**Jenkins**

**Q 1​ What is Jenkins ?**

Answer : Jenkins is an open source automation server written in java with plugins built for continuous integration purpose. ​It​ helps to automate the non-human part of software development process, with ​continuous integration​ and facilitating technical aspects of continuous delivery.

**Q 2​ ​Why do we use Jenkins?** It is used for continuous integration & continuous delivery.It is the most widely accepted tools for C.I because of its flexibility and amount of plugins that is supports. It is supporting various development testing techniques ,Example Git, Maven, Selenium etc. The ​Jenkins software ​enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.

**Q 3​ ​What is Maven and what is Jenkins?** Answer # ​Maven is a build tool​. It helps in build and version control. However, ​Jenkins is continuous integration system​, where in maven is used for build. Jenkins can be used to automate the deployment process.

**Q 4​ ​What is meant by continuous integration in Jenkins?** # ​Continuous integration​ is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently and all development work is integrated as early as possible. Every commit made in the repository and then build.The resulting artifacts are automatically created and tested. This allows the teams to detect the problems early.

**Advantage of Continuous integration**—1) Since after every commit to the source code an auto build is triggered and then it is automatically deployed on the test server. 2) If the test results shows that there is a bug in the code then the developers only have to check the last commit made to the source code. 3) This also increases the frequency of new software releases. 4) The concerned team are always provided with relevant feedback.

**Q 5​ ​Why do we use Jenkins with selenium?** Answer # Running ​Selenium tests in Jenkins​ allows you to run your tests every time your software changes and deploy the software to a new environment when the tests pass. Jenkins can schedule your tests to run at specific time.

**Q 6 Jenkins CI CD Interview --​What are CI Tools?**

Answer # Here is the list of the top 8 ​Continuous Integration tools​:

● Jenkins ● TeamCity ● Travis CI ● Go CD ● Bamboo ● GitLab CI ● CircleCI ● Codeship

**Q 7-Jenkins Pipeline Interview-​What is a CI CD pipeline?**

Answer # A ​continuous integration​ and deployment pipeline (​CD/CI​) is such an important aspect of a software project. It saves a ton of manual, error-prone deployment work. It results in higher quality software for continuous integration, ​automated tests​, and code metrics.

**Q 8​ ​What is build ​pipeline​ in Jenkins? Answer** # Job chaining in ​Jenkins​ is the process of automatically starting other job(s) after the execution of a job. This approach lets you build ​multi-step build pipelines​ or trigger the rebuild of a project if one of its dependencies is updated.

**Jenkins**

**Q 9​ ​What is a Jenkins pipeline?** Answer # The ​Jenkins Pipeline plugin​ is a game changer for Jenkins users. Based on a ​Domain Specific Language (DSL) ​ in Groovy, the Pipeline plugin makes pipelines scriptable and it is an incredibly powerful way to develop complex, multi-step ​DevOps pipelines​.

**Q 10​ ​What is a DSL ​Jenkins​?** Answer # The Jenkins “Job DSL / Plugin” is made up of two parts: The Domain Specific Language (DSL) itself that allows users to describe jobs using a Groovy-based language, and a Jenkins plugin which manages the scripts and the updating of the Jenkins jobs which are created and maintained as a result.

**Q 11​ ​What is continuous integration and deployment?** Answer # ​Continuous Integration (CI)​ is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

**Q 12​ ​What is the tool used for provisioning and configuration?** Answer # Ansible is an agent-less configuration management as well as orchestration tool. In Ansible, the configuration modules are called “Playbooks”. Like other tools, Ansible can be used for cloud provisioning.

**Q 13​ ​What is the difference between Maven, Ant ​and​ Jenkins?**

Answer # Maven and ANT are build tool but main difference is that maven also provides dependency management, standard project layout and project management. On difference between Maven, ANT and Jenkins, later is a continuous integration tool which is much more than build tool.

**Q 14​ ​Which SCM tools Jenkins supports?**

Answer # Jenkins supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands.

**Devops Interview Questions Jenkins ​**

**Q 15​ ​How schedule a build in Jenkins?** Answer # In Jenkins, under the job configuration we can define various build triggers. Simple find the ‘Build Triggers’ section, and check the ‘ Build Periodically’ checkbox. With the periodically build you can schedule the build definition by the date or day of the week and the time to execute the build.

The format of the ‘Schedule’ textbox is as follows:

MINUTE (0-59), HOUR (0-23), DAY (1-31), MONTH (1-12), DAY OF THE WEEK (0-7)

​

**Q 16​ ​What is a ​Jenkinsfile​?** Answer # A Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control. Creating a Jenkinsfile, which is checked into source control, provides a number of immediate benefits:

1. Code review/iteration on the Pipeline 2. Audit trail for the Pipeline 3. Single source of truth for the Pipeline, which can be viewed and edited by multiple members of the project.

**Jenkins**

**Q 17​ ​How do you create Multibranch Pipeline in Jenkins?**

Answer # The Multibranch Pipeline project type enables you to implement different Jenkinsfiles for different branches of the same project. In a Multibranch Pipeline project, Jenkins automatically discovers, manages and executes Pipelines for branches which contain a Jenkinsfile in source control.

**Q 18​ ​What are Jobs in Jenkins?** Answer # ​Jenkins​ can be used to perform the typical build server work, such as doing continuous/official/nightly builds, run tests, or perform some repetitive batch tasks. This is called “​free-style software project​” in Jenkins.

**Q 18​ ​How do you create a Job in Jenkins?**

**In order to create a job in Jenkins I maintain the following steps-**

1. **I create a Freestyle project by giving the job name then**
2. **I select the Git under the source code management tab to locate the files which need to be build.**
3. **In build trigger section we use two options –**
4. **One is build periodically which is mainly for nightly basis and**
5. **Another one Poll SCM to check whether changes were made for new commits**
6. **I select Invoke top level maven under build environment section to set the goal and POM**
7. **Then we use Post build action option to generate reports by installing the plugin**
8. **At the end I select Email notification to get the report and save it to successfully to create a job.**

Go to ​Jenkins​ top page, select “New Job”, then choose “Build a free-style software project”. This job type consists of the following elements: optional ​SCM​, such as​ CVS​ or ​Subversion​ where your source code resides. optional triggers to control when Jenkins will perform builds.

some sort of build script that performs the build ​(ant, maven, shell script, batch file​, etc.) where the real work happens optional steps to collect information out of the build, such as archiving the artifacts and/or recording javadoc and test results.

optional steps to notify other people/systems with the build result, such as sending e-mails, IMs, updating issue tracker, etc.

**Q 19​ ​How do you configuring automatic builds in Jenkins?** Answer # ​Builds in Jenkins​ can be triggered periodically (on a schedule, specified in configuration), or when source changes in the project have been detected, or they can be automatically triggered by requesting the URL:

[**http://YOURHOST/jenkins/job/PROJECTNAME/build**](http://YOURHOST/jenkins/job/PROJECTNAME/build)

**Q 19​ ​How to create a backup and copy files in Jenkins?**

Answer # To create a backup, all you need to do is to periodically back up your ​JENKINS\_HOME​ directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory.

**Jenkins**

**1. Define the process of Jenkins?**

Answer:  Please start with the brief introduction about Jenkins as explained above in  introduction section. The process of Jenkins is as follows:

● In Source code management repository, developers need to commit their  code on daily basis or as per the team’s instructions because Jenkins check  the repository changes at regular intervals. SCM can be GIT, SVN, and TFS.  GIT and SVN are widely used nowadays.

● After committing the changes, Jenkins will detect the changes from a  repository, will pull the changes and process of new build will be started.

● It will iterate through the different stages that are mentioned in the Jenkins  pipeline. If one stage will get successfully completed then it will go to another  task or stage.

● If particular stage fails, Jenkins build also stop there and it will not proceed  further. Jenkins has a feature to send the email notification to a concerned  team or to particular email id, which has been configured for Jenkins.

● If all the stages will successfully be completed then it deploys the code in  particular mentioned server and testing will get started.

● If testing stage gets passed successfully, Jenkins will share the results with  the team. The whole process keeps on iterating whenever code will be  committed in SCM.

**2.Mention pre-requisites for using Jenkins?**

**Answer:  Pre-requisites are:**

● Access to source code management repository like GIT or SVN repository.

● Build script that should be working like mavens builds script.

**3. Explain the terms Agent, post-section, Jenkins file?**

Answer:Agent: – It is directive to tell Jenkins to execute the pipeline in particular  manner and order.

Post-section – If we have to add some notification and to perform other tasks at the end of a pipeline,It will definitely run at the end of  every pipeline’s execution.

Jenkins file: It is a text file having the information about Jenkins pipeline and is checked into source control.

**4. What is the use of pipelines in Jenkins?**

**Answer:** Pipeline plugin is used in Jenkins for making the Jenkins Pipeline, which  gives us the view of stages or tasks to perform one after the other in pipeline  form. It models the series of related tasks. Pipelines help the teams to review,  edit and iterate upon the tasks. Pipelines are durable and it can optionally  stop and wait for human approval as well to start the next task. A pipeline is  extensible and can perform work in parallel. It supports complex CD  requirements.

**Jenkins**

**5. Explain the advantages of using Jenkins?** Answer:  Advantages of using Jenkins are:

● Do not need to check the code and start to build manually.

● Whenever there are changes in source code, an automatic build will get  started.

● If there are any build failures, automatic email sends to concerned team.

● Bugs or defects can be easily tracked and resolved at early stages.

● Helps in achieving the continuous integration agile development and  test-driven development.

● It makes the process faster and smoother.

**6. Mention the steps to set up Jenkins Job**?  Answer:  The steps are:

● Go to Jenkins page, select ‘new job’ ● Enter the name of the job and choose ‘build free-style project’.  ● Click ok and start configuring your job.  ● Put the details of SVN or GIT repository, where the source code has been  placed.  ● Provide all other information regarding plugins that required using in a project  like for unit testing, code coverage, code quality.  ● Mention the build tool script like Maven script.  ● Collect the information regarding results  ● Configure the steps to send the notifications via emails, etc.

**7. How do you start the Jenkins manually?  Answer:  The steps are as follow: -**

● Open the command line prompt and go to Jenkins installation directory.

● ‘Jenkins.exe start’ command helps in starting the Jenkins.

● ‘Jenkins.exe restart’ to restart Jenkins and ‘Jenkins.exe stop’ to stop  Jenkins.

**8.Mention steps to copy/move Jenkins from one server to  other? ;**  The steps are as follows:

● Copy the whole Jenkins directory from the first server and paste it to other  servers.  ● Make a copy of existing job by cloning a job directory with a different name.  ● Rename an existing job by renaming the directory.

**9. How to create a backup for Jenkins?**:Take periodically back up of JENKINS\_HOME directory by just copy this  directory to other places. Rename, clone and replication of job can also be  done.

**10. How can we schedule build in Jenkins**? Answer:  Whenever a change is committed in a repository, scheduled at a specified  time, build manually and other builds get completed.

**11. Which plugins are useful in Jenkins?  Answer:  Some plugins are listed below:**

● Maven (build tool)  ● GIT (SCM)  ● Selenium (continuous testing)  ● Amazon EC2  ● Puppet (Configuration management)  ● Nagios (Continuous monitoring) ● Copy artifact  ● HTML publisher

**12. How do you secure Jenkins?** Answer --● Global security should be enabled.  ● Jenkins should be integrated with appropriate plugins.  ● Automate the process of setting rights and privileges.  ● Limit the physical access to folders.  ● Periodically run security audits. …………

**Jenkins**

**Q What is plugin? And the categories of Plugin—**

A plugin is a software add-on that is installed on a program, enhancing it’s capabilities. There are certain plugin for 1) Testing -Junit, Selenium etc. 2) Reporting – HTML publisher 3) Notification- Jenkins Build Notification plugin. 4) Development -Deploy plugin.5) Compile- Maven etc.

**Q What is Jenkins Dashboard?** Answer – The Dashboard consists of --a blue navigation bar at the top. – The pipelines list as well as –The Favorties list (Heath Icons and Run Status).

**Q Difference between Hudson and Jenkins?** There is no difference between the Hudson and Jenkins. Hudson was only the earlier name of Jenkins.

**Q Jenkins process in Continuous integration--? Testing**

**Developers 1,2,3🡪Source Code Repository🡪 Jenkins Server**

**Prepares a Build Production**

**Step 1-** First Developers commit changes to the source code.

**Step 2-** Continuous integration server **pulls that code and triggers a build.**

**Step 3-** The build application is then deployed on the testing server for testing .

**Step 4 –** After testing the application, it is then deployed on the production server**.**

**Step- 5** The concerned team are constantly notified about the build and the test Result.

**Q Describe the Nightly Build in Jenkins .**

**Code will be pulled only at night**

**Git Repository Build Server**

Commit changes to the source code Change made in the source code

throughout the day are build together.

Developers

…………………………………

Nightly built can be considered as a predecessor to continuous integration. There are developers who are committing changes to the code that is present in a shared repository. In the night, there is a build server. This build server will hold the shared repository for changes and then it will pull that code and prepare a build. So In that way whatever commits are made throughout the day are compiled in the night.

**Q How to create a build pipeline in jenkins?**

**To set up build pipeline First --**

1. **we create our jobs then**
2. **install build pipeline plugin to select the manage plugin under the manage Jenkins in Jenkins dashboard.**
3. **To show the jobs in a build pipeline we add a view name to click plus(+) option in the Jenkins dashboard and to add our jobs we select “+98 under upstream/downstream config.**
4. **At the end at ‘display options’ we usually raise number of displayed build to show all the last runs and to click the run button we can explore quickly.**

**To create a simple pipeline from the Jenkins interface, perform the following steps:**

1. Click New Item on your **Jenkins** dashboard, enter a name for your (**pipeline**) job, select **Pipeline**, and click OK.
2. In the Script text area of the configuration screen, enter your **pipeline** syntax.

**Jenkins**

**Q What is Continuous Delivery? How Jenkins fit into my work?**

**GitHub**

**Q. What is Git and Why do we use Git?**

Git is a distributed version control system and Source code management system designed to handle everything of small and large projects with speed and efficiency. **To use Git we** can maintain our software resource, source code, and documents. It is also a great place to Share our project within the team member.

**Git works in two ways -1**) **Local Repository**-Whenever we integrate our directory with Git and my work directory or folder in my local computer then it would be treated as Local repository. 2) **Remote repo/Online Repository-**Whenever we upload our local repository into Github or Gitlab then it would be treated as Remote repository through which we can access anywhere by the remote repository.

Local Repository and Remote repository are almost same but local repository exist in computer and Remote repository exist in server in which we can excess from any palce.

**Q. Difference between Git and Github?**

Git is a processor as like Engine. Git is a distributed version control system to mange Source code repository or history. While Github is like a server or networking system. It is a hosting service for git repositories. So Git is a tool and Github is the service for projects that user Git.

In Github you can’t create free private repositories for open source project. But in Git lab can create pvt repo.

**Q. Difference between Github and GitLab?**

Github is like a server or networking system. It provides service by which you can create your repository. While GitLab is a web based Devops life cycle tool. It provides a single operation for the entire software development and operations life cycle.

**Q What is staging Area in Git ?**

The staging area is a simple file, generally contained in Git directory that stores information about what will go into your next commit.

A head is simply a reference to a commit object. In every Repository there is a default head referred as “Master”. A repository can contain any number of Heads.

**Q: What is a clone in GitHub?**

Cloning a Git repository means we can create a local copy of the code provided by the developer.

You can simply do it with a command line**:git clone**

git://github.com/facebook/facebook-ios-sdk.git . and we can have the code in the facebook-ios-sdk directory.

**Q: How much space do we get on GitHub?** We get a space of 1 GB but if it exceeds 1GB, we receive a polite email from GitHub Support requesting to reduce the size of the repository and scale it down. In addition, here we placed a limit of files exceeding 100 MB in size.

**GitHub**

Q**: What do you know about GitHub and its repository?** Basically, It is a server or networking system for the projects that use Git. It is an open source and distributed service by which you can create your repository. It is also a hosting service for git repositories.

A Repository is basically the directory of the Git where the metadata of the same is stored. The data might be shared or private depending on the project. This repository tracks all change made to files in the project, building a history overtime.

**Q: How it is possible for you as a user of Git to define the information of a user, behavior of a repository as well as the information of preferences in the programming?**

This can simply be done with the help of a command named Git config. Although there are other methods but getting the results through the command prompt always make sure of originality and reliability.

**Q: Tell what exactly do you know about GIT stash?** It is used when there is a need of storing the current state of a project so that the users can continue with the same at a later stage. There is often a need to switch to another job when one is active and developers can simply keep up the pace with such a situation with the Stash. It simply enables the users to not to lose their edits.

**Q: Name the tool that can be deployed for Git migration?** SubGit

**Q: Can you tell us a few benefits of using the GitHub over other platforms?**

**There** are a very large number of benefits that developer can easily make sure of with this approach. The very first one is the high availability of the GitHub along with an excellent support. In addition to this, all the users can simply make sure of the data replication, as well as the redundancy of the same. Moreover, the error-free outcomes are exactly what for which GitHub is widely appreciated. It is a platform independent and users can easily get the results in the desired manner without compromising with anything. Also, it is collaboration-friendly and users can simply use it in the way they are comfortable.

**Q: What language is considered in Git and what is the benefit of same in this approach according to you?** Git is purely based on the C and the same make sure of imposing a limit on the overhead of runtimes which are generally associated with other platforms in its class. Also, c makes it compatible with all the other domains and developer’s already existing work.

**Q: Compare Git with SVM?** Git is an open source Distributed version control system. It is decentralized. It creates only .git folder or directory. It is a bit difficult to learn for some developer because it has more concept and command to learn. It does not have good UI. Git becomes slow when it deals with binary files that changes frequently. And it takes an extra step to create a new feature. **While SVM(sub version)** is a centralized version control system. In the SVM, You have to always connect to a central repo for check-in. It is much easier to learn. SVM can handle binary files easily. It create .svn directory in each folder. It has good UI. It is easier to create a new feature.

When it comes to handling the data with large size, Git is not widely preferred. In multiple branches, the Git fails to support the commits while the SVN can do so easily.

**GitHub**

**Q: When it comes to software development, what are the major factors the user should be careful about?**

The software should be developed by understanding the exact needs of the client or the task which it has to perform. It should be rich in terms of features and API. Moreover, it should be secure and reliable enough to be trusted by the organizations. There are other factors such as the length of the code and the factors that can influence the same which are also necessary to pay attention to.

**Q: On what projects you have already worked on which are based on Github?** This question is often asked in the IT interviews. You need to give a short or a detailed overview of the projects you have handled, the problems you faced, the outcome of the project, the benefits organization and you as a developer derived from it, the scope of the project and the time taken to complete it. Moreover, you should mention what sort of experience you derived from them.

**Q: What is the upper limit on the heads in the Gits?** There is no limit and users are free to involve any number of heads in the repository. It can be considered as a simple standard reference to a commit object. The commit object couldn’t be same for all the heads.

**Q: Tell us whatever you know about the Github development process?**

It is basically nothing but quite similar to that of a life cycle of any specific software. Thus, you should have knowledge about the life cycle of software and the factors that can directly influence the same. There are actually several activities which are a part of the process and they are: 1. Analysis of the requirement 2. Specifications of the project 3. Architecture of the software 4. Real time implementation 5. Testing of software 6. Documentation and reporting 7. Maintenance 8. Training and support available with the same.

**Q: Do you think GitHub is batter as compared to Subversion of same? Why or why not?** One of the best things about the Github is it’s an open source technology where the developers are free to run the versions of their projects without worrying about anything. This is exactly what that enable the developers to have a quick review of all the changes made to the code over time. The users can also keep a track or actual code before modification. With a sub version, nothing like this is possible. Therefore, Github is a better option.

**Learn Git Tutorial**

**Q: What is branching it Git and what are its benefits?** The users are free to make as many branches as they want. A branch is nothing but a set of tasks that is created by the users. A branch represents an independent line of development. It is a part of everyday development process. The first branch name is Master.

It allows us to go to previous work keeping your recent work intact. It actually boosts up speed and enables users to perform multiple tasks at the same time. You can create your own branch and jump between those branches. It allows us to work on different versions of the same files in parallel.

**Q: What was your biggest problem till date while working on a Github project?**

Generally, the problems are quite common in any development process. Depending on the nature and type of tasks, you might have faced a lot of issues. You can mention them all here and can genuinely answer how they were actually sorted out. Problem solving is actually learning how to tackle the challenges.

**Q: How many characters are there in the SHAI name?**

It is a 40 Character String that can vary in some special cases. 15. Can you put the Computer software and computer program separate from one another by a simple comparison? Basically, a computer program is nothing but the part of a programming code and the same is responsible for the successful execution of a task. On the other hand, computer software includes the code for programming including the guide on how to use it and concerned documents.

**Q: How it is possible for a developer to simply update the remote references related to the different objects?**

This is possible through the Git PUSH. In fact, it is actually the prime function of the same.

**Q: What do you know about the significance of software development?**

In the century we are living in, the overall time taken to accomplish a task largely matters when it comes to corporate level. The software are powerful in saving a lot of time. Also, they impose a strict upper limit on the dependency of a task on the humans. The tasks are generally governed, controlled, monitored and accomplished by the software in the current time. Thus, the scope is software development is blooming and users should pay attention to using the best available technology of the same.

**Q: Is it possible in the Git to merge a branch into the master? How can you find the same?**

Yes, it is possible and the users can easily keep the pace up with the same anytime. The users can directly check the list under the branch section to know more about this.

**Q: Are you familiar with the Git Clone?** It is basically a command which is deployed when it comes to copying a Git repository which already exists. There are a lot of programmers who make use of this. The best thing is it really doesn’t matter whether the project is large or small, the same can easily be considered whenever the need of same is felt.

**GitHub**

**Q: Name any two Git repository hosting services which are common These are Visual Studio Online and Git Enterprise Q: What is SHAI name?**

It is basically a string character which is responsible for the identification of the commit objects. The fact is users are free to make the changes to the default commit objects and the same is used for knowing and locating the overall changes made with a track record of the same.

**Q: Is it possible to create a repository in the Git? What is the step that needs to be performed before doing the same?**

Yes, it is possible and the users have to first create a directory. The same defines the project under consideration and the information related to the same.

**Q: What is Gitlog and when you can use it?**

It is basically a command that can be executed when it comes to finding the history of a project according to the date, changes made, the developer who handled it and usefulness of the same.

**Q: What is conflict situation in the Git and how it can be solved?**

A conflict arises when two separate branches have made edits to the same line in a file. Or when a file has been deleted in one branch but edited in the other. Conflicts will most likely happen when working in a team environment. The best manner to solve this concern is to simply edit the files through the appropriate procedure.

Basically we used branching concept before merging our code into main branch. We created another branch to implement, create, edit, deleting everything. After completing our branch team work, we push our branch to a head or in charge to verify everything. And then after verifying he merged into main branch. That’s why we did not face any conflict.

**Q: Can you name an alternative method for performing the merging task in the Git?**

This is called as Rebasing. Generally, the users don’t prefer this method and this is because it takes a lot of time for the accomplishment of a lot of tasks that matters a lot. Sometimes a lot of unexpected errors can declare their presence in case this task is not performed accurately. This is a method that is useful only for those who have a great experience in Git technology. Syntax of rebase is “git rebase[new-commit]”.

**Q: Name a few graphical Git clients for the Linux platforms?** These are Git-g, Git cola, Git GUI and Giggle These clients can easily be used in conjunction with each other or they can be deployed independently .

**Q: What is the significance of Git version control?**

When it comes to simply tracking the background of an array of files and changing their version, this approach is generally preferred. This actually works by capturing the snapshot of the moments and tasks associated. All the information remains present in the repository. However, if the users want, they can simply keep it at any desired location.

**GitHub**

**Q: What do you mean by the Commit message?**

It is basically a message which you can see on the screen while working on Git whenever a change is committed. It is possible to keep a record of all the changes made by the user in an editor. The history of changes needs to be explored lately for a specific task.

**Q: How can the users enhance the functionality of a branch in Git?** It is possible to do so by adding a desired feature in any of the branches. This is generally done through a command Git merge and the best part is there is no limit on adding the features in the branch. Any branch can have any number of features.

**Q: What is a gist in Git?** Gists are a great way to share the work of any developer. They can share parts of files, full applications or single files. Anyone can access gists at https://gist.github.com. Each Gist is Git repository, which means, it can be forked, and cloned.

**Q: How can we create a gist?** Creating a gist requires a very simple process as depicted in the steps below: - 1. Sign in to GitHub. 2. We should the navigate to the gist home page. 3. After this, we need to type an optional description and name for the gist. 4. Key in the text of your gist into the gist text box. 5. Following this we should select either to create a public gist or to create a secret gist.

**Q: What is a gist programming?** GitHub provides a hosting service that facilitates a web-based Git repository. It includes all the functionality of Git with additional features added in. The gist is an additional attribute added to GitHub, which facilitates the sharing of code snippets, notes, to do lists and more. We can save our Gist’s as secret or public in the repository.

…………

**GitHub**

If you’re one of those developers who still don’t use any version control system, I don’t know how you’re still managing to get work done. In this post, I’m focusing on important git commands that gets all (almost) your work done ( I know you’re a GUI person ).

**1) git config  Utility​** : To set your user name and email in the main configuration file. How to​ : To check your name and email type in ​git config --global user.name​ and ​git config --global user.email​. And to set your new email or name ​git config --global user.name = “Sharif”​ and ​git config --global user.email = “​sharifbd1975@yahoo.com​”

**2) git init  Utility**​ : To initialise a git repository for a new or existing project. How to​ : ​git init​ in the root of your project directory.

**3) git clone  Utility​** : To copy a git repository from remote source, also sets the remote to original source so that you can pull again. How to​ : ​git clone <:clone git url:>

**4) git status**  Utility​ : To check the status of files you’ve changed in your working directory, i.e, what all has changed since your last commit. How to​ : ​git status​ in your working directory. lists out all the files that have been changed.

**5) git add  Utility​ :** adds changes to stage/index in your working directory. How to​ : ​git add.

**6) git commit  Utility**​ : commits your changes and sets it to new commit object for your remote. How to​ : ​git commit -m”sweet little commit message”

**7) git push/git pull  Utility**​ : Push or Pull your changes to remote. If you have added and committed your changes and you want to push them. Or if your remote has updated and you want those latest changes. How to​ : ​git pull <:remote:><:branch:>​ and ​git push <:remote:><:branch:>

**8) git branch  Utility​ : Lists out all the branches. How to​ : ​git branch​ or ​git branch -a​ to list all the remote branches as well.**

**9) git checkout**

**Utility​ :** Switch to different branches How to​ : ​git ​checkout​ <:branch:> or \*\*\_git checkout -b <:branch:>​ if you want to ​create​ and switch to a new branch.

10**) git stash  Utility​** : Save changes that you don’t want to commit immediately. How to​ : ​git stash​ in your working directory. ​git stash​ apply if you want to bring your saved changes back.

11**) git merge  Utility​** : Merge two branches you were working on. How to​ : Switch to branch you want to merge everything in. git merge ​<:branch\_you\_want\_to\_merge:>

**12) git reset  Utility**​ : You know when you commit changes that are not complete, this sets your index to the latest commit that you want to work on with. How to​ : ​git reset <:mode:><:COMMIT:>

**13) git remote  Utility​** : To check what remote/source you have or add a new remote. How to​ : ​git remote​ to check and list. And ​git remote add <:remote\_url:>

These are the commands that I feel are essential and get things done, at least for me. Comment here if you think I’ve missed something important or if something can be done differently.