

Continuous Integration with Travis CI

Instructor:
John J Rofrano

Senior Technical Staff Member, DevOps Champion
IBM T.J. Watson Research Center
rofrano@us.ibm.com (@JohnRofrano)

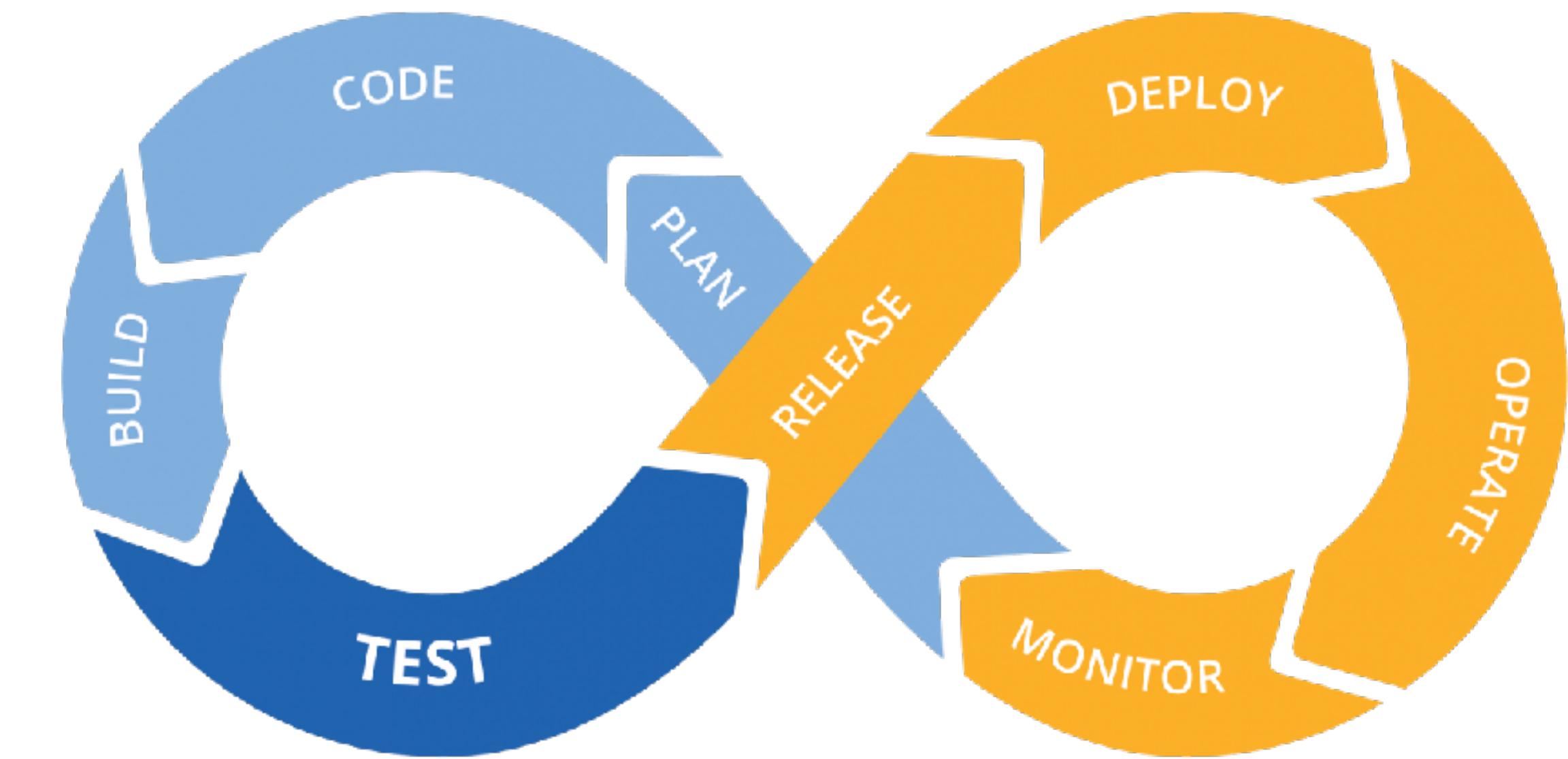
What Will You Learn?

- Gain a basic understanding of Continuous Integration
- Learn how to use Travis for your GitHub Repositories
- Understand how to construct a `.travis.yml` file to run jobs



Continuous Integration (CI) vs Continuous Delivery (CD)

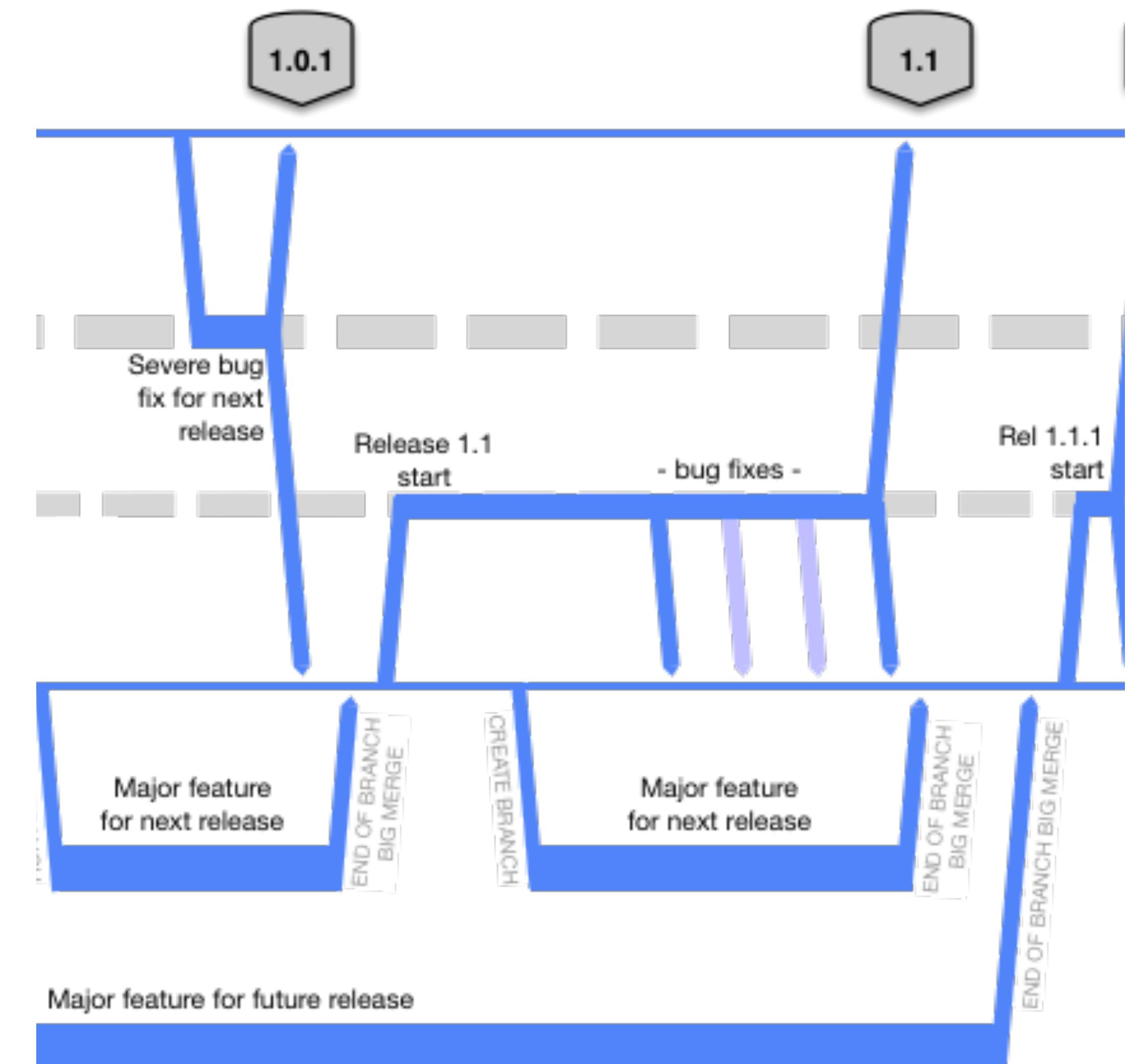
- Continuous Integration
 - The process of continuously integrating every developer change into the master branch after a set of tests have passed resulting in potentially deployable code
- Continuous Delivery
 - A series of practices designed to ensure that code can be rapidly and safely deployed to production by delivering every change to a production-like environment



Traditional Development

- Developers work in long lived development branches
- Branches are periodically merged into a release (with lots of breakage)
- Builds are run periodically (e.g., nightly) on release candidate branch
- Developers continue to add to the development branch which drifts further and further from the master branch

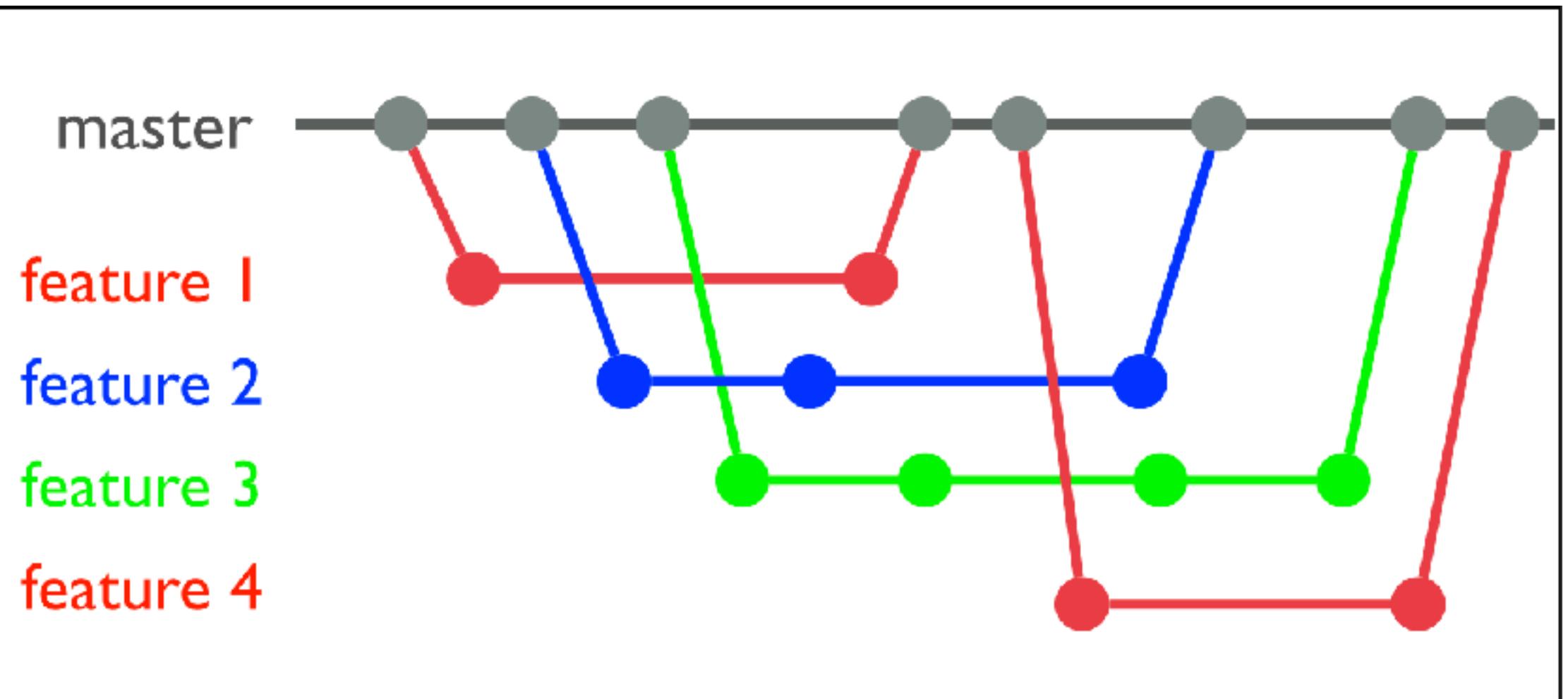
Long lived development branches breed merge nightmares



Continuous Integration

- A development practice that requires developers to integrate code into a shared repository often (e.g., daily)
- Developers work in short lived feature branches that are merged into master
- Each check-in is then verified by an automated build, allowing teams to detect problems early and often

Short feature branches makes merging easy



Everyone commits to the baseline

- Committing regularly, every developer can reduce the number of conflicting changes
 - Checking in a week's worth of work runs the risk of conflicting with other features and can be very difficult to resolve
- Using Pull Requests allows team members to communicate about the change they are making
- Committing all changes at least once a day (once per feature built) is generally considered part of the definition of Continuous Integration



Every commit (to baseline) should be built

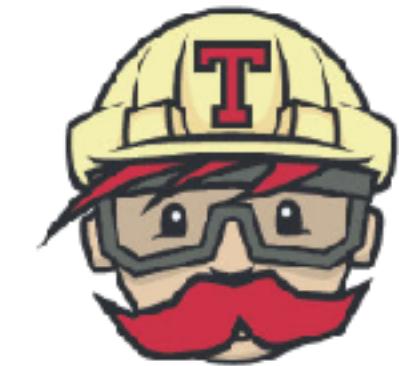
- The system should build commits to the current working version to verify that they integrate correctly
- A common practice is to use automated Continuous Integration where a continuous integration server or daemon monitors the revision control system for changes, then automatically runs the build process
- Once the code is built, all tests should run to confirm that it behaves as the developers expect it to behave (i.e, make the build self-testing)



Jenkins



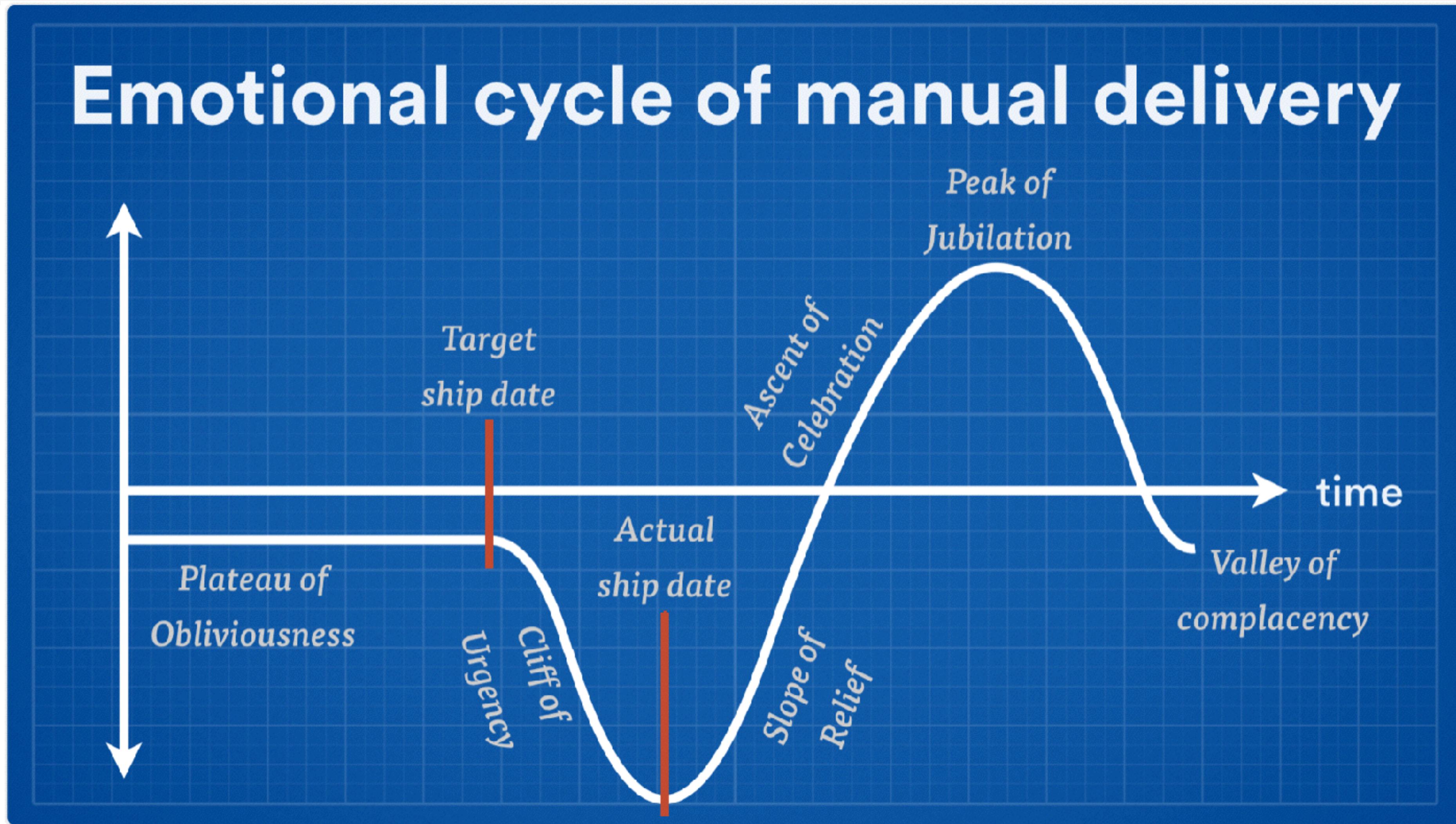
circleci



Travis CI

Tools like Jenkins, Travis CI, and Circle CI can build every Pull Request for you before merging to master

Delivery Should not be Emotional



Delivery Should Continuous



Delivery Should Continuous



Benefits of CI/CD

- Faster reaction times to changes
- Reduced code integration risk
- Higher code quality
- The code in version control works
- Less deployment time



Five Key Principles at the heart of Continuous Delivery

- Build quality in
- Work in small batches
- Computers perform repetitive tasks,
people solve problems
- Relentlessly pursue continuous
improvement
- Everyone is responsible

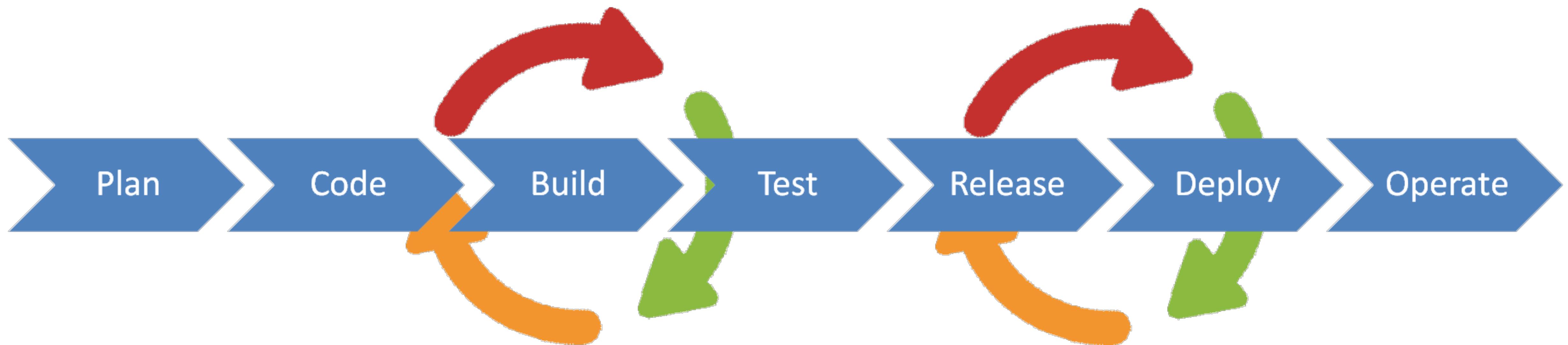


A CI/CD Pipeline Needs

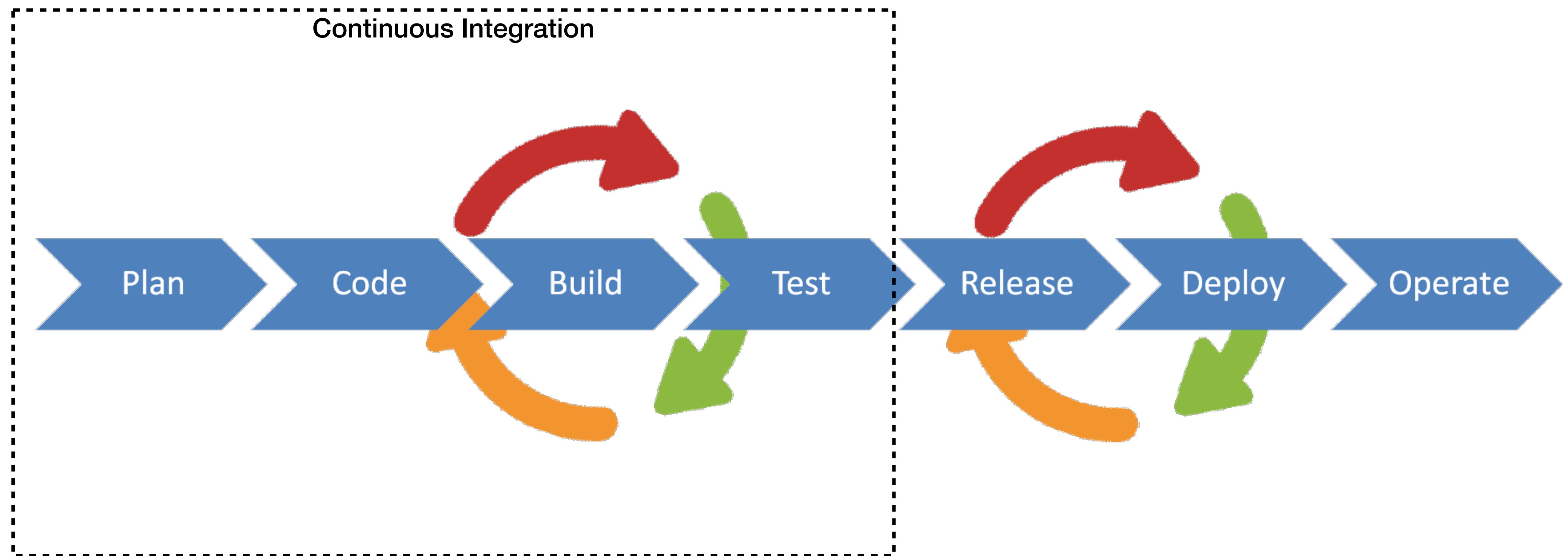
- A code **repository** to host and manage all your source code
- A **build server** to build the application from source code
- An **integration server/orchestrator** to automate the build and run test code
- An **artifact repository** to store all the binaries and other artifacts of the application
- Tools for **automatic** configuration and **deployment**



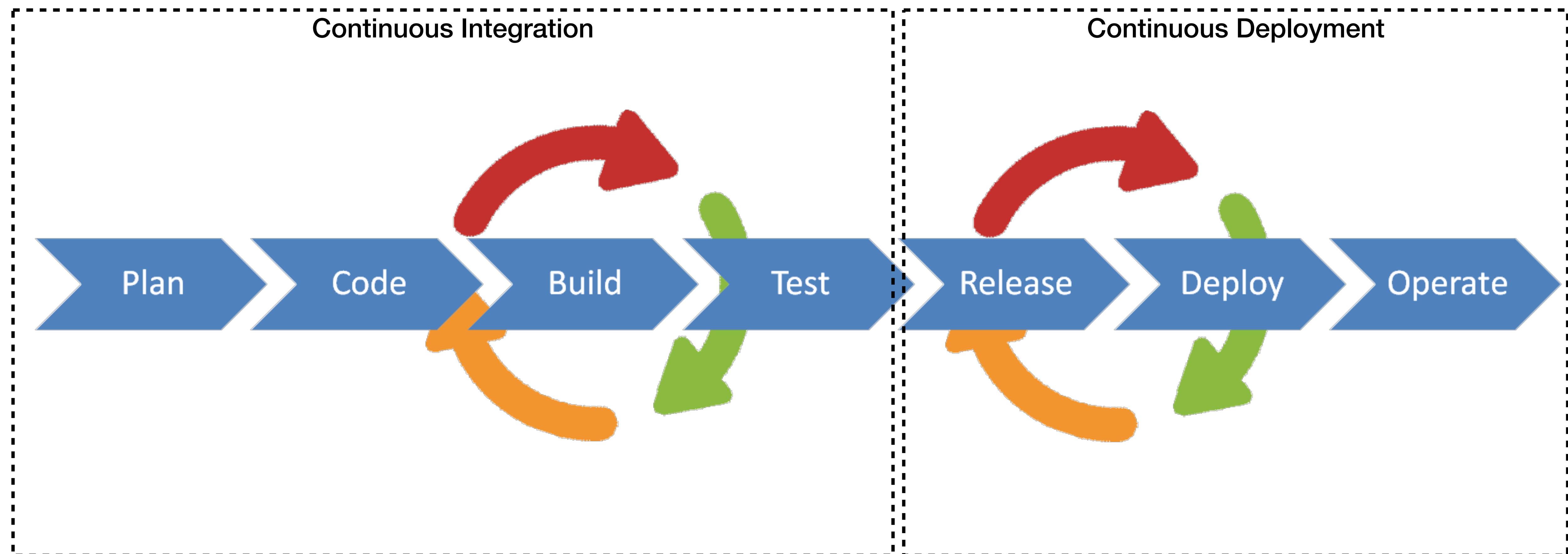
Continuous Integration and Deployment



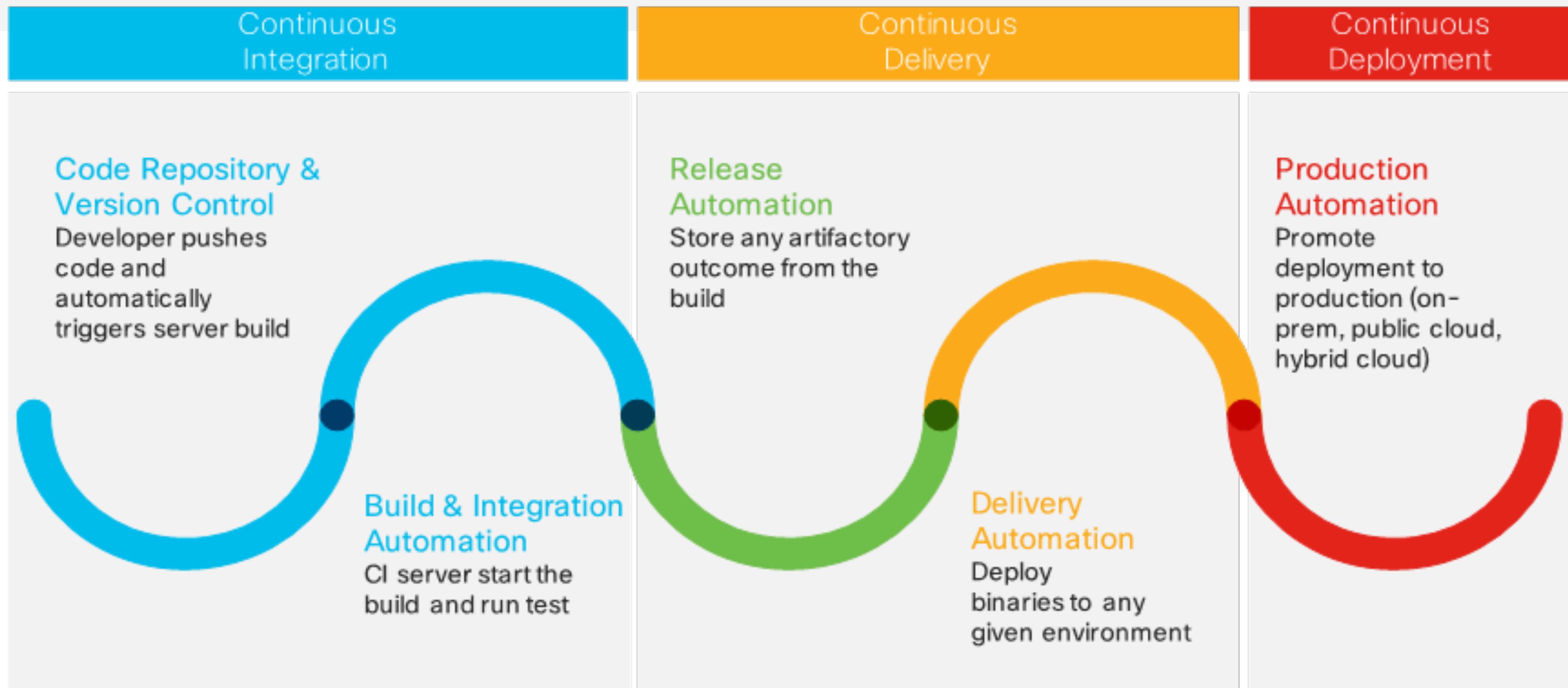
Continuous Integration and Deployment



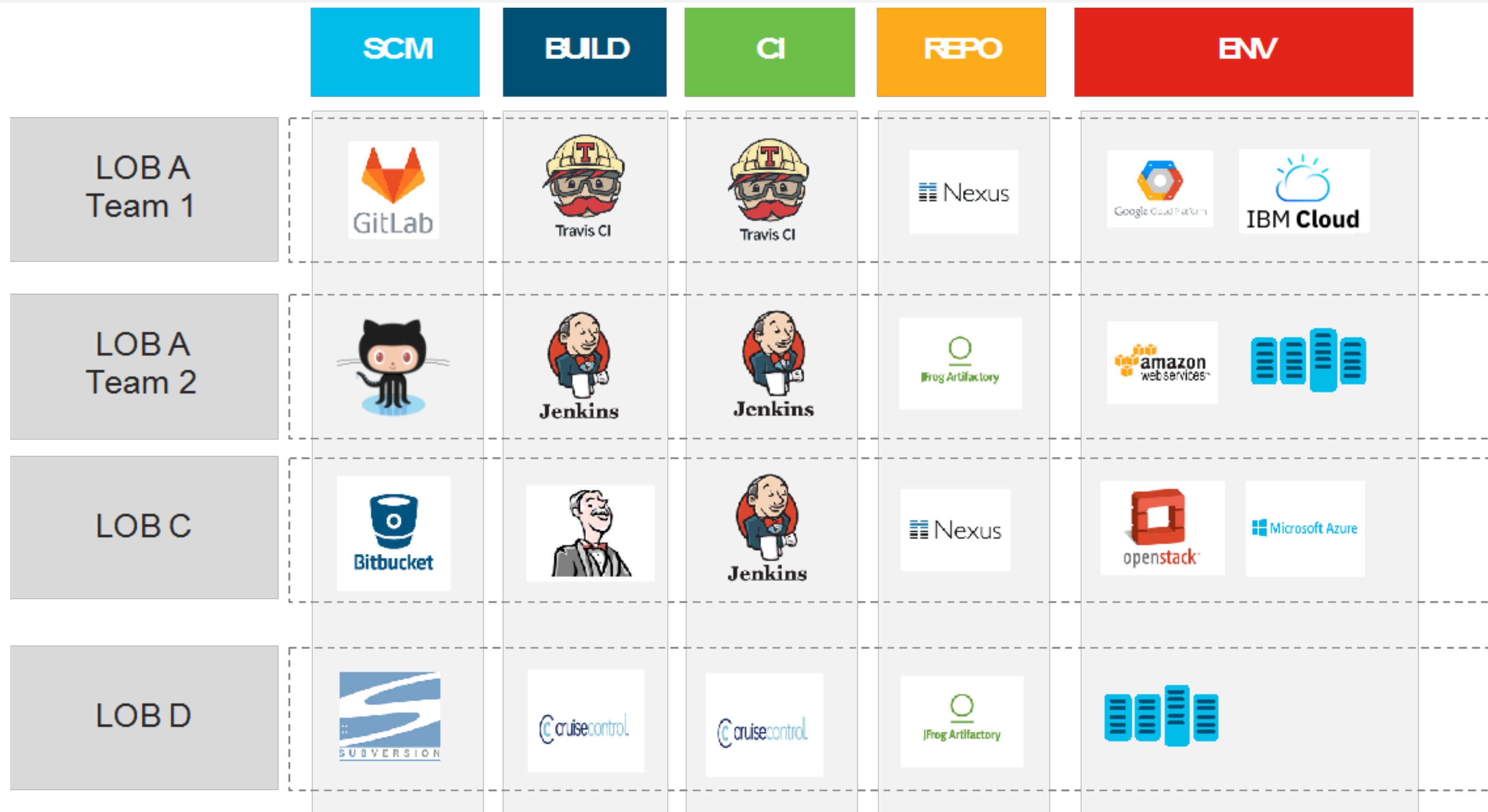
Continuous Integration and Deployment



CI/CD + Continuous Deployment



Pipelines can use many Tools



Automation is key but insufficient on it's own

- For business to take advantage of continuous integration and continuous delivery, they must:
 - Have an **application design** that allows services to be deployed quickly and independently
 - Have automated all of their **testing**
 - Have organizational trust that allows **autonomy** and **shared responsibility**
 - Have **fast feedback loops** with minimal Mean Time to Recovery



Some CI/ CD Providers



Jenkins



Travis CI



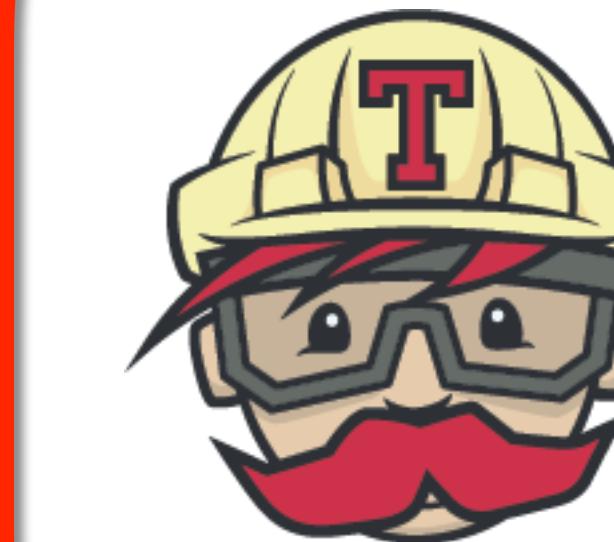
semaphore

...there's a whole lot more

Some CI/ CD Providers



Jenkins



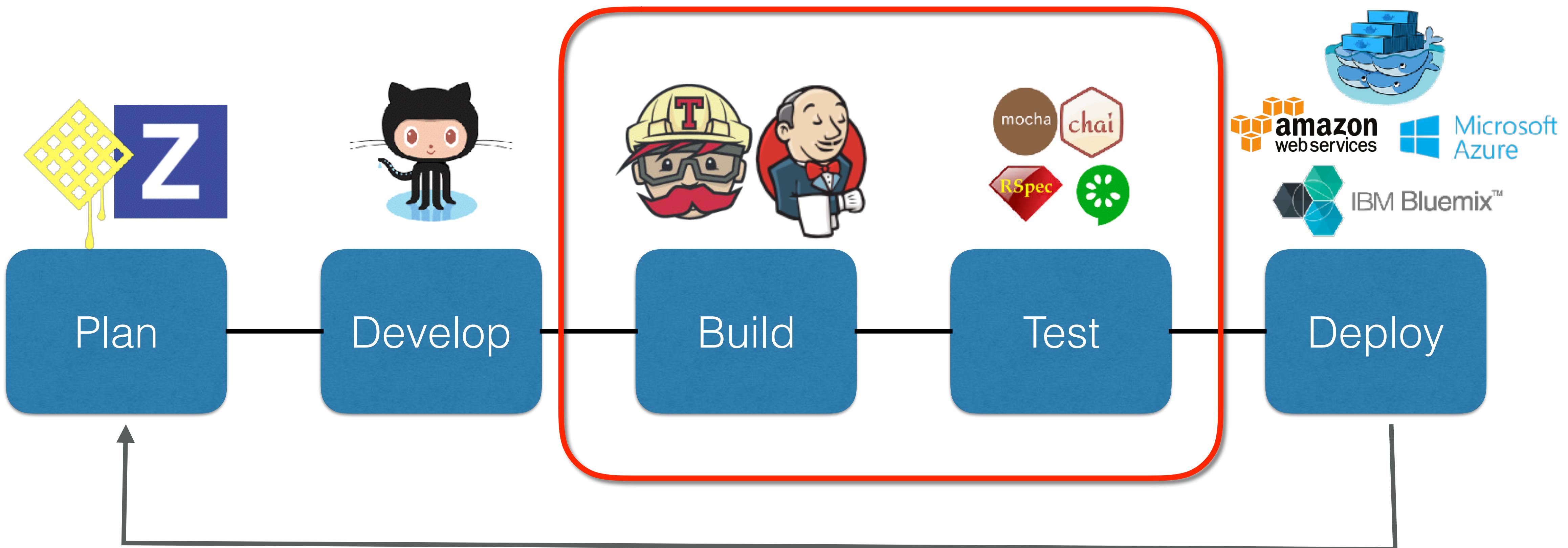
Travis CI



semaphore

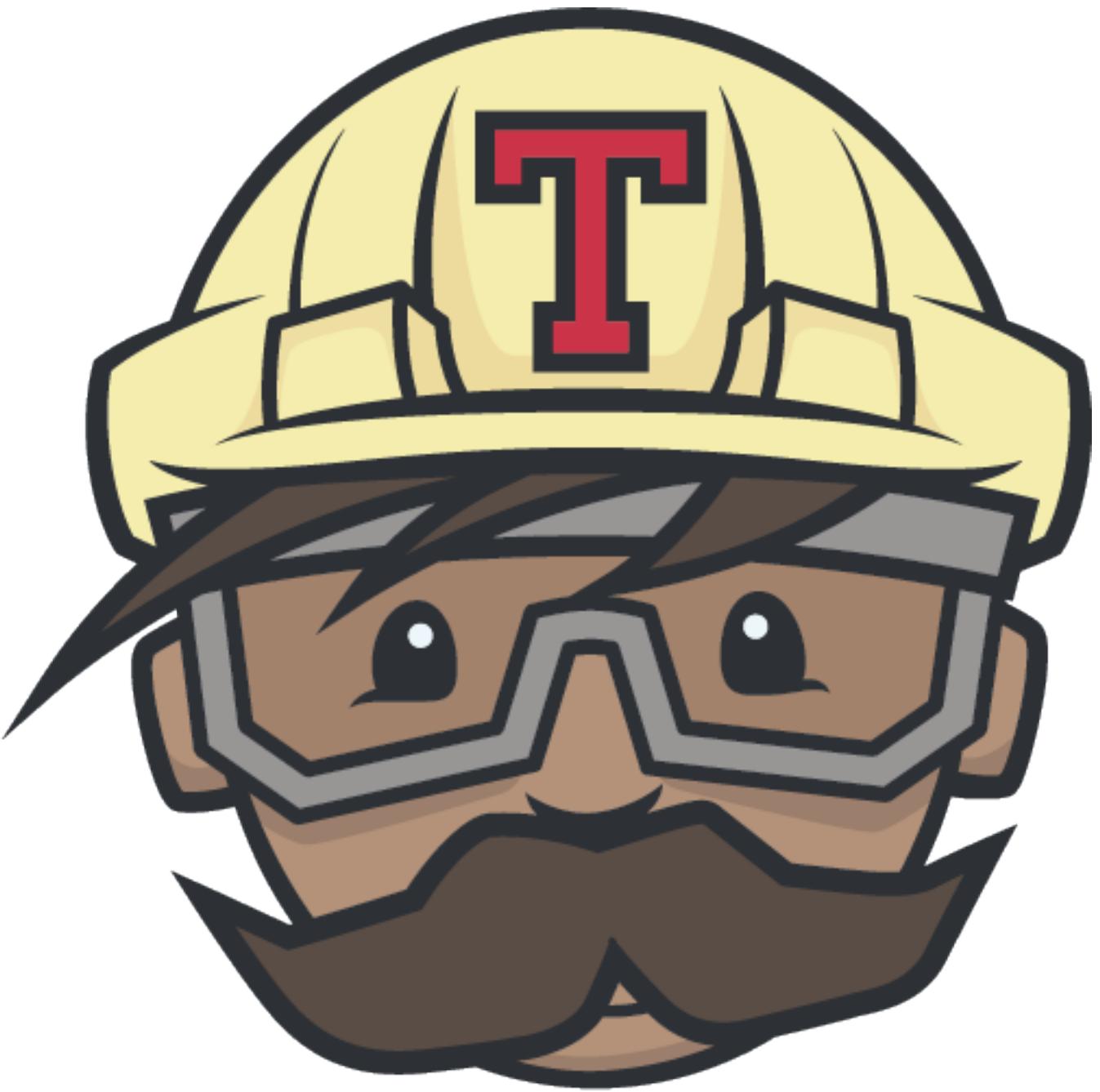
...there's a whole lot more

DevOps Pipeline



What is Travis CI?

- Travis CI is a Continuous Integration tool
- It allows you to treat your CI pipeline as code
- It can also do deployments for Continuous Delivery



Continuous Delivery Goals

- Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time
 - Martin Fowler
- That means:
 - The Master branch should always be ready to deploy
 - You need a way to know if something will "break the build"
- Travis CI gives you a way to achieve this



Know if the Build is Broken

- Automated Testing is CRITICAL to Continuous Integration
- Travis CI can work with any language and any test suite
- Lots of languages and test suite support is built-in
- Database support is built-in
- Webhooks allow Travis-CI to run on every Pull Request
- Status is delivered to the Pull Request before a merge



Travis CI Language Support

- C
- C++
- Clojure
- C#
- D
- Dart
- Erlang
- F#
- Go
- Groovy
- Haskell
- Haxe
- Java
- JavaScript (with Node.js)
- Julia
- Objective-C
- Perl
- PHP
- Python
- R
- Ruby
- Rust
- Scala
- Smalltalk
- Visual Basic

Travis CI Databases Support

- STARTING SERVICES
- MYSQL
- POSTGRESQL
- MARIADB
- SQLITE3
- MONGODB
- COUCHDB
- RABBITMQ
- RIAK
- MEMCACHED
- REDIS
- CASSANDRA
- NEO4J
- ELASTICSEARCH
- RETHINKDB
- MULTIPLE DATABASE BUILDS

Travis CI

<http://travis-ci.org/>

- Travis CI is available at:
 - <http://travis-ci.org/>
- This is where you manage which repositories to use Travis with
- This is also where the logs from your build can be seen

The screenshot shows the Travis CI web interface. On the left, there's a sidebar titled 'My Repositories' listing three repositories: 'one-fish/two-fish', 'hop-on/pop', and 'horton-hears/awho'. Each entry includes a green checkmark icon, the repository name, a build number (#), duration, and a timestamp indicating when it finished. On the right, the main area displays the 'green-eggs / ham' repository. It shows a 'Current' tab with a green checkmark icon and the text 'master adding in Oh the places you'll go!'. Below this, there are sections for 'Commit abc123', 'Branch master', and a note from 'Sven Fuchs' about authoring and committing. At the bottom, there are buttons for 'Job log' and 'View config', and links for 'Remove log' and 'Raw log'.

How Does Travis CI Work?

- You create a `.travis.yml` file in the root of your project
- Depending on how you configure Travis CI
 - Whenever you push to master a build is executed
 - Whenever you issue a pull request a build is executed
- Build runs in a isolated environment (VM or Docker)



Example .travis.yml file

```
.travis.yml

language: python
python:
  - "2.7"

# command to install dependencies
install: "pip install -r requirements.txt"

# command to run tests
before_script: redis-cli ping
script: python -m unittest test_server

services:
  - redis-server
```

Travis CI - Build Lifecycle

```
addons:  
before_install:  
install:  
before_script:  
script:  
after_success:  
after_failure:  
before_deploy:  
deploy:  
after_deploy:  
after_script:
```

Hands-On

“live session”

Some Assembly Required

- Tools you will need to complete this lab:
 - GitHub Account
 - Travis CI Account
 - Git Client
 - Text Editor



Contents of .travis.yml

```
language: python
python:
  - "3.6"

services:
  - postgresql

env:
  global:
    - DATABASE_URI: 'postgres://postgres:@localhost:5432/test'

# command to install dependencies
install: "pip install -r requirements.txt"

# command to run tests
before_script:
  - psql -c 'create database test;' -U postgres

script:
  - nosetests

after_success:
  - codecov
```

How to You Know What Parameters to Use

PostgreSQL #

Start PostgreSQL in your `.travis.yml`:

```
.travis.yml
services:
- postgresql
```

Using PostgreSQL in your Builds #

The default user for accessing the local PostgreSQL server is `postgres` with a blank password.

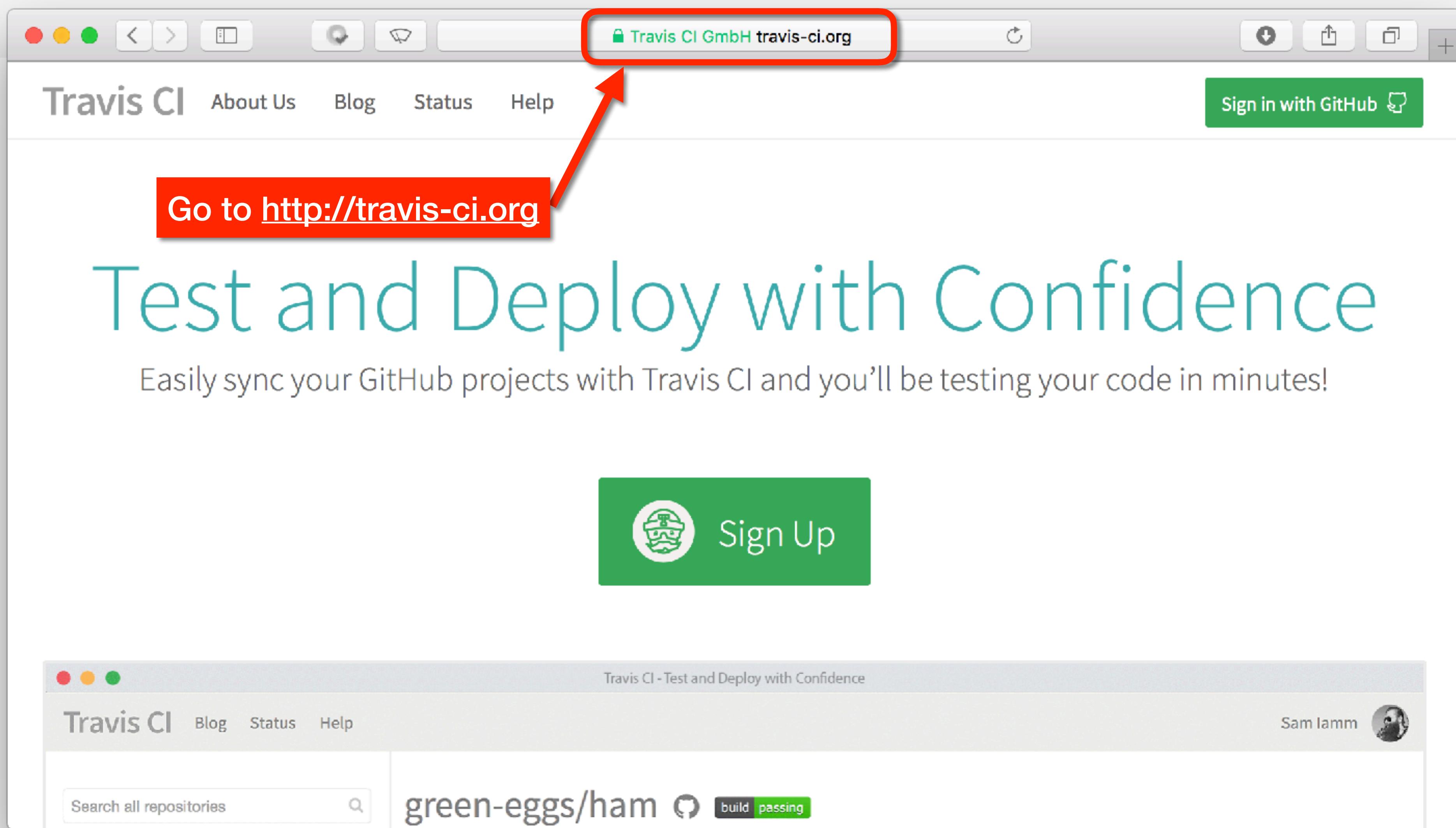
Create a database for your application by adding a line to your `.travis.yml`:

```
.travis.yml
before_script:
- psql -c 'create database travis_ci_test;' -U postgres
```

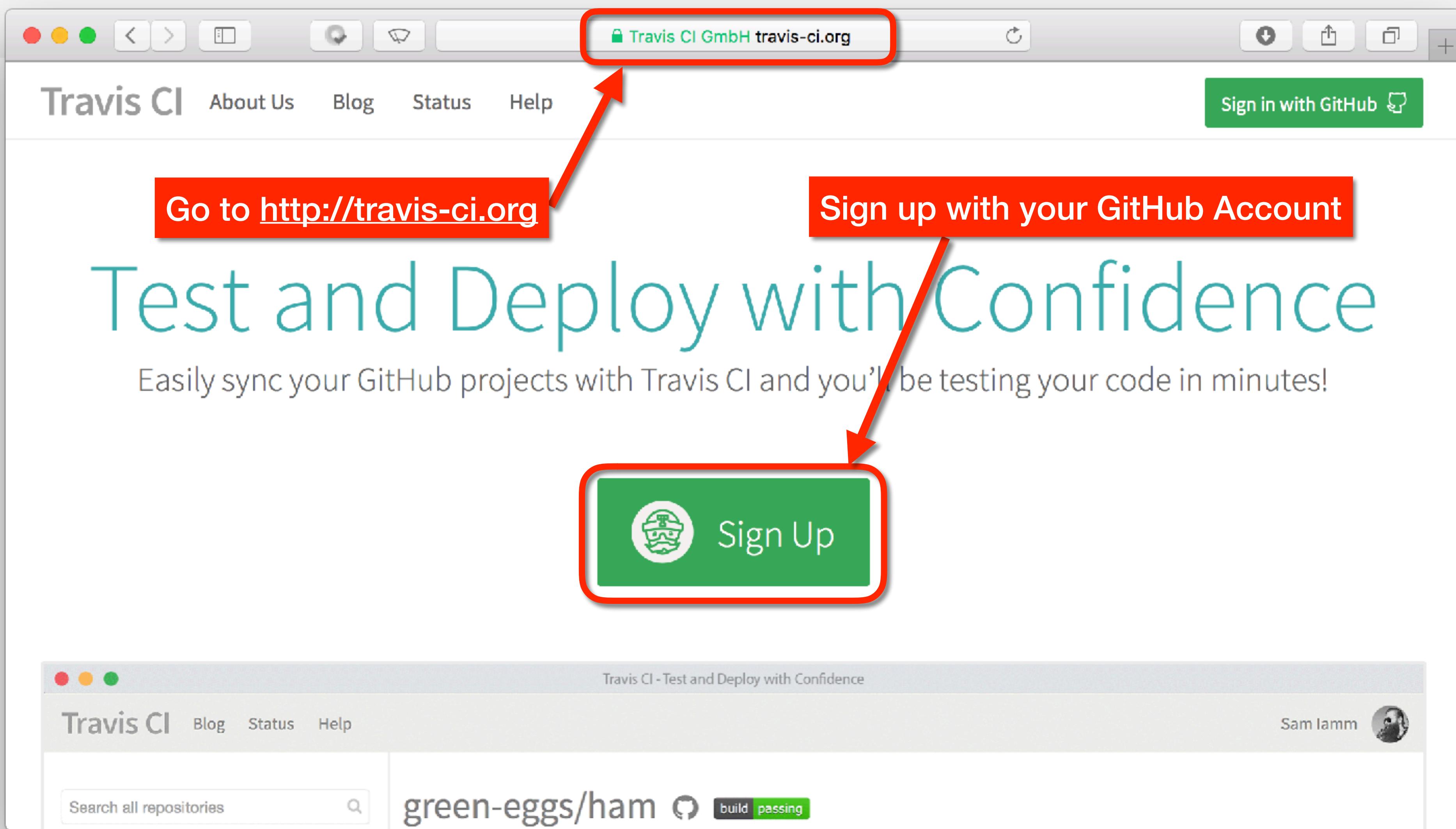
For a Rails application, you can now use the following `database.yml` configuration to access the database locally:

```
database.yml
test:
  adapter: postgresql
  database: travis_ci_test
```

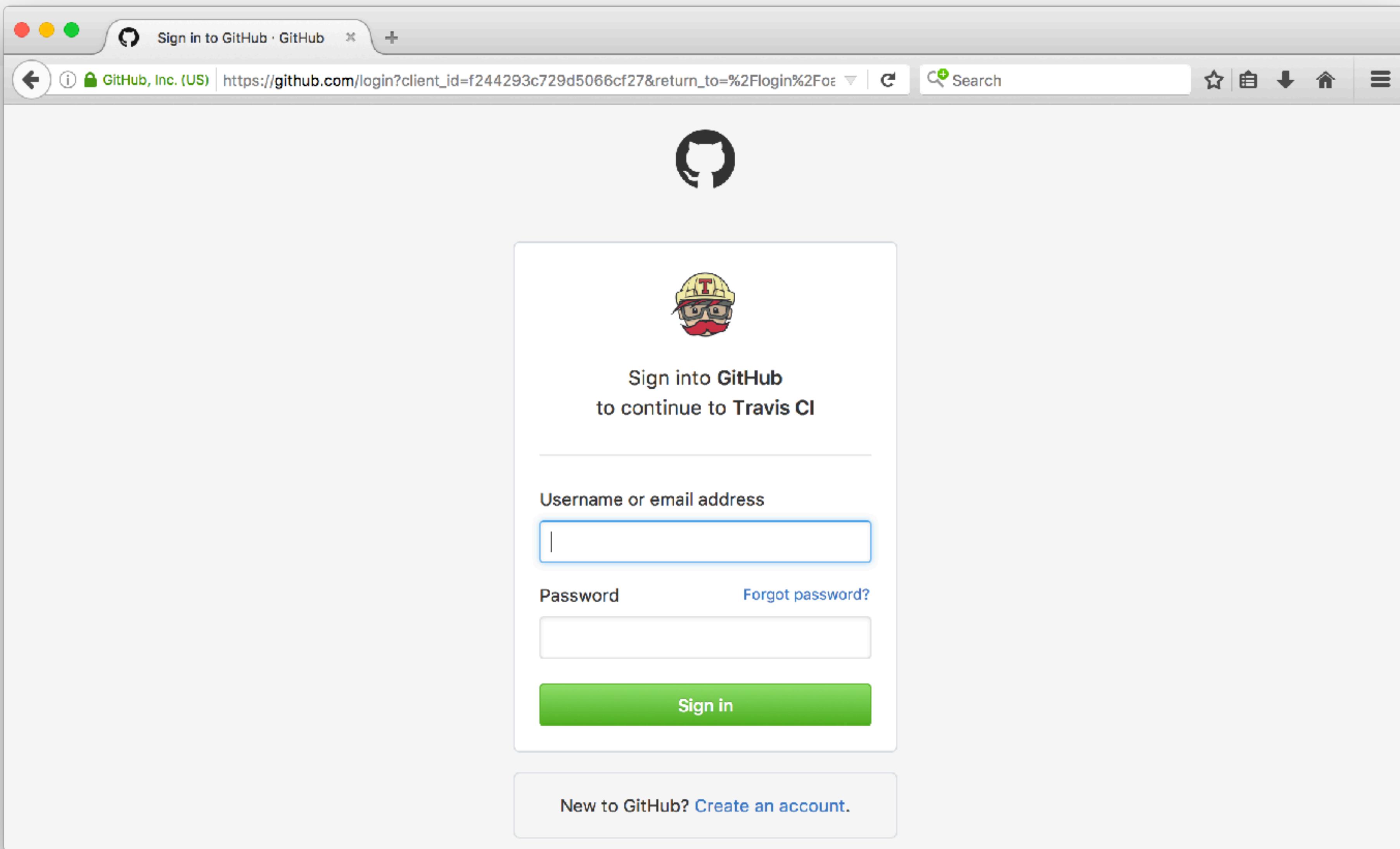
Sign up for Travis CI



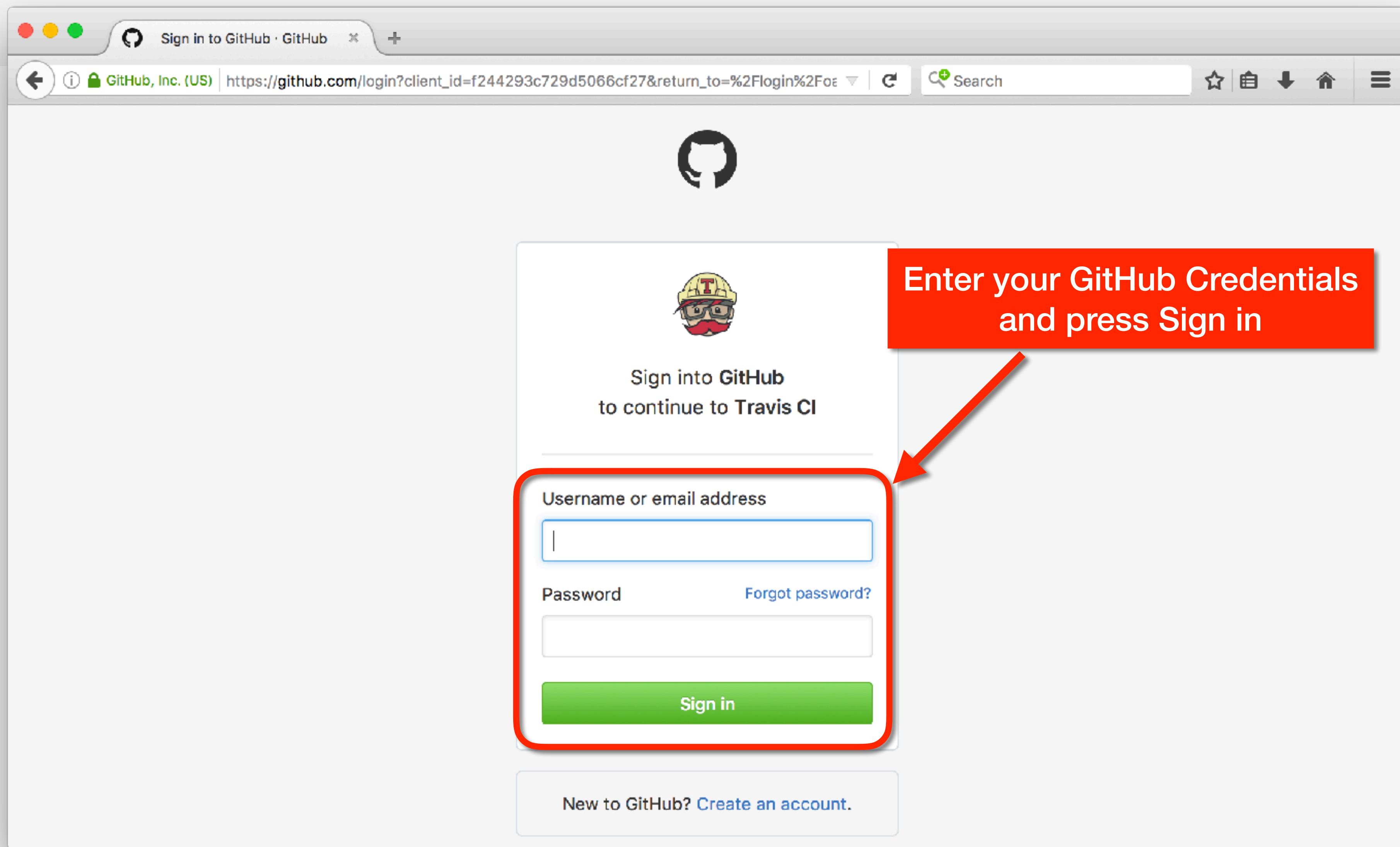
Sign up for Travis CI



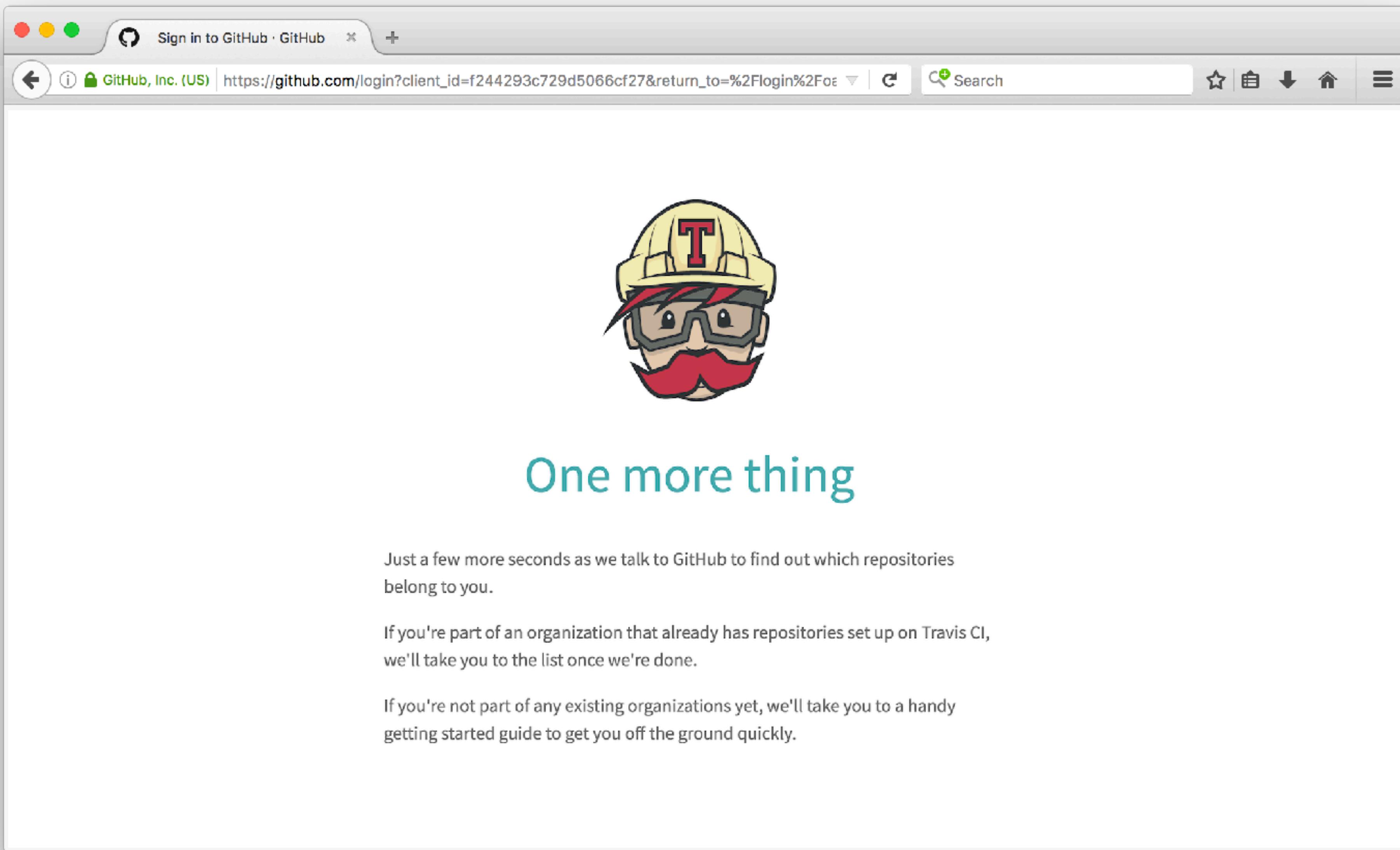
Sign-in Using GitHub



Sign-in Using GitHub



Wait while Syncing...



Just a few more seconds as we talk to GitHub to find out which repositories belong to you.

If you're part of an organization that already has repositories set up on Travis CI, we'll take you to the list once we're done.

If you're not part of any existing organizations yet, we'll take you to a handy getting started guide to get you off the ground quickly.

Add your Repository

The screenshot shows the Travis CI website's 'Getting Started' guide. At the top, there's a search bar and navigation links for 'Blog', 'Status', and 'Help'. On the right, a user profile for 'John Rofrano' is visible. The main content area has a heading 'First time here? Let's get you started!' followed by two numbered steps:

- 1 Activate GitHub Repositories**

Once you're signed in, and we've initially synchronized your repositories from GitHub, go to your [profile](#) page for open source or for your private projects.

You'll see all the organizations you're a member of and all the repositories you have access to. The ones you have administrative access to are the ones you can enable the service hook for.

Flip the switch to on for all repositories you'd like to enable.
- 2 Add .travis.yml file to your repository**

In order for Travis CI to build your project, you need to tell the system a little bit about it. You'll need to add a file named `.travis.yml` to the root of your repository.

If `.travis.yml` is not in the repository, is misspelled or is not valid YAML, Travis CI will ignore it.

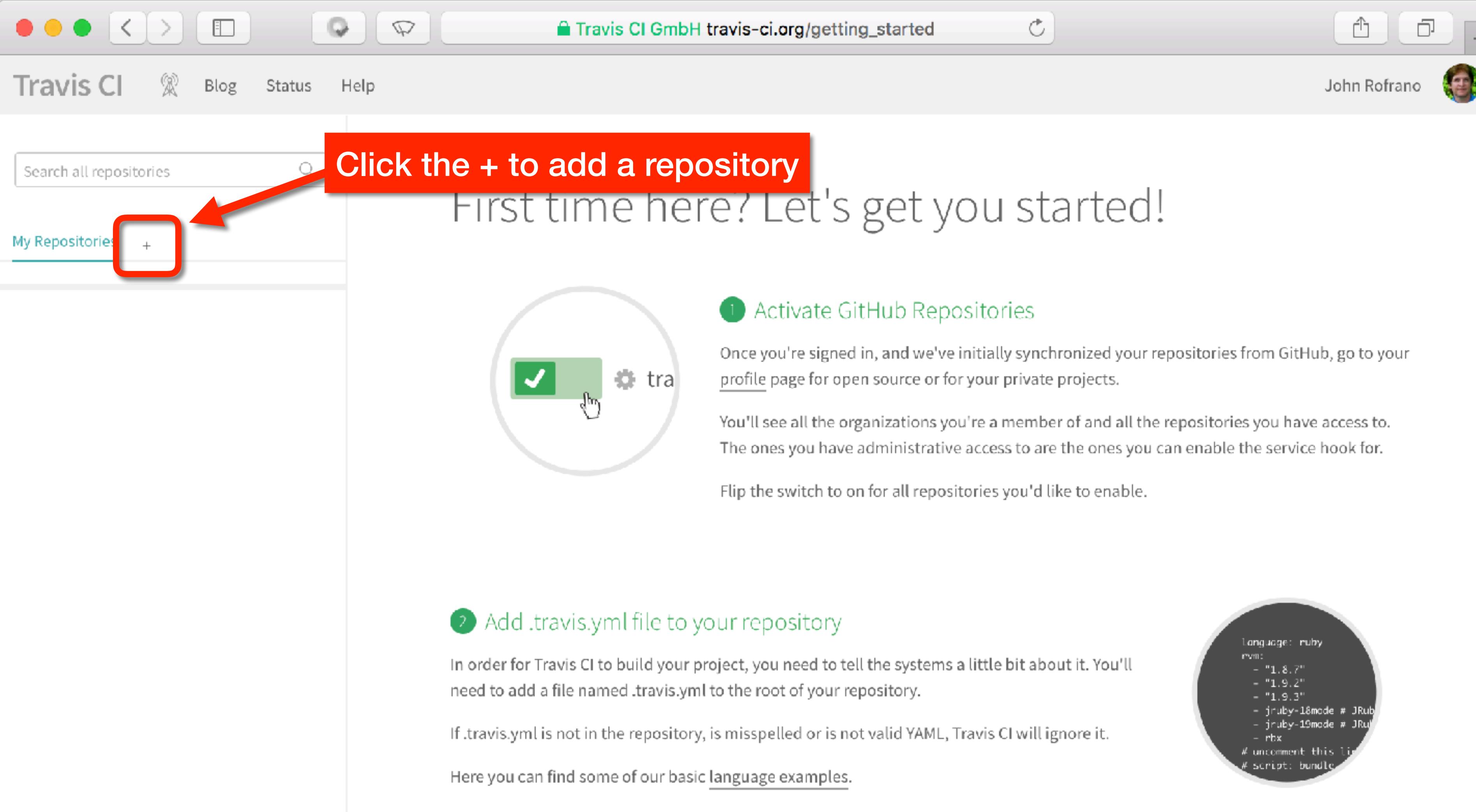
Here you can find some of our basic [language examples](#).

A circular inset on the right shows a snippet of a `.travis.yml` configuration file for Ruby:

```
language: ruby
rvm:
  - "1.8.7"
  - "1.9.2"
  - "1.9.3"
  - jruby-18mode # JRuby
  - jruby-19mode # JRuby
  - rbx
```

uncomment this line
script: bundle

Add your Repository



Click the + to add a repository

First time here? Let's get you started!

1 Activate GitHub Repositories

Once you're signed in, and we've initially synchronized your repositories from GitHub, go to your [profile](#) page for open source or for your private projects.

You'll see all the organizations you're a member of and all the repositories you have access to. The ones you have administrative access to are the ones you can enable the service hook for.

Flip the switch to on for all repositories you'd like to enable.

2 Add .travis.yml file to your repository

In order for Travis CI to build your project, you need to tell the system a little bit about it. You'll need to add a file named `.travis.yml` to the root of your repository.

If `.travis.yml` is not in the repository, is misspelled or is not valid YAML, Travis CI will ignore it.

Here you can find some of our basic [language examples](#).

```
Language: ruby
rvm:
  - "1.8.7"
  - "1.9.2"
  - "1.9.3"
  - jruby-18mode # JRuby
  - jruby-19mode # JRuby
  - rbx

# uncomment this line if you're using rbx
# script: bundle
```

General Layout

The screenshot shows the Travis CI user profile page for John Rofrano. The top navigation bar includes links for Travis CI, Dashboard, Changelog, Documentation, Help, and a user icon. The main content area displays the user's profile picture, name (John Rofrano), handle (@rofrano), and a "Sync account" button. A sidebar on the left features sections for "MY ACCOUNT" (with a "Sync account" button), "A SINGLE PLACE FOR ALL YOUR BUILDS" (with a "Sign up for the beta" button), and "ORGANIZATIONS" (listing "IBM DevOps" and "NYU Devops"). The right side shows a "Legacy Services Integration" section with a "Filter repositories" input field and a list of repositories: "bluemix-starter-code", "devops-workshop", "dockerfiles", and "hitcounter", each with a "Settings" button.

travis-ci.org

Travis CI

Dashboard Changelog Documentation Help

John Rofrano @rofrano

Sync account

Repositories Settings

We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

Legacy Services Integration

Filter repositories

bluemix-starter-code

devops-workshop

dockerfiles

hitcounter

Settings

Settings

Settings

Settings

MISSING AN ORGANIZATION?
Review and add your authorized organizations.

General Layout

The screenshot shows the Travis CI user profile page for John Rofrano. The top navigation bar includes links for Dashboard, Changelog, Documentation, and Help. A red callout box highlights the text "Your personal repositories will be listed here". A red arrow points from this text to the repository list below. Another red box highlights the "Legacy Services Integration" section, which contains a filter input and four repository entries: bluemix-starter-code, devops-workshop, dockerfiles, and hitcounter, each with a settings icon.

MY ACCOUNT

John Rofrano

Sync account

A SINGLE PLACE FOR ALL YOUR BUILDS

You can now have all your public and private repositories together at travis-ci.com

Sign up for the beta

ORGANIZATIONS

IBM DevOps

NYU Devops

MISSING AN ORGANIZATION?

Review and add your authorized organizations.

travis-ci.org

John Rofrano

@rofrano

Your personal repositories will be listed here

Repositories Settings

We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

Legacy Services Integration

Filter repositories

bluemix-starter-code

devops-workshop

dockerfiles

hitcounter

Settings

Settings

Settings

Settings

General Layout

The screenshot shows the Travis CI account settings interface. At the top, there's a navigation bar with links for Dashboard, Changelog, Documentation, and Help. On the right, there's a user profile picture and a dropdown menu.

The main area starts with a "MY ACCOUNT" section containing a profile picture and the name "John Rofrano". Below it is a "Sync account" button. A prominent red callout box with white text says:

If you have a GitHub Organization
Click here to add your organization

A red arrow points from this callout to a red-bordered box labeled "ORGANIZATIONS" which contains two entries: "IBM DevOps" and "NYU Devops".

Below the organizations section, a message says "MISSING AN ORGANIZATION? Review and add your authorized organizations."

To the right, there's a "Legacy Services Integration" section with a "Filter repositories" input field. It lists four repositories with "Settings" buttons:

- bluemix-starter-code
- devops-workshop
- dockerfiles
- hitcounter

Sync if you don't see your new repo

A screenshot of a web browser showing the Travis CI account sync interface. The URL in the address bar is `travis-ci.org`. The page title is "Travis CI". On the left, there's a "MY ACCOUNT" section with a profile picture of "John Rofrano" and a "Sync account" button, which is highlighted with a red box and an arrow pointing to it from the text above. Below this, there's a section titled "A SINGLE PLACE FOR ALL YOUR BUILDS" with text about integrating public and private repositories. A "Sign up for the beta" button is also present. On the right, the user profile "John Rofrano" is shown with the handle "@rofrano". Below the profile, there are tabs for "Repositories" (which is underlined) and "Settings". A message states, "We're only showing your public repositories. You can find your private projects on travis-ci.com". Under "Legacy Services Integration", there's a "Filter repositories" input field and a list of repositories: "bluemix-starter-code", "devops-workshop", "dockerfiles", and "hitcounter", each with a "X" button and a "Settings" button.

Press Sync Account if you don't see your repo

MY ACCOUNT

John Rofrano

Sync account

A SINGLE PLACE FOR ALL YOUR BUILDS

You can now have all your public and private repositories together at travis-ci.com

Sign up for the beta

ORGANIZATIONS

IBM DevOps

NYU Devops

MISSING AN ORGANIZATION?

Review and add your authorized organizations.

John Rofrano
@rofrano

Repositories Settings

We're only showing your public repositories. You can find your private projects on travis-ci.com.

Legacy Services Integration

Filter repositories

bluemix-starter-code

devops-workshop

dockerfiles

hitcounter

X Settings

X Settings

X Settings

X Settings

Turn on the devops-workshop Repo

The screenshot shows the Travis CI account dashboard for John Rofrano (@rofrano). The interface includes a navigation bar with links for Dashboard, Changelog, Documentation, and Help. On the left, there's a 'MY ACCOUNT' section with a profile picture and the name 'John Rofrano'. Below it is a 'Sync account' button. A promotional section for 'A SINGLE PLACE FOR ALL YOUR BUILDS' encourages users to have all their public and private repositories together at travis-ci.com, with a 'Sign up for the beta' button. An 'ORGANIZATIONS' section lists 'IBM DevOps'. A red box highlights the text 'Find your devops-workshop repo' with a red arrow pointing towards the repository list. The main content area displays the 'Repositories' tab, which shows four repositories: 'bluemix-starter-code', 'devops-workshop', 'dockerfiles', and 'hitcounter'. Each repository has a 'Settings' button next to it.

travis-ci.org

Travis CI

Dashboard Changelog Documentation Help

John Rofrano @rofrano

Sync account

A SINGLE PLACE FOR ALL YOUR BUILDS

You can now have all your public and private repositories together at travis-ci.com

Sign up for the beta

ORGANIZATIONS

IBM DevOps

Find your devops-workshop repo

MISSING AN ORGANIZATION?
Review and add your authorized organizations.

Repositories Settings

We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

Legacy Services Integration

Filter repositories

Repository	Settings
bluemix-starter-code	(button)
devops-workshop	(button)
dockerfiles	(button)
hitcounter	(button)

Turn on the devops-workshop Repo

The screenshot shows the Travis CI dashboard for a user named John Rofrano (@rofrano). On the left, there's a sidebar with account information, sync options, and organization links. The main area displays the user's profile picture and name, along with a message about public repositories. Below this, a section titled 'Legacy Services Integration' lists several repositories. One repository, 'devops-workshop', has its switch button highlighted with a red box and a red arrow pointing to it from the text 'Find your devops-workshop repo'. Another red box with the text 'Flick the Switch' points to the same button. The repositories listed are:

- bluemix-starter-code
- devops-workshop
- dockerfiles
- hitcounter

Find your devops-workshop repo → **Flick the Switch**

travis-ci.org

Travis CI Dashboard Changelog Documentation Help

MY ACCOUNT John Rofrano Sync account

A SINGLE PLACE FOR ALL YOUR BUILDS You can now have all your public and private repositories together at travis-ci.com Sign up for the beta

ORGANIZATIONS IBM DevOps

MISSING AN ORGANIZATION? Review and add your authorized organizations.

John Rofrano @rofrano

Repositories Settings

We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

Legacy Services Integration

Filter repositories

bluemix-starter-code

devops-workshop

dockerfiles

hitcounter

Click Gear Icon

The screenshot shows the Travis CI account settings interface. On the left, there's a sidebar with sections for 'MY ACCOUNT' (showing a profile picture of John Rofrano and a 'Sync account' button), 'A SINGLE PLACE FOR ALL YOUR BUILDS' (with a message about public and private repositories), 'ORGANIZATIONS' (listing 'IBM DevOps' and 'NYU Devops'), and 'MISSING AN ORGANIZATION?' (with a link to add authorized organizations). The main area displays the user 'John Rofrano' (@rofrano) with a profile picture. Below the profile are tabs for 'Repositories' (selected) and 'Settings'. A message states: 'We're only showing your public repositories. You can find your private projects on [travis-ci.com](#)'. A red callout box with the text 'Click Settings to go to the Dashboard' has a red arrow pointing to the 'Settings' button for the repository 'devops-workshop'. This repository also has a checked toggle switch next to it. Other repositories listed include 'bluemix-starter-code', 'dockerfiles', and 'hitcounter'.

MY ACCOUNT

John Rofrano

Sync account

A SINGLE PLACE FOR ALL YOUR BUILDS

You can now have all your public and private repositories together at travis-ci.com

Sign up for the beta

ORGANIZATIONS

IBM DevOps

NYU Devops

MISSING AN ORGANIZATION?

Review and add your authorized organizations.

travis-ci.org

Dashboard Changelog Documentation Help

John Rofrano @rofrano

Repositories Settings

We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).

Legacy Services Integration

Click Settings to go to the Dashboard

Filter repositories

bluemix-starter-code

devops-workshop

dockerfiles

hitcounter

Settings

Settings

Settings

Settings

You Control The Build

The screenshot shows the Travis CI web interface. At the top, there's a navigation bar with links for Dashboard, Changelog, Documentation, and Help. On the right side of the header is a user profile picture. Below the header, the repository name "rofrano / devops-workshop" is displayed, along with a "build unknown" status indicator. The main content area has tabs for Current, Branches, Build History, Pull Requests, and Settings, with the Settings tab currently selected. The Settings section is divided into several sections: General, Auto Cancellation, and a section for Continuous Deployment. In the General section, there are two toggle switches: "Build pushed branches" (which is turned on) and "Limit concurrent jobs" (which is turned off). In the Auto Cancellation section, there are two more toggle switches: "Auto cancel branch builds" (which is turned on) and "Auto cancel pull request builds" (which is also turned on). On the left side of the screen, there's a sidebar titled "My Repositories" containing a list of four repositories: "nyu-devops/lab-flask-bdd" (status: green, # 80), "nyu-devops/lab-travis-ci" (status: green, # 38), "rofrano/lab-travis-ci" (status: red, !, # 4), and "nyu-devops/lab-flask-tdd" (status: green, # 10). Each repository entry includes build duration and completion time information.

You Control The Build

The default is Pushes and Pull Requests

My Repositories

Repository	#	Status
✓ nyu-devops/lab-flask-bdd	80	Duration: 1 min 9 sec Finished: 2 months ago
✓ nyu-devops/lab-travis-ci	38	Duration: 47 sec Finished: 3 months ago
! rofrano/lab-travis-ci	4	Duration: 53 sec Finished: 3 months ago
✓ nyu-devops/lab-flask-tdd	10	Duration: 58 sec Finished: 4 months ago

General

Build pushed branches

Build pushed pull requests

Limit concurrent jobs

Auto Cancellation

Auto Cancellation allows you to only run builds for the latest commits in the queue. This setting can be applied to builds for Branch builds and Pull Request builds separately. Builds will only be canceled if they are waiting to run, allowing for any running jobs to finish.

Auto cancel branch builds

Auto cancel pull request builds

You Control The Build

Press the badge

Travis CI Dashboard Changelog Documentation Help

Search all repositories

My Repositories +

Repository	#	Status
nyu-devops/lab-flask-bdd	80	✓
Duration: 1 min 9 sec		
Finished: 2 months ago		
nyu-devops/lab-travis-ci	38	✓
Duration: 47 sec		
Finished: 3 months ago		
! rofrano/lab-travis-ci	4	!
Duration: 53 sec		
Finished: 3 months ago		
nyu-devops/lab-flask-tdd	10	✓
Duration: 58 sec		
Finished: 4 months ago		

rofrano / devops-workshop

build unknown

Current Branches Build History Pull Requests Settings More options

General

Build pushed branches Limit concurrent jobs

Build pushed pull requests

Auto Cancellation

Auto Cancellation allows you to only run builds for the latest commits in the queue. This setting can be applied to builds for Branch builds and Pull Request builds separately. Builds will only be canceled if they are waiting to run, allowing for any running jobs to finish.

Auto cancel branch builds Auto cancel pull request builds

Get your Badge

The screenshot shows the Travis CI web interface with the repository `rofrano / devops-workshop` selected. A modal dialog box is open, titled "Status Image". Inside the dialog, there are fields for "BRANCH" (set to "master") and "Image URL" (containing the URL `https://travis-ci.org/rofrano/devops-workshop.svg?branch=master`). Below the dialog, there is a section about "Auto Cancellation" with two toggle switches: "Auto cancel branch builds" and "Auto cancel pull request builds", both of which are checked.

Travis CI

Dashboard Changelog Documentation Help

Search all repositories

rofrano / devops-workshop build unknown

My Repositories

- ✓ nyu-devops/lab-flask-bdd # 80
Duration: 1 min 9 sec
Finished: 2 months ago
- ✓ nyu-devops/lab-travis-ci # 38
Duration: 47 sec
Finished: 3 months ago
- ! rofrano/lab-travis-ci # 4
Duration: 53 sec
Finished: 3 months ago
- ✓ nyu-devops/lab-flask-tdd # 10
Duration: 58 sec
Finished: 4 months ago

Status Image

BRANCH

master

Image URL

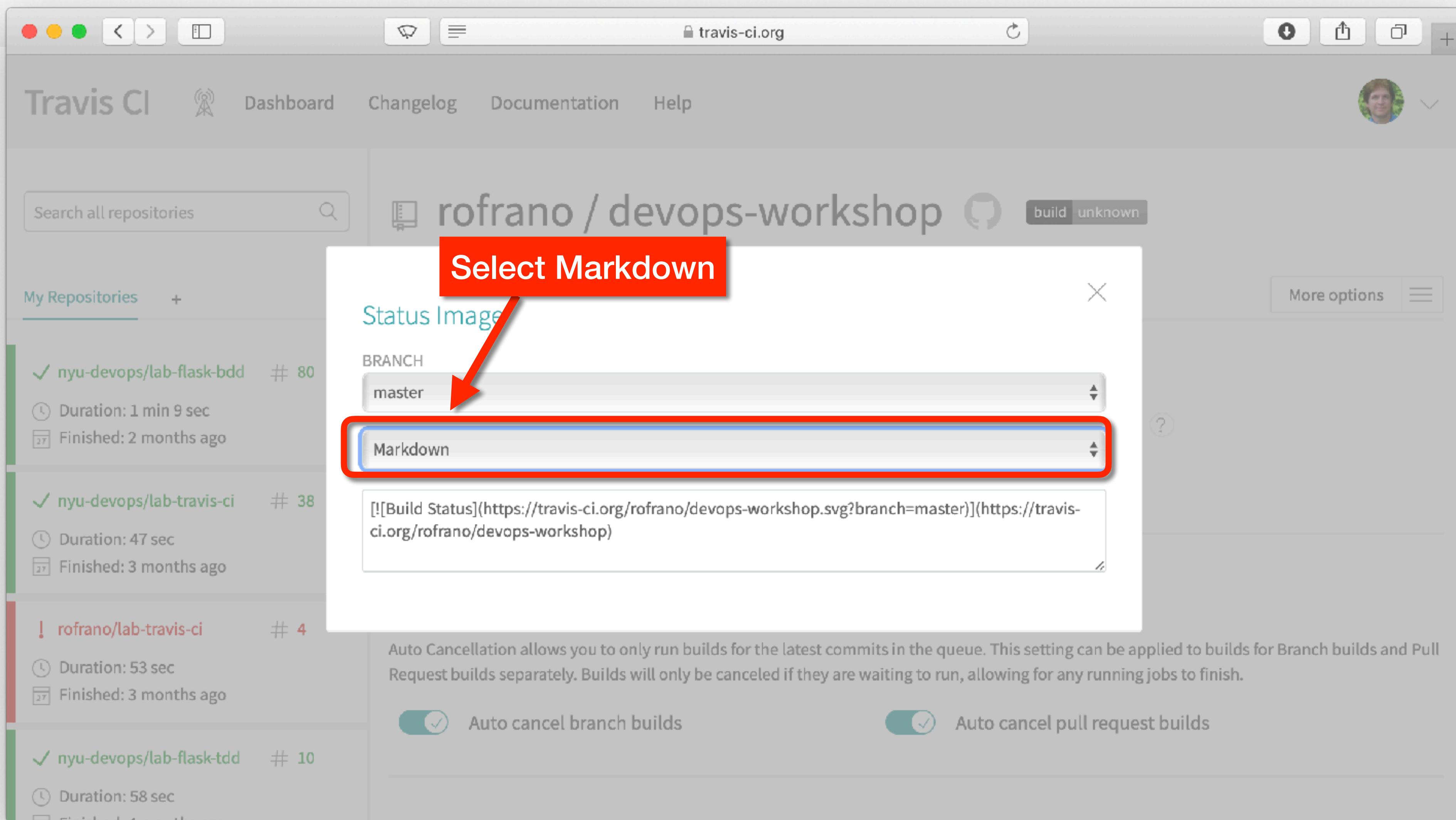
`https://travis-ci.org/rofrano/devops-workshop.svg?branch=master`

Auto Cancellation allows you to only run builds for the latest commits in the queue. This setting can be applied to builds for Branch builds and Pull Request builds separately. Builds will only be canceled if they are waiting to run, allowing for any running jobs to finish.

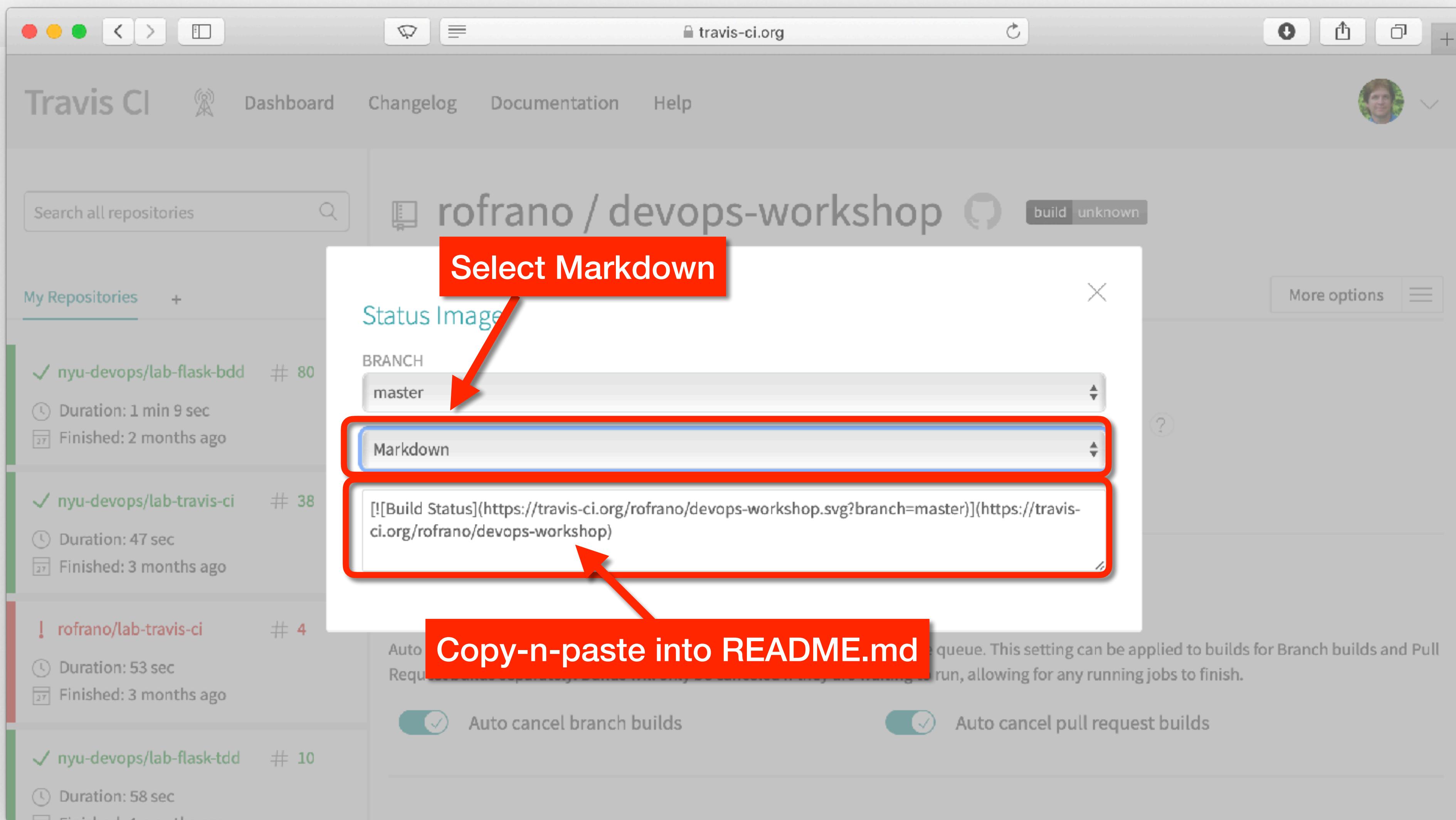
Auto cancel branch builds

Auto cancel pull request builds

Get your Badge



Get your Badge



Go To Your Repo

The screenshot shows a GitHub repository page for a project titled "DevOps Workshop Training that uses Vagrant, VirtualBox, Docker, and Minikube". The repository has 8 commits, 1 branch, 0 releases, 1 contributor, and is licensed under Apache-2.0. The latest commit was made 1 hour ago by user rofrano, changing the database URI default. The commit history lists various files being added or modified, such as .coveragerc, .gitignore, LICENSE, Procfile, README.md, Vagrantfile, and requirements.txt, along with initial app configurations and vagrant command expansions.

File / Action	Description	Time Ago
.coveragerc	added tests	16 hours ago
.gitignore	initial app	16 hours ago
LICENSE	Initial commit	11 days ago
Procfile	initial app	16 hours ago
README.md	Expanded on using vagrant commands	10 days ago
Vagrantfile	changed database uri default	1 hour ago
requirements.txt	initial app	16 hours ago
lesson 1 flask	lesson 1 flask	16 hours ago
service	changed database uri default	1 hour ago
tests	added tests	16 hours ago
lessons/01-flask		16 hours ago

Go To Your Repo

DevOps Workshop Training that uses Vagrant, VirtualBox, Docker, and Minikube

Manage topics

8 commits 1 branch 0 releases 1 contributor Apache-2.0

Branch: master New pull request Create new file Upload files Find File Clone or download

rofrano changed database uri default Latest commit 931bcc3 1 hour ago

lessons/01-flask lesson 1 flask 16 hours ago

service changed database uri default 1 hour ago

tests 16 hours ago

.coveragerc 16 hours ago

.gitignore initial app 16 hours ago

LICENSE Initial commit 11 days ago

Procfile initial app 16 hours ago

README.md Expanded on using vagrant commands 10 days ago

Vagrantfile changed database uri default 1 hour ago

requirements.txt initial app 16 hours ago

Edit

Edit the README.md to add the badge

Edit the Readme.md

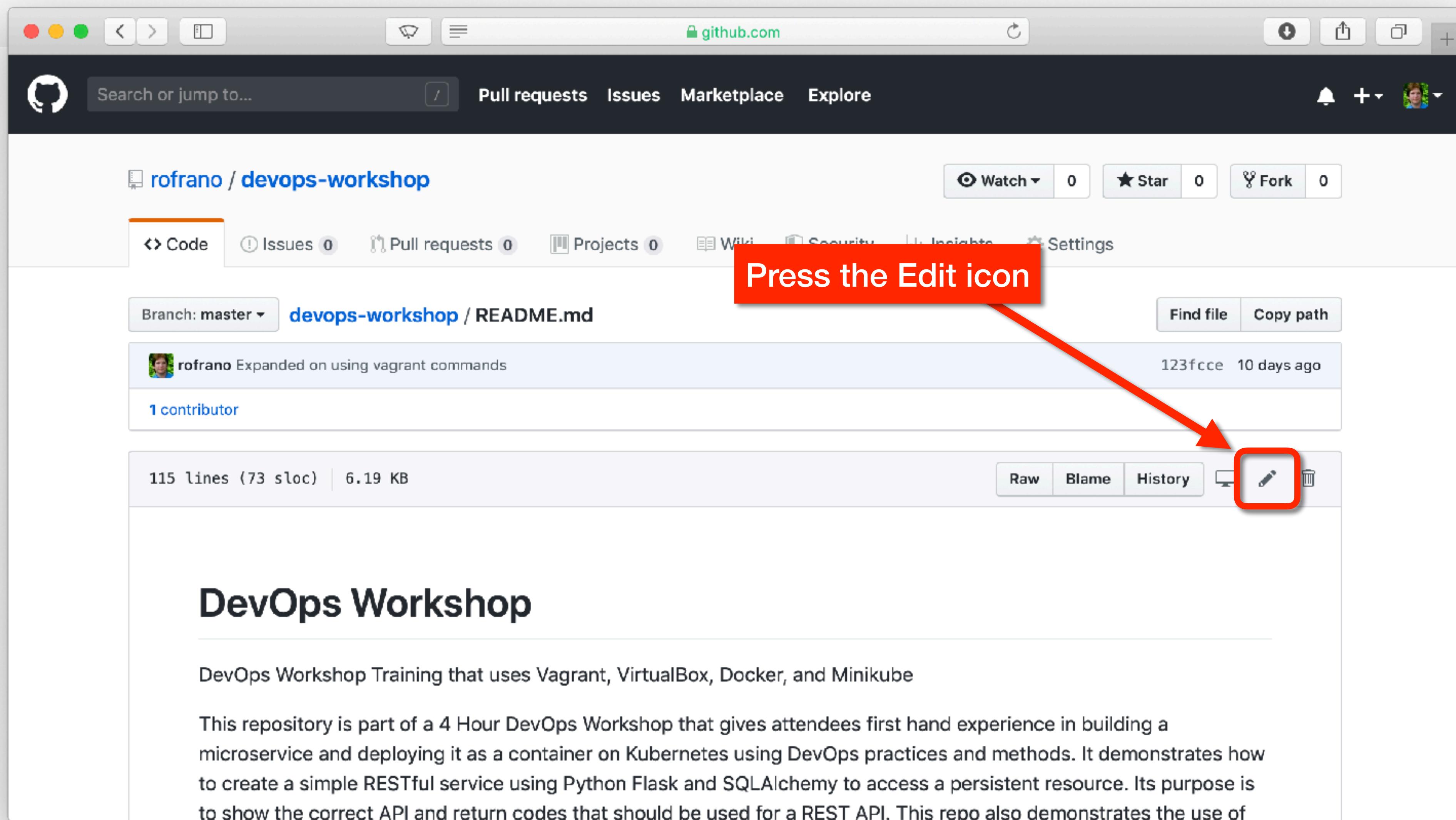
The screenshot shows a GitHub repository page for 'rofrano / devops-workshop'. The page includes a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there are buttons for Watch (0), Star (0), and Fork (0). The main content area displays the README.md file, which contains the following text:

```
DevOps Workshop

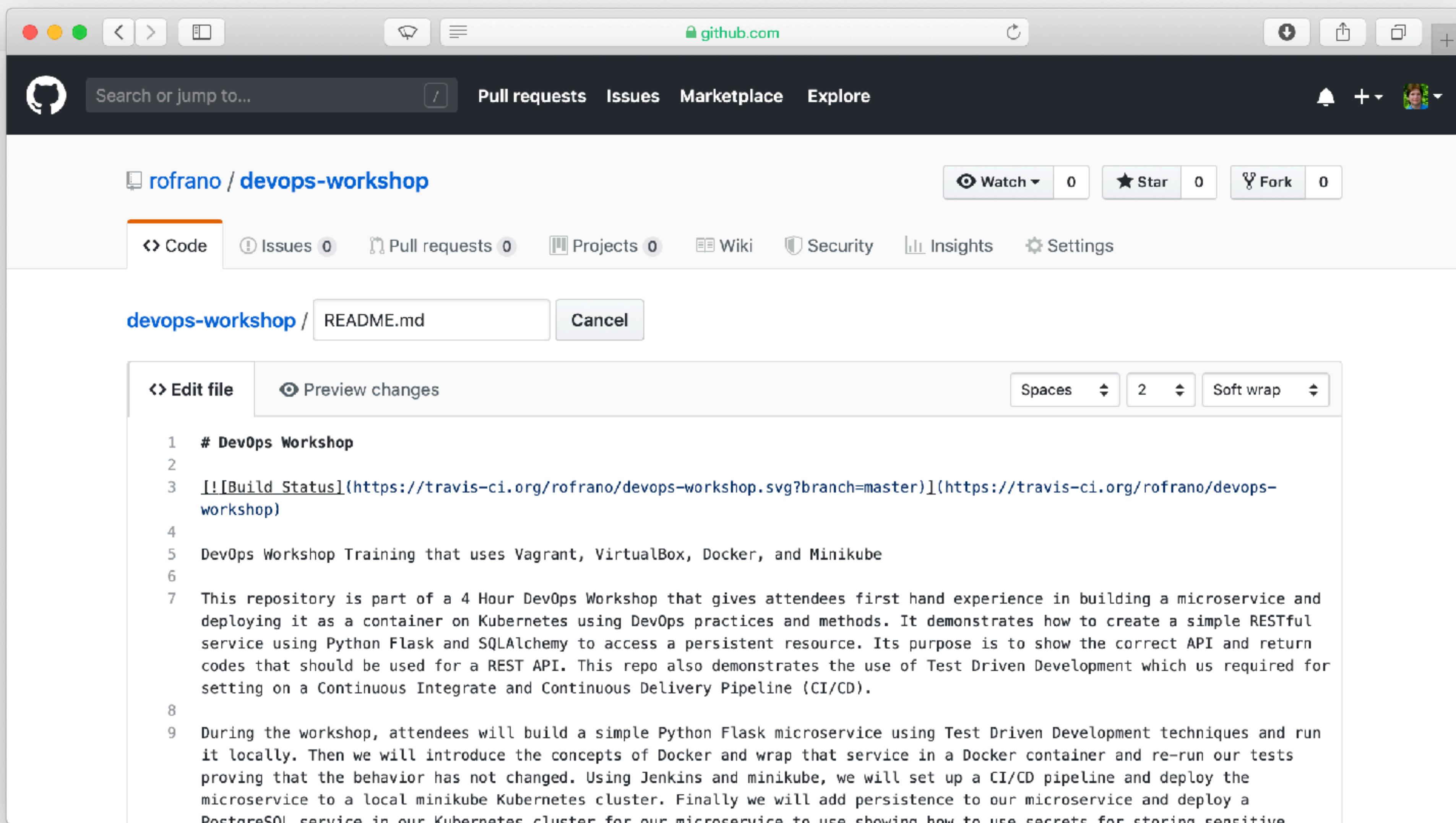
DevOps Workshop Training that uses Vagrant, VirtualBox, Docker, and Minikube

This repository is part of a 4 Hour DevOps Workshop that gives attendees first hand experience in building a microservice and deploying it as a container on Kubernetes using DevOps practices and methods. It demonstrates how to create a simple RESTful service using Python Flask and SQLAlchemy to access a persistent resource. Its purpose is to show the correct API and return codes that should be used for a REST API. This repo also demonstrates the use of
```

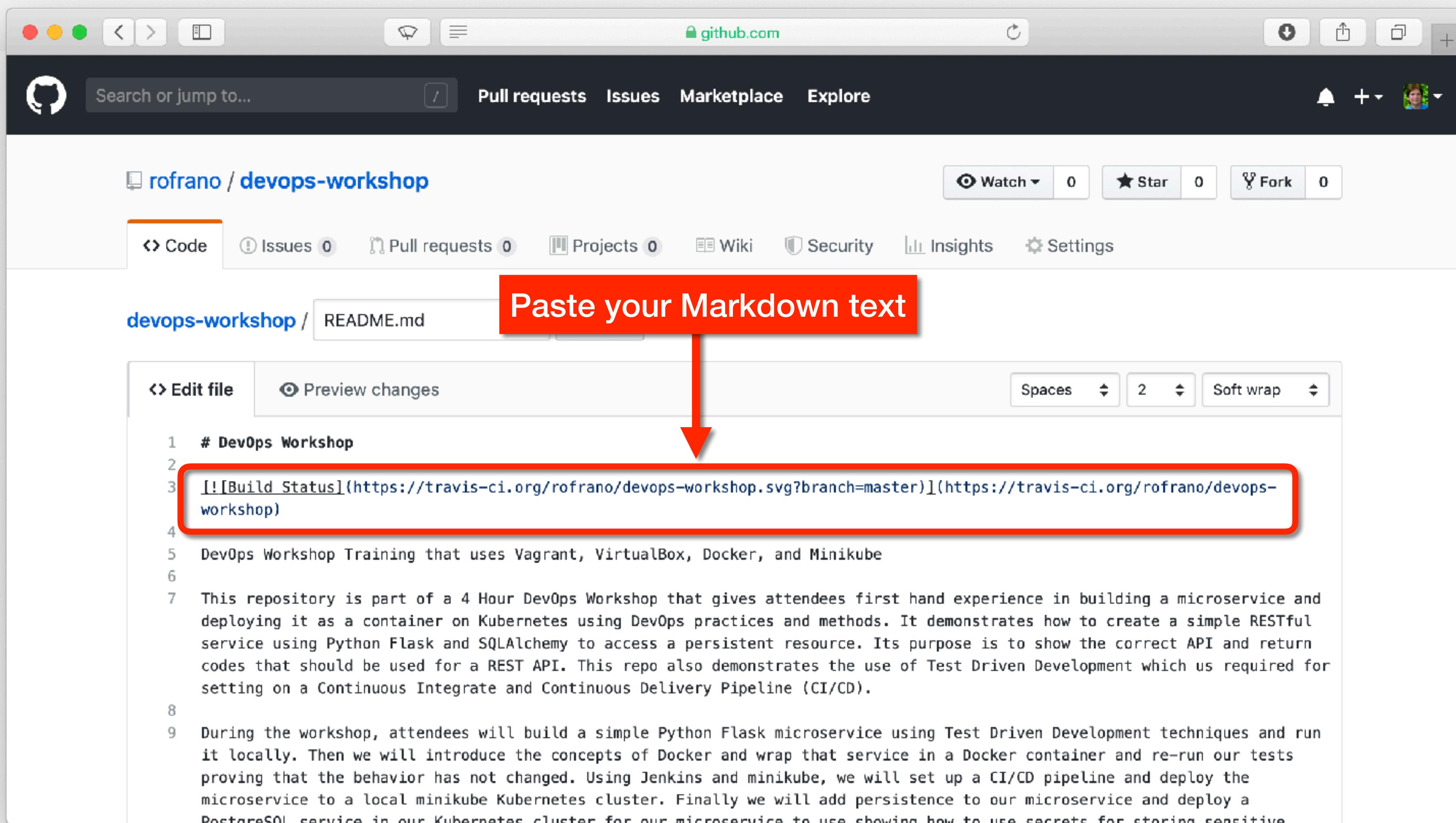
Edit the Readme.md



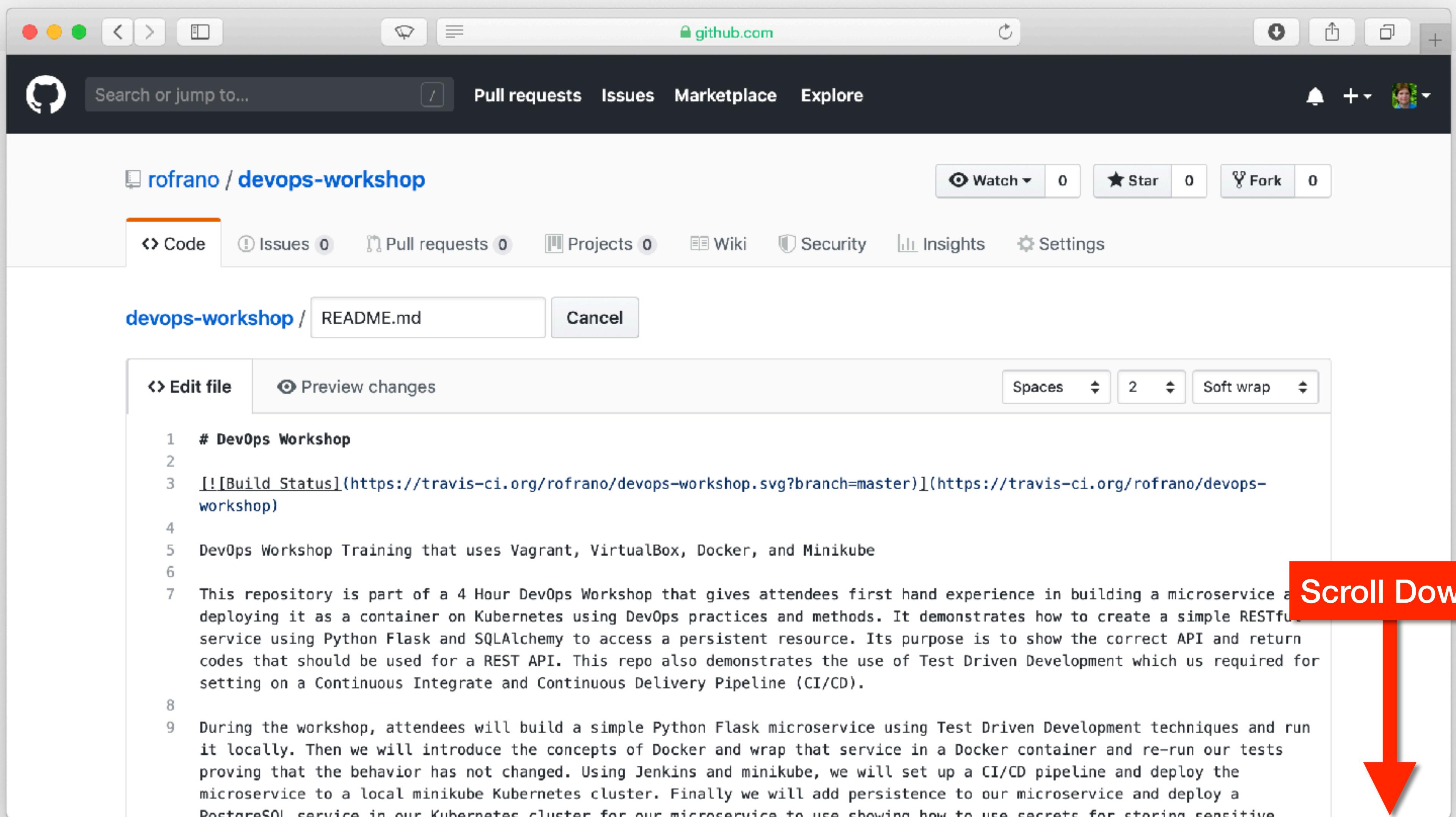
Paste the Badge Markdown



Paste the Badge Markdown



Paste the Badge Markdown



Commit the Change

A screenshot of a GitHub commit dialog box. The background shows a portion of a README.md file with code comments. The dialog box itself has a title "Commit changes". It contains two sections: "Update README.md" which lists "Added Travis CI badge markdown to README.md", and "Create a new branch for this commit and start a pull request." with a link "Learn more about pull requests.". Below this, there are two radio button options: "Commit directly to the master branch." (unchecked) and "Create a new branch for this commit and start a pull request." (checked). The checked option is associated with a branch name "rofrano-patch-1". At the bottom of the dialog are two buttons: "Propose file change" (green) and "Cancel".

113
114 This shows that I have a vagrant VM defined under `/Users/rofrano/GitHub/devops-workshop` and that it is currently in the `poweroff` state.
115
116 **## Copyright**
117 (c) 2019 John Rofrano, All Rights Reserved
118

Commit changes

Update README.md

Added Travis CI badge markdown to README.md

Commit directly to the `master` branch.

Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

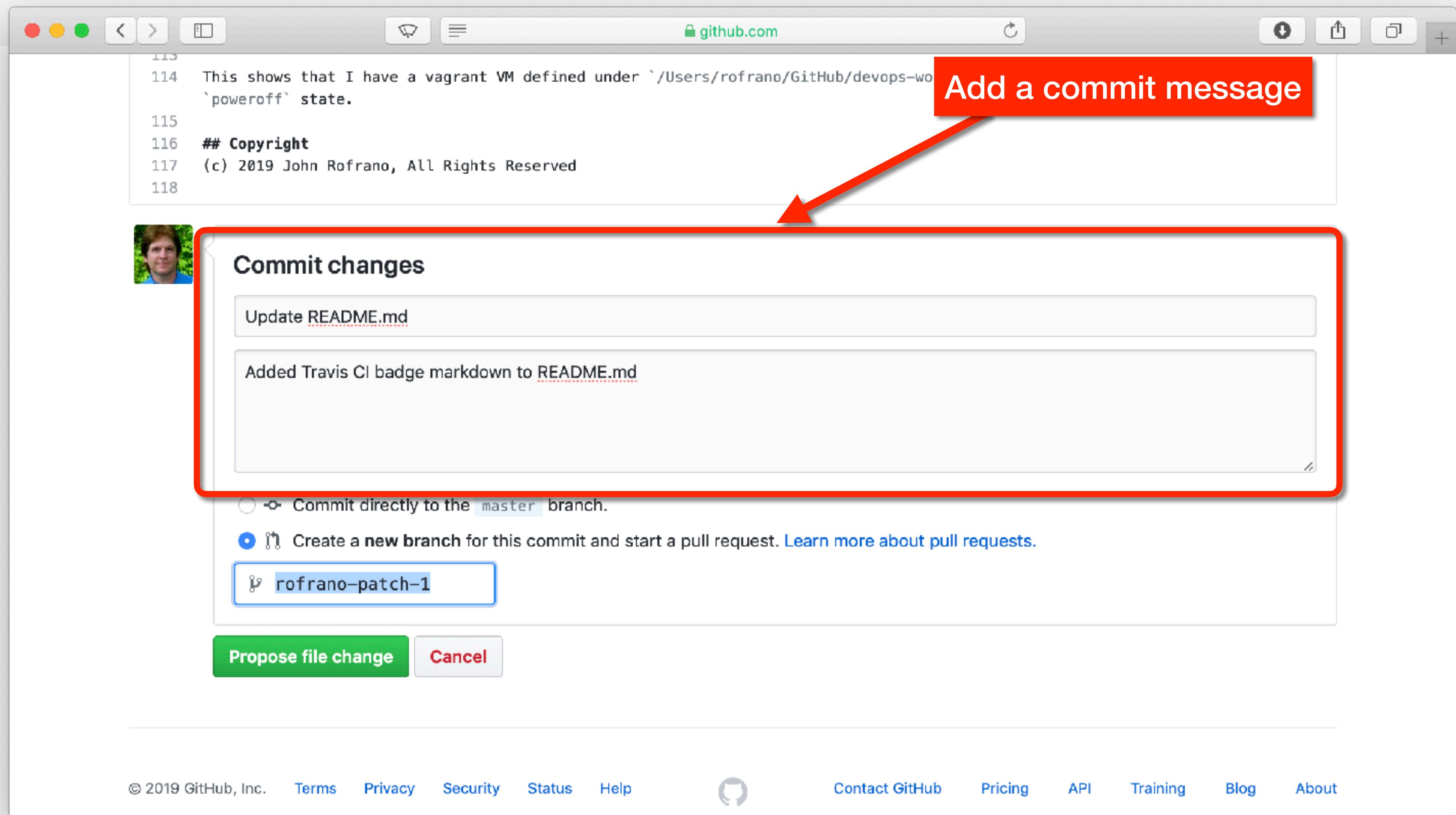
`rofrano-patch-1`

Propose file change **Cancel**

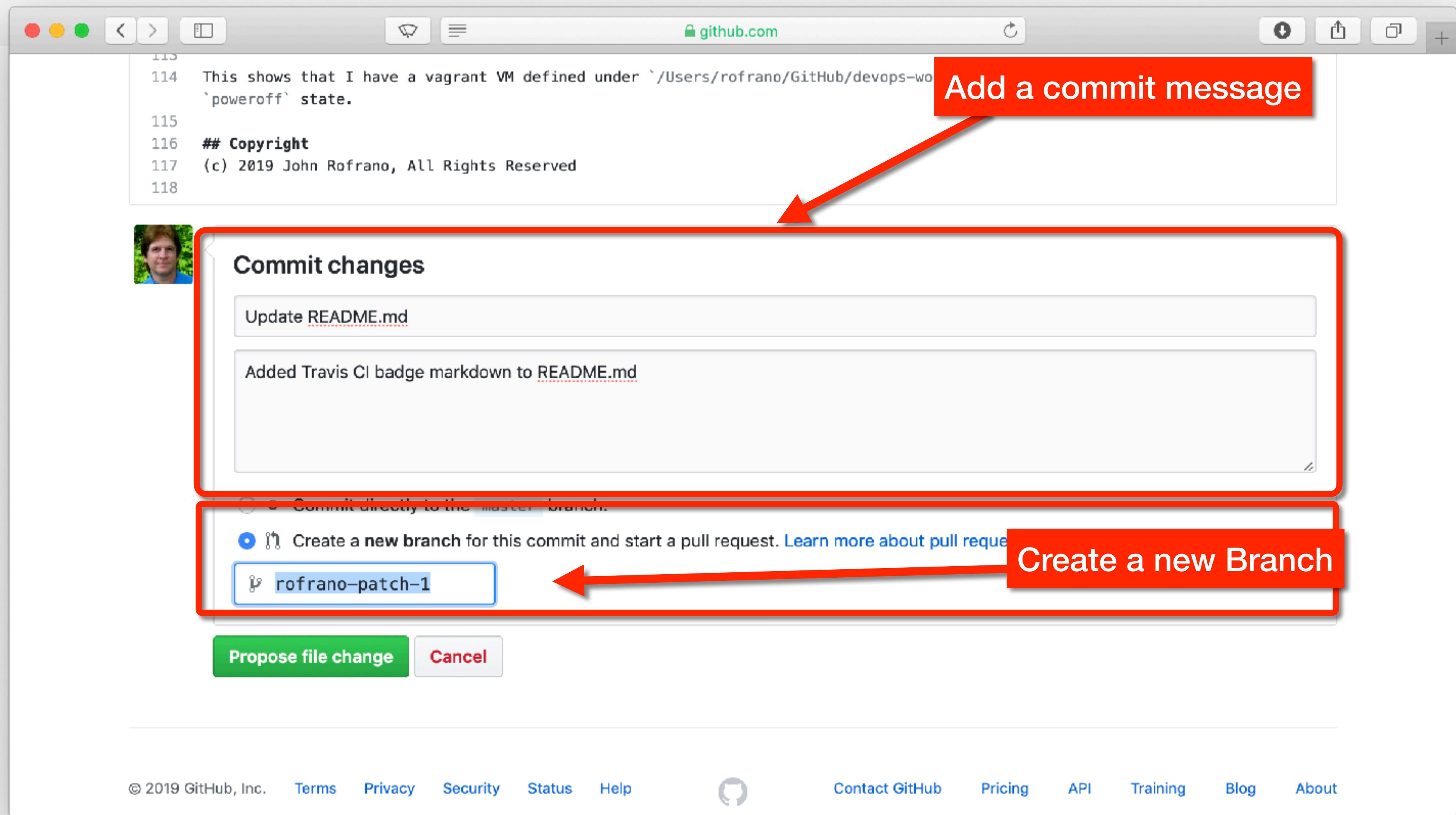
© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

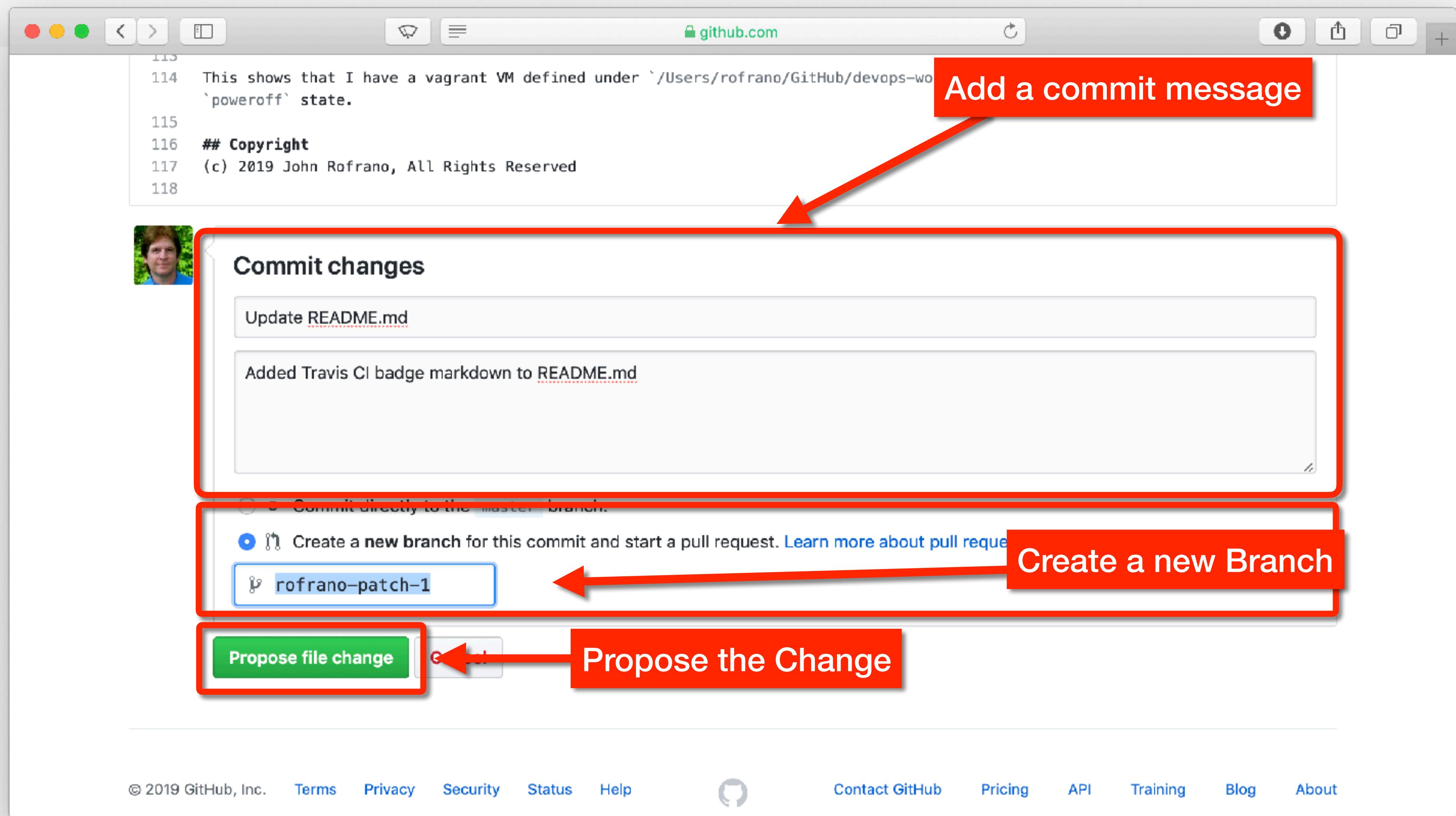
Commit the Change



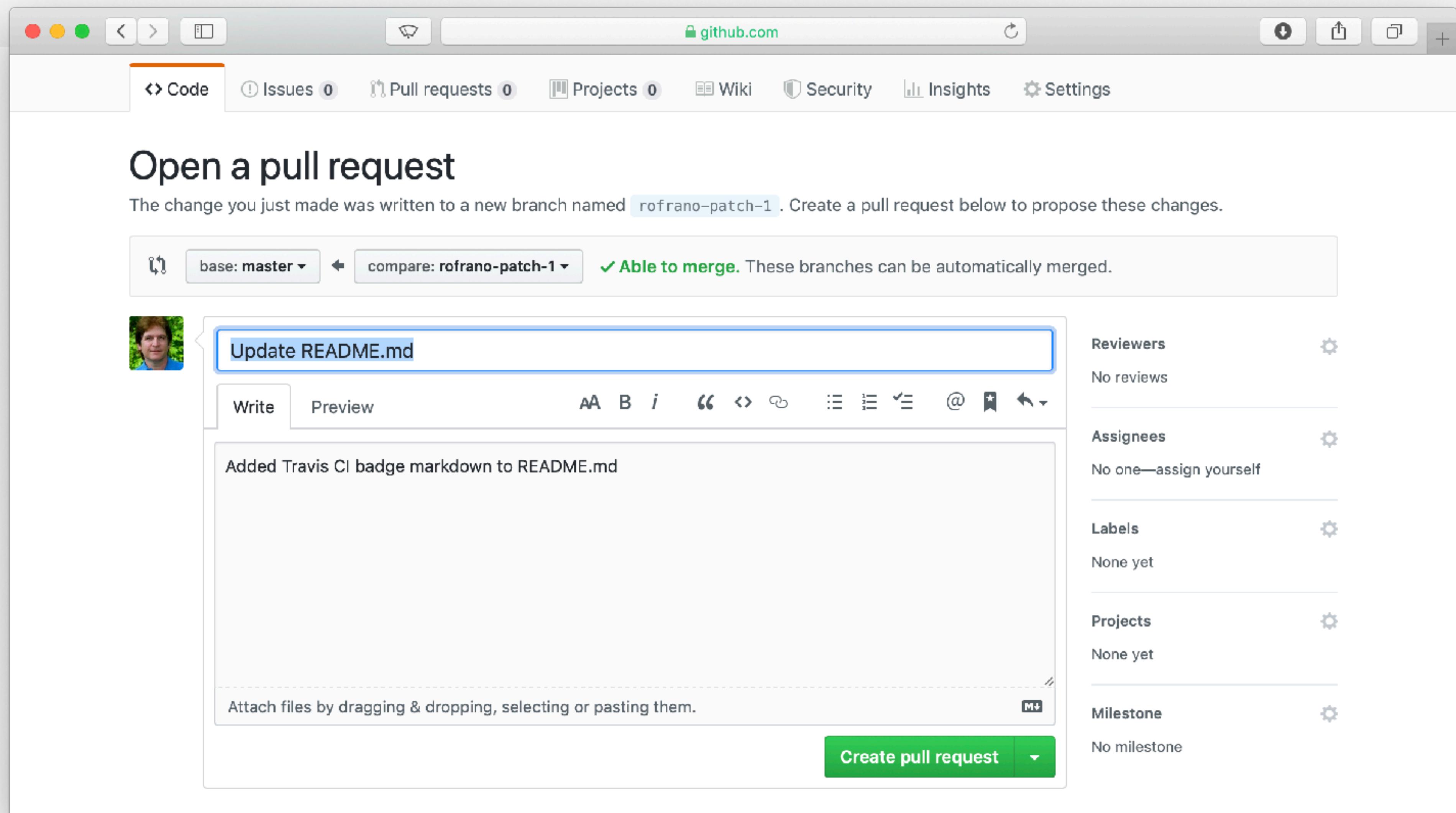
Commit the Change



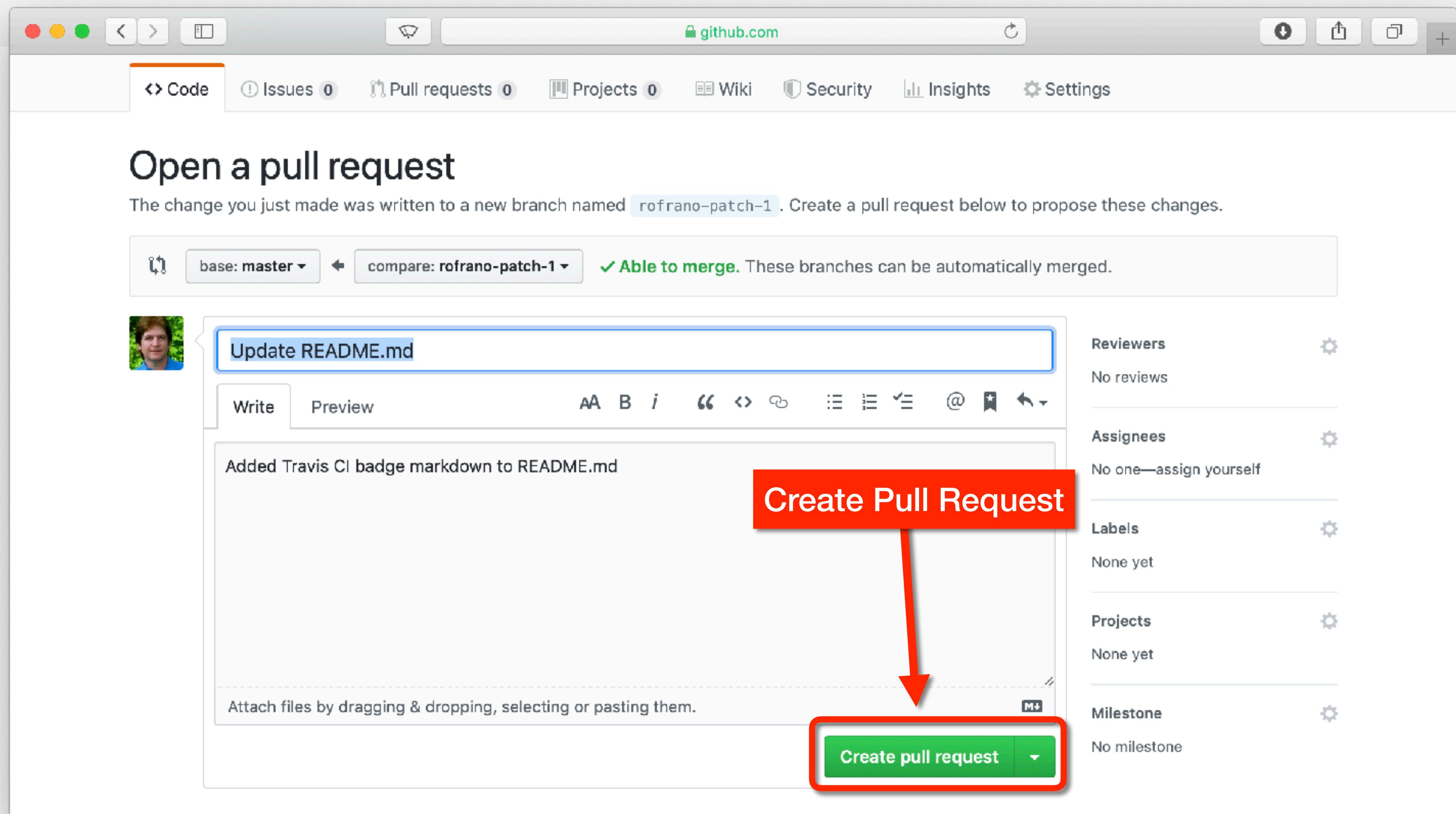
Commit the Change



Create Pull Request



Create Pull Request



Tests are Running

The screenshot shows a GitHub pull request interface. At the top, it says "Update README.md #1" and "rofrano wants to merge 4 commits into master from rofrano-patch-1". A red callout box with the text "The tests are running on Travis CI" has an arrow pointing to the Travis CI status section. This section shows two pending checks: "continuous-integration/travis-ci/pr" and "continuous-integration/travis-ci/push", both labeled "Pending — The Travis CI build is in prog...". Below this, a green checkmark indicates "This branch has no conflicts with the base branch" and "Merging can be performed automatically." On the right side, there are settings for assignees, labels, projects, milestones, and notifications. At the bottom, there's a "Merge pull request" button and a note about opening in GitHub Desktop or viewing command line instructions.

The tests are running on Travis CI

Add more commits by pushing to the [rofrano-patch-1](#) branch on [rofrano/devops-workshop](#).

Some checks haven't completed yet

2 pending checks

- continuous-integration/travis-ci/pr Pending — The Travis CI build is in prog...
- continuous-integration/travis-ci/push Pending — The Travis CI build is in pr...

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Tests are Running

The screenshot shows a GitHub pull request interface. At the top, it says "Update README.md #1" and "rofrano wants to merge 4 commits into master from rofrano-patch-1". A red callout box with white text says "The tests are running on Travis CI". Below the commit history, there's a note: "Add more commits by pushing to the rofrano-patch-1 branch on rofrano/devops-workshop." A red arrow points from this note to a yellow icon of a branch. A larger red box highlights the Travis CI status section, which shows "Some checks haven't completed yet" with "2 pending checks". It lists two items: "continuous-integration/travis-ci/pr Pending — The Travis CI build is in prog..." and "continuous-integration/travis-ci/pr Pending — The Travis CI build is in prog...". A red arrow points from a red callout box with white text "Click on Details to go to Travis-CI and see how it's building" to the "Details" link next to the second pending check. On the right side of the screen, there are settings for assignees, labels, projects, milestones, and notifications.

The tests are running on Travis CI

Add more commits by pushing to the [rofrano-patch-1](#) branch on [rofrano/devops-workshop](#).

Some checks haven't completed yet
2 pending checks

- continuous-integration/travis-ci/pr Pending — The Travis CI build is in prog...
- continuous-integration/travis-ci/pr Pending — The Travis CI build is in prog...

This branch has
Merging can be performed

Merge pull request

You can also [open this in GitHub Desktop](#) or view command line instructions.

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Customize

Unsubscribe

You're receiving notifications because you authored the thread.

1 participant

Details

Details

Check back @ travis-ci.org

The screenshot shows the Travis CI web interface. At the top, there's a navigation bar with links for Dashboard, Changelog, Documentation, and Help. On the right, there's a user profile picture and a search bar labeled "Search all repositories". Below the navigation, the main content area shows the repository "rofrano / devops-workshop". The repository page has tabs for Current, Branches, Build History, and Pull Requests, with "Current" being active. A pull request titled "Pull Request #1 Update README.md" is shown, with status "#3 started" and "Running for 4 sec". The commit message is "Commit a2e009b #1: Update README.md". The branch is "master". The author is listed as "John Rofrano". Below the pull request, it says "Python: 3.7". On the left sidebar, under "My Repositories", there are four entries: "rofrano/devops-workshop" (Duration: 4 sec), "nyu-devops/lab-flask-bdd" (Duration: 1 min 9 sec, Finished: 2 months ago), "nyu-devops/lab-travis-ci" (Duration: 47 sec, Finished: 3 months ago), and "rofrano/lab-travis-ci" (Duration: 53 sec, Finished: 3 months ago). At the bottom, there are buttons for "Job log" and "View config", and options to "Remove log" or "Raw log".

Green When Successful

The screenshot shows the Travis CI web interface. The top navigation bar includes links for Dashboard, Changelog, Documentation, Help, and a user profile. The main header displays the repository name `rofrano / devops-workshop` and a build status of "build unknown".

The left sidebar lists "My Repositories" with the following details:

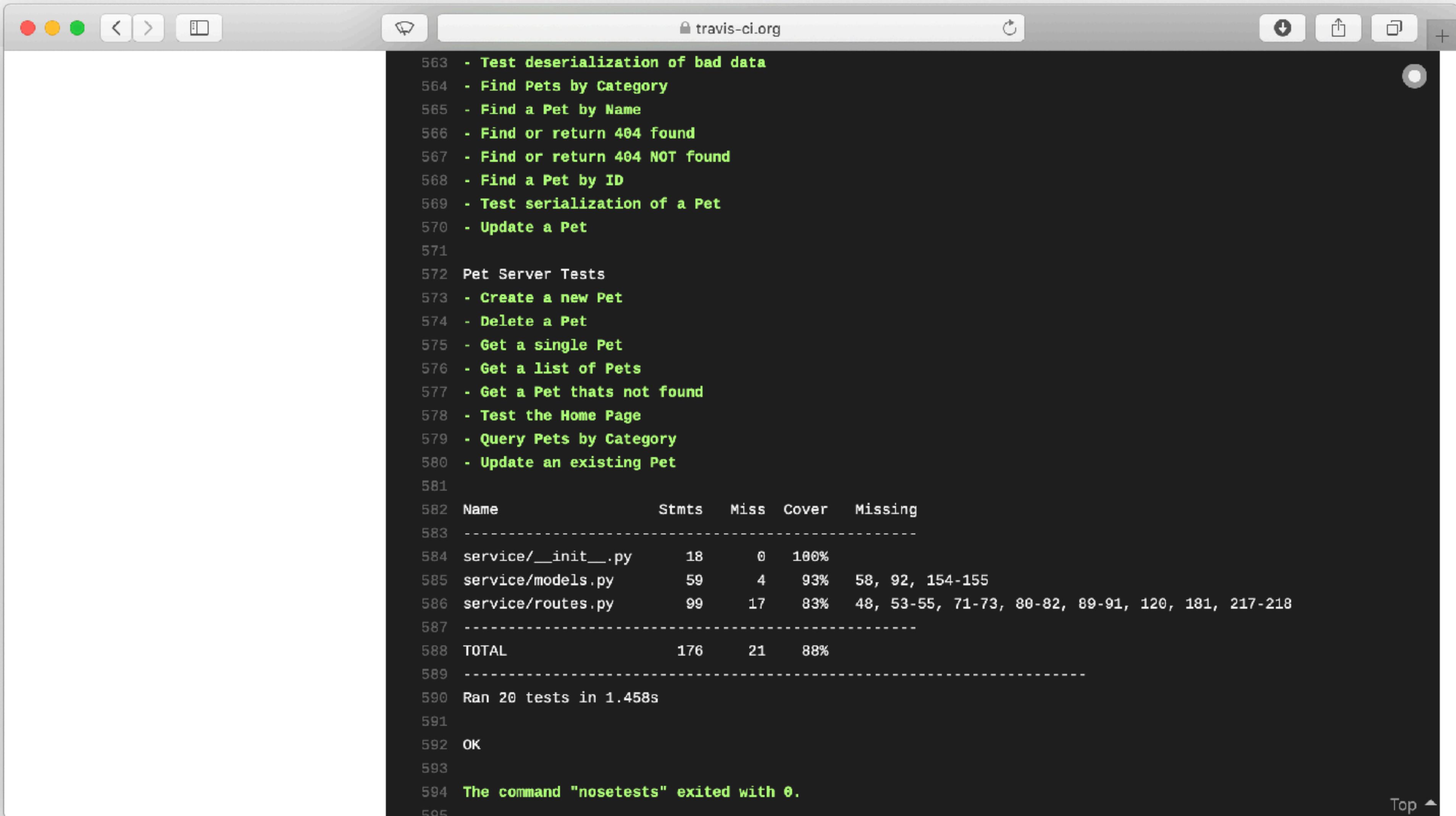
- `rofrano/devops-workshop` # 5: Duration: 43 sec, Finished: 18 minutes ago (Status: green)
- `nyu-devops/lab-flask-bdd` # 80: Duration: 1 min 9 sec, Finished: 2 months ago (Status: green)
- `nyu-devops/lab-travis-ci` # 38: Duration: 47 sec, Finished: 3 months ago (Status: green)
- `rofrano/lab-travis-ci` # 4: Duration: 53 sec, Finished: 3 months ago (Status: red)

The right panel shows the "Current" tab selected, displaying the details of the most recent build for Pull Request #1:

- Pull Request #1 Update README.md**: Status: **#5 passed**, Duration: 43 sec, Ran for 43 sec, Finished: 18 minutes ago.
- Commit: `4c2f403`
- Branch: master
- Author: John Rofrano
- Language: Python: 3.6

At the bottom, there are links for "Job log" and "View config", and buttons for "Remove log" and "Raw log".

Scroll Down to see the Log

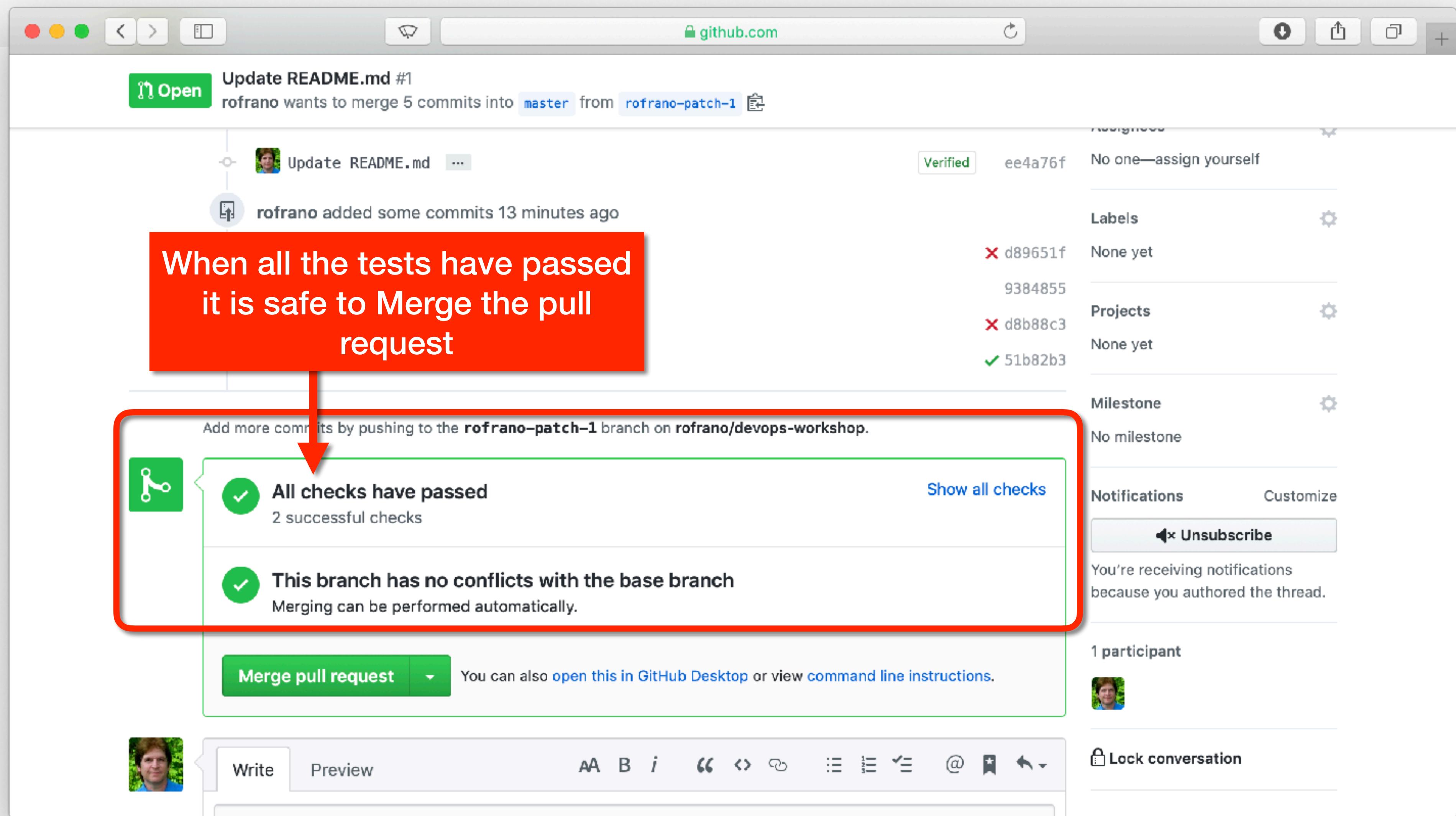
A screenshot of a web browser window titled "travis-ci.org". The main content area displays a terminal log from a nosetests run. The log shows test cases for Pet Server Tests, including serialization and deserialization of bad data, finding pets by category, name, ID, and returning 404 errors. It also includes tests for creating, deleting, getting, listing, and updating pets. Coverage analysis is provided for three files: service/__init__.py, service/models.py, and service/routes.py. The coverage table is as follows:

Name	Stmts	Miss	Cover	Missing
service/__init__.py	18	0	100%	
service/models.py	59	4	93%	58, 92, 154-155
service/routes.py	99	17	83%	48, 53-55, 71-73, 80-82, 89-91, 120, 181, 217-218
TOTAL	176	21	88%	

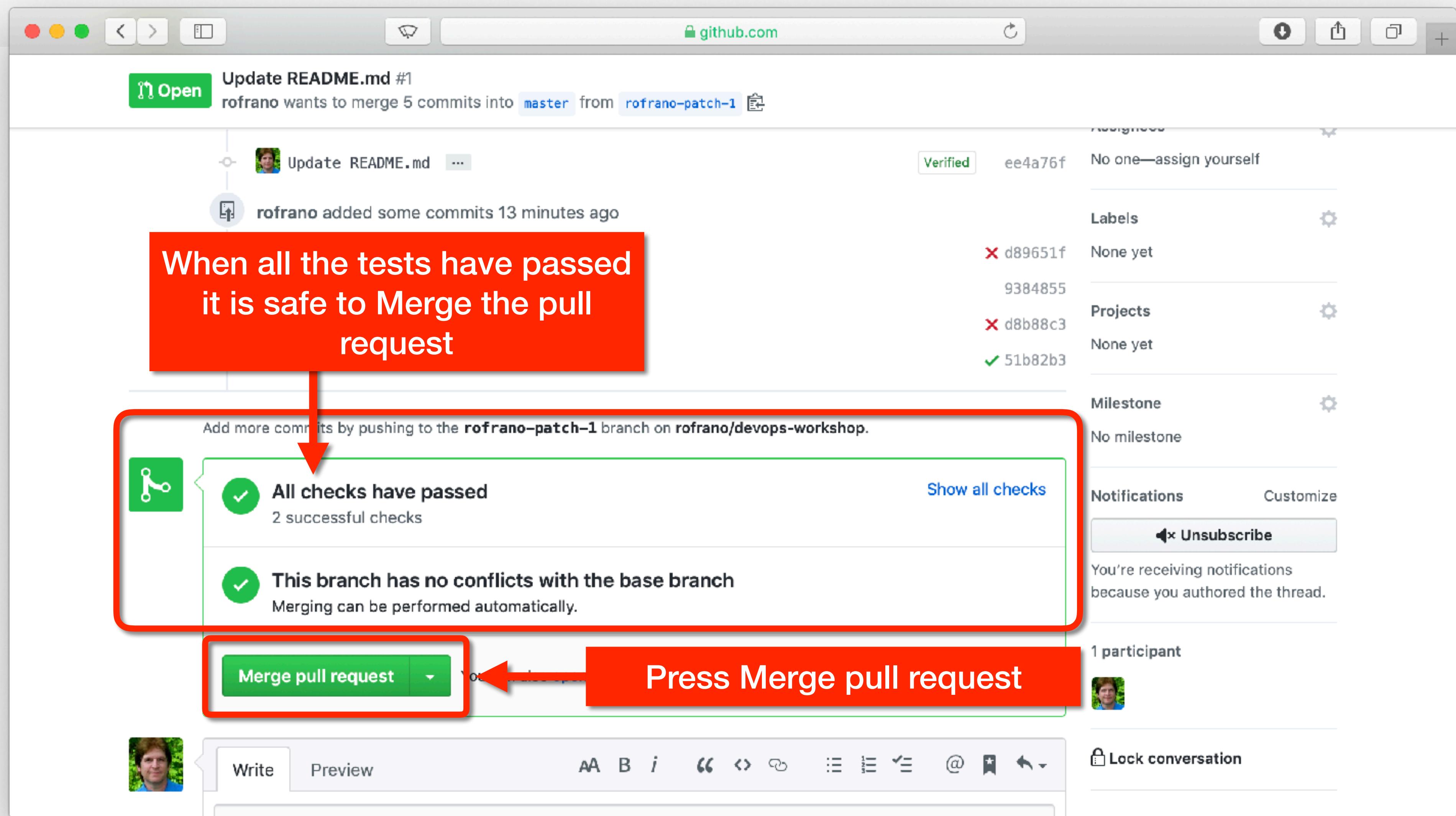
The log concludes with the message "Ran 20 tests in 1.458s", "OK", and "The command "nosetests" exited with 0." A "Top" button is visible in the bottom right corner of the browser window.

```
563 - Test deserialization of bad data
564 - Find Pets by Category
565 - Find a Pet by Name
566 - Find or return 404 found
567 - Find or return 404 NOT found
568 - Find a Pet by ID
569 - Test serialization of a Pet
570 - Update a Pet
571
572 Pet Server Tests
573 - Create a new Pet
574 - Delete a Pet
575 - Get a single Pet
576 - Get a list of Pets
577 - Get a Pet that's not found
578 - Test the Home Page
579 - Query Pets by Category
580 - Update an existing Pet
581
582 Name           Stmtns  Miss  Cover  Missing
583 -----
584 service/__init__.py    18      0   100%
585 service/models.py      59      4   93%   58, 92, 154-155
586 service/routes.py     99     17   83%   48, 53-55, 71-73, 80-82, 89-91, 120, 181, 217-218
587 -----
588 TOTAL             176     21   88%
589 -----
590 Ran 20 tests in 1.458s
591
592 OK
593
594 The command "nosetests" exited with 0.
595
```

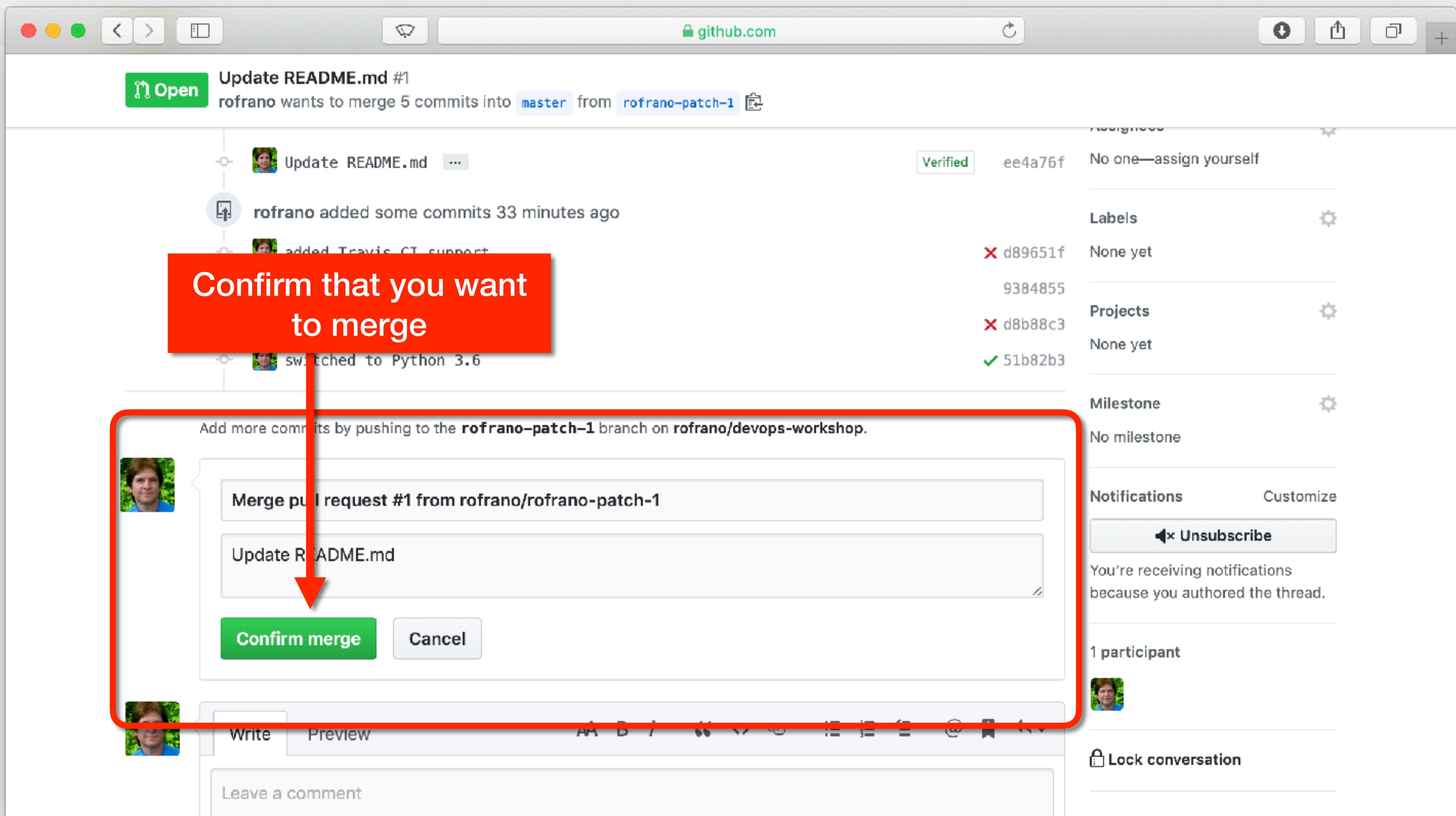
Back @ GitHub.com we are Green to Go



Back @ GitHub.com we are Green to Go



Are you sure?



Always Delete the Branch

The screenshot shows a GitHub pull request merge page. At the top, a purple banner indicates the pull request has been merged. Below the banner, the commit history shows a merge commit from the branch `rofrano-patch-1` into the `master` branch. A red callout box with white text reads: "Delete the branch after merge. You will create a new branch for every feature and don't need it anymore". A red arrow points from this callout to a button labeled "Delete branch" located in a summary box at the bottom of the merge commit. The summary box also states: "Pull request successfully merged and closed. You're all set—the `rofrano-patch-1` branch can be safely deleted." The right side of the screen displays the pull request's details, including the author (rofrano), commit history, and merge status.

What if the tests Fail?

The screenshot shows a GitHub pull request interface. At the top, it says "Update README.md #1" and "rofrano wants to merge 4 commits into master from rofrano-patch-1". A red callout box with white text states: "If any of the tests fail, Travis CI will let you know so that you don't merge code that will break the master branch!" An arrow points from this callout to the "All checks have failed" section. This section is highlighted with a red border. It contains two failing checks: "continuous-integration/travis-ci/pr — The Travis CI build failed" and "continuous-integration/travis-ci/push — The Travis CI build failed". Below this, a green checkmark indicates "This branch has no conflicts with the base branch". A "Merge pull request" button is at the bottom left, and a note says "You can also open this in GitHub Desktop or view command line instructions". On the right side, there are settings for assignees, labels, projects, milestones, notifications, and participants.

If any of the tests fail, Travis CI will let you know so that you don't merge code that will break the master branch!

Add more commits by pushing to the **rofrano-patch-1** branch on **rofrano/devops-workshop**.

All checks have failed 2 failing checks

continuous-integration/travis-ci/pr — The Travis CI build failed

continuous-integration/travis-ci/push — The Travis CI build failed

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Assignees: ee4a76f No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Customize

Unsubscribe

1 participant

Lock conversation

What if the tests Fail?

The screenshot shows a GitHub pull request interface. At the top, it says "Update README.md #1" and "rofrano wants to merge 4 commits into master from rofrano-patch-1". A red callout box contains the text: "If any of the tests fail, Travis CI will let you know so that you don't merge code that will break the master branch!". Below this, a red box highlights the "All checks have failed" section, which lists two failing checks: "continuous-integration/travis-ci/pr — The Travis CI build failed" and "continuous-integration/travis-ci/push — The Travis CI build failed". A red arrow points from this section to another red callout box at the bottom right that says "GitHub will turn the Merge button grey to let you know not to merge". The "Merge pull request" button is highlighted with a red border and has a dropdown arrow.

If any of the tests fail, Travis CI will let you know so that you don't merge code that will break the master branch!

Add more commits by pushing to the **rofrano-patch-1** branch on **rofrano/devops-workshop**.

All checks have failed 2 failing checks

continuous-integration/travis-ci/pr — The Travis CI build failed

continuous-integration/travis-ci/push — The Travis CI build failed

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

GitHub will turn the Merge button grey to let you know not to merge

Display your Badges

The screenshot shows a GitHub repository page for a project named "DevOps Workshop". At the top, there's a list of files and their commit history:

File	Commit Message	Time Ago
.travis.yml	switched to Python 3.6	29 minutes ago
LICENSE	Initial commit	11 days ago
Procfile	initial app	17 hours ago
README.md	switched to Python 3.6	29 minutes ago
Vagrantfile	added Travis CI support	41 minutes ago
requirements.txt	initial app	17 hours ago
setup.cfg	added tests	17 hours ago

Below the file list, the README.md file is displayed. It contains the title "DevOps Workshop" and a green badge with the text "build passing". A red arrow points from a callout box to this badge.

Displays your build test status
Note: This lets developers know that the build is good to use

DevOps Workshop Training that uses Vagrant, VirtualBox, Docker, and Minikube

This repository is part of a 4 Hour DevOps Workshop that gives attendees first hand experience in building a microservice and deploying it as a container on Kubernetes using DevOps practices and methods. It demonstrates how to create a simple RESTful service using Python Flask and SQLAlchemy to access a persistent resource. Its purpose is to show the correct API and return codes that should be used for a REST API. This repo also demonstrates the use of Test Driven Development which is required for setting up a Continuous Integrate and Continuous Delivery Pipeline (CI/CD).

During the workshop, attendees will build a simple Python Flask microservice using Test Driven Development techniques.

Summary

- You should have a basic understanding of how Travis CI works
- You learned how to do create a `.travis.yml` file for your project
- You should be able to integrate Travis CI into your build process

