Red Hat
Enterprise Linux

DO124

Red Hat System Administration I

Francesco
Marchioni

Red Hat

Pausa 10 minuti
Dopo il lab

Red Hat

Presentazioni prima di tutto !

Francesco Marchioni - fmarchio@redhat.com
Red Hat Certified Architect (RHCA)

Red Hat

Benvenuti in Red Hat Enterprise Linux 10 !

- Aspetti fondamentali del Sistema Operativo
- Utilizzo della Command Line
- Installazione e gestione applicativi
- Gestione efficiente dell' Input/Output
- Sicurezza a prova di vulnerabilità
- Gestione rete e connessioni
- Utilizzo dei moduli di AI in modo efficiente

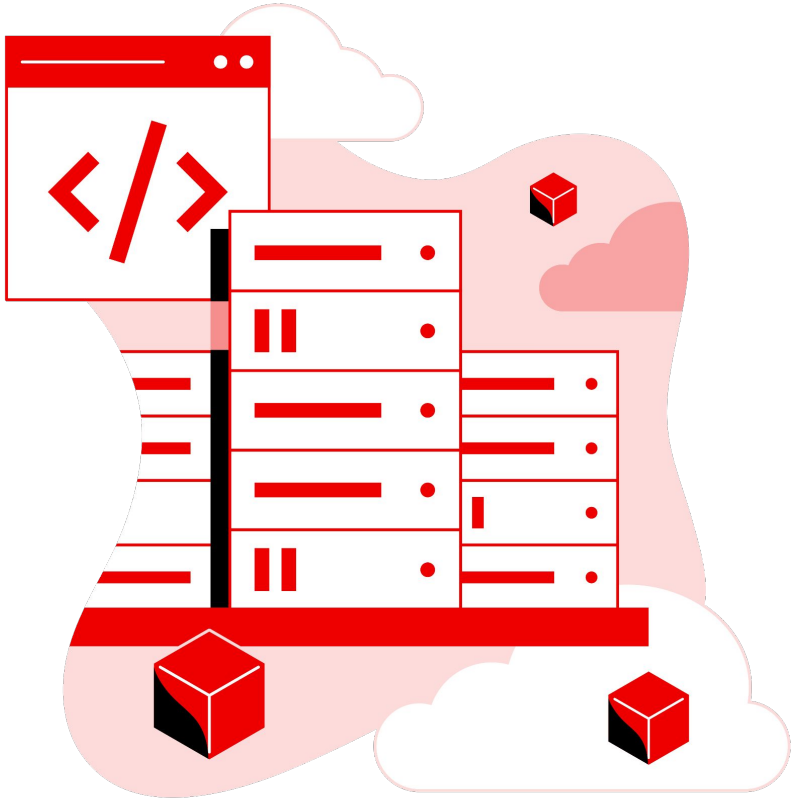
2

Red Hat Enterprise Linux

Accessing the Command Line

Introduzione alla Bash

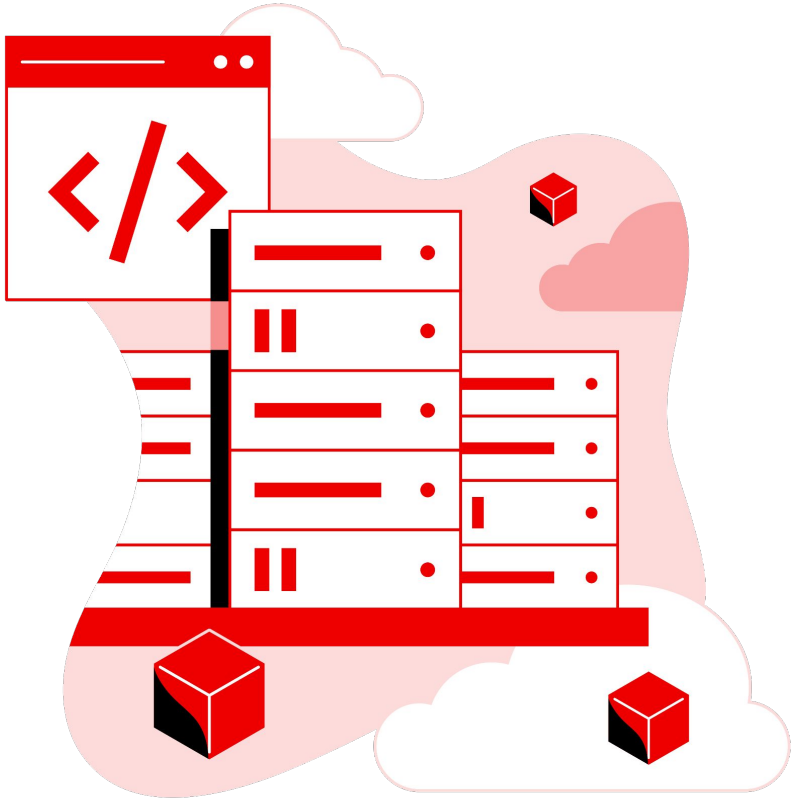
Cosa è la Bash shell ?



- Bash (Bourne Again SHell) è la shell predefinita in RHEL e in molte distribuzioni Linux.
- È un interprete di comandi che permette all'utente di interagire con il sistema operativo.
- Consente di:
  - Eseguire comandi e programmi,
  - Navigare nel file system,
  - Automatizzare operazioni tramite script.
- Combina funzionalità di shell interattiva e linguaggio di scripting.

Introduzione alla Bash

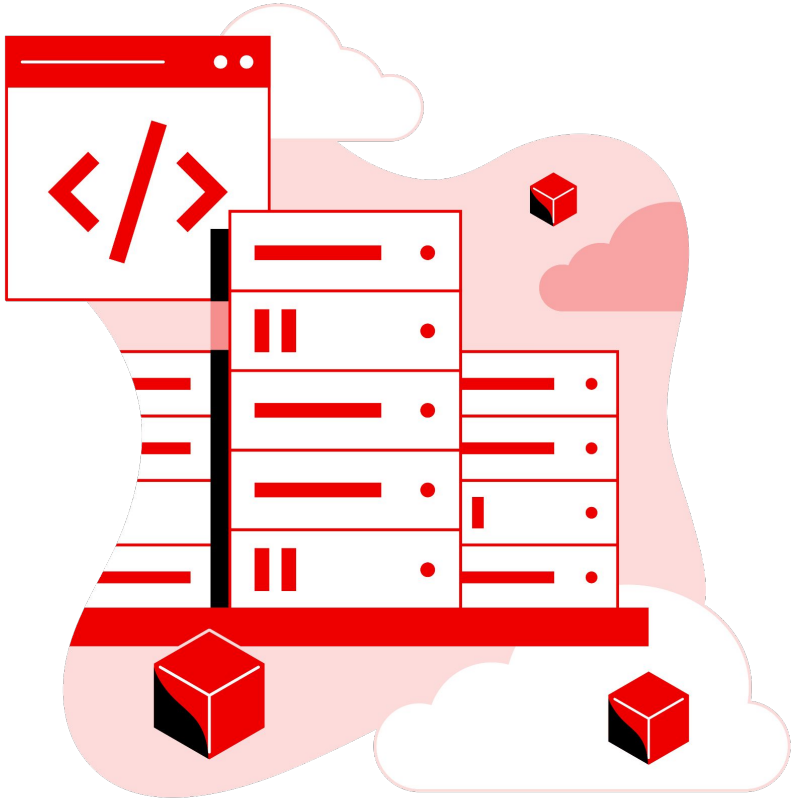
Caratteristiche principali



- Prompt: (es. [utente@host ~]$).
- Comandi di base:
  - `ls, cd, pwd, cp, mv, rm, cat, echo.`
- Redirezione e pipe:
  - > (output su file), < (input da file), | (passaggio output → input).
- Variabili: memorizzano valori temporanei (`VAR=value, $VAR`).
- Script Bash: file di testo con comandi sequenziali (`#!/bin/bash`).

Introduzione alla Bash

Composizione di un Comando



- `ls -al /home/student/directory`

Comando Opzione Argomento

Red Hat

Accesso a un Sistema Locale

L'accesso locale avviene tramite console fisica o virtuale. Dopo l'autenticazione con username e password, l'utente ottiene un prompt della shell per interagire con il sistema.

- **Console fisica**: tastiera e schermo collegati al computer.
- **Console virtuali** (tty1–tty6): sessioni indipendenti accessibili con Ctrl+Alt+Fn.
- **Login testuale**: inserimento di utente e password → shell prompt.
- **Login grafico**: interfaccia grafica avviata su una console virtuale (tipicamente tty1).
- **Server headless**: privi di tastiera e monitor; accesso via serial console o rete

Accesso a un Sistema Remoto

Quando il sistema non è fisicamente accessibile, l'accesso avviene tramite rete. In ambiente Linux lo strumento standard è SSH, che fornisce una connessione sicura e cifrata.

- **Accesso più comune**: SSH (Secure Shell).
  - Esempio: `ssh utente@server`
- **Connessione crittografata**: protegge password e dati da intercettazioni.
- **Autenticazione**:
  - con password, oppure
  - con coppia di chiavi (privata + pubblica).
- Ambienti cloud/VM spesso accettano solo autenticazione a chiave pubblica.

3

Red Hat Enterprise Linux
Getting Help from Local Documentation

Accesso alla documentazione (man)

- Documentazione ufficiale dei comandi e delle funzioni in Linux.
- Abbreviazione di manual page.
- Fornisce descrizione, sintassi, opzioni, esempi, riferimenti.
- Utilità per l'amministratore:
  - Accesso immediato alla documentazione offline
  - Standardizzato in tutto l'ecosistema Unix/Linux
  - Fonte primaria per troubleshooting e uso corretto dei comandi

Accesso alla documentazione

Comando “man”

Comando

`man <argomento> (es. man ssh )`
Pagine man conservate in /usr/share/man.
Divise in sezioni

- (1) User commands
- (2) system call,
- (3) Library functions
- (4) Special files
- (5) File formats
- (6) Games and screensavers
- (7) Conventions, and miscellaneous
- (8) System administration

Intestazioni

Ogni man page usa dei blocchi standard presenti in tutte le pagine

- NAME (nome e breve descrizione),
- SYNOPSIS (sintassi del comando),
- DESCRIPTION,
- OPTIONS,
- EXAMPLES,
- FILES, SEE ALSO.

Navigazione

E’ possibile navigare all’interno della man page usando caratteri o combinazioni di caratteri

- Spazio / PageDown → avanti,
- PageUp → indietro,
- /string → cerca testo,
- n / Shift+n → ripeti ricerca avanti/indietro,
- g / Shift+G → inizio/fine documento,
- q → esci.

4

Red Hat Enterprise Linux
Registering Systems for Red Hat Support

Come registrare un sistema RHEL ?

Registrare un sistema con rhc

```
$ rhc connect

$ rhc connect
--activation-key=<key_name>
```

rhc (Red Hat Connector) permette di registrare un sistema RHEL per la gestione centralizzata da console.redhat.com. È il metodo moderno per collegare sistemi on-premise o cloud ai servizi Red Hat.

- Collega il sistema a Red Hat Hybrid Cloud Console.
- Richiede un account Red Hat valido.
- Dopo la registrazione, il sistema è gestibile via console web.
- Consente accesso a:
  - Aggiornamenti software,
  - Insights per la sicurezza,
  - Automazione Ansible.

Come registrare un sistema RHEL ?

Registrare un sistema con subscription-manager

```
$ sudo subscription-manager
register --username <utente>
--password <password>


$ sudo subscription-manager
register --activationkey=<chiave>
--org=<org_id>
```

subscription-manager è l'interfaccia classica per registrare un sistema RHEL ai servizi Red Hat e associare una sottoscrizione per ricevere aggiornamenti e supporto.

- Richiede credenziali Red Hat o Activation Key.
- Dopo la registrazione:
  - abilitazione repository,
  - accesso a pacchetti e aggiornamenti,
  - gestione delle sottoscrizioni con subscription-manager list.

Red Hat

5

Red Hat Enterprise Linux
Getting AI-assisted Help

Command-line Assistant in RHEL 10

- Strumento opzionale basato su AI generativa (RHEL Lightspeed)
- Basato su IBM watsonx AI API large language model (LLM) e fornito as Service su sistemi registrati
- Interpreta domande in linguaggio naturale
- Suggerisce comandi, esempi, soluzioni, troubleshooting
- Riduce il tempo speso a cercare nella documentazione esterna
- Supporto a log analysis, troubleshooting, domande su RHEL

```
dnf install command-line-assistant
```

AI Command Line Assistant

Esempio di sessione

```
[student@servera ~]$ c chat "How do I change the prompt to show the IP Address?"
 + Asking RHEL Lightspeed
This feature uses AI technology. Do not include any personal information or other sensitive information in your input.
Interactions may be used to improve Red Hat's products or services.
──────────────────────────────────────────────────────────────────────────────

To change your shell prompt to display the IP address, you can use the PS1 environment variable in your shell configuration f
ile, typically .bashrc or .bash_profile. Here's a basic example of how to modify your prompt to show the IP address:

1. First, identify your IP address. You can do this by running the following command in your terminal:

   ``bash
   ip addr show
   `

   Look for the inet line under the network interface you are interested in (usually enp0s3 or ens3). The IP address will be
listed next to it.

2. Open your shell configuration file using a text editor. For Bash, this is usually .bashrc or .bash_profile. Use nano as th
e text editor for this example:

   `bash
   nano ~/.bashrc
   `

3. Add the following line to your configuration file, replacing <your_ip_address> with the actual IP address you identified i
n step 1:

   `bash
   export PS1='[\u@\h \W(\<IP\>)]\$ '
   `
```

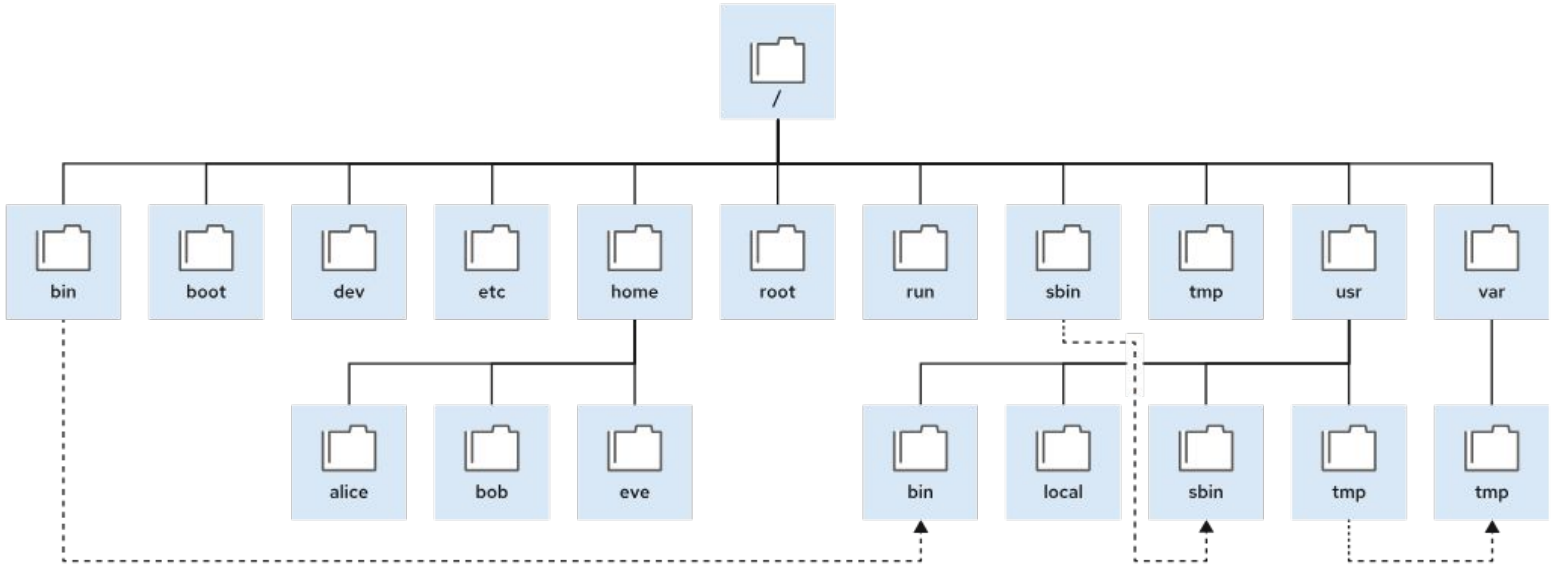
6

Red Hat Enterprise Linux

Navigating the File-system

Red Hat

Struttura del File System



Ogni sistema operativo Linux si basa su un File system gerarchico.

- Navigando dalla root (/) ci si sposta nell'alberatura del file system
- Ogni directory ospita files o cartelle dedicate a funzioni specifiche
- Alcuni files o cartelle sono persistenti. Altre sono periodicamente aggiornati/rimossi

Come navigare nel File System RHEL

Comandi Principali

```
cd /home/user/Documents
cd ./Documents

ls -l
ls -l ~

pwd

tree .
```

E' possibile navigare in un File System specificando il percorso assoluto (/var/www) oppure il percorso relativo (./www)

- `cd` : Sposta la directory corrente nel percorso relativo o assoluto.
- `pwd`: Mostra la directory corrente.
- `tree`: Mostra in formato albero la cartella corrente o altra
- `ls`: Mostra il contenuto della directory corrente o di altra indicata

7

Red Hat Enterprise Linux
Managing Files

Gestione dei files

In Linux i file si creano, modificano e rimuovono dalla shell con semplici comandi di base.

- **Creazione files**
  - `touch file.txt` → crea un file vuoto.
  - `nano file.txt o vi file.txt` → crea/modifica un file con editor.
  - `echo "testo" > file.txt` → crea file con contenuto.
- **Visualizzare contenuto**
  - `cat file.txt` → mostra intero file.
  - `less file.txt` → visualizza a schermate.
  - `head file.txt / tail file.txt` → prime o ultime righe.
- **Rimuovere file**
  - `rm file.txt` → elimina file.
  - `rm -f file.txt` → elimina forzatamente senza conferma.

Gestione directories

Le directory organizzano i file in una struttura gerarchica a cartelle.

- **Creare directory**
  - `mkdir nuova_dir` → crea una directory.
  - `mkdir -p dir1/dir2` → crea struttura annidata.
- **Navigare tra le directory**
  - `cd dir` → entra in directory.
  - `cd ..` → torna alla directory superiore.
  - `pwd` → mostra percorso attuale.
- **Rimuovere directory**
  - `rmdir dir_vuota` → elimina solo directory vuote.
  - `rm -r dir` → elimina directory e contenuto ricorsivamente .

Creazione Collegamenti - 1

Hard Links

```
# Crea Hard Link
ln file_originale file_hardlink

# Visualizza inodes
ls -il

# Numero hard links a files
ls -al
```

Un **hard link** è un collegamento diretto a un file esistente. Più hard link puntano allo stesso contenuto fisico sul disco.

- Entrambi i nomi (originale e hard link) condividono lo stesso inode.
- Il contenuto rimane accessibile finché esiste almeno un hard link.
- Cancellando un nome, gli altri restano validi.
- Limitazioni:
  - non funzionano su directory (per default),
  - non possono attraversare filesystem diversi.

Creazione Collegamenti - 2

Soft Links

```
# Crea Soft Link
ln -s file_originale
link_simbolico

lrwxrwxrwx 1 user user   12 Aug 26
file_link -> file_originale
```

Un **soft** link (o symlink) è un file speciale che contiene un "collegamento" al percorso di un altro file o directory.

- È uno "shortcut" al file o directory.
- Se il file originale viene cancellato → il link diventa rotto (dangling link).
- Può puntare anche a directory o ad altri filesystem.
- Identificabile facilmente con ls -l (compare l nei permessi):

File e String Expansions in Bash

Meccanismi di Shell expansion disponibili

```
# Pathname expansion
ls *.txt

# Brace expansion
echo file{1,2,3}.log

# Tilde expansion
cd ~

# Variable expansion
name="Francesco"
echo "Ciao, $name!"

# Command substitution
echo "Oggi è: $(date)"
```

La Bash shell supporta diversi meccanismi di espansione, che permettono di generare nomi di file, percorsi e stringhe in modo dinamico.

- **Pathname expansion**
  - Seleziona file con pattern matching.
- **Brace expansion**
  - Genera più stringhe.
- **Tilde expansion**
  - Espande ~ alla home dell'utente.
- **Variable expansion**
  - Inserisce il valore di una variabile.
- **Command substitution**
  - Usa l'output di un comando dentro un altro.

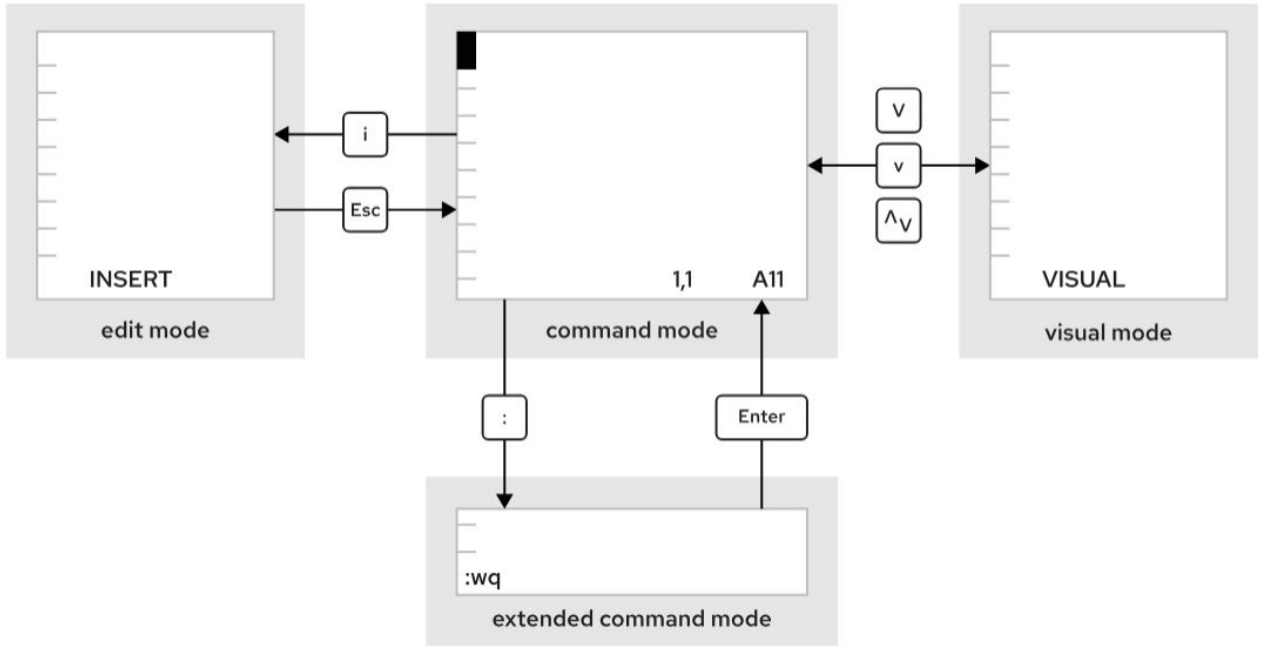
8

Red Hat Enterprise Linux
Editing Files

Red Hat

Editing dei files

L'editor vi



- Editor di testo modale presente su quasi tutti i sistemi Unix/Linux.
- Leggero, veloce, e ideale per modifiche rapide da terminale.
- Supporta diverse modalità di editing
- Espandibile tramite plugins
- Configurabile per un uso custom

9

Red Hat Enterprise Linux

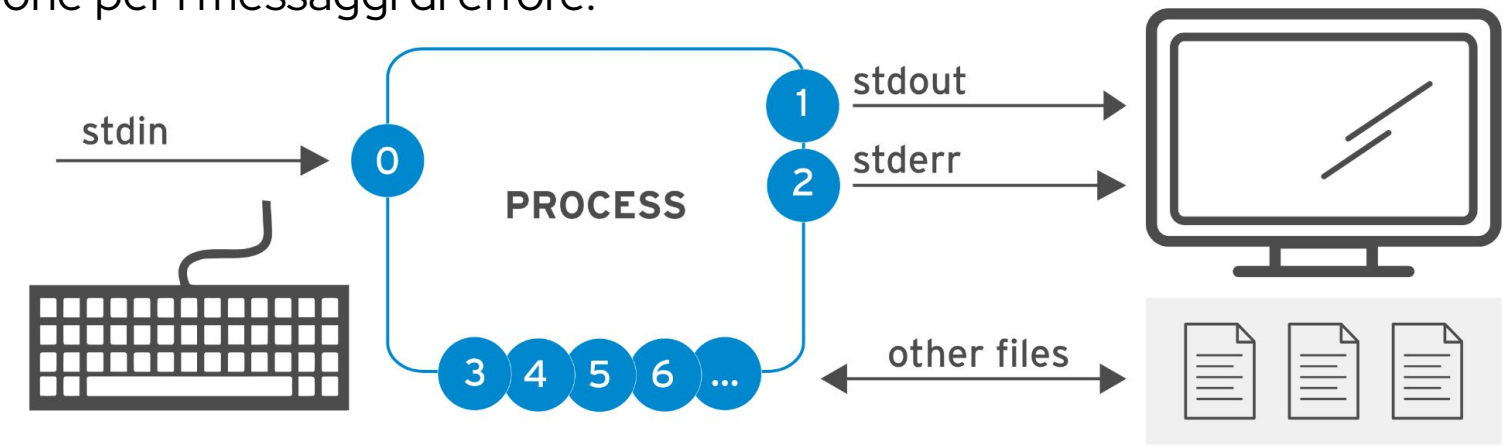
Redirecting Shell Input and Output

Red Hat

Gestione Input e Output

Ogni processo in Linux usa tre flussi predefiniti per comunicare con l'utente o con altri processi.

- **Standard Input (stdin)**
  - Sorgente di input (di default la tastiera).
- **Standard Output (stdout)**
  - Destinazione per i messaggi normali (di default il terminale).
- **Standard Error (stderr)**
  - Destinazione per i messaggi di errore.



Gestione Input e Output

Modifica origine e/o destinazione dei flussi di Input/Output

```
# Redirezione output
ls > elenco.txt

# Redirezione (append) output
ls >> elenco.txt

# Prende input
wc -l < elenco.txt

# Redirezione errori
./script.sh 2> errori.txt

# Redirezione stdout+errori
./script.sh &> errori.txt
```

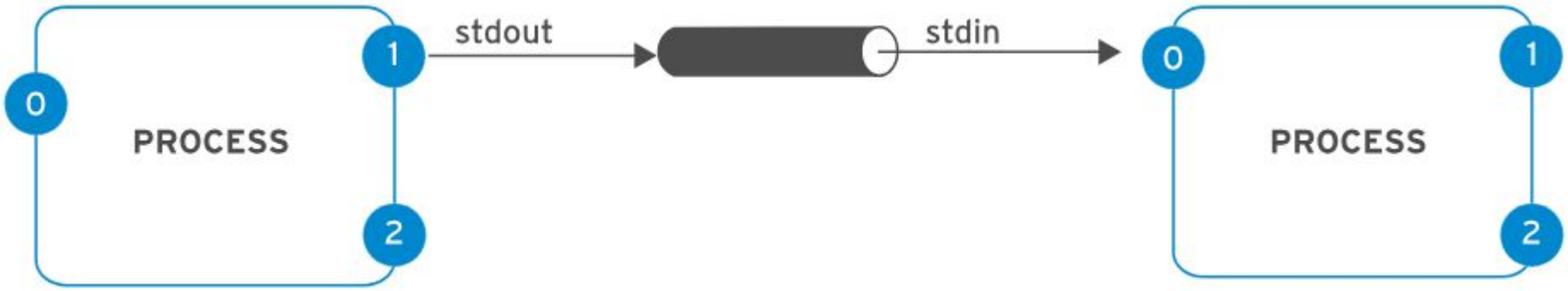
È possibile modificare la destinazione o la sorgente dei flussi tramite redirezione.

- **Redirezione output**
  - `>` → scrive su file
  - `>>` → aggiunge a file.
- **Redirezione input**
  - `<` → prende input da file.
- **Gestione errori**
  - `2>` → reindirizza errori.
  - `&>` → reindirizza stdout + stderr insieme.

Gestione Input e Output e Pipelines

Le pipeline permettono di collegare più comandi, passando l'output di uno come input del successivo. Sono fondamentali per costruire flussi di elaborazione dei dati direttamente da shell.

- Ogni comando nella pipeline legge da stdin e scrive su stdout.
- È possibile combinare pipeline con redirezioni (>, 2>) per salvare output o errori su file.
- Permette di creare flussi complessi senza file temporanei.
- **Es: ls -al | grep "myfile"**



10

Red Hat Enterprise Linux

Managing Users and Groups

Red Hat

Utenti e Gruppi

Ogni file, directory o processo è di proprietà di un utente e di un gruppo. Gli utenti rappresentano individui o servizi che interagiscono con il sistema. I gruppi sono collezioni di utenti.

- **Scopo**: L'uso di utenti e gruppi garantisce che solo le persone autorizzate possano accedere o modificare file e risorse di sistema.
- **Identificatori**: Ogni utente ha un User ID (UID) e ogni gruppo ha un Group ID (GID), entrambi numeri univoci
- **Permessi**: I permessi su file e directory sono definiti per il proprietario (user), il gruppo (group) e tutti gli altri (other).

Superuser : il commando “sudo”

Red Hat

Superuser 👑

L'utente root è l'amministratore del sistema con permessi illimitati. Per eseguire attività amministrative, gli utenti non-root devono ottenere i privilegi di superuser.

- **su (switch user):** Permette di cambiare utente. Richiede la password dell'utente a cui si vuole passare.
- **sudo (superuser do)**: Permette a un utente autorizzato di eseguire un comando come root. Richiede la password dell'utente corrente.
- **Vantaggi di sudo**: Non richiede la password di root, permette di tracciare chi ha eseguito un comando e riduce i rischi di errori fatali tipici dell'uso continuo del terminale come root.

Confronto tra modalità

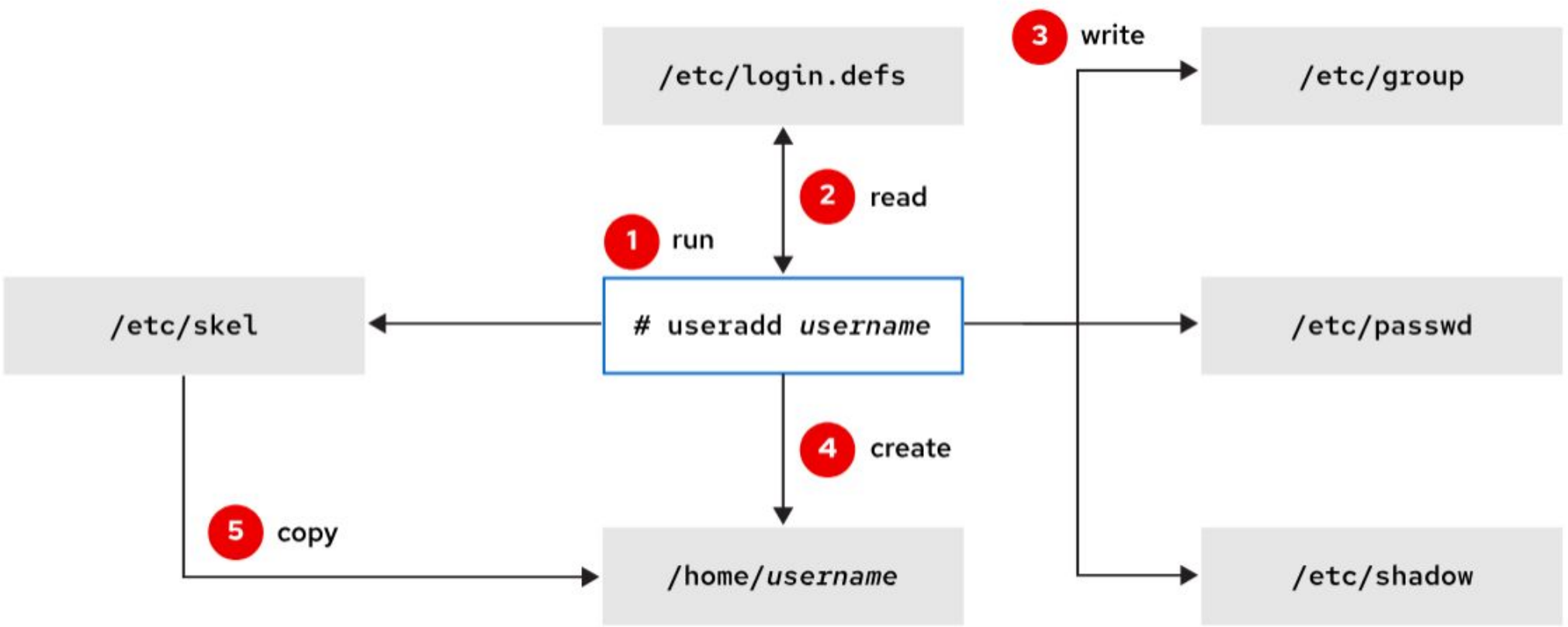
	su	su -	sudo
Switch User	Si	Si	Per singolo comando
Environment	Utente corrente	Nuovo utente	Utente corrente
Password da inserire	Del nuovo utente	Del nuovo utente	Utente corrente
Privilegi	Stessi del nuovo utente	Stessi del nuovo utente	Configurabili
Log comandi	Solo del comando su	Solo del comando su	Del comando privilegiato

Gestione Utenti

Gli account utente locali sono creati e gestiti direttamente sul sistema. I comandi per la gestione degli utenti:

- **useradd**: Crea un nuovo account utente. Si può specificare la directory home, la shell predefinita e l'appartenenza a gruppi.
- **usermod**: Modifica le proprietà di un utente esistente. Ad esempio, è possibile cambiare il nome utente, la directory home, l'ID utente (UID) o aggiungere l'utente a un nuovo gruppo.
- **userdel**: Elimina un account utente. L'opzione -r elimina anche la directory home e il contenuto dell'utente.

Gestione Utenti: Files utilizzati



Red Hat

Gestione Gruppi

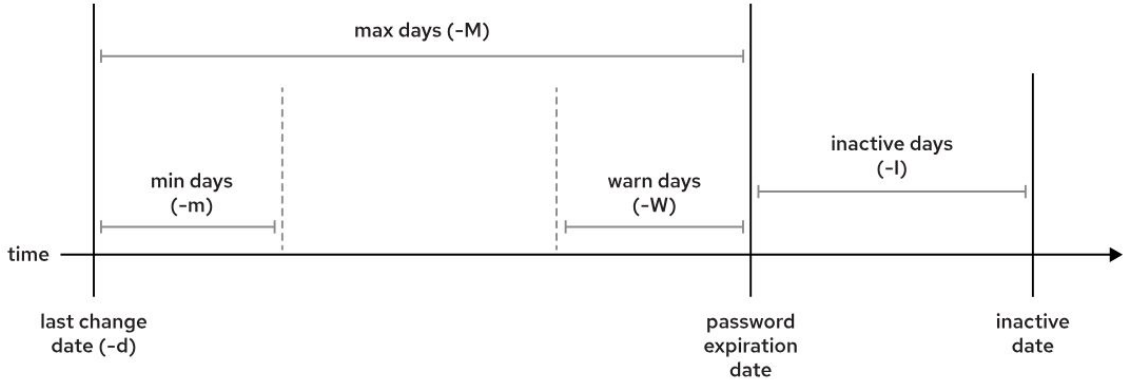
La gestione dei gruppi è essenziale per assegnare autorizzazioni a un insieme di utenti in modo efficiente.

- **groupadd**: Crea un nuovo gruppo.
- **groupmod**: Modifica le proprietà di un gruppo esistente. Ad esempio, per cambiare il nome del gruppo o il suo Group ID (GID).
- **groupdel**: Elimina un gruppo.
- **gpasswd**: Aggiunge o rimuove utenti da un gruppo esistente.
- **File chiave**: Le informazioni sui gruppi sono memorizzate in /etc/group.efinita e l'appartenenza a gruppi.

Gestione Passwords

Le password sono gestite con comandi specifici che permettono agli amministratori di impostarle, forzare il loro cambio o bloccare gli account.

- **passwd**: Imposta o modifica la password per un utente. Quando viene eseguito da un utente normale, cambia solo la propria password; se eseguito da root, può cambiare la password di qualsiasi utente.
- Bloccare/sbloccare: È possibile bloccare un account con **passwd -l** e sbloccarlo con **passwd -u**.
- Forzare il cambio: È possibile forzare il cambio della password al prossimo accesso con **chage**.



11

Red Hat Enterprise Linux

Controlling Access to Files

Red Hat

Introduzione ai Permessi

- I file in Linux hanno permessi che definiscono chi può leggere, scrivere o eseguire. Comprendere i permessi è fondamentale per la sicurezza del sistema.
- Ogni file ha proprietario, gruppo e altri utenti
- Tre tipi di permessi: r (read), w (write), x (execute)
- Applicati a: user (u), group (g), others (o)

```
ls -l file.txt
-rw-r--r-- 1 user group 1200 Sep 12 10:00 file.txt
```

Comprendere i permessi assegnati

- Il sistema assegna permessi ai file per controllare accesso e modifiche.

File:

r → leggere contenuto

w → modificare contenuto

x → eseguire come programma

```
ls -l file.txt
-rw-r--r-- 1 user group 1200 Sep 12 10:00 file.txt
```

Directory:

r → elencare file

w → creare/cancellare file

x → entrare nella directory

```
ls -l
drwxr-xr-x. 1 user group 1200 Sep 12 10:00 dir1
```

Controllo ownership files e directory

- È importante verificare i permessi per capire chi può fare cosa.
- Comando principale: **ls -l**
- Mostra tipo, permessi, proprietario, gruppo, dimensione, data
- **ls -ld dir** mostra i permessi della directory stessa

```
ls -ld /etc
drwxr-xr-x. 141 root root 8192 Sep 13 09:00 /etc
```

Modifica permessi di files e directory

- Ogni file e directory ha permessi di accesso che determinano chi può leggere, scrivere o eseguire. Il comando principale per modificarli è chmod (change mode).
- Puoi modificare i permessi solo se sei il proprietario del file o hai privilegi di root.
- Due Metodi di Modifica:
  - **Simbolico**: usa lettere e operatori per aggiungere o rimuovere permessi.
  - **Ottale**: usa numeri per rappresentare combinazioni di permessi.

Modifica Permessi con metodo Simbolico

Il metodo simbolico usa lettere per definire permessi.

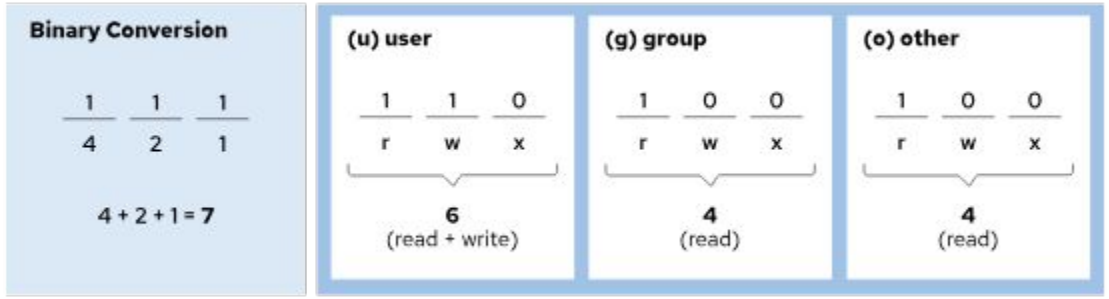
- Utenti: u (user), g (group), o (others), a (all)
- Permessi: r, w, x

```
chmod a+r file.txt        # Aggiunge lettura a tutti
chmod u+x script.sh       # Aggiunge esecuzione all'utente
chmod o-r file.txt        # Rimuove lettura agli altri
chmod a=r file.txt        # Imposta solo lettura per tutti
```

Modifica Permessi con metodo Ottale

- Il metodo ottale usa numeri per rappresentare i permessi.
- r=4, w=2, x=1
- Somma i valori per ogni categoria

```
chmod 644 file.txt   # rw-r--r--
chmod 755 script.sh  # rwxr-xr-x
```



Red Hat

Metodo Simbolico vs Ottale

- 🛠️ Tipo di shell o script: Nei Makefile o negli script di provisioning, la leggibilità può essere cruciale.
- 🎯 Precisione richiesta: Vuoi modificare un singolo permesso o impostare tutto da zero?
- 👩‍💻 Preferenze personali: Alcuni trovano il metodo simbolico più leggibile, altri preferiscono la sintesi dell'ottale.

Vuoi aggiungere/rimuovere un permesso specifico	Simbolico
Vuoi impostare tutti i permessi da zero	Ottale
Vuoi replicare permessi standard (es. 644, 755)	Ottale
Vuoi modificare solo il gruppo o altri utenti	Simbolico

Come cambiare l'ownership o gruppo

Con chown e chgrp puoi cambiare proprietario o gruppo.

- chown user file → cambia proprietario
- chown user:group file → cambia proprietario e gruppo
- chgrp group file → cambia solo gruppo

```
#  Il file report.txt avrà come proprietario maria e come gruppo mktg.
chown maria:mktg report.txt

# Il file avrà come proprietario maria
chown maria report.txt

# Il gruppo developers sarà proprietario del file code.py
chgrp developers code.py
```

Permessi speciali

Oltre a rwx esistono permessi speciali.

- SUID (4xxx): esegue un file con i permessi del proprietario
- SGID (2xxx): esegue con i permessi del gruppo o assegna gruppo ai file creati in dir
- Sticky bit (1xxx): su directory, solo owner può cancellare i propri file

```
chmod 4755 /usr/bin/program   # SUID
chmod 2775 /shared            # SGID
chmod 1777 /tmp               # sticky
```

Default Permissions

La umask definisce i permessi di default per nuovi file e directory.

- Default: file 666 - umask, dir 777 - umask
- Tipica umask: 0022 → file 644, dir 755
- Il file /etc/profile imposta la umask per tutti gli utenti al login.

```
umask          # mostra valore attuale
umask 0007     # nuovi file 660, dir 770
```

Vero o falso ?

```
drwxrwxr-x.  2 database1 consultant1   4096 Mar  4 10:23 .
drwxr-xr-x. 10 root      root          4096 Mar  1 17:34 ..

-rw-rw-r--.  1 operator1 operator1     1024 Mar  4 11:02 app1.log
-rw-r--rw-.  1 operator1 consultant1   3144 Mar  4 11:02 app2.log
-rw-rw-r--.  1 database1 consultant1  10234 Mar  4 10:14 db1.conf
-rw-r-----.  1 database1 consultant1   2048 Mar  4 10:18 db2.conf
```

User	Group memberships
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

operator1 è l'unico utente che può cambiare il file app1.log file

VERO !

Vero o falso ?

```
drwxrwxr-x.  2 database1 consultant1   4096 Mar  4 10:23 .
drwxr-xr-x. 10 root      root          4096 Mar  1 17:34 ..

-rw-rw-r--.  1 operator1 operator1     1024 Mar  4 11:02 app1.log
-rw-r--rw-.  1 operator1 consultant1   3144 Mar  4 11:02 app2.log
-rw-rw-r--.  1 database1 consultant1  10234 Mar  4 10:14 db1.conf
-rw-r-----.  1 database1 consultant1   2048 Mar  4 10:18 db2.conf
```

User	Group memberships
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

L'utente database2 non può modificare il file app2.log .

FALSO!

Vero o falso ?

```
drwxrwxr-x.  2 database1 consultant1   4096 Mar  4 10:23 .
drwxr-xr-x. 10 root      root          4096 Mar  1 17:34 ..

-rw-rw-r--.  1 operator1 operator1     1024 Mar  4 11:02 app1.log
-rw-r--rw-.  1 operator1 consultant1   3144 Mar  4 11:02 app2.log
-rw-rw-r--.  1 database1 consultant1  10234 Mar  4 10:14 db1.conf
-rw-r-----.  1 database1 consultant1   2048 Mar  4 10:18 db2.conf
```

User	Group memberships
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

L'utente database1 può visualizzare il contenuto del file app2.log ma non può modificarlo.

VERO!

Vero o falso ?

```
drwxrwxr-x.  2 database1 consultant1   4096 Mar  4 10:23 .
drwxr-xr-x. 10 root      root          4096 Mar  1 17:34 ..

-rw-rw-r--.  1 operator1 operator1     1024 Mar  4 11:02 app1.log
-rw-r--rw-.  1 operator1 consultant1   3144 Mar  4 11:02 app2.log
-rw-rw-r--.  1 database1 consultant1  10234 Mar  4 10:14 db1.conf
-rw-r-----.  1 database1 consultant1   2048 Mar  4 10:18 db2.conf
```

User	Group memberships
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

L'utente database1 non può cancellare i files app1.log e app2.log.

FALSO !

Vero o falso ?

```
drwxrwxr-t.  2 database1 consultant1   4096 Mar  4 10:23 .
drwxr-xr-x. 10 root      root          4096 Mar  1 17:34 ..

-rw-rw-r--.  1 operator1 operator1     1024 Mar  4 11:02 app1.log
-rw-r--rw-.  1 operator1 consultant1   3144 Mar  4 11:02 app2.log
-rw-rw-r--.  1 database1 consultant1  10234 Mar  4 10:14 db1.conf
-rw-r-----.  1 database1 consultant1   2048 Mar  4 10:18 db2.conf
```

User	Group memberships
operator1	operator1, consultant1
database1	database1, consultant1
database2	database2, operator2
contractor1	contractor1, operator2

L'utente consultant1 può creare files in questa cartella ma non può cancellare app1.log

VERO !

12

Red Hat Enterprise Linux

Installing and Updating Software with RPM

Red Hat

Investigare i pacchetti RPM

RPM = Red Hat Package Manager, formato standard per i pacchetti software su RHEL.

Contiene: binari, configurazioni, metadata (dipendenze, versione, architettura).

Comandi utili per eseguire verifiche:

- rpm -q <pacchetto> → verifica se il pacchetto è installato.
- rpm -qi <pacchetto> → mostra informazioni sul pacchetto.
- rpm -ql <pacchetto> → elenca i file installati.

Analizzare file RPM non installati

È possibile analizzare un file .rpm prima di installarlo.

Comandi principali:

- rpm -qpi <file.rpm> → informazioni sul pacchetto.
- rpm -qpl <file.rpm> → lista dei file che verranno installati.

Utile per:

- Verificare versioni prima dell'installazione.
- Capire se contiene i binari necessari.

Installare pacchetti con DNF

DNF è il gestore pacchetti predefinito su RHEL 8+ (successore di YUM).

- Gestisce automaticamente le dipendenze.
- Comandi principali:
  - dnf install <pacchetto> → installa un pacchetto.
  - dnf remove <pacchetto> → rimuove un pacchetto.
- Confronto con RPM:
  - RPM installa pacchetti locali senza risolvere dipendenze.
  - DNF scarica dai repository e risolve automaticamente le dipendenze.

Aggiornare pacchetti con DNF

DNF semplifica gli aggiornamenti del sistema:

- dnf update <pacchetto> → aggiorna un pacchetto specifico.
- dnf upgrade → aggiorna tutti i pacchetti installati.

Opzioni utili:

- dnf check-update → mostra gli aggiornamenti disponibili.
- dnf history → cronologia delle operazioni eseguite.

Gestire repository con DNF

I repository sono collezioni di pacchetti accessibili a DNF.

- Configurazioni in: /etc/yum.repos.d/.
- Comandi principali:
  - dnf repolist → elenca i repository attivi.
  - dnf repolist all → mostra tutti i repository, inclusi quelli disabilitati.
  - dnf config-manager --enable <repo> → abilita un repository.
  - dnf config-manager --disable <repo> → disabilita un repository.

Abilitare repository extra

Alcuni repository sono opzionali (ad esempio codeready-builder).

Per abilitarli:

- subscription-manager repos --enable=<repo-name>

Esempio:

- subscription-manager repos --enable=rhel-10-for-x86_64-appstream-rpms

Permette di accedere a pacchetti aggiuntivi o librerie per sviluppatori.

Cercare e investigare pacchetti con DNF

DNF permette di cercare pacchetti in repository attivi.

Comandi utili:

- dnf search <termine> → cerca pacchetti per nome o descrizione.
- dnf info <pacchetto> → informazioni dettagliate.
- dnf provides <file> → identifica quale pacchetto contiene un file/binario.

Aggiornare tutto il sistema in sicurezza

Prima di aggiornare, è buona pratica:

- Verificare repository attivi.
- Usare dnf check-update.

Comando per aggiornare tutto:

- dnf upgrade --refresh

Con dnf history undo <ID> è possibile ripristinare lo stato precedente (rollback).

13

Red Hat Enterprise Linux

Installing and Updating Applications with Flatpak

Red Hat

Introduzione a Flatpak su RHEL

- Gestione di applicazioni containerizzate per l'utente
- Flatpak consente di distribuire applicazioni in ambienti isolati.
- Integrato in RHEL come gestore preinstallato.
- Supporta repository remoti e personalizzati.
- Utile per applicazioni desktop in ambienti enterprise.

Configurare Flatpak

- Verifica e strumenti di base
- Controllare la presenza del pacchetto `flatpak`.
- Installare strumenti aggiuntivi se mancanti.
- Verificare la configurazione iniziale con `flatpak remotes`.
- Identificare i repository predefiniti disponibili.

Gestione dei Repository con Flatpak

- Abilitazione, disabilitazione e personalizzazione
- Disabilitare o riabilitare un remote con `flatpak remote-modify`.
- Rimuovere un remote con `flatpak remote-delete`.
- Aggiungere un repository personalizzato con `flatpak remote-add`.
- Verificare i remoti configurati con `flatpak remotes --show-details`.

Uso pratico di Flatpak

- Operazioni comuni sugli applicativi
- Cercare applicazioni: `flatpak search <nome>`.
- Installare un'applicazione: `flatpak install <remote> <app>`.
- Aggiornare tutte le app: `flatpak update`.
- Rimuovere un'app: `flatpak uninstall <app>`.
- Ottenere informazioni: `flatpak info <app>`.

14

Red Hat Enterprise Linux

Accessing Removable Media

Red Hat

File System e Dispositivi a Blocchi in RHEL

Identificare e comprendere le basi

- RHEL usa XFS come file system predefinito, con supporto per ext4 ed exFAT.
- I contenuti di un file system sono accessibili tramite mount point.
- I dispositivi a blocchi sono rappresentati come file sotto /dev.
- Esempi: /dev/sda, /dev/vda, /dev/nvme0, /dev/mmcblk0.

Partizioni e LVM

Come organizzare e gestire lo storage ?

- Un disco può essere suddiviso in partizioni (/dev/sda1, /dev/vdb2, /dev/nvme0n1p1).
- Le partizioni possono contenere file system diversi o ruoli distinti.
- LVM (Logical Volume Manager) aggrega dispositivi in gruppi logici.
- I volumi logici si trovano in /dev/<vg>/<lv> oppure tramite /dev/mapper.

Esaminare lo Spazio Disco

Strumenti principali di verifica

- df -h: mostra i file system montati e lo spazio disponibile.
- du -h <dir>: analizza lo spazio occupato in una directory.
- ls -l /dev/<device>: rivela il tipo di dispositivo (es. block device).
- File system temporanei (tmpfs, devtmpfs) risiedono in memoria e non persistono dopo reboot.

Dalla risorsa fisica al file system

- Disco fisico (es. /dev/sda)
- → suddiviso in partizioni (/dev/sda1, /dev/sda2, ...)
- → ogni partizione ha un file system (XFS, ext4, ...)
- → ogni file system è montato su un mount point (/ , /home, /boot,)
- Struttura ad albero unificata: più dischi e partizioni si combinano in un unico filesystem root (/).

```
[ Disco /dev/sda ]
   ├── /dev/sda1 → montato su /boot
   ├── /dev/sda2 → montato su /
   └── /dev/sda3 → montato su /home
```

Montare un File System

- Comando: mount <device> <mount_point>
- Supporta device file (/dev/sda3) o UUID per maggiore stabilità.
- Il mount point deve esistere ed essere vuoto.
- ⚠️ Se non è vuoto, i file pre-esistenti vengono nascosti fino all'unmount.

```
mount /dev/sda3 /mnt/data
```

Identificare i Dispositivi a Blocchi

- lsblk: mostra dispositivi, partizioni, e punti di mount.
- lsblk -fp: include UUID e tipo di file system.
- Montaggio via UUID:
- mount UUID="<uuid>" /mnt/data.
- Evita dipendere dall'ordine di rilevamento dei dispositivi.

```
lsblk -fp
NAME     FSTYPE FSVER LABEL UUID                   FSAVAIL FSUSE% MOUNTPOINTS
/dev/sda
├─/dev/sda1
├─/dev/sda2 vfat   FAT16    7B77-95E7              191.4M  4%    /boot/efi
└─/dev/sda3 xfs        root  15507695-...-...95983f  7.4G   24%  /
```

Identificare i Dispositivi a Blocchi

- Comando: umount <mount_point>.
- Tutti i processi devono cessare di usare il file system.
- Se il target è “busy”, usare lsof <mount_point> per identificare i processi attivi.
- Unmount sempre prima di scollegare dispositivi rimovibili.
- ⚠️ I file system sono smontati automaticamente in fase di reboot/shutdown.

```
lsblk -fp
NAME     FSTYPE FSVER LABEL UUID                 FSAVAIL FSUSE% MOUNTPOINTS
/dev/sdalsof /mnt/data
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash 1593 root  cwd    DIR 253,17         6  128 /mnt/data
lsof 2532 root  cwd    DIR 253,17        19  128 /mnt/data
lsof 2533 root  cwd    DIR 253,17        19  128 /mnt/data
├─/dev/sda1
├─/dev/sda2 vfat   FAT16    7B77-95E7                 191.4M  4%    /boot/efi
└─/dev/sda3 xfs        root  15507695-...-...95983f  7.4G   24%  /
```

Comando locate

- Ricerca file per nome o percorso in un database pre-generato
- È molto veloce ma non aggiornato in tempo reale
- Richiede aggiornamenti periodici del database (updatedb)
- Opzioni comuni:
- -i → ricerca case-insensitive
- -n NUM → limita i risultati
- Accesso limitato ai file e directory in base ai permessi dell'utente

```
locate passwd
/etc/passwd
/etc/passwd-
/etc/pam.d/passwd
```

Comando updatedb

- Aggiorna manualmente il database usato da locate
- Generalmente il sistema lo esegue in automatico una volta al giorno
- Eseguito dall'utente root per garantire copertura completa
- Dopo un aggiornamento, locate restituirà anche i file creati di recente

```
updatedb
```

Comando find

- Esegue una ricerca scandendo la gerarchia di directory in tempo reale
- Più lento di locate, ma sempre accurato
- Permette criteri avanzati:
  - Nome (-name, -iname, con wildcard)
  - Proprietario (-user, -group, -uid, -gid)
  - Permessi (-perm)
  - Dimensioni (-size +10M, -size -5k)
  - Tempo di modifica (-mmin, -mtime)
  - Tipo (-type f/d/l/b)
- Restituisce solo file accessibili in base ai permessi dell'utente

```
find . -name "*.log"

find /var/tmp -name "*.tmp" -mtime +7 -exec rm -f {} \;
```

15

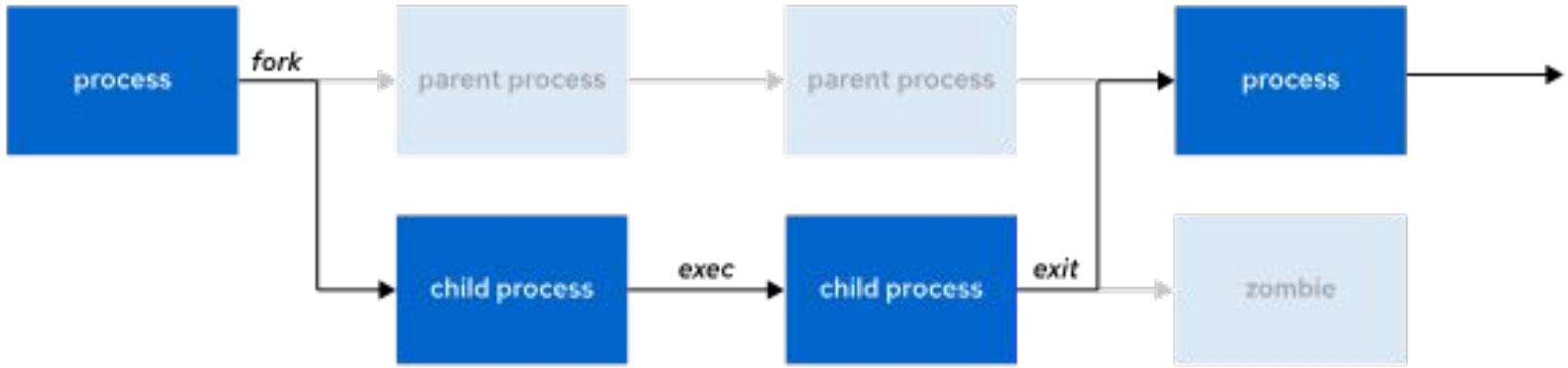
Red Hat Enterprise Linux

Monitoring and Managing Linux Processes

Red Hat

Ciclo di vita di un Processo

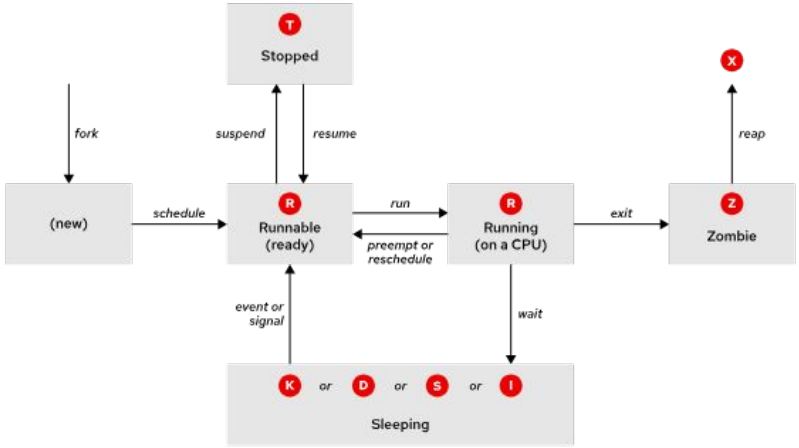
- ⚙️ Un processo è un'istanza attiva di un programma in esecuzione, con risorse e ambiente proprio
- 🔁 Un processo padre può duplicarsi tramite fork, creando un processo figlio indipendente.
- 🆔 Ogni processo ha un PID univoco e conserva il PPID del padre per tracciamento e sicurezza.
- 🌳 Tutti i processi discendono da systemd



Red Hat

Ciclo di vita di un Processo

- 🟢 Running (R): Il processo è in esecuzione o pronto per essere eseguito.
- 🟡 Sleeping (S): In attesa di un evento o risorsa (es. I/O), può essere risvegliato.
- 🔴 Stopped (T): Esecuzione sospesa manualmente (es. con SIGSTOP o Ctrl+Z).
- ⚫ Zombie (Z): Terminato ma non ancora rimosso dal processo padre.
- 🔵 Uninterruptible Sleep (D): In attesa di I/O non interrompibile (es. accesso disco).
- ⚪ Traced/Debugged (t): Sotto controllo da un debugger (es. ptrace).



Comando ps

ps (process status) mostra una istantanea dei processi in esecuzione al momento

- Opzioni utili:
  - ps aux per tutti i processi con dettagli utente, CPU, memoria
  - ps -ef formato completo con informazioni sulle relazioni tra processi (PPID, UID, ecc.)
- Permessi: l'utente può vedere solo processi a cui ha accesso
- Usi tipici: diagnosticare processi critici, controllare PID

```
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 08:00 ?        00:00:02 /usr/lib/systemd/systemd
root       567     1  0 08:01 ?        00:00:00 /usr/lib/systemd/systemd-jou
francesco 1324  1300  0 08:05 pts/0    00:00:00 bash
francesco 1350  1324  0 08:06 pts/0    00:00:00 ps -ef
```

Comando top

- Visualizza informazioni aggiornate periodicamente (CPU, memoria, carico sistema, numero di processi)
- Mostra processi ordinati solitamente per uso CPU o memoria
- phoenixNAP | Global IT Services
- Interattivo: è possibile inviare segnali (es. kill), modificare priorità (nice), filtrare o ordinare le colonne mentre top è in esecuzione
- Usi frequenti: identificare processi che usano troppe risorse, verificare memory leak, capire l'utilizzo di swap, carico dei core CPU

```
top - 19:49:03 up  2:15,  2 users,  load average: 0.12, 0.08, 0.05
Tasks: 123 total,   1 running, 122 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.3 us,  0.7 sy,  0.0 ni, 96.5 id,  0.3 wa,  0.0 hi,  0.2 si,  0.(
MiB Mem :   7982.3 total,   1523.4 free,   2341.2 used,   4117.7 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   5321.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1324 francesco  20   0   123m   10m   5m   S   0.3  0.1   0:00.12 bash
 1350 francesco  20   0   105m   8m    4m   R   1.2  0.1   0:00.03 top
```


Red Hat

Esecuzione processi in background

- Aggiungere & alla fine di un comando → lo esegue in background
- Con pipeline (|), tutto il pipeline diventa un unico job
- Comando jobs → elenca i job della sessione corrente
  - Stato: Running, Stopped

```
sleep 10000 &
[1] 5947

ls | sort | mail -s "Sort output" &
[1] 5998

jobs
[1]+ Running  sleep 10000 &
```

Controllo dei jobs

- fg %<job_number> → porta un job in foreground
- Ctrl+Z → sospende un processo in foreground (stato: Stopped)
- bg %<job_number> → riattiva un job sospeso in background
- Avviso: la shell avverte se ci sono job sospesi quando si tenta di uscire

```
sleep 10000
^Z
[1]+  Stopped                 sleep 10000

bg %1
[1]+ sleep 10000 &
```

Segnali che notificano eventi ai processi

- Un segnale è un'interruzione software inviata a un processo.
- Può derivare da: errori, eventi esterni (I/O, timer), o comandi espliciti.
- Segnali comuni da tastiera:
  - Ctrl+C → SIGINT (termina il processo)
  - Ctrl+Z → SIGTSTP (sospende il processo)
  - Ctrl+\ → SIGQUIT (core dump + termina)
- I segnali si identificano per nome (es. -SIGHUP) o numero (es. -1).

Invio di segnali con kill

- Il comando kill consente di inviare segnali a processi specifici
  - kill -l → elenca tutti i segnali disponibili.
  - kill PID → invia SIGTERM (terminazione “gentile”).
  - kill -9 PID o kill -SIGKILL PID → terminazione forzata.
  - ps aux | grep <nome> → trova i PID da terminare.
- Varianti utili:
  - pkill → invia segnali per nome processo/utente/tty.
  - pgrep → trova i PID senza terminarli.
  - killall <nome> → termina tutti i processi con quel nome.

Controllo e terminazione utenti

- w e who → visualizzano utenti, terminali, e sessioni attive.
- pkill -t ttyN → termina tutti i processi su un terminale specifico.
- pkill -u <utente> → invia segnali a tutti i processi di un utente.
- Solo root può terminare processi di altri utenti.
- pstree -p <utente> → mostra la gerarchia processi → kill selettivo.
- jobs e kill %N → terminano job in background nella stessa shell.

Comando uptime

- Mostra da quanto tempo il sistema è attivo, il numero di utenti connessi e il carico medio della CPU.

```
20:15:03 up 2 days,  3:42,  2 users,  load average: 0.12, 0.08, 0.05
```

- Ora corrente → 20:15:03
- Tempo di attività → up 2 days, 3:42
- Utenti connessi → 2 users
- Load average → carico medio su 1, 5 e 15 minuti
- 📘 Nota: Il load average rappresenta il numero medio di processi in attesa di CPU. Valori sotto 1 indicano un sistema poco carico (su CPU singola).

Comando lscpu

- Visualizza informazioni dettagliate sull'architettura della CPU.

```
Architecture:      x86_64
CPU(s):            8
Thread(s) per core:  2
Core(s) per socket:  4
Vendor ID:         GenuineIntel
Model name:        Intel(R) Core(TM) i7-8650U
```

- Architettura → tipo di CPU (es. x86_64)
- CPU(s) → numero totale di thread visibili
- Core(s) → core fisici per socket
- Vendor/Model → produttore e modello della CPU

📘 Nota: Utile per analizzare la parallelizzazione e ottimizzare carichi di lavoro in ambienti multi-core.

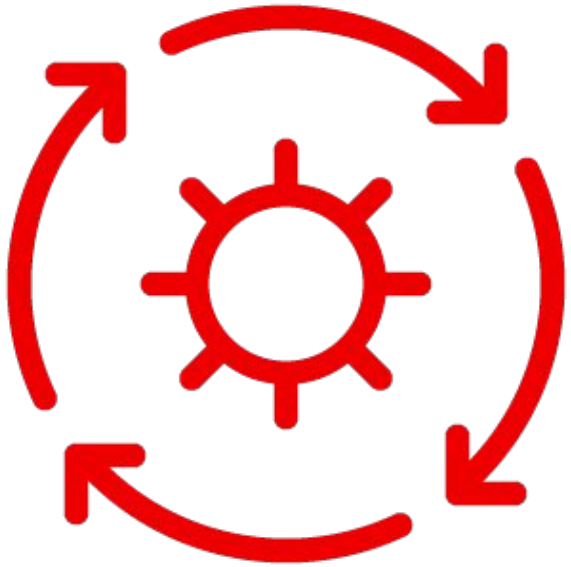
16

Red Hat Enterprise Linux

Controlling Services and Daemons

Red Hat

systemd



- Gestisce avvio del sistema e servizi in esecuzione.
- Avvia risorse, demoni e processi sia al boot che a runtime.
- Caratteristiche principali:
  - Parallelizzazione → avvio più veloce.
  - Avvio on-demand di demoni.
  - Gestione automatica delle dipendenze.
  - Uso di control groups (cgroups) per tracciare processi correlati.

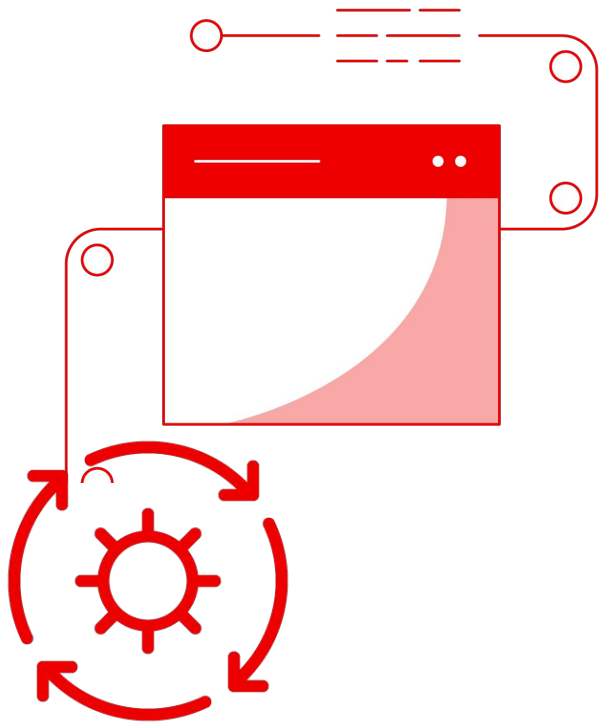
Avvio di un processo con systemd

- Avvio ed arresto del servizio sshd

```
sudo systemctl start sshd.service

systemctl status sshd.service

● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-05-22 14:47:33 UTC; 2h 48min ago
```

```
sudo systemctl stop sshd.service

systemctl is-active sshd.service
inactive
```

Le units gestite da systemd



- I demoni (es: httpd, sshd, chronyd) girano in background.
- In systemd, i servizi sono gestiti tramite units.
- Units più comuni:
  - Service (.service) → rappresenta un demone (es. web server).
  - Socket (.socket) → attiva servizi su richiesta.
  - Path (.path) → attiva servizi su eventi del filesystem.
- Ogni unit ha un nome e un tipo → identificazione univoca.

Comandi per controllare systemd

- Visualizzare i servizi attivi:
  - systemctl list-units --type=service
- Visualizzare tutte le unità (attive e inattive):
  - systemctl list-units --type=service --all
- Visualizzare i file di unità installati:
  - systemctl list-unit-files --type=service
- Stati comuni:
  - enabled → avvio al boot
  - disabled → non avvio automatico
  - static → avviato solo da altre unità
  - masked → disabilitato completamente

Abilitazione al boot di un servizio

- Usa systemctl enable nome-servizio per attivare il servizio al boot.
- 🔗 Questo comando crea un link simbolico nelle directory di avvio (/etc/systemd/system/...).
- Verifica lo stato con systemctl is-enabled nome-servizio.

```
$ sudo systemctl enable sshd
Created symlink /etc/systemd/system/multi-user.target.wants/sshd.service →
/usr/lib/systemd/system/sshd.service.

$ systemctl is-enabled sshd
enabled
```

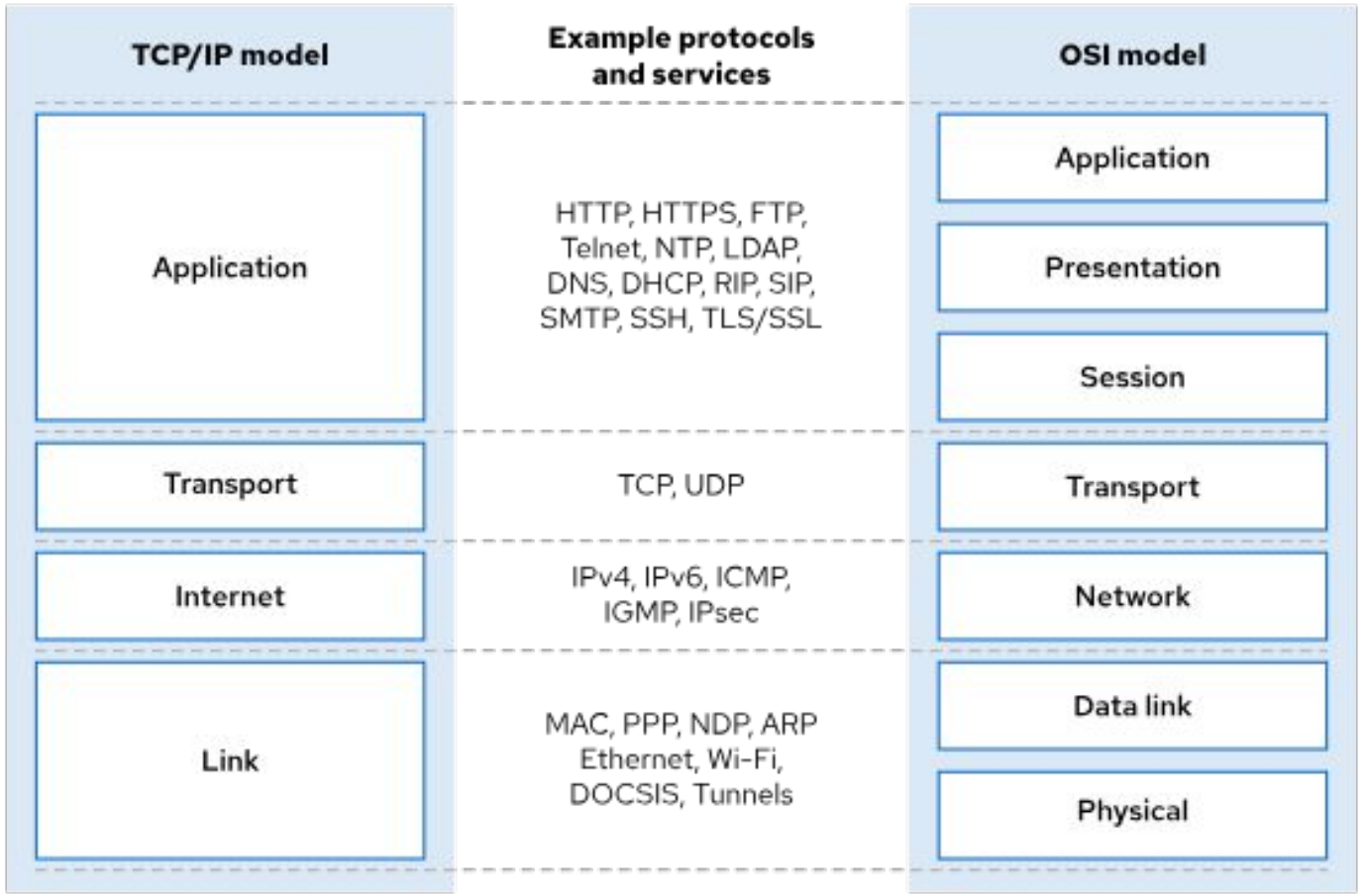
17

Red Hat Enterprise Linux

Introduction to Networking

Red Hat

TCP/IP vs OSI Network model

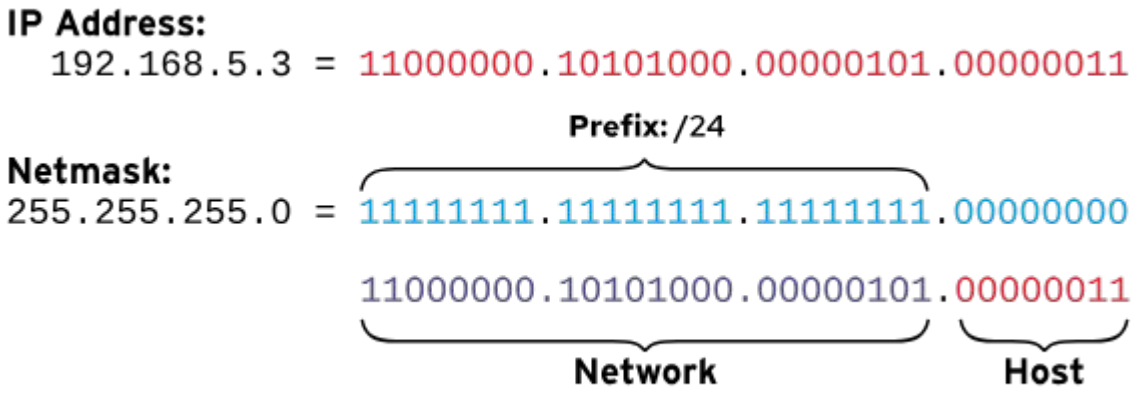


Network Interface Naming

- Linux RHEL usa nomi persistenti delle interfacce di rete.
- Nomi basati su firmware e topologia PCI, non sull'ordine di rilevamento.
- Formati principali:
  - en → Ethernet, wl → WLAN, ww → WWAN
  - eno1 → Ethernet onboard 1
  - ens3 → Ethernet in slot PCI 3
  - enp2s3 → Ethernet PCI bus 2 slot 3
  - enp0s1f0 → funzione 0 di un dispositivo multifunzione
- Vantaggio: nomi stabili anche aggiungendo o rimuovendo hardware.

IPv4 Networking

- 32-bit, 4 ottetti decimali separati da punti.
- Suddiviso in network prefix + host identifier.
- Subnetting: divide una rete in segmenti più piccoli.
- Netmask: definisce quanti bit servono per il network prefix.
  - /24 → 256 indirizzi totali, 254 host utilizzabili
  - /19 → 8190 host
  - /8 → 16.7 milioni host

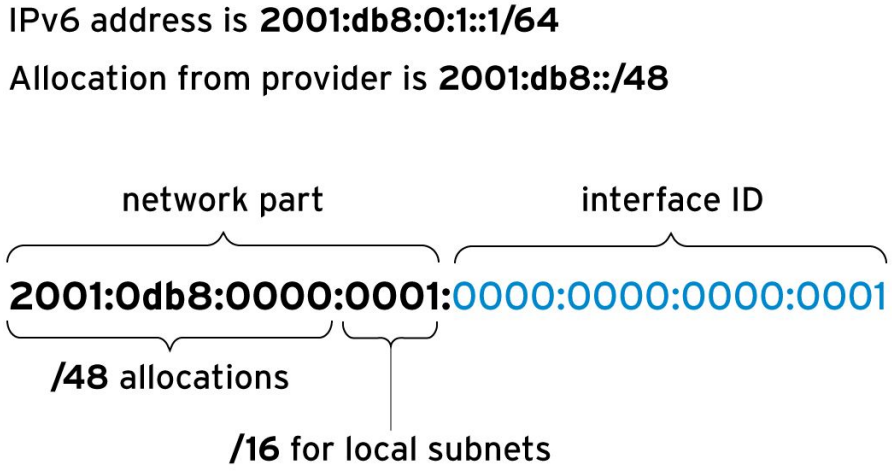


Network Interface Naming

- Routing:
  - Ogni host usa una routing table per scegliere l'interfaccia e il router.
  - Entry più specifica (prefisso più lungo) ha priorità.
  - Default route = 0.0.0.0/0 → invio a gateway predefinito.
- Configurazione IP:
  - DHCP → assegnazione automatica IP, netmask, gateway, DNS
  - Static IP → configurazione manuale tramite file di configurazione
  - Strumento utile: ipcalc → calcola rete, broadcast e range host.

IPv6 Networking

- 128-bit, 8 gruppi di 4 cifre esadecimali separati da :
- Riduzioni:
- Rimuovere zeri iniziali: 0010 → 10
- Gruppi consecutivi di zeri → :: (una sola volta)
- Notazione porta: [2001:db8::1]:80
- Subnetting IPv6:
- Standard: /64 → 64 bit per host, fino a 2^64 dispositivi
- ISP tipico: /48 → 16 bit per sottoreti → 65,536 subnet



Commando iplink

- Mostra tutte le interfacce di rete disponibili.
- Include nome interfaccia, stato (UP/DOWN) e MAC address.
- Importante per identificare quale interfaccia è collegata a quale rete.

```
user@host:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 ...
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 8942 ...
```

Commando ipaddr

- Mostra gli indirizzi IPv4 e IPv6 assegnati a una interfaccia.
- Indica lo stato dell'interfaccia, il MAC, e eventuali alias.
- Distinguere tra scope globale e link-local.

```
user@host:~$ ip addr show ens3
inet 172.25.250.10/24 brd 172.25.250.255 scope global ens3
inet6 2001:db8:0:1:5054:ff:fe00:b/64 scope global
inet6 fe80::5054:ff:fe00:fa0a/64 scope link
```

Commando ip -s link

- Mostra statistiche RX/TX: pacchetti ricevuti/inviati, errori, pacchetti persi.
- Utile per diagnosticare problemi di rete o congestione.

```
user@host:~$ ip -s link show ens3
RX:  bytes packets errors dropped ...
TX:  bytes packets errors dropped ...
```

Commando ping / ping6

- Verifica la connettività IPv4/IPv6 verso un host remoto.
- -c N limita il numero di pacchetti inviati.
- Mostra RTT (round-trip time) e eventuali perdite di pacchetti.

```
user@host:~$ ping -c3 192.0.2.254
3 packets transmitted, 3 received, 0% packet loss
```

Commando iproute /ip 6 route

- Mostra le route per reti locali e default gateway.
- Determina quale interfaccia invia i pacchetti verso una destinazione.

```
user@host:~$ ip route
default via 192.0.2.254 dev ens3
192.0.2.0/24 dev ens3 src 192.0.2.2
10.0.0.0/8 dev ens4 src 10.0.0.11
```

Commando tracepath /tracepath6

- Mostra i router intermedi tra host sorgente e destinazione.
- Utile per diagnosticare problemi di routing o latenze..

```
user@host:~$ tracepath access.redhat.com
 4:  71-32-28-145.rcmt.qwest.net 48.853ms
 5:  dcp-brdr-04.inet.qwest.net 100.732ms
```

Commando ss

- Mostra socket TCP/UDP, stato, indirizzi locali e remoti.
- Sostituisce il vecchio netstat.
- Utile per diagnosticare servizi in ascolto o connessioni attive.

```
user@host:~$ ss -tan
LISTEN 0 100 127.0.0.1:25 0.0.0.0:*
ESTAB  0 52 172.25.250.10:22 172.25.250.9:57560
```

Aggiornare il Nome Host

Il nome host identifica il sistema all'interno di una rete. Può essere modificato temporaneamente o permanentemente usando hostname o hostnamectl.

- Visualizzare il nome host corrente: **hostname** o **hostnamectl** status
- Modifica temporanea: **hostname** nuovo-hostname (resettabile al riavvio)
- Modifica permanente: sudo **hostnamectl** set-hostname nuovo-hostname
- Aggiornare /etc/hosts se necessario

Nessun riavvio richiesto per la modifica permanente

```
# Temporaneo
hostname webserver01

# Permanente
sudo hostnamectl set-hostname webserver01.example.com

# Verifica
hostnamectl
```

18

Red Hat Enterprise Linux

Managing Network Configuration

Red Hat

Network Manager Command Line Interface (nmcli)

- Strumento a riga di comando per gestire la rete su RHEL e sistemi basati su NetworkManager.
- Permette di visualizzare, creare, modificare, attivare e disattivare connessioni di rete.
- Alternativa alla GUI, utile su server o in automazione.

```
nmcli –help
```

Visualizzazione network connections

- Mostra tutte le connessioni conosciute da NetworkManager.
- Include tipo di connessione, interfaccia associata, stato e UUID.

```
user@host:~$ nmcli connection show
NAME     UUID                                  TYPE      DEVICE
ens3     1a2b3c4d-5e6f-7a8b-9c0d-1234567890ab  ethernet  ens3
ens4     2b3c4d5e-6f7a-8b9c-0d1e-234567890abc  ethernet  ens4
```

Visualizzazione stato interfacce

- Mostra lo stato attuale delle interfacce: attiva, inattiva, disconnessa.
- Include tipo di dispositivo, stato e connessione attiva.

```
user@host:~$ nmcli device status
DEVICE  TYPE      STATE     CONNECTION
ens3 ethernet  connected  ens3
ens4 ethernet  disconnected  --
lo   loopback unmanaged  --
```

Connessione e disconnessione ad interfaccia

- Con nmcli connection up attiva una connessione di rete già configurata.
- Necessario quando si crea o modifica una connessione o dopo un reboot.
- Con nmcli connection down disattiva una connessione di rete già configurata.

```
user@host:~$ nmcli connection up ens4
Connection 'ens4' successfully activated (DUID: ...)

user@host:~$ nmcli connection down ens4
Connection 'ens4' successfully deactivated (DUID: ...)
```

19

Red Hat Enterprise Linux

Configuring and Securing SSH

Red Hat

Il Ruolo delle Chiavi Host SSH

Le chiavi host SSH identificano un server in modo univoco e permettono di garantire la sicurezza delle connessioni SSH.

- Identificano il server al client
- Prevengono attacchi “man-in-the-middle”
- Ogni server ha chiavi host diverse per algoritmi come RSA, ECDSA o ED25519

```
# Visualizzare le chiavi host sul server
sudo ls /etc/ssh/ssh_host_*
```

Configurare StrictHostKeyChecking

Il controllo delle chiavi host (Host Key Checking) garantisce che il client si connetta al server corretto confrontando la chiave host.

- Impostazione predefinita in SSH: verifica automatica delle chiavi
- Disabilitare con StrictHostKeyChecking no (solo per test)
- Configurabile in /etc/ssh/ssh_config o ~/.ssh/config

```
# Connessione con controllo chiavi
ssh user@server.example.com

# Disabilitare temporaneamente il controllo
ssh -o StrictHostKeyChecking=no user@server.example.com
```

Verifica delle Fingerprint delle Chiavi Host SSH

Verificare la fingerprint della chiave host garantisce che la connessione SSH sia sicura e il server autentico.

- Fingerprint = hash della chiave host
- Utile al primo accesso al server
- Può essere verificata confrontando il client con /etc/ssh/ssh_host_*.pub sul server

```
# Visualizzare fingerprint RSA
ssh-keygen -lf /etc/ssh/ssh_host_rsa_key.pub

# Confronto con client
ssh-keyscan server.example.com
```

Rigenerare le Chiavi Host SSH sui Server

Rigenerare le chiavi host è necessario in caso di compromissione o aggiornamenti di sicurezza.

- Usare ssh-keygen per generare nuove chiavi
- Riavviare il servizio SSH per applicare le nuove chiavi
- Aggiornare eventuali client che conoscono la vecchia chiave

```
# Rigenerare chiave RSA
sudo ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""

# Riavviare SSH
sudo systemctl restart sshd
```

Gestire i File Known Hosts sui Client

Il file ~/.ssh/known_hosts conserva le chiavi host dei server già visitati dal client.

- Evita avvisi di sicurezza ad ogni connessione
- Può contenere chiavi multiple per host diversi
- Rimuovere chiavi obsolete o cambiate per evitare errori di connessione

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

# Rimuovere chiave obsoleta
ssh-keygen -R server.example.com
```

Autenticazione SSH basata su chiavi

L'autenticazione tramite chiavi SSH permette di collegarsi ai server senza usare password. Si basa su crittografia a chiave pubblica, dove una chiave privata resta segreta sul client e la chiave pubblica viene copiata sul server.

- Le password possono essere rubate, le chiavi private protette sono più sicure
- Chiave privata = credenziale dell'utente
- Chiave pubblica = verifica la corrispondenza con la chiave privata
- Il server cripta una sfida usando la chiave pubblica; il client la decifra per autenticarsi

```
# Generare una coppia di chiavi
ssh-keygen

# Chiave privata: ~/.ssh/id_ed25519
# Chiave pubblica: ~/.ssh/id_ed25519.pub
```

Generazione delle chiavi SSH

Su RHEL 10 il tipo di chiave predefinito è Ed25519, più sicuro e performante di RSA. È possibile usare RSA se necessario, soprattutto in modalità FIPS.

- Ed25519 = chiave più corta ma sicura
- RSA = ancora supportata, necessaria in FIPS
- Proteggere la chiave privata con una passphrase per maggiore sicurezza

```
# Creazione chiave Ed25519 senza passphrase
ssh-keygen

# Creazione chiave Ed25519 con passphrase
ssh-keygen -f ~/.ssh/key-with-pass
Enter passphrase: my-secret
```

Distribuzione della chiave pubblica e login

Per accedere senza password, la chiave pubblica va copiata sul server remoto. Il comando ssh-copy-id facilita questa operazione.

- Copia la chiave pubblica sul server remoto
- Il server verifica il client senza richiedere password
- Se la chiave privata ha una passphrase, il client la digita solo localmente

```
# Copiare la chiave pubblica sul server
ssh-copy-id -i ~/.ssh/key-with-pass.pub user@remotehost

# Login usando la chiave privata
ssh -i ~/.ssh/key-with-pass user@remotehost
```

Gestione delle chiavi con ssh-agent

Se la chiave privata è protetta da passphrase, l'ssh-agent può memorizzarla in memoria per ridurre l'inserimento ripetuto.

- ssh-agent = gestore delle chiavi, memorizza passphrase temporaneamente
- Migliora sicurezza e comodità
- Supportato automaticamente su GNOME; su terminale va avviato manualmente

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

# Aggiungere chiavi all'agent
ssh-add ~/.ssh/id_ed25519
ssh-add ~/.ssh/key-with-pass

# Visualizzare chiavi caricate
ssh-add -l
```

Configurazione client/server e sicurezza

È buona pratica configurare sia client che server per migliorare sicurezza e comodità.

- Client: file ~/.ssh/config per preconfigurare host, utenti e chiavi
- Server: /etc/ssh/sshd_config per limitare login root e disabilitare password
- Parametri chiave server:
- PermitRootLogin no → vieta login diretto di root
- PasswordAuthentication no → vieta login con password, forza chiave privata

```
# Esempio client SSH config
cat ~/.ssh/config
Host remotehost
    HostName remotehost.example.com
    User user
    IdentityFile ~/.ssh/id_ed25519

# Ricaricare server SSH dopo modifica config
sudo systemctl reload sshd
```

Red Hat
Enterprise Linux

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


youtube.com/c/RedHatEnterpriseLinux


twitter.com/RHEL


Reddit.com/r/redhat

Red Hat