

# 라우팅된 PON 배포 24.1.2 확인

## 목차

---

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[검증 단계 - XR 라우터](#)

[컨피그레이션 확인](#)

[Linux 네트워킹](#)

[물리적 및 하위 인터페이스 컨피그레이션](#)

[PON-Controller 컨피그레이션](#)

[확인](#)

[JSON 파일](#)

[연결 확인](#)

[컨테이너 상태](#)

[날짜 및 시간](#)

[메시지 추적](#)

[검증 단계 - Linux VM](#)

[확인](#)

[status.sh 스크립트](#)

[넷플랜](#)

[IP 연결](#)

[MongoDB 컨피그레이션 파일](#)

[시스템 레벨 로그 위치](#)

[참조 설명서](#)

---

## 소개

이 문서에서는 VM(Virtual Machine) 및 XR Router에서 Cisco Routed PON(Passive Optical Network) 솔루션의 검증에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

Cisco에서는 이러한 항목에 대한 지식을 권장합니다.

- Cisco IOS® XR 소프트웨어
- Linux
- 가상 머신 환경

# 사용되는 구성 요소

이 문서의 정보는 나열된 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- NCS-540-28Z4C-SYS-A XR Router
- Cisco IOS® XR Software 24.1.2
- Routed PON 버전 24.1.2
- Ubuntu 버전 20.04.06 LTS

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## 검증 단계 - XR 라우터

### 컨피그레이션 확인

#### Linux 네트워킹

연결에 사용되는 VRF(Virtual Routing and Forwarding)가 Linux 네트워킹 컨피그레이션에 반영되어 있는지 확인합니다. 이 예에서는 VRF Mgmt-intf가 구성되었습니다. 또한 source-hint default-route가 올바른 업링크 인터페이스에 대해 설정되었는지 확인합니다. 나열된 예의 연결은 인터페이스 MgmtEth0/RP0/CPU0/0을 사용합니다.

컨피그레이션 예시:

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

#### 물리적 및 하위 인터페이스 컨피그레이션

OLT(Optical Line Terminal) Pluggable이 삽입되는 인터페이스가 올바르게 컨피그레이션에서 종료되지 않았는지 확인합니다. 또한 하위 인터페이스가 4090으로 태그된 dot1q이며 관련 물리적 인터페이스에 적용되는지 확인합니다.

컨피그레이션 예시:

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

명령 확인:

```
<#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
```

전역 컨피그레이션에서 LLDP가 활성화되어 있는지 확인합니다.

```
<#root>
```

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

```
show run | include lldp
```

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

```
lldp
```

### PON-Controller 컨피그레이션

xr-pon-ctrl RPM이 설치되어 있고 활성화 상태인지 확인합니다. 그렇지 않은 경우 하드 디스크에 NCS540i-iosxr-optional-RPMs-24.2.11.tar이 있는지 확인하고(Linux 셸의 경우 경로는 /misc/disk1/), 소프트웨어 일치 RPM이 포함된 local-repo가 올바르게 참조되었는지 확인합니다.

---

참고: 시스템 전체 RPM의 설치 및 관리에 대한 정보는 [System Setup and Software Installation Guide for Cisco NCS 540 Series Routers, IOS XR Release 24.1.x, 24.2.x 링크](#)에서 찾을 수 있습니다

---

예:

```
<#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 컨트롤러가 올바르게 연결된 파일, 파일 경로 및 VRF로 구성되어 있는지 확인합니다.

예:

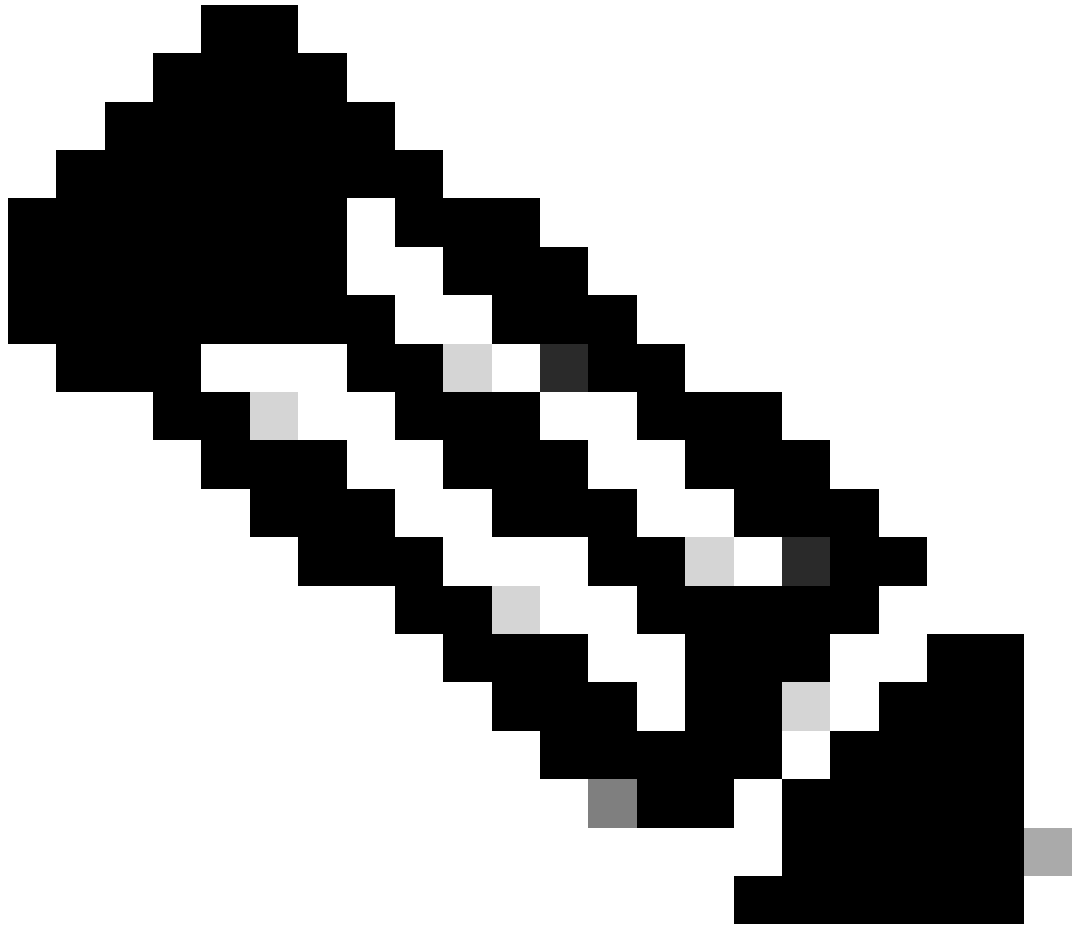
```
<#root>
```

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

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

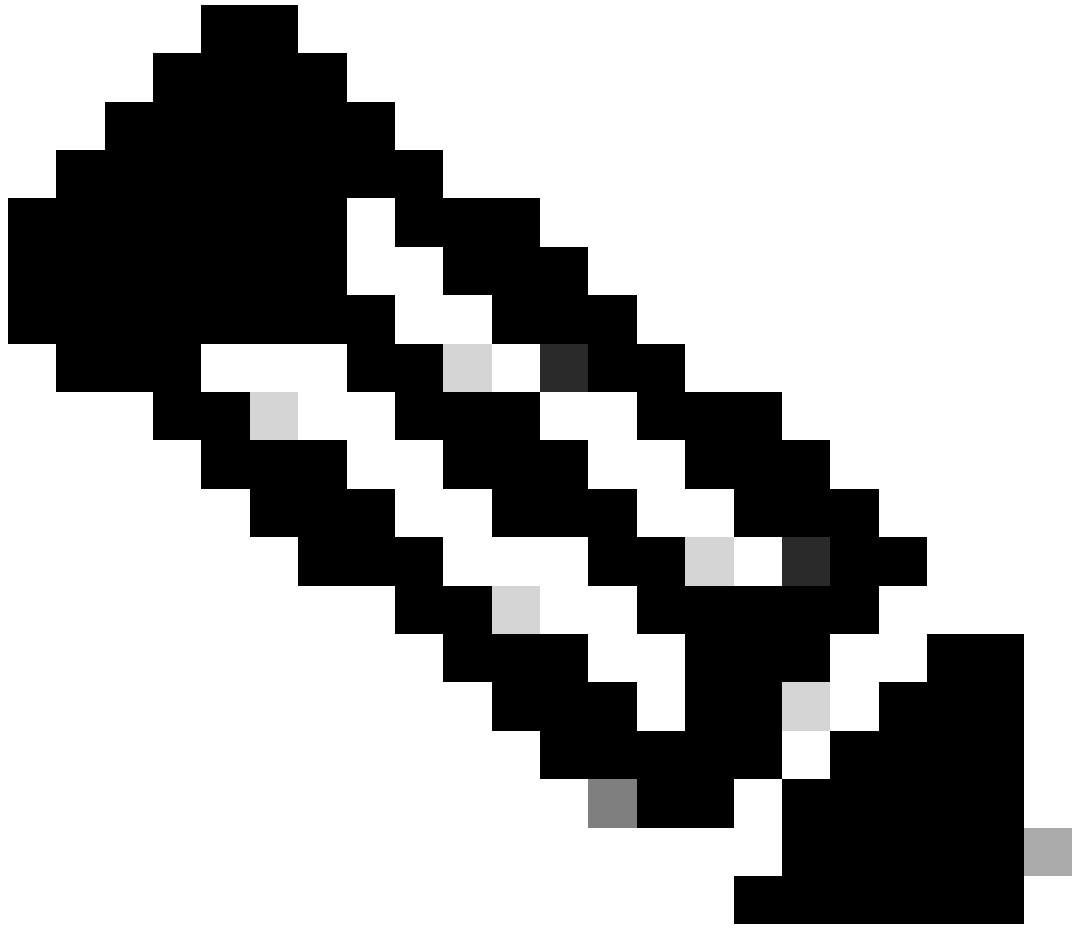
## 확인

### JSON 파일



참고: PonCntlInit.json 파일 예는 VM에 Routed PON Manager 소프트웨어를 설치할 때 포함됩니다.

---



참고: PON Manager를 단일 VM에 설치하면 MongoDB IP와 VM IP가 같은 VM에 설치됩니다.

---



---

참고: 나열된 예에서는 TLS를 사용하지 않습니다. TLS를 사용하는 경우 설치에 사용자 이름과 비밀번호가 올바르게 설정되었는지 확인합니다.

---

MongoDB의 IP가 host: 섹션에 설정되어 PON 컨트롤러가 연결하는 것에 일치하는지 확인합니다. 또한 구성된 포트가 VM의 mongod.conf 파일의 포트와 일치하는지 확인합니다.

예:

```
<#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"
}
```

## 연결 확인

XR 라우터에서 MongoDB/VM 호스팅 라우팅된 PON Manager를 ping합니다. VRF를 사용하는 경우 VRF에서 소스를 가져옵니다.

예:

```
<#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#
```

## 컨테이너 상태

PON 컨트롤러는 XR 라우터의 docker 컨테이너에서 실행됩니다. XR 라우터의 linux 셸에 로그인하여 컨테이너의 상태를 확인한 다음 docker ps 명령을 실행합니다. 현재 가동 중인 컨테이너와 활성 컨테이너가 있는 경우 이를 표시합니다.

예:

```
<#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
--------------	-------	---------	---------	--------	-------	-------

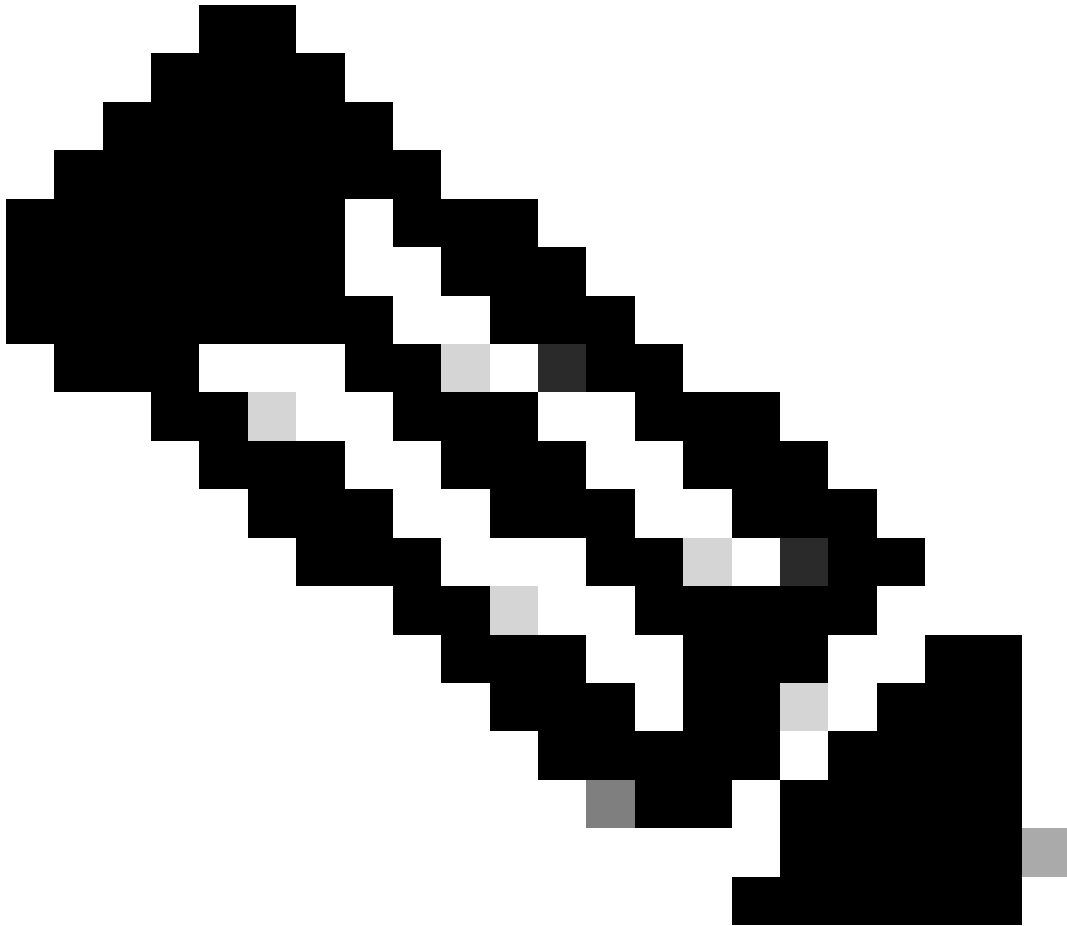
```
2e700f202ee3
```

tibit-poncnt1.xr:R4.0.0	"/usr/bin/supervisor..."	3 days ago	Up	3 days		
-------------------------	--------------------------	------------	----	--------	--	--

pon\_ctlr

docker 컨테이너가 실행되고 있지 않으면 JSON 파일의 내용과 파일 구조를 확인합니다. Docker 컨테이너의 로그에서 활성 오류가 있는지 확인합니다. 로그 예는 컨트롤러에 등록하는 ONU를 나타냅니다. 이는 또한 컨테이너 및 OLT와 관련된 모든 도커 레벨 오류를 출력합니다. 또한 간단한 show logging을 실행하여 오류 메시지를 확인하는 방법을 안내받을 수 있습니다.

---



참고: 사용법 -follow는 docker 내의 최신 로그 내용을 표시합니다.

---

예:

```
<#root>
```

```
[node0_RPO_CPU0:~]$
```

```
docker logs pon_ctlr
```

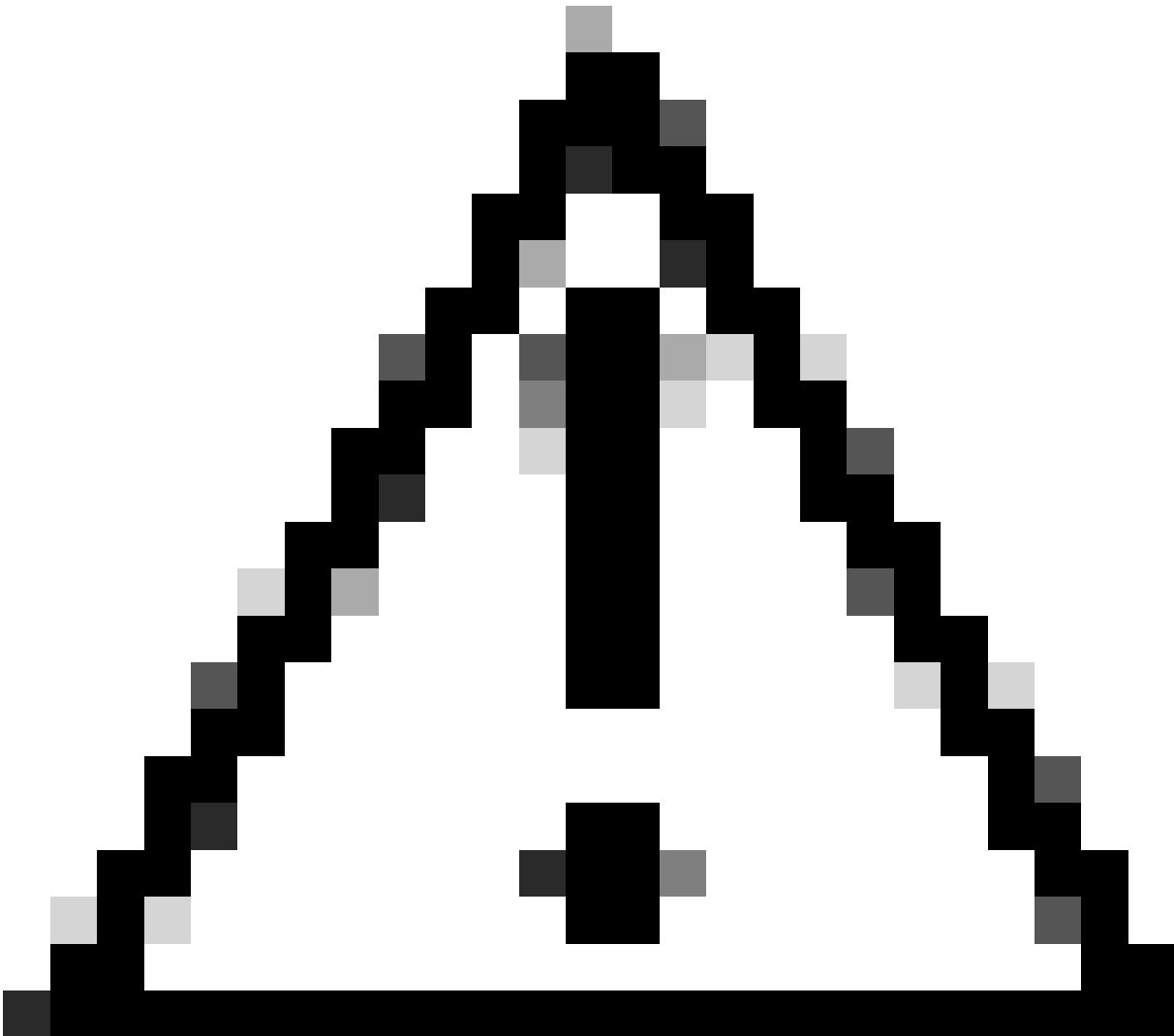
2024-07-16 15:05:11.630 PonCnt1 System Status

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

## 날짜 및 시간

XR 라우터와 VM이 라우티드 PON Manager를 호스팅하는 시간과 날짜가 일치하는지 확인합니다. 가능한 경우 최적의 정확성을 위해 동일한 NTP 서버를 사용합니다.

---



주의: VM과 XR 라우터 간에 NTP가 동기화되지 않으면 라우팅된 PON Manager의 OLT 가 시성에 직접적인 영향을 미칩니다.

---

예:

```
<#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
```

컨피그레이션 예시:

```
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
```

메시지 추적

PON 프로세스는 ltrace를 통해 추가 로깅을 생성합니다. 이 로그에서는 이 프로세스와 관련된 오류를 확인합니다.

예:

```
<#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
```

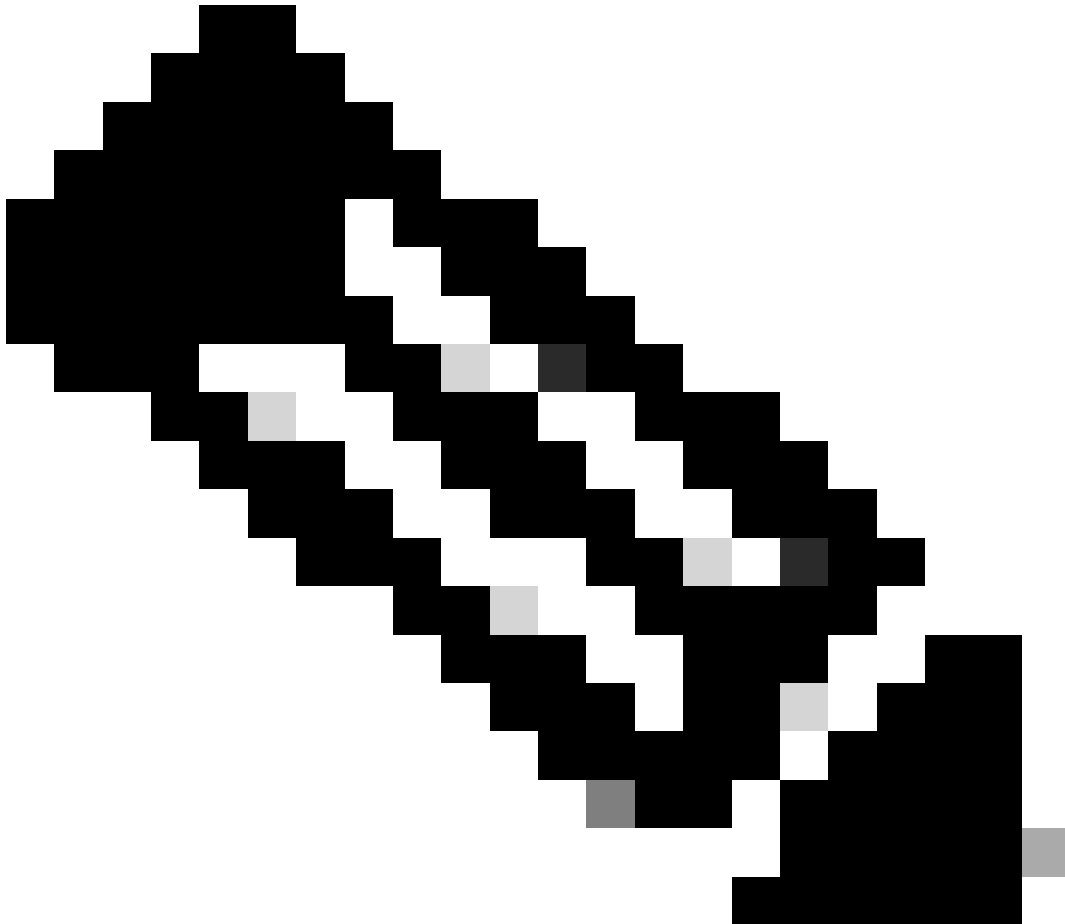
## 검증 단계 - Linux VM

확인

## status.sh 스크립트

Routed PON Manager 설치 디렉토리 내에는 각 관련 프로세스의 현재 상태를 표시하는 셸 스크립트(status.sh)가 있습니다. 이 스크립트를 높은 권한으로 실행하여 나열된 각 서비스가 실행 중인지 확인합니다. 서비스 중 하나가 실행되고 있지 않은 경우 먼저 설치를 수행할 때 실행된 설치 스크립트를 확인하고 설치 설명서에 따라 적절한 인수가 설정되었는지 확인합니다.

---



참고: Cisco Routed PON Manager 설치 가이드는 Cisco Routed PON [Manager 설치 가이드 링크](#)에서 [확인할 수 있습니다](#)

---

moongod.봉사  
apache2.service  
netconf.service  
netopeer2-server.service

예:

<#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



•

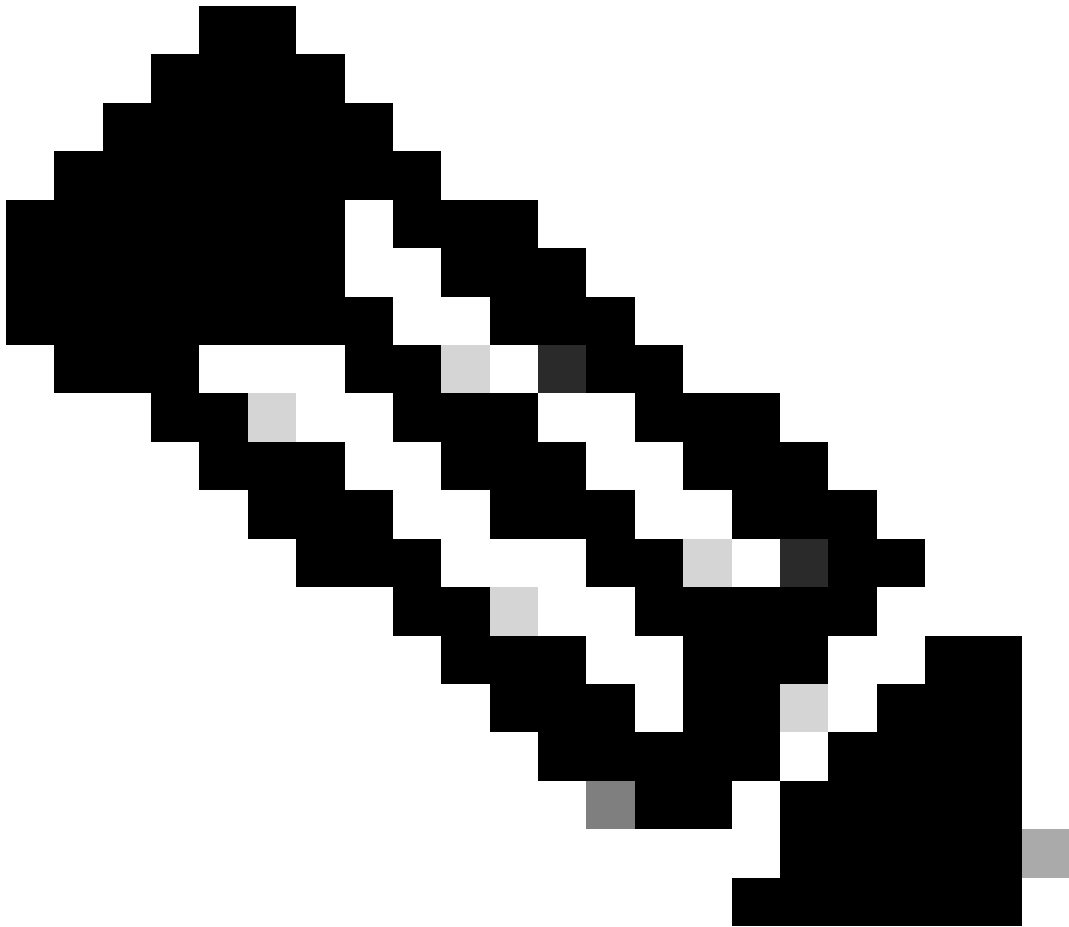
`tibit-netopeer2-server.service`

- 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

넷플랜

Netplan을 검증하고 IP 정보가 유효한지, VM 네트워크 인터페이스 이름이 정확한지, VLAN ID 4090이 생성되어 할당되었는지, 유효한 Netplan YAML 트리 구조를 사용하고 있는지 확인합니다.



---

참고: netplan YAML 파일은 /etc/netplan/에 있습니다.

---

예:

```
<#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
```

VM의 IP 컨피그레이션을 확인하고 구성된 네트워크 어댑터가 netplan YAML 파일에 나열된 것과 일치하는지 확인합니다.

---

참고: `sudo netplan --debug apply`의 사용은 응용 프로그램 이전에 netplan을 테스트할 때 유용합니다.

---

예:

```
<#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 연결

ping을 통해 PON 컨트롤러를 호스팅하는 XR 라우터에 대한 IP 연결을 확인합니다.

예:

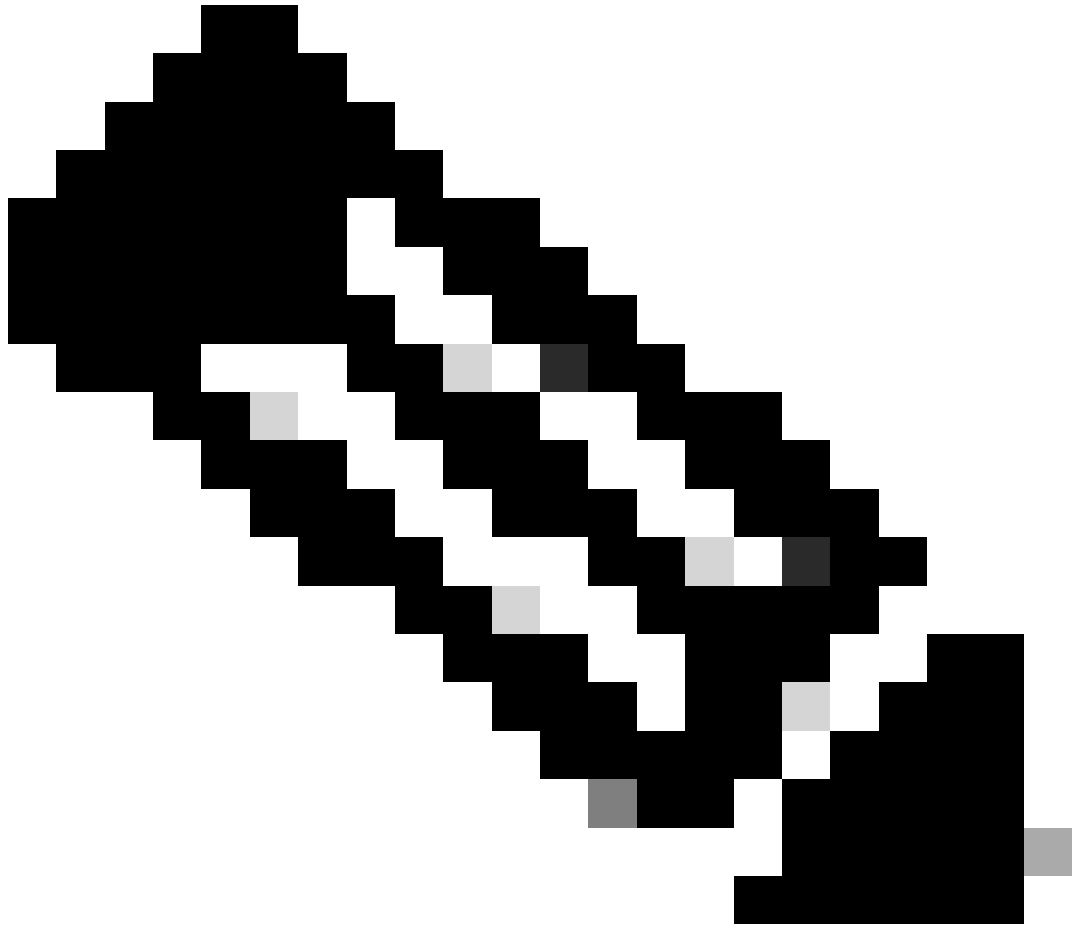
```
<#root>
```

```
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 Port 27017이 열려 있는지 확인합니다. MongoDB에 비표준 포트를 사용하는 경우 netstat -tunl을 통해 해당 포트가 열려/수신되는지 확인합니다.



참고: 표준 MongoDB TCP 포트는 27017.

---

---

참고: 4단계에 나열된 컨피그레이션 파일은 MongoDB에서 사용할 TCP 포트 컨피그레이션도 설정합니다.

---

예:

<#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			

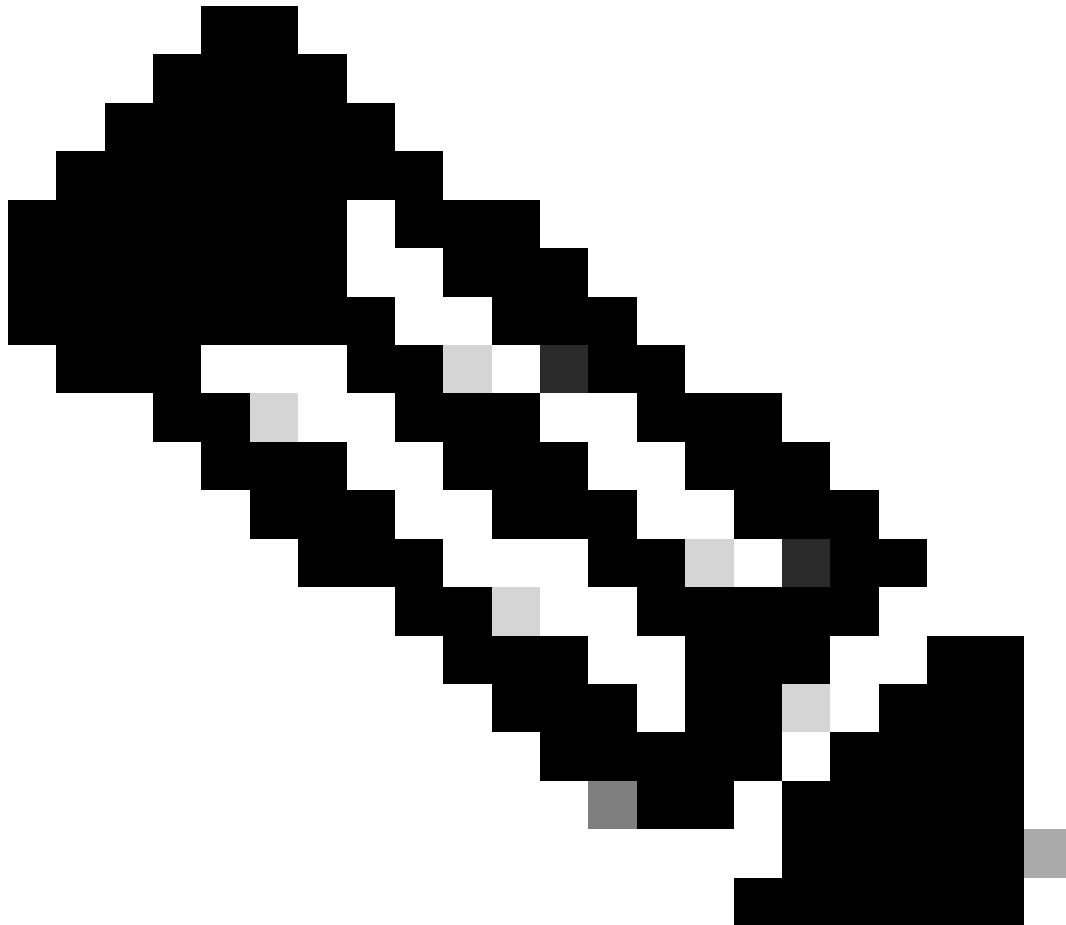
10.122.140.232:27017

0.0.0.0:\* LISTEN

## MongoDB 컨피그레이션 파일

mongod.conf 파일이 정확하고 bindIP:에 올바른 IP가 나열되어 있는지 확인합니다.

---



참고: MongoDB 구성 파일은 /etc/mongod.conf에 있습니다.

---

예:

```
<#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 --
```

## 시스템 레벨 로그 위치

각 서비스에 대한 시스템 레벨 로그는 linux 내에서 관리됩니다. 이러한 로그는 /var/log 디렉토리 내에, 특히 이러한 트리 아래에 저장됩니다.

MongoDB 로그: /var/log/mongod/mongod.log  
Apache 로그: /var/log/apache2/<filename>.log  
가상 머신 Syslog: /var/log/syslog

## 참조 설명서

- [Cisco 지원 및 다운로드 페이지](#)
- [Cisco Routed PON 솔루션 페이지](#)
- [Cisco Routed PON 설치 설명서](#)
- [Cisco Routed PON 구축 설명서](#)
- [Cisco Routed PON, Cisco IOS XR 릴리스 24.1.1 및 24.1.2의 릴리스 정보](#)



이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.