Lab0 2

Phase 1 — Enterprise Foundation

Lab 02 — Linux Host Networking & Validation

Section A — Enterprise Build Scenario

Lab Type

Enterprise Foundation Build Lab

Difficulty

Beginner → Intermediate

Estimated Time

90–120 Minutes

Lab Objective

In this lab, you will deploy and validate a dedicated Linux services host inside an enterprise lab environment.

You will learn how to:

- deploy an Ubuntu server in EVE-NG,
- configure static IP addressing,
- validate Layer 2 and Layer 3 connectivity,
- test enterprise management protocols,
- verify DNS and gateway operations,
- and establish a reusable Linux services node for future automation and enterprise service labs.

Why This Lab Matters

In real enterprise environments, Linux systems are heavily used for:

- automation platforms,
- monitoring systems,
- syslog servers,
- SNMP tools,

- REST APIs,
- Ansible,
- Python scripting,
- and infrastructure testing.

Even though lightweight VPCS nodes are used for most enterprise validation tasks, full Linux systems become critical for enterprise services and automation workflows.

This lab introduces the operational workflow required to manage Linux systems inside enterprise networks.

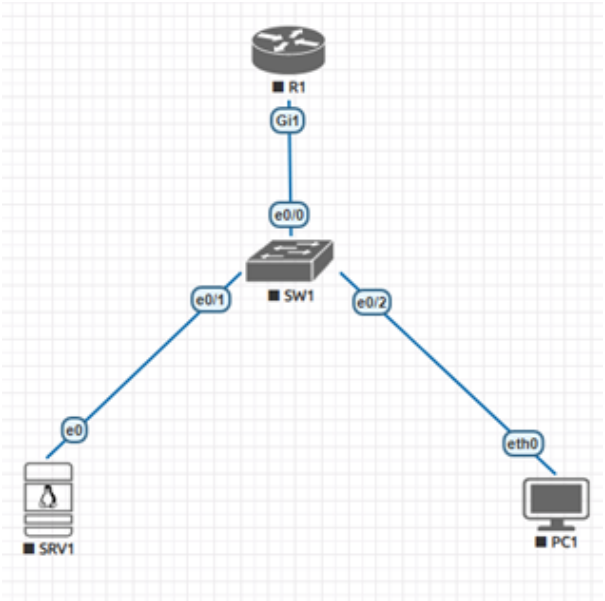
Enterprise Scenario

Your company is preparing an enterprise infrastructure services network.

You are assigned to:

- deploy a Linux services server,
- integrate it into the enterprise topology,
- validate connectivity,
- and prepare the system for future automation and monitoring services.

Topology for EVE-NG & GNS3



Connection Table

Device	Interface	Connected To	Interface
R1	Ethernet0/1	SW1	Ethernet0/0
SW1	Ethernet0/1	SRV1	eth0
SW1	Ethernet0/2	PC1	eth0

IP Addressing Table

Device	Interface	IP Address	Gateway
R1	Ethernet0/1	192.168.20.1/24	N/A
SRV1	eth0	192.168.20.10/24	192.168.20.1
PC1	eth0	192.168.20.20/24	192.168.20.1

Enterprise Naming Standards

Device Type	Naming Convention
Router	R1
Switch	SW1
Linux Services Server	SRV1
Lightweight Test Host	PC1

Required Images

Device	Purpose
Cisco IOLL3	Enterprise routing
Cisco IOLL2	Enterprise switching
Ubuntu Server	Enterprise services node
VPCS	Lightweight endpoint validation

Step 1 — Deploy Nodes in EVE-NG

Goal

Create the enterprise services topology.

Add The Following Nodes

Node	Quantity
IOLL2 Router	1
IOLL2 Switch	1
Ubuntu Server	1

VPCS Host	1

Step 2 — Connect The Enterprise Topology

Goal

Create all network connections.

Use the Connection Table above.

Step 3 — Configure PC1

Commands

```
ip 192.168.20.20/24 192.168.20.1

set pcname PC1

save
```

Verification

```
PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.20.20/24
GATEWAY     : 192.168.20.1
DNS         :
MAC         : 00:50:79:66:68:04
LPORT       : 20000
RHOST:PORT  : 127.0.0.1:30000
MTU         : 1500
```

Expected Result

PC1 should have the correct addressing and gateway.

Step 4 — Configure SW1

Commands

```
enable

conf t

hostname SW1

interface Ethernet0/0
```

```
no shutdown

interface Ethernet0/1

no shutdown

interface Ethernet0/2

no shutdown

end

write memory
```

Verification

```
SW1#show interfaces status

Port         Name               Status       Vlan       Duplex  Speed Type
Et0/0                           connected    1            full   auto 10/100/1000BaseT
X
Et0/1                           connected    1            full   auto 10/100/1000BaseT
X
Et0/2                           connected    1            full   auto 10/100/1000BaseT
X
Et0/3                           connected    1            full   auto 10/100/1000BaseT
X
```

Expected Result

All interfaces should be operational.

Step 5 — Configure R1

Commands

```
enable

conf t

hostname R1

interface Ethernet0/1

ip address 192.168.20.1 255.255.255.0

no shutdown

end
```

```
write memory
```

Verification

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
Ethernet0/0            unassigned      YES unset  administratively down down
Ethernet0/1            192.168.20.1    YES manual up                    up
Ethernet0/2            unassigned      YES unset  administratively down down
Ethernet0/3            unassigned      YES unset  administratively down down
```

Expected Result

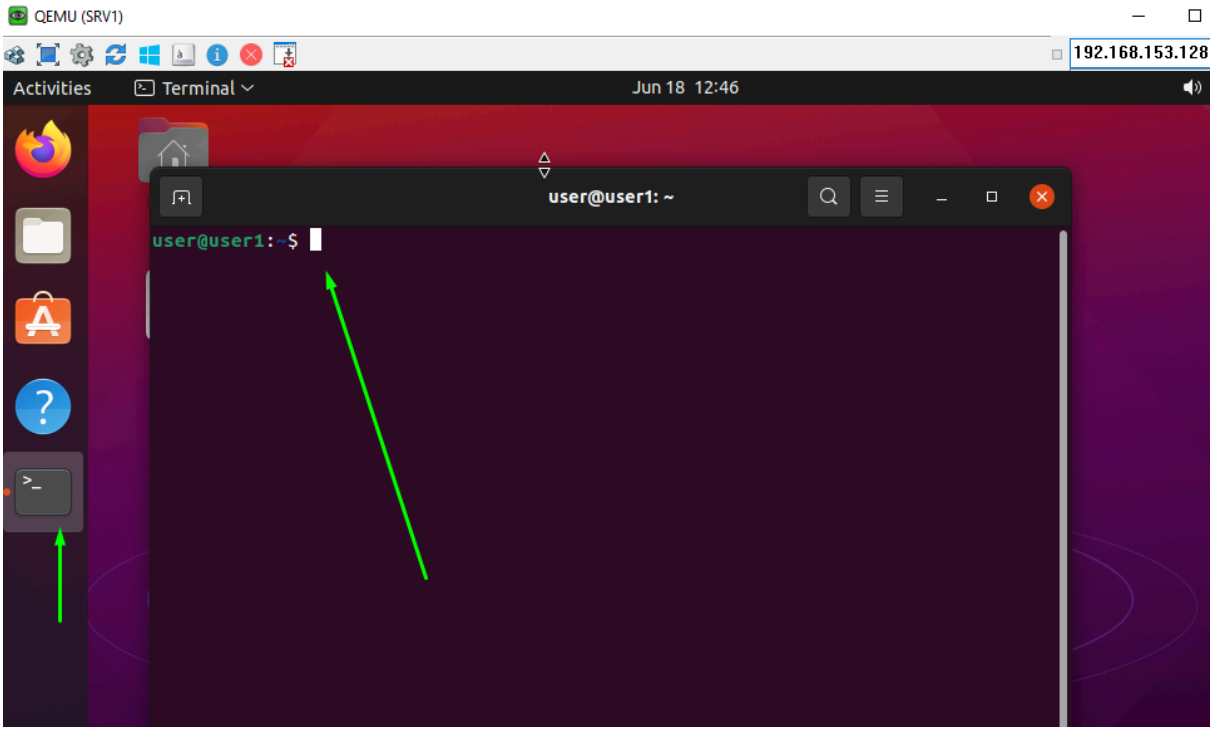
GigabitEthernet1 should be UP/UP.

Step 6 — Configure Ubuntu Server Networking (password: Test123)

Goal

Configure static enterprise IP addressing.

Commands




```
Ip a (verify interface: ens3)

sudo ip addr add 192.168.20.10/24 dev ens3
```

```
sudo ip link set ens3 up

sudo ip route add default via 192.168.20.1
```

⚠️ Important Note:

Before testing, always run the ip a command to verify the interface name and confirm the IP address is assigned correctly.

This IP configuration method is temporary and may be removed after a while and by rebooting it will remove

Verification

```
user@user1:~$ ip addr show ens3

2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defa
ult qlen 1000
    link/ether 00:50:00:00:05:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 192.168.20.10/24 scope global ens3
       valid_lft forever preferred_lft forever


 user@user1:~$ ip route

default via 192.168.20.1 dev ens3
169.254.0.0/16 dev ens3 scope link metric 1000
192.168.20.0/24 dev ens3 proto kernel scope link src 192.168.20.10
```

Expected Result

- ens3 should be UP
- Static IP should appear
- Default route should exist

Step 7 — Validate Gateway Reachability

From SRV1

```
user@user1:~$ ping -c 2 192.168.20.1

PING 192.168.20.1 (192.168.20.1) 56(84) bytes of data.
64 bytes from 192.168.20.1: icmp_seq=1 ttl=255 time=2.61 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=255 time=6.03 ms
```

Expected Result

Gateway should respond successfully.

Step 8 — Validate End-to-End Connectivity

From SRV1

```
user@user1:~$ ping -c 2 192.168.20.20

PING 192.168.20.1 (192.168.20.20) 56(84) bytes of data.
64 bytes from 192.168.20.20: icmp_seq=1 ttl=255 time=2.61 ms
64 bytes from 192.168.20.20: icmp_seq=2 ttl=255 time=6.03 ms
```

Expected Result

PC1 should respond successfully.

Step 9 — Validate Reverse Connectivity

From PC1

```
PC1> ping 192.168.20.10

84 bytes from 192.168.20.10 icmp_seq=1 ttl=64 time=1.532 ms
84 bytes from 192.168.20.10 icmp_seq=2 ttl=64 time=2.451 ms
```

Expected Result

SRV1 should respond successfully.

Step 10 — Enterprise Linux Validation

Goal

Validate enterprise Linux operations.

Verify ARP Table

```
user@user1:~$ ip neigh

192.168.20.20 dev ens3 lladdr 00:50:79:66:68:04 REACHABLE
192.168.20.1 dev ens3 lladdr aa:bb:cc:00:20:10 STALE
```

Verify Interface Status

```
user@user1:~$ ip link show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAU
LT group default qlen 1000
    link/ether 00:50:00:00:05:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
```

Verify Routing Table

```
user@user1:~$ ip route

default via 192.168.20.1 dev ens3
169.254.0.0/16 dev ens3 scope link metric 1000
192.168.20.0/24 dev ens3 proto kernel scope link src 192.168.20.10
```

Expected Result

- Interface operational
- Gateway reachable
- ARP entries populated
- Route functionality depends on proper gateway configuration

Step 11 — Basic Enterprise Verification

R1 Verification

```
show ip interface brief

show running-config
```

Switch (SW1) Verification

```
show interfaces status

show mac address-table
```

Ubuntu Verification

```
ip addr

ip route

ip neigh

ping 192.168.20.1
```

VPCS Verification

```
show ip

ping 192.168.20.10
```

Common Mistakes

Problem	Cause
Ubuntu unreachable	Interface down
Gateway unreachable	Wrong default route
Ping failure	Incorrect subnet mask
SSH failure	SSH not enabled
Missing ARP entries	Layer 2 issue

Enterprise Engineering Notes

- Linux services hosts are typically deployed separately from lightweight endpoint testing hosts.
- Enterprise automation systems usually run on Linux servers.
- Always validate Layer 2 before troubleshooting Layer 3.
- Static addressing simplifies early enterprise lab troubleshooting.

Step 13 — Save Configurations

Router (R1)

```
write memory
```

Switch (SW1)

```
write memory
```

Step 14 — Backup Linux Network Configuration

Goal

Document enterprise services configuration.

Suggested Actions

```
ip addr > srv1-ip.txt

ip route > srv1-route.txt
```

What You Learned

In this lab you learned:

- Enterprise Linux deployment
- Static IP configuration
- Gateway validation
- Enterprise connectivity testing
- ARP verification
- SSH testing
- Enterprise operational workflows

Verification Score

Task	Status
Ubuntu configured	☐
Router configured	☐
Switch configured	☐
Connectivity verified	☐
SSH tested	☐
Configurations saved	☐

Section B — Enterprise Troubleshooting Scenario

Incident Scenario

Users report that the enterprise Linux services server cannot communicate with the default gateway.

Your task:

- investigate the issue,
- identify the root cause,
- restore enterprise connectivity.

Step 1 — Introduce Failure

On SRV1

sudo ip route del default

Step 2 — Verify Failure Symptoms

From SRV1

```
ping 192.168.20.1
```

Expected Result

Gateway communication should fail.

Step 3 — Begin Investigation

Verify Routing Table

```
ip route
```

Expected Result

Default route should be missing.

Root Cause Analysis

The enterprise Linux server lost its default gateway configuration.

Without a default route:

- external communication fails,
- enterprise services become unreachable,
- automation systems may fail to operate.

Step 4 — Restore Enterprise Connectivity

Commands

```
sudo ip route add default via 192.168.20.1
```

Step 5 — Validate Recovery

From SRV1

```
ping 192.168.20.1
```

Expected Result

Gateway communication should be restored.

Final Enterprise Validation

Ubuntu

```
ip route

ping 192.168.20.1
```

Router

```
show ip interface brief
```

Switch

```
show interfaces status
```

Final Engineering Note

Enterprise troubleshooting always follows a structured workflow:

1. Verify physical connectivity
2. Verify interface state
3. Verify Layer 2 operation
4. Verify IP addressing
5. Verify routing

6. Restore service
7. Validate end-to-end recovery