Cheatsheet On Git&Github

Day 12 : 90Days of DevOps Challenge

SS Shivraj Salunkhe · Apr 9, 2023 · 6 min read

Git commands :

1. `git clone`: Creates a copy of a remote repository on the local machine. This command is used to set up a local development environment from a remote repository.
2. `git branch`: Lists all the local branches in the repository. This command is used to manage multiple development branches in the same repository.
3. `git checkout`: Switches to a different branch or commit. This command is used to switch between different versions of the code.
4. `git merge`: Merges changes from one branch into another. This command is used to incorporate changes from a feature branch into the main development branch.
5. `git pull`: Pulls changes from the remote repository to the local repository. This command is used to keep the local repository up to date with the remote

repository.

6. `git push`: Pushes local commits to the remote repository. This command is used to share changes made locally with other team members.
7. `git rebase`: Reapplies changes from one branch onto another. This command is used to incorporate changes from a feature branch into the main development branch in a more streamlined way than merging.
8. `git tag`: Creates a lightweight reference to a specific commit. This command is used to mark specific points in the development history, such as a release or a stable build.
9. `git init`: Initializes a new Git repository in the current directory.
10. `git add <file>`: Adds a file to the staging area (also known as the index) for the next commit.
11. `git commit`: Commits the changes in the staging area to the Git repository.
12. `git status`: Shows the current state of the repository, including which files are staged or unstaged.
13. `git diff`: Shows the differences between the current state of the repository and the last commit.
14. `git log`: Shows the commit history for the repository.
15. `git remote`: Lists all the remote repositories that are connected to the local repository.
16. `git stash`: Stashes changes in the working directory that are not yet ready to be committed.

17. `git reset`: Resets the repository to a previous state.
18. `git fetch`: Fetches the latest changes from the remote repository without merging them into your local branch.
19. `git revert`: Reverts a commit by creating a new commit that undoes the changes made by the previous commit.
20. `git cherry-pick`: Applies changes from a specific commit to the current branch.

These **Git & GitHub** commands are essential for managing code changes and collaborating on software development projects in a DevOps environment.

File and directory management Commands :

1. `ls`: Lists the contents of the current directory.
2. `cd`: Changes the current working directory.
3. `pwd`: Prints the current working directory.
4. `mkdir`: Creates a new directory.
5. `rmdir`: Removes a directory.
6. `rm`: Removes a file.
7. `cp`: Copies a file.
8. `mv`: Moves or renames a file.
9. `touch`: Creates a new file.
10. `cat`: Displays the contents of a file.

11. `head`: Displays the first few lines of a file.
12. `tail`: Displays the last few lines of a file.
13. `grep`: Searches for a specific pattern in a file.
14. `chmod`: Changes the permissions of a file or directory.
15. `chown`: Changes the owner of a file or directory.
16. `ln`: Creates a symbolic link or hard link to a file or directory.
17. `du`: Shows the disk usage of a file or directory.
18. `df`: Shows the amount of free disk space on a file system.
19. `find`: Searches for files and directories in a directory hierarchy.
20. `tar`: Archives and compresses files and directories.

System information and management Commands :

1. `top`: Displays the current system processes and their resource usage. This command is used to monitor the system and identify processes that may be using too many resources.
2. `ps`: Displays information about running processes. This command is used to identify specific processes and their status.
3. `kill`: Terminates a process by sending a signal to it. This command is used to stop a process that is causing issues.

4. `ping`: Sends a packet to a network host to test connectivity. This command is used to check network connectivity and identify network issues.
5. `traceroute`: Shows the route that packets take to reach a network host. This command is used to identify network latency or routing issues.
6. `ifconfig`: Displays information about network interfaces. This command is used to configure and troubleshoot network interfaces.
7. `netstat`: Shows network statistics and active network connections. This command is used to monitor network activity and identify network issues.
8. `df`: Shows the amount of free disk space on a file system. This command is used to check disk usage and identify disk space issues.
9. `du`: Shows the disk usage of a file or directory. This command is used to check file or directory size and identify disk usage issues.
10. `free`: Displays information about system memory usage. This command is used to check memory usage and identify memory-related issues.
11. `vmstat`: Shows virtual memory statistics. This command is used to monitor system performance and identify memory-related issues.
12. `uptime`: Shows how long the system has been running and the average system load over the last 1, 5, and 15 minutes. This command is used to monitor system performance.
13. `date`: Displays the current date and time. This command is used to set or check the system clock.
14. `whoami`: Shows the current user. This command is used to verify user permissions.

15. `useradd`: Adds a new user to the system. This command is used to create new user accounts.
16. `usermod`: Modifies user account information. This command is used to modify user accounts.
17. `passwd`: Changes the password for a user account. This command is used to update user account passwords.

Package management Commands :

1. `apt-get`: A package management utility for Debian and Ubuntu-based systems. This command is used to install, update, and remove software packages from the system.
2. `yum`: A package management utility for Red Hat and CentOS-based systems. This command is used to install, update, and remove software packages from the system.
3. `dnf`: A newer package management utility for Fedora and other Red Hat-based systems. It is similar to `yum` and is used to install, update, and remove software packages from the system.
4. `pacman`: A package management utility for Arch Linux-based systems. This command is used to install, update, and remove software packages from the system.
5. `dpkg`: A package management utility for Debian and Ubuntu-based systems. This command is used to install, update, and remove software packages from the

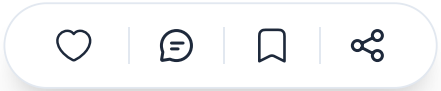
system.

6. `rpm`: A package management utility for Red Hat and CentOS-based systems. This command is used to install, update, and remove software packages from the system.
7. `zypper`: A package management utility for SUSE Linux-based systems. This command is used to install, update, and remove software packages from the system.
8. `apt-cache`: A command used to search for software packages on a Debian or Ubuntu-based system.
9. `yum search`: A command used to search for software packages on a Red Hat or CentOS-based system.
10. `pacman -Ss`: A command used to search for software packages on an Arch Linux-based system.
11. `apt-mark`: A command used to mark packages as manually installed or automatically installed on a Debian or Ubuntu-based system.
12. `yum groupinstall`: A command used to install a group of software packages on a Red Hat or CentOS-based system.
13. `pacman -Syu`: A command used to update all installed packages on an Arch Linux-based system.
14. `dpkg-reconfigure`: A command used to reconfigure an installed package on a Debian or Ubuntu-based system.

Thank You for Reading the Blog.. 👍

connect with me : linkedin.com/in/shivraj-salunkhe-5881141a4

follow my blog channel : **shivrajofficial.hashnode.dev**



Subscribe to my newsletter

Read articles from directly inside your inbox.
Subscribe to the newsletter, and don't miss out.

Enter your email address SUBSCRIBE

Devops cheatsheet GitHub Devops articles

WRITTEN BY

Shivraj Salunkhe

Follow

Dedicated and hardworking undergraduate student pursuing a degree in Information Technology with a passion for learning, leadership experience, and real-world skills

through internships and part-time jobs in IT sector. Actively looking for new Opportunities and committed to do personal and professional growth.

MORE ARTICLES

Shivraj Salunkhe

Python Libraries for DevOps

-> Reading JSON in Python To read JSON files in Python, you can use the json module. import json # ...

Shivraj Salunkhe

Python Data Types and Data Structures for DevOps

-> Data Types In DevOps, Python is a widely used programming language for scripting, automation, an...

Shivraj Salunkhe

Basics of Python Important for DevOps

What is Python ? Python is a high-level, interpreted programming language that is popular for its s...