Guía de implementación de redundancia de HA CSR1000v en Microsoft Azure con AzureCLI 2.0

Contenido

Introducción

Este documento proporciona una guía de configuración paso a paso sobre cómo implementar routers CSR1000v para alta disponibilidad en la nube de Microsoft Azure con AzureCLI 2.0. Su objetivo es ofrecer a los usuarios conocimientos prácticos sobre HA y la capacidad de implementar un banco de pruebas completamente funcional.

Hay varios métodos para implementar imágenes en Azure y el método más familiar para la mayoría de los usuarios es a través del portal web. Sin embargo, AzureCLI es una herramienta rápida y potente una vez que esté familiarizado con ella.

Para obtener información más detallada sobre Azure, cómo implementar un CSR1000v a través del portal web y HA, consulte la Guía de implementación de Cisco CSR 1000v para Microsoft

Azure e información relacionada.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Una cuenta de Microsoft Azure
- 2 CSR1000v y 1 máquina virtual Windows/Linux
- AzureCLI 2.0

Componentes Utilizados

La información en este documento se basa en 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. Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Objetivo

Implemente 2 routers CSR1000v y 1 VM (windows/linux). Simula el tráfico continuo desde el Data Center privado (VM) a Internet (8.8.8.8). Simule una conmutación por fallas HA y observe que HA ha tenido éxito al confirmar que la tabla de ruteo de Azure ha conmutado el tráfico de CSR-A a la interfaz privada de CSR-B.

Topología

Para comprender completamente la topología y el diseño es importante antes del inicio de la configuración. Esto ayuda a resolver cualquier problema potencial más adelante.

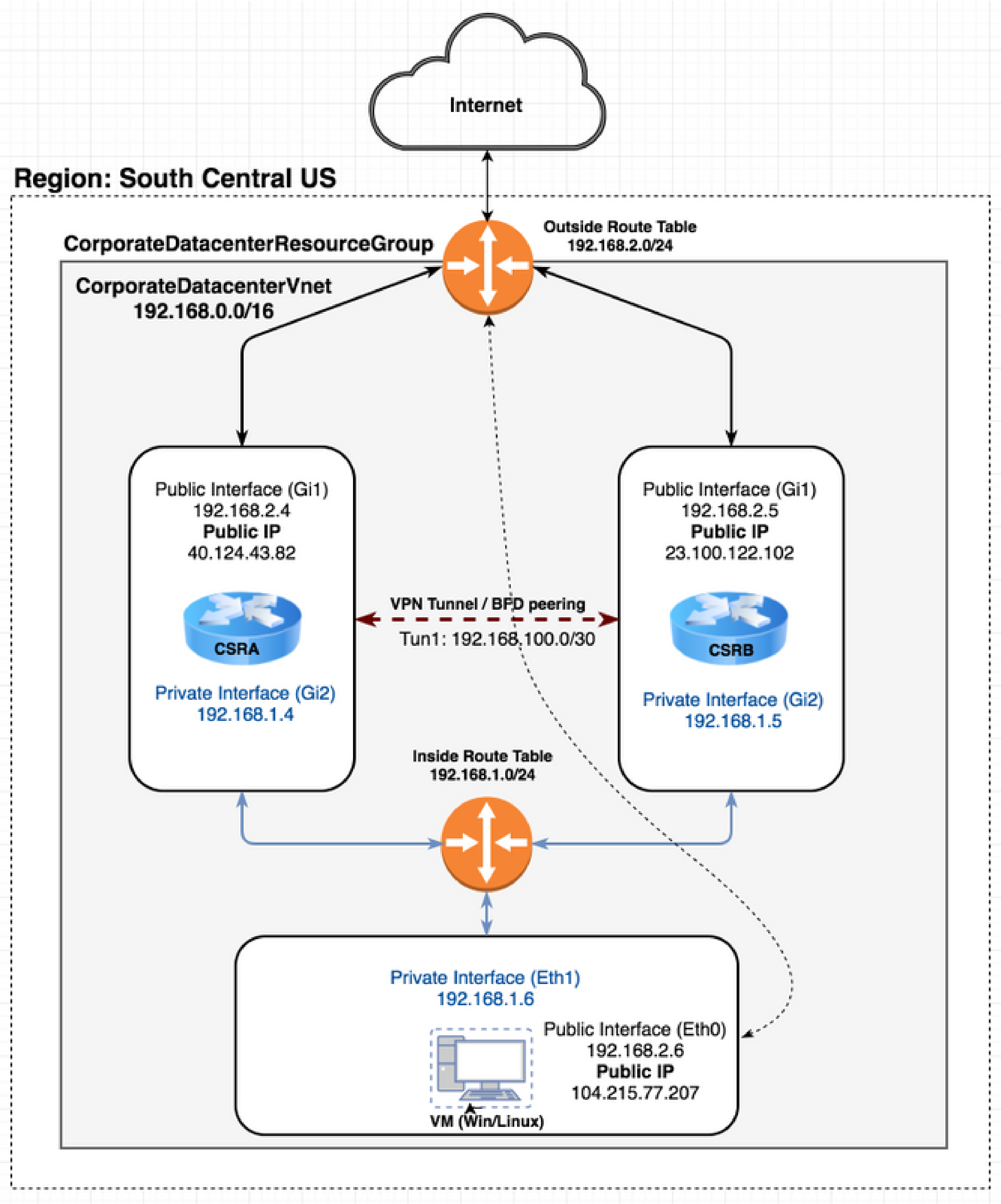
Puede haber varios escenarios de implementaciones de HA en función de los requisitos del usuario. Para este ejemplo, configure la redundancia HA con estos parámetros:

- 1x - Región (centro sur de EE. UU.)
- 1x - Grupo de recursos (CorporateDatacenterResourceGroup)
- 1 Vnet (CorporateDatacenterVnet)
- 6 interfaces de red (3 interfaces internas y 3 interfaces exteriores)
- 2x - Tablas de ruta (InsideRoutetable y OutsideRoutetable)
- 2x - Routers CSR1000v (Cisco IOS-XE® Denali 16.7.1)
- 1x - VM (Linux/Windows)

Por ahora, el acceso a Internet a través de la interfaz pública se deja habilitado en la máquina virtual para que pueda acceder y configurarlo. Generalmente, todo el tráfico normal debe fluir a través de la tabla de ruta privada. La interfaz pública de la máquina virtual puede desactivarse posteriormente para que no se filtre tráfico accidentalmente.

La simulación de tráfico se realiza realizando ping desde la interfaz privada de la VM → tabla de ruta interna → CSRA → 8.8.8.8. En un escenario de failover, observe que la tabla de ruta privada ha conmutado la ruta para señalar a la interfaz privada de CSRB.

Diagrama de la red



Terminology

- Grupo de recursos: esta es una forma de que Azure realice un seguimiento de todos sus recursos, como máquinas virtuales y vnets. Normalmente se utiliza para administrar todos los artículos y para realizar un seguimiento de los cargos.
- Vnet: red virtual (similar a VPC en la terminología aws)
- Tabla de rutas: contiene las reglas para una subred y puede reenviar tráfico específico a una dirección IP o actuar como un punto final de VPN.

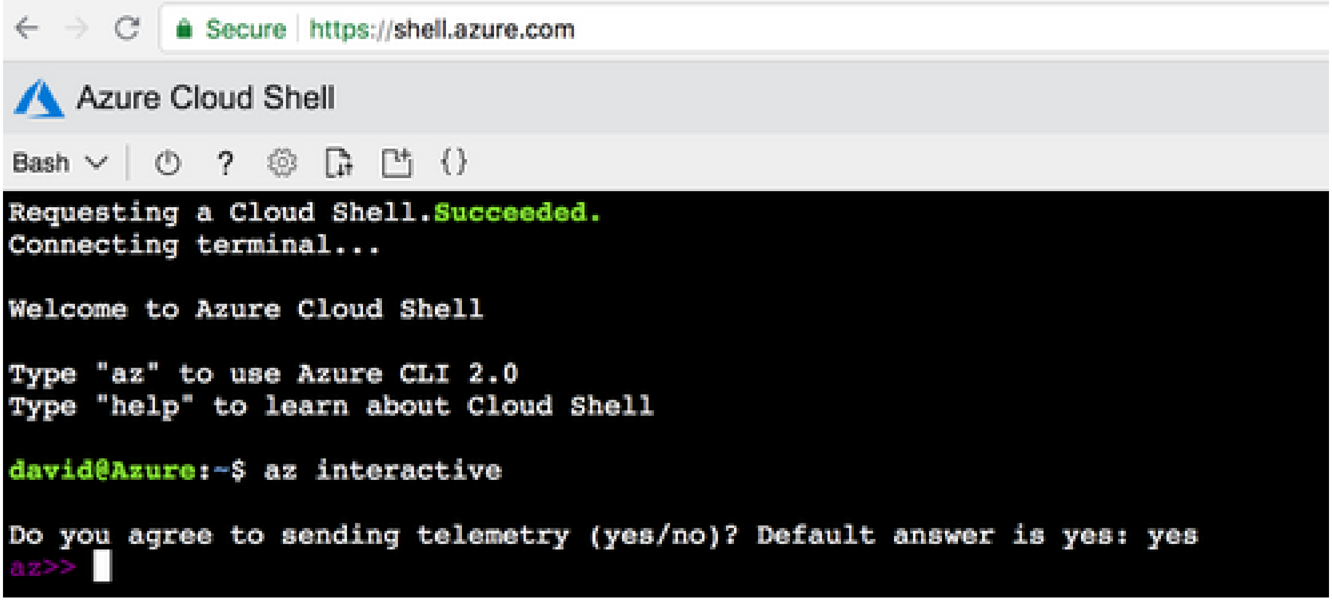
Restricciones

- El propio Azure puede introducir aproximadamente un retraso de 40-50 segundos en un failover de HA.

Configuración

Hay algunos métodos para implementar VM en Azure:

1. Portal Web - Documentación de HA en cisco.com
2. Powershell - Modelo basado en la línea de comandos para administrar los recursos de Azure.
3. AzureCLI 2.0 - También basado en la línea de comandos. Es de código abierto y está escrito en python y debe instalarse en el sistema local. Para escribir este documento, AzureCLI 2.0 es la última versión.
4. Azure Cloud Shell: elija la opción **Bash Shell** en lugar de la **opción PowerShell** para utilizar AzureCLI a través del shell. No es necesaria ninguna instalación para este método.



```
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>>
```

Powershell y AzureCLI son similares, pero los comandos para AzureCLI son más sencillos. Ambos se pueden ejecutar en Windows, MacOS, Linux. Refiérase a Elegir la herramienta adecuada para Azure y junto a los comandos de Azure CLI y PowerShell para realizar una comparación.

Para este ejemplo, implemente todos los recursos con AzureCLI o Cloud Shell. AzureCLI se puede instalar en MacOS, Windows o Linux con pasos ligeramente diferentes. No hay diferencia en la configuración a través del resto del procedimiento entre AzureCLI y 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
```

> **Nota:** Esta plantilla es útil para realizar un seguimiento de todos los ID y la configuración que se utiliza posteriormente para configurar HA en los CSR.

Overview

Paso 1. Instale AzureCLI 2.0.

1. Siga los pasos de instalación para Windows, MacOS o Linux en la documentación de AzureCLI 2.0.
2. Para MacOS:
   ```
   $ brew update && brew install azure-cli
   ```
3. Inicie sesión en Azure y siga las instrucciones para autenticar la sesión.
   ```
   $ az login
   ```
4. Una vez completada la autenticación del explorador, la información de suscripción de Azure se devuelve en formato JSON:
   ```
   [
     {
       "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. Antes de comenzar con el resto de los pasos de configuración, aquí encontrará algunos comandos y consejos útiles sobre AzureCLI.

- Para obtener ayuda con los subcomandos disponibles y lo que hacen, utilice la opción **-h**.

```
$ az account -h
```

- Todos los resultados se devuelven en formato JSON de forma predeterminada. Para facilitar la lectura, puede utilizar la opción **—tabla de resultados** para mostrarla en una tabla.

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

- Obtenga una lista de todas las vm disponibles o reemplace la opción —**all** por una de las otras opciones a continuación para filtrar la tabla.

```
$ 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.
```

- Refiérase a la documentación de Microsoft [Azure CLI 2.0](#) para obtener información detallada sobre todos los comandos de configuración.

Paso 2. Cree un grupo de recursos.

- Un grupo de recursos es un contenedor que contiene recursos relacionados para una solución de Azure. Asigne un nombre al grupo de recursos y elija una ubicación para implementar el contenedor. Este ejemplo utiliza el centro sur de EE. UU.

```
$ 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
}
```

- Plantilla (Agregar 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
```

Paso 3. Cree una red virtual.

- Un Vnet es un espacio de direcciones IP donde se implementa nuestra red. Este rango se divide en subredes más pequeñas y se asigna a las interfaces. Asigne un nombre al vnet, asígnelo al grupo de recursos creado en el paso 2 y asigne un rango de prefijos. Si no especifica un prefijo, Azure generalmente le asigna 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": []
  }
}
```

Paso 4. Crear tablas de ruta.

1. Cree una tabla de rutas para las 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"
   ```

```
}
```

Plantilla (adición de la tabla de rutas)

```
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. Cree una tabla de rutas para las interfaces de cara externa.

```
$ 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"
}
```

Paso 5. Crear subredes.

1. Cree una subred /24 desde el espacio que asignó al vnet en el paso 3 y, a continuación, asígnele a la Tabla de rutas 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. Cree otra subred /24 desde el espacio asignado para el vnet y asígnele a la tabla de rutas 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
}
```

Paso 6. Cree un router CSR1000v.

Cada VM necesita tener 2 interfaces (interna y externa), lo que significa 2 NIC por VM. Cree las 2 NIC y asocie una IP pública a la NIC externa.

1. Cree la dirección IP pública.

```
$ 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/pu
blicIPAddresses/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. Cree la NIC externa y asocie la dirección IP pública a ella.

```
$ az network nic create --name CSRAOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet  --
public-ip-address CSRAPublicIP
{
  "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/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. Cree la NIC interna.

```
$ az network nic create --name CSRAInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzkplk2sxe5i0l1ccksytfab.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. Enumere las imágenes CSR1000v disponibles en Azure. Este ejemplo utiliza el nombre de ejecución 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. Implemente el CSR1000v con el nombre **urn** de la imagen.

```
$ 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/vi
rtualMachines/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": ""
}
```

Después de unos minutos, se inicia el nuevo CSR1000v.

```
$ 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. Inicie sesión en el CSR1000v y verifique la funcionalidad.

```
$ 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
```

Paso 7. Cree el segundo router CSR1000v.

1. Cree la dirección IP pública.

```
$ 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/pu
blicIPAddresses/CSRB",
    "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. Cree la NIC externa y asocie la dirección IP pública a ella.

```
$ az network nic create --name CSRBOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet  --
public-ip-address CSRBPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzkplk2sxe5i0l1ccksytfab.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/CSRBOutsideInterface",
    "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/CSRBOutsideInterface/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. Cree la NIC interna.

```
$ az network nic create --name CSRBInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i0l1ccksytfab.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/CSRBInsideInterface",
    "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. Implemente el segundo CSR1000v con la misma imagen **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
"Cisco1234567" --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": ""
}
```

Paso 8. Cree una VM host con el mismo procedimiento en el paso 6. Este ejemplo utiliza UbuntuLTS.

1. Cree la dirección IP pública.

```
$ 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/pu
blicIPAddresses/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. Cree la NIC externa y asocie la Subred externa y la dirección IP pública a ella. Cuando se asocian subredes con NIC, se asigna automáticamente una dirección IP a la NIC. En este ejemplo, OutsideSubnet es 192.168.2.0/24 y la dirección IP asignada automáticamente a la NIC es 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": "gzkplk2sxe5i0l1ccksytfab.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/ne
tworkInterfaces/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/networkInterfaces/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/publicIPAddresses/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/virtualNetworks/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. Cree la NIC interna.

```
$ az network nic create --name VMHostInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "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/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. Implemente la VM Ubuntu. Este ejemplo utiliza 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              Sku                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/vi
rtualMachines/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": ""
}
```

Paso 9. Agregue rutas a las tablas de ruteo y VM.

1. Agregue una ruta predeterminada para que la subred interna enrute el tráfico a través del CSR A estableciendo la dirección IP del siguiente salto como 192.168.1.4. Esto se hace en 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. Agregue una ruta para que el tráfico de la red llegue a Internet en 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. Inicie sesión en la VM Ubuntu y agregue una ruta para forzar el tráfico a través de la interfaz interna a 8.8.8.8. La tabla de rutas de Azure utiliza automáticamente la primera IP de una subred como su gateway. La subred de la interfaz interna (eth1) es 192.168.1.0/24 lo que significa que 192.168.1.1 es la dirección gw predeterminada para la máquina virtual 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
```

Plantilla (adición de 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
```

Nota: NAT se debe configurar en los routers CSR1000v en el Paso 10 para hacer ping a Internet (8.8.8.8).**Nota:** Los pasos 10-14 abarcan la configuración de los routers CSR1000v para HA. Los pasos abreviados de la Guía de implementación de Cisco CSR 1000v para Microsoft Azure se proporcionan a partir de Configure a Trustpool. Visite la guía para obtener más información.

Paso 10. Configure los routers CSR1000v.

1. Configuración de un Trustpool en ambos routers 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 un túnel IPSec entre los routers Cisco CSR 1000v y habilite la detección de reenvío bidireccional (BFD) y un protocolo de routing (EIGRP o BGP) en el túnel entre los routers para la detección de fallos entre pares. **Nota:** La dirección de destino del túnel en la configuración es la dirección IP pública del peer CSR.Configuración 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 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
```

Configuración de 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 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. Se utiliza la misma configuración para NAT y routing en ambos routers CSR1000v. Esto es para la disponibilidad de Internet de VM a través de la interfaz 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. Agregar controles de acceso (IAM) para una tabla de rutas. En AzureCLI, permita que la aplicación (CSRA y CSRB) modifique InsideRouteTable en Azure durante una conmutación por error. Observe el **id** de InsideRouteTable que se utilizará como la opción **—scopes** en la siguiente sección.

```
$ 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",
...
```

Plantilla (adición de ID de suscripción)

```
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. Cree la función IAM para InsideRouteTable. La opción **—scopes** se toma del **campo id** del resultado anterior. Observe el **id de aplicación**, la **contraseña** (que es la clave de aplicación) y la **ID de arrendatario**.

```
$ az ad sp create-for-rbac -n "InsideRouteTableIAM" --role "network contributor" --scopes
/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/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"
}
```

Plantilla (Adición de clave de aplicación, ID de aplicación y ID de arrendatario)

```
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-xxxxxxxxxxxx
  app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
  tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
  resource-group CorporateDatacenterResourceGroup
```

6. Configure la redundancia en la nube en ambos routers. La única diferencia entre la configuración en ambos routers son los pares bfd y el gateway predeterminado.

Configuración 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-xxxxxxxxxxxx
  app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
  tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
  resource-group CorporateDatacenterResourceGroup
```

Configuración de 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-xxxxxxxxxxxx
  app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
  tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
  resource-group CorporateDatacenterResourceGroup
```

Verificar alta disponibilidad

1. Verifique las configuraciones de BFD y nube.

```
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-xxxxxxxxxxxx
  tenant id = ae49849c-2622-4d45-b95e-xxxxxxxxxxxx
  application id = 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxxxxx
  application key = aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
  route-table  = InsideRoutetable
  cidr         = 8.8.8.8/32
  Default Gateway IP = 192.168.1.4
```

2. Ejecute un ping y un traceroute desde la máquina virtual hasta el destino. Asegúrese de que el ping esté a través de la interfaz 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. Traceroute muestra que la trayectoria de la VM a 8.8.8.8 es a través de la interfaz interna de 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. Cierre la interfaz de túnel 1 de CSRA para simular un 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 el tráfico ahora fluye a través de la interfaz privada de 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
```

Nota: Azure cloud puede producir un retraso al fallar. La demora no debe superar 1 minuto.

Troubleshoot

- Habilite los debugs para observar los mensajes durante el failover de HA.

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

- Los errores de autenticación y credenciales se deben a controles de acceso no válidos que permiten al CSR1000v realizar llamadas API a la tabla de rutas de Azure. Verifique dos veces que los ID adecuados estén configurados en el paso 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"}
```

Información Relacionada

- Azure CLI 2.0
- Guía de implementación de Cisco CSR 1000v para Microsoft Azure
- Elección de la herramienta adecuada para los comandos Azure CLI y PowerShell en paralelo