

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 a file `server_config.txt` using Nano:

```
nano server_config.txt
```

-

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).



```
einfochips@AHMLPT1108: ~/DevOPs_Training/Day-1
GNU nano 4.8 server_config.txt Modified
Server Name: WebServer01
IP Address: 192.168.1.100
OS: Ubuntu 20.04

```

2. Using Vi

Edit the same file with Vi:

```
vi server_config.txt
```

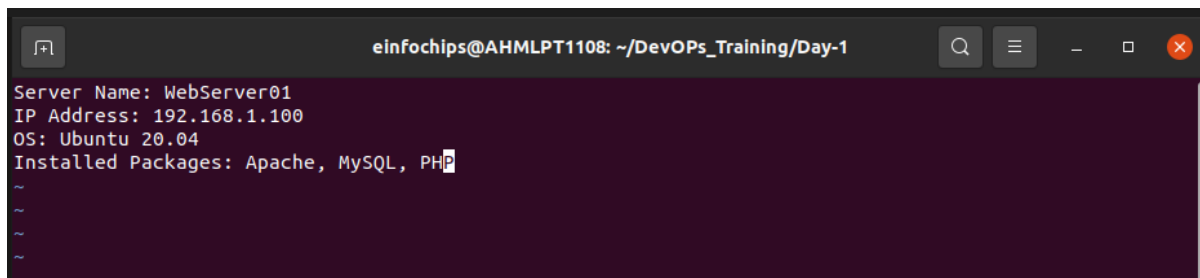
-

Append the following text:

Installed Packages: Apache, MySQL, PHP

-

- Save and exit (Esc, :wq).



```
einfochips@AHMLPT1108: ~/DevOPs_Training/Day-1
Server Name: WebServer01
IP Address: 192.168.1.100
OS: Ubuntu 20.04
Installed Packages: Apache, MySQL, PHP
~
~
~
~
```

3. Using Vim

Further edit the file with Vim:

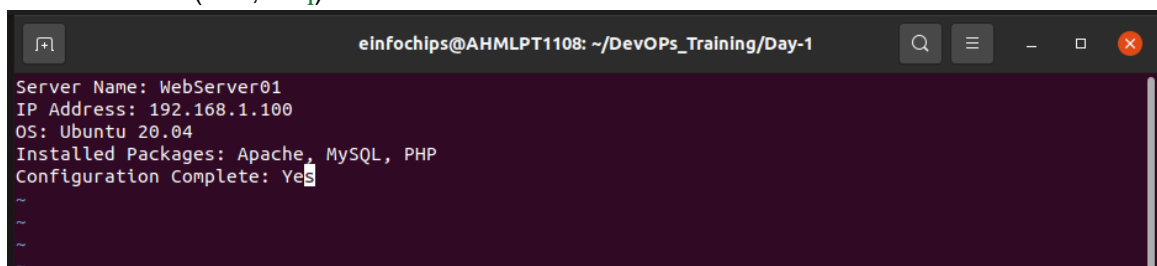
```
vim server_config.txt
```

-

Add the following text:

Configuration Complete: Yes

- Save and exit (Esc, :wq).



```
einfochips@AHMLPT1108: ~/DevOPs_Training/Day-1
Server Name: WebServer01
IP Address: 192.168.1.100
OS: Ubuntu 20.04
Installed Packages: Apache, MySQL, PHP
Configuration Complete: Yes
~
~
~
~
```

Part 2: User & Group Management (20 minutes)

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

4. Adding/Removing Users

Add a new user **developer**:

```
sudo adduser developer
```

Remove the user **developer**:

```
sudo deluser developer
```

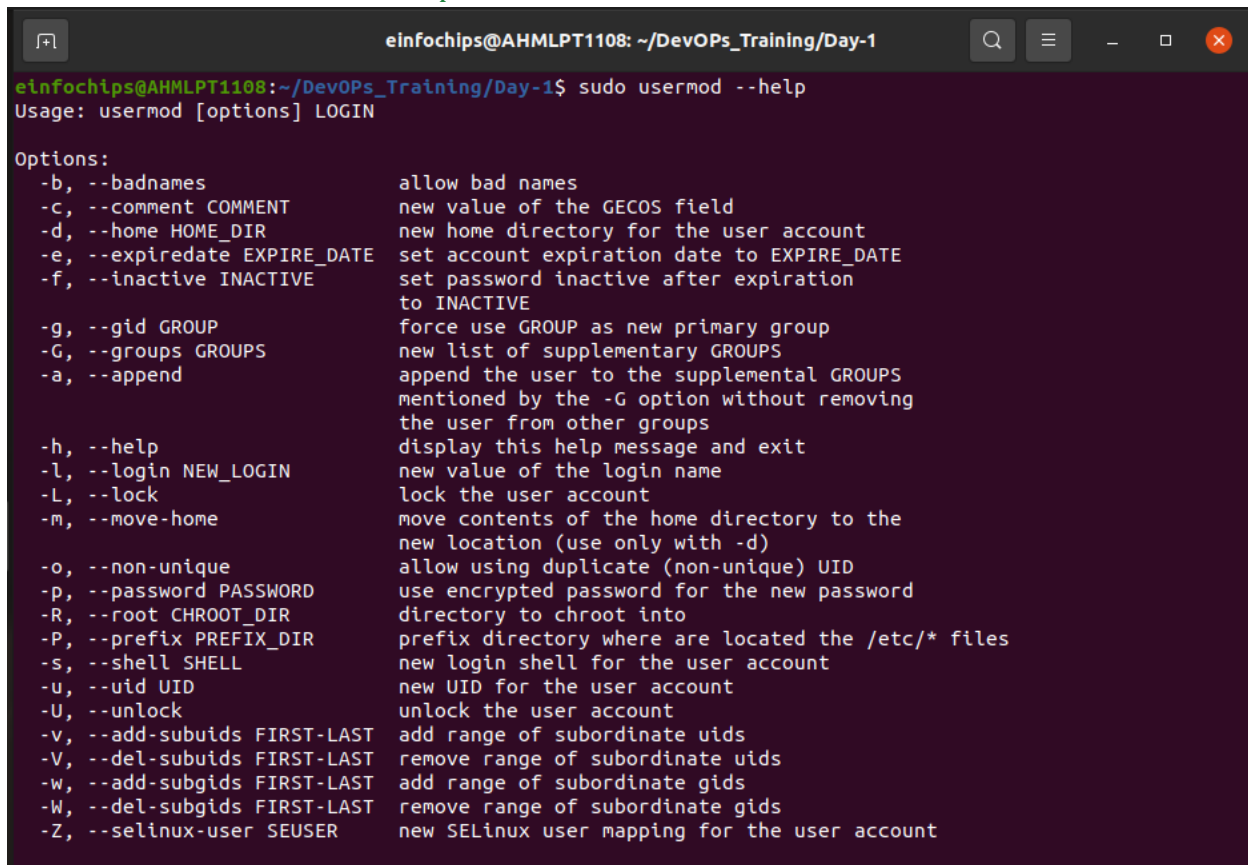
5. Managing Groups

Create a group **devteam**:

```
sudo groupadd devteam
```

Add the user **developer** to the **devteam** group:

```
sudo usermod -aG devteam developer
```

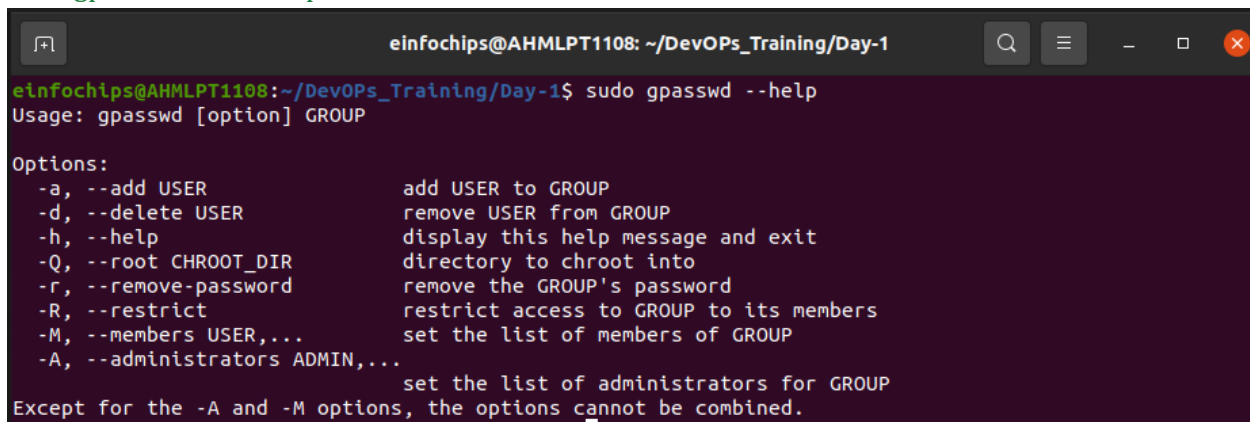


```
einfochips@AHMLPT1108: ~/DevOps_Training/Day-1
einfochips@AHMLPT1108:~/DevOps_Training/Day-1$ sudo usermod --help
Usage: usermod [options] LOGIN

Options:
-b, --badnames                allow bad names
-C, --comment COMMENT        new value of the GECOS field
-d, --home HOME_DIR          new home directory for the user account
-e, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE
-f, --inactive INACTIVE      set password inactive after expiration
                              to INACTIVE
-g, --gid GROUP               force use GROUP as new primary group
-G, --groups GROUPS           new list of supplementary GROUPS
-a, --append                  append the user to the supplemental GROUPS
                              mentioned by the -G option without removing
                              the user from other groups
-h, --help                    display this help message and exit
-l, --login NEW_LOGIN         new value of the login name
-L, --lock                    lock the user account
-m, --move-home               move contents of the home directory to the
                              new location (use only with -d)
-o, --non-unique              allow using duplicate (non-unique) UID
-p, --password PASSWORD       use encrypted password for the new password
-R, --root CHROOT_DIR         directory to chroot into
-P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
-s, --shell SHELL             new login shell for the user account
-u, --uid UID                 new UID for the user account
-U, --unlock                  unlock the user account
-V, --add-subuids FIRST-LAST add range of subordinate uids
-V, --del-subuids FIRST-LAST remove range of subordinate uids
-w, --add-subgids FIRST-LAST add range of subordinate gids
-W, --del-subgids FIRST-LAST remove range of subordinate gids
-Z, --selinux-user SEUSER     new SELinux user mapping for the user account
```

Remove the user **developer** from the **devteam** group:

`sudo gpasswd -d developer devteam`



```
einfochips@AHMLPT1108: ~/DevOps_Training/Day-1
einfochips@AHMLPT1108:~/DevOps_Training/Day-1$ sudo gpasswd --help
Usage: gpasswd [option] GROUP

Options:
  -a, --add USER          add USER to GROUP
  -d, --delete USER       remove USER from GROUP
  -h, --help              display this help message and exit
  -Q, --root CHROOT_DIR   directory to chroot into
  -r, --remove-password   remove the GROUP's password
  -R, --restrict          restrict access to GROUP to its members
  -M, --members USER,... set the list of members of GROUP
  -A, --administrators ADMIN,... set the list of administrators for GROUP

Except for the -A and -M options, the options cannot be combined.
```

Part 3: File Permissions Management (20 minutes)

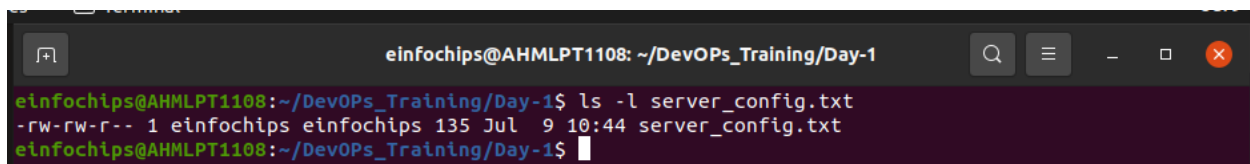
Scenario: Ensure that only the appropriate users have access to specific files and directories.

6. Understanding File Permissions

View permissions for `server_config.txt`:

`ls -l server_config.txt`

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



```
einfochips@AHMLPT1108: ~/DevOps_Training/Day-1
einfochips@AHMLPT1108:~/DevOps_Training/Day-1$ ls -l server_config.txt
-rw-rw-r-- 1 einfochips einfochips 135 Jul  9 10:44 server_config.txt
einfochips@AHMLPT1108:~/DevOps_Training/Day-1$
```

7. Changing Permissions and Ownership

Change permissions to read/write for the owner and read-only for others:

`chmod 644 server_config.txt`

Verify the change:

`ls -l server_config.txt`

-

Change the owner to **developer** and the group to **devteam**:

```
sudo chown developer:devteam server_config.txt
```

-

Verify the change:

```
ls -l server_config.txt
```

```
einfochips@AHMLPT1108: ~/DevOPs_Training/Day-1
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ chmod 644 server_config.txt
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ ls -l server_config.txt
-rw-r--r-- 1 einfochips einfochips 135 Jul  9 10:44 server_config.txt
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$
```

Part 4: Controlling Services and Daemons (20 minutes)

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

8. Managing Services with systemctl

Start the Apache service:

```
sudo systemctl start apache2
```

```
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ sudo systemctl start apache2
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-07-09 10:50:45 IST; 1s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 635789 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 635793 (apache2)
    Tasks: 6 (limit: 18919)
   Memory: 34.8M
    CGroup: /system.slice/apache2.service
            └─635793 /usr/sbin/apache2 -k start
              └─635794 /usr/sbin/apache2 -k start
                └─635795 /usr/sbin/apache2 -k start
                  └─635796 /usr/sbin/apache2 -k start
                    └─635797 /usr/sbin/apache2 -k start
                      └─635798 /usr/sbin/apache2 -k start

Jul 09 10:50:45 AHMLPT1108 systemd[1]: Starting The Apache HTTP Server...
Jul 09 10:50:45 AHMLPT1108 apachectl[635792]: AH00558: apache2: Could not reliably determine the ser>
Jul 09 10:50:45 AHMLPT1108 systemd[1]: Started The Apache HTTP Server.
lines 1-19/19 (END)
```

Stop the Apache service:

```
sudo systemctl stop apache2
```

-

Enable the Apache service to start on boot:

```
sudo systemctl enable apache2
```

-

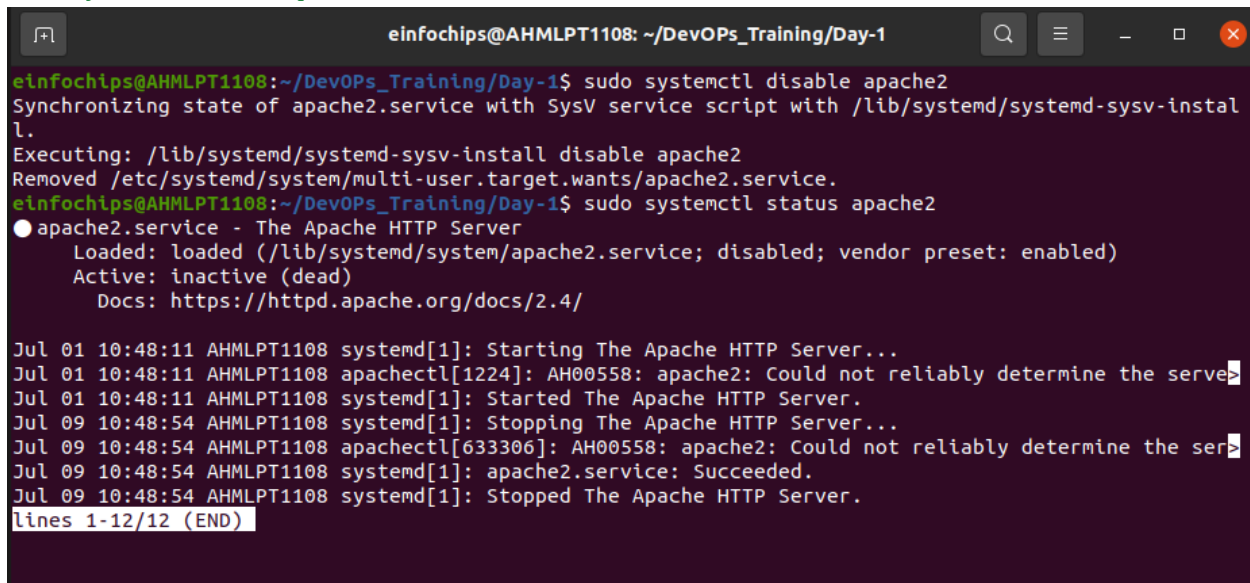
Disable the Apache service:

```
sudo systemctl disable apache2
```

-

Check the status of the Apache service:

```
sudo systemctl status apache2
```



```
einfochips@AHMLPT1108: ~/DevOPs_Training/Day-1
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ sudo systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: https://httpd.apache.org/docs/2.4/

Jul 01 10:48:11 AHMLPT1108 systemd[1]: Starting The Apache HTTP Server...
Jul 01 10:48:11 AHMLPT1108 apachectl[1224]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add the 'ServerName' directive to the configuration to avoid this warning.
Jul 01 10:48:11 AHMLPT1108 systemd[1]: Started The Apache HTTP Server.
Jul 09 10:48:54 AHMLPT1108 systemd[1]: Stopping The Apache HTTP Server...
Jul 09 10:48:54 AHMLPT1108 apachectl[633306]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add the 'ServerName' directive to the configuration to avoid this warning.
Jul 09 10:48:54 AHMLPT1108 systemd[1]: apache2.service: Succeeded.
Jul 09 10:48:54 AHMLPT1108 systemd[1]: Stopped The Apache HTTP Server.
lines 1-12/12 (END)
```

9. 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.

10. Viewing Processes

List all running processes:

```
ps aux
```

-

Use `top` to view processes in real-time:

```
top
```

-

11. Managing Processes

Identify a process to kill using `ps` or `top`, then kill it:

`kill <PID>`

-

Change the priority of a process (e.g., running `sleep` with a lower priority):

`nice -n 10 sleep 100 &`

-

Change the priority of the process using `renice`:

`renice +10 <PID>`

```
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ nice -n 10 sleep 100 &
[8] 636507
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$ renice +10 636507
636507 (process ID) old priority 10, new priority 10
einfochips@AHMLPT1108:~/DevOPs_Training/Day-1$
```

Creating and Deploying a Static Website with Apache2

Preparation (5 minutes)

- Ensure you have access to a Linux environment (e.g., virtual machines, EC2 instances, or local installations) with sudo privileges.

Activity Breakdown

Part 1: Installing Apache2 (5 minutes)

12. Update Package Lists

Open the terminal and run:

`sudo apt update`

-

13. Install Apache2

Install Apache2 by running:

`sudo apt install apache2`

-

14. Start and Enable Apache2

Start the Apache2 service:

```
sudo systemctl start apache2
```

-

Enable Apache2 to start on boot:

```
sudo systemctl enable apache2
```

-

15. Verify Installation

- Open a web browser and navigate to http://your_server_ip. You should see the Apache2 default page.

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
-- apache2.conf
    -- ports.conf
-- mods-enabled
    |-- *.load
    -- *.conf
-- conf-enabled
    -- *.conf
-- sites-enabled
    -- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to any file apart of those located in `/var/www`

Part 2: Creating the Website (10 minutes)

16. Navigate to the Web Directory

Change to the web root directory:

```
cd /var/www/html
```

```
einfochips@AHMLPT1108:/var/www/html$ pwd
/var/www/html
einfochips@AHMLPT1108:/var/www/html$
```

17. Create a New Directory for the Website

Create a directory named `mystaticwebsite`:

```
sudo mkdir mystaticwebsite
```

```
einfochips@AHMLPT1108:/var/www/html$ sudo mkdir mystaticwebsite
[sudo] password for einfochips:
einfochips@AHMLPT1108:/var/www/html$ ls
index.html  latest.tar.gz  mystaticwebsite  wordpress
einfochips@AHMLPT1108:/var/www/html$
```

Change ownership of the directory:

```
sudo chown -R $USER:$USER /var/www/html/mystaticwebsite
```

-

18. Create HTML File

Create and edit the `index.html` file:

```
nano /var/www/html/mystaticwebsite/index.html
```

```
einfochips@AHMLPT1108:/var/www/html$ sudo chown -R $USER:$USER /var/www/html/mystaticwebsite
einfochips@AHMLPT1108:/var/www/html$
```

Add the following content:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>My Static Website</title>
```

```
<link rel="stylesheet" type="text/css" href="styles.css">
```

```
</head>
```

```
<body>

<h1>Welcome to My Static Website</h1>

<p>This is a simple static website using Apache2.</p>

<script src="script.js"></script>

</body>

</html>
```

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



```
eic@DevOpsTraining: /var/www... x  einfochips@AHMLPT1108: /var... x  einfochips@AHMLPT1108: ~/D... x  ▼
GNU nano 4.8                                mystaticwebsite/index.html                                Modified
<!DOCTYPE html>

<html>

<head>

  <title>My Static Website</title>

  <link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

  <h1>Welcome to My Static Website</h1>

  <p>This is a simple static website using Apache2.</p>

  <script src="script.js"></script>

</body>

</html> |
```

19. Create CSS File

Create and edit the `styles.css` file:

```
nano /var/www/html/mystaticwebsite/styles.css
```

-

Add the following content:

```
body {  
  
    font-family: Arial, sans-serif;  
  
    background-color: #f0f0f0;  
  
    text-align: center;  
  
    margin: 0;  
  
    padding: 20px;  
  
}
```

```
h1 {  
  
    color: #333;  
  
}
```

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



The screenshot shows a terminal window with three tabs. The active tab is titled 'einfochips@AHMLPT1108: /var...' and shows the nano text editor editing a file named 'mystaticwebsite/styles.css'. The editor's status bar at the top indicates 'GNU nano 4.8'. The content of the file is the CSS code provided in the previous blocks, with the cursor positioned at the end of the last line. The terminal has a dark background with light-colored text.

```
einfochips@AHMLPT1108: ~/D... x einfochips@AHMLPT1108: /var... x einfochips@AHMLPT1108: ~/D... x  
GNU nano 4.8 mystaticwebsite/styles.css  
body {  
  
    font-family: Arial, sans-serif;  
  
    background-color: #f0f0f0;  
  
    text-align: center;  
  
    margin: 0;  
  
    padding: 20px;  
  
}  
  
h1 {  
  
    color: #333;  
  
} 
```

20. Create JavaScript File

Create and edit the `script.js` file:

```
nano /var/www/html/mystaticwebsite/script.js
```

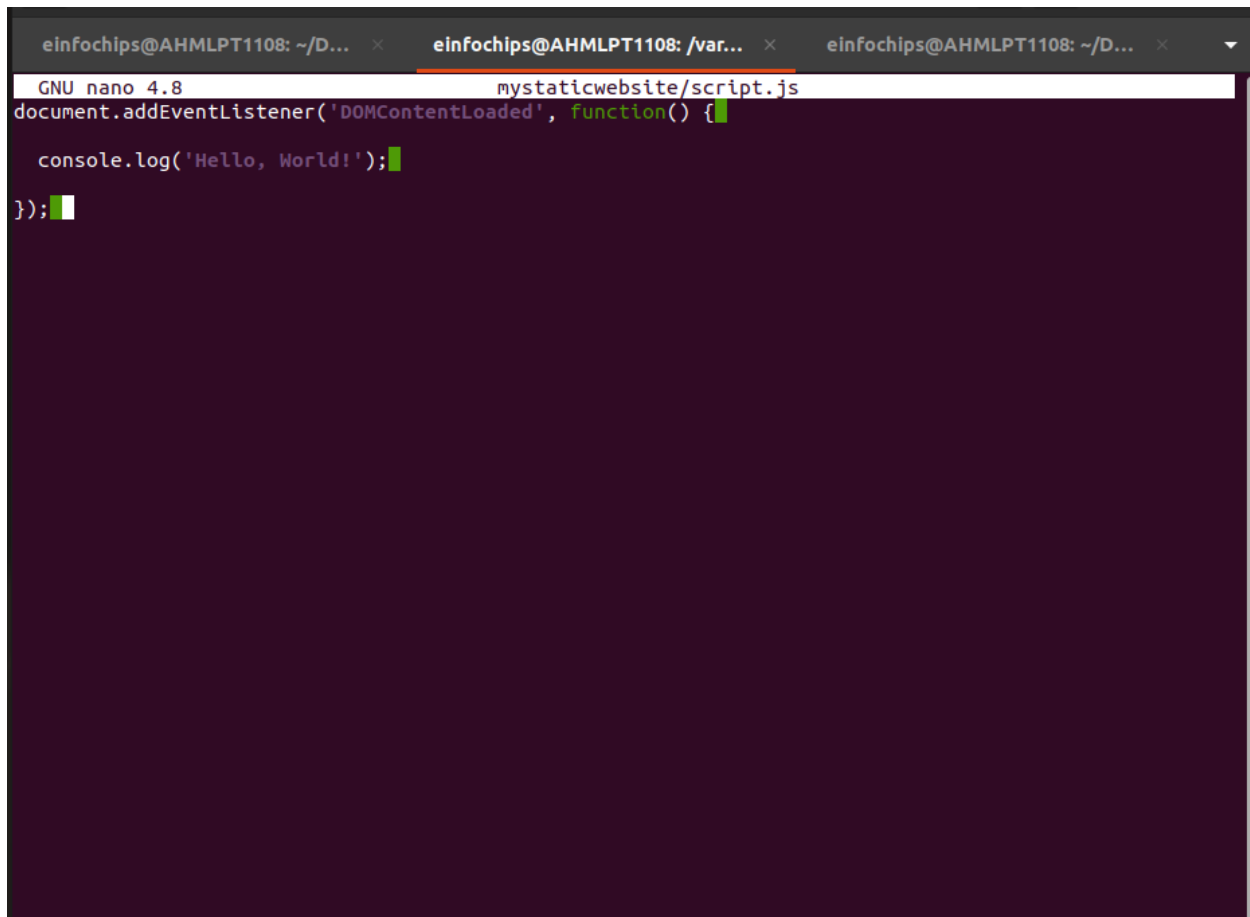
-

Add the following content:

```
document.addEventListener('DOMContentLoaded', function() {  
  
    console.log('Hello, World!');  
  
});
```

-

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



```
einfochips@AHMLPT1108: ~/D... x einfochips@AHMLPT1108: /var... x einfochips@AHMLPT1108: ~/D... x  
GNU nano 4.8 mystaticwebsite/script.js  
document.addEventListener('DOMContentLoaded', function() {  
  
    console.log('Hello, World!');  
  
});
```

ls

21. Add an Image

Download or copy an image file (e.g., `logo.png`) to the website directory:

```
cp /path/to/your/logo.png /var/www/html/mystaticwebsite/logo.png
```

-

Update `index.html` to include the image:

```
<body>
```

```
<h1>Welcome to My Static Website</h1>
```

```

```

```
<p>This is a simple static website using Apache2.</p>
```

```
<script src="script.js"></script>
```

```
</body>
```

```
GNU nano 4.8                                index.html
<!DOCTYPE html>

<html>
<head>
  <title>My Static Website</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1>Welcome to My Static Website</h1>
  
  <p>This is a simple static website using Apache2.</p>
  <script src="script.js"></script>
</body>
<html>
```

Part 3: Configuring Apache2 to Serve the Website (10 minutes)

22. Create a Virtual Host File

Create and edit the virtual host configuration file:

```
sudo nano /etc/apache2/sites-available/mystaticwebsite.conf
```

-

Add the following content:

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@localhost
```

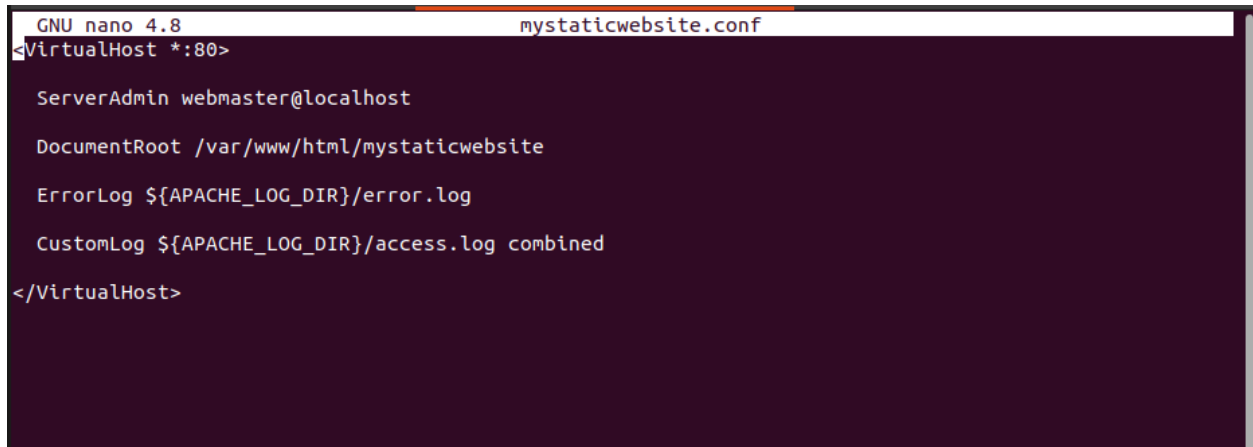
```
DocumentRoot /var/www/html/mystaticwebsite
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

CustomLog \${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

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



```
GNU nano 4.8                                mystaticwebsite.conf
<VirtualHost *:80>

    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html/mystaticwebsite

    ErrorLog ${APACHE_LOG_DIR}/error.log

    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```


23. Enable the New Virtual Host

Enable the virtual host configuration:

```
sudo a2ensite mystaticwebsite.conf
```

24. Disable the Default Site

Disable the default site configuration:

```
sudo a2dissite 000-default.conf
```

25. Reload Apache2

Reload the Apache2 service to apply the changes:

```
sudo systemctl reload apache2
```

```
einfochips@AHMLPT1108:/etc/apache2/sites-available$ sudo a2ensite mystaticwebsite.conf
Enabling site mystaticwebsite.
To activate the new configuration, you need to run:
  systemctl reload apache2
einfochips@AHMLPT1108:/etc/apache2/sites-available$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
einfochips@AHMLPT1108:/etc/apache2/sites-available$ sudo systemctl reload apache2
einfochips@AHMLPT1108:/etc/apache2/sites-available$
```

26. Test the Configuration

- Open a web browser and navigate to http://your_server_ip. You should see the static website with the HTML, CSS, JS, and image.

