Maidenhead Bridge

Private Cloud Private Access (PriCPA)

Cloud Security Connector PriCPA for Azure

Administrator Guide

Version 1.0.5

August 2023

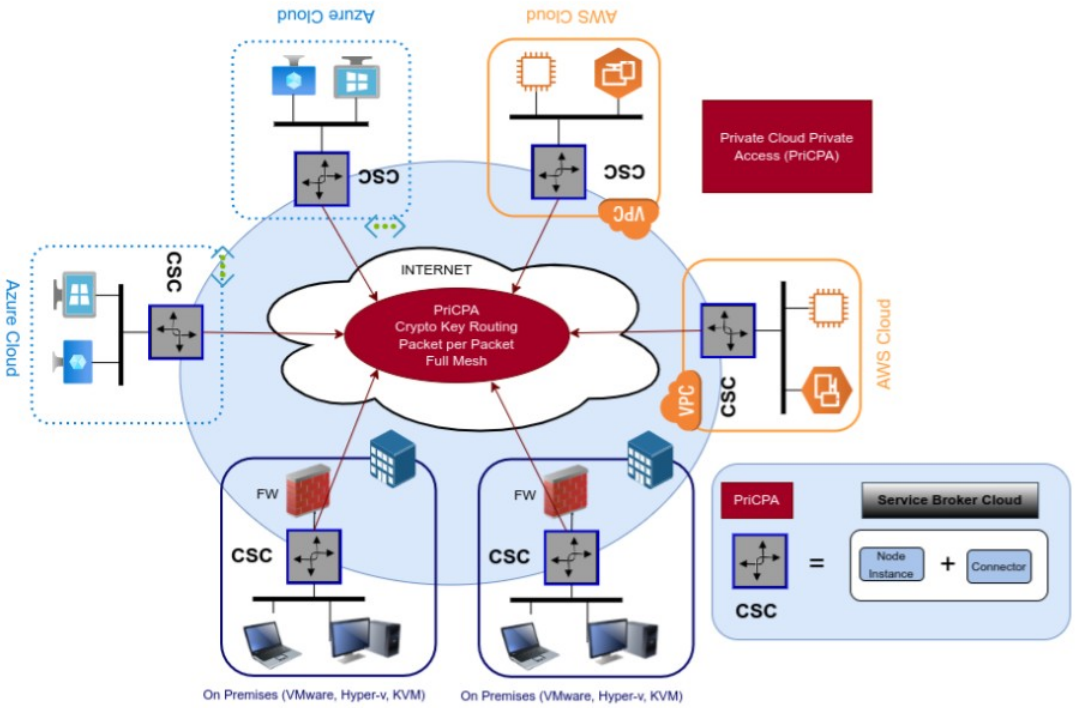
Table of Contents

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:

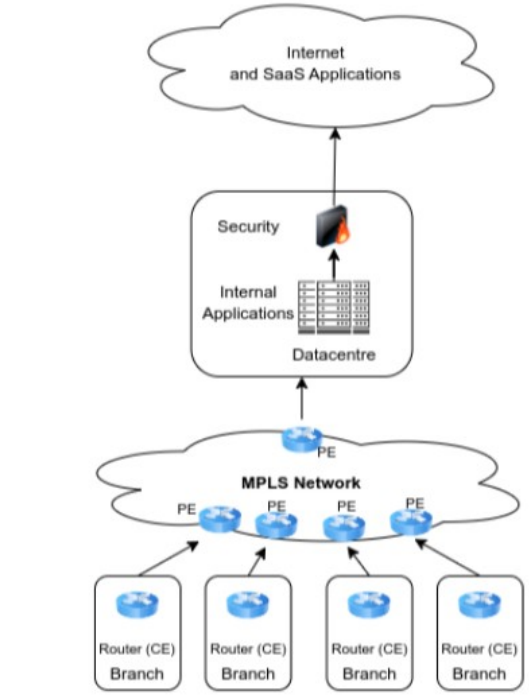
- 80% savings compared with VPN Gateways or Service Broker Cloud.
- Any to Any Communications. (Crypto Key Routing)
- All traffic is encrypted.
- All protocols are supported.
- Scalable: Multiple sites and multiple PriCPA clouds.
- Zero Trust.
- Networking as a code.

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

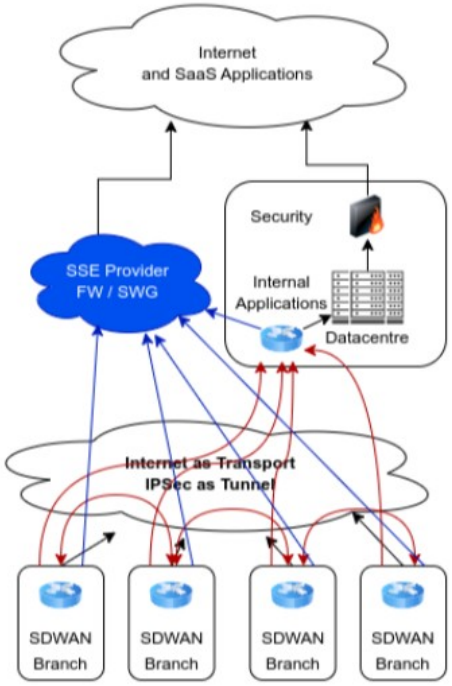


Benefits:

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.



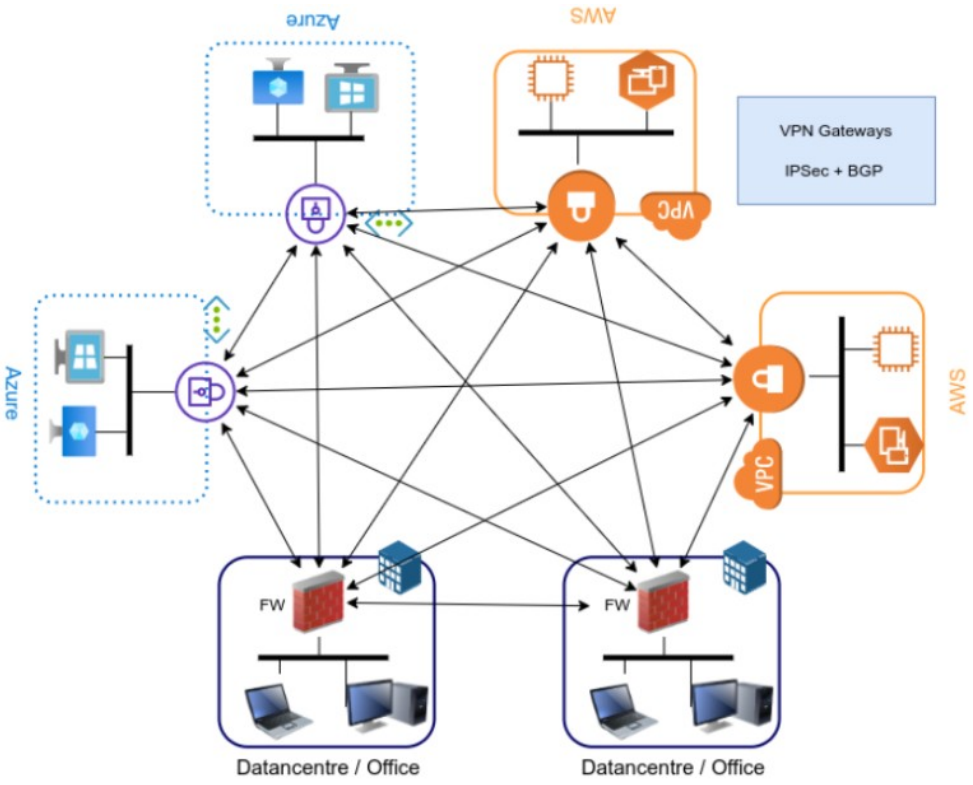
Benefits:

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)



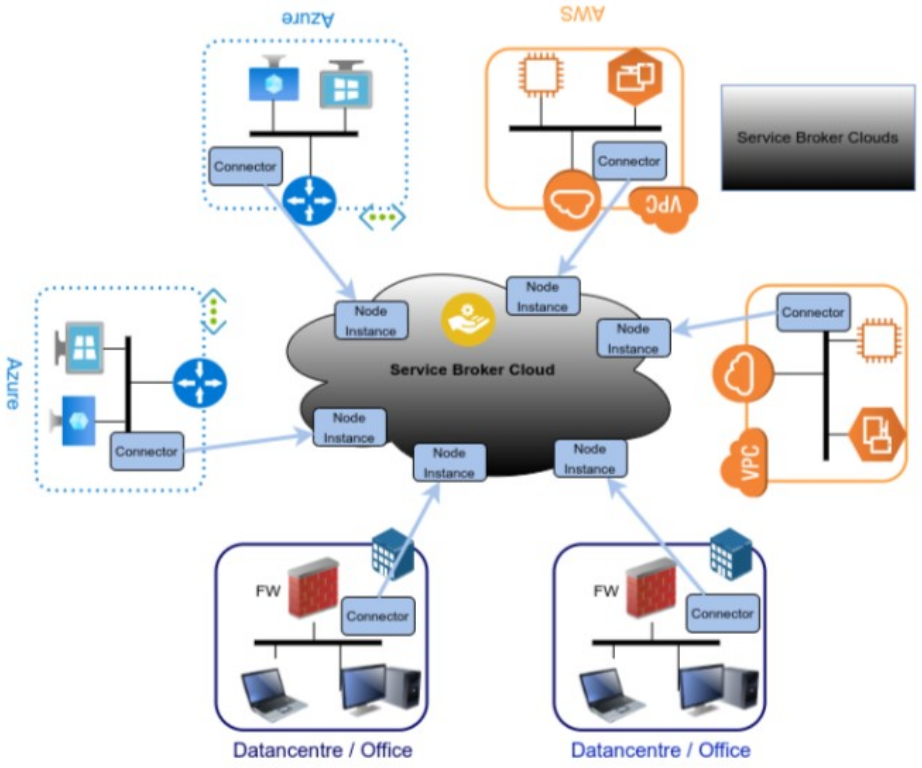
Benefits:

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



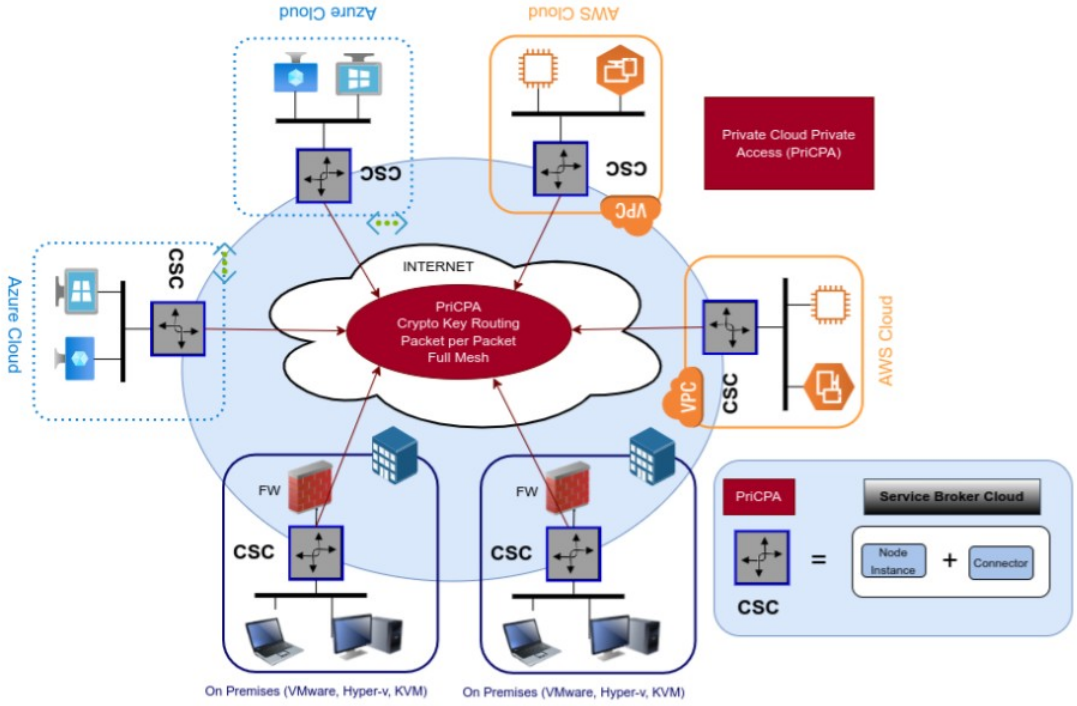
Benefits:

1. Zero Trust.
2. Site scalability (if Nodes available in your country)

Drawbacks:

1. Very Expensive.
2. No support for all protocols.
3. Data in transit is not encrypted. Extra license required.
4. Poor performance. Reduced even more if full encryption is activated.
5. Natted Traffic. Not visibility end to end.

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 for Azure.

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 via Azure CLI.
- **Simplicity:**
  - No Networking knowledge required.
  - No operational burden for Administrators.
  - Networking as a code: Single JSON file for policies.
  - DevOps automated deployment: Azure ARM or Terraform.

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

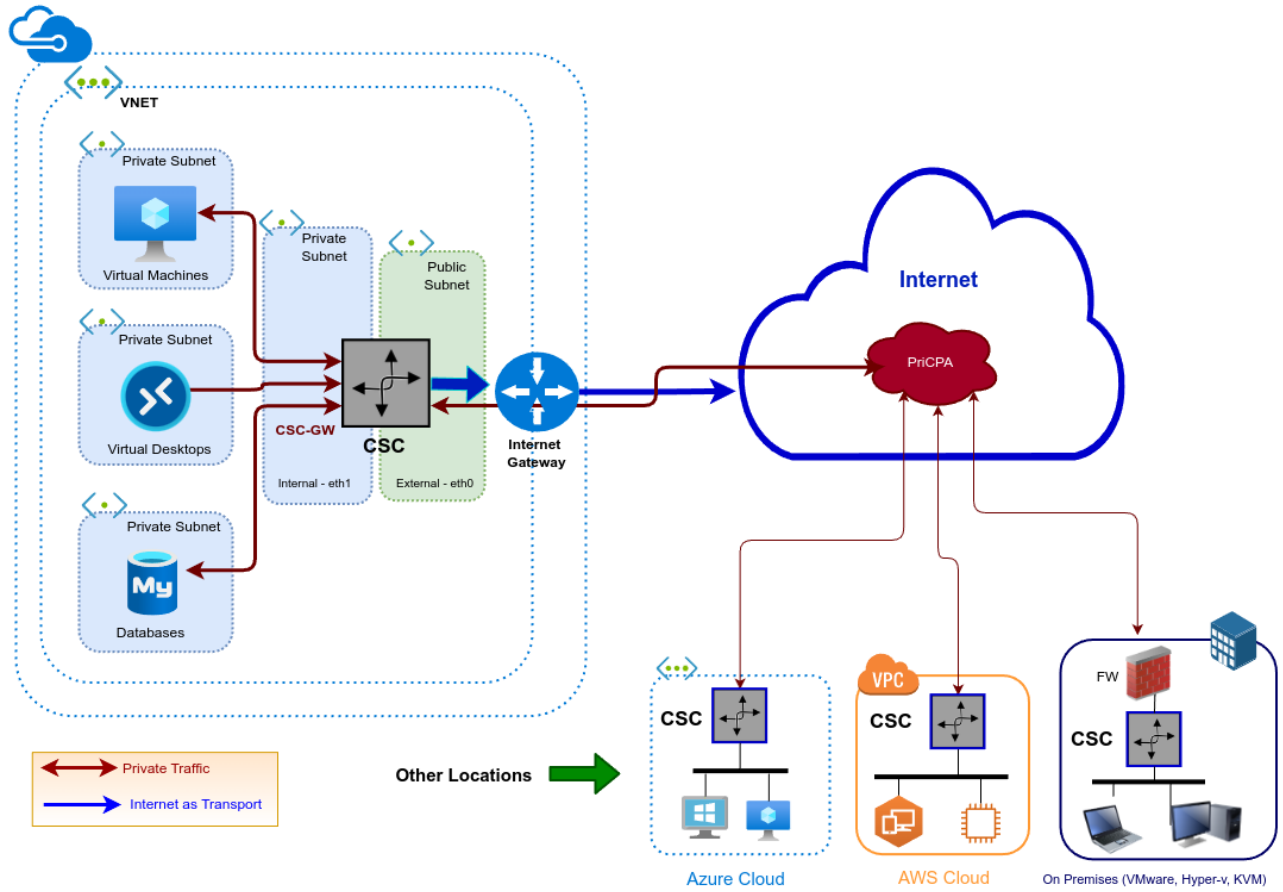
- 2 Steps configuration: Onboard the Node to PriCPA Cloud and Deploy Policies (Single JSON file).

- **Visibility:**
  - Traffic Logs and System Logs.
  - Traffic visibility End to End.
  - Source IPs preserved.
- **High Availability:**
  - Automatic Route provisioning ("next-hop") via Azure CLI.
  - Automatic configuration of "Floating Public IP".
  - Automatic re routing to Management Networks.
- **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).

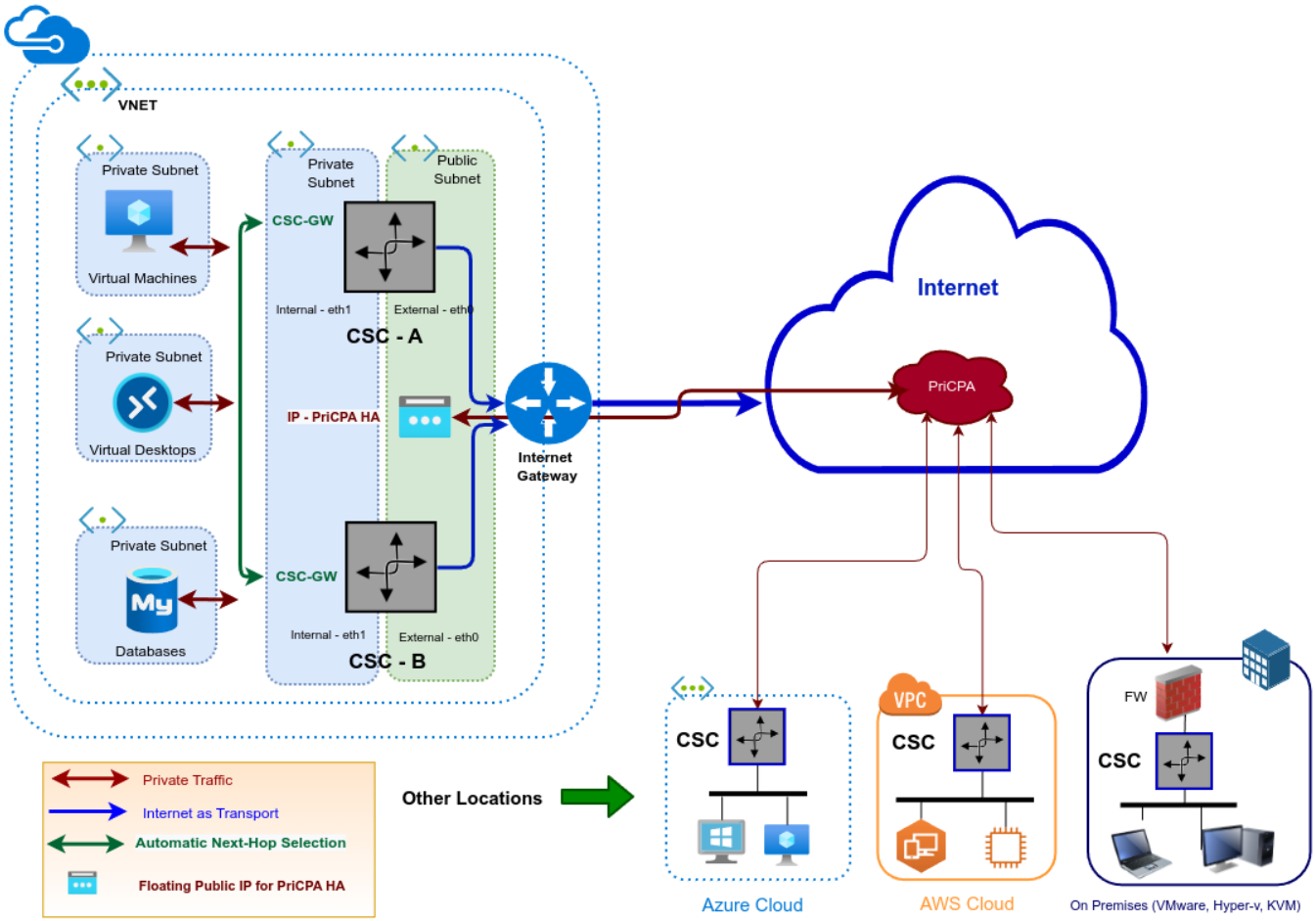
2 The CSC PriCPA for Azure uses Wireguard protocol. Wireguard is a trademark of Jason Donenfeld.

3 Diagrams

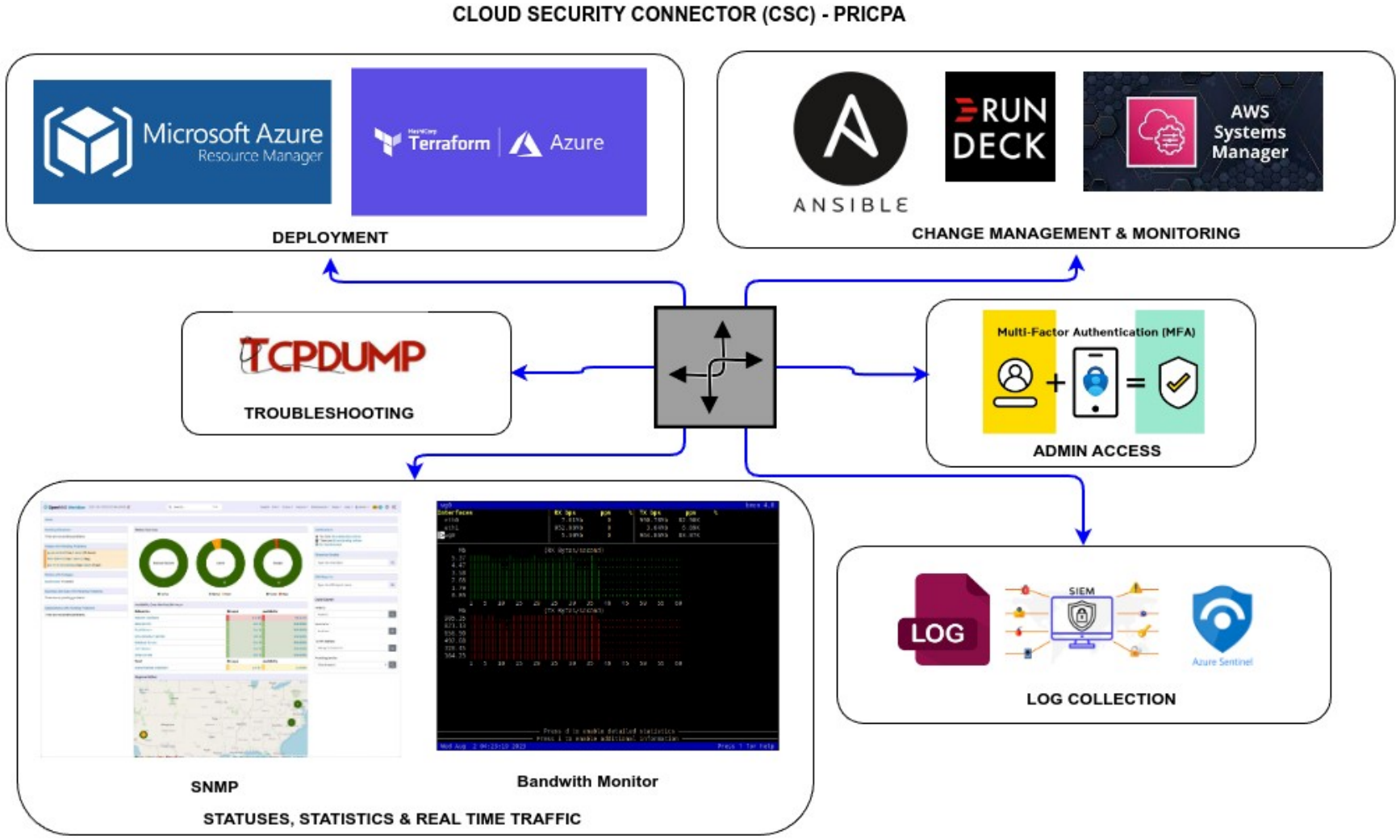
3.1 CSC PriCPA for Azure – Single deployment



3.2 CSC PriCPA for Azure – High Availability Deployment

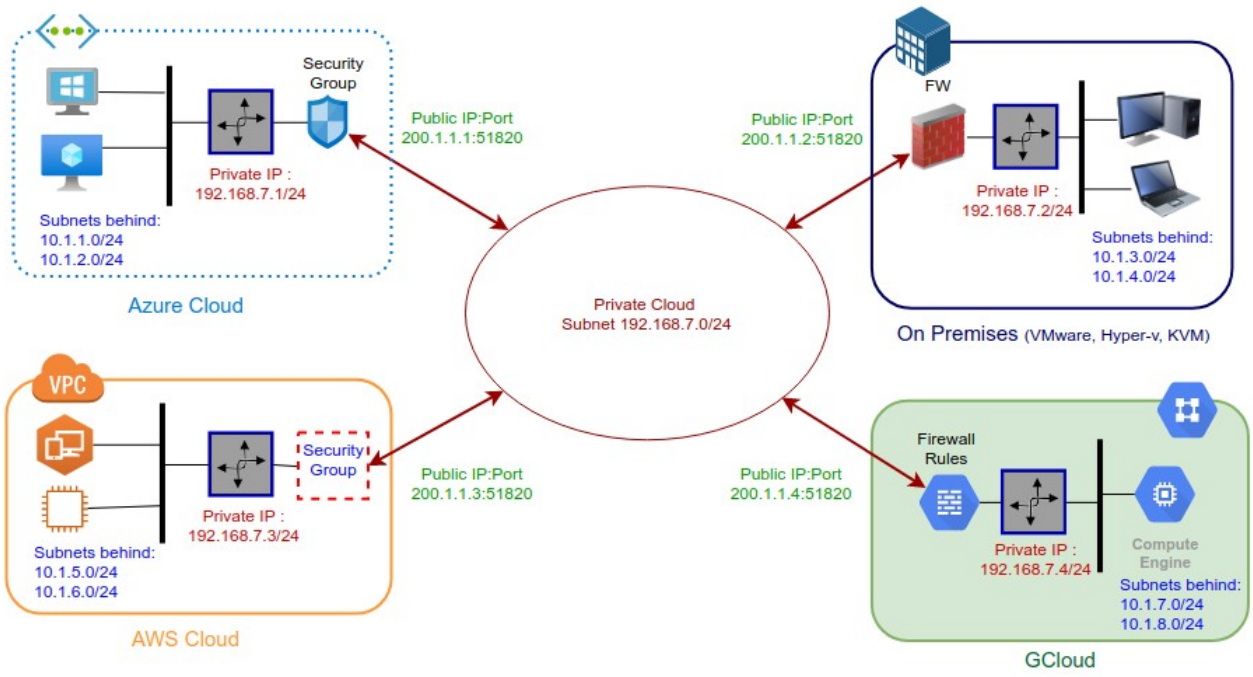


3.3 Deployment, management, troubleshooting, and monitoring integrations



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 is /24, up to 255 CSCs can participate in this Private Cloud.
2. Assign a Cloud Private IP to each CSC. In this example, we are assigning 192.168.7.1 to 192.168.7.4
3. The Public IP to be used will be automatically selected by the CSC on Public Clouds. 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, it is recommended to use the same port at all locations. 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 Groups Rules): The CSC for Azure, AWS and Gcloud will implement the firewall rules automatically. Manual FW rules are required when the CSC is "On-Premises". The CSC provides a JSON file with the necessary rules for FW configuration.

5 Creating the CSC PriCPA for Azure

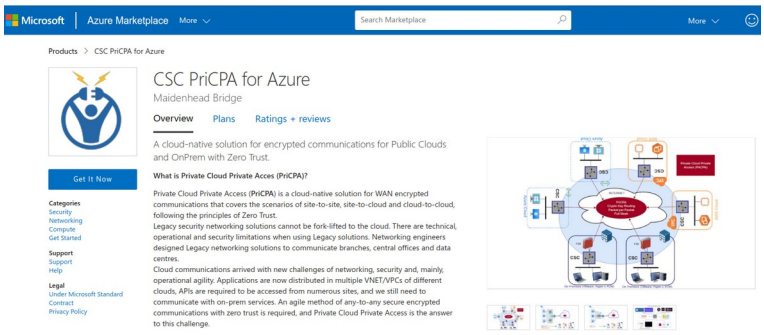
5.1 Prerequisites

Before launching the CSC PriCPA for Azure, you need to have these elements ready:

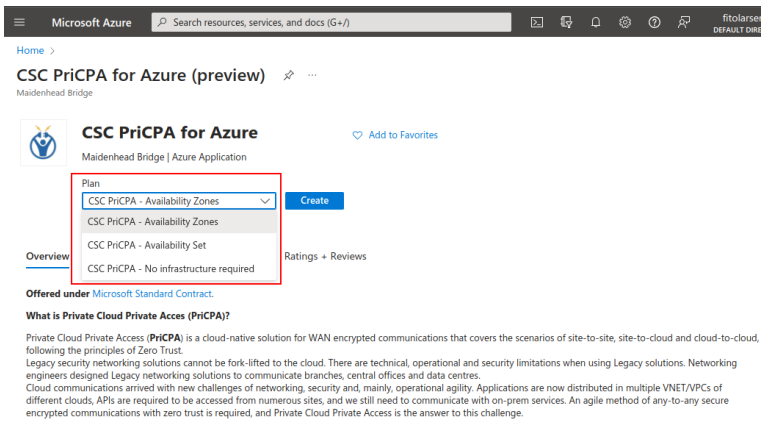
1. **(Optional) SSH Key:** If you want to access the CSC using SSH keys. If not, you can use a password during the installation.
2. **Virtual Network**
3. **External Subnet:** The External Subnet must be on the same Virtual Network as the Internal Subnet.
4. **Internal Subnet:** The Internal Subnet must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

5.2 Launching the CSC PriCPA from Azure Marketplace

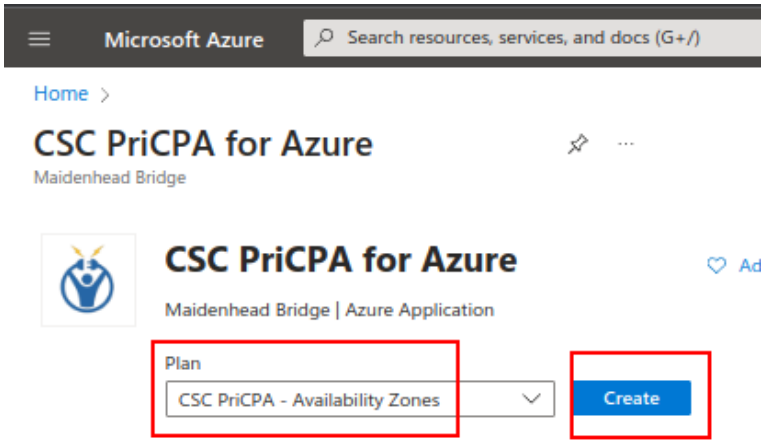
Go to Azure Marketplace, search for "Maidenhead Bridge", and select **"CSC PriCPA for Azure"**.



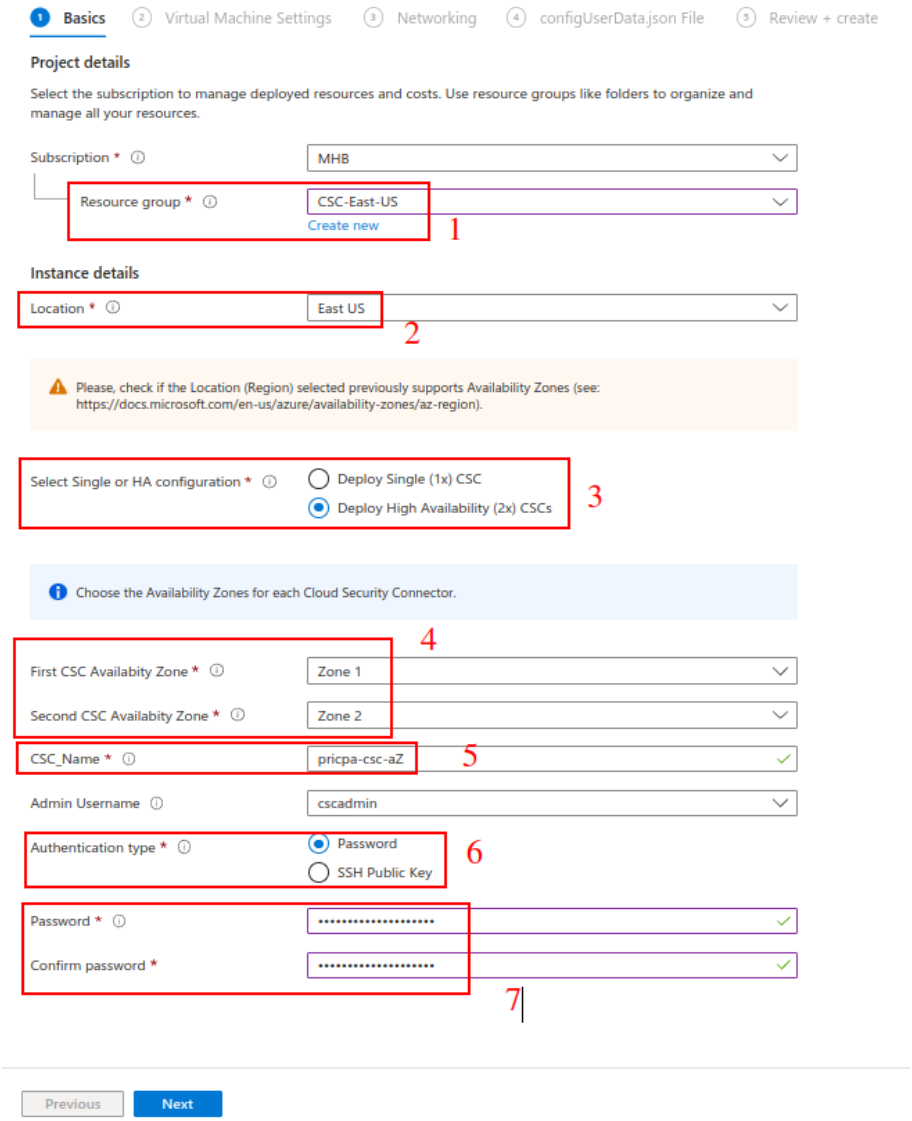
→ Click "Get it Now"



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



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

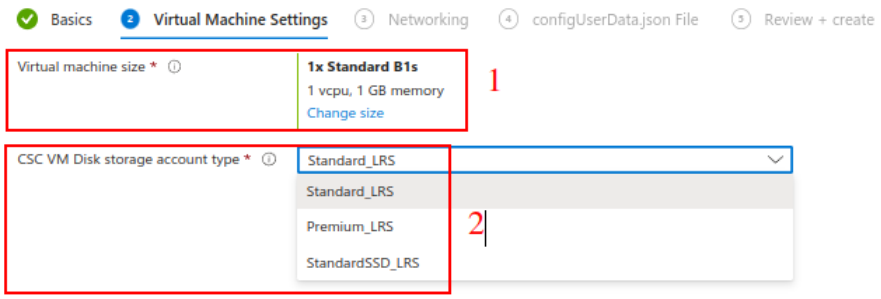


Fill the values on "Basics"

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

6. For the username "cscadmin", choose to use Password or SSH key.

→ Click "Next".



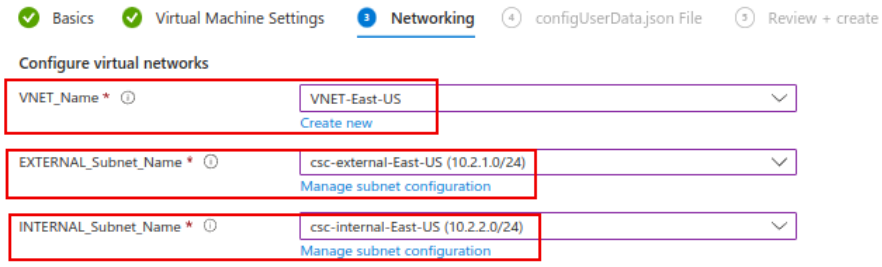
→ Select the Virtual Machine size and Storage Type.

Note:

The CSC PriCPA is a very light VM and can run on small machine sizes such as Standard B1s. If the CSC PriCPA is very loaded, we recommend increasing the CPU and Memory of the VM.

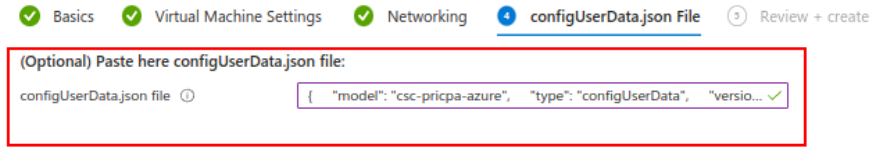
Microsoft created the VM Size "Standard Fx" family for Virtual Appliances. We recommend using the "Standard Fx" series. (i.e. Standard F2).

→ Click "Next"



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

(Optional) -> Paste configUserData.json file.



IMPORTANT: See Appendix B for format and examples of the configUserdata.json file.

Via configUserData.json file, you can pass values to parameters during the installation of the CSC. You can setup:

1. AWS SSM agent registration values.
2. DNS servers
3. Syslog servers and traffic log configuration.
4. PriCPA Local configuration values, Peers URL and Remote Management Networks.
5. SSH Restrictions via eth1 and wg0.
6. Admin Management: Enable csccli user and SSH Key.

configUserData.json (blank)

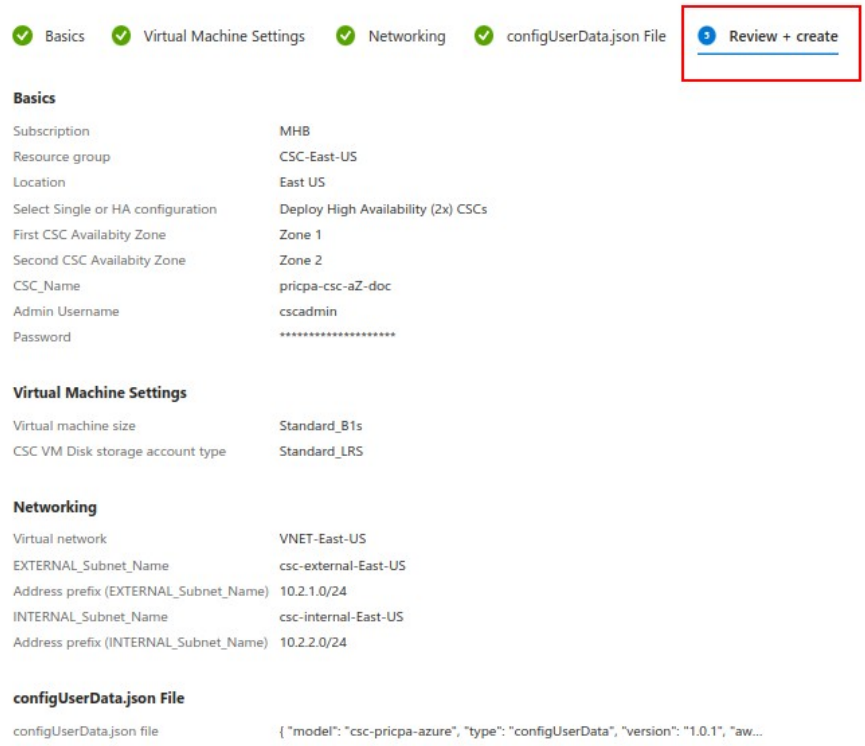
*The fields in **bold** are not configurable. So please, do not modify.*

configUserData.json

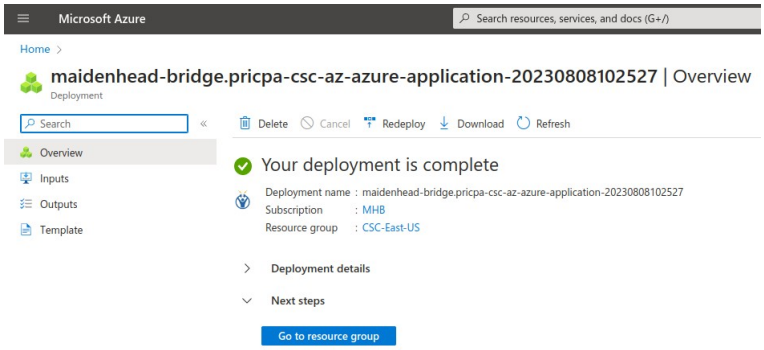
```
{
   "model": "csc-pricpa-azure",
   "type": "configUserData",
   "version": "1.0.1",
   "awsSsmAgent": {
      "enable": "no",
      "activationCode": "",
      "activationId": "",
      "awsRegion": ""
   },
   "dns": {
      "useCloudDns": "yes",
      "primaryDnsIp": "",
      "secondaryDnsIp": ""
   },
   "syslog": {
      "enable": "no",
      "primaryServer": {
         "ip": "",
         "port": ""
      },
      "secondaryServer": {
         "ip": "",
         "port": ""
      },
      "trafficLogs": {
         "enable": "no"
      }
   },
   "priCPA": {
      "enable": "no",
      "nodeName": "",
```

```
      "location": "",
      "description": "",
      "publicUdpPort": "51820",
      "privateCirdIp": "",
      "persistentKeepAlive": "no",
      "peersJsonFileUrl": "",
      "remoteManagementNetworks": []
   },
   "sshRestrictions": {
      "eth1": {
         "enable": "no",
         "allowedNetworks": []
      },
      "wg0": {
         "enable": "no",
         "allowedNetworks": []
      }
   },
   "adminManagement": {
      "csccli": {
         "enable": "no",
         "sshPublicKey": ""
      }
   }
}
```

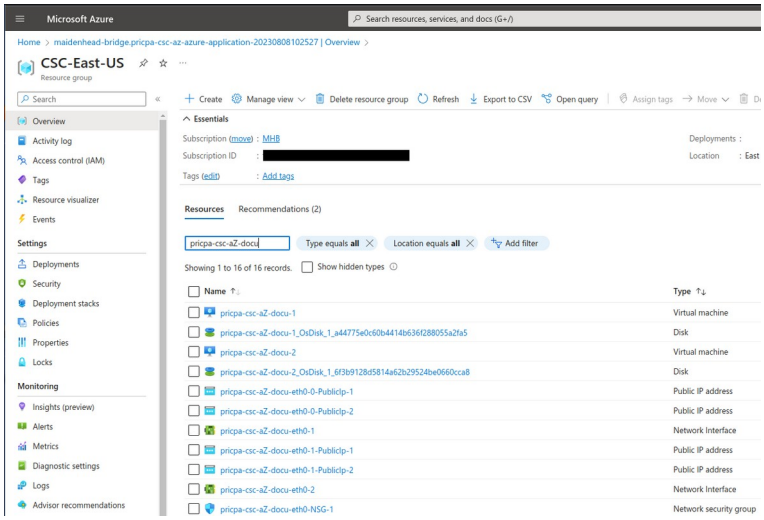

→ Click Next.



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



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

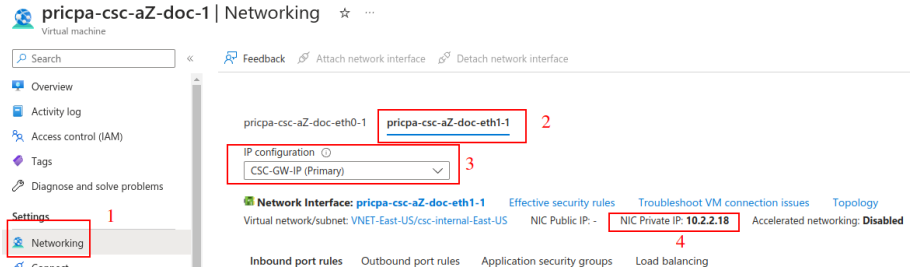


→ Done! Your CSCs PriCPA for Azure are deployed.

6 Accessing for first time to your CSC PriCPA

6.1 SSH to the Admin Console using CSC GW IP

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



2. In this example, “NIC Private IP” is: 10.2.2.18
3. From a machine inside the Virtual Network or via remotely via PriCPA, ssh the CSC using username “cscadmin” and key or password:

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

   ssh cscadmin@<eth1 Private IP>

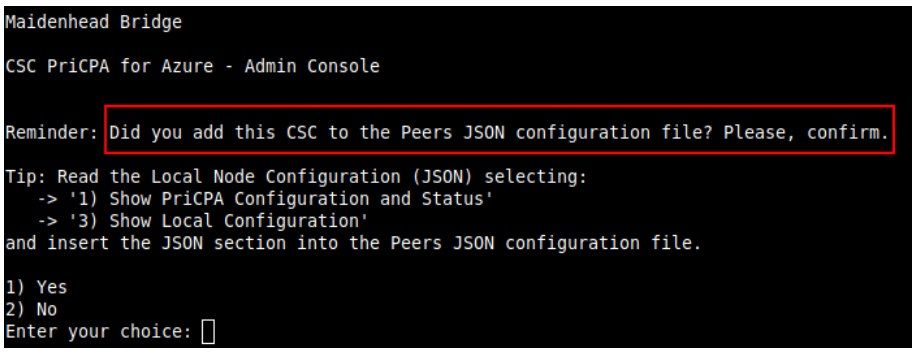
Important: Please, wait 2 minutes before to SSH the CSC to allow all processes to complete.

6.1.1 Initial Screen when using configUserData.json file

When passing PriCPA parameters via the configUserData.json file, the CSC will automatically do the Local Configuration (First Step) for PriCPA.

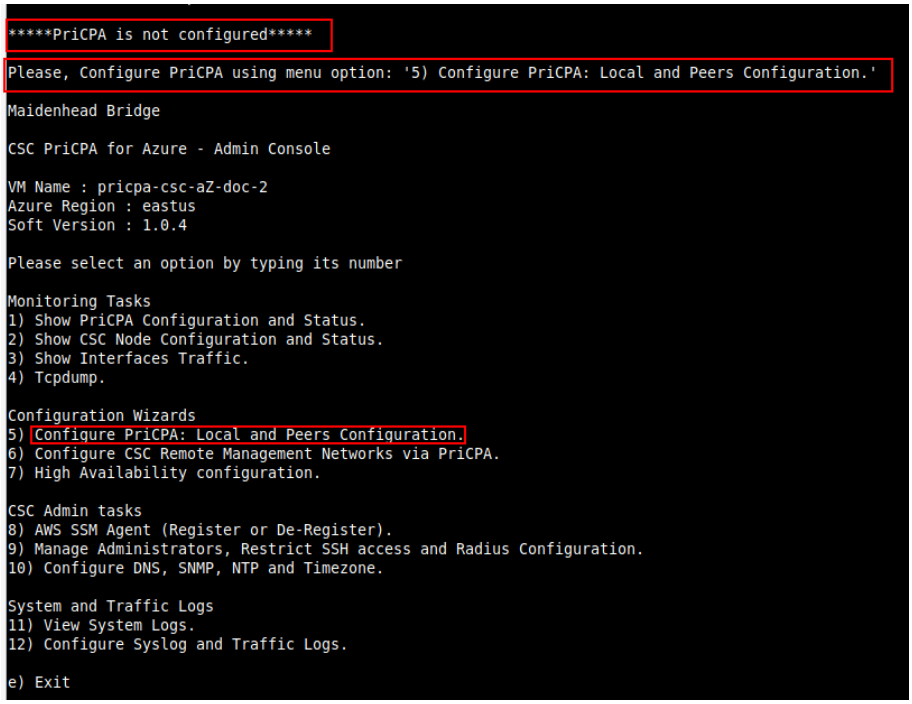
configUserData.json, example of priCPA values:

Initial screen:



The initial screen presents a reminder to do the Second Step of the PriCPA configuration: Add this CSC to the Peers JSON configuration file, including the "Local Configuration" plus "Subnets behind" this CSC and the "Private Apps". See section for "Configuring PriCPA" for details.

6.1.2 Initial screen without using configUserData.json file

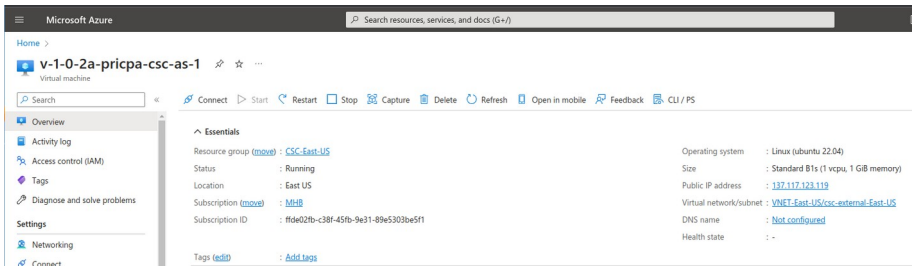


See section 8.1.2 for detailed configuration.

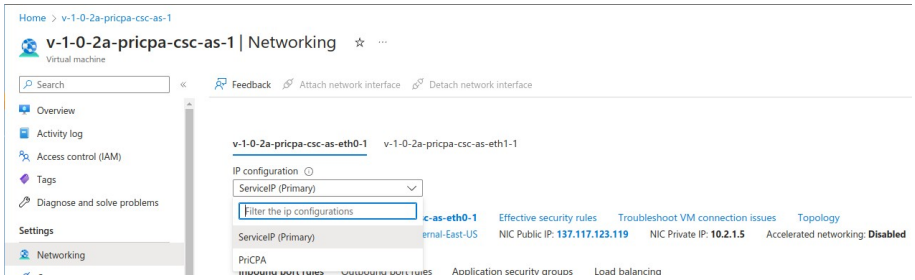
7 Resources creates by the ARM template

The following resources are created by the ARM template:

1. Virtual Machine



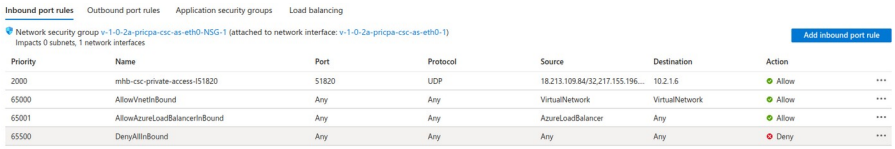
2. Interfaces External and Internal.



The CSC PriCPA has 2 Public IPs: ServiceIP and PriCPA. The ServiceIP is for outbound communications only to Azure CLI, Security Patches repositories, etc. The PriCPA is for Inbound/Outbound encrypted communications with other CSCs of the Private Cloud.

3. Security Group for External Interface. [3] [4]

3.1. Inbound Rules



3.2. Outbound Rules

3 The CSC contains Internal Firewall Rules on each interface that are more specific in some cases. Therefore, there is double protection: The Azure Security Group and the internal Firewall Rules of the CSC.

4 The CSC automatically updates the internal FW rules and Security Groups to allow Peers to communicate with each other. (Rules with index 2xxx)

Inbound port rules | **Outbound port rules** | Application security groups | Load balancing

Network security group v-1-0-2a-pricpa-csc-as-eth0-NSG-1 (attached to network interface: v-1-0-2a-pricpa-csc-as-eth0-1)
Impacts 0 subnets, 1 network interfaces

Add outbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
2010	mhb-csc-private-access-O51820	51820	UDP	10.2.1.6	18.213.109.84/32,217.155.196...	Allow
2020	mhb-csc-private-access-O51821	51821	UDP	10.2.1.6	82.68.6.74/32	Allow
4000	AllowPing	Any	ICMP	Any	Any	Allow
4030	AllowHTTP	80	TCP	Any	Any	Allow
4040	AllowHTTPS	443	TCP	Any	Any	Allow
4050	AllowPublicDNS	53	UDP	Any	Any	Allow
4060	AllowNTP	123	UDP	Any	Any	Allow
4070	DenyAllOutbound	Any	Any	Any	Any	Deny
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

4. Security Group for Internal Interface.

4.1. Inbound Rules

Inbound port rules | Outbound port rules | Application security groups | Load balancing

Network security group v-1-0-2a-pricpa-csc-as-eth1-NSG-1 (attached to network interface: v-1-0-2a-pricpa-csc-as-eth1-1)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
4000	AllowNet-10.0.0.0-8	Any	Any	10.0.0.0/8	Any	Allow
4010	AllowNet-172.16.0.0-12	Any	Any	172.16.0.0/12	Any	Allow
4020	AllowNet-192.168.0.0-16	Any	Any	192.168.0.0/16	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

4.2. Outbound Rules

Inbound port rules | **Outbound port rules** | Application security groups | Load balancing

Network security group v-1-0-2a-pricpa-csc-as-eth1-NSG-1 (attached to network interface: v-1-0-2a-pricpa-csc-as-eth1-1)
Impacts 0 subnets, 1 network interfaces

Add outbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

8 Configuring PriCPA

SSH the CSC to access the Admin Console and go to the section "Configuration Wizards".

```
Configuration Wizards
5) Configure PriCPA: Local and Peers Configuration.
6) Configure CSC Remote Management Networks via PriCPA.
7) High Availability configuration.
```

Select "5) Configure PriCPA: Local and Peers Configuration."

The configuration of PriCPA is four simple steps:

```
Selection: 5

Private Access Configuration Wizard


_____________________________________________________________

Steps to configure Private Access:

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

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

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

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

8.1 Create the Local configuration (First node of the HA pair or Single deployment)

8.1.1 Using configUserData.json file

You can pass the Local configuration parameters via configUserData.json file during the initial deployment.

Here an example:

```
"priCPA": {
    "enable": "yes",
    "nodeName": "pricpa-csc-aZ-doc",
    "location": "Azure East US",
    "description": "PriCPA Node for Documentation",
    "publicUdpPort": "51820",
    "privateCirdIp": "192.168.7.25/24",
    "persistentKeepAlive": "no",
    "peersJsonFileUrl": "https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
    "remoteManagementNetworks": [
        "172.19.0.0/24",
        "192.168.1.0/24",
        "192.168.6.0/24"
    ]
},
```

The CSC will read this information and create the Local configuration on the First node of the HA deployment or when it is a single deployment. As a result, the Local Configuration for PriCPA will be ready:

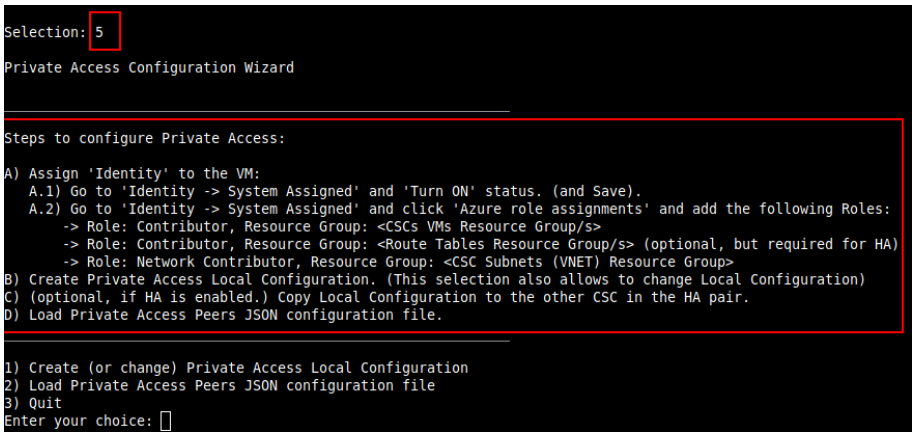
```
1) Create (or change) Private Access Local Configuration
2) Load Private Access Peers JSON configuration file
3) Quit
Enter your choice: 1

Identity Type: SystemAssigned

Private Access is enabled.

The current values configured are:

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

Token: UU5vUGUzSHMxMUxlYWlGcXpnTWxORnRHN0dvWXdjMWhOb2wzMndHU1VVRT0K

Private Access Local Config JSON file:

{
  "peers": [
    {
      "nodeName": "pricpa-csc-aZ-doc",
      "location": "Azure East US",
      "description": "PriCPA Node for Documentation",
      "publicKey": "gG0HbRzQ3Yn1UZByxeIY6cQJzn/k65A/IE5dbtjvHSU=",
      "publicIpAndUdpPort": "20.127.203.54:51820",
      "privateCirdIp": "192.168.7.25/24",
      "persistentKeepAlive": "no",
      "networks": [],
      "privateApps": []
    }
  ]
}

Please, select next action:

1) Change configuration
2) Reset to default values
3) Restart Service
4) Quit
Enter your choice:
```

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

8.1.2 Manual Configuration

→ From Main Menu, select "5) Configure PriCPA: Local and Peers Configuration."

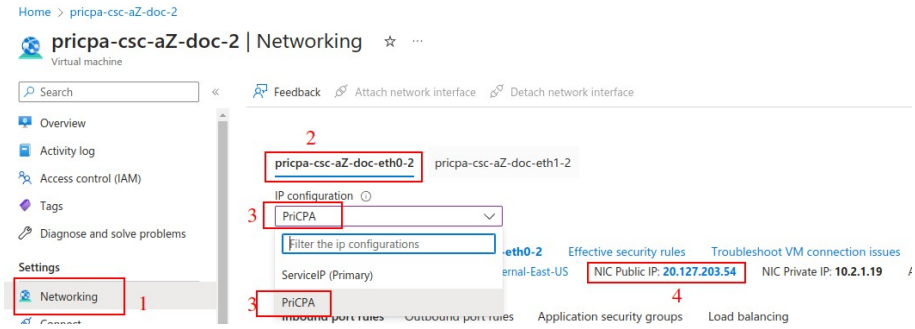
→ Select "1) Create (or change) Private Access Local Configuration"



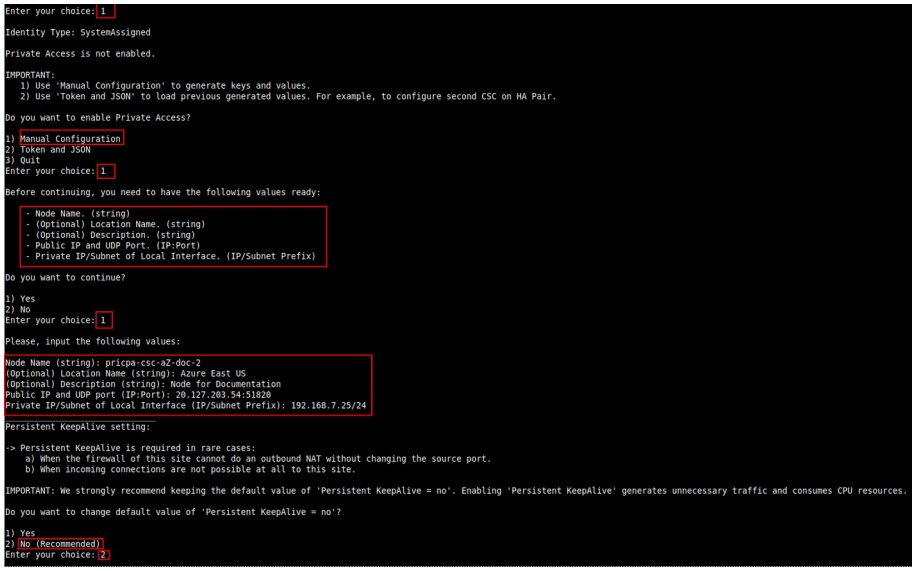
→ Select "1) Manual Configuration" and input the values requested.

Note: The Public IP to input here is the labelled "PriCPA" on External Interface.

Public IP for PriCPA:



- Run Wizard



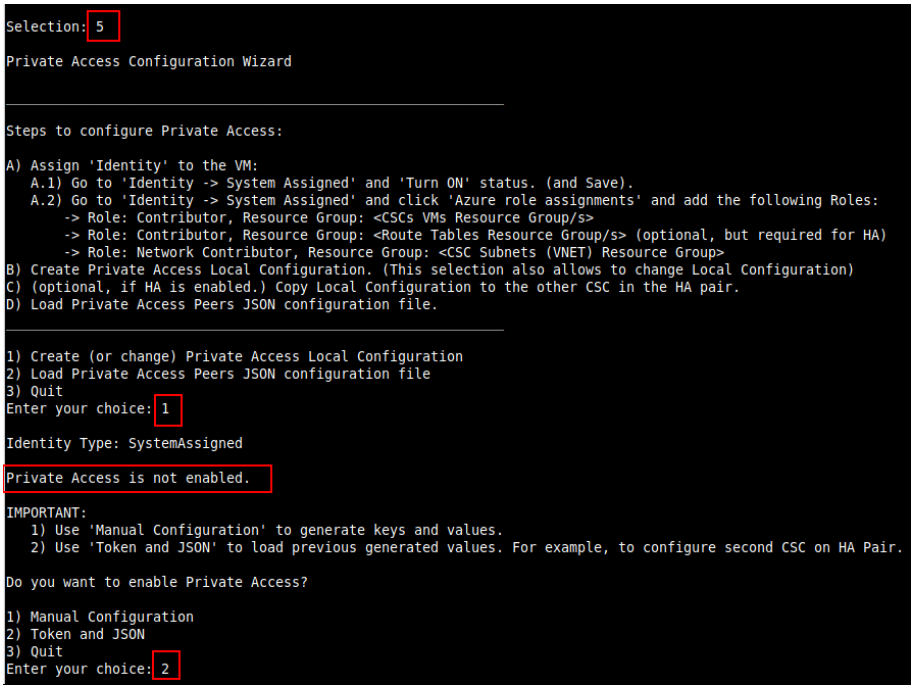
- Apply values

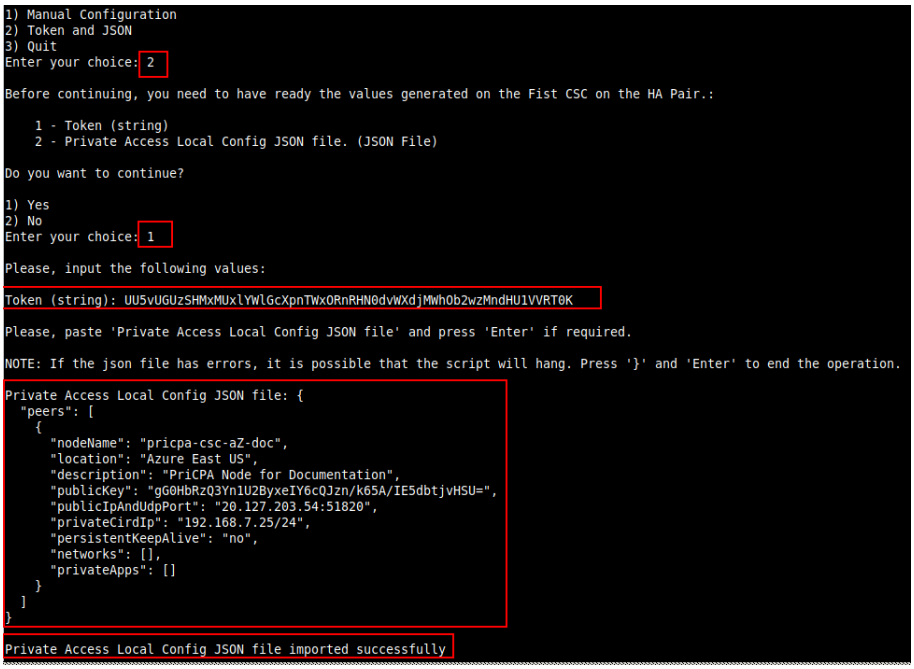
IMPORTANT: *The "Token" and "Private Access Local Config JSON file" will be used to create the local configuration on the second node of the HA pair. Please, keep these values in a safe place. You can use these values to reconfigure any node of the HA Pair if necessary in the future. For example, if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

8.1.3 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 Local Config JSON file imported successfully


The values to configure are:

Node Name: "pricpa-csc-aZ-doc"
Public IP and UDP Port: 20.127.203.54:51820
Private IP/Subnet of Local Interface: 192.168.7.25/24
Location Name: "Azure East US"
Description: "PriCPA Node for Documentation"
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 pricpa-csc-aZ-doc-2.


Press Enter to continue...
```

8.2 Create the Private Access Peers JSON file

The Private Access Peers JSON file contains:

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

Here some examples.

8.2.1 Full mesh Private Access Peers JSON file

Consider the following example:

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

The Local Config JSON file of each node is:

ns-cgc00001

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

ns-cgc00002

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

ns-cgc00003

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

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

Basic Peers Configuration JSON file

```
{
   "peers": [
      {
         "nodeName": "ns-cgc00001",
         "description": "Node on VMware Server 1",
         "location": "HQ",
         "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
         "publicIpAndUdpPort": "200.1.1.1:51821",
         "privateCirdIp": "192.168.7.1/24",
         "persistentKeepAlive": "no",
          "networks": [
             "10.1.1.0/24",
             "10.1.2.0/24"
          ],
         "privateApps": []
      },
      {
         "nodeName": "ns-cgc00002",
         "description": "Node on VMware Server 2",
         "location": "Datacentre 2",
         "publicKey": "xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
         "publicIpAndUdpPort": "200.1.1.2:51821",
         "privateCirdIp": "192.168.7.2/24",
         "persistentKeepAlive": "no",
          "networks": [
             "10.2.1.0/24",
             "10.2.2.0/24"
          ],
         "privateApps": []
      },
      {
         "nodeName": "ns-cgc00003",
         "description": "Node on VMware Server 3",
```

```
        "location": "Branch",
        "publicKey": "TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=",
        "publicIpAndUdpPort": "200.1.1.3:51821",
        "privateCirdIp": "192.168.7.3/24",
        "persistentKeepAlive": "no",
         "networks": [
            "10.3.1.0/24",
            "10.3.2.0/24"
         ],
        "privateApps": []
    }
  ]
}
```

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

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

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

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

ns-cgc00001

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

```
            ],
            "destinationSinglePorts": [
               ""
            ],
            "destinationPortRange": {
               "fromPort": "",
               "toPort": ""
            }
         }
      ]
   },
```

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

Now, completing our "Peers Configuration JSON file":

Full Mesh Peers Configuration JSON file.

```
{
   "peers": [
      {
         "nodeName": "ns-cgc00001",
         "description": "Node on VMware Server 1",
         "location": "HQ",
         "publicKey": "yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=",
         "publicIpAndUdpPort": "200.1.1.1:51821",
         "privateCirdIp": "192.168.7.1/24",
         "persistentKeepAlive": "no",
         "networks": [
            "10.1.1.0/24",
            "10.1.2.0/24"
         ],
          "privateApps": [
             {
                "description": "Allow all traffic to this site",
                "ipProtocol": "all",
                "sourceCirdIp": [
                   "0.0.0.0/0"
                ],
                "destinationCirdIp": [
                   "10.1.1.0/24",
                   "10.1.2.0/24"
                ],
                "destinationSinglePorts": [
                   ""
                ],
                "destinationPortRange": {
                   "fromPort": "",
                   "toPort": ""
                }
             }
          ]
      },
      {
         "nodeName": "ns-cgc00002",
         "description": "Node on VMware Server 2",
         "location": "Datacentre 2",
         "publicKey": "xTlBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=",
         "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": "TrMvSoP4jYQlY6RlzBgbssQqY3vxI2Pi+y71lOWWXX0=",
    "publicIpAndUdpPort": "200.1.1.3:51821",
    "privateCirdIp": "192.168.7.3/24",
    "persistentKeepAlive": "no",
    "networks": [
        "10.3.1.0/24",
        "10.3.2.0/24"
    ],
    "privateApps": [
        {
            "description": "Allow all traffic to this site",
            "ipProtocol": "all",
            "sourceCirdIp": [
                "0.0.0.0/0"
            ],
            "destinationCirdIp": [
                "10.3.1.0/24",
                "10.3.2.0/24"
            ],
            "destinationSinglePorts": [
                ""
            ],
            "destinationPortRange": {
                "fromPort": "",
                "toPort": ""
            }
        }
    ]
  }
 ]
}
```


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

8.2.2 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 [5]: 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.*

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

8.2.2.1 Example of "privateApps" for a Windows Domain controller

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

The port information was taken from this article:

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

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

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

8.2.2.2 Example of "privateApps" for Internal Web Server.

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

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

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

```
]
```

8.2.3 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 or Azure CLI commands.
3. Manual: Copy/Paste the JSON file on each CSCs.

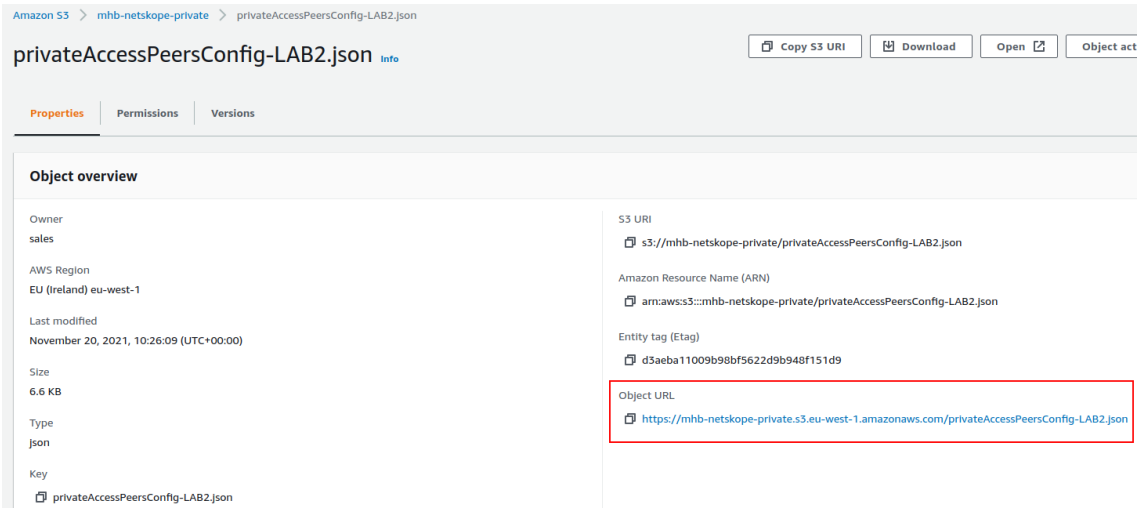
In this section we are going to explain two methods: URL and Manual Copy. The DevOps method is explained on Section12: DevOps operations. (TBC)

8.2.3.1 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[6] or similar. Obtain the download URL.

Example of AWS bucket:

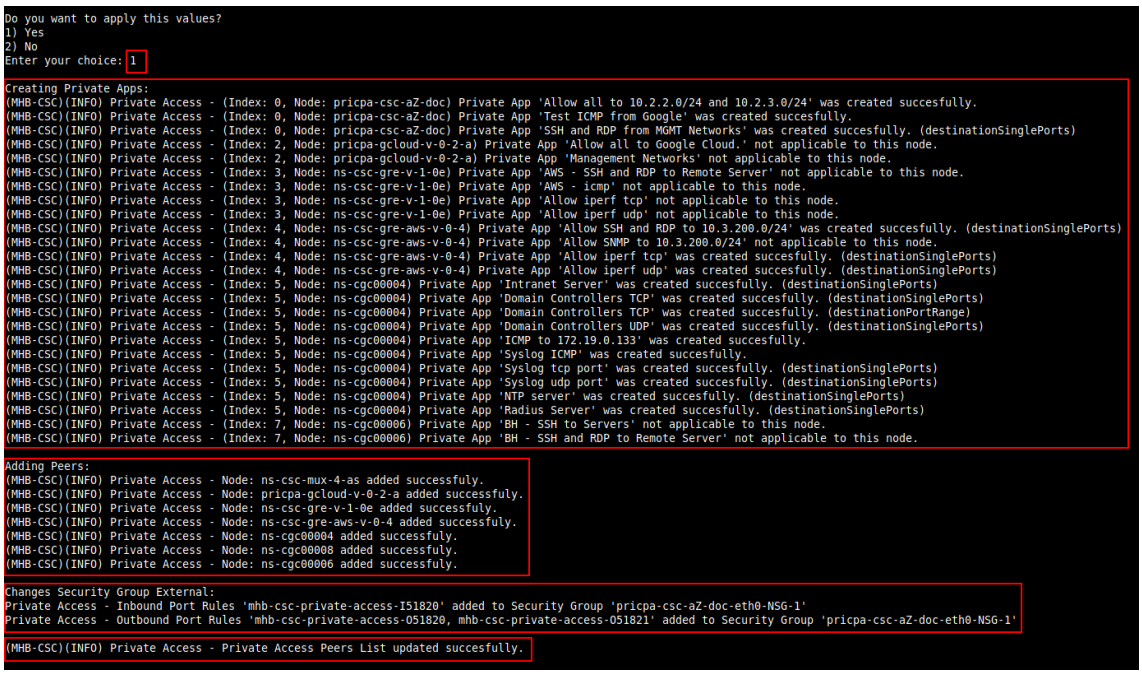


2. Configure the URL on each CSC.

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

SSH the CSC and go to Main Menu -> 5) Configure PriCPA: Local and Peers Configuration, then 2) Load Private Access Peers JSON configuration file

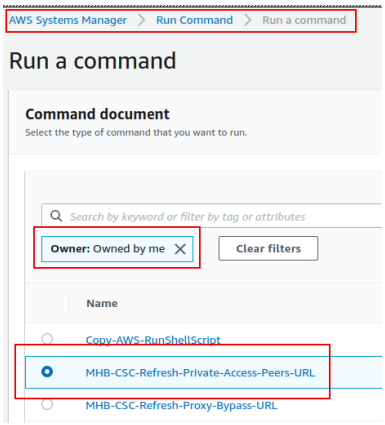
At this moment, you have the option to review the privateApps to configure in Compact or JSON format and to apply the values.



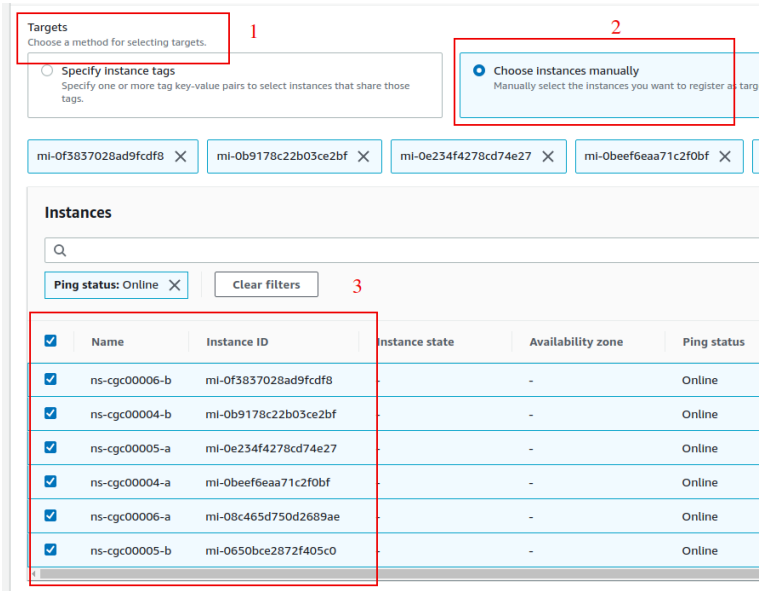
3. The next time you want to refresh the Private Access Peers JSON file, update the file, deploy it on the same location URL and Run Command: "Refresh Private Access Peers URL" using AWS SSM Agent or Rundeck or similar.

AWS System Manager:

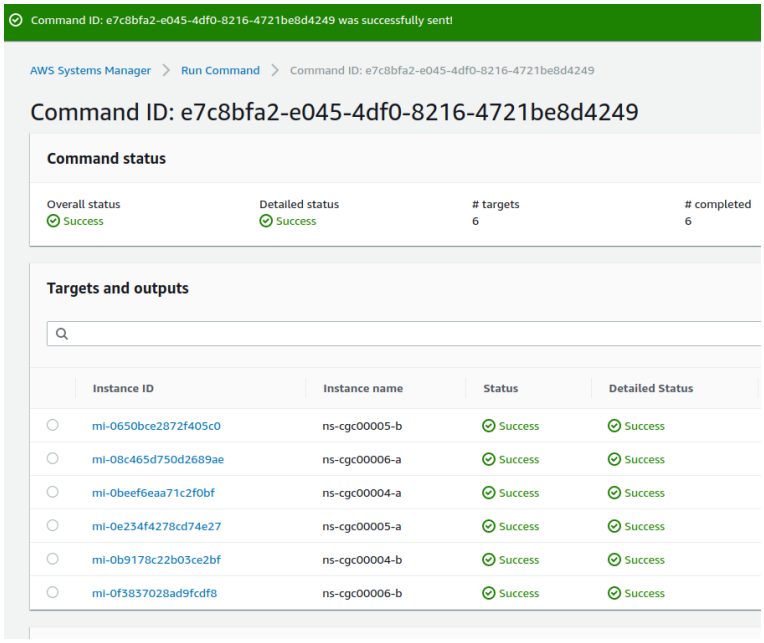
- Go to AWS Systems Manager -> Run Command -> and Select "MHB-CSC-Refresh-Private-Access-Peers-URL"



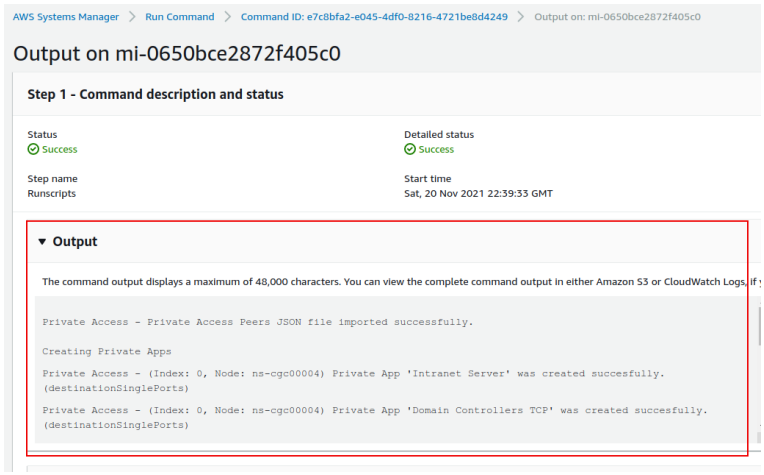
- Move down the screen and select all CSCs:



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

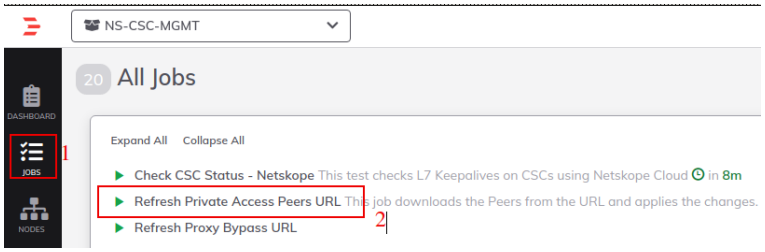


- To see the individual result, right click on the Instance ID and open it on a new TAB. Check the "Output"

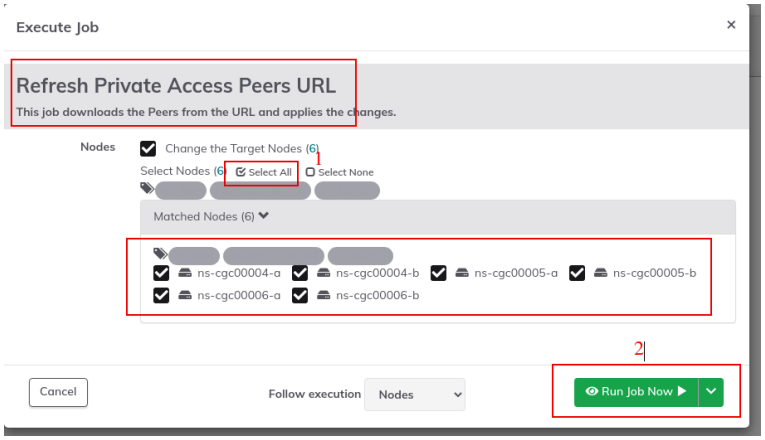


Using Rundeck

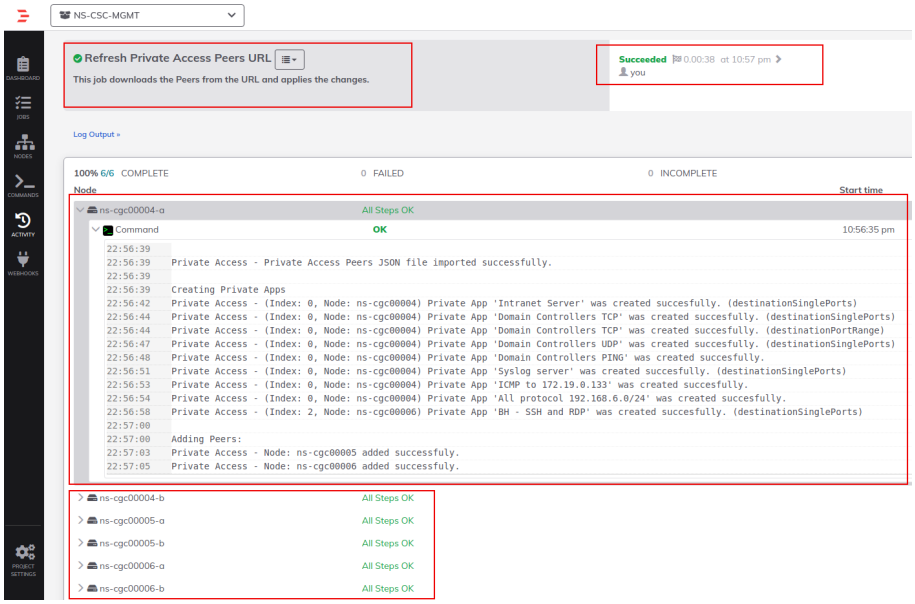
- Go to the Project <name> -> All Jobs -> Run " Refresh Private Access Peers URL"



- Select ALL nodes and click Run.



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



8.2.3.2 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/rw9sSunY780kljTZ1g=",
            "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-IS1280' added to Security Group 'zs-csc-mux-4-as-d-eth0-NSG-2'
Private Access - Outbound Port Rules 'mhb-csc-private-access-OS1820, mhb-csc-private-access-OS1821' 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!

8.3 Configure CSC Remote Management via Private Access.

If you want to access the CSC via PriCPA (not via eth1), you need to define the remote IP or Subnet. Also, you need to define the remote IP or Subnet that will be reached via PriCPA from the CSC. Examples are:

- SSH (Source IP/Subnet)
- SNMP (Source IP/Subnet)
- SIEM / Syslog (Destination IP/Subnet)

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

The configuration will be:

```
Configuration Wizards
5) Configure PriCPA: Local and Peers Configuration.
6) Configure CSC Remote Management Networks via PriCPA.
7) High Availability configuration.

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: 6

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

No Management Networks are configured.

Do you want to configure Management Networks?

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

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

Do you want to add another Management Network?

1) Yes
2) No
Enter your choice: 2

Management Networks to configure:

Management Networks Qty = 1
Management Network= 172.19.0.0/24

Do you want to apply changes?

1) Yes
2) No
Enter your choice: 1
Private Access - Management Network 172.19.0.0/24 was added on pricpa-csc-aZ-doc-1
```

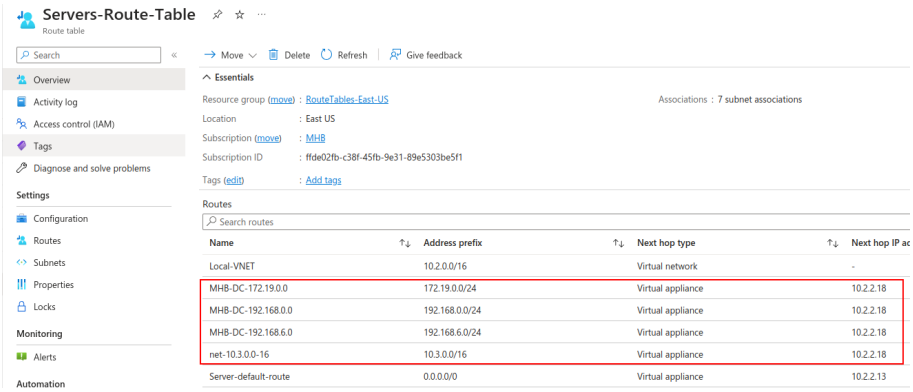
You can add multiple "Management Networks".

9 High Availability configuration

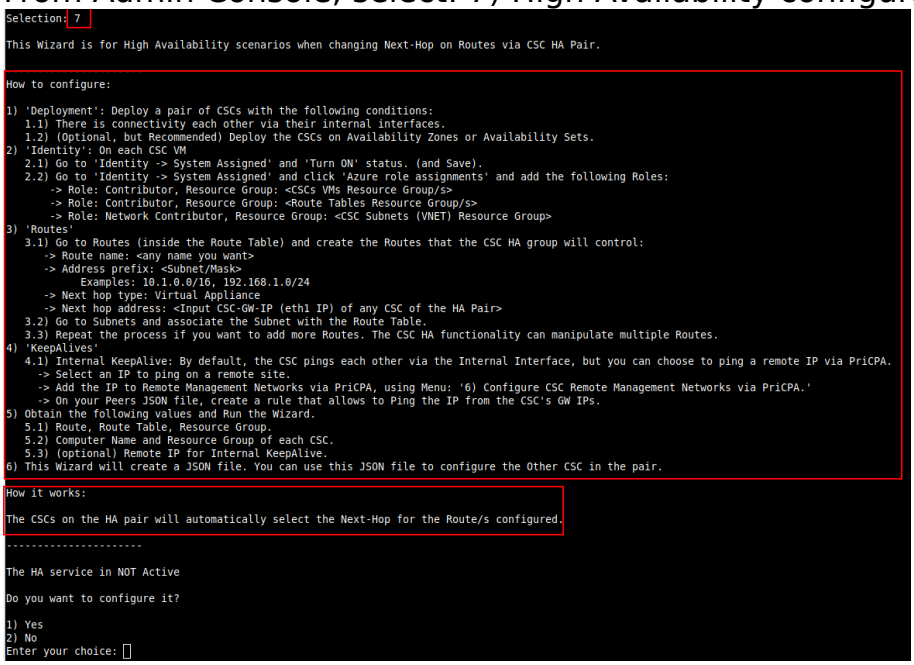
When deployed as High Availability pair, the CSCs can manage the "Next-Hop" of the Azure route/s configured.

The CSC active will assign its CSC GW IP as the "Next-Hop" of the routes configured. You can configure several routes; there is no limit.

For example:



From Admin Console, select: 7) High Availability configuration.



Help provided:

How to configure:

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

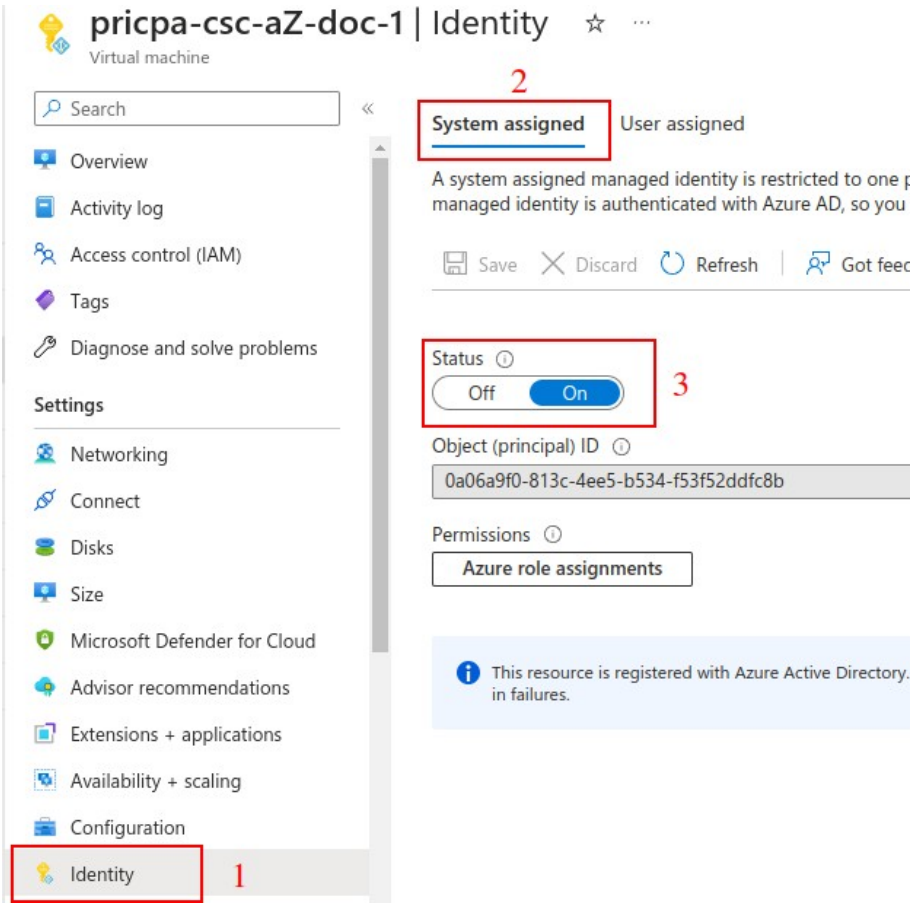
1.1) There is connectivity each other via their internal interfaces.

1.2) (Optional, but Recommended) Deploy the CSCs on Availability Zones or Availability Sets.

2) 'Identity': On each CSC VM

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

Example:



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

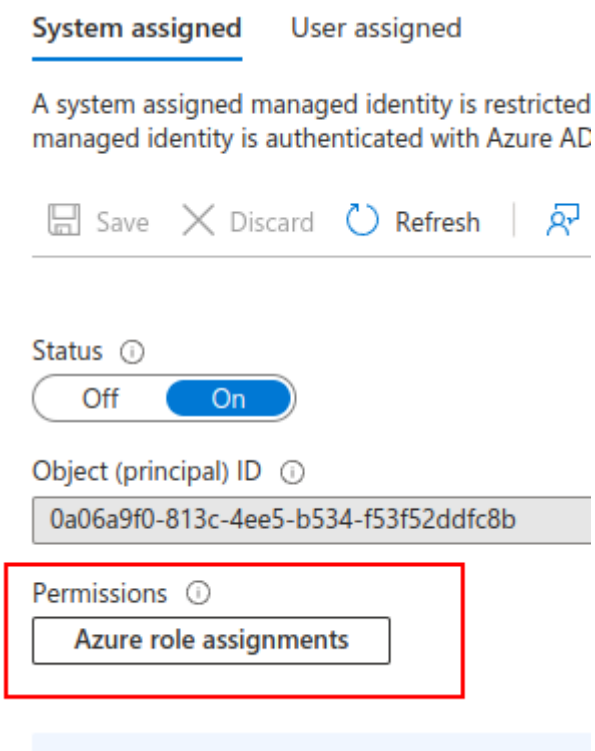
2.2) Go to 'Identity -> System Assigned' and click 'Azure role assignments' and add the following Roles:

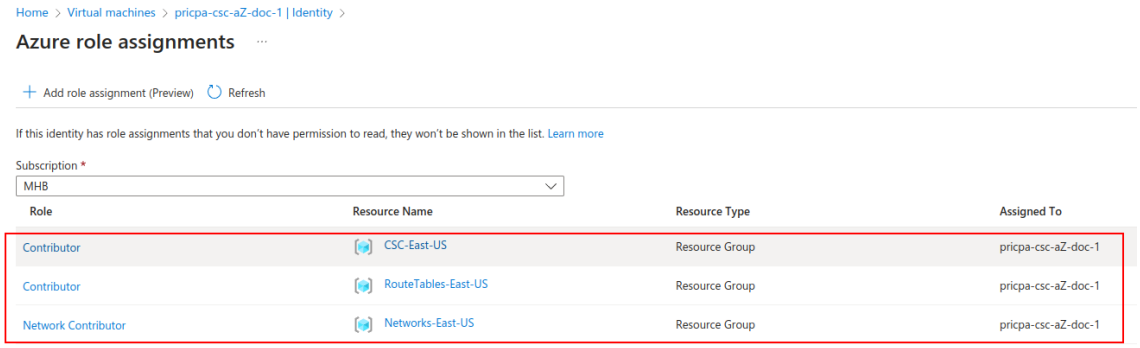
-> Role: Contributor, Resource Group: <CSCs VMs Resource Group/s>

-> Role: Contributor, Resource Group: <Route Tables Resource Group/s>

-> Role: Network Contributor, Resource Group: <CSC Subnets (VNET) Resource Group>

Example:





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

3) 'Routes'

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

-> Route name: <any name you want>

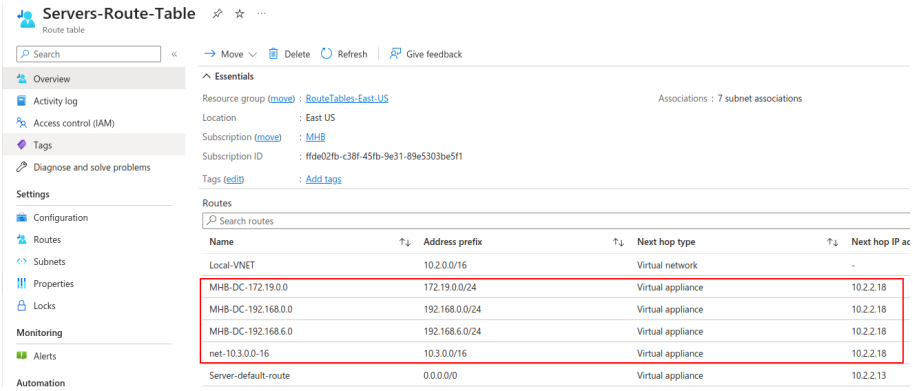
-> Address prefix: <Subnet/Mask>

Examples: Examples: 10.1.0.0/16, 192.168.1.0/24

-> Next hop type: Virtual Appliance

-> Next hop address: <Input CSC-GW-IP (eth1 IP) of any CSC of the HA Pair>

Example:



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

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

4) 'KeepAlives'

4.1) Internal KeepAlive: By default, the CSC pings each other via the Internal Interface, but you can choose to ping a remote IP via PriCPA.

-> Select an IP to ping on a remote site.

-> Add the IP to Remote Management Networks via PriCPA, using Menu: '6) Configure CSC Remote Management Networks via PriCPA.'

-> On your Peers JSON file, create a rule that allows to Ping the IP from the CSC's GW IPs.

5) Obtain the following values and Run the Wizard.

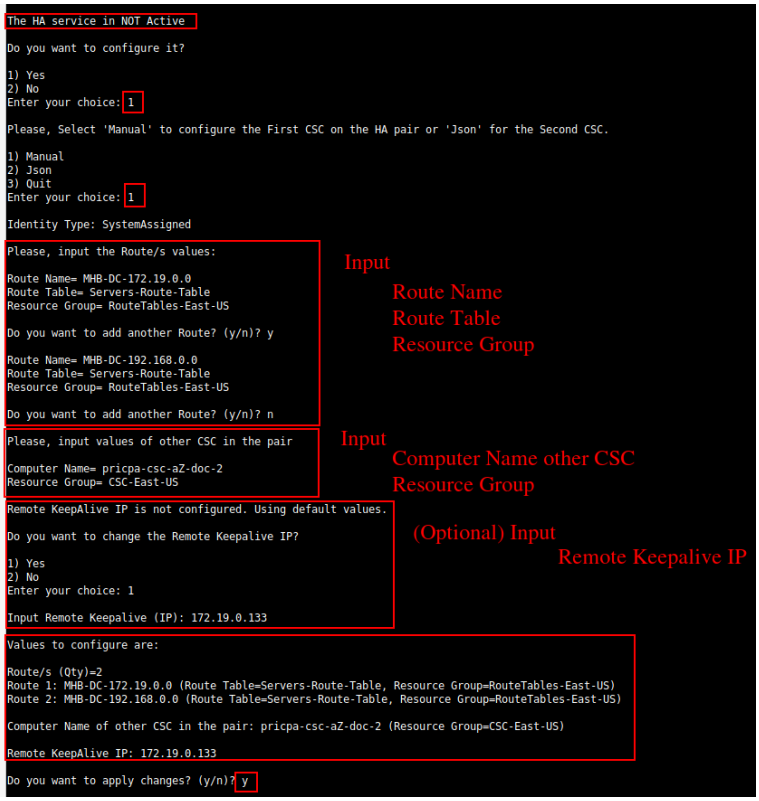
5.1) Route, Route Table, Resource Group.

5.2) Computer Name and Resource Group of each CSC.

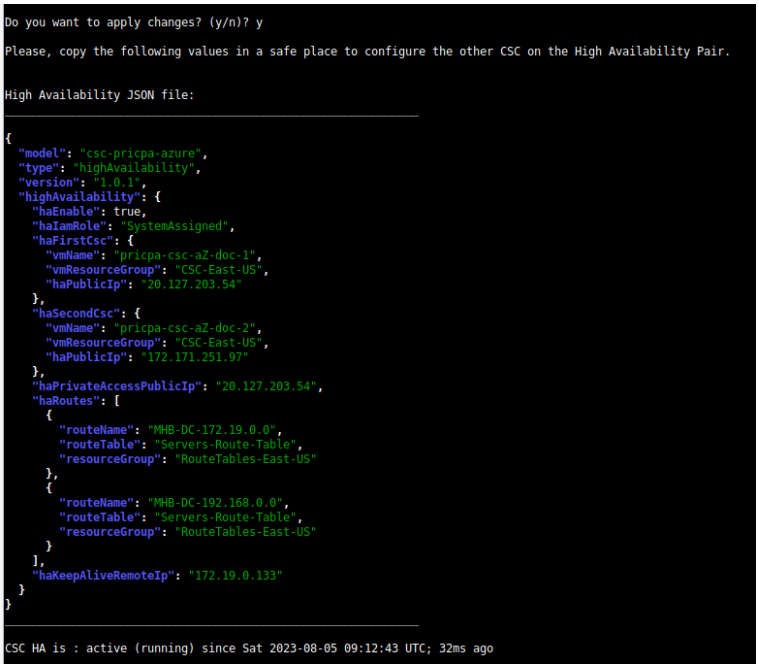
5.3) (optional) Remote IP for Internal KeepAlive.

Example:

First CSC on the HA Pair – Manual Configuration:

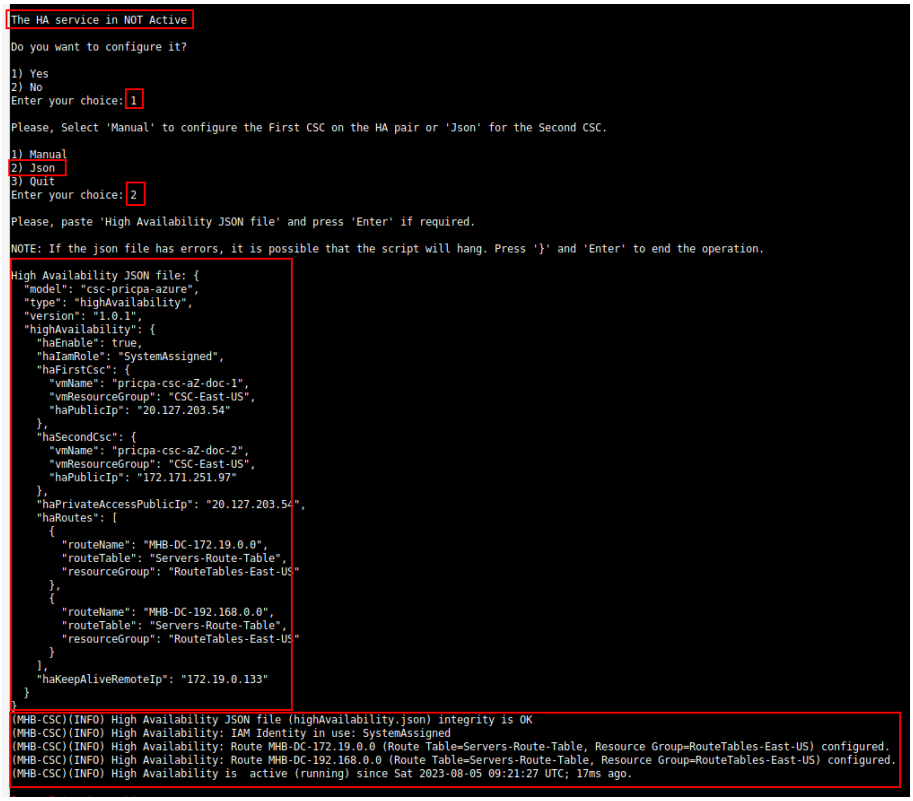


This Wizard will create a JSON file. You can use this JSON file to configure the Other CSC in the pair.



Example:

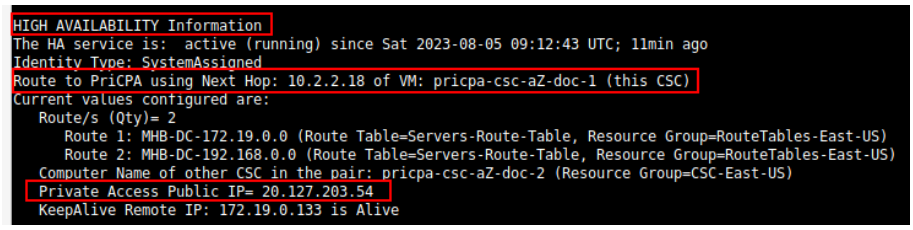
Second CSC on the HA Pair – (paste) JSON Configuration:



How it works:

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

Check HA using -> 2) Show CSC Node Configuration and Status.:



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

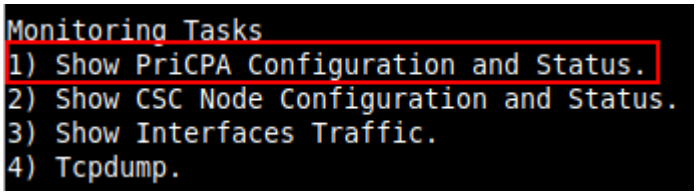
Logs generated by High Availability:

```
Aug  5 09:14:35 cscadmin: (MHB-CSC)(INFO) Route to PriCPA using Next Hop: 10.2.2.18 of CSC: pricpa-csc-aZ-doc-1
```

10 Show Configurations and Status Private Access.

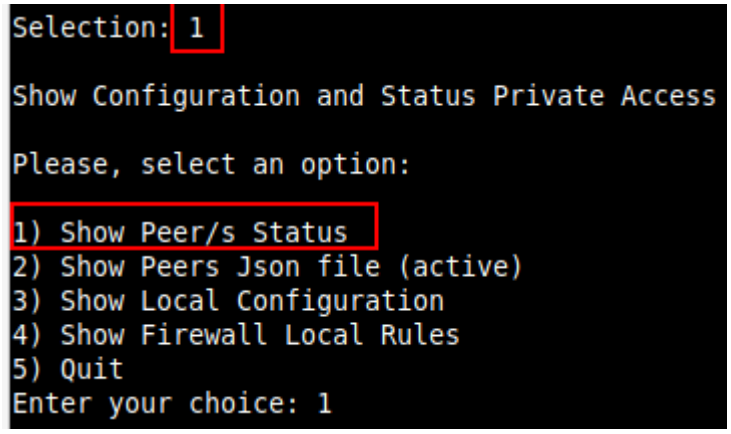
10.1 Using SSH Admin console

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

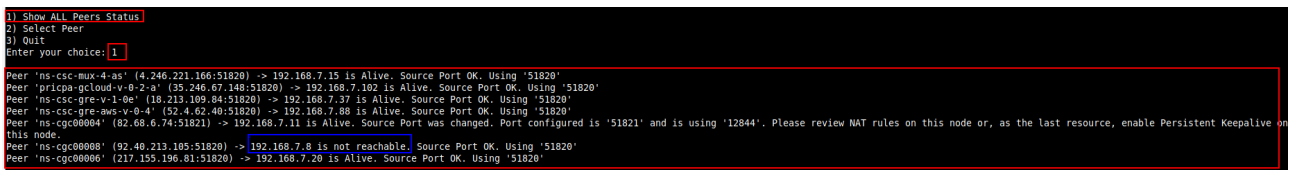


10.1.1 Show Peer/s Status

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



1. Show All Peers Status



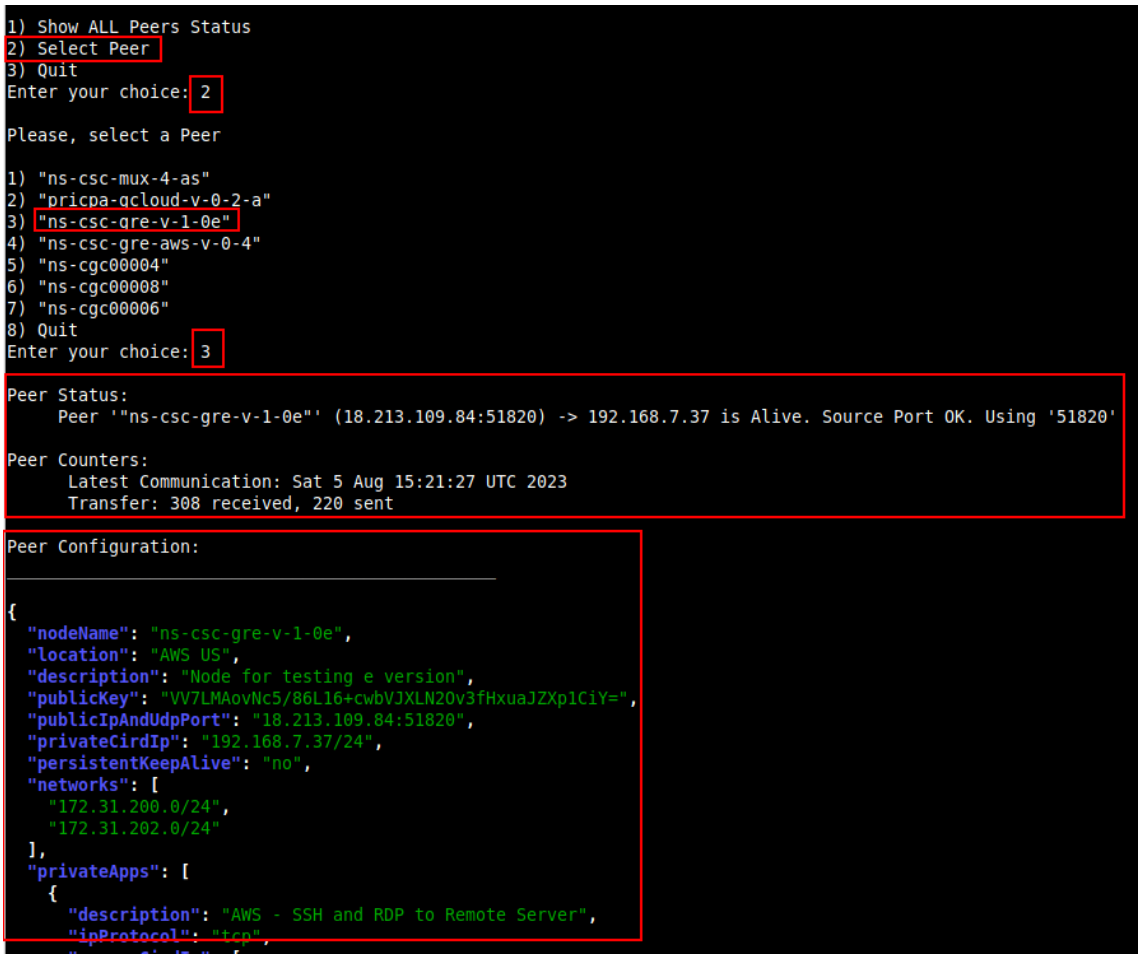
IMPORTANT: This section shows if the Peer is Alive and the "Source Port" that arrives at this node from the Peer. The Source Port information is essential to validate that the NAT on the Remote Peer is correct or if the FW on the other end is changing the Source Port. Please, correct the NAT on the remote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

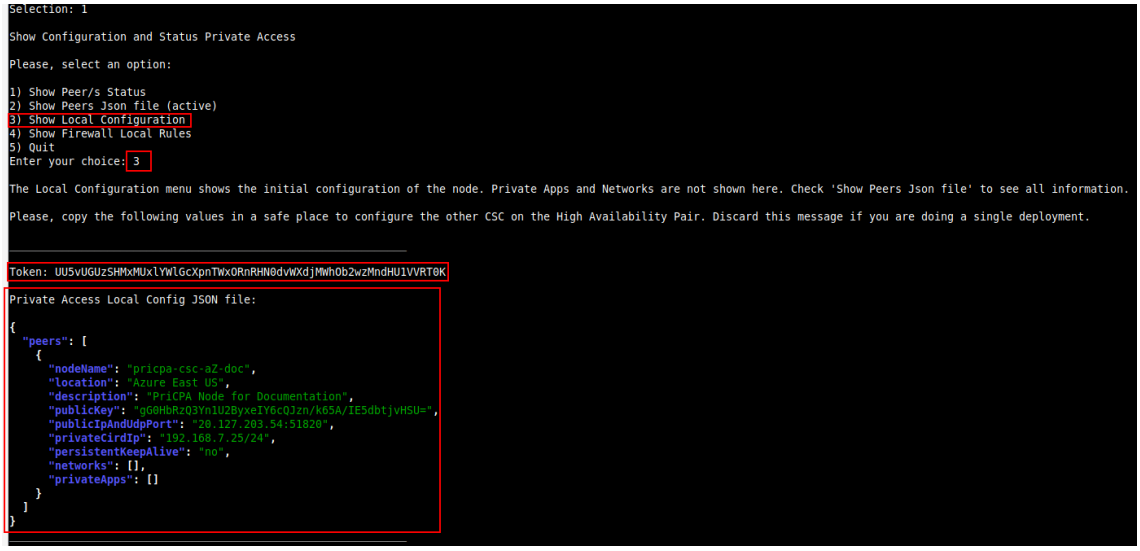
10.1.2 Show Peers Json file (active)

This menu shows the active Private Access Peers Json file.



10.1.3 Show Local Configuration

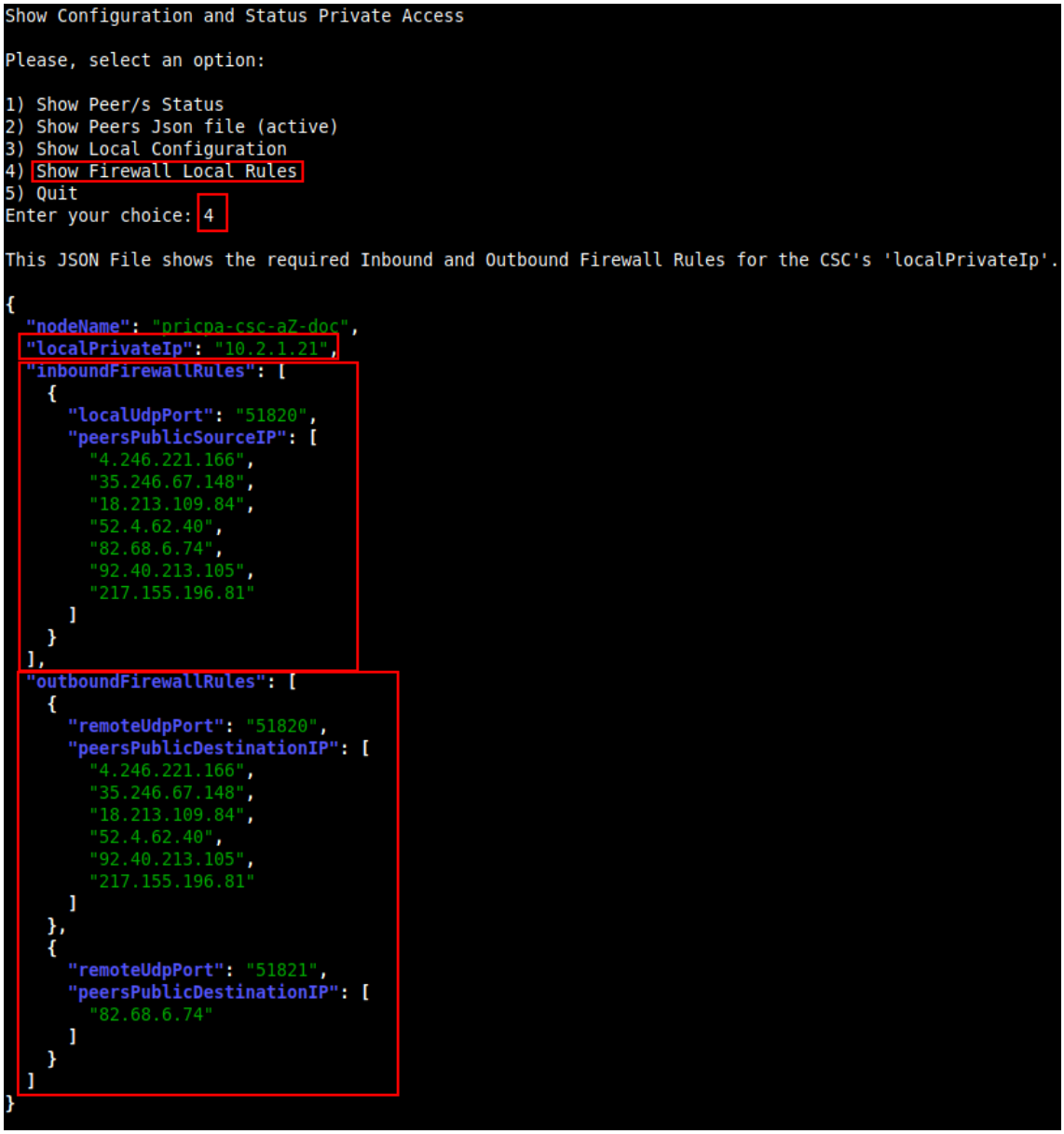
This menu shows the Local configuration of the node: Token and Local JSON File.



10.1.4 Show Firewall Local Rules

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

> ***Note:*** *The CSC 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

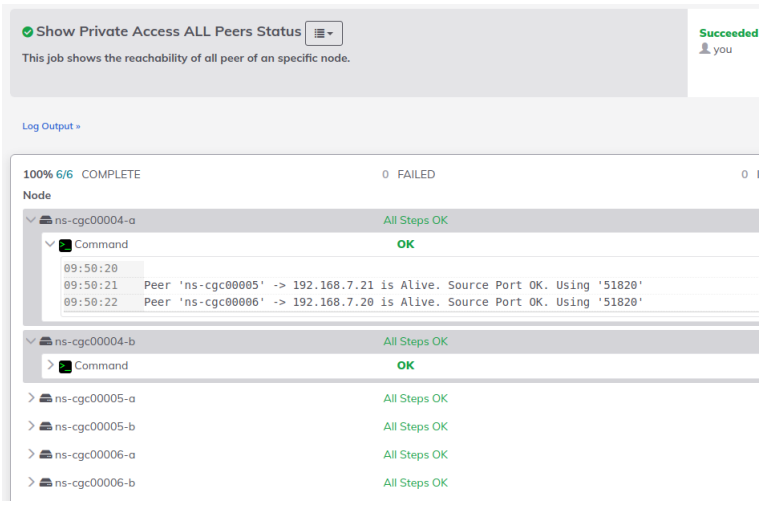
In this case, the information provided is only "Show ALL Peer Status"

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

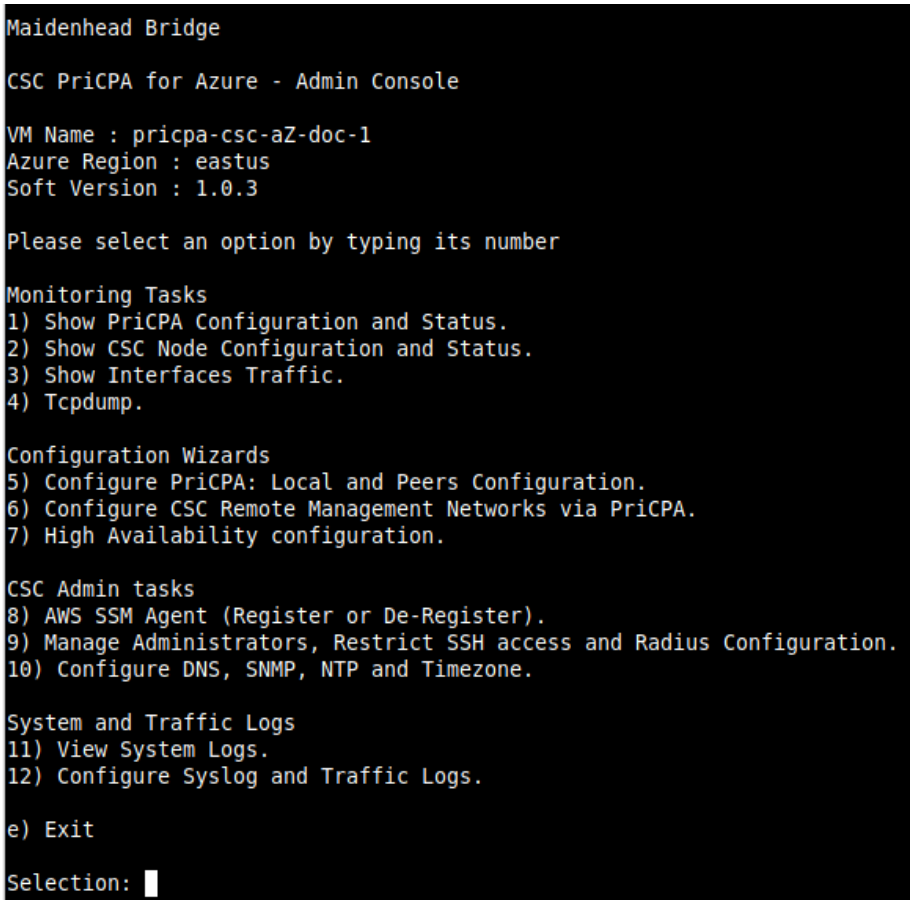
On Rundeck, run Job: "Show Private Access ALL Peers Status". Select the nodes. The output will show:



11 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 and run tcpdump.
- **Configuration Wizards:** Configure PriCPA, Remote Management Networks and HA.
- **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.

11.1 Monitoring Tasks

11.1.1 Show PriCPA Configuration and Status.

See previous section for detailed information.

11.1.2 Show CSC Node Configuration and Status.

```
Selection: 2

GENERAL INFORMATION
Name: pricpa-csc-aZ-doc-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_B1s
CSC date: Sat 5 Aug 15:58:42 UTC 2023
Soft version: 1.0.3 | CSC Model: CSC PriCPA for Azure
Azure Cloud: AzureCloud
Availability Zone : 1

INTERFACES INFORMATION
External: Service IP (eth0): 10.2.1.20/24 | PriCPA IP: 10.2.1.21 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.18/24 | Network Gateway: 10.2.2.1

PUBLIC IP Address INFORMATION
Service IP: 20.127.203.20
PriCPA Public IP: 20.127.203.54

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

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"}

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

HIGH AVAILABILITY Information
The HA service is:  active (running) since Sat 2023-08-05 09:12:43 UTC; 6h ago
Identity Type: SystemAssigned
Route to PriCPA using Next Hop: 10.2.2.18 of VM: pricpa-csc-aZ-doc-1 (this CSC)
Current values configured are:
   Route/s (Qty)= 2
      Route 1: MHB-DC-172.19.0.0 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
      Route 2: MHB-DC-192.168.0.0 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
   Computer Name of other CSC in the pair: pricpa-csc-aZ-doc-2 (Resource Group=CSC-East-US)
   Private Access Public IP= 20.127.203.54
   KeepAlive Remote IP: 172.19.0.133 is Alive
```

11.1.2.1 GENERAL INFORMATION

This section contains general information about the instance:

```
GENERAL INFORMATION
Name: pricpa-csc-aZ-doc-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_B1s
CSC date: Sat 5 Aug 15:58:42 UTC 2023
Soft version: 1.0.3 | CSC Model: CSC PriCPA for Azure
Azure Cloud: AzureCloud
Availability Zone : 1
```

11.1.2.2 INTERFACES INFORMATION

This section contains the interfaces information:

```
INTERFACES INFORMATION
External: Service IP (eth0): 10.2.1.20/24 | PriCPA IP: 10.2.1.21 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.18/24 | Network Gateway: 10.2.2.1
```

11.1.2.3 PUBLIC IP Address INFORMATION

This section shows the Service Public IP and the PriCPA IP.

```
PUBLIC IP Address INFORMATION
Service IP: 20.127.203.20
PriCPA Public IP: 20.127.203.54
```

Note: *When the CSC is on HA but not active, the PriCPA Public IP shown is not the "PriCPA Floating IP". The "PriCPA Floating IP" is displayed if the CSC is the HA active.*

11.1.2.4 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): 1.1.1.1 is Alive
DNS Server (2): 8.8.8.8 is Alive
```

11.1.2.5 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"}
```

11.1.2.6 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): 172.19.0.5/5514 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

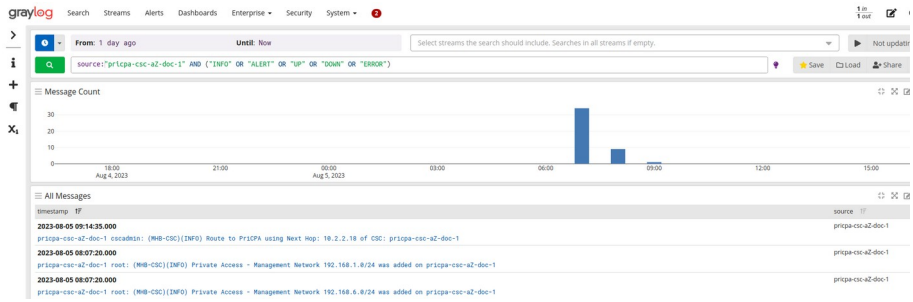
Traffic Logs:

- ALLOW
- BLOCK

11.1.2.6.1 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")



11.1.2.6.2 Traffic Logs example:

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

11.1.2.7 HIGH AVAILABILITY Information

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

```
HIGH AVAILABILITY Information
The HA service is: active (running) since Sat 2023-08-05 09:12:43 UTC; 6h ago
Identity Type: SystemAssigned
Route to PriCPA using Next Hop: 10.2.2.18 of VM: pricpa-csc-aZ-doc-1 (this CSC)
Current values configured are:
   Route/s (Qty)= 2
      Route 1: MHB-DC-172.19.0.0 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
      Route 2: MHB-DC-192.168.0.0 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
   Computer Name of other CSC in the pair: pricpa-csc-aZ-doc-2 (Resource Group=CSC-East-US)
   Private Access Public IP= 20.127.203.54
   KeepAlive Remote IP: 172.19.0.133 is Alive
```

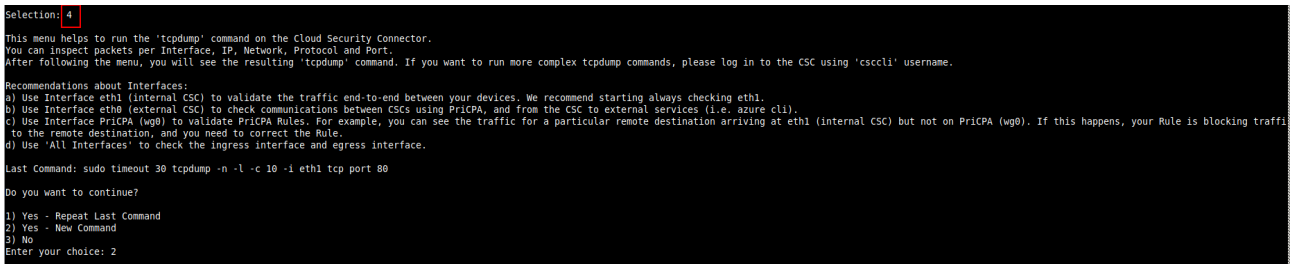
- If HA service is active.
- The Identity Type in use.
- The current “Next Hop” active for all "Route/s" configured.
- Amount of Routes configured.
- The Route names.
- Which is the VM Name of other CSC on the HA pair.
- Private Access Public IP.
- KeepAlive Remote IP status (if configured)

11.1.3 Show Interfaces Traffic

Use this section to see the traffic in real time.

11.1.4 Tcpdump.

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

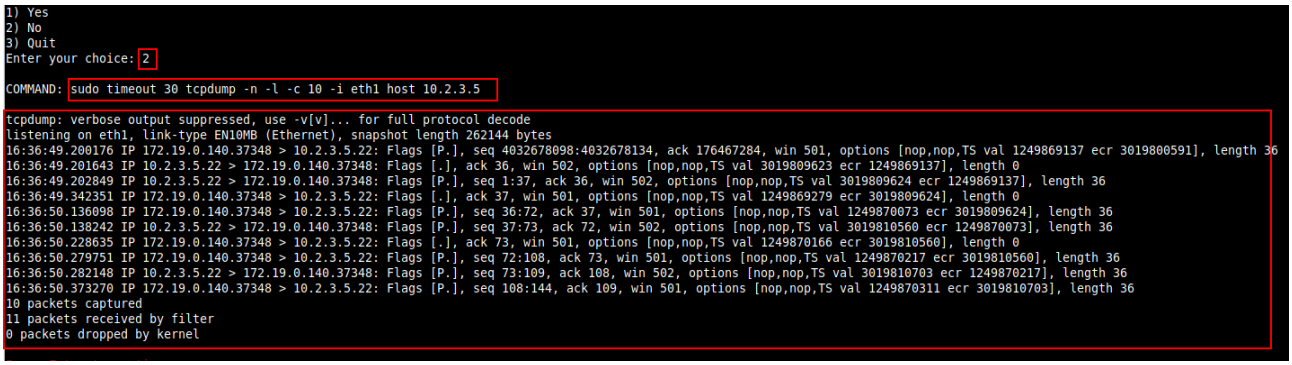


You can repeat the last command or running a new command. Example running a new command:

- 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.



11.2 Configuration Wizards

```
Configuration Wizards
5) Configure PriCPA: Local and Peers Configuration.
6) Configure CSC Remote Management Networks via PriCPA.
7) High Availability configuration.
```

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

11.3 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.
```

11.3.1 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[7] using AWS Systems Manager.

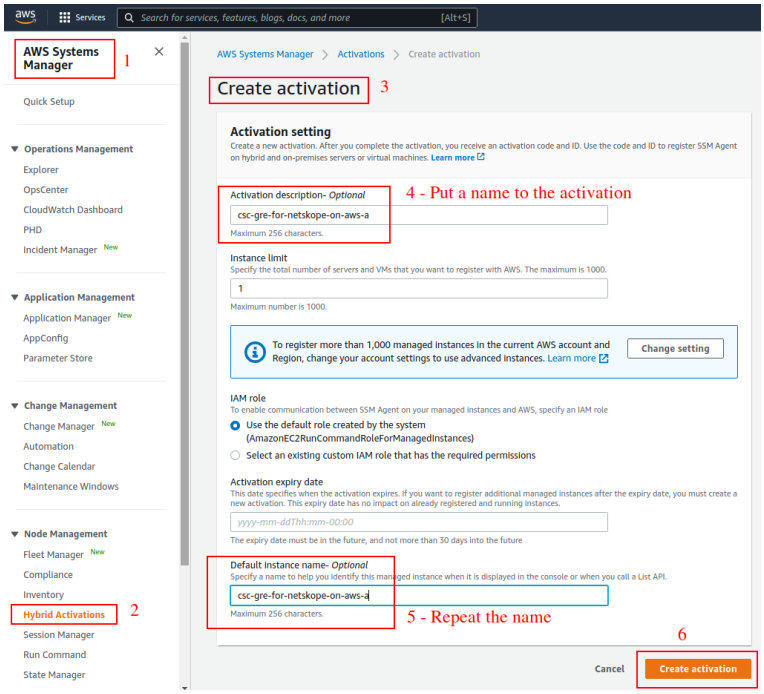
***Note**: You can learn more about “Run Commands” on Appendix B*

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

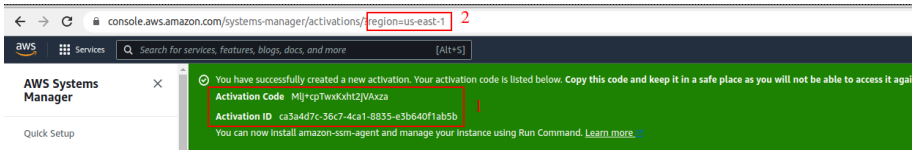
11.3.1.1 Create a "Hybrid Activation" from AWS console.

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

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

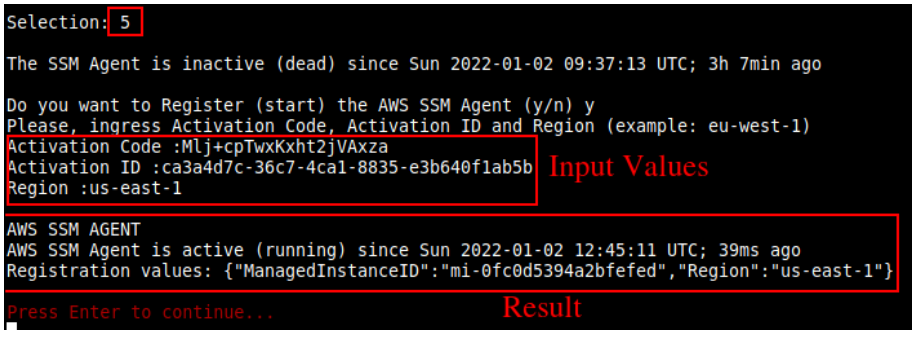


→ Click "Create Activation"

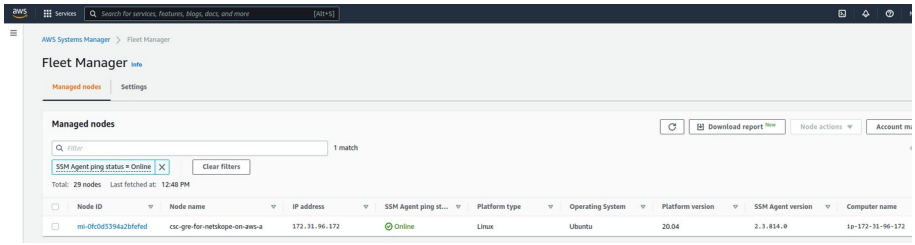


The values of Actication Code, Activation ID and Region are required to register the CSC. Keep this values on a safe place.

11.3.1.2 Register the CSC



11.3.1.3 View the Registered CSC on AWS Systems Manager



11.3.2 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:
```

11.3.2.1 Manage Administrators: cscadmin and csccli

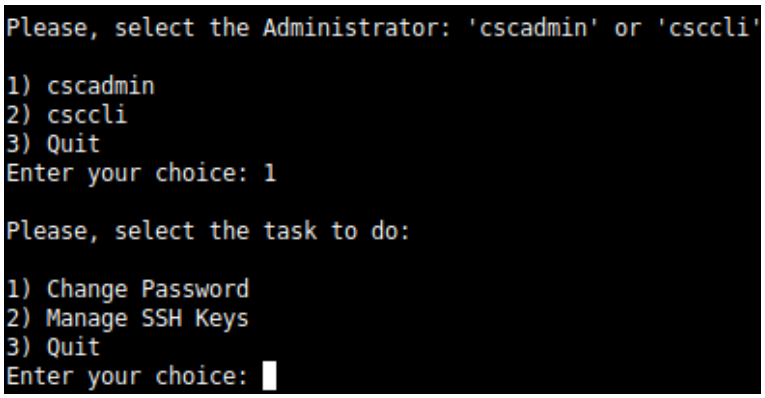
The CSC PriCPA for AZure has 2 users configured: cscadmin (for SSH Administrator Console Access), csccli (standard user, disabled by default.).

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:
```

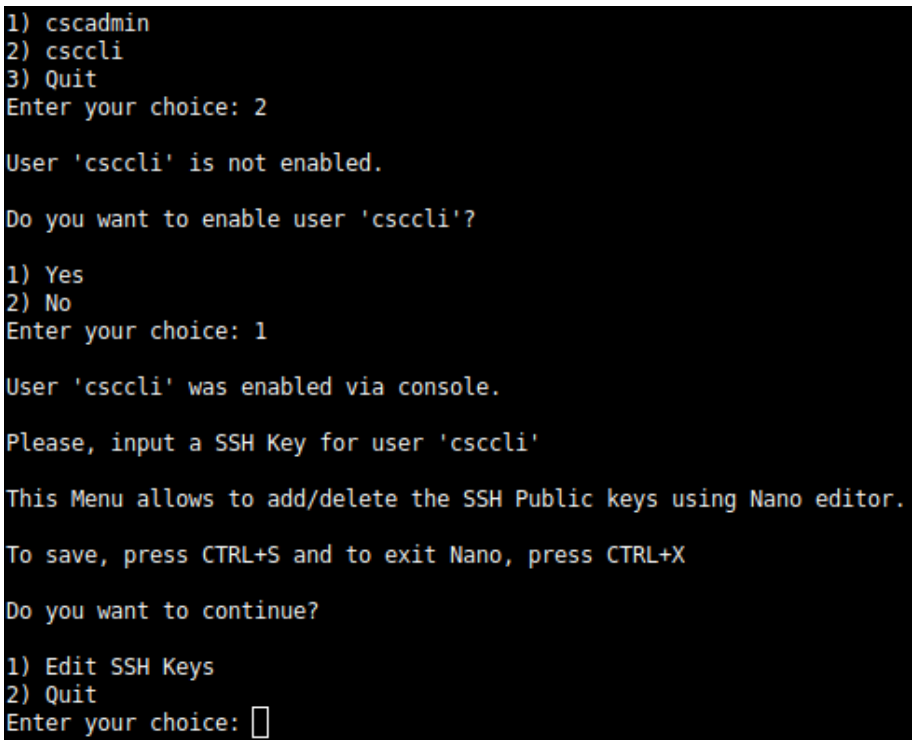
Note: *the user "cscadmin" cannot be disabled.*

11.3.2.1.1 "cscadmin" settings



11.3.2.1.2 "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.



11.3.2.1.3 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.

11.3.2.2 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:

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

This wizard allows restricting the SSH access to the CSC.

IMPORTANT (1): DEFAULT VALUES.
     -> Internal Interface (eth1): SSH the CSC to CSC GW IP (10.2.2.15) is allowed from any Host or Subnet.
     -> External Interface (eth0): SSH the CSC to any eth0 IP is permanently blocked and cannot be changed.
     -> PriCPA Interface (wg0): SSH the CSC to wg0 IP (192.168.7.16) is allowed from any other PriCPA node that belongs to the PriCPA Subnet. (192.168.7.0/24)

WARNING! You can isolate this node if the configuration is wrong.
Be careful with these settings. We recommend being precise with the Host or Subnet configured here.
Subnet Prefixes less than /8 are not accepted.

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

Current values configured are:

SSH to CSC GW IP (10.2.2.15) is allowed only from: 10.2.0.0/16 172.19.0.0/24 192.168.1.0/24 192.168.6.0/24
SSH to PriCPA IP (192.168.7.16) is allowed only from: 10.2.0.0/16 172.19.0.0/24 192.168.1.0/24

Do you want to change values?

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

11.3.2.3 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.

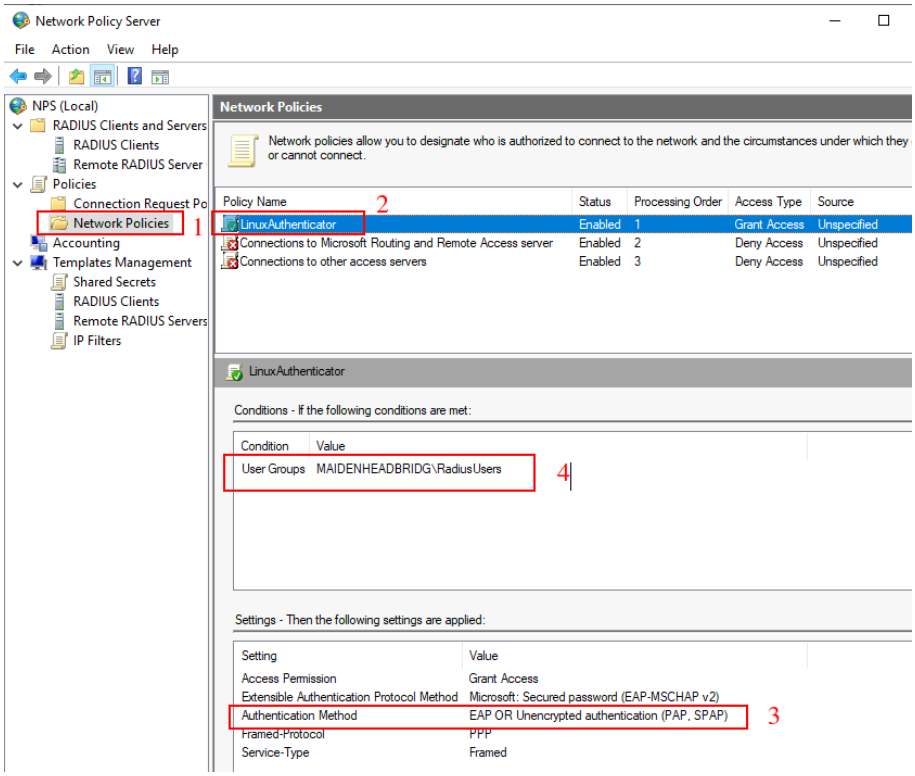
-> Configuration on the CSC: Add Radius Server and User:

```
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

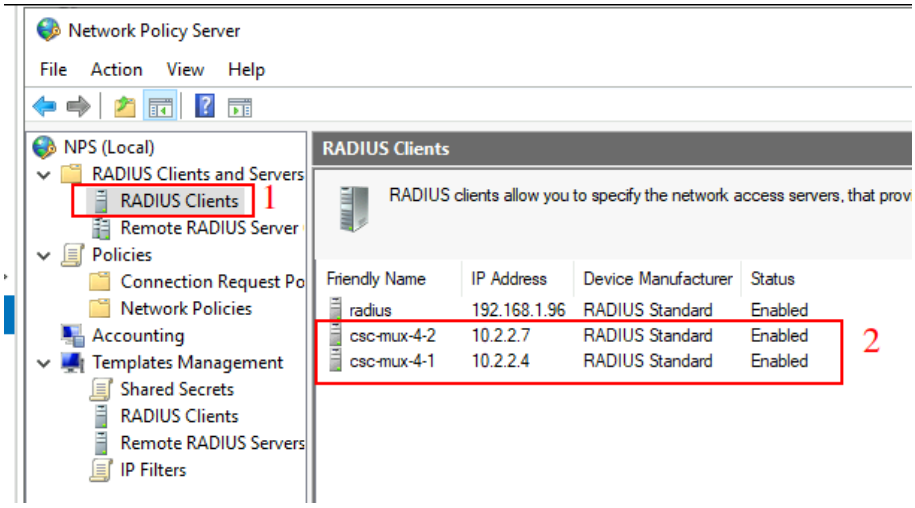
1 - Create Network Policy

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.

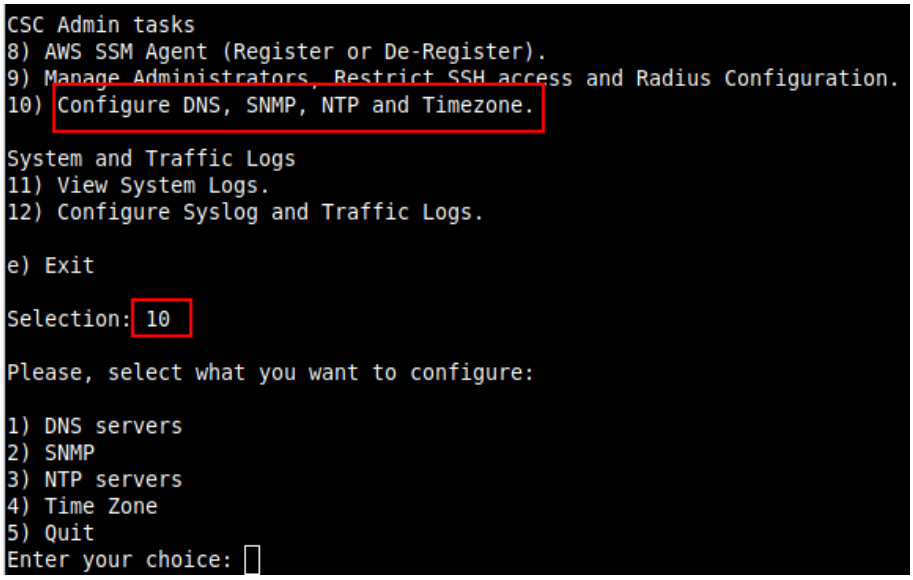


2 - Add the CSC as Radius Clients:

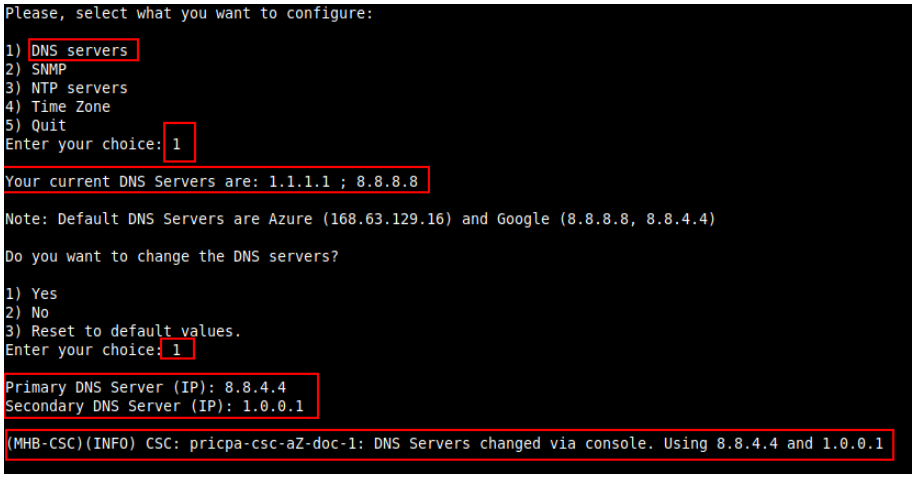
Note: The traffic will arrive to the NPS with source IP: CSC GW IP



11.4 Configure DNS, SNMP, NTP and Timezone.



11.4.1 DNS servers

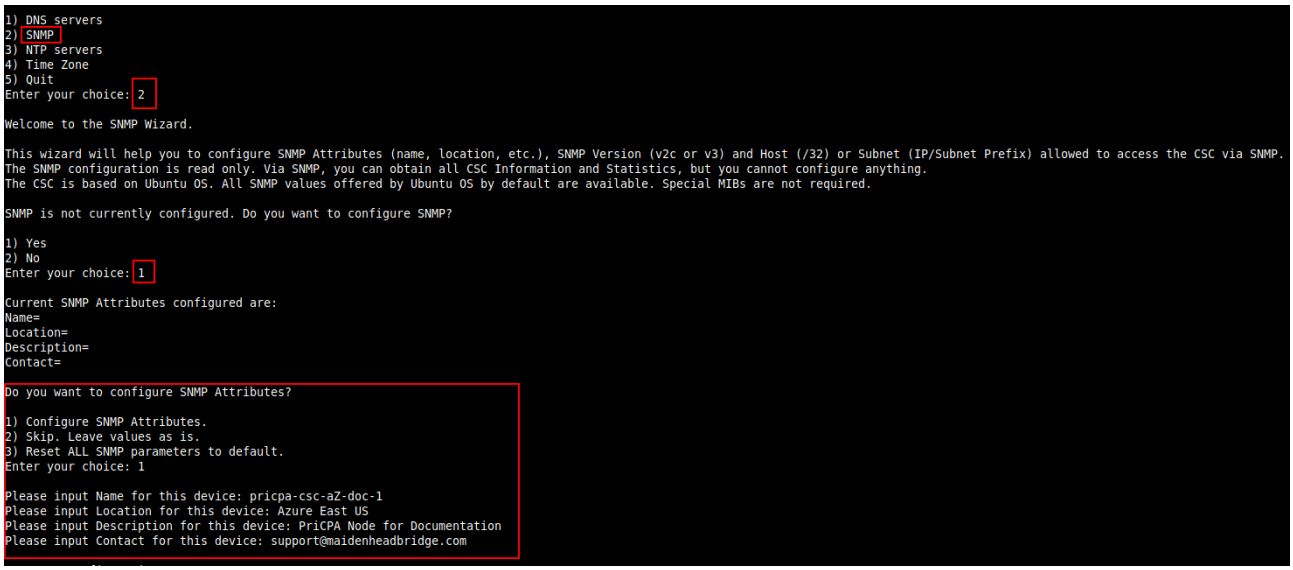


11.4.2 SNMP

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.

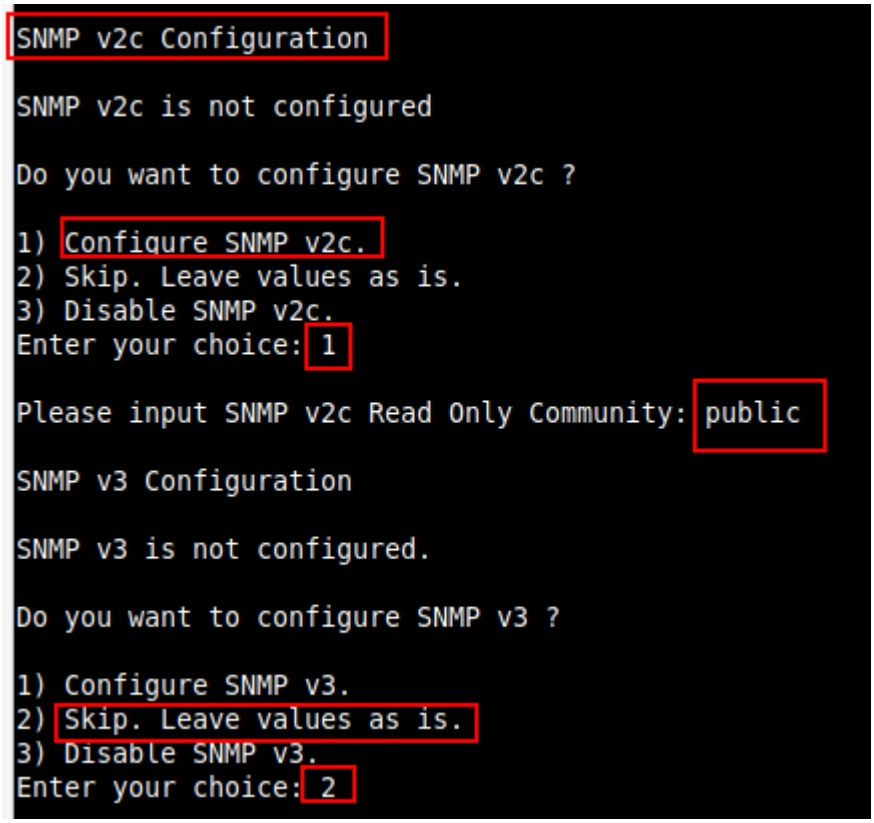
11.4.2.1 Configure SNMP attributes



11.4.2.2 SNMP v2c configuration

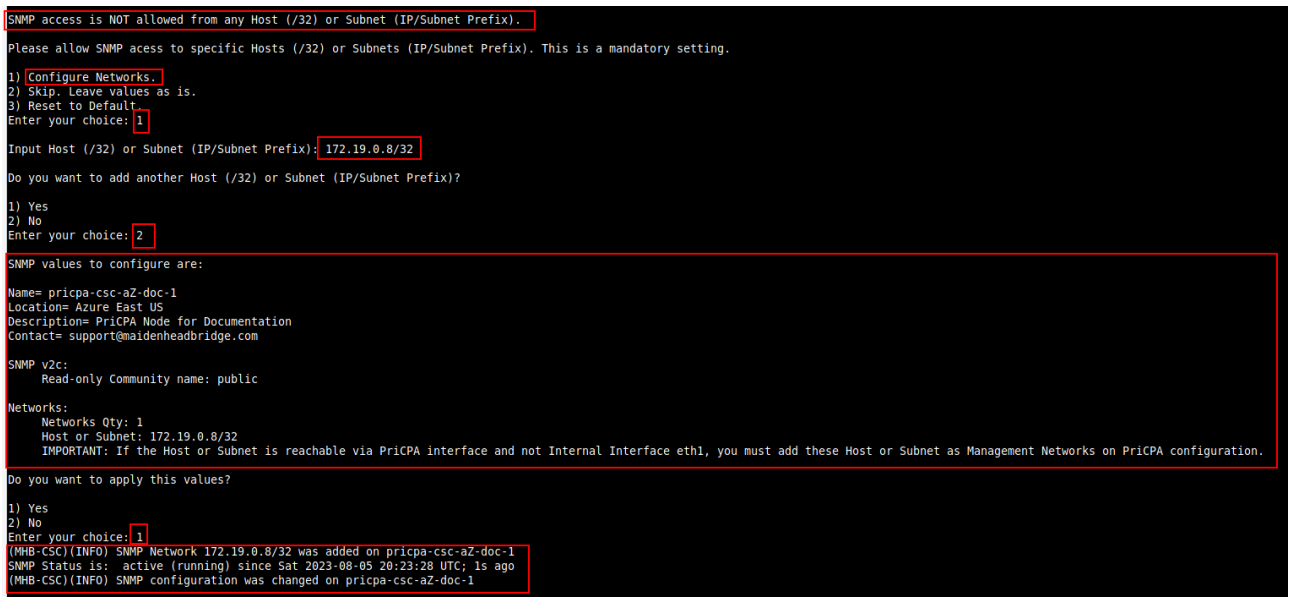
SNMP version 2c requires the "read only community" and the IP or Subnet of the SNMP platform.

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



11.4.2.3 SNMP Networks

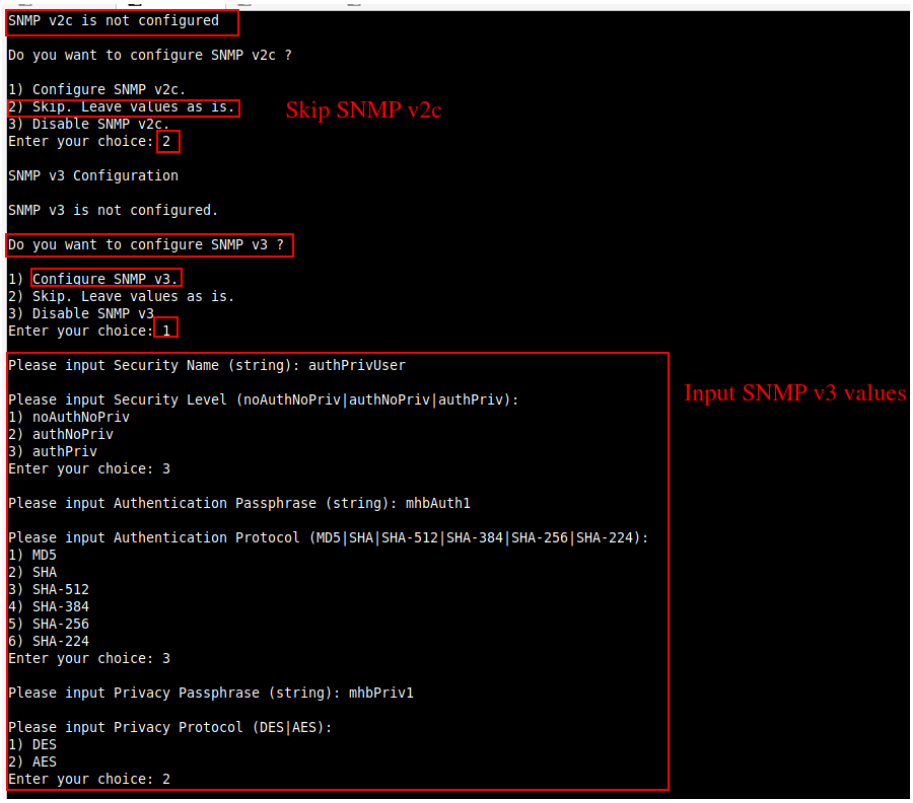
The CSC blocks all SNMP request by default. You need to enable the source IPs (or Subnets) that will query the CSC using SNMP. This setting is mandatory for SNMP v2c and v3.



11.4.2.4 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



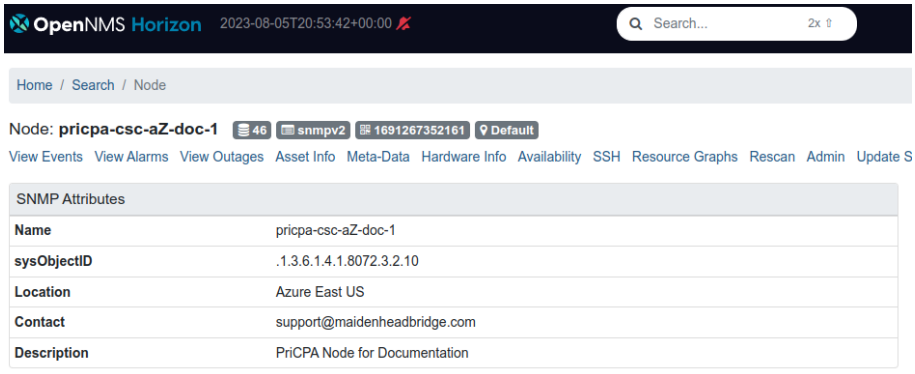
Configure Networks and Confirm values:

```
SNMP access is NOT allowed from any Host (/32) or Subnet (IP/Subnet Prefix).

Please allow SNMP acess to specific Hosts (/32) or Subnets (IP/Subnet Prefix). This is a mandatory setting.

1) Configure Networks.
2) Skip. Leave values as is.
3) Reset to Default.
Enter your choice: 1

Input Host (/32) or Subnet (IP/Subnet Prefix): 172.19.0.8/32

Do you want to add another Host (/32) or Subnet (IP/Subnet Prefix)?

1) Yes
2) No
Enter your choice: 2

SNMP values to configure are:

Name= pricpa-csc-aZ-doc-2
Location= Azure East US
Description= PriCPA Node for Documentation - 2
Contact= support@maidenheadbridge.com

SNMP v3:
     SecurityName= authPrivUser
     SecurityLevel= authPriv
     AuthPassphrase= mhbAuth1
     AuthProtocol= SHA-512
     PrivacyPassphrase= mhbPriv1
     PrivacyProtocol= AES

Networks:
     Networks Qty: 1
     Host or Subnet: 172.19.0.8/32
     IMPORTANT: If the Host or Subnet is reachable via PriCPA interface and not Internal Interface eth1, you must add these Host or Subnet as Management Networks on PriCPA configuration.

Do you want to apply this values?

1) Yes
2) No
Enter your choice: 1
(MHB-CSC)(INFO) SNMP Network 172.19.0.8/32 was added on pricpa-csc-aZ-doc-2
SNMP Status is:  active (running) since Sat 2023-08-05 20:40:18 UTC; 1s ago
(MHB-CSC)(INFO) SNMP configuration was changed on pricpa-csc-aZ-doc-2
```

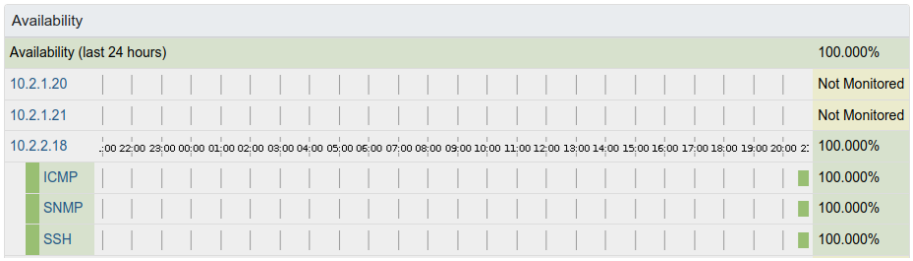
11.4.2.5 What can you do with SNMP?

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

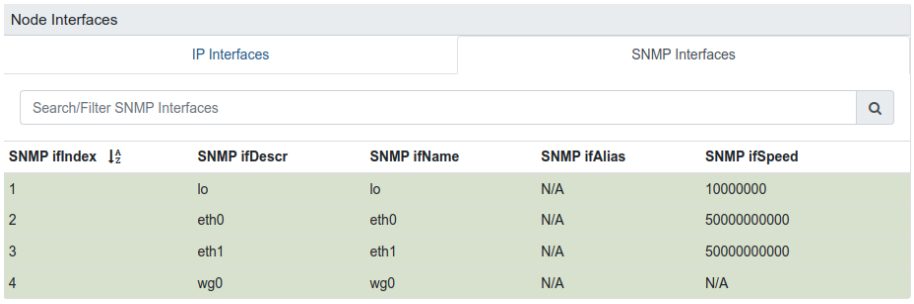
11.4.2.5.1 Node Information



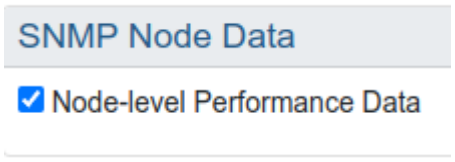
11.4.2.5.2 Node Availability

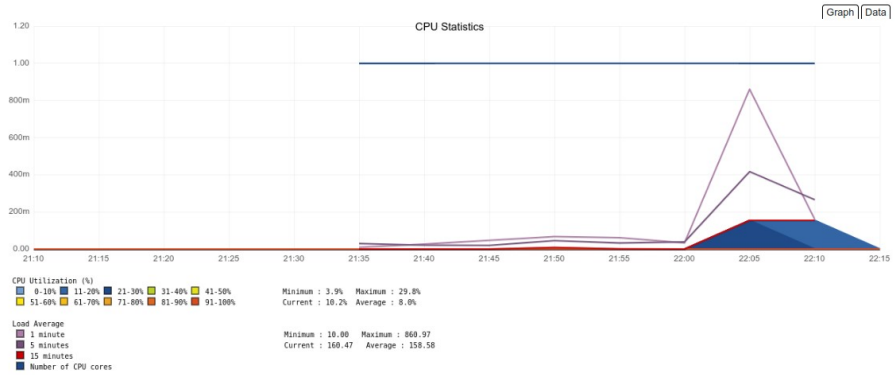


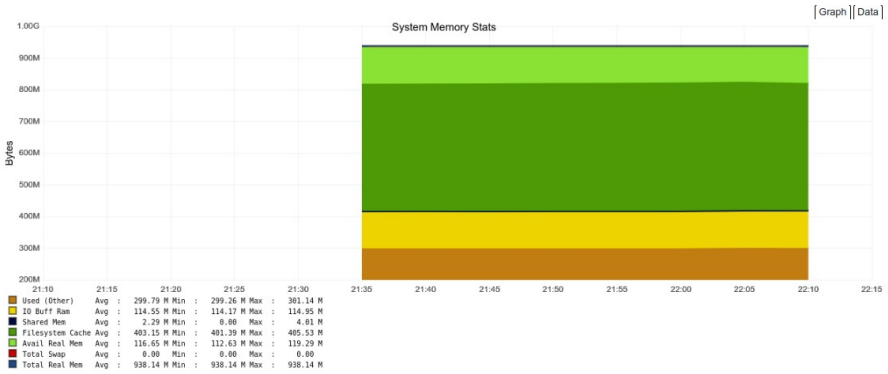
11.4.2.5.3 Node Interfaces (IP & SNMP)



11.4.2.5.4 Node Statistics (CPU, Memory, etc)

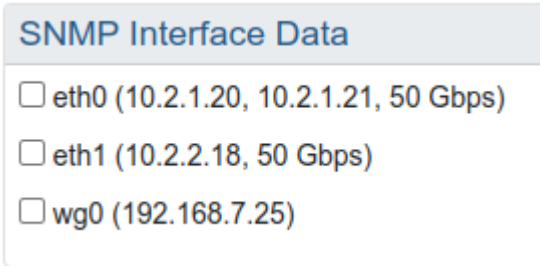




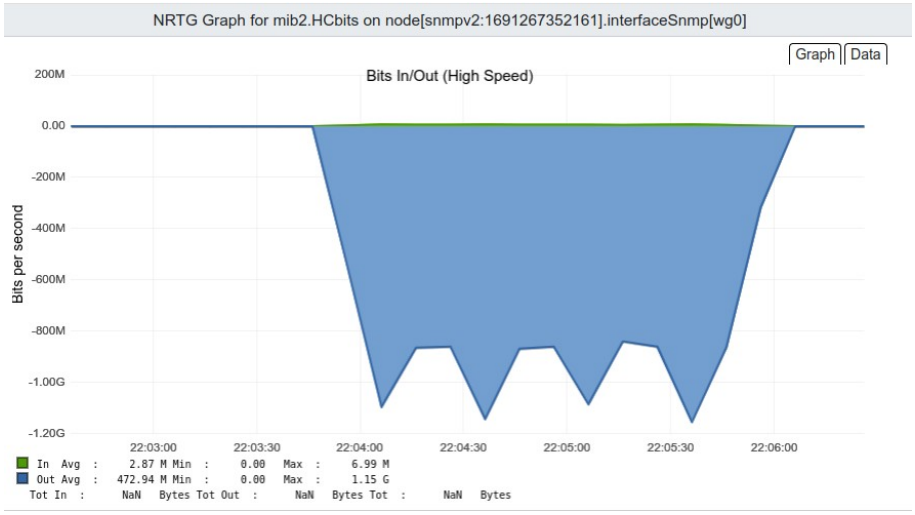


11.4.2.5.5 Interfaces Traffic

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



Example of real time traffic on PriCPA interface:



11.4.3 NTP Servers

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

```
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
```

NTP Servers.

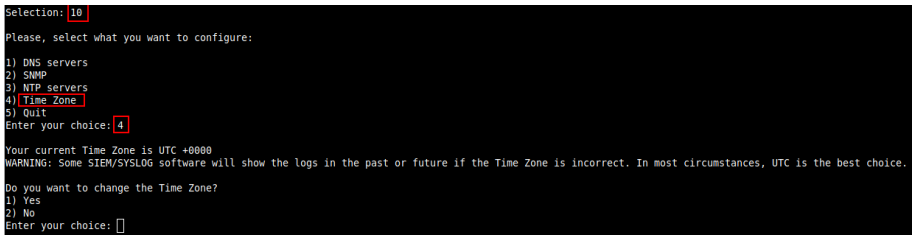
Check the Status:

```
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)."
```

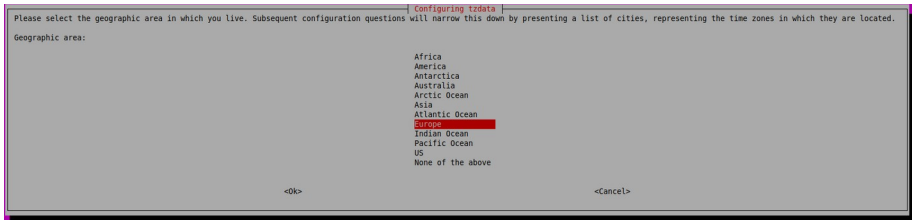
The NTP Server connects correctly when the Status is: "Initial synchronization to time server xxxx".

11.4.4 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.



11.5 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.
```

11.5.1 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.
```

11.5.2 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
```

12 Remote Management

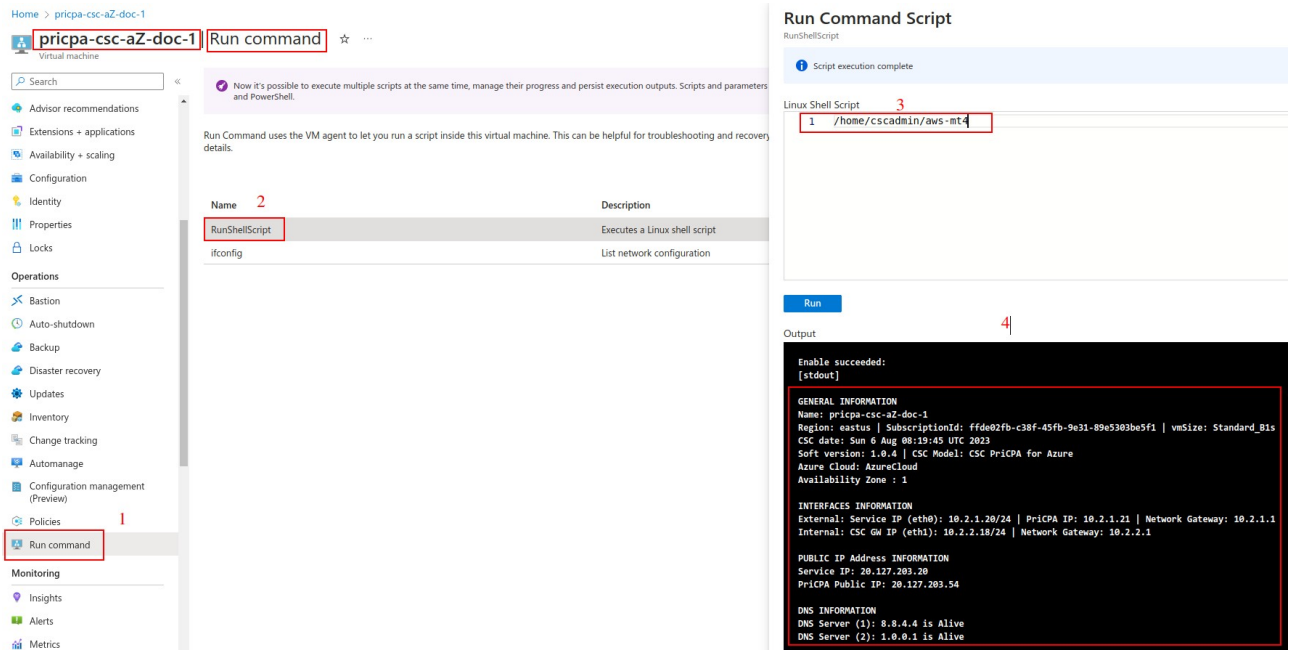
You can use several tools to Remote Manage the CSC. In this chapter, we are showing how to use Azure "Run Command", AWS Systems Manager (Fleet Manager) and Rundeck.

12.1 Azure "Run Command"

12.1.1 Using Azure Portal

Azure portal allows to "Run Command" per VM. "Run Command" is particularly useful if you want to do a quick check, not SSH the CSC.

Instructions: Select the VM go to Run Command → RunShellScript and on “Linux Shell Script” put the command showed in the below table.



12.1.2 Using Azure CLI

The command to execute is the following:

Linux:

```
$az vm run-command invoke -g <ResourceGroup> -n <VmName> --
command-id RunShellScript --scripts "<CSC Command>" | jq -
r .value[].message
```

Please, note that we are using the program "jq" to extract "message" information and to present it to the Linux terminal.

Example 1: MHB-CSC-ShowConfigurationAndStatus

```
$ az vm run-command invoke -g CSC-East-US -n pricpa-csc-aZ-doc-1 --command-id RunShellScript --scripts /home/cscadmin/aws-mt4 | jq -r .value[].message
Enable succeeded:
[stdout]

GENERAL INFORMATION
Name: pricpa-csc-aZ-doc-1
Region: eastus | SubscriptionId: ffde02fb-c38f-45fb-9e31-89e5303be5f1 | vmSize: Standard_B1s
CSC date: Sun 6 Aug 08:25:25 UTC 2023
Soft version: 1.0.4 | CSC Model: CSC PriCPA for Azure
Azure Cloud: AzureCloud
Availability Zone : 1

INTERFACES INFORMATION
External: Service IP (eth0): 10.2.1.20/24 | PriCPA IP: 10.2.1.21 | Network Gateway: 10.2.1.1
Internal: CSC GW IP (eth1): 10.2.2.18/24 | Network Gateway: 10.2.2.1

PUBLIC IP Address INFORMATION
Service IP: 20.127.203.20
PriCPA Public IP: 20.127.203.54

DNS INFORMATION
DNS Server (1): 8.8.4.4 is Alive
DNS Server (2): 1.0.0.1 is Alive

AWS SSM AGENT
AWS SSM Agent is active (running) since Sat 2023-08-05 22:00:56 UTC; 10h ago
Registration values: {"ManagedInstanceID":"mi-02f61b13176c35082","Region":"eu-west-2"}

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

HIGH AVAILABILITY Information
The HA service is:  active (running) since Sat 2023-08-05 22:03:01 UTC; 10h ago
Identity Type: SystemAssigned
Route to PriCPA using Next Hop: 10.2.2.18 of VM: pricpa-csc-aZ-doc-1 (this CSC)
Current values configured are:
   Route/s (Qty)= 2
      Route 1: MHB-DC-172.19.0.0 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
      Route 2: MHB-DC-192.168.0.0 (Route Table=Servers-Route-Table, Resource Group=RouteTables-East-US)
   Computer Name of other CSC in the pair: pricpa-csc-aZ-doc-2 (Resource Group=CSC-East-US)
   Private Access Public IP= 20.127.203.54
   KeepAlive Remote IP: 172.19.0.133 is Alive
```

Example 2: MHB-CSC-Show-Private-Access-ALL-Peers-Status

```
$ az vm run-command invoke -g CSC-East-US -n pricpa-csc-aZ-doc-1 --command-id RunShellScript --scripts "/home/cscadmin/aws-show-private-access-all-peers-status" | jq -r .value[].message
Enable succeeded:
[stdout]

Peer 'ns-csc-mux-4-as' (4.246.221.166:51820) -> 192.168.7.15 is Alive. Source Port OK. Using '51820'
Peer 'pricpa-gcloud-v-0-2-a' (35.246.67.148:51820) -> 192.168.7.102 is Alive. Source Port OK. Using '51820'
Peer 'ns-csc-gre-v-1-0e' (18.213.109.84:51820) -> 192.168.7.37 is Alive. Source Port OK. Using '51820'
Peer 'ns-csc-gre-aws-v-0-4' (52.4.62.40:51820) -> 192.168.7.88 is Alive. Source Port OK. Using '51820'
Peer 'ns-cgc00004' (82.68.6.74:51821) -> 192.168.7.11 is Alive. Source Port was changed. Port configured is '51821' and is using '12844'. Please review NAT rules on this node or, as the last resource, enable Persistent Keepalive on this
ode.
Peer 'ns-cgc00008' (92.40.213.105:51820) -> 192.168.7.8 is not reachable. Source Port OK. Using '51820'
Peer 'ns-cgc00006' (217.155.196.81:51820) -> 192.168.7.20 is Alive. Source Port OK. Using '51820'

[stderr]
```

12.1.3 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

7	MHB-CSC-Reload-High-Availability	/home/cscadmin/aws-reload-high-availability-json

12.2 AWS Systems Manager

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

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

12.2.1 Create Documents

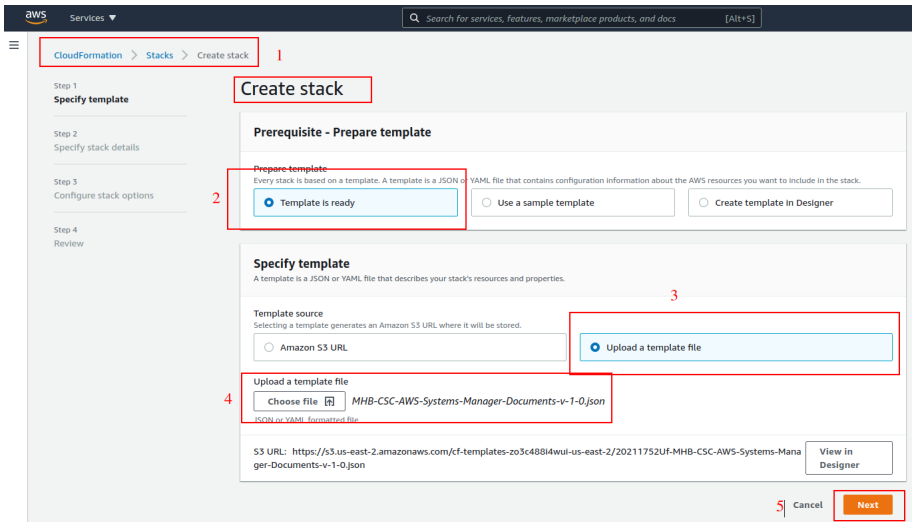
We provide a CloudFormation template to create all "Documents" in one shot.

Steps:

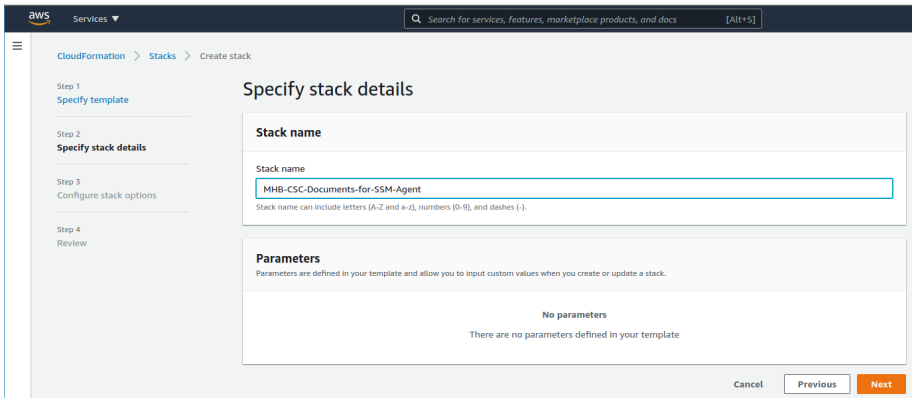
1. Download the CloudFormation template from:

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

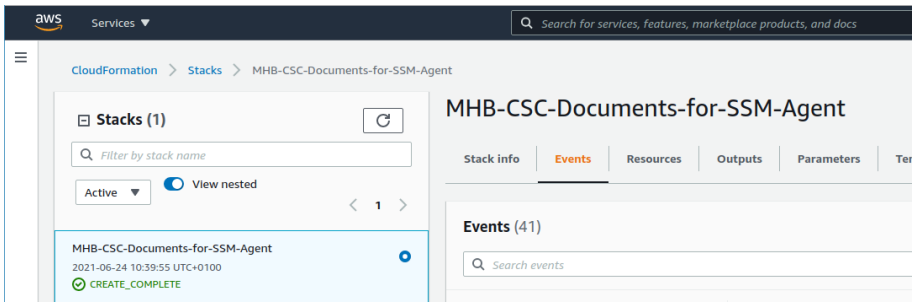
2. Deploy Stack. Go to Cloudformation →Create Stack→ Upload a template file



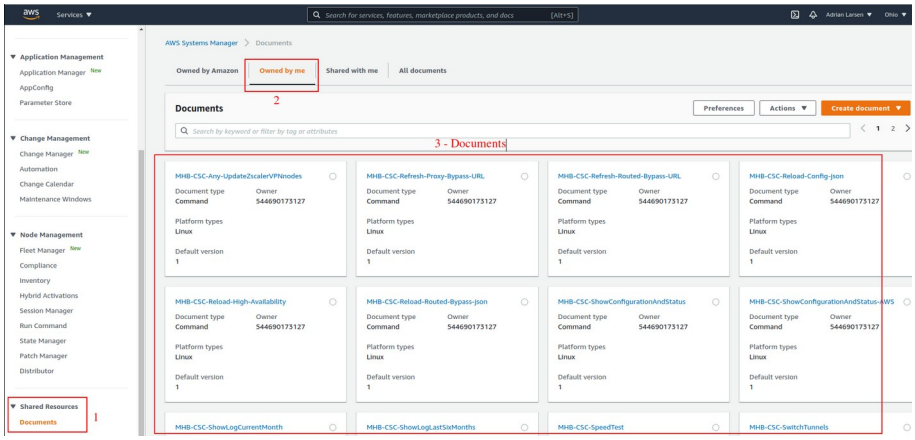
3. Click next.
4. Put the Stack Name



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!

12.2.2 Run Commands

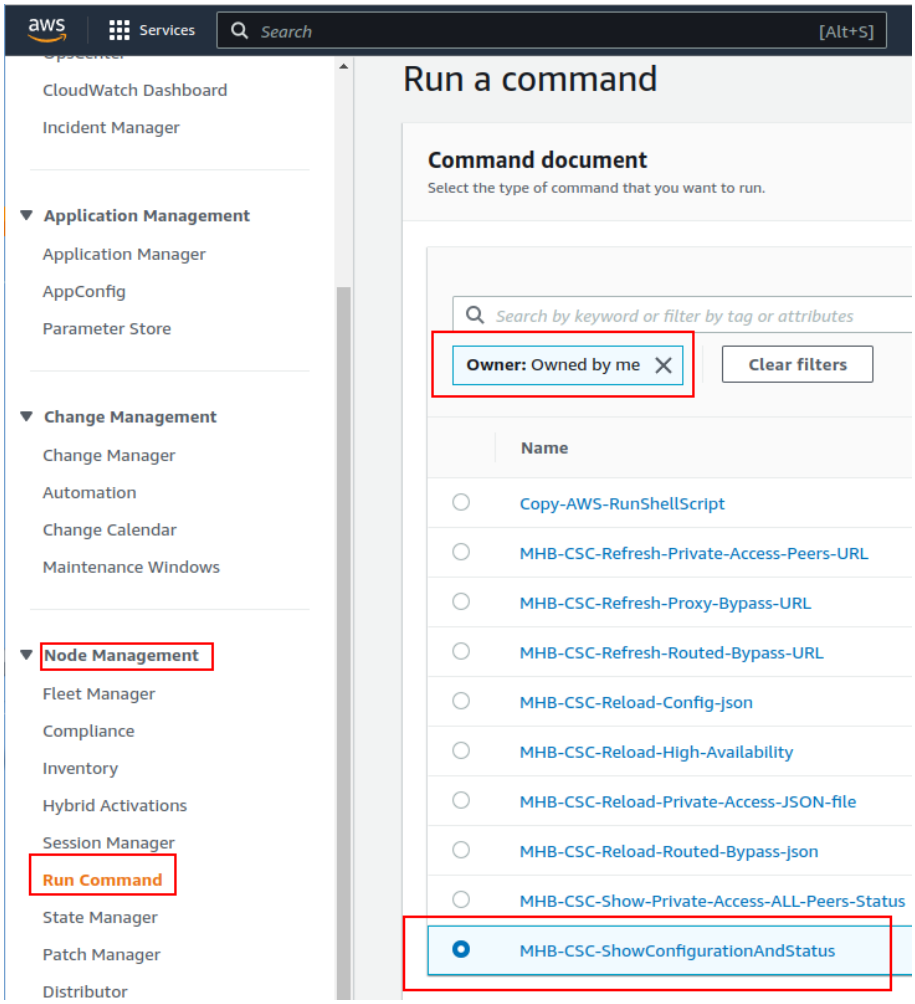
After you have created the Documents, you are ready to Run Commands on the CSC.

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

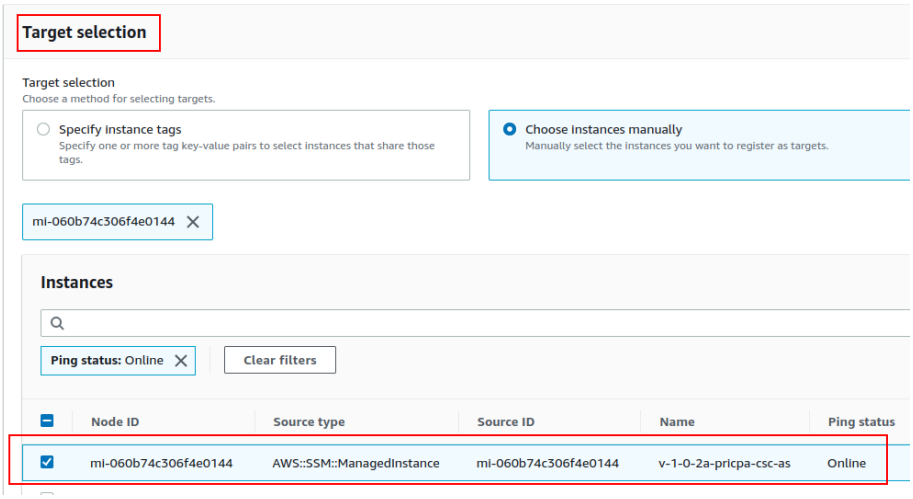
To "Run Commands", go to AWS Systems Manager → Instances & Nodes → Run Command.

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

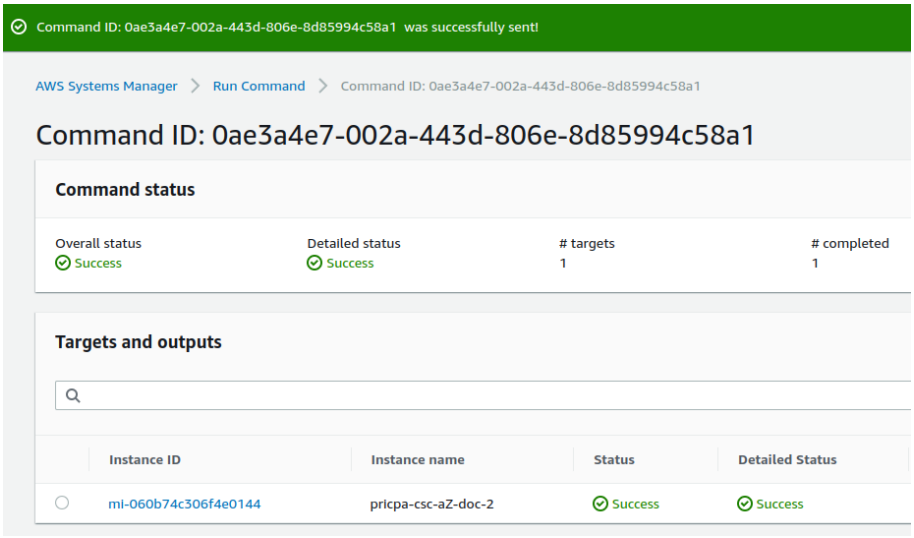
1. Run a Command
2. Select the Document created (Tip: Select "Owned by me")



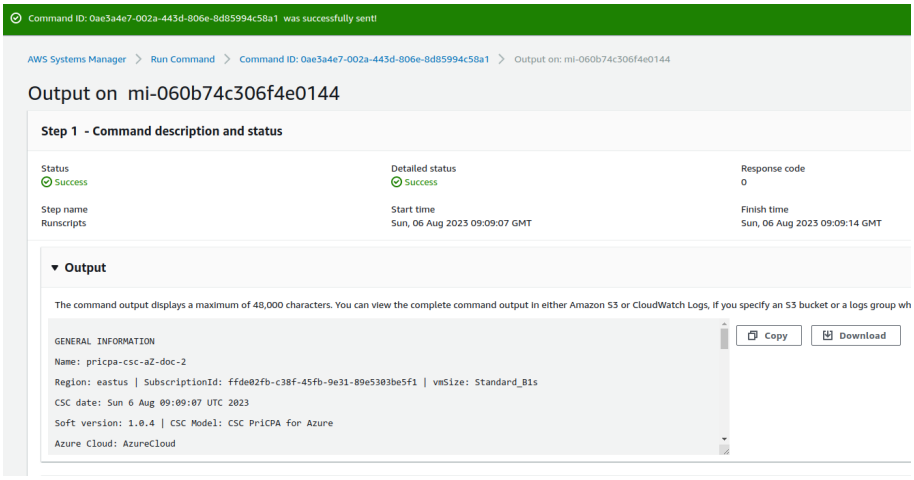
3. Scroll down and Select the Instances



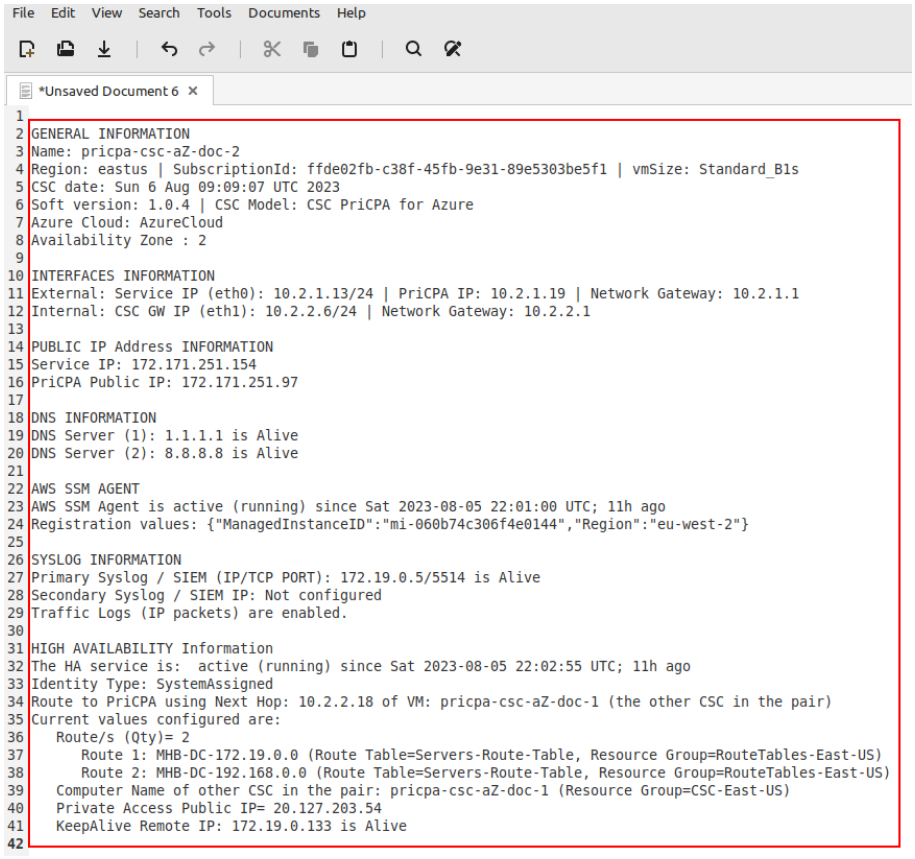
4. Click "Run" . Wait for the Command Status "success"



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)



12.2.3 List of Documents available for "Run Command"

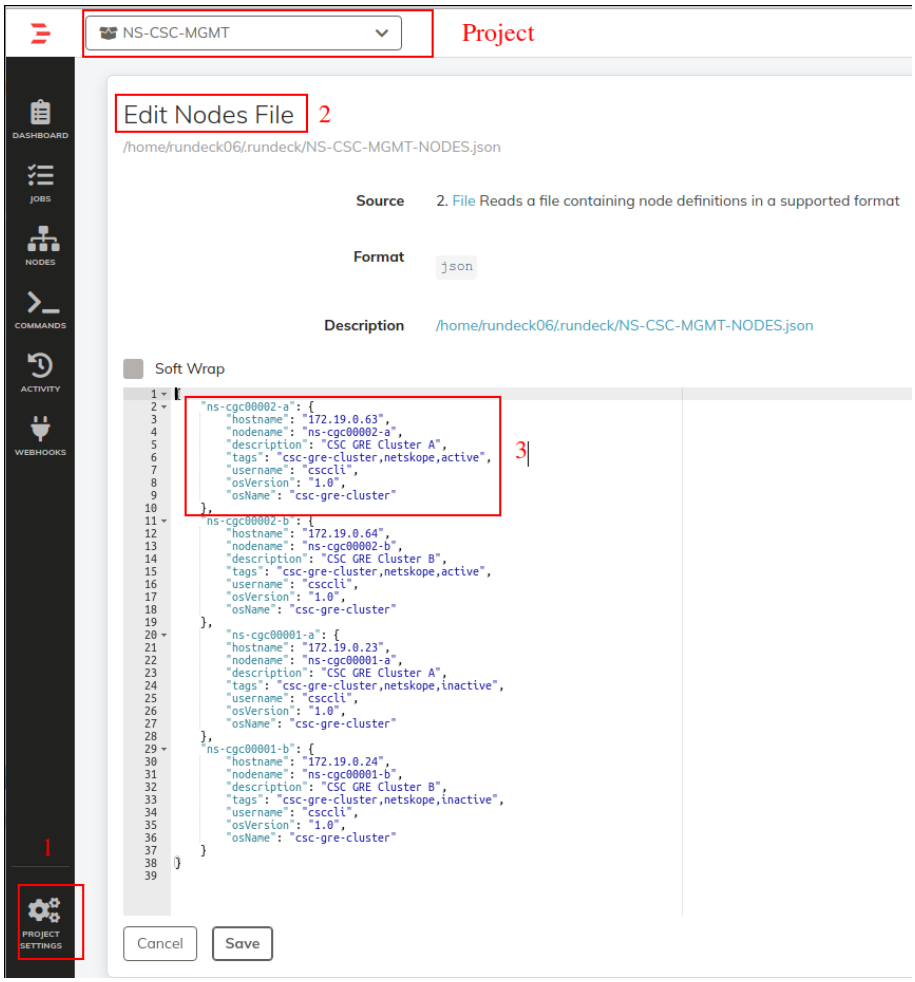
1. "MHB-CSC-ShowConfigurationAndStatus": Executes "Show Configuration and Status"
2. "MHB - CSC - Show Private Access ALL Peers status": Show the Status of all Private Access Peers.
3. "MHB - CSC - Refresh Private Access Peers URL": Refresh the Private Access Peers list using the values of the JSON file 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.
7. "MHB-CSC-Reload-High-Availability": Reloads the values of highAvailability.json file. (for CSC on AWS, Azure and Gcloud. Not in use on CSC for Virtual Platforms.

12.3 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:

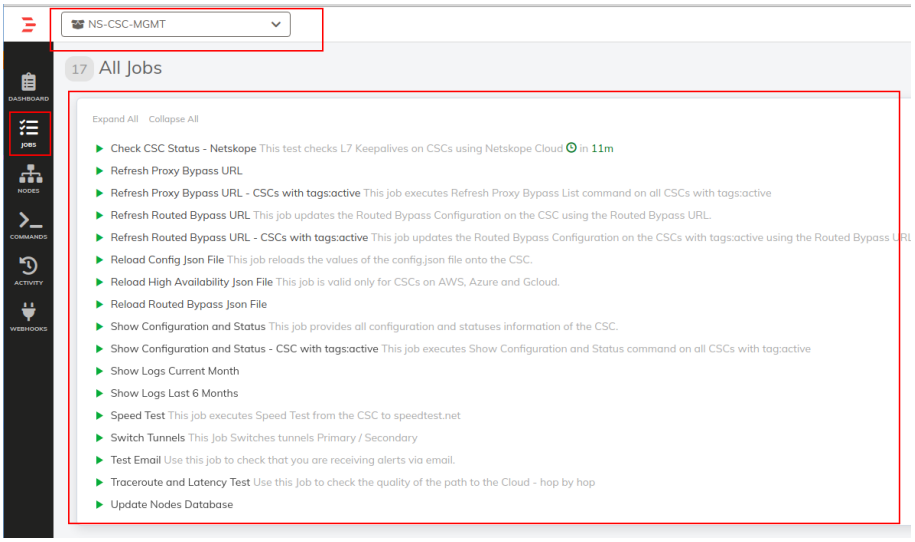
1. Install Rundeck. Intructions at: https://www.rundeck.com/open-source
2. Create a Project.
3. Enable user "csccli" and setup the SSH Public key on each CSC.
4. On the Project, setup the SSH Private and define the nodes:



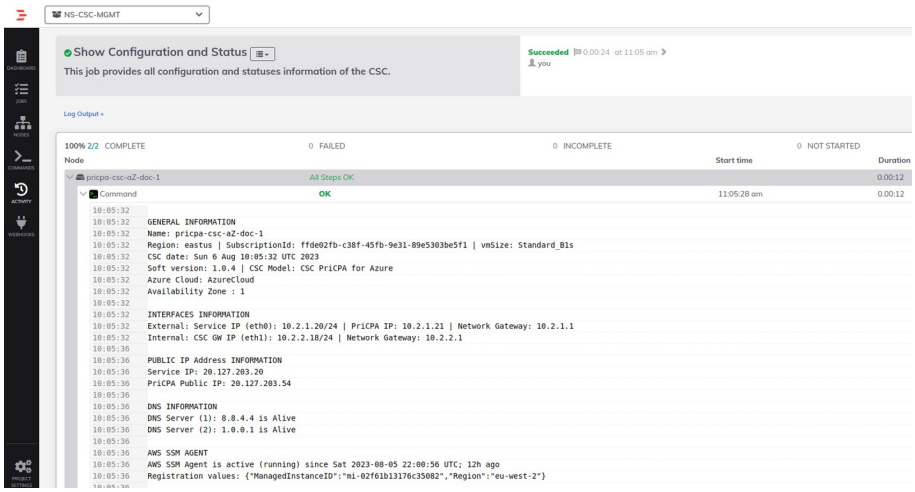
5. Create the jobs. Please, contact Support at http://support.maidenheadbridge.com for the latest Job List.

12.3.1 Jobs

The following screen shows the list of Jobs available.



12.3.2 Running job "Show Configuration and Status"



13 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.
2. **highAvailability.json**: Allows administrators to configure the CSC on HA pair.

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

13.1 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": "TrMvSoP4jYQlY6RlzBgbssQqY3vxI2Pi+y71lOWWXX0=",
      "publicIpAndUdpPort": "200.1.1.3:51821",
      "privateCirdIp": "192.168.7.3/24",
      "persistentKeepAlive": "no",
      "networks": [ "10.3.1.0/24", "10.3.2.0/24" ],
      "privateApps": [
        {
          "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.

13.2 highAvailability.json file

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

Steps:

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

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

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

```
{
   "model": "csc-pricpa-azure",
   "type": "highAvailability",
   "version": "1.0.1",
   "highAvailability": {
      "haEnable": false,
      "halamRole": "",
      "haFirstCsc": {
         "vmName": "",
         "vmResourceGroup": "",
         "haPublicIp": ""
      },
      "haSecondCsc": {
         "vmName": "",
         "vmResourceGroup": "",
         "haPublicIp": ""
      },
      "haPrivateAccessPublicIp": "",
      "haRoutes": [],
      "haKeepAliveRemoteIp": ""
   }
}
```

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

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

```
{
 "model": "csc-pricpa-azure",
 "type": "highAvailability",
 "version": "1.0.1",
 "highAvailability": {
  "haEnable": true,
  "haIamRole": "SystemAssigned",
  "haFirstCsc": {
   "vmName": "pricpa-csc-aZ-doc-1",
   "vmResourceGroup": "CSC-East-US",
   "haPublicIp": "20.127.203.54"
  },
  "haSecondCsc": {
   "vmName": "pricpa-csc-aZ-doc-2",
   "vmResourceGroup": "CSC-East-US",
   "haPublicIp": "172.171.251.97"
  },
  "haPrivateAccessPublicIp": "20.127.203.54",
  "haRoutes": [
   {
    "routeName": "MHB-DC-172.19.0.0",
    "routeTable": "Servers-Route-Table",
    "resourceGroup": "RouteTables-East-US"
   },
   {
    "routeName": "MHB-DC-192.168.0.0",
    "routeTable": "Servers-Route-Table",
    "resourceGroup": "RouteTables-East-US"
   }
  ],
  "haKeepAliveRemoteIp": "172.19.0.133"
 }
}
```

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

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

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

14 Appendixes

14.1 Appendix A: Release Notes

14.1.1 Version 1.0.5 (August 2023)

The version 1.0.5 has the following enhancements.

- The CSC nows checks Internet using: “http://www.msftconnecttest.com/connecttest.txt” instead “https://management.azure.com/healthcheck?api-version=2014-04-01”. The change was done because the later destination URL counts against the Azure limits and can be rate limited. Also, because it is slow to respond.

14.1.2 Version 1.0.4 (August 2023)

This is the initial public release of the CSC PriCPA for Azure.

14.2 Appendix B: configUserData.json file

14.2.1 Parameters

Via configUserData.json file, you can pass values to parameters during the installation of the CSC. You can setup:

1. The AWS SSM agent registration values.
2. DNS servers
3. Syslog servers and traffic log configuration.
4. PriCPA Local configuration values, Peers URL and Remote Management Networks.
5. SSH Restrictions via eth1 and wg0.
6. Admin Management: Enable csccli user and SSH Key.

14.2.2 configUserData.json file (blank)

configUserData.json (blank)

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

configUserData.json

```
{
   "model": "csc-pricpa-azure",
   "type": "configUserData",
   "version": "1.0.1",
   "awsSsmAgent": {
      "enable": "no",
      "activationCode": "",
      "activationId": "",
      "awsRegion": ""
   },
   "dns": {
      "useCloudDns": "yes",
      "primaryDnsIp": "",
      "secondaryDnsIp": ""
   },
   "syslog": {
      "enable": "no",
      "primaryServer": {
         "ip": "",
         "port": ""
      },
      "secondaryServer": {
         "ip": "",
         "port": ""
      },
      "trafficLogs": {
         "enable": "no"
      }
   },
   "priCPA": {
      "enable": "no",
      "nodeName": "",
      "location": "",
      "description": "",
      "publicUdpPort": "51820",
      "privateCirdIp": "",
      "persistentKeepAlive": "no",
      "peersJsonFileUrl": "",
      "remoteManagementNetworks": []
   },
   "sshRestrictions": {
      "eth1": {
         "enable": "no",
         "allowedNetworks": []
```

```
      },
      "wg0": {
         "enable": "no",
         "allowedNetworks": []
      }
   },
   "adminManagement": {
      "csccli": {
         "enable": "no",
         "sshPublicKey": ""
      }
   }
}
```

14.2.3 configUserData.json file: Example

configUserData.json


```
{
   "model": "csc-pricpa-azure",
   "type": "configUserData",
   "version": "1.0.1",
   "awsSsmAgent": {
      "enable": "yes",
      "activationCode": "KSrTW3e/GpWRG6nTfezp",
      "activationId": "55e60844-3c5b-4ac5-b8c6-c887adc208da",
      "awsRegion": "eu-west-2"
   },
   "dns": {
      "useCloudDns": "no",
      "primaryDnsIp": "1.1.1.1",
      "secondaryDnsIp": "8.8.8.8"
   },
   "syslog": {
      "enable": "yes",
      "primaryServer": {
         "ip": "172.19.0.5",
         "port": "5514"
      },
      "secondaryServer": {
         "ip": "",
         "port": ""
      },
      "trafficLogs": {
         "enable": "yes"
      }
   },
   "priCPA": {
      "enable": "yes",
      "nodeName": "pricpa-csc-aZ-doc",
      "location": "Azure East US",
      "description": "PriCPA Node for Documentation",
      "publicUdpPort": "51820",
      "privateCirdIp": "192.168.7.25/24",
      "persistentKeepAlive": "no",
      "peersJsonFileUrl": "https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
      "remoteManagementNetworks": [
         "172.19.0.0/24",
         "192.168.1.0/24",
         "192.168.6.0/24"
      ]
   },
   "sshRestrictions": {
      "eth1": {
         "enable": "yes",
         "allowedNetworks": [
            "172.19.0.0/24",
            "192.168.1.0/24",
            "192.168.6.0/24",
            "10.0.0.0/8"
         ]
      },
      "wg0": {
         "enable": "yes",
         "allowedNetworks": [
            "172.19.0.0/24",
            "192.168.1.0/24",
            "192.168.6.0/24",
            "10.0.0.0/8"
         ]
      }
   },
   "adminManagement": {
      "csccli": {
         "enable": "yes",
         "sshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDPm+99wX1/ZhtDIKWh+Uv4TrEYboLoLJlRV6NZctrrkbpq/
WuSCtY9ghLnJs4jmSSaNzSSCZ5ywpa3LxmYZ60huoUvYEXBR+lj7MX+trVsifYUe6aJgjPzH8q3x2X72bS20jBQovrNoeN6DZRWWLzLZ4xyczOF+samsm6l/
O3jop68KG6+FyLcxFM4DddlJrw29sMi9BJmOzA0Ejl2r3x/
Niz+PWqgbvg5Aqm69+uPbJsd6t5egsBsXsKi62blv2rX5hMyZQbxpjcq7BUAc4QcxwZH76X2Y3QVkGnjKXjkFJMcLd6qzN6Su3yYqn41H8ffN2C0rSKD38fvwNDDJTk
mZ93PW9mWQweNuWMvLxQTG14z0qR9VDnIkMtxiJCZyJpUYe6RQDga0nweFIOGBO7N9fA/KzA8r/GjI52E5KlEQQ725pQXcjHZHFjzTeiD1ZjjEBAURUx0DLldBTsO/
oisGUT+pZNQAx6gmX/YxDE/le7qdjuf48aHjwH+uGL1/Q0= cscadmin"
      }
   }
}
```


14.2.3.1 awsSsmAgent


```
  "awsSsmAgent": {
     "enable": "yes",
```

```
    "activationCode": "KSrTW3e/GpWRG6nTfezp",
    "activationId": "55e60844-3c5b-4ac5-b8c6-c887adc208da",
    "awsRegion": "eu-west-2"
},
```

Insert here the values for the activation of the AWS SSM agent. See section 11.3.1 for more details.

14.2.3.2 dns

```
"dns": {
    "useCloudDns": "no",
    "primaryDnsIp": "1.1.1.1",
    "secondaryDnsIp": "8.8.8.8"
},
```

Select "UserCloudDns": "yes" if you want to use Azure DNS (primary) and Google DNS (secondary)

Select "UserCloudDns": "no" and put the values of "primaryDnsIp" and "secondaryDnsIp" if you want to use your own DNS servers.

14.2.3.3 syslog

```
"syslog": {
    "enable": "yes",
    "primaryServer": {
        "ip": "172.19.0.5",
        "port": "5514"
    },
    "secondaryServer": {
        "ip": "",
        "port": ""
    },
    "trafficLogs": {
        "enable": "yes"
    }
}
```

Configure "enable": "yes" and put a value on "primaryServer", "ip" and "port". The secondary servers is optional.

If you want the CSC to collect traffic logs, put "trafficLogs", "enable": "yes"

14.2.3.4 priCPA

```
    "priCPA": {
    "enable": "yes",
    "nodeName": "pricpa-csc-aZ-doc",
    "location": "Azure East US",
    "description": "PriCPA Node for Documentation",
    "publicUdpPort": "51820",
    "privateCirdIp": "192.168.7.25/24",
    "persistentKeepAlive": "no",
    "peersJsonFileUrl": "https://mhb-netskope-private.s3.eu-west-1.amazonaws.com/privateAccessPeersConfig-LAB2.json",
    "remoteManagementNetworks": [
        "172.19.0.0/24",
        "192.168.1.0/24",
        "192.168.6.0/24"
    ]
},
```

In this section, you can configure the Local values for PriCPA, the Peers URL and the Remote Management Networks. See section 8.1.1 for more details.

14.2.3.5 sshRestrictions


```
  "sshRestrictions": {
     "eth1": {
        "enable": "yes",
        "allowedNetworks": [
           "10.2.0.0/16",
           "172.19.0.0/24",
           "192.168.1.0/24",
           "192.168.6.0/24"
        ]
     },
     "wg0": {
        "enable": "yes",
        "allowedNetworks": [
           "10.2.0.0/16",
           "172.19.0.0/24",
           "192.168.1.0/24"
        ]
     }
  },
```


In this section, you can configure from which networks you can access the CSC via SSH. You can configure when the traffic arrives from the local internal interface (eth1) or via PriCPA (wg0).

14.2.3.6 adminManagement


```
  "adminManagement": {
     "csccli": {
        "enable": "yes",
        "sshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDPm+99wX1/ZhtDIKWh+Uv4TrEYboLoLJlRV6NZctrrkbpq/
WuSCtY9ghL456s4jmSSaNzSSCZ5ywpa3LxmYZ60huoUvYEXBR+lj7MX+trVsifYUe6aJgjPzH8q3x2X72bS20jBQovrNoeN6DZRWWLzLZ4xyczOF+samsm6I/
O3jop68KG6+FydfxFM4DddlJrw29sMi9BJmOzA0EjI2r3x/
Niz+PWqgbvg5Aq9+uPbJsd6t5egsBsXsKi62bIv2rX5hMyZQbxpjcq7BUAc4QcxwZH76X2Y3QVkGnjKXjkFJMcLd6qzN6Su3yYqn41H8ffN2C0rSKD38fvwNDDJTkmZ9
3PW9mWQweNuWMvLxQTG14z0qR9VDnIkMtxiJCZyJpUYe6RQDga0nweFIOGBO7N9fA/KzA8r/GjI52E5KlEQQ725pQXcjHZHFjzTeiD1ZjjEBAURUx0DLldBTsO/
oisGUT+pZNQAx6gmX/YxDE/le7qdjuf48aHjwH+uGL1/Q0= cscadmin"
     }
  }
```


In this section, you can enable access to the terminal console using the "csccli" user. You need to add here the SSH Public Key.

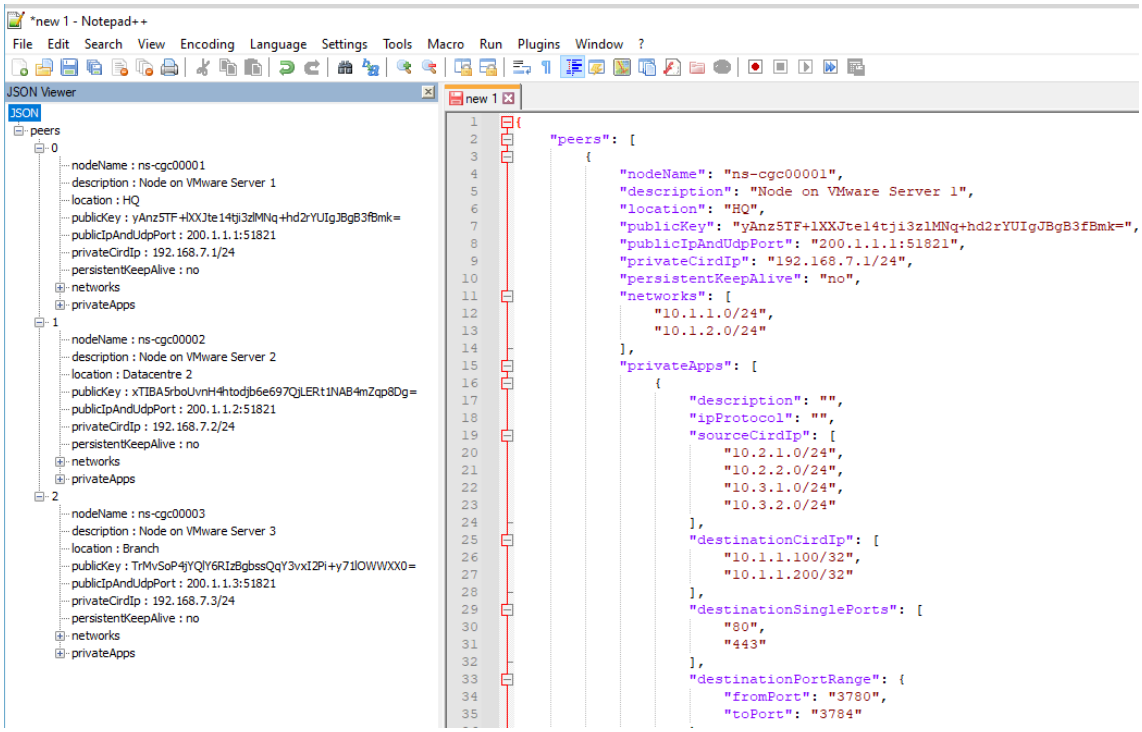
14.3 Appendix C: JSON formatters (Visual Code, Notepad ++)

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

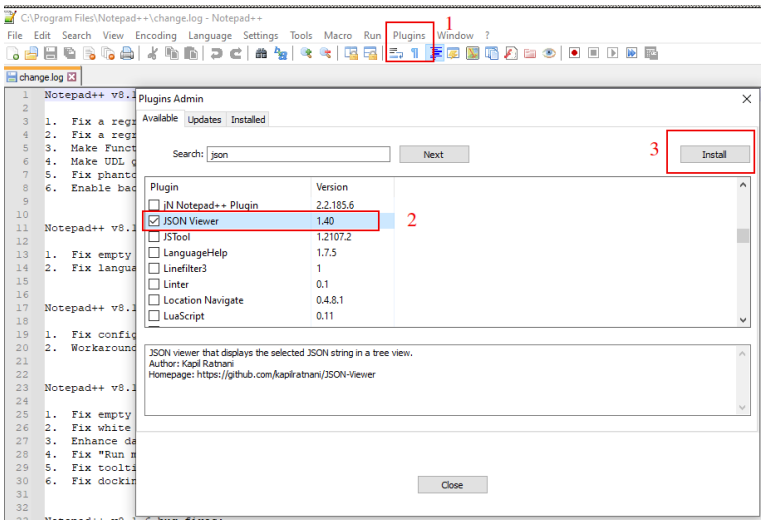
14.3.1 Visual Code

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"

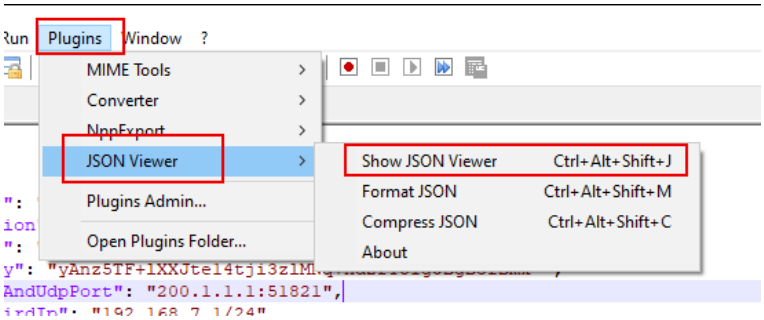
14.3.2 Notepad ++



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



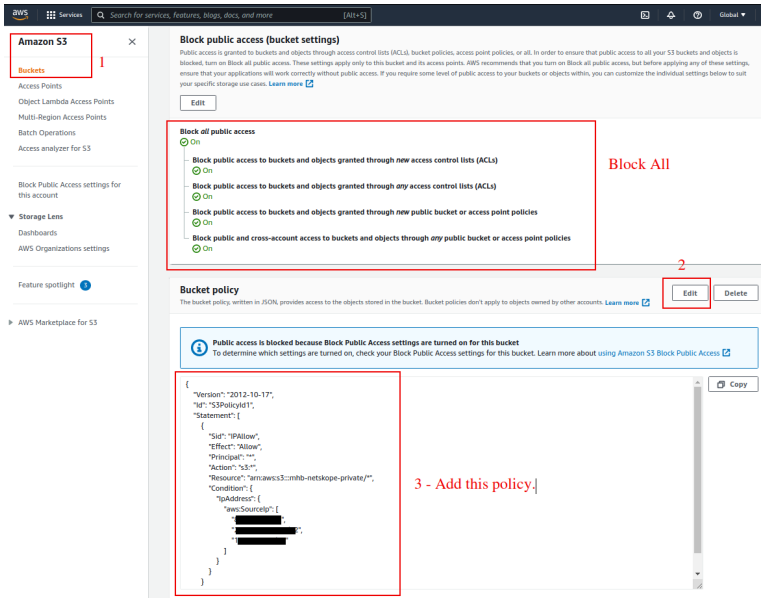
3. Create your JSON file.
4. To Check your JSON file go to: Plugins -> JSON Viewer -> Show JSON Viewer.



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

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

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




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

3. Done!