

Guida alla distribuzione della ridondanza HA CSR1000v in Microsoft Azure con Azure CLI 2.0

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Obiettivo](#)

[Topologia](#)

[Esempio di rete](#)

[Terminologia](#)

[Restrizioni](#)

[Configurazione](#)

[Panoramica](#)

[Passaggio 1. Installare AzureCLI 2.0.](#)

[Passaggio 2. Creare un gruppo di risorse.](#)

[Passaggio 3. Creare una rete virtuale.](#)

[Passaggio 4. Creazione delle tabelle dei cicli di lavorazione.](#)

[Passaggio 5. Creazione di subnet.](#)

[Passaggio 6. Creare un router CSR1000v.](#)

—

[Passaggio 7. Creare il secondo router CSR1000v.](#)

[Passaggio 8. Creare una VM host con la stessa procedura descritta nel passaggio 6. In questo esempio viene utilizzato UbuntuLTS.](#)

[Passaggio 9. Aggiungere route alle tabelle di routing e alle macchine virtuali.](#)

[Passaggio 10. Configurare i router CSR1000v.](#)

[Verifica dell'elevata disponibilità](#)

[Risoluzione dei problemi](#)

[Informazioni correlate](#)

Introduzione

In questo documento viene fornita una guida dettagliata alla configurazione su come distribuire i router CSR1000v per l'alta disponibilità nel cloud di Microsoft Azure con Azure CLI 2.0. L'obiettivo è quello di fornire agli utenti una conoscenza pratica dell'HA e la capacità di installare un banco di prova completamente funzionale.

Esistono vari metodi per distribuire immagini in Azure e il metodo più comune per la maggior parte degli utenti è tramite il portale Web. Tuttavia, AzureCLI è uno strumento rapido e potente una volta acquisita dimestichezza con il programma.

Per informazioni più dettagliate su Azure, su come distribuire un CSR1000v tramite il portale Web e su HA, vedere la sezione [Cisco CSR 1000v Deployment Guide for Microsoft Azure and Related](#)

Information.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Un account di Microsoft Azure
- 2 CSR1000v e 1 macchina virtuale Windows/Linux
- Azure CLI 2.0

Componenti usati

Il riferimento delle informazioni contenute in questo documento è Cisco IOS-XE® Denali 16.7.1

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Obiettivo

Distribuire 2 router CSR1000v e 1 VM (Windows/Linux). Simulare il traffico continuo dal centro dati privato (VM) a Internet (8.8.8.8). Simulare un failover HA e osservare che ha avuto esito positivo confermando che la tabella di routing di Azure ha modificato il traffico dall'interfaccia privata di CSR-A a CSR-B.

Topologia

Per comprendere appieno la topologia e la progettazione è importante prima di iniziare la configurazione. In questo modo è possibile risolvere eventuali problemi in un secondo momento.

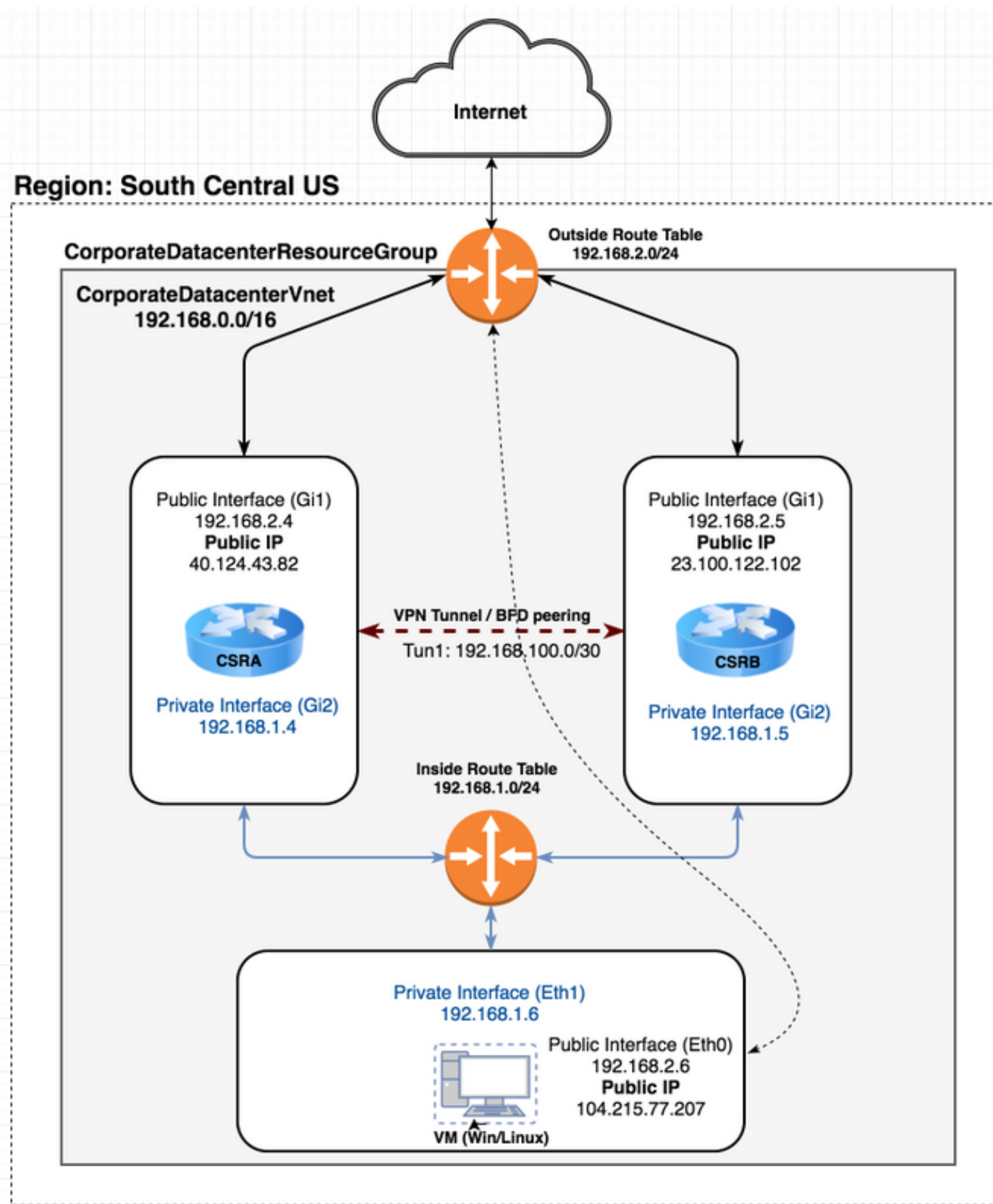
Possono esistere diversi scenari di installazioni HA in base ai requisiti dell'utente. Per questo esempio, configurare la ridondanza HA con queste impostazioni:

- 1x - Regione (Stati Uniti centro-meridionali)
- 1x - Gruppo di risorse (CorporateDatacenterResourceGroup)
- 1x - Rete virtuale (CorporateDatacenterVnet)
- 6 interfacce di rete (3 interfacce interne e 3 interfacce esterne)
- 2 tabelle di routing (InsideRoutetable e OutsideRoutetable)
- 2 router - CSR1000v (Cisco IOS-XE® Denali 16.7.1)
- 1x - VM (Linux/Windows)

Per il momento, l'accesso a Internet tramite l'interfaccia pubblica rimane abilitato sulla VM in modo che sia possibile accedervi e configurarlo. In genere, tutto il traffico normale deve passare attraverso la tabella di route privata. L'interfaccia pubblica sulla VM può essere successivamente disabilitata in modo che non si verifichi alcuna perdita accidentale di traffico.

La simulazione del traffico viene eseguita eseguendo il ping dall'interfaccia privata della VM → all'interno della tabella di route → CSRA → 8.8.8.8. In uno scenario di failover, osservare che la tabella di route privata ha cambiato la route in modo da puntare all'interfaccia privata di CSRB.

Esempio di rete



Terminologia

- Gruppo di risorse - Consente ad Azure di tenere traccia di tutte le risorse, ad esempio macchine virtuali e reti virtuali. In genere viene utilizzato per gestire tutti gli articoli e per tenere traccia dei costi.
- Vnet - Una rete virtuale (simile alla terminologia VPC in aws).
- Tabella di routing: contiene le regole per una subnet e può inoltrare il traffico specifico a un indirizzo IP o funzionare come un endpoint VPN.

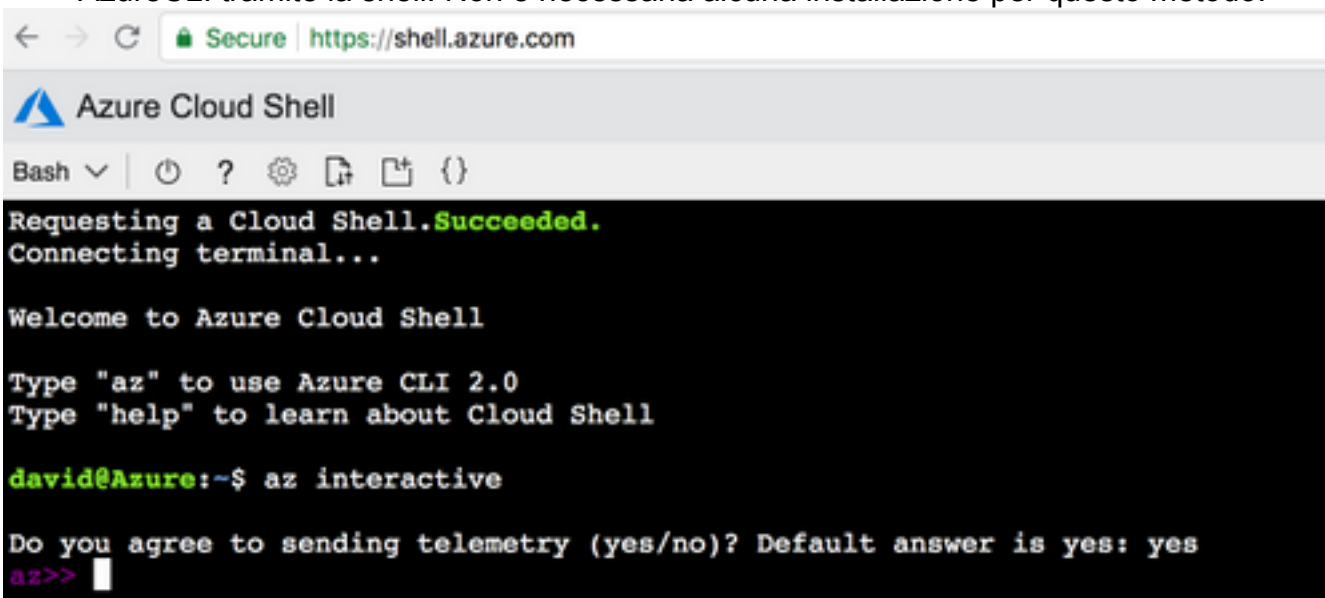
Restrizioni

- Azure stesso può introdurre un ritardo di circa 40-50 secondi in un failover HA.

Configurazione

Esistono alcuni metodi per distribuire le macchine virtuali in Azure:

1. [Portale Web](#) - Documentazione sulla disponibilità su cisco.com
2. Powershell - Modello basato sulla riga di comando per la gestione delle risorse di Azure.
3. [AzureCLI 2.0](#) - Basato anche sulla riga di comando. È open source e scritto in python e deve essere installato sul sistema locale. Per poter scrivere questo documento, AzureCLI 2.0 è la versione più recente.
4. [Azure Cloud Shell](#): scegliere l'opzione **Bash Shell** anziché l'opzione **Powershell** per usare AzureCLI tramite la shell. Non è necessaria alcuna installazione per questo metodo.



```
← → ↻ 🔒 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 e AzureCLI sono simili, ma i comandi per AzureCLI sono più semplici. Entrambi possono essere eseguiti su Windows, MacOS, Linux. Per un confronto, fare riferimento a [Scelta degli strumenti corretti per Azure e comandi CLI e PowerShell affiancati](#) per [Azure](#).

Per questo esempio, distribuire tutte le risorse con AzureCLI o Cloud Shell. AzureCLI può essere installato in MacOS, Windows o Linux con passaggi leggermente diversi. Non esistono differenze

nella configurazione nel resto della procedura tra AzureCLI e Azure Cloud Shell.

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

Nota: Questo modello è utile per tenere traccia di tutti gli ID e della configurazione utilizzati successivamente per configurare HA sui CSR.

Panoramica

Passaggio 1. Installare AzureCLI 2.0.

1. Seguire la procedura di installazione per Windows, MacOS o Linux nella documentazione di [AzureCLI 2.0](#).
2. Per MacOS:
`$ brew update && brew install azure-cli`
3. Accedere ad Azure e seguire le istruzioni per autenticare la sessione.
`$ az login`
4. Al termine dell'autenticazione del browser, le informazioni sulla sottoscrizione di Azure vengono restituite in formato JSON:

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

5. Prima di iniziare a eseguire gli altri passaggi della configurazione, di seguito sono riportati alcuni comandi e suggerimenti utili su AzureCLI.
- Per informazioni sui sottocomandi disponibili e sulle relative funzioni, usare l'opzione **-h**.

```
$ az account -h
```

- Per impostazione predefinita, tutti gli output vengono restituiti in formato JSON. Per una migliore leggibilità, potete utilizzare l'opzione **—tabella di output** per visualizzare in una tabella.

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

- Ottenere un elenco di tutte le macchine virtuali disponibili o sostituire l'opzione **—all** con una delle altre opzioni seguenti per filtrare la tabella.

```
$ 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.

- Per informazioni dettagliate su tutti i comandi di configurazione, consultare la documentazione di [Azure CLI 2.0](#) di Microsoft.

Passaggio 2. Creare un gruppo di risorse.

- Un gruppo di risorse è un contenitore che contiene le risorse correlate per una soluzione Azure. Assegnare un nome al gruppo di risorse e scegliere il percorso in cui distribuire il contenitore. In questo esempio vengono utilizzati gli Stati Uniti centro-meridionali.

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

- Modello (aggiunta del gruppo di risorse)

```
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
```

Passaggio 3. Creare una rete virtuale.

- Una rete virtuale è uno spazio di indirizzi IP in cui viene installata la rete. L'intervallo viene quindi suddiviso in subnet più piccole e assegnato alle interfacce. Assegnare un nome alla rete virtuale, assegnarla al gruppo di risorse creato nel passaggio 2 e allocare un intervallo di prefissi. Se non si specifica un prefisso, viene in genere assegnato il valore 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": []
  }
}
```

Passaggio 4. Creazione delle tabelle dei cicli di lavorazione.

1. Create una tabella di stesura per le interfacce rivolte all'interno.

```
$ 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"
```

```
}
```

Modello (aggiunta della tabella di route)

```
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. Create una tabella di stesura per le interfacce rivolte all'esterno.

```
$ 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"
}
```

Passaggio 5. Creazione di subnet.

1. Creare una subnet /24 dallo spazio assegnato alla rete virtuale al passaggio 3, quindi assegnarla alla tabella di route interna.

```
$ az network vnet subnet create --address-prefix 192.168.1.0/24 --name InsideSubnet --
resource-group CorporateDatacenterResourceGroup --vnet-name CorporateDatacenterVnet --
route-table InsideRoutetable
{
  "addressPrefix": "192.168.1.0/24",
  "etag": "W/\a0dbd178-3a45-48fb-xxxx-xxxxxxxxxxxx\"",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
  "ipConfigurations": null,
  "name": "InsideSubnet",
  "networkSecurityGroup": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceNavigationLinks": null,
  "routeTable": {
    "disableBgpRoutePropagation": null,
    "etag": null,
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ro
uteTables/InsideRoutetable",
    "location": null,
```



```

    "name": null,
    "provisioningState": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "routes": null,
    "subnets": null,
    "tags": null,
    "type": null
  },
  "serviceEndpoints": null
}

```

2. Creare un'altra subnet /24 dallo spazio assegnato alla rete virtuale e assegnarla alla tabella di route esterna.

```

$ 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
}

```

Passaggio 6. Creare un router CSR1000v.

Ogni VM deve avere 2 interfacce (interne ed esterne), ovvero 2 NIC per VM. Creare le due schede NIC e associare un indirizzo IP pubblico alla scheda NIC esterna.

1. Creare l'indirizzo IP pubblico.

```

$ 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. Creare la scheda NIC esterna e associarvi l'indirizzo IP pubblico.

```

$ 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. Creare la scheda NIC interna.

```

$ az network nic create --name CSRAInsideInterface --resource-group
CorporateDatacenterResourceGroup --subnet InsideSubnet --vnet CorporateDatacenterVnet
{
  "NewNIC": {
    "dnsSettings": {
      "appliedDnsServers": [],
      "dnsServers": [],
      "internalDnsNameLabel": null,
      "internalDomainNameSuffix": "g1lzkplk2sxe5i01l1ccksytfab.jx.internal.cloudapp.net",
      "internalFqdn": null
    },
    "enableAcceleratedNetworking": false,
    "enableIpForwarding": false,

```

```

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

```

4. Elenca le immagini CSR1000v disponibili in Azure. In questo esempio viene utilizzato il nome urn di **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. Distribuire CSR1000v con il nome urn dell'immagine.

```
$ 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/virtualMachines/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": ""
}
```

Dopo alcuni minuti, il nuovo CSR1000v si avvia.

```
$ 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. Accedere a CSR1000v e verificare la funzionalità.

```
$ 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
```

Passaggio 7. Creare il secondo router CSR1000v.

1. Creare l'indirizzo IP pubblico.

```
$ 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/CSRBPUBLICIP",
    "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. Creare la scheda NIC esterna e associarvi l'indirizzo IP pubblico.

```

$ 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. Creare la scheda NIC interna.

```

$ 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/ne
tworkInterfaces/CSRBInsideInterface/ipConfigurations/ipconfig1",
    "loadBalancerBackendAddressPools": null,
    "loadBalancerInboundNatRules": null,
    "name": "ipconfig1",
    "primary": true,
    "privateIpAddress": "192.168.1.5",
    "privateIpAddressVersion": "IPv4",
    "privateIpAllocationMethod": "Dynamic",
    "provisioningState": "Succeeded",
    "publicIpAddress": null,
    "resourceGroup": "CorporateDatacenterResourceGroup",
    "subnet": {
      "addressPrefix": null,
      "etag": null,
      "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/vi
rtualNetworks/CorporateDatacenterVnet/subnets/InsideSubnet",
      "ipConfigurations": null,
      "name": null,
      "networkSecurityGroup": null,
      "provisioningState": null,
      "resourceGroup": "CorporateDatacenterResourceGroup",
      "resourceNavigationLinks": null,
      "routeTable": null,
      "serviceEndpoints": null
    }
  }
},
  "location": "southcentralus",
  "macAddress": null,
  "name": "CSRBInsideInterface",
  "networkSecurityGroup": null,
  "primary": null,
  "provisioningState": "Succeeded",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "resourceGuid": "085c88fc-9e78-49be-a5a7-xxxxxxxxxxxx",
  "tags": null,
  "type": "Microsoft.Network/networkInterfaces",
  "virtualMachine": null
}
}

```

4. Distribuire il secondo CSR1000v con la stessa immagine **cisco:cisco-csr-1000v:16_7:16.7.120171201**.

```

$ az vm create --resource-group CorporateDatacenterResourceGroup --name CSRB --location
southcentralus --image cisco:cisco-csr-1000v:16_7:16.7.120171201 --nics
CSRBOutsideInterface CSRBInsideInterface --admin-username cisco --admin-password
"Cisc0l234567" --authentication-type password
{
  "fqdns": "",
  "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-
xxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Compute/vi
rtualMachines/CSRB",
  "location": "southcentralus",
  "macAddress": "00-0D-3A-5D-8C-51,00-0D-3A-5D-85-2A",
  "powerState": "VM running",
  "privateIpAddress": "192.168.2.5,192.168.1.5",
  "publicIpAddress": "23.100.122.102",
  "resourceGroup": "CorporateDatacenterResourceGroup",
  "zones": ""
}

```

```
}
```

Passaggio 8. Creare una VM host con la stessa procedura descritta nel passaggio 6. In questo esempio viene utilizzato UbuntuLTS.

1. Creare l'indirizzo IP pubblico.

```
$ 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. Creare la scheda NIC esterna e associarvi la subnet esterna e l'indirizzo IP pubblico.

Quando le subnet sono associate alle schede NIC, alla scheda NIC viene assegnato automaticamente un indirizzo IP. Nell'esempio, la subnet esterna è 192.168.2.0/24 e l'indirizzo IP assegnato automaticamente alla scheda NIC è 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-xxxxxxxxxxxxx\\"",
    "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/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. Creare la scheda NIC interna.

```
$ az network nic create --name VMHostInsideInterface --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/\\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxxx\\"",  
    "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-  
xxxxxxxxxxxxx/resourceGroups/CorporateDatacenterResourceGroup/providers/Microsoft.Network/ne  
tworkInterfaces/VMHostInsideInterface",  
    "ipConfigurations": [  
      {  
        "applicationGatewayBackendAddressPools": null,  
        "applicationSecurityGroups": null,  
        "etag": "W/\\"dda7eacf-4670-40c2-999c-xxxxxxxxxxxxx\\"",  
        "id": "/subscriptions/09e13fd4-def2-46aa-xxxx-  
xxxxxxxxxxxxx/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-  
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": "VMHostInsideInterface",  
    "networkSecurityGroup": null,  
    "primary": null,  
    "provisioningState": "Succeeded",  
    "resourceGroup": "CorporateDatacenterResourceGroup",
```

```

    "resourceGuid": "8ef12cdd-cc31-432e-99cf-xxxxxxxxxxxx",
    "tags": null,
    "type": "Microsoft.Network/networkInterfaces",
    "virtualMachine": null
  }
}

```

4. Distribuire la VM Ubuntu. In questo esempio viene utilizzato UbuntuLTS.

```

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

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

```

Passaggio 9. Aggiungere route alle tabelle di routing e alle macchine virtuali.

1. Aggiungere un percorso predefinito per la subnet interna per instradare il traffico attraverso il CSR A impostando l'indirizzo IP dell'hop successivo come 192.168.1.4. Questa operazione viene eseguita nella tabella InsideRouteTable.

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

2. Aggiungere una route per il traffico in rete per raggiungere Internet in OutsideRouteTable.

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

3. Accedere alla VM Ubuntu e aggiungere una route per forzare il traffico attraverso l'interfaccia interna alla versione 8.8.8.8. La tabella di route di Azure utilizza automaticamente il primo IP in una subnet come gateway. La subnet dell'interfaccia interna (eth1) è 192.168.1.0/24, quindi 192.168.1.1 è l'indirizzo gw predefinito per la VM host.

```
$ ifconfig
eth0 Link encap:Ethernet HWaddr 00:0d:3a:5d:b7:cb
inet addr:192.168.2.6 Bcast:192.168.2.255 Mask:255.255.255.0
inet6 addr: fe80::20d:3aff:fe5d:b7cb/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:3986 errors:0 dropped:0 overruns:0 frame:0
TX packets:2881 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3475393 (3.4 MB) TX bytes:592740 (592.7 KB)

eth1 Link encap:Ethernet HWaddr 00:0d:3a:5d:b8:9b
inet addr:192.168.1.6 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::20d:3aff:fe5d:b89b/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2 errors:0 dropped:0 overruns:0 frame:0
TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:762 (762.0 B) TX bytes:1620 (1.6 KB)
```

```
$ sudo route add -host 8.8.8.8 gw 192.168.1.1 dev eth1
$ route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
```

0.0.0.0	192.168.2.1	0.0.0.0	UG	0	0	0 eth0
8.8.8.8	192.168.1.1	255.255.255.255	UGH	0	0	0 eth1
168.63.129.16	192.168.2.1	255.255.255.255	UGH	0	0	0 eth0
169.254.169.254	192.168.2.1	255.255.255.255	UGH	0	0	0 eth0
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0 eth1
192.168.2.0	0.0.0.0	255.255.255.0	U	0	0	0 eth0

Modello (aggiunta di 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

```

Nota: NAT deve essere configurato sui router CSR1000v nel passaggio 10 per eseguire il ping su Internet (8.8.8.8). **Nota:** I passaggi da 10 a 14 riguardano la configurazione dei router CSR1000v per HA. I passaggi abbreviati della [Guida alla distribuzione di Cisco CSR 1000v per Microsoft Azure](#) vengono forniti a partire da Configura un pool di fiducia. Per ulteriori informazioni, consultare la Guida.

Passaggio 10. Configurare i router CSR1000v.

1. Configurazione di un pool di fiducia su entrambi i router 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. Configurare un tunnel IPsec tra i router Cisco CSR 1000v e abilitare il rilevamento dell'inoltro bidirezionale (BFD) e un protocollo di routing (EIGRP o BGP) sul tunnel tra i router per il rilevamento degli errori dei peer. **Nota:** L'indirizzo di destinazione del tunnel nella configurazione è l'indirizzo IP pubblico del CSR peer.

```

Configurazione 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
Configurazione 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 stessa configurazione per NAT e routing viene utilizzata su entrambi i router CSR1000v. Questo è per la raggiungibilità di Internet VM attraverso l'interfaccia interna.

```

interface GigabitEthernet1
  ip nat outside
!
interface GigabitEthernet2
  ip nat inside
!
ip nat inside source list 10 interface GigabitEthernet1 overload
access-list 10 permit 192.168.1.0 0.0.0.255 /* Translating the inside subnet of the VM */
!
ip route 0.0.0.0 0.0.0.0 192.168.2.1
ip route 192.168.1.0 255.255.255.0 GigabitEthernet2 192.168.1.1

```

4. aggiungere controlli di accesso (IAM) per una tabella di route. In AzureCLI, consentire all'applicazione (CSRA e CSRB) di modificare InsideRouteTable in Azure durante un failover. Notare l'ID di InsideRouteTable da utilizzare come opzione **—scopes** nella sezione successiva.

```

$ 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",

```


...

Modello (aggiunta dell'ID sottoscrizione)

```
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. Creare il ruolo IAM per InsideRouteTable. L'opzione **—scopes** viene ricavata dal campo **id** dell'output precedente. Prendere nota dell'**app-id**, della **password** (che è l'**app-key**) e dell'**ID tenant**.

```
$ 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-xxxxxxxxxxxxx",
"displayname": "InsideRouteTableIAM",
"name": "http://InsideRouteTableIAM",
"password": "aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx",
"tenant": "ae49849c-2622-xxxx-xxxx-xxxxxxxxxxxxx"
}
```

Modello (aggiunta di app-key, app-id e tenant-id)

```
redundancy
cloud provider azure 100
bfd peer
route-table InsideRoutetable
default-gateway ip
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-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. Configurare la ridondanza del cloud su entrambi i router. L'unica differenza tra la configurazione su entrambi i router sono i peer bfd e il gateway predefinito. Configurazione **CSRA**

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

Configurazione CSRB

```
redundancy
cloud provider azure 100
bfd peer 192.168.101.1
route-table InsideRoutetable
default-gateway ip 192.168.1.5
cidr ip 8.8.8.8/32
app-key aaafc573-e84e-42ac-b4e3-xxxxxxxxxxxxx
```

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

Verifica dell'elevata disponibilità

1. Verificare le configurazioni BFD e 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-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. Eseguire il ping e il traceroute tra la macchina virtuale e la destinazione. Accertarsi che il ping sia eseguito tramite l'interfaccia eth1 interna.

```
$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.6 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=54 time=10.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=54 time=10.6 ms
```

```
$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1 192.168.1.4 (192.168.1.4) 1.516 ms 1.503 ms 1.479 ms
```

```
cisco@VmHost:~$ ping -I eth1 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.1.6 eth1: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=10.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=10.2 ms
```

3. Traceroute mostra che il percorso dalla macchina virtuale alla versione 8.8.8.8 passa attraverso l'interfaccia interna di 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. Arrestare l'interfaccia tunnel 1 di CSRA per simulare un failover.

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

5. Il traffico passa ora attraverso l'interfaccia privata di CSRB.

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

Nota: Il cloud di Azure potrebbe introdurre un ritardo durante il failover. Il ritardo non deve superare 1 minuto.

Risoluzione dei problemi

- Abilitare i debug per osservare i messaggi durante il failover HA.

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

- Gli errori di autenticazione e credenziali sono dovuti a controlli di accesso non validi che consentono a CSR1000v di effettuare chiamate API alla tabella route di Azure. Verificare che gli ID corretti siano configurati nel passaggio 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"}
```

Informazioni correlate

- [Azure CLI 2.0](#)
- [Guida alla distribuzione di Cisco CSR 1000v per Microsoft Azure](#)
- [Scelta degli strumenti appropriati per Azure e comandi affiancati CLI di Azure e PowerShell](#)