**GIT/GITHUB :**

**➔ Introduction of GITHUB**

**➔ Introduction of GIT**

**➔ Installing and configurations of git and github**

**➔ Basic operations on git**

**➔ Git Commands**

**➔ Revision, Repository, Working copy**

**➔ Branches/tags**

**➔ Basic operations on git:**

* **Create files**
* **Edit the file**
* **Save the file**
* **View the file**
* **Add the file**
* **Check status**
* **Commit the file**
* **Push the file**

**➔ Git Commands:**

**>>Git init**

**>>git clone gitrepourl**

**>>**

**>>git status**

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

**>>git config –global user.name “github-username”**

**>> touch filename**

**>> 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**

**>> git status**

**>> git add filename**

**>>git status**

**>>git commit –m “message”**

**>> git push –u** [**https://your-token@github.com/username/repo.git**](https://your-token@github.com/username/repo.git) **main**

**8/2/23**

**Git operation:**

**pull**

**Creating branches**

**Create new file**

**Editing file data**

**Add file**

**Commit file**

**Push file**

**Git commands:**

**>> git pull origin main**

**>>git clone giturl**

**>>git checkout -b branchname**

**>>git branch**

**>>git status**

**>>ls**

**>>vim filename**

**>>git add filename**

**>> git commit –m “message”**

**>>git push –u** [**https://your-token@github.com/username/repo.git**](https://your-token@github.com/username/repo.git) **newbranchname**

**>>git checkout main**

**>> git push –u** [**https://your-token@github.com/username/repo.git**](https://your-token@github.com/username/repo.git) **main**

**----------------------------------------------------------------------------------------------**

**Git pull origin main**

**Git branch**

**Git checkout –b branchname**

**Git branch –a**

**Git branch –r**

**git push –u** [**https://your-token@github.com/username/repo.git**](https://your-token@github.com/username/repo.git) **branchname**

**git branch –merge**

**git switch main(branchname)**

**SSH KEY**

**Pubic key which we need to place in github for the authentication…**

**>>Ssh-keygen**

**>> vim id\_rsa.pub**

**Tomcat Installtion😘**

Download & Installation:-

* In Google type Apache Software download
* Click on First link
* Inside Core 5th option is installer
* Please click on that and download
* xxThen run that file
* If you don't have admin rights, it will prompt for password
* Click Next, Next... Finish
* How to check successfully installed or not?
* Open browser and type localhost:8080
* It will display tomcat home page.
* By default tomcat will install in the below location
* C:\Program Files (x86)\Apache Software Foundation\Tomcat 9.0
* If you don't have admin rights, please give admin rights to your directory
* by following below steps
* Right Click on Tomcat 9.0 Folder
* Properties
* Click on security tab
* Select users from top (scroll down)
* Click on Edit
* Select all options as allow
* click on apply, okay
* Inside Tomcat 9.0 Folder you can able to see below folders
* bin --- binaries (.sh, .bat...)
* lib --- libraries (jar..)
* conf --- Configurations (xml..)
* webapps --- To deploy our code
* go to webapps folder, and create a Folder with your project Name (Ex: SBI)
* Inside SBI directory you can create a file with the name of index.html
* and copy html code
* then you can access your webpage with the name of
* localhost:8080/SBI
* Note: Please make sure your file extion is correct. If you want to cross check
* click on view -> show filenames and extensions
* Custom pages:
* For example, If you create a file with the name of akhil.html
* and copy html code
* then you can access your webpage with the name of
* localhost:8080/SBI/akhil.html

**Devops flow**

**Continues Development**

* **Developers uses various tools to develop the code. Development code get push on to source code management**
* **Visual studio, eclipse**
* **Most of the cases devops engineers doesn’t write application code**
* **Devops engineers are responsible to maintain the using code management tools**
* **Git , bigbucket, mercurial, perforce, Sub version**

**Continues Integration – jenkins**

* **Multiple steps are involved in continuous intergration**
* **Unit testing – junit , nunit (plugin)**
* **Code build – maven, ANT, Gradle (plugins & setup)**
* **Code Analysis – Veracode, SonarQube(Plugins & setups)**
* **Code Artifacts – Nexus, jfrog Artifactory(Plugins & setups)**

**Continues Deployment**

* **Ansible**
* **Urban{code}**
* **Jensins**
* **Puppet**
* **AWS code deploy**

**Continuous deployment need target test/QA/production environment**

* **Docker**
* **Kubernetes**
* **Data center**
* **AWS**
* **Azure**

**Configaration management –Ansible , CHEF, saltstack, Puppet**

**Continues Testing**

**Selenuem , Apache J meter, Tricentis**

**Continues Monitoring**

**Nagios , Zabbix, Prometheus**

**Priority**

**Ansible**

**Kubernetes**

**Jenkins**

**Docker**

**AWS**