



Maidenhead Bridge



Cloud Security Connector GRE – AWS

with Private Cloud Private Access

(For Amazon Web Services – AWS)

Version 1.0

January 2022

Table of Contents

1 Introduction to Cloud Security Connectors for Netskope.....	6
2 Key benefits of the Cloud Security Connector GRE.....	6
3 Network Diagrams.....	8
3.1 CSC GRE for AWS – Single deployment.....	8
3.2 CSC GRE for AWS – High Availability Deployment.....	9
3.3 CSC GRE for AWS – High Availability when using Single Exit to Internet.....	10
3.4 Steering: Routing and Proxying all together.....	11
3.5 Private Cloud Private Access (PriCPA).....	12
4 Creating the Cloud Security Connector for AWS.....	13
4.1 Basic Mode deployment.....	14
4.1.1 Prerequisites.....	14
4.1.2 Prerequisites EXAMPLE:.....	14
4.1.3 Launching the CSC from AWS Marketplace.....	15
4.1.4 Launching the CSC using the CloudFormation template directly.....	15
4.1.5 Accessing for first time to your CSC.....	18
4.2 Adding your Public IP to the Netskope console.....	19
4.3 Advanced Mode Deployment (filling User Data).....	20
4.3.1 Prerequisites.....	20
4.3.1.1 configUserData.json file fields values.....	21
4.3.1.1.1 File Header – Do not change.....	21
4.3.1.1.2 "dns":.....	21
4.3.1.1.3 "syslogServers":.....	21
4.3.1.1.4 "ssmAgent":.....	21
4.3.1.1.5 "greCredentials":.....	22
4.3.1.1.6 "tunnelRedundancy":.....	22
4.3.1.1.7 "bypassProxyPacUrl":.....	23
4.3.1.1.8 "routedBypassJsonFileUrl":.....	23
4.3.2 Filling "User Data" on the CloudFormation Template.....	24
4.3.3 Log information when using Advanced Deployment.....	25
5 Resources creates by the CloudFormation template.....	26
6 The Cloud Security Connector Admin Console:.....	28
6.1 Monitoring Tasks.....	30
6.1.1 Show Configuration and Status.....	30
6.1.1.1 GENERAL INFORMATION.....	31
6.1.1.2 INTERFACES INFORMATION.....	31
6.1.1.3 TRAFFIC REDIRECTION Options.....	31
6.1.1.4 ELASTIC (PUBLIC) IPs INFORMATION.....	32
6.1.1.5 DNS INFORMATION.....	32
6.1.1.6 NETSKOPE INFORMATION.....	32
6.1.1.7 TUNNEL STATUS.....	33
6.1.1.8 HTTP://WWW.NOTSKOPE.COM PAGE STATUS.....	33
6.1.1.9 PROXY BYPASS - EGRESS INTERFACE STATUS.....	33

6.1.1.10 ROUTED BYPASS.....	34
6.1.1.11 AWS SSM AGENT.....	34
6.1.1.12 SYSLOG/SIEM Servers Information.....	34
6.1.1.13 HIGH AVAILABILITY Information.....	34
6.1.2 Show Interfaces Traffic.....	35
6.1.3 Traceroute and Latency Test.....	35
6.1.4 SPEED TEST.....	36
6.2 CSC Admin Tasks.....	37
6.2.1 AWS SSM Agent (Register or De-Register).....	37
6.2.1.1 Create a "Hybrid Activation" from AWS console.....	37
6.2.1.2 Register the CSC.....	38
6.2.1.3 View the Registered CSC on AWS Systems Manager.....	38
6.2.2 Manage Administrators.....	39
6.2.2.1 Managing the SSH Key of a User.....	39
6.2.3 Change Timezone.....	39
6.3 Proxy Bypass.....	41
6.3.1 Proxy Bypass - Traffic Flow.....	41
6.3.2 View Current Proxy Bypass List.....	41
6.3.3 Configure Proxy Bypass List.....	41
6.3.3.1 Auto - Proxy Bypass PAC URL.....	42
6.3.3.2 Manual Proxy Bypass Configuration.....	43
6.4 Routed Bypass.....	45
6.4.1 Routed Bypass - Traffic Flow.....	45
6.4.2 View Current Routed Bypass List.....	45
6.4.2.1 Compact.....	45
6.4.2.2 Json.....	46
6.4.3 Configure Routed Bypass List.....	47
6.4.3.1 Routed Bypass URL.....	47
6.4.3.2 Manual (Paste Routed Bypass JSON file).....	48
6.5 Log Information.....	49
6.5.1 View Current Month.....	49
6.5.2 View Last 6 Months.....	49
6.6 Configuration Wizards.....	50
6.6.1 Change Nodes, DNS servers, Syslog and more.....	50
6.6.1.1 Running the Configuration Wizard.....	51
6.6.2 Switch Tunnels - Primary / Secondary.....	54
6.6.3 Update Netskope Nodes Databases.....	54
6.6.4 High Availability configuration.....	55
6.6.4.1 High Availability configuration on detail.....	57
6.6.4.1.1 Deploy a pair of CSC on the different availability zones.....	57
6.6.4.1.2 Create an IAM role with the following policies.....	58
6.6.4.1.3 Get the 'Route Table ID' of the Route Table/s where there is Default Route (0.0.0.0/0) to Internet.....	59
6.6.4.1.4 Create "Endpoints" to AWS services (EC2, SNS, S3, etc.).....	60

6.6.4.1.5 Create SNS message for Alerts.....	61
6.6.4.1.6 Run the HA Wizard on the First CSC.....	61
6.6.4.1.7 Configure the second CSC on the HA pair.....	62
6.6.4.1.8 Checking HA Status.....	63
6.6.4.1.9 Notifications from CSC on HA.....	63
7 Steering traffic to NewEdge with the CSC GRE for AWS.....	64
7.1 CSC on HA Pair.....	64
7.1.1 Network Diagram.....	64
7.1.2 Prerequisites.....	65
7.1.3 Routing traffic via the CSC HA pair.....	65
7.1.3.1 Traffic to Netskope.....	65
7.1.3.2 "Routed Bypass" traffic.....	65
7.1.4 Proxy traffic via the CSC HA Pair.....	65
7.1.4.1 Using PAC files.....	65
7.1.4.1.1 PAC file for Load Balancing.....	65
7.1.4.1.2 PAC file using Netskope's Global Proxy.....	67
7.1.4.1.3 PAC file using CSC's VIP and Bypass Proxy Address (Pri/Sec).....	68
7.1.4.2 Using Explicit Proxy on devices that cannot support PAC files.....	69
7.2 CSC Single.....	70
7.2.1 Network Diagram.....	70
7.2.2 Prerequisites.....	70
7.2.3 Routing traffic via the CSC Single.....	71
7.2.3.1 Traffic to Netskope.....	71
7.2.3.2 "Routed Bypass" traffic.....	71
7.2.4 Proxy traffic via the CSC Single.....	71
7.2.4.1 Using PAC files.....	71
7.2.4.1.1 PAC file using Netskope's Global Proxy.....	72
7.2.4.1.2 PAC file using CSC's VIP and Bypass Proxy Address (Pri/Sec).....	73
7.2.4.2 Using Explicit Proxy on devices that cannot support PAC files.....	74
7.3 Testing traffic to Netskope.....	75
7.3.1 www.notskope.com.....	75
7.3.2 https://ip.maidenheadbridge.com.....	76
7.3.3 SpeedTest.....	77
8 Private Cloud Private Access.....	78
8.1 What is Private Cloud Private Access (PriCPA)?.....	78
8.2 PriCPA Network Diagrams.....	78
8.2.1 High Level Network Diagram.....	78
8.2.2 Low Level Network Diagram – PriCPA only.....	79
8.3 Configuring PriCPA.....	80
8.3.1 Create the Local configuration (first node of the cluster).....	81
8.3.2 Create the Local configuration (second node of HA Pair).....	83
8.3.3 Create the Private Access Peers JSON file.....	84
8.3.3.1 Full mesh Private Access Peers JSON file.....	84
8.3.3.2 Understanding "privateApps" configuration and values.....	89



8.3.3.3 Example of "privateApps" for a Windows Domain controller.....	91
8.3.3.4 Example of "privateApps" for Internal Web Server.....	91
8.3.4 Load the "Private Access Peers JSON file" to the CSCs.....	92
8.3.4.1 Using "Private Access Peers URL".....	92
8.3.4.2 <i>Manual: Copy and Paste "Private Access Peers Json file"</i>	98
8.4 Show Configurations and Status Private Access.....	99
8.4.1 Via SSH console.....	99
8.4.1.1 Show Peer/s Status.....	99
8.4.1.2 Show Peers Json file (active).....	100
8.4.1.3 Show Local Configuration.....	101
8.4.1.4 Show Firewall Local Rules.....	101
8.4.2 via AWS Systems Manager or Rundeck.....	102
8.4.2.1 AWS Systems Manager.....	102
8.4.2.2 Rundeck.....	102
8.5 Configure CSC Remote Management via Private Access.....	103
9 Remote Management using AWS and Rundeck.....	104
9.1 AWS Systems Manager.....	104
9.1.1 Create Documents.....	104
9.1.2 Run Commands.....	106
9.1.3 List of Documents available for "Run Command".....	109
9.2 Rundeck.....	110
9.2.1 Jobs.....	111
9.2.2 Running job "Show Configuration and Status".....	111
10 DevOps operations.....	112
10.1 config.json file.....	113
10.2 routedBypassRulesFile.json.....	114
10.3 privateAccessPeersConfig.json.....	116
10.4 highAvailability.json file.....	118
11 Appendixes.....	119
11.1 Appendix A: Routed Bypass JSON file if you don't have Cloud Firewall License.....	119
11.2 Appendix B: Release Notes.....	120
11.2.1 Version 1.0.....	120
11.3 Appendix C: JSON formatters (Visual Code, Notepad++).....	121
11.3.1 Visual Code.....	121
11.3.2 Notepad ++.....	122
11.4 Appendix D: Securing an AWS Bucket by source IP.....	124

1 Introduction to Cloud Security Connectors for Netskope.

The Cloud Security Connector (CSC) is a device that enables easy deployments of the Netskope SASE solution in any customer environment.

The Cloud Security Connector (CSC) for AWS is an EC2 instance that connects internal AWS resources to Netskope NewEdge.

The CSC for AWS lets you connect securely to Netskope NewEdge up to 1 Gbps without hassle.

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

After launching the CSC from the AWS Marketplace using the CloudFormation template provided, the CSC will automatically select the best Netskope Edge nodes and do the GRE tunnels Primary and Secondary.

The CSC GRE contains the perfect configuration for GRE tunnels, firewall rules and routing tables that are necessary.

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

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

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

Simple to install with complete management using Amazon Systems Manager, Rundeck (or similar, like Ansible, Salt, Etc.) and SSH.

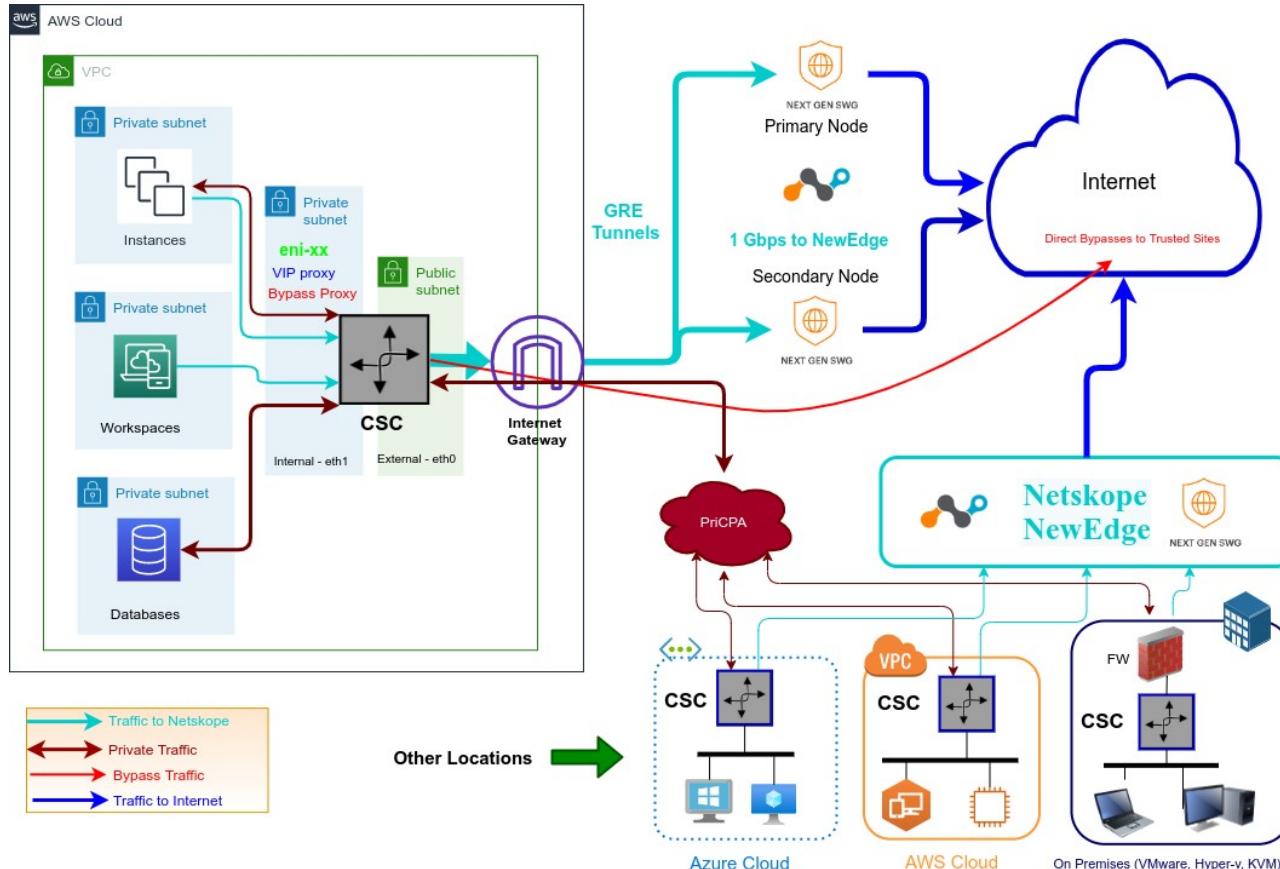
2 Key benefits of the Cloud Security Connector GRE

- No Networking knowledge is required.
- Automated deployment using Cloudformation or your tool of choice. (i.e. Terraform)
- Enables any Location to be connected to Netskope NewEdge up to 1 Gbps.
- With Private Cloud Private Access you can connect all sites securely on a Zero Trust model. The CSC secures your Private Traffic between your physical and cloud locations.
- The CSC comes with the optimal values to work with Netskope NewEdge.
- Full tunnel redundancy.

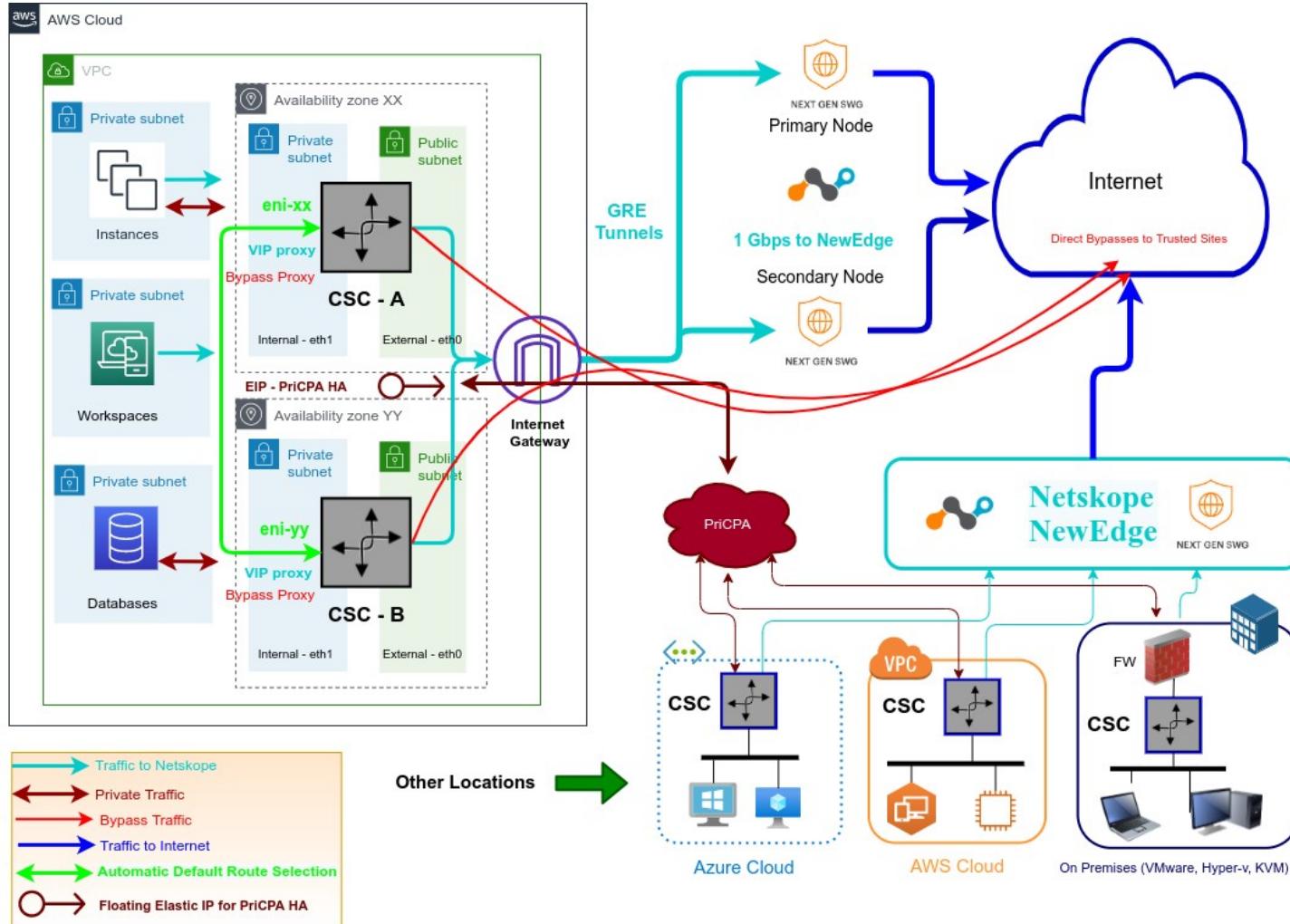
- High Availability with automatic default route to Internet selection on multiple routes.
- All traffic steering options supported:
 - Route all traffic to Netskope (or http/s only).
 - Use of PAC files.
 - Use of Explicit Proxy.
 - No default Route scenarios.
- Multiple options to Bypass Traffic:
 - Layer 7 Proxy Bypass to Trusted Web Sites.
 - Layer 4 Routed Bypass: TCP, UDP and ICMP per source/destination Network and Port (UDP/TCP)
- Cloud Firewall and Cloud Web Security.
- Complete visibility of internal IPs on Neskope Console.
- No operational burden for Administrators.
- Full hardened device.
- Multiple tools for testing and troubleshooting included: Speed Test, MTR (MyTraceRoute), Keepalives statuses, Etc.
- Allow the internal communication between your locations with Private Cloud Private Access.
- Management via SSH, AWS Systems Manager, Rundeck or similar. (Ansible, Salt, Etc.)
- It runs on a cheap AWS instance: t2, t3a and t3 instances.

3 Network Diagrams

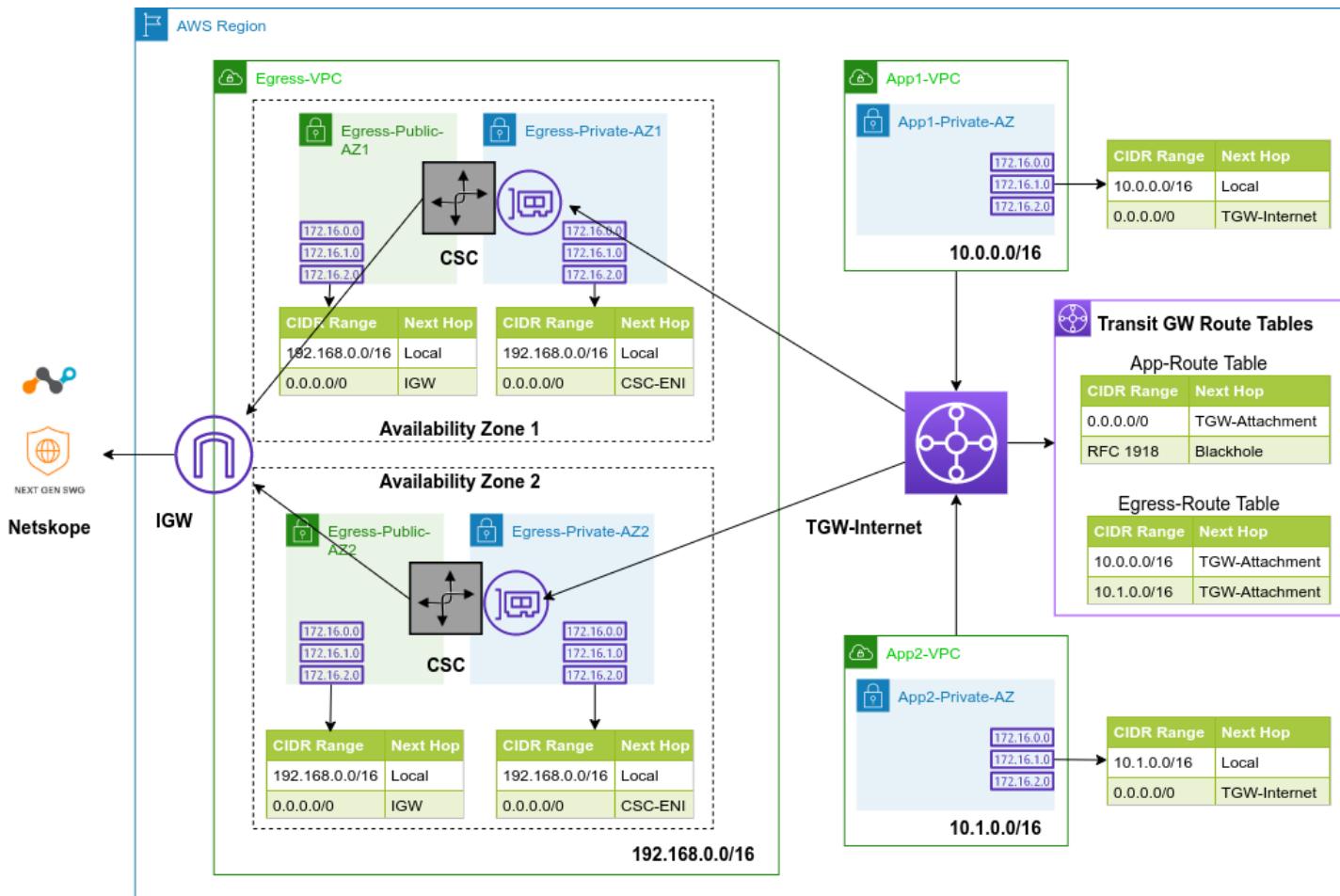
3.1 CSC GRE for AWS – Single deployment



3.2 CSC GRE for AWS – High Availability Deployment



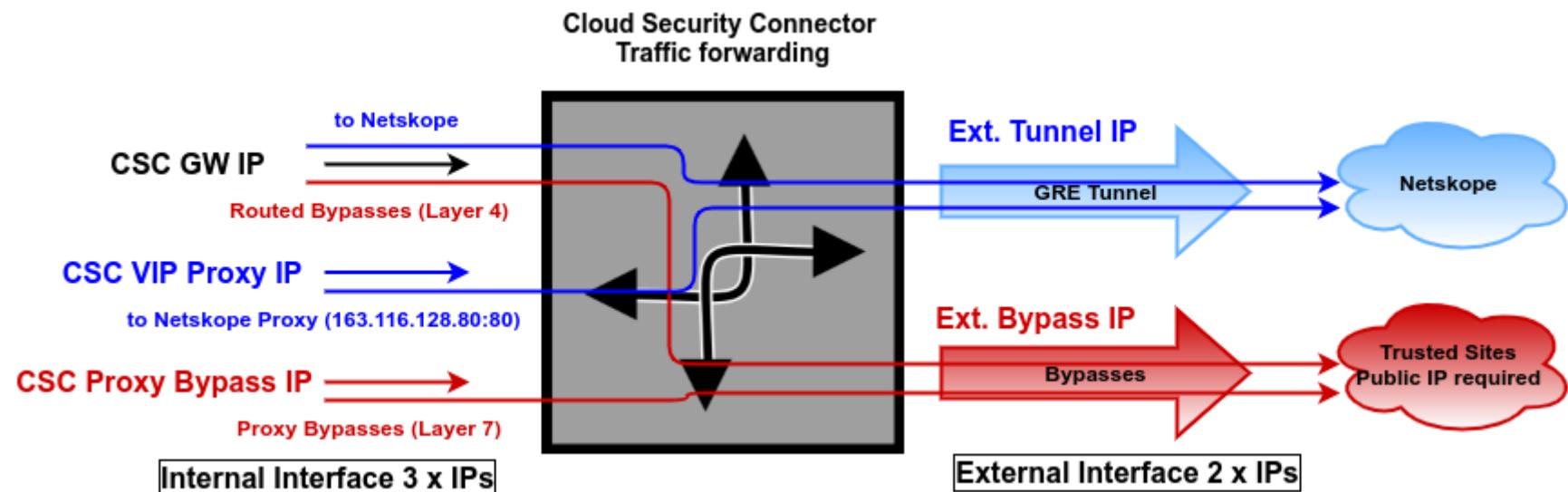
3.3 CSC GRE for AWS – High Availability when using Single Exit to Internet



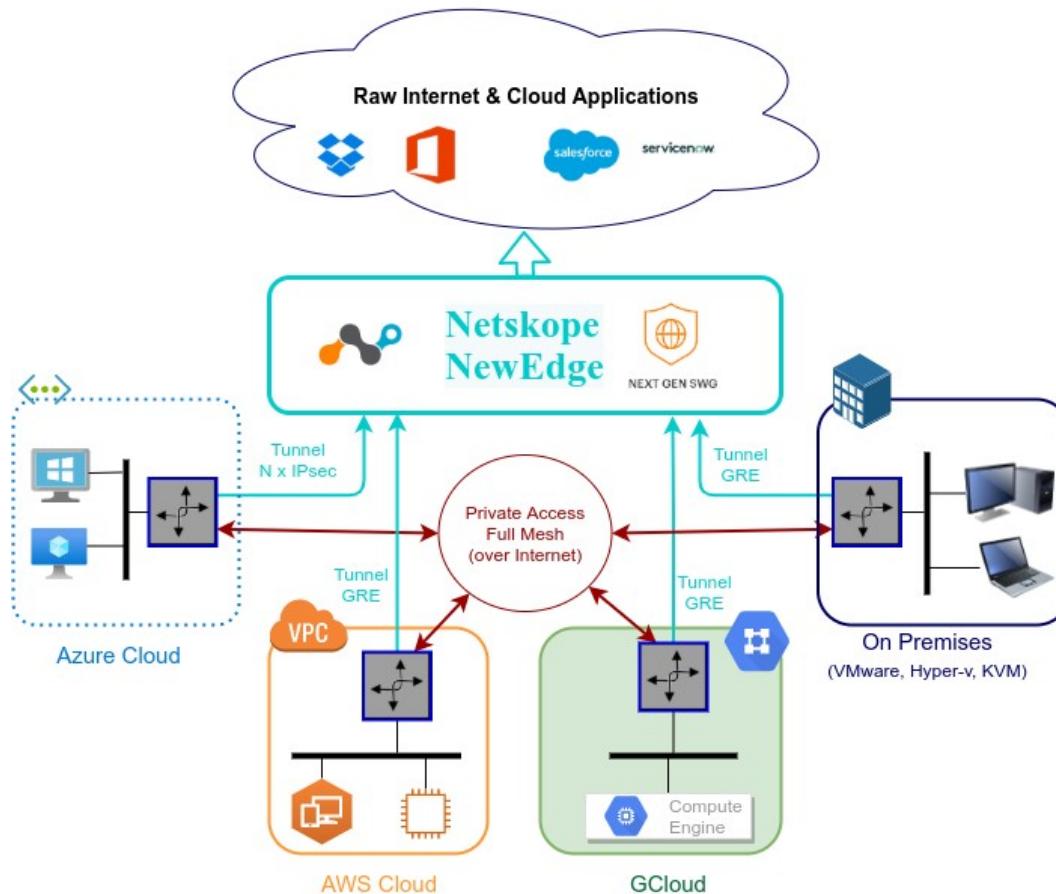
3.4 Steering: Routing and Proxying all together.

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

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



3.5 Private Cloud Private Access (PriCPA)



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

4 Creating the Cloud Security Connector for AWS

There are two ways to deploy the CSC: Basic and Advanced mode (Adding "User Data" on the CloudFormation template.). The difference between these methods are:

1. Basic Mode:

- 1.1. Deploy the CSC via Cloudformation, selecting External and Internal Subnet and no further parametrization (leave Cloudformation field "User Data" empty).
- 1.2. The Cloudformation template will create all AWS resources (Instance, Security Groups, Public IPs, etc.)
- 1.3. After the initial boot, the CSC will select the nearest Primary and Secondary Netskope NewEdge nodes and automatically create the tunnels.
- 1.4. Your next task is to add the Public IP of the GRE tunnels to your Netskope console. (Go to Settings -> Security Cloud Platform -> Traffic Steering -> GRE.) You can obtain the Public IP from AWS or access the CSC via SSH console.

2. Advanced Mode (filling User Data):

- 2.1. Deploy the CSC via Cloudformation, selecting External and Internal Subnet and paste the userConfigData.json file on the "User Data" field.
- 2.2. The Cloudformation template will create all AWS resources (Instance, Security Groups, Public IPs, etc.)
- 2.3. After the initial boot, the CSC will select the nearest Primary and Secondary Netskope NewEdge nodes and automatically create the tunnels.
- 2.4. Your next task is to add the Public IP of the GRE tunnels to your Netskope console. (Go to Settings -> Security Cloud Platform -> Traffic Steering -> GRE.) You can obtain the Public IP from AWS or access the CSC via SSH console.
- 2.5. Bypass Proxy and Routed Bypass rules will be applied from the URLs provided.
- 2.6. AWS SSM Agent registration will be done automatically.

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*

Virtual Private Cloud		Name	VPC ID	State	IPv4 CIDR
Your VPCs		Net 172-31	vpc-of32a676	available	172.31.0.0/16

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

net-172-31-200	subnet-8360ecd9	available	vpc-of32a676 Net 172-31	172.31.200.0/24	232	us-east-1d
Net-172-31-96	subnet-818c0ddb	available	vpc-of32a676 Net 172-31	172.31.96.0/24	233	us-east-1d

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

net-172-31-200	subnet-8360ecd9	available	vpc-of32a676 Net 172-31	172.31.200.0/24	232	us-east-1d
Net-172-31-96	subnet-818c0ddb	available	vpc-of32a676 Net 172-31	172.31.96.0/24	233	us-east-1d

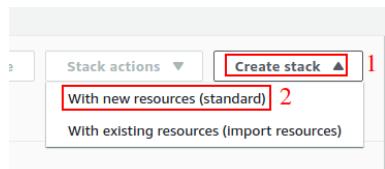
4.1.3 Launching the CSC from AWS Marketplace

1. Search for "Maidenhead Bridge" on AWS Marketplace.
2. Select "**Cloud Security Connector GRE for Netskope**"
3. Click "**Continue to Subscribe**" and accept the EULA.
4. Click "**Continue to Configuration**" and select the Region.
5. Click "**Continue to Launch**" and select "Choose Action" → "Launch Cloudformation".#
6. Click "**Launch**" and Next. You will redirected to the CloudFormation Page.

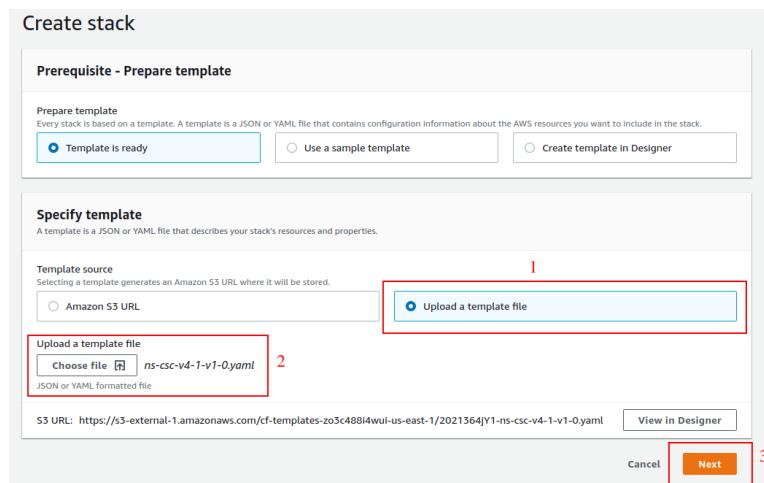
See next section for further steps.

4.1.4 Launching the CSC using the CloudFormation template directly.

1. Download the Cloudformation template using this link ([click here](#)).
2. On you AWS Console, Click "**Services**" and select **CloudFormation**.
3. On the right hand side, click "**Create Stack**" and select "**With new resources (standard)**"



4. Select "**Upload a template file**", then "**Choose file**"



5. Click "**Next**".
6. The page to "**Specify Stack details**" will appear.

Specify stack details

Stack name

Stack name 1
 Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
 Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Network Configuration
 Which VPC should this be deployed to?
 Select a VPC.
 2

External Subnet
 Select an External Subnet (WARNING !! must be the same availability zone than Internal Subnet)

Internal Subnet
 Select an Internal Subnet (WARNING !! must be the same availability zone than External Subnet)

Amazon EC2 Configuration

Name
 The name of the instance
 3

AWS Instance Type
 Select one of the instance types

Key Name
 Key Pair name

User Data
 (Optional) Advanced Deployment: Paste here configUserData.json file content values.
 4

Cancel Previous Next

7. Specify Details. Please insert here your values:

→ **Stack Name**

→ **VPC**

→ **External Subnet** (WARNING !! must be the same availability zone as Internal Subnet)

→ **Internal Subnet** (WARNING !! must be the same availability zone as External 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	Vcpu	Memory	Bandwidth
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 instance price, where t3a.medium is the cheapest and m5zn.xlarge is the more expensive. Some recommendations:

- You can use t3a.medium or t3.medium when the traffic required is less than 1 Gbps.
- Use any instance in Green in all other cases.
- Avoid using t2 instances if possible because of bandwidth constraints.

8. Click "Next"
9. "Options Section": Click "Next"
10. "Review": Click "Create Stack"

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

The screenshot shows the AWS CloudFormation Stacks page. There is one stack listed:

Stack name	Status	Created time	Description
csc-gre-for-netskope-on-aws	CREATE_COMPLETE	2021-12-31 06:57:10 UTC+0000	AWS CloudFormation template for Cloud Security Connector GRE Single for Netskope. Created 2021-12-08 by Maidenhead Bridge

Done! Your CSC is deployed.

4.1.5 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.

Instances (1/1) [Info](#) 1

Search [Clear filters](#)

Instance state = running	X	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/> csc-gre-for-netskope-on-aws	i-0c5989eb0b10cc968	Running	t3a.large	2/2 checks passed	No alarms	us-east-1d	

Instance: i-0c5989eb0b10cc968 (csc-gre-for-netskope-on-aws)

▼ Network Interfaces [Info](#)

Network interfaces (2) [3](#)

Filter network interfaces

Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address
eni-04de6da497ffb7fb8	csc-gre-single-external-interface	-	-	3.233.191.93	172.31.96.190
eni-0fb3dddeac9cc8b1	csc-gre-single-internal-interface 4	-	-	- CSC GW IP ->	172.31.200.250

2. Find the "csc-gre-single-internal-interface" and take a look at the first Private IP address (CSC GW IP). This example is: 172.31.200.250
3. From a machine inside the VPC, ssh the CSC using the Key.

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

In our example, the value is \$ ssh -i us-east-key.pem cscadmin@172.31.200.250

```
ubuntu@ip-172-31-200-163:~$ ssh -i us-east-key.pem cscadmin@172.31.200.250
Maidenhead Bridge
Cloud Security Connector GRE on AWS for Netskope - Admin Console
Please, on your Netskope console, go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> GRE and check 'GRE configurations' to validate that 3.233.191.93 is added.
Did you configure the GRE IP 3.233.191.93 on the Netskope console? Please, confirm.
1) Yes
2) No
Enter your choice: 1
```

4. Your CSC is ready. Your next task is to add the Public IP of the GRE tunnels on your Netskope console. See next section for details.

4.2 Adding your Public IP to the Netskope console

On your Netskope Console go to: Settings -> Security Cloud Platform -> (Traffic Steering) GRE and click "NEW GRE CONFIGURATION"

New GRE Configuration X

Traffic will be steered from your source devices (e.g. router, firewall) to Netskope points of presence (POPs).

CONFIGURATION NAME * 1
csc-gre-for-netskope-on-aws

TUNNEL TYPE *
Default

SOURCE PEER * 2
3.233.191.93

i Remember to configure the tunnel on your peer device using the Netskope POPs information to complete the tunnel configuration.

CANCEL SAVE AND VIEW POPs 3 SAVE

1. Put a Name for the GRE configuration in "Configuration Name"
2. Put your Public IP in "Source Peer"
3. Save.

Your new configuration will appear on the GRE Menu and after a while the Keepalive Status will show "Seen".

GRE Configurations						
8 CREATED		Sort by: Name				
NAME		SOURCE IDENTITY	NETSKOPE POP	USER TRAFFIC STATUS	USER TRAFFIC LAST UPDATED	KEEPALIVE LAST UPDATED
<input type="checkbox"/>	csc-gre-for-netskope-on-aws	3.233.191.93	IAD2 - Washington, DC, US	● Not Seen	12-31-2021 7:58:46 AM	12-31-2021 7:58:46 AM ● Seen

4.3 Advanced Mode Deployment (filling User Data)

In Advanced Mode Deployment, you can pass values of configuration to the CSC directly, like: Proxy Bypass URL, Routed Bypass URL, Netskope Nodes, DNS, AWS SSM Agent registration, etc.

4.3.1 Prerequisites

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

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

configUserData.json

```
{  
  "model": "csc-gre-ns-aws",  
  "type": "configUserData",  
  "version": "1.0",  
  "dns": {  
    "useCloudDNS": true,  
    "primaryDnsIP": "",  
    "secondaryDnsIP": ""  
  },  
  "syslogServers": {  
    "primarySyslogIP": "",  
    "secondarySyslogIP": "",  
    "syslogTcpPort": 514  
  },  
  "ssmAgent": {  
    "activationCode": "",  
    "activationID": "",  
    "awsRegion": ""  
  },  
  "greCredentials": {  
    "autoDiscovery": true,  
    "grePublicIP": "",  
    "primaryGreGateway": "",  
    "primaryProbeIp": "",  
    "primaryLocation": "",  
    "secondaryGreGateway": "",  
    "secondaryProbeIp": "",  
    "secondaryLocation": ""  
  },  
  "tunnelRedundancy": {  
    "returnToPrimaryTunnel": true  
  },  
  "bypassProxyPacUrl": "",  
  "routedBypassJsonFileUrl": ""  
}
```

4.3.1.1 configUserData.json file fields values

Note: We recommend using Visual Studio Code to validate the integrity of JSON files.

4.3.1.1.1 File Header – Do not change.

```
"model": "csc-gre-ns-aws",
"type": "configUserData",
"version": "1.0",
```

4.3.1.1.2 "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 DNS servers. In this case, you must fill the "primaryDnsIP" and "secondaryDnsIP" values.

4.3.1.1.3 "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.

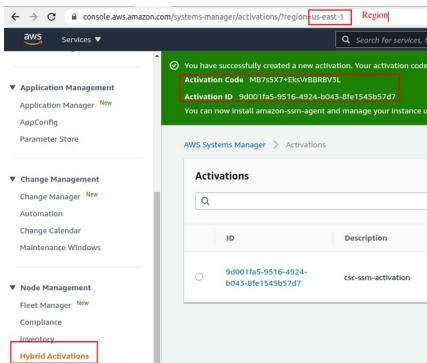
Leave these fields blank if you don't want to configure Syslog Servers.

4.3.1.1.4 "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.



4.3.1.1.5 "greCredentials":

```
"greCredentials": {
    "autoDiscovery": true,
    "grePublicIP": "",
    "primaryGreGateway": "",
    "primaryProbelpAddress": "",
    "primaryLocation": "",
    "secondaryGreGateway": "",
    "secondaryProbelpAddress": "",
    "secondaryLocation": ""
},
```

Select "autoDiscovery": true, for automatic discovery and setup of the nearest Netskope NewEdge nodes (Primary and Secondary)

Select "autoDiscovery": false for Manual input of Primary and Secondary nodes. Please, be careful with the format for "primaryLocation": and "secondaryLocation":. The format must be: <Country Code>,<City Name without spaces>,<Netskope Node ID>

Example:

```
"greCredentials": {
    "autoDiscovery": true,
    "grePublicIP": "",
    "primaryGreGateway": "163.116.146.36",
    "primaryProbelpAddress": "10.146.6.209",
    "primaryLocation": "US,Washington,IAD2",
    "secondaryGreGateway": "163.116.135.36",
    "secondaryProbelpAddress": "10.135.6.209",
    "secondaryLocation": "US,NewYork,NYC1"
},
```

4.3.1.1.6 "tunnelRedundancy":

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

Please select 'true' if you want the CSC to return to the Primary tunnel (after 10 min of stability) when using the Secondary tunnel.

Select 'false' if you want to remain using the Secondary Tunnel and not return to Primary. (Secondary will be nominated as 'new' Primary).

4.3.1.1.7 "bypassProxyPacUrl":

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

Insert here your Proxy Bypass PAC URL. The PAC URL contains the list of Proxy Bypasses to implement. For example:<https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac>

4.3.1.1.8 "routedBypassJsonFileUrl":

```
"routedBypassJsonFileUrl": ""
```

Insert here your Routed Bypass URL that points to the JSON file with the Routed Bypass Rules (I.e. routedBypassRulesFile.json) . For example:<https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json>

4.3.2 Filling "User Data" on the CloudFormation Template.

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

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

Specify stack details

Stack name

Stack name
csc-gre-aws-for-netskope
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Network Configuration
Which VPC should this be deployed to?
Select a VPC.
vpc-0f32a676 (172.31.0.0/16) (Net 172-31)

External Subnet
Select an External Subnet (WARNING !! must be on the same availability zone as Internal Subnet)
subnet-818c0ddb (172.31.96.0/24) (Net-172-31-96)

Internal Subnet
Select an Internal Subnet (WARNING !! must be on the same availability zone as External Subnet)
subnet-030dde6bf759ec79e (172.16.200.0/24) (net-172.16.200.0-24)

Amazon EC2 Configuration

Name
The name of the instance
csc-gre-aws-for-netskope

AWS Instance Type
Select one of the instance types
t3a.large

Key Name
Key Pair name
us-east-key

UserData
(Optional) Advanced Deployment: Paste here configUserData.json file content values.

Paste configUserData.json here below

```
{ "model": "csc-gre-ns-aws", "type": "configUserData", "version": "1.0", "dns": { "useCloudDNS": true, "primaryDnsIP": "", "second
```

Cancel Previous Next

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

4.3.3 Log information when using Advanced Deployment

When using Advanced Deployment, all resources are created automatically on the CSC.

```
Jan 2 18:44:57 ip-172-31-201-168 root: [MHB-CSC]INFO SYSCFG configured using Primary=172.31.202.163 Secondary= none and MCP ports 514
Jan 2 18:44:59 ip-172-31-201-168 root: [MHB-CSC]INFO AWS SSM Agent registered successfully to AWS Lambda. Registration values: {"ManagedInstanceID": "mi-0ff86ef8572caaf4d", "Region": "us-east-1"}
Jan 2 18:44:59 ip-172-31-201-168 root: [MHB-CSC]INFO Proxy Bypass List updated successfully. (using PAC URL https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac)
Jan 2 18:45:03 ip-172-31-201-168 root: [MHB-CSC]INFO GRE tunnel configured with Primary Node: 163.116.146.36 (US,NewYork,NYC1). Secondary Node: 163.116.135.36 (US,NewYork,NYC1).
Jan 2 18:45:03 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass Rules JSON file created successfully from config.json (using Routed Bypass URL https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json)
Jan 2 18:45:54 ip-172-31-201-168 root: [MHB-CSC]INFO CSC GRE for AWS was powered ON: Sun 2 Jan 18:45:53 UTC 2022
Jan 2 18:45:54 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass - Routed Bypass Rules JSON file integrity is OK
Jan 2 18:45:54 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass - (Index: 0) Rule "0365 Login URLs 1" was created successfully.
Jan 2 18:45:54 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass - (Index: 1) Rule "0365 Login URLs 2" was created successfully.
Jan 2 18:45:55 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass - (Index: 2) Rule "0365 Login URLs 3" was created successfully.
Jan 2 18:45:55 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass - (Index: 3) Rule "0365 Login URLs 4" was created successfully.
Jan 2 18:45:55 ip-172-31-201-168 root: [MHB-CSC]INFO Routed Bypass - (Index: 4) Rule "portquiz.net" was created successfully.
Jan 2 18:46:26 ip-172-31-201-168 root: [MHB-CSC]DOWN No active tunnel since: Sun 2 Jan 18:46:26 UTC 2022
Jan 2 18:51:27 ip-172-31-201-168 root: [MHB-CSC]UP Primary tunnel is active since: Sun 2 Jan 18:51:26 UTC 2022
```

The only manual task required is to add the Tunnel Public IP on your Netskope Console. You can read the Tunnel Public IP from the SSH console or AWS console.

<input checked="" type="checkbox"/>	csc-gre-for-netskope-on-aws-b	i-08cd4663125d25b0e	Running	2/2 checks passed	No alarms	+	us-east-1a
Instance: i-08cd4663125d25b0e (csc-gre-for-netskope-on-aws-b)							
Filter network interfaces							
Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address		
eni-0a75cf038298d8e31	csc-gre-single-internal-interface	-	-	-	172.31.202.71		
eni-0a359402727e8051b	csc-gre-single-external-interface	External Interface	-	3.225.70.2	172.31.201.168	Public IP	

5 Resources creates by the CloudFormation template

The following resources are created by the CloudFormation template:

1. Instance.

Instance: i-0283558d4cfb35311 (csc-gre-for-netskope-on-aws-a)

Details		Security	Networking	Storage	Status checks	Monitoring	Tags
<p>▼ Instance summary Info</p> <p>Instance ID i-0283558d4cfb35311 (csc-gre-for-netskope-on-aws-a)</p> <p>IPv6 address -</p> <p>Hostname type IP name: ip-172-31-96-172.ec2.internal</p> <p>Instance type t3a.medium</p> <p>Public IPv4 address 54.197.86.154 [open address]</p> <p>Instance state Running</p> <p>Private IP DNS name (IPv4 only) ip-172-31-96-172.ec2.internal</p> <p>Elastic IP addresses 54.197.86.154 [Public IP] 54.80.198.195 [Public IP]</p> <p>Private IPv4 addresses 172.31.96.172 172.31.200.220</p> <p>Public IPv4 DNS ec2-54-197-86-154.compute-1</p> <p>Answer private resource DNS name -</p> <p>VPC ID vpc-0f52a676 (Net 172-31)</p>							

2. Interfaces External and Internal.

Instance: i-0283558d4cfb35311 (csc-gre-for-netskope-on-aws-a)

Details		Security	Networking	Storage	Status checks	Monitoring	Tags																								
<p>► Networking details Info</p> <p>▼ Network Interfaces Info</p> <p>Network interfaces (2)</p> <table border="1"><thead><tr><th colspan="6">Filter network interfaces</th></tr><tr><th>Interface ID</th><th>Description</th><th>IPv4 Prefixes</th><th>IPv6 Prefixes</th><th>Public IPv4 address</th><th>Private IPv4 address</th></tr></thead><tbody><tr><td>eni-040f25fe5fc7482bd</td><td>csc-gre-single-external-interface</td><td>-</td><td>-</td><td>54.197.86.154</td><td>172.31.96.172</td></tr><tr><td>eni-0c122595d457ffce6</td><td>csc-gre-single-internal-interface</td><td>-</td><td>-</td><td>-</td><td>172.31.200.220</td></tr></tbody></table>								Filter network interfaces						Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address	eni-040f25fe5fc7482bd	csc-gre-single-external-interface	-	-	54.197.86.154	172.31.96.172	eni-0c122595d457ffce6	csc-gre-single-internal-interface	-	-	-	172.31.200.220
Filter network interfaces																															
Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address																										
eni-040f25fe5fc7482bd	csc-gre-single-external-interface	-	-	54.197.86.154	172.31.96.172																										
eni-0c122595d457ffce6	csc-gre-single-internal-interface	-	-	-	172.31.200.220																										

3. 1 x Elastic IP for the GRE tunnel and 1 x Elastic IP used by Bypass functionality and Private Access.

Instance: i-0283558d4cfb35311 (csc-gre-for-netskope-on-aws-a)

Details		Security	Networking	Storage	Status checks	Monitoring	Tags																				
<p>► Networking details Info</p> <p>► Network Interfaces Info</p> <p>▼ Elastic IP addresses Info</p> <p>Elastic IP addresses (2)</p> <table border="1"><thead><tr><th colspan="5">Filter Elastic IP addresses</th></tr><tr><th>Name</th><th>Allocated IPv4 address</th><th>Type</th><th>Address pool</th><th>Allocation ID</th></tr></thead><tbody><tr><td>-</td><td>54.197.86.154</td><td>Public IP</td><td>amazon</td><td>eipalloc-0df081e07f203226b</td></tr><tr><td>-</td><td>54.80.198.195</td><td>Public IP</td><td>amazon</td><td>eipalloc-0300fc5daadb967e6</td></tr></tbody></table>								Filter Elastic IP addresses					Name	Allocated IPv4 address	Type	Address pool	Allocation ID	-	54.197.86.154	Public IP	amazon	eipalloc-0df081e07f203226b	-	54.80.198.195	Public IP	amazon	eipalloc-0300fc5daadb967e6
Filter Elastic IP addresses																											
Name	Allocated IPv4 address	Type	Address pool	Allocation ID																							
-	54.197.86.154	Public IP	amazon	eipalloc-0df081e07f203226b																							
-	54.80.198.195	Public IP	amazon	eipalloc-0300fc5daadb967e6																							

4. Security Group for External Interface.^{1 2}

4.1. Inbound Rules

Inbound rules

Inbound rules

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
No security group rules found							

4.2. Outbound Rules

Outbound rules (7)

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
sgr-00487b96af717bd...	IPv4	DNS (UDP)	UDP	53	0.0.0.0/0	-	-
sgr-0257708d21547...	IPv4	HTTPS	TCP	443	0.0.0.0/0	-	-
sgr-07c602128c23fa3c	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	-	-
sgr-0678ddde240191...	IPv4	Custom UDP	UDP	123	0.0.0.0/0	-	-
sgr-0a21249632a208a...	IPv4	Custom TCP	TCP	1024 - 65535	0.0.0.0/0	-	-
sgr-095e3797f84a717c8	IPv4	GRE (47)	GRE (47)	All	0.0.0.0/0	-	-
sgr-075a0cc833cde32a7	IPv4	HTTP	TCP	80	0.0.0.0/0	-	-

5. Security Group for Internal Interface.

5.1. Inbound Rules

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
sgr-093a7c3f9aa5cdcf	IPv4	All traffic	All	All	192.168.0.0/16	-	-
sgr-0fdcfc49e3017ec19c	IPv4	All traffic	All	All	172.16.0.0/12	-	-
sgr-0172314d75c2c52...	IPv4	All traffic	All	All	10.0.0.0/8	-	-

5.2. Outbound Rules

Outbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
sgr-04cf1ddbb47fe8415	IPv4	All traffic	All	All	0.0.0.0/0	-	-

-
- 1 The CSC contains Firewall Rules on each interface that are more specific in some cases. For example, the CSC only allows reaching the configured Netskope Nodes for GRE traffic. Therefore, there is double protection: The AWS Security Group and the internal Firewall Rules of the CSC.
 - 2 When using Private Access (PriCPA), the CSC automatically updates the internal FW rules and Security Groups to allow Peers to communicate with each other.

6 The Cloud Security Connector Admin Console:

The CSC's SSH Console simplifies administrative 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 (using the CSC GW IP), you will receive the Admin Console.

For example:

The screenshot shows the AWS Cloud Security Connector Admin Console interface. At the top, it displays the instance ID: i-0283558d4cfb35311, status: Running, and instance type: t3a.medium. Below this, the 'NETWORK INTERFACES' section lists two interfaces:

Interface ID	Description	IPv4 Prefixes	IPv6 Prefixes	Public IPv4 address	Private IPv4 address
eni-040f25fe5fc7482bd	csc-gre-single-external-interface	-	-	54.197.86.154	172.31.96.172
eni-0c122595d457ffce6	csc-gre-single-internal-interface	-	-	CSC GW IP	172.31.200.220

```
ssh -i us-east-key.pem cscadmin@172.31.200.220
```

The initial screen will show a reminder to configure the Public IP of the GRE tunnel on your Netskope Console:

The screenshot shows a modal dialog box with the title "Maidenhead Bridge". It contains the following text:

Cloud Security Connector GRE on AWS for Netskope - Admin Console
Please, on your Netskope console, go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> GRE and check 'GRE configurations' to validate that 54.197.86.154 is added.
Did you configure the GRE IP 54.197.86.154 on the Netskope console? Please, confirm.
1) Yes
2) No
Enter your choice: []

→ Please, configure the Public IP of the GRE tunnel on your Netskope console.

The screenshot shows the "GRE" section of the AWS Cloud Platform. It includes a search bar, filter button, and buttons for "NEW GRE CONFIGURATION" and "NETSKOPE POPS". The main area displays a table of GRE configurations:

GRE Configurations		8 CREATED
NAME	SOURCE IDENTITY	
csc-gre-for-netskope-on-aws-a	54.197.86.154	[checkbox]

→ Select 1) Yes to confirm.

```
Maidenhead Bridge

Cloud Security Connector GRE on AWS for Netskope - Admin Console

EC2 Instance ID : i-0283558d4cfb35311
AWS Availability Zone : us-east-1d
Soft Version : 1.0

Please select an option by typing its number

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

CSC Admin tasks
5) AWS SSM Agent (Register or De-Register)
6) Manage Administrators
7) Change Timezone

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

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

Log Information
12) View Current Month
13) View Last 6 Months

Configuration Wizards
14) Change Nodes, DNS servers, Syslog and more.
15) Switch Tunnels - Primary / Secondary.
16) Update Netskope Nodes Databases.
17) High Availability configuration.

MHB Labs - Private Access - (Preview)
18) Show Configurations and Status Private Access.
19) Configure Private Access.
20) Configure CSC Remote Management via Private Access.

e) Exit

Selection: 
```

The Main Sections are:

- **Monitoring Tasks:** To check statuses, real-time traffic, speed, etc.
- **CSC Admin Tasks:** To register the CSC for AWS management, manage administrator and change 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.
- **MHB Labs – Private Access:** To configure and monitor Private Access.

6.1 Monitoring Tasks

6.1.1 Show Configuration and Status

```
GENERAL INFORMATION
Availability Zone: us-east-1d
EC2 Instance Id: i-0283558d4cfb35311 | Instance Type: t3a.medium | ami-id: ami-08c2df7e5356a8de6
External Interface (eth0) Subnet-id: subnet-818c0ddb | Interface-id: eni-040f25fe5fc7482bd | Security-Group-id: sg-0b81ea809def53ee8
Internal Interface (eth1) Subnet-id: subnet-8360ec09 | Interface-id: eni-0c122595d457ffce6 | Security-Group-id: sg-007aab93a98a659b
CSC date: Sun 2 Jan 20:09:45 UTC 2022
Soft version : 1.0

INTERFACES INFORMATION
External: Tunnel IP (eth0): 172.31.96.172/24 | Bypass Proxy Egress IP: 172.31.96.185 | Network Gateway: 172.31.96.1 is Alive
Internal: CSC GW IP (eth1): 172.31.200.220/24 | Network Gateway: 172.31.200.1 is Alive

TRAFFIC REDIRECTION Options
To Netskope: VIP Proxy: 172.31.200.68:80 | Route all traffic via CSC GW IP | Netskope Global Proxy IP: 163.116.128.80:80 via CSC GW IP
Direct to Internet: Bypass Proxy: 172.31.200.128:3128 | Netskope Global Proxy IP: 163.116.128.80:3128 via CSC GW IP

ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 54.197.86.154
Bypass Proxy Public IP: 54.80.198.195

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

NETSKOPE INFORMATION
GRE tunnels egress Public IP: 54.197.86.154

Primary Tunnel:
    Node : US,Washington,IAD2
    Node Public IP: 163.116.146.36
    Node Probe: 10.146.6.209
Secondary Tunnel:
    Node : US,NewYork,NYCI1
    Node Public IP: 163.116.135.36
    Node Probe: 10.135.6.209

TUNNEL STATUS
Primary Tunnel (reachability):
    Node Keepalive is: Alive
    GRE Tunnel IP is: Alive
Secondary Tunnel (reachability):
    Node Keepalive is: Alive
    GRE Tunnel IP is: Alive
returnToPrimaryTunnel: false

Tunnel Status: Primary tunnel is active since: Sun 2 Jan 16:26:12 UTC 2022

HTTP://WWW.NOTSKOPE.COM PAGE STATUS
163.116.146.117 Ashburn, United States (IAD2)

PROXY BYPASS - EGRESS INTERFACE STATUS
Proxy Bypass Egress Interface 172.31.96.185 can reach test page (https://ip.maidenheadbridge.com) via Public IP 54.80.198.195

ROUTED BYPASS
Using Routed Bypass URL: https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json
Routed Bypass URL https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json is reachable
Routed Bypass Rules configured via URL: 5

AWS SSM AGENT
AWS SSM Agent is active (running) since Sun 2022-01-02 16:25:38 UTC; 3h 44min ago
Registration values: {"ManagedInstanceId": "mi-0fc0d5394a2bfefed", "Region": "us-east-1"}

SYSLOG INFORMATION
SYSLOG Server (1) IP: 172.31.200.163 is Alive
SYSLOG Server (2) IP is not configured
SYSLOG TCP Port: 514

HIGH AVAILABILITY Information
The HA service is: active (running) since Sun 2022-01-02 20:04:14 UTC; 5min 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-0c122595d457ffce6 (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-08cd4663125d25b0e
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification
Private Access Public IP= 54.80.198.195

Press Enter to continue...
```

6.1.1.1 GENERAL INFORMATION

This section contains general information about the instance:

```
GENERAL INFORMATION
Availability Zone: us-east-1d
EC2 Instance id: i-0283558d4cfb35311 | Instance Type: t3a.medium | ami-id: ami-08c2df7e5356a8de6
External Interface (eth0) Subnet-id: subnet-818c0ddb | Interface-id: eni-040f25fe5fc7482bd | Security-Group-id: sg-0b81ea809def53ee8
Internal Interface (eth1) Subnet-id: subnet-8360ecd9 | Interface-id: eni-0c122595d457ffce6 | Security-Group-id: sg-007aabb93a98a659b
CSC date: Sun 2 Jan 11:13:36 UTC 2022
Soft version : 1.0
```

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

6.1.1.2 INTERFACES INFORMATION

This section contains the interfaces information:

```
INTERFACES INFORMATION
External: Tunnel IP (eth0): 172.31.96.172/24 | Bypass Proxy Egress IP: 172.31.96.185 | Network Gateway: 172.31.96.1 is Alive
Internal: CSC GW IP (eth1): 172.31.200.220/24 | Network Gateway: 172.31.200.1 is Alive
```

6.1.1.3 TRAFFIC REDIRECTION Options

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

```
TRAFFIC REDIRECTION Options
To Netskope: VIP Proxy: 172.31.200.68:80 | Route all traffic via CSC GW IP | Netskope Global Proxy IP: 163.116.128.80:80 via CSC GW IP
Direct to Internet: Bypass Proxy: 172.31.200.128:3128 | Netskope Global Proxy IP: 163.116.128.80:3128 via CSC GW IP
```

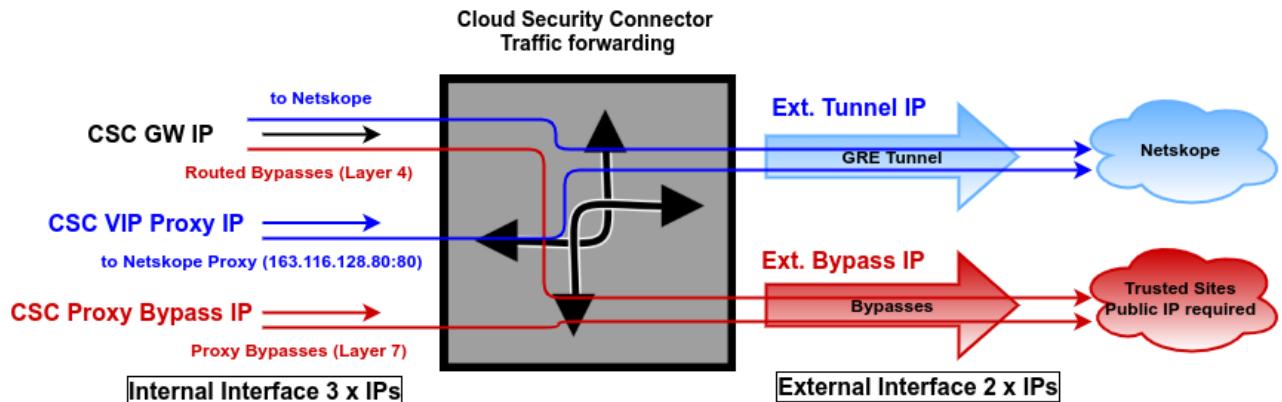
The objective of the Cloud Security Connectors of Maidenhead Bridge is to provide a simple architecture, 100% proven that works when connecting to Netskope.

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 Netskope when using Cloud Firewall. Traffic routed via CSC GW IP can be bypassed from Netskope using "Routed Bypasses" (Layer 4).
- **VIP Proxy:** This Virtual IP Proxy translates the packets directly to the Netskope 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-xxxyyzz)

Here an illustration about this:



Important: Please, see Chapter 7 for detailed information about traffic redirection (with examples)

6.1.1.4 ELASTIC (PUBLIC) IPs INFORMATION

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

ELASTIC (PUBLIC) IPs INFORMATION
GRE tunnels Public IP: 54.197.86.154
Bypass Proxy Public IP: 54.80.198.195

6.1.1.5 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

6.1.1.6 NETSKOPE INFORMATION

This section show the GRE tunnel information.



```
NETSKOPE INFORMATION
GRE tunnels egress Public IP: 54.197.86.154

Primary Tunnel:
    Node : US,Washington,IAD2
    Node Public IP: 163.116.146.36
    Node Probe: 10.146.6.209

Secondary Tunnel:
    Node : US,NewYork,NYC1
    Node Public IP: 163.116.135.36
    Node Probe: 10.135.6.209
```

6.1.1.7 TUNNEL STATUS

This section shows the KeepAlives & Probes to Netskope's Primary and Secondary Nodes and the Tunnel Status.

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

Tunnel Status: Primary tunnel is active since: Sun 2 Jan 09:58:23 UTC 2022
```

6.1.1.8 HTTP://WWW.NOTSKOPE.COM PAGE STATUS

This section shows the result of a HTTP GET from inside the CSC to URL: <http://www.notskope.com>

```
HTTP://WWW.NOTSKOPE.COM PAGE STATUS
163.116.146.119 Ashburn, United States (IAD2)
```

6.1.1.9 PROXY BYPASS - EGRESS INTERFACE STATUS

This sections validates if the Proxy Bypass can access internet directly going to <https://ip.maidenheadbridge.com>

```
PROXY BYPASS - EGRESS INTERFACE STATUS
Proxy Bypass Egress Interface 172.31.96.185 can reach test page (https://ip.maidenheadbridge.com) via Public IP 54.80.198.195
```

6.1.1.10 ROUTED BYPASS

This section shows the configuration of Routed Bypasses.

```
ROUTED BYPASS
Using Routed Bypass URL: https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json
Routed Bypass URL https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json is reachable
Routed Bypass Rules configured via URL: 5
```

6.1.1.11 AWS SSM AGENT

This section shows the status of the AWS SSM Agent.

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Sun 2022-01-02 12:45:11 UTC; 6min ago
Registration values: {"ManagedInstanceId": "mi-0fc0d5394a2bfefed", "Region": "us-east-1"}
```

6.1.1.12 SYSLOG/SIEM Servers Information

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

```
SYSLOG INFORMATION
SYSLOG Server (1) IP: 172.31.200.163 is Alive
SYSLOG Server (2) IP is not configured
SYSLOG TCP Port: 514
```

6.1.1.13 HIGH AVAILABILITY Information

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

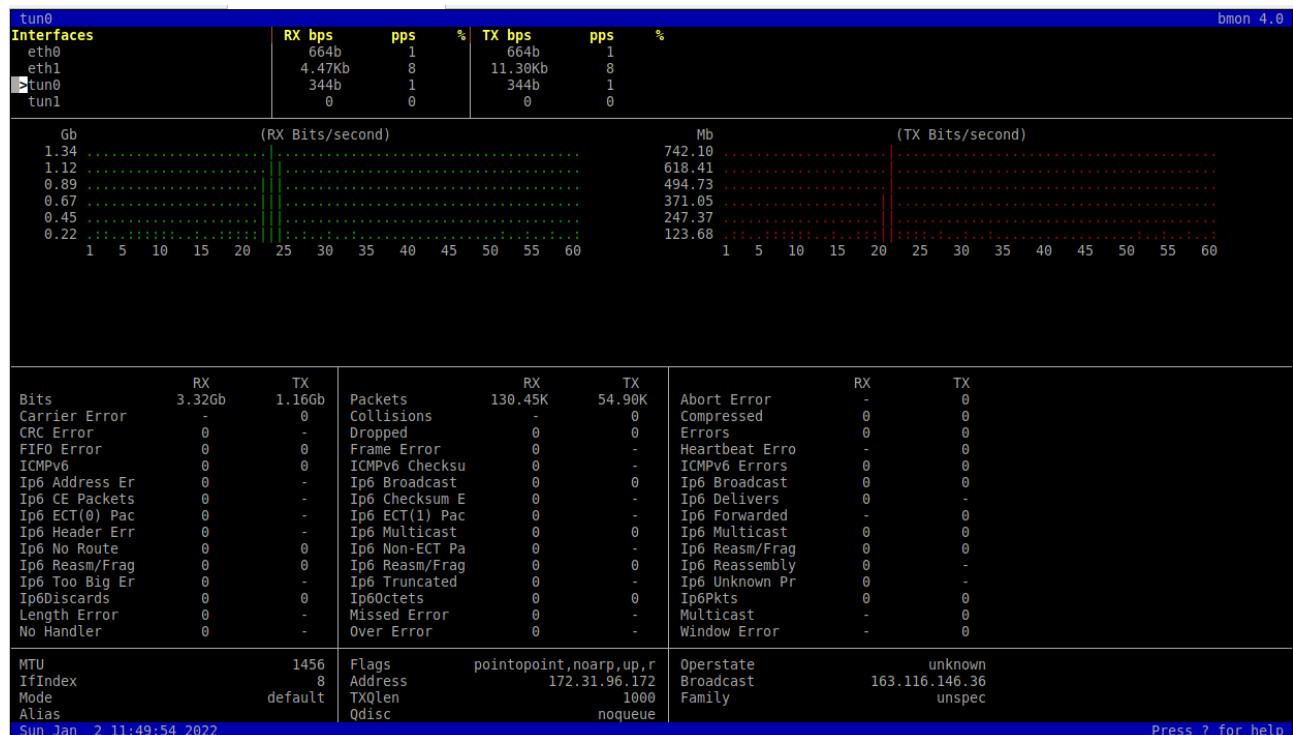
```
HIGH AVAILABILITY Information
The HA service is: active (running) since Sun 2022-01-02 20:04:14 UTC; 12h ago
IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
The Default Route to Internet is using this Gateway (Target): eni-0c122595d457ffce6 (this CSC)
Current values configured 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-08cd4663125d25b0e
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification
Private Access Public IP= 54.80.198.195
```

- 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.
- Private Access Public IP.

6.1.2 Show Interfaces Traffic

Use this section to see the traffic in real time.



6.1.3 Traceroute and Latency Test

This test can validate the quality of the Internet path between your location and Netskope. You can run it with tunnels down or up. When the tunnels are up, it does a “Reverse Path” test from your active Netskope 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 10 probes DIRECT to Primary / Secondary Netskope Nodes and a Reverse test via Active Tunnel to Public IP: 54.197.86.154.
 NOTE 1: Max Hops is equal 30. This test can take a while.
 NOTE 2: If you cannot see intermediate steps to Primary / Secondary, check if ICMP Time exceed (icmp type 11) is allowed to reach IP: 172.31.96.172 from the Internet.
 NOTE 3: For the Reverse test to work, you need to allow ICMP out to 'Any' on the Netskope Console.

Testing Primary Node: 163.116.146.36
 Start: 2022-01-02T11:54:11+0000
 HOST: ip-172-31-96-172

	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS???	???	10	0.0	0.0	0.0	0.0	0.0
2. AS???	???	10	0.0	0.0	0.0	0.0	0.0
3. AS???	???	10	0.0	0.0	0.0	0.0	0.0
4. AS???	241.0.4.85	10	0.4	0.4	0.3	0.5	0.1
5. AS???	243.253.19.70	10	0.3	0.4	0.3	0.4	0.0
6. AS???	240.0.36.17	10	0.4	0.4	0.3	0.4	0.0
7. AS???	240.0.36.6	10	0.4	0.4	0.3	0.6	0.1
8. AS???	242.0.163.33	10	0.6	0.6	0.4	0.8	0.2
9. AS???	52.93.28.155	10	16.4	3.0	1.0	16.4	4.7
10. AS???	100.100.2.72	10	1.2	1.5	1.0	3.7	0.8
11. AS174	be5855.ccr42.dca01.atlas.cogentco.com (38.140.146.185)	10	2.6	2.6	2.5	2.8	0.1
12. AS174	be3084.ccr41.iad02.atlas.cogentco.com (154.54.30.66)	10	2.8	2.9	2.7	3.4	0.2
13. AS174	te0-0-0.agr11.iad02.atlas.cogentco.com (154.54.44.198)	10	2.7	2.7	2.6	3.0	0.1
14. AS174	te0-0-1.nrl12.b045972-0.iad02.atlas.cogentco.com (154.24.32.134)	10	3.2	3.3	3.2	3.8	0.2
15. AS1299	netskope-svc070515-ic356561.ip.twelve99-cust.net (62.115.147.51)	10	2.7	2.7	2.6	2.7	0.0
16. AS55256	163.116.146.36	10	2.5	2.6	2.5	2.6	0.1

Testing Secondary Node: 163.116.135.36
 Start: 2022-01-02T11:54:27+0000
 HOST: ip-172-31-96-172

	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS???	???	10	0.0	0.0	0.0	0.0	0.0
2. AS???	???	10	0.0	0.0	0.0	0.0	0.0
3. AS???	???	10	0.0	0.0	0.0	0.0	0.0
4. AS???	241.0.4.219	10	0.4	0.3	0.3	0.4	0.0
5. AS???	243.253.22.206	10	0.3	0.4	0.3	0.6	0.1
6. AS???	240.0.40.16	10	0.3	0.4	0.3	1.2	0.3
7. AS???	240.0.40.7	10	0.3	0.3	0.3	0.4	0.0
8. AS???	242.0.171.129	10	0.4	0.5	0.3	0.7	0.1
9. AS???	52.93.28.173	10	1.3	1.6	0.9	4.5	1.1
10. AS???	100.100.8.6	10	1.1	1.1	1.0	1.2	0.1
11. AS???	100.91.192.9	10	6.5	6.5	6.5	6.7	0.1
12. AS???	52.93.1.1	10	11.5	14.2	7.1	18.6	4.1
13. AS???	52.93.51.32	10	6.5	7.4	6.5	13.0	2.0
14. AS???	ipv4.de-cix.nyc.us.as55256.netskope.com (206.82.104.169)	10	7.0	7.0	6.9	7.2	0.1
15. AS55256	163.116.135.36	10	6.9	7.0	6.9	7.1	0.1

Reverse path from: 163.116.146.36 to your Public IP: 54.197.86.154
 Start: 2022-01-02T11:54:42+0000
 HOST: ip-172-31-96-172

	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS???	ip-10-146-6-209.ec2.internal (10.146.6.209)	10	2.6	2.7	2.6	2.8	0.0
2. AS???	???	10	0.0	0.0	0.0	0.0	0.0
3. AS55256	163.116.146.3	10	3.3	3.2	3.1	3.4	0.1
4. AS1299	ash-bl-link.ip.twelve99-cust.net (62.115.147.50)	10	3.6	3.7	3.5	4.1	0.2
5. AS1299	vadata-svc077201-lag003749.ip.twelve99-cust.net (62.115.63.81)	10	3.8	4.1	3.3	6.4	1.1
6. AS???	???	10	0.0	0.0	0.0	0.0	0.0
7. AS???	???	10	0.0	0.0	0.0	0.0	0.0
8. AS???	52.93.29.202	10	4.8	4.5	4.2	4.8	0.2
9. AS???	???	10	0.0	0.0	0.0	0.0	0.0

6.1.4 SPEED TEST

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

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

Retrieving speedtest.net configuration...
Testing from Netskope (163.116.146.119)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by Netprotect (Ashburn, VA) [0.81 km]: 5.564 ms
Testing download speed...
Download: 1317.16 Mbit/s
Testing upload speed...
Upload: 1137.61 Mbit/s
```

6.2 CSC Admin Tasks

- CSC Admin tasks
- 5) AWS SSM Agent (Register or De-Register)
 - 6) Manage Administrators
 - 7) Change Timezone

6.2.1 AWS SSM Agent (Register or De-Register)

The CSC AWS has installed the AWS SSM Agent that allows you to check remotely the status of the CSC and “Run Commands” using AWS Systems Manager. You can manage all CSCs models³ using AWS Systems Manager.

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

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

6.2.1.1 Create a "Hybrid Activation" from AWS console.

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

1 AWS Systems Manager

2 Hybrid Activations

3 Create activation

4 - Put a name to the activation

5 - Repeat the name

6 Create activation

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

→ Click "Create Activation"

The screenshot shows the AWS Systems Manager Activations page. A success message is displayed: "You have successfully created a new activation. Your activation code is listed below. Copy this code and keep it in a safe place as you will not be able to access it again." Below the message are two fields: "Activation Code" (MLj+cpTwxKxht2jVaxza) and "Activation ID" (ca3a4d7c-36c7-4ca1-8835-e3b640f1ab5b). A link to "Learn more" is also present.

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

6.2.1.2 Register the CSC

The terminal window shows the following output:

```
Selection: 5
The SSM Agent is inactive (dead) since Sun 2022-01-02 09:37:13 UTC; 3h 7min 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 :MLj+cpTwxKxht2jVaxza
Activation ID :ca3a4d7c-36c7-4ca1-8835-e3b640f1ab5b
Region :us-east-1
AWS SSM AGENT
AWS SSM Agent is active (running) since Sun 2022-01-02 12:45:11 UTC; 39ms ago
Registration values: {"ManagedInstanceId": "mi-0fc0d5394a2bfefed", "Region": "us-east-1"}
```

Press Enter to continue...

6.2.1.3 View the Registered CSC on AWS Systems Manager

The screenshot shows the AWS Systems Manager Fleet Manager interface under the "Managed nodes" tab. It displays a table of registered nodes. One node is highlighted with a red border: "mi-0fc0d5394a2bfefed" (Node ID), "csc-gre-for-netskope-on-aws-a" (Node name), "172.31.96.172" (IP address), "Online" (SSM Agent ping status), "Linux" (Platform type), "Ubuntu" (Operating System), "20.04" (Platform version), "2.3.814.0" (SSM Agent version), and "ip-172-31-96-172" (Computer name).

6.2.2 Manage Administrators

The CSC for AWS has 3 users configured: cscadmin (for SSH Administrator Console Access), csccli (standard user) and ubuntu (standard user).

From this menu, you can edit the SSH Keys of each one or to disable the user. By default, cscadmin and ubuntu user have the same SSH key configured and are enabled. The user csccli is disabled by default.

```
Selection: 6

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

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

Note: the user "cscadmin" cannot be disabled.

6.2.2.1 Managing the SSH Key of a User

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

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

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

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

1) Manage SSH Keys
2) Quit
Enter your choice: 1

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

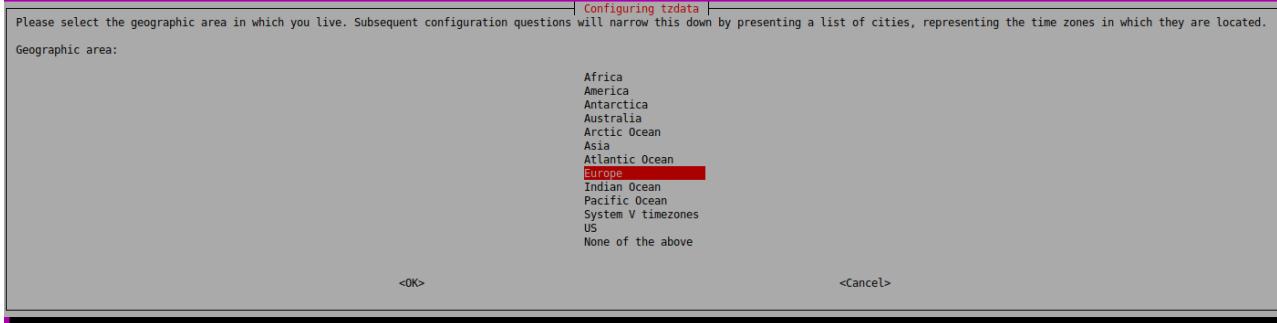
To save, press CTRL+S and to exit Nano, press CTRL+X

Do you want to continue?

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

6.2.3 Change Timezone

Use this menu to select the timezone of the CSC.

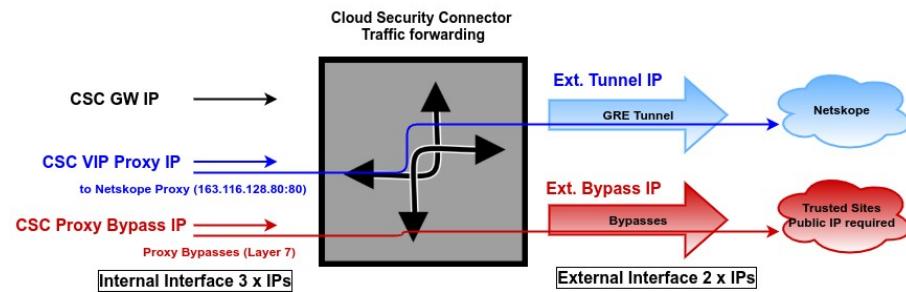


6.3 Proxy Bypass

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

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

6.3.1 Proxy Bypass - Traffic Flow



6.3.2 View Current Proxy Bypass List

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

```
Selection: 8

This is the list of current Domains configured:
login.microsoftonline.com
login.microsoft.com
login.windows.net
portquiz.net

Press Enter to continue...
```

6.3.3 Configure Proxy Bypass List

This menu allows to configure the Proxy Bypass List.

```
Please, select method to configure Proxy Bypass List
1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 1
```

6.3.3.1 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" and host the PAC file somewhere on the internet, for example, using an AWS S3 bucket. The CSC will read the "Proxy Bypass List" from the "Proxy Bypass PAC file" URL.

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

Example of Proxy Bypass PAC:

```
Please, select method to configure Proxy Bypass List
1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 1

Please, select method to configure Proxy Bypass List
1) Configure Proxy Bypass PAC URL and/or Update Proxy Bypasses
2) See PAC Proxy Bypass Example
3) Quit
Enter your choice: 2

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 Test Page
    (shExpMatch(host, "ip.maidenheadbridge.com"))) {
    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

Configuring the Proxy Bypass PAC URL and Refresh the List

```
Selection: 9
Please, select method to configure Proxy Bypass List
1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 1
Please, select method to configure Proxy Bypass List
1) Configure Proxy Bypass PAC URL and/or Update Proxy Bypasses
2) See PAC Proxy Bypass Example
3) Quit
Enter your choice: 1
Proxy Bypass Configuration
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.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
Bypass PAC URL:https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac
Do you want to refresh Proxy Bypass List?
1) Yes
2) No
Enter your choice: 1
This is your current Proxy Bypass List
login.microsoftonline.com
login.microsoft.com
login.windows.net
portquiz.net

Do you want apply changes?
1) Yes
2) No
Enter your choice: 1
Proxy Bypass List updated sucessfully.
```

6.3.3.2 Manual Proxy Bypass Configuration.

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

1. Select Option 2)

```
1) Auto - Proxy Bypass PAC URL
2) Manual
3) Quit
Enter your choice: 2
Please, read the instructions carefully:
You are going to edit the list using NANO editor
The following formats are accepted:
Full Domains: 'www.example.com'
Wildcard for subdomains: '.example.com' - This will allow all subdomains of example.com
To save, press CTRL-X and 'Yes'
Paid attention to ERROR messages if any. ERRORS must be corrected before to continue
Do you want to continue? (y/n)? 
```

2. Input "y"

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

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

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

```
This is your current Proxy Bypass List

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

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

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

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

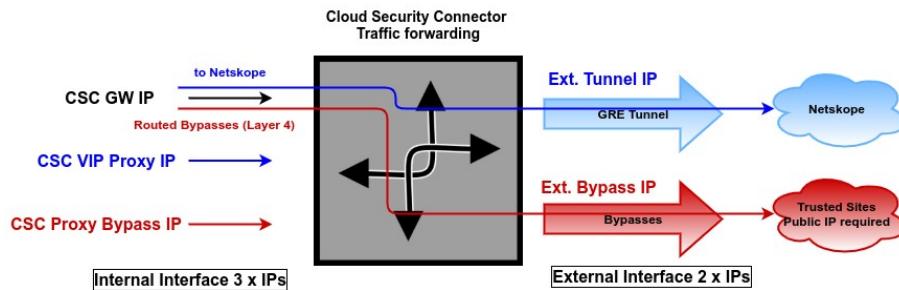
Proxy Bypass List updated sucessfully.
```

6.4 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 Netskope. 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

6.4.1 Routed Bypass - Traffic Flow



6.4.2 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: 
```

6.4.2.1 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: "0365 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: "0365 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: "0365 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: "0365 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"
```

6.4.2.2 Json

```
Enter your choice: 2

{
  "routedBypassRules": [
    {
      "description": "0365 Login URLs 1",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "80",
      "toPort": "80"
    },
    {
      "description": "0365 Login URLs 2",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "20.190.128.0/18",
      "fromPort": "443",
      "toPort": "443"
    },
    {
      "description": "0365 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": "0365 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
```

6.4.3 Configure Routed Bypass List

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

```
Selection: 11

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

6.4.3.1 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://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json>

```
Enter your choice: 1
Your Routed Bypass URL configured is: https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json
Do you want to change the Routed Bypass URL?
1) Yes
2) No
Enter your choice: 1
Please, input Routed Bypass URL
Routed Bypass URL: https://csc-gre-aws.s3.amazonaws.com/routedBypassRulesFile.json
Do you want to refresh the Routed Bypass List?
1) Yes
2) No
Enter your choice: 1
Routed Bypass JSON file imported successfully

You can review your values before to apply. Please, Select 'Compact' or 'Json' format.
1) Compact
2) Json
3) No review is needed
Enter your choice: 1
Current Values configured are:
Index: 0, Protocol: 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: 52.120.289.10/32, FromPort: 80, To Port: 80, Description: "portquiz2.net"
Index: 4, Protocol: tcp, SourceIP: 0.0.0.0/0, DestinationIP: 10.126.0.0/18, FromPort: 443, To Port: 443, Description: "portquiz2.net"
Index: 5, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 13.107.24.0/18, FromPort: 3478, To Port: 3481, Description: "Skype and Teams UDP 4"
Index: 6, Protocol: udp, SourceIP: 0.0.0.0/0, DestinationIP: 52.12.0.0/14, 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 successfully.
(Index: 1) Rule "O365 Login URLs 2" was created successfully.
(Index: 2) Rule "O365 Login URLs 3" was created successfully.
(Index: 3) Rule "portquiz2.net" was created successfully.
(Index: 4) Rule "O365 Login URLs 4" was created successfully.
(Index: 5) Rule "Skype and Teams UDP 1" was created successfully.
(Index: 6) Rule "Skype and Teams UDP 2" was created successfully.
(Index: 7) Rule "Skype and Teams UDP 3" was created successfully.

Routed Bypass List updated succesfully.

Press ENTER to continue
```

6.4.3.2 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: "0365 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: "0365 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: "0365 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: "0365 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 "0365 Login URLs 1" was created succesfully.
(Index: 1) Rule "0365 Login URLs 2" was created succesfully.
(Index: 2) Rule "0365 Login URLs 3" was created succesfully.
(Index: 3) Rule "portquiz.net" was created succesfully.
(Index: 4) Rule "0365 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
```

6.5 Log Information

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

```
Log Information
12) View Current Month
13) View Last 6 Months
```

6.5.1 View Current Month

```
Selection: 12

Current Month (October 2021) Logs for ns-cgc00002-a

Oct  4 18:31:06 root: (MHB-CSC)(STANDBY) ns-cgc00002-a is Cluster StandBy - No active tunnels
Oct  4 18:31:11 root: (MHB-CSC)(UP) CSC GRE Cluster was powered ON: Mon  4 Oct 18:31:11 UTC 2021
Oct  4 18:31:15 root: (MHB-CSC)(INFO) Routed Bypass Rules JSON file integrity is OK
Oct  4 18:31:16 root: (MHB-CSC)(INFO) (Index: 0) Rule "0365 Login URLs 1" was created successfully.
Oct  4 18:31:17 root: (MHB-CSC)(INFO) (Index: 1) Rule "0365 Login URLs 2" was created successfully.
Oct  4 18:31:18 root: (MHB-CSC)(INFO) (Index: 2) Rule "0365 Login URLs 3" was created successfully.
Oct  4 18:31:18 root: (MHB-CSC)(INFO) (Index: 3) Rule "portquiz.net" was created successfully.
Oct  4 18:31:19 root: (MHB-CSC)(INFO) (Index: 4) Rule "0365 Login URLs 4" was created successfully.
Oct  4 18:31:20 root: (MHB-CSC)(INFO) (Index: 5) Rule "Skype and Teams UDP 1" was created successfully.
Oct  4 18:31:21 root: (MHB-CSC)(INFO) (Index: 6) Rule "Skype and Teams UDP 2" was created successfully.
Oct  4 18:31:21 root: (MHB-CSC)(INFO) (Index: 7) Rule "Skype and Teams UDP 3" was created successfully.
Oct  4 18:32:16 root: (MHB-CSC)(UP) Primary tunnel is active since: Mon  4 Oct 18:32:16 UTC 2021
Oct  4 18:32:16 root: (MHB-CSC)(ACTIVE) ns-cgc00002-a is Cluster Active
Oct  4 18:38:01 cscadmin: (MHB-CSC)(INFO) User 'csccli' was enabled via console.
Oct  4 18:38:07 cscadmin: (MHB-CSC)(INFO) SSH Key file was modified for user 'csccli'.
```

6.5.2 View Last 6 Months

```
Selection: 13

Last 6 Months Logs up to Current Month (October 2021) for ns-cgc00002-a

Oct  4 18:31:06 root: (MHB-CSC)(STANDBY) ns-cgc00002-a is Cluster StandBy - No active tunnels
Oct  4 18:31:11 root: (MHB-CSC)(UP) CSC GRE Cluster was powered ON: Mon  4 Oct 18:31:11 UTC 2021
Oct  4 18:31:15 root: (MHB-CSC)(INFO) Routed Bypass Rules JSON file integrity is OK
Oct  4 18:31:16 root: (MHB-CSC)(INFO) (Index: 0) Rule "0365 Login URLs 1" was created successfully.
Oct  4 18:31:17 root: (MHB-CSC)(INFO) (Index: 1) Rule "0365 Login URLs 2" was created successfully.
Oct  4 18:31:18 root: (MHB-CSC)(INFO) (Index: 2) Rule "0365 Login URLs 3" was created successfully.
Oct  4 18:31:18 root: (MHB-CSC)(INFO) (Index: 3) Rule "portquiz.net" was created successfully.
Oct  4 18:31:19 root: (MHB-CSC)(INFO) (Index: 4) Rule "0365 Login URLs 4" was created successfully.
Oct  4 18:31:20 root: (MHB-CSC)(INFO) (Index: 5) Rule "Skype and Teams UDP 1" was created successfully.
Oct  4 18:31:21 root: (MHB-CSC)(INFO) (Index: 6) Rule "Skype and Teams UDP 2" was created successfully.
Oct  4 18:31:21 root: (MHB-CSC)(INFO) (Index: 7) Rule "Skype and Teams UDP 3" was created successfully.
Oct  4 18:32:16 root: (MHB-CSC)(UP) Primary tunnel is active since: Mon  4 Oct 18:32:16 UTC 2021
Oct  4 18:32:16 root: (MHB-CSC)(ACTIVE) ns-cgc00002-a is Cluster Active
```

6.6 Configuration Wizards

In this section, you can run the Configuration Wizard to change GRE Nodes, DNS servers, Etc; Switch tunnels, Update Netskope Nodes Databases and configure High Availability.

Configuration Wizards
14) Change Nodes, DNS servers, Syslog and more.
15) Switch Tunnels - Primary / Secondary.
16) Update Netskope Nodes Databases.
17) High Availability configuration.

6.6.1 Change Nodes, DNS servers, Syslog and more.

With this wizard you can change:

1. Netskope Nodes
2. DNS Servers
3. Bypass Proxy PAC URL
4. Routed Bypass JSON URL
5. Syslog Servers.

```
Selection: 14
Welcome to the CSC GRE Configuration Wizard
Please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> GRE and check 'GRE configurations' to validate that 82.68.6.74 is added.
Current Values Configured:
-----
NETSKOPE INFORMATION
GRE tunnels egress Public IP: 82.68.6.74

Primary Tunnel:
    Node : GB,London,LON1
    Node Public IP: 163.116.162.36
    Node Probe: 10.162.6.209
Secondary Tunnel:
    Node : GB,Manchester,MAN1
    Node Public IP: 163.116.165.36
    Node Probe: 10.165.6.209
returnToPrimaryTunnel: true
-----
DNS Servers: 172.19.0.100 ; 1.1.1.1
-----
Bypass Proxy PAC URL
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac
-----
Routed Bypass URL
Your current Routed Bypass URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json
-----
Syslog / SIEM information
Primary SysLog / SIEM IP: 172.19.0.199
Secondary Syslog / SIEM IP: Not configured
Syslog / SIEM TCP port: 514
-----
Are you ready to continue?
1) Yes
2) No
Enter your choice: 
```

6.6.1.1 Running the Configuration Wizard

Select Nodes, Auto o Manual.

If you select "Auto" the CSC will select the nearest Netskope Nodes to "GRE Tunnels egress Public IP". If you select Manual, you can choose manually the Nodes Primary and Secondary.

```
Are you ready to continue?  
1) Yes  
2) No  
Enter your choice: 1  
-----  
NETSKOPE INFORMATION  
GRE tunnels egress Public IP: 82.68.6.74  
  
Primary Tunnel:  
    Node : GB,London,LON1  
    Node Public IP: 163.116.162.36  
    Node Probe: 10.162.6.209  
Secondary Tunnel:  
    Node : GB,Manchester,MAN1  
    Node Public IP: 163.116.165.36  
    Node Probe: 10.165.6.209  
returnToPrimaryTunnel: true  
  
Do you want to change the Netskope Tunnel values?  
1) Yes  
2) No  
Enter your choice: 1  
-----  
Please, select Manual or Auto Node Selection  
1) Manual  
2) Auto  
3) Quit  
Enter your choice: 1
```

Selecting "Manual"

-> Select your Primary Node.

```
Please, select your Primary Node (Country/City/NodeID)  
1) AE,Dubai,DXB1 9) BR,SaoPaulo,SAO1 17) DE,Frankfurt,FRA1 25) IN,Chennai,MAA1 33) NZ,Auckland,AKL1 41) US,LosAngeles,LAX1  
2) AE,Dubai,DXB2 10) CA,Montreal,YM01 18) ES,Madrid,MAD1 26) IN,Delhi,DEL1 34) SE,Stockholm,STO1 42) US,Miami,MIA1  
3) AR,BuenosAires,BUE1 11) CA,Toronto,YYZ1 19) FR,Paris,PAR1 27) IN,Mumbai,MUM1 35) SG,Singapore,SIN1 43) US,Chicago,ORD1  
4) AT,Vienna,AT1 12) CA,Vancouver,YVR1 20) FR,Lyon,LYN1 28) IT,Rome,ROM1 36) US,Atlanta,ATL1 44) US,Phoenix,PHX1  
5) AU,Brisbane,BNE1 13) CH,Zurich,ZUR1 21) GB,London,LON1 29) JP,Osaka,OSA1 37) US,Atlanta,ATL1 45) US,SanFrancisco,SFO1  
6) AU,Melbourne,MEL1 14) CL,Santiago,SC1 22) GB,Manchester,MAN1 30) JP,Tokyo,NRT1 38) US,Chicago,ORD1 46) US,Seattle,SEA1  
7) AU,Perth,PER1 15) CO,Bogota,BOG1 23) HK,HongKong,HKG1 31) KR,Seoul,ICN1 39) US,Dallas,DFW1 47) US,Washington,IAD2  
8) AU,Sydney,SYD1 16) DE,Dusseldorf,DUS1 24) IL,TelAviv,TLV1 32) NL,Amsterdam,AMS1 40) US,Denver,DEN1 48) ZA,Johannesburg,JNB1  
Enter your choice: 43 49) Not in the list? Input Manually  
50) Quit
```

-> Select your Secondary Node

.....									
Please, select your Secondary Node (Country/City/NodeId)									
1) AE,Dubai,DXB1	9) BR,SaoPaulo,SAO1	17) DE,Frankfurt,FRA1	25) IN,Chennai,MMA1	33) NZ,Auckland,AKL1	41) US,LosAngeles,LAX1	49) Not in the list? Input Manually			
2) AE,Dubai,DXB2	10) CA,Monreal,YQ1	18) ES,Madrid,MD1	26) IN,Delhi,DEL1	34) SE,Stockholm,STO1	42) US,Miami,MTA1	50) Quit			
3) AR,BuenosAires,BUE1	11) CA,Toronto,YZ1	19) FR,Marseille,MRS1	27) IN,Mumbai,BOM1	35) SG,Singapore,SIN1	43) US,NewYork,NYC1				
4) AT,Vienna,VIE1	12) CA,Vancouver,YVR1	20) FR,Paris,PAR1	28) IT,Milan,MIL1	36) US,Ashburn,DC11	44) US,Phoenix,PHX1				
5) AU,Brisbane,BNE1	13) CH,Zurich,ZUR1	21) GB,London,LON1	29) JP,Osaka,OSA1	37) US,Atlanta,ATL1	45) US,SanFrancisco,SFO1				
6) AU,Perth,PER1	14) CO,Bogota,BOG1	22) JP,Tokyo,TOK1	30) JP,Osaka,OSA1	38) US,Chicago,CDG1	46) US,Seattle,SEA1				
7) AU,Perth,PER1	15) CO,Bogota,BOG1	23) HK,HongKong,HKG1	31) KR,Seoul,ICN1	39) US,Dallas,DFW1	47) US,Washington,IAD2				
8) AU,Sydney,SYD1	16) DE,Dusseldorf,DUS1	24) IL,TelAviv,TLV1	32) NL,Amsterdam,AMS1	40) US,Denver,DEN1	48) ZA,Johannesburg,JNB1				
Enter your choice: 47									

-> Select '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)

```
'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
```

DNS Configuration

```
-----  
DNS Configuration  
  
Your current DNS Servers are: 172.19.0.100 ; 1.1.1.1  
  
Do you want to change the DNS servers?  
1) Yes  
2) No  
Enter your choice: 2
```

Proxy Bypass Configuration

```
-----  
Proxy Bypass Configuration  
  
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac  
  
Do you want to change the Proxy Bypass PAC URL?  
1) Yes  
2) No  
Enter your choice: 2  
  
Do you want to refresh Proxy Bypass List?  
1) Yes  
2) No  
Enter your choice: 2
```

Routed Bypass Configuration

```
Routed Bypass Configuration
Your Routed Bypass URL configured is: https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json
Do you want to change the Routed Bypass URL?
1) Yes
2) No
Enter your choice: 2

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

Syslog / SIEM Configuration

```
Syslog / SIEM Configuration
Primary Syslog / SIEM IP: 172.19.0.199
Secondary Syslog / SIEM IP: Not configured
Syslog / SIEM TCP port: 514

Do you want to change Syslog / SIEM Servers values?
1) Yes
2) No
3) Reset default values
Enter your choice: 2
```

At the end of the Wizard, you will be asked to confirm this values.

```
Please confirm these values:
-----
NETSKOPE INFORMATION
GRE tunnels egress Public IP: 82.68.6.74

Primary Tunnel:
    Node : US,NewYork,NYC1
    Node Public IP: 163.116.135.36
    Node Probe: 10.135.6.209
Secondary Tunnel:
    Node : US,Washington,IAD2
    Node Public IP: 163.116.146.36
    Node Probe: 10.146.6.209
returnToPrimaryTunnel: true
-----
DNS Servers: 172.19.0.100 ; 1.1.1.1
-----
Bypass Proxy PAC URL
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-csc-bypass-pac-documentation.pac
-----
Routed Bypass URL
Your current Routed Bypass URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile-documentation.json
-----
Primary Syslog / SIEM server IP: 172.19.0.199
Syslog / SIEM TCP port IP: 514
-----
Do you want to implement these values? (The CSC will reboot)
1) Yes
2) No
Enter your choice: 1
```

Done!

6.6.2 Switch Tunnels - Primary / Secondary.

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

```
Selection: 15

NETSKOPE INFORMATION
GRE tunnels egress Public IP: 82.68.6.74

Primary Tunnel:
    Node : GB,London,LON1
    Node Public IP: 163.116.162.36
    Node Probe: 10.162.6.209
Secondary Tunnel:
    Node : GB,Manchester,MAN1
    Node Public IP: 163.116.165.36
    Node Probe: 10.165.6.209

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

Tunnel Status: Primary tunnel is active since: Fri 8 Oct 03:19:08 BST 2021

HTTP://WWW.NOTSKOPE.COM PAGE STATUS
163.116.162.116 London, United Kingdom (LON1)
-----
Do you want to Switch Primary / Secondary Tunnel?
Selecting Yes will disrupt all current connections.

1) Yes
2) No
Enter your choice: □
```

6.6.3 Update Netskope Nodes Databases.

This command retrieves the latest Netskope Node Database.

```
Selection: 16

Checking Netskope Nodes Databases...
This CSC has the latest version: 1.2
```

6.6.4 High Availability configuration

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

```
Selection: 17

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

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

{

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

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

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

-----
The HA service is NOT Active
Do you want to configure it?
1) Yes
2) No
Enter your choice: 
```

Help provided:

How to configure:

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

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

- 3) Get the 'Route Table ID' of the Route Table/s where there is Default Route (0.0.0.0/0) to Internet
- 4) Get the 'Instance ID' of the other CSC on the pair
- 5) Create a SNS notification and get the 'ARN'.

- 6) Run the Wizard on the FIRST CSC and input the following values manually: (all values are mandatory)
 - 6.1) Route Table ID/s (where there is Default Route to internet).
 - 6.2) Instance ID of other CSC on the pair.
 - 6.3) ARN of the SNS message for Notifications of Route changes.
- 7) Run the Wizard on the SECOND CSC pasting the JSON file obtained from the FIRST CSC

How it works:

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

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

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

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

6.6.4.1 High Availability configuration on detail

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

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

Instances (3) Info							
<input type="checkbox"/> Search		Connect					
<input type="checkbox"/> Instance state = running X		Clear filters					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	csc-gre-for-netskope-on-aws-a	i-0283558d4cfb35311	Running	t3a.medium	2/2 checks passed	No alarms	+ us-east-1d
<input type="checkbox"/>	csc-gre-for-netskope-on-aws-b	i-08cd4663125d25b0e	Running	t3a.medium	2/2 checks passed	No alarms	+ us-east-1a

6.6.4.1.2 Create an IAM role with the following policies

Identity and Access Management (IAM) 1

Dashboard

Access management

User groups

Users

Roles 2

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:█████████████████████:role/csc-ha-aws-role
Role description	Allows EC2 instances to call AWS services on your behalf. Edit
Instance Profile ARNs	arn:aws:iam:█████████████████████:instance-profile/csc-ha-aws-role
Path	/
Creation time	2019-10-26 09:18 UTC
Last activity	2022-01-02 09:21 UTC (Today)
Maximum session duration	1 hour Edit

Permissions Trust relationships Tags Access Advisor Revoke sessions

▼ Permissions policies (1 policy applied)

Attach policies 3

Policy name ▾

csc-ha-aws-iam

Policy summary { JSON Edit policy

```

1 - {
2 -   "Version": "2012-10-17",
3 -   "Statement": [
4 -     {
5 -       "Sid": "VisualEditor0",
6 -       "Effect": "Allow",
7 -       "Action": [
8 -         "ec2:DisassociateAddress",
9 -         "ec2:AuthorizeSecurityGroupEgress",
10 -        "ec2:AuthorizeSecurityGroupIngress",
11 -        "ec2:DescribeAddresses",
12 -        "ec2:DescribeInstances",
13 -        "sns:ListSubscriptionsByTopic",
14 -        "ec2:CreateTags",
15 -        "ec2:DescribeSecurityGroups",
16 -        "ec2:ReplaceRoute",
17 -        "ec2:ReplaceSecurityGroupIngress",
18 -        "sns:Publish",
19 -        "ec2:DescribeSecurityGroupRules",
20 -        "ec2:AssociateAddressWithSecurityGroup"
21 -      ]
22 -     }
23 -   ]
24 - }

```

4 - Copy/Paste the Policy

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

Rigth click the instance → Security → Modify IAM role

EC2 > Instances > i-0283558d4cfb35311 > Modify IAM role

Modify IAM role Info

Attach an IAM role to your instance.

Instance ID
 i-0283558d4cfb35311 (csc-gre-for-netskope-on-aws-a)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

Create new IAM role

Cancel Save

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

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

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

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

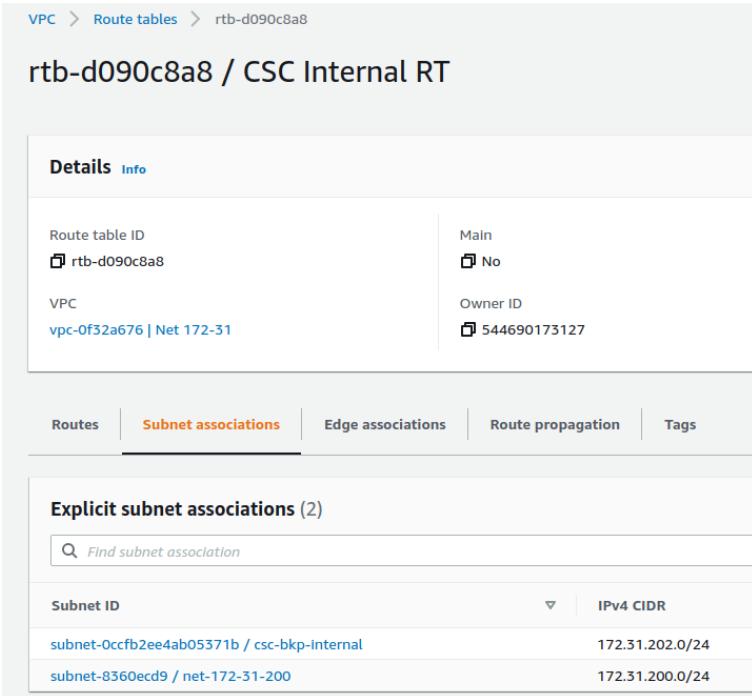
The screenshot shows the AWS VPC Route Tables page. A red box highlights the 'rtb-d090c8a8 / CSC Internal RT' route table. Another red box highlights the 'vpc-0f32a676' filter in the sidebar. The 'Routes' tab is selected in the main content area, showing four routes. One route, with a destination of '0.0.0.0/0', has its target set to 'eni-0c122595d457ffce6'. This target is also highlighted with a red box.

Destination	Target	Status
217.155.196.81/32	igw-04fa065a58fbe0e32	Active
82.68.6.72/29	igw-04fa065a58fbe0e32	Active
172.31.0.0/16	local	Active
0.0.0.0/0	eni-0c122595d457ffce6	Active

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

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

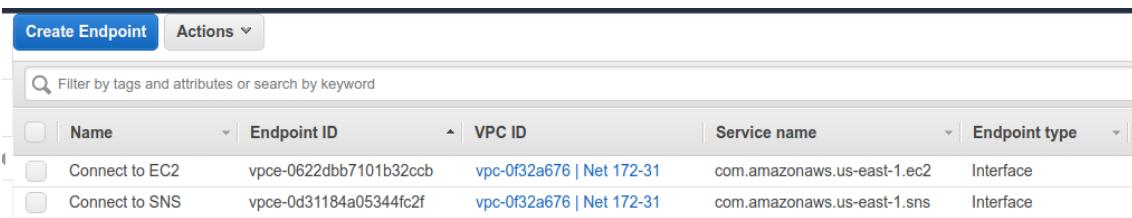
Next, apply the Subnet Associations to the Routing Table:



The screenshot shows the AWS VPC Route Tables page. The route table ID is rtb-d090c8a8, labeled as 'CSC Internal RT'. The main section displays basic details: Route table ID (rtb-d090c8a8), Main status (No), VPC (vpc-0f32a676 | Net 172-31), and Owner ID (544690173127). Below this, there are tabs for Routes, Subnet associations (which is selected), Edge associations, Route propagation, and Tags. Under the Subnet associations tab, it says 'Explicit subnet associations (2)' and lists two subnets: subnet-0ccfb2ee4ab05371b / csc-bkp-internal (IPv4 CIDR 172.31.202.0/24) and subnet-8360ecd9 / net-172-31-200 (IPv4 CIDR 172.31.200.0/24).

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

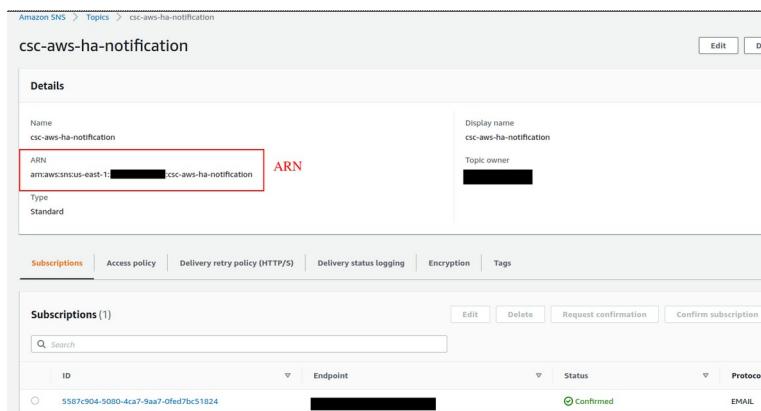
When changing the default route to the internet via Netskope, your subnets will potentially lose contact with some AWS services: EC2, SNS, S3, etc. The CSC on the HA requires creating two endpoints: EC2 and SNS.



The screenshot shows the AWS Endpoints page. The 'Create Endpoint' button is highlighted. A search bar allows filtering by tags and keywords. The table lists two endpoints: 'Connect to EC2' (Endpoint ID vpce-0622dbb7101b32ccb, VPC ID vpc-0f32a676 | Net 172-31, Service name com.amazonaws.us-east-1.ec2, Endpoint type Interface) and 'Connect to SNS' (Endpoint ID vpce-0d31184a05344fc2f, VPC ID vpc-0f32a676 | Net 172-31, Service name com.amazonaws.us-east-1.sns, Endpoint type Interface).

6.6.4.1.5 Create SNS message for Alerts.

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



The screenshot shows the 'Topics' section of the Amazon SNS console. A single topic named 'csc-aws-ha-notification' is listed. The ARN (arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification) is highlighted with a red box. Other details shown include the display name 'csc-aws-ha-notification' and the topic owner. Below the main topic information, there is a 'Subscriptions' tab showing one confirmed subscription to an email endpoint.

6.6.4.1.6 Run the HA Wizard on the First CSC

Input the values manually on the First CSC.

```
The HA service is NOT Active
Do you want to configure it?
1) Yes
2) No
Enter your choice: 1  1
Please, Select 'Manual' to configure the First CSC on the HA pair or 'Json' for the Second CSC.
1) Manual
2) Json
3) Quit
Enter your choice: 1  2
IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
-----
Please, input the following values:
Route Table ID= rtb-d090c8a8  3
Do you want to add another Route Table ID?
1) Yes
2) No
Enter your choice: 2
Instance ID of other CSC in the pair= i-08cd4663125d25b0e
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification  4
-----
Values to configure are:
Routing Tables=1
Routing Table ID= rtb-d090c8a8
Instance ID of other CSC in the pair= i-08cd4663125d25b0e
SNS message ARN= arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification
Do you want to apply changes?
1) Yes
2) No
Enter your choice: 1  5
```

```

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

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

High Availability JSON file:

{
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role",
    "haSnsMessageArn": "arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification",
    "haInstanceIdFirstCsc": "i-0f15f29fc4d69eb6",
    "haInstanceIdSecondCsc": "i-08cd4663125d25b0e",
    "haBypassPublicIpFirstCsc": "23.23.210.207",
    "haBypassPublicIpSecondCsc": "174.129.249.225",
    "haPrivateAccessPublicIp": "23.23.210.207",
    "haVPC": "vpc-0f32a676",
    "haRouteTables": [
      {
        "routeTableId": "rtb-d090c8a8"
      }
    ]
  }
}

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

```

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

6.6.4.1.7 Configure the second CSC on the HA pair.

Run the HA Wizard on the second CSC.

```

Do you want to configure it?
1) Yes
2) No
Enter your choice: 1

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

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

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

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

Private Access Local Config JSON file: {
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role",
    "haSnsMessageArn": "arn:aws:sns:us-east-1:544690173127:csc-aws-ha-notification",
    "haInstanceIdFirstCsc": "i-0f15f29fc4d69eb6",
    "haInstanceIdSecondCsc": "i-08cd4663125d25b0e",
    "haBypassPublicIpFirstCsc": "23.23.210.207",
    "haBypassPublicIpSecondCsc": "174.129.249.225",
    "haPrivateAccessPublicIp": "23.23.210.207",
    "haVPC": "vpc-0f32a676",
    "haRouteTables": [
      {
        "routeTableId": "rtb-d090c8a8"
      }
    ]
  }
}

(MHB-CSC) [INFO] High Availability: IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
(MHB-CSC) [INFO] High Availability JSON file (highAvailability.json) integrity is OK
(MHB-CSC) [INFO] High Availability: IAM role in use: arn:aws:iam::544690173127:instance-profile/csc-ha-aws-role
(MHB-CSC) [INFO] High Availability: Using VPC vpc-0f32a676. All Route Table IDs must belong to VPC vpc-0f32a676.
(MHB-CSC) [INFO] High Availability: Route Table ID rtb-d090c8a8 configured.
(MHB-CSC) [INFO] High Availability is active (running) since Mon 2022-01-10 20:46:12 UTC; 18ms ago.

Press Enter to continue...

```

3 - Paste JSON

Done!

6.6.4.1.8 Checking HA Status

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

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

Press Enter to continue...
```

6.6.4.1.9 Notifications from CSC on HA

Each CSC on the pair will send notifications when:

- There is no connectivity at all with Netskope. No CSC is able to reach Netskope.
- At power up the CSC will notify the current “eni-xxxx” used as default GW to internet
- On routing change, the CSCs will notify the changes.

Example of notifications:

AWS Notification Message [External](#) ➤ [Inbox](#)

csc-aws-ha-notification <no-reply@sns.amazonaws.com>
to me ▾ Sun, 2 Jan, 20:00 C

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

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

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

Logs generated:

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

7 Steering traffic to NewEdge with the CSC GRE for AWS.

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

When connecting Instances, Workspaces, AppStream, etc., to Netskope using the CSC, you have two options of steering traffic: routing and proxying.

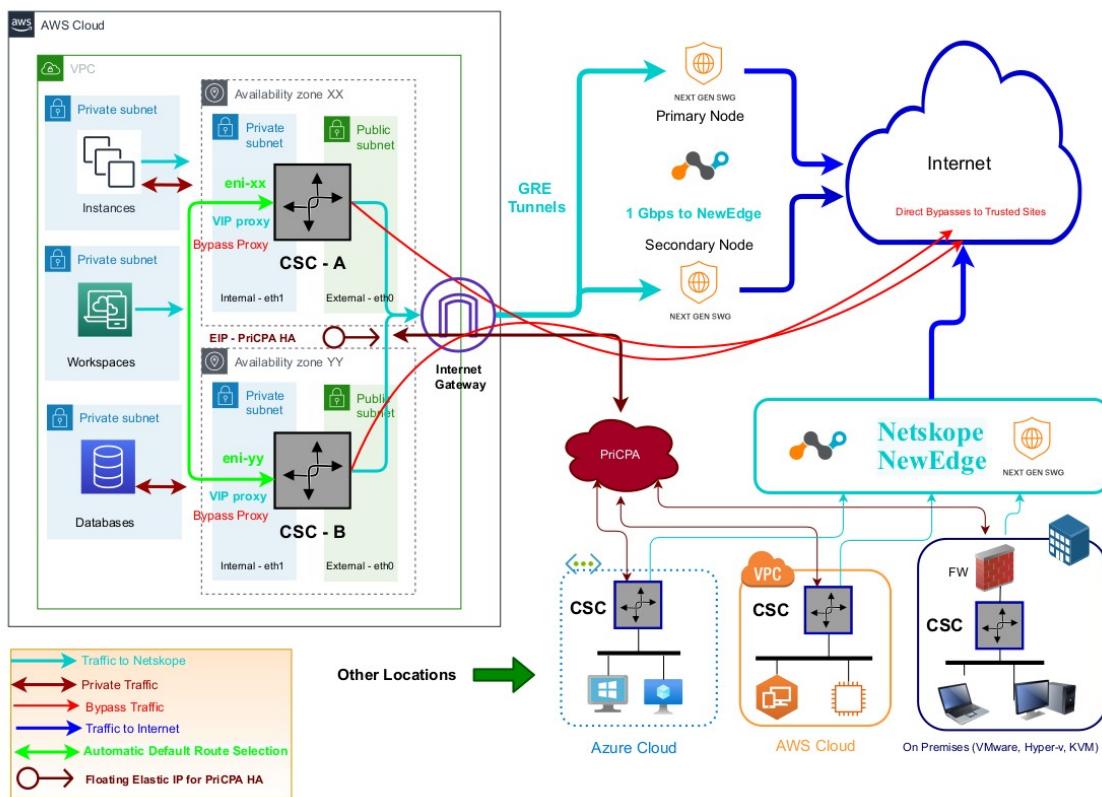
The options are not mutually exclusive. You can use both at the same time. Moreover, when the CSC is on a HA pair, you can use both simultaneously, duplicating the capacity for Web Traffic to 2 Gbps.

In both cases, it is possible to bypass traffic from the tunnel to Netskope and send it directly using the "routed" and "proxied" bypass functionalities.

This chapter will dig into more detail about the configuration required, showing examples when the CSC is on High Availability and when using a single CSC.

7.1 CSC on HA Pair

7.1.1 Network Diagram



7.1.2 Prerequisites

1. Deploy 2 x CSC as HA Pair. (See section "6.6.4 High Availability configuration")
2. Create the Routed Bypasses. (See section "6.4 Routed Bypass")
3. Obtain the VIP Proxy and Bypass Proxy of each CSC running "Show Configuration and Status" from SSH console or AWS Systems Manager.
4. Created the Proxy Bypasses. (See section "6.3 Proxy Bypass")

7.1.3 Routing traffic via the CSC HA pair.

7.1.3.1 *Traffic to Netskope*

The "High Availability" setup will manage the Target of the default route to the internet (0.0.0.0/0) on all route tables configured. Nothing extra is required. Your only task is to attach the Subnets to the Route Tables.

7.1.3.2 *"Routed Bypass" traffic*

The "Routed Bypass" functionality will do the task. Your task is to create the JSON file and to add the Routed Bypasses to it.

7.1.4 Proxy traffic via the CSC HA Pair.

7.1.4.1 *Using PAC files*

You can manage the traffic "to Netskope" and "Proxy Bypass" on a single PAC file when using PAC files.

You have three options:

1. Use both CSC at the same time doing Load Balancing per Source IP. With this method, you can achieve 2 Gbps for Web traffic.
2. Use Netskope's Global Proxy: VIP 163.116.128.80:80, Bypass 163.116.128.80:3128
3. Use CSC's VIP and Bypass Addresses as Primary and Secondary proxy.

7.1.4.1.1 *PAC file for Load Balancing*

See Section 2) of the PAC file below. In this section, the Source IP of the device is read on the variable "nicIP".

The values of variables "tonetskope" and "bypassproxy" are assigned by odd or even values of "nicIP".

PAC file with Load Balancing

```
function FindProxyForURL(url, host) {
// =====
// Section 1: 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";

// =====
// Section 2: Define Variables

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

// Assigning values to "tonetskope" and "bypassproxy"
if (isInNet(nicIp, "0.0.0.0", "0.0.0.1")) {
    var tonetskope = "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 tonetskope = "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.
return tonetskope
}
```

7.1.4.1.2 PAC file using Netskope's Global Proxy

See section 2 of the PAC file below. This section defines the use of IP: 163.116.128.80 as a proxy for both "tonetskope" and "bypassproxy", with different ports for each one.

How does this work? The IP 163.116.128.80 will be routed via the default route to the internet (0.0.0.0/0) to the eni-xx of the CSC active of the HA pair. The CSC can intercept the IP 163.116.128.80 and redirect the traffic to the tunnel if the port is 80 and bypass when the port is 3128.

```
PAC file using Netskope's Global Proxy

function FindProxyForURL(url, host) {
// =====
// Section 1: 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";

// =====
// Section 2: Define Variables

var tonetskope = "PROXY 163.116.128.80:80";
var bypassproxy = "PROXY 163.116.128.80:3128";

// =====
// 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.
return tonetskope
}
```

7.1.4.1.3 PAC file using CSC's VIP and Bypass Proxy Address (Pri/Sec)

PAC file using CSC's VIP and Bypass Proxy Address for Primary and Secondary Proxy.

```
function FindProxyForURL(url, host) {
// =====
// Section 1: 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";

// =====
// Section 2: Define Variables

var tonetskope = "PROXY csc1vip:80; PROXY csc2vip:80";
var bypassproxy = "PROXY csc1bypass:3128; PROXY csc2bypass: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.
return tonetskope
}
```

7.1.4.2 Using Explicit Proxy on devices that cannot support PAC files.

For devices that cannot be configured with PAC files, you need to set up an Explicit Proxy (IP:Port) and exclusions.

The recommendation, in this case, is to use the Netskope's Global Proxy IP (163.116.128.80) for Explicit Proxy and to allow the exclusion to do direct to the Internet using "Routing Bypasses".

Here an example of a Linux Server.

Settings Variables for http, https and no_proxy⁴

```
export5 http_proxy=http://163.116.128.80:80  
export https_proxy=http://163.116.128.80:80  
export no_proxy=portquiz.net
```

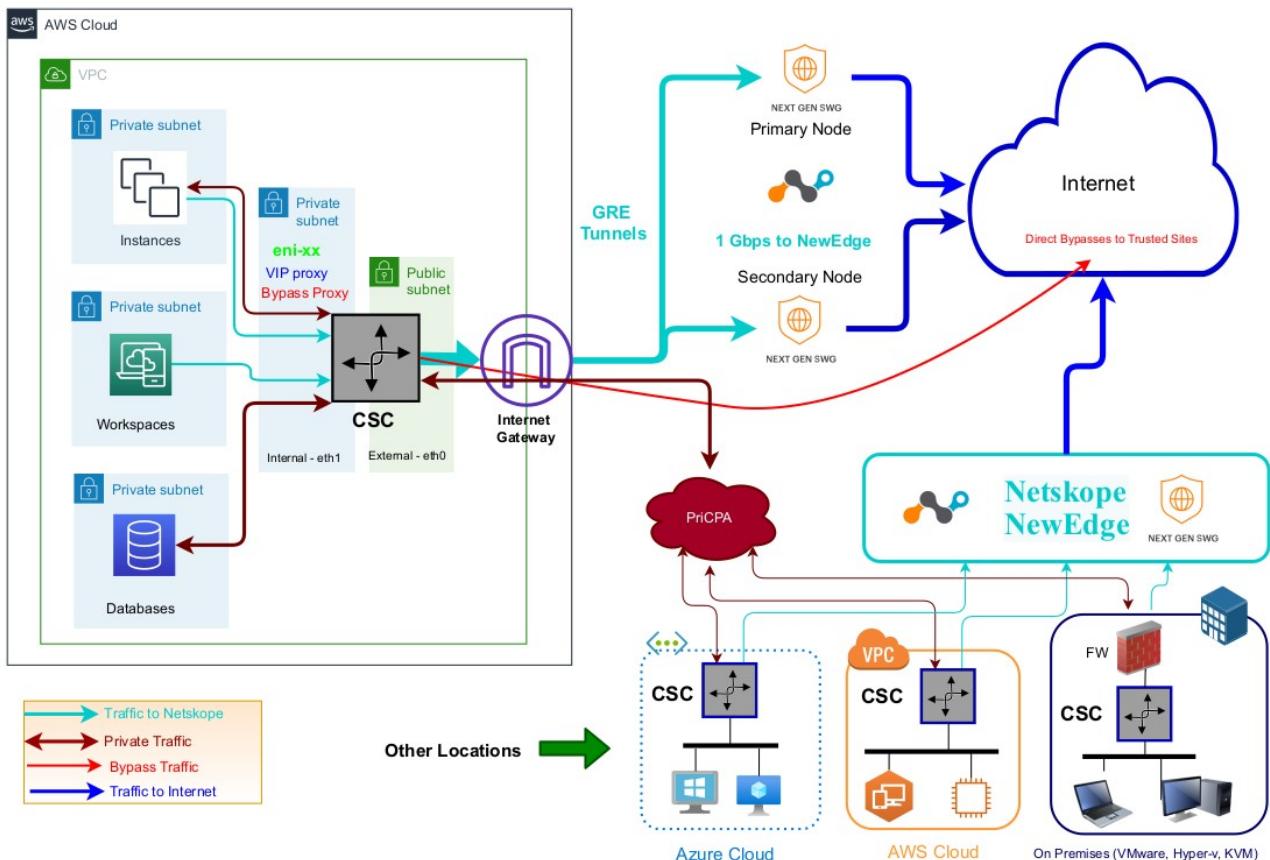
Test	Return	Observations
curl http://www.notskope.com	163.116.146.120 Ashburn, United States (IAD2)	OK. http via Netskope.
curl https://ip.maidenheadbridge.com	163.116.146.120	OK. https via Netskope
curl http://portquiz.net	Port 80 test successful! Your IP: 54.80.198.195	OK. portquiz.net via bypass using "Routed Bypass"

⁴ Add this lines to "/etc/environment" to make this changes permanent.

⁵ Use command \$unset <variable name> to clear the values.

7.2 CSC Single

7.2.1 Network Diagram



7.2.2 Prerequisites

1. Obtain the eni-xx value of the Internal Interface of the CSC running "Show Configuration and Status" from SSH console or AWS Systems Manager.
2. Create the Routed Bypasses. (See section "6.4 Routed Bypass")
3. Obtain the VIP Proxy and Bypass Proxy of each CSC running "Show Configuration and Status" from SSH console or AWS Systems Manager.
4. Created the Proxy Bypasses. (See section "6.3 Proxy Bypass")

7.2.3 Routing traffic via the CSC Single.

7.2.3.1 *Traffic to Netskope*

On your Route Tables, configure the default route to the Internet (0.0.0.0/0) with "Target" the eni-xx of the CSC.

7.2.3.2 *"Routed Bypass" traffic*

The "Routed Bypass" functionality will do the task. Your task is to create the JSON file and to add the Routed Bypasses to it.

7.2.4 Proxy traffic via the CSC Single.

7.2.4.1 *Using PAC files*

You can manage the traffic "to Netskope" and "Proxy Bypass" on a single PAC file when using PAC files.

You have two options:

1. Use Netskope's Global Proxy: VIP 163.116.128.80:80, Bypass 163.116.128.80:3128
2. Use CSC's VIP and Bypass Addresses as Primary

7.2.4.1.1 PAC file using Netskope's Global Proxy

See section 2 of the PAC file below. This section defines the use of IP: 163.116.128.80 as a proxy for both "tonetskope" and "bypassproxy", with different ports for each one.

How does this work? The IP 163.116.128.80 will be routed via the default route to the internet (0.0.0.0/0) to the eni-xx of the CSC. The CSC can intercept the IP 163.116.128.80 and redirect the traffic to the tunnel if the port is 80 and bypass when the port is 3128.

PAC file using Netskope's Global Proxy

```
function FindProxyForURL(url, host) {
// =====
// Section 1: 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) || isnInNet(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";

// =====
// Section 2: Define Variables

var tonetskope = "PROXY 163.116.128.80:80";
var bypassproxy = "PROXY 163.116.128.80:3128";

// =====
// 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.
return tonetskope
}
```

7.2.4.1.2 PAC file using CSC's VIP and Bypass Proxy Address (Pri/Sec)

PAC file using CSC's VIP and Bypass Proxy Address for Primary and Secondary Proxy.

```
function FindProxyForURL(url, host) {
// =====
// Section 1: 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";

// =====
// Section 2: Define Variables

var tonetskope = "PROXY cscvip:80";
var bypassproxy = "PROXY cscbypass: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.
return tonetskope
}
```

7.2.4.2 Using Explicit Proxy on devices that cannot support PAC files.

For devices that cannot be configured with PAC files, you need to set up an Explicit Proxy (IP:Port) and exclusions.

With a single CSC, you have to options:

1. Use the Netskope's Global Proxy IP (163.116.128.80) for Explicit Proxy and to allow the exclusion to do direct to the Internet using "Routing Bypasses".
2. Use CSC VIP IP for Explicit Proxy and to allow the exclusion to do direct to the Internet using "Routing Bypasses".

Here an example of a Linux Server using the Netskope's Global Proxy IP (163.116.128.80)

Settings Variables for http, https and no_proxy⁶

```
export7 http_proxy=http://163.116.128.80:80  
export https_proxy=http://163.116.128.80:80  
export no_proxy=portquiz.net
```

Test	Return	Observations
curl http://www.notskope.com	163.116.146.120 Ashburn, United States (IAD2)	OK. http via Netskope.
curl https://ip.maidenheadbridge.com	163.116.146.120	OK. https via Netskope
curl http://portquiz.net	Port 80 test successful! Your IP: 54.80.198.195	OK. portquiz.net via bypass using "Routed Bypass"

⁶ Add this lines to "/etc/environment" to make this changes permanent.

⁷ Use command \$unset <variable name> to clear the values.

7.3 Testing traffic to Netskope

7.3.1 www.notskope.com

The page www.notskope.com shows the Netskope Datacenter.

Using the browser:

https://www.notskope.com

notskope.com

Apps Outgoing Port Tester https://www.notsko... https://ip.maidenhe... Welcome page

netskope Node

You are using a Netskope NewEdge data center in London, United Kingdom (LON1)

Note: For this utility to work you must be steering the notskope.com domain to Netskope

Your source IP address is 163.116.162.117

Your XFF IP address is 82.68.6.74 **GRE IP**

Last update 9th October 2021

Using curl command from CMD or Terminal

Proxy environment:

Command	curl --proxy http://<CSC VIP>:80 www.notskope.com (i.e. \$curl --proxy http://172.19.0.61:80 www.notskope.com)
Expected Result	<NewEdge Node IP> <City>,<Country> (<NodeID> (i.e. 163.116.162.117 London, United Kingdom (LON1))

Routed environment:

Command	curl www.notskope.com (i.e. \$curl www.notskope.com)
Expected Result	<NewEdge Node IP> <City>, <Country> (<NodeID> (i.e. 163.116.162.117 London, United Kingdom (LON1))

7.3.2 https://ip.maidenheadbridge.com

Maidenhead Bridge provides a HTTPS page to check the IP.

Using the Browser:



Using curl command from CMD or Terminal

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

Proxy environment:

Command	curl -k --proxy http://<CSC VIP>:80 https://ip.maidenheadbridge.com (i.e. \$curl --proxy http://172.19.0.61:80 https://ip.maidenheadbridge.com)
Expected Result	<Netskope Node IP> (i.e. 163.116.162.117)

Routed environment:

Command	curl -k https://ip.maidenheadbridge.com (i.e. \$curl -k https://ip.maidenheadbridge.com)
Expected Result	<Netskope Node IP> (i.e. 163.116.162.117)

7.3.3 SpeedTest

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

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

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

Retrieving speedtest.net configuration...
Testing from Netskope (163.116.146.119)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by Netprotect (Ashburn, VA) [0.81 km]: 5.564 ms
Testing download speed.....
Download: 1317.16 Mbit/s
Testing upload speed.....
Upload: 1137.61 Mbit/s
```

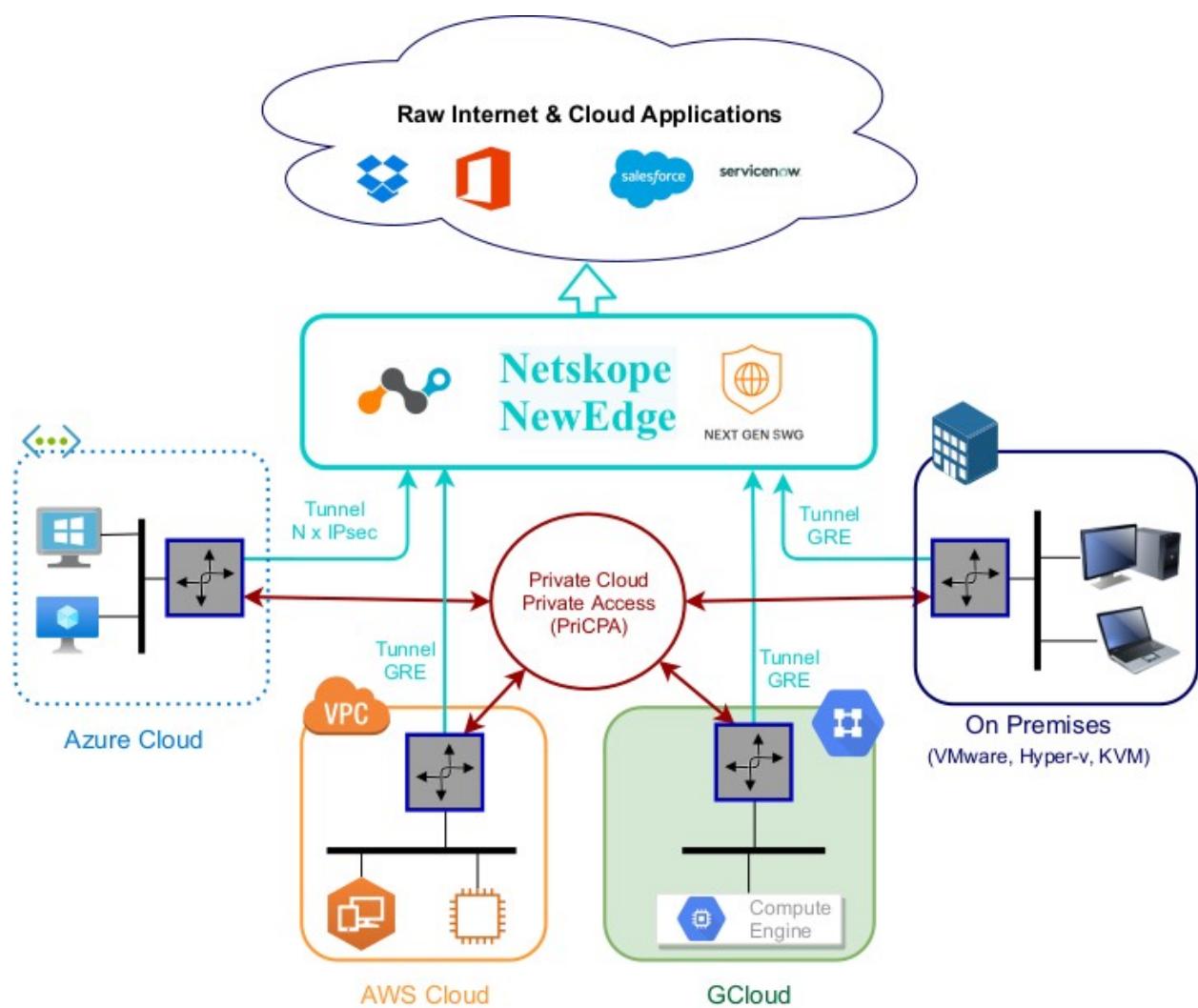
8 Private Cloud Private Access

8.1 What is Private Cloud Private Access (PriCPA)?

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

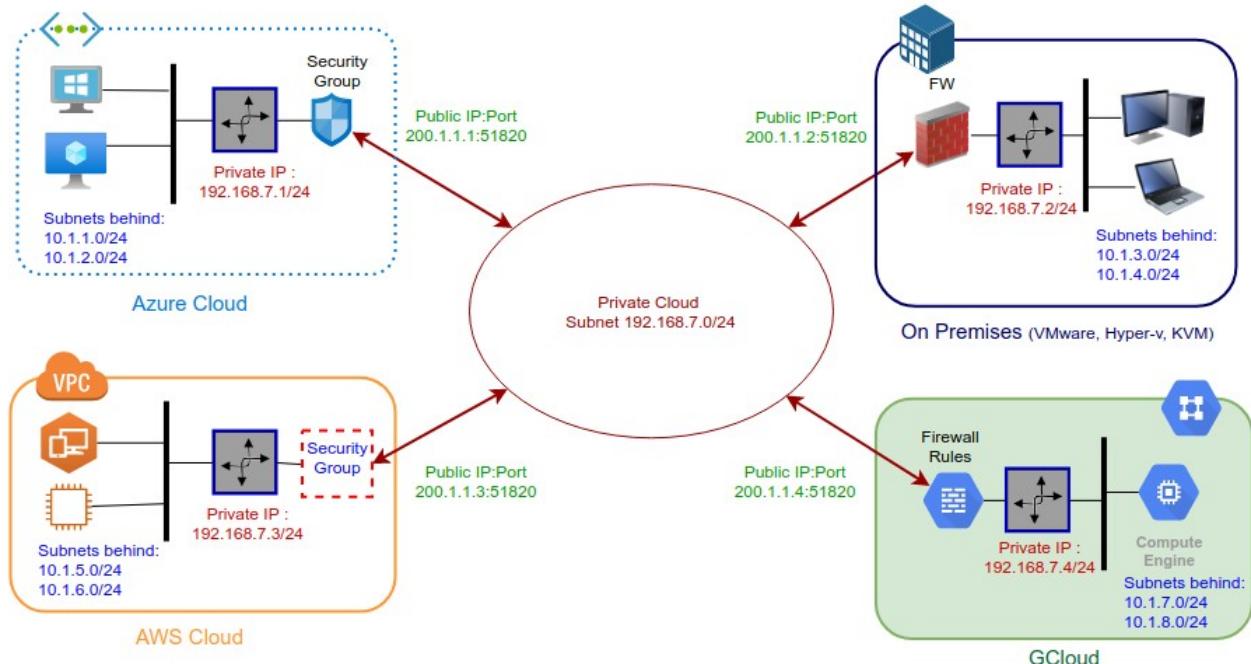
8.2 PriCPA Network Diagrams

8.2.1 High Level Network Diagram



8.2.2 Low Level Network Diagram – PriCPA only

The following network diagram shows the IP addressing for PriCPA.



Steps to design your Private Cloud:

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

8.3 Configuring PriCPA

The Main Menu has a section for Private Access:

```
MHB Labs - Private Access - (Preview)
18) Show Configurations and Status Private Access.
19) Configure Private Access.
20) Configure CSC Remote Management via Private Access.
```

The configuration of PriCPA is four simple steps:

1. Create the Local Node configuration. This step will initialize and enable Private Access on the Node. The result of this operation will show a "Token" and "Private Access Local JSON file".
2. (HA Pair only) Initialize the second Node of the HA pair using the "Token" and "Private Access Local JSON file".
3. Create an IAM role with the following permissions and apply it to the CSCs:

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

4. Create and distribute the Private Access Peers JSON file to all nodes.

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

8.3.1 Create the Local configuration (first node of the cluster)

- From Main Menu, select "19) Configure Private Access."

```
Selection: 19

Private Access Configuration Wizard

Steps to configure Private Access:

-> Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
-> (optional, if HA is enabled.) Copy Local Configuration to the other CSC in the HA pair.
-> Create an IAM role with the following permissions and apply it to the CSC:

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

-> Load Private Access Peers JSON configuration file.

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

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

```

Enter your choice: 1
Private Access is not enabled.

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

Do you want to enable Private Access?

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

```

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

```

Enter your choice: 1
Before continuing, you need to have the following values ready:
  - Node Name. (string)
    - (Optional) Location Name. (string)
    - (Optional) Description. (string)
    - Public IP and UDP Port. (IP:Port)
    - Private IP/Subnet of Local Interface. (IP/Subnet Prefix)

Do you want to continue?
1) Yes
2) No
Enter your choice: 1

Please, input the following values:
Node Name (string): csc-gre-for-netskope-on-aws
(Optional) Location Name (string): aws-us-east-1
(Optional) Description (string): HA Pair VPC 172.31.0.0/16
Public IP and UDP port (IP:Port): 54.80.198.195:51820
Private IP/Subnet of Local Interface (IP/Subnet Prefix): 192.168.7.89/24
CSC Single: Use here "Bypass Proxy Public IP"
CSC HA: Use here "Private Access Public IP" configured.

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

IMPORTANT: We strongly recommend keeping the default value of 'Persistent KeepAlive = no'. Enabling 'Persistent KeepAlive' generates unnecessary traffic and consumes CPU resource
Do you want to change default value of 'Persistent KeepAlive = no'?
1) Yes
2) No (Recommended)
Enter your choice: 2 Leave default.

The values to configure are:
Node Name: csc-gre-for-netskope-on-aws
Public IP and UDP Port: 54.80.198.195:51820
Private IP/Subnet of Local Interface: 192.168.7.89/24
Location Name: aws-us-east-1
Description: HA Pair VPC 172.31.0.0/16
Persistent KeepAlive: no

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

```

➤ Apply values

Enter your choice: 1

Private Access - Private Access service is enabled on i-0283558d4cfb35311.

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

Token: b0I1bFZBK09TV3ViWkFuL3dZ0mUyTWUrZnQwQ2ljU1RkY0xrdWdkeWlVbzOK

Private Access Local Config JSON file:

```
{
  "peers": [
    {
      "nodeName": "csc_gre-for-netskope-on-aws",
      "location": "aws-us-east-1",
      "description": "HA Pair VPC 172.31.0.0/16",
      "publicKey": "Z26FHcsMATHhC7cDXj0hdUS61LPKE90McA33KRYz0c=",
      "publicIpAndUdpPort": "54.80.198.195:51820",
      "privateCidrIp": "192.168.7.89/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

Press Enter to continue...

IMPORTANT: Keep this information in a safe place.

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

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

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

Go to 19) Configure Private Access. → 1) Create (or change) Private Access Local Configuration → 2) Token and JSON

```

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

Private Access is not enabled.

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

Do you want to enable Private Access?

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

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

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

Do you want to continue?

1) Yes
2) No
Enter your choice: 1

```

```

Do you want to continue?
1) Yes
2) No
Enter your choice: 1 1
Please, input the following values:
Token (string): b0I1bFZBK09TV3ViWkFuL3dZQmUyTWUrZnQwQ2ljU1RKY0xrdWdkeWLVbz0K 2
Please, paste 'Private Access Local Config JSON file' and press 'Enter' if required.
NOTE: If the json file has errors, it is possible that the script will hang. Press '}' and 'Enter' to end the operation.

Private Access Local Config JSON file: {
  "peers": [
    {
      "nodeName": "csc-gre-for-netskope-on-aws",
      "location": "aws-us-east-1",
      "description": "HA Pair VPC 172.31.0.0/16",
      "publicKey": "2Z6bFHcsMATHhC7cdXu0hdUS6lLPKE90McA33KRYz0c=",
      "publicIpAndUdpPort": "54.80.198.195:51820",
      "privateCirdIp": "192.168.7.89/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
} 3

Private Access Local Config JSON file imported successfully

The values to configure are:
Node Name: "csc-gre-for-netskope-on-aws"
Public IP and UDP Port: 54.80.198.195:51820
Private IP/Subnet of Local Interface: 192.168.7.89/24
Location Name: "aws-us-east-1"
Description: "HA Pair VPC 172.31.0.0/16"
Persistent KeepAlive: no 4

Do you want to apply this values?
1) Yes
2) No
Enter your choice: 1 5
Private Access - Private Access service is enabled on i-08cd4663125d25b0e. 6

```

8.3.3 Create the Private Access Peers JSON file

The Private Access Peers JSON file contains:

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

Here some examples.

8.3.3.1 Full mesh Private Access Peers JSON file

Consider the following example:

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

The Local Config JSON file of each node is:

ns-cgc00001

```
{  
  "peers": [  
    {  
      "nodeName": "ns-cgc00001",  
      "description": "Node on VMware Server 1",  
      "location": "HQ",  
      "publicKey": "yAnz5TF+IXXJte14tji3zlMNq+hd2rYUlgJBgB3fBmk=",  
      "publicIpAndUdpPort": "200.1.1.1:51821",  
      "privateCirdIp": "192.168.7.1/24",  
      "persistentKeepAlive": "no",  
      "networks": [],  
      "privateApps": []  
    }  
  ]  
}
```

ns-cgc00002

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

ns-cgc00003

```
{  
  "peers": [  
    {  
      "nodeName": "ns-cgc00003",  
      "description": "Node on VMware Server 3",  
      "location": "Branch",  
      "publicKey": "TrMvSoP4jYQlY6RlzBgbssQqY3vxI2Pi+y71lOWWX0=",  
      "publicIpAndUdpPort": "200.1.1.3:51821",  
      "privateCirdIp": "192.168.7.3/24",  
      "persistentKeepAlive": "no",  
      "networks": [],  
      "privateApps": []  
    }  
  ]  
}
```

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

Basic Peers Configuration JSON file

```
{  
  "peers": [  
    {  
      "nodeName": "ns-cgc00001",  
      "description": "Node on VMware Server 1",  
      "location": "HQ",  
      "publicKey": "yAnz5TF+IXXJte14tji3zIMNq+hd2rYUlgJBgB3fBmk=",  
      "publicIpAndUdpPort": "200.1.1.1:51821",  
      "privateCirdIp": "192.168.7.1/24",  
      "persistentKeepAlive": "no",  
      "networks": [  
        "10.1.1.0/24",  
        "10.1.2.0/24"  
      ],  
      "privateApps": []  
    },  
    {  
      "nodeName": "ns-cgc00002",  
      "description": "Node on VMware Server 2",  
      "location": "Datacentre 2",  
      "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",  
      "publicIpAndUdpPort": "200.1.1.2:51821",  
      "privateCirdIp": "192.168.7.2/24",  
      "persistentKeepAlive": "no",  
      "networks": [  
        "10.2.1.0/24",  
        "10.2.2.0/24"  
      ],  
      "privateApps": []  
    },  
    {  
      "nodeName": "ns-cgc00003",  
      "description": "Node on VMware Server 3",  
      "location": "Branch",  
      "publicKey": "TrMvSoP4jYQlY6RlzBgbssQqY3vxI2Pi+y71lOWWWXX0=",  
      "publicIpAndUdpPort": "200.1.1.3:51821",  
      "privateCirdIp": "192.168.7.3/24",  
      "persistentKeepAlive": "no",  
      "networks": [  
        "10.3.1.0/24",  
        "10.3.2.0/24"  
      ],  
      "privateApps": []  
    }  
  ]  
}
```

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

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

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

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

ns-cgc00001
{ "nodeName": "ns-cgc00001", "description": "Node on VMware Server 1", "location": "HQ", "publicKey": "yAnz5TF+IXXJte14tji3zIMNq+hd2rYUlgJBgB3fBmk=", "publicIpAndUdpPort": "200.1.1.1:51821", "privateCirdIp": "192.168.7.1/24", "persistentKeepAlive": "no", "networks": ["10.1.1.0/24", "10.1.2.0/24"], "privateApps": [{ "description": "Allow all traffic to this site", "ipProtocol": "all", "sourceCirdIp": ["0.0.0.0/0"], "destinationCirdIp": ["10.1.1.0/24", "10.1.2.0/24"], "destinationSinglePorts": [""], "destinationPortRange": { "fromPort": "", "toPort": "" } }] },

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

Now, completing our "Peers Configuration JSON file":

Full Mesh Peers Configuration JSON file.

```
{  
  "peers": [  
    {  
      "nodeName": "ns-cgc00001",  
      "description": "Node on VMware Server 1",  
      "location": "HQ",  
      "publicKey": "yAnz5TF+IXXJte14tjI3zIMNq+hd2rYUlgJBgB3fBmk=",  
      "publicIpAndUdpPort": "200.1.1.1:51821",  
      "privateCirdIp": "192.168.7.1/24",  
      "persistentKeepAlive": "no",  
      "networks": [  
        "10.1.1.0/24",  
        "10.1.2.0/24"  
      ],  
      "privateApps": [  
        {  
          "description": "Allow all traffic to this site",  
          "ipProtocol": "all",  
          "sourceCirdIp": [  
            "0.0.0.0/0"  
          ],  
          "destinationCirdIp": [  
            "10.1.1.0/24",  
            "10.1.2.0/24"  
          ],  
          "destinationSinglePorts": [  
            ""  
          ],  
          "destinationPortRange": {  
            "fromPort": "",  
            "toPort": ""  
          }  
        }  
      ]  
    },  
    {  
      "nodeName": "ns-cgc00002",  
      "description": "Node on VMware Server 2",  
      "location": "Datacentre 2",  
      "publicKey": "xTIBASrboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",  
      "publicIpAndUdpPort": "200.1.1.2:51821",  
      "privateCirdIp": "192.168.7.2/24",  
      "persistentKeepAlive": "no",  
      "networks": [  
        "10.2.1.0/24",  
        "10.2.2.0/24"  
      ],  
      "privateApps": [  
        {  
          "description": "Allow all traffic to this site",  
          "ipProtocol": "all",  
          "sourceCirdIp": [  
            "0.0.0.0/0"  
          ],  
          "destinationCirdIp": [  
            "10.2.1.0/24",  
            "10.2.2.0/24"  
          ],  
          "destinationSinglePorts": [  
            ""  
          ],  
          "destinationPortRange": {  
            "fromPort": "",  
            "toPort": ""  
          }  
        }  
      ]  
    },  
    {  
      "nodeName": "ns-cgc00003",  
      "description": "Node on VMware Server 3",  
      "location": "Branch",  
      "publicKey": "TrMVs0P4jYQlY6RlzBgbssQqY3vxI2Pi+y71lOWWXX0=",  
      "publicIpAndUdpPort": "200.1.1.3:51821",  
      "privateCirdIp": "192.168.7.3/24",  
      "persistentKeepAlive": "no",  
      "networks": [  
        "10.3.1.0/24"  
      ]  
    }  
  ]  
}
```

```

    "10.3.2.0/24"
],
"privateApps": [
{
  "description": "Allow all traffic to this site",
  "ipProtocol": "all",
  "sourceCirdIp": [
    "0.0.0.0/0"
  ],
  "destinationCirdIp": [
    "10.3.1.0/24",
    "10.3.2.0/24"
  ],
  "destinationSinglePorts": [
    ""
  ],
  "destinationPortRange": {
    "fromPort": "",
    "toPort": ""
  }
}
]
}

```

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

8.3.3.2 Understanding "privateApps" configuration and values

Question 1: Where to configure the "privateApps"?

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

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

Question 2 : What about the values to configure?

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

Accepts single value only -> ""

Accepts single or multiple values -> []

```

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

```

Examples:

Single value (""):

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

Single or Multiple values ([]):

```
"sourceCirdIp": ["0.0.0.0/0"],  
  
"destinationCirdIp": ["10.1.1.100/32", "10.1.2.100/32"],  
"destinationSinglePorts": [ "80", "443" ],
```

The following table shows all field and values accepted

Field	Value Type	Values to configure	Example
"description": "",	Single	String	"description": "Intranet Server Access",
"ipProtocol": "",	Single	tcp,udp,icmp or all	"ipProtocol": "tcp",
"sourceCirdIp": [],	Single or Multiple	Networks in the range of: 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 and 0.0.0.0/0	"sourceCirdIp": ["10.2.1.0/24", "10.2.2.0/24", "10.3.1.0/24", "10.3.2.0/24"],
"destinationCirdIp": [],	Single or Multiple	Networks in the range of ⁸ : 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16	"destinationCirdIp": ["10.1.1.100/32", "10.1.1.200/32"],
"destinationSinglePorts": [],	Single or Multiple	Single Port of the range 1 to 65535	"destinationSinglePorts": ["80", "443"],
"destinationPortRange": { "fromPort": "", "toPort": "" }	Single	Single Port of the range 1 to 65535	"destinationPortRange": { "fromPort": "3780", "toPort": "3784" }

⁸ The expected value here is a value that belongs to the "network" defined behind the CSC. For example, of the network behind the CSC is 10.1.1.0/24, any destination configured must belong to 10.1.1.0/24, like 10.1.1.100/32.

8.3.3.3 Example of "privateApps" for a Windows Domain controller

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

The port information was taken from this article:

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

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

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

8.3.3.4 Example of "privateApps" for Internal Web Server.

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

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

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

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

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

There are three methods available:

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

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

8.3.4.1 *Using "Private Access Peers URL"*

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

1. Place the Private Access Peers JSON file on an internal web server or an AWS bucket⁹ or similar. Obtain the download URL.

Example of AWS bucket:

Amazon S3 > mhb-netskope-private > privateAccessPeersConfig-LAB2.json

privateAccessPeersConfig-LAB2.json [Info](#)

[Copy S3 URI](#) [Download](#) [Open](#) [Object act](#)

[Properties](#) [Permissions](#) [Versions](#)

Object overview

Owner	sales
AWS Region	EU (Ireland) eu-west-1
Last modified	November 20, 2021, 10:26:09 (UTC+00:00)
Size	6.6 KB
Type	json
Key	privateAccessPeersConfig-LAB2.json

S3 URI
s3://mhb-netskope-private/privateAccessPeersConfig-LAB2.json

Amazon Resource Name (ARN)
arnaws:s3:::mhb-netskope-private/privateAccessPeersConfig-LAB2.json

Entity tag (Etag)
d3aeba11009b98bf5622d9b948f151d9

Object URL
<https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json>

2. Configure the URL on each CSC.

Ssh the each CSC and go to Main Menu -> 19) Configure Private Access

⁹ See Appendix D to learn how to secure an AWS S3 bucket by Source IP.

```

MHB Labs - Private Access - (Preview)
18) Show Configurations and Status Private Access.
19) Configure Private Access.
20) Configure CSC Remote Management via Private Access.

e) Exit

Selection: 19 | 1

Private Access Configuration Wizard

Steps to configure Private Access:

-> Create Private Access Local Configuration. (This selection also allows to change Local Configuration)
-> (optional, if HA is enabled.) Copy Local Configuration to the other CSC in the HA pair.
-> Create an IAM role with the following permissions and apply it to the CSC:

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

-> Load Private Access Peers JSON configuration file.

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

```

```

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

Private Access is enabled.

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

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

Do you want to configure the Private Acces Peers URL?
1) Yes
2) No
Enter your choice: 1 | 3 | 4

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

Do you want to refresh the Private Access Peers List?
1) Yes
2) No
Enter your choice: 1 | 5

Private Access Peers JSON file imported successfully

```

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

```
Private Access Peers JSON file imported successfully

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

Private Apps Review

Index: 0, NodeName: ns-csc-gre-aws-v-0-4, Location: vpc-10-3-0-0, publicIpAndUdpPort: 52.4.62.40:51820, privateCirdIp: 192.168.7.88/24, Private Apps Qty: 3
Index: 1, NodeName: csc-aws-v-0-1, Location: us-east-1, publicIpAndUdpPort: 34.230.146.174:51820, privateCirdIp: 192.168.7.100/24, Private Apps Qty: 0
Index: 2, NodeName: csc-gre-for-neteskope-on-aws, Location: aws-us-east-1, publicIpAndUdpPort: 54.20.193.195:51820, privateCirdIp: 192.168.7.89/24, Private Apps Qty: 3
Index: 3, NodeName: ns-cgc000004, Location: mhb-bh-dc, publicIpAndUdpPort: 82.68.6.76:51820, privateCirdIp: 192.168.7.21/24, Private Apps Qty: 1
Index: 4, NodeName: ns-cgc000005, Location: mhb-dc-kvm, publicIpAndUdpPort: 82.68.6.76:51820, privateCirdIp: 192.168.7.21/24, Private Apps Qty: 0
Index: 5, NodeName: ns-cgc000006, Location: mhb-bh-dc, publicIpAndUdpPort: 217.155.196.81:51820, privateCirdIp: 192.168.7.20/24, Private Apps Qty: 2

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

Private Apps Created

Creating Private Apps:
Private Access - [(Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SSH and RDP to 10.3.200.0/24' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf udp' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 2, Node: ns-csc-gre-for-neteskope-on-aws) Private App 'Allow SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 2, Node: ns-csc-gre-for-neteskope-on-aws) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 2, Node: ns-csc-gre-for-neteskope-on-aws) Private App 'Allow iperf udp' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'Intranet Server' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'Domain Controllers TCP' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'Domain Controllers UDP' was created successfully. (destinationPortRange)
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'Domain Controllers UDP' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'ICMP to 172.19.0.133' was created successfully.
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'Syslog ICMP' was created successfully.
Private Access - [(Index: 3, Node: ns-cgc000004) Private App 'Syslog tcp port' was created successfully. (destinationSinglePorts)
Private Access - [(Index: 5, Node: ns-cgc000006) Private App 'SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)

Adding Peers:
Private Access - [Node: ns-csc-gre-aws-v-0-4 added successfully.
Private Access - [Node: ns-csc-aws-v-1 added successfully.
Private Access - [Node: ns-cgc000004 added successfully.
Private Access - [Node: ns-cgc000005 added successfully.
Private Access - [Node: ns-cgc000006 added successfully.

Nodes Added

Security Group Updated.

Changes Security Group External:
Private Access - Inbound Rules: [{}FromPort": 51820, "ToPort": 51820, "IpProtocol": "udp", "IpRanges": [{"CidrIp": "217.155.196.81/32", "Description": "mhb-rule"}, {"CidrIp": "34.230.146.174/32", "Description": "mhb-rule"}], [{"CidrIp": "52.4.62.40/32", "Description": "mhb-rule"}, {"CidrIp": "82.68.6.74/32", "Description": "mhb-rule"}]] added to Security Group 'sg-0b81ea809def53ee8'
Private Access - Inbound Rules: [{}FromPort": 51820, "ToPort": 51820, "IpProtocol": "udp", "IpRanges": [{"CidrIp": "217.155.196.81/32", "Description": "mhb-rule"}, {"CidrIp": "34.230.146.174/32", "Description": "mhb-rule"}], [{"CidrIp": "52.4.62.40/32", "Description": "mhb-rule"}, {"CidrIp": "82.68.6.76/32", "Description": "mhb-rule"}]] added to Security Group 'sg-0b81ea809def53ee8'

Private Access - Private Access Peers List updated successfully.
```

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

AWS System Manager:

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

AWS Systems Manager > Run Command > Run a command

Run a command

Command document

Select the type of command that you want to run.

Search by keyword or filter by tag or attributes

Owner: Owned by me X Clear filters

Name
<input type="radio"/> Copy-AWS-RunShellScript
<input checked="" type="radio"/> MHB-CSC-Refresh-Private-Access-Peers-URL
<input type="radio"/> MHB-CSC-Refresh-Proxy-Bypass-URL

- Move down the screen and select all CSCs:

Targets
Choose a method for selecting targets.

Specify instance tags
Specify one or more tag key-value pairs to select instances that share those tags.

Choose instances manually
Manually select the instances you want to register as targets.

mi-0f3837028ad9fcdf8 X mi-0b9178c22b03ce2bf X mi-0e234f4278cd74e27 X mi-0beef6eaa71c2f0bf X

Instances

Ping status: Online X Clear filters 3

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Availability zone	Ping status
<input checked="" type="checkbox"/>	ns-cgc00006-b	mi-0f3837028ad9fcdf8	-		Online
<input checked="" type="checkbox"/>	ns-cgc00004-b	mi-0b9178c22b03ce2bf	-		Online
<input checked="" type="checkbox"/>	ns-cgc00005-a	mi-0e234f4278cd74e27	-		Online
<input checked="" type="checkbox"/>	ns-cgc00004-a	mi-0beef6eaa71c2f0bf	-		Online
<input checked="" type="checkbox"/>	ns-cgc00006-a	mi-08c465d750d2689ae	-		Online
<input checked="" type="checkbox"/>	ns-cgc00005-b	mi-0650bce2872f405c0	-		Online

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

Command ID: e7c8bfa2-e045-4df0-8216-4721be8d4249 was successfully sent!

AWS Systems Manager > Run Command > Command ID: e7c8bfa2-e045-4df0-8216-4721be8d4249

Command ID: e7c8bfa2-e045-4df0-8216-4721be8d4249

Command status

Overall status	Detailed status	# targets	# completed
Success	Success	6	6

Targets and outputs

Instance ID	Instance name	Status	Detailed Status
mi-0650bce2872f405c0	ns-cgc00005-b	Success	Success
mi-08c465d750d2689ae	ns-cgc00006-a	Success	Success
mi-0beef6eaa71c2f0bf	ns-cgc00004-a	Success	Success
mi-0e234f4278cd74e27	ns-cgc00005-a	Success	Success
mi-0b9178c22b03ce2bf	ns-cgc00004-b	Success	Success
mi-0f3837028ad9fcdf8	ns-cgc00006-b	Success	Success

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

Output on mi-0650bce2872f405c0

Step 1 - Command description and status

Status
Success

Detailed status
Success

Step name
Runscripts

Start time
Sat, 20 Nov 2021 22:39:33 GMT

▼ Output

The command output displays a maximum of 48,000 characters. You can view the complete command output in either Amazon S3 or CloudWatch Logs if:

```
Private Access - Private Access Peers JSON file imported successfully.  
Creating Private Apps  
Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Intranet Server' was created successfully.  
(destinationSinglePorts)  
Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created successfully.  
(destinationSinglePorts)
```

Using Rundeck

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

The screenshot shows the Rundeck interface with a blue network background. On the left, there's a dark sidebar with icons for Dashboard, JOBS (highlighted with a red box), and NODES. The main area is titled 'All Jobs' with a count of 20. Below it are buttons for 'Expand All' and 'Collapse All'. A list of jobs is shown, with the second item, 'Refresh Private Access Peers URL', highlighted by a red box and labeled with a red number '2'.

- Select ALL nodes and click Run.

The screenshot shows the 'Execute Job' dialog for the 'Refresh Private Access Peers URL' job. It has a title 'Execute Job' and a sub-section 'Refresh Private Access Peers URL' with the note 'This job downloads the Peers from the URL and applies the changes.' Under the 'Nodes' section, there's a checkbox 'Change the Target Nodes (6)' followed by 'Select Nodes (6)' with 'Select All' checked (highlighted with a red box) and 'Select None' uncheckable. Below is a list of nodes: ns-cgc00004-a, ns-cgc00004-b, ns-cgc00005-a, ns-cgc00005-b, ns-cgc00006-a, and ns-cgc00006-b, all with checkboxes checked (highlighted with a red box). At the bottom are 'Cancel', 'Follow execution', 'Nodes' dropdown, and a large green 'Run Job Now' button (highlighted with a red box and labeled with a red number '2').

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

The screenshot shows the Rundeck interface after the job has run. The top bar shows the project 'NS-CSC-MGMT'. The main area shows the job 'Refresh Private Access Peers URL' with a status of 'Succeeded' (highlighted with a red box). Below is a log output table:

Node	Step	Status	Start Time
ns-cgc00004-a	Command	OK	10:56:35 pm
ns-cgc00004-a	Private Access - Private Access Peers JSON file imported successfully.		
ns-cgc00004-a	Creating Private Apps		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Intranet Server' was created successfully. (destinationSinglePorts)		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created successfully. (destinationSinglePorts)		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created successfully. (destinationPortRange)		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Domain Controllers UDP' was created successfully. (destinationSinglePorts)		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Domain Controllers PING' was created successfully.		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Syslog server' was created successfully. (destinationSinglePorts)		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'ICMP to 172.19.0.133' was created successfully.		
ns-cgc00004-a	Private Access - (Index: 0, Node: ns-cgc00004) Private App 'All protocol 192.168.6.0/24' was created successfully.		
ns-cgc00004-a	Private Access - (Index: 2, Node: ns-cgc00006) Private App 'BH - SSH and RDP' was created successfully. (destinationSinglePorts)		
ns-cgc00004-a	Adding Peers:		
ns-cgc00004-a	Private Access - Node: ns-cgc00005 added successfully.		
ns-cgc00004-a	Private Access - Node: ns-cgc00006 added successfully.		
ns-cgc00004-b	All Steps OK		
ns-cgc00005-a	All Steps OK		
ns-cgc00005-b	All Steps OK		
ns-cgc00006-a	All Steps OK		
ns-cgc00006-b	All Steps OK		

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

From Main Menu, go to 19) Configure Private Access, follow the steps below and Paste the Private Access Peers Json File:

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

Private Access is enabled.

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

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

Do you want to paste the Private Access Peers JSON File?
1) Yes
2) No
Enter your choice: [1] 3

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

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

Private Access Peers JSON file:
  "peers": [
    {
      "nodeName": "ns-csc-gre-aws-v-0-4",
      "location": "vpc-10-3-0-0",
      "description": "Node en US east VPC 10.3.0.0/24",
      "Paste the JSON file here"
    }
  }

}

Private Access Peers JSON file imported successfully.

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

Private Apps:
Index: 0, NodeName: ns-csc-gre-aws-v-0-4, Location: vpc-10-3-0-0, publicIpAndUpdPort: 52.4.62.40:51820, privateCidrIp: 192.168.7.88/24, Private Apps Qty: 3
Index: 1, NodeName: csc-aws-v-0-1, Location: us-east-1, publicIpAndUpdPort: 34.230.146.174:51820, privateCidrIp: 192.168.7.100/24, Private Apps Qty: 0
Index: 2, NodeName: ns-cdc00004, Location: MHB-DC-KVM, publicIpAndUpdPort: 82.68.6.74:51820, privateCidrIp: 192.168.7.11/24, Private Apps Qty: 3
Index: 3, NodeName: ns-cdc00004, Location: MHB-DC-KVM, publicIpAndUpdPort: 82.68.6.74:51821, privateCidrIp: 192.168.7.11/24, Private Apps Qty: 6
Index: 4, NodeName: ns-cgc00005, Location: MHB-DC-KVM, publicIpAndUpdPort: 82.68.6.76:51820, privateCidrIp: 192.168.7.21/24, Private Apps Qty: 0
Index: 5, NodeName: ns-cgc00006, Location: MHB-BH-DC, publicIpAndUpdPort: 217.155.196.81:51820, privateCidrIp: 192.168.7.20/24, Private Apps Qty: 2

Do you want to apply this values?
1) Yes
2) No
Enter your choice: [1] 2
  Creating Private Apps

Creating Private Apps:
Private Access - (Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SSH and RDP to 10.3.200.0/24' was created successfully. (destinationSinglePorts)
Private Access - (Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf tcp' was created successfully. (destinationsinglePorts)
Private Access - (Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf udp' was created successfully. (destinationsinglePorts)
Private Access - (Index: 0, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)
Private Access - (Index: 2, Node: csc-gre-for-netskope-on-aws) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 2, Node: csc-gre-for-netskope-on-aws) Private App 'Allow iperf udp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Intranet Server' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Allow SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created successfully. (destinationPortRange)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Domain Controllers UDP' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'ICMP to 172.19.0.133' was created successfully.
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Syslog ICMP' was created successfully.
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Syslog port 514' was created successfully. (destinationSinglePorts)
Private Access - (Index: 5, Node: ns-cgc00006) Private App 'SSH to Servers' was created successfully. (destinationsinglePorts)
Private Access - (Index: 5, Node: ns-cgc00006) Private App 'BH - SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)

Adding Peers:
Private Access - Node: ns-csc-gre-aws-v-0-4 added successfully.
Private Access - Node: csc-aws-v-0-1 added successfully.
Private Access - Node: ns-cgc00004 added successfully.
Private Access - Node: ns-cgc00005 added successfully.
Private Access - Node: ns-cgc00006 added successfully.

Nodes Added
Security Group updated

Changes Security Group External:
Private Access - Inbound Rules: [{"FromPort": 51820, "ToPort": 51820, "IpProtocol": "udp", "IpRanges": [{"CidrIp": "217.155.196.81/32", "Description": "mhb-rule"}, {"CidrIp": "34.230.146.174/32", "Description": "mhb-rule"}], ("CidrIp": "92.4.62.40/32", "Description": "mhb-rule"), ("CidrIp": "82.68.6.74/32", "Description": "mhb-rule")}, {"CidrIp": "82.68.6.76/32", "Description": "mhb-rule"}]} added to Security Group 'sg-010bb6b19ace1465'
Private Access - Outbound Rules: [{"ToPort": 51820, "FromPort": 51820, "IpProtocol": "udp", "IpRanges": [{"CidrIp": "217.155.196.81/32", "Description": "mhb-rule"}, {"CidrIp": "34.230.146.174/32", "Description": "mhb-rule"}, {"CidrIp": "52.4.62.40/32", "Description": "mhb-rule"}, {"CidrIp": "82.68.6.76/32", "Description": "mhb-rule"}]}, {"CidrIp": "82.68.6.74/32", "Description": "mhb-rule"}]} added to Security Group 'sg-010bb6b19ace1465'

Private Access - Private Access Peers List updated successfully.
```

Done!

8.4 Show Configurations and Status Private Access.

8.4.1 Via SSH console

From Main Menu, go to 18) Show Configurations and Status Private Access.

```
MHB Labs - Private Access - (Preview)
18) Show Configurations and Status Private Access.
19) Configure Private Access.
20) Configure CSC Remote Management via Private Access.

e) Exit

Selection: 18

Show Configuration and Status Private Access
Please, select an option:
1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 1
```

8.4.1.1 Show Peer/s Status

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

```
Enter your choice: 1

Please, select an option:
1) Show ALL Peers Status
2) Select Peer
3) Quit
Enter your choice: 1
```

1. Show All Peers Status

```
Enter your choice: 1

Peer 'ns-csc-gre-aws-v-0-4' (52.4.62.40:51820) -> 192.168.7.88 is Alive. Source Port OK. Using '51820'
Peer 'csc-aws-v-0-1' (34.230.146.174:51820) -> 192.168.7.100 is Alive. Source Port OK. Using '51820'
Peer 'ns-cgc00004' (82.68.6.74:51821) -> 192.168.7.11 is Alive. Source Port OK. Using '51821'
Peer 'ns-cgc00005' (82.68.6.76:51820) -> 192.168.7.21 is Alive. Source Port OK. Using '51820'
Peer 'ns-cgc00006' (217.155.196.81:51820) -> 192.168.7.20 is Alive. Source Port OK. Using '51820'
```

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

2. Select Peer

This section shows a more detailed information about the Peer.

```
1) Show All Peers Status
2) Select Peer
3) Quit
Enter your choice: [2] 1
Please, select a Peer
1) "ns-csc-gre-aws-v-0-4"
2) "csc-aws-v-0-1"
3) "ns-cgc00004"
4) "ns-cgc00005"
5) "ns-cgc00006"
6) Quit
Enter your choice: [5] 2
Peer Status:
  Peer ''ns-cgc00006'' (217.155.196.81:51820) -> 192.168.7.20 is Alive. Source Port OK. Using '51820'
Peer Counters:
  Latest Communication: Thu 6 Jan 10:25:19 UTC 2022
  Transfer: 1.1Ki received, 788 sent
Peer Configuration:
{
  "nodeName": "ns-cgc00006",
  "description": "CSC on Bournemouth branch",
  "location": "MHB-BH-DC",
  "publicKey": "B00glrseH+p3tWgk04j9rVawX2Fbgkj0d0JlyMITSmI=",
  "publicIpAndUdpPort": "217.155.196.81:51820",
  "privateCirdIp": "192.168.7.20/24",
  "persistentKeepAlive": "no",
```

8.4.1.2 Show Peers Json file (active)

This menu shows the active Private Access Peers Json file.

```
Selection: [18] 1
Show Configuration and Status Private Access
Please, select an option:
1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: [2] 2
{
  "peers": [
    {
      "nodeName": "ns-csc-gre-aws-v-0-4",
      "location": "vpc-10-3-0-0",
      "description": "Node en US east VPC 10.3.0.0/24",
      "publicKey": "mU4StCat4sWl3xVxaMXcRZjZTuP9G9l/0SL2bsFCh2o=",
      "publicIpAndUdpPort": "52.4.62.40:51820",
      "privateCirdIp": "192.168.7.88/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.3.200.0/24"
      ],
      "privateApps": [
```

8.4.1.3 Show Local Configuration

This menu shows the Local configuration of the node.

```
Selection: 18
Show Configuration and Status Private Access
Please, select an option:
1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 3

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

{
  "peers": [
    {
      "nodeName": "csc-gre-for-netskope-on-aws",
      "location": "aws-us-east-1",
      "description": "HA Pair VPC 172.31.0.0/16",
      "publicKey": "2Z6bFHcsMATHnc7cDXu0hdUS6llPKE90McA33KRYz0c=",
      "publicIpAndUpPort": "54.80.198.195:51820",
      "privateCidrIp": "192.168.7.89/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

8.4.1.4 Show Firewall Local Rules

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

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

```
Selection: 18
Show Configuration and Status Private Access
Please, select an option:
1) Show Peer/s Status
2) Show Peers Json file (active)
3) Show Local Configuration
4) Show Firewall Local Rules
5) Quit
Enter your choice: 4

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

{
  "nodeName": "csc-gre-for-netskope-on-aws",
  "localPrivateIp": "172.31.96.185",
  "inboundFirewallRules": [
    {
      "localUpdPort": "51820",
      "peersPublicSourceIP": [
        "52.4.62.40",
        "34.230.146.174",
        "82.68.6.74",
        "82.68.6.76",
        "217.155.190.81"
      ]
    }
  ],
  "outboundFirewallRules": [
    {
      "remoteUpdPort": "51820",
      "peersPublicDestinationIP": [
        "52.4.62.40",
        "34.230.146.174",
        "82.68.6.76",
        "217.155.190.81"
      ]
    },
    {
      "remoteUpdPort": "51821",
      "peersPublicDestinationIP": [
        "82.68.6.74"
      ]
    }
  ]
}
```

8.4.2 via AWS Systems Manager or Rundeck

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

8.4.2.1 AWS Systems Manager

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

The screenshot shows the AWS Systems Manager interface for a run command. The command ID is caa5bcf8-3946-4408-b394-d92dd45cb49e, and the output is for node mi-08c465d750d2689ae. The output details show a success status with a start time of Sun, 21 Nov 2021 09:46:15 GMT. The output section displays command logs for two peers: ns-cgc00004 and ns-cgc00005, both of which are alive and using port 51820.

8.4.2.2 Rundeck

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

The screenshot shows the Rundeck interface for a job titled "Show Private Access ALL Peers Status". The job status is Succeeded. The log output shows a summary of 100% complete with 6/6 nodes. The detailed log shows successful steps for nodes ns-cgc00004-a, ns-cgc00004-b, ns-cgc00005-a, ns-cgc00005-b, ns-cgc00006-a, and ns-cgc00006-b, all indicating "All Steps OK". Log entries show peer status checks for ports 51820 and 51821.

8.5 Configure CSC Remote Management via Private Access.

When the CSC is in HA, like the CSC GRE Cluster, only the active node belongs to the Private Cloud. The Standby is not. For this reason, if you want to reach the Standby node using SSH, you must configure Remote Management on both CSC of the Cluster (or HA pair).

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

The configuration will be:

```
MHB Labs - Private Access - (Preview)
18) Show Configurations and Status Private Access.
19) Configure Private Access.
20) Configure CSC Remote Management via Private Access.

e) Exit

Selection: 20

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

No Management Networks are configured.

Do you want to configure Management Networks?

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

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

Do you want to add another Management Network?

1) Yes
2) No
Enter your choice: 2

Management Networks to configure:

Management Networks Qty = 1
Management Network= 172.19.0.0/24

Do you want to apply changes?

1) Yes
2) No
Enter your choice: 1
Private Access - Management Network 172.19.0.0/24 was added on i-0283558d4cfb35311
```

9 Remote Management using AWS and Rundeck

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

9.1 AWS Systems Manager

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

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

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

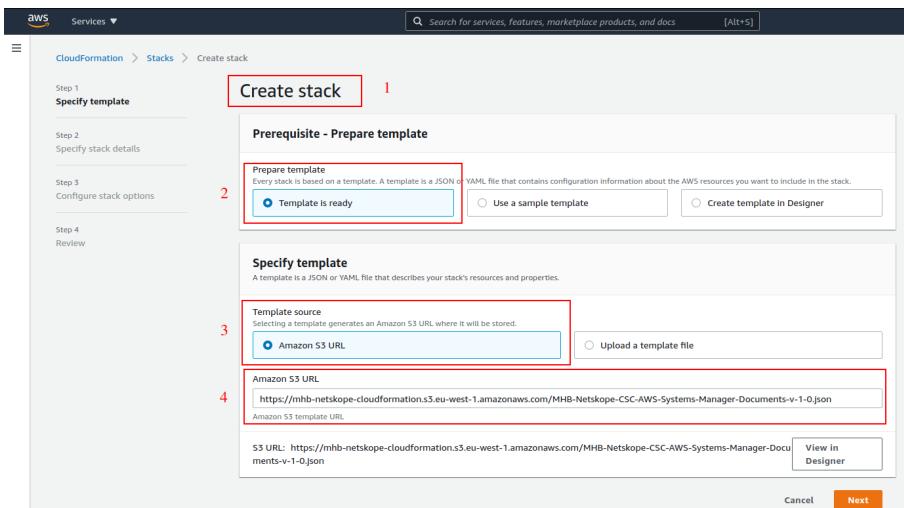
9.1.1 Create Documents

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

Steps:

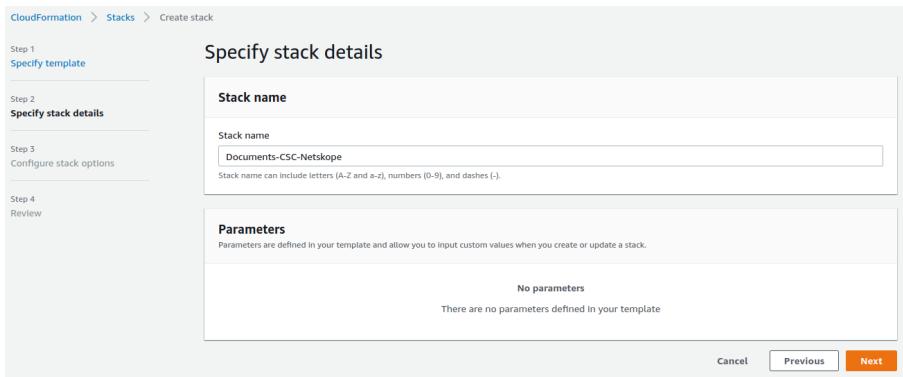
1. Download the CloudFormation template from:

<https://mhb-netskope-cloudformation.s3.eu-west-1.amazonaws.com/MHB-Netskope-CSC-AWS-Systems-Manager-Documents-v-1-1.json>



2. Deploy Stack. Go to Cloudformation → Create Stack
3. Insert the Amazon S3 URL and click next.

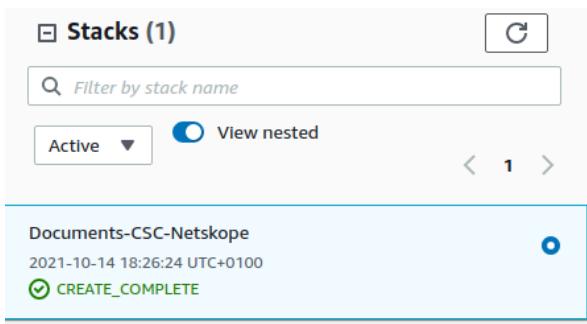
4. Put the Stack Name



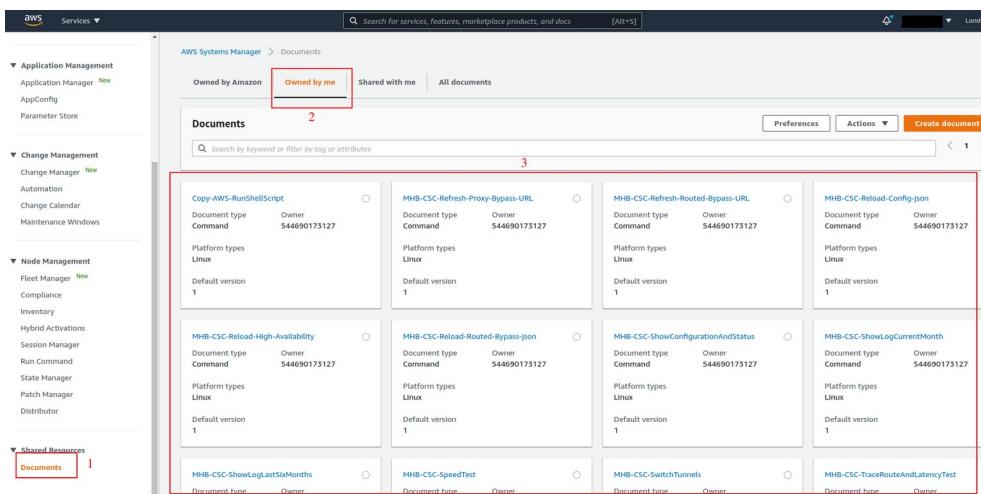
The screenshot shows the 'Specify stack details' step of the CloudFormation 'Create stack' wizard. On the left, a sidebar lists steps: Step 1 'Specify template', Step 2 'Specify stack details' (which is selected), Step 3 'Configure stack options', and Step 4 'Review'. The main area has a title 'Specify stack details' and a 'Stack name' input field containing 'Documents-CSC-Netskope'. Below it is a note: 'Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-)'. A 'Parameters' section follows, stating 'No parameters' and 'There are no parameters defined in your template'. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

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

6. Wait the Stack to complete.



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



The screenshot shows the AWS Systems Manager 'Documents' page. The left sidebar includes sections for Application Management, Change Management, Node Management, and Shared Resources (with 'Documents' highlighted). The main area shows a grid of documents under the 'Owned by me' tab. One document, 'Copy-AWS-RunShellScript', is selected and highlighted with a red box. Other documents listed include 'MHB-CSC-Refresh-Proxy-Bypass-URL', 'MHB-CSC-Refresh-Routed-Bypass-URL', 'MHB-CSC-Reload-Config.json', 'MHB-CSC-Reloader-High-Availability', 'MHB-CSC-Reload-Routed-Bypass.json', 'MHB-CSC>ShowConfigurationAndStatus', 'MHB-CSC>ShowLogCurrentMonth', 'MHB-CSC>ShowLogLastSixMonths', 'MHB-CSC-SpeedTest', 'MHB-CSC-SwitchTunnels', and 'MHB-CSC-TraceRouteAndLatencyTest'. A red box also highlights the 'Owned by me' tab in the top navigation bar.

8. Done!

9.1.2 Run Commands

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

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

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

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

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

The screenshot shows the AWS Systems Manager interface. The left sidebar has sections for Explorer, OpsCenter, CloudWatch Dashboard, and PHD. Under Application Management, there are links for Application Manager (New), AppConfig, and Parameter Store. Under Change Management, there are links for Change Manager (New), Automation, Change Calendar, and Maintenance Windows. The Node Management section is expanded, showing Fleet Manager (New), Compliance, Inventory, Hybrid Activations, Session Manager, and a red-bordered 'Run Command' button. The main content area shows the 'Run a command' page with a breadcrumb navigation: AWS Systems Manager > Run Command > Run a command. A search bar and a filter button ('Owner: Owned by me X Clear filters') are present. A list of command documents is shown, with 'MHB-CSC-ShowConfigurationAndStatus' selected and highlighted with a red border. Other documents listed include Copy-AWS-RunShellScript, MHB-CSC-Refresh-Proxy-Bypass-URL, MHB-CSC-Refresh-Routed-Bypass-URL, MHB-CSC-Reload-Config-Json, MHB-CSC-Reload-High-Availability, MHB-CSC-Reload-Routed-Bypass-Json, and MHB-CSC-ShowLogCurrentMonth.

3. Scroll down and Select the Instances

Command parameters

Targets

Choose a method for selecting targets.

Specify instance tags
Specify one or more tag key-value pairs to select instances that share those tags.

Choose Instances manually
Manually select the instances you want to register as targets.

mi-0160555d766bf22c6 × mi-0100c70a3ad29e8b5 ×

Instances

Ping status: Online × Clear filters

Name	Instance ID	Instance state	Availability zone	Ping status	Last ping
ns-cgc00002-a	mi-0160555d766bf22c6	-	-	Online	14/10/2021 17:59:34
ns-cgc00002-b	mi-0100c70a3ad29e8b5	-	-	Online	14/10/2021 17:59:34

Other parameters

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

Command ID: 17f0c6ea-d610-43cd-a900-3e0d12af4dc0 was successfully sent!

AWS Systems Manager > Run Command > Command ID: 17f0c6ea-d610-43cd-a900-3e0d12af4dc0

Command ID: 17f0c6ea-d610-43cd-a900-3e0d12af4dc0

Command status

Overall status	Detailed status	# targets
Success	Success	2

Targets and outputs

Instance ID	Instance name	Status	Detailed Status
mi-0100c70a3ad29e8b5	ns-cgc00002-b	Success	Success
mi-0160555d766bf22c6	ns-cgc00002-a	Success	Success

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

Output on mi-0100c70a3ad29e8b5

Step 1 - Command description and status

Status	Detailed status	Response code
Success	Success	0

Step name: Runscripts
Start time: Thu, 14 Oct 2021 17:59:34 GMT
Finish time: Thu, 14 Oct 2021 17:59:34 GMT

Output

The command output displays a maximum of 48,000 characters. You can view the complete command output in either Amazon S3 or CloudWatch Logs, if you specify an S3 bucket or CloudWatch log stream in the command parameters.

```

GENERAL INFORMATION
Company : Maidenhead Bridge
Location : HQKRM
CSC ID : ns-cgc00002-b
CSC date: Thu 14 Oct 18:59:34 BST 2021
Soft version : 1.0

```

Error

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

```
*Unsaved Document 1 ×
1
2 GENERAL INFORMATION
3 Company : Maidenhead Bridge
4 Location : H0kvm
5 CSC ID : ns-cgc00002-b
6 CSC date: Thu 14 Oct 18:59:34 BST 2021
7 Soft version : 1.0
8
9 INTERFACES INFORMATION
10 External: Tunnel IP: 192.168.1.60 | Bypass Proxy Egress IP: 192.168.1.61 | CSC IP(eth0): 192.168.1.63/24 | Network Gateway: 192.168.1.240 is Alive
11 Internal: CSC GW IP: 172.19.0.60 | CSC IP(eth1): 172.19.0.64/24 | Network Gateway: 172.19.0.133 is Alive
12
13 TRAFFIC REDIRECTION Options
14 To Netskope: VIP Proxy: 172.19.0.61:80 | Route all traffic via CSC GW IP | Netskope Global Proxy IP: 163.116.128.80:80 via CSC GW IP
15 Direct to Internet: Bypass Proxy: 172.19.0.62:3128 | Netskope Global Proxy IP: 163.116.128.80:3128 via CSC GW IP
16
17 DNS INFORMATION
18 DNS Server (1) IP: 172.19.0.100 is Alive
19 DNS Server (2) IP: 1.1.1.1 is Alive
20
21 NETSKOPE INFORMATION
22 GRE tunnels egress Public IP: 82.68.6.74
23
24 Primary Tunnel:
25     Node : GB,London,LON1
26     Node Public IP: 163.116.162.36
27     Node Probe: 10.162.6.209
28 Secondary Tunnel:
29     Node : GB,Manchester,MAN1
30     Node Public IP: 163.116.165.36
31     Node Probe: 10.165.6.209
32
33 TUNNEL STATUS
34 Primary Tunnel (reachability):
35     Node Keepalive is: Alive
36     GRE Tunnel IP is: Standby - This CSC (ns-cgc00002-b) is Cluster Standby
37 Secondary Tunnel (reachability):
38     Node Keepalive is: Alive
39     GRE Tunnel IP is: Standby - This CSC (ns-cgc00002-b) is Cluster Standby
40 returnToPrimaryTunnel: true
41
42 Tunnel Status: No active tunnel since: Tue 5 Oct 19:27:15 UTC 2021
43
44 HTTP://WWW.NOTSKOPE.COM PAGE STATUS
45 No test performed - This CSC (ns-cgc00002-b) is Cluster Standby
46
47 PROXY BYPASS - EGRESS INTERFACE STATUS
48 No test performed - This CSC (ns-cgc00002-b) is Cluster Standby
49
50 ROUTED BYPASS
51 Using Routed Bypass URL: https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json
52 Routed Bypass URL https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json is reachable
53 Routed Bypass Rules configured via URL: 8
54
55 AWS SSM AGENT
56 AWS SSM Agent is active (running) since Tue 2021-10-05 20:27:11 BST; 1 weeks 1 days ago
57 Registration values: {"ManagedInstanceId":"mi-0100c70a3ad29e8b5","Region":"eu-west-2"}
58
59 SYSLOG INFORMATION
60 SYSLOG Server (1) IP: 172.19.0.199 is Alive
61 SYSLOG Server (2) IP is not configured
62 SYSLOG TCP Port: 514
63
64 HIGH AVAILABILITY Information
65 This CSC (ns-cgc00002-b) is Cluster STANDBY
```

9.1.3 List of Documents available for "Run Command"

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

9.2 Rundeck

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

Installation Steps:

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

The screenshot shows the Rundeck interface for editing node definitions. The left sidebar has icons for DASHBOARD, JOBS, NODES, COMMANDS, ACTIVITY, and WEBHOOKS. A red box highlights the 'NODES' icon. The main area shows a dropdown menu for 'NS-CSC-MGMT' and a 'Project' tab. Below is a section titled 'Edit Nodes File' with a red box around it, showing the file path '/home/rundeck06/rundeck/NS-CSC-MGMT-NODES.json'. The file content is displayed in a code editor with line numbers. A red box highlights the first node definition, which is a JSON object with fields like 'hostname', 'nodename', 'description', 'tags', 'username', 'osVersion', and 'osName'. Another red box highlights the second node definition, which is similar but under a different key. The code editor also shows 'Source' (File) and 'Format' (json) settings. At the bottom are 'Cancel' and 'Save' buttons. A red box highlights the 'PROJECT SETTINGS' icon in the bottom-left corner of the sidebar.

```
1 ns-cgc00002-a : {  
2   "hostname": "172.19.0.63",  
3   "nodename": "ns-cgc00002-a",  
4   "description": "CSC GRE Cluster A",  
5   "tags": "csc-gre-cluster,netskope,active",  
6   "username": "csccli",  
7   "osVersion": "1.8",  
8   "osName": "csc-gre-cluster"  
9 },  
10 ns-cgc00002-b : {  
11   "hostname": "172.19.0.64",  
12   "nodename": "ns-cgc00002-b",  
13   "description": "CSC GRE Cluster B",  
14   "tags": "csc-gre-cluster,netskope,active",  
15   "username": "csccli",  
16   "osVersion": "1.0",  
17   "osName": "csc-gre-cluster"  
18 },  
19 ns-cgc00001-a : {  
20   "hostname": "172.19.0.23",  
21   "nodename": "ns-cgc00001-a",  
22   "description": "CSC GRE Cluster A",  
23   "tags": "csc-gre-cluster,netskope,inactive",  
24   "username": "csccli",  
25   "osVersion": "1.0",  
26   "osName": "csc-gre-cluster"  
27 },  
28 ns-cgc00001-b : {  
29   "hostname": "172.19.0.24",  
30   "nodename": "ns-cgc00001-b",  
31   "description": "CSC GRE Cluster B",  
32   "tags": "csc-gre-cluster,netskope,inactive",  
33   "username": "csccli",  
34   "osVersion": "1.0",  
35   "osName": "csc-gre-cluster"  
36 },  
37 }  
38 }  
39 }
```

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

9.2.1 Jobs

The following screen shows the list of Jobs available.

NS-CSC-MGMT

17 All Jobs

Expand All Collapse All

- ▶ Check CSC Status - Netskope This test checks L7 Keepalives on CSCs using Netskope Cloud OK in 11m
- ▶ Refresh Proxy Bypass URL
- ▶ Refresh Proxy Bypass URL - CSCs with **tags:active** This job executes Refresh Proxy Bypass List command on all CSCs with tags:active
- ▶ Refresh Routed Bypass URL This job updates the Routed Bypass Configuration on the CSC using the Routed Bypass URL.
- ▶ Refresh Routed Bypass URL - CSCs with **tags:active** This job updates the Routed Bypass Configuration on the CSCs with tags:active using the Routed Bypass URL
- ▶ Reload Config Json File This job reloads the values of the config.json file onto the CSC.
- ▶ Reload High Availability Json File This job is valid only for CSCs on AWS, Azure and Gcloud.
- ▶ Reload Routed Bypass Json File
- ▶ Show Configuration and Status This job provides all configuration and statuses information of the CSC.
- ▶ Show Configuration and Status - CSC with **tags:active** This job executes Show Configuration and Status command on all CSCs with tag:active
- ▶ Show Logs Current Month
- ▶ Show Logs Last 6 Months
- ▶ Speed Test This job executes Speed Test from the CSC to speedtest.net
- ▶ Switch Tunnels This Job Switches tunnels Primary / Secondary
- ▶ Test Email Use this job to check that you are receiving alerts via email.
- ▶ Traceroute and Latency Test Use this job to check the quality of the path to the Cloud - hop by hop
- ▶ Update Nodes Database

9.2.2 Running job "Show Configuration and Status"

NS-CSC-MGMT

✓ Show Configuration and Status - CSC with **tags:active** []

This job executes Show Configuration and Status command on all CSCs with tag:active

Succeeded 0.00:09 at 7:38 pm

Log Output »

100% 2/2 COMPLETE		0 FAILED	0 INCOMPLETE	0 NOT STARTED	Start time	Duration
Node	ns-cgc00002-a	All Steps OK	OK		7:38:08 pm	0:00:05
Command						
18:38:11	GENERAL INFORMATION					
18:38:11	Company : Maidenhead Bridge					
18:38:11	Location : HQKvm					
18:38:11	CSC ID : ns-cgc00002-a					
18:38:11	CSC date: Thu 14 Oct 19:38:10 BST 2021					
18:38:11	Soft version : 1.0					
18:38:11	INTERFACES INFORMATION					
18:38:11	External: Tunnel IP: 192.168.1.60 Bypass Proxy Egress IP: 192.168.1.61 CSC IP(eth0): 192.168.1.62/24 Network Gateway: 192.168.1.240 is Alive					
18:38:11	Internal: CSC GW IP: 172.19.0.60 CSC IP(eth1): 172.19.0.63/24 Network Gateway: 172.19.0.133 is Alive					
18:38:11	TRAFFIC REDIRECTION Options					
18:38:11	To Netskope: VIP Proxy: 172.19.0.61:80 Route all traffic via CSC GW IP Netskope Global Proxy IP: 163.116.128.80:80 via CSC GW IP					
18:38:11	Direct to Internet: Bypass Proxy: 172.19.0.62:3128 Netskope Global Proxy IP: 163.116.128.80:3128 via CSC GW IP					
18:38:11	DNS INFORMATION					
18:38:11	DNS Server (1) IP: 172.19.0.100 is Alive					
18:38:11	DNS Server (2) IP: 1.1.1.1 is Alive					
18:38:11	NETSKOPE INFORMATION					
18:38:11	GRE tunnels egress Public IP: 82.68.6.74					
18:38:11	Primary Tunnel:					
18:38:11	Node : GB,London,LON1					
18:38:11	Node Public IP: 163.116.162.36					
18:38:11	Node Probe: 10.162.6.209					
18:38:11	Secondary Tunnel:					

10 DevOps operations

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

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

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

The JSON files are available are:

1. **config.json**: Allows administrators to modify specific values on the CSC like GRE Primary, Secondary nodes, DNS, Syslog, Routed Bypass URL, Proxy Bypass URL, Etc.
2. **routedBypassRulesFile.json**: Allows administrators to manually configure Routed Bypass Rules if not using the Routed Bypass URL method.
3. **privateAccessPeersConfig.json**: Use this Json file to configure "networks" and "privateApps" on your Private Cloud.
4. **highAvailability.json**: Allows administrators to configure the CSC on HA pair.

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

10.1 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-ns-vm",
  "version": "1.1",
  "dns": {
    "primaryDnsIP": "172.19.0.100",
    "secondaryDnsIP": "172.19.0.101"
  },
  "bypassProxyPublicIPgrePublicIProutedBypassPublicIP
```

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.

10.2 routedBypassRulesFile.json

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

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

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

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

- 
2. Create a AWS bucket 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.

10.3 privateAccessPeersConfig.json

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

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

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

```
{  
  "peers": [  
    {  
      "nodeName": "ns-cgc00001",  
      "description": "Node on VMware Server 1",  
      "location": "HQ",  
      "publicKey": "yAnz5TF+IXXJte14tji3zIMNq+hd2rYUlgJBgB3fBmk=",  
      "publicIpAndUdpPort": "200.1.1.1:51821",  
      "privateCirdip": "192.168.7.1/24",  
      "persistentKeepAlive": "no",  
      "networks": ["10.1.1.0/24", "10.1.2.0/24"],  
      "privateApps": [  
        {  
          "description": "Allow all traffic to this site",  
          "ipProtocol": "all",  
          "sourceCirdip": ["0.0.0.0/0"],  
          "destinationCirdip": ["10.1.1.0/24", "10.1.2.0/24"],  
          "destinationSinglePorts": [ "" ],  
          "destinationPortRange": { "fromPort": "", "toPort": "" }  
        }  
      ]  
    },  
    {  
      "nodeName": "ns-cgc00002",  
      "description": "Node on VMware Server 2",  
      "location": "Datacentre 2",  
      "publicKey": "xtIBASrboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",  
      "publicIpAndUdpPort": "200.1.1.2:51821",  
      "privateCirdip": "192.168.7.2/24",  
      "persistentKeepAlive": "no",  
      "networks": ["10.2.1.0/24", "10.2.2.0/24"],  
      "privateApps": [  
        {  
          "description": "Allow all traffic to this site",  
          "ipProtocol": "all",  
          "sourceCirdip": ["0.0.0.0/0"],  
          "destinationCirdip": ["10.2.1.0/24", "10.2.2.0/24"],  
          "destinationSinglePorts": [ "" ],  
          "destinationPortRange": { "fromPort": "", "toPort": "" }  
        }  
      ]  
    },  
    {  
      "nodeName": "ns-cgc00003",  
      "description": "Node on VMware Server 3",  
      "location": "Branch",  
      "publicKey": "TrMvSoP4jYQlY6RlzBgbsQqY3vxI2Pi+y71lOWWWXX0=",  
      "publicIpAndUdpPort": "200.1.1.3:51821",  
      "privateCirdip": "192.168.7.3/24",  
      "persistentKeepAlive": "no",  
      "networks": ["10.3.1.0/24", "10.3.2.0/24"],  
      "privateApps": [  
        {  
          "description": "Allow all traffic to this site",  
          "ipProtocol": "all",  
          "sourceCirdip": ["0.0.0.0/0"],  
          "destinationCirdip": ["10.3.1.0/24", "10.3.2.0/24"],  
          "destinationSinglePorts": [ "" ],  
          "destinationPortRange": { "fromPort": "", "toPort": "" }  
        }  
      ]  
    }  
  ]  
}
```

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

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

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

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

10.4 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": false,
    "halamRole
```

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,
    "halamRole
```

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.

11 Appendixes

11.1 Appendix A: Routed Bypass JSON file if you don't have Cloud Firewall License.

If you decide to configure the default route to the internet via the CSC and don't have a Cloud Firewall license, you need to send only HTTP and HTTPS via the GRE tunnel and the rest of the traffic via Routed Bypass.

The following JSON file does the work to redirect only Web traffic via the GRE tunnel, and the rest goes directly via the Bypass Interface.

```
{
  "routedBypassRules": [
    {
      "description": "Bypass ICMP all",
      "ipProtocol": "icmp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "0.0.0.0/0",
      "fromPort": "",
      "toPort": ""
    },
    {
      "description": "Bypass TCP Ports I",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "0.0.0.0/0",
      "fromPort": "1",
      "toPort": "79"
    },
    {
      "description": "Bypass TCP Ports II",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "0.0.0.0/0",
      "fromPort": "81",
      "toPort": "442"
    },
    {
      "description": "Bypass TCP Ports III",
      "ipProtocol": "tcp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "0.0.0.0/0",
      "fromPort": "444",
      "toPort": "65535"
    },
    {
      "description": "Bypass UDP Ports all",
      "ipProtocol": "udp",
      "sourceCirdIp": "0.0.0.0/0",
      "destinationCirdIp": "0.0.0.0/0",
      "fromPort": "1",
      "toPort": "65535"
    }
  ]
}
```



11.2 Appendix B: Release Notes

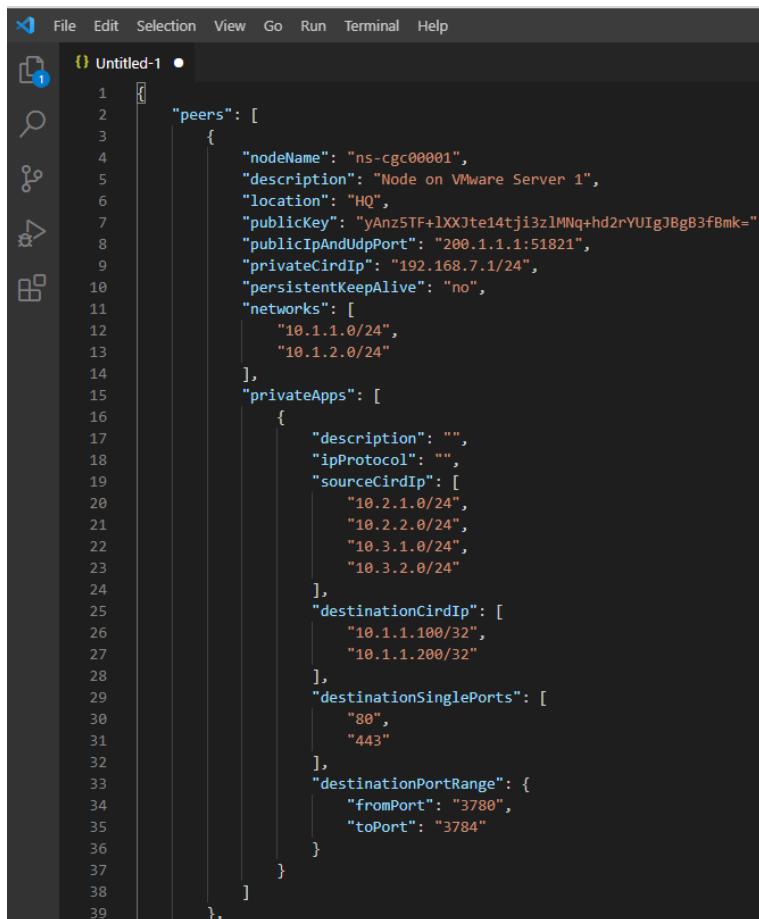
11.2.1 Version 1.0

This is the initial version of the Cloud Security Connector for Netskope on AWS.

11.3 Appendix C: JSON formatters (Visual Code, Notepad ++)

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

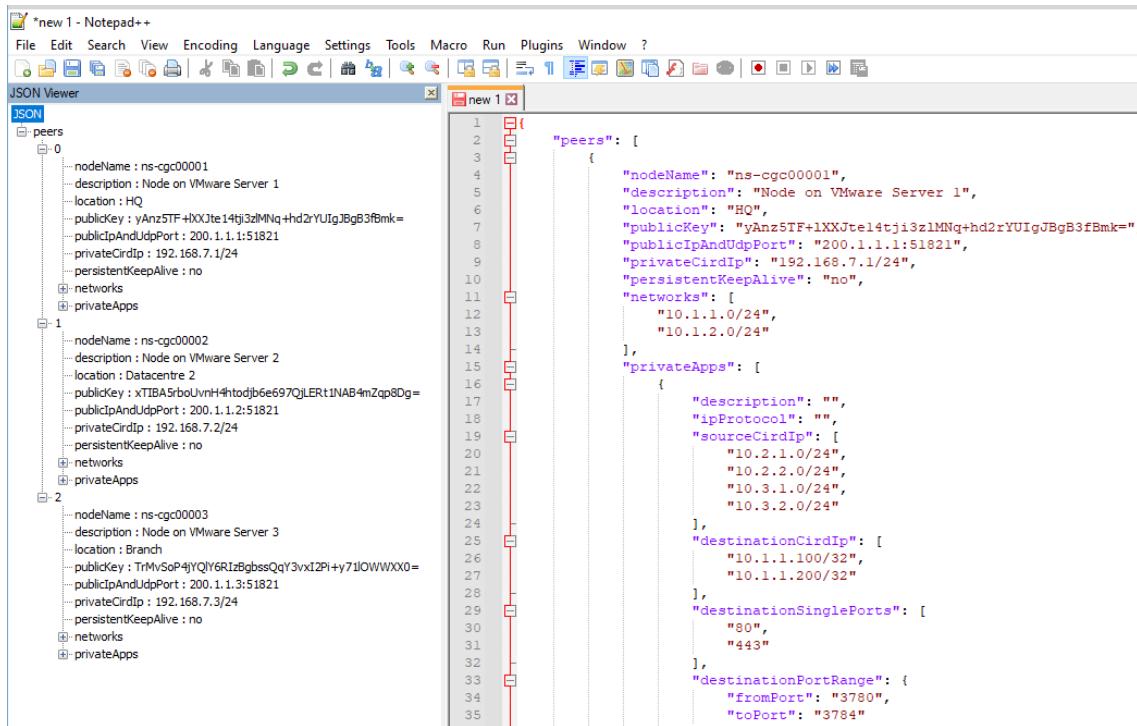
11.3.1 Visual Code



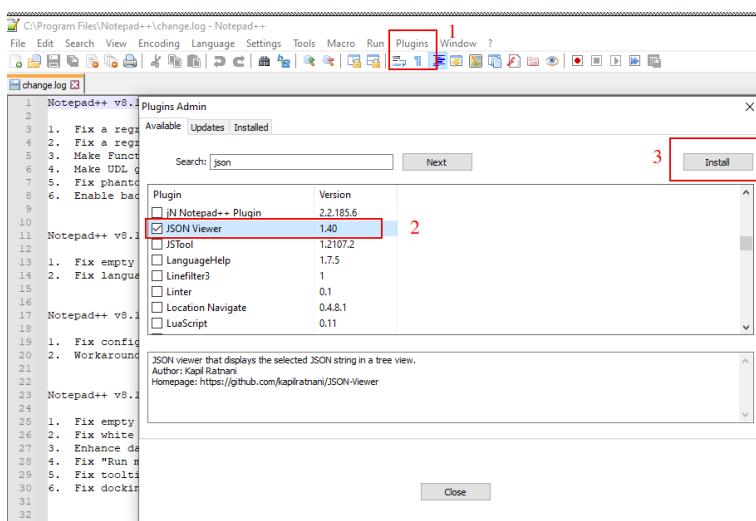
```
1  {
2   "peers": [
3     {
4       "nodeName": "ns-cgc00001",
5       "description": "Node on VMware Server 1",
6       "location": "HQ",
7       "publicKey": "yAnz5TF+1XXJte14tj13z1MNq+hd2rYUIgJBgB3fBmk=",
8       "publicIpAndUdpPort": "200.1.1.1:51821",
9       "privateCirdIp": "192.168.7.1/24",
10      "persistentKeepAlive": "no",
11      "networks": [
12        "10.1.1.0/24",
13        "10.1.2.0/24"
14      ],
15      "privateApps": [
16        {
17          "description": "",
18          "ipProtocol": "",
19          "sourceCirdIp": [
20            "10.2.1.0/24",
21            "10.2.2.0/24",
22            "10.3.1.0/24",
23            "10.3.2.0/24"
24          ],
25          "destinationCirdIp": [
26            "10.1.1.100/32",
27            "10.1.1.200/32"
28          ],
29          "destinationSinglePorts": [
30            "80",
31            "443"
32          ],
33          "destinationPortRange": {
34            "fromPort": "3780",
35            "toPort": "3784"
36          }
37        }
38      ],
39    }
40  },
```

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

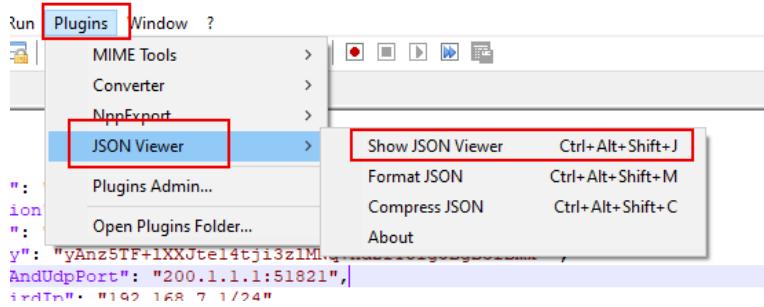
11.3.2 Notepad ++



1. Download: <https://notepad-plus-plus.org/downloads/>
2. Install JSON Viewer Plug in.



3. Create your JSON file.
4. To Check your JSON file go to: Plugins -> JSON Viewer -> Show JSON Viewer.



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

11.4 Appendix D: Securing an AWS Bucket by source IP.

1. On your AWS console create a bucket with default values for permissions: "Block all Public Access = on"
2. On Bucket Policy, add your Public IPs in "aws:SourceIp":[]

The screenshot shows the AWS S3 console. On the left, the navigation pane includes 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', and 'AWS Organizations settings'. A 'Feature spotlight' section is also present.

Block public access (Bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access is to your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you can turn on Block all public access, but before applying any of these settings, consider how they will affect your specific storage use cases. Learn more [?]

Block all public access

On

- Block public access to buckets and objects granted through new access control lists (ACLs)
- On
- Block public access to buckets and objects granted through any access control lists (ACLs)
- On
- Block public access to buckets and objects granted through new public bucket or access point policies
- On
- Block public and cross-account access to buckets and objects through any public bucket or access point policies
- On

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more [?]

Block All

2 - Click 'Edit' button

3 - Add this policy.

```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::mhb-netskope-private/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "200.1.1.1/32",
            "200.1.1.2/32",
            "200.2.0.0/24"
          ]
        }
      }
    }
  ]
}
```

3. Done!