Linux Basics – Detailed Beginner Guide (DevOps Training)

This document explains Linux fundamentals in a simple and practical way. The goal is to help understand how DevOps engineers work inside cloud servers using terminal commands. All explana are written for students who are new to Linux.

What is Linux?

Linux is an operating system just like Windows or macOS, but it is mostly controlled using instead of clicking icons. Most cloud servers such as AWS, Azure, and GCP use Linux because secure, and lightweight.

In DevOps, engineers use Linux to: - Manage servers - Deploy applications - Install tools - Automate tasks

Windows = Graphical interface (mouse + clicks) Linux = Command Line Interface (CLI)

Understanding the Linux Terminal

The terminal is where you type commands to control the system.

Example:

```
pwd
```

This shows your current location in the system.

Think of the terminal like giving instructions directly to the computer.

Navigation Commands (Moving Around in Linux)

pwd – Present Working Directory

Shows where you are currently located.

```
pwd
```

Example output:

```
/home/user
```

ls – List Files

Displays files and folders inside the current directory.

```
ls
```

More detailed view:

```
ls -l
```

cd – Change Directory

Used to move between folders.

```
cd foldername
cd ..
cd ~
```

- cd .. → move one level back
- cd ~ → go to home directory

Creating Files and Folders

mkdir – Make Directory

Creates a new folder.

```
mkdir project
```

touch – Create File

Creates an empty file.

```
touch index.html
```

These commands are used daily by developers and DevOps engineers.

Editing Files in Linux

nano – Simple Text Editor

```
nano index.html
```

Type your content inside nano editor.

Save the file: - CTRL + O → Enter - CTRL + X to exit

Nano is beginner-friendly and commonly used in cloud servers.

Viewing File Content

cat – Show File Content

```
cat index.html
```

Displays everything inside the file.

echo – Write Text into File

```
echo "Hello Linux" > file.txt
```

Creates a file with text content quickly.

File Permissions (Basic Understanding)

Linux controls who can access files using permissions.

Check permissions:

```
ls -l
```

Change permissions:

```
chmod 777 index.html
```

Explanation: - Read (r) - Write (w) - Execute (x)

`7 7 7` means everyone has full access.

System Information Commands (Basic)

Check system details:

```
uname -a
```

Clear terminal screen:

```
clear
```

View running processes:

```
ps
```

These commands help understand what is happening inside the system.

Why Linux is Important in DevOps

Most real-world deployments run on Linux servers. Learning these basic commands helps studer understand: - Cloud infrastructure - Server management - CI/CD workflows - Automation environments

You do not need advanced Linux knowledge at the beginner level. Focus on navigation, file ha understanding how commands work.

Practice Commands (Hands-On)

Try running these commands step-by-step:

```
mkdir linux_practice
cd linux_practice
touch test.txt
nano test.txt
cat test.txt
chmod 777 test.txt
ls -l
pwd
```

This practice covers the essential Linux basics used in DevOps training.

🎯 Student Activity – Git Practice Task

Follow this activity step-by-step. Do not copy all commands at once. Run each command and c
output.

Task Objective

Create a small project, track it using Git, and prepare it for pushing to GitHub.

Steps to Perform

```
mkdir git_activity
cd git_activity
touch about.html
nano about.html
```

Inside the file, add your name and department:

```
<h2>Student DevOps Practice</h2>
<p>Your Name</p>
```

Save and exit nano.

Initialize Git:

```
git init
```

Check file status:

```
git status
```

Add file to staging:

```
git add .
```

Create a commit:

```
git commit -m "student activity commit"
```

Expected Learning Outcome

- Understand how Git tracks new files
- Learn staging vs commit concept
- Build confidence using terminal commands

Quick Recap

Basic Linux + Git workflow used in this session:

Navigate → Create File → git init → git add → git commit

These fundamentals help beginners understand how DevOps engineers manage code and servers.