Microsoft Azure上带有AzureCLI 2.0的CSR1000v HA冗余部署指南

目录

简介

先决条件

- 要求
- 使用的组件

目标

拓扑

网络图

术语

限制

配置

概述

- 步骤1.安装AzureCLI 2.0。
- 步骤2.创建资源组。
- 步骤3.创建Vnet。
- 步骤4.创建路由表。
- 步骤5.创建子网。
- 步骤6.创建CSR1000v路由器。
  -
- 步骤7.创建第二台CSR1000v路由器。
- 步骤8.使用步骤6中的相同步骤创建主机VM。 本示例使用UbuntuLTS。
- 步骤9.将路由添加到路由表和虚拟机。
- 步骤10.配置CSR1000v路由器。

验证高可用性

故障排除

相关信息

简介

本文档提供了有关如何使用AzureCLI 2.0在Microsoft Azure云中部署CSR1000v路由器以实现高可用性的分步配置指南。 它旨在向用户提供高可用性实用知识以及部署全功能测试平台的能力。

在Azure上部署图像有多种方法，大多数用户最熟悉的方法是通过Web门户。 但是，一旦您熟悉AzureCLI，它就会成为一个快速且功能强大的工具。

有关Azure、如何通过Web门户和HA部署CSR1000v的更多深入背景，请参阅Cisco CSR 1000v Microsoft Azure部署指南和相关信息部分。

先决条件

要求

Cisco 建议您了解以下主题：

- Microsoft Azure帐户
- 2个CSR1000v和1个Windows/Linux虚拟机
- AzureCLI 2.0

使用的组件

本文档中的信息基于Cisco IOS-XE® Denali 16.7.1

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

目标

部署2台CSR1000v路由器和1台虚拟机(windows/linux)。 模拟从专用数据中心(VM)到互联网(8.8.8.8)的连续流量。 模拟HA故障切换，通过确认Azure路由表已将流量从CSR-A切换到CSR-B的专用接口，观察HA已成功。

拓扑

在开始配置之前，为了充分了解拓扑和设计，这一点非常重要。这有助于排除以后可能出现的任何问题。

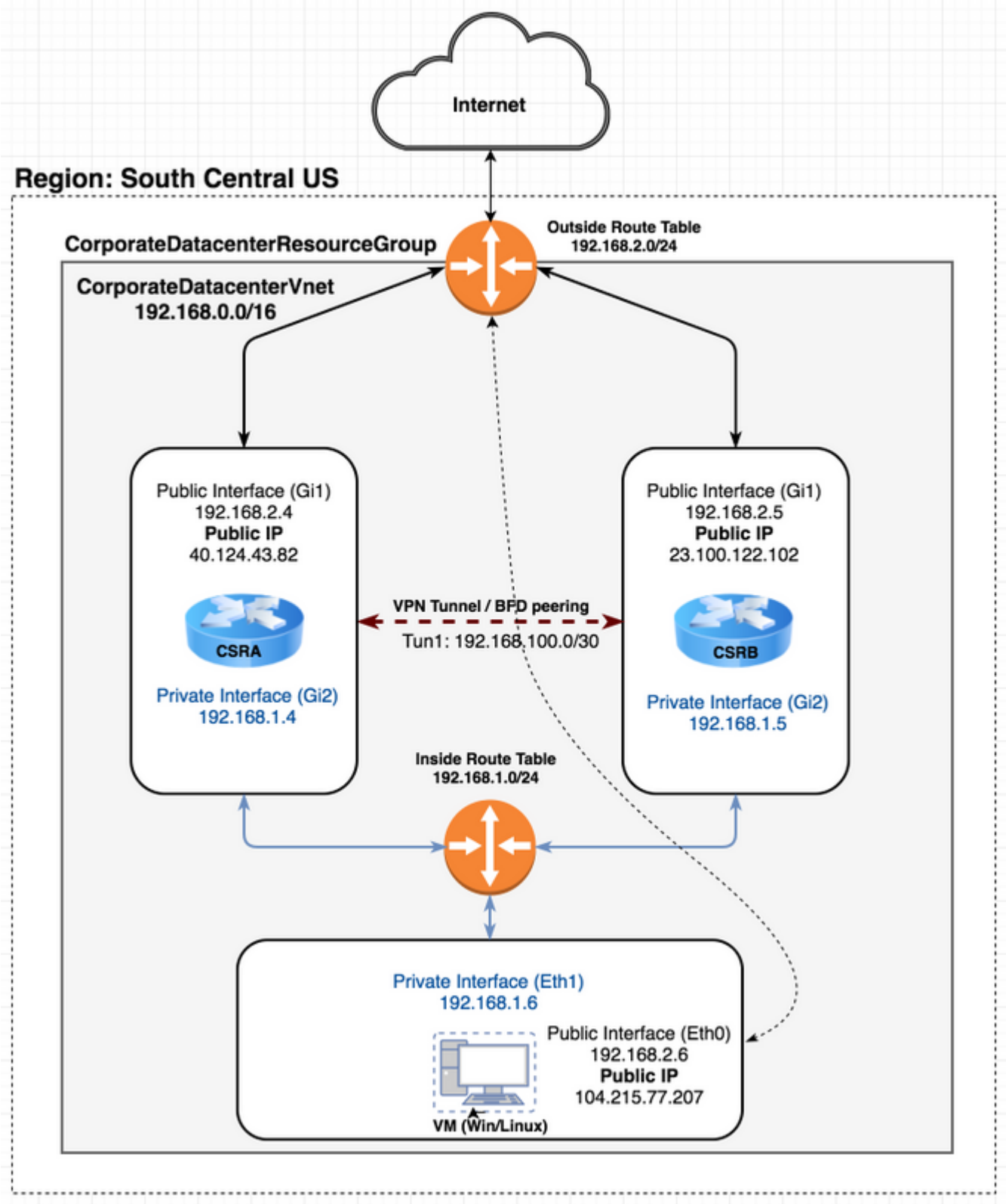
可根据用户需求提供各种HA部署方案。在本示例中，使用以下设置配置HA冗余：

- 1x — 地区（美国中南部）
- 1x — 资源组(CorporateDatacenterResourceGroup)
- 1x - Vnet(CorporateDatacenterVnet)
- 6个 — 网络接口（3个面向内部接口和3个面向外部接口）
- 2x — 路由表（InsideRoutetable和OutsideRoutetable）
- 2x - CSR1000v路由器(Cisco IOS-XE® Denali 16.7.1)
- 1x — 虚拟机(Linux/Windows)

目前，通过公共接口的Internet访问在VM上保持启用状态，以便您可以访问和配置它。通常，所有正常流量应流经专用路由表。稍后可以禁用VM上的公共接口，以免流量意外泄漏。

流量模拟通过从VM的专用接口→路由表内部→ CSRA → 8.8.8执行。在故障切换场景中，观察专用路由表已将路由切换到指向CSRB的专用接口。

网络图



术语

- 资源组 — 这是Azure跟踪所有资源（如虚拟机和vnet）的一种方式。这通常用于管理所有项目并跟踪费用。
- Vnet — 虚拟网络。（类似于aws术语中的VPC）
- 路由表 — 其中包含子网的规则，可将特定流量转发到IP地址或像VPN终端一样操作。

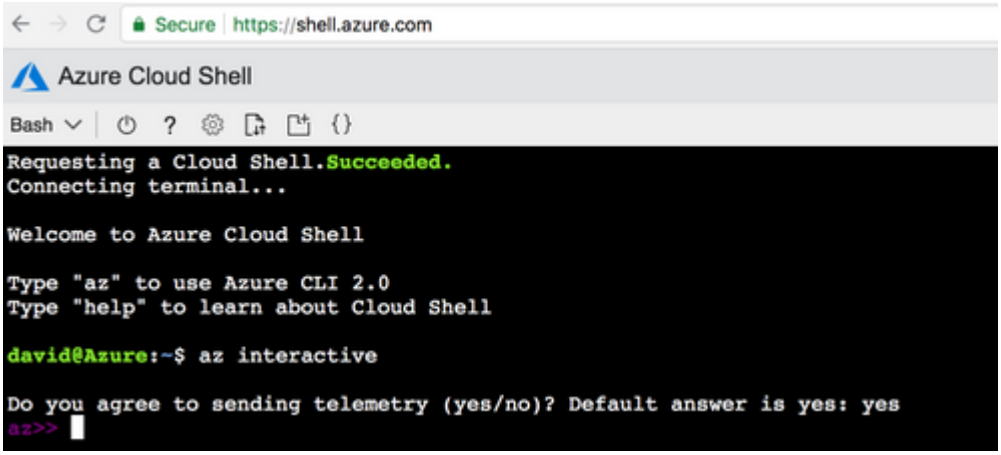
限制

- Azure本身可能在HA故障转移中引入大约40-50秒的延迟。

配置

在Azure上部署虚拟机有几种方法：

1. Web门户- cisco.com上的HA文档
2. Powershell — 用于管理Azure资源的基于命令行的模型。
3. AzureCLI 2.0 -也基于命令行。它是开源的，用python编写，需要安装在本地系统上。要编写此文档，AzureCLI 2.0是最新版本。
4. Azure Cloud Shell — 选择Bash shell选项而不是Powershell选项，以通过外壳使用AzureCLI。无需安装此方法。



Powershell和AzureCLI类似，但AzureCLI的命令更简单。 两者都可以在Windows、MacOS和Linux上运行。 请参阅为Azure选择正确的工具，并排选择Azure CLI和PowerShell命令，进行比较。

对于此示例，请使用AzureCLI或Cloud Shell部署所有资源。 AzureCLI可安装在MacOS、Windows或Linux上，但步骤略有不同。 AzureCLI和Azure Cloud Shell之间的其余步骤在配置方面没有任何差异。

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

注意：此模板有助于跟踪所有ID和配置，稍后用于在CSR上配置HA。

概述

步骤1.安装AzureCLI 2.0。

1. 按照AzureCLI 2.0文档中Windows、MacOS或Linux的安装步骤进行操作。
2. 对于MacOS:

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

3. 登录Azure并按照说明对会话进行身份验证。

```
$ az login
```

4. 浏览器身份验证完成后，将以JSON格式返回您的Azure订阅信息：

```
[
  {
    "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. 在开始执行其余配置步骤之前，请在此列出一些有用的命令和关于AzureCLI的提示。

- 有关可用子命令及其功能的帮助，请使用-h选项。

```
$ az account -h
```

- 默认情况下，所有输出都以JSON格式返回。为便于阅读，可以使用—output table选项在表中显示。

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

- 获取所有可用vm的列表，或者将 — all选项替换为以下其中一个选项以过滤表。

```
$ 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,
```

- 有关所有配置命令的详细信息，请参阅Microsoft的Azure CLI 2.0文档。

步骤2.创建资源组。

- 资源组是为Azure解决方案保存相关资源的容器。 为您的资源组指定一个名称，然后选择一个位置以部署容器。 本示例使用美国中南部。

```
$ 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/CorporateDatacenterResource
  "location": "southcentralus",
  "managedBy": null,
  "name": "CorporateDatacenterResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

- 模板（添加资源组）

```
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
```

步骤3.创建Vnet。

- Vnet是部署网络的ip地址空间。 然后，将此范围拆分为更小的子网并分配给接口。 为vnet指定一个名称，将其分配到步骤2中创建的资源组，并分配前缀范围。 如果不指定前缀，Azure通常会分配给10.0.0.0/16。

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

步骤4.创建路由表。

1. 为面向内部的接口创建路由表。

```
$ az network route-table create --name InsideRoutetable --resource-group CorporateDatacenterResour
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"45088005-cb6f-4356-bb18-xxxxxxxxxxxx\"",
```

```
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterRes
  "location": "southcentralus",
  "name": "InsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/routeTables"
}
```

- 模板（添加路由表）

```
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. 创建面向外部接口的路由表。

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

步骤5.创建子网。

1. 从步骤3中为vnet分配的空间创建/24子网，然后将其分配到内部路由表。

```
$ az network vnet subnet create --address-prefix 192.168.1.0/24 --name InsideSubnet --resource-gro
{
  "addressPrefix": "192.168.1.0/24",
  "etag": "W/\"a0dbd178-3a45-48fb-xxxx-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterRes
  "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/CorporateDatacenterR
    "location": null,
    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}
```


2. 从为vnet分配的空间创建另一个/24子网，并将其分配给外部路由表。


```
$ az network vnet subnet create --address-prefix 192.168.2.0/24 --name OutsideSubnet --resource-gr
{
  "addressPrefix": "192.168.2.0/24",
  "etag": "W/\"874d1019-90a0-44fd-a09c-0aed8f2ede5b\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterRes
  "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/CorporateDatacenterR
    "location": null,
    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}
```


步骤6.创建CSR1000v路由器。

每个VM需要有2个接口（内部和外部），这意味着每个VM有2个NIC。 创建两个NIC并将公共IP与外部NIC关联。

1. 创建公有IP地址。

```
$ az network public-ip create --name CSRAPublicIP --resource-group CorporateDatacenterResourceGrou
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"38306703-153b-456b-b2e4-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR
    "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. 创建外部NIC并将公有IP地址与其关联。

```
$ az network nic create --name CSRAOutsideInterface --resource-group CorporateDatacenterResourceGr
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "plk2sxe5i0l1ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\"06fd60de-6547-4992-b506-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"06fd60de-6547-4992-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacer
        "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/CorporateDatac
          "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/CorporateDatac
          "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. 创建内部网卡。

```
$ az network nic create --name CSRAInsideInterface --resource-group CorporateDatacenterResourceGro
{
  "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/CorporateDatacenterR
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacer
        "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/CorporateDatac
          "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. 列出Azure上可用的CSR1000v映像。 此示例使用urn name 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. 使用映像的urn名称部署CSR1000v。

```
$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRA --location southcentr
 Running ..
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterRes
  "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": ""
}
```

几分钟后，新的CSR1000v启动。

```
$ 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. 登录CSR1000v并验证功能。

```
$ 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
```

步骤7.创建第二台CSR1000v路由器。

1. 创建公有IP地址。

```
$ az network public-ip create --name CSRBPublicIP --resource-group CorporateDatacenterResourceGrou
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"f0f98dac-ea56-4efe-8da6-81a221ac3474\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR
    "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. 创建外部NIC并将公有IP地址与其关联。

```
$ az network nic create --name CSRBOutsideInterface --resource-group CorporateDatacenterResourceGr
{
  "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/CorporateDatacenterR
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacer
        "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/CorporateDatac
          "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/CorporateDatac
          "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. 创建内部网卡。


```
$ az network nic create --name CSRBInsideInterface --resource-group CorporateDatacenterResourceGro
{
  "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/CorporateDatacenterR
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"15edf738-fc77-431c-80f3-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacer
        "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/CorporateDatac
          "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. 使用相同的映像cisco:cisco-csr-1000v:16_7:16.7.120171201部署第二个CSR1000v。

```
$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRB --location southcentr
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterRes
  "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": ""
}
```

步骤8.使用步骤6中的相同步骤创建主机VM。 本示例使用UbuntuLTS。

1. 创建公有IP地址。

```
$ az network public-ip create --name VMHostPublicIP --resource-group CorporateDatacenterResourceGr
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"5943a230-1eeb-4cf0-b856-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterR
    "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. 创建外部NIC并将OutsideSubnet和公共IP地址与其关联。 当子网与NIC关联时，会自动为NIC分配IP地址。在本示例中，OutsideSubnet为192.168.2.0/24，自动分配给网卡的IP地址为192.168.2.6。

```
$ az network nic create --name VMHostOutsideInterface --resource-group CorporateDatacenterResource
{
  "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/CorporateDatacenterR
"ipConfigurations": [
  {
    "applicationGatewayBackendAddressPools": null,
    "applicationSecurityGroups": null,
    "etag": "W/\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacer
    "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/CorporateDatac
      "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/CorporateDatac
      "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. 创建内部网卡。


```
$ az network nic create --name VMHostInsideInterface --resource-group CorporateDatacenterResourceG
{
  "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/CorporateDatacenterR
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacer
        "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/CorporateDatac
          "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. 部署Ubuntu VM。本示例使用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
-------------  ----------------------  ------------------  ----------------------------------------
CentOS         OpenLogic               7.3                 OpenLogic:CentOS:7.3:latest
CoreOS         CoreOS                  Stable              CoreOS:CoreOS:Stable:latest
Debian         credativ                8                   credativ:Debian:8:latest
openSUSE-Leap  SUSE                    42.3                SUSE:openSUSE-Leap:42.3:latest
RHEL           RedHat                  7.3                 RedHat:RHEL:7.3:latest
SLES           SUSE                    12-SP2              SUSE:SLES:12-SP2:latest
UbuntuServer   Canonical               16.04-LTS           Canonical:UbuntuServer:16.04-LTS:latest
WindowsServer  MicrosoftWindowsServer  2016-Datacenter     MicrosoftWindowsServer:WindowsServer:20
WindowsServer  MicrosoftWindowsServer  2012-R2-Datacenter  MicrosoftWindowsServer:WindowsServer:20
WindowsServer  MicrosoftWindowsServer  2012-Datacenter     MicrosoftWindowsServer:WindowsServer:20
WindowsServer  MicrosoftWindowsServer  2008-R2-SP1         MicrosoftWindowsServer:WindowsServer:20
```

```
$ az vm create --resource-group CorporateDatacenterResourceGroup --name VmHost --location southcer
{
 "fqdns": "",
 "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterReso
 "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": ""
}
```

步骤9.将路由添加到路由表和虚拟机。

1. 通过将下一跳IP地址设置为192.168.1.4，为内部子网添加默认路由以通过CSR A路由流量。
   此操作在InsideRouteTable中完成。

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

```
}
```

2. 在OutsideRouteTable上为网络中的流量添加到达Internet的路由。

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

3. 登录Ubuntu VM并添加路由以强制流量通过内部接口到8.8.8.8。 Azure路由表会自动使用子网中的第一个IP作为其网关。 内部接口的(eth1)子网是192.168.1.0/24，这意味着192.168.1.1是主机VM的默认gw地址。

```
$ 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
```

- 模板（添加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
```

注意：在第10步中，必须在CSR1000v路由器上配置NAT，才能对互联网(8.8.8.8)执行ping操作。

注意：第10-14步介绍为HA配置CSR1000v路由器。 Cisco CSR 1000v Microsoft Azure部署指南的缩写步骤从Configure a Trustpool开始。请访问指南了解完整详情。

步骤10.配置CSR1000v路由器。

1. 在两台CSR1000v路由器上配置信任池

```
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. 在Cisco CSR 1000v路由器之间配置ipsec隧道，并在路由器之间的隧道上启用双向转发检测(BFD)和路由协议（EIGRP或BGP），以检测对等体故障。

注意：配置中的隧道目标地址是对等CSR的公共IP地址。

- CSRA配置

```
crypto isakmp policy 1
 encr aes 256
 authentication pre-share
crypto isakmp key cisco address 0.0.0.0
!
crypto ipsec transform-set uni-perf esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile vti-1
 set security-association lifetime kilobytes disable
 set security-association lifetime seconds 86400
 set transform-set uni-perf
 set pfs group2
!
```

```
interface Tunnel1
 ip address 192.168.101.1 255.255.255.252
 bfd interval 500 min_rx 500 multiplier 3
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 23.100.122.102 /* Public IP of the peer CSR */
 tunnel protection ipsec profile vti-1
!
router eigrp 1
 bfd all-interfaces
 network 192.168.101.0
```

- CSRB配置

```
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. 两个CSR1000v路由器上使用相同的NAT和路由配置。这是为了通过内部接口实现虚拟机互联网可达性。

```
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. 为路由表添加访问控制(IAM)。 在AzureCLI中，允许应用程序（CSRA和CSRB）在故障转移

期间修改Azure中的InsideRouteTable。

- 记下InsideRouteTable的id，该表将用作下节中的 — scope选项。

```
$ az network route-table show --resource-group CorporateDatacenterResourceGroup --name InsideRoute
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"f0c85464-bba0-465a-992a-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterRes
  "location": "southcentralus",
  "name": "InsideRoutetable",

...
```

- 模板（添加订用ID）

```
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. 为InsideRouteTable创建IAM角色。-scope选项取自上一输出的id字段。
   - 请注意app-id、password（是应用密钥）和租户id。

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

- 模板（添加应用密钥、应用ID和租户ID）

```
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-xxxxxxxxxxxx
  subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxx
  tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx
  resource-group CorporateDatacenterResourceGroup
```

6. 在两台路由器上配置云冗余。两台路由器上配置之间的唯一区别是bfd对等体和默认网关。
    - CSRA配置

```
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-xxxxxxxxxxxx
  subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxx
  tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx
  resource-group CorporateDatacenterResourceGroup
```

    - CSRB配置

```
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-xxxxxxxxxxxx
  subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
  app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxx
  tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxx
  resource-group CorporateDatacenterResourceGroup
```

验证高可用性

1. 检查BFD和云配置。

```
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-xxxxxxxxxxxx
  route-table  = InsideRoutetable
  cidr         = 8.8.8.8/32
  Default Gateway IP = 192.168.1.4
```

2. 从VM对目标运行ping和traceroute。确保ping通过内部eth1接口。

```
$ 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显示从VM到8.8.8.8的路径通过CSRA的内部接口。

```
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. 关闭CSRA的tunnel 1接口以模拟故障切换。

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

```
CSRA(config-if)#sh
```

5. 观察流量现在流经CSRB的专用接口。

```
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
```

注意：Azure云可能会在故障转移时引入延迟。延迟不应超过1分钟。

故障排除

- 启用调试，以在HA故障切换期间观察消息。

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

- 身份验证和凭据错误是由无效的访问控制造成的，该访问控制允许CSR1000v对Azure路由表进行API调用。 仔细检查步骤10中是否配置了正确的ID。

```
*Jul 13 23:29:53.365: CLOUD-HA : res content iov_len=449 iov_base={"error":"invalid_client","error_desc
Error validating credentials. AADSTS50012: Invalid client secret is provided.\r\nTrace ID: 56873e4b-378
Correlation ID: cce94817-29eb-4ebd-833a-\r\nTimestamp: 2018-07-13 23:29:54Z","error_codes":[70002,50012
23:29:54Z","trace_id":"56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx","correlation_id":"cce94817-29eb-4ebd-833a
```

相关信息

- Azure CLI 2.0
- 适用于Microsoft Azure的思科CSR 1000v部署指南
- 为Azure和并排的Azure CLI和PowerShell命令选择正确的工具

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。