**How do you revert a commit that had already been pushed and made public?**

creates a new commit with patches that cancel out the changes introduced in specific commits. In case the commit that needs to be reverted has already been published or changing the repository history is not an option, *git revert* can be used to revert commits. Running the following command will revert the last two commits:

git revert HEAD~2..HEAD

**How do you squash last N commits into a single commit**?

Squashing multiple commits into a single commit will overwrite history, and should be done with caution. However, this is useful when working in feature branches. To squash the last N commits of the current branch, run the following command (with {N} replaced with the number of commits that you want to squash):

git rebase -i HEAD~{N}

Upon running this command, an editor will open with a list of these N commit messages, one per line. Each of these lines will begin with the word “pick”. Replacing “pick” with “squash” or “s” will tell Git to combine the commit with the commit before it. To combine all N commits into one, set every commit in the list to be squash except the first one. Upon exiting the editor, and if no conflict arises, *git rebase* will allow you to create a new commit message for the new combined commit.

**What is GIT stash?**

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.  So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

**What is the function of ‘git stash apply’?**

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

**How will you know in GIT if a branch has been already merged into master?**

Git branch —merged lists the branches that have been merged into the current branch

Git branch —no merged lists the branches that have not been merged

**What is a ‘conflict’ in git?**

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

**How can conflict in git resolved?**

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge,  run “git commit”.  Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

**What is a Jenkins job?**

A Job/Project is the fundamental unit of a logical work (like a software build, an automation task, test execution, etc) using the Jenkins automation server and other required plugins, configurations & infrastructures.

Jobs can be of different types like - a freestyle project, a multi-configuration project, a pipeline project, a multi-branch project, etc.

**How can you set up a Jenkins job?**

To set up a Jenkins job, you may follow these steps:

* Select New item from the menu
* Next, enter a name for the job and select a free-style job
* Click on OK to create a new job
* Hence, the next page that appears will allow you to configure your job.

**Name some more continuous Integration tools other than Jenkins.**

Some of the top continuous integration tools other than Jenkins are:

* TeamCity
* Travis CI
* Go CD
* Bamboo
* GitLab CI
* CircleCI
* Codeship

**Explain the process in which Jenkins works?**

The process in which Jenkins works is as follows:

* Jenkins checks changes in repositories regularly, and developers must secure their code regularly.
* Once the changes are defined, Jenkins detects them and uses them to prepare a new build.
* After that, Jenkins will transverse through various stages in its usual pipeline. As one stage completes, the process will move further on to the next stage.
* If a stage fails, Jenkins build will stop there, and the software will email the team using it. When completed successfully, the code implements itself in the proper server so that testing begins.
* After the successful testing phase, Jenkins shares the results with the team using it.

**Name the Jenkins suite’s essential plugins?**

The Jenkin suite’s essential plugins are Docker, Jira, Slack Notification, Maven, Amazon E2C, jUnit, Pipeline, Mailer, and Greenballs.

**What is a trigger? Give an example of how the repository is polled when a new commit is detected.**

Triggers are used to define when and how pipelines should be executed.

When Jenkins is integrated with an SCM tool, for example, Git, the repository can be polled every time there is a commit.

* The Git plugin should be first installed and set up.
* After this, you can build a trigger that specifies when a new build should be started. For example, you can create a job that polls the repository and triggers a build when a change is committed.

**Name a Jenkins environment variable you have used in a shell script or batch file.**

There are numerous environment variables that are available by default in any Jenkins build job. A few commonly used ones include:

* $JOB\_NAME
* $NODE\_NAME
* $WORKSPACE
* $BUILD\_URL
* $JOB\_URL

Note that, as new Jenkins plug-ins are configured, more environment variables become available. For example, when the Jenkins Git plug-in is configured, new Jenkins Git environment variables, such as $GIT\_COMMIT and $GIT\_URL, become available to be used in scripts.