Catalyst 9K平台上的GuestShell故障排除

目录

简介

本文档介绍如何对Cat9K交换机上的访客外壳问题进行故障排除。

先决条件

要求

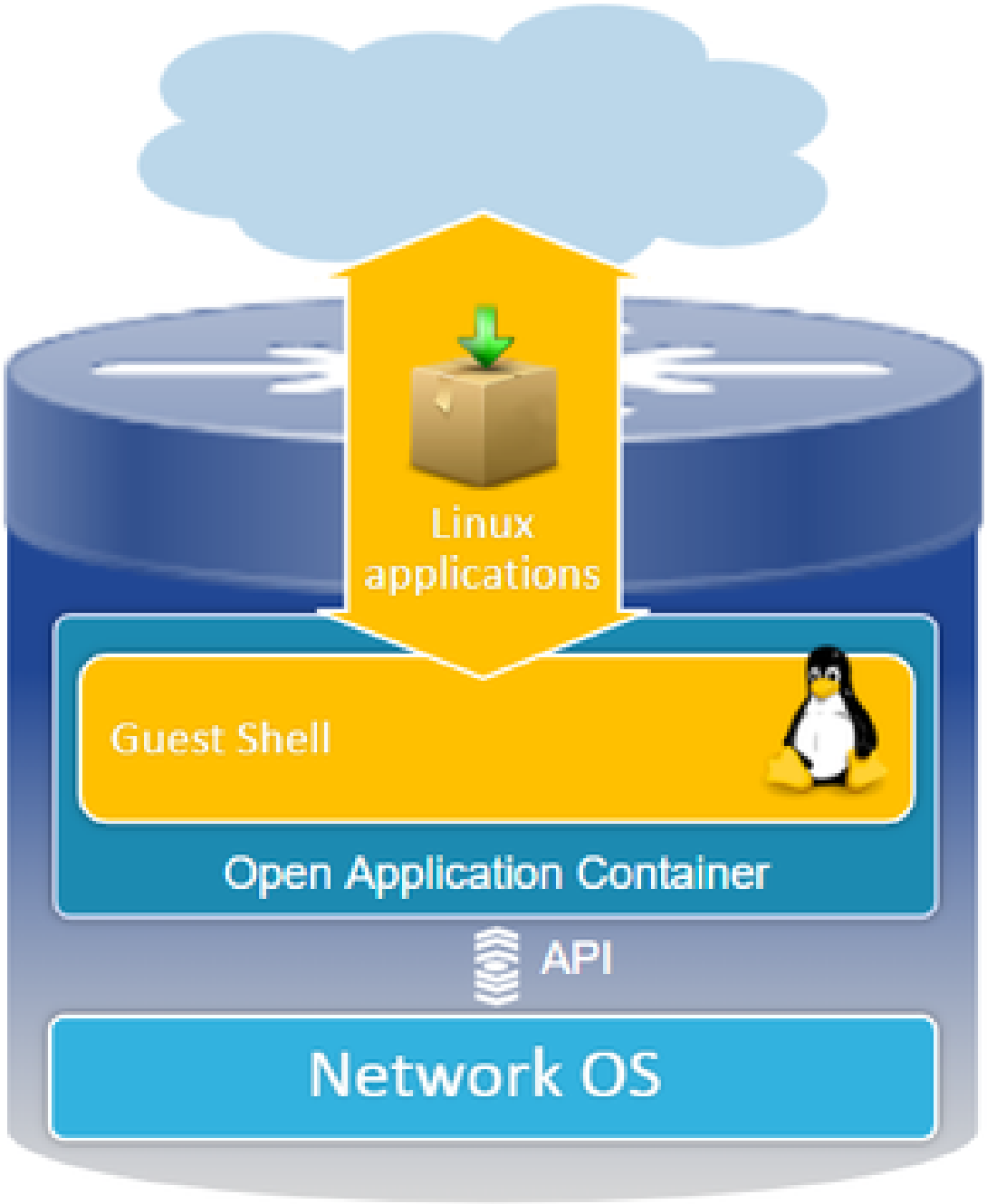
- 基本了解Cisco IOS® XE软件
- Linux文件系统
- Linux命令

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Catalyst 9200
- Catalyst 9300

- Catalyst 9400
- Catalyst 9500
- Catalyst 9600
- Cisco IOS XE 17.9.1及更高版本



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。

背景信息

Guest Shell概述

- Guest Shell提供在Linux容器(LXC)内运行的隔离执行环境。
- 网络接入:管理员可以通过Linux网络接口连接到网络，从而实现强大的连接和管理。
- Bootflash访问：可以直接访问Bootflash存储，有助于高效的文件管理和系统操作。
- Cisco IOS CLI访问：管理员能够直接与Cisco IOS命令行界面交互，从而实现无缝集成和控制。
- 脚本执行：该环境支持安装和执行Python脚本，从而实现自动化和自定义。
- 应用支持：32位和64位Linux应用程序均可安装和运行，提供灵活性和各种功能可能性。

故障排除

.

配置

1.启用IOX。

<#root>

```
Switch#
conf terminal
Switch(config)#
iox

Switch(config)#
```

等待1-5分钟，使IOX初始化。确保IOX正常运行。您可以检查CLI会话期间显示的消息，也可以查看show命令的输出。

<#root>

```
*Mar 10 15:35:40.206: %UICFGEXP-6-SERVER_NOTIFIED_START: Switch 1 R0/0: psd: Server iox has been notifi
*Mar 10 15:35:51.186: %IOX-3-PD_PARTITION_CREATE: Switch 1 R0/0: run_ioxn_caf: IOX may take upto 5 mins
*Mar 10 15:37:56.643: %IOX-3-IOX_RESTARTABITLITY: Switch 1 R0/0: run_ioxn_caf: Stack is in N+1 mode, di
*Mar 10 15:38:05.835: %IM-6-IOX_ENABLEMENT: Switch 1 R0/0: ioxman: IOX is ready

.
Switch#show iox-service

IOx Infrastructure Summary:
---------------------------
IOx service (CAF)              :
Running
IOx service (HA)               :
Running
```

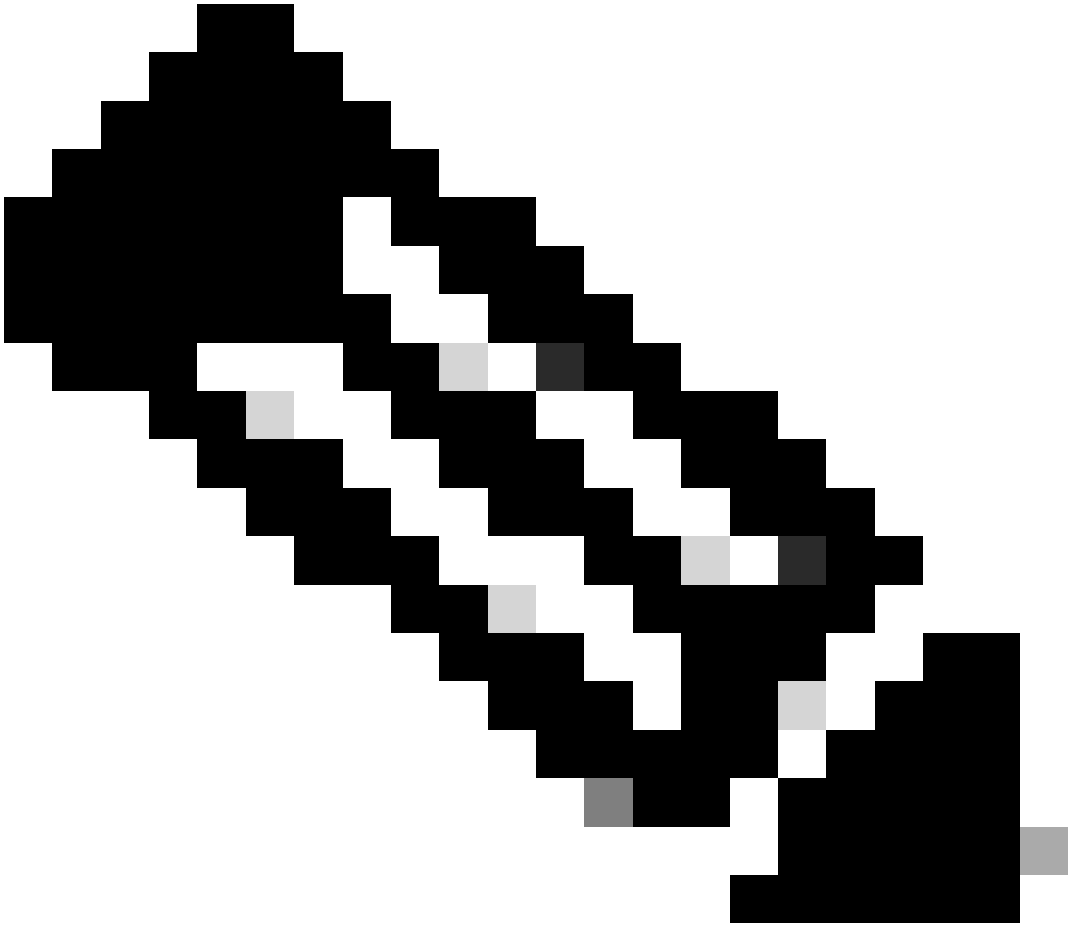
```
IOx service (IOxman)           :
Running

IOx service (Sec storage)      :
Running

Libvirtd 5.5.0                 :
Running

Dockerd v19.03.13-ce           :
Running

Sync Status                    : Disabled
```

2.网络配置。

本示例使用AppGigabitEthernet接口提供网络访问。

<#root>

```
!
interface AppGigabitEthernet1/0/1
 switchport trunk allowed vlan 50
 switchport mode trunk
!
app-hosting appid guestshell
 app-vnic AppGigabitEthernet trunk
  vlan 50 guest-interface 0
guest-ipaddress 192.168.10.10 netmask 255.255.255.0
 name-server0 192.168.10.254
end
!
```

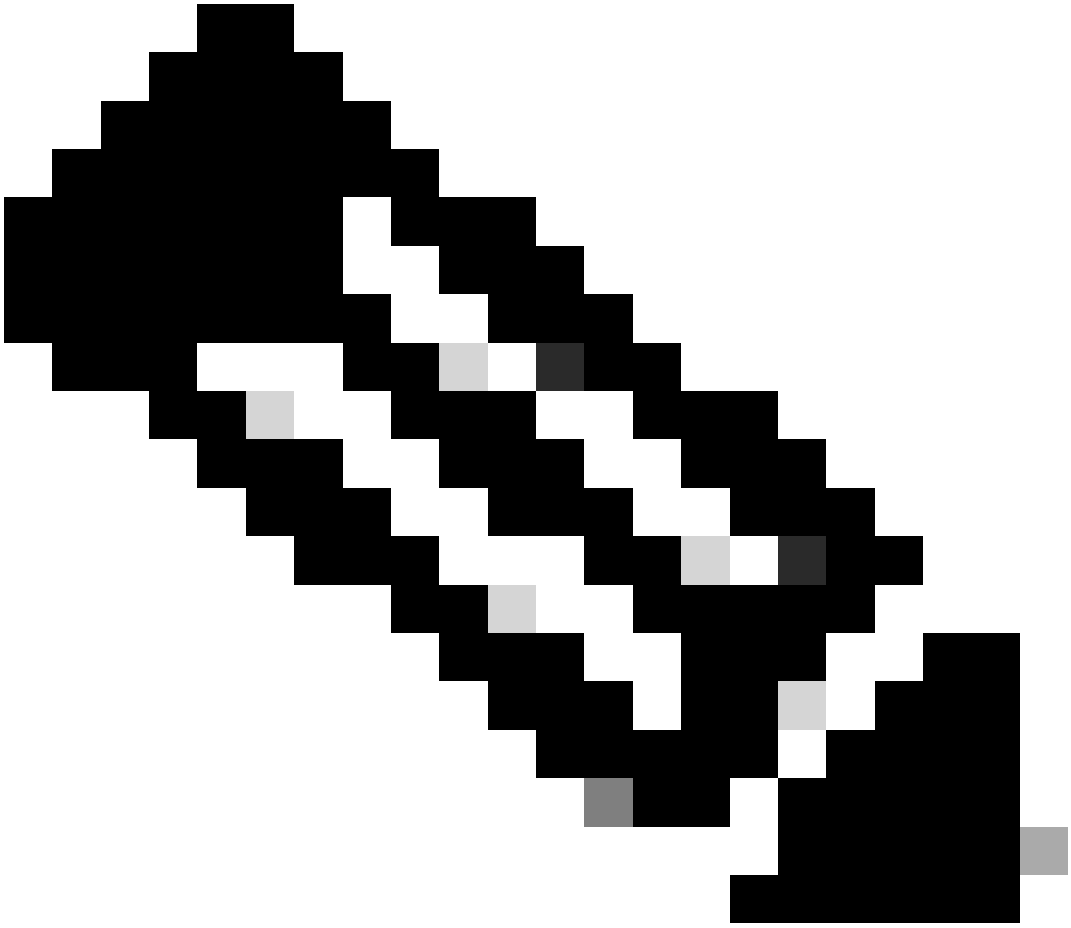


注意：GuestShell使用静态IP地址，因为默认情况下它不包括DHCP客户端服务。您可以安装DHCP客户端服务以动态获取IP地址。

本示例使用管理接口(Gi0/0)提供网络访问。在管理模式中，Guestshell使用网络管理端口将Cisco IOS配置的管理端口IP地址用作网关。

<#root>

```
!
app-hosting appid guestshell


app-vnic management guest-interface 0

 name-server0 8.8.8.8
!
```



注意：使用管理接口时，Guestshell eth0接口的默认配置硬编码为IP地址192.168.30.2。

您可以配置虚拟端口组接口：

- 虚拟端口组配置：使用静态IP地址建立虚拟端口组。
- Internet 连接:确保前端口接口连接到Internet以允许外部访问。
- NAT 配置:在虚拟端口组和前端口之间设置NAT，以促进网络地址转换。
- GuestShell IP分配：将IP地址分配给与虚拟端口组接口位于同一子网内的GuestShell。

```
!
interface VirtualPortGroup0
 ip address 192.168.35.1 255.255.255.0
 ip nat inside
!
interface GigabitEthernet1/0/3
 no switchport
 ip address 192.168.100.10 255.255.255.0
 ip nat outside
!
```

```
ip route 0.0.0.0 0.0.0.0 192.168.100.254
ip route vrf Mgmt-vrf 0.0.0.0 0.0.0.0 192.168.10.254
ip nat inside source static tcp 192.168.35.2 7023 192.168.100.10 7023 extendable
!
!
ip access-list standard NAT_ACL
 10 permit 192.168.0.0 0.0.255.255
!
app-hosting appid guestshell
 app-vnic gateway1 virtualportgroup 0 guest-interface 0
  guest-ipaddress 192.168.35.2 netmask 255.255.255.0
 app-default-gateway 192.168.35.1 guest-interface 0
 name-server0 8.8.8.8
end
```

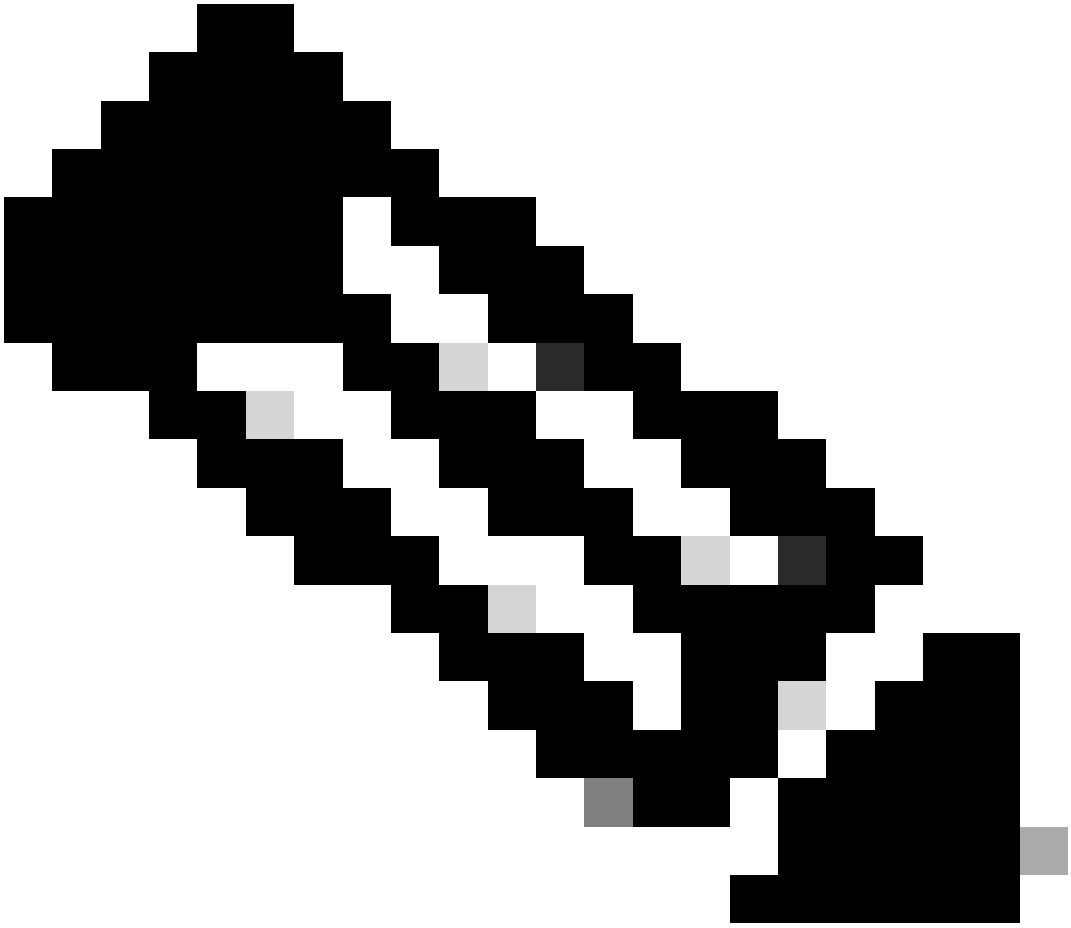
3.启用guestshell。

<#root>

```
Switch#
```

guestshell enable

```
Interface will be selected if configured in app-hosting
Please wait for completion
guestshell installed successfully
Current state is: DEPLOYED
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully
```



注意：guestshell enable命令是安装guestshell的唯一方法。使用应用托管CLI激活/连接/停用可用于控制guestshell。建议您改用guestshell exec CLI。

Guestshell生命周期

Guestshell disable:Disable guesthell会删除对guestshell的访问并终止当前会话。保留文件/数据，您可以使用guestshell enable恢复访问。

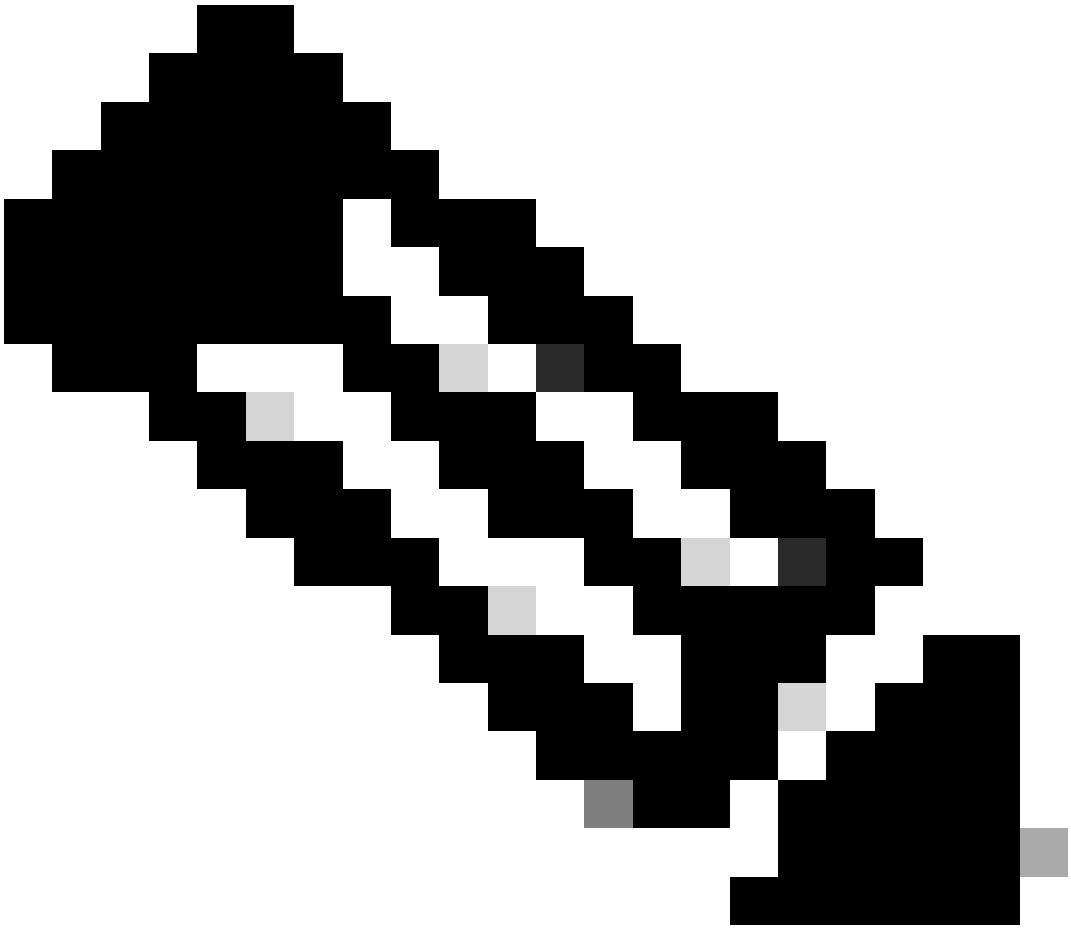
```
<#root>

Switch#

guestshell disable

Guestshell disabled successfully
```

Guestshell destroy:这将以不可逆的方式删除访客shell文件系统。所有文件/数据/脚本/设置外壳/已

安装的软件包和模块。

<#root>

```
Switch#
guestshell destroy

Guestshell destroyed successfully
```



注意：运行此命令时，所有数据都会不可逆地丢失。

Guestshell运行：Guestshell运行命令guestshell run bash在Guest Shell中创建一个shell，允许您在/bin和/sbin下使用任何Linux二进制文件。

<#root>

```
Switch#
```

guestshell run bash

```
[guestshell@guestshell ~]$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=2 ttl=254 time=0.517 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=254 time=0.552 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=254 time=0.447 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=254 time=0.549 ms
```

Guestshell运行python:使用此命令启动交互式python解释程序。

<#root>

```
Switch#
```

guestshell run python3

```
Python 3.6.8 (default, Dec 22 2020, 19:04:08)
[GCC 8.4.1 20200928 (Red Hat 8.4.1-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

验证

您可以使用以下命令验证Guestshell:

<#root>

```
Switch#
```

show app-hosting detail appid guestshell

```
App id                 : guestshell
Owner                  : iox
State                  :
```

RUNNING

```
Application
  Type                 : lxc
  Name                 : GuestShell
  Version              : 3.3.0
  Description          : Cisco Systems Guest Shell XE for x86_64
  Author               : Cisco Systems
  Path                 : /guestshell/:guestshell.tar
  URL Path             :
  Multicast            : yes
Activated profile name : custom

Resource reservation
  Memory               : 256 MB
  Disk                 : 1 MB
  CPU                  : 800 units
  CPU-percent          : 11 %
```

```
  VCPU                : 1

Platform resource profiles
  Profile Name                  CPU(unit)  Memory(MB)  Disk(MB)
  ----------------------------------------------------------------

Attached devices
  Type              Name               Alias
  ---------------------------------------------
  serial/shell     iox_console_shell   serial0
  serial/aux       iox_console_aux     serial1
  serial/syslog    iox_syslog          serial2
  serial/trace     iox_trace           serial3

Network interfaces
   ---------------------------------------
eth0:
   MAC address         : 52:54:dd:5b:c4:b8
   IPv4 address        : 192.168.30.2
   IPv6 address        : ::
   Network name        : mgmt-bridge200

Port forwarding
  Table-entry  Service  Source-port  Destination-port
  ------------------------------------------------
  Switch#show app-hosting list
App id                                   State
---------------------------------------------------------
guestshell
```

RUNNING

```
Switch#
```

guestshell run sudo ifconfig

```
eth0: flaguestshell=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.10  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::5054:ddff:fece:a7c9  prefixlen 64  scopeid 0x20
        ether 52:54:dd:ce:a7:c9  txqueuelen 1000  (Ethernet)
        RX packets 3  bytes 266 (266.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9  bytes 726 (726.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flaguestshell=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 338  bytes 74910 (73.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 338  bytes 74910 (73.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

调整资源大小

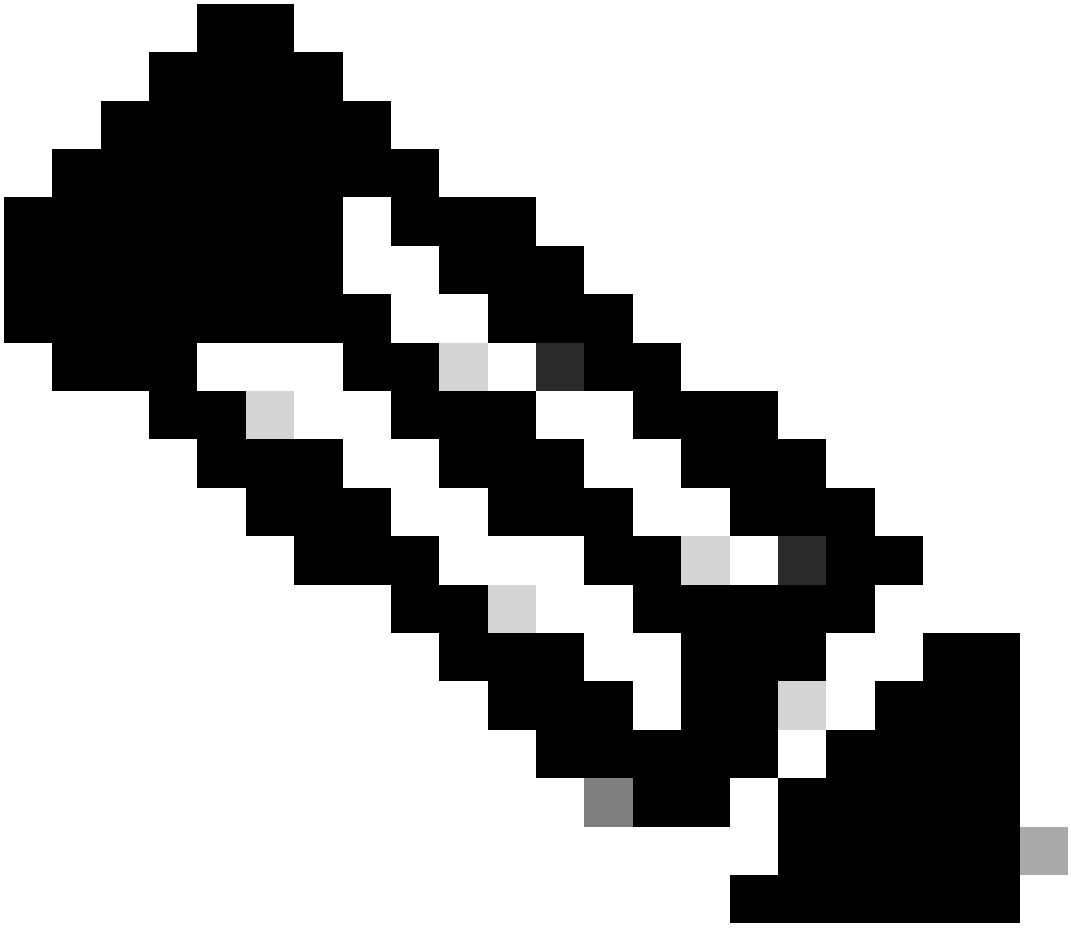
<#root>

```
!
app-hosting appid guestshell
 app-vnic management guest-interface 0


app-resource profile custom

  cpu 1000
  memory 512
  persist-disk 200
!
```

您需要禁用然后启用Guest Shell以使更改生效。

<#root>

```
Switch#

guestshell disable

Guestshell disabled successfully
Switch#

guestshell enable

*Mar 11 01:17:46.841: %SYS-5-CONFIG_I: Configured from console by coguestshell enable
Interface will be selected if configured in app-hosting
Please wait for completion
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully
Switch#

show app-hosting detail appid guestshell | sec Resource reservation


Resource reservation
  Memory               : 512 MB
  Disk                 : 200 MB
  CPU                  : 1110 units
  CPU-percent          : 15 %
  VCPU                 : 1
```



注意：在交换机上，资源大小调整最多允许达到平台的最大限制，请参阅Cisco Catalyst 9000平台硬件资源应用。

常见情况

DHCP 配置

问题：DHCP客户端二进制(DHCLIENT)不存在。

解决方案

您可以使用Yum实用程序并使用sudo yum install dhcp-client命令来安装DHCP客户端。但是，请注意，CentOS Stream 8的存储库不再可用。

1.必须避免在应用托管中设置IP地址以允许DHCP正常工作。

```
!
interface AppGigabitEthernet1/0/1
 switchport trunk allowed vlan 50
 switchport mode trunk
!
app-hosting appid guestshell
 app-vnic AppGigabitEthernet trunk
  vlan 50 guest-interface 0
name-server0 8.8.8.8
!
```

2.将mirrorlist更改为yum repo中的baseurl vault。

<#root>

Switch#

guestshell run bash

[guestshell@guestshell ~]$

sudo find /etc/yum.repos.d/ -type f -exec sed -i 's/mirrorlist=/#mirrorlist=/g' {} +

[guestshell@guestshell ~]$

sudo find /etc/yum.repos.d/ -type f -exec sed -i 's/#baseurl=/baseurl=/g' {} +

[guestshell@guestshell ~]$

sudo find /etc/yum.repos.d/ -type f -exec sed -i 's/mirror.centos.org/

/g' {} +

```
[guestshell@guestshell ~]$ cat /etc/yum.repos.d/CentOS-Stream-RealTime.reporepo
# CentOS-Stream-RealTime.repo
# The mirrorlist system uses the connecting IP address of the client and the
# update status of each mirror to pick current mirrors that are geographically
# close to the client.  You should use this for CentOS updates unless you are
# manually picking other mirrors.
#
# If the mirrorlist does not work for you, you can try the commented out
# baseurl line instead.

[rt]
name=CentOS Stream $releasever - RealTime
#mirrorlist=http://mirrorlist.centos.org/?release=$stream&arch=$basearch&repo=RT&infra=$infra
```

baseurl=http://

/$contentdir/$stream/RT/$basearch/os/

```
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
```

3.安装软件包。

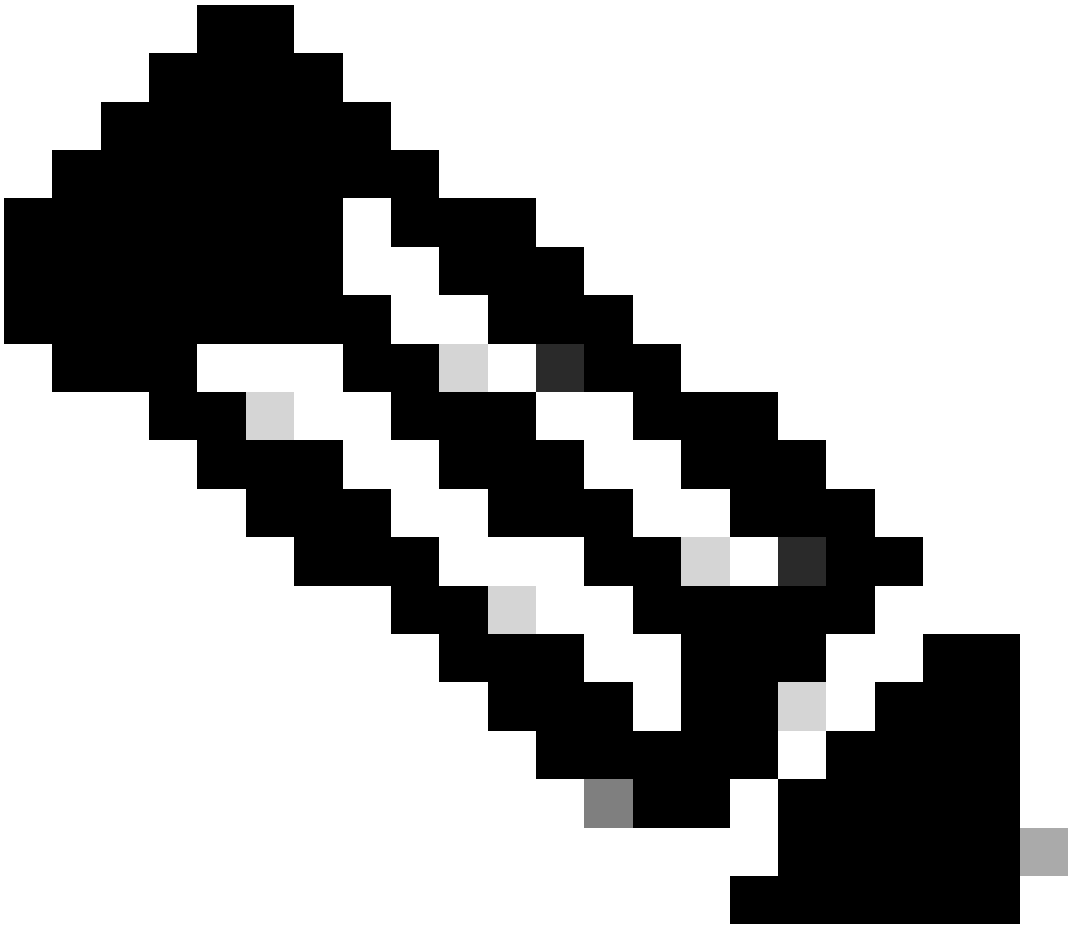
<#root>

```
guestshell@guestshell ~]$
```

sudo yum install dhcp-client

```
ast metadata expiration check: 0:50:34 ago on Wed Mar 12 17:44:46 2025.
Dependencies resolved.
================================================================================
 Package                Architecture Version                 Repository    Size
================================================================================
Installing:
 dhcp-client            x86_64       12:4.3.6-50.el8         baseos       319 k
Installing dependencies:
 bind-export-libs       x86_64       32:9.11.36-13.el8       baseos       1.1 M
 dhcp-common            noarch       12:4.3.6-50.el8         baseos       208 k
 dhcp-libs              x86_64       12:4.3.6-50.el8         baseos       148 k

Transaction Summary
================================================================================
Install  4 Packages

Total download size: 1.8 M
Installed size: 3.9 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): dhcp-client-4.3.6-50.el8.x86_64.rpm      284 kB/s | 319 kB     00:01
(2/4): dhcp-common-4.3.6-50.el8.noarch.rpm      171 kB/s | 208 kB     00:01
(3/4): dhcp-libs-4.3.6-50.el8.x86_64.rpm        572 kB/s | 148 kB     00:00
(4/4): bind-export-libs-9.11.36-13.el8.x86_64.r 577 kB/s | 1.1 MB     00:02
--------------------------------------------------------------------------------
Total                                           908 kB/s | 1.8 MB     00:02
CentOS Stream 8 - BaseOS                        1.6 MB/s | 1.6 kB     00:00
Importing GPG key 0x8483C65D:
 Userid     : "CentOS (CentOS Official Signing Key) <CentOS Official Signing Key>"
 Fingerprint: 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : dhcp-libs-12:4.3.6-50.el8.x86_64                       1/4
  Installing       : dhcp-common-12:4.3.6-50.el8.noarch                     2/4
  Installing       : bind-export-libs-32:9.11.36-13.el8.x86_64              3/4
  Running scriptlet: bind-export-libs-32:9.11.36-13.el8.x86_64              3/4
```

```
  Installing       : dhcp-client-12:4.3.6-50.el8.x86_64                    4/4
  Running scriptlet: dhcp-client-12:4.3.6-50.el8.x86_64                    4/4
  Verifying        : bind-export-libs-32:9.11.36-13.el8.x86_64             1/4
  Verifying        : dhcp-client-12:4.3.6-50.el8.x86_64                    2/4
  Verifying        : dhcp-common-12:4.3.6-50.el8.noarch                    3/4
  Verifying        : dhcp-libs-12:4.3.6-50.el8.x86_64                      4/4

Installed:
  bind-export-libs-32:9.11.36-13.el8.x86_64  dhcp-client-12:4.3.6-50.el8.x86_64
  dhcp-common-12:4.3.6-50.el8.noarch         dhcp-libs-12:4.3.6-50.el8.x86_64

Complete!
```



注意：（对于CALO）可以为Yum配置代理。这允许Yum通过代理服务器连接进行软件包下载：

[guestshell@guestshell ~]$ echo "proxy=http://<IP_address:port>/" | sudo tee -a /etc/yum.conf > /dev/null

---

4..请求eth0的DHCP IP地址。

<#root>

```
[guestshell@guestshell ~]$
```

sudo dhclient eth0

5.通过检查IP地址分配来验证DHCP客户端是否正常工作。

<#root>

```
Switch#
```

guestshell run ifconfig

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
```

inet 192.168.10.2 netmask 255.255.255.0 broadcast 192.168.10.255

```
        inet6 fe80::5054:ddff:fea0:4aef  prefixlen 64  scopeid 0x20
        ether 52:54:dd:a0:4a:ef  txqueuelen 1000  (Ethernet)
        RX packets 1516  bytes 2009470 (1.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 687  bytes 54603 (53.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 773  bytes 90658 (88.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 773  bytes 90658 (88.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

DNF数据包管理器更新失败

问题：由于以下错误，无法在Guestshell上完成sudo dnf update -y:

```
[guestshell@guestshell ~]$ sudo dnf upgrade --refresh
Warning: failed loading '/etc/yum.repos.d/CentOS-Base.repo', skipping.
```

解决方案

1.重新安装并升级RPM包。

<#root>

```
! Clean packages

[guestshell@guestshell ~]$ sudo dnf clean all

! Reinstall and update the tpm2-tss package:

[guestshell@guestshell ~]$

sudo dnf install tpm2-tss-2.3.2-3.el8

[guestshell@guestshell ~]$

sudo dnf upgrade rpm
```

2.单独安装git软件包。

<#root>

```
[guestshell@guestshell ~]$

sudo dnf install git -y
```

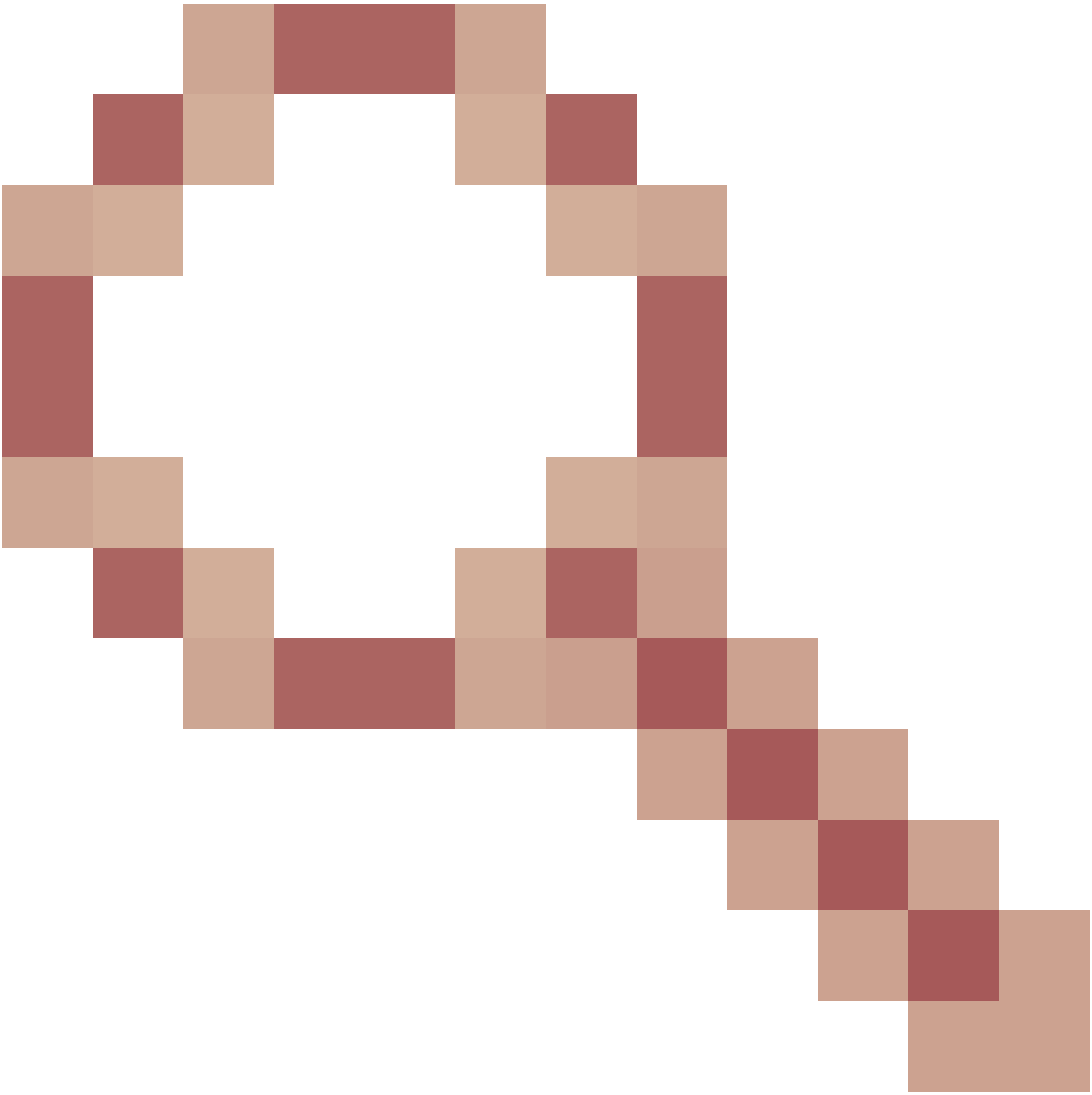
升级后丢失的Guestshell访问

问题：升级到版本17.08.01或更高版本后，Guest Shell变得不可访问，从而导致无法执行脚本。

```
Switch#guestshell run bash
Switch#
```

解决方案

此问题与思科漏洞ID CSCwi63075相关



，在交换机上启用FIPS时，在升级期间触发。

1.禁用FIPS。

<#root>

Switch#

configure terminal

Switch(config)#

no fips authorization-key

FIPS: Authorization-key erased ONLY from the Flash.
But the authorization-key is still operational. Use

```
"reload"

 command for complete removal of key and to enter into non fips-mode.
Make sure to remove fips key from all the members of the stack individually
```

2.您可以升级到包含修补程序、版本17.12.04或任何更高版本的其中一个版本。

限制IPv6编址

问题：此问题涉及在Cisco Guestshell环境中的接口上出现意外的IPv6地址。但是，并非有意配置。

在使用IPV4的交换机上观察到的配置如下：

<#root>

```
!
app-hosting appid guestshell
 app-vnic AppGigabitEthernet trunk
  vlan 50 guest-interface 0

guest-ipaddress 192.168.20.10 netmask 255.255.255.0  <-- IPv4 address configured

 app-default-gateway 192.168.20.1 guest-interface 0
 app-resource profile custom
 name-server1 192.168.20.1
 !
```

在Guestshell环境中运行ifconfig命令时，IPv4和IPv6地址都变为可见。

<#root>

```
lat1-2-ssw01.gts#guestshell run bash
[guestshell@guestshell ~]$

sudo ifconfig

eth0: flags=4163  mtu 1500
        inet 192.168.20.10  netmask 255.255.255.0 broadcast 192.168.20.255

 inet6 2620:119:5022:515:5054:ddff:fe41:c643  prefixlen 64  scopeid 0x0
        inet6 fe80::5054:ddff:fe41:c643  prefixlen 64  scopeid 0x20

        ether 52:54:dd:41:c6:43  txqueuelen 1000  (Ethernet)
        RX packets 7829  bytes 1750981 (1.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5551  bytes 744320 (726.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 292  bytes 63812 (62.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 292  bytes 63812 (62.3 KiB)
```

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

解决方案

步骤 1：在Guestshell中禁用IPv6路由和编址。

<#root>

```
lat1-2-ssw01.gts#
guestshell run bash
[guestshell@guestshell ~]$
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
[guestshell@guestshell ~]$
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
[guestshell@guestshell ~]$
sudo sysctl -w net.ipv6.conf.lo.disable_ipv6=1
```

步骤 2：检验IPv6是否已禁用。

```
[guestshell@guestshell ~]$ /sbin/ifconfig
eth0: flags=4163  mtu 1500
        inet 192.168.20.10  netmask 255.255.255.0 broadcast 192.168.20.255
        ether 52:54:dd:41:c6:43  txqueuelen 1000  (Ethernet)
        RX packets 7829  bytes 1750981 (1.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5551  bytes 744320 (726.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 292  bytes 63812 (62.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 292  bytes 63812 (62.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

运行Python脚本时出现磁盘空间错误

问题：此问题涉及在Guestshell中的嵌入式事件管理器(EEM)上运行的Python脚本。由于磁盘空间不足，脚本失败，错误为：

<#root>

```
guestshell run python3 /flash/guest-share/monitoring.py -rt True -bgp True
---- pushing bgp status ----
OSError: [Errno 28] No space left on device

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/flash/guest-share/monitoring_periodic_tasks.py", line 18, in
    print(bgp_status())
  File "/bootflash/guest-share/monitoring_bgp_status.py", line 15, in bgp_status
    vrf = cli.cli('show vrf')
  File "/usr/lib/python3.6/site-packages/cli/__init__.py", line 311, in cli
    _log_to_file("CLI execution invoked for '" + command + "'")
  File "/usr/lib/python3.6/site-packages/cli/__init__.py", line 87, in _log_to_file
    logfile.close()

OSError: [Errno 28] No space left on device

! This error indicates that the disk space allocated for logging command executions within the Guestshe
```

解决方案

要解决磁盘空间耗尽的问题，您需要增加Guestshell环境的持久磁盘大小：

1.修改应用资源配置文件，以增加持久磁盘的大小。

<#root>

```
Switch(config-app-hosting)#

app-resource profile custom

Switch(config-app-hosting-profile)#

persist-disk 100

Switch(config-app-hosting-profile)#

cpu 800

Switch(config-app-hosting-profile)#

memory 256

Switch(config-app-hosting-profile)#

end
```

2.保存配置并启用/禁用guestshell。

```
Switch# write memory
Switch#guestshell disable
Guestshell disabled successfully
Switch#guestshell enable
Interface will be selected if configured in app-hosting
Please wait for completion
```

```
guestshell installed successfully
Current state is: DEPLOYED
guestshell activated successfully
Current state is: ACTIVATED
guestshell started successfully
Current state is: RUNNING
Guestshell enabled successfully
```

日志记录Syslog

可以在guestshell中运行以下命令以显示系统日志消息：

<#root>

```
[guestshell@guestshell ~]$
```

sudo logger -p 1 "Priority 1"

```
[guestshell@guestshell ~]$
```

sudo cat /var/log/messages

```
Mar 11 02:05:24 localhost systemd[248]: user@0.service: Failed at step PAM spawning /usr/lib/systemd/sy
Mar 11 02:05:24 localhost systemd[1]: user@0.service: Failed with result 'protocol'.
Mar 11 02:05:24 localhost systemd[1]: Failed to start User Manager for UID 0.
Mar 11 02:05:24 localhost systemd[1]: Stopping /run/user/0 mount wrapper...
Mar 11 02:05:24 localhost systemd[1]: run-user-0.mount: Succeeded.
Mar 11 02:05:24 localhost systemd[1]: user-runtime-dir@0.service: Succeeded.
Mar 11 02:05:24 localhost systemd[1]: Stopped /run/user/0 mount wrapper.
Mar 11 02:05:24 localhost root[250]: Priority 1
[guestshell@guestshell ~]$ sudo cat /var/log/secure
Mar 11 02:05:24 localhost systemd[248]: pam_unix(systemd-user:account): expired password for user root
Mar 11 02:05:24 localhost sudo[246]: pam_systemd(sudo:session): Failed to create session: Start job for
Mar 11 02:05:24 localhost sudo[246]: pam_unix(sudo:session): session opened for user root by (uid=0)
```

跟踪日志

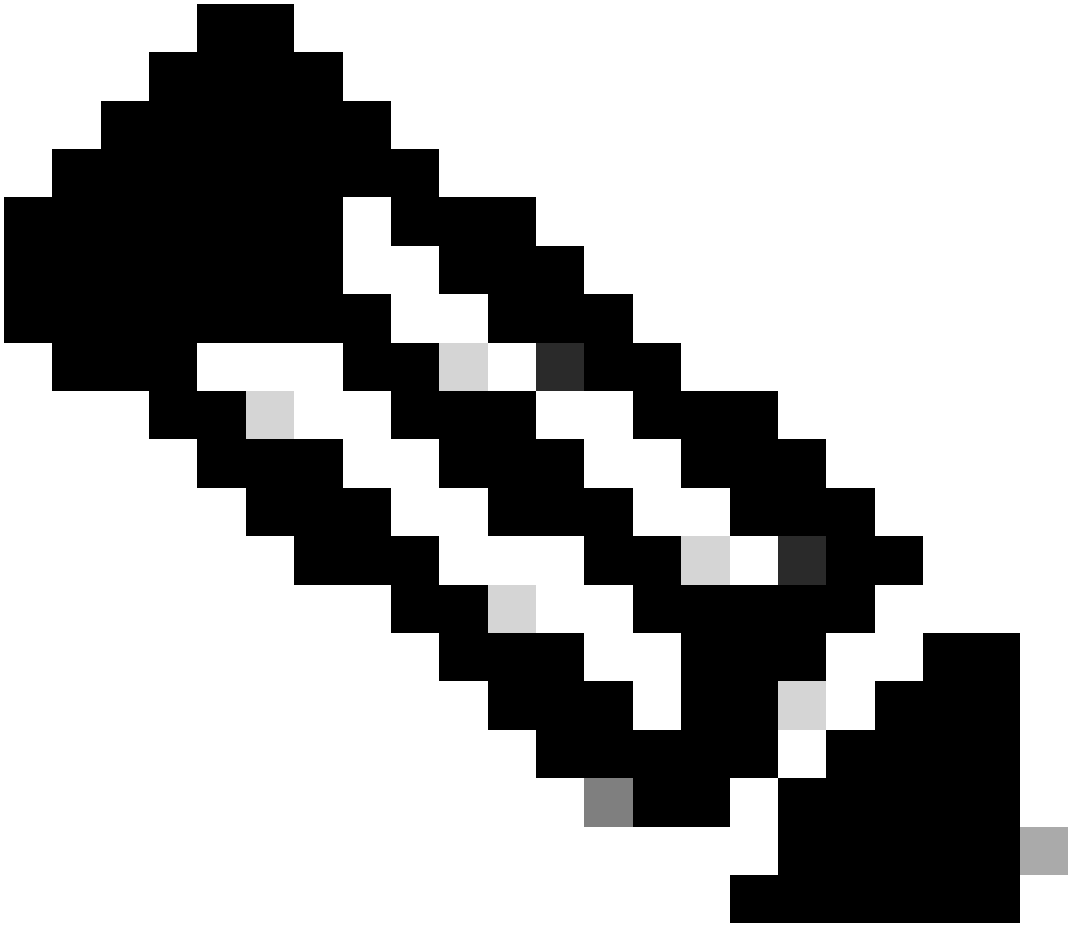
可以使用app-hosting move appid guestshell log到bootflash:folder_name来旋转文件。这可用于旋转到目标目录，但也会旋转到traceloguestshell目录。

<#root>

```
Switch#
```

app-hosting move appid guestshell log to bootflash:

```
Successfully moved tracelog to flash:/ioxapploguestshell/iox_R0-0_R0-0.14195_0.20250311023831.bin.gz
```



注意：IOX guestshell.log始终是/tmp/rp/trace·下的活动tracelog文件。当此文件达到1MB时，会自动将其旋转为带有时间戳的bootflash:traceloguestshell/，并启动一个新文件。

相关信息

- Cisco Catalyst 9000系列交换机上的应用托管白皮书
- Cisco Catalyst 9300系列交换机硬件安装指南
- 可编程性配置指南，Cisco IOS XE 17.9.x
- Catalyst 9300 Stackwise系统架构白皮书
- Cisco Bug ID CSCwi63075 - 在升级到17.08.01或更高版本后，如果启用FIPS模式，则无法通过CLI访问Guestshell
- 技术支持和文档 - Cisco Systems

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。