Maidenhead Bridge

Private Cloud Private Access (PriCPA)

Cloud Security Connector PriCPA for Virtual Platforms

Administrator Guide

Version 1.1.0

April 2024

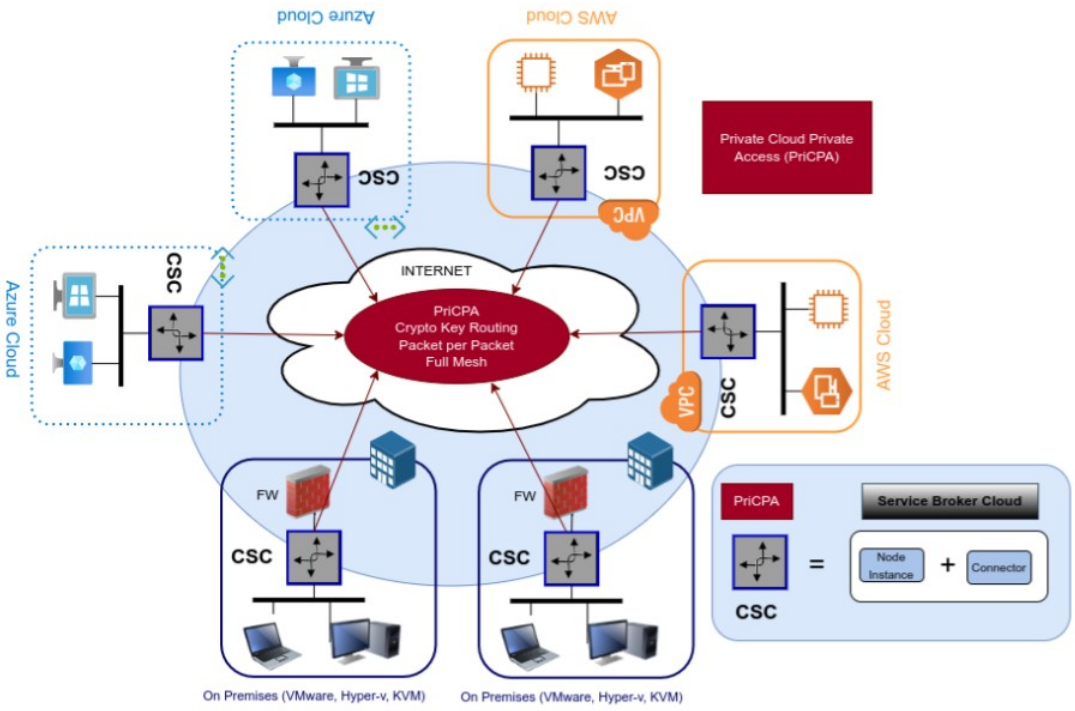
Table of Contents

1 Introduction to Private Cloud Private Access (PriCPA)

1.1 What is PriCPA?

Private Cloud Private Access (PriCPA) is a cloud-native solution for WAN communications that covers the scenarios of site-to-site, site-to-cloud and cloud-to-cloud, following the principles of Zero Trust. Legacy security networking solutions cannot be forklifted to the cloud. There are technical, operational and security limitations when using Legacy solutions. Networking engineers designed Legacy networking solutions to communicate branches, central offices and data centres.

Cloud communications arrived with new challenges of networking, security and, mainly, operational agility. Applications are now distributed in multiple VNET/VPCs of different clouds, APIs are required to be accessed from numerous sites, and we still need to communicate with on-prem services. An agile method of any-to-any secure encrypted communications with zero trust is required, and Private Cloud Private Access is the answer to this challenge.



Benefits:

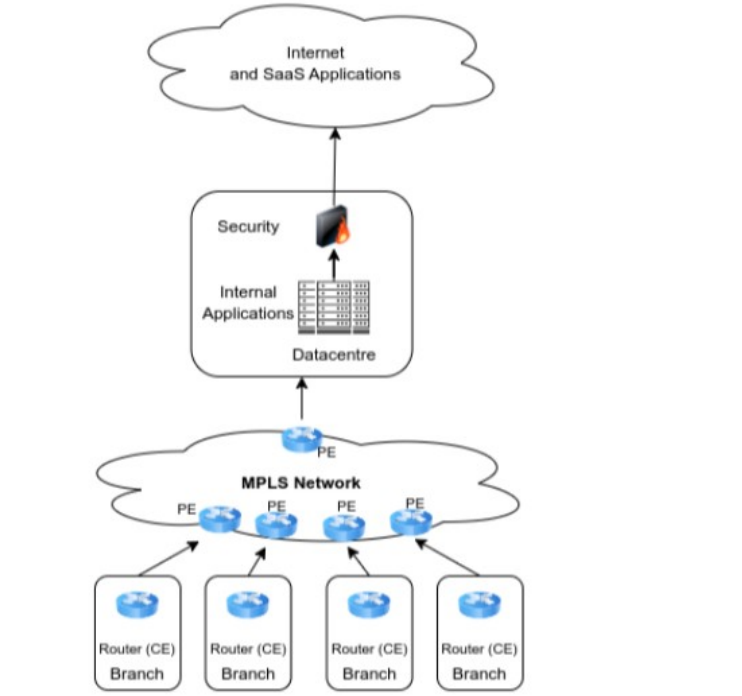
- 80% savings compared with VPN Gateways or Service Broker Cloud.
- Any to Any Communications. (Crypto Key Routing)
- All traffic is encrypted.
- All protocols are supported.
- Scalable: Multiple sites and multiple PriCPA clouds.
- Zero Trust.
- Networking as a code.

Benefits:

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

1.2 The evolution of WAN communications

1.2.1 MPLS & SDWAN

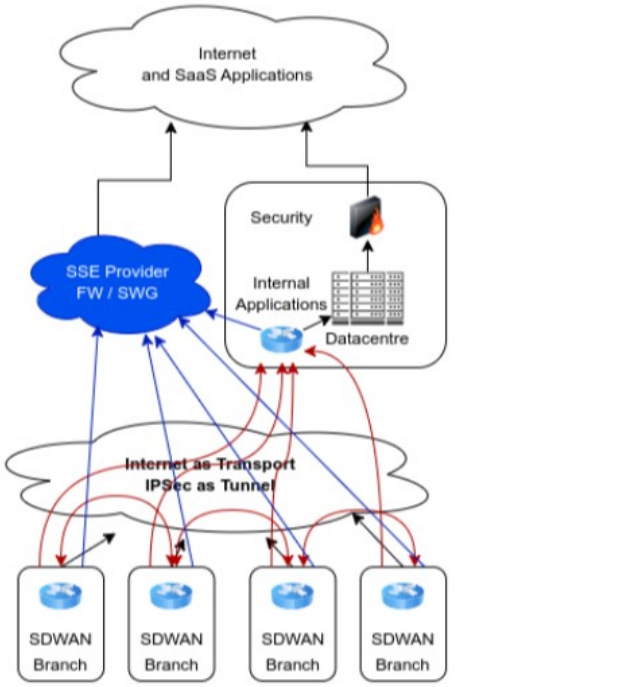


Benefits:

1. Any to Any Communications. (Label Switching)
2. Traffic Engineering.
3. All protocols are supported.

Drawbacks:

1. Unencrypted Private Traffic.
2. Expensive Links.
3. Bottleneck to the Internet.
4. Not Zero Trust.



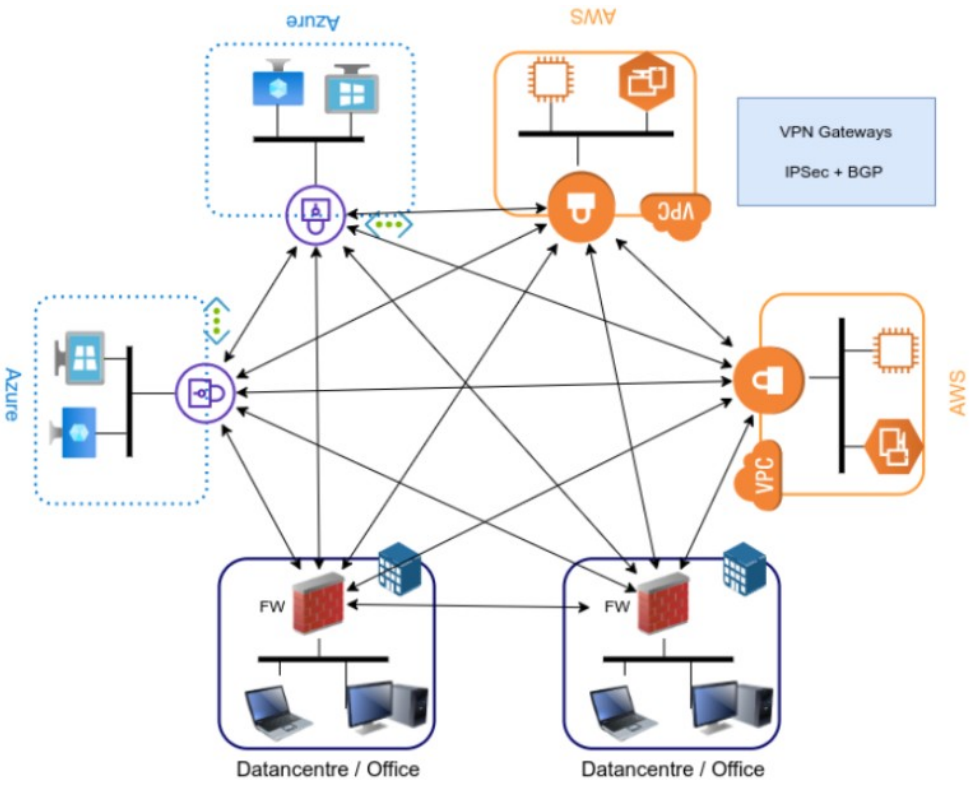
Benefits:

1. Cheap Internet Links as Transport
2. Encrypted Private Traffic.
3. All protocols are supported.

Drawbacks:

1. Lack of Scalability for Any to Any Traffic: Ten sites require 99 tunnels! [nx(n-1)]
2. Outdated / Unmaintained Linux Kernels - Security Risk.
3. Not Cloud friendly.
4. Not Zero Trust

1.2.2 VPN Gateways (IPsec + BGP)



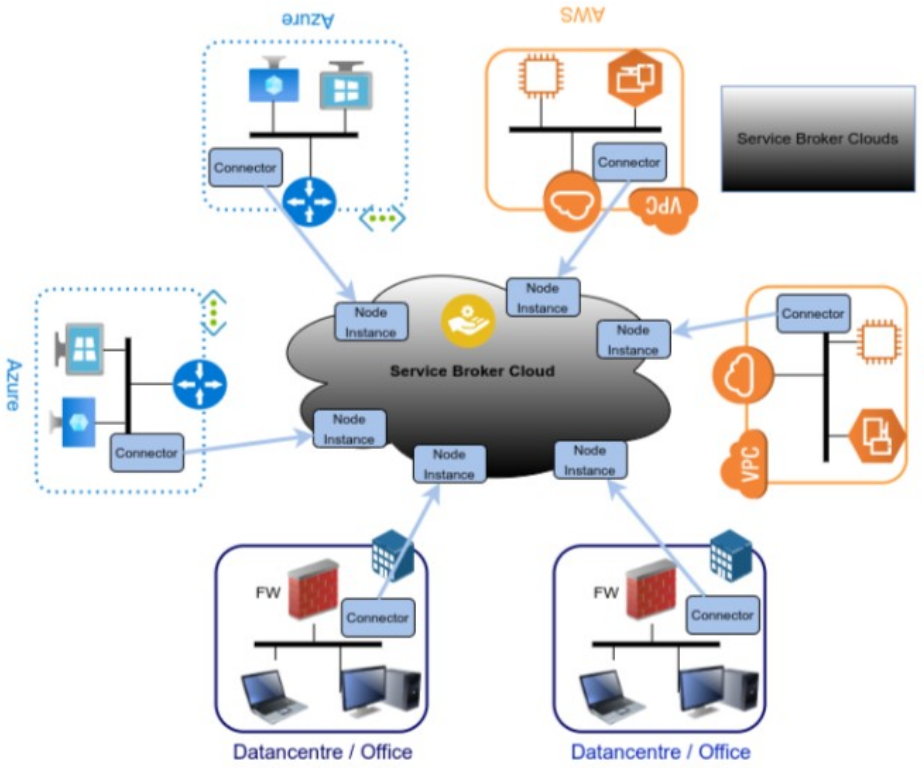
Benefits:

1. Encrypted Private Traffic.
2. All protocols are supported.

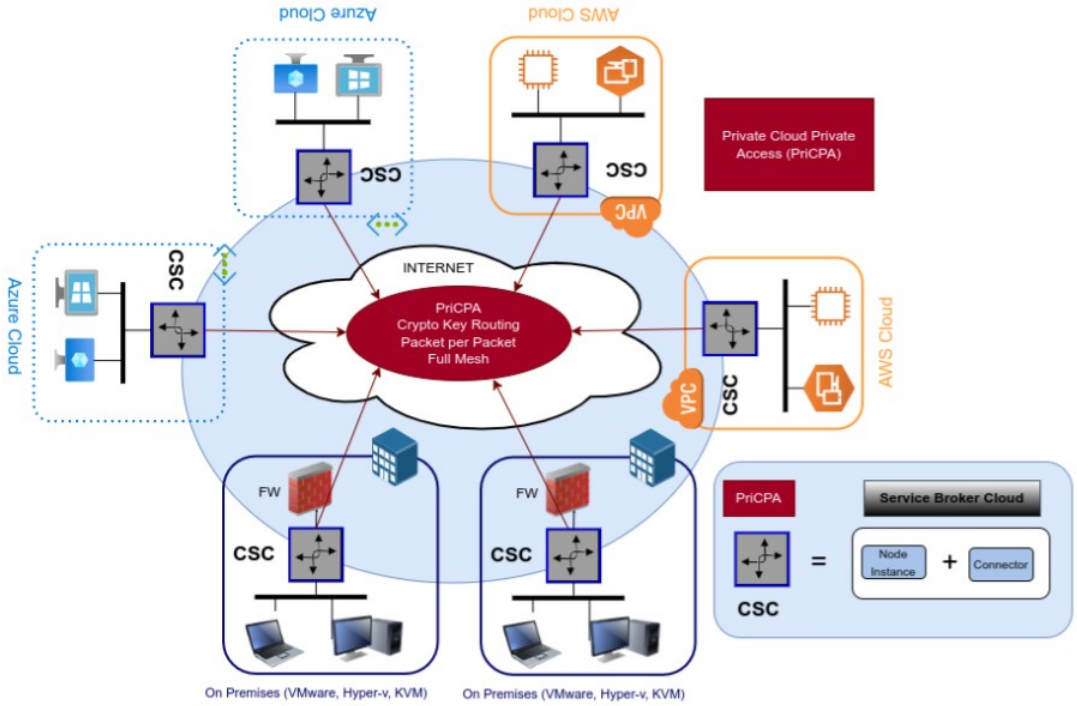
Drawbacks:

1. Expensive VPN Gateways.
2. Lack of Scalability for Any to Any Traffic: Ten sites require 99 tunnels! [nx(n-1)].
3. Not Zero Trust.
4. Complex to set up and maintain. IPsec + BGP

1.2.3 Service Broker Cloud



1.2.4 Maidenhead Bridge: Private Cloud Private Access



Benefits:

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

Benefits:

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

1.3 Comparing WAN technologies

The following table compares the differences among WAN solutions available:

Feature	PriCPA	Secure Broker	VPN Gateways	SD-WAN	MPLS
Full path Encryption	Yes	Partial	Yes	Yes	No
Supports all protocols	Yes	No	Yes	Yes	Yes
DevOps deployment	Yes	Yes	Partial	No	No
Networking as a code	Yes	No	Partial	No	No
Zero Trust	Yes	Yes	No	No	No
Cloud Native	Yes	Yes	Partial	No	No
Scalable any-to-any	Yes	Yes	No	No	Yes
Internet as transport	Yes	Yes	Yes	Yes	No
Simple Setup	Yes	Partial	No	No	No
Traffic visibility end-to-end	Yes	No	Yes	Yes	Yes
Per Packet Encryption	Yes	No	No	No	No
Crypto Key Routing	Yes	No	No	No	No
Very Cost effective	Yes	No	No	No	No

2 Key benefits of the Cloud Security Connector PriCPA.

With Private Cloud Private Access, you can connect all sites securely on a Zero Trust model. The CSC PriCPA secures your Private Traffic between your physical and cloud locations. The key benefits are:

- **Savings:**
  - 80% savings compared with Cloud VPN Gateways or Service Broker Clouds.
  - Reduced TCO.
- **Performance and Scalability:**
  - High Performance: 1 Gbps encrypted traffic per CSC.
  - Multiple sites can be deployed.
  - Multiple PriCPA clouds can be created.
- **Flexibility:**
  - Any to Any Communications. (Crypto Key Routing).
  - All protocols are supported.
- **Security:**
  - Full hardened device.
  - All traffic is encrypted using latest state of the art encryption protocols.[1]
  - Zero Trust.
  - Blocks Lateral movement.
  - Automatic Security Group provisioning (Azure. AWS. Gcloud)
- **Simplicity:**
  - No Networking knowledge required.
  - No operational burden for Administrators.
  - Networking as a code: Single JSON file for policies.
  - DevOps automated deployment: Azure ARM, Cloudformation Template, Terraform, etc.
  - 2 Steps configuration: Onboard the Node to PriCPA Cloud and Deploy Policies (Single JSON file).

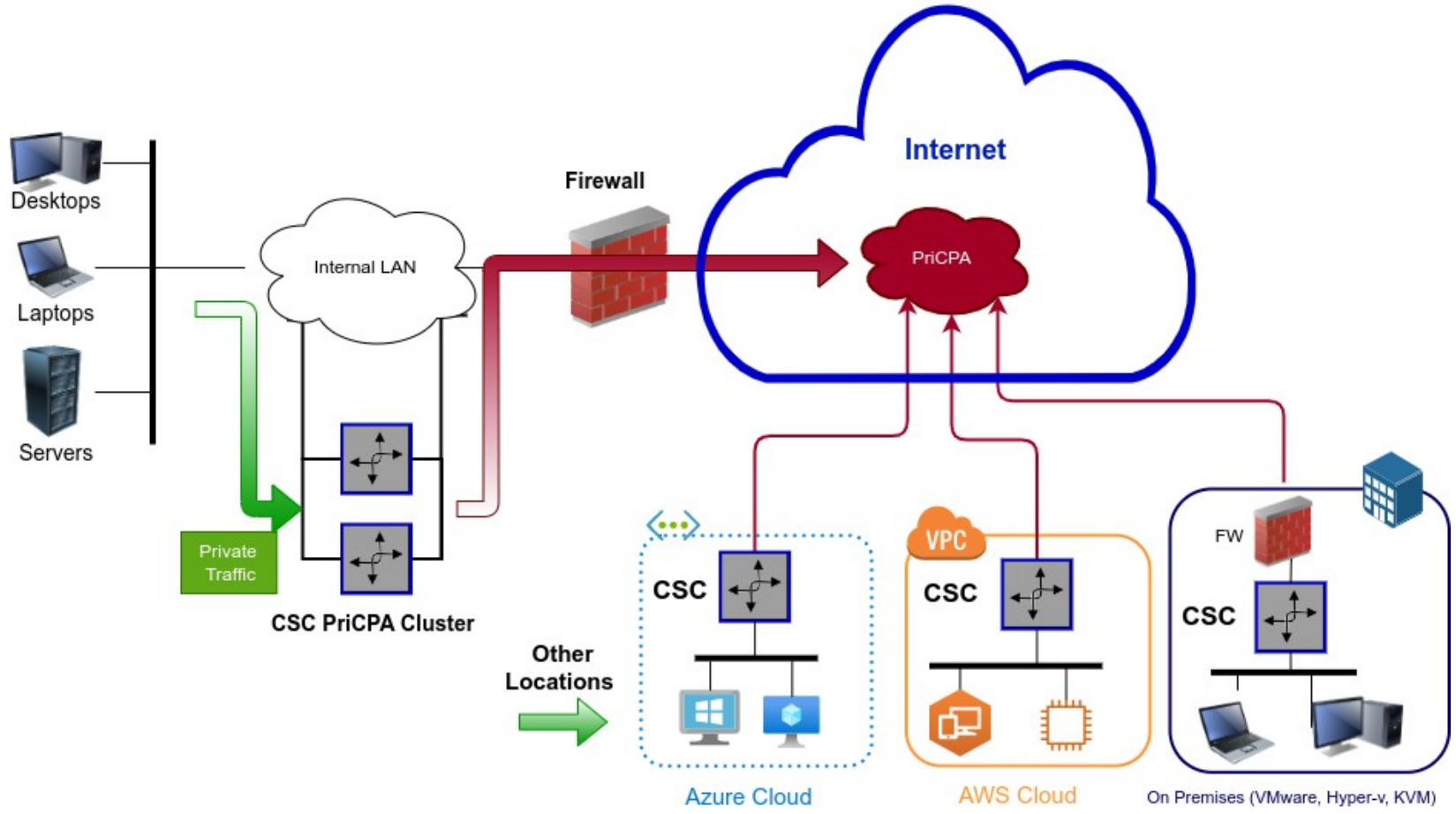
1 The CSC PriCPA uses Wireguard protocol. Wireguard is a trademark of Jason Donenfeld.

- **Visibility:**
  - Traffic Logs and System Logs.
  - Traffic visibility End to End.
  - Source IPs preserved.
- **High Availability:**
  - Local Clustering.
  - Multiple uplinks supported. .
- **Compatibility:**
  - 100% Compatible with CSCs for Zscaler and Netskope.
  - 100% Compatible with devices that supports Wireguard[2] Protocol.
- **Simple Management:**
  - Local Management: SSH Admin Console with configuration wizards, full status reporting.
  - Remote Management: No proprietary software required. You can use any change management tool to configure and update the CSC, such as Azure CLI "Run Command", AWS System Manager (SSM agent), Ansible, Rundeck, scripting via SSH or similar.
  - SNMP v2c and v3 support.
  - Radius/MFA for SSH Admin Console access.
  - SIEM/Syslog integration for Traffic and Systems Logs.
  - TCPDump integrated in the SSH Admin Console.
  - Linux terminal console allowed (csccli user).

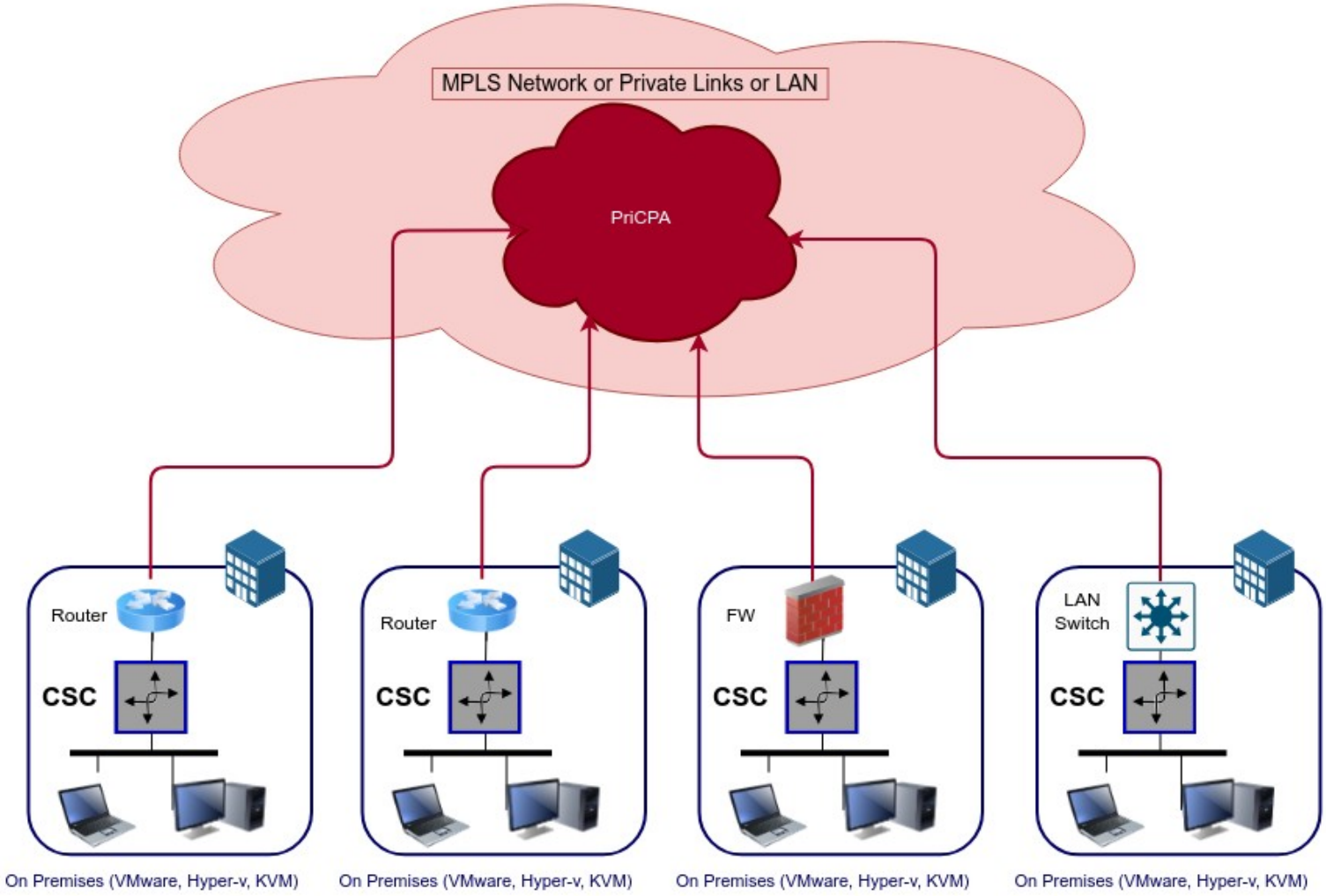
2 The CSC PriCPA uses Wireguard protocol. Wireguard is a trademark of Jason Donenfeld.

3 Network Diagrams

3.1 PriCPA over Internet

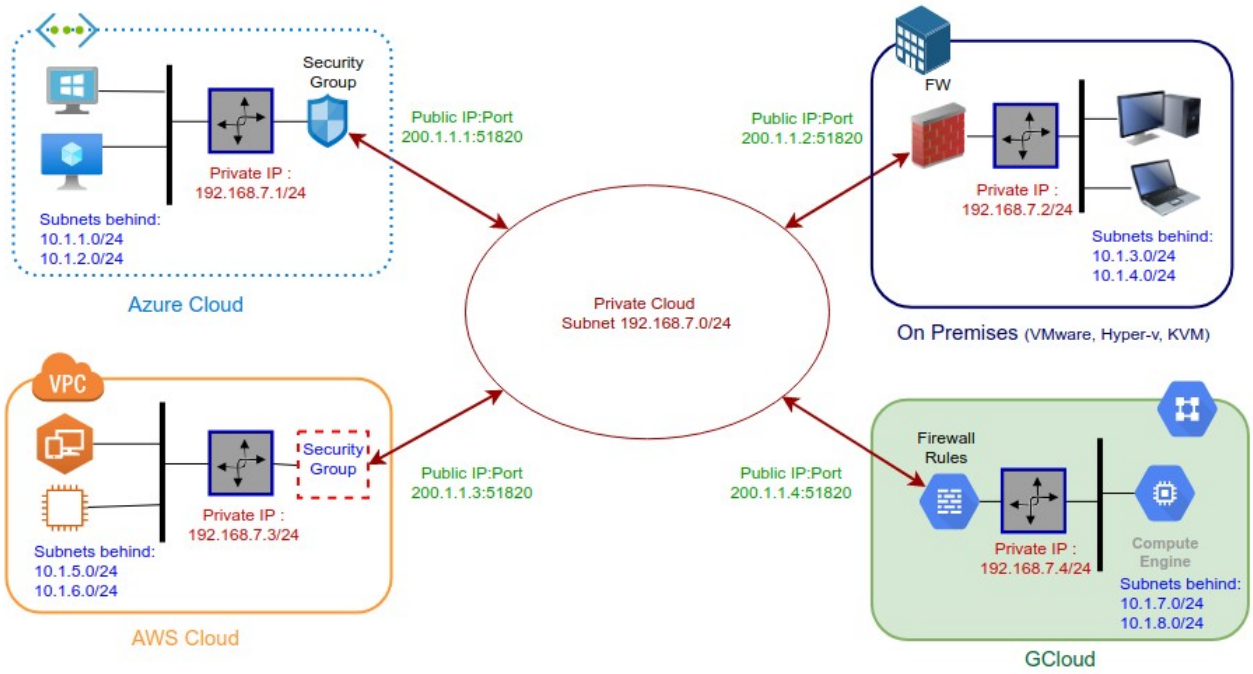


3.2 PriCPA over MPLS or Private Links or LAN



4 Designing your Private Cloud

The following network diagram shows an example of IP addressing for PriCPA.



Steps to Design Your Private Cloud:

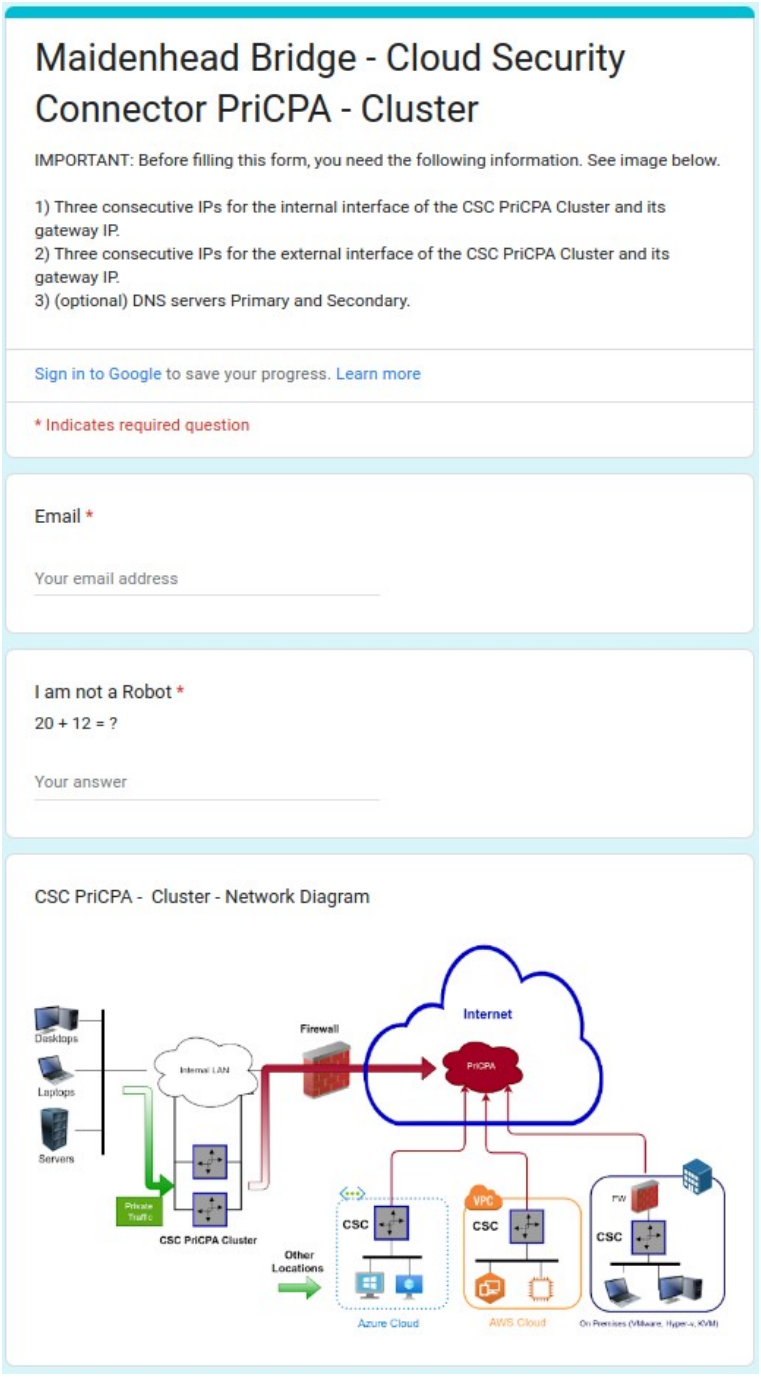
1. Select a Subnet for your Private Cloud. The example above is 192.168.7.0/24. Due to the Subnet being /24, up to 255 CSCs can participate in this Private Cloud.
2. Assign a Cloud Private IP to each CSC. In this example, we are assigning 192.168.7.1 to 192.168.7.4
3. The CSC on Public Clouds will automatically select the Public IP[3]. When the CSC is On-Prem, you must choose a public IP configured on your Firewall. You can select the UDP port to use at each location. For simplicity, using the same port at all locations is recommended. The default UDP port is 51820.
4. Gather the information on the Private Subnets behind each CSC. This information will be required when configuring the Peers.
5. Firewall Rules (or Security Group 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 necessary rules for FW configuration

3 IMPORTANT: The "Public IP" can be an Internet Public IP or an RFC 1418 IP (10/8, 172.16/12 and 192.168/16) when using private networks as transport. (e.g. MPLS). You can mix the type of IPs if the CSC has an uplink on the Internet and another on an MPLS network.

5 Creating the CSC PriCPA for Virtual Platforms

5.1 Filling the form

After you buy the CSC, you will receive a welcome email indicating that you must fill out the form with your data.



Maidenhead Bridge - Cloud Security Connector PriCPA - Cluster

IMPORTANT: Before filling this form, you need the following information. See image below.

1) Three consecutive IPs for the internal interface of the CSC PriCPA Cluster and its gateway IP.
2) Three consecutive IPs for the external interface of the CSC PriCPA Cluster and its gateway IP.
3) (optional) DNS servers Primary and Secondary.

Sign in to Google to save your progress. Learn more

* Indicates required question

Email *

Your email address

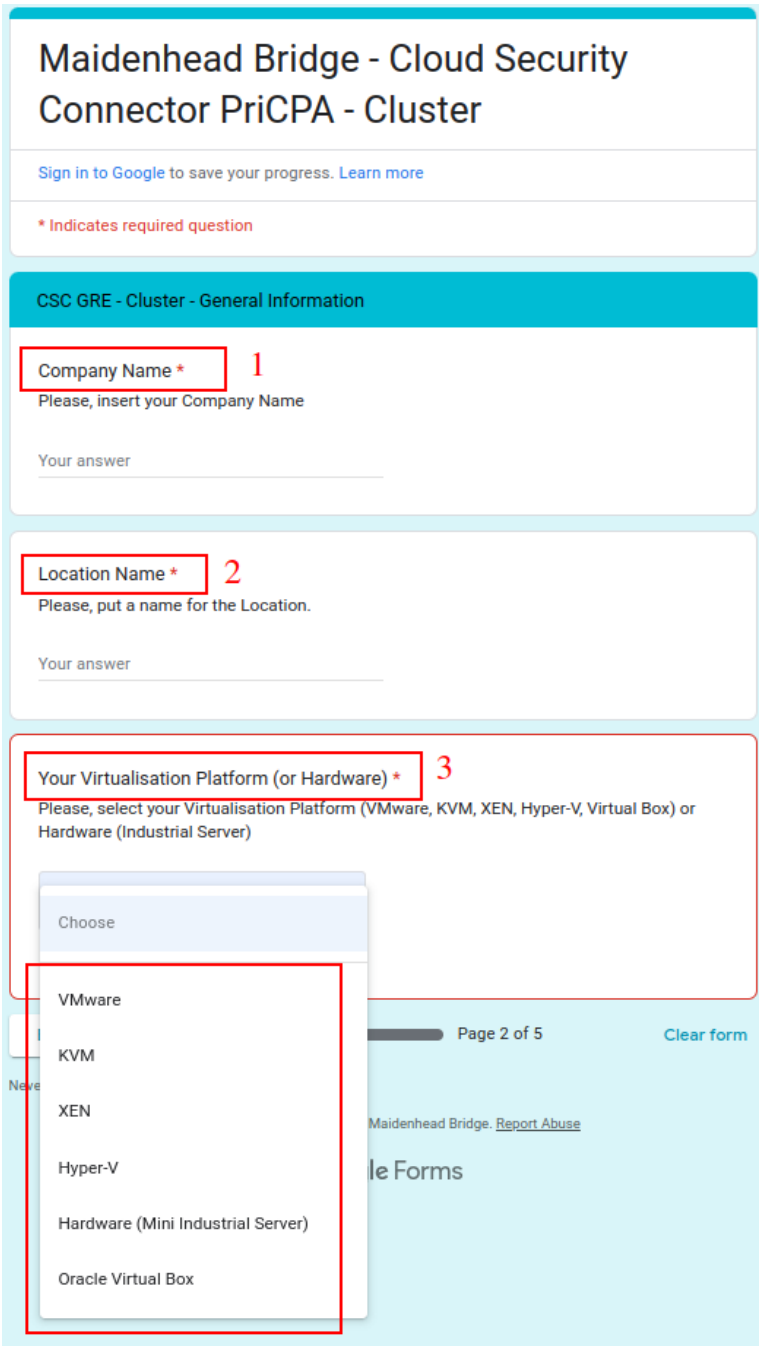
I am not a Robot *

20 + 12 = ?

Your answer

CSC PriCPA - Cluster - Network Diagram

5.1.1 General Information

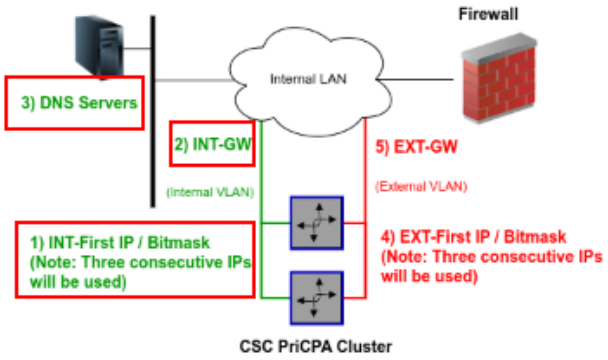


5.1.2 IP Addressing (Internal)

IP Addressing

Please note that all addresses are PRIVATE (RFC 1918)

CSC GRE - Cluster - IP Addresses



Internal IP address (Private IPs - RFC 1918)

IMPORTANT! The CSC PriCPA Cluster requires THREE IPs of the same subnet for Internal Interfaces . For simplicity, we ask for the first one and the system will configure the other two.

For example, if you put INT-First-IP/BitMask = 192.168.24.100/24 , the IPs 192.168.24.101-102 will be also reserved.

The First IP is the Internal Cluster IP (and GW to other locations). The second is the VM-a internal interface and the third is VM-b internal interface.

(1) INT-First-IP/BitMask * 1

This is the First IP of Internal Interfaces. Please, ingress it on the format of <ip>/<bitmask> . For example: 192.168.24.100/24

172.19.0.22/24

(2) INT-GW * 2

Please, enter the value of Internal Gateway

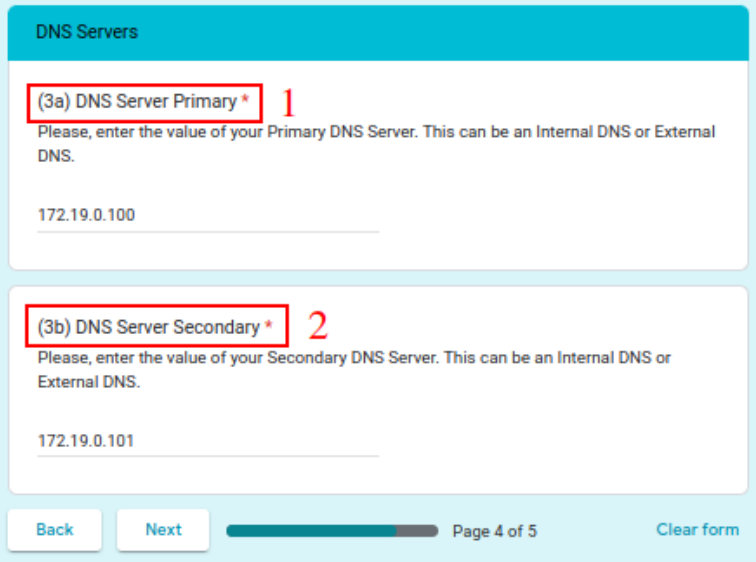
172.19.0.254

(3) DNS Servers * 3

The CSC has Google DNS Servers configured (8.8.8.8, 8.8.4.4). If you want to configure your Corporate DNS servers, please do it here. Do you want to use your DNS servers?

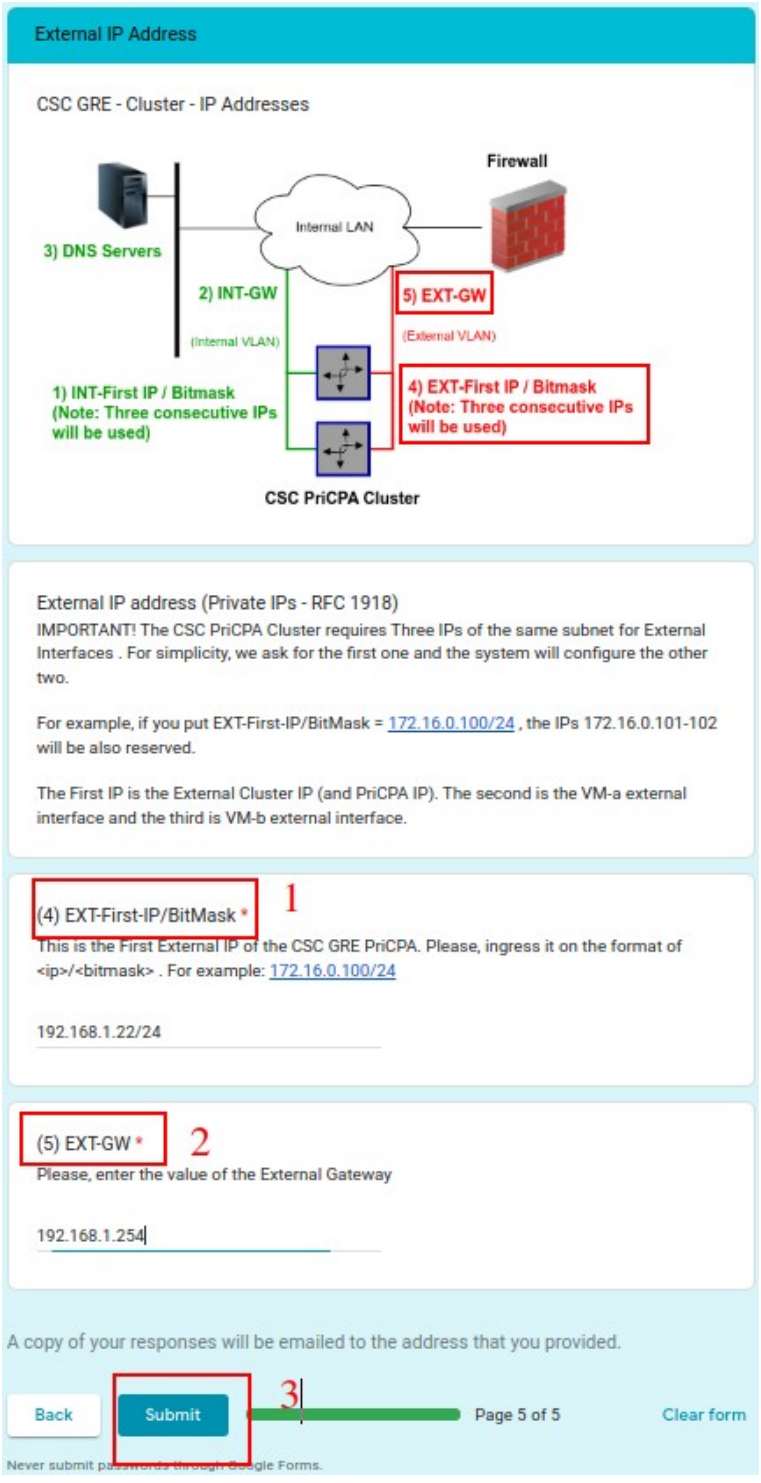
Yes

5.1.3 DNS servers



- Click "Next"

5.1.4 External IP address



Important: After filling out the form, you will receive the URL links to download the CSC VM images in the format you selected: VMware (OVA), HyperV (.vhdx disk) or KVM (qcow2 disk)

6 Firewall Requirements

The CSC PriCPA Cluster uses three IPs on the external interface, and it is required to set up specific NAT and Allow Rules in your Firewall for all of them.

The following table shows the name and purpose of each one.

External IP#	Name	Purpose
First	PriCPA IP	Source IP of this PriCPA Node.
Second	CSC IP(eth0) -a	External IP of the VM -a
Third	CSC IP(eth0) -b	External IP of the VM -b

6.1 NAT requirements

External IP#	Name	NAT Type required:	via Public IP
First	PriCPA IP	STATIC[4] or DYNAMIC.	The selected Public “Static IP” for this PriCPA Node.
Second	CSC IP(eth0) -a	DYNAMIC (also called 1:N NAT)	Any Public IP
Third	CSC IP(eth0) -b	DYNAMIC (also called 1:N NAT)	Any Public IP

4 When using Private Access, it is advisable to use Static Nat to avoid changing the packet's Source Port.

6.2 Allow Rules required

6.2.1 Firewall Local Rules JSON file

The CSC provides a JSON file with the list rules required for the PriCPA cloud. This file is updated every time you reload or refresh the Peers JSON file.

You can consult the file via SSH Admin Console:

1) Show PriCPA Configuration and Status. → 5) Show Firewall Local Rules

Here is an example:

```
{
  "nodeName": "pricpa-vm-c001001-b",
  "localPrivateIp": "192.168.1.25",
  "inboundFirewallRules": [
    {
      "localUdpPort": "51820",
      "peersPublicSourceIP": [
        "20.124.34.7",
        "92.40.213.220",
        "74.235.124.204",
        "4.246.221.166"
      ]
    }
  ],
  "outboundFirewallRules": [
    {
      "remoteUdpPort": "51820",
      "peersPublicDestinationIP": [
        "20.124.34.7",
        "92.40.213.220",
        "74.235.124.204",
        "4.246.221.166"
      ]
    }
  ]
}
```

If you want to retrieve the JSON from the CSC, the location is:

```
/usr/local/etc/mhb-csc/privateAccessLocalFirewallRules.json
```

6.2.2 Outbound Rules:

The following table shows the allow rules required.

External IP#	Source	Protocol	Ports / Service	Destination
First	PriCPA IP (**"localPrivateIp":)**	UDP	Configurable	Public IP of other PriCPA nodes. See "**outboundFirewallRules**" on Local Firewall JSON File.
		TCP	443	ip.maidenheadbridge.com
Second and Third	CSC IP(eth0) -a CSC IP(eth0) -b	ICMP	echo-request	FW Gateway ([5]) .
		TCP	80, 443	**Internet** (or specific to Ubuntu repos, ip.maidenheadbridge.com AWS Systems Manager ([6]), Others ([7]))
		UDP	53	If using Public DNS servers. (ie. 8.8.8.8, 8.8.4.4)
		UDP	123	If using Public NTP servers (i.e. ntp.ubuntu.com)

6.2.3 Inbound Rules:

External IP#	Destination	Protocol	Ports / Service	Source
First	PriCPA IP (**"localPrivateIp":)**	UDP	Configurable	Public IP of other PriCPA nodes. See "**inboundFirewallRules**" on Local Firewall JSON File.
Second and Third	CSC IP(eth0) -a CSC IP(eth0) -b	n/a	n/a	n/a

5 The CSC PriCPA Cluster pings the Gateway IP of the Firewall to check reachability.

6 When using AWS SSM Agent, allow HTTPS from the csc-external-a (-b) to AWS. The AWS destinations are: ssm.**<AWS region>**.amazonaws.com, ec2messages.**<AWS region>**.amazonaws.com

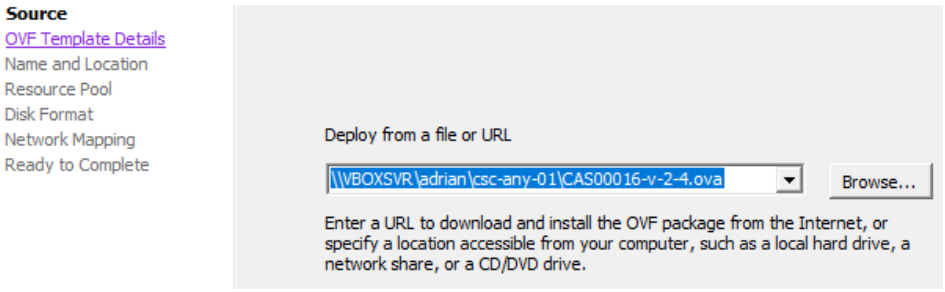
7 The CSC retrieves the Private Access JSON URL via csc-external-a (-b).

7 Installing the OVA or Disk file in your Virtual Platform.

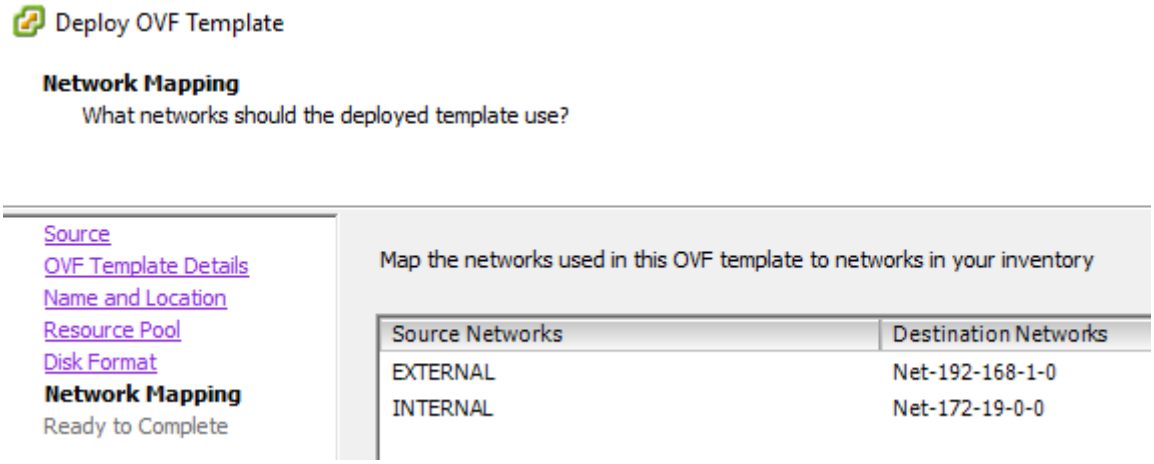
The following examples shows the installation on Vmware, Hyper-V and KVM.

7.1 Using VMware 5.x

1. Go to vSphere, File > Deploy OVF template
2. Select the OVA File:



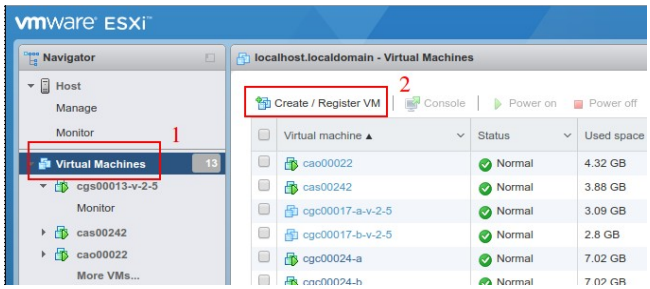
3. OVF Template Details: Click Next
4. Name and Location: Put the Name you want.
5. Resource Pool: Place the VM where you want.
6. Disk Format: Click Next
7. **Network Mapping: Please map the interfaces EXTERNAL and INTERNAL to your interfaces. Here an example:**



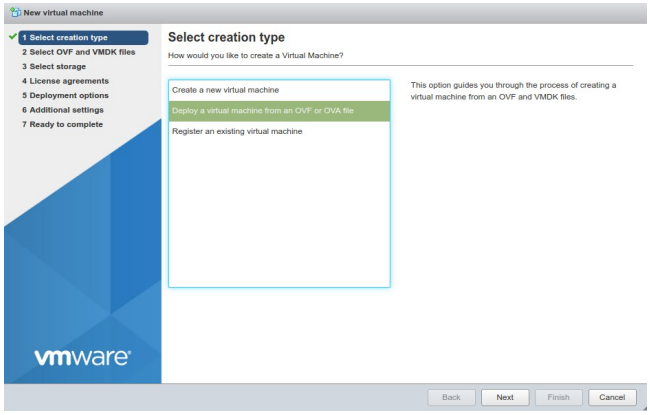
8. Click "Next"
9. Click "Finish"

7.2 Using VMware 6.x

1. Go to Virtual Machines → Create/Register VM

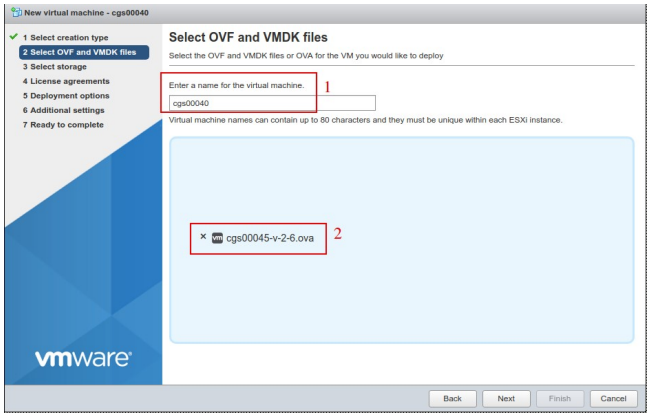


2. Deploy a virtual machine from an OVF or OVA file



3. Click "Next"

4. Put a "Name" and "Select the OVA File"



5. Click "Next"

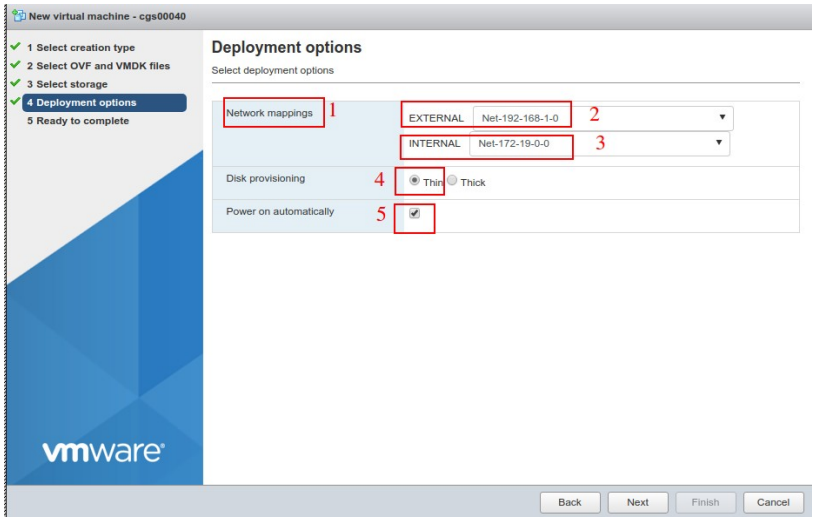
6. Select Storage and click Next

7. On "Deployment options", Select:

a) "Network Mappings" → Select "EXTERNAL" and "INTERNAL" interfaces of the CSC.

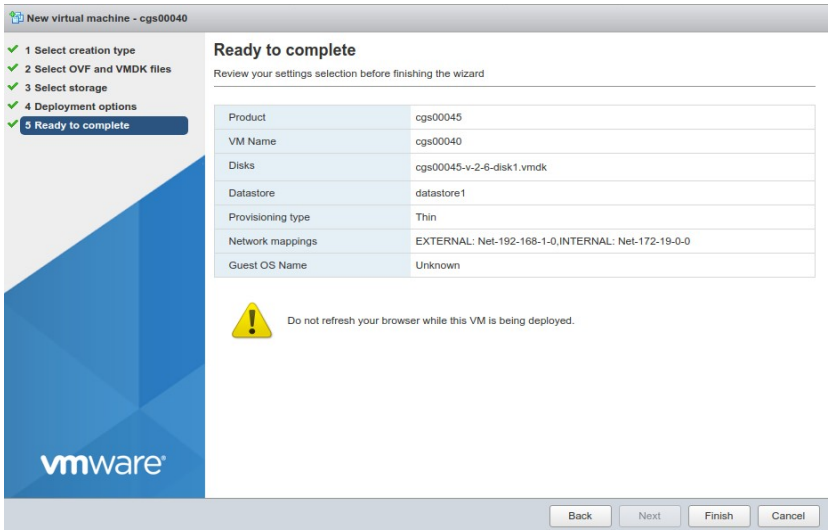
b) Disk Provisioning: Thin

c) Power on Automatically



8. Click "Next"

9. The next screen will show all values:

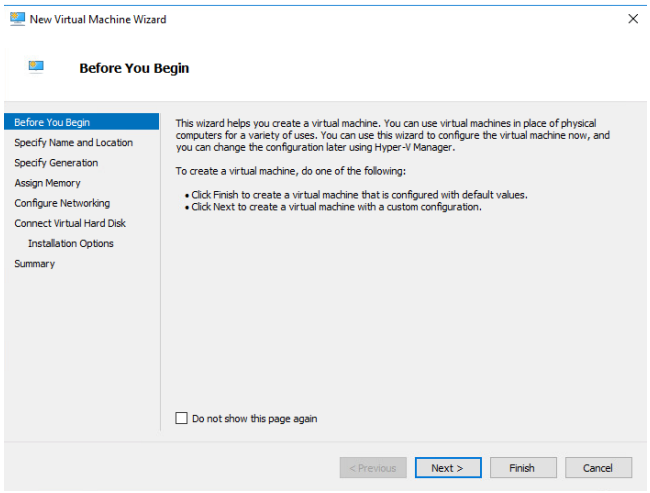


10. Click "Finish"

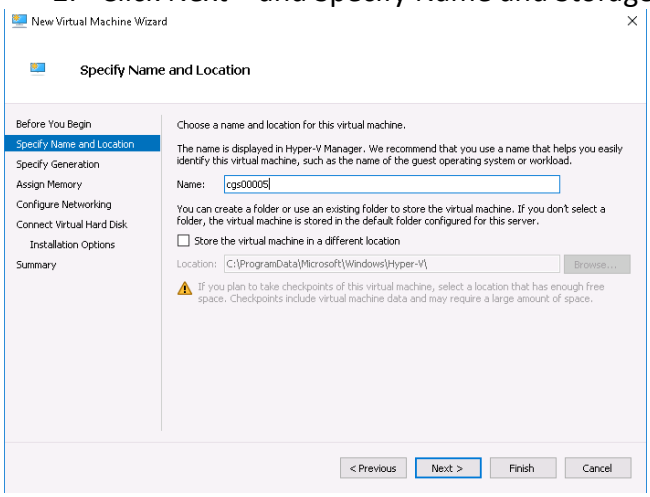
7.3 Using Hyper-V

Before to start: You will receive the CSC disk (.vhdx) on zip format. Please unzip it and place it on your Virtual Machine directory before to start this wizard.

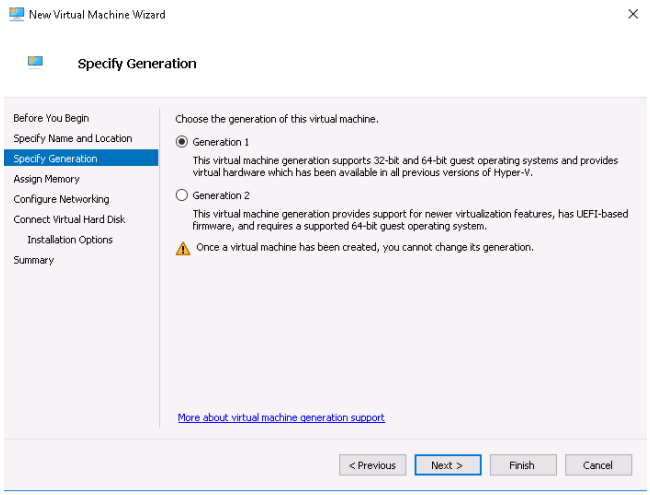
1. Go to Hyper-V and Click → Action → New



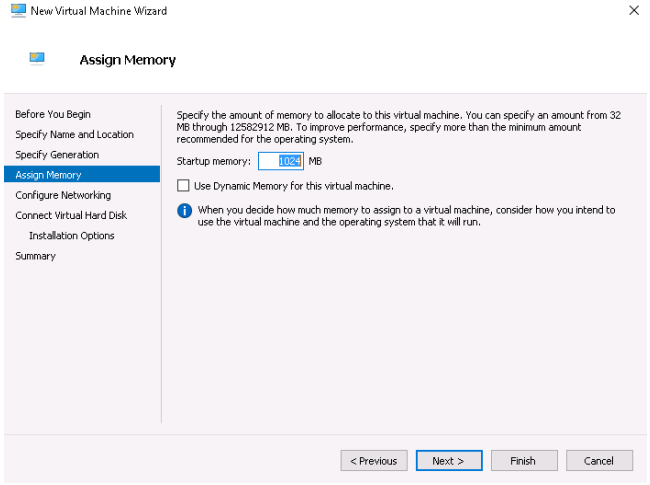
2. Click Next > and Specify Name and Storage



3. Click Next > Select "Generation 1"

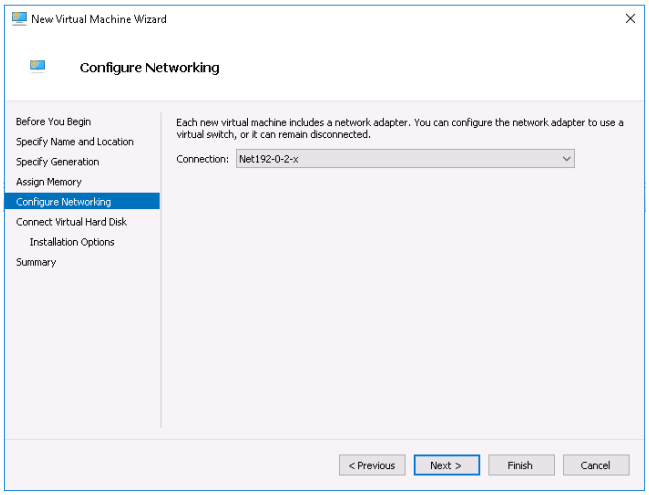


4. Click Next > Assign Memory: 4096 MB



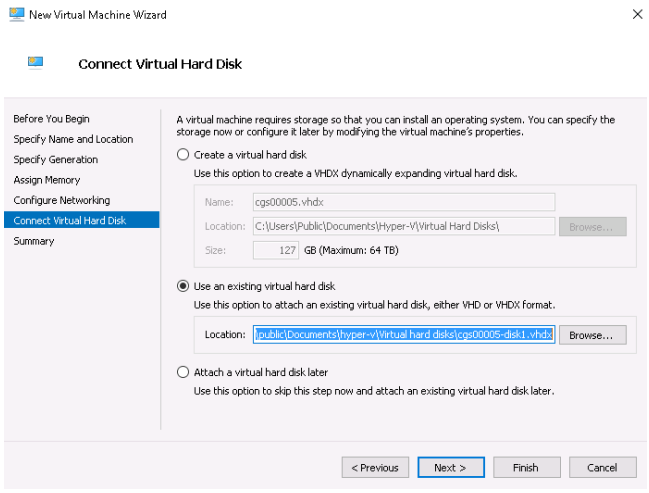
5. Click Next > Configure Networking

IMPORTANT: *This is the EXTERNAL interface of the CSC. We are going to add the Internal Interface later.*



6. Click Next > Connect Virtual Hard Disk

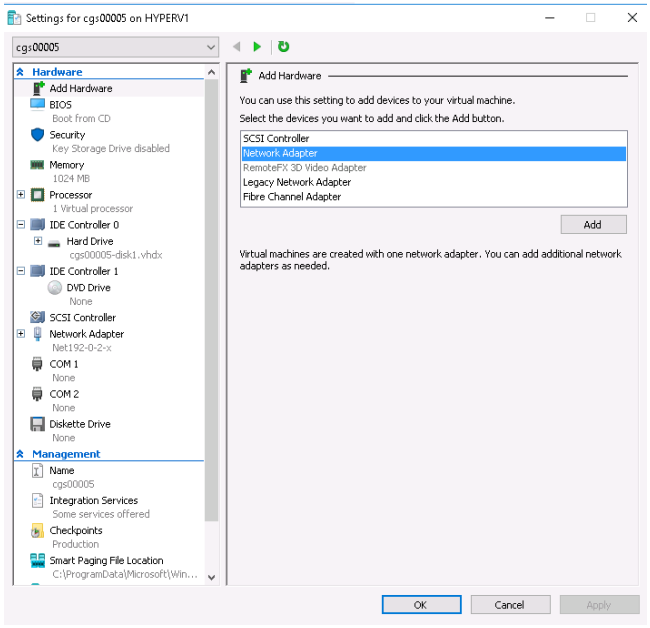
Select the unzipped disk on "Use an existing virtual disk"



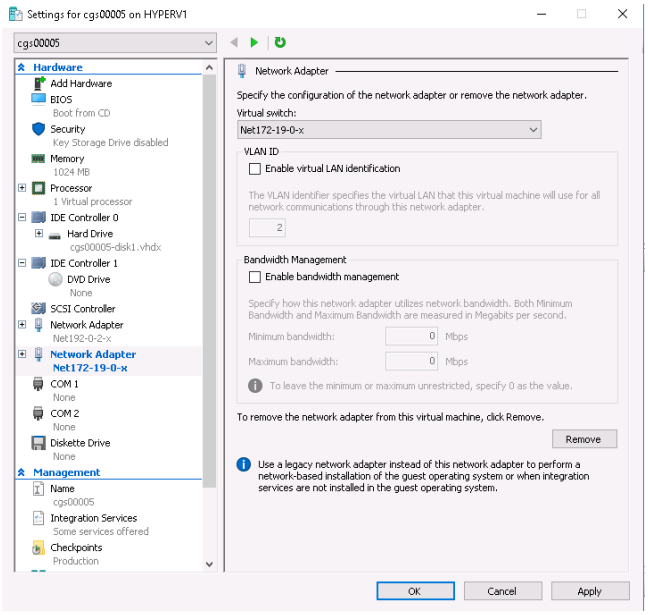
7. Click Next > Summary > Finish .

The machine will be created but we need to add the INTERNAL Interface.

8. Right Click the machine created > Settings > Add Hardware > Network Adapter



9. Click Add > and connect it to your INTERNAL virtual switch



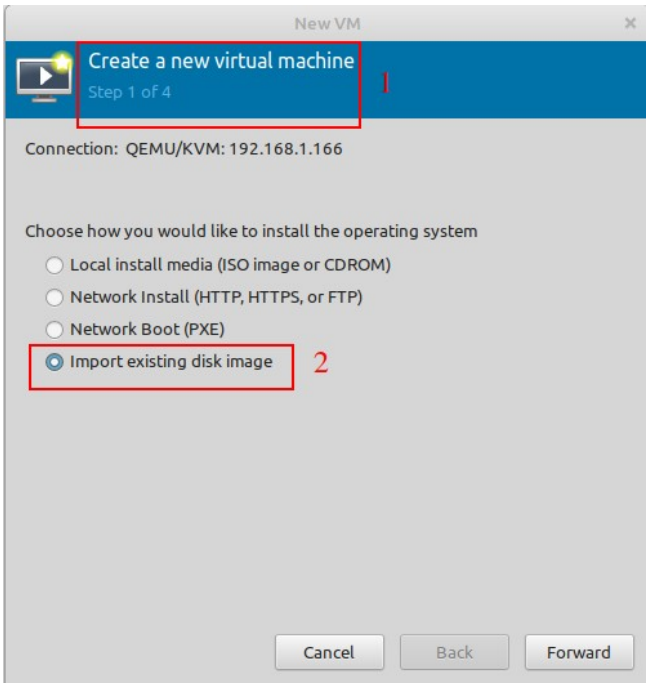
10. Click Apply and OK.

7.4 Using KVM

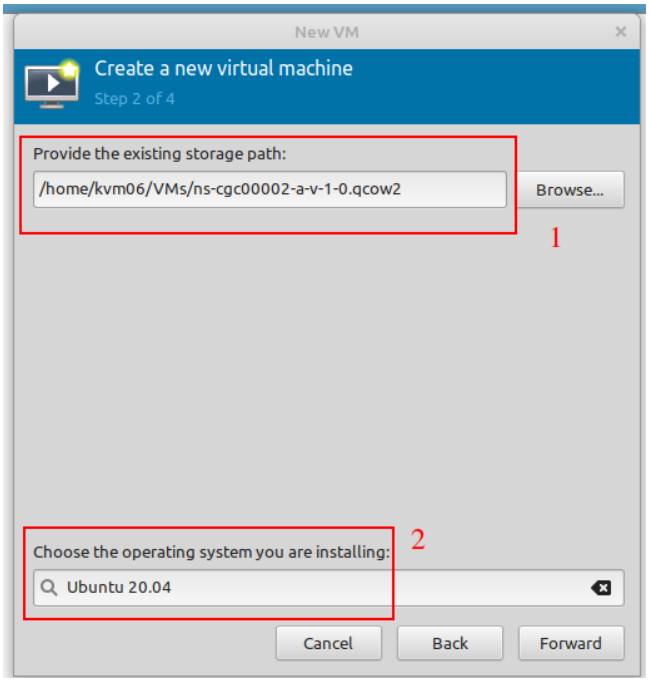
When using KVM, you will receive the disks of the CSC PriCPA Cluster in qcow2 format.

The following example shows the installation on a KVM server using Virtual Machine Manager (VMM)

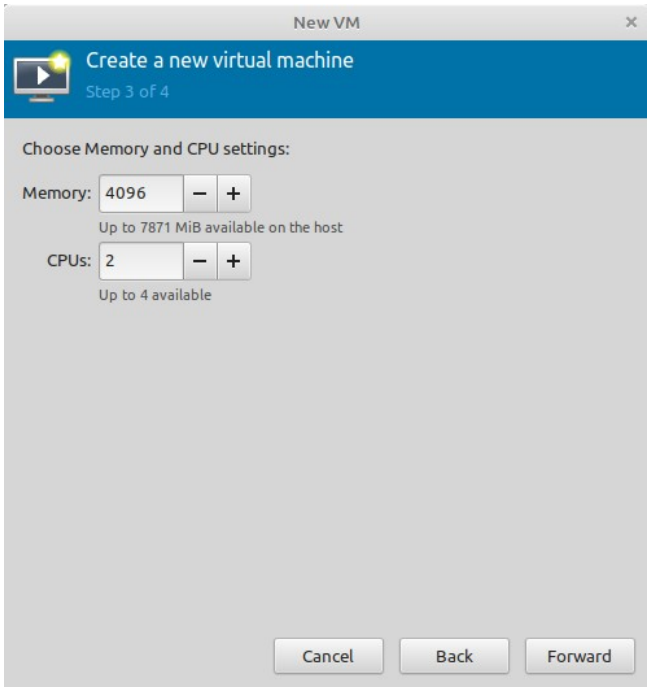
1. Go to New -> Create a new Virtual Machine and select "Import existing disk image"



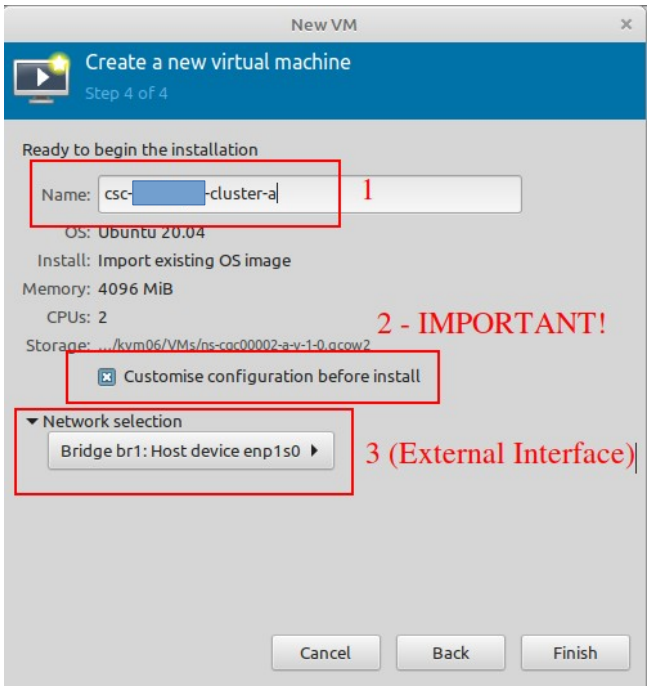
2. Click "Forward"
3. Browse for the Disk and select Ubuntu 22.04 (or another Ubuntu version if 22.04 is not available)



4. Click Forward.
5. Select 2 x CPU and 4 GB Memory (or more).

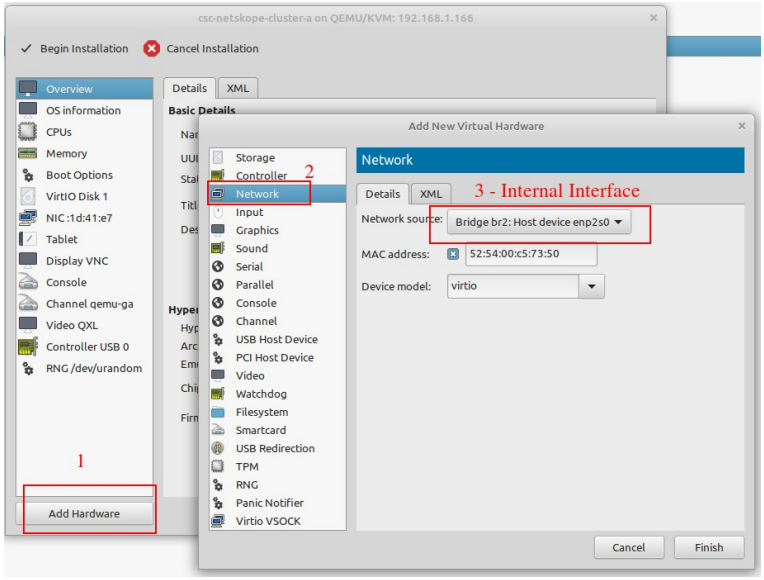


6. Click Forward.
7. Put the Name of the CSC, Select "Customise configuration before install" and choose here the External Interface.



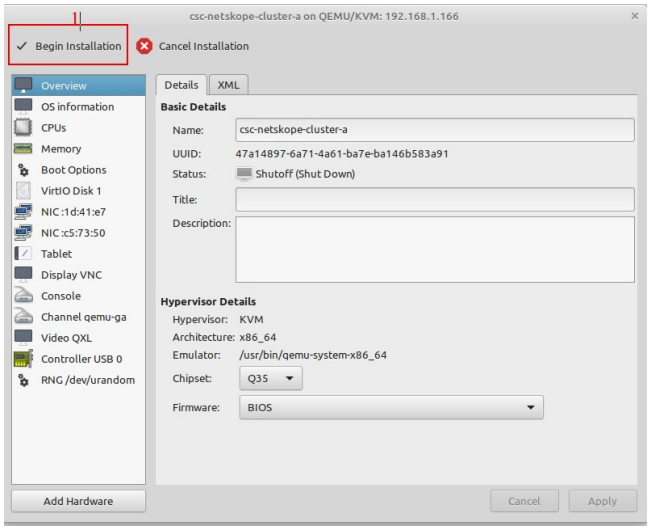
8. Click Finish.

9. We need to add now the Hardware for the Internal Interface. Click "Add Hardware", select "Network" and on Network Source choose the Internal Interface of the CSC.



10. Click Finish.

11. The last step is: "Begin Installation"



12. Done! Repeat the same process for the other CSC.

7.5 VM sizing

The CSC is a very efficient device and consumes few CPU and RAM resources. By default, we ship it with 2 x CPU, 4 x GB RAM and 16 GB disk. If you are going to have high PriCPA traffic from several Peers, please increase CPU to 4 and RAM to 8 GB.

8 Powering up the CSC GRE

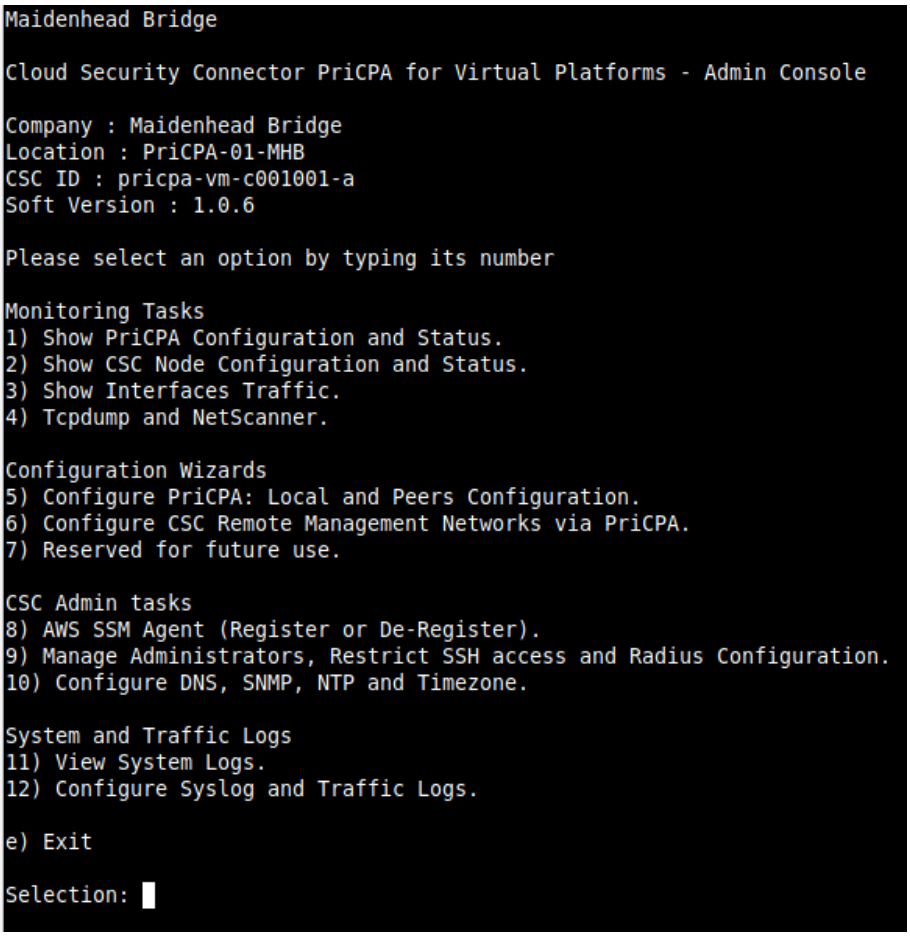
1. Power on the Virtual Machines.
2. SSH to the CSC using : ssh **cscadmin**@< CSC IP(eth1) -a > or < CSC IP(eth1) -b >. On the CSC PriCPA Cluster CSC IP(eth1) -a is the second internal IP and CSC IP(eth1) -b is the third.

When prompted, put the following username and password to login on the CSC Console:

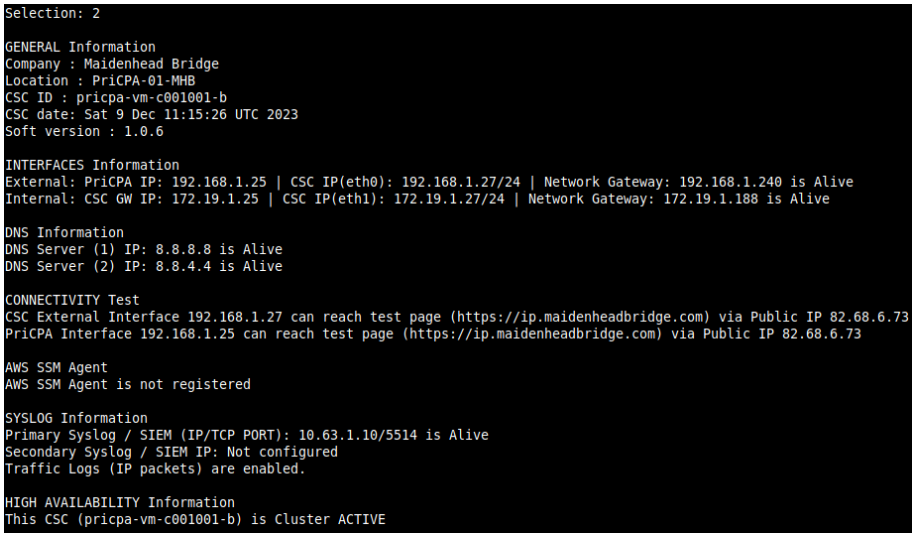
Username: **cscadmin**

Password: **maidenheadbridge**

Note: SSH to the EXTERNAL interface IPs is not allowed.



Select 1) Show CSC Node Configuration and Status and verify the IPs configurations:



Congratulations! Your CSC is up and running.

In the next chapter, we are going to discuss the PriCPA configuration in detail.

9 Configuring PriCPA

The Main Menu has a section with Configuration Wizards:

```
Configuration Wizards
5) Configure PriCPA: Local and Peers Configuration.
6) Configure CSC Remote Management Networks via PriCPA.
7) Reserved for future use.
```

In a few simple steps, you can configure PriCPA:

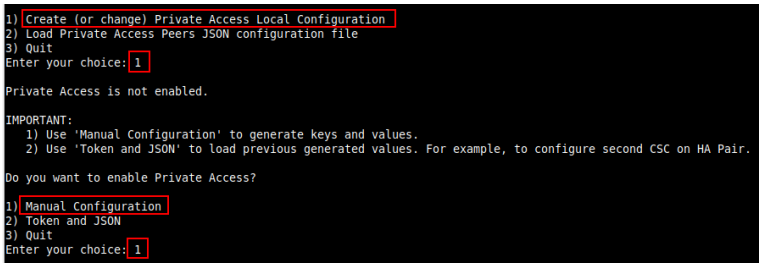
1. Create the Local Node configuration. This step will initialize and enable Private Access on the Node. The result of this operation will show a "Token" and "Private Access Local JSON file".
2. Initialize the second Node of the HA pair using the "Token" and "Private Access Local JSON file".
3. Create and distribute the Private Access Peers JSON file to all nodes.

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

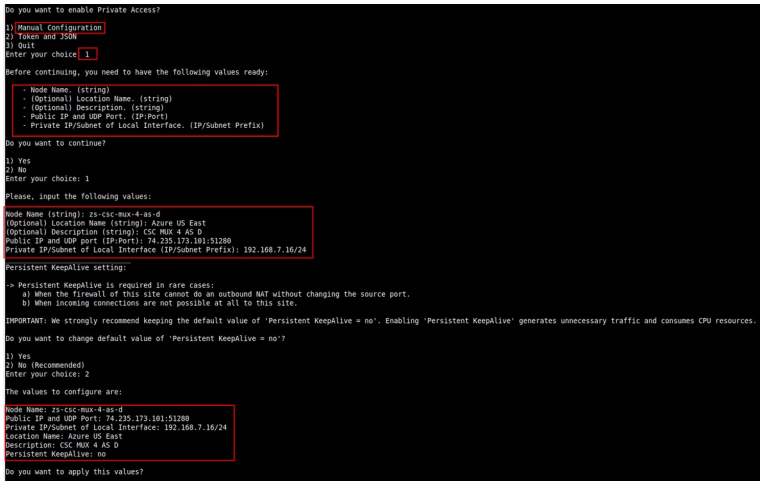
9.1 Create the Local configuration (First node of the HA pair)

```
Configuration Wizards
5) Configure PriCPA: Local and Peers Configuration.
6) Configure CSC Remote Management Networks via PriCPA.
7) Reserved for future use.
```

- From Main Menu, select "5) Configure PriCPA: Local and Peers Configuration."
- Select "1) Create (or change) Private Access Local Configuration"



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



- Apply values

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.*

9.2 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 5) Configure PriCPA: Local and Peers Configuration. → 1) Create (or change) Private Access Local Configuration → 2) Token and JSON

```
Private Access Configuration Wizard

Steps to configure Private Access:

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

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

Private Access is not enabled.

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

Do you want to enable Private Access?

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

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

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

Do you want to continue?

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

```
Do you want to continue?

1) Yes
2) No
Enter your choice: 1

Please, input the following values:

Token (string): YU9WVk9zSUdkRVNnb1UzVmZNZXZISDIvU2RYSWVpdHFpOG01aTAyU04zTT0K

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

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

Private Access Local Config JSON file: {
  "peers": [
    {
      "nodeName": "zs-csc-mux-4-as-d",
      "location": "Azure US East",
      "description": "CSC MUX 4 AS D",
      "publicKey": "4QJ7QPswdTx+mrLMbgLBube0/rw9sSunY780kljTZ1g=",
      "publicIpAndUdpPort": "74.235.173.101:51280",
      "privateCirdIp": "192.168.7.16/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}

Private Access Local Config JSON file imported successfully

The values to configure are:

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

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1

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

9.3 Create the Private Access Peers JSON file

The Private Access Peers JSON file contains:

1. The Local configuration of each Peer.
2. The "backupPublicIPs" (used when you two or more uplinks to the internet)
3. The "networks" behind each Peer.
4. The "privateApps" allowed to be reached on each Peer.

Here some examples.

9.3.1 Full mesh Private Access Peers JSON file

Consider the following example:

We have 3 nodes and we want to allow full communication between sites for all port and protocols. The Local Config JSON file of each node is:

ns-cgc00001

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

```
    }
  ]
}
```

ns-cgc00003

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

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

Basic Peers Configuration JSON file

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

```
        "backupPublicIPs": [],
        "networks": [
          "10.2.1.0/24",
          "10.2.2.0/24"
        ],
        "privateApps": []
    },
    {
        "nodeName": "ns-cgc00003",
        "description": "Node on VMware Server 3",
        "location": "Branch",
        "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
        "publicIpAndUdpPort": "200.1.1.3:51821",
        "privateCirdIp": "192.168.7.3/24",
        "persistentKeepAlive": "no",
        "backupPublicIPs": [],
        "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, and "backupPublicIPs".
- 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 allow traffic to any between subnets, we need to add the corresponding "private apps" to each node. For example, for node: "ns-cgc00001"

ns-cgc00001

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

```
        "10.1.1.0/24",
        "10.1.2.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [
            "0.0.0.0/0"
          ],
          "destinationCirdIp": [
            "10.1.1.0/24",
            "10.1.2.0/24"
          ],
          "destinationSinglePorts": [
            ""
          ],
          "destinationPortRange": {
            "fromPort": "",
            "toPort": ""
          }
        }
      ]
    },
```

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

Now, completing our "Peers Configuration JSON file":

Full Mesh Peers Configuration JSON file.

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

```
      ]
    },
    {
      "nodeName": "ns-cgc00002",
      "description": "Node on VMware Server 2",
      "location": "Datacentre 2",
      "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
      "publicIpAndUdpPort": "200.1.1.2:51821",
      "privateCirdIp": "192.168.7.2/24",
      "persistentKeepAlive": "no",
      "backupPublicIPs": []
      "networks": [
        "10.2.1.0/24",
        "10.2.2.0/24"
      ],
      "privateApps": [
        {
          "description": "Allow all traffic to this site",
          "ipProtocol": "all",
          "sourceCirdIp": [
            "0.0.0.0/0"
          ],
          "destinationCirdIp": [
            "10.2.1.0/24",
            "10.2.2.0/24"
          ],
          "destinationSinglePorts": [
            ""
          ],
          "destinationPortRange": {
            "fromPort": "",
            "toPort": ""
          }
        }
      ]
    },
    {
      "nodeName": "ns-cgc00003",
      "description": "Node on VMware Server 3",
      "location": "Branch",
      "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "backupPublicIPs": []
      "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.

9.3.2 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 fields and values accepted:

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

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

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

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.

9.3.3 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": "" }
        }
      ]
```

9.3.3.1 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": "" }
        }
      ]
```

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

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

There are three methods available:

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

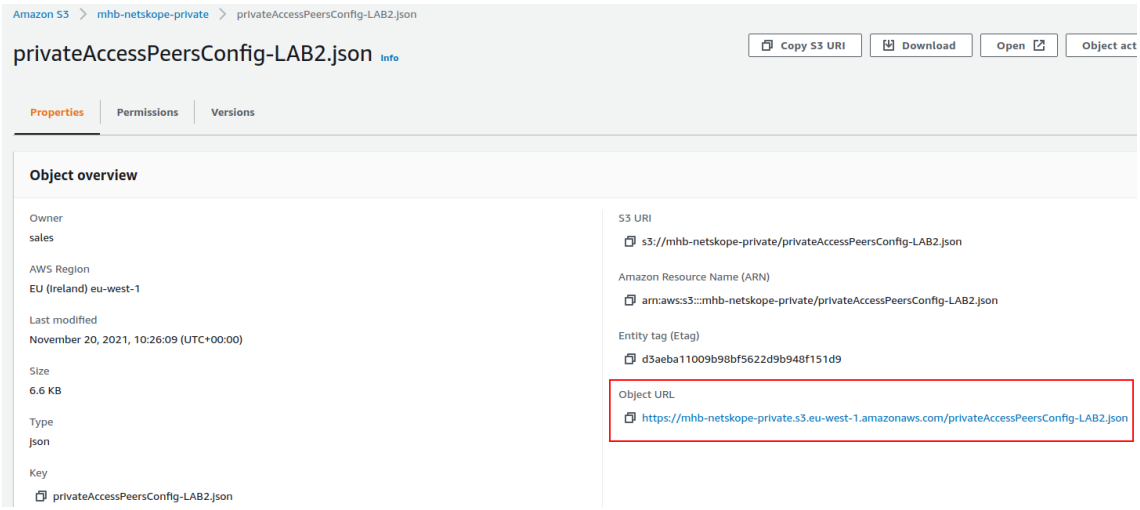
In this section we are going to explain two methods: URL and Manual Copy. The DevOps method is explained on Section12: DevOps operations.

9.4.1 Using "Private Access Peers URL"

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

1. Place the Private Access Peers JSON file on an internal web server or an AWS bucket[9] 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 -> 5) Configure PriCPA: Local and Peers Configuration.

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

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

Private Access is enabled.

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

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

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

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

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

Private Access Peers JSON file imported successfully


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

Private Apps:

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

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

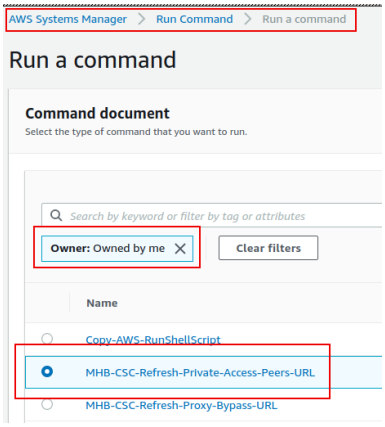
At this moment, you have the option to review the privateApps to configure in Compact or JSON format and to apply the values.

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

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

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

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

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

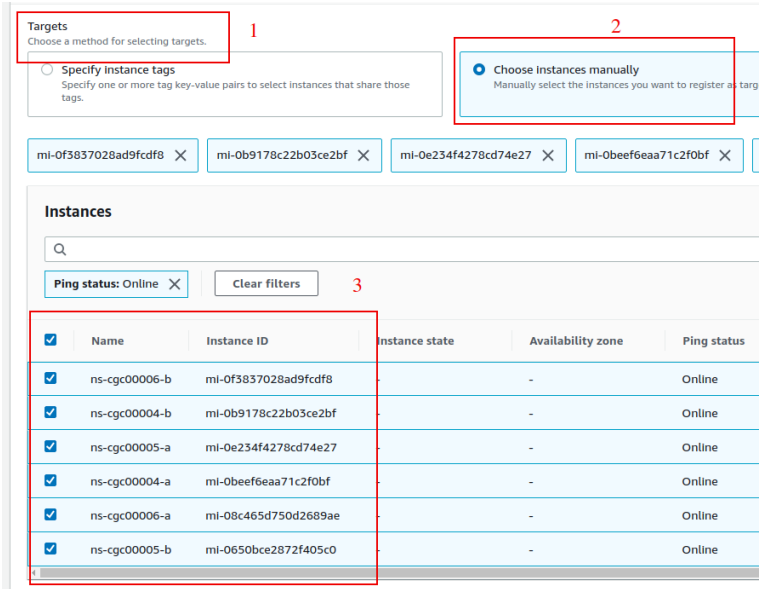
3. The next time you want to refresh the Private Access Peers JSON file, update the file, deploy it on the same location URL and Run Command: "Refresh Private Access Peers URL" using AWS SSM Agent or Rundeck.

AWS System Manager:

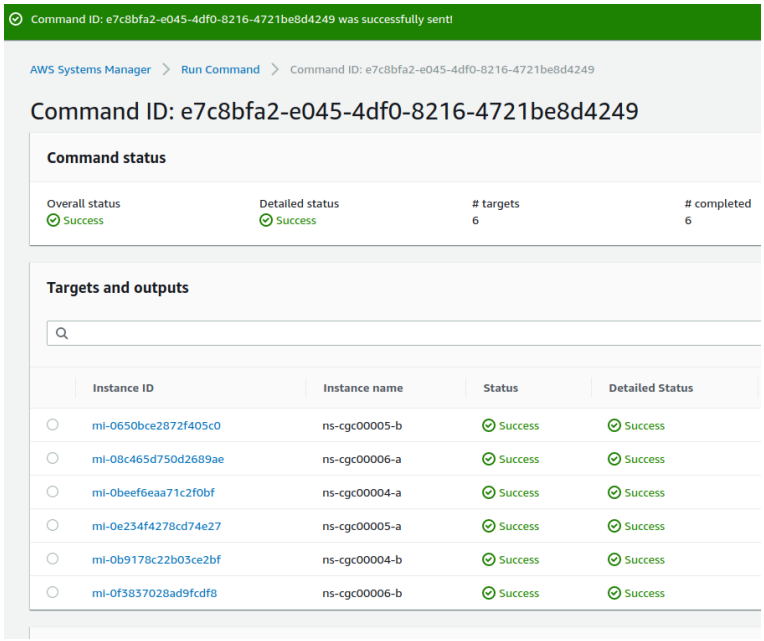
- Go to AWS Systems Manager -> Run Command -> and Select "MHB-CSC-Refresh-Private-Access-Peers-URL"



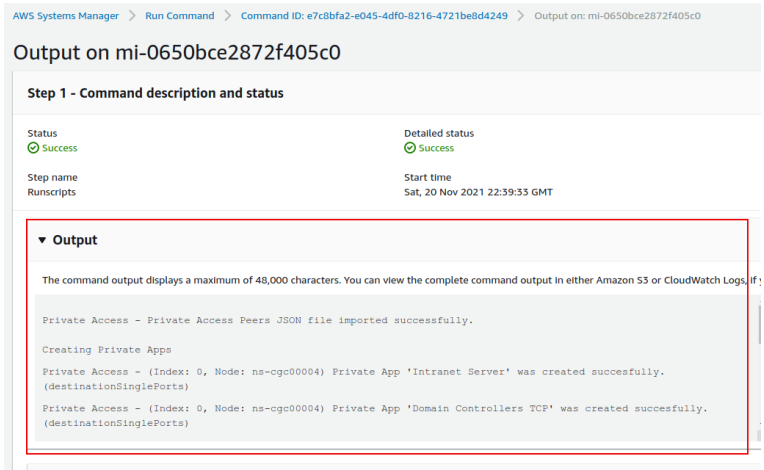
- Move down the screen and select all CSCs:



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

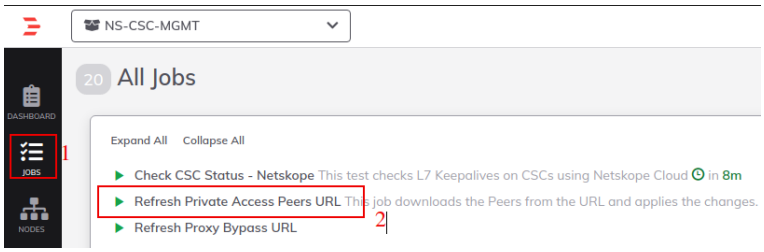


- To see the individual result, right click on the Instance ID and open it on a new TAB. Check the "Output"

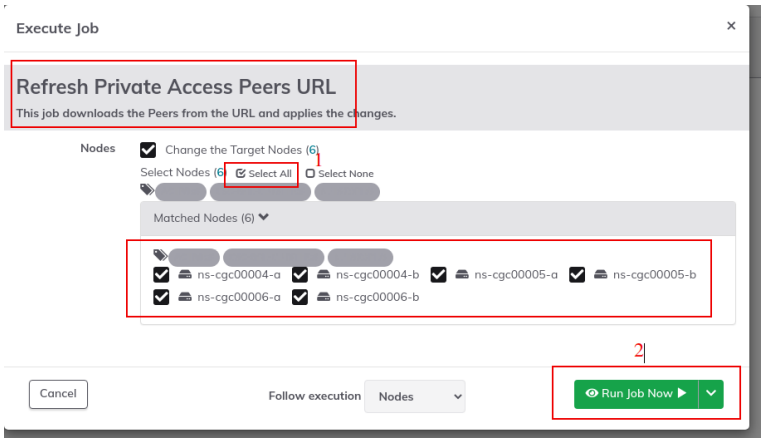


Using Rundeck

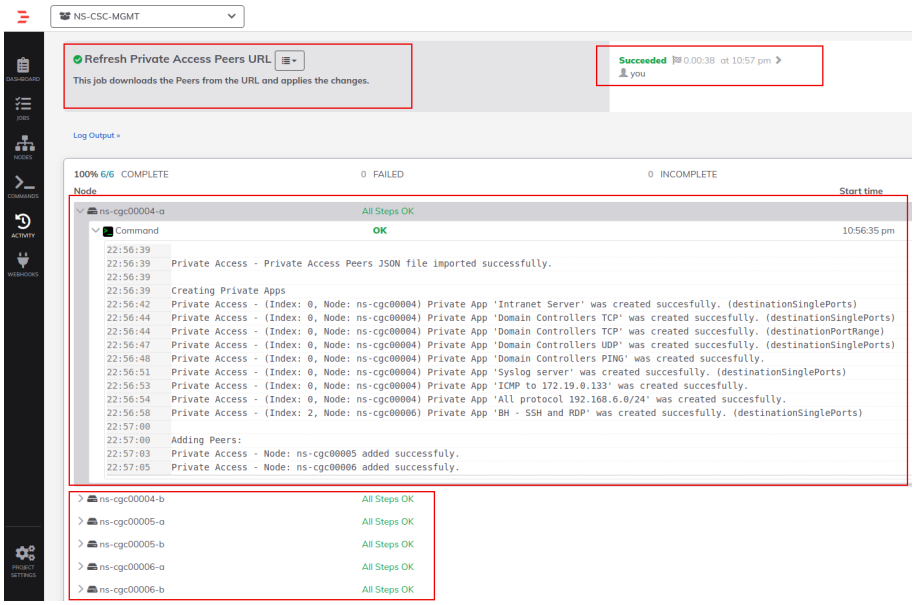
- Go to the Project <name> -> All Jobs -> Run " Refresh Private Access Peers URL"



- Select ALL nodes and click Run.



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



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

From Main Menu, go to 5) Configure PriCPA: Local and Peers Configuration., follow the steps below and Paste the Private Access Peers Json File:

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

Private Access is enabled.

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


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

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

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

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

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

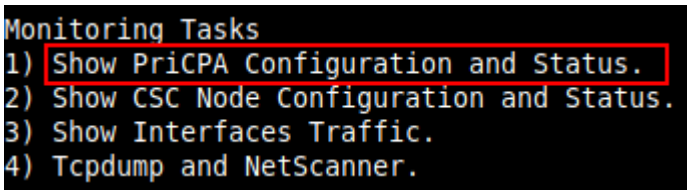
```
Private Access Peers JSON file imported successfully


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

Private Apps:

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

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

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

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

Changes Security Group External:
Private Access - Inbound Port Rules 'mhb-csc-private-access-151280' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-2'
Private Access - Outbound Port Rules 'mhb-csc-private-access-051820, mhb-csc-private-access-051821' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-2'

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

Done!

10 Show Configurations and Status Private Access.

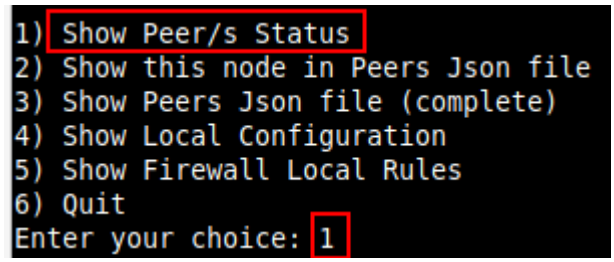
10.1 Using SSH Admin console

From Main Menu, go to 1) Show PriCPA Configuration and Status.

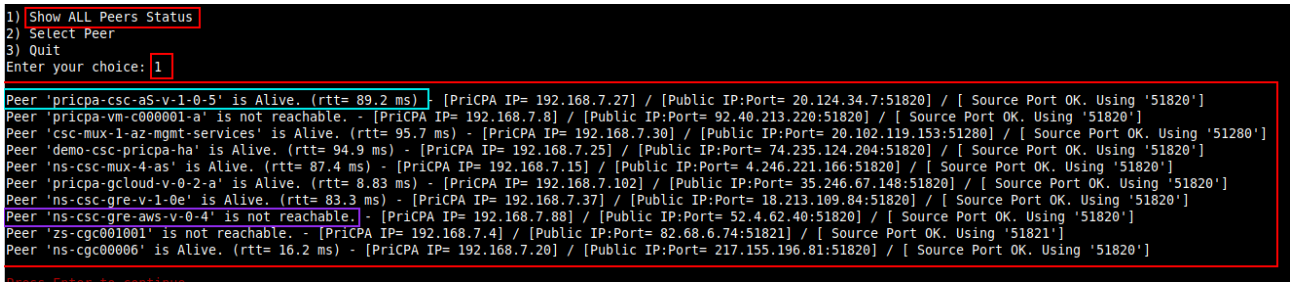


10.1.1 Show Peer/s Status

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



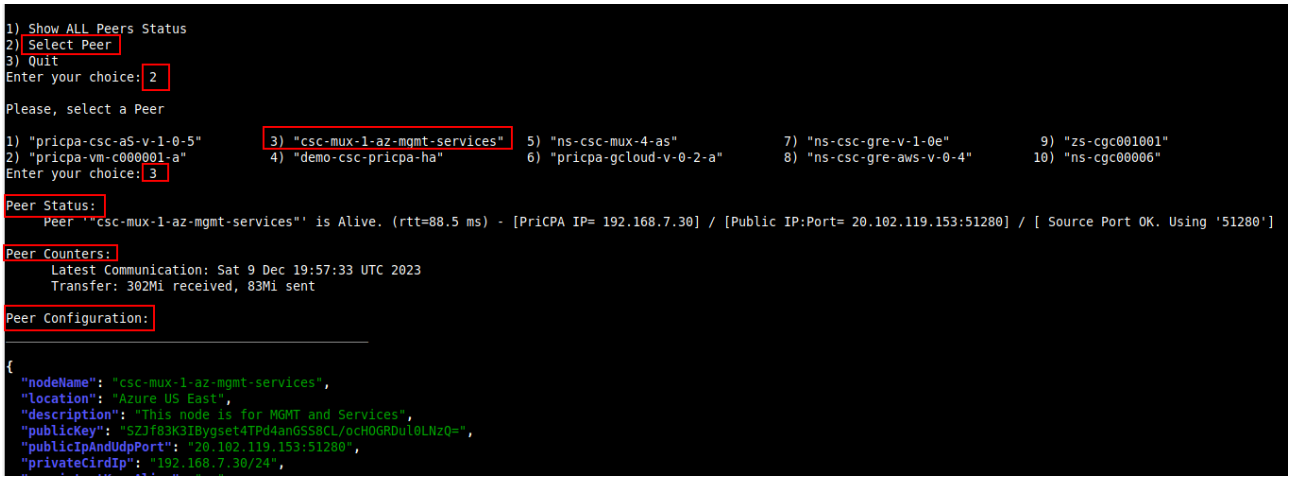
1. Show All Peers Status



IMPORTANT: This section shows the Peer is Alive, the Round Trip Time (RTT) to the Peer, the PricPA IP and Public IP of the Peer 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 the Source Port differs from the one expected. Also, consider enabling KeepAlives on peers where the FW changes the Source Port.
It is unusual to see "Source Port was changed" when the CSC is on Public Clouds (Azure, AWS or Gcloud), but it often happens when the CSC is On-Prem behind a traditional FW.

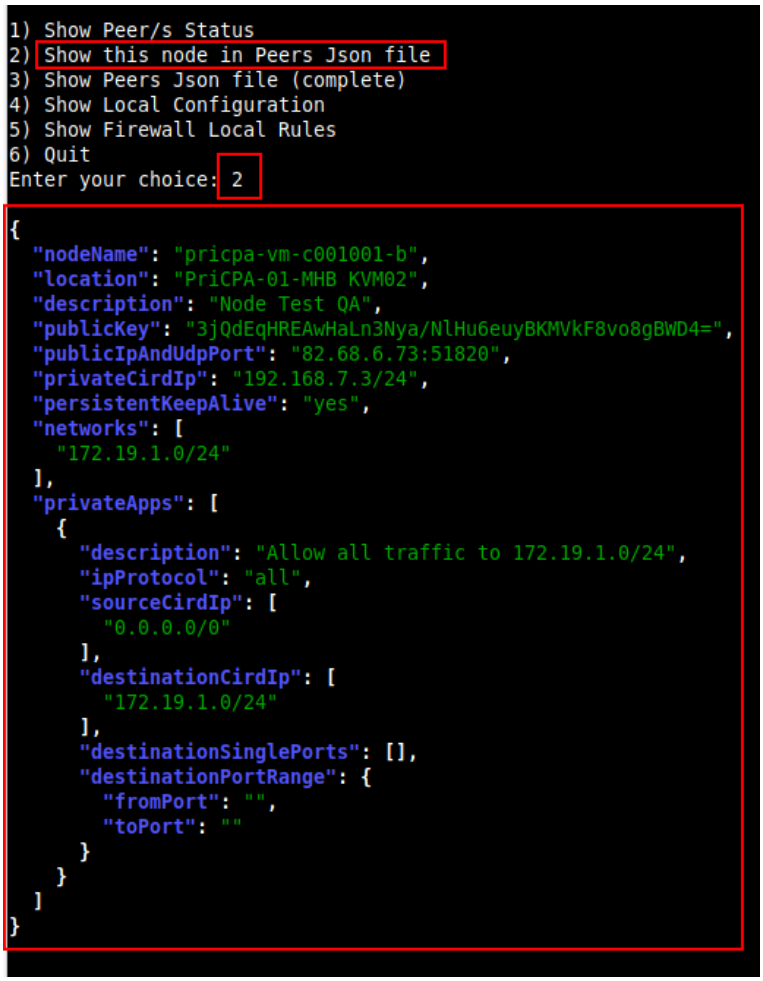
2. Select Peer

This section shows a more detailed information about the Peer.



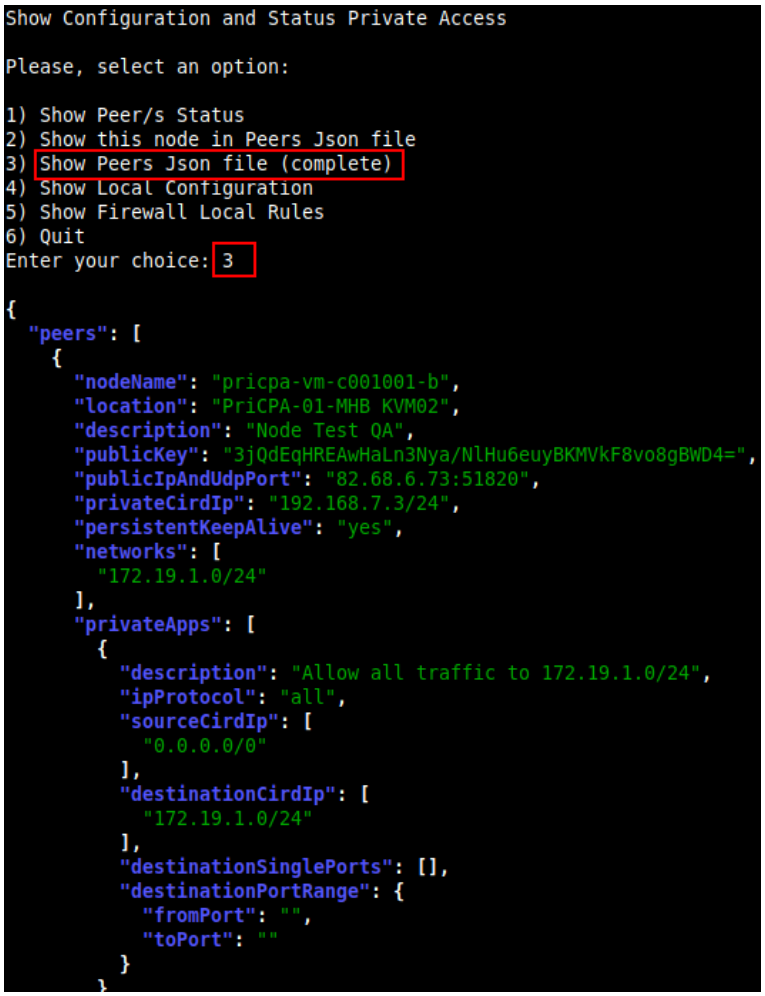
10.1.2 Show this node in Peers Json file

This menu shows the configuration of the local peer on the Peers Json file.



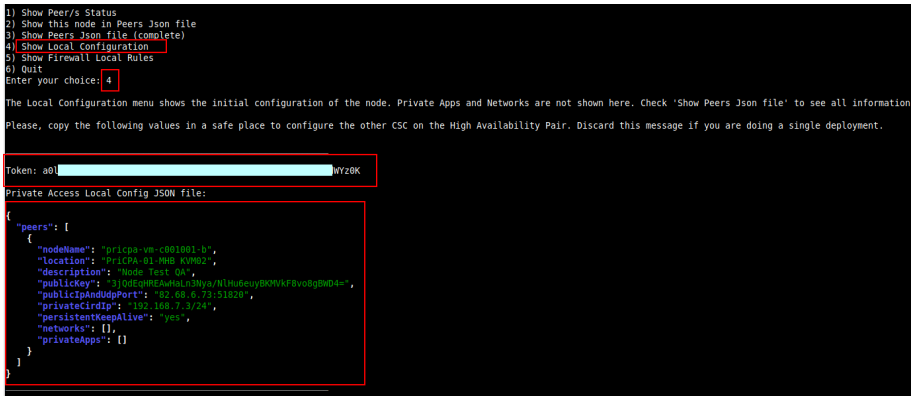
10.1.3 Show Peers Json file (complete)

This menu shows the active Private Access Peers Json file.



10.1.4 Show Local Configuration

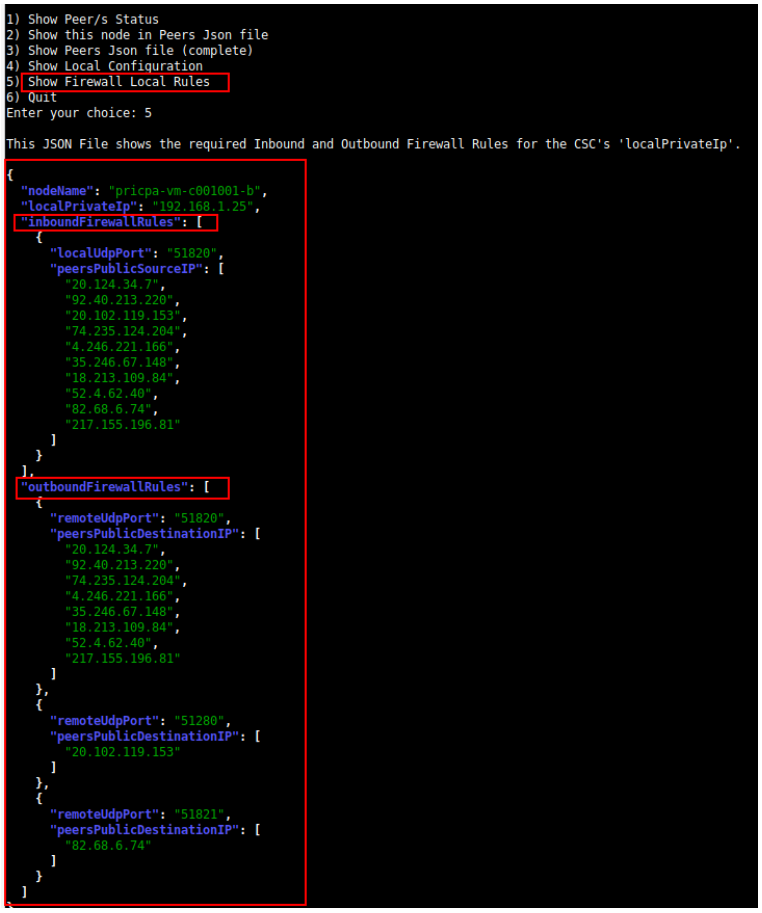
This menu shows the Local configuration of the node.



10.1.5 Show Firewall Local Rules

This menu shows in JSON format the Rules required on your Firewall for the PriCPA cloud.

> ***Note:*** *The CSC on Public Cloud (Azure, AWS, Gcloud) does the refresh of the External Security Group every time you update the Peers configuration. No manual configuration is required. When the CSC is "On-Prem" (VMware, Hyper-V, KVM) use this information to configure your firewall.*



10.2 Using AWS Systems Manager or Rundeck

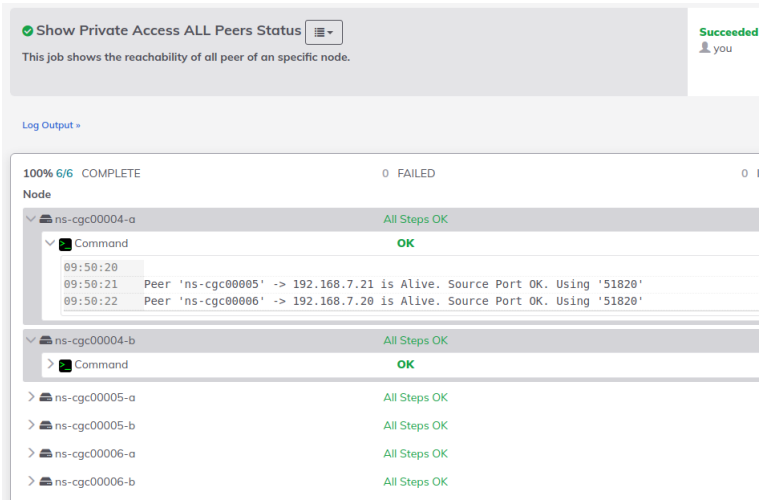
In this case, the information provided is only "Show ALL Peer Status"

10.2.1 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:

10.2.2 Rundeck

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

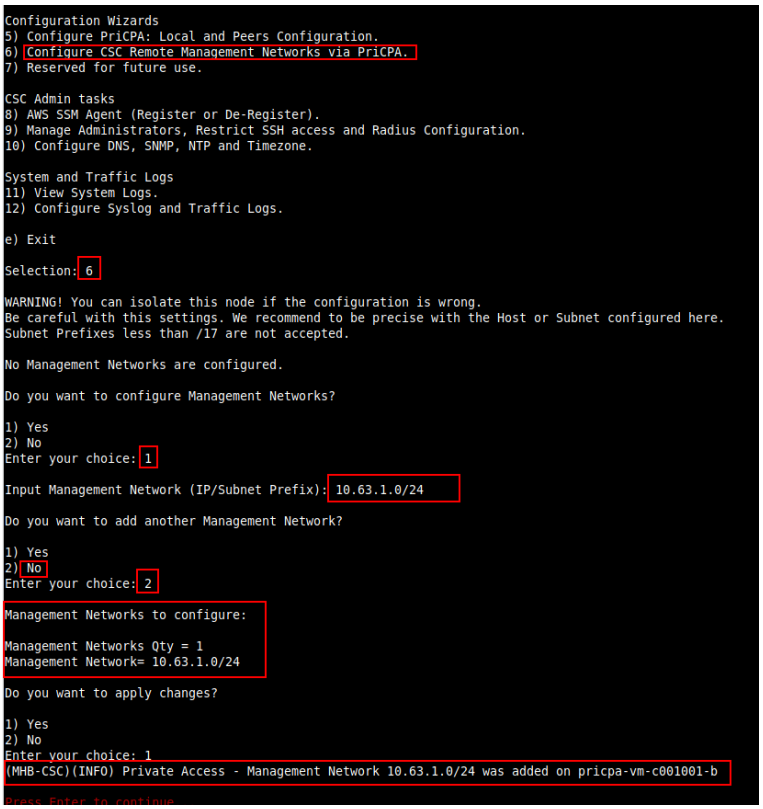


11 Configure CSC Remote Management via Private Access.

Only the active node belongs to the Private Cloud when the CSC is in the HA pair. For this reason, if you want to reach "the Other CSC" node using SSH via PriCPA (wg0 interface), 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 data centre, you have the Subnet 10.63.1.0/24, and from that Subnet, you want to reach ALL CSCs on the Private Cloud.

The configuration will be:

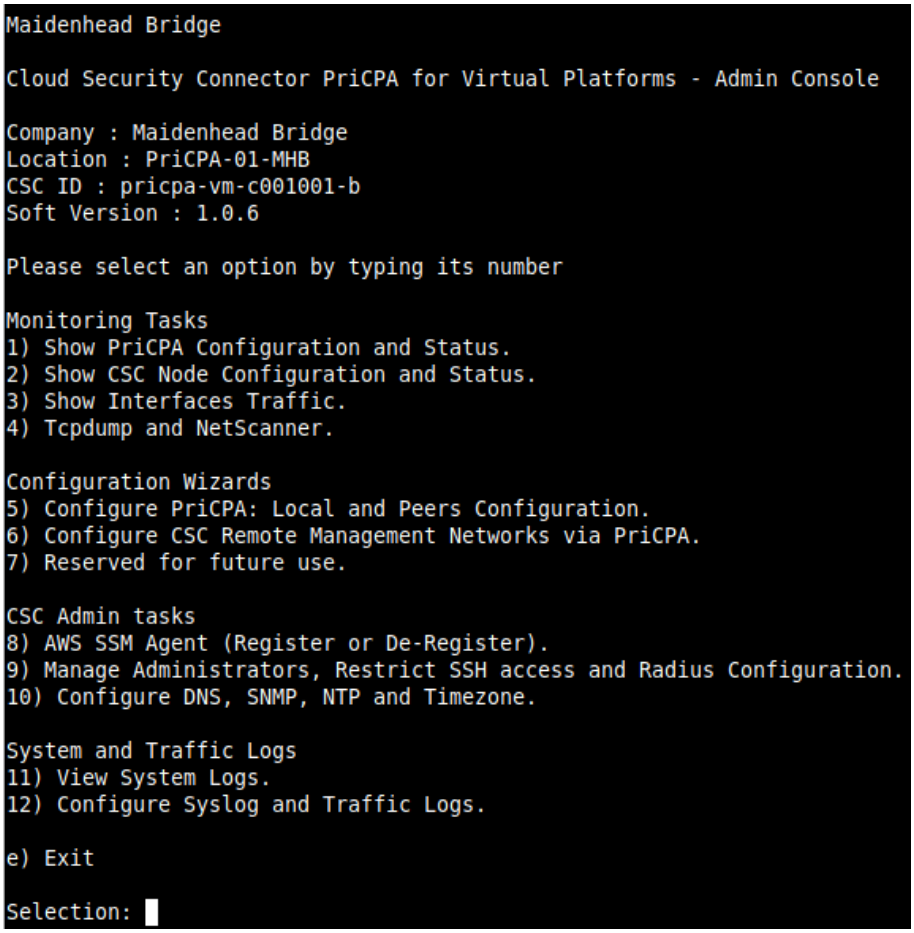


Important: This setting is unnecessary if you can reach the CSC from the management network via eth1 of the CSC. Enable Remote Management only when you need to get the CSC via PriCPA (interface wg0)

12 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.



The Main Sections are:

- **Monitoring Tasks:** To check configuration, statuses, real-time traffic, run tcpdump and scan the local network.
- **Configuration Wizards:** Configure PriCPA and Remote Management Networks.
- **CSC Admin Tasks:** To register the CSC for AWS management, manage administrators, restrict SSH, configure radius, DNS Servers, SNMP, NTP and Timezone.
- **System and Traffic Logs:** Shows Systems logs, configure Syslog Servers and enable/disable traffic logs.

12.1 Monitoring Tasks

12.1.1 Show PriCPA Configuration and Status.

See previous section for detailed information.

```
Show Configuration and Status Private Access

Please, select an option:

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

12.1.2 Show CSC Node Configuration and Status.

```
GENERAL Information
Company : Maidenhead Bridge
Location : PriCPA-01-MHB
CSC ID : pricpa-vm-c001001-b
CSC date: Sat 9 Dec 21:54:17 UTC 2023
Soft version : 1.0.6

INTERFACES Information
External: PriCPA IP: 192.168.1.25 | CSC IP(eth0): 192.168.1.27/24 | Network Gateway: 192.168.1.240 is Alive
Internal: CSC GW IP: 172.19.1.25 | CSC IP(eth1): 172.19.1.27/24 | Network Gateway: 172.19.1.188 is Alive

DNS Information
DNS Server (1) IP: 8.8.8.8 is Alive
DNS Server (2) IP: 8.8.4.4 is Alive

CONNECTIVITY Test
CSC External Interface 192.168.1.27 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.73
PriCPA Interface 192.168.1.25 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.73

AWS SSM Agent
AWS SSM Agent is not registered

SYSLOG Information
Primary Syslog / SIEM (IP/TCP PORT): 10.63.1.10/5514 is Alive
Secondary Syslog / SIEM IP: Not configured
Traffic Logs (IP packets) are enabled.

HIGH AVAILABILITY Information
This CSC (pricpa-vm-c001001-b) is Cluster ACTIVE
```

12.1.2.1 GENERAL Information

This section contains general information about the instance:

```
GENERAL Information
Company : Maidenhead Bridge
Location : PriCPA-01-MHB
CSC ID : pricpa-vm-c001001-b
CSC date: Sat 9 Dec 21:54:17 UTC 2023
Soft version : 1.0.6
```

12.1.2.2 INTERFACES Information

This section contains the interfaces information:

```
INTERFACES Information
External: PriCPA IP: 192.168.1.25 | CSC IP(eth0): 192.168.1.27/24 | Network Gateway: 192.168.1.240 is Alive
Internal: CSC GW IP: 172.19.1.25 | CSC IP(eth1): 172.19.1.27/24 | Network Gateway: 172.19.1.188 is Alive
```

12.1.2.3 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
DNS Server (1) IP: 8.8.8.8 is Alive
DNS Server (2) IP: 8.8.4.4 is Alive
```

12.1.2.4 CONNECTIVITY Test

This test checks connectivity to Internet via the CSC IP (eth0) and via the PriCPA IP. In both cases, the interfaces are testing against the site https://ip.maidenheadbridge.com

```
CONNECTIVITY Test
CSC External Interface 192.168.1.27 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.73
PriCPA Interface 192.168.1.25 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.73
```

12.1.2.5 AWS SSM Agent

This section shows the status of the AWS SSM Agent.

```
AWS SSM AGENT
AWS SSM Agent is active (running) since Fri 2023-08-04 15:35:30 UTC; 24h ago
Registration values: {"ManagedInstanceID":"mi-02f61b13176c35082","Region":"eu-west-2"}
```

12.1.2.6 SYSLOG Information

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

```
SYSLOG Information
Primary Syslog / SIEM (IP/TCP PORT): 10.63.1.10/5514 is Alive
Secondary Syslog / SIEM IP: Not configured
Traffic Logs (IP packets) are enabled.
```

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

SystemLogs:

- UP
- DOWN
- INFO
- ALERT
- ERROR

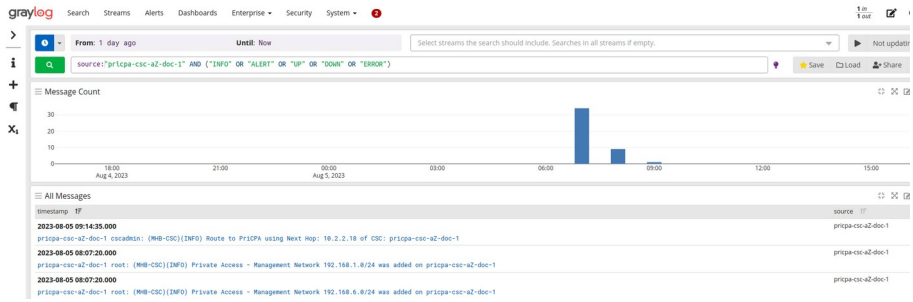
Traffic Logs:

- ALLOW
- BLOCK

12.1.2.6.1 System Logs example:

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

Using GrayLog Server: source: "pricpa-csc-aZ-doc-1" AND ("INFO" OR "ALERT" OR "UP" OR "DOWN" OR "ERROR")



12.1.2.6.2 Traffic Logs example:

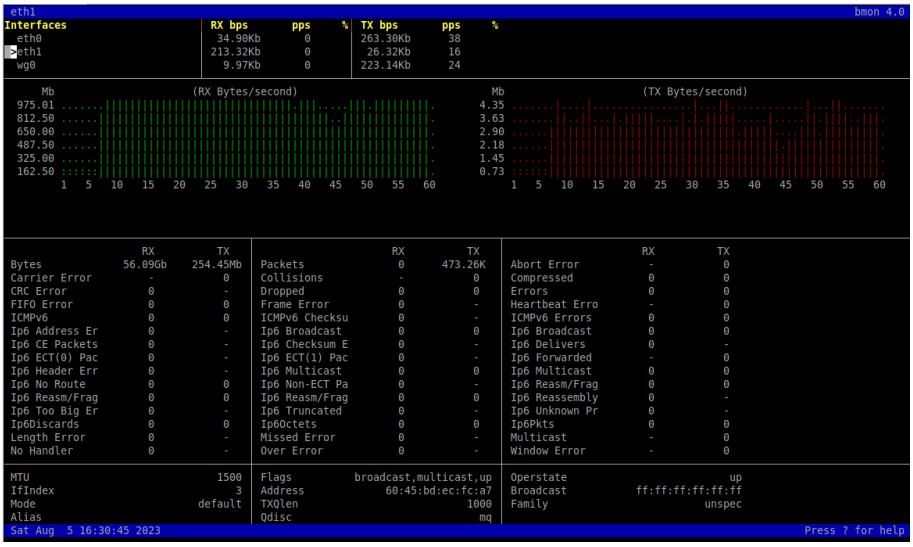
Using GrayLog Server: source:"pricpa-csc-aZ-doc-1" AND ("ALLOW" OR "BLOCK")

12.1.2.7 HIGH AVAILABILITY Information

```
HIGH AVAILABILITY Information
This CSC (pricpa-vm-c001001-b) is Cluster ACTIVE
```

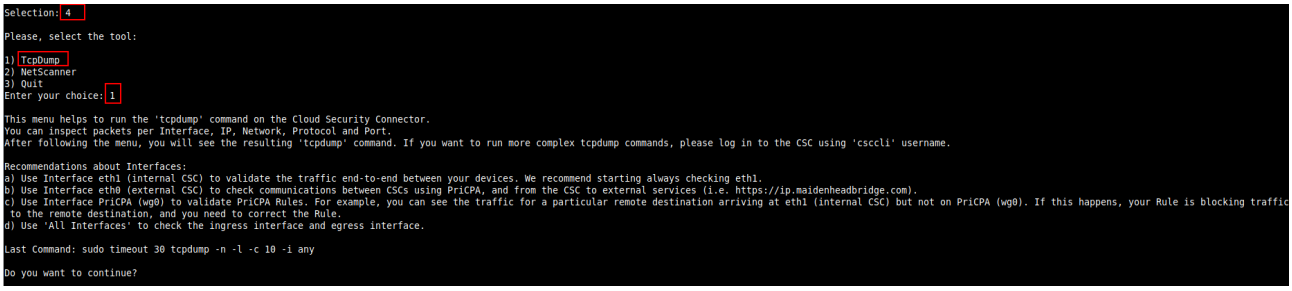
12.1.3 Show Interfaces Traffic

Use this section to see the traffic in real time.



12.1.4 Tcpdump.

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



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

- Select the options:

```
Do you want to continue?

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

Please select the Interface.

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

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

1) Host
2) Net
3) Source/Destination IPs
4) Any
5) Quit
Enter your choice: 1
Host (IP): 10.2.3.5

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

1) TCP
2) UDP
3) ICMP
4) Any
5) Quit
Enter your choice: 4

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

Do you want to change default values?

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

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

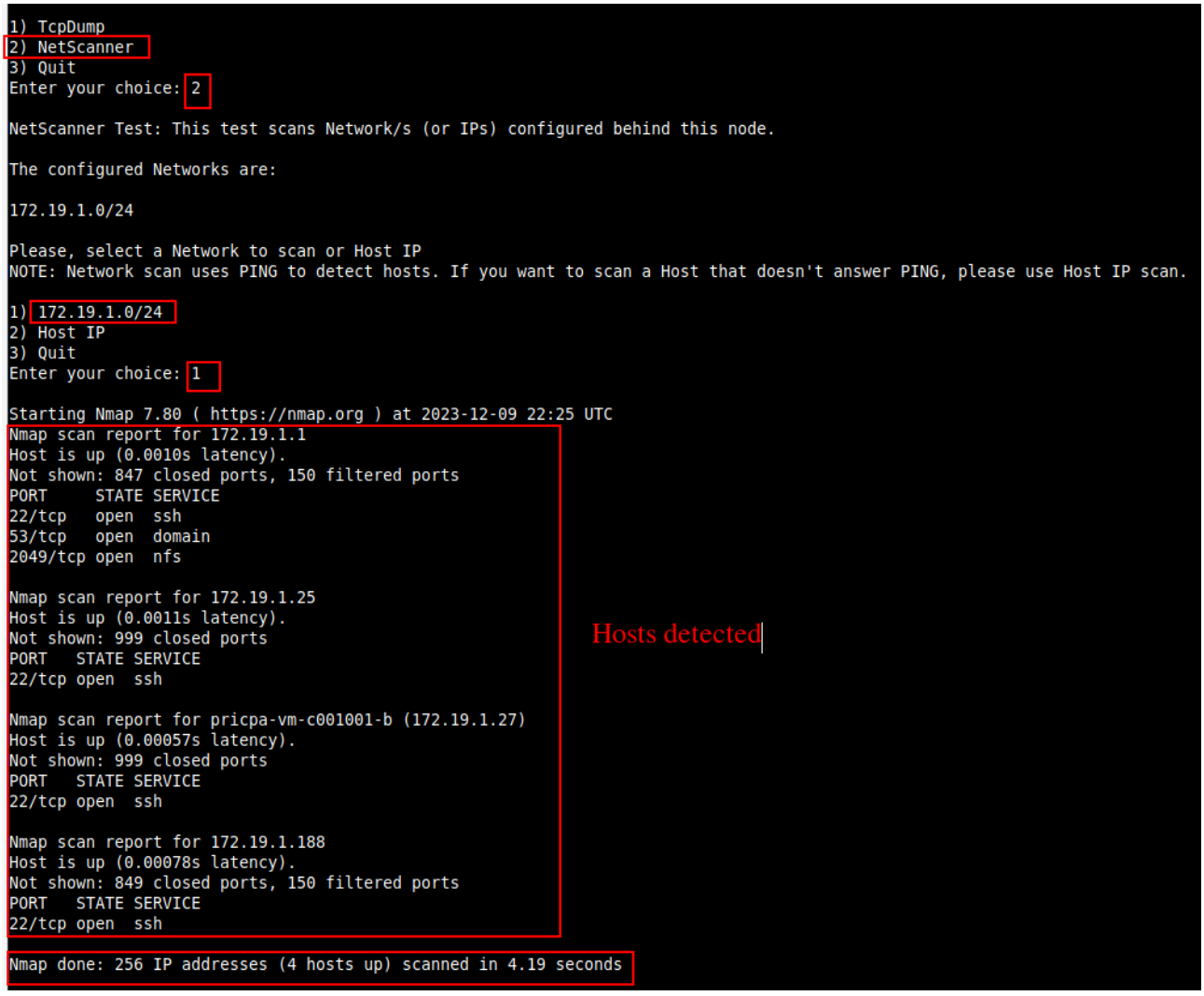
```
1) Yes
2) No
3) Quit
Enter your choice: 2

COMMAND: sudo timeout 30 tcpdump -n -l -c 10 -i eth1 host 10.2.3.5

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:36:49.200176 IP 172.19.0.140.37348 > 10.2.3.5.22: Flags [P.], seq 4032678098:4032678134, ack 176467284, win 501, options [nop,nop,TS val 1249869137 ecr 3019800591], length 36
16:36:49.201643 IP 10.2.3.5.22 > 172.19.0.140.37348: Flags [.], ack 36, win 502, options [nop,nop,TS val 3019809623 ecr 1249869137], length 0
16:36:49.202849 IP 10.2.3.5.22 > 172.19.0.140.37348: Flags [P.], seq 1:37, ack 36, win 502, options [nop,nop,TS val 3019809624 ecr 1249869137], length 36
16:36:49.342351 IP 172.19.0.140.37348 > 10.2.3.5.22: Flags [.], ack 37, win 501, options [nop,nop,TS val 1249869279 ecr 3019809624], length 0
16:36:50.136098 IP 172.19.0.140.37348 > 10.2.3.5.22: Flags [P.], seq 36:72, ack 37, win 501, options [nop,nop,TS val 1249870073 ecr 3019809624], length 36
16:36:50.138242 IP 10.2.3.5.22 > 172.19.0.140.37348: Flags [P.], seq 37:73, ack 72, win 502, options [nop,nop,TS val 3019810560 ecr 1249870073], length 36
16:36:50.228635 IP 172.19.0.140.37348 > 10.2.3.5.22: Flags [.], ack 73, win 501, options [nop,nop,TS val 1249870166 ecr 3019810560], length 0
16:36:50.279751 IP 172.19.0.140.37348 > 10.2.3.5.22: Flags [P.], seq 72:108, ack 73, win 501, options [nop,nop,TS val 1249870217 ecr 3019810560], length 36
16:36:50.282148 IP 10.2.3.5.22 > 172.19.0.140.37348: Flags [P.], seq 73:109, ack 108, win 502, options [nop,nop,TS val 3019810703 ecr 1249870217], length 36
16:36:50.373270 IP 172.19.0.140.37348 > 10.2.3.5.22: Flags [P.], seq 108:144, ack 109, win 501, options [nop,nop,TS val 1249870311 ecr 3019810703], length 36
10 packets captured
11 packets received by filter
0 packets dropped by kernel
```

12.1.5 Netscanner

The Netscanner helps to discover and scan the devices connected to the defined "subnets behind this node" on the Peers Configuration File.

The Netscanner can scan the entire subnet with a generic scan or check a single host more profoundly.

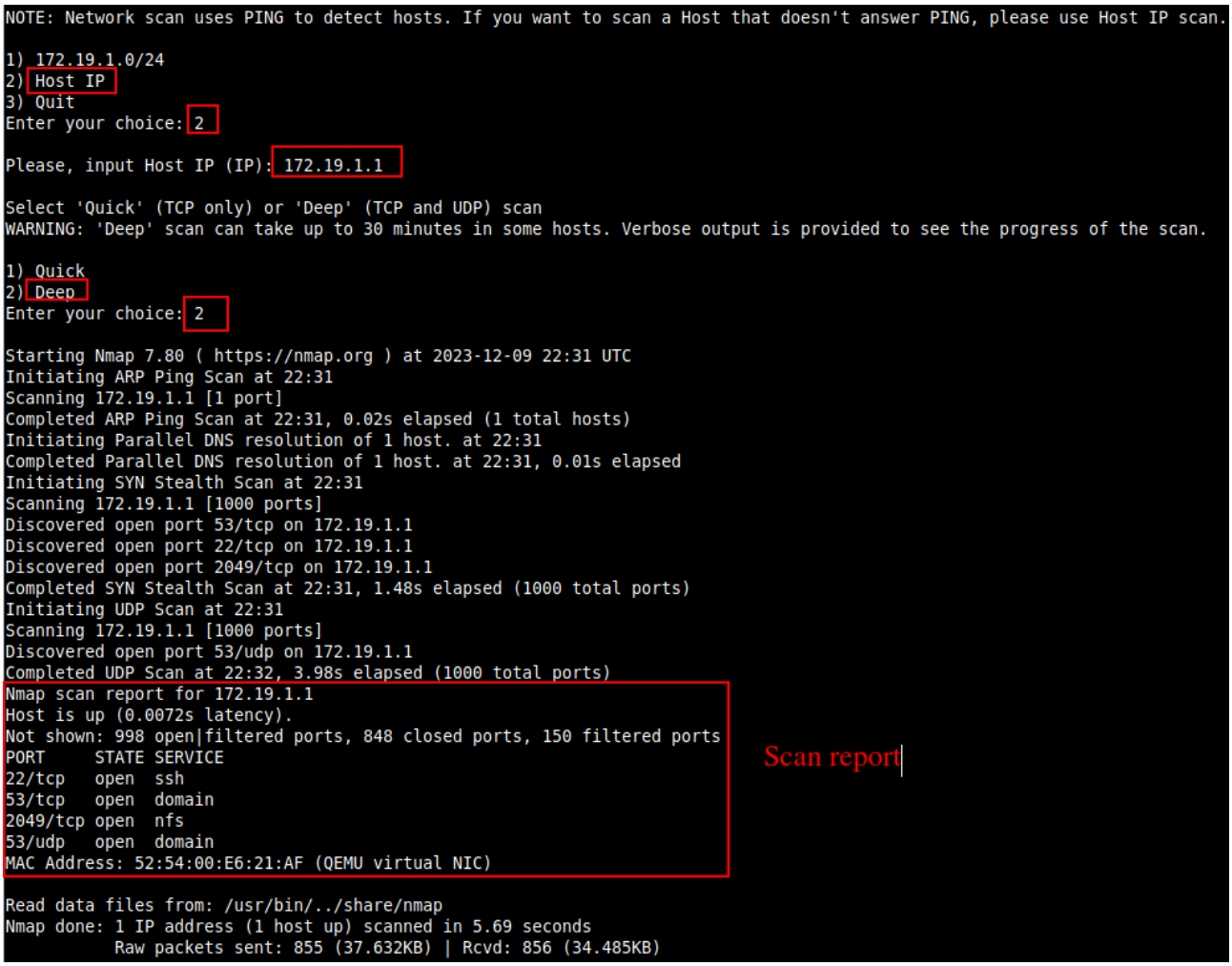
12.1.5.1 Scanning a entire Subnet



12.1.5.2 Scanning a Host IP

In this section you can do a Quick or Deep scan of host IP.

Note: Run Deep scan if you want to detect UDP ports.



12.2 Configuration Wizards

```
Configuration Wizards
5) Configure PriCPA: Local and Peers Configuration.
6) Configure CSC Remote Management Networks via PriCPA.
7) Reserved for future use.
```

Please, see previous chapter for detailed information about "Configuration Wizards"

12.3 CSC Admin Tasks

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

12.3.1 AWS SSM Agent (Register or De-Register)

The CSC AWS has installed the AWS SSM Agent that allows you to check remotely the status of the CSC and “Run Commands” using AWS Systems Manager. You can manage all CSCs models[10] using AWS Systems Manager.

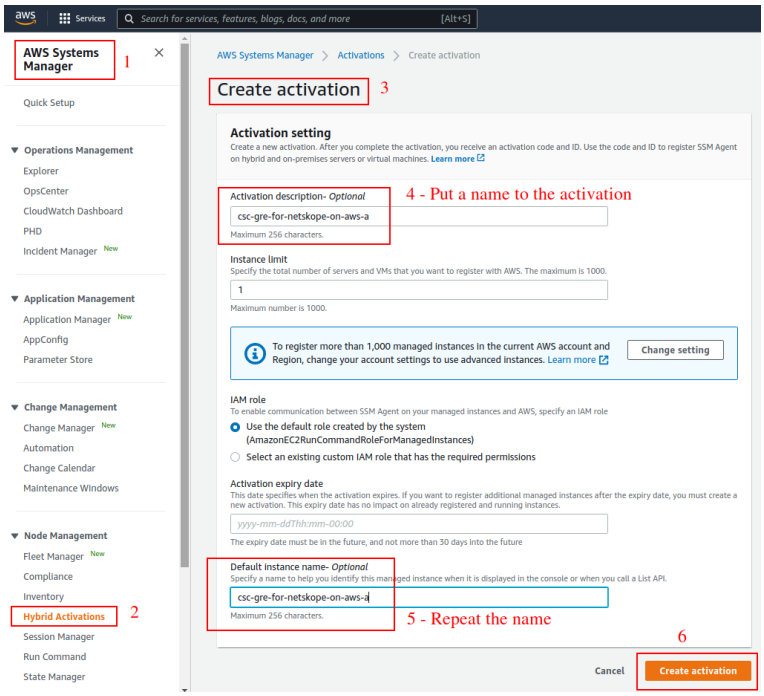
***Note**: You can learn more about “Run Commands” on Appendix B*

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

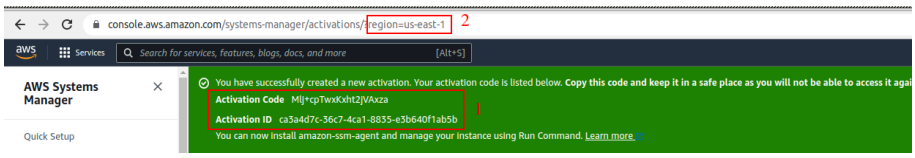
12.3.1.1 Create a "Hybrid Activation" from AWS console.

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

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

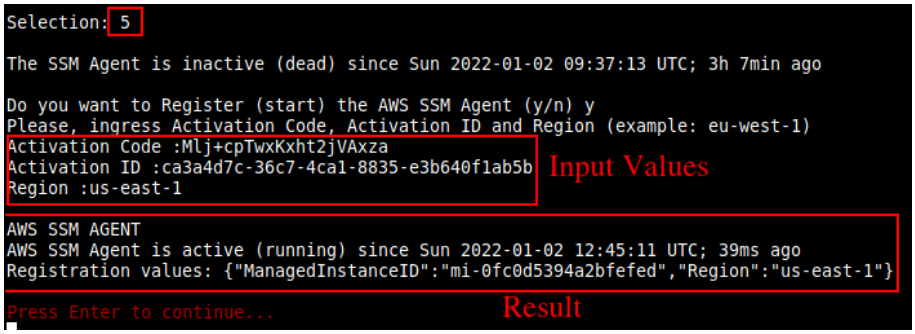


→ Click "Create Activation"

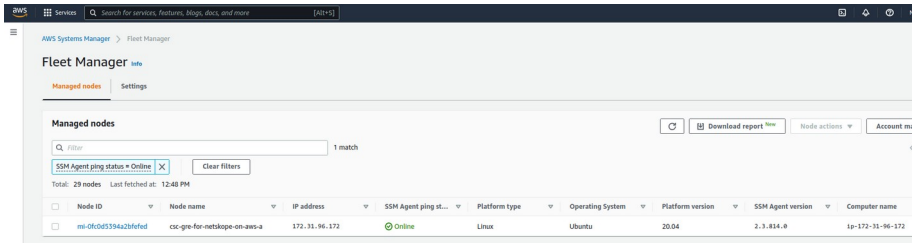


The values of Actication Code, Activation ID and Region are required to register the CSC. Keep this values on a safe place.

12.3.1.2 Register the CSC



12.3.1.3 View the Registered CSC on AWS Systems Manager



12.3.2 Manage Administrators, Restrict SSH access and Radius Configuration

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

```
Selection: 9

Please, select the task to do:

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

12.3.2.1 Manage Administrators: cscadmin and csccli

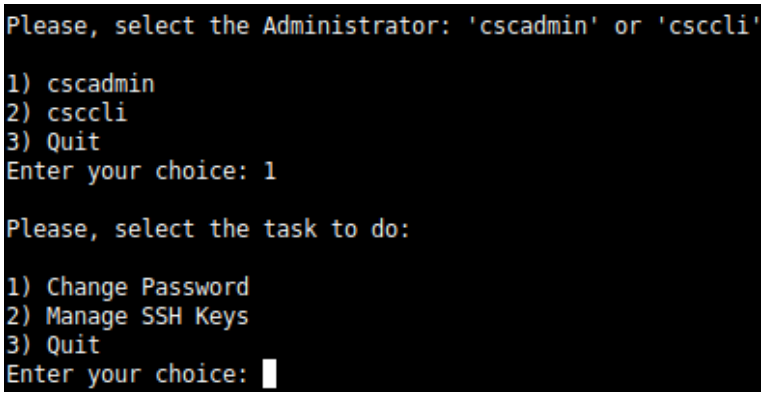
The CSC PriCPA for AZure has 2 users configured: cscadmin (for SSH Administrator Console Access), csccli (standard user, disabled by default.).

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

```
Enter your choice: 1

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

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

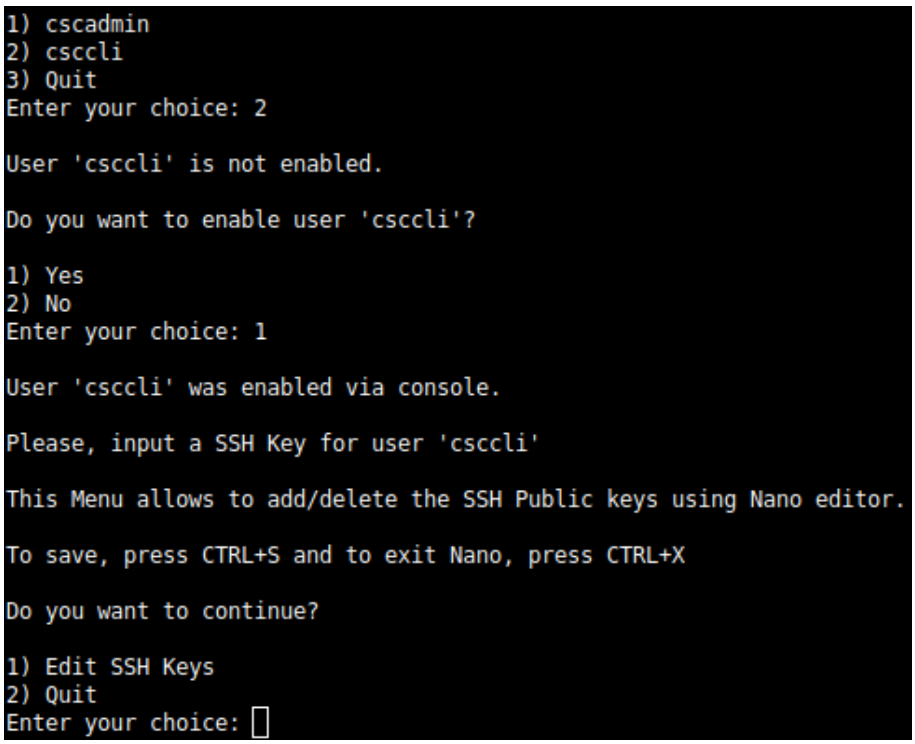
Note: *the user "cscadmin" cannot be disabled.*

12.3.2.1.1 "cscadmin" settings



12.3.2.1.2 "csccli" settings

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



12.3.2.1.3 Managing the SSH Key of a User

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

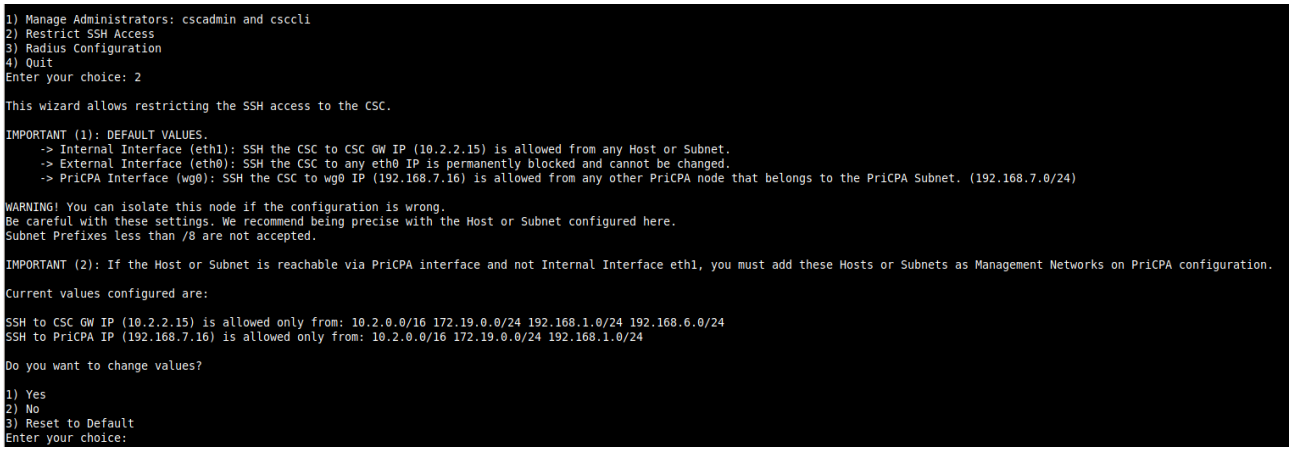
12.3.2.2 Restrict SSH Access

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

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

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

Example of configuration:



12.3.2.3 Radius Configuration

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

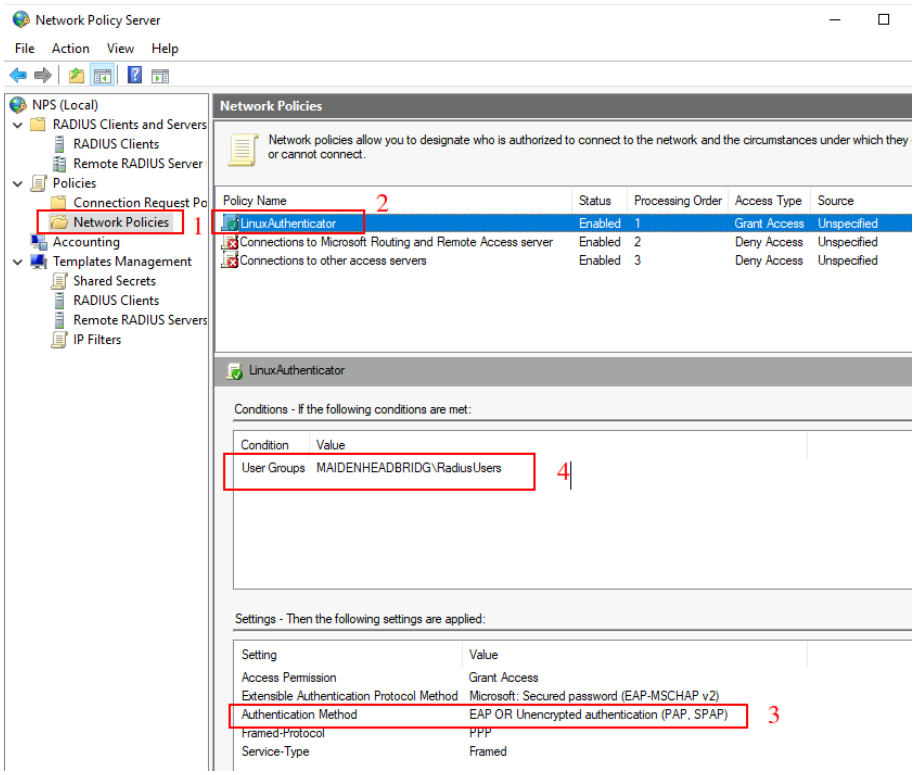
-> Configuration on the CSC: Add Radius Server and User:

```
Selection: 6

Please, select the task to do:

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

Welcome to the Radius Authentication Wizard.

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

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

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

1) Yes
2) No
Enter your choice: 1

Radius Servers:

No Radius Severs are configured.

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

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

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

1) Yes
2) No
Enter your choice: 2

No Radius Users are configured

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

Input Username: radius_user

Do you want to add another Username ?

1) Yes
2) No
Enter your choice: 2

Radius values to configure are:

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

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

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1

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

-> Example Configuration Windows NPS

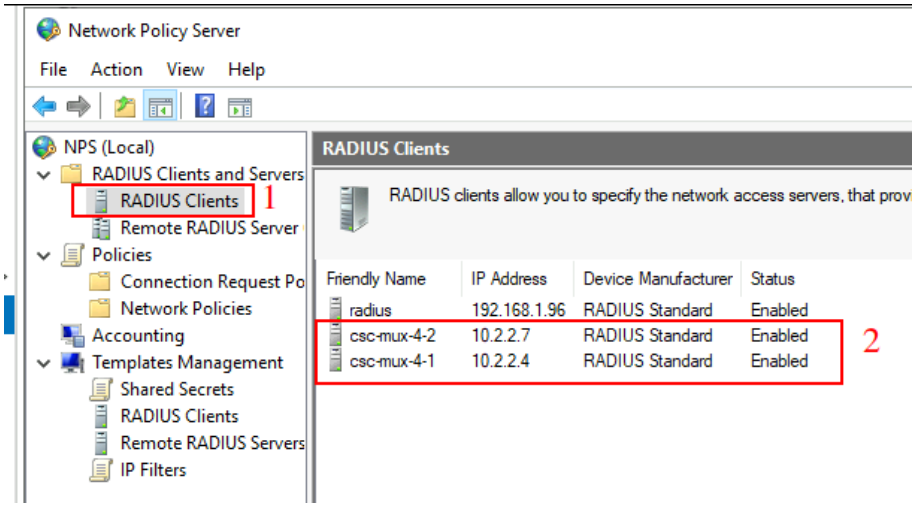
1 - Create Network Policy

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



2 - Add the CSC as Radius Clients:

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



12.4 Configure DNS, SNMP, NTP and Timezone.

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

System and Traffic Logs
11) View System Logs.
12) Configure Syslog and Traffic Logs.

e) Exit

Selection: 10

Please, select what you want to configure:

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

12.4.1 DNS servers

You can change DNS servers here. The default DNS values are Google DNS 8.8.8.8 and 8.8.4.4

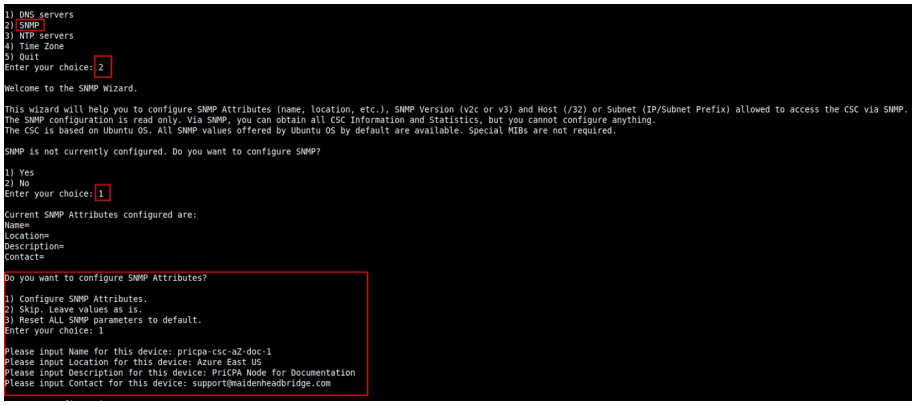
```
Selection: 10

Please, select what you want to configure:

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

Your current DNS Servers are: 8.8.8.8 ; 8.8.4.4

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

Do you want to change the DNS servers?

1) Yes
2) No
Enter your choice: 1

Primary DNS Server (IP): 172.19.0.100
Secondary DNS Server (IP): 1.1.1.1

(MHB-CSC)(INFO) CSC: pricpa-vm-c001001-b: DNS Servers changed via console. Using 172.19.0.100 and 1.1.1.1
```

12.4.2 SNMP

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

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

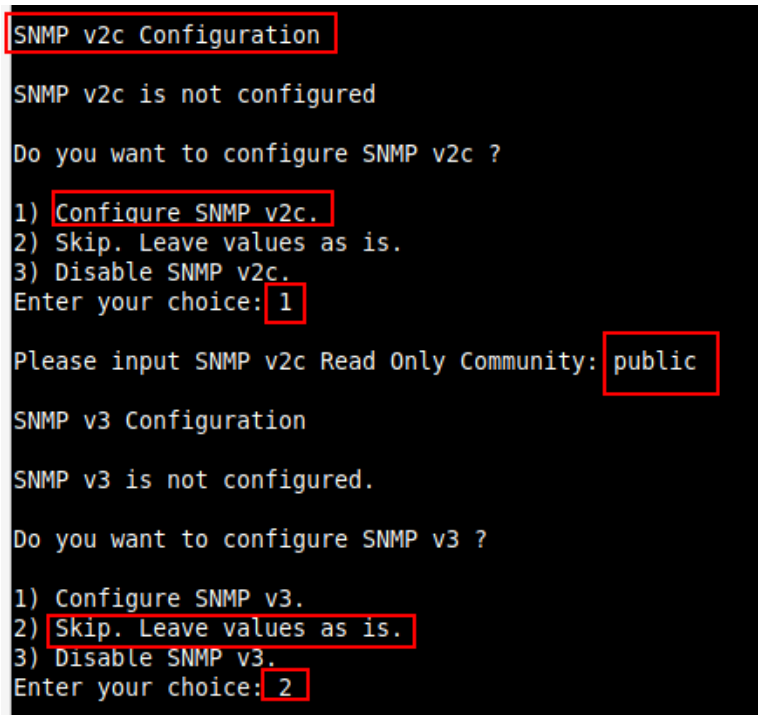
12.4.2.1 Configure SNMP attributes



12.4.2.2 SNMP v2c configuration

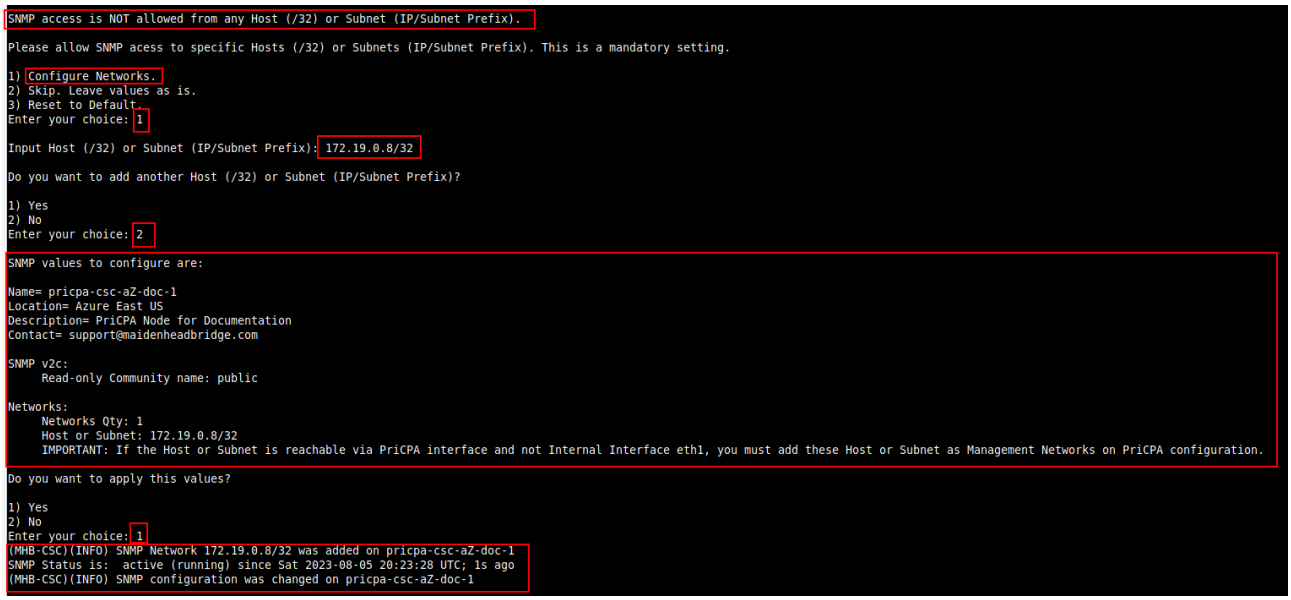
SNMP version 2c requires the "read only community" and the IP or Subnet of the SNMP platform.

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



12.4.2.3 SNMP Networks

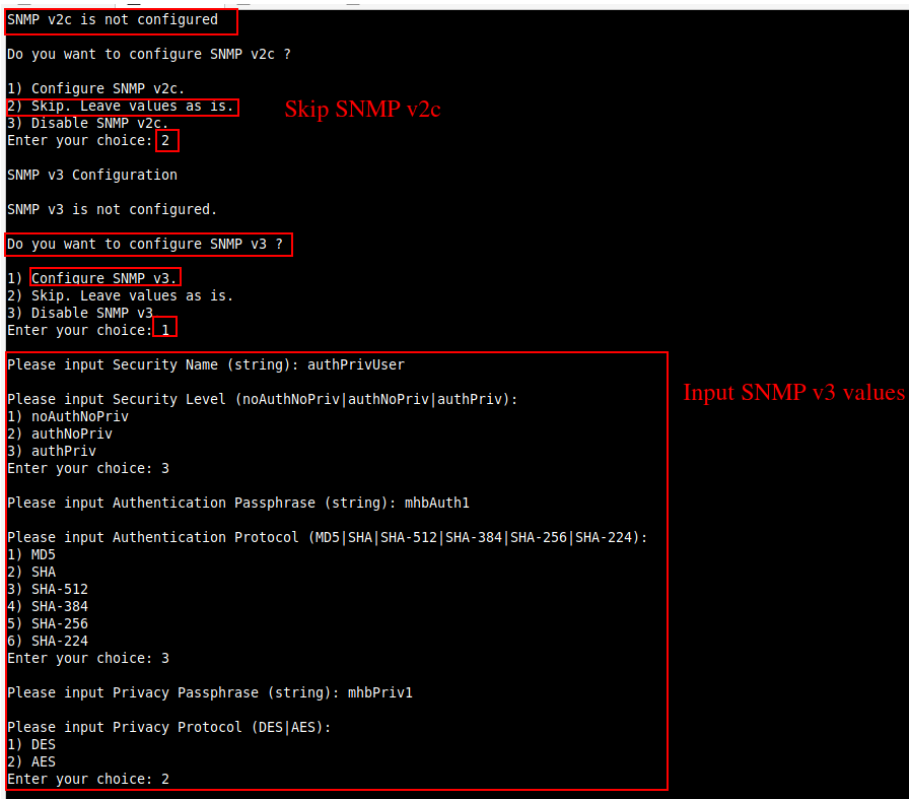
The CSC blocks all SNMP request by default. You need to enable the source IPs (or Subnets) that will query the CSC using SNMP. This setting is mandatory for SNMP v2c and v3.



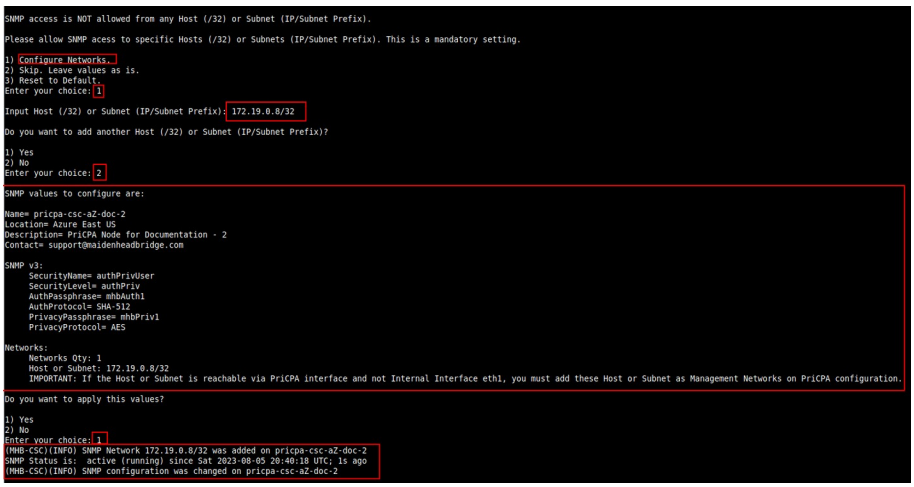
12.4.2.4 SNMP v3 configuration

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

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



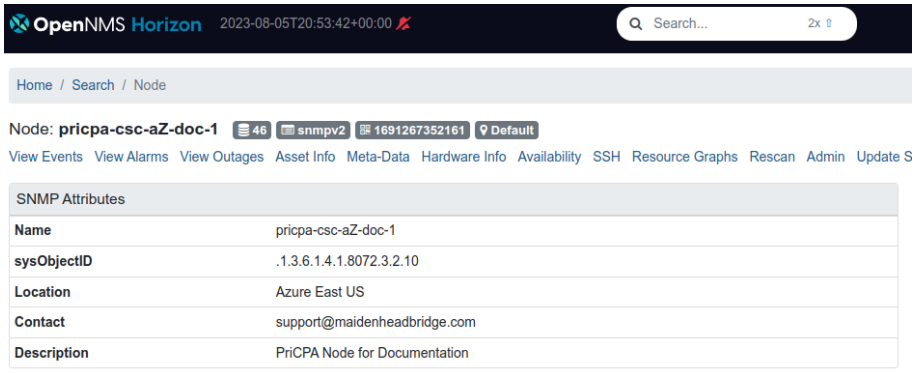
Configure Networks and Confirm values:



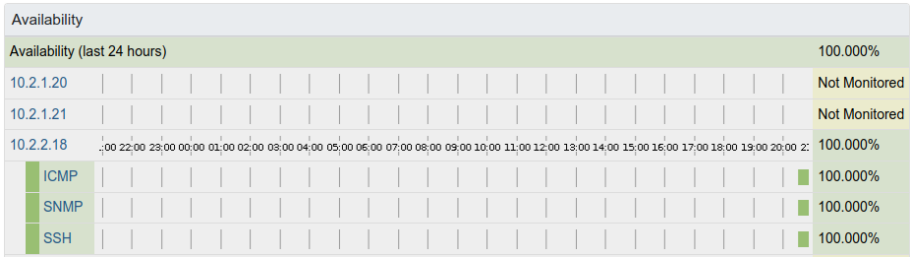
12.4.2.5 What can you do with SNMP?

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

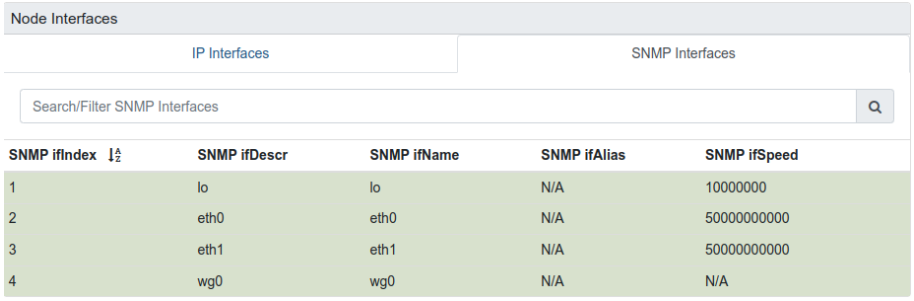
12.4.2.5.1 Node Information



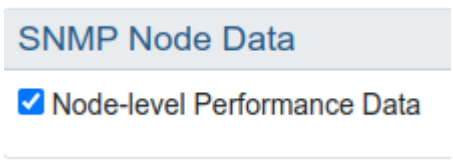
12.4.2.5.2 Node Availability

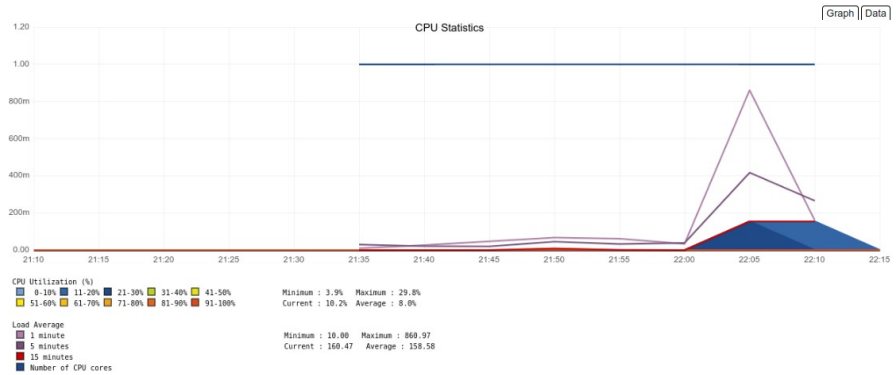


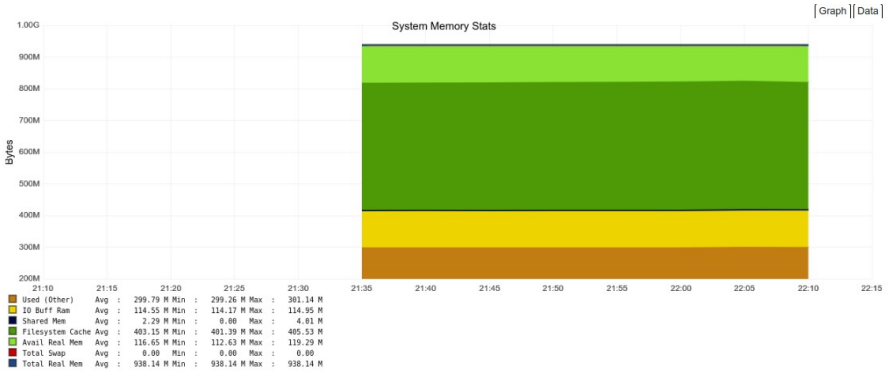
12.4.2.5.3 Node Interfaces (IP & SNMP)



12.4.2.5.4 Node Statistics (CPU, Memory, etc)

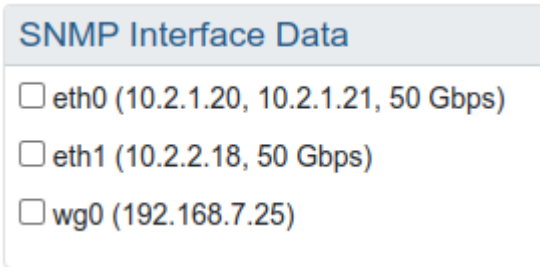




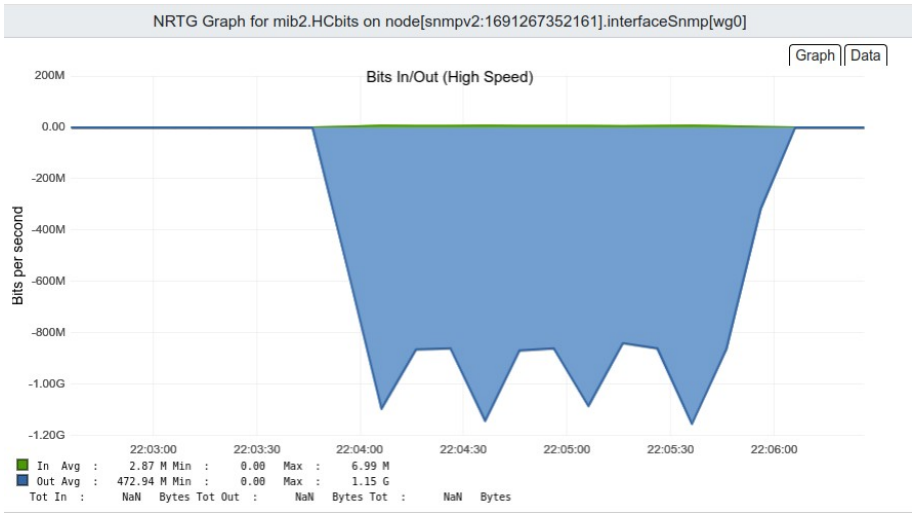


12.4.2.5.5 Interfaces Traffic

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



Example of real time traffic on PriCPA interface:



12.4.3 NTP Servers

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

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

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

You are using default Ubuntu NTP servers.

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

Do you want to change the NTP servers?

1) Yes
2) No
Enter your choice: 1

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

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

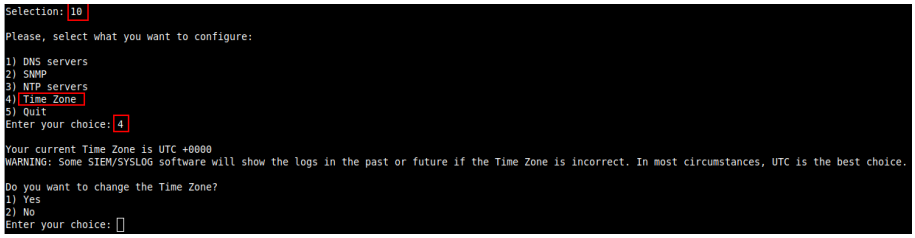
Check the Status:

```
Selection: 10

Please, select what you want to configure:

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

Your current NTP Servers are: 172.19.0.199 ; 192.168.1.199

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

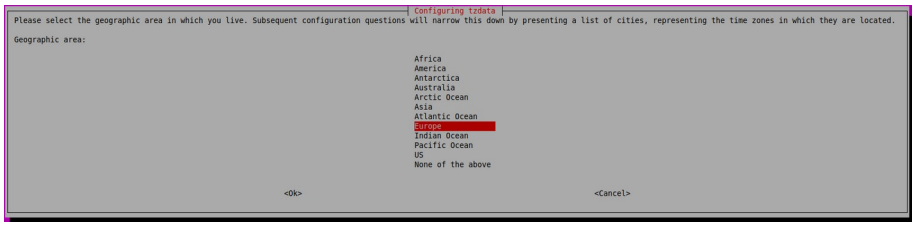
The NTP Server connects correctly when the Status is: "Initial synchronization to time server xxxx".

12.4.4 Change Timezone

Use this menu to select the timezone of the CSC.



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



12.5 System and Traffic Logs

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

```
System and Traffic Logs
11) View System Logs.
12) Configure Syslog and Traffic Logs.
```

12.5.1 View System Logs

```
Selection: 11

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

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

Current Month (August 2023) Logs for pricpa-csc-aZ-doc-1

Aug  4 15:34:50 root: (MHB-CSC)(INFO) CSC: pricpa-csc-aZ-doc-1: DNS Servers changed via console. Using 1.1.1.1 and 8.8.8.8
Aug  4 15:34:50 root: (MHB-CSC)(INFO) CSC: pricpa-csc-aZ-doc-1: Syslog Servers using (IP/TCP PORT): 172.19.0.5/5514
Aug  4 15:34:54 root: (MHB-CSC)(INFO) AWS SSM Agent is  active (running) since Fri 2023-08-04 15:34:54 UTC; 34ms ago
Aug  4 15:34:54 root: (MHB-CSC)(INFO) AWS SSM Agent Registration values are: {"ManagedInstanceID":"mi-02f61b13176c35082","Region":"eu-west-2"}
Aug  4 15:34:55 root: (MHB-CSC)(INFO) Private Access - Private Access service is enabled on pricpa-csc-aZ-doc-1. via configUserData JSON file.
Aug  4 15:34:56 root: (MHB-CSC)(INFO) PriCPA Remote Management Network: 172.19.0.0/24 was added.
Aug  4 15:34:56 root: (MHB-CSC)(INFO) PriCPA Remote Management Network: 192.168.1.0/24 was added.
Aug  4 15:34:56 root: (MHB-CSC)(INFO) PriCPA Remote Management Network: 192.168.6.0/24 was added.
```

12.5.2 Configure Syslog and Traffic Logs

```
Selection: 12

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

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

Do you want to change these values?

NOTE: Reset to default values will reboot the CSC because Traffic Logs are enabled.

1) Yes
2) No
3) Reset default values
Enter your choice: 1

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

Traffic Logs are enabled. Do you want to disable Traffic Logs?

1) Yes
2) No
Enter your choice: 2

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

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

1) Yes
2) No
Enter your choice: 2

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

1) Yes
2) No
Enter your choice: 1
(MHB-CSC)(INFO) CSC: pricpa-csc-aZ-doc-1: Syslog Servers changed via console. Using (IP/TCP PORT): 172.19.0.5/5514
```

13 Remote Management

You can use several tools to manage the CSC remotely. This chapter shows how to use SSH, AWS Systems Manager (Fleet Manager) and Rundeck.

13.1 Using SSH.

Almost all remote management tools can run SSH commands. The CSC PriCPA has a list of commands created for this purpose. See the Command Table below.

You can execute commands enabling the user "csccli" and running an SSH command with the following format:

ssh -i **<ssh-key>** csccli@**<CSC IP(eth1)>** "sudo **<comand>**; exit"

Example:

ssh -i ns-cscadmin-key csccli@172.19.1.27 "sudo /home/cscadmin/aws-mt4; exit"
GENERAL Information Company : Maidenhead Bridge Location : PriCPA-01-MHB CSC ID : pricpa-vm-c001001-b CSC date: Mon 11 Dec 12:26:05 UTC 2023 Soft version : 1.0.6 INTERFACES Information External: PriCPA IP: 192.168.1.25 | CSC IP(eth0): 192.168.1.27/24 | Network Gateway: 192.168.1.240 is Alive Internal: CSC GW IP: 172.19.1.25 | CSC IP(eth1): 172.19.1.27/24 | Network Gateway: 172.19.1.188 is Alive DNS Information DNS Server (1) IP: 172.19.0.100 is Alive DNS Server (2) IP: 1.1.1.1 is Alive CONNECTIVITY Test CSC External Interface 192.168.1.27 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.73 PriCPA Interface 192.168.1.25 can reach test page (https://ip.maidenheadbridge.com) via Public IP 82.68.6.73 AWS SSM Agent AWS SSM Agent is not registered SYSLOG Information Primary Syslog / SIEM (IP/TCP PORT): 10.63.1.10/5514 is Alive Secondary Syslog / SIEM IP: Not configured Traffic Logs (IP packets) are enabled. HIGH AVAILABILITY Information This CSC (pricpa-vm-c001001-b) is Cluster ACTIVE

13.1.1 Commands table

Test #	Descritption	CSC Command
1	MHB-CSC-ShowConfigurationAndStatus	/home/cscadmin/aws-mt4
2	MHB-CSC-Show-Private-Access-ALL-Peers-Status	/home/cscadmin/aws-show-private-access-all-peers-status
3	MHB-CSC-Refresh-Private-Access-Peers-URL	/home/cscadmin/aws-refresh-private-access-peers-url
4	MHB-CSC-Reload-Private-Access-JSON-file	/home/cscadmin/aws-reload-private-access-peers-json
5	MHB-CSC-ShowLogCurrentMonth	/home/cscadmin/aws-l-current-month
6	MHB-CSC-ShowLogLastSixMonths	/home/cscadmin/aws-l-last-6-months

13.2 AWS Systems Manager

The easiest and most 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 must 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".

13.2.1 Create Documents

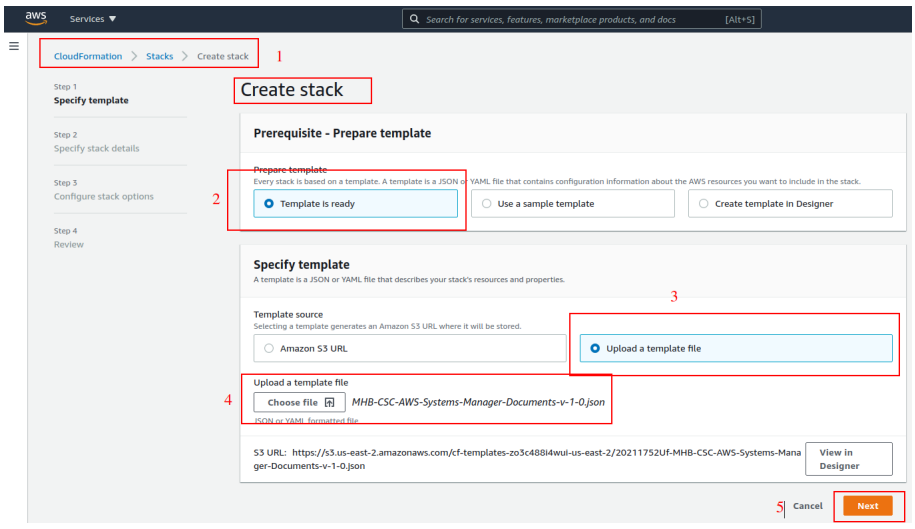
We provide a CloudFormation template to create all "Documents" in one shot.

Steps:

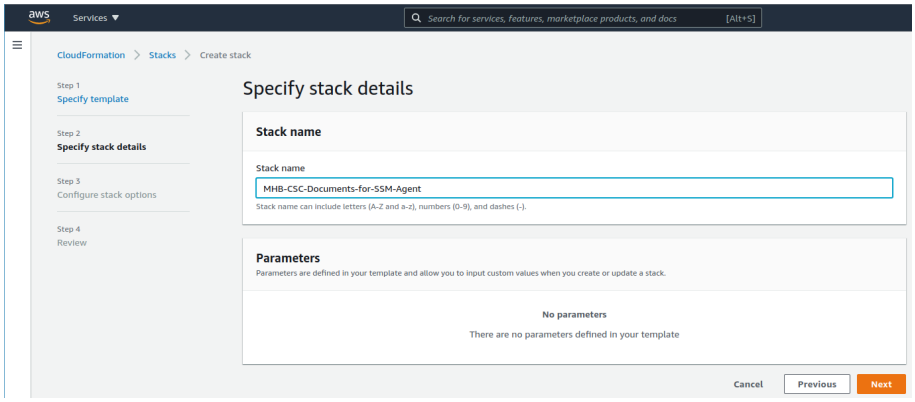
1. Download the CloudFormation template from:

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

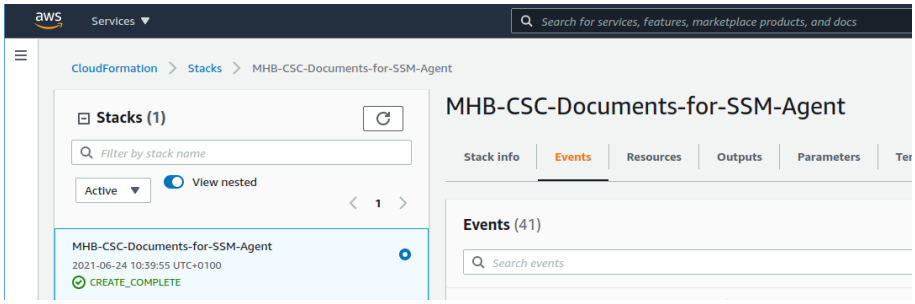
2. Deploy Stack. Go to Cloudformation →Create Stack→ Upload a template file



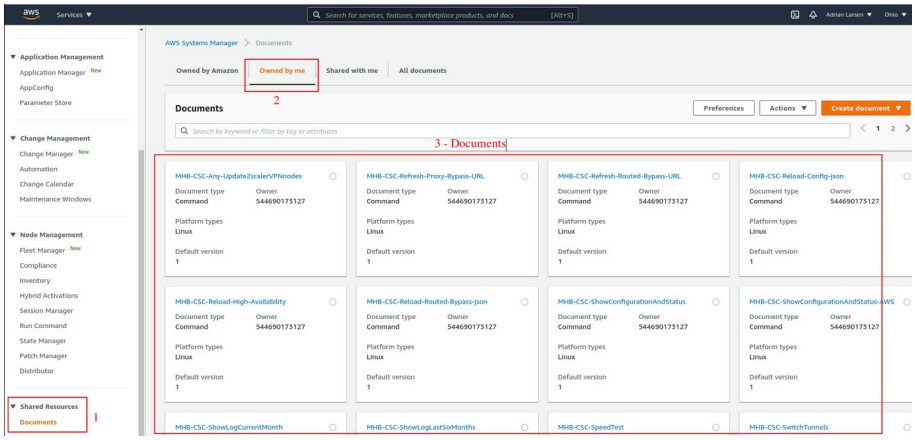
3. Click next.
4. Put the Stack Name



5. Click Next -> Next -> Create Stack.
6. Wait the Stack to complete.



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



8. Done!

13.2.2 Run Commands

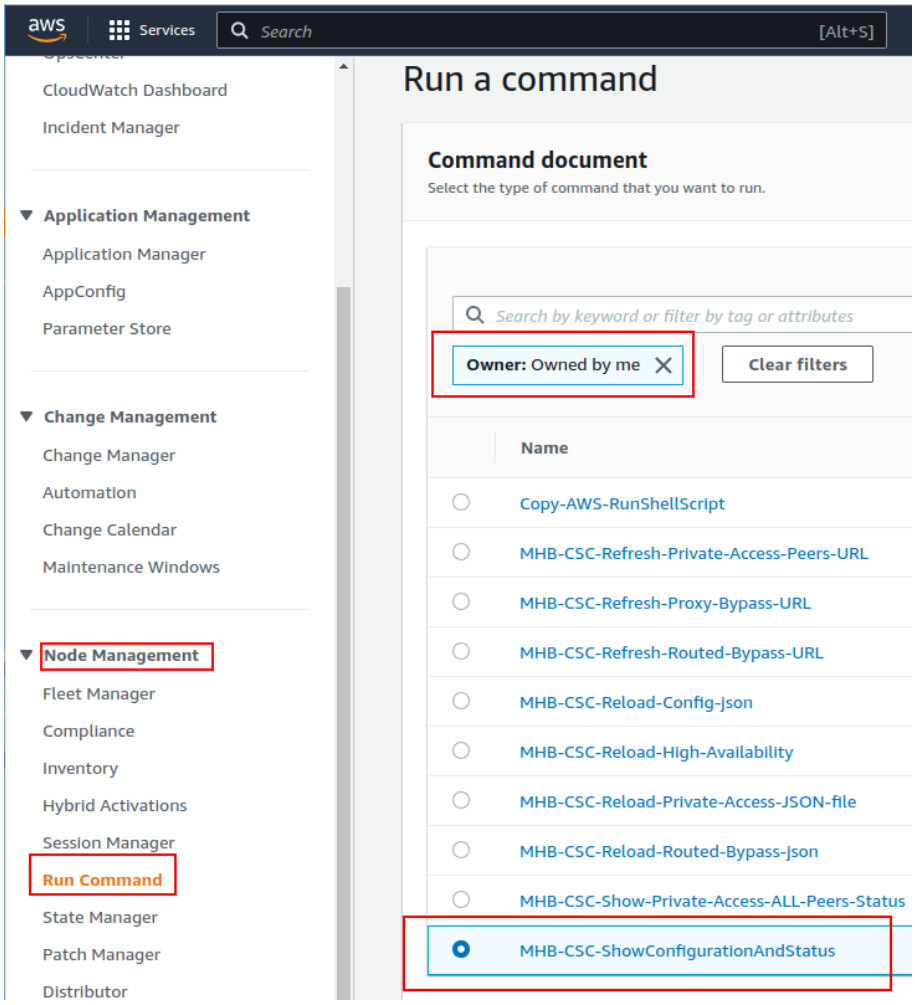
After you have created the Documents, you are ready to Run Commands on the CSC.

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

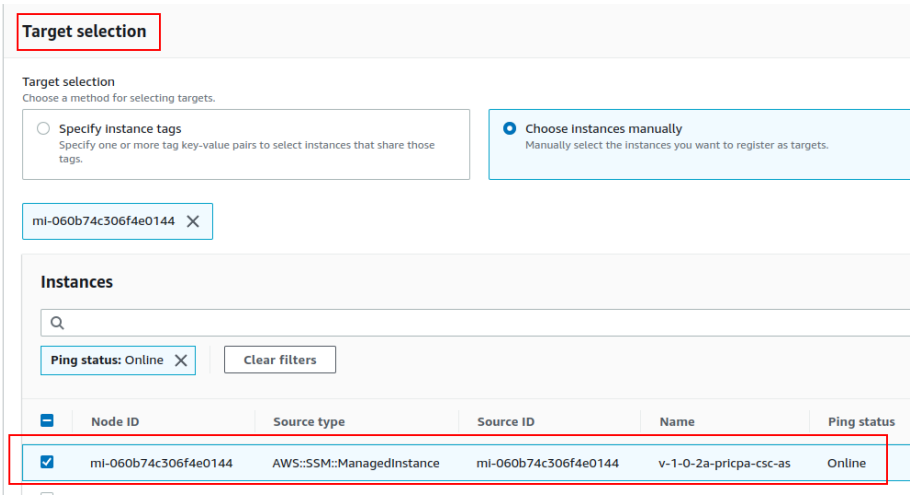
To "Run Commands", go to AWS Systems Manager → Instances & Nodes → Run Command.

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

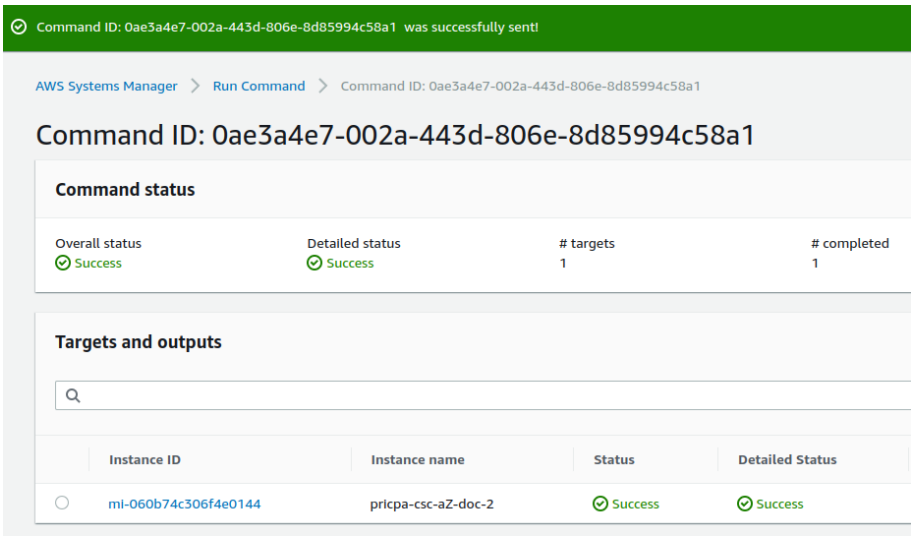
1. Run a Command
2. Select the Document created (Tip: Select "Owned by me")



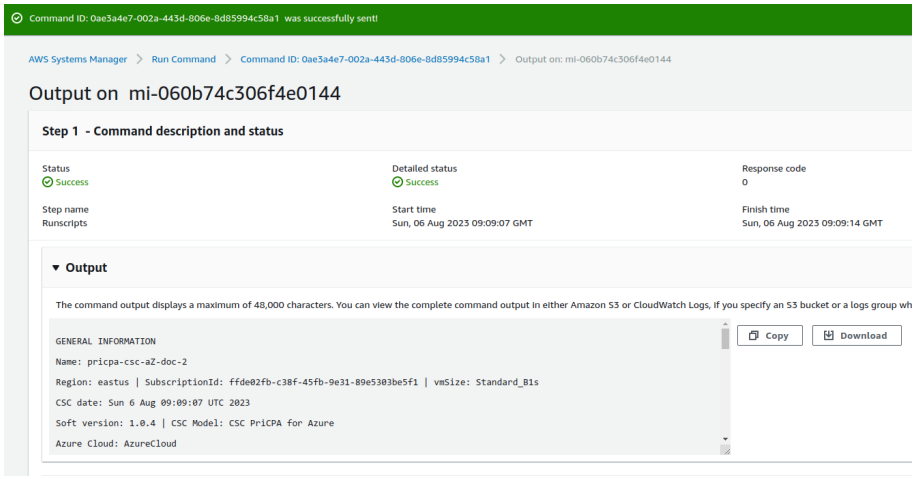
3. Scroll down and Select the Instances



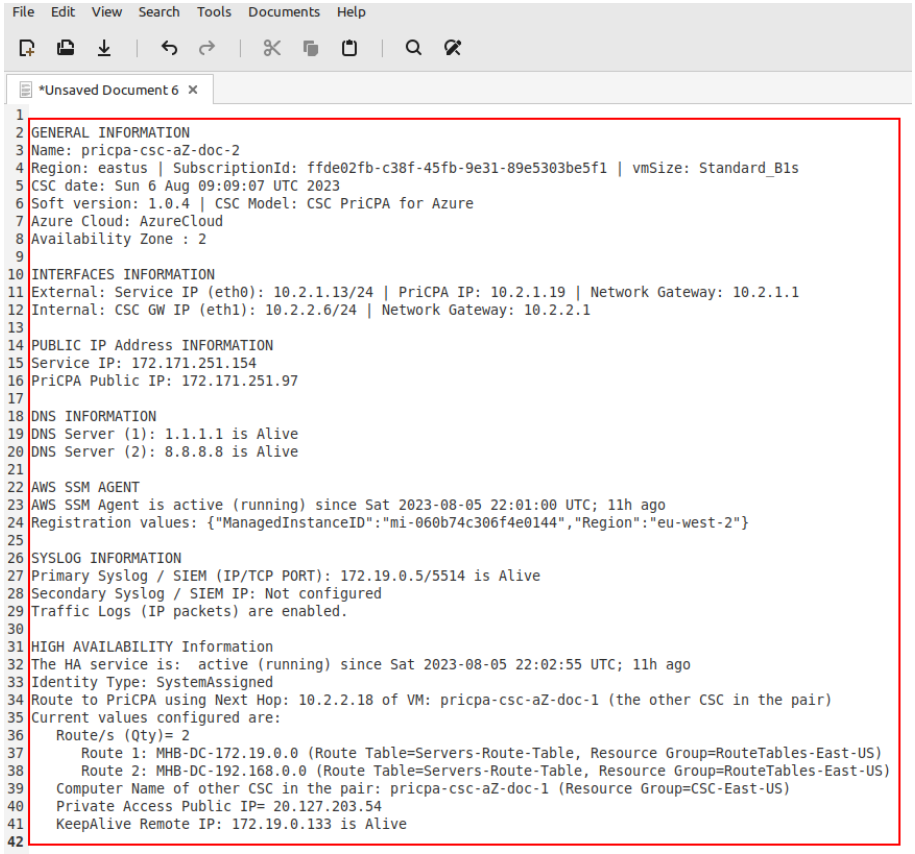
4. Click "Run" . Wait for the Command Status "success"



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



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



13.2.3 List of Documents available for "Run Command"

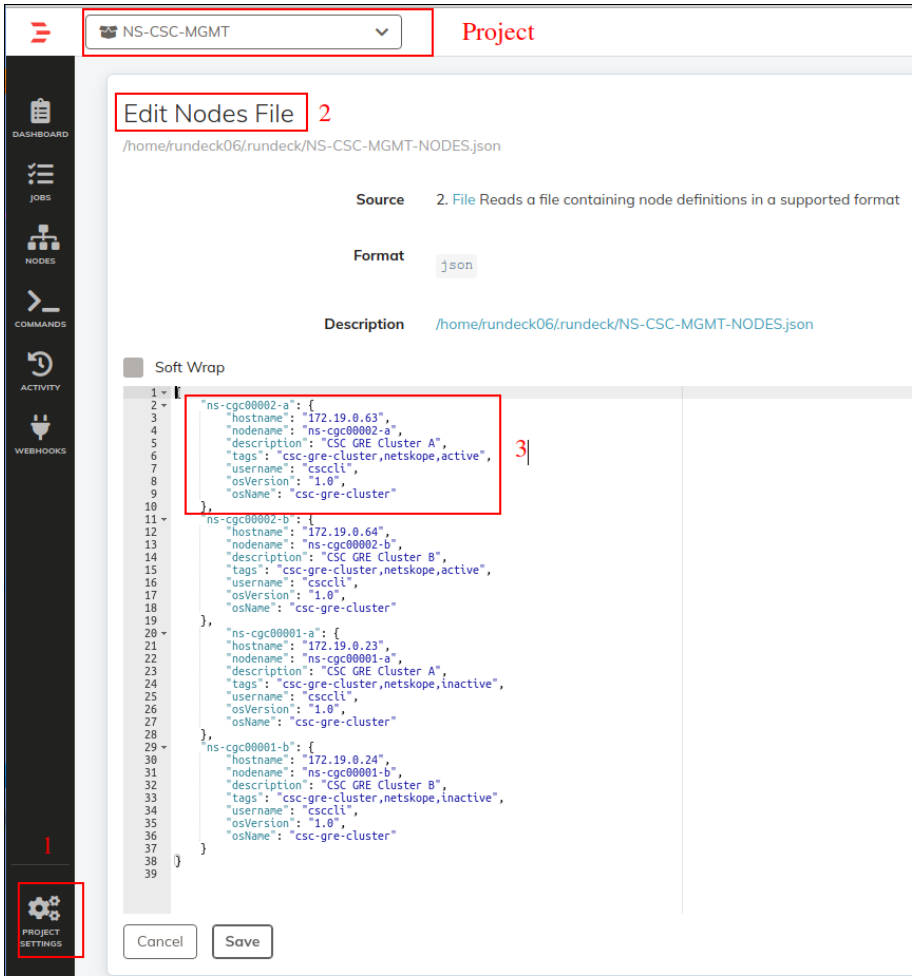
1. "MHB-CSC-ShowConfigurationAndStatus": Executes "Show Configuration and Status"
2. "MHB - CSC - Show Private Access ALL Peers status": Show the Status of all Private Access Peers.
3. "MHB - CSC - Refresh Private Access Peers URL": Refresh the Private Access Peers list using the values of the JSON file stored in the URL configured.
4. "MHB - CSC - Reload Private Access Peers JSON file": Reloads the values of privateAccessPeersConfig.json
5. "MHB-CSC-ShowLogCurrentMonth": Shows current month logs.
6. "MHB-CSC-ShowLogLastSixMonths": Shows last six month logs.

13.3 Rundeck

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

Installation Steps:

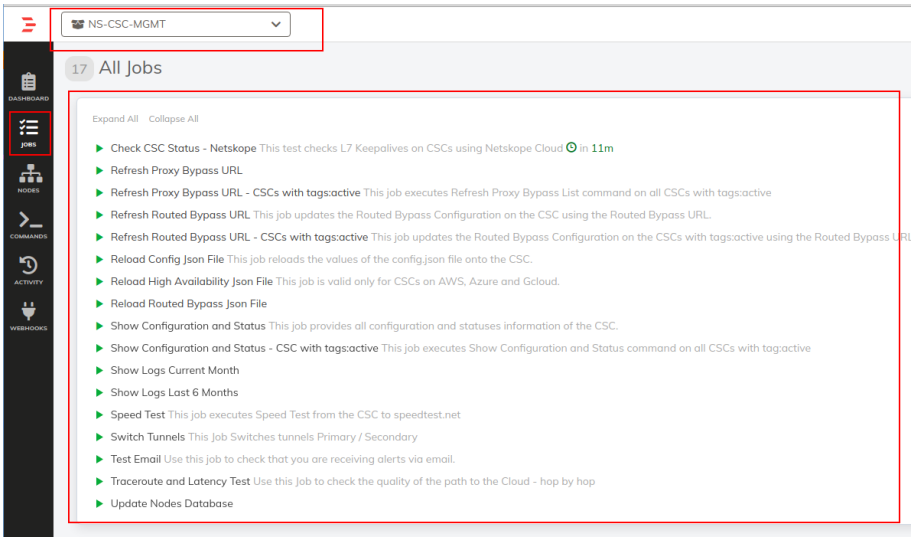
1. Install Rundeck. Intructions at: https://www.rundeck.com/open-source
2. Create a Project.
3. Enable user "csccli" and setup the SSH Public key on each CSC.
4. On the Project, setup the SSH Private and define the nodes:



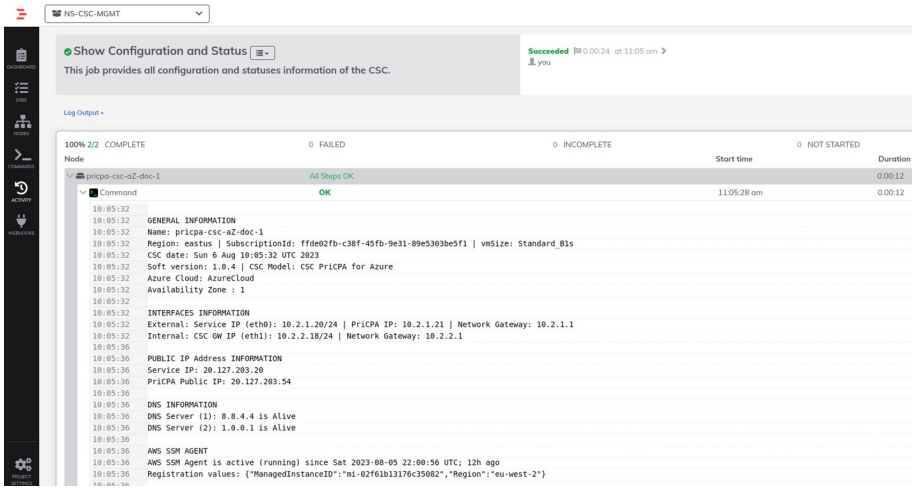
5. Create the jobs. Please, contact Support at http://support.maidenheadbridge.com for the latest Job List.

13.3.1 Jobs

The following screen shows the list of Jobs available.



13.3.2 Running job "Show Configuration and Status"



14 DevOps operations

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

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

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

The JSON files are available are:

1. **privateAccessPeersConfig.json:** Use this Json file to configure "networks" and "privateApps" on your Private Cloud.

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

14.1 privateAccessPeersConfig.json

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

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

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


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


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

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

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

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

15 Appendixes

15.1 Appendix A: Release Notes

15.1.1 Version 1.1.0 (April 2024)

Version 1.1.0 comes with the following enhancements:

- New! Multiple uplinks are supported. In this version, you can define multiple uplink IPs that can be public or private. This functionality provides multiple uplink redundancy via multiple ISPs or Private Networks (e.g., MPLS).
- New! Introducing a new feature that's as easy to use as powerful. You can now create a PriCPA cloud within your private networks. For instance, you can use MPLS to transport the PriCPA cloud. This functionality empowers you to encrypt and add zero trust to an MPLS network, all with a few simple steps.
- Other: Minor bug fixes and cosmetic changes.

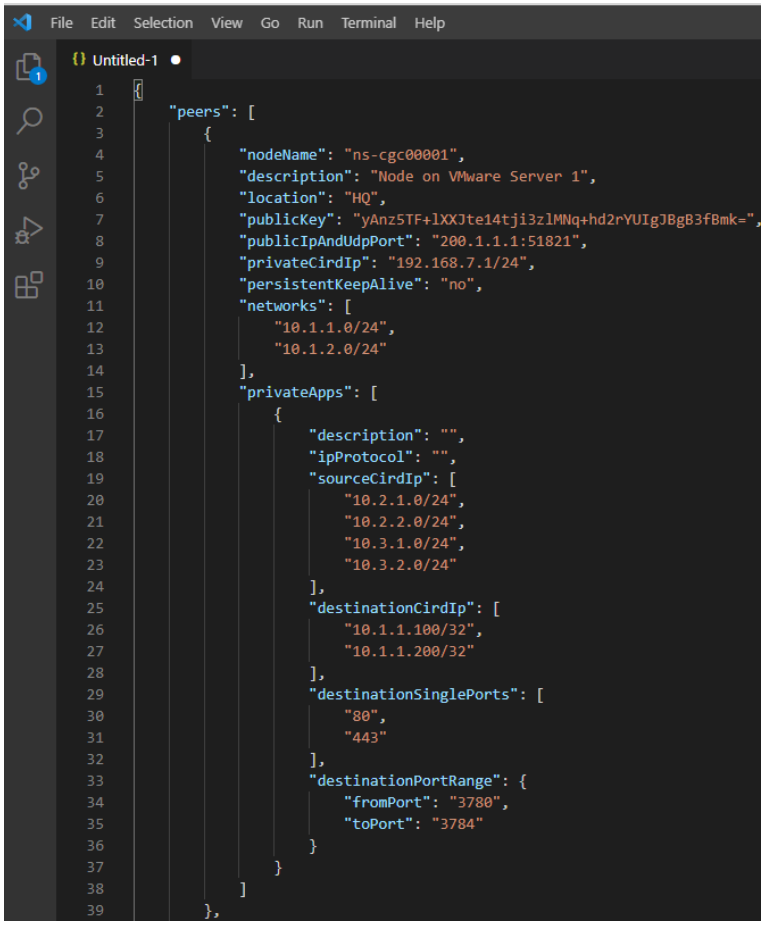
15.1.2 Version 1.0.6 (December 2023)

This is the initial public release of the CSC PriCPA for Virtual Platforms.

15.2 Appendix B: JSON formatters (Visual Code, Notepad ++)

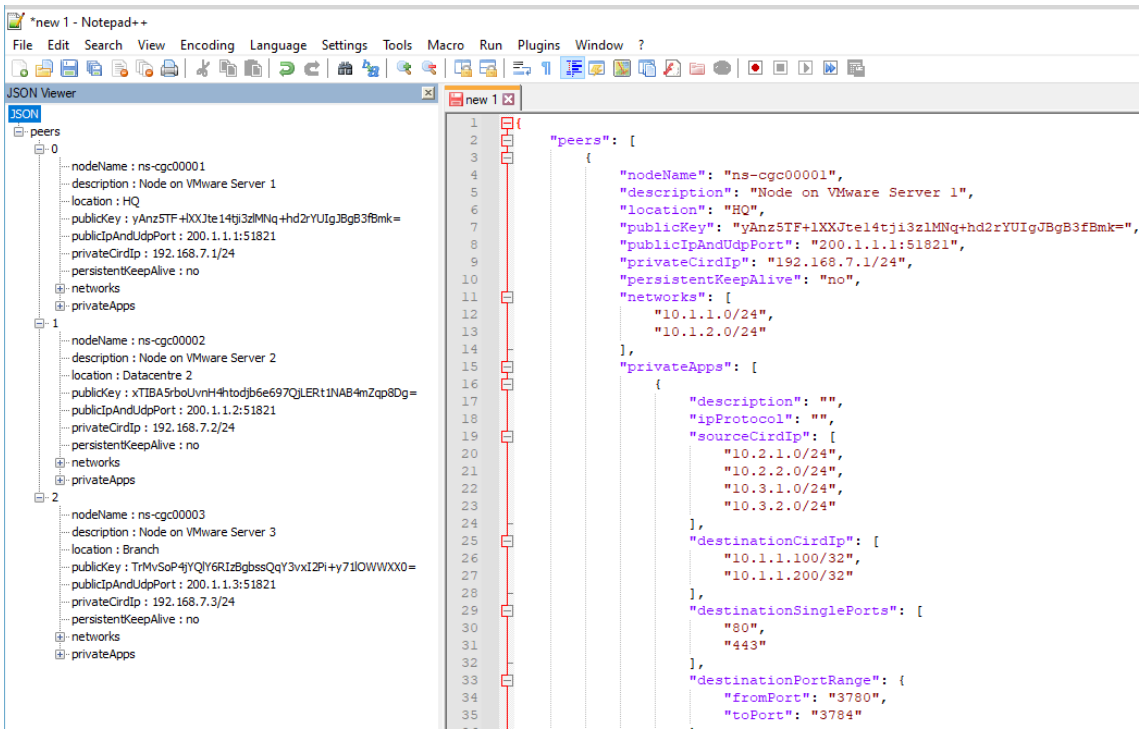
We strongly recommend using Software that can show errors on your JSON file and also can format (beautify) the file for better visibility. Below two examples.

15.2.1 Visual Code

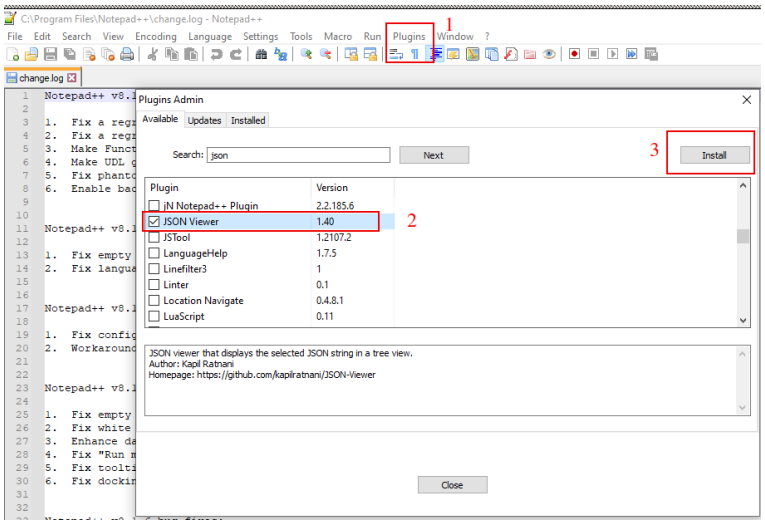


1. Download : https://code.visualstudio.com/download
2. Select your platform and install.
3. Create your JSON.
   - 3.1. Visual Code will show the errors in RED.
   - 3.2. To "Beautify" your JSON file press:
     - 3.2.1. On Windows: "Shift + Alt + F"
     - 3.2.2. On MAC: "Shift + Option + F"
     -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

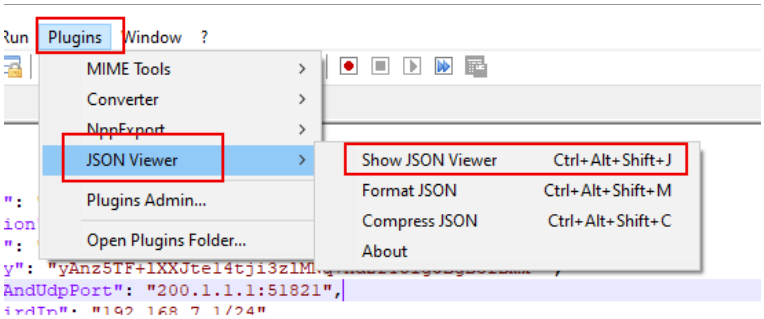
15.2.2 Notepad ++



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.

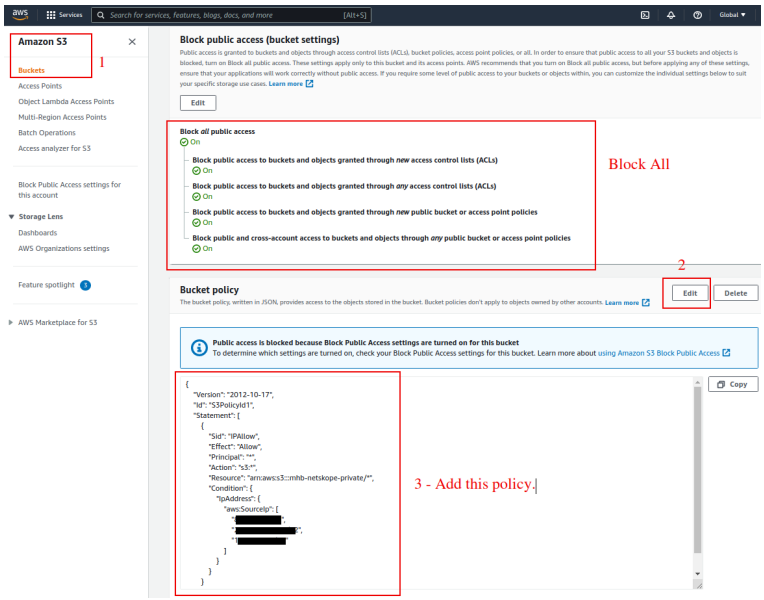


5. To format ("Beautify") your JSON go to: Plugins -> JSON Viewer -> Format JSON

15.3 Appendix C: Securing an AWS Bucket by source IP.

1. On your AWS console create a bucket with default values for permissions: "Block *all* Public Access = on"

2. On Bucket Policy, add your Public IPs in "aws:SourceIp":[]




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

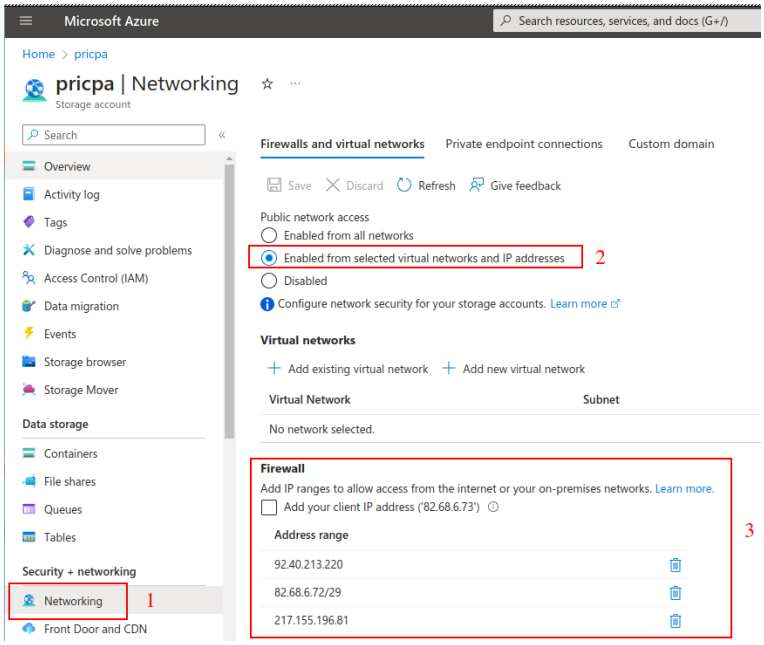
3. Done!

15.4 Appendix D: Securing Azure Blob by Source IP and Shared Access Signature.

IMPORTANT: An additional setup is required is the CSC and the Blob are on the same Azure Region. See below.

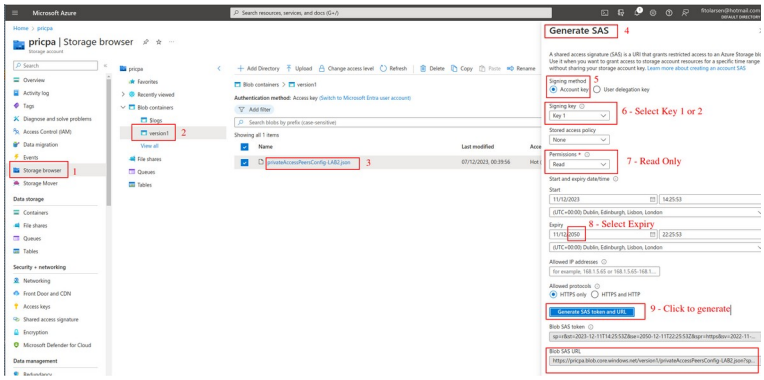
15.4.1 Create a Storage Account with permission per Source IP

Please, create an storage account for PriCPA and put the source IP allowed.



15.4.2 Create a SAS signature and "Blob SAS Url"

Upload the Peer JSON file to a blob and select generate a SAS

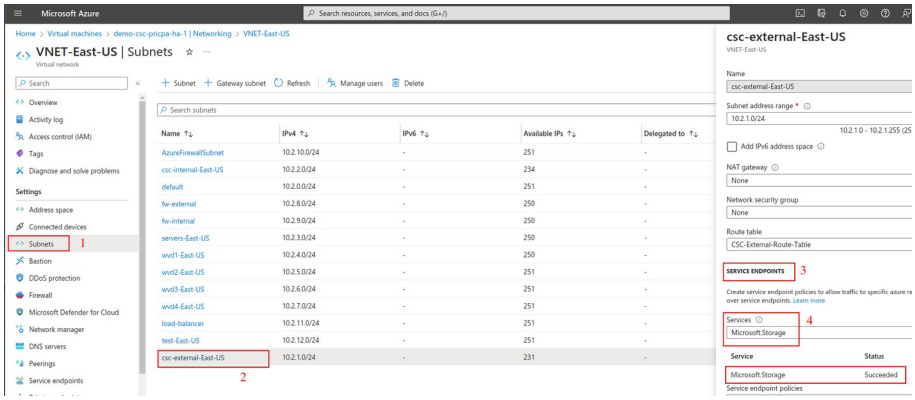


Use the "Blob SAS URL" to configure as your "URL" for Peers Json File.

15.4.3 Blob & CSC on the same Azure Region

If the CSC and the Blob are on the same region, the following settings are required:

1. Go to your VNET -> External Subnet of the CSC and configure: SERVICE ENDPOINTS -> Services: Microsoft.Storage



2. Go to the Blob configuration and allow the External Subnet of the CSC.