Branching Strategies in Git:

- To manage the process of writing, merging, and deploying code with the help of VCS\
- Keep the Prj repo – organized, error-free and avoid merge-conflicts(many users push and pull code at same time from same repo)

1.) Creation of a branch:

➢ Commands used:

git branch <branch__name>

git checkout <branch_name>(To move to that particular branch from curr)

git checkout -b <branch_name>(create & move at same time)

git branch(list all local branch , current branch with *)

git branch -r(remote branches)

git branch -a(local+ remote branches)

git branch -d <bname>(Dlt branch only if merged already)

git branch -D <bname>(Force dlt even if not merged)

git branch -m <oldname> <newname> (renaming)

git branch -v(last commit)

git branch –merged(List the merged branches)

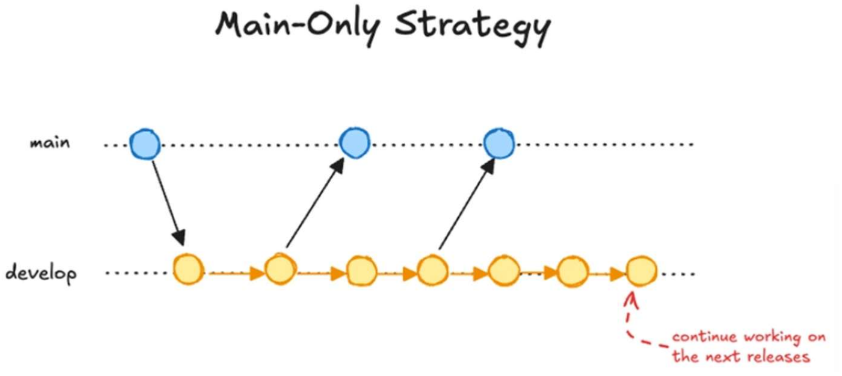
git branch –no-merged(list branches that are not merged yet)

2.) Merging (Overview of it in general):

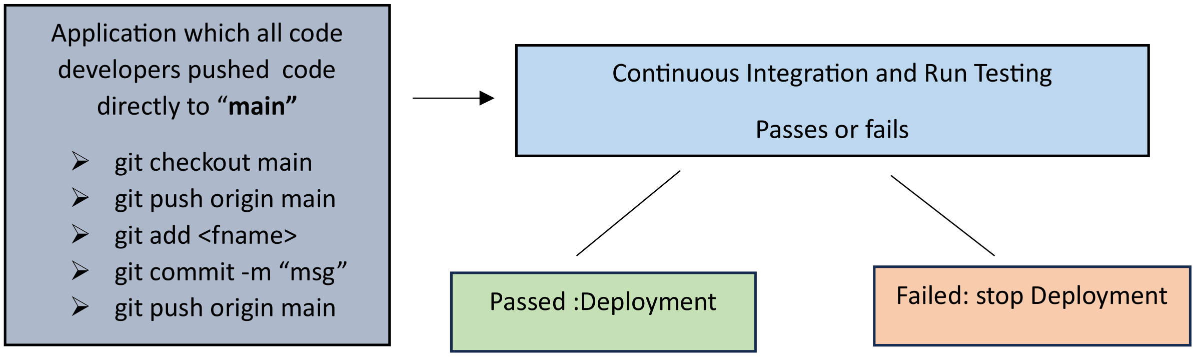
I have a prj with “main” branch(with commit-A,B,C) , and now I plan with add a new branch-“Feature”(have commit-D,E) to it , in-order to combine the work from 1 branch into another branch, I use “**Merging”**

➢ git merge <branch_name>

3.) Main-Only Stragety:



- All Development happens on single branch – main or master



Application which all code developers pushed code directly to "**main**"

- git checkout main
- git push origin main
- git add <fname>
- git commit -m "msg"
- git push origin main

Continuous Integration and Run Testing

Passes or fails

Passed :Deployment

Failed: stop Deployment

- **Regression**: New change that affect the working as like adding a new feature that affect the previous functionality of the application

  git revert <new_commit> (returning to stable previous state)
  git push origin main

- If that bad commit is deployed then we can **"rollback"** to previous deployment

  deploy <prev_version> again

Pros	**Cons**
• No additional branch, No conflict	• Difficult to track process of each features
• Simple , easy to undertand	• Only unit test can be done
• Used when few dev are involved	• Hard to review code

4.) <u>Feature Branching:</u>