CSR1000v HA版本2配置指南（在Microsoft Azure）

目录

简介

本文档用作Azure中高可用性版本2(HAv2)的补充配置指南。有关完整详细信息，请参阅《Cisco CSR 1000v Microsoft Azure部署指南》。Cisco IOS-XE® Denali 16.9.1s首先支持HAv2。

在HAv2中，HA的实施已从Cisco IOS XE代码中移出，并在guestshell容器中运行。有关guestshell的详细信息，请参阅《可编程性配置指南》*中的"访*客外壳"部分。在HAv2中，冗余节点的配置在guestshell中使用一组Python脚本执行。

先决条件

要求

Cisco 建议您了解以下主题：

- Microsoft Azure帐户。
- 2x CSR1000v路由器，带2x千兆接口。面向外部的接口必须位于GigabitEthernet1(eth0)上。
- 至少Cisco IOS-XE® Denali 16.9.1s。

使用的组件

本文档中的信息基于从Azure Marketplace本地部署的Cisco IOS-XE® Denali 16.9.1s。

在Azure中通过本文档中的步骤部署的资源可能会产生成本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原

始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

限制

- 必须在eth0上配置面向外部公共的接口，该接口与GigabitEthernet1对应。只能通过虚拟机上的主接口访问Azure元数据服务器。
- 如果HAv1 IOS配置存在，则必须在HAv2配置之前在guestshell中将其删除。HAv1配置由冗余和**云提**供**商命**令组成。

配置

步骤1.为应用托管配置IOX。

1. 启用IOX应用托管。为VirtualPortGroup0分配私有IP地址。使用面向公共的接口NAT VirtualPortGroup0，以允许访客外壳访问互联网。在本例中，GigabitEthernet1的IP为10.3.0.4。

```
vrf definition GS
!
iox
app-hosting appid guestshell
app-vnic gateway1 virtualportgroup 0 guest-interface 0
guest-ipaddress 192.168.35.102 netmask 255.255.255.0
app-default-gateway 192.168.35.101 guest-interface 0
name-server0 8.8.8.8
!
interface VirtualPortGroup0
vrf forwarding GS
ip address 192.168.35.101 255.255.255.0
ip nat inside
!
interface GigabitEthernet1
ip nat outside
!
ip access-list standard GS_NAT_ACL
permit 192.168.35.0 0.0.0.255
!
ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 vrf GS overload
!
! The static route points to the gig1 private ip address gateway
ip route vrf GS 0.0.0.0 0.0.0.0 GigabitEthernet1 10.1.0.1 global
```

注意：从Azure Marketplace部署的新实例可能已预配置了iox。

步骤2.在Guestshell中安装Python软件包。

1. 启用guestshell和登录。
   ```
   csr-1#guestshell enable
   csr-1#guestshell
   ```
2. Ping www.google.com以验证guestshell是否可以访问Internet。如果无法访问，请检查应用托管IOS配置中的name-server配置，或在guestshell中的resolv.conf中添加服务器。
   ```
   [guestshell@guestshell ~]$ ping www.google.com
   PING www.google.com (172.217.14.228) 56(84) bytes of data.
   ```

```
64 bytes from sea30s02-in-f4.1e100.net (172.217.14.228): icmp_seq=1 ttl=51 time=4.89 ms
64 bytes from sea30s02-in-f4.1e100.net (172.217.14.228): icmp_seq=2 ttl=51 time=5.02 ms
```

运行curl以验证元数据是否可删除。面向外部的接口必须是Gig1(eth0)。 否则，请检查Azure安全组、路由或可能阻止169.254.169.254.169.254的其他功能不是可ping地址。

```
[guestshell@guestshell ~]$ curl -H Metadata:true
"http://169.254.169.254/metadata/instance?api-version=2018-04-02"
{"compute":{"location":"westus2","name":"csr-david-2","offer":"cisco-csr-
1000v","osType":"Linux","placementGroupId":"","plan":{"name":"16_7","product":"cisco-csr-
1000v","publisher":"cisco"},"platformFaultDomain":"0","platformUpdateDomain":"0","publicKey
s":[],"publisher":"cisco","resourceGroupName":"RG-David-
2","sku":"16_7","subscriptionId":"09e13fd4-def2-46aa-a056-
xxxxxxxxxxx","tags":"","version":"16.7.120171201","vmId":"f8f32b48-daa0-4053-8ba4-
xxxxxxxxxxxxx","vmScaleSetName":"","vmSize":"Standard_DS2_v2","zone":""},"network":{"interf
ace":[{"ipv4":{"ipAddress":[{"privateIpAddress":"10.3.0.5","publicIpAddress":"21.53.135.210
"}],"subnet":[{"address":"10.3.0.0","prefix":"24"}]},"ipv6":{"ipAddress":[]},"macAddress":"
000D3A93F"},{"ipv4":{"ipAddress":[{"privateIpAddress":"10.3.1.5","publicIpAddress":""}],"su
bnet":[{"address":"10.3.1.0","prefix":"24"}]},"ipv6":{"ipAddress":[]},"macAddress":"000D3A9
61"}]}}[
```

3. 安装Python软件包。 **注意**：请勿使用sudo模式安装软件包。确保使用**- user**选项。如果无法执行所有三个步骤，则会将软件包安装到错误的文件夹中。这可能导致ImportErrors。要修复安装不正确的软件包，可能需要运行IOS命令**guestshell destroy** 并重新开始。

```
[guestshell@guestshell ~]$ pip install csr_azure_guestshell~=1.1 --user
[guestshell@guestshell ~]$ pip install csr_azure_ha~=1.0 --user
[guestshell@guestshell ~]$ source ~/.bashrc
```

4. 确保软件包正确安装在**/home/guestshell/.local/lib/python2.7/site-packages中**。

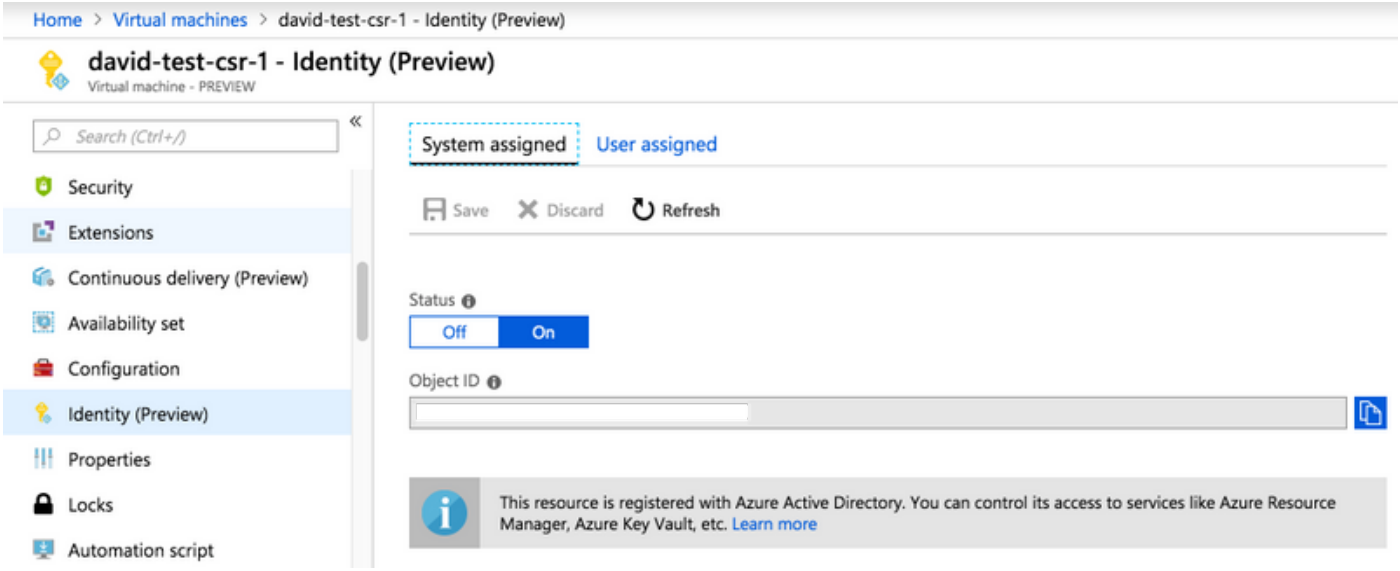
```
[guestshell@guestshell ~]$ which show_node.py
~/.local/lib/python2.7/site-packages/csr_azure_ha/client_api/show_node.py
```

步骤3.配置CSR1000v API调用的身份验证。

允许CSR1000v对Azure进行API调用的方法有2种。

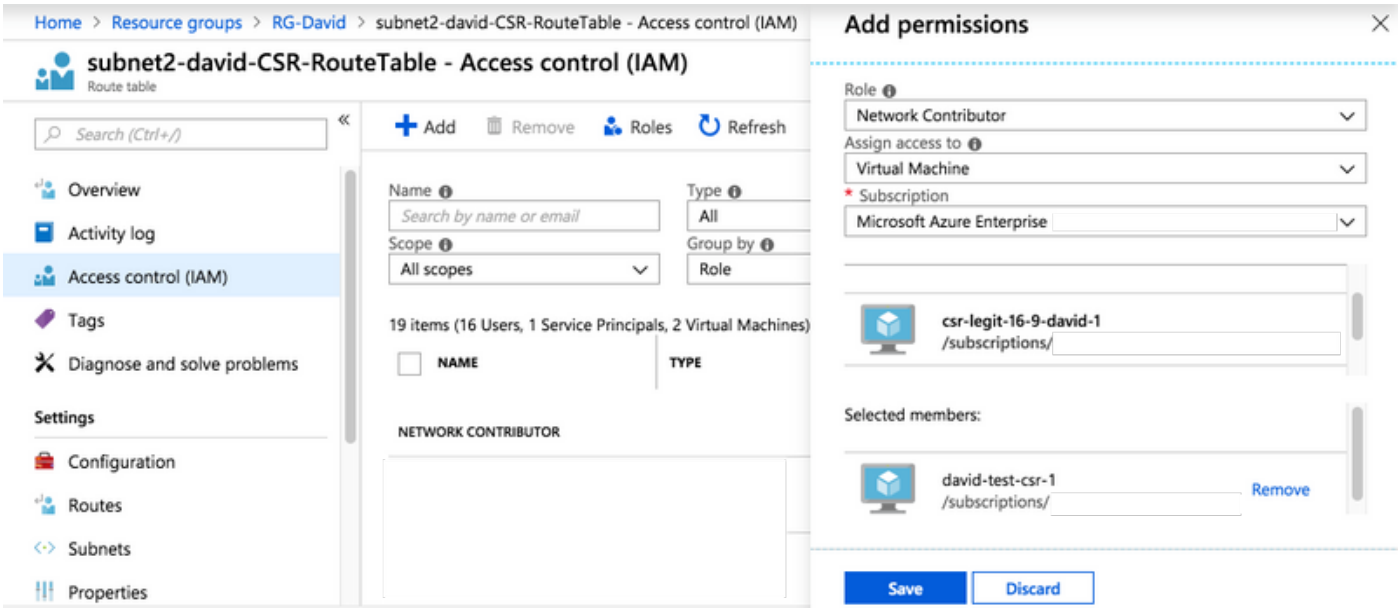
1. Azure Active Directory(AAD) — 这是标准HAv1方法，也可在HAv2中使用。请记**下要在create_node.py脚本中**使用的租户ID、app-id、app-key。有关详细信息，请访问在Microsoft Azure Active Directory中创建应用。 **注意**：HAv1中使用的应用密钥是编码密钥。HAv2中使用的应用密钥是未编码的密钥。如果未记录未编码的密钥，则可能需要创建新密钥，因为密钥不可恢复。
2. Microsoft拥有托管服务身份(MSI)服务，可自动为虚拟机创建应用。有关MSI的详细信息，请访问https://docs.microsoft.com/en-us/azure/active-directory/managed-service-identity/overview。HA版本2可以使用MSI服务对Cisco CSR 1000v进行身份验证。HA版本1不能使用MSI。

步骤1.为每个CSR1000v虚拟机启用MSI。导航至Azure门户中的VM。导航至**Identity**，然**后单击System Assigned > On > Save**。



步骤2.在“子**网路由表**”下，为了允许来自CSR1000v路由器的API调用，请选择**访问控制(IAM)，然**后单击**添加**。

步骤3.选择**Role - Network Contributor**。选择**将访问权限分配到 — 虚拟**机。选择适当**的订阅**。从已打开其MSI的列表中选择VM。



步骤4.在Guestshell中配置HAv2。

1. 使用**create_node.py脚**本添加HA配置。 要检查所有标志参数定义，请查看Cisco CSR 1000v Deployment Guide for Microsoft Azure的表3和表4。 本示例使用AAD身份验证，**该身份验证需要app-id(a)、tenant-id(d)和app-key(k)标**志。 如果使用MSI身份验证，则不需要这些额外标志。节**点[-i]标**志是任意数字。 如果需要更新到多个路由表，请使用唯一节点编号创建多个节点。

```
create_node.py -i 100 -p azure -s 09e13fd4-def2-46aa-a056-xxxxxxxxxx -g RG-David -t
subnet2-david-CSR-RouteTable -r 8.8.8.8/32 -n 10.3.1.4 -a 1e0f69c3-b6aa-46cf-b5f9-
xxxxxxxxx -d ae49849c-2622-4d45-b95e-xxxxxxxxx -k bDEN1k8batJqpeqjAuUvaUCZn5Md6rWEi=
```

2. 使用**set_params.py**添加或更改单个参数。

```
set_params.py -i 100 [option1] [option2]
```

3. 使用**clear_params.py**清除各个参数。

```
clear_params.py -i 100 [option1] [option2]
```

4. 使用delete_node.py删除节点。
   ```
   delete_node.py -i 100
   ```

步骤5.将EEM配置为触发故障切换。

带peerFail选项的node_event.py脚本是HAv2如何触发故障转移并更新Azure路由表。在这里，您可以灵活地编写自己的逻辑程序。可以在IOS中使用EEM运行node_event.py，或在guestshell中编写python脚本。

一个示例是使用EEM捕获接口关闭状态以触发node_event.py。

```
event manager applet HAv2_interface_flap
 event syslog pattern "Interface GigabitEthernet2, changed state to down"
 action 1 cli command "enable"
 action 2 cli command "guestshell run node_event.py -i 100 -e peerFail"
```

可以在guestshell中手动运行node_event.py以测试实际故障切换。

```
[guestshell@guestshell ~]$ node_event.py -i 100 -e peerFail
```

HAv2还可以使用revert选项将路由还原回原始路由器。这是模拟抢占的可选配置。create_node.py中的 — m主标志需要在主路由器上设置。这是使用BFD监控接口状态的示例。

```
event manager applet bfd_session_up
 event syslog pattern ".*BFD_SESS_UP.*"
 action 1 cli command "enable"
 action 2 cli command "guestshell run node_event.py -i 100 -e revert"

[guestshell@guestshell ~]$ set_params.py -i 100 -m
```

验证

1. 确保所有三个进程都处于活动状态。
   ```
   systemctl status auth-token
   systemctl status azure-ha
   systemctl status waagent
   ```
2. 重新启动任何失败的。
   ```
   sudo systemctl start waagent
   sudo systemctl start azure-ha
   sudo systemctl start auth-token
   ```
3. 验证由create_node.py添加的配置的两种方法。
   ```
   show_node.py -i 100

   [guestshell@guestshell ~]$ cat azure/HA/node_file
   {'appKey': 'bDEN1k8batJqWEiGXAxSR4Y=', 'index': '100', 'routeTableName': 'subnet2-david-
   CSR-RouteTable', 'route': '8.8.8.8/32', 'nextHop': '10.3.1.4', 'tenantId': 'ae49849c-2622-
   4d45-b95e-xxxxxxxxxx', 'resourceGroup': 'RG-David', 'appId': '1e0f69c3-b6aa-46cf-b5f9-
   xxxxxxxxxx', 'subscriptionId': '09e13fd4-def2-46aa-a056-xxxxxxxxxx', 'cloud': 'azure'}
   ```
4. 软模拟备用路由器上的故障切换。这实际上不会导致故障切换，但会验证配置是否有效。检查步骤6中的日志。
   ```
   node_event.py -i 100 -e verify
   ```
5. 在备用路由器上触发实际故障切换事件。在Azure中，检查路由表是否更新了通往新跃点的路由。检查步骤6中的日志。

```
node_event.py -i 100 -e peerFail
```

6. node_event.py在触发时生成2类日志。这有助于验证故障切换是否成功或排除故障。每次都生成新事件文件。但是，routeTableGetRsp每次都被覆盖，因此通常只有一个文件。

```
[guestshell@guestshell ~]$ ls -latr /home/guestshell/azure/HA/events/
total 5
drwxr-xr-x 3 guestshell root 1024 Sep 18 23:01 ..
drwxr-xr-x 2 guestshell root 1024 Sep 19 19:40 .
-rw-r--r-- 1 guestshell guestshell 144 Sep 19 19:40 routeTableGetRsp
-rw-r--r-- 1 guestshell guestshell 390 Sep 19 19:40 event.2018-09-19 19:40:28.341616
-rw-r--r-- 1 guestshell guestshell 541 Sep 18 23:09 event.2018-09-18 23:09:58.413523
```

故障排除

步骤1. Python软件包错误地安装在/usr/lib/python2.7/site-packages/中。销毁guestshell并执行配置步骤。

```
[guestshell@guestshell ~]$ create_node.py -h
bash: create_node.py: command not found

[guestshell@guestshell ~]$ ls /usr/lib/python2.7/site-packages/
```

正确的安装路径是~/.local/lib/python2.7/site-packages/。

```
[guestshell@guestshell ~]$ which show_node.py
~/.local/lib/python2.7/site-packages/csr_azure_ha/client_api/show_node.py
```

步骤2.如果在步骤3中未配置或配置错误，则可能会生成令牌错误。对于AAD身份验证，如果使用的应用密钥无效或URL编码，则在触发node_event.py后可能会看到身份验证错误。

```
[guestshell@guestshell ~]$ cat /home/guestshell/azure/HA/events/routeTableGetRsp
{"error":{"code":"AuthenticationFailedMissingToken","message":"Authentication failed. The
'Authorization' header is missing the access token."}}

[guestshell@guestshell ~]$ cat /home/guestshell/azure/HA/events/event.2018-09-19\
23\:02\:55.581684

Event type is verify
appKey zGuYMyXQha5Kqe8xdufhUJ9eX%2B1zIhLsuw%3D
index 100
routeTableName subnet2-david-CSR-RouteTable
route 8.8.8.8/32
nextHop 10.3.1.4
tenantId ae49849c-2622-4d45-b95e-xxxxxxxxxx
resourceGroup RG-David
appId 1e0f69c3-b6aa-46cf-b5f9-xxxxxxxxxx
subscriptionId 09e13fd4-def2-46aa-a056-xxxxxxxxxx
cloud azure
All required parameters have been provided
Requesting token using Azure Active Directory
Token=
Failed to obtain token
Reading route table
Route GET request failed with code 401
```

步骤3.如果tenant-id或app-id不正确。

```
[guestshell@guestshell ~]$ cat azure/tools/TokenMgr/token_get_rsp
{"error":"invalid_request","error_description":"AADSTS90002: Tenant 1e0f69c3-b6aa-46cf-b5f9-
xxxxxxxxxx not found. This may happen if there are no active subscriptions for the tenant. Check
with your subscription administrator.\r\nTrace ID: 8bc80efc-f086-46ec-83b9-
xxxxxxxxxxx\r\nCorrelation ID: 2c6062f8-3a40-4b0e-83ec-xxxxxxxxxxx\r\nTimestamp: 2018-09-19
23:58:02Z","error_codes":[90002],"timestamp":"2018-09-19 23:58:02Z","trace_id":"8bc80efc-f086-
46ec-83b9-xxxxxxxxxxx","correlation_id":"2c6062f8-3a40-4b0e-83ec-xxxxxxxxxxx"}
```

步骤4.在软件包安装过程中，可能已使用sudo模式， — 未包括用户，或源~/.bashrc未运行。这会导致create_node.py失败或生成ImportError。

```
[guestshell@guestshell ~]$ create_node.py -i 1 -p azure -s d91490ec -g RG -t RT -r 10.12.0.0/11
-n 10.2.0.31 -m secondary
/usr/lib64/python2.7/site-packages/cryptography/hazmat/primitives/constant_time.py:26:
CryptographyDeprecationWarning: Support for your Python version is deprecated. The next version
of cryptography will remove support. Please upgrade to a 2.7.x release that supports
hmac.compare_digest as soon as possible.
utils.DeprecatedIn23,
create_node -i 1 -p azure -s d91490ec -g RG -t RT -r 10.12.0.0/11 -n 10.2.0.31 -m secondary
failed

[guestshell@guestshell ~]$ create_node.py -i 1 -p azure -s d91490ec -g RG -t RT -r 10.1.0.0/18 -
n 10.2.0.31 -m secondary
Traceback (most recent call last):
  File "/usr/bin/create_node.py", line 5, in
    import ha_api
ImportError: No module named ha_api
```

步骤5.检查软件包安装历史记录。

```
[guestshell@guestshell ~]$ cat azure/HA/install.log
Installing the Azure high availability package
Show the current PATH
/usr/local/bin:/usr/bin:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_ha/client_api
Show the current PYTHONPATH
:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_guestshell:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_guestshell/TokenMgr:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_guestshell/MetadataMgr:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_guestshell/bin:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_ha/client_api:/home/guestshell/.local/lib/python2.7/site-
packages/csr_azure_ha/server
```

步骤6.检查HA配置日志。

```
[guestshell@guestshell ~]$ cat azure/HA/azha.log
2018-09-24 16:56:29.215743 High availability server started with pid=7279
2018-09-24 17:03:20.602579 Server processing create_node command
2018-09-24 17:03:20.602729 Created new node with index 100
```

步骤6.运行debug_ha.sh脚本，将所有日志文件收集到一个tar文件中。

```
[guestshell@guestshell ~]$ bash ~/azure/HA/debug_ha.sh
```

文件放在bootflash中，可从guestshell和IOS访问。

```
[guestshell@guestshell ~]$ ls /bootflash/ha_debug.tar
/bootflash/ha_debug.tar
```

```
csr-david-2#dir | i debug
   28  -rw-            92160  Sep 27 2018 22:42:54 +00:00  ha_debug.tar
```