Amazon AWSでのCSR1000v HA冗長性デプロイガイド

内容

概要

このドキュメントでは、Amazon AWSクラウドでハイアベイラビリティを実現するためにCSR1000vルータをデプロイする方法に関する設定ガイドについて説明します。これは、ユーザにHAに関する実用的な知識を提供し、完全に機能するテストベッドを導入できるようにすることを目的としています。

AWSとHAの詳細な背景については、セクションを*参照*してください。

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。

- Amazon AWSアカウント
- 2つのCSR1000vおよび1つのLinux/Windows AMIを同じリージョンに配置
- HAバージョン1は、Cisco IOS-XE®バージョン16.5 ~ 16.9でサポートされます。 16.11以降では、HAバージョン3を使用します。

使用するコンポーネント

このドキュメントの情報は、Cisco IOS-XE® Denali 16.7.1に基づくものです。

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。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

目標

複数のアベイラビリティーゾーン環境で、プライベートデータセンター(VM)からインターネットへの連続トラフィックをシミュレートします。HAフェールオーバーをシミュレートし、ルーティングテーブルがトラフィックをCSRHAからCSRHA1のプライベートインターフェイスに切り替えたため、HAが成功することを確認します。

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、後で潜在的な問題をトラブルシューティングするのに役立ちます。

ネットワーク要件に基づいて、HA導入のさまざまなシナリオがあります。この例では、HA冗長性は次の設定で設定されています。

- 1x – 地域
- 1x - VPC
- 3x – アベイラビリティーゾーン
- 6x – ネットワークインターフェイス/サブネット（3xパブリック側/3xプライベート側）
- 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® Denali 16.3.1a以降)
- 1x:VM(Linux/Windows)

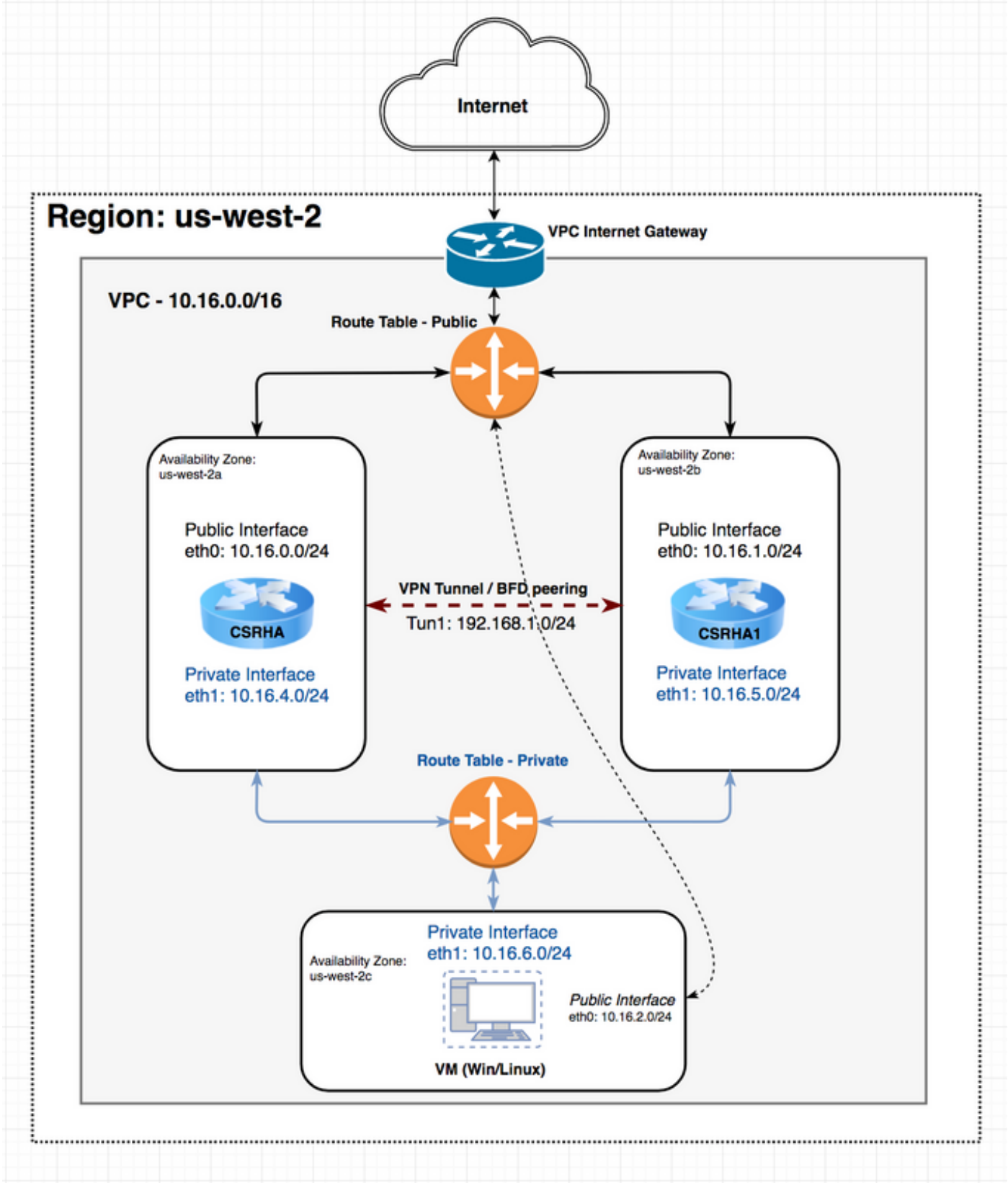
1つのHAペアに2台のCSR1000vルータがあり、2つの異なるアベイラビリティーゾーンに存在します。ハードウェアの復元力を高めるために、各可用性ゾーンを個別のデータセンターと考えてください。

3番目のゾーンはVM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。現時点では、インターネットアクセスはパブリックインターフェイスを介して有効になっているため

、VMにアクセスして設定できます。通常、すべての通常のトラフィックはプライベートルートテーブルを通過する必要があります。

トラフィックシミュレーションのため、CSRHA→8.8.8.8を使用→てVMの→ライベートインターフェイスとプライベートルートテーブルにpingを実行します。フェールオーバーシナリオでは、プライベートルートテーブルがCSRHA1のプライベートインターフェイスを指すようにルートを切り替えたことを確認します。

ネットワーク図



用語

RTB：ルートテーブルID。

CIDR：ルートテーブルで更新されるルートの宛先アドレス。

ENI：トラフィックがルーティングされるCSR 1000vギガビットインターフェイスのネットワークインターフェイスID。

たとえば、CSRHAが失敗すると、CSRHA1がAWSルートテーブル内のルートを引き継ぎ、自身のENIを指すように更新します。

REGION:CSR 1000vのAWSリージョン。

制約事項

- プライベートサブネットの場合は、IPアドレス10.0.3.0/24を使用しないでください。これは、ハイアベイラビリティのためにCisco CSR 1000vの内部で使用されます。Cisco CSR 1000vでは、AWSルートテーブルを変更するREST API呼び出しを行うために、パブリックインターネットアクセシビリティが必要です。

- CSR1000vのgig1インターフェイスをVRF内に配置しないでください。 HAはそれ以外では動作しません。

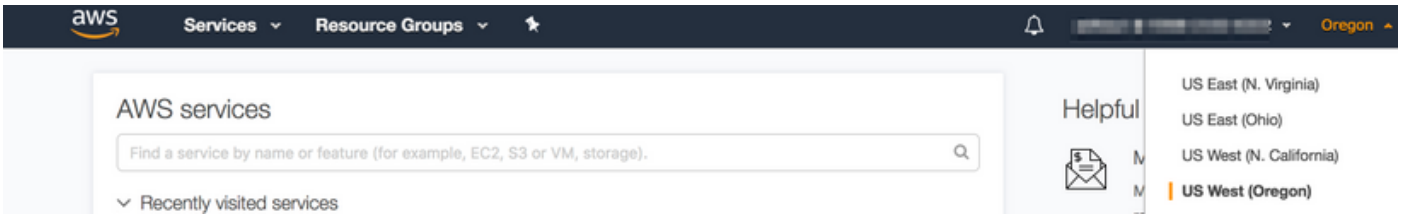
コンフィギュレーション

設定の一般的なフローは、最も包括的な機能（リージョン/VPC）から開始し、最も具体的な機能（インターフェイス/サブネット）まで下っていきます。 ただし、特定の設定順序はありません。開始する前に、まずトポロジを理解することが重要です（トポロジの説明を参照）。

> **ヒント**：すべての設定（VPC、インターフェイス、サブネット、ルートテーブルなど）に名前を付けます。

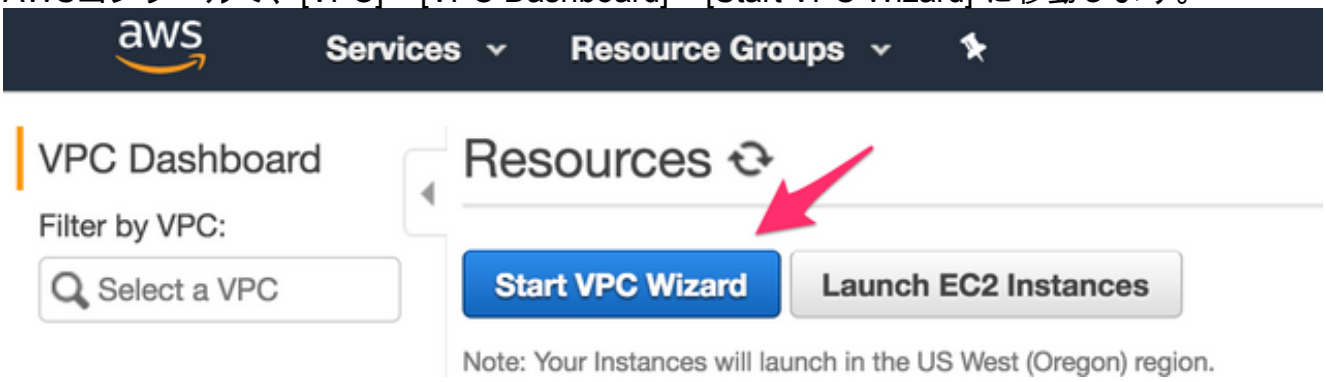
ステップ1：地域を選択します。

この例では、米国西部（オレゴン）を使用します。

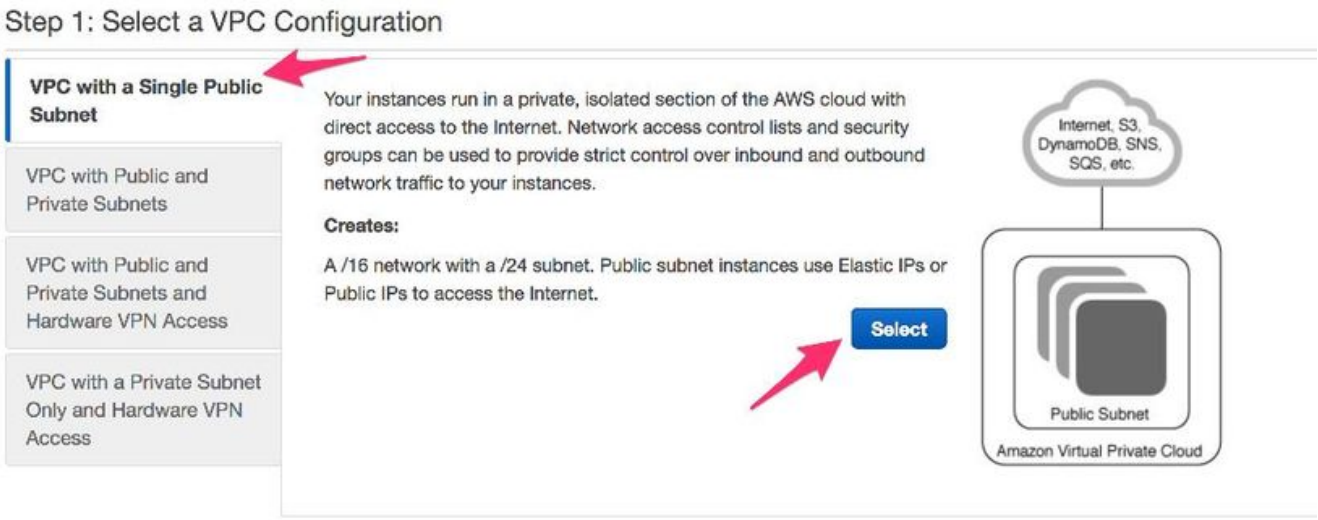


ステップ2:VPCを作成します。

1. AWSコンソールで、[VPC] > [VPC Dashboard] > [Start VPC Wizard] に移動します。

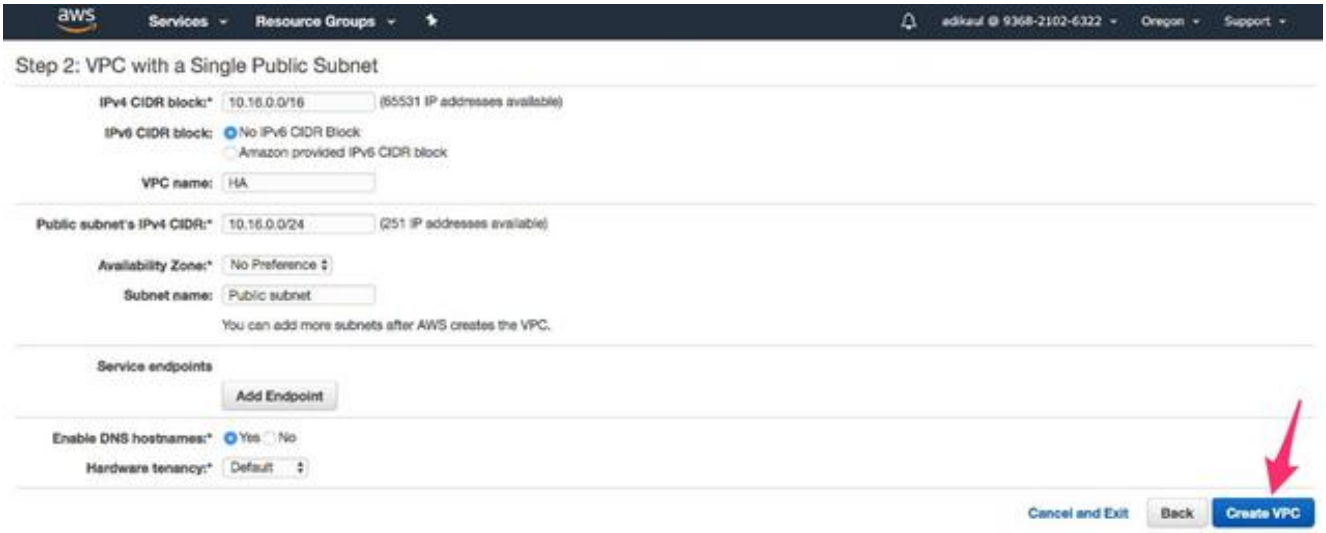


2. [VPC with a Single Public Subnet]を選択します。

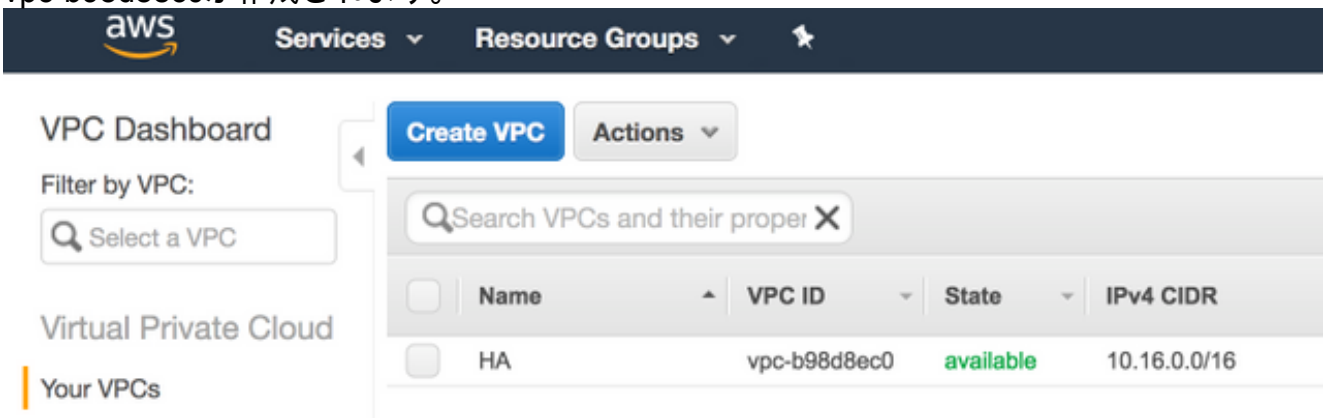


3. VPCを作成すると、自由に使用する/16ネットワークが割り当てられます。

4. また、/24パブリックサブネットが割り当てられています。パブリックサブネットインスタンスは、デバイスがインターネットにアクセスするためにElastic IPまたはパブリックIPを使用します。



5. vpc-b98d8ec0が作成されます。



ステップ3:VPCのセキュリティグループを作成します。

セキュリティグループは、トラフィックを許可または拒否するACLのようなものです。

1. Securityの下で**Security Groups**をクリックし、上記で作成したHAという名前のVPCに関連付けられた**Create your Security Group**をクリックします。

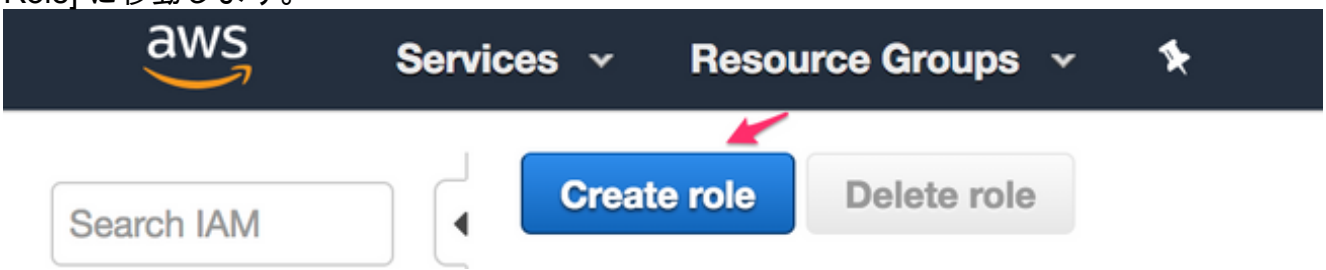
2. [Inbound Rules]で、sg-1cf47d6dに許可するトラフィックを定義します。この例では、[All Traffic]を許可します。

ステップ4：ポリシーを使用してIAMロールを作成し、VPCに関連付けます。

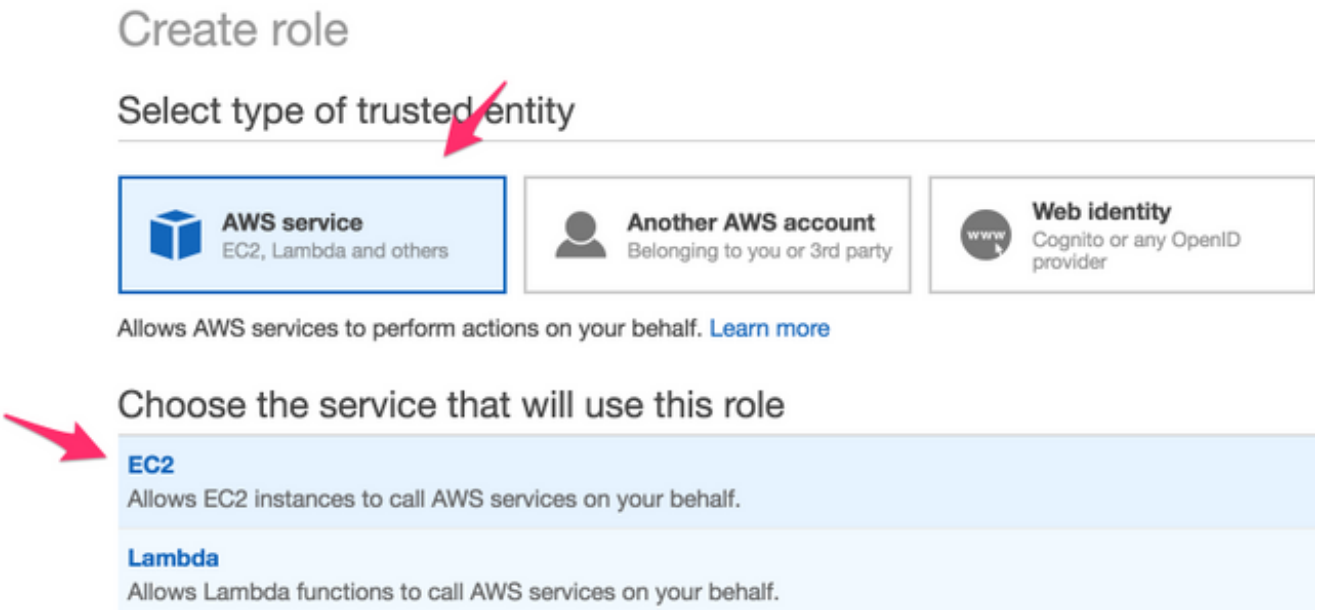
IAMはAmazon APIへのCSRアクセスを許可します。

CSR1000vは、AWS APIコマンドを呼び出してルートテーブルを変更するためのプロキシとして使用されます。デフォルトでは、AMIはAPIにアクセスできません。この手順ではIAMロールを作成し、このロールはCSRインスタンスの起動時に使用されます。IAMは、CSRがAWS APIを使用および変更するためのアクセス認証情報を提供します。

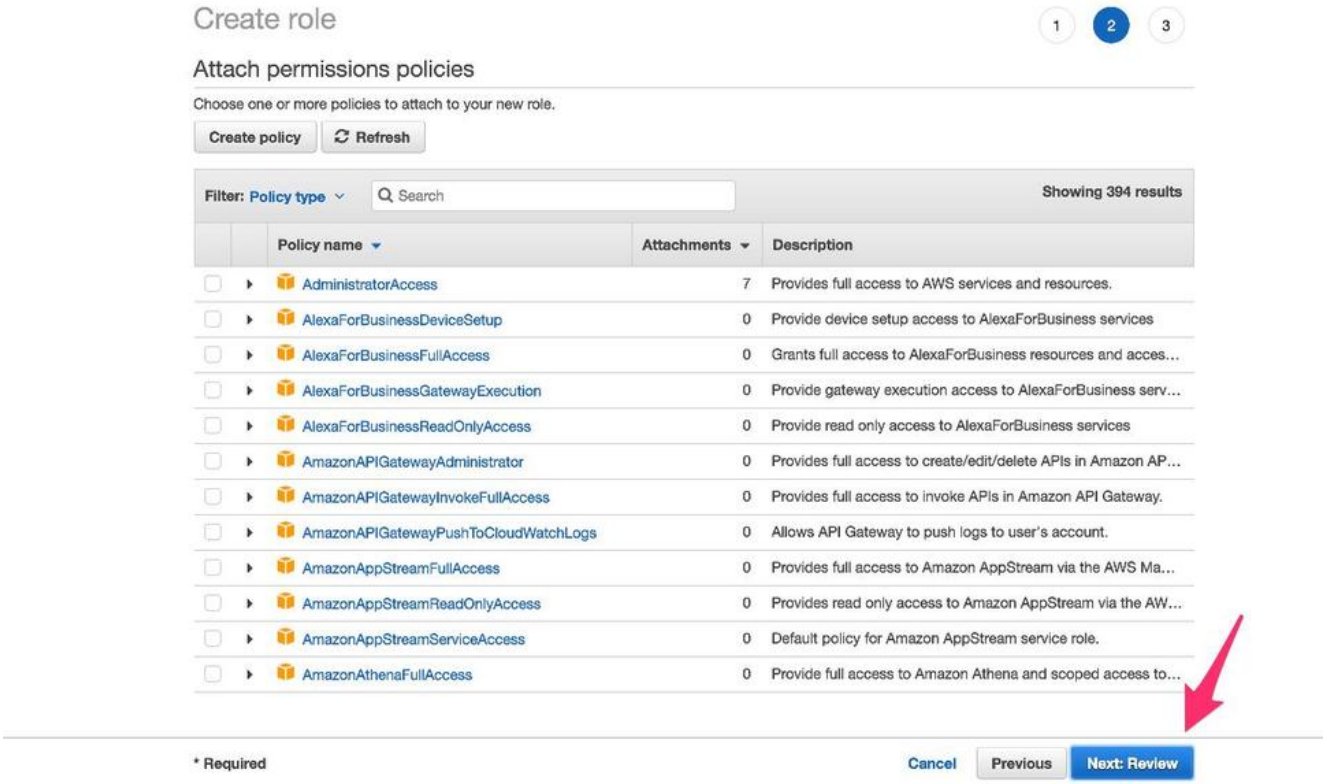
1. IAMロールを作成します。図に示すように、IAMダッシュボードを参照し、[Roles] > [Create Role] に移動します。



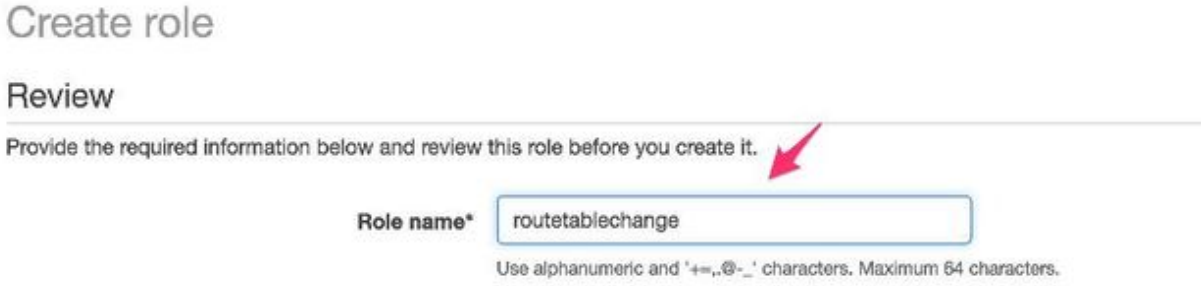
2. 図に示すように、代わりにEC2インスタンスがAWSを呼び出すことを許可します。



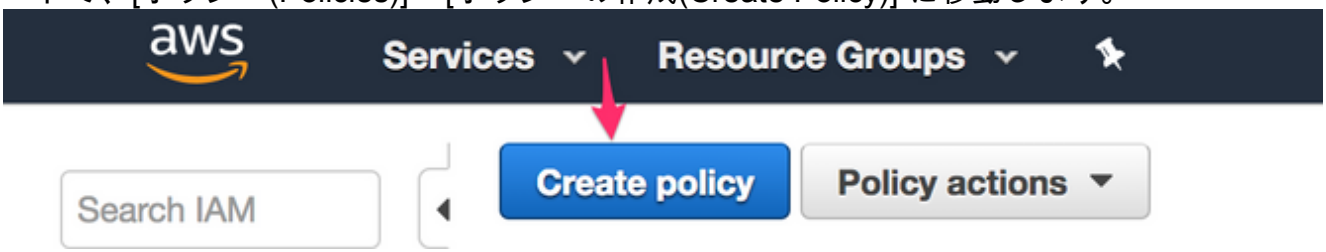
3. ロールを作成し、[Next] をクリックします。図に示すように、確認します。



4. ロール名を指定します。この例では、図に示すように、ロール名はroutetablechangeです。

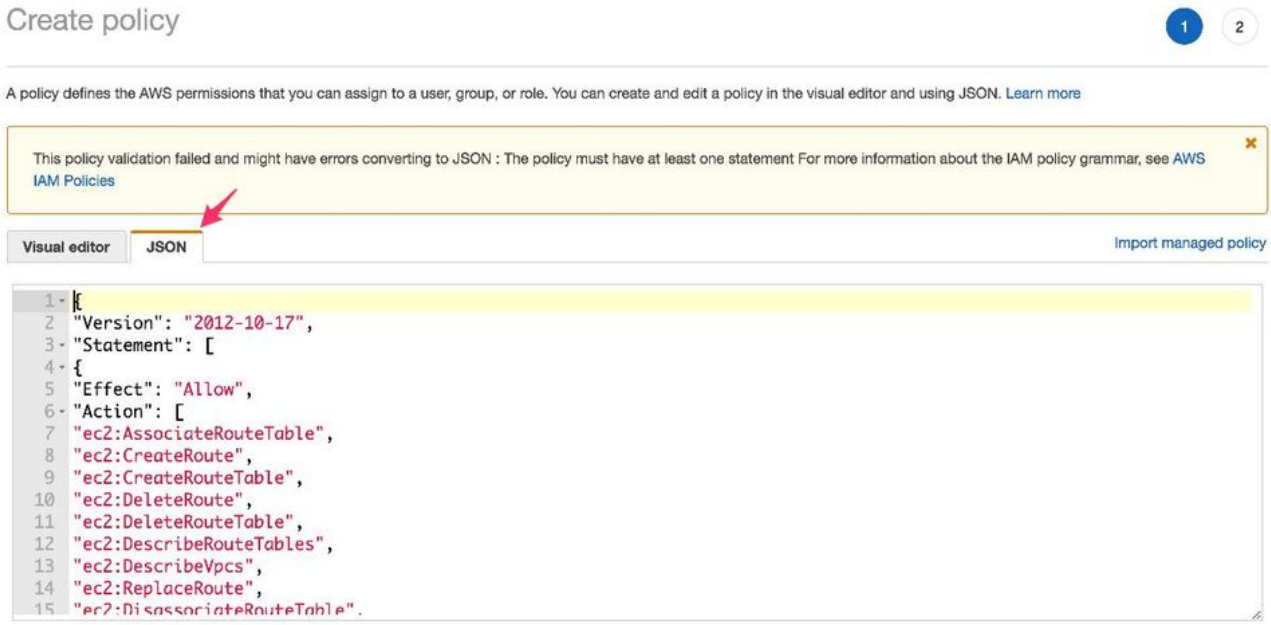


5. 次に、ポリシーを作成し、上で作成したロールに適用する必要があります。IAMダッシュボードで、[ポリシー(Policies)] > [ポリシーの作成(Create Policy)] に移動します。

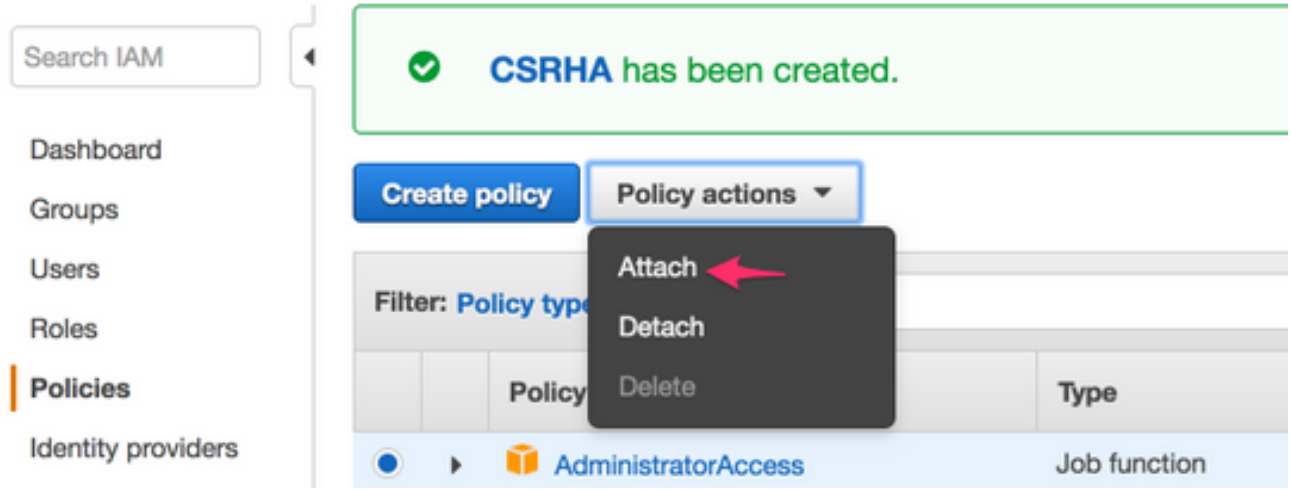



```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"ec2:AssociateRouteTable",
"ec2:CreateRoute",
"ec2:CreateRouteTable",
"ec2:DeleteRoute",
"ec2:DeleteRouteTable",
"ec2:DescribeRouteTables",
"ec2:DescribeVpcs",
"ec2:ReplaceRoute",
"ec2:DisassociateRouteTable",
"ec2:ReplaceRouteTableAssociation"
```

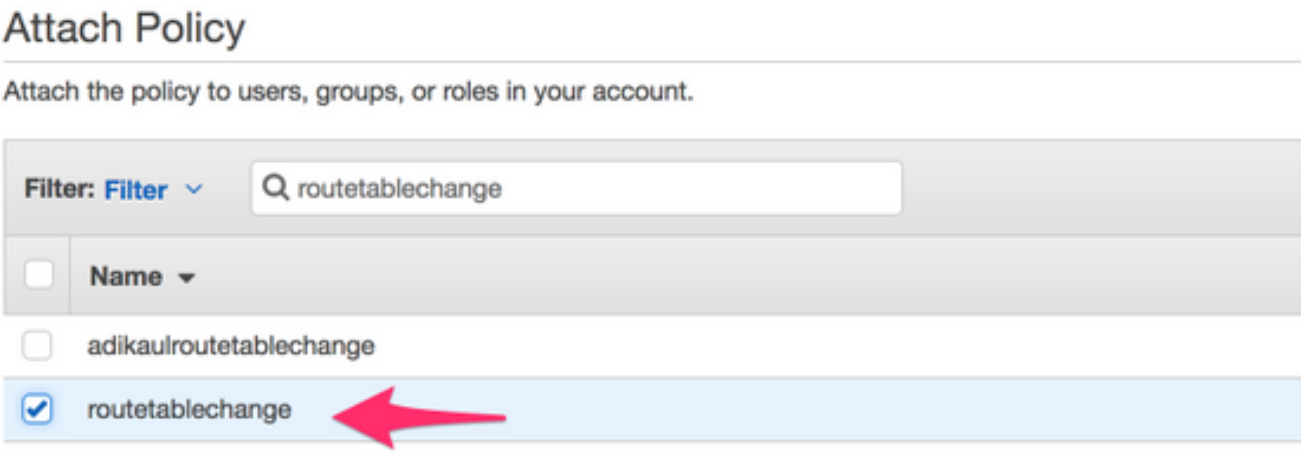
```
],
"Resource": "*"
}
]
}
```



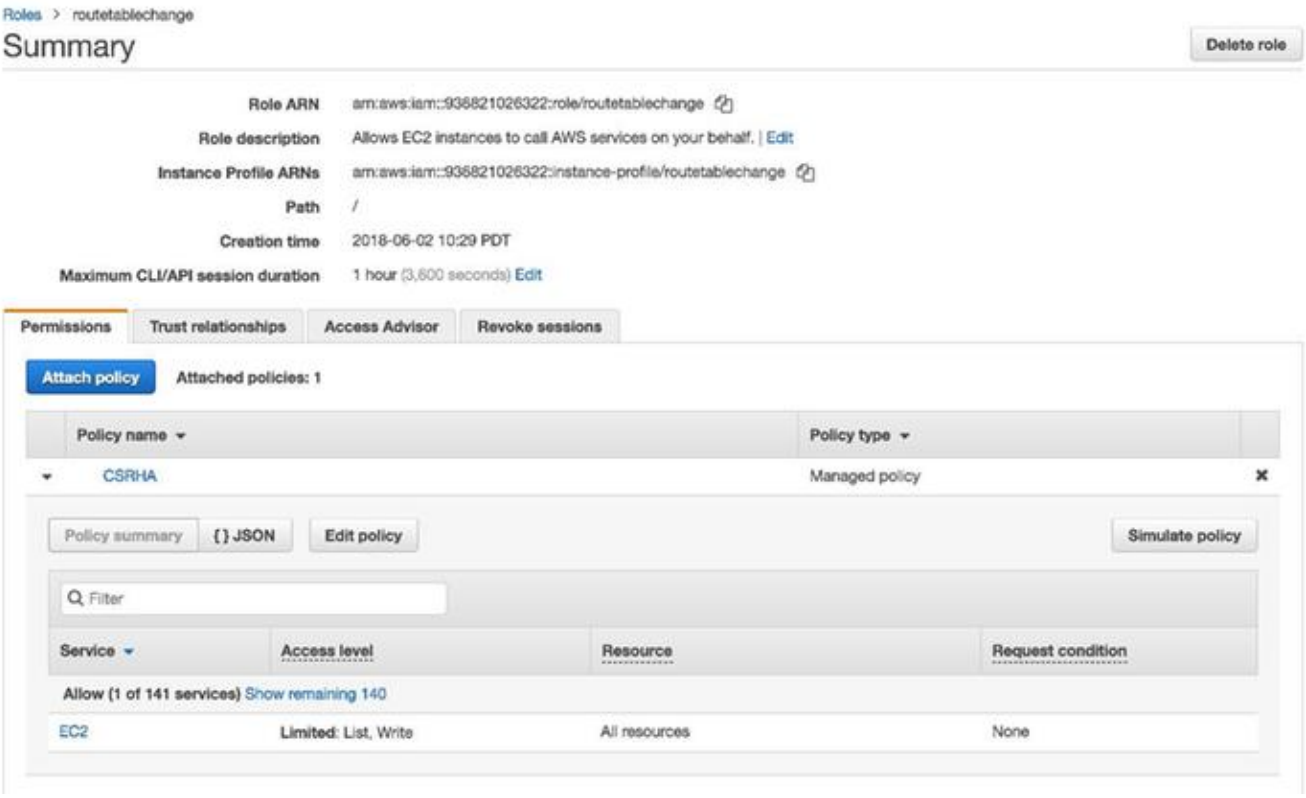
6. ポリシー名を指定し、作成したロールに関連付けます。この例では、図に示すように、ポリシー名はCSRHA with Administrator Accessです。



7. 図に示すように、作成したroutetablechangeというロールにポリシーを割り当てます。



8. サマリー.

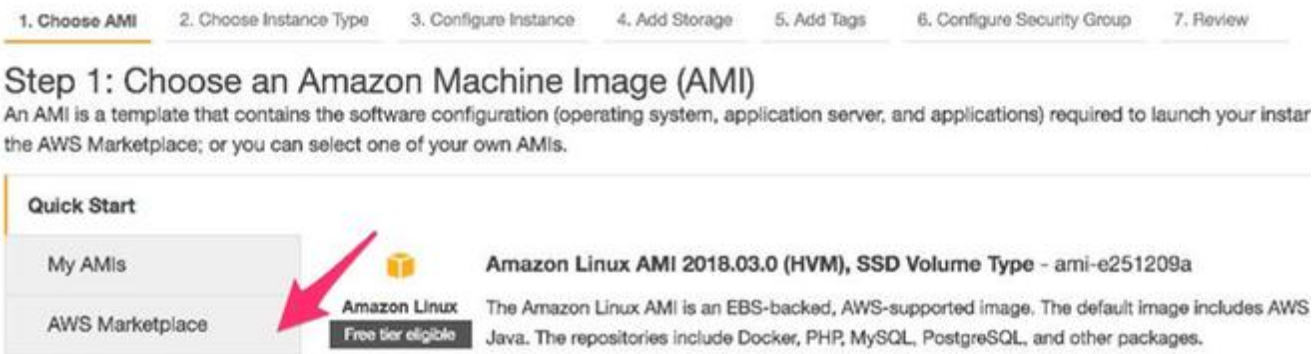


ステップ5：作成したAMIロールでCSR1000vを起動し、パブリック/プライベートサブネットを関連付けます。

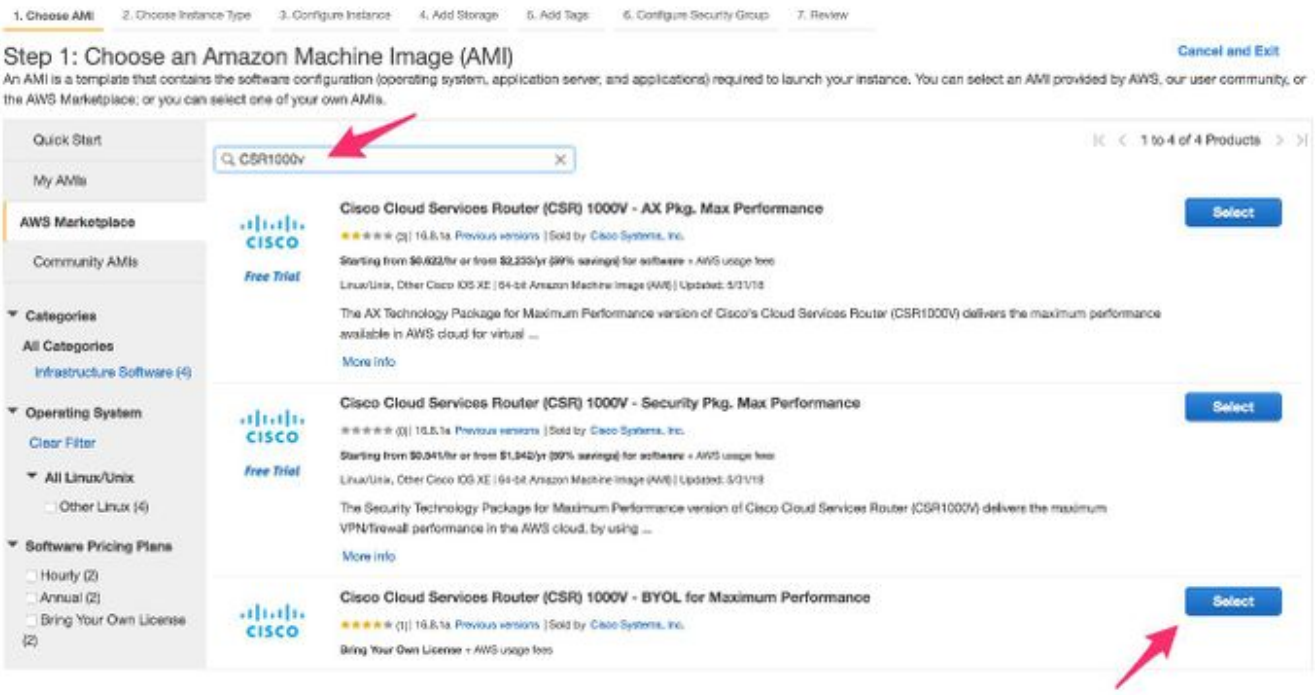
各CSR1000vルータには2つのインターフェイス（パブリック1つ、プライベート1つ）があり、それぞれ独自のアベイラビリティーゾーンにあります。各CSRは別々のデータセンターにあると考えることができます。

1. AWSコンソールで**EC2**を選択し、**Launch Instance**をクリックします。

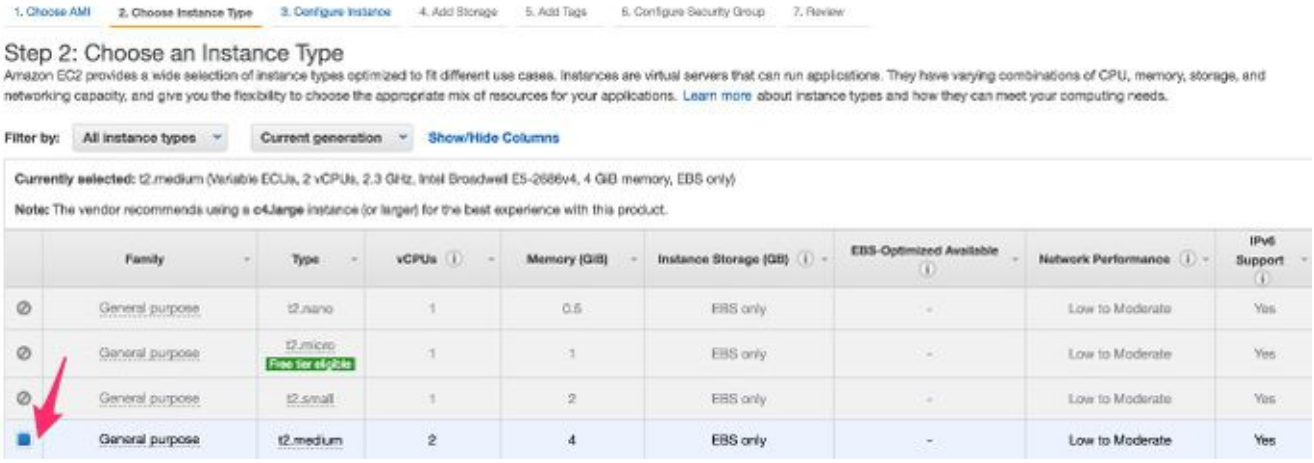
2. [AWS Marketplace]を選択します。



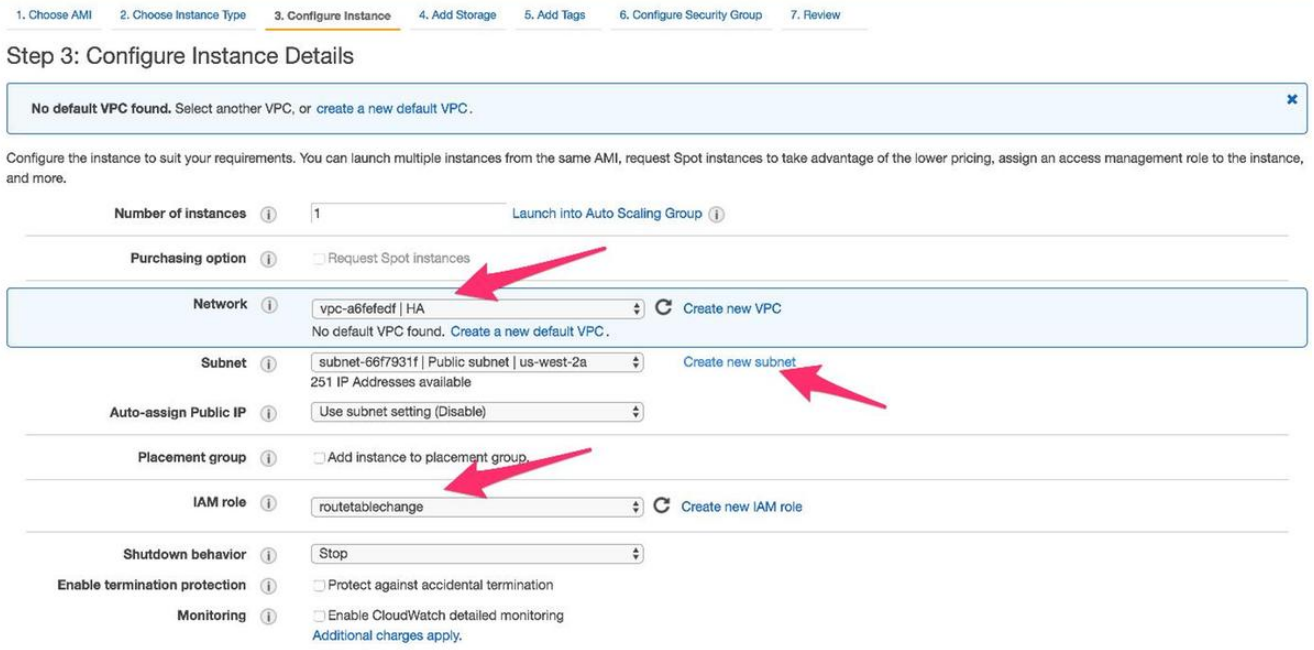
3. 「CSR1000v」と入力します。この例では、[Maximum Performance]にCisco Cloud Services Router(CSR)1000V - BYOLを使用します。



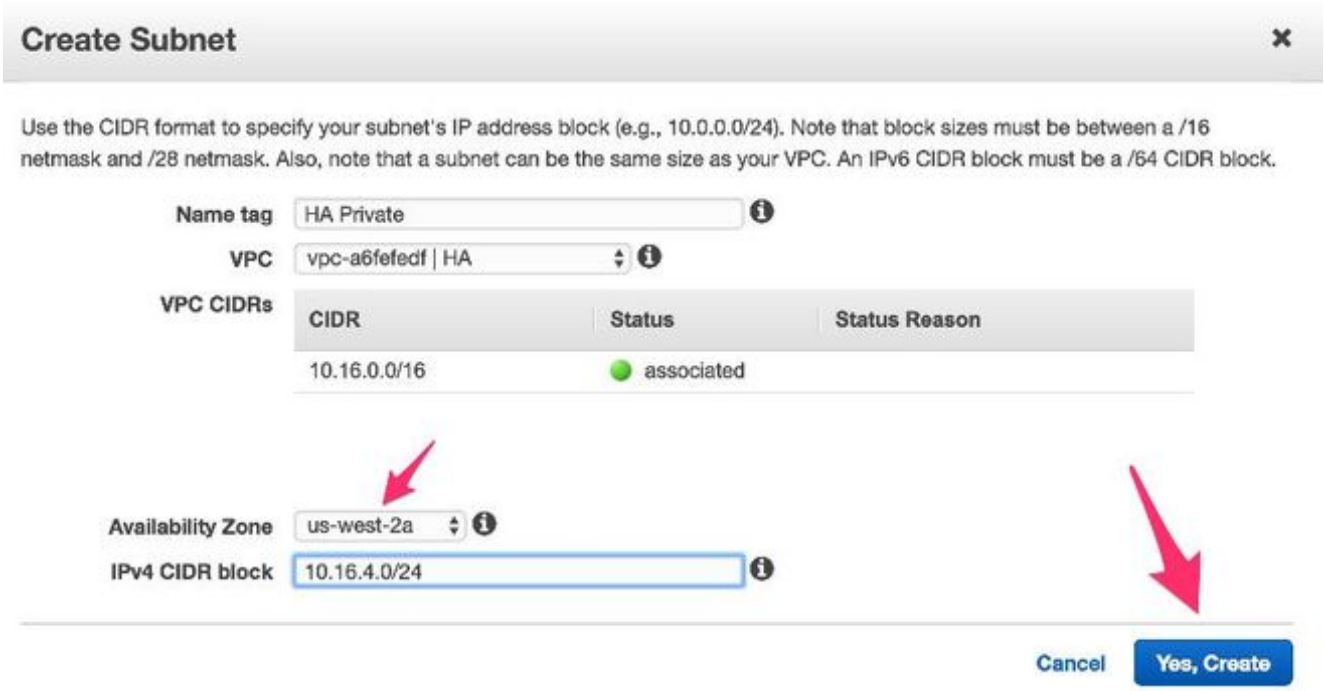
4. インスタンスタイプを選択します。この例では、選択されているタイプはt2.mediumです。



5. インスタンスが設定されている間は、上記で作成したVPCと上記のIAMロールを選択する必要があります。さらに、プライベート側インターフェイスに関連付けるプライベートサブネットを作成します。



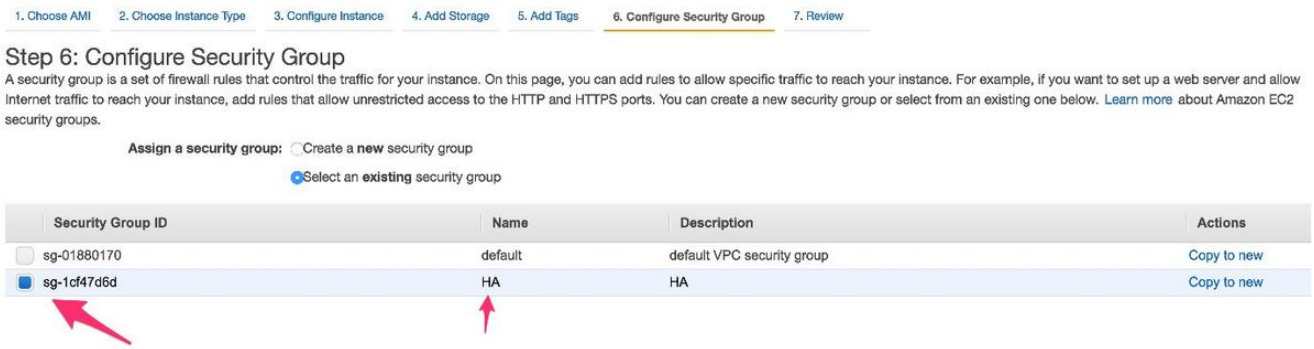
6. [Create new Subnet for Private Subnet]をクリックします。この例では、NameタグはHA Privateです。パブリックサブネットと同じアベイラビリティーゾーンにあることを確認します。



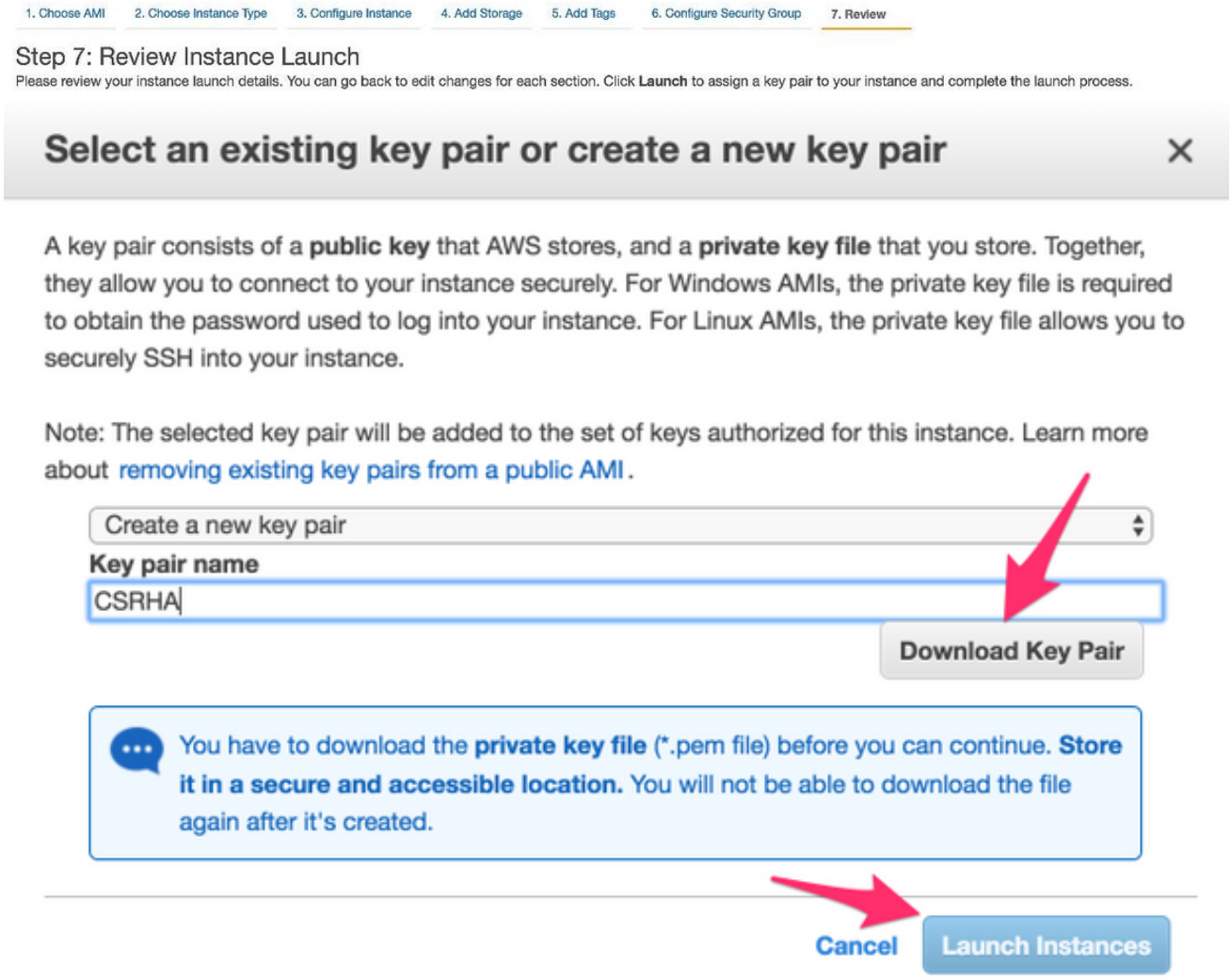
7. 下にスクロールし、[Configure Instance Details]で[Add Device] をクリックします（図を参照）。

8. セカンダリインターフェイスを追加した後、作成したプライベートサブネット（HAプライベート）を関連付けます。Eth0はパブリック側、Eth1はプライベート側のインターフェイスです。 **注：**前の手順で作成したサブネットは、このドロップダウンに表示されないことがあります。サブネットを表示するには、ページを更新またはキャンセルして再起動する必要があります。

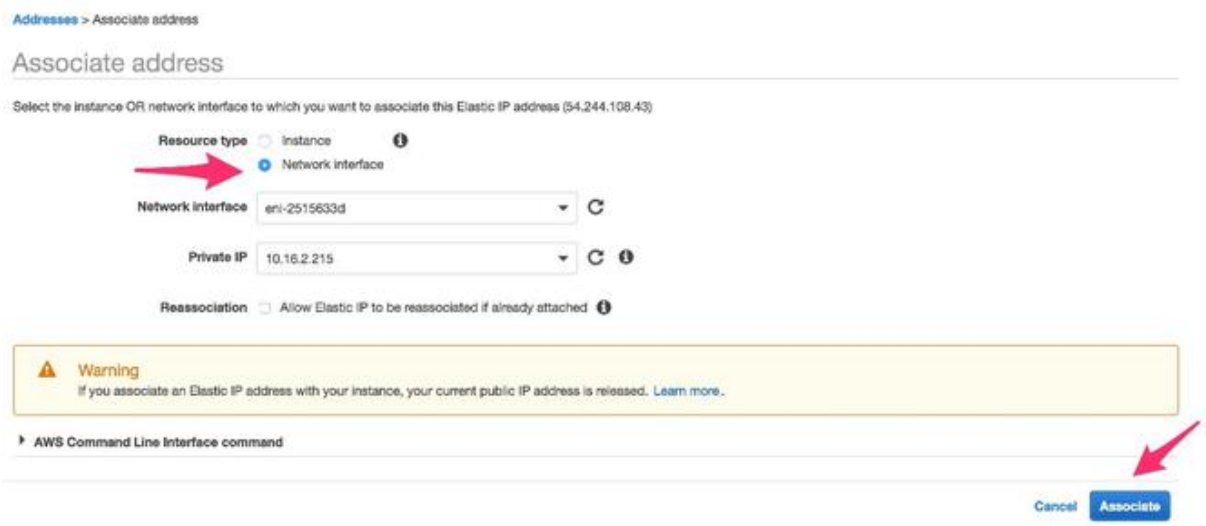
9. VPCで作成したセキュリティグループを選択し、ルールが正しく定義されていることを確認します。



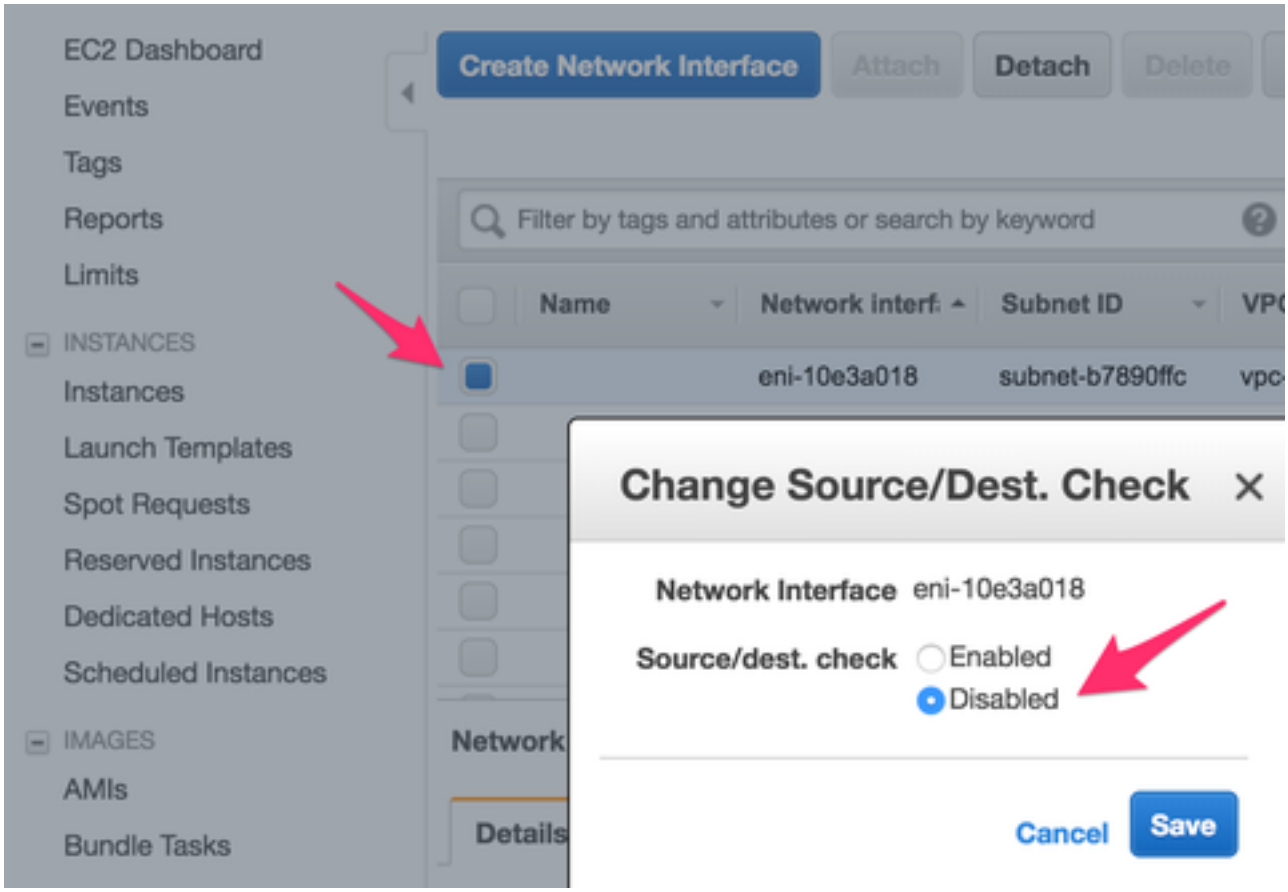
10. 新しいキーペアを作成し、秘密キーをダウンロードします。デバイスごとに1つのキーを再利用できます。　**注：秘密キーを紛失すると、CSRに再びログインできなくなります。キーを回復する方法はありません。**



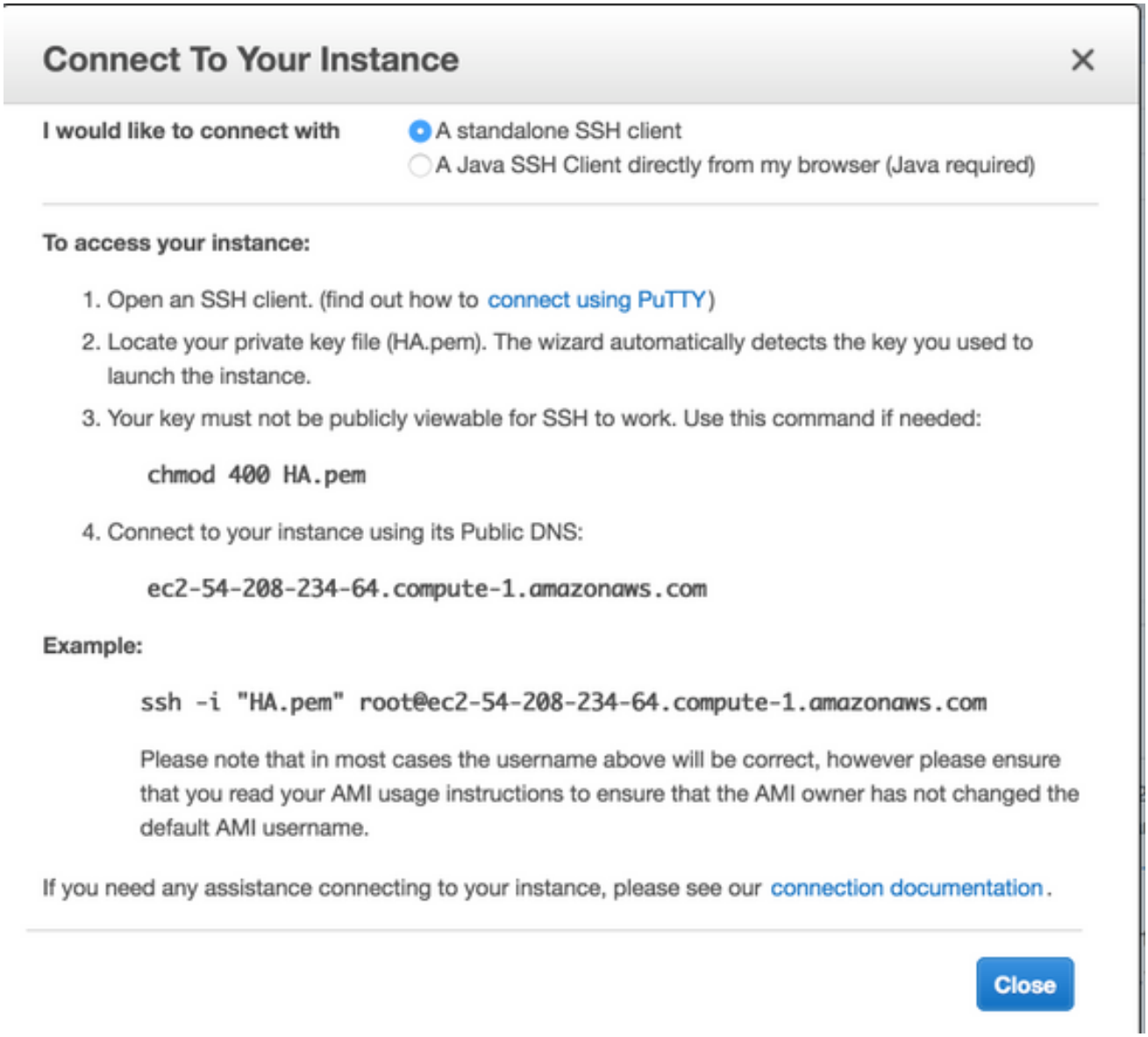
11. 作成したインスタンスのパブリックインターフェイスのENIにElastic IPを関連付け、[AWS console] > [EC2 Management] > [Network Security] > [Elastic IP's] に移動します。 **注：パブリック/プライベートの用語は、ここで混乱を招く可能性があります。この例では、パブリックインターフェイスの定義は、インターネットに面したインターフェイスであるEth0です。AWSの観点から見ると、パブリックインターフェイスはプライベートIPです。**



12. [EC2] > [Network Interfaces] に移動するときに、[Source/Dest Check]を無効にします。各ENIの送信元/宛先チェックを確認します。 デフォルトでは、すべてのENIは、このSource/Destチェックが有効になっています。アンチスプーフィング機能は、ENIがトラフィックを転送する前にトラフィックの宛先であることを確認することによって、ENIが実際には意図されていないトラフィックによってオーバーランすることを回避することを意味しました。ルータがパケットの実際の宛先であることはほとんどありません。この機能は、すべてのCSR中継ENIで無効にする必要があります。無効にしないと、パケットを転送できません。



13. CSR1000vに接続します。 **注**：AWSがCSR1000vにSSHで提供したユーザ名が、rootとして誤ってリストされる場合があります。必要に応じて、これをec2-userに変更します。**注**：SSHでDNSアドレスにpingできる必要があります。ec2-54-208-234-64.compute-1.amazonaws.comです。ルータのパブリックサブネット/eniがパブリックルートテーブルに関連付けられていることを確認します。サブネットをルートテーブルに関連付ける方法については、ステップ8に進みます。



ステップ6：ステップ5を繰り返し、HA用に2番目のCSR1000vインスタンスを作成します。

パブリックサブネット： 10.16.1.0/24

プライベートサブネット：10.16.5.0/24

この新しいAMIのelastic ipアドレスにpingできない場合は、手順8に進んで、パブリックサブネットがパブリックルートテーブルに関連付けられていることを確認します。

ステップ7：ステップ5を繰り返し、AMI MarketplaceからVM(Linux/Windows)を作成します。

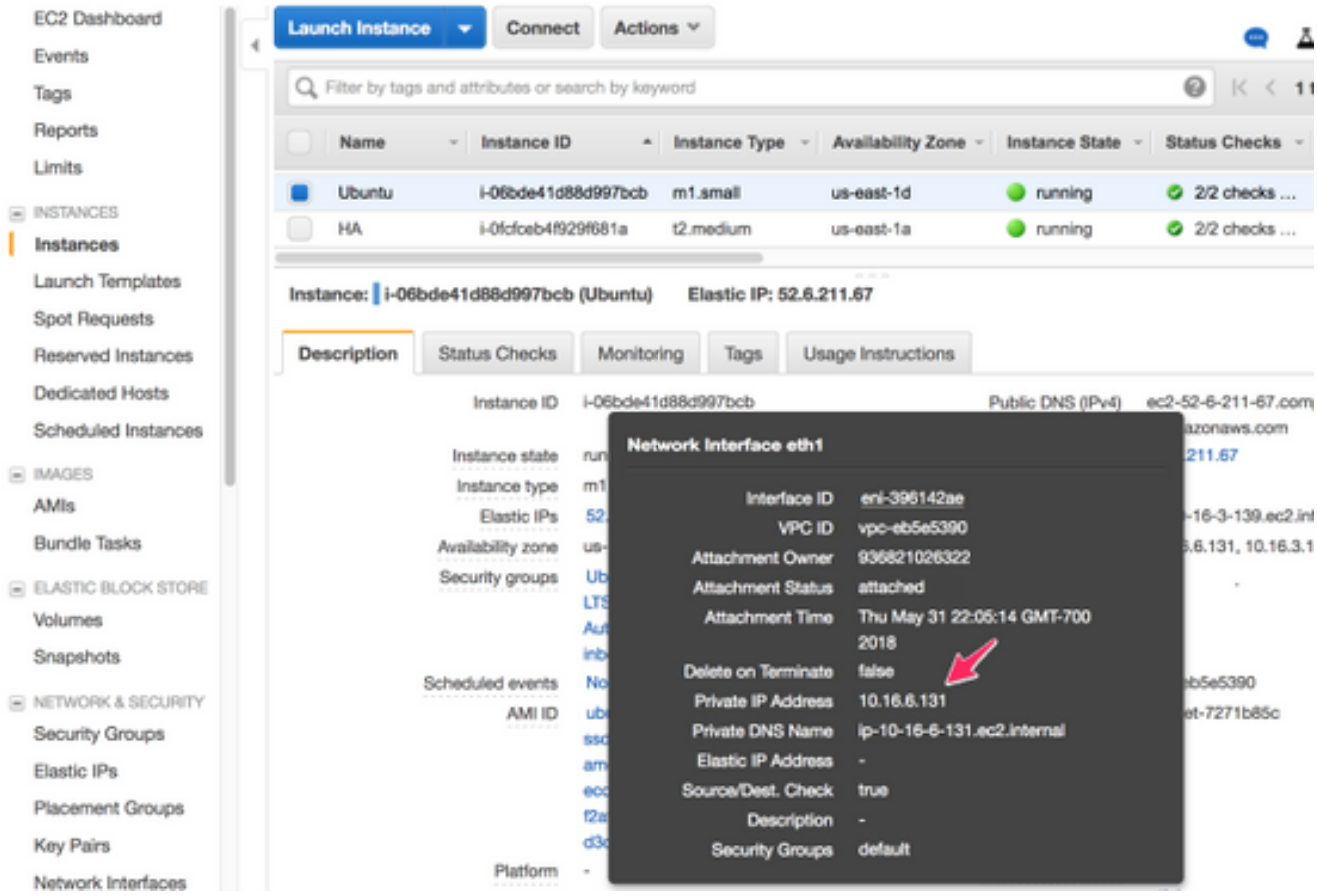
この例では、MarketplaceでUbuntu Server 14.04 LTSを使用します。

パブリックサブネット： 10.16.2.0/24

プライベートサブネット：10.16.6.0/24

この新しいAMIのelastic ipアドレスにpingできない場合は、手順8に進んで、パブリックサブネットがパブリックルートテーブルに関連付けられていることを確認します。

1. Eth0はデフォルトでパブリックインターフェイス用に作成されます。プライベートサブネット用にeth1という名前の2番目のインターフェイスを作成します。



2. Ubuntuで設定するIPアドレスは、AWSによって割り当てられたeth1プライベートインターフェイスです。

```
ubuntu@ip-10-16-2-139:~$ cd /etc/network/interfaces.d/

ubuntu@ip-10-16-2-139:/etc/network/interfaces.d$ sudo vi eth1.cfg

auto eth1
iface eth1 inet static
  address 10.16.6.131
  netmask 255.255.255.0
  network 10.16.6.0
  up route add -host 8.8.8.8 gw 10.16.6.1 dev eth1
```

3. インターフェイスをフラップするか、VMをリブートします。

```
ubuntu@ip-10-16-2-139:/etc/network/interfaces.d$ sudo ifdown eth1 && sudo ifup eth1
ubuntu@ip-10-16-2-139:/etc/network/interfaces.d$ sudo reboot
```

4. テストのために8.8.8.8をpingします。8.8.8.8ルートがステップ7ごとに追加されていることを確認します。

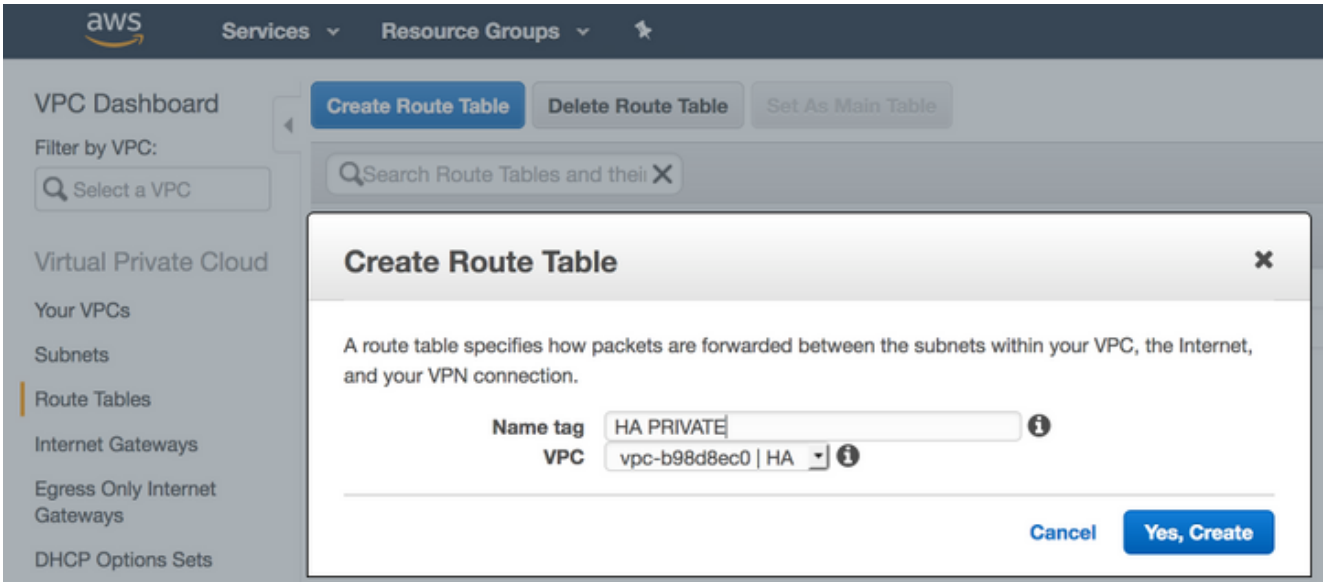
```
ubuntu@ip-10-16-2-139:~$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 10.16.2.1 0.0.0.0 UG 0 0 0 eth0
8.8.8.8 10.16.6.1 255.255.255.255 UGH 0 0 0 eth1     <--------------
10.16.3.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
10.16.6.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
```

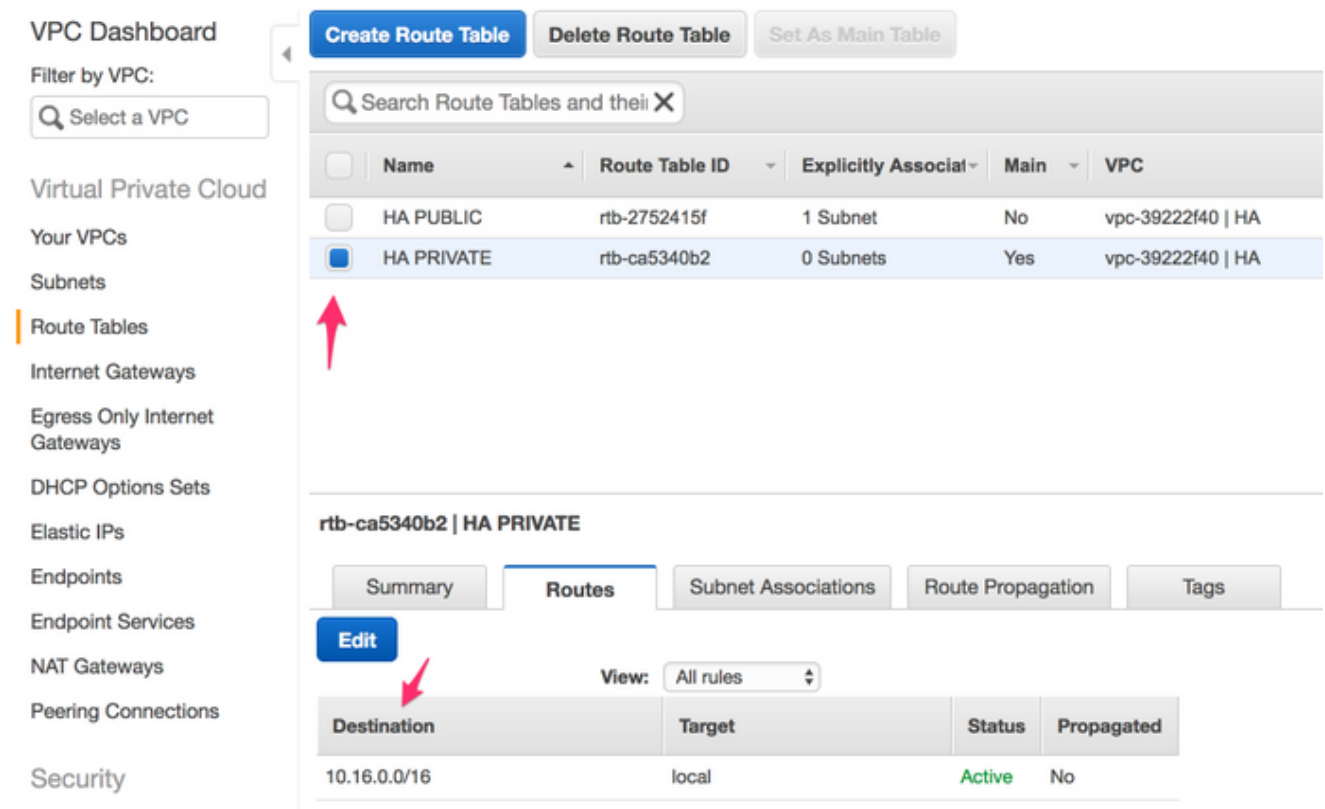
8.8.8.8が表に記載されていない場合は、手動で追加します。

```
ubuntu@ip-10-16-2-139:~$ sudo route add -host 8.8.8.8 gw 10.16.6.1 dev eth1
```

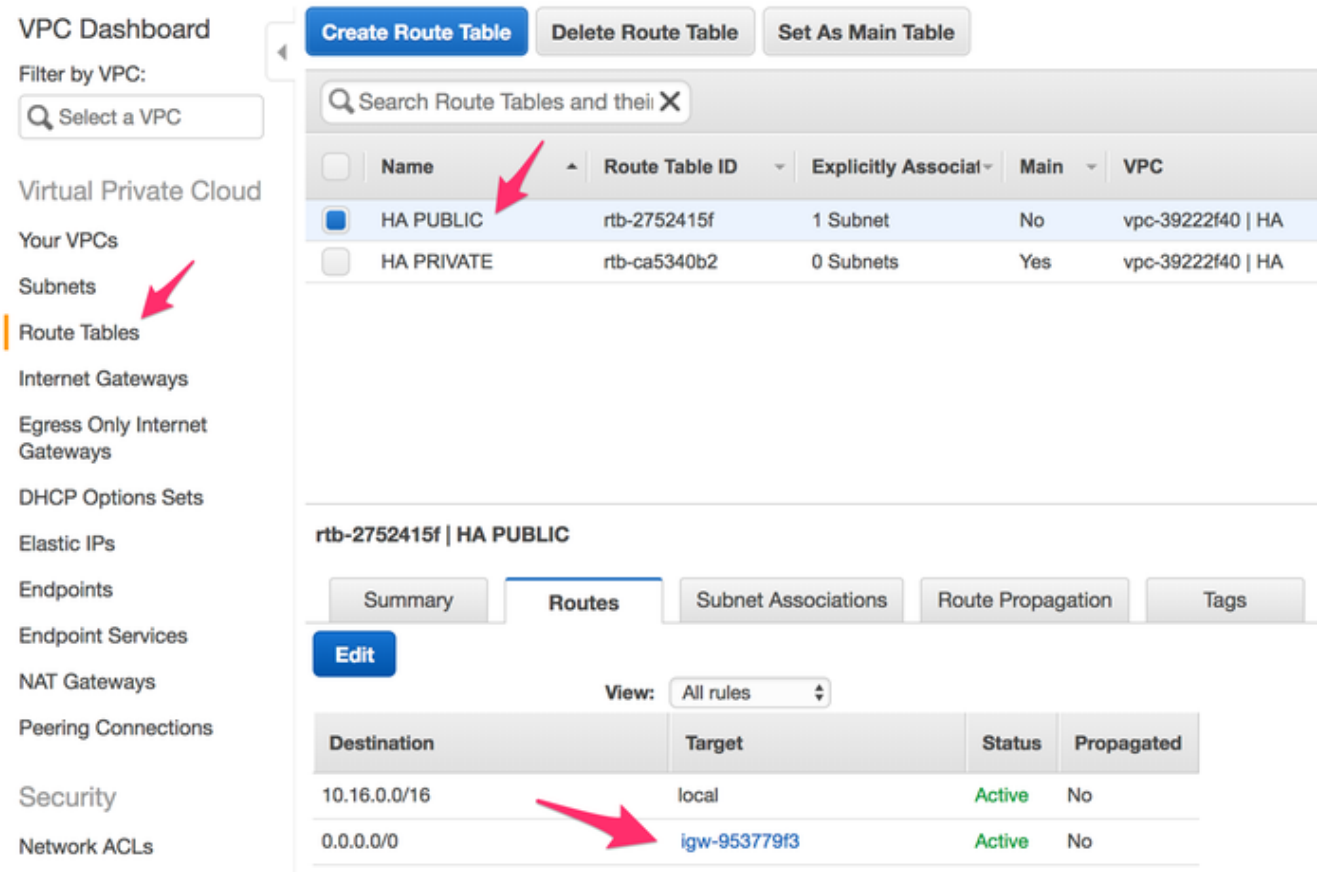
ステップ8：プライベートおよびパブリックルートテーブルを設定します。

1. ステップ2のウィザードでVPCを作成すると、2つのルートテーブルが自動的に作成されます。ルートテーブルが1つしかない場合は、図に示すように、プライベートサブネット用に別のルートテーブルを作成します。

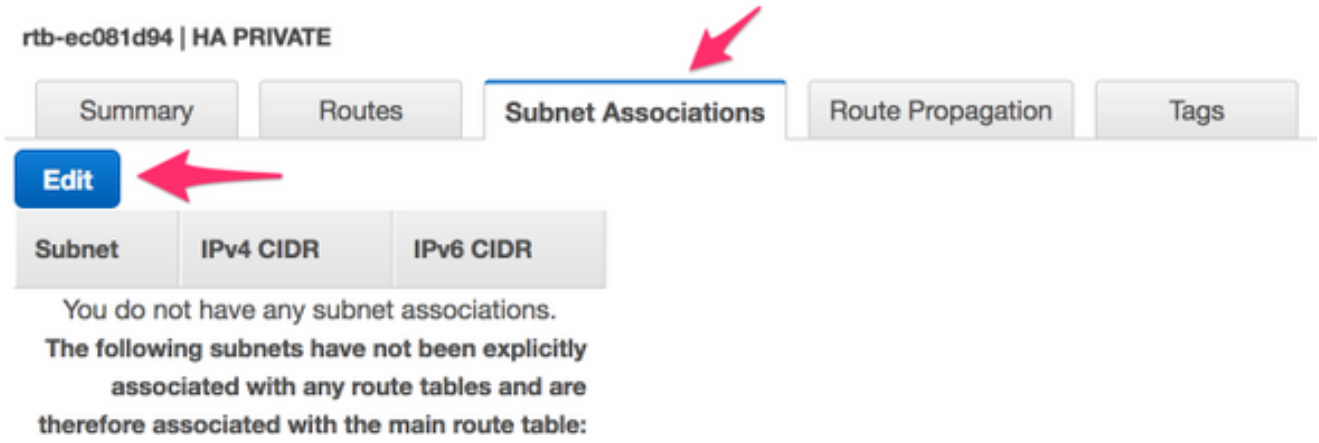




2. 2つのルートテーブルのビューを次に示します。PUBLICルートテーブルには、インターネットゲートウェイ(igw-95377973)が自動的に接続されます。2つのテーブルにラベルを付けます。PRIVATEテーブルにはこのルートを含めることはできません。



3. 6つのサブネットすべてを適切なルートテーブルに関連付けます 3パブリックインターフェイスは、パブリックルートテーブルに関連付けられています。パブリックサブネット：10.16.0.0/24, 10.16.1.0/24, 10.16.2.0/24 3プライベートインターフェイスはプライベートルートテーブルに関連付けられます。プライベートサブネット：10.16.4.0/24, 10.16.5.0/24, 10.16.6.0/24



手順9:BFDと任意のルーティングプロトコルを使用して、ネットワークアドレス変換(NAT)とGREトンネルを設定します。

CSR 1000vのElastic IPを介してGeneric Routing Encapsulation(GRE)トンネルを設定します（誤った障害を検出するDHCPリース更新の問題を回避するために推奨されます）。 高速コンバージェンスが必要な場合は、双方向フォワーディング検出(BFD)値を、この例に示す値よりもアグレッシブに設定できます。ただし、これが原因で、断続的な接続中にBFDピアダウンイベントが発生する可能性があります。この例の値は、1.5秒以内にピア障害を検出します。AWS APIコマンドが実行されてからVPCルーティングテーブルの変更が有効になるまでの間に、約20秒の可変遅延があります。

- CSRHAの設定
  GREおよびBFD:HAフェールオーバーの状態を確認するために使用されます。

```
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 52.10.183.185 /* Elastic IP of the peer CSR */
!
router eigrp 1
  bfd interface Tunnel1
  network 192.168.1.0
  passive-interface GigabitEthernet1
```

NATとルーティング：プライベートインターフェイスを介したVMインターネットの到達可能性に使用されます。

```
interface GigabitEthernet1
  ip address dhcp
  ip nat outside
  no shutdown
!
interface GigabitEthernet2
  ip address dhcp
  ip nat inside
  no shutdown
!
ip nat inside source list 10 interface GigabitEthernet1 overload
!
access-list 10 permit 10.16.6.0 0.0.0.255
!
ip route 10.16.6.0 255.255.255.0 GigabitEthernet2 10.16.4.1
```

- CSRHA1の設定
  GREおよびBFD:HAフェールオーバーの状態を確認するために使用されます。

```
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 50.112.227.77 /* Elastic IP of the peer CSR */
!
router eigrp 1
  bfd interface Tunnel1
  network 192.168.1.0
  passive-interface GigabitEthernet1
```

NATとルーティング：プライベートインターフェイスを介したVMインターネットの到達可能性に使用されます。

```
interface GigabitEthernet1
  ip address dhcp
  ip nat outside
  no shutdown
!
interface GigabitEthernet2
```

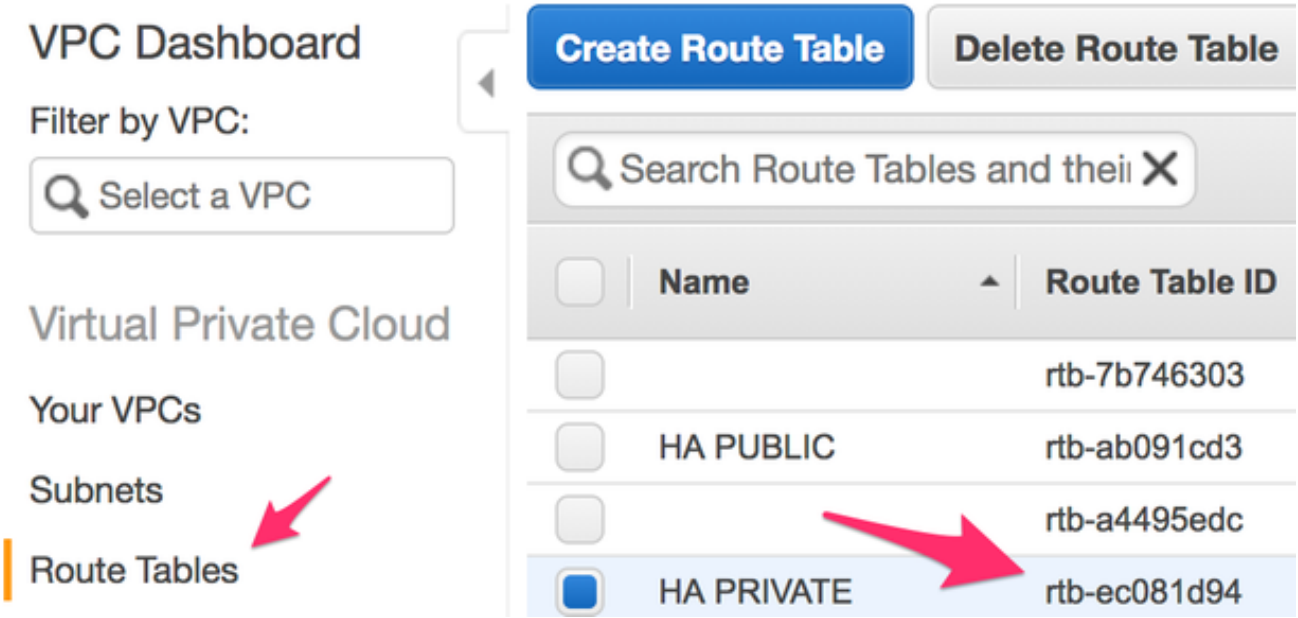
```
  ip address dhcp
  ip nat inside
  no shutdown
!
ip nat inside source list 10 interface GigabitEthernet1 overload
!
access-list 10 permit 10.16.6.0 0.0.0.255
!
ip route 10.16.6.0 255.255.255.0 GigabitEthernet2 10.16.5.1
```

ステップ10：ハイアベイラビリティの設定（Cisco IOS XE Denali 16.3.1a以降）

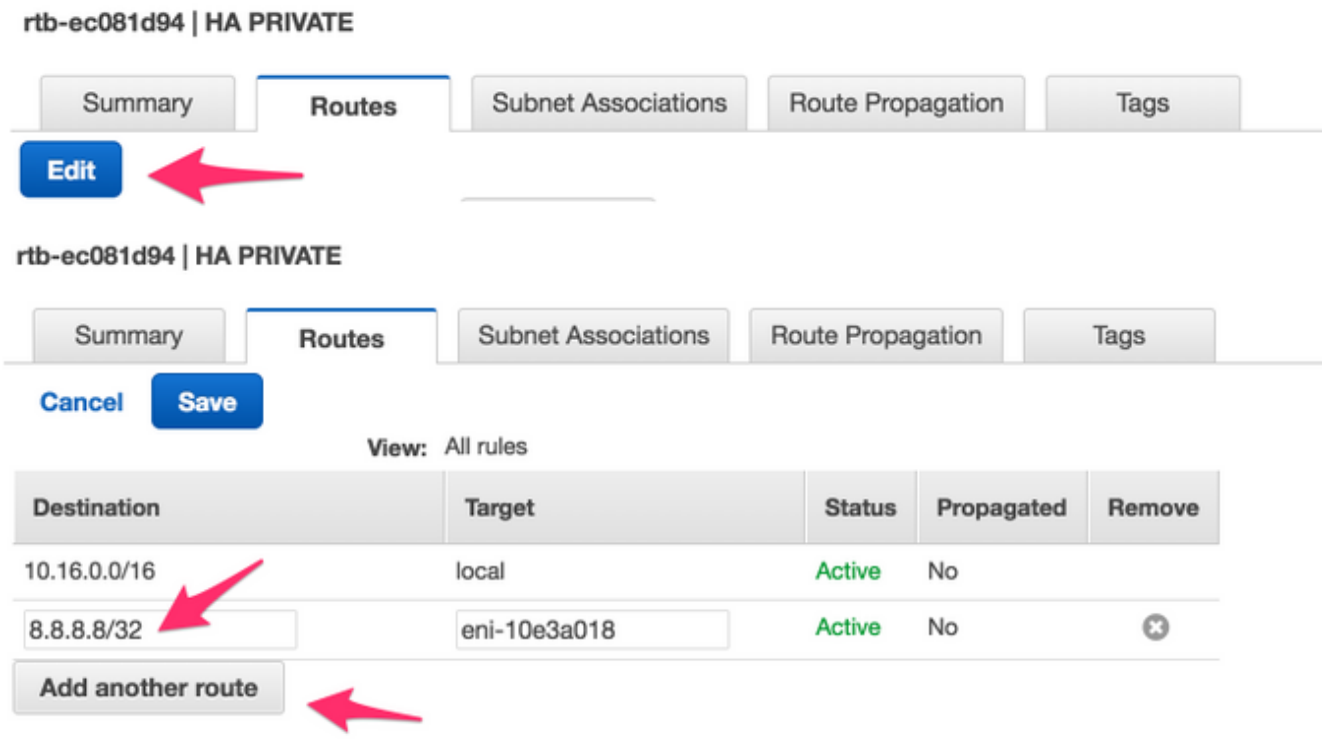
次に示すcloud provider awsコマンドを使用して各CSR 1000vを設定し、BFDピアダウンイベントを監視します。BFDピアダウンなどのAWS HAエラーが検出された後に、(VPC)ルートテーブルID、ネットワークインターフェイスID、およびCIDRへのルーティング変更を定義するには、このコマンドを使用します。

```
CSR(config)# redundancy
CSR(config-red)# cloud provider [aws | azure] node-id
# bfd peer ipaddr
# route-table table-name
# cidr ip ipaddr/prefix
# eni elastic-network-intf-name
# region region-name
```

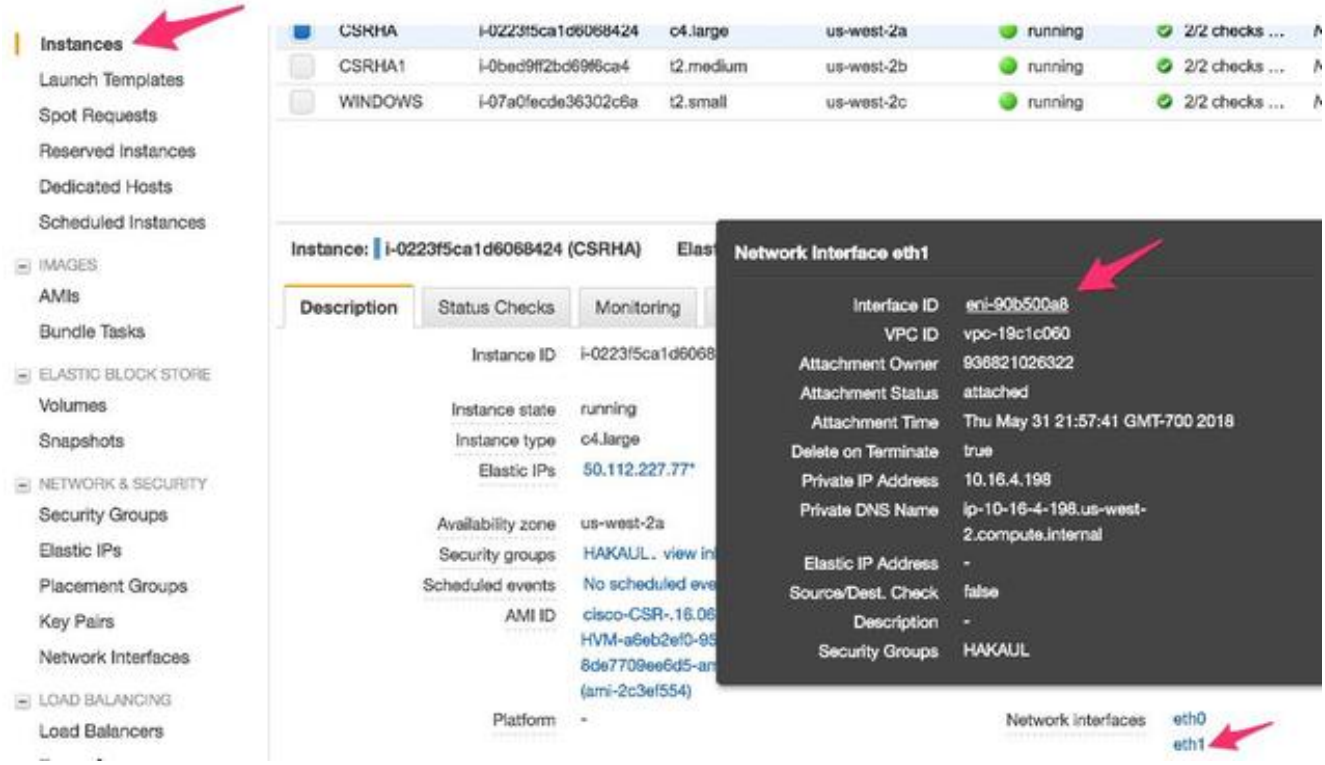
1. #bfd peer ipaddrは、ピアトンネルIPアドレスです。
   ```
   CSRHA#show bfd neighbors

   IPv4 Sessions
   NeighAddr LD/RD RH/RS State Int
   192.168.1.2 4097/4097 Up Up Tu1
   ```
2. AWSコンソ#route-tableルの下にテーブル名が表示されます。[VPC] > [Route Tables] に移動します。このアクションにより、プライベートルートテーブルが変更されます。



3. #cidr ip ipaddr/prefixは、ルートテーブルで更新されるルートの宛先アドレスです。AWSコンソールで、[VPC] > [Route Tables] に移動します。下にスクロールして[Edit] をクリックし、[Add another route] をクリックします。8.8.8.8のテスト宛先アドレスとCSRHAのプライベートENIを追加します。



4. #eni elastic-network-intf-nameがEC2インスタンスで見つかりました。対応する各CSRのプライベート側インターフェイスeth1をクリックし、インターフェイスIDを使用します。



5. 名前は#regionAWSドキュメント内で見つかったコード名です。このリストは変更されたり拡大したりする可能性があります。最新のアップデートについては、Amazonの[Region and Availability Zones]ドキュメントをご覧ください。

Code	Name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
ca-central-1	Canada (Central)
eu-central-1	EU (Frankfurt)
eu-west-1	EU (Ireland)
eu-west-2	EU (London)
eu-west-3	EU (Paris)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-northeast-3	Asia Pacific (Osaka-Local)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-south-1	Asia Pacific (Mumbai)
sa-east-1	South America (São Paulo)

CSRHAの冗長性の設定例

```
redundancy
cloud provider aws 1
  bfd peer 192.168.1.2
  route-table rtb-ec081d94
  cidr ip 8.8.8.8/32
  eni eni-90b500a8
  region us-west-2
```

CSRHA1の冗長性の設定例

```
redundancy
cloud provider aws 1
  bfd peer 192.168.1.1
  route-table rtb-ec081d94
  cidr ip 8.8.8.8/32
  eni eni-10e3a018
  region us-west-2
```

ハイアベイラビリティの確認

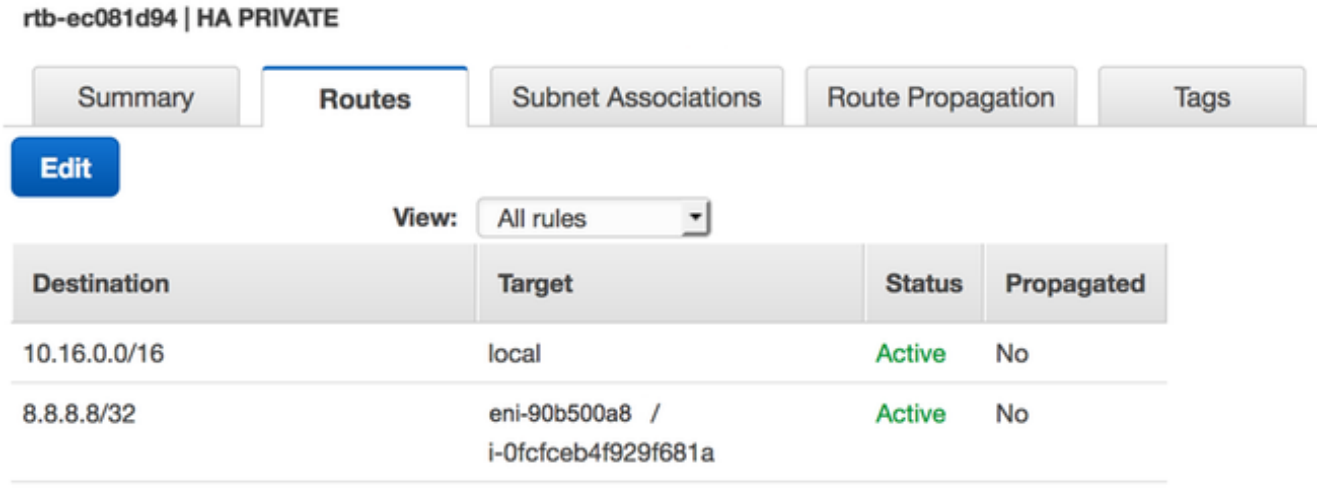
1. BFDとクラウドの設定を確認します。

```
CSRHA#show bfd nei

IPv4 Sessions
NeighAddr LD/RD RH/RS State Int
192.168.1.2 4097/4097 Up Up Tu1


CSRHA#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H Address Interface Hold Uptime SRTT RTO Q Seq
(sec) (ms) Cnt Num
0 192.168.1.2 Tu1 12 00:11:57 1 1470 0 2


CSRHA#show redundancy cloud provider aws 1

Cloud HA: work_in_progress=FALSE
Provider : AWS node 1
State : idle
BFD peer     = 192.168.1.2
BFD intf     = Tunnel1
route-table  = rtb-ec081d94
cidr         = 8.8.8.8/32
eni          = eni-90b500a8
region       = us-west-2
```

2. VMから宛先に連続してpingを実行します。pingがプライベートeth1インターフェイスを経由していることを確認します。

```
ubuntu@ip-10-16-3-139:~$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 10.16.6.131 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=50 time=1.60 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=50 time=1.62 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=50 time=1.57 ms
```

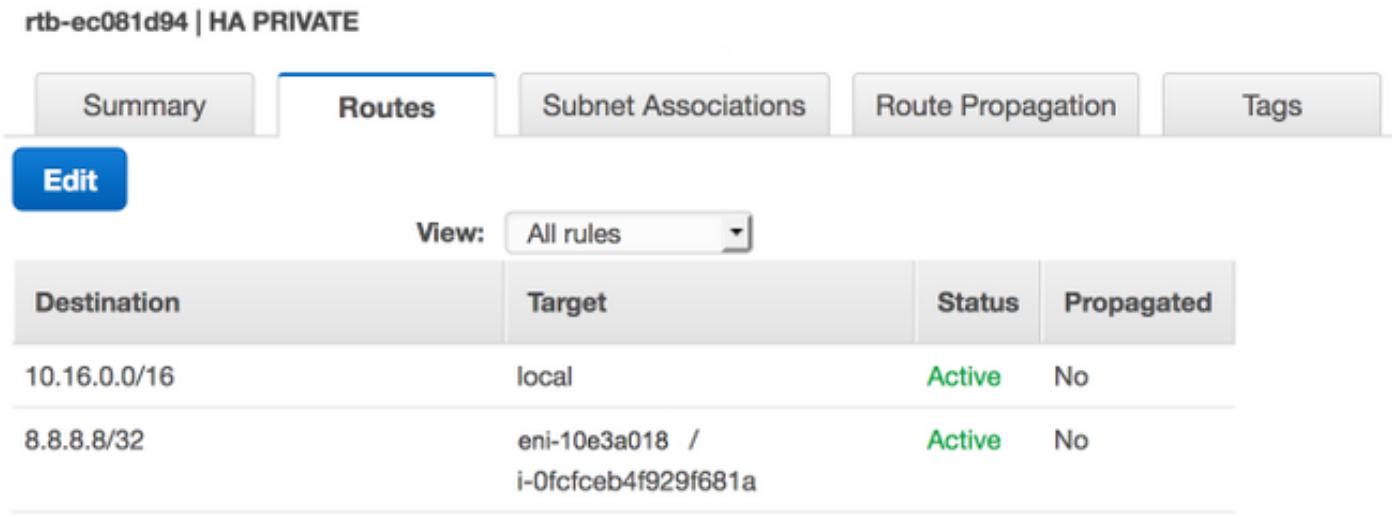
3. プライベートルートテーブルを確認します。eniは現在、これがトラフィックであるCSRHAのプライベートインターフェイスです。



4. CSRHAのTunnel1をシャットダウンして、HAフェールオーバーをシミュレートします。

```
CSRHA(config)#int Tun1
CSRHA(config-if)#shut
```

5. ルートテーブルがCSRHA1のプライベートインターフェイスである新しいENIを指していることを確認します。



トラブルシュート

- リソースが関連付けられていることを確認します。VPC、サブネット、インターフェイス、ルートテーブルなどを作成する場合、これらの多くは自動的に相互に関連付けられません。彼らはお互いに何の知識も持っていない。
- Elastic IPと任意のプライベートIPが正しいインターフェイスに関連付けられており、正しいサブネットが正しいルートテーブルに追加され、正しいルータに接続されていること、正しいVPCとゾーンがIAMロールとセキュリティグループにリンクされていることを確認します。
- ENIごとにSource/Destチェックを無効にします。
- Cisco IOS XE 16.3.1a以降では、次の追加の検証コマンドを使用できます。

```
show redundancy cloud provider [aws | azure] node-id
debug redundancy cloud [all | trace | detail | error]
debug ip http all
```

- デバッグでよく見られる障害を次に示します。

問題：httpc_send_requestが失敗しました

解決策：Httpは、CSRからAWSにAPIコールを送信するために使用されます。DNSがインスタンスにリストされているDNS名を解決できることを確認します。HTTPトラフィックがブロックされていないことを確認します。

```
*May 30 20:08:06.922: %VXE_CLOUD_HA-3-FAILED: VXE Cloud HA BFD state transitioned, AWS node 1
event httpc_send_request failed
*May 30 20:08:06.922: CLOUD-HA : AWS node 1 httpc_send_request failed (0x12)
URL=http://ec2.us-east-2b.amazonaws.com
```

問題：ルートテーブルrtb-9c0000f4とインターフェイスeni-32791318が異なるネットワークに属している

解決策：リージョン名とENIが異なるネットワークで誤って設定されている。リージョンとENIは

、ルータと同じゾーンに存在する必要があります。

```
*May 30 23:38:09.141: CLOUD-HA : res content iov_len=284 iov_base=<?xml version="1.0"
encoding="UTF-8"?>
<Response><Errors><Error><Code>InvalidParameterValue</Code><Message>route table rtb-9c0000f4 and
interface eni-32791318 belong to different
networks</Message></Error></Errors><RequestID>af3f228c-d5d8-4b23-b22c-
f6ad999e70bd</RequestID></Response>
```

問題：この操作を実行する権限がありません。Encoded Authorization Failureメッセージ。

解決策：IAM JSONロール/ポリシーが正しく作成されていないか、CSRに適用されていません。IAMロールは、CSRがAPI呼び出しを実行することを許可します。

```
*May 30 22:22:46.437: CLOUD-HA : res content iov_len=895 iov_base=<?xml version="1.0"
encoding="UTF-8"?>
<Response><Errors><Error><Code>UnauthorizedOperation</Code><Message>You are not authorized to
perform this operation. Encoded
authorization failure message: qYvEB4MUdOB8m2itSteRgnOuslAaxhAbDph5qGRJkjJbrESajbmF5HWUR-
MmHYeRAlpKZ3Jg_y-
_tMlYel5l_ws8Jd9q2W8YDXBl3uXQqfW_cjjrgy9jhnGY0nOaNu65aLpfqui8kS_4RPOpm5grRFfo99-
8uv_N3mYaBqKFPn3vUcSYKBmxFIIkJKcjY9esOeLIOWDcnYGGu6AGGMoMxWDtk0K8nwk4IjLDcnd2cDXeENS45w1PqzKGPsH
v3wD28TS5xRjIrPXYrT18UpV6lLA_09Oh4737VncQKfzbz4tPpnAkoW0mJLQ1vDpPmNvHUpEng8KrGWYNfbfemoDtWqIdABf
aLLLmh4saNtnQ_OMBoTi4toBLEb2BNdMkl1UVBIxqTqdFUVRS**MSG 00041 TRUNCATED** **MSG 00041
CONTINUATION
#01**qLosAb5Yx0DrOsLSQwzS95VGvQM_n87LBHYbAWWhqWj3UfP_zmiak7dlm9P41mFCucEB3Cs4FRsFtb-
9q44VtyQJaS2sU2nhGe3x4uGEsl7F1pNv5vhVeYOZB3tbOfbV1_Y4trZwYPFgLKgBShZp-WNmUKUJsKc1-
6KGqmp7519imvh66JgwgmU9DT_qAZ-jEjkqWjBrxg6krw</Message></Error></Errors><RequestID>4cf31249-
2a6e-4414-ae8d-6fb825b0f398</RequestID></Response>
```

関連情報

- VPCゲートウェイの冗長性 – Cisco
- アマゾンウェブサービス向けCisco CSR 1000vシリーズクラウドサービスルータ導入ガイド
- インスタンスタイプの細分化
- EC2およびVPC
- EC2ユーザガイドのElastic Network Interfacesには、インスタンスタイプごとのENIの数が含まれています
- LinuxでのEnhanced Networkingの使い方、役立つ背景情報
- 専用インスタンス/テナントの説明と操作方法
- 一般的なEC2ドキュメント
- 一般的なVPCドキュメント
- リージョンとアベイラビリティーゾーン
- CSR1000v高可用性バージョン3

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。