

Configuration Management

Roberto A. Bittencourt

Based on Dave Matthews and on Github's materials

Configuration Management

- ▶ Management of an evolving system in a **controlled** way
 - ▶ **Version control** tracks component changes as they happen
 - ▶ **System Building** assembles components for testing and
 - ▶ release
 - ▶ **Change Management** addresses stakeholder proposals
 - ▶ **Release Management** plans and prepares for distribution

Version Management

- ▶ **Two models**
 - ▶ **Centralized** – master repository maintains all versions (SVN)
 - ▶ **Distributed** – multiple copies exist at the same time (git)
- ▶ **Features of both**
 - ▶ Version and release identification
 - ▶ Change history recording
 - ▶ Independent Development
 - ▶ Project Support
 - ▶ Storage Management

Distributed Model Benefits

- ▶ A backup mechanism for the master repository
- ▶ Allows developers to work offline
 - ▶ commit changes without a network connection
 - ▶ Developers can compile and test locally

System Building


- ▶ Build script generation (configuration file)
- ▶ Build system integration with version control system
- ▶ Minimal recompilation (what changed or affected)
- ▶ Executable system creation
- ▶ Test automation (check build not broken by changes)
- ▶ Report success or failure of build and test
- ▶ Documentation (release notes) automatically generated

Change Management

- ▶ Ensure changes are applied in a controlled way
 - ▶ requirements, bugs, ...
- ▶ Consider factors in decisions to changes
 - ▶ Consequences
 - ▶ Benefits
 - ▶ Number of users affected
 - ▶ Cost
 - ▶ Product release cycle

Release Management

- ▶ Plan the release
- ▶ Prepare the system for release
 - ▶ Configuration files
 - ▶ Data files
 - ▶ Installation program
 - ▶ Electronic and paper documentation
 - ▶ Packaging and associated publicity
- ▶ Document the release



GitHub and git

GitHub and git

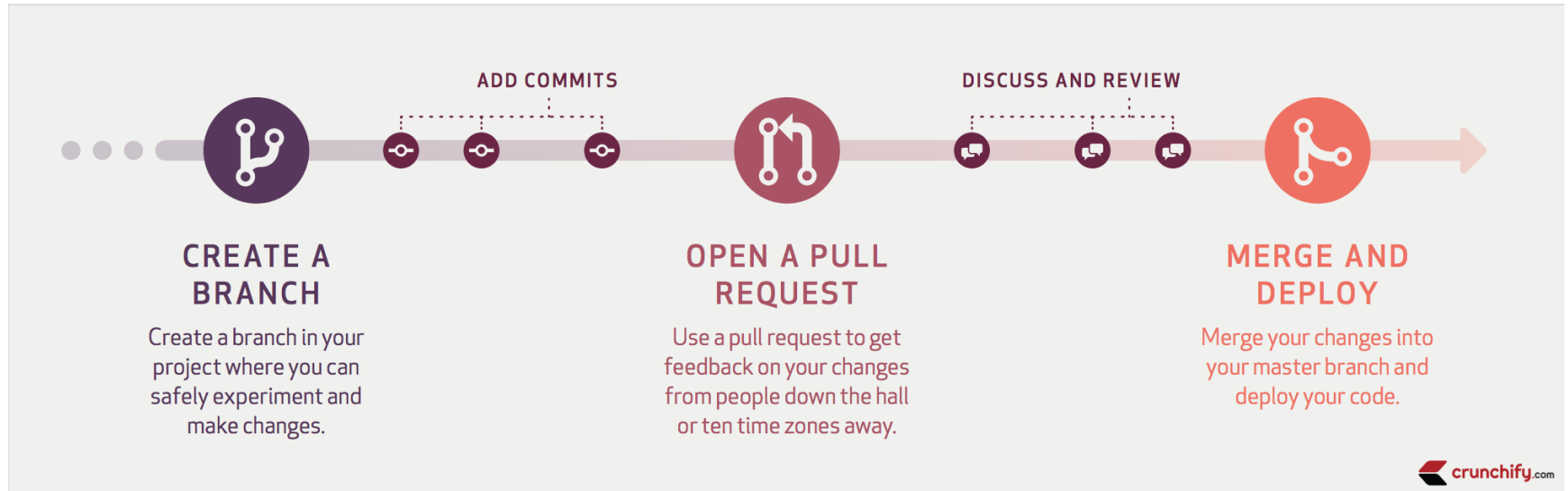
▶ GitHub (server)

- ▶ On the internet, holds the master repo, issues, releases, ...
- ▶ Code on master should always build/test/run with no problems
- ▶ No changes are made directly in master
- ▶ Proposed changes are reviewed, approved, then merged

▶ git (client)

- ▶ On a remote/local machine
- ▶ Holds a clone of the master
- ▶ Changes made in branches pushed back to the GitHub master and merged via a pull request

Using GitHub and git



GitHub

Select, estimate and assign a pending issue

git

Refresh (pull) or clone repo if needed. Create and checkout a new local branch in your repo clone

git

Modify and add files and directories in the repo. Build and test before you commit.

git

Add your changes to the branch. Commit the branch with the #issue. Push branch to master.

GitHub

Open a pull request for the commit for review by others.

GitHub

Address and merge conflicts or comments. Merge the pull request and confirm.

git commands – local setup

- ▶ *# install git on your local system*
- ▶ *# configure git username, email*
- ▶ **git config --global user.name “[firstname lastname]”**
- ▶ **git config --global user.email “[valid-email]”**
- ▶ *# clone your team repo or the class repo*
- ▶ **git clone [masterURL]**

git commands – start a new branch

- ▶ *# update your local copy before you start*
- ▶ **git pull origin master**
- ▶ *# start a new branch*
- ▶ **git branch [newbranchname]**
- ▶ **git checkout [newbranchname]** *# never master!*
- ▶ *# on a single line*
- ▶ **git checkout -b [newbranchname]**
- ▶ *# verify the branch just to be sure, never master*
- ▶ **git branch**

git commands – commit and push

- ▶ *# create, edit, rename, move, or delete files under “.”*
- ▶ *# build and test to verify changes work*
- ▶ **git add .** *# add all changes to branch*
- ▶ **git status** *# verify proposed changes are listed*
- ▶ **git commit -m “closes #999”** *# associate with task 999*
- ▶ **git push origin [branchname]**

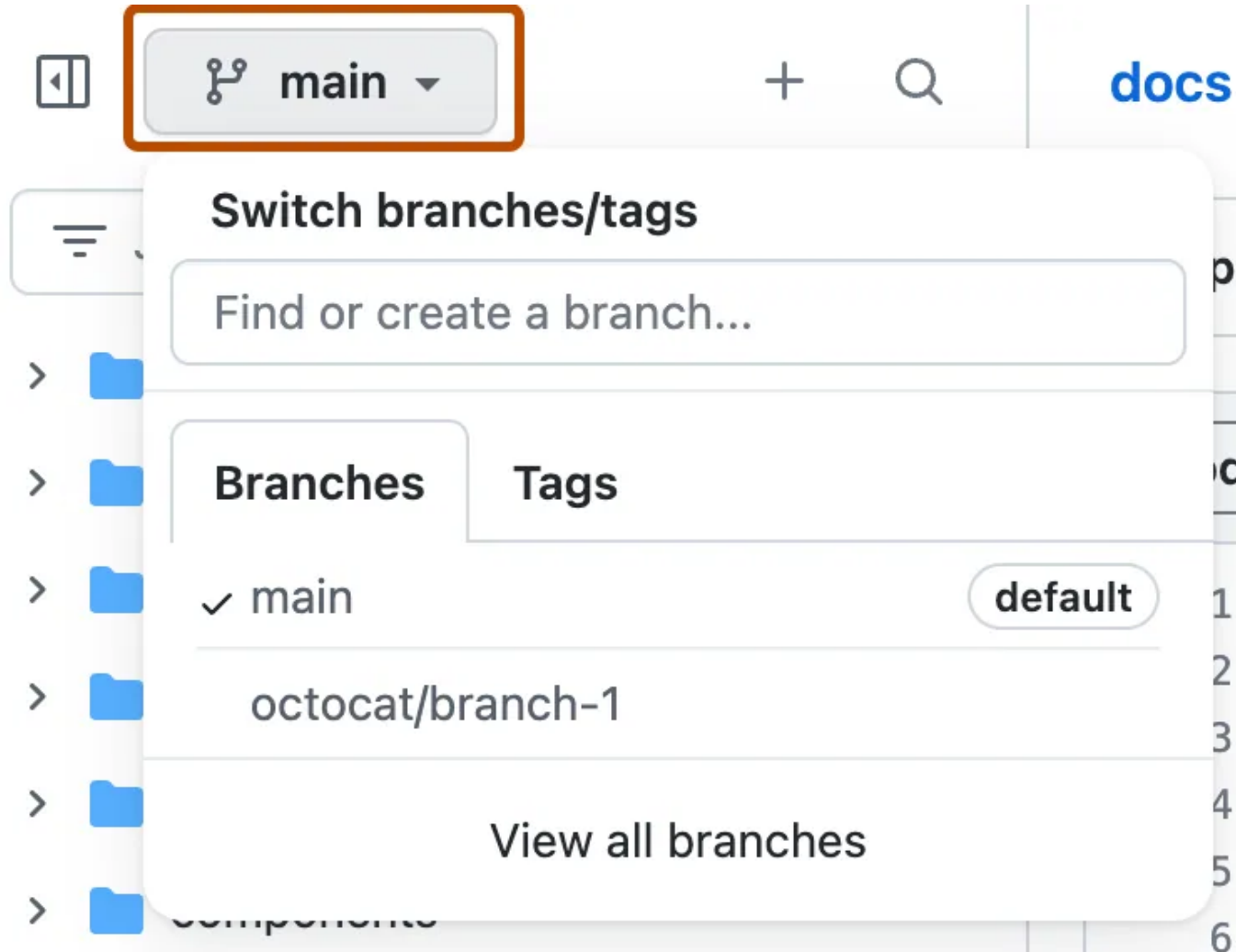
git commands – merge conflicts

- ▶ *# find files with merge conflicts on your local repo*
- ▶ **git status**
- ▶ *# edit files to resolve the conflicts between*
- ▶ *# <<<<<< HEAD and >>>>>> BRANCH-NAME*
- ▶ *# re build and test*
- ▶ **git add .**
- ▶ **git commit -m “resolved merge conflict”**
- ▶ **git push origin [branchname]**


GitHub Etiquette


- ▶ No changes made directly to master branch
 - ▶ Never checkout master
- ▶ All changes made in local/separate branches and merged via pull requests
- ▶ All pull requests associated with an issue
- ▶ Never break master
 - ▶ It should always build/test/run successfully

GitHub - Switching branches



GitHub – Make changes in your branch

 Files

octo-repo / .github / 

Add file ▾ ...

octo-repo / .github / workflows / codeql-analysis.yml in main

Cancel changes

Commit changes...


Edit

Preview

Spaces ▾

2 ▾


No wrap ▾



```
1 # For most projects, this workflow file will not need changing; you simply need
2 # to commit it to your repository.
3 #
```

Update README.md

Add an optional extended description...

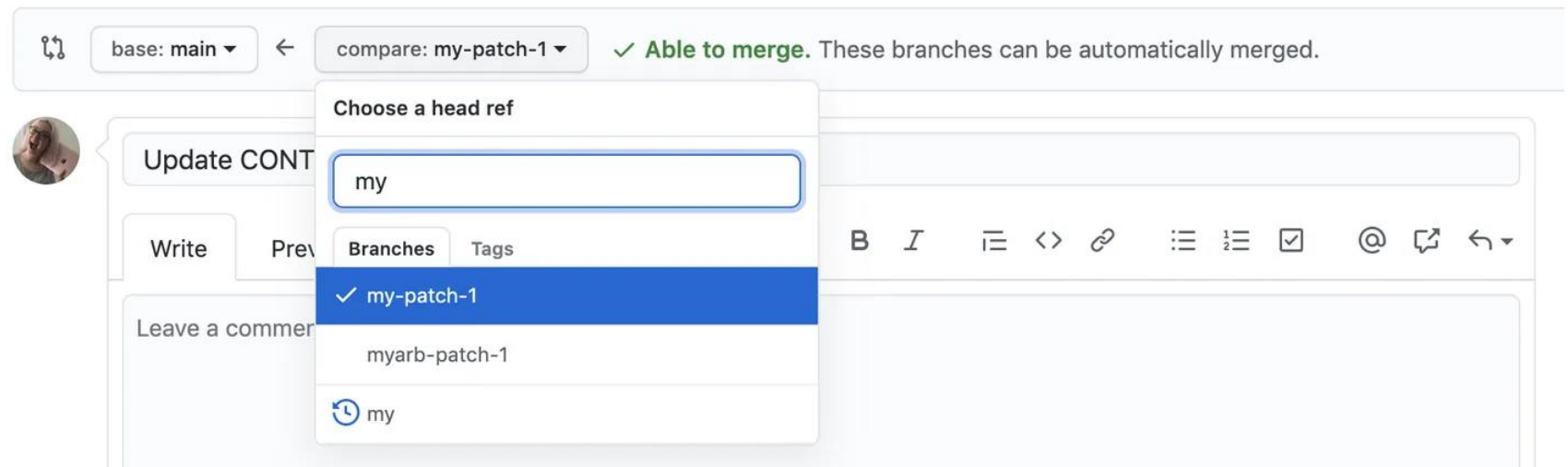
 ✓ octocat@github.com

it



GitHub - Create a pull request

Open a pull request





Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



GitHub – Review proposed changes



  github / docs

Q Type to search

[Code](#) [Issues 46](#) [Pull requests 19](#) [Discussions](#) [Actions](#) [Projects 6](#) [Security 7](#) [Insights](#) [Settings](#)

[Conversation 0](#) [Commits 1](#) [Checks 1](#) [Files changed 24](#)

2  .github/workflows/codeql-analysis.yml 

↑

@@ -13,7 +13,7 @@ name: "CodeQL"

13 13

14 14 on:

15 15 push:

16 - branches










16 + branches

17 pull_reque


18 # The br

19 branches

↓

Write Preview  H B I  <>     @   

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

Cancel

Add single comment

Start a review

▶ 19

GitHub – Merge pull requests

Merge pull request



You can also [open this in GitHub Desktop](#)

✓ Create a merge commit

All commits from this branch will be added to the base branch via a merge commit.

Squash and merge



The 1 commit from this branch will be added to the base branch.







Rebase and merge

The 1 commit from this branch will be rebased and added to the base branch.

asting them.

GitHub – Address merge conflicts

  github / docs

Q Type  to search     

<> Code Issues 46 Pull requests 19 Discussions Actions Projects 6 Security 7 Insights Settings

 **This branch has conflicts that must be resolved**
Use the [web editor](#) or the [command line](#) to resolve conflicts.
Conflicting files

Resolve conflicts

content/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue.md

and main and committing changes → octocat-pull-req...

content/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue.md 1 conflict Prev ^ Next v 

Mark as resolved

```
1 ---
2 title: Linking a pull request to an issue
3 intro: 'You can link a pull request {% ifversion link-existing-branches-to-issue %}or branch {% endif %}to an issue to show that a fix is
4 redirect_from:
5   - /github/managing-your-work-on-github/managing-your-work-with-issues-and-pull-requests/linking-a-pull-request-to-an-issue
6   - /articles/closing-issues-via-commit-message
7   - /articles/closing-issues-via-commit-message
```

Resolving conflicts between octocat-pull-req... and main and committing changes → octocat-pull-req...

Commit merge

1 conflicting file	content/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue.md	✓ Resolved
linking-a-pull-request-to-an-issue.md ...linking-a-pull-request-to-an-issue.md	<pre>1 --- 2 title: Linking a pull request to an issue 3 intro: 'You can link a pull request {% ifversion link-existing-branches-to-issue %}or branch {% endif %}to an issue to show that a fix i 4 redirect_from: 5 - /github/managing-your-work-on-github/managing-your-work-with-issues-and-pull-requests/linking-a-pull-request-to-an-issue 6 - /articles/closing-issues-via-commit-message</pre>	