

Software Verification

#3 정적분석 도구, 단위/시스템테스트 도구

Software Verification Team 4

강 송 신 정 상 승 모 연 화

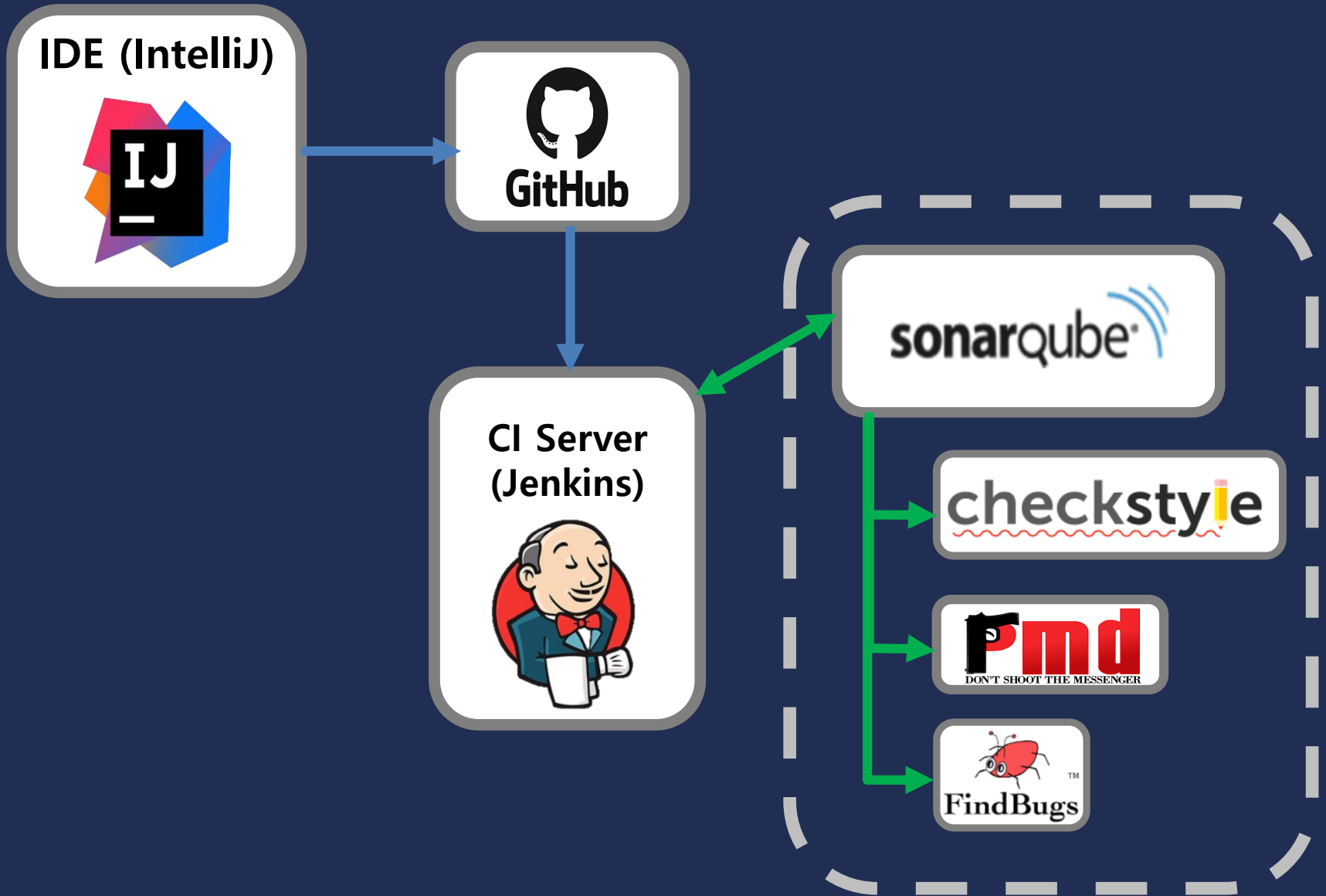
Software Verification

#3 정적분석 도구, 단위/시스템테스트 도구

CONTENTS

- ⊕ 01 Overall Structure
- ⊕ 02 Static analyzer – SonarQube
- ⊕ 03 Static analyzer – PMD
- ⊕ 04 Static analyzer – Checkstyle
- ⊕ 05 Static analyzer – FindBugs
- ⊕ 06 Summary

01 Overall structure



01 Overall structure

I. Jenkins – Redmine 연동 문제

→ plugin 노후화 / gem version 충돌 문제

급한대로 바로가기를 통한 연결. 해결될 가능성은 낮지만 고민해봐야 할 문제.

(Redmine version 4.0.3 ⇒ 3.x.x 로 downgrade ??)

II. Jenkins – Mantis 연동 문제

→ 추후 해결이 안될 시 GitHub or TestLink – Mantis 대체 연동을 고려

01 Overall structure

I. Jenkins – Redmine 연동 문제

→ zapier webhook으로 해결.

하지만 아직 일감 분류 문제 해결이 필요.

SonarQube Quality Gate

sv_project **OK**
server-side processing: **Success**

고정링크

- [Last build_. \(#122\), 12 sec 전](#)
- [Last stable build_. \(#121\), 28 sec 전](#)
- [Last successful build_. \(#121\), 28 sec 전](#)
- [Last failed build_. \(#120\), 4 min 43 sec 전](#)
- [Last unsuccessful build_. \(#120\), 4 min 43 sec 전](#)
- [Last completed build_. \(#121\), 28 sec 전](#)

| Build ID | Time |
|----------|--------------------|
| #122 | 2019. 5. 2 오후 9:14 |
| #121 | 2019. 5. 2 오후 9:14 |
| #120 | 2019. 5. 2 오후 9:10 |
| #119 | 2019. 5. 2 오후 9:08 |

| # | 유형 | 상태 | 우선순위 | 제목 |
|-----|----|----|------|---|
| 294 | 결함 | 신규 | 보통 | software_verification_project / 126 / SUCCESS |
| 293 | 결함 | 신규 | 보통 | software_verification_project / 125 / SUCCESS |
| 292 | 결함 | 신규 | 보통 | software_verification_project / 124 / SUCCESS |
| 291 | 결함 | 신규 | 보통 | software_verification_project / 123 / FAILURE |
| 290 | 결함 | 신규 | 보통 | software_verification_project / 122 |
| 289 | 결함 | 신규 | 보통 | software_verification_project / 121 |
| 288 | 결함 | 신규 | 보통 | software_verification_project / 120 |
| 287 | 결함 | 신규 | 보통 | software_verification_project / 119 |

01 Overall structure

Ⅱ. Jenkins – Mantis 연동 문제

- TestLink – Mantis 대체 연동 시 TestLink – Redmine 연동이 불가
- 정적 분석 툴 추가 ⇒ 삭제
- 대체제로 testcase 설치 예정 (현재 DB 설정 충돌 문제 발생)

02 Static Analyzer - SonarQube



- 조직에서 개발된 코드의 지속적인 인스펙션을 통해 품질 목표를 달성할 수 있게 해주는 플랫폼
- 소스코드 품질 현황을 시각화, 리스크 분석 및 소스코드에서 발생하는 문제를 해결



지속적인 인스펙션

지속적인 통합처럼 지속적으로 코드 인스펙션을 통해 품질 문제를 해결합니다.



품질 중앙화

조직의 다양한 언어로 개발된 코드의 품질을 가시화하고 단일 위치에 관리합니다.



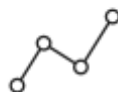
DevOps 통합

SonarQube는 다양한 빌드 시스템과 CI 엔진을 지원하여 DevOps에 쉽게 통합됩니다.



품질 요구사항 설정

품질 게이트를 통해 표준화된 코드 품질 요구사항을 설정할 수 있습니다.



다중 언어 분석

SonarQube는 20개의 프로그램 언어에 대한 코드 분석을 지원합니다.



플러그인 라이브러리

다수의 플러그인을 통해 SonarQube의 기능을 확장할 수 있습니다.

02 Static Analyzer - SonarQube

- SonarQube의 경우 기본 DB로 maven의 H2DB를 사용하므로, maven을 먼저 설치해준다.

(필요 시 다른 DB로 대체 가능. 설치 후 환경변수 등록할 것.)

```
wget http://apache.mirror.cdnetworks.com/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz
tar -zxvf apache-maven-3.6.1-bin.tar.gz
```


02 Static Analyzer - SonarQube

- SonarQube 최신 버전을 다운로드 하여 압축을 해제하고,
bin directory 내부의 linux directory 중 해당 OS에 맞는 directory에 들어가
sonar.sh 파일을 실행시킨다.

Get the LTS (Long-term Support): SonarQube 6.7.x ⓘ

[See what's new](#) - [Documentation](#) - [Upgrade Guide](#) - [Requirements](#) - [Release notes](#)

Community Edition 6.7.7

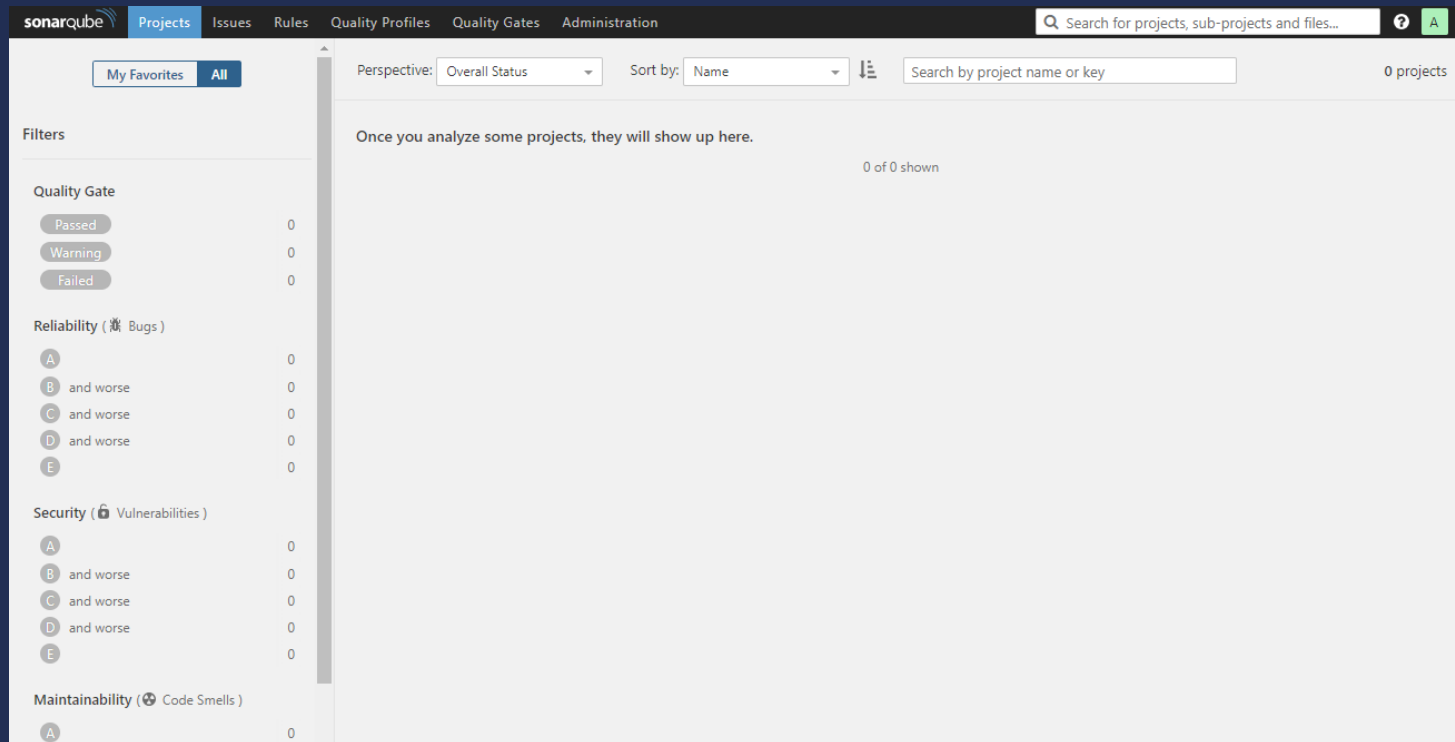
SonarSource Commercial Editions

With 6.7 LTS, [Commercial Editions](#) are available directly via the SonarQube Marketplace. Do settings (using your license key).

```
wget https://sonarsource.bintray.com/Distribution/sonarqube/sonarqube-6.7.7.zip
unzip sonarqube-6.7.7.zip
cd sonarqube-6.7.7/bin/linux-x86-64
./sonar.sh start
```

02 Static Analyzer - SonarQube

- 해당 SonarQube의 default port 번호는 9000이므로,
server DNS 뒤에 :9000 port 번호를 입력하여 SonarQube에 접속한다.



The screenshot displays the SonarQube web interface. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is located on the right. The main content area shows a list of projects, currently empty, with a message: 'Once you analyze some projects, they will show up here.' and '0 of 0 shown'. The left sidebar contains filters for 'Quality Gate' (Passed, Warning, Failed), 'Reliability (Bugs)', 'Security (Vulnerabilities)', and 'Maintainability (Code Smells)'. Each filter category shows a count of 0.

02 Static Analyzer - SonarQube

- Jenkins 와 연동하기 위해 SonarQube Scanner Plugin과 Sonar scanner를 설치한다.

플러그인 설치/업그레이드 중

준비

- Checking internet connectivity
- Checking update center connectivity
- Success

Maven Integration 🟡 Downloaded Successfully. Will be activated during the next boot

SonarQube Scanner 🟡 설치중

Jenkins 재시작 🟡 대기중

➡ [메인 페이지로 돌아가기](#)
(설치된 플러그인을 바로 사용할 수 있습니다.)

➡ 설치가 끝나고 실행중인 작업이 없으면 Jenkins 재시작.

```
wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-3.3.0.1492-linux.zip
unzip sonar-scanner-cli-3.3.0.1492-linux.zip
mv /home/JungMo/sonar-scanner-cli-3.3.0.1492-linux /opt/sonar-scanner
```

02 Static Analyzer - SonarQube

➤ SonarQube Token 생성 후

해당 Token과 SonarQube server 주소를 이용하여

Jenkins에 SonarQube server 등록한다.

The screenshot shows the SonarQube Administrator interface. At the top, there is a navigation bar with 'Profile', 'Security', 'Notifications', and 'Projects'. The 'Security' tab is selected. Below the navigation bar, there is a section titled 'Tokens'. It contains a paragraph of text explaining the purpose of User Tokens. Below the text is a table with two columns: 'Name' and 'Created'. The table has one row with the name 'jenkins' and the creation date 'May 2, 2019'. To the right of the 'Created' column, there is a red 'Revoke' button. At the bottom of the page, there is a 'Generate New Token' section with an input field for 'Enter Token Name' and a 'Generate' button.

The screenshot shows the SonarQube configuration page for adding a new installation. It features a checkbox labeled 'Enable injection of SonarQube server configuration as build environment variables', which is checked. Below this, there is a paragraph of text explaining the purpose of this feature. The page contains several form fields: 'Name' with the value 'Sonarqube', 'Server URL' with the value 'SonarQube 서버주소', and 'Server authentication token' with a masked input field. There is a 'Default is http://localhost:9000' label next to the 'Server authentication token' field. Below the form fields, there is a '고급...' button and a red 'Delete SonarQube' button. At the bottom of the page, there is an 'Add SonarQube' button and a link for 'List of SonarQube installations'.

02 Static Analyzer - SonarQube

➤ 설치한 Sonar Scanner를 Jenkins와 연동시켜 준다.

```
mv /opt/sonar-scanner /var/lib/jenkins/sonar-scanner-3.3.0.1492-linux
```

SonarQube Scanner

SonarQube Scanner installations

[Add SonarQube Scanner](#)

| | |
|--|--|
| ☰ SonarQube Scanner | |
| Name | <input type="text" value="Sonar scanner"/> |
| SONAR_RUNNER_HOME | <input type="text" value="/var/lib/jenkins/sonar-scanner-3.3.0.1492-linux"/> |
| <input type="checkbox"/> Install automatically | |

[Add SonarQube Scanner](#)

List of SonarQube Scanner installations on this system

02 Static Analyzer - SonarQube

- SonarQube와 연동하기 위한 Jenkins project로 이동하여 빌드 환경을 설정한다.

빌드 환경

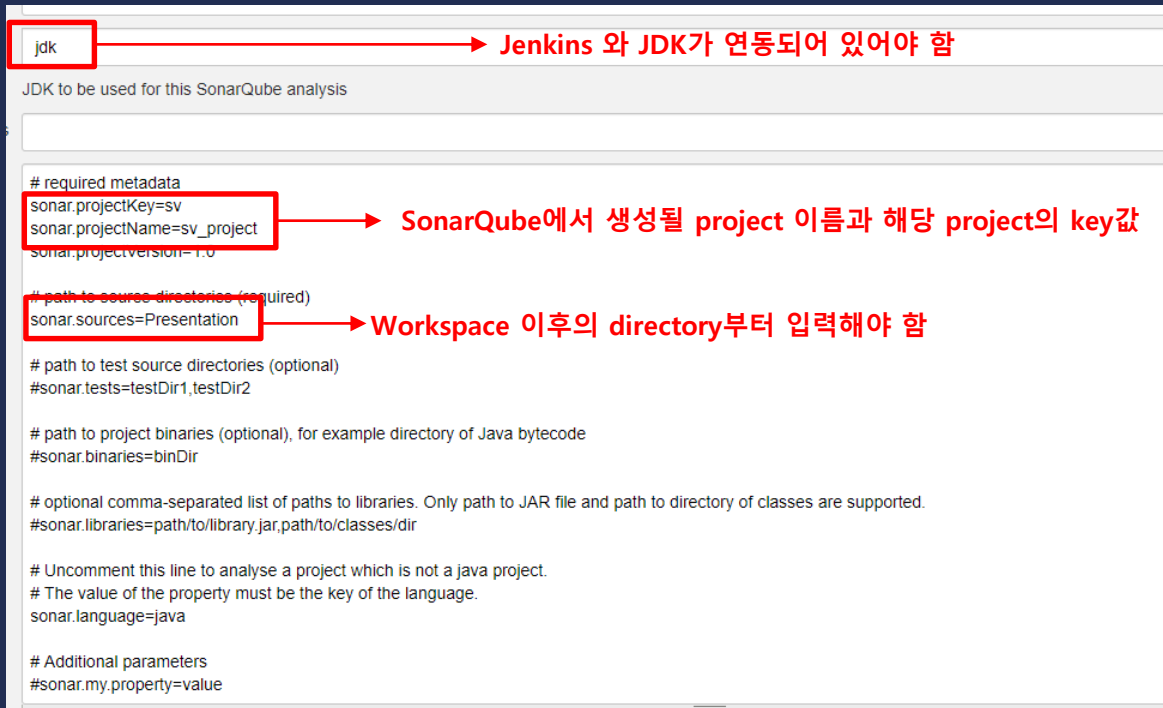
- Delete workspace before build starts
- Use secret text(s) or file(s)
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Inspect build log for published Gradle build scans
- Prepare SonarQube Scanner environment
- With Ant

Add build step ▾

- Execute SonarQube Scanner
- Execute Windows batch command
- Execute shell
- Generate Redmine Metrics Report
- GitHub PR: set 'pending' status
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit
- SonarScanner for MSBuild - Begin Analysis
- SonarScanner for MSBuild - End Analysis

02 Static Analyzer - SonarQube

- SonarQube와 연동하기 위한 Jenkins project로 이동하여 빌드 환경을 설정한다.



The screenshot shows the SonarQube configuration page in Jenkins. A red box highlights the 'jdk' field, with an arrow pointing to the text 'Jenkins 와 JDK가 연동되어 있어야 함'. Below this, the 'sonar.projectName=sv_project' line is highlighted with a red box and an arrow pointing to 'SonarQube에서 생성될 project 이름과 해당 project의 key값'. Another red box highlights the 'sonar.sources=Presentation' line, with an arrow pointing to 'Workspace 이후의 directory부터 입력해야 함'. The rest of the configuration text is visible but not highlighted.

```
jdk  
JDK to be used for this SonarQube analysis  
  
# required metadata  
sonar.projectKey=sv  
sonar.projectName=sv_project  
sonar.projectVersion=1.0  
  
# path to source directories (required)  
sonar.sources=Presentation  
  
# path to test source directories (optional)  
#sonar.tests=testDir1,testDir2  
  
# path to project binaries (optional), for example directory of Java bytecode  
#sonar.binaries=binDir  
  
# optional comma-separated list of paths to libraries. Only path to JAR file and path to directory of classes are supported.  
#sonar.libraries=path/to/library.jar,path/to/classes/dir  
  
# Uncomment this line to analyse a project which is not a java project.  
# The value of the property must be the key of the language.  
sonar.language=java  
  
# Additional parameters  
#sonar.my.property=value
```

02 Static Analyzer - SonarQube

➤ 실행

The screenshot shows the SonarQube Quality Gate page for the project 'sv_project'. The Quality Gate is in a 'Passed' state. Key metrics include 0 Bugs, 0 Vulnerabilities, 3h Code Smells (Debt), and 0.0% Coverage. A 'Build History' table is visible on the left, and a '고정링크' (Fixed Links) section provides links to various build logs.

| Build ID | Time | Status |
|----------|--------------------|---------|
| #122 | 2019. 5. 2 오후 9:14 | Success |
| #121 | 2019. 5. 2 오후 9:14 | Success |
| #120 | 2019. 5. 2 오후 9:10 | Failure |
| #119 | 2019. 5. 2 오후 9:08 | Failure |

The screenshot shows the SonarQube dashboard for the project 'sv_project'. The Quality Gate is 'Passed'. Metrics include 0 Bugs, 0 Vulnerabilities, 3h Code Smells (Debt), and 0.0% Coverage. The dashboard also shows 'New Bugs' and 'New Vulnerabilities' as 0, and 'New Debt' and 'New Code Smells' as 0. The 'Coverage' section shows 0.0% coverage on new code.

| | | | | | |
|--------------------------|-----|----|----|----|---|
| <input type="checkbox"/> | 293 | 결함 | 신규 | 보통 | software_verification_project / 125 / SUCCESS |
| <input type="checkbox"/> | 292 | 결함 | 신규 | 보통 | software_verification_project / 124 / SUCCESS |
| <input type="checkbox"/> | 291 | 결함 | 신규 | 보통 | software_verification_project / 123 / FAILURE |
| <input type="checkbox"/> | 290 | 결함 | 신규 | 보통 | software_verification_project / 122 |
| <input type="checkbox"/> | 289 | 결함 | 신규 | 보통 | software_verification_project / 121 |
| <input type="checkbox"/> | 288 | 결함 | 신규 | 보통 | software_verification_project / 120 |

03 Static Analyzer - PMD



PMD(Program May Dependable)

- 미사용 변수, 비어있는 코드 block, 불필요한 object 생성과 같이 defect을 유발할 수 있는 코드를 검사한다.
- 미리 정의한 ruleset을 기반으로 구문을 분석한다.
- Java에서 많이 사용하지만, JavaScript, XML 등 다른 언어도 지원한다.

03 Static Analyzer - PMD

- 사용할 ruleset을 선택하여 pmd-rule-set.xml 생성

(아래는 생성한 xml파일의 일부)

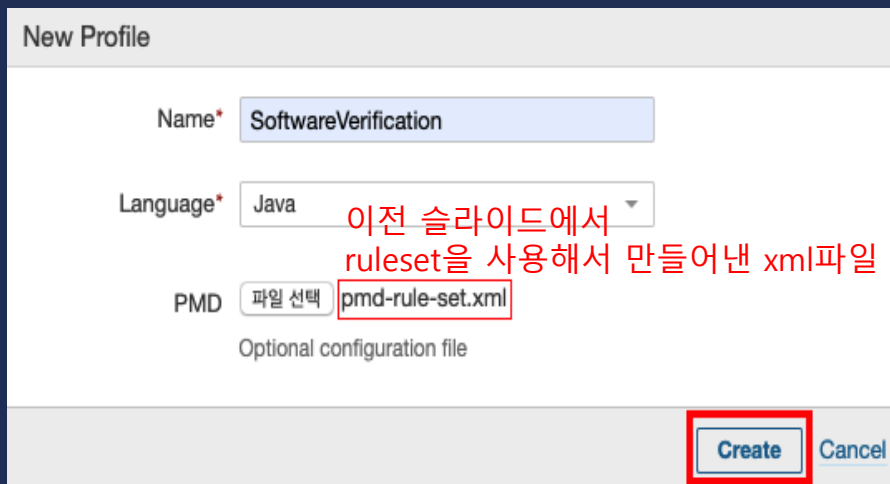
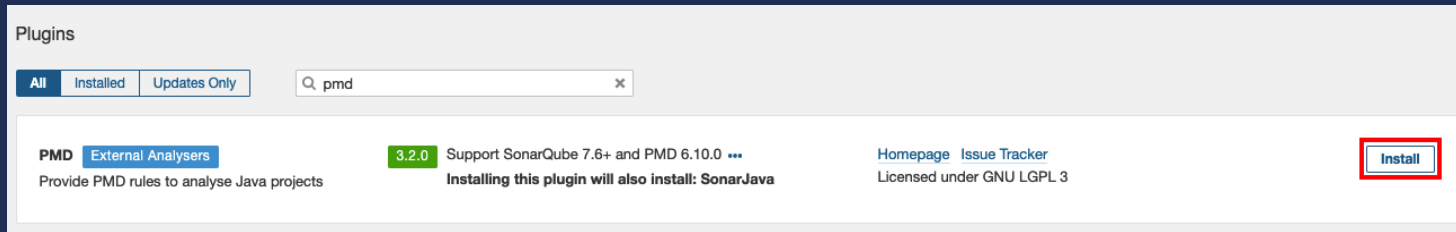
```
<rule ref="category/java/bestpractices.xml/AvoidStringBufferField" />
<rule ref="category/java/bestpractices.xml/AvoidUsingHardCodedIP" />
<rule ref="category/java/bestpractices.xml/CheckResultSet" />
<rule ref="category/java/bestpractices.xml/ConstantsInInterface" />
<rule ref="category/java/bestpractices.xml/DefaultLabelNotLastInSwitchStmt" />
<rule ref="category/java/bestpractices.xml/ForLoopCanBeForeach" />
<rule ref="category/java/bestpractices.xml/GuardLogStatement" />
<rule ref="category/java/codestyle.xml/FormalParameterNamingConventions" />
<rule ref="category/java/codestyle.xml/ClassNameingConventions" />
<rule ref="category/java/codestyle.xml/LocalVariableNamingConventions" />
<rule ref="category/java/codestyle.xml/MethodNamingConventions" />
<rule ref="category/java/codestyle.xml/PackageCase" />
<rule ref="category/java/design.xml/SimplifyBooleanReturns" />
<rule ref="category/java/design.xml/SimplifyConditional" />
<rule ref="category/java/design.xml/SingularField" />
```

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

> PMD ruleset reference

03 Static Analyzer - PMD

- SonarQube의 Market Place에서 PMD plugin을 검색하여 추가하고, Quality Profiles tab 에서 Create를 통해 ruleset을 모은 New Profile을 만든다.



03 Static Analyzer - PMD

➤ 다음과 같이 Quality Profile 항목에

PMD ruleset을 활용한 SoftwareVerification Profile이 import된 것을 볼 수 있다.

| Tag | Repository | Default Severity | Status | Available Since | Template | Quality Profile |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | | | | | | Drupal PHP (Built-in) |
| | | | | | | FOR-2 PHP (Built-in) |
| | | | | | | SoftwareVerification Java active |
| | | | | | | Sonar way C# (Built-in) |

| Rule Name | Severity | Language | Category | Sub-category | Action |
|---|------------|----------|------------|----------------|------------|
| Avoid Catching Throwable | Deprecated | Java | Code Smell | error-handling | Deactivate |
| Avoid Constants Interface | Deprecated | Java | Code Smell | | Deactivate |
| Avoid Decimal Literals In Big Decimal Constructor | Deprecated | Java | Code Smell | | Deactivate |
| Avoid Instanceof Checks In Catch Clause | Deprecated | Java | Code Smell | | Deactivate |
| Avoid Multiple Unary Operators | Deprecated | Java | Code Smell | | Deactivate |
| Avoid Protected Field In Final Class | Deprecated | Java | Code Smell | | Deactivate |
| Avoid Protected Method In Final Class Not Extending | Deprecated | Java | Code Smell | | Deactivate |
| Avoid StringBuffer field | Deprecated | Java | Code Smell | | Deactivate |

04 Static Analyzer – Checkstyle



- 소스코드 내에서 다양한 위반사항에 대해 알 수 있다.
- 개발자들이 체크인 전에 위반사항을 고칠 수 있다.
- 미리 정해 놓은 코딩 규칙을 팀원들이 보다 쉽게 적용할 수 있게 도와주는 도구
→ error를 조기에 찾을 수 있음.

ex) 사용하지 않은 pointer는 Null로 초기화 한다.

04 Static Analyzer – Checkstyle

➤ PMD와 마찬가지로 Market Place에서 Checkstyle Plugin을 찾아 설치한다.

설치 시 Rules Repository에 Checkstyle Java 항목이 추가된다.

The screenshot shows the SonarQube Marketplace interface for the Checkstyle plugin. At the top, there are tabs for 'All' and 'Updates Only', and a search bar containing 'checkstyle'. Below this, the plugin name 'Checkstyle' is displayed with a sub-label 'External Analysers' and a note 'Analyze Java code with Checkstyle'. To the right, it indicates '4.19 installed' and provides links for 'Issue Tracker', 'Licensed under LGPL-3.0', and 'Developed by Checkstyle'. An 'Uninstall' button is visible in the top right corner.

This screenshot shows the 'Repository' section of the SonarQube interface before the Checkstyle plugin is installed. A list of various analyzers is shown with their respective counts. The 'Checkstyle Java' entry is not present in this list.

| Repository | Count |
|--------------------------|-------|
| SonarAnalyzer Java | 529 |
| FindBugs Java | 449 |
| FindBugs Contrib Java | 302 |
| SonarAnalyzer C# | 301 |
| PMD Java | 268 |
| SonarAnalyzer JavaScript | 183 |
| Pylint Python | 180 |
| SonarAnalyzer PHP | 121 |
| Find Security Bugs Java | 113 |
| SonarQube Flex | 73 |



This screenshot shows the 'Repository' section of the SonarQube interface after the Checkstyle plugin has been installed. The 'Checkstyle Java' entry is now present in the list and is highlighted with a red box.

| Repository | Count |
|--------------------------|-------|
| SonarAnalyzer Java | 529 |
| FindBugs Java | 449 |
| FindBugs Contrib Java | 302 |
| SonarAnalyzer C# | 301 |
| PMD Java | 268 |
| SonarAnalyzer JavaScript | 183 |
| Pylint Python | 180 |
| Checkstyle Java | 157 |
| SonarAnalyzer PHP | 121 |
| Find Security Bugs Java | 113 |

05 Static Analyzer – FindBugs

- Java program에서 잠재적으로 발생 가능한 버그를 찾는 것이 주 목적인 정적 분석 tool.
- Compile된 java byte code를 읽어서 검사
 - 속도가 빠름, 하지만 build 과정이 필수
(SonarQube에서는 해당 option 선택 가능)
- 실제 결함/버그를 잘 찾아주는 편이다.

(static analysis에서 발생가능한 false alarm이 존재할 수 있음)



05 Static Analyzer – FindBugs

➤ 역시 FindBugs도 Market Place에서 FindBugs Plugin을 찾아 설치한다.

설치 시 Rules Repository에 각종 FindBugs Java 항목이 추가된다.

Findbugs **External Analyzers** 3.9.3 installed [Homepage](#) [Issue Tracker](#) [Uninstall](#)
Analyze Java, Scala, Closure and JSP code with SpotBugs. 3.1.11
Updates: **3.11.0** Use SpotBugs 3.1.12 ...
Licensed under GNU LGPL 3
Developed by [SpotBugs Team](#)

| Repository | |
|---------------------|-----|
| SonarAnalyzer Java | 529 |
| PMD Java | 268 |
| Checkstyle Java | 157 |
| PMD Unit Tests Java | 17 |
| Common Java Java | 6 |



| Repository | |
|--------------------------------|-----|
| SonarAnalyzer Java | 529 |
| FindBugs Java | 449 |
| FindBugs Contrib Java | 302 |
| PMD Java | 268 |
| Checkstyle Java | 157 |
| Find Security Bugs Java | 113 |
| PMD Unit Tests Java | 17 |
| Common Java Java | 6 |

05 Static Analyzer – FindBugs

➤ Quality Profile 쪽에도 Rules들이 추가된다.

| Java, 6 profile(s) | Projects | Rules | Updated | Used |
|---|----------|--------|--------------|-------------|
| FindBugs <small>Built-in</small> | Default | 0 | Never | 1 hour ago |
| FindBugs + FB-Contrib <small>Built-in</small> | 0 | 0 | Never | Never |
| FindBugs Security Audit <small>Built-in</small> | 0 | 0 | Never | Never |
| FindBugs Security Minimal <small>Built-in</small> | 0 | 0 | Never | Never |
| SoftwareVerification | 0 | 98 131 | 19 hours ago | 2 hours ago |
| Sonar way <small>Built-in</small> | 0 | 349 | Never | 1 hour ago |

| Java, 6 profile(s) | Projects | Rules | Updated | Used |
|---|----------|--------|--------------|-------------|
| FindBugs <small>Built-in</small> | Default | 443 | Never | 1 hour ago |
| FindBugs + FB-Contrib <small>Built-in</small> | 0 | 745 | Never | Never |
| FindBugs Security Audit <small>Built-in</small> | 0 | 121 | Never | Never |
| FindBugs Security Minimal <small>Built-in</small> | 0 | 91 | Never | Never |
| SoftwareVerification | 0 | 98 131 | 19 hours ago | 2 hours ago |
| Sonar way <small>Built-in</small> | 0 | 349 | Never | 1 hour ago |

06 Summary

- Quality Profile 탭 우측 상단의 Create를 눌러 Profile을 만든다.

New Profile

Name*

Language*

PMD 선택된 파일 없음
Optional configuration file

FindBugs 선택된 파일 없음
Optional configuration file

Checkstyle 선택된 파일 없음
Optional configuration file

06 Summary

- 생성한 Profile을 선택한 후 Change Project로 SonarQube project를 할당할 수 있고, 할당 이후에는 Activate More 을 클릭하여 각종 rule을 마음대로 추가할 수 있다.

Java, 7 profile(s)

- FindBugs Built-in
- FindBugs + FB-Contrib Built-in
- FindBugs Security Audit Built-in
- FindBugs Security Minimal Built-in
- SoftwareVerification
- Sonar way Built-in
- custom**

Projects

With Without All Search

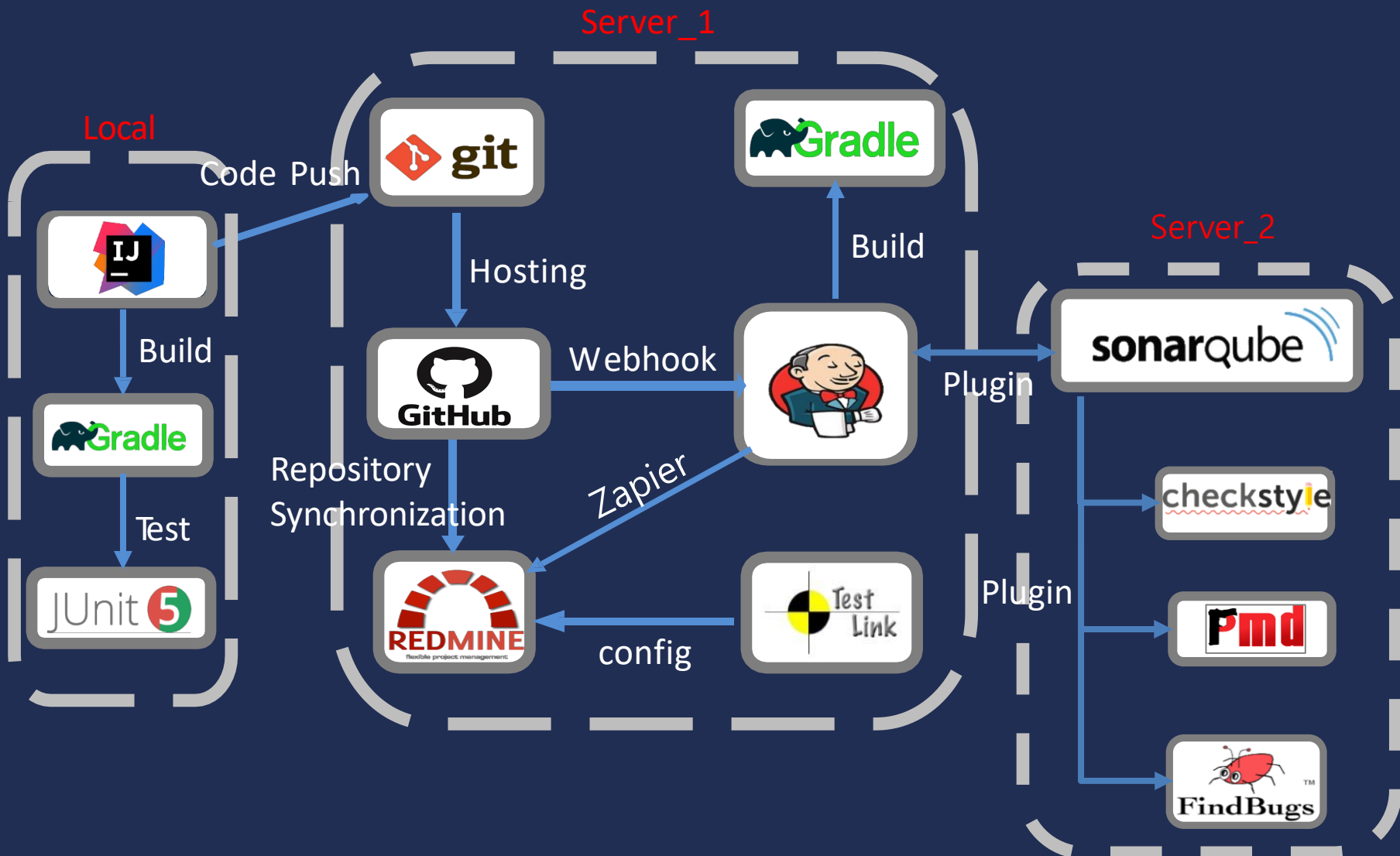
- sv_project

| Rules | Active | Inactive |
|-------------------|--------|----------|
| Total | 0 | 1.8k |
| 🐛 Bugs | 0 | 586 |
| 🔒 Vulnerabilities | 0 | 200 |
| 💩 Code Smells | 0 | 1.1k |

Activate More

Sonar way rules not included ⓘ 349

06 Summary



감사합니다

THANK YOU

Software Verification Team 4

강 송 신 정 상 승 모 연 화