Troubleshoot Guide for GIT

Conflict resolution is one of the main challenges that is faced by the beginners.

|  |  |
| --- | --- |
| Presenter: | Neha Asthana |
| Topic: | GIT, GIT HUB |
| CONFLICTSGit conflicts may arise due to various reasons, few are listed below for easy troubleshooting | Probable best resolution in each scenario is listed below |
| Merge Conflict:Two devs are working on different parts of the same code file and try to merge their changes. However, there are conflicting changes that cannot be automatically resolved. | * **git fetch** - to get the latest changes from the remote repository * **git checkout <branch>** - to switch to the branch where the merge will happen * **git merge <other\_branch>** - to merge the changes from the other branch into the current branch * resolve the conflicts manually by editing the conflicting files * **git add <conflicting\_file>** - to stage the resolved files * **git commit** - to commit the changes with a message explaining the resolution * **git push** - to push the changes to the remote repository |
| **Push Conflict:**Two testers are working on the same branch and both try to push their changes to the remote repository. However, only one of them can push at a time, causing a conflict. | * **git pull** - to get the latest changes from the remote repository and merge them with the local changes * resolve the conflicts manually by editing the conflicting files * **git add <conflicting\_file>** - to stage the resolved files * **git commit** - to commit the changes with a message explaining the resolution * **git push** - to push the changes to the remote repository |
| Branch Conflict:Two testers create separate branches to work on different features, but they both end up modifying the same file. When they try to merge their branches, there are conflicts that need to be resolved. | * **git checkout <branch>** - to switch to the branch where the merge will happen * **git merge <other\_branch>** - to merge the changes from the other branch into the current branch * resolve the conflicts manually by editing the conflicting files * **git add <conflicting\_file>** - to stage the resolved files * **git commit** - to commit the changes with a message explaining the resolution * **git push** - to push the changes to the remote repository |
| Pull Request Conflict:One dev submits a pull request to merge their changes into the main branch, but another dev has already made changes to the same code. The pull request cannot be automatically merged and needs manual intervention. | * review the changes in the pull request and identify the conflicts * create a new branch from the main branch * **git fetch** - to get the latest changes from the remote repository * **git checkout <new\_branch>** - to switch to the new branch * **git merge <original\_branch>** - to merge the changes from the original branch into the new branch * resolve the conflicts manually by editing the conflicting files * **git add <conflicting\_file>** - to stage the resolved files * **git commit** - to commit the changes with a message explaining the resolution * **git push** - to push the changes to the remote repository * create a new pull request from the new branch |