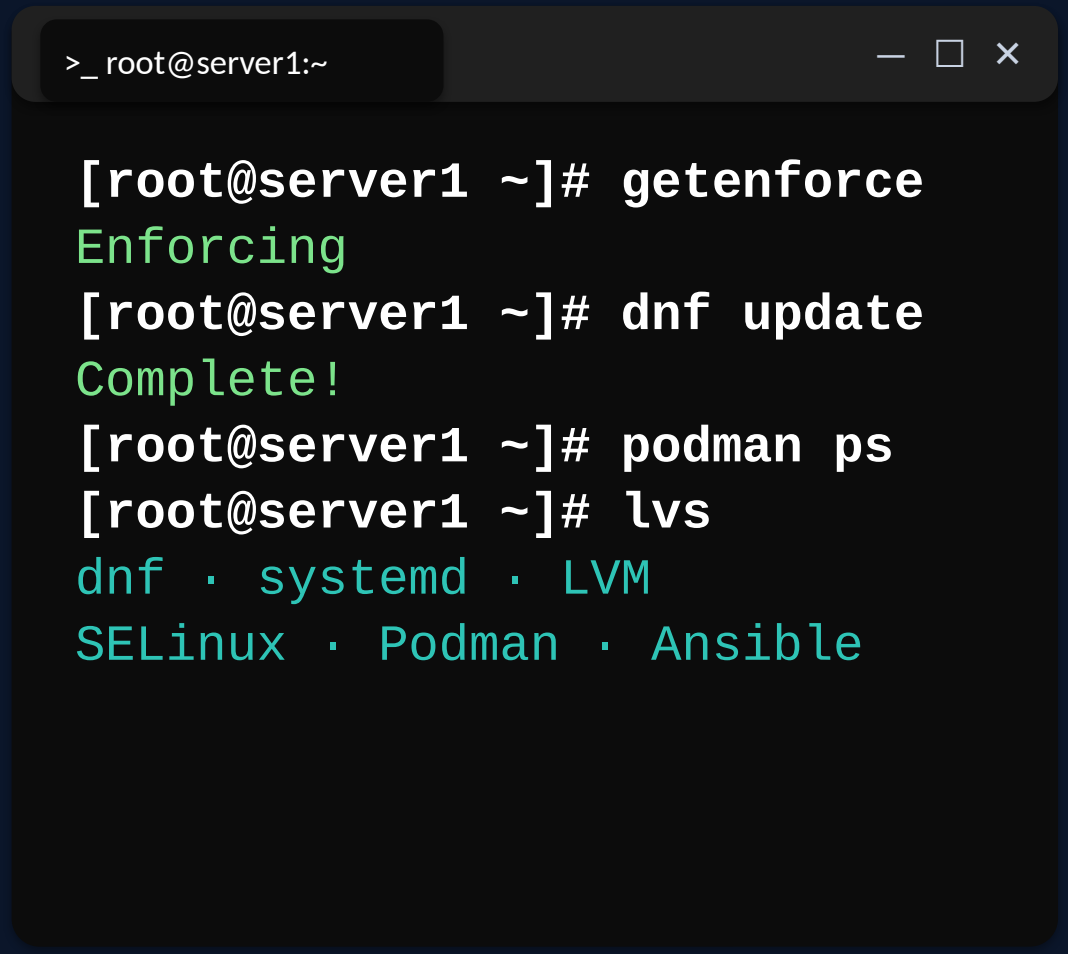


LESSON 4 OF 12

Red Hat Enterprise Linux

RHEL 9 & 10 · Admin · SELinux

Users, Groups, Permissions, ACLs & sudo

Full Course · RHEL 10.3 & 9.9 (Nov 2026)

What You Will Learn in Lesson 4

- Users, groups, UIDs and the files behind them
- Creating and managing users and groups
- Passwords and password aging
- File permissions: read, write, execute; chmod and chown
- Special permissions: setgid, sticky bit; umask
- ACLs for fine-grained access
- sudo and the wheel group

Users and Groups: The Basics

Item	Details
root	UID 0: can do anything
System users	UID 1–999: run services such as apache or sshd
Normal users	UID 1000 and up: people
Primary group	Each user has one; new files get this group
Supplementary groups	Extra groups, such as wheel for administrators
id student	uid=1000(student) gid=1000(student) groups=…,10(wheel)

The Files Behind Users

File	Holds
/etc/passwd	name:x:UID:GID:comment:home:shell — readable by everyone
/etc/shadow	Password hashes and aging — root only
/etc/group	group:x:GID:members
/etc/gshadow	Group passwords and admins — root only
/etc/login.defs	Defaults: UID ranges, password aging, home creation
/etc/skel	Files copied into every new home directory

Create and Manage Users and Groups

```
[student@server1 ~]$ sudo groupadd -g 2000 developers
[student@server1 ~]$ sudo useradd -c 'Sara Ahmadi' -G developers sara
[student@server1 ~]$ sudo passwd sara
[student@server1 ~]$ sudo useradd -u 1500 -s /sbin/nologin svc-backup   # no login
[student@server1 ~]$ sudo usermod -aG wheel sara       # -a: append, keep old groups!
[student@server1 ~]$ sudo usermod -L sara ; sudo usermod -U sara   # lock / unlock
[student@server1 ~]$ sudo userdel -r olduser           # delete with home directory
[student@server1 ~]$ id sara ; getent group developers
```

Passwords and Password Aging

```
[student@server1 ~]$ sudo chage -l sara
Last password change                    : Nov 20, 2026
Password expires                        : never
[student@server1 ~]$ sudo chage -M 90 -m 1 -W 7 sara     # max 90 days, warn 7 days before
[student@server1 ~]$ sudo chage -d 0 sara                 # must change at next login
[student@server1 ~]$ sudo chage -E 2027-06-30 contractor  # account expires
[student@server1 ~]$ grep ^PASS_ /etc/login.defs            # defaults for new users
[student@server1 ~]$ cat /etc/security/pwquality.conf       # minlen, complexity
```

File Permissions

Part of ls -l	Meaning
-rw-r----- 1 sara developers report.txt	Type, permissions, owner, group, name
rw- r-- ---	Owner (u) / group (g) / others (o)
r = 4, w = 2, x = 1	Read, write, execute — add them: rw- = 6
On files	r read, w change, x run as a program
On directories	r list, w create/delete files, x enter (cd)
Who wins?	Owner first, then group, then others

chmod and chown

```
[student@server1 ~]$ chmod 640 report.txt            # rw-r-----
[student@server1 ~]$ chmod u+x,g-w,o= script.sh      # symbolic form
[student@server1 ~]$ chmod -R g+rwX /srv/projects    # X: execute on directories only
[student@server1 ~]$ sudo chown sara:developers report.txt
[student@server1 ~]$ sudo chown -R :developers /srv/projects
[student@server1 ~]$ ls -l report.txt
-rw-r----- 1 sara developers 2048 Nov 20 10:00 report.txt
[student@server1 ~]$ stat -c '%A %a %U:%G %n' report.txt
```

Special Permissions and umask

Setting	Effect
setuid (4, u+s)	Program runs as its owner, e.g. /usr/bin/passwd
setgid on a directory (2, g+s)	New files inherit the directory's group — great for teams
sticky bit (1, o+t)	Only the owner can delete their files, e.g. /tmp
chmod 2770 /srv/projects	Team folder: group shares, others locked out
umask 022 / 002 / 077	Removes rights from new files: 644/755, 664/775, 600/700
Set umask	~/.bashrc for a user; /etc/login.defs for defaults

ACLs: Fine-Grained Access

```
[student@server1 ~]$ setfacl -m u:ahmad:rw report.txt       # one extra user
[student@server1 ~]$ setfacl -m g:auditors:r report.txt     # one extra group
[student@server1 ~]$ setfacl -R -m d:g:developers:rwX /srv/projects   # default ACL
[student@server1 ~]$ getfacl report.txt
# file: report.txt
user::rw-
user:ahmad:rw-
group::r--
group:auditors:r--
mask::rw-
[student@server1 ~]$ setfacl -x u:ahmad report.txt ; setfacl -b report.txt   # remove
```

sudo: Safe Administrator Rights

```
[student@server1 ~]$ sudo -l                    # what may I run?
[student@server1 ~]$ sudo -i                    # root shell (logged)
[student@server1 ~]$ sudo visudo -f /etc/sudoers.d/webteam
%webteam  ALL=(root)  /usr/bin/systemctl restart httpd, /usr/bin/systemctl status httpd
ahmad     ALL=(ALL)   ALL

# wheel group already has full sudo in /etc/sudoers:
%wheel    ALL=(ALL)   ALL
[student@server1 ~]$ sudo journalctl _COMM=sudo --since today   # who ran what
```

Summary and Homework

- useradd, usermod -aG, passwd, chage, userdel -r
- rwx = 421; chmod 640, chmod g+s, chown user:group
- setgid team folders; umask controls new file rights
- setfacl/getfacl when one owner and one group are not enough
- sudo + wheel; least privilege with /etc/sudoers.d
- Homework: create group developers, users sara and ahmad, and a setgid team folder /srv/projects

Thank You! 🙏

Free videos & PDFs: www.MyWebUniversity.com

Slides for this lesson: MyWebUniversity.com/presentations

YouTube: @MyWebUniversity · @MyWebUniversity-Dari

Wahidullah Lutfy — Founder and Director of IT of MyWebUniversity, and CS teacher

Book: The Direct Path to Linux Ubuntu (Amazon · Barnes & Noble)

Learn it, then pass it on.