

Software V&V

- CTIP 환경구축 -

1팀

컴퓨터공학부 201711381 김소현
컴퓨터공학부 201711394 민하은
컴퓨터공학부 201711401 염혜지
컴퓨터공학부 201711407 이가현
컴퓨터공학부 201711435 홍예주

목차

1

CTIP 환경 구성도

- 1) IDE - IntelliJ
- 2) Build - Gradle
- 3) Unit Test - Junit
- 4) 형상관리 - Git & GitHub
- 5) Server - AWS
- 6) Bug Tracking - Redmine
- 7) CI Tool - Jenkins
- 8) Communication Tool - Slack

2

AWS 서버 구축

3

Jenkins & Gradle, Git 연동

4

Redmine 설치 & Git 연동

5

Jenkins & Redmine 연동 with Zapier

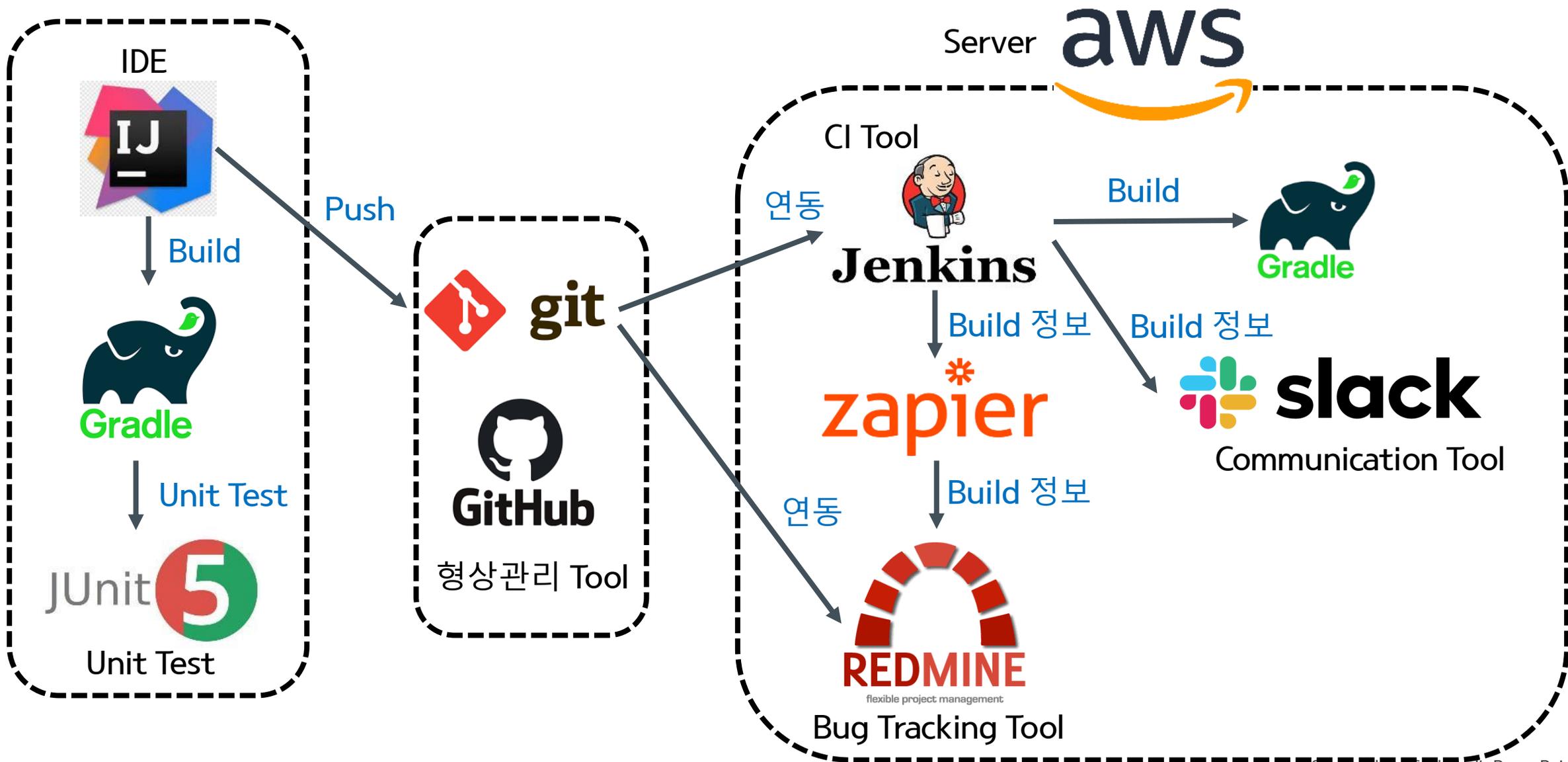
6

Slack & Jenkins 연동

PART 1

CTIP 환경 구성도

CTP 환경 구성도





* AWS를 선택한 이유

- 1) 1시간 단위로 사용한 만큼만 지불하면 되므로 비용이 적게 들어간다.
- 2) 서버의 생성과 폐기를 쉽게 할 수 있다.
- 3) 생성할 서버의 규모를 자유롭게 조절할 수 있다.
- 4) 구글에 AWS에 대한 많은 양의 자료가 있어서 사용하는데 진 입장벽이 낮다.

* AWS 선택 시 감안해야 하는 점

- 1) 프리티어 사용시 요금 발생(요금 폭탄)을 주의해야 한다.



Jenkins

* Jenkins를 선택한 이유

- 1) 무료로 사용할 수 있다.
- 2) 다양한 플러그인 지원한다.
- 3) Git Hub 연동이 편리하다.
- 4) 많은 사용자들을 보유하고 있고, 관련 문서가 다양하다.

* Jenkins 선택 시 감안해야 하는 점

- 1) redmine 등 Issue tracking 과 연계가 불편하거나 완벽하지 않다.
=> Zapier를 이용해 Redmine과 Jenkins 연동 문제 해결

Bug Tracking Tool - Why Redmine?



* Redmine을 선택한 이유

- 1) 무료로 사용할 수 있는 오픈소스 프로그램이다.
- 2) 프로젝트 관리와 버그 추적 기능, 일정관리 기능을 모두 제공한다.
- 3) 팀원들에게 작업을 분리해서 할당할 수 있고, 작업기한을 정할 수 있다.
- 4) Git Hub 연동이 편리하다.

* Redmine 선택 시 감안해야 하는 점

- 1) Jenkins와 연계가 불편하거나 완벽하지 않다.
=> Zapier를 이용해 Redmine과 Jenkins 연동 문제 해결



* Slack을 선택한 이유

- 1) 무료로 사용할 수 있다.
- 2) Jenkins와 Redmine 서비스의 알림을 받을 수 있다.
- 3) 파일을 쉽게 공유하고, 그 파일에 대한 별도의 커뮤니케이션이 가능하다.
- 4) 대화가 편리하다.

* Slack 선택 시 감안해야 하는 점

- 1) 크게 감안할 점이 없다.

PART 2
AWS 서버 구축

1. EC2 서비스 선택

The screenshot shows the AWS console's '솔루션 구축' (Solution Builder) page. The 'EC2를 사용하여 가장 먼저 시작' (Get started with EC2) option is highlighted with a yellow box. Other options include Elastic Beanstalk, Lightsail, Route 53, AWS IoT, and CloudEndure. The page also features an 'AWS 탐색' (AWS Explorer) section on the right with various service cards like Amazon Redshift, AWS Fargate, Amazon S3, and AWS Marketplace.

2. Amazon Machine Image 선택

The screenshot shows the AWS console's '단계 1: Amazon Machine Image(AMI) 선택' (Step 1: Amazon Machine Image(AMI) Selection) page. The 'Amazon Linux 2 AMI (HVM, SSD Volume Type)' is highlighted with a yellow box. Other AMIs like Ubuntu Server and SUSE Linux Enterprise are also visible. The page includes a search bar and a list of AMIs with their respective details and '선택' (Select) buttons.

3. 인스턴스 유형 선택 & 인스턴스 세부정보 알맞게 구성

1. AMI 선택 2. 인스턴스 유형 선택 3. 인스턴스 구성 4. 스토리지 추가 5. 태그 추가 6. 보안 그룹 구성 7. 검토

단계 2: 인스턴스 유형 선택

Amazon EC2는 각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스턴스는 애플리케이션을 실행할 수 있는 가상 서버입니다. 이러한 인스턴스에는 CPU, 메모리, 스토리지 및 네트워킹 용량의 다양한 조합이 있으며, 애플리케이션에 사용할 적절한 리소스 조합을 유연하게 선택할 수 있습니다. 인스턴스 유형과 이 인스턴스 유형이 컴퓨팅 요건을 충족하는 방식에 대해 자세히 알아보기

필터링 기준: 모든 인스턴스 유형 현재 세대 열 표시/숨기기

현재 선택된 항목: t2.micro (Variable ECU, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB 메모리, EBS 전용)

	그룹	유형	vCPUs	메모리 (GiB)	인스턴스 스토리지 (GB)	EBS 최적화 사용 가능	네트워:
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS 전용	-	낮음
<input checked="" type="checkbox"/>	General purpose	t2.micro 프리 티어 사용 가능	1	1	EBS 전용	-	낮음
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS 전용	-	낮음
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS 전용	-	낮음
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS 전용	-	낮음

취소 이전 검토 및 시작 다음: 인스턴스 세부 정보 구성

4. 인스턴스 생성 후 연결

인스턴스에 연결 정보
다음 옵션 중 하나를 사용하여 인스턴스 i-00970f8947ef5062a에 연결

EC2 인스턴스 연결 | Session Manager | SSH 클라이언트

인스턴스 ID
i-00970f8947ef5062a

퍼블릭 IP 주소
3.34.98.246

사용자 이름
ec2-user

사용자 지정 사용자 이름을 사용하여 연결하거나 인스턴스 시작에 사용한 AMI의 기본 사용자 이름 ec2-user(를) 사용합니다.

참고: 대부분의 경우 추정된 사용자 이름은 정확합니다. 하지만 AMI 사용 지침을 읽고 AMI 소유자가 기본 AMI 사용자 이름을 변경했는지 확인하십시오.

취소 **연결**

```
Last login: Tue Mar 30 11:25:25 2021 from ec2-13-209-1-57.ap-northeast-2.compute.amazonaws.com
```

```

_ | _ | )
_ | ( _ /  Amazon Linux 2 AMI
_ | \ _ | _ |

```

```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-43-21 ~]$
```

AWS 서버 구축 시 주의할 점

1. 스토리지 설정

1) 크기 8GB보다 크게 할 것

: 추후 Jenkins의 JDK 등을 설치할 때 스토리지 크기가 부족한 문제점이 생겼다.

=> 스토리지를 20GB로 늘려줬다.

```
[root@ip-172-31-41-160 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        482M   0  482M   0% /dev
tmpfs           492M   0  492M   0% /dev/shm
tmpfs           492M  412K  492M   1% /run
tmpfs           492M   0  492M   0% /sys/fs/cgroup
/dev/xvda1      20G   8.0G   13G  40% /
tmpfs           99M   0   99M   0% /run/user/0
```

2) 종료 시 삭제 옵션 주의

: 해당 옵션이 체크시 종료하면 컴퓨터와 저장장치가 모두 폐기된다.

=> 인스턴스를 종료하지 않도록 주의한다.

단계 4: 스토리지 추가

인스턴스가 다음 스토리지 디바이스 설정으로 시작됩니다. 추가 EBS 볼륨 및 인스턴스 스토어 볼륨을 인스턴스에 연결하거나 루트 볼륨의 설정을 편집할 수 있습니다. 인스턴스를 시작한 후 추가 EBS 볼륨을 연결할 수도 있지만, 인스턴스 스토어 볼륨은 연결할 수 없습니다. Amazon EC2의 스토리지 옵션에 대해 자세히 알아보십시오.

블륨 유형 | 디바이스 | 스냅샷 | 크기(GiB) | 볼륨 유형 | IOPS | 처리량(MB/초) | 종료 시 삭제 | 암호화

루트 | /dev/xvda | snap-0c54158c3fe61c7b | 8 | 범용 SSD(gp2) | 100/3000 | 해당 사항 없음 | | 암호화되지 않음

새 볼륨 추가

프리 티어 사용 가능 고객은 최대 30GB의 EBS 볼륨(SSD) 또는 마그네틱 스토리지를 사용할 수 있습니다. 프리 티어 자격 및 사용량 제한에 대해 자세히 알아보기.

튜토리얼

2. 보안 그룹 설정 (= 방화벽 접근권한 설정)

- 8080, 80 포트 해제

: Jenkins 사용을 위한 8080 포트와 Redmine을 위한 80 포트를 꼭 허용해줘야 한다.

포트 범위	프로토콜	원본	보안 그룹
80	TCP	0.0.0.0/0	launch-wizard-1
80	TCP	::/0	launch-wizard-1
8080	TCP	0.0.0.0/0	launch-wizard-1
8080	TCP	::/0	launch-wizard-1
22	TCP	0.0.0.0/0	launch-wizard-1

단계 6: 보안 그룹 구성

보안 그룹은 인스턴스에 대한 트래픽을 제어하는 방화벽 규칙 세트입니다. 이 페이지에서는 특정 트래픽을 인스턴스에 도달하도록 허용할 규칙을 추가할 수 있습니다. 예를 들면 웹 서버를 설정하여 인터넷 트래픽을 인스턴스에 도달하도록 허용하려는 경우 HTTP 및 HTTPS 트래픽에 대한 무제한 액세스를 허용하는 규칙을 추가합니다. 새 보안 그룹을 생성하거나 아래에 나와 있는 기존 보안 그룹 중에서 선택할 수 있습니다. Amazon EC2 보안 그룹에 대해 자세히 알아보기.

보안 그룹 열람: 새 보안 그룹 생성
 기존 보안 그룹 선택

보안 그룹 이름:

설명:

유형	프로토콜	포트 범위	소스	설명
SSH	TCP	22	사용자 지정 0.0.0.0/0	예: SSH for Admin Desktop

[규칙 추가](#)

경고
 소스가 0.0.0.0인 규칙은 모든 IP 주소에서 인스턴스에 액세스하도록 허용합니다. 알려진 IP 주소의 액세스만 허용하도록 보안 그룹을 설정하는 것이 좋습니다.

PART 3

Jenkins & Gradle, Git 연동

1. AWS 서버에 Java / Git / Jenkins 설치

1) Java 설치 command

```
: sudo yum install java-1.8.0-openjdk-devel.x86_64 -y
```

2) Git 설치 command

```
: sudo yum install git -y
```

3) Jenkins 설치 command

```
: sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo  
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key  
sudo yum upgrade  
sudo yum install jenkins -y
```



```
Installed:
  java-1.8.0-openjdk-devel.x86_64 1:1.8.0.242.b08-0.50.amzn1

Dependency Installed:
  avahi-libs.x86_64 0:0.6.25-12.17.amzn1      cups-lib
  gnutls.x86_64 0:2.12.23-21.18.amzn1        java-1.8
  java-1.8.0-openjdk-headless.x86_64 1:1.8.0.242.b08-0.50.amzn1  jbigkit-
  libtiff.x86_64 0:4.0.3-32.34.amzn1         lksctp-t

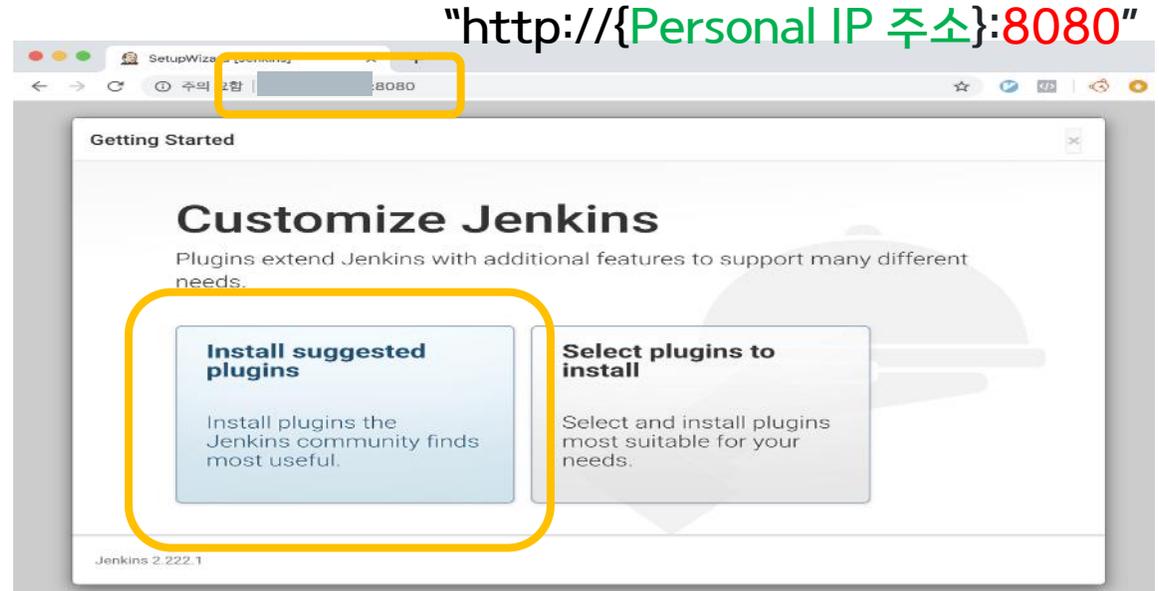
Complete!
```

2. Jenkins Service 시작 및 웹페이지 접속

* Jenkins Service 실행
command
: `sudo service jenkins start`

1) Jenkins Service 실행 후 새로 인터넷 창을 켜서
"http://{Personal IP 주소}:8080" 입력.
=> EC2 인스턴스의 인바운드 규칙에서 8080 포트
가 해제되어 있는지 확인!!

2) Plugins 설치 후 Admin User 생성.



Getting Started

Create First Admin User

계정명:

암호:

암호 확인:

이름:

이메일 주소:

1. Jenkins에 JDK 설치

JDK

JDK installations

Add JDK

JDK

Name JAVA8

Install automatically

Install Oracle Java SE Development Kit from the website

Version Java SE Development Kit 8u202

I agree to the Java SE Development Kit License Agreement

Installing JDK requires Oracle account. Please enter your username/password

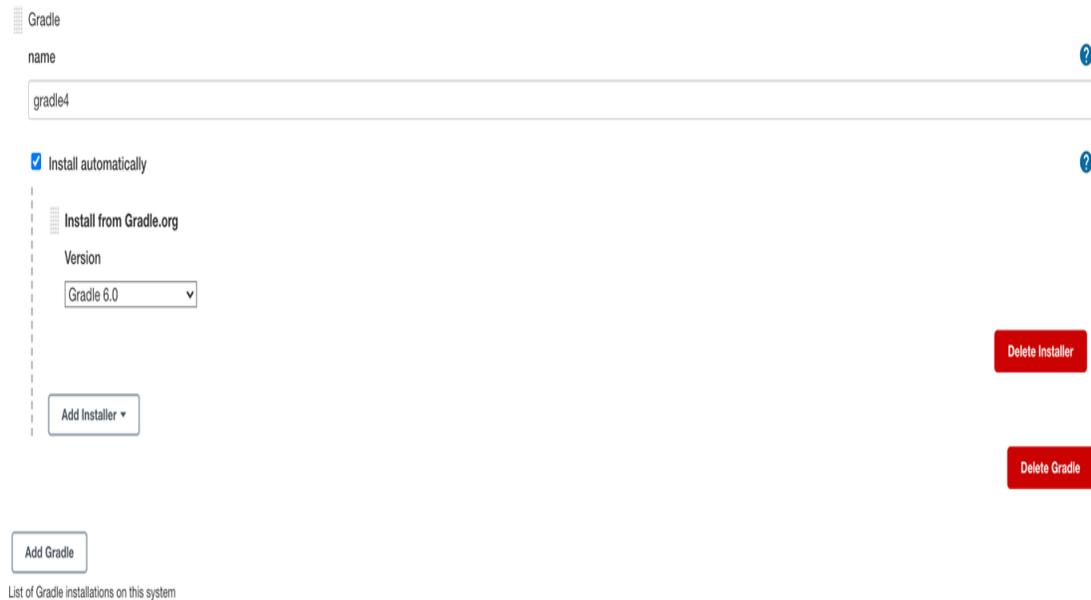
Oracle Java SE 11+ is not available for business, commercial or production use without a commercial license.

Public updates for Oracle Java SE 8 released after January 2019 will not be available for business, commercial or production use without a commercial license.

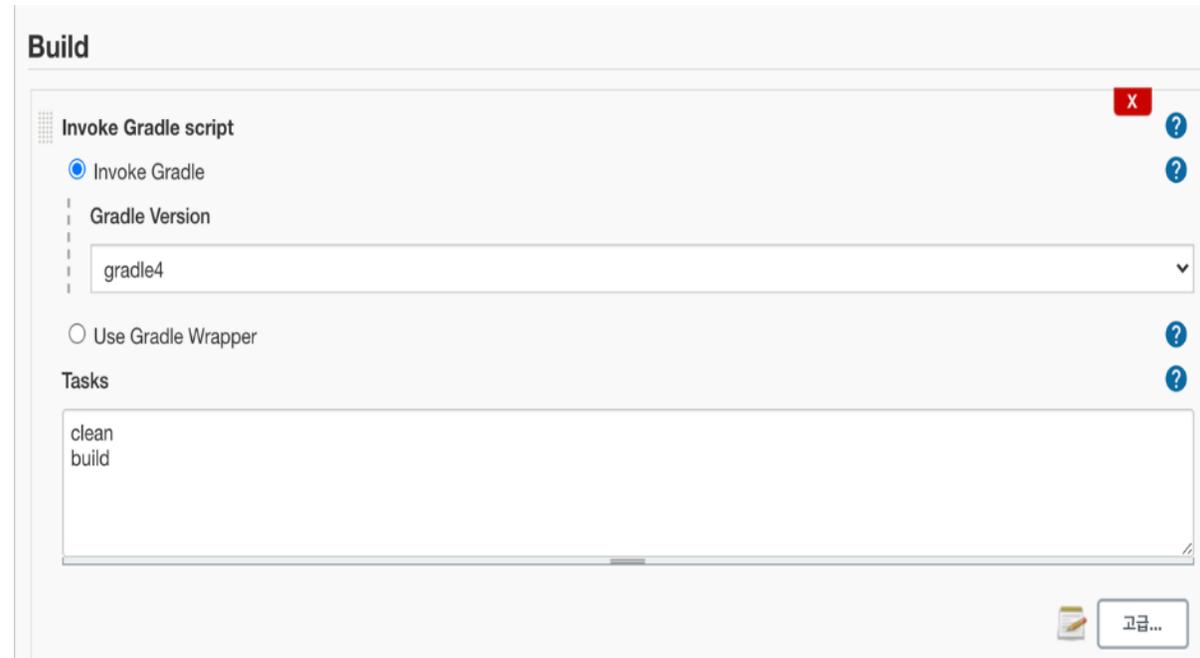
[Oracle Java SE Licensing FAQ](#)

Add JDK > JDK Name 작성 > Install automatically 체크 > Version 선택
> I agree to the JAVA SE Development Kit License Agreement 체크
> Please enter your username/password 클릭 후 Oracle ID/Password 입력

2. Jenkins에 Gradle Plugin 설치 & Global Tool Configuration 설정



The screenshot shows the Jenkins 'Global Tool Configuration' page for Gradle. The 'name' field is set to 'gradle4'. The 'Install automatically' checkbox is checked. Under the 'Install from Gradle.org' section, the 'Version' dropdown is set to 'Gradle 6.0'. There are two red buttons: 'Delete Installer' and 'Delete Gradle'. At the bottom, there is an 'Add Gradle' button and a link to 'List of Gradle installations on this system'.



The screenshot shows the Jenkins 'Build' configuration page. The 'Invoke Gradle script' section is selected. The 'Invoke Gradle' radio button is chosen. The 'Gradle Version' dropdown is set to 'gradle4'. The 'Use Gradle Wrapper' radio button is unselected. The 'Tasks' section contains 'clean' and 'build'. There is a red 'X' icon in the top right corner of the 'Invoke Gradle script' section and a '고급...' button at the bottom right.

1. 깃허브에 프로젝트 올릴 때 주의할 점

```
[root@ip-172-31-41-160 ~]# cd /var/lib/jenkins/workspace/test
[root@ip-172-31-41-160 test]# ls
build build.gradle gradle gradlew gradlew.bat settings.gradle src
```

프로젝트에 들어갔을 때 바로 build.gradle 파일이 존재해야 build를 인식할 수 있다.

/var/lib/Jenkins/workspace/{젠킨스 프로젝트 이름}
위치에 build.gradle이 있어야 한다.

따라서 깃허브에 올릴 때 폴더로 감싸지 않고 프로젝트를 바로 올려야 한다.

1. Github Token 발급 (Jenkins 관리 시스템 설정에서 사용)

Personal settings

Edit personal access token

If you've lost or forgotten this token, you can regenerate it, but be aware that any scripts or applications using this token will need to be updated. [Regenerate token](#)

Token description
jenkins

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 type="checkbox"/>	admin:org	Full control of orgs and teams
<input type="checkbox"/>	write:org	Read and write org and team membership
<input type="checkbox"/>	read:org	Read org and team membership
<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

2. Jenkins 관리 시스템 설정 (GitHub Server 탭에서 Credentials 생성)

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain
Global credentials (unrestricted)

Kind
Secret text

Scope
Global (Jenkins, nodes, items, all child items, etc)

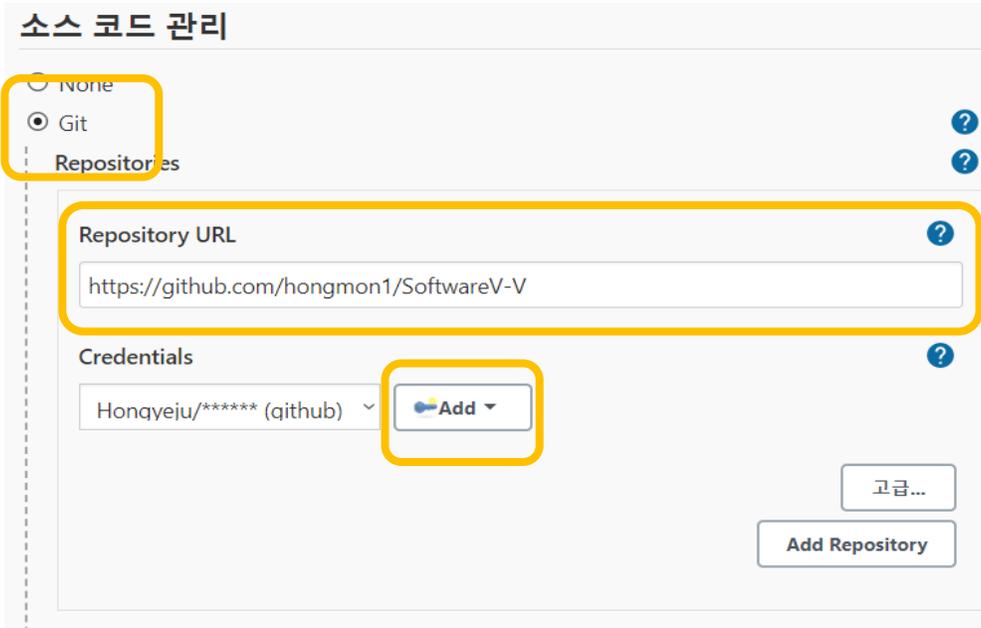
Secret

ID

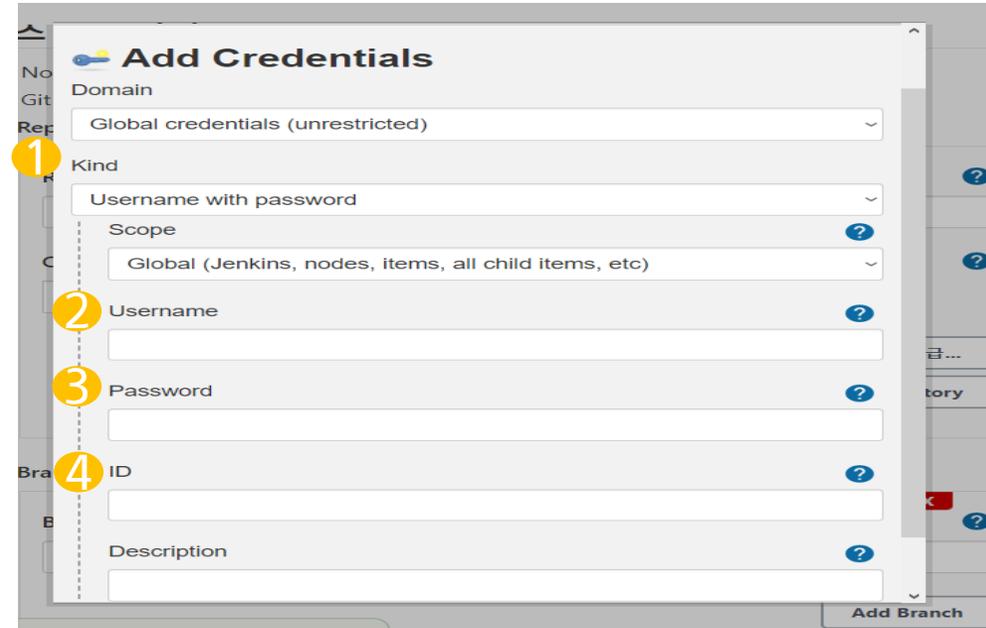
Description

- 1) Kind : Secret text로 설정
- 2) Secret : 앞에서 발급받은 깃허브 토큰 입력
- 3) ID : Github ID
- 4) Description - 해당 Credentials 정보

3. Jenkins Project 생성



- 1) Jenkins에서 Freestyle Project 생성한다.
- 2) 생성한 Project의 소스 코드 관리 메뉴에서
-> Git 체크
-> Repository URL에 레포지터리 주소 입력
-> Credentials ADD를 눌러준다.



- 1) Kind : Username with password 선택
- 2) Username : Jenkins 생성 시 입력한 이름
- 3) Password : Jenkins 계정 비밀번호
- 4) ID : Jenkins 계정 아이디

PART 4

Redmine 설치 & Git 연동

1. Bitnami 설치 파일 다운로드 및 실행

* Java 설치 command

```
: sudo wget https://bitnami.com/redirect/to/501550/bitnami-redmine-4.0.3-0-linux-x64-installer.run
```

- Bitnami 파일

: Subversion, PhpMyAdmin,
Redmine, Git을

한번에 설치가능하게 해주는 파일.

```
[root@ip-172-31-43-21 ~]# sudo wget https://bitnami.com/redirect/to/501550/bitnami-redmine-4.0.3-0-linux-x64-installer.run
--2021-04-02 07:14:51-- https://bitnami.com/redirect/to/501550/bitnami-redmine-4.0.3-0-linux-x64-installer.run
Resolving bitnami.com (bitnami.com)... 50.17.235.25
Connecting to bitnami.com (bitnami.com)|50.17.235.25|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://downloads.bitnami.com/files/stacks/redmine/4.0.3-0/bitnami-redmine-4.0.3-0-linux-x64-installer.run [following]
--2021-04-02 07:14:52-- https://downloads.bitnami.com/files/stacks/redmine/4.0.3-0/bitnami-redmine-4.0.3-0-linux-x64-installer.run
Resolving downloads.bitnami.com (downloads.bitnami.com)... 54.230.62.4, 54.230.62.69, 54.230.62.74, ...
Connecting to downloads.bitnami.com (downloads.bitnami.com)|54.230.62.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 215005234 (205M) [binary/octet-stream]
Saving to: 'bitnami-redmine-4.0.3-0-linux-x64-installer.run.3'

100%[=====>] 215,005,234 109MB/s in 1.9s

2021-04-02 07:14:54 (109 MB/s) - 'bitnami-redmine-4.0.3-0-linux-x64-installer.run.3' saved [215005234/215005234]
```

2. 설치파일 실행

* 설치파일 실행 command
: `./bitnami-redmine-4.0.3-0-linux-x64-installer.run`

```
-----
설치할 구성 요소를 선택하십시오. 설치하지 않을 구성 요소는 선택을 취소하십시오. 계속할 준비가 되면 다음을 클릭하십시오.
Subversion [Y/n] :y
PhpMyAdmin [Y/n] :y
Redmine : Y (Cannot be edited)
Git [Y/n] :n
위의 선택이 정확합니까? [Y/n]: y
-----
```

Git은 앞에서 Jenkins 사용을 위해 이미 설치했음

```
-----
Admin 계정 생성
Bitnami Redmine Stack 관리자 계정을 생성합니다.
이름 [User Name]: admin
이메일 주소 [user@example.com]: admin@google.com
로그인 계정명 [user]: admin
패스워드 :
패스워드를 재입력 :
```

Redmine 접속 계정
생성

3. Redmine 실행

```
* Redmine 재실행 command  
: sudo /opt/본인 레드마인  
폴더이름/ctlscript.sh restart
```

* Redmine 실행 후 새로 인터넷 창을 켜서
"http://{Personal IP 주소}:80" 입력.
=> EC2 인스턴스의 인바운드 규칙에서 80 포트가 해제되어
있는지 확인!!



Redmine 설치 시 주의사항

1. 각종 Warning 발생

1) MySQL requires a library that was not found in your system: libtinfo.so.5. Warning 발생

⇒ `sudo yum install ncurses-compat-libs`

2) Perl 모듈 Warning(Data::Dumper가 존재하지 않음)

⇒ `sudo yum install perl perl-Data-Dumper`

2. Memory 부족 발생 (Unable to create symbolic link 에러)

⇒ `sudo dd if=/dev/zero of=/mnt/swap.0 bs=1024 count=1048576`

⇒ `sudo mkswap /mnt/swap.0`

⇒ `sudo su`

⇒ `echo "/mnt/swap.0 swap swap defaults 0 0" >> /etc/fstab`

⇒ `swapon /mnt/swap.0`

⇒ `sudo swapon -s`

Redmine & Git 연동

1. AWS EC2에 Git 저장소 Clone

* 설치파일 실행 command

```
git config --global credential.helper 'store --file  
~/.credentials'  
git clone --mirror https://...../xxxx.git
```

EC2에 Redmine을 설치해뒀기 때문에,
Git 저장소를 클론해 같은 서버에 있는 것 처럼 속여야 한다.

```
[root@ip-172-31-43-21 redmine-4.0.3-0]# git clone --mirror https://github.com/hongmon1/SoftwareV-V.git  
Cloning into bare repository 'SoftwareV-V.git'...  
remote: Enumerating objects: 3, done.  
remote: Counting objects: 100% (3/3), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.
```

Redmine & Git 연동

2. Crontab 설정

redmine에서 사용할 로컬 저장소는 원 저장소의 변경 사항을 계속 받아와야한다.
=> crontab을 이용해 주기적으로 받아옴

1) git 저장소를 클론받아온 디렉토리(redmine 설치 디렉토리)에 update.sh를 작성

```
#!/bin/bash
cd /opt/redmine-4.0.3-0/SoftwareV-V.git
git remote update
```

2) Crontab 수정 (command : crontab -e)

```
* * * * * /opt/redmine-4.0.3-0/update.sh >> /opt/redmine-4.0.3-2/cron.log 2>&1
# crontab script end
```

3. Redmine 설정

설정

일반 표시방식 인증 API 프로젝트 사용자 일감 추적 시간추적 파일 메일 알림 수신 메일 **저장소**

지원할 SCM(Source Control Management)

SCM	명령
<input checked="" type="checkbox"/> Subversion	✓ svn
<input checked="" type="checkbox"/> Mercurial	! hg
<input checked="" type="checkbox"/> Cvs	! cvs
<input checked="" type="checkbox"/> Bazaar	! bzr
<input checked="" type="checkbox"/> Git	✓ git
<input type="checkbox"/> Filesystem	

SCM 명령을 config/configuration.yml에서 수정할 수 있습니다. 수정후에는 재시작하십시오.

커밋을 자동으로 가져오기

저장소 관리에 WS를 사용

API 키 키 생성

저장소 보기에 표시할 개정이력의 최대 갯수

Apply text formatting to commit messages

- Git 체크
- 커밋자동으로 가져오기 체크
- 저장소 관리에 WS를 사용 체크

4. Redmine 프로젝트 설정

저장소 추가

형상관리시스템

Git

주 저장소

식별자

test

1 에서 255 글자 소문자(a-z),숫자,대쉬(-)와 밑줄(_)만 가능합니다.
식별자는 저장후에는 수정할 수 없습니다.

저장소 경로 *

/opt/redmine-4.0.3-0

클라이언트 bare 저장소 (예: /gitrepo, c:\gitrepo)

경로 인코딩

기본: UTF-8

파일이나 폴더의 마지막 커밋을 보고

만들기 취소

- 저장소 경로
- : 저장소 경로는 위에서 clone받은 EC2 깃 저장소 위치를 작성한다.

5. 저장소에서 Redmine & Git 연동 확인

test

+ 개요 작업내역 일감 소요 시간 Gantt 차트 달력 뉴스 문서 위키 파일 **저장소** 설정

test @ master

이름

📄 사용툴.txt

최근 개정판

#	날짜	저자	
● 41005c1f	2021/03/30 08:32	hongmon1	test commit

Redmine & Git 연동 시 주의할 점

1. Git 저장소 자동 업데이트 문제

1) update.sh 파일이 실행파일 상태가 아니면 crontab에서 자동으로 업데이트를 수행할 수가 없다.

```
* update.sh 파일을 실행파일로 바꾸는 command  
: chmod +x update.sh
```

2) 그 후 crontab -e로 crontab 내용도 update.sh 실행파일 수행으로 수정한다.

```
* * * * * /opt/redmine-4.0.3-0/./update.sh >> /opt/redmine-4.0.3-0/cron.log 2>&1  
# crontab script end
```

3) 수행이 잘 되었는지 확인해보고 싶다면 cron.log 파일을 확인해본다.

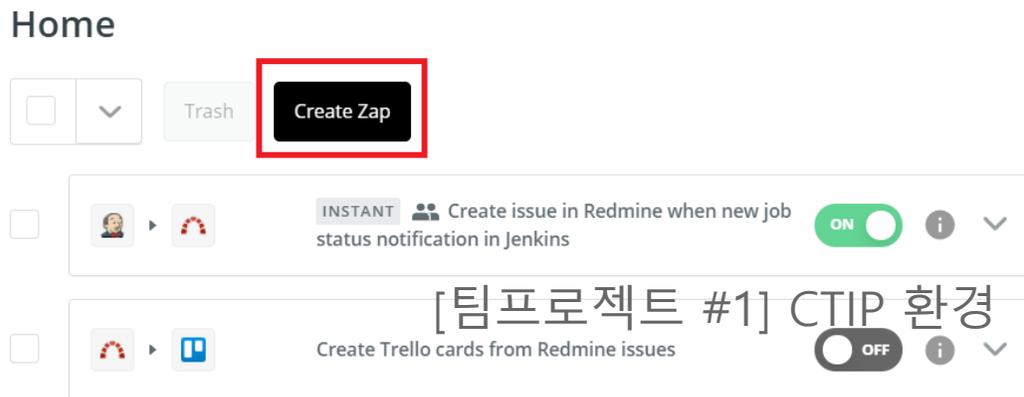
```
From https://github.com/hongmon1/SoftwareV-V  
 751bb9f..61dad12  master    -> master  
Fetching origin  
Fetching origin  
From https://github.com/hongmon1/SoftwareV-V  
 61dad12..9d8ea55  master    -> master  
Fetching origin  
Fetching origin
```

PART 5

Jenkins & Redmine 연동 with Zapier

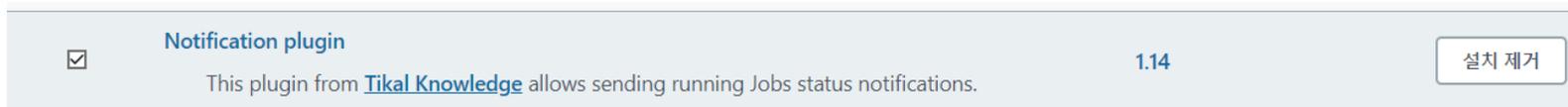
Jenkins & Redmine 연동 - with Zapier

1. Zapier에서 새로운 Zap을 생성

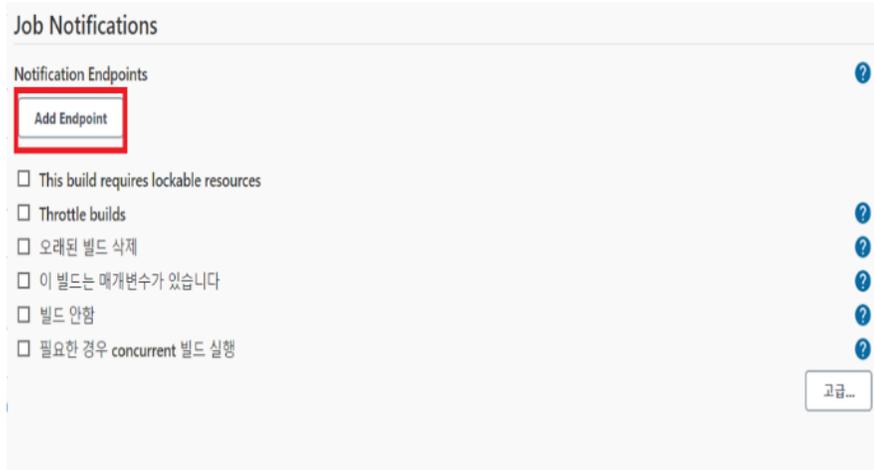


- 1) Trigger -> Jenkins 선택(입력)
- 2) Trigger Event -> New Job Status Notification 선택
- 3) Webhook URL 복사
- 4) Test Trigger
- 5) Action -> Redmine 선택(입력)
- 6) Action Event -> Create Issue 선택(사용자 재량)
- 7) Redmine 연동

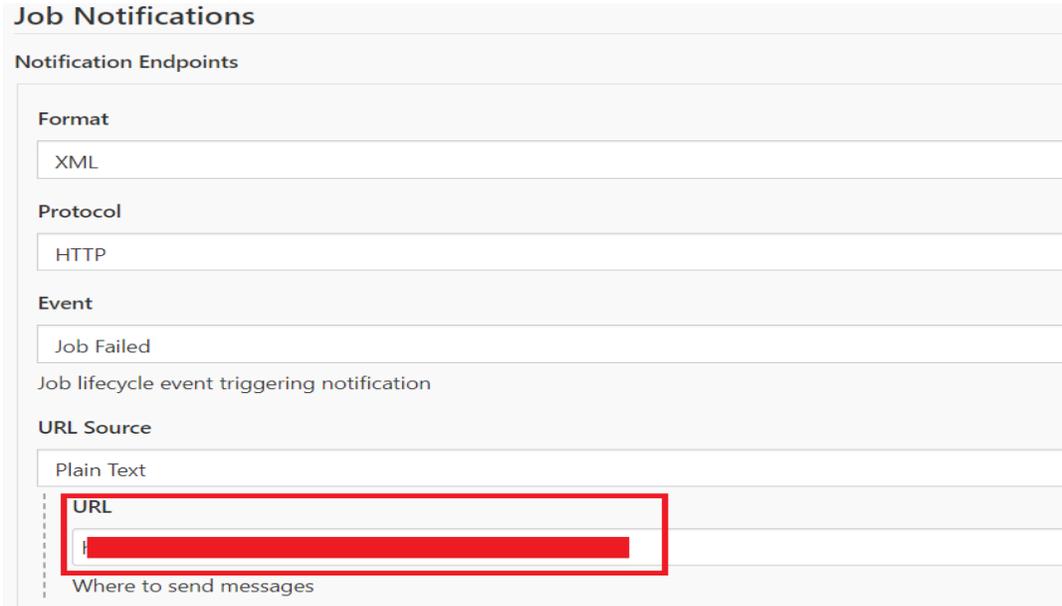
2. Jenkins에서 Notification Plugin 설치



3. Jenkins 프로젝트 설정



연동할 프로젝트 -> 구성 -> 소스코드관리 위에
Job Notifications 항목에서 Add Endpoint 클릭.



- 1) Format : XML
- 2) Protocol : HTTP
- 3) Event : 자유 선택(해당 이벤트 발생 시 zapier 쪽에 알림
이
가며 redmine이 수신한다)
- 4) URL Source : 위에서 zapier에서 jenkins 연결 때 복사한
링크를 붙여넣는다.

4. Test

젠킨스에서 build 시 Redmine에 일감이 생긴다면 성공

SoftwareV-V

개요 [작업내역](#) [일감](#) [소요 시간](#) [Gantt 차트](#) [달력](#) [뉴스](#) [문서](#) [위키](#) [파일](#) [저장소](#)

개요

일감 추적

	진행중	완료됨	합계
결함	32	0	32
새기능	1	0	1
지원	0	0	0

[모든 일감 보기](#) | [요약](#) | [달력](#) | [Gantt 차트](#)

소요 시간

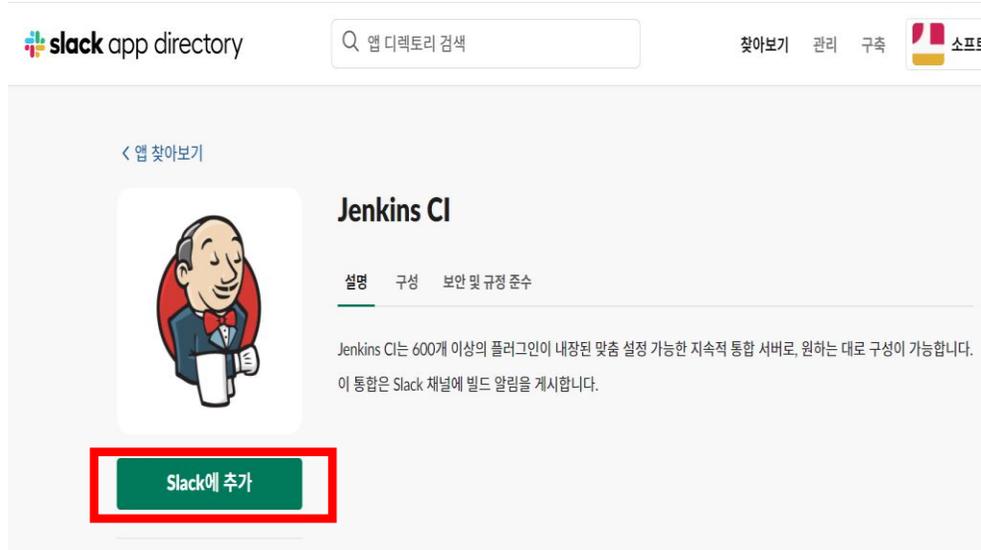
0.00 시간

[자세히](#) | [보고서](#)

PART 6

Slack & Jenkins 연동

1. Slack에 Jenkins CI 추가



설치되면 왼쪽 탐색 모음에서 Jenkins 관리를 다시 클릭한 후 시스템 구성으로 이동합니다. 글로벌 Slack Notifier 설정 섹션을 찾아 다음 값을 추가합니다.

- 팀 하위 도메인: `vv-8ji4438`
- 통합 토큰 자격 증명 ID: `XyZwZEShW25h71RAcK9uZPC` 토큰을 값으로 사용하여 암호 텍스트 자격 증명을 생성합니다.

기타 필드는 선택 사항입니다. 자세한 내용을 알아보려면 옆에 있는 물음표 아이콘을 클릭하세요. 작업을 완료하면 저장을 누릅니다.

참고: 아래 스크린샷에 나온 통합 토큰을 자신의 토큰으로 교체해야 합니다.

팀 하위 도메인 : 우리 팀 슬랙 주소

통합 토큰 자격 증명 ID : 이를 통하여 credential key 생성

2. Jenkins 플러그인 설정에서 Slack Notification Plugin 다운로드

Slack Notification Plugin

Integrates Jenkins with Slack, allows publishing build statuses, messages and files to Slack channels.

3. Jenkins에서 Slack 시스템 설정

Slack

Workspace ?

vv-8ji4438

Credential ?

this_is_real_slack Add

Default channel / member id ?

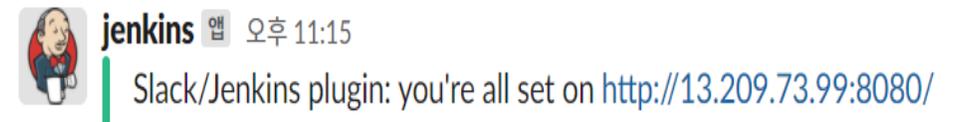
Custom slack app bot user ?

고급...

Test Connection

#주의할점#

test connection을 눌러야 슬랙으로 알람이 가면서 제대로 연결된 지 확인할 수 있다.





Q&A