Configurar CSR1000v HA versão 3 em AWS, Azure e GCP

Contents

Introduction

Este documento descreve as etapas para configurar os roteadores CSR1000v para High Availability Version 3 (HAv3) em Amazon Web Services (AWS), Microsoft Azure e Google Cloud Platform (GCP).

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Nuvens AWS, Azure ou GCP.
- Roteadores CSR1000v.
- Cisco IOS®-XE.

Este artigo pressupõe que a configuração de rede subjacente já foi concluída e se concentra na configuração de HAv3.

Os detalhes completos da configuração estão no Guia de configuração do software Cisco CSR 1000v e Cisco ISRv.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Uma conta AWS, Azure ou GCP.
- 2 roteadores CSR1000v.
- Um 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. Se a sua rede estiver ativa, certifique-se de que você entende o impacto potencial de qualquer comando.

Informações de Apoio

A Cisco recomenda que você tenha conhecimento de diferentes versões de HA disponíveis:

- HAv1: A configuração de HA é executada como comandos IOS e depende do BFD como o mecanismo para detectar falha.
- HAv2/HAv3: A implementação foi movida para o contêiner do shell como scripts python. O BFD é opcional e scripts personalizados podem ser gravados para detectar falha e failover de disparo. A configuração do Azure HAv2 é em grande parte semelhante ao HAv3 com diferenças menores nos pacotes de instalação do pip e na configuração de redundância do IOS.
- HAv3: A implementação do HA foi largamente removida do código Cisco IOS®-XE e é executada no contêiner do shell de convidado.

O HAv3 está disponível no Cisco IOS®-XE Polaris 16.11.1s e adiciona vários novos recursos:

- **Independente da nuvem:** Essa versão de alta disponibilidade funciona em roteadores CSR 1000v em qualquer provedor de serviços em nuvem. Embora existam algumas diferenças na terminologia e nos parâmetros da nuvem, o conjunto de funções e scripts usados para configurar, controlar e mostrar os recursos de alta disponibilidade são comuns nos diferentes provedores de serviços de nuvem. A versão 3 de alta disponibilidade (HAv3) é suportada em roteadores CSR 1000v em AWS, Azure e GCP. O suporte para o provedor de GCP foi adicionado em 16.11.1. Verifique com a Cisco se há suporte atual para alta disponibilidade nas nuvens do provedor individual.
- **Operação ativa/ativa:** Você pode configurar os roteadores Cisco CSR 1000v para estarem ativos simultaneamente, o que permite o compartilhamento de carga. Nesse modo de operação, cada rota em uma tabela de rotas tem um dos dois roteadores que serve como roteador principal e o outro como roteador secundário. Para habilitar o compartilhamento de carga, pegue todas as rotas e divida-as entre os dois roteadores Cisco CSR 1000v. Observe que essa funcionalidade é nova para nuvens baseadas em AWS.
- **Reversão para CSR primário após recuperação de falhas:** Você pode designar um Cisco CSR 1000v como o roteador principal para uma rota específica. Enquanto o Cisco CSR 1000v está ativo. é o próximo salto para a rota. Se esse Cisco CSR 1000v falhar, o peer Cisco CSR 1000v assumirá como o próximo salto para a rota, mantendo a conectividade de rede. Quando o roteador original se recupera da falha, recupera a propriedade da rota e é o roteador do próximo salto. Essa funcionalidade também é nova para as nuvens baseadas em AWS.
- **Scripts fornecidos pelo usuário:** O shell de convidado é um contêiner no qual você pode implantar seus próprios scripts. O HAv3 expõe uma interface de programação a scripts fornecidos pelo usuário. Isso significa que agora você pode gravar scripts que podem

disparar eventos de failover e reversão. Você também pode desenvolver seus próprios algoritmos e acionadores para controlar quais Cisco CSR 1000v fornece os serviços de encaminhamento para uma determinada rota. Essa funcionalidade é nova para nuvens baseadas em AWS.
- **Novo mecanismo de configuração e implantação:** A implementação do HA foi removida do código Cisco IOS®-XE. O código de alta disponibilidade agora é executado no contêiner do shell de convidados. Para obter mais informações sobre o shell de convidado, consulte a seção "Guest Shell" no Guia de configuração de programabilidade. Em HAv3, a configuração de nós de redundância é executada no shell de convidado que usa um conjunto de scripts Python. Este recurso foi introduzido para nuvens baseadas em AWS.

  Note: Os recursos implantados no AWS, Azure ou GCP nas etapas deste documento podem acarretar um custo.

Topologia

Antes do início da configuração, é importante entender a topologia e o projeto completamente. Isso ajuda a solucionar possíveis problemas posteriormente.

Embora o diagrama de topologia de rede seja baseado em AWS, a implantação de rede subjacente entre nuvens é relativamente semelhante. A topologia de rede também é independente da versão HA usada, seja HAv1, HAv2 ou HAv3.

Para este exemplo de topologia, a redundância de HA é configurada com estas configurações no AWS:

- 1x - Região
- 1x - VPC
- 3x - Zonas de disponibilidade
- 4x - Interfaces/sub-redes de rede (2x público face/2x privado face)
- 2x - Tabelas de rota (públicas e privadas)
- 2x - Roteadores CSR1000v (Cisco IOS®-XE 17.01.01)

Há 2x roteadores CSR1000v em um par de HA, em duas zonas de disponibilidade diferentes. A terceira zona é uma instância privada, que simula um dispositivo em um datacenter privado. Geralmente, todo o tráfego normal deve fluir pela tabela de rota privada (ou interna).

Diagrama de Rede

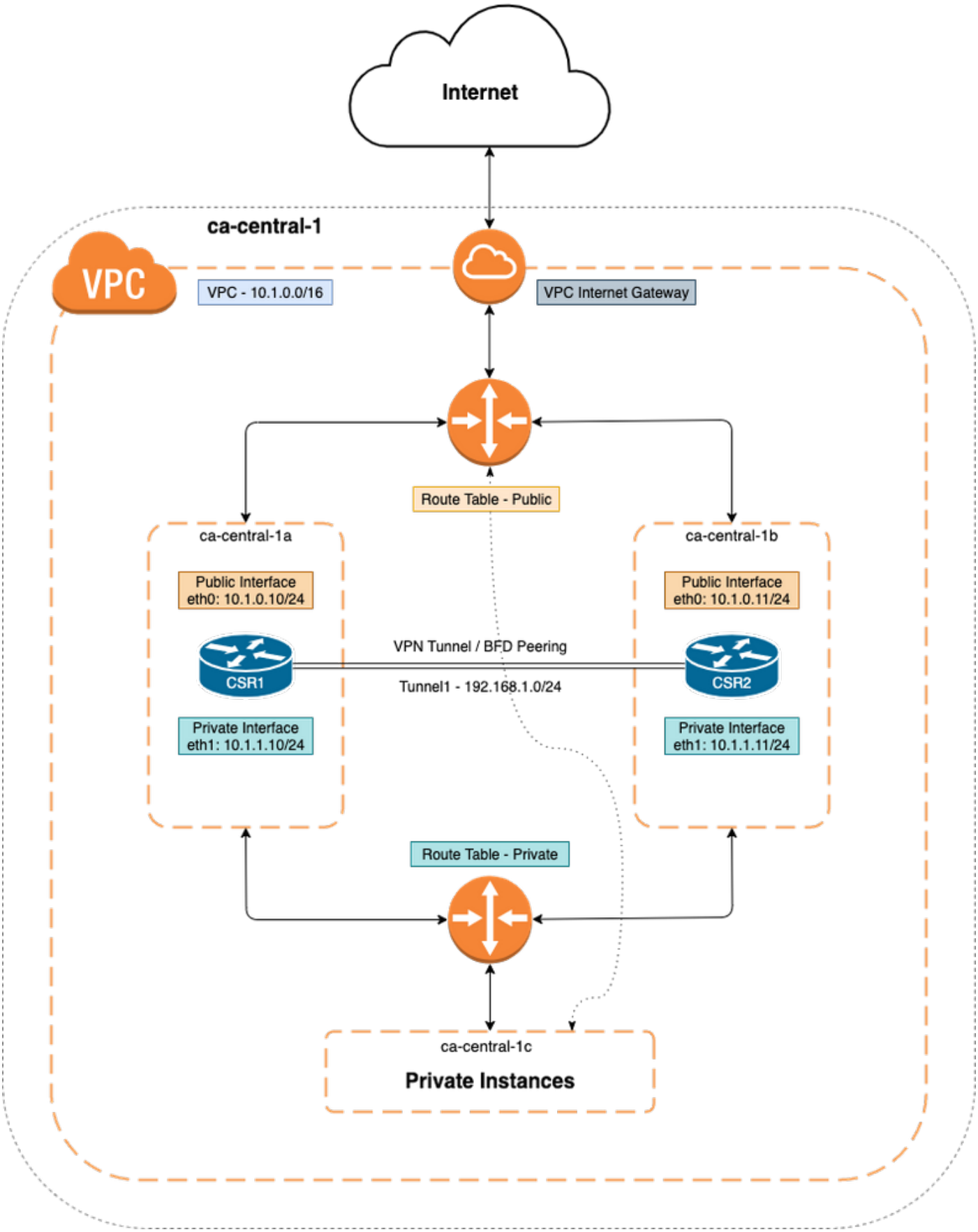


Diagrama de Rede

Configurar os roteadores CSR1000v

Configuração independente da nuvem

Etapa 1. Configure a hospedagem de aplicativos e o shell de convidados IOX, o que fornece acessibilidade de ip para o shell de convidados. Essa etapa pode ser configurada automaticamente por padrão quando o CSR1000v é instalado.

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

Etapa 2. Habilitar e fazer login no guestshell.

```
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 ~]$
```

> **Note**: Para obter mais informações sobre o shell, consulte - Guia de configuração de programabilidade

Etapa 3. Confirme se o guestshell consegue se comunicar com a 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
```

Etapa 4. (Opcional) Ative a detecção de encaminhamento bidirecional (BFD) e um protocolo de roteamento como EIGRP (Enhanced Interior Gateway Routing Protocol) ou BGP (Border Gateway Protocol) no túnel para detecção de falha de peer. Configure um túnel VxLAN ou IPsec entre os roteadores Cisco CSR 1000v.

- Túnel IPsec entre os roteadores 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 os roteadores 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

Etapa 4.1. (Opcional) Configure o EIGRP sobre Interfaces de Túnel.

router eigrp 1 bfd interface Tunnel1 network 192.168.1.0 0.0.0.255

- Scripts personalizados podem ser usados para disparar failover, por exemplo:

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

Configuração específica do AWS

- Parâmetros AWS HA

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.

Etapa 1. Configure a autenticação com IAM.

Para que o roteador CSR1000v atualize uma tabela de roteamento na rede AWS, o roteador precisa ser autenticado. No AWS, você deve criar uma política que permita que o roteador CSR 1000v acesse a tabela de rotas. Uma função do IAM é criada, que usa essa política e é aplicada ao recurso EC2.

Depois que as instâncias CSR 1000v EC2 são criadas, a função IAM criada precisa ser conectada a cada roteador.

A política usada na nova função do IAM é:

{ "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": "*" }] }

Note: Consulte a função do IAM com uma Política e associe-a ao VPC para obter etapas detalhadas.

Etapa 2. Instale o pacote HA python.

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

Etapa 3. Configure os parâmetros HA no roteador primário.

```
[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
```

Etapa 4. Configure os parâmetros HA no roteador secundário.

```
[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
```

- O formato do nó é:

```
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
```

Configuração Específica do Azure

- Parâmetros do Azure HA

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.

Note: A interface externa deve ser configurada em GigabitEthernet1. Esta é a interface usada para acessar as APIs do Azure. Caso contrário, o HA não pode funcionar corretamente. No guestshell, certifique-se de que o comando curl pode buscar metadados do Azure.

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

Etapa 1. A autenticação para Chamadas API CSR1000v deve ser ativada com o Azure Ative Diretory (AAD) ou com a Identidade de Serviço Gerenciado (MSI). Consulte Configurar Autenticação para Chamadas de API CSR1000v para obter etapas detalhadas. Sem essa etapa,

o roteador CSR1000v não pode ser autorizado a atualizar a tabela de rotas.

Parâmetros 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.

Etapa 2. Instale o pacote HA python.

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

Etapa 3. Configure os parâmetros HA no roteador principal (MSI ou AAD podem ser usados para esta etapa).

- Com autenticação 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
```

- Com autenticação AAD (são necessários flags -a, -d, -k adicionais).

```
[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=
```

Etapa 4. Configure os parâmetros HA no roteador secundário.

- Com autenticação 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
```

- Com autenticação AAD (são necessários flags -a, -d, -k adicionais)

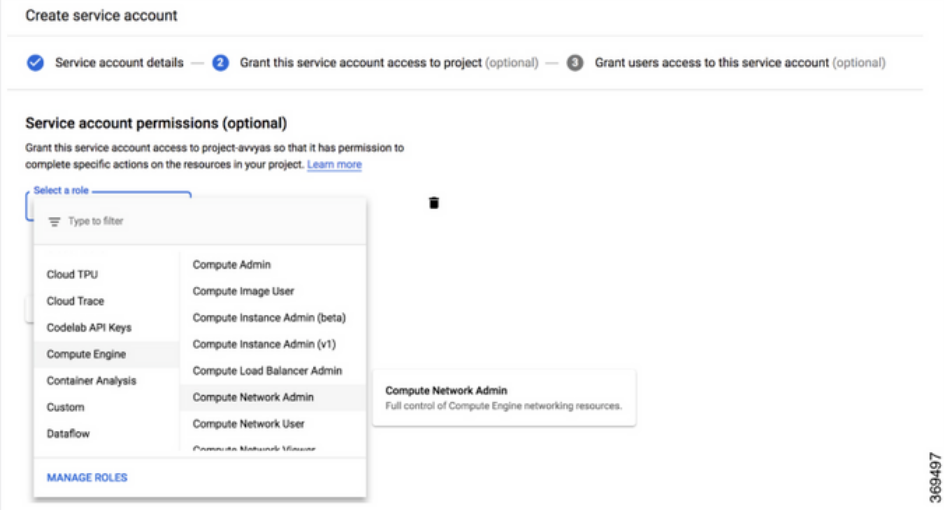
```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx --g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.0.0.11 -m secondary -a 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxx -d ae49849c-2622-4d45-b95e-xxxxxxxxx -k bDEN1k8batJqpeqjAuUvaUCZn5Md6rWEi=
```

Configuração específica de GCP

- Parâmetros de HA de GCP

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.

Note: Certifique-se de que a conta de serviço associada aos roteadores CSR 1000v tenha pelo menos uma permissão de administrador de rede de computação.

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.

Etapa 1. Instale o pacote HA python.

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

Etapa 2. Configure os parâmetros HA no roteador primário.

[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

Etapa 3. Configure os parâmetros HA no roteador secundário.

[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

Verificar

Use esta seção para confirmar se a sua configuração funciona corretamente.

Etapa 1. Aciona um failover com o flag node_event.py peerFail.

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

Etapa 2. Navegue até a Tabela de Rotas Privadas do seu Provedor de Nuvem, verifique se a rota atualizou o próximo salto para o novo endereço IP.

Troubleshoot

Atualmente, não existem informações disponíveis específicas sobre Troubleshooting para esta configuração.

Informações Relacionadas

- As etapas detalhadas de configuração de HAv3 estão disponíveis no Guia de configuração de software Cisco CSR 1000v e Cisco ISRv
- A configuração do Azure HAv2 é em grande parte semelhante ao HAv3 com diferenças menores nos pacotes de instalação do pip e na configuração de redundância do IOS. A documentação está disponível no Guia de Configuração do CSR1000v HA Versão 2 no Microsoft Azure
- A configuração do Azure HAv1 com CLI é encontrada no Guia de Implantação de Redundância de HA CSR1000v no Microsoft Azure com AzureCLI 2.0
- A configuração AWS HAv1 é encontrada no Guia de implantação de redundância de HA do CSR1000v no Amazon AWS
- Suporte Técnico e Documentação - Cisco Systems