

CSR1000v HA Redundancy Deployment Guide no Microsoft Azure com AzureCLI 2.0

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Meta](#)

[Topologia](#)

[Diagrama de Rede](#)

[Terminology](#)

[Restrições](#)

[Configuração](#)

[Overview](#)

[Etapa 1. Instale o AzureCLI 2.0.](#)

[Etapa 2. Criar um Grupo de Recursos.](#)

[Etapa 3. Crie um Vnet.](#)

[Etapa 4. Criar Tabelas de Rotas.](#)

[Etapa 5. Criar Sub-Redes.](#)

[Etapa 6. Crie um roteador CSR1000v.](#)

—

[Passo 7. Crie o segundo roteador CSR1000v.](#)

[Etapa 8. Crie uma VM de host com o mesmo procedimento na etapa 6. Este exemplo usa UbuntuLTS.](#)

[Etapa 9. Adicione rotas a tabelas de roteamento e VMs.](#)

[Etapa 10. Configure os roteadores CSR1000v.](#)

[Verificar alta disponibilidade](#)

[Troubleshoot](#)

[Informações Relacionadas](#)

Introduction

Este documento fornece um guia de configuração passo a passo sobre como implantar roteadores CSR1000v para alta disponibilidade na nuvem do Microsoft Azure com AzureCLI 2.0. O objetivo é dar aos usuários conhecimento prático de HA e a capacidade de implantar um campo de teste totalmente funcional.

Há vários métodos para implantar imagens no Azure e o método mais familiar para a maioria dos usuários é através do portal da Web. No entanto, o AzureCLI é uma ferramenta rápida e poderosa quando você está familiarizado com ela.

Para obter informações mais detalhadas sobre o Azure, como implantar um CSR1000v por meio do portal da Web e HA, consulte o [Guia de Implantação do Cisco CSR 1000v para o Microsoft](#)

[Azure](#) e a seção Informações Relacionadas.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Uma conta do Microsoft Azure
- 2 máquinas virtuais CSR1000v e 1 Windows/Linux
- AzureCLI 2.0

Componentes Utilizados

As informações neste documento são baseadas no Cisco IOS-XE® Denali 16.7.1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Meta

Implante 2 roteadores CSR1000v e 1 VM (windows/linux). Simule o tráfego contínuo do datacenter privado (VM) para a Internet (8.8.8.8). Simule um failover de HA e observe que o HA obteve êxito confirmando que a tabela de roteamento do Azure mudou o tráfego do CSR-A para a interface privada do CSR-B.

Topologia

Para entender totalmente a topologia e o projeto é importante antes do início da configuração. Isso ajuda a solucionar possíveis problemas posteriormente.

Pode haver vários cenários de implantações de HA com base nos requisitos do usuário. Para este exemplo, configure a redundância de HA com estas configurações:

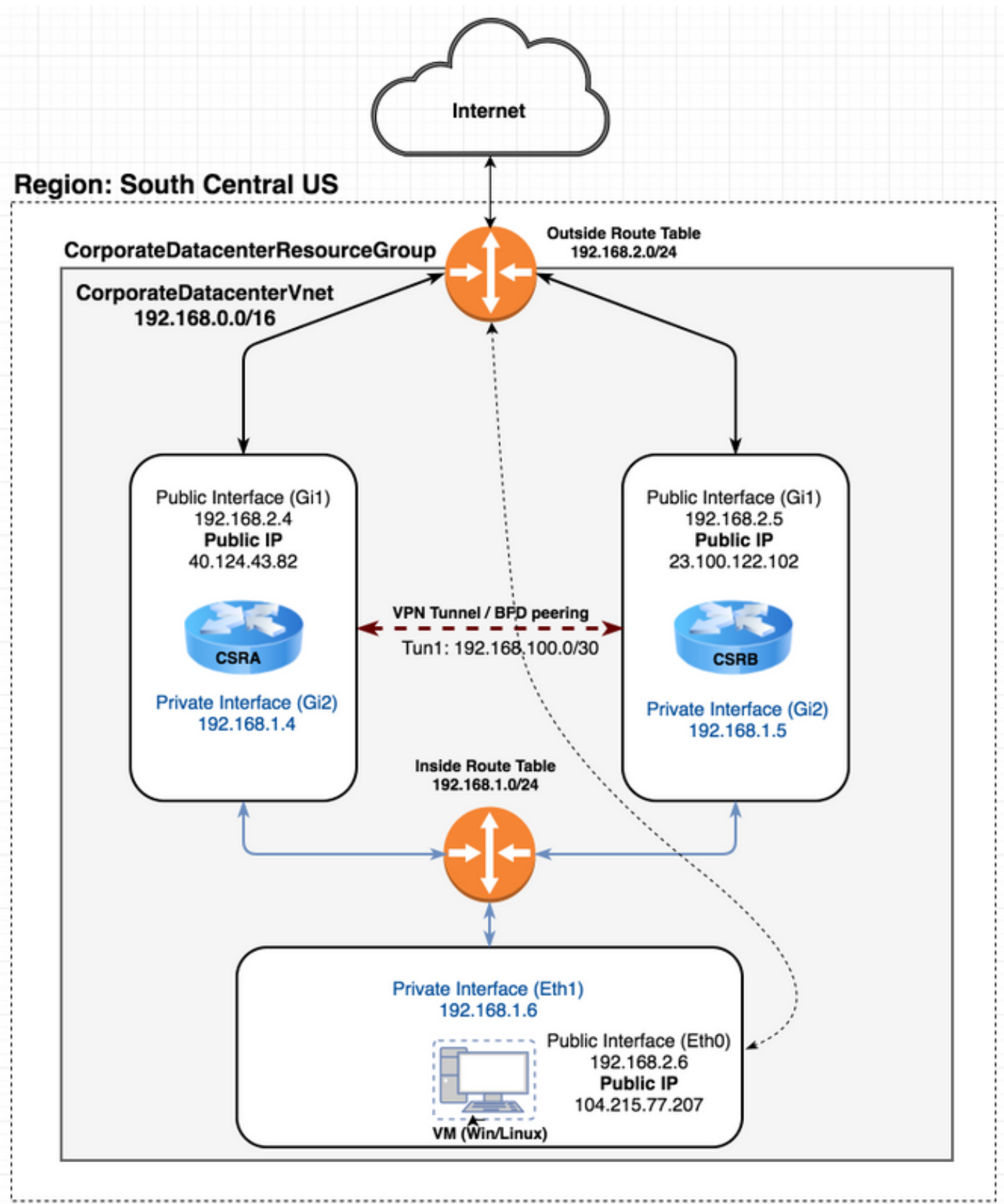
- 1x - Região (Centro-Sul dos EUA)
- 1x - Grupo de recursos (CorporateDatacenterResourceGroup)
- 1x - Vnet (CorporateDatacenterVnet)
- 6x - Interfaces de rede (3x Inside Facing e 3x Outside Facing)
- 2x - Tabelas de rota (InsideRoutetable e OutsideRoutetable)
- 2x - Roteadores CSR1000v (Cisco IOS-XE® Denali 16.7.1)
- 1x - VM (Linux/Windows)

Por enquanto, o acesso à Internet através da interface pública é deixado habilitado na VM para que você possa acessá-la e configurá-la. Geralmente, todo o tráfego normal deve fluir pela tabela de rota privada. A interface pública na VM pode ser desabilitada posteriormente para que nenhum tráfego vaze acidentalmente.

A simulação de tráfego é realizada por ping da interface privada da VM → tabela de rotas internas

→ CSRA → 8.8.8.8. Em um cenário de failover, observe que a tabela de rota privada mudou a rota para apontar para a interface privada do CSRB.

Diagrama de Rede



Terminology

- Grupo de Recursos - Esta é uma forma do Azure controlar todos os seus recursos, como máquinas virtuais e vnets. Geralmente, ele é usado para gerenciar todos os itens e controlar os encargos.
- Vnet - Uma rede virtual. (semelhante ao VPC na terminologia do ws)
- Tabela de rotas - Contém as regras de uma sub-rede e pode encaminhar tráfego específico para um endereço ip ou atuar como um ponto de extremidade de VPN.

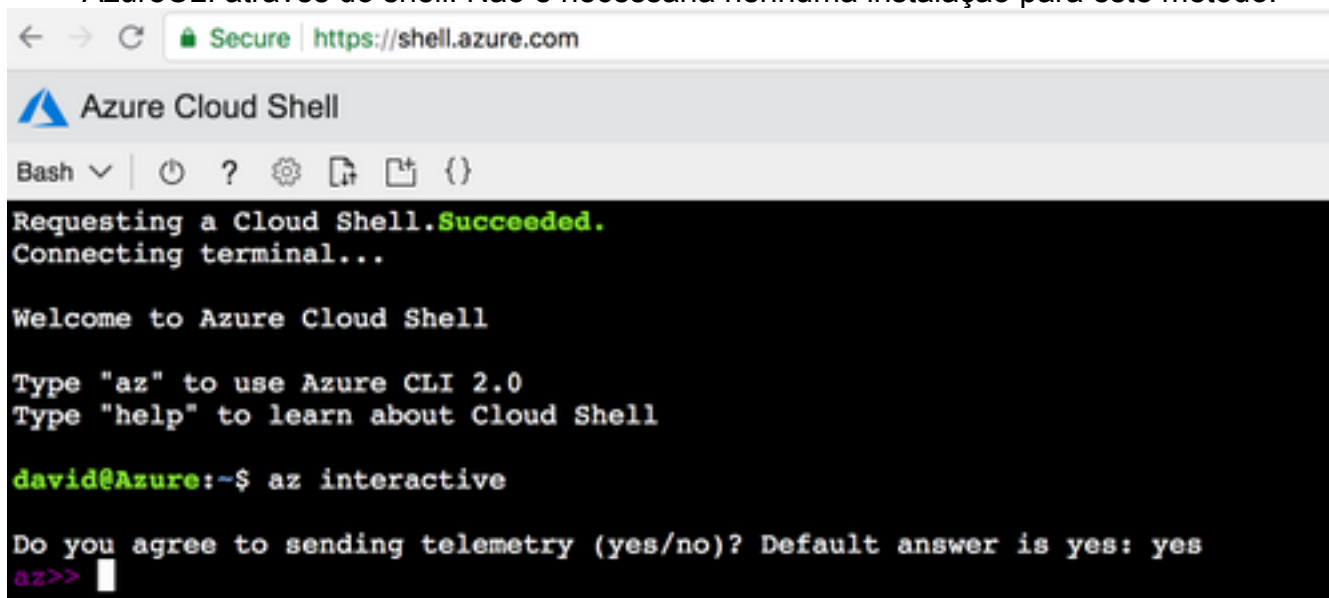
Restrições

- O próprio Azure pode introduzir aproximadamente um atraso de 40 a 50 segundos em um failover de HA.

Configuração

Há alguns métodos para implantar VMs no Azure:

1. [Portal da Web](#) - Documentação de HA em cisco.com
2. Powershell - Modelo baseado em linha de comando para gerenciamento de recursos do Azure.
3. [AzureCLI 2.0](#) - Também baseado em linha de comando. Ele é de código aberto e escrito em python e precisa ser instalado no sistema local. Para gravar este documento, o AzureCLI 2.0 é a versão mais recente.
4. [Azure Cloud Shell](#) - Escolha a opção **Bash shell** em vez da opção **Powershell** para usar o AzureCLI através do shell. Não é necessária nenhuma instalação para este método.



```
← → ↻ 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>>
```

O Powershell e o AzureCLI são semelhantes, mas os comandos para o AzureCLI são mais diretos. Ambos podem ser executados no Windows, MacOS, Linux. Consulte [Escolhendo a ferramenta certa para o Azure e os comandos lado a lado do Azure CLI e do PowerShell](#) para uma comparação.

Para este exemplo, implante todos os recursos com o AzureCLI ou Cloud Shell. O AzureCLI pode ser instalado no MacOS, Windows ou Linux com etapas ligeiramente diferentes. Não há diferença na configuração durante o resto do procedimento entre o AzureCLI e o Azure Cloud Shell.

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

Note: Este modelo é útil para controlar todas as IDs e configurações que são posteriormente usadas para configurar o HA nos CSRs.

Overview

Etapa 1. Instale o AzureCLI 2.0.

1. Siga as etapas de instalação para Windows, MacOS ou Linux na documentação [do AzureCLI 2.0](#).
2. Para MacOS:

```
$ brew update && brew install azure-cli
```
3. Faça login no Azure e siga as instruções para autenticar sua sessão.

```
$ az login
```
4. Quando a autenticação do navegador estiver concluída, as informações da assinatura do Azure serão retornadas no formato JSON:

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

5. Antes de começar com o resto das etapas de configuração, aqui estão alguns comandos e dicas úteis sobre o AzureCLI.

- Para obter ajuda com subcomandos disponíveis e o que eles fazem, use a opção **-h**.

```
$ az account -h
```

- Todas as saídas são retornadas no formato JSON por padrão. Para facilitar a leitura, você

pode usar a opção **—tabela de saída** para exibir em uma tabela.

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

- Obtenha uma lista de todas as vms disponíveis ou substitua a opção **all** por uma das opções abaixo para filtrar a tabela.

```
$ 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, provide '--publisher', '--offer' or '--sku'. Partial name search is supported.

- Consulte a documentação [do Azure CLI 2.0](#) da Microsoft para obter informações detalhadas sobre todos os comandos de configuração.

Etapa 2. Criar um Grupo de Recursos.

- Um Grupo de Recursos é um contêiner que contém recursos relacionados para uma solução do Azure. Dê um nome ao Grupo de Recursos e escolha um local para implantar o contêiner. Este exemplo usa o Centro-Sul dos EUA.

```
$ 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
}
```

- Modelo (Adicionando grupo de recursos)

```
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
```

Etapa 3. Crie um Vnet.

- Um Vnet é um espaço de endereços ip onde nossa rede é implantada. Esse intervalo é então dividido em sub-redes menores e atribuído às interfaces. Atribua um nome à sua vnet, atribua-a ao grupo de recursos criado na etapa 2 e aloque um intervalo de prefixo. Se você não especificar um prefixo, o Azure geralmente lhe atribui 10.0.0.0/16.

```
$ az network vnet create --name CorporateDatacenterVnet --resource-group
CorporateDatacenterResourceGroup --address-prefix 192.168.0.0/16
{
  "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/virtual
Networks/CorporateDatacenterVnet",
    "location": "southcentralus",
    "name": "CorporateDatacenterVnet",
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "3d95d732-e46a-4fae-a34b-xxxxxxxxxxxx",
    "subnets": [],
    "tags": {},
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
```

Etapa 4. Criar Tabelas de Rotas.

1. Crie uma tabela de rotas para as interfaces internas.

```
$ 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/ro
uteTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
```

```
"tags": null,
"type": "Microsoft.Network/routeTables"
}
```

Modelo (Adicionando tabela de rota)

```
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. Crie uma tabela de rotas para as interfaces externas.

```
$ 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/ro
uteTables/OutsideRoutetable",
  "location": "southcentralus",
  "name": "OutsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/routeTables"
}
```

Etapa 5. Criar Sub-Redes.

1. Crie uma sub-rede /24 a partir do espaço atribuído à vnet na etapa 3 e atribua-a à Tabela de Rotas Internas.

```
$ az network vnet subnet create --address-prefix 192.168.1.0/24 --name InsideSubnet --
resource-group CorporateDatacenterResourceGroup --vnet-name CorporateDatacenterVnet --
route-table InsideRoutetable
{
  "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/vi
rtualNetworks/CorporateDatacenterVnet/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/ro
```



```

uteTables/InsideRoutetable",
  "location": null,
  "name": null,
  "provisioningState": null,
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": null,
  "subnets": null,
  "tags": null,
  "type": null
},
"serviceEndpoints": null
}

```

2. Crie outra sub-rede /24 do espaço atribuído à vnet e atribua-a à Tabela de Rotas Externas.

```

$ az network vnet subnet create --address-prefix 192.168.2.0/24 --name OutsideSubnet --
resource-group CorporateDatacenterResourceGroup --vnet-name CorporateDatacenterVnet --
route-table OutsideRoutetable
{
  "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/vi
rtualNetworks/CorporateDatacenterVnet/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/ro
uteTables/OutsideRoutetable",
    "location": null,
    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}

```

Etapa 6. Crie um roteador CSR1000v.

Cada VM precisa ter 2 interfaces (internas e externas), o que significa 2 NICs por VM. Crie as 2 NICs e associe um IP público à NIC externa.

1. Crie o endereço IP público.

```

$ az network public-ip create --name CSRAPublicIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"38306703-153b-456b-b2e4-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-

```

```

xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRA",
  "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. Crie a NIC externa e associe o endereço IP público a ela.

```

$ az network nic create --name CSRAOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address CSRAPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "plk2sxe5i01lcksytfab.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/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\06fd60de-6547-4992-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface/ipConfigurations/ipconfig1",
        "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/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu

```

```

blicIPAddresses/CSRAPublicIP",
  "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/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
  "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. Crie a NIC interna.

```

$ az network nic create --name CSRAInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "g1lzklk2sxe5i011ccksytfab.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/ne
tworkInterfaces/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/ne
tworkInterfaces/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/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
          "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. Liste as imagens CSR1000v disponíveis no Azure. Este exemplo usa o nome urn de **cisco:cisco-csr-1000v:16_7:16.7.120171201**.

```

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. Implante o CSR1000v com o nome de urn da imagem.

```
$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRA --location southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics CSRAOutsideInterface CSRAInsideInterface --admin-username cisco --admin-password "Cisco1234567" --authentication-type password
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": ""
}
```

Após alguns minutos, o novo CSR1000v é inicializado.

```
$ 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. Faça login no CSR1000v e verifique a funcionalidade.

```
$ 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
```

Passo 7. Crie o segundo roteador CSR1000v.

1. Crie o endereço IP público.

```
$ az network public-ip create --name CSRBPUBLICIP --resource-group CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "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. Crie a NIC externa e associe o endereço IP público a ela.

```

$ az network nic create --name CSRBOURSIDEINTERFACE --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address CSRBPUBLICIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzkplk2sxe5i011ccksytfab.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/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBOURSIDEINTERFACE",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBOURSIDEINTERFACE/ipConfigurations/ipconfig1",
        "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/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/CSRBPUBLICIP",
          "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/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "CSRBOutsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "c3f05156-ad07-4abd-a006-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

3. Crie a NIC interna.

```

$ az network nic create --name CSRBIInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i01lcksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"15edf738-fc77-431c-80f3-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBIInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,

```



```

    "applicationSecurityGroups": null,
    "etag": "W/\\"15edf738-fc77-431c-80f3-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBInsideInterface/ipConfigurations/ipconfig1",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.1.5",
    "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/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
      "ipConfigurations": null,
      "name": null,
      "networkSecurityGroup": null,
      "provisioningState": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null,
      "serviceEndpoints": null
    }
  }
},
  "location": "southcentralus",
  "macAddress": null,
  "name": "CSRBInsideInterface",
  "networkSecurityGroup": null,
  "primary": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceGuid": "085c88fc-9e78-49be-a5a7-xxxxxxxxxxxx",
  "tags": null,
  "type": "Microsoft.Network/networkInterfaces",
  "virtualMachine": null
}
}

```

4. Implante o segundo CSR1000v com a mesma imagem **cisco:cisco-csr-1000v:16_7:16.7.120171201**.

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRB --location
southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics
CSRBOutsideInterface CSRBInsideInterface --admin-username cisco --admin-password
"Cisc0l234567" --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/vi
rtualMachines/CSRB",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-8C-51,00-0D-3A-5D-85-2A",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.5,192.168.1.5",
  "publicIpAddress": "23.100.122.102",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

```
}
```

Etapa 8. Crie uma VM de host com o mesmo procedimento na etapa 6. Este exemplo usa UbuntuLTS.

1. Crie o endereço IP público.

```
$ az network public-ip create --name VMHostPublicIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\\"5943a230-1eeb-4cf0-b856-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/VMHostPublicIP",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "104.215.77.207",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "VMHostPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "ea19c10a-2fd3-498f-b984-xxxxxxxxxxxx",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}
```

2. Crie a NIC externa e associe a OutsideSubnet e o endereço IP público a ela. Quando as sub-redes são associadas às NICs, um endereço IP é automaticamente atribuído à NIC. Neste exemplo, OutsideSubnet é 192.168.2.0/24 e o endereço IP atribuído automaticamente à NIC é 192.168.2.6.

```
$ az network nic create --name VMHostOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address VMHostPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gzkplk2sxe5i011ccksytfab.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/providers/Microsoft.Network/networkInterfaces/VMHostOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,

```

```
"applicationSecurityGroups": null,
"etag": "W/\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",
"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostOutsideInterface/ipConfigurations/ipconfig1",
"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/providers/Microsoft.Network/pu
blicIPAddresses/VMHostPublicIP",
  "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/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
  "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. Crie a NIC interna.

```
$ az network nic create --name VMHostInsideInterface --resource-group  
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet  
{  
  "NewNIC": {  
    "dnsSettings": {  
      "appliedDnsServers": [],  
      "dnsServers": [],  
      "internalDnsNameLabel": null,  
      "internalDomainNameSuffix": "zkplk2sxe5i01lckcsytfab.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/ne  
tworkInterfaces/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/ne  
tworkInterfaces/VMHostInsideInterface/ipConfigurations/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/vi  
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",  
          "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. Implante a VM do Ubuntu. Este exemplo usa UbuntuLTS.

```
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
UrnAlias	Version		
CentOS	OpenLogic	7.3	OpenLogic:CentOS:7.3:latest
CentOS	latest		
CoreOS	CoreOS	Stable	CoreOS:CoreOS:Stable:latest
CoreOS	latest		
Debian	credativ	8	credativ:Debian:8:latest
Debian	latest		
openSUSE-Leap	SUSE	42.3	SUSE:openSUSE-Leap:42.3:latest
openSUSE-Leap	latest		
RHEL	RedHat	7.3	RedHat:RHEL:7.3:latest
RHEL	latest		
SLES	SUSE	12-SP2	SUSE:SLES:12-SP2:latest
SLES	latest		
UbuntuServer	Canonical	16.04-LTS	Canonical:UbuntuServer:16.04-
LTS:latest		UbuntuLTS	latest
WindowsServer	MicrosoftWindowsServer	2016-Datacenter	
MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest			Win2016Datacenter latest
WindowsServer	MicrosoftWindowsServer	2012-R2-Datacenter	
MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest			Win2012R2Datacenter latest
WindowsServer	MicrosoftWindowsServer	2012-Datacenter	
MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest			Win2012Datacenter latest
WindowsServer	MicrosoftWindowsServer	2008-R2-SP1	
MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest			Win2008R2SP1 latest

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name VmHost --location southcentralus --image UbuntuLTS --admin-user cisco --admin-password Cisco1234567 --nics VMHostOutsideInterface VMHostInsideInterface --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/virtualMachines/VmHost",
  "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": ""
}

```

Etapa 9. Adicione rotas a tabelas de roteamento e VMs.

1. Adicione uma rota padrão para a sub-rede interna para rotar o tráfego através do CSR A definindo o endereço IP do próximo salto como 192.168.1.4. Isso é feito na InsideRouteTable.

```

$ az network route-table route create --address-prefix 8.8.8.8/32 --name default_route --
next-hop-type VirtualAppliance --resource-group CorporateDatacenterResourceGroup --route-
table-name InsideRouteTable --next-hop-ip-address 192.168.1.4
{
  "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/ro
uteTables/InsideRouteTable/routes/default_route",
  "name": "default_route",
  "nextHopIpAddress": "192.168.1.4",
  "nextHopType": "VirtualAppliance",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup"
}

```

2. Adicione uma rota para o tráfego na rede para acessar a Internet na OutsideRouteTable.

```

$ az network route-table route create --address-prefix 8.8.8.8/32 --name internet --next-
hop-type Internet --resource-group CorporateDatacenterResourceGroup --route-table-name
OutsideRouteTable
{
  "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/ro
uteTables/OutsideRouteTable/routes/internet",
  "name": "internet",
  "nextHopIpAddress": null,
  "nextHopType": "Internet",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup"
}

```

3. Faça login na VM do Ubuntu e adicione uma rota para forçar o tráfego através da interface interna para 8.8.8.8. A tabela de rotas do Azure usa automaticamente o primeiro IP em uma sub-rede como seu gateway. A sub-rede da interface interna (eth1) é 192.168.1.0/24, o que significa que 192.168.1.1 é o endereço gw padrão para a VM do host.

```

$ 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

Modelo (Adicionando cidr ip)

```

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

```

Note: O NAT deve ser configurado nos roteadores CSR1000v na Etapa 10 para fazer ping na Internet (8.8.8.8). **Note:** As Etapas 10-14 abrangem a configuração dos roteadores CSR1000v para HA. As etapas abreviadas do [Guia de implantação do Cisco CSR 1000v para Microsoft Azure](#) são fornecidas a partir de Configure a Trustpool. Visite o guia para obter detalhes completos.

Etapa 10. Configure os roteadores CSR1000v.

1. Configurar um Trustpool em ambos os roteadores CSR1000v

```

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. Configure um túnel ipsec entre os roteadores Cisco CSR 1000v e ative a detecção de encaminhamento bidirecional (BFD) e um protocolo de roteamento (EIGRP ou BGP) no túnel entre os roteadores para detecção de falha de peer. **Note:** O endereço destino do túnel na configuração é o endereço IP público do peer CSR. Configuração de CSRA

```

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 Tunnell
  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
Configuração do CSRB
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 Tunnell
  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. A mesma configuração para NAT e roteamento é usada nos roteadores CSR1000v. Isso é para acessibilidade da Internet da VM através da interface interna.

```

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. Adicione controles de acesso (IAM) para uma tabela de rotas. No AzureCLI, permita que o aplicativo (CSRA e CSRB) modifique a InsideRouteTable no Azure durante um failover.

Observe a **id** da InsideRouteTable a ser usada como a opção **—scopes** na próxima seção.

```

$ 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/ro
uteTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
  ...

```


Modelo (Adicionando ID de assinatura)

```
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-xxxxxxxxxxxxx
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

5. Crie a função IAM para InsideRouteTable. A opção **—scopes** é retirada do campo **id** da saída anterior. Observe o **app-id**, a **senha** (que é o **app-key**) e o **ID do espaço**.

```
$ az ad sp create-for-rbac -n "InsideRouteTableIAM" --role "network contributor" --scopes
/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/InsideRoutetable --years 2099
{
"appId": "576dd4f1-c08d-xxxx-xxxx-xxxxxxxxxxxxx",
"displayname": "InsideRouteTableIAM",
"name": "http://InsideRouteTableIAM",
"password": "aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx",
"tenant": "ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx"
}
```

Modelo (Adicionando app-key, app-id e ID do espaço)

```
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. Configure a redundância de nuvem em ambos os roteadores. A única diferença entre a configuração em ambos os roteadores são os peers bfd e o gateway padrão. Configuração de CSRA

```
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
```

Configuração do CSRB

```
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-xxxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup
```

Verificar alta disponibilidade

1. Verifique as configurações de BFD e de nuvem.

```
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
Tunnell            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
Tunnell            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          = Tunnell
resource group    = CorporateDatacenterResourceGroup
subscription id   = 09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxxxx
tenant id         = ae49849c-2622-4d45-b95e-xxxxxxxxxxxxxx
application id    = 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxxxxxxx
application key   = aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxxx
route-table      = InsideRoutetable
cidr              = 8.8.8.8/32
Default Gateway IP = 192.168.1.4
```

2. Execute um ping e um traceroute da VM para o destino. Verifique se o ping está na interface eth1 interna.

```
$ 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. O Traceroute mostra que o caminho da VM para 8.8.8.8 é através da interface interna do CSRA.

```
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. Desative a interface do túnel 1 do CSRA para simular um failover.

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

5. Observe que o tráfego agora flui através da interface privada do CSRB.

```
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
```

Note: A nuvem do Azure pode apresentar um atraso durante o failover. O atraso não deve exceder 1 minuto.

Troubleshoot

- Ative as depurações para observar mensagens durante o failover de HA.

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

- Os erros de autenticação e credencial são devidos a controles de acesso inválidos que permitem que o CSR1000v faça chamadas de API para a tabela de rotas do Azure. Verifique se as IDs corretas estão configuradas na etapa 10.

```
*Jul 13 23:29:53.365: CLOUD-HA : res content iov_len=449
iov_base={"error":"invalid_client","error_description":"AADSTS70002:
Error validating credentials. AADSTS50012: Invalid client secret is provided.\r\nTrace ID:
56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx\r\n
Correlation ID: cce94817-29eb-4ebd-833a-\r\nTimestamp: 2018-07-13
23:29:54Z","error_codes":[70002,50012],"timestamp":"2018-07-13
23:29:54Z","trace_id":"56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx","correlation_id":"cce94817-29eb-
4ebd-833a"}

```

Informações Relacionadas

- [Azure CLI 2.0](#)
- [Guia de implantação do Cisco CSR 1000v para Microsoft Azure](#)
- [Escolha da ferramenta certa para os comandos do Azure e do Azure CLI e do PowerShell lado a lado](#)