Guide de déploiement de la redondance CSR1000v HA sur Microsoft Azure avec AzureCLI 2.0

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés

Objectif

Topologie

Diagramme du réseau

Terminologie

Restrictions

Configuration

Aperçu

- Étape 1 : installation d'AzureCLI 2.0
- Étape 2. Créer un groupe de ressources
- Étape 3 : création d'un réseau virtuel
- Étape 4 : création de tables de routage
- Étape 5. Création de sous-réseaux
- Étape 6 : création d'un routeur CSR1000v
- _
- Étape 7 : création du deuxième routeur CSR1000v
- Étape 8. Créez une machine virtuelle hôte avec la même procédure à l'étape 6. Cet exemple utilise UbuntuLTS.
- Étape 9 : ajout de routes aux tables de routage et aux machines virtuelles
- Étape 10. Configuration des routeurs CSR1000v

Vérifier la haute disponibilité

Dépannage

Informations connexes

Introduction

Ce document fournit un guide de configuration étape par étape sur la façon de déployer des routeurs CSR1000v pour la haute disponibilité dans le cloud Microsoft Azure avec AzureCLI 2.0. Il vise à donner aux utilisateurs une connaissance pratique de la haute disponibilité et la capacité de déployer un banc d'essai entièrement fonctionnel.

Il existe différentes méthodes pour déployer des images sur Azure et la méthode la plus courante

pour la plupart des utilisateurs est le portail Web. Cependant, AzureCLI est un outil rapide et puissant une fois que vous le connaissez.

Pour obtenir des informations plus détaillées sur Azure, la manière de déployer un CSR1000v via le portail Web et la haute disponibilité, reportez-vous à la section Guide de déploiement Cisco CSR 1000v pour Microsoft Azure et les informations associées.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Un compte Microsoft Azure
- 2 CSR1000v et 1 machine virtuelle Windows/Linux
- AzureCLI 2.0

Composants utilisés

Les informations contenues dans ce document sont basées sur 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 votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Objectif

Déployez 2 routeurs CSR1000v et 1 machine virtuelle (Windows/Linux). Simulez le trafic continu du data center privé (VM) vers Internet (8.8.8.8). Simulez un basculement de HA et observez que HA a réussi en confirmant que la table de routage Azure a commuté le trafic de CSR-A vers l'interface privée de CSR-B.

Topologie

Pour bien comprendre la topologie et la conception, il est important de commencer la configuration. Cela permet de résoudre les problèmes potentiels ultérieurement.

Il peut y avoir différents scénarios de déploiements haute disponibilité en fonction des besoins de l'utilisateur. Pour cet exemple, configurez la redondance haute disponibilité avec les paramètres suivants :

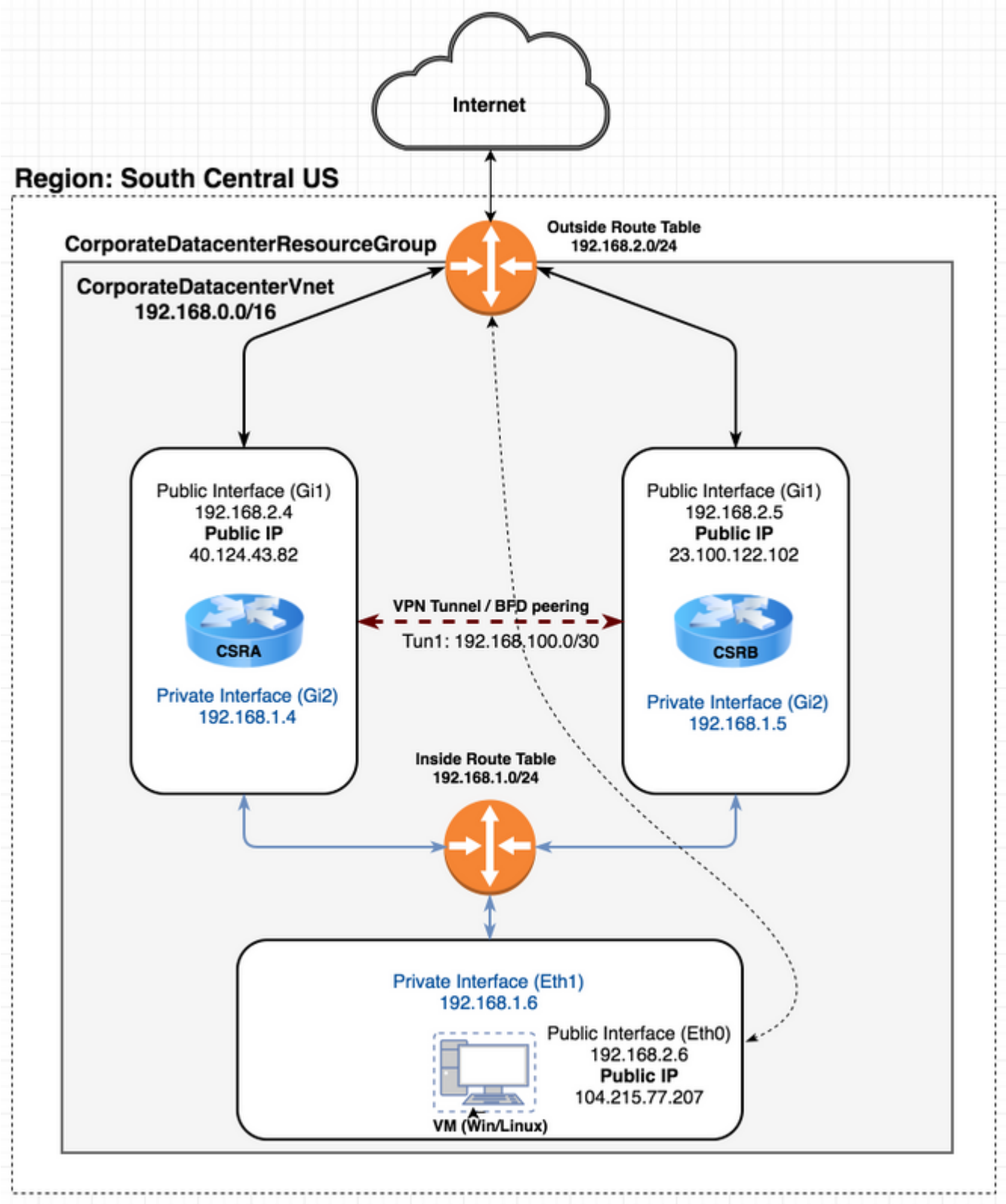
- 1x - Région (Centre-Sud des États-Unis)
- 1x - Groupe de ressources (CorporateDatacenterResourceGroup)
- 1x - Vnet (CorporateDatacenterVnet)
- 6x - Interfaces réseau (3x face interne et 3x face externe)
- 2x - Tables de routage (InsideRoutetable et OutsideRoutetable)

- 2x - Routeurs CSR1000v (Cisco IOS-XE® Denali 16.7.1)
- 1x - VM (Linux/Windows)

Pour l'instant, l'accès à Internet via l'interface publique reste activé sur la machine virtuelle afin que vous puissiez y accéder et la configurer. En règle générale, tout le trafic normal doit transiter par la table de routage privé. L'interface publique de la machine virtuelle peut être désactivée ultérieurement afin d'éviter toute fuite accidentelle de trafic.

La simulation de trafic est effectuée en envoyant une requête ping à partir de l'interface privée de la VM → dans la table de routage → CSRA → 8.8.8.8. Dans un scénario de basculement, observez que la table de routage privée a commuté la route vers l'interface privée de CSRB.

Diagramme du réseau



Terminologie

- Groupe de ressources : Azure peut ainsi effectuer le suivi de toutes vos ressources, telles que les machines virtuelles et les réseaux locaux virtuels. Il est généralement utilisé pour gérer tous les éléments et pour effectuer le suivi des frais.
- Vnet : réseau virtuel (similaire à VPC dans la terminologie AWS).

- Table de routage : contient les règles d'un sous-réseau et peut transférer un trafic spécifique vers une adresse IP ou agir comme un point d'extrémité VPN.

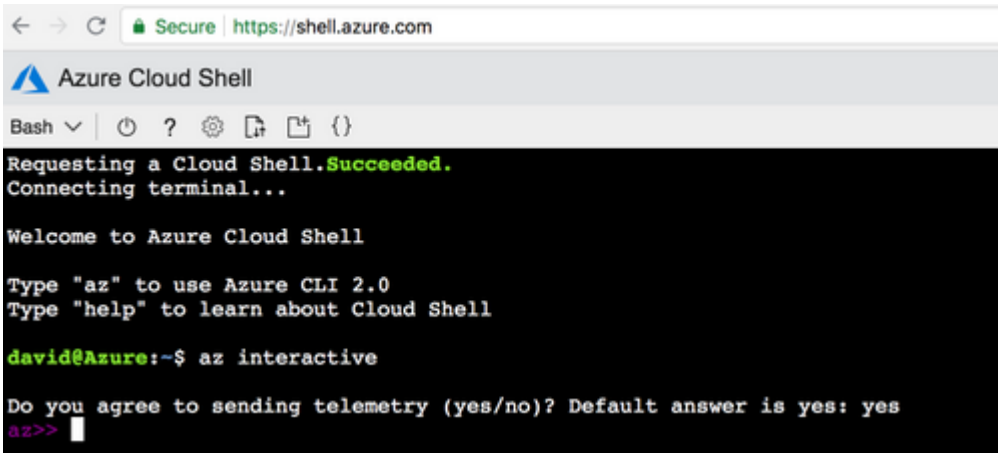
Restrictions

- Azure lui-même peut introduire un délai d'environ 40 à 50 secondes dans un basculement haute disponibilité.

Configuration

Il existe quelques méthodes pour déployer des machines virtuelles sur Azure :

1. Portail Web - Documentation HA sur cisco.com
2. Powershell : modèle basé sur une ligne de commande pour la gestion des ressources Azure.
3. AzureCLI 2.0 - également basé sur la ligne de commande. Il est open source et écrit en python et doit être installé sur votre système local. Afin d'écrire ce document, AzureCLI 2.0 est la dernière version.
4. Azure Cloud Shell - Choisissez l'option Bash shell au lieu de l'option Powershell pour utiliser AzureCLI via l'interpréteur de commandes. Aucune installation n'est nécessaire pour cette méthode.



Powershell et AzureCLI sont similaires, mais les commandes pour AzureCLI sont plus simples. Les deux peuvent fonctionner sous Windows, MacOS, Linux. Référez-vous à Choix de l'outillage approprié pour Azure et aux commandes côte à côte de l'interface de ligne de commande Azure et PowerShell pour une comparaison.

Pour cet exemple, déployez toutes les ressources avec AzureCLI ou Cloud Shell. AzureCLI peut être installé sur MacOS, Windows ou Linux avec des étapes légèrement différentes. Il n'y a aucune différence de configuration dans le reste de la procédure entre AzureCLI et 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
```

Remarque : Ce modèle est utile pour effectuer le suivi de tous les ID et de la configuration qui seront utilisés ultérieurement pour configurer la haute disponibilité sur les CSR.

Aperçu

Étape 1 : installation d'AzureCLI 2.0

1. Suivez les étapes d'installation pour Windows, MacOS ou Linux dans la documentation AzureCLI 2.0.
2. Pour MacOS :

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

3. Connectez-vous à Azure et suivez les instructions pour authentifier votre session.

   ```
   $ az login
   ```

4. Une fois l'authentification du navigateur terminée, vos informations d'abonnement Azure sont renvoyées au format 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. Avant de commencer avec le reste des étapes de configuration, voici quelques commandes et conseils utiles sur AzureCLI.

- Pour obtenir de l'aide sur les sous-commandes disponibles et leur fonction, utilisez l'option -h.

```
$ az account -h
```

- Toutes les sorties sont retournées au format JSON par défaut. Pour une lisibilité plus facile, vous pouvez utiliser l'option —output table pour afficher dans un tableau.

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

- Obtenez une liste de toutes les machines virtuelles disponibles ou remplacez l'option —all par l'une des autres options ci-dessous pour filtrer le tableau.

```
$ 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,
```

- Reportez-vous à la documentation Azure CLI 2.0 de Microsoft pour obtenir des informations détaillées sur toutes les commandes de configuration.

Étape 2. Créer un groupe de ressources

- Un groupe de ressources est un conteneur qui contient des ressources associées pour une solution Azure. Donnez un nom à votre groupe de ressources et choisissez un emplacement pour déployer le conteneur. Cet exemple utilise South Central US.

```
$ 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
}
```

- Modèle (Ajout d'un groupe de ressources)

```
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
```

Étape 3 : création d’un réseau virtuel

- Un réseau virtuel est un espace d’adresses IP où votre réseau est déployé. Cette plage est ensuite divisée en sous-réseaux plus petits et attribuée aux interfaces. Attribuez un nom à votre réseau virtuel, attribuez-le au groupe de ressources créé à l'étape 2 et attribuez une plage de préfixes. Si vous ne spécifiez pas de préfixe, Azure vous attribue généralement 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": []
  }
}
```

Étape 4 : création de tables de routage

1. Créez une table de routage pour les interfaces internes.

```
$ 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"
}
```

   - Modèle (Ajout d'une table de routage)

```
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. Créez une table de routage pour les interfaces externes.

```
$ 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"
}
```

Étape 5. Création de sous-réseaux

1. Créez un sous-réseau /24 à partir de l'espace que vous avez attribué au réseau virtuel à l'étape 3, puis attribuez-le à la table de routage interne.

```
$ 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. Créez un autre sous-réseau /24 à partir de l'espace que vous avez attribué au réseau virtuel et attribuez-le à la table de routage externe.

```
$ 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
}
```

Étape 6 : création d’un routeur CSR1000v

Chaque machine virtuelle doit avoir 2 interfaces (internes et externes), ce qui signifie 2 cartes réseau par machine virtuelle. Créez les deux cartes réseau et associez une adresse IP publique à la carte réseau externe.

1. Créez l'adresse IP publique.

```
$ 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. Créez la carte réseau externe et associez-lui l'adresse IP publique.


```
$ 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. Créez la carte réseau interne.

```
$ 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. Indiquez les images CSR1000v disponibles sur Azure. Cet exemple utilise le nom d'urne 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. Déployez le routeur CSR1000v avec le nom d'urne de l'image.


```
$ 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": ""
}
```

Après quelques minutes, le nouveau routeur CSR1000v démarre.

```
$ 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. Connectez-vous au routeur CSR1000v et vérifiez son fonctionnement.

```
$ 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
```

Étape 7 : création du deuxième routeur CSR1000v

1. Créez l'adresse IP publique.

```
$ 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. Créez la carte réseau externe et associez-lui l'adresse IP publique.

```
$ 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. Créez la carte réseau interne.


```
$ 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. Déployez le deuxième routeur CSR1000v avec la même image cisco:cisco-csr-1000v:16_7:16.7.120171201.


```
$ 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": ""
}
```


Étape 8. Créez une machine virtuelle hôte avec la même procédure à l'étape 6. Cet exemple utilise UbuntuLTS.

1. Créez l'adresse IP publique.


```
$ 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. Créez la carte réseau externe et associez-lui le sous-réseau externe et l'adresse IP publique. Lorsque des sous-réseaux sont associés à des cartes réseau, une adresse IP lui est automatiquement attribuée. Dans cet exemple, OutsideSubnet est 192.168.2.0/24 et l'adresse IP attribuée automatiquement à la carte réseau est 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. Créez la carte réseau interne.


```
$ 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. Déployez la machine virtuelle Ubuntu. Cet exemple utilise 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/CorporateDatacenterResc
 "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": ""
}
```

Étape 9 : ajout de routes aux tables de routage et aux machines virtuelles

1. Ajoutez une route par défaut pour le sous-réseau interne afin d'acheminer le trafic via CSR A en définissant l'adresse IP du tronçon suivant sur 192.168.1.4. Cette opération est effectuée dans InsideRouteTable.

```
$ 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. Ajoutez une route pour le trafic dans le réseau afin d'atteindre Internet sur OutsideRouteTable.

```
$ 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. Connectez-vous à la machine virtuelle Ubuntu et ajoutez une route pour forcer le trafic via l'interface interne vers 8.8.8.8. La table de routage Azure utilise automatiquement la première adresse IP d'un sous-réseau comme passerelle. Le sous-réseau de l'interface interne (eth1) est 192.168.1.0/24, ce qui signifie que 192.168.1.1 est l'adresse gw par défaut

de la machine virtuelle hôte.

```
$ 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
```

- Modèle (ajout 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
```

Remarque : NAT doit être configuré sur les routeurs CSR1000v à l'étape 10 afin d'envoyer une requête ping à Internet (8.8.8.8).

Remarque : Les étapes 10 à 14 traitent de la configuration des routeurs CSR1000v

pour la haute disponibilité. Des étapes abrégées du Guide de déploiement de Cisco CSR 1000v pour Microsoft Azure sont fournies à partir de Configurer un pool de confiance. Visitez le guide pour plus de détails.

Étape 10. Configuration des routeurs CSR1000v

1. Configurer un pool de confiance sur les deux routeurs 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. Configurez un tunnel ipsec entre les routeurs Cisco CSR 1000v et activez la détection de transfert bidirectionnel (BFD) et un protocole de routage (EIGRP ou BGP) sur le tunnel entre les routeurs pour la détection de panne d'homologue.

Remarque : L'adresse de destination du tunnel dans la configuration est l'adresse IP publique de l'homologue CSR.

- Configuration 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
```

- Configuration 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. La même configuration pour NAT et le routage est utilisée sur les deux routeurs CSR1000v. Cela concerne l'accessibilité Internet de la machine virtuelle via l'interface interne.

```
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. Ajouter des contrôles d'accès (IAM) pour une table de routage. Dans AzureCLI, autorisez l'application (CSRA et CSRB) à modifier InsideRouteTable dans Azure lors d'un basculement.
    - Notez l'ID de la table InsideRouteTable à utiliser comme option —scopes dans la section suivante.

```
$ 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",
```

```
...
```

- Modèle (Ajout de l'ID d'abonnement)

```
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. Créez le rôle IAM pour InsideRouteTable. L'option —scopes est tirée du champ id de la sortie précédente.
   - Notez l'app-id, le mot de passe (qui est la clé de l'application) et l'id du locataire.

```
$ 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"
}
```

- Modèle (Ajout de la clé d'application, de l'ID d'application et de l'ID de client)

```
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. Configurez la redondance cloud sur les deux routeurs. La seule différence entre la configuration sur les deux routeurs est les homologues bfd et la passerelle par défaut.
   - Configuration 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-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
```

- Configuration 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-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
```

Vérifier la haute disponibilité

1. Vérifiez les configurations BFD et cloud.

```
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. Exécutez une commande ping et traceroute depuis la machine virtuelle vers la destination. Assurez-vous que la requête ping passe par l'interface interne eth1.

```
$ 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. La commande traceroute indique que le chemin entre la VM et 8.8.8.8 passe par l'interface interne 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. Arrêtez l'interface du tunnel 1 de CSRA pour simuler un basculement.

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

5. Notez que le trafic passe désormais par l'interface privée 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
```

Remarque : Le cloud Azure peut entraîner un délai lors du basculement. Le délai ne doit pas dépasser 1 minute.

Dépannage

- Activer les débogages pour observer les messages pendant le basculement haute disponibilité.

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

- Les erreurs d'authentification et d'informations d'identification sont dues à des contrôles d'accès non valides qui permettent au CSR1000v d'effectuer des appels API vers la table de routage Azure. Vérifiez à nouveau que les ID appropriés sont configurés à l'étape 10.

```
*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-xxxxxxxxxxxxx","correlation_id":"cce94817-29eb-4ebd-833a
```

Informations connexes

- Azure CLI 2.0
- Guide de déploiement de Cisco CSR 1000v pour Microsoft Azure
- Choix de l'outillage approprié pour Azure et commandes côte à côte Azure CLI et PowerShell

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.