Root/test12#$

Sysadmin/abc

-Dotnet –info

-whereis dotnet

=>List all installed packages on Ubuntu

dpkg --get-selections

dpkg --get-selections | grep -i dotnet

=>Install .NET SDK (includes ASP.NET Core)

-wget https://packages.microsoft.com/config/ubuntu/22.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb

-sudo dpkg -i packages-microsoft-prod.deb

-sudo apt update

-sudo apt install -y dotnet-sdk-8.0

=>Verify Installation

-Dotnet –version

lsb\_release –a #Ver

cat /etc/os-release #Ver

hostnamectl #host

dotnet --info

**🖥️ System Information**

CopyEdit

uname -a # Kernel and system info

hostname # Show hostname

uptime # How long the system has been running

whoami # Show current user

top # Real-time process viewer

htop # Advanced version of top (install it first)

neofetch # Stylish system info (install with `sudo apt install neofetch`)

**📁 File & Directory Commands**

CopyEdit

ls # List files in directory

ls -l # Detailed listing permission

cd [dir] # Change directory

pwd # Print current directory

mkdir [dir] # Create new directory

rm [file] # Remove file

rm -r [dir] # Remove directory and contents

cp [src] [dest] # Copy files

mv [src] [dest] # Move or rename files

touch [file] # Create empty file

find . -name "\*.txt" # Find files

find . -name "\*.txt" | less # Find files one by one wise

find . -name "\*.txt" | more # Find files down arrow to page wise

**🧰 Package Management (APT)**

CopyEdit

sudo apt update # Refresh package index

sudo apt upgrade # Upgrade all packages

sudo apt install [package] # Install a package

sudo apt remove [package] # Remove a package

sudo apt purge [package] # Remove including config

sudo apt autoremove # Remove unused packages

dpkg -i [package.deb] # Install a .deb file

**🧑‍💻 User Management**

CopyEdit

adduser [name] # Add new user

sudo passwd [user] # Set or change password

usermod -aG sudo [user] # Give user sudo rights

who # Show who is logged in

**🔐 Permissions**

CopyEdit

chmod +x [file] # Make file executable

chmod 755 [file] # Set permission

chown user:user [file] # Change ownership

**🔎 Searching**

CopyEdit

grep "text" [file] # Search in file

grep -r "text" . # Recursive search in directory

find / -name [file] # Find file by name

**🌐 Networking**

CopyEdit

ip a # Show IP addresses

ping [host] # Check network connection

ifconfig # Network interfaces (install with `net-tools`)

curl [url] # Fetch URL

wget [url] # Download file

netstat -tulpn # Show open ports (install `net-tools`)

**📦 System Monitoring & Processes**

CopyEdit

ps aux # Show all running processes

kill [PID] # Kill a process

kill -9 [PID] # Force kill

df -h # Disk usage

du -sh \* # Directory size

free -h # RAM usage

**🔁 Process Management**

CopyEdit

jobs # Show background jobs

fg # Bring job to foreground

bg # Send job to background

**🔧 Disk & USB Tools**

CopyEdit

mount # Mount a device

umount # Unmount a device

lsblk # List block devices

fdisk -l # Show disk partitions

df -h # Disk free space

**🔄 System Management**

CopyEdit

reboot # Reboot system

shutdown now # Shutdown immediately

shutdown -h +10 # Shutdown in 10 minutes

sudo systemctl restart [service] # Restart a service

sudo systemctl status [service] # Service status

**📦 Snap & Flatpak (Alternative Package Managers)**

CopyEdit

sudo snap install [package] # Install via snap

sudo snap remove [package]

flatpak install flathub [package] # Install via flatpak

flatpak run [package]

**🧪 Development Tools**

CopyEdit

gcc file.c -o file # Compile C program

python3 script.py # Run Python script

node file.js # Run Node.js script

git clone [repo] # Clone git repo

**📜 Other Useful Commands**

CopyEdit

history # Show command history

alias ll='ls -la' # Create alias

clear # Clear terminal screen

man [command] # Manual for command

Would you like a downloadab

**Bash**, **KSH**, and **CSH** are all **types of Unix/Linux shells** — basically command-line interpreters that allow users to interact with the operating system by typing commands.

Virtualisation allows one physical machine (like a server or computer) to run **multiple virtual machines (VMs)**, each behaving like an independent system with its own operating system and applications.

Shell: Is a Command-Line Interpreter that connects a user to Operating System and allows to execute the commands.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| date | Check today date | | |  | |
| cat /etc/shells | Let us check how types of shells are available in our syste | | |  | |
| Ps $$ | Let us check our shell process ID & kill the shell process from another session   * ps: Stands for **process status** – it's used to view information about active processes. * $$: This is a special variable in **bash** and many other shells. It holds the **process ID (PID)** of the currently running shell. | | | PID TTY TIME CMD  1234 pts/0 00:00:00 bash  PID: 1226  TTY: tty1  STAT: Ss  Time: 0:00  Command: -bash | |
| echo $SHELL | Let us check what shell we have? | | | /bin/bash | |
| whereis os-release | Os-release: /etc/os-release | | |  | |
| uptime -help | get information on “uptime: | | |  | |
| man date | s get more information on “date | | |  | |
| pwd | Where do we stand in the system as a user | | |  | |
| Cal | Calendar | | |  | |
| Ssh [root@10.0.0.4](mailto:root@10.0.0.4) |  | | |  | |
| Sudo poweroff | Power off the linux machine | | |  | |
| exit |  | | |  | |
| uname –a | All System Information: | | | Linux myserver 5.14.0-503.14.1.el9\_5.x86\_64 #1 SMP PREEMPT\_DYNAMIC Fri Nov 15 12:04:32 UTC 2024 x86\_64 x86\_64 x86\_64 GNU/Linux | |
| uname -r | KERNEL RELEASE VERSION | | | 5.14.0-503.14.1.el9\_5.x86\_64 | |
| lscpu | **display detailed information about the CPU architecture** of your system.   To check if your CPU supports 64-bit.   To see how many physical/logical CPUs are available.   To verify virtualization support.   To troubleshoot performance or compatibility issues. | | | Architecture: x86\_64  CPU op-mode(s): 32-bit, 64-bit  Byte Order: Little Endian  CPU(s): 8  On-line CPU(s) list: 0-7  Thread(s) per core: 2  Core(s) per socket: 4  Socket(s): 1  Vendor ID: GenuineIntel  Model name: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz  CPU MHz: 1992.000  L1d cache: 32K  L1i cache: 32K  L2 cache: 256K  L3 cache: 8192K | |
| cat /proc/cpuinfo | **detailed information about each processor core** on your system. It reads data from the **/proc/cpuinfo** virtual file, which contains info provided by the Linux kernel. | | | processor : 0  vendor\_id : GenuineIntel  cpu family : 6  model : 158  model name : Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz  stepping : 10  cpu MHz : 1992.000  cache size : 8192 KB  physical id : 0  siblings : 8  core id : 0  cpu cores : 4  apicid : 0  initial apicid : 0  fpu : yes  flags : fpu vme de pse tsc ... | |
| pwd |  | | |  | |
| Ls |  | | |  | |
| cd |  | | |  | |
| cat |  | | |  | |
| su - username | **Switch user** | | |  | |
| Echo $PATH |  | | |  | |
| Echo $SHELL |  | | |  | |
| echo $HOST |  | | |  | |
| echo $USER |  | | |  | |
|  |  | | |  | |
| Commands to Set Up Apache on Rocky Linux  Once you're logged in to your EC2 instance using PuTTY: | | | | | |
| sudo dnf update -y | | | Update the system | |  |
| sudo dnf install httpd -y | | | **Install Apache (httpd)** | |  |
| sudo systemctl start httpd | | | Start Apache | |  |
| sudo systemctl enable httpd | | | Enable Apache to start on boot | |  |
| sudo firewall-cmd --permanent --add-service=http  sudo firewall-cmd –reload | | | Allow HTTP through the firewall (if firewalld is running)  ***Note:*** *This step may not be needed if your AWS Security Group already allows port 80 (HTTP). But it’s safe to run anyway.* | |  |
| http://<your-ec2-public-ip> | | | Test Apache  pen your browser and go to | |  |
|  | |  | |  | |
|  | |  | |  | |
|  | |  | |  | |

Ps $$:

|  |  |
| --- | --- |
| * **PID** | * **Process ID** – a unique number assigned to each running process by the operating system. In this example, 1234 is the ID of the bash shell. |

|  |  |
| --- | --- |
| * **TTY** | * **Terminal Type** – the terminal associated with the process. pts/0 refers to a "pseudo-terminal" (like when you're using a terminal emulator or SSH session). |

|  |  |
| --- | --- |
| * **TIME** | * **CPU Time** – the total amount of CPU time the process has used since it started. 00:00:00 means it hasn't used any noticeable CPU time yet. |

|  |  |
| --- | --- |
| * **CMD** | * **Command** – the command that started the process. In this case, it's bash, which is your shell. |

myserver Hostname of machine

5.14.0-503.14.1.el9\_5.x86\_64 Kernel version #1 Kernel built version SMP Symmetric multiprocessing support PREEMPT\_DYNAMIC Preemption model used by the kernel Fri Nov 15 12:04:32 UTC 2024 Build time and date of the kernel X86\_64 (3 times) Machine hardware name, Processor type, Hardware platform GNU/Linux Operating System

|  |
| --- |
| #ls : #ls –a : #ls –al : #al –al /abc.txt  #ls –l a.txt : ll a.txt same results shows  #ls -il files and directories with permission  permissions, ownership, timestamps, and disk block locations  #ls /dev sda\* list of files starting from sda  pwd print working directory  cat a.txt view file  #cat /dev os-release  #ls /etc/os-release to view linux OS detail  #uptime How long the System has been running with date, time user info.  #uptime –p up 43 minutes  #runlevel what services and processes are running.  #top **to monitor system processes in real time** — like Task Manager for the terminal.  ps -e command lists **all running processes** on the system.  ps = process status  -e = show every process (same as –a)  hostname –I just print IP Address i.e. 192.168.1.100  hostnamectl to check the host name  hostname to check the host name  sudo hostnamectl set-hostname webserver1 to change hostname permanently - not change  hostnamectl set-hostname webserver1 to change hostname  hostname newhostname to change hostname  uname –r prints **only the kernel version** of your Linux system.  uname –a **all detail system information** related to the kernel and machine.  uname –m shows the **machine hardware name**  lsb\_release –a command not found  kernelctl stuatus command not found  cpuinfo command not found  cat /proc/cpu  lscpu **summary of the CPU architecture and hardware info.**, including number of cores, threads,  model, and more.  slblk information about **block devices** such as hard drives, SSDs, USB drives, and partitions.  lsblk shows the tree **from disks down to partitions**  Slblk –s reverses that — it shows a **device (like a partition or LVM)** and **all the block devices it**  **depends on** (its parents).  #ls –l /dev/input shows input devices (keyboard, mouse, touchpads) as **character devices**  free shows **memory usage** (RAM and swap) on your system.  free –m shows output in Megabytes  free –g shows output in Gigabytes  #du Show size of current directory and its subdirectories:  **"disk usage"** /size and is used to check how much space files and directories are using.  #du –h Show sizes in **human-readable** format:  #du –sh Show **only the total size** of a directory:  #du –h /var Show **the total size** of a var folder  #du -h abc.txt Show **the total size** of a abc.txt file.  #du -h --max-depth=1 /var Show size of all subdirectories, human-readable:  #df to **display the amount of disk space** available and used on **mounted filesystems**.  #df –h **human-readable** (displays sizes in KB, MB, GB)  df –hT Show all file systems with type  #ifconfig ifconfig: command not found  ifconfig is part of the old net-tools package, which is **deprecated** on most  modern Linux distributions (Ubuntu, Debian, CentOS, etc.) and is **not installed by**  **default anymore**.  ifconfig is part of the legacy net-tools package.  #sudo dnf install net-tools  #ifconfig  #mkdir test  #rmdir test  #echo ‘This is line 1’ > abc.txt  #echo ‘add lines >> abc.txt  #cat abc.txt  #rm abc.txt  rm file1.txt file2.log file3.jpg  #ls –l a\* (list of file start with a)  #chmod ugo+rwx abc.txt (add permission to User,group,other to read,write,execute)  #chmod ugo-rwx abc.txt (remove permission to User,group,other to read,write,execute)  #touch abc.txt (create new file)  #vim abc.txt command not found  # sudo dnf install vim  #vim a.txt  #i (press i) for edit mode / type your text / Press ESC to return mode  :w (save), :q (quit), :wq (save and quit), :ql (quit without saving)  #cd change directory  #cp abc.txt /test copy file into test dir  #cp –r abc.txt /test copy file into test dir  #cp a\* /test copy files start with ‘a’ into test dir  #rm -rf \* remove all files  #rm -rf /home/noman remove dir  httpd **HTTP Daemon** — it's a **web server software** (computer act like a web server)  that accept **HTTP requests** (website URL) in browsers) and serves **web pages, files, or data** in  response.  # dnf install httpd –y (it install httpd package – for root user)  #sudo dnf install httpd (it install httpd package – for others users)  #do dnf install httpd –y (it install httpd package – for others users without asking)  #sudo systemctl start httpd (Start the httpd service)  #sudo systemctl enable httpd (**Enable httpd to start on boot**)  #sudo systemctl status httpd (Check status (optional))  #sudo firewall-cmd --permanent (add-service=http)  #sudo firewall-cmd --permanent (add-service=https)  #sudo firewall-cmd (reload) |
| #ls **list the contents of a directory**.   | **Command** | **Description** | | --- | --- | | ls -l | Long listing format (shows permissions, owner, size, etc.) | | ls -a | Shows **all** files, including hidden ones (those starting with .) | | ls -lh | Human-readable sizes (K, M, G) | | ls -R | Recursively lists subdirectories | | ls -lt | Sorts by modification time, newest first | |
| **#runlevel** defines what services and processes are running.  In short determines the system's mode of operation, like single-user mode, multi-user  mode, or graphical mode.   | **Runlevel** | **Mode** | | --- | --- | | 0 | Halt / Shutdown | | 1 | Single-user mode (Rescue) | | 3 | Multi-user mode (Text-based) | | 5 | Multi-user mode (Graphical) | | 6 | Reboot | |
| #uname –r prints **only the kernel version** of your Linux system.  5.14.0-327.el9.x86\_64  5.14.0 → Kernel version number  -327.el9 → Distribution-specific build or patch version (here, for Rocky Linux 9)  x86\_64 → Kernel architecture (64-bit) |
| #uname –a **all system information** related to the CUP (machine) and kernel.  Linux rocky-linux 5.14.0-327.el9.x86\_64 #1 SMP Thu Jul 27 13:15:32 UTC 2025 x86\_64 x86\_64 x86\_64 GNU/Linux   | **Part** | **Meaning** | | --- | --- | | Linux | Kernel name | | rocky-linux | Hostname | | 5.14.0-327.el9.x86\_64 | Kernel version | | #1 SMP Thu Jul 27 13:15:32 UTC 2025 | Kernel build info (build number, SMP support, build date) | | x86\_64 (three times) | Machine hardware, processor, platform | | GNU/Linux | Operating system | |
| #uname –m  x86\_64   | **Output** | **Architecture** | **Description** | | --- | --- | --- | | x86\_64 | 64-bit | Most common for modern systems | | i686 | 32-bit | Older 32-bit Intel/AMD CPUs | | aarch64 | 64-bit ARM | Used on newer ARM devices, servers | | armv7l | 32-bit ARM | Used on Raspberry Pi and similar | |  |  |  | |  |  |  | |
| #lsblk information about **block devices** such as hard drives, SSDs, USB drives, and partitions.  NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS  sda 8:0 0 100G 0 disk  ├─sda1 8:1 0 50G 0 part /  ├─sda2 8:2 0 45G 0 part /home  └─sda3 8:3 0 5G 0 part [SWAP]  sr0 11:0 1 1024M 0 rom  #lsblk –s reverses that — it shows a **device (like a partition or LVM)** and **all the block devices it**  **depends on** (its parents).  sda  ├─sda1  ├─sda2  └─sda3 |
| #df –h **human-readable** (displays sizes in KB, MB, GB)  Filesystem Size Used Avail Use% Mounted on  /dev/sda1 50G 25G 23G 53% /  tmpfs 1.9G 2.0M 1.9G 1% /run  /dev/sdb1 100G 60G 37G 62% /mnt/data   | **Column** | **Meaning** | | --- | --- | | **Filesystem** | The device or virtual FS (e.g., /dev/sda1) | | **Size** | Total size of the filesystem | | **Used** | Space used | | **Avail** | Space available | | **Use%** | Usage percentage | | **Mounted on** | Where it is mounted in the directory tree | |
|  |
| When to use sudo:   * sudo is needed when you run commands that require administrative (root) privileges. Installing software (dnf install), modifying system files, starting/stopping system services, changing firewall settings — these all usually need sudo. * Regular users (non-root) cannot perform these actions without sudo.   Do *all* users use sudo?   * Only users who have been granted permission to use sudo (usually members of the wheel or sudo group) can run commands with sudo. * If a user is *not* in those groups, trying sudo will either:   + Ask for a password and then deny access, or   + Simply deny access outright. |
|  |
|  |

#ls –l permission, ownership, time, files/dir names

#ls –il total size, permission, ownership, time, files/dir names

|  |  |
| --- | --- |
| Static IP address | **that does not change** over time. It is manually assigned to a device (like a server, PC, router, or printer) and remains constant |
| dynamic IP address | which is automatically assigned by a DHCP server and can change over time. |
|  |  |

**copy folder with directories and files**

sudo useradd user1

sudo passwd user1

sudo useradd user2

sudo passwd user2

su user1

mkdir mywork

cd mywork

mkdir 012025

mkdir 022025

cd 012025

touch abc.txt xyz.txt

su root

sudo cp -r /home/user1/. /home/user2

sudo mv /home/user1/\* /home/user3 (to move dir and files to other folder)

**HOW DO YOU RECOVER ROOT PASSWORD**

1. Start (Power on) the System.

2. Wait for GRUB menu - press e to edit.

3. Find the line starting with linux and type the rd.break at the end.

4. Press Ctrl+x

5. mount –o remount,rw /sysroot

6. chroot /sysroot (change root to sysroot)

7. touch /.autorelabel

8. exit

9. exit

**DELETING USER**

The userdel command can’t delete a user if they are **currently logged in** or **have processes running**. You must first stop those processes.

#adduser noman

#userdel noman Delete the User

#Ps –u noman to see **all processes by the user**:

sudo kill -9 4472 Kill Specific Process (e.g. 4472)

sudo pkill -u noman Kill All User Processes

sudo userdel noman Delete the User

sudo userdel -r noman With deleting home directory

sudo rm -rf /home/abc **forcefully and recursively** remove everything under /home/abc.

getent passwd noman verify user has deleted

sudo userdel abc Deletes the user account, but **leaves the home directory**

sudo userdel -r abc Deletes the user **and** their home directory.

sudo pkill -u abc Kills all running processes of user abc, but doesn’t remove files.

Note: If just ran userdel without -r, their files remain in /home.

There are 5 Type of Users in Linux:

1. System User (This is Root User and the Admin)

2. Normal User (This is a user created by Root User)

3. Network User (this is a remote user that is logs-in Windows Active Directory or LDAP in the case of Linux Users}

4. System User based on an installed application like Oracle etc.

5. Sudo User (normal users who have admin or root privileges.

#w (show currently logged in user)

#who (show currently logged in user with detail i.e. user tty login date/time)

#whoami (show currently logged in user, only user name)

# cut -d: -f1 /etc/passwd (all users that exist, only name)

#getent passwd (all users that exist, with detail)

Note: the best way to switch user ‘Ctrl+Alt+F2’

#su - (root)

#su – noman (other user)

|  |
| --- |
| Prob: I login to root user then I logged in as 'su - noman' . when execute 'w' then only root user show.  Even though you're now running commands as noman (via su - noman), you're **still using the same terminal session that belongs to root**.   * The w command shows users **logged in through a TTY or PTY** (e.g., SSH sessions, console logins, terminal windows). * su - noman switches the user identity **within the same session** — it doesn't create a new login session.   So w continues to show **root**, because the controlling terminal still belongs to root.  Even though you're now running commands as noman (via su - noman), you're **still using the same terminal session that belongs to root**.   * The w command shows users **logged in through a TTY or PTY** (e.g., SSH sessions, console logins, terminal windows). * su - noman switches the user identity **within the same session** — it doesn't create a new login session.   So w continues to show **root**, because the controlling terminal still belongs to root.  Solution: How to Actually Log In as noman (and see in w) To see noman listed as a logged-in user in w, you need to **log in as noman from a new terminal or SSH session**, not via su. ✔ Option 1: Switch TTY  1. Press **Ctrl+Alt+F2** (or F3, F4...) to switch to another virtual terminal. 2. Log in as noman directly. |

To close user ‘noman’ (it will not display through #w command)

#pkill -u noman

#exit

$w

$whoami

$lastlog

#lastb (bad logins)

#last (List of Users Logged-in and Logged out / shutdown)

A **user group** is a collection of users in Linux that share **the same permissions** for accessing files, directories, or resources.

### Why Groups Matter:

* Makes **permission management easier**.
* Lets you assign access to multiple users **at once**.

$groups Check Groups for the Current User

$groups noman Check Groups for a Specific User

$id noman Detailed Group Info (Using id)

$cat /etc/group List All Groups on the System

$getent group List All Groups on the System

Solution: Add noman to the wheel Group

$su – log in as root

#usermod -aG wheel noman Add noman to the wheel group - This **grants sudo access** to the user.

#su – noman

$groups noman Have noman Log Out and Back In and confirm wheel group access

$ whoami it show ‘noman’

$sudo whoami it show ‘root’ That confirms noman can now use sudo.

$sudo grep wheel /etc/sudoers

sudo groupadd permitgroup Create the New Group

sudo usermod –aG permitgroup noman add user ‘noman’ to group

groups noman Make sure permitgroup is listed.

getent group permitgroup verify it created or not

sudo groupdel permitgroup delete group

**Create user when logging the msg display: Authentication failure**

getent passwd noman (Make sure the user was created successfully)

sudo useradd -m noman (If not, recreate the user)

sudo passwd noman (Set the password) or

echo "noman:abc" | sudo chpasswd (Set the password)

sudo passwd -S noman (Check Account Status)

sudo passwd -u noman (If status is L (locked), unlock it)

udo journalctl –xe (Check logs for exact error messages)