### **Comprehensive Linux Operations**

#### **Project Overview**

This project spans various aspects of Linux system administration, including file management, user and group management, service control, process handling, and more. You will be completing tasks that simulate real-world scenarios, providing hands-on experience with Linux commands and configurations.

### **Project Breakdown**

#### **Part 1: Creating and Editing Text Files (20 minutes):**

**Scenario:** You are tasked with documenting the configurations and settings for a new server. You'll use different text editors to create and update these documents.

**1.Using Nano:**

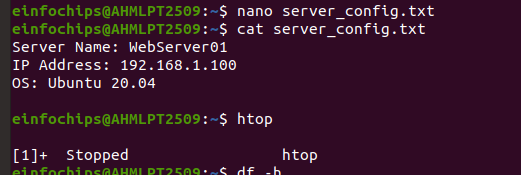
**Create and Open File with Nano**:

Create a file server\_config.txt using Nano:  
 nano server\_config.txt

**2. Add the following content:**  
  
Server Name: WebServer01

IP Address: 192.168.1.100

OS: Ubuntu 20.04



* Save and exit (Ctrl+O, Enter, Ctrl+X).

**Server Name**: WebServer01

- specifies the name of the server

* used to identify the server within a network or system.

**IP Address**: 192.168.1.100

* IP address of the server,
* IP address is used to identify and locate the server within a network.

**OS**: Ubuntu 20.04

* operating system running on the server,
* which is Ubuntu 20.04. This information is useful for knowing the environment and compatibility for software and applications.

**3. Using Vi:**

**1. Edit the same file with Vi:**  
  
vi server\_config.txt

**2. Append the following text:**  
  
Installed Packages: Apache, MySQL, PHP

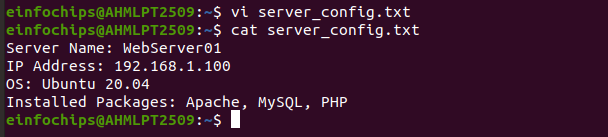
- Save and exit (Esc, :wq).

**Enter Insert Mode**:

* Press i to enter insert mode.
* see -- INSERT -- at the bottom of the terminal, indicating that you can now edit the file.

**3. Append the Specified Content**:

Installed Packages: Apache, MySQL, PHP



**4. Save and Exit:**

* Press **Esc** to exit insert mode.
* Type **:wq** and press Enter. This command writes the file (saves it) and quits Vi.

**4. Using Vim:**

**1.Further edit the file with Vim:**  
  
 vim server\_config.txt

**2.Add the following text:**

Configuration Complete: Yes

* Save and exit (Esc, :wq).

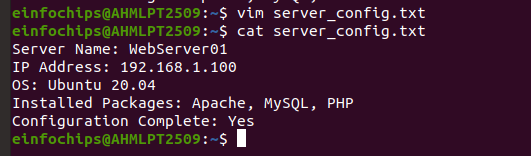
**3. Enter Insert Mode:**

**-** Press i to enter insert mode.

- see -- INSERT -- at the bottom of the terminal, indicating that you can now edit the file.

**4. Add the Specified Content**:

vim server\_config.txt



**5. save and Exit**:

* Press Esc to exit insert mode.
* Type :wq and press Enter. This command writes the file (saves it) and quits Vim.

#### **Part 2: User & Group Management (20 minutes):**

**Scenario:** You need to set up user accounts and groups for a new team joining the project.

1. **Adding/Removing Users:**

**Add a new user developer:**  
  
 sudo adduser developer

* **sudo**: This stands for "superuser do" used to execute commands with superuser (root) privileges. need root privileges to add a new user.
* **adduser**: add a new user to the system. It is more user-friendly than the useradd command because it creates a home directory and sets up the user's environment.
* **developer**: This is the username of the new user you are adding.

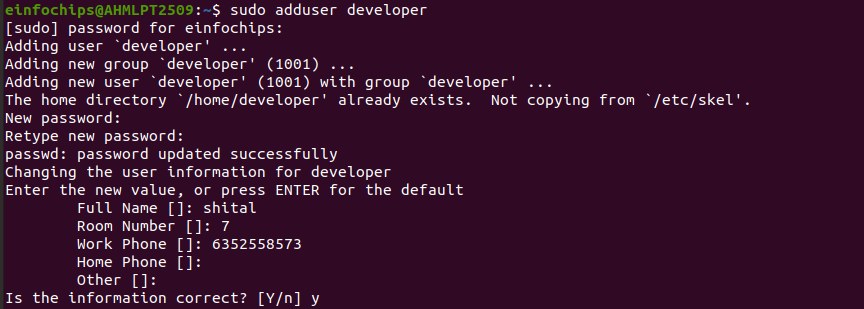
- After running the sudo adduser developer command,

**Enter Password**:

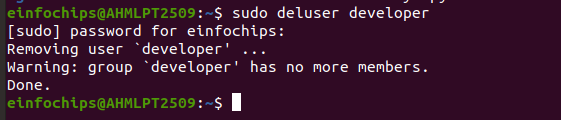
* enter your own password to confirm , necessary privileges to run the sudo command.

**Set User Password**:

* enter a password for the new user (developer), need to enter it twice to confirm.



**2. Remove the user developer:**  
  
sudo deluser developer



* **sudo**: stands for "superuser do" and is used to execute commands with superuser (root) privileges. You need root privileges to remove a user.
* **deluser**: This is a command to remove a user from the system. It removes the user's entry from the /etc/passwd file, effectively deleting the user account.
* **developer**: This is the username of the user you are removing.

**User Deletion**: The deluser command will remove the user account from the system.

**Home Directory**:

* By default, the deluser command does not remove the user's home directory.
* If you also want to remove the home directory, you need to add the --remove-home option:

sudo deluser --remove-home developer

**Group Membership**: The user will be removed from any groups they were a member of.

### **Example of Removing a User and Their Home Directory:**

* If you want to remove the user developer and also delete their home directory.

sudo deluser --remove-home developer

* Execute sudo deluser developer to remove the user.
* execute sudo deluser --remove-home developer to also remove the user's home directory.

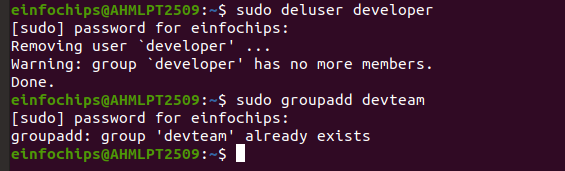
**2. Force Removal**:

- If the user is currently logged in or if there are running processes belonging to the user, you might need to forcefully remove the user:

sudo deluser --force --remove-home developer

**3. Managing Groups:**

**1. Create a group devteam:**  
  
sudo groupadd devteam



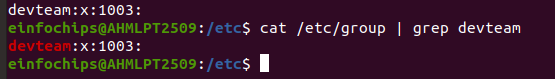
* **sudo**: This stands for "superuser do" and is used to execute commands with superuser (root) privileges. You need root privileges to add a new group.

**GID Assignment**:

* system assigns a unique group ID (GID) to the new group. This GID is used to identify the group within the system.

cd /etc/group

cat /etc/group | grep devteam

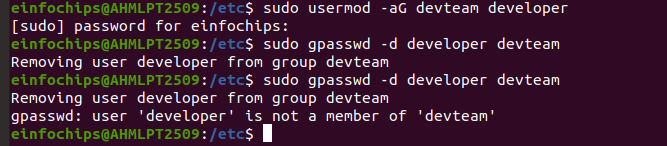


* x: placeholder for the group password (not commonly used).
* 1003: The unique GID assigned to the group.

**2. Add the user developer to the devteam group:**  
  
sudo usermod -aG devteam developer



**3. Remove the user developer from the devteam group:**  
  
sudo gpasswd -d developer devteam



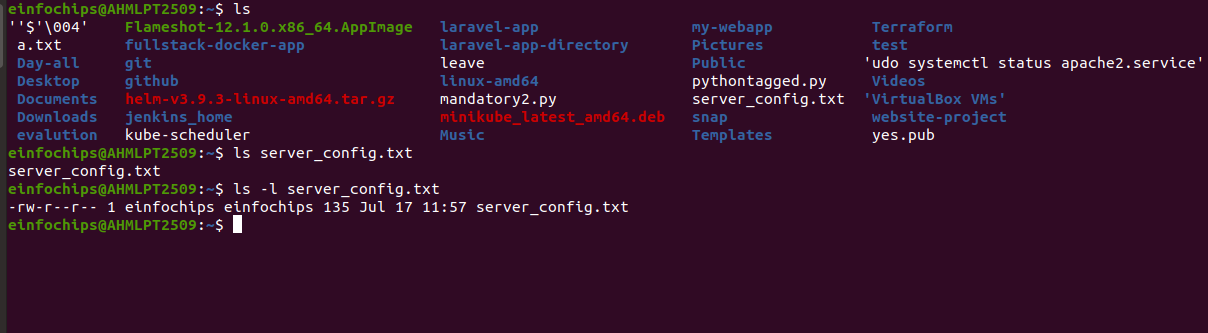
#### **# Part 3: File Permissions Management (20 minutes):**

**Scenario:** Ensure that only the appropriate users have access to specific files and directories.

**1. Understanding File Permissions:**

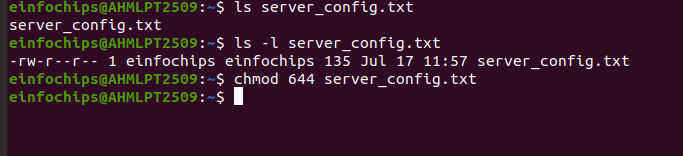
View permissions for server\_config.txt:  
  
ls -l server\_config.txt

* Discuss the output (e.g., -rw-r--r--).

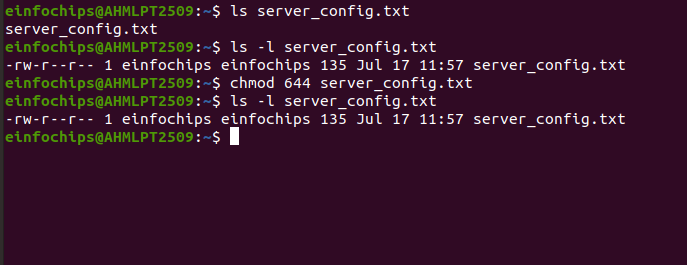


**2. Changing Permissions and Ownership:**

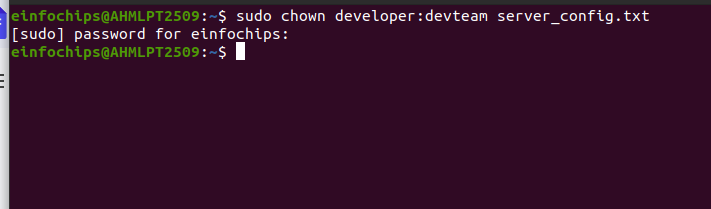
**1. Change permissions to read/write for the owner and read-only for others:**  
  
chmod 644 server\_config.txt



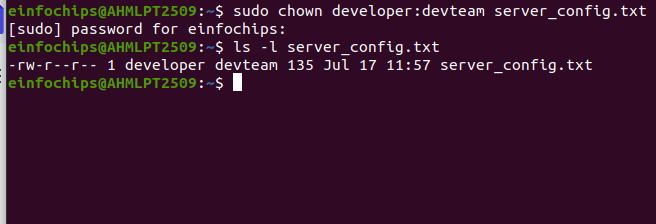
**2.Verify the change:**  
  
ls -l server\_config.txt



**3. Change the owner to developer and the group to devteam:**  
  
sudo chown developer:devteam server\_config.txt



**4. Verify the change:**  
  
ls -l server\_config.txt



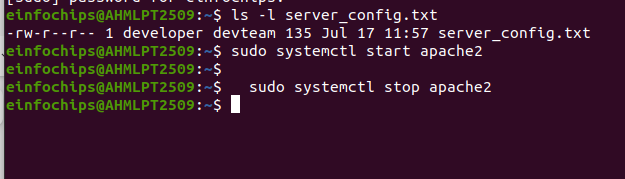
#### **# Part 4: Controlling Services and Daemons (20 minutes):**

**Scenario:** Manage the web server service to ensure it is running correctly and starts on boot.

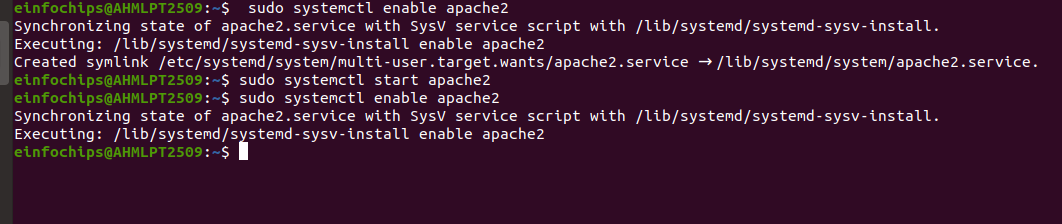
1. **Managing Services with systemctl:**

**1.Start the Apache service:**  
  
 sudo systemctl start apache2

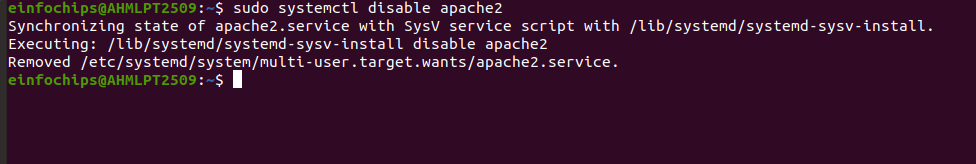
**2. Stop the Apache service:**  
  
 sudo systemctl stop apache2



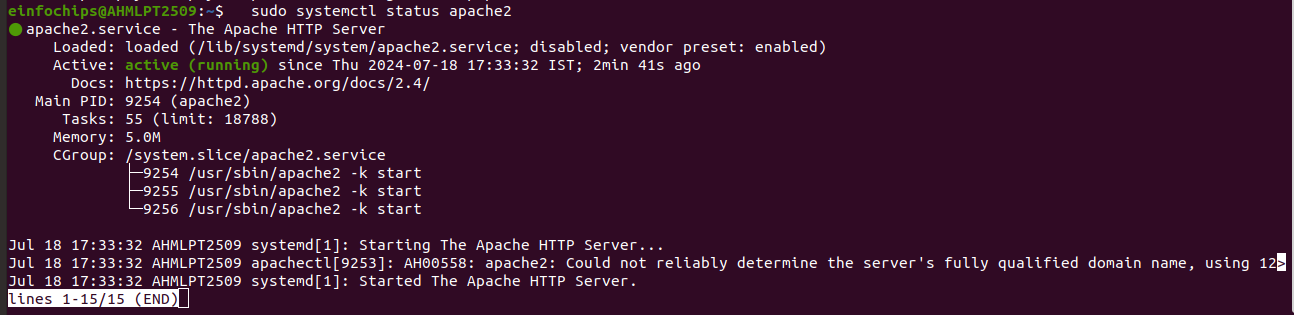
**3. Enable the Apache service to start on boot:**  
  
 sudo systemctl enable apache2



**4. Disable the Apache service:**  
  
 sudo systemctl disable apache2



**5. Check the status of the Apache service:**  
  
 sudo systemctl status apache2



**2. Understanding Daemons:**

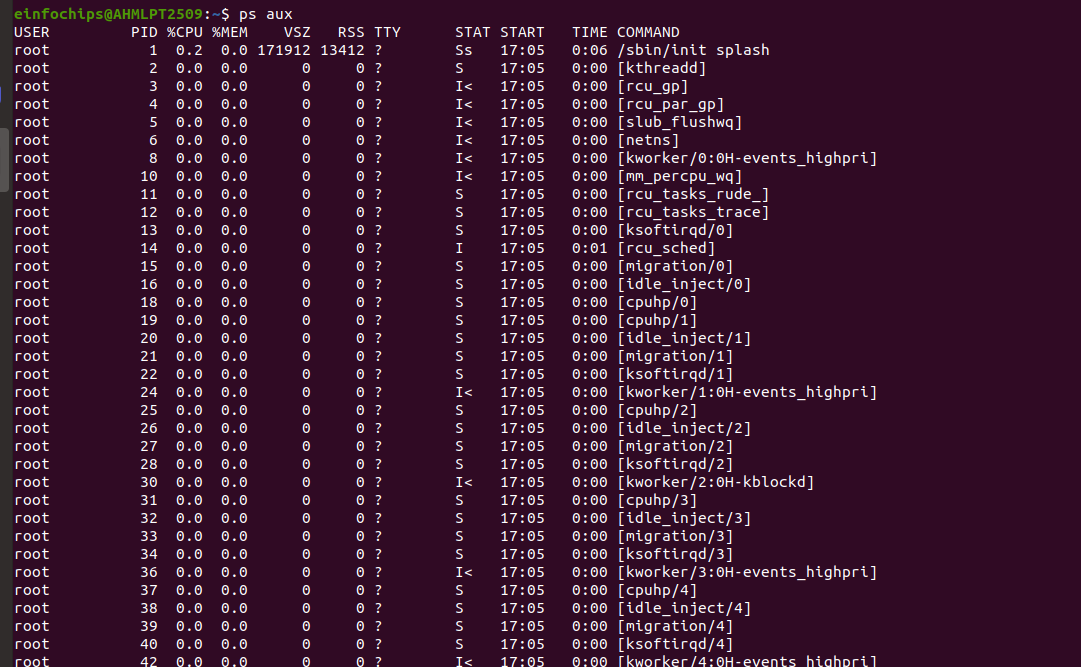
- Discuss the role of the sshd daemon in providing SSH access to the server.

#### **# Part 5: Process Handling (20 minutes):**

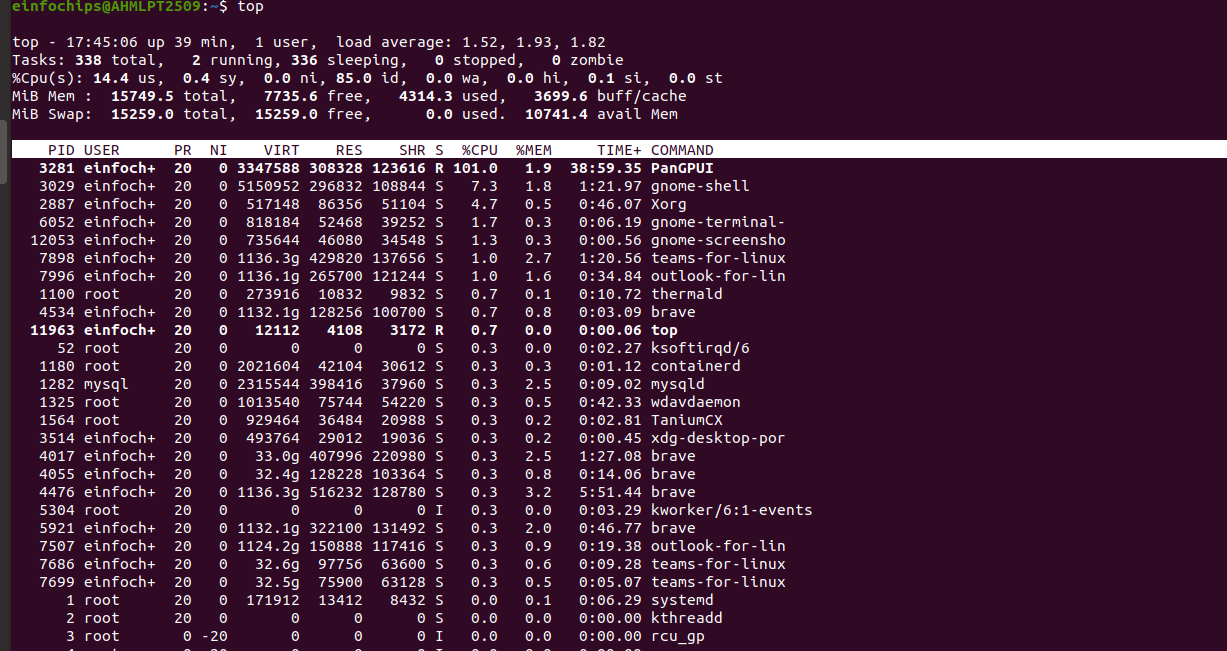
**Scenario:** Monitor and manage processes to ensure the server is performing optimally.

1. **Viewing Processes:**

**1. List all running processes:**  
  
 ps aux

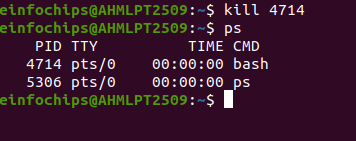


**2. Use top to view processes in real-time:**  
  
 top



1. **Managing Processes:**

**1. Identify a process to kill using ps or top, then kill it:**  
  
 kill <PID>



**3. Change the priority of a process (e.g., running sleep with a lower priority):**  
  
nice -n 10 sleep 100 &

**4.Change the priority of the process using renice:**  
  
renice +10 <PID>

