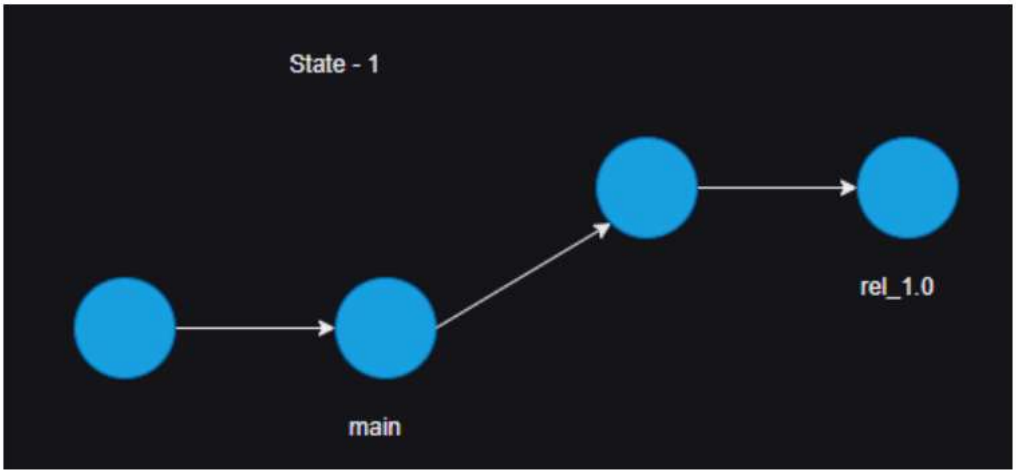
Rewriting History in Git

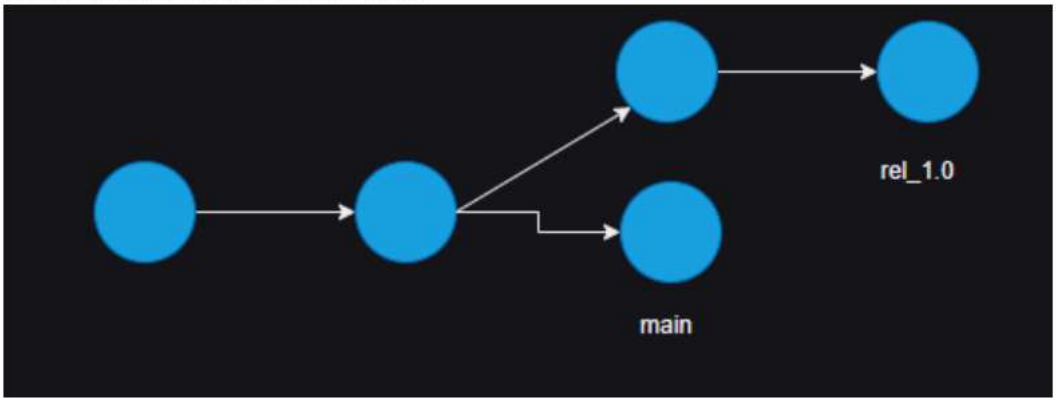
- This involves making changes in commits
- To change the commit message of recent commit use `git commit --amend`
- Also use interactive rebasing Refer Here

Rebasing

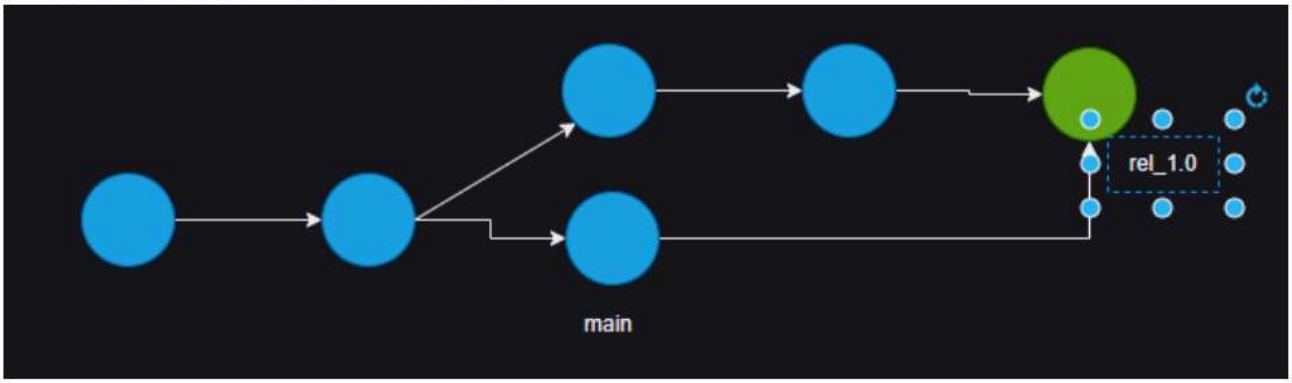
- Refer Here for docs
- Consider the following situation in git history



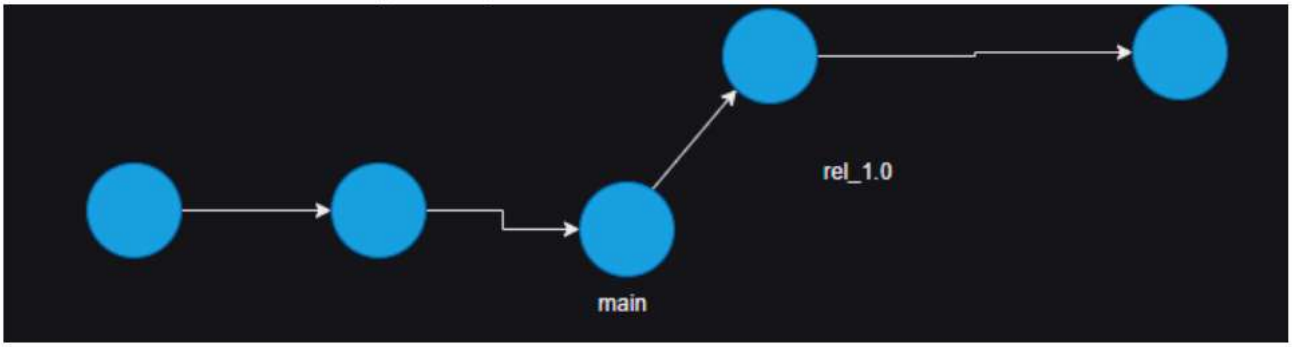
- Now lets add one commit to main



- We want changes of main into rel_1.0
- If we merge changes from main into rel_1.0



- In Git rebase can rewrite history altering commits

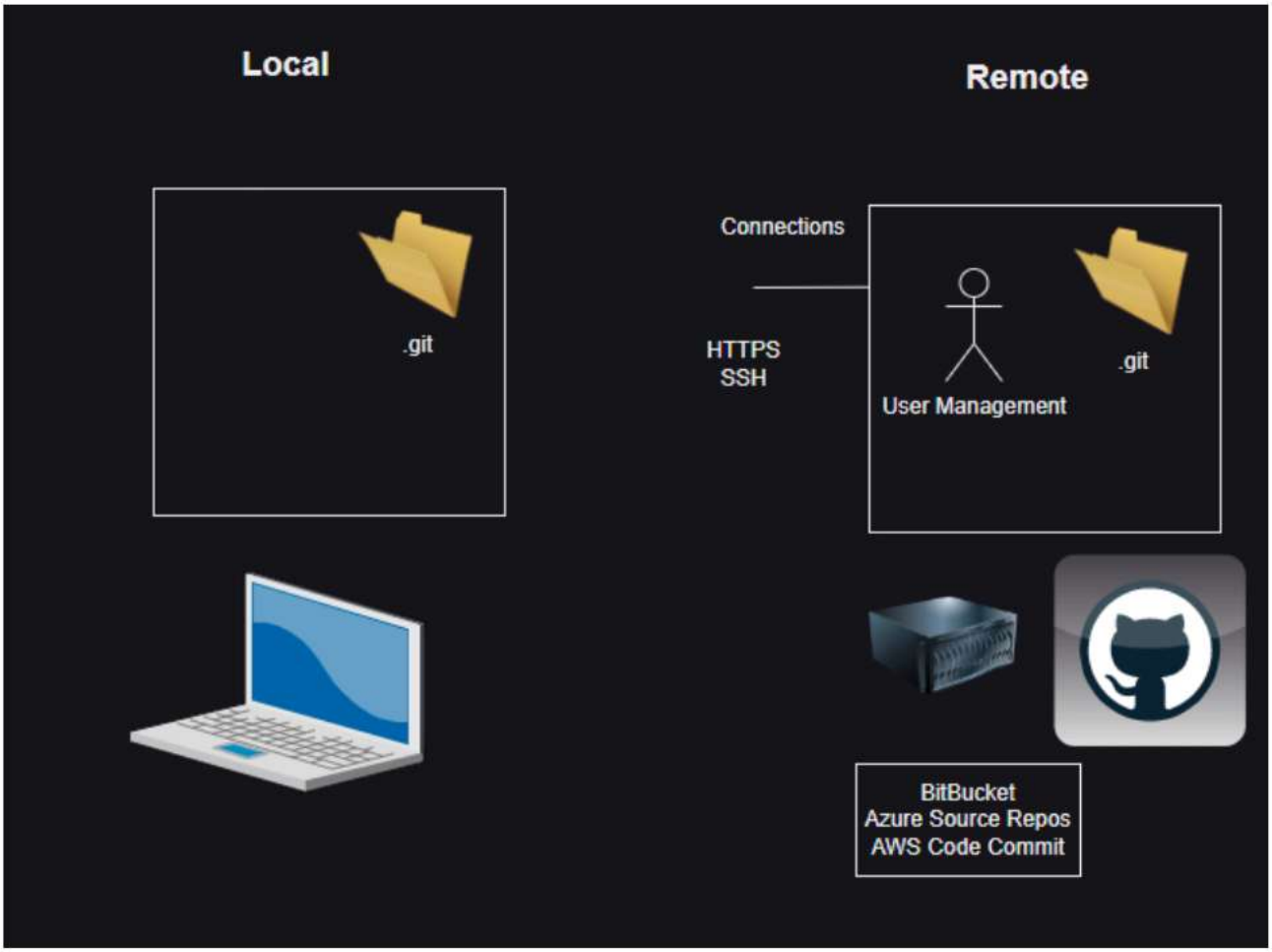


Cherrypicks

- Refer Here for official docs
- Summary:
    - To bring changes from one branch to another we have two options
        - merge
        - rebase
    - To bring individual commits or sequence of commits from other branch into current branch, we can use cherrypick

Remotes

- Overview



- It allows connections from other users
- For authentication we have
    - https (username and password/token)
    - ssh (username and key)
- User management
- A typical user performs 3 major actions
    - Get the repository into local (clone)
    - Once we have repo (local), we
        - submit changes (push)
        - get changes (pull)

Exercise

- Demonstrate
    - merge
    - rebase
    - cherry pick
- Demonstrate
    - three way merge
    - fast forward merge
- Demonstrate rewriting hisory
    - changing commit messages
    - make changes in previous commit
    - combine commits
    - remove commits
- Create a new repo, make 3 or 4 commits. Try removing any commit and then recover the removed commit (Optional)