# ASSIGNMENT 02

COURSE

Linux Commands | Level: Beginner to Intermediate

SUBMITTED ON

01/05/2025

# Table of Contents

# Linux Commands Assignment: Web Development Team Management

## Objective

This document details the execution of Linux commands to manage files, users, permissions, and system resources for a web development team working on "ProjectX." The tasks simulate real-world system administration scenarios, performed on a Linux system (e.g., Ubuntu on an AWS EC2 instance). Each task includes commands, outputs, explanations, and screenshots.

## Task 1: Basic Linux Commands

**Scenario**

Developers need a workspace set up for ProjectX.

**Steps, Commands, Outputs, and Explanations**

1. **Create project directory and navigate into it**
   **Commands:**
   - mkdir /var/www/ProjectX
   - cd /var/www/ProjectX

**Output:**

```
[ec2-user@ip-172-31-21-125 var]$ sudo mkdir /var/www/ProjectX
[ec2-user@ip-172-31-21-125 var]$ cd /var/www/ProjectX
[ec2-user@ip-172-31-21-125 ProjectX]$
```

**Explanation:**
The mkdir /var/www/ProjectX command creates a directory for ProjectX in /var/www, a common location for web applications. The cd /var/www/ProjectX command navigates into the new directory.

2. **Create files for frontend and backend**
   **Command:**
   - touch index.html app.py README.md

**Explanation:**
The touch command creates three empty files: index.html (frontend), app.py (backend), and README.md (project documentation).

3. **Check current working directory**
   **Command:**
   - pwd

**Output:**

```
[ec2-user@ip-172-31-21-125 ProjectX]$ pwd
/var/www/ProjectX
[ec2-user@ip-172-31-21-125 ProjectX]$
```

**Explanation:**
The pwd command prints the current working directory, confirming the user is in /var/www/ProjectX.

4. **List files with detailed information**
   **Command:**
   - ls -l

**Output:**

```
[ec2-user@ip-172-31-21-125 ProjectX]$ ls -l
total 0
-rw-r--r--. 1 root root 0 May  1 09:56 README.md
-rw-r--r--. 1 root root 0 May  1 09:56 app.py
-rw-r--r--. 1 root root 0 May  1 09:56 index.html
[ec2-user@ip-172-31-21-125 ProjectX]$
```

**Explanation:**
The ls -l command lists files in long format, showing permissions (rw-r--r--), owner (root), group (root), size (0 bytes), and creation time.

5. **Display system disk usage**
   **Command:**

   - df -h

**Output:**

```
[ec2-user@ip-172-31-21-125 ProjectX]$ df -h
Filesystem        Size  Used Avail Use% Mounted on
devtmpfs          4.0M     0  4.0M   0% /dev
tmpfs             453M     0  453M   0% /dev/shm
tmpfs             181M  432K  181M   1% /run
/dev/nvme0n1p1    8.0G  1.6G  6.5G  20% /
tmpfs             453M     0  453M   0% /tmp
/dev/nvme0n1p128   10M  1.3M  8.7M  13% /boot/efi
tmpfs              91M     0   91M   0% /run/user/1000
[ec2-user@ip-172-31-21-125 ProjectX]$
```

**Explanation:**
The df -h command shows disk usage in human-readable format.

6. **View file content**
   **Commands:**

   - sudo bash -c 'echo "Welcome to ProjectX" > README.md'
   - cat README.md

**Output:**

```
[ec2-user@ip-172-31-21-125 ProjectX]$ sudo bash -c 'echo "Welcome to ProjectX" > README.md'
[ec2-user@ip-172-31-21-125 ProjectX]$ cat README.md
Welcome to ProjectX
[ec2-user@ip-172-31-21-125 ProjectX]$
```

**Explanation:**
The echo command writes "Welcome to ProjectX" to README.md, overwriting existing content. The cat command displays the file's content to verify.

# Task 2: User and Group Permission Management

**Scenario**

Create a user group for developers and assign permissions accordingly.

**Steps, Commands, Outputs, and Explanations**

1. **Create a developer group and users**
   **Commands:**
   - groupadd devteam
   - useradd bhatti
   - useradd malik
   - usermod -aG devteam bhatti
   - usermod -aG devteam malik

**Output:**

```
[ec2-user@ip-172-31-21-125 /]$ sudo groupadd devteam
[ec2-user@ip-172-31-21-125 /]$ sudo useradd bhatti
[ec2-user@ip-172-31-21-125 /]$ sudo useradd malik
[ec2-user@ip-172-31-21-125 /]$ usermod -aG devteam bhatti
usermod: Permission denied.
usermod: cannot lock /etc/passwd; try again later.
[ec2-user@ip-172-31-21-125 /]$ sudo usermod -aG devteam bhatti
[ec2-user@ip-172-31-21-125 /]$ sudo usermod -aG devteam malik
[ec2-user@ip-172-31-21-125 /]$
```

**Explanation:**
The groupadd devteam command creates a group named devteam. The useradd commands create users bhatti and malik. The usermod -aG commands add bhatti and malik to the devteam group (-a appends to avoid overwriting existing groups).

2. **Assign the group ownership of the project directory**
   **Command:**
   - chgrp -R devteam /var/www/ProjectX

**Explanation:**
The chgrp -R command recursively changes the group ownership of /var/www/ProjectX and its contents to devteam, enabling group-based access control.

3. **Set appropriate directory permissions**
   **Command:**

   - chmod -R 770 /var/www/ProjectX

**Explanation:**
The chmod -R 770 command recursively sets permissions to rwxrwx---, allowing the owner and group full access (read, write, execute) while denying access to others.

4. **Verify permissions**
   **Command:**

   - ls -ld /var/www/ProjectX

**Output:**

```
[ec2-user@ip-172-31-21-125 /]$ ls -ld /var/www/ProjectX
drwxrwx---. 2 root devteam 55 May  1 09:56 /var/www/ProjectX
[ec2-user@ip-172-31-21-125 /]$
```

**Explanation:**
The ls -ld command shows the directory's permissions (rwxrwx---), owner (root), and group (devteam), confirming the group ownership and permission changes.

5. **Check user group memberships**
   **Command:**

   - groups bhatti

**Output:**

```
[ec2-user@ip-172-31-21-125 /]$ groups bhatti
bhatti : bhatti devteam
[ec2-user@ip-172-31-21-125 /]$
```

**Explanation:**
The groups command lists the groups for user bhatti, confirming membership in bhatti (primary group) and devteam (supplementary group).

## Task 3: Change Ownership

**Scenario**

The lead developer (bhatti) should be the owner of the project files.

**Steps, Commands, Outputs, and Explanations**

1. **Change ownership of the directory to bhatti and group devteam**
   **Command:**
   - chown -R bhatti:devteam /var/www/ProjectX

**Explanation:**
The chown -R command recursively changes the owner to bhatti and the group to devteam for /var/www/ProjectX and its contents.

2. **Verify ownership changes**
   **Command:**
   - ls -l /var/www/ProjectX

**Output:**

```
[ec2-user@ip-172-31-21-125 /]$ sudo ls -l /var/www/ProjectX
total 4
-rwxrwx---. 1 bhatti devteam 20 May  1 10:02 README.md
-rwxrwx---. 1 bhatti devteam  0 May  1 09:56 app.py
-rwxrwx---. 1 bhatti devteam  0 May  1 09:56 index.html
[ec2-user@ip-172-31-21-125 /]$
```

**Explanation:**
The ls -l command confirms that all files in /var/www/ProjectX are now owned by bhatti and the devteam group, with appropriate permissions.

3. **Switch to user bhatti and create a new file**
   **Commands:**
   - su - bhatti
   - cd /var/www/ProjectX
   - touch config.yaml
   - ls -l

**Output:**

```
[ec2-user@ip-172-31-21-125 /]$ su - bhatti
Password:
Last login: Thu May  1 10:16:15 UTC 2025 on pts/0
[bhatti@ip-172-31-21-125 ~]$ cd /var/www/ProjectX
[bhatti@ip-172-31-21-125 ProjectX]$ touch config.yaml
[bhatti@ip-172-31-21-125 ProjectX]$ ls -l
total 4
-rwxrwx---. 1 bhatti devteam 20 May  1 10:02 README.md
-rwxrwx---. 1 bhatti devteam  0 May  1 09:56 app.py
-rw-r--r--. 1 bhatti bhatti   0 May  1 10:17 config.yaml
-rwxrwx---. 1 bhatti devteam  0 May  1 09:56 index.html
[bhatti@ip-172-31-21-125 ProjectX]$
```

**Explanation:**
The su - bhatti command switches to the bhatti user. Navigating to /var/www/ProjectX and using touch config.yaml creates a new file. The ls -l command verifies the new file is owned by bhatti and the devteam group.

## Task 4: System-Level Commands

**Scenario**

Monitor system performance and manage services for the web application.

**Steps, Commands, Outputs, and Explanations**

1. **Check system resource usage (CPU, memory)**
   **Command:**

   - top

**Output:**

```
top - 10:18:27 up  1:26,  1 user,  load average: 0.00, 0.02, 0.00
Tasks: 102 total,   1 running, 101 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    904.9 total,    516.9 free,    186.2 used,    201.8 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    588.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   5330 bhatti    20   0  224040   3368   2836 R   6.7   0.4   0:00.01 top
      1 root      20   0  172292  17172  10676 S   0.0   1.9   0:01.14 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     14 root      20   0       0      0      0 S   0.0   0.0   0:00.07 ksoftirqd/0
     15 root      20   0       0      0      0 I   0.0   0.0   0:00.10 rcu_preempt
     16 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration/0
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     20 root      rt   0       0      0      0 S   0.0   0.0   0:00.05 migration/1
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.07 ksoftirqd/1
     23 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     26 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
```

**Explanation:**
The top command provides a real-time view of system resource usage, including CPU, memory, and running processes.

2. **Check running processes for ProjectX**
   **Command:**

   - ps aux | grep ProjectX

**Output:**

```
[bhatti@ip-172-31-21-125 /]$ ps aux | grep ProjectX
bhatti      5392  0.0  0.2 222316  2176 pts/0    S+   10:19   0:00 grep --color=auto ProjectX
[bhatti@ip-172-31-21-125 /]$
```

**Explanation:**
The ps aux | grep ProjectX command lists processes related to ProjectX. The output shows a Python process running app.py, indicating the web application is active. The grep process is from the command itself.

3. **View system logs for troubleshooting**
   **Command:**

   - tail -n 50 /var/log/syslog

**Explanation:**
The tail -n 50 /var/log/syslog command displays the last 50 lines of the system log, useful for troubleshooting issues like service startup failures or errors related to ProjectX.

# Conclusion

This assignment demonstrated practical Linux administration tasks, including setting up a project workspace, managing user and group permissions, changing file ownership, and monitoring system performance. These skills are critical for managing web application environments in real-world scenarios.