Maidenhead Bridge

Cloud Security Connector for AWS

Enabling Zscaler (ZIA) for AWS customers

Administrator Guide

CSC Version 3.0

June 2021

Table of Contents

1 Introduction

The Cloud Security Connector (CSC) for AWS is an EC2 instance that connects internal AWS resources to Zscaler Internet Access (ZIA).

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

Simple to install, and not further management required.

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

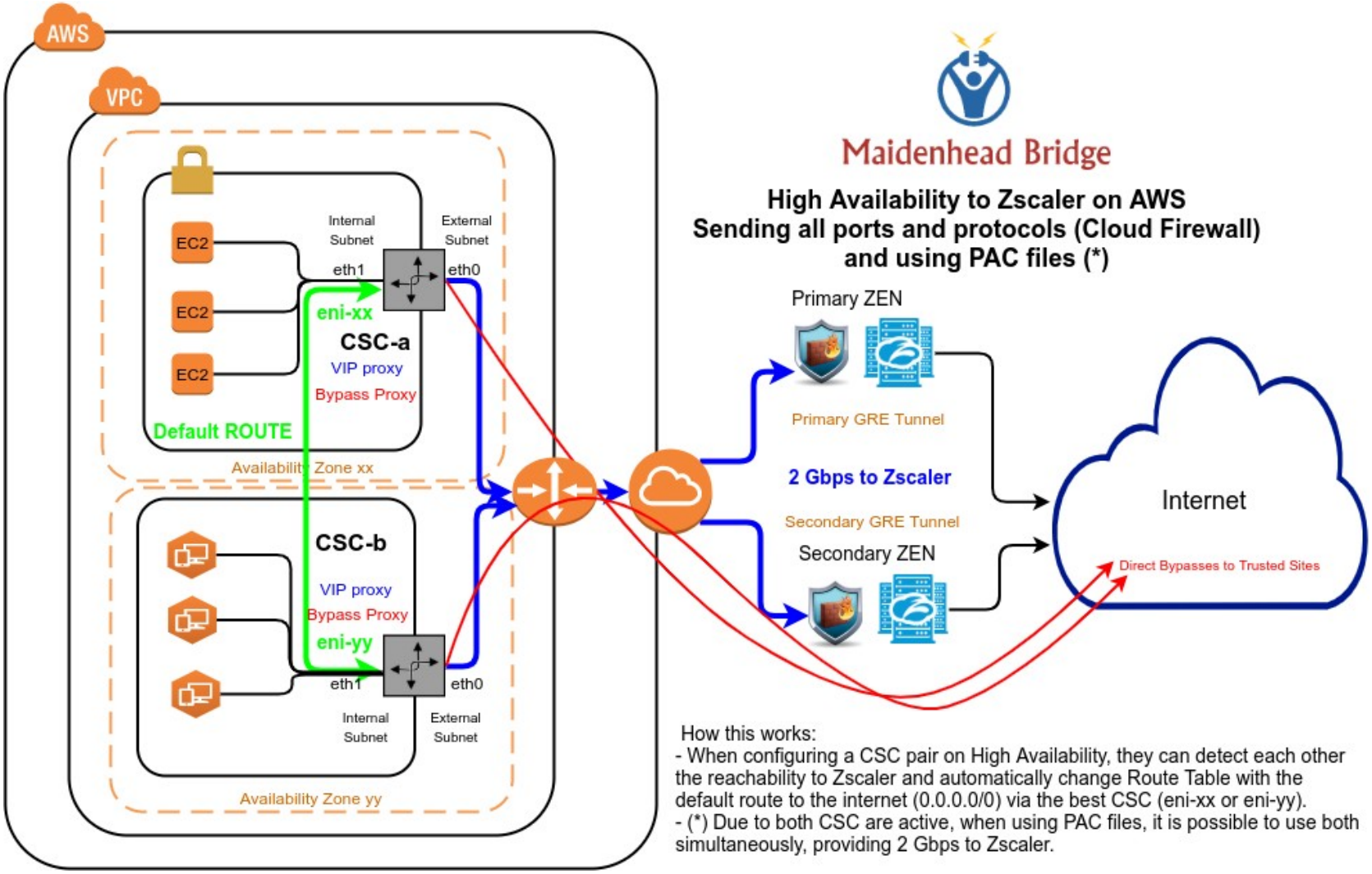
In addition to this, the CSC provides high availability changing the default route to Zscaler when configured as High Availability pair, and an easy way to manage direct bypasses to trusted sites using your public IP.

2 Key benefits of the Cloud Security Connector for AWS

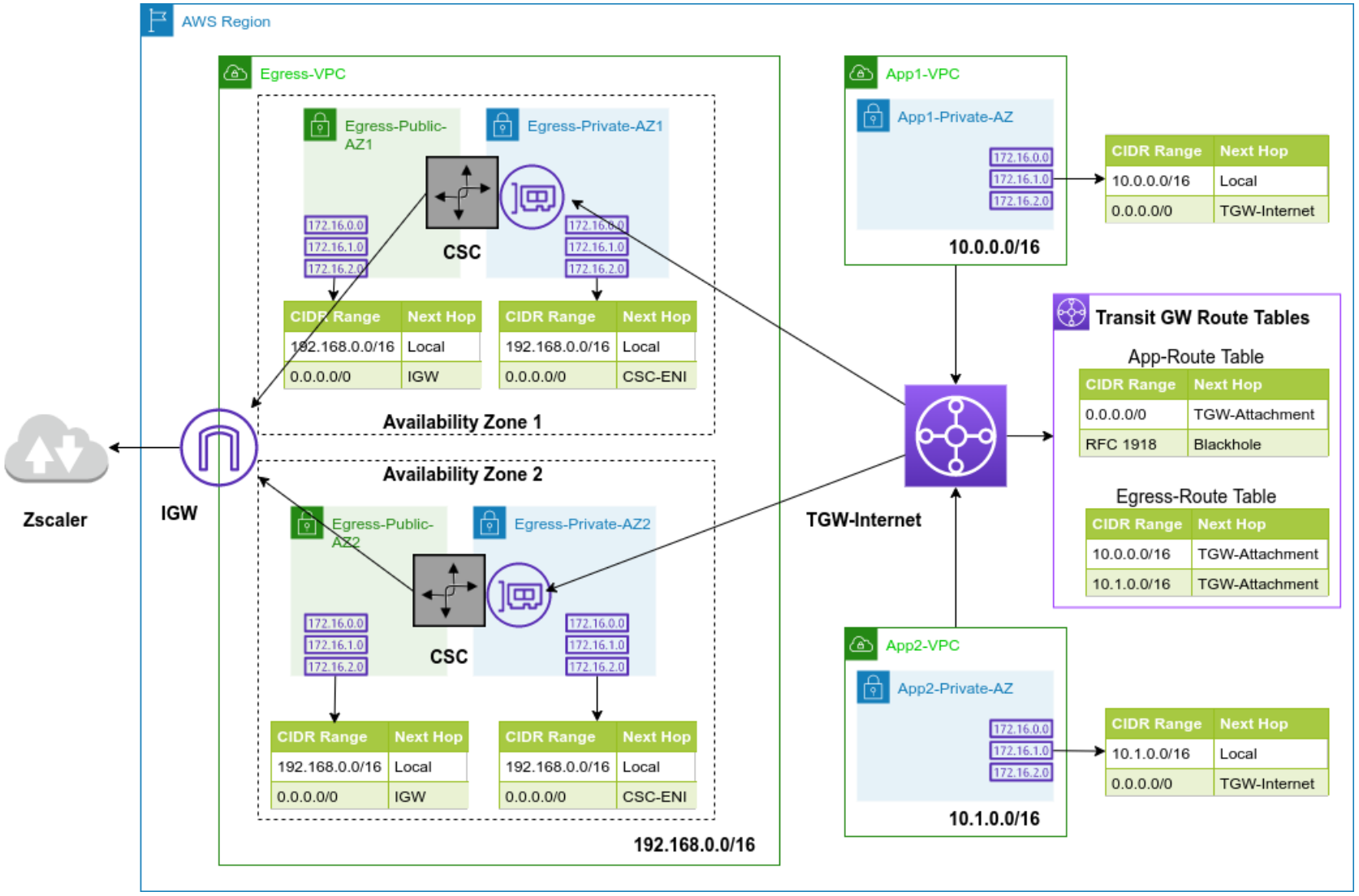
- Automated deployment using CloudFormation template and Zscaler API.
- Full tunnel redundancy.
- All parametrization required for AWS and Zscaler is already configured with the optimal values according to Zscaler Best practices.
- All Zscaler functionalities can be used: Firewall and Web Security.
- Complete visibility of internal IPs.
- Easy way to do Bypasses to trusted sites.
- No operational burden for Administrators.
- It runs on a cheap AWS instance: t2, t3a and t3 instances.
- Automatic default route selection on multiple Route Tables.

3 The CSC on the AWS architecture

3.1 High Availability Configuration - 2 Gbps - (See Appendix A for details)



3.2 Single Exit to Internet when using Transit Gateway.



4 Deploy the Cloud Security Connector (CSC)

There are two ways to deploy the CSC: **Basic and Advanced mode** (Zscaler API integration). The difference between these methods are:

- **Basic Mode:**
  - Deploy the CSC via Cloudformation, selecting External and Internal Subnet and no further parametrization (leave Cloudformation field "User Data" empty). The Cloudformation template will create all AWS resources (Instance, Security Groups, Public IPs, etc.), but the Tunnel and Bypass configuration are manual.
  - SSH the CSC and run Configuration Wizard to obtain the Public IP for the GRE tunnel.
  - Create the Location on the Zscaler console manually using the Public IP shown on the Configuration Wizard.
- **Advanced Mode (Zscaler API integration):**
  - Deploy the CSC via Cloudformation, selecting External and Internal Subnet and paste the userConfigData.json file on the "User Data" field. The following resources will be created and configured:
    - All AWS resources: Instance, Security Groups, Public IPs. etc.
    - Zscaler resources: Static IP, GRE credentials, selection of nodes Primary/Secondary and Location.
    - Bypass Proxy and Routed Bypass.
    - AWS SSM registration.

Note: With Advanced Mode, all AWS and Zscaler resources are created in one shot, and there is no requirement to SSH the CSC. You can manage the CSC using AWS Systems Manager via AWS SSM agent.

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.

4.1.2 Prerequisites EXAMPLE:

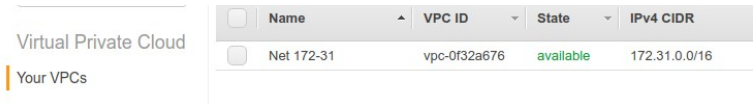
Following an EXAMPLE of prerequisites and how to obtain it.

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

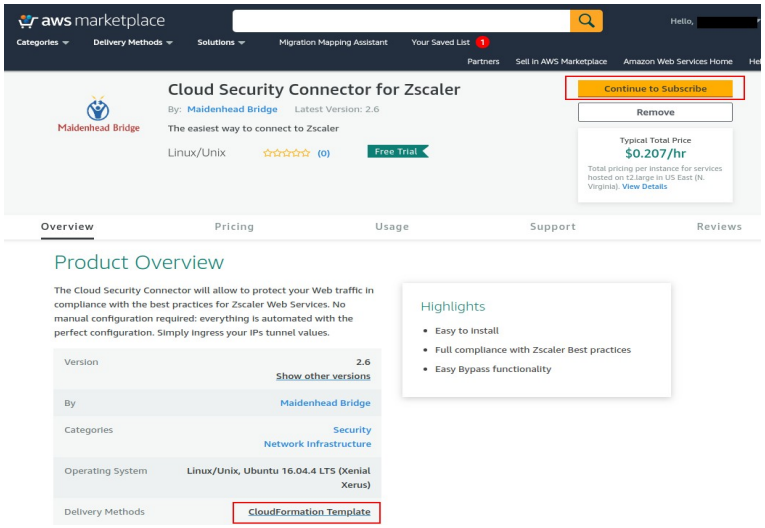


3 – External Subnet: subnet-818c0ddb (Note: Availability Zone us-east-1d and VPC ID vpc-of32a676)

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

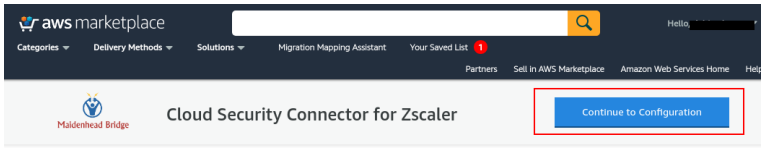
1. Go to the Cloud Security Connector for Zscaler product page at the AWS Market:



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

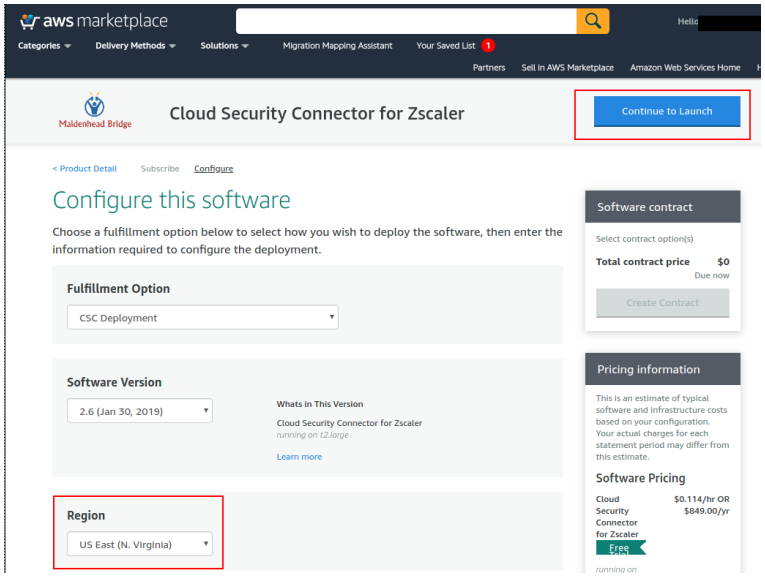
→ Click ***"Continue to Subscribe"***

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



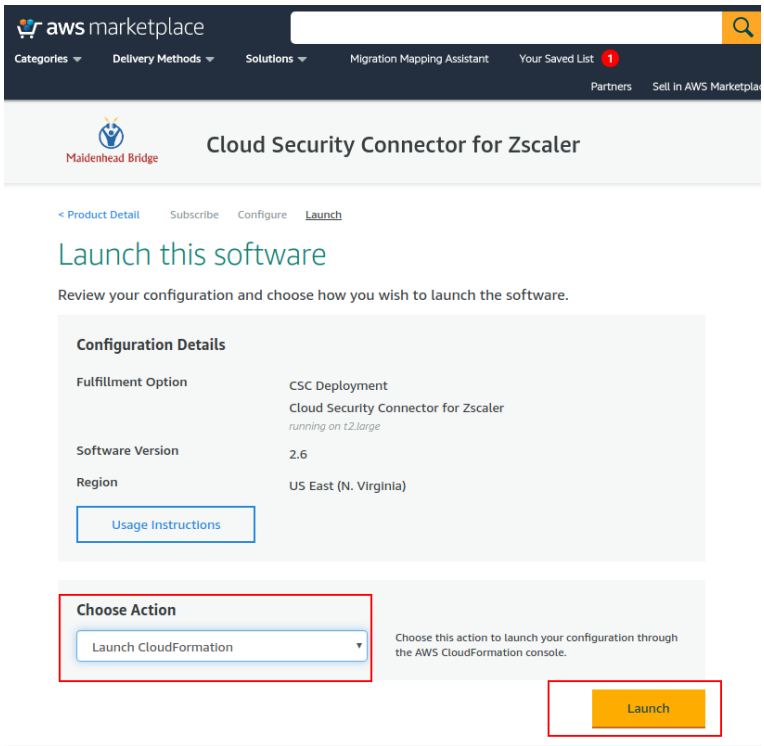
→ Click ***"Continue to Configuration"***

3. Select "Region"



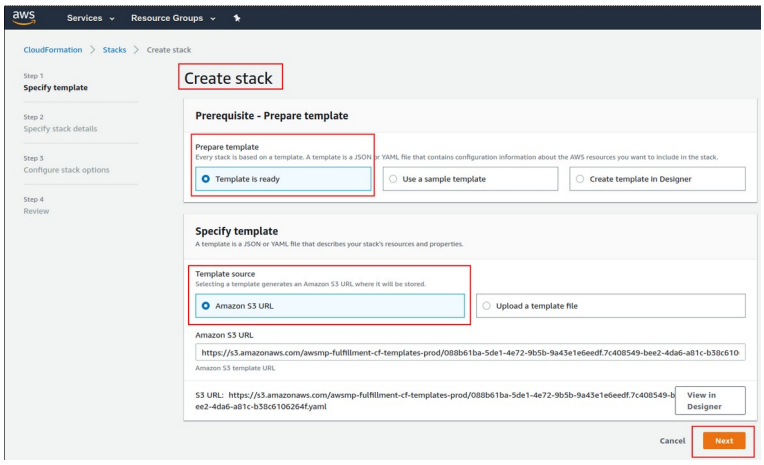
→ Click ***"Continue to Launch"***

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

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

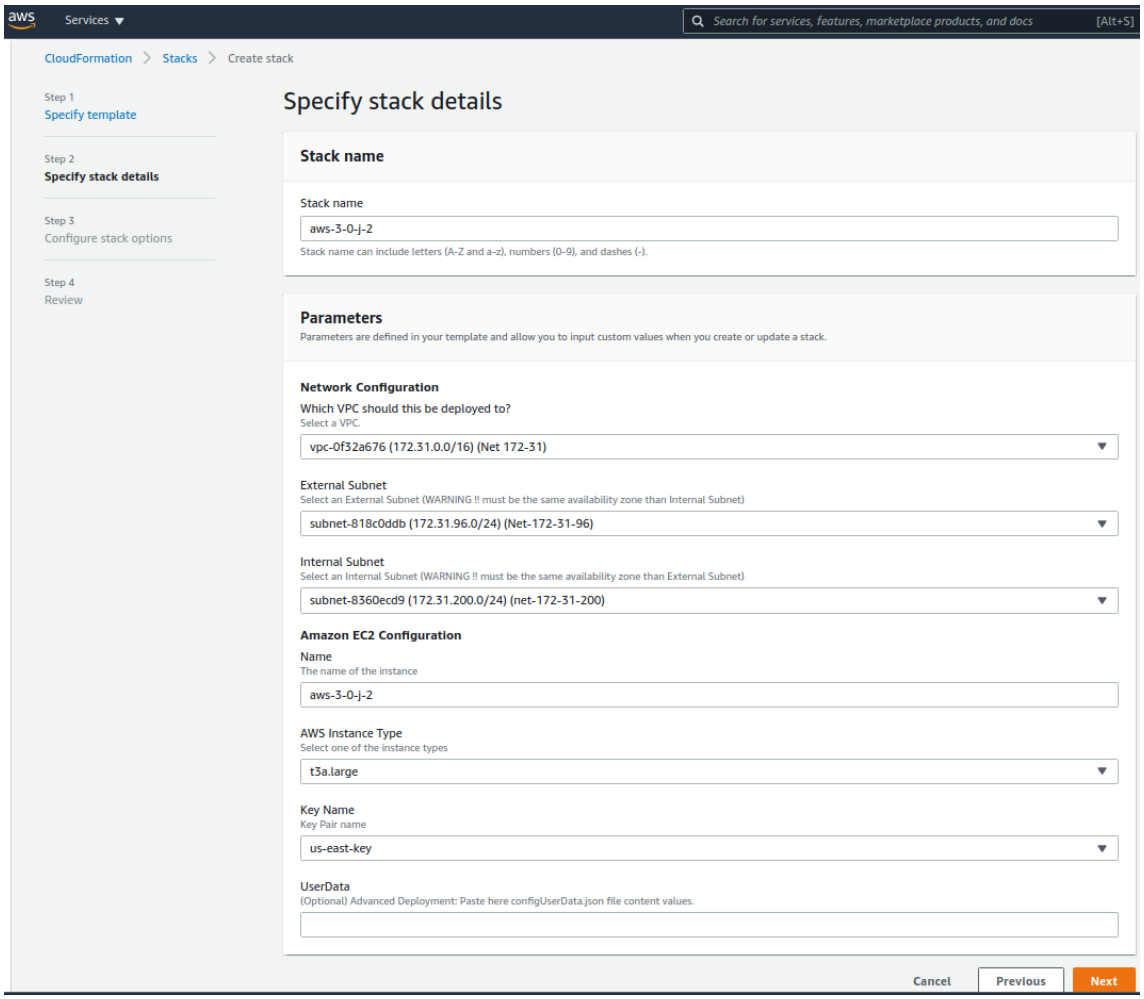
https://aws.amazon.com/ec2/instance-types/ and https://aws.amazon.com/ec2/pricing/on-demand/

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)

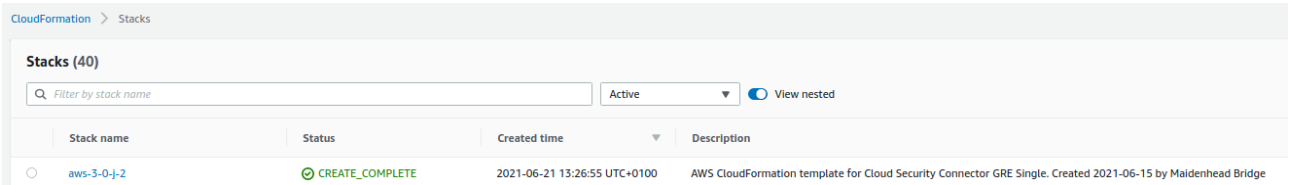


→ Click ***"Next"***

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

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

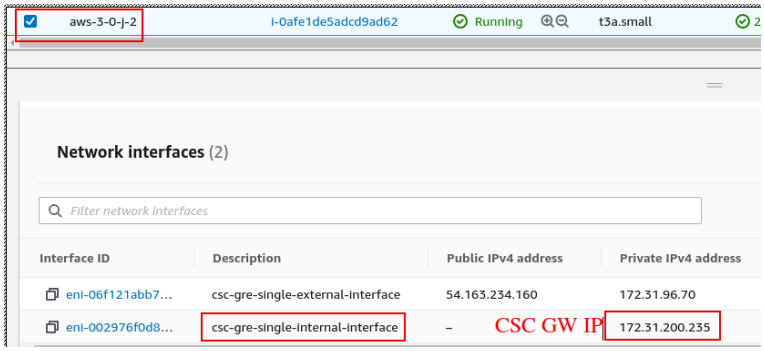
The Stack will show "status" CREATE_IN_PROGRESS, and after a while:



Done! Your CSC is deployed.

4.1.4 Accessing for first time to your CSC

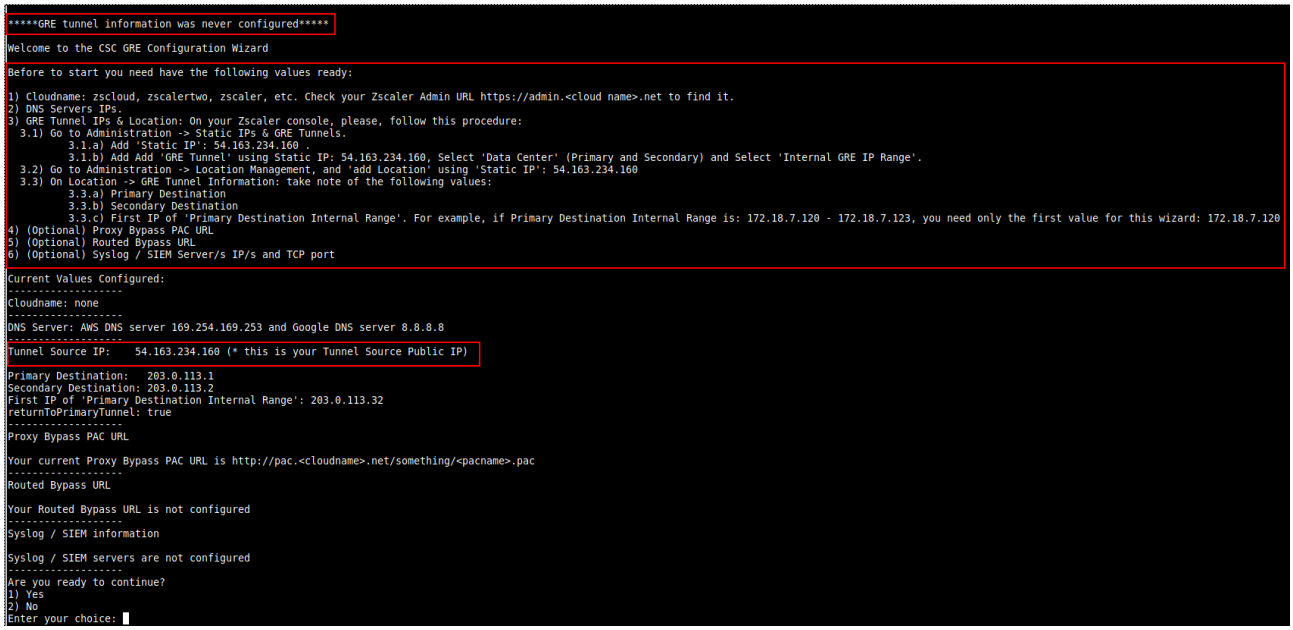
1. Go to your EC2 Dashboard → Instances and select the CSC created. Go to "Networking" and scroll down.



2. Find the "csc-gre-single-internal-interface" and take a look at the first Private IP address (CSC GW IP). This example is: 172.31.200.235
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 172.31.200.235



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".

Note: *If the create "Static IP" & "GRE Tunnel" menus are not available on your console, please, submit a ticket to Zscaler asking for the values. When you received the answer, please create the 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 172.31.200.235. Doing and SSH from a machine on subnet 172.31.200.0/24 to the CSC, the initial wizard appear.

In this example:

Key Name: us-east-key.pem

Username: cscadmin (use always "cscadmin")

CSC IP: 172.31.200.91

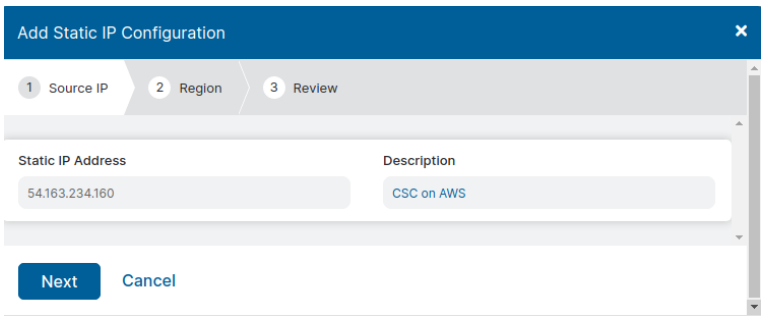
$ ssh -i *us-east-key.pem cscadmin@172.31.200.235*

```
*****GRE tunnel information was never configured*****

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) DNS Servers IPs.
3) GRE Tunnel IPs & Location: On your Zscaler console, please, follow this procedure:
  3.1) Go to Administration -> Static IPs & GRE Tunnels.
         3.1.a) Add 'Static IP': 54.163.234.160 .
         3.1.b) Add Add 'GRE Tunnel' using Static IP: 54.163.234.160, Select 'Data Center' (Primary and Secondary) and Select 'Internal GRE IP Range'.
  3.2) Go to Administration -> Location Management, and 'add Location' using 'Static IP': 54.163.234.160
  3.3) On Location -> GRE Tunnel Information: take note of the following values:
         3.3.a) Primary Destination
         3.3.b) Secondary Destination
         3.3.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
4) (Optional) Proxy Bypass PAC URL
5) (Optional) Routed Bypass URL
6) (Optional) Syslog / SIEM Server/s IP/s and TCP port

Current Values Configured:
------------------
Cloudname: none
------------------
DNS Server: AWS DNS server 169.254.169.253 and Google DNS server 8.8.8.8
------------------
Tunnel Source IP:    54.163.234.160 (* this is your Tunnel Source Public IP)

Primary Destination:   203.0.113.1
Secondary Destination: 203.0.113.2
First IP of 'Primary Destination Internal Range': 203.0.113.32
returnToPrimaryTunnel: true
------------------
Proxy Bypass PAC URL

Your current Proxy Bypass PAC URL is http://pac.<cloudname>.net/something/<pacname>.pac
------------------
Routed Bypass URL

Your Routed Bypass URL is not configured
------------------
Syslog / SIEM information

Syslog / SIEM servers are not configured
------------------
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: 54.163.234.160

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"

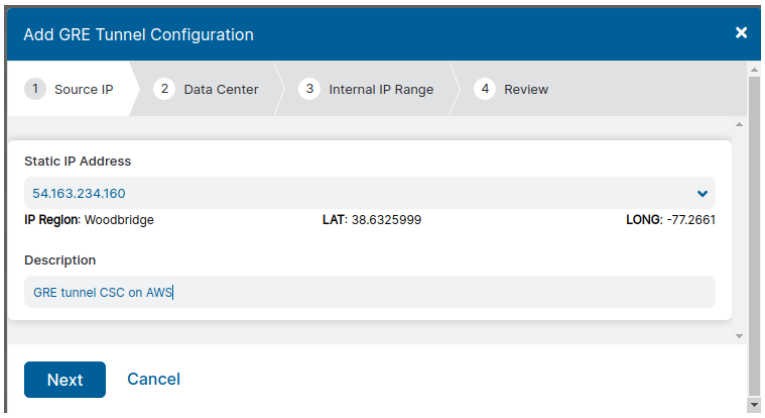


Click "Next", "Next" and "Save". Activate the changes.

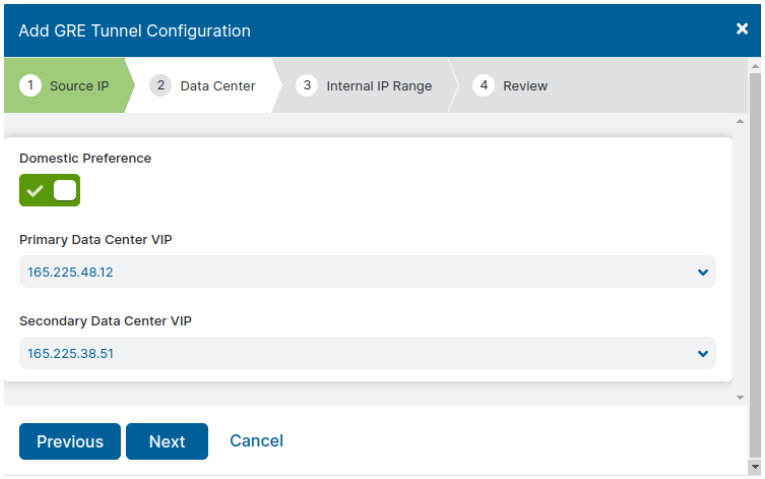
4.1.5.2.2 Create "GRE Tunnel"

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

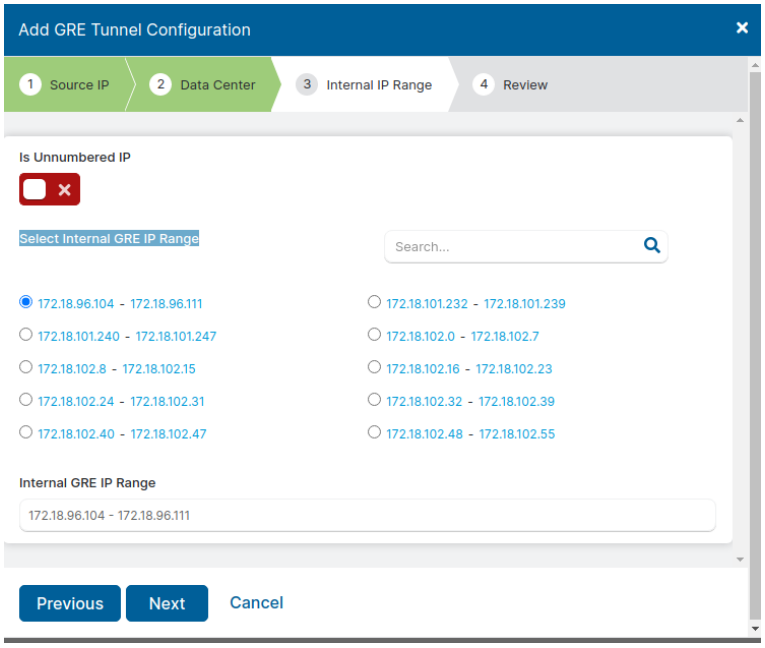
Select the "Static IP" created in the previous step.



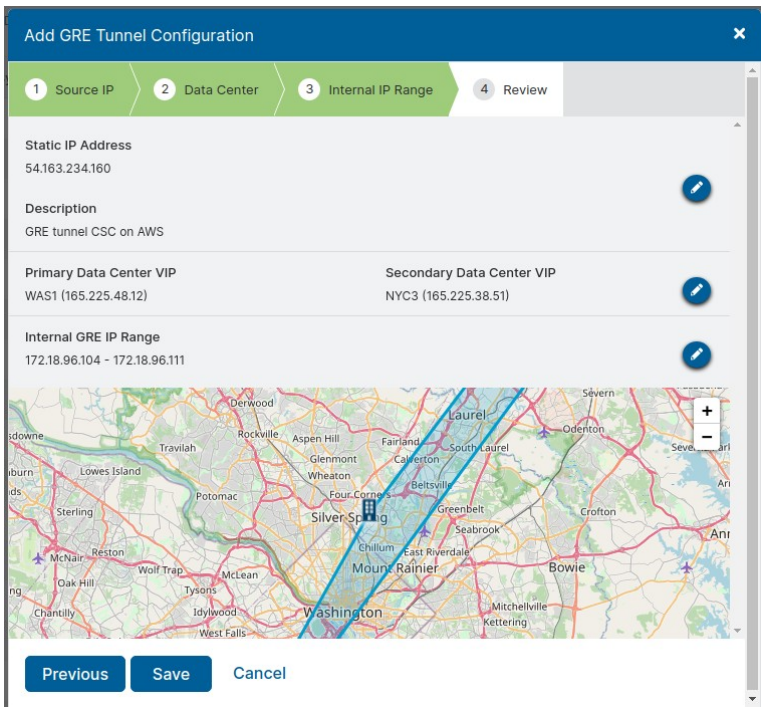
Click "Next" and select "Domestic Preference" if you prefer.



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



Click "Next"

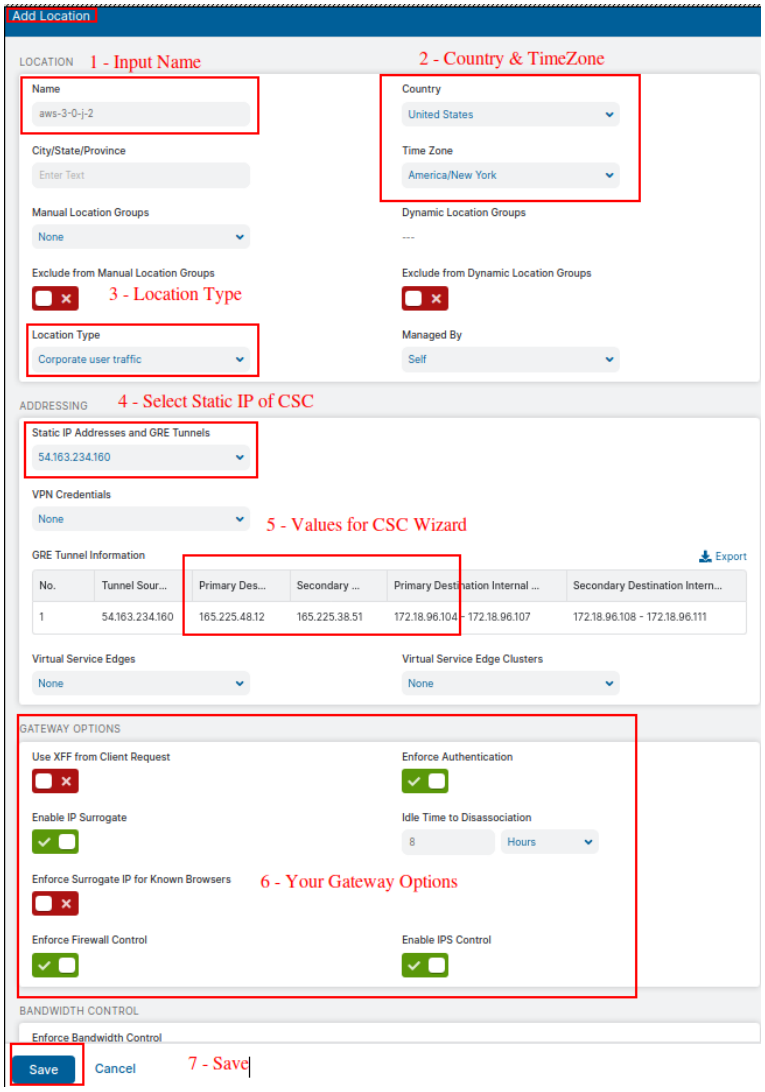


and "Save". Activate the Changes.

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) Not in the list? Ingress Manually
8) Quit
Enter your choice: 1
------------------
```

2. Enter your DNS Servers or use AWS DNS server & Google (8.8.8.8)

```
------------------
DNS Configuration

You are using AWS DNS server 169.254.169.253 and Google DNS server 8.8.8.8

Do you want to change the DNS servers?
1) Yes
2) No
Enter your choice: 2
------------------
```

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

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

Your current GRE tunnels configuration is:

Tunnel Source IP:    54.163.234.160

Primary Destination:    203.0.113.1
Secondary Destination:  203.0.113.2
First IP of 'Primary Destination Internal Range': 203.0.113.32
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): 165.225.48.12
Secondary Destination (IP): 165.225.38.51
First IP of 'Primary Destination Internal Range': 172.18.96.104
```

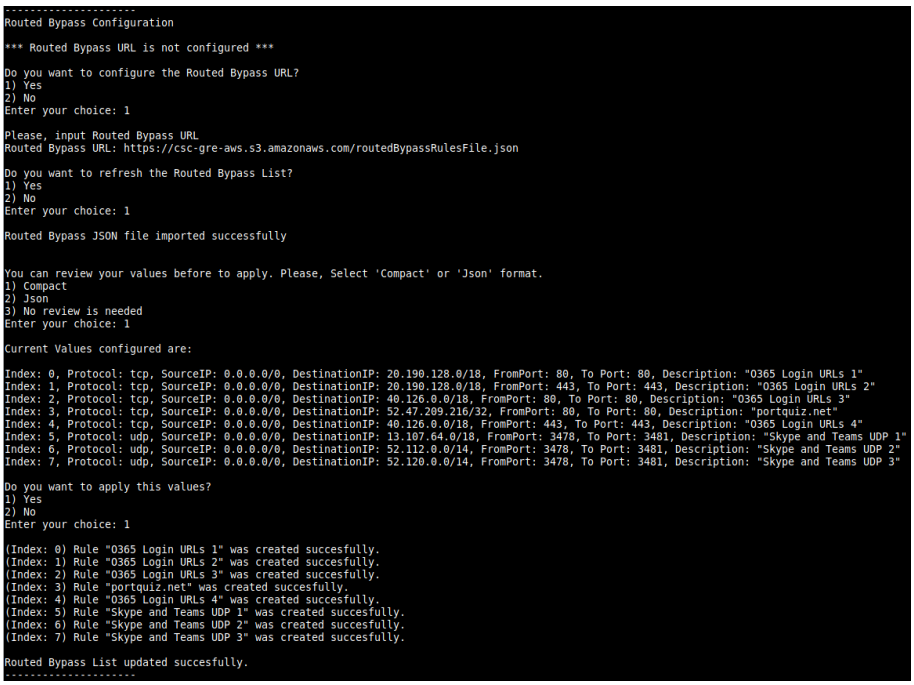
4. Select "ReturnToPrimary" true or false.

```
'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
--------------------
```

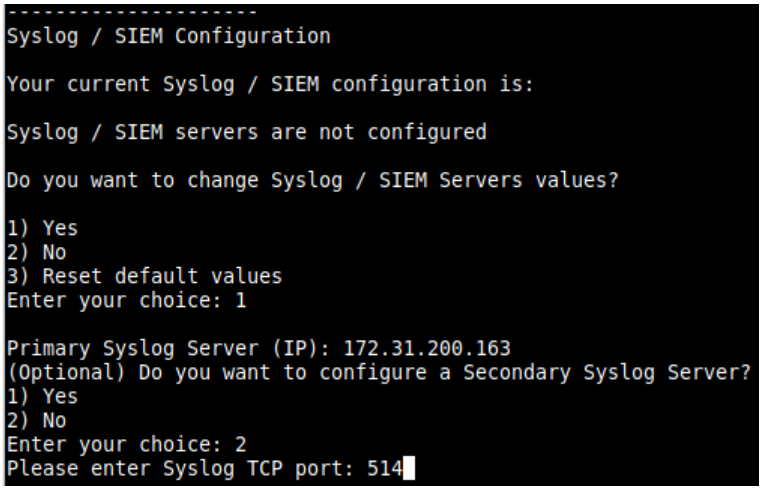
5. (Optional) Enter your Proxy Bypass PAC URL and refresh the Proxy Bypass List.

```
-------------------
Proxy Bypass Configuration

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

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

Please, input Proxy Bypass PAC URL
Proxy Bypass PAC URL:http://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac

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

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

This is your current Bypass Proxy List

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


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

Proxy Bypass List updated sucessfully
```

6. (Optional) Enter your Routed Bypass URL and refresh the Routed Bypass Rules.



7. (Optional) Enter your SIEM server IP and Logs.



8. Finally, check and confirm the values:

```
Please confirm these values:
------------------
Cloudname:  zscalerthree
------------------
DNS Server: AWS DNS server 169.254.169.253 and Google DNS server 8.8.8.8
------------------
GRE tunnels IP values:

Tunnel Source IP (IP):  54.163.234.160

Primary Destination:   165.225.48.12
Secondary Destination: 165.225.38.51
First IP of 'Primary Destination Internal Range': 172.18.96.104
returnToPrimaryTunnel: true
------------------
Proxy Bypass PAC URL
Your current Bypass PAC URL is http://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac
------------------
Routed Bypass URL
Your current Routed Bypass PAC URL is https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json
------------------
Primary Syslog / SIEM server IP: 172.31.200.163
Syslog / SIEM TCP port IP: 514
------------------
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.

4.2 Advanced Mode Deployment (using Zscaler API)

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

4.2.1 Prerequisites

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

userDataConfig.json

```
{
   "model": "csc-gre-aws",
   "version": "1.0",
   "cloudName": "",
   "apiTokenID": "",
   "dns": {
      "useCloudDNS": true,
      "primaryDnsIP": "",
      "secondaryDnsIP": ""
   },
   "bypassProxyPacUrl": "",
   "syslogServers": {
      "primarySyslogIP": "",
      "secondarySyslogIP": "",
      "syslogTcpPort": 514
   },
   "ssmAgent": {
      "activationCode": "",
      "activationID": "",
      "awsRegion": ""
   },
   "tunnelRedundancy": {
      "returnToPrimaryTunnel": true
   },
   "nodeSelection": {
      "withinCountryPreferred": true
   },
   "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
   },
   "routedBypassJsonFileUrl": "",
   "routedBypassRules": []
}
```

4.2.1.1 userDataConfig.json file fields and values

We recommend to use Visual Studio Code to validate the integrity of json files.

4.2.1.1.1 Fixed values - do not change

```
"model": "csc-gre-aws",
"version": "1.0",
```

4.2.1.1.2 cloudName

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

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

4.2.1.1.3 apiTokenID

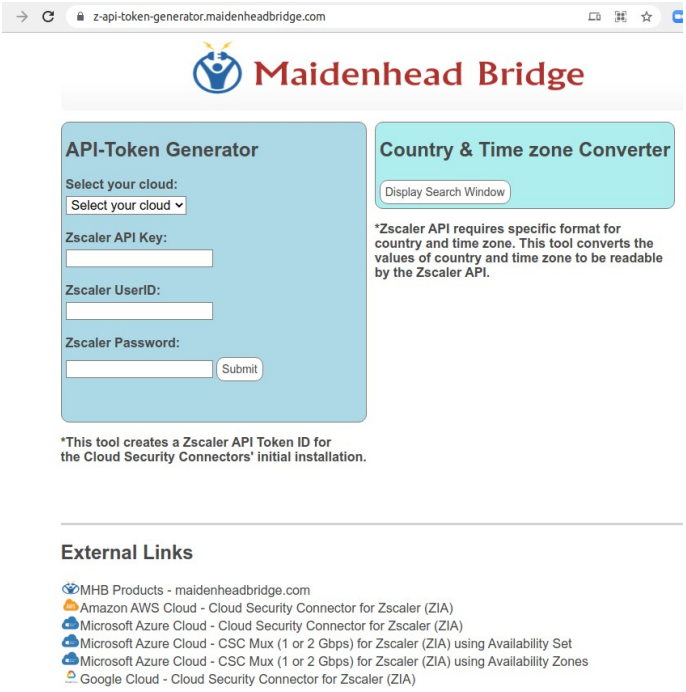
```
"apiTokenID": "",
```

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".

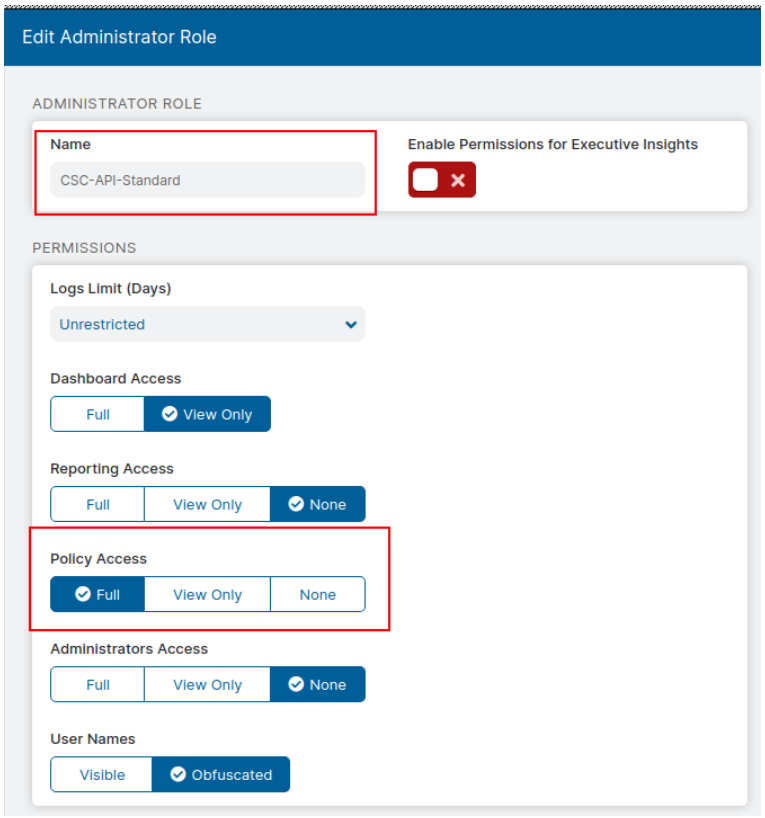
Alternatively, you can use, at your own risk, a utility page we created for this purpose. The page is here: https://z-api-token-generator.maidenheadbridge.com/

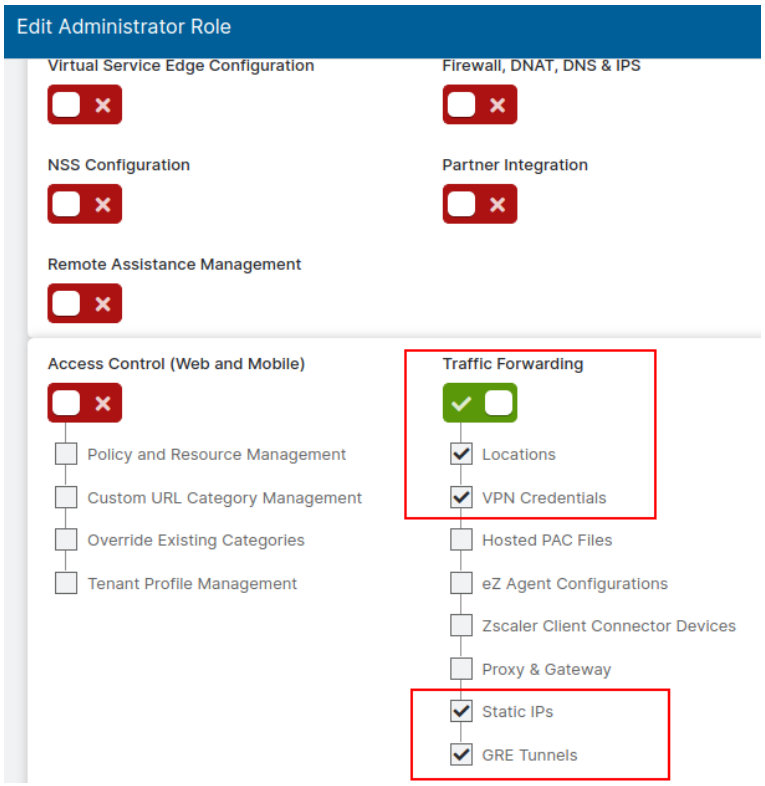


If you want to use the page: https://z-api-token-generator.maidenheadbridge.com/ to generate the "API Token", you have two options to create the values of the Zscaler API Key, Username and Password on the Zscaler console. Option one is to use the "Organization API Key", and option two is to use any unused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

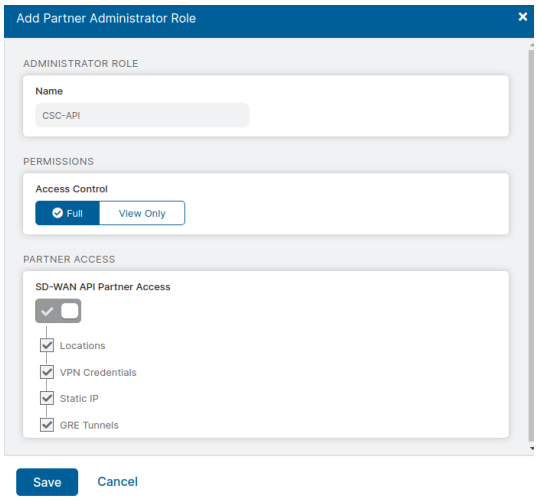
3. Create an Administrator User and apply the Role. Go to Administration → Administrator Management → Add Administrator. The Administrator User will look like this:

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

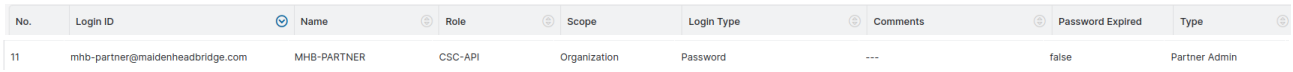
Done! Now you are the values requested: Zscaler API Key, Zscaler User ID / Password.

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.

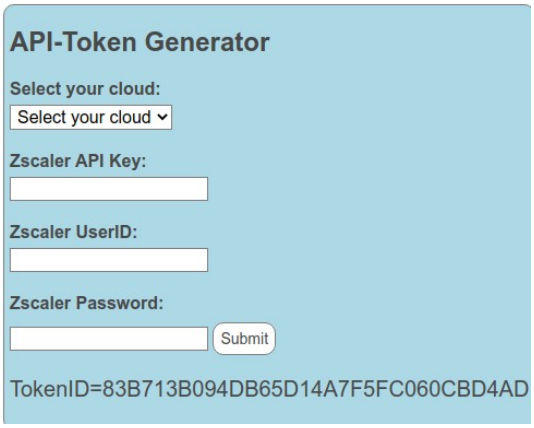


3. Create a Partner Administrator User and apply the Role. Go to Administration → Administrator Management → Add Partner Administrator. The Partner Administrator User will look like this:



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”.

4.2.1.1.4 dns

```
"dns": {
    "useCloudDNS": true,
    "primaryDnsIP": "",
    "secondaryDnsIP": ""
},
```

Select “useCloudDNS” : true, if you want to use the AWS DNS 169.254.169.253 and Google 8.8.8.8 servers.
Select “useCloudDNS” : false, if you want to use your own dns servers. In this case, you must fill the values of "primaryDnsIP" and "secondaryDnsIP".

4.2.1.1.5 bypassProxyPacUrl

```
"bypassProxyPacUrl": "",
```

Insert here you Proxy Bypass PAC URL. The PAC URL contains the list of Proxy Bypasses to implement. For example: http://pac.zscalerthree.net/maidenheadbridge.com/bypass.pac

4.2.1.1.6 syslogServers

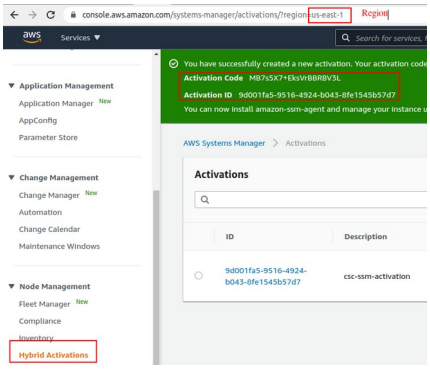
```
"syslogServers": {
   "primarySyslogIP": "",
   "secondarySyslogIP": "",
   "syslogTcpPort": 514
},
```

Input the IP value of your Primary Syslog Server and the TCP port on “syslogTcpPort”. Secondary Syslog IP is optional.
If you don’t want to configure Syslog Servers, leave this fields blank.

4.2.1.1.7 ssmAgent

```
"ssmAgent": {
   "activationCode": "",
   "activationID": "",
   "awsRegion": ""
},
```

The CSC can be managed using AWS Systems Manager.
Input here the values of Activation Code, Activation ID and AWS Region of the “Hybrid Activation” of the AWS Systems Manager.



4.2.1.1.8 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)

4.2.1.1.9 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.

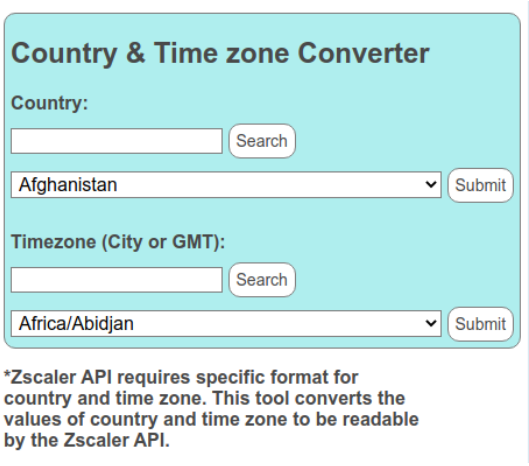
4.2.1.1.10 location

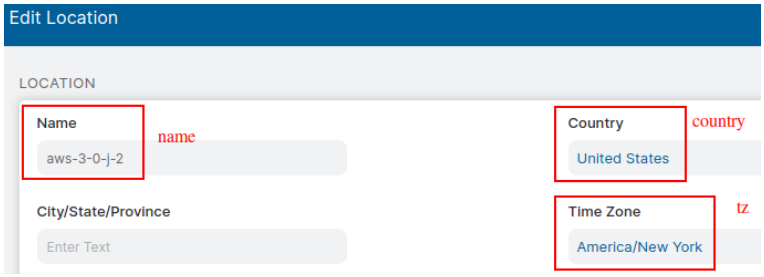
```
"location": {
    "name": "",
    "country": "",
    "tz": "",
    "ipAddresses": [
        "auto"
    ],
    "authRequired": true,
    "xffForwardEnabled": false,
    "surrogateIP": true,
    "idleTimeInMinutes": 480,
    "displayTimeUnit": "MINUTE",
    "surrogateIPEnforcedForKnownBrowsers": false,
    "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.*

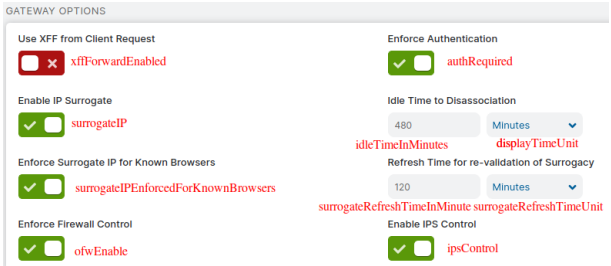




For example, if you want to use the values shown in the image above, you need to configure:

```
"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."



4.2.1.1.11 routedBypassJsonFileUrl

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

```
"routedBypassJsonFileUrl": "https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json",
"routedBypassRules": []
```

4.2.1.1.12 routedBypassRules

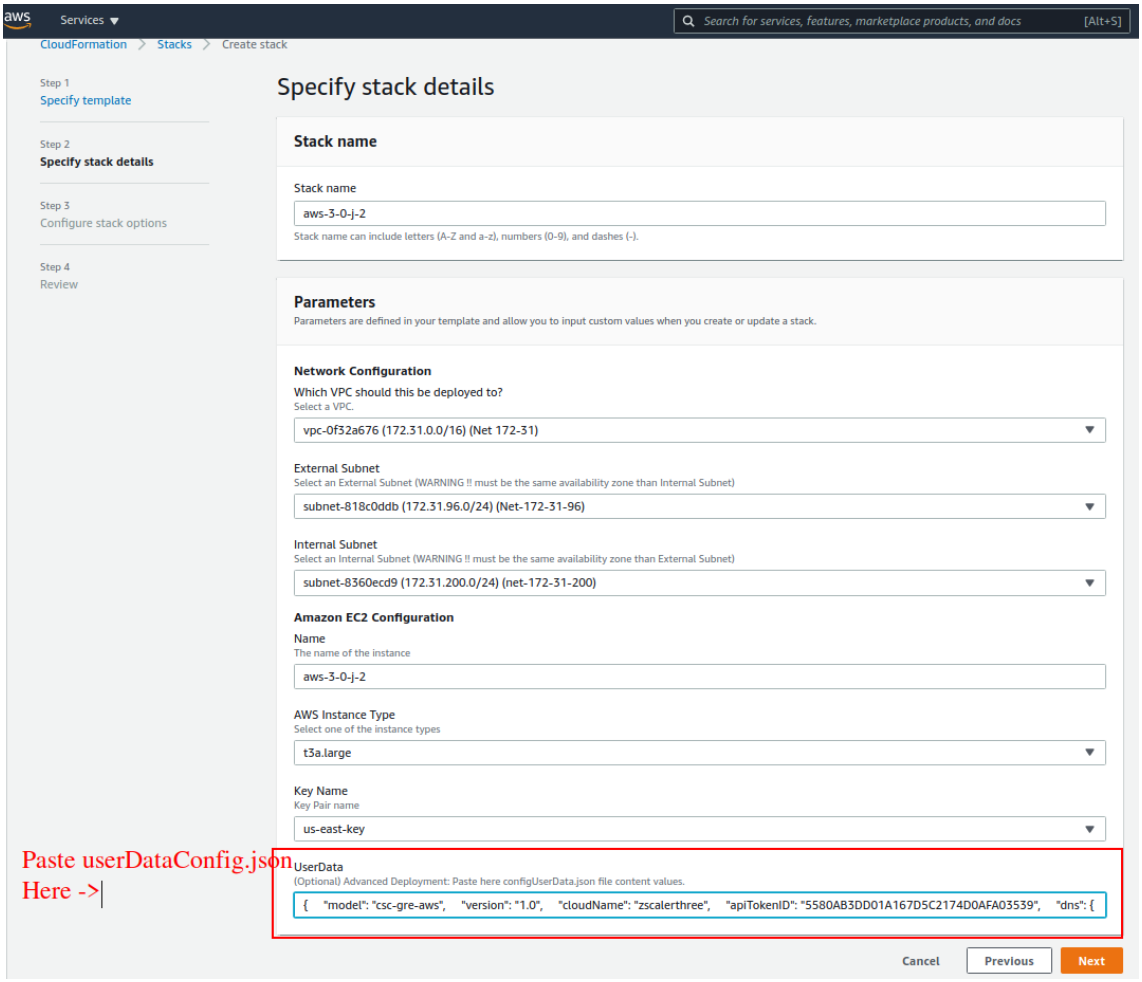
In the case you don't want to use the Routed Bypass URL method, you can include here your Routed Rules directly. For example:

```
"routedBypassJsonFileUrl": "",
"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"
  }
]
```

4.2.2 Advanced Mode Deployment using CloudFormation

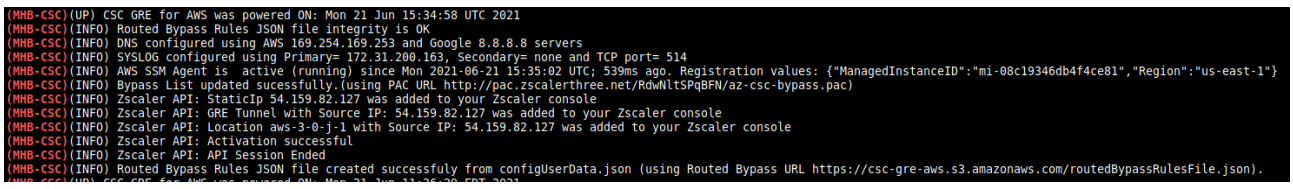
The only difference between Basic and Advanced Deployment is filling the last section of the CloudFormation template. (UserData).

Simple copy the contents of the userDataConfig.json file and paste it on 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.

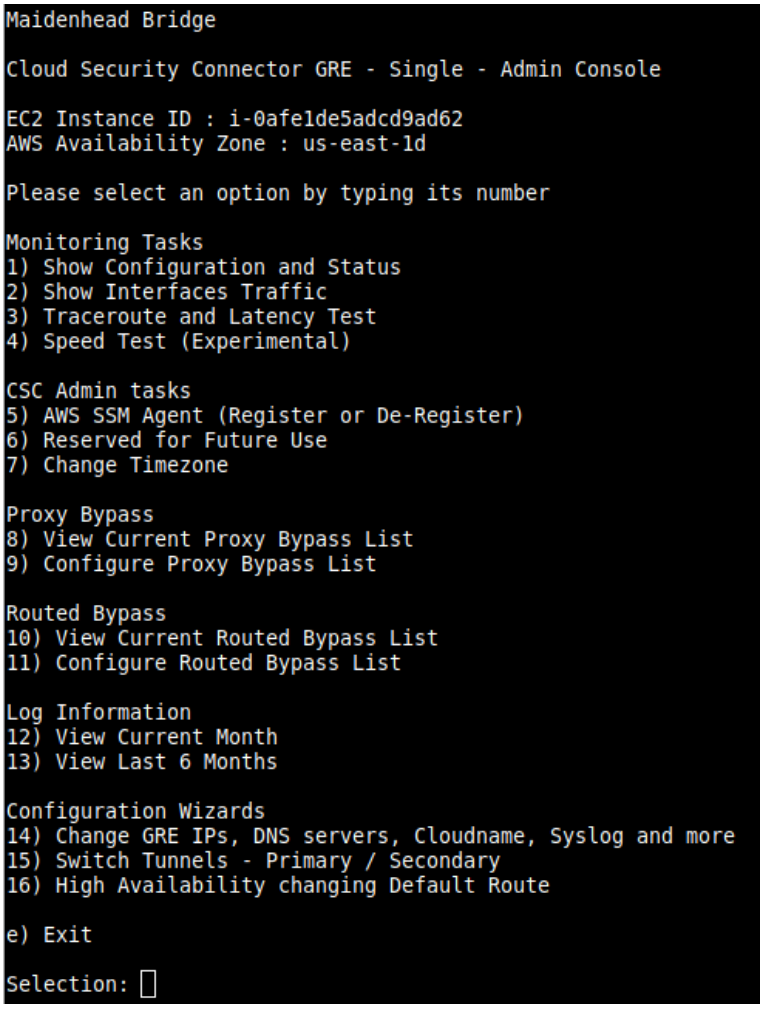


5 The Cloud Security Connector Admin Console:

The CSC SSH Console was created to simplify admin tasks showing what is essential to administrators for operation and troubleshooting. In addition to this, using AWS System Manager, you can do all monitoring tasks via AWS Console. Register the CSC instance on AWS as a managed instance, and you are ready to control the CSC using all AWS System Manager tools.

When accessing the console via SSH, you will receive the Admin Console. For example:

```
ssh -i us-east-key.pem cscadmin@172.31.200.235
```



The Main Sections are:

- **Monitoring Tasks:** To check statuses, real-time traffic, speed, etc.
- **CSC Admin Tasks:** To register the CSC for AWS management, change password and timezone.
- **Proxy Bypass:** To manage the Proxy Bypass PAC URL or to enter the Proxy Bypasses manually.

- **Routed Bypass:** To manage the Routed Bypass URL or to enter the Routed Bypasses manually.
- **Log Information:** Shows activity logs.
- **Configuration Wizards:** To rerun the initial wizard, switch tunnels and configuring HA.

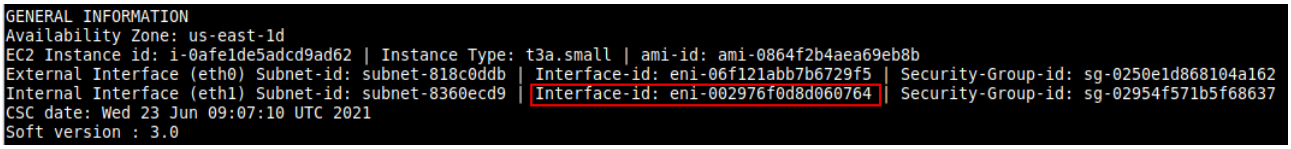
5.1 Monitoring Tasks

5.1.1 Show Configuration and Status

```
Selection: 1

GENERAL INFORMATION
Availability Zone: us-east-1d
EC2 Instance id: i-0afe1de5adcd9ad62 | Instance Type: t3a.small | ami-id: ami-0864f2b4aea69eb8b
External Interface (eth0) Subnet-id: subnet-818c0ddb | Interface-id: eni-06f121abb7b6729f5 | Security-Group-id: sg-0250e1d868104a162
Internal Interface (eth1) Subnet-id: subnet-8360ecd9 | Interface-id: eni-002976f0d8d060764 | Security-Group-id: sg-02954f571b5f68637
CSC date: Wed 23 Jun 08:55:42 UTC 2021
Soft version : 3.0

INTERFACES INFORMATION
External: Tunnel IP (eth0): 172.31.96.70/24 | Bypass Proxy Egress IP: 172.31.96.114 | Network Gateway: 172.31.96.1 is Alive
Internal: CSC GW IP (eth1): 172.31.200.235/24 | Network Gateway: 172.31.200.1 is Alive

TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 172.31.200.87:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 172.31.200.66:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP

ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 54.163.234.160
Bypass Proxy Public IP: 18.204.121.250

DNS INFORMATION
DNS Server (1) AWS DNS IP: 169.254.169.253
DNS Server (2) Google DNS IP: 8.8.8.8

ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
GRE tunnels egress Public IP: 54.163.234.160
Primary Tunnel:
               ZEN Public IP: 165.225.48.12
               Tunnel IPs (local/zen): 172.18.96.105 / 172.18.96.106
Secondary Tunnel:
               ZEN Public IP: 165.225.38.51
               Tunnel IPs (local/zen): 172.18.96.109 / 172.18.96.110

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 22 Jun 15:22:36 UTC 2021

HTTP://IP.ZSCALER.COM PAGE STATUS
You are accessing the Internet via Zscaler Cloud: Washington DC in the zscalerthree.net cloud.
Your Gateway IP Address is 54.163.234.160

BYPASS PROXY - EGRESS INTERFACE STATUS
Bypass Proxy Egress Interface 172.31.96.114 can reach test page (http://pac.zscalerthree.net)

ROUTED BYPASS
Using Routed Bypass URL: https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json
Routed Bypass URL https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json is reachable
Routed Bypass Rules configured via URL: 8

AWS SSM AGENT
AWS SSM Agent is active (running) since Wed 2021-06-23 08:54:48 UTC; 54s ago
Registration values: {"ManagedInstanceID":"mi-0904b06fee9ef0c74","Region":"us-east-1"}

SYSLOG/SIEM Servers Information
Primary Syslog IP: 172.31.200.163
Secondary Syslog IP: Not configured
Syslog TCP port: 514

HIGH AVAILABILITY Information
The HA service is:  active (running) since Wed 2021-06-23 08:48:49 UTC; 6min 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-002976f0d8d060764 (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-08f80bd2952ff13ed
SNS message ARN= arn:aws:sns:us-east-1:544690173127:AWS-Zscaler-Notification
```

5.1.1.1 GENERAL INFORMATION

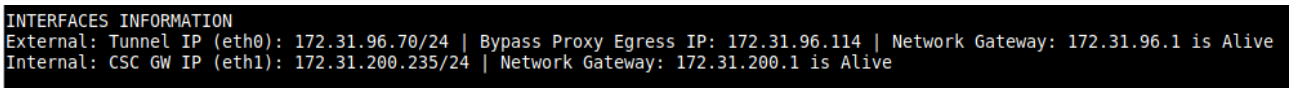
This section contains general information about the instance:



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

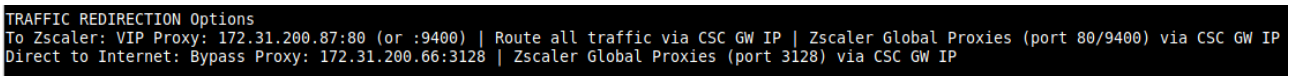
5.1.1.2 INTERFACES INFORMATION

This section contains the interfaces information:



5.1.1.3 TRAFFIC REDIRECTION Options

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



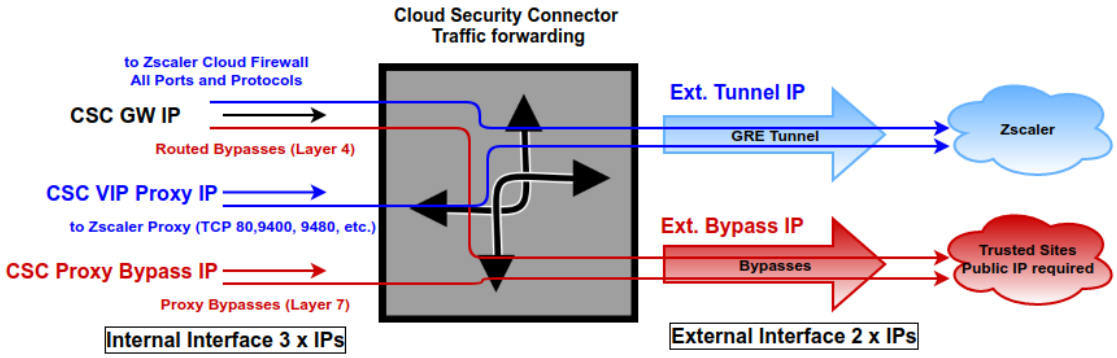
The objective of the Cloud Security Connectors of Maidenhead Bridge is to provide a simple architecture, 100% proven that works, to connect to Zscaler.

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

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

***(*)** On AWS routing tables, the value to use as a GW is the “Interface-id:” (eni-xxyyzz)*

Here an illustration about this:



Important: Please, see Appendix A for detailed information about traffic redirection (with examples)

5.1.1.4 ELASTIC (PUBLIC) IPs INFORMATION

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

In our example:

```
ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 54.163.234.160
Bypass Proxy Public IP: 18.204.121.250
```

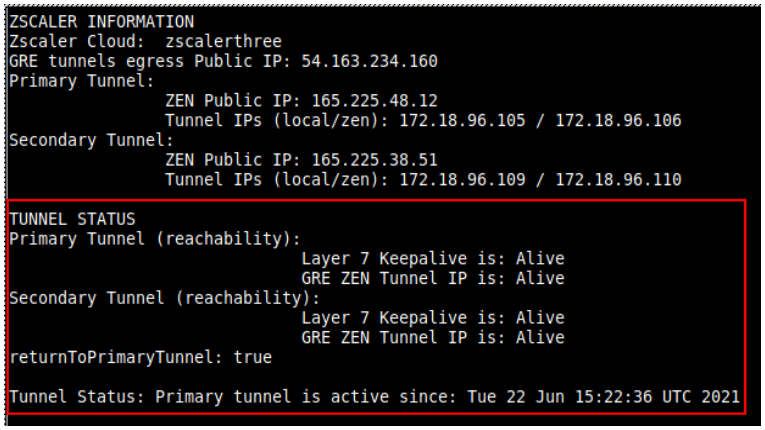
5.1.1.5 DNS INFORMATION

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

```
DNS INFORMATION
DNS Server (1) AWS DNS IP: 169.254.169.253
DNS Server (2) Google DNS IP: 8.8.8.8
```

5.1.1.6 ZSCALER INFORMATION

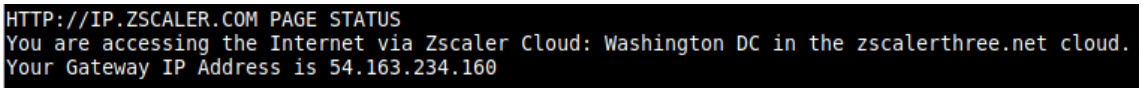
This section shows the GRE tunnel information and Tunnel Status.



Please, pay attention to the reachability of the Keepalives and Tunnels and the Tunnel Status.

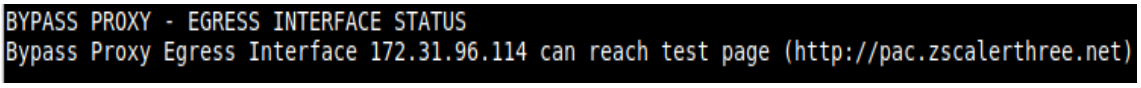
5.1.1.7 HTTP://IP.ZSCALER.COM PAGE STATUS

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



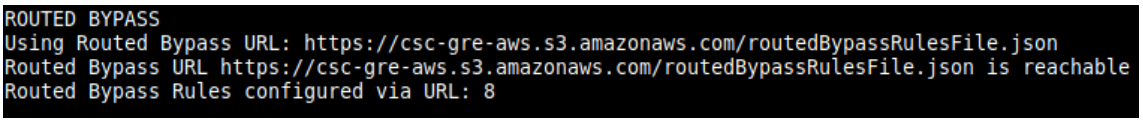
5.1.1.8 BYPASS PROXY – EGRESS INTERFACE STATUS

This sections validates if the Bypass Proxy can access internet directly going to http://pac.<cloudname>.net



5.1.1.9 ROUTED BYPASS

This section shows the configuration of Routed Bypasses.



5.1.1.10 AWS SSM AGENT

This section shows the status of the AWS SSM Agent.

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Wed 2021-06-23 08:54:48 UTC; 12min ago
Registration values: {"ManagedInstanceID":"mi-0904b06fee9ef0c74","Region":"us-east-1"}
```

5.1.1.11 SYSLOG/SIEM Servers Information

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

```
SYSLOG/SIEM Servers Information
Primary Syslog IP: 172.31.200.163
Secondary Syslog IP: Not configured
Syslog TCP port: 514
```

5.1.1.12 HIGH AVAILABILITY Information

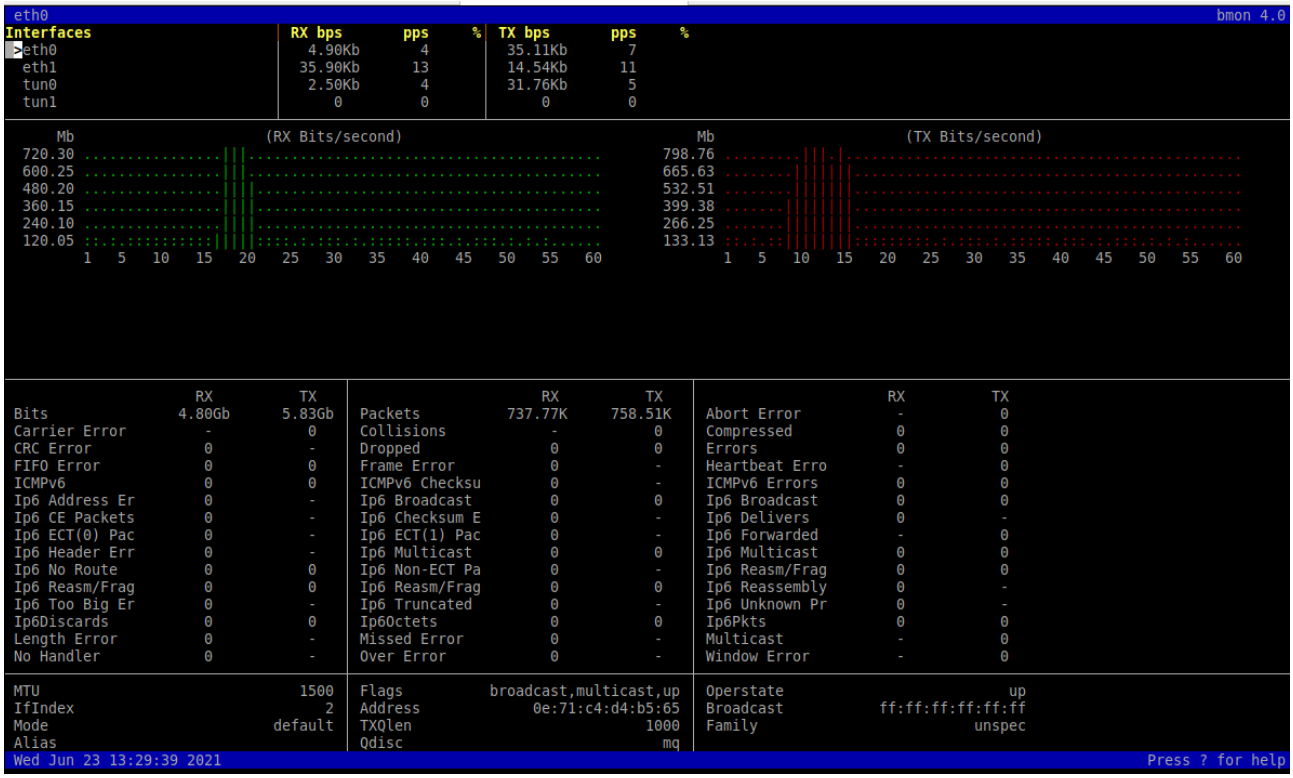
This section all the information when the CSC are configured on HA pair:

```
HIGH AVAILABILITY Information
The HA service is:  active (running) since Wed 2021-06-23 08:48:49 UTC; 18min 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-002976f0d8d060764 (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-08f80bd2952ff13ed
SNS message ARN= arn:aws:sns:us-east-1:544690173127:AWS-Zscaler-Notification
```

- 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.

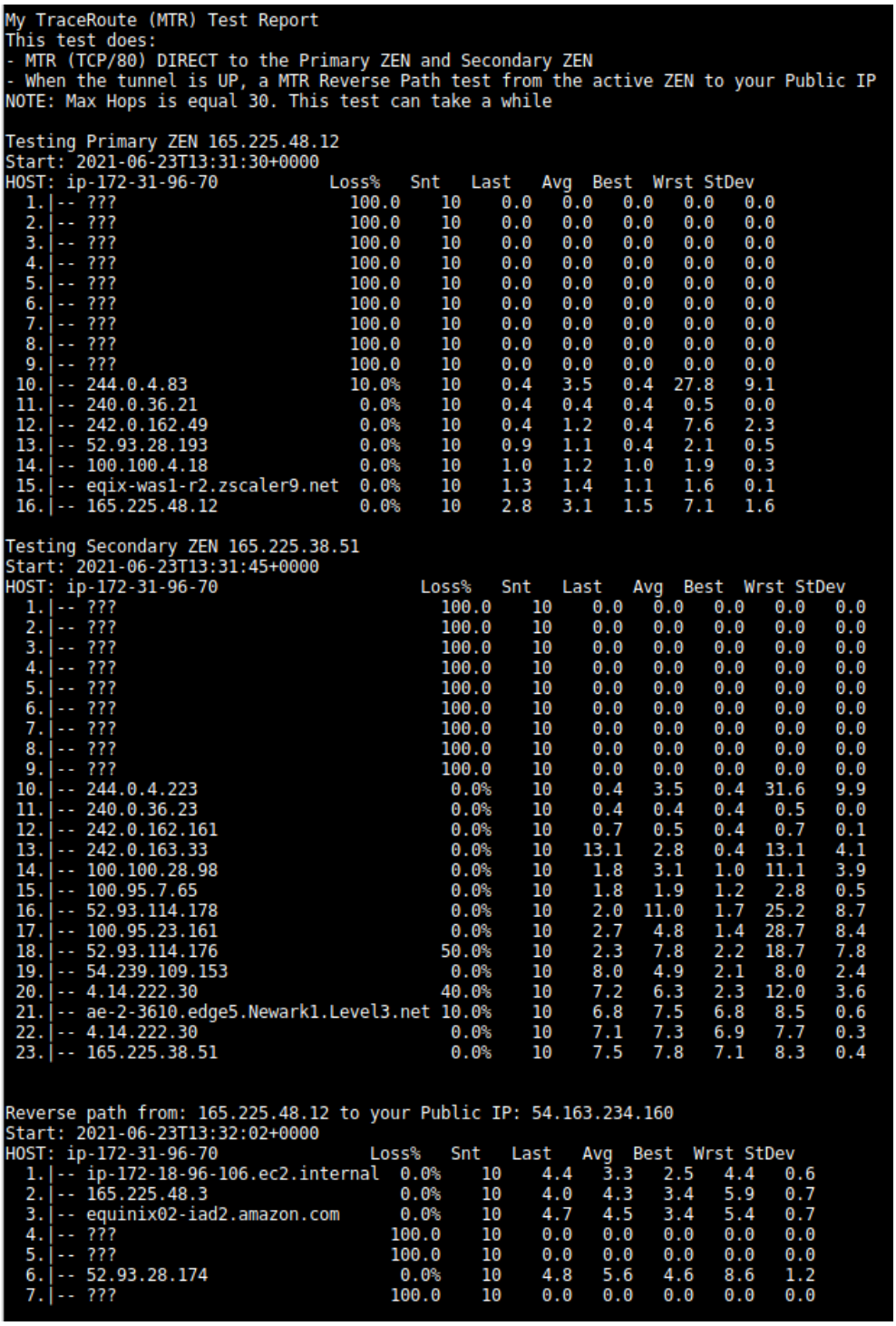
5.1.2 Show Interfaces Traffic

You can use this section to see the traffic in real time.



5.1.3 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.

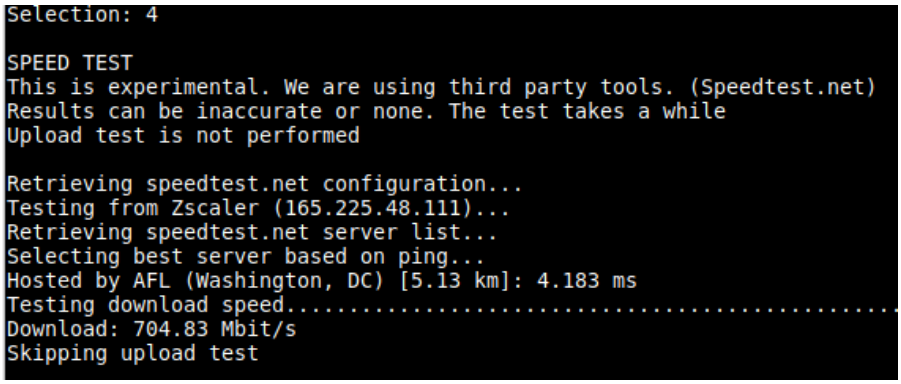


```
My TraceRoute (MTR) Test Report
This test does:
- MTR (TCP/80) DIRECT to the Primary ZEN and Secondary ZEN
- When the tunnel is UP, a MTR Reverse Path test from the active ZEN to your Public IP
NOTE: Max Hops is equal 30. This test can take a while

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
```

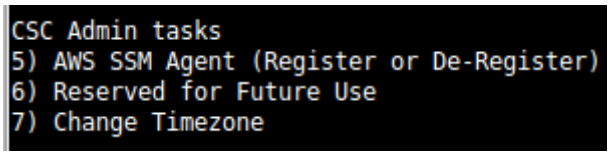
5.1.4 SPEED TEST

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

We are using t3a.small instance for the CSC, and the download value was 704.83 Mbit/s.



5.2 CSC Admin Tasks



5. AWS SSM Agent (Register or De-Register)

6. Reserved.

7. Change Timezone: In case if needed, you can select your Timezone here.

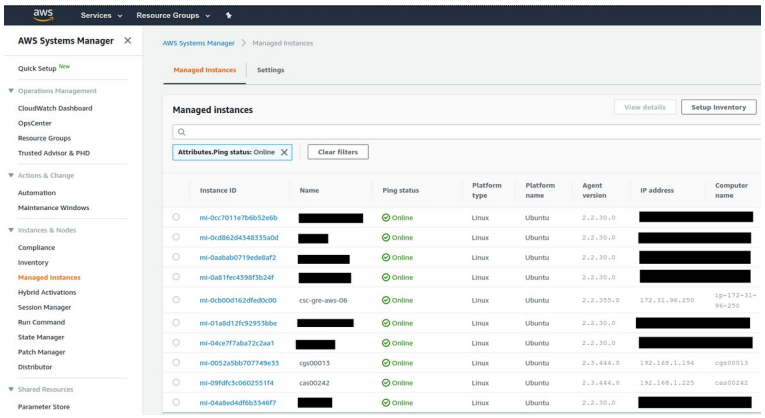
5.2.1 AWS SSM Agent (Register / De-Register)

The CSC AWS has installed the AWS SSM Agent that allows you to check remotely the status of the CSC and “Run Commands” using AWS Systems Manager.

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

Important (**): You can manage the CSC for AWS from ANY AWS Availability Zone. In this example, we have running the CSC for AWS on N. Virginia (us-east-1) but we want to manage it from the Ireland (eu-west-1). It is advisable to manage all CSC (for AWS/Azure/Vmware/Hyper-V) from the same AWS availability zone.

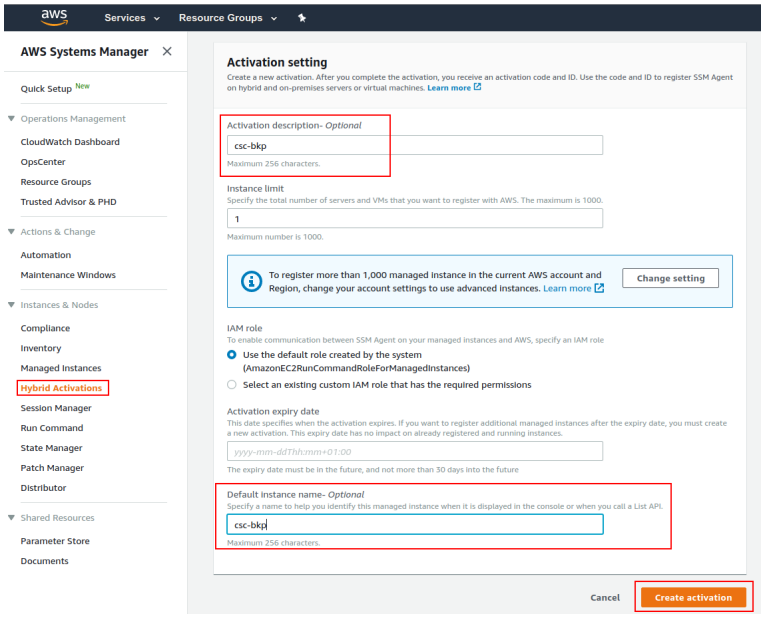
Here is the screenshot of our CSCs under management before to add the CSC for AWS:



Please, note that in this example the availability zone is eu-west-1

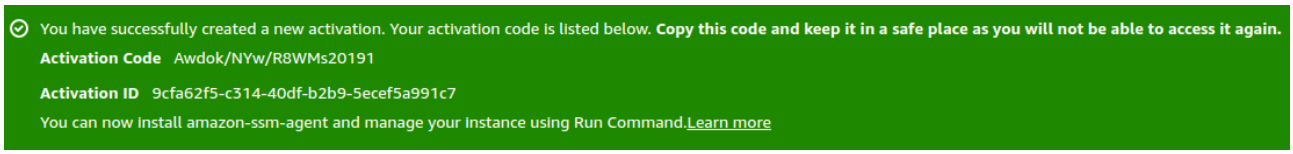
The steps required to register the AWS SSM Agent are two:

1. Go to: AWS Systems Manager → Hybrid Activations → click "Create activation"



Note: We recommend to create an Activation per CSC and on "Default instance name" to put the name of the CSC instance (or CSC ID o or the name of your "Location" for easy identification)

When you click "Create an Activation" you will receive the following information:



Please, keep copy this values on a safe place. You will need this to register the AWS SSM client on the CSC.

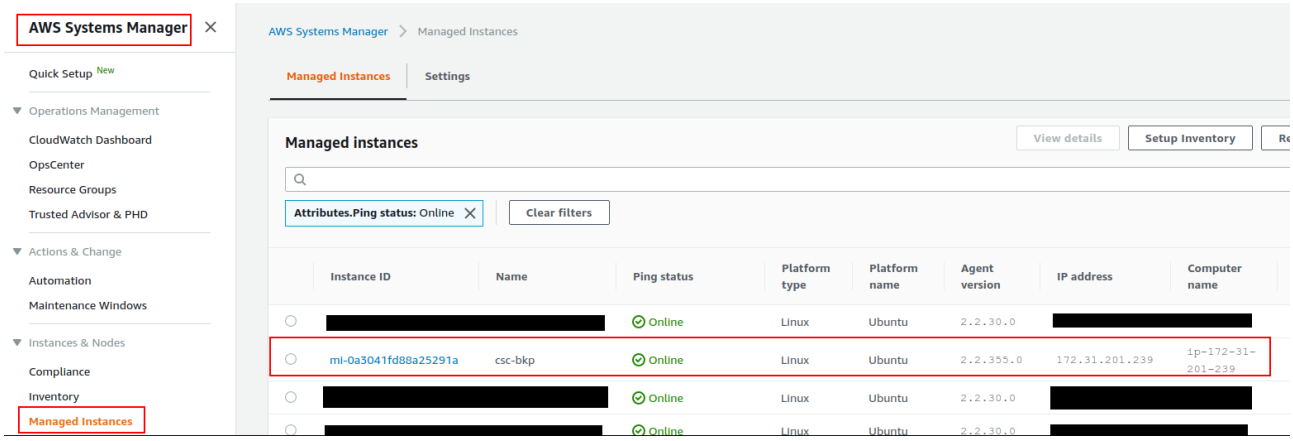
2. From the CSC Admin Tasks Menu, select “5) AWS SSM Agent (Register or De-Register)”. You will asked for the Activation Code, Activation ID and AWS Region where to register the CSC. (Check your AWS URL https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=**eu-west-1**#)

```
Selection: 5

The SSM Agent is inactive (dead) since Wed 2019-10-16 12:51:23 GMT; 9min ago

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 :Awdok/NYw/R8WMs20191
Activation ID :9cfa62f5-c314-40df-b2b9-5ecef5a991c7
Region :eu-west-1
```

Done! Check now “Show Configuration and Status” :

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Wed 2019-10-16 13:01:21 GMT; 1min 26s ago
Registration values: {"ManagedInstanceID":"mi-0a3041fd88a25291a","Region":"eu-west-1"}
```

Go to AWS Systems Manager → Managed Instances and you will see the new CSC added. (csc-bkp in this case)



5.2.2 Reserved for future use

This menu is reserved for future use.

5.2.3 Change Timezone

You can change the Timezone using this menu.

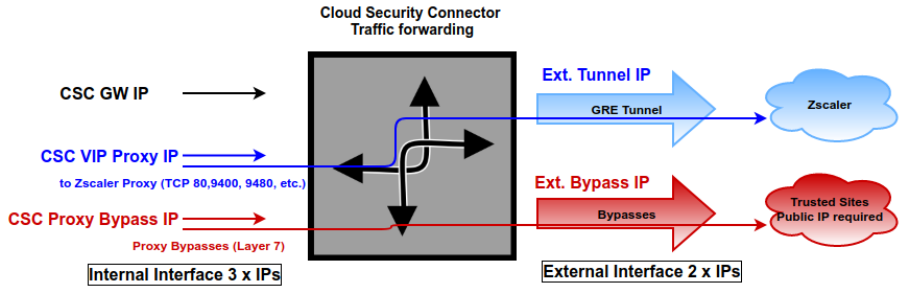
--- o ---

5.3 Proxy Bypass

When using PAC files, the Bypass Proxy allows you to connect certain domains direct to the Internet. By default, all domains are blocked, and you need to insert the domains that you want to allow to go direct.

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

5.3.1 Proxy Bypass - Traffic Flow



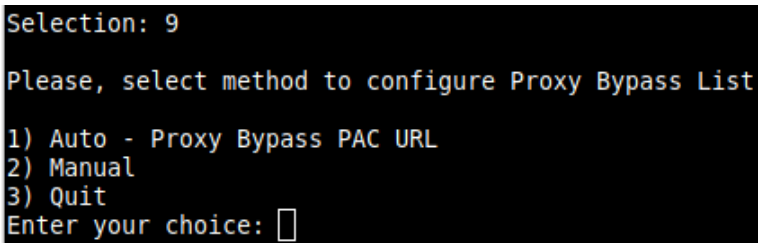
5.3.2 View Current Proxy Bypass List

This commands shows the current domains and subdomains allows to go direct to Internet. By default the list is “blank” blocking all traffic.

```
Selection: 8

This is the list of current Domains configured:

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

5.3.3 Configure Proxy Bypass List

In order to configure the Bypass List you have two options:



5.3.3.1 Auto – Proxy Bypass PAC URL

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

By default, the CSC has configured this PAC URL:

*http://pac.**<yourcloudname>**.net/**something**/<pacname>.pac*

** You can change this URL via console menu. You can use an internal URL if you want.*

The "Proxy Bypass PAC file" idea is to act as a central repository of all Layer 7 bypasses required. Moreover, if you manage the CSCs using AWS, you can update all CSCs in your network doing one AWS Run Command.

Example of "Proxy Bypass PAC file"

```
function FindProxyForURL(url, host) {
      var bypassproxy="PROXY 1.1.1.1:3128; PROXY 2.2.2.2:3128";

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

   // Bypass via CSC Public IPs (Examples)
      // Okta Domains (for Location Rules)
   if ((shExpMatch(host, "*.okta.com")) ||
      (shExpMatch(host, "*.oktacdn.com")) ||
      (shExpMatch(host, "*.okta-emea.com")) ||
      (shExpMatch(host, "login.mydomain.com")) ||
      // O365 Domains for ConditionalAccess
      (shExpMatch(host, "login.microsoftonline.com")) ||
      (shExpMatch(host, "login.microsoft.com")) ||
      (shExpMatch(host, "login.windows.net")) ||
      // IP / Port test page
      (shExpMatch(host, "portquiz.net"))) {
      return bypassproxy
   }
// ======================================================
```

```
   return bypassproxy
}
```

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

Note 2: The Bypass Proxy port is 3128

5.3.3.2 Example Using Proxy Bypass

Scenario: It is required to send Okta and O365 Authentication URLs via the customer's public IP and no Zscaler IPs.

Requirements:

1. Create the “Proxy Bypass PAC” with the list of domains you want to bypass on your Zscaler console. Copy the URL.

Proxy Bypass PAC on Zscaler Console:

PAC FILE CONTENTS

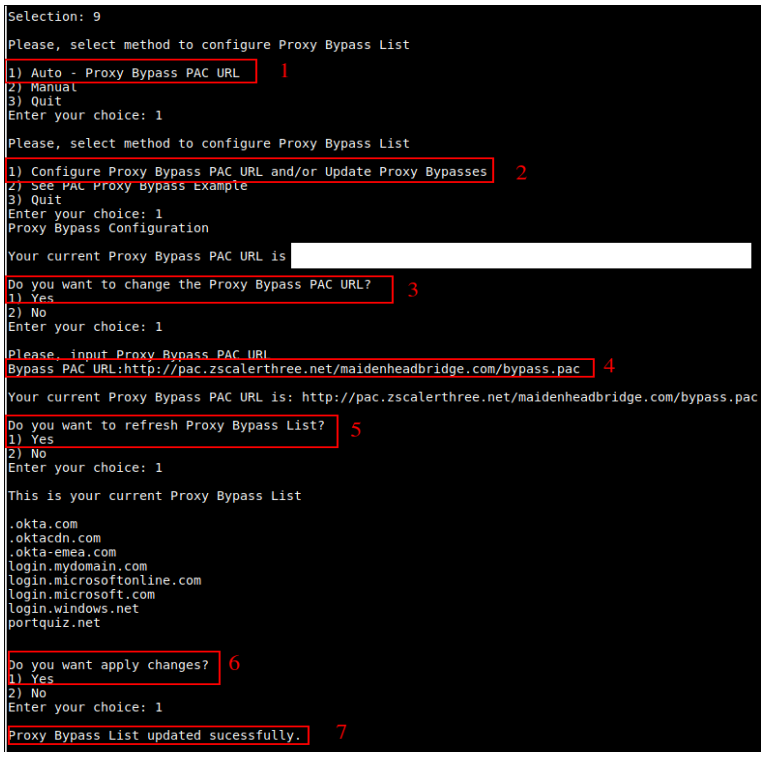
```
function FindProxyForURL(url, host) {
        var bypassproxy="PROXY 1.1.1.1:3128; PROXY 2.2.2.2:3128";

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

    // Bypass via CSC Public IPs (Examples)
        // Okta Domains (for Location Rules)
    if ((shExpMatch(host, "*.okta.com")) ||
        (shExpMatch(host, "*.oktacdn.com")) ||
        (shExpMatch(host, "*.okta-emea.com")) ||
        (shExpMatch(host, "login.mydomain.com")) ||
        // O365 Domains for ConditionalAccess
        (shExpMatch(host, "login.microsoftonline.com")) ||
        (shExpMatch(host, "login.microsoft.com")) ||
        (shExpMatch(host, "login.windows.net")) ||
        // IP / Port test page
        (shExpMatch(host, "portquiz.net"))) {
        return bypassproxy
    }
// ======================================================
    return bypassproxy
}
```

Note: The line “var bypassproxy = "PROXY 1.1.1.1:3128; PROXY 2.2.2.2:3128";” is doing nothing here. You can leave as is on the Proxy Bypass PAC, but you need to correct the production PAC values.

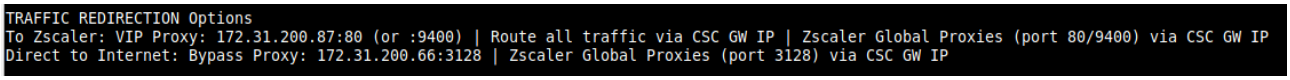
Bypass PAC URL on this example

http://pac.zscalerthree.net/maidenheadbridge.com/bypass.pac

2. Configure the URL of the “Proxy Bypass PAC” on each CSC and refresh the list.



3. Check your VIP Proxy and Bypass Proxy using “Show Configuration and Status” on the CSC.



4. Create your production PAC file and Copy/Paste section 3 for Proxy Bypasses, and put the correct values for variables “tozscaler” and “bypassproxy”

```
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: Assigning values to "tozscaler" and "bypassproxy"
    //

    // CSC VIP
    var tozscaler = "PROXY 172.31.200.87:80";
    // CSC Proxy Bypass
    var bypassproxy = "PROXY 172.31.200.66:3128";

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

```
    // Bypass via CSC Public IPs (Examples)
    // Okta Domains (for Location Rules)
    if ((shExpMatch(host, "*.okta.com")) ||
        (shExpMatch(host, "*.oktacdn.com")) ||
        (shExpMatch(host, "*.okta-emea.com")) ||
        (shExpMatch(host, "login.mydomain.com")) ||
        // O365 Domains for ConditionalAccess
        (shExpMatch(host, "login.microsoftonline.com")) ||
        (shExpMatch(host, "login.microsoft.com")) ||
        (shExpMatch(host, "login.windows.net")) ||
        // IP / Port test page
        (shExpMatch(host, "portquiz.net"))) {
        return bypassproxy
    }
    // ============================================================
    // Section 4: Default Traffic

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

Production PAC file URL on this example

http://pac.zscalerthree.net/maidenheadbridge.com/production.pac

5. Checking traffic “tozscaler” and “bypass” using CURL command:

CURL command is available on Linux and Win10. You can check the traffic “tozscaler” or “proxy bypass” using the following commands:

Traffic “tozscaler”

Put the values of the VIP Proxy on “--proxy”

In this example:

Linux:

```
curl -s --proxy http://172.31.200.87:80 ip.zscaler.com | grep You
```

Win 10:

```
curl -s --proxy http://172.31.200.87:80 ip.zscaler.com | findstr You
```

Result expected:

```
<div class="headline">You are accessing the Internet via Zscaler Cloud: Washington DC 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">165.225.48.118</span></div>
<div class="details">Your Gateway IP Address is <span class="detailOutput">54.163.234.160</span></div>
```

Proxy Bypass Traffic:

We are testing the website “portquiz.net” to check that the Bypass works. The page “portquiz.net” returns your public IP and TCP port. Please, note this is a third-party tool, and sometimes it takes a long to answer or is unresponsive.

In this example:

Linux: curl -s --proxy http://172.31.200.66:3128 portquiz.net

Win 10: curl -s --proxy http://172.31.200.66:3128 portquiz.net

Result expected:

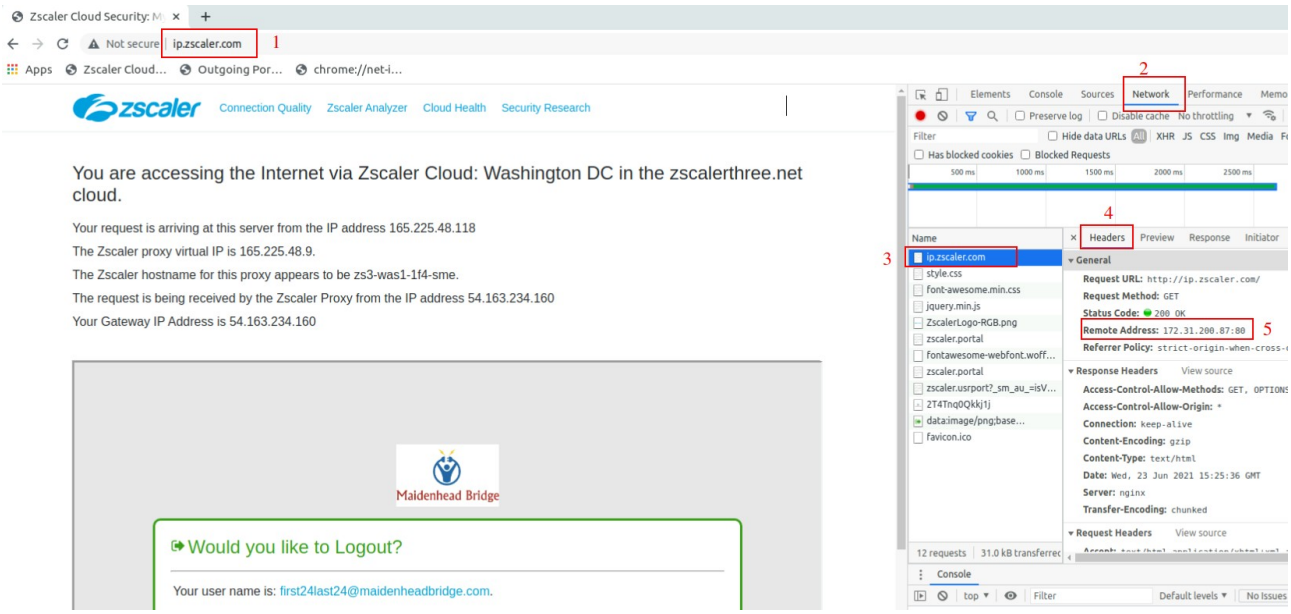
Port 80 test successful! Your IP: **18.204.121.250**

6. Checking traffic “tozscaler” and “bypass” using Browser command:

Please, setup the Browser proxy using the production PAC URL. In our example is: http://pac.zscalerthree.net/maidenheadbridge.com/production.pac

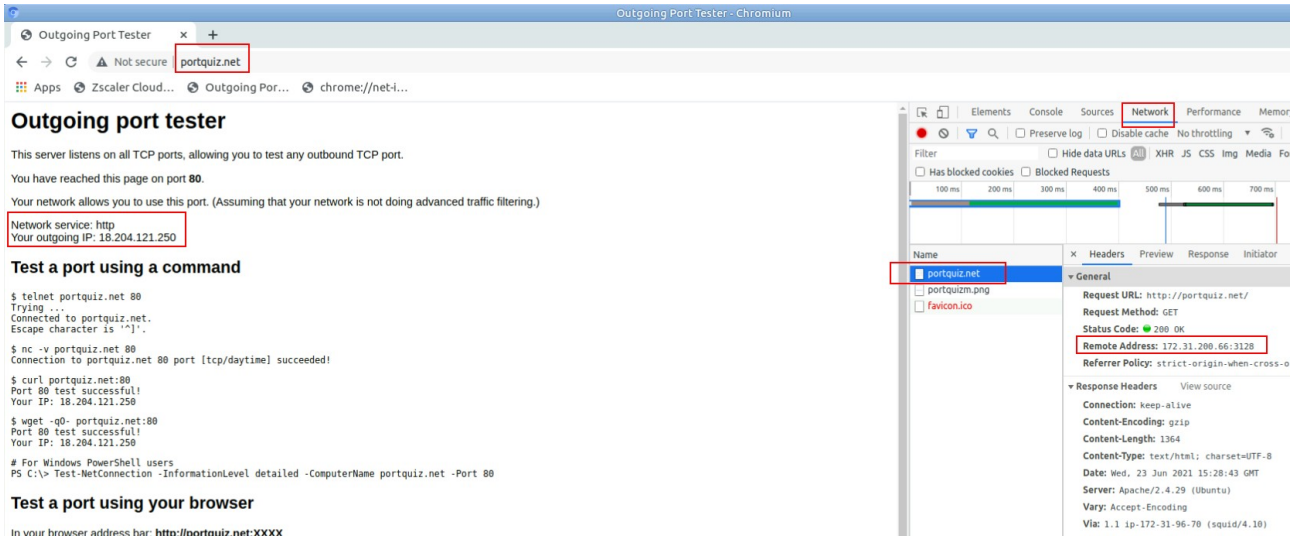
Traffic “tozscaler”

Go to page: ip.zscaler.com and using Chrome Developer tools check the proxy in use:



Proxy Bypass Traffic

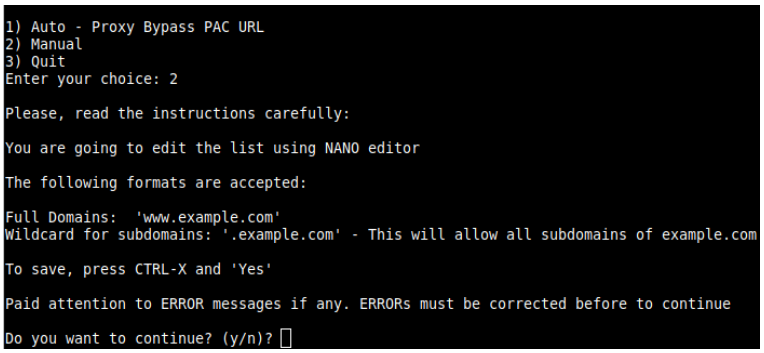
Using our example, we are going to http://portquiz.net and using Chrome Developer tools check the proxy in use:



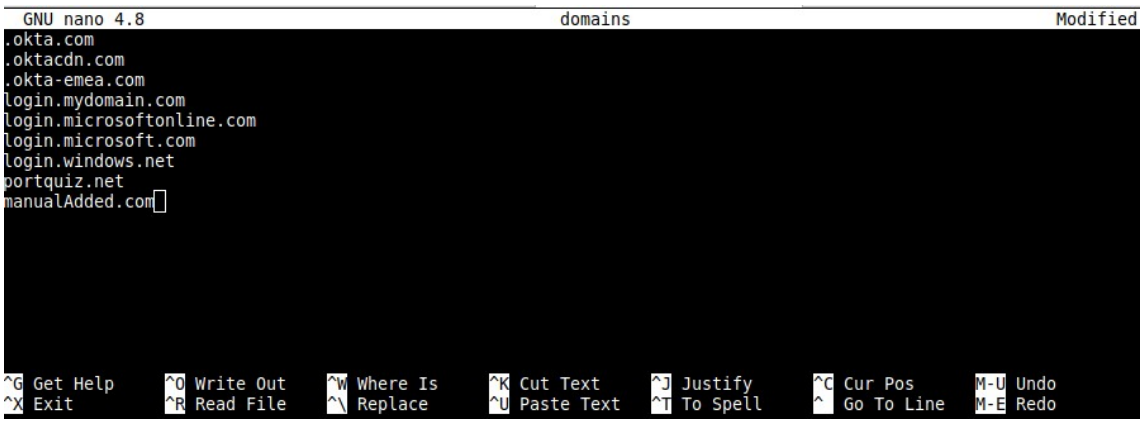
5.3.3.3 Manual

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

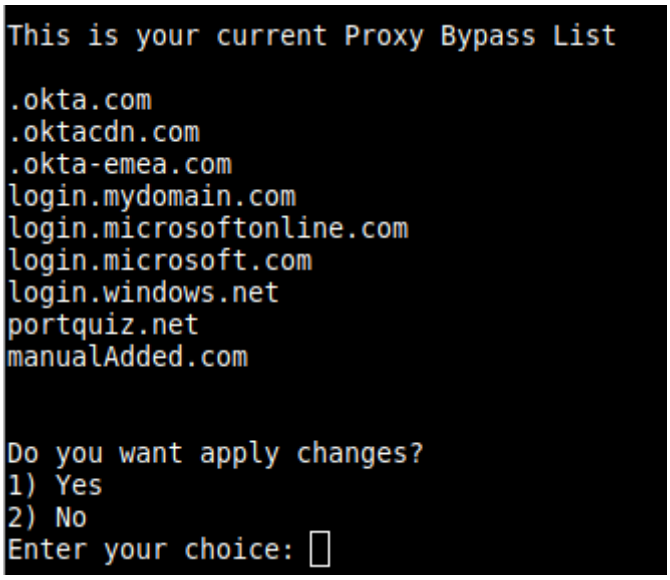
1. Select Option 2)



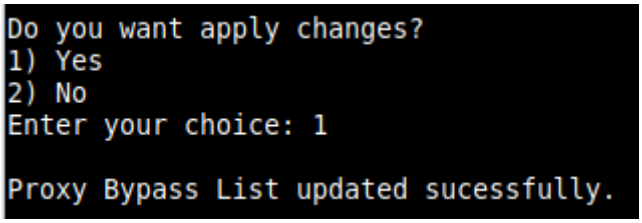
2. Input “y”



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.



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

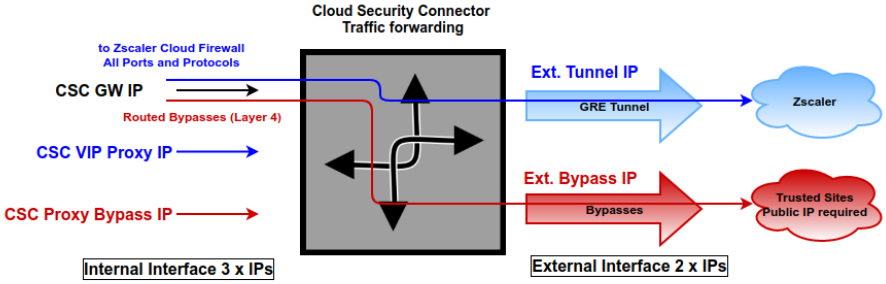


5.4 Routed Bypass

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 GRE tunnel to Zscaler. If you want to bypass the GRE tunnel, you need to create a Routed Bypass Rule.

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

5.4.1 Routed Bypass - Traffic Flow



5.4.2 View Current Routed Bypass List

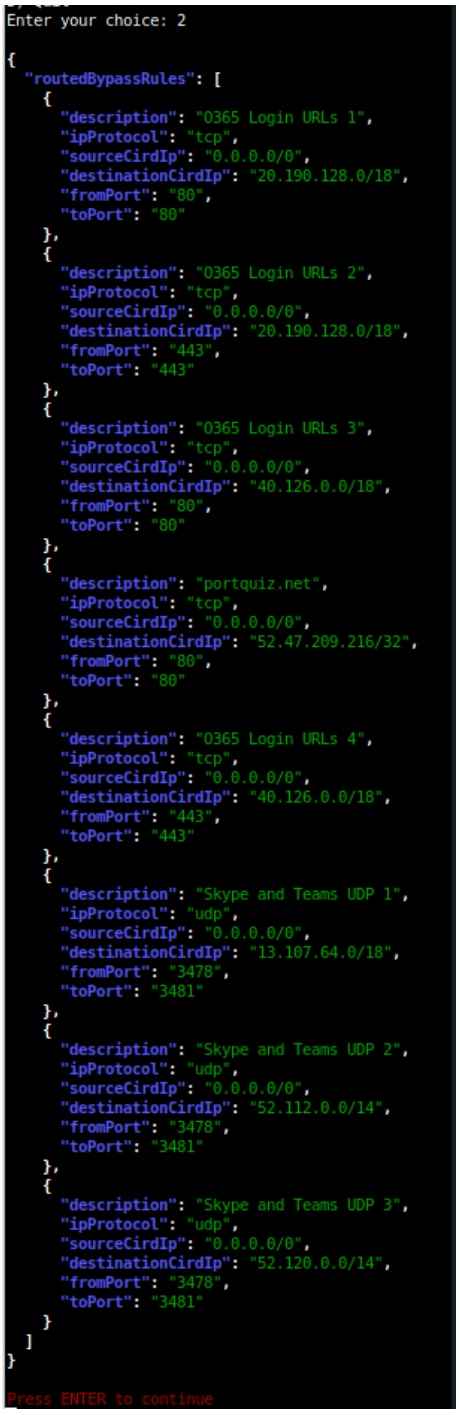
You can select to view the Routed Bypass Rules in Compact format or JSON.

```
Selection: 10

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

5.4.2.1 Compact

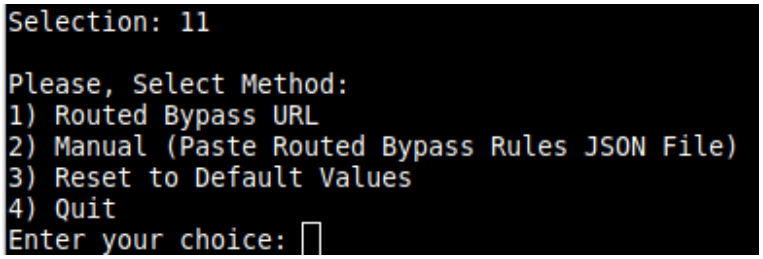
```
Current Values configured are:

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"
```

5.4.2.2 Json


Enter your choice: 2
{
"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"
}
]
}
Press ENTER to continue

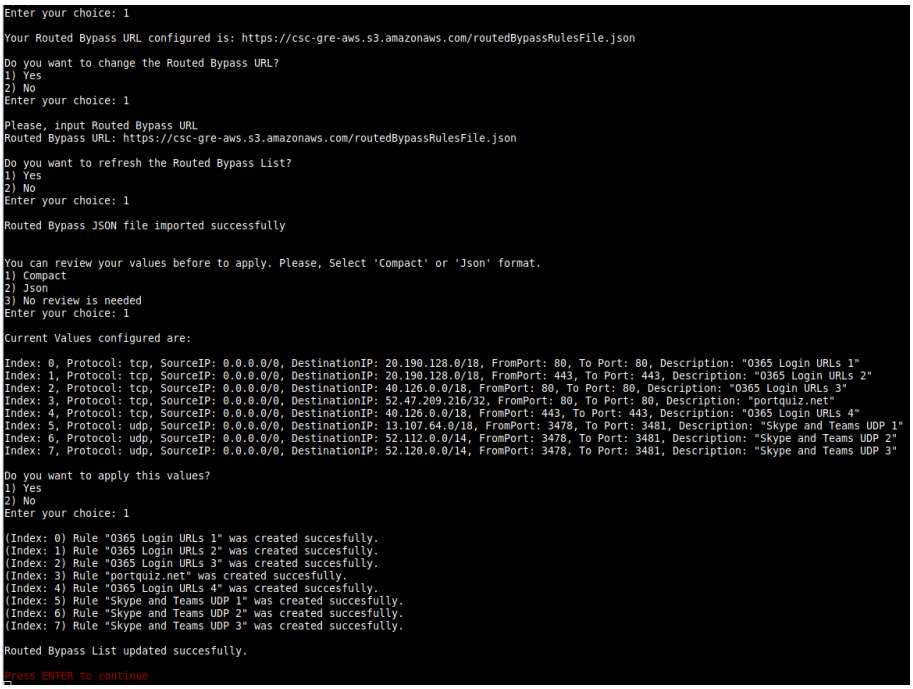
5.4.3 Configure Routed Bypass List

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



5.4.3.1 Routed Bypass URL

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



5.4.3.2 Manual (Paste Routed Bypass JSON file)

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

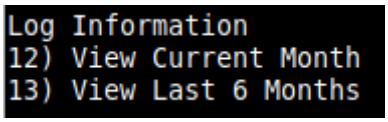
```
Enter your choice: 2


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:

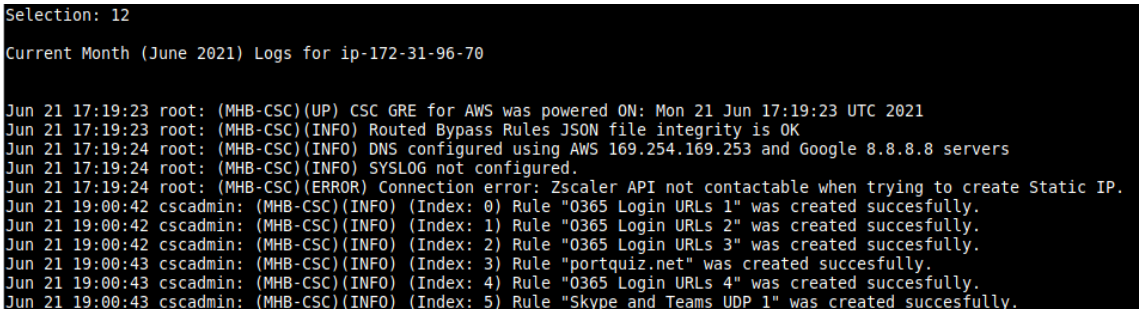
```
        },
        {
            "description": "Skype and Teams UDP 3",
            "ipProtocol": "udp",
            "sourceCirdIp": "0.0.0.0/0",
            "destinationCirdIp": "52.120.0.0/14",
            "fromPort": "3478",
            "toPort": "3481"
        }
    ]
}

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
```

5.5 Log Information

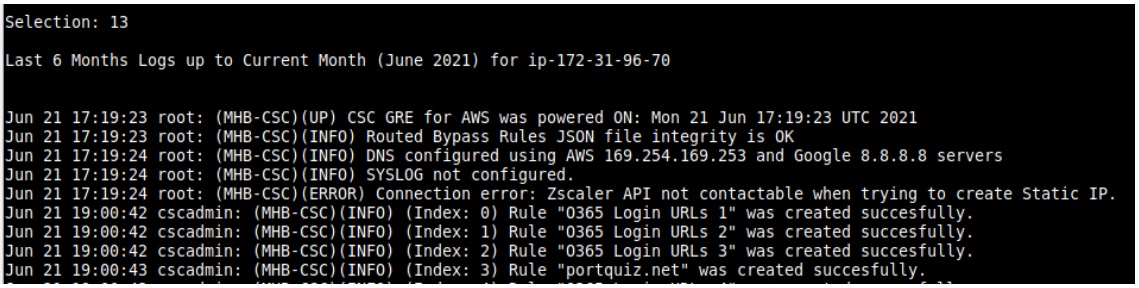
This section shows the Logs. You can see the Current Month or Last 6 Months.



5.5.1.1 View Current Month

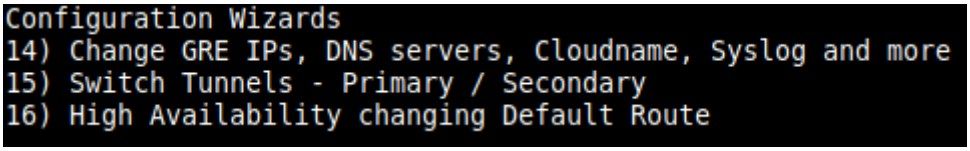


5.5.1.2 View Last 6 Months



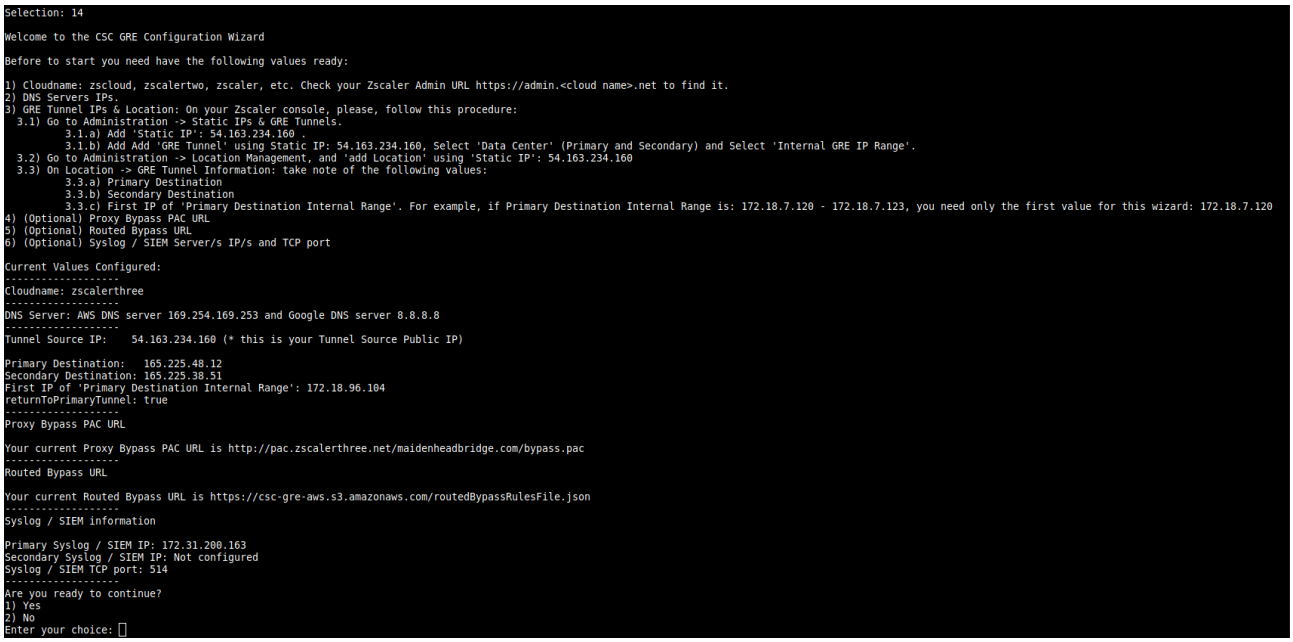
5.6 Configuration Wizards

In this section you can run the initial configuration wizard to change GRE IPs, DNS servers and/or Cloud Name, an easy way to Switch tunnels and to configure High Availability.



5.6.1 Change GRE IPs, DNS servers, Cloudname, Syslog and more

The CSC GRE Configuration Wizard explained at the beginning of this manual.



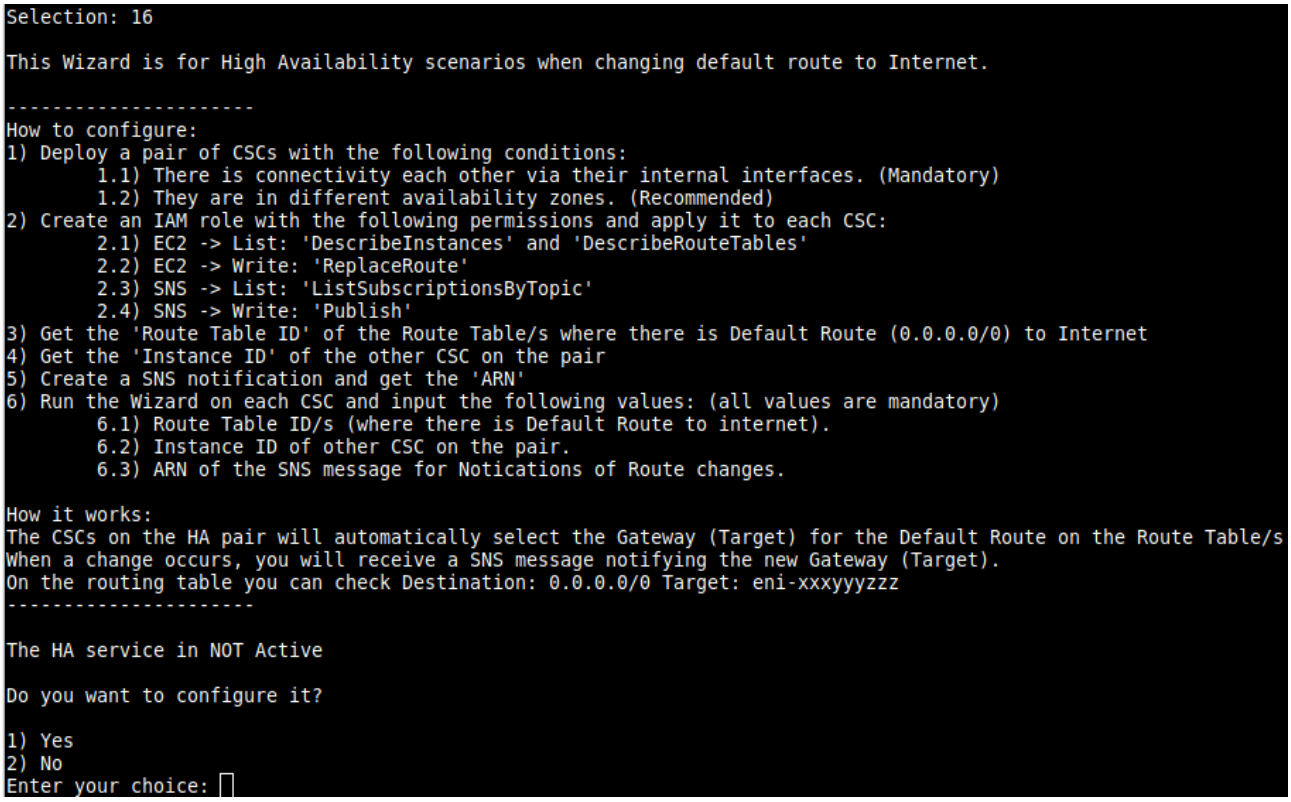
5.6.2 Switch Tunnels - Primary / Secondary

This section shows your current settings and statuses and allows to switch tunnels.

```
Selection: 15

------------------
ZSCALER INFORMATION
Zscaler Cloud:  zscalerthree
GRE tunnels egress Public IP: 54.163.234.160
Primary Tunnel:
                ZEN Public IP: 165.225.48.12
                Tunnel IPs (local/zen): 172.18.96.105 / 172.18.96.106
Secondary Tunnel:
                ZEN Public IP: 165.225.38.51
                Tunnel IPs (local/zen): 172.18.96.109 / 172.18.96.110

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: Tue 22 Jun 15:22:36 UTC 2021

HTTP://IP.ZSCALER.COM PAGE STATUS
You are accessing the Internet via Zscaler Cloud: Washington DC in the zscalerthree.net cloud.
Your Gateway IP Address is 54.163.234.160
------------------
Do you want to Switch Primary / Secondary Tunnel?
Selecting Yes will disrupt all current connections.

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

5.6.3 High Availability changing Default Route

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



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:

    2.1) EC2 → List: 'DescribeInstances' and 'DescribeRouteTables'

    2.2) EC2 → Write: 'ReplaceRoute'

    2.3) SNS → List: 'ListSubscriptionsByTopic'

    2.4) SNS → Write: 'Publish'

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 each CSC and input the following values: (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.

How it works:

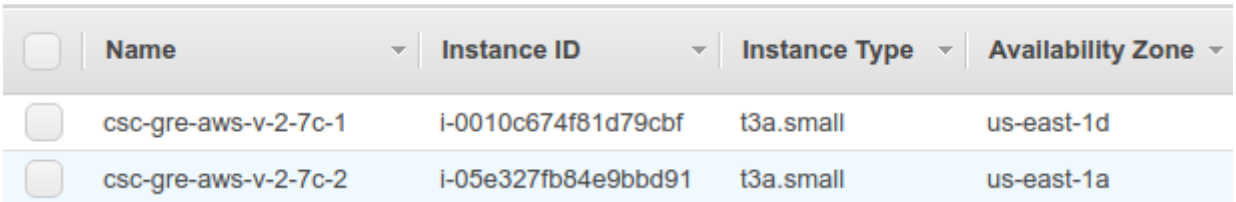
The CSCs on the HA pair will automatically select the Gateway (Target) for the Default Route on the Route Table/s.

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

5.6.3.1 High Availability configuration on detail

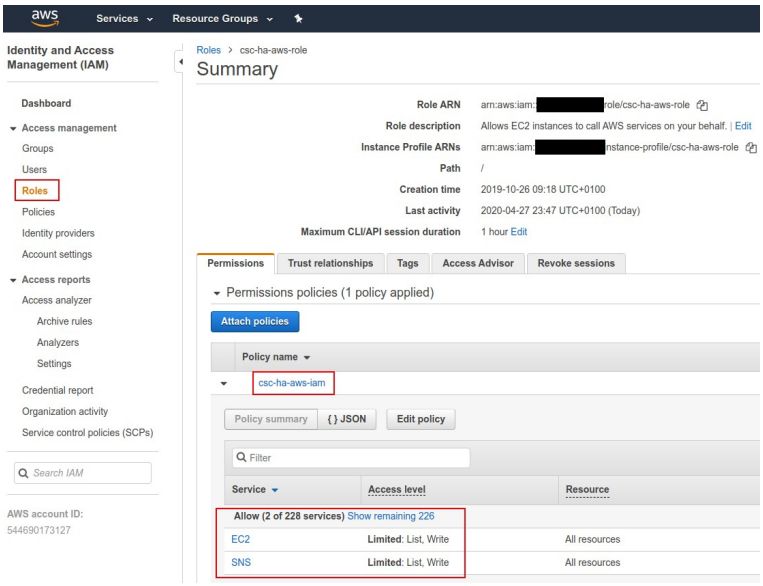
1. **Deploy a pair of CSC on the different availability zones.**

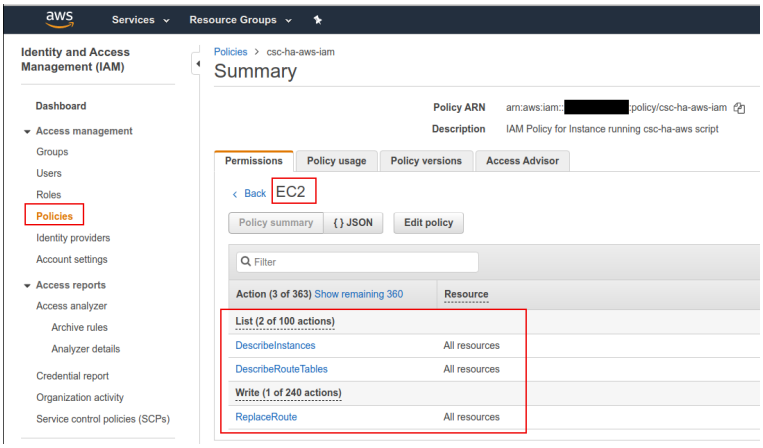


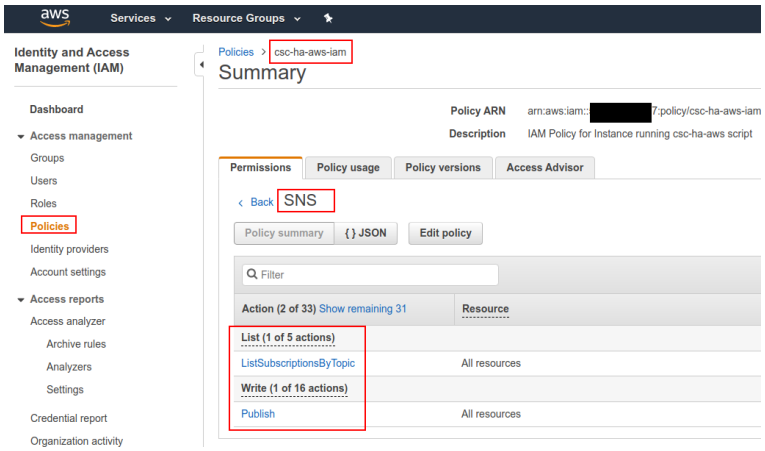
Name	Instance ID	Instance Type	Availability Zone
csc-gre-aws-v-2-7c-1	i-0010c674f81d79cbf	t3a.small	us-east-1d
csc-gre-aws-v-2-7c-2	i-05e327fb84e9bbd91	t3a.small	us-east-1a

2. **Create an IAM role with the following policies:**

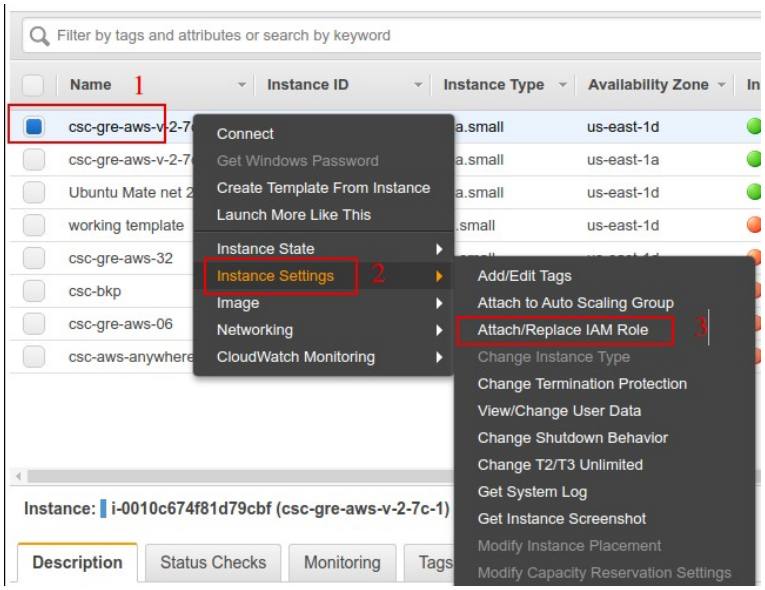
EC2 → List: 'DescribeInstances' and 'DescribeRouteTables'
EC2 → Write: 'ReplaceRoute'
SNS → List: 'ListSubscriptionsByTopic'
SNS → Write: 'Publish'

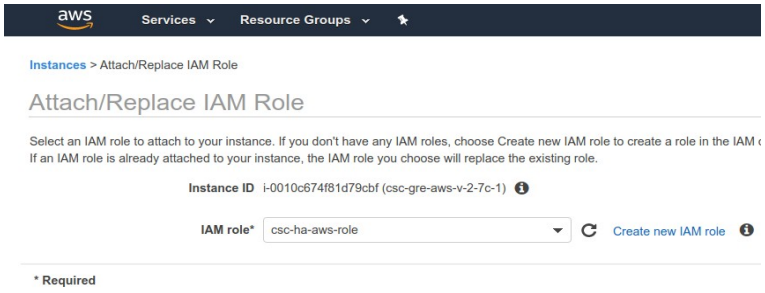

aws
Services
Resource Groups
Identity and Access Management (IAM)
Dashboard
Access management
Groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
Search IAM
AWS account ID:
544690173127
Roles > csc-ha-aws-role
Summary
Role ARN
arn:aws:iam:[redacted]role/csc-ha-aws-role
Role description
Allows EC2 instances to call AWS services on your behalf. | Edit
Instance Profile ARNs
arn:aws:iam:[redacted]instance-profile/csc-ha-aws-role
Path
/
Creation time
2019-10-26 09:18 UTC+0100
Last activity
2020-04-27 23:47 UTC+0100 (Today)
Maximum CLI/API session duration
1 hour Edit
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (1 policy applied)
Attach policies
Policy name
csc-ha-aws-iam
Policy summary
{ } JSON
Edit policy
Filter
Service
Access level
Resource
Allow (2 of 228 services) Show remaining 226
EC2
Limited: List, Write
All resources
SNS
Limited: List, Write
All resources


aws
Services
Resource Groups
Identity and Access Management (IAM)
Dashboard
Access management
Groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzer details
Credential report
Organization activity
Service control policies (SCPs)
Policies > csc-ha-aws-iam
Summary
Policy ARN
arn:aws:iam::[redacted]policy/csc-ha-aws-iam
Description
IAM Policy for Instance running csc-ha-aws script
Permissions
Policy usage
Policy versions
Access Advisor
Back
EC2
Policy summary
{ } JSON
Edit policy
Filter
Action (3 of 363) Show remaining 360
Resource
List (2 of 100 actions)
DescribeInstances
All resources
DescribeRouteTables
All resources
Write (1 of 240 actions)
ReplaceRoute
All resources

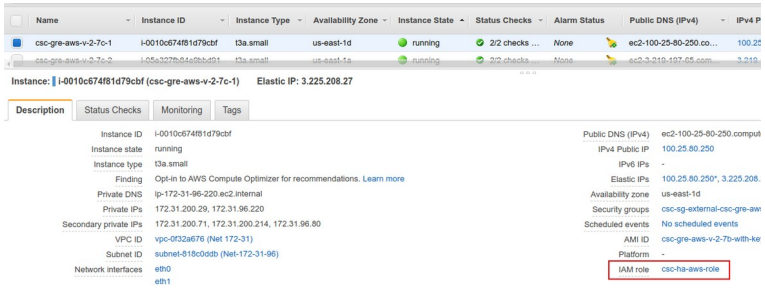

aws
Services
Resource Groups
Identity and Access Management (IAM)
Dashboard
Access management
Groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Policies > csc-ha-aws-iam
Summary
Policy ARN
arn:aws:iam::[redacted]7:policy/csc-ha-aws-iam
Description
IAM Policy for Instance running csc-ha-aws script
Permissions
Policy usage
Policy versions
Access Advisor
Back
SNS
Policy summary
{ } JSON
Edit policy
Filter
Action (2 of 33) Show remaining 31
Resource
List (1 of 5 actions)
ListSubscriptionsByTopic
All resources
Write (1 of 16 actions)
Publish
All resources

3. Apply the IAM roles to each CSC on the pair.





Check the IAM role is assigned:

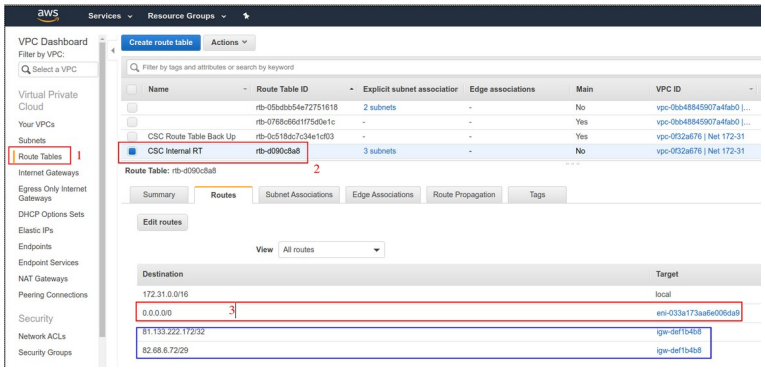


Attached the role to the other CSC on the pair as well.

4. Obtain your Route Table ID:

Note: You can add multiple Route Table ID on this version 2.8

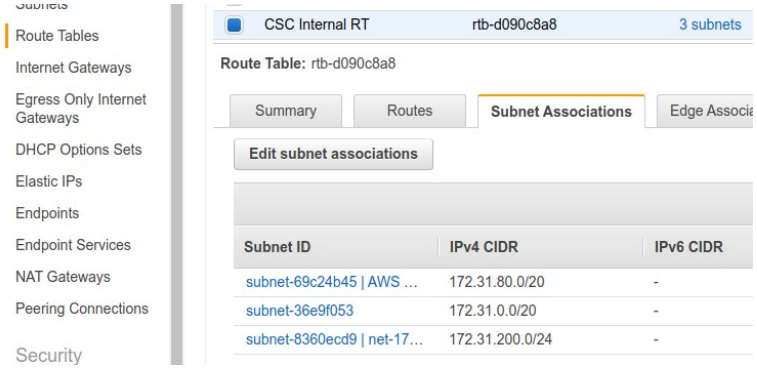
Go to VPC → Route Tables and get your table ID



Note 1: The CSC pair will modify the default route (0.0.0.0/0)"Target", setting the "eni-xxyy" values. Other Destination will remain untouched.

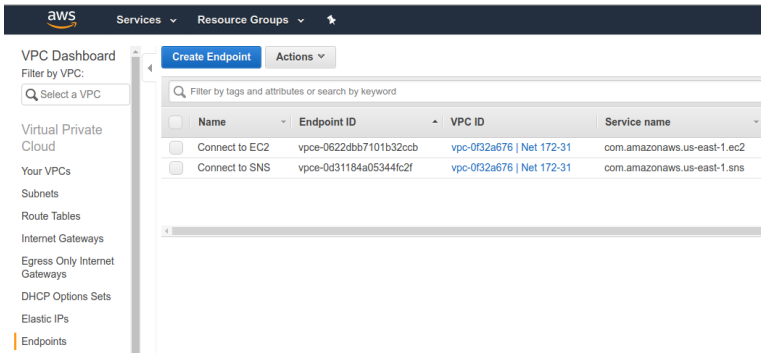
Note 2: We sure to add other destinations, like your internal subnets or your public IPs via the proper "Target" in order to do not loose connectivity to the VPC.

Apply the Subnet Associations to the Routing Table:



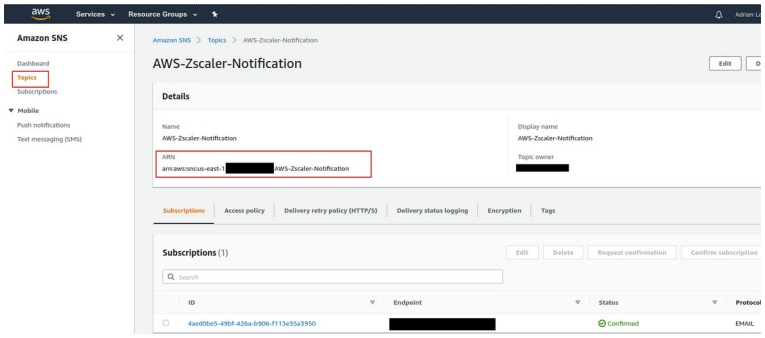
5. Create "Endpoints" to AWS services (EC2, SNS, S3, etc)

When changing the default route to internet via Zscaler you potentially will lost contact with some AWS services. For the CSC on the pair is mandatory to create two endpoints: to EC2 and SNS. Please, note that you may require to add some more for your services.



6. Create SNS message for Alerts.

Obtain the ARN



7. Finally Run the HA Wizard on each CSC

```
Do you want to configure it? (y/n)? y

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

Please, input the following values
Route Table ID= rtb-d090c8a8
Do you want to add another Route Table ID? (y/n)? y
Route Table ID= rtb-0994dbce551a782f0
Do you want to add another Route Table ID? (y/n)? n
Instance ID of other CSC in the pair= i-0dabbfe38df52b9cd
SNS message ARN= arn:aws:sns:us-east-1:544690173127:AWS-Zscaler-Notification

Values to configure are:

Routing Tables=2
Routing Table ID= rtb-d090c8a8
Routing Table ID= rtb-0994dbce551a782f0
Instance ID of other CSC in the pair= i-0dabbfe38df52b9cd
SNS message ARN= arn:aws:sns:us-east-1:544690173127:AWS-Zscaler-Notification

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

CSC HA is : active (running) since Mon 2020-04-27 23:16:36 UTC; 28ms ago
```

Done!

8. Notifications from CSC on HA

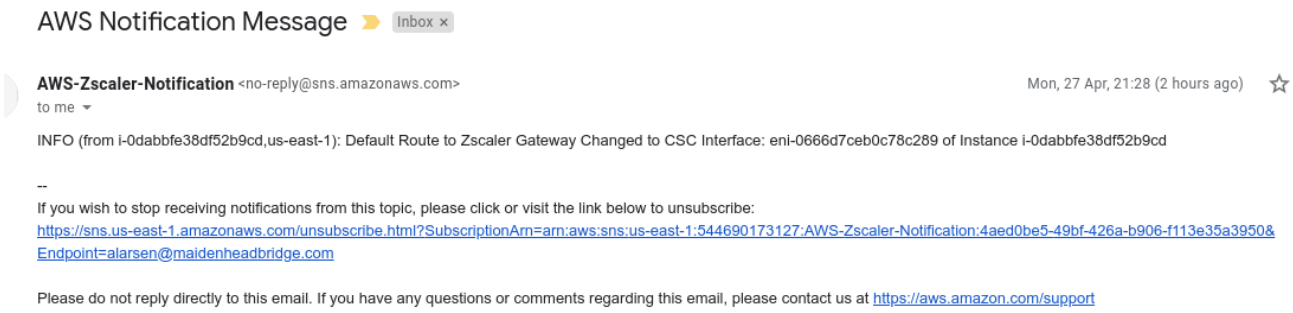
Each CSC on the pair will send notifications when:

→ 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:



Also, Logs are generated:

```
Apr 27 23:17:40 root: (MHB-CSC)(INFO) Default Route to Zscaler using CSC Interface: eni-033c4c5e5de9e784b of Instance i-027f6d85b46cb4b2b
```

6 DevOps operations

One of the objectives of version 3.0 is to create a product that can be deployed and managed without the requirement to SSH the CSC for configuration.

Using Advanced Deployment, it is possible to configure all AWS and Zscaler resources, and the CSC is ready for production after. 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 version 3.0 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.

Three JSON files are available on the CSC:

1. **highAvailability.json**: Allows administrators to configure the CSC on HA pair.
2. **config.json**: Allows administrators to modify specific values on the CSC like GRE Primary, Secondary ZEN nodes, DNS, Syslog, Routed Bypass URL, Proxy Bypass URL, etc.
3. **routedBypassRulesFile.json**: Allows administrators manually configure Routed Bypass Rules when the recommended Routed Bypass URL is not in use.

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

6.1 highAvailability.json file

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

Steps:

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

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

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

```
{
  "highAvailability": {
    "haEnable": ,
    "haIamRole": "",
    "haSnsMessageArn": "",
    "haInstanceIdOtherCsc": "",
    "haVPC": "",
    "haRouteTables": [
      {
        "routeTableId": ""
      }
    ]
```

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

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

```
{
    "highAvailability": {
        "haEnable": true,
        "haIamRole": "",
        "haSnsMessageArn": "arn:aws:sns:us-east-1:544690173127:AWS-Zscaler-Notification",
        "haInstanceIdOtherCsc": "i-070e878dbb0418f24",
        "haVPC": "vpc-0f32a676",
        "haRouteTables": [
            {
                "routeTableId": "rtb-d090c8a8"
            },
            {
                "routeTableId": "rtb-0994dbce551a782f0"
            }
        ]
    }
}
```

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

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

4. Apply the IAM Role to the CSC via AWS Console and Run Document "MHB-CSC-Reload-High-Availability" to apply the changes.

6.2 config.json file

You can use this file to change DNS, Log Servers, GRE Nodes (Primary/Secondary), etc.

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

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

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

```
{
  "model": "csc-gre-aws",
  "version": "1.0",
  "cloudName": "zscalerthree",
  "dns": {
    "useCloudDNS": true,
    "primaryDnsIP": "",
    "secondaryDnsIP": ""
  },
  "bypassProxyPacUrl": "http://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac",
  "syslogServers": {
    "primarySyslogIP": "172.31.200.163",
    "secondarySyslogIP": "",
    "syslogTcpPort": 514
  },
  "tunnelRedundancy": {
    "returnToPrimaryTunnel": false
  },
  "routedBypassJsonFileUrl": "https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json",
  "greCredentials": {
    "grePublicIP": "54.159.82.127",
    "primaryDestination": "165.225.8.30",
    "secondaryDestination": "165.225.38.51",
    "firstIpPrimaryDestinationRange": "172.17.152.96"
  }
}
```

2. Create a AWS bucket and place the modified "config.json" file on it.
3. Download the file to the CSC. Run Command "AWS-RunShellScript"

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

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

6.3 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/config.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 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
```

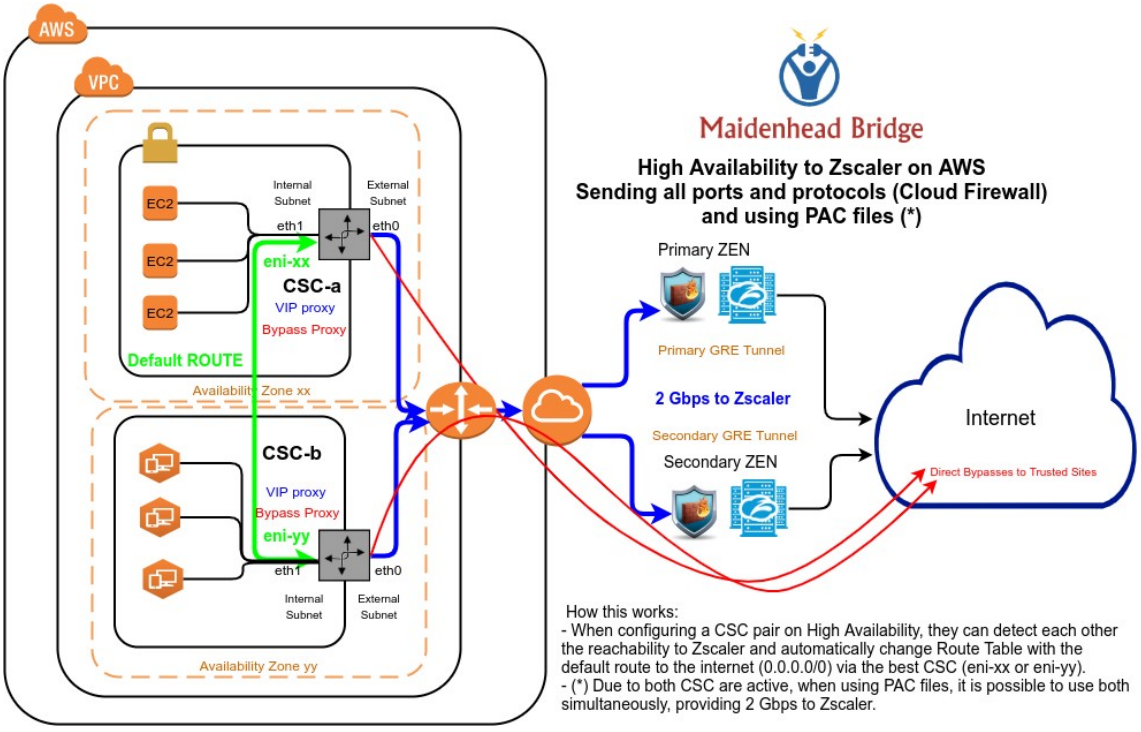
4. Run Document "MHB-CSC-Reload-Routed-Bypass-json" to apply the changes.

7 Appendix A – Traffic Redirection Example

7.1 High Availability Configuration - 2 Gbps

In this example, we will explain how to send all ports and protocols to Zscaler Cloud Firewall and how to use both CSC at the same time using PAC files.

7.1.1 Network Diagram



7.1.2 Traffic Redirection - Cloud Firewall

The configuration steps to send all ports and protocols to Zscaler Cloud Firewall are:

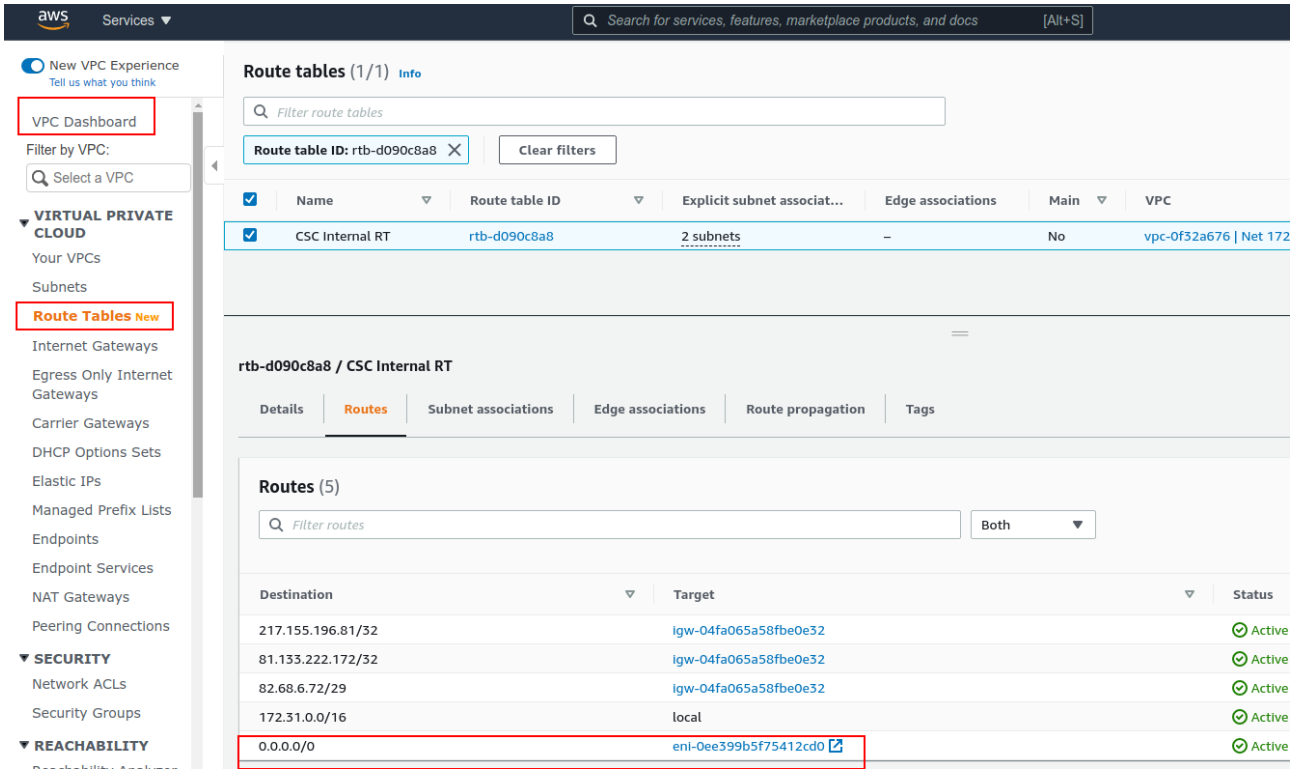
1. Route Table/s: Set the 0.0.0.0/0 with target the "eni-xx" of the CSC. This configuration will send all traffic to Zscaler Cloud Firewall.
2. Create the Routed Bypass Rules on the CSC for traffic going directly to the Internet to trusted sites, using your public IP.
3. Check the routes to other destinations on your Route Table and create "Endpoints" to AWS services if are required.

7.1.2.1 Route Table/s: Default Route to Internet via Zscaler.

The steps are:

1. Select the Route Table/s you want to use to send all traffic to Zscaler.
2. Configure the default route 0.0.0.0/0 via the eni-xx of one of the CSC on the HA pair. For example:

```
GENERAL INFORMATION
Availability Zone: us-east-1a
EC2 Instance id: i-0e3fabfae11f1e604 | Instance Type: t3a.small | ami-id: ami-09359699947139821
External Interface (eth0) Subnet-id: subnet-0da31f7b082ec13aa | Interface-id: eni-06dc04732bbb02d41 | Security-Group-id: sg-0c200e6d8282d52b8
Internal Interface (eth1) Subnet-id: subnet-0ccfb2ee4ab05371b | Interface-id: eni-0ee399b5f75412cd0 | Security-Group-id: sg-08e3a2a37380a9361
CSC date: Tue 29 Jun 03:25:21 EDT 2021
Soft version : 3.0
```



3. Complete the High Availability Setup. See section 5.6.3 High Availability of this manual.

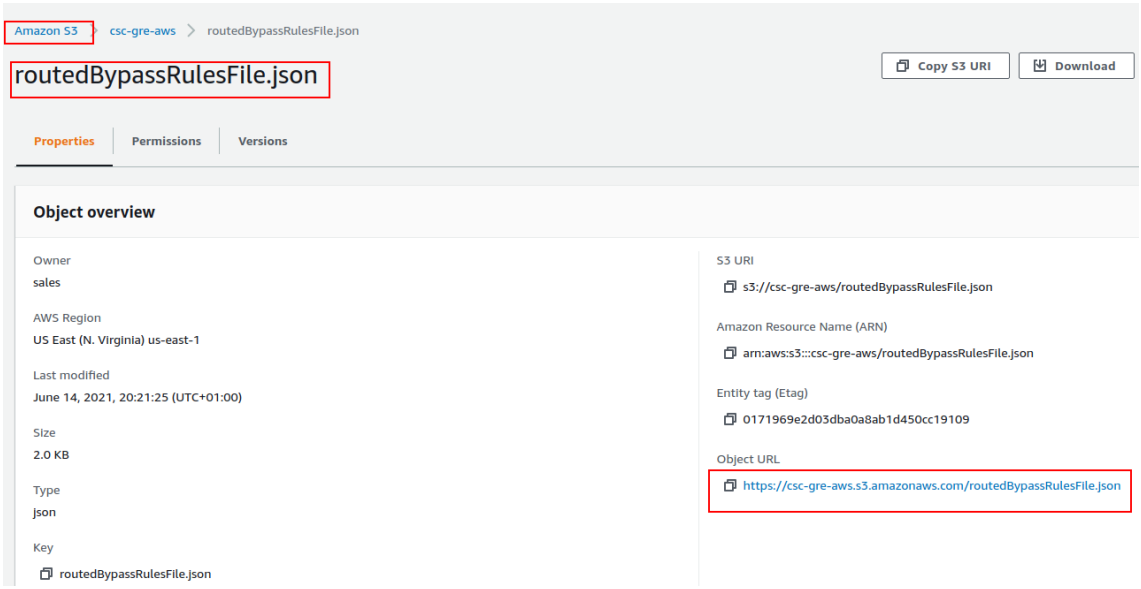
7.1.2.2 Routed Bypasses via your Public IP.

1. Create the routedBypassRulesFile.json (See Section 5.4) and place it on a AWS bucket. Get the URL of the bucket. On the routedBypassRulesFile.json you can create rules per source/destination IP/Subnet and destination port. (File example at: https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json)

Example of a Routed Bypass Rule:

```
{
   "description": "O365 Login URLs 1",
   "ipProtocol": "tcp",
   "sourceCirdIp": "0.0.0.0/0",
   "destinationCirdIp": "20.190.128.0/18",
   "fromPort": "80",
   "toPort": "80"
},
```

Example AWS Bucket:



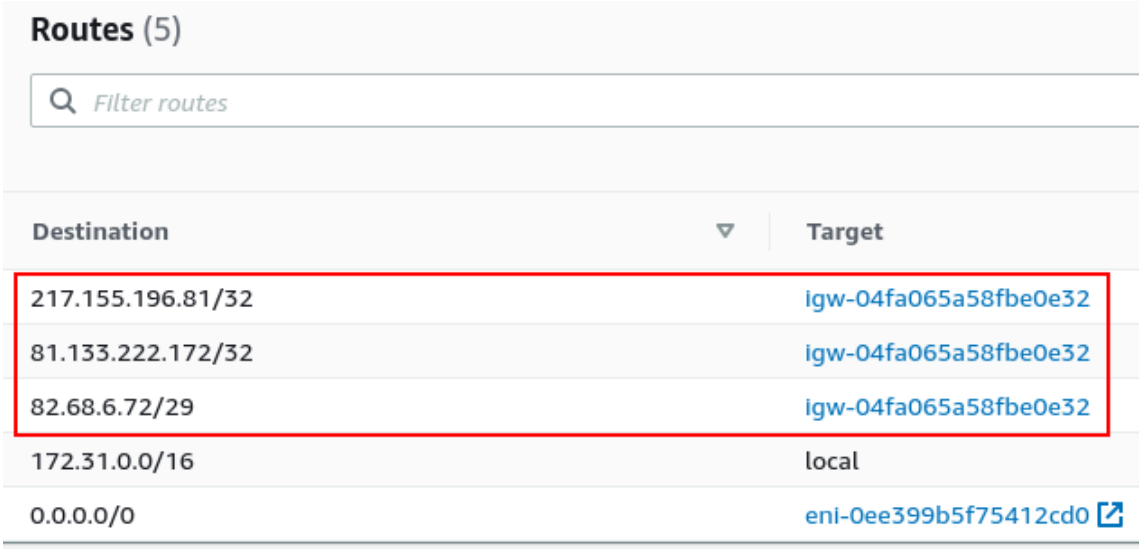
2. Configure the Routed Bypass URL (Object URL of AWS bucket) on each CSC and Refresh the Routed Bypass List.

7.1.2.3 Other routes and AWS Endpoints.

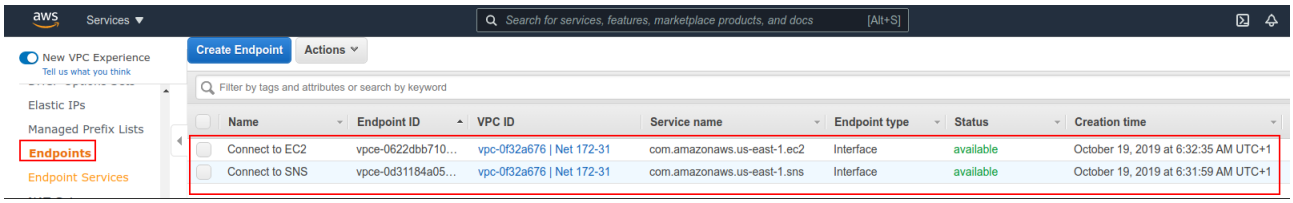
When changing the default route via Zscaler, you can lose connectivity to specific internal hosts (i.e. bastion hosts) and AWS Endpoints.

Please, review these services and create the appropriate configuration for them.

Example: Return Route for Bastion Hosts:



Example of AWS Endpoints:



Note: this Endpoints are required for the CSC's HA service.

7.1.3 Traffic Redirection - PAC files.

You can use both Traffic Redirection method at the same time: Routed and Proxied. This section will explain the Proxy process to send traffic to Zscaler and do Proxy Bypasses via your Public IP.

The steps are:

1. Obtain the CSC VIP proxy IP and Bypass Proxy IP of each CSC.
2. Get your list of URLs to send direct via your Public IP and create the Bypass PAC file for your CSCs. Configure Proxy Bypass on each CSC.
3. Create the PAC file to use on your devices.

7.1.3.1 Obtain the CSC VIP proxy IP and Bypass Proxy IP of each CSC.

From "Show Configuration and Status" menu:

```
TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 172.31.200.33:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 172.31.200.224:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP
```

```
TRAFFIC REDIRECTION Options
To Zscaler: VIP Proxy: 172.31.202.248:80 (or :9400) | Route all traffic via CSC GW IP | Zscaler Global Proxies (port 80/9400) via CSC GW IP
Direct to Internet: Bypass Proxy: 172.31.202.99:3128 | Zscaler Global Proxies (port 3128) via CSC GW IP
```

CSC Name	VIP Proxy	Bypass Proxy
csc-gre-v-3-0l-a	172.31.200.33:80	172.31.200.224:3128
csc-gre-v-3-0l-b	172.31.202.248:80	172.31.202.99:3128

7.1.3.2 Create the Proxy Bypass PAC to configure on the PAC URL on the CSCs.

Create a Proxy Bypass PAC on your Zscaler console. Obtain the PAC URL and configure Proxy Bypass functionality on each CSC. (See section 5.3 Proxy Bypass)

Example of Proxy Bypass PAC:

PAC FILE CONTENTS

```
function FindProxyForURL(url, host) {
        var bypassproxy="PROXY 1.1.1.1:3128; PROXY 2.2.2.2:3128";

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

    // Bypass via CSC Public IPs (Examples)
        // Okta Domains (for Location Rules)
    if ((shExpMatch(host, "*.okta.com")) ||
        (shExpMatch(host, "*.oktacdn.com")) ||
        (shExpMatch(host, "*.okta-emea.com")) ||
        (shExpMatch(host, "login.mydomain.com")) ||
        // O365 Domains for ConditionalAccess
        (shExpMatch(host, "login.microsoftonline.com")) ||
        (shExpMatch(host, "login.microsoft.com")) ||
        (shExpMatch(host, "login.windows.net")) ||
        // IP / Port test page
        (shExpMatch(host, "portquiz.net"))) {
        return bypassproxy
    }
// ======================================================
    return bypassproxy
}
```

CSC's Proxy Bypass URL example:

http://pac.zscalerthree.net/maidenheadbridge.com/bypass.pac

Proxy PAC in Support Portal:
https://maidenheadbridge.freshdesk.com/a/solutions/articles/33000264777

Example of Proxy Bypass PAC content:

```
function FindProxyForURL(url, host) {
      var bypassproxy="PROXY 1.1.1.1:3128; PROXY 2.2.2.2:3128";

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

   // Bypass via CSC Public IPs (Examples)
      // Okta Domains (for Location Rules)
   if ((shExpMatch(host, "*.okta.com")) ||
      (shExpMatch(host, "*.oktacdn.com")) ||
      (shExpMatch(host, "*.okta-emea.com")) ||
      (shExpMatch(host, "login.mydomain.com")) ||
      // O365 Domains for ConditionalAccess
      (shExpMatch(host, "login.microsoftonline.com")) ||
      (shExpMatch(host, "login.microsoft.com")) ||
      (shExpMatch(host, "login.windows.net")) ||
      // IP / Port test page
      (shExpMatch(host, "portquiz.net"))) {
      return bypassproxy
   }
// ======================================================
   return bypassproxy
}
```

After creating the Bypass Proxy PAC file, go to CSC's menu and configure the Proxy Bypass URL on each CSC and refresh the Proxy Bypass List. See section 5.3 for more details.

7.1.3.3 Create the PAC file for your devices.

In this example, the PAC file for your devices is divided into sections:

1. Section 1 contains the standard values of any Zscaler PAC file.
2. Section 2 contains the assignment of IPs for the variables "tozscaler" and "bypassproxy". Section 2 shows a simple way to do a Load Balance per Source IP (odd/even) to use both CSC simultaneously for Web Traffic. In Section 2, you need to replace csc1vip, csc1bypass, csc2vip and csc2bypass for the IP values taken from the CSC console.
3. Section 3 contains the list of Domains to proxy bypass.
4. Section 4 contains the line for default traffic to Zscaler.

```
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: Load Balancing: 2 x Cloud Security Connectors 1 Gbps
   // AWS to Zscaler: 2 Gbps

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

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

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

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

   // Bypass via CSC Public IPs (Examples)
   // Okta Domains (for Location Rules)
   if ((shExpMatch(host, "*.okta.com")) ||
      (shExpMatch(host, "*.oktacdn.com")) ||
      (shExpMatch(host, "*.okta-emea.com")) ||
      (shExpMatch(host, "login.mydomain.com")) ||
      // O365 Domains for ConditionalAccess
      (shExpMatch(host, "login.microsoftonline.com")) ||
      (shExpMatch(host, "login.microsoft.com")) ||
      (shExpMatch(host, "login.windows.net")) ||
      // IP / Port test page
      (shExpMatch(host, "portquiz.net"))) {
      return bypassproxy
   }
   // ======================================================
   // Section 4: Default Traffic

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

8 Appendix B – AWS Systems Manager “Run Commands” to monitor the CSC.

The easiest and cheapest 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/.

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.

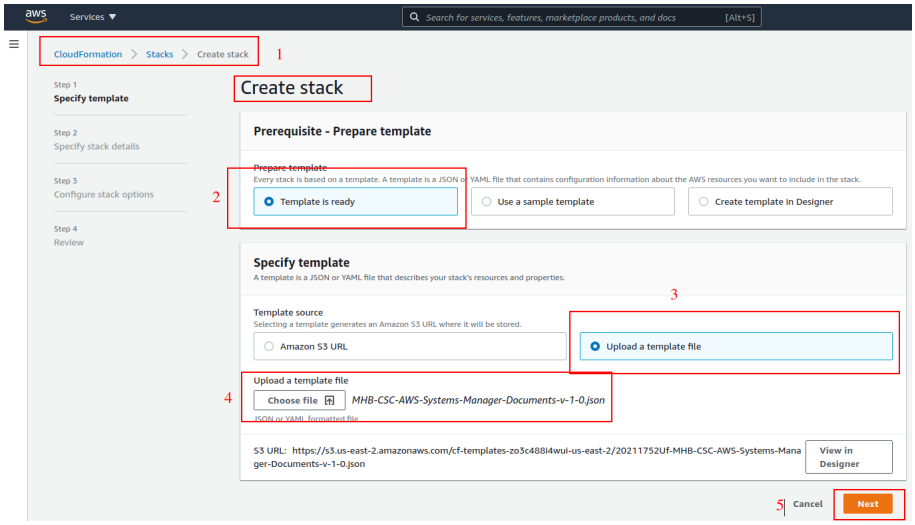
This section explains how to create the "Documents" and how to "Run Commands".

8.1 AWS Systems Manager: Create Documents

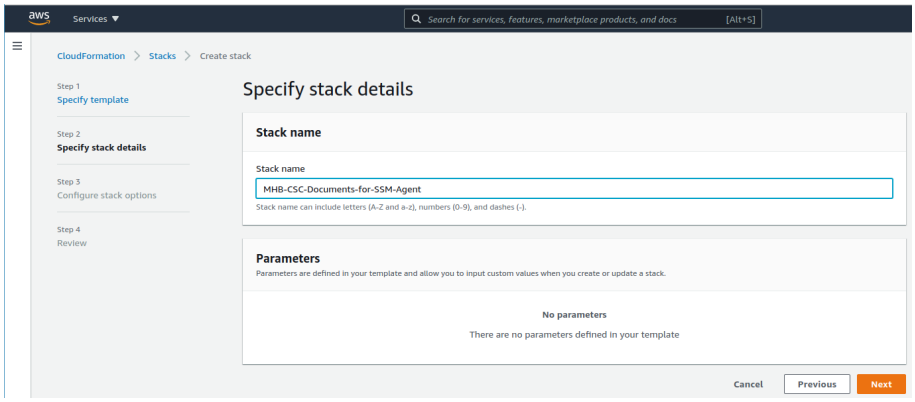
We provide a CloudFormation template to create all "Documents" in one shot.

Steps:

1. Download the CloudFormation template from: https://maidenheadbridge.freshdesk.com/support/solutions/folders/33000214143
2. Deploy Stack. Go to Cloudformation →Create Stack

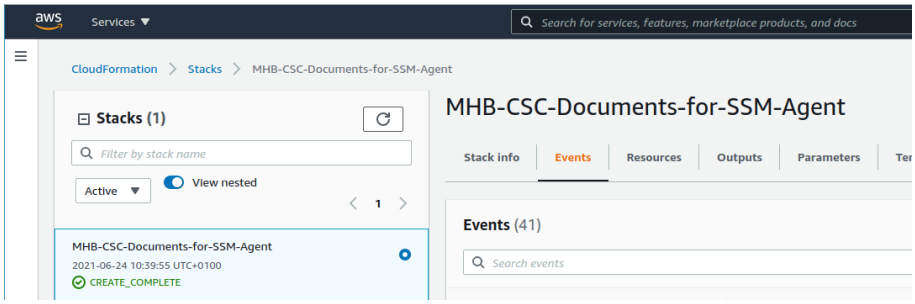


Click "Next" and put a name to the Stack.

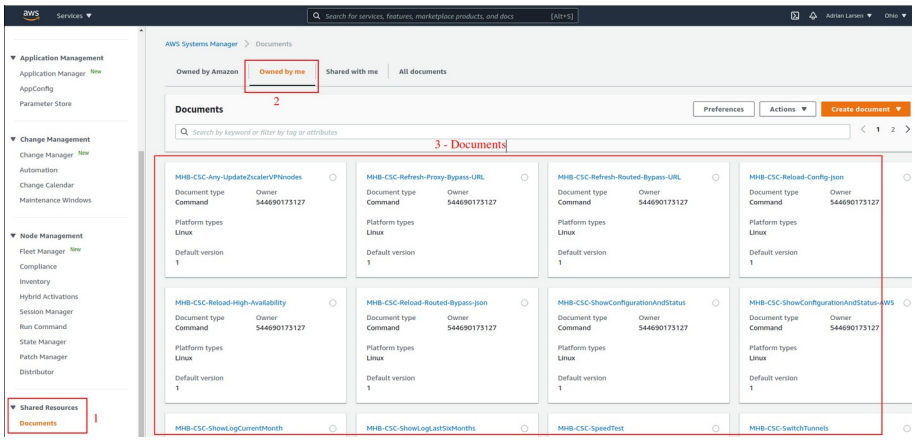


Click "Next", "Next" and "Create Stack".

Wait for the Stack to complete.



3. Check the Documents created on AWS Systems Manager. Go to Systems Manager.



8.2 Run Commands

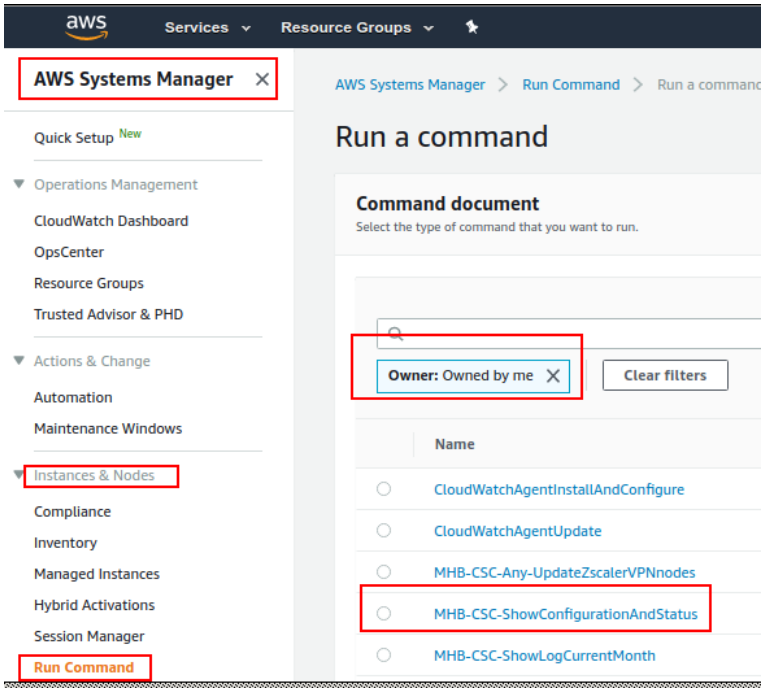
After you created the Documents, you are ready to Run Commands on the CSC.

You can see the results of the operation on the “Output” section or to store the results on a S3 Buckets for further inspection.

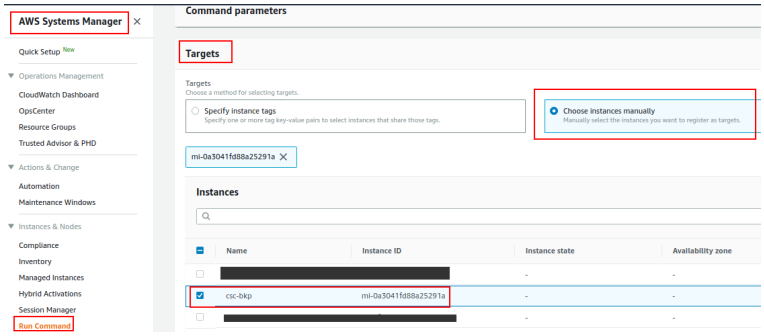
To Run Commands go to: AWS Systems Manager → Instances & Nodes → Run Command

Here an example of Running: MHB-CSC-ShowConfigurationAndStatus

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

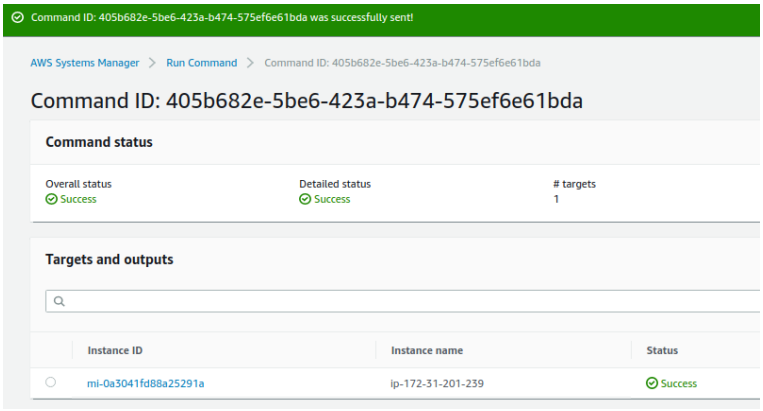


3. Scroll down and Select the Instances
4. We are selecting only one instance, but you can select as much as you want.

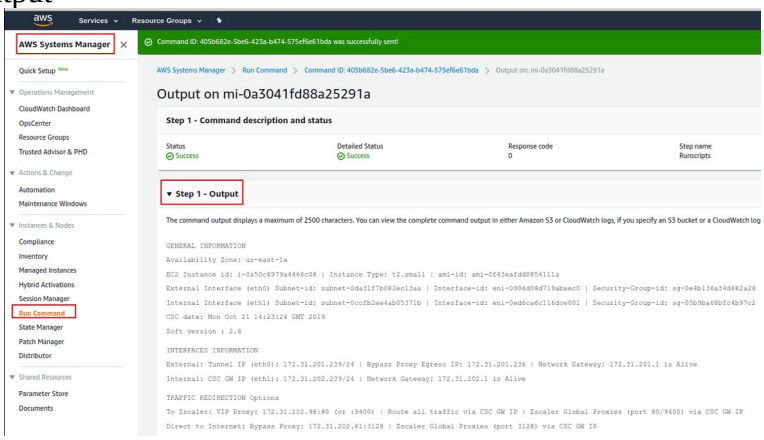


5. Click Run

Next Screen is:



6. Click “Instance ID” (mi-0a3041fd88a25291a)
7. Expand “Output”



8.3 List of Documents available for "Run Command"

1. "MHB-CSC-ShowConfigurationAndStatus-AWS": Executes "Show Configuration and Status" test on CSCs for AWS with version 2.8 or below.
2. "MHB-CSC-ShowConfigurationAndStatus": Same as above, for all CSC all platforms.
3. "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.
4. "MHB-CSC-Refresh-Routed-Bypass-URL": Refresh the Routed Bypass list using the values of the JSON file stored in the URL configured.
5. "MHB-CSC-SpeedTest": Performs speedtest.net on the CSC.
6. "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.
7. "MHB-CSC-ShowLogCurrentMonth": Shows current month logs.
8. "MHB-CSC-ShowLogLastSixMonths": Shows last six month logs.
9. "MHB-CSC-SwitchTunnels": Switch tunnels.
10. "MHB-CSC-Reload-Config-json": Reloads the values of config.json file.
11. "MHB-CSC-Reload-High-Availability": Reloads the values of highAvailability.json file.
12. "MHB-CSC-Reload-Routed-Bypass-json": Reloads the values of routedBypassRulesFile.json.
13. "MHB-CSC-Any-UpdateZscalerVPNnodes": Updates the VPN node database on IPsec models. Not in use on the CSC for AWS.

9 Appendix C: JSON Files examples.

In this section, you have an example for the JSON files in use. JSON files are located at:

```
ls /usr/local/etc/mhb-csc
config.json  configUserData.json  highAvailability.json  routedBypassRulesFile.json
```

9.1 configUserData.json

configUserData.json is used at first boot when the integration with the Zscaler API is done.

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

```
{
  "model": "csc-gre-aws",
  "version": "1.0",
  "cloudName": "zscalerthree",
  "apiTokenID": "57C3B00DC671076CE42B34B764BEE178",
  "dns": {
    "useCloudDNS": true,
    "primaryDnsIP": "",
    "secondaryDnsIP": ""
  },
  "bypassProxyPacUrl": "http://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac",
  "syslogServers": {
    "primarySyslogIP": "172.31.200.163",
    "secondarySyslogIP": "",
    "syslogTcpPort": 514
  },
  "ssmAgent": {
    "activationCode": "CJzTwPN2Bgs3ksjinLr2",
    "activationID": "fef65652-8cf0-4ac7-89a2-4bcf35b0f77f",
    "awsRegion": "us-east-1"
  },
  "tunnelRedundancy": {
    "returnToPrimaryTunnel": false
  },
  "nodeSelection": {
    "withinCountryPreferred": true
  },
  "location": {
    "name": "aws-3-0-j-1",
    "country": "UNITED_STATES",
    "tz": "UNITED_STATES_AMERICA_NEW_YORK",
    "ipAddresses": [
      "auto"
    ],
    "authRequired": true,
    "xffForwardEnabled": false,
    "surrogateIP": true,
    "idleTimeInMinutes": 480,
    "displayTimeUnit": "MINUTE",
    "surrogateIPEnforcedForKnownBrowsers": false,
    "surrogateRefreshTimeInMinutes": 120,
    "surrogateRefreshTimeUnit": "MINUTE",
    "ofwEnabled": true,
    "ipsControl": true
  },
  "routedBypassJsonFileUrl": "https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json",
  "routedBypassRules": []
}
```

9.2 config.json

config.json is used when changes are done via AWS SSM agent. After changing this file, you need to use the Document: "MHB-CSC-Reload-Config-json".

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

```
{
  "model": "csc-gre-aws",
  "version": "1.0",
  "cloudName": "zscalerthree",
  "dns": {
    "useCloudDNS": true,
    "primaryDnsIP": "",
    "secondaryDnsIP": ""
  },
  "bypassProxyPacUrl": "http://pac.zscalerthree.net/RdwNltSPqBFN/az-csc-bypass.pac",
  "syslogServers": {
    "primarySyslogIP": "172.31.200.163",
    "secondarySyslogIP": "",
    "syslogTcpPort": 514
  },
  "tunnelRedundancy": {
    "returnToPrimaryTunnel": false
  },
  "routedBypassJsonFileUrl": "https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json",
  "greCredentials": {
    "grePublicIP": "54.159.82.127",
    "primaryDestination": "165.225.8.30",
    "secondaryDestination": "165.225.38.51",
    "firstIpPrimaryDestinationRange": "172.17.152.96"
  }
}
```

9.3 highAvailability.json

highAvailability.json is used when changes are done via AWS SSM agent. After changing this file, you need to use the Document: "MHB-CSC-Reload-High-Availability".

```
{
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role",
    "haSnsMessageArn": "arn:aws:sns:us-east-1:544690173127:AWS-Zscaler-Notification",
    "haInstanceIdOtherCsc": "i-08f80bd2952ff13ed",
    "haVPC": "vpc-0f32a676",
    "haRouteTables": [
      {
        "routeTableId": "rtb-d090c8a8"
      }
    ]
  }
}
```

9.4 routedBypassRulesFile.json

highAvailability.json is used when changes are done via AWS SSM agent. After changing this file, you need to use the Document: "MHB-CSC-Reload-Routed-Bypass-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"
    }
  ]
}
```

10 Appendix D: Release Notes

10.1 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.

10.2 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.

10.3 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.

10.4 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.