Essential Linux Commands for Backend Development

Below is a clean format: **Command → What it does → Example**

---

1. Navigation & File Operations

ls – list files
Example: `ls -la`

cd – change directory
Example: `cd backend/`

pwd – show current directory
Example: `pwd`

mkdir – create a new folder
Example: `mkdir project`

rm – delete a file
Example: `rm index.js`

rm -r – delete a folder
Example: `rm -r node_modules/`

cp – copy a file
Example: `cp app.js app_backup.js`

mv – move/rename file or folder
Example: `mv oldname.js newname.js`

touch – create empty file
Example: `touch server.js`

---

2. Permissions

chmod – change file permissions
Example: `chmod 755 script.sh`

chown – change file ownership
Example: `sudo chown ubuntu:ubuntu file.txt`

sudo – run command as admin
Example: `sudo apt update`

---

3. System Info

top – view running processes
Example: `top`

htop – better process viewer
Example: `htop`

df -h – disk usage in human format
Example: `df -h`

du -sh – folder size
Example: `du -sh logs/`

free -h – RAM usage
Example: `free -h`

uname -a – system info
Example: `uname -a`

---

4. Package Management (Ubuntu/Debian)

apt update – refresh package list
Example: `sudo apt update`

apt upgrade – upgrade packages
Example: `sudo apt upgrade`

apt install – install a package
Example: `sudo apt install nodejs`

---

5. Networking

ping – test internet
Example: `ping google.com`

curl – fetch URL response
Example: `curl https://api.example.com`

wget – download file
Example: `wget https://example.com/file.zip`

ip a – show network info
Example: `ip a`

netstat -tulnp – check open ports
Example: `sudo netstat -tulnp`

---

6. Git (Backend Essential)

git init – start a git repo
Example: `git init`

git clone – download repo
Example: `git clone https://github.com/user/repo.git`

git add . – stage all changes
Example: `git add .`

git commit -m – commit changes
Example: `git commit -m "initial commit"`

git push – upload changes
Example: `git push origin main`

git pull – download updates
Example: `git pull`

---

7. Node.js Commands

node file.js – run JS file
Example: `node server.js`

npm install – install packages
Example: `npm install express`

npm start – run project script
Example: `npm start`

pm2 start – run backend in background
Example: `pm2 start server.js`

---

8. Docker Commands

docker pull – download image
Example: `docker pull node:latest`

docker images – list images
Example: `docker images`

docker ps – running containers
Example: `docker ps`

docker run – start container
Example: `docker run -p 3000:3000 myapp`

docker stop – stop container
Example: `docker stop container_id`

docker logs – view logs
Example: `docker logs container_id`

---

9. Process & Services

ps aux – list processes
Example: `ps aux | grep node`

kill – stop a process
Example: `kill 1234`

systemctl start – start a service
Example: `sudo systemctl start nginx`

systemctl restart – restart service
Example: `sudo systemctl restart nginx`

systemctl status – check service status
Example: `sudo systemctl status nginx`

---

10. Zipping & Unzipping

tar -czvf – compress folder
Example: `tar -czvf backup.tar.gz project/`

tar -xzvf – extract
Example: `tar -xzvf backup.tar.gz`

zip -r – create zip
Example: `zip -r code.zip src/`

unzip – extract zip
Example: `unzip code.zip`

---

11. Searching Files/Inside Files

grep – search text
Example: `grep "error" server.log`

grep -r – recursive search
Example: `grep -r "database" ./src`

find – find files
Example: `find . -name "*.js"`

---

12. Environment Variables

export – set variable
Example: `export PORT=5000`

echo – display variable
Example: `echo $PORT`

env – list env vars
Example: `env`

---

13. Logs

tail – view last lines
Example: `tail server.log`

tail -f – live logs (very useful)
Example: `tail -f server.log`

journalctl -u – service logs
Example: `journalctl -u nginx`