

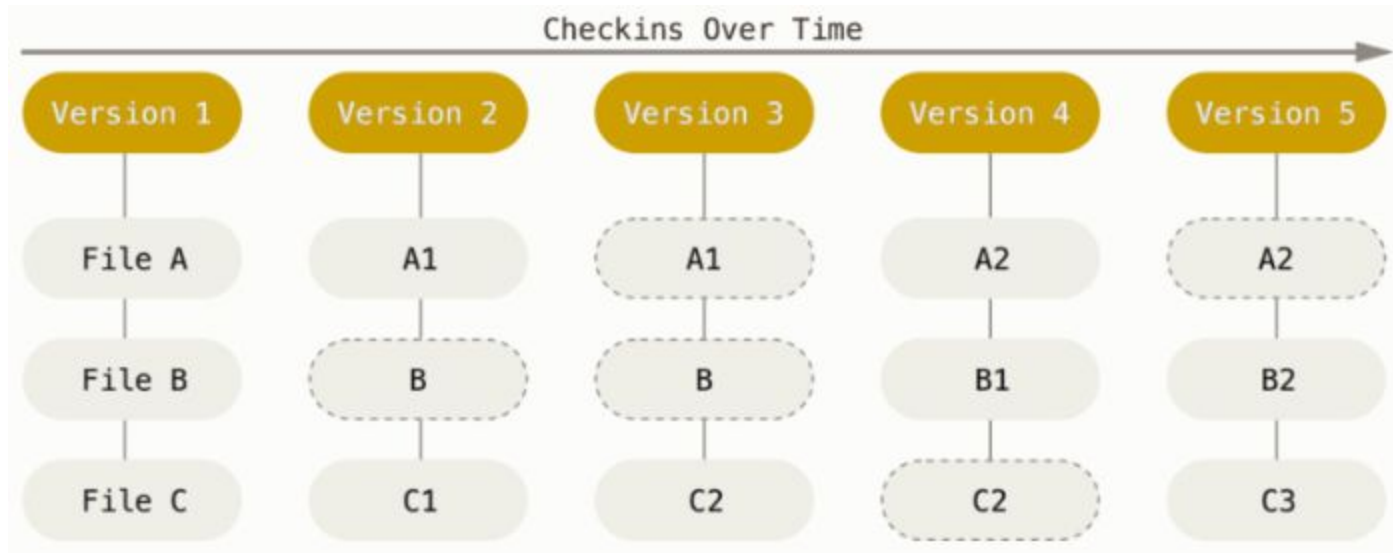
# **Introduction to Version Control with Git**

# Git

- Version control
  - Keeps every version
  - Compare versions
  - Restore version
- Distributed
- Non linear
  - Supports branching

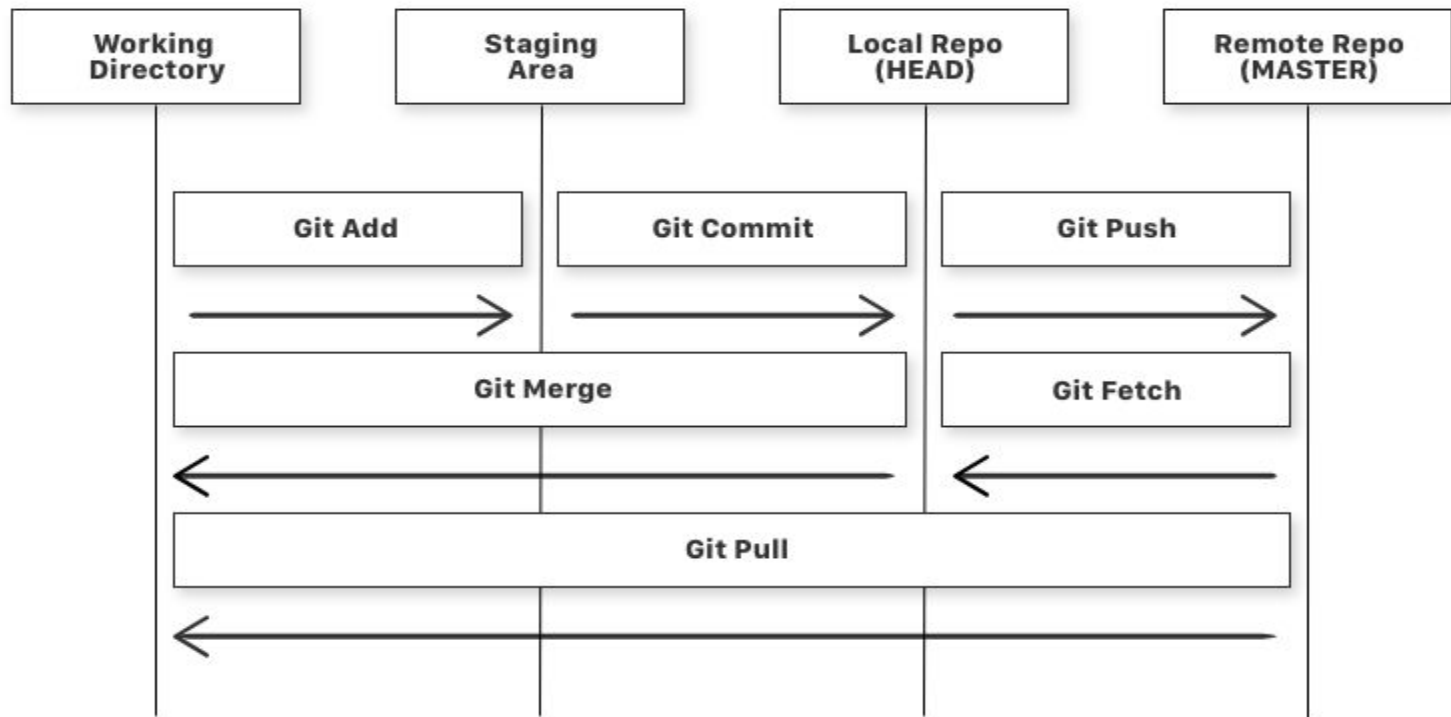
# Git

- Snapshots
- Local operations
- Integrity checksum
  - SHA-1 hash
  - 40 character hash



# Git - States

- File states
  - Modified
  - Staged
  - Committed
- Working directory
- Staging area
- Repository (.git directory)
- Workflow
  - Working directory => Staging area => .git directory



**The command line ...**

**This course is Windows friendly!**



# Git - Installation

- Windows
  - <https://git-scm.com/download/win> (32bit/64bit - any)
- Linux
  - Fedora/CentOs
    - `sudo dnf install git-all`
  - Debian
    - `sudo apt install git-all`
  - <https://git-scm.com/download/linux>

## Git - Configuration

- `git config`
- System wide
  - `git config --system`
- User wide
  - `git config --global`
- Repository
  - `git config --local`
- Each level overrides the previous
- View configuration
  - `git config --list --show-origin`

## Git - Identity Setup

- Git identity baked into commits
  - `git config --global user.name "Devdutt M R"`
  - `git config --global user.email devdinator@gmail.com`

## Git - Editor Setup

- Set your editor preferences
  - `git config --global core.editor 'path_to_your_favorite_editor'`
  - `git config --global core.editor 'C:/Program Files/Notepad++/notepad++.exe -multiInst -notabbar -nosession -noPlugin'`
- Not setting editor can have side-effects

## Git - Default branch

- `master` (most flavors, including atlassian)
- `main` (github)

## Git - Help

- Verb documentation
  - `git help <verb>`
  - `git <verb> --help`
- Concise documentation
  - `git <verb> -h`

**Hands-on!**

**GitHub**

<https://github.com/>



## GitHub - Sign up

- Use NIE account (No personal account)

## Git - Repository

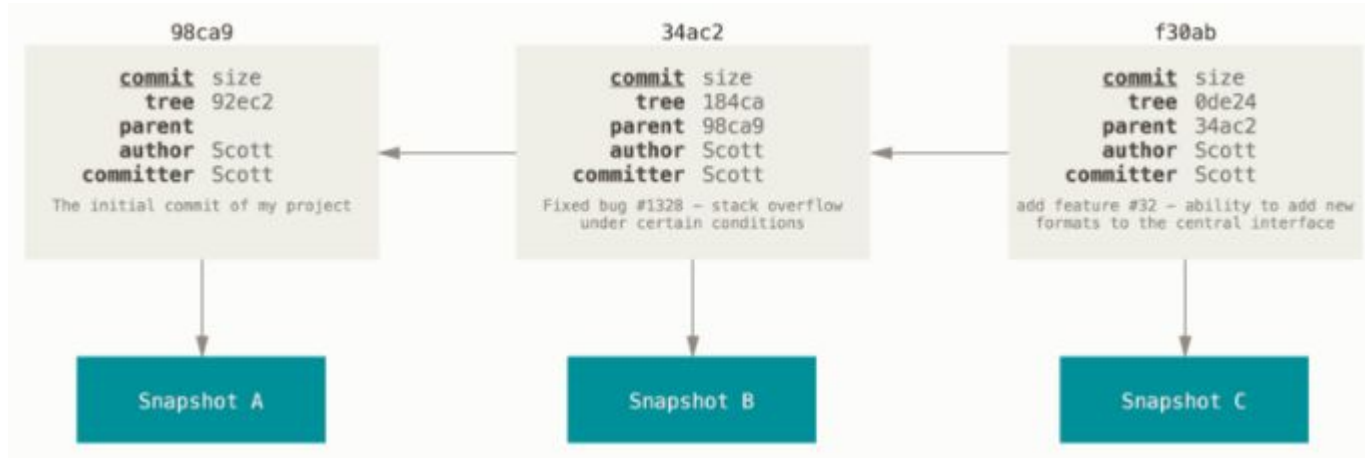
- New repository
  - Navigate to desired directory
  - `git init`
- Clone repository
  - `git clone <repo-clone-url>`
  - `git clone https://github.com/devdinator/dev-ops.git`
- Clone with directory name
  - `git clone <repo-clone-url> <directory-name>`
- Use `https` protocol

# Git - Commit

- Object
  - Pointer to the snapshot
  - Author's name
  - Email address
  - Commit message
  - ...
- Checksum

# Git - Commit

- Commit tree



## Git - Commit

- Make some changes
- Track the changes
  - `git add <pattern>`
  - `git add ClassFile.cs`
- Commit the changes
  - `git commit`
  - Add commit message in editor
  - Save and close

## Git - Commit

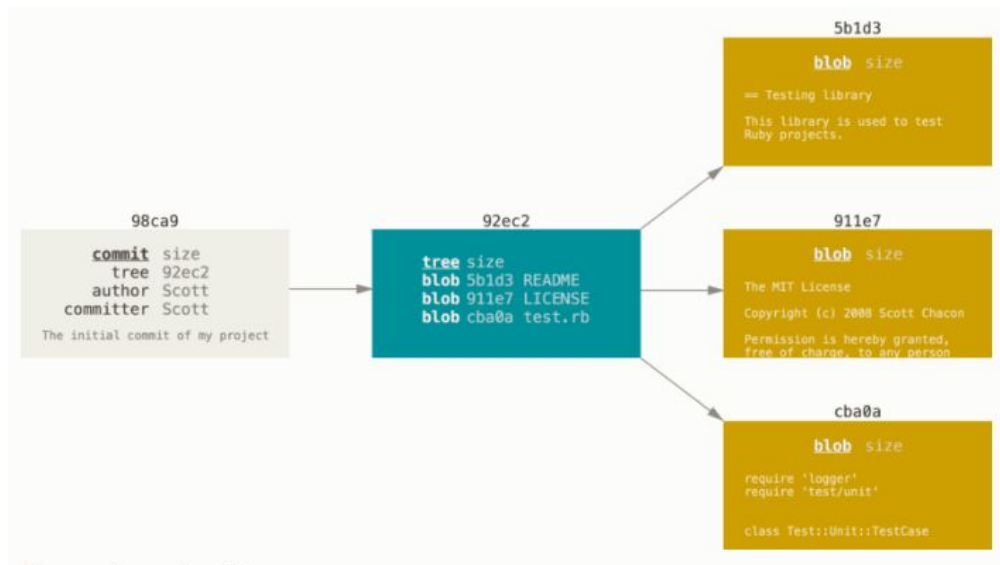
- From IDE
- Microsoft and JetBrains IDEs support git
  - Visual Studio
  - VS Code
  - Rider
  - IntelliJ
  - ...

## Git - Commit

- Stage
  - `git add <regex>`
- Commit
  - `git commit`
  - If errors out, ensure editor is correctly set

## Git - Directory

- 3 files in a directory
- Staging calculates checksum



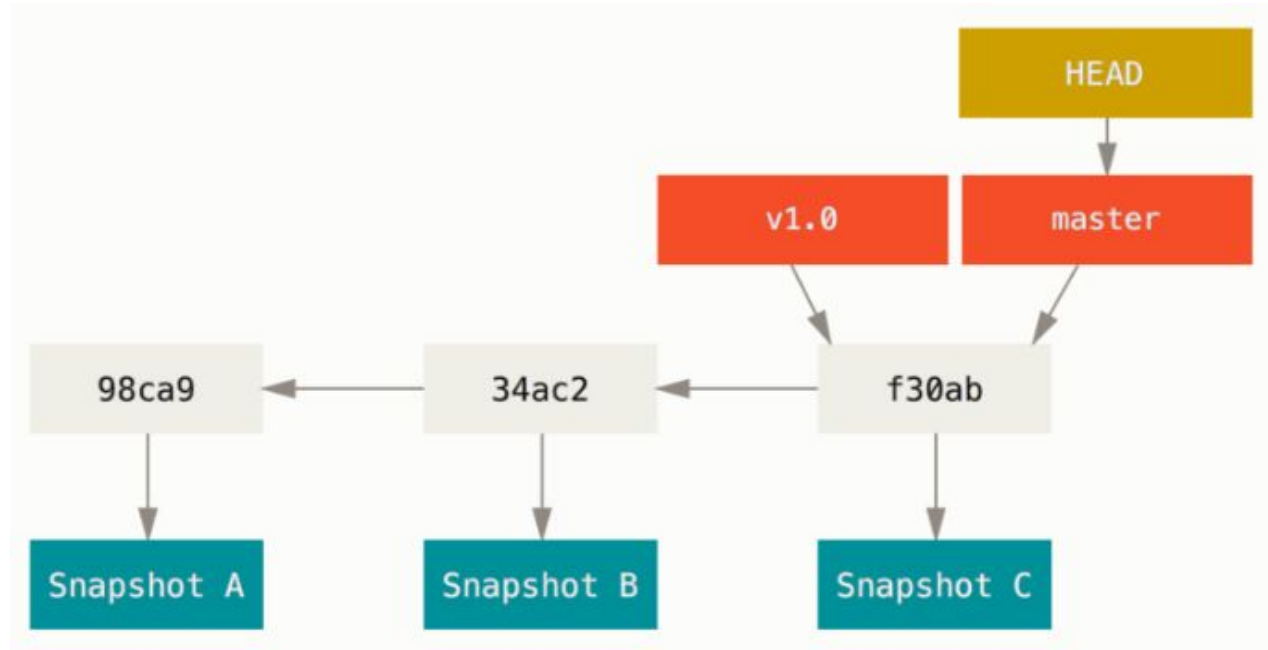


## Git - Branching

- Diverge from main/master
- Not messing main/master
- **Killer feature**
- Lightweight and instantaneous
- Switching branches is fast

## Git - Branching

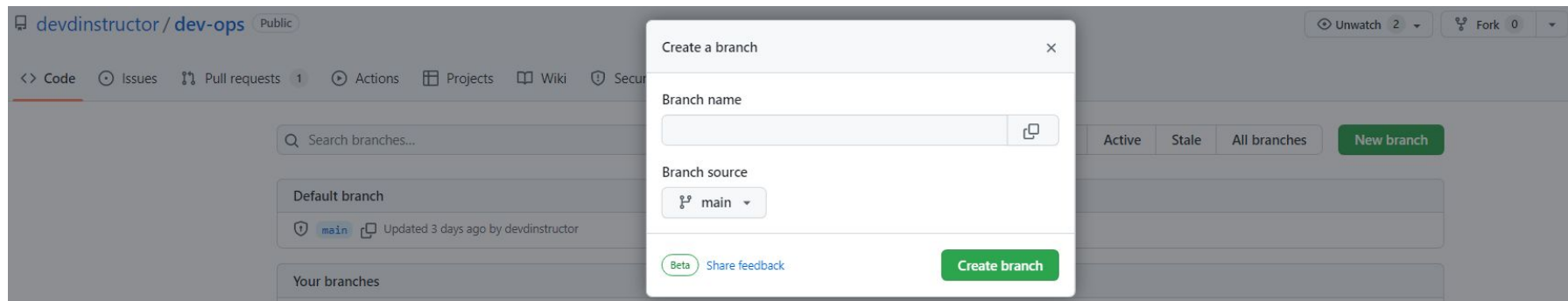
- Pointer to a commit



## Git - Branching

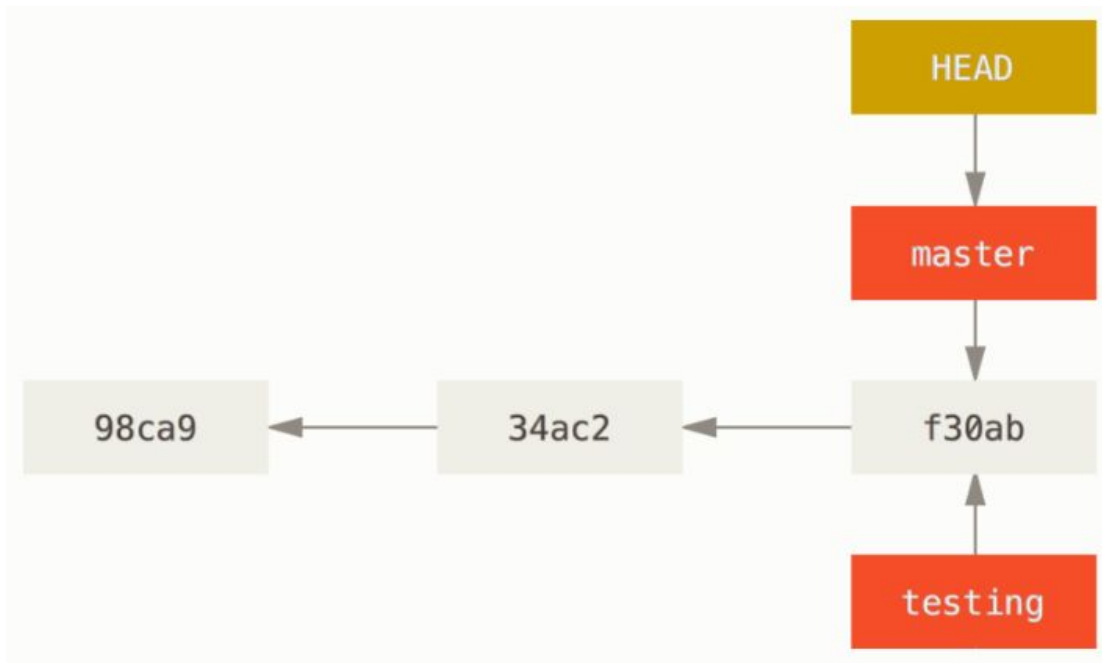
- Go to repository
- Select **Branches** submenu under **Code** menu
- Select **New Branch**
- Name your branch
- Select from **main**
- Create branch

# Git - Branching



## Git - Branching

- HEAD points to the current branch

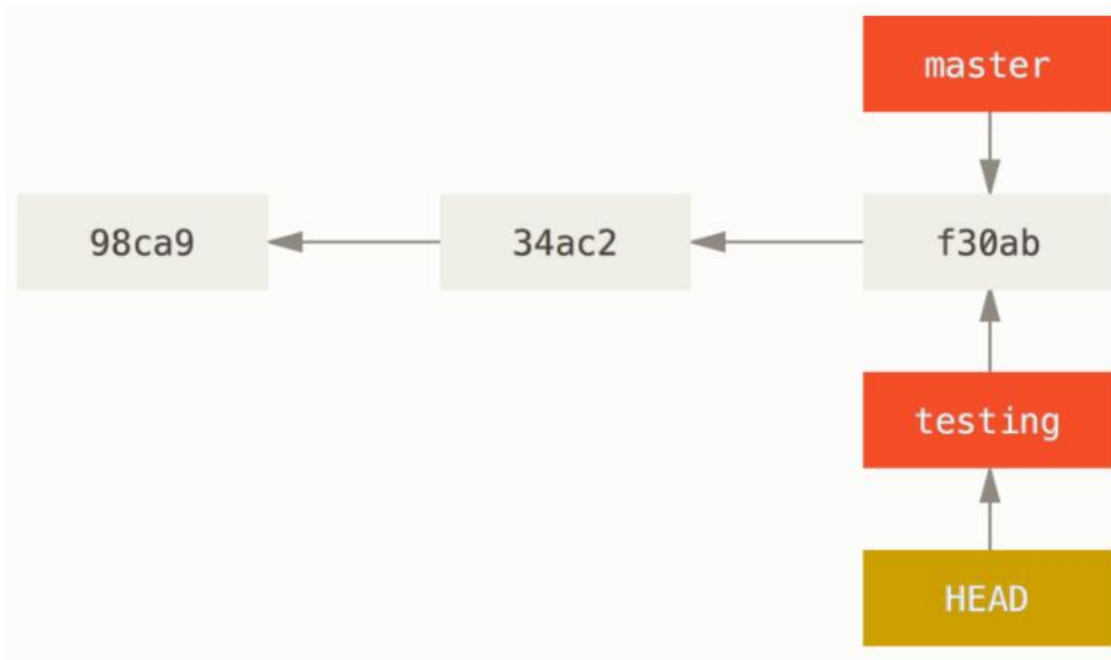


## Git - Branching

- Sync metadata to local
  - `git fetch --prune`
- Switch branches
  - `git checkout <branch-name>`

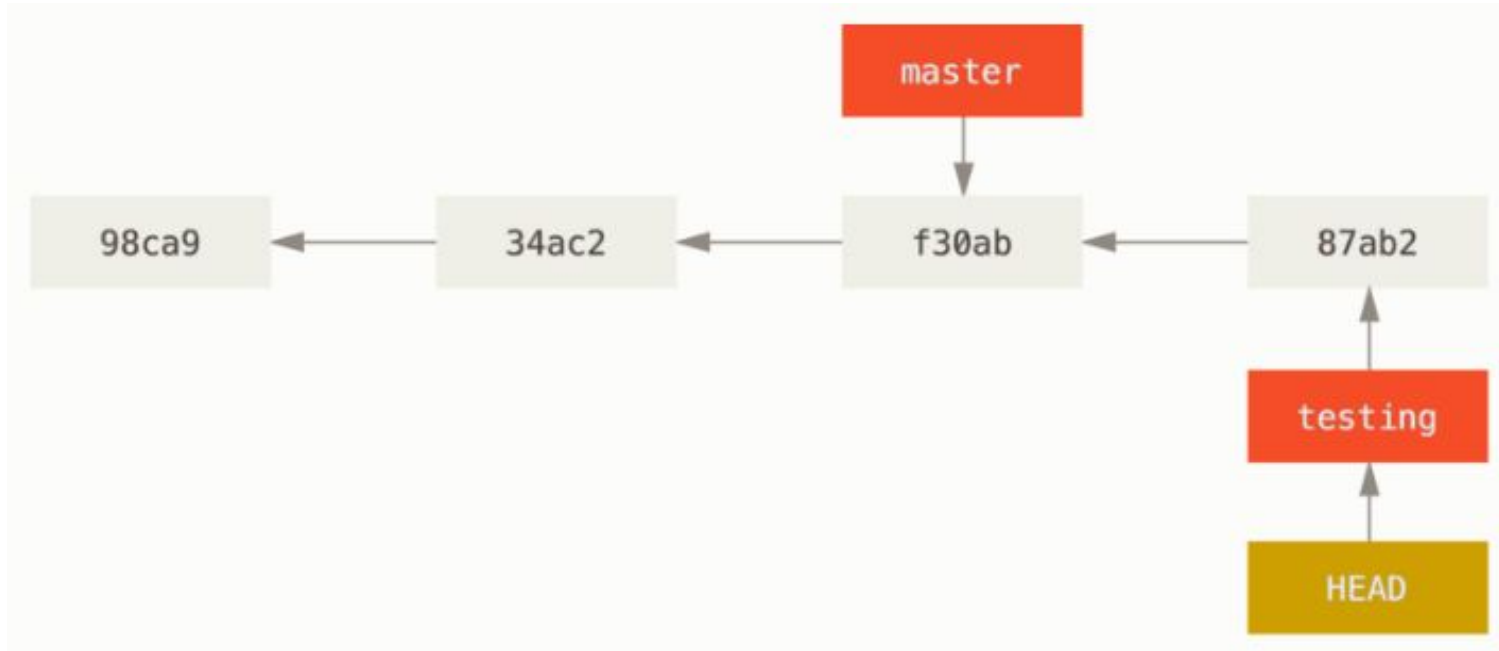
## Git - Branching

- HEAD points to the new branch



## Git - Branching

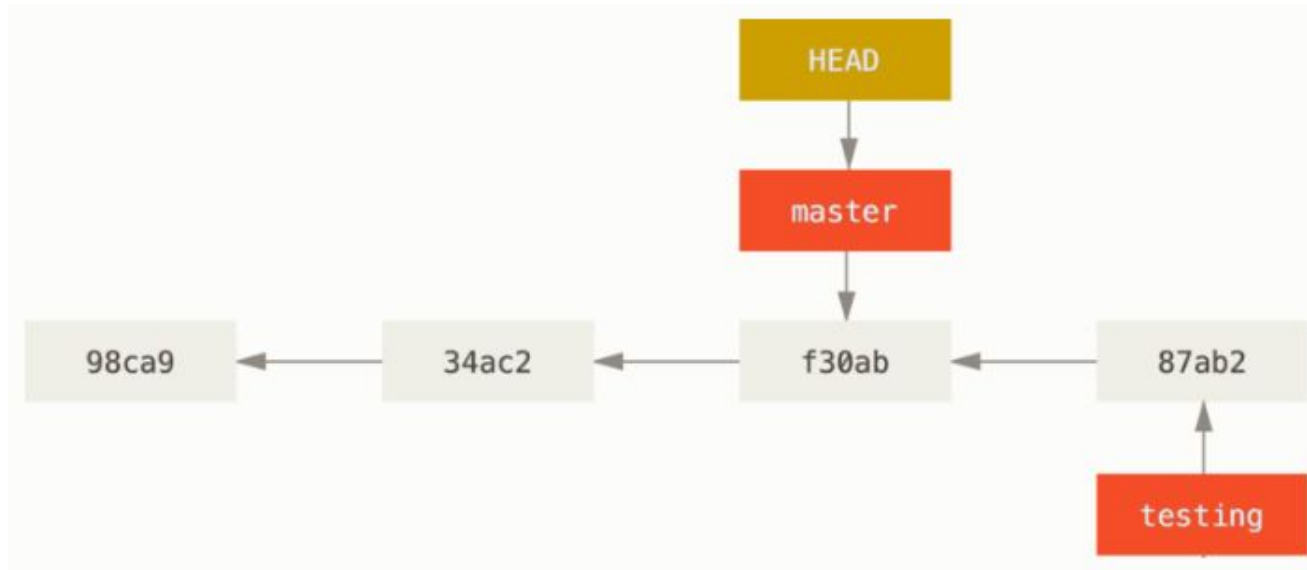
- On subsequent commit





## Git - Branching

- Switching back to master
- Reverts changes



# Git - History

- View history

- `git log`

```
D:\Projects\nie\dev-ops>git log
commit af166177060cae896db72ae92f071a1f2e7a5507 (HEAD -> feature/add-documentation-file, origin/feature/add-documentation-file)
Author: Devdutt Rajaram <[REDACTED]@[REDACTED].com>
Date:   Fri Oct 21 23:18:03 2022 +0530

    Adds readme file.

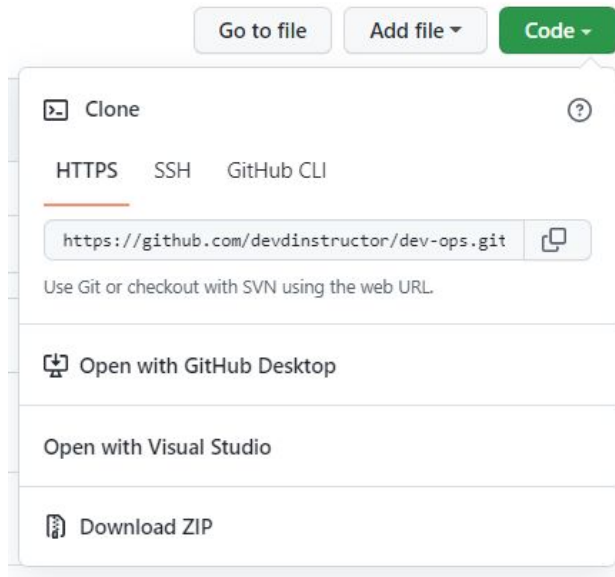
commit ef67546e7c6d4ee6a27ef9343099e156f6c83e99 (origin/master, origin/HEAD, master)
Author: Devdutt M R <devdinstructor@gmail.com>
Date:   Fri Oct 21 16:40:12 2022 +0000

    Initial commit
```

**Let's submit your assignment**

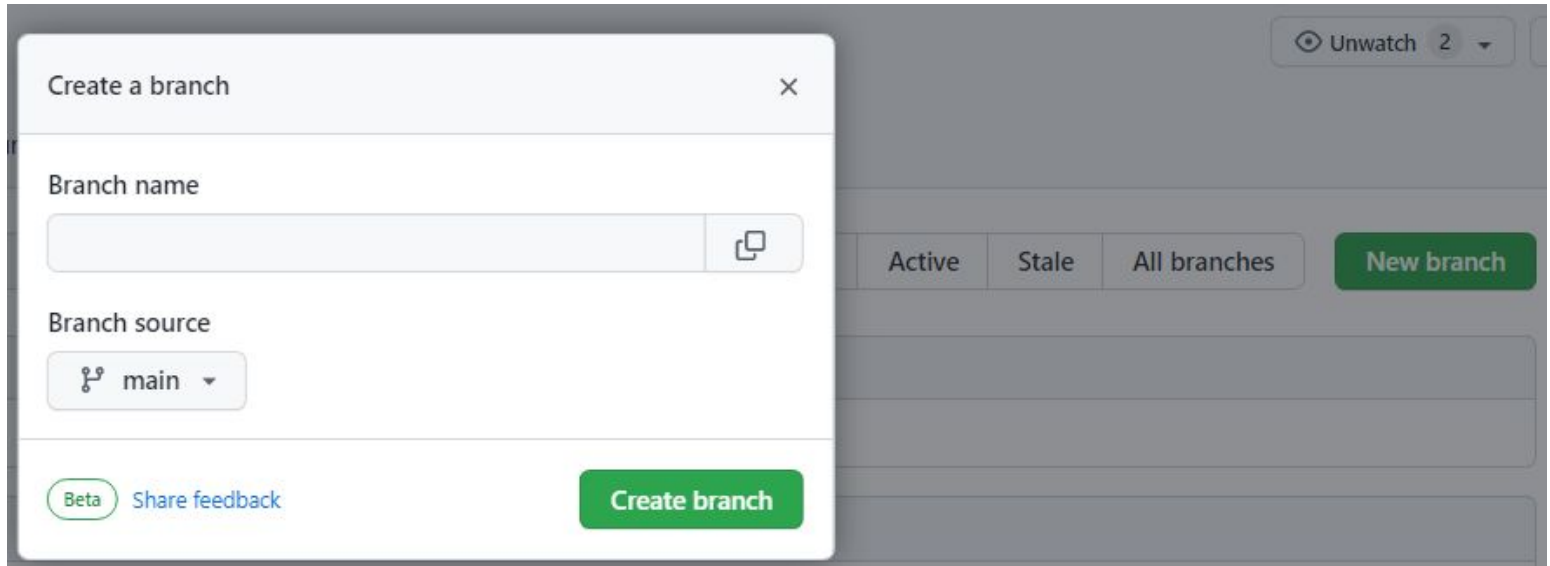
# Assignment submission - Cloning

- Clone the repo
  - Code tab => Code dropdown



## Assignment submission - Branching

- Create a branch
  - [Code](#) tab => [Branches](#) link => [New branch](#) button



**The previous operation was on remote**

## Assignment submission - Branching

- List all local branches
  - `git branch`
- List all branches
  - `git branch -a`

## Assignment submission - Branching

- Sync remote metadata to local
  - `git fetch`



## Assignment submission - Branching

- Checkout the new branch
  - `git checkout <your-created-branch>`

**Wait!**

## **Ground rules**

## Assignment submission - Ground rules

- Refer the pull request
  - Pull requests tab => <the open pull request> => Files changed tab



## Assignment submission - Making changes

- Create your assignment directory
  - Under assignments directory
  - Directory name must match USN
    - E.g., `4NI19IS007_A`
- Add the assignment file to the directory

## Assignment submission- Staging changes

- Stage your changes

- `git add *`

## Assignment submission - Committing changes

- Commit your changes
  - `git commit`
  - Provide a sensible commit gist
  - If errored out, go back and set your git editor

**How did it go?**



# Assignment submission - History

- View history

- `git log`

```
D:\Projects\nie\dev-ops>git log
commit af166177060cae896db72ae92f071a1f2e7a5507 (HEAD -> feature/add-documentation-file, origin/feature/add-documentation-file)
Author: Devdutt Rajaram <[REDACTED]@[REDACTED].com>
Date:   Fri Oct 21 23:18:03 2022 +0530

    Adds readme file.

commit ef67546e7c6d4ee6a27ef9343099e156f6c83e99 (origin/master, origin/HEAD, master)
Author: Devdutt M R <devdinstructor@gmail.com>
Date:   Fri Oct 21 16:40:12 2022 +0000

    Initial commit
```

**Remember, most git operations are local**

**So was your commit**

**Time to go global!**

## Assignment submission - Going global


- Push your changes to the remote
  - `git push`
- You should see your changes in github



## Assignment submission - Creating pull request


- Create new pull request
  - `Pull requests` tab => `New pull request` button
  - Or, the GitHub hint
- To main branch
  - `base: main`
- From your branch
  - `compare: <select-your-branch>`




# Assignment submission - Creating pull request


- Review the commit, create pull request


 base: main ← compare: create-assignments-directory ✓ **Able to merge.** These branches can be automatically merged.



 **Defines assignment submission rules. #1**  
No description available 

Create another pull request to discuss and review the changes again. [Learn about pull requests](#) 

 1 commit  2 files changed  1 contributor

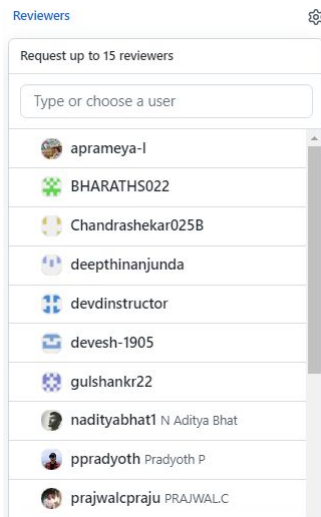
 Commits on Oct 27, 2022

Defines assignment submission rules.  
 devdutt91 committed 23 hours ago

 c24b1b5 

# Assignment submission - Creating pull request

- Add reviewers
  - Pull requests tab => Reviewers link => Gear icon
  - Add devdinstrutor and rakshitham as reviewers





## Issues

- Push your changes to the remote
  - `git push`
- You should see your changes in github

## Branching - Advanced

- List branches
  - `git branch`  
  
    `branch-1`  
    \* `master`  
    `branch-2`

## Branching - Advanced

- List branches with commits

- `git branch -v`

```
branch-1 93bfv11 <message>
```

```
* master    a9s8yas <message>
```

```
branch-2 as9874d <message>
```

## Branching - Advanced

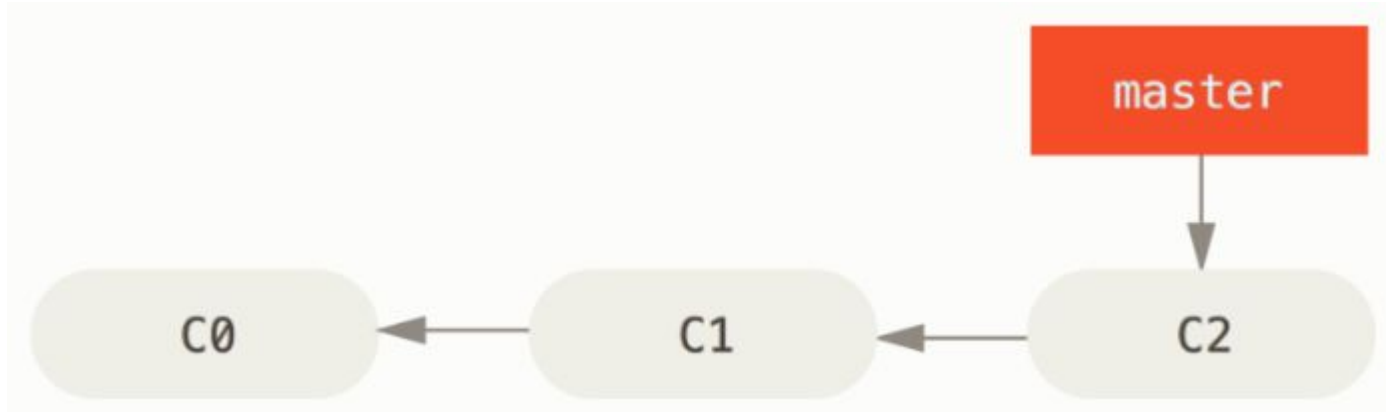
- Renaming branch locally
  - `git branch -move <old-branch-name> <new-branch-name>`
- Pushing to server
  - `git push --set-upstream <new-server-branch-name>`

## Branching - Merge

- Merge commits from another branch into current branch

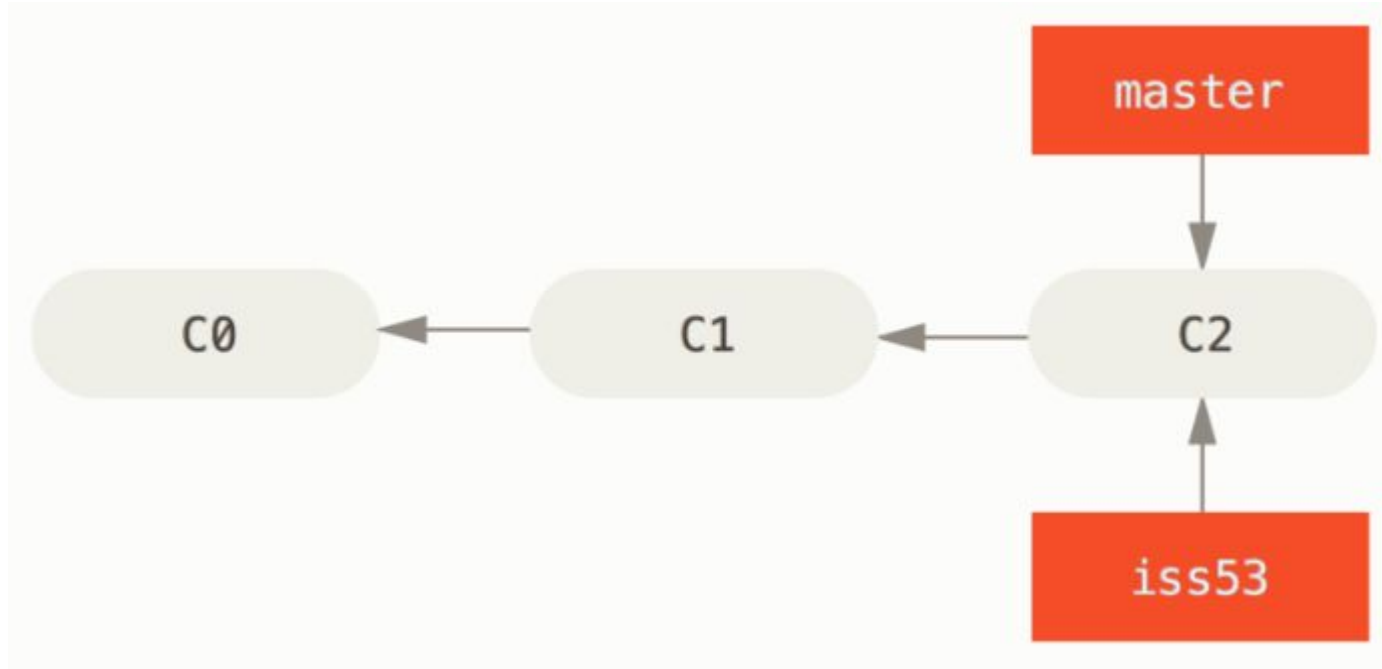
## Branching - Merge

- Clean state



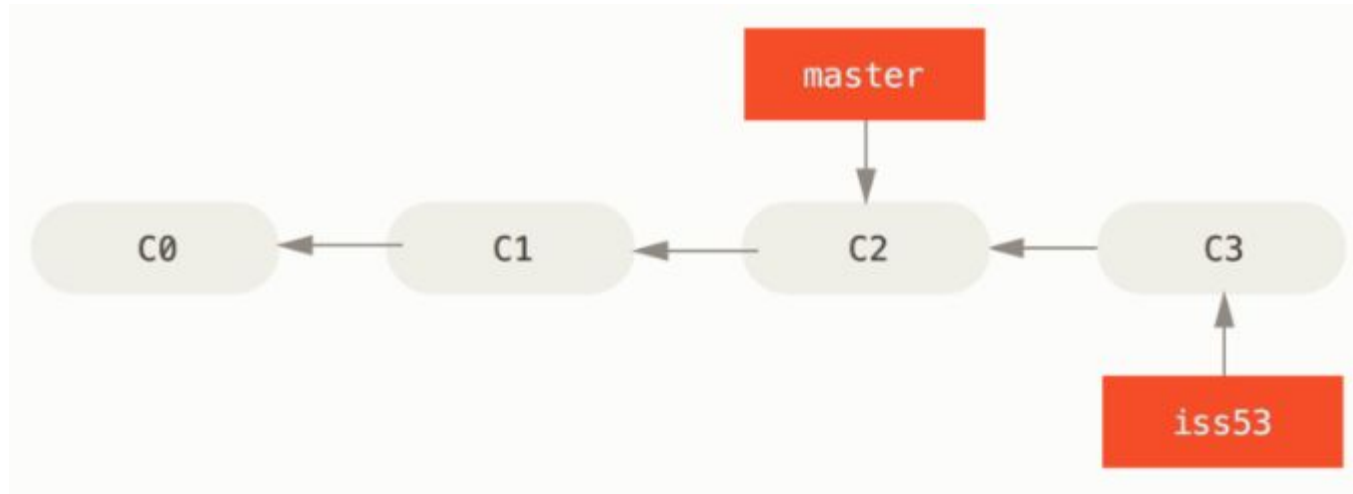
## Branching - Merge

- Create a new branch (iss53)



## Branching - Merge

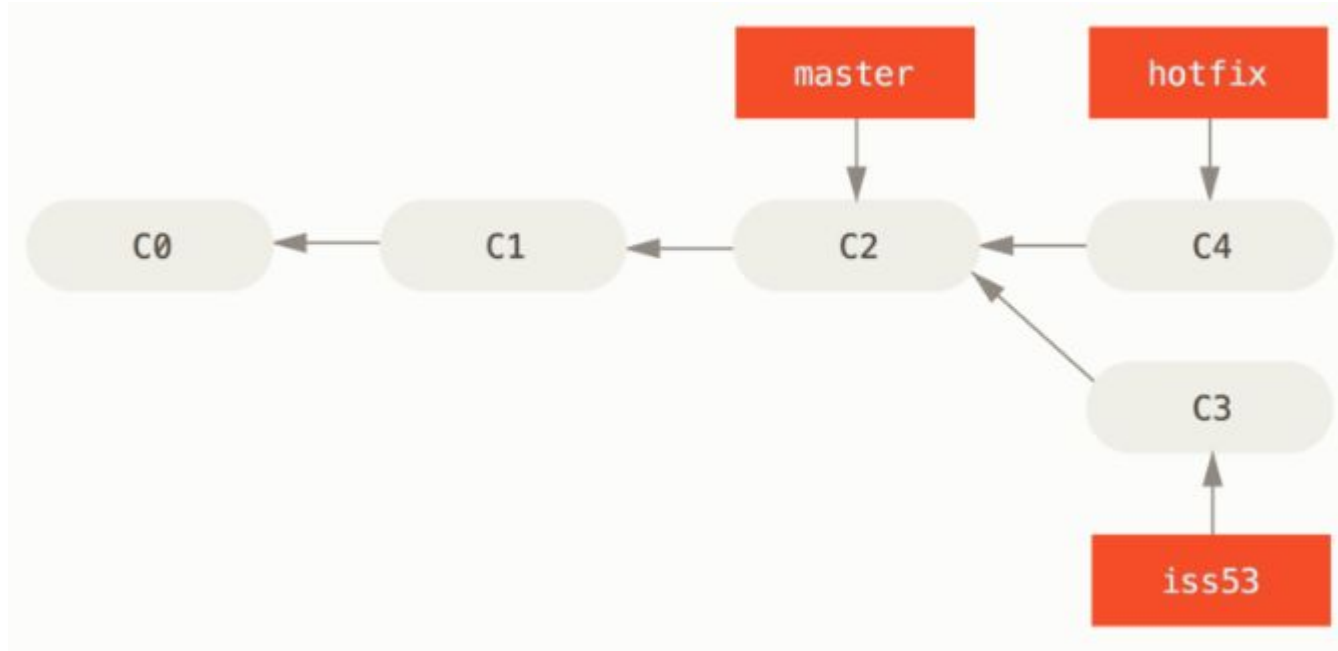
- Work on the new branch (iss53)





## Branching - Merge

- Create another branch (hotfix) from master, work on it



## Branching - Merge

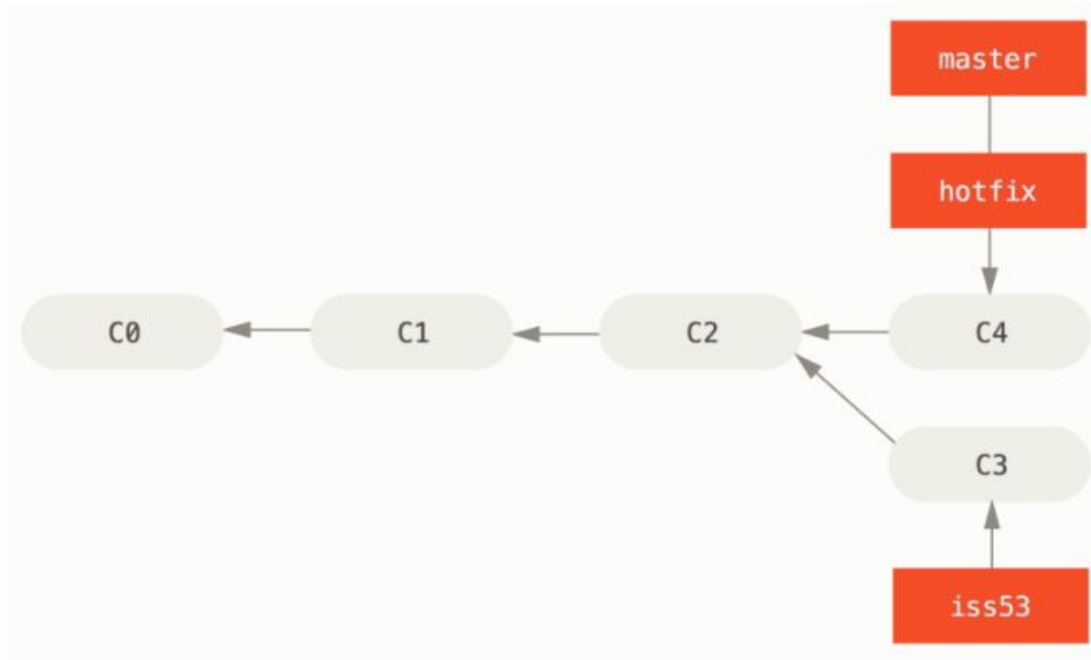
- Merge another branch (hotfix) into current branch (master)
- Checkout the branch to merge into
  - `git checkout <branch-to-merge-into>`
- Merge the other branch
  - `git merge <branch-to-merge>`

## Branching - Merge

- Checkout the branch to merge into (master)
  - `git checkout master`
- Merge the other branch (hotfix)
  - `git merge hotfix`

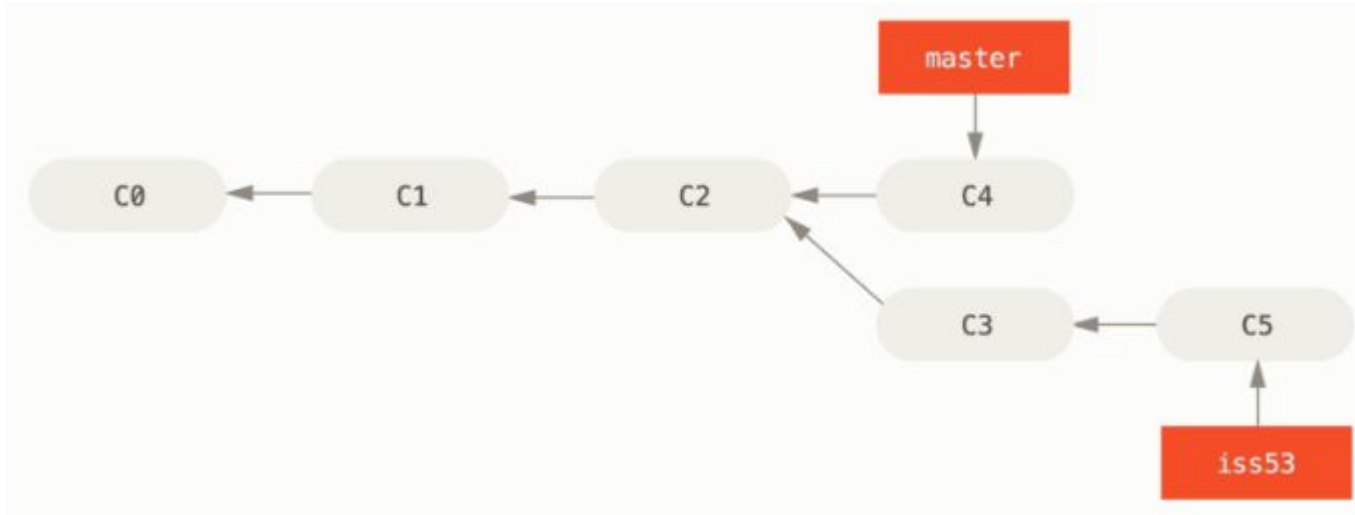
## Branching - Merge

- Merge (fast forward) another branch (hotfix) into current branch (master)



## Branching - Merge

- More work on the other branch (iss53)

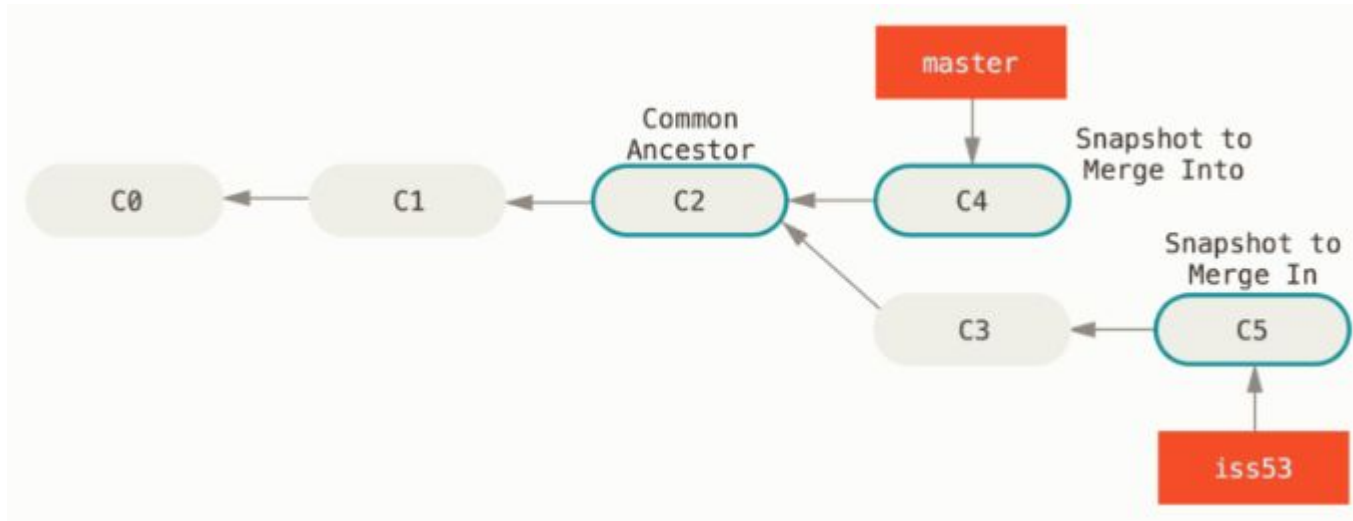


## Branching - Merge

- Merge another branch (iss53) into current branch (master)
  - `git checkout master`
  - `git merge iss53`

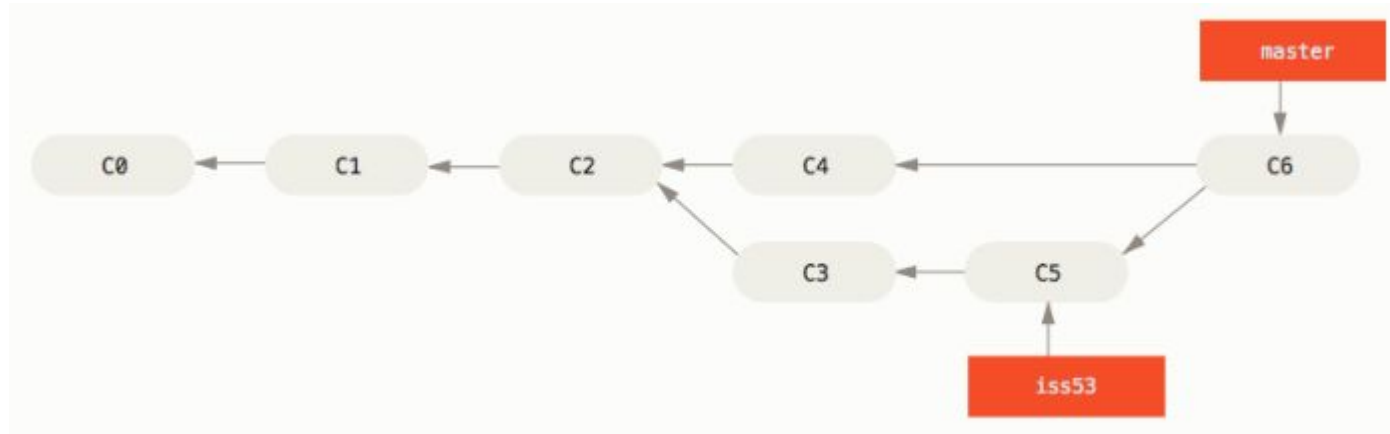
## Branching - Merge

- Merge another branch (iss53) into current branch (master)



## Branching - Merge

- 3-way merge commit



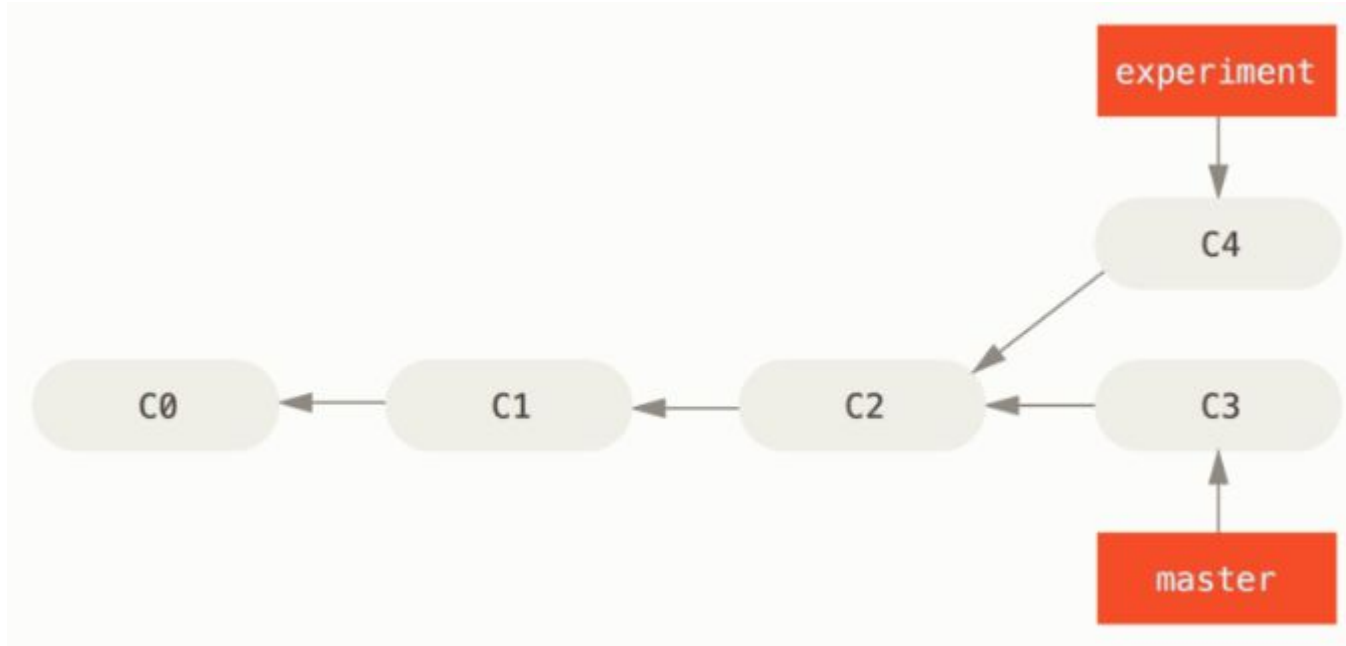


## Branching - Rebase

- Replay commits from another branch onto the current branch

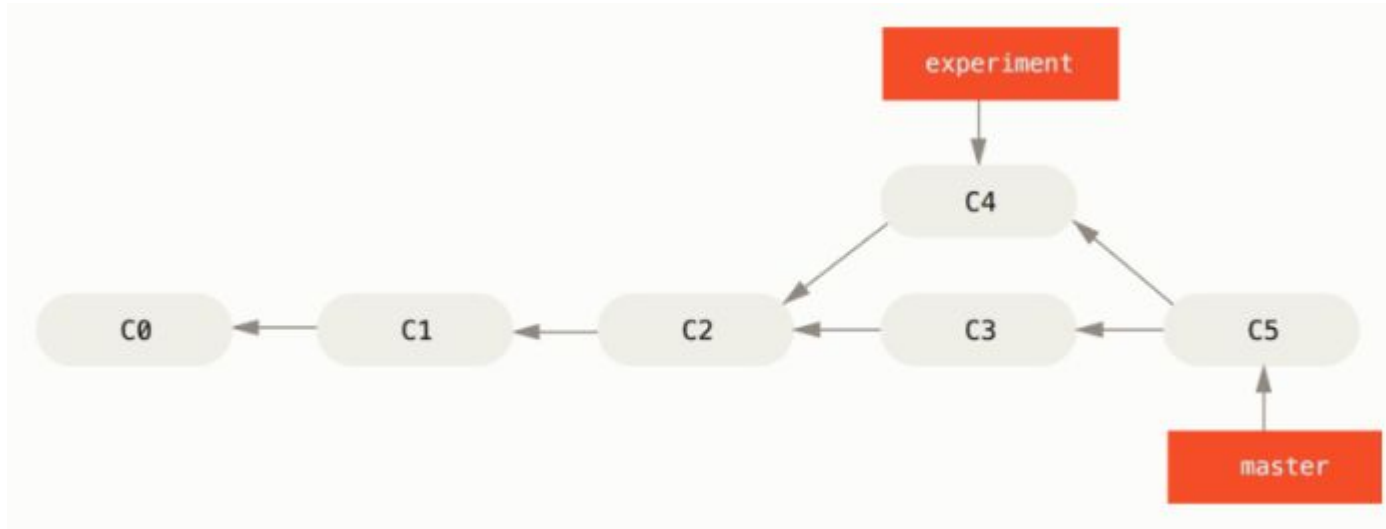
## Branching - Rebase

- Clean state



## Branching - Rebase

- Merge would have created this



## Branching - Rebase

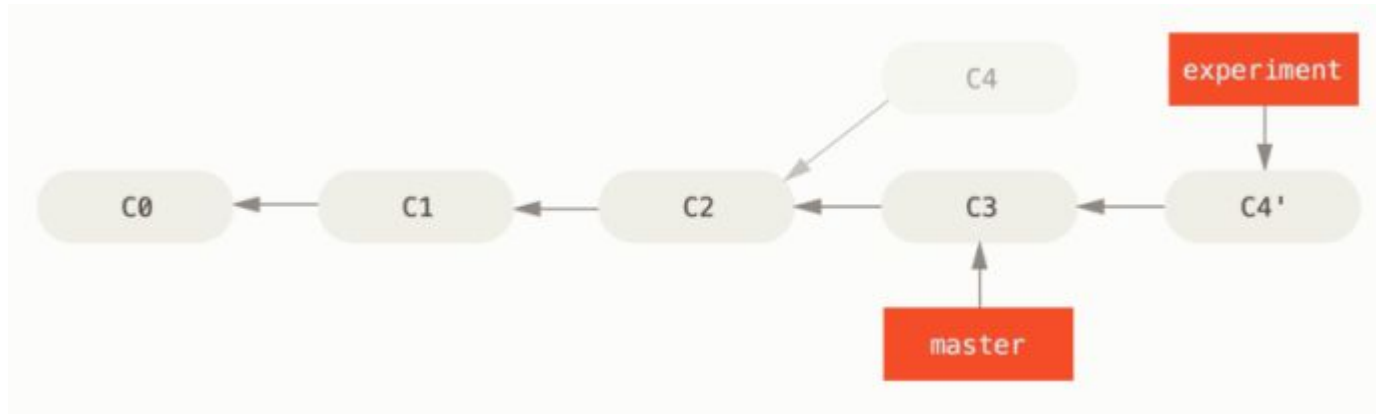
- Rebase current branch (experiment) from another branch (master)
- Checkout branch to rebase from
  - `git checkout <branch-to-rebase>`
- Rebase current branch from another branch
  - `git rebase <branch-to-rebase-from>`

## Branching - Rebase

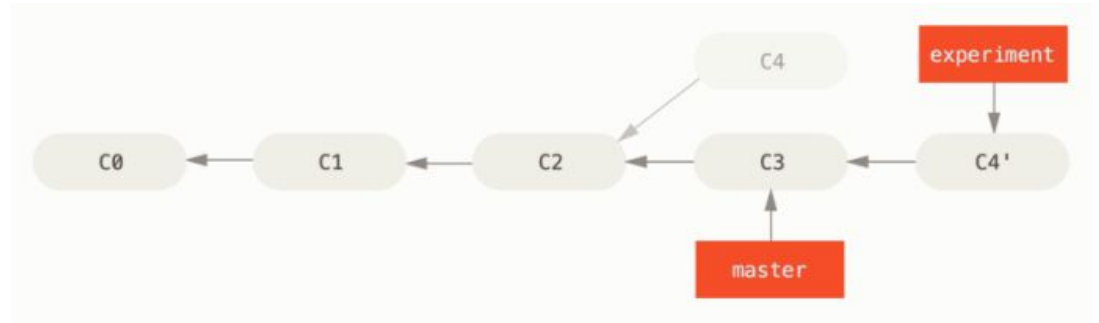
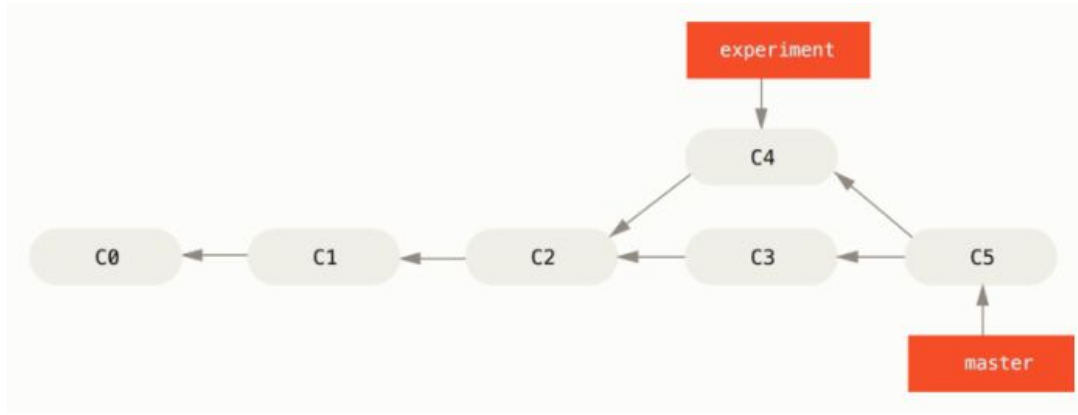
- Checkout branch to rebase from (experiment)
  - `git checkout experiment`
- Rebase current branch from another branch (master)
  - `git rebase master`

## Branching - Rebase

- Rebase current branch (experiment) from another branch (master)



## Branching - Merge vs. Rebase



## Branching - Merge vs. Rebase

- The two snapshot are the same
- Rebase maintains a cleaner linear history



## Branching - Resolving conflicts

- Conflict
- Same file modified differently in branches
- Git unable to create a merge commit automatically
- Arises on a PR, during merge or rebase

## Branching - Resolving conflicts

- Conflict
- Same file modified differently in branches
- Git unable to create a merge commit automatically
- Arises on a PR, during merge or rebase

## Branching - Resolving conflicts

```
13  <<<<<< update-program-file
14      if (arr[idx] == num)
15      {
16          pos = idx;
17          break;
18      }
19  =====
20      if (arr[idx] != num)
21      {
22          continue;
23      }
24
25      pos = idx;
26      break;
27  >>>>>> main
```

## Branching - Resolving conflicts

- Resolve by manually modifying the file
- Any file having conflicts must be staged and committed
- Staging the file marks it as resolved
  - `git add <file>`
- Commit as usual
  - `git commit`

## Branching - Resolving conflicts

- Sync with master
  - `git fetch --prune`
- Rebase from master
  - `git rebase origin master`
- Resolve conflicts
- Continue rebase
  - `git rebase -continue`

## Branching - Squash

- Squash last n commits in the current branch
  - `git rebase -i HEAD~n`
- On the editor window
  - `pick 8as81hd`
  - `pick 8as81hd`
  - `pick 09dsahy`

## Branching - Squash

- Squash
  - `pick 8as81hd`
  - `squash 8as81hd`
  - `squash 09dsahy`
- Provide commit message and save

## Branching - Other

- Drop
  - `drop 8as81hd`
  - `pick 8as81hd`
  - `drop 09dsahy`
- Reword
  - `pick 8as81hd`
  - `reword 8as81hd`
  - `reword 09dsahy`