

# 最速でJavaエンジニア になって稼ぐ

---

2019/01/12

人生逃げ切りオンラインサロン内



# 目次的なやつ

---

- CI(継続的インテグレーション)とは
- CD(継続的デリバリー)とは
- Jenkins
- CircleCI
- Jenkinsを用いたテスト自動化
- Jenkins実践

CI(継続的インテグレーション)とは



- CI(Continuous Integration)とは日本語で**継続的インテグレーション**と呼ばびますが、意味不明なので、CIとはこういうものぐらいで覚えておいてもらえれば良いと思います。
- シンプルに言うならば「**ソフトウェアの変更を常にテストするモノ**」ぐらいの認識を持っていれば良いと思います。
- CIは手法の事で、特定のツールを指す言葉ではない為、CIツールと呼ばれるツールが沢山あります。どれを使っても同じ事は実現できるので、後はツールの提供している機能などで取捨選択します。
- 実際にCIツールでやる事もシンプルです。
  - 1.ソフトウェアの変更が行われる
  - 2.行われた変更を検出する。
  - 3.ソフトウェアのテストやビルドを設定された内容に基づいて行う。
  - 4.成否を連絡・通知する。

CD(継続的デリバリー)とは



- CD(Continuous Delivery)とは日本語で**継続的デリバリー**と呼ばれますが、CIと同じく意味不明なので以下略。
- CDはCIを拡張したようなもので「**CIの後に特定の環境へリリースする**」という一連の流れをCDと呼びます。変更をテストして、OKだったらリリース。
- CD自体はCIツールで大体実現できる為、CDツールというものはありません。正確に言うとCI出来るなら、CDもやりたいのが殆どな為です。
- ただ、CDは意外と難しい側面があります。というのもCIで自動テストするまでは良いですが、実際にリリースするかどうかの判断はまた別だったりするからです。
- 結局CIは良いけど、CDはダメというケースも業務系システムでは良くあるので利用されない場合もあります。
- ただWeb系のサービスやシステムでは積極的に活用されている事例もあります。皆さんが使っているGitHubは1日に200回近くリリースデプロイが実行されているそうです。ぶっちゃけ想像できないくらい凄いです。

Jenkins



- JenkinsはCI/CDを実現するためのいわゆるCIツールです。
- 非常に色々なケースに対応しており、汎用性が高いのですが、何でもできるという都合上、利用者側で設定したりするのが微妙に難しいという難点を抱えています。慣れてくると全く気にならないのですが、初見では取っつきにくいかもしれません。
- またCIツールとしては古くからあるため、知見やプラグインも多い為、とりあえずJenkinsというケースも多々あります。
- 昔はHudsonと呼ばれていて、今でもJenkinsのコード内にはその名残が残っています。またデザインについても都度ブラッシュアップされており、バージョンアップも盛んです。まず押さえておくべきCIツールです。
- またCIはその特性上、別に言語などに縛られません。なんでもいけます。
- スケジューラーとしての機能もあるので、定時起動バッチみたいなことも出来ます。
- とにかく多機能すぎるので、やりたいと思ったことは大体出来てしまいます。凄い。



CircleCI

- CircleCIはクラウドサービスとして提供されているCIツールです。
- Jenkinsはオンプレ型なので自分でインストールする必要がありますが、CircleCIはそういった要素も不要なので、お手軽に使えます。またWebサービスで提供されている事もあってか、機能も洗練されUI面も優秀です。
- ただもちろんJenkinsのように自由に何でもというわけではないのと、Webサービスなので利用料が別途かかります。こころへんは案件の条件と相談になりますが、最近はCircleCIを利用するケースも増えてきています。
- 特に大きなメリットとしてはJenkinsと違い、すぐに利用可能であることや、デフォルトでおおよそ利用する機能は入っている事。自社で全て管理しなくてもいいこと、などがあります。
- 個人的には今後はこういうサービスが強くなるのかな、と思ったりしてます。



Jenkinsを用いたテスト自動化

- 今まで作成したアプリにあるテストコードはEclipseを開いて手動実行していましたが、それを止めます。完全自動実行に切り替える為にJenkinsを導入します。
- メリットは「自分が実行するのを忘れていても、必ず実行してくれる」という点に尽きます。手動実行とは属人性が高い操作なので、勝手にやってくれるようにするのが最も良いです。自分を信じない、大事です。
- ただし手動実行をしなくなるわけではないです。自動テスト自体は自分の開発したコードのテストも兼ねているので、何度も実行しながら開発は進みます。しかし、開発が進むとテストコードも増えるので、常に全てのテストを実行するとコストがかかるので、自分の修正した部分だけをテストします。
- すると意図しない時に他のテストが落ちたりします(順序依存とか)。そういったものを検出するのも一つの目的になります。



# Jenkins実践

ローカルに環境を作るところからやります。  
参考サイトを元に皆さんもやってみてください。



Jenkinsをインストールして使ってみよう[Mac/Linux/Windows]  
<https://www.buildinsider.net/enterprise/jenkins/001>

[Jenkins] Windows7にJenkinsをインストールする方法まとめ  
<https://qiita.com/koara-local/items/5fea213a9cf7033a95c1>

GitHubとJenkinsのWebhook連携  
[https://qiita.com/tz2i5i\\_ebinuma/items/528ea5163bb2df379852](https://qiita.com/tz2i5i_ebinuma/items/528ea5163bb2df379852)

次回までの宿題



Jenkinsの導入と実践完了まで。  
次回からはAWS上に環境作り始めますので、AWSアカウントの取得もお願いします。

おしまい