# 클로저 개발팀을 위한 지속적인 통합

김만명

# 소개

- https://youtu.be/wiRmVsTarAI?t=3m22s
- https://github.com/infokraft

# 무엇을 발표할까?

- CI(Continuous Integration; 지속적인 통합) 서비스 도입 과정과 결과
- 클로저 프로젝트 경험

# 최근 프로젝트

https://github.com/infokraft/toxfree/graphs/commit-activity

- 서버: Ring, Jetty, Compojure, MySQL
- 클라이언트: Reagent, re-frame, Datatables
- 플랫폼: Github, AWS

# CI 도입 이유

- 배포가 복잡
- 한 사람이 배포를 도맡는 현상
- 버그 때문에 개발이 지연되는 경험

# CI 선택

## Github에서 많이 사용하는 CI 서비스

- Travis CI : https://travis-ci.com/plans
- Circle CI : https://circleci.com/pricing/

Add projects - CircleCI | mmkim78/guestbook | Continuous Integration and De

Manmyung

https://circleci.com

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t...

circleci

Features | Enterprise | Pricing | Customers | Support | More | Go to app

CircleCI 2.0 is now in closed beta — **see what's coming and request access.**

# Ship better code, faster.

The modern **continuous integration and delivery** platform
that software teams **love** to use.

Sign Up For Free

Add projects - CircleCI | mmkim78/guestbook | Signup - CircleCI | Manmyung

https://circleci.com/signup/

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t...

circleci

# You're almost there.
# Now let's add some code.

Signing up with CircleCI is **free**. Next, you'll be taken to GitHub or Bitbucket to authenticate so you can start shipping faster.

Authorize GitHub | Authorize Bitbucket

By clicking on "Authorize GitHub" or "Authorize Bitbucket" you are agreeing to our Terms of Service and Privacy Policy.

Add projects - CircleCI | Authorize Circle | GitHub | Manmyung

GitHub, Inc. [US] https://github.com/login/oauth/authorize?client_id=78a2ba87f071c28e65bb&redirect_uri=https%3A%2F%2Fcircleci.com%2Fauth%...

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t...

Search GitHub | Pull requests | Issues | Gist | ToDo

# Authorize application

Circle by @circleci would like permission to access your account

## Review permissions

**Personal user data**
Email addresses (read-only)

**Repositories**
Public and private

Authorize application

### Circle

CircleCI is a hosted Continuous Integration and Delivery service, trusted by many companies to access their code safely and responsibly, including Stripe, Kickstarter, Shopify, Cisco, Salesforce, Red Bull, and Tapjoy. See our commitment to privacy and security: https://circleci.com/privacy, and read about the permissions we need and why we need them: https://circleci.com/docs/github-permissions

Visit application's website

Sign in now to use ZenHub

Add projects - CircleCI

Manmyung

https://circleci.com/add-projects

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t...

BUILDS
INSIGHTS
ADD PROJECTS
TEAM
ACCOUNT SETTINGS
DOCS
SUPPORT
CHANGELOG
LOGOUT

# Add Projects

Reload Organizations

CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

To kick things off, you'll need to pick some projects to build:

1 Choose an organization that you are a member of.

2 Choose a repo to add to CircleCI. We'll start a new build for you each time someone pushes a new commit.

You can also follow a project that's already been added to CircleCI. You'll see your followed projects in Builds and Insights.

GitHub | Bitbucket

Are you missing an organization? You or an admin may need to enable CircleCI for your organization in GitHub's application permissions. Refresh this list after you have updated permissions.

mmkim78

infokraft

Get started by selecting your GitHub username or organization.

Choose a repo you want to test and we'll do the rest!

Add projects - CircleCI

Manmyung

https://circleci.com/add-projects

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t...

BUILDS
INSIGHTS
ADD PROJECTS
TEAM
ACCOUNT SETTINGS
DOCS
SUPPORT
CHANGELOG
LOGOUT

# Add Projects

Reload Organizations

CircleCI helps you ship better code, faster. Let's add some projects on CircleCI.

To kick things off, you'll need to pick some projects to build:

1 Choose an organization that you are a member of.

GitHub | Bitbucket

Are you missing an organization? You or an admin may need to enable CircleCI for your organization in GitHub's application permissions. Refresh this list after you have updated permissions.

mmkim78

infokraft

2 Choose a repo to add to CircleCI. We'll start a new build for you each time someone pushes a new commit.

You can also follow a project that's already been added to CircleCI. You'll see your followed projects in Builds and Insights.

Linux | OS X

Filter projects...

Show Forks

toxfree — Build project

guestbook — Build project

mmkim78/toxfree #2 - CircleC

Manmyung

https://circleci.com/gh/mmkim78/toxfree/2

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t... | Pull Requests · info... | CircleCI

Builds » mmkim78 » toxfree » master » build 2

Rebuild | Project Settings

BUILDS
INSIGHTS
ADD PROJECTS
TEAM
ACCOUNT SETTINGS
DOCS
SUPPORT
CHANGELOG
LOGOUT

FIXED

Finished: 21 min ago (02:16)

Previous: 1

Parallelism: 1x out of 1x

Queued: 01:10 waiting + 00:00 in queue

Triggered by: Manmyung Kim (pushed 6922274)

COMMITS (1)

manmyung 6922274 .

Test Summary | Queue (01:10) | Debug via SSH | Artifacts | circle.yml | Build Timing

Set Up Test Summary

Show containers: All (1) Successful (1) Failed (0)

0 (02:14)

Add Containers +

INFRASTRUCTURE

Starting the build 00:14

Start container *config* 00:14

Enable SSH 00:04

CHECKOUT

Restore source cache *cache*

Checkout using deploy key: 0f:a2:c6:b9:de:09:55:10:0e:b8:2b:d1:59:81:db:76 *config* 00:08

mmkim78/toxfree #2 - CircleC

Manmyung

https://circleci.com/gh/mmkim78/toxfree/2

Gmail | Clojure - cheatsheet | 네이버 영어사전 | Google 번역 | Clojure_CI | gnab/remark: A sim... | Infokraft | Toxfree-Test | Toxfree-Release | Boards · infokraft/t... | Pull Requests · info... | CircleCI

Builds » mmkim78 » toxfree » master » build 2

Rebuild | Project Settings

BUILDS | INSIGHTS | ADD PROJECTS | TEAM | ACCOUNT SETTINGS | DOCS | SUPPORT | CHANGELOG | LOGOUT

0 (02:14)

Add Containers +

MACHINE

| | | |
|---|---|---|
| Configure the build | *cache* | 00:07 |
| Restore cache | *cache* | 00:16 |

DEPENDENCIES

| | | |
|---|---|---|
| $ lein deps | *inference* | 00:06 |

DATABASE

| | | |
|---|---|---|
| Save cache | *cache* | 00:05 |

TEST

| | | |
|---|---|---|
| lein test | *inference* | 00:24 |

TEARDOWN

| | | |
|---|---|---|
| Collect test metadata | | 00:04 |
| Collect artifacts | | 00:18 |
| Disable SSH | | 00:00 |

# 알림

## 1. 이메일

## 2. 채팅

- 지원 서비스
- 사용예

## 3. 상태 배지

- 사용예

## 4. 웹훅(Webhook)

# CircleCI 설정 수정

## circle.yml

https://circleci.com/docs/manually/

# 데이터베이스

circle.yml

```
database:
  override:
    - mysql -u root -vvf < create-test-db.sql
```

create-test-db.sql

```
CREATE DATABASE `toxfree_test` character set utf8;
```

# 환경 설정 파일

circle.yml

```
checkout:
  post:
    - cp ./profiles-default.clj ./profiles.clj
```

profiles-default.clj

```
{:profiles/dev
 {:env
  {:db {:subprotocol "mysql"
        :subname "//localhost:3306/toxfree"
        :user "root"
        :password ""
        :useSSL false}}}}
```

# 테스트

circle.yml

```
test:
  override:
    - lein trampoline test
    - lein doo phantom test once
  post:
    - lein clean
    - lein compile
    - lein cljsbuild once dev
```

# 테스트

지속적인 통합에서 테스트의 역할

# 서버

```
HTTP 요청 -->       -->       -->
            핸들러     쿼리단     DB
HTTP 응답 <--       <--       <--
```

- 단위 테스트: 핸들러 테스트, DB 테스트
- 우리가 테스트하고 싶은 것: 어떤 요청을 보내면 어떤 응답이 오나?

# 테스트

지속적인 통합에서 테스트의 역할

# 서버

```
HTTP 요청 -->       -->       -->
            핸들러     쿼리단     DB
HTTP 응답 <--       <--       <--
```

- 단위 테스트: 핸들러 테스트, DB 테스트
- 통합 테스트: HTTP 요청과 응답을 이용해 서버단 테스트

routing_test.clj

```
(deftest depts-routing
  (with-test-db
    (let [{dept-id "dept-id" :as create-res}
          (request :post "/dept/create" {"dept" "연구1실"})]
      (testing "/dept/create"
        (is (= {"dept-id" dept-id, "dept" "연구1실"}
               create-res)))
      (testing "/depts/read"
        (is (= [["dept-id" "dept"] [dept-id "연구1실"]]
               (request :get "/depts/read"))))
      (testing "/dept/update/:dept-id"
        (is (= {"dept-id" dept-id, "dept" "연구6실"}
               (request :post (str "/dept/update/" dept-id) {"dept" "연구6실"}))))
      (testing "update 확인"
        (is (= [["dept-id" "dept"] [dept-id "연구6실"]]
               (request :get "/depts/read"))))
      (testing "/dept/delete/:dept-id"
        (is (= dept-id
               (request :delete (str "/dept/delete/" dept-id)))))
      (testing "delete 확인"
        (is (= [["dept-id" "dept"]]
               (request :get "/depts/read")))))))
```

**routing_test.clj**

```
(deftest depts-routing
  (with-test-db
    (let [{dept-id "dept-id" :as create-res}
          (request :post "/dept/create" {"dept" "연구1실"})]
      (testing "/dept/create"
        (is (= {"dept-id" dept-id, "dept" "연구1실"}
               create-res)))
      (testing "/depts/read"
        (is (= [["dept-id" "dept"] [dept-id "연구1실"]]
               (request :get "/depts/read"))))
      (testing "/dept/update/:dept-id"
        (is (= {"dept-id" dept-id, "dept" "연구6실"}
               (request :post (str "/dept/update/" dept-id) {"dept" "연구6실"}))))
      (testing "update 확인"
        (is (= [["dept-id" "dept"] [dept-id "연구6실"]]
               (request :get "/depts/read"))))
      (testing "/dept/delete/:dept-id"
        (is (= dept-id
               (request :delete (str "/dept/delete/" dept-id)))))
      (testing "delete 확인"
        (is (= [["dept-id" "dept"]]
               (request :get "/depts/read")))))))
```

- 테스트 DB 를 대상으로 실행
- 한 테스트가 끝나면 테스트 DB 원상복구

**routing_test.clj**

```
(deftest depts-routing
  (with-test-db
    (let [{dept-id "dept-id" :as create-res}
          (request :post "/dept/create" {"dept" "연구1실"})]
      (testing "/dept/create"
        (is (= {"dept-id" dept-id, "dept" "연구1실"}
               create-res)))
      (testing "/depts/read"
        (is (= [["dept-id" "dept"] [dept-id "연구1실"]]
               (request :get "/depts/read"))))
      (testing "/dept/update/:dept-id"
        (is (= {"dept-id" dept-id, "dept" "연구6실"}
               (request :post (str "/dept/update/" dept-id) {"dept" "연구6실"}))))
      (testing "update 확인"
        (is (= [["dept-id" "dept"] [dept-id "연구6실"]]
               (request :get "/depts/read"))))
      (testing "/dept/delete/:dept-id"
        (is (= dept-id
               (request :delete (str "/dept/delete/" dept-id)))))
      (testing "delete 확인"
        (is (= [["dept-id" "dept"]]
               (request :get "/depts/read")))))))
```

```
(defmacro with-test-db [& body]
  `(jdbc/with-db-transaction [t-conn# test-db*]
     (jdbc/db-set-rollback-only! t-conn#)
     (with-redefs [db* t-conn#]
       ~@body)))
```

routing_test.clj

```
(deftest depts-routing
  (with-test-db
    (let [{dept-id "dept-id" :as create-res}
          (request :post "/dept/create" {"dept" "연구1실"})]
      (testing "/dept/create"
        (is (= {"dept-id" dept-id, "dept" "연구1실"}
               create-res)))
      (testing "/depts/read"
        (is (= [["dept-id" "dept"] [dept-id "연구1실"]]
               (request :get "/depts/read"))))
      (testing "/dept/update/:dept-id"
        (is (= {"dept-id" dept-id, "dept" "연구6실"}
               (request :post (str "/dept/update/" dept-id) {"dept" "연구6실"}))))
      (testing "update 확인"
        (is (= [["dept-id" "dept"] [dept-id "연구6실"]]
               (request :get "/depts/read"))))
      (testing "/dept/delete/:dept-id"
        (is (= dept-id
               (request :delete (str "/dept/delete/" dept-id)))))
      (testing "delete 확인"
        (is (= [["dept-id" "dept"]]
               (request :get "/depts/read")))))))
```

```
(use-fixtures
  :once
  (fn [f]
    (init/migrate test-db*)
    (f)))
```

**routing_test.clj**

```
(deftest depts-routing
  (with-test-db
    (let [{dept-id "dept-id" :as create-res}
          (request :post "/dept/create" {"dept" "연구1실"})]
      (testing "/dept/create"
        (is (= {"dept-id" dept-id, "dept" "연구1실"}
               create-res)))
      (testing "/depts/read"
        (is (= [["dept-id" "dept"] [dept-id "연구1실"]]
               (request :get "/depts/read"))))
      (testing "/dept/update/:dept-id"
        (is (= {"dept-id" dept-id, "dept" "연구6실"}
               (request :post (str "/dept/update/" dept-id) {"dept" "연구6실"}))))
      (testing "update 확인"
        (is (= [["dept-id" "dept"] [dept-id "연구6실"]]
               (request :get "/depts/read"))))
      (testing "/dept/delete/:dept-id"
        (is (= dept-id
               (request :delete (str "/dept/delete/" dept-id)))))
      (testing "delete 확인"
        (is (= [["dept-id" "dept"]]
               (request :get "/depts/read")))))))
```

- 19개의 테스트, 94개의 단정문

routing_test.clj

```
(deftest depts-routing
  (with-test-db
    (let [{dept-id "dept-id" :as create-res}
          (request :post "/dept/create" {"dept" "연구1실"})]
      (testing "/dept/create"
        (is (= {"dept-id" dept-id, "dept" "연구1실"}
               create-res)))
      (testing "/depts/read"
        (is (= [["dept-id" "dept"] [dept-id "연구1실"]]
               (request :get "/depts/read"))))
      (testing "/dept/update/:dept-id"
        (is (= {"dept-id" dept-id, "dept" "연구6실"}
               (request :post (str "/dept/update/" dept-id) {"dept" "연구6실"}))))
      (testing "update 확인"
        (is (= [["dept-id" "dept"] [dept-id "연구6실"]]
               (request :get "/depts/read"))))
      (testing "/dept/delete/:dept-id"
        (is (= dept-id
               (request :delete (str "/dept/delete/" dept-id)))))
      (testing "delete 확인"
        (is (= [["dept-id" "dept"]]
               (request :get "/depts/read")))))))
```

- 서버 수정할 때 안정감
- 최신 상태의 문서

# 테스트

## 클라이언트

```
        -->     -->       --> HTTP 요청
브라우저     DOM     처리단
        <--     <--       <-- HTTP 응답
```

# 테스트

## 클라이언트

```
        -->     -->       --> HTTP 요청
브라우저     DOM     처리단
        <--     <--       <-- HTTP 응답
```

- 단위 테스트: 처리단에서 변환로직
- 10개의 테스트, 49개의 단정문

# 테스트

## 클라이언트

```
        -->     -->       --> HTTP 요청
브라우저     DOM     처리단
        <--     <--       <-- HTTP 응답
```

- 클라이언트 수정시 안정감

# 배포

## 개발 절차

- Git 협업 방법
- https://github.com/infokraft/toxfree/pull/138
- https://github.com/infokraft/toxfree/pull/261

마스터 브랜치에 머지되면 테스트 서버에 바로 배포해서 확인

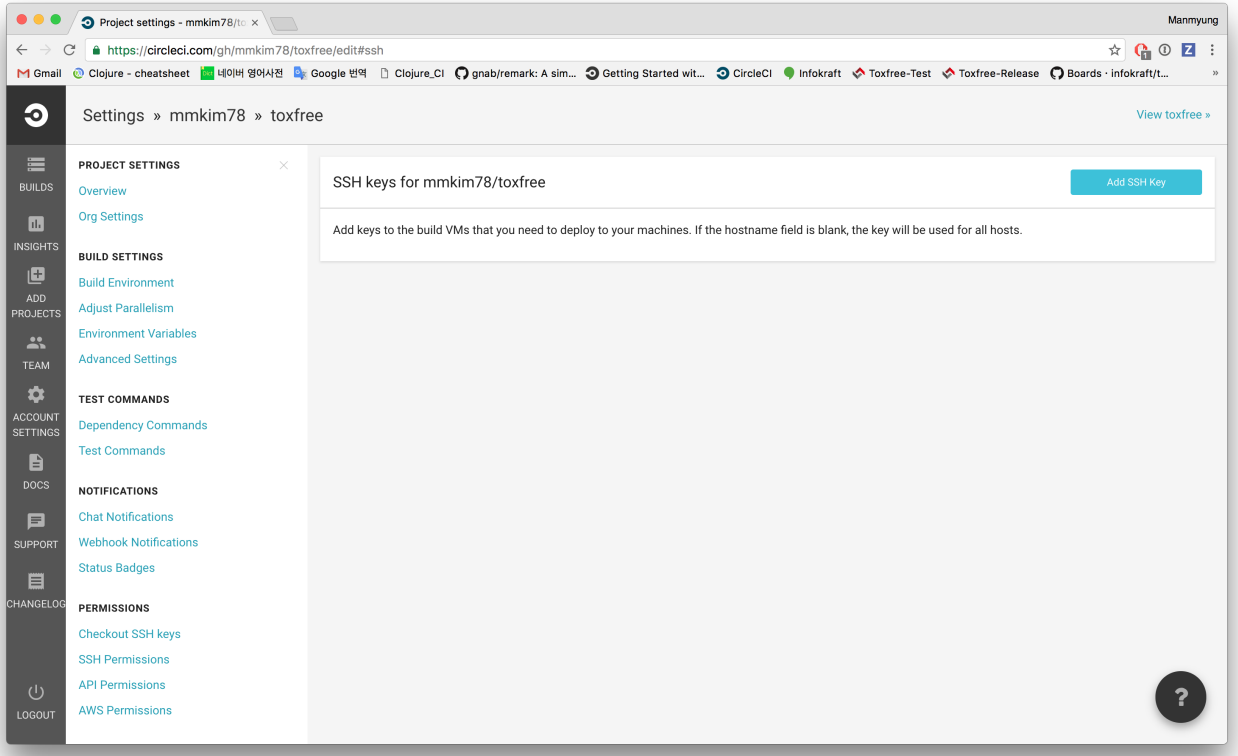
# 배포

## 다양한 배포 방법

https://circleci.com/docs/

# 배포

## SSH 키 세팅


Project settings - mmkim78/to
Manmyung
https://circleci.com/gh/mmkim78/toxfree/edit#ssh
Gmail
Clojure - cheatsheet
네이버 영어사전
Google 번역
Clojure_CI
gnab/remark: A sim...
Getting Started wit...
CircleCI
Infokraft
Toxfree-Test
Toxfree-Release
Boards · infokraft/t...
Settings » mmkim78 » toxfree
View toxfree »
BUILDS
INSIGHTS
ADD PROJECTS
TEAM
ACCOUNT SETTINGS
DOCS
SUPPORT
CHANGELOG
LOGOUT
PROJECT SETTINGS
Overview
Org Settings
BUILD SETTINGS
Build Environment
Adjust Parallelism
Environment Variables
Advanced Settings
TEST COMMANDS
Dependency Commands
Test Commands
NOTIFICATIONS
Chat Notifications
Webhook Notifications
Status Badges
PERMISSIONS
Checkout SSH keys
SSH Permissions
API Permissions
AWS Permissions
SSH keys for mmkim78/toxfree
Add SSH Key
Add keys to the build VMs that you need to deploy to your machines. If the hostname field is blank, the key will be used for all hosts.

# 배포

## 테스트 서버

circle.yml

```
deployment:
  master:
    branch: master
    commands:
      - lein clean
      - lein uberjar 2> error.log
      - scp target/toxfree-0.1.0-SNAPSHOT-standalone.jar ubuntu@111.111.111.111:~/target
      - ssh ubuntu@111.111.111.111 "sudo pkill java;sudo java -jar target/toxfree-0.1.0-SNAPSHOT-stand
        alone.jar 80 &> log/log-$CIRCLE_BUILD_NUM.txt &"
```

- 마스터 브랜치에 커밋이 되면 배포
- $CIRCLE_BUILD_NUM
- 테스트 서버에서 확인

# 배포

## 릴리즈 서버

circle.yml

```
deployment:
  release:
    tag: /release-.*/
    commands:
      - lein clean
      - lein uberjar 2> error.log
      - scp target/toxfree-0.1.0-SNAPSHOT-standalone.jar ubuntu@222.222.222.222:~/target/toxfree-0.1.0
        -SNAPSHOT-standalone-$CIRCLE_TAG.jar
      - ssh ubuntu@222.222.222.222 "sudo pkill java;sudo java -jar target/toxfree-0.1.0-SNAPSHOT-stand
        alone-$CIRCLE_TAG.jar 80 $CIRCLE_TAG &> log/log-$CIRCLE_TAG.txt &"
```

- `release-*` 태그가 푸시되면 배포
- `$CIRCLE_TAG`

# CircleCI 메모리 문제

## 배포에서 빌드 깨지는 문제

https://circleci.com/gh/infokraft/toxfree/402

## 참고문서

https://circleci.com/docs/oom/

## 해결

circle.yml

```
machine:
  environment:
    _JAVA_OPTIONS: "-Xms512m -Xmx1024m"
```

# 느낀점

- 쉬운 배포
- 오류를 빨리 발견
- 엔드 투 엔드(End to End) 테스트 기대: 참고
- 팀을 위한 툴?

# 감사합니다