

S o f t w a r e V & V

CTIP 환경 구축

T	E	A	M		2
2012	12	5	19	김 선	우
2015	10	6	24	김 용	현
2016	11	2	61	민 지	호
2016	11	2	93	전 다	윤

CONTENTS

01

SonaQube

02

PMD

03

Check Style

04

Spot Bug

05

Test Link

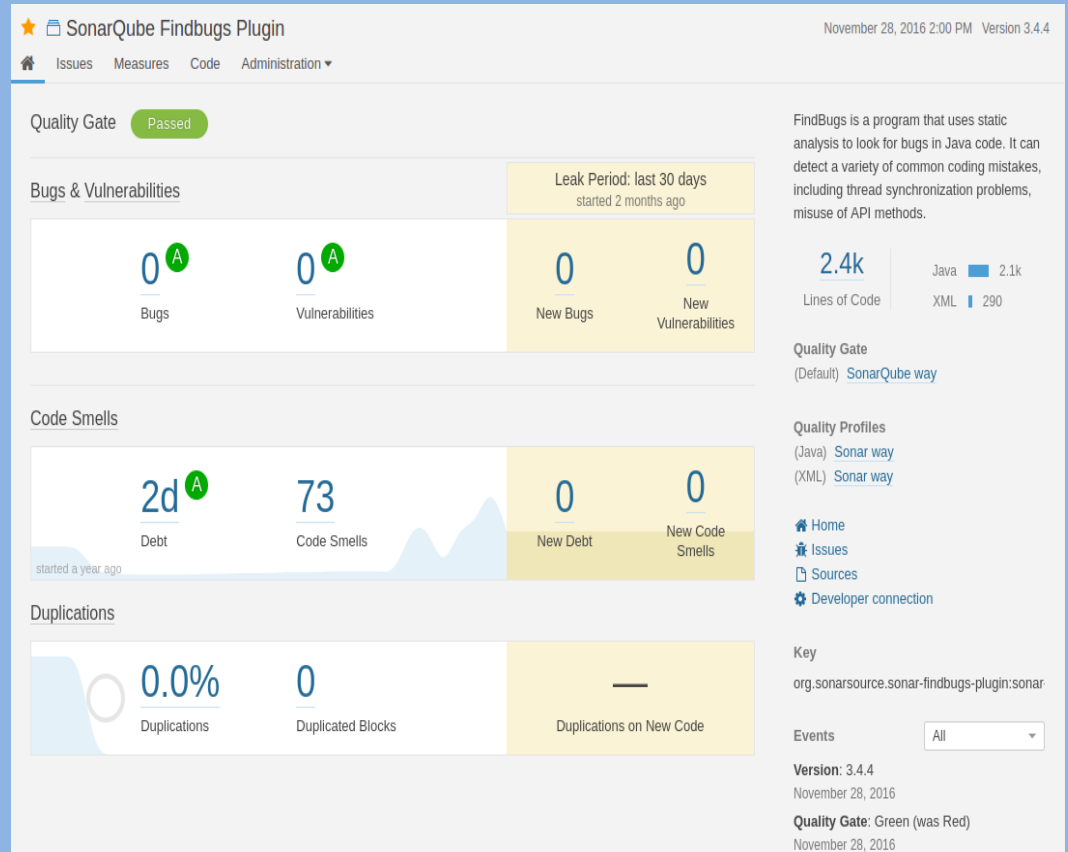
06

Overall Structure

Sonar Qube



20개 이상의 프로그래밍 언어에서 버그, 코드 스멜, 보안 취약점을 발견할 목적으로 정적 코드 분석으로 자동 리뷰를 수행하기 위한 지속적인 코드 품질 검사용 오픈 소스 플랫폼



01

```
ubuntu@ip-172-31-17-171:~$ wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.3.1.34397.zip
```

```
#sonar.web.port=9000
# TCP port for incoming HTTP connections. Default value is 9000.
sonar.web.port=9000
```

```
#----- PostgreSQL 9.3 or greater
# By default the schema named "public" is used. It can be overridden with the parameter "currentSchema".
sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube?currentSchema=my_schema
-----
# - The embedded H2 database is used by default. It is recommended for tests but
-----
# Same as previous property, but allows to not repeat all other settings like -Xmx
sonar.web.javaAdditionalOpts=-server
licensed products.

# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
sonar.jdbc.username=sonar
sonar.jdbc.password=sonar
```

01

```
ubuntu@ip-172-31-17-171:~$ sudo apt-get install postgresql
```

```
ubuntu@ip-172-31-17-171:~$ sudo -i -u postgres
postgres@ip-172-31-17-171:~$ psql
psql (10.12 (Ubuntu 10.12-0ubuntu0.18.04.1))
Type "help" for help.

postgres=# CREATE USER sonar WITH ENCRYPTED PASSWORD 'sonar';
CREATE ROLE
postgres=# CREATE DATABASE sonar OWNER sonar;
CREATE DATABASE
postgres=# ALTER ROLE sonar WITH createdb;
ALTER ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE sonar TO sonar;
GRANT
postgres=# █
```

```
ubuntu@ip-172-31-17-171:/usr/lib/systemd/system$ sudo vi sonar.service
```

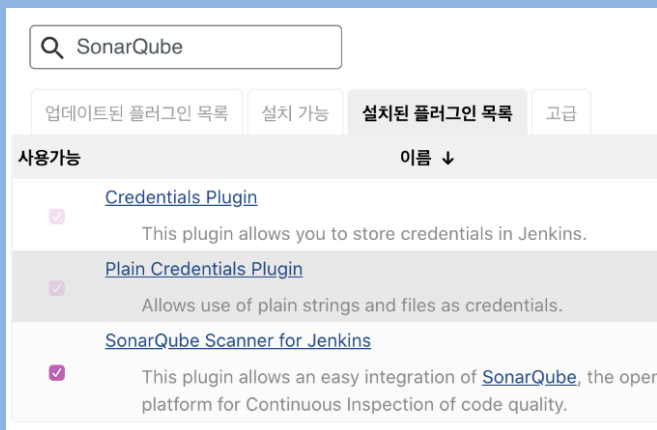
01

```
ubuntu@ip-172-31-17-171:/usr/lib/systemd/system$ systemctl daemon-reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: Ubuntu (ubuntu)
Password:
==== AUTHENTICATION COMPLETE ====
ubuntu@ip-172-31-17-171:/usr/lib/systemd/system$ systemctl enable sonar
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: Ubuntu (ubuntu)
Password:
==== AUTHENTICATION COMPLETE ====
Created symlink /etc/systemd/system/multi-user.target.wants/sonar.service → /usr/lib/systemd/system/sonar.service.
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: Ubuntu (ubuntu)
Password:
==== AUTHENTICATION COMPLETE ====
ubuntu@ip-172-31-17-171:/usr/lib/systemd/system$ systemctl start sonar
```

```
root@ip-172-31-17-171:~# ufw allow 9000
Rules updated
Rules updated (v6)
```

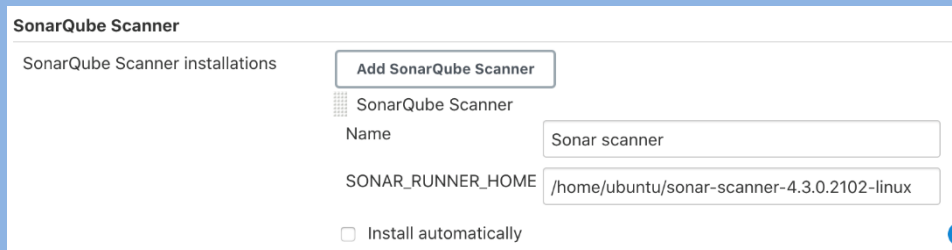
```
ubuntu@ip-172-31-17-171:~$ wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.3.0.2102-linux.zip  
--2020-05-17 08:58:22-- https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.3.0.2102-linux.zip
```

Sonarqube scanner 설치



Jenkins에 Sonarqube scanner 플러그인 설치

01



SonarQube Scanner

SonarQube Scanner installations

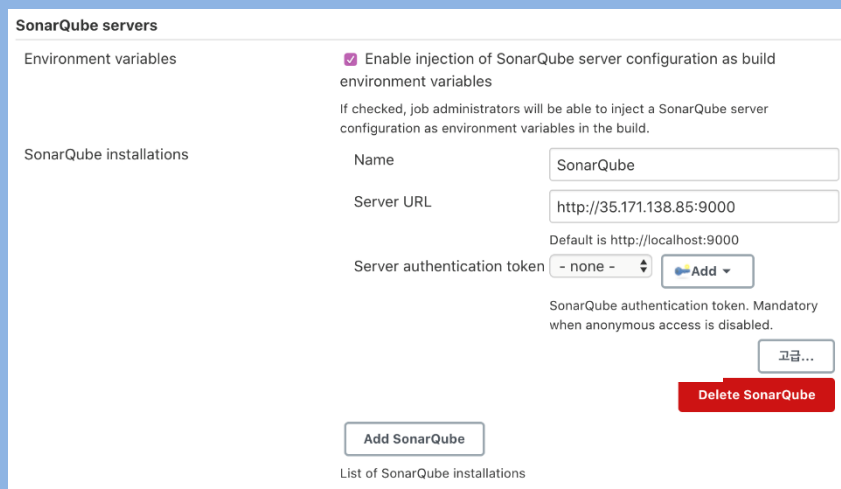
SonarQube Scanner

Name

SONAR_RUNNER_HOME

Install automatically

Jenkins에 Sonarqube scanner install 경로 설정



SonarQube servers

Environment variables

Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

List of SonarQube installations

Jenkins에 Sonarqube Url 추가

Execute SonarQube Scanner

Task to run

JDK

JDK to be used for this SonarQube analysis

Path to project properties

Analysis properties

```
sonar.projectKey = DWS
sonar.projectName= DWS
sonar.login=admin
sonar.password=admin
sonar.host.url = http://localhost:9000/
sonar.sources=src/main/java
sonar.tests=src/test/java
sonar.projectVersion=1.0
sonar.sourceEncoding=UTF-8
sonar.java.binaries=build/classes
```

Additional arguments

JVM Options

1. 프로젝트 빌드 구성에 들어가 빌드 시, Execute sonarqube scanner를 선택한다.
2. Analysis properties를 명시한다.
(프로젝트 내에 sonar.properties 파일로 만들어줘도 된다.)

PMD



응용 프로그램 코드에서 발견된 문제를 보고 하는 오픈 소스 정적 소스 코드 분석기

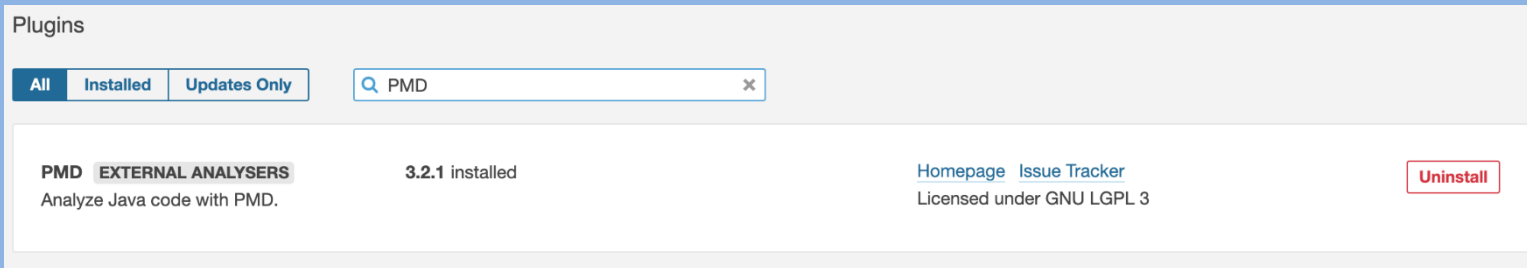
PMD는 내장 규칙 세트가 포함되어 사용자 정의 규칙을 작성하는 기능을 지원

올바른 형식의 소스 파일만 처리할 수 있으므로 컴파일 오류를 보여주지 않음

▼ Repository

SonarAnalyzer Java	549
FindBugs Java	448
SonarAnalyzer C#	379
Pylint Python	370
FindBugs Contrib Java	307
PMD Java	268
SonarAnalyzer JavaScript	221
SonarAnalyzer PHP	175
Checkstyle Java	172
SonarAnalyzer TypeScript	167

소나큐브에 PMD 플러그인 추가



PMD ruleset 생성

```
<rule ref="category/java/bestpractices.xml/AbstractClassWithoutAbstractMethod" />
<rule ref="category/java/design.xml/AbstractClassWithoutAnyMethod" />
<rule ref="category/java/codestyle.xml/AbstractNaming" />
<rule ref="category/java/bestpractices.xml/ArrayIsStoredDirectly" />
<rule ref="category/java/errorprone.xml/AssignmentInOperand" />
<rule ref="category/java/codestyle.xml/AtLeastOneConstructor" />
<rule ref="category/java/errorprone.xml/AvoidAssertAsIdentifier" />
```

참조

https://pmd.github.io/latest/pmd_rules_java_errorprone.html

<https://github.com/jensgerdes/sonar-pmd/blob/master/docs/RULES.md>

02

New Profile

Name*
softwareV_V

Language*
Java

Parent:
None

PMD
Choose File pmd-rule-set.xml
Optional configuration file

Create Cancel

생성한 ruleset.xml 파일을 이용하여 Profile 생성

▼ Quality Profile ? SOFTWARE... Clear

Drupal PHP (Built-in)

PMD Java

PSR-2 PHP (Built-in)

softwareV_V Java active inactive

softwareV_V가 Quality Profile에 추가됨

03

Check Style

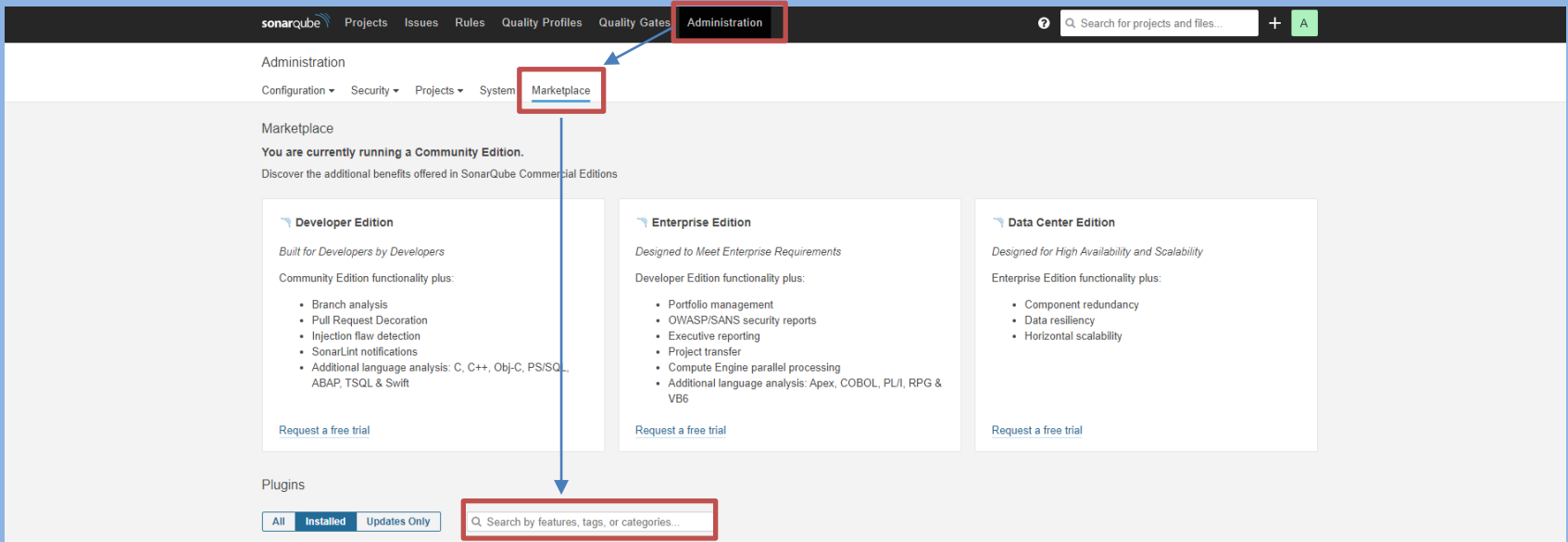


Java 코드가 코딩규칙을 준수하는지 확인하기 위해 개발에 사용되는 정적 코드 분석 도구

코드의 품질, 가독성, 재사용성을 향상시키고 개발 비용을 줄일 수 있으며 수행하는 검사는 주로 코드 표시로 제한된다

▼ Repository

SonarAnalyzer Java	549
FindBugs Java	448
SonarAnalyzer C#	379
Pylint Python	370
FindBugs Contrib Java	307
PMD Java	268
SonarAnalyzer JavaScript	221
SonarAnalyzer PHP	175
Checkstyle Java	172
SonarAnalyzer TypeScript	167



1. Sonarqube에 접속해서 Administration -> Marketplace를 선택한 후, plugin에서 checkstyle을 install 한다.
2. Sonarqube를 재실행하면 checkstyle을 사용 가능하다.

Checkstyle **EXTERNAL ANALYSERS**
Analyze Java code with Checkstyle

4.31 installed

[Issue Tracker](#)
Licensed under [LGPL-3.0](#)
Developed by [Checkstyle](#)

[Uninstall](#)

Spot Bug



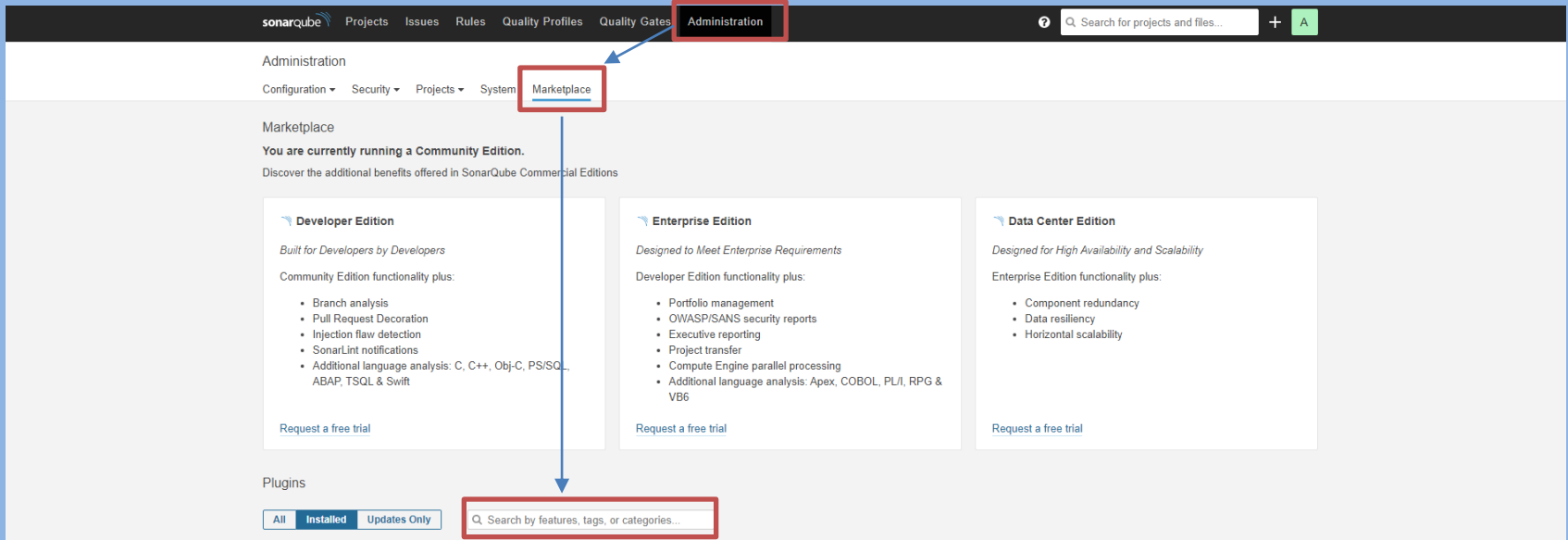
자바 바이트 코드를 분석하여 버그 패턴을 발견하는 정적 분석 공개 소프트웨어

기존 FindBugs를 계승한 프로젝트이다.

플러그인으로 웹 보안 감사지원하는 FindSecurityBugs가 있다.

▼ Repository

SonarAnalyzer Java	549
FindBugs Java	448
SonarAnalyzer C#	379
Pylint Python	370
FindBugs Contrib Java	307
PMD Java	268
SonarAnalyzer JavaScript	221
SonarAnalyzer PHP	175
Checkstyle Java	172
SonarAnalyzer TypeScript	167



1. Sonarqube에서는 Findbugs로 검색 해야한다.
2. Sonarqube에 접속해서 Administration -> Marketplace를 선택한 후, plugin에서 Findbugs을 install 한다.
3. Sonarqube를 재실행하면 Findbugs을 사용 가능하다.

Findbugs **EXTERNAL ANALYSERS** 4.0.0 installed
Analyze Java, Scala, Closure and JSP code with SpotBugs. 4.0.0

[Homepage](#) [Issue Tracker](#)
Licensed under [GNU LGPL 3](#)
Developed by [SpotBugs Team](#)

Uninstall

Test Link

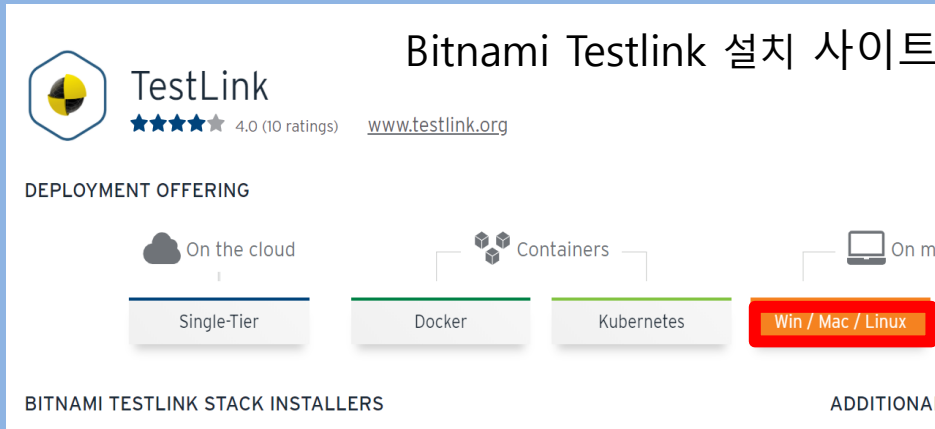


- 웹 환경 기반의 테스트를 진행
- 테스트의 우선순위 부여 기능 및 사용자의 활동영역 정의(리더, 테스터)
- 다양한 종류의 DB를 지원(MySQL, PostgreSQL, MS-SQL 등)
- 트리구조를 이용한 테스트 케이스 작성기능

-설치/연동 방법-

1. Ubuntu에 TestLink 설치
2. Jenkins에 TestLink PlugIn 설치
3. Jenkins 환경 설정
4. TestLink 환경 설정
5. JenKins Item 환경 설정

1. Ubuntu에 TestLink 설치



Bitnami Testlink 설치 사이트

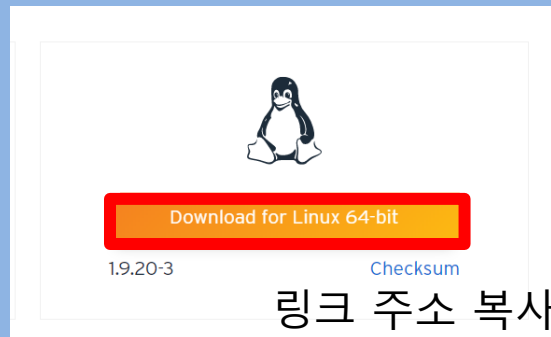
TestLink 4.0 (10 ratings) www.testlink.org

DEPLOYMENT OFFERING

On the cloud Containers On my

Single-Tier Docker Kubernetes Win / Mac / Linux

BITNAMI TESTLINK STACK INSTALLERS ADDITIONAL



Download for Linux 64-bit

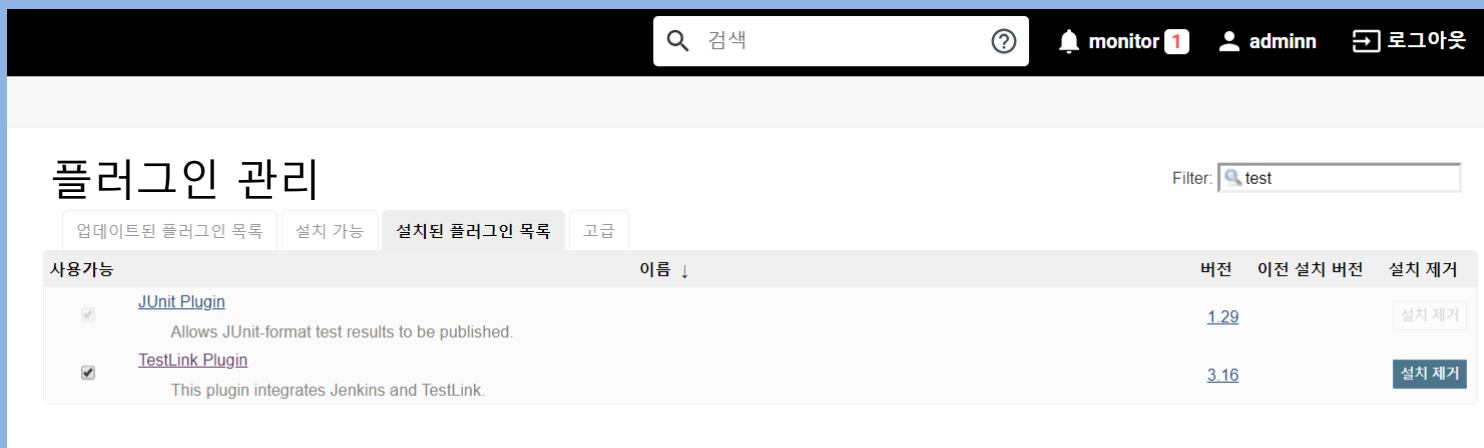
1.9.20-3 Checksum

링크 주소 복사

```
wget https://bitnami.com/redirect/to/1004236/bitnami-testlink-1.9.20-3-linux-x64-installer.run
```

```
./https://bitnami.com/redirect/to/1004236/bitnami-testlink-1.9.20-3-linux-x64-installer.run
```

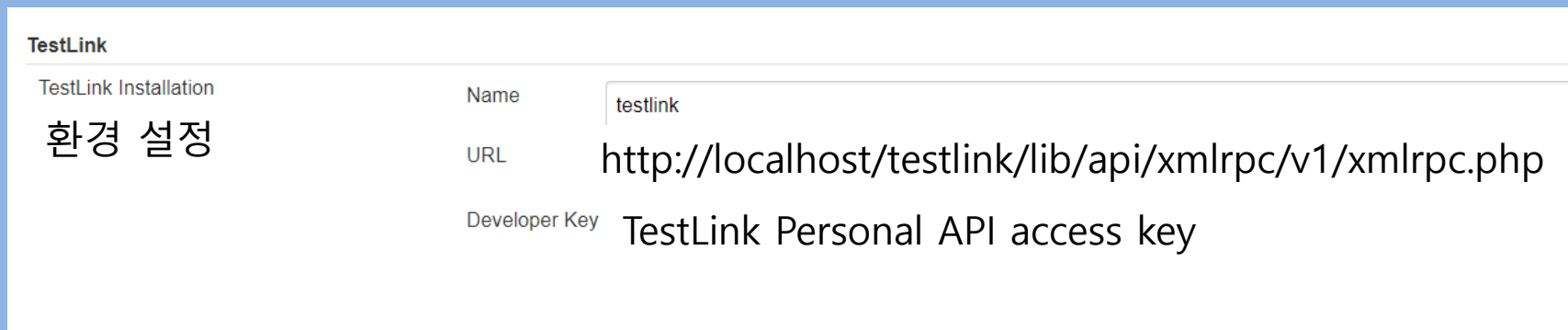
2. Jenkins에 TestLink PlugIn 설치



The screenshot shows the Jenkins 'Plugin Management' page. The search filter is set to 'test'. Two plugins are listed:

사용가능	이름 ↓	버전	이전 설치 버전	설치 제거
<input checked="" type="checkbox"/>	JUnit Plugin Allows JUnit-format test results to be published.	1.29		설치 제거
<input checked="" type="checkbox"/>	TestLink Plugin This plugin integrates Jenkins and TestLink.	3.16		설치 제거

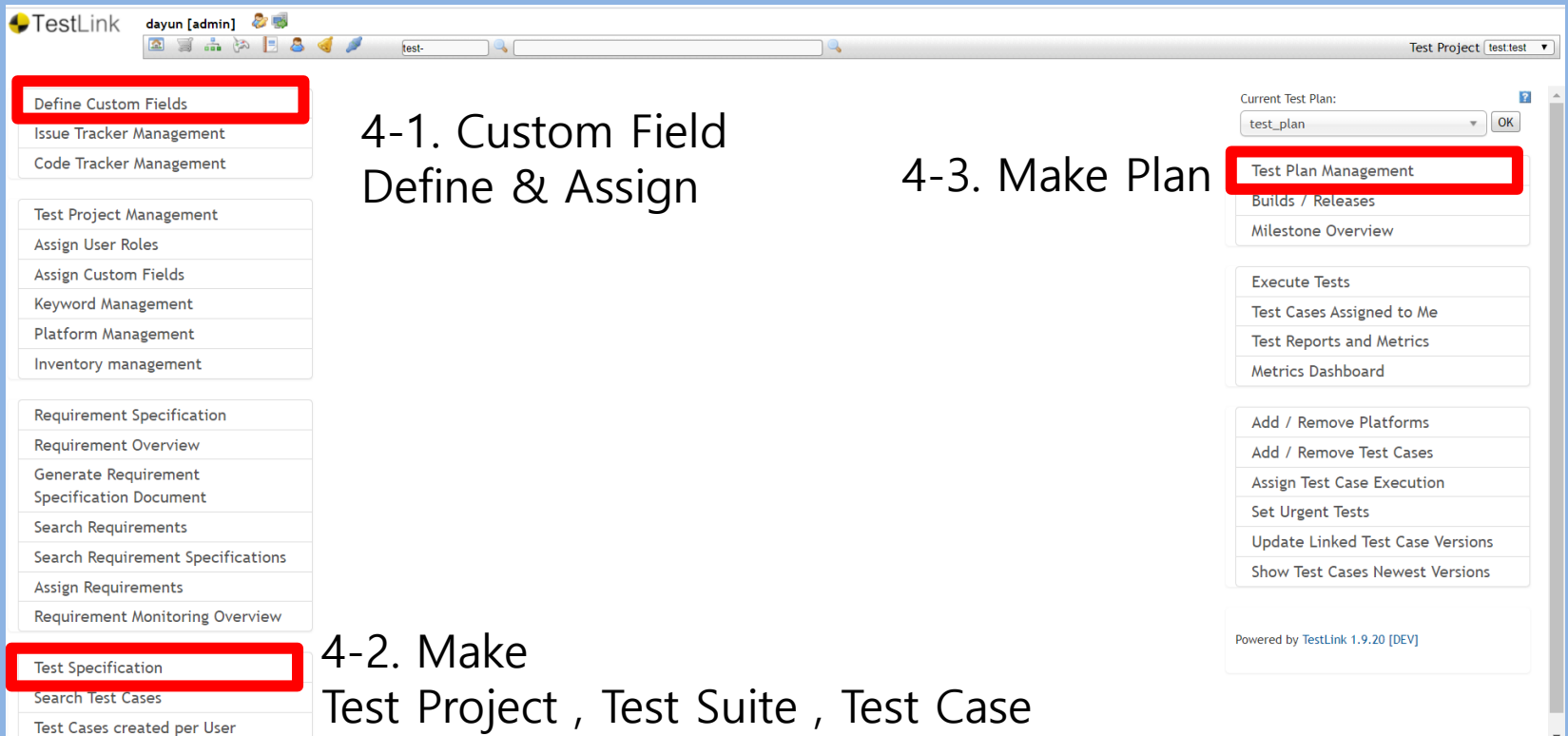
3. Jenkins에 환경 설정



The screenshot shows the 'TestLink' configuration page in Jenkins. The configuration is as follows:

TestLink	
TestLink Installation	
환경 설정	
Name	testlink
URL	http://localhost/testlink/lib/api/xmlrpc/v1/xmlrpc.php
Developer Key	TestLink Personal API access key

4. TestLink 환경 설정



The screenshot shows the TestLink administration interface. On the left, there is a navigation menu with several categories. The 'Define Custom Fields' option is highlighted with a red box. Below it, 'Test Specification' is also highlighted with a red box. The main content area is divided into three sections: '4-1. Custom Field Define & Assign', '4-2. Make Test Project , Test Suite , Test Case', and '4-3. Make Plan'. The 'Test Plan Management' option in the right-hand menu is highlighted with a red box. The interface includes a top navigation bar with the user name 'dayun [admin]' and the current project 'Test Project | test.test'. The bottom of the page shows the version information 'Powered by TestLink 1.9.20 [DEV]'.

4-1. Custom Field Define & Assign

4-3. Make Plan

4-2. Make Test Project , Test Suite , Test Case

5. JenKins Item 환경 설정

Invoke TestLink

TestLink Configuration

TestLink Version	<input type="text" value="testlink"/>	?
Test Project Name	<input type="text" value="test"/>	?
Test Plan Name	<input type="text" value="test_plan"/>	?
Build Name	<input type="text" value="examples-\$BUILD_NUMBER"/>	?
Custom Fields	<input type="text" value="java class"/>	?
Test Plan Custom Fields	<input type="text"/>	?

고급...

Result Seeking Strategy

Test Result Seeking Strategies

JUnit class name

Include Pattern	<input type="text" value="**/TEST-test1Test.xml"/>	?
Key Custom Field	<input type="text" value="test1Test"/>	?
Attach JUnit XML	<input checked="" type="checkbox"/>	?
Include test notes	<input checked="" type="checkbox"/>	?

Add strategy ▾

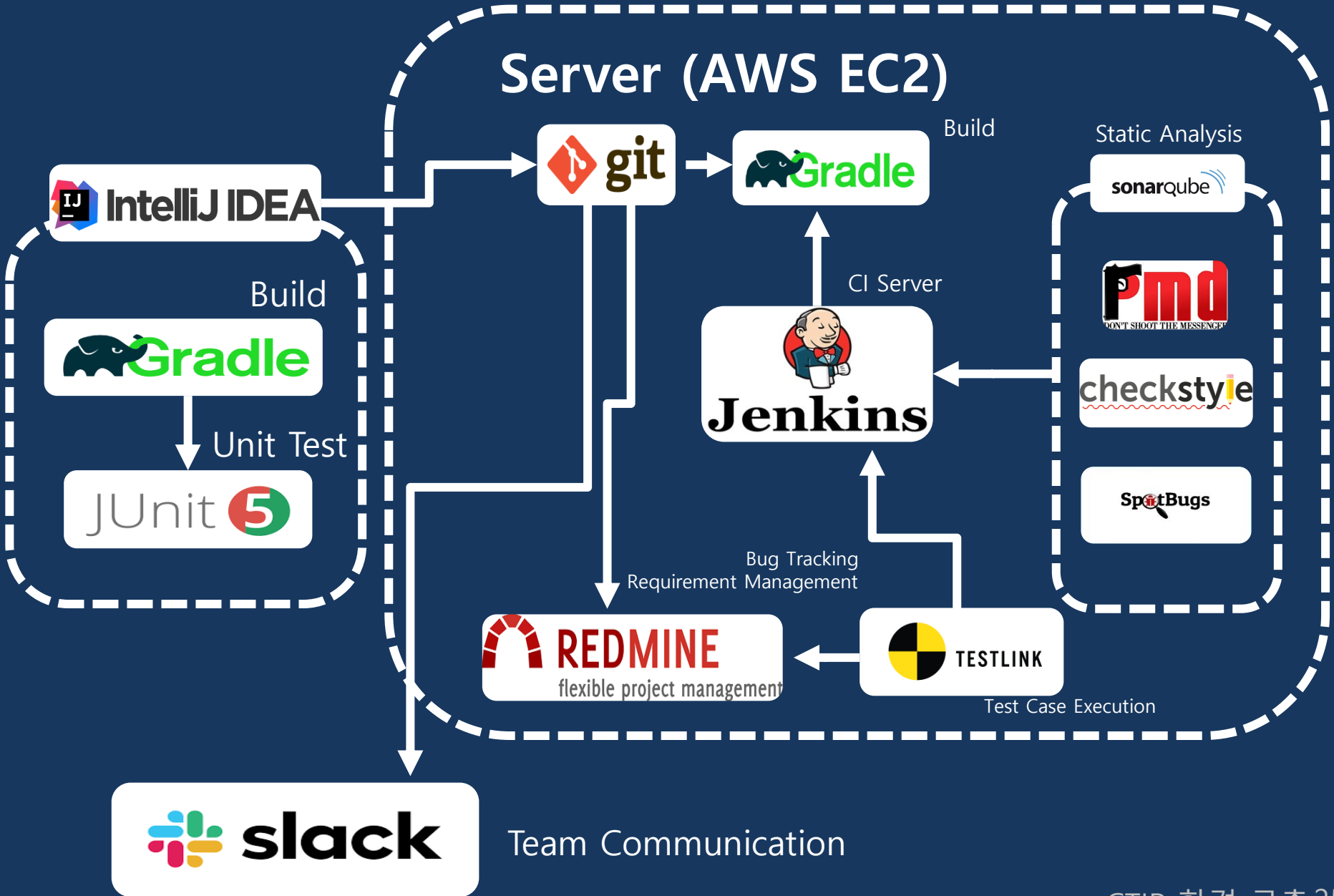
1. Redmine – Testlink 연동

-> Redmine의 요구사항으로부터 Testlink test case 생성

2. Redmine, Testlink, Sonarqube에서 환경 전수를 위한 Project configuration

Overall structure

s o f t w a r e v & v



**THANK
YOU**