Farakh Farooq

[farakhfrq@gmail.com](mailto:farakhfrq@gmail.com)

✅ Step 1: Create Project Directory and Navigate Into It

Command:

mkdir /var/www/ProjectX

cd /var/www/ProjectX

Explanation:

- mkdir /var/www/ProjectX:

• Creates a new directory named 'ProjectX' inside /var/www.

• This will serve as the workspace for the development team.

- cd /var/www/ProjectX:

• Navigates into the ProjectX directory so that further file operations happen inside it.

✅ Step 2: Create Files for Frontend and Backend

Command:

touch index.html app.py README.md

Explanation:

- Creates three empty files:

• index.html – for frontend HTML content

• app.py – for backend Python application

• README.md – for project description/documentation

- The `touch` command is commonly used to create empty files quickly.

✅ Step 3: Check Current Working Directory

Command:

pwd

Explanation:

- Prints the current directory you are in (stands for "print working directory").

- Confirms that you're working inside `/var/www/ProjectX`.

Example Output:

/var/www/ProjectX

✅ Step 4: List Files with Detailed Information

Command:

ls -l

Explanation:

- Lists files in long format, showing:

• File permissions

• Number of links

• Owner

• Group

• File size

• Modification date and time

• File name

Example Output:

-rw-r--r-- 1 root root 0 Apr 26 10:00 app.py

-rw-r--r-- 1 root root 0 Apr 26 10:00 index.html

-rw-r--r-- 1 root root 0 Apr 26 10:00 README.md

✅ Step 5: Display System Disk Usage

Command:

df -h

Explanation:

- Displays the amount of disk space used and available on the file system.

- The `-h` flag means "human-readable", showing sizes in MB/GB.

- Helps determine whether you have enough space before deploying or storing large files.

✅ Step 6: View File Content

Command:

echo "Welcome to ProjectX" > README.md

cat README.md

Explanation:

- echo "Welcome to ProjectX" > README.md:

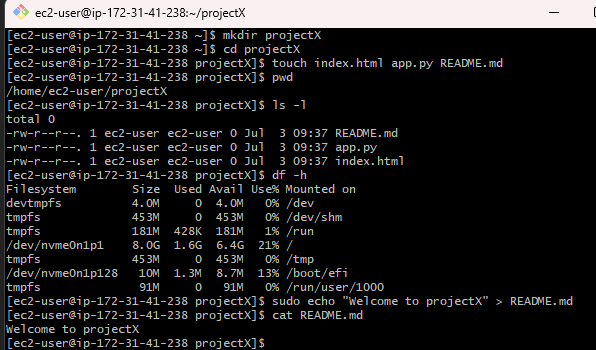
• Writes the text into the README.md file (overwrites any existing content).

- cat README.md:

• Displays the content of README.md to confirm the write was successful.

Example Output:

Welcome to ProjectX



✅ Step 1: Create a Developer Group and Users

Command:

groupadd devteam

useradd bhatti

useradd malik

usermod -aG devteam bhatti

usermod -aG devteam malik

Explanation:

- groupadd devteam:

• Creates a new group called 'devteam' for the developers.

- useradd bhatti / useradd malik:

• Creates two new user accounts named 'bhatti' and 'malik'.

- usermod -aG devteam bhatti:

• Adds 'bhatti' to the 'devteam' group without removing him from other groups.

- usermod -aG devteam malik:

• Adds 'malik' to the 'devteam' group in the same way.

✅ Step 2: Assign Group Ownership to Project Directory

Command:

chgrp -R devteam /var/www/ProjectX

Explanation:

- chgrp -R devteam /var/www/ProjectX:

• Changes the \*\*group ownership\*\* of the ProjectX directory and its contents to 'devteam'.

• The `-R` flag applies this change \*\*recursively\*\* to all files and subdirectories.

✅ Step 3: Set Appropriate Directory Permissions

Command:

chmod -R 770 /var/www/ProjectX

Explanation:

- chmod -R 770:

• Grants read, write, and execute permissions to the \*\*owner and group\*\*.

• Denies all permissions to \*\*others\*\*.

• Ensures that only members of the 'devteam' group can modify the project files.

✅ Step 4: Verify Directory Permissions

Command:

ls -ld /var/www/ProjectX

Explanation:

- Lists detailed information about the ProjectX directory including:

• Permissions

• Owner

• Group

• Date modified

• Directory name

Example Output:

drwxrwx--- 2 root devteam 4096 Apr 26 10:00 ProjectX

- This confirms that:

• Owner: root

• Group: devteam

• Permissions: read/write/execute for owner and group only

✅ Step 5: Check User Group Memberships

Command:

groups bhatti

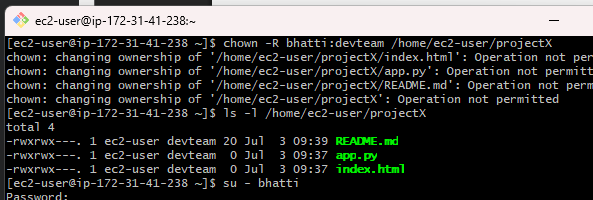
Explanation:

- Displays all the groups that the user 'bhatti' is a member of.

- Used to confirm that 'bhatti' was successfully added to the 'devteam' group.

Example Output:

bhatti : bhatti devteam



✅ Step 1: Change Ownership to User 'bhatti' and Group 'devteam'

Command:

chown -R bhatti:devteam /var/www/ProjectX

Explanation:

- Changes both the \*\*owner\*\* and the \*\*group\*\* of all files and folders inside /var/www/ProjectX.

- `-R` (recursive) ensures that the ownership is applied to all contents of the directory.

Result:

• Owner: bhatti

• Group: devteam

• Ensures the lead developer has full control over the project files.

✅ Step 2: Verify Ownership Changes

Command:

ls -l /var/www/ProjectX

Explanation:

- Lists detailed info for all files in the directory.

- Confirms that ownership is now set to:

• User: bhatti

• Group: devteam

Example Output:

-rw-rw---- 1 bhatti devteam 0 Apr 26 10:00 app.py

-rw-rw---- 1 bhatti devteam 0 Apr 26 10:00 index.html

-rw-rw---- 1 bhatti devteam 0 Apr 26 10:00 README.md

- The permissions also show that the \*\*owner and group\*\* can read and write the files.

✅ Step 3: Switch to User 'bhatti' and Create a New File

Command:

su - bhatti

cd /var/www/ProjectX

touch config.yaml

ls -l

Explanation:

- su - bhatti:

• Switches the current user session to 'bhatti'.

• The `-` ensures the user gets their full login environment.

- cd /var/www/ProjectX:

• Navigates to the project directory where bhatti now has ownership.

- touch config.yaml:

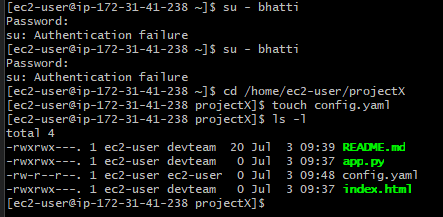
• Creates a new file named config.yaml.

- ls -l:

• Lists files to confirm the new file was created by 'bhatti' and belongs to the correct group.

Expected Output:

-rw-rw---- 1 bhatti devteam 0 Apr 26 10:10 config.yaml



✅ Step 1: Check System Resource Usage (CPU, Memory)

Command:

top

Explanation:

- Launches a real-time, interactive view of system processes and resource usage.

- Displays:

• CPU usage

• Memory usage

• Running processes

• Load average

• System uptime

Controls:

- Press `q` to exit the `top` command interface.

Use Case:

Helps identify if the system is under heavy load or if a process is using too many resources.

✅ Step 2: Check Running Processes Related to ProjectX

Command:

ps aux | grep ProjectX

Explanation:

- `ps aux`: Lists all running processes with detailed info.

- `| grep ProjectX`: Filters the output to show only lines containing “ProjectX”.

Use Case:

Helps locate any services or scripts related to the ProjectX directory or application that are currently running.

Note:

If the process isn't running or doesn’t include "ProjectX" in the name/path, the output might be e

