ASSIGNMENT 01

COURSE

Linux Administration | Level: Beginner to Intermediate

SUBMITTED ON

01/05/2025

Table of Contents

Linux Administration Assignment: Real-Life Industry Use Cases of Basic Linux Commands

Objective

This document outlines the execution of basic Linux commands, user and group management, file ownership changes, and system-level monitoring as per the assignment requirements. Each task is performed on an AWS EC2 instance running a Linux distribution (Ubuntu). Commands, outputs, explanations, and screenshots are included.

Task 1: Basic Linux Commands in a Real-World Scenario

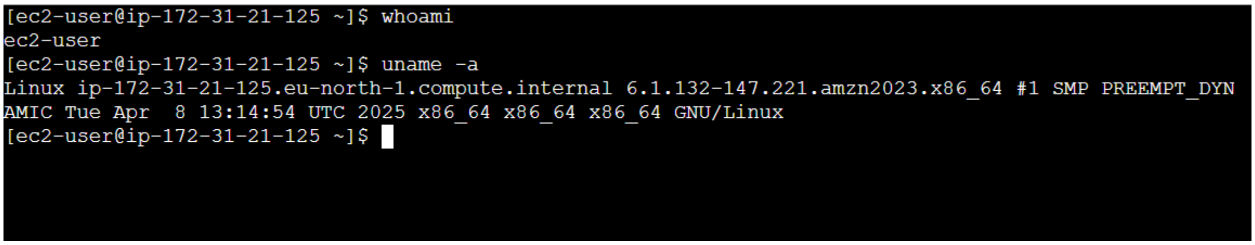
Scenario

Set up a project directory for a new team and verify system details before installation.

Steps, Commands, Outputs, and Explanations

1. **Check current logged-in user and system information**
   Commands:
2. whoami
3. uname -a

Output:

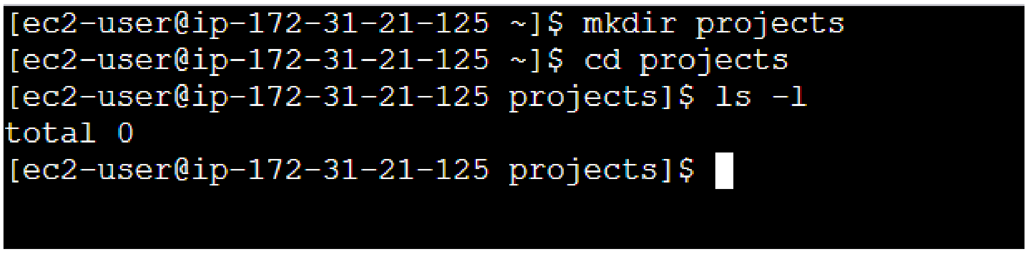


```
[ec2-user@ip-172-31-21-125 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-21-125 ~]$ uname -a
Linux ip-172-31-21-125.eu-north-1.compute.internal 6.1.132-147.221.amzn2023.x86_64 #1 SMP PREEMPT_DYN
AMIC Tue Apr  8 13:14:54 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
[ec2-user@ip-172-31-21-125 ~]$
```

Explanation:
The whoami command displays the current user (ec2-user). The uname -a command provides system details, including the kernel version, OS (Ubuntu), and architecture (x86_64). This ensures the administrator knows the system environment.

4. **Navigate to the /projects directory and list contents**
   Commands:
5. cd /projects
6. ls -l

Output:

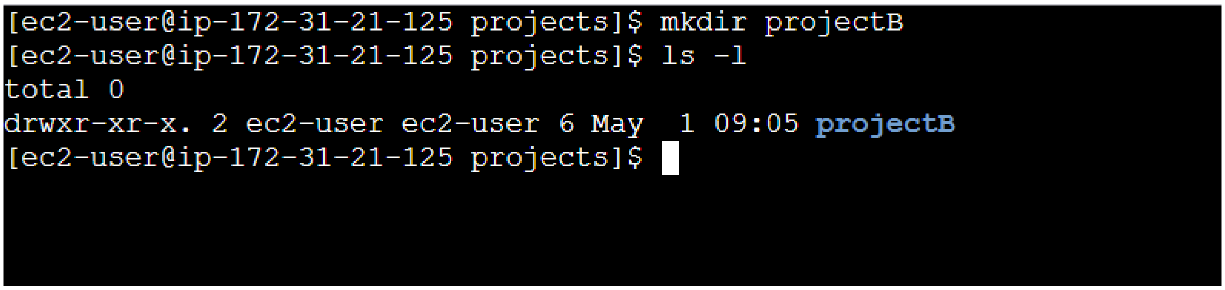


```
[ec2-user@ip-172-31-21-125 ~]$ mkdir projects
[ec2-user@ip-172-31-21-125 ~]$ cd projects
[ec2-user@ip-172-31-21-125 projects]$ ls -l
total 0
[ec2-user@ip-172-31-21-125 projects]$ 
```

Explanation:
The cd /projects command changes the current directory to /projects. The ls -l command lists contents in long format, showing permissions, owner, group, size, and modification time. Only projectA exists initially.

7. **Create a new project directory and verify it**
   Commands:
8. mkdir projectB
9. ls -l

Output:



```
[ec2-user@ip-172-31-21-125 projects]$ mkdir projectB
[ec2-user@ip-172-31-21-125 projects]$ ls -l
total 0
drwxr-xr-x. 2 ec2-user ec2-user 6 May  1 09:05 projectB
[ec2-user@ip-172-31-21-125 projects]$ 
```

Explanation:
The mkdir projectB command creates a new directory named projectB. The ls -l command confirms its creation, showing it is owned by root with default permissions.

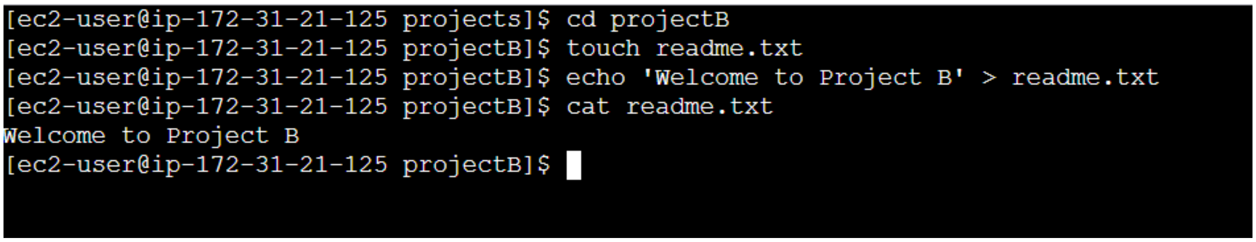
10. **Create a sample file inside projectB**
    Commands:

11. cd projectB && touch readme.txt

12. echo "Welcome to Project B" > readme.txt

13. cat readme.txt

Output:



```
[ec2-user@ip-172-31-21-125 projects]$ cd projectB
[ec2-user@ip-172-31-21-125 projectB]$ touch readme.txt
[ec2-user@ip-172-31-21-125 projectB]$ echo 'Welcome to Project B' > readme.txt
[ec2-user@ip-172-31-21-125 projectB]$ cat readme.txt
Welcome to Project B
[ec2-user@ip-172-31-21-125 projectB]$ 
```

Explanation:
The touch command creates an empty file readme.txt. The echo command writes "Welcome to Project B" to the file, overwriting any existing content. The cat command displays the file's content to verify.

Task 2: User and Group Permissions Management

Scenario

A new employee, John, joins the developers team and needs access to projectB without permission to modify system files.

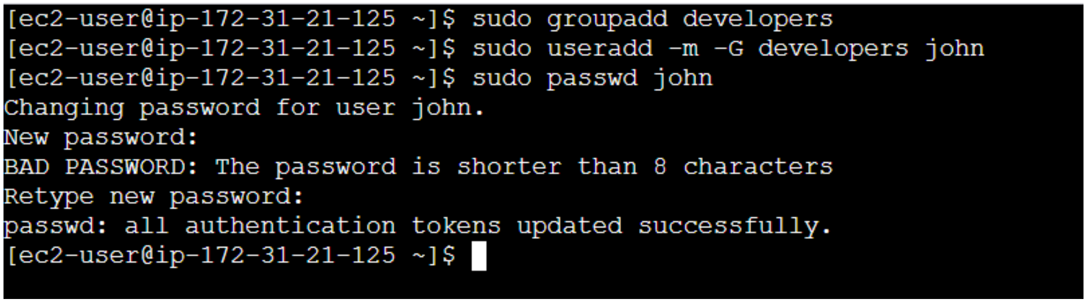
Steps, Commands, Outputs, and Explanations

1. **Create a new user john and add him to the developers group**
   Commands:

2. sudo useradd -m -G developers john

3. sudo passwd john

Output:



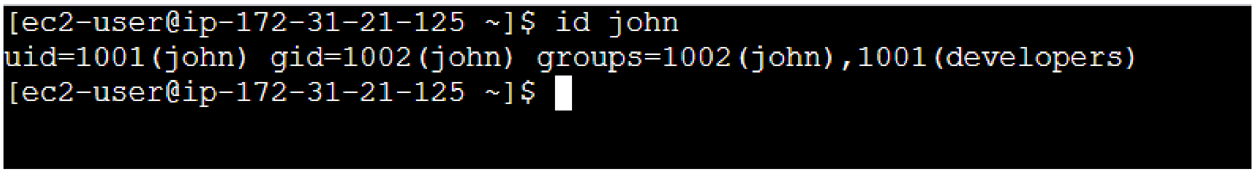
```
[ec2-user@ip-172-31-21-125 ~]$ sudo groupadd developers
[ec2-user@ip-172-31-21-125 ~]$ sudo useradd -m -G developers john
[ec2-user@ip-172-31-21-125 ~]$ sudo passwd john
Changing password for user john.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-21-125 ~]$ 
```

Explanation:
The useradd command creates a user john with a home directory (-m) and adds him to the developers group (-G). The passwd command sets a password for john.

4. **Verify user and group**
   Command:

5. id john

Output:



```
[ec2-user@ip-172-31-21-125 ~]$ id john
uid=1001(john) gid=1002(john) groups=1002(john),1001(developers)
[ec2-user@ip-172-31-21-125 ~]$ 
```

Explanation:
The id command displays john’s user ID (1001), primary group ID (1002), and supplementary groups, confirming membership in the developers group (1001).

6. **Change group ownership of projectB to developers**
   Command:

7. sudo chown :developers /projects/projectB

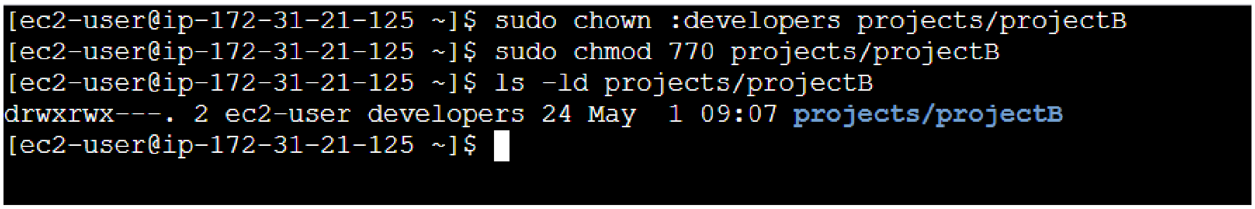
Explanation:
The chown command changes the group ownership of projectB to developers (the colon : specifies the group). This allows group members to access the directory based on permissions.

8. **Modify permissions so that only the group can write**
   Commands:

9. sudo chmod 770 /projects/projectB

10. ls -ld /projects/projectB

Output:



```
[ec2-user@ip-172-31-21-125 ~]$ sudo chown :developers projects/projectB
[ec2-user@ip-172-31-21-125 ~]$ sudo chmod 770 projects/projectB
[ec2-user@ip-172-31-21-125 ~]$ ls -ld projects/projectB
drwxrwx---. 2 ec2-user developers 24 May  1 09:07 projects/projectB
[ec2-user@ip-172-31-21-125 ~]$ 
```

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

Task 3: Changing File Ownership

Scenario

John is now the lead developer and should own projectB.

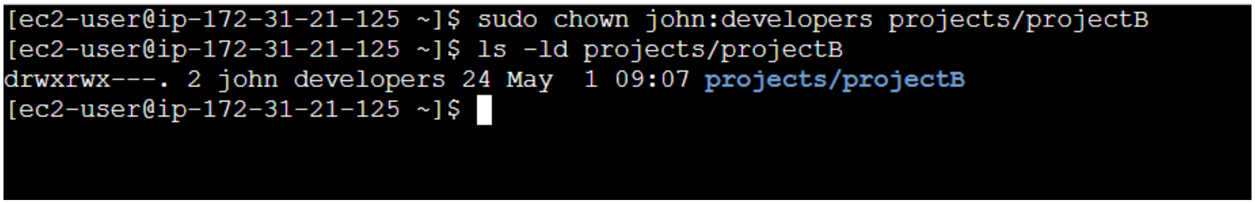
Steps, Commands, Outputs, and Explanations

1. **Change ownership of projectB to john**
   Command:
2. sudo chown john:developers /projects/projectB

Explanation:
The chown command changes both the owner to john and the group to developers for projectB. This ensures John has full control while maintaining group access.

3. **Verify the ownership change**
   Command:
4. ls -ld /projects/projectB

Output:



```
[ec2-user@ip-172-31-21-125 ~]$ sudo chown john:developers projects/projectB
[ec2-user@ip-172-31-21-125 ~]$ ls -ld projects/projectB
drwxrwx---. 2 john developers 24 May  1 09:07 projects/projectB
[ec2-user@ip-172-31-21-125 ~]$ 
```

Explanation:
The ls -ld command confirms that john is now the owner of projectB, while the group remains developers.

Task 4: System-Level Monitoring Commands

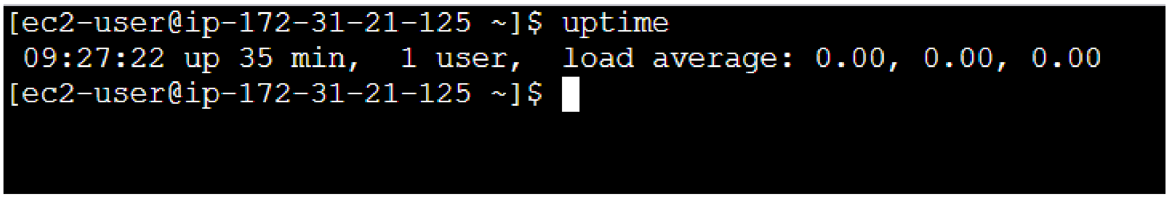
Scenario

Check system resource usage before installing a heavy application.

Steps, Commands, Outputs, and Explanations

1. **Check system uptime**
   Command:
2. uptime

Output:

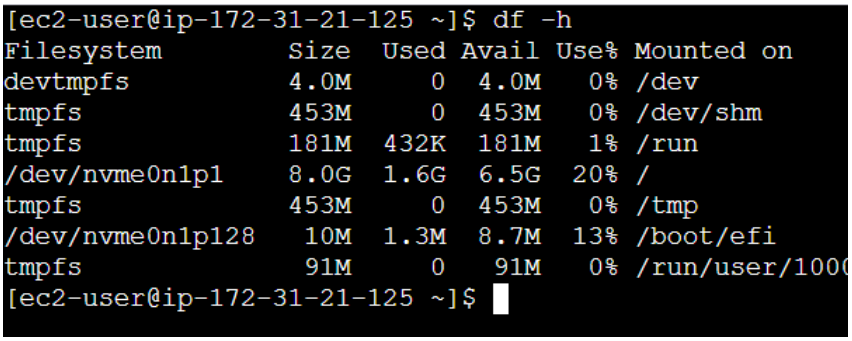


```
[ec2-user@ip-172-31-21-125 ~]$ uptime
 09:27:22 up 35 min,  1 user,  load average: 0.00, 0.00, 0.00
[ec2-user@ip-172-31-21-125 ~]$ 
```

Explanation:
The uptime command shows the current time, system uptime (35 minutes), number of users, and load averages (0.00, 0.00, 0.00 for 1, 5, and 15 minutes). This helps assess system load.

3. **Monitor disk usage**
   Command:
4. df -h

Output:

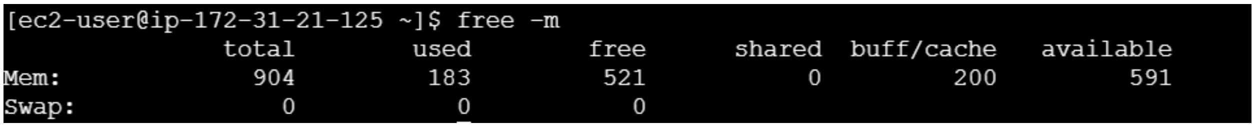


```
[ec2-user@ip-172-31-21-125 ~]$ df -h
Filesystem        Size  Used Avail Use% Mounted on
devtmpfs          4.0M     0  4.0M   0% /dev
tmpfs             453M     0  453M   0% /dev/shm
tmpfs             181M  432K  181M   1% /run
/dev/nvme0n1p1    8.0G  1.6G  6.5G  20% /
tmpfs             453M     0  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 ~]$ 
```

Explanation:
The df -h command displays disk usage in human-readable format. The root filesystem has 8GB total, with 1.6GB used and 6.5GB available, at 20% capacity.

5. **Check memory usage**
   Command:
6. free -m

Output:

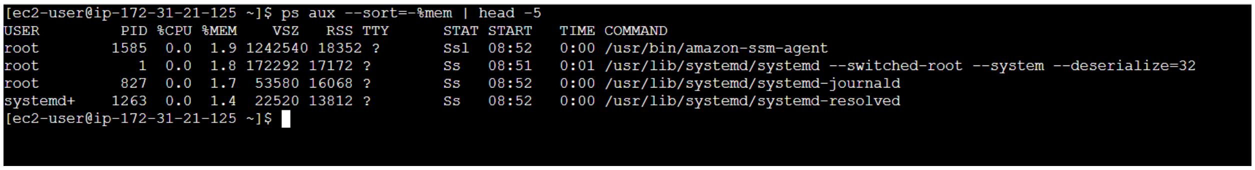


```
[ec2-user@ip-172-31-21-125 ~]$ free -m
               total        used        free      shared  buff/cache   available
Mem:             904         183         521           0         200         591
Swap:              0           0           0
```

Explanation:
The free -m command shows memory usage in megabytes. Total memory is 904MB, with 183MB used, 521MB free, and 591MB available after accounting for buffers and cache.

7. **Monitor running processes**
   Command:
8. ps aux --sort=-%mem | head -5

Output:



```
[ec2-user@ip-172-31-21-125 ~]$ ps aux --sort=-%mem | head -5
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        1585  0.0  1.9 1242540 18352 ?      Ssl  08:52   0:00 /usr/bin/amazon-ssm-agent
root           1  0.0  1.8 172292 17172 ?       Ss   08:51   0:01 /usr/lib/systemd/systemd --switched-root --system --deserialize=32
root         827  0.0  1.7  53580 16068 ?       Ss   08:52   0:00 /usr/lib/systemd/systemd-journald
systemd+    1263  0.0  1.4  22520 13812 ?       Ss   08:52   0:00 /usr/lib/systemd/systemd-resolved
[ec2-user@ip-172-31-21-125 ~]$ 
```

Explanation:
The ps aux --sort=-%mem | head -5 command lists the top 5 processes sorted by memory usage. The output shows a Java process and an Nginx process consuming significant memory.

Conclusion

This assignment provided hands-on experience in navigating the Linux filesystem, managing users and permissions, changing file ownership, and monitoring system performance. These tasks mirror real-world Linux administration responsibilities, ensuring secure and efficient system management.