Continuous Integration Using Github Action

Juneyoung Lee & Sung-Hwan Lee

Continuous Integration?

- Build your project & run tests whenever...
  - A new commit is created
  - A pull request is made
- Helps developers to be aware of validity of a patch + the master branch

My pull request #1

Open **aqjune** wants to merge 1 commit into `master` from `hi`

Conversation 0 | Commits 1 | Checks 0 | Files changed 1

aqjune commented 29 seconds ago

No description provided.

. 45dcb67

Add more commits by pushing to the **hi** branch on **aqjune/swppbackend**.

Some checks haven't completed yet
1 in progress check

Hide all checks

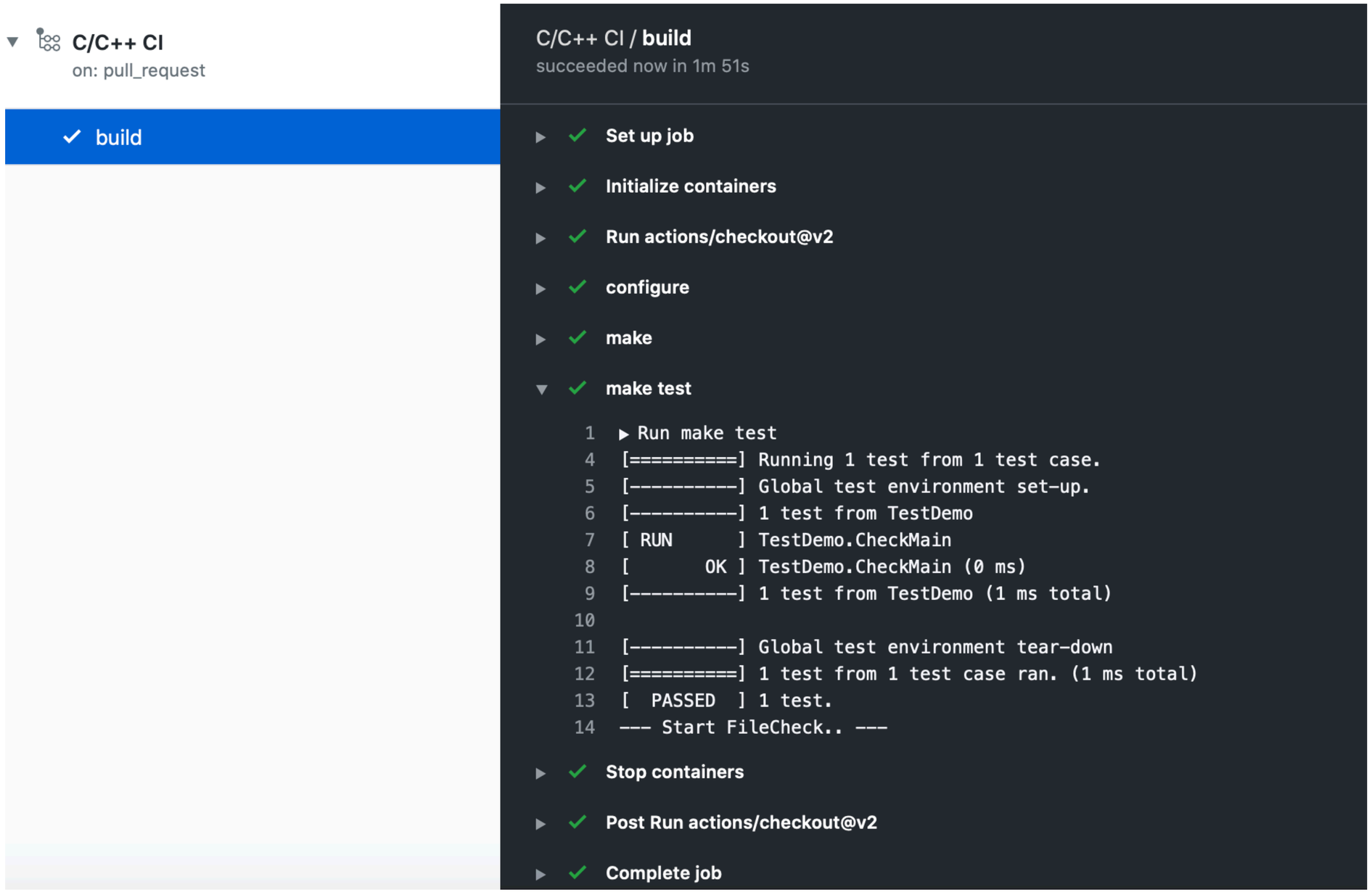
C/C++ CI / build (pull_request) *In progress — This check has started...* Details

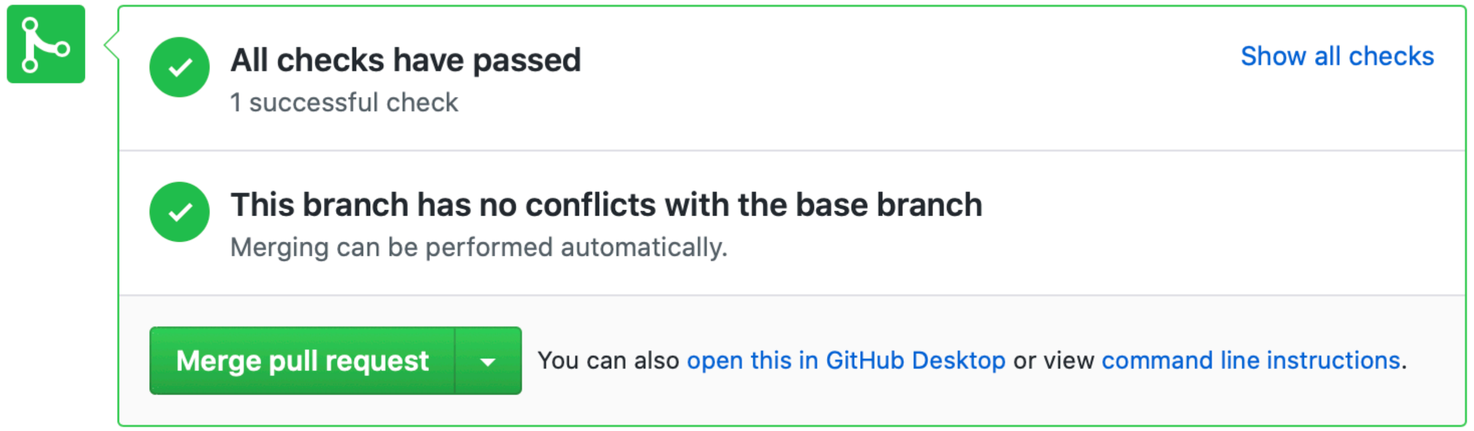
This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

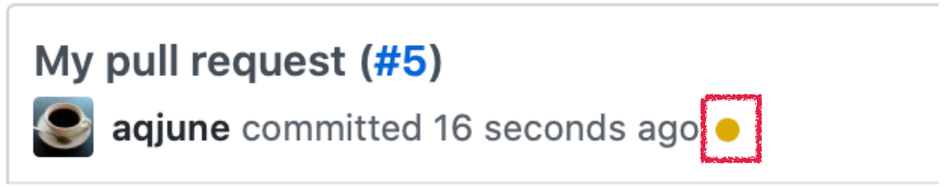
You can also open this in GitHub Desktop or view command line instructions.

GitHub is checking validity of 'hi' by cloning and building it.


C/C++ CI
on: pull_request
build
C/C++ CI / build
succeeded now in 1m 51s
Set up job
Initialize containers
Run actions/checkout@v2
configure
make
make test
1 ▶ Run make test
4 [==========] Running 1 test from 1 test case.
5 [----------] Global test environment set-up.
6 [----------] 1 test from TestDemo
7 [RUN] TestDemo.CheckMain
8 [OK] TestDemo.CheckMain (0 ms)
9 [----------] 1 test from TestDemo (1 ms total)
10
11 [----------] Global test environment tear-down
12 [==========] 1 test from 1 test case ran. (1 ms total)
13 [PASSED] 1 test.
14 --- Start FileCheck.. ---
Stop containers
Post Run actions/checkout@v2
Complete job



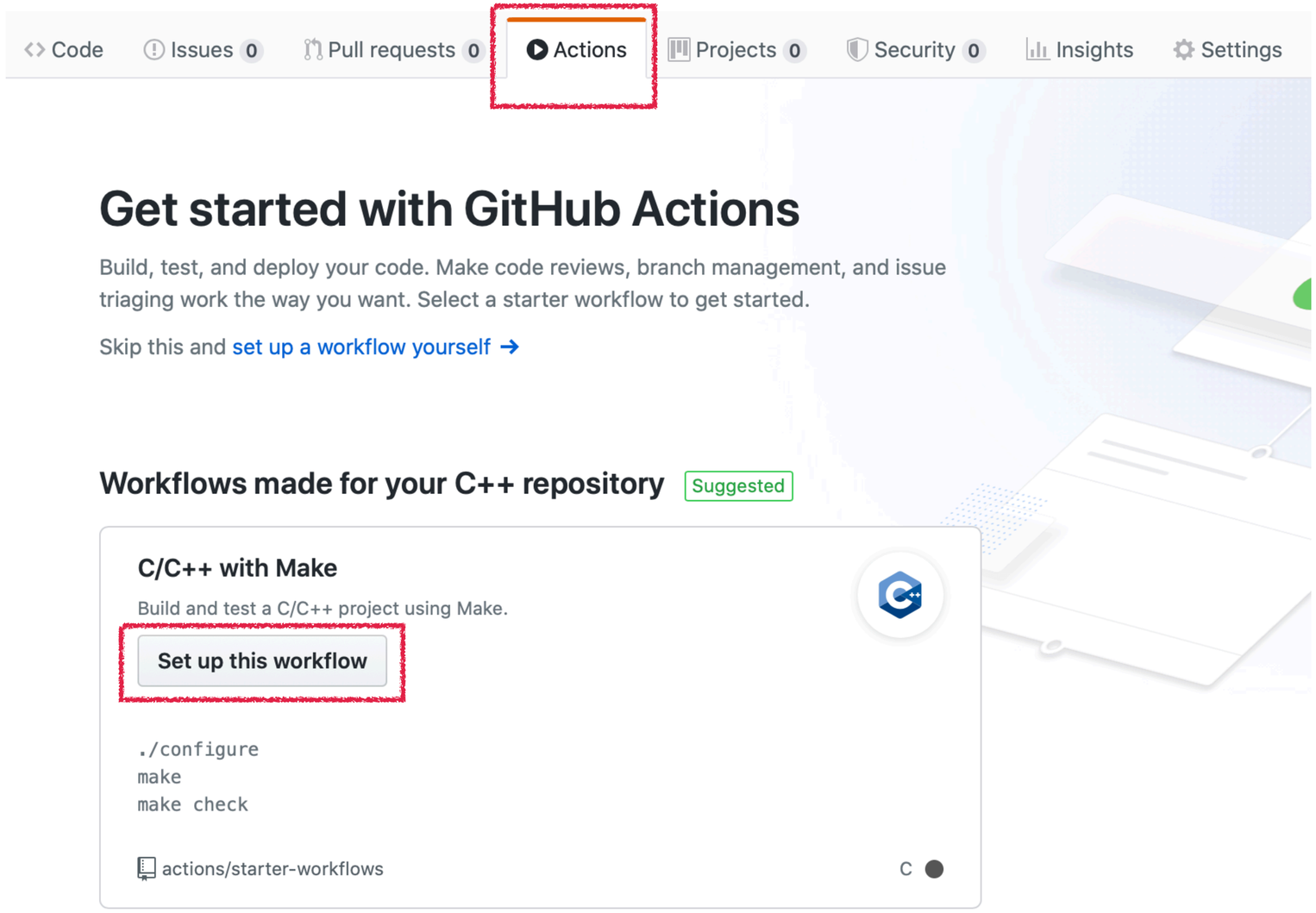
(Don’t forget to do ‘squash and merge’)



GitHub is running scripts after the pull request is merged;

It runs on master branch too.

How to use CI for your project?



Start commit

Edit new file | Preview | Spaces | 2 | No wrap

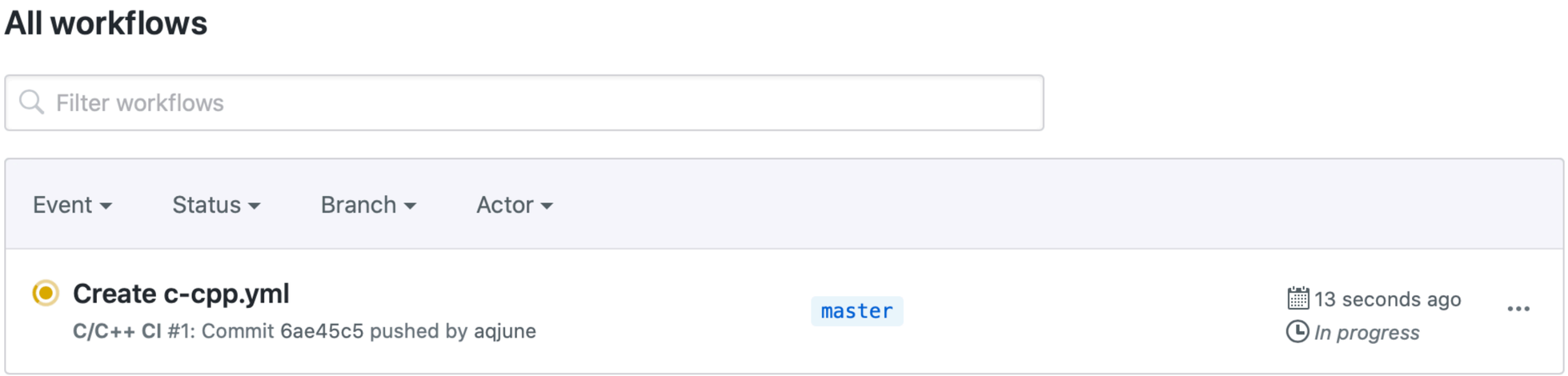
```yaml
name: CI

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

jobs:
  build:

    runs-on: ubuntu-latest
    container: sunghwanlee/swpp202101-ci

    steps:
    - uses: actions/checkout@v2
    - name: configure
      run: ./configure.sh /llvmscript/llvm-12.0-releaseassert/bin
    - name: make
      run: make
    - name: make test
      run: make test

```

Run this script whenever there is a push or PR to master

Uses a docker container

(1) checkout the current branch

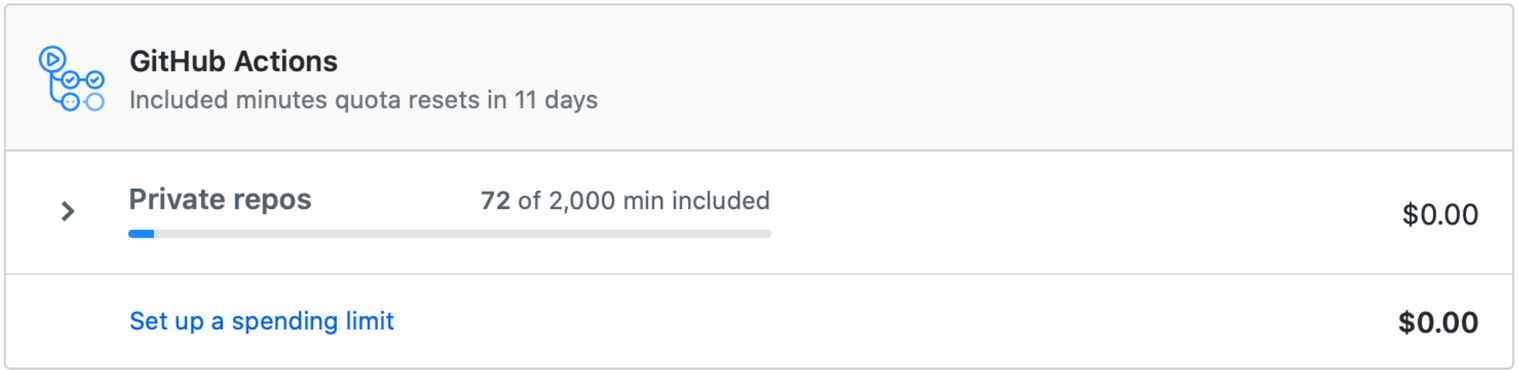
(2) Run configure.sh, make, make test



Now you're done. :)

Should I pay to run CI?

- You'll be given a free-tier quota per month: 2000 min, 500 MB *
- https://github.com/settings/billing shows the usage
- In our case, space isn't used (it is for packaging); time is the constraint



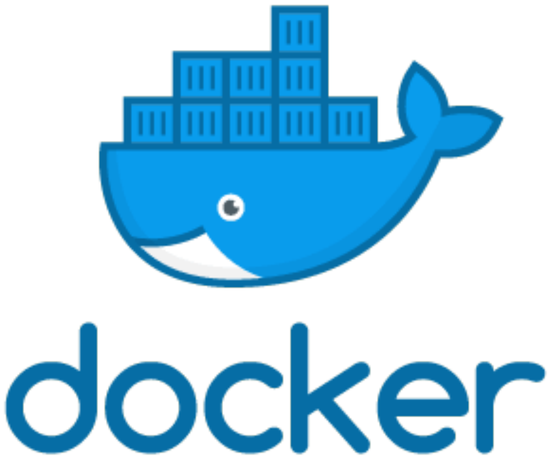
* https://help.github.com/en/github/setting-up-and-managing-billing-and-payments-on-github/about-billing-for-github-actions

Isn’t compiling LLVM needed?

- Note that you didn’t have to compile LLVM to run CI
- The secret is behind ‘container’ option
- It is given a lightweight virtual machine that has LLVM already built in
- It is called a *docker container*, and published online through *DockerHub*

Docker

- A lightweight virtual machine maintainer
- Has two important concepts:
  - *Image*: a read-only disk (analogous to .iso file)
  - *Container*: a virtual machine created from an image
- Dockerfile specifies how to create an image
  - Start from Ubuntu 20.04, apt-get install abc, clone LLVM, ...



DockerHub

hub.docker.com

- Git : GitHub = Docker : DockerHub
- As GitHub does, a single account has several repositories
- A repo contains several tags (think of branches in Git)
- Each tag corresponds to an image
- We provide a docker image for setting up CI for our project: https://hub.docker.com/repository/docker/sunghwanlee/swpp202101-ci