Q1) What is GIT?

Answer: Git is a conveyed version control system for following changes in source code amid programming improvement. It is intended for organizing work among software engineers, yet it tends to be utilized to follow changes in any arrangement of documents. Its objectives incorporate speed, information respectability, and backing for appropriated, non-straight workflows.

Q2) What is the command you can use to write a require message?

þÿ A n s w e r : g i t c o m m i t m , m i s o u r m e s s a g e .

Q3) What is the variance between GIT and SVN?

Answer:

GIT :

Git is a circulated VCS; .

Git has a brought together server and store,

The content in Git is put away as metadata,

Git branches are simpler to work with than SVN branche

SVN :

SVN is a non-appropriated VCS

SVN does not have a unified server or store

SVN stores documents of substance

Q4) What are the benefits of using GIT?

Answer: One of the greatest focal points of Git is its fanning abilities. Not at all like incorporated adaptation control frameworks, Git branches are modest and simple to combine. This encourages the component branch work process prevalent with numerous Git clients. Highlight branches give a confined domain to each change to your code base.

Q5) What language is used in GIT?

þÿ A n s w e r : C o m m o n l y t h e l a n g u a g e s u s e d i n G i t i s , C , P y t h o n , P e r l

þÿ Q 6) W h a t i s t h e m e a n i n g o f G I T P U S H i n G I T ?

Answer: Git gives you a ton of approaches to allude to a commit however for most purposes þÿ you ll simply utilize the name of a branch with git push

Q7) Why GIT better than Subversion?

Answer: Designers like Git on account of its successful spreading model. In Git, branches are just references to a certain commit, making them lightweight yet ground-breaking. Git enables you to create, erase, and change a branch whenever without influencing the commits

Q8) What is Production Area or Index in GIT?

þÿ Answer: The git index is the place you place documents you need comr
The index is otherwise called store, registry reserve, current catalog reserve, organizing territory, þÿ arranged documents. Before you commit (checkin) records to the git st þÿ initially put the documents in the git index .

Q9) What is GIT stash?

Answer: Stashing takes the filthy condition of your working registry that is, our adjusted tracked documents and organized changes and spares it on a heap of incomplete changes that we can reapply whenever.

Q10) What is GIT stash drop?

þÿ Answer: Git stash is a transitory stockpiling. When you re prepared to p
known point of interest, you can reestablish the spared state effectively: git stash pop . Popping your reserve expels the progressions from your reserve and reapplies the last spared state.

Q11) Name a few Git repository hosting services

Answer:

GitHub

Pikacode

Visual Studio Online

GitEnterprise

net

Q12) What is the function of git clone?

Answer: git clone is a Git direction line utility which is utilized to focus on a current archive and make a clone, or duplicate of the objective vault.

þÿ Q 1 3) W h a t i s t h e f u n c t i o n o f g i t c o n f i g ?

Answer: The git config direction is a comfort work that is utilized to set Git setup esteems on a worldwide or neighborhood venture level. These design levels relate to .gitconfig content documents. Executing git config will alter an arrangement content document.

Q14) What does commit object contain?

Answer: The commit object contains the index tree object hash, parent commit hash, creator, committer, date and message.

Q15) How can you make a repository in Git?

Answer: In Git, to generate a repository, create a directory for the project if it does not exist, and then run command git init. By running this command . git directory will be generated in the project directory, the directory does not need to be empty.

Q16) What is head in git and how many heads can be generated in a repository?

Answer: There can be any number of heads in a GIT vault. As a matter of course there is one head known as HEAD in every store in GIT

Q17) How do resolve merge conflicts?

Answer:

Open Terminal Git Bash the terminal.

Navigate into the local Git archive that has the consolidation conflict.

Produce a rundown of the records influenced by the consolidation conflict.

Open your preferred content tool, for example, Atom, and navigate to the record that has consolidate conflicts.

Q18) How do we push a branch to github?

Answer:

þÿ M a k e a n o t h e r b r a n c h : g i t c h e c k o u t b f e a t u r e _ b r a n c h _ n a m e .
Alter, include and submit your
records.
þÿ P u s h y o u r b r a n c h t o t h e r e m o t e r e p o s i t o r y : g i t p u s h u s o u r c e f e a t u r e _

Q19) What is Release Branch in Git

Answer: The master and develop branches structure the base of any vault in Git. The master branch contains a version of the code that is underway, and the develop branch contains a version that is expected to be discharged in a forthcoming version.

Q20) What is a conflict in git?

Answer: Merge conflicts happen when you merge branches that have contending commits, and Git needs your assistance to choose which changes to fuse in the last merge.Usually, the progressions are on various lines, or even in various documents, which makes the merge basic for PCs to get it.

Q21) How can conflict in git resolved?

Answer: Conflicts for the most part emerge when two individuals have changed similar lines in a

document, or in the event that one designer erased a record while another engineer was altering þÿit. & Git will check the document as being tangled and end the blending þÿdesigners duty to determine the contention.

Q22) How to remove a file from Git Tracking ?

Answer: We need to remove it from your tracked records (all the more precisely, remove it from your arranging territory) and afterward submit. The git rm order does that, and furthermore þÿremoves the document from your working catalog so you don t consider i record whenever around.

Q23) What is alternative option for merging in git?

Answer: Git rebase is the alternate option for merging in Git

Q24) What is the syntax for Rebase in git ?

þÿAnswer: git rebase [-i | interactive] [] [exec] [onto][[]]

Q25) What is the variance between git remote and git clone?

Answer: The main difference between git clone and git remote is that git clone is utilized to make another nearby repository while git remote is utilized in a current repository.

Q26) What is GIT version control / Benefits of Version Control ?

Answer: Version control systems are a classification of programming apparatuses that assistance a product group oversee changes to source code after some time. Version control programming monitors each adjustment to the code in an uncommon sort of database. In the event that an oversight is made, designers can look to better days and contrast prior versions of the code with assistance fix the misstep while limiting disturbance to all colleagues.

Q27) What is Sub git?

þÿAnswer: Subgit is a device for a smooth, tranquil SVN to Git movement
þÿan organization wide movement from SVN to Git that is:

Q28) Why to use Subgit?

It is vastly improved than git svn
No prerequisite to change the foundation that is as of now put
Allows to utilize all git and all sub adaptation highlights
Provides authentic tranquil relocation
experience.

Q29) What is the meaning of git different in git?

þÿAnswer: Git (/ajt/) is a circulated adaptation control framework for follo
þÿcode amid programming advancement. & Git was made by Linus Torvalds
improvement of the Linux part, with other portion engineers adding to its underlying
advancement.

Q30) What is git status is used for?

Answer: git status. The git status order shows the condition of the working registry and the
organizing region. It gives you a chance to see which changes have been arranged, which
þÿhaven t, and which documents aren t being followed by Git.

þÿQ31) What is the variance between the git diff and gi

Answer: Distinction between git status and git diff

Git diff direction will demonstrate to you the careful distinction, or state in the event that you
added a few lines to a document, it will demonstrate to you those lines which are altered. A
þÿstraightforward git diff direction will reveal to you the distinction betw
condition of the documents, and the condition of the records in organizing zone

Q32) What is the meaning of git checkout in git?

þÿ A n s w e r : I n G i t t e r m s , a c h e c k o u t i s t h e d e m o n s t r a t i o n o f e x c h a n g i n g b
adaptations of an objective element. The git checkout direction works upon three particular substances: records, submits, and branches.

Q33) What is the function of git rm?

Answer: The essential capacity of git rm is to expel followed documents from the Git list. Furthermore, git rm can be utilized to expel documents from both the arranging file and the working index.

Q34) What is the meaning of git stash apply?

þÿ A n s w e r : g i t s t a s h b r i e f l y r e t i r e s (o r s t a s h e s) t r a n s f o r m s y o u v e m a d e t o
you can take a shot at something different, and after that return and re-apply them later on.

Q35) What is the use of git log?

Answer: Git logs enable you to audit and peruse a background marked by everything that happens to a vault. The history is constructed utilizing git-log , a basic apparatus with a huge amount of alternatives for showing submit history.

Q36) What is git add is used for?

Answer: At the point when utilized without anyone else, git include will advance pending changes from the working catalog to the arranging territory. The git status order is utilized to analyze the present condition of the vault and can be utilized to affirm a git include advancement.

Q37) What is the meaning of git reset?

Answer: It resets the record, yet not the work tree. This implies every one of your records are flawless, however any contrasts between the first submit and the one you reset to will appear as neighborhood adjustments (or unmanaged documents) with git status.

Q38) What is git Is-tree?

Answer: Th Working Tree in Git is a catalog (and its records and subdirectories) on your þÿdocument framework that is related with a vault. & When you open the r being overseen as a Git archive then you are get to the Working Tree.

Q39) How git instaweb is used?

Answer: The internet browser that ought to be utilized to see the gitweb page. This will be passed to the git web{litdd}browse aide content alongside the URL of the gitweb.

þÿQ40) What does hooks consist of in git?

Answer: Git hooks are contents that Git executes previously or after occasions, for example, þÿsubmit, push, and get. Git snares are a worked in highlight no compell þÿanything. Git hooks are run locally. These hooks contents are just restr creative ability.

Q41) Explain what is commit message?

Answer: We will in general pound the subject of a submit message with the body: Writing a subject of a submit together with the body of that submit message is the incorrect approach. When you see your submit message is getting too long to even think about explaining this implies submit is doing such a large number of things separate it

Q42) How can you fix a broken commit?

Answer: fixing broken submit messages. submitted 12 Jan 2009. You recently dedicated that þÿmarvelous component/test/bug, however something simply isn t right. Ei

Q43) Why is it advisable to create an added commit rather than amending an existing commit?

þÿAnswer: git commit amend in reality just influences increments to the þÿcan organize documents .. We ve included this in JIRA as a component þÿadequate for me, I m authoritatively prepared to change to SourceTree

þÿQ44) What is bare repository in GIT?

Answer: An exposed Git repository is commonly utilized as a Remote Repository that is sharing a þÿrepository among a few unique individuals. You don t do work directly in þÿso there s no Working Tree (the documents in your project that you alte repository information.

Q45) What is the use of Version Control System?

Answer: Version Control System can be defined as a software which allows several developers to þÿwork simultaneously and also keeps the entire history of the tasks perfo þÿallow developers to make over writing of other s work. There are two typ systems available:

Distributed or Decentralized Version Control System

Centralized Version Control system.

Q46) Explain centralized version control system and state its disadvantages.

Answer: CVCS or Centralized Version Control System allows a centralized server to save files in it and allows access to several people. The main advantage of CVCS is, it allows whole team to communicate at a time in a centralized server. But, CVCS fails with the central server. If the centralized server shuts down for a period of time, communication to the central sever completely stops. Sometimes, there is a risk of data loss and through which the data stored in the central server may get lost. This happens because of the corrupted drive in the central server.

Q47) Explain Distributed version control system.

Answer: DVCS overcomes the disadvantages of CVCS. This system allows its clients to take snapshots of the current directory. It also saves a duplicate copy (mirror) in the repository. Because of this feature, a client can restore his data though the server is down.

Q48) What do you mean by GIT?

Answer: GIT is basically a version control system and is a DVCS. GIT uses revised version control system of DVCS. It is also a scripts management system which manages source codes with enhanced speed. GIT is available open source with General Public License Version II of GNU. Since GIT follows DVCS strategy, it never relies on a centralized server. GIT allows its users to perform several functions offline. User can perform the changes he wish, he can visualize the logs, can form branches and so many functions in offline itself. User need network connection alone to modify the content he wishes to do and can consider the current change he made.

Q49) What are the advantages of GIT?

Answer: GIT has so many advantages as follows:

- GIT is available as Open source software with GNU license and is available free.
- Since GIT performs its functions locally, user will get good speed.

Since GIT follows DVCS strategy, it does not depend on a central server.

- User can able to perform several operations offline.
- GIT scripts depend on C language and therefore runtime overheads occur in high level languages can be prevented.
- Since GIT takes a duplicate copy, a mirror of the complete depository, it saves storage space because the size of the data will be much small.

GIT does not require any powerful hardware.

GIT provides enhanced security to the stored contents in the database using secure hash function called SHA1.

- The branch management in GIT is quite simple. GIT takes seconds to build, merge and delete branches.

þÿQ50) What is Local Repository in GIT?

Answer: Generally version control software contains a separate working space. In that space, developers modify the scripts. Once they finalize, that will be stored in the repository. GIT tool provides a replica copy of the entire repository. Developers can make changes as per their requirement in that repository itself. They can add a file, rename a file, delete a file, move a file, make modifications etc.,

Q51) State the differences between SVN and GIT.

Answer:

SVN	GIT
SVN allows users to handle many projects to be stored in the same repository.	GIT users don’t prefer to handle many projects from a same repository.
SVN follows Centralized version control system	GIT follows Decentralized version control system
SVN allow only online commits	GIT allows commits even in offline
`Slower Push/Pull operation	Quicker Push/Pull operation
SVN does not share anything automatically	Commit automatically shares all the works.

Q52) Explain the basic workflow of GIT.

Answer: The basic workflow of GIT can be divided into three steps. They are:

User can alter a file from the current directory

User can add those files into the staging area

þÿUser can do commit operation. Commit transfers files from staging. A P
in the GIT repository permanently.

þÿ Q 5 3) W h a t d o y o u m e a n b y b l o b s ?

Answer: Blob refers to Binary Large Object. In GIT, every version of a file is called as a blob. Generally, a blob will contain file data whereas it will not store any metadata about a particular file. Simply, blob is a binary file. In the DB of Git, blob is named as SHA1(Secured hash function) þÿ h a s h o f t h a t p a r t i c u l a r f i l e . G e n e r a l l y , G I T w o n t a d d r e s s i t s f i l e s b y n a content addressed.

Q54) What is a tree in GIT?

Answer: Tree refers to an object which means a directory. Tree keeps blobs and sub directories in it. A tree is nothing but a binary file which saves the references of blobs. A tree is a SHA1 hash of tree object.

Q55) Explain commit in GIT.

Answer: Commit keeps the working state of the repository. A commit can also be named by the secure hash function, SHA1. A commit object can be considered as a node of the linked list. Each of this commit objects will contain a pointer to the parent commit object. To track the history of a commit object, we can look into the parent pointer to get the same. If a commit object contains several parent commits, that commit can be built by combining two branches together.

Q56) What do you mean by branches in GIT?

Answer: Branches were meant to build yet another line of development. Git contains a master branch by default similar to trunk in subversion. Generally, branches were built to work on something new. Once the meant work is completed, that particular branch is merged to the master branch and that particular branch is deleted. Further, each branch is referenced by Head, which is a pointer that points to the current commit in the branch. If user makes a commit, Head is upgraded with the current commit.

Q57) What is the purpose of Tags?

Answer: The purpose of tags is to allocate some meaningful name with a particular version

present in the repository. Tags and branches looks alike, but tags are permanent. So, tag is a branch but no one can make changes to it. If an user build a tag for a specific commit, though he create yet another commit, the same will not be updated. Developers generally build tags for product releases.

Q58) Explain Clone operation in GIT.

Answer: The purpose of clone operation is to build a specimen or a mirror of the repository. Clone verifies the current working copy and also creates a mirror of the entire repository. It allows users to perform various functions in the local repository. All these operations can be performed offline without even a network connection. But, if the user needs to sync the repository specimens, a network connection is required to perform synchronization.

Q59) Explain Push operation in GIT.

þÿ Answer: A push copy the changes made from a local repository instanc
repository. This saves the changes into the Git repository permanently. A Push operation is similar to a commit operation in subversion.

Q60) Explain Pull operation in GIT.

þÿ Answer: A pull copy the changes made from a remote repository to a lo
if we need to synchronize between two repository instances, pull operation can be performed. A Pull operation is similar to update in the subversion.

þÿ Q 6 1) W h a t d o y o u m e a n b y H e a d ?

Answer: Head refers to a pointer which points to the current commit in the branch. If an user makes a commit, Head is upgraded with the current commit.

Q62) What is the language used in Git?

þÿ Answer: The language used in Git is C . Since C avoids the runtime o
level programming languages and therefore it is quick. So, Git is quicker. Also, C language is

very easy to learn and is user friendly. Program execution is faster in C and it contains simple syntax.

Q63) What do you mean by revision in Git?

Answer: Revision refers to the version of the source code which is written by a developer. Generally revisions in Git amount to commits. User can identify the commits through the SHA1 secured hashes.

Q64) What is an URL in the Git?

Answer: URL in Git is used to save the config file and it represent the place where the Git repository is located.

Q65) What is git config in Git?

Answer: Git supports a tool to set configuration variables, called as git config tool. All the global configurations in .gitconfig file present in the current working directory or home directory is stored by Git. But, user has to set the variables as global otherwise, by default it is stored in the local þÿdirectory. For that, user has to set the configuration parameters to glob option.

Q66) How do you install Git client in a machine?

Answer: The installation of Git client varies according to the corresponding linux platform you are using. The syntax varies between various linux platforms. For an instance, if the user uses RPM based Linux/GNU distribution, use yum command :

```
þÿyum  y install git-core
```

If the user uses Debian based Linux/GNU distribution, use apt-get command as follows:

```
sudo apt-get install git-core
```

To identify the git version, you can check using:

þÿ g i t v e r s i o n

Q67) What is the first step to establish connection to Git?

Answer: Like every setup, initialization is the first step to establish connection to Git. Once you finish the initialization of git, do push/pull source code. The command to initialize Git is:

$git init

Q68) What is the command to get the source code from Git hub?

Answer: The command is:

$git remote add origin git url

$git pull origin repository name

Q69) What is the command to clone git hub repository?

Answer: The command is:

git clone url project-name/foldername -b develop

Q70) What do you mean by git bash?

Answer: Git bash us a source controller which is used to save the source code developed by the developer.

Q71) What is the command to check git status?

Answer: The following is the command to check git status:

$git status

Q71) How do you add new files to the git directory?

Answer: The command to add new files is:

$git add A

Q72) What is the command to make commit?

Answer: The command to make commit is :

þÿ$ g i t c o m m i t - b C o m m i t d e s c r i p t i o n

This command will commit the files.

Q73) How will you push the committed source code to the server?

Answer: The command to push the committed source code to server is:

git push -u origin master

Q74) Write the syntax to do rebasing in Git.

Answer: The syntax for rebasing is

$ git rebase [new-commit]

Q75) What is the use of Git instaweb?

Answer: Git instaweb lets a web browser automatically and runs the web server through an þÿ i n t e r f a c e t o t h e u s e r s l o c a l r e p o s i t o r y .

Q76) Explain the life cycle of Git.

Answer: The life cycle of Git contains the following stages:

- User has to clone the Git repository as a current/working copy.
- User can alter the current/working copy by editing/adding files.
- **It is also possible to consider changes from other developers and the same can be updated.**
- Before commit, user has to review the changes performed.
- After review, commit. Later push the changes to the repository.
- If suppose something goes wrong, make corrections to the last commit and then push that changes to the repository.

Q77) How do you get a Git Repository?

Answer: Git repository can be obtained in any of the following two ways:

- User can use a local directory which is not under version control that time. Convert that to a Git Repository.
- User can take an existing Git Repository and can clone it.

þÿ B o t h t h e s e o p t i o n s w i l l c r e a t e a G i t R e p o s i t o r y i n t h e u s e r s l o c a l m a c h

Q78) How do you view the commit history?

Answer: After creating various commits or by cloning a repository using an existing commit history, user can check the commit history using git log command. Git hub contains simple þÿ e x a m p l e s i n a b a s i c p r o j e c t f o l d e r c a l l e d s i m p l e g i t . Y o u h a v e t o r u n t command:

$git clone https://github.com/schacon/simplegit-progit

$git log

The output could looks like this:

$git log

Commit dg56agghbx34987ahj4567845z67a4765az790dc43

þÿ A u t h o r : & & & & & & & & & & & . .

þÿ D a t e & & & & & & & & & & & & & . .

Version number changed

Commit 76545adgg745a564dg7421j45690xcz3679axcvbn6

þÿ A u t h o r : & & & & & & & & & & & . .

þÿ D a t e & & & & & & & & & & & & & . .

Removed unwanted test

Commit 124adhkk468900afcjlll129kk480837adcv4689axny8

þÿ A u t h o r : & & & & & & & & & & & . .

þÿ D a t e & & & & & & & & & & & & & . .

First commit

The git log command lists the commands present in that repository in the reverse order by default. The most recent commit will be shown first. `In the command list present above, it contains commit along with the secured hash SHA1 checksum, name of the author, his e-mail id, date authored and the commit message.

Q79) What is the difference between author and committer in Git?

Answer: Author is the person who authored the project i.e., who originally wrote the project. Committer is the person who committed changes to the project recently. Therefore, if an user sends a patch to a project and if any of the member in the group applies the patch, both the author and the committer will be credited.

Q80) How will you undo things in Git?

Answer: Generally, humans were prone to errors. Git allows some basic tools to undo changes þÿ which are made by a user. But, we can t undo everything. So, user has t a work or he may suffer in loss of the work in Git. Commonly, we use this undo option while we commit very early and forgot to add some files to it. Sometimes, the commit message might go wrong. In those cases, user can redo that commit. User can change he actually wanted to do and þÿ then he can commit again through amend option.

þÿ $ git commit amend

This command uses your staging for the commit.

Q81) What do you mean by remote repository in Git?

Answer: If the user wishes to combine or merge on any Git project, he must know about managing the remote repositories. Remote repository refers to a version of a project which is hosted over the internet or in a network elsewhere. We may find various remote repositories and every repository among them will perform read/write operations or read-only operations. Merging þÿ with some other s work needs to manage these remote repositories. Use and fro from them while sharing our work to others. Managing remote repositories will perform many of the following tasks: user has to know the manner to add remote repositories, delete some remote repositories which are not valid, has to manage several branches by defining every branch whether they are tracked or not and many more.

Q82) How will you add remote repositories in Git?

Answer: User can use git clone command which actually adds the origin remote for him. For an instance, if the user wishes to add a new remote repository in Git as a short name, run this

command:

$git remote add<Short name><URL>

$git remote

Origin

$git remote add pb https://github.com/author1/ticgit

$git remote-v

þÿ O r i g i n h t t p s : / / g i t h u b . c o m / a u t h o r 1 / t i c g i t & & & & & & & . F e t c h i n g

þÿ O r i g i n h t t p s : / / g i t h u b . c o m / a u t h o r 1 / t i c g i t & & & & & & & & P u s h i n g

þÿ P b h t t p s : / / g i t h u b . c o m / a u t h o r 2 / t i c g i t & & & & & & & & & & F e t c h i n g

þÿ P b h t t p s : / / g i t h u b . c o m / a u t h o r 2 / t i c g i t & & & & & & & & & & . P u s h i n g

Instead of using the URL, we can use the string pb on the command line now. For an instance,
þÿ l e t s c o n s i d e r t h e u s e r l i k e s t o f e t c h t h e e n t i r e i n f o r m a t i o n f r o m a u t h o r
his repository, he can use git fetch pb command.

$git fetch pb

Remote: counting objects, 59 completed.

Remote: compressing objects: 100% (48/48), completed

Remote: Total 59 (delta 13), reused 43 (delta 8)

Unpacking objects: 100% (59/59) completed

From https://github.com/author2/ticgit.

þÿ*[new branch] master && pb/master

þÿ*[new branch] ticgit && &pb/ ticgit

Information present in the repository of author 2 will be accessed locally through pb/master. User can now combine it with any of the branches.

Q83) How will you list your tags in Git?

Answer: Git has a support to add tags to some particular points in the history of repository which we feel important. Adding tags is one of the most common feature present in Version control systems. Generally, users use this feature to spot release points. User can list the existing tags þÿquite easily in Git. The command git tag will list the tags.

$git tag

V1.0

V2.0

This command will list the tags used in the alphabetical order.

Q84) What are the types of tags that Git supports? Explain them.

Answer: There are two types of tags supported by Git. They are:

- Light weight tag
- Annotated tag

Light weight tags never changes. It works like a branch. These tags is actually a pointer to a particular commit.

Annotated tags were stored as entire objects in the Git DB. These type of tags were check summed and consists of the name of the tagger, date, his mail id, the tagging message. Further, the tag has to signed and verified by the GPG (GNU Privacy Guard).

Q85) How will you create annotated tags?

þÿ A n s w e r : U s e r c a n c r e a t e a n a n n o t e d t a g i n G i t q u i t e e a s i l y . g i t t a g a w tag. The command to create such a tag is:

þÿ $ g i t t a g a v 1 . 5 m u s i n g v e r s i o n 1 . 5

$ git tag

V0.1

V1.3

V1.4

Here, -m will contain tagging message specified by the tagger, which can be saved with the tag. If you want to see the tag data present and also the commit which is tagged using git, you can use the git show command. This command shows the information of the tagger, the date of the commit tagged and also the tagger message and the commit information.

Q86) Write the workflow of branching and merging with some example?

Answer: Branching and Merging in practice requires some stage of operations to be performed.

þÿ L e t s e x p l a i n t h e p r o c e s s w i t h a s i m p l e e x a m p l e

- Make some work on a website
- Build a branch for a new user

Now, start doing some work in the created branch

If suppose, some bug occurs at this stage and it is need to be fixed asap, perform the following operations:

- þÿ M o v e t o t h e u s e r s p r o d u c t i o n b r a n c h
- Build a branch to clear the bug

Once you test it, combine the hotfix branch and then push it to production

- Move to your actual work from where you switched and start working on it.

Q87) What do you mean by long running branches?

Answer: Because of its enhanced feature support to the users, Git offers three way easy merging of one branch to another branch several times for a long period. Many branches were open always which can be used in various stages of your development cycle which can be merged often from few of them to others. The code in the master branch will be completely stable which is the only possible code which has released. Developers who use Git generally uses this kind of þÿ t e c h n i q u e . D e v e l o p e r s u s e y e t a n o t h e r p a r a l l e l b r a n c h c a l l e d n e x t o r the stability but there is no certainty to be stable always. Once it reaches the stable state, it can further be merged to the master. Normally, they are used to pull from topic branches once they are stable which ensures they could clear all the tests without any errors.

Q88) What do you mean by topic branches?

Answer: As opposed to long running branches, topic branches are short lived branches. If an user need to create a project of any size, he can go for topic branches. These type of branches can be built and used for one specific feature or task. This particular feature will not be available in any of the version control system since because this feature is not cost effective to build and combine branches. But, Git supports this feature. Using Git, this can be very ordinary for building, merging, working and deleting branches many times in a day.

Q89) How will you track remote branches in Git?

Answer: Tracking remote branches in Git can be done by checking a local branch. This monitoring will create a branch automatically which is called tracking branch. The branch a tracking branch tracks is so called as an upstream branch. These tracking branches will establish a direct connection with a remote branch. They are local branches. Tracking a remote branch can þÿbe done using git pull. Once you type this command, Git automatically to be fetched and which branch has to be merged.

Q90) Explain the protocols that Git uses to transfer data.

Answer: To transfer data, Git mainly uses the following four protocols:

- Local
- HTTP
- SSH and
- Git

Local protocol is the generally used protocol by developers and is the very basic protocol. In a local protocol, the remote repository is present in another directory with the same host. This protocol will be used if every member in a team enjoys access to a shared file system like NFS mount or sometimes every person in the team create logs from a same computer. The second option might create catastrophic loss generally since all of your code repositories will be present in the same computer and therefore using this option is not advised.

Git also uses HTTP protocol for communication using two different modes: Smart HTTP and Dumb HTTP. Smart HTTP protocol functions much similar to SSH or Git protocols but could run in standard HTTP ports using several authentication mechanisms of HTTP. If suppose the server þÿhaven t responded with smart HTTP mechanism, the Git client will switch HTTP protocol. Dumb protocol can be set up quite simply. User has to place a bare Git repository in HTTP root document and has to set up a particular post-update hook. After this, anyone who wish to access the web server where you put the repository and also they can clone the repository.

SSH protocol is another common protocol for data transport for Git to self host through SSH.

þÿ S S H p r o t o c o l i s s e t u p m o r e c o m m o n l y i n m o s t p l a c e s a n d i t i s v e r y s i m p

also an authenticated network protocol

Git protocol contains a special daemon which contains as a package with Git. This protocol

þÿ w o r k s o n a d e d i c a t e d p o r t a n d d o e s n t c o n t a i n a n y a u t h e n t i c a t i o n . T h i s

SSH.

Q91) How can you resolve merge conflicts through command line?

Answer: Generally merge conflicts happen while user alters one line of a file or if one deletes the same file and another edits the same file. To resolve such conflict, user has to choose the change we need to make from several branches in a new commit. Before merging branches, user has to resolve this kind of merge conflict with a new commit.

- Open the terminal window : TerminalGit Bash the terminal
- Move to the local Git repository which contains the merge conflict

þÿ c d R E P O S I T O R Y N A M E

Q92) Create list of files which contains merge conflict?

- Open the text editor of your wish (namely Atom) and move to the file which contains merge conflicts.
- To identify the starting point of the merge conflict from your file, search that file for conflict marker.
- þÿ I f t h e u s e r l i k e s t o r e t a i n h i s b r a n c h s c h a n g e s a l o n e , r e t a i n t h a t c h a

change which will make changes in both of the branches. Then, delete the conflict markers and do changes in the final merge you wish to make.

Now, add a change or stage your changes

$ git add

Then, commit your changes using some comment.

þÿ$ git commit m Merge Conflict is resolved

Q93) What do you mean by a bare repository?

Answer: Generally a bare repository in Git will contain details about the version control and the
þÿidle files (files not working) and also it won t have the special Git sub d
will contain the Git sub directory contents in its main directory and the current working directory
þÿwill contain i) a Git sub directory comprising of Git connected history of
þÿverified copies of user s project file(working tree).

Q94) Explain the steps to resolve a conflict in git?

Answer: The following are the steps to resolve a conflict in Git:

Locate the files which generated the conflict.

þÿDo required changes to the files and ensure the conflict won t repeat

Then, using $ git add, add the files

Using $ git commit, commit the changed file.

Share on Facebook

Share on Twitter

Share on Linkedin

Share on Pinterest