";
   var bypassproxy = "PROXY 172.19.0.62:3128";

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

   // O365 Domains for ConditionalAccess
   if ((shExpMatch(host, "login.microsoftonline.com")) ||
      (shExpMatch(host, "login.microsoft.com")) ||
      (shExpMatch(host, "login.windows.net")) ||
      // IP Test Page
      (shExpMatch(host, "ipinfo.io"))) {
      return bypassproxy;
   }
   // ==========================================================
   // Section 4: Default Traffic
   /* Default Traffic Forwarding. Forwarding to Zen on port 80, but you can use port 9400 also */
   return tozscaler;
}
```

"Show Configuration and Status" menu provide the values of CSC VIP and CSC Proxy Bypass:

```
TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 172.19.0.61:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 172.19.0.62:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP
```

The CSC has configured the PAC file for bypasses.

```
Selection: 8

Proxy Bypass Mode: standard

This is the list of current Domains configured:

login.microsoftonline.com
login.microsoft.com
login.windows.net
ipinfo.io
```

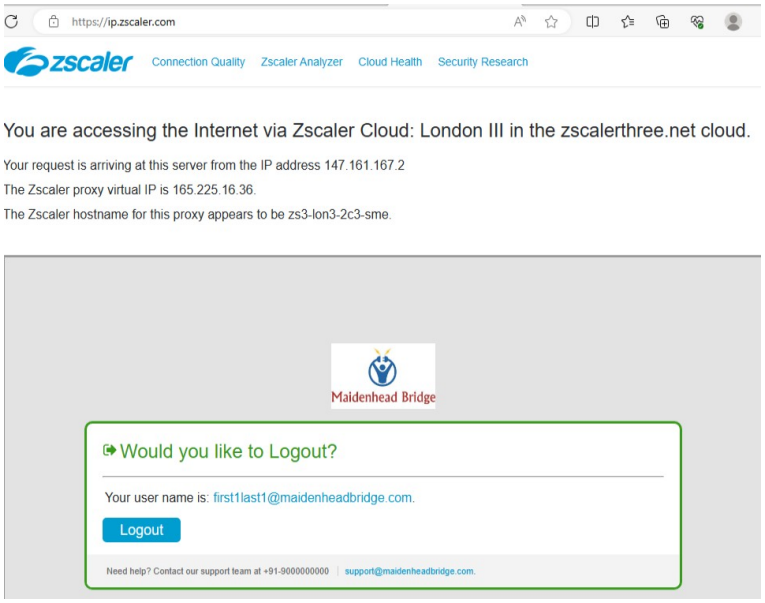
7.1 To Zscaler traffic test

7.1.1 Using a browser

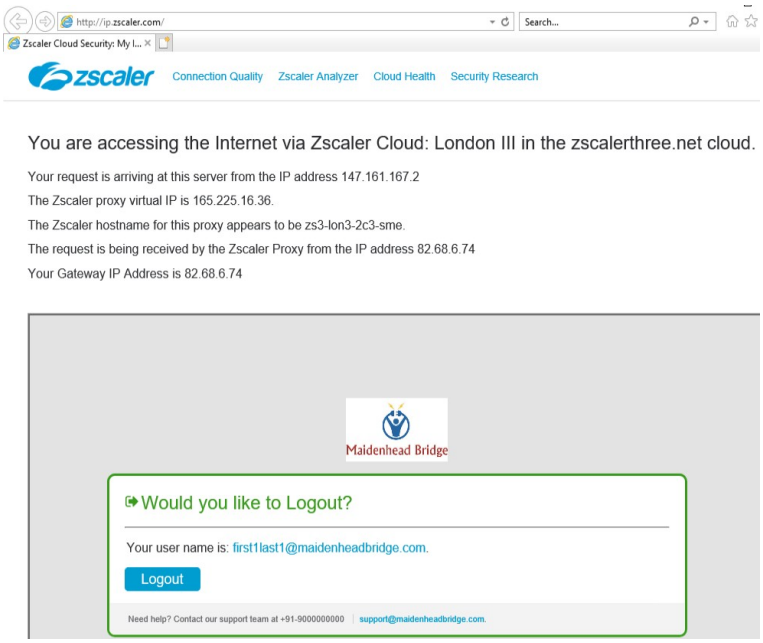
- http://ip.zscaler.com test page.

NOTE: The page ip.zscaler.com doesn't provides the same information in all browsers.

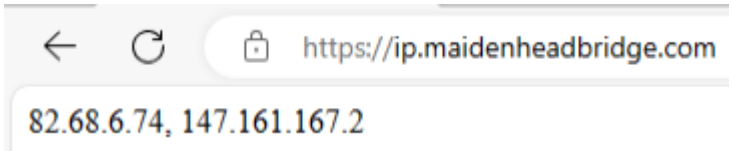
Using Edge:



Using IE:



- https://ip.maidenheadbridge.com test page.



7.1.2 Using Curl Command via CMD

Open CMD and run the following command:

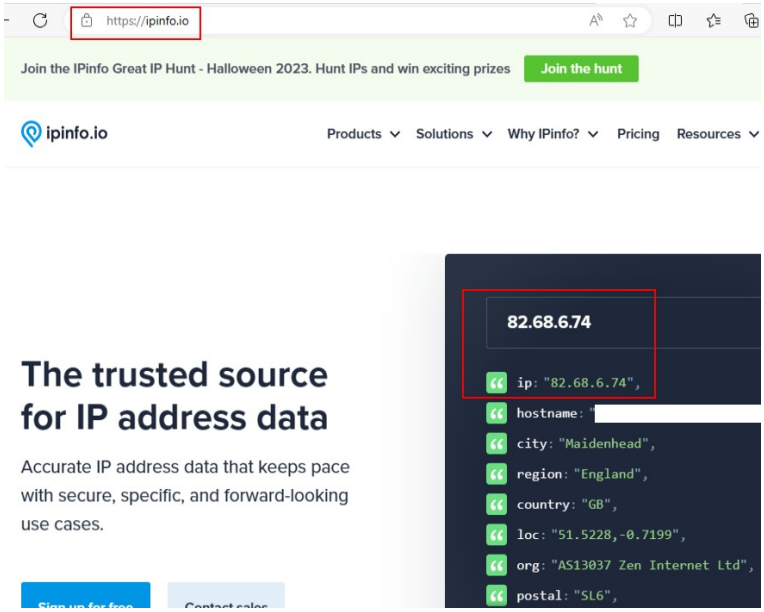
> curl -s --proxy http://<CSC VIP>:80 http://ip.zscaler.com | findstr "You"

```
I:\Users\[redacted]>curl -s --proxy http://172.19.0.61:80 http://ip.zscaler.com | findstr "You"
<div class="headline">You are accessing the Internet via Zscaler Cloud: London III in the zscalerthree.net cloud.</div>
<div class="details" style="margin-top: 20px">Your request is arriving at this server from the IP address <span class="detailOutput">147.161.167.2</span></div>
<div class="details">Your Gateway IP Address is <span class="detailOutput">82.68.6.74</span></div>
```

7.2 Bypass Traffic test

7.2.1 Using a Browser

Go to the bypassed domain: "ipinfo.io". You will see your local public IP.



7.2.2 Using Curl Command via CMD

Open CMD and run the following command:

> curl -s --proxy http://<CSC Bypass Proxy IP>:3128 http://ipinfo.io

```
C:\Users\[redacted]>curl -s --proxy http://172.19.0.62:3128 http://ipinfo.io
{
  "ip": "82.68.6.74",
  "hostname": "[redacted]",
  "city": "Maidenhead",
  "region": "England",
  "country": "GB",
  "loc": "51.5228,-0.7199",
  "org": "AS13037 Zen Internet Ltd",
  "postal": "SL6",
  "timezone": "Europe/London",
  "readme": "https://ipinfo.io/missingauth"
}
```

7.3 Speed test

You can run "Speed Test" from the SSH Console of the CSC. This test runs via the GRE tunnel active.

```
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

Retrieving speedtest.net configuration...
Testing from Amazon.com (44.222.9.224)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by GSL Networks (Ashburn, VA) [0.81 km]: 3.535 ms
Testing download speed.................................................
Download: 2526.46 Mbit/s
Testing upload speed...................................................
Upload: 272.16 Mbit/s
```

8 The Cloud Security Connector Admin Console:

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

Access to SSH Admin Console: $ssh -i <SSH Key> **cscadmin@<CSC GW IP>**

```
Maidenhead Bridge

Cloud Security Connector GRE AWS for Zscaler - Admin Console

EC2 Instance ID :i-0c89374c5a3a0fbf4
AWS Availability Zone : us-east-1e
CSC Hostname : ip-10-3-2-192
Soft Version : 4.0.3

Please select an option by typing its number

Monitoring Tasks
1) Show Configuration and Status
2) Show Interfaces Traffic
3) Tcpdump, Traceroute/Latency Test and NetScanner
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) Configure DNS, SNMP, NTP and Timezone

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

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

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

Configuration Wizards
14) Configure Zscaler Nodes and GRE values
15) Switch Zscaler Tunnels - Primary / Secondary
16) High Availability Configuration

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

e) Exit
```

The Main Sections are:

- **Monitoring Tasks:** To check configuration, statuses, real-time traffic, tcpdump, traceroute and speed..
- **CSC Admin Tasks:** To register the CSC for AWS management, manage administrators, restrict SSH, configure radius, DNS, SNMP, NTP and time-zone.
- **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, GRE Values, Switch tunnels and configure High Availability.
- **Private Cloud Private Access (PriCPA):** Show Configuration and Statuses, create Local Configuration, configure priCPA peers and add Remote Management Networks.

8.1 Monitoring Tasks

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

8.1.1 Show Configuration and Status

```
GENERAL INFORMATION
CSC Hostname : ip-10-3-2-192
Availability Zone: us-east-1e
EC2 Instance id: i-0c89374c5a3a0fbf4 | Instance Type: t2.small | ami-id: ami-067bd9f666359655e
External Interface (eth0) Subnet-id: subnet-09fa17be9c948d4a9 | Interface-id: eni-067c93accf06c6c61 | Security-Group-id: sg-0650cdeb6403e915b
Internal Interface (eth1) Subnet-id: subnet-012afe90f3ee719ad | Interface-id: eni-09b29fd44ccb479c3 | Security-Group-id: sg-0b4c310462330c184
CSC date: Wed 7 Feb 14:13:43 UTC 2024
Soft version : 4.0.3

INTERFACES INFORMATION
External: Tunnel IP (eth0): 10.3.2.192/24 | Bypass Proxy Egress IP: 10.3.2.109 | Network Gateway: 10.3.2.1 is Alive
Internal: CSC GW IP (eth1): 10.3.20.99/24 | Network Gateway: 10.3.20.1 is Alive

TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.3.20.179:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.3.20.197:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP

ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 44.222.9.224
Bypass Proxy Public IP: 18.214.102.160

DNS INFORMATION
DNS Server (1) IP: 1.1.1.1 is Alive
DNS Server (2) IP: 1.0.0.1 is Alive

ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
GRE tunnels egress Public IP: 44.222.9.224
Primary Tunnel:
                ZEN Public IP: 165.225.8.30
                Tunnel IPs (local/zen): 172.20.11.225 / 172.20.11.226
Secondary Tunnel:
                ZEN Public IP: 170.85.68.129
                Tunnel IPs (local/zen): 172.20.11.229 / 172.20.11.230

TUNNEL STATUS
Primary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
Secondary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
returnToPrimaryTunnel: true

Tunnel Status: Primary tunnel is active since: Tue 6 Feb 11:11:59 UTC 2024

http://ip.zscaler.com INFORMATION
Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.69.13, via Public IP: 44.222.9.224

PROXY BYPASS
Proxy Bypass Mode: standard
Default Traffic Behaviour: Block
Proxy Bypass PAC URL is: https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac
Proxy Bypass Rules configured via URL: 8
Proxy Bypass Egress Interface 10.3.2.109 can reach test page (https://ip.maidenheadbridge.com) via Public IP 18.214.102.160

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 Tue 2024-02-06 11:11:12 UTC; 1 day 3h ago
Registration values: {"ManagedInstanceID":"mi-0c43f00e92e17735b","Region":"us-east-1"}

SYSLOG INFORMATION
Primary Syslog / SIEM (IP/TCP PORT): 10.63.1.10/5514 is Not reachable (ping failure)
Secondary Syslog / SIEM IP: Not configured
Traffic Logs (IP packets) are enabled.

HIGH AVAILABILITY Information
The HA service in NOT Active
```

8.1.1.1 GENERAL INFORMATION

This section contains general information about the CSC:

```
GENERAL INFORMATION
CSC Hostname : ip-10-3-2-192
Availability Zone: us-east-1e
EC2 Instance id: i-0c89374c5a3a0fbf4 | Instance Type: t2.large | ami-id: ami-067bd9f666359655e
External Interface (eth0) Subnet-id: subnet-09fa17be9c948d4a9 | Interface-id: eni-067c93accf06c6c61 | Security-Group-id: sg-0650cdeb6403e915b
Internal Interface (eth1) Subnet-id: subnet-012afe90f3ee719ad | Interface-id: eni-09b29fd44ccb479c3 | Security-Group-id: sg-0b4c310462330c184
CSC date: Wed 7 Feb 15:01:56 UTC 2024
Soft version : 4.0.3
```

Important: Please, note the “Interface-id:” value. You will need it if routing traffic via the CSC.

8.1.1.2 INTERFACES INFORMATION

This section contains the interfaces information and you can check here is the Gateways, external and internal, are reachable. (Network Gateway is Alive)

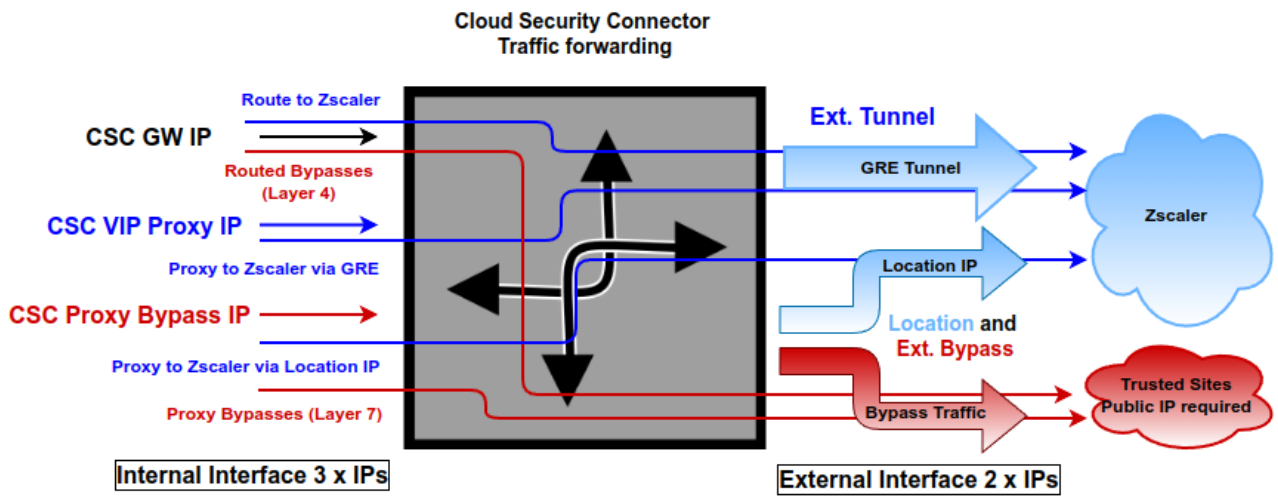
```
INTERFACES INFORMATION
External: Tunnel IP (eth0): 10.3.2.192/24 | Bypass Proxy Egress IP: 10.3.2.109 | Network Gateway: 10.3.2.1 is Alive
Internal: CSC GW IP (eth1): 10.3.20.99/24 | Network Gateway: 10.3.20.1 is Alive
```

8.1.1.3 TRAFFIC REDIRECTION Options.

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

```
TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 10.3.20.179:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 10.3.20.197:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP
```

The following diagram shows the multiple options available:



- See section "Traffic forwarding to Zscaler ZIA and Bypasses." for more details.

8.1.1.4 ELASTIC (PUBLIC) IPs INFORMATION

This section displays the EIP used for the GRE tunnel and Bypasses.

```
ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 44.222.9.224
Bypass Proxy Public IP: 18.214.102.160
```

8.1.1.5 DNS INFORMATION

This section displays the DNS information.

```
DNS INFORMATION
DNS Server (1) IP: 1.1.1.1 is Alive
DNS Server (2) IP: 1.0.0.1 is Alive
```

8.1.1.6 ZSCALER INFORMATION

This section shows the GRE tunnels IP information.

```
ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
GRE tunnels egress Public IP: 44.222.9.224
Primary Tunnel:
                ZEN Public IP: 165.225.8.30
                Tunnel IPs (local/zen): 172.20.11.225 / 172.20.11.226
Secondary Tunnel:
                ZEN Public IP: 170.85.68.129
                Tunnel IPs (local/zen): 172.20.11.229 / 172.20.11.230
```

8.1.1.7 TUNNEL STATUS

This section shows the Keepalives statuses and the Tunnel status.

```
TUNNEL STATUS
Primary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
Secondary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
returnToPrimaryTunnel: true

Tunnel Status: Primary tunnel is active since: Wed 7 Feb 14:44:36 UTC 2024
```

8.1.1.8 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.

```
http://ip.zscaler.com INFORMATION
Node: Washington DC in the zscalerthree.net cloud. ZEN Instance IP: 136.226.107.81, via Public IP: 44.222.9.224
```

8.1.1.9 PROXY BYPASS

This section shows the Proxy Bypass Settings: Standard mode or Advanced mode.

Standard mode:

```
PROXY BYPASS
Proxy Bypass Mode: standard
Default Traffic Behaviour: Block
Proxy Bypass PAC URL is: https://pac.zscalerthree.net/maidenheadbridge.com/zs-cgc001001.pac
Proxy Bypass Rules configured via URL: 4
Proxy Bypass Egress Interface 192.168.1.61 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.74
```

Advanced mode:

```
PROXY BYPASS
Proxy Bypass Mode: advanced
Default Traffic Behaviour: To Zscaler - autoPrimary (165.225.16.37) / autoSecondary (147.161.141.129)
Proxy Bypass JSON file URL is: https://mhb-csc-pac.s3.amazonaws.com/proxyBypassRulesFile.json
Proxy Bypass Rules Internal Rules configured via JSON file URL: 2
Proxy Bypass Rules External Rules configured via JSON file URL: 4
Proxy Bypass Egress Interface 192.168.1.61 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.74
```

8.1.1.10 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-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
```

8.1.1.11 AWS SSM AGENT

This section shows the status of the AWS SSM Agent.

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Mon 2023-11-06 18:06:33 UTC; 1 day 15h ago
Registration values: {"ManagedInstanceID":"mi-0f8fcb40f04117844","Region":"eu-west-2"}
```

8.1.1.12 SYSLOG INFORMATION

When configured, this section will show the IP/s and TCP port of your Syslog/SIEM server and if Traffic Logs are enabled. (Note: Systems Logs are always enabled)

```
SYSLOG INFORMATION
Primary Syslog / SIEM (IP/TCP PORT): 10.63.1.10/5514 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

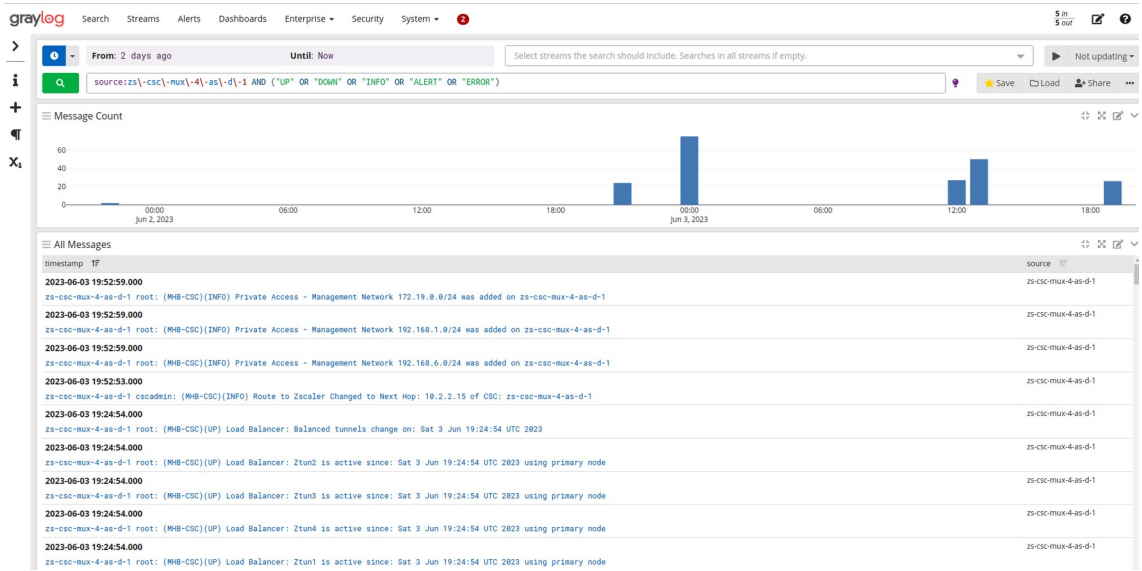
Traffic Logs:

- ALLOW
- BLOCK

8.1.1.12.1 System Logs example:

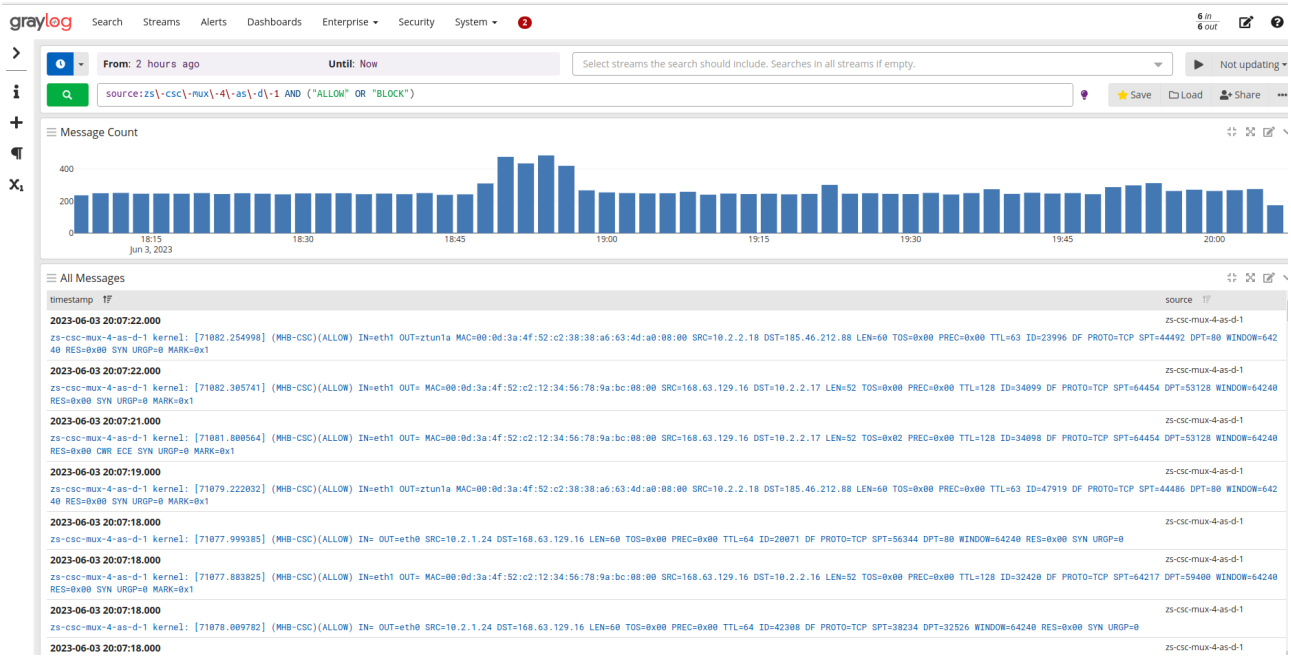
To obtain your System Logs, you can search by CSC name plus the following TAG. For example:

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



8.1.1.12.2 Traffic Logs example:

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



8.1.1.13 HIGH AVAILABILITY Information

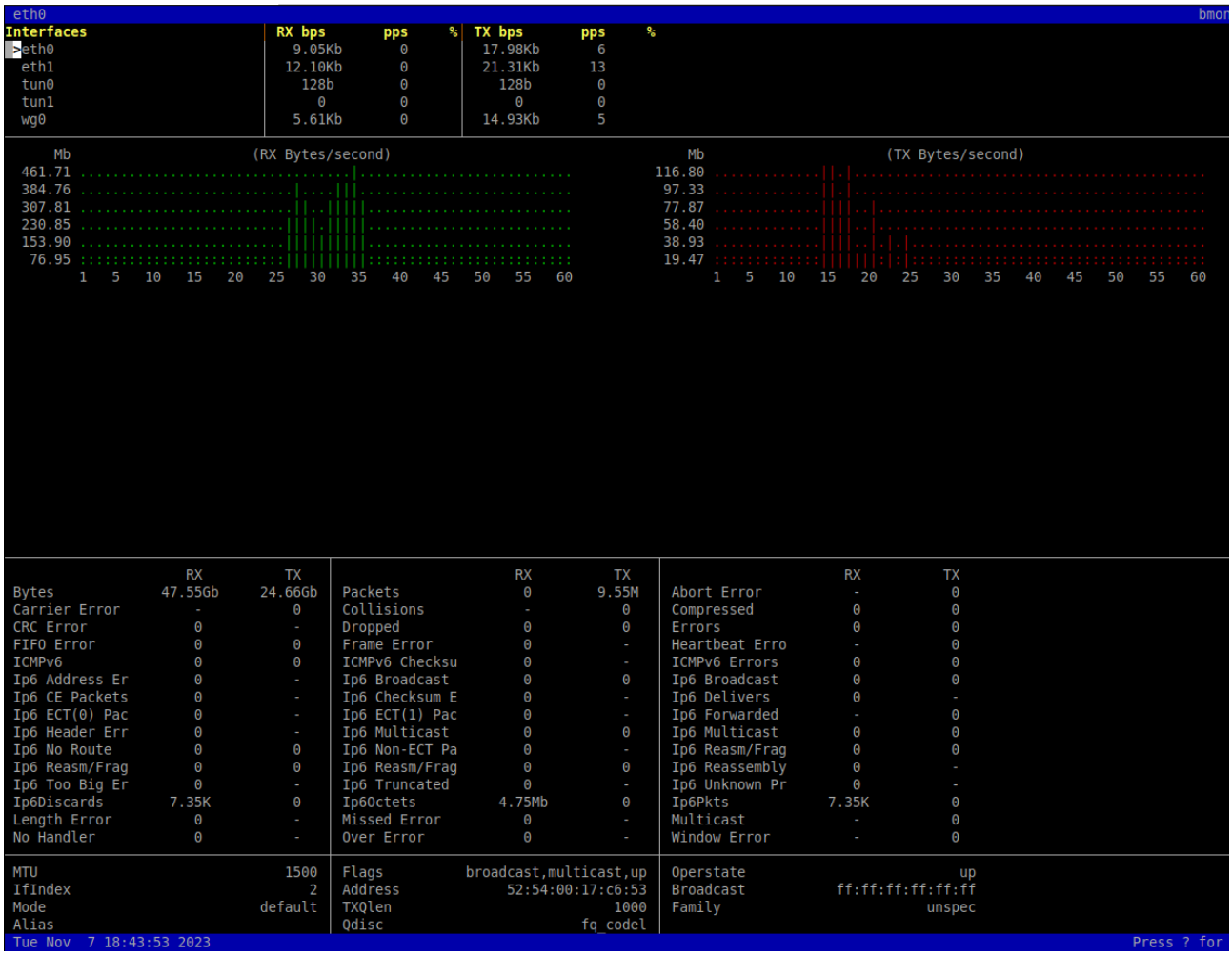
This section all the information when the CSC are configured on HA pair:

```
HIGH AVAILABILITY Information
The HA service is:  active (running) since Mon 2022-01-10 20:46:12 UTC; 3min 53s ago
IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
The Default Route to Internet is using this Gateway (Target): eni-0a75cf038298d8e31 (this CSC)
Current values configured are:
Route Table/s Configured (Qty)= 1  (using VPC: vpc-0f32a676)
Route Table ID= rtb-d090c8a8
Instance ID of other CSC in the pair= i-0f15f29fcc4d69eb6
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification
Private Access Public IP= 23.23.210.207

Press Enter to continue...
```

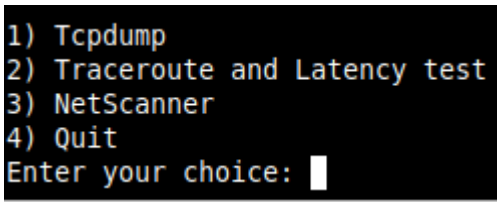
- If HA service is active.
- The IAM role in use.
- The current “eni-xxyy” that is the default GW to the Internet for the Route Table/s.
- Amount of Route Tables configured and VPC in use.
- The Route table ID/s.
- Which is the Instance ID of other CSC on the HA pair.
- The SNS message used for notification.
- PriCPA Floating Public IP.

8.1.2 Show Interfaces Traffic

Use this section to see the traffic in real time.



8.1.3 Tcpdump, Traceroute/Latency Test and NetScanner



8.1.3.1 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
```

8.1.3.2 Traceroute and Latency Test

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

```
My TraceRoute (MTR) Test Report
This test does:
- MTR (TCP/80) DIRECT to the Primary ZEN and Secondary ZEN
- When the tunnel is UP, a MTR Reverse Path test from the active ZEN to your Public IP
NOTE: Max Hops is equal 30. This test can take a while

Testing Primary ZEN 165.225.48.12
Start: 2021-06-23T13:31:30+0000
HOST: ip-172-31-96-70           Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  2.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  3.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  4.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  5.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  6.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  7.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  8.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
  9.|-- ???                       100.0    10    0.0   0.0   0.0   0.0   0.0
 10.|-- 244.0.4.83                10.0%    10    0.4   3.5   0.4  27.8   9.1
 11.|-- 240.0.36.21                0.0%    10    0.4   0.4   0.4   0.5   0.0
 12.|-- 242.0.162.49               0.0%    10    0.4   1.2   0.4   7.6   2.3
 13.|-- 52.93.28.193               0.0%    10    0.9   1.1   0.4   2.1   0.5
 14.|-- 100.100.4.18               0.0%    10    1.0   1.2   1.0   1.9   0.3
 15.|-- eqix-was1-r2.zscaler9.net  0.0%    10    1.3   1.4   1.1   1.6   0.1
 16.|-- 165.225.48.12              0.0%    10    2.8   3.1   1.5   7.1   1.6

Testing Secondary ZEN 165.225.38.51
Start: 2021-06-23T13:31:45+0000
HOST: ip-172-31-96-70                    Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  2.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  3.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  4.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  5.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  6.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  7.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  8.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
  9.|-- ???                                100.0    10    0.0   0.0   0.0   0.0   0.0
 10.|-- 244.0.4.223                         0.0%    10    0.4   3.5   0.4  31.6   9.9
 11.|-- 240.0.36.23                         0.0%    10    0.4   0.4   0.4   0.5   0.0
 12.|-- 242.0.162.161                       0.0%    10    0.7   0.5   0.4   0.7   0.1
 13.|-- 242.0.163.33                        0.0%    10   13.1   2.8   0.4  13.1   4.1
 14.|-- 100.100.28.98                       0.0%    10    1.8   3.1   1.0  11.1   3.9
 15.|-- 100.95.7.65                         0.0%    10    1.8   1.9   1.2   2.8   0.5
 16.|-- 52.93.114.178                       0.0%    10    2.0  11.0   1.7  25.2   8.7
 17.|-- 100.95.23.161                       0.0%    10    2.7   4.8   1.4  28.7   8.4
 18.|-- 52.93.114.176                      50.0%    10    2.3   7.8   2.2  18.7   7.8
 19.|-- 54.239.109.153                      0.0%    10    8.0   4.9   2.1   8.0   2.4
 20.|-- 4.14.222.30                        40.0%    10    7.2   6.3   2.3  12.0   3.6
 21.|-- ae-2-3610.edge5.Newark1.Level3.net 10.0%    10    6.8   7.5   6.8   8.5   0.6
 22.|-- 4.14.222.30                         0.0%    10    7.1   7.3   6.9   7.7   0.3
 23.|-- 165.225.38.51                       0.0%    10    7.5   7.8   7.1   8.3   0.4


Reverse path from: 165.225.48.12 to your Public IP: 54.163.234.160
Start: 2021-06-23T13:32:02+0000
HOST: ip-172-31-96-70                Loss%   Snt   Last   Avg  Best  Wrst StDev
  1.|-- ip-172-18-96-106.ec2.internal  0.0%    10    4.4   3.3   2.5   4.4   0.6
  2.|-- 165.225.48.3                   0.0%    10    4.0   4.3   3.4   5.9   0.7
  3.|-- equinix02-iad2.amazon.com      0.0%    10    4.7   4.5   3.4   5.4   0.7
  4.|-- ???                          100.0    10    0.0   0.0   0.0   0.0   0.0
  5.|-- ???                          100.0    10    0.0   0.0   0.0   0.0   0.0
  6.|-- 52.93.28.174                   0.0%    10    4.8   5.6   4.6   8.6   1.2
  7.|-- ???                          100.0    10    0.0   0.0   0.0   0.0   0.0
```

8.1.3.3 NetScanner

This test scans Network/s (or IPs) configured behind this node. This test is handy in finding active apps.

```
1) Tcpdump
2) Traceroute and Latency test
3) NetScanner
4) Quit
Enter your choice: 3

NetScanner Test: This test scans Network/s (or IPs) configured behind this node.

The configured Networks are:

10.3.20.0/24

Please, select a Network to scan or Host IP
NOTE: Network scan uses PING to detect hosts. If you want to scan a Host that doesn't answer PING, please use Host IP scan.

1) 10.3.20.0/24
2) Host IP
3) Quit
Enter your choice:
```

Selecting the Subnet in this case, NetScanner found 3 IPs with SSH enabled.

```
1) 10.3.20.0/24
2) Host IP
3) Quit
Enter your choice: 1

Starting Nmap 7.80 ( https://nmap.org ) at 2024-02-08 01:12 UTC
Nmap scan report for 10.3.20.99
Host is up (0.00039s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for 10.3.20.179
Host is up (0.00042s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for 10.3.20.197
Host is up (0.00045s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (3 hosts up) scanned in 7.04 seconds
```

8.1.4 SPEED TEST

This test is experimental because we use third-party tools (speedtest.net), but it works fine in most cases.

```
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

Retrieving speedtest.net configuration...
Testing from Amazon.com (44.222.9.224)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by GSL Networks (Ashburn, VA) [0.81 km]: 3.535 ms
Testing download speed.................................................
Download: 2526.46 Mbit/s
Testing upload speed...................................................
Upload: 272.16 Mbit/s
```

Note: *Using GRE tunnels you can reach up to 3 Gbps to Zscaler.*

8.2 CSC Admin Tasks

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

8.2.1 AWS SSM Agent (Register or De-Register) (TBC Check Run Commands list)

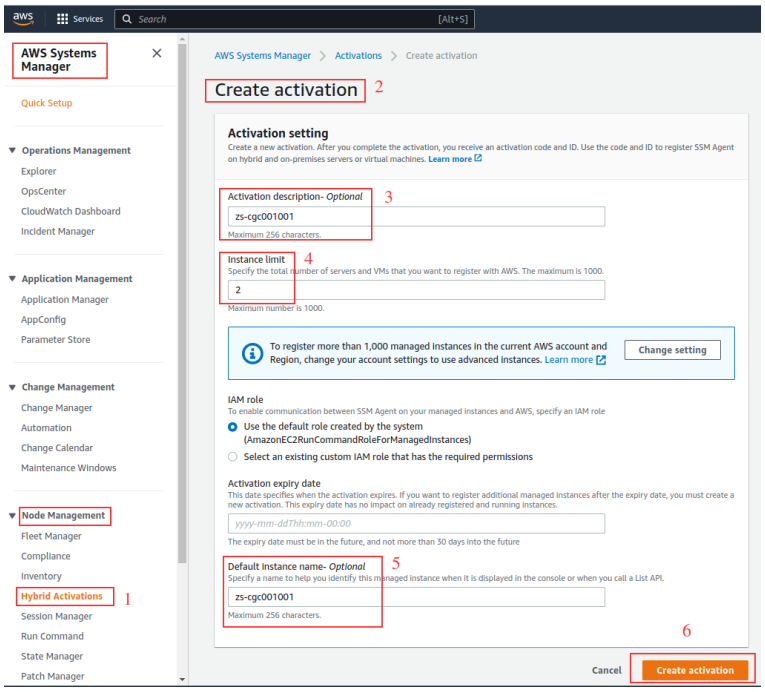
The CSC AWS has installed the AWS SSM Agent that allows you to check remotely the status of the CSC and “Run Commands” using AWS Systems Manager. You can manage all CSCs models[5] using AWS Systems Manager.

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

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

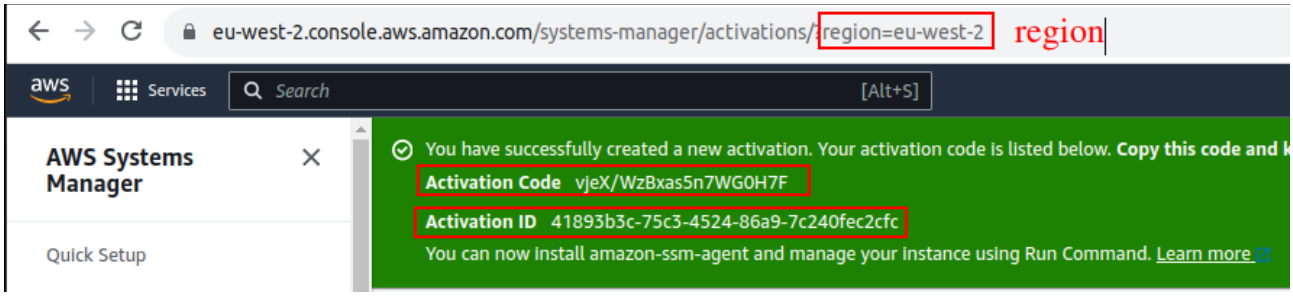
8.2.1.1 Create a "Hybrid Activation" from AWS console.

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



→ Click "Create Activation"

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

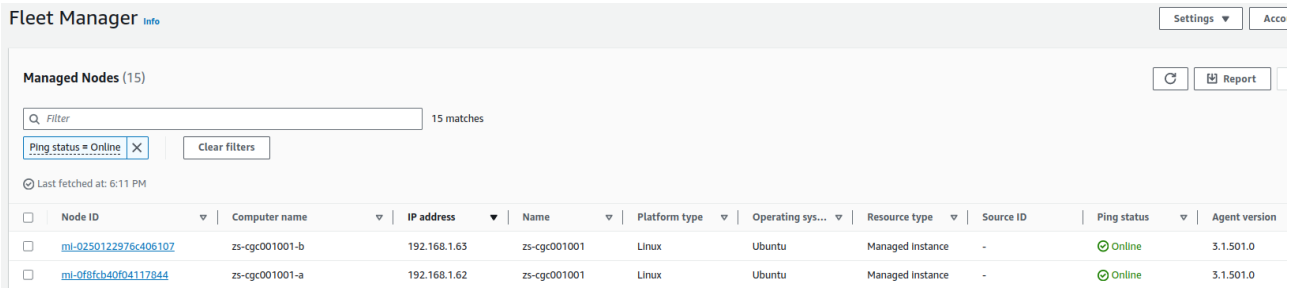


The values of Activation Code, Activation ID and Region are required to register the CSC. Keep this values on a safe place.

8.2.1.2 Register the CSCs

```
Do you want to Register (start) the AWS SSM Agent (y/n) y
Please, ingress Activation Code, Activation ID and Region (example: eu-west-1)
Activation Code :vjeX/WzBxas5n7WG0H7F
Activation ID :41893b3c-75c3-4524-86a9-7c240fec2cfc
Region :eu-west-2

(MHB-CSC)(INFO) AWS SSM Agent is  active (running) since Mon 2023-11-06 18:06:33 UTC; 64ms ago
(MHB-CSC)(INFO) AWS SSM Agent Registration values are: {"ManagedInstanceID":"mi-0f8fcb40f04117844","Region":"eu-west-2"}
```

8.2.1.3 View the Registered CSC on AWS Systems Manager



8.2.2 Manage Administrators, Restrict SSH access and Radius Configuration

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

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

8.2.2.1 Manage Administrators: cscadmin, csccli and ubuntu

The CSC GRE for AWS has 2 users enabled by default: "cscadmin" (for SSH Administrator Console Access) and "ubuntu" (Linux Terminal access). Additionally, you can enable the "csccli" user (Linux Terminal access).

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

```
Selection: 6

Please, select the task to do:

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

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

8.2.2.1.1 "cscadmin" settings

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

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

Please, select the task to do for user 'cscadmin':

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

8.2.2.1.2 "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.

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

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

User 'csccli' is enabled.

Please, select the action to take.

1) Disable csccli User
2) Change SSH Key
3) Quit
Enter your choice:
```

8.2.2.1.3 "ubuntu" settings

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

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

User 'ubuntu' is enabled.

Please, select the action to take.

1) Disable ubuntu User
2) Change SSH Key
3) Quit
Enter your choice:
```

8.2.2.2 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:
```

8.2.2.3 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.

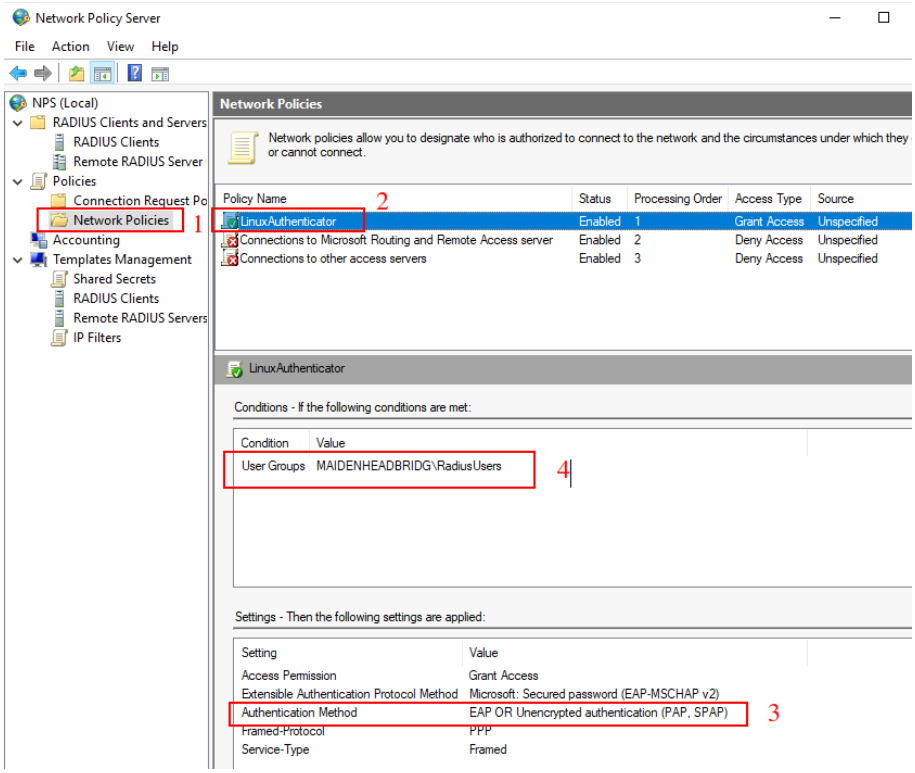
-> Configuration on the CSC: Add Radius Server and User:

```
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

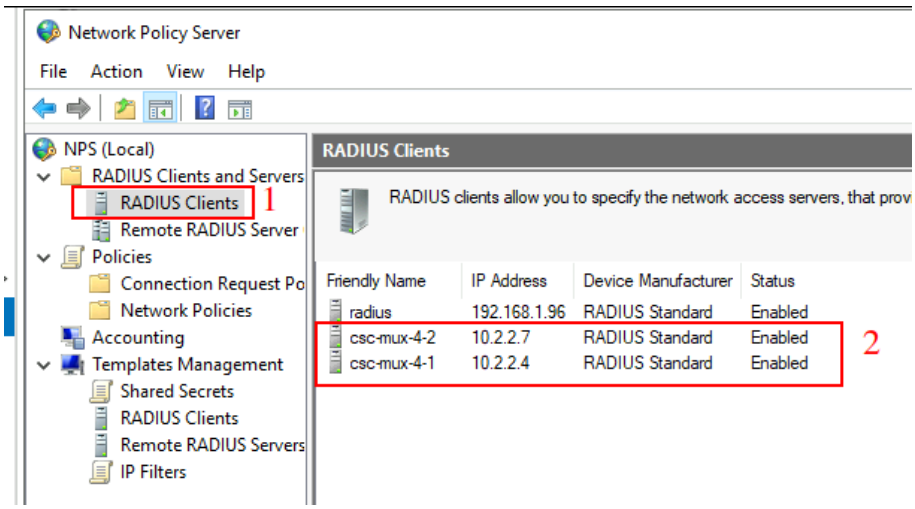
1 - Create Network Policy

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



8.2.3 Configure DNS, SNMP, NTP and Timezone.

8.2.3.1 DNS

```
Selection: 7

Please, select what you want to configure:

1) DNS servers
2) SNMP
3) NTP servers
4) Time Zone
5) Quit
Enter your choice: 1

Your current DNS Servers are: 8.8.8.8 ; 8.8.4.4

Note: Default DNS Servers are Google (8.8.8.8, 8.8.4.4)

Do you want to change the DNS servers?

1) Yes
2) No
Enter your choice: 1

Primary DNS Server (IP): 172.19.0.100
Secondary DNS Server (IP): 172.19.0.134

(MHB-CSC)(INFO) CSC: zs-cgc001001-a: DNS Servers changed via console. Using 172.19.0.100 and 172.19.0.134
```

8.2.3.2 SNMP

The CSC uses Ubuntu Server as its OS and offers all SNMP values of a standard Ubuntu Server. The CSC 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.

8.2.3.2.1 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
```

8.2.3.2.2 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.
3) Disable SNMP v3.
Enter your choice: 2
```

Skip SNMP v3

8.2.3.2.3 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
```

8.2.3.2.4 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
```

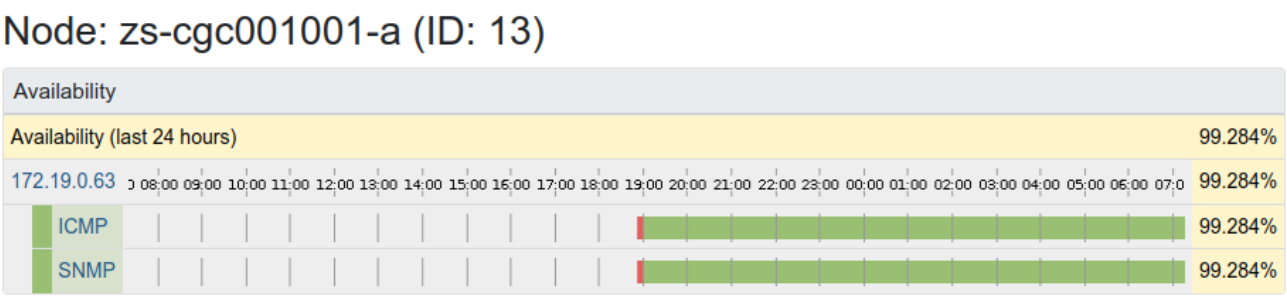
8.2.3.2.5 What can you do with SNMP?

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

8.2.3.2.5.1 Node Information

SNMP Attributes	
Name	zs-cgc001001-a
sysObjectID	.1.3.6.1.4.1.8072.3.2.10
Location	MHB-DC - KVM07
Contact	support@maidenheadbridge.com
Description	Test for Documentation

8.2.3.2.5.2 Node Availability



8.2.3.2.5.3 Node Interfaces (IP & SNMP)

Node Interfaces

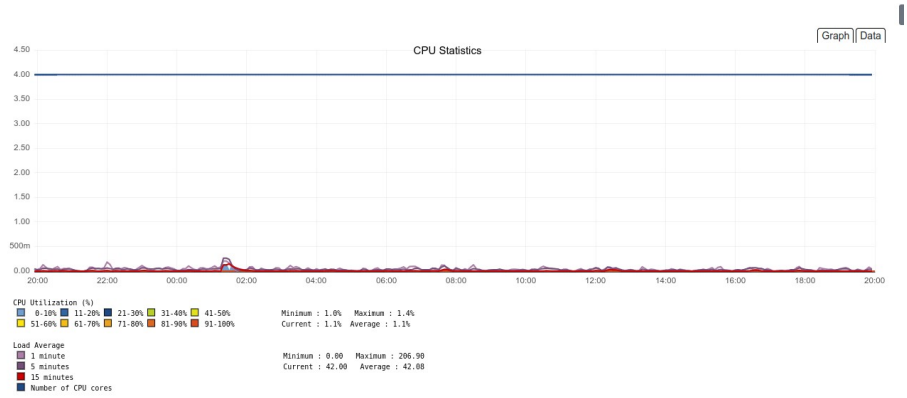
IP Interfaces | SNMP Interfaces

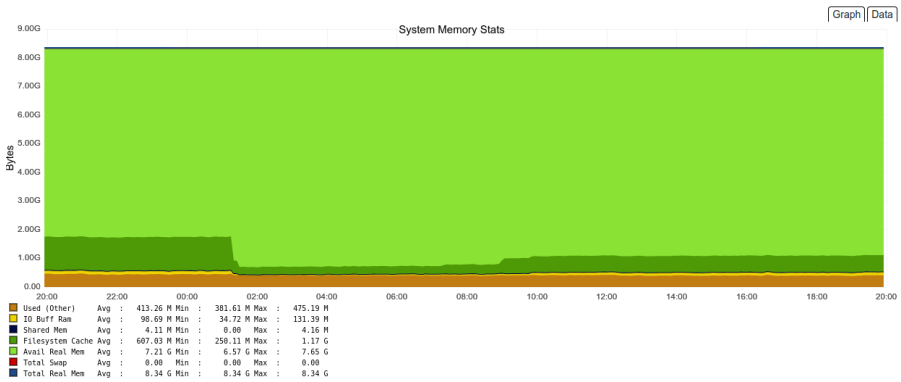
Search/Filter IP Interfaces

IP Address	IP Host Name	SNMP ifIndex	Managed
172.17.4.217	172.17.4.217	8	M
172.17.4.221	172.17.4.221	9	M
172.19.0.60	172.19.0.60	3	M
172.19.0.61	172.19.0.61	3	M
172.19.0.62	172.19.0.62	3	M
172.19.0.63	172.19.0.63	3	M
192.168.1.60	192.168.1.60	2	M
192.168.1.61	192.168.1.61	2	M
192.168.1.62	192.168.1.62	2	M
192.168.7.4	192.168.7.4	10	M

First Previous 1 2 Next Last

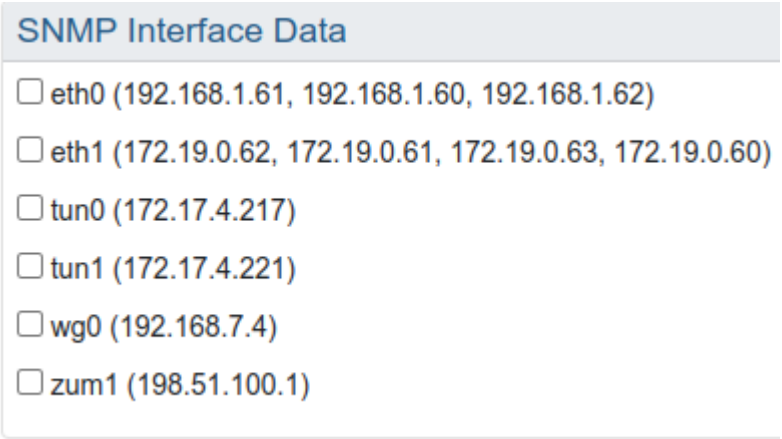
8.2.3.2.5.4 Node Statistics (CPU, Memory, etc)



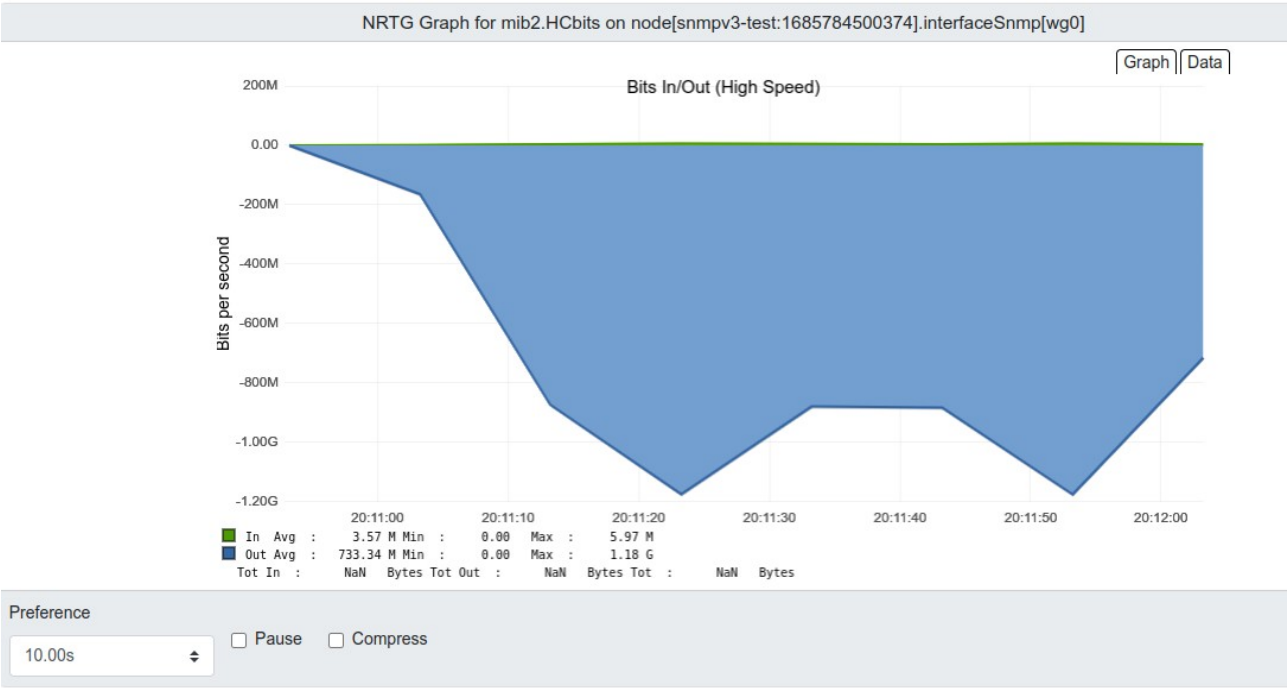


8.2.3.2.5.5 Interfaces Traffic

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



Example of real time traffic on PriCPA interface:



8.2.3.3 NTP

By default, the CSC PriCPA uses "ntp.ubuntu.com". You can configure here your NTP Servers.

```
Please, select what you want to configure:

1) DNS servers
2) SNMP
3) NTP servers
4) Time Zone
5) Quit
Enter your choice: 3

You are using default Ubuntu NTP servers.

Status: "Initial synchronization to time server 185.125.190.58:123 (ntp.ubuntu.com)."

Do you want to change the NTP servers?

1) Yes
2) No
Enter your choice: 1

Primary NTP Server (IP): 172.19.0.199
Secondary NTP Server (IP): 192.168.1.199

(MHB-CSC)(INFO) CSC: pricpa-csc-aZ-doc-1: NTP Servers changed via console. Using 172.19.0.199 and 192.168.1.199
```

Check the Status:

```
Selection: 10

Please, select what you want to configure:

1) DNS servers
2) SNMP
3) NTP servers
4) Time Zone
5) Quit
Enter your choice: 3

Your current NTP Servers are: 172.19.0.199 ; 192.168.1.199

Status: "Initial synchronization to time server 172.19.0.199:123 (172.19.0.199)."
```

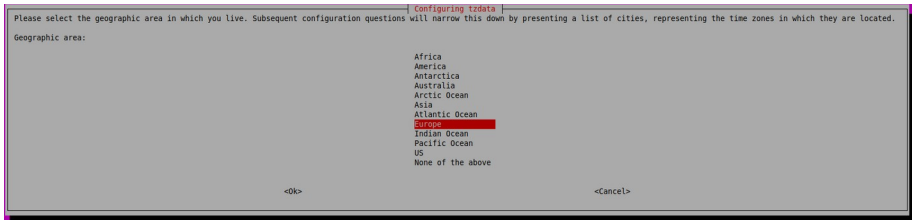
The NTP Server connects correctly when the Status is: "Initial synchronization to time server xxxx".

8.2.3.4 Time Zone

Use this menu to select the timezone of the CSC.

```
Selection: 10

Please, select what you want to configure:

1) DNS servers
2) SNMP
3) NTP servers
4) Time Zone
5) Quit
Enter your choice: 4

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:
```

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.


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
US
None of the above
<Ok>
<Cancel>

8.3 Proxy Bypass

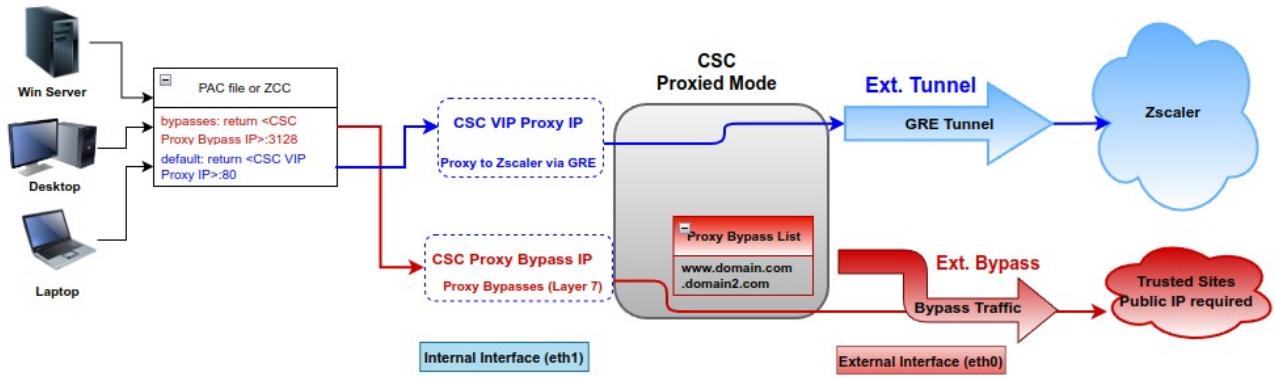
```
Proxy Bypass
8) View Current Proxy Bypass List
9) Configure Proxy Bypass
```

There are two main modes of Proxy Bypass: Standard and Advanced. The default behaviour of Standard mode is to block all domains that are not on the bypass list. In contrast, the default behaviour of Advanced mode is to send all domains to Zscaler (upstream proxy) that are not on the bypass list.

See Chapter "Traffic forwarding to Zscaler ZIA and Bypasses." for a detailed explanation of different use cases.

8.3.1 Standard Mode

8.3.1.1 Network Diagram



8.3.1.2 Configuration using PAC file

- Select "Configure Standard Mode"

- Select method: PAC URL

```
Enter your choice: 1

Please, select method to configure Proxy Bypass List

1) 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 Proxy Bypass PAC URL is not configured.

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

Please, input Proxy Bypass PAC URL
Bypass PAC URL https://pac.zscalerthree.net/maidenheadbridge.com/zs-cgc001001.pac

Your current Proxy Bypass PAC URL is https://pac.zscalerthree.net/maidenheadbridge.com/zs-cgc001001.pac

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

This is your current Proxy Bypass List

login.microsoftonline.com
login.microsoft.com
login.windows.net
ipinfo.io


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

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

8.3.1.3 Manual 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:

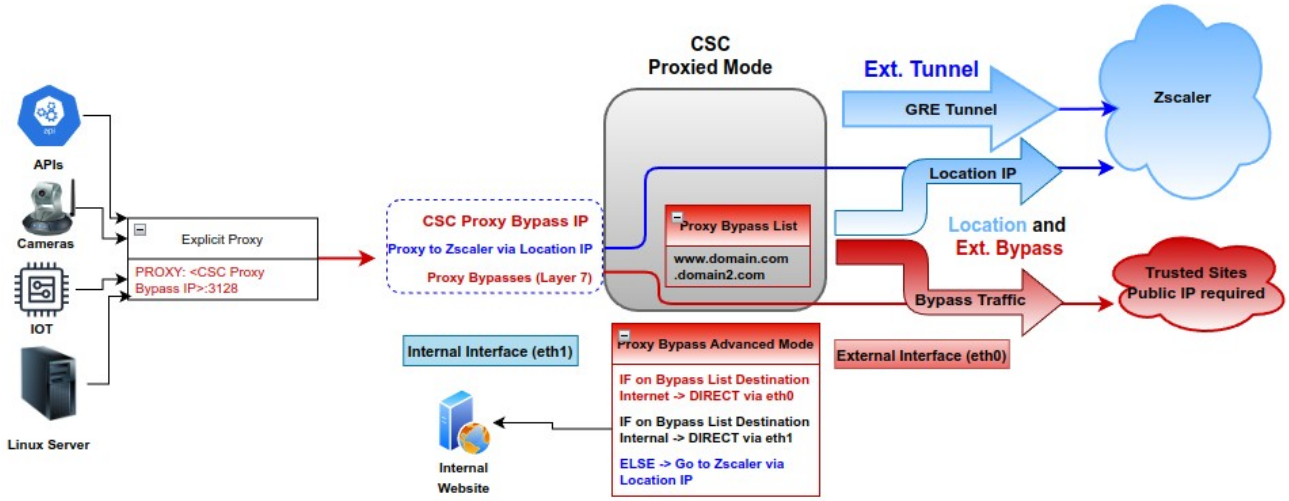
```
Do you want apply changes?
1) Yes
2) No
Enter your choice: 1

Proxy Bypass List updated sucessfully.
```

8.3.1.4 "View Current Proxy Bypass List"

```
Selection: 8

Proxy Bypass Mode: standard

This is the list of current Domains configured:

login.microsoftonline.com
login.microsoft.com
login.windows.net
ipinfo.io
```

8.3.2 Advanced Mode

8.3.2.1 Network Diagram



8.3.2.2 Create a "Location IP" on the Zscaler console

Run "Show Configuration and Status" and get your "Bypass Proxy Public IP"

```
ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 44.222.9.224
Bypass Proxy Public IP: 18.214.102.160
```

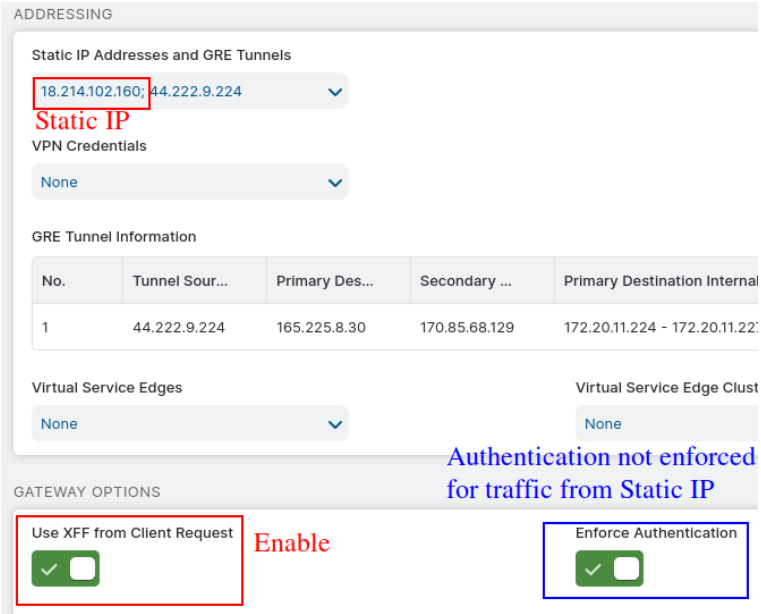
On your Zscaler console, go to Administration -> Static IPs & GRE Tunnels and add the "Bypass Proxy Public IP"

Associate the Static IP created to a Location. You can create a new one or add this Static IP to the Location created for the GRE tunnel.

IMPORTANT:

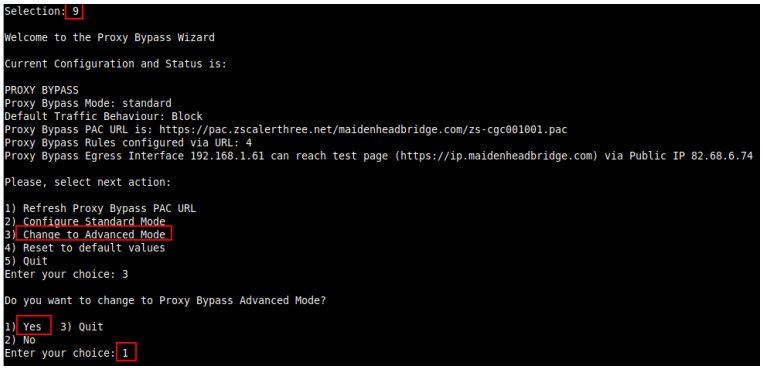
1 - Enable "Use XFF from Client Request. " This will allow you to see your devices' internal IPs and apply rules per source IP.

2 - The CSC will use the Zcaler nodes as "upstream proxies" and send the traffic using port TCP 9840. Zscaler Cloud exempts all traffic arriving from a known location on TCP/9840 from authentication, even if your Location settings enable Enforce Authentication.

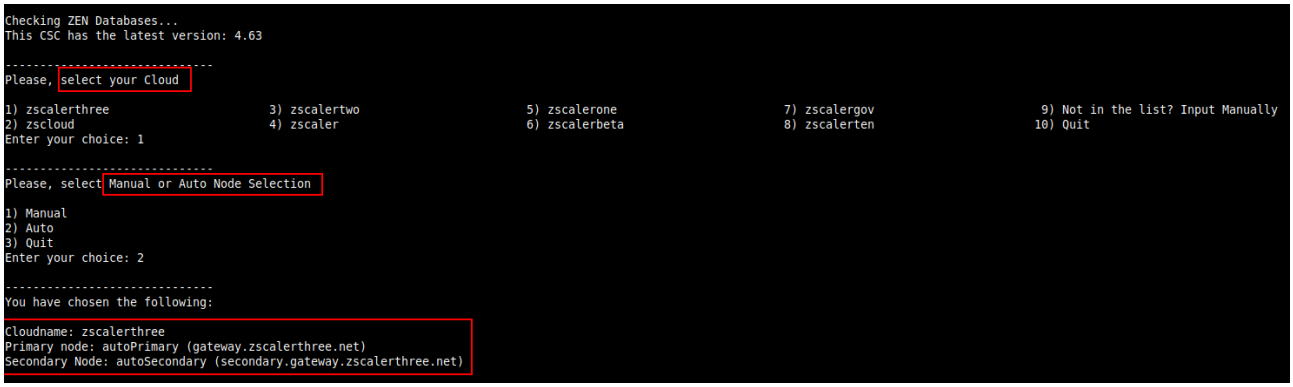


8.3.2.3 Configuration using JSON URL

- Change from Standard to Advanced Mode



- Select your Zscaler Cloud and Nodes. (Primary and Secondary.)



➢ Configure Proxy Bypass JSON file URL

Proxy Bypass JSON file URL example

```
{
  "model": "csc-gre-zs-vm",
  "type": "proxyBypassAdvanced",
  "version": "1.0",
  "help": ".domain.com matches domain.com and any subdomain of <>.domain.com. Do not use asterisk '*'",
  "proxyBypassRules": {
    "internalSites": [
      ".domainInternal.com",
      "fqdn-internal.com"
    ],
    "externalSites": [
      ".externalDomain.com",
      "fqdn-external.com",
      "ip.maidenheadbridge.com",
      "ipinfo.io"
    ]
  }
}
```

```
Internal and External Bypass Configuration

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


*** Proxy Bypass JSON URL is not configured ***

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

Please, input Proxy Bypass JSON URL
Proxy Bypass JSON URL: https://mhb-csc-pac.s3.amazonaws.com/proxyBypassRulesFile.json

Do you want to refresh the Proxy Bypass List (via JSON file URL)?
1) Yes
2) No
Enter your choice: 1

Proxy Bypass JSON file imported successfully
```

➢ Review and Apply values

```
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:

Internal Sites
Index: 0, ".domainInternal.com"
Index: 1, "fqdn-internal.com"

External Sites
Index: 0, ".externalDomain.com"
Index: 1, "fqdn-external.com"
Index: 2, "ip.maidenheadbridge.com"
Index: 3, "ipinfo.io"

Validating Configuration

Your Cloud is: zscalerthree

Checking Node autoPrimary Proxy hostname gateway.zscalerthree.net
Proxy hostname gateway.zscalerthree.net has IP 165.225.16.37
Node autoPrimary is Alive

Checking Node autoSecondary Proxy hostname secondary.gateway.zscalerthree.net
Proxy hostname secondary.gateway.zscalerthree.net has IP 147.161.141.129
Node autoSecondary is Alive

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

(MHB-CSC)(INFO) Proxy Bypass Advanced Mode is enabled using nodes: autoPrimary (165.225.16.37) and autoSecondary (147.161.141.129).

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

8.3.2.4 Configuration pasting JSON file

- Go to 9) Configure Proxy Bypass -> 3) Configure Advanced Mode -> Select No to change the Zscaler nodes -> Internal and External Bypass Configuration -> Manual

```
Internal and External Bypass Configuration

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


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

Please, paste Proxy 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.

Proxy Bypass JSON file: {
  "model": "csc-gre-zs-vm",
  "type": "proxyBypassAdvanced",
  "version": "1.0",
  "help": ".domain.com matches domain.com and any subdomain of <>.domain.com. Do not use asterisk '*'",
  "proxyBypassRules": {
    "internalSites": [
      ".domainInternal.com",
      "fqdn-internal.com"
    ],
    "externalSites": [
      ".externalDomain.com",
      "fqdn-external.com",
      "ip.maidenheadbridge.com",
      "ipinfo.io"
    ]
  }
}

Proxy Bypass JSON file imported successfully
```

- Review and Apply the configuration.

```
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:

Internal Sites
Index: 0, ".domainInternal.com"
Index: 1, "fqdn-internal.com"

External Sites
Index: 0, ".externalDomain.com"
Index: 1, "fqdn-external.com"
Index: 2, "ip.maidenheadbridge.com"
Index: 3, "ipinfo.io"

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

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

8.3.2.5 "View Current Proxy Bypass List"

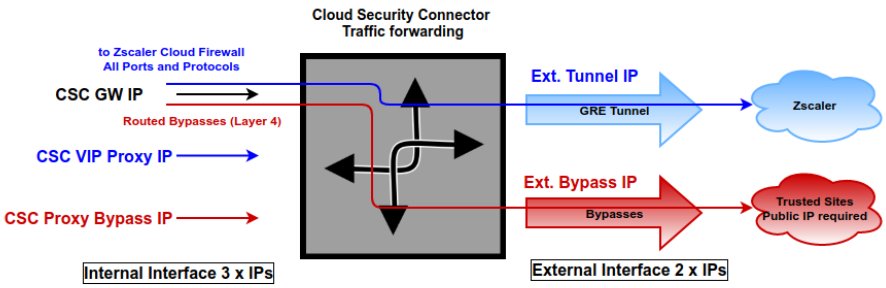
```
Selection: 8

Proxy Bypass Mode: advanced

This is the list of current Domains configured:

External domains

.externalDomain.com
fqdn-external.com
ip.maidenheadbridge.com
ipinfo.io


Internal domains

.domainInternal.com
fqdn-internal.com
```

8.4 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
```

8.4.1 Routed Bypass - Traffic Flow



8.4.2 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:
```

8.4.2.1 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"
```

8.4.2.2 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"
    },
```

8.4.3 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:
```

8.4.3.1 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.
```

8.4.3.2 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
```

8.5 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
```

8.5.1 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.
```

8.5.2 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.
```

8.6 Configuration Wizards

In this section, you can run the Configuration Wizard to change Zscaler Nodes and GRE values. It also provides a simple way to Switch tunnels.

```
Configuration Wizards
14) Configure Zscaler Nodes and GRE values.
15) Switch Zscaler Tunnels - Primary / Secondary.
16) Reserved for future use.
```

8.6.1 Configure Zscaler Nodes and GRE values.

This wizard allows you to change the current values configured. The initial screen shows the values required. Please see the section "Creating the CSC GRE Cluster" for detailed information about creating the values of "Static IP", "GRE tunnel", and "Location."

- Initial screen.

```
Selection: 14

Welcome to the CSC GRE Configuration Wizard

Before to start you need have the following values ready:

1) Cloudname: zscloud, zscalertwo, zscaler, etc. Check your Zscaler Admin URL https://admin.<cloud name>.net to find it.
2) GRE Tunnel IPs & Location: On your Zscaler console, please, follow this procedure:
  2.1) Go to Administration -> Static IPs & GRE Tunnels.
          2.1.a) Add 'Static IP': 82.68.6.74 .
          2.1.b) Add Add 'GRE Tunnel' using Static IP: 82.68.6.74, Select 'Data Center' (Primary and Secondary) and Select 'Internal GRE IP Range'.
  2.2) Go to Administration -> Location Management, and 'add Location' using 'Static IP': 82.68.6.74
  2.3) On Location -> GRE Tunnel Information: take note of the following values:
          2.3.a) Primary Destination
          2.3.b) Secondary Destination
          2.3.c) First IP of 'Primary Destination Internal Range'. For example, if Primary Destination Internal Range is: 172.18.7.120 - 172.18.7.123, you need only the first value for this wizard: 172.18.7.120

Current Values Configured:
------------------
Cloudname: zscalerthree
------------------
Tunnel Source IP:    82.68.6.74 (* this is your Tunnel Source Public IP)

Primary Destination:   165.225.16.36
Secondary Destination: 165.225.76.39
First IP of 'Primary Destination Internal Range': 172.17.4.216
returnToPrimaryTunnel: true
------------------
Are you ready to continue?
1) Yes
2) No
Enter your choice:
```

- Configuring values

```
------------------
Cloud Configuration

Your current Cloud is: zscalerthree

Do you want to change the Cloud Name?
1) Yes                                  Cloud Selection
2) No
Enter your choice: 1

Please select or input your Cloud Name
1) zscalerthree          3) zscalertwo          5) zscalerone          7) Not in the list? Ingress Manually
2) zscloud               4) zscaler             6) zscalerbeta         8) Quit
Enter your choice: 1

GRE tunnels Configuration

Your current GRE tunnels configuration is:

Tunnel Source IP:    82.68.6.74

Primary Destination:    165.225.16.36
Secondary Destination:  165.225.76.39
First IP of 'Primary Destination Internal Range': 172.17.4.216
returnToPrimaryTunnel: true

Do you want to change the GRE tunnels configuration?
1) Yes
2) No
Enter your choice: 1                    GRE values configuration

Please, Insert the GRE values:

Tunnel Source Public IP (IP): 82.68.6.74
Primary Destination (IP): 165.225.16.36
Secondary Destination (IP): 165.225.76.39
First IP of 'Primary Destination Internal Range': 172.17.4.216

'returnToPrimaryTunnel' variable:
Please select 'true' if you want the CSC to return to Primary tunnel (after 10 min of stability) when using Secondary tunnel.
Select 'false' if you want to remain using Secondary Tunnel and not to return to Primary.(Secondary will be nominated as 'new' Primary)

1) true                 "returnToPrimaryTunnel" variable configuration
2) false
Enter your choice:
```

- Confirm values (the CSC will reboot)

```
Please confirm these values:
-------------------
Cloudname:  zscalerthree
-------------------
GRE tunnels IP values:

Tunnel Source IP (IP):  82.68.6.74

Primary Destination:   165.225.16.36
Secondary Destination: 165.225.76.39
First IP of 'Primary Destination Internal Range': 172.17.4.216
returnToPrimaryTunnel: true
-------------------
Do you want to implement these values? (The CSC will reboot)
1) Yes
2) No
Enter your choice:
```

8.6.2 Switch Tunnels - Primary / Secondary.

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

```
Configuration Wizards
14) Configure Zscaler Nodes and GRE values.
15) Switch Zscaler Tunnels - Primary / Secondary.
16) Reserved for future use.
```

```
Selection: 15

-------------------
ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
GRE tunnels egress Public IP: 82.68.6.74        Current values
Primary Tunnel:
                ZEN Public IP: 165.225.16.36
                Tunnel IPs (local/zen): 172.17.4.217 / 172.17.4.218
Secondary Tunnel:
                ZEN Public IP: 165.225.76.39
                Tunnel IPs (local/zen): 172.17.4.221 / 172.17.4.222

TUNNEL STATUS
Primary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive
Secondary Tunnel (reachability):
                              Layer 7 Keepalive is: Alive
                              GRE ZEN Tunnel IP is: Alive

Tunnel Status: Primary tunnel is active since: Wed 8 Nov 11:33:09 UTC 2023

HTTP://IP.ZSCALER.COM PAGE STATUS
You are accessing the Internet via Zscaler Cloud: London III in the zscalerthree.net cloud.
Your Gateway IP Address is 82.68.6.74

-------------------
Do you want to Switch Primary / Secondary Tunnel?
Selecting Yes will disrupt all current connections.

1) Yes
2) No
Enter your choice 1

Tunnels switched via Console on: Wed  8 Nov 11:40:02 UTC 2023
```

8.6.3 High Availability configuration

In this section, you can configure the CSC on the HA pair to manage the default route to the Internet automatically.

```
Selection: 16

This Wizard is for High Availability scenarios when changing default route to Internet.

----------------------
How to configure:
1) Deploy a pair of CSCs with the following conditions:
        1.1) There is connectivity each other via their internal interfaces. (Mandatory)
        1.2) They are in different availability zones. (Recommended)
2) Create an IAM role with the following permissions and apply it to each CSC:
_____________________________________

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateAddress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:DescribeAddresses",
                "ec2:DescribeInstances",
                "sns:ListSubscriptionsByTopic",
                "ec2:CreateTags",
                "ec2:DescribeSecurityGroups",
                "ec2:ReplaceRoute",
                "ec2:RevokeSecurityGroupIngress",
                "sns:Publish",
                "ec2:DescribeSecurityGroupRules",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:AssociateAddress",
                "ec2:DescribeRouteTables"
            ],
            "Resource": "*"
        }
    ]
}
_____________________________________

3) Get the 'Route Table ID' of the Route Table/s where there is Default Route (0.0.0.0/0) to Internet
4) Get the 'Instance ID' of the other CSC on the pair
5) Create a SNS notification and get the 'ARN'
6) Run the Wizard on the FIRST CSC and input the following values manually: (all values are mandatory)
  6.1) Route Table ID/s (where there is Default Route to internet).
  6.2) Instance ID of other CSC on the pair.
  6.3) ARN of the SNS message for Notications of Route changes.
7) Run the Wizard on the SECOND CSC pasting the JSON file obtained from the FIRST CSC

How it works:
The CSCs on the HA pair will automatically select the Gateway (Target) for the Default Route on the Route Table/s.
When a change occurs, you will receive a SNS message notifying the new Gateway (Target).
On the routing table you can check Destination: 0.0.0.0/0 Target: eni-xxxyyyzzz
The 'Private Access Public IP' will be moved to the CSC with the default route to the Internet.
----------------------

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) Deploy a pair of CSCs with the following conditions:

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

   1.2) They are in different availability zones. (Recommended)

2) Create an IAM role with the following permissions and apply it to each CSC:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:DisassociateAddress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:DescribeAddresses",
        "ec2:DescribeInstances",
        "sns:ListSubscriptionsByTopic",
        "ec2:CreateTags",
        "ec2:DescribeSecurityGroups",
        "ec2:ReplaceRoute",
        "ec2:RevokeSecurityGroupIngress",
        "sns:Publish",
        "ec2:DescribeSecurityGroupRules",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:AssociateAddress",
        "ec2:DescribeRouteTables"
      ],
      "Resource": "*"
    }
  ]
}
```

3) Get the 'Route Table ID' of the Route Table/s where there is Default Route (0.0.0.0/0) to Internet

4) Get the 'Instance ID' of the other CSC on the pair

5) Create a SNS notification and get the 'ARN'.

6) Run the Wizard on the FIRST CSC and input the following values manually: (all values are mandatory)

   6.1) Route Table ID/s (where there is Default Route to internet).

   6.2) Instance ID of other CSC on the pair.

   6.3) ARN of the SNS message for Notications of Route changes.

7) Run the Wizard on the SECOND CSC pasting the JSON file obtained from the FIRST CSC

How it works:

The CSCs on the HA pair will automatically select the Gateway (Target) for the Default Route on the Route Table/s.

When a change occurs, you will receive a SNS message notifying the new Gateway (Target).

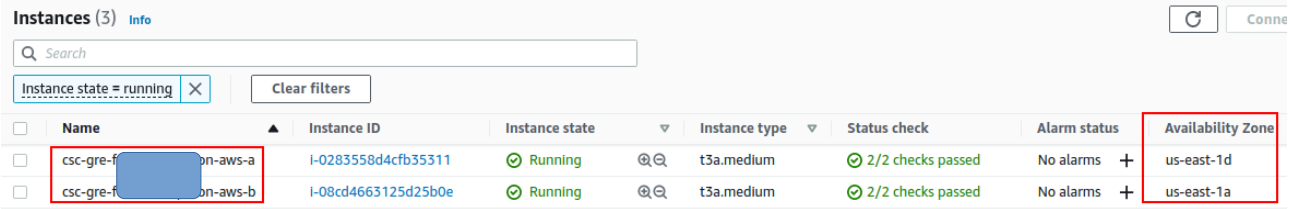
On the routing table you can check Destination: 0.0.0.0/0 Target: eni-xxxyyyzzz

The 'Private Access Public IP' will be moved to the CSC with the default route to the Internet.

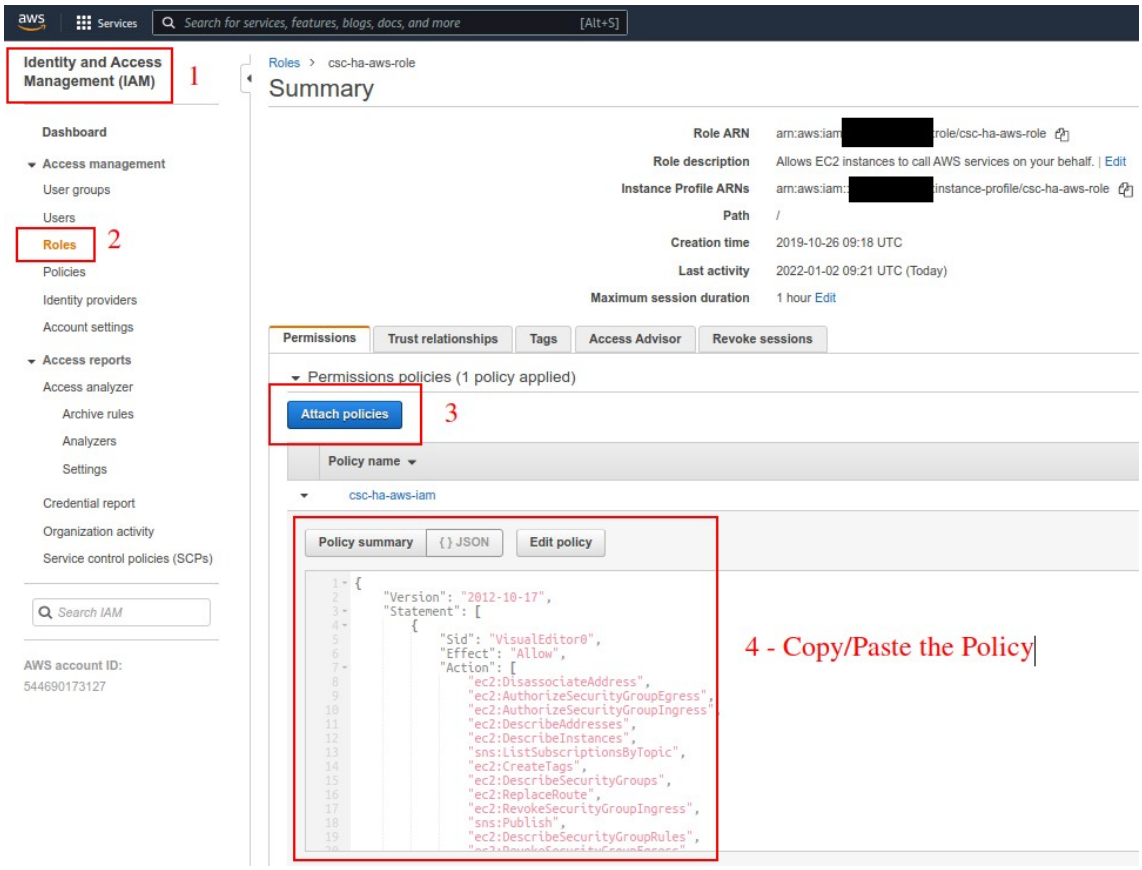
8.6.3.1 High Availability configuration on detail

This section shows in detail how to deploy a pair of CSC on High Availability.

8.6.3.1.1 Deploy a pair of CSC on the different availability zones.

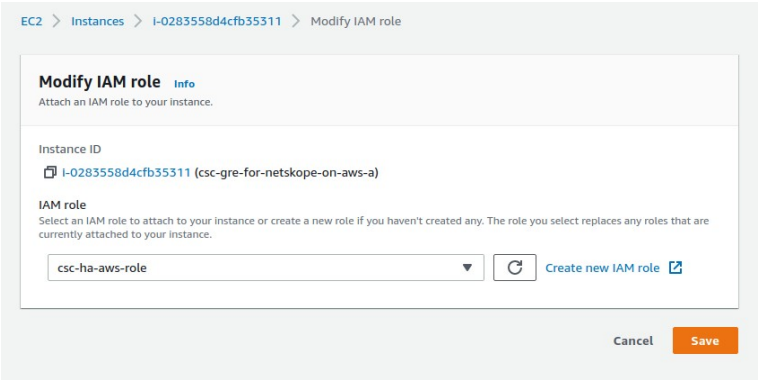


8.6.3.1.2 Create an IAM role with the following policies



Next, apply the Role created to each CSC on the pair.

Right click the instance → Security → Modify IAM role

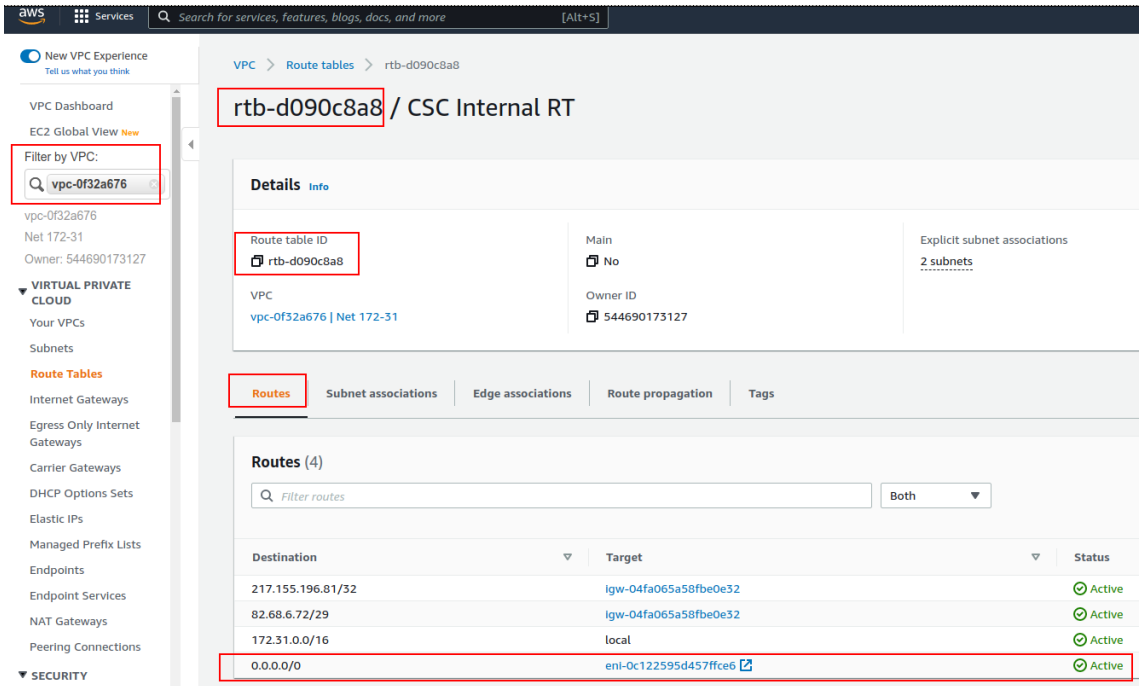


Select the Role and Save. Do the same for the other CSC.

8.6.3.1.3 Get the 'Route Table ID' of the Route Table/s where there is Default Route (0.0.0.0/0) to Internet

Go to VPC → Route Tables and get the 'Route Table ID' of the Route Table/s where there is Default Route (0.0.0.0/0) to Internet.

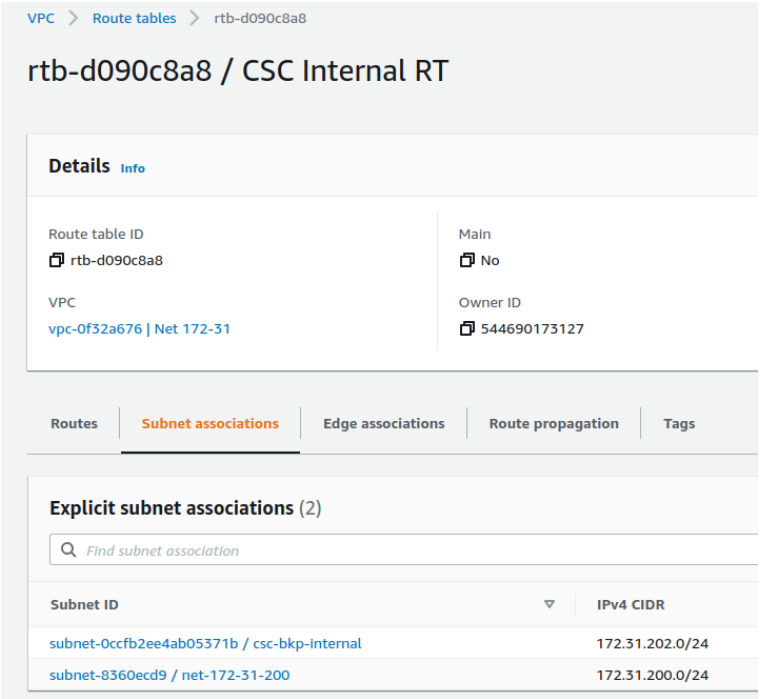
Create or modify the route 0.0.0.0/0 via Target → eni-xxyy (select the eni number from the internal interface of one CSC on the pair)



Note 1: The CSC pair will modify the "Target" of Route 0.0.0.0/0. Other Destinations will remain untouched.

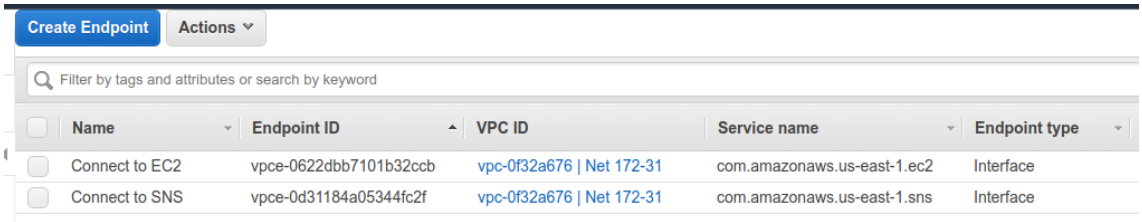
Note 2: Be sure to add other destinations, like your internal subnets or your public IPs, via the proper "Target" to avoid losing connectivity to the VPC

Next, apply the Subnet Associations to the Routing Table:



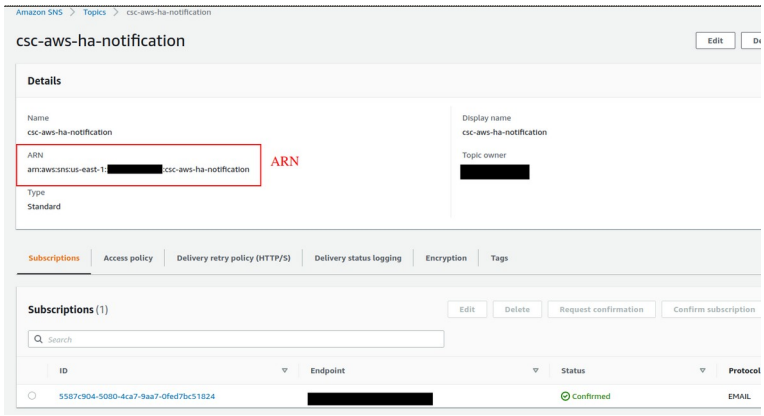
8.6.3.1.4 Create "Endpoints" to AWS services (EC2, SNS, S3, etc.)

When changing the default route to the internet via Zscaler, your subnets will potentially lose contact with some AWS services: EC2, SNS, S3, etc. The CSC on the HA requires creating two endpoints: EC2 and SNS.



8.6.3.1.5 Create SNS message for Alerts.

Go to Amazon SNS → Topics and create a Topic. Obtain the ARN



8.6.3.1.6 Run the HA Wizard on the First CSC

Input the values manually on the First CSC.

```
The HA service in NOT Active

Do you want to configure it?

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

IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role

---------------------
Please, input the following values:

Route Table ID= rtb-d090c8a8   3

Do you want to add another Route Table ID?

1) Yes
2) No
Enter your choice: 2

Instance ID of other CSC in the pair= i-08cd4663125d25b0e

SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification   4

---------------------
Values to configure are:

Routing Tables=1
Routing Table ID= rtb-d090c8a8
Instance ID of other CSC in the pair= i-08cd4663125d25b0e
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification

Do you want to apply changes?

1) Yes
2) No
Enter your choice: 1   5
```

```
Do you want to apply changes?

1) Yes
2) No
Enter your choice: 1

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

High Availability JSON file:
_____________________________________________________________

{
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role",
    "haSnsMessageArn": "arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification",
    "haInstanceIdFirstCsc": "i-0f15f29fcc4d69eb6",
    "haInstanceIdSecondCsc": "i-08cd4663125d25b0e",
    "haBypassPublicIpFirstCsc": "23.23.210.207",
    "haBypassPublicIpSecondCsc": "174.129.249.225",
    "haPrivateAccessPublicIp": "23.23.210.207",
    "haVPC": "vpc-0f32a676",
    "haRouteTables": [
      {
        "routeTableId": "rtb-d090c8a8"
      }
    ]
  }
}
_____________________________________________________________

CSC HA is : active (running) since Mon 2022-01-10 20:38:02 UTC; 20ms ago

Press Enter to continue...
```

Please, copy the JSON file. You will need to paste it on the second CSC on the HA Pair.

8.6.3.1.7 Configure the second CSC on the HA pair.

Run the HA Wizard on the second CSC.

```
Do you want to configure it?

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

Private Access Local Config JSON file: {
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role",
    "haSnsMessageArn": "arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification",
    "haInstanceIdFirstCsc": "i-0f15f29fcc4d69eb6",
    "haInstanceIdSecondCsc": "i-08cd4663125d25b0e",
    "haBypassPublicIpFirstCsc": "23.23.210.207",
    "haBypassPublicIpSecondCsc": "174.129.249.225",
    "haPrivateAccessPublicIp": "23.23.210.207",
    "haVPC": "vpc-0f32a676",
    "haRouteTables": [
      {
        "routeTableId": "rtb-d090c8a8"
      }
    ]
  }
}                                                    3 - Paste JSON
(MHB-CSC)(INFO) High Availability: IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
(MHB-CSC)(INFO) High Availability JSON file (highAvailability.json) integrity is OK
(MHB-CSC)(INFO) High Availability: IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
(MHB-CSC)(INFO) High Availability: Using VPC vpc-0f32a676. All Route Table IDs must belong to VPC vpc-0f32a676.
(MHB-CSC)(INFO) High Availability: Route Table ID rtb-d090c8a8 configured.
(MHB-CSC)(INFO) High Availability is  active (running) since Mon 2022-01-10 20:46:12 UTC; 18ms ago.

Press Enter to continue...
```

Done!

8.6.3.1.8 Checking HA Status

Run "Show Configuration and Status" and check High Availability Section.

```
HIGH AVAILABILITY Information
The HA service is:  active (running) since Mon 2022-01-10 20:46:12 UTC; 3min 53s ago
IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
The Default Route to Internet is using this Gateway (Target): eni-0a75cf038298d8e31 (this CSC)
Current values configured are:
Route Table/s Configured (Qty)= 1  (using VPC: vpc-0f32a676)
Route Table ID= rtb-d090c8a8
Instance ID of other CSC in the pair= i-0f15f29fcc4d69eb6
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification
Private Access Public IP= 23.23.210.207

Press Enter to continue...
```

8.6.3.1.9 Notifications from CSC on HA

Each CSC on the pair will send notifications when:

→ There is no connectivity at all with Zscaler. No CSC is able to reach Zscaler.

→ At power up the CSC will notify the current “eni-xxyy” used as default GW to internet

→ On routing change, the CSCs will notify the changes.

Example of notifications:

AWS Notification Message External Inbox ×

csc-aws-ha-notification <no-reply@sns.amazonaws.com> Sun, 2 Jan, 20:0

to me

INFO (from i-0283558d4cfb35311,us-east-1): Default Route to Netskope using CSC Interface: eni-0c122595d457ffce6 of Instance i-0283558d4cfb35311

--

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification:5587c904-5080-4ca7-9aa7-0fed7bc51824&Endpoint=alarsen@maidenheadbridge.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Logs generated:

```
Jan  2 20:05:19 ip-172-31-201-168 root: (MHB-CSC)(INFO) Default Route to Netskope using CSC Interface: eni-0c122595d457ffce6 of Instance i-0283558d4cfb35311
Jan  2 20:05:24 ip-172-31-96-172 root: (MHB-CSC)(INFO) Default Route to Netskope using CSC Interface: eni-0c122595d457ffce6 of Instance i-0283558d4cfb35311
```

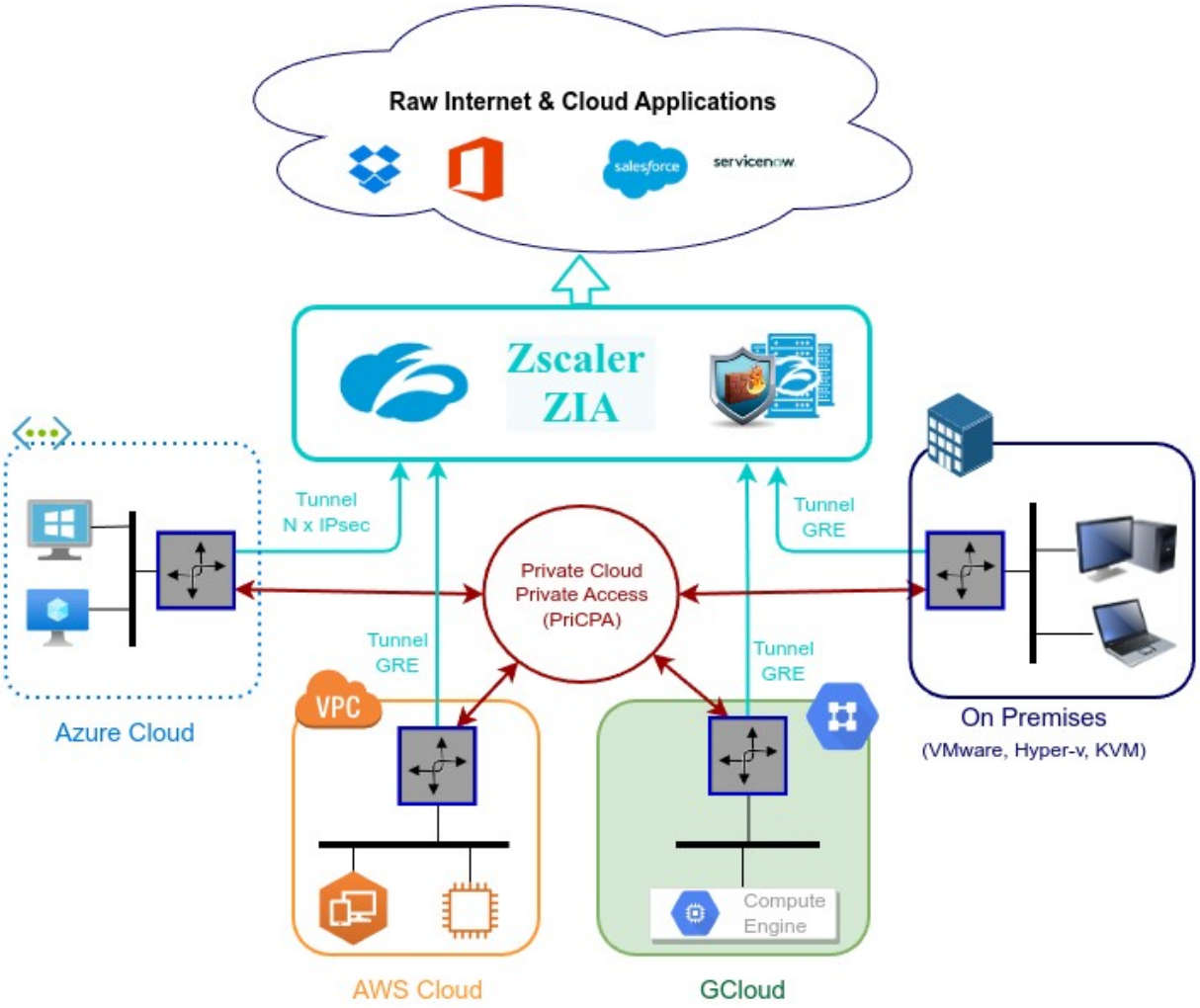
9 Private Cloud Private Access

9.1 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 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 whom inside your Private Cloud.

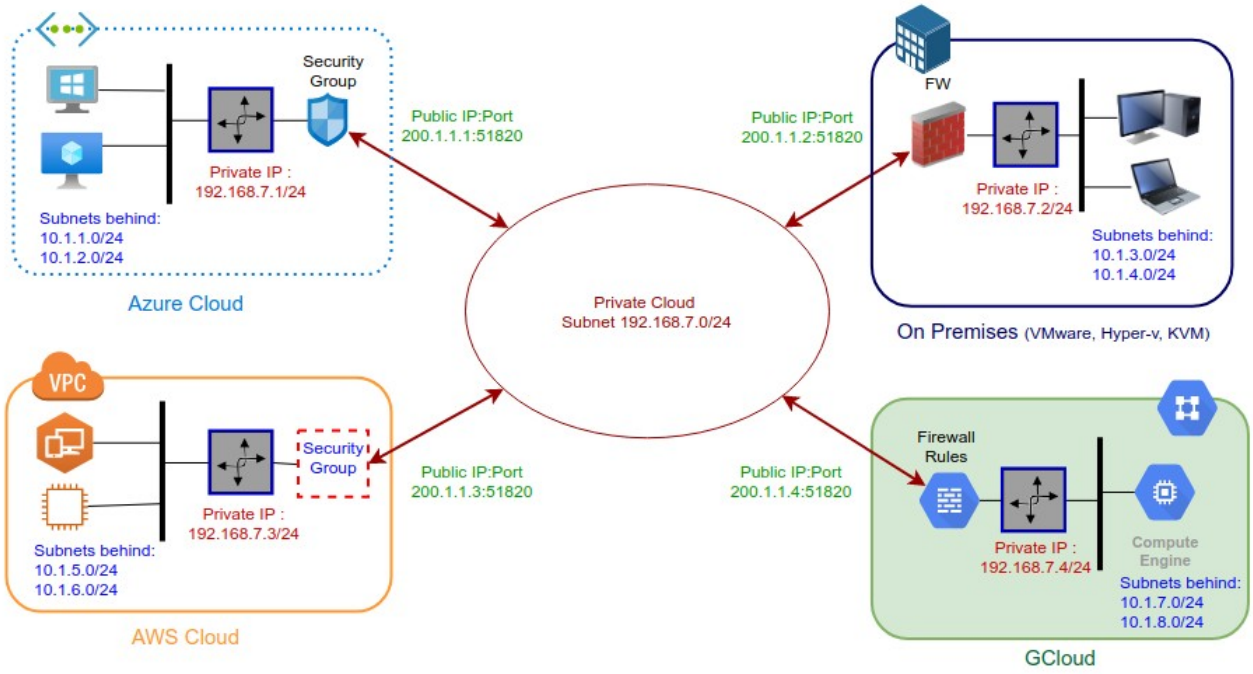
9.2 PriCPA Network Diagrams

9.2.1 High Level Network Diagram



9.2.2 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.

9.3 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.
```

In a few simple steps, you can configure PriCPA:

1. 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".
2. Initialize the second Node of the HA pair using the "Token" and "Private Access Local JSON file".
3. 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 D for more detail about how to install these programs and the plugins required.*

9.3.1 Create the Local configuration (First node of the HA pair)

```
Selection: 18

Private Access Configuration Wizard

Steps to configure Private Access:

  - Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
  - Copy Local Configuration to the other CSC in the HA pair.
  - 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:
```

- From Main Menu, select "18) Configure Private Access."
- Select "1) Create (or change) Private Access Local Configuration"

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

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: 1
```

- 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": "4QJ7DPswdTx+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.*

9.3.2 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

```
Private Access Configuration Wizard

Steps to configure Private Access:

  - Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
  - Copy Local Configuration to the other CSC in the HA pair.
  - 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

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.
```

9.3.3 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.

9.3.3.1 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.

9.3.3.2 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 [6]: 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.*

6 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.

9.3.3.3 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": "" }
        }
      ]
```

9.3.3.4 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": "" }
        }
      ]
```

9.3.4 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.

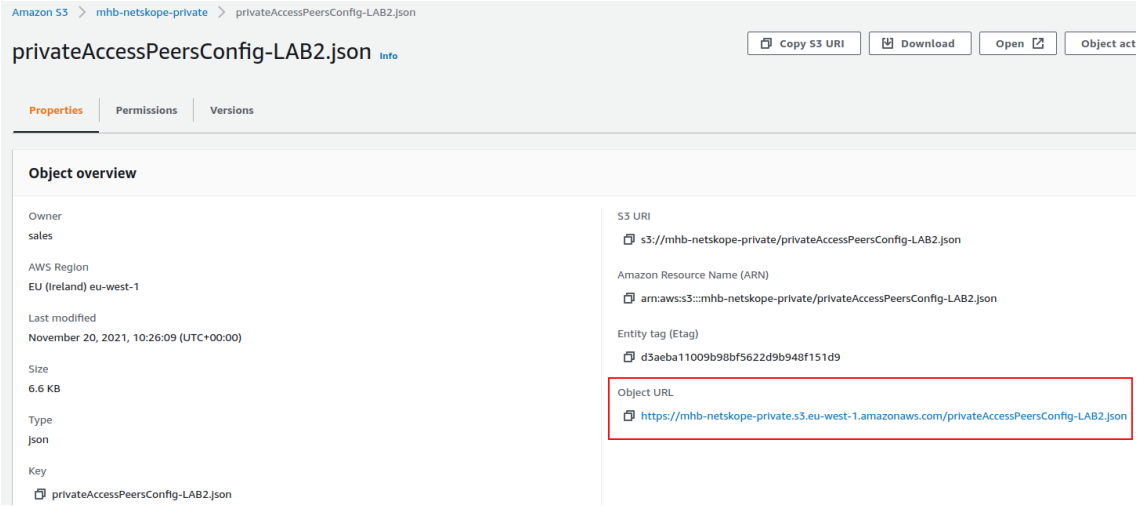
In this section we are going to explain two methods: URL and Manual Copy. The DevOps method is explained on Section11: DevOps operations.

9.3.4.1 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[7] 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

7 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
```

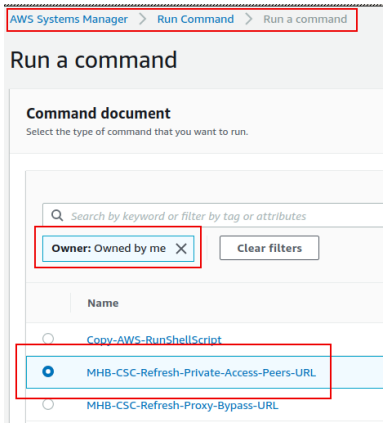
At this moment, you have the option to review the privateApps to configure in Compact or JSON format and to apply the values.

```
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.
```

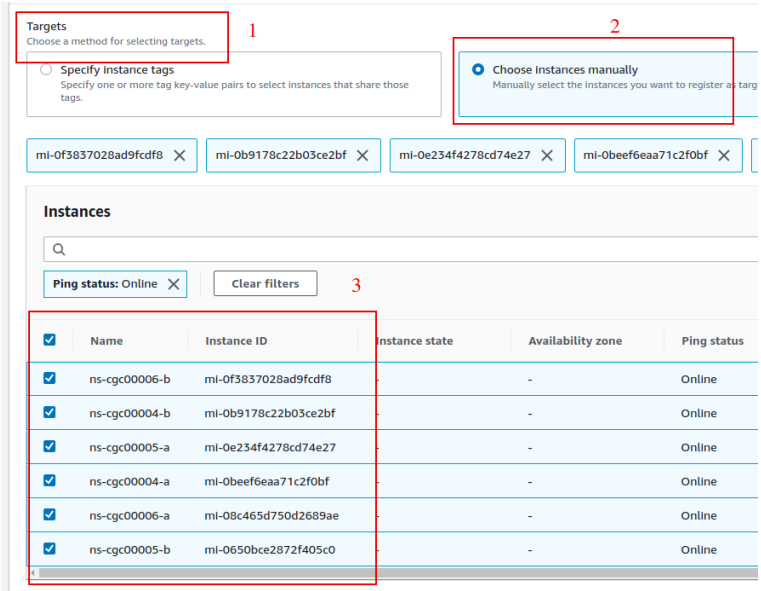
3. The next time you want to refresh the Private Access Peers JSON file, update the file, deploy it on the same location URL and Run Command: "Refresh Private Access Peers URL" using AWS SSM Agent or Rundeck.

AWS System Manager:

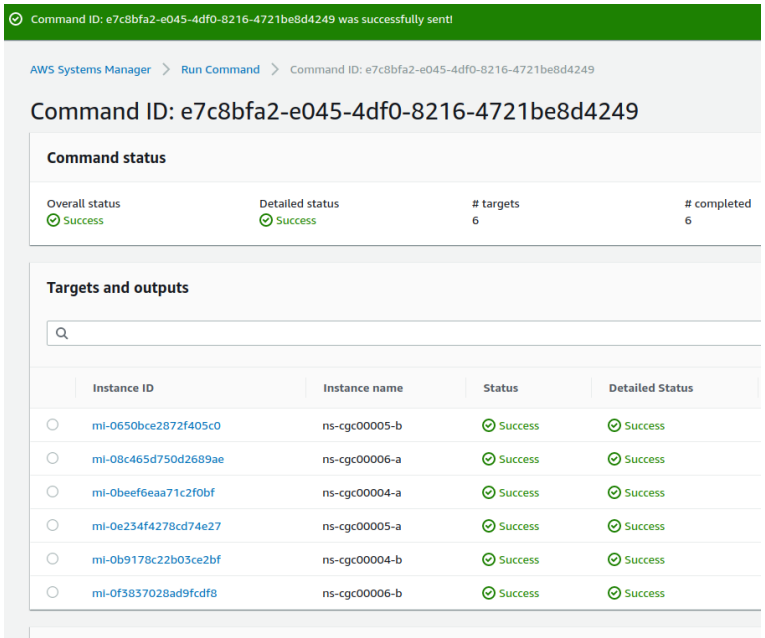
- Go to AWS Systems Manager -> Run Command -> and Select "MHB-CSC-Refresh-Private-Access-Peers-URL"



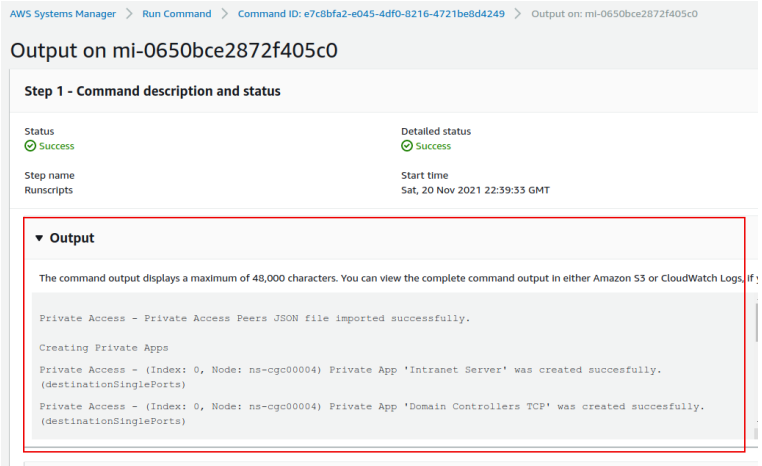
- Move down the screen and select all CSCs:



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

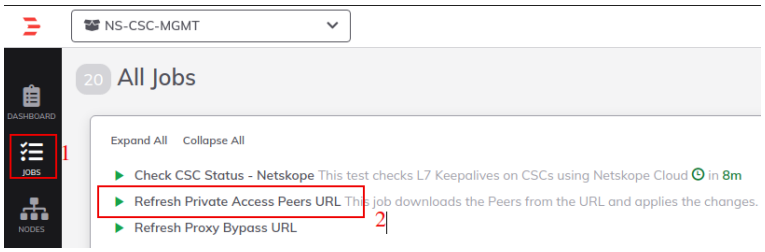


- To see the individual result, right click on the Instance ID and open it on a new TAB. Check the "Output"

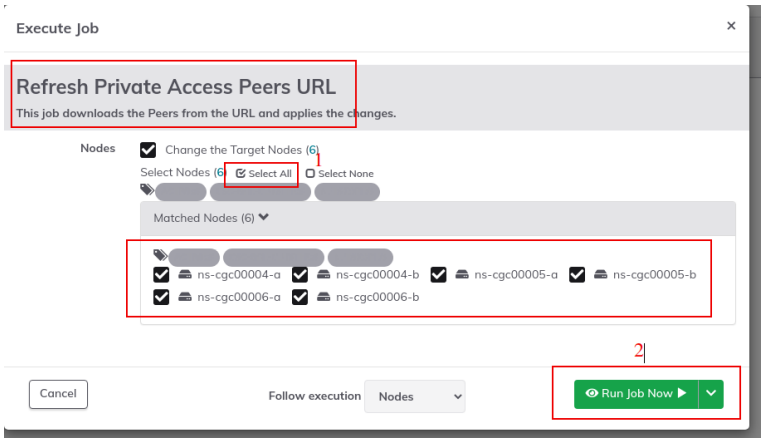


Using Rundeck

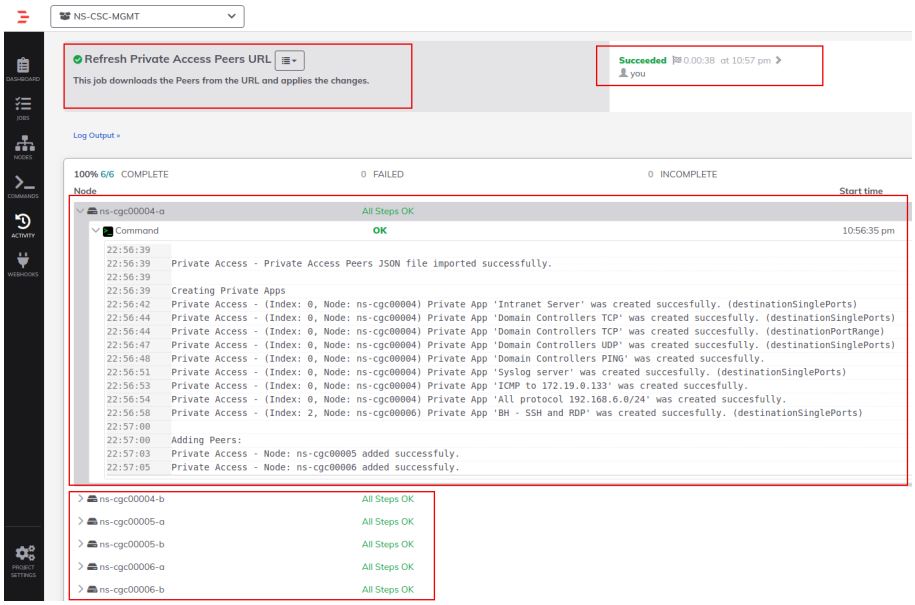
- Go to the Project <name> -> All Jobs -> Run " Refresh Private Access Peers URL"



- Select ALL nodes and click Run.



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



9.3.4.2 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!

9.4 Show Configurations and Status Private Access.

9.4.1 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
```

9.4.1.1 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": [
```

9.4.1.2 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": [
            ""
```

9.4.1.3 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": []
    }
  ]
}
```

9.4.1.4 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"
      ]
    }
  ]
}
```

9.4.2 Using AWS Systems Manager or Rundeck

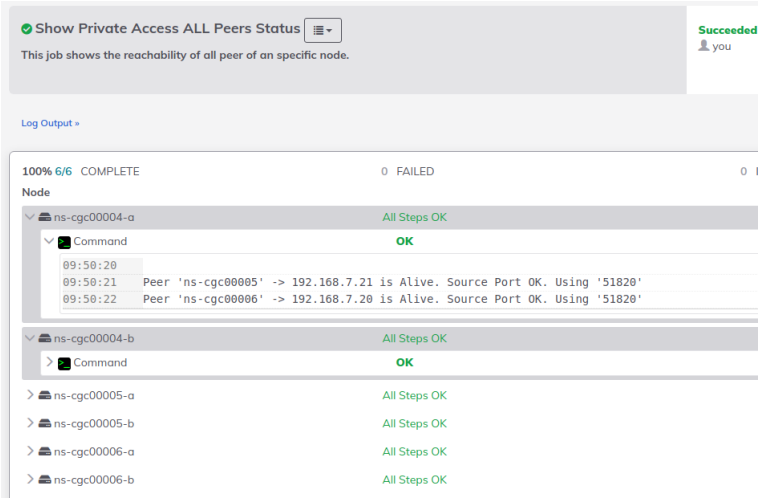
In this case, the information provided is only "Show ALL Peer Status"

9.4.2.1 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:

9.4.2.2 Rundeck

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



9.5 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
```

10 Remote Management

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

Both AWS Systems Manager and Rundeck can "Run Commands". If you want to use another system, here is the commands table with the tests and commands to run.

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

10.1 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".

10.1.1 Create Documents

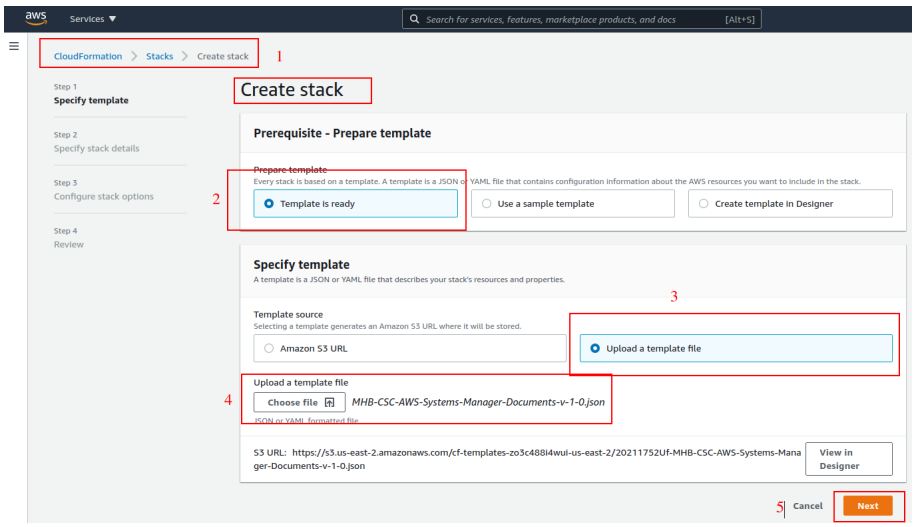
We provide a CloudFormation template to create all "Documents" in one shot.

Steps:

1. Download the CloudFormation template from:

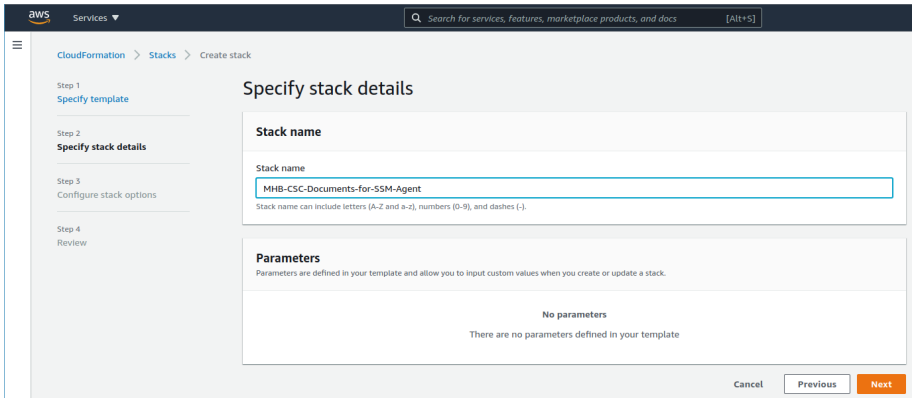
https://maidenheadbridge.freshdesk.com/support/solutions/articles/33000280930-create-documents-to-manage-the-csc-via-aws-systems-manager

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



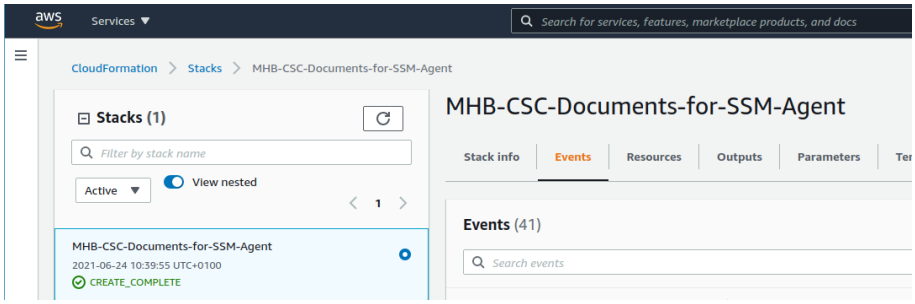
3. Click next.

4. Put the Stack Name

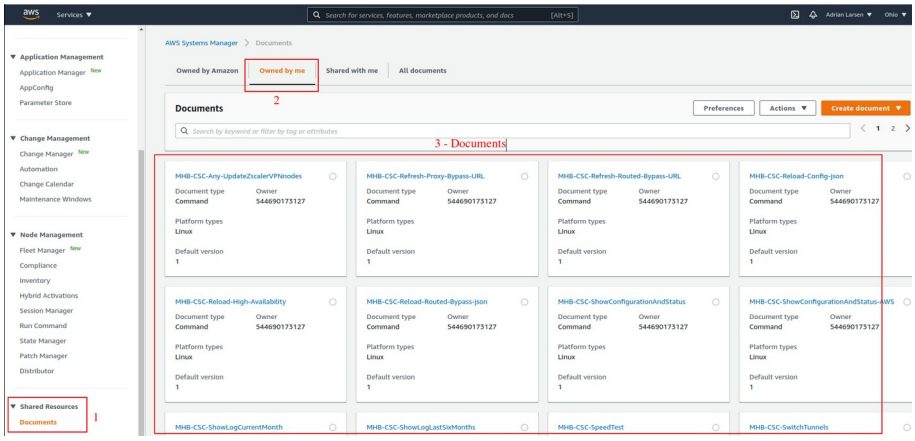


5. Click Next -> Next -> Create Stack.

6. Wait the Stack to complete.



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



8. Done!

10.1.2 Run Commands

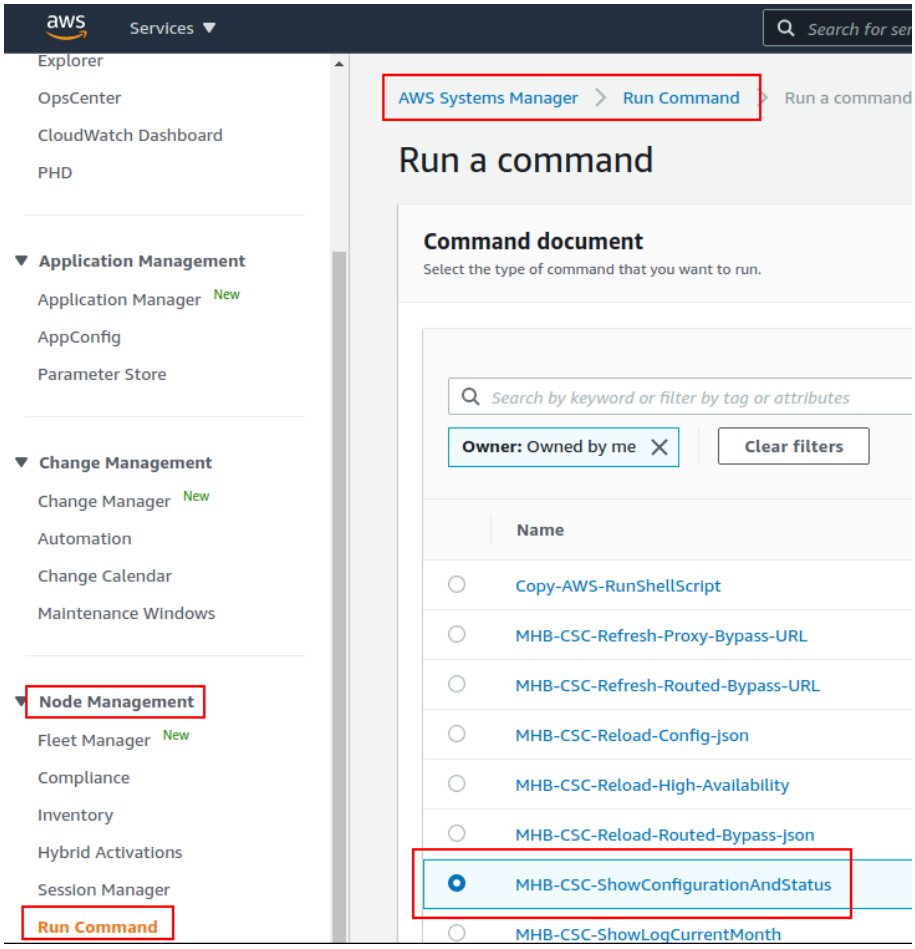
After you have created the Documents, you are ready to Run Commands on the CSC.

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

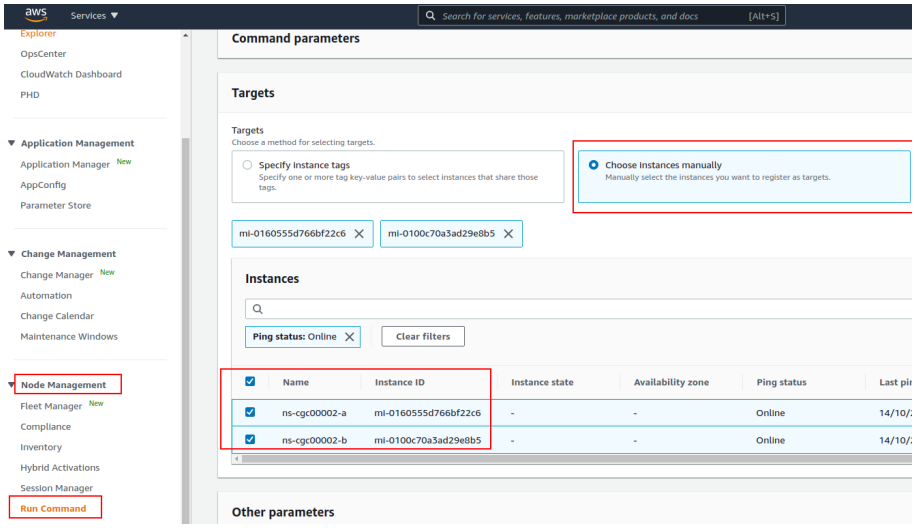
To "Run Commands", go to AWS Systems Manager → Instances & Nodes → Run Command.

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

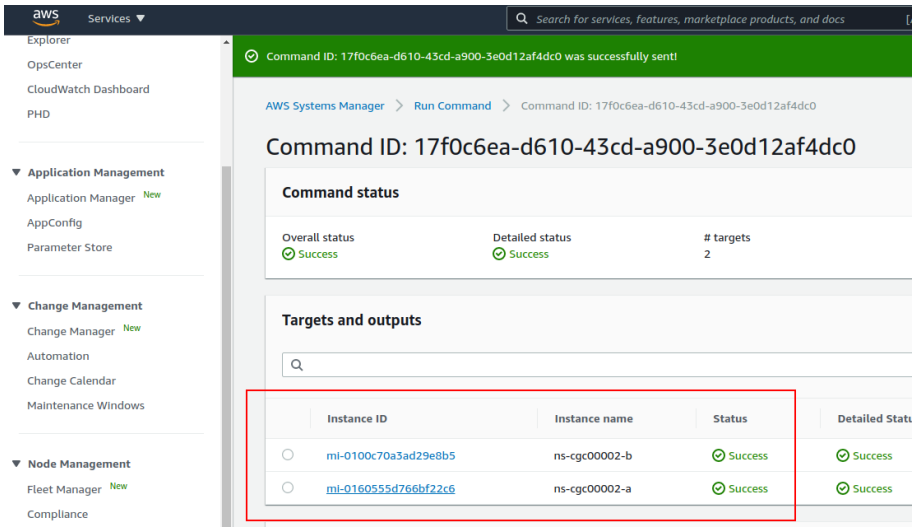
1. Run a Command
2. Select the Document created (Tip: Select "Owned by me")



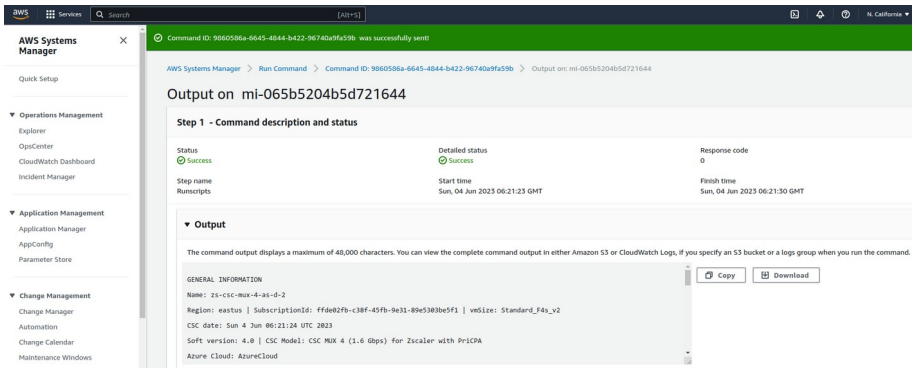
3. Scroll down and Select the Instances



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
```

10.1.3 List of Documents available for "Run Command"

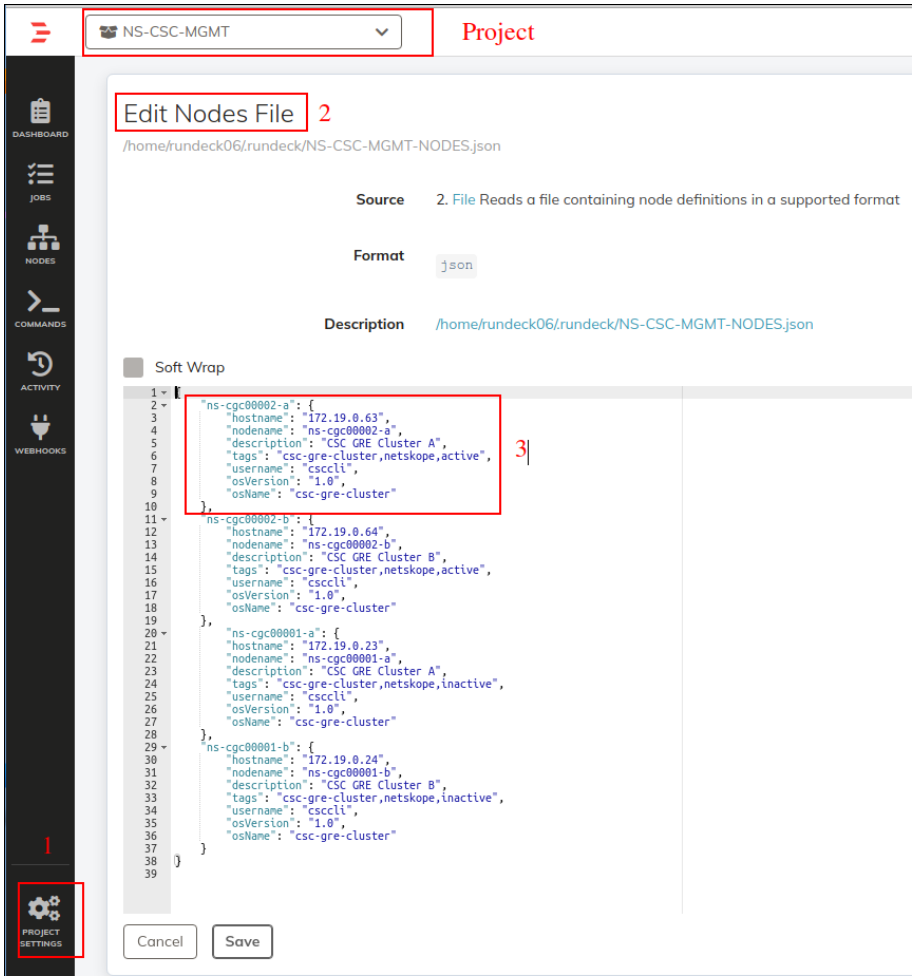
1. "MHB-CSC-ShowConfigurationAndStatus": Executes "Show Configuration and Status"
2. "MHB-CSC-SpeedTest": Performs speedtest.net on the CSC.
3. "MHB-CSC-TraceRouteAndLatencyTest": Performs MyTraceRoute test against the Primary and Secondary ZEN. It also does a Reverse Test from the tunnel active to your 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. (deprecated)
10. "MHB-CSC-Reload-High-Availability": Reloads the values of highAvailability.json file. (deprecated)
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.

10.2 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:

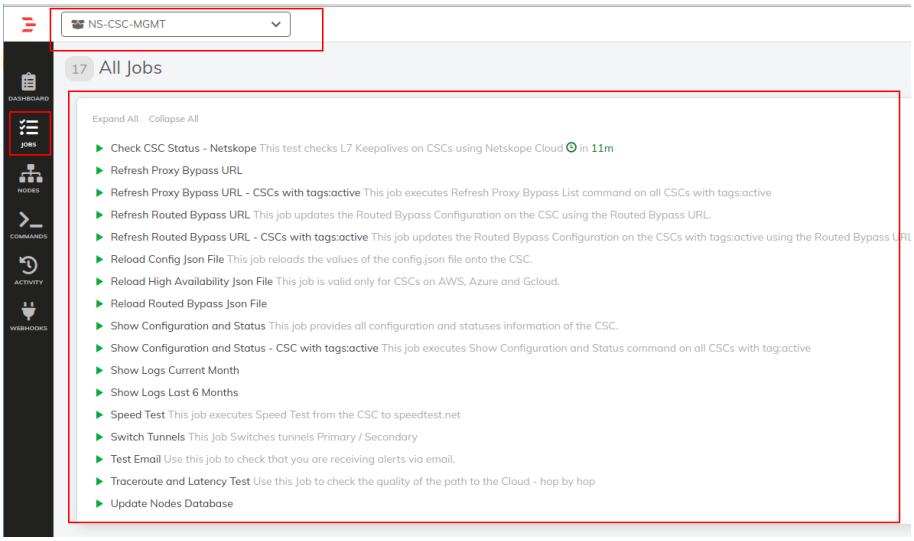
1. Install Rundeck. Intructions at: https://www.rundeck.com/open-source
2. Create a Project.
3. Enable user "csccli" and setup the SSH Public key on each CSC.
4. On the Project, setup the SSH Private and define the nodes:



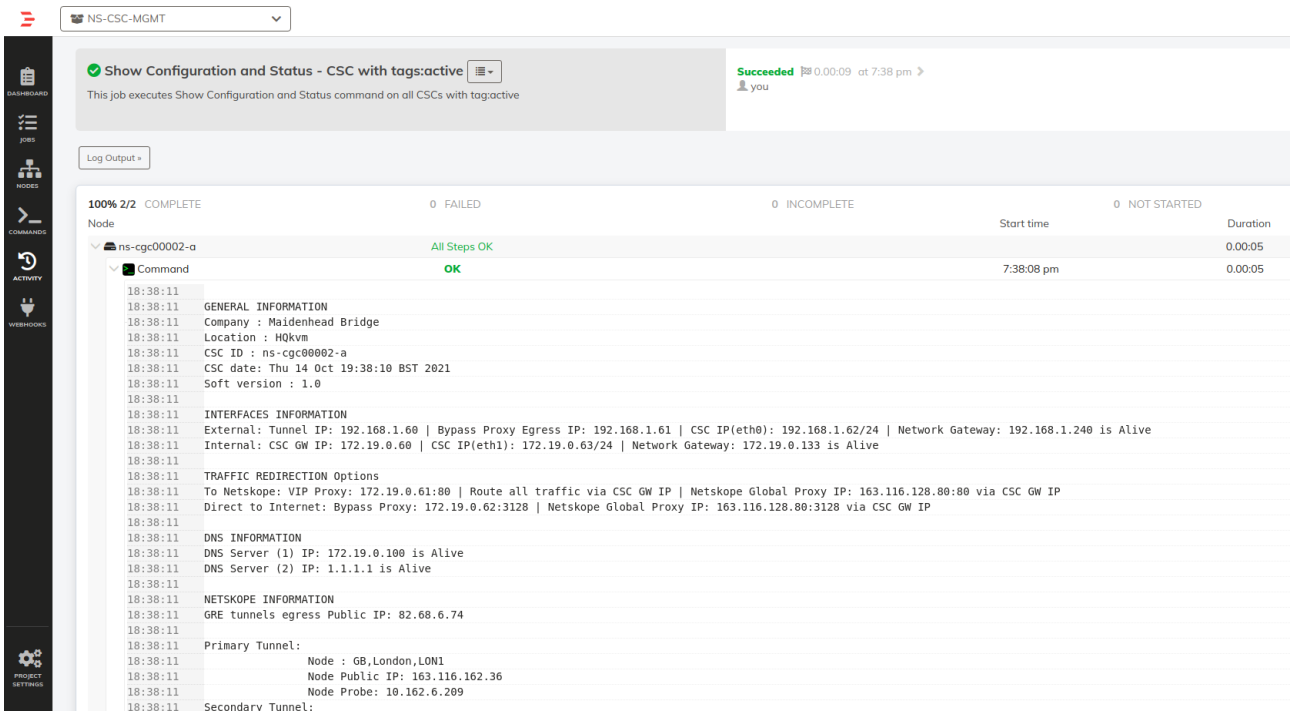
5. Create the jobs. Please, contact Support at http://support.maidenheadbridge.com for the latest Job List.

10.2.1 Jobs

The following screen shows the list of Jobs available.



10.2.2 Running job "Show Configuration and Status"



11 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.

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

11.1 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.

11.2 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.

12 Appendixes

12.1 Appendix A: Release Notes

12.1.1 Version 4.0

Version 4.0 comes with the following enhancements:

- New! Private Cloud Private Access: PriCPA is a unique 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 whom inside your Private Cloud.
- New! Proxy Bypass Advanced Mode: This functionality was created for servers and devices with Explicit Proxy settings. It provides connectivity to Zscaler (upstream Proxy), DIRECT via local public IP and also connectivity to internal websites.
- 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 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.
- New! TCPdump functionality is provided via the Admin console for easy troubleshooting of IP traffic.
- New! Netscanner functionality helps to find internal Apps behind the CSC.
- Base OS upgraded to Ubuntu 22.04

12.1.2 Version 3.0

Version 3.0 comes with the following enhancements:

1. New! Zscaler API integration for the automatic creation of Static IP, GRE Tunnels, ZEN node Selection and Location on the Zscaler console.
2. New! Routed Bypass functionality. Routed Bypass functionality allows to create Layer 4 bypasses when traffic is routed via the CSC's Gateway IP. You can do bypasses per Source/Destination IP/Subnet, protocol TCP or UDP and any port range.

3. New! When the CSC switches to the Secondary node, you can decide to remain using the Secondary node (returnToPrimaryTunnel=false) or change back to the Primary node (returnToPrimaryTunnel=true) after 10 minutes of stability of the Primary Tunnel.
4. Cloud DNS setting is now AWS DNS (primary) and Google DNS 8.8.8.8 (secondary)
5. Some cosmetic changes on Menus.
6. Base OS is now Ubuntu 20.04.

12.1.3 Version 2.8

Version 2.8 comes with the following enhancements:

1. New! You can configure multiple Route Tables on High Availability.
2. New! OS base system is Ubuntu 18.04.4 LTS (bionic).
3. Updated “Configuration and Status” Menu.
4. Forced route to AWS DNS via eth1.

12.1.4 Version 2.7

Version 2.7 comes with the following enhancements:

1. New! “High Availability changing default route”. You can now configure a pair of CSC on High Availability to automatically manipulate the default route to the internet via Zscaler.
2. Updated “Configuration and Status” Menu.
3. MTR (MyTraceRoute Test) now runs directly via TCP/80 to the ZEN Primary and Secondary.

12.1.5 Version 2.6

Version 2.6 comes with the following enhancements:

1. Added to Wizard Configuration menu: From the Wizard, you can change the Syslog Servers and GRE tunnel IPs, DNS Servers and Bypass PAC URL.
2. New! Switch tunnels configuration wizard. In some circumstances, customers asked us an easy way to switch tunnels Primary / Secondary. Now is possible to do with a single command.
3. Logs to Syslog server. On version 2.6, you can set up one or two Syslog servers to send the information about Tunnel Status.
4. Updated “Configuration and Status” Menu.

12.2 Appendix B: configUserData.json file

12.2.1 configUserData.json file


```
{
  "model": "zs-aws-csc-gre",
  "type": "configUserData",
  "version": "2.0",
  "dns": {
    "useCloudDns": "yes",
    "primaryDnsIp": "",
    "secondaryDnsIp": ""
  },
  "awsSsmAgent": {
    "enable": "no",
    "activationCode": "",
    "activationId": "",
    "awsRegion": ""
  },
  "syslog": {
    "enable": "no",
    "primaryServer": {
      "ip": "",
      "port": ""
    },
    "secondaryServer": {
      "ip": "",
      "port": ""
    },
    "trafficLogs": {
      "enable": "no"
    }
  },
  "bypasses": {
    "proxyBypass": {
      "standardMode": {
        "pacUrl": ""
      }
    },
    "routedBypass": {
      "jsonUrl": ""
    }
  },
  "priCPA": {
    "enable": "no",
    "nodeName": "",
    "location": "",
    "description": "",
    "publicUdpPort": "",
    "privateCirdIp": "",
    "persistentKeepAlive": "no",
    "peersJsonFileUrl": "",
    "remoteManagementNetworks": []
  },
  "sshRestrictions": {
    "eth1": {
      "enable": "no",
      "allowedNetworks": []
```

```
    },
    "wg0": {
      "enable": "no",
      "allowedNetworks": []
    }
  },
  "adminManagement": {
    "csccli": {
      "enable": "no",
      "sshPublicKey": ""
    }
  },
  "zscalerApi": {
    "apiTokenID": "",
    "cloudName": "",
    "tunnelRedundancy": {
      "returnToPrimaryTunnel": true
    },
    "nodeSelection": {
      "withinCountryPreferred": true
    },
    "location": {
      "name": "",
      "country": "",
      "tz": "",
      "ipAddresses": [
        "auto"
      ],
      "authRequired": true,
      "xffForwardEnabled": true,
      "surrogateIP": true,
      "idleTimeInMinutes": 480,
      "displayTimeUnit": "MINUTE",
      "surrogateIPEnforcedForKnownBrowsers": true,
      "surrogateRefreshTimeInMinutes": 120,
      "surrogateRefreshTimeUnit": "MINUTE",
      "ofwEnabled": true,
      "ipsControl": true
    }
  }
}
```

12.2.2 userDataConfig.json file fields and values

12.2.2.1 Fixed values - do not change

```
"model": "zs-aws-csc-gre",
"type": "configUserData",
"version": "2.0",
```

12.2.2.2 DNS configuration

By default ("useCloudDns": "yes",) the CSC uses AWS DNS servers and Google (8.8.8.8, 8.8.4.4)

```
"dns": {
  "useCloudDns": "yes",
  "primaryDnsIp": "",
  "secondaryDnsIp": ""
},
```

If you want to use your DNS servers, for example, 192.168.1.100 and 192.168.1.101, you need to configure this section in the following way:

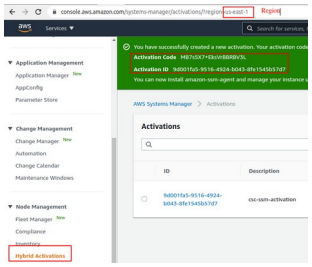
```
"dns": {
  "useCloudDns": "no",
  "primaryDnsIp": "192.168.1.100",
  "secondaryDnsIp": "192.168.1.101"
},
```

12.2.2.3 AWS SSM Agent

The AWS SSM Agent is used to monitor de CSC via AWS Systems Manager (Fleet Manager).

```
"awsSsmAgent": {
  "enable": "no",
  "activationCode": "",
  "activationId": "",
  "awsRegion": ""
},
```

Set "enable": "yes" and input here the values of the Activation Code, Activation ID, and AWS Region of the "Hybrid Activation" of the AWS Systems Manager.



12.2.2.4 Syslog Configuration

By default, the CSC collects System Logs internally, and Traffic Logs are disabled. If you want to send the System Logs to a Syslog Server, please configure the IP and Port. If you will also send the Traffic Logs, please enable: "yes".

Example: I want to send "System Logs" to Syslog Servers are 192.168.1.100 and 192.168.1.101, using port 5514 and "Traffic Logs" are disabled.

```
"syslog": {
  "enable": "no",
  "primaryServer": {
    "ip": "192.168.1.100",
    "port": "5514"
  },
  "secondaryServer": {
    "ip": "192.168.1.101",
    "port": "5514"
  },
  "trafficLogs": {
    "enable": "no"
  }
},
```

12.2.2.5 Bypasses: Proxy and Routed Bypass

By default, Proxy and Routed bypasses are disabled. In this section you can configure the PAC URL for the Proxy Bypass (Standard Mode) and the JSON file URL for the Routed Bypass.

Example:

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

Route Bypass JSON file URL: https://mhb-csc-pac.s3.amazonaws.com/routedBypassRulesFile.json

```
"bypasses": {
  "proxyBypass": {
    "standardMode": {
      "pacUrl": "https://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac"
    }
  },
  "routedBypass": {
    "jsonUrl": "https://mhb-csc-pac.s3.amazonaws.com/routedBypassRulesFile.json"
  }
},
```

12.2.2.6 Private Cloud Private Access (PriCPA)

By default, PriCPA is disabled on the CSC. Via configUserData.json file you can create the PriCPA Local Configuration, add the "peerJsonFileUrl" and "remoteManagementNetworks".

```
"priCPA": {
  "enable": "no",
  "nodeName": "",
  "location": "",
  "description": "",
  "publicUdpPort": "",
  "privateCirdIp": "",
  "persistentKeepAlive": "no",
  "peersJsonFileUrl": "",
  "remoteManagementNetworks": []
},
```

If you pass PriCPA values via configUserdata.json and also enable the AWS SSM agent, you can get access to the CSC's SSH Admin console without requiring a bastion host behind the eth1 of the CSC.

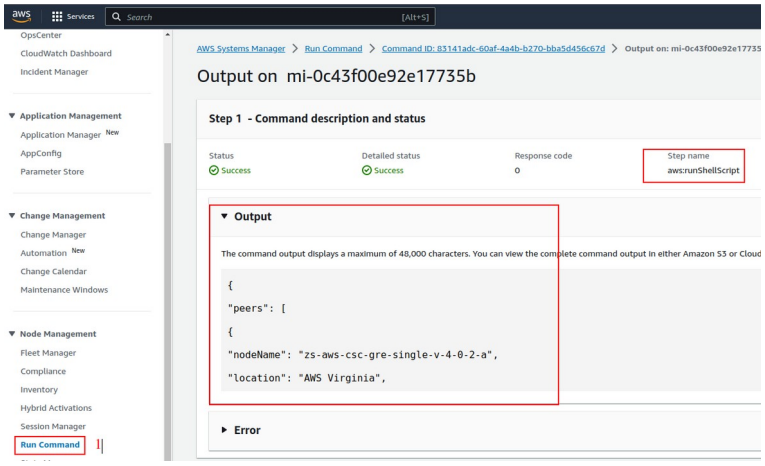
Here an example:

```
"priCPA": {
  "enable": "yes",
  "nodeName": "zs-aws-csc-gre-single-v-4-0-2-a",
  "location": "AWS Virginia",
  "description": "Develpment Node Zscaler with PriCPA",
  "publicUdpPort": "51280",
  "privateCirdIp": "192.168.7.22/24",
  "persistentKeepAlive": "no",
  "peersJsonFileUrl": "https://mhb-zscaler-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
  "remoteManagementNetworks": [
    "10.63.0.0/24",
    "172.19.0.0/24",
    "192.168.1.0/24",
    "192.168.6.0/24"
  ]
},
```

After the initial boot of the CSC, via AWS SSM, run the "aws:runShellScript" with the command:

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

and retrieve the PriCPA Local configuration:




```
{
 "peers": [
  {
   "nodeName": "zs-aws-csc-gre-single-v-4-0-2-a",
   "location": "AWS Virginia",
   "description": "Develpment Node Zscaler with PriCPA",
   "publicKey": "SGf6U5XVu3GT0jvClups8iBuvoK7R+irD6MuhduatU8=",
   "publicIpAndUdpPort": "18.214.102.160:51280",
   "privateCirdIp": "192.168.7.22/24",
   "persistentKeepAlive": "no",
   "networks": [],
   "privateApps": []
  }
 ]
}
```


Copy the section in bold to your "privatePeersConfig.json" file, and add the "networks" behind the CSC and the "privateApps".

Example:

```
{
    "nodeName": "zs-aws-csc-gre-single-v-4-0-2-a",
    "location": "AWS Virginia",
    "description": "Develpment Node Zscaler with PriCPA",
    "publicKey": "SGf6U5XVu3GT0jvCIups8iBuvoK7R+irD6MuhduatU8=",
    "publicIpAndUdpPort": "18.214.102.160:51280",
    "privateCirdIp": "192.168.7.22/24",
    "persistentKeepAlive": "no",
    "networks": [
        "10.3.20.0/24"
    ],
    "privateApps": [
        {
            "description": "Allow SSH to net 10.3.20.0/24 from MGMT Networks",
            "ipProtocol": "tcp",
            "sourceCirdIp": [
                "10.63.0.0/24",
                "172.19.0.0/24",
                "192.168.1.0/24",
                "192.168.6.0/24"
            ],
            "destinationCirdIp": [
                "10.3.20.0/24"
            ],
            "destinationSinglePorts": [
                "22"
            ],
            "destinationPortRange": {
                "fromPort": "",
                "toPort": ""
            }
        }
    ]
},
```

Local Config

Network behind the CSC

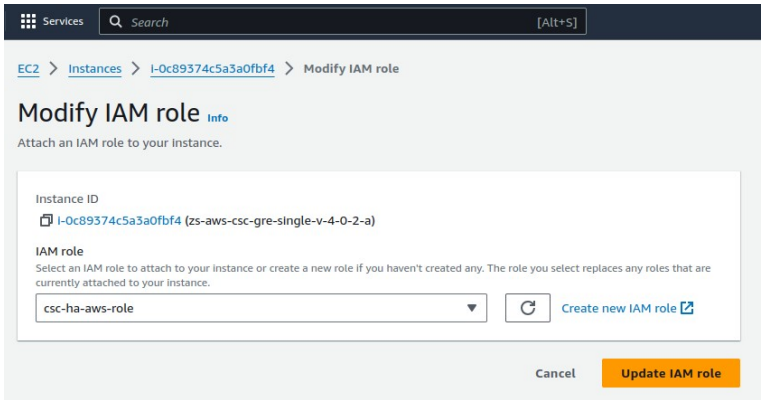
App

Before deploying the updated "privateAccessPeersFile.json " to all CSCs on the PriCPA Cloud, you need to apply an IAM role to this CSC.

Create an IAM role with the following permissions and apply it to the CSC:

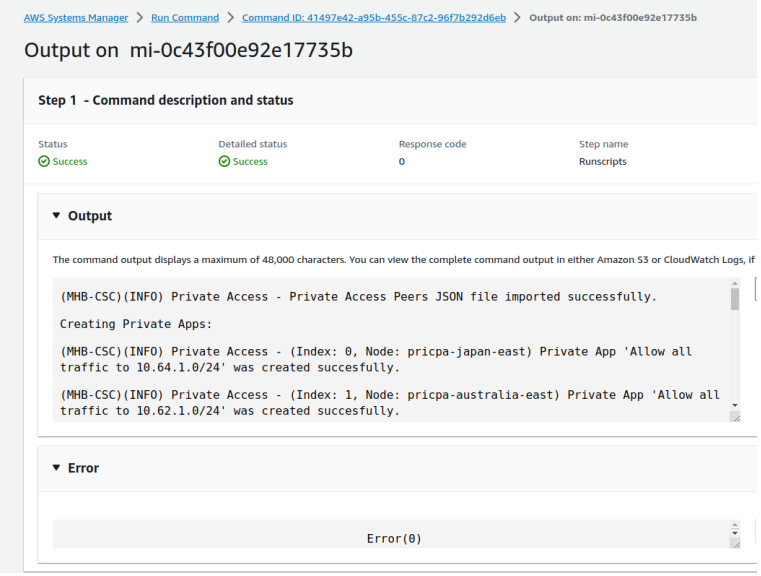
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:DisassociateAddress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:DescribeAddresses",
        "ec2:DescribeInstances",
        "sns:ListSubscriptionsByTopic",
        "ec2:CreateTags",
        "ec2:DescribeSecurityGroups",
        "ec2:ReplaceRoute",
        "ec2:RevokeSecurityGroupIngress",
        "sns:Publish",
        "ec2:DescribeSecurityGroupRules",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:AssociateAddress",
        "ec2:DescribeRouteTables"
      ],
      "Resource": "*"
    }
  ]
}
```

Apply the IAM role to the CSC. (Right click the Instance -> Security -> Modify IAM Role



Click "Update IAM Role"

For the next step, run the document "MHB-CSC-Refresh-Private-Access-Peers-URL" on all CSCs on the PriCPA cloud and wait to finish.



Done!

You can SSH the CSC from the Management Networks defined.

12.2.2.7 SSH restrictions

By default, it is allowed to SSH the eth1 interface from anywhere and the wg0 from any other CSC that belongs to the PriCPA subnet.

Using this section, you can select the subnets from you want to SSH the CSC.

```
  "sshRestrictions": {
    "eth1": {
      "enable": "no",
      "allowedNetworks": []
    },
    "wg0": {
      "enable": "no",
      "allowedNetworks": []
    }
  },
```

Example:

```
"sshRestrictions": {
    "eth1": {
      "enable": "yes",
      "allowedNetworks": [
        "172.19.0.0/24",
        "192.168.1.0/24",
        "192.168.6.0/24",
        "10.3.200.0/24"
      ]
    },
    "wg0": {
      "enable": "yes",
      "allowedNetworks": [
        "172.19.0.0/24",
        "192.168.1.0/24",
        "192.168.6.0/24",
        "10.3.200.0/24"
      ]
    }
  },
```

12.2.2.8 Admin Management

After the initial launch, the user cscadmin and ubuntu are available. In this section, you can enable the user "csccli" and provide the SSH public key.

```
  "adminManagement": {
    "csccli": {
      "enable": "no",
      "sshPublicKey": ""
    }
  },
```

12.2.2.9 Zscaler APi values

Please, see next Appendix C for details.

```
"zscalerApi": {
  "apiTokenID": "",
  "cloudName": "",
  "tunnelRedundancy": {
    "returnToPrimaryTunnel": true
  },
  "nodeSelection": {
    "withinCountryPreferred": true
  },
  "location": {
    "name": "",
    "country": "",
    "tz": "",
    "ipAddresses": [
      "auto"
    ],
    "authRequired": true,
    "xffForwardEnabled": true,
    "surrogateIP": true,
    "idleTimeInMinutes": 480,
    "displayTimeUnit": "MINUTE",
    "surrogateIPEnforcedForKnownBrowsers": true,
    "surrogateRefreshTimeInMinutes": 120,
    "surrogateRefreshTimeUnit": "MINUTE",
    "ofwEnabled": true,
    "ipsControl": true
  }
}
```

12.3 Appendix C: Advanced Mode Deployment (using Zscaler API)

In Advanced Mode Deployment, you will create the CSC, AWS resources and Zscaler Location in one shot.

12.3.1 Prerequisites

The prerequisites are the same as Basic Mode Deployment: External Subnet, Internet Subnet, SSH Key, plus the addition to pasting the contents of the "userDataConfig.json" file on the UserData field of the Cloudformation template.

12.3.2 zscalerApi values

In this Appendix C, we will explain the section of the Zscale API of the "userDataConfig.json" file. For other values, please see Appendix B.

userDataConfig.json - Zscaler API section

```
"zscalerApi": {
    "apiTokenID": "",
    "cloudName": "",
    "tunnelRedundancy": {
      "returnToPrimaryTunnel": true
    },
    "nodeSelection": {
      "withinCountryPreferred": true
    },
    "location": {
      "name": "",
      "country": "",
      "tz": "",
      "ipAddresses": [
        "auto"
      ],
      "authRequired": true,
      "xffForwardEnabled": true,
      "surrogateIP": true,
      "idleTimeInMinutes": 480,
      "displayTimeUnit": "MINUTE",
      "surrogateIPEnforcedForKnownBrowsers": true,
      "surrogateRefreshTimeInMinutes": 120,
      "surrogateRefreshTimeUnit": "MINUTE",
      "ofwEnabled": true,
      "ipsControl": true
    }
  }
```

12.3.2.1 apiTokenID

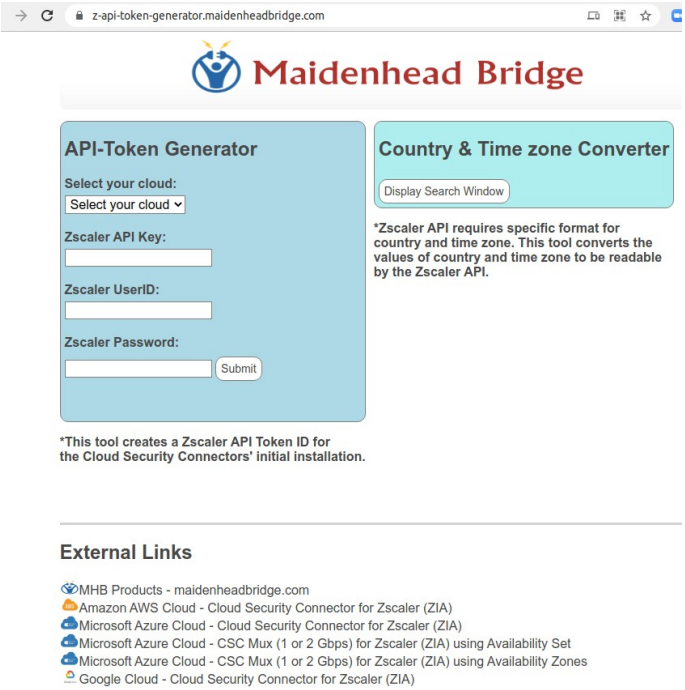
```
"apiTokenID": "",
```

When launching the CSC using Zscaler API, you need to generate a "Token" to allow the CSC to talk with the Zscaler API.

Important: The TokenID is not valid after 20 minutes of created. Also, a new TokenID must be generated each time you launch a CSC due to the CSC logoff the session at the end of the auto-provision process.

You can find how to generate a "Token" at: https://help.zscaler.com/zia/api-getting-started, section "Authenticate and create an API session".

Alternatively, you can use, at your own risk, a utility page we created for this purpose. The page is here: https://z-api-token-generator.maidenheadbridge.com/

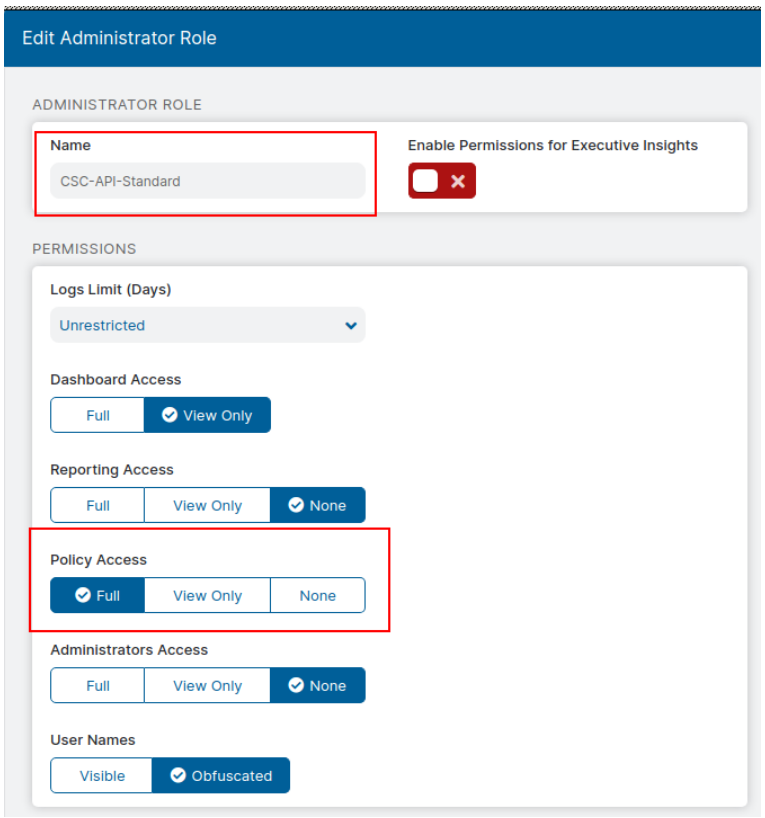


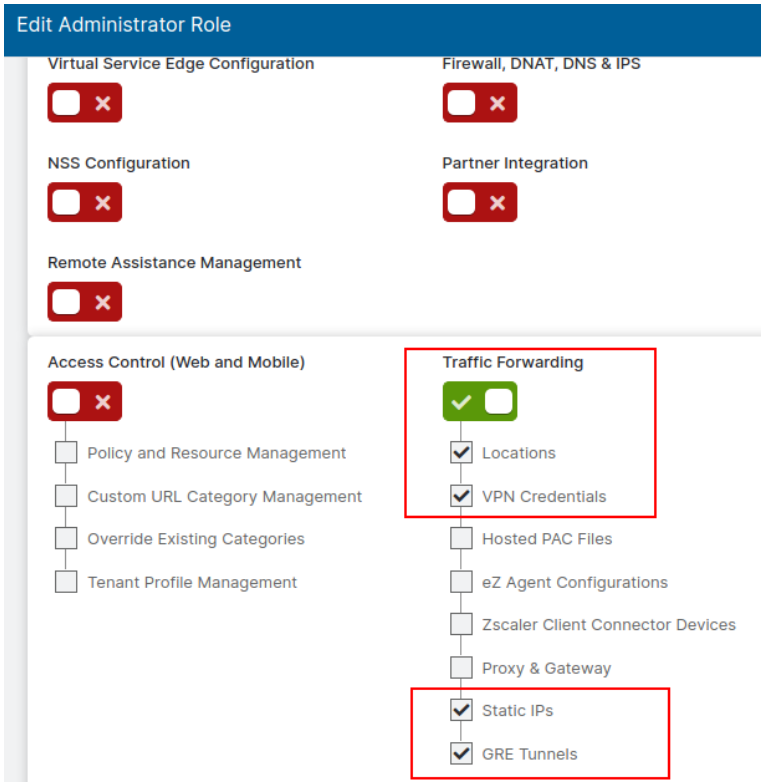
If you want to use the page https://z-api-token-generator.maidenheadbridge.com/ to generate the "API Token, " you have two options for creating the values of the Zscaler API Key, Username and Password on the Zscaler console. Option one uses the "Organization API Key, " and option two uses any unused API "SD-WAN" Partner Key. We will explain both.

Using "Organization API Key":

1. Enable API Key for your Organization and "Add API key" (Administration → API key Management).

2. Create a "Role" for the Admin User. Go to Administration → Role Management → Add Role. Allow Permissions: "Policies Full Access" and Functional Scope: “Traffic Forwarding” : Locations, Static IPs, GRE Tunnels (and VPN credentials if you want to use the same role for other CSC models).





You can deny other settings. The Role will look like this:

No.	Name	Full Access	View-Only Access	User Names	Functional Scope	Type
2	CSC-API-Standard	Policy	Dashboard	Obfuscated	GRE Tunnels, Locations, Static IPs, VPN Credentials	Standard Admin

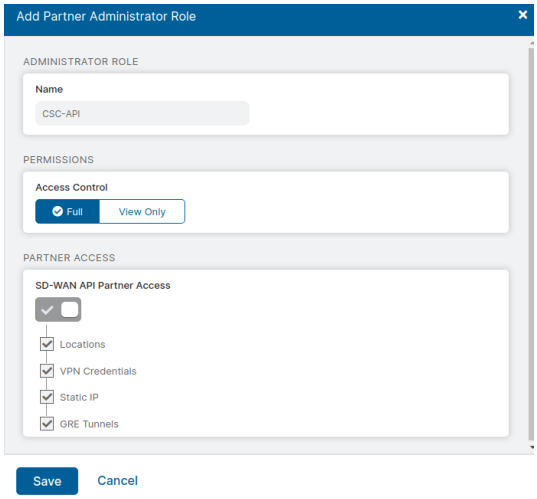
3. Create an Administrator User and apply the Role. Go to Administration → Administrator Management → Add Administrator. The Administrator User will look like this:

No.	Login ID	Name	Role	Scope	Login Type	Comments	Password Expired	Type
6	csc-api-standard@maidenheadbridge.com	CSC API Standard Admin	CSC-API-Standard	Organization	Password	---	false	Standard Admin

Done! Now you are the values requested: Zscaler API Key, Zscaler User ID / Password.

Using any "SDWAN API Key":

1. Go to Administration → Partner Integration → SDWAN → Add Partner Key. Select any vendor name that is not in use to create the key.
2. Create a Partner Role. Go to Administration → Role Management → Add Partner Administrator Role.

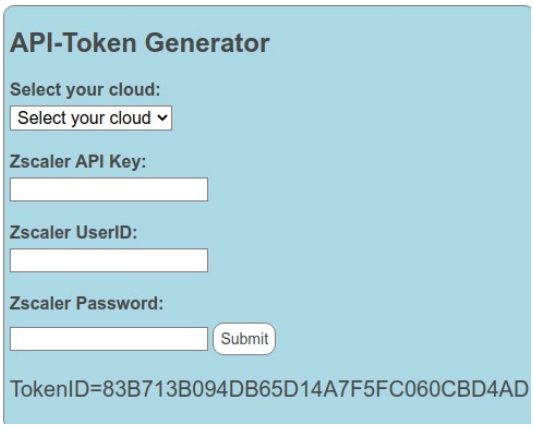


3. Create a Partner Administrator User and apply the Role. Go to Administration → Administrator Management → Add Partner Administrator. The Partner Administrator User will look like this:

No.	Login ID	Name	Role	Scope	Login Type	Comments	Password Expired	Type
11	mhb-partner@maidenheadbridge.com	MHB-PARTNER	CSC-API	Organization	Password	---	false	Partner Admin

Done! Now you are the values requested: Zscaler API Key, Zscaler User ID / Password.

The next step is to fill the values at https://z-api-token-generator.maidenheadbridge.com/ and to click "Submit" to obtain the "TokenID" value.



Use this value for "apiTokenID".

12.3.2.2 cloudName

```
"cloudName": "",
```

Insert here your Zscaler Cloud name: zscalerthree, zscloud, zscalertwo, zscaler, etc.

12.3.2.3 returnToPrimaryTunnel

```
"tunnelRedundancy": {
  "returnToPrimaryTunnel": true
},
```

Please select 'true' if you want the CSC to return to Primary tunnel (after 10 min of stability) when using Secondary tunnel.
Select 'false' if you want to remain using Secondary Tunnel and not to return to Primary.
(Secondary will be nominated as 'new' Primary)

12.3.2.4 nodeSelection

```
"nodeSelection": {
  "withinCountryPreferred": true
},
```

Select "withinCountryPreferred": true if you want the Primary and Secondary Node Zscaler ZEN nodes to belong to the same country as the preference of selection instead of geo-proximity.

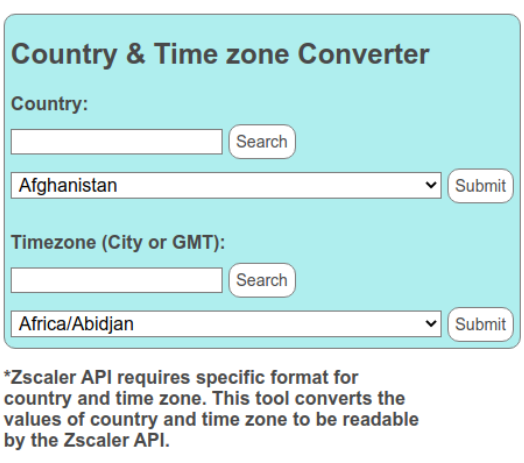
12.3.2.4.1 location

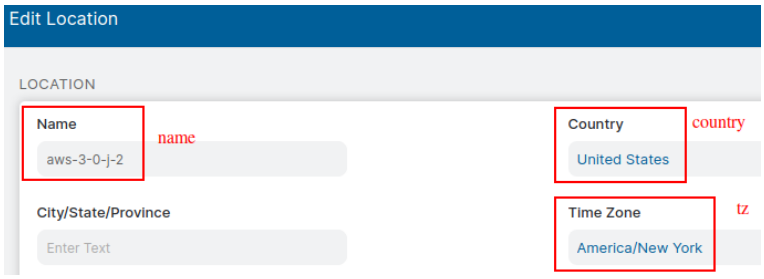
```
"location": {
  "name": "",
  "country": "",
  "tz": "",
  "ipAddresses": [
    "auto"
  ],
  "authRequired": true,
  "xffForwardEnabled": false,
  "surrogateIP": true,
  "idleTimeInMinutes": 480,
  "displayTimeUnit": "MINUTE",
  "surrogateIPEnforcedForKnownBrowsers": false,
  "surrogateRefreshTimeInMinutes": 120,
  "surrogateRefreshTimeUnit": "MINUTE",
  "ofwEnabled": true,
  "ipsControl": true
},
```

These values are the Location values to configure on the Zscaler Console. Except for ipAddress that is not configurable (leave "auto"), the rest of the values are configurable.

Values for "name", "country" and "tz"

IMPORTANT: *The API requires special format for "country" and "tz". Use https://z-api-token-generator.maidenheadbridge.com/ to generate the proper values.*

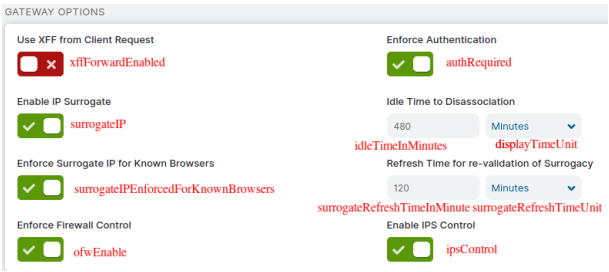




For example, if you want to use the values shown in the image above, you need to configure:

```
"location": {
  "name": "aws-3-0-j-2",
  "country": "UNITED_STATES",
  "tz": "UNITED_STATES_AMERICA_NEW_YORK",
  "ipAddresses": [
    "auto"
  ],
  "authRequired": true,
  "xffForwardEnabled": false,
  "surrogateIP": true,
  "idleTimeInMinutes": 480,
  "displayTimeUnit": "MINUTE",
  "surrogateIPEnforcedForKnownBrowsers": true,
  "surrogateRefreshTimeInMinutes": 120,
  "surrogateRefreshTimeUnit": "MINUTE",
  "ofwEnabled": true,
  "ipsControl": true
},
```

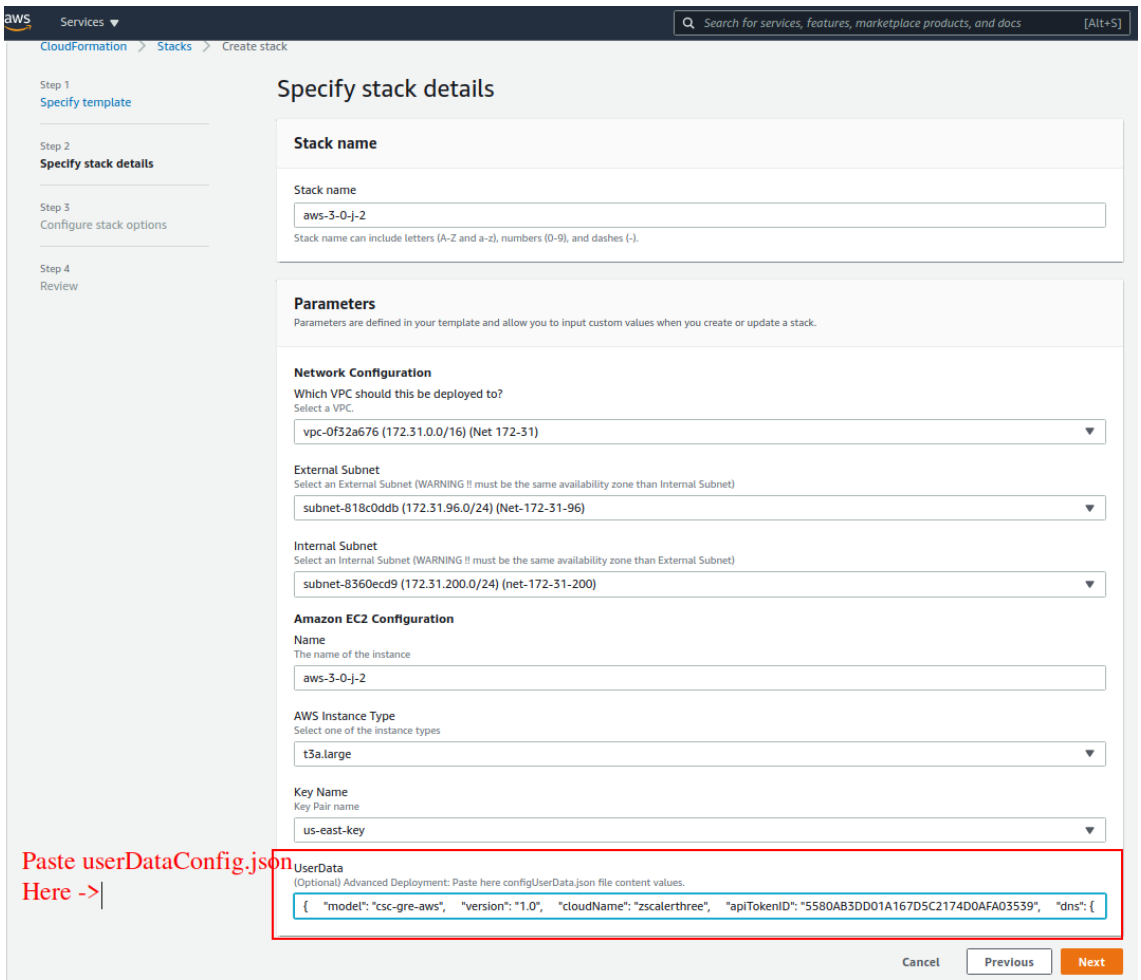
The rest of the values correspond to "GATEWAY OPTIONS."



12.3.3 Advanced Mode Deployment using CloudFormation

In Advanced Deployment, you must fill in the last section of the CloudFormation template. (UserData) with the Zscaler API data.

Copy the contents of the userDataConfig.json file and paste it into the section UserData.



and Click "Next", "Next" , "Create Stack".

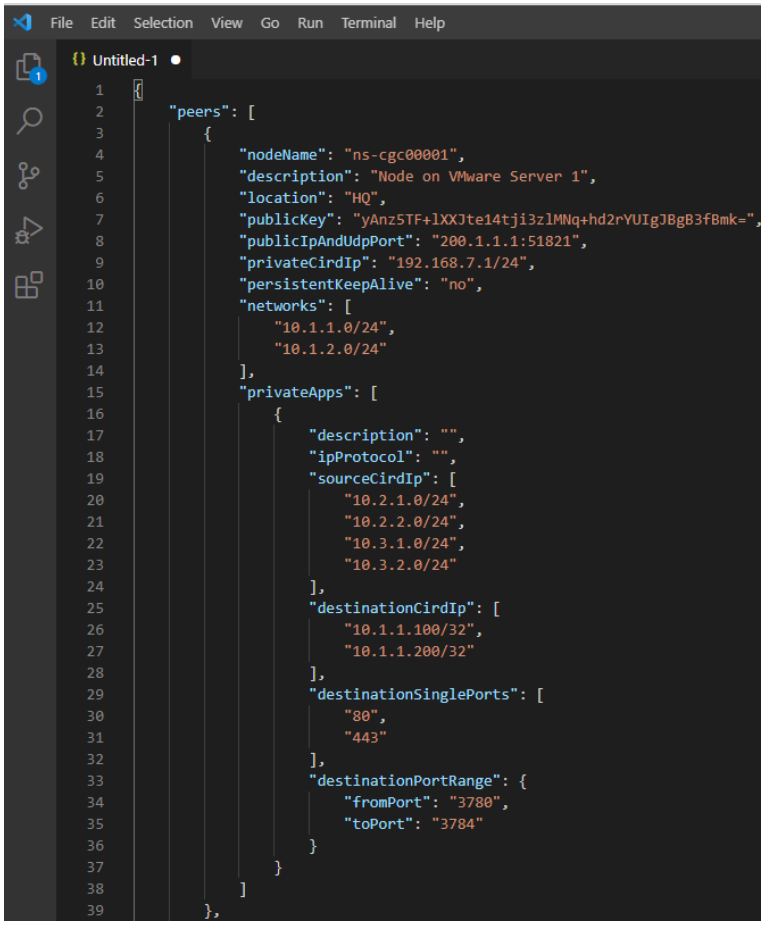
If you have the Syslog configured, you will see the creation of the resources during the deployment.

```
(MHB-CSC)(UP) CSC GRE for AWS was powered ON: Mon 21 Jun 15:34:58 UTC 2021
(MHB-CSC)(INFO) Routed Bypass Rules JSON file integrity is OK
(MHB-CSC)(INFO) DNS configured using AWS 169.254.169.253 and Google 8.8.8.8 servers
(MHB-CSC)(INFO) SYSLOG configured using Primary= 172.31.200.163, Secondary= none and TCP port= 514
(MHB-CSC)(INFO) AWS SSM Agent is  active (running) since Mon 2021-06-21 15:35:02 UTC; 539ms ago. Registration values: {"ManagedInstanceID":"mi-08c19346db4f4ce81","Region":"us-east-1"}
(MHB-CSC)(INFO) Bypass List updated sucessfully.(using PAC URL http://pac.zscalerthree.net/RdwWltSPqBFN/az-csc-bypass.pac)
(MHB-CSC)(INFO) Zscaler API: StaticIp 54.159.82.127 was added to your Zscaler console
(MHB-CSC)(INFO) Zscaler API: GRE Tunnel with Source IP: 54.159.82.127 was added to your Zscaler console
(MHB-CSC)(INFO) Zscaler API: Location aws-3-0-j-1 with Source IP: 54.159.82.127 was added to your Zscaler console
(MHB-CSC)(INFO) Zscaler API: Activation successful
(MHB-CSC)(INFO) Zscaler API: API Session Ended
(MHB-CSC)(INFO) Routed Bypass Rules JSON file created successfuly from configUserData.json (using Routed Bypass URL https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json).
```

12.4 Appendix D: JSON formatters (Visual Code, Notepad ++)

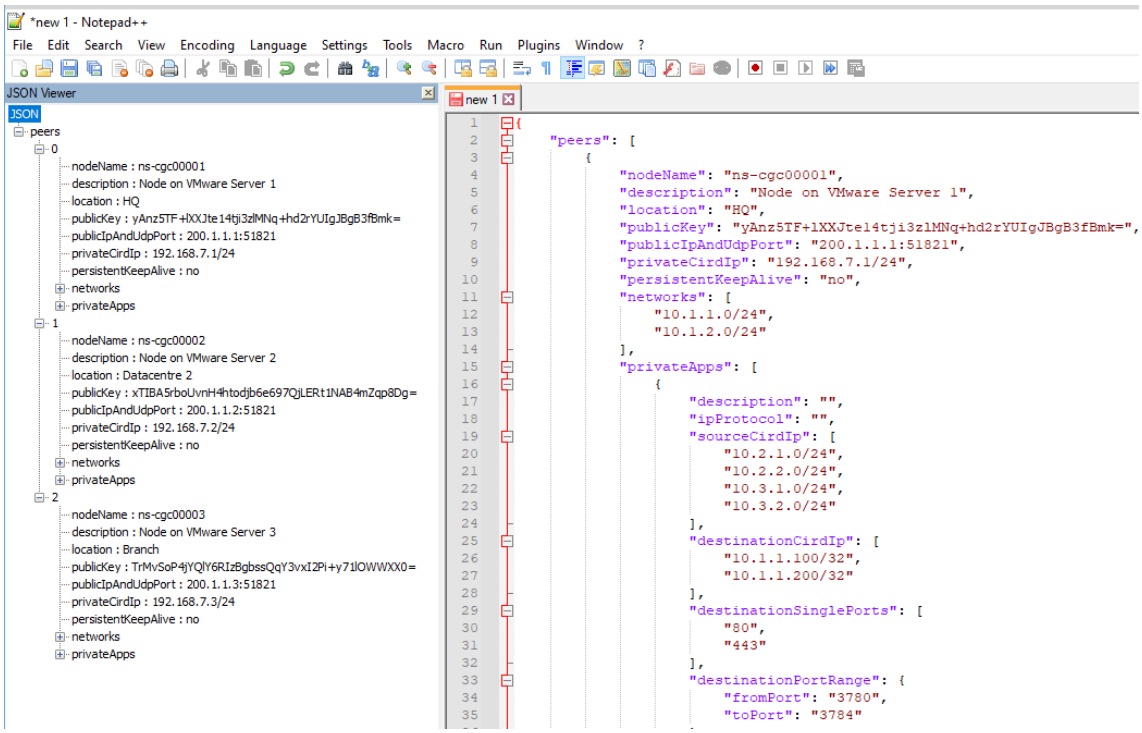
We strongly recommend using Software that can show errors on your JSON file and also can format (beautify) the file for better visibility. Below two examples.

12.4.1 Visual Code

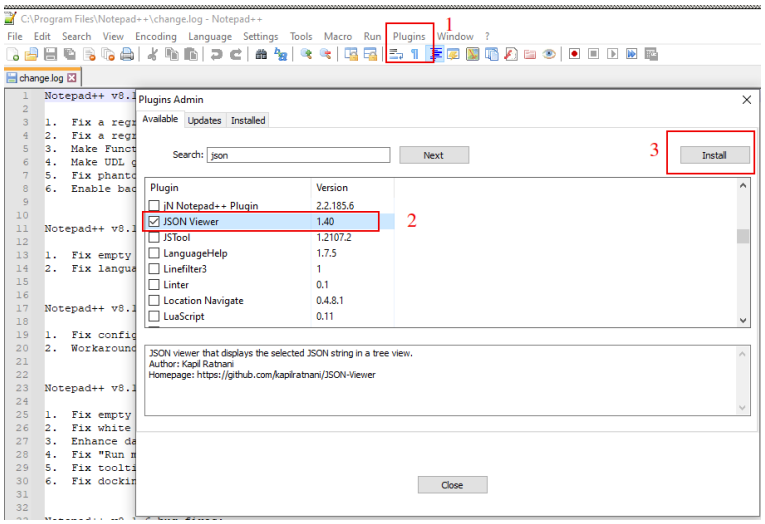


1. Download : https://code.visualstudio.com/download
2. Select your platform and install.
3. Create your JSON.
   - 3.1. Visual Code will show the errors in RED.
   - 3.2. To "Beautify" your JSON file press:
     - 3.2.1. On Windows: "Shift + Alt + F"
     - 3.2.2. On MAC: "Shift + Option + F"
     - 3.2.3. On Linux: " Ctrl + Shift + I"

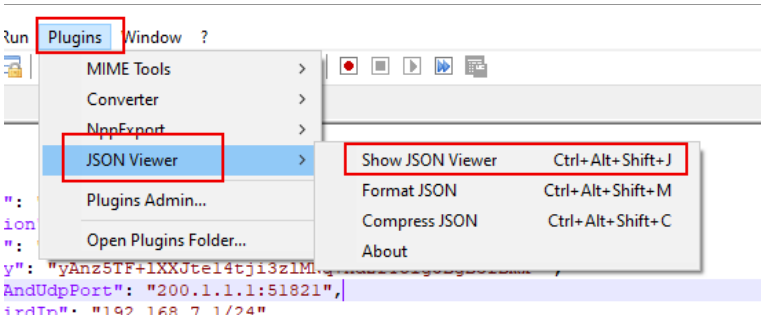
12.4.2 Notepad ++



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.

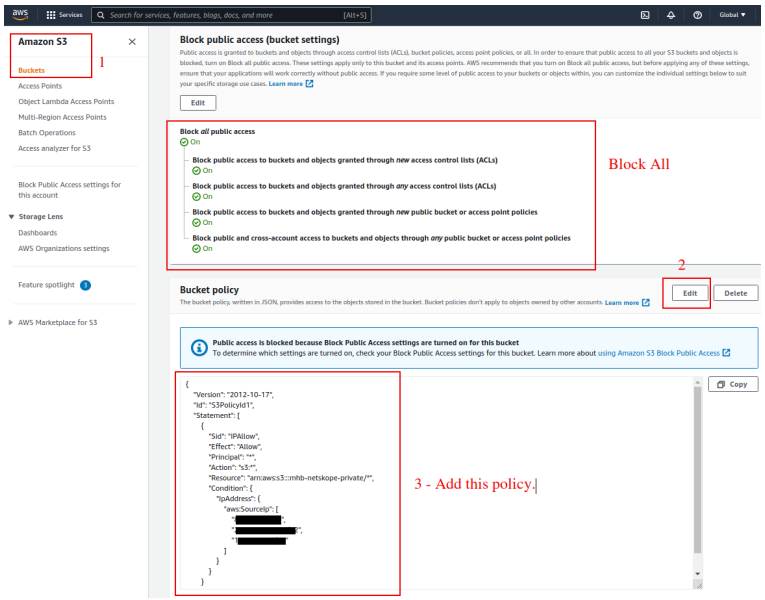


5. To format ("Beautify") your JSON go to: Plugins -> JSON Viewer -> Format JSON

12.5 Appendix E: 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!