**DevOps:**

**DevOps** is a methodology in the software development and IT industry. Used as a set of practices and tools, DevOps integrates and automates the work of [software development](https://en.wikipedia.org/wiki/Software_development) (*Dev*) and [IT operations](https://en.wikipedia.org/wiki/IT_operations) (*Ops*) as a means for improving and shortening the [systems development life cycle](https://en.wikipedia.org/wiki/Systems_development_life_cycle). DevOps is complementary to [agile software development](https://en.wikipedia.org/wiki/Agile_software_development); several DevOps aspects came from the *agile* way of working.

**Tools we are going to use:**

**Release: Jenkins**

**Build: Maven**

**Code management:**

**Github**

**Operate: Ansible, Kubernetes**

**Deploy: Docker**

**Monitoring: Prometheus, Grphana**

**Code management:**

* Github
* Git Lab
* Big bucket
* Visual studio management.

Most of the companies are using this tool for managing their code based on project requirement.

**Build Tools:**

* Maven
* Sbt

**Release management Tool:**

* Jenkins (most popular)
* Code ship

**Deployment Tool:**

* Docker
* AWS
* DC/OC

We can deploy our code in Docker and AWS and also in some other environments.

**Operation Tool:**

After deployment if we need to deploy the service inside the server or product we can we following tools.

* Ansible
* Kubernetes
* Chef

These are tools used to manage our configurations. Configuration like if we want to restart the service, if we want to update the service, configuration files in all those servers. Automation tools are taking care by those operation tools.

**GIT HUB**

**GIT:**

Git is version control system, a tool to manage your source code history installed and maintained in your local system. Git is the tool to validate and manage the code, we can collaborate the code with different team members to validate and work together. Work collaboratively entire team can work very easily without effecting any others work

**GIT HUB:**

Git hub is a hosting service for repositories, we can manage our source code and maintain in the history of source code in git hub.

**What is git hub?**

* Web-based git repository hosting service.
* Easy management of code.
* Open-source software for version control.
* Effective collaboration.
* Bug tracker.

**Installing and configurations of git and github:**

**Account creation:**

* Open github.com in browser
* Click on signup and proceed the steps
* Username : Unique, No Capital letters, No spaces
* Please remember Email-id, Username, Password
* After signup, please proceed with sign in
* Choose collaborative coding and click on continue
* Choose Free edition

**Create repository:**

* Click on + symbol (top right corner) -> click on new repository
* Repository name : Give any name
* Visibility : public
* Click on create repository
* Once you create repository, you will get a link in below format.
* Please copy your URL and paste in notepad and save for future purpose.

**How to get your token?**

github.com -> Settings -> Developer Settings (Left hand side - last option) ->

Personal access token -> tokens classic -> generate new token

-> generate new token classic

Note: Give any name (no capital, no space)

Expiration: No expiration

Scope: Select all (Select whatever you want - usually we will select all)

Generate token

You will get a big alphanumeric token. Please save carefully

**For Client Installation:**

git-scm.com/download/win

Copy paste above link in browser and download 64 bit (standalone)

Then install by using next, next... install, finish

How to know installation succeeded or not?

If you right click anywhere, you can able to see github (git Bash, git GUI) options

**Basic operations on git:**

* Create files
* Folder creation
* Edit the file
* Save the file
* View the file
* Add the file
* Check status
* Commit the file
* Push the file
* Pull the files and branches

>>git init (Once Git is installed, navigate to the directory you want to place under version control and create an empty Git repository).

>> git status (Review the resulting list of files)

>>git clone gitrepourl (cloning into the particular repository or folder)

>>git config --global user.email “[sst@email.com](mailto:sst@email.com)” (configure with mail)

>>git config –global user.name “git-user name” (configure with git-user name)

>> touch file-name (creating the new file)

>> vim file name (it will enter into code insert editor using “i” we enter the source code and using “ :wq!” we will save the source code)

>>cat filename (It will shows file data what will it containing )

>> git status ( status of the file )

>> git add filename (If all files in the list should be shared with everyone who has access to the repository, a single command will add everything in your current directory and its subdirectories)

>>git commit –m “message” (This creates a new commit with the given message. A commit is like a save or snapshot of your entire project. You can now push, or upload, it to a remote repository, and later you can jump back to it if necessary. If you omit the -m parameter, your default editor will open and you can edit and save the commit message there.)

>> git push –u <https://your-token@github.com/username/repo.git> main (files are pushing into the github using git-token and git-user name)

>> git pull origin main (if we done any changes in remote repository ,if we wants to access in local repositories we use pull command)

**Creating branches in Local Repositories:**

>>git clone gitrepourl (cloning into the particular repository or folder)

>>git pull origin branch-name ( we get the remote branches to local repositories )

>>git checkout –b branch name (it will create the new branch in main branch , and also it contains what are files in main branch it will get into new-branch)

>> git branch (it will shows the what are the branches are in main branch)

>>git status

>>git branch –a (it will shows all branches include remote repositories branches, what are in main branch)

>> git branch –r (it is also like above command only)

>>vim filename (do some changes in file)

>>git add filename (it will add to the git)

>>git commit –m “message”

>>git push –u <https://your-token@github.com/username/repo.git> branch-name

>> git checkout main (it will switch to main)

>>git switch branchname (instead on “checkout” we can use “switch” command)

**Creating branches in Remote Repositories:**

**Step1:** Go to github in that select one repository; in that select main branch in that main branch create one new branch.

**Step2:** In new branch we will see all files what main branch contains.

**Step3:** In that new branch, create new file in remote repository itself. (or) you can modify the old files also, once you done some changes means, preview that changes and give commit changes.

**Step4:** After all modification we have to add that files to main branch, using merge only.

**Step5:** Now rise the pull request, in top option we have Pull Request>> New pull Request>> Compare changes>> Create pull request>> Merge pull request >> Confirm merge.

**Step6:** After confirming the merge, that new file or modified file going to be merged in main branch.

>>cd repository name

>> git pull

>>git branch

>>git branch –r

>>git branch -a

>> git checkout branch(name of the branch)

>>git checkout main

>> git switch branch