

Aplicații Practice: Rețea și Servere Linux

Timp lucru: 120min

Introducere

Acest material acoperă configurarea practică de rețea în Linux și instalarea a două servere de bază: SSH și Nginx.

1. Vei învăța:

- Configurare IP statică vs DHCP
- Vizualizare și interpretare tabel de rutare
- Instalare și configurare SSH server
- Instalare Nginx cu structură de directoare

!!! Material pentru admini începători Linux. Necesară privilegii root/sudo.

1. Configurare IP: Static vs DHCP

1.1 Ce este DHCP?

DHCP (Dynamic Host Configuration Protocol) atribuie automat adrese IP dispozitivelor din rețea.

Avantaje DHCP:

- Configurare automată (zero intervenție)
- Ideal pentru laptop-uri și dispozitive mobile
- Management centralizat de DHCP server

1.2 Ce este IP Static?

IP static înseamnă configurare manuală a adresei IP, care rămâne fixă.

Avantaje IP Static:

- Adresă fixă, predictibilă
- Necesar pentru servere (SSH, web, DNS)
- Control total asupra configurației

1.3 Verificare Configurație Curentă

Înainte de modificări, verifică configurația existentă:

```
# Verifică interfețele de rețea
ip addr show
# SAU
ip a

# Verifică doar o interfață
ip addr show eth0
```

Output-ul va arăta:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP>
    inet 192.168.1.100/24 brd 192.168.1.255
        inet6 fe80::a00:27ff:fe4e:66a1/64
```

- inet = adresă IPv4
- /24 = subnet mask (255.255.255.0)
- UP = interfața este activă

1.4 Configurare IP Static (Ubuntu/Debian)

!!! Pe Ubuntu 18.04+, configurarea se face prin Netplan. Pe versiuni mai vechi sau Debian, folosești /etc/network/interfaces.

Metoda 1: Netplan (Ubuntu 18.04+)

Editează fișierul de configurare:

```
sudo nano /etc/netplan/01-netcfg.yaml
```

Conținut pentru IP static:

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.1.100/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

Aplică configurația:

```
sudo netplan apply
```

```
# Verifică
ip addr show eth0
```

Metoda 2: /etc/network/interfaces (Debian)

```
sudo nano /etc/network/interfaces

# Adaugă:
auto eth0
iface eth0 inet static
  address 192.168.1.100
  netmask 255.255.255.0
  gateway 192.168.1.1
  dns-nameservers 8.8.8.8 8.8.4.4

# Restart networking
sudo systemctl restart networking
```

1.5 Revenire la DHCP

Pentru a reveni la DHCP:

```
# Netplan (Ubuntu)
sudo nano /etc/netplan/01-netcfg.yaml

network:
  version: 2
  ethernets:
    eth0:
      dhcp4: yes

sudo netplan apply

# -----
# /etc/network/interfaces (Debian)
auto eth0
iface eth0 inet dhcp

sudo systemctl restart networking
```

1.6 Test Conectivitate

```
# Ping gateway  
ping -c 4 192.168.1.1  
  
# Ping internet  
ping -c 4 8.8.8.8  
  
# Ping hostname (test DNS)  
ping -c 4 google.com
```

2. Tabelul de Rutare

2.1 Ce este Tabelul de Rutare?

Tabelul de rutare indică sistemului unde să trimită pachetele de rețea în funcție de adresa IP destinație.

2.2 Vizualizare Tabel de Rutare

Există mai multe comenzi pentru afișare:

```
# Comandă nouă (recomandată)
ip route show
# SAU scurt:
ip r

# Comandă veche (deprecated dar funcțională)
route -n

# Detaliat cu interfețe
netstat -rn
```

2.3 Interpretare Output

Exemplu output ip route show:

```
default via 192.168.1.1 dev eth0 proto dhcp metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.100
```

Explicații:

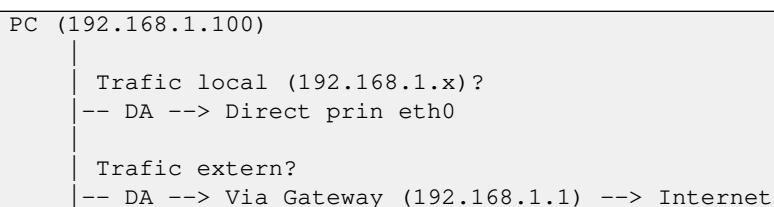
Linia 1: default via 192.168.1.1

- default = orice destinație necunoscută (0.0.0.0/0)
- via 192.168.1.1 = gateway-ul (routerul)
- dev eth0 = prin interfața eth0
- Înseamnă: "Tot traficul pentru internet merge la 192.168.1.1"

Linia 2: 192.168.1.0/24

- 192.168.1.0/24 = rețeaua locală
- dev eth0 = direct prin eth0
- src 192.168.1.100 = IP-ul tău sursă
- Înseamnă: "Traficul local (192.168.1.x) nu trece prin router"

2.4 Diagrama Simplificată



2.5 Adăugare/Stergere Rute (Avansat)

!!! Aceste comenzi sunt pentru scenarii avansate. În mod normal, rutele sunt configurate automat.

```
# Adaugă rută statică
sudo ip route add 10.0.0.0/8 via 192.168.1.254 dev eth0

# Sterge rută
sudo ip route del 10.0.0.0/8

# Schimbă gateway default
sudo ip route del default
sudo ip route add default via 192.168.1.254
```

3. SSH Server

3.1 Ce este SSH?

SSH (Secure Shell) permite conectarea la distanță securizată la un server Linux. Întreaga comunicație este criptată.

Utilizări SSH:

- Administrare servere remote
- Transfer fișiere securizat (SCP, SFTP)
- Tunelare conexiuni (port forwarding)
- Executare comenzi remote

3.2 Instalare OpenSSH Server

```
# Verifică dacă e instalat
systemctl status ssh
# SAU
systemctl status sshd

# Instalare (dacă lipsește)
sudo apt update
sudo apt install openssh-server -y

# Verifică că rulează
sudo systemctl status ssh

# Pornire automată la boot
sudo systemctl enable ssh
```

3.3 Verificare Port și Conexiuni

SSH ascultă implicit pe portul 22:

```
# Verifică portul SSH
sudo ss -tulpn | grep ssh
# SAU
sudo netstat -tulpn | grep :22

# Output așteptat:
tcp      0      0 0.0.0.0:22    0.0.0.0:*      LISTEN      1234/sshd
```

3.4 Test Conexiune Locală

```
# Conectare la localhost
ssh localhost
# SAU
ssh 127.0.0.1

# Conectare cu username specific
ssh user@localhost
# Prima dată vei primi:
The authenticity of host 'localhost' can't be established.
RSA key fingerprint is SHA256:...
Are you sure you want to continue? yes
# Ieșire din sesiune SSH
exit
```

3.5 Test Conexiune de pe Alt PC

De pe alt computer din rețea:

```
# Conectare la server
ssh user@192.168.1.100

# Cu port specific (dacă ai schimbat)
ssh -p 2222 user@192.168.1.100
```

3.6 Configurare Bază SSH

Fișierul de configurare principal:

```
sudo nano /etc/ssh/sshd_config
```

Setări importante:

```
# Port (implicit 22, schimbă pentru securitate)
Port 22
# Dezactivează login root (RECOMANDAT)
PermitRootLogin no

# Autentificare cu parolă (yes) sau doar cu cheie (no)
PasswordAuthentication yes

# Număr maxim de încercări autentificare
MaxAuthTries 3
```

După modificări, restart SSH:

```
sudo systemctl restart ssh
# SAU
sudo systemctl restart sshd
```

!!! Nu dezactiva PasswordAuthentication dacă nu ai configurat SSH keys! Riiști să te blochezi!

3.7 Firewall (UFW)

Dacă folosești firewall, permite portul SSH:

```
# Verifică status firewall
sudo ufw status

# Permite SSH
sudo ufw allow ssh
# SAU specific port
sudo ufw allow 22/tcp

# Activează firewall (dacă e dezaktivat)
sudo ufw enable
```

3.8 Troubleshooting SSH

Probleme comune:

1. "Connection refused"

```
# Verifică că SSH rulează
sudo systemctl status ssh

# Dacă nu rulează, pornește-l
sudo systemctl start ssh
```

2. "Permission denied"

- Username greșit
- Parolă greșită
- PermitRootLogin no (dacă încerci root)

3. Firewall blochează

```
sudo ufw allow 22/tcp  
sudo systemctl restart ufw
```

4. Nginx Web Server

4.1 Ce este Nginx?

Nginx (pronunțat "engine-x") este un web server rapid și eficient, folosit pentru servire pagini web, proxy reverse, load balancing.

Utilizări Nginx:

Web server pentru site-uri statice și dinamice

Reverse proxy pentru aplicații

Load balancer pentru trafic mare

4.2 Instalare Nginx

```
# Update repository
sudo apt update

# Instalare Nginx
sudo apt install nginx -y

# Verifică status
sudo systemctl status nginx

# Pornire automată
sudo systemctl enable nginx
```

Testează în browser:

Deschide browser și accesează: http://localhost sau http://IP_SERVER

Ar trebui să vezi pagina "Welcome to nginx!"

4.3 Structura Directoarelor Nginx

```
# Directorul web principal (document root)
/var/www/html/

# Configurații site-uri
/etc/nginx/sites-available/ # Configurații disponibile
/etc/nginx/sites-enabled/   # Configurații active (symlinks)

# Configurație principală
/etc/nginx/nginx.conf

# Log-uri
/var/log/nginx/access.log    # Cereri HTTP
/var/log/nginx/error.log      # Erori
```

4.4 Crearea Structurii de Directoare

Vom crea următoarea structură:

```
/var/www/html/
├── index.html (pagina principală)
└── docs/
    ├── 2024/
    │   └── fisier_2024.html
    ├── 2025/
    │   └── fisier_2025.html
    └── 2026/
        └── fisier_2026.html
```

Pasul 1: Creează directoarele

```
sudo mkdir -p /var/www/html/docs/{2024,2025,2026}
```

Pasul 2: Creează fișierele HTML

```
# Fișier 2024
sudo tee /var/www/html/docs/2024/fisier_2024.html > /dev/null <<EOF
<!DOCTYPE html>
<html>
<head>
    <title>Documente 2024</title>
</head>
<body>
    <h1>Arhivă Documente 2024</h1>
    <p>Acesta este conținutul pentru anul 2024.</p>
    <p>Data creării: $(date)</p>
    <a href="/">Înapoi la pagina principală</a>
</body>
</html>
EOF

# Similar pentru 2025
sudo tee /var/www/html/docs/2025/fisier_2025.html > /dev/null <<EOF
<!DOCTYPE html>
<html>
<head>
    <title>Documente 2025</title>
</head>
<body>
    <h1>Arhivă Documente 2025</h1>
    <p>Acesta este conținutul pentru anul 2025.</p>
    <a href="/">Înapoi la pagina principală</a>
</body>
</html>
EOF

# Similar pentru 2026
sudo tee /var/www/html/docs/2026/fisier_2026.html > /dev/null <<EOF
<!DOCTYPE html>
<html>
<head>
    <title>Documente 2026</title>
</head>
<body>
    <h1>Arhivă Documente 2026</h1>
    <p>Acesta este conținutul pentru anul 2026.</p>
    <a href="/">Înapoi la pagina principală</a>
</body>
</html>
EOF
```

Pasul 3: Creează pagina principală index.html

```

sudo tee /var/www/html/index.html > /dev/null <<EOF
<!DOCTYPE html>
<html lang="ro">
<head>
    <meta charset="UTF-8">
    <title>Server Documente</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            max-width: 800px;
            margin: 50px auto;
            padding: 20px;
        }
        h1 { color: #333; }
        .link-box {
            background: #f0f0f0;
            padding: 15px;
            margin: 10px 0;
            border-radius: 5px;
        }
        a {
            color: #0066cc;
            text-decoration: none;
            font-size: 18px;
        }
        a:hover { text-decoration: underline; }
    </style>
</head>
<body>
    <h1> Server Documente - Arhivă Anuală</h1>
    <p>Bine ai venit! Selectează anul pentru documente:</p>

    <div class="link-box">
        <strong> Anul 2024:</strong><br>
        <a href="/docs/2024/fisier_2024.html">→ Accesează documente 2024</a>
    </div>

    <div class="link-box">
        <strong> Anul 2025:</strong><br>
        <a href="/docs/2025/fisier_2025.html">→ Accesează documente 2025</a>
    </div>

    <div class="link-box">
        <strong> Anul 2026:</strong><br>
        <a href="/docs/2026/fisier_2026.html">→ Accesează documente 2026</a>
    </div>

    <hr>
    <p><a href="/docs/"> Browse toate directoarele /docs/</a></p>
</body>
</html>
EOF

```

Pasul 4: Setează permisiuni corecte

```

sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/

```

4.5 Configurare Nginx pentru Listare Directoare

Pentru a activa listarea automată directoarelor /docs/:

```
sudo nano /etc/nginx/sites-available/default
```

Adaugă în blocul server {} (după location / {}):

```
server {
    listen 80 default_server;
    root /var/www/html;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }

    # ADAUGĂ ACEST BLOC:
    location /docs/ {
        autoindex on;           # Activează listare directoare
        autoindex_exact_size off; # Afisează KB/MB în loc de bytes
        autoindex_localtime on;  # Timestamp local
    }
}
```

Testează configurația și reload:

```
# Verifică sintaxa
sudo nginx -t
```

```
# Dacă "syntax is ok", reload
sudo systemctl reload nginx
```

4.6 Testare Finală

Accesează în browser:

1. Pagina principală:

```
http://IP_SERVER/
http://localhost/
```

Vei vedea pagina cu linkuri către anii 2024, 2025, 2026.

2. Directorul docs cu listare:

```
http://IP_SERVER/docs/
```

Vei vedea lista cu subdirectoare: 2024/, 2025/, 2026/

3. Fișierele individuale:

```
http://IP_SERVER/docs/2024/fisier_2024.html
http://IP_SERVER/docs/2025/fisier_2025.html
http://IP_SERVER/docs/2026/fisier_2026.html
```

4.7 Log-uri Nginx

Pentru debugging și monitorizare:

```
# Urmărește access log (cine accesează site-ul)
sudo tail -f /var/log/nginx/access.log

# Urmărește error log (erori)
sudo tail -f /var/log/nginx/error.log

# Ultimele 20 linii access
sudo tail -20 /var/log/nginx/access.log
```

4.8 Comenzi Utile Nginx

```
# Start/stop/restart
sudo systemctl start nginx
sudo systemctl stop nginx
sudo systemctl restart nginx

# Reload config fără restart (preferabil)
sudo systemctl reload nginx

# Status
sudo systemctl status nginx

# Testare config
sudo nginx -t

# Versiune
nginx -v
```

5. Rezumat și Verificare Finală

5.1 Checklist Complet

- IP Static/DHCP configurat și testat (ping gateway, ping internet)
- Tabel rutare verificat (ip route show)
- SSH server instalat și funcțional (ssh localhost)
- Nginx instalat și rulează (systemctl status nginx)
- Directoare /docs/2024/2025/2026 create
- Fișiere HTML create în fiecare subdirector
- Pagina principală funcționează (<http://IP/>)
- Listare directoare funcționează (<http://IP/docs/>)
- Toate linkurile funcționează

5.2 Comenzi Esențiale - Quick Reference

```

# NETWORKING
ip addr show           # Verifică IP
ip route show          # Tabel rutare
ping -c 4 8.8.8.8      # Test conectivitate

# SSH
sudo systemctl status ssh # Status SSH
ssh user@IP             # Conectare SSH
sudo nano /etc/ssh/sshd_config # Config SSH

# NGINX
sudo systemctl status nginx   # Status Nginx
sudo nginx -t               # Test config
sudo systemctl reload nginx  # Reload config
sudo tail -f /var/log/nginx/access.log # Log-uri

# FISIERE
ls -la /var/www/html/docs/  # Listare directoare
sudo nano /var/www/html/index.html # Editare pagină

```

5.3 Diagrama Finală



!!! Păstrează serverul actualizat: sudo apt update && sudo apt upgrade