

BGP Adjacency Validation and Automated Health Reporting

NX-OSv Switch (DC Role) & Cisco 8000v IOS-XE (Edge Role)

Anthony Edjenuwa

January 2, 2026

Abstract

This report documents a production-style lab validating an eBGP adjacency between an NX-OSv switch (data-centre leaf/border role) and a Cisco IOS-XE 8000v router (enterprise WAN/edge role). The work demonstrates layered troubleshooting methodology, adjacency verification, loopback route exchange, and automated health reporting using a Linux automation host. An AI component is optionally used to generate interpretative commentary and structured output; the API endpoint is redacted for security.

Disclosure (API Redaction)

The AI API endpoint is intentionally excluded by the author.

[API excluded by author — insert your endpoint here]

1 Scenario (Real-World Context)

A typical enterprise network uses BGP between a data-centre routing domain and a WAN edge router. This enables scalable exchange of infrastructure prefixes (loopbacks), service routes, and future segmentation. In operations, BGP adjacency failures are among the most common outage causes due to reachability, ASN mismatch, VRF mismatch, or TCP/179 filtering.

2 Lab Topology

BGP Adjacency Validation and Automated Health Reporting

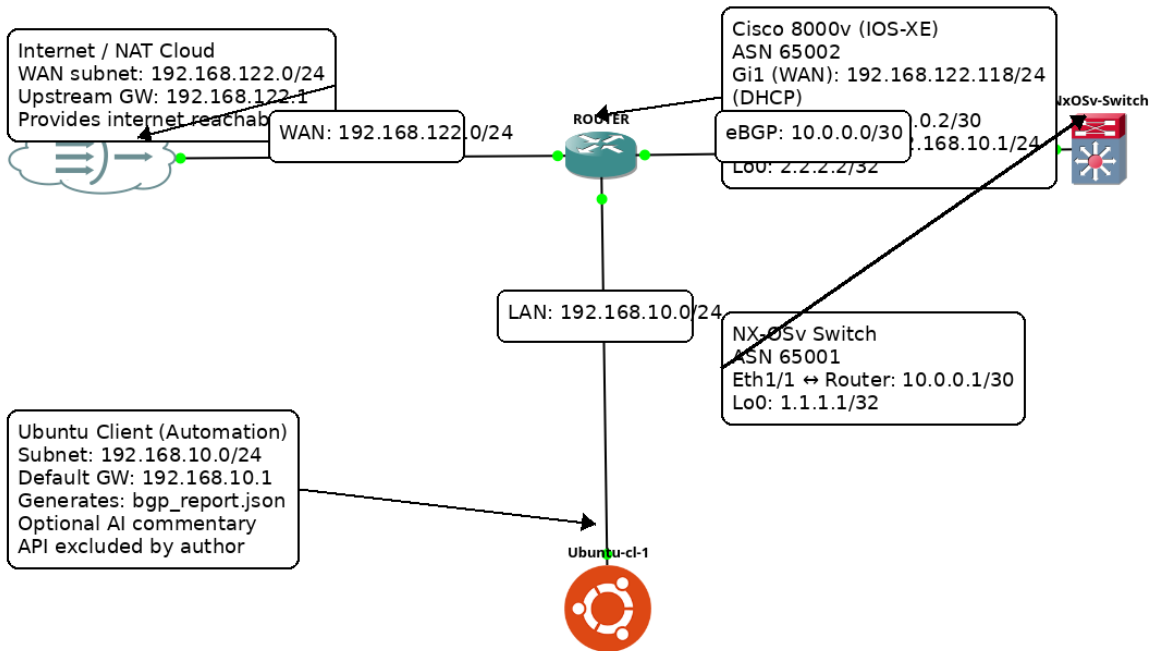


Figure 1: Lab topology with labelled interfaces, ASNs, and addressing (labels positioned to avoid covering devices).

Table 1: Device roles and addressing summary

Node	Role / Interface	Addressing
Cisco 8000v	ASN	65002
Cisco 8000v	WAN (Gi1)	192.168.122.118/24 (DHCP), GW 192.168.122.1
Cisco 8000v	Transit (Gi2)	10.0.0.2/30 (eBGP peer to NX-OS)
Cisco 8000v	LAN (Gi3)	192.168.10.1/24 (default gateway for Ubuntu)
Cisco 8000v	Loopback0	2.2.2.2/32
NX-OSv Switch	ASN	65001
NX-OSv Switch	Transit (Eth1/1)	10.0.0.1/30 (eBGP peer to IOS-XE)
NX-OSv Switch	Loopback0	1.1.1.1/32
Ubuntu Client	Validation Host	192.168.10.0/24, GW 192.168.10.1

3 Implementation Summary

3.1 BGP Intent

- Establish eBGP between 10.0.0.1 (NX-OS) and 10.0.0.2 (IOS-XE).
- Advertise loopbacks 1.1.1.1/32 and 2.2.2.2/32.
- Validate adjacency state (Established) and route installation in RIB/FIB.

3.2 Key Configuration (Representative)

NX-OS (AS 65001):

Listing 1: NX-OS representative BGP configuration

```
router bgp 65001
  router-id 1.1.1.1
  neighbor 10.0.0.2 remote-as 65002
  address-family ipv4 unicast
    network 1.1.1.1/32
```

IOS-XE (AS 65002):

Listing 2: IOS-XE representative BGP configuration

```
router bgp 65002
  bgp router-id 2.2.2.2
  neighbor 10.0.0.1 remote-as 65001
  address-family ipv4
```

```
network 2.2.2.2 mask 255.255.255.255
```

4 Verification Method (Operational Troubleshooting Flow)

This lab follows a layered operational workflow:

1. L1/L2: Interfaces up/up and stable.
2. L3: Peer reachability on 10.0.0.0/30 and ARP resolution.
3. L4: TCP/179 not blocked by ACL/firewall.
4. BGP: Neighbor state transitions to Established.
5. Routes: Loopbacks learned and installed in the routing table.

4.1 Evidence Commands Used

Listing 3: NX-OS verification commands

```
show bgp ipv4 unicast summary
show bgp ipv4 unicast
show ip route 2.2.2.2/32
```

Listing 4: IOS-XE verification commands

```
show ip bgp summary
show ip bgp
show ip route 1.1.1.1
```

Listing 5: Ubuntu validation commands

```
ping -c 2 192.168.10.1
ping -c 2 8.8.8.8
```

5 Results

5.1 Control-Plane Outcome

The BGP session reached **Established** state on both peers. Each side received one prefix, confirming successful adjacency and route exchange.

5.2 Route Exchange Outcome

- NX-OS originates 1.1.1.1/32 and learns 2.2.2.2/32 via 10.0.0.2.
- IOS-XE originates 2.2.2.2/32 and learns 1.1.1.1/32 via 10.0.0.1.

6 AI Response (Redacted API) and Structured Output

The AI component is used to generate structured commentary and/or JSON output to assist troubleshooting workflows. The endpoint is redacted.

6.1 API Redaction Statement

[API excluded by author — insert your endpoint here]

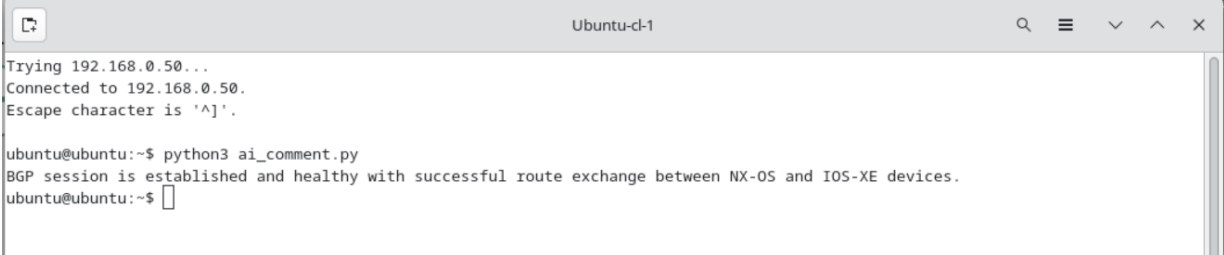
6.2 Sample AI Structured JSON Response (from Lab Run)

Listing 6: AI JSON response captured during testing (endpoint redacted)

```
{
  "issue_summary": "BGP session is established and healthy.",
  "bgp_state_detected": "Established on both sides",
  "most_likely_root_cause": "No issue detected (session Established and routes
    exchanged)",
  "confidence": 1.0,
  "evidence": [
    "Prefix exchange confirmed: NX-OS has 2.2.2.2/32 via 10.0.0.2",
    "Prefix exchange confirmed: IOS-XE has 1.1.1.1/32 via 10.0.0.1",
    "Internet reachability confirmed from edge/Ubuntu"
  ],
  "next_commands_nxos": [
    "show bgp ipv4 unicast summary",
    "show bgp ipv4 unicast",
    "show ip route 2.2.2.2/32"
  ],
  "next_commands_iosxe": [
    "show ip bgp summary",
    "show ip bgp",
    "show ip route 1.1.1.1"
  ],
  "safe_fix_steps": [],
}
```

```
"verification_steps": [  
  "Confirm neighbor remains Established (no flaps).",  
  "Ping loopbacks using source loopback on both devices.",  
  "Confirm routes remain installed in the RIB."  
]  
}
```

7 Devices Health Report

A terminal window titled 'Ubuntu-cl-1' showing the output of a script. The output indicates a successful connection to 192.168.0.50 and a healthy BGP session between NX-OS and IOS-XE devices.

```
Trying 192.168.0.50...  
Connected to 192.168.0.50.  
Escape character is '^['.  
  
ubuntu@ubuntu:~$ python3 ai_comment.py  
BGP session is established and healthy with successful route exchange between NX-OS and IOS-XE devices.  
ubuntu@ubuntu:~$
```

Figure 2: Devices Health Report.

8 Conclusion

This lab demonstrates a stable eBGP adjacency between NX-OS and IOS-XE with validated route exchange and operational evidence collection. The automation host successfully validated connectivity and captured structured AI output without exposing the API endpoint. This methodology aligns with real-world enterprise troubleshooting and monitoring practices.