

Continuous Integration & Delivery

Best Practices Workshop, March 25-26 2019, Hampton VA

Carlos Cruz

Jules Kouatchou

Brent Smith

NASA GSFC Code 606/610 (ASTG/GMAO)
Greenbelt, Maryland 20771



Introduction

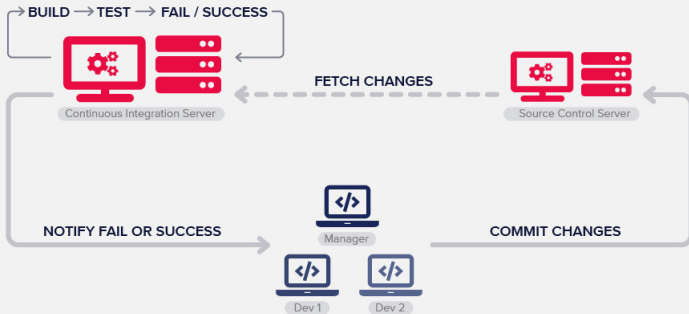
GitHub with Travis CI

Errata



Continuous Integration

Continuous Integration (CI) is a development practice of updating application code frequently thus spawning automatic builds and checks to alert the developer earlier of possible bugs or errors.



An Example of Benefits

```
// 99 little bugs in the code.  
99 little bugs in the code.  
Take one down, patch it around.  
127 little bugs in the code_
```



GitHub Repository

We are going to start with a new GitHub repository for a Python application:

Create a new repository

A repository contains all project files, including the revision history.

Owner



Repository name *

test ✓

Great repository names are short and memorable. Need inspiration? How about [congenial-enigma](#)?

Description (optional)

testing

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: Python ▾

Add a license: None ▾



Create repository



Add to Travis CI

If you have already associated Travis CI with your GitHub account, then you can add your new repository to Travis CI. You will need to go to your GitHub settings (under your profile icon):

- Personal settings
- Profile
- Account
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Security
- Sessions
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications

Applications

[Installed GitHub Apps](#)[Authorized GitHub Apps](#)[Authorized OAuth Apps](#)

GitHub Apps augment and extend your workflows on GitHub with commercial, open source, and homegrown tools.



Travis CI

[Configure](#)

Give Travis CI Access

Select your new repository in the Travis CI configuration:

Repository access

☐ All repositories

This applies to all current *and* future repositories.

☒ Only select repositories

Select repositories ▾

Selected 1 repository

edmondb/test



Save

Cancel



Add a travis YAML file

After setting the configuration of the Travis CI utility within GitHub, your browser should be now showing a Travis CI webpage. Let's go back to GitHub and add the proper file(s) so that Travis CI can build the application.

test / or [cancel](#)

<> Edit new file

Preview

```
1 language: python
2 python:
3   - "3.6"
4 script: python run.py
```




Build your application

After the commit of our `.travis.yml` file, the Travis CI will build/attempt to run our application periodically **AND** for every subsequent commit. Let's add the `run.py` file so that we can see how a success looks:

test / or [cancel](#)

`<>` Edit new file

 Preview

```
1  print('Hello!')
```



Build your application

After the commit, a Travis CI build is triggered and thus runs the code line `python run.py` with a success message like below:

edmondb / test

 master



Build #2 was fixed >



24 secs



Brent Smith

[b5604a6 CHANGESSET →](#)

Create `run.py`



Failed Builds

If we had waited before adding the `run.py` file, or had put something in that Python script that broke (perhaps by just removing the parentheses), we would receive an error email like the following:

edmondb / test

 master

 **Build #1 failed** >

 15 secs

 Brent Smith

[ef21bce CHANGESSET →](#)

Create `.travis.yml`



Resources

- Github-Travis CI
- Travis CI
- Jenkins
- GitLab
- aws.amazon.com/devops/continuous-integration/
- CI Tutorial
- www.thoughtworks.com/continuous-integration
- codeship.com/continuous-integration-essentials
- dzone.com/articles/what-is-continuous-integration-1
- Microsoft Azure
- Atlassian Bamboo



Questions?

