

90DaysOfDevOps Challenge - Day 2 - Basic Linux Commands

Welcome to Day 2 of the #90DaysOfDevOps challenge! Today, I'll be diving into the world of basic Linux commands. These commands are essential for navigating and managing files and directories in a Linux environment. Let's explore some of the most commonly used commands and their examples.

Listing Commands

The `ls` command is used to list the subdirectories and files available in the present directory. Here's the basic syntax:

```
ls [option_flag] [arguments]
```

Let's look at some examples:

- `ls -l`: This command lists files and directories in a long list format, providing extra information such as permissions, owner, size, and modification date.
- `ls -a`: It lists all files and directories, including hidden ones that start with a dot (.).
- `ls *.sh`: This command lists all files with a `.sh` extension, helpful when you want to filter files based on a specific pattern.
- `ls -i`: It lists files and directories along with their respective index numbers (inodes).
- `ls -d */`: This command lists only directories, allowing you to view the directory structure without the files. You can also specify a pattern to narrow down the results.

Directory Commands

- `pwd`: The `pwd` command prints the current working directory, displaying the absolute path of the directory you're currently in.
- `cd path_to_directory`: This command changes the current directory to the provided path. For example, `cd /home/user/Documents` will navigate to the "Documents" directory.
- `cd ~` or just `cd`: Using `cd ~` or simply `cd` without any arguments takes you to the home directory.
- `cd -`: The `cd -` command allows you to go back to the last working directory you were in.
- `cd ..`: This command moves you one step back in the directory structure. It's useful for navigating up one level.
- `cd ../..`: You can use `cd ../..` to change directory by two levels, moving up twice in the directory hierarchy.

- **mkdir directoryName**: The **mkdir** command is used to create a new directory in a specific location. Here are some examples:

```
mkdir newFolder          # Creates a new folder named 'newFolder'

mkdir .NewFolder         # Creates a hidden directory (prefixing with
a dot)

mkdir A B C D            # Creates multiple directories at the same
time

mkdir /home/user/Mydirectory # Creates a new folder in a specific
location

mkdir -p A/B/C/D         # Creates a nested directory structure
```

Congratulations! You've learned some of the basic Linux commands for listing files and directories, navigating through directories, and creating new folders. These commands are the building blocks of your journey towards becoming a DevOps professional.

Stay tuned for Day 3 of the #90DaysOfDevOps challenge, where I'll explore more advanced Linux commands and their practical applications.