

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

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Objectif](#)

[Topologie](#)

[Diagramme du réseau](#)

[Terminologie](#)

[Restrictions](#)

[Configuration](#)

[Aperçu](#)

[Étape 1. Installer AzureCLI 2.0.](#)

[Étape 2. Créez un groupe de ressources.](#)

[Étape 3. Créez un réseau virtuel.](#)

[Étape 4. Créer des tables de routage.](#)

[Étape 5. Créez des sous-réseaux.](#)

[Étape 6. Créez un routeur CSR1000v.](#)

–

[Étape 7. Créez le deuxième routeur CSR1000v.](#)

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

[Étape 9. Ajoutez des routes aux tables de routage et aux machines virtuelles.](#)

[Étape 10. Configurez les routeurs CSR1000v.](#)

[Vérifier la haute disponibilité](#)

[Dépannage](#)

[Informations connexes](#)

Introduction

Ce document fournit un guide de configuration étape par étape sur le déploiement 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 familière pour la plupart des utilisateurs est via le portail Web. Cependant, AzureCLI est un outil rapide et puissant une fois que vous y êtes familier.

Pour plus d'informations sur Azure, comment déployer un CSR1000v via le portail Web et la HA, reportez-vous à la section [Guide de déploiement de Cisco CSR 1000v pour Microsoft Azure](#) et les informations connexes.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

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

Components Used

Les informations de 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). Simuler le trafic continu du data center privé (VM) vers Internet (8.8.8.8). Simuler un basculement de HA et observer 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

Afin de bien comprendre la topologie et la conception est importante avant le début de la configuration. Cela permet de résoudre les problèmes potentiels ultérieurement.

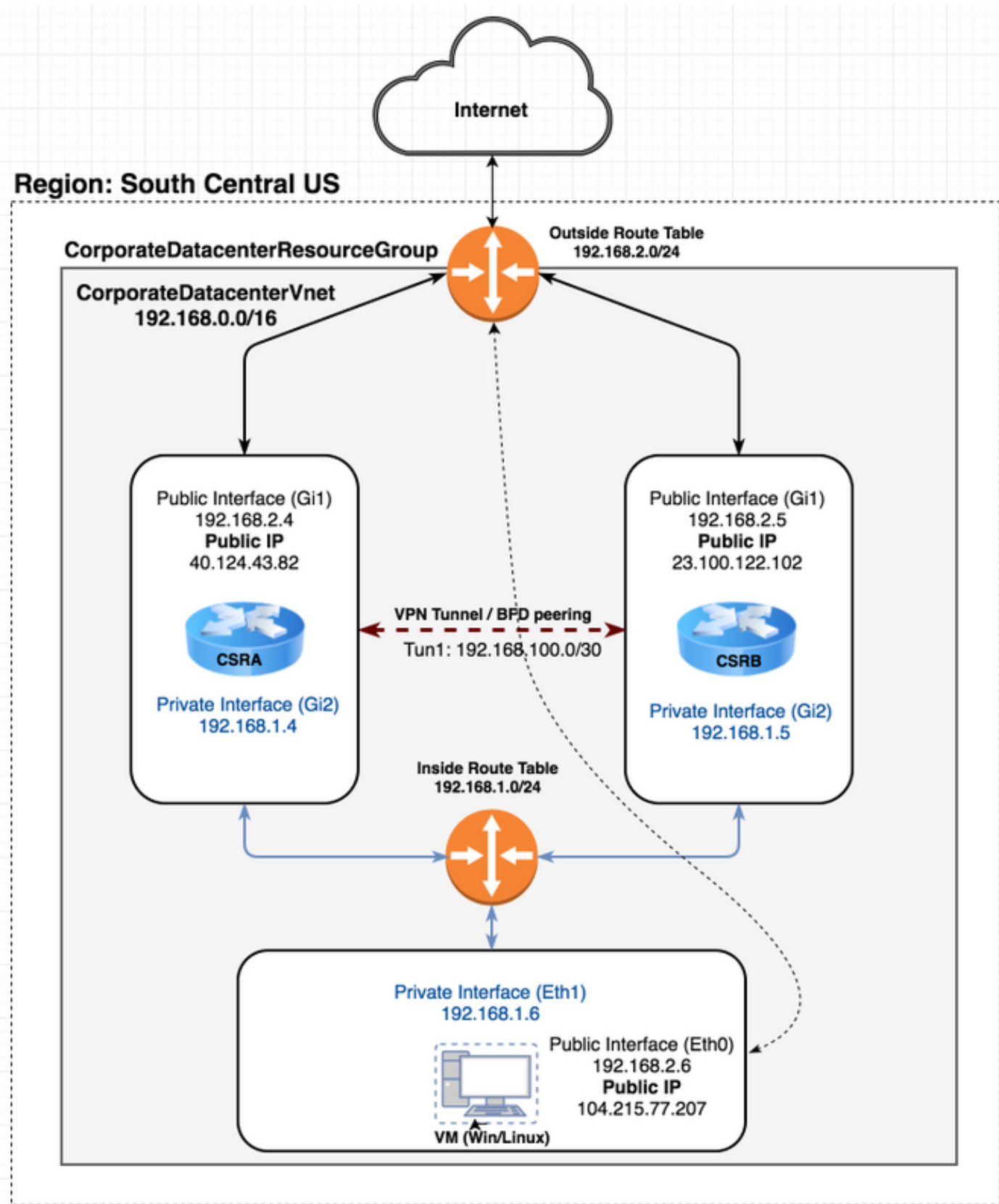
Il peut y avoir différents scénarios de déploiements de haute disponibilité en fonction des besoins de l'utilisateur. Pour cet exemple, configurez la redondance HA avec les paramètres suivants :

- 1x - Région (Sud-Centre des États-Unis)
- 1x - Groupe de ressources (CorporateDatacenterResourceGroup)
- 1 x - Vnet (CorporateDatacenterVnet)
- 6 fois - Interfaces réseau (3 fois face à l'intérieur et 3 fois face à l'extérieur)
- 2x - Tables de routage (InsideRoutetable et OutsideRoutetable)
- 2x - routeurs CSR1000v (Cisco IOS-XE® Denali 16.7.1)
- 1 x - 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 traverser la table de routage privée. L'interface publique de la machine virtuelle peut être ultérieurement désactivée afin qu'aucun trafic ne soit accidentellement divulgué.

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

Diagramme du réseau



Terminologie

- Groupe de ressources - C'est une façon pour Azure de suivre toutes vos ressources telles que les machines virtuelles et les réseaux virtuels. Il est généralement utilisé pour gérer tous les articles et pour suivre les frais.
- Vnet : réseau virtuel.(similaire à VPC dans la terminologie des lois)
- 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.

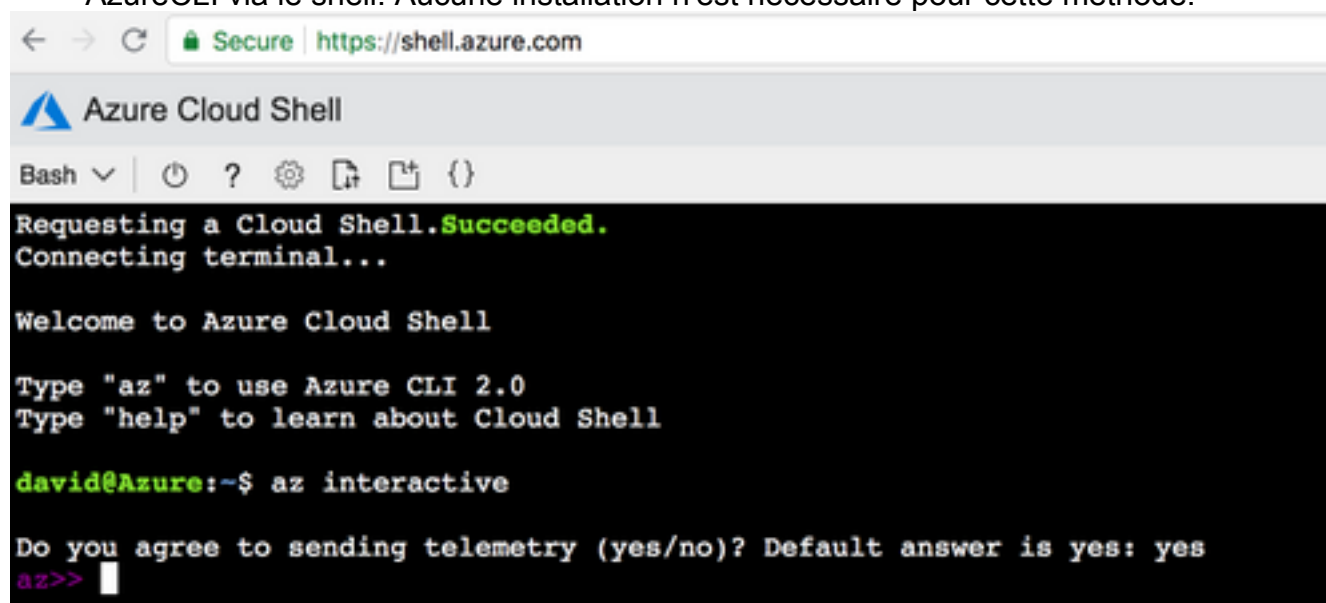
Restrictions

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

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 la 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. Pour 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 le shell. Aucune installation n'est nécessaire pour cette méthode.



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

Welcome to Azure Cloud Shell

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

david@Azure:~$ az interactive

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

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'outil approprié pour Azure et côte à côte pour les commandes CLI et PowerShell Azure](#) 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
```

Note: Ce modèle est utile pour suivre tous les ID et la configuration qui sont utilisés ultérieurement pour configurer la HA sur les CSR.

Aperçu

Étape 1. Installer AzureCLI 2.0.

1. Suivez les étapes d'installation de Windows, MacOS ou Linux dans la documentation [AzureCLI 2.0](#).
2. Pour MacOS :
3. Connectez-vous à Azure et suivez les instructions pour authentifier votre session.
4. Une fois l'authentification du navigateur terminée, vos informations d'abonnement Azure sont retournées au format JSON :

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

5. 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 sur ce qu'elles font, 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, provide '--publisher', '--offer' or '--sku'. Partial name search is supported.

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

Étape 2. Créez 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/CorporateDatacenterResourceGroup",
  "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. Créez un réseau virtuel.

- Un réseau virtuel est un espace d'adresses IP où notre réseau est déployé. Cette plage est ensuite divisée en sous-réseaux plus petits et affectée aux interfaces. Donnez un nom à votre vnet, affectez-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
CorporateDatacenterResourceGroup --address-prefix 192.168.0.0/16
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "192.168.0.0/16"
      ]
    },
    "ddosProtectionPlan": null,
    "dhcpOptions": {
      "dnsServers": []
    },
    "enableDdosProtection": false,
    "enableVmProtection": false,
    "etag": "W/\"7c39a7a9-46e5-4082-a016-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtual
Networks/CorporateDatacenterVnet",
    "location": "southcentralus",
    "name": "CorporateDatacenterVnet",
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "3d95d732-e46a-4fae-a34b-xxxxxxxxxxxx",
    "subnets": [],
    "tags": {},
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
```

Étape 4. Créer des tables de routage.

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

```
$ az network route-table create --name InsideRoutetable --resource-group
CorporateDatacenterResourceGroup
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"45088005-cb6f-4356-bb18-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
```

```

    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": [],
    "subnets": null,
    "tags": null,
    "type": "Microsoft.Network/routeTables"
  }
}

```

Modèle (ajout de 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
CorporateDatacenterResourceGroup
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\a89b6230-9542-468c-b4b2-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/OutsideRoutetable",
  "location": "southcentralus",
  "name": "OutsideRoutetable",
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "routes": [],
  "subnets": null,
  "tags": null,
  "type": "Microsoft.Network/routeTables"
}

```

Étape 5. Créez des sous-réseaux.

1. Créez un sous-réseau /24 à partir de l'espace que vous avez affecté 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-group CorporateDatacenterResourceGroup --vnet-name CorporateDatacenterVnet --
route-table InsideRoutetable
{
  "addressPrefix": "192.168.1.0/24",
  "etag": "W/\a0dbd178-3a45-48fb-xxxx-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
  "ipConfigurations": null,
  "name": "InsideSubnet",
  "networkSecurityGroup": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceNavigationLinks": null,
  "routeTable": {

```



```

    "disableBgpRoutePropagation": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRoutetable",
    "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 affecté au réseau virtuel et affectez-le à la table de routage externe.

```

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

```

Étape 6. Créez un routeur CSR1000v.

Chaque machine virtuelle doit avoir 2 interfaces (internes et externes), ce qui signifie 2 cartes réseau par machine virtuelle. Créez les 2 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
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\\"38306703-153b-456b-b2e4-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRA",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "40.124.43.82",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRAPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "610e1631-331a-4971-8502-xxxxxxxxxxxx",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

2. Créez la carte réseau externe et associez-y l'adresse IP publique.

```

$ az network nic create --name CSRAOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address CSRAPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "plk2sxe5i0llcksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"06fd60de-6547-4992-b506-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"06fd60de-6547-4992-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.2.4",
        "privateIpAddressVersion": "IPv4",

```

```

    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": {
      "dnsSettings": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRAPublicIP",
      "idleTimeoutInMinutes": null,
      "ipAddress": null,
      "ipConfiguration": null,
      "ipTags": null,
      "location": null,
      "name": null,
      "provisioningState": null,
      "publicIpAddressVersion": null,
      "publicIpAllocationMethod": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceGuid": null,
      "sku": null,
      "tags": null,
      "type": null,
      "zones": null
    },
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "subnet": {
      "addressPrefix": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
      "ipConfigurations": null,
      "name": null,
      "networkSecurityGroup": null,
      "provisioningState": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null,
      "serviceEndpoints": null
    }
  }
},
"location": "southcentralus",
"macAddress": null,
"name": "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
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],

```

```

    "dnsServers": [],
    "internalDnsNameLabel": null,
    "internalDomainNameSuffix": "gllzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
    "internalFqdn": null
  },
  "enableAcceleratedNetworking": false,
  "enableIpForwarding": false,
  "etag": "W/\\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/CSRAInsideInterface",
  "ipConfigurations": [
    {
      "applicationGatewayBackendAddressPools": null,
      "applicationSecurityGroups": null,
      "etag": "W/\\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/CSRAInsideInterface/ipConfigurations/ipconfig1",
      "loadBalancerBackendAddressPools": null,
      "loadBalancerInboundNatRules": null,
      "name": "ipconfig1",
      "primary": true,
      "privateIpAddress": "192.168.1.4",
      "privateIpAddressVersion": "IPv4",
      "privateIpAllocationMethod": "Dynamic",
      "provisioningState": "Succeeded",
      "publicIpAddress": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "subnet": {
        "addressPrefix": null,
        "etag": null,
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
        "ipConfigurations": null,
        "name": null,
        "networkSecurityGroup": null,
        "provisioningState": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "resourceNavigationLinks": null,
        "routeTable": null,
        "serviceEndpoints": null
      }
    }
  ],
  "location": "southcentralus",
  "macAddress": null,
  "name": "CSRAInsideInterface",
  "networkSecurityGroup": null,
  "primary": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceGuid": "0f7ae52a-47c3-4563-9fe0-b1484e88296e",
  "tags": null,
  "type": "Microsoft.Network/networkInterfaces",
  "virtualMachine": null
}
}

```

4. Répertoriez 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 CSR1000v avec le nom urn de l'image.

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRA --location
southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics
CSRAOutsideInterface CSRAInsideInterface --admin-username cisco --admin-password
"Cisco1234567" --authentication-type password
Running ..
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/vi
rtualMachines/CSRA",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-83-58,00-0D-3A-5D-89-27",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.4,192.168.1.4",
  "publicIpAddress": "40.124.43.82",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

Après quelques minutes, le nouveau 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 les fonctionnalités.

```

$ 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. Créez le deuxième routeur CSR1000v.

1. Créez l'adresse IP publique.

```

$ az network public-ip create --name CSRBPUBLICIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,

```

```

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

```

2. Créez la carte réseau externe et associez-y l'adresse IP publique.

```

$ az network nic create --name CSRBOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address CSRBPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gllzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\ee0a0b41-42f6-4ac2-91c2-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBOutsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.2.5",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": {
          "dnsSettings": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-

```

```

xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/CSRBPUBLICIP",
    "idleTimeoutInMinutes": null,
    "ipAddress": null,
    "ipConfiguration": null,
    "ipTags": null,
    "location": null,
    "name": null,
    "provisioningState": null,
    "publicIpAddressVersion": null,
    "publicIpAllocationMethod": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": null,
    "sku": null,
    "tags": null,
    "type": null,
    "zones": null
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "CSRBOURSIDEINTERFACE",
"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
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i01lckcsytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,

```



```

    "etag": "W/\\"15edf738-fc77-431c-80f3-xxxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBIInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"15edf738-fc77-431c-80f3-xxxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRBIInsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.5",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {
          "addressPrefix": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
          "ipConfigurations": null,
          "name": null,
          "networkSecurityGroup": null,
          "provisioningState": null,
          "resourceGroup": "CorporateDatacenterResourceGroup",
          "resourceNavigationLinks": null,
          "routeTable": null,
          "serviceEndpoints": null
        }
      }
    ],
    "location": "southcentralus",
    "macAddress": null,
    "name": "CSRBIInsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "085c88fc-9e78-49be-a5a7-xxxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

4. Déployez le deuxième CSR1000v avec la même image **cisco:cisco-csr-1000v:16_7:16.7.120171201**.

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRBI --location
southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics
CSRBOOutsideInterface CSRBIInsideInterface --admin-username cisco --admin-password
"Cisc0l234567" --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/vi
rtualMachines/CSRBI",

```

```

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

```

2. Créez la carte réseau externe et associez-y le OutsideSubnet et l'adresse IP publique.

Lorsque des sous-réseaux sont associés à des cartes réseau, une adresse IP est automatiquement attribuée à la carte réseau. 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
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address VMHostPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,

```

```
"etag": "W/\\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\" ",
"id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostOutsideInterface",
"ipConfigurations": [
  {
    "applicationGatewayBackendAddressPools": null,
    "applicationSecurityGroups": null,
    "etag": "W/\\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\" ",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostOutsideInterface/ipConfigurations/ipconfig1",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.2.6",
    "privateIpAddressVersion": "IPv4",
    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": {
      "dnsSettings": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/VMHostPublicIP",
      "idleTimeoutInMinutes": null,
      "ipAddress": null,
      "ipConfiguration": null,
      "ipTags": null,
      "location": null,
      "name": null,
      "provisioningState": null,
      "publicIpAddressVersion": null,
      "publicIpAllocationMethod": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceGuid": null,
      "sku": null,
      "tags": null,
      "type": null,
      "zones": null
    },
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "VMHostOutsideInterface",
"networkSecurityGroup": null,
```

```

    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "89588a04-6ba6-467d-a86f-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

3. Créez la carte réseau interne.

```

$ az network nic create --name VMHostInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "zkplk2sxe5i01lcksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostInsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxx\\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/VMHostInsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.1.6",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": null,
        "resourceGroup": "CorporateDatacenterResourceGroup",
        "subnet": {
          "addressPrefix": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
          "ipConfigurations": null,
          "name": null,
          "networkSecurityGroup": null,
          "provisioningState": null,
          "resourceGroup": "CorporateDatacenterResourceGroup",
          "resourceNavigationLinks": null,
          "routeTable": null,
          "serviceEndpoints": null
        }
      }
    ]
  }
},
],

```

```

    "location": "southcentralus",
    "macAddress": null,
    "name": "VMHostInsideInterface",
    "networkSecurityGroup": null,
    "primary": null,
    "provisioningState": "Succeeded",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "8ef12cdd-cc31-432e-99cf-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

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

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name VmHost --location
southcentralus --image UbuntuLTS --admin-user cisco --admin-password Cisco1234567 --nics
VMHostOutsideInterface VMHostInsideInterface --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/vi
rtualMachines/VmHost",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-B7-CB,00-0D-3A-5D-B8-9B",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.6,192.168.1.6",
  "publicIpAddress": "104.215.77.207",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

Étape 9. Ajoutez des routes aux tables de routage et aux machines virtuelles.

1. Ajoutez une route par défaut pour le sous-réseau interne pour acheminer le trafic via CSR A en définissant l'adresse IP du tronçon suivant sur 192.168.1.4. Ceci est fait sur `InsideRouteTable`.

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

2. Ajoutez une route pour le trafic sur le réseau pour atteindre Internet sur `OutsideRouteTable`.

```
$ az network route-table route create --address-prefix 8.8.8.8/32 --name internet --next-hop-type Internet --resource-group CorporateDatacenterResourceGroup --route-table-name OutsideRouteTable
{
  "addressPrefix": "8.8.8.8/32",
  "etag": "W/\"d2c7e32e-8d32-4856-a3a6-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/OutsideRouteTable/routes/internet",
  "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 à travers 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 l'adresse ip cidr)

```

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

```

Note: NAT doit être configuré sur les routeurs CSR1000v à l'étape 10 afin d'envoyer une requête ping à Internet (8.8.8.8). **Note:** Les étapes 10 à 14 couvrent 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. Consultez le guide pour plus de détails.

Étape 10. Configurez les routeurs CSR1000v.

1. Configurer un Trustpool 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 défaillance des homologues. **Note:** 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 Tunnell
ip address 192.168.101.1 255.255.255.252
bfd interval 500 min_rx 500 multiplier 3
tunnel source GigabitEthernet1
tunnel mode ipsec ipv4
tunnel destination 23.100.122.102 /* Public IP of the peer CSR */
tunnel protection ipsec profile vti-1
!
router eigrp 1
bfd all-interfaces
network 192.168.101.0

```

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 Tunnell
ip address 192.168.101.2 255.255.255.252
bfd interval 500 min_rx 500 multiplier 3
tunnel source GigabitEthernet1
tunnel mode ipsec ipv4
tunnel destination 40.124.43.82 /* Public IP of the peer CSR */
tunnel protection ipsec profile vti-1
!
router eigrp 1
bfd all-interfaces
network 192.168.101.0

```

3. La même configuration pour NAT et Routing est utilisée sur les deux routeurs CSR1000v. Il s'agit de l'accessibilité Internet des machines virtuelles 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 pendant un basculement. Notez l'id de InsideRouteTable à utiliser comme option **—scopes** dans la section suivante.

```

$ az network route-table show --resource-group CorporateDatacenterResourceGroup --name
InsideRoutetable

```



```
{
  "disableBgpRoutePropagation": false,
  "etag": "W/\"f0c85464-bba0-465a-992a-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRoutetable",
  "location": "southcentralus",
  "name": "InsideRoutetable",
  ...
}
```

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 **—étendues** est prise dans le champ **id** de la sortie précédente. Notez l'**ID d'application**, le **mot de passe** (qui est la clé d'application) et l'**ID de locataire**.

```
$ az ad sp create-for-rbac -n "InsideRouteTableIAM" --role "network contributor" --scopes
/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/routeTables/InsideRoutetable --years 2099
{
  "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 locataire)

```
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 du 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-xxxxxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxxxxx
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
Tunnell	192.168.101.1	YES	manual	up	up

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

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

```
CSRA#show bfd neighbors
```

```
IPv4 Sessions
```

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

```
CSRA#show redundancy cloud provider azure 100
```

```
Cloud HA: work_in_progress=FALSE
Provider : AZURE node 100
State : idle
BFD peer      = 192.168.101.2
BFD intf      = Tunnell
resource group = CorporateDatacenterResourceGroup
subscription id = 09e13fd4-def2-46aa-xxxx-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 de la machine virtuelle vers la destination.

Assurez-vous que la requête ping passe par l'interface eth1 interne.

```
$ 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 montre que le chemin de la machine virtuelle vers 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 tunnell
CSRA(config-if)#sh
```

5. Observez que le trafic passe maintenant par l'interface privée du CSRB.

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

Note: Le nuage Azure peut provoquer un retard lors d'une défaillance. Le délai ne doit pas dépasser 1 minute.

Dépannage

- Activez les débogages pour observer les messages pendant le basculement HA.

```
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 à CSR1000v d'effectuer des appels API vers la table de routage Azure. Vérifiez deux fois 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_description":"AADSTS70002:
Error validating credentials. AADSTS50012: Invalid client secret is provided.\r\nTrace ID:
56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx\r\n
Correlation ID: cce94817-29eb-4ebd-833a-\r\nTimestamp: 2018-07-13
23:29:54Z","error_codes":[70002,50012],"timestamp":"2018-07-13
23:29:54Z","trace_id":"56873e4b-3781-4ee6-8bd9-xxxxxxxxxxxx","correlation_id":"cce94817-29eb-
4ebd-833a"}
```

Informations connexes

- [Azure CLI 2.0](#)
- [Guide de déploiement de Cisco CSR 1000v pour Microsoft Azure](#)
- [Choix de l'outil approprié pour Azure et côte à côte pour les commandes Azure CLI et PowerShell](#)