

V&V프로젝트 발표-2

CTIP 환경 구축(1)

공민정, 이규은, 김태형, 최지현

- 1 CI
- 2 Code Configuration Management
- 3 Requirements Management, Bug Tracking
- 4 Team Communication
- 5 IDE
- 6 Automatic Build
- 7 Unit Testing
- 8 기타 연동
- 9 총정리

INDEX

01 CI(Continuous Integration):



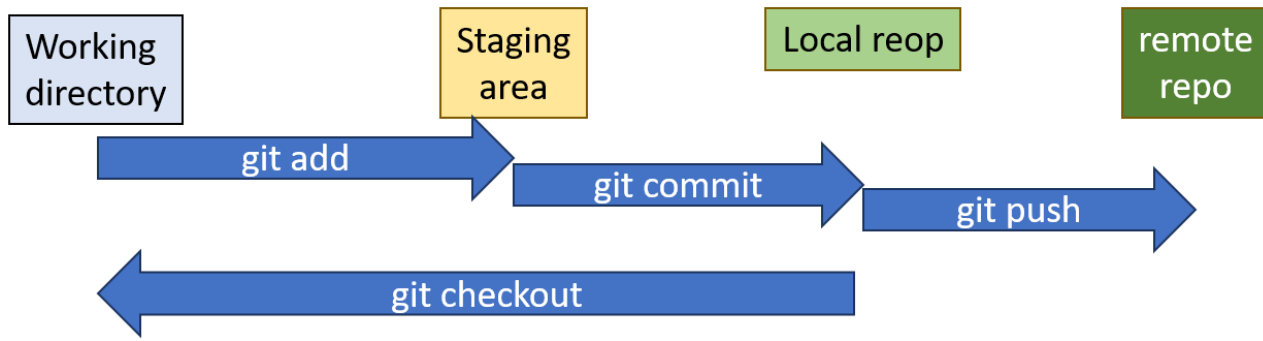
Jenkins

Jenkins

- ✓ 다수의 개발자들이 하나의 프로그램을 개발할 때, 버전 충돌 방지를 위해 지속적으로 통합 서비스를 제공하는 CI 툴
- ✓ 정기적인 빌드+git과 같은 버전 관리 시스템과 연동하여 commit을 감지하면 자동적으로 자동화 테스트가 포함된 빌드 작동
- ✓ 500여 가지가 넘는 플러그인 제공

02 Code Configuration Management:

- 형상관리도구란?
 - 코드의 업데이트를 적용하고 기록하여 여러 개발자의 협업을 가능하게 하고 이전 버전으로 되돌리는 등의 기능을 제공한다.
- Git 과 SVN의 차이점
 - SVN은 각 개발자의 commit이 중앙 서버의 코드에 즉시 업데이트되어 다수의 협업이 어렵다는 단점이 있다.
 - 반면 Git은 개발자 각자 고유의 로컬 저장소를 가지며 로컬 저장소에 commit하여 테스트를 거친 코드를 본 코드에 적용(push)할 수 있다.
- Git 의 코드 적용 과정





- 용어설명
 - working directory : 개인의 작업공간으로 아직 local repo에 commit 되기 전의 path.
 - staging area : 저장소에 커밋하기 전에 커밋을 준비하는 위치.
 - local repo : 커밋된 모든 파일을 포함하는 로컬 저장소.
 - remote repo : 로컬 저장소에서 push된 내용만을 저장하는 원격 저장소.
 - git commit : 업데이트 한 내용을 저장소에 적용하는 과정.
 - git pull : 원격저장소의 작업내용을 바탕으로 로컬 저장소 및 working directory의 모든 코드를 새로 생긴하는 것.
 - git add : working directory의 파일을 commit하기 전에 임시로 추가하는 것.
 - git fetch : 원격저장소의 작업내용으로 로컬 저장소를 갱신하는 것.

03 Requirements Management

- 의미: 요구 공학 프로세스나 시스템 개발 과정에서 발생하는 요구사항의 변화를 이해하고 통제하는 프로세스
- 비즈니스 자체의 변화 or 시스템의 이해도가 점차 높아짐
 - > 새로운 요구사항 추가될 수 있음
- 서로 다른 관점 때문에 다양한 요구사항 발생 가능
 - > 모순 발생 가능

⇒ 요구사항은 불완전, 불일치하다.

∴ 요구사항 관리가 필요하다

03

Redmine - Requirements Management, Bug Tracking

기능

- 웹 기반 프로젝트 관리 프로그램
- Issue Tracking System, Bug Tracking System
- Ruby on Rails로 작성됨
- 오픈 소스 ALM 솔루션에서 프로젝트 관리, 요구사항 관리, 이슈 관리, 변경 관리, 테스트 관리 등
- 직관적인 UI
- 무료



- ✓ 다중 프로젝트 지원
- ✓ 유연한 역할기반 접근제어(RBAC)
- ✓ 다양한 모듈 제공
 - 일감관리, 시간추적, 뉴스, 문서, 파일, 위키, 저장소, 게시판, 달력, 간트차트
- ✓ 알림, 이메일 알람 기능
- ✓ 형상 관리 소프트웨어와의 통합
- ✓ 다중 LDAP 기반 인증 지원
- ✓ 다국어 지원
- ✓ 플러그인 기능 지원
- ✓ 여러 데이터베이스 지원
 - : MySQL, PostgreSQL, MS-SQL, SQLite
- ✓ 관련 모바일 앱

04 Team Communication: slack

- Slack 의 특징과 주요 기능

1. 무료 버전으로 가입해도 인원수와 기능 제한 없음
2. 단톡방과 유사한 '채널' 생성 가능
3. 알람 설정 기능의 세분화 : 해당채널 모든 내용, 나의 아이디만 언급할 경우, 사전에 지정해둔 중요 단어 언급할 경우, 1:1 메시지인 경우, 알림 끄
4. 검색 용이성
5. 파일 공유와 파일에 대한 커뮤니케이션 가능
6. 기존에 사용 중인 서비스의 알림 통합 ex) google drive, dropbox, github, trello
7. 파워풀한 API 제공
 - > 개발자들이 원하는 정보를 slack을 통해 알림을 받을 수 있음

∴ 업무와 소통을 하나로 묶고 사용자의 편리함을 더함 -> 협업 플랫폼의 대표가 됨.



VS



Workspace 기반	Project 기반
Developer Community Documentation	Plugin Syntax Checker
무료	유료

Why IntelliJ?

- IDEA intelligent
 - code prediction
 - syntax checker
 - Refactoring
- Built-in Developer Tools
 - Version control
 - Build tools
- 다른 Testing Tool과 연동이 쉽다.
- 안정적이고 설치하기 쉬운 plugin

06

Automatic Build:



Gradle

- JVM에서 실행되는 스크립트 언어 Groovy 기반의 DSL
 - JAVA와 호환되고 문법이 가까움
 - Xml방식의 Maven보다 쉬움
- Task based core model
 - Directed Acyclic Graphs (DAGs) of tasks (units of work)
 - Abstract and Concrete
- 종속성에 따라 DAG를 만들어 작업 실행 순서를 결정
 - 모든 Build Process를 작업 그래프로 모델링하여 유연한 빌드 가능
- Plugin 을 통한 확장성
- 빠른 속도

07 Unit Test: JUnit 5

JUnit

JUnit assert 관련 주요 method

- ✓ 단위 테스트를 위한 프레임워크
- ✓ 문자 혹은 GUI 기반으로 실행됨
- ✓ Annotation(@)으로 간결한 사용 가능
- ✓ 단정문(Assert)으로 결과 판별
- ✓ 테스트 결과 확인+최적화된 코드를 유추하는 기능도 제공

Method	내용
assertArrayEquals(a,b)	배열 a, b가 일치하는가?
assertEquals(a,b)	객체 a, b의 값이 일치하는가?
assertSame(a,b)	객체 a, b가 같은 객체인가?
assertTrue(a)	a가 참인가?
assertFalse(a)	a가 거짓인가?
assertNotNull(a)	객체 a가 NULL이 아닌가?

07 Junit – simple example

```
1 public class Calculator {  
2     public static int  
3 }  
4
```

Safe delete 'Calculator' ▶
Create Test ▶
Create subclass ▶
Make 'Calculator' package-private ▶
Add Javadoc ▶
Press `\Space` to open preview

Create Test

Testing library: JUnit4

Class name: CalculatorTest

Superclass: Calculator

Destination package:

Generate:
 setUp/@Before
 tearDown/@After

Generate test methods for: Show inherited methods

Member
 sum(a:int, b:int):int

Cancel OK

```
Tests passed: 1 of 1 test - 3ms  
> task :processTestResources NO-SOURCE  
> Task :testClasses  
> Task :test  
Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.  
Use '--warning-mode all' to show the individual deprecation warnings.  
See https://docs.gradle.org/6.1/userguide/command\_line\_interface.html#sec:command\_line\_warnings  
BUILD SUCCESSFUL in 1s  
3 actionable tasks: 2 executed, 1 up-to-date  
12:56:00 오전: Task execution finished ':test --tests "CalculatorTest"'.  
12:49 오전 Tests passed: 1  
12:49 오전 Tests passed: 1  
12:49 오전 Tests passed: 1  
12:56 오전 Tests passed: 1  
12:56 오전 Tests passed: 1  
Event Log
```

08 Slack 과 Redmine 연결

```
(base) ~$ cd redmine-4.1.1
(base) ~$ ls
CONTRIBUTING.md  app          ap          log          vendor
Gemfile          appveyor.yml doc          plugins
Gemfile.lock     bin          extra       public
README.rdoc      config      files       test
Rakefile         config.ru   lib         tmp
(base) ~$ cd plugins
(base) ~$ ls
README
(base) ~$ git clone https://github.com/sciyoshi/redmine-slack.git redmine_slack
Cloning into 'redmine_slack'...
remote: Enumerating objects: 255, done.
remote: Total 255 (delta 0), reused 0 (delta 0), pack-reused 255
Receiving objects: 100% (255/255), 92.57 KiB | 110.00 KiB/s, done.
Resolving deltas: 100% (80/80), done.
(base) ~$ source ~/.bashrc
(base) ~$ bundle install
[DEPRECATED] Your Gemfile contains multiple primary sources. Using `source` more than once without a block is a security risk, and may result in installing unexpected gems. To resolve this warning, use a block to indicate which gems should come from the secondary source. To upgrade this warning to an error, run `bundle config set disable_multisource true`.
The dependency tzinfo-data (>= 0) will be unused by any of the platforms Bundler is installing for. Bundler is installing for ruby but the dependency is only for x86-mingw32, x64-mingw32, x86-mswin32. To add those platforms to the bundle, run `bundle lock --add-platform x86-mingw32 x64-mingw32 x86-mswin32`.
The dependency ffi (>= 0) will be unused by any of the platforms Bundler is installing for. Bundler is installing for ruby but the dependency is only for x86-mingw32, x64-mingw32, x86-mswin32. To add those platforms to the bundle, run `bundle lock --add-platform x86-mingw32 x64-mingw32 x86-mswin32`.
Fetching gem metadata from https://rubygems.org/.....
Fetching gem metadata from https://rubygems.org/.
Resolving dependencies...
Using rake 13.0.1
Using concurrent-ruby 1.1.6
Using i18n 1.6.0
Using minitest 5.14.0
Using thread_safe 0.3.6
Using tzinfo 1.2.7

Bundle complete! 32 Gemfile dependencies, 60 gems now installed.
Gems in the groups development, test, rmagick and deprecation were not installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
(base) ~$ bundle exec rails server webrick -e production
```

레드마인
폴더

```
vim
~d ~/.rbenv ]| && \
export PATH=${HOME}/.rbenv/bin:${PATH} && \
eval "$(rbenv init -)"

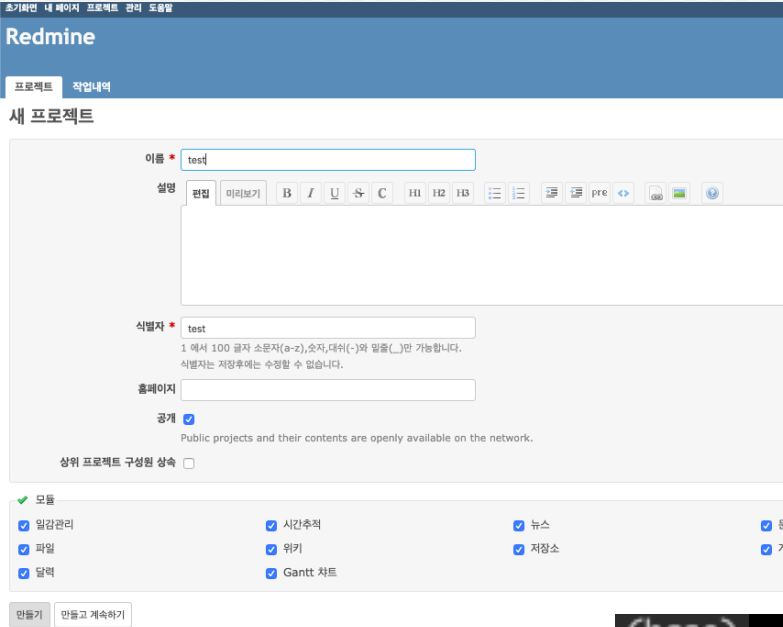
-- INSERT --
```

~./bashrc
-> GemNotFoundException 해결

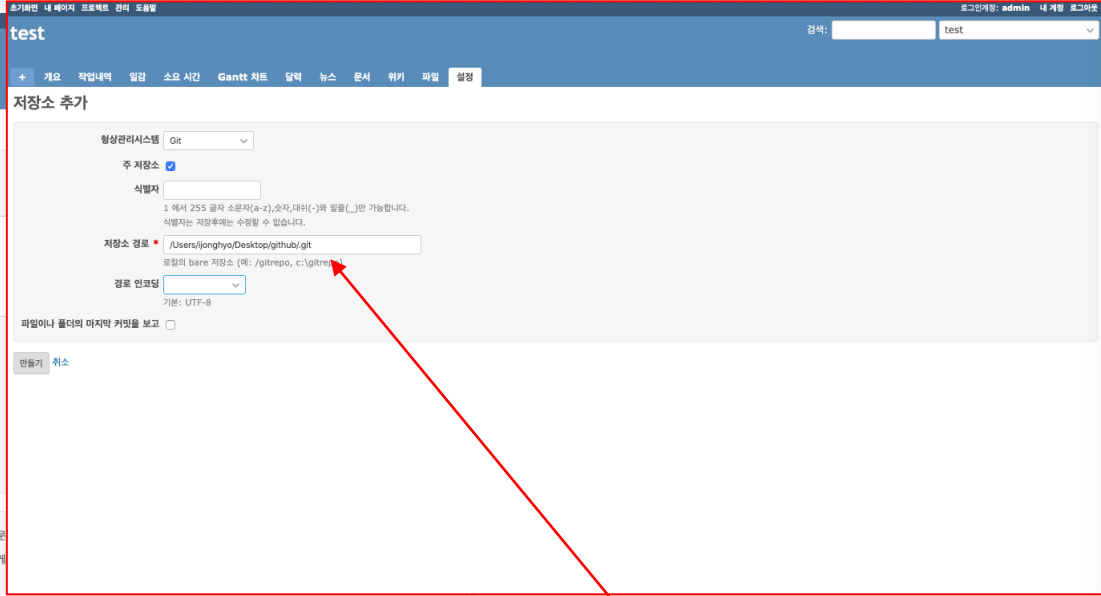
```
(base) ~$ bundle exec rails server webrick -e production
=> Booting...
=> Rails 5.2.4.2 application starting in production on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
[2020-05-06 21:38:16] INFO WEBrick 1.4.2
[2020-05-06 21:38:16] INFO ruby 2.6.6 (2020-03-31) [x86_64-darwin19]
[2020-05-06 21:38:16] INFO WEBrick::HTTPServer#start: pid=30546 port=3000
127.0.0.1 - - [06/May/2020:21:38:18 KST] "GET / HTTP/1.1" 200 5180
- -> /
```

redmine 연결

08 Redmine과 Git 연결



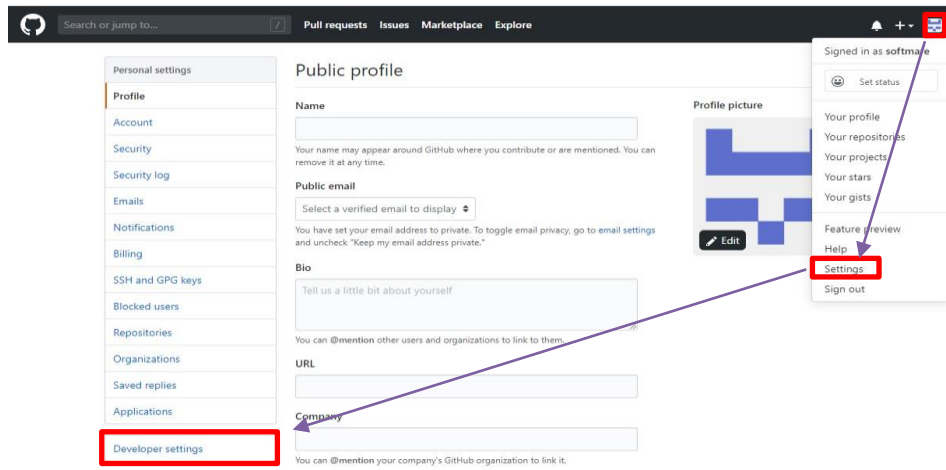
Redmine 새 프로젝트 생성 폼. 이름: test, 식별자: test, 공개: . 하위 프로젝트 구성원 상속: . 모듈 선택: 일감관리, 파일, 달력, 시간추적, 위키, Gantt 차트, 뉴스, 저장소, 문서, 개.



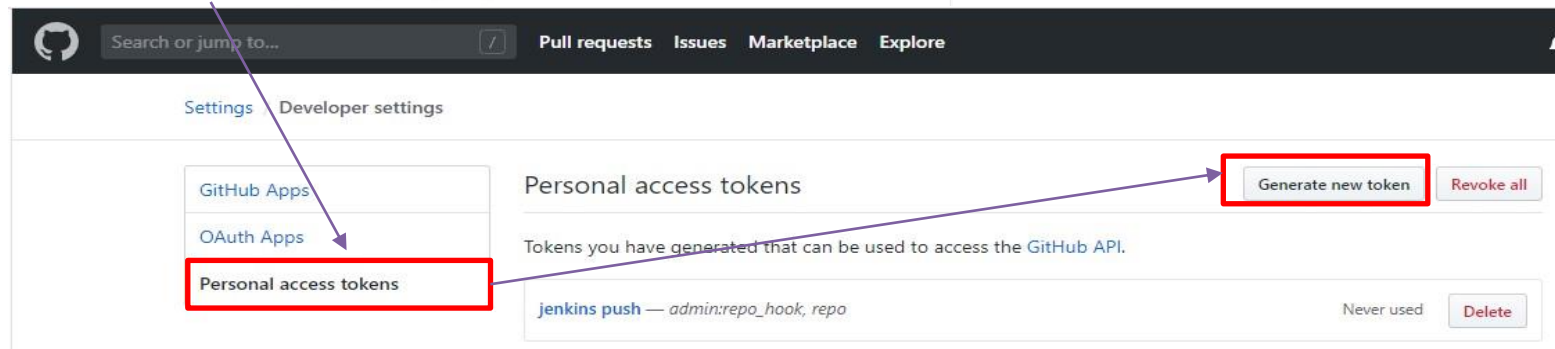
Redmine '저장소 추가' 폼. 저장소 종류: Git, 주 저장소: , 식별자: test, 저장소 경로: /Users/jonghyo/Desktop/github/git, 경로 인코딩: UTF-8. '만들기' 버튼이 활성화되어 있음.

```
(base) $ cd Desktop/github
(base) $ ls -la
.  ..  .DS_Store  .git
(base) $
```

08 Jenkins과 Github 연결



1. Push를 감지할 repository의 관리자 계정의 secret text를 생성



08 Jenkins과 Github 연결

- GitHub Apps
- OAuth Apps
- Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used to authenticate to the API over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

jenkins push alarm

What's this token for?

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo:deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> write:packages	Upload packages to github package registry
<input type="checkbox"/> read:packages	Download packages from github package registry
<input type="checkbox"/> delete:packages	Delete packages from github package registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org membership, read and write org and team membership, read org projects
<input type="checkbox"/> write:org	
<input type="checkbox"/> read:org	
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks

Pull requests Issues Marketplace Explore

Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your new personal access token now. You won't be able to see it again!

<input checked="" type="checkbox"/> 6284277605f8ef297d1bd138d4b23517949f26b1	Delete
jenkins push — admin:repo_hook, repo	Never used Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Secret text가 생성됨
-> 이 text를 복사하여 저장해 놓는다.

08 Jenkins과 Github 연결

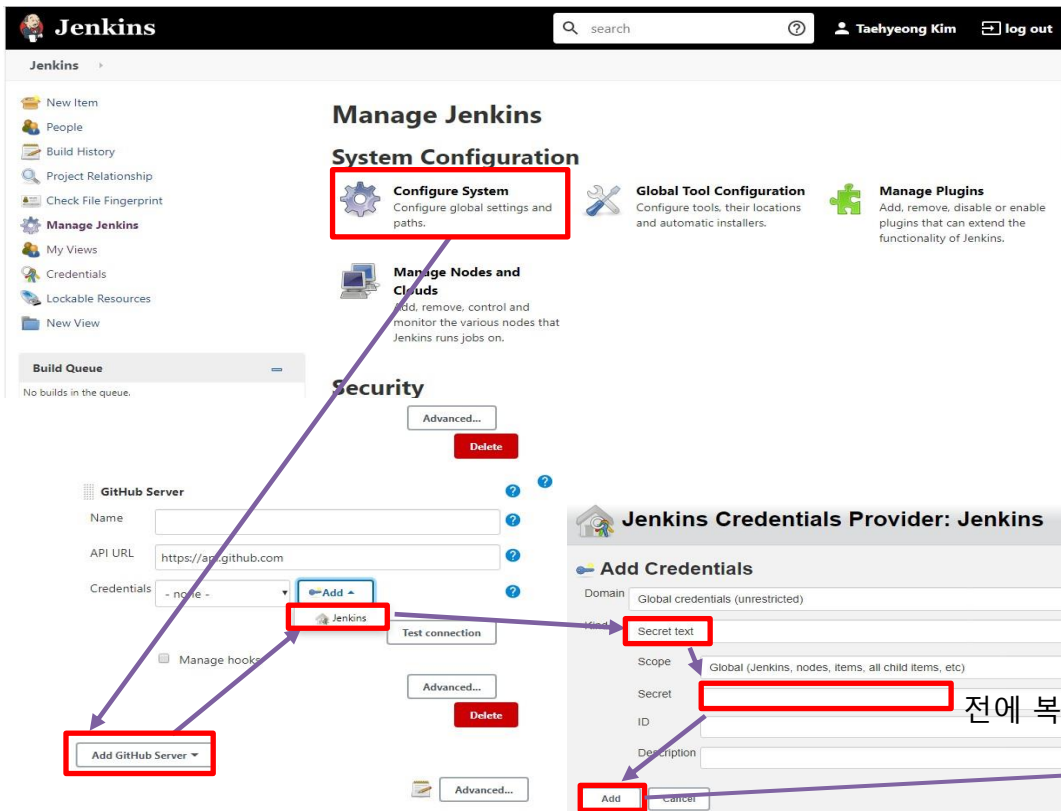
The screenshot shows the Jenkins web interface for a project named 'test'. The top navigation bar includes the Jenkins logo, a search bar, and the breadcrumb 'Jenkins > test'. A left sidebar contains navigation options: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure' (highlighted with a red box), 'GitHub Hook Log', and 'Rename'. The main content area is titled 'Project test' and includes a sub-header 'This project is for test.' Below this are icons for 'Workspace' and 'Recent Changes'. A 'Permalinks' section lists four links: 'Last build (#3), 21 hr ago', 'Last stable build (#3), 21 hr ago', 'Last successful build (#3), 21 hr ago', and 'Last completed build (#3), 21 hr ago'. At the bottom, there is a 'Build History' section with a search bar and a table of three builds (#3, #2, #1) with their respective timestamps. Below the build history are links for 'Atom feed for all' and 'Atom feed for failures'.

2. 프로젝트의 빌드 트리거에서 GitHub hook을 체크한다.

This panel shows the 'Build Triggers' configuration for the project. It contains several options, each with a checkbox and a help icon (question mark):

- Trigger builds remotely (e.g., from scripts)
- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling (highlighted with a red box)
- Poll SCM

08 Jenkins과 Github 연결



3. 젠킨스에 secret text를 등록한다.

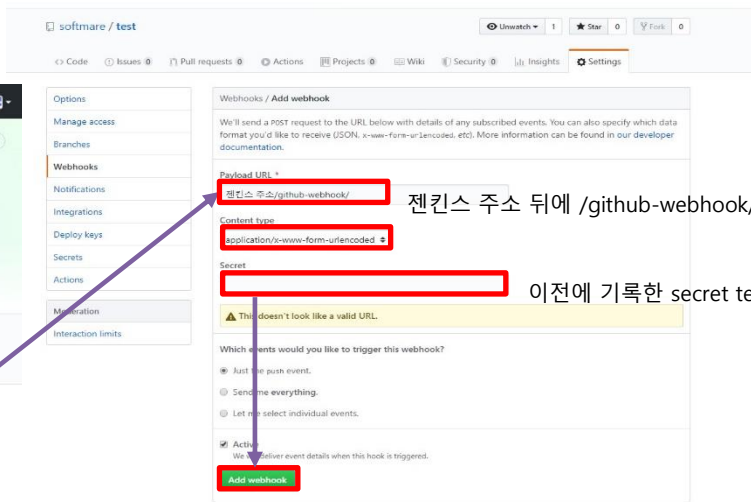
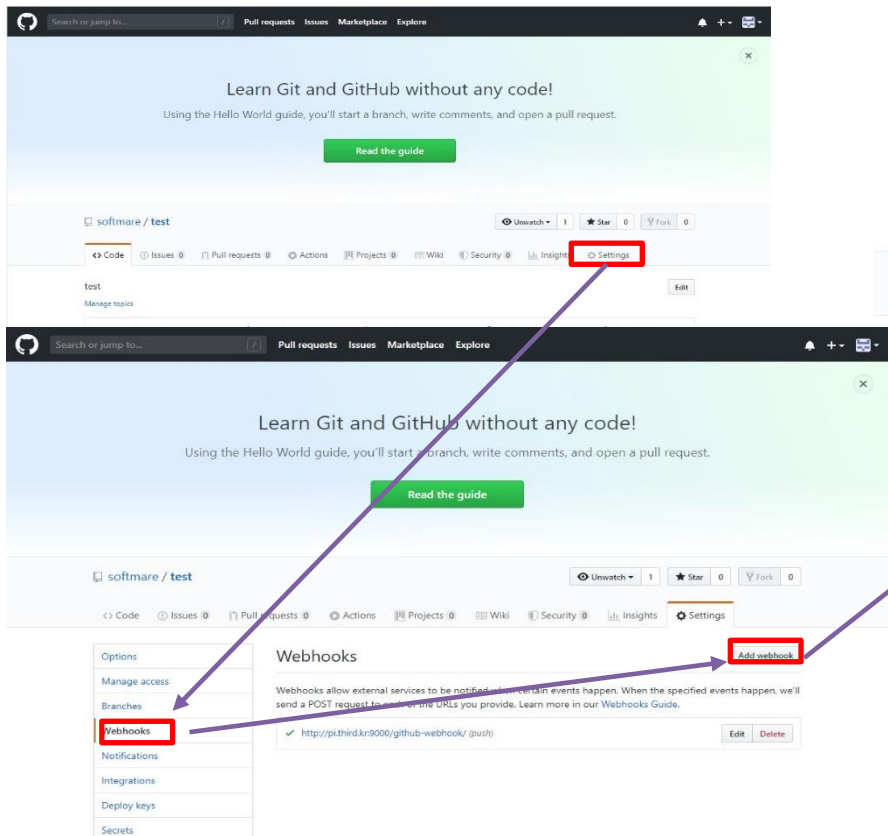


이후 설정을 저장.

전에 복사하였던 secret text를 붙여넣기 한다.

08 Jenkins과 Github 연결

4. Repo의 설정화면에서 jenkins의 주소 입력

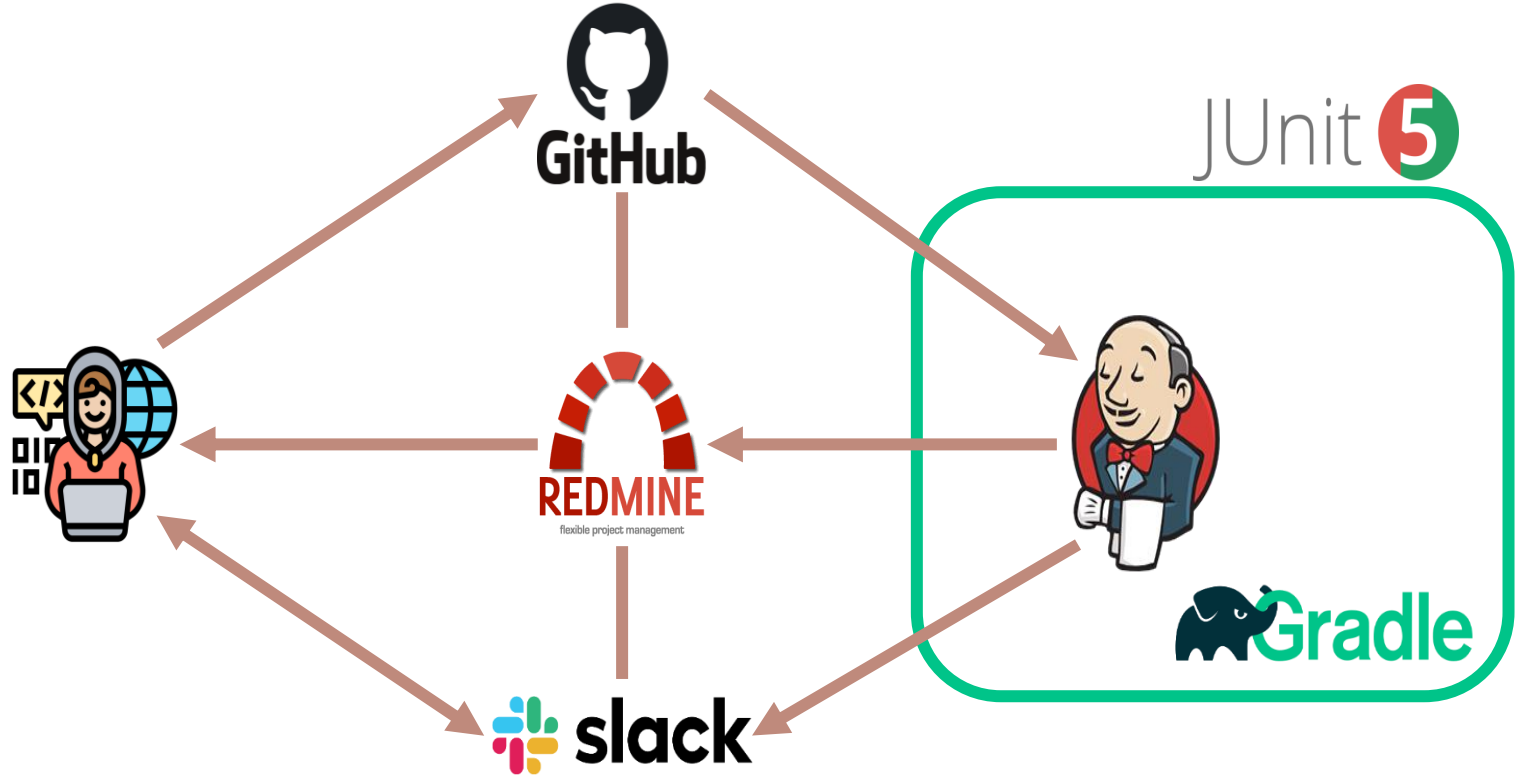


제닉스 주소 뒤에 /github-webhook/ 을 붙인다.

이전에 기록한 secret text 붙여넣기.

이 과정을 마치면 해당 repo에 push가 이루어질 경우 jenkins에서는 자동으로 build를 수행한다.

09 총정리





감사합니다!