

# Amazon AWS의 CSR1000v HA 리던던시 구축 설명서

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[목표](#)

[토폴로지](#)

[네트워크 다이어그램](#)

[용어](#)

[제한 사항](#)

[설정](#)

[1단계. 지역을 선택합니다.](#)

[2단계. VPC를 생성합니다.](#)

[3단계. VPC에 대한 보안 그룹을 생성합니다.](#)

[4단계. IAM 역할을 정책과 생성하고 VPC에 연결합니다.](#)

[5단계. 생성한 AMI 역할로 CSR1000v를 시작하고 퍼블릭/프라이빗 서브넷을 연결합니다.](#)

[6단계. 5단계를 반복하고 HA에 대해 두 번째 CSR1000v 인스턴스를 생성합니다.](#)

[7단계. 5단계를 반복하여 AMI Marketplace에서 VM\(Linux/Windows\)을 생성합니다.](#)

[8단계. 프라이빗 및 퍼블릭 경로 테이블을 구성합니다.](#)

[9단계. BFD를 사용하여 NAT\(Network Address Translation\) 및 GRE 터널 및 모든 라우팅 프로토콜을 구성합니다.](#)

[10단계. 고가용성을 구성합니다\(Cisco IOS XE Denali 16.3.1a 이상\).](#)

[고가용성 확인](#)

[문제 해결](#)

[문제: httpc send request 실패](#)

[문제: 경로 테이블 rtb-9c0000f4와 인터페이스 eni-32791318이 다른 네트워크에 속함](#)

[문제/장애: 이 작업을 수행할 권한이 없습니다. 인코딩된 권한 부여 실패 메시지입니다.](#)

[관련 정보](#)

## 소개

이 문서에서는 Amazon AWS 클라우드에서 고가용성용 CSR1000v 라우터를 구축하는 방법에 대한 컨피그레이션 가이드를 설명합니다. 사용자에게 HA에 대한 실용적인 지식과 완전한 기능을 갖춘 테스트베드를 구축할 수 있는 기능을 제공하기 위한 것입니다.

AWS 및 HA에 대한 자세한 배경은 [섹션](#)을 참조하십시오.

## 사전 요구 사항

## 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Amazon AWS 계정
- 동일한 지역에 CSR1000v 2개 및 Linux/Windows AMI 1개
- HA 버전 1은 Cisco IOS-XE® 버전 16.5~16.9에서 지원됩니다. 16.11 이상에서는 HA 버전 3을 사용합니다.

## 사용되는 구성 요소

이 문서의 정보는 Cisco IOS-XE® Denali 16.7.1을 기반으로 합니다.

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

## 목표

다중 가용 영역 환경에서 VM(사설 데이터 센터)에서 인터넷으로의 연속 트래픽을 시뮬레이션합니다. HA 장애 조치를 시뮬레이션하고 라우팅 테이블이 CSRHA에서 CSRHA1의 프라이빗 인터페이스로 트래픽을 스위칭할 때 HA가 성공하는지 확인합니다.

## 토폴로지

컨피그레이션을 시작하기 전에 토폴로지와 설계를 완전히 이해하는 것이 중요합니다. 이렇게 하면 나중에 잠재적인 문제를 해결할 수 있습니다.

네트워크 요구 사항에 따라 다양한 HA 구축 시나리오가 있습니다. 이 예에서는 HA 이중화가 다음 설정으로 구성됩니다.

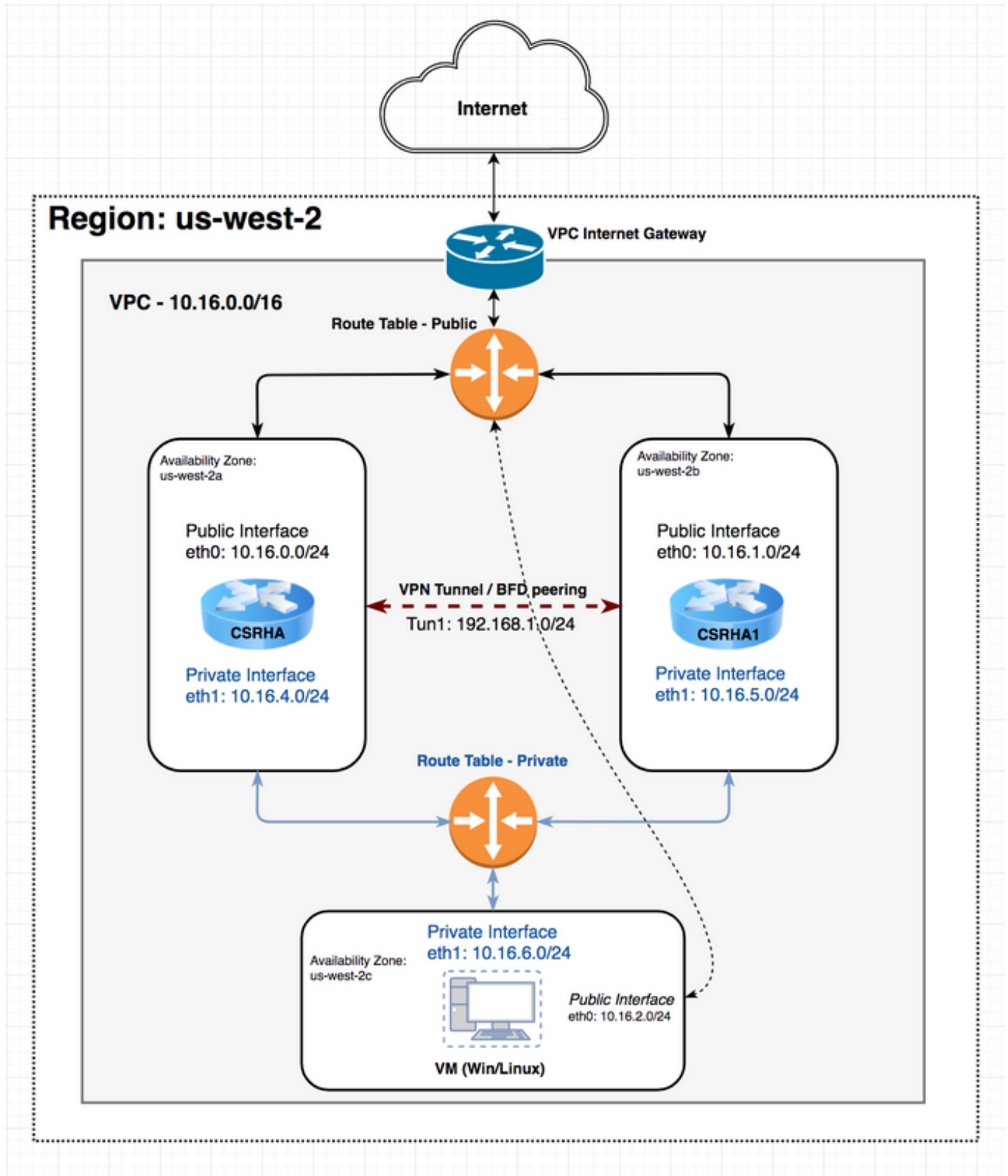
- 1x - 지역
- 1x - VPC
- 3x - 가용 영역
- 6x - 네트워크 인터페이스/서브넷(3x 공용/3x 사설)
- 2x - 경로 테이블(퍼블릭 및 프라이빗)
- 2x - CSR1000v 라우터(Cisco IOS-XE® Denali 16.3.1a 이상)
- 1x - VM(Linux/Windows)

HA 쌍에는 서로 다른 두 가용 영역에 2x CSR1000v 라우터가 있습니다. 각 가용 영역을 별도의 데이터 센터로 간주하여 추가적인 하드웨어 복원력을 제공합니다.

세 번째 영역은 사설 데이터 센터의 디바이스를 시뮬레이션하는 VM입니다. 지금은 VM에 액세스하고 구성할 수 있도록 의 공용 인터페이스를 통해 인터넷 액세스가 활성화됩니다. 일반적으로 모든 일반 트래픽은 전용 경로 테이블을 통과해야 합니다.

트래픽 시뮬레이션을 위해 CSRHA → 8.8.8.8에 → VM의 사설 인터페이스 → 사설 경로 테이블을 ping합니다. 장애 조치 시나리오에서 프라이빗 경로 테이블이 CSRHA1의 프라이빗 인터페이스를 가리키는 경로를 전환했는지 확인합니다.

# 네트워크 다이어그램



## 용어

RTB - 경로 테이블 ID입니다.

CIDR - 경로 테이블에서 업데이트할 경로의 대상 주소입니다.

ENI - 트래픽이 라우팅되는 CSR 1000v 기가비트 인터페이스의 네트워크 인터페이스 ID입니다. 예를 들어, CSRHA가 실패하면 CSRHA1이 AWS 경로 테이블의 경로를 자신의 ENI를 가리키도록 업데이트합니다.

REGION - CSR 1000v의 AWS 리전입니다.

## 제한 사항

- 프라이빗 서브넷의 경우 IP 주소 10.0.3.0/24을 사용하지 마십시오. 이 주소는 고가용성을 위한 Cisco CSR 1000v에서 내부적으로 사용됩니다. Cisco CSR 1000v는 AWS 경로 테이블을 변경하는 REST API 호출을 수행하기 위해 공용 인터넷 액세스 권한이 있어야 합니다.
- CSR1000v의 gig1 인터페이스를 VRF에 넣지 마십시오. HA는 달리 작동하지 않습니다.

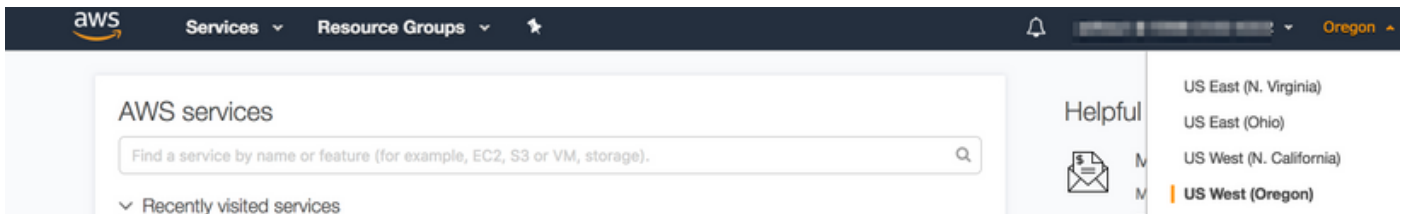
## 설정

컨피그레이션의 일반적인 흐름은 가장 포괄적인 기능(Region/VPC)에서 시작하여 가장 구체적인 기능(Interface/subnet)으로 내려가는 것입니다. 그러나 구체적인 구성 순서는 없습니다. 시작하기 전에 먼저 토폴로지를 파악하는 것이 중요합니다.

팁: 모든 설정(VPC, 인터페이스, 서브넷, 경로 테이블 등)에 이름을 지정합니다.

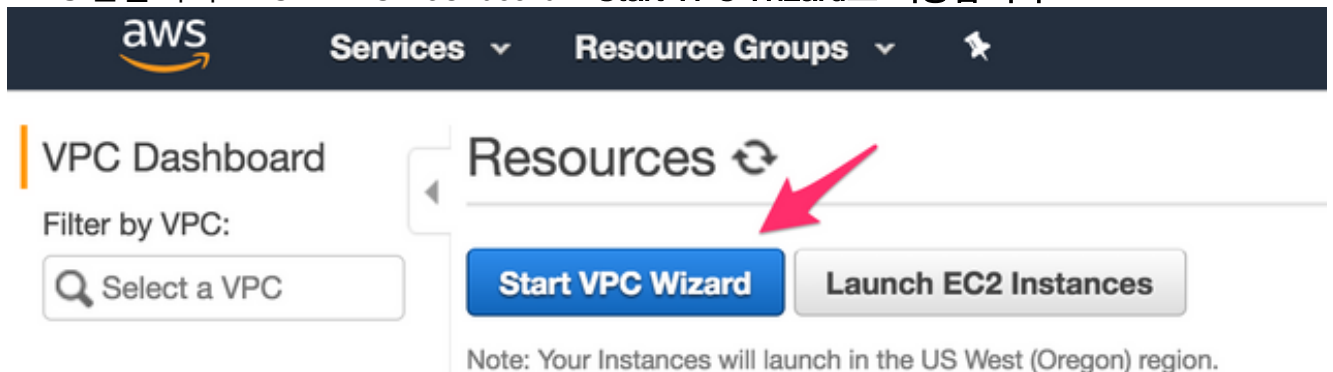
### 1단계. 지역을 선택합니다.

이 예에서는 미국 서부(오레곤)를 사용합니다.



### 2단계. VPC를 생성합니다.

1. AWS 콘솔에서 VPC > VPC Dashboard > Start VPC Wizard로 이동합니다.



2. 단일 퍼블릭 서브넷이 있는 VPC를 선택합니다.

### Step 1: Select a VPC Configuration

**VPC with a Single Public Subnet**

Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

**Creates:**

A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

**Select**

Internet, S3, DynamoDB, SNS, SQS, etc.

Public Subnet

Amazon Virtual Private Cloud

3. VPC를 생성할 때 원하는 대로 사용할 /16 네트워크가 할당됩니다.

4. /24 공용 서브넷도 할당됩니다. 공용 서브넷 인스턴스는 디바이스에 대한 Elastic IP 또는 Public IP를 사용하여 인터넷에 액세스합니다.

Step 2: VPC with a Single Public Subnet

IPv4 CIDR block\*: 10.16.0.0/16 (85531 IP addresses available)

IPv6 CIDR block:  No IPv6 CIDR Block  
 Amazon provided IPv6 CIDR block

VPC name: HA

Public subnet's IPv4 CIDR\*: 10.16.0.0/24 (251 IP addresses available)

Availability Zone\*: No Preference

Subnet name: Public subnet

You can add more subnets after AWS creates the VPC.

Service endpoints

Enable DNS hostnames\*:  Yes  No

Hardware tenancy\*: Default

5. vpc-b98d8ec0이 생성됩니다.

VPC Dashboard

Filter by VPC:

Virtual Private Cloud

Your VPCs

<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR
<input type="checkbox"/>	HA	vpc-b98d8ec0	available	10.16.0.0/16

### 3단계. VPC에 대한 보안 그룹을 생성합니다.

보안 그룹은 트래픽을 허용하거나 거부하는 ACL과 같습니다.

1. Security(보안)에서 **Security Groups(보안 그룹)**를 클릭하고 위에서 생성한 HA라는 VPC와 연결된 보안 그룹을 생성합니다.



2. Inbound Rules(인바운드 규칙)에서 sg-1cf47d6d에 대해 허용할 트래픽을 정의합니다. 이 예에서는 모든 트래픽을 허용합니다.

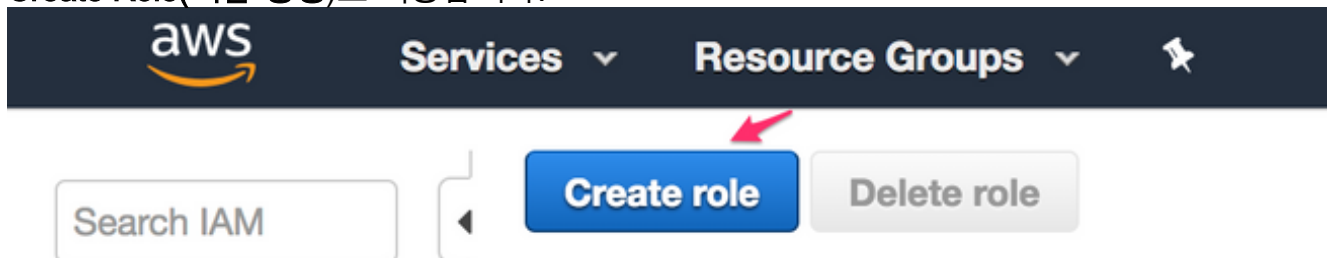


#### 4단계. IAM 역할을 정책과 생성하고 VPC에 연결합니다.

IAM은 Amazon API에 대한 CSR 액세스 권한을 부여합니다.

CSR1000v는 AWS API 명령을 호출하여 경로 테이블을 수정하는 프록시로 사용됩니다. 기본적으로 AMI는 API에 액세스할 수 없습니다. 이 절차에서는 IAM 역할을 생성하며, 이 역할은 CSR 인스턴스를 시작하는 동안 사용됩니다. IAM은 CSR이 AWS API를 사용하고 수정할 수 있는 액세스 자격 증명을 제공합니다.


1. IAM 역할을 생성합니다. 이미지에 표시된 대로 IAM 대시보드로 이동하고 Roles(역할) > Create Role(역할 생성)로 이동합니다.



2. 이미지에 표시된 것처럼 EC2 인스턴스가 사용자를 대신하여 AWS를 호출할 수 있도록 허용합니다.

## Create role

### Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

Allows AWS services to perform actions on your behalf. [Learn more](#)

### Choose the service that will use this role

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

3. 역할을 만들고 다음을 클릭합니다. 이미지에 표시된 대로 검토합니다.

## Create role

1 2 3

### Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#) [Refresh](#)

Filter: Policy type  Showing 394 results

	Policy name	Attachments	Description
<input type="checkbox"/>	AdministratorAccess	7	Provides full access to AWS services and resources.
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	0	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessFullAccess	0	Grants full access to AlexaForBusiness resources and acces...
<input type="checkbox"/>	AlexaForBusinessGatewayExecution	0	Provide gateway execution access to AlexaForBusiness serv...
<input type="checkbox"/>	AlexaForBusinessReadOnlyAccess	0	Provide read only access to AlexaForBusiness services
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	0	Provides full access to create/edit/delete APIs in Amazon AP...
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	0	Allows API Gateway to push logs to user's account.
<input type="checkbox"/>	AmazonAppStreamFullAccess	0	Provides full access to Amazon AppStream via the AWS Ma...
<input type="checkbox"/>	AmazonAppStreamReadOnlyAccess	0	Provides read only access to Amazon AppStream via the AW...
<input type="checkbox"/>	AmazonAppStreamServiceAccess	0	Default policy for Amazon AppStream service role.
<input type="checkbox"/>	AmazonAthenaFullAccess	0	Provide full access to Amazon Athena and scoped access to...

\* Required

[Cancel](#)

[Previous](#)

[Next: Review](#)

4. 역할 이름을 지정합니다. 이 예에서 그림과 같이 역할 이름은 routetablechange입니다.

## Create role

### Review

Provide the required information below and review this role before you create it.

Role name\*

routetablechange

Use alphanumeric and '+,=,@,-' characters. Maximum 64 characters.

5. 그런 다음 정책을 생성하고 위에서 생성한 역할에 연결해야 합니다. IAM 대시보드를 선택하고 Policies(정책) > Create Policy(정책 생성)로 이동합니다.



Search IAM Create policy Policy actions

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

Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

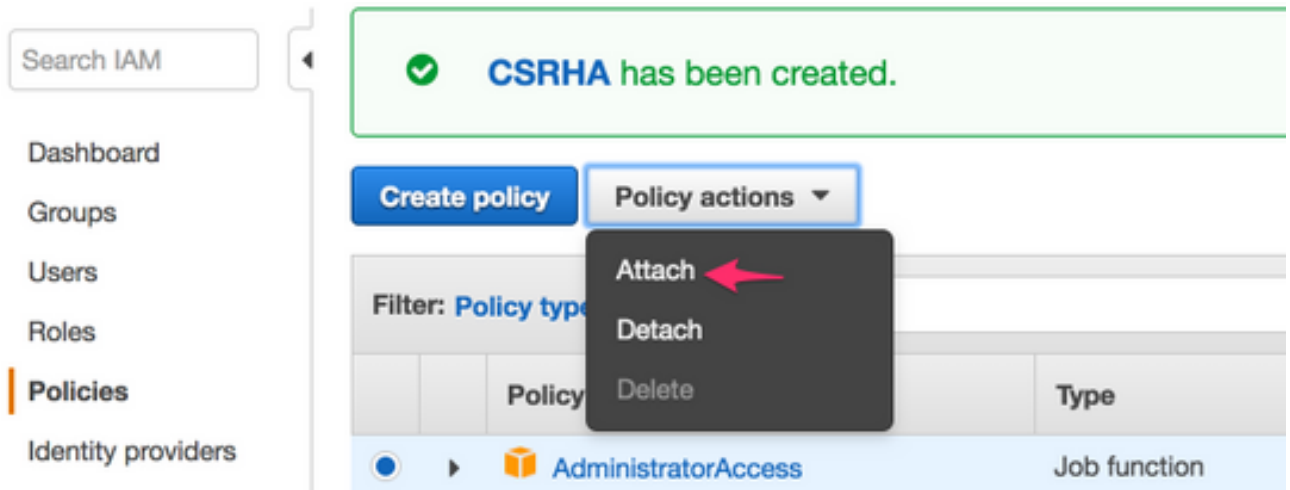
This policy validation failed and might have errors converting to JSON : The policy must have at least one statement For more information about the IAM policy grammar, see AWS IAM Policies

Visual editor JSON Import managed policy

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "ec2:AssociateRouteTable",
8-         "ec2:CreateRoute",
9-         "ec2:CreateRouteTable",
10-        "ec2>DeleteRoute",
11-        "ec2>DeleteRouteTable",
12-        "ec2:DescribeRouteTables",
13-        "ec2:DescribeVpcs",
14-        "ec2:ReplaceRoute",
15-        "ec2:DisassociateRouteTable".
```

6. 정책 이름을 지정하고 생성한 역할에 연결합니다. 이 예에서 정책 이름은 이미지에 표시된 대로 관리자 액세스 권한이 있는 CSRHA입니다.

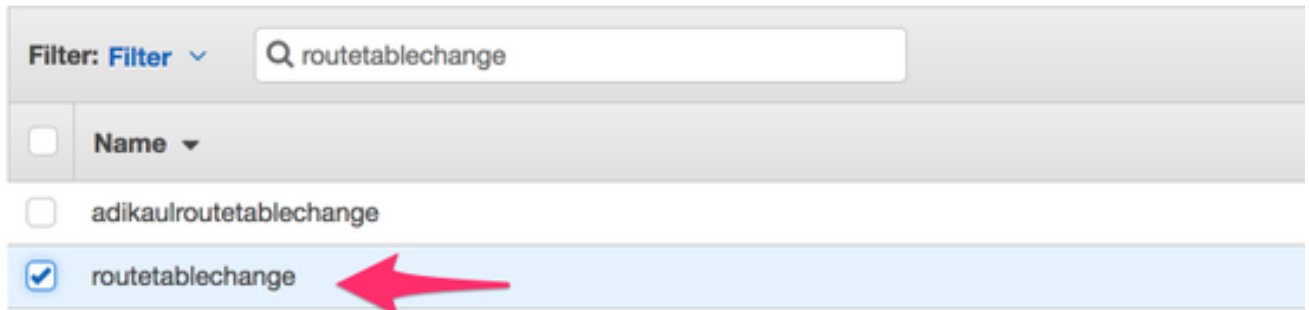




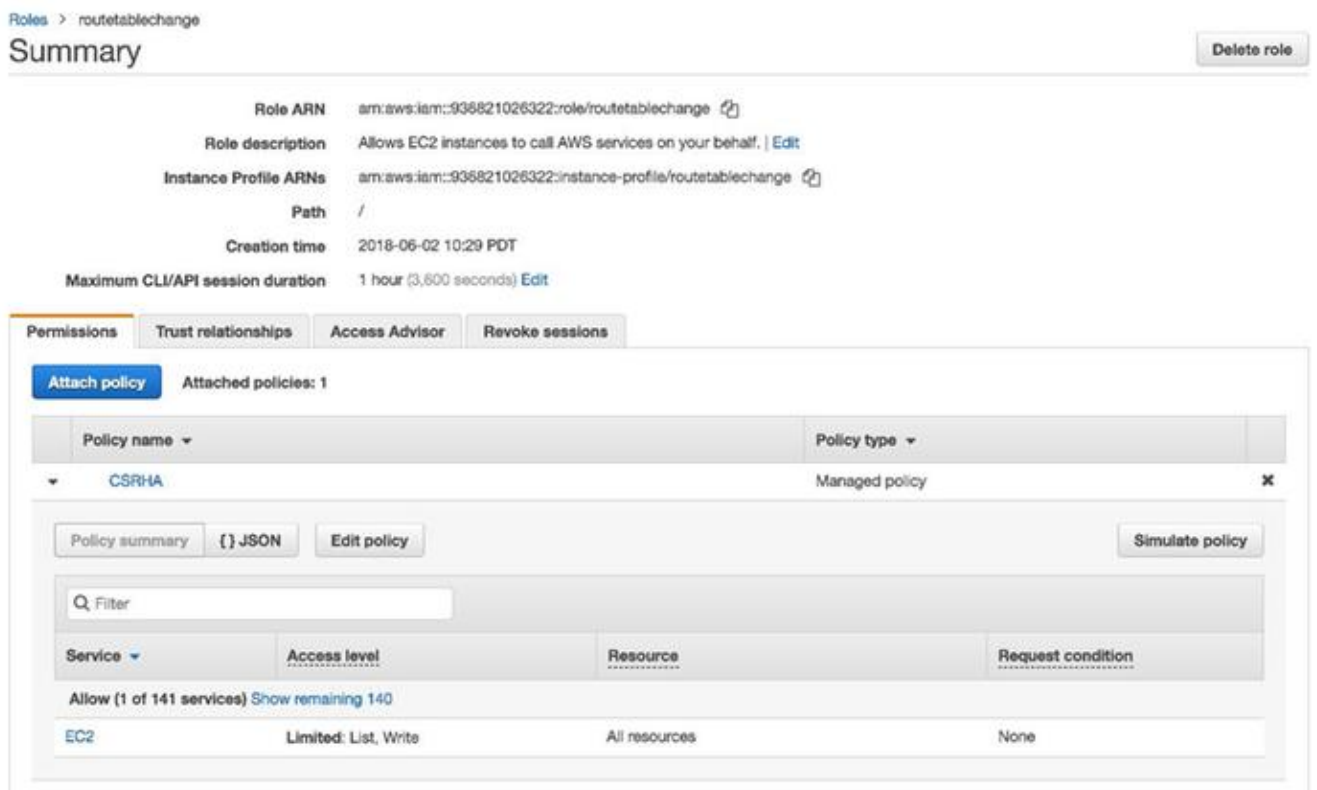
7. 그림에 표시된 것처럼, 사용자가 생성한 역할에 정책을 routetablechange라고 연결합니다.

## Attach Policy

Attach the policy to users, groups, or roles in your account.



8. 요약.



5단계. 생성한 AMI 역할로 CSR1000v를 시작하고 퍼블릭/프라이빗 서브넷을 연결합니다.

각 CSR1000v 라우터에는 2개의 인터페이스(1개의 공용, 1개의 사설)가 있으며 자체 가용 영역에 있습니다. 각 CSR을 별도의 데이터 센터에 있는 것으로 생각할 수 있습니다.

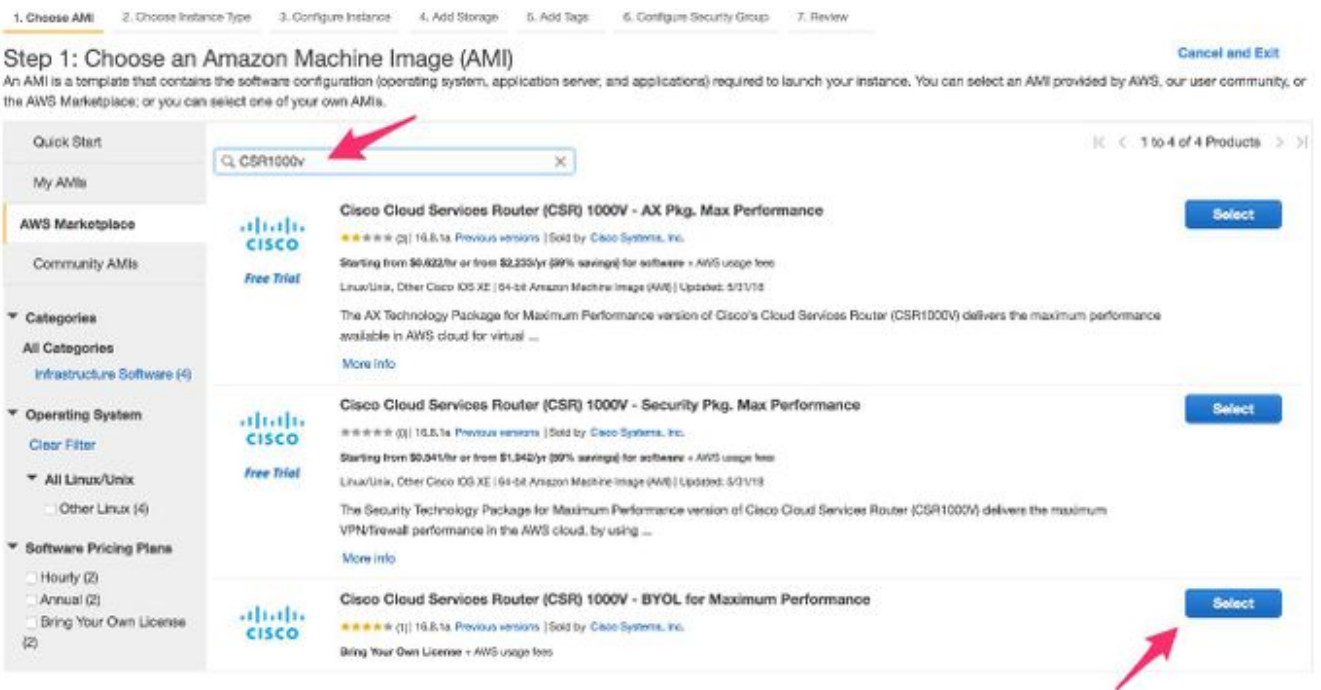
1. AWS 콘솔에서 EC2를 선택한 다음 인스턴스 시작을 클릭합니다.



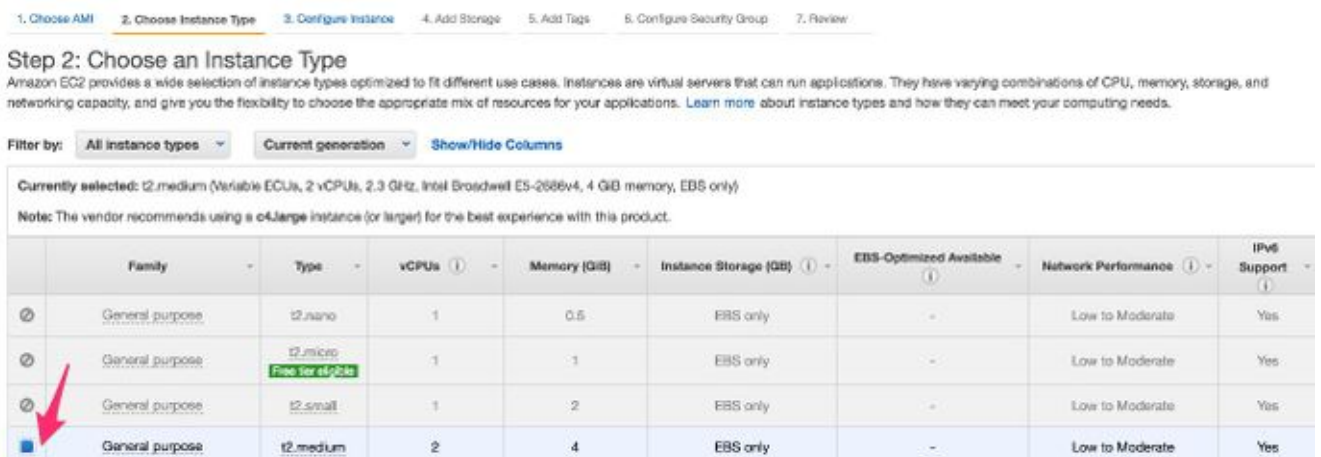
2. AWS Marketplace를 선택합니다.



3. CSR1000v를 입력합니다. 이 예에서는 최대 성능을 위해 Cisco CSR(Cloud Services Router) 1000V - BYOL을 사용합니다.



4. 인스턴스 유형을 선택합니다. 이 예에서 선택한 유형은 t2.medium입니다.



5. 인스턴스가 구성된 동안에는 위에서 생성한 VPC를 위의 IAM 역할과 함께 선택해야 합니다. 또한 프라이빗 페이싱 인터페이스에 연결하는 프라이빗 서브넷을 생성합니다.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

No default VPC found. Select another VPC, or create a new default VPC.

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option:  Request Spot instances

Network: vpc-a6fefedf | HA [Create new VPC](#)  
No default VPC found. [Create a new default VPC.](#)

Subnet: subnet-66f7931f | Public subnet | us-west-2a [Create new subnet](#)  
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

Placement group:  Add instance to placement group.

IAM role: routetablechange [Create new IAM role](#)

Shutdown behavior: Stop

Enable termination protection:  Protect against accidental termination

Monitoring:  Enable CloudWatch detailed monitoring  
Additional charges apply.

6. Create new Subnet for Private Subnet을 클릭합니다. 이 예에서 Name 태그는 HA Private입니다. 공용 서브넷과 동일한 가용 영역에 있는지 확인합니다.

### Create Subnet

Use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Note that block sizes must be between a /16 netmask and /28 netmask. Also, note that a subnet can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: HA Private

VPC: vpc-a6fefedf | HA

VPC CIDRs	CIDR	Status	Status Reason
	10.16.0.0/16	● associated	

Availability Zone: us-west-2a

IPv4 CIDR block: 10.16.4.0/24

[Cancel](#) [Yes, Create](#)

7. 아래로 스크롤하여 Configure Instance Details(인스턴스 세부사항 구성)에서 이미지에 표시된 대로 Add Device(디바이스 추가)를 클릭합니다.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Network interfaces

Device	Network interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-66f7931f	Auto-assign	<a href="#">Add IP</a>	

[Add Device](#)

8. 보조 인터페이스를 추가한 후 HA Private이라는 프라이빗 서브넷을 연결합니다. Eth0은 공용

인터페이스이고 Eth1은 사실 인터페이스입니다. **참고:** 이전 단계에서 만든 서브넷은 이 드롭 다운에 표시되지 않을 수 있습니다. 페이지를 새로 고치거나 취소하고 서브넷을 표시하려면 다시 시작해야 할 수 있습니다



9. VPC에서 생성한 보안 그룹을 선택하고 규칙이 올바르게 정의되었는지 확인합니다.

- 1. Choose AMI
- 2. Choose Instance Type
- 3. Configure Instance
- 4. Add Storage
- 5. Add Tags
- 6. Configure Security Group
- 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-01880170	default	default VPC security group	Copy to new
<input checked="" type="checkbox"/> sg-1cf47d6d	HA	HA	Copy to new

10. 새 키 쌍을 만들고 개인 키를 다운로드합니다. 모든 디바이스에 하나의 키를 재사용할 수 있습니다. **참고:** 개인 키를 분실한 경우 CSR에 다시 로그인할 수 없습니다. 키를 복구하는 방법은 없습니다

- 1. Choose AMI
- 2. Choose Instance Type
- 3. Configure Instance
- 4. Add Storage
- 5. Add Tags
- 6. Configure Security Group
- 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.



## Select an existing key pair or create a new key pair



A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

**Key pair name**  
CSRHA

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

11. Elastic IP를 생성한 인스턴스의 Public Interface의 ENI와 연결하고 **AWS Console > EC2 Management > Network Security > Elastic IP**로 이동합니다. 참고: 공공/민간 용어가 여기에서 혼란을 일으킬 수 있습니다. 이 예에서 공용 인터페이스의 정의는 인터넷 연결 인터페이스인 Eth0입니다. AWS의 관점에서 볼 때 Cisco의 공용 인터페이스는 프라이빗 IP입니다

EC2 Dashboard

Allocate new address

Events

Addresses > Associate address

Associate address

Select the instance OR network interface to which you want to associate this Elastic IP address (54.244.108.43)

Resource type  Instance  Network interface

Network interface eni-2515633d

Private IP 10.16.2.215

Reassociation  Allow Elastic IP to be reassociated if already attached

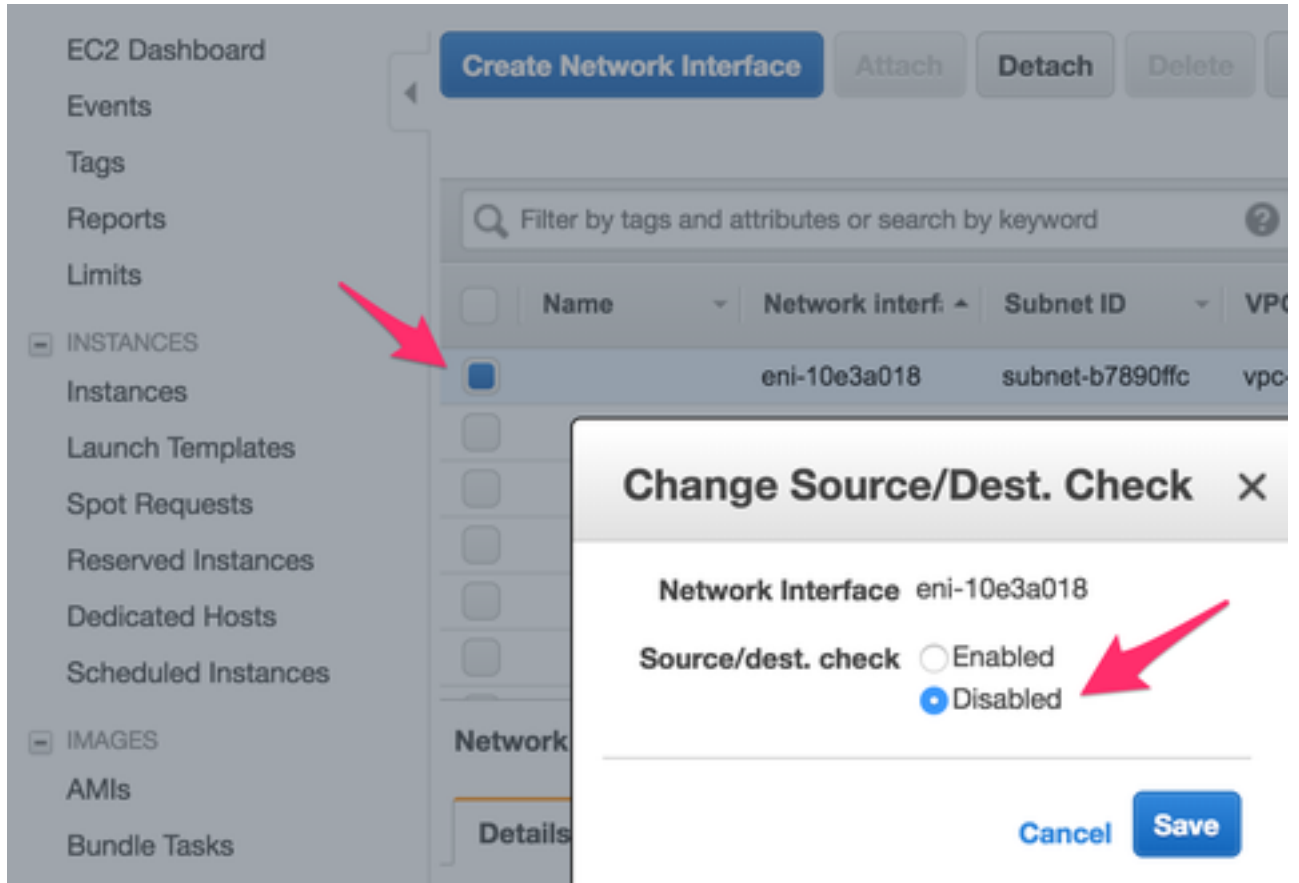
Warning  
If you associate an Elastic IP address with your instance, your current public IP address is released. Learn more.

AWS Command Line Interface command

Cancel Associate

12. **EC2 > 네트워크 인터페이스**로 이동할 때 **Source/Dest Check**를 비활성화합니다. Source/Dest(소스/대상) 확인에 대한 각 ENI를 확인합니다. 기본적으로 모든 ENI에는 이 Source/Dest(소스/대상) 검사가 활성화되어 있습니다. 스푸핑 방지 기능은 ENI가 트래픽을 전달하기 전에 트래픽의 대상인지 확인하여 ENI에 적합하지 않은 트래픽이 과도하게 발생하

는 것을 방지하기 위한 것입니다. 라우터가 패킷의 실제 대상이 되는 경우는 거의 없습니다. 이 기능은 모든 CSR 전송 ENI에서 비활성화해야 합니다. 그렇지 않으면 패킷을 전달할 수 없습니다



13. CSR1000v에 연결합니다. **참고:** CSR1000v에 SSH하기 위해 AWS에서 제공한 사용자 이름이 root로 잘못 나열될 수 있습니다. 필요한 경우 이를 ec2-user로 변경합니다. **참고:** SSH에 DNS 주소를 ping할 수 있어야 합니다. ec2-54-208-234-64.compute-1.amazonaws.com입니다. 라우터의 공용 서브넷/eni가 공용 경로 테이블과 연결되어 있는지 확인합니다. 라우트 테이블에 서브넷을 연결하는 방법에 대한 8단계를 간단히 진행합니다

## Connect To Your Instance



I would like to connect with

A standalone SSH client

A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (HA.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 HA.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-54-208-234-64.compute-1.amazonaws.com
```

Example:

```
ssh -i "HA.pem" root@ec2-54-208-234-64.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

**6단계. 5단계를 반복하고 HA에 대해 두 번째 CSR1000v 인스턴스를 생성합니다.**

공용 서브넷: 10.16.1.0/24

개인 서브넷: 10.16.5.0/24

이 새 AMI의 탄력적 IP 주소를 ping할 수 없는 경우 8단계로 이동하여 공용 서브넷이 공용 경로 테이블과 연결되어 있는지 확인합니다.

**7단계. 5단계를 반복하여 AMI Marketplace에서 VM(Linux/Windows)을 생성합니다.**

이 예에서는 마켓플레이스에서 Ubuntu Server 14.04 LTS를 사용합니다.

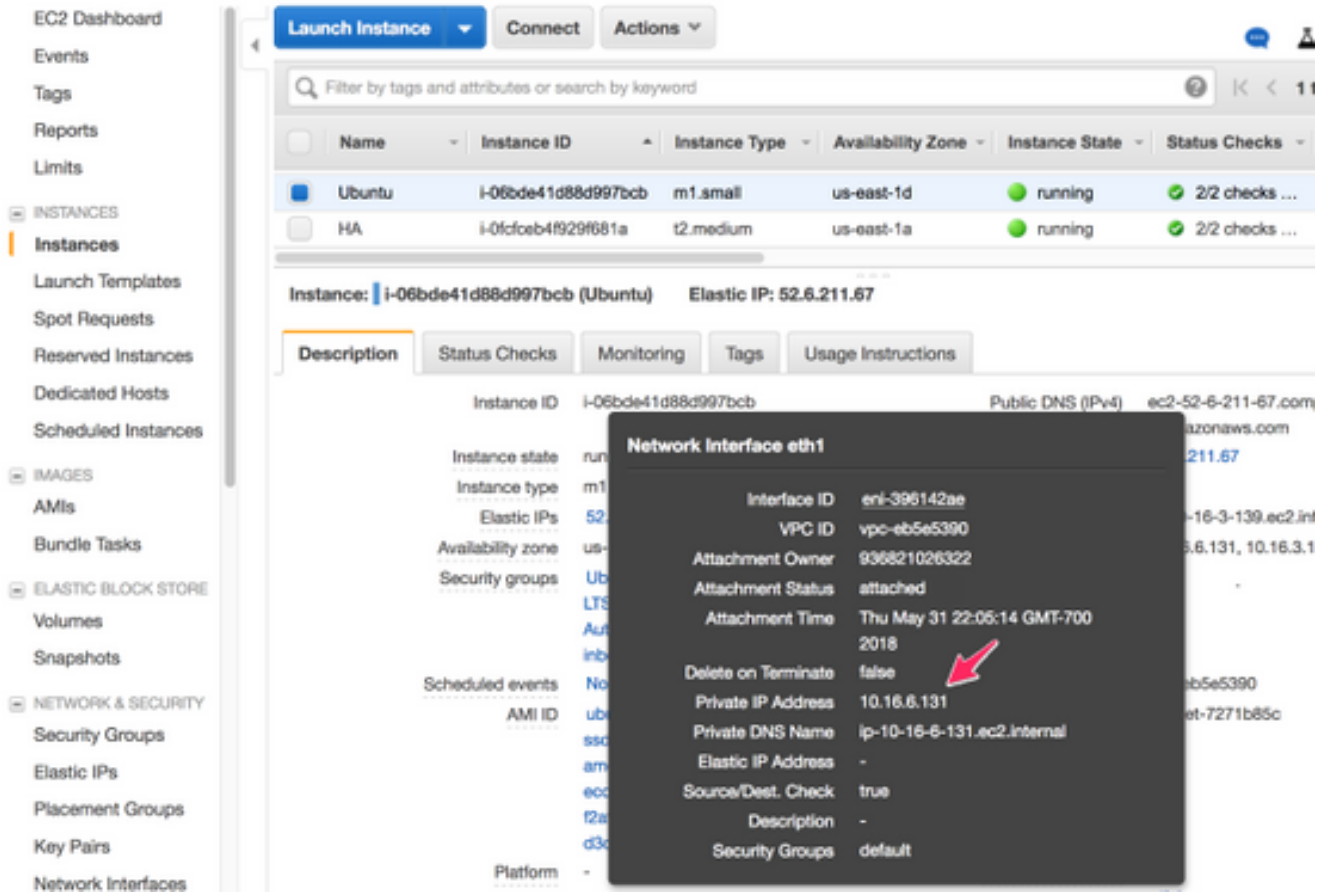
공용 서브넷: 10.16.2.0/24

개인 서브넷: 10.16.6.0/24

이 새 AMI의 탄력적 IP 주소를 ping할 수 없는 경우 8단계로 이동하여 공용 서브넷이 공용 경로 테이블과 연결되어 있는지 확인합니다.



1. Eth0은 기본적으로 공용 인터페이스에 대해 생성됩니다. 사설 서브넷에 대해 eth1이라는 두 번째 인터페이스를 만듭니다.



2. Ubuntu에서 구성하는 IP 주소는 AWS에서 할당한 eth1 전용 인터페이스입니다.

```
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. 인터페이스를 펼치거나 VM을 재부팅합니다.

```
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. 테스트의 경우 Ping 8.8.8.8. 7단계별로 8.8.8.8 경로가 추가되었는지 확인합니다.

```
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
```

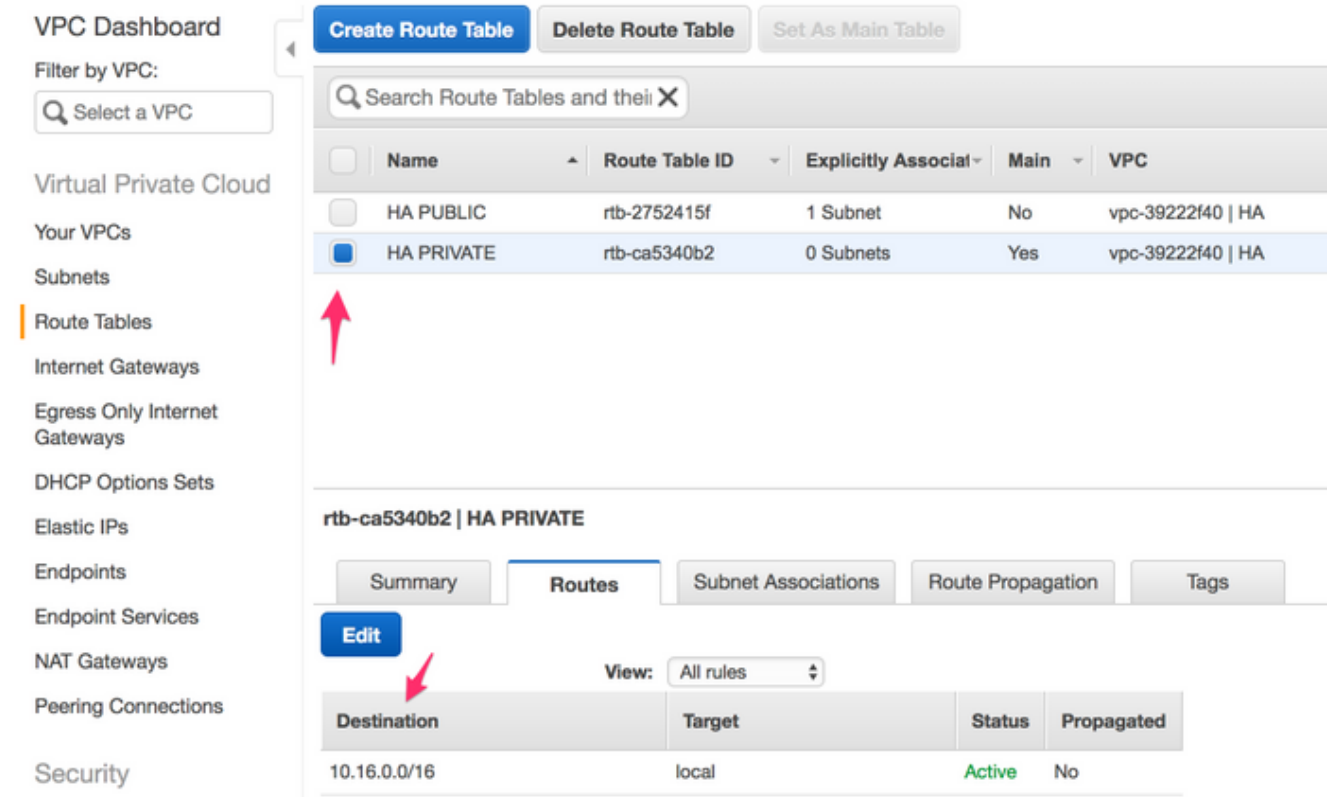
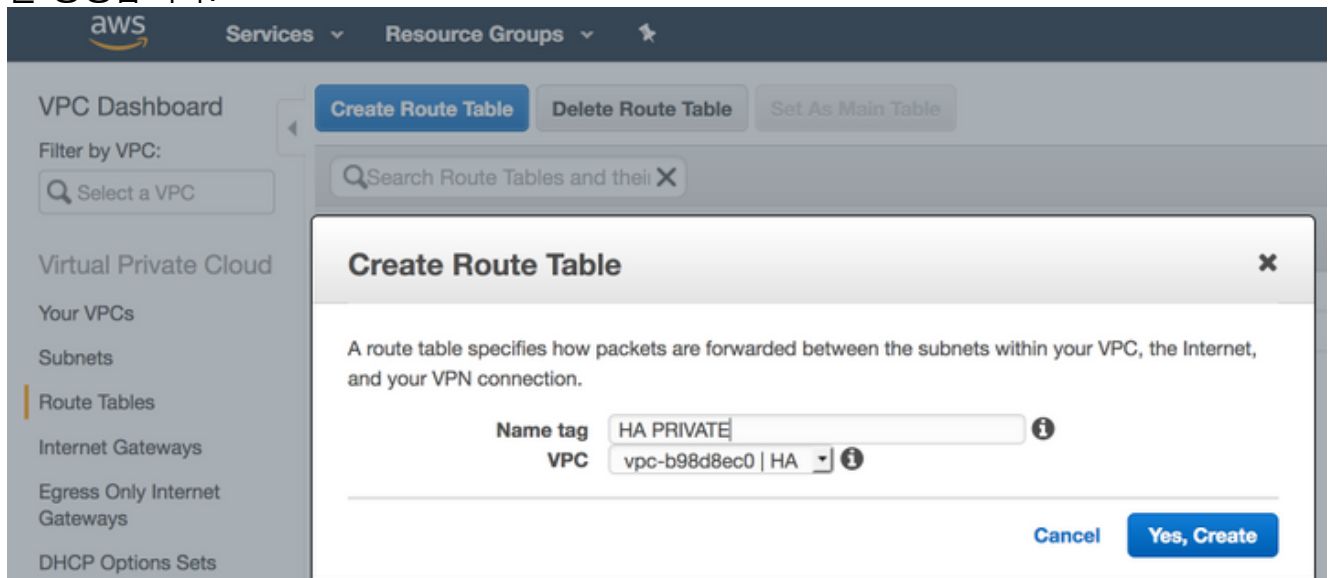
8.8.8.8이 표에 나열되지 않으면 다음을 수동으로 추가합니다.

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

## 8단계. 프라이빗 및 퍼블릭 경로 테이블을 구성합니다.

1. 2단계에서 마법사를 통한 VPC가 생성되면 두 개의 경로 테이블이 자동으로 생성됩니다. 하나의 경로 테이블만 있는 경우 이미지에 표시된 대로 프라이빗 서브넷에 대해 다른 경로 테이블

을 생성합니다.



2. 다음은 두 개의 경로 테이블의 보기입니다. PUBLIC Route Table(공용 경로 테이블)에는 인터넷 게이트웨이(igw-95377973)가 자동으로 연결됩니다. 이 두 테이블에 적절하게 레이블을 지정합니다. PRIVATE 테이블에는 이 경로가 없어야 합니다.

## VPC Dashboard

Filter by VPC:

Select a VPC

## Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Create Route Table

Delete Route Table

Set As Main Table

Search Route Tables and their

<input type="checkbox"/>	Name	Route Table ID	Explicitly Associat	Main	VPC
<input checked="" type="checkbox"/>	HA PUBLIC	rtb-2752415f	1 Subnet	No	vpc-39222f40   HA
<input type="checkbox"/>	HA PRIVATE	rtb-ca5340b2	0 Subnets	Yes	vpc-39222f40   HA

### rtb-2752415f | HA PUBLIC

Summary

Routes

Subnet Associations

Route Propagation

Tags

Edit

View: All rules

Destination	Target	Status	Propagated
10.16.0.0/16	local	Active	No
0.0.0.0/0	igw-953779f3	Active	No

3. 6개의 모든 서브넷을 적절한 경로 테이블에 연결 3 공용 인터페이스는 공용 경로 테이블과 연결됩니다. 공용 서브넷: 10.16.0.0/24, 10.16.1.0/24, 10.16.2.0/24 3 프라이빗 인터페이스는 프라이빗 경로 테이블과 연결됩니다. 개인 서브넷: 10.16.4.0/24, 10.16.5.0/24, 10.16.6.0/24

### rtb-ec081d94 | HA PRIVATE

Summary

Routes

Subnet Associations

Route Propagation

Tags

Edit

Subnet IPv4 CIDR IPv6 CIDR

You do not have any subnet associations.

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

## 9단계. BFD를 사용하여 NAT(Network Address Translation) 및 GRE 터널 및 모든 라우팅 프로토콜을 구성합니다.

CSR 1000v의 Elastic IP를 통해 GRE(Generic Routing Encapsulation) 터널을 구성합니다(잘못된 장애를 탐지하는 DHCP 임대 갱신 문제를 방지하기 위해 권장됨). 더 빠른 컨버전스가 필요한 경우 BFD(Bidirectional Forwarding Detection) 값을 이 예에 표시된 값보다 더 공격적으로 구성할 수 있습니다. 그러나 간헐적 연결 중에 BFD 피어 다운 이벤트가 발생할 수 있습니다. 이 예의 값은 1.5초 내에 피어 장애를 탐지합니다. AWS API 명령이 실행되는 시간과 VPC 라우팅 테이블 변경 사항이 적용되는 시간 사이에는 몇 초 정도의 가변 지연이 있습니다.

- CSRHA의 컨피그레이션  
GRE 및 BFD - HA 장애 조치 조건을 관찰하는 데 사용됩니다.

```

interface Tunnell
  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 Tunnell
  network 192.168.1.0
  passive-interface GigabitEthernet1

```

**NAT 및 라우팅 - 프라이빗 인터페이스를 통한 VM 인터넷 연결에 사용됩니다.**

```

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

```

- CSRHA1의 컨피그레이션

**GRE 및 BFD - HA 장애 조치 조건을 관찰하는 데 사용됩니다.**

```

interface Tunnell
  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 Tunnell
  network 192.168.1.0
  passive-interface GigabitEthernet1

```

**NAT 및 라우팅 - 프라이빗 인터페이스를 통한 VM 인터넷 연결에 사용됩니다.**

```

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
```

## 10단계. 고가용성을 구성합니다(Cisco IOS XE Denali 16.3.1a 이상).

아래에 지정된 cloud provider aws 명령을 사용하여 각 CSR 1000v를 구성하여 BFD 피어 다운 이벤트를 모니터링합니다. BFD 피어 다운과 같은 AWS HA 오류가 감지된 후 (VPC) Route-table-id, Network-interface-id 및 CIDR에 대한 라우팅 변경 사항을 정의하려면 이 명령을 사용합니다.

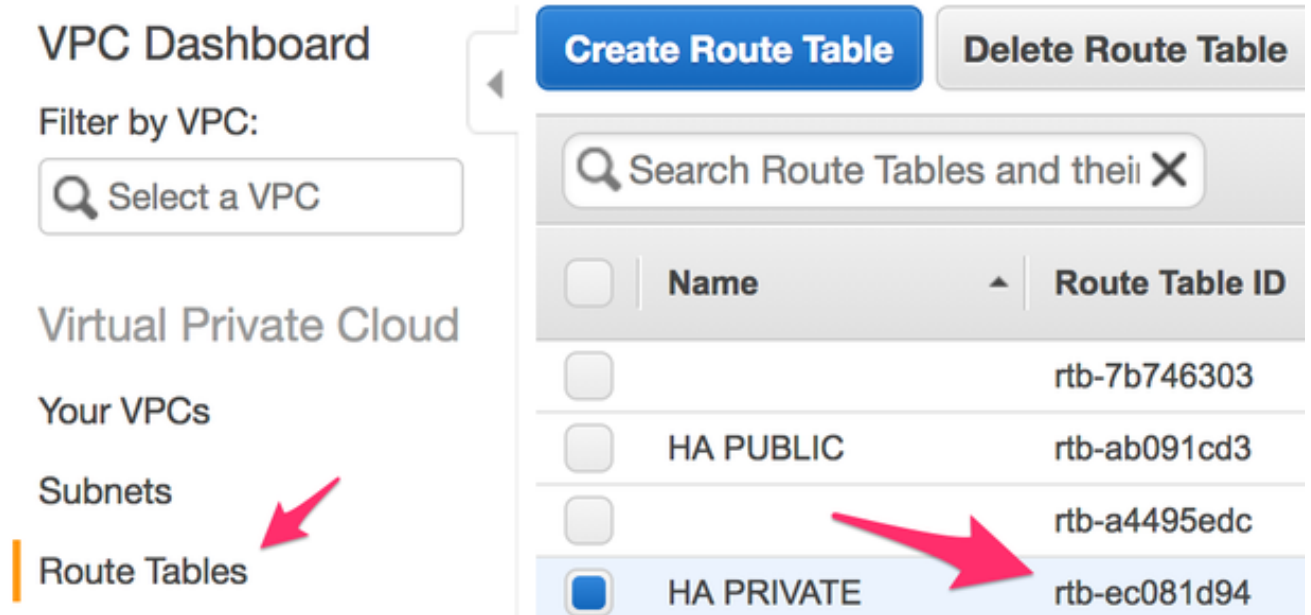
```
CSR(config)# redundancy
CSR(config-red)# cloud provider [aws | azure] node-id
# bfd peer ipaddr
# route-table table-name
# cidr ip ipaddr/prefix
# eni elastic-network-intf-name
# region region-name
```

1. #bfd 피어 ipaddr은 피어 터널 ip 주소입니다.

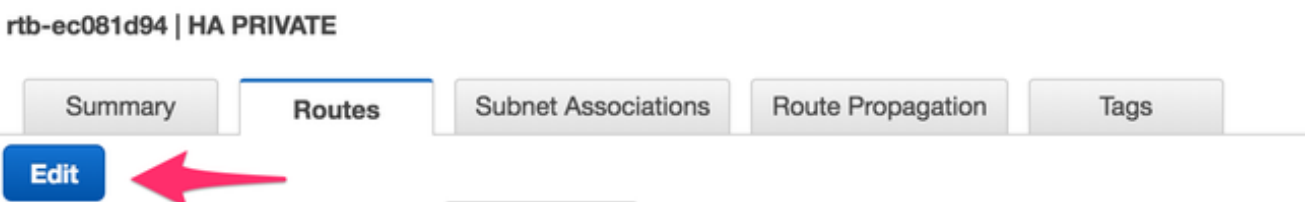
```
CSRHA#show bfd neighbors
```

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

2. #route-table 테이블 이름은 AWS 콘솔에서 VPC > Route Tables로 이동합니다. 이렇게 하면 전용 경로 테이블이 변경됩니다.



3. #cidr ip ipaddr/prefix는 경로 테이블에서 업데이트할 경로의 대상 주소입니다. AWS 콘솔에서 VPC > 경로 테이블로 이동합니다. 아래로 스크롤하여 Edit(수정)를 클릭한 다음 Add another route(다른 경로 추가)를 클릭합니다. 테스트 대상 주소 8.8.8.8과 CSRHA의 비공개 ENI를 추가합니다.



Summary Routes Subnet Associations Route Propagation Tags

Cancel Save

View: All rules

Destination	Target	Status	Propagated	Remove
10.16.0.0/16	local	Active	No	
8.8.8.8/32	eni-10e3a018	Active	No	✕

Add another route

4. #eni elastic-network-intf-name은 EC2 인스턴스에서 찾을 수 있습니다. 해당하는 각 CSR에 대한 Private facing(개인 연결) 인터페이스 eth1을 클릭하고 인터페이스 ID를 사용합니다.

Instances Launch Templates Spot Requests Reserved Instances Dedicated Hosts Scheduled Instances

IMAGES AMIs Bundle Tasks

ELASTIC BLOCK STORE Volumes Snapshots

NETWORK & SECURITY Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

LOAD BALANCING Load Balancers

Instance	AMI	Instance ID	Instance type	Availability zone	Status	Checks
CSRHA	i-0223f5ca1d6068424	i-0223f5ca1d6068424	c4.large	us-west-2a	running	2/2 checks ...
CSRHA1	i-0bed9ff2bd6996ca4	i-0bed9ff2bd6996ca4	t2.medium	us-west-2b	running	2/2 checks ...
WINDOWS	i-07a0fecde36302c6a	i-07a0fecde36302c6a	t2.small	us-west-2c	running	2/2 checks ...

Instance: i-0223f5ca1d6068424 (CSRHA) Elastic Network Interfaces

Interface ID	Private IP Address
eni-90b500a8	10.16.4.198

Network interface eth1

Interface ID: eni-90b500a8  
 VPC ID: vpc-19c1c060  
 Attachment Owner: 936821026322  
 Attachment Status: attached  
 Attachment Time: Thu May 31 21:57:41 GMT-700 2018  
 Delete on Terminate: true  
 Private IP Address: 10.16.4.198  
 Private DNS Name: ip-10-16-4-198.us-west-2.compute.internal  
 Elastic IP Address: -  
 Source/Dest. Check: false  
 Description: -  
 Security Groups: HAKAUL

Network interfaces eth0 eth1

5. #region 이름은 AWS 문서에 있는 코드 이름입니다. 이 목록은 변경되거나 늘어날 수 있습니다. 최신 업데이트를 찾으려면 Amazon의 리전 및 가용 영역 문서를 방문하십시오.



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)

## CSRHA의 이중화 컨피그레이션 예

```

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

```

## CSRHA1의 이중화 컨피그레이션 예

```

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

```



# 고가용성 확인

## 1. BFD 및 클라우드 컨피그레이션을 확인합니다.

```
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. VM에서 대상으로 지속적인 ping을 실행합니다. ping이 전용 eth1 인터페이스를 통과하는지 확인합니다.

```
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. 프라이빗 경로 테이블을 확인합니다. eni는 현재 CSRHA의 프라이빗 인터페이스이며, 여기서 트래픽이 이 인터페이스입니다.

rtb-ec081d94 | HA PRIVATE

Summary Routes Subnet Associations Route Propagation Tags

Edit

View: All rules

Destination	Target	Status	Propagated
10.16.0.0/16	local	Active	No
8.8.8.8/32	eni-90b500a8 / i-0fcfceb4f929f681a	Active	No

## 4. HA 장애 조치를 시뮬레이션하기 위해 CSRHA의 Tunnel1을 종료합니다.

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

## 5. 경로 테이블이 CSRHA1의 전용 인터페이스인 새 ENI를 가리키는 지 확인합니다.

Summary

Routes

Subnet Associations

Route Propagation

Tags

Edit

View: All rules

Destination	Target	Status	Propagated
10.16.0.0/16	local	Active	No
8.8.8.8/32	eni-10e3a018 / i-0fcfcecb4f929f681a	Active	No

## 문제 해결

- 리소스가 연결되었는지 확인합니다. VPC, 서브넷, 인터페이스, 경로 테이블 등을 생성할 때 이들 중 상당수는 자동으로 서로 연결되지 않습니다. 그들은 서로에 대해 전혀 모른다.
- Elastic IP 및 모든 Private IP가 올바른 인터페이스와 연결되어 있는지 확인합니다. 올바른 서브넷이 올바른 경로 테이블에 추가되고 올바른 라우터와 올바른 VPC 및 영역에 연결되며 IAM 역할 및 보안 그룹과 연결되어 있는지 확인합니다.
- ENI별로 Source/Dest 검사를 비활성화합니다.
- Cisco IOS XE 16.3.1a 이상의 경우 사용 가능한 추가 확인 명령입니다.

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

- 디버깅에 나타나는 일반적인 실패는 다음과 같습니다.

### 문제: httpc\_send\_request 실패

해결 방법: Http는 CSR에서 AWS로 API 호출을 전송하는 데 사용됩니다. DNS가 인스턴스에 나열된 DNS 이름을 확인할 수 있는지 확인합니다. http 트래픽이 차단되지 않았는지 확인합니다.

```
*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
```

### 문제: 경로 테이블 rtb-9c0000f4와 인터페이스 eni-32791318이 다른 네트워크에 속함

해결 방법: 다른 네트워크에서 지역 이름 및 ENI가 잘못 구성되었습니다. 지역 및 ENI는 라우터와 동일한 영역에 있어야 합니다.

```
*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>
```

**문제/장애: 이 작업을 수행할 권한이 없습니다. 인코딩된 권한 부여 실패 메시지입니다.**

**해결 방법: IAM JSON 역할/정책이 잘못 생성되었거나 CSR에 적용되지 않았습니다. IAM 역할은 CSR이 API 호출을 하도록 권한을 부여합니다.**

```
*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: qYvEB4MUdOB8m2itSteRgnOuslAaxhAbDph5qGRJk jJbrESajbmF5HWUR-MmHYeRALpKZ3Jg_y-_tMlYe15l_ws8Jd9q2W8YDXB13uXQqfW_cjjrgy9jhnGY0nOaNu65aLpfqui8kS_4RPOpm5grRFfo99-8uv_N3mYaBqKFPn3vUcSYKBmxFIikJKc jY9esOeLIOWDcnYGGu6AGGMoMxWDtk0K8nwk4IjLDcnd2cDXeENS45w1PqzKGPshv3wD28TS5xRjIrPXyRt18UpV6lLA_09Oh4737VncQKfzbz4tPpnAkoW0mJLQ1vDpPmNvHUpEng8KrGWYNfbfemoDtWqIdABfaLLm4saNtnQ_OMB0Ti4toBLEb2BNdMkl1UVBIxqTqdFUVRS**MSG 00041 TRUNCATED** **MSG 00041 CONTINUATION
#01**qLosAb5Yx0DrOsLSQwzS95VGvQM_n87LBHYbAWWhqWj3UfP_zmiak7d1m9P41mFCucEB3Cs4FRsFtb-9q44VtyQJaS2sU2nhGe3x4uGEsl7F1pNv5vhVeYOZB3tbOfbV1_Y4trZwYPPfGLKgBShZp-WNmUKUJsKc1-6KGqmp7519imvh66JgwgU9DT_qAZ-jEjkqWjBrxg6krw</Message></Error></Errors><RequestID>4cf31249-2a6e-4414-ae8d-6fb825b0f398</RequestID></Response>
```

## 관련 정보

- [VPC 게이트웨이 이중화 - Cisco](#)
- [Amazon Web Services용 Cisco CSR 1000v Series 클라우드 서비스 라우터 구축 설명서](#)
- [인스턴스 유형 분류](#)
- [EC2 및 VPC](#)
- [EC2 사용 설명서의 Elastic Network Interfaces에는 인스턴스 유형당 ENI 개수가 포함됩니다.](#)
- [Linux에서 향상된 네트워킹 방법, 유용한 배경 정보](#)
- [전용 인스턴스/테넌시 설명 및 방법](#)
- [일반 EC2 설명서](#)
- [일반 VPC 설명서](#)
- [지역 및 가용 영역](#)
- [CSR1000v 고가용성 버전 3](#)

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.