

验证路由PON部署24.1.2

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[验证步骤- XR路由器](#)

[配置检查](#)

[Linux网络](#)

[物理接口和子接口配置](#)

[PON控制器配置](#)

[验证](#)

[JSON文件](#)

[连接检查](#)

[容器状态](#)

[日期和时间](#)

[跟踪消息](#)

[验证步骤- Linux VM](#)

[验证](#)

[status.sh脚本](#)

[Netplan](#)

[IP 连通性](#)

[MongoDB配置文件](#)

[系统级日志位置](#)

[参考文档](#)

简介

本文档介绍在虚拟机(VM)和XR路由器上验证思科路由PON (被动光纤网络) 解决方案。

先决条件

要求

思科建议了解以下主题。

- 思科IOS® XR软件
- Linux
- 虚拟机环境

使用的组件

本文档中的信息基于列出的软件和硬件版本：

- NCS-540-28Z4C-SYS-A XR路由器
- 思科IOS® XR软件24.1.2
- 路由PON版本24.1.2
- Ubuntu版本20.04.06 LTS

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。

验证步骤- XR路由器

配置检查

Linux网络

确保用于连接的VRF（虚拟路由和转发）反映在linux网络配置中。在本示例中，已配置VRF Mgmt-intf。此外，请确保为正确的上行链路接口设置source-hint default-route。所列示例中的连接使用接口MgmtEth0/RP0/CPU0/0。

配置示例：

```
linux networking
vrf Mgmt-intf
  address-family ipv4
    default-route software-forwarding
    source-hint default-route interface MgmtEth0/RP0/CPU0/0
```

物理接口和子接口配置

确保OLT（光纤线路终端）可插拔接口插入正确且未关闭配置。此外，确认该子接口带有dot1q标记4090，并已应用于关联的物理接口。

配置示例：

```
interface TenGigE0/0/0/0
description PON OLT
!
interface TenGigE0/0/0/0.4090
encapsulation dot1q 4090
```

命令验证：

<#root>

RP/0/RP0/CPU0:F340.16.19.N540-1#

show ip interface brief

Tue Jul 16 15:08:28.786 UTC

Interface	IP-Address	Status	Protocol	Vrf-Name
-----------	------------	--------	----------	----------

TenGigE0/0/0/0				
----------------	--	--	--	--

unassigned				
------------	--	--	--	--

Up				
----	--	--	--	--

Up				
----	--	--	--	--

default				
---------	--	--	--	--

TenGigE0/0/0/0.4090				
---------------------	--	--	--	--

unassigned				
------------	--	--	--	--

Up				
----	--	--	--	--

Up				
----	--	--	--	--

default				
---------	--	--	--	--

RP/0/RP0/CPU0:F340.16.19.N540-1#

show interface TenGigE0/0/0/0.4090

Wed Jul 17 13:17:07.754 UTC

TenGigE0/0/0/0.4090 is up, line protocol is up

Interface state transitions: 5

Hardware is VLAN sub-interface(s), address is c47e.e0b3.9b04

Internet address is Unknown

MTU 1518 bytes, BW 10000000 Kbit (Max: 10000000 Kbit)

reliability 255/255, txload 0/255, rxload 0/255

Encapsulation 802.1Q Virtual LAN, VLAN Id 4090

, loopback not set

确保在全局配置中启用LLDP。

<#root>

RP/0/RP0/CPU0:F340.16.19.N540-1#

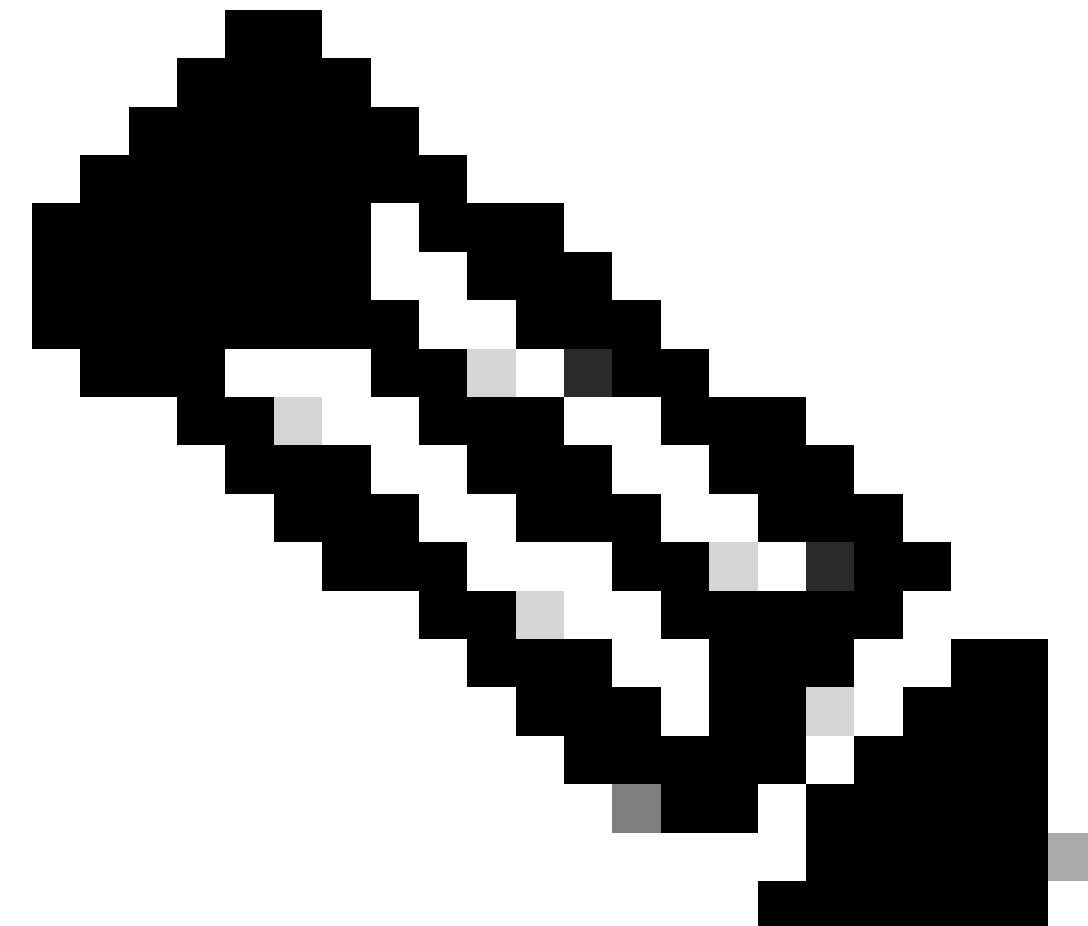
```
show run | include lldp
```

Thu Jul 18 20:16:12.073 UTC

lldp

PON控制器配置

确保已安装xr-pon-ctrl RPM且处于活动状态。如果不是，请确认硬盘上存在NCS540I-iosxr-optional-RPMs-24.2.11.tar（在Linux shell中，路径为/misc/disk1/），并且正确引用包含软件匹配RPM的本地回购。



注意：有关系统范围RPM上的安装和管理信息，请访问以下链接：[思科NCS 540系列路由器 \(IOS XR版本24.1.x、24.2.x\) 的系统设置和软件安装指南](#)

示例：

```
<#root>
```

```
RP/0/RP0/CPU0:F340.16.19.N540-2#
```

```
show install active summary | include xr-pon
```

```
Tue Jul 16 14:59:16.082 UTC
```

```
xr-pon-ctrl 24.1.2v1.0.0-1
```

```
<#root>
```

```
install
```

```
  repository local-repo
```

```
  url file:///
```

```
harddisk:/optional-RPMs-2412
```

确保PON控制器配置了正确关联的文件、文件路径和VRF。

示例：

```
<#root>
```

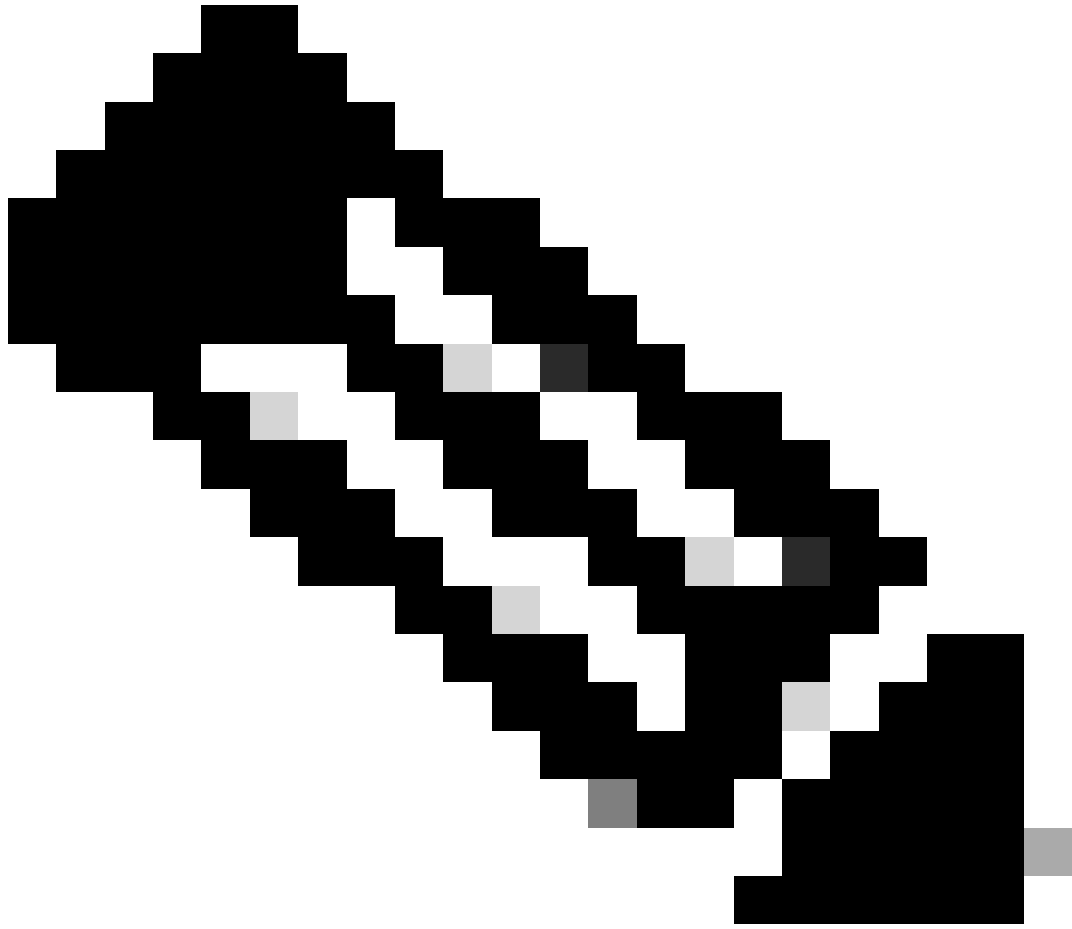
```
pon-ctrl
```

```
  cfg-file
```

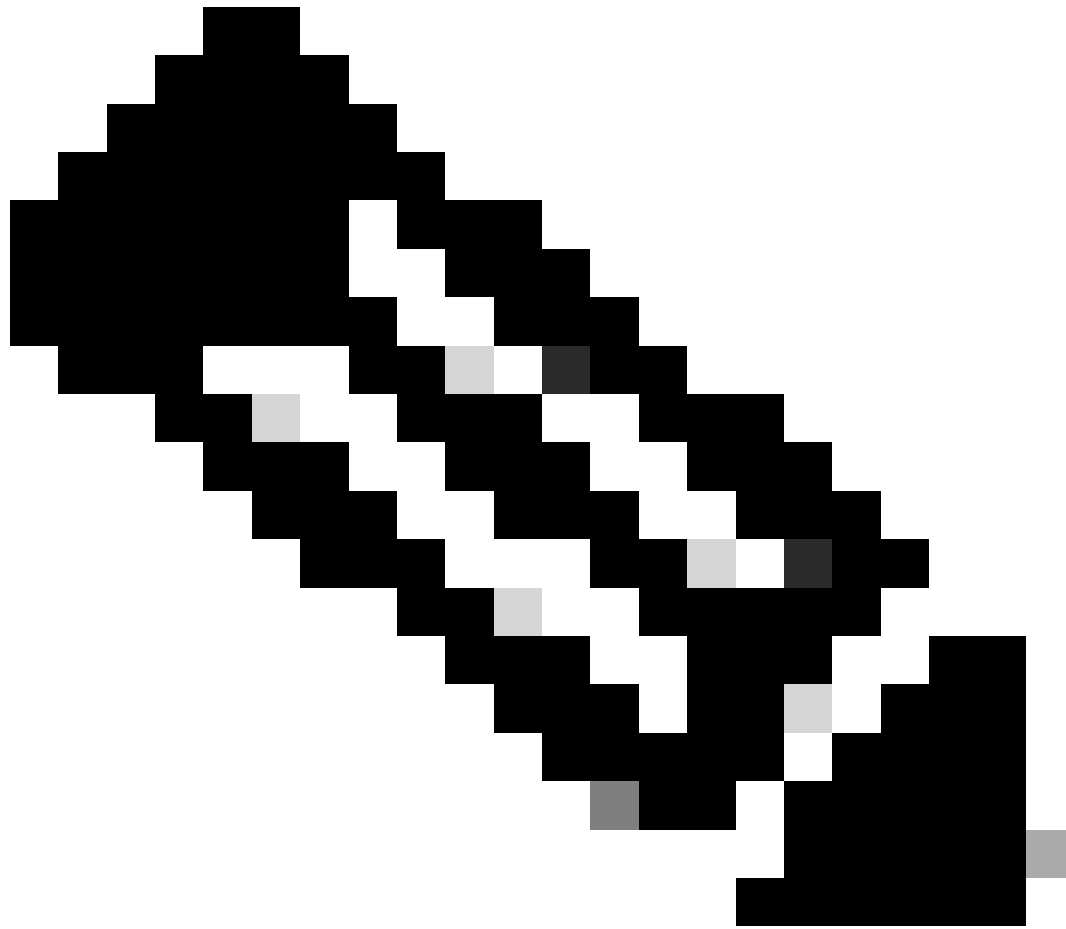
```
harddisk:/PonCntlInit.json vrf Mgmt-intf
```

验证

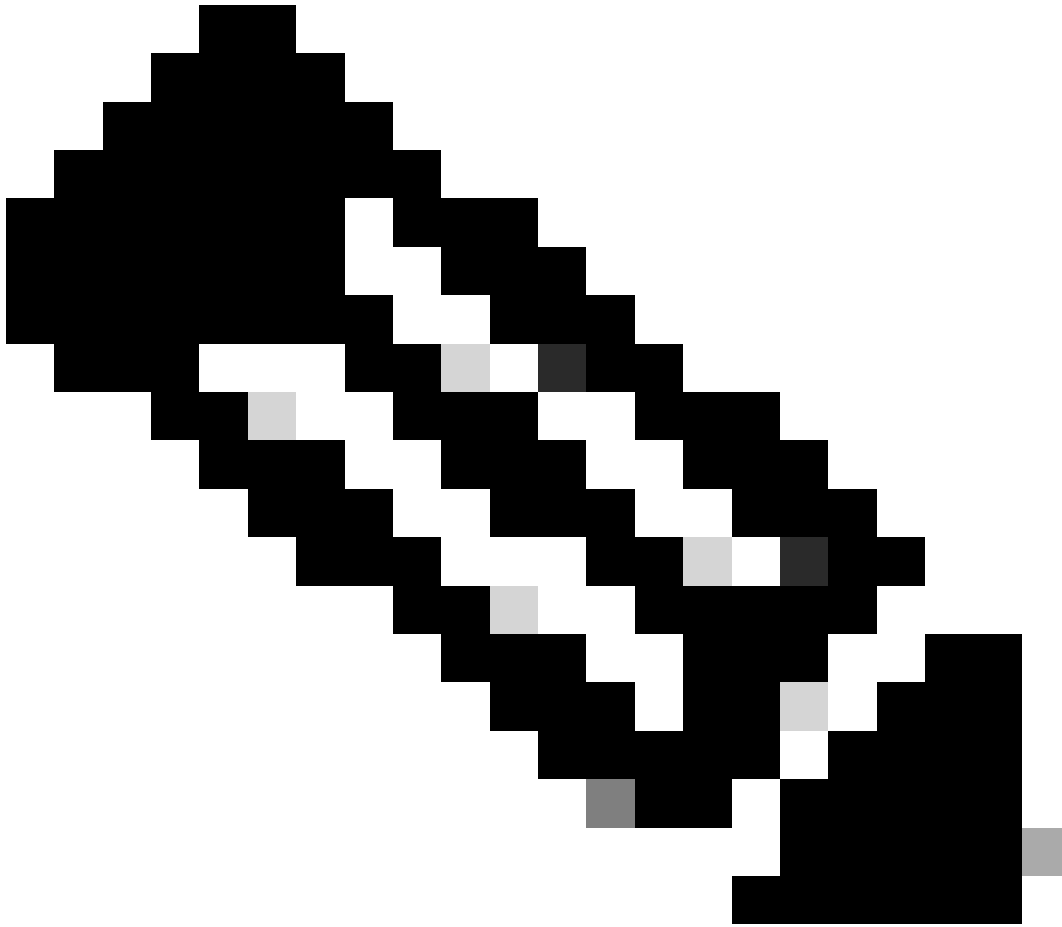
JSON文件



注意：PonCntlInit.json文件示例包含在虚拟机上安装路由PON管理器软件中。



注意：如果只安装一个PON Manager虚拟机，则MongoDB IP和VM IP相同。



注意：所列示例不使用TLS。如果您使用的是TLS，请确保为您的安装正确设置了用户名和口令。

确保在host：部分中设置MongoDB的IP以匹配PON控制器连接的设备。此外，确认配置的端口与VM中mongod.conf文件的端口相匹配。

示例：

```
<#root>
```

```
{  
  "CNTL": {  
    "Auth": false,  
    "CFG Version": "R4.0.0",
```

```
"DHCPv4": true, <- DHCP set to true for CPE devices, Default is false.
```

```
"DHCPv6":
```



```
true

,
<- DHCP set to true for CPE devices, Default is false.

  "PPPoE": false,
  "UMT interface": "tibitvirt",

"
Maximum CPEs Allowed": 0,
  "Maximum CPE Time": 0
},
"DEBUG": {},
"JSON": {
  "databaseDir": "/opt/tibit/poncntl/database/",
  "defaultDir": "/opt/tibit/poncntl/database/"
},
"Local Copy": {
  "CNTL-STATE": false,
  "OLT-STATE": false,
  "ONU-STATE": false
},
"Logging": {
  "Directory": "/var/log/tibit",
  "FileCount": 3,
  "FileSize": 10240000,
  "Tracebacks": false,
  "Timestamp": false,
  "Facility" : "user"
},
"MongoDB": {
  "auth_db": "tibit_users",
  "auth_enable": false,
  "ca_cert_path": "/etc/cisco/ca.pem",
  "compression": false,
  "write_concern": "default",

"host": "10.122.140.232", <- MongoDB IP

  "name": "tibit_pon_controller",

"password": "", <- Left Empty - Not using TLS

  "port": "27017", <- MongoDB TCP Port

  "tls_enable": false, <- Set to False to leave TLS disabled

"username": "", <- Left Empty - Not using TLS

  "dns_srv": false,
  "db_uri": "",
  "replica_set_enable": false,
```

```
    "validate_cfg": true
  },
  "databaseType": "MongoDB",
  "interface": "veth_pon_g1b"
}
```

连接检查

从XR路由器，对MongoDB/VM托管路由PON管理器执行ping操作。如果使用VRF，则从VRF获取源。

示例：

```
<#root>
```

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

```
ping vrf Mgmt-intf 10.122.140.232
```

```
Tue Jul 16 15:09:52.780 UTC
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.122.140.232 timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
```

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

容器状态

PON控制器在XR路由器上的坞站容器上运行。通过登录XR路由器中的linux shell检查容器状态，然后运行命令docker ps。这将显示当前开启和活动容器（如果有）。

示例：

```
<#root>
```

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

```
run
```

```
Tue Jul 16 15:14:26.059 UTC
```

```
[node0_RP0_CPU0:~]$docker ps
```

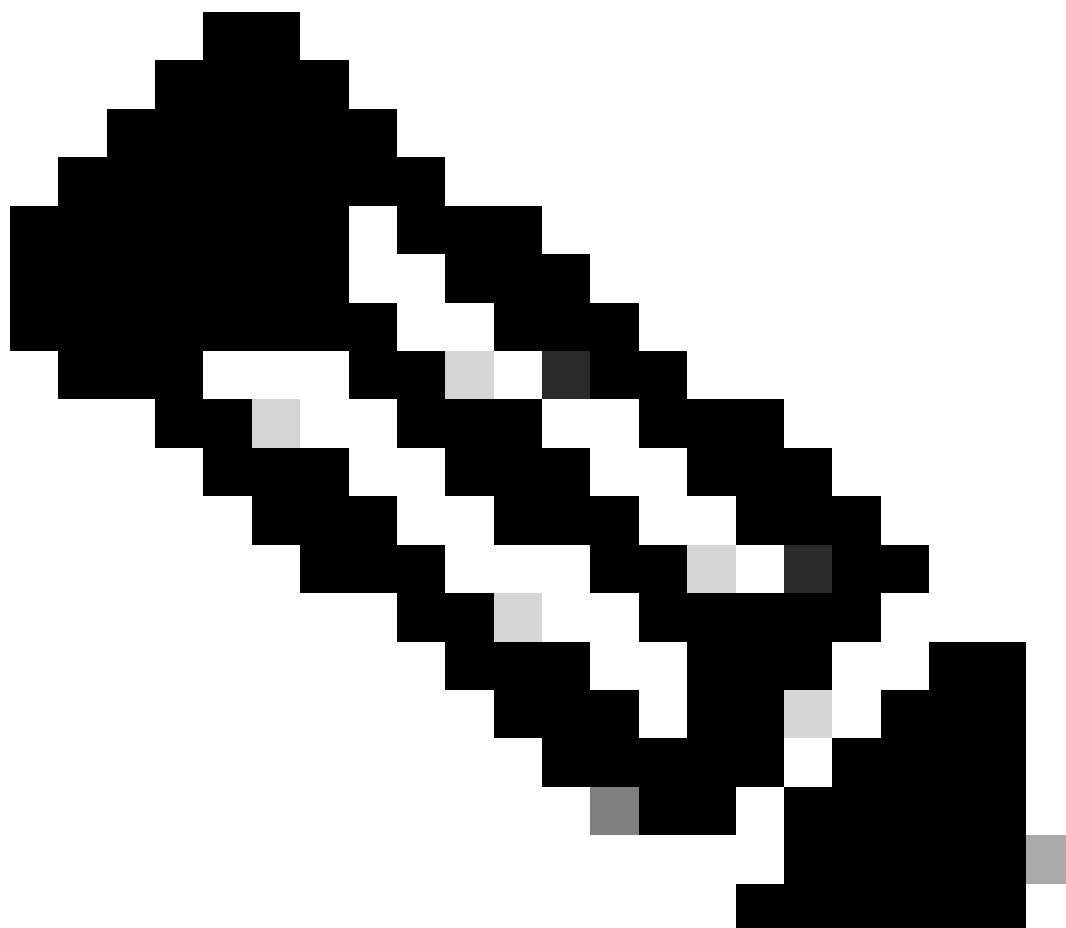
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
2e700f202ee3
```

	tibit-poncntl.xr:R4.0.0	"/usr/bin/supervisor..."	3 days ago	Up	3 days	
--	-------------------------	--------------------------	------------	----	--------	--

```
pon_ctlr
```

如果docker容器未运行，请检查JSON文件的内容和文件结构。检查docker容器的日志中是否存在任何活动错误。日志示例显示一个ONU注册到控制器。这还会打印有关容器和OLT的任何Docker级别错误。此外，可以通过运行简单的show logging来检查错误消息来获得指导。



注意：用法—follow在docker中显示最新的日志内容。

示例：

```
<#root>
```

```
[node0_RP0_CPU0:~]$
```

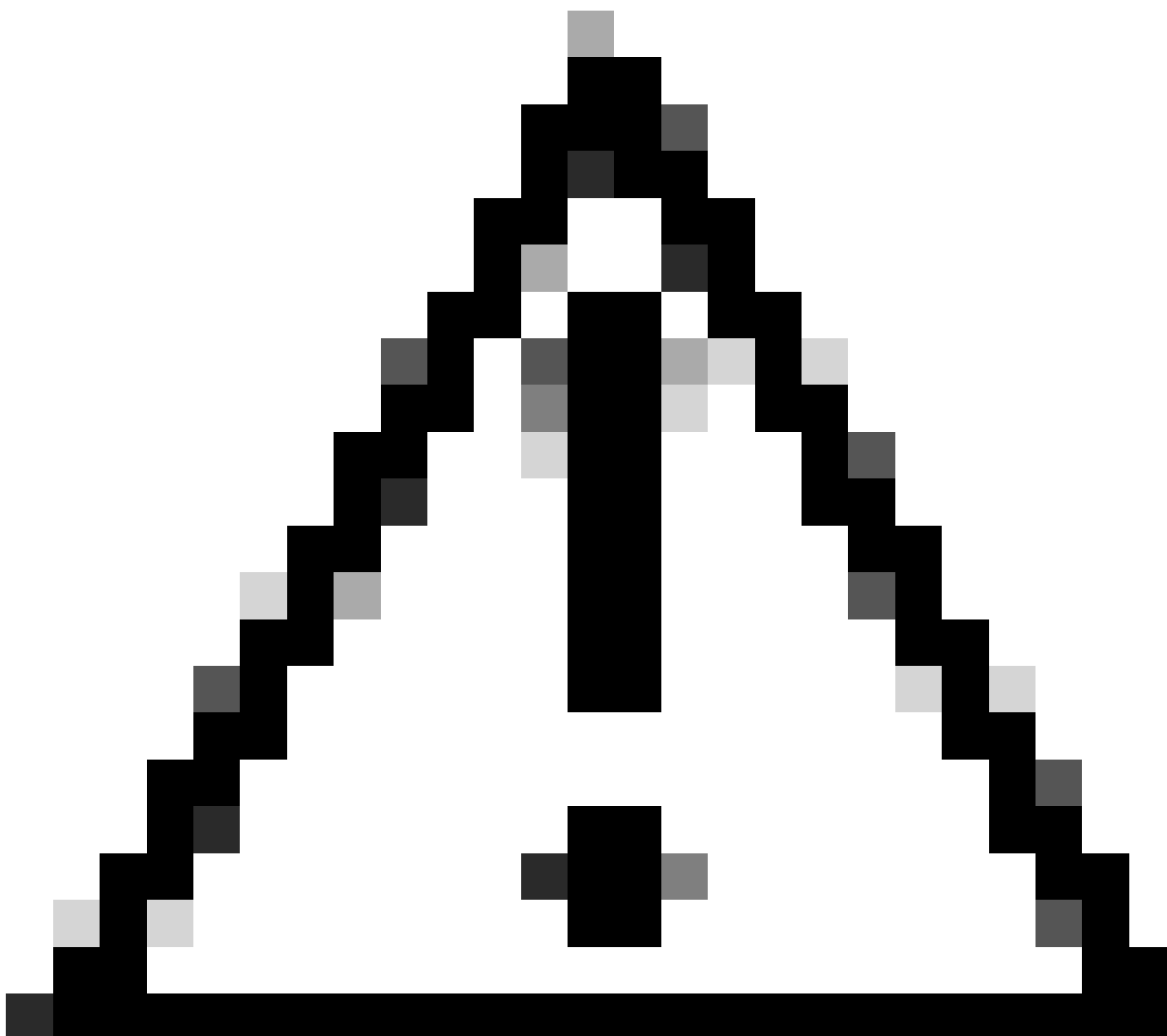
```
docker logs pon_ctr
```

```
2024-07-16 15:05:11.630 PonCnt1 System Status
```

```
{  
"e0:9b:27:36:aa:76": {  
"OLT State": "Primary",  
"ONU Active Count": 1,  
"ONUs": {  
"CIGG2410503f": "Registered"
```

日期和时间

确保XR路由器上的时间和日期与托管路由PON管理器的VM匹配。如果可能，请使用相同的NTP服务器以获得最佳准确性。



注意：VM和XR路由器之间的NTP不同步直接影响路由PON管理器中的OLT可视性。

示例：

```
<#root>
```

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

```
show clock
```

```
Tue Jul 16 15:25:03.781 UTC
```

```
15:25:03.827 UTC Tue Jul 16 2024
```

配置示例：

```
ntp
```

```
server vrf Mgmt-intf 172.18.108.14 source MgmtEth0/RP0/CPU0/0
```

```
server vrf Mgmt-intf 172.18.108.15 prefer source MgmtEth0/RP0/CPU0/0
```

跟踪消息

PON进程通过ltrace生成其他日志记录。检查这些日志，查找与此进程相关的任何错误。

示例：

```
<#root>
```

```
RP/0/RP0/CPU0:F340.16.19.N540-1#
```

```
show pon-ctrl ltrace all reverse location all
```

```
Wed Jul 17 13:25:43.747 UTC
```

```
670 wrapping entries (4224 possible, 896 allocated, 0 filtered, 670 total)
```

```
Jul 10 19:17:55.066 pon_ctrl/event 0/RP0/CPU0 t6986 pon_ctrl_config_sysdb.c:117:Successfully connected
```

```
Jul 10 19:17:55.039 pon_ctrl/event 0/RP0/CPU0 t6986 pon_ctrl_main.c:372:Successfully registered with i
```

```
Jul 10 19:17:55.006 pon_ctrl/event 0/RP0/CPU0 t7082 pon_ctrl_utls.c:353:IP LINK: ip link delete veth_po
```

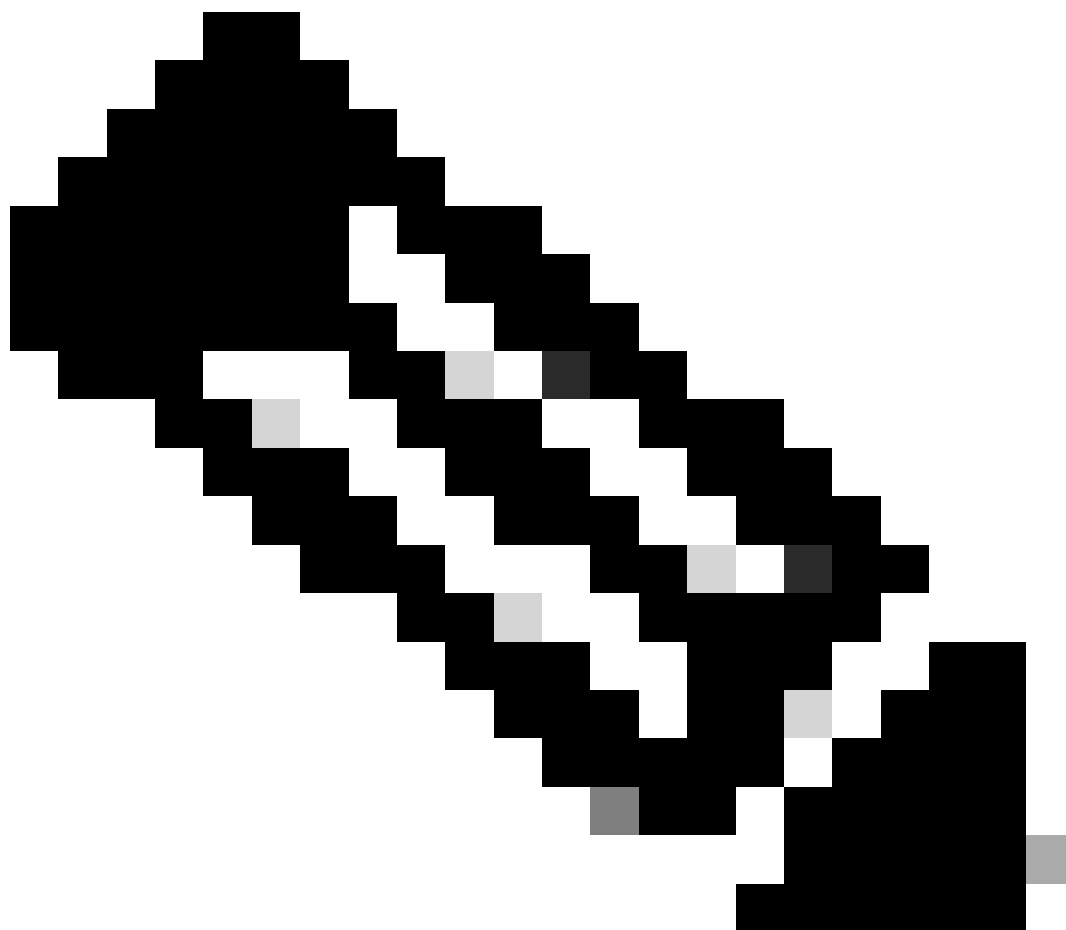
验证步骤- Linux VM

验证

status.sh脚本

在Routed PON Manager安装目录中，有一个shell脚本(status.sh)，用于显示每个关联进程的当前状态。以提升的权限运行此脚本，以验证列出的每项服务是否都已启动并正在运行。如果其中一个

服务未运行，首先检查执行安装时运行的安装脚本，并确保按照安装指南设置正确的参数。



注意：Cisco Routed PON Manager安装指南位于以下链接：[Cisco Routed PON Manager安装指南](#)

```
mongod.service
apache2.service
netconf.service
netopeer2-server.service
```

示例：

```
<#root>
```

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CC0/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

```
sudo ./status.sh
```

[sudo] password for rpon:
MCMS Component Versions:
PON Manager: R4.0.0
PON NETCONF: R4.0.0
PON Controller: Not Installed

•

mongod.service

- MongoDB Database Server
Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-06-27 08:46:25 EDT; 2 weeks 5 days ago

Main PID: 52484 (mongod)
Memory: 1.5G
CGroup: /system.slice/mongod.service
└─52484 /usr/bin/mongod --config /etc/mongod.conf

•

apache2.service

- The Apache HTTP Server
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
Active: active (running) since Fri 2024-07-12 06:33:30 EDT; 4 days ago

Process: 103015 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
Main PID: 96525 (apache2)
Tasks: 123 (limit: 9403)
Memory: 27.0M
CGroup: /system.slice/apache2.service
├─96525 /usr/sbin/apache2 -k start
├─103029 /usr/sbin/apache2 -k start
├─103030 /usr/sbin/apache2 -k start
└─103031 /usr/sbin/apache2 -k start

•

tibit-netconf.service

- Tibit Communications, Inc. NetCONF Server
Loaded: loaded (/lib/systemd/system/tibit-netconf.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-06-27 08:47:44 EDT; 2 weeks 5 days ago

Main PID: 60768 (tibit-netconf)
Tasks: 17 (limit: 9403)
Memory: 60.7M
CGroup: /system.slice/tibit-netconf.service
├─60768 /opt/tibit/netconf/bin/tibit-netconf
└─60786 /opt/tibit/netconf/bin/tibit-netconf

•

tibit-netopeer2-server.service

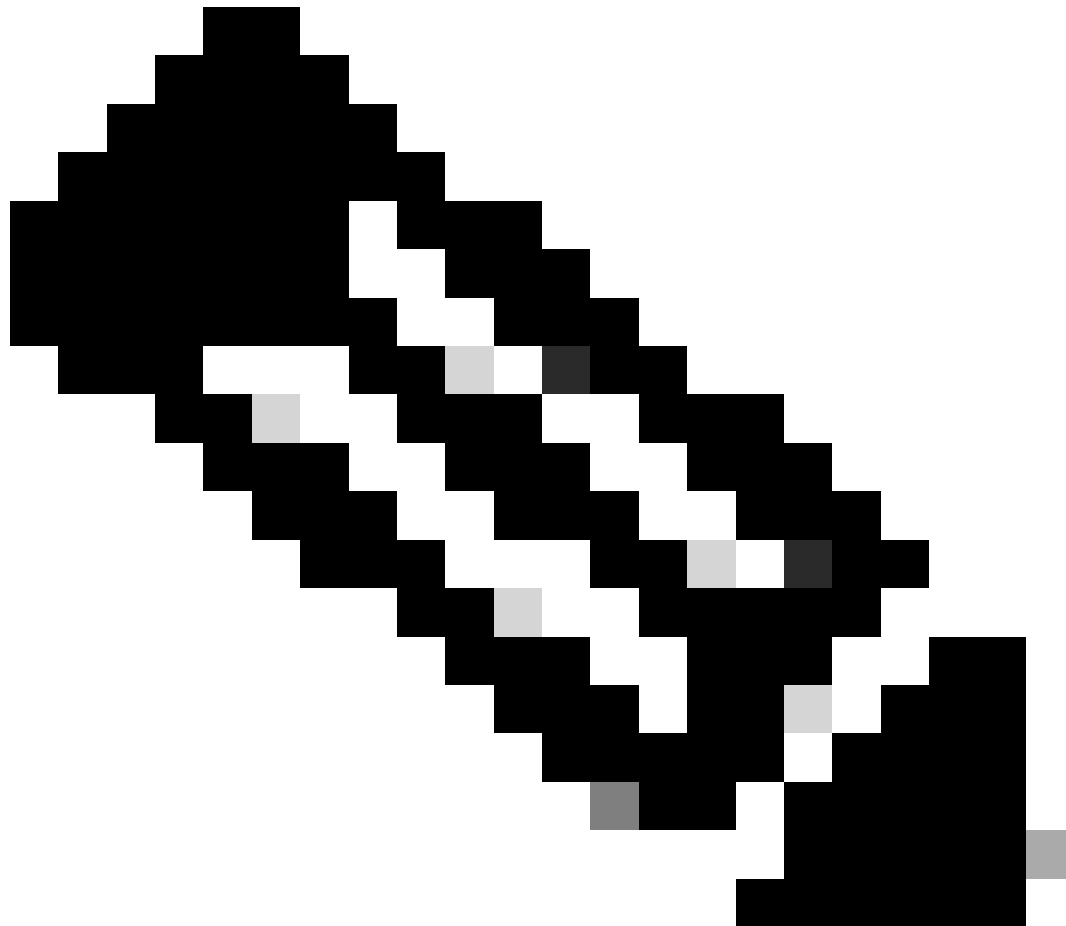
- Tibit Communications, Inc. Netopeer2 Server
Loaded: loaded (/lib/systemd/system/tibit-netopeer2-server.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2024-06-27 08:47:44 EDT; 2 weeks 5 days ago

Main PID: 60772 (netopeer2-serve)
Tasks: 7 (limit: 9403)
Memory: 6.0M
CGroup: /system.slice/tibit-netopeer2-server.service
└─60772 /opt/tibit/netconf/bin/netopeer2-server -v 1 -t 55

Netplan

验证Netplan并确保IP信息有效、VM网络接口名称正确、创建和分配VLAN ID 4090以及它使用有效的Netplan YAML树结构。



注意：netplan YAML文件位于/etc/netplan/中。

示例：


```
<#root>
```

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CC0/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

```
cat /etc/netplan/01-network-manager-all.yaml
```

```
network:
```

```
  version: 2
```

```
  Renderer: Network Manager
```

```
  ethernets:
```

```
    ens192: <- VM Network Adapter
```

```
      dhcp4: no <- No DHCP as the IP is set statically
```

```
      dhcp6: no
```

```
      addresses: [10.122.140.232/28] <- IP of the VM Network adapter
```

```
      gateway4: 10.122.140.225 <- GW of the IP Network
```

```
      nameservers:
```

```
        addresses: [172.18.108.43,172.18.108.34] <- Network DNS
```

```
    vlans:
```

```
      vlan.4090:
```

```
        id: 4090
```

```
    link: ens192 <- VM Network adapter
```

```
      dhcp4: no
```

```
      dhcp6: no
```

验证VM的IP配置以及配置的网络适配器是否与netplan YAML文件中列出的相匹配。

注意：sudo netplan的用法—debug apply在测试应用程序之前的netplan时非常有用。

示例：

```
<#root>
```

```
rpon@rpon-mgr:~$ ifconfig
```

```
ens192
```

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

```
inet 10.122.140.232
```

```
netmask 255.255.255.240 broadcast 10.122.140.239
```

```
inet6 fe80::df4d:8d4d:4836:82aa prefixlen 64 scopeid 0x20<link>
```

```
ether 00:50:56:84:3f:8f txqueuelen 1000 (Ethernet)
```

```
RX packets 68933231 bytes 21671670389 (21.6 GB)
```

```
RX errors 0 dropped 129 overruns 0 frame 0
```

```
TX packets 36820200 bytes 71545432788 (71.5 GB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
-- snipped for brevity --
```

```
vlan.4090
```

```
: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet6 fe80::250:56ff:fe84:3f8f prefixlen 64 scopeid 0x20<link>
  ether 00:50:56:84:3f:8f txqueuelen 1000 (Ethernet)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 1044 bytes 140547 (140.5 KB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

IP 连通性

通过ping验证与托管PON控制器的XR路由器的IP连接。

示例：

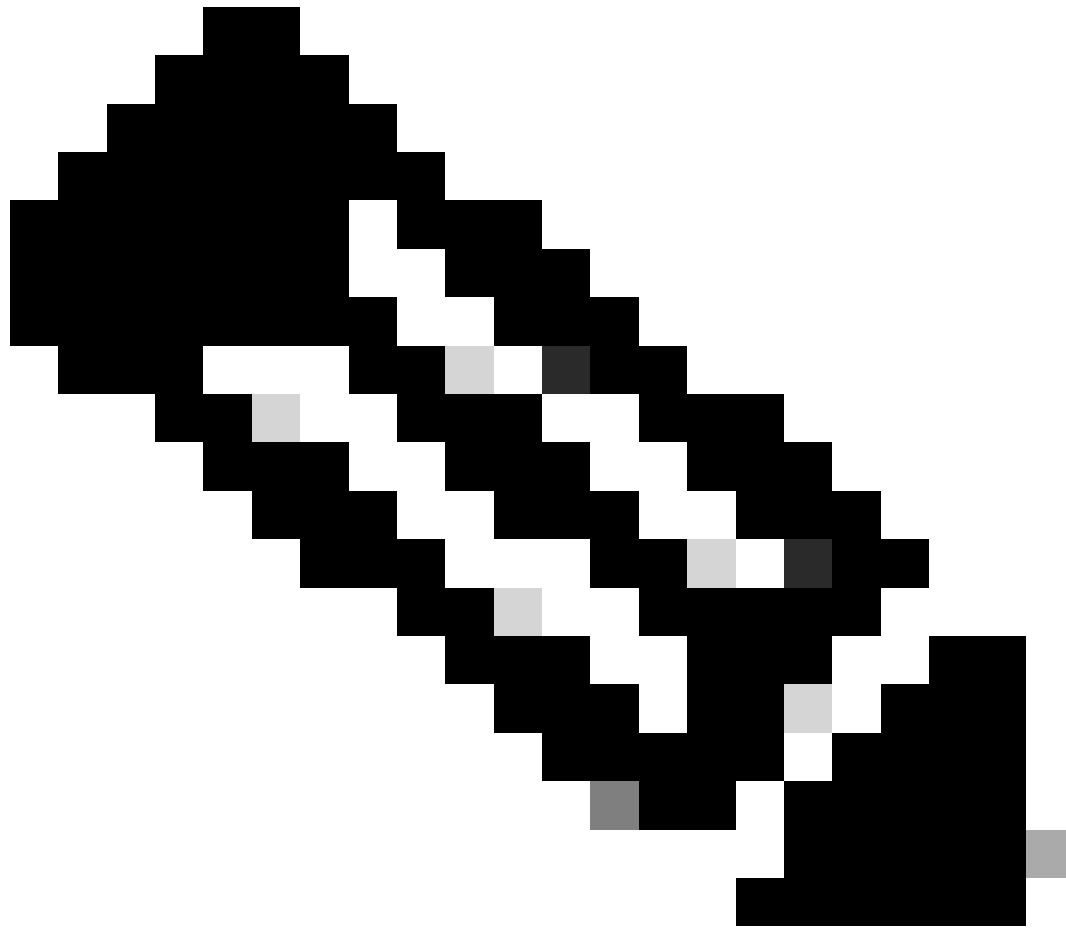
```
<#root>
```

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CC0/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

```
ping 10.122.140.226
```

```
PING 10.122.140.226 (10.122.140.226) 56(84) bytes of data.
64 bytes from 10.122.140.226: icmp_seq=1 ttl=255 time=1.01 ms
64 bytes from 10.122.140.226: icmp_seq=2 ttl=255 time=1.03 ms
64 bytes from 10.122.140.226: icmp_seq=3 ttl=255 time=1.13 ms
^C
--- 10.122.140.226 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.009/1.054/1.128/0.052 ms
```

验证MongoDB TCP端口27017是否已打开。如果为MongoDB使用非标准端口，请通过netstat -tunl验证该端口是否打开/侦听。



注意：标准MongoDB TCP端口为27017。

注意：步骤4中列出的配置文件还设置了MongoDB要使用的TCP端口配置。

示例：

```
<#root>
```

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CC0/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

```
netstat -tunl
```

```
Active Internet connections (only servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	127.0.0.53:53	0.0.0.0:*	LISTEN
tcp	0	0			

```
127.0.0.1:27017
```

```
0.0.0.0:*
```

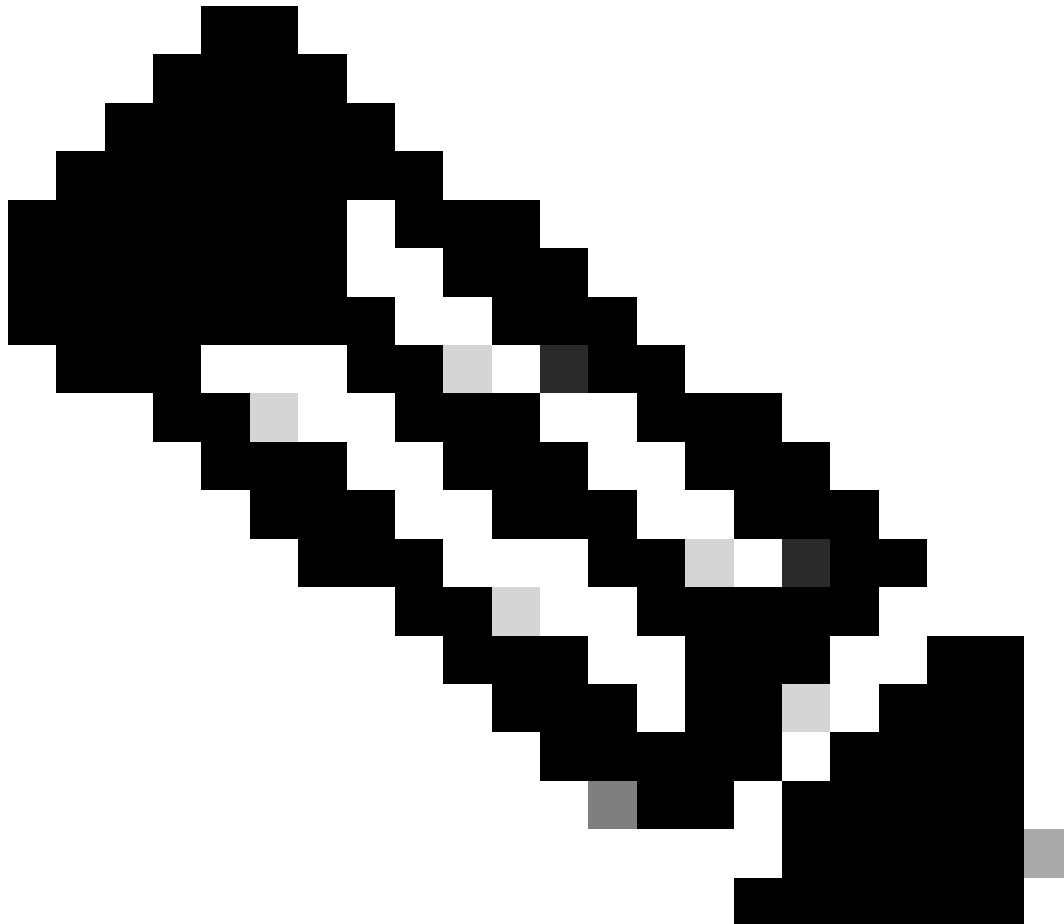
```
tcp 0 0
```

```
10.122.140.232:27017
```

0.0.0.0:* LISTEN

MongoDB配置文件

验证mongod.conf文件是否正确并且bindIP：下列出的IP是否正确。



注意：MongoDB配置文件位于/etc/mongod.conf

示例：

```
<#root>
```

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CC0/R4.0.0-Cisco-UB2004-sign/R4.0.0-Cisco-UB2004$
```

```
cat /etc/mongod.conf
```

```
# mongod.conf

storage:
dbPath: /var/lib/mongodb
journal:
enabled: true

systemLog:
destination: file
logAppend: true
path: /var/log/mongodb/mongod.log
logRotate: reopen

# network interfaces
net:

port: 27017

bindIp: 127.0.0.1,10.122.140.232

processManagement:
pidFilePath: /var/run/mongodb/mongod.pid
timezoneInfo: /usr/share/zoneinfo

replication:
replSetName: "rs0"

-- snipped for brevity --
```

系统级日志位置

每个服务的系统级日志在linux内管理。这些日志存储在/var/log目录中，特别是这些树下。

MongoDB日志 : /var/log/mongod/mongod.log
Apache日志 : /var/log/apache2/<filename>.log
虚拟机系统日志 : /var/log/syslog

参考文档

- [思科支持和下载页面](#)
- [思科路由PON解决方案页面](#)
- [思科路由PON安装指南](#)
- [思科路由PON部署指南](#)
- [思科路由PON、思科IOS XR版本24.1.1和24.1.2的发行版本注释](#)

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）。