Code Coverage & Continuous Integration

Jared O’Neal
Mathematics and Computer Science Division
Argonne National Laboratory

Better Scientific Software Tutorial
SC19, Denver, Colorado

License, Citation and Acknowledgements

License and Citation

- This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
- **The requested citation for the overall tutorial is: David E. Bernholdt, Anshu Dubey, Michael A. Heroux, and Jared O'Neal, Better Scientific Software tutorial, in SC '19: International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, Colorado, 2019. DOI: 10.6084/m9.figshare.10114880**
- Individual modules may be cited as *Module Authors, Module Title*, in Better Scientific Software Tutorial…

Acknowledgements

- Alicia Klinvex
- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- This work was performed in part at the Argonne National Laboratory, which is managed managed by UChicago Argonne, LLC for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.

CODE COVERAGE

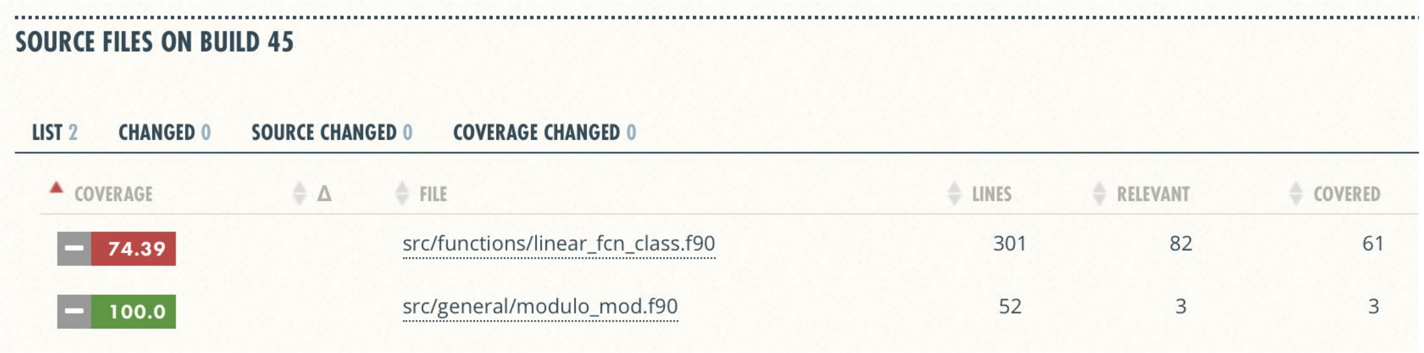
How do we determine what other tests are needed?

Code coverage tools

- Expose parts of the code that aren’t being tested
- gcov
  - standard utility with the GNU compiler collection suite
  - Compile/link with –coverage & turn off optimization
  - counts the number of times each statement is executed
- lcov
  - a graphical front-end for gcov
  - available at http://ltp.sourceforge.net/coverage/lcov.php
- Hosted servers (*e.g.* coveralls, codecov)
  - graphical visualization of results
  - push results to server through continuous integration server

Code Coverage Output

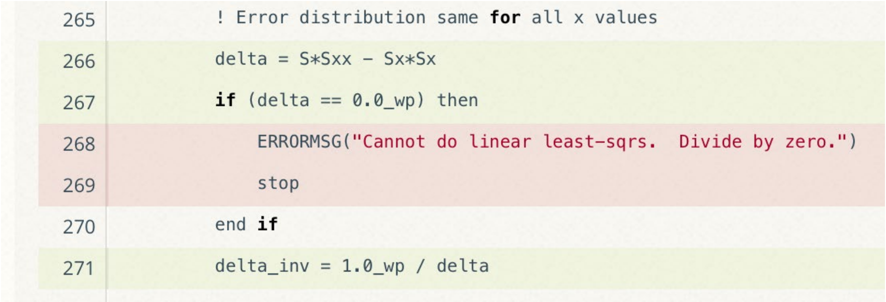
Overall Analysis



SOURCE FILES ON BUILD 45

COVERAGE	Δ	FILE	LINES	RELEVANT	COVERED
74.39		src/functions/linear_fcn_class.f90	301	82	61
100.0		src/general/modulo_mod.f90	52	3	3

Detailed Analysis



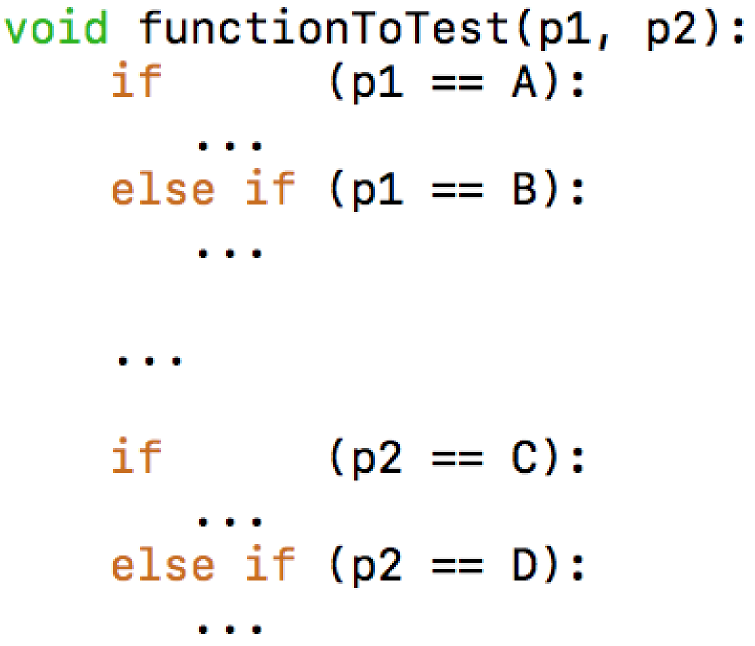
```
        ! Error distribution same for all x values
        delta = S*Sxx - Sx*Sx
        if (delta == 0.0_wp) then
            ERRORMSG("Cannot do linear least-sqrs.  Divide by zero.")
            stop
        end if
        delta_inv = 1.0_wp / delta
```

https://github.com/jrdoneal/infrastructure

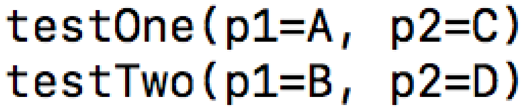
Code Coverage is Popular

- gcov also works for C and Fortran
- Other tools exist for other languages
  - JCov for Java
  - Coverage.py for python
  - Devel::Cover for perl
  - profile for MATLAB
  - *etc*.

Limitations



```
void functionToTest(p1, p2):
    if      (p1 == A):
        ...
    else if (p1 == B):
        ...

    ...

    if      (p2 == C):
        ...
    else if (p2 == D):
        ...
```



```
testOne(p1=A, p2=C)
testTwo(p1=B, p2=D)
```

- 100% coverage by line
- Checks 2 of 4 pathways only
- Possibility for bugs

Other Code Coverage

Test-Driven Development

- Covers functionality coevolved with tests
- Limited if we have only unit tests

Requirements & Verification

- Covers higher-level functionality and constraints
- Depends on completeness

CONTINUOUS INTEGRATION

The Short & Sweet of Continuous Integration

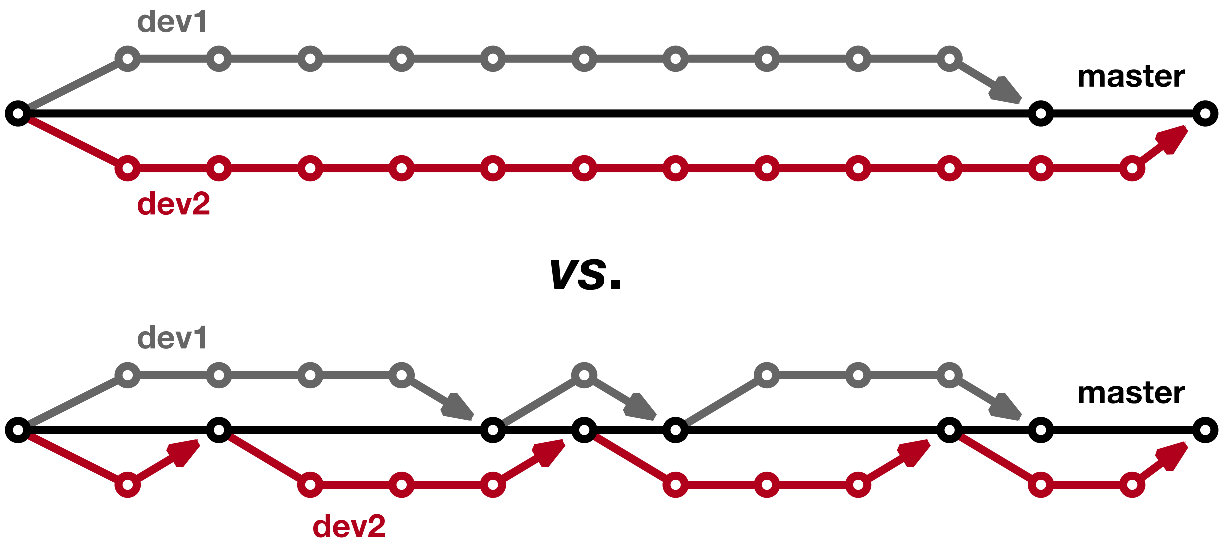
A master branch that always works

- DVCS workflow isolate master from integration environment
- Extend workflow to address difficulties of integrating
  - Minimize likelihood of merge conflict
  - Detect bugs immediately
  - Make debugging process quick and easy

Work Decomposition

Commit and integrate often

- Limit divergence between feature and master branches
- Decreased probability of conflict
- Conflict resolution is simpler and less risky



Error Detection

Test at integration to identify failures immediately

- Control quality of code
- Isolate failure to few commits
- No context switching for programmer

We want a system that

- triggers automated builds/tests on target environments when code changes and
- ideally tests on proposed merge product without finalizing merge.

Test Servers

Servers that

- automate the execution of a test suite or a subset of a test suite,
- allow for running tests on different environments,
- host an interface for viewing results, and
- allows for configuring when the tests are run.

Examples

- CTest/CDash
- Jenkins
- Travis CI and GitLab CI

Cloud-based Test Servers

- Linked to VCS hosts
  - GitHub & Travis CI
  - GitLab CI
  - BitBucket Pipelines
- Automated builds/tests triggered *via* pushes and pull requests
- Builds/tests can be run on cloud systems
- Test results are reported in repository’s web interface
- Can trigger code coverage analysis & documentation build

Continuous integration (CI)

- Has existed for some time and interest is growing
- HPC community working to adapt CI for HPC machines
- Setup, maintenance, and monitoring required
- Prerequisites
  - A reasonably-automated build system
  - An automated test system with significant test coverage & useful feedback
  - Builds/tests must finish in reasonable about of time
  - Ability to bundle subset of tests

CI HELLO WORLD

Simplest CI example

https://github.com/jrdoneal/CI_HelloWorld
https://travis-ci.org/jrdoneal/CI_HelloWorld

CI example w/ multiple platforms and specific compiler versions

https://github.com/jrdoneal/CI_Multiplatform

Code coverage, testing and CI tutorial (C++)

https://github.com/amklinv/morpheus

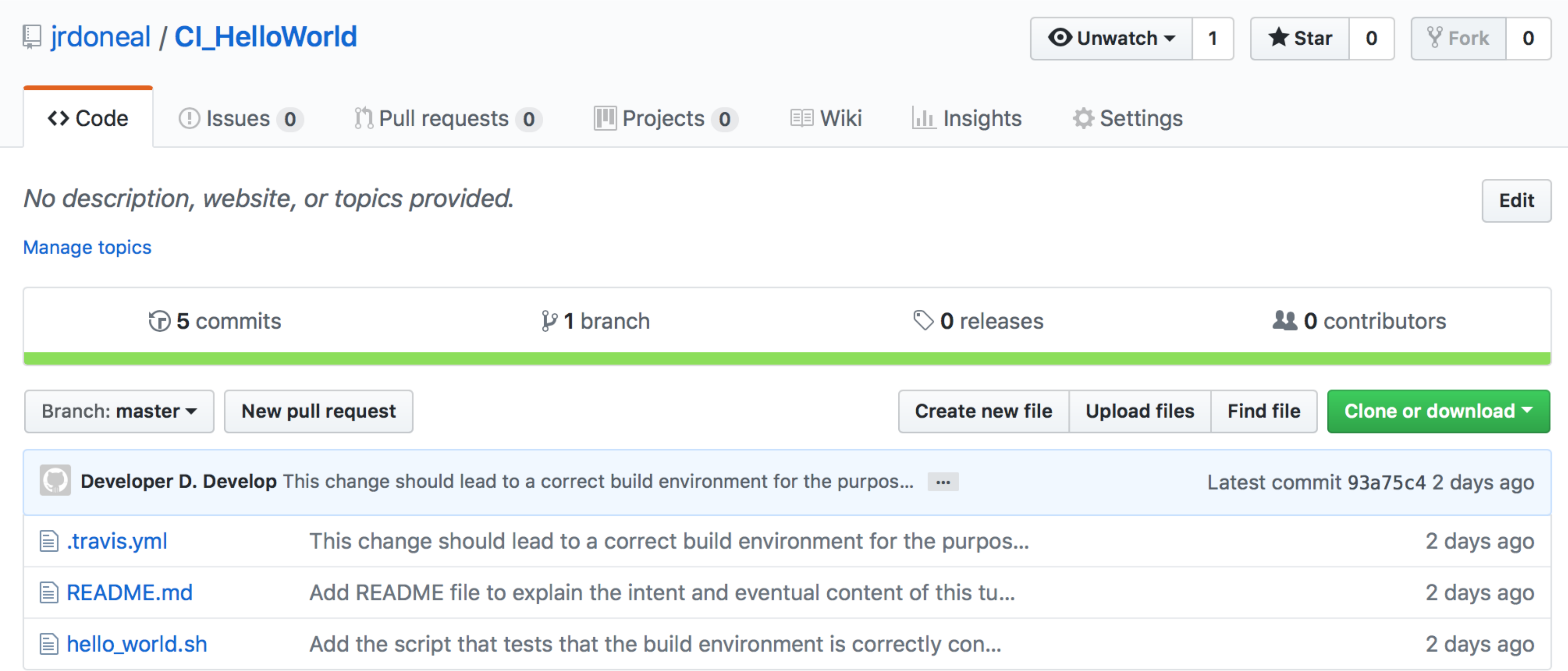
Code coverage, testing, and CI example (Fortran, C++)

https://github.com/jrdoneal/infrastructure

CI HELLO WORLD – BACKUP SLIDES

GitHub Repository Page

https://github.com/jrdoneal/CI_HelloWorld



No description, website, or topics provided.

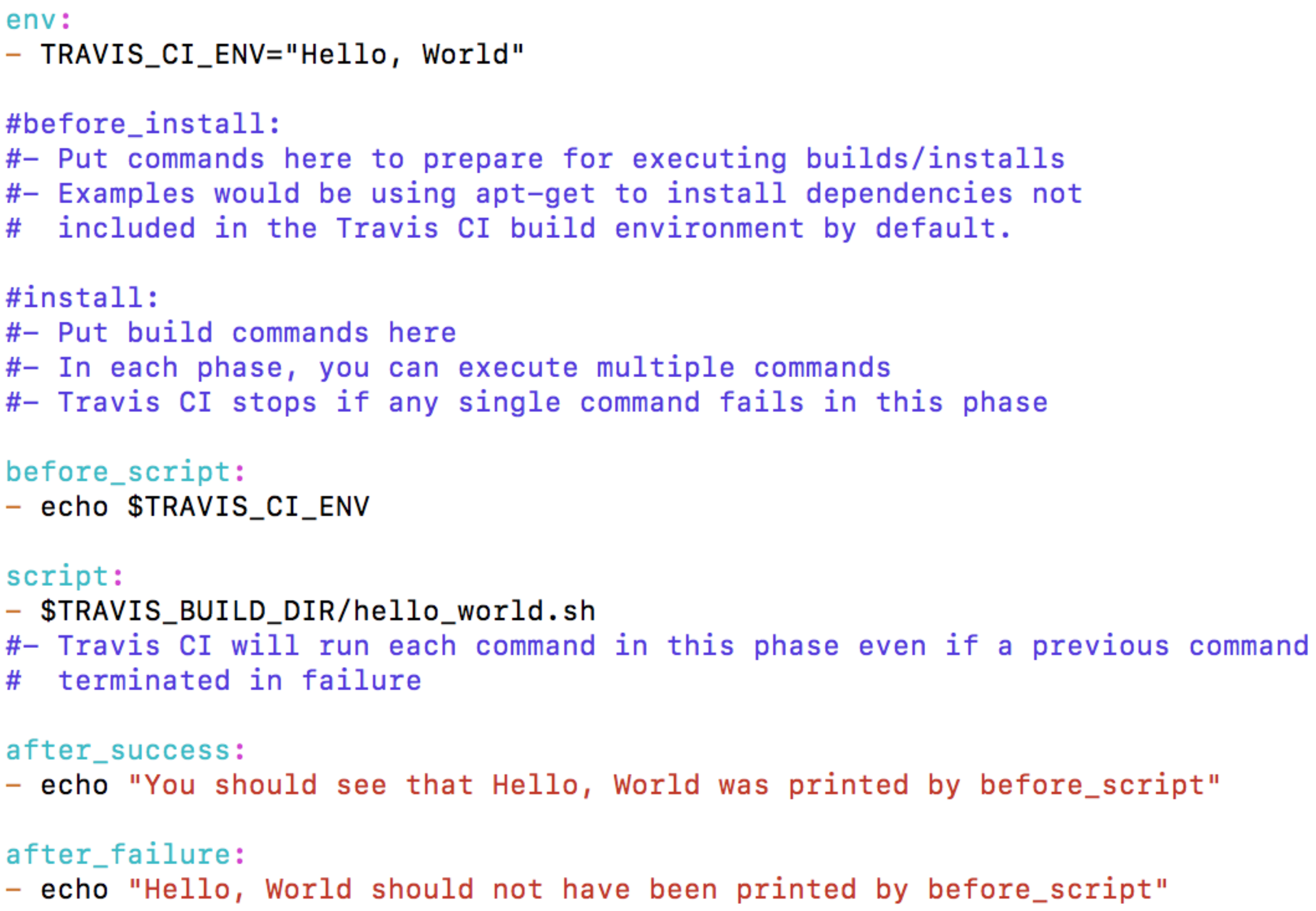
5 commits · 1 branch · 0 releases · 0 contributors

Developer D. Develop This change should lead to a correct build environment for the purpos... … Latest commit 93a75c4 2 days ago

.travis.yml	This change should lead to a correct build environment for the purpos...	2 days ago
README.md	Add README file to explain the intent and eventual content of this tu...	2 days ago
hello_world.sh	Add the script that tests that the build environment is correctly con...	2 days ago

Travis CI Configuration File

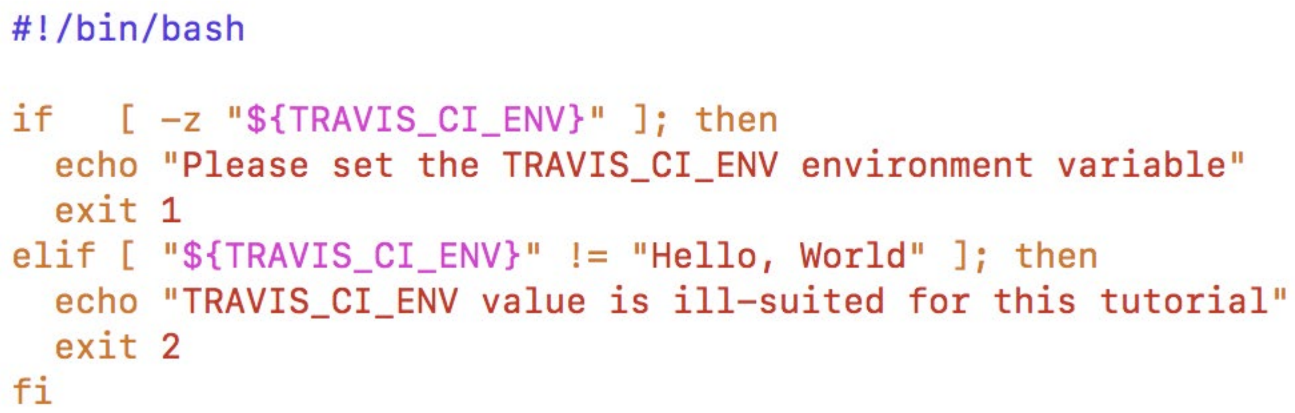
.travis.yml



```
env:
- TRAVIS_CI_ENV="Hello, World"

#before_install:
#- Put commands here to prepare for executing builds/installs
#- Examples would be using apt-get to install dependencies not
#  included in the Travis CI build environment by default.

#install:
#- Put build commands here
#- In each phase, you can execute multiple commands
#- Travis CI stops if any single command fails in this phase

before_script:
- echo $TRAVIS_CI_ENV

script:
- $TRAVIS_BUILD_DIR/hello_world.sh
#- Travis CI will run each command in this phase even if a previous command
#  terminated in failure

after_success:
- echo "You should see that Hello, World was printed by before_script"

after_failure:
- echo "Hello, World should not have been printed by before_script"
```

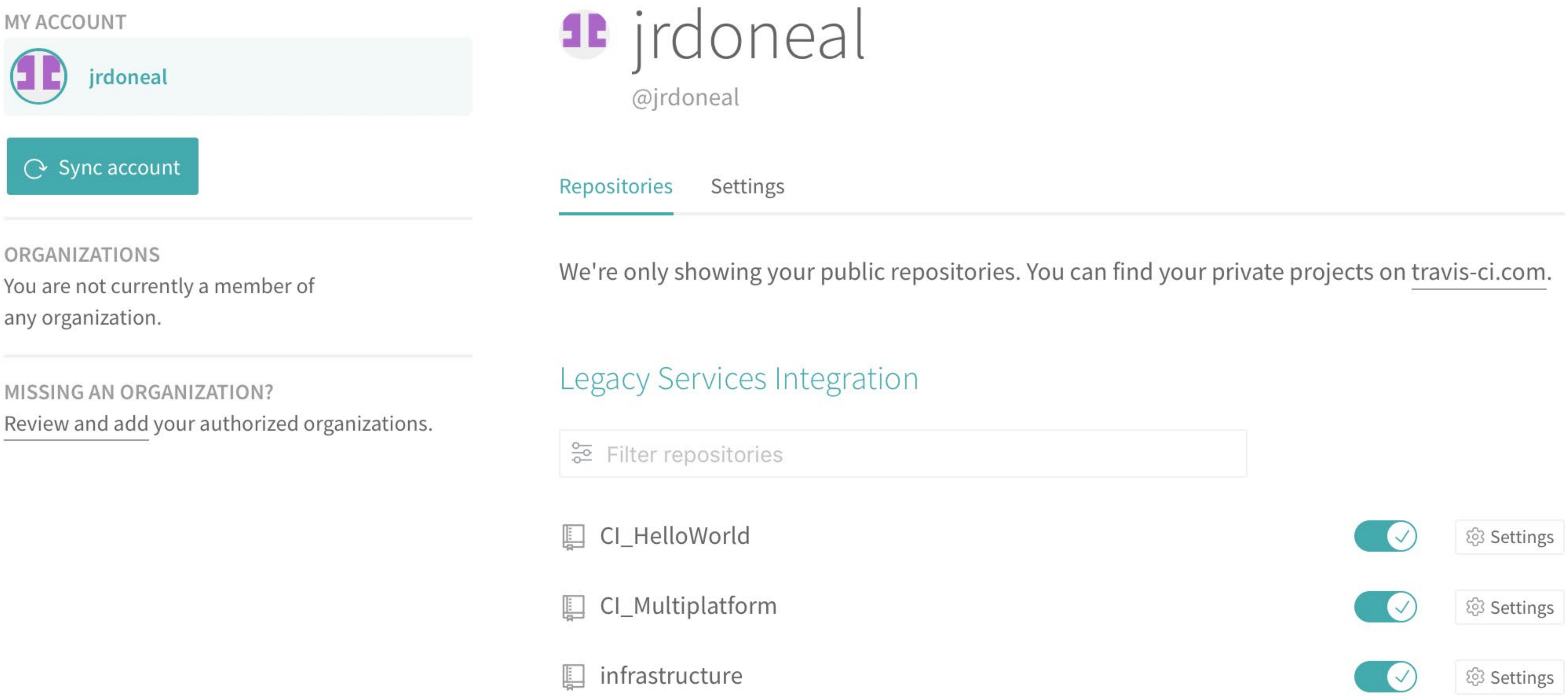
The Script Phase

hello_world.sh



```
#!/bin/bash

if   [ -z "${TRAVIS_CI_ENV}" ]; then
  echo "Please set the TRAVIS_CI_ENV environment variable"
  exit 1
elif [ "${TRAVIS_CI_ENV}" != "Hello, World" ]; then
  echo "TRAVIS_CI_ENV value is ill-suited for this tutorial"
  exit 2
fi
```

Connecting GitHub & Travis CI



MY ACCOUNT

jrdoneal

Sync account

ORGANIZATIONS

You are not currently a member of any organization.

MISSING AN ORGANIZATION?

Review and add your authorized organizations.

jrdoneal

@jrdoneal

Repositories Settings

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

Legacy Services Integration

Filter repositories

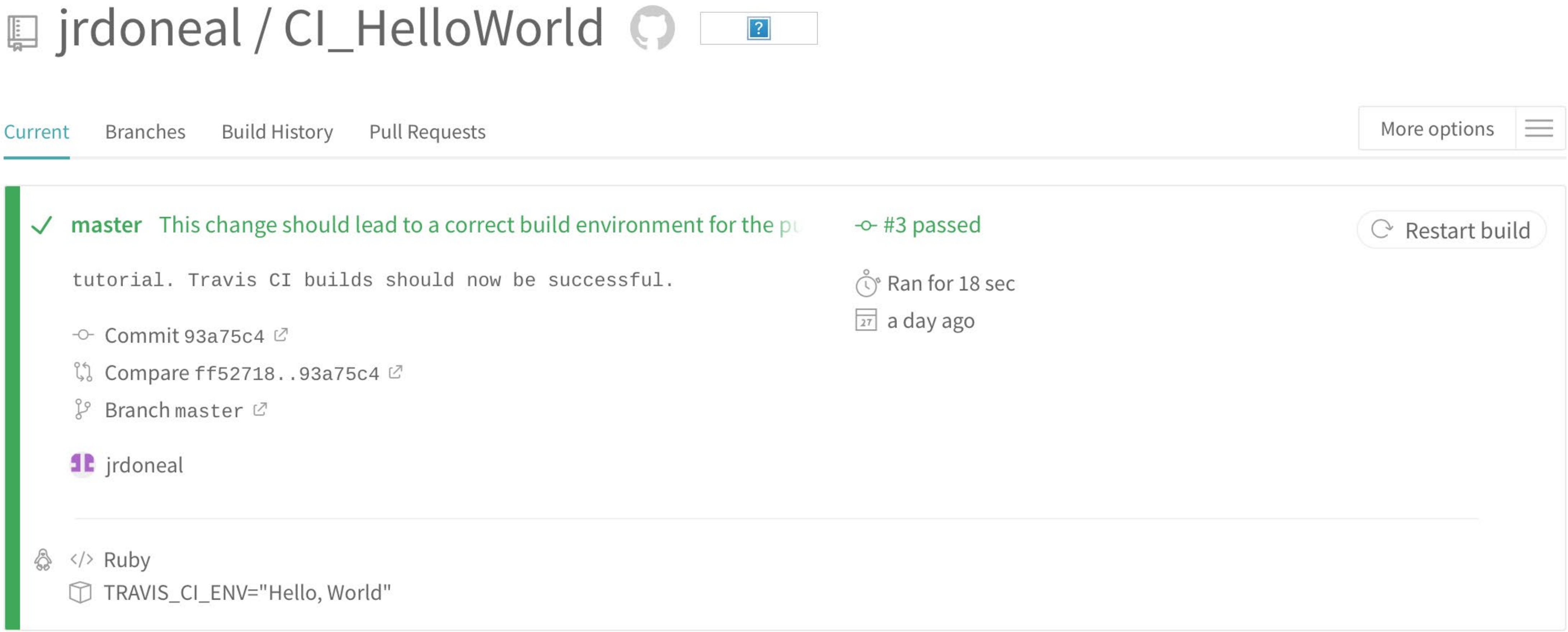
CI_HelloWorld ✓ Settings

CI_Multiplatform ✓ Settings

infrastructure ✓ Settings

Repository in Travis CI

https://travis-ci.org/jrdoneal/CI_HelloWorld



jrdoneal / CI_HelloWorld

✓ master This change should lead to a correct build environment for the pu

```
tutorial. Travis CI builds should now be successful.
```

#3 passed

Ran for 18 sec

a day ago

Commit 93a75c4

Compare ff52718..93a75c4

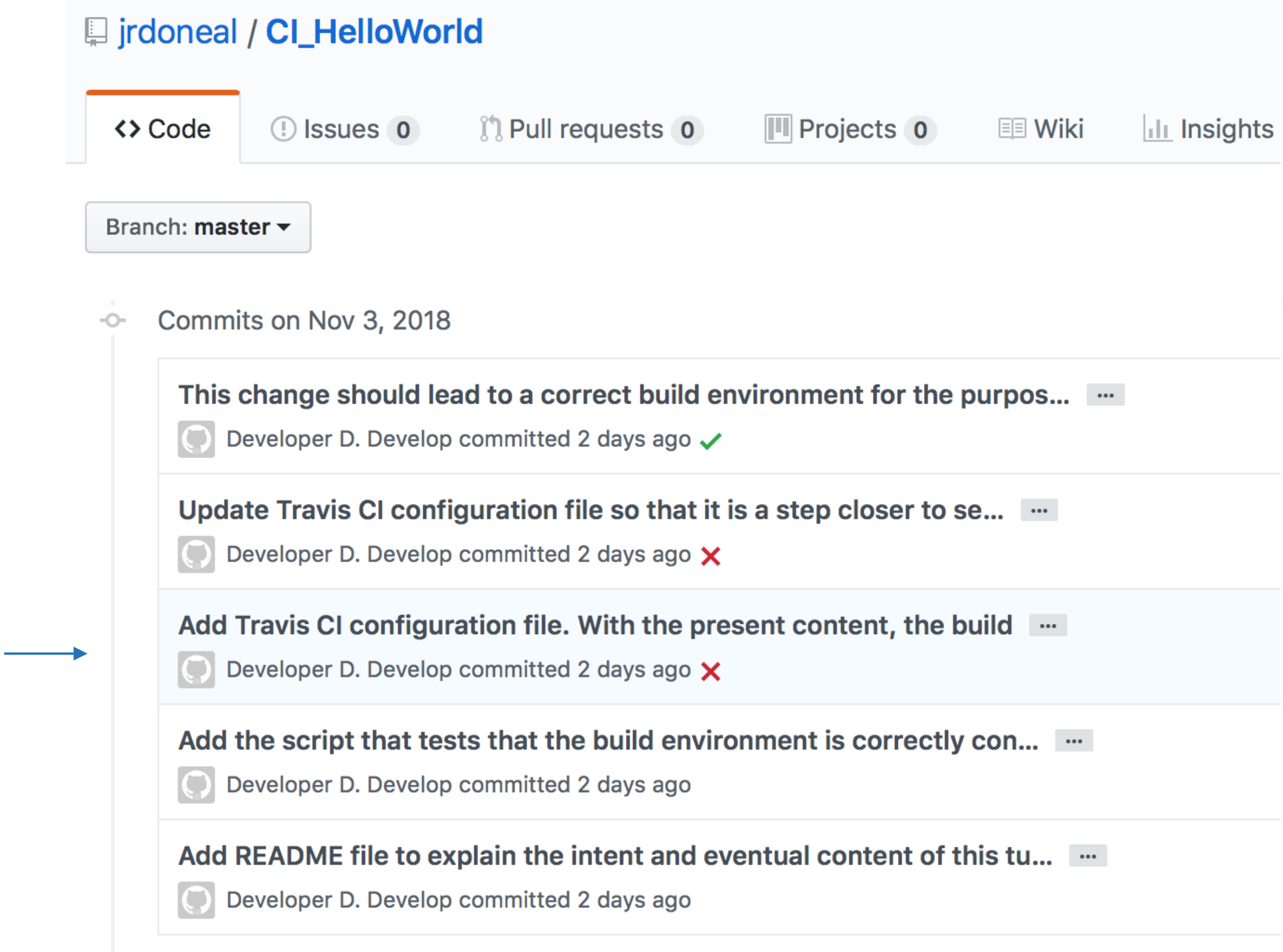
Branch master

jrdoneal

Ruby

TRAVIS_CI_ENV="Hello, World"

Commit History



jrdoneal / CI_HelloWorld

Commits on Nov 3, 2018

This change should lead to a correct build environment for the purpos… ···
Developer D. Develop committed 2 days ago ✓

Update Travis CI configuration file so that it is a step closer to se… ···
Developer D. Develop committed 2 days ago ✗

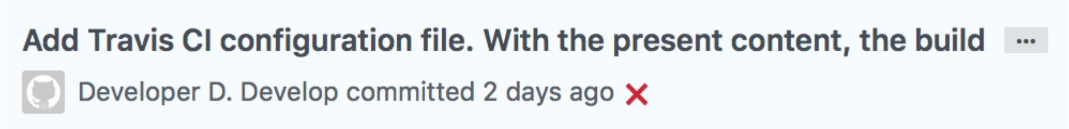
Add Travis CI configuration file. With the present content, the build ···
Developer D. Develop committed 2 days ago ✗

Add the script that tests that the build environment is correctly con… ···
Developer D. Develop committed 2 days ago

Add README file to explain the intent and eventual content of this tu… ···
Developer D. Develop committed 2 days ago

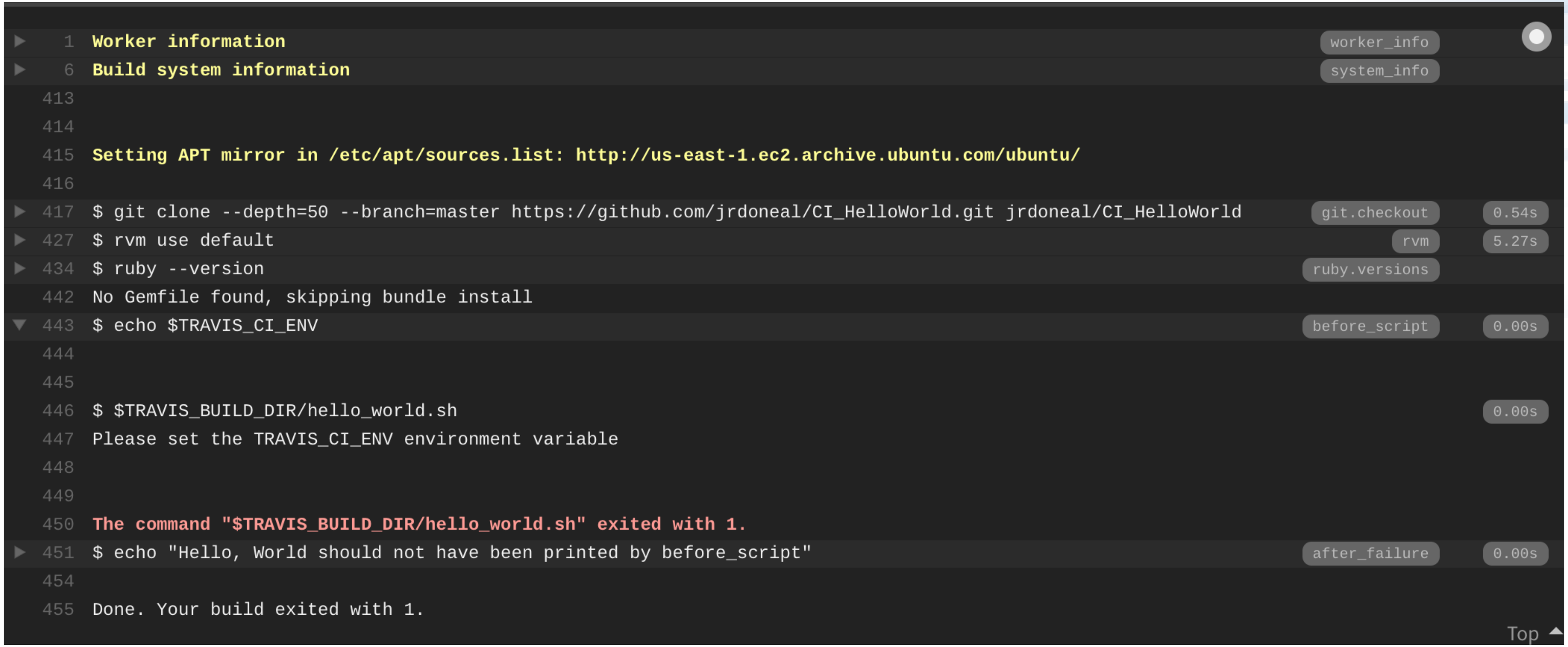
.travis.yml added →

Travis CI Build History



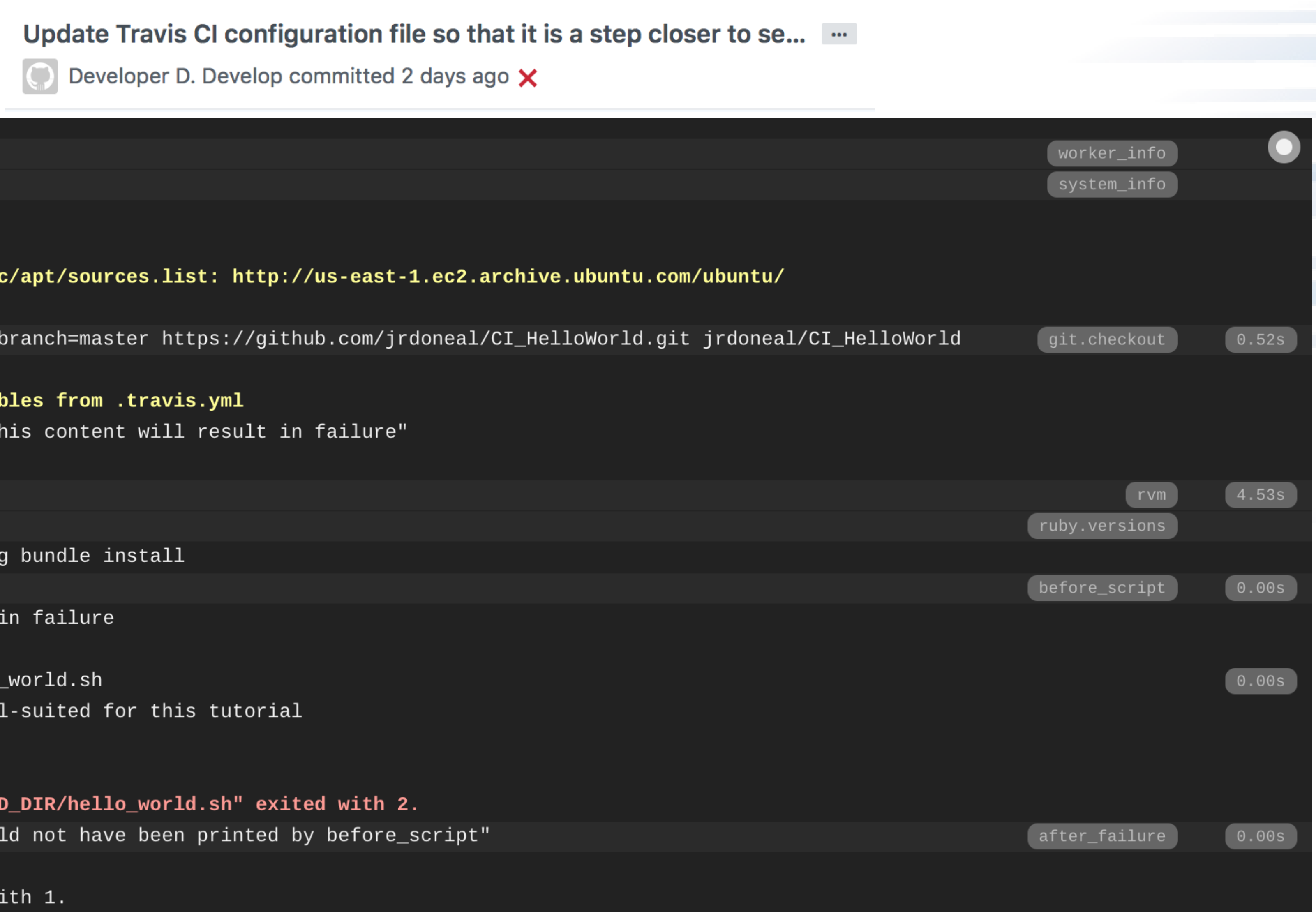
Add Travis CI configuration file. With the present content, the build …

Developer D. Develop committed 2 days ago ✗



```
▶ Worker information
▶ Build system information


Setting APT mirror in /etc/apt/sources.list: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/

▶ $ git clone --depth=50 --branch=master https://github.com/jrdoneal/CI_HelloWorld.git jrdoneal/CI_HelloWorld
▶ $ rvm use default
▶ $ ruby --version
No Gemfile found, skipping bundle install
▼ $ echo $TRAVIS_CI_ENV


$ $TRAVIS_BUILD_DIR/hello_world.sh
Please set the TRAVIS_CI_ENV environment variable


The command "$TRAVIS_BUILD_DIR/hello_world.sh" exited with 1.
▶ $ echo "Hello, World should not have been printed by before_script"

Done. Your build exited with 1.
```

Travis CI Build History

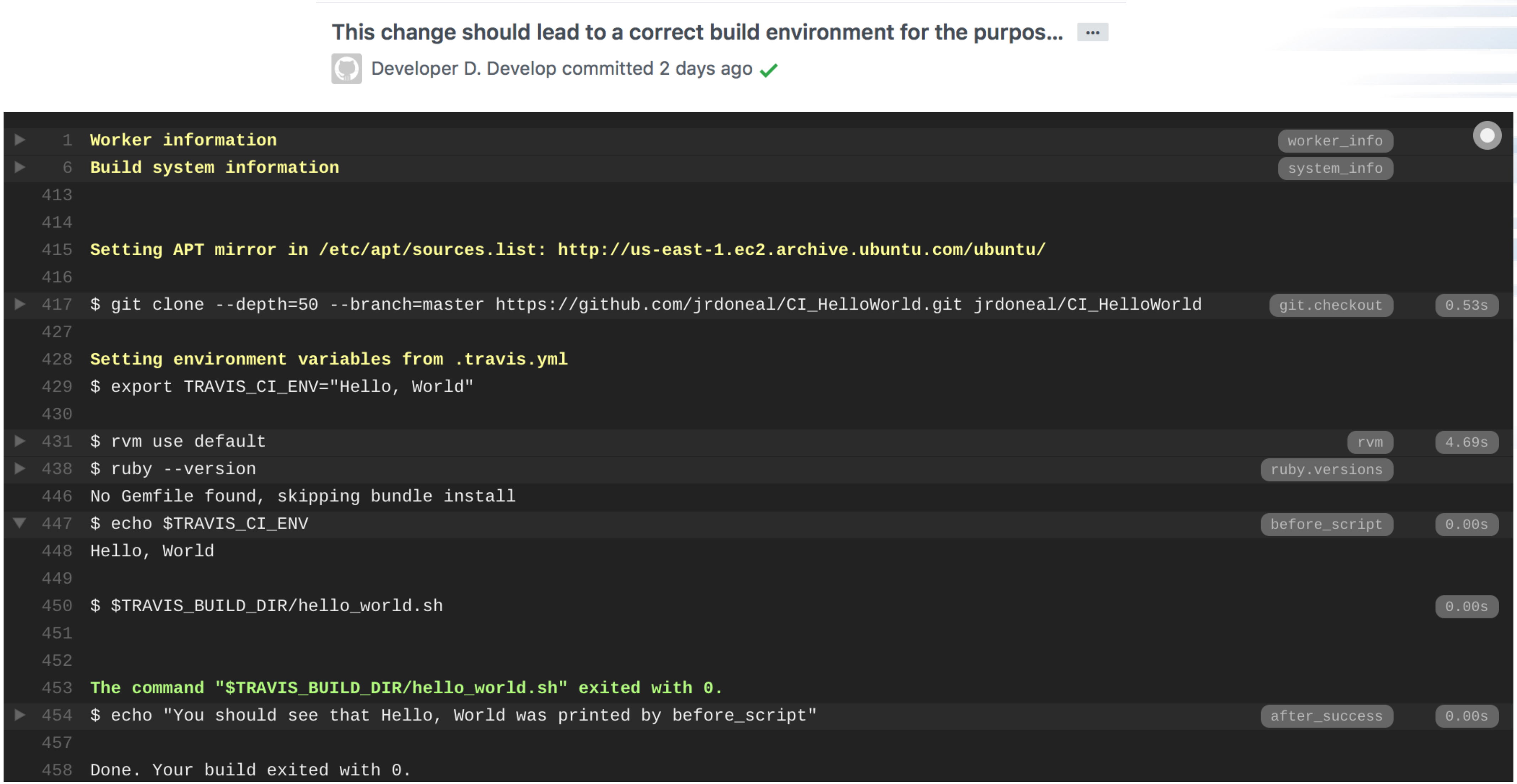


Update Travis CI configuration file so that it is a step closer to se... ···

Developer D. Develop committed 2 days ago ✗

```
▶ Worker information                                                                     worker_info
▶ Build system information                                                               system_info


Setting APT mirror in /etc/apt/sources.list: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/

▶ $ git clone --depth=50 --branch=master https://github.com/jrdoneal/CI_HelloWorld.git jrdoneal/CI_HelloWorld   git.checkout   0.52s

Setting environment variables from .travis.yml
$ export TRAVIS_CI_ENV="This content will result in failure"

▶ $ rvm use default                                                                      rvm   4.53s
▶ $ ruby --version                                                                       ruby.versions
No Gemfile found, skipping bundle install
▼ $ echo $TRAVIS_CI_ENV                                                                  before_script   0.00s
This content will result in failure

$ $TRAVIS_BUILD_DIR/hello_world.sh                                                       0.00s
TRAVIS_CI_ENV value is ill-suited for this tutorial


The command "$TRAVIS_BUILD_DIR/hello_world.sh" exited with 2.
▶ $ echo "Hello, World should not have been printed by before_script"                    after_failure   0.00s

Done. Your build exited with 1.
```

Travis CI Build History



This change should lead to a correct build environment for the purpos... ···

Developer D. Develop committed 2 days ago ✓

```
▶ Worker information                                                                            worker_info
▶ Build system information                                                                      system_info


  Setting APT mirror in /etc/apt/sources.list: http://us-east-1.ec2.archive.ubuntu.com/ubuntu/

▶ $ git clone --depth=50 --branch=master https://github.com/jrdoneal/CI_HelloWorld.git jrdoneal/CI_HelloWorld   git.checkout   0.53s

  Setting environment variables from .travis.yml
  $ export TRAVIS_CI_ENV="Hello, World"

▶ $ rvm use default                                                                             rvm   4.69s
▶ $ ruby --version                                                                              ruby.versions
  No Gemfile found, skipping bundle install
▼ $ echo $TRAVIS_CI_ENV                                                                         before_script   0.00s
  Hello, World

  $ $TRAVIS_BUILD_DIR/hello_world.sh                                                                            0.00s


  The command "$TRAVIS_BUILD_DIR/hello_world.sh" exited with 0.
▶ $ echo "You should see that Hello, World was printed by before_script"                        after_success   0.00s

  Done. Your build exited with 0.
```

! →