Amministrazione Server Linux

Network e Condivisione Risorse

Prof. Fedeli Massimo -

25 dicembre 2025

Indice

1 Introduzione all'Amministrazione dei Server

L'amministrazione dei server Linux rappresenta una competenza fondamentale per chiunque lavori nel campo dell'informatica moderna. Con la crescente diffusione di servizi cloud, data center e infrastrutture distribuite, la capacità di configurare, gestire e monitorare server Linux è diventata essenziale.

1.1 Panoramica

Un server Linux è fondamentalmente un sistema operativo configurato per fornire servizi specifici ad altri computer o utenti attraverso una rete. A differenza dei sistemi desktop, i server sono progettati per funzionare continuamente, 24 ore su 24, 7 giorni su 7, richiedendo elevata affidabilità, sicurezza e prestazioni ottimali.

1.2 Caratteristiche Fondamentali dei Server

Le caratteristiche che distinguono un server da un normale sistema desktop includono:

- **Disponibilità continua**: I server devono rimanere operativi senza interruzioni, garantendo l'accesso costante ai servizi forniti.
- **Gestione remota**: La maggior parte dell'amministrazione viene effettuata remotamente, senza accesso fisico alla macchina.
- **Sicurezza rafforzata**: I server esposti alla rete richiedono misure di sicurezza avanzate per proteggere dati e servizi.
- **Monitoraggio automatico**: Sistemi di logging e reporting automatici consentono di individuare e risolvere problemi rapidamente.
- **Scalabilità**: I server devono poter gestire carichi di lavoro crescenti senza degradazione delle prestazioni.

1.3 Gestione delle Porte e dei Servizi

Ogni servizio di rete su un sistema Linux comunica attraverso porte specifiche. Le porte sono numeri che identificano punti di accesso logici attraverso i quali i dati vengono scambiati. Comprendere la gestione delle porte è cruciale per:

1. Configurare i firewall in modo appropriato
2. Identificare i servizi in esecuzione
3. Diagnosticare problemi di connettività
4. Implementare politiche di sicurezza efficaci

Porte Standard

Alcuni servizi utilizzano porte standard ben note:

- SSH: porta 22
- HTTP: porta 80
- HTTPS: porta 443
- FTP: porta 21
- DNS: porta 53
- SMTP: porta 25

1.4 Monitoraggio Continuo

A differenza dei sistemi desktop che vengono spenti quando non utilizzati, i server operano continuamente. Questo richiede:

- Sistemi di logging centralizzati
- Alert automatici per condizioni anomale
- Report periodici sullo stato del sistema
- Strumenti di analisi delle prestazioni
- Backup automatici e pianificati

Il monitoraggio continuo permette agli amministratori di identificare problemi potenziali prima che diventino critici, garantendo la massima uptime e qualità del servizio.

2 Procedura Standard di Installazione e Configurazione

L'installazione e la configurazione di un server Linux seguono generalmente un processo standardizzato in cinque fasi principali. Questo approccio metodico garantisce che tutti gli aspetti critici vengano considerati e implementati correttamente.

2.1 Fase 1: Installazione del Server

La fase di installazione costituisce il primo passo nella creazione di un server funzionante. In questa fase è importante:

2.1.1 Scelta della Distribuzione

Le distribuzioni Linux più comuni per server includono:

- **Red Hat Enterprise Linux (RHEL)**: Distribuzione commerciale con supporto enterprise
- **Fedora**: Versione community che anticipa le funzionalità di RHEL
- **Ubuntu Server**: Popolare per la facilità d'uso e l'ampia documentazione
- **CentOS/Rocky Linux**: Alternative gratuite compatibili con RHEL
- **Debian**: Stabile e affidabile, base per molte altre distribuzioni

2.1.2 Installazione dei Pacchetti

In sistemi basati su RPM (come Fedora e RHEL), l'installazione avviene tramite gestori di pacchetti come `yum` o `dnf`. I pacchetti sono spesso organizzati in gruppi funzionali per facilitare l'installazione di servizi completi.

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

Listing 1: Esempio di installazione di un gruppo di pacchetti

2.1.3 Principali Categorie di Server

Server di Logging (rsyslog)

Il servizio rsyslog è fondamentale per raccogliere messaggi di log da vari componenti del sistema. Può funzionare sia localmente che come server di logging remoto centralizzato.

```
# dnf install rsyslog
# systemctl enable rsyslog
# systemctl start rsyslog
```

Listing 2: Installazione e avvio di rsyslog

Server di Stampa (CUPS)

Il Common UNIX Printing Service fornisce funzionalità di server di stampa, permettendo la condivisione di stampanti in rete.

```
# dnf install cups system-config-printer
# systemctl enable cups
# systemctl start cups
```

Listing 3: Installazione CUPS

Server Web (Apache/Nginx)

Il server web Apache (pacchetto httpd) è il più diffuso per servire contenuti HTTP. Include moduli per vari linguaggi di programmazione e funzionalità avanzate.

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

Listing 4: Installazione Apache con moduli

Server FTP (vsftpd)

Il Very Secure FTP daemon è il server FTP predefinito in molte distribuzioni, scelto per le sue caratteristiche di sicurezza.

```
# dnf install vsftpd
# systemctl enable vsftpd
# systemctl start vsftpd
```

Listing 5: Installazione vsftpd

Server di File Windows (Samba)

Samba permette ai sistemi Linux di condividere file e stampanti con client Windows utilizzando il protocollo SMB/CIFS.

```
# dnf install samba samba-client
# systemctl enable smb nmb
# systemctl start smb nmb
```

Listing 6: Installazione Samba

Server NFS

Network File System è lo standard Linux/UNIX per condividere directory tra sistemi sulla rete.

```
# dnf install nfs-utils
# systemctl enable nfs-server
# systemctl start nfs-server
```

Listing 7: Installazione NFS

Server di Posta (Postfix/Sendmail)

I server di posta elettronica, o Mail Transport Agent (MTA), gestiscono l'invio e la ricezione di email.

```
# dnf install postfix dovecot
# systemctl enable postfix dovecot
# systemctl start postfix dovecot
```

Listing 8: Installazione Postfix

Server di Directory (LDAP)

I servizi di directory forniscono autenticazione e autorizzazione centralizzate.

```
# dnf install openldap-servers openldap-clients
# systemctl enable slapd
# systemctl start slapd
```

Listing 9: Installazione OpenLDAP

Server DNS (BIND)

Berkeley Internet Name Domain fornisce servizi di risoluzione nomi DNS.

```
# dnf install bind bind-utils
# systemctl enable named
# systemctl start named
```

Listing 10: Installazione BIND

Server NTP (chrony)

Network Time Protocol sincronizza l'orologio di sistema con server di tempo pubblici o privati.

```
# dnf install chrony
# systemctl enable chronyd
# systemctl start chronyd
```

Listing 11: Installazione chrony

Server Database (PostgreSQL/MySQL/MariaDB)

I database relazionali sono fondamentali per molte applicazioni web e aziendali.

```
# dnf install mariadb-server
# systemctl enable mariadb
# systemctl start mariadb
# mysql_secure_installation
```

Listing 12: Installazione MariaDB

2.2 Fase 2: Configurazione del Server

Dopo l'installazione, i pacchetti server richiedono configurazione specifica per funzionare correttamente e in modo sicuro.

2.2.1 File di Configurazione

La maggior parte dei servizi Linux utilizza file di configurazione in formato testo semplice, tipicamente ubicati nella directory `/etc` o sue sottodirectory.

Utilizzo di vim per l'Editing

Si raccomanda l'uso di `vim` invece di `vi` per modificare i file di configurazione. Vim fornisce evidenziazione della sintassi che aiuta a identificare errori:

```
# vim /etc/httpd/conf/httpd.conf
```

Vim cambia il colore del testo quando rileva errori di sintassi o opzioni non valide.

2.2.2 Struttura dei File di Configurazione

Molti servizi moderni utilizzano una struttura modulare:

- File di configurazione principale (es. `/etc/httpd/conf/httpd.conf`)
- Directory per configurazioni aggiuntive (es. `/etc/httpd/conf.d/`)
- File `.conf` nella directory vengono inclusi automaticamente

Questo approccio permette ai pacchetti aggiuntivi di inserire le proprie configurazioni senza modificare il file principale.

```
/etc/httpd/
conf/
   --- httpd.conf           # Configurazione principale
---conf.d/
   --- ssl.conf             # Configurazione SSL
   --- php.conf             # Configurazione PHP
     ---welcome.conf         # Pagina di benvenuto
---- conf.modules.d/          # Moduli Apache
```

Listing 13: Esempio di struttura Apache

2.2.3 Configurazione Predefinita di Sicurezza

I pacchetti server in RHEL e Fedora sono installati con configurazioni conservative che privilegiano la sicurezza rispetto alla funzionalità completa immediata.

Servizi Limitati di Default

Alcuni servizi come mail server (postfix/sendmail) e DNS (bind) sono configurati per default ad ascoltare solo su localhost. Questo significa che:

- Il servizio è operativo ma non accessibile dall'esterno
- È necessaria configurazione manuale per renderlo pubblico
- Questa impostazione previene accessi non autorizzati accidentali

2.3 Fase 3: Avvio del Server

I servizi Linux possono essere gestiti attraverso sistemi di init diversi a seconda della distribuzione.

2.3.1 Sistemi di Gestione dei Servizi

systemd (Moderno)

Utilizzato in RHEL 7+, Fedora, Ubuntu 16.04+:

```
# Verifica stato servizio
systemctl status httpd

# Avvia servizio
systemctl start httpd

# Ferma servizio
systemctl stop httpd

# Riavvia servizio
```

```
systemctl restart httpd

# Ricarica configurazione senza riavvio
systemctl reload httpd

# Abilita avvio automatico
systemctl enable httpd

# Disabilita avvio automatico
systemctl disable httpd

# Verifica se abilitato
systemctl is-enabled httpd
```

Listing 14: Comandi systemd

SysVinit (Legacy)

Utilizzato in RHEL 6 e sistemi più vecchi:

```
# Verifica stato
service httpd status

# Avvia servizio
service httpd start

# Ferma servizio
service httpd stop

# Configura avvio automatico
chkconfig httpd on

# Verifica configurazione avvio
chkconfig --list httpd
```

Listing 15: Comandi SysVinit

2.3.2 Processi Daemon

La maggior parte dei servizi è implementata come processi daemon con caratteristiche specifiche:

Permessi Utente e Gruppo

I daemon spesso vengono eseguiti con utenti dedicati non-root per limitare i danni in caso di compromissione:

```
# Apache gira come utente apache
ps aux | grep httpd
apache   1234  ... /usr/sbin/httpd

# NTP gira come utente ntp
ps aux | grep ntpd
ntp      5678  ... /usr/sbin/ntpd
```

Listing 16: Esempi di utenti daemon

File di Configurazione Daemon

Molti servizi hanno file in `/etc/sysconfig/` per passare argomenti al daemon:

```
# File: /etc/sysconfig/rsyslog
# Opzioni da passare a rsyslogd
SYSLOGD_OPTIONS="-m 0 -r"
# -m 0  : disabilita timestamp marks
# -r    : accetta log remoti
```

Listing 17: Esempio /etc/sysconfig/rsyslog

Numeri di Porta

I servizi di rete utilizzano porte specifiche per comunicare. Le porte standard sono definite in `/etc/services`:

```
# Mostra porte TCP/UDP in ascolto
netstat -tulpn

# Alternativa moderna
ss -tulpn

# Output esempio:
# tcp    0   0 0.0.0.0:22     0.0.0.0:*   LISTEN   1234/sshd
# tcp    0   0 0.0.0.0:80     0.0.0.0:*   LISTEN   5678/httpd
```

Listing 18: Visualizzazione porte in uso

Sicurezza delle Porte

Quando si cambia la porta di un servizio:

- Aggiornare le regole del firewall
- Verificare le policy SELinux
- Documentare la modifica
- Comunicare il cambio agli utenti

2.3.3 Servizi Non-Daemon

Non tutti i servizi vengono eseguiti come daemon:

Servizi on-demand

Alcuni servizi vengono avviati solo quando necessario tramite socket systemd o xinetd (legacy).

Servizi one-shot

Alcuni servizi vengono eseguiti una sola volta all'avvio e poi terminano.

Task schedulati

Servizi gestiti da cron vengono eseguiti a intervalli predefiniti.

3 Sicurezza del Server

La sicurezza è un aspetto fondamentale dell'amministrazione server che richiede un approccio multi-livello e continuo.

3.1 Protezione tramite Password

Le password rappresentano la prima linea di difesa nella sicurezza di un sistema Linux.

3.1.1 Best Practices per le Password

1. **Complessità**: Le password devono essere sufficientemente complesse
   - Lunghezza minima di 12-14 caratteri
   - Combinazione di maiuscole, minuscole, numeri e simboli
   - Evitare parole del dizionario e informazioni personali
2. **Rotazione**: Cambiare le password regolarmente
3. **Unicità**: Non riutilizzare password su sistemi diversi
4. **Storage sicuro**: Utilizzare password manager

3.1.2 Disabilitazione Login Root Diretto

Una best practice fondamentale è impedire il login diretto come root:

```
# File: /etc/ssh/sshd_config
PermitRootLogin no

# Riavvia SSH per applicare
systemctl restart sshd
```

Listing 19: Configurazione accesso root

Dopo questa modifica, gli utenti devono:

1. Effettuare login con account utente normale
2. Utilizzare `su` o `sudo` per ottenere privilegi root

3.1.3 PAM (Pluggable Authentication Module)

PAM fornisce un framework flessibile per l'autenticazione:

```
# File: /etc/pam.d/password-auth
auth        required      pam_faillock.so preauth
auth        sufficient    pam_unix.so try_first_pass
auth        required      pam_faillock.so authfail

# Blocca account dopo 5 tentativi falliti
account     required      pam_faillock.so
```

Listing 20: Esempio configurazione PAM

Funzionalità PAM includono:

- Limitazione tentativi di login
- Requisiti di complessità password
- Autenticazione a due fattori
- Logging degli accessi

3.1.4 Autenticazione tramite Chiave Pubblica

L'autenticazione basata su chiavi è più sicura delle password:

```
# Sul client, genera chiavi SSH
ssh-keygen -t rsa -b 4096

# Copia chiave pubblica sul server
ssh-copy-id user@server.example.com

# Ora puoi connetterti senza password
ssh user@server.example.com
```

Listing 21: Generazione coppia di chiavi

Vantaggi:

- Impossibile indovinare la chiave con brute force
- Può essere utilizzata senza password (con passphrase opzionale)
- Una chiave può autenticare su multipli server
- Facilita l'automazione (backup, script)

3.2 Firewall

I firewall sono essenziali per controllare il traffico di rete in ingresso e uscita.

3.2.1 iptables

iptables è il sistema di firewalling del kernel Linux:

```
# Visualizza regole correnti
iptables -L -v -n

# Permetti SSH (porta 22)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Permetti HTTP/HTTPS
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT

# Blocca tutto il resto
iptables -P INPUT DROP

# Salva regole
iptables-save > /etc/sysconfig/iptables
```

Listing 22: Comandi iptables base

3.2.2 firewalld

firewalld fornisce un'interfaccia dinamica per gestire iptables:

```
# Verifica stato
firewall-cmd --state

# Mostra zone attive
firewall-cmd --get-active-zones

# Permetti servizio HTTP
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https

# Permetti porta specifica
firewall-cmd --permanent --add-port=8080/tcp

# Ricarica configurazione
firewall-cmd --reload

# Lista servizi permessi
firewall-cmd --list-services
```

Listing 23: Comandi firewalld

Concetti chiave firewalld:

- **Zone**: Livelli di fiducia (public, trusted, dmz, etc.)
- **Servizi**: Gruppi predefiniti di porte
- **Permanent**: Cambiamenti persistono al riavvio
- **Runtime**: Cambiamenti temporanei

3.3 SELinux (Security-Enhanced Linux)

SELinux fornisce controllo di accesso obbligatorio (MAC) per proteggere il sistema.

3.3.1 Modalità SELinux

```
# Verifica stato corrente
getenforce

# Visualizza configurazione
sestatus
# Cambia modalita temporaneamente
setenforce 0  # Permissive
setenforce 1  # Enforcing

# Configurazione permanente
# File: /etc/selinux/config
SELINUX=enforcing  # o permissive, disabled
```

Listing 24: Gestione SELinux

Modalità operative:

- **Enforcing**: SELinux blocca azioni non autorizzate
- **Permissive**: SELinux registra violazioni ma non blocca
- **Disabled**: SELinux disattivato

3.3.2 Context e Policy

SELinux utilizza context per determinare l'accesso:

```
# Visualizza context di file
ls -Z /var/www/html/
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index
    .html

# Ripristina context corretti
restorecon -Rv /var/www/html/

# Cambia context manualmente
chcon -t httpd_sys_content_t /path/to/file

# Modifica permanente
semanage fcontext -a -t httpd_sys_content_t "/custom/web/path(/.*)?"
```

Listing 25: Gestione context SELinux

3.3.3 Boolean SELinux

I boolean permettono di modificare il comportamento di SELinux:

```
# Lista tutti i boolean
getsebool -a

# Verifica boolean specifico
getsebool httpd_can_network_connect

# Abilita boolean temporaneamente
setsebool httpd_can_network_connect on

# Abilita boolean permanentemente
setsebool -P httpd_can_network_connect on

# Boolean comuni per httpd
getsebool -a | grep httpd
```

Listing 26: Gestione boolean

3.3.4 Porte SELinux

SELinux controlla anche le porte utilizzabili dai servizi:

```
# Mostra porte associate a httpd
semanage port -l | grep http

# Permetti httpd su porta non-standard
semanage port -a -t http_port_t -p tcp 8080

```

```
# Rimuovi permesso porta
semanage port -d -t http_port_t -p tcp 8080
```

Listing 27: Gestione porte SELinux

3.3.5 Troubleshooting SELinux

```
# Installa strumenti di troubleshooting
dnf install setroubleshoot-server

# Analizza log SELinux
sealert -a /var/log/audit/audit.log

# Mostra messaggi recenti
sealert -l "*"

# Log audit in tempo reale
tail -f /var/log/audit/audit.log | grep denied
```

Listing 28: Debug problemi SELinux

Approccio Sistematico a SELinux

Quando si riscontrano problemi con SELinux:

1. Verificare i log con sealert
2. Controllare file contexts con ls -Z
3. Verificare boolean rilevanti
4. Controllare permessi porte se necessario
5. Non disabilitare SELinux senza prima capire il problema

3.4 TCP Wrappers

TCP Wrappers fornisce un livello addizionale di controllo accessi:

```
# File: /etc/hosts.allow
# Permetti SSH solo da rete locale
sshd: 192.168.1.0/255.255.255.0

# File: /etc/hosts.deny
# Blocca tutto il resto
sshd: ALL

# Formato generale
# servizio: client [: opzione : opzione ...]
```

Listing 29: Configurazione TCP Wrappers

Limitazioni TCP Wrappers

TCP Wrappers non è più considerato molto sicuro e il suo uso sta diminuendo. È preferibile utilizzare:

- firewalld o iptables per filtraggio rete
- SELinux per controllo accessi
- Configurazioni specifiche del servizio

3.5 Configurazioni di Sicurezza Specifiche

Molti servizi hanno opzioni di sicurezza nei propri file di configurazione:

```
# File: /etc/httpd/conf/httpd.conf

# Limita accesso per directory
<Directory "/var/www/html/private">
    Require ip 192.168.1.0/24
    Require user john jane
</Directory>

# Disabilita directory listing
Options -Indexes

# Previeni accesso a .htaccess
<FilesMatch "^\.ht">
    Require all denied
</FilesMatch>
```

Listing 30: Esempio sicurezza Apache

4 Monitoraggio del Server

Il monitoraggio continuo è essenziale per mantenere i server operativi e sicuri.

4.1 Logging di Sistema con rsyslog

rsyslog è il sistema di logging standard su Linux moderno.

4.1.1 Architettura rsyslog

```
# File: /etc/rsyslog.conf

#### MODULI ####
# Supporto log locali
module(load="imuxsock")

# Accesso al journal systemd
module(load="imjournal" StateFile="imjournal.state")

# Log kernel
#module(load="imklog")

# Ricezione log remoti UDP
#module(load="imudp")
#input(type="imudp" port="514")

# Ricezione log remoti TCP
#module(load="imtcp")
#input(type="imtcp" port="514")
```

Listing 31: File di configurazione rsyslog

4.1.2 Regole di Logging

```
#### RULES ####

# Log kernel alla console
#kern.*                         /dev/console

# Info e superiori (esclusi mail, authpriv, cron)
*.info;mail.none;authpriv.none;cron.none    /var/log/messages

# Log autenticazione
authpriv.*                      /var/log/secure

# Log email
mail.*                          -/var/log/maillog

# Log cron
cron.*                          /var/log/cron

# Log emergenze a tutti gli utenti
*.emerg                         :omusrmsg:*
```

Listing 32: Regole rsyslog

Formato regole: `facility.priority destination`

Facility comuni:

- `kern`: Kernel
- `mail`: Sistema mail
- `authpriv`: Autenticazione
- `cron`: Scheduler
- `daemon`: Daemon di sistema
- `*`: Tutte le facility

Priority (in ordine crescente):

- `debug`: Informazioni di debug
- `info`: Informazioni generali
- `notice`: Condizioni normali ma significative
- `warning/warn`: Condizioni di attenzione
- `err/error`: Errori
- `crit`: Condizioni critiche
- `alert`: Azione immediata richiesta
- `emerg/panic`: Sistema inutilizzabile

4.1.3 Analisi File di Log

```
# Formato tipico:
# DATA ORA HOST SERVIZIO[PID]: MESSAGGIO

Feb 25 11:04:32 server01 network: Bringing up interface eth0: succeeded
Feb 25 13:01:14 server01 vsftpd(pam_unix)[10565]: authentication failure
    ; user=chris
Feb 25 14:44:24 server01 su(pam_unix)[11439]: session opened for user
    root by chris(uid=500)
```

Listing 33: Formato messaggi log

4.1.4 Logging Remoto Centralizzato

Configurazione Client

```
# File: /etc/rsyslog.conf (sul client)

# Invia tutti i log al loghost
*.* @loghost.example.com        # UDP
# oppure
*.* @@loghost.example.com       # TCP

# Invia solo log specifici
*.info;mail.none;authpriv.none;cron.none @loghost.example.com
authpriv.* @loghost.example.com
```

```
mail.* @loghost.example.com
```

Listing 34: Invio log a server remoto

Configurazione Server (Loghost)

```
# File: /etc/rsyslog.conf (sul loghost)

# Abilita ricezione UDP
module(load="imudp")
input(type="imudp" port="514")

# Abilita ricezione TCP
module(load="imtcp")
input(type="imtcp" port="514")

# Riavvia servizio
systemctl restart rsyslog
```

Listing 35: Configurazione loghost

```
# Permetti porta 514 UDP/TCP
firewall-cmd --permanent --add-port=514/udp
firewall-cmd --permanent --add-port=514/tcp
firewall-cmd --reload

# Verifica porte in ascolto
netstat -tulpn | grep 514
```

Listing 36: Apertura firewall per logging remoto

Vantaggi Logging Centralizzato

- Un unico punto per revisionare log di multipli server
- Log preservati anche se un server viene compromesso
- Facilita correlazione eventi tra server
- Semplifica compliance e audit

Considerazioni di sicurezza:

- Log trasferiti in chiaro (può essere criptato con TLS)
- Loghost diventa target critico da proteggere
- Considerare utilizzo di loghost dedicato

4.2 logwatch

logwatch analizza automaticamente i log e invia report via email:

```
# Installazione
dnf install logwatch

# Il servizio parte automaticamente da cron
# File: /etc/cron.daily/0logwatch

# Configurazione locale
# File: /etc/logwatch/conf/logwatch.conf
MailTo = admin@example.com
```

```
Detail = High
Range = yesterday
Service = All
```

Listing 37: Installazione e configurazione logwatch

Configurazioni importanti:

- `MailTo`: Destinatario report
- `Detail`: Livello dettaglio (Low/Med/High)
- `Range`: Periodo analisi (Yesterday/Today/All)
- `Service`: Servizi da analizzare

```
# Leggi mail come root
mail

# Mostra messaggio
& 1

# Esci
& x
```

Listing 38: Visualizzazione report logwatch

4.3 System Activity Reporter (sar)

sar raccoglie e visualizza statistiche di sistema nel tempo.

4.3.1 Installazione e Attivazione

```
# Installazione
dnf install sysstat

# Abilitazione servizio
systemctl enable sysstat
systemctl start sysstat

# Il servizio raccoglie dati ogni 10 minuti
# File dati in /var/log/sa/
```

Listing 39: Setup sar

4.3.2 Utilizzo CPU

```
# Mostra utilizzo CPU da mezzanotte
sar -u

# Output:
# Linux 5.3.8-200.fc30.x86_64 (server01) 11/28/2019 _x86_64_ (4 CPU)
#
# 11:30:05 PM CPU  %user  %nice  %system  %iowait  %steal  %idle
# 11:40:06 PM all   0.90   0.00     1.81     1.44    0.28   95.57
# ...

```

```
# Mostra dati specifici giorno
sar -u -f /var/log/sa/sa15

# Report live: campiona ogni 2 sec, 5 volte
sar -u 2 5
```

Listing 40: Analisi CPU con sar

Metriche CPU:

- `%user`: Tempo in modalità utente
- `%system`: Tempo in modalità kernel
- `%iowait`: Attesa I/O disco
- `%idle`: CPU inattiva
- `%steal`: Tempo rubato (virtualizzazione)

4.3.3 Attività Disco

```
# Statistiche disco
sar -d

# Output mostra:
# DEV        tps   rkB/s   wkB/s   areq-sz   aqu-sz   await
# dev8-0   49.31 5663.94   50.38    115.89     0.03    1.00
```

Listing 41: Analisi disco con sar

Metriche disco:

- `tps`: Trasferimenti per secondo
- `rkB/s`: KB letti per secondo
- `wkB/s`: KB scritti per secondo
- `await`: Tempo medio attesa (ms)

4.3.4 Attività Rete

```
# Statistiche interfacce rete
sar -n DEV

# Live: ogni 5 sec, 2 volte
sar -n DEV 5 2

# Output:
# IFACE   rxpck/s   txpck/s   rxkB/s   txkB/s   rxcmp/s   txcmp/s
# eth0      125.3      98.7    256.4    189.2       0.0       0.0
# lo         45.2      45.2     12.3     12.3       0.0       0.0
```

Listing 42: Analisi rete con sar

4.3.5 Memoria

```
# Utilizzo memoria
sar -r

# Swap
sar -S

# Paging
sar -B
```

Listing 43: Analisi memoria con sar

4.4 Monitoraggio con Cockpit

Cockpit fornisce interfaccia web per monitoraggio real-time:

```
# Installazione
dnf install cockpit

# Avvio servizio
systemctl enable --now cockpit.socket

# Apri firewall
firewall-cmd --permanent --add-service=cockpit
firewall-cmd --reload

# Accesso via browser
# https://localhost:9090
# https://server-ip:9090
```

Listing 44: Installazione Cockpit

Funzionalità Cockpit:

- Grafici real-time CPU, memoria, disco, rete
- Gestione servizi systemd
- Aggiornamenti sistema
- Log e journal
- Gestione storage
- Configurazione rete
- Account utente
- Terminal integrato

5 Gestione Spazio su Disco

Il monitoraggio e la gestione dello spazio su disco sono cruciali per prevenire interruzioni del servizio.

5.1 Comando df

df mostra lo spazio disponibile sui filesystem montati:

```
# Formato standard (blocchi 1K)
df

# Output:
# Filesystem     1K-blocks     Used Available Use% Mounted on
# /dev/sda3       30645460  2958356  26130408  11% /
# /dev/sda2          46668     8340     35919  19% /boot

# Formato leggibile
df -h

# Output:
# Filesystem      Size  Used Avail Use% Mounted on
# /dev/sda3        29G  2.9G   24G  11% /
# /dev/sda2        46M  8.2M   25M  19% /boot

# Solo filesystem reali (escludi tmpfs, devtmpfs)
df -h -x tmpfs -x devtmpfs

# Mostra inodes invece di spazio
df -i

# Tipo filesystem specifico
df -t ext4
df -t xfs
```

Listing 45: Utilizzo df

5.2 Comando du

du calcola l'utilizzo disco per directory e file:

```
# Spazio utilizzato da directory
du /home/user

# Formato leggibile
du -h /home/user

# Output:
# 114K    /home/user/httpd/stuff
# 234K    /home/user/httpd
# 137K    /home/user/uucp/data
# 701K    /home/user/uucp
# 1.0M    /home/user

# Solo totale
du -sh /home/user
```

```
# 1.0M    /home/user

# Top 10 directory per dimensione
du -h /var | sort -hr | head -10

# Limita profondita ricerca
du -h --max-depth=2 /var

# Escludi directory
du -h --exclude="*.tmp" /var
```

Listing 46: Utilizzo du

5.3 Comando find per Consumo Disco

find può identificare file che consumano spazio secondo vari criteri:

```
# File di un utente specifico ordinati per dimensione
find / -xdev -user john -print | xargs ls -ldS > /tmp/john-files.txt

# File piu grandi di 100MB
find / -xdev -size +100M | xargs ls -ldS > /tmp/large-files.txt

# File piu grandi di 1GB
find / -xdev -size +1G -ls

# File modificati negli ultimi 7 giorni piu grandi di 50MB
find /var/log -mtime -7 -size +50M -ls

# File non acceduti da piu di 365 giorni
find /home -atime +365 -size +10M -ls

# Directory piu grandi
find / -xdev -type d -exec du -sh {} \; | sort -hr | head -20

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

# Cache vecchie
find ~/.cache -type f -mtime +90 -delete
```

Listing 47: Ricerca file per dimensione

Opzioni find utili:

- `-xdev`: Non attraversa filesystem diversi
- `-size +100M`: File > 100 megabyte
- `-mtime -7`: Modificati negli ultimi 7 giorni
- `-atime +365`: Non acceduti da oltre 365 giorni
- `-user name`: Proprietà utente specifico
- `-type f`: Solo file
- `-type d`: Solo directory

Pulizia Spazio Disco

Aree comuni dove recuperare spazio:

- `/var/log`: Log vecchi
- `/tmp`: File temporanei
- `/var/tmp`: File temporanei persistenti
- `/home/*/.cache`: Cache utenti
- `/var/cache/yum`: Cache package manager
- Vecchi kernel in /boot
- Core dumps
- Backup non necessari

5.4 Rotazione Log con logrotate

logrotate gestisce automaticamente la rotazione dei file di log:

```
# File principale: /etc/logrotate.conf
# Configurazioni per servizi: /etc/logrotate.d/

# Esempio: /etc/logrotate.d/httpd
/var/log/httpd/*log {
    daily
    rotate 52
    missingok
    notifempty
    sharedscripts
    compress
    delaycompress
    postrotate
        /bin/systemctl reload httpd.service > /dev/null 2>/dev/null ||
            true
    endscript
}
```

Listing 48: Configurazione logrotate

Opzioni logrotate:

- `daily/weekly/monthly`: Frequenza rotazione
- `rotate N`: Mantieni N copie
- `compress`: Comprimi log vecchi
- `delaycompress`: Non comprimere log più recente
- `missingok`: Non errore se log mancante
- `notifempty`: Non ruotare se vuoto
- `sharedscripts`: Esegui script una volta
- `postrotate`: Script dopo rotazione

6 Gestione Remota con SSH

Secure Shell (SSH) è lo standard de facto per l'accesso remoto sicuro ai sistemi Linux.

6.1 Architettura SSH

6.1.1 Componenti SSH

Server SSH

```
# RHEL/Fedora
dnf install openssh openssh-server openssh-clients

# Ubuntu
apt-get install openssh-server openssh-client

# Verifica installazione
rpm -qa | grep openssh
# o
dpkg -l | grep openssh
```

Listing 49: Pacchetti SSH

6.2 Configurazione Server SSH

6.2.1 Gestione Servizio sshd

```
# RHEL/Fedora
systemctl status sshd
systemctl start sshd
systemctl enable sshd

# Ubuntu
systemctl status ssh
systemctl start ssh
systemctl enable ssh

# Riavvio dopo modifiche configurazione
systemctl restart sshd
```

Listing 50: Gestione servizio sshd

6.2.2 File di Configurazione

```
# File: /etc/ssh/sshd_config

# Porta di ascolto
Port 22

# Indirizzo di ascolto
#ListenAddress 0.0.0.0
#ListenAddress ::

# Versione protocollo
Protocol 2

```

```
# Login root
PermitRootLogin no

# Autenticazione password
PasswordAuthentication yes

# Autenticazione chiave pubblica
PubkeyAuthentication yes

# File authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

# X11 forwarding
X11Forwarding yes

# Timeout
ClientAliveInterval 300
ClientAliveCountMax 2

# Limita utenti
AllowUsers user1 user2
#DenyUsers user3

# Limita gruppi
AllowGroups sshusers
#DenyGroups noremote

# Banner pre-login
#Banner /etc/ssh/banner

# Subsystem SFTP
Subsystem sftp /usr/libexec/openssh/sftp-server
```

Listing 51: Configurazione sshd

Sicurezza Configurazione SSH

Best practices:

- `PermitRootLogin no`: Sempre disabilitare
- Cambiare porta di default se esposto a internet
- Usare chiavi invece di password quando possibile
- Limitare utenti/gruppi con Allow/Deny
- Abilitare logging dettagliato
- Configurare timeout appropriati
- Disabilitare protocollo SSH-1 (solo SSH-2)

6.3 Client SSH

6.3.1 Login Remoto

```
# Login base
ssh user@hostname
ssh user@192.168.1.100

```

```
# Porta non-standard
ssh -p 2222 user@hostname

# Verbose (debugging)
ssh -v user@hostname
ssh -vv user@hostname  # Piu dettagli

# Prima connessione - verifica host key
ssh user@newhost
# The authenticity of host 'newhost (192.168.1.50)' can't be established.
# RSA key fingerprint is SHA256:abc123...
# Are you sure you want to continue connecting (yes/no)? yes
# Warning: Permanently added 'newhost' (RSA) to the list of known hosts.

# File known_hosts
cat ~/.ssh/known_hosts
```

Listing 52: Connessione SSH

6.3.2 Esecuzione Remota

```
# Esegui comando singolo
ssh user@host hostname
ssh user@host "df -h"

# Comando con pipe (quote necessarie)
ssh user@host "ps aux | grep httpd"

# Comando multipli
ssh user@host "cd /var/log && tail -20 messages"

# Variabili locali in comandi remoti
LOCAL_VAR="test"
ssh user@host "echo $LOCAL_VAR"  # Espanso localmente
ssh user@host 'echo $LOCAL_VAR'  # Espanso remotamente
```

Listing 53: Comandi remoti via SSH

6.3.3 X11 Forwarding

```
# Abilita X11 forwarding
ssh -X user@host

# Avvia applicazione grafica
ssh -X user@host gedit

# Applicazione in background
ssh -X user@host "gedit &"

# Multipli programmi grafici
ssh -X user@host
[remote]$ firefox &
[remote]$ gimp &
[remote]$ system-config-printer &
```

Listing 54: Applicazioni grafiche via SSH

Requisiti X11 forwarding:

- Server X in esecuzione localmente
- `X11Forwarding yes` in sshd_config
- Variabile $DISPLAY settata
- xauth installato

6.4 Trasferimento File SSH

6.4.1 scp - Secure Copy

```
# File locale -> remoto
scp file.txt user@host:/path/to/destination/

# File remoto -> locale
scp user@host:/path/to/file.txt /local/path/

# Directory ricorsiva
scp -r directory/ user@host:/path/

# Preserva attributi
scp -p file.txt user@host:/path/

# Porta non-standard
scp -P 2222 file.txt user@host:/path/

# Limita bandwidth (KB/s)
scp -l 1000 large-file.iso user@host:/path/

# Copia tra due host remoti
scp user1@host1:/file.txt user2@host2:/path/

# Multipli file
scp file1.txt file2.txt user@host:/path/

# Wildcard
scp *.log user@host:/var/log/backup/
```

Listing 55: Copia file con scp

6.4.2 rsync su SSH

rsync è superiore a scp per backup e sincronizzazione:

```
# Sincronizzazione base
rsync -avz /local/dir/ user@host:/remote/dir/

# Opzioni comuni
# -a: archive (ricorsivo, preserva tutto)
# -v: verbose
```

```
# -z: compressione
# -P: progress + partial
# -h: human-readable

# Dry-run (simulazione)
rsync -avzn /local/ user@host:/remote/

# Delete: sincronizzazione esatta
rsync -avz --delete /local/ user@host:/remote/

# Esclusioni
rsync -avz --exclude '*.tmp' --exclude '.git' /local/ user@host:/remote/

# Bandwidth limit
rsync -avz --bwlimit=1000 /local/ user@host:/remote/

# Backup incrementale
rsync -avz --backup --backup-dir=/backup/$(date +%Y%m%d) \
      /source/ user@host:/dest/

# Solo differenze
rsync -avzc /local/ user@host:/remote/
# -c: checksum invece di timestamp

# Progress dettagliato
rsync -avzh --progress /large-file.iso user@host:/path/
```

Listing 56: Sincronizzazione con rsync

Vantaggi rsync vs scp:

- Trasferisce solo differenze
- Riprende trasferimenti interrotti
- Preserva attributi file e link simbolici
- Supporta delete per mirror esatto
- Più efficiente per directory grandi

6.4.3 sftp - Secure FTP

```
# Connessione
sftp user@host

# Comandi sftp
sftp> ls                    # Lista directory remota
sftp> lls                   # Lista directory locale
sftp> pwd                   # Directory remota corrente
sftp> lpwd                  # Directory locale corrente
sftp> cd /path              # Cambia dir remota
sftp> lcd /path             # Cambia dir locale

# Download
sftp> get file.txt          # Scarica file
sftp> get -r directory/     # Scarica directory ricorsiva
```

```
sftp> mget *.log              # Download multipli

# Upload
sftp> put file.txt            # Carica file
sftp> put -r directory/       # Carica directory
sftp> mput *.pdf              # Upload multipli

# Gestione file
sftp> mkdir newdir            # Crea directory remota
sftp> rmdir olddir            # Rimuovi directory remota
sftp> rm file.txt             # Elimina file remoto
sftp> rename old.txt new.txt  # Rinomina file remoto

# Permessi
sftp> chmod 644 file.txt      # Cambia permessi
sftp> chown user file.txt     # Cambia proprietario

# Informazioni
sftp> df -h                   # Spazio disco remoto
sftp> !                       # Shell locale
sftp> !ls                     # Comando shell locale

# Uscita
sftp> exit
sftp> quit
sftp> bye
```

Listing 57: Sessione interattiva sftp

6.5 Autenticazione Basata su Chiavi

L'autenticazione con chiavi SSH elimina la necessità di password.

6.5.1 Generazione Chiavi

```
# Genera chiave RSA (4096 bit)
ssh-keygen -t rsa -b 4096

# Genera chiave (piu moderna)
ssh-keygen -t ed25519

# Con commento personalizzato
ssh-keygen -t rsa -b 4096 -C "work-laptop"

# Processo interattivo
# Generating public/private rsa key pair.
# Enter file in which to save the key (/home/user/.ssh/id_rsa): [ENTER]
# Enter passphrase (empty for no passphrase): [PASSWORD or ENTER]
# Enter same passphrase again: [PASSWORD or ENTER]
# Your identification has been saved in /home/user/.ssh/id_rsa
# Your public key has been saved in /home/user/.ssh/id_rsa.pub

# File generati
ls -l ~/.ssh/
# -rw-------  1 user user 3243 Nov 28 10:00 id_rsa          # Privata
# -rw-r--r--  1 user user  743 Nov 28 10:00 id_rsa.pub      # Pubblica
```

Listing 58: Creazione coppia chiavi SSH

6.5.2 Distribuzione Chiave Pubblica

```
# Metodo automatico (preferito)
ssh-copy-id user@host

# Con chiave specifica
ssh-copy-id -i ~/.ssh/id_ed25519.pub user@host

# Porta non-standard
ssh-copy-id -p 2222 user@host

# Metodo manuale
cat ~/.ssh/id_rsa.pub | ssh user@host "mkdir -p ~/.ssh && \
    cat >> ~/.ssh/authorized_keys && \
    chmod 700 ~/.ssh && \
    chmod 600 ~/.ssh/authorized_keys"

# Verifica permessi sul server
ssh user@host "ls -la ~/.ssh/"
# drwx------  2 user user 4096 Nov 28 10:05 .ssh
# -rw-------  1 user user  743 Nov 28 10:05 authorized_keys
```

Listing 59: Copia chiave su server

Permessi Critici

I permessi devono essere esatti per funzionare:

- `~/.ssh`: 700 (drwx——)
- `~/.ssh/authorized_keys`: 600 (-rw——-)
- `~/.ssh/id_rsa`: 600 (-rw——-)
- `~/.ssh/id_rsa.pub`: 644 (-rw-r–r–)

Se i permessi sono sbagliati, l'autenticazione fallisce silenziosamente.

6.5.3 Utilizzo Chiavi

```
# Login automatico (se chiave senza passphrase)
ssh user@host
# Nessuna password richiesta!

# Con chiave specifica
ssh -i ~/.ssh/id_work user@host

# Multiple chiavi per host diversi
# File: ~/.ssh/config
Host server1
    HostName 192.168.1.100
    User admin
    IdentityFile ~/.ssh/id_server1

Host server2
    HostName 192.168.1.200
```

```
    User root
    IdentityFile ~/.ssh/id_server2
    Port 2222

# Uso configurazione
ssh server1  # Usa automaticamente le impostazioni
```

Listing 60: Login con chiavi

6.5.4 Agent SSH

ssh-agent gestisce chiavi con passphrase:

```
# Avvia agent
eval $(ssh-agent)

# Aggiungi chiave
ssh-add ~/.ssh/id_rsa
# Enter passphrase: [inserisci una volta]

# Lista chiavi caricate
ssh-add -l

# Rimuovi tutte le chiavi
ssh-add -D

# Timeout automatico (secondi)
ssh-add -t 3600 ~/.ssh/id_rsa  # Valida per 1 ora

# Agent automatico in .bashrc
if [ -z "$SSH_AUTH_SOCK" ]; then
    eval $(ssh-agent)
    ssh-add ~/.ssh/id_rsa
fi
```

Listing 61: Uso ssh-agent

6.5.5 Disabilitazione Password

```
# File: /etc/ssh/sshd_config
PasswordAuthentication no
ChallengeResponseAuthentication no
PubkeyAuthentication yes

# Riavvia sshd
systemctl restart sshd

# Ora solo chiavi funzionano
ssh user@host
# Permission denied (publickey,gssapi-keyex,gssapi-with-mic)
```

Listing 62: Forzare solo autenticazione chiave

6.6 Configurazione Client SSH

```
# File: ~/.ssh/config

# Configurazione globale
Host *
    ServerAliveInterval 60
    ServerAliveCountMax 3
    Compression yes

# Server specifico
Host prod
    HostName production.example.com
    User deploy
    Port 2222
    IdentityFile ~/.ssh/id_production
    ForwardAgent yes
    LocalForward 8080 localhost:80

# Jump host (bastion)
Host internal-server
    HostName 10.0.1.100
    User admin
    ProxyJump bastion.example.com

# Wildcard
Host *.dev.local
    User developer
    IdentityFile ~/.ssh/id_dev
    StrictHostKeyChecking no
```

Listing 63: File di configurazione client

Opzioni utili:

- `ServerAliveInterval`: Keep-alive
- `Compression`: Comprimi traffico
- `ForwardAgent`: Inoltra agent SSH
- `LocalForward`: Port forwarding locale
- `RemoteForward`: Port forwarding remoto
- `ProxyJump`: SSH attraverso bastion
- `StrictHostKeyChecking`: Controllo host key

7 Gestione Server in Enterprise

In ambienti enterprise con numerosi server, l'amministrazione manuale diventa impraticabile. Sono necessari approcci automatizzati e scalabili.

7.1 Deployment Automatizzato

7.1.1 PXE Boot

PXE (Preboot Execution Environment) permette l'installazione di sistemi via rete:

1. Client si avvia via rete (boot PXE)
2. Riceve indirizzo IP via DHCP
3. Download bootloader da server TFTP
4. Caricamento kernel e initrd
5. Avvio installazione automatica

Vantaggi:

- Installazioni senza intervento umano
- Installazioni multiple simultanee
- Configurazione standardizzata
- Risparmio tempo e riduzione errori

7.1.2 Kickstart (RHEL/Fedora)

File kickstart definisce installazione automatizzata:

```
# File: ks.cfg

# Installazione automatica
install
text

# Lingua e tastiera
lang it_IT.UTF-8
keyboard it

# Timezone
timezone Europe/Rome

# Root password
rootpw --iscrypted $6$encrypted_password

# Partizioni
clearpart --all --initlabel
part /boot --fstype=xfs --size=500
part pv.01 --size=1 --grow
volgroup vg00 pv.01
logvol / --fstype=xfs --name=root --vgname=vg00 --size=10240
logvol /home --fstype=xfs --name=home --vgname=vg00 --size=5120
logvol swap --name=swap --vgname=vg00 --size=2048
```

```
25
26 # Rete
27 network --bootproto=dhcp --device=eth0 --onboot=yes
28
29 # Firewall
30 firewall --enabled --service=ssh
31
32 # SELinux
33 selinux --enforcing
34
35 # Pacchetti
36 %packages
37 @core
38 @base
39 openssh-server
40 vim
41 %end
42
43 # Post-installazione
44 %post
45 # Aggiorna sistema
46 yum -y update
47
48 # Configura SSH
49 sed -i 's/#PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
50
51 # Aggiungi utente
52 useradd -m -G wheel admin
53 echo 'admin:password' | chpasswd
54 %end
```

Listing 64: Esempio kickstart

7.2 Sistemi Generici

7.2.1 Approccio Immutabile

Host system generici e immutabili semplificano gestione:

- Sistema base minimale identico
- Applicazioni containerizzate
- Configurazione via cloud-init
- Aggiornamenti atomici
- Rollback facile

7.2.2 Containerizzazione

```
1 # Esegui applicazione con dipendenze
2 podman run -d --name webapp \
3     -p 8080:80 \
4     -v /data:/var/www/html \
5     webapp-image:latest
6
```

```
# Container porta tutto il necessario
# Host rimane pulito
# Rimozione semplice
podman rm -f webapp
```

Listing 65: Applicazioni in container

Vantaggi:

- Isolamento applicazioni
- Dipendenze self-contained
- Portabilità
- Versioning semplice
- Deployment rapido

7.3 Separazione Management/Worker

7.3.1 Architettura

Nodi Management (Control Plane)

- Gestiscono cluster
- Schedulano workload
- Monitorano stato
- API endpoints
- Storage metadata

Nodi Worker

- Eseguono workload
- Report a management
- Intercambiabili
- Scalabili orizzontalmente

7.3.2 Platform Examples

OpenStack

- Controller nodes: API, database, message queue
- Compute nodes: Macchine virtuali
- Storage nodes: Block/object storage
- Network nodes: SDN, routing

Kubernetes/OpenShift

- Master nodes: API server, scheduler, controller
- Worker nodes: Container runtime, kubelet

- etcd nodes: Distributed config store

7.4 Configuration Management

7.4.1 Ansible

```
# File: webserver.yml
---
- name: Configura web server
  hosts: webservers
  become: yes

  tasks:
    - name: Installa Apache
      yum:
        name: httpd
        state: present

    - name: Copia configurazione
      template:
        src: httpd.conf.j2
        dest: /etc/httpd/conf/httpd.conf
      notify: restart httpd

    - name: Assicura servizio attivo
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Apri firewall
      firewalld:
        service: http
        permanent: yes
        state: enabled
        immediate: yes

  handlers:
    - name: restart httpd
      service:
        name: httpd
        state: restarted
```

Listing 66: Playbook Ansible esempio

7.4.2 Benefici Automazione

- **Consistenza**: Configurazione identica su tutti i nodi
- **Velocità**: Deploy simultaneo su centinaia di server
- **Documentazione**: Codice come documentazione
- **Versionamento**: Git per tracking modifiche
- **Testing**: Validazione prima del deploy
- **Rollback**: Ritorno a configurazione precedente

8 Best Practices e Raccomandazioni

8.1 Sicurezza

1. **Principio del minimo privilegio**
   - Utenti con solo permessi necessari
   - Servizi con utenti dedicati non-root
   - sudo configurato per specifici comandi
2. **Difesa in profondità**
   - Firewall configurato
   - SELinux in enforcing
   - Autenticazione multi-fattore
   - Encryption dati sensibili
3. **Aggiornamenti regolari**
   - Patch di sicurezza tempestive
   - Testing in ambiente non-production
   - Finestre di manutenzione pianificate
4. **Audit e compliance**
   - Log centralizzati
   - Review regolari
   - Retention policy
   - Alert su eventi critici

8.2 Monitoraggio

1. **Metriche essenziali**
   - CPU, memoria, disco, rete
   - Latenza servizi
   - Error rates
   - Disponibilità
2. **Alert intelligenti**
   - Threshold appropriati
   - Escalation policy
   - Evitare alert fatigue
   - Documentare run books
3. **Analisi proattiva**

- Trend analysis
- Capacity planning
- Performance baselines
- Predictive maintenance

8.3 Backup e Disaster Recovery

1. **Strategia 3-2-1**
   - 3 copie dati
   - 2 media diversi
   - 1 copia off-site
2. **Testing regolare**
   - Restore test mensili
   - DR drill trimestrali
   - Documentazione procedure
   - RTO/RPO definiti
3. **Automazione**
   - Backup schedulati
   - Verification automatica
   - Alert su failure
   - Retention policy

8.4 Documentazione

1. **Architecture diagrams**
   - Topologia rete
   - Flow dati
   - Dipendenze servizi
2. **Runbooks**
   - Procedure operative
   - Troubleshooting guides
   - Emergency procedures
3. **Change management**
   - Log modifiche
   - Approval process
   - Rollback plans

9 Conclusioni

L'amministrazione di server Linux richiede una combinazione di competenze tecniche, metodologie consolidate e attenzione continua ai dettagli. I principi fondamentali coperti in questo documento costituiscono la base per una gestione efficace e sicura dell'infrastruttura IT.

9.1 Punti Chiave

- **Processo Sistematico**: Installazione, configurazione, avvio, sicurezza e monitoraggio formano un ciclo continuo
- **Sicurezza Prioritaria**: Implementare difese multiple e mantenere vigilanza costante
- **Automazione Essenziale**: Scalare l'amministrazione attraverso strumenti e pratiche automatizzate
- **Monitoraggio Continuo**: Osservabilità del sistema per prevenire e risolvere problemi
- **Documentazione Vitale**: Mantenere documentazione accurata facilita troubleshooting e knowledge transfer

9.2 Evoluzione Continua

Il panorama IT è in costante evoluzione. Gli amministratori di sistema devono:

- Mantenersi aggiornati su nuove tecnologie
- Adottare best practices emergenti
- Bilanciare stabilità e innovazione
- Investire in formazione continua
- Partecipare a comunità professionali

9.3 Fondamenti Duraturi

Nonostante l'evoluzione tecnologica, alcuni principi rimangono costanti:

- Comprensione profonda dei sistemi
- Metodologia rigorosa nel problem-solving
- Attenzione alla sicurezza
- Commitment alla reliability
- Comunicazione efficace

Il successo nell'amministrazione di server Linux deriva dall'applicazione consistente di questi principi, adattati al contesto specifico e alle esigenze dell'organizzazione.

9.4 Risorse Aggiuntive

Per approfondire gli argomenti trattati:

- Documentazione ufficiale delle distribuzioni Linux
- Man pages (`man comando`)

- Community forums e mailing lists
- Certificazioni professionali (RHCSA, RHCE, LFCS)
- Libri di riferimento su amministrazione sistemistica
- Blog e conferenze tecniche

L'amministrazione server è tanto un'arte quanto una scienza, richiedendo esperienza pratica combinata con solide basi teoriche. Questo documento fornisce le fondamenta su cui costruire expertise attraverso pratica continua e apprendimento costante.