

TravisCI入門

参考文献：「Github実践入門」

参考サイト：<https://knowledge.sakura.ad.jp/3754/>

発表者 吉富 楓雅

目次

- TravisCIとは？
- 実際に使ってみましょう！
- まとめ

TravisCIとは？



TravisCIとは



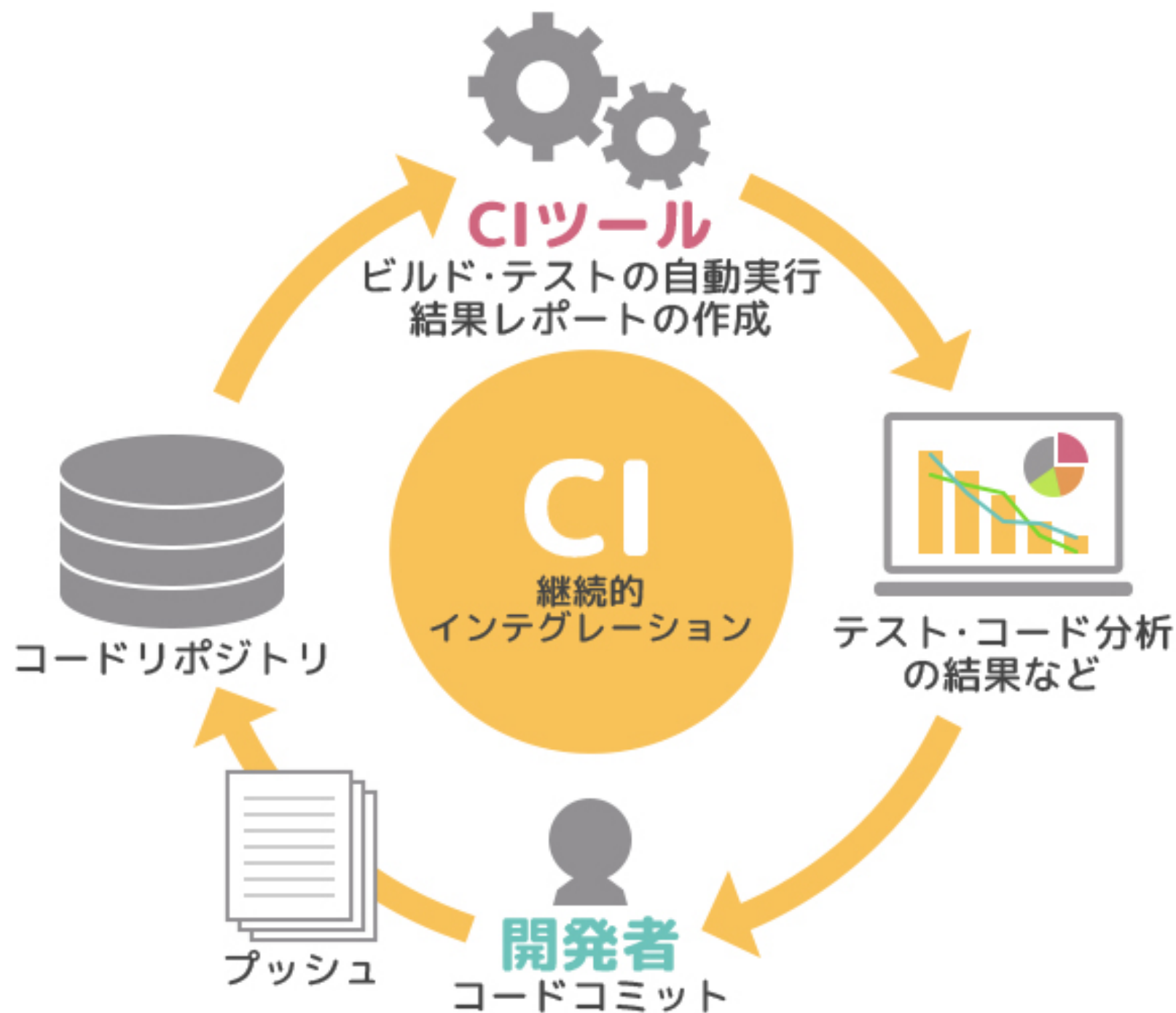
Travis CI

- オープンソースコミュニティ向けのCI(継続的インテグレーション)を無料でホストしてくれるサービス
- Ruby, PHP, Perl, Python, Java, JavaScriptなど様々な言語に対応している

CIとは

- 開発者が加えた変更や修正をコミットするたびに、自動化されたビルド(コンパイル, テスト, デプロイ)が実行することにより、欠陥を早期発見しソフトウェアの品質を維持する手法
 - * ソフトウェア開発において短期間で品質管理を行う
- XP(Extreme Programming)の1つ
 - * XP：ケントベックらによって定式化されたソフトウェア開発手法.
 - * 柔軟性の高い開発手法であるため、難易度の高い開発やビジネスの要求が刻々と変わるような状況に向いた開発手法. 事前計画よりも柔軟性を重視する.
 - * つまりアジャイル開発手法の1つである

CIのイメージ



CIの利点

1. テストの自動化

- * 手動でのテスト工数を削減し開発工数を増加させることが可能

2. 開発状況の把握

- * ビルド結果の履歴を時系列に並べて視認可能な状態にすることにより、ビルドエラーの発生状況や自動テストのテスト結果などの状況を把握することができる

3. ソースコードの静的解析

- * コーディングルールやバグのチェックツールを組み込むことにより、開発中のソースコードの問題をあぶり出すことができる

それでは実際に使って
みましょう！



。。。の前に



GitHubアカウントの作成

- ・ターミナルに以下のコマンドを入力後, GitHubのサイトへ行きアカウント作成
- ・ユーザ名とメールアドレスは各個人のものを使用してください

```
OSSnoMacBook-Pro:~ FugaYoshitomi$ git config --global user.name "Fuga Yoshitomi"  
[OSSnoMacBook-Pro:~ FugaYoshitomi$ git config --global user.email "s206299@center.wakayama-u.ac.jp"  
OSSnoMacBook-Pro:~ FugaYoshitomi$ git config --global color.ui auto  
[OSSnoMacBook-Pro:~ FugaYoshitomi$
```

SSH Keyの設定

- 以下のコマンドを入力

```
OSSnoMacBook-Pro:~ FugaYoshitomi$ ssh-keygen -t rsa -C "s206299@center.wakayama-u.ac.jp"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/FugaYoshitomi/.ssh/id_rsa):
Created directory '/Users/FugaYoshitomi/.ssh'.
Enter passphrase (empty for no passphrase): _____
Enter same passphrase again: _____
Your identification has been saved in /Users/FugaYoshitomi/.ssh/id_rsa.
Your public key has been saved in /Users/FugaYoshitomi/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:fGf5uG2XwsIqYWCxQoJH3uLI6ltfqSeI9ppDSW8zGd0 s206299@center.wakayama-u.ac.jp
The key's randomart image is:
+---[RSA 2048]---+
|...
|.o.o .
|.= o +
|. + + = E .
|o.+ = . S . +
|.o * o.. o o
|o o.+ .o. . o . .
|. + o...oo o =...
|.=. oo ... o.o.
+---[SHA256]---+
OSSnoMacBook-Pro:~ FugaYoshitomi$
```

パスフレーズ作成・入力

公開鍵の登録

- GitHubのサイトに戻ってアカウント設定(Account Settings)のボタンを押す
 - * 「SSH Keys」のメニューを開く
- 「Add SSH key」を押すと入力欄が表示される
 - * Titleは適当な鍵の名前を, Keyにはid_rsa.pubの内容をコピーして貼り付ける
- ターミナルに[`pbcopy < ~/.ssh/id_rsa.pub`]と入力することで公開鍵のキーをコピーできる
 - * [ssh-rsa][公開鍵の内容][自身が登録したメールアドレス]

認証されたかを確認

- 無事に登録されると登録したアドレスにメールが届く
- 実際に動作確認したい時は以下のコマンドを入力

```
OSSnoMacBook-Pro:~$ ssh -T git@github.com  
Warning: Permanently added the RSA host key for IP address '192.30.255.112' to the list of known hosts.  
Hi hyo-hyo-tommy! You've successfully authenticated, but GitHub does not provide shell access.  
OSSnoMacBook-Pro:~$
```

この表示が出ていれば成功

ではいいよいよTravisCI
を使ってみましょう！



リポジトリの作成

- 以下のコマンドを入力してローカルリポジトリを作成する

```
mkdir “ディレクトリ名”  
cd “ディレクトリ名”  
git init  
git add hello.html  
git commit -m “add new file”  
git status  
git push origin master
```

← 適当なファイルを作成
ディレクトリに追加

← この後Githubの
ユーザ名とパスワードを
聞かれるので入力

TravisCIに登録

- <http://travis-ci.org/>にアクセス
 - * sign upのボタンを押してGithubのアカウントでログイン
- 先ほど作成したリポジトリが表示されているかを確認
- TravisCIにかかけたいものにチェックをいれておく

.travis.ymlファイルの作成

- 以下のコマンドを入力して.travis.ymlファイルを作成

```
cd “ディレクトリ名”  
touch .travis.yml  
vim .travis.yml
```

- vimコマンドの使い方については以下の通り

```
a: 編集開始  
Escキー: 編集終了  
ZZ: 上書き保存
```

- これを15ページの三行目以降と同様にリポジトリに追加

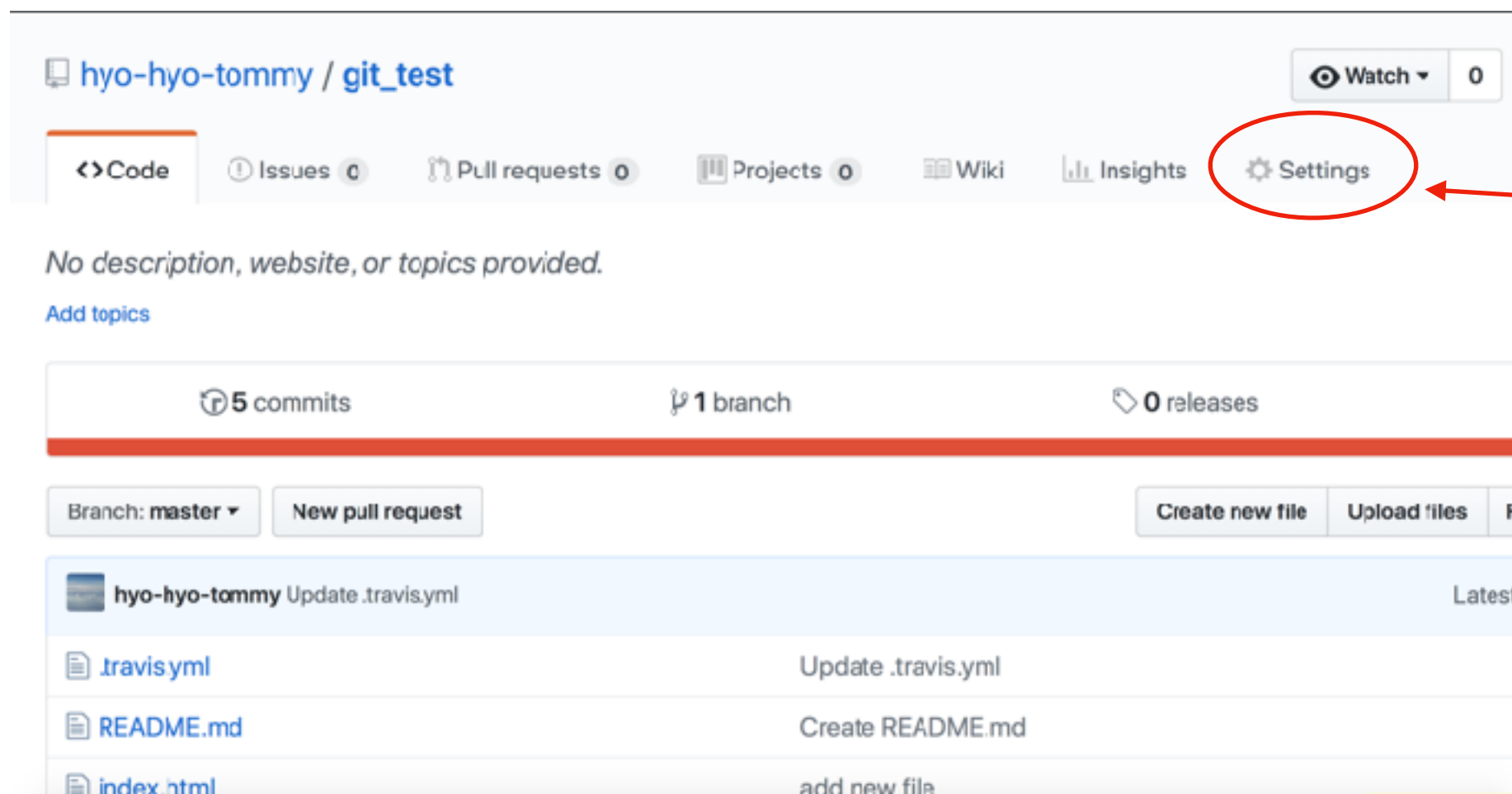
.travis.yml ファイルの記述

- 記述については例えば以下の通り
 - * サイトによって書き方が異なります

```
1  language: node_js
2
3  node_js:
4    - stable
5
6  script:
7    - npm run build
8
9  branches:
10   only:
11     - master
12
13  notifications:
14    email:
15      on_success: never
16      on_failure: always
17
18  deploy:
19    # このフィールドに次回以降デプロイ設定を書く
20
```

Githubとの連携

- Githubに戻り, 作成したリポジトリのページに移動
 - * Settingsボタンをクリック



Test Hookの設定

- Integrations&Servicesをクリック
 - * TravisCIにチェックマークが入っていることを確認してクリック
- ユーザ名・トークンを入力
 - * トークンはTravisCIのプロフィールページ左側に記載されている
- ドメインは特に何も記述しなくても動きました
- Update Servicesをクリック

以上でTravisCIの設定は
終了です
テスト内容は自由に記述
してください



例えばこんな感じ

The screenshot shows the Travis CI interface for the repository `hyo-hyo-tommy / git_test`. The build status is `unknown`. The left sidebar shows a list of repositories, with `hyo-hyo-tommy/git_test` selected. The main area shows the build details for the `master` branch, where the build failed at step 1. The job log shows the following output:

```
1 Worker information
2 Build system information
3
403
404 Network availability confirmed.
405
406 Running apt-get update by default has been disabled.
407 You can opt into running apt-get update by setting this in your .travis.yml file:
408
409 addons:
410   apt:
411     update: true
```

何かエラーが出ているっぽい。。。