



Maidenhead Bridge



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

(For Azure Cloud)

Version 1.2

December 2022

Table of Contents

1 Introduction to Cloud Security Connectors for Netskope.....	6
2 Key benefits of the Cloud Security Connector Mux for Azure.....	6
3 Network Diagrams.....	8
3.1 CSC Mux for Azure – Single deployment.....	8
3.2 CSC Mux for Azure – High Availability Deployment.....	9
3.3 Steering: Routing and Proxying all together.....	10
3.4 Private Cloud Private Access (PriCPA).....	11
4 Creating the CSC Mux 4 or 8 for Azure.....	12
4.1 Prerequisites.....	12
4.2 Launching the CSC Mux for Azure Marketplace.....	12
5 Accessing for first time to your CSC.....	20
5.1 Creating the IPsec tunnels on Netskope console.....	21
5.2 Checking status on the CSC console.....	22
6 Resources creates by the ARM template.....	23
7 The Cloud Security Connector Admin Console:.....	25
7.1 Monitoring Tasks.....	28
7.1.1 Show Configuration and Status.....	28
7.1.1.1 GENERAL INFORMATION.....	29
7.1.1.2 INTERFACES INFORMATION.....	29
7.1.1.3 TRAFFIC REDIRECTION Options.....	29
7.1.1.4 PUBLIC IP Address INFORMATION.....	30
7.1.1.5 DNS INFORMATION.....	30
7.1.1.6 NETSKOPE INFORMATION.....	30
7.1.1.7 LOAD BALANCING INFORMATION.....	31
7.1.1.8 IPSEC INFORMATION.....	31
7.1.1.9 HTTP://WWW.NOTSKOPE.COM PAGE STATUS.....	31
7.1.1.10 PROXY BYPASS - EGRESS INTERFACE STATUS.....	32
7.1.1.11 ROUTED BYPASS.....	32
7.1.1.12 AWS SSM AGENT.....	32
7.1.1.13 SYSLOG/SIEM Servers Information.....	32
7.1.1.14 HIGH AVAILABILITY Information.....	32
7.1.2 Show Interfaces Traffic.....	33
7.1.3 Traceroute and Latency Test.....	33
7.1.4 SPEED TEST.....	34
7.2 CSC Admin Tasks.....	35
7.2.1 AWS SSM Agent (Register or De-Register).....	35
7.2.1.1 Create a "Hybrid Activation" from AWS console.....	35
7.2.1.2 Register the CSC.....	36
7.2.1.3 View the Registered CSC on AWS Systems Manager.....	36
7.2.2 Manage Administrators.....	37
7.2.2.1 "cscadmin" settings.....	37
7.2.2.2 "csccli" settings.....	37

7.2.2.3 Managing the SSH Key of a User.....	38
7.2.3 Change Timezone.....	38
7.3 Proxy Bypass.....	39
7.3.1 Proxy Bypass - Traffic Flow.....	39
7.3.2 View Current Proxy Bypass List.....	39
7.3.3 Configure Proxy Bypass List.....	39
7.3.3.1 Auto - Proxy Bypass PAC URL.....	40
7.3.3.2 Manual Proxy Bypass Configuration.....	41
7.4 Routed Bypass.....	43
7.4.1 Routed Bypass - Traffic Flow.....	43
7.4.2 View Current Routed Bypass List.....	43
7.4.2.1 Compact.....	43
7.4.2.2 Json.....	44
7.4.3 Configure Routed Bypass List.....	45
7.4.3.1 Routed Bypass URL.....	45
7.4.3.2 Manual (Paste Routed Bypass JSON file).....	46
7.5 Log Information.....	47
7.5.1 View Current Month.....	47
7.5.2 View Last 6 Months.....	47
7.6 Configuration Wizards.....	48
7.6.1 Change Nodes, DNS servers, Syslog and more.....	48
7.6.1.1 Running the Configuration Wizard.....	49
7.6.2 Switch Tunnels - Primary / Secondary.....	52
7.6.3 Update Netskope Nodes Databases.....	52
7.6.4 High Availability configuration.....	53
8 Steering traffic to NewEdge with the CSC Mux for Azure.....	61
8.1 CSC on HA Pair.....	61
8.1.1 Network Diagram.....	61
8.1.2 Prerequisites.....	62
8.1.3 Routing traffic via the CSC HA pair.....	62
8.1.3.1 Traffic to Netskope.....	62
8.1.3.2 "Routed Bypass" traffic.....	62
8.1.4 Proxy traffic via the CSC HA Pair.....	62
8.1.4.1 Using PAC files.....	62
8.1.4.1.1 PAC file for Load Balancing.....	62
8.1.4.1.2 PAC file using Netskope's Global Proxy.....	64
8.1.4.1.3 PAC file using CSC's VIP and Bypass Proxy Address (Pri/Sec).....	65
8.1.4.2 Using Explicit Proxy on devices that cannot support PAC files.....	66
8.2 CSC Single.....	67
8.2.1 Network Diagram.....	67
8.2.2 Prerequisites.....	67
8.2.3 Routing traffic via the CSC Single.....	68
8.2.3.1 Traffic to Netskope.....	68
8.2.3.2 "Routed Bypass" traffic.....	68

8.2.4 Proxy traffic via the CSC Single.....	68
8.2.4.1 Using PAC files.....	68
8.2.4.1.1 PAC file using Netskope's Global Proxy.....	69
8.2.4.1.2 PAC file using CSC's VIP and Bypass Proxy Address (Pri/Sec).....	70
8.2.4.2 Using Explicit Proxy on devices that cannot support PAC files.....	71
8.3 Testing traffic to Netskope.....	72
8.3.1 www.notskope.com.....	72
8.3.2 https://ip.maidenheadbridge.com.....	73
8.3.3 SpeedTest.....	74
9 Private Cloud Private Access.....	75
9.1 What is Private Cloud Private Access (PriCPA)?.....	75
9.2 PriCPA Network Diagrams.....	75
9.2.1 High Level Network Diagram.....	75
9.2.2 Low Level Network Diagram – PriCPA only.....	76
9.3 Configuring PriCPA.....	77
9.3.1 Create the Local configuration (first node of the cluster).....	78
9.3.2 Create the Local configuration (second node of HA Pair).....	80
9.3.3 Create the Private Access Peers JSON file.....	81
9.3.3.1 Full mesh Private Access Peers JSON file.....	81
9.3.3.2 Understanding "privateApps" configuration and values.....	85
9.3.3.3 Example of "privateApps" for a Windows Domain controller.....	88
9.3.3.4 Example of "privateApps" for Internal Web Server.....	88
9.3.4 Load the "Private Access Peers JSON file" to the CSCs.....	89
9.3.4.1 Using "Private Access Peers URL".....	89
9.3.4.2 <i>Manual: Copy and Paste</i> "Private Access Peers Json file".....	95
9.4 Show Configurations and Status Private Access.....	96
9.4.1 Via SSH console.....	96
9.4.1.1 Show Peer/s Status.....	96
9.4.1.2 Show Peers Json file (active).....	97
9.4.1.3 Show Local Configuration.....	98
9.4.1.4 Show Firewall Local Rules.....	98
9.4.2 via AWS Systems Manager or Rundeck.....	99
9.4.2.1 AWS Systems Manager.....	99
9.4.2.2 Rundeck.....	99
9.5 Configure CSC Remote Management via Private Access.....	100
10 Remote Management using AWS and Rundeck.....	101
10.1 AWS Systems Manager.....	101
10.1.1 Create Documents.....	101
10.1.2 Run Commands.....	103
10.1.3 List of Documents available for "Run Command".....	106
10.2 Rundeck.....	107
10.2.1 Jobs.....	108
10.2.2 Running job "Show Configuration and Status".....	108
11 DevOps operations.....	109

11.1 config.json file.....	110
11.2 routedBypassRulesFile.json.....	111
11.3 privateAccessPeersConfig.json.....	113
11.4 highAvailability.json file.....	115
12 Appendixes.....	117
12.1 Appendix A: Routed Bypass JSON file if you don't have Cloud Firewall License.....	117
12.2 Appendix B: Release Notes.....	118
12.2.1 Version 1.0.....	118
12.2.2 Version 1.1.....	118
12.2.3 Version 1.2.....	118
12.3 Appendix C: JSON formatters (Visual Code, Notepad ++).	119
12.3.1 Visual Code.....	119
12.3.2 Notepad ++.....	120
12.4 Appendix D: Securing an AWS Bucket by source IP.....	122

1 Introduction to Cloud Security Connectors for Netskope.

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

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

The CSC Mux for Azure lets you connect securely to Netskope NewEdge up to 4 Gbps without hassle.

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

After launching the CSC Mux from the Azure Marketplace using the ARM template provided, the CSC Mux will automatically select the best Netskope Edge nodes and do 4 (CSC Mux 4) or 8 (CSC Mux 8) IPsec tunnels to Primary and Secondary Netskope Nodes.

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

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

In addition to this, the CSC Mux provides high availability changing the default route to 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 (PriCPA) functionality that allows you to create a full mesh among the CSCs communicating your private traffic on a Zero Trust model.

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

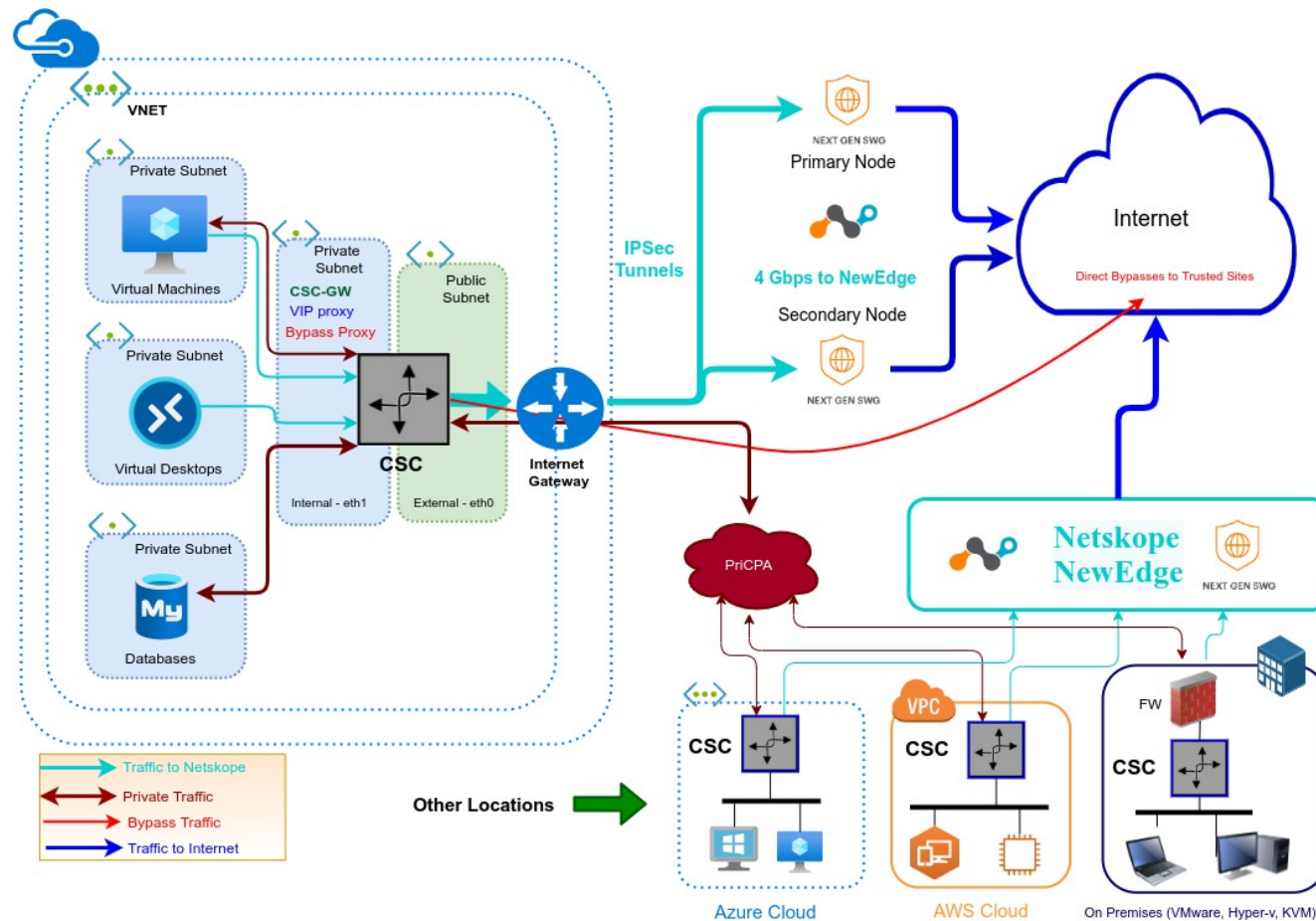
2 Key benefits of the Cloud Security Connector Mux for Azure

- No Networking knowledge is required.
- Automated deployment from Marketplace, ARM template or your tool of choice. (i.e. Terraform)
- Enables any Location to be connected to Netskope NewEdge up to 4 Gbps.

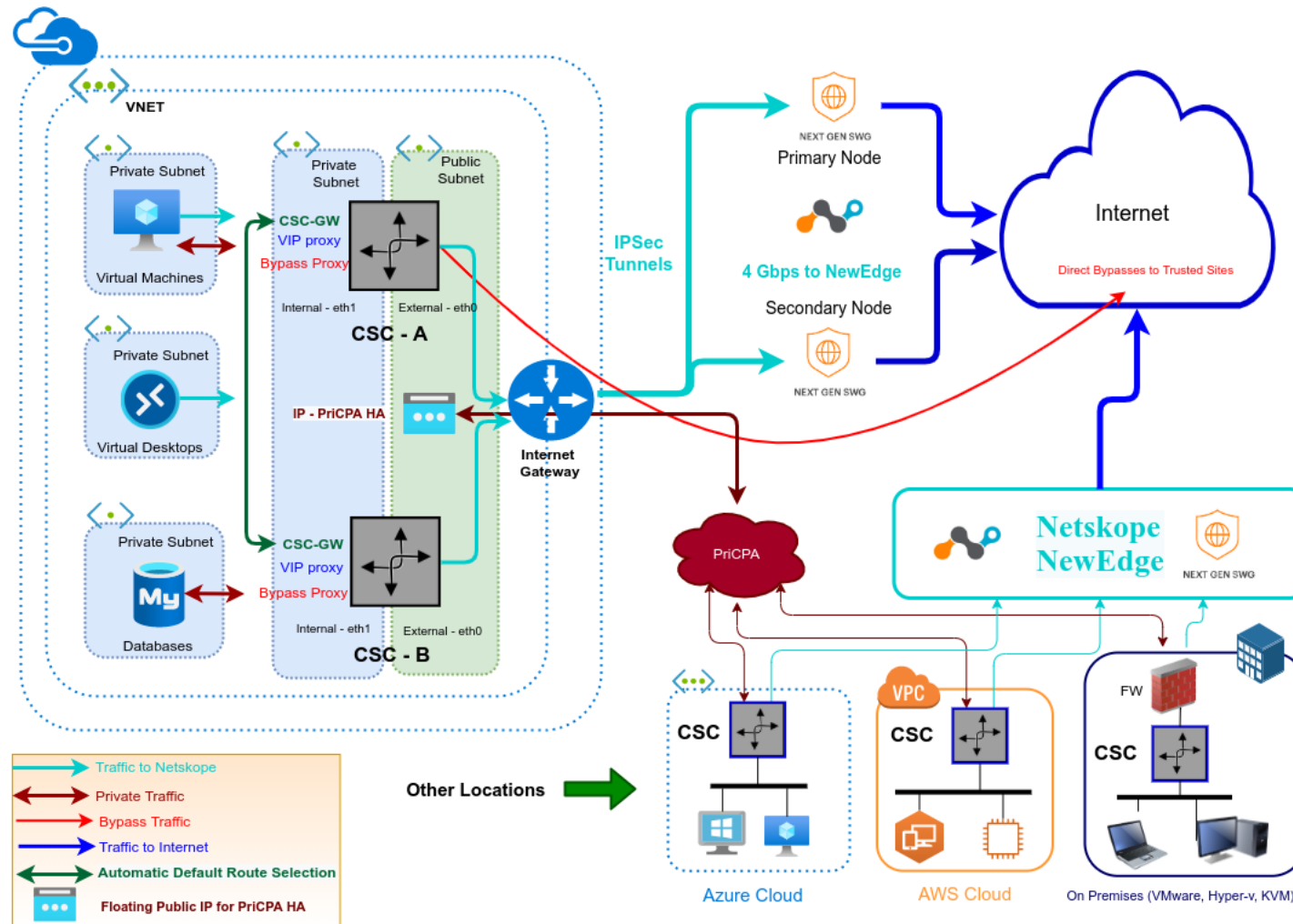
- With Private Cloud Private Access you can connect all sites securely on a Zero Trust model. The CSC secures your Private Traffic between your physical and cloud locations.
- The CSC comes with the optimal values to work with Netskope NewEdge.
- Full tunnel redundancy.
- High Availability with automatic 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 Netskope 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.)

3 Network Diagrams

3.1 CSC Mux for Azure – Single deployment



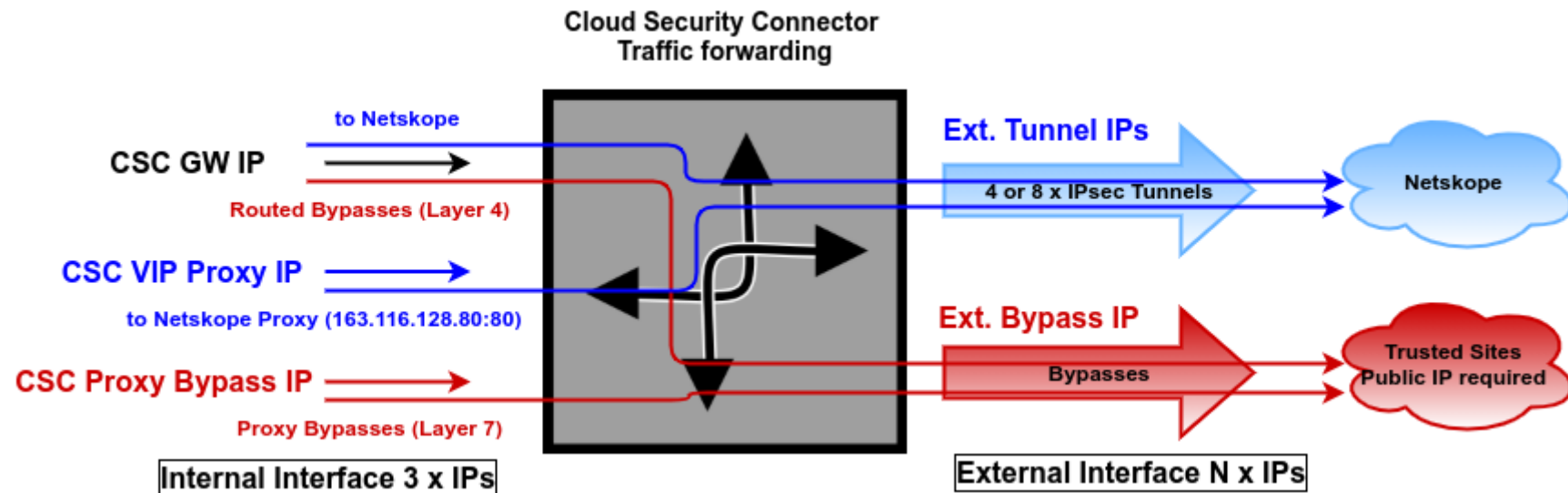
3.2 CSC Mux for Azure – High Availability Deployment



3.3 Steering: Routing and Proxying all together.

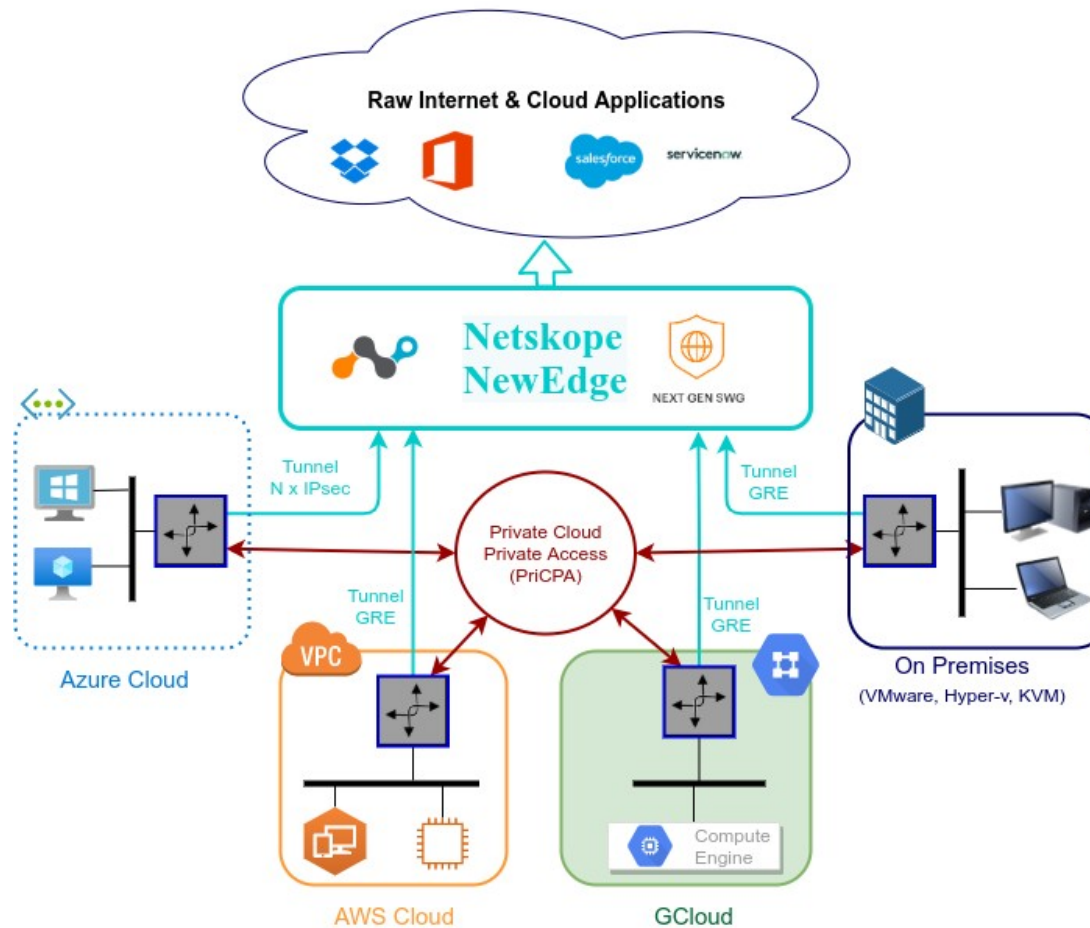
The most significant benefit of the Cloud Security Connector for Netskope is that it covers all possible scenarios (routed traffic, PAC files, explicit proxy, Etc.) for any device on your organization: Laptops, Desktops, Servers, IoT devices, Virtual Desktops, Etc.

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



3.4 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 CSC Mux 4 or 8 for Azure

4.1 Prerequisites

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

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


4.2 Launching the CSC Mux for Azure Marketplace

Go to Azure Marketplace, search for "Maidenhead Bridge", and select **"Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA"**.

The screenshot shows the Azure Marketplace interface. At the top, there's a search bar with 'Search Marketplace' and a magnifying glass icon. Below the search bar, the breadcrumb 'Products > Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA' is visible. The main content area features the product logo (a blue circle with a white figure holding a torch) and the title 'Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA' by 'Maidenhead Bridge'. Below the title are tabs for 'Overview', 'Plans', and 'Ratings + reviews'. The 'Overview' tab is selected. The description states: 'The easiest way to connect to Netskope NewEdge and communicate private Cloud Workloads. The Cloud Security Connector (CSC) is a device that enables easy deployments of the Netskope SSE solution in any customer environment. There are CSC models for Virtual Platforms (VMware, Hyper-V) and Public Clouds (Azure, AWS, etc.).' A list of bullet points follows, detailing the product's features and capabilities. On the left side, there's a sidebar with 'Categories' (Compute, Networking, Security), 'Support' (Support, Help), and 'Legal' (Under Microsoft Standard, Contract, Privacy Policy). A red box highlights the 'Get It Now' button.

Microsoft | Azure Marketplace More ▾ Search Marketplace

Products > Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

 Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA
Maidenhead Bridge

Overview Plans Ratings + reviews

The easiest way to connect to Netskope NewEdge and communicate private Cloud Workloads.

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

- The Cloud Security Connector Multiplex (CSC Mux) for Azure is a virtual machine connecting internal Azure resources to Netskope NewEdge.
- The CSC Mux for Azure lets you connect securely to Netskope NewEdge up to 4 Gbps without hassle.
- The primary purpose of the CSC family is simplicity. The CSC for Azure comes with all configurations required.
- After launching the CSC Mux from the Azure Marketplace using the ARM template provided, the CSC Mux will automatically select the best Netskope Edge nodes and do 4 (CSC Mux 4) or 8 (CSC Mux 8) IPsec tunnels to Primary and Secondary Netskope Nodes.
- The CSC Mux contains the perfect configuration for IPsec tunnels, firewall rules, and necessary routing tables.
- All Netskope functionalities are available, providing complete visibility of all Internet traffic.
- In addition to this, the CSC Mux provides high availability changing the default route to 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 (PriCPA) functionality that allows you to create a full mesh among the CSCs communicating your private traffic on a Zero Trust model.
- Simple to install with complete management using DevOps change management tools like Amazon Systems Manager, Rundeck, Ansible, etc; and SSH.


Categories
Compute
Networking
Security

Support
Support
Help

Legal
Under Microsoft Standard
Contract
Privacy Policy

Get It Now

→ Click "Get it Now"



Create this app in Azure

Cloud Security Connector Mux (model 4 and 8) for Netskope ...
By Maidenhead Bridge

Software plan

CSC Mux 4 for Netskope - Availability Zones Deployment

Pricing: This solution template deploys software components and Azure infrastructure components. The price is the cost of those components.


Details: The easiest way to connect to Netskope NewEdge and communicate private Cloud Workloads.

This app requires some basic profile information. You have provided the information already so you're good to go! [Edit](#)

By clicking "Continue", I grant Microsoft permission to share my supplied contact information with the provider so that they can contact me regarding this product and related products. The shared information will be handled in accordance with the [Microsoft Standard Contract](#) and provider's [privacy statement](#).

Continue

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




Microsoft Azure

[Home](#) >

Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

Maidenhead Bridge



Plan

CSC Mux 4 for Netskope - Availability... [Create](#)

Check Plan

Overview | [Plans](#) | [Usage Information + Support](#) | [Reviews](#)

Offered under [Microsoft Standard Contract](#).

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

- The Cloud Security Connector Multiplex (CSC Mux) for Azure is a virtual machine connecting internal Azure resources to Netskope NewEdge.
- The CSC Mux for Azure lets you connect securely to Netskope NewEdge up to 4 Gbps without hassle.
- The primary purpose of the CSC family is simplicity. The CSC for Azure comes with all configurations required.
- After launching the CSC Mux from the Azure Marketplace using the ARM template provided, the CSC Mux will automatically select the best Netskope Edge nodes and do 4 (CSC Mux 4) or 8 (CSC Mux 8) IPsec tunnels to Primary and Secondary Netskope Nodes.
- The CSC Mux contains the perfect configuration for IPsec tunnels, firewall rules, and necessary routing tables.
- All Netskope functionalities are available, providing complete visibility of all Internet traffic.
- In addition to this, the CSC Mux provides high availability changing the default route to 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 (PriCPA) functionality that allows you to create a full mesh among the CSCs communicating your private traffic on a Zero Trust model.
- Simple to install with complete management using DevOps change management tools like Amazon Systems Manager, Rundeck, Ansible, etc; and SSH.

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

Microsoft Azure

Search resources, services, and docs (G+)

Home

>

Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA (preview)

>

Create Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

1 Basics

2 Virtual Machine Settings

3 Networking

4 configUserData.json File

5 Review + create

Project details

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

Subscription *

Pay-As-You-Go

Resource group *

CSC-East-US

Create new

Instance details

Location *

East US

⚠

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

Select Single or HA configuration *

☐ Deploy Single (1x) CSC
 ☒ Deploy High Availability (2x) CSCs

ℹ

Choose the Availability Zones for each Cloud Security Connector.

First CSC Availability Zone *

Zone 1

Second CSC Availability Zone *

Zone 2

CSC_Name *

csc-mux-4-az

Admin Username

cscadmin

Authentication type *

☒ Password
 ☐ SSH Public Key

Password *

.....

✓

Confirm password *

.....

✓

< Previous

Next

Fill the values on "Basics"

1. Resource Group.
2. Location.

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

→ Click "Next".

Microsoft Azure

Search resources, services, and docs (G+)

Home > Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA (preview) >

Create Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

✓ Basics 4 Virtual Machine Settings 3 Networking 4 configUserData.json File 3 Review + create

Virtual machine size * ⓘ

1x Standard D4s v4
4 vcpus, 16 GB memory
[Change size](#)

CSC VM Disk storage account type * ⓘ

Standard_LRS

< Previous Next

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

If you want to change the values, take into account the minimum requirements:

- CSC Mux 4 requires 2 x Cores, 4GB RAM and Accelerated Networking.
- CSC Mux 8 requires 4 x Cores, 8GB RAM and Accelerated Networking.

→ Click "Next"

Microsoft Azure

Search resources, services, and docs (G+/I)

Home > Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA (preview) >

Create Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

✓ Basics ✓ Virtual Machine Settings **3 Networking** 4 configUserData.json File 5 Review + create

Configure virtual networks

VNET_Name * ⓘ VNET-East-US Create new

EXTERNAL_Subnet_Name * ⓘ csc-external-East-US (10.2.1.0/24) Manage subnet configuration

INTERNAL_Subnet_Name * ⓘ csc-internal-East-US (10.2.2.0/24) Manage subnet configuration

< Previous Next

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

Microsoft Azure

Search resources, services, and docs (G+/I)

Home > Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA (preview) >

Create Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

✓ Basics ✓ Virtual Machine Settings ✓ Networking **4 configUserData.json File** 5 Review + create

(Optional) Paste here configUserData.json file:

configUserData.json file { "model": "csc-mux-ns-azure", "type": "configUserData", "versi..." } ✓

< Previous Next

(Optional) -> Paste configUserData.json file.

Via configUserData.json file, you can pass values to parameters during the installation of the CSC. You can setup: DNS Servers, Syslog Servers, AWS SSM Agent, IPSec Tunnel Nodes (auto discovery or manually), and the URLs for "Proxy Bypass" and "Routed Bypass".

Example of configUserData.json

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

configUserData.json

```
{
  "model": "csc-mux-ns-azure",
  "type": "configUserData",
  "version": "1.0",
  "dns": {
    "useCloudDNS": true,
    "primaryDnsIP": "",
    "secondaryDnsIP": ""
  },
  "syslogServers": {
    "primarySyslogIP": "10.2.3.4",
    "secondarySyslogIP": "",
    "syslogTcpPort": 514
  },
  "ssmAgent": {
    "activationCode": "eWvxKiEnssPknMpR2VP6",
    "activationID": "a7a5f1ab-2373-4dc0-adb6-c4f004a8968a",
    "awsRegion": "eu-west-2"
  },
  "ipsecInformation": {
    "autoDiscovery": false,
    "primaryIpsecPop": "163.116.146.38",
    "primaryProbepAddress": "10.146.6.216",
    "primaryLocation": "US,Washington,IAD2",
    "secondaryIpsecPop": "163.116.135.38",
    "secondaryProbepAddress": "10.135.6.216",
    "secondaryLocation": "US,NewYork,NYC1"
  },
  "bypassProxyPacUrl": "https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-bypass-vm.pac",
  "routedBypassJsonFileUrl": "https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json"
}
```

→ Click Next.

Create Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA

✓ Validation Passed

Check

✓ Basics

✓ Virtual Machine Settings

✓ Networking

✓ configUserData.json File

1 Review + create

PRODUCT DETAILS

Cloud Security Connector Mux (model 4 and 8) for Netskope with PriCPA
by Maidenhead Bridge
[Microsoft Enterprise Contract](#) | [Privacy policy](#)

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Name

Preferred e-mail address *

Preferred phone number *

Basics

Subscription	Pay-As-You-Go
Resource group	CSC-East-US
Location	East US
Select Single or HA configuration	Deploy High Availability (2x) CSCs
First CSC Availability Zone	Zone 1
Second CSC Availability Zone	Zone 2
CSC_Name	csc-mux-4-az
Admin Username	cscadmin
Password	*****

Virtual Machine Settings

Virtual machine size	Standard_D4s_v4
CSC VM Disk storage account type	Standard_LRS

Networking

Virtual network	VNET-East-US
EXTERNAL_Subnet_Name	csc-external-East-US
Address prefix (EXTERNAL_Subnet_Name)	10.2.1.0/24
INTERNAL_Subnet_Name	csc-internal-East-US
Address prefix (INTERNAL_Subnet_Name)	10.2.2.0/24

configUserData.json File

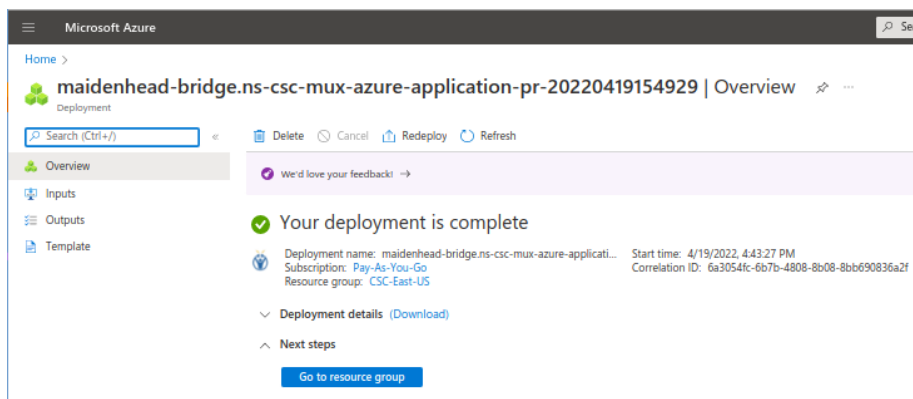
Create

< Previous

Next >

[Download a template for automation](#)

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



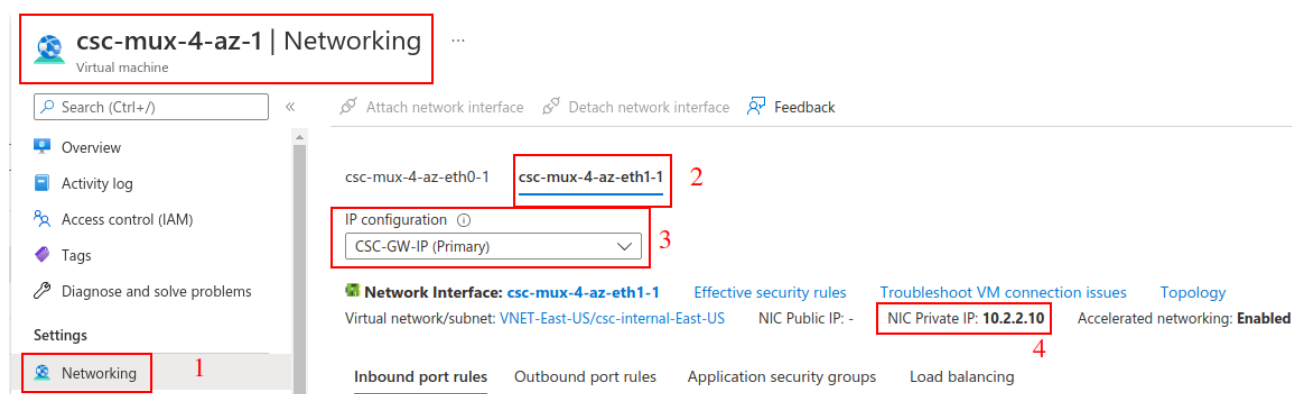
-> Click "Go to resource group" and you will see the components created.

Name ↑	Type ↑
csc-mux-4-az-1	Virtual machine
csc-mux-4-az-1_OsDisk_1_1c7026d54b84124b78a4040102774b	Disk
csc-mux-4-az-2	Virtual machine
csc-mux-4-az-2_OsDisk_1_29e7a5190e4c1686d7124934634b3f	Disk
csc-mux-4-az-eth0-0-PublicIp-1	Public IP address
csc-mux-4-az-eth0-0-PublicIp-2	Public IP address
csc-mux-4-az-eth0-1	Network interface
csc-mux-4-az-eth0-1-PublicIp-1	Public IP address
csc-mux-4-az-eth0-1-PublicIp-2	Public IP address
csc-mux-4-az-eth0-2	Network interface
csc-mux-4-az-eth0-2-PublicIp-1	Public IP address
csc-mux-4-az-eth0-2-PublicIp-2	Public IP address
csc-mux-4-az-eth0-3-PublicIp-1	Public IP address
csc-mux-4-az-eth0-3-PublicIp-2	Public IP address
csc-mux-4-az-eth0-4-PublicIp-1	Public IP address
csc-mux-4-az-eth0-4-PublicIp-2	Public IP address
csc-mux-4-az-eth0-NSG-1	Network security group
csc-mux-4-az-eth0-NSG-2	Network security group
csc-mux-4-az-eth1-1	Network interface
csc-mux-4-az-eth1-2	Network interface
csc-mux-4-az-eth1-NSG-1	Network security group
csc-mux-4-az-eth1-NSG-2	Network security group

→ Done! Your CSCs Mux for Azure are deployed.

5 Accessing for first time to your CSC

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



2. In this example, “NIC Private IP” is: 10.2.2.10
3. From a machine inside the Virtual Network, ssh the CSC using username “cscadmin” and key or password:

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

```
ssh cscadmin@<eth1 Private IP>
```

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

CSC Mux initial screen shows the tunnel information to import in the Netskope console.

```
Maidenhead Bridge
Cloud Security Connector Mux on Azure for Netskope - Admin Console

On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and check 'IPSEC Tunnels' to validate you imported the CSV file shown below.

CSV file:

tunnel_name,source_identity,source_ip_address,primary_pop,failover_pop,encryption_cipher,psk,maximum_bandwidth,enabled
csc-mux-4-az-1.nstun1,20.127.156.167,20.127.156.167,IAD2,NYC1,AES128-CBC,wGUM0RD0LqTFJl0vS8i40FanXGJUIcF,250,true
csc-mux-4-az-1.nstun2,20.127.156.168,20.127.156.168,IAD2,NYC1,AES128-CBC,wYA6cLokIP7RS2bpxqypEd0UvZ90d5NS,250,true
csc-mux-4-az-1.nstun3,20.127.156.169,20.127.156.169,IAD2,NYC1,AES128-CBC,10QPKSV8cu6r5dMgblm4u83yPNZU,250,true
csc-mux-4-az-1.nstun4,20.127.156.166,20.127.156.166,IAD2,NYC1,AES128-CBC,05r3RJvgBXXuMhigXPHtUCwpAhk0PDgu,250,true

Instructions to Import the CSV file:
1 - Copy and paste the CSV file's contents on a Text Editor and save it as '<filename>.csv'. Important: Do not add blank lines at the end of the file.
2 - On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and click 'IMPORT TUNNELS FROM CSV' and select the CSV file.

Did you 'Import tunnels from CSV' on the Netskope console? Please, confirm.
1) Yes
2) No
Enter your choice: █
```

Create a CSV file with this information

5.1 Creating the IPsec tunnels on Netskope console

The instructions are :

→ Copy and paste the CSV file's contents on a Text Editor and save it as '<filename>.csv'.

Important: Do not add blank lines at the end of the file.

Here the information of the initial screen:

```
Maidenhead Bridge
Cloud Security Connector Mux on Azure for Netskope - Admin Console

On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and check 'IPSEC Tunnels' to validate you imported the CSV file shown below.

CSV file:

tunnel_name,source_identity,source_ip_address,primary_pop,failover_pop,encryption_cipher,psk,maximum_bandwidth,enabled
csc-mux-4-az-1-nstun1,20.127.156.167,20.127.156.167,IAD2,NYC1,AES128-CBC,wdGUN0RD0lqtF3l0vs8i40FanXG3Uicf,250,true
csc-mux-4-az-1-nstun2,20.127.156.168,20.127.156.168,IAD2,NYC1,AES128-CBC,WYA6cl0kIP7RS2bpxqypEd0QVZg0d5NS,250,true
csc-mux-4-az-1-nstun3,20.127.156.169,20.127.156.169,IAD2,NYC1,AES128-CBC,1oQPbKSY8cuk6x5dNM0ibnM4u8JyFWZU,250,true
csc-mux-4-az-1-nstun4,20.127.156.166,20.127.156.166,IAD2,NYC1,AES128-CBC,Q5rJRJvgBXXuXPMtUCwpAhk0PDgu,250,true

Instructions to Import the CSV file:
1 - Copy and paste the CSV file's contents on a Text Editor and save it as '<filename>.csv'. Important: Do not add blank lines at the end of the file.
2 - On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and click 'IMPORT TUNNELS FROM CSV' and select the CSV file.

Did you 'Import tunnels from CSV' on the Netskope console? Please, confirm.
1) Yes
2) No
Enter your choice: █
```

Create a CSV file with this information

→ On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and click "IMPORT TUNNELS FROM CSV" and select the CSV file.

Import IPsec Tunnels

Previewing file: csc-netskope-ipsec-tunnels-7.csv

4 tunnel configurations found

NAME	SOURCE IP ADD..	SOURCE IDENT..	PRIMARY POP	FAILOVER POP	ENCRYPTION	PSK	MAX BANDWID..
csc-mux-4-az-1.i	20.127.156.167	20.127.156.167	IAD2	NYC1	AES128-CBC	wdGUN0RDQlq	250 Mbps
csc-mux-4-az-1.i	20.127.156.168	20.127.156.168	IAD2	NYC1	AES128-CBC	WYA6cl0kIP7RS	250 Mbps
csc-mux-4-az-1.i	20.127.156.169	20.127.156.169	IAD2	NYC1	AES128-CBC	1oQPbKSY8cuk	250 Mbps
csc-mux-4-az-1.i	20.127.156.166	20.127.156.166	IAD2	NYC1	AES128-CBC	Q5rJRJvgBXXu	250 Mbps


Importing tunnels will not delete your existing tunnels

CANCEL

IMPORT DIFFERENT FILE

IMPORT

→ Click "IMPORT" and wait a while to see the tunnels up.



← Security Cloud Platform

Configuration

TRAFFIC STEERING

Steering Configuration

App Definition

Publishers

IPSec

GRE

Explicit Proxy

NETSKOPE CLIENT

Users

Security Cloud Platform > Traffic Steering >

IPSec

Create and manage secure IPSec tunnels from your source devices such as routers and firewalls to Netskope's point of presence (POPs). View Supported IPSec Options [here](#).

ADD NEW TUNNEL

IMPORT TUNNELS FROM CSV

IPSec Tunnels

4 TUNNELS

Sort by: Status

<input type="checkbox"/> STATUS ↑	NAME	SOURCE IDENTITY	PRIMARY POP	ENCRYPTION
<input type="checkbox"/> ↑	csc-mux-4-az-1.nstun1	20.127.156.167	163.116.146.38 (IAD2 - Washington, DC, US)	AES128-CBC
<input type="checkbox"/> ↑	csc-mux-4-az-1.nstun2	20.127.156.168	163.116.146.38 (IAD2 - Washington, DC, US)	AES128-CBC
<input type="checkbox"/> ↑	csc-mux-4-az-1.nstun3	20.127.156.169	163.116.146.38 (IAD2 - Washington, DC, US)	AES128-CBC
<input type="checkbox"/> ↑	csc-mux-4-az-1.nstun4	20.127.156.166	163.116.146.38 (IAD2 - Washington, DC, US)	AES128-CBC

→ Done! The CSC Mux is ready. If you deployed 2 x CSCs, please repeat the process for the other CSC.

5.2 Checking status on the CSC console.

→ On the initial screen, confirm the Import of the CSV file:

```
Did you 'Import tunnels from CSV' on the Netskope console? Please, confirm.
1) Yes
2) No
Enter your choice: 1
```

→ and run "Show Configuration and Status". Check the Load Balancer and Tunnel Information:

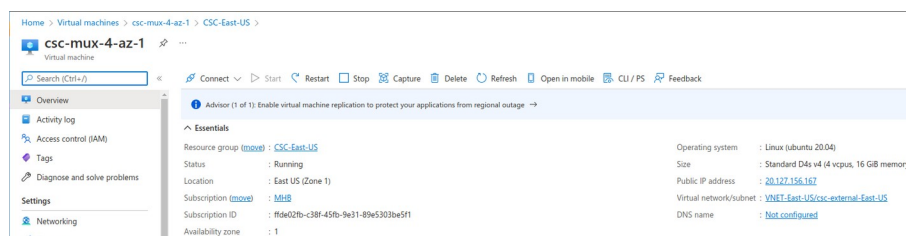
```
NETSKOPE INFORMATION
Tunnels Name:
  NSTun1: csc-mux-4-az-1.nstun1
  NSTun2: csc-mux-4-az-1.nstun2
  NSTun3: csc-mux-4-az-1.nstun3
  NSTun4: csc-mux-4-az-1.nstun4
Primary Tunnel:
  Node : US,Washington,IAD2
  Node Public IP: 163.116.146.38 is Alive
  Node Probe: 10.146.6.216
Secondary Tunnel:
  Node : US,NewYork,NYC1
  Node Public IP: 163.116.135.38 is Alive
  Node Probe: 10.135.6.216
LOAD BALANCING INFORMATION
Last change: Wed 20 Apr 16:35:33 UTC 2022
(UP) NSTun1 is active, using primary.
(UP) NSTun2 is active, using primary.
(UP) NSTun3 is active, using primary.
(UP) NSTun4 is active, using primary.
IPSEC INFORMATION
NSTun1 connected to: US,Washington,IAD2, IPsec uptime: 29 minutes, since Apr 20 16:21:24 2022, Last Security Association: ESTABLISHED 29 minutes ago
NSTun2 connected to: US,Washington,IAD2, IPsec uptime: 16 minutes, since Apr 20 16:34:21 2022, Last Security Association: ESTABLISHED 16 minutes ago
NSTun3 connected to: US,Washington,IAD2, IPsec uptime: 29 minutes, since Apr 20 16:21:34 2022, Last Security Association: ESTABLISHED 29 minutes ago
NSTun4 connected to: US,Washington,IAD2, IPsec uptime: 16 minutes, since Apr 20 16:34:19 2022, Last Security Association: ESTABLISHED 16 minutes ago
HTTP://WWW.NOTSKOPE.COM PAGE STATUS
NSTun1 is connected to 163.116.146.117 Ashburn, United States (IAD2)
NSTun2 is connected to 163.116.146.115 Ashburn, United States (IAD2)
NSTun3 is connected to 163.116.146.119 Ashburn, United States (IAD2)
NSTun4 is connected to 163.116.146.114 Ashburn, United States (IAD2)
```

→ Done!

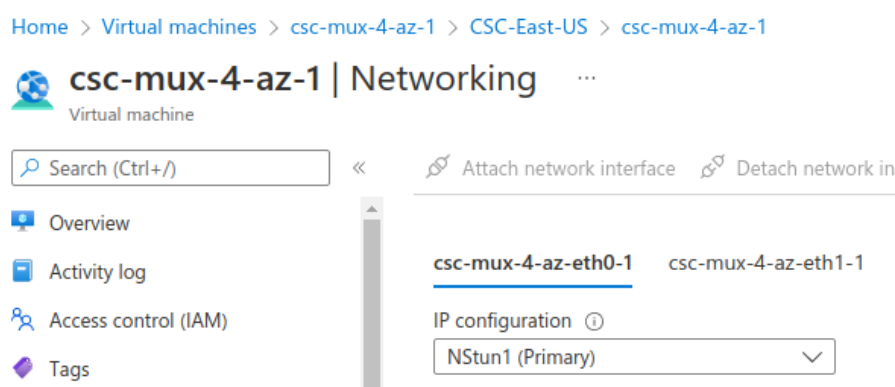
6 Resources creates by the ARM template

The following resources are created by the ARM template:

1. Virtual Machine



2. Interfaces External and Internal.



3. 4 (or 8) x Public IP for the IPsec tunnels and 1 x Public IP used by Bypass functionality and Private Access.

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

4.1. Inbound Rules

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
Inbound Security Rules						
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✓ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny

- 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 IPsec traffic. Therefore, there is double protection: The Azure Security Group and the internal Firewall Rules of the CSC.
- 2 When using Private Access (PriCPA), the CSC automatically updates the internal FW rules and Security Groups to allow Peers to communicate with each other.

4.2. Outbound Rules

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
> Inbound Security Rules						
▼ Outbound Security Rules						
4000	AllowPing	Any	ICMP	Any	Any	✔ Allow
4010	AllowUDP500	500	UDP	Any	Any	✔ Allow
4020	AllowUDP4500	4500	UDP	Any	Any	✔ Allow
4030	AllowHTTP	80	TCP	Any	Any	✔ Allow
4040	AllowHTTPS	443	TCP	Any	Any	✔ Allow
4050	AllowPublicDNS	53	UDP	Any	Any	✔ Allow
4060	DenyAllOutbound	Any	Any	Any	Any	✖ Deny
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✔ Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	✔ Allow
65500	DenyAllOutBound	Any	Any	Any	Any	✖ Deny

5. Security Group for Internal Interface.

5.1. Inbound Rules

Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
▼ Inbound Security Rules						
4000	AllowNet-10.0.0.0-8	Any	Any	10.0.0.0/8	Any	✔ Allow
4010	AllowNet-172.16.0.0-12	Any	Any	172.16.0.0/12	Any	✔ Allow
4020	AllowNet-192.168.0.0-16	Any	Any	192.168.0.0/16	Any	✔ Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✔ Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✔ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✖ Deny

5.2. Outbound Rules

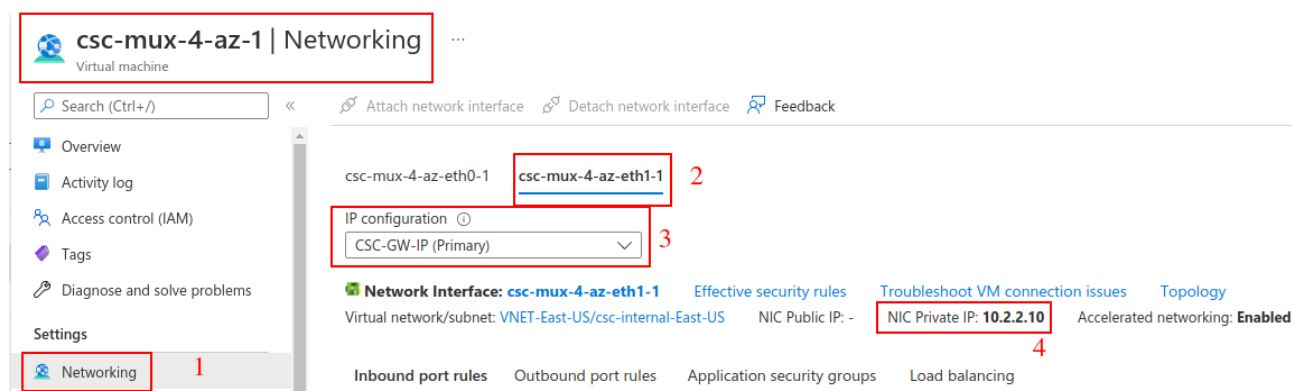
> Inbound Security Rules						
▼ Outbound Security Rules						
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✔ Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	✔ Allow
65500	DenyAllOutBound	Any	Any	Any	Any	✖ Deny

7 The Cloud Security Connector Admin Console:

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

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

For example:



```
ssh cscadmin@10.2.2.10
```

CSC Mux initial screen shows the tunnel information to import in the Netskope console.

```
Maidenhead Bridge
Cloud Security Connector Mux on Azure for Netskope - Admin Console

On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and check 'IPSEC Tunnels' to validate you imported the CSV file shown below.

CSV file:

tunnel name,source identity,source ip address,primary pop,failover pop,encryption cipher,psk,maximum bandwidth,enabled
csc-mux-4-az-1.nstun1,20.127.156.167,20.127.156.167,IAD2,NYC1,AES128-CBC,wdGUN0RD0lqtFJl0vS8i40FanXGJUIcF,250,true
csc-mux-4-az-1.nstun2,20.127.156.168,20.127.156.168,IAD2,NYC1,AES128-CBC,WYA6cLokIP7RS2bpxqypEd0QVZg0d5NS,250,true
csc-mux-4-az-1.nstun3,20.127.156.169,20.127.156.169,IAD2,NYC1,AES128-CBC,1oQPbKSY8cuk6x5dNMgibnM4u8JyFWZU,250,true
csc-mux-4-az-1.nstun4,20.127.156.166,20.127.156.166,IAD2,NYC1,AES128-CBC,Q5rJRJvgBXXuMnigXPMTUCwPAhk0PDgu,250,true

Instructions to Import the CSV file:
1 - Copy and paste the CSV file's contents on a Text Editor and save it as '<filename>.csv'. Important: Do not add blank lines at the end of the file.
2 - On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and click 'IMPORT TUNNELS FROM CSV' and select the CSV file.

Did you 'Import tunnels from CSV' on the Netskope console? Please, confirm.
1) Yes
2) No
Enter your choice: █
```

Create a CSV file with this information

→ Select 1) Yes to confirm.

```
Maidenhead Bridge

CSC MUX 4 for Netskope on Azure - Admin Console

VM Name : csc-mux-4-az-1
Azure Region : eastus
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
```

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.

7.1 Monitoring Tasks

7.1.1 Show Configuration and Status

```
GENERAL INFORMATION
Name: csc-mux-4-az-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_D4s_v4
CSC date: Fri 22 Apr 09:41:49 UTC 2022
Soft version: 1.0 | CSC Model: CSC MUX 4 for Netskope on Azure
Azure Cloud: AzureCloud

INTERFACES INFORMATION
External: Tunnel IPs (eth0): 10.2.1.22-[23,24,25]/24 | Bypass Proxy Egress IP 10.2.1.26 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.10/24 | Network Gateway: 10.2.2.1

TRAFFIC REDIRECTION Options
To Netskope: VIP Proxy: 10.2.2.11:80 | 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: 10.2.2.12:3128 | Netskope Global Proxy IP: 163.116.128.80:3128 via CSC GW IP

PUBLIC IP Address INFORMATION
IPsec tunnels Public IP: 20.127.156.167, 20.127.156.168, 20.127.156.169, 20.127.156.166
Bypass Public IP: 52.249.220.21

DNS INFORMATION
Using Azure DNS: 168.63.129.16

NETSKOPE INFORMATION
Tunnels Name:
    NStun1: csc-mux-4-az-1.nstun1
    NStun2: csc-mux-4-az-1.nstun2
    NStun3: csc-mux-4-az-1.nstun3
    NStun4: csc-mux-4-az-1.nstun4
Primary Tunnel:
    Node : US,Washington,IA02
    Node Public IP: 163.116.146.38 is Alive
    Node Probe: 10.146.6.216
Secondary Tunnel:
    Node : US,NewYork,NYC1
    Node Public IP: 163.116.135.38 is Alive
    Node Probe: 10.135.6.216

LOAD BALANCING INFORMATION
Last change: Fri 22 Apr 03:06:20 UTC 2022
(UP) NStun1 is active, using primary.
(UP) NStun2 is active, using primary.
(UP) NStun3 is active, using primary.
(UP) NStun4 is active, using primary.

IPSEC INFORMATION
NStun1 connected to: US,Washington,IA02, IPsec uptime: 6 hours, since Apr 22 03:05:09 2022, Last Security Association: ESTABLISHED 6 hours ago
NStun2 connected to: US,Washington,IA02, IPsec uptime: 6 hours, since Apr 22 03:05:09 2022, Last Security Association: ESTABLISHED 6 hours ago
NStun3 connected to: US,Washington,IA02, IPsec uptime: 6 hours, since Apr 22 03:05:09 2022, Last Security Association: ESTABLISHED 6 hours ago
NStun4 connected to: US,Washington,IA02, IPsec uptime: 6 hours, since Apr 22 03:05:09 2022, Last Security Association: ESTABLISHED 6 hours ago

HTTP://WWW.NETSKOPE.COM PAGE STATUS
NStun1 is connected to 163.116.146.118 Ashburn, United States (IA02)
NStun2 is connected to 163.116.146.119 Ashburn, United States (IA02)
NStun3 is connected to 163.116.146.116 Ashburn, United States (IA02)
NStun4 is connected to 163.116.146.114 Ashburn, United States (IA02)

PROXY BYPASS - EGRESS INTERFACE STATUS
Proxy Bypass Egress Interface 10.2.1.26 can reach test page (https://ip.maidenheadbridge.com) via Public IP 52.249.220.21

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

AWS SSM AGENT
AWS SSM Agent is active (running) since Thu 2022-04-21 11:50:19 UTC; 21h ago
Registration values: {"ManagedInstanceID":"mi-098d04570e3254cda","Region":"eu-west-2"}

SYSLOG INFORMATION
SYSLOG Server (1) IP: 10.2.3.4 is Alive
SYSLOG Server (2) IP is not configured
SYSLOG TCP Port: 514

HIGH AVAILABILITY Information
The HA service is: active (running) since Fri 2022-04-22 01:21:00 UTC; 8h ago
Identity Type: SystemAssigned
Route to Netskope using Next Hop: 10.2.2.13 of VM: csc-mux-4-az-2 (the other CSC in the pair)
Current values configured are:
    Route/s (Qty)= 2
    Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
    Route 2: to-163.116.128.80 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
    Computer Name of other CSC in the pair: csc-mux-4-az-2 (Resource Group=CSC-East-US)
    Private Access Public IP= 20.127.156.184
```

7.1.1.1 GENERAL INFORMATION

This section contains general information about the instance:

```
GENERAL INFORMATION
Name: csc-mux-4-az-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_D4s_v4
CSC date: Wed 20 Apr 16:51:12 UTC 2022
Soft version: 1.0 | CSC Model: CSC MUX 4 for Netskope on Azure
Azure Cloud: AzureCloud
```

7.1.1.2 INTERFACES INFORMATION

This section contains the interfaces information:

```
INTERFACES INFORMATION
External: Tunnel IPs (eth0): 10.2.1.22-[23,24,25]/24 | Bypass Proxy Egress IP 10.2.1.26 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.10/24 | Network Gateway: 10.2.2.1
```

7.1.1.3 TRAFFIC REDIRECTION Options

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

```
TRAFFIC REDIRECTION Options
To Netskope: VIP Proxy: 10.2.2.11:80 | 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: 10.2.2.12:3128 | Netskope Global Proxy IP: 163.116.128.80:3128 via CSC GW IP
```

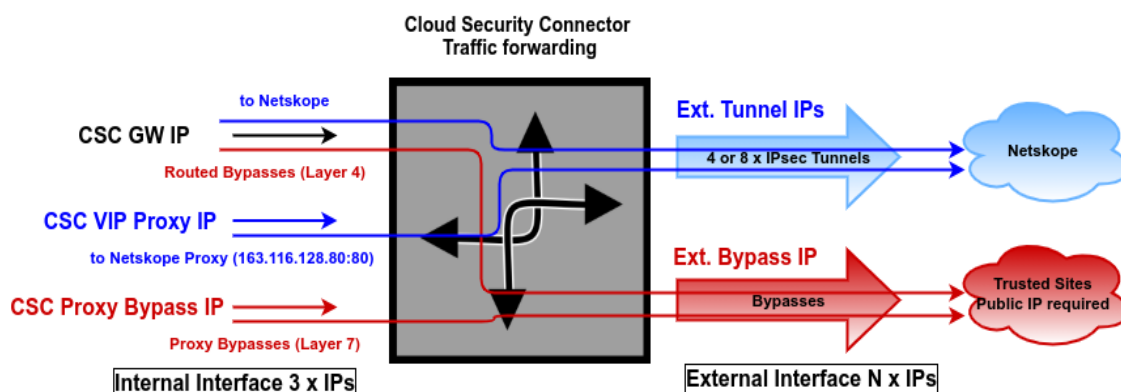
The objective of the Cloud Security Connectors of Maidenhead Bridge is to provide a simple architecture, 100% proven that works when connecting to Netskope.

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

Here an illustration about this:



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

7.1.1.4 PUBLIC IP Address INFORMATION

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

```
PUBLIC IP Address INFORMATION
IPsec tunnels Public IP: 20.127.156.167, 20.127.156.168, 20.127.156.169, 20.127.156.166
Bypass Public IP: 52.249.220.21
```

7.1.1.5 DNS INFORMATION

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

```
DNS INFORMATION
Using Azure DNS: 168.63.129.16
```

7.1.1.6 NETSKOPE INFORMATION

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

```

NETSKOPE INFORMATION
Tunnels Name:
    NStun1: csc-mux-4-az-1.nstun1
    NStun2: csc-mux-4-az-1.nstun2
    NStun3: csc-mux-4-az-1.nstun3
    NStun4: csc-mux-4-az-1.nstun4
Primary Tunnel:
    Node : US,Washington,IAD2
    Node Public IP: 163.116.146.38 is Alive
    Node Probe: 10.146.6.216
Secondary Tunnel:
    Node : US,NewYork,NYC1
    Node Public IP: 163.116.135.38 is Alive
    Node Probe: 10.135.6.216

```

7.1.1.7 LOAD BALANCING INFORMATION

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

```

LOAD BALANCING INFORMATION
Last change: Wed 20 Apr 16:35:33 UTC 2022
(UP)  NStun1 is active, using primary.
(UP)  NStun2 is active, using primary.
(UP)  NStun3 is active, using primary.
(UP)  NStun4 is active, using primary.

```

7.1.1.8 IPSEC INFORMATION

This section shows the status of each IPsec tunnel.

```

IPSEC INFORMATION
NStun1 connected to: US,Washington,IAD2, IPsec uptime: 29 minutes, since Apr 20 16:21:24 2022, Last Security Association: ESTABLISHED 29 minutes ago
NStun2 connected to: US,Washington,IAD2, IPsec uptime: 16 minutes, since Apr 20 16:34:21 2022, Last Security Association: ESTABLISHED 16 minutes ago
NStun3 connected to: US,Washington,IAD2, IPsec uptime: 29 minutes, since Apr 20 16:21:34 2022, Last Security Association: ESTABLISHED 29 minutes ago
NStun4 connected to: US,Washington,IAD2, IPsec uptime: 16 minutes, since Apr 20 16:34:19 2022, Last Security Association: ESTABLISHED 16 minutes ago

```

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

This section shows the result of a HTTP GET from inside each tunnel (NStunx) to URL:

<http://www.otskope.com>

```

HTTP://WWW.NOTSKOPE.COM PAGE STATUS
NStun1 is connected to 163.116.146.117 Ashburn, United States (IAD2)
NStun2 is connected to 163.116.146.115 Ashburn, United States (IAD2)
NStun3 is connected to 163.116.146.119 Ashburn, United States (IAD2)
NStun4 is connected to 163.116.146.114 Ashburn, United States (IAD2)

```

7.1.1.10 PROXY BYPASS - EGRESS INTERFACE STATUS

This section validates if the Proxy Bypass can access internet directly going to

<https://ip.maidenheadbridge.com>

```
PROXY BYPASS - EGRESS INTERFACE STATUS
Proxy Bypass Egress Interface 10.2.1.26 can reach test page (https://ip.maidenheadbridge.com) via Public IP 20.127.156.184
```

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

7.1.1.12 AWS SSM AGENT

This section shows the status of the AWS SSM Agent.

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Tue 2022-04-19 15:47:18 UTC; 1 day 1h ago
Registration values: {"ManagedInstanceID":"mi-098d04570e3254cda","Region":"eu-west-2"}
```

7.1.1.13 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: 10.2.3.4 is Alive
SYSLOG Server (2) IP is not configured
SYSLOG TCP Port: 514
```

7.1.1.14 HIGH AVAILABILITY Information

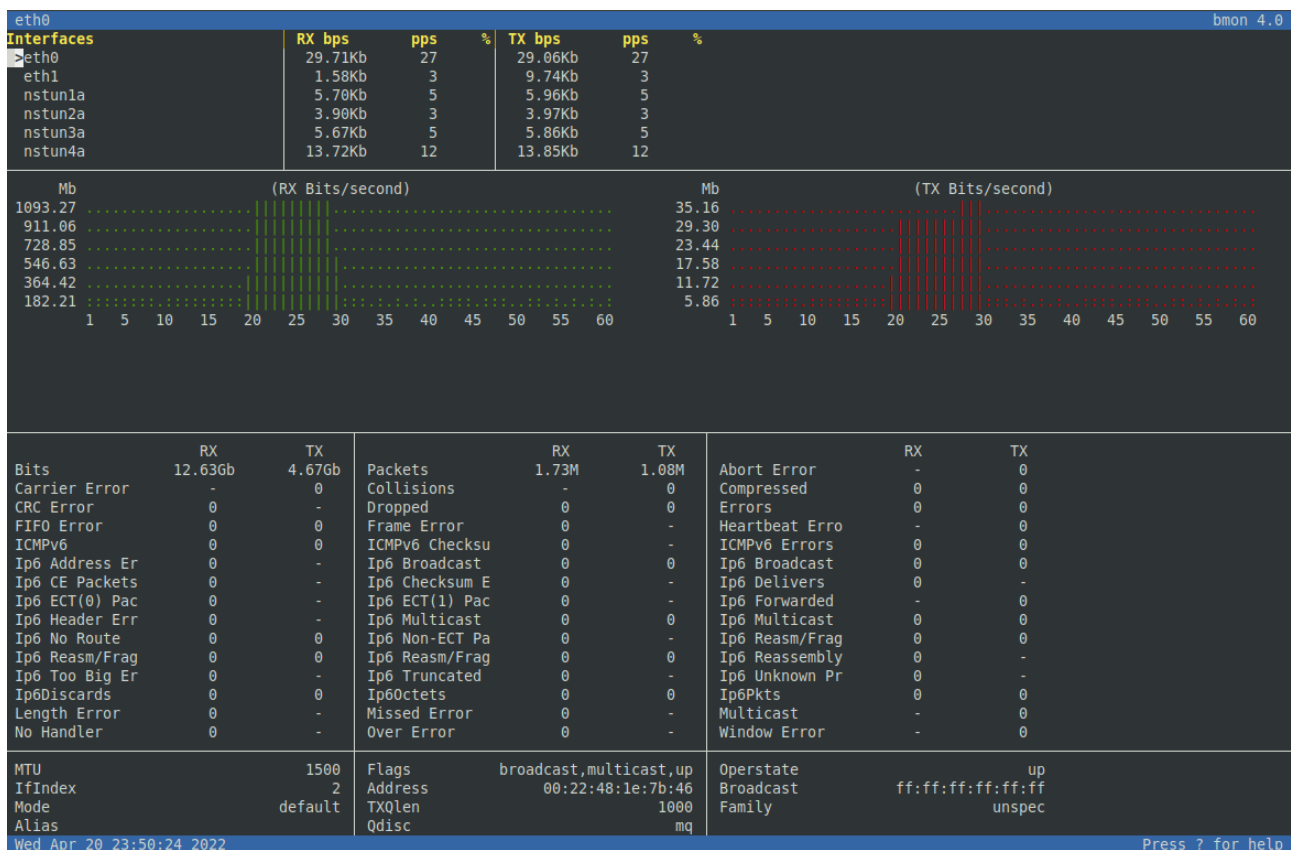
This section shows all the information when the CSC Mux is configured on HA:

```
HIGH AVAILABILITY Information
The HA service is: active (running) since Thu 2022-04-21 12:54:39 UTC; 11h ago
Identity Type: SystemAssigned
Route to Netskope using Next Hop: 10.2.2.10 of VM: csc-mux-4-az-1 (this CSC)
Current values configured are:
  Route/s (Qty)= 2
    Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
    Route 2: to-163.116.128.80 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
  Computer Name of other CSC in the pair: csc-mux-4-az-2 (Resource Group=CSC-East-US)
  Private Access Public IP= 20.127.156.184
```

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

7.1.2 Show Interfaces Traffic

Use this section to see the traffic in real time.



7.1.3 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.

```

Selection: 3

My TraceRoute (MTR) Test Report
This test does 10 probes DIRECT to Primary / Secondary IPsec Nodes and a Reverse test via NStun1 to NStun1 Public IP
Notes:
- IMPORTANT: It is required to allow ICMP Time exceeded (type 11) on the Inbound rule of the Security Group of eth0 to destination IP: 10.2.1.22
Without this security rule added, you will not be able to see the results of middle hops.
- When the NStun1 is UP, a Reverse Path test from the active Node to NStun1 Public IP is performed
- Max Hops is equal 30. This test can take a while

Testing Primary Node: US,Washington,IAD2 : > 163.116.146.38
Start: 2022-04-21T11:32:26+0000
HOST: csc-mux-4-az-1

```

	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
2. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
3. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
4. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
5. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
6. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
7. AS8075 be-160-0.ibr04.bl20.ntwk.msn.net (104.44.22.211)	0.0%	10	2.4	2.4	2.1	3.2	0.3
8. AS8075 ae162-0.icr02.bl20.ntwk.msn.net (104.44.21.234)	0.0%	10	1.8	4.1	1.6	20.3	5.8
9. AS8075 ae55-0.ash-96cbe-1b.ntwk.msn.net (104.44.238.132)	0.0%	10	15.3	4.8	1.5	15.3	5.0
10. AS??? as55256.ashburn.megaport.com (206.53.170.37)	0.0%	10	1.6	1.6	1.6	1.8	0.1
11. AS55256 163.116.146.38	0.0%	10	1.7	1.8	1.7	1.9	0.1

```

Testing Secondary Node: US,NewYork,NYC1 : > 163.116.135.38
Start: 2022-04-21T11:32:43+0000
HOST: csc-mux-4-az-1

```

	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
2. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
3. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
4. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
5. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
6. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
7. AS8075 be-164-0.ibr04.bl20.ntwk.msn.net (104.44.32.42)	0.0%	10	8.1	8.1	7.8	8.3	0.2
8. AS8075 be-9-0.ibr02.nyc30.ntwk.msn.net (104.44.28.54)	0.0%	10	7.9	15.4	7.9	76.8	21.6
9. AS8075 ae24-0.ear05.nyc30.ntwk.msn.net (104.44.33.214)	0.0%	10	8.7	8.4	7.5	10.3	0.7
10. AS8075 ae27-0.ier01.ewr30.ntwk.msn.net (104.44.231.67)	0.0%	10	7.2	7.3	6.8	9.9	0.9
11. AS??? UnAssigned34.nyiix.net (198.32.160.34)	0.0%	10	6.9	7.0	6.8	7.8	0.3
12. AS55256 163.116.135.38	0.0%	10	7.3	7.5	7.3	7.9	0.2

```

Reverse path from: US,Washington,IAD2 to your Public IP: 20.127.156.167
Start: 2022-04-21T11:33:00+0000
HOST: csc-mux-4-az-1

```

	Loss%	Snt	Last	Avg	Best	Wrst	StDev
1. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
2. AS??? 10.146.6.216	0.0%	10	1.8	3.7	1.8	12.8	3.9
3. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0
4. AS55256 163.116.146.2	0.0%	10	2.3	2.3	2.0	2.6	0.2
5. AS??? as8075.ashburn.megaport.com (206.53.170.12)	0.0%	10	2.5	3.1	2.4	6.1	1.2
6. AS8075 ae72-0.icr02.bl7.ntwk.msn.net (104.44.233.21)	10.0%	10	2.7	4.1	2.6	12.0	3.0
7. AS8075 be-162-0.ibr04.bl7.ntwk.msn.net (104.44.21.217)	10.0%	10	3.7	4.6	3.1	12.6	3.0
8. AS8075 ae164-0.icr03.bl7.ntwk.msn.net (104.44.32.35)	0.0%	10	2.8	4.5	2.7	18.3	4.9
9. AS??? ???	100.0	10	0.0	0.0	0.0	0.0	0.0

7.1.4 SPEED TEST

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

```

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

Aggregated Bandwidth Download: 913.35 Mbps

```

Note: At the moment of writing this documentation, Netskope provides 250 Mbps per IPsec tunnel. The CSC Mux 4 can aggregate 4 x IPsec tunnels (~ 1 Gbps total), and the CSC Mux 8 can aggregate 8 x IPsec tunnels (~ 2 Gbps Total).

7.2 CSC Admin Tasks

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

7.2.1 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".

7.2.1.1 Create a "Hybrid Activation" from AWS console.

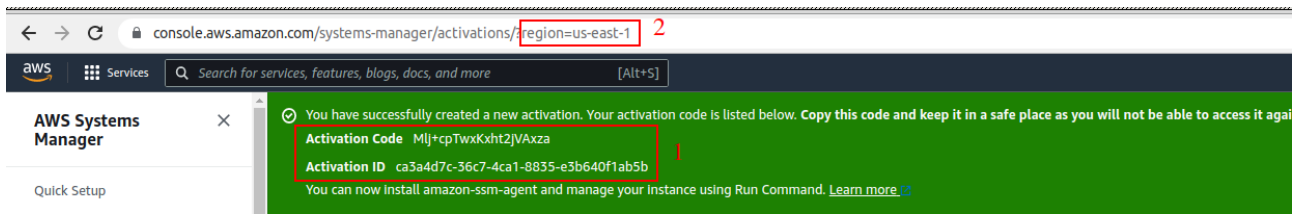
On your AWS Console, go to Services → Systems Manager → Node Management → Hybrid Activations and click "Create". Fill the values on shown below:

The screenshot shows the AWS Systems Manager console with the 'Create activation' page. The left sidebar shows the navigation menu with 'AWS Systems Manager' (1) and 'Hybrid Activations' (2) highlighted. The main content area shows the 'Create activation' form (3) with the following fields and annotations:

- Activation description- Optional** (4 - Put a name to the activation): A text box containing 'csc-gre-for-netskope-on-aws-a'.
- Instance limit**: A text box containing '1'.
- IAM role**: A radio button selected for 'Use the default role created by the system (AmazonEC2RunCommandRoleForManagedInstances)'.
- Activation expiry date**: A date picker showing 'yyyy-mm-ddThh:mm-00:00'.
- Default instance name- Optional** (5 - Repeat the name): A text box containing 'csc-gre-for-netskope-on-aws-a'.
- Create activation** button (6): A blue button at the bottom right of the form.

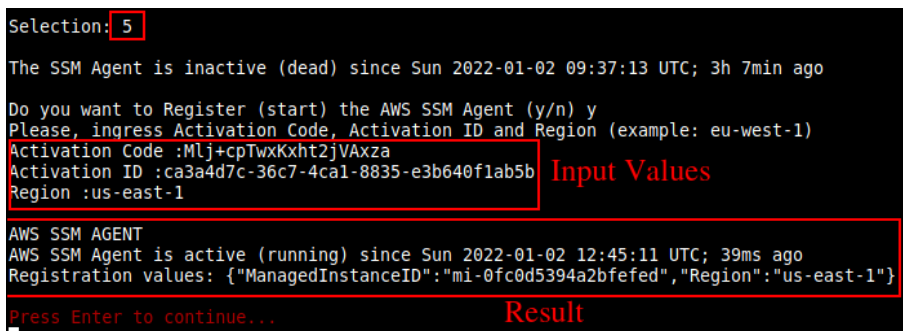
³ For Vmware, Hyper-V, KVM, Azure, Gcloud and AWS.

→ Click "Create Activation"

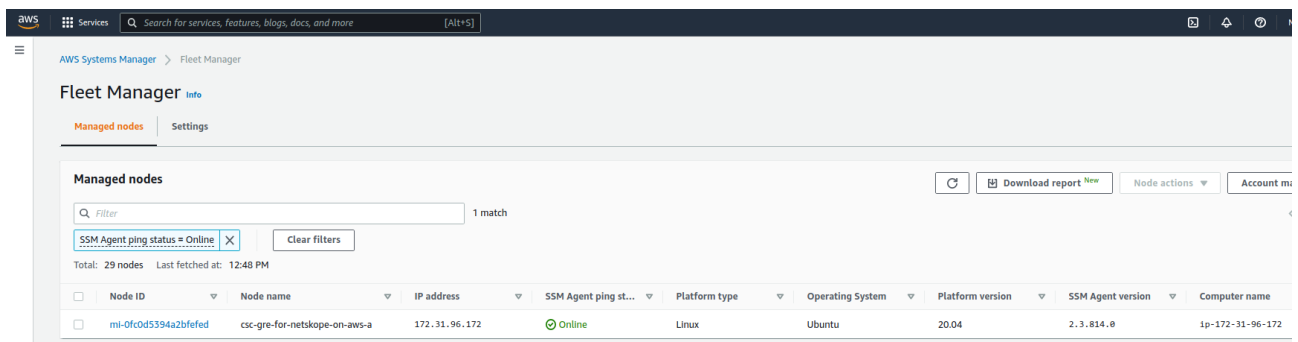


The values of Activation Code, Activation ID and Region are required to register the CSC. Keep these values in a safe place.

7.2.1.2 Register the CSC



7.2.1.3 View the Registered CSC on AWS Systems Manager



7.2.2 Manage Administrators

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

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

```
Selection: 6
Please, select the Administrator: 'cscadmin' or 'csccli'
1) cscadmin
2) csccli
3) Quit
Enter your choice: █
```

Note: the user "cscadmin" cannot be disabled.

7.2.2.1 "cscadmin" settings

```
Please, select the Administrator: 'cscadmin' or 'csccli'
1) cscadmin
2) csccli
3) Quit
Enter your choice: 1
Please, select the task to do:
1) Change Password
2) Manage SSH Keys
3) Quit
Enter your choice: █
```

7.2.2.2 "csccli" settings

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

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

User 'csccli' is not enabled.

Do you want to enable user 'csccli'?

1) Yes
2) No
Enter your choice: 1

User 'csccli' was enabled via console.

Please, input a SSH Key for user 'csccli'

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

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

Do you want to continue?

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

7.2.2.3 *Managing the SSH Key of a User*

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

7.2.3 Change Timezone

Use this menu to select the timezone of the CSC.

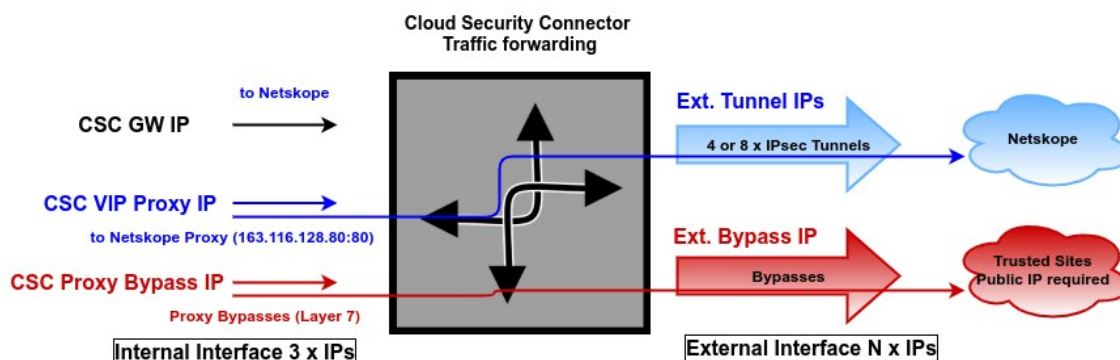


7.3 Proxy Bypass

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
```

7.3.1 Proxy Bypass - Traffic Flow



7.3.2 View Current Proxy Bypass List

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

```
Selection: 8
This is the list of current Domains configured:
.okta.com
.oktacdn.com
.okta-emea.com
.login.mydomain.com
.login.microsoftonline.com
.login.microsoft.com
.login.windows.net
.zoho.com
.portquiz.net
```

7.3.3 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: █
```

7.3.3.1 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 as 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"))) ||
        // 0365 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 successfully.
```

7.3.3.2 Manual Proxy Bypass Configuration.

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

1. Select Option 2)

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

2. Input "y"

```
GNU nano 4.8 domains Modified
.okta.com
.oktacdn.com
.okta-emea.com
login.mydomain.com
login.microsoftonline.com
login.microsoft.com
login.windows.net
portquiz.net
manualAdded.com
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^I To Spell   ^_ Go To Line  M-E Redo
```

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

```
This is your current Proxy Bypass List

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

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

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

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

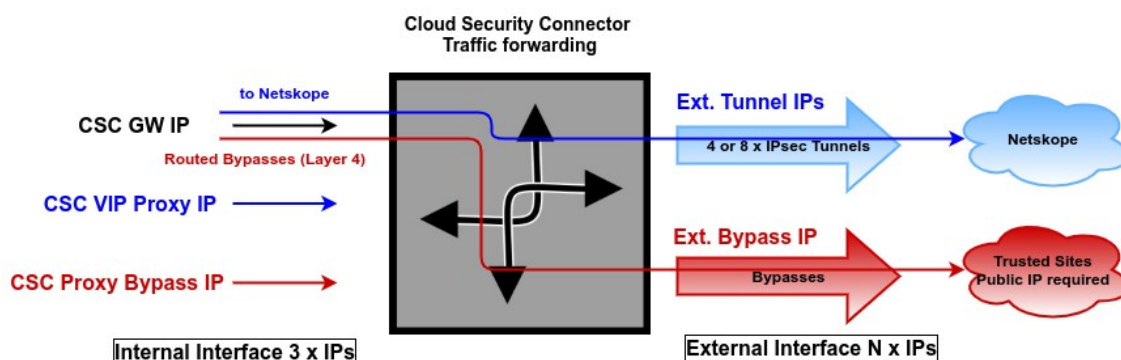
Proxy Bypass List updated sucessfully.
```

7.4 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
```

7.4.1 Routed Bypass - Traffic Flow



7.4.2 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: 
```

7.4.2.1 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"
```

7.4.2.2 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
```

7.4.3 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: █
```

7.4.3.1 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://mhbm-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: "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 successfully.
(Index: 1) Rule "0365 Login URLs 2" was created successfully.
(Index: 2) Rule "0365 Login URLs 3" was created successfully.
(Index: 3) Rule "portquiz.net" was created successfully.
(Index: 4) Rule "0365 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 successfully.
Press ENTER to continue
```

7.4.3.2 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
```

7.5 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
```

7.5.1 View Current Month

```
Selection: 12
Current Month (April 2022) Logs for csc-mux-4-az-1
Apr 19 15:46:05 root: (MHB-CSC)(DOWN) Load Balancer: All NStunnels are inactive since: Tue 19 Apr 15:46:05 UTC 2022
Apr 19 15:46:06 root: (MHB-CSC)(INFO) Routed Bypass - Routed Bypass Rules JSON file integrity is OK
Apr 19 15:46:07 root: (MHB-CSC)(INFO) Config User Data JSON file integrity is OK
Apr 19 15:46:07 root: (MHB-CSC)(INFO) DNS configured using AWS 168.63.129.16 and Google 8.8.8.8 servers
Apr 19 15:46:07 root: (MHB-CSC)(INFO) SYSLOG configured using Primary= 10.2.3.4, Secondary= none and TCP port= 514
Apr 19 15:46:10 root: (MHB-CSC)(INFO) AWS SSM Agent is active (running) since Tue 2022-04-19 15:46:10 UTC; 294ms ago. Registration values: {"ManagedInstanceId":"mi-098d04570e3254cda","Region":"eu-west-2"}
Apr 19 15:46:26 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-bypass-vm.pac)
Apr 19 15:46:28 root: (MHB-CSC)(INFO) IPSEC tunnels configured with Primary Node: 163.116.146.38 (US,Washington,IAD2), Secondary Node: 163.116.135.38 (US,NewYork,NYC1).
Apr 19 15:46:29 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.json).
Apr 19 15:46:29 root: (MHB-CSC)(INFO) Rebooting the CSC after loading configUserData.json.
Apr 19 15:46:29 root: (MHB-CSC)(UP) CSC Mux 4 for Azure was powered ON: Tue 19 Apr 15:46:29 UTC 2022
Apr 19 15:48:22 root: (MHB-CSC)(DOWN) Load Balancer: All NStunnels are inactive since: Tue 19 Apr 15:48:27 UTC 2022
Apr 19 15:48:27 root: (MHB-CSC)(INFO) Routed Bypass - Routed Bypass Rules JSON file integrity is OK
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 0) Rule "0365 Login URLs 1" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 1) Rule "0365 Login URLs 2" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 2) Rule "0365 Login URLs 3" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 3) Rule "portquiz.net" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 4) Rule "0365 Login URLs 4" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 5) Rule "Skype and Teams UDP 1" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 6) Rule "Skype and Teams UDP 2" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 7) Rule "Skype and Teams UDP 3" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 8) Rule "ip.maidenheadbridge.com 1" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 9) Rule "ip.maidenheadbridge.com 2" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 10) Rule "ip.maidenheadbridge.com 3" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 11) Rule "ip.maidenheadbridge.com 4" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(UP) CSC Mux 4 for Azure was powered ON: Tue 19 Apr 15:48:30 UTC 2022
```

7.5.2 View Last 6 Months

```
Selection: 13
Last 6 Months Logs up to Current Month (April 2022) for csc-mux-4-az-1
Apr 19 15:46:05 root: (MHB-CSC)(DOWN) Load Balancer: All NStunnels are inactive since: Tue 19 Apr 15:46:05 UTC 2022
Apr 19 15:46:06 root: (MHB-CSC)(INFO) Routed Bypass - Routed Bypass Rules JSON file integrity is OK
Apr 19 15:46:07 root: (MHB-CSC)(INFO) Config User Data JSON file integrity is OK
Apr 19 15:46:07 root: (MHB-CSC)(INFO) DNS configured using AWS 168.63.129.16 and Google 8.8.8.8 servers
Apr 19 15:46:07 root: (MHB-CSC)(INFO) SYSLOG configured using Primary= 10.2.3.4, Secondary= none and TCP port= 514
Apr 19 15:46:10 root: (MHB-CSC)(INFO) AWS SSM Agent is active (running) since Tue 2022-04-19 15:46:10 UTC; 294ms ago. Registration values: {"ManagedInstanceId":"mi-098d04570e3254cda","Region":"eu-west-2"}
Apr 19 15:46:26 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-bypass-vm.pac)
Apr 19 15:46:28 root: (MHB-CSC)(INFO) IPSEC tunnels configured with Primary Node: 163.116.146.38 (US,Washington,IAD2), Secondary Node: 163.116.135.38 (US,NewYork,NYC1).
Apr 19 15:46:29 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.json).
Apr 19 15:46:29 root: (MHB-CSC)(INFO) Rebooting the CSC after loading configUserData.json.
Apr 19 15:46:29 root: (MHB-CSC)(UP) CSC Mux 4 for Azure was powered ON: Tue 19 Apr 15:46:29 UTC 2022
Apr 19 15:48:22 root: (MHB-CSC)(DOWN) Load Balancer: All NStunnels are inactive since: Tue 19 Apr 15:48:27 UTC 2022
Apr 19 15:48:27 root: (MHB-CSC)(INFO) Routed Bypass - Routed Bypass Rules JSON file integrity is OK
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 0) Rule "0365 Login URLs 1" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 1) Rule "0365 Login URLs 2" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 2) Rule "0365 Login URLs 3" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 3) Rule "portquiz.net" was created successfully.
Apr 19 15:48:28 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 4) Rule "0365 Login URLs 4" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 5) Rule "Skype and Teams UDP 1" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 6) Rule "Skype and Teams UDP 2" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 7) Rule "Skype and Teams UDP 3" was created successfully.
Apr 19 15:48:29 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 8) Rule "ip.maidenheadbridge.com 1" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 9) Rule "ip.maidenheadbridge.com 2" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 10) Rule "ip.maidenheadbridge.com 3" was created successfully.
Apr 19 15:48:30 root: (MHB-CSC)(INFO) Routed Bypass - (Index: 11) Rule "ip.maidenheadbridge.com 4" was created successfully.
```

7.6 Configuration Wizards

In this section, you can run the Configuration Wizard to change IPsec 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.
```

7.6.1 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 MUX 4 for Netskope on Azure - Configuration Wizard

Current Values Configured:
-----
NETSKOPE INFORMATION
Primary Tunnel:
    Node : US,Washington,IAD2
    Node Public IP: 163.116.146.38 is Alive
    Node Probe: 10.146.6.216
Secondary Tunnel:
    Node : US,NewYork,NYC1
    Node Public IP: 163.116.135.38 is Alive
    Node Probe: 10.135.6.216
-----
DNS Servers: Azure DNS server 168.63.129.16 and Google DNS server 8.8.8.8
-----
Bypass Proxy PAC URL
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-bypass-vm.pac
-----
Routed Bypass URL
Your current Routed Bypass URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json
-----
Syslog / SIEM information
Primary Syslog / SIEM IP: 10.2.3.4
Secondary Syslog / SIEM IP: Not configured
Syslog / SIEM TCP port: 514
-----
Are you ready to continue?
1) Yes
2) No
Enter your choice: █
```

7.6.1.1 Running the Configuration Wizard

Select Nodes, Auto or Manual.

If you select "Auto" the CSC will select the nearest Netskope Nodes to "IPSec 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
Primary Tunnel:
Node : US,Washington,IAD2
Node Public IP: 163.116.146.38 is Alive
Node Probe: 10.146.6.216
Secondary Tunnel:
Node : US,NewYork,NYC1
Node Public IP: 163.116.135.38 is Alive
Node Probe: 10.135.6.216

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      7) CA,Montreal,YMQ1      13) FR,Paris,PAR1      19) IN,Delhi,DEL1      25) SG,Singapore,SIN1      31) US,NewYork,NYC1      37) Quit
2) AR,BuenosAires,BUE1  8) CA,Vancouver,YVR1    14) GB,London,LON1    20) IN,Mumbai,BOM1    26) US,Atlanta,ATL1    32) US,Phoenix,PHX1
3) AT,Vienna,VIE1      9) CL,Santiago,SCL1     15) GB,Manchester,MAN1  21) IT,Milan,MIL1     27) US,Chicago,ORD1    33) US,Seattle,SEA1
4) AU,Melbourne,MEL1   10) CO,Bogota,BOG1      16) HK,HongKong,HKG1   22) JP,Osaka,OSA1     28) US,Dallas,DFW1    34) US,Washington,IAD2
5) AU,Sydney,SYD1      11) DE,Frankfurt,FRA1   17) IL,TelAviv,TLV1    23) KR,Seoul,ICN1     29) US,LosAngeles,LAX1  35) ZA,Johannesburg,JNB1
6) BR,SaoPaulo,SAO1    12) ES,Madrid,MAD1      18) IN,Chennai,MAA1    24) SE,Stockholm,STO1  30) US,Miami,MIA1      36) Not in the list? Input Manually
Enter your choice: 1
```

→ Select your Secondary Node

```
Please, select your Secondary Node (Country/City/NodeID)
1) AE,Dubai,DXB1      7) CA,Montreal,YMQ1      13) FR,Paris,PAR1      19) IN,Delhi,DEL1      25) SG,Singapore,SIN1      31) US,NewYork,NYC1      37) Quit
2) AR,BuenosAires,BUE1  8) CA,Vancouver,YVR1    14) GB,London,LON1    20) IN,Mumbai,BOM1    26) US,Atlanta,ATL1    32) US,Phoenix,PHX1
3) AT,Vienna,VIE1      9) CL,Santiago,SCL1     15) GB,Manchester,MAN1  21) IT,Milan,MIL1     27) US,Chicago,ORD1    33) US,Seattle,SEA1
4) AU,Melbourne,MEL1   10) CO,Bogota,BOG1      16) HK,HongKong,HKG1   22) JP,Osaka,OSA1     28) US,Dallas,DFW1    34) US,Washington,IAD2
5) AU,Sydney,SYD1      11) DE,Frankfurt,FRA1   17) IL,TelAviv,TLV1    23) KR,Seoul,ICN1     29) US,LosAngeles,LAX1  35) ZA,Johannesburg,JNB1
6) BR,SaoPaulo,SAO1    12) ES,Madrid,MAD1      18) IN,Chennai,MAA1    24) SE,Stockholm,STO1  30) US,Miami,MIA1      36) Not in the list? Input Manually
Enter your choice: 1
```

Note: If required, you can add the Node manually using the option "Not in the list? Input Manually."

→ (optional) Change the Tunnel Names.

```
-----
NETSKOPE INFORMATION
Tunnels Name:
      NStun1: csc-mux-4-az-1.nstun1
      NStun2: csc-mux-4-az-1.nstun2
      NStun3: csc-mux-4-az-1.nstun3
      NStun4: csc-mux-4-az-1.nstun4
-----
Do you want to change the Tunnels Name?
Note: The CSC will append 'nstun<number>' to the <Name> value. The format will be <Name>.nstun<number>. By default, the value of Name is the VM Name.
1) Yes
2) No
Enter your choice: 2
```

→ (optional) Regenerate the PreShare Key of all tunnels

```
-----
Do you want to regenerate the PreShare Key of all tunnels?
1) Yes
2) No
Enter your choice: 2
```

DNS Configuration

```
-----
DNS Configuration
You are using Azure DNS server 168.63.129.16 and Google DNS server 8.8.8.8
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-bypass-vm.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.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: 10.2.3.4
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: █
```

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

```
Please confirm these values:
-----
Primary Tunnel:
Node : US,Washington,IAD2
Node Public IP: 163.116.146.38 is Alive
Node Probe: 10.146.6.216
Secondary Tunnel:
Node : US,NewYork, NYC1
Node Public IP: 163.116.135.38 is Alive
Node Probe: 10.135.6.216
-----
CSV file:
-----
tunnel name,source identity,source ip address,primary pop,failover pop,encryption cipher,psk,maximum bandwidth,enabled
csc-mux-4-az-1.nstun1,20.127.156.167,20.127.156.167,IAD2,NYC1,AES128-CBC,wGUN0R0QlqtF3l0vS8140FanXG3UicF,250,true
csc-mux-4-az-1.nstun2,20.127.156.168,20.127.156.168,IAD2,NYC1,AES128-CBC,WYA6c1okIP7RS2bxpypEd0QV2g0d5NS,250,true
csc-mux-4-az-1.nstun3,20.127.156.169,20.127.156.169,IAD2,NYC1,AES128-CBC,1o0P6KSY8cuk6x5dNMg1bnM4u8JyFWZU,250,true
csc-mux-4-az-1.nstun4,20.127.156.166,20.127.156.166,IAD2,NYC1,AES128-CBC,Q5r3RJvgBXXUwnigXPMtUCwpAhk0PDgu,250,true
-----
Instructions to Import the CSV file:
1 - Copy and paste the CSV file's contents on a Text Editor and save it as '<filename>.csv'. Important: Do not add blank lines at the end of the file.
2 - On your Netskope console, please go to page: Settings -> Security Cloud Platform -> Traffic Steering Section -> IPSEC and click 'IMPORT TUNNELS FROM CSV' and select the CSV file.
-----
DNS Servers: Azure DNS server 168.63.129.16 and Google DNS server 8.8.8.8
-----
Bypass Proxy PAC URL
Your current Proxy Bypass PAC URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-bypass-vm.pac
-----
Routed Bypass URL
Your current Routed Bypass URL is https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json
-----
Primary Syslog / SIEM server IP: 10.2.3.4
Syslog / SIEM TCP port IP: 514
-----
Do you want to implement these values? (The CSC will reboot)
1) Yes
2) No
Enter your choice: █
```

Done!

7.6.2 Switch Tunnels - Primary / Secondary.

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

```
Selection: 15

NETSKOPE INFORMATION
Primary Tunnel:
    Node : US,Washington,IAD2
    Node Public IP: 163.116.146.38 is Alive
    Node Probe: 10.146.6.216
Secondary Tunnel:
    Node : US,NewYork,NYC1
    Node Public IP: 163.116.135.38 is Alive
    Node Probe: 10.135.6.216
-----
Do you want to Switch Primary / Secondary Tunnel?
Selecting Yes will disrupt all current connections.
1) Yes
2) No
Enter your choice: 1

IPSEC tunnels configured with Primary Node: 163.116.135.38 (US,NewYork,NYC1), Secondary Node: 163.116.146.38 (US,Washington,IAD2).
Tunnels switched via Console on: Thu 21 Apr 12:04:48 UTC 2022
```

7.6.3 Update Netskope Nodes Databases.

This command retrieves the latest Netskope Node Database.

```
Selection: 16

Checking Netskope Nodes Databases...
This CSC has the latest version: 1.3
```

7.6.4 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 Next-Hop on Routes via CSC HA Pair.

-----
How to configure:

1) 'Deployment': Deploy a pair of CSCs with the following conditions:
  1.1) There is connectivity each other via their internal interfaces.
  1.2) (Optional, but Recommended) Deploy the CSCs on Availability Zones or Availability Sets.
2) 'Identity': On each CSC VM
  2.1) Go to 'Identity -> System Assigned' and 'Turn ON' status. (and Save).
  2.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:
      -> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>
      -> Role: Contributor, Resource Group: <Route Tables Resource Group/s>
      -> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>
3) 'Routes'
  3.1) Go to Routes (inside the Route Table) and create the Routes that the CSC HA group will control:
      -> Route name: <any name you want>
      -> Address prefix: <Subnet/Mask>
          Examples: 0.0.0.0/0 (if you want to send all traffic via Netskope) or 163.116.128.80/32 (when using PAC files and/or Explicit Proxy)
      -> Next hop type: Virtual Appliance
      -> Next hop address: <Input CSC-GW-IP (eth1, first IP) of any CSC>
  3.2) Go to Subnets and associate the Subnet with the Route Table.
  3.3) Repeat the process if you want to add more Routes. The CSC HA functionality can manipulate multiple Routes.
4) Obtain the following values and Run the Wizard.
  4.1) Route, Route Table, Resource Group.
  4.2) Computer Name and Resource Group of each CSC.
5) This Wizard will create a JSON file. You can use this JSON file to configure the Other CSC in the pair.

How it works:

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

-----

The HA service is NOT Active

Do you want to configure it?

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

Help provided:

How to configure:

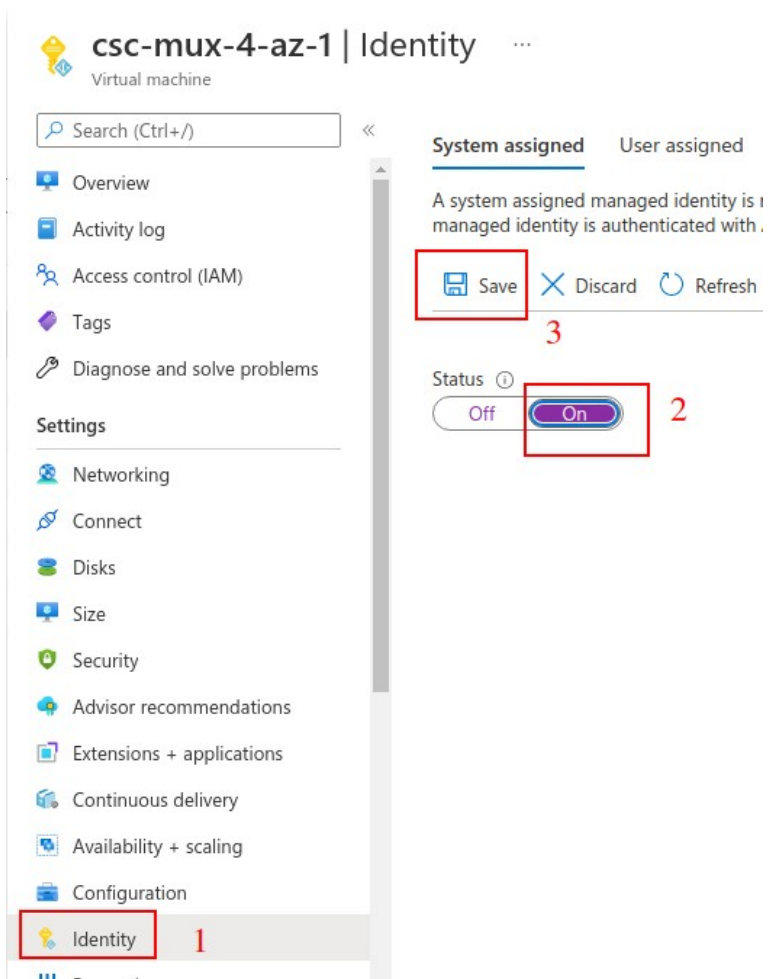
1) 'Deployment': Deploy a pair of CSCs with the following conditions:

- 1.1) There is connectivity each other via their internal interfaces.
- 1.2) (Optional, but Recommended) Deploy the CSCs on Availability Zones or Availability Sets.

2) 'Identity': On each CSC VM

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

Example:

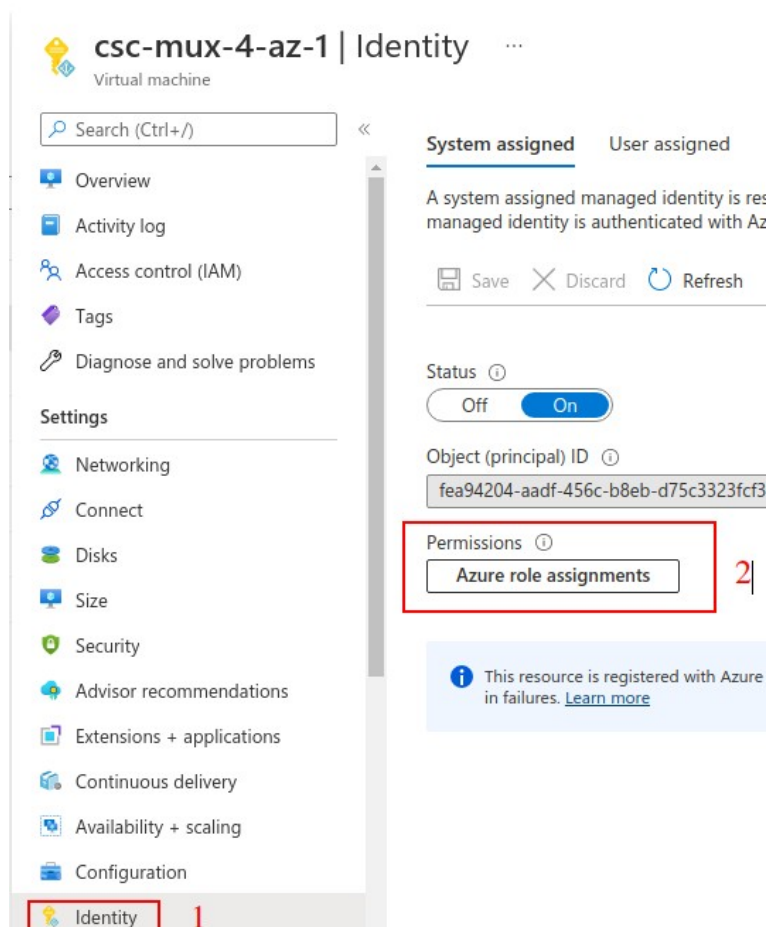


Note: Repeat the same step on the other CSC on the HA Pair.

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

- > Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>
- > Role: Contributor, Resource Group: <Route Tables Resource Group/s>
- > Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>

Example:



Home > csc-mux-4-az-1 >




Azure role assignments ...

+ Add role assignment (Preview) Refresh

If this identity has role assignments that you don't have permission to read, they won't be shown in the list. [Learn more](#)

Subscription *

MHB

Role	Resource Name	Resource Type	Assigned To
Contributor	 CSC-East-US	Resource Group	csc-mux-4-az-1
Contributor	 RouteTables-East-US	Resource Group	csc-mux-4-az-1
Network Contributor	 Networks-East-US	Resource Group	csc-mux-4-az-1

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

3) 'Routes'

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

-> Route name: <any name you want>

-> Address prefix: <Subnet/Mask>

Examples: 0.0.0.0/0 (if you want to send all traffic via Netskope) or 163.116.128.80/32 (when using PAC files and/or Explicit Proxy)

-> Next hop type: Virtual Appliance

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

Example:

Servers-Route-Table Route table

Search (Ctrl+/) << → Move ▾ Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Configuration

Routes

Subnets

Properties

Locks

Monitoring

Alerts

Automation

Tasks (preview)

Export template

Essentials

Resource group (move) : [RouteTables-East-US](#) Associations : 2 subnet associations

Location : East US

Subscription (move) : [MHB](#)

Subscription ID : ffde02fb-c38f-45fb-9e31-89e5303be5f1

Tags (edit) : [Click here to add tags](#)

Routes

Search routes

Name	Address prefix	Next hop type	Next hop IP address
Net-10-2-9-0	10.2.9.0/24	Virtual appliance	10.2.9.4
Server-default-route	0.0.0.0/0	Virtual appliance	10.2.2.10
to-163.116.128.80	163.116.128.80/32	Virtual appliance	10.2.2.10

Subnets

Search subnets

Name	Address range	Virtual network	Security group
servers-East-US	10.2.3.0/24	VNET-East-US	-
csc-internal-East-US	10.2.2.0/24	VNET-East-US	-

CSC GW IP

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

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

4) Obtain the following values and Run the Wizard.

4.1) Route, Route Table, Resource Group.

4.2) Computer Name and Resource Group of each CSC.

Example:

First CSC on the HA Pair – Manual Configuration:

```

The HA service is NOT Active
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: 1

Identity Type: SystemAssigned
Please, input the Route/s values:
Route Name= Server-default-route
Route Table= Servers-Route-Table
Resource Group= RouteTables-East-US
Route
Do you want to add another Route? (y/n)? y
Route Name= to-163.116.128.80
Route Table= Servers-Route-Table
Resource Group= RouteTables-East-US
Route
Do you want to add another Route? (y/n)? n
Please, input values of other CSC in the pair
Computer Name= csc-mux-4-az-2
Resource Group= CSC-East-US
Computer Name Other CSC
Values to configure are:
Route/s (Qty)=2
Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
Route 2: to-163.116.128.80 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
Computer Name of other CSC in the pair: csc-mux-4-az-2 (Resource Group=CSC-East-US)
Do you want to apply changes? (y/n)? y

```

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

```

Do you want to apply changes? (y/n)? y
Please, copy the following values in a safe place to configure the other CSC on the High Availability Pair.

High Availability JSON file:

{
  "model": "csc-mux-ns-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "SystemAssigned",
    "haFirstCsc": {
      "vmName": "csc-mux-4-az-1",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "20.127.156.184"
    },
    "haSecondCsc": {
      "vmName": "csc-mux-4-az-2",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "52.249.220.21"
    },
    "haPrivateAccessPublicIp": "20.127.156.184",
    "haRoutes": [
      {
        "routeName": "Server-default-route",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      },
      {
        "routeName": "to-163.116.128.80",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      }
    ]
  }
}

CSC HA is : active (running) since Thu 2022-04-21 12:54:39 UTC; 11ms ago

```

Example:

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

```

The HA service is NOT Active
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: {
  "model": "csc-mux-ns-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "SystemAssigned",
    "haFirstCsc": {
      "vmName": "csc-mux-4-az-1",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "20.127.156.184"
    },
    "haSecondCsc": {
      "vmName": "csc-mux-4-az-2",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "52.249.220.21"
    },
    "haPrivateAccessPublicIp": "20.127.156.184",
    "haRoutes": [
      {
        "routeName": "Server-default-route",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      },
      {
        "routeName": "to-163.116.128.80",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      }
    ]
  }
}

(MHB-CSC)(INFO) High Availability JSON file (highAvailability.json) integrity is OK
(MHB-CSC)(INFO) High Availability: IAM Identity in use: SystemAssigned
(MHB-CSC)(INFO) High Availability: Route Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
(MHB-CSC)(INFO) High Availability: Route to-163.116.128.80 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
(MHB-CSC)(INFO) High Availability is active (running) since Thu 2022-04-21 12:59:57 UTC; 9ms ago.

```

How it works:

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

Check HA using Show Configuration and Status:

```
HIGH AVAILABILITY Information
The HA service is: active (running) since Thu 2022-04-21 12:54:39 UTC; 11h ago
Identity Type: SystemAssigned
Route to Netskope using Next Hop: 10.2.2.10 of VM: csc-mux-4-az-1 (this CSC)
Current values configured are:
  Route/s (Qty)= 2
    Route 1: Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
    Route 2: to-163.116.128.80 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
  Computer Name of other CSC in the pair: csc-mux-4-az-2 (Resource Group=CSC-East-US)
Private Access Public IP= 20.127.156.184
```

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

Logs generated by High Availability:

```
Apr 21 12:56:20 csc-mux-4-az-1 root: (MHB-CSC)(INFO) Route to Netskope using Next Hop: 10.2.2.10 of CSC: csc-mux-4-az-1
Apr 21 12:59:37 csc-mux-4-az-2 cscadmin: (MHB-CSC)(INFO) High Availability JSON file (highAvailability.json) integrity is OK
Apr 21 12:59:50 csc-mux-4-az-2 cscadmin: (MHB-CSC)(INFO) High Availability: IAM Identity in use: SystemAssigned
Apr 21 12:59:56 csc-mux-4-az-2 cscadmin: (MHB-CSC)(INFO) High Availability: Route Server-default-route (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
Apr 21 12:59:56 csc-mux-4-az-2 cscadmin: (MHB-CSC)(INFO) High Availability: Route to-163.116.128.80 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US) configured.
Apr 21 12:59:57 csc-mux-4-az-2 cscadmin: (MHB-CSC)(INFO) High Availability is active (running) since Thu 2022-04-21 12:59:57 UTC; 9ms ago.
Apr 21 13:01:39 csc-mux-4-az-2 root: (MHB-CSC)(INFO) Route to Netskope using Next Hop: 10.2.2.10 of CSC: csc-mux-4-az-1
```

8 Steering traffic to NewEdge with the CSC Mux for Azure.

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

When connecting Virtual Machines, Virtual Desktops, etc., to Netskope using the CSC Mux, you have two options of steering traffic: routing and proxying.

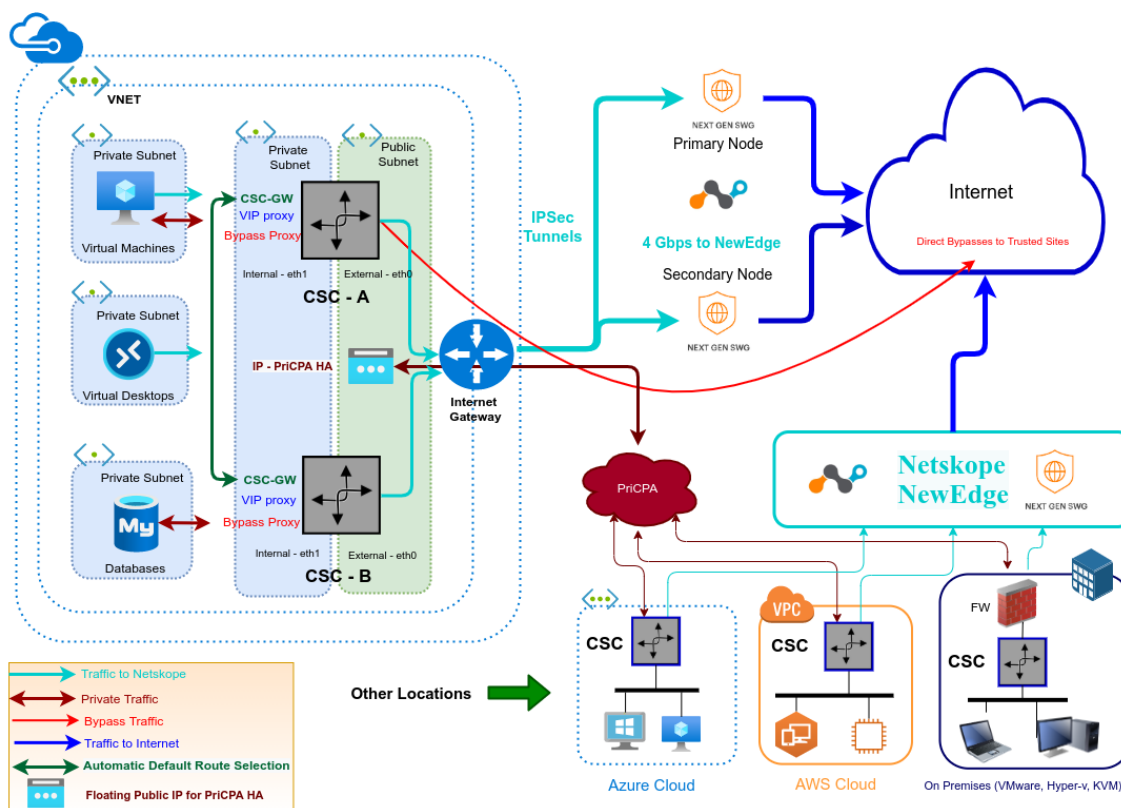
The options are not mutually exclusive. You can use both at the same time. Moreover, when the CSC Mux is on a HA pair, you can use both simultaneously, duplicating the capacity for Web Traffic to 2 Gbps (CSC Mux 4) or 4 Gbps (CSC Mux 8).

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 Mux is on High Availability and when using a single CSC Mux.

8.1 CSC on HA Pair

8.1.1 Network Diagram



8.1.2 Prerequisites

1. Deploy 2 x CSC Mux as HA Pair. (See section "7.6.4 High Availability configuration")
2. Create the Routed Bypasses. (See section "7.4 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 "7.3 Proxy Bypass")

8.1.3 Routing traffic via the CSC HA pair.

8.1.3.1 *Traffic to Netskope*

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

8.1.3.2 *"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.

8.1.4 Proxy traffic via the CSC HA Pair.

8.1.4.1 *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 (CSC Mux 4) or 4 Gbps (CSC Mux 8) 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.

8.1.4.1.1 **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) || isIPNet(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 (isIPNet(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 (isIPNet(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  
}
```

8.1.4.1.2 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) with "Next-Hop": 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 = /^([0-9]{1,3}\.){4}[0-9]{1,3}$/;  
    var resolved_ip = dnsResolve(host);  
  
    /* Don't send non-FQDN or private IP auths to us */  
    if (isPlainHostName(host) || isIPNet(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  
}
```

8.1.4.1.3 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
}
```

8.1.4.2 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
```

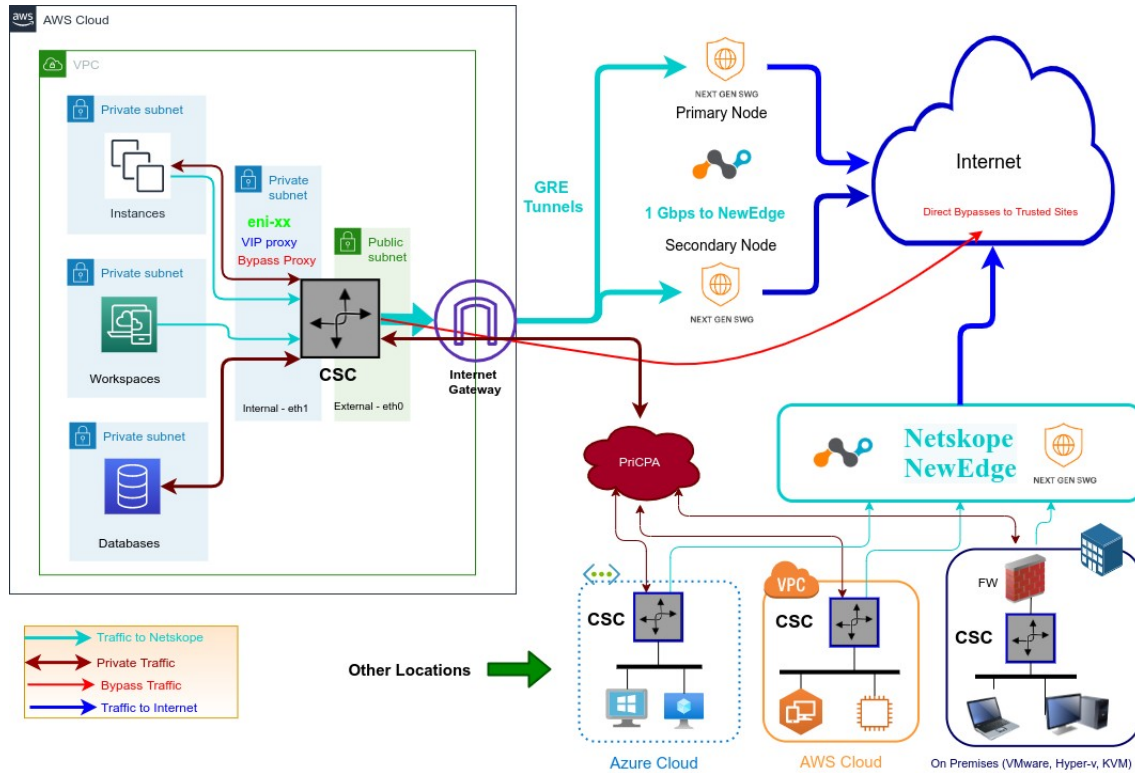
Test	Return	Observations
curl http://www.netskope.com	163.116.146.120 Ashburn, United States (IAD2)	OK. http via Netskope.
curl https://ip.maidenheadbridge.com	163.116.146.120	OK. https via Netskope
curl http://portquiz.net	Port 80 test successful! Your IP: 54.80.198.195	OK. portquiz.net via bypass using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

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

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

8.2 CSC Single

8.2.1 Network Diagram



8.2.2 Prerequisites

1. Obtain the CSC GW IP 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 "7.4 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 "7.3 Proxy Bypass")

8.2.3 Routing traffic via the CSC Single.

8.2.3.1 *Traffic to Netskope*

On your Route/s, configure the default route to the Internet (0.0.0.0/0) with "Next-Hop" the CSC-GW-IP of the CSC.

8.2.3.2 *"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.

8.2.4 Proxy traffic via the CSC Single.

8.2.4.1 *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

8.2.4.1.1 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) with "Next-Hop": 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 = /^([0-9]{1,3}\.){4}[0-9]{1,3}$/;  
    var resolved_ip = dnsResolve(host);  
  
    /* Don't send non-FQDN or private IP auths to us */  
    if (isPlainHostName(host) || isIPNet(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  
}
```

8.2.4.1.2 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) || isNetNet(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  
}
```

8.2.4.2 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 two 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_proxy ⁶		
export ⁷ 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.netskope.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.

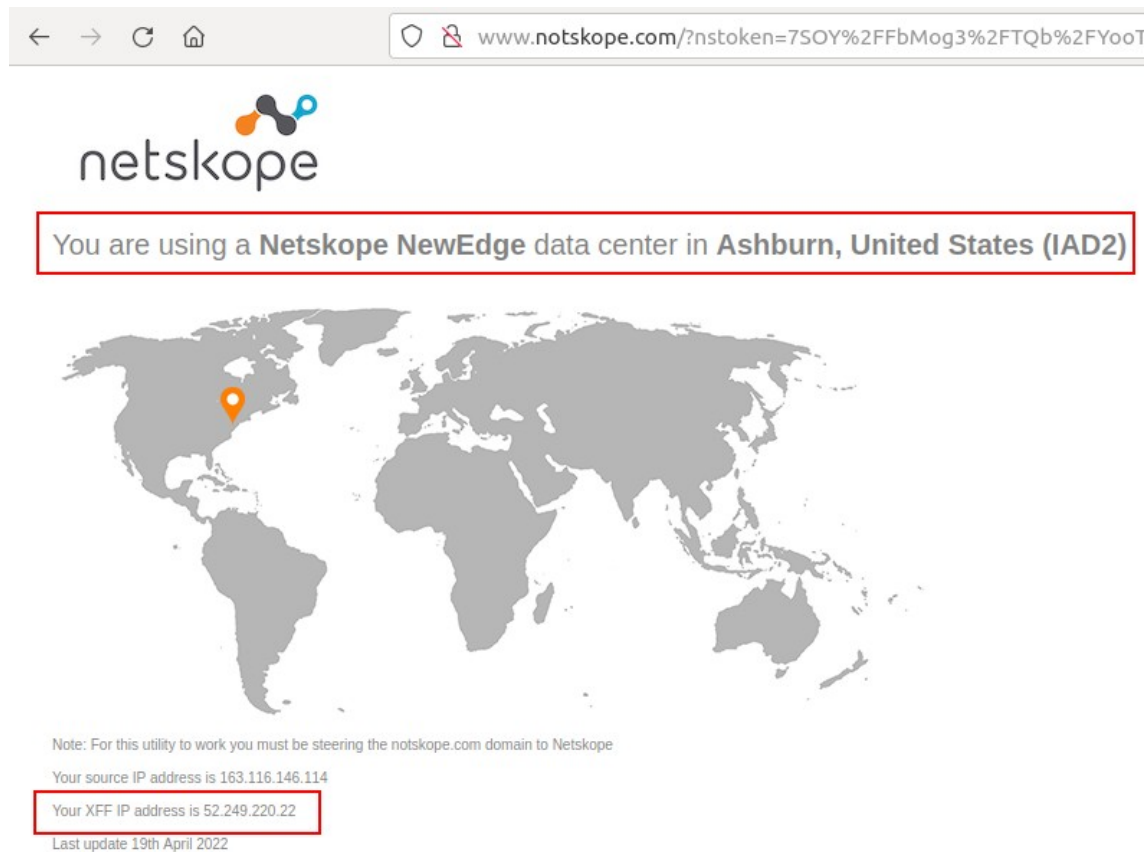
⁷ Use command `$unset <variable name>` to clear the values.

8.3 Testing traffic to Netskope

8.3.1 www.netskope.com

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

Using the browser:



Using curl command from CMD or Terminal

Proxy environment:

Command	<code>curl --proxy http://<CSC VIP>:80 www.netskope.com</code> (i.e. <code>\$curl --proxy http://10.2.2.14:80 www.netskope.com</code>)
Expected Result	<NewEdge Node IP> <City>, <Country> (<NodeID>) (i.e. 163.116.146.114 Ashburn, United States (IAD2))

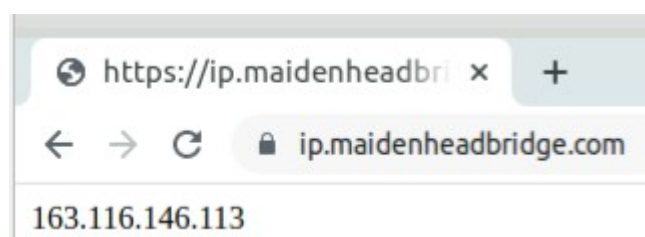
Routed environment:

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

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

Routed environment:

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

8.3.3 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)
```

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

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

Aggregated Bandwidth Download: 913.35 Mbps
```

Note: At the moment of writing this documentation, Netskope provides 250 Mbps per IPsec tunnel. The CSC Mux 4 can aggregate 4 x IPsec tunnels (~ 1 Gbps total), and the CSC Mux 8 can aggregate 8 x IPsec tunnels (~ 2 Gbps Total).

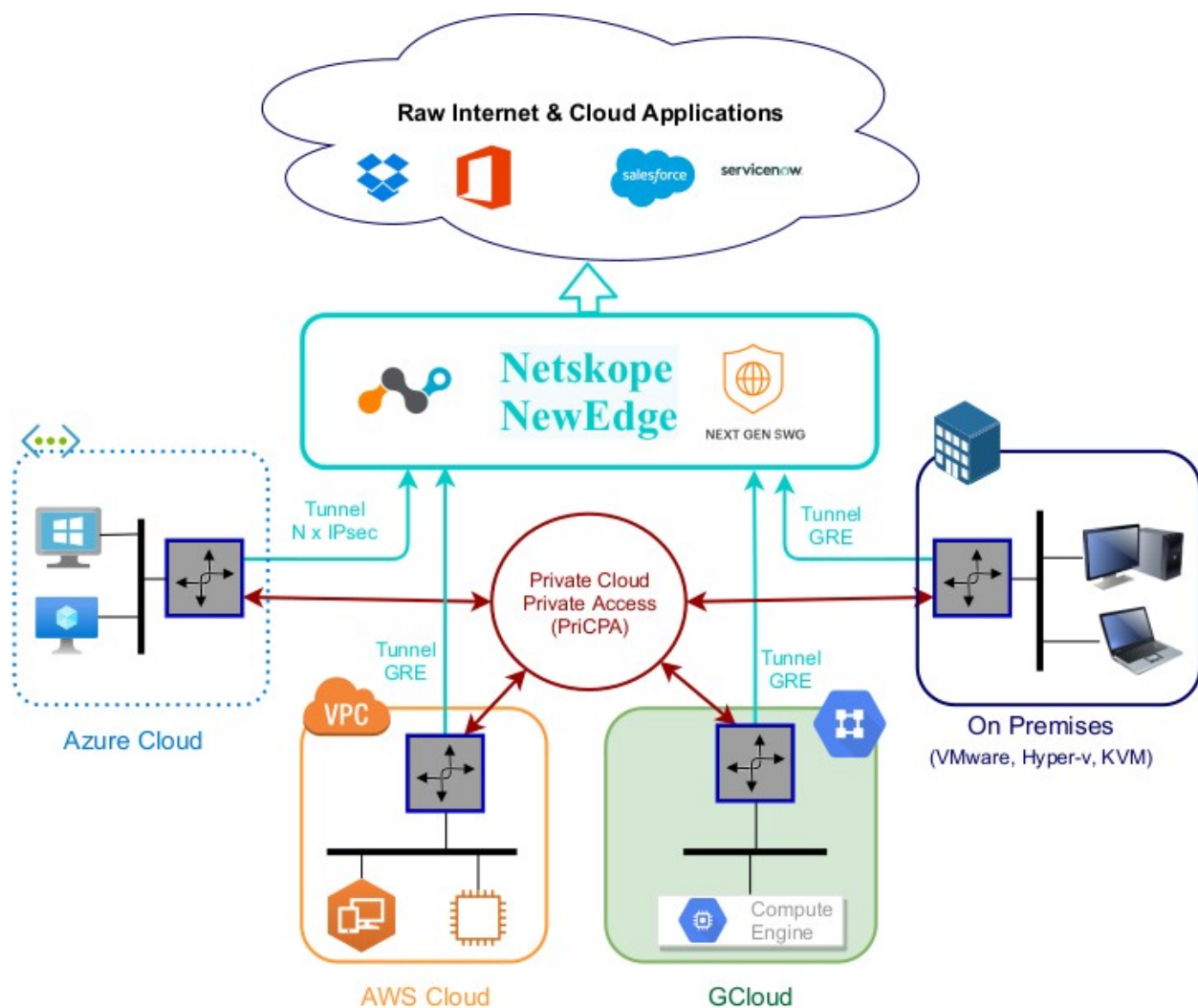
9 Private Cloud Private Access

9.1 What is Private Cloud Private Access (PriCPA)?

Private Cloud Private Access (PriCPA) is a new functionality of the Cloud Security Connector. PriCPA allows you to create a Private Cloud among all CSCs for private traffic. In 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.

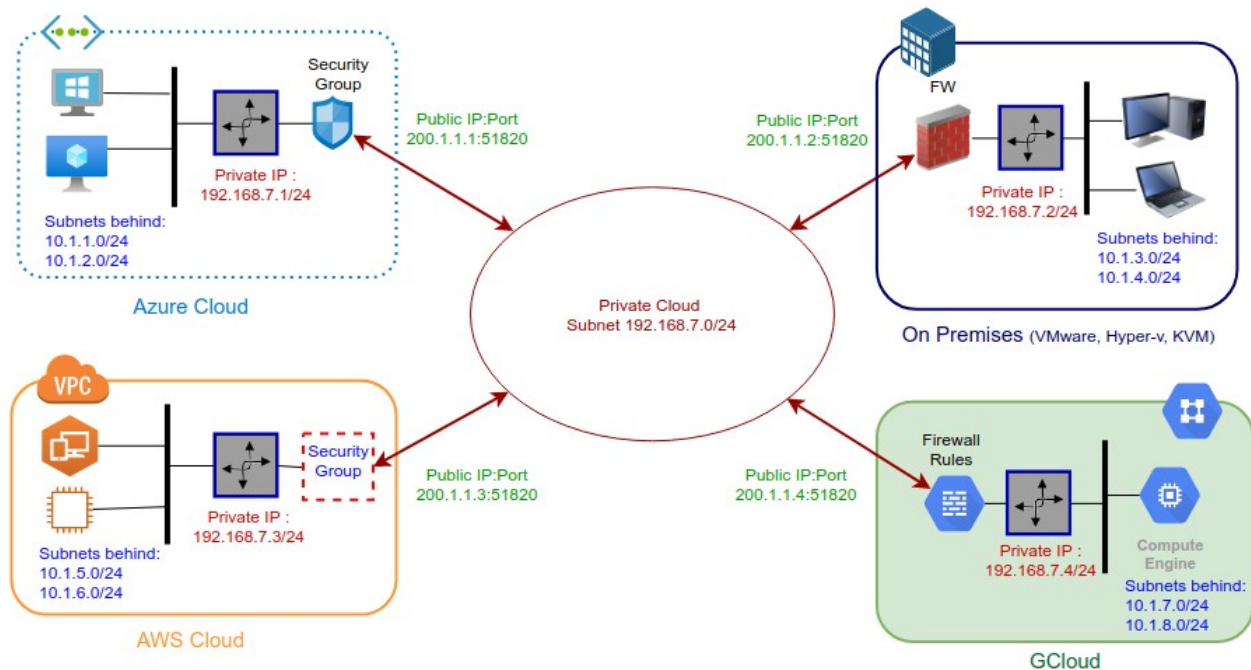
9.2 PriCPA Network Diagrams

9.2.1 High Level Network Diagram



9.2.2 Low Level Network Diagram – PriCPA only

The following network diagram shows the IP addressing for PriCPA.



Steps to design your Private Cloud:

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

9.3 Configuring PriCPA

The Main Menu has a section for Private Access:

```
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:

```
Selection: 19
Private Access Configuration Wizard

Steps to configure Private Access:

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

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

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

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

9.3.1 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:

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

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

→ Select "1) 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.

```

Do you want to enable Private Access?
1) Manual Configuration
2) Token and JSON
3) Quit
Enter your choice: 1

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

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

Please, input the following values:
Node Name (string): csc-mux-4-az
(Optional) Location Name (string): Azure US East
(Optional) Description (string): CSC Mux 4 (HA Pair) on RG: CSC-East-US
Public IP and UDP port (IP:Port): 20.127.156.184:51820
Private IP/Subnet of Local Interface (IP/Subnet Prefix): 192.168.7.39/24

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

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

Do you want to change default value of 'Persistent KeepAlive = no'?
1) Yes
2) No (Recommended)
Enter your choice: 2

The values to configure are:
Node Name: csc-mux-4-az
Public IP and UDP Port: 20.127.156.184:51820
Private IP/Subnet of Local Interface: 192.168.7.39/24
Location Name: Azure US East
Description: CSC Mux 4 (HA Pair) on RG: CSC-East-US
Persistent KeepAlive: no

```

➤ Apply values

```

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

Private Access - Private Access service is enabled on csc-mux-4-az-1.

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

Token: aUxTUzU0C9aNNFJwLZPbkrUOG9DRUVnNEduMwPMjBJTUhVwU5SdUVV0D8K

Private Access Local Config JSON file:
{
  "peers": [
    {
      "nodeName": "csc-mux-4-az",
      "location": "Azure US East",
      "description": "CSC Mux 4 (HA Pair) on RG: CSC-East-US",
      "publicKey": "IkwC5FkuSTLq0yZf9Kh23GU15Hkzy5jViy1biWbuqxg=",
      "publicIpAndUdpPort": "20.127.156.184:51820",
      "privateIpAndUdpPort": "192.168.7.39/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}

```

IMPORTANT: Keep this values on 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.

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

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

Go to 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

Please, input the following values:

Token (string): aUxTUzU0OC9aNNfJwLZPbKRUG9DRUVnNEduMwPMjBJTUhVWU5SdUUVV0D0K

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-mux-4-az",
      "location": "Azure US East",
      "description": "CSC Mux 4 (HA Pair) on RG: CSC-East-US",
      "publicKey": "IkwC5FkuSTLq0yZf9Kh23GU15Hkzy5jViylbiWbuqxg=",
      "publicIpAndUdpPort": "20.127.156.184:51820",
      "privateCidrIp": "192.168.7.39/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}

Private Access Local Config JSON file imported successfully

The values to configure are:

Node Name: "csc-mux-4-az"
Public IP and UDP Port: 20.127.156.184:51820
Private IP/Subnet of Local Interface: 192.168.7.39/24
Location Name: "Azure US East"
Description: "CSC Mux 4 (HA Pair) on RG: CSC-East-US"
Persistent KeepAlive: no

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1

Private Access - Private Access service is enabled on csc-mux-4-az-2.
```

9.3.3 Create the Private Access Peers JSON file

The Private Access Peers JSON file contains:

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

Here some examples.

9.3.3.1 *Full mesh Private Access Peers JSON file*

Consider the following example:

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

The Local Config JSON file of each node is:

ns-cgc00001

```
{
  "peers": [
    {
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+lXXJte14tji3zIMNq+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": "TrMvSoP4jYQlY6RlZBgbssQqY3vxl2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

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

Basic Peers Configuration JSON file

```
{
  "peers": [
    {
      "nodeName": "ns-cgc00001",
      "description": "Node on VMware Server 1",
      "location": "HQ",
      "publicKey": "yAnz5TF+IXXJte14tjj3zlMNq+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+y71lOWWXX0=",
  "publicIpAndUdpPort": "200.1.1.3:51821",
  "privateCirdIp": "192.168.7.3/24",
  "persistentKeepAlive": "no",
  "networks": [
    "10.3.1.0/24",
    "10.3.2.0/24"
  ],
  "privateApps": []
}
]
}

```

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

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

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

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

```

ns-cgc00001
{
  "nodeName": "ns-cgc00001",
  "description": "Node on VMware Server 1",
  "location": "HQ",
  "publicKey": "yAnz5TF+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+IXXJte14tj3zIMNq+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": ""
      }
    }
  ],
  "node": {
    "nodeName": "ns-cgc00003",
    "description": "Node on VMware Server 3",
    "location": "Branch",
    "publicKey": "TrMvSoP4jYQlY6RlZBgbssQqY3vxl2Pi+y71IOWWXX0=",
    "publicIpAndUdpPort": "200.1.1.3:51821",
    "privateCirdIp": "192.168.7.3/24",
    "persistentKeepAlive": "no",
    "networks": [
      "10.3.1.0/24",
      "10.3.2.0/24"
    ],
    "privateApps": [
      {
        "description": "Allow all traffic to this site",
        "ipProtocol": "all",
        "sourceCirdIp": [
          "0.0.0.0/0"
        ],
        "destinationCirdIp": [
          "10.3.1.0/24",
          "10.3.2.0/24"
        ],
        "destinationSinglePorts": [
          ""
        ],
        "destinationPortRange": {
          "fromPort": "",
          "toPort": ""
        }
      }
    ]
  }
}

```

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

9.3.3.2 Understanding "privateApps" configuration and values

Question 1: Where to configure the "privateApps"?

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

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

Question 2 : What about the values to configure?

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

Accepts single value only -> ""

Accepts single or multiple values -> []

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

Examples:

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	"destinationCirdIp": ["10.1.1.100/32", "10.1.1.200/32"

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.

		172.16.0.0/12 192.168.0.0/16],
"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" }

9.3.3.3 Example of "privateApps" for a Windows Domain controller

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

The port information was taken from this article:

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

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

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

9.3.3.4 Example of "privateApps" for Internal Web Server.

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

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

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

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

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

There are three methods available:

1. URL: (Recommended) Using "Private Access Peers URL" and running the command "Refresh Private Access Peers URL" using AWS Systems Manager 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.

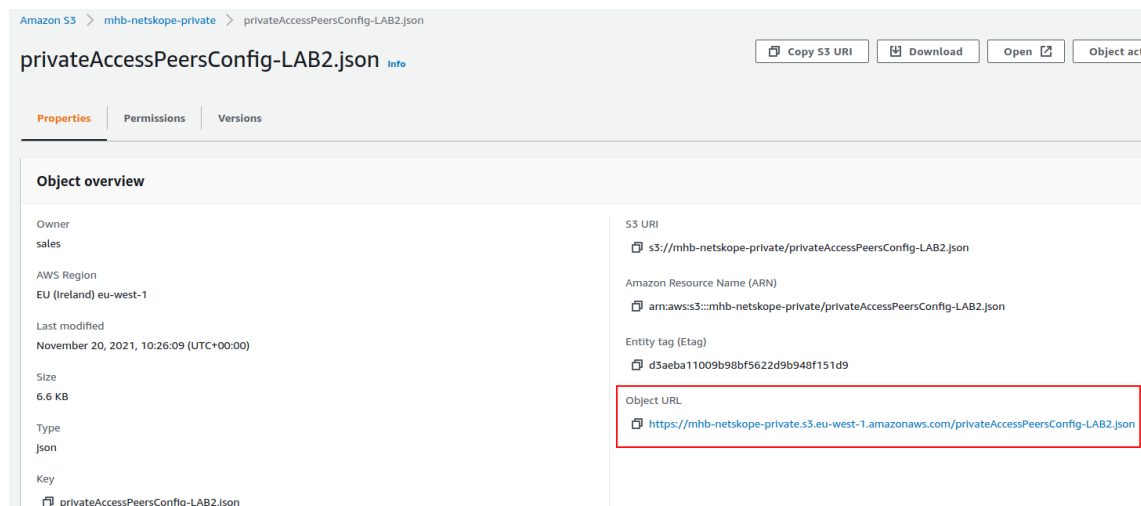
In this section we are going to explain two methods: URL and Manual Copy. The DevOps method is explained on Section12: DevOps operations.

9.3.4.1 Using "Private Access Peers URL"

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

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

Example of AWS bucket:



2. Configure the URL on each CSC.

Ssh the each CSC and go to Main Menu -> 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

Private Access Configuration Wizard

Steps to configure Private Access:

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

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

```

```

Private Access is enabled.

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

Your Private Acces Peers URL configured is: null

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

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

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

Private Access Peers JSON file imported successfully

```

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

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

Private Apps:
Index: 0, NodeName: csc-mux-4-az, Location: Azure US East, publicIpAndUdpPort: 20.127.156.184:51820, privateCirdIp: 192.168.7.39/24, Private Apps Qty: 1
Index: 1, NodeName: ns-csc-gre-v-1-0e, Location: AWS US, publicIpAndUdpPort: 18.213.189.84:51820, privateCirdIp: 192.168.7.37/24, Private Apps Qty: 4
Index: 2, 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: 3, NodeName: ns-cgc00004, Location: MHB-DC-KVM, publicIpAndUdpPort: 82.68.6.74:51821, privateCirdIp: 192.168.7.11/24, Private Apps Qty: 6
Index: 4, NodeName: ns-cgc00006, Location: MHB-BH-DC, publicIpAndUdpPort: 217.155.196.81:51820, privateCirdIp: 192.168.7.20/24, Private Apps Qty: 2

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

Creating Private Apps:
Private Access - (Index: 0, Node: csc-mux-4-az) Private App 'Allow all to Azure' was created successfully.
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'AWS - SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'AWS - icmp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf udp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 2, 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: 2, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 2, Node: ns-csc-gre-aws-v-0-4) 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 'Domain Controllers TCP' was created successfully. (destinationSinglePorts)
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. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Syslog ICMP' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Syslog tcp port' was created successfully. (destinationSinglePorts)
Private Access - (Index: 4, Node: ns-cgc00006) Private App 'BH - SSH to Servers' was created successfully. (destinationSinglePorts)
Private Access - (Index: 4, Node: ns-cgc00006) Private App 'BH - SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)

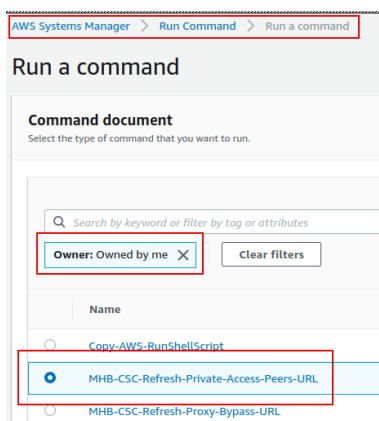
Adding Peers:
Private Access - Node: ns-csc-gre-v-1-0e added successfully.
Private Access - Node: ns-csc-gre-aws-v-0-4 added successfully.
Private Access - Node: ns-cgc00004 added successfully.
Private Access - Node: ns-cgc00006 added successfully.

Changes Security Group External:
Private Access - Inbound Port Rules 'mhb-csc-private-access-151820' added to Security Group 'csc-mux-4-az-eth0-NSG-1'
Private Access - Outbound Port Rules 'mhb-csc-private-access-051820, mhb-csc-private-access-051821' added to Security Group 'csc-mux-4-az-eth0-NSG-1'
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"



- 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-0f3837028ad9fcd8 X mi-0b9178c22b03ce2bf X mi-0e234f4278cd74e27 X mi-0beef6eaa71c2f0bf X

Instances

Search:

Ping status: Online X Clear filters

	Name	Instance ID	Instance state	Availability zone	Ping status
<input checked="" type="checkbox"/>	ns-cgc00006-b	mi-0f3837028ad9fcd8	Running	-	Online
<input checked="" type="checkbox"/>	ns-cgc00004-b	mi-0b9178c22b03ce2bf	Running	-	Online
<input checked="" type="checkbox"/>	ns-cgc00005-a	mi-0e234f4278cd74e27	Running	-	Online
<input checked="" type="checkbox"/>	ns-cgc00004-a	mi-0beef6eaa71c2f0bf	Running	-	Online
<input checked="" type="checkbox"/>	ns-cgc00006-a	mi-08c465d750d2689ae	Running	-	Online
<input checked="" type="checkbox"/>	ns-cgc00005-b	mi-0650bce2872f405c0	Running	-	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

Search:

	Instance ID	Instance name	Status	Detailed Status
<input type="radio"/>	mi-0650bce2872f405c0	ns-cgc00005-b	Success	Success
<input type="radio"/>	mi-08c465d750d2689ae	ns-cgc00006-a	Success	Success
<input type="radio"/>	mi-0beef6eaa71c2f0bf	ns-cgc00004-a	Success	Success
<input type="radio"/>	mi-0e234f4278cd74e27	ns-cgc00005-a	Success	Success
<input type="radio"/>	mi-0b9178c22b03ce2bf	ns-cgc00004-b	Success	Success
<input type="radio"/>	mi-0f3837028ad9fcd8	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"

AWS Systems Manager > Run Command > Command ID: e7c8bfa2-e045-4df0-8216-4721be8d4249 > Output on: mi-0650bce2872f405c0

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

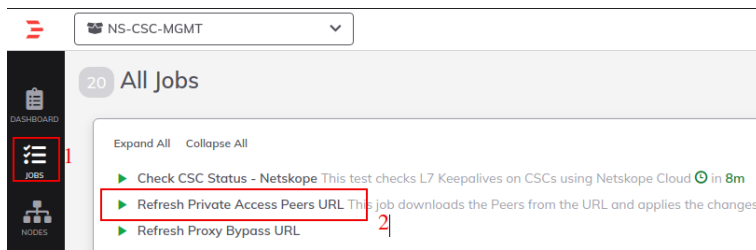
The command output displays a maximum of 48,000 characters. You can view the complete command output in either Amazon S3 or CloudWatch Logs, if:

```
Private Access - Private Access Peers JSON file imported successfully.

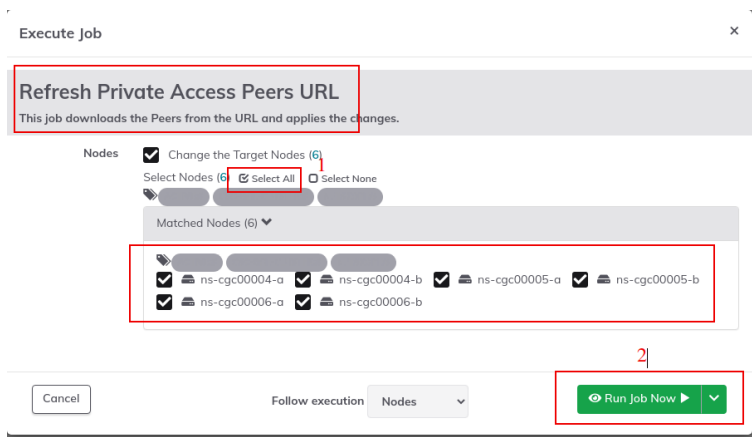
Creating Private Apps
Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Intranet Server' was created successfully.
(destinationSinglePorts)
Private Access - (Index: 0, Node: ns-cgc00004) Private App 'Domain Controllers TCP' was created successfully.
(destinationSinglePorts)
```

Using Rundeck

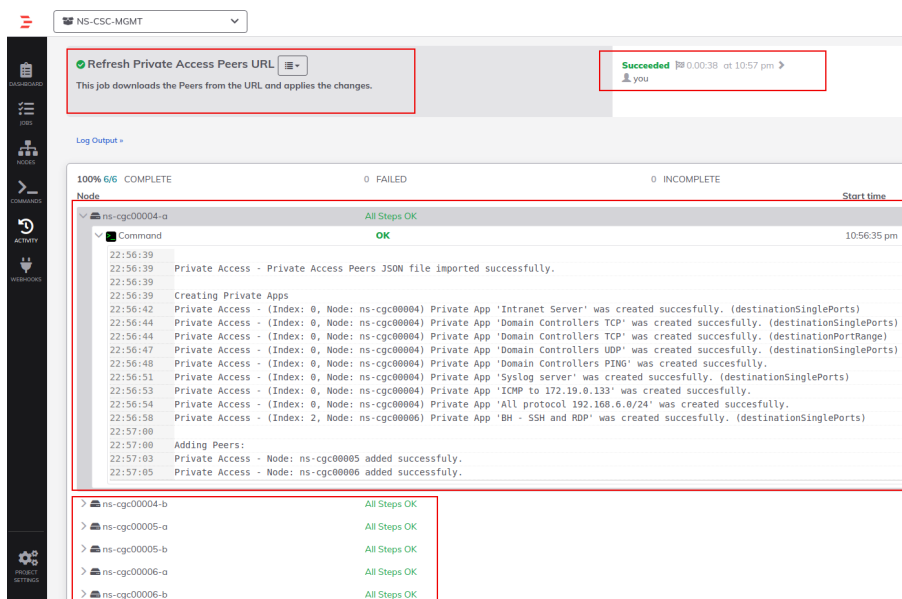
- Go to the Project <name> -> All Jobs -> Run " Refresh Private Access Peers URL"



- Select ALL nodes and click Run.



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



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

From Main Menu, go to 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: 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: 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

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

Private Apps:

Index: 0, NodeName: csc-mux-4-az, Location: Azure US East, publicIpAndUdpPort: 20.127.156.184:51820, privateCirdip: 192.168.7.39/24, Private Apps Qty: 1
Index: 1, NodeName: ns-csc-gre-v-1-0e, Location: AWS US, publicIpAndUdpPort: 10.213.100.84:51820, privateCirdip: 192.168.7.37/24, Private Apps Qty: 4
Index: 2, 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: 3, NodeName: ns-cgc00004, Location: MHB-DC-KWM, publicIpAndUdpPort: 82.68.6.74:51821, privateCirdip: 192.168.7.11/24, Private Apps Qty: 6
Index: 4, NodeName: ns-cgc00006, Location: MHB-BH-DC, publicIpAndUdpPort: 217.155.196.81:51820, privateCirdip: 192.168.7.20/24, Private Apps Qty: 2

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

Creating Private Apps:
Private Access - (Index: 0, Node: csc-mux-4-az) Private App 'Allow all to Azure' was created successfully.
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'AWS - SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'AWS - tcp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 1, Node: ns-csc-gre-v-1-0e) Private App 'Allow iperf udp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 2, 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: 2, Node: ns-csc-gre-aws-v-0-4) Private App 'Allow iperf tcp' was created successfully. (destinationSinglePorts)
Private Access - (Index: 2, Node: ns-csc-gre-aws-v-0-4) 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 'Domain Controllers TCP' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Domain Controllers UDP' was created successfully. (destinationPortRange)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'ICMP to 172.19.0.133' was created successfully. (destinationSinglePorts)
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Syslog ICMP' was created successfully.
Private Access - (Index: 3, Node: ns-cgc00004) Private App 'Syslog tcp port' was created successfully. (destinationSinglePorts)
Private Access - (Index: 4, Node: ns-cgc00006) Private App 'BH - SSH to Servers' was created successfully. (destinationSinglePorts)
Private Access - (Index: 4, Node: ns-cgc00006) Private App 'BH - SSH and RDP to Remote Server' was created successfully. (destinationSinglePorts)

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

Changes Security Group External:
Private Access - Inbound Port Rules 'mhb-csc-private-access-IS1820' added to Security Group 'csc-mux-4-az-eth0-NSG-1'
Private Access - Outbound Port Rules 'mhb-csc-private-access-051820, mhb-csc-private-access-051821' added to Security Group 'csc-mux-4-az-eth0-NSG-1'

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

Done!

9.4 Show Configurations and Status Private Access.

9.4.1 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: █
```

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

2. Select Peer

This section shows a more detailed information about the Peer.

```
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": "MH8-BH-DC",
  "publicKey": "B000lrseH+p3tWgk04j9rVawX2Fbqkj0d0JlyWlTsmI=",
  "publicIpAndUdpPort": "217.155.196.81:51820",
  "privateCirdIp": "192.168.7.20/24",
  "persistentKeepAlive": "no",
}
```

9.4.1.2 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/OSL2bsFCh2o=",
      "publicIpAndUdpPort": "52.4.62.40:51820",
      "privateCirdIp": "192.168.7.88/24",
      "persistentKeepAlive": "no",
      "networks": [
        "10.3.200.0/24"
      ],
      "privateApps": [

```

9.4.1.3 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": "2Z6bFHcsMATHHC7cDXu0hdUS6lLPKE98MCA33KRYz0c=",
      "publicIpAndUdpPort": "54.80.198.195:51820",
      "privateCidrIp": "192.168.7.89/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}
```

9.4.1.4 Show Firewall Local Rules

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

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

```
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": [
    {
      "localUdpPort": "51820",
      "peersPublicSourceIP": [
        "52.4.62.40",
        "34.230.146.174",
        "82.68.6.74",
        "82.68.6.76",
        "217.155.196.81"
      ]
    }
  ],
  "outboundFirewallRules": [
    {
      "remoteUdpPort": "51820",
      "peersPublicDestinationIP": [
        "52.4.62.40",
        "34.230.146.174",
        "82.68.6.76",
        "217.155.196.81"
      ]
    },
    {
      "remoteUdpPort": "51821",
      "peersPublicDestinationIP": [
        "82.68.6.74"
      ]
    }
  ]
}
```

9.4.2 via AWS Systems Manager or Rundeck

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

9.4.2.1 AWS Systems Manager

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

The screenshot shows the AWS Systems Manager console. At the top, it says 'AWS Systems Manager > Run Command > Command ID: caa5bcf8-3946-4408-b394-d92dd45cb49e > Output on: mi-08c465d750d2689ae'. Below this, it says 'Output on mi-08c465d750d2689ae'. Under 'Step 1 - Command description and status', the 'Status' is 'Success' and the 'Detailed status' is 'Success'. The 'Step name' is 'Runscripts' and the 'Start time' is 'Sun, 21 Nov 2021 09:46:15 GMT'. Under 'Output', it says 'The command output displays a maximum of 48,000 characters. You can view the complete command output in either Amazon S3 or CloudWatch I'. The output shows two lines: 'Peer 'ns-cgc00004' -> 192.168.7.11 is Alive. Source Port OK. Using '51821'' and 'Peer 'ns-cgc00005' -> 192.168.7.21 is Alive. Source Port OK. Using '51820''.

9.4.2.2 Rundeck

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

The screenshot shows the Rundeck interface. At the top, it says 'Show Private Access ALL Peers Status' with a dropdown menu. Below this, it says 'This job shows the reachability of all peer of an specific node.' On the right, it says 'Succeeded' with a user icon. Below this, it says 'Log Output >'. The main section shows a list of nodes and their execution status. The first node is 'ns-cgc00004-a' with 'All Steps OK'. The second node is 'ns-cgc00004-b' with 'All Steps OK'. The third node is 'ns-cgc00005-a' with 'All Steps OK'. The fourth node is 'ns-cgc00005-b' with 'All Steps OK'. The fifth node is 'ns-cgc00006-a' with 'All Steps OK'. The sixth node is 'ns-cgc00006-b' with 'All Steps OK'. The top bar shows '100% 6/6 COMPLETE' and '0 FAILED'.

9.5 Configure CSC Remote Management via Private Access.

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

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

The configuration will be:

```
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
```

10 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.

10.1 AWS Systems Manager

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

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

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

10.1.1 Create Documents

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

Steps:

1. Download the CloudFormation template from:

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

The screenshot shows the AWS CloudFormation console's 'Create stack' wizard. The 'Specify template' step is selected. Red boxes and numbers highlight the following elements:

- 1. The 'Create stack' button at the top.
- 2. The 'Template is ready' radio button under the 'Prepare template' section.
- 3. The 'Amazon S3 URL' radio button under the 'Specify template' section.
- 4. The 'Amazon S3 URL' text input field containing the URL: `https://mhb-netskope-cloudformation.s3.eu-west-1.amazonaws.com/MHB-Netskope-CSC-AWS-Systems-Manager-Documents-v-1-0.json`.

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

4. Put the Stack Name

CloudFormation > Stacks > Create stack

Step 1
Specify template

Step 2
Specify stack details

Step 3
Configure stack options

Step 4
Review

Specify stack details

Stack name

Stack name

Documents-CSC-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.

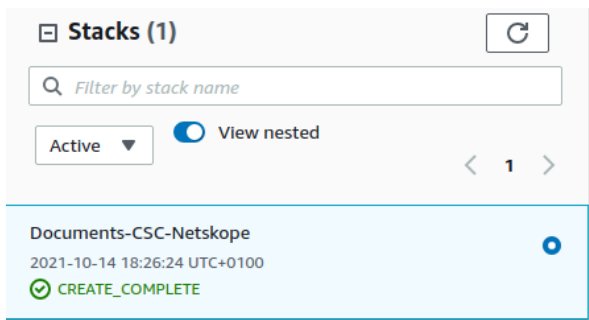
No parameters

There are no parameters defined in your template

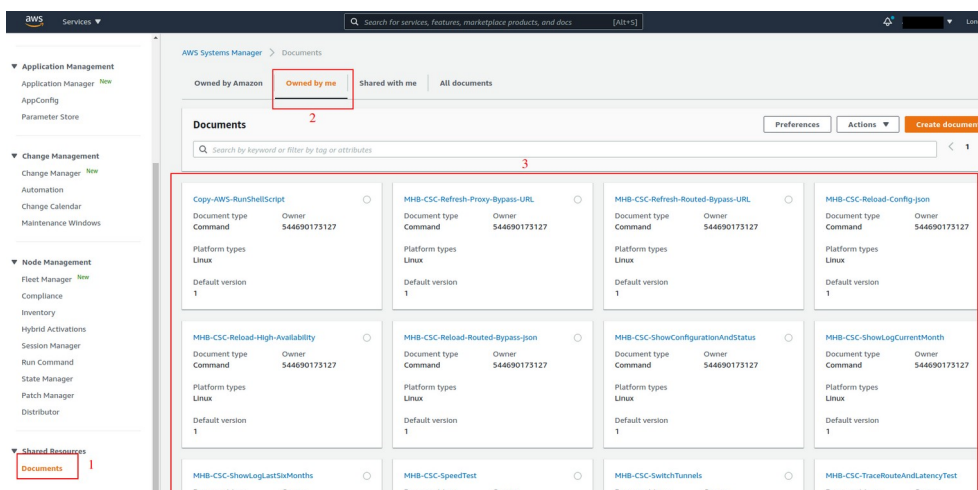
Cancel Previous Next

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

6. Wait the Stack to complete.



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



8. Done!

10.1.2 Run Commands

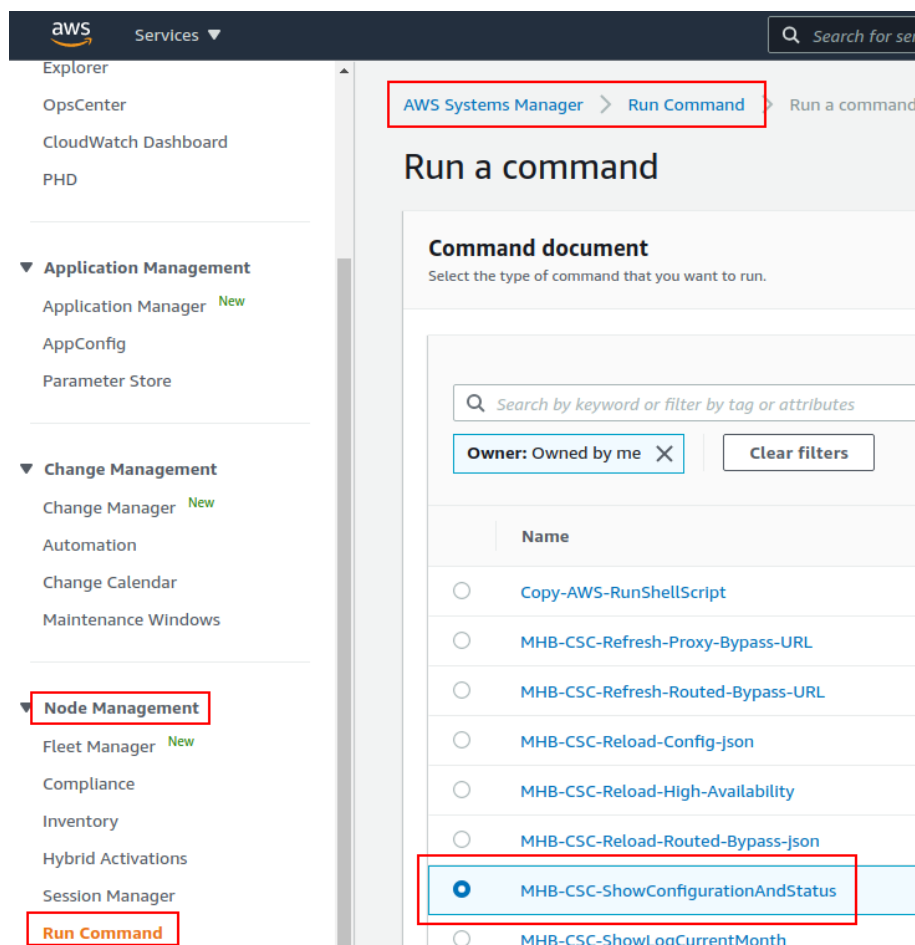
After you have created the Documents, you are ready to Run Commands on the CSC.

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

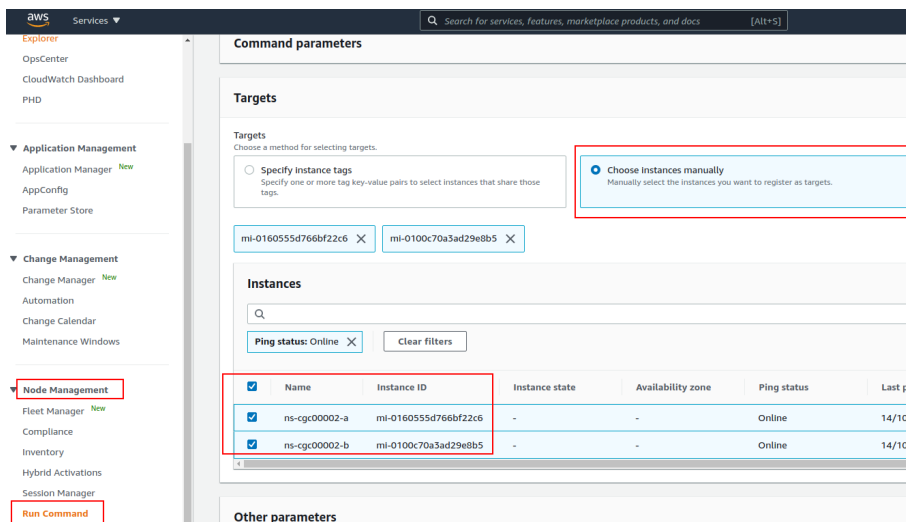
To "Run Commands", go to AWS Systems Manager → Instances & Nodes → Run Command.

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

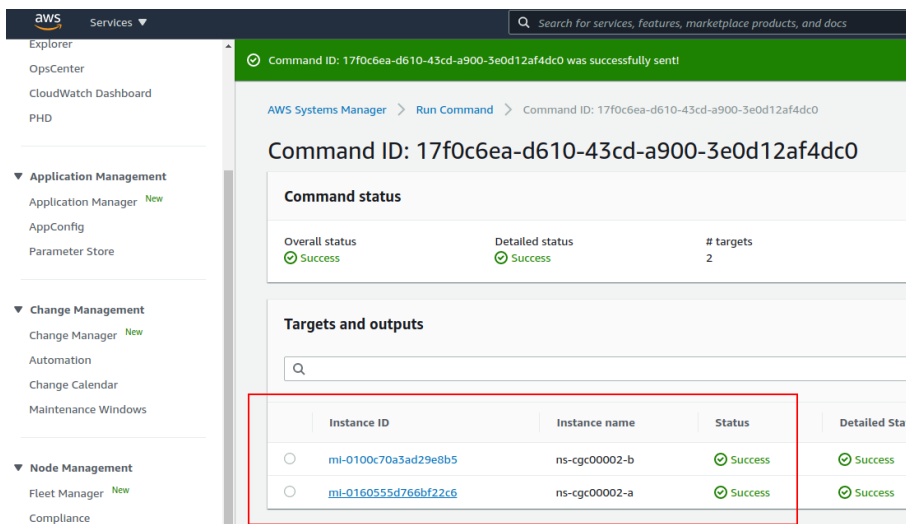
1. Run a Command
2. Select the Document created (Tip: Select "Owned by me")



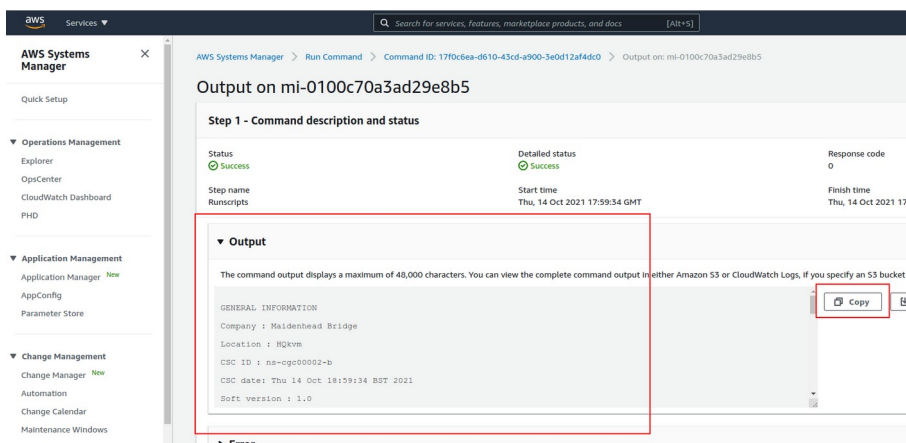
3. Scroll down and Select the Instances



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



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



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

```
*Unsaved Document 1 x
1
2 GENERAL INFORMATION
3 Company : Maidenhead Bridge
4 Location : HQkvm
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.NETSKOPE.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
```

10.1.3 List of Documents available for "Run Command"

1. "MHB-CSC-ShowConfigurationAndStatus": Executes "Show Configuration and Status"
2. "MHB-CSC-SpeedTest": Performs speedtest.net on the CSC.
3. "MHB-CSC-TraceRouteAndLatencyTest": Performs MyTraceRoute test against the Primary and Secondary ZEN. It also does a Reverse Test from the tunnel active to your Public IP if the tunnel is up.
4. "MHB-CSC-Refresh-Proxy-Bypass-URL": Refresh the Proxy Bypass list using the values of the Proxy Bypass PAC file stored in the URL configured.
5. "MHB-CSC-Refresh-Routed-Bypass-URL": Refresh the Routed Bypass list using the values of the JSON file stored in the URL configured.
6. "MHB-CSC-ShowLogCurrentMonth": Shows current month logs.
7. "MHB-CSC-ShowLogLastSixMonths": Shows last six month logs.
8. "MHB-CSC-SwitchTunnels": Switch tunnels.
9. "MHB-CSC-Reload-Config-json": Reloads the values of config.json file.
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.

10.2 Rundeck

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

Installation Steps:

1. Install Rundeck. 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 web interface. At the top, a dropdown menu is set to 'NS-CSC-MGMT' and the word 'Project' is displayed. Below this, the 'Edit Nodes File' section is active, showing the file path '/home/rundeck06/rundeck/NS-CSC-MGMT-NODES.json'. The 'Source' is '2. File Reads a file containing node definitions in a supported format' and the 'Format' is 'json'. The 'Description' is '/home/rundeck06/rundeck/NS-CSC-MGMT-NODES.json'. A 'Soft Wrap' button is visible. The main area displays a JSON configuration for nodes. A red box highlights the first node definition, and a red '3' is next to it. The JSON is as follows:

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

At the bottom left, a 'PROJECT SETTINGS' button is highlighted with a red box. At the bottom right, 'Cancel' and 'Save' buttons are visible.

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

The following screen shows the list of Jobs available.



11 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.

11.1 config.json file

You can use this file to change DNS, Log Servers, IPsec 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-mux-ns-azure",
  "type": "config",
  "version": "1.0",
  "dns": {
    "useCloudDNS": true,
    "primaryDnsIP": "",
    "secondaryDnsIP": ""
  },
  "syslogServers": {
    "primarySyslogIP": "10.2.3.4",
    "secondarySyslogIP": "",
    "syslogTcpPort": 514
  },
  "ipsecInformation": {
    "reconfigureTunnelsNodes": false,
    "regeneratePreSharedKeys": false,
    "autoDiscovery": true,
    "primaryIpsecPop": "163.116.146.38",
    "primaryProbepAddress": "10.146.6.216",
    "primaryLocation": "US,Washington,IAD2",
    "secondaryIpsecPop": "163.116.135.38",
    "secondaryProbepAddress": "10.135.6.216",
    "secondaryLocation": "US,NewYork,NYC1",
    "nstunPublicIP": {
      "nstun1": "20.127.156.167",
      "nstun2": "20.127.156.168",
      "nstun3": "20.127.156.169",
      "nstun4": "20.127.156.166"
    }
  },
  "bypassProxyPacUrl": "https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/mhb-netskope-bypass-vm.pac",
  "routedBypassJsonFileUrl": "https://mhb-netskope-pac-files.s3.eu-west-1.amazonaws.com/routedBypassRulesFile.json",
  "bypassProxyPublicIP": "20.127.156.184",
  "routedBypassPublicIP": "20.127.156.184",
  "privateAccesPublicIpPort": "20.127.156.184:51820",
  "privateAccesPeersJsonFileUrl": "https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json"
}
```

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.

11.2 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.

11.3 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+IXXJte14tj3zIMNq+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": "TrMvSoP4jYQlY6RlZBgbssQqY3vxl2Pi+y71IOWWXX0=",
      "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.

11.4 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
```

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

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

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

```
{
  "model": "csc-mux-ns-azure",
  "type": "highAvailability",
  "version": "1.0",
  "highAvailability": {
    "haEnable": true,
    "haIamRole": "SystemAssigned",
    "haFirstCsc": {
      "vmName": "csc-mux-4-az-1",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "20.127.156.184"
    },
    "haSecondCsc": {
      "vmName": "csc-mux-4-az-2",
      "vmResourceGroup": "CSC-East-US",
      "haBypassPublicIp": "52.249.220.21"
    },
    "haPrivateAccessPublicIp": "20.127.156.184",
    "haRoutes": [
      {
        "routeName": "Server-default-route",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      },
      {
        "routeName": "to-163.116.128.80",
        "routeTable": "Servers-Route-Table",
        "resourceGroup": "RouteTables-East-US"
      }
    ]
  }
}
```

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

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

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

12 Appendixes

12.1 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 IPsec tunnel and the rest of the traffic via Routed Bypass.

The following JSON file does the work to redirect only Web traffic via the IPsec 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"
    }
  ]
}
```

12.2 Appendix B: Release Notes

12.2.1 Version 1.0

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

12.2.2 Version 1.1

Version 1.1 contains the following enhancements.

- Solved the problem when the Azure CLI returns the values of Public IP not ordered by interface ID. This problem affected only deployments in High Availability.

12.2.3 Version 1.2

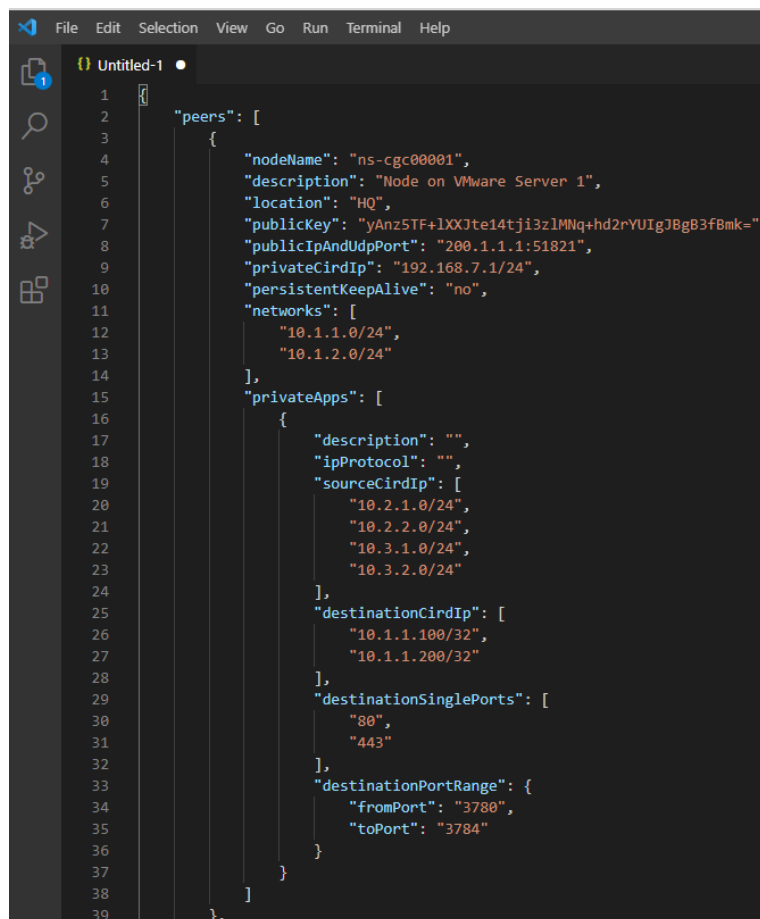
Version 1.2 contains the following enhancements.

- Solved the problem when trying to configure Custom DNS servers.
- Cosmetic changes in some Logs.
- Solved the problem when user "csccli" was disabled, and it is not possible to enable it again via Admin Console.
- In some circumstances, the CSC was not rebooting and was required to reboot via the Azure console. This issue is solved in this release.
- config.json file has added two new values: "reconfigureTunnelsNodes" and "regeneratePreSharedKeys". These new parameters are to avoid or force the reconfiguration of the tunnels or Pre Shares keys each time the reload config command is executed.

12.3 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.

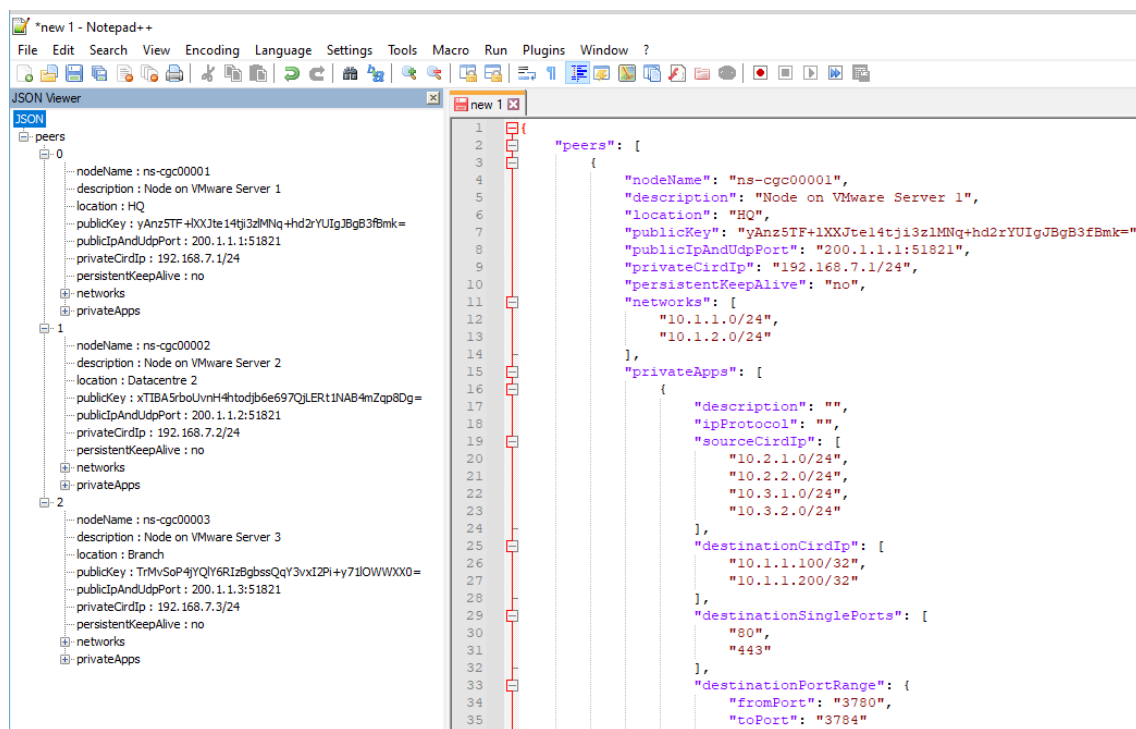
12.3.1 Visual Code



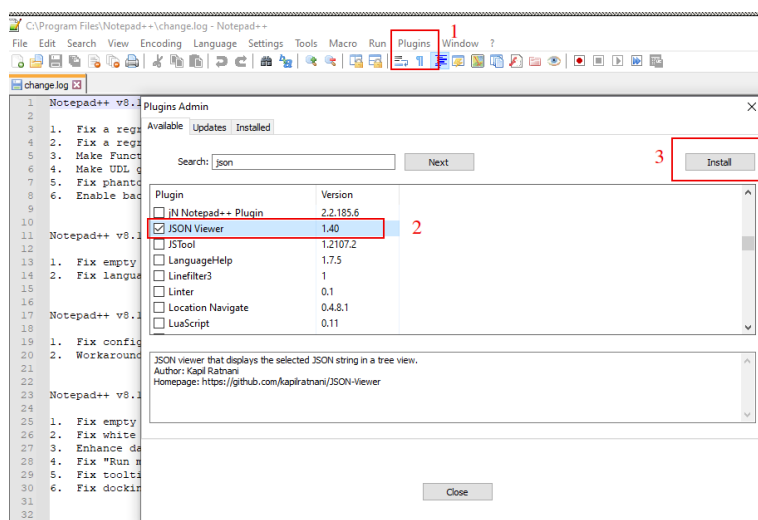
```
1  {
2    "peers": [
3      {
4        "nodeName": "ns-cgc00001",
5        "description": "Node on VMware Server 1",
6        "location": "HQ",
7        "publicKey": "yAnz5TF+lXXJte14tji3zIMNq+hd2rYUIGJBg83fBmk=",
8        "publicIpAndUdpPort": "200.1.1.1:51821",
9        "privateCirdIp": "192.168.7.1/24",
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   ]
41 }
```

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

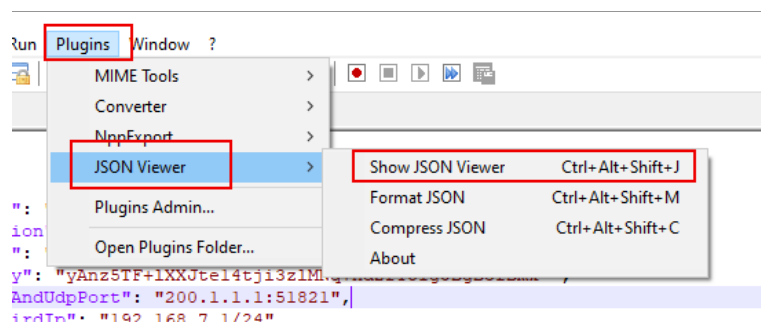
12.3.2 Notepad ++



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



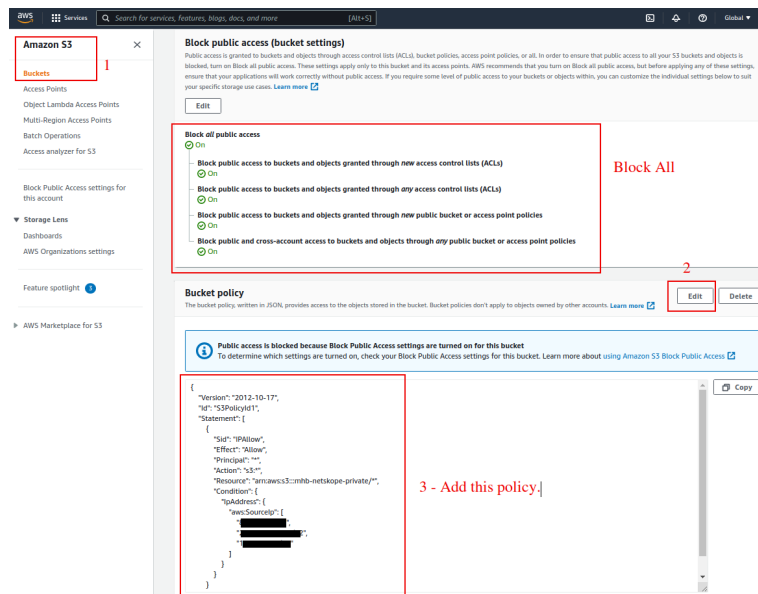
3. Create your JSON file.
4. To Check your JSON file go to: Plugins -> JSON Viewer -> Show JSON Viewer.



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

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

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



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

3. Done!