#90DaysOfDevOps Challenge - Day 12 - Linux & Git Wrap Up

Welcome to Day 12 of the #90DaysOfDevOps challenge. I'm happy to share my cheat-sheet summarizing all the essential commands and concepts I've learned during our Linux and Git/GitHub hands-on sessions.

📄 Linux & Git Cheat-Sheet 🚀

🐧 Linux Cheat-Sheet:

Navigation:

- `cd [directory]`: Change your current directory to the specified directory.
- `ls`: Display a list of files and directories in the current directory.
- `pwd`: Print the absolute path of the current working directory.
- `mkdir [directory]`: Create a new directory with the specified name.
- `rm [file]`: Delete the specified file.
- `rm -r [directory]`: Remove the specified directory and its contents recursively.

File Operations:

- `touch [file]`: Create a new file with the specified name.
- `cat [file]`: Display the contents of the specified file.
- `cp [source] [destination]`: Copy the file from the source location to the destination location.
- `mv [source] [destination]`: Move or rename the file from the source location to the destination location.
- `chmod [permissions] [file]`: Change the permissions of the specified file.

User Management:

- `sudo [command]`: Execute the specified command with superuser privileges.
- `useradd [username]`: Create a new user with the specified username.
- `passwd [username]`: Set a password for the specified user.
- `su [username]`: Switch to the specified user.
- `userdel [username]`: Delete the specified user.

Process Management:

- `ps`: Display a snapshot of the current running processes.
- `top`: Monitor the real-time system resources and running processes.
- `kill [pid]`: Terminate a process with the specified process ID.

Networking:

- `ping [host]`: Send ICMP Echo Request packets to the specified host to check network connectivity.
- `ifconfig`: Display information about the network interfaces on your system.
- `netstat`: Display network connections, routing tables, and network interface statistics.

🐙 Git-GitHub Cheat-Sheet:

Repository Management:

- `git init`: Initialize a new Git repository in the current directory.
- `git clone [repository]`: Clone the specified repository from GitHub to your local machine.
- `git add [file]`: Add the specified file to the staging area for the next commit.
- `git commit -m "[message]"`: Commit the staged changes with a descriptive message.
- `git push`: Push the committed changes to a remote repository.
- `git pull`: Fetch and merge the latest changes from a remote repository.

Branching and Merging:

- `git branch`: List all the branches in the repository.
- `git branch [branch]`: Create a new branch with the specified name.
- `git checkout [branch]`: Switch to the specified branch.
- `git merge [branch]`: Merge the changes from the specified branch into the current branch.

Collaboration:

- `git remote add [name] [url]`: Add a remote repository with the specified name and URL.
- `git fetch [remote]`: Fetch the latest changes from the specified remote repository.
- `git pull [remote] [branch]`: Pull the latest changes from the remote branch and merge them into the current branch.
- `git push [remote] [branch]`: Push the local commits to the remote branch.

Feel free to use this cheat-sheet as a quick reference during your DevOps journey. Let it serve as a reminder of the powerful commands and concepts we've learned.

As we wrap up the Linux and Git/GitHub section, it's time to gear up for Day 13, where we'll embark on a new journey into the world of Python. Get ready for new learnings, challenges, and discoveries. Stay tuned!