*DevOps*

**DevOps**

(Tutorial – Channel Name: Technical Guftgu)

Youtube - <https://www.youtube.com/watch?v=JXiJNCtWmcA&list=PLBGx66SQNZ8aPsFDwb79JrS2KQBTIZo10&index=14>

**First Published On: 25th Jul 2021**

**Last Updated On: 25th Jul 2021**

**DevOps**

|  |
| --- |
| **Commands - Installation**  **sudo su**  **yum update –y**  **yum install git –y**  **git –version**  **git config global user.name “akshay”**  **git config global user.email** [**akshay@gmail.com**](mailto:akshay@gmail.com)  **Commands – Git operations**  *(goto working directory)*  **git init** *(only first time)*  **git status**  (*add the necessary files*)  **git add .**  **git commit –m “<<comment>>”**  **git log**  **git show <<commit id>>**  **git remote add origin <<https://github repo.git>>** *(only first time)*  **git push –u origin master**  **Advanced Commands**  If you want to use **multiple GitHub accounts** on a same computer,  Go to that particular working directory,  **git config --global credential.useHttpPath true**  **git remote add origin <<https://github repo.git>>** *(only first time)*  (you will be asked for web credentials again)  **git push –u origin master**  If above command does not work, use switch -f  **git push –f origin master**  Now you can go to respective folders and just do,  **git push –u origin master or with -f** |

|  |  |
| --- | --- |
| **Commands - Installation**  **sudo su**  **yum update –y**  **yum install git –y**  **git –version**  **git config global user.name “akshay”**  **git config global user.email** [**akshay@gmail.com**](mailto:akshay@gmail.com)  **Commands – Git operations**  *(goto working directory)*  **git init** *(only first time)*  **git status**  (*add the necessary files*)  **git add .**  **git commit –m “<<comment>>”**  **git log**  **git show <<commit id>>**  **git remote add origin <<https://github repo.git>>** *(only first time)*  **git push –u origin master**  **git pull origin master**  **Advanced Commands**  If you want to use **multiple GitHub accounts** on a same computer,  Go to that particular working directory,  **git config --global credential.useHttpPath true**  **git remote add origin <<https://github repo.git>>** *(only first time)*  (you will be asked for web credentials again)  **git push –u origin master**  If above command does not work, use switch -f  **git push –f origin master**  Now you can go to respective folders and just do,  **git push –u origin master**    **Git Branch**      **Git Merge**      **Git Stash**        **Git Reset**    **Git Revert** | * Stages of Git * When a repository is initialized configured, a .git folder is created in the repository, this folder is hidden * The .git folder contains logical separation into below stages   + Workspace/ Working directory – the area where we are currently typing the code   + Staging Area – the area where you add your code that is shortlisted   + Local Repository - when you commit (save) your final code * Every time when committed, git assigns a Commit ID which is 40 alphanumeric characters, as well as we can add tag as well * After commit, to store it into a central repository we push the code to Github * Commit is a process of taking a snapshot of the current code, this snapshot is incremental snapshot, i.e. only the new lines of code will be taken as a snapshot which is very beneficial as will not waste unnecessary storage space to store the complete code everytime * Github is central repository that provides storage space and is used for storing the code * Next time, when another developer of the same project wants to continue further, he will first pull the code so that all the code including the new updated code is added into his local repository. * This is how the contribution of code in team takes place * Branch is another repository * When first repository is created, it creates a Master Branch, and all the code will be saved to Master Branch * When some code is written, and developer wishes to add some new feature but not disturb current ongoing code, he can create another branch (xyz) so that another developer can start working on it to add a new feature. * Once the code in the branch (xyz) is finished and is working as expected, developers can merge all the code into master branch * You can take example of flipkart like app, developers can choose to create multiple branches, - for login page, for cart functionality, for payment, etc. and when all code is successfully working, they can then merge into master branch. * Git provides data integrity – data is consistent and secure in local and central repo * Uses SHA1 - generates a binary code and code is sent to the receiver along with this code, the receiver must validate this code, if it matches, it is secure, if anyone in the middle makes a slight change, the code gets changed, this means the code will not match at receiver end. * Git Installation Commands * Global username or email can be anything – need not necessarily be valid ones   **Next, Create an account in GitHub**    The logo is “Octa Cat”  GitHub is owned by Microsoft  They wanted octopus (to resemble branches) but it was already taken so they used a cat with 4 legs   * ***sudo su*** – switch user * ***which git*** – * ***mkdir mumbaigit*** – create a folder * ***git init*** – initialize git * ***cat > mumbai1*** – create a file and add some code   press Ctrl+d to come out of the file   * ***git status*** – to know the commit details * Observe that the file is in red, i.e. its not added in tracker for commit * ***git add .*** – to move the changes to staging area * observe that the the new file mumbai1 has changed to green because it was added to staging area * ***git commit –m “<<comment>>”*** – to commit the changes added to staging area to local storage * ***git log* –** get status of commit * ***git show <<commit id>>* -** details of commit id, like lines added or removed * ***git remote add origin <<git hub repo link.git>> -*** to connect local repo to remote central repo * ***git push –u origin master –*** to push the committed code to remote central repo. * ***git pull –u origin master -*** to pull the committed code from remote central repo to local repo. * If you want to ignore some fines of your repository, add the files in gitignore as shown * Add and commit the gitignore file to take effect * Observe that the .css and .java files are ignored and will not be tracked and pushed   Observe in Credentials manager, the credentials will be saved for respective repositories   * ***git branch <<branch name>>*** - To create a new branch * ***git branch*** – to know your current branch * ***git checkout <<branch name to switch to>>*** - to switch to another branch * **Note: the files created will be visible to all branches if it is not committed and is available in working directory.**   **Git Merge**   * ***git merge <<branch name>>* -** To merge the files from branch to master * To clone any repository, goto that repository and copy the URL as shown * Enter this command and a new repository in local repo will be created. |