Configuración de CSR1000v HA versión 3 en AWS, Azure y GCP

Contenido

Introducción

Este documento describe los pasos para configurar los routers CSR1000v para la versión 3 de alta disponibilidad (HAv3) en Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP).

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Nubes AWS, Azure o GCP.
- Routers CSR1000v.
- Cisco IOS®-XE.

Este artículo asume que la configuración de red subyacente ya se ha completado y se centra en la configuración de HAv3.

Los detalles completos de la configuración se encuentran en la Guía de Configuración del Software Cisco CSR 1000v y Cisco ISRv.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Una cuenta AWS, Azure o GCP.
- 2 routers CSR1000v.
- Un mínimo de Cisco IOS®-XE Polaris 16.11.1s

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si su red está activa, asegúrese de comprender el impacto potencial de cualquier comando.

Antecedentes

Cisco recomienda que tenga conocimiento de diferentes versiones de HA disponibles:

- HAv1: La configuración HA se realiza como comandos IOS y se basa en BFD como el mecanismo para detectar fallas.
- HAv2/HAv3: La implementación se ha movido al contenedor del shell de invitado como scripts python. BFD es opcional y se pueden escribir secuencias de comandos personalizadas para detectar fallas y activar fallas. La configuración de Azure HAv2 es en gran medida similar a HAv3 con diferencias menores en los paquetes de instalación de pip y en la configuración de redundancia de IOS.
- HAv3: La implementación de HA se ha movido en gran medida fuera del código Cisco IOS®-XE y se ejecuta en el contenedor de la consola de invitado.

HAv3 está disponible en Cisco IOS®-XE Polaris 16.11.1s y agrega varias nuevas funciones:

- **Independiente de la nube:** Esta versión de alta disponibilidad funciona en los routers CSR 1000v en cualquier proveedor de servicios en la nube. Aunque existen algunas diferencias en la terminología y los parámetros de la nube, el conjunto de funciones y scripts utilizados para configurar, controlar y mostrar las funciones de alta disponibilidad son comunes entre los diferentes proveedores de servicios en la nube. La versión 3 de alta disponibilidad (HAv3) es compatible con los routers CSR 1000v en AWS, Azure y GCP. El soporte para el proveedor GCP se ha agregado en 16.11.1. Consulte con Cisco para conocer la compatibilidad actual con la alta disponibilidad en las nubes de cada proveedor.
- **Operación activa/activa:** Puede configurar ambos routers Cisco CSR 1000v para que estén activos simultáneamente, lo que permite compartir la carga. En este modo de funcionamiento, cada ruta en una tabla de rutas tiene uno de los dos routers que sirven como router primario y el otro router como router secundario. Para habilitar el uso compartido de carga, tome todas las rutas y dividéndolas entre los dos routers Cisco CSR 1000v. Tenga en cuenta que esta funcionalidad es nueva para las nubes basadas en AWS.
- **Reversión a CSR primario después de la recuperación de fallas:** Puede designar un Cisco CSR 1000v como el router principal para una ruta determinada. Mientras que este Cisco CSR 1000v está activo. es el siguiente salto para la ruta. Si este CSR 1000v de Cisco falla, el Cisco CSR 1000v de igual a igual toma el relevo como el salto siguiente para la ruta, manteniendo la conectividad de red. Cuando el router original se recupera de la falla, reclama la propiedad de la ruta y es el router de salto siguiente. Esta funcionalidad también es nueva para las nubes basadas en AWS.
- **Guiones proporcionados por el usuario:** El shell de invitado es un contenedor en el que puede implementar sus propios scripts. HAv3 expone una interfaz de programación a scripts

proporcionados por el usuario. Esto implica que ahora puede escribir secuencias de comandos que puedan desencadenar eventos de conmutación por fallas y de reversión. También puede desarrollar sus propios algoritmos y activadores para controlar qué Cisco CSR 1000v proporciona los servicios de reenvío para una ruta determinada. Esta funcionalidad es nueva para las nubes basadas en AWS.

- **Nuevo mecanismo de configuración e implementación:** La implementación de HA se ha movido fuera del código Cisco IOS®-XE. El código de alta disponibilidad se ejecuta ahora en el contenedor del shell de invitado. Para obtener más información sobre guestshell, consulte la sección "Shell de invitado" de la Guía de configuración de capacidad de programación. En HAv3, la configuración de nodos de redundancia se realiza en el shell de invitado que utiliza un conjunto de scripts Python. Esta función se ha introducido ahora para las nubes basadas en AWS.

Nota: Los recursos implementados en AWS, Azure o GCP a partir de los pasos de este documento pueden generar un costo.

Topología

Antes de que comience la configuración, es importante comprender completamente la topología y el diseño. Esto ayuda a resolver cualquier problema potencial más adelante.

Aunque el diagrama de topología de red se basa en AWS, la implementación de red subyacente entre nubes es relativamente similar. La topología de red también es independiente de la versión de HA utilizada, ya sea HAv1, HAv2 o HAv3.

Para este ejemplo de topología, la redundancia HA se configura con estos ajustes en AWS:

- 1x - Región
- 1x - VPC
- 3 veces: zonas de disponibilidad
- 4 interfaces/subredes de red (2 interfaces públicas/2 caras privadas)
- 2 tablas de ruta (pública y privada)
- 2 routers CSR1000v (Cisco IOS®-XE 17.01.01)

Hay dos routers CSR1000v en un par HA, en dos zonas de disponibilidad diferentes. La tercera zona es una instancia privada, que simula un dispositivo en un Data Center privado. Generalmente, todo el tráfico normal debe fluir a través de la tabla de ruta privada (o interna).

Diagrama de la red

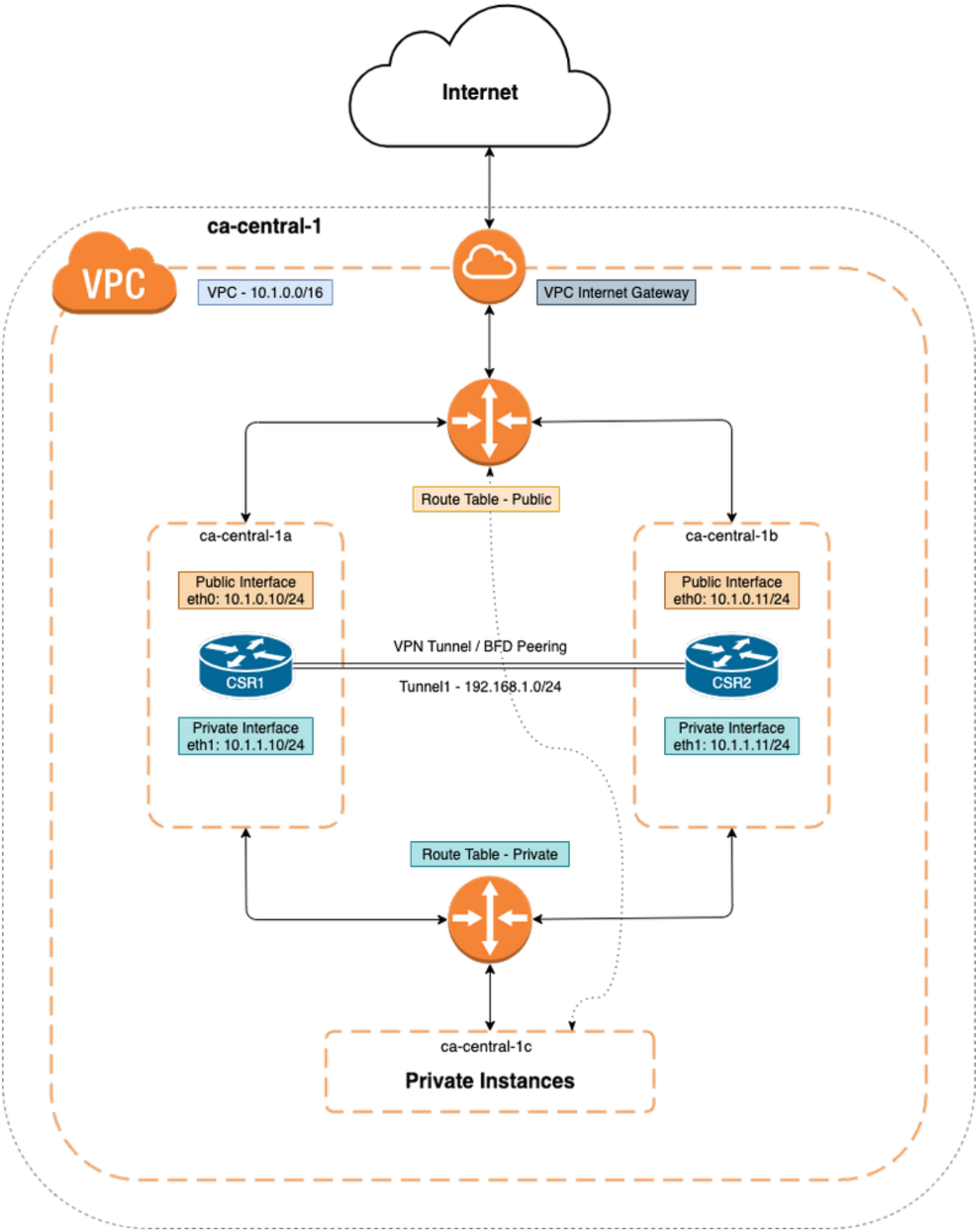


Diagrama de la red

Configuración de routers CSR1000v

Configuración independiente de la nube

Paso 1. Configure el alojamiento de aplicaciones IOX y el shell de invitado, esto proporciona alcance de ip en el shell de invitado. Este paso se puede configurar automáticamente de forma predeterminada cuando se despliega CSR1000v.

vrf definition GS ! iox app-hosting appid guestshell app-vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress 192.168.35.102 netmask 255.255.255.0 app-default-gateway 192.168.35.101 guest-interface 0 name-server0 8.8.8.8 ! interface VirtualPortGroup0 vrf forwarding GS ip address 192.168.35.101 255.255.255.0 ip nat inside ! interface GigabitEthernet1 ip nat outside ! ip access-list standard GS_NAT_ACL permit 192.168.35.0 0.0.0.255 ! ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 vrf GS overload ! ! The static route points to the G1 ip address's gateway ip route vrf GS 0.0.0.0 0.0.0.0 GigabitEthernet1 10.1.0.1 global

Paso 2. Habilite e inicie sesión en el shell de invitado.

```
Device#guestshell enable
Interface will be selected if configured in app-hosting
Please wait for completion
guestshell installed successfully
Current state is: DEPLOYED
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully

Device#guestshell
[guestshell@guestshell ~]$
```

> **Nota:** Para obtener más información sobre el shell de invitado, consulte - Guía de Configuración de Programmability

Paso 3. Confirme que el shell pueda comunicarse con Internet.

```
[guestshell@guestshell ~]$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=109 time=1.74 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=2.19 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=109 time=2.49 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=109 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=109 time=3.04 ms
```

Paso 4. (Opcional) Habilite la detección de reenvío bidireccional (BFD) y un protocolo de routing como protocolo de routing de gateway interior mejorado (EIGRP) o protocolo de gateway fronterizo (BGP) en el túnel para la detección de fallos entre pares. Configure un túnel VxLAN o IPsec entre los routers Cisco CSR 1000v.

- Túnel IPsec entre los routers Cisco CSR 1000v.

crypto isakmp policy 1 encr aes 256 authentication pre-share crypto isakmp key cisco address crypto ipsec transform-set uni-perf esp-aes 256 esp-sha-hmac mode tunnel crypto ipsec profile vti-1 set security-association lifetime kilobytes disable set security-association lifetime seconds 86400 set transform-set uni-perf set pfs group2 interface Tunnel1 ip address 192.168.1.1 255.255.255.0 bfd interval 500 min_rx 500 multiplier 3 tunnel source GigabitEthernet1 tunnel destination redundancy cloud-ha bfd peer Example - #CSR1 ! interface Tunnel1 ip address 192.168.1.1 255.255.255.0 bfd interval 500 min_rx 500 multiplier 3 tunnel source GigabitEthernet1 tunnel destination 10.1.0.11 ! redundancy cloud-ha bfd peer 192.168.1.2 #CSR2 ! interface Tunnel1 ip address 192.168.1.2 255.255.255.0 bfd interval 500 min_rx 500 multiplier 3 tunnel source GigabitEthernet1 tunnel destination 10.1.0.10 ! redundancy cloud-ha bfd peer 192.168.1.1

- Túnel VxLAN entre los routers Cisco CSR 1000v.

Example: interface Tunnel100 ip address 192.168.1.1 255.255.255.0 bfd interval 500 min_rx 500 multiplier 3 tunnel source GigabitEthernet1 tunnel mode vxlan-gpe ipv4 tunnel destination tunnel vxlan vni 10000 redundancy cloud-ha bfd peer

Paso 4.1. (Opcional) Configure EIGRP sobre Interfaces de Túnel.

router eigrp 1 bfd interface Tunnel1 network 192.168.1.0 0.0.0.255

- Se pueden utilizar scripts personalizados para activar la conmutación por fallas, por ejemplo:

event manager applet Interface_GigabitEthernet2 event syslog pattern “Interface GigabitEthernet2, changed state to administratively down” action 1 cli command “enable” action 2 cli command “guestshell run node_event.py -i 10 -e peerFail” exit exit

Configuración específica de AWS

- Parámetros HA de AWS

Parameter	Switch	Description
Node Index	-i	Index that is used to uniquely identify this node. Valid values: 1-1023.
Region Name	-rg	Name of the region that contains the route table. For example, us-west-2.
Route Table Name	-t	Name of the route table to be updated. The name of the route table must begin with the substring rtb-. For example, rtb-001333c29ef2aec5f
Route	-r	If a route is unspecified, then the redundancy node is considered to apply to all routes in the routing table. The CSR cannot change routes which are of type local or gateway.
Next Hop Interface	-n	Name of the interface to which packets should be forwarded in order to reach the destination route. The name of the interface must begin with the substring eni-. For example, eni-07160c7e740ac8ef4.
Mode	-m	Indicates whether this router is the primary or secondary router for servicing this route. Valid values are primary or secondary. This is an optional parameter. The default value is secondary.

Paso 1. Configure la autenticación con IAM.

Para que el router CSR1000v actualice una tabla de ruteo en la red AWS, el router debe autenticarse. En AWS, debe crear una política que permita al router CSR 1000v acceder a la tabla de rutas. A continuación, se crea una función IAM que utiliza esta política y se aplica al recurso EC2.

Después de crear las instancias CSR 1000v EC2, la función IAM creada debe estar conectada a cada router.

La política utilizada en la nueva función de IAM es:

{ "Version": "2012-10-17", "Statement": [{ "Sid": "VisualEditor0", "Effect": "Allow", "Action": ["logs:CreateLogStream", "cloudwatch:", "s3:", "ec2:AssociateRouteTable", "ec2:CreateRoute", "ec2:CreateRouteTable", "ec2:DeleteRoute", "ec2:DeleteRouteTable", "ec2:DescribeRouteTables", "ec2:DescribeVpcs", "ec2:ReplaceRoute", "ec2:DescribeRegions", "ec2:DescribeNetworkInterfaces", "ec2:DisassociateRouteTable", "ec2:ReplaceRouteTableAssociation", "logs:CreateLogGroup", "logs:PutLogEvents"], "Resource": "*" }] }

Nota: Refiérase a la función IAM con una política y asóciela al VPC para ver los pasos detallados.

Paso 2. Instale el paquete HA python.

```
[guestshell@guestshell ~]$ pip install csr_aws_ha --user
[guestshell@guestshell ~]$ source ~/.bashrc
```

Paso 3. Configure los parámetros HA en el router primario.

```
[guestshell@guestshell ~]$ create_node.py -i 10 -t rtb-01c5b0633a3422575 -rg ca-central-1 -n eni-0bc1912748614df2a -r 0.0.0.0/0 -m primary
```

Paso 4. Configure los parámetros HA en el router secundario.

```
[guestshell@guestshell ~]$ create_node.py -i 10 -t rtb-01c5b0633a3422575 -rg ca-central-1 -n eni-0e351ab1b8f416728 -r 0.0.0.0/0 -m secondary
```

- El formato del nodo es:

```
create_node.py -i n -t rtb-private-route-table-id -rg region-id -n eni-CSR-id -r route(x.x.x.x/x) -m
```

Configuración específica de Azure

- Azure HA Parameters

The following table specifies the redundancy parameters that are specific to Microsoft Azure:

Parameter Switch	Switch	Description
Node Index	-i	The index that is used to uniquely identify this node. Valid values: 1–255.
Cloud Provider	-p	Specifies the type of Azure cloud: azure, azusgov, or azchina.
Subscription ID	-s	The Azure subscription id.
Resource Group Name	-g	The name of the route table to be updated.
Route Table Name	-t	The name of the route table to be updated.
Route	-r	IP address of the route to be updated in CIDR format. Can be IPv4 or IPv6 address. If a route is unspecified, then the redundancy node is considered to apply to all routes in the routing table of type “virtual appliance”.
Next Hop Address	-n	The IP address of the next hop router. Use the IP address that is assigned to this CSR 1000v on the subnet which utilizes this route table. Can be an IPv4 or IPv6 address.
Mode	-m	Indicates whether this router is the primary or secondary router for servicing this route. Default value is secondary.

Nota: La interfaz de cara externa se debe configurar en GigabitEthernet1. Esta es la interfaz utilizada para alcanzar las API de Azure. De lo contrario, HA no puede funcionar correctamente. Dentro del shell, asegúrese de que el comando curl pueda obtener metadatos de Azure.

```
[guestshell@guestshell ~]$ curl -H "Metadata:true" http://169.254.169.254/metadata/instance?api-version=2020-06-01
```

Paso 1. La autenticación para llamadas API CSR1000v debe estar habilitada con Azure Active Directory (AAD) o con la identidad de servicio administrado (MSI). Consulte Configuración de la Autenticación para Llamadas API CSR1000v para ver los pasos detallados. Sin este paso, el

router CSR1000v no puede estar autorizado para actualizar la tabla de rutas.

Parámetros AAD

Parameter Name	Switch	Description
Cloud Provider	-p	Specifies which Azure cloud is in use {azure | azusgov | azchina}
Tenant ID	-d	Identifies the AAD instance.
Application ID	-a	Identifies the application in AAD.
Application Key	-k	Access key that is created for the application. Key should be specified in unencoded URL format.

Paso 2. Instale el paquete HA python.

```
[guestshell@guestshell ~]$ pip install csr_azure_ha --user
[guestshell@guestshell ~]$ source ~/.bashrc
```

Paso 3. Configure los parámetros HA en el router primario (se puede utilizar MSI o AAD para este paso).

- Con autenticación MSI.

```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx -g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.1.0.10 -m primary
```

- Con autenticación AAD (se requieren indicadores -a, -d, -k adicionales).

```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx -g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.1.0.10 -m primary -a 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxx -d ae49849c-2622-4d45-b95e-xxxxxxxxx -k bDEN1k8batJqpeqjAuUvaUCZn5Md6rWEi=
```

Paso 4. Configure los parámetros HA en el router secundario.

- Con autenticación MSI

```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx -g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.1.0.11 -m secondary
```

- Con autenticación AAD (se requieren indicadores adicionales -a, -d, -k)

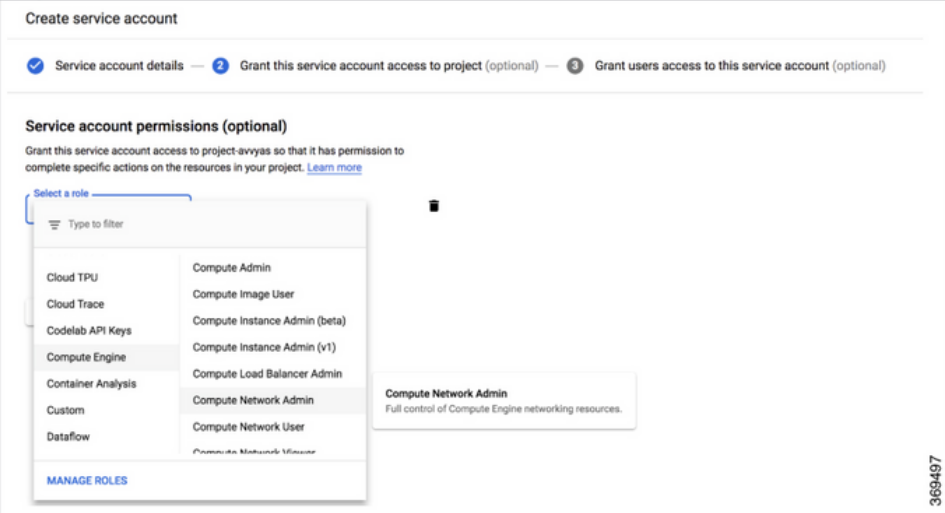
```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx --g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.0.0.11 -m secondary -a 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxx -d ae49849c-2622-4d45-b95e-xxxxxxxxx -k bDEN1k8batJqpeqjAuUvaUCZn5Md6rWEi=
```

Configuración específica de GCP

- Parámetros GCP HA

Parameter	Is this parameter required?	Switch	Description
Node Index	Yes	-i	The index that is used to uniquely identify this node. Valid values: 1-255.
Cloud Provider	Yes	-p	Specify gcp for this parameter.
Project	Yes	-g	Specify the Google Project ID.
routeName	Yes	-a	The route name for which this CSR is next hop. For example from Fig. 2, if we are configuring node on CSR 1, this would be route-vpc2-csr1.
peerRouteName	Yes	-b	The route name for which the BFD peer CSR is next hop. For example from Fig. 2, if we are configuring node on CSR 1, this would be route-vpc2-csr2.
Route	yes	-r	The IP address of the route to be updated in CIDR format. Can be IPv4 or IPv6 address. If a route is unspecified, then the redundancy node is considered to apply to all routes in the routing table of type virtual appliance. Note: Currently Google cloud does not have IPv6 support in VPC.
Next hop address	Yes	-n	The IP address of the next hop router. Use the IP address that is assigned to this CSR 1000v on the subnet which utilizes this route table. The value can be an IPv4 or IPv6 address. Note: Currently Google cloud does not have IPv6 support in VPC.
hopPriority	Yes	-o	The route priority for the route for which the current CSR is the next hop.
VPC	Yes	-v	The VPC network name where the route with the current CSR as the next hop exists.

Nota: Asegúrese de que la cuenta de servicio asociada a los routers CSR 1000v tenga al menos un permiso de administración de red informática.

Command or Action	Purpose
Ensure that the service account associated with the CSR 1000v routers at least have a Compute Network Admin permission.	 You can also provide the required permissions in a credentials file with name 'credentials.json' and place it under the /home/guestshell directory. The credentials file overrides the permissions supplied through the service account associated with the CSR 1000v instance.

Paso 1. Instale el paquete HA python.

```
[guestshell@guestshell ~]$ pip install csr_gcp_ha --user
[guestshell@guestshell ~]$ source ~/.bashrc
```

Paso 2. Configure los parámetros HA en el router primario.

[guestshell@guestshell ~]$ create_node -i 1 -g -r dest_network -o 200 -n nexthop_ip_addr -a route-vpc2-csr1 -b route-vpc2-csr2 -p gcp -v vpc_name

Paso 3. Configure los parámetros HA en el router secundario.

[guestshell@guestshell ~]$ create_node -i 1 -g -r dest_network -o 200 -n nexthop_ip_addr -a route-vpc2-csr2 -b route-vpc2-csr1 -p gcp -v vpc_name

Verificación

Use esta sección para confirmar que su configuración funciona correctamente.

Paso 1. Activa una conmutación por fallas con el indicador node_event.py peerFail.

[guestshell@guestshell ~]$ node_event.py -i 10 -e peerFail 200: Node_event processed successfully

Paso 2. Navegue hasta la Tabla de rutas privadas de su proveedor de nube, verifique que la ruta haya actualizado el siguiente salto a la nueva dirección IP.

Troubleshoot

Actualmente, no hay información específica de troubleshooting disponible para esta configuración.

Información Relacionada

- Encontrará pasos detallados de configuración de HAv3 en Cisco CSR 1000v y en la Guía de Configuración del Software Cisco ISRv
- La configuración de Azure HAv2 es en gran medida similar a HAv3 con diferencias menores en los paquetes de instalación de pip y en la configuración de redundancia de IOS. La documentación se encuentra en la Guía de Configuración de CSR1000v HA Versión 2 en Microsoft Azure
- La configuración de Azure HAv1 con CLI se encuentra en la Guía de implementación de redundancia HA CSR1000v en Microsoft Azure con AzureCLI 2.0
- La configuración de AWS HAv1 se encuentra en la Guía de implementación de redundancia de HA CSR1000v en Amazon AWS
- Soporte Técnico y Documentación - Cisco Systems