AWS、Azure、およびGCPでのCSR1000v HAバージョン3の設定

内容

概要

このドキュメントでは、Amazon Web Services(AWS)、Microsoft Azure、およびGoogle Cloud Platform(GCP)でハイアベイラビリティバージョン3(HAv3)用のCSR1000vルータを設定する手順について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- AWS、Azure、またはGCPクラウド
- CSR1000vルータ
- Cisco IOS®-XE

この記事では、基盤となるネットワーク設定がすでに完了しており、HAv3設定に焦点を当てていることを前提としています。

設定の詳細については、『Cisco CSR 1000v and Cisco ISRv Software Configuration Guide』を参照してください。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- AWS、Azure、またはGCPアカウント。
- CSR1000vルータ2台
- Cisco IOS®-XE Polaris 16.11.1以上

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。ネットワークが稼働中の場合は、コマンドが及ぼす潜在的な影響について確実に理解しておく必要があります。

背景説明

次のHAバージョンに関する知識があることが推奨されます。

- HAv1:HA設定はIOSコマンドとして実行され、障害を検出するメカニズムとしてBFDに依存します。
- HAv2/HAv3:実装は、Pythonスクリプトとしてguestshellコンテナに移動されました。BFDはオプションであり、障害を検出してフェールオーバーをトリガーするためのカスタムスクリプトを記述できます。Azure HAv2の構成は、主にHAv3に似ており、pipインストールパッケージとIOS冗長構成のマイナーな違いがあります。
- HAv3:HAの実装は、主にCisco IOS®-XEコードから削除され、guestshellコンテナで実行されています。

HAv3はCisco IOS®-XE Polaris 16.11.1から入手でき、次の新機能が追加されています。

- **クラウド非依存**：このバージョンのハイアベイラビリティは、任意のクラウドサービスプロバイダーのCSR 1000vルータで機能します。クラウドの用語とパラメータには若干の違いがありますが、ハイアベイラビリティ機能の設定、制御、および表示に使用される一連の機能とスクリプトは、さまざまなクラウドサービスプロバイダーで共通です。ハイアベイラビリティバージョン3(HAv3)は、AWS、Azure、およびGCPのCSR 1000vルータでサポートされています。GCPプロバイダーのサポートは16.11.1で追加されました。各プロバイダーのクラウドにおけるハイアベイラビリティの現在のサポートについては、シスコにお問い合わせください。
- **アクティブ/アクティブ操作**：両方のCisco CSR 1000vルータを同時にアクティブに設定できるため、ロードシェアリングが可能です。この動作モードでは、ルートテーブル内の各ルートに、プライマリルータとして機能する2台のルータと、セカンダリルータとして機能する他のルータがあります。ロードシェアリングを有効にするには、すべてのルートを使用して、2台のCisco CSR 1000vルータ間で分割します。この機能は、AWSベースのクラウドで新たに追加されました。
- **障害回復後のプライマリCSRへの復帰**：Cisco CSR 1000vを特定のルートのプライマリルータとして指定できます。このCisco CSR 1000vはアップ状態です。ルートのネクストホップです。このCisco CSR 1000vに障害が発生すると、ピアCisco CSR 1000vがルートのネクストホップとして引き継ぎ、ネットワーク接続が維持されます。元のルータが障害から回復すると、そのルータはルートの所有権を再要求し、ネクストホップルータになります。この機能は、AWSベースのクラウドにも新しく追加されました。
- **ユーザ指定のスクリプト**：guestshellは、独自のスクリプトを展開できるコンテナです。HAv3では、プログラミングインターフェイスがユーザ指定のスクリプトに公開されます。これは、フェールオーバーと復帰の両方のイベントをトリガーするスクリプトを記述できることを意味します。また、独自のアルゴリズムとトリガーを開発して、特定のルートに対して

転送サービスを提供するCisco CSR 1000vを制御することもできます。この機能は、AWSベースのクラウドで新たに追加されました。
- **新しい設定および導入メカニズム**：HAの実装は、Cisco IOS®-XEコードから移行されました。現在、高可用性コードはguestshellコンテナで実行されます。guestshellの詳細については、『プログラマビリティ設定ガイド』の「ゲストシェル」セクションを参照してください。HAv3では、冗長ノードの設定は、一連のPythonスクリプトを使用するguestshellで実行されます。この機能は、AWSベースのクラウドに導入されました。

  注：AWS、Azure、またはGCPにデプロイされたリソースは、このドキュメントの手順に従うとコストがかかることがあります。

トポロジ

設定を開始する前に、トポロジと設計を完全に理解することが重要です。これは、今後発生する可能性のある問題のトラブルシューティングに役立ちます。

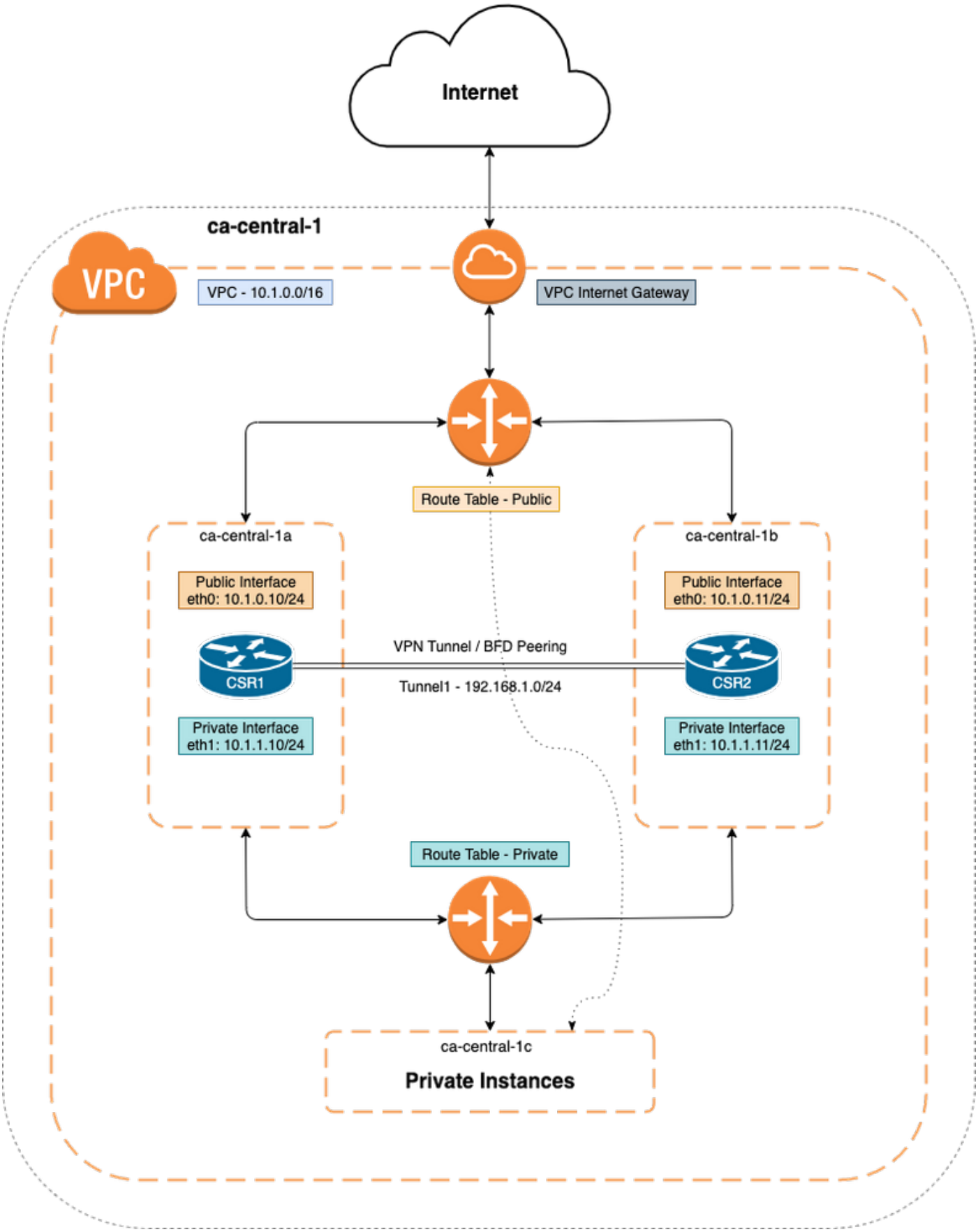
ネットワークトポロジ図はAWSに基づいていますが、クラウド間の基盤となるネットワークの展開は比較的類似しています。 ネットワークトポロジは、HAv1、HAv2、またはHAv3のいずれであっても、使用されるHAバージョンに依存しません。

このトポロジ例では、AWSでHA冗長性を次の設定で設定します。

- 1x – 地域
- 1x - VPC
- 3x：可用性ゾーン
- 4x – ネットワークインターフェイス/サブネット（パブリック向き2x、プライベート向き2x）
- 2x – ルートテーブル（パブリックおよびプライベート）
- 2x - CSR1000vルータ(Cisco IOS®-XE 17.01.01)

HAペアには2台のCSR1000vルータがあり、2つの異なるアベイラビリティゾーンにあります。3番目のゾーンはプライベートインスタンスで、プライベートデータセンター内のデバイスをシミュレートします。通常、すべての通常のトラフィックは、プライベート（または内部）ルートテーブルを通過する必要があります。

ネットワーク図



ネットワーク図

CSR1000vルータの設定

クラウドに依存しない設定

ステップ1:IOXアプリケーションホスティングとguestshellを設定します。これにより、guestshellへのIP到達可能性が提供されます。 この手順は、CSR1000vのデポ時にデフォルトで自動的に設定できます。

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

ステップ2：有効にして、guestshellにログインします。

```
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 ~]$
```

注：guestshellの詳細については、 - Programmability Configuration Guide

ステップ3:guestshellがインターネットと通信できることを確認します。

```
[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
```

ステップ4: (オプション) ピア障害検出のために、トンネルに対して双方向フォワーディング検出(BFD)とルーティングプロトコルをEnhanced Interior Gateway Routing Protocol(EIGRP)またはボーダーゲートウェイプロトコル(BGP)として有効にします。Cisco CSR 1000vルータ間にVxLANトンネルまたはIPsecトンネルを設定します。

- Cisco CSR 1000vルータ間のIPsecトンネル。

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ルータ間のVxLANトンネル

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

ステップ 4.1： （オプション）トンネルインターフェイス上でEIGRPを設定します。

router eigrp 1 bfd interface Tunnel1 network 192.168.1.0 0.0.0.255

- カスタムスクリプトを使用して、フェールオーバーをトリガーできます。次に例を示します。

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固有の設定

- AWS HAパラメータ

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.

ステップ1:IAMで認証を設定します。

CSR1000vルータがAWSネットワークのルーティングテーブルを更新するには、ルータを認証する必要があります。AWSでは、CSR 1000vルータにルートテーブルへのアクセスを許可するポリシーを作成する必要があります。その後、このポリシーを使用してEC2リソースに適用されるIAMロールが作成されます。

CSR 1000v EC2インスタンスを作成した後、作成したIAMロールを各ルータに関連付ける必要があります。

新しいIAMロールで使用されるポリシーは次のとおりです。

{ "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": "*" }] }

注：詳細な手順については、ポリシーを持つIAMロールを参照し、VPCに関連付けてくださ

い。

ステップ2:HA Pythonパッケージをインストールします。

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

ステップ3：プライマリルータでHAパラメータを設定します。

```
[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
```

ステップ4：セカンダリルータでHAパラメータを設定します。

```
[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
```

- ノードの形式：

```
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固有の構成

- Azure HAパラメーター

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.

注：GigabitEthernet1にOutside側インターフェイスを設定する必要があります。 これは、Azure APIに到達するために使用されるインターフェイスです。それ以外の場合、HAは正しく機能しません。guestshell内で、curlコマンドがAzureからメタデータを取得できることを確認します。

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

ステップ1:CSR1000v API呼び出しの認証は、Azure Active Directory (AAD)またはマネージドサービスID (MSI)のいずれかで有効にする必要があります。 詳細な手順[については、「CSR1000v APIコールの認証の設定」](#)を参照してください。 この手順がないと、CSR1000vルータはルートテーブルの更新を許可されません。

AADパラメータ

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.

ステップ2:HA Pythonパッケージをインストールします。

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

ステップ3：プライマリルータでHAパラメータを設定します（このステップではMSIまたはAADを使用できます）。

- MSI認証。

```
[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認証（追加の – a、-d、-kフラグが必要）

```
[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=
```

ステップ4：セカンダリルータでHAパラメータを設定します。

- MSI認証

```
[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認証（追加の – a、-d、-kフラグが必要）

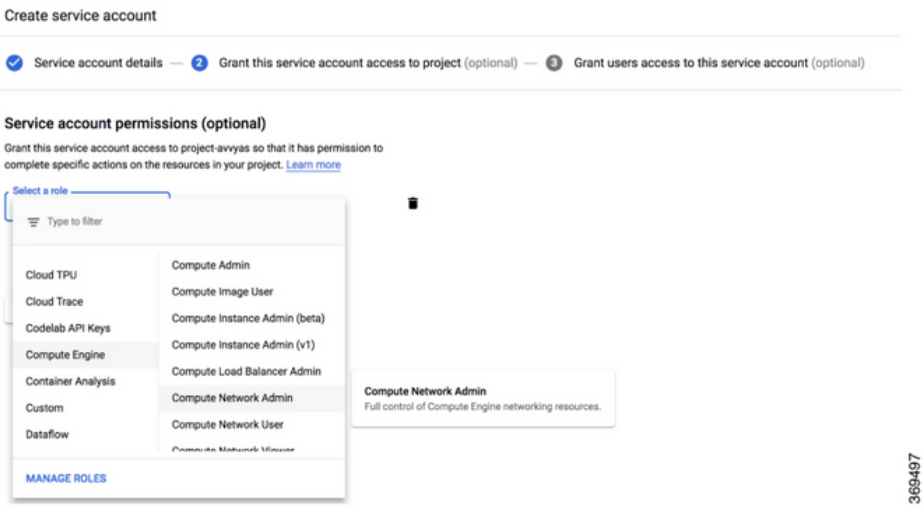
```
[guestshell@guestshell ~]$ create_node -i 10 -p azure -s xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx --g ResourceGroup -t Private-RouteTable -r 0.0.0.0/0 -n 10.0.0.11 -m secondary -a 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxx -d ae49849c-2622-4d45-b95e-xxxxxxxxx -k bDEN1k8batJqpeqjAuUvaUCZn5Md6rWEi=
```

GCP固有の設定

- GCP HAパラメータ

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.

注：CSR 1000vルータに関連付けられているサービスアカウントに、少なくともコンピューティングネットワーク管理者権限があることを確認します。

Command or Action	Purpose
Ensure that the service account associated with the CSR 1000v routers at least have a Compute Network Admin permission.	

ステップ1:HA Pythonパッケージをインストールします。

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

ステップ2：プライマリルータでHAパラメータを設定します。

[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

ステップ3：セカンダリルータでHAパラメータを設定します。

[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

確認

ここでは、設定が正常に機能しているかどうかを確認します。

ステップ1:node_event.py peerFailフラグを使用してフェールオーバーをトリガーします。

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

ステップ2：クラウドプロバイダーの[Private Route Table]に移動し、ルートが新しいIPアドレスへのネクストホップを更新したことを確認します。

トラブルシュート

現在、この設定に関する特定のトラブルシューティング情報はありません。

関連情報

- HAv3設定手順の詳細については、『Cisco CSR 1000v and Cisco ISRv Software Configuration Guide』を参照してください
- Azure HAv2の構成は、主にHAv3に似ており、pipインストールパッケージとIOS冗長構成のマイナーな違いがあります。 ドキュメントについては、Microsoft Azureの『CSR1000v HA Version 2 Configuration Guide』を参照してください
- CLIを使用したAzure HAv1の構成については、Azure CLI 2.0を使用したMicrosoft Azureの『CSR1000v HA Redundancy Deployment Guide』を参照してください
- AWS HAv1の設定については、Amazon AWSのCSR1000v HA冗長導入ガイドを参照してください
- テクニカル サポートとドキュメント – Cisco Systems