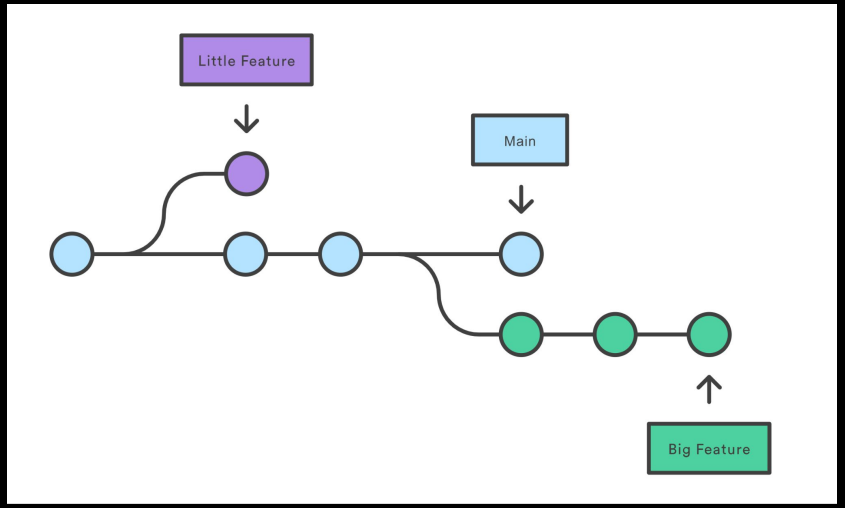
6. Branching and Merging

1. What are Branches
2. Branches on Github
3. git branch
4. git checkout & git switch
5. git fetch
6. git merge
7. Branch Tracking
8. Resolving conflicts
9. Pull requests and Merging on Github
10. Reviewing Code on Github

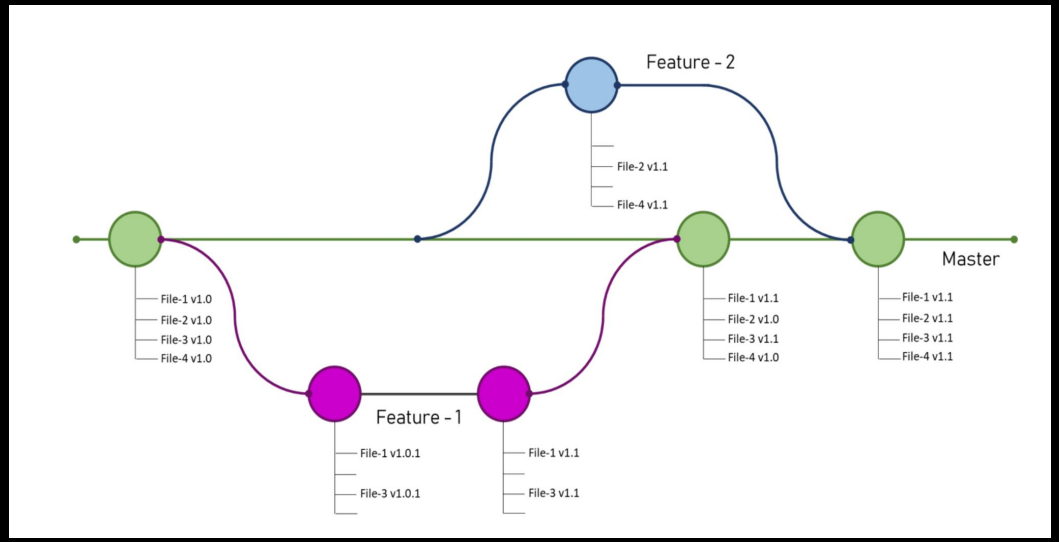
React

6.1 What are Branches ?



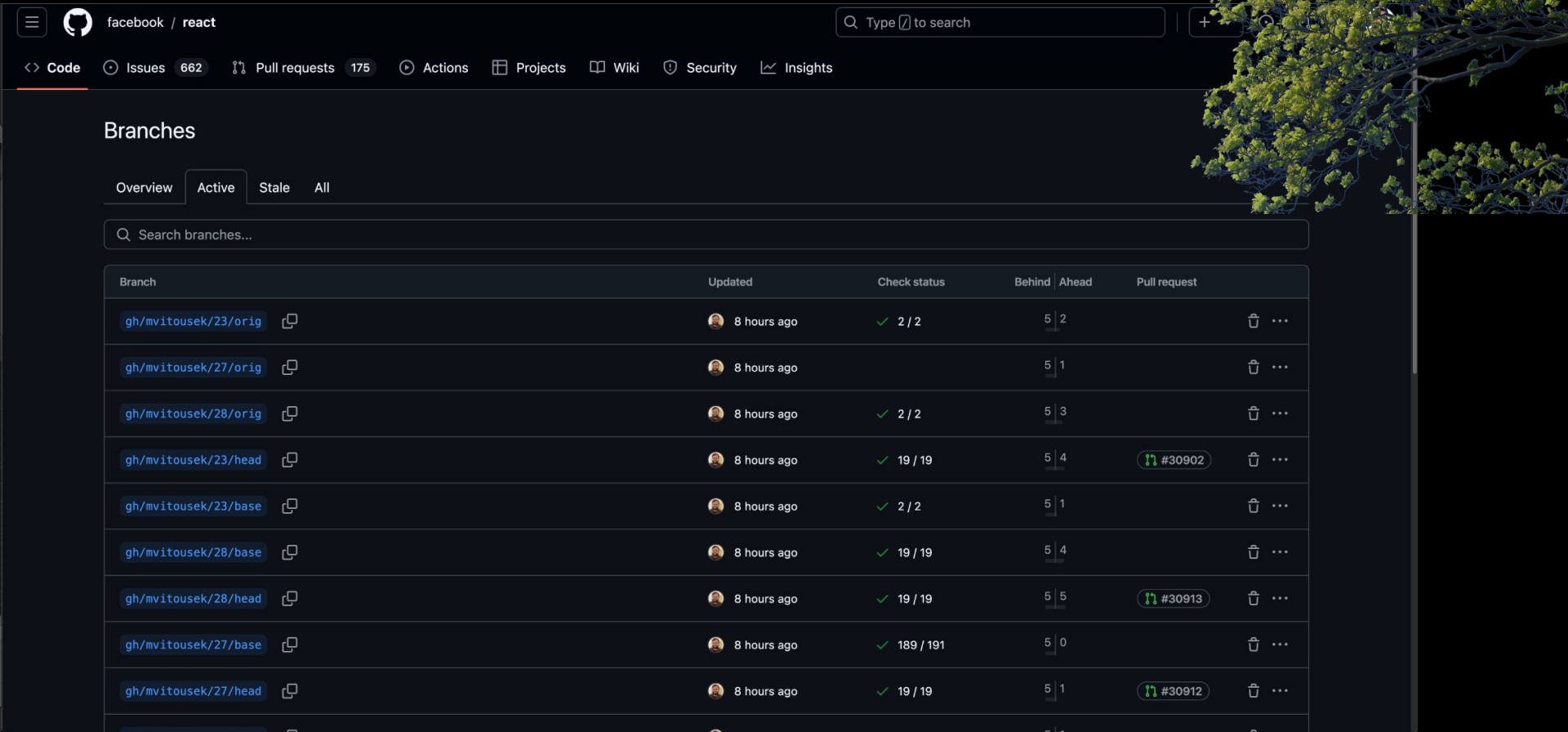
- Git branches are pointers to snapshots of your changes.
- New branches isolate work when adding features or fixing bugs.
- Branching prevents unstable code from affecting the main codebase.
- It allows you to clean up the commit history before merging.
- Each branch represents an independent line of development and records new commits separately.

6.1 What are Branches ?



- **Isolated Development**: Git branches allow working on features independently without affecting the main codebase.
- **Parallel Workflows**: Multiple branches enable team members to collaborate without conflicts.
- **Version Control**: Each branch tracks a separate set of changes.
- **Easy Integration**: Completed branches can be merged back into the main project.

6.2 Branches on GitHub



6.3 git branch

```
git branch
```

List all of the branches in your repository. This is synonymous with `git branch --list`.

```
git branch <branch>
```

Create a new branch called `<branch>`. This does *not* check out the new branch.

```
git branch -d <branch>
```

Delete the specified branch. This is a “safe” operation in that Git prevents you from deleting the branch if it has unmerged changes.

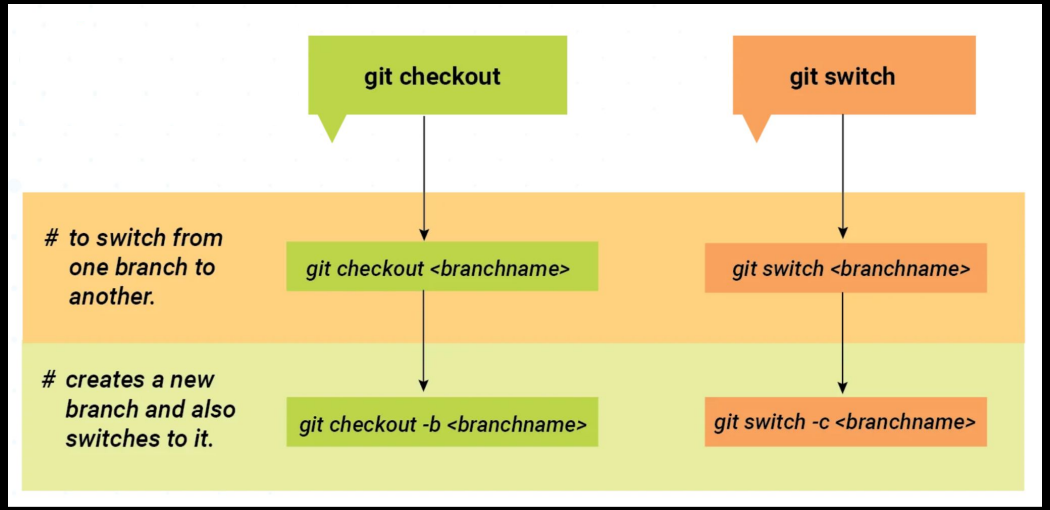
```
git branch -D <branch>
```

Force delete the specified branch, even if it has unmerged changes. This is the command to use if you want to permanently throw away all of the commits associated with a particular line of development.

```
git branch -m <branch>
```

- Lists all branches in the repository.
- Can create, delete, or rename branches.
- It's important to understand that branches are just pointers to commits. When you create a branch, all Git needs to do is create a new pointer, it doesn’t change the repository in any other way.

6.4 git checkout & git switch



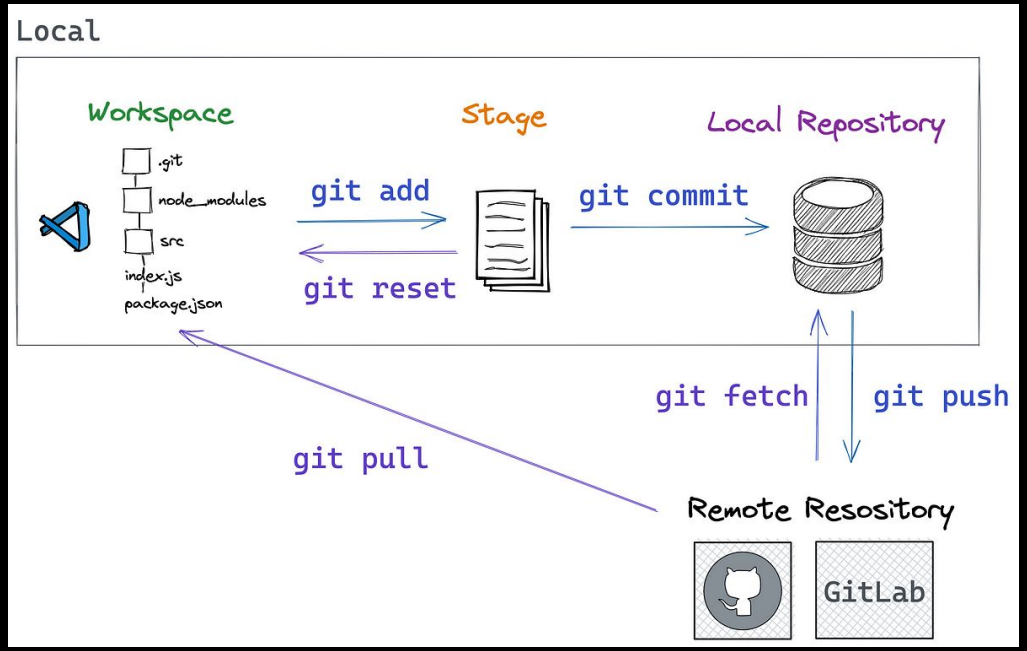
git checkout

- Switches between branches or restores files.
- Can also create and switch to a new branch.

git switch

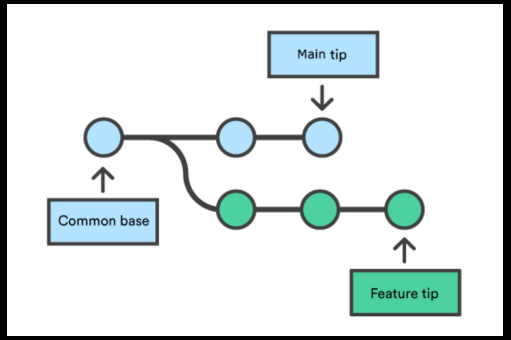
- A simpler command to switch between branches.
- Similar to git checkout but only for switching branches.

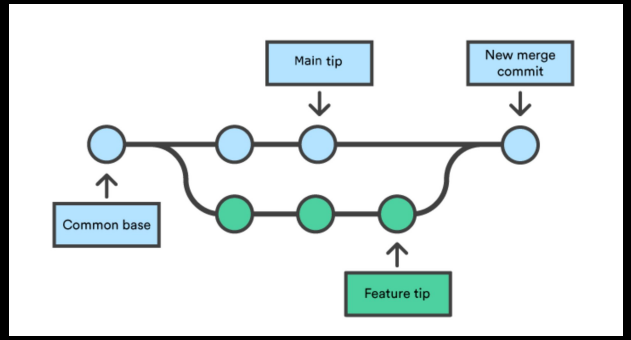
6.5 git fetch



- Downloads changes from a remote repository without merging them.
- Updates local tracking branches.

6.6 git merge






```
# Merge in the new-feature branch
git checkout main
git merge new-feature
git branch -d new-feature
```

- Combines changes from one branch into the current branch.
- Merges commit histories and resolves conflicts if needed.
- Git merge will combine multiple sequences of commits into one unified history.

6.7 Branch Tracking

1. **View Tracking Information**:

```
git branch -vv
```

2. **Tracking a Remote Branch**:

   - You can set a local branch to track a remote branch, meaning it will automatically fetch updates from the remote branch.
   - Command:

```
git branch -u origin/branch_name
```

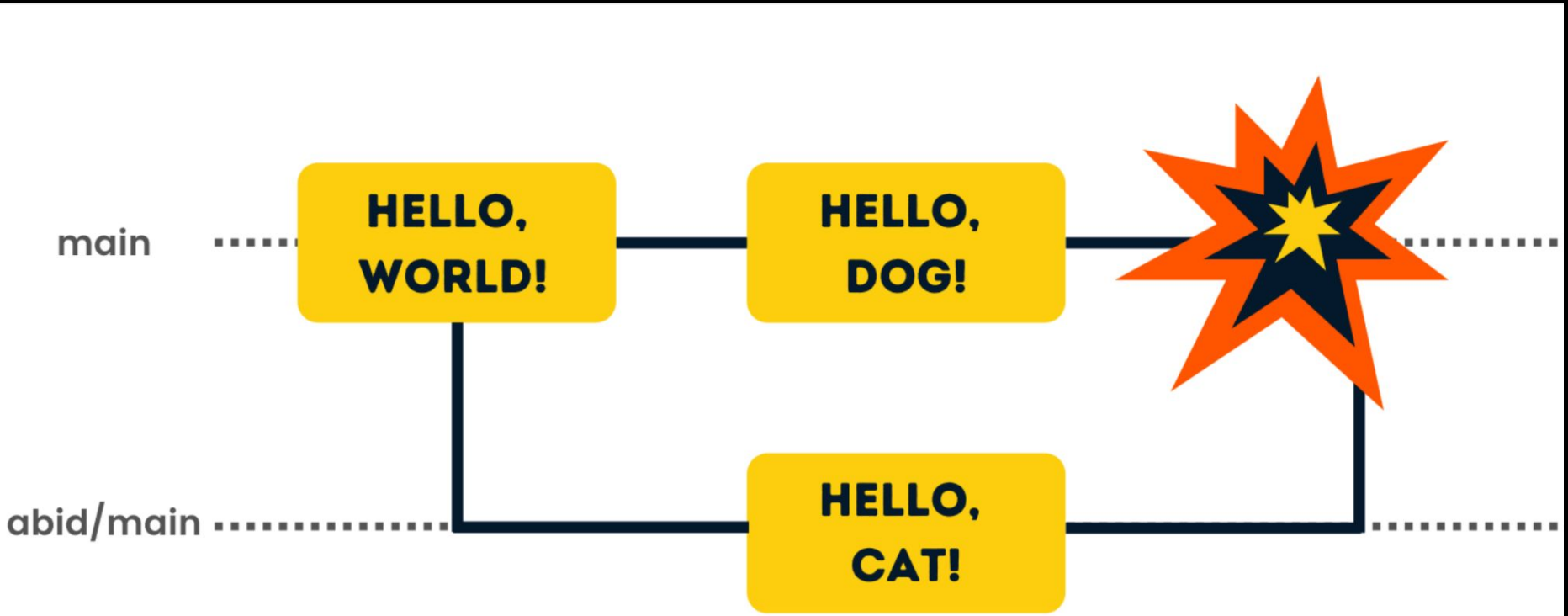
3. **Set Upstream During Checkout**:

   - When creating a new branch and you want it to track the remote branch immediately, use:

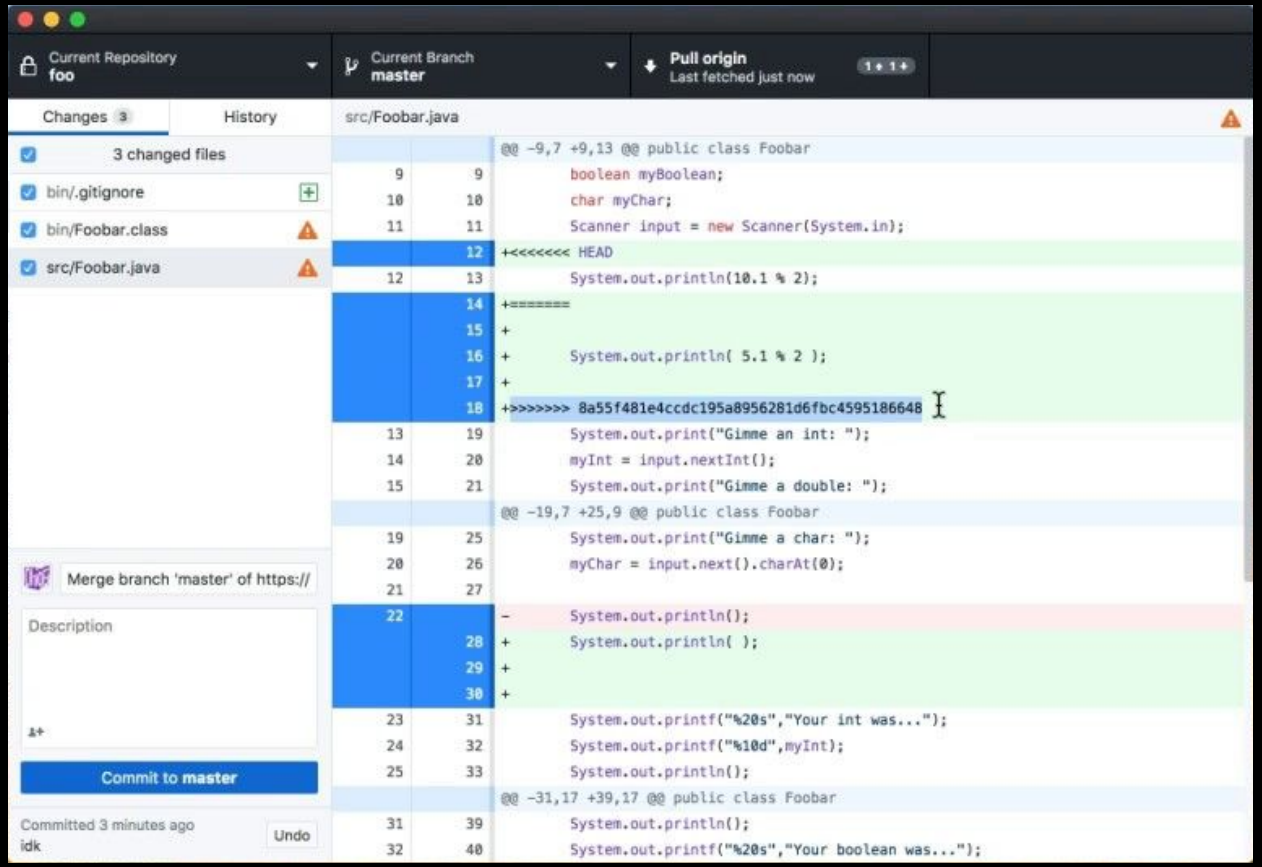
```
git checkout -b branch_name --track origin/branch_name
```

React

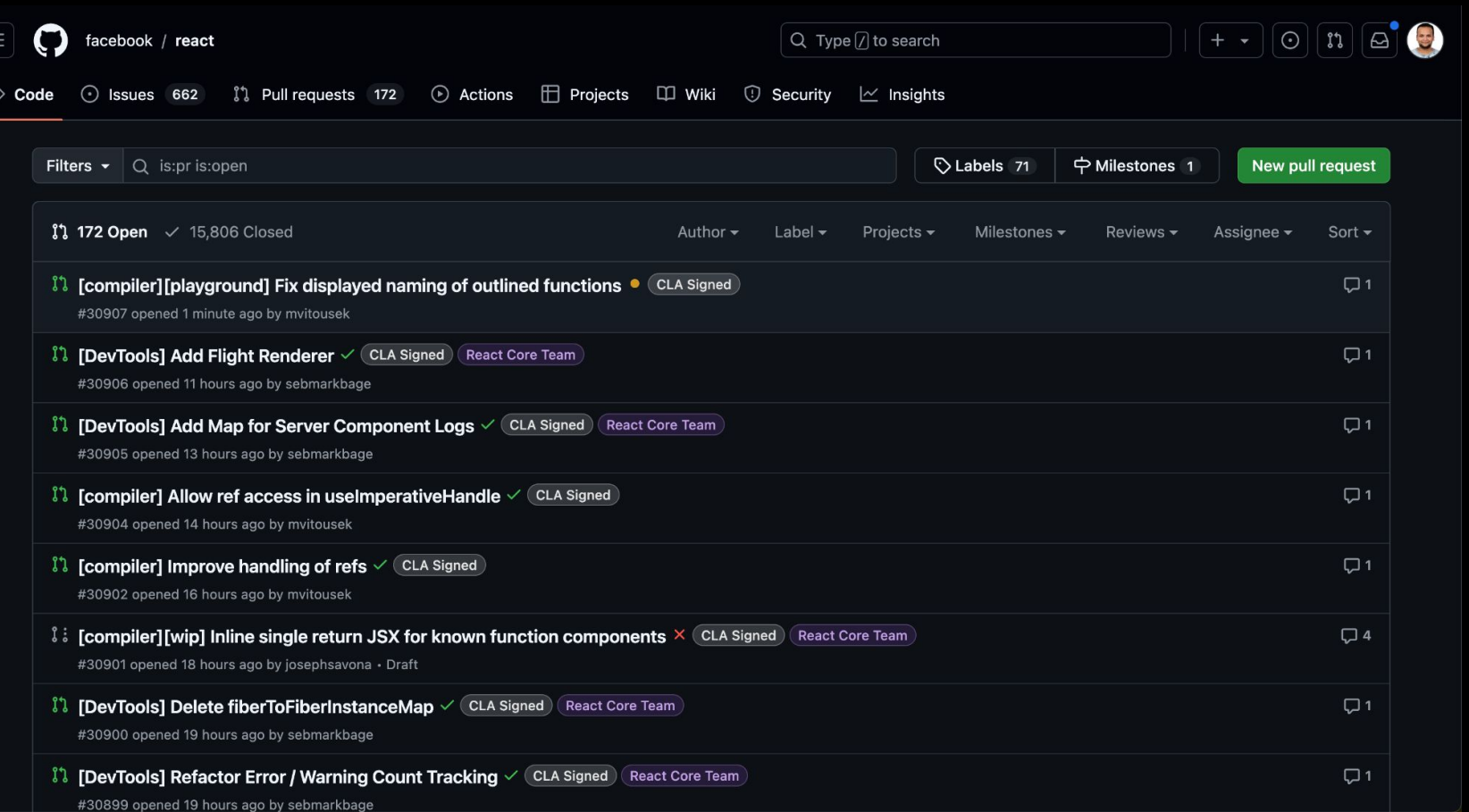
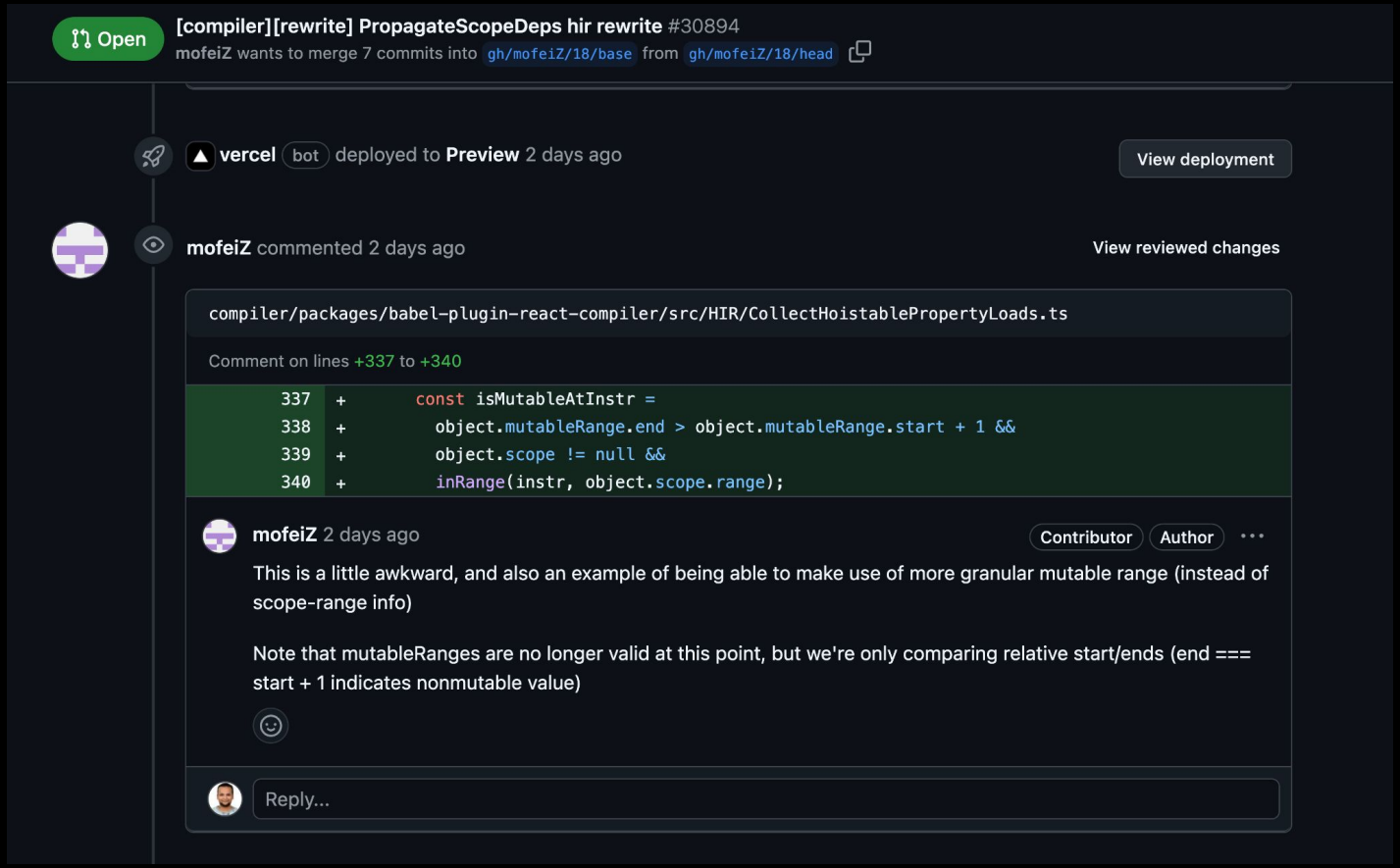
6.8 Resolving conflicts



6.8 Resolving conflicts



6.9 Pull Requests and Merging on Github



6.10 Reviewing Code on Github