

Linux Commands

Basic Commands:

1. Listing the contents of a directory:

```
$ ls /directory_name
```

Example: \$ ls /etc

2. Changing directories:

```
$ cd /directory_name
```

Example: \$ cd /home

3. Displaying the current directory:

```
$ pwd
```

4. Creating a new directory:

```
$ mkdir new_directory
```

Example: \$ mkdir /home/user/new_folder

5. Copying files between directories:

```
$ cp /source/file /destination/
```

Example: \$ cp /etc/hostname /home/user/

6. Moving files between directories:

```
$ mv /source/file /destination/
```

Example: \$ mv /home/user/file.txt /home/user/new_folder/

7. Viewing file content:

```
$ cat /file_path
```

Example: `$ cat /etc/passwd`

8. Viewing system logs:

`$ tail -f /var/log/syslog`

9. Displaying disk space usage:

`$ df -h`

10. Finding files in the directory structure:

`$ find /directory_name -name 'filename'`

Example: `$ find /var -name '*.log'`

These commands will help you navigate and manage files within your AWS Ubuntu VM. The directory structure is essential to understand where system configurations, logs, and user data are stored.

Text Editors

1. Nano

- **Nano** is a simple, user-friendly text editor that is available on most Linux distributions by default.
- It is easy to use, making it ideal for beginners.

Key Features:

- Simple and intuitive interface.
- On-screen shortcuts for common commands.
- No modes (unlike Vim), so typing is straightforward.

Basic Commands:

Open a file in Nano: `nano filename.txt`

- **Save changes:** `Ctrl + O`
- **Exit:** `Ctrl + X`
- **Search:** `Ctrl + W`
- **Cut a line:** `Ctrl + K`
- **Paste:** `Ctrl + U`

To copy text in **Nano**, follow these steps:

1. Move the cursor to the beginning of the text you want to copy.
2. Press **Ctrl + ^** (this marks the beginning of the selection).
3. Use the arrow keys to highlight the text you want to copy.
4. Once the text is selected, press **Alt + 6** (this copies the selected text to the clipboard).

2. Vim

- **Vim** is a powerful, advanced text editor with a steep learning curve but great flexibility for more experienced users.
- It is based on **Vi**, an older editor, and is highly configurable with features such as syntax highlighting, macros, and plugins.

Key Features:

- Modal editing (Normal, Insert, and Visual modes).
- Extremely efficient once you get used to the commands.
- Ideal for advanced text manipulation and development tasks.

Basic Commands:

Open a file in vim: `vim filename.txt`

- Switch to **Insert mode** (to start typing): `i`
- **Save changes:**

First, press `Esc` to return to Normal mode.
Then type: `:w` (write)

- **Save and exit:** `:wq`
- **Exit without saving:** `:q!`
- **Search for text:** `/search_term`
- **Paste:** `p`

Feature	Nano	Vim
Interface	Simple, beginner-friendly	Advanced, modal editing
Learning Curve	Very easy	Steep, but highly efficient
Modes	No modes (simple editing)	Multiple modes (Normal, Insert)
Feature Set	Basic text editing features	Extensive, customizable, plugins
Undo	Multi-level	Multi-level undo
Syntax Highlighting	No	Yes
Customization	Limited	Highly customizable
Available by Default	Usually installed by default	May need to be installed
Use Case	Quick and simple edits	Advanced editing, scripting, dev

- **Nano** is best suited for users who need a simple text editor without much learning or complexity. It's perfect for quick, straightforward edits.
- **Vim** is for advanced users who want power, flexibility, and the ability to perform complex text manipulations quickly. It is widely used by developers and sysadmins.

Creating Files

1. touch Command

The touch command is used to create empty files.

Create a New File: If the specified file doesn't exist, touch will create an empty file with that name.

```
touch filename.txt
```

This will create an empty file named filename.txt.

Update File Timestamps: If the file already exists, touch updates the file's modification and access times to the current time without altering the file content.

```
touch existingfile.txt
```

Create Multiple Files: You can create multiple files at once using touch by specifying multiple filenames.

```
touch file1.txt file2.txt file3.txt
```

Common Options:

-c or --no-create: Prevents the creation of a file if it doesn't exist. Only modifies the timestamp if the file already exists.

```
touch -c file.txt
```

-a: Changes only the access time of a file.

```
touch -a file.txt
```

-m: Changes only the modification time of a file.

```
touch -m file.txt
```

-d: You can specify a specific date/time to set for the file's timestamp.

```
touch -d "2023-10-18 12:34" file.txt
```

2. echo Command

The echo command can be used to create a file with content.

```
echo "This is some text" > file.txt
```

This will create a file named file.txt and write the text "This is some text" into the file. If the file already exists, echo will overwrite its content.

- To **append** text instead of overwriting, use >>

```
echo "This is additional text" >> file.txt
```

3. cat Command

The cat command is usually used to display the contents of a file, but it can also be used to create a new file with content.

```
cat > file.txt
```

Type your content and then press Ctrl + D to save and exit.

- You can also append content to an existing file:

```
cat >> file.txt
```

4. printf Command

Similar to echo, but more flexible for formatting.

```
printf "Hello, World!\n" > file.txt
```

This creates a file named file.txt with the text "Hello, World!" and a newline.

5. vi or vim Command

You can use vi or vim to create and edit a file interactively.

```
vi filename.txt
```

Once you're in the editor, press i to enter **Insert mode**, type your content, and then save and exit with:

- Esc, then :wq

6. nano Command

The nano command is another interactive editor for creating files.

```
nano filename.txt
```

Type your content, then save with Ctrl + O, and exit with Ctrl + X.

7. > (Redirect Operator)

You can create an empty file using the redirect operator.

```
> filename.txt
```

This will create an empty file named filename.txt.

Command	Usage
<code>touch filename.txt</code>	Creates an empty file.
<code>echo "text" > filename.txt</code>	Creates a file with specific content.
<code>cat > filename.txt</code>	Creates a file with content from stdin.
<code>printf "text" > filename.txt</code>	Creates a file with formatted content.
<code>vi filename.txt</code>	Opens interactive editor for creation.
<code>nano filename.txt</code>	Opens interactive editor for creation.
<code>> filename.txt</code>	Creates an empty file using redirection.