

AzureCLI 2.0을 사용하는 Microsoft Azure에 대한 CSR1000v HA 리던던시 배포 가이드

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[목표](#)

[토폴로지](#)

[네트워크 다이어그램](#)

[용어](#)

[제한 사항](#)

[설정](#)

[개요](#)

[1단계. AzureCLI 2.0을 설치합니다.](#)

[2단계. 자원 그룹을 생성합니다.](#)

[3단계. Vnet을 생성합니다.](#)

[4단계. 경로 테이블을 생성합니다.](#)

[5단계. 서브넷을 생성합니다.](#)

[6단계. CSR1000v 라우터를 생성합니다.](#)

-

[7단계. 두 번째 CSR1000v 라우터를 생성합니다.](#)

[8단계. 6단계에서 동일한 절차로 호스트 VM을 생성합니다. 이 예에서는 UbuntuLTS를 사용합니다.](#)

[9단계. 라우팅 테이블 및 VM에 경로를 추가합니다.](#)

[10단계. CSR1000v 라우터를 구성합니다.](#)

[고가용성 확인](#)

[문제 해결](#)

[관련 정보](#)

소개

이 문서에서는 AzureCLI 2.0을 사용하는 Microsoft Azure 클라우드에서 고가용성을 위한 CSR1000v 라우터를 배포하는 방법에 대한 단계별 구성 가이드를 제공합니다. 사용자에게 HA에 대한 실용적인 지식과 완전한 기능을 갖춘 테스트베드를 구축할 수 있는 기능을 제공하기 위한 것입니다.

Azure에 이미지를 배포하는 방법은 다양하며 대부분의 사용자에게 가장 익숙한 방법은 웹 포털을 통해 제공됩니다. 그러나 AzureCLI는 익숙해지면 빠르고 강력한 도구입니다.

Azure, 웹 포털을 통해 CSR1000v를 배포하는 방법 및 HA에 대한 자세한 내용은 [Cisco CSR 1000v Deployment Guide for Microsoft Azure](#) and Related Information 섹션을 참조하십시오.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- Microsoft Azure 계정
- CSR1000v 2개 및 Windows/Linux 가상 머신 1개
- AzureCLI 2.0

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

2개의 CSR1000v 라우터와 1개의 VM(windows/linux)을 구축합니다. VM(Private Datacenter)에서 인터넷(8.8.8.8)으로의 연속 트래픽을 시뮬레이션합니다. HA 장애 조치를 시뮬레이션하고 Azure 라우팅 테이블이 CSR-A에서 CSR-B의 전용 인터페이스로 트래픽을 전환했는지 확인하여 HA가 성공했는지 확인합니다.

토폴로지

토폴로지와 설계를 완전히 이해하려면 구성을 시작하기 전에 먼저 해야 합니다. 이렇게 하면 나중에 잠재적인 문제를 해결할 수 있습니다.

사용자의 요구 사항에 따라 다양한 HA 구축 시나리오가 있을 수 있습니다. 이 예에서는 다음 설정으로 HA 이중화를 구성합니다.

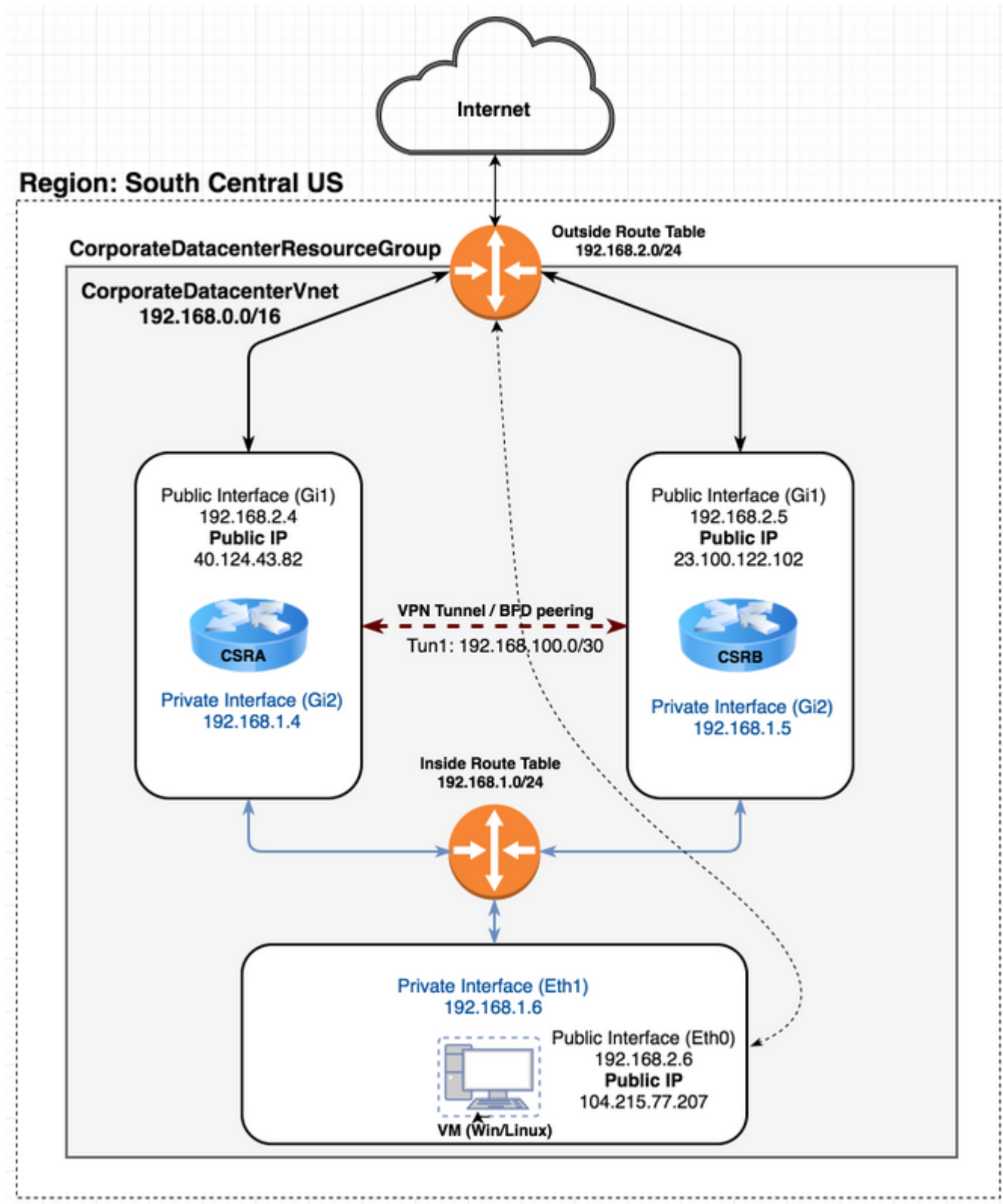
- 1x - 지역(미국 중남부)
- 1x - 리소스 그룹(CorporateDatacenterResourceGroup)
- 1x - Vnet(CorporateDatacenterVnet)
- 6x - 네트워크 인터페이스(내부 페이싱 3x 및 외부 페이싱 3x)
- 2x - 경로 테이블(InsideRoutetable 및 OutsideRoutetable)
- 2x - CSR1000v 라우터(Cisco IOS-XE® Denali 16.7.1)
- 1x - VM(Linux/Windows)

지금은 공용 인터페이스를 통한 인터넷 액세스가 VM에서 활성화된 상태로 유지되므로 이를 액세스하고 구성할 수 있습니다. 일반적으로 모든 일반 트래픽은 전용 경로 테이블을 통과해야 합니다. 나

중에 트래픽이 실수로 유출되지 않도록 VM의 공용 인터페이스를 비활성화할 수 있습니다.

트래픽 시뮬레이션은 VM의 프라이빗 인터페이스에서 CSRA → 8.8.8.8 → 내부 경로 테이블 → ping하여 수행됩니다. 장애 조치 시나리오에서 프라이빗 경로 테이블이 경로를 CSRB의 프라이빗 인터페이스로 전환했음을 확인합니다.

네트워크 다이어그램



용어

- 리소스 그룹 - Azure에서 가상 컴퓨터 및 vnet과 같은 모든 리소스를 추적하는 방법입니다. 이것은 일반적으로 모든 항목을 관리하고 요금을 추적하는 데 사용됩니다.
- Vnet - 가상 네트워크(aws 용어의 VPC와 유사)
- 경로 테이블 - 서브넷에 대한 규칙을 포함하며 특정 트래픽을 ip 주소로 전달하거나 VPN 엔드 포인트처럼 작동할 수 있습니다.

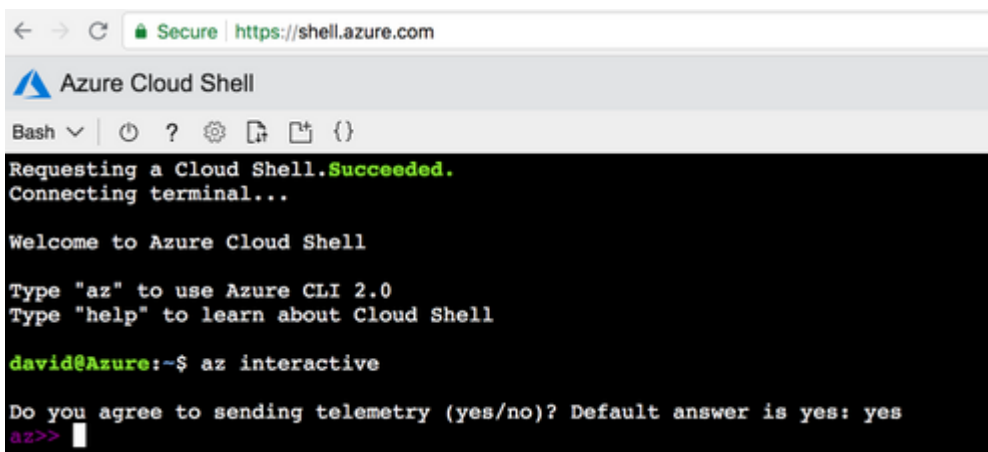
제한 사항

- Azure 자체는 HA 장애 조치에서 약 40-50초 지연을 도입할 수 있습니다.

설정

Azure에서 VM을 배포하는 몇 가지 방법이 있습니다.

1. [웹 포털](#) - cisco.com의 HA 문서
2. Powershell - Azure 리소스 관리를 위한 명령줄 기반 모델입니다.
3. [AzureCLI 2.0](#) - 명령줄 기반이기도 합니다. 오픈 소스이며 python으로 작성되었으므로 로컬 시스템에 설치해야 합니다. 이 문서를 작성하기 위해 AzureCLI 2.0이 최신 버전입니다.
4. [Azure Cloud Shell](#) - 셸을 통해 AzureCLI를 사용하려면 Powershell 옵션 대신 Bash 셸 옵션을 선택합니다. 이 방법에는 설치가 필요하지 않습니다.



```
< > ↻ Secure | https://shell.azure.com
Azure Cloud Shell
Bash ▾ | ⏻ ? ⚙️ 📄 🗑️ {}
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI 2.0
Type "help" to learn about Cloud Shell

david@Azure:~$ az interactive

Do you agree to sending telemetry (yes/no)? Default answer is yes: yes
az>>
```

Powershell과 AzureCLI는 유사하지만 AzureCLI에 대한 명령은 더 간단합니다. 둘 다 Windows, MacOS, Linux에서 실행할 수 있습니다. 비교는 [Azure에 적합한 도구 선택 및 Azure CLI 및 PowerShell 명령을 나란히](#) 참조하십시오.

이 예에서는 AzureCLI 또는 클라우드 셸을 사용하여 모든 리소스를 배포합니다. AzureCLI는 MacOS, Windows 또는 Linux에서 약간 다른 단계를 통해 설치할 수 있습니다. AzureCLI와 Azure Cloud Shell 간의 나머지 절차를 통해 컨피그레이션에는 차이가 없습니다.

```
redundancy
cloud provider azure 100
bfd peer
route-table
default-gateway ip
cidr ip
app-key
subscription-id
app-id
tenant-id
resource-group
```

 참고: 이 템플릿은 나중에 CSR에서 HA를 구성하는 데 사용되는 모든 ID 및 컨피그레이션을 추적하는 데 유용합니다.

개요

1단계. AzureCLI 2.0을 설치합니다.

1. AzureCLI [2.0](#) 설명서의 Windows, MacOS 또는 Linux 설치 [단계를](#) 따릅니다.
2. MacOS의 경우

```
$ brew update && brew install azure-cli
```

3. Azure에 로그인하고 지침에 따라 세션을 인증합니다.

```
$ az login
```

4. 브라우저 인증이 완료되면 Azure 구독 정보가 JSON 형식으로 반환됩니다.

```
[
  {
    "cloudName": "AzureCloud",
    "id": "09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx",
    "isDefault": true,
    "name": "Microsoft Azure Enterprise",
    "state": "Enabled",
    "tenantId": "ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx",
    "user": {
      "name": "cisco@cisco.com",
      "type": "user"
    }
  }
]
```

5. 나머지 구성 단계를 시작하기 전에 AzureCLI에 대한 몇 가지 유용한 명령과 팁을 알아보십시오.

- 사용 가능한 하위 명령 및 하위 명령의 작업에 대한 도움말을 보려면 -h 옵션을 사용합니다.

```
$ az account -h
```

- 기본적으로 모든 출력은 JSON 형식으로 반환됩니다. 쉽게 읽을 수 있도록 `--output table` 옵션을 사용하여 테이블에 표시할 수 있습니다.

```
$ az account list-locations --output table
```

- 사용 가능한 모든 vm 목록을 가져오거나 `--all` 옵션을 아래 다른 옵션 중 하나로 대체하여 테이블을 필터링합니다.

```
$ az vm image list --all --output table
```

You are retrieving all the images from server which could take more than a minute. To shorten the wait,

- 모든 구성 명령에 대한 자세한 내용은 Microsoft의 [Azure CLI 2.0](#) 설명서를 참조하십시오.

2단계. 자원 그룹을 생성합니다.

- 리소스 그룹은 Azure 솔루션에 대한 관련 리소스를 포함하는 컨테이너입니다. 리소스 그룹에 이름을 지정하고 컨테이너를 구축할 위치를 선택합니다. 이 예에서는 미국 중남부 지역을 사용합니다.

```
$ az account list-locations --output table
```

DisplayName	Latitude	Longitude	Name
East Asia	22.267	114.188	eastasia
Southeast Asia	1.283	103.833	southeastasia
Central US	41.5908	-93.6208	centralus
East US	37.3719	-79.8164	eastus
East US 2	36.6681	-78.3889	eastus2
West US	37.783	-122.417	westus
North Central US	41.8819	-87.6278	northcentralus
South Central US	29.4167	-98.5	southcentralus

```
$ az group create --name CorporateDatacenterResourceGroup --location "South Central US"
```

```
{
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup",
  "location": "southcentralus",
  "managedBy": null,
  "name": "CorporateDatacenterResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
}
```

```
"tags": null
}
```

- 템플릿(리소스 그룹 추가)

```
redundancy
cloud provider azure 100
bfd peer
route-table
default-gateway ip
cidr ip
app-key
subscription-id
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

3단계. Vnet을 생성합니다.

- Vnet은 네트워크가 구축된 ip 주소의 공간입니다. 이 범위는 더 작은 서브넷으로 분할되어 인터페이스에 할당됩니다. vnet에 이름을 지정하고 2단계에서 만든 리소스 그룹에 할당한 다음 접두사 범위를 할당합니다. 접두사를 지정하지 않으면 Azure는 일반적으로 사용자에게 10.0.0.0/16을 할당합니다.

```
$ az network vnet create --name CorporateDatacenterVnet --resource-group CorporateDatacenterResourceGroup
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "192.168.0.0/16"
      ]
    },
    "ddosProtectionPlan": null,
    "dhcpOptions": {
      "dnsServers": []
    },
    "enableDdosProtection": false,
    "enableVmProtection": false,
    "etag": "W/\"7c39a7a9-46e5-4082-a016-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualNetworks/CorporateDatacenterVnet",
    "location": "southcentralus",
    "name": "CorporateDatacenterVnet",
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "3d95d732-e46a-4fae-a34b-xxxxxxxxxxxx",
    "subnets": [],
    "tags": {},
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
```

4단계. 경로 테이블을 생성합니다.

1. 내부 연결 인터페이스에 대한 경로 테이블을 생성합니다.

```
$ az network route-table create --name InsideRoutetable --resource-group CorporateDatacenterResourceGroup \
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"45088005-cb6f-4356-bb18-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/routeTables"
}
```

- 템플릿(route-table 추가)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip
app-key
subscription-id
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

2. 외부 연결 인터페이스에 대한 경로 테이블을 생성합니다.

```
$ az network route-table create --name OutsideRoutetable --resource-group CorporateDatacenterResourceGroup \
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"a89b6230-9542-468c-b4b2-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/OutsideRoutetable",
  "location": "southcentralus",
  "name": "OutsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/routeTables"
}
```

5단계. 서브넷을 생성합니다.

1. 3단계에서 vnet에 할당한 공간에서 /24 서브넷을 만든 다음 Inside Route Table(내부 경로 테이블)에 할당합니다.

```
$ az network vnet subnet create --address-prefix 192.168.1.0/24 --name InsideSubnet --resource-group <resource-group>
{
  "addressPrefix": "192.168.1.0/24",
  "etag": "W/\"a0dbd178-3a45-48fb-xxxx-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/subnets/InsideSubnet",
  "ipConfigurations": null,
  "name": "InsideSubnet",
  "networkSecurityGroup": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceNavigationLinks": null,
  "routeTable": {
    "disableBgpRoutePropagation": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRouteTable",
    "location": null,
    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}
```

2. vnet에 할당한 공간에서 다른 /24 서브넷을 만들고 외부 경로 테이블에 할당합니다.

```
$ az network vnet subnet create --address-prefix 192.168.2.0/24 --name OutsideSubnet --resource-group <resource-group>
{
  "addressPrefix": "192.168.2.0/24",
  "etag": "W/\"874d1019-90a0-44fd-a09c-0aed8f2ede5b\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/subnets/OutsideSubnet",
  "ipConfigurations": null,
  "name": "OutsideSubnet",
  "networkSecurityGroup": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceNavigationLinks": null,
  "routeTable": {
    "disableBgpRoutePropagation": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/OutsideRouteTable",
    "location": null,
    "name": null,

```

```

    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}

```

6단계. CSR1000v 라우터를 생성합니다.

각 VM에는 2개의 인터페이스(내부 및 외부)가 있어야 하는데, 이는 VM당 2개의 NIC를 의미합니다. 2개의 NIC를 생성하고 공용 IP를 외부 NIC에 연결합니다.

1. 공용 IP 주소를 생성합니다.

```

$ az network public-ip create --name CSRAPublicIP --resource-group CorporateDatacenterResourceGroup
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"38306703-153b-456b-b2e4-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRAPublicIP",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "40.124.43.82",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRAPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "610e1631-331a-4971-8502-xxxxxxxxxxxx",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

2. 외부 NIC를 생성하고 공용 IP 주소를 여기에 연결합니다.

```

$ az network nic create --name CSRAOutsideInterface --resource-group CorporateDatacenterResourceGroup
{
  "NewNIC": {
    "dnsSettings": {

```

```
"appliedDnsServers": [],
"dnsServers": [],
"internalDnsNameLabel": null,
"internalDomainNameSuffix": "plk2sxe5i0l1ccksytfab.jx.internal.cloudapp.net",
"internalFqdn": null
},
"enableAcceleratedNetworking": false,
"enableIpForwarding": false,
"etag": "W/\"06fd60de-6547-4992-b506-xxxxxxxxxxxx\"",
"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
"ipConfigurations": [
  {
    "applicationGatewayBackendAddressPools": null,
    "applicationSecurityGroups": null,
    "etag": "W/\"06fd60de-6547-4992-xxxx-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.2.4",
    "privateIpAddressVersion": "IPv4",
    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": {
      "dnsSettings": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
      "idleTimeoutInMinutes": null,
      "ipAddress": null,
      "ipConfiguration": null,
      "ipTags": null,
      "location": null,
      "name": null,
      "provisioningState": null,
      "publicIpAddressVersion": null,
      "publicIpAllocationMethod": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceGuid": null,
      "sku": null,
      "tags": null,
      "type": null,
      "zones": null
    },
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "subnet": {
      "addressPrefix": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
      "ipConfigurations": null,
      "name": null,
      "networkSecurityGroup": null,
      "provisioningState": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null,
      "serviceEndpoints": null
    }
  }
],
"location": "southcentralus",
"macAddress": null,
```

```

    "name": "CSRAOutsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "93413822-e819-4644-ac0d-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

3. 내부 NIC를 생성합니다.

```

$ az network nic create --name CSRAInsideInterface --resource-group CorporateDatacenterResourceGroup --location southcentralus --mac-address 00056960000000000000000000000000
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzklk2sxe5i0l1ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/CSRAInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/CSRAInsideInterface/ipconfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.4",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {
          "addressPrefix": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/subnets/subnet1",
          "ipConfigurations": null,
          "name": null,
          "networkSecurityGroup": null,
          "provisioningState": null,
          "resourceGroup": "CorporateDatacenterResourceGroup",
          "resourceNavigationLinks": null,
          "routeTable": null,
          "serviceEndpoints": null
        }
      }
    ],
    "location": "southcentralus",
    "macAddress": null,
  }
}

```

```

    "name": "CSRAInsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "0f7ae52a-47c3-4563-9fe0-b1484e88296e",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

4. Azure에서 사용 가능한 CSR1000v 이미지를 나열합니다. 이 예에서는 urn 이름인 cisco:cisco-csr-1000v:16_7:16.7.120171201을 사용합니다.

```

az vm image list --all --publisher Cisco --offer cisco-csr-1000v
[
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_5",
    "urn": "cisco:cisco-csr-1000v:16_5:16.5.120170418",
    "version": "16.5.120170418"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_5",
    "urn": "cisco:cisco-csr-1000v:16_5:16.5.220171128",
    "version": "16.5.220171128"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_6",
    "urn": "cisco:cisco-csr-1000v:16_6:16.6.120170804",
    "version": "16.6.120170804"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_6",
    "urn": "cisco:cisco-csr-1000v:16_6:16.6.220171219",
    "version": "16.6.220171219"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "16_7",
    "urn": "cisco:cisco-csr-1000v:16_7:16.7.120171201",
    "version": "16.7.120171201"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "3_16",
    "urn": "cisco:cisco-csr-1000v:3_16:3.16.420170208",
    "version": "3.16.420170208"
  },
  {

```

```

    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "3_16",
    "urn": "cisco:cisco-csr-1000v:3_16:3.16.520170215",
    "version": "3.16.520170215"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "csr-azure-byol",
    "urn": "cisco:cisco-csr-1000v:csr-azure-byol:16.40.120170206",
    "version": "16.40.120170206"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "csr-azure-byol",
    "urn": "cisco:cisco-csr-1000v:csr-azure-byol:3.16.0",
    "version": "3.16.0"
  },
  {
    "offer": "cisco-csr-1000v",
    "publisher": "cisco",
    "sku": "csr-azure-byol",
    "urn": "cisco:cisco-csr-1000v:csr-azure-byol:3.16.2",
    "version": "3.16.2"
  }
]

```

5. 이미지의 URN 이름으로 CSR1000v를 구축합니다.

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRA --location southcentralus --image cisco-csr-1000v --size Standard_D2s_v3 --show-details
Running ..
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/virtualMachines/CSRA",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-83-58,00-0D-3A-5D-89-27",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.4,192.168.1.4",
  "publicIpAddress": "40.124.43.82",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

몇 분 후에 새로운 CSR1000v가 부팅됩니다.

```

$ az vm list --resource-group CorporateDatacenterResourceGroup --show-details --output table
Name ResourceGroup PowerState PublicIps Fqdns Location Zones
-----
CSRA CorporateDatacenterResourceGroup VM running 40.124.43.82 southcentralus

```

6. CSR1000v에 로그인하여 기능을 확인합니다.

```
$ ssh cisco@40.124.43.82
The authenticity of host '40.124.43.82 (40.124.43.82)' can't be established.
RSA key fingerprint is SHA256:q33FHw7RlkDn
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.124.43.82' (RSA) to the list of known hosts.
Password:

CSRA#
CSRA#show ip interface brief
Interface IP-Address OK? Method Status Protocol
GigabitEthernet1 192.168.2.4 YES DHCP up up
GigabitEthernet2 192.168.1.4 YES DHCP up up
```

7단계. 두 번째 CSR1000v 라우터를 생성합니다.

1. 공용 IP 주소를 생성합니다.

```
$ az network public-ip create --name CSRBPUBLICIP --resource-group CorporateDatacenterResourceGroup
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"f0f98dac-ea56-4efe-8da6-81a221ac3474\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRBPUBLICIP",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "23.100.122.102",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRBPUBLICIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "aa03bc26-22df-4696-bd77-ca29df029d7d",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}
```

2. 외부 NIC를 생성하고 공용 IP 주소를 여기에 연결합니다.

```
$ az network nic create --name CSRBOURSIDEINTERFACE --resource-group CorporateDatacenterResourceGroup
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": []
    }
  }
}
```

```
"internalDnsNameLabel": null,
"internalDomainNameSuffix": "gllzklk2sxe5i01lccksytfab.jx.internal.cloudapp.net",
"internalFqdn": null
},
"enableAcceleratedNetworking": false,
"enableIpForwarding": false,
"etag": "W/\"ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
"ipConfigurations": [
  {
    "applicationGatewayBackendAddressPools": null,
    "applicationSecurityGroups": null,
    "etag": "W/\"ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.2.5",
    "privateIpAddressVersion": "IPv4",
    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": {
      "dnsSettings": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
      "idleTimeoutInMinutes": null,
      "ipAddress": null,
      "ipConfiguration": null,
      "ipTags": null,
      "location": null,
      "name": null,
      "provisioningState": null,
      "publicIpAddressVersion": null,
      "publicIpAllocationMethod": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceGuid": null,
      "sku": null,
      "tags": null,
      "type": null,
      "zones": null
    },
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "subnet": {
      "addressPrefix": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR",
      "ipConfigurations": null,
      "name": null,
      "networkSecurityGroup": null,
      "provisioningState": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null,
      "serviceEndpoints": null
    }
  }
],
"location": "southcentralus",
"macAddress": null,
"name": "CSRBOOutsideInterface",
"networkSecurityGroup": null,
```

```

    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "c3f05156-ad07-4abd-a006-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

3. 내부 NIC를 생성합니다.

```
$ az network nic create --name CSRBIInsideInterface --resource-group CorporateDatacenterResourceGroup --location southcentralus --mac-address 000000000000 --ip-configurations [{"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ipConfigurations/CSRBIInsideInterface_ipconfig1"}]
```

$$\left. \begin{array}{l} \{ \\ \{ \end{array} \right\}$$

다.

 $\{$

ntuLTS를 사용합니다.

공용 IP 주소를 생성합니다.

{

```

    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

2. 외부 NIC를 생성하고 OutsideSubnet과 공용 IP 주소를 연결합니다. 서브넷이 NIC에 연결되면 IP 주소가 NIC에 자동으로 할당됩니다. 이 예에서 OutsideSubnet은 192.168.2.0/24이며 NIC에 자동으로 할당되는 IP 주소는 192.168.2.6입니다.

```

$ az network nic create --name VMHostOutsideInterface --resource-group CorporateDatacenterResourceGroup --subnet OutsideSubnet --ip-address 192.168.2.6 --public-ip-address 192.168.2.6 --public-ip-address-version IPv4 --dns-name-label vmhostoutsideinterface --dns-suffix gzkp1k2sxe5i0l1ccksytfab.jx.internal.cloudapp.net --tags vmhostoutsideinterface
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gzkp1k2sxe5i0l1ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/subnets/OutsideSubnet/nics/VMHostOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/subnets/OutsideSubnet/ipconfigurations/VMHostOutsideInterface",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.2.6",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": {
          "dnsSettings": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/publicIPAddresses/VMHostOutsideInterface",
          "idleTimeoutInMinutes": null,
          "ipAddress": null,
          "ipConfiguration": null,
          "ipTags": null,
          "location": null,
          "name": null,
          "provisioningState": null,
          "publicIpAddressVersion": null,
          "publicIpAllocationMethod": null,
          "resourceGroup": "CorporateDatacenterResourceGroup",
          "resourceGuid": null,
          "sku": null,
          "tags": null,
          "type": null,
          "zones": null
        },
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {

```

```

        "addressPrefix": null,
        "etag": null,
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/vmHostOutsideInterface",
        "ipConfigurations": null,
        "name": null,
        "networkSecurityGroup": null,
        "provisioningState": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "resourceNavigationLinks": null,
        "routeTable": null,
        "serviceEndpoints": null
    }
},
{
    "location": "southcentralus",
    "macAddress": null,
    "name": "VMHostOutsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "89588a04-6ba6-467d-a86f-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
}
}

```

3. 내부 NIC를 생성합니다.

```

$ az network nic create --name VMHostInsideInterface --resource-group CorporateDatacenterResourceGroup --location southcentralus --subnet vmHostSubnet --ip-address 192.168.1.6 --private-ip-address 192.168.1.6 --private-ip-allocation-method dynamic --provisioning-state Succeeded --tags CorporateDatacenterResourceGroup=vmHostInsideInterface
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i0l1ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/vmHostInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/vmHostInsideInterface/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.6",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {

```

```

        "addressPrefix": null,
        "etag": null,
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualMachineNetworkInterfaces/VMHostInsideInterface",
        "ipConfigurations": null,
        "name": null,
        "networkSecurityGroup": null,
        "provisioningState": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "resourceNavigationLinks": null,
        "routeTable": null,
        "serviceEndpoints": null
    }
},
{
    "location": "southcentralus",
    "macAddress": null,
    "name": "VMHostInsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "8ef12cdd-cc31-432e-99cf-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
}
}

```

4. Ubuntu VM을 구축합니다. 이 예에서는 UbuntuLTS를 사용합니다.

```

az vm image list --output table
You are viewing an offline list of images, use --all to retrieve an up-to-date list

```

Offer	Publisher	Skus	Urn
CentOS	OpenLogic	7.3	OpenLogic:CentOS:7.3:latest
CoreOS	CoreOS	Stable	CoreOS:CoreOS:Stable:latest
Debian	credativ	8	credativ:Debian:8:latest
openSUSE-Leap	SUSE	42.3	SUSE:openSUSE-Leap:42.3:latest
RHEL	RedHat	7.3	RedHat:RHEL:7.3:latest
SLES	SUSE	12-SP2	SUSE:SLES:12-SP2:latest
UbuntuServer	Canonical	16.04-LTS	Canonical:UbuntuServer:16.04-LTS:latest
WindowsServer	MicrosoftWindowsServer	2016-Datacenter	MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest
WindowsServer	MicrosoftWindowsServer	2012-R2-Datacenter	MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest
WindowsServer	MicrosoftWindowsServer	2012-Datacenter	MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest
WindowsServer	MicrosoftWindowsServer	2008-R2-SP1	MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name VmHost --location southcentralus --image Canonical:UbuntuServer:16.04-LTS:latest
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualMachineNetworkInterfaces/VMHostInsideInterface",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-B7-CB,00-0D-3A-5D-B8-9B",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.6,192.168.1.6",
  "publicIpAddress": "104.215.77.207",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

```
}
```

9단계. 라우팅 테이블 및 VM에 경로를 추가합니다.

1. 다음 hops IP 주소를 192.168.1.4로 설정하여 CSR A를 통해 트래픽을 라우팅하려면 내부 서브넷에 대한 기본 경로를 추가합니다. 이 작업은 InsideRouteTable에서 수행합니다.

```
$ az network route-table route create --address-prefix 8.8.8.8/32 --name default_route --next-hop-type VirtualAppliance --next-hop-address 192.168.1.4 --resource-group CorporateDatacenterResourceGroup
{
  "addressPrefix": "8.8.8.8/32",
  "etag": "W/\"ef9e650a-5d70-455d-b958-5a0efc07e7ad\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/default_route/routes/default_route",
  "name": "default_route",
  "nextHopIpAddress": "192.168.1.4",
  "nextHopType": "VirtualAppliance",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup"
}
```

2. OutsideRouteTable에서 인터넷에 연결하기 위해 네트워크에 트래픽에 대한 경로를 추가합니다.

```
$ az network route-table route create --address-prefix 8.8.8.8/32 --name internet --next-hop-type Internet --next-hop-address null --resource-group CorporateDatacenterResourceGroup
{
  "addressPrefix": "8.8.8.8/32",
  "etag": "W/\"d2c7e32e-8d32-4856-a3a6-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/default_route/routes/internet",
  "name": "internet",
  "nextHopIpAddress": null,
  "nextHopType": "Internet",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup"
}
```

3. Ubuntu VM에 로그인하고 내부 인터페이스를 통한 트래픽을 8.8.8.8로 강제하는 경로를 추가합니다. Azure 경로 테이블은 서브넷의 첫 번째 IP를 게이트웨이로 자동으로 사용합니다. 내부 인터페이스(eth1)의 서브넷은 192.168.1.0/24입니다. 즉, 192.168.1.1이 호스트 VM의 기본 gw 주소입니다.

```
$ ifconfig
eth0 Link encap:Ethernet HWaddr 00:0d:3a:5d:b7:cb
  inet addr:192.168.2.6 Bcast:192.168.2.255 Mask:255.255.255.0
  inet6 addr: fe80::20d:3aff:fe5d:b7cb/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:3986 errors:0 dropped:0 overruns:0 frame:0
  TX packets:2881 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:3475393 (3.4 MB) TX bytes:592740 (592.7 KB)

eth1 Link encap:Ethernet HWaddr 00:0d:3a:5d:b8:9b
  inet addr:192.168.1.6 Bcast:192.168.1.255 Mask:255.255.255.0
  inet6 addr: fe80::20d:3aff:fe5d:b89b/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:2 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:762 (762.0 B) TX bytes:1620 (1.6 KB)
```

```
$ sudo route add -host 8.8.8.8 gw 192.168.1.1 dev eth1
$ route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          192.168.2.1     0.0.0.0          UG    0      0      0 eth0
8.8.8.8          192.168.1.1     255.255.255.255 UGH    0      0      0 eth1
168.63.129.16    192.168.2.1     255.255.255.255 UGH    0      0      0 eth0
169.254.169.254 192.168.2.1     255.255.255.255 UGH    0      0      0 eth0
192.168.1.0      0.0.0.0         255.255.255.0    U     0      0      0 eth1
192.168.2.0      0.0.0.0         255.255.255.0    U     0      0      0 eth0
```

- 템플릿(cidr ip 추가)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key
subscription-id
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

 참고: 인터넷(8.8.8.8)을 ping하려면 10단계에서 CSR1000v 라우터에서 NAT를 구성해야 합니다.

 참고: 10-14단계에서는 HA용 CSR1000v 라우터의 컨피그레이션을 다룹니다.
[Microsoft Azure용 Cisco CSR 1000v 배포 가이드의 약식 단계](#)는 트러스트 풀 구성부터 제공됩니다. 자세한 내용은 설명서를 참조하십시오.

10단계. CSR1000v 라우터를 구성합니다.

1. 두 CSR1000v 라우터 모두에서 트러스트 풀 구성

```
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)#crypto pki trustpool import url http://www.cisco.com/security/pki/trs/ios.p7b
Reading file from http://www.cisco.com/security/pki/trs/ios.p7b
Loading http://www.cisco.com/security/pki/trs/ios.p7b !!!
% PEM files import succeeded.
```

2. Cisco CSR 1000v 라우터 간의 ipsec 터널을 구성하고 피어 오류 탐지를 위해 라우터 간의 터널에서 BFD(Bi-directional Forwarding Detection) 및 EIGRP 또는 BGP(Routing Protocol)를 활성화합니다.



참고: 컨피그레이션의 터널 대상 주소는 피어 CSR의 공용 IP 주소입니다.

- CSRA 컨피그레이션

```
crypto isakmp policy 1
  encr aes 256
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0
!
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 192.168.101.1 255.255.255.252
  bfd interval 500 min_rx 500 multiplier 3
  tunnel source GigabitEthernet1
  tunnel mode ipsec ipv4
  tunnel destination 23.100.122.102 /* Public IP of the peer CSR */
  tunnel protection ipsec profile vti-1
!
router eigrp 1
  bfd all-interfaces
  network 192.168.101.0
```

- CSRB 컨피그레이션

```
crypto isakmp policy 1
  encr aes 256
  authentication pre-share
crypto isakmp key cisco address 0.0.0.0
!
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 192.168.101.2 255.255.255.252
  bfd interval 500 min_rx 500 multiplier 3
  tunnel source GigabitEthernet1
  tunnel mode ipsec ipv4
```

```

tunnel destination 40.124.43.82 /* Public IP of the peer CSR */
tunnel protection ipsec profile vti-1
!
router eigrp 1
  bfd all-interfaces
  network 192.168.101.0

```

3. NAT 및 라우팅에 대한 동일한 컨피그레이션은 두 CSR1000v 라우터에서 모두 사용됩니다. 이는 내부 인터페이스를 통한 VM 인터넷 연결성을 위한 것입니다.

```

interface GigabitEthernet1
  ip nat outside
!
interface GigabitEthernet2
  ip nat inside
!
ip nat inside source list 10 interface GigabitEthernet1 overload
access-list 10 permit 192.168.1.0 0.0.0.255 /* Translating the inside subnet of the VM */
!
ip route 0.0.0.0 0.0.0.0 192.168.2.1
ip route 192.168.1.0 255.255.255.0 GigabitEthernet2 192.168.1.1

```

4. 경로 테이블에 대한 액세스 제어(IAM)를 추가합니다. AzureCLI에서 응용 프로그램(CSRA 및 CSRB)이 장애 조치 중에 Azure에서 InsideRouteTable을 수정하도록 허용합니다.
- 다음 섹션에서 —scopes 옵션으로 사용할 InsideRouteTable의 ID를 확인합니다.

```

$ az network route-table show --resource-group CorporateDatacenterResourceGroup --name InsideRouteTable
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"f0c85464-bba0-465a-992a-xxxxxxxxxxxx\"\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRouteTable",
  "location": "southcentralus",
  "name": "InsideRouteTable",
  ...

```

- 템플릿(subscription-id 추가)

```

redundancy
cloud provider azure 100
bfd peer
route-table InsideRouteTable
default-gateway ip
cidr ip 8.8.8.8/32
app-key
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup

```

5. InsideRouteTable에 대한 IAM 역할을 만듭니다. 범위 — 옵션은 이전 출력의 id 필드에서 가져옵니다.

- app-id, password(app-key) 및 테넌트 ID를 기록합니다.

```
$ az ad sp create-for-rbac -n "InsideRouteTableIAM" --role "network contributor" --scopes /subscriptions/09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
{
  "appId": "576dd4f1-c08d-xxxx-xxxx-xxxxxxxxxxxxx",
  "displayName": "InsideRouteTableIAM",
  "name": "http://InsideRouteTableIAM",
  "password": "aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx",
  "tenant": "ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx"
}
```

- 템플릿(app-key, app-id 및 tenant-id 추가)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup
```

6. 두 라우터에서 클라우드 이중화를 구성합니다. 두 라우터의 컨피그레이션 간의 유일한 차이점은 bfd 피어와 기본 게이트웨이입니다.

- CSRA 컨피그레이션

```
redundancy
cloud provider azure 100
bfd peer 192.168.101.2
route-table InsideRoutetable
default-gateway ip 192.168.1.4
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup
```

- CSRB 컨피그레이션

```
redundancy
cloud provider azure 100
bfd peer 192.168.101.1
route-table InsideRoutetable
default-gateway ip 192.168.1.5
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
```

tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup

고가용성 확인

1. BFD 및 클라우드 컨피그레이션을 확인합니다.

```
CSRA#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	192.168.2.4	YES	DHCP	up	up
GigabitEthernet2	192.168.1.4	YES	DHCP	up	up
Tunnel1	192.168.101.1	YES	manual	up	up

```
CSRB#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
GigabitEthernet1	192.168.2.5	YES	DHCP	up	up
GigabitEthernet2	192.168.1.5	YES	DHCP	up	up
Tunnel1	192.168.101.2	YES	NVRAM	up	up

```
CSRA#show bfd neighbors
```

IPv4 Sessions					
NeighAddr		LD/RD	RH/RS	State	Int
192.168.101.2		4097/4097	Up	Up	Tu1

```
CSRA#show redundancy cloud provider azure 100
Cloud HA: work_in_progress=FALSE
Provider : AZURE node 100
State : idle
BFD peer      = 192.168.101.2
BFD intf      = Tunnel1
resource group = CorporateDatacenterResourceGroup
subscription id = 09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxxx
tenant id = ae49849c-2622-4d45-b95e-xxxxxxxxxxxxx
application id = 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxxxxxx
application key = aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
route-table    = InsideRoutetable
cidr            = 8.8.8.8/32
Default Gateway IP = 192.168.1.4
```

2. VM에서 대상으로 ping 및 traceroute를 실행합니다. ping이 내부 eth1 인터페이스를 통과하는지 확인합니다.

```
$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.6 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=54 time=10.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=54 time=10.6 ms
```

```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.1.4 (192.168.1.4)  1.516 ms  1.503 ms  1.479 ms
```

```
cisco@VmHost:~$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.6 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=10.2 ms
```

3. Traceroute는 VM에서 8.8.8.8까지의 경로가 CSRA의 내부 인터페이스를 거친다는 것을 보여줍니다.

```
cisco@VmHost:~$ sudo traceroute -I 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.1.4 (192.168.1.4)  34.003 ms  34.000 ms  33.998 ms
```

4. CSRA의 터널 1 인터페이스를 종료하여 장애 조치를 시뮬레이션합니다.

```
CSRA#config t
Enter configuration commands, one per line.  End with CNTL/Z.
CSRA(config)#int tunnel1
CSRA(config-if)#sh
```

5. 이제 트래픽이 CSRB의 프라이빗 인터페이스를 통해 이동하는지 확인합니다.

```
cisco@VmHost:~$ sudo traceroute -I 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.1.5 (192.168.1.5)  1.294 ms  1.291 ms  1.290 ms
```



참고: Azure 클라우드에서 장애 조치 시 지연이 발생할 수 있습니다. 지연은 1분 이하여야 합니다.

문제 해결

- HA 장애 조치 중에 메시지를 관찰하려면 디버그를 활성화합니다.

```
CSRA#debug redundancy cloud all
CSRA#debug ip http all
```

- 인증 및 자격 증명 오류는 CSR1000v가 Azure 경로 테이블에 대한 API 호출을 수행할 수 있도록 하는 잘못된 액세스 컨트롤로 인해 발생합니다. 10단계에서 적절한 ID가 구성되어 있는지 다시 한 번 확인합니다.

```
*Jul 13 23:29:53.365: CLOUD-HA : res content iov_len=449 iov_base={"error":"invalid_client","error_desc":  
Error validating credentials. AADSTS50012: Invalid client secret is provided.\r\nTrace ID: 56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx",  
Correlation ID: cce94817-29eb-4ebd-833a-\r\nTimestamp: 2018-07-13 23:29:54Z","error_codes":[70002,50012],  
23:29:54Z","trace_id":"56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx","correlation_id":"cce94817-29eb-4ebd-833a-
```

관련 정보

- [Azure CLI 2.0](#)
- [Microsoft Azure용 Cisco CSR 1000v 배포 가이드](#)
- [Azure 및 Azure CLI 및 PowerShell 명령에 대한 올바른 도구 선택](#)

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.