

# 1. File and Directory Management

- **ls** – List directory contents
- **cd** – Change directory
- **pwd** – Print working directory
- **cp** – Copy files and directories
- **mv** – Move or rename files and directories
- **rm** – Remove files or directories
- **mkdir** – Make directories
- **rmdir** – Remove empty directories
- **touch** – Change file timestamps or create empty files
- **find** – Search for files in a directory hierarchy
- **locate** – Find files by name
- **tree** – Display directories in a tree-like format
- **chmod** – Change file permissions
- **chown** – Change file owner and group
- **chgrp** – Change group ownership
- **stat** – Display file or file system status

# 2. File Viewing and Editing

- **cat** – Concatenate and display file content
- **tac** – Concatenate and display file content in reverse
- **more** – View file content interactively (page by page)
- **less** – View file content interactively (scrollable)
- **head** – Output the first part of a file
- **tail** – Output the last part of a file
- **nano** – Text editor (terminal-based)
- **vim** / **vi** – Advanced text editors
- **emacs** – Text editor
- **grep** – Search text using patterns
- **sed** – Stream editor for filtering and transforming text
- **awk** – Pattern scanning and processing language
- **cut** – Remove sections from each line of files
- **sort** – Sort lines of text files

- **uniq** – Report or omit repeated lines

### 3. Process Management

- **ps** – Report a snapshot of current processes
- **top** – Display Linux tasks
- **htop** – Interactive process viewer (advanced top)
- **kill** – Send a signal to a process, typically to terminate
- **killall** – Terminate processes by name
- **bg** – Resume a suspended job in the background
- **fg** – Bring a job to the foreground
- **jobs** – List active jobs
- **nice** – Run a program with modified scheduling priority
- **renice** – Alter priority of running processes
- **uptime** – Show how long the system has been running
- **time** – Measure program running time

### 4. Disk Management

- **df** – Report file system disk space usage
- **du** – Estimate file space usage
- **fdisk** – Partition table manipulator for Linux
- **lsblk** – List information about block devices
- **mount** – Mount a file system
- **umount** – Unmount a file system
- **parted** – A partition manipulation program
- **mkfs** – Create a file system
- **fsck** – File system consistency check and repair
- **blkid** – Locate/print block device attributes

### 5. Networking

- **ifconfig** – Configure network interfaces
- **ip** – Show/manipulate routing, devices, and tunnels
- **ping** – Send ICMP Echo requests to network hosts

- **netstat** – Network statistics
- **ss** – Socket statistics (faster than netstat)
- **traceroute** – Trace the route packets take to a network host
- **nslookup** – Query Internet name servers interactively
- **dig** – DNS lookup utility
- **wget** – Non-interactive network downloader
- **curl** – Transfer data with URLs
- **scp** – Secure copy files between hosts
- **ssh** – Secure shell for remote login
- **ftp** – File Transfer Protocol client

## 6. User and Group Management

- **useradd** – Add a user to the system
- **usermod** – Modify a user account
- **userdel** – Delete a user account
- **groupadd** – Add a group to the system
- **groupdel** – Delete a group
- **passwd** – Change user password
- **chage** – Change user password expiry information
- **whoami** – Print the current logged-in user
- **who** – Show who is logged in
- **w** – Show who is logged in and what they're doing
- **id** – Display user and group information
- **groups** – Show user's groups

## 7. System Information and Monitoring

- **uname** – Print system information
- **hostname** – Show or set the system's hostname
- **uptime** – How long the system has been running
- **dmesg** – Boot and system messages
- **free** – Display memory usage
- **top** – Display Linux tasks
- **vmstat** – Report virtual memory statistics

- **lscpu** – Display information about the CPU architecture
- **lsusb** – List USB devices
- **lspci** – List PCI devices
- **lshw** – List hardware configuration

## 8. Archiving and Compression

- **tar** – Archive files
  - `tar -czf archive.tar.gz /path/to/directory` – Compress files using gzip
  - `tar -xzf archive.tar.gz` – Extract gzipped tarball
  - `tar -cf archive.tar /path/to/directory` – Create a tarball
  - `tar -xf archive.tar` – Extract tarball
- **zip** – Package and compress files into a ZIP archive
- **unzip** – Extract files from a ZIP archive
- **gzip** – Compress files using the gzip algorithm
- **gunzip** – Decompress files compressed with gzip
- **bzip2** – Compress files using the bzip2 algorithm
- **bunzip2** – Decompress files compressed with bzip2
- **xz** – Compress files using the xz algorithm
- **unxz** – Decompress files compressed with xz

## 9. Package Management (Depends on Distribution)

### Debian-based (e.g., Ubuntu)

- **apt-get** – APT package handling utility
  - `apt-get install <package>` – Install a package
  - `apt-get update` – Update package list
  - `apt-get upgrade` – Upgrade installed packages
  - `apt-get remove <package>` – Remove a package
- **apt-cache** – Query APT cache
  - `apt-cache search <package>` – Search for a package
  - `apt-cache show <package>` – Show package details

### Red Hat-based (e.g., CentOS, Fedora)

- **yum** – Package manager for RPM-based systems
  - yum install <package> – Install a package
  - yum update – Update installed packages
  - yum remove <package> – Remove a package
- **dnf** – Next-generation package manager (Fedora, CentOS 8+)
  - dnf install <package> – Install a package
  - dnf update – Update installed packages
  - dnf remove <package> – Remove a package

## General Commands

- **rpm** – RPM package manager
  - rpm -i <package.rpm> – Install an RPM package
  - rpm -e <package> – Remove an RPM package
- **dpkg** – Debian package manager
  - dpkg -i <package.deb> – Install a Debian package
  - dpkg -r <package> – Remove a Debian package

## System Services and Daemon Management

- **systemctl** – Control the systemd system and service manager
  - systemctl start <service> – Start a service
  - systemctl stop <service> – Stop a service
  - systemctl restart <service> – Restart a service
  - systemctl enable <service> – Enable a service to start on boot
  - systemctl disable <service> – Disable a service from starting on boot
  - systemctl status <service> – Check service status
- **service** – Older service management command (used in non-systemd systems)
  - service <service> start – Start a service
  - service <service> stop – Stop a service
  - service <service> restart – Restart a service
  - service <service> status – Check service status

## 10. Scheduling Tasks

- **cron** – Daemon for running scheduled commands
  - `crontab -e` – Edit cron jobs for the current user
  - `crontab -l` – List the current user's cron jobs
  - `crontab -r` – Remove the current user's cron jobs
- **at** – Run commands at a specified time
  - `at 09:00` – Schedule a command to run at 09:00 AM
- **batch** – Run commands when the system load is low
- **sleep** – Delay for a specified time
  - `sleep 5s` – Sleep for 5 seconds

## 11. File Permissions and Security

- **chmod** – Change file permissions
- **chown** – Change file owner and group
- **chgrp** – Change the group ownership of a file
- **umask** – Set default permissions for new files
- **setfacl** – Set file access control lists (ACL)
- **getfacl** – Get file access control lists (ACL)
- **sudo** – Execute a command as another user (usually root)
- **visudo** – Edit the sudoers file safely
- **passwd** – Change a user's password
- **sudoers** – Manage sudo access for users
- **gpasswd** – Administer group password
- **ss** – Display socket statistics (for secure network connections)

## 12. System Backup and Restore

- **rsync** – Remote file and directory synchronization
  - `rsync -avz source/ destination/` – Synchronize files
  - `rsync -avz -e ssh source/ user@remote:/destination/` – Sync over SSH
- **cpio** – Copy files to and from archives
- **dd** – Low-level copying and backup of entire filesystems

- `dd if=/dev/sda of=/path/to/backup.img` – Backup a disk/partition
- `dd if=/path/to/backup.img of=/dev/sda` – Restore a disk/partition

## 13. System Diagnostics and Troubleshooting

- **dmesg** – Print the kernel ring buffer messages (system boot and hardware-related messages)
- **journalctl** – Query and view logs from systemd's journal
- **strace** – Trace system calls and signals
  - `strace <command>` – Trace a command's system calls
- **lsof** – List open files (useful for debugging)
  - `lsof <file>` – Show processes using a specific file
- **vmstat** – Report virtual memory statistics
- **iostat** – Report CPU and I/O statistics
- **mpstat** – Report CPU usage statistics
- **pidstat** – Report statistics by process
- **free** – Display memory usage
- **uptime** – How long the system has been running
- **watch** – Execute a program periodically, showing output
  - `watch -n 1 free` – Watch memory usage every second
- **lshw** – List hardware configuration
- **htop** – Interactive process viewer (better than top)
- **netstat** – Network statistics (deprecated in favor of ss)
- **ss** – Show socket statistics (more efficient than netstat)

## 14. Networking & Remote Management

- **ifconfig** – Configure network interfaces (older command, replaced by ip)
- **ip** – A more modern alternative for managing network interfaces and routing
  - `ip addr` – Show IP addresses
  - `ip link` – Show or manipulate network interfaces

- `ip route` – Show or manipulate routing tables
- **ss** – Display socket statistics (useful for diagnosing network issues)
- **nmap** – Network exploration tool (can be used for security auditing)
- **telnet** – User interface to the TELNET protocol (less common nowadays)
- **nc** (Netcat) – Network utility for reading and writing from network connections
  - `nc -l -p 1234` – Listen on port 1234
  - `nc <host> <port>` – Connect to a host and port
- **iptables** – Administration tool for IPv4 packet filtering and NAT (Network Address Translation)
- **firewalld** – Frontend for managing firewall rules (used in some distros like Fedora and CentOS)
- **ufw** – Uncomplicated firewall (front-end for iptables)
  - `ufw enable` – Enable firewall
  - `ufw allow <port>` – Allow traffic on a specific port
- **tcpdump** – Command-line packet analyzer
- **curl** – Transfer data from or to a server using various protocols (HTTP, FTP, etc.)
- **wget** – Download files from the web via HTTP, HTTPS, FTP
- **scp** – Secure copy over SSH (used to copy files between systems)
  - `scp file.txt user@remote:/path/to/destination/` – Copy file to remote server
- **rsync** – Remote file and directory synchronization (often used for backups)
  - `rsync -avz /local/path/ remote:/remote/path/` – Sync directories

## 15. Text Processing Utilities

- **grep** – Search for patterns within files
  - `grep 'pattern' file.txt` – Search for a pattern in a file
  - `grep -r 'pattern' /dir/` – Recursively search for a pattern



- **sed** – Stream editor for filtering and transforming text
  - `sed 's/old/new/g' file.txt` – Replace old with new globally
- **awk** – A powerful text processing language
  - `awk '{print $1}' file.txt` – Print the first column of each line in a file
- **cut** – Remove sections from each line of a file
  - `cut -d ':' -f 1 /etc/passwd` – Print the first field of each line, delimited by ":"
- **sort** – Sort lines of text files
  - `sort file.txt` – Sort file content in ascending order
- **uniq** – Report or omit repeated lines in a file
  - `sort file.txt | uniq` – Sort and remove duplicate lines
- **tee** – Read from standard input and write to standard output and files
  - `echo "text" | tee file.txt` – Write to file and show output on screen
- **tr** – Translate or delete characters
  - `echo "hello" | tr 'a-z' 'A-Z'` – Convert lowercase to uppercase
- **paste** – Merge lines of files
  - `paste file1.txt file2.txt` – Combine lines of file1 and file2 side by side
- **wc** – Word, line, character, and byte count
  - `wc -l file.txt` – Count lines in a file
  - `wc -w file.txt` – Count words in a file

## 16. System Shutdown and Reboot

- **shutdown** – Shut down the system
  - `shutdown -h now` – Immediately shut down
  - `shutdown -r now` – Reboot the system
  - `shutdown -h +10` – Shut down after 10 minutes
- **reboot** – Reboot the system
- **halt** – Halt the system immediately (equivalent to turning off power)
- **poweroff** – Power off the system

- **init** – Change the runlevel (old-style system manager)
  - init 0 – Shutdown
  - init 6 – Reboot

## 17. File System Mounting and Management

- **mount** – Mount a file system
  - mount /dev/sda1 /mnt – Mount partition to a directory
- **umount** – Unmount a file system
  - umount /mnt – Unmount the file system mounted at /mnt
- **fstab** – File system table (configuration file for mounting file systems)
  - /etc/fstab – View and configure persistent mount points
- **blkid** – Display block device attributes
- **fsck** – Check and repair a file system
  - fsck /dev/sda1 – Check and repair /dev/sda1

## 18. Filesystem Permissions and Security

- **chmod** – Change file permissions
  - chmod 755 file.txt – Give read, write, and execute permissions to owner, and read-execute permissions to others
- **chown** – Change file owner and group
  - chown user:group file.txt – Change owner and group of a file
- **chgrp** – Change group ownership of a file
  - chgrp group file.txt – Change the group of a file
- **umask** – Set default permissions for new files
  - umask 022 – Set default permissions for newly created files to 755
- **setfacl** – Set access control lists (ACL) for file permissions
- **getfacl** – Get access control lists (ACL) for file permissions

# 19. Containerization and Orchestration

## Docker

- **docker** – Docker command-line interface (CLI) for managing containers
  - **docker run <image>** – Run a container from an image
  - **docker ps** – List running containers
  - **docker ps -a** – List all containers, including stopped ones
  - **docker build -t <image\_name> .** – Build an image from a Dockerfile
  - **docker exec -it <container\_id> bash** – Start an interactive bash shell inside a running container
  - **docker stop <container\_id>** – Stop a container
  - **docker rm <container\_id>** – Remove a container
  - **docker logs <container\_id>** – View logs of a container
  - **docker images** – List available images
  - **docker rmi <image\_name>** – Remove an image
  - **docker network ls** – List Docker networks
  - **docker-compose** – Manage multi-container Docker applications
    - **docker-compose up** – Start up a multi-container environment
    - **docker-compose down** – Stop and remove containers created by docker-compose
    - **docker-compose logs** – View logs from containers managed by docker-compose

## Kubernetes (k8s)

- **kubectl** – Command-line tool for interacting with Kubernetes clusters
  - **kubectl get pods** – List pods in the current namespace
  - **kubectl get nodes** – List nodes in the cluster
  - **kubectl get services** – List services in the cluster

- **kubectl apply -f <file>.yaml** – Apply configuration from a file (e.g., a deployment or pod configuration)
- **kubectl create -f <file>.yaml** – Create a resource from a file
- **kubectl delete -f <file>.yaml** – Delete a resource defined in a file
- **kubectl exec -it <pod\_name> -- bash** – Execute a command inside a pod (e.g., open a shell)
- **kubectl logs <pod\_name>** – View the logs of a pod
- **kubectl describe pod <pod\_name>** – Get detailed information about a pod
- **kubectl scale deployment <deployment\_name> -- replicas=<number>** – Scale a deployment to the desired number of replicas
- **kubectl rollout restart deployment <deployment\_name>** – Restart a deployment
- **kubectl port-forward pod <pod\_name> <local\_port>:<remote\_port>** – Forward a port from a pod to localhost

## Helm

- **helm** – Kubernetes package manager for deploying applications
  - **helm install <release\_name> <chart\_name>** – Install a Helm chart
  - **helm upgrade <release\_name> <chart\_name>** – Upgrade a Helm release
  - **helm list** – List all Helm releases
  - **helm delete <release\_name>** – Delete a Helm release
  - **helm search <chart\_name>** – Search for a Helm chart

## 20. Automation and Configuration Management

### Ansible

- **ansible** – Automation tool for configuration management

- **ansible all -m ping** – Ping all hosts defined in the inventory
- **ansible-playbook playbook.yml** – Run an Ansible playbook
- **ansible -m command -a 'command' <host>** – Run a single command on a target host
- **ansible-playbook --check playbook.yml** – Dry-run a playbook to see what would change
- **ansible-playbook --limit <host> playbook.yml** – Run a playbook on a specific host or group
- **ansible-playbook --extra-vars "key=value"** – Pass extra variables to a playbook

## Terraform

- **terraform** – Infrastructure as code tool for provisioning and managing cloud resources
  - **terraform init** – Initialize a working directory for Terraform configuration
  - **terraform plan** – Show an execution plan (preview of what changes will be made)
  - **terraform apply** – Apply the changes described in a Terraform configuration
  - **terraform destroy** – Destroy infrastructure created by Terraform
  - **terraform validate** – Validate the configuration files
  - **terraform show** – Show the current state of the infrastructure

## Puppet

- **puppet** – Configuration management tool
  - **puppet apply <manifest.pp>** – Apply a Puppet manifest locally
  - **puppet agent --test** – Test the Puppet agent (can be used to run a one-off run)

- **puppet resource** – Show the current state of resources (files, services, etc.)

## 21. CI/CD Tools and Commands

### Jenkins

- **jenkins** – Continuous integration tool
  - **java -jar jenkins.war** – Start Jenkins from a WAR file
  - Access Jenkins through `http://localhost:8080` by default

### GitLab CI

- **.gitlab-ci.yml** – Configuration file for GitLab CI/CD pipelines (typically resides in your repository)
  - **gitlab-runner register** – Register a new runner with GitLab
  - **gitlab-runner run** – Run the GitLab Runner to process jobs

### GitHub Actions

- GitHub Actions uses YAML configuration files (typically located in `.github/workflows/`)
  - **actions/checkout@v2** – Checkout the repository code in your CI pipeline
  - **actions/setup-node@v2** – Setup Node.js for use in a pipeline
  - **docker/setup-buildx-action@v1** – Set up Docker Buildx for building multi-platform images

## 22. Cloud Services

### AWS CLI (Amazon Web Services)

- **aws** – Command-line tool for managing AWS services
  - **aws configure** – Configure AWS CLI with your credentials

- **aws s3 cp file.txt s3://bucket-name/** – Copy a file to an S3 bucket
- **aws ec2 describe-instances** – Describe EC2 instances
- **aws ec2 start-instances --instance-ids <id>** – Start an EC2 instance
- **aws ec2 stop-instances --instance-ids <id>** – Stop an EC2 instance
- **aws s3 sync** – Sync directories with an S3 bucket

## Azure CLI (Microsoft Azure)

- **az** – Command-line tool for managing Azure services
  - **az login** – Log in to your Azure account
  - **az vm list** – List all virtual machines
  - **az vm start --name <vm\_name> --resource-group <resource\_group>** – Start an Azure VM
  - **az storage blob upload** – Upload files to an Azure blob storage
  - **az group create** – Create a new resource group in Azure

## Google Cloud SDK (gcloud)

- **gcloud** – Command-line tool for Google Cloud Platform
  - **gcloud auth login** – Log in to Google Cloud
  - **gcloud compute instances list** – List compute instances
  - **gcloud compute instances stop <instance\_name>** – Stop a Google Cloud VM instance
  - **gcloud app browse** – Open the current Google App Engine application in a browser

# 23. Logging and Monitoring

## Prometheus

- **prometheus** – Open-source system monitoring and alerting toolkit
  - **prometheus** – Start Prometheus server (usually runs as a service in the background)
  - **prometheus --config.file=<config\_file>** – Start Prometheus with a specific config file

## Grafana

- **grafana-cli** – Command-line interface for managing Grafana plugins
  - **grafana-cli plugins install <plugin-name>** – Install a plugin in Grafana

## ELK Stack (Elasticsearch, Logstash, Kibana)

- **elasticsearch** – Search engine for logging and data analytics
  - **curl -XGET 'localhost:9200/\_cluster/health?pretty'** – Get cluster health status
- **logstash** – Server-side data processing pipeline
  - **logstash -f <config\_file>** – Run Logstash with the specified configuration file
- **kibana** – Web interface for visualizing Elasticsearch data
  - Kibana is generally accessed through a web browser (<http://localhost:5601>)