

ルーテッドPON導入の検証24.1.2

内容

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[検証手順：XRルータ](#)

[設定の確認](#)

[Linuxネットワーク](#)

[物理インターフェイスとサブインターフェイスの設定](#)

[PONコントローラの設定](#)

[検証](#)

[JSONファイル](#)

[接続の確認](#)

[コンテナステータス](#)

[日付と時刻](#)

[トレースメッセージ](#)

[検証手順：Linux VM](#)

[検証](#)

[status.shスクリプト](#)

[ネットプラン](#)

[IP接続](#)

[MongoDB設定ファイル](#)

[システムレベルログの場所](#)

[参考資料](#)

はじめに

このドキュメントでは、仮想マシン(VM)およびXRルータでのCisco Routed PON(Passive Optical Network)ソリューションの検証について説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます。

- Cisco IOS® XRソフトウェア
- Linux
- 仮想マシン環境

使用するコンポーネント

このドキュメントの情報は、次に示すソフトウェアとハードウェアのバージョンに基づくものです。

- NCS-540-28Z4C-SYS-A XRルータ
- Cisco IOS® XRソフトウェア24.1.2
- ルーテッドPONバージョン24.1.2
- Ubuntuバージョン20.04.06 LTS

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

検証手順：XRルータ

設定の確認

Linuxネットワークング

接続に使用するVirtual Routing and Forwarding(VRF)がLinuxネットワーク設定に反映されていることを確認します。この例では、VRF Mgmt-intfが設定されています。また、source-hint default-routeが正しいアップリンクインターフェイスに設定されていることを確認します。上記の例の接続では、インターフェイスMgmtEth0/RP0/CPU0/0を使用しています。

設定例：

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

物理インターフェイスとサブインターフェイスの設定

OLT(Optical Line Terminal)プラグインが挿入されているインターフェイスが正しく、設定でシャットダウンされていないことを確認します。さらに、サブインターフェイスが4090でタグ付けされたdot1qであり、関連付けられた物理インターフェイスに適用されていることを確認します。

設定例：

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

コマンドの検証：

<#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

LLDPがグローバルコンフィギュレーションで有効になっていることを確認します。

```
<#root>
```

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

```
show run | include lldp
```

```
Thu Jul 18 20:16:12.073 UTC
```

```
lldp
```

PONコントローラの設定

xr-pon-ctrl RPMがインストールされていて、アクティブであることを確認します。そうでない場合は、NCS540i-iosxr-optional-RPMs-24.2.11.tarがハードディスク (Linuxシェルでは /misc/disk1/) に存在し、ソフトウェアが一致したRPMを含むlocal-repoが正しく参照されていることを確認します。

注：システム全体のRPMのインストールと管理の詳細については、[Cisco NCS 540シリーズルータのシステムセットアップおよびソフトウェアインストールガイド、IOS XRリリース24.1.x、24.2.xを参照してください。](#)

以下に例を挙げます。

```
<#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-ctrlr 24.1.2v1.0.0-1
```

```
<#root>
```

```
install  
  repository local-repo  
  url file:///
```

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

PONコントローラが正しく関連付けられたファイル、ファイルパス、およびVRFで設定されていることを確認します。

以下に例を挙げます。

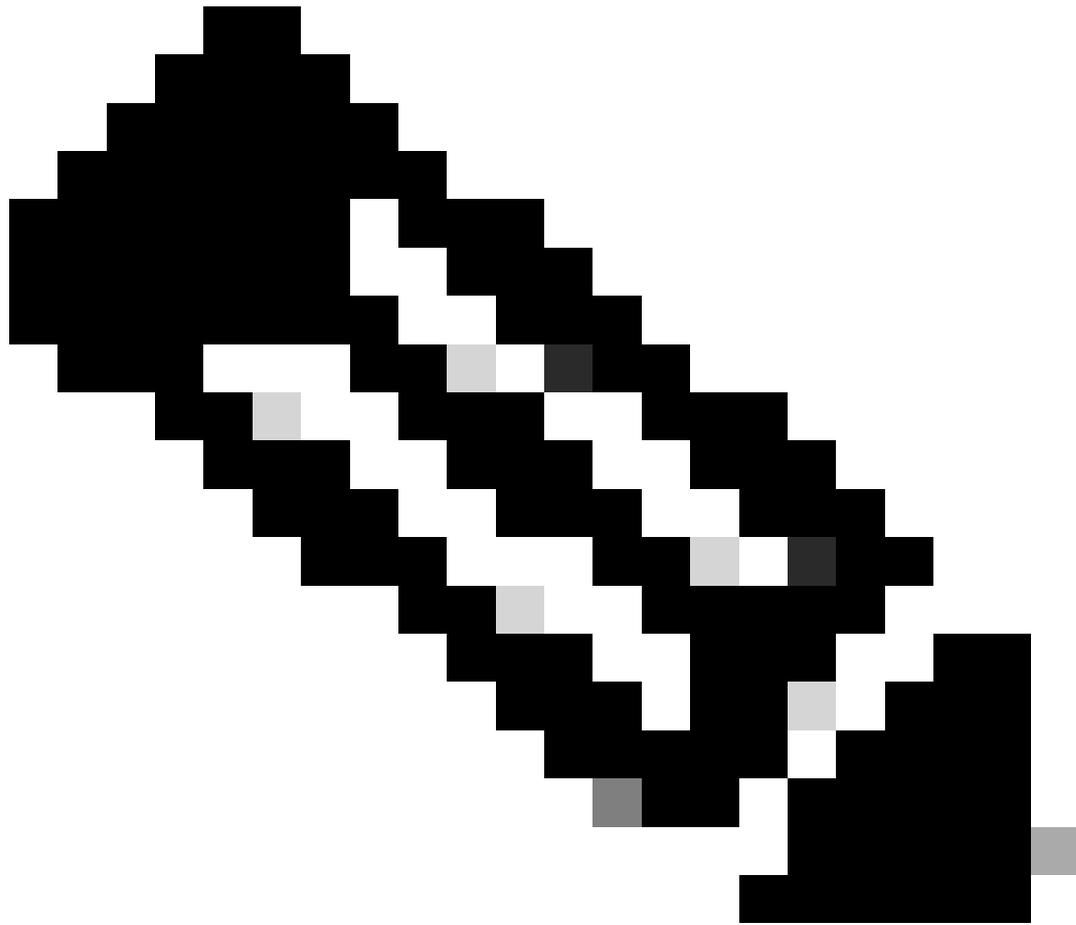
```
<#root>
```

```
pon-ctrl  
  cfg-file
```

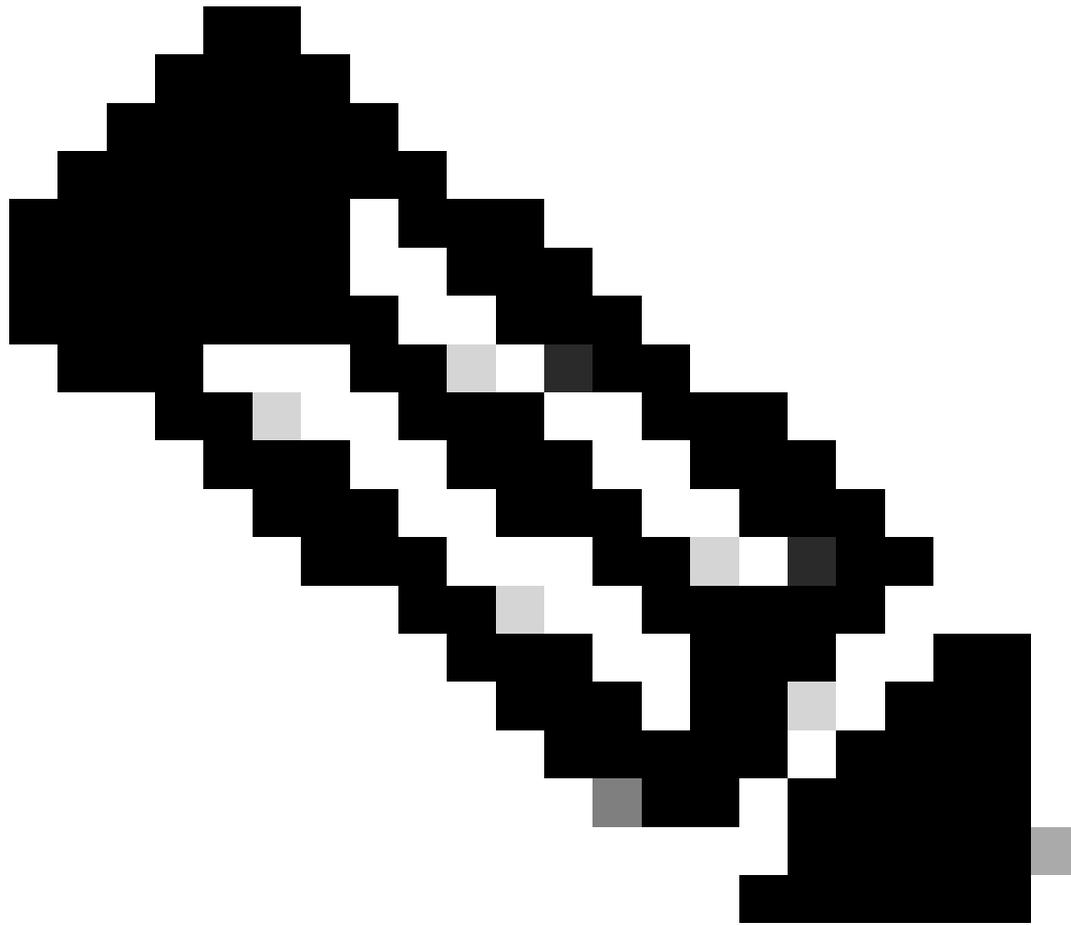
```
harddisk:/PonCntlInit.json vrf Mgmt-intf
```

検証

JSONファイル



注:PonCntlInit.jsonファイルの例は、VMでのRouted PON Manager(RPON)ソフトウェアのインストールに含まれています。



注:PON Managerの単一VMインストールでは、MongoDB IPとVM IPが同じです。

注：上記の例では、TLSを使用していません。TLSを使用している場合は、ユーザ名とパスワードがインストール環境に合わせて正しく設定されていることを確認します。

MongoDBのIPがhost:セクションで、PONコントローラの接続先と一致するように設定されていることを確認します。さらに、設定されたポートがVMのmongod.confファイルのポートと一致することを確認します。

以下に例を挙げます。

```
<#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": "tibatvirt",

"
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"
}
```

接続の確認

XRルータから、Routed PON ManagerをホストするMongoDB/VMにpingを実行します。VRFを使用している場合は、VRFからソースを取得します。

以下に例を挙げます。

```
<#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#
```

コンテナステータス

PONコントローラは、XRルータのDockerコンテナで動作します。XRルータのLinuxシェルにログインしてコンテナのステータスを確認し、コマンドdocker psを実行します。現在アップ状態でアクティブなコンテナが存在する場合は、これが表示されます。

以下に例を挙げます。

```
<#root>
```

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

```
run
```

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

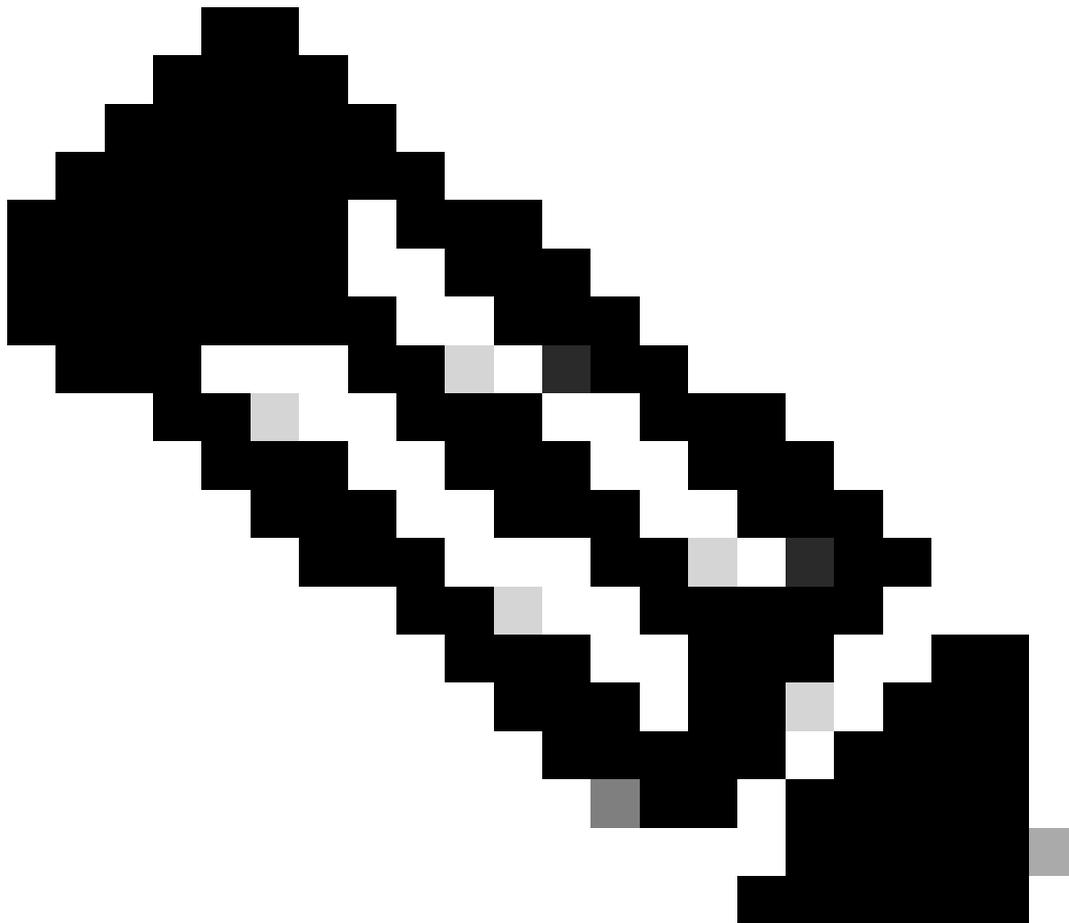
```
[node0_RP0_CPU0:~]$docker ps
```

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

2e700f202ee3						
--------------	--	--	--	--	--	--

pon_ctlr

Dockerコンテナが実行されていない場合は、JSONファイルの内容とファイル構造を確認します。Dockerコンテナのログを調べて、アクティブなエラーがないかどうかを確認します。ログの例は、コントローラに登録するONUを示しています。これにより、コンテナとOLTに関するDockerレベルのエラーも出力されます。さらに、エラーメッセージを確認するための簡単なshow loggingを実行することでも、ガイダンスを得ることができます。



注：—followを使用すると、Docker内の最新のログコンテンツが表示されます。

以下に例を挙げます。

<#root>

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

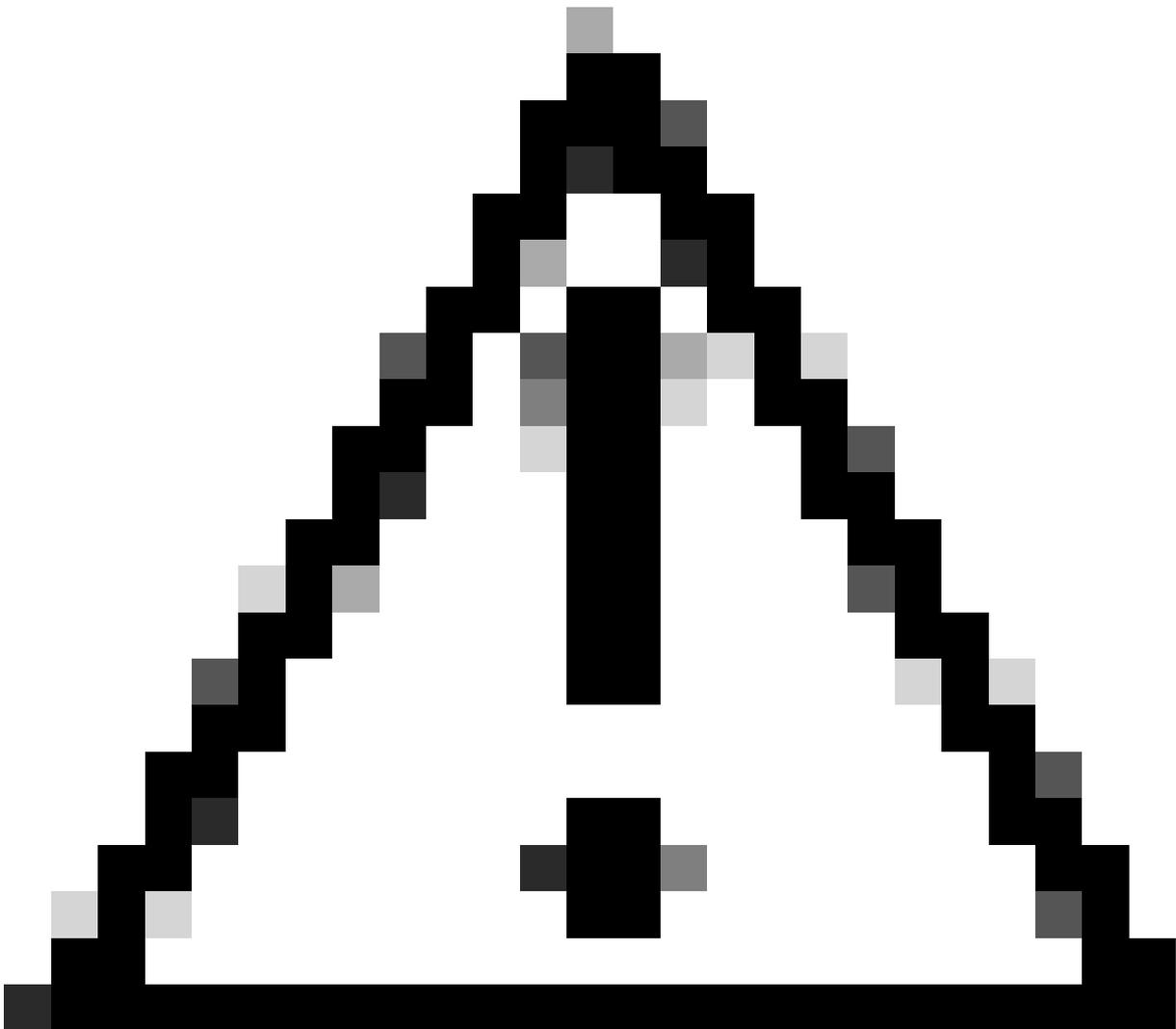
```
docker logs pon_ctr
```

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

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

日付と時刻

XRルータの日時と、Routed PON Manager(RPON)をホストしているVMの日時が一致していることを確認します。可能であれば、最適な精度を得るために同じNTPサーバを使用します。



注意:VMとXRルータ間でNTPが同期されていないと、Routed PON ManagerでのOLTの可視性に直接影響します。

以下に例を挙げます。

```
<#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
```

設定例：

```
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
```

トレースメッセージ

PONプロセスはltraceを通じて追加のロギングを生成します。このプロセスに関連するエラーがないか、これらのログを確認します。

以下に例を挙げます。

```
<#root>
```

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

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

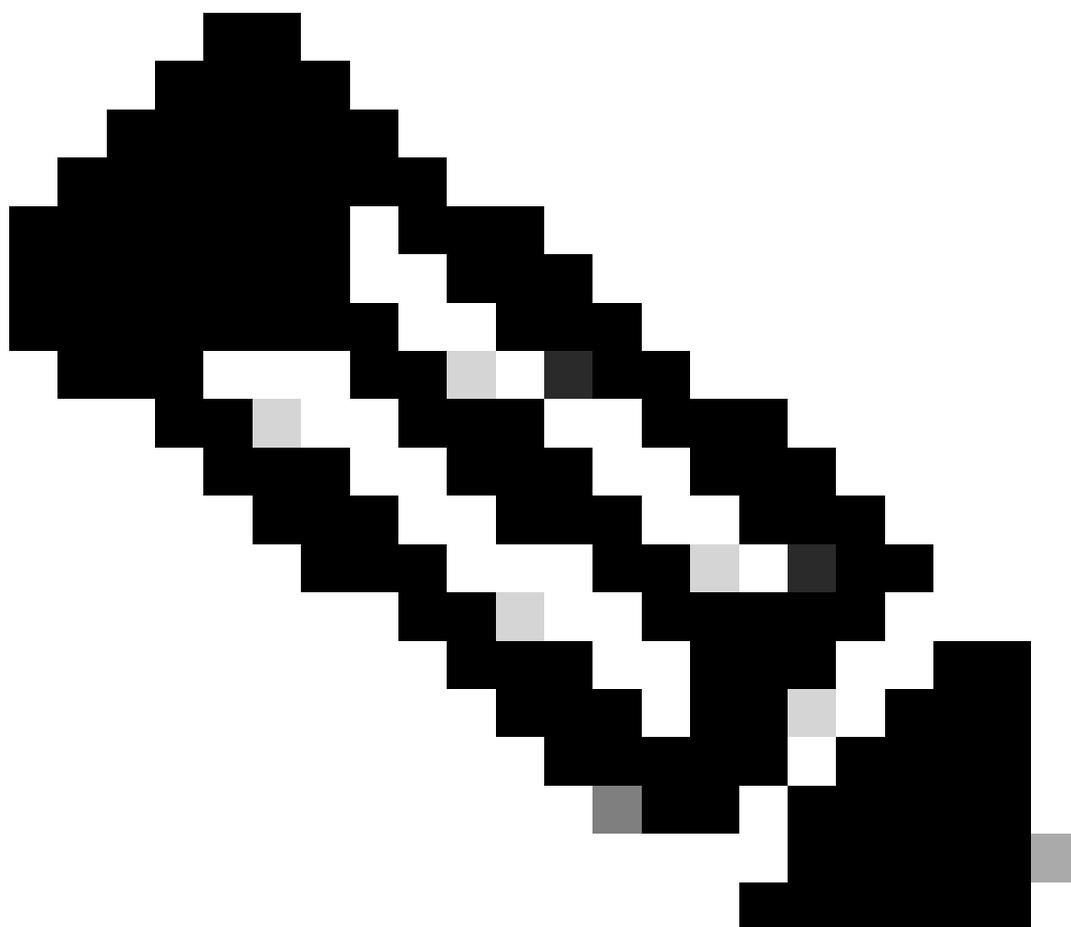
```
Wed Jul 17 13:25:43.747 UTC  
670 wrapping entries (4224 possible, 896 allocated, 0 filtered, 670 total)  
Jul 10 19:17:55.066 pon_ctrl/event 0/RP0/CPU0 t6986 pon_ctrl_config_sysdb.c:117:Successfully connected  
Jul 10 19:17:55.039 pon_ctrl/event 0/RP0/CPU0 t6986 pon_ctrl_main.c:372:Successfully registered with i  
Jul 10 19:17:55.006 pon_ctrl/event 0/RP0/CPU0 t7082 pon_ctrl_utils.c:353:IP LINK: ip link delete veth_po
```

検証手順 : Linux VM

検証

status.shスクリプト

Routed PON Managerのインストールディレクトリ内には、関連付けられた各プロセスの現在のステータスを表示するシェルスクリプト(status.sh)があります。このスクリプトを管理者特権で実行して、リストされている各サービスが稼働していることを確認します。サービスの1つが実行されていない場合は、最初にインストールの実行時に実行されたインストールスクリプトを確認し、インストールガイドに適切な引数が設定されていることを確認します。



注: Cisco Routed PON Managerインストールガイドは、次のリンクから入手できます。
[Cisco Routed PON Manager Installation Guide](#)

mongod.serviceという名前の
apache2サービス
netconfサービス
netopeer2-server.serviceという名前の

以下に例を挙げます。

<#root>

```
rpon@rpon-mgr:~/PON_MANAGER_SIGNED_CCO/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

ネットプラン

Netplanを検証し、IP情報が有効であること、VMネットワークインターフェイス名が正しいこと、VLAN ID 4090が作成されて割り当てられていること、および有効なNetplan YAMLツリー構造を使用していることを確認します。

注: netplan YAMLファイルは/etc/netplan/にあります。

以下に例を挙げます。

```
<#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

vlan:

vlan.4090:

id: 4090

link: ens192 <- VM Network adapter

dhcp4: no

dhcp6: no

VMのIP設定を確認し、設定したネットワークアダプタがnetplan YAMLファイルにリストされている内容と一致することを確認します。

注：sudo netplan —debug applyの使用は、適用する前にネットプランをテストするときに便利です。

以下に例を挙げます。

```
<#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 接続

PONコントローラをホストするXRルータへのIP接続をpingで確認します。

以下に例を挙げます。

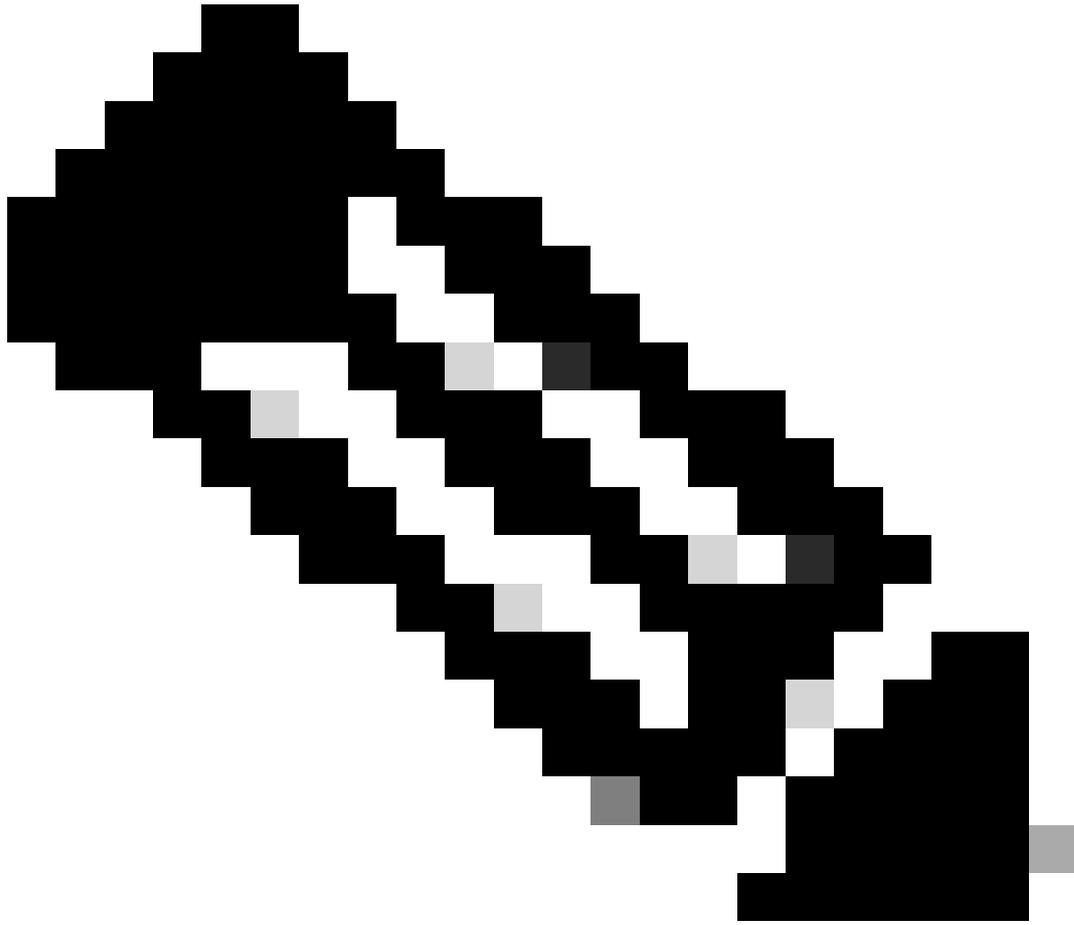
```
<#root>
```

```
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
```

MongoDB TCPポート27017が開いていることを確認します。MongoDBに標準外のポートを使用している場合は、netstat -tunlを使用してポートがオープン/リスニングであることを確認します。



注：標準のMongoDB TCPポートは27017です。

注：ステップ4に記載されているコンフィギュレーションファイルは、MongoDBが使用するTCPポート設定も行います。

以下に例を挙げます。

<#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
```

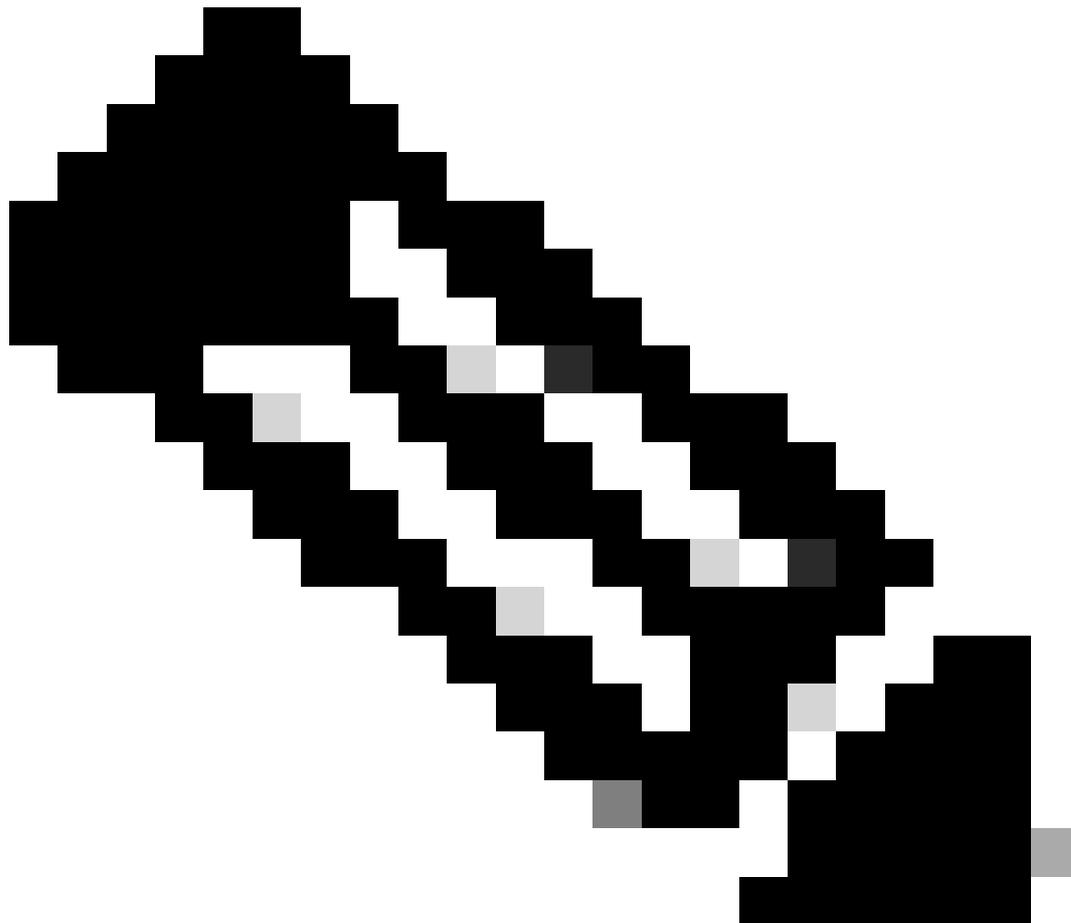
			0.0.0.0:*		LISTEN
tcp	0	0			

```
10.122.140.232:27017
```

```
0.0.0.0:*      LISTEN
```

MongoDB設定ファイル

mongod.confファイルが正確であり、bindIP:に正しいIPがリストされていることを確認します。



注:MongoDBコンフィギュレーションファイルは、/etc/mongod.confにあります

以下に例を挙げます。

```
<#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 --
```

システムレベルログの場所

各サービスのシステムレベルのログはLinux内で管理されます。これらのログは/var/logディレクトリ内、特にこれらのツリーの下に保存されます。

MongoDBログ : /var/log/mongod/mongod.log
Apacheログ : /var/log/apache2/<filename>.log
仮想マシンのsyslog:/var/log/syslog

参考資料

- [シスコのサポートおよびダウンロードページ](#)
- [CiscoルーテッドPONソリューションページ](#)
- [CiscoルーテッドPONインストレーションガイド](#)
- [CiscoルーテッドPON導入ガイド](#)
- [CiscoルーテッドPON、Cisco IOS XRリリース24.1.1および24.1.2のリリースノート](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。