AWS, Azure 및 GCP에서 CSR1000v HA 버전 3 구성

목차

소개

이 문서에서는 AWS(Amazon Web Services), Microsoft Azure 및 GCP(Google Cloud Platform)에서 HAv3(High Availability Version 3)용 CSR1000v 라우터를 구성하는 단계에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- AWS, Azure 또는 GCP 클라우드.
- CSR1000v 라우터.
- Cisco IOS®-XE

이 문서에서는 기본 네트워크 컨피그레이션이 이미 완료되었다고 가정하고 HAv3 컨피그레이션에 초점을 맞춥니다.

전체 컨피그레이션 세부 정보는 Cisco CSR 1000v 및 Cisco ISRv 소프트웨어 컨피그레이션 가이드에서 확인할 수 있습니다.

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- AWS, Azure 또는 GCP 계정입니다.
- 2 CSR1000v 라우터
- 최소 Cisco IOS®-XE Polaris 16.11.1

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

Cisco에서는 다음과 같은 다양한 HA 버전에 대해 알고 있는 것이 좋습니다.

- HAv1:HA 컨피그레이션은 IOS 명령으로 수행되며 BFD를 장애 탐지 메커니즘으로 사용합니다.
- HAv2/HAv3:구현이 Python 스크립트로 게스트 셸 컨테이너로 이동되었습니다.BFD는 선택 사항이며 사용자 지정 스크립트를 작성하여 장애를 감지하고 장애 조치를 트리거할 수 있습니다.Azure HAv2 컨피그레이션은 pip 설치 패키지 및 IOS 이중화 컨피그레이션의 사소한 차이점으로 인해 HAv3과 크게 유사합니다.
- HAv3:HA의 구현은 대부분 Cisco IOS®-XE 코드에서 이동되었으며 게스트 셸 컨테이너에서 실행됩니다.

HAv3는 Cisco IOS®-XE Polaris 16.11.1에서 사용할 수 있으며 다음과 같은 몇 가지 새로운 기능을 추가합니다.

- **클라우드에 구애받지 않음:**이 고가용성 버전은 모든 클라우드 서비스 제공업체의 CSR 1000v 라우터에서 작동합니다.클라우드 용어 및 매개변수에는 몇 가지 차이점이 있지만, 고가용성 기능을 구성, 제어 및 표시하는 데 사용되는 기능 및 스크립트 세트는 서로 다른 클라우드 서비스 제공자에서 공통적입니다.HAv3(High Availability Version 3)은 AWS, Azure 및 GCP의 CSR 1000v 라우터에서 지원됩니다.GCP 공급자에 대한 지원이 16.11.1에 추가되었습니다. 개별 공급자의 클라우드에서 현재 고가용성을 지원하려면 Cisco에 문의하십시오.
- **활성/활성 작업:**두 Cisco CSR 1000v 라우터를 동시에 활성화하도록 구성하여 로드 공유를 가능하게 할 수 있습니다.이 작동 모드에서 경로 테이블의 각 경로에는 기본 라우터로 사용되는 두 라우터 중 하나와 보조 라우터로 사용되는 다른 라우터가 있습니다.로드 공유를 활성화하려면 모든 경로를 가져와 두 Cisco CSR 1000v 라우터 간에 분할합니다.이 기능은 AWS 기반 클라우드의 새로운 기능입니다.
- **장애 복구 후 기본 CSR로의 복귀:**Cisco CSR 1000v를 지정된 경로의 기본 라우터로 지정할 수 있습니다.이 Cisco CSR 1000v는 작동 중이지만 경로의 다음 홉입니다.이 Cisco CSR 1000v에 장애가 발생하면 피어 Cisco CSR 1000v가 해당 라우트의 다음 홉으로 인수되어 네트워크 연결을 유지합니다.원래 라우터가 실패에서 복구되면 경로의 소유권을 회수하고 다음 hop 라우터입니다.이 기능은 AWS 기반 클라우드에서도 새로운 기능입니다.
- **사용자 제공 스크립트:**게스트 셸은 자체 스크립트를 배포할 수 있는 컨테이너입니다.HAv3는 사용자가 제공한 스크립트에 프로그래밍 인터페이스를 표시합니다.따라서 이제 페일오버 및 복귀 이벤트를 모두 트리거할 수 있는 스크립트를 작성할 수 있습니다.특정 경로에 대해 어떤 Cisco CSR 1000v가 포워딩 서비스를 제공하는지 제어하는 자체 알고리즘 및 트리거를 개발할 수도 있습니다.이 기능은 AWS 기반 클라우드에 대한 새로운 기능입니다.
- **새로운 구성 및 구축 메커니즘:**HA 구현이 Cisco IOS®-XE 코드에서 제거되었습니다.이제 고가용성 코드가 게스트 셸 컨테이너에서 실행됩니다.게스트 셸에 대한 자세한 내용은 Programmability Configuration Guide의 "Guest Shell" 섹션을 참조하십시오.HAv3에서 이중화 노드의 컨피그레이션은 Python 스크립트 세트를 사용하는 게스트 셸에서 수행됩니다.이 기능은 이제 AWS 기반 클라우드용으로 도입되었습니다.

참고:이 문서의 단계에서 AWS, Azure 또는 GCP에 배포된 리소스는 비용을 초래할 수 있습니다.

토폴로지

구성을 시작하기 전에 토폴로지와 설계를 완전히 이해하는 것이 중요합니다.이는 나중에 발생할 수 있는 문제를 해결하는 데 도움이 됩니다.

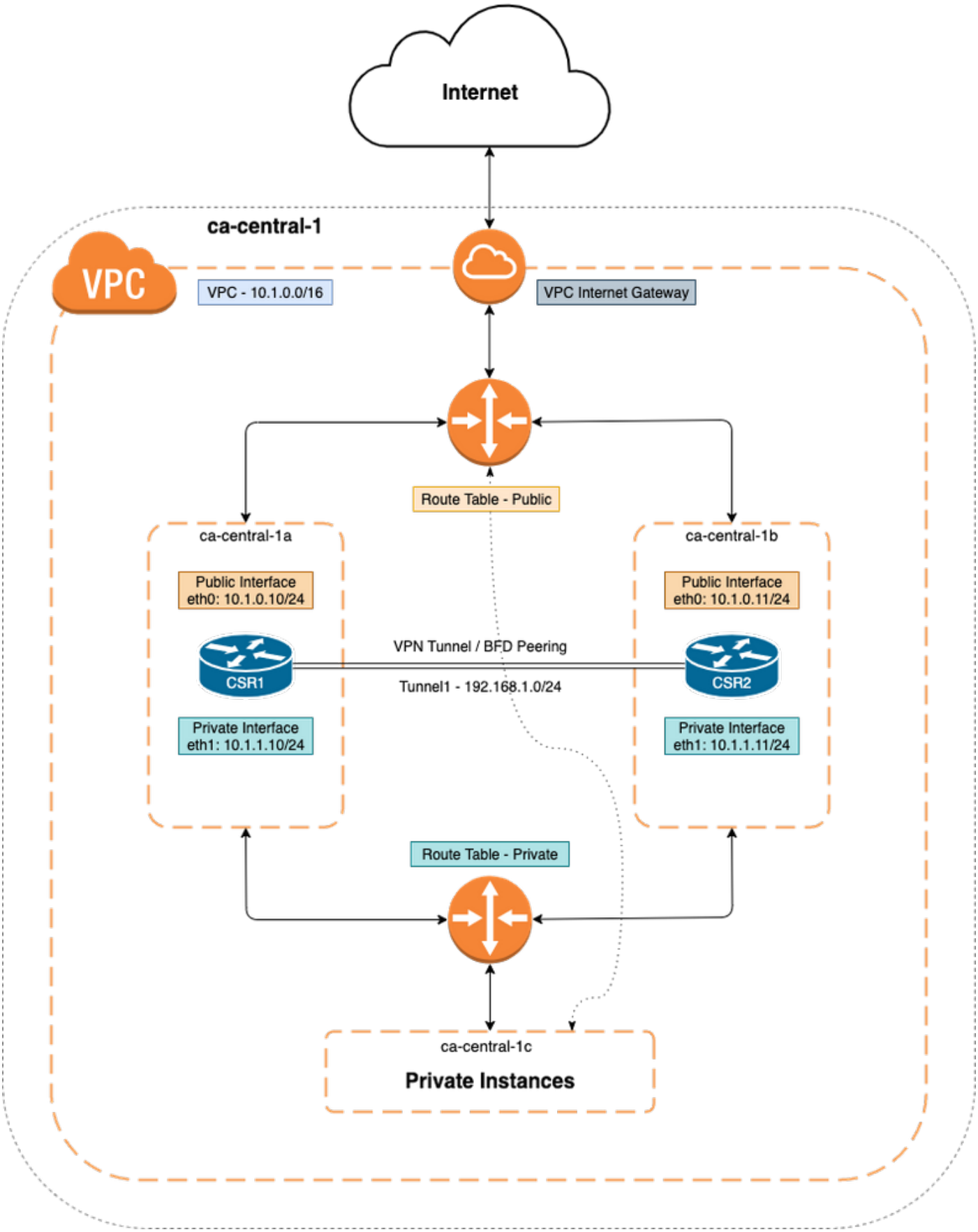
네트워크 토폴로지 다이어그램은 AWS를 기반으로 하지만 클라우드 간의 기본 네트워크 구축은 상대적으로 유사합니다. 또한 네트워크 토폴로지는 HAv1, HAv2 또는 HAv3에 관계없이 사용되는 HA 버전과는 독립적입니다.

이 토폴로지 예제에서는 HA 이중화가 AWS에서 다음 설정으로 구성됩니다.

- 1x - 지역
- 1x - VPC
- 3x - 가용 영역
- 4x - 네트워크 인터페이스/서브넷(공용 접지 2개/전용 접지 2개)
- 2x - 경로 테이블(공용 및 사설)
- 2x - CSR1000v 라우터(Cisco IOS®-XE 17.01.01)

HA 쌍에는 2개의 다른 가용 영역에 2x CSR1000v 라우터가 있습니다.세 번째 영역은 프라이빗 데이터 센터의 디바이스를 시뮬레이션하는 프라이빗 인스턴스입니다.일반적으로 모든 일반 트래픽은 프라이빗(또는 내부) 경로 테이블을 통과해야 합니다.

네트워크 다이어그램



네트워크 다이어그램

CSR1000v 라우터 구성

클라우드 독립적인 구성

1단계. IOX 앱 호스팅 및 게스트 셸을 구성하면 게스트 셸에 IP 연결성을 제공합니다. 이 단계는 CSR1000v를 구축할 때 기본적으로 자동으로 구성할 수 있습니다.

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

2단계. 게스트 셸을 활성화하고 로그인합니다.

```
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 ~]$
```

> **참고:**게스트 셸에 대한 자세한 내용은 - 프로그래밍 기능 컨피그레이션 가이드를 참조하십시오.

3단계. 게스트셸이 인터넷에 통신할 수 있는지 확인합니다.

```
[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
```

4단계. (선택 사항) 피어 장애 탐지를 위해 터널에 BFD(Bi-Directional Forwarding Detection) 및 라우팅 프로토콜을 EIGRP(Enhanced Interior Gateway Routing Protocol) 또는 BGP(Border Gateway Protocol)로 활성화합니다.Cisco CSR 1000v 라우터 간에 VxLAN 또는 IPsec 터널을 구성합니다.

- Cisco CSR 1000v 라우터 간 IPsec 터널

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

- Cisco CSR 1000v 라우터 간의 VxLAN 터널

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

4.1단계. (선택 사항) 터널 인터페이스를 통해 EIGRP를 구성합니다.

router eigrp 1 bfd interface Tunnel1 network 192.168.1.0 0.0.0.255

- 사용자 지정 스크립트를 사용하여 장애 조치를 트리거할 수 있습니다. 예:

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 특정 구성

- AWS HA 매개변수

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.

1단계. IAM으로 인증을 구성합니다.

CSR1000v 라우터가 AWS 네트워크에서 라우팅 테이블을 업데이트하려면 라우터를 인증해야 합니다.AWS에서는 CSR 1000v 라우터가 경로 테이블에 액세스하도록 허용하는 정책을 생성해야 합니다.그런 다음 이 정책을 사용하는 IAM 역할이 생성되어 EC2 리소스에 적용됩니다.

CSR 1000v EC2 인스턴스가 생성된 후 생성된 IAM 역할을 각 라우터에 연결해야 합니다.

새 IAM 역할에 사용되는 정책은 다음과 같습니다.

{ "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": "*" }] }

참고:자세한 단계는 IAM 역할을 Policy와 함께 참조하여 VPC에 연결합니다.

2단계. HA Python 패키지를 설치합니다.

[guestshell@guestshell ~]$ **pip install csr_aws_ha --user**
[guestshell@guestshell ~]$ **source ~/.bashrc**
3단계. 기본 라우터에서 HA 매개변수를 구성합니다.

[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
4단계. 보조 라우터에서 HA 매개변수를 구성합니다.

[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

- 노드 형식:

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

Azure 특정 구성

- Azure HA 매개 변수
  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.

참고:외부 연결 인터페이스는 GigabitEthernet1에서 구성해야 합니다. Azure API에 연결하는 데 사용되는 인터페이스입니다.그렇지 않으면 HA가 제대로 작동하지 않습니다.게스트 셸 내에서 curl 명령이 Azure에서 메타데이터를 가져올 수 있는지 확인합니다.

[guestshell@guestshell ~]$ curl -H "Metadata:true" http://169.254.169.254/metadata/instance?api-version=2020-06-01
1단계. CSR1000v API 호출에 대한 인증은 Azure AAD(Active Directory) 또는 MSI(Managed Service Identity)로 사용하도록 설정해야 합니다. 자세한 단계는 CSR1000v API 호출용 인증 구성을 참조하십시오. 이 단계를 수행하지 않으면 CSR1000v 라우터에 경로 테이블을 업데이트할 권한이 없습니다.

AAD 매개변수

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.

2단계. HA Python 패키지를 설치합니다.

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

3단계. 기본 라우터에서 HA 매개변수를 구성합니다(이 단계에서 MSI 또는 AAD를 사용할 수 있음).

- MSI 인증을 사용합니다.

```
[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
```

- AAD 인증 사용(추가 -a, -d, -k 플래그 필요)

```
[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=
```

4단계. 보조 라우터에서 HA 매개변수를 구성합니다.

- MSI 인증

```
[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
```

- AAD 인증 사용(추가 -a, -d, -k 플래그 필요)

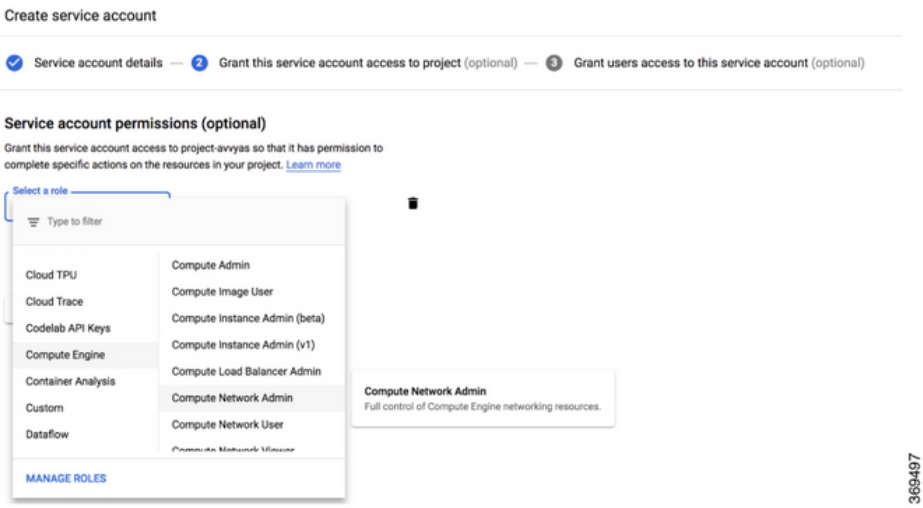
```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx --g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.0.0.11 -m secondary -a 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxx -d ae49849c-2622-4d45-b95e-xxxxxxxxx -k bDEN1k8batJqpeqjAuUvaUCZn5Md6rWEi=
```

GCP 특정 컨피그레이션

- GCP HA 매개변수

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.

참고:CSR 1000v 라우터와 연결된 서비스 계정에 적어도 컴퓨팅 네트워크 관리 권한이 있어야 합니다.

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.

1단계. HA Python 패키지를 설치합니다.

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

2단계. 기본 라우터에서 HA 매개변수를 구성합니다.

[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

3단계. 보조 라우터에서 HA 매개변수를 구성합니다.

[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

다음을 확인합니다.

이 섹션을 사용하여 컨피그레이션이 제대로 작동하는지 확인합니다.

1단계. node_event.py peerFail 플래그를 사용하여 장애 조치를 트리거합니다.

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

2단계. 클라우드 공급자의 Private Route Table(프라이빗 경로 테이블)로 이동하여 경로가 새 IP 주소로 next-hop을 업데이트했는지 확인합니다.

문제 해결

현재 이 컨피그레이션에 사용할 수 있는 특정 문제 해결 정보가 없습니다.

관련 정보

- 자세한 HAv3 컨피그레이션 단계는 Cisco CSR 1000v 및 Cisco ISRv 소프트웨어 컨피그레이션 가이드에서 찾을 수 있습니다.
- Azure HAv2 컨피그레이션은 pip 설치 패키지 및 IOS 이중화 컨피그레이션의 사소한 차이점으로 인해 HAv3과 크게 유사합니다. 문서는 Microsoft Azure의 CSR1000v HA 버전 2 구성 가이드에서 찾을 수 있습니다.
- CLI를 사용하는 Azure HAv1 구성은 AzureCLI 2.0을 사용하는 Microsoft Azure의 CSR1000v HA 이중화 배포 가이드에서 찾을 수 있습니다.
- AWS HAv1 컨피그레이션은 Amazon AWS의 CSR1000v HA 리던던시 구축 가이드에서 찾을 수 있습니다.
- 기술 지원 및 문서 – Cisco Systems