

CSR1000v HA versie 3 configureren op AWS, uurs en GCP

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Topologie](#)

[Netwerkdigram](#)

[CSR1000v routers configureren](#)

[Cloudfafhankelijke configuratie](#)

[AWS-specifieke configuratie](#)

[evening Specific Configuration](#)

[GCP specifieke configuratie](#)

[Verifiëren](#)

[Problemen oplossen](#)

[Gerelateerde informatie](#)

Inleiding

In dit document worden de stappen beschreven om CSR1000v-routers voor hoge beschikbaarheid versie 3 (HAV3) te configureren op Amazon Web Services (AWS), Microsoft Messenger en Google Cloud Platform (GCP).

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- AWS, Kuurklokjes of GCP-wolken.
- CSR1000v-routers.
- Cisco IOS®-XE.

Dit artikel gaat ervan uit dat de onderliggende netwerkconfiguratie al is voltooid en richt zich op de HAV3-configuratie.

De volledige configuratiegegevens zijn in de [Cisco CSR 1000v en Cisco ISRv-softwareconfiguratie Guide](#) gevonden.

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Een AWS-, uur- of GCP-account.
- 2 CSR1000v-routers.
- Een minimum aan Cisco IOS®-XE Polaris 16.11.1s

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, zorg er dan voor dat u de mogelijke impact van elke opdracht begrijpt.

Achtergrondinformatie

Cisco raadt aan dat u kennis hebt van verschillende HA-versies die beschikbaar zijn:

- HA v1: HA-configuratie wordt uitgevoerd als IOS-opdrachten en is afhankelijk van BFD als het mechanisme om een storing op te sporen.
- HA v2/HA3: De implementatie is verplaatst naar de guestshell container als python scripts. BFD is optioneel en aangepaste scripts kunnen worden geschreven om een storing op te sporen en failover te activeren. De configuratie van de Koudde HA v2 lijkt grotendeels op HA v3 met kleine verschillen in de pakketten van de pip installeert en IOS overvullige configuratie.
- HA3: De implementatie van HA is grotendeels verwijderd uit de Cisco IOS®-XE code en loopt in de guestshell container.

HA v3 is beschikbaar bij Cisco IOS®-XE Polaris 16.11.1s en voegt verschillende nieuwe functies toe:

- **Cloudagnost:** Deze versie van hoge beschikbaarheid is functioneel op CSR 1000v-routers op elke cloudserviceprovider. Terwijl er een paar verschillen zijn in de wolkenterminologie en de parameters, zijn de set functies en scripts die gebruikt worden om te configureren, te controleren en de hoge beschikbaarheid functies te laten zien gemeenschappelijk voor de verschillende cloudserviceproviders. High Availability versie 3 (HA v3) wordt ondersteund in CSR 1000v-routers op AWS-, KRV-beveiliging en GCP. Ondersteuning voor de GCP-provider is toegevoegd in 16.11.1. Controleer dit bij Cisco voor huidige ondersteuning van een hoge beschikbaarheid in de wolken van de individuele provider.
- **Actieve/actieve bediening:** U kunt beide routers van Cisco CSR 1000v configureren om tegelijkertijd actief te zijn, wat werklastverdeling toestaat. In deze wijze van verrichting, heeft elke route in een routekaart één van de twee routers als primaire router en de andere router als secundaire router dienen. Neem alle routes en splitter deze op tussen de twee Cisco CSR 1000v-routers om het delen van de lading mogelijk te maken. Merk op dat deze functionaliteit nieuw is voor op AWS gebaseerde wolken.
- **Terug naar primaire CSR na fouterstel:** U kunt een Cisco CSR 1000v als primaire router voor een bepaalde route aanwijzen. Terwijl dit Cisco CSR 1000v omhoog is. is het de volgende hop voor de route. Als deze Cisco CSR 1000v mislukt, neemt de peer Cisco CSR 1000v de volgende hop over voor de route, met behoud van de netwerkconnectiviteit. Wanneer de oorspronkelijke router van de mislukking herstelt, vordert het eigendom van de route terug en is de volgende hoprouter. Deze functionaliteit is ook nieuw voor de op AWS gebaseerde wolken.
- **Door de gebruiker meegeleverde geschriften:** De guestshell is een container waarin je je eigen scripts kunt inzetten. HA v3 legt een programmeerinterface bloot aan door de gebruiker

meegeleverde scripts. Dit impliceert dat je nu scripts kunt schrijven die zowel failover- als reversiegebeurtenissen kunnen veroorzaken. U kunt ook uw eigen algoritmen en triggers ontwikkelen om te controleren welke Cisco CSR 1000v de verzendservices voor een bepaalde route biedt. Deze functionaliteit is nieuw voor op AWS gebaseerde wolken.

- **Nieuw configuratie- en implementatiemechanisme:** De implementatie van HA is verwijderd uit de Cisco IOS®-XE code. Hoge beschikbaarheid code draait nu in de guestshell container. Zie het gedeelte "Guest Shell" in de Programmakeuze voor meer informatie over het programma. In HAv3 wordt de configuratie van overtollige knopen uitgevoerd in de guestshell die een reeks Python scripts gebruikt. Deze optie is nu geïntroduceerd voor wolken op basis van AWS.

Opmerking: Resources die worden ingezet in AWS, aken, uur of GCP van de stappen in dit document kunnen kosten met zich meebrengen.

Topologie

Voordat de configuratie begint, is het belangrijk om de topologie en het ontwerp volledig te begrijpen. Dit helpt om eventuele problemen later op te lossen.

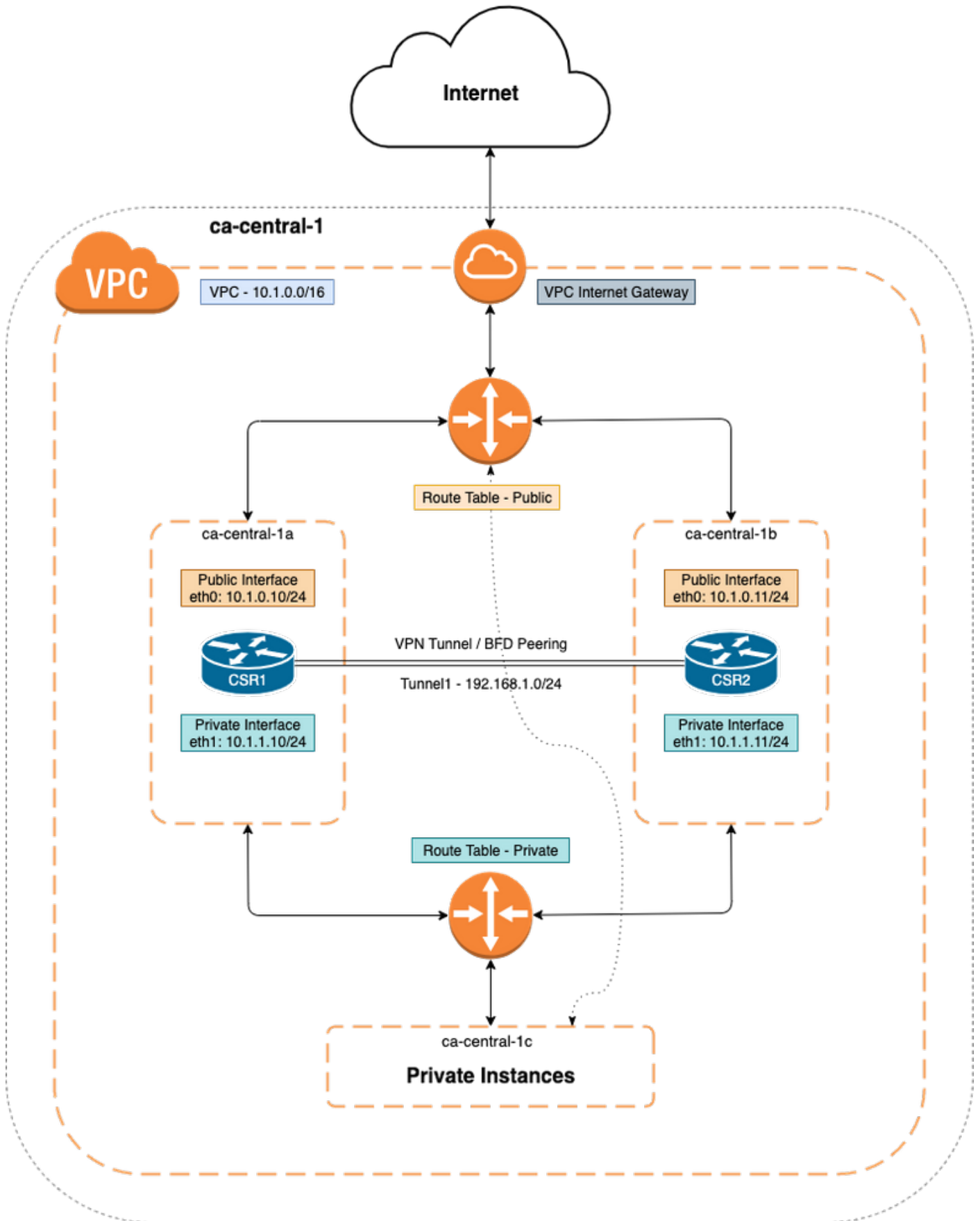
Hoewel het diagram van de netwerktopologie op AWS is gebaseerd, is de onderliggende netwerkimplementatie tussen wolken relatief gelijk. De netwerktopologie is ook onafhankelijk van de gebruikte HA-versie, of die nu HAv1, HAv2 of HAv3 is.

Voor dit topologievoorbeeld, wordt de overtolligheid HA met deze instellingen in AWS gevormd:

- 1x - regio
- 1x - VPC
- 3x - Beschikbare zones
- 4x - Netwerkinterfaces/subnetten (2x openbare gezichten/2x particuliere gezichten)
- 2x - Routetabellen (openbaar en particulier)
- 2x - CSR 1000v routers (Cisco IOS®-XE 17.01.01)

Er zijn 2x CSR1000v routers in een HA-paar, in twee verschillende beschikbaarheid zones. De derde zone is een privé-instantie, die een apparaat in een privé-datacenter simuleert. Over het algemeen moet al het normale verkeer door de privé (of binnenste) routetabel lopen.

Netwerkdigram



Netwerkdigram

CSR1000v routers configureren

Cloudfafhankelijke configuratie

Stap 1. Het configureren van IOX app-host en guestshell, biedt ip-bereikbaarheid in guestshell. Deze stap kan automatisch worden ingesteld wanneer CSR1000v wordt gedeactiveerd.

```
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
```

Stap 2. Schakel de kreuschelp in en inloggen.

```
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 ~]$
```

Opmerking: Zie voor meer informatie over de [krepelhouder](#) - [Programmakeuze voor configuratie](#)

Stap 3. Bevestig het bereik op het 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
```

Stap 4. (Optioneel) Schakel Bi-directional Forwarding Detection (BFD) en een routeringsprotocol in als Enhanced Interior Gateway Routing Protocol (DHCP) of Border Gateway Protocol (BGP) aan de tunnel voor detectie van peer-fouten. Configureer een VxLAN- of IPsec-tunnel tussen de Cisco CSR 1000v-routers.

- IPsec-tunnel tussen de Cisco CSR 1000v-routers.

```
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
```

- VxLAN-tunnel tussen de Cisco CSR 1000v-routers.

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

Stap 4.1. (Optioneel) Configureer RSP via tunnelinterfaces.

```
router eigrp 1 bfd interface Tunnel1 network 192.168.1.0 0.0.0.255
```

- Aangepaste scripts kunnen worden gebruikt om failover op te roepen, bijvoorbeeld:

```
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
```

AWS-specifieke configuratie

- AWS HA-parameters

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.

Stap 1. Verificatie met IAM configureren.

Om een router in CSR1000v bij te werken moet een routingtabel in het AWS-netwerk worden bijgewerkt, moet de router worden geauthentiseerd. In AWS, moet u een beleid creëren dat de router CSR 1000v om tot de routekaart toegang te hebben toestaat. Vervolgens wordt een IAM-rol gecreëerd die dit beleid gebruikt en van toepassing is op de EG2-middelen.

Nadat de CSR 1000v EC2 instanties zijn gecreëerd, moet de gemaakte IAM rol aan elke router worden bevestigd.

Het beleid dat in de nieuwe IAM-rol wordt gebruikt is:

```
{ "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": "*" } ] }
```

Opmerking: Raadpleeg de [IAM-rol met een beleid en associeer deze met de VPC](#) voor uitgebreide stappen.

Stap 2. Installeer de verpakking met de HA-python.

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

Stap 3. Configureer de HA-parameters op de primaire router.

```
[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
```

Stap 4. Configureer de HA-parameters op de secundaire router.

```
[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
```

- Knooppunt is:

```
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
```

evening Specific Configuration

- 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.

Opmerking: De buitenkant tegenoverliggende interface moet op Gigabit Ethernet1 zijn geconfigureerd. Dit is de interface die gebruikt wordt om de Kuurlappen te bereiken. HA kan anders niet goed werken. Binnen de guestshell, zorg ervoor dat de curl opdracht metagegevens kan halen van de burens.

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

Stap 1. Verificatie voor CSR1000v API-oproepen moet worden ingeschakeld met ofwel landbouw-actieve map (AAD) of beheerde Service Identity (MSI). Raadpleeg [Verificatie configureren voor CSR1000v API-oproepen](#) voor gedetailleerde stappen. Zonder deze stap, kan de router CSR1000v niet geautoriseerd worden om de routingtabel bij te werken.

AAD-parameters

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.

Stap 2. Installeer de verpakking met de HA-python.

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

Stap 3. Het configureren van HA-parameters op de primaire router (MSI of AAD kan voor deze stap worden gebruikt).

- Met MSI-verificatie.

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

- Met AAD-verificatie (extra - a, -d, -k vlaggen vereist).

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

Stap 4. Configureer de HA-parameters op de secundaire router.

- Met MSI-verificatie

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

- Met AAD-authenticatie (extra -a, -d, -k vlaggen vereist)

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

GCP specifieke configuratie

- GCP HA-parameters

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.

Opmerking: Zorg ervoor dat de servicekaart die is gekoppeld aan de CSR 1000v-routers ten minste een computingsysteem hebben.

Command or Action	Purpose																
Ensure that the service account associated with the CSR 1000v routers at least have a Compute Network Admin permission.	<p>Create service account</p> <p>1 Service account details — 2 Grant this service account access to project (optional) — 3 Grant users access to this service account (optional)</p> <p>Service account permissions (optional)</p> <p>Grant this service account access to project-avvyas so that it has permission to complete specific actions on the resources in your project. Learn more</p> <p>Select a role</p> <div style="border: 1px solid #ccc; padding: 5px;"> <p>Type to filter</p> <table border="0"> <tr><td>Cloud TPU</td><td>Compute Admin</td></tr> <tr><td>Cloud Trace</td><td>Compute Image User</td></tr> <tr><td>CodeLab API Keys</td><td>Compute Instance Admin (beta)</td></tr> <tr><td>Compute Engine</td><td>Compute Instance Admin (v1)</td></tr> <tr><td>Container Analysis</td><td>Compute Load Balancer Admin</td></tr> <tr><td>Custom</td><td>Compute Network Admin</td></tr> <tr><td>Dataflow</td><td>Compute Network User</td></tr> <tr><td></td><td>Compute Network Viewer</td></tr> </table> <p>MANAGE ROLES</p> </div> <p>Compute Network Admin Full control of Compute Engine networking resources.</p> <p>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.</p>	Cloud TPU	Compute Admin	Cloud Trace	Compute Image User	CodeLab API Keys	Compute Instance Admin (beta)	Compute Engine	Compute Instance Admin (v1)	Container Analysis	Compute Load Balancer Admin	Custom	Compute Network Admin	Dataflow	Compute Network User		Compute Network Viewer
Cloud TPU	Compute Admin																
Cloud Trace	Compute Image User																
CodeLab API Keys	Compute Instance Admin (beta)																
Compute Engine	Compute Instance Admin (v1)																
Container Analysis	Compute Load Balancer Admin																
Custom	Compute Network Admin																
Dataflow	Compute Network User																
	Compute Network Viewer																

369497

Stap 1. Installeer de verpakking met de HA-python.

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

Stap 2. Configureer de HA-parameters op de primaire router.

```
[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
```

Stap 3. Configureer de HA-parameters op de secundaire router.

```
[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
```

Verifiëren

Gebruik dit gedeelte om te bevestigen dat de configuratie correct werkt.

Stap 1. Start een failover met de vlag `Knoop_event.pyFail`.

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

Stap 2. Navigeer naar de Private Route Tabel van uw Cloud Provider, controleer of de route de volgende-hop naar het nieuwe IP-adres heeft bijgewerkt.

Problemen oplossen

Er is momenteel geen specifieke troubleshooting-informatie beschikbaar voor deze configuratie.

Gerelateerde informatie

- Gedetailleerde HAV3-configuratiestappen zijn gevonden in [Cisco CSR 1000v en Cisco ISRV-softwareconfiguratie Guide](#)
- De configuratie van de Koude HAV2 lijkt grotendeels op HAV3 met kleine verschillen in de pakketten van de pip installeert en IOS overtollige configuratie. Documentatie vindt u op [CSR1000v HA versie 2 Configuration Guide op Microsoft Outlook](#)
- Koude HAV1-configuratie met CLI is te vinden op [CSR1000v HA-implementatiegids voor redundantie bij Microsoft karwei, te vinden bij de Klantenservice Klantenservice Klantenservice: Klantenservice: Klantenservice: Klantenservice; Klankast:](#)
- De AWS HAV1-configuratie is te vinden op [CSR1000v HA-implementatiegids voor redundantie op Amazon AWS](#)
- [Technische ondersteuning en documentatie – Cisco Systems](#)