

How To Use Git At Scylla Efficiently

Goals of our work

Goals of our work

- Implement features

Goals of our work

- Implement features
- Make code reviews easier to perform

Goals of our work

- Implement features
- Make code reviews easier to perform
- Keep changes understandable in the future

Clone repository

```
# create a fork on GitHub first  
$ git clone git@github.com:ewienik/zpp-git.git  
$ cd zpp-git  
$ git status  
$ git log --oneline
```

Switch branch

```
$ git switch feature  
$ git status
```

Create branch

```
$ git switch -c change-docs  
$ git status
```


Create commits

```
# edit files
$ git status
$ git diff
$ git add .
$ git add -p
$ tig
$ git commit
# repeat as needed
```

Push changes

```
$ git log --oneline
$ git status
$ tig
# create a new branch
$ git push --set-upstream origin change-docs
# update existing branch
$ git push --force-with-lease
```

Rebase with master

```
# two developers working in parallel  
$ git fetch origin  
$ git fetch --all  
$ git rebase origin/feature # not git merge  
# resolve conflicts if any
```

Cleanup before review

Cleanup before review

- divide commits logically

Cleanup before review

- divide commits logically
- write good commit messages

Cleanup before review

- divide commits logically
- write good commit messages
- commits should be as a story

Rearrange, squash or fix commits

```
$ git log --oneline  
$ tig  
$ git rebase -i origin/feature --keep-base  
# rearrange commits in editor
```


Split commits

```
$ git rebase -i origin/feature --keep-base  
# break or edit at some commit to split  
$ git log -1  
$ git reset --soft HEAD~1  
# split staged/not staged changes  
$ git commit -m "part 1"  
$ git add .  
$ git commit -m "part 2"
```

Update commits

```
$ git rebase -i origin/feature --keep-base  
# break or edit at some commit to update  
$ git log -1  
$ git status # nothing to commit, working tree clean  
$ git commit --amend # update description  
# add modification  
$ git add .  
$ git commit --amend # update description and content  
$ git log -1
```

Rebase with stacked branches

```
$ git rebase -i origin/feature --keep-base --update-refs  
# move refs to the correct commits
```

<https://www.codetinkerer.com/2023/10/01/stacked-branches-with-vanilla-git.html>

Rework after review

Tools

<https://git-scm.com/>

<https://jonas.github.io/tig/>

<https://jj-vcs.github.io/>