Software V&V

# CTIP 환경 구축


TEAM 2

2012125191 김선우

2015106241 김용현

2016112611 민지호

2016112931 전다윤

# CONTENTS

# Intelli J vs Eclipse


vs


장점 : 무료로 자유롭게 사용할 수 있다
단점 : 자바로 만들어져 JVM 위에서 실행되므로 느리고, 무겁다

장점 : 우수한 스마트 완성 기능으로 생산성이 높다
단점 : Ultimate 버전은 유료 라이선스가 필요하다

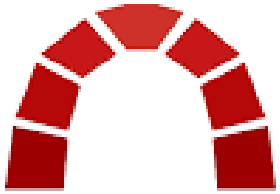
# “ REDMINE ”

**화면 기반의 프로젝트 관리에 도움이 되도록 달력과 간트 차트를 제공**

**프로젝트** : 프로젝트를 구분하는 단위이며,
프로젝트 내에 일감, 뉴스, 저장소 등의 모듈 사용 가능

**일감** : 업무의 단위이며, 업무 지시 및 조회 그리고 관리가 가능

**간트 차트** : 일감 진행 상황을 시간의 흐름에 따라 보여주는 기능



REDMINE
flexible project management

# “ REDMINE ”

**Bug Tracking 툴로 REDMINE을 선택한 이유?**

유사 프로그램으로 MENTIS가 있으나, 버그 트래킹의 기능만 제공하고 다른 툴과의 연동이 어렵다는 한계점이 있다.

따라서, 무료이면서 다양한 기능을 제공하는 REDMINE을 선택

# WHY SLACK?

**1) 다양한 플러그인 제공**

플러그인을 통해 기존 앱들과 연동 가능 ex) Github, Jenkins

**2) 원하는 내용만 push 알림 받을 수 있음**

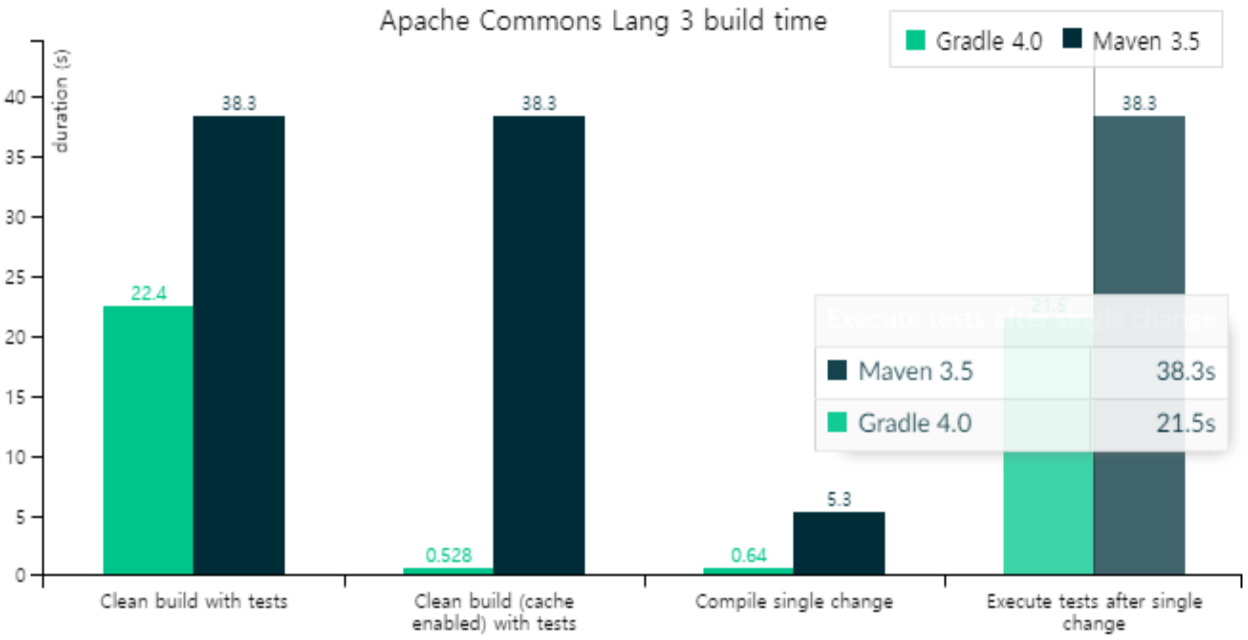
해당 채널의 모든 내용, 아이디 언급 시, 사전에 지정해둔 단어 등으로 분류해서 알

알림 가능

**3) 보안**



# Gradle VS Maven




**Gradle이 Maven보다 좋은점**

1. Gradle은 Groovy를 사용하기 때문에, 동적인 빌드는 플러그인을 호출하거나 직접 코드를 짜면 된다.
2. Build라는 동적인 요소를 XML로 정의하기에는 어려운 부분이 많다.
3. 속도가 빠르다.

## CI(Continuous Integration)?

팀 구성원들이 작업한 내용을 정기적으로 통합하는 것

CI를 성공적으로 구현할 경우 애플리케이션에 대한 코드 변경 사항이 정기적으로 빌드 및 테스트되어 공유 repository에 통합된다.

ex) 형상 관리 된 commit된 소스코드들을 정기적으로  통합시켜준다.

Jenkins

Travis CI

Jenkins

## Jenkins 사용이유

1. CI를 지원해주는 도구

2. Plugin을 통해 다른 도구와의 호환성이 높다.

ex) RedMine : 요구사항 분석 ,GitHub : 형상 관리, test 결과 및 스케쥴 : Slack

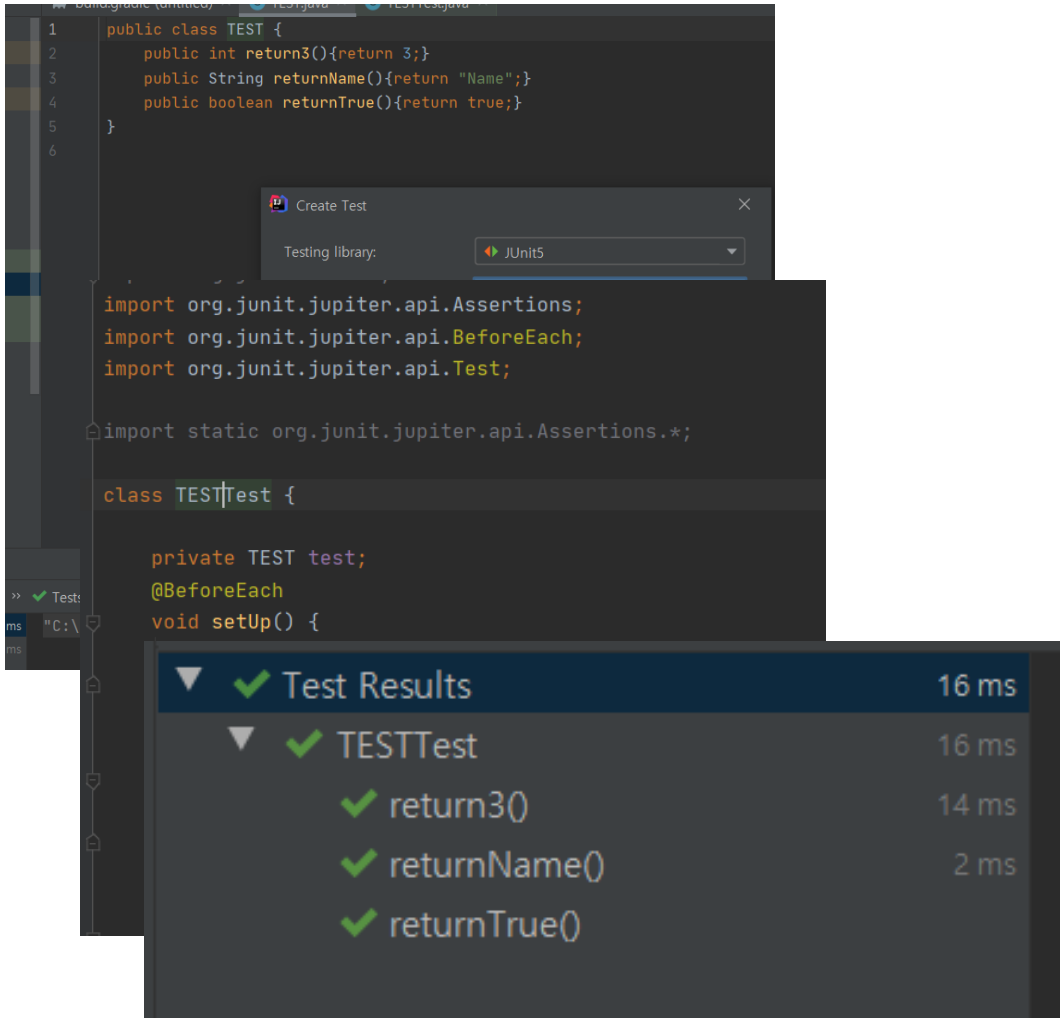
3. 자동화 테스트 : Git과의 연동을 통해 코드변경시 자동 테스트 진행

## Unit Testing?

Unit Testing 은 프로그래밍에서 코드의 특정 모듈이 의도된 대로 정확히 작동하는지 검증하는 절차로써 함수와 메소드에 대한 테스트 케이스를 작성하는 절차이다. 이 과정을 통해 코드 변경으로 인해 문제가 발생할 경우, 단시간 내에 이를 파악하고 바로 잡을 수 있도록 해준다


**Junit**

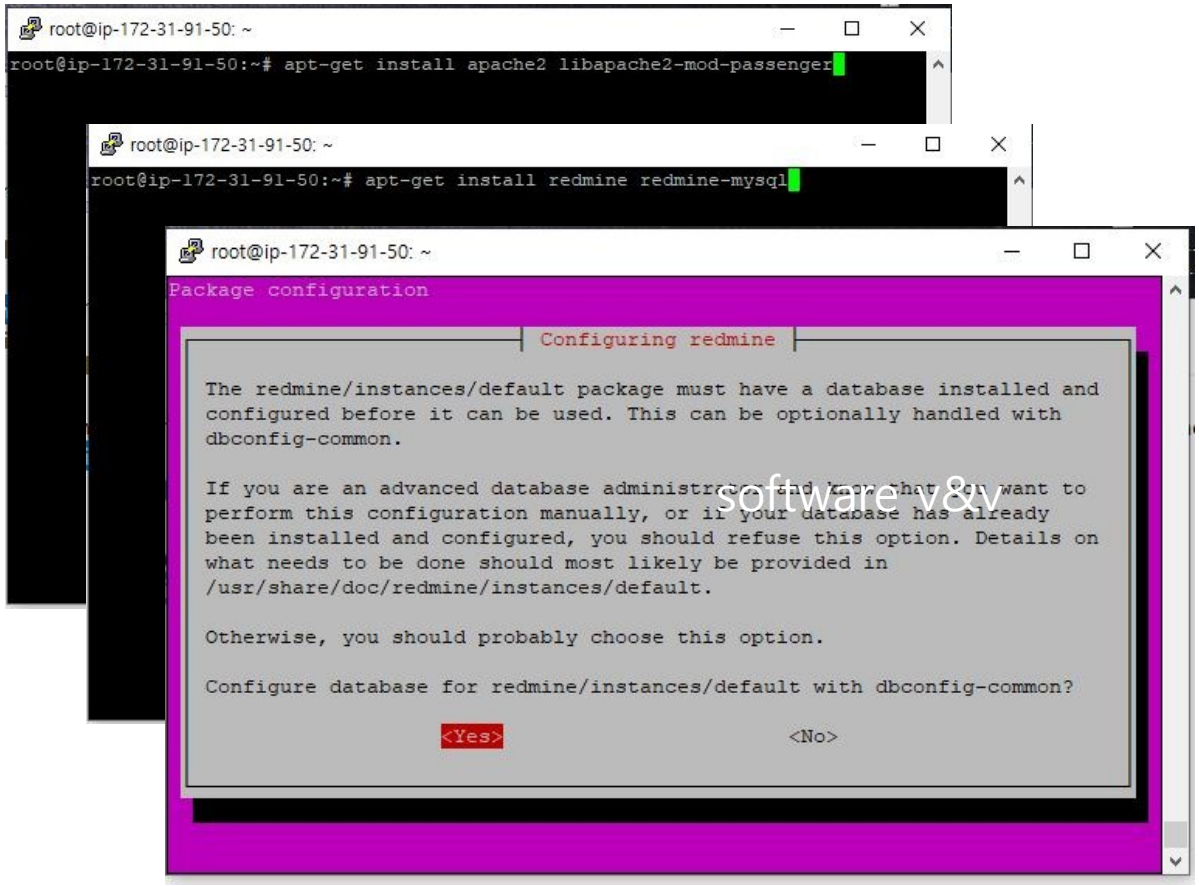
1. 자바의 단위 테스트를 위한 프레임워크
2. 테스트의 수행결과 표시 및 P or F 표기
3. Annotation 지원




1. Junit 에 필요한 lib설치 (Fix)
2. TestCode 작성
3. Test

# RedMine 설치 (1)

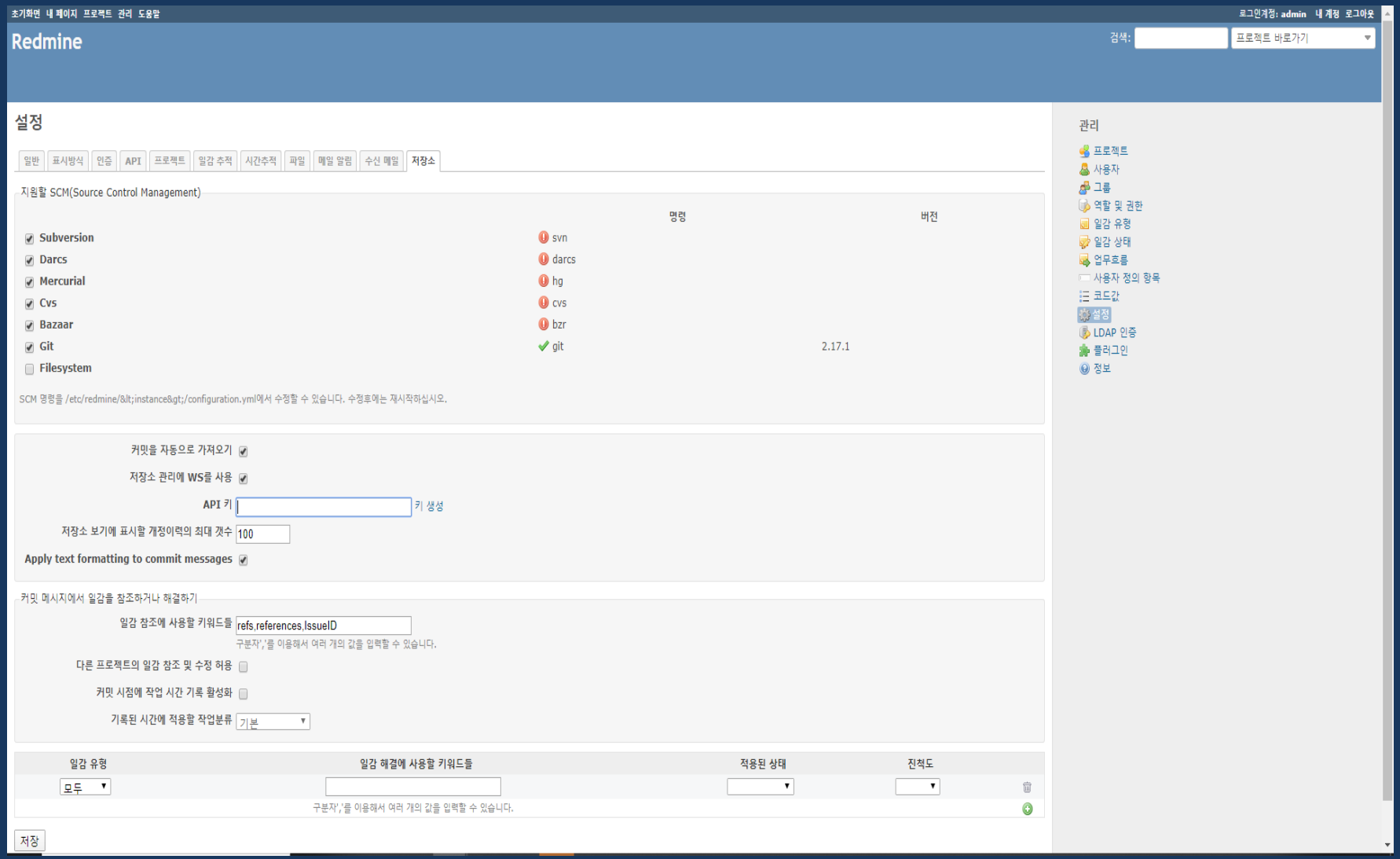





1. 레드마인 설치시 필요한 패키지 설치
   ->(Apache2, Mysql, Ruby, Gem)
2. 레드마인 설치
3. 개인 설정

# RedMine GitHub 연동(1)

s o f t w a r e  v & v




RedMine 관리 -> 설정 -> 저장소
Git 지정 및 커밋을 자동으로 가져오기, 저장소 관리에 WS 사용 체크

# RedMine GitHub 연동(2)

1. Git remote 저장소를 Redmine이 설치된 서버에 클론하기

```
ubuntu@ip-172-31-17-171:/var/lib/jenkins$ sudo git clone --mirror https://github
.com/minjyo/softwareV_V.git
Cloning into bare repository 'softwareV_V.git'...
remote: Enumerating objects: 123, done.
remote: Counting objects: 100% (123/123), done.
remote: Compressing objects: 100% (84/84), done.
remote: Total 123 (delta 33), reused 123 (delta 33), pack-reused 0
Receiving objects: 100% (123/123), 5.03 MiB | 28.61 MiB/s, done.
Resolving deltas: 100% (33/33), done.
```

2. 로컬 저장소의 주소를 Redmine 저장소 경로에 넣기

3. Git remote를 업데이트하는 sh 파일 생성

..estfolder/src (zsh) ⌘1 ubuntu@ip-172-31-1... ⌘2

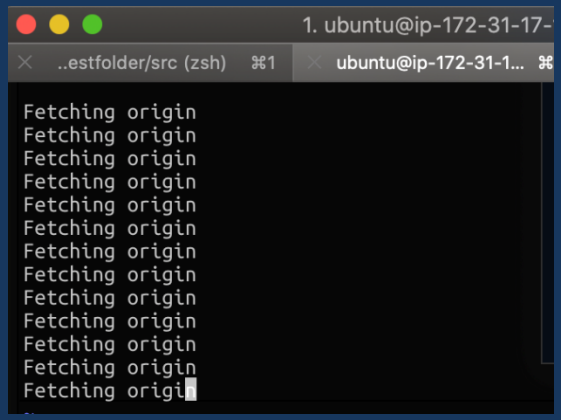
```
#!/bin/bash
cd /home/ubuntu/softwareV_V.git
git remote update
```

4. sh 파일을 주기적으로 실행시켜주도록 설정

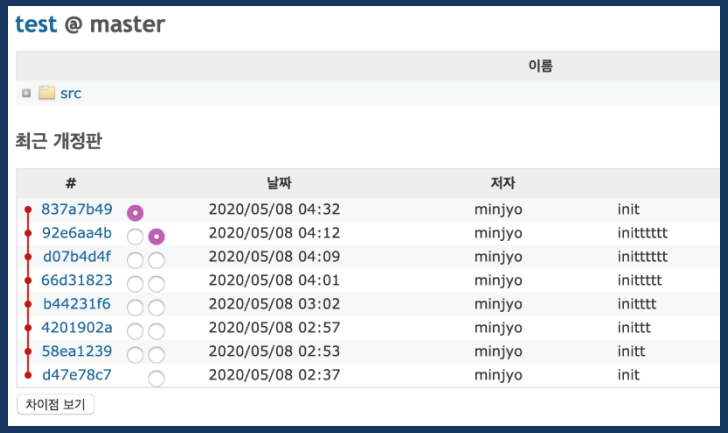
```
* * * * * /home/ubuntu/update.sh >> /home/ubuntu/cron.log 2>&1
# crontab script end
```

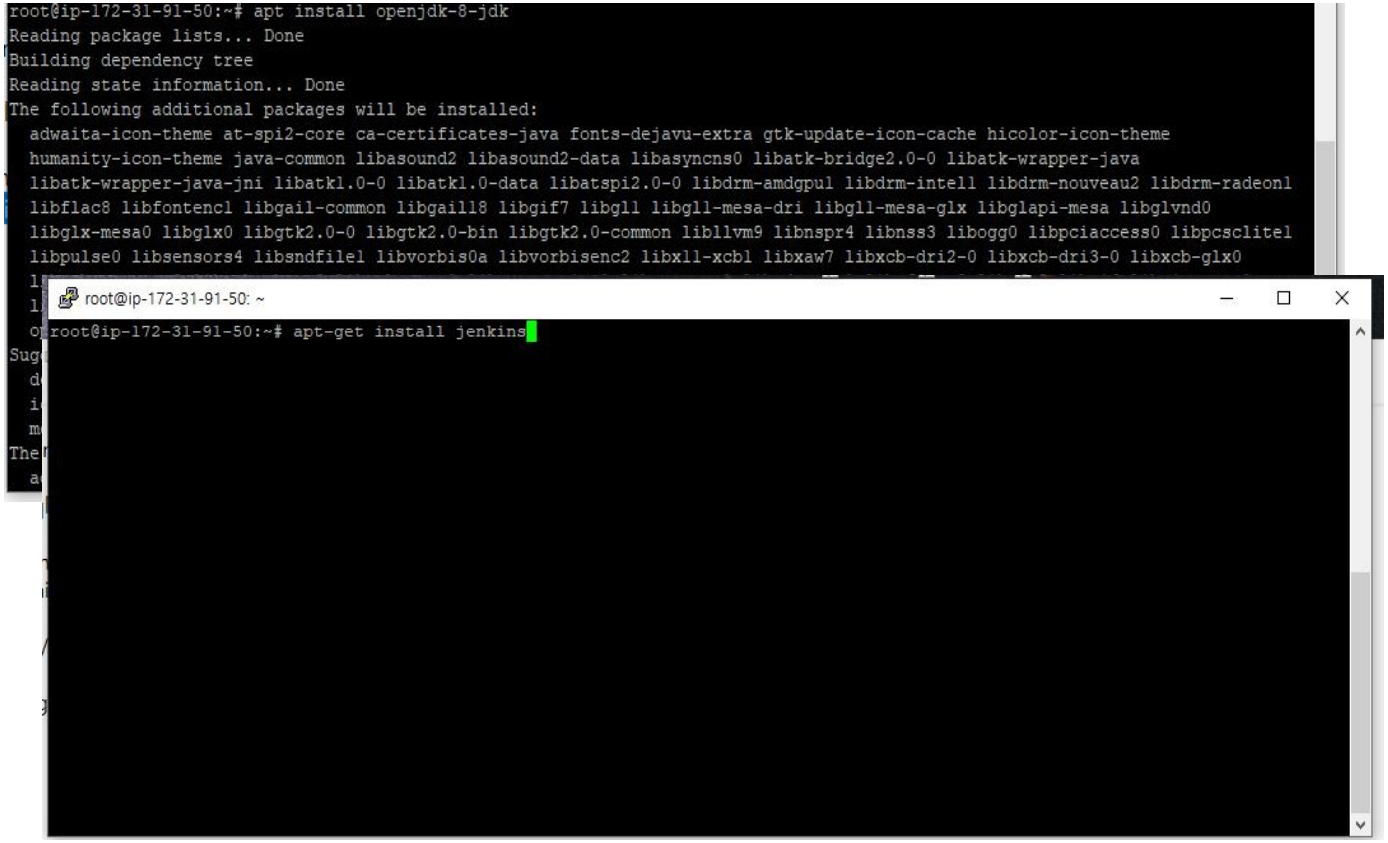
# RedMine GitHub 연동(3)

## 5. 생성된 로그 파일 확인




## 6. Redmine에서 업데이트 된 내용 확인

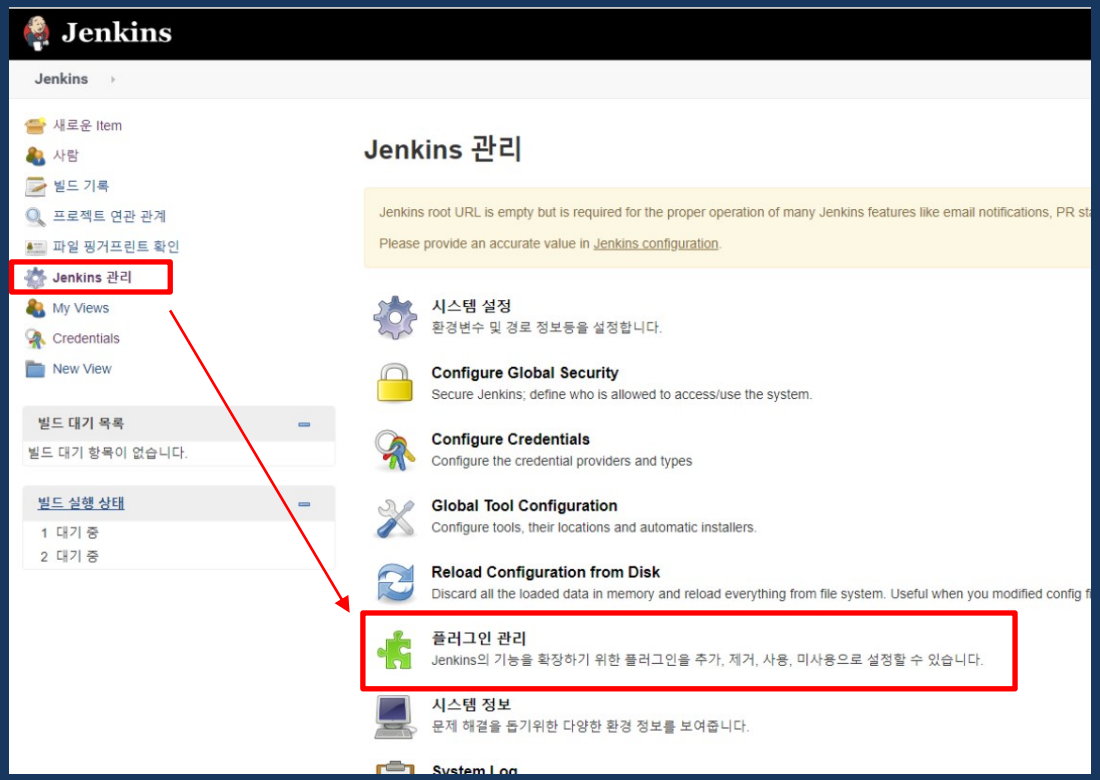





1. 자바 JDK 다운
2. Jenkins 설치

# JenKins Github 연동

```
root@ip-172-31-37-209:~# ssh-keygen -t rsa -f /var/lib/jenkins/.ssh/softwarevv
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/softwarevv.
Your public key has been saved in /var/lib/jenkins/.ssh/softwarevv.pub.
The key fingerprint is:
SHA256:/DGe31mXpPAV4aH83iPqZhljuBA/iqE1YIkkm2YphuA root@ip-172-31-37-209
The key's randomart image is:
+---[RSA 2048]----+
|                 o |
|o.           . o o|
|*oo .         o o |
|=E +   o       . .|
|= . .   S +.    + |
|     + . * Bo = o|
|    o + o B ++ ++|
|   . . . . =...oo|
|          +o. o  |
+----[SHA256]-----+
root@ip-172-31-37-209:~# cd /var/lib/jenkins/.ssh
root@ip-172-31-37-209:/var/lib/jenkins/.ssh# ls -al
total 16
drwxr-xr-x  2 root    root    4096 May  3 06:36 .
drwxr-xr-x 17 jenkins jenkins 4096 May  3 05:54 ..
-rw-------  1 root    root    1675 May  3 06:36 softwarevv
-rw-r--r--  1 root    root     403 May  3 06:36 softwarevv.pub
```




1. CI serve에서 key generation

2. GitHub, Github integration 플러그인 설치

3. Credentials 탭에서 비밀키 등록

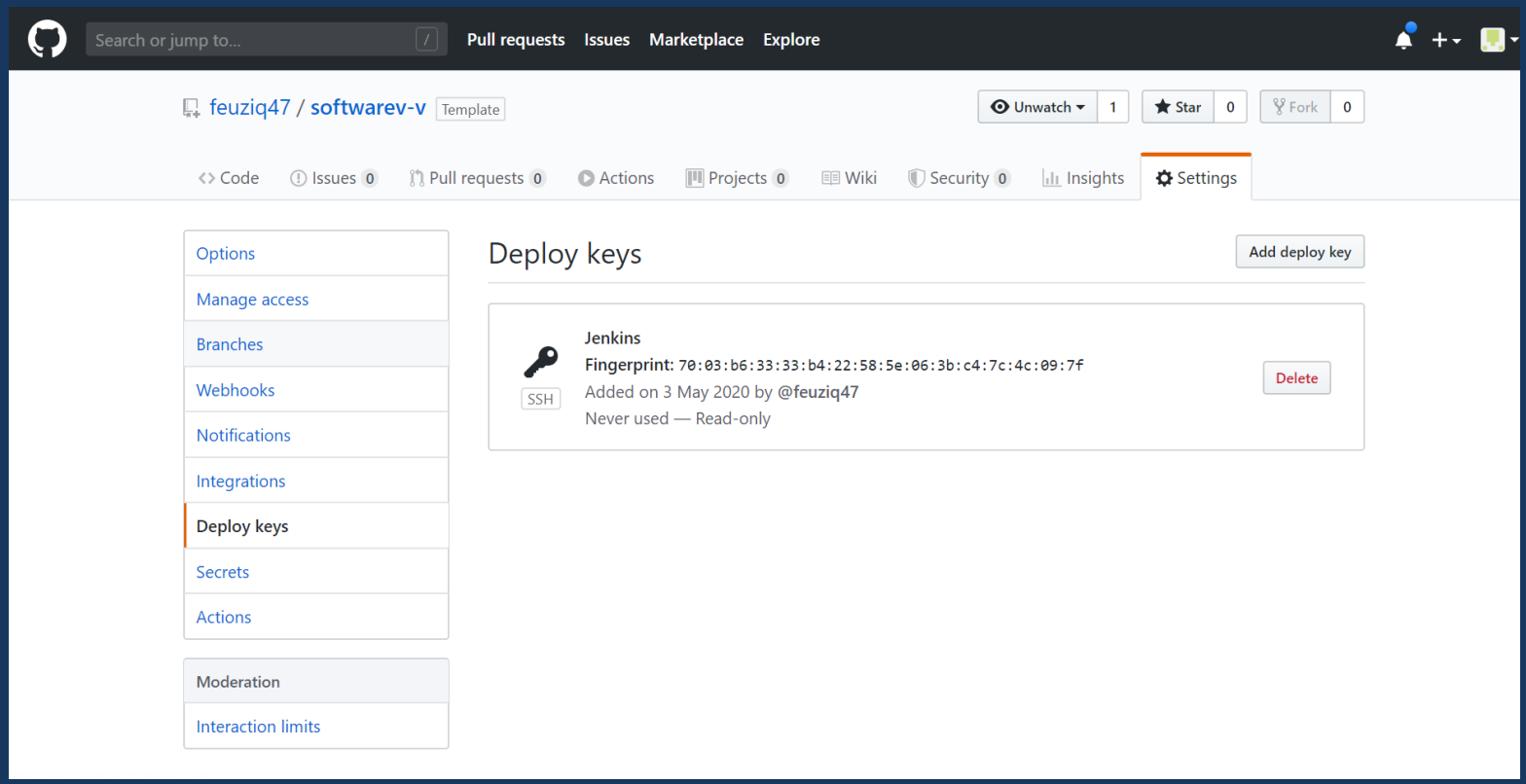
4. Github에서 Jenkins와 연동시킬 repository를 생성한다.

4. 새로운 item 만들어 소스코드 관리에서 git repo URL을 작성하고, 만든 credential을 적용한다.

5. 빌드 유발에서는 Github hook trigger for GITScm polling을 선택한다.

# JenKins Github 연동

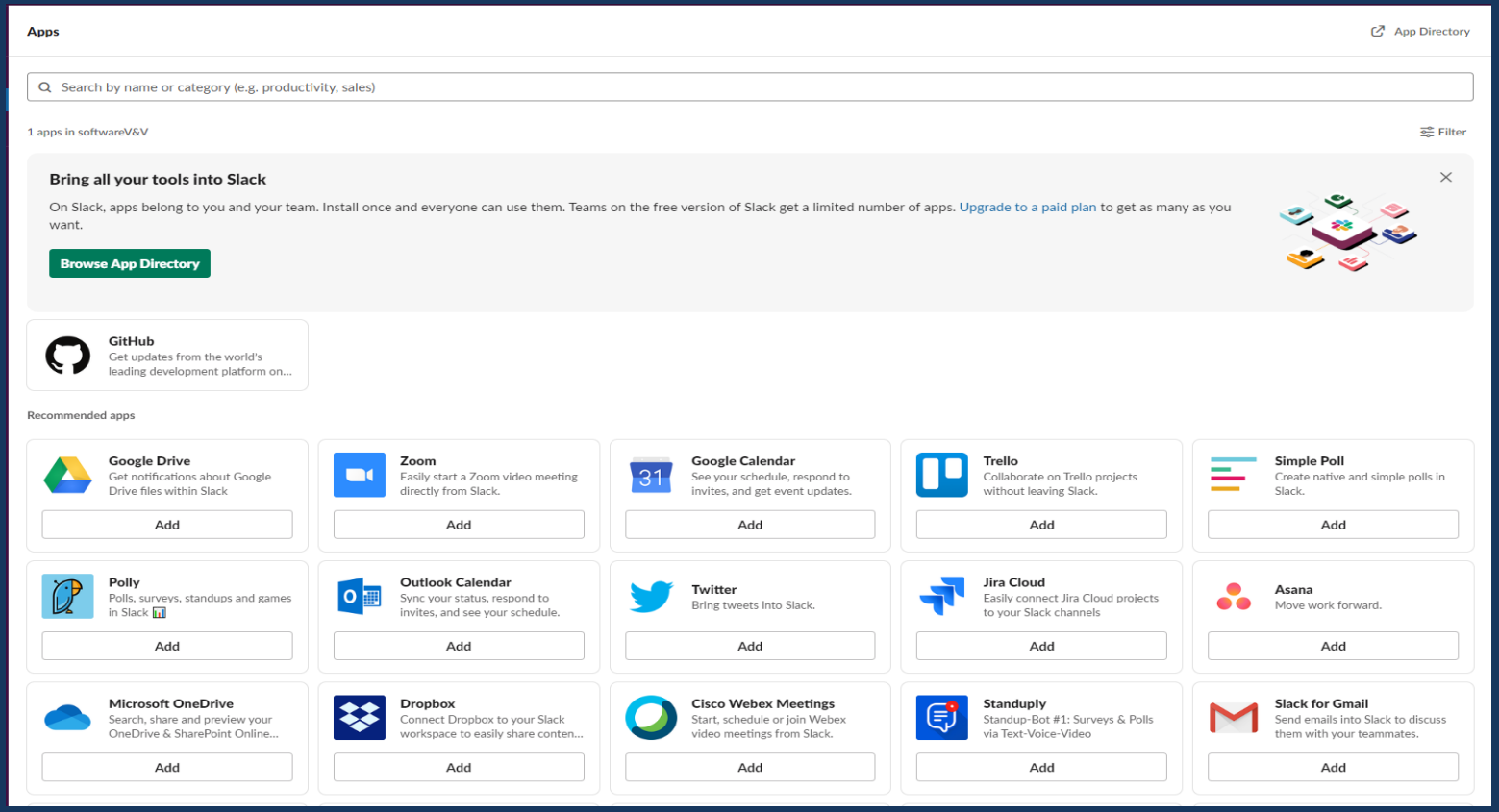





1. settings -> deploy keys -> add deploy key를 선택하여 생성한 .pub 파일 붙여넣는다.
2. settings -> Webhooks -> add webhook을 선택하여 payloadURL에 Jenkins IP/github-webhook을 입력하여 Active에 체크한다.
3. 이후 git repository에 푸시하면 젠킨스에서 push된 것을 탐지하여 자동으로 빌드를 수행한다.

# Slack Jenkins 연동

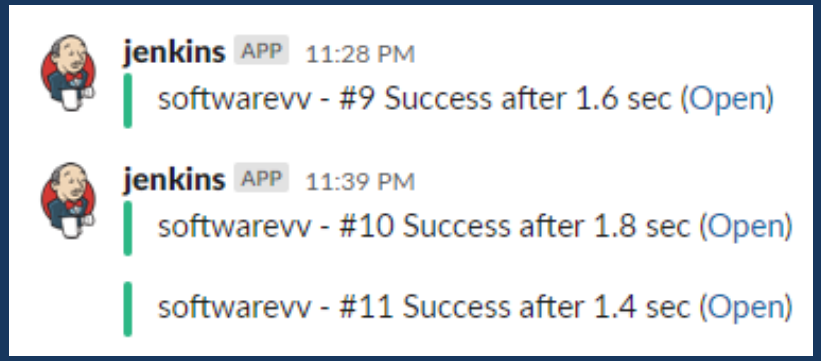





1. Slack의 App에서 Jenkins CI 추가
2. team token을 저장해 놓는다. (Jenkins에서 slack 설정 시 secret text 부분에 입력)

# Slack Jenkins 연동

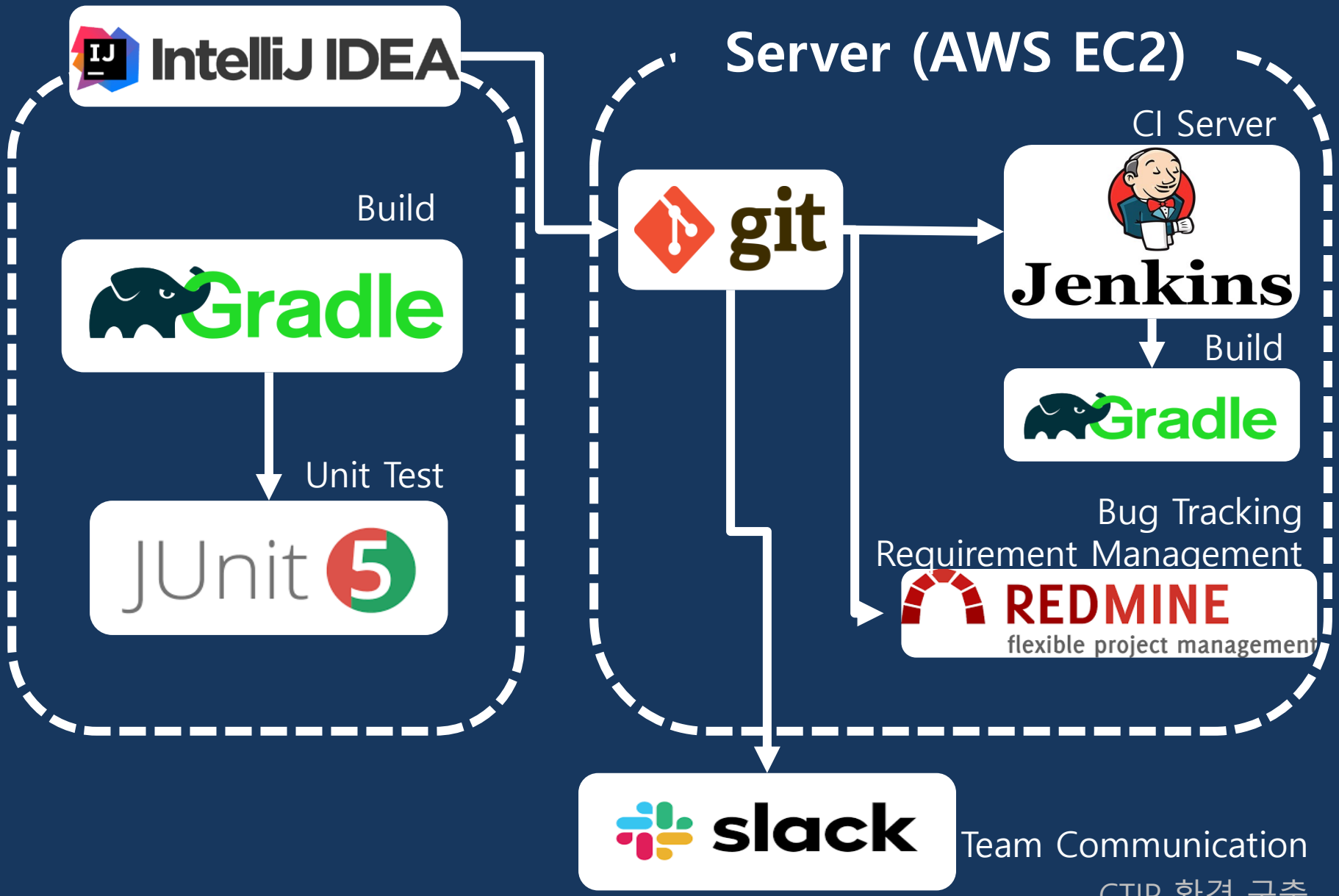



1. Jenkins 플러그인 관리에서 slack notification 추가
2. Jenkins 시스템 설정에서 slack 탭에 workspace, channel 정보를 입력한다.
3. GitHub에 푸시이후, 해당 채널에서 빌드 결과 확인

# Overall structure






CTIP 환경 구축

과 목 이 름 이 나 영 문 타 이 틀

# THANK YOU

발 표 자