Amministrazione Server Linux

Network e Condivisione Risorse

Prof. Fedeli Massimo

Tutti i diritti riservati

25 dicembre 2025

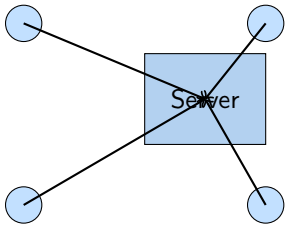
Sommario

Definizione

Un **server Linux** è un sistema operativo configurato per fornire servizi specifici ad altri computer attraverso una rete, operando 24/7/365.

Caratteristiche principali:

- Disponibilità continua
- Gestione remota
- Sicurezza rafforzata
- Monitoraggio automatico
- Scalabilità



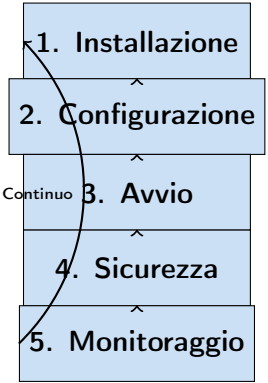
Porte e Servizi Standard

Servizio	Porta	Protocollo
SSH	22	TCP
HTTP	80	TCP
HTTPS	443	TCP
FTP	21	TCP
DNS	53	TCP/UDP
SMTP	25	TCP
MySQL	3306	TCP
PostgreSQL	5432	TCP

Importante

La gestione corretta delle porte è cruciale per sicurezza e funzionalità!

Le 5 Fasi di Setup



Fase 1: Installazione - Distribuzioni

Enterprise:

- Red Hat Enterprise Linux (RHEL)
- Ubuntu Server
- SUSE Linux Enterprise

Community:

- Fedora
- Debian
- Rocky Linux / AlmaLinux

Installazione Pacchetti

```
# dnf grouplist
# dnf groupinstall "Web Server"
# dnf install httpd mod_ssl
```

Principali Tipi di Server

Web & File

- Apache/Nginx
- Samba (SMB/CIFS)
- NFS
- FTP (vsftpd)

Database & Mail

- MariaDB/MySQL
- PostgreSQL
- Postfix
- Dovecot

Infrastruttura

- DNS (BIND)
- DHCP
- LDAP
- NTP (chrony)

Esempio: Installazione Apache

```
# dnf install httpd mod_ssl
# systemctl enable httpd
# systemctl start httpd
```

Fase 2: File di Configurazione

Struttura tipica in /etc:

```
/etc/httpd/
          conf/
                httpd.conf          # Config principale
          conf.d/
                ssl.conf            # Moduli aggiuntivi
                php.conf
          conf.modules.d/
```

Best Practice

- Usa `vim` invece di `vi` (syntax highlighting)
- Backup prima di modificare: `cp file file.bak`
- Test configurazione prima di riavviare servizio
- Documenta le modifiche

Fase 3: Gestione Servizi con systemd

Comandi Base:

```
# Stato servizio
systemctl status httpd

# Start/Stop
systemctl start httpd
systemctl stop httpd
systemctl restart httpd

# Ricarica config
systemctl reload httpd
```

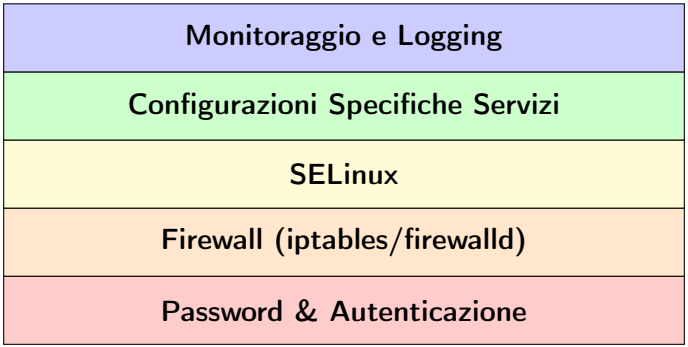
Avvio Automatico:

```
# Abilita all'avvio
systemctl enable httpd

# Disabilita
systemctl disable httpd

# Verifica
systemctl is-enabled httpd
```

Attenzione

Verificare sempre lo stato dopo modifiche con `systemctl status`!

Livelli di Sicurezza



Principio: Difesa in profondità (Defense in Depth)

Autenticazione: Password vs Chiavi SSH

Password

- − Vulnerabile a brute force
- − Può essere intercettata
- − Deve essere ricordata
- \+ Semplice da configurare

```
# Disabilita login root
# /etc/ssh/sshd_config
PermitRootLogin no
```

Chiavi SSH

- \+ Impossibile brute force
- \+ Crittograficamente sicura
- \+ Automazione possibile
- \+ No password da ricordare

```
# Genera chiave
ssh-keygen -t rsa -b 4096

# Copia su server
ssh-copy-id user@server
```

Firewall: firewalld

Concetti Base:

- **Zone**: Livelli di fiducia
- **Servizi**: Porte predefinite
- **Runtime**: Temporaneo
- **Permanent**: Persistente

```
# Stato firewall
firewall-cmd --state

# Zone attive
firewall-cmd --get-active-zones

# Permetti HTTP
firewall-cmd --permanent \
  --add-service=http
firewall-cmd --reload
```

Esempio: Apertura Porta Custom

```
firewall-cmd --permanent --add-port=8080/tcp
firewall-cmd --reload
firewall-cmd --list-ports
```

SELinux: Security Enhanced Linux

Modalità Operative

Enforcing	Blocca azioni non autorizzate (PRODUZIONE)
Permissive	Registra violazioni ma non blocca (DEBUG)
Disabled	SELinux disattivato (NON RACCOMANDATO)

Context

- User
- Role
- Type
- Level

Boolean

- On/Off switches
- Modifica policy
- Runtime/Permanent

Porte

- Port types
- Servizi associati
- Permessi custom

SELinux: Comandi Essenziali

```
# Verifica stato
getenforce
sestatus

# Cambia modalit  (temporaneo)
setenforce 0  # Permissive
setenforce 1  # Enforcing

# Context file
ls -Z /var/www/html/
restorecon -Rv /var/www/html/

# Boolean
getsebool httpd_can_network_connect
setsebool -P httpd_can_network_connect on

# Porte
semanage port -l | grep http
semanage port -a -t http_port_t -p tcp 8080
```

Sistema di Logging: rsyslog

Architettura

Facility.Priority → Destination

Facility:

- `kern` - Kernel
- `mail` - Email
- `authpriv` - Auth
- `cron` - Scheduler
- `daemon` - Servizi

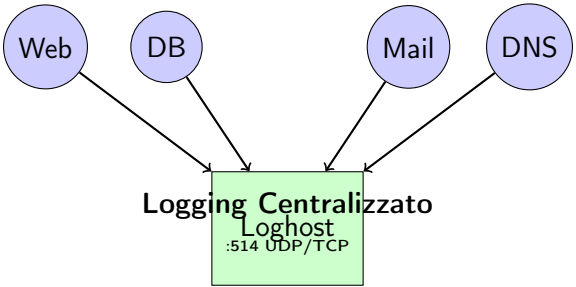
Priority (crescente):

- `debug` < `info` < `notice`
- `warning` < `err` < `crit`
- `alert` < `emerg`

File: /var/log/

messages, secure, maillog, cron, httpd/, audit/

Logging Centralizzato



Client:

```
# /etc/rsyslog.conf
*.* @loghost  # UDP
*.* @@loghost # TCP
```

Server:

```
module(load="imudp")
input(type="imudp" port="514")
```

System Activity Reporter (sar)

Cosa Monitora?

- **CPU**: Utilizzo, idle, I/O wait
- **Memoria**: RAM, swap, paging
- **Disco**: Throughput, latenza, IOPS
- **Rete**: Pacchetti, bandwidth, errori

Raccolta Dati:

- Ogni 10 minuti
- Stored in /var/log/sa/
- 1 mese di history

Visualizzazione:

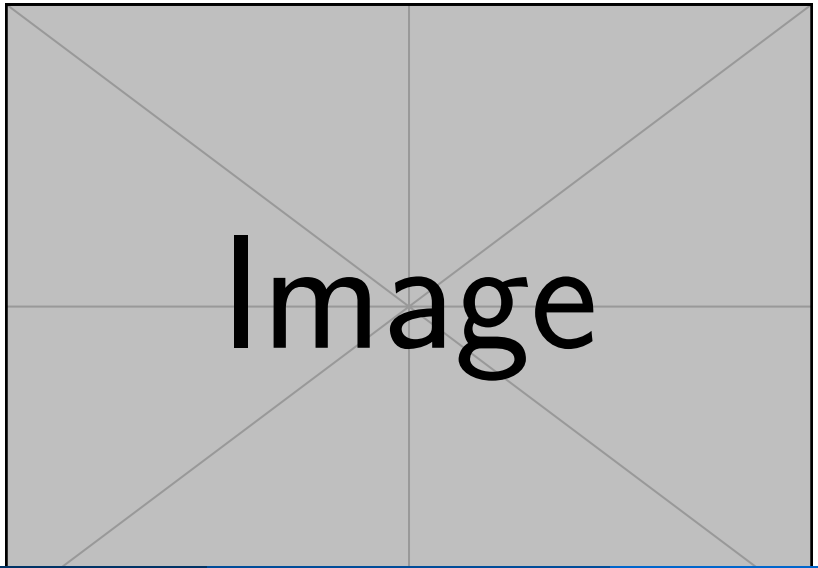
- Report storici
- Live monitoring
- Export per analisi

sar: Esempi Pratici

```
# CPU usage da mezzanotte
sar -u

# Disco I/O
sar -d

# Network traffic
sar -n DEV

# Memoria
sar -r

# Live: campiona ogni 2 sec, 10 volte
sar -u 2 10

# Dati giorno specifico
sar -u -f /var/log/sa/sa15
```

Pro Tip

Cockpit: Monitoring Web-Based



Secure Shell (SSH)

Perché SSH?

Sostituisce protocolli insicuri (telnet, rlogin, rsh, rcp) con comunicazione **crittografata end-to-end**.

Client Tools:

- `ssh` - Remote login
- `scp` - Secure copy
- `sftp` - Secure FTP
- `rsync` - Sync incrementale

Funzionalità:

- Login remoto
- Esecuzione comandi
- Trasferimento file
- Port forwarding
- X11 forwarding
- Tunnel VPN

SSH: Configurazione Server

```
# /etc/ssh/sshd_config

Port 22                              # Porta (considera 2222)
PermitRootLogin no                   # NO LOGIN ROOT!
PasswordAuthentication yes           # Si/No password
PubkeyAuthentication yes             # Chiavi SSH
X11Forwarding yes                    # GUI remoto
ClientAliveInterval 300              # Keep-alive
AllowUsers user1 user2               # Whitelist utenti
DenyUsers baduser                    # Blacklist utenti
```

Sicurezza Critica

`PermitRootLogin no` è **OBBLIGATORIO** in produzione!

SSH: Autenticazione con Chiavi

Setup (una volta):

```
# 1. Genera chiave (client)
ssh-keygen -t rsa -b 4096 -C "mio-laptop"

# 2. Copia su server
ssh-copy-id user@server.com

# 3. Test
ssh user@server.com   # No password!
```

Vantaggi

+ Sicurezza massima
+ Automazione (script, backup)
+ Una chiave, N server
+ Revoca facile (rimuovi chiave pubblica)

Trasferimento File

scp - Copia Singola

```
# Locale -> Remoto
scp file.txt user@host:/path/

# Remoto -> Locale
scp user@host:/file.txt ./

# Ricorsivo
scp -r dir/ user@host:/path/
```

rsync - Sincronizzazione

```
# Sync con delete
rsync -avz --delete \
  /local/ user@host:/remote/

# Bandwidth limit
rsync -avz --bwlimit=1000 \
  /src/ user@host:/dst/
```

rsync vs scp

rsync trasferisce solo **differenze** = molto più efficiente!

sftp: FTP Sicuro

```
$ sftp user@server
sftp> ls                # Lista remota
sftp> lls               # Lista locale
sftp> get file.txt      # Download
sftp> put file.txt      # Upload
sftp> get -r dir/       # Download ricorsivo
sftp> put -r dir/       # Upload ricorsivo
sftp> mget *.log        # Download multipli
sftp> mkdir newdir      # Crea directory
sftp> rm file.txt       # Elimina file
sftp> bye               # Esci
```

Quando usare sftp?

Sessioni **interattive** di esplorazione e trasferimento file.

Monitoraggio Spazio: df e du

df - Filesystem

```
# Human-readable
df -h

# Exclude tmpfs
df -h -x tmpfs -x devtmpfs

# Inodes
df -i

# Tipo specifico
df -t xfs
```

du - Directory

```
# Directory usage
du -h /var

# Solo totale
du -sh /var

# Top 10 largest
du -h /var | sort -hr | head

# Max depth
du -h --max-depth=2 /var
```

Pro Tip

`df` per filesystem totali, `du` per drill-down dettagliato

find: Ricerca Avanzata

```
# File > 100MB
find / -xdev -size +100M -ls

# File utente specifico, ordinati
find / -xdev -user john | xargs ls -lhS > /tmp/john.txt

# Modificati ultimi 7 giorni > 50MB
find /var/log -mtime -7 -size +50M

# Non acceduti da 1 anno
find /home -atime +365 -size +10M

# Pulizia file temporanei vecchi
find /tmp -type f -mtime +30 -delete

# Top 20 directory
find / -xdev -type d -exec du -sh {} \; | sort -hr | head -20
```

logrotate: Rotazione Automatica

Configurazione

File principale: `/etc/logrotate.conf`
Configs servizi: `/etc/logrotate.d/*`

Opzioni Comuni:

- `daily/weekly/monthly`
- `rotate N` - Copie
- `compress` - Gzip
- `delaycompress`
- `missingok`
- `notifempty`

Scripts:

- `prerotate`
- `postrotate`
- `sharedscripts`

Esecuzione:
Via cron: `/etc/cron.daily/`

Da Gestione Manuale ad Automazione Scalabile

Tradizionale (NON scalabile):

- × Installazione manuale
- × Configurazione host-by-host
- × SSH ad ogni server
- × Updates individuali
- × Inconsistenze

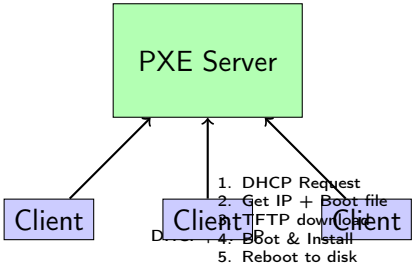
Enterprise (Scalabile):

- ✓ PXE boot
- ✓ Config management
- ✓ Orchestrazione
- ✓ Automazione
- ✓ Consistenza

Regola d'oro

Se devi fare la stessa cosa su > 3 server $\rightarrow$ **AUTOMATIZZA**!

PXE Boot: Installazione di Massa



Vantaggi:

- Installa 100 server simultaneamente
- Configurazione standardizzata
- Zero intervento umano

Configuration Management

Tool Popolari

Ansible, Puppet, Chef, Salt

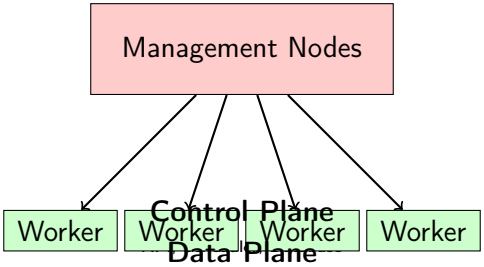
Principi:

1. **Infrastructure as Code** (IaC)
2. **Idempotenza**: Esecuzione multipla = stesso risultato
3. **Declarativo**: "Cosa" non "Come"
4. **Versionamento**: Git per tracking

Esempio: 1 comando, 100 server

```
ansible webservers -m yum -a "name=httpd state=latest"
```

Architettura Management/Worker



Esempi: Kubernetes, OpenShift, OpenStack

Best Practices: Sicurezza

1. **Principio del Minimo Privilegio**
   - Utenti: solo permessi necessari
   - Servizi: utenti dedicati non-root
   - sudo: comandi specifici
2. **Difesa in Profondità**
   - Firewall + SELinux + App security
   - Multi-factor authentication
   - Encryption (data at rest & in transit)
3. **Updates & Patching**
   - Security patches ASAP
   - Test in staging first
   - Finestre manutenzione pianificate

Best Practices: Monitoraggio

Metriche Essenziali (Golden Signals)

- **Latency**: Response time
- **Traffic**: Request rate
- **Errors**: Error rate
- **Saturation**: Resource utilization

Alert Intelligenti:

- Threshold basati su baseline
- Escalation policy chiara
- Evita alert fatigue (troppi falsi positivi)
- Runbook documentati

Proattività:

- Trend analysis
- Capacity planning
- Predictive maintenance

Best Practices: Backup

Strategia 3-2-1

- **3** copie dei dati
- Su **2** media diversi (disco + tape/cloud)
- **1** copia off-site (disaster recovery)

Testing Regolare:

- Restore test mensili
- DR drill trimestrali
- Documenta RTO/RPO (Recovery Time/Point Objective)

Remember

Backup non testato = Backup non esistente!

Best Practices: Documentazione

Architecture Docs:

- Network diagrams
- Data flow
- Dependencies map
- Infrastructure inventory

Operational Docs:

- Runbooks
- Troubleshooting guides
- Emergency procedures
- On-call playbooks

Change Docs:

- Change log
- Approval workflow
- Rollback plans
- Post-mortem reports

Tools:

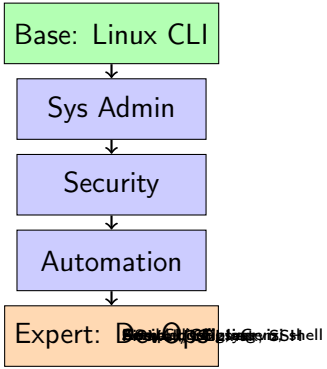
- Wiki (Confluence)
- Git (docs as code)
- Diagrams (draw.io)
- CMDB

"Documentazione obsoleta > Nessuna documentazione"

Riepilogo: 5 Pilastri

1. **Setup Sistematico**
   - Installazione → Config → Start → Secure → Monitor
2. **Sicurezza Multi-Layer**
   - Password/Keys + Firewall + SELinux + App Config
3. **Monitoraggio Continuo**
   - rsyslog + sar + Cockpit + logwatch
4. **Gestione Remota SSH**
   - ssh + scp + rsync + sftp
5. **Automazione Enterprise**
   - PXE + Config Management + Orchestration

Roadmap di Apprendimento



Risorse per Approfondire

Documentazione:

- Man pages: `man comando`
- RHEL Docs
- Arch Wiki
- Gentoo Handbook

Certificazioni:

- RHCSA
- RHCE
- LFCS
- CompTIA Linux+

Community:

- Stack Overflow
- Reddit: r/linux, r/linuxadmin
- IRC/Discord channels
- Local LUG

Libri:

- UNIX & Linux Sys Admin Handbook
- Linux Command Line (Shotts)
- How Linux Works (Ward)

Takeaway Messages

3 Principi Fondamentali

1. **Automazione**: Se ripeti > 2 volte, scrivi script
2. **Sicurezza**: Defense in depth, mai singolo punto di fallimento
3. **Documentazione**: Future-you ti ringrazierà

L'Admin Ideale

- **Lazy**: Automatizza tutto il possibile
- **Paranoico**: Assume sempre il peggio (security)
- **Curioso**: Continua ad imparare
- **Metodico**: Processo $>$ Improvvisazione

Grazie per l'Attenzione!

Domande?

Prof. Fedeli Massimo
Tutti i diritti riservati