



Git SCM - Lab 102

Step 5 of 5

3. Go ahead and some content to the **CHANGELOG** file  
4. Push the **Commit changes** button.

This will introduce some new changes on your remote repository that are not yet tracked by your local repository.

**Pull the updates**

To get latest refs and content from your remote projects, use;

```
git pull origin ✓
```

This will fetch the remote changes and merge them with the local branch and update the content of the working tree. The output will look like;

```
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://git.itwork.cloud/aossama/simple-html-app
1459192..90f500b master -> origin/master
Updating 1459192..90f500b
Fast-forward
 CHANGELOG | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 CHANGELOG
```

Now inspect the local working directory using **ls** ✓, and you'll notice the newly created **CHANGELOG** file has been applied to the tree.

**CONTINUE**

**KataCode Editor**

Terminal

```
Compressing objects: 100% (18/18), done.
Writing objects: 100% (13/13), 1.80 KiB | 0 bytes/s, done.
Total 13 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/LukashevichVitali/example.git
 * [new branch] master -> master
Branch master set up to track remote branch master from origin.
$ git fetch origin
$ ls
Dockerfile Jenkinsfile README.md about.html index.html
$ git diff master origin/master
$ ls
Dockerfile Jenkinsfile README.md about.html index.html
$ git pull origin
Already up-to-date.
$ ls
Dockerfile Jenkinsfile README.md about.html index.html
$
```

Git SCM - Lab 201

Step 4 of 4

**Delete merged branch**

After you've merged the branch, it's now safe to do some house keeping and delete the merged branch, use;

```
git branch -d my-feature ✓
```

**Resolving conflict**

If the two branches you're trying to merge both changed the same part of the same file, Git won't be able to figure out which version to use. When such a situation occurs, it stops right before the merge commit so that you can resolve the conflicts manually.

How conflicts are presented When Git encounters a conflict during a merge, it will edit the content of the affected files with visual indicators that mark both sides of the conflicted content. These visual markers are: <<<<<<, =====, and >>>>>>. Its helpful to search a project for these indicators during a merge to find where conflicts need to be resolved.

here is some content not affected by the conflict  
<<<<<< master  
this is conflicted text from master  
=====

this is conflicted text from Feature branch  
>>>>>> feature-717

**CONTINUE**

**KataCode Editor**

Terminal

```
See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': LukashevichVitali
Password for 'https://github.com':
Counting objects: 3, done.
Delta compression using up to 12 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 328 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/LukashevichVitali/example.git
 8927973..e8cc255 master -> master
$ git branch -d my-feature
Deleted branch my-feature (was e8cc255).
$
```

Courseware | VCS | Git.com

LukashevichVitali/project-atomic

Git SCM - Lab 202 | assame |

https://www.katacoda.com/assame/scenarios/git-scm-lab-202

O'REILLYKatacoda

KATACODA OVERVIEW & SOLUTIONS

YOUR PROFILELOG OUT

Git SCM - Lab 202

Step 5 of 5

- Add, commit and push the modified content.

### Review, Approve and Merge

1. Review the *Commits* and *Changes* on the Merge request page.
2. Optionally you can specify that the source branch be removed upon it's merge.
3. When satisfied with the merge request, go ahead and click on **Merge** button.

This will cause a new commit on the master branch with the content of the *contact-us* branch.

### Pull Latest Content

Now as the master has changed on the remote upstream repository, you'll need to pull the latest changes on your local repository. Use;

```
cd ../ ✓
git checkout master ✓
git pull ✓
```

To download and update your working directory to the latest changes on the repository.

CONTINUE

project-atomic/

CHANGELOG

LICENSE

README.md

app.js

package.json

package.json

package.json

app.js

```
1 const express = require('express')
2 const app = express()
3
4 app.get('/', (req, res) => res.send('Hello World!'))
5
6 app.listen(3000, () => console.log('Example app listening on port 3000!'))
7
```

Terminal

2 files changed, 2 insertions(+)

create mode 100644 CHANGELOG

create mode 100644 LICENSE

\$ cd ../

\$ git checkout master

fatal: Not a git repository (or any of the parent directories): .git

\$ git pull

fatal: Not a git repository (or any of the parent directories): .git

\$ pwd

/home/scrapbook/tutorial

\$ cd project-atomic/

\$ git checkout master

error: pathspec 'master' did not match any file(s) known to git.

\$ git pull

Username for 'https://github.com': LukashevichVitali

Password for 'https://LukashevichVitali@github.com':

Already up-to-date.

\$