

Git

Saturday, 10 February 2024

12:20

Creating a (local) git repository

- You can configure git via config
 - Username, email etc
 - E.g. `git config --global user.name "kameswari chebrolu"`
- “init” : Used to create a Git repository
 - `git init`
- After initialization, other files can be added

git status

- Tells current state of the repository

git add

- Add files to staging area
 - `git add file1.txt file2.txt`

Git log

`git log`

`git log file1.txt` (commit history of that file)

- A long hexadecimal number you see is the commit's hash, helps identify a commit
 - can use just 5 digits mostly in commands

git show

git show :filename

Example: git show :file1.txt

Shows the content of file1.txt in the staging area

git show commit:filename

Example: git show HEAD:file1.txt

Shows the content of file1.txt in HEAD

Example: git show 5b80ea8:file1.txt

Shows the content of file1.txt in the commit object 5b80ea8

git diff


git diff <commit>: shows the diff between the current working tree and the <commit>

git diff --cached <commit>: shows the diff between your staged changes and the <commit>

Undoing Changes: checkout and commit

- You can move backwards in time by checking out an older commit.
 - git checkout commit-id
 - Will replace the contents of working directory by the contents of that older commit
 - Useful for “look but don't touch” way to explore the older code
 - Get back to most recent commit via git checkout master
- Ability to rollback individual files to old versions: git checkout commit-id path-to-a-file
 - Then can use git commit if you want everything else to use current and this file to be some older version

bash

 Copy code

```
git checkout <commit-id> -- <path-to-file>
```

Replace `**<commit-id>**` with the commit identifier (SHA-1 hash) of the commit you want to check out and `**<path-to-file>**` with the relative path to the file you want to retrieve.

For example, let's say you want to retrieve the version of the file `**example.txt**` as it existed in commit `**abc123**`:

bash

 Copy code

```
git checkout abc123 -- example.txt
```

git branch/switch

- git branch: List the branches
- git switch -c testing: create a new branch
 - “testing” is the name of this new branch

git checkout master (ensure you are in master branch; you want to merge testing into this)

git merge -m "merging" testing (merge testing into master)

Often this may result in a conflict, which you need to resolve.

- After you resolve, you need to add and commit the files with conflict into master

Note testing still exists and not affected by merge

- git checkout testing