

AzureCLI 2.0を使用したMicrosoft AzureでのCSR1000v HA冗長導入ガイド

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[目標](#)

[トポロジ](#)

[ネットワーク図](#)

[用語](#)

[制約事項](#)

[コンフィギュレーション](#)

[概要](#)

[ステップ1: AzureCLI 2.0をインストールします。](#)

[ステップ2: リソースグループの作成](#)

[ステップ3: Vnetを作成します。](#)

[ステップ4: ルートテーブルの作成](#)

[ステップ5: サブネットを作成します。](#)

[ステップ6: CSR1000vルータを作成します。](#)

[ステップ7: 2番目のCSR1000vルータを作成します。](#)

[ステップ8: ステップ6と同じ手順でホストVMを作成します。この例では、UbuntuLTSを使用しています。](#)

[ステップ9: ルーティングテーブルとVMにルートを追加します。](#)

[手順10: CSR1000vルータを設定します。](#)

[ハイアベイラビリティの確認](#)

[トラブルシューティング](#)

[関連情報](#)

概要

このドキュメントでは、AzureCLI 2.0を使用してMicrosoft Azureクラウドでハイアベイラビリティ用のCSR1000vルータを展開する方法を順を追って構成ガイドを示します。HAの実践的な知識と、完全に機能するテストベッドの導入機能をユーザに提供することを目的としています。

Azureにイメージを展開する方法は様々で、ほとんどのユーザーにとって最も身近な方法はWebポータルを通じて行います。ただし、AzureCLIは、使い慣れた後の迅速で強力なツールです。

Azureの詳細なバックグラウンド、Webポータルを介したCSR1000vの導入方法、およびHAについては、「[Cisco CSR 1000v Deployment Guide for Microsoft Azureおよび関連情報](#)」セクションを参照してください。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Microsoft Azureアカウント
- CSR1000v x 2およびWindows/Linux仮想マシンx 1
- AzureCLI 2.0

使用するコンポーネント

このドキュメントの情報は、Cisco IOS-XE® Denali 16.7.1に基づくものです

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

目標

CSR1000vルータ2台とVM 1台(windows/linux)を導入します。プライベートデータセンター (VM)からインターネット(8.8.8.8)への連続トラフィックをシミュレートします。HAフェールオーバーをシミュレートし、AzureルーティングテーブルがトラフィックをCSR-AからCSR-Bのプライベートインターフェイスに切り替えたことを確認して、HAが成功したことを確認します。

トポロジ

トポロジと設計を完全に理解するためには、設定を開始する前に重要です。これは、今後発生する可能性のある問題のトラブルシューティングに役立ちます。

ユーザの要件に基づいて、HA導入のさまざまなシナリオが考えられます。この例では、次の設定でHA冗長性を設定します。

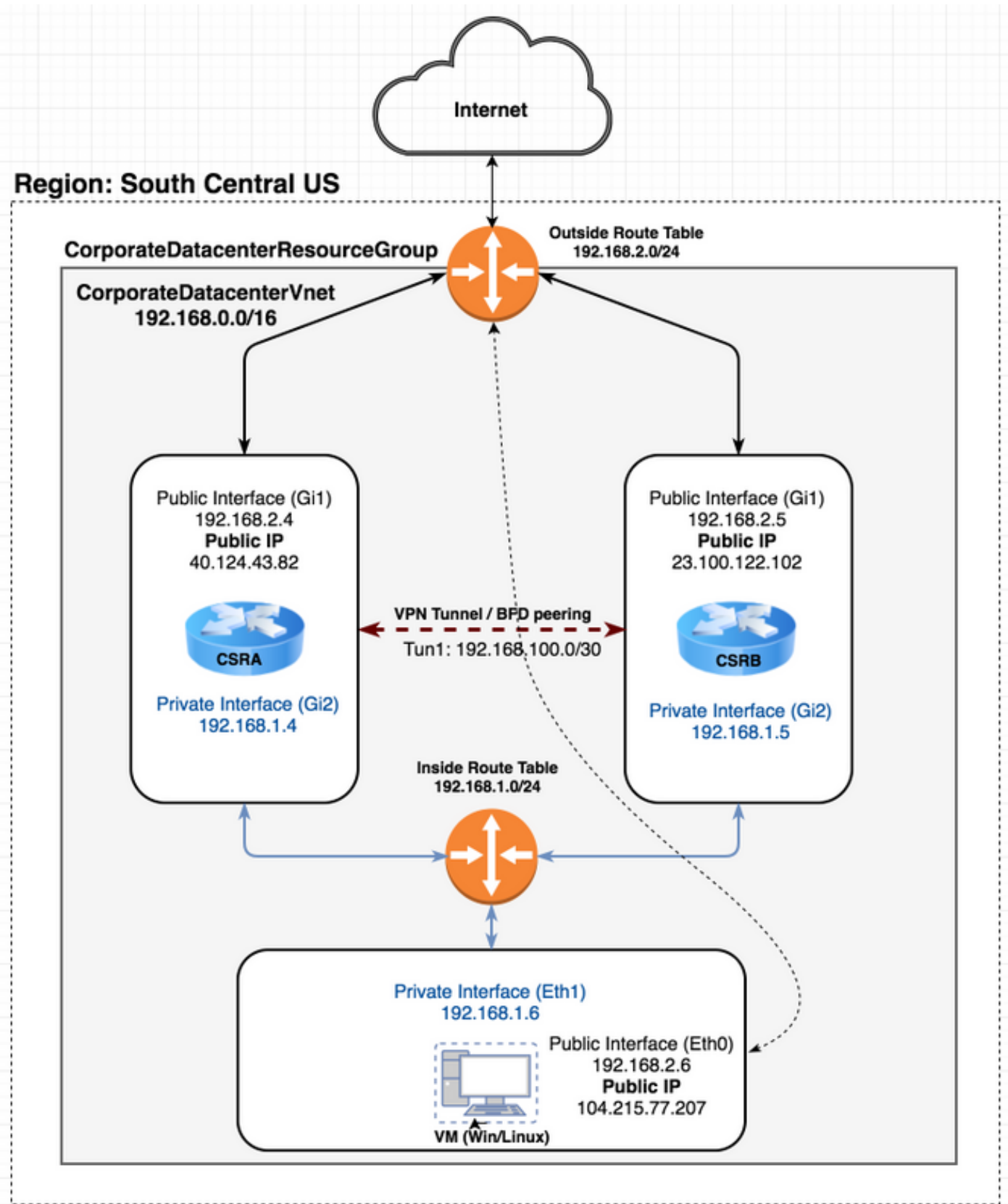
- 1x – 地域 (米国中南部)
- 1x – リソースグループ(CorporateDatacenterResourceGroup)
- 1x - Vnet(CorporateDatacenterVnet)
- 6x – ネットワークインターフェイス (内向き3x、外向き3x)
- 2x : ルートテーブル (InsideRoutetableおよびOutsideRoutetable)
- 2x - CSR1000vルータ(Cisco IOS-XE® Denali 16.7.1)
- 1x - VM(Linux/Windows)

ここでは、パブリックインターフェイスを介したインターネットアクセスがVM上で有効のままであるため、アクセスして設定できます。通常、すべての通常のトラフィックはプライベートルートテーブルを通過する必要があります。VMのパブリックインターフェイスを後で無効にして、誤ってトラフィックがリークされないようにすることができます。

トラフィックシミュレーションは、VMのプライベートインターフェイス→内部ルートテーブル→CSRA → 8.8.8.8からpingを実行して実行されます。フェールオーバーシナリオでは、プライバー

トルートテーブルがCSRのプライベートインターフェイスをポイントするようにルートを切り替えたことを確認します。

ネットワーク図



用語

- リソースグループ – これは、Azureが仮想マシンやvnetなどのすべてのリソースを追跡する方法です。これは通常、すべての品目を管理し、料金を追跡するために使用されます。
- Vnet – 仮想ネットワーク (awsの用語ではVPCと同様)
- ルートテーブル：サブネットのルールが含まれ、特定のトラフィックをIPアドレスに転送したり、VPNエンドポイントのように機能したりできます。

制約事項

- Azure自体では、HAフェールオーバーで約40 ~ 50秒の遅延が発生する場合があります。

コンフィギュレーション

AzureにVMを展開するには、いくつかの方法があります。

1. [Webポータル](#):cisco.comのHAドキュメント
2. Powershell - Azureリソースを管理するためのコマンドラインベースのモデルです。
3. [AzureCLI 2.0 – コマンド・ライン・ベース](#)も含まれます。これはオープンソースであり、Pythonで記述されており、ローカルシステムにインストールする必要があります。このドキュメントを作成するには、AzureCLI 2.0が最新バージョンです。
4. [Azure Cloud Shell - Powershellオプション](#)の代わりにBashシェルオプションを選択し、シェルでAzureCLIを使用します。この方法では、インストールは必要ありません。

```
← → ↻ 🔒 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とAzureCLIは類似していますが、AzureCLIのコマンドはより簡単です。どちらもWindows、MacOS、Linuxで実行できます。比較については、[「Azureに適したツールの選択」](#)および[「Azure CLIおよびPowerShellコマンドを並べて使用する」](#)を参照してください。

この例では、AzureCLIまたはクラウドシェルを使用してすべてのリソースを展開します。AzureCLIは、MacOS、Windows、またはLinuxに少し異なる手順でインストールできます。AzureCLIとAzure Cloud Shellの残りの手順では、構成に違いはありません。

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

注：このテンプレートは、後でCSRでHAを設定するために使用されるすべてのIDと設定を追跡するのに役立ちます。

概要

ステップ1: Azure CLI 2.0をインストールします。

1. Azure CLI 2.0ドキュメントのWindows、MacOS、またはLinuxのインストール手順に[従います](#)。

2. MacOSの場合：

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

3. Azureにログインし、指示に従ってセッションを認証します。

```
$ az login
```

4. ブラウザ認証が完了すると、Azureサブスクリプション情報がJSON形式で返されます。

```
[
  {
    "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. 以降の設定手順を開始する前に、Azure CLIの便利なコマンドとヒントをいくつか紹介します。

- 使用可能なサブコマンドとその機能に関するヘルプを表示するには、**-hオプション**を使用します。

```
$ az account -h
```

- デフォルトでは、すべての出力がJSON形式で返されます。読みやすくするために、**—output table**オプションを使用して表に表を表示することができます。

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

- 使用可能なすべてのVMのリストを取得するか、**—all**オプションを下の他のオプションの1つ

に置き換えてテーブルをフィルタします。

```
$ 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.

- すべての構成コマンドの詳細については、[MicrosoftのAzure CLI 2.0のドキュメント](#)を参照してください。

ステップ2: リソースグループの作成

- リソースグループは、Azureソリューションに関連するリソースを保持するコンテナです。リソースグループに名前を付け、コンテナを展開する場所を選択します。この例では、South Central USを使用しています。

```
$ 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
}
```

- テンプレート (リソースグループの追加)

```
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
```

ステップ3: Vnetを作成します。

- Vnetは、ネットワークが展開されるIPアドレスの領域です。この範囲は、より小さなサブネットに分割され、インターフェイスに割り当てられます。vnetに名前を付け、ステップ2で作成したリソースグループに割り当て、プレフィックス範囲を割り当てます。プレフィックスを指定しない場合、Azureは通常10.0.0.0/16を割り当てます。

```
$ 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": []
  }
}
```

ステップ4：ルートテーブルの作成

1. 内部向けインターフェイスのルートテーブルを作成します。

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

テンプレート (ルートテーブルの追加)

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. Outside側インターフェイスのルートテーブルを作成します。

```

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

```

ステップ5：サブネットを作成します。

1. 手順3でvnetに割り当てたスペースから/24サブネットを作成し、それを内部ルートテーブルに割り当てます。

```

$ 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/virtualNetworks/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. vnetに割り当てたスペースから別の/24サブネットを作成し、外部ルートテーブルに割り当てます。

```

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

```

ステップ6:CSR1000vルータを作成します。

各VMには2つのインターフェイス(内部および外部)が必要です。これは、VMごとに2つのNICを意味します。2つのNICを作成し、パブリックIPを外部NICに関連付けます。

1. パブリックIPアドレスを作成します。

```

$ az network public-ip create --name CSRAPublicIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\"38306703-153b-456b-b2e4-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/CSRA",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "40.124.43.82",

```

```

    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRAPublicIP",
    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "610e1631-331a-4971-8502-xxxxxxxxxxxx",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

2. 外部NICを作成し、パブリックIPアドレスを関連付けます。

```

$ az network nic create --name CSRAOutsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --
public-ip-address CSRAPublicIP
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "plk2sxe5i01lcksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"06fd60de-6547-4992-b506-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"06fd60de-6547-4992-xxxx-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAOutsideInterface/ipConfigurations/ipconfig1",
        "loadBalancerBackendAddressPools": null,
        "loadBalancerInboundNatRules": null,
        "name": "ipconfig1",
        "primary": true,
        "privateIpAddress": "192.168.2.4",
        "privateIpAddressVersion": "IPv4",
        "privateIpAllocationMethod": "Dynamic",
        "provisioningState": "Succeeded",
        "publicIpAddress": {
          "dnsSettings": null,
          "etag": null,
          "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/CSRAPublicIP",
          "idleTimeoutInMinutes": null,
          "ipAddress": null,
          "ipConfiguration": null,

```

```

    "ipTags": null,
    "location": null,
    "name": null,
    "provisioningState": null,
    "publicIpAddressVersion": null,
    "publicIpAllocationMethod": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": null,
    "sku": null,
    "tags": null,
    "type": null,
    "zones": null
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "CSRAOutsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "93413822-e819-4644-ac0d-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

3. 内部NICを作成します。

```

$ az network nic create --name CSRAInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "g1lzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne
tworkInterfaces/CSRAInsideInterface",

```

```

    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\"bebe539f-b5ff-40fa-a122-5c27951afeb1\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/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. Azureで利用可能なCSR1000vイメージをリストします。この例では、**cisco:cisco-csr-1000v:16_7:16.7.120171201**というurn名を使用します。

```

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. CSR1000vをイメージのURN名で展開します。

```

$ 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": ""
}

```

数分後、新しいCSR1000vが起動します。

```

$ 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. CSR1000vにログインし、機能を確認します。

```

$ 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

```

ステップ7:2番目のCSR1000vルータを作成します。

1. パブリックIPアドレスを作成します。

```

$ az network public-ip create --name CSRBPUBLICIP --resource-group
CorporateDatacenterResourceGroup --idle-timeout 30 --allocation-method Static
{
  "publicIp": {
    "dnsSettings": null,
    "etag": "W/\f0f98dac-ea56-4efe-8da6-81a221ac3474\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/pu
blicIPAddresses/CSRB",
    "idleTimeoutInMinutes": 30,
    "ipAddress": "23.100.122.102",
    "ipConfiguration": null,
    "ipTags": [],
    "location": "southcentralus",
    "name": "CSRBPUBLICIP",

```

```

    "provisioningState": "Succeeded",
    "publicIpAddressVersion": "IPv4",
    "publicIpAllocationMethod": "Static",
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": "aa03bc26-22df-4696-bd77-ca29df029d7d",
    "sku": {
      "name": "Basic",
      "tier": "Regional"
    },
    "tags": null,
    "type": "Microsoft.Network/publicIPAddresses",
    "zones": null
  }
}

```

2. 外部NICを作成し、パブリックIPアドレスを関連付けます。

```

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

```

```

    "publicIpAllocationMethod": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": null,
    "sku": null,
    "tags": null,
    "type": null,
    "zones": null
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
}
],
"location": "southcentralus",
"macAddress": null,
"name": "CSRBOutsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "c3f05156-ad07-4abd-a006-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

3. 内部NICを作成します。

```

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

```



```

xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/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-
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": "CSRBIInsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "085c88fc-9e78-49be-a5a7-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

4. 2つ目のCSR1000vを同じイメージcisco:cisco-csr-1000v:16_7:16.7.120171201で導入します

```

o
$ 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
"Cisco1234567" --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/virtualMachines/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": ""
}

```

ステップ8 : ステップ6と同じ手順でホストVMを作成します。 この例では、UbuntuLTSを使用しています。

1. パブリックIPアドレスを作成します。

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

2. 外部NICを作成し、外部サブネットとパブリックIPアドレスを関連付けます。 サブネットがNICに関連付けられると、NICにIPアドレスが自動的に割り当てられます。この例では、OutsideSubnetは192.168.2.0/24であり、NICに自動的に割り当てられるIPアドレスは192.168.2.6です。

```
$ az network nic create --name VMHostOutsideInterface --resource-group CorporateDatacenterResourceGroup --subnet OutsideSubnet --vnet CorporateDatacenterVnet --public-ip-address VMHostPublicIP {
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "gzkplk2sxe5i011ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,
    "etag": "W/\\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/VMHostOutsideInterface",
    "ipConfigurations": [
      {
        "applicationGatewayBackendAddressPools": null,
        "applicationSecurityGroups": null,
        "etag": "W/\\"2c70c97b-6470-42c8-b481-xxxxxxxxxxxx\"",
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
```

```

xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/networkInterfaces/VMHostOutsideInterface/ipConfigurations/ipconfig1",
  "loadBalancerBackendAddressPools": null,
  "loadBalancerInboundNatRules": null,
  "name": "ipconfig1",
  "primary": true,
  "privateIpAddress": "192.168.2.6",
  "privateIpAddressVersion": "IPv4",
  "privateIpAllocationMethod": "Dynamic",
  "provisioningState": "Succeeded",
  "publicIpAddress": {
    "dnsSettings": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/publicIPAddresses/VMHostPublicIP",
    "idleTimeoutInMinutes": null,
    "ipAddress": null,
    "ipConfiguration": null,
    "ipTags": null,
    "location": null,
    "name": null,
    "provisioningState": null,
    "publicIpAddressVersion": null,
    "publicIpAllocationMethod": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceGuid": null,
    "sku": null,
    "tags": null,
    "type": null,
    "zones": null
  },
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "subnet": {
    "addressPrefix": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/virtualNetworks/CorporateDatacenterVnet/subnets/OutsideSubnet",
    "ipConfigurations": null,
    "name": null,
    "networkSecurityGroup": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "resourceNavigationLinks": null,
    "routeTable": null,
    "serviceEndpoints": null
  }
},
"location": "southcentralus",
"macAddress": null,
"name": "VMHostOutsideInterface",
"networkSecurityGroup": null,
"primary": null,
"provisioningState": "Succeeded",
"resourceGroup": "CorporateDatacenterResourceGroup",
"resourceGuid": "89588a04-6ba6-467d-a86f-xxxxxxxxxxxx",
"tags": null,
"type": "Microsoft.Network/networkInterfaces",
"virtualMachine": null
}
}

```

3. 内部NICを作成します。

```
$ 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. Ubuntu VMを導入します。この例では、UbuntuLTSを使用しています。

```

az vm image list --output table
You are viewing an offline list of images, use --all to retrieve an up-to-date list
Offer          Publisher          Sku          Urn
UrnAlias       Version
-----
CentOS         OpenLogic          7.3          OpenLogic:CentOS:7.3:latest
CentOS         latest
CoreOS         CoreOS             Stable       CoreOS:CoreOS:Stable:latest
CoreOS         latest
Debian         credativ           8            credativ:Debian:8:latest
Debian         latest
openSUSE-Leap SUSE               42.3         SUSE:openSUSE-Leap:42.3:latest
openSUSE-Leap latest
RHEL           RedHat             7.3          RedHat:RHEL:7.3:latest
RHEL           latest
SLES           SUSE               12-SP2       SUSE:SLES:12-SP2:latest
SLES           latest
UbuntuServer  Canonical           16.04-LTS    Canonical:UbuntuServer:16.04-
LTS:latest    UbuntuLTS     latest
WindowsServer MicrosoftWindowsServer 2016-Datacenter
MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest Win2016Datacenter latest
WindowsServer MicrosoftWindowsServer 2012-R2-Datacenter
MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest Win2012R2Datacenter latest
WindowsServer MicrosoftWindowsServer 2012-Datacenter
MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest Win2012Datacenter latest
WindowsServer MicrosoftWindowsServer 2008-R2-SP1
MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest Win2008R2SP1 latest

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

```

ステップ9 : ルーティングテーブルとVMにルートを追加します。

1. ネクストホップのIPアドレスを192.168.1.4に設定して、CSR Aを介してトラフィックをルーティングする内部サブネットのデフォルトルートを追加します。これは InsideRouteTableで行われます。

```

$ 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. OutsideRouteTableのインターネットに到達するためのネットワーク内のトラフィックのルートを追加します。

```
$ 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. Ubuntu VMにログインし、内部インターフェイスを通過するトラフィックを強制的に8.8.8.8に転送するルートを追加します。 Azureルートテーブルは、サブネット内の最初のIPをゲートウェイとして自動的に使用します。 内部インターフェイス(eth1)のサブネットは192.168.1.0/24です。つまり、192.168.1.1がホストVMのデフォルトゲートウェイアドレスです。

```
$ 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

テンプレート (cidr ipの追加)

```

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

```

注：インターネット(8.8.8.8)をpingするには、手順10でCSR1000vルータにNATを設定する必要があります。**注：**手順10 ~ 14では、HA用CSR1000vルータの設定について説明します。[Cisco CSR 1000v Deployment Guide for Microsoft Azure](#)の手順は、「[トラストプールの設定](#)」以降で説明されています。詳細については、ガイドをご覧ください。

手順10:CSR1000vルータを設定します。

1. 両方のCSR1000vルータにトラストプールを設定する

```

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. Cisco CSR 1000vルータ間にipsecトンネルを設定し、ピア障害検出のためにルータ間のトンネルでBi-directional Forwarding Detection(BFD)とルーティングプロトコル (EIGRPまたはBGP) を有効にします。 **注：**設定のトンネルの宛先アドレスは、ピアCSRのパブリックIPアドレスです。CSRAの設定

```

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
CSRBの設定
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. 両方のCSR1000vルータで、NATとルーティングに同じ設定が使用されます。これは、内部インターフェイスを介したVMインターネットの到達可能性を目的としています。

```

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. ルートテーブルのアクセスコントロール(IAM)を追加します。 AzureCLIで、フェールオーバー中にアプリケーション (CSRAおよびCSRB) がAzureのInsideRouteTableを変更できるようにします。 次のセクションで、—scopesオプションとして使用するInsideRouteTableのIDに注意してください。

```

$ 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",
  ...

```


テンプレート (サブスクリプションIDの追加)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id
tenant-id
resource-group CorporateDatacenterResourceGroup
```

5. InsideRouteTableのIAMロールを作成します。—scopesオプションは、前の出力のidフィールドから取得されます。app-id、password(app-key)、およびテナントIDをメモします。

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

テンプレート (app-key、app-id、およびtenant-idの追加)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup
```

6. 両方のルータでクラウドの冗長性を設定します。両方のルータの設定の唯一の違いは、bfdピアとデフォルトゲートウェイです。CSRAの設定

```
redundancy
cloud provider azure 100
bfd peer 192.168.101.2
route-table InsideRoutetable
default-gateway ip 192.168.1.4
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
resource-group CorporateDatacenterResourceGroup
```

CSRBの設定

```
redundancy
cloud provider azure 100
bfd peer 192.168.101.1
route-table InsideRoutetable
default-gateway ip 192.168.1.5
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
subscription-id 09e13fd4-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
app-id 576dd4f1-c08d-46b9-cccc-xxxxxxxxxxxxx
tenant-id ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx
```

ハイアベイラビリティの確認

1. BFDとクラウドの設定を確認します。

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

2. VMから宛先にpingおよびtracerouteを実行します。pingが内部eth1インターフェイスを経由していることを確認します。

```
$ 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は、VMから8.8.8.8へのパスがCSRAの内部インターフェイスを経由していることを示します。

```
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. CSRAのtunnel 1インターフェイスをシャットダウンして、フェールオーバーをシミュレートします。

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

5. トラフィックがCSRのプライベートインターフェイスを通過していることを確認します。

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

注：Azureクラウドでは、フェールオーバー時に遅延が発生する可能性があります。遅延は1分以下にする必要があります。

トラブルシューティング

- デバッグを有効にして、HAフェールオーバー中のメッセージを確認します。

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

- 認証と資格情報のエラーは、CSR1000vがAzureルートテーブルにAPI呼び出しを行うことができるようにする無効なアクセスコントロールに起因します。手順10で適切なIDが設定されていることを再確認します。

```
*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"}
```

関連情報

- [Azure CLI 2.0](#)
- [Cisco CSR 1000v Deployment Guide for Microsoft Azure](#)
- [Azure CLIコマンドとPowerShellコマンドを並べて使用する場合に最適なツールを選択する](#)