

# LFCS: Networking (25%)

This section covers **Linux networking**, hostname resolution, time synchronization, OpenSSH configuration, packet filtering, routing, bridges, and load balancing.

---

## Goals

### Networking 25%

1. Configure IPv4 and IPv6 networking and hostname resolution
  2. Set and synchronize system time using time servers
  3. Monitor and troubleshoot networking
  4. Configure the OpenSSH server and client
  5. Configure packet filtering, port redirection, and NAT
  6. Configure static routing
  7. Configure bridge and bonding devices
  8. Implement reverse proxies and load balancers
- 

## 1. Configure IPv4/IPv6 Networking & Hostname Resolution

### Learn

- Assign IP addresses (static & dynamic).
- Configure hostnames and DNS resolution.
- Understand `/etc/hosts`, `/etc/resolv.conf`, and `hostnamectl`.

### Commands

```
# View network interfaces
ip addr show
ip link show

# Configure static IP (example)
sudo nmcli con mod eth0 ipv4.addresses 192.168.1.100/24
sudo nmcli con mod eth0 ipv4.gateway 192.168.1.1
sudo nmcli con mod eth0 ipv4.dns 8.8.8.8
sudo nmcli con mod eth0 ipv4.method manual
sudo nmcli con up eth0

# Set hostname
hostnamectl set-hostname myhost
hostnamectl status

# DNS check
cat /etc/resolv.conf
ping google.com
```

## Exercises

1. Assign a static IP to an interface and verify connectivity.
  2. Change the hostname and update `/etc/hosts` accordingly.
- 

# 2. Time Synchronization

## Learn

- Sync system time using `chrony` or `ntpd`.
- Check and update time zone.

## Commands

```
# Check current time
timedatectl
```

```
# Set time zone
timedatectl set-timezone Europe/Paris

# Sync time using chrony
sudo systemctl start chronyd
sudo systemctl enable chronyd
chronyc tracking
chronyc sources
```

## Exercises

1. Configure NTP synchronization with a public server.
2. Verify time sync status using `chronyc tracking`.

---

# 3. Monitor & Troubleshoot Networking

## Learn

- Use tools to check connectivity and troubleshoot issues.

## Commands

```
ping <host>
traceroute <host>
ss -tulnp      # check listening ports
netstat -rn    # routing table
curl -I http://example.com
dig example.com # DNS query test
```

## Exercises

1. Ping a remote host and check for packet loss.
  2. Check which service is listening on port 80.
  3. Use `traceroute` to identify network path latency.
- 

## 4. OpenSSH Configuration

### Learn

- Configure SSH server and client.
- Manage SSH keys and permissions.

### Commands

```
# Start and enable SSH server
sudo systemctl start sshd
sudo systemctl enable sshd

# Connect to remote server
ssh user@remote_host

# Generate SSH key pair
ssh-keygen -t rsa -b 4096

# Copy SSH public key to remote server
ssh-copy-id user@remote_host

# Change SSH port (example: 2222)
sudo nano /etc/ssh/sshd_config

# Port 2222
sudo systemctl reload sshd
```

### Exercises

1. Configure SSH to listen on a non-default port.
2. Set up key-based authentication.
3. Disable root password login for security.

---

# 5. Packet Filtering, Port Redirection, and NAT

## Learn

- Configure firewalls using `iptables` or `firewalld`.
- Perform NAT and port forwarding for network access.

## Commands

```
# Check firewall status
sudo firewall-cmd --state

# Allow SSH port 22
sudo firewall-cmd --add-port=22/tcp --permanent
sudo firewall-cmd --reload

# Add NAT rule (masquerade)
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# View rules
sudo iptables -t nat -L -n -v
```

## Exercises

1. Open a specific port (e.g., 8080) on the firewall.
2. Configure port forwarding for SSH (e.g., 2222 → 22).
3. Enable masquerading to share internet from one interface to another.

---

# 6. Static Routing

# Learn

- Add static routes to control network traffic paths.

## Commands

```
# Add a static route
sudo ip route add 192.168.2.0/24 via 192.168.1.1 dev eth0

# Delete a static route
sudo ip route del 192.168.2.0/24

# Show routes
ip route show
```

## Exercises

1. Add a static route to reach a remote network.
  2. Verify connectivity via the static route using `ping` or `traceroute`.
- 

# 7. Bridge & Bonding Devices

## Learn

- Create network **bridges** for virtual machines or containers.
- Create **bonded interfaces** for redundancy or link aggregation.

## Commands

```
# Create a bridge
sudo nmcli con add type bridge con-name br0 ifname br0
sudo nmcli con add type bridge-slave con-name eth0-br0 ifname eth0 master br0

# Create bonded interface
```

```
sudo nmcli con add type bond con-name bond0 ifname bond0 mode active-backup
sudo nmcli con add type bond-slave con-name eth1-bond0 ifname eth1 master bond0
sudo nmcli con add type bond-slave con-name eth2-bond0 ifname eth2 master bond0
```

```
# View connection status
```

```
nmcli con show
```

## Exercises

1. Create a bridge interface and attach an Ethernet device.
2. Configure a bonded interface using two NICs in active-backup mode.

---

# 8. Reverse Proxies & Load Balancers

## Learn

- Understand reverse proxies and load balancing concepts.
- Use **Nginx** as a basic reverse proxy or load balancer.

## Commands

```
# Install and configure Nginx
sudo apt install nginx -y

# Nginx reverse proxy configuration example
sudo nano /etc/nginx/conf.d/reverse.conf

# Example config:
# server {
#     listen 80;
#     location / {
#         proxy_pass http://backend:8080;
#     }
}
```

```
# }
```

```
# Enable and reload Nginx  
sudo systemctl enable nginx  
sudo systemctl reload nginx
```

## Exercises

1. Configure Nginx to proxy requests to a backend server.
2. Implement round-robin load balancing for multiple backend servers.
3. Test the setup using `curl` or a web browser.

## Exam Tips

- Always verify **connectivity and DNS resolution** (`ping`, `dig`, `nslookup`).
- Understand **network configuration files**:
  - `/etc/NetworkManager/system-connections/`
  - `/etc/sysconfig/network-scripts/` (RHEL-based)
- Practice switching between **NetworkManager CLI** (`nmcli`) and `ip` commands.
- Familiarize yourself with **firewalld zones** and **NAT configurations**.
- Understand **basic Nginx directives** for proxying and load balancing.

Revision #3

Created 2025-11-07 10:34:21 UTC by Loïc

Updated 2025-11-07 14:07:54 UTC by Loïc