

코드 품질 향상을 위한 SonarQube 활용

시네틱스 대표: 한동준

handongjoon@gmail.com

dongjoon.han@synetics.kr

교육 개요

□ 목표

- 정적분석이 무엇이고, 왜 사용하는지 이해한다.
- 정적분석 도구인 SonarQube를 설치하고, 기본 설정 후 예제 프로젝트를 분석할 수 있다.
- SonarQube 분석 결과를 확인하고 개선점을 식별할 수 있다.
- SonarQube를 지속적 통합 도구인 Jenkins와 연동할 수 있다.

□ 강의 순서

- 정적분석 개요
- SonarQube 소개
- SonarQube 설치
- 기본 설정
- 정적 분석 룰 설정
- 분석 실행
- 주요 IDE 별 sonarlint 설치 방법
- SonarQube와 Jenkins 연동

01 정적분석 개요



소프트웨어 품질(Software Quality)

□ 정의

- 소프트웨어가 지닌 바람직한 속성의 정도 [IEEE]
- 요구되는 기능을 발휘할 수 있는 소프트웨어 특성의 정도 [DoD]
- 소프트웨어가 기능, 성능 및 만족도에 있어서 명시된 요구사항 및 내재된 요구사항을 얼마나 충족하는 가를 나타내는 소프트웨어 특성의 총체 [Pressman]

□ 소프트웨어 품질의 구분

- 제품(Product) 품질
 - 제품 자체가 가지는 품질
 - 완성된 소프트웨어가 운용될 환경에 올려져 최종 시스템이 완성되었을 때, 소비자가 요구하는 바에 얼마나 부합되는지를 나타내는 품질
- 프로세스(Process) 품질
 - 소프트웨어를 개발하기 위해 필요한 모든 개발 활동이 계획을 준수하여 개발하였는가를 나타내는 품질
 - 그 활동이 효과적인지에 대해 검토 (Review) 및 감사 (Audit) 활동을 통해 확인

소프트웨어 제품 품질과 V&V

□ Verification

- 제품이 올바르게 생성되고 있는가?(Are we building the product right?)

『Boehm』

- 소프트웨어가 정확한 요구사항에 부합하여 구현되었음을 보장하는 활동
- '요구사항 명세서에 맞게 올바른 방법으로 제품을 만들고 있음'을 보장

□ Validation

- 올바른 제품을 생성하고 있는가?(Are we building the right product?)

『Boehm』

- 소프트웨어가 고객이 의도한 요구사항에 따라 구현되었음을 보장하는 활동
- '고객이 의도한 환경이나 사용 목적에 맞게 올바른 제품을 만들고 있음'을 보장

Verification & Validation(테스트) 종류

● 정적(Static)인 방법

- 소프트웨어를 실행하지 않고 결함을 찾아내는 것
- 여러 참여자들이 모여 소프트웨어를 검토하여 결함을 찾아내거나 정적 검증 도구 이용
- 소프트웨어 개발 중에 생성되는 모든 산출물들에 대해서 적용 가능
- 대표적인 방법:
 - 동료검토(Peer Review)
 - 인스펙션(Inspection)
 - 워크스루(Walk-through)
 - 데스크체크(Desk Check)
 - 도구를 이용한 정적 분석
 - 룰 기반 정적 분석(PMD, BugFind, SonarQube 등)
 - 의존성 분석(Jdepend, Doxygen, Lattix 등)
 - 품질 매트릭 분석

● 동적(Dynamic)인 방법

- 소프트웨어를 실행하여 결함을 찾아냄
- 발견된 결함은 디버깅 활동으로 확인하여 수정함
- 대표적인 방법
 - 테스트
 - 블랙박스 테스트
 - 화이트박스 테스트

정적 분석 개요

□ 정의

- 실제 실행 없이 컴퓨터 소프트웨어, 특히 소스코드를 분석하는 것 (위키피디아)

□ (개발자를 위한) 사용 시기

- 컴파일 전에 소스코드 품질 검토
 - 소스코드 품질이란? 복잡도, 라인수, 복사-붙여넣기 등

□ 사용 목적

- 소스코드의 잠재적인 품질 문제(낮은 품질) 발견
 - 네트워크 자원 누수, 높은 복잡도, 추천하지 않는 패턴 등
 - 네트워크 자원 누수 예) 네트워크 소켓의 open()는 있으나, 명시적인 close()가 없는 경우
- 결함의 조기 발견
 - 메모리 누수, 버퍼 오버플로우 등
- 소스코드 표준 준수 확인
 - 코딩 컨벤션(명명규칙), 보안

```
// 임시 저장값 불러오기
SharedPreferences prefs = getSharedPreferences("temp", MODE_...
String notice = prefs.getString("notice_temp", "");
String[] weeks = { "일", "월", "화", "수", "목", "금", "토" };
// 날짜 불러오기
Calendar m = Calendar.getInstance();

year = m.get(Calendar.YEAR);
month = m.get(Calendar.MONTH) + 1;
day = m.get(Calendar.DATE);

// 카드 인터페이스를 불러온다.
// init CardView
mCardView = (CardView) findViewById(R.id.cardsview);
mCardView.setSwipeable(true);
// add AndroidViews Cards

// 공지사항을 불러온다.
if (notice.matches("")) {
    // else f
```

소스코드



정적 분석 도구

File	Line	Priority	Type	Category
AddSubject.java:5	5	Normal	UnusedImports	Import Statements
AddSubject.java:163	163	Normal	EmptyStatementNotInLoop	Empty Code
AddTodo.java:4	4	Normal	UnusedImports	Import Statements
AddTodo.java:13	13	Normal	UnusedImports	Import Statements
Avoid unused imports such as "java.awt.FlowLayout".				
Item.java:12	12	Normal	UnusedLocalVariable	Unused Code
ChangeSubject.java:5	5	Normal	UnusedImports	Import Statements
ChangeSubject.java:195	195	Normal	EmptyStatementNotInLoop	Empty Code

분석 결과

[예제] MISRA 룰의 Good/Bad Case

MISRA_15_03 [Contents]

switch 문의 마지막 절이 default 절이어야 함

마지막 절에 default가 오는 것은 방어적 프로그래밍을 위함이다. default 절에서는 적절한 행동을 취하거나 아무 행동을 하지 않으면 적절한 주석을 달아야 한다.

[0090] switch문에 default 가 존재 검사

BAD

```
switch ( defaults )
{
  case 3U:
  {
    use_uint16 ( defaults );
    break;
  }
  case 4U:
  {
    use_uint16 ( defaults );
    break;
  }
  /* Not compliant. No default clause. */
}
```

GOOD

```
switch ( defaults )
{
  case 3U:
  {
    use_uint16 ( defaults );
    break;
  }
  case 4U:
  {
    use_uint16 ( defaults );
    break;
  }
  default:
  {
    use_uint16 ( defaults );
    break;
  }
  /* Compliant */
}
```


SW 품질 확보를 위한

가성비

더 적은 노력과
일찍 수정해야 적은 비용

소프트웨어 결함은

구현 단계에서 가장 많이 유입

**소프트웨어 개발은
노동집약적 산업**

제품 품질 메트릭

□ 규모 관련

- 라인 수(LOC: Line of Code)
- 주석 제외 라인 수
- 주석 비율
- 함수 별 라인 수

□ 복잡도 관련

- 순환 복잡도(Cyclomatic Complexity): 함수의 제어 흐름이 얼마나 복잡한지 측정. 분기문 +1
- 인지 복잡도(Cognitive Complexity): 함수가 얼마나 중첩문이 많은지 측정. Nesting Depth 라고도 표현

□ 테스트 관련

- 기능 커버리지
- 함수 커버리지
- 문장/분기/MCDC 커버리지

□ 룰 기반 정적 분석 관련

- 정적 분석 룰 위반 수

제품 품질 메트릭

□ 결함 관련

- 소스코드 라인 대비 결함 비율
- 유형 별 결함 비율
- 원인 별 결함 비율

□ 의존성 관련

- 함수 별 호출하는 건수
- 함수 별 호출되는 건수
- 변경 영향 비율(Stability): 한 함수가 변경되면 코드의 몇 %가 영향받는가?
- 상호 참조 비율: 상호 호출하는 파일이 있는가? (변경에 대한 영향도 파악 목적)

□ 출시 관련

- 출시 후 결함 수

룰 기반(Rule based) 정적 분석 소개

□ 개요

- 사전에 정해진 룰 가이드라인을 소스코드가 만족하는지 분석하는 도구
 - 룰 가이드라인은 검증 도구가 해석할 수 있는 룰과 이에 대한 가이드로 구성되며, 룰셋은 룰을 카테고리화 한 것
 - 예제 참고

□ 대표 룰 가이드라인(코딩 표준)

- MISRA: Motor Industry Software Reliability Association에서 개발한 안전성, 호환성, 신뢰성을 위한 C, C++ 룰
- CERT: SEI CERT(Computer Emergency Response Team)의 SW 개발보안을 위한 C, C++, Java 룰
- 행정자치부 개발보안(시큐어 코딩): 행정기관의 SW 개발보안을 위한 JAVA, C, Android-Java 룰
- 기타 각 도구/언어/도메인/회사에 따른 룰 가이드라인이 존재
 - 예) 구 Sun 룰 / SonarQube 룰 / PMD 룰 / 방사청 룰

PMD 를 적용 사례

o PMD 제공 전체 룰								
#	Rule Set	Rule 명	선정후보 (QA)	선정후보 (개발자)	룰반영	Sonar 반영	한글 Rule 설명	
48	Design (java)	UnnecessaryLocalBeforeReturn	O		O	O	불필요한 지역 변수 생성은 피해야 함	Avoid the creation of
50	Design (java)	UncommentedEmptyMethod	O		O	O	주석 없는 빈 메소드는 주석 필요	Uncommented Empty
51	Design (java)	UncommentedEmptyConstructor	O		O	O	주석 없는 빈 생성자는 주석 필요	Uncommented Empty
75	Basic (java)	ForLoopShouldBeWhileLoop		O	O	O	for를 while로 간략화 할 수 있는 경우,	Some for loops can
78	Basic (java)	ReturnFromFinallyBlock	O		O	O	finally 블록에서 반환 하는것은 피해야	Avoid returning from
79	Basic (java)	UnconditionalIfStatement	O		O	O	항상 true 이거나 false인 조건에서는 if	Do not use "if" state
80	Basic (java)	BooleanInstantiation		O	O	O	Boolean 객체는 인스턴스화를 지양함	Avoid instantiating B
88	Basic (java)	AvoidUsingOctalValues	O		O	O	integer는 0으로 시작하면 안됨. 8진수	Integer literals shou
89	Basic (java)	AvoidUsingHardCodedIP	O		O	O	IP주소를 코드에 하드코딩 하면 안됨	Application with har
101	Strict Exceptions (java)	AvoidThrowingRawExceptionType	O		O	O	가공되지 않은 Exception을 throw하는	Avoid throwing cert
102	Strict Exceptions (java)	AvoidThrowingNullPointerException	O		O	O	NullPointerException을 throw하는 것	Avoid throwing Null
113	Android (java)	DoNotHardCodeSDCard		O	O		"/sdcard"를 사용하는 대신 Environme	Use Environment.ge
116	Java Logging (java)	SystemPrintln	O		O	O	System.out.println은 보통 디버그 목적	References to Syste
120	Controversial (java)	OnlyOneReturn		O	O	O	메소드는 1개의 return만을 사용해야 함	A method should ha
121	Controversial (java)	AssignmentInOperand	O		O	O	피연산자내에 할당문이 사용됨. 해당	Avoid assignments i
138	Controversial (java)	AvoidLiteralsInIfCondition		O	O	O	조건문에서 하드 코딩된 literal의 사용	Avoid using hard-co
144	Type Resolution (java)	LooseCoupling		O	O	O	implementation 타입(예. HashSet) 사	Avoid using implem
148	Empty Code (java)	EmptyCatchBlock	O		O	O	빈 catch 블록은 피해야 함	Empty Catch Block f
149	Empty Code (java)	EmptyIfStmt	O		O	O	빈 if 문장은 피해야 함	Empty If Statement f
150	Empty Code (java)	EmptyWhileStmt	O		O	O	빈 while 문장은 피해야 함	Empty While Stateme
151	Empty Code (java)	EmptyTryBlock	O		O	O	빈 try 블록은 피해야 함	Avoid empty try blo
152	Empty Code (java)	EmptyFinallyBlock	O		O	O	빈 finally 블록은 피해야 함	Empty finally blocks
153	Empty Code (java)	EmptySwitchStatements	O		O	O	빈 switch 문장은 피해야 함	Empty switch statem
154	Empty Code (java)	EmptySynchronizedBlock		O	O	O	빈 Synchronized 블록은 피해야 함	Empty synchronized
155	Empty Code (java)	EmptyStatementNotInLoop	O		O	O	for나 while 내의 빈 문장(또는 ;만 있는	An empty statement
156	Empty Code (java)	EmptyInitializer		O	O		빈 초기화는 피해야 함	Empty initializers ser
158	Empty Code (java)	EmptyStaticInitializer		O	O	O	빈 static 초기화는 피해야 함	An empty static initi
159	String and StringBuffer (java)	AvoidDuplicateLiterals		O	O	O	반복되는 String literal은 constant로	Code containing du

룰 기반 정적 분석 도구

□ 대표 도구

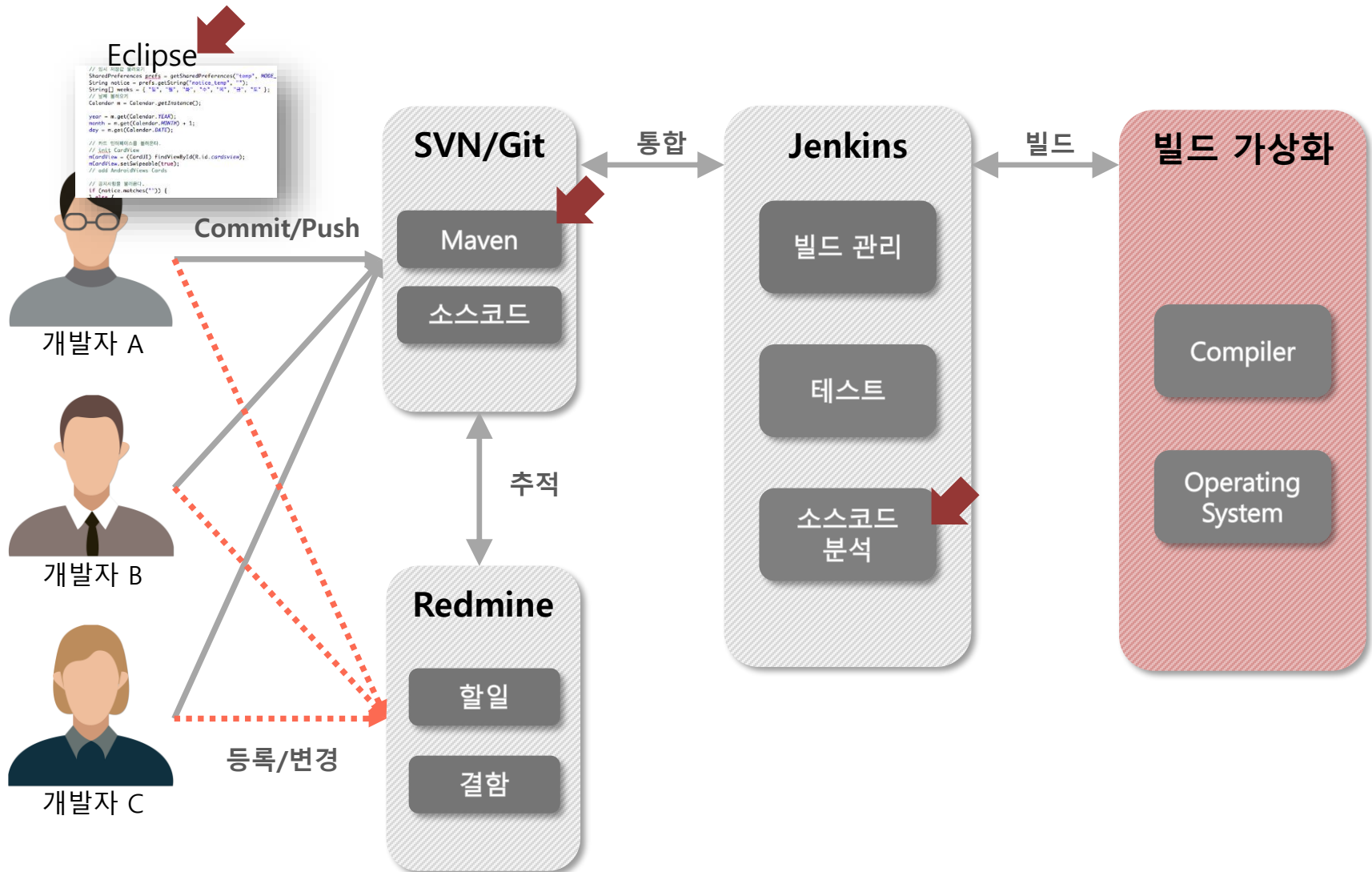
언어	국산	외산	오픈소스	특징
C/C++	Sparrow(파수) CodeInspector(Suresoft) Resort(Soft4Soft)	QAC/QAF(PRQA) Coverity(Synopsys) Polyspace-Bug Finder(MathWorks) Klockwork(Rogue Wave)	CppCheck	<ul style="list-style-type: none"> 국산 도구는 행자부 개발 보안 검증 ISO 26262 인증 (오픈소스 제외)
Java	Sparrow(파수) CodeInspector(Suresoft) Resort(Soft4Soft)	QAC/QAF(PRQA) Coverity(Synopsys)	PMD FindBug	<ul style="list-style-type: none"> 국산 도구는 행자부 개발 보안 검증 보안의 경우 제외하고, 오픈소스 위주 사용



□ 도구 선택 시 확인 사항

- 분석 대상 언어(C/C++ or JAVA)
- 1종, 2종 오류 발생을
 - 1종: 결함인데 못잡는 것
 - 2종: 결함이 아닌데 결함이라 하는 것

[참고] ALM에서 정적 분석 도구 역할



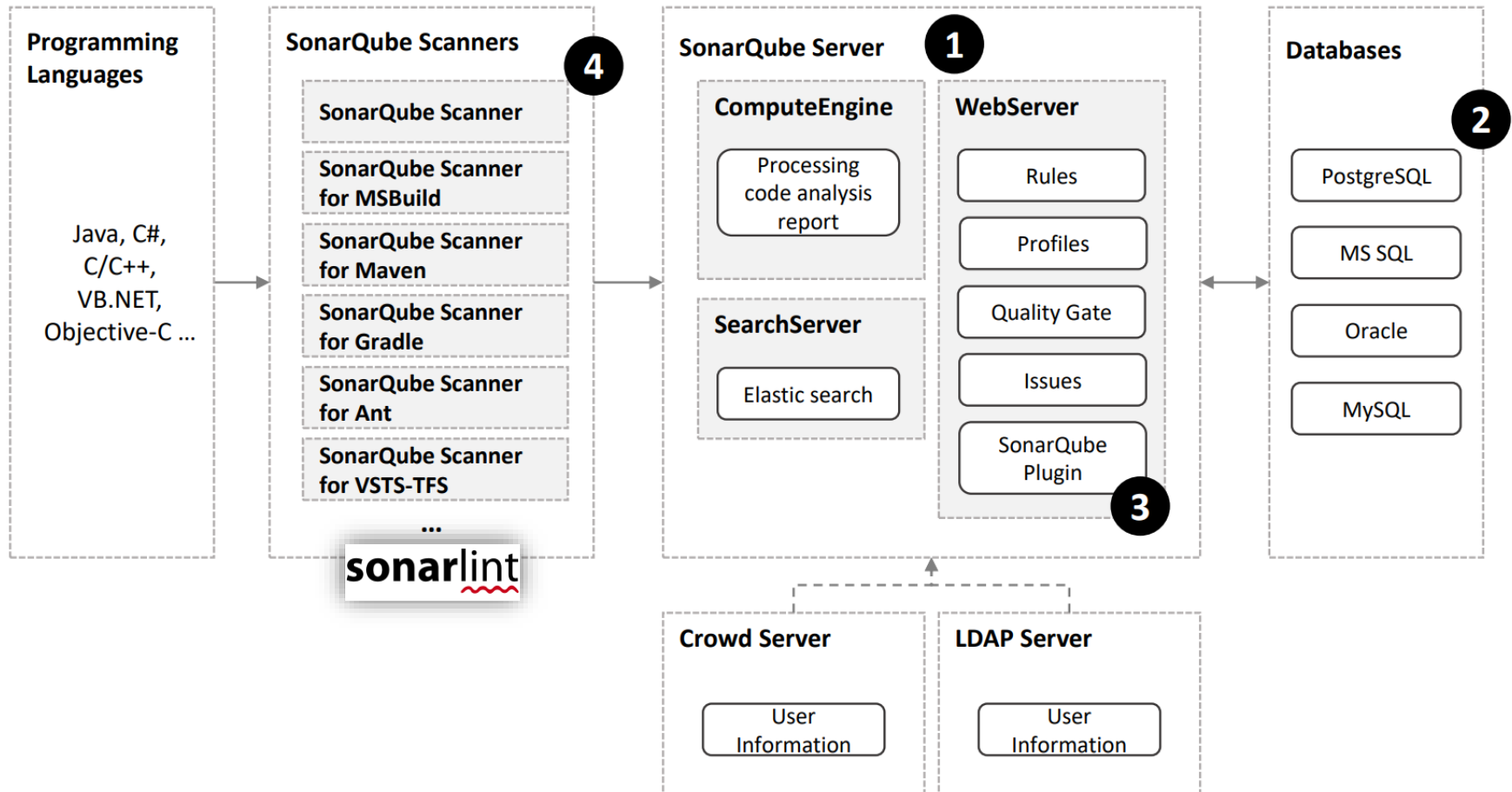
02 SonarQube 소개



SonarQube 소개

□ 개요

- 27개 이상 언어의 버그, 취약점, 코드악취 발견을 위한 정적 분석 및 웹을 통한 결과 게시를 지원
- Server, DB, Scanner, Plugin의 4개 서브 시스템으로 구성



Sonarlint(Sonarscanner IDE Plugin)

□ 개요

- IDE에서 플러그인 형식으로 SonarQube의 정적 분석을 실행할 수 있는 Plugin

*참고: 개발자가 IDE에서 정적 분석을 실행하고 결과를 조치하는 것이 가장 효율적임

□ 공식 지원 IDE

- Eclipse
- IntelliJ IDEA 계열
 - IntelliJ
 - PyCham
 - PHPStorm
 - Android Studio
- Visual Studio
 - 2017~2022
- VS Code

□ 설치방법

- Plugin 설치 메뉴에서 "sonarlint" 검색

SonarQube 룰셋

SonarQube는 Java, Object C, Python을 포함한 29개 이상의 개발 언어를 지원하며,
버그, 코드악취, 취약점으로 코딩 룰을 분류함



BUGS

버그: 실행 중 기대하지 않은 행동을 할 수 있는 코드나 잠재 버그



CODE SMELLS

코드악취: 유지보수성을 저하시키는 기술 부채를 의미



VULNERABILITIES

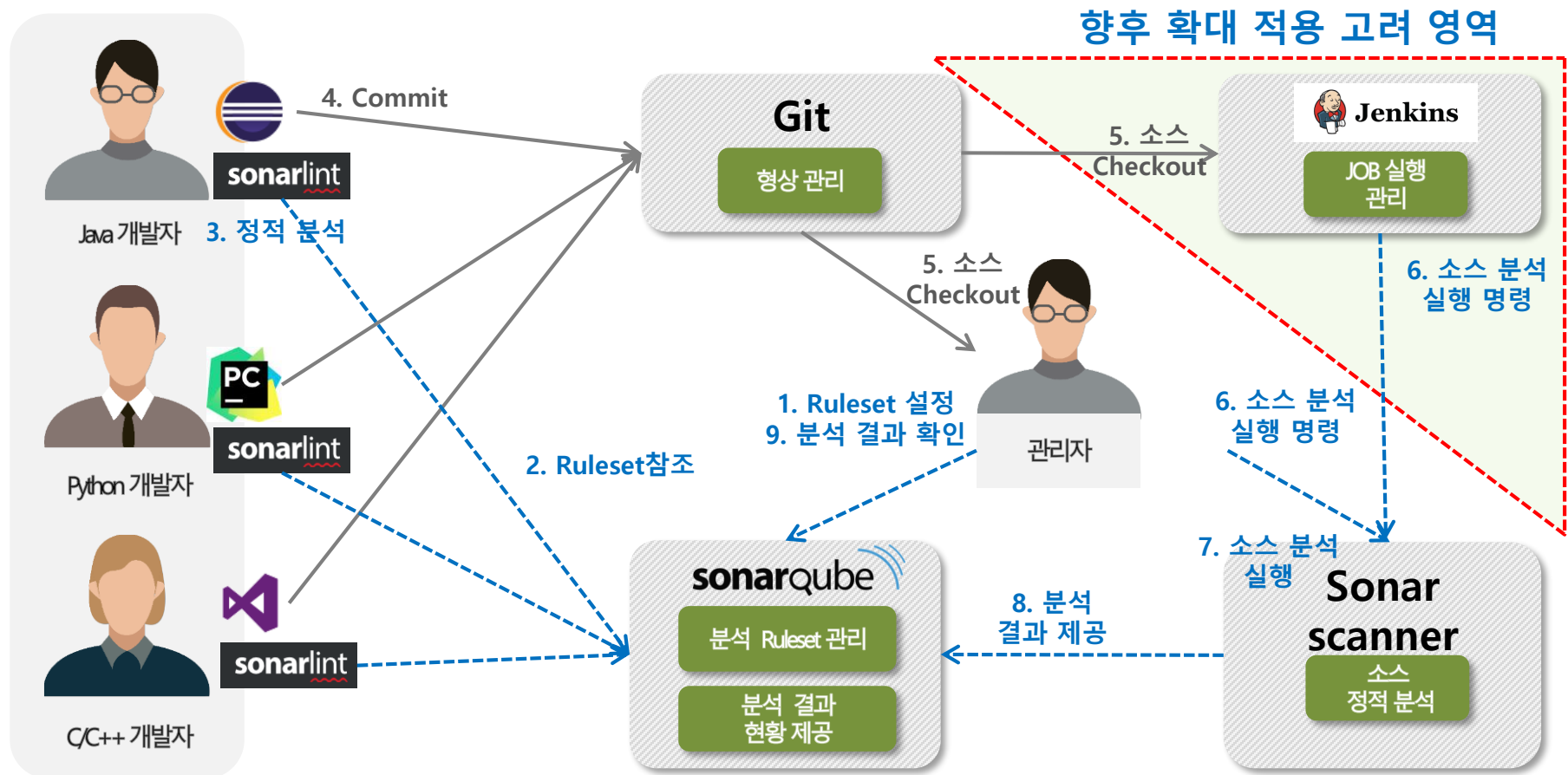
취약점: 설계와 다른 방향으로 프로그램을 실행시킬 수 있는 약점

구분	Java	C/C++	C#	Python	Objective C
버그	118	73	74	13	47
코드악취	332	253	256	40	177
취약점	48	2	21	1	2
Security Hotspot	30	0	22	0	0
총합	528	328	373	54	226

[룰셋 구분]

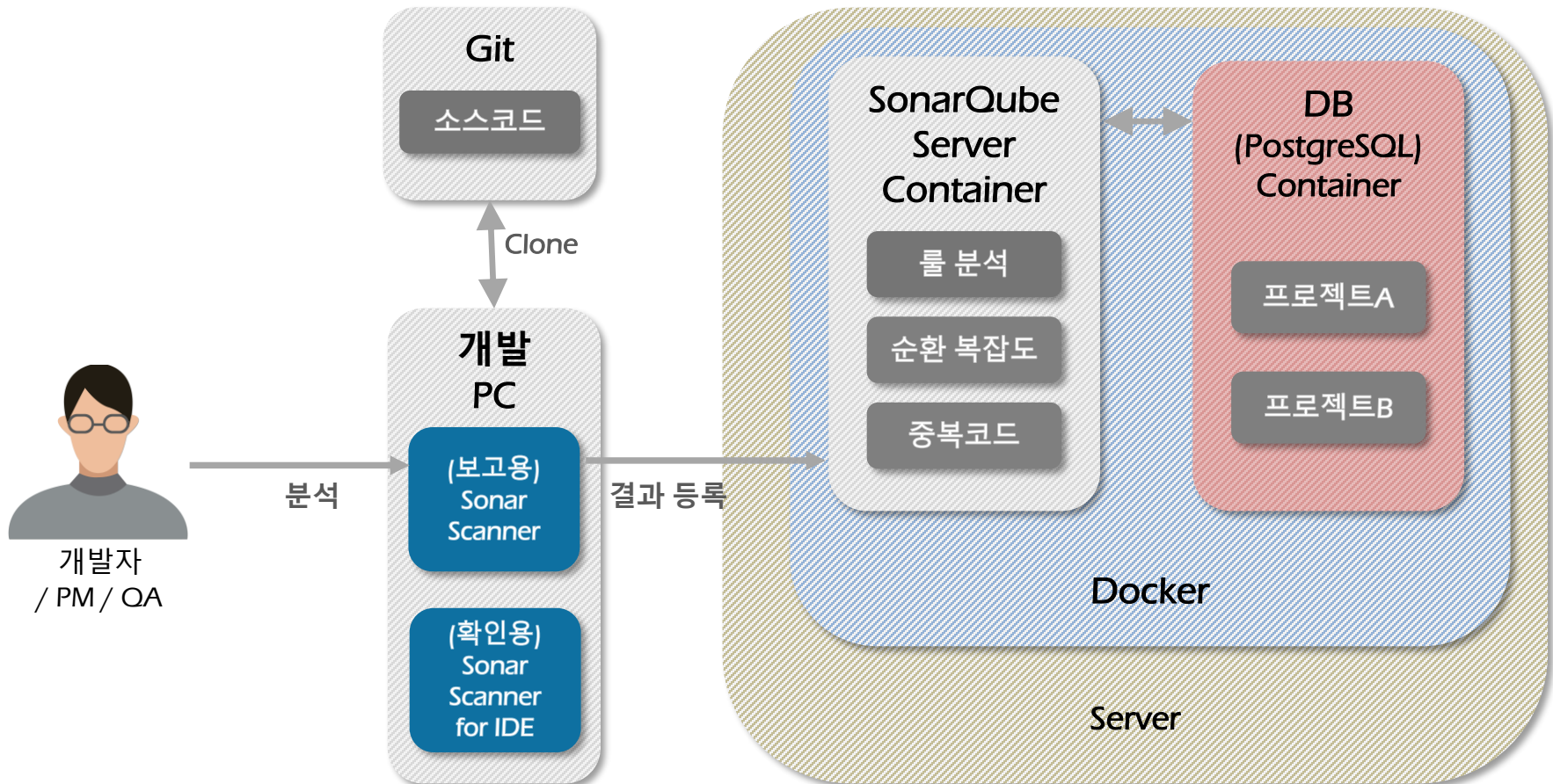
SonarQube 동작 구성

개발자는 IDE 플러그인, 관리자는 Sonar scanner를 사용하여 소스 정적 분석 수행
개발자는 IDE에서 바로 오류 조치, 관리자는 웹을 통하여(SonarQube) 분석 결과확인



Docker를 이용한 SonarQube 구성도

제공된 리눅스 서버에 컨테이너 가상화 환경(Docker)을 구성하고,
SonarQube와 DB를 컨테이너로 생성하여 운영



03 SonarQube 설치



설치를 위한 다양한 방법

□ 기본 설치 프로그램 활용

- 다운로드: <https://www.sonarqube.org/downloads/>
- 실행 방법
 - Bin 폴더의 OS 별 실행
- 단점
 - Java 11 요구 / H2 DB 사용 여부

□ Docker 활용

- 다운로드
 - <https://github.com/SonarSource/docker-sonarqube/blob/master/example-compose-files/sq-with-postgres/docker-compose.yml>
- 실행 방법
 - docker-compose up
- 단점
 - Docker가 설치된 8GB 이상의 머신 필요

□ 가상 머신(Virtual Box) 활용

- 다운로드: <https://bitnami.com/stack/sonarqube/virtual-machine> (Bitnami 패키지)
- 실행 방법
 - Virtual Box에서 OVA Import
- 단점
 - 운영 환경에서 사용 가능한지 검토 필요

03-1 기본 설치 프로그램 활용



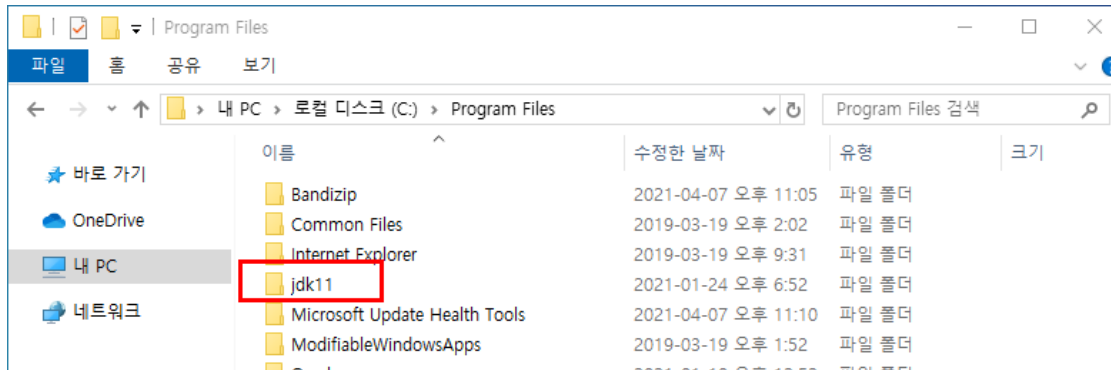
OpenJDK 11 설치 (Windows 가정)

❑ OpenJDK 다운로드

- <https://github.com/adoptopenjdk/adoptopenjdk>

❑ 적정 위치에 압축 해제 및 환경 변수 지정

- 본 예제에서는, Program Files 하위에 위치
- 편의를 위해, 폴더명은 JDK11로 변경

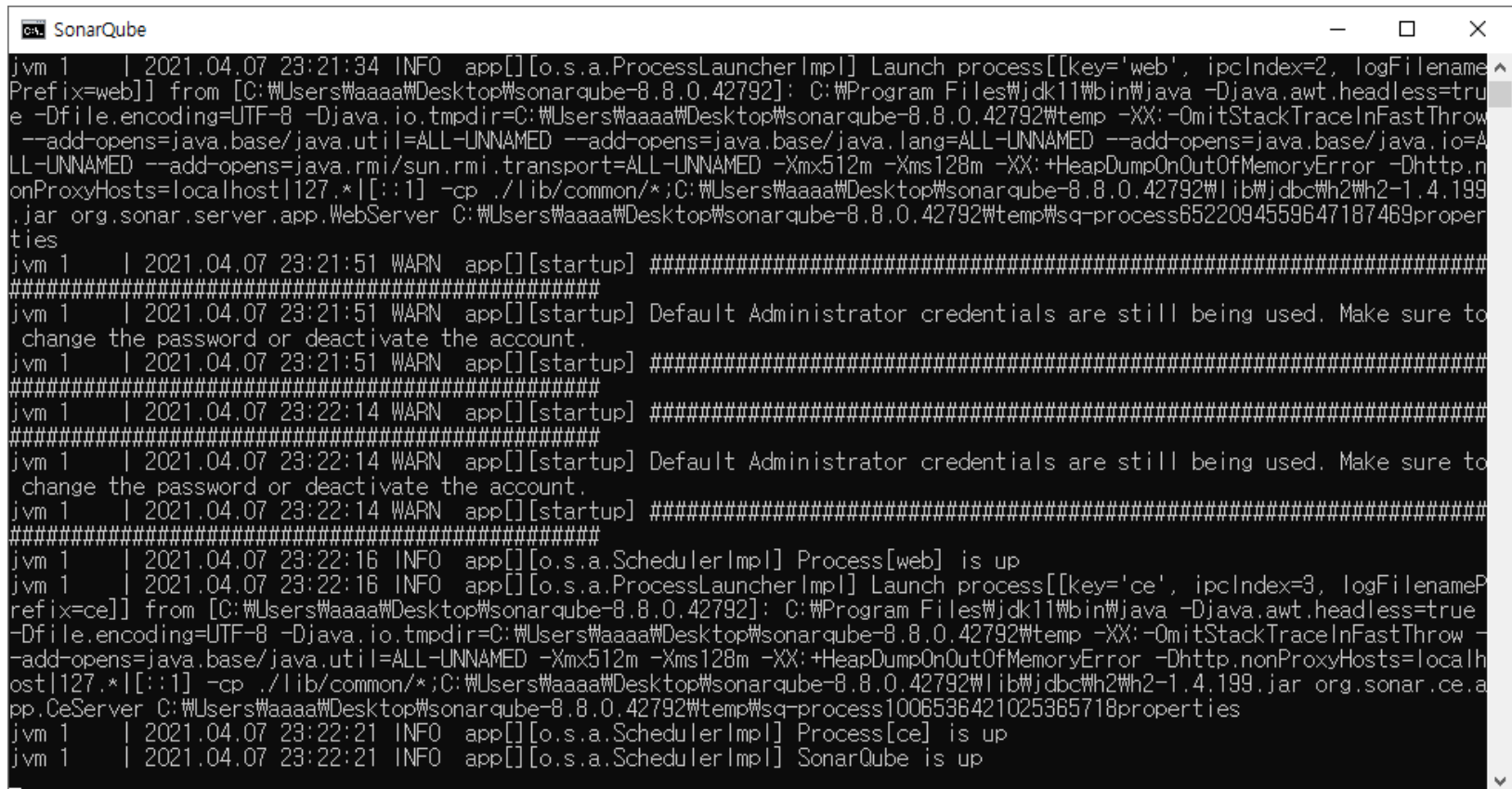


- 환경 변수 지정
 - Path에 bin 폴더 추가
 - JAVA_HOME에 jdk11 폴더 지정

SonarQube 실행

❑ bin\windows-x86-64 폴더에 실행 파일 위치

- StartSonar 더블클릭하여 실행



```

jvm 1 | 2021.04.07 23:21:34 INFO app[] [o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilename=
Prefix=web]] from [C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792]: C:\Program Files\jdk11\bin\java -Djava.awt.headless=true
-Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792\temp -XX:-OmitStackTraceInFastThrow
--add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[:1] -cp ./lib/common/*;C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792\lib\jdbc\h2\h2-1.4.199.jar org.sonar.server.app.WebServer C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792\temp\sq-process6522094559647187469properties
jvm 1 | 2021.04.07 23:21:51 WARN app[] [startup] #####
#####
jvm 1 | 2021.04.07 23:21:51 WARN app[] [startup] Default Administrator credentials are still being used. Make sure to
change the password or deactivate the account.
jvm 1 | 2021.04.07 23:21:51 WARN app[] [startup] #####
#####
jvm 1 | 2021.04.07 23:22:14 WARN app[] [startup] #####
#####
jvm 1 | 2021.04.07 23:22:14 WARN app[] [startup] Default Administrator credentials are still being used. Make sure to
change the password or deactivate the account.
jvm 1 | 2021.04.07 23:22:14 WARN app[] [startup] #####
#####
jvm 1 | 2021.04.07 23:22:16 INFO app[] [o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2021.04.07 23:22:16 INFO app[] [o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenameP
refix=ce]] from [C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792]: C:\Program Files\jdk11\bin\java -Djava.awt.headless=true
-Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792\temp -XX:-OmitStackTraceInFastThrow --add-opens=java.base/java.util=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|[:1] -cp ./lib/common/*;C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users\Waaaa\Desktop\sonarqube-8.8.0.42792\temp\sq-process1006536421025365718properties
jvm 1 | 2021.04.07 23:22:21 INFO app[] [o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2021.04.07 23:22:21 INFO app[] [o.s.a.SchedulerImpl] SonarQube is up
```

JAVA가 제대로 설치되지 않은 경우

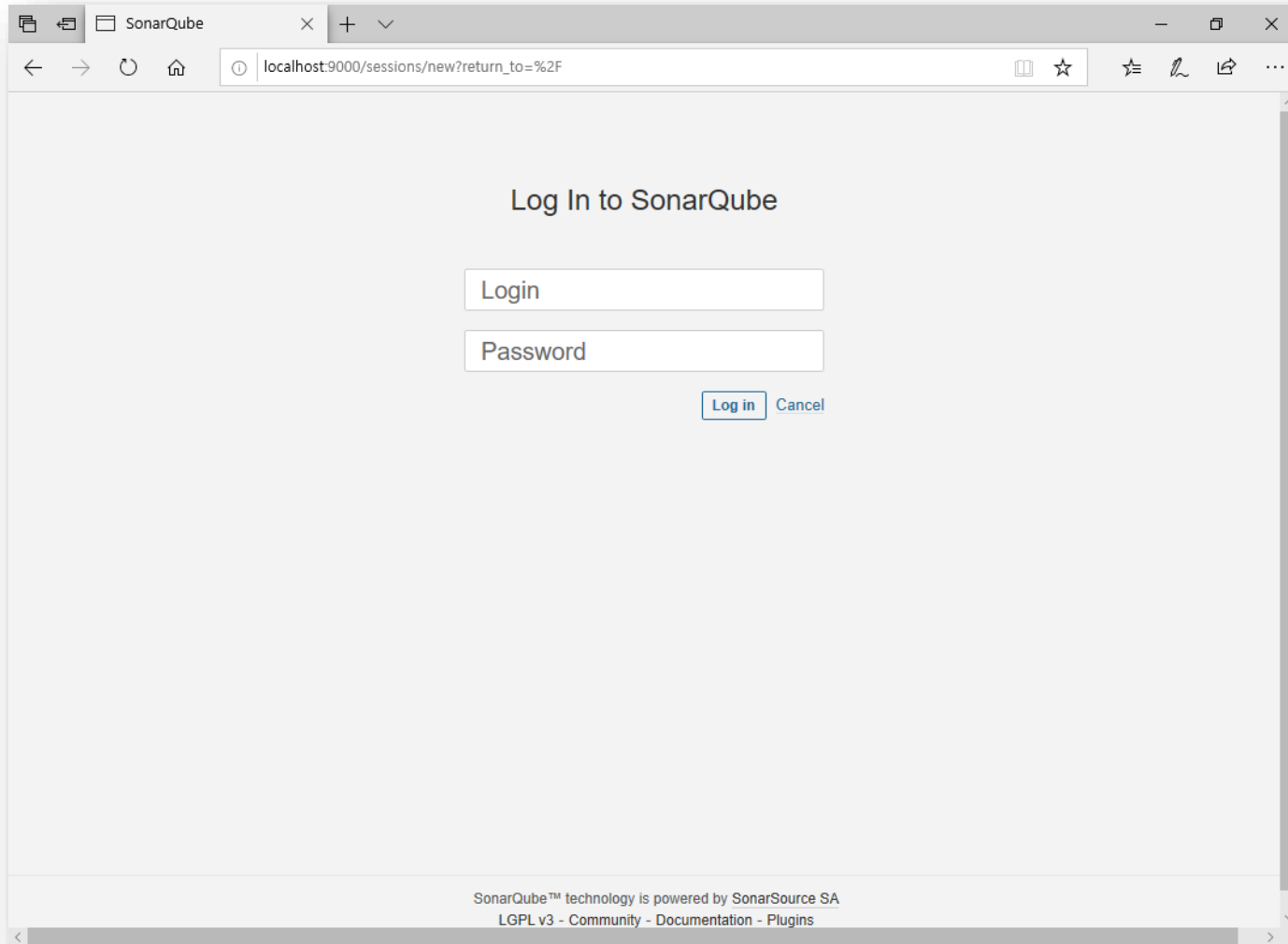
□ 에러 발생

```
SonarQube
wrapper --> Wrapper Started as Console
wrapper Launching a JVM...
jvm 1 Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
jvm 1 Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
jvm 1
jvm 1 WrapperSimpleApp: Encountered an error running main: java.lang.IllegalStateException: SonarQube requires Java
11 to run
jvm 1 java.lang.IllegalStateException: SonarQube requires Java 11 to run
jvm 1 at com.google.common.base.Preconditions.checkState(Preconditions.java:508)
jvm 1 at org.sonar.application.App.checkJavaVersion(App.java:93)
jvm 1 at org.sonar.application.App.start(App.java:56)
jvm 1 at org.sonar.application.App.main(App.java:97)
jvm 1 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
jvm 1 at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
jvm 1 at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
jvm 1 at java.lang.reflect.Method.invoke(Unknown Source)
jvm 1 at org.tanukisoftware.wrapper.WrapperSimpleApp.run(WrapperSimpleApp.java:240)
jvm 1 at java.lang.Thread.run(Unknown Source)
wrapper <-- Wrapper Stopped
계속하려면 아무 키나 누르십시오 . . .
```

초기 화면

□ 접속은 IP:9000

- <http://localhost:9000>
- 초기 계정: admin / admin



초기 비밀번호 변경

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

03-2 Docker 활용



Docker-Compose 활용

❑ Docker-Compose

- Docker 컨테이너를 미리 설정된 값으로 실행할 수 있도록 만든 설정 파일

❑ 사전조건

- Docker와 Docker-compose 설치 (공식 가이드에 따름)
 - <https://docs.docker.com/engine/install/ubuntu/#installation-methods>
 - <https://docs.docker.com/compose/install/>

❑ 다운로드 위치

- <https://github.com/SonarSource/docker-sonarqube/blob/master/example-compose-files/sq-with-postgres/docker-compose.yml>
- <https://github.com/SonarSource/docker-sonarqube.git> 를 Git으로 Clone 하는 것을 추천

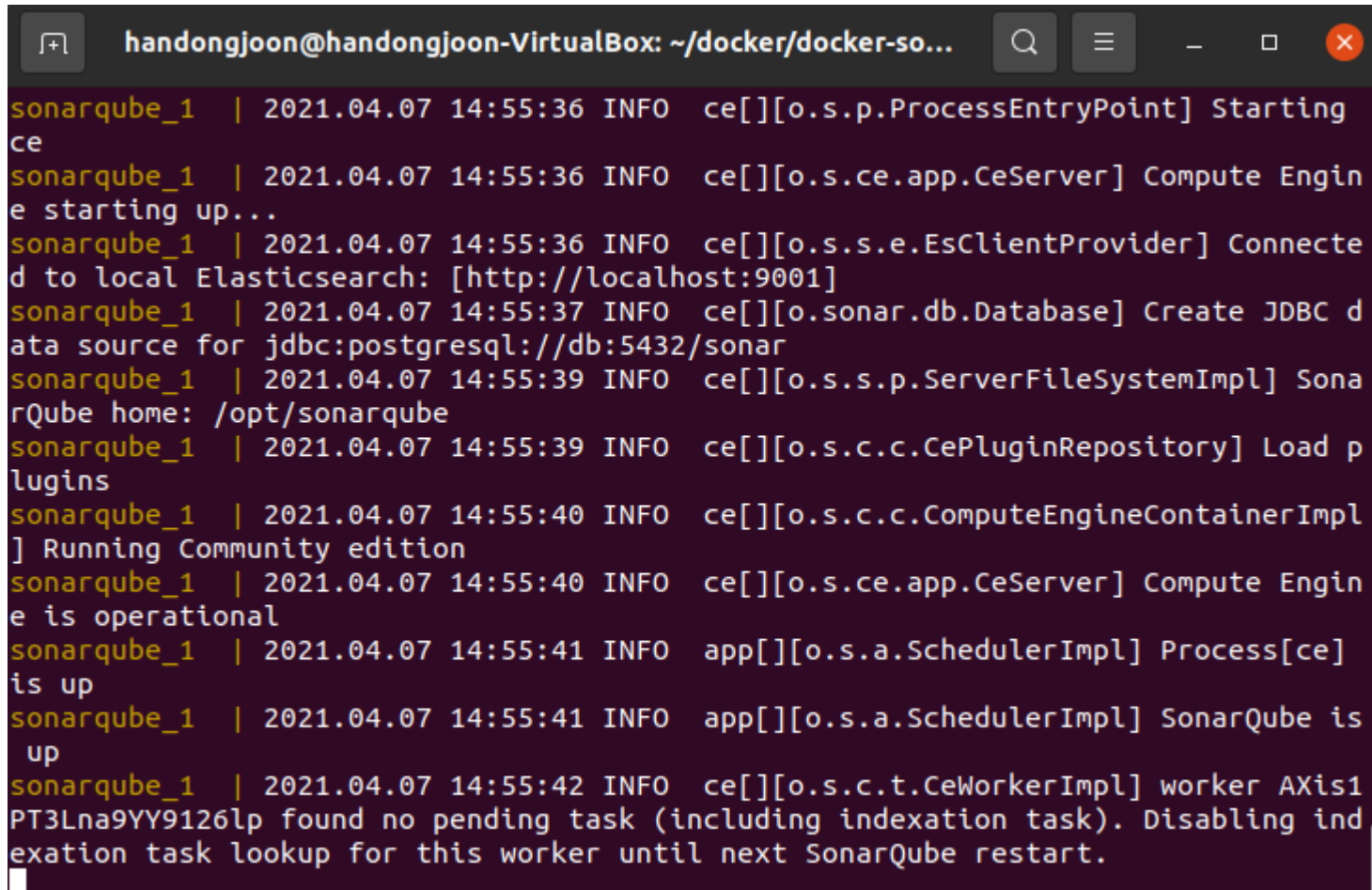
❑ SonarQube docker-compose

```
1  version: "3"
2  services:
3    sonarqube:
4      image: sonarqube:community
5      depends_on:
6        - db
```


Docker 컨테이너 실행

❑ docker-compose.yml 파일이 있는 위치에서, 다음 명령 실행

- docker-compose up

A terminal window with a dark background and light-colored text. The window title bar shows the user 'handongjoon' on a 'handongjoon-VirtualBox' machine, with the current directory being '~/docker/docker-so...'. The terminal output shows the logs for a SonarQube container named 'sonarqube_1'. The logs indicate the container is starting, the Compute Engine is starting up, it connects to a local Elasticsearch instance at http://localhost:9001, creates a JDBC data source for PostgreSQL, loads plugins, and finally reports that the SonarQube is operational and up. The last log entry shows a worker finding no pending tasks and disabling indexation task lookup.

```
handongjoon@handongjoon-VirtualBox: ~/docker/docker-so...  
sonarqube_1 | 2021.04.07 14:55:36 INFO ce[][o.s.p.ProcessEntryPoint] Starting  
ce  
sonarqube_1 | 2021.04.07 14:55:36 INFO ce[][o.s.ce.app.CeServer] Compute Engin  
e starting up...  
sonarqube_1 | 2021.04.07 14:55:36 INFO ce[][o.s.s.e.EsClientProvider] Connecte  
d to local Elasticsearch: [http://localhost:9001]  
sonarqube_1 | 2021.04.07 14:55:37 INFO ce[][o.sonar.db.Database] Create JDBC d  
ata source for jdbc:postgresql://db:5432/sonar  
sonarqube_1 | 2021.04.07 14:55:39 INFO ce[][o.s.s.p.ServerFileSystemImpl] Sona  
rQube home: /opt/sonarqube  
sonarqube_1 | 2021.04.07 14:55:39 INFO ce[][o.s.c.c.CePluginRepository] Load p  
lugins  
sonarqube_1 | 2021.04.07 14:55:40 INFO ce[][o.s.c.c.ComputeEngineContainerImpl  
] Running Community edition  
sonarqube_1 | 2021.04.07 14:55:40 INFO ce[][o.s.ce.app.CeServer] Compute Engin  
e is operational  
sonarqube_1 | 2021.04.07 14:55:41 INFO app[][o.s.a.SchedulerImpl] Process[ce]  
is up  
sonarqube_1 | 2021.04.07 14:55:41 INFO app[][o.s.a.SchedulerImpl] SonarQube is  
up  
sonarqube_1 | 2021.04.07 14:55:42 INFO ce[][o.s.c.t.CeWorkerImpl] worker AXIS1  
PT3Lna9YY9126lp found no pending task (including indexation task). Disabling ind  
exation task lookup for this worker until next SonarQube restart.
```

03-3 가상머신 활용



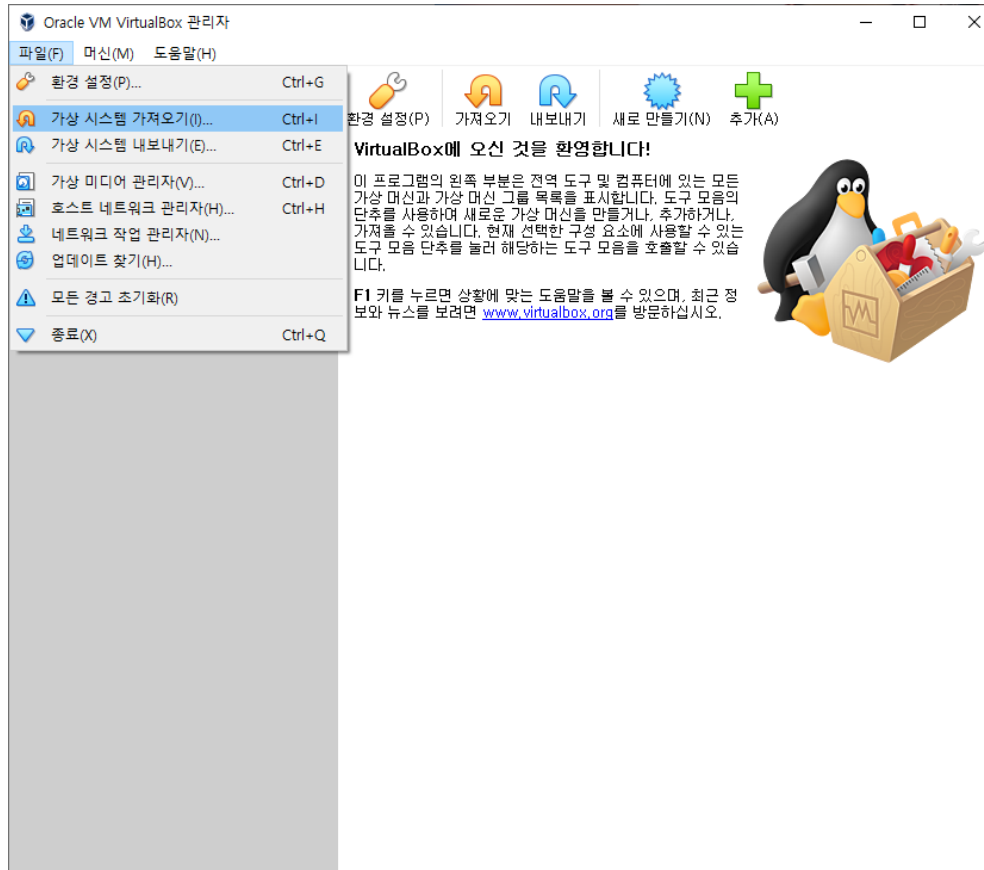
OVA 파일을 이용한 가상머신 설치

□ 사전조건

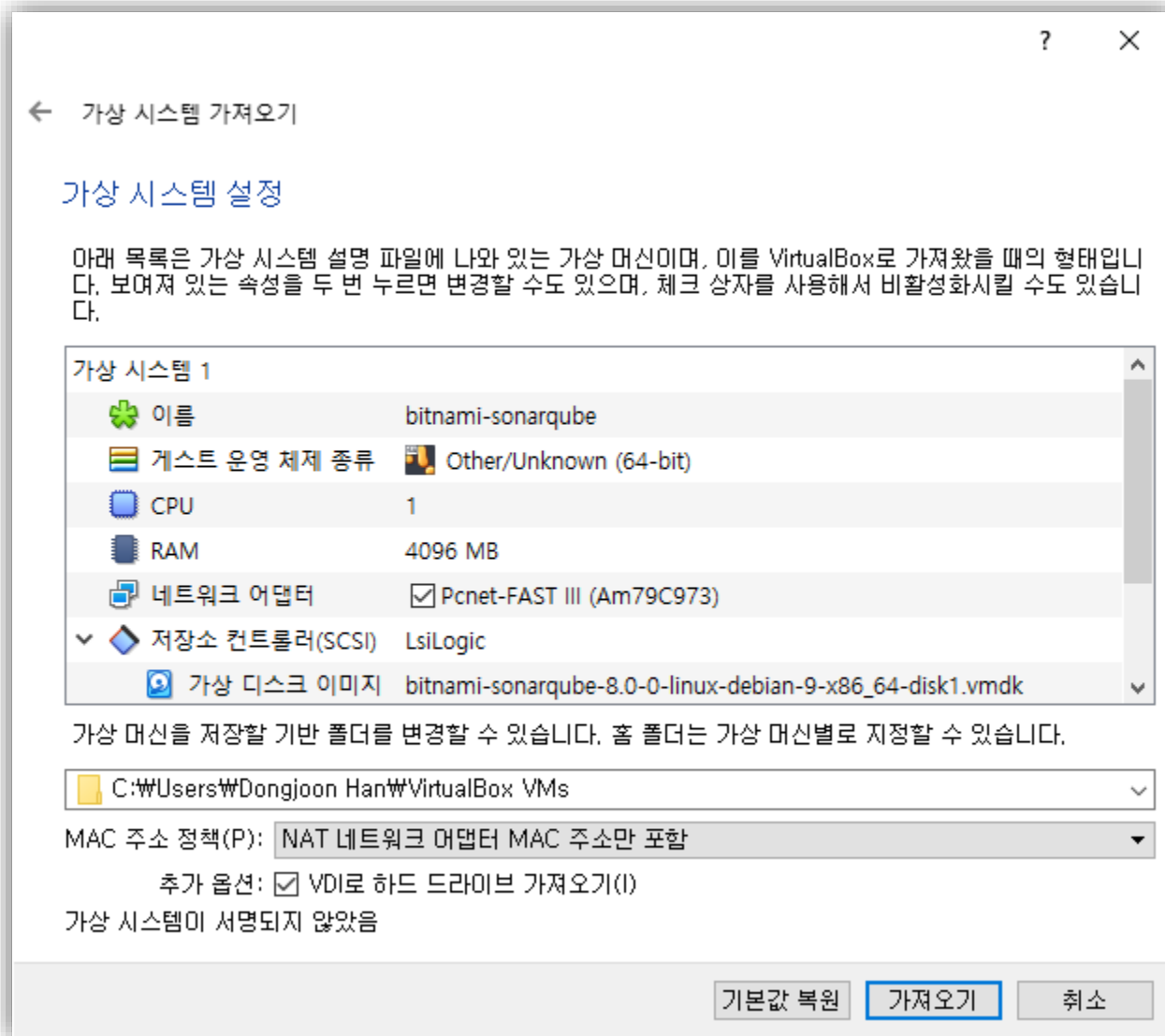
- Virtual Box 설치

□ 실행 방법

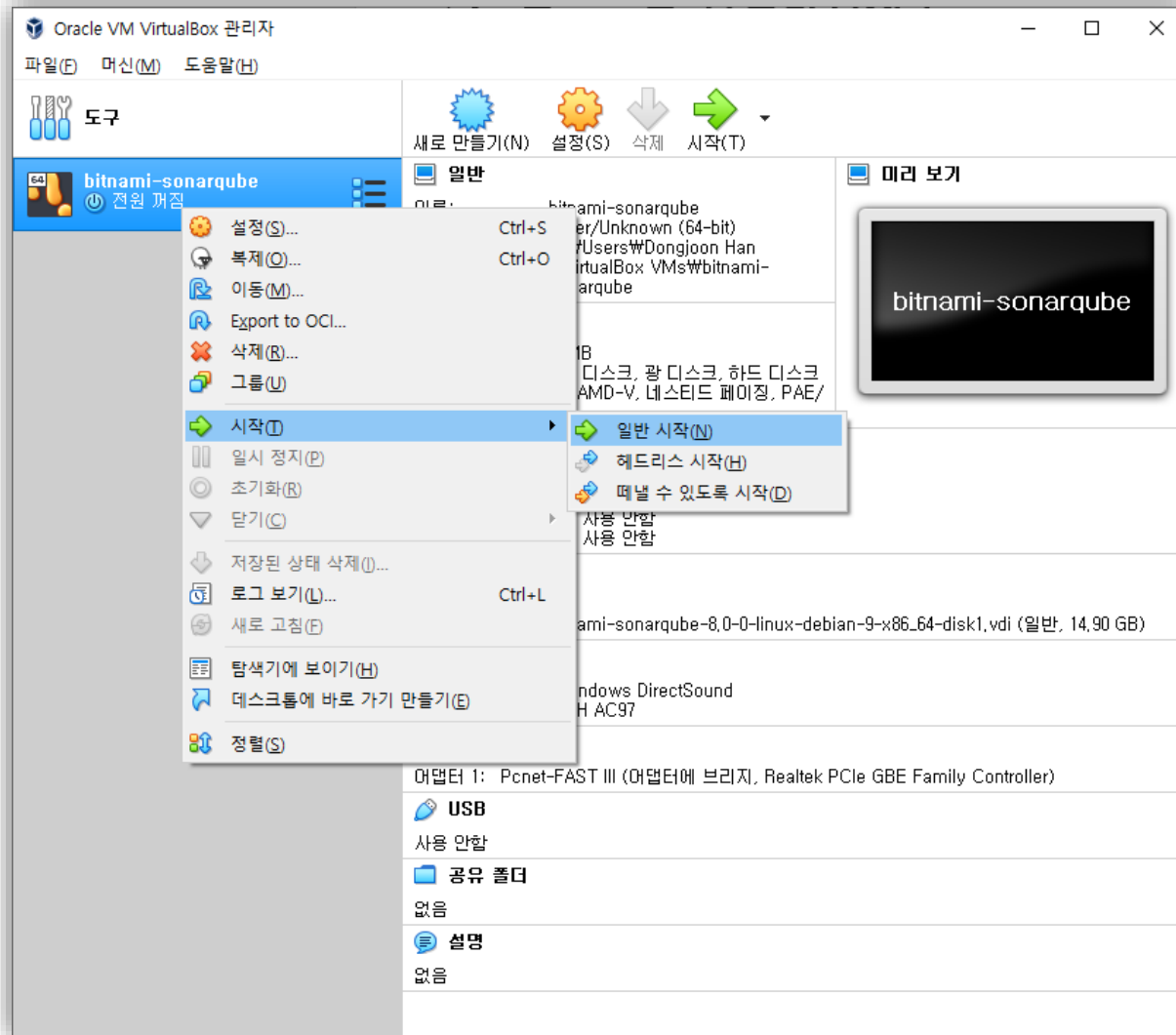
- 다운로드 후 더블클릭
- (또는) 파일 – 가상 시스템 가져오기



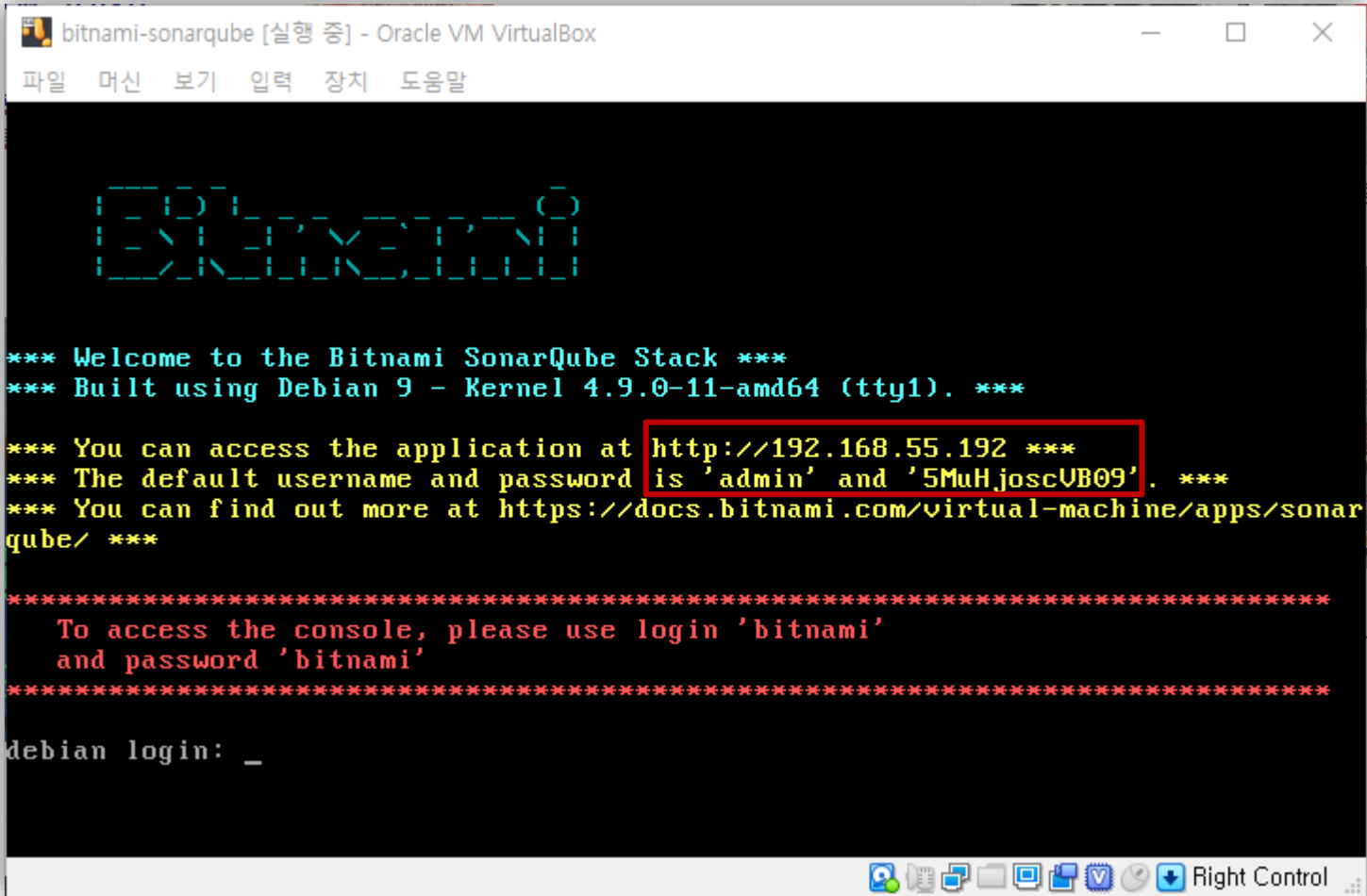
가상 시스템 설정 확인 후 가져오기



SonarQube 가상 머신 시작



IP와 SonarQube admin 계정 확인



```
bitnami-sonarqube [실행 중] - Oracle VM VirtualBox
파일  머신  보기  입력  장치  도움말

  ____  _
 / ___|| | | |
| |___| |_| |
 \___|_____|_|

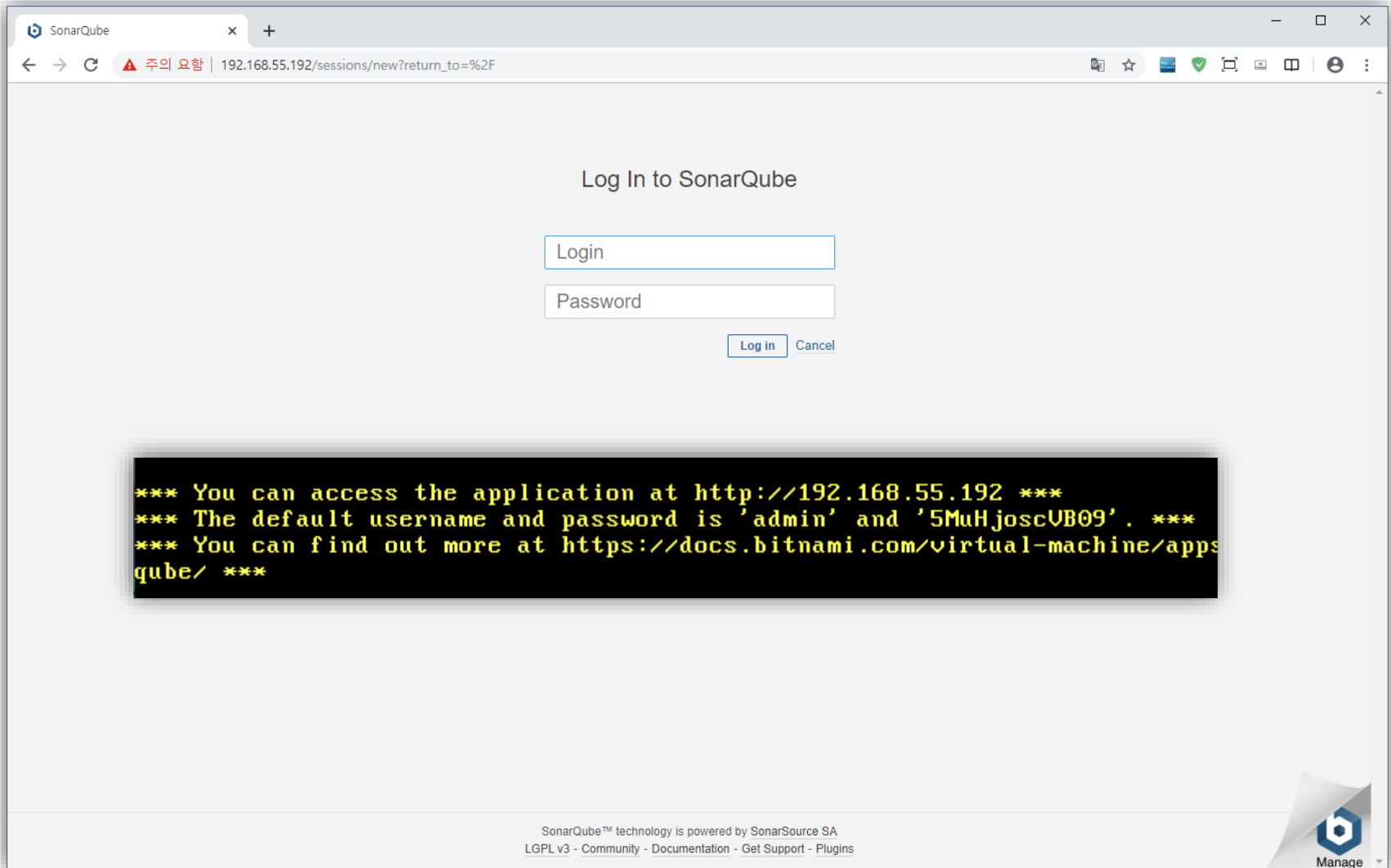
*** Welcome to the Bitnami SonarQube Stack ***
*** Built using Debian 9 - Kernel 4.9.0-11-amd64 (tty1). ***

*** You can access the application at http://192.168.55.192 ***
*** The default username and password is 'admin' and '5MuHjoscUB09'. ***
*** You can find out more at https://docs.bitnami.com/virtual-machine/apps/sonarqube/ ***

*****
  To access the console, please use login 'bitnami'
  and password 'bitnami'
*****

debian login: _
```

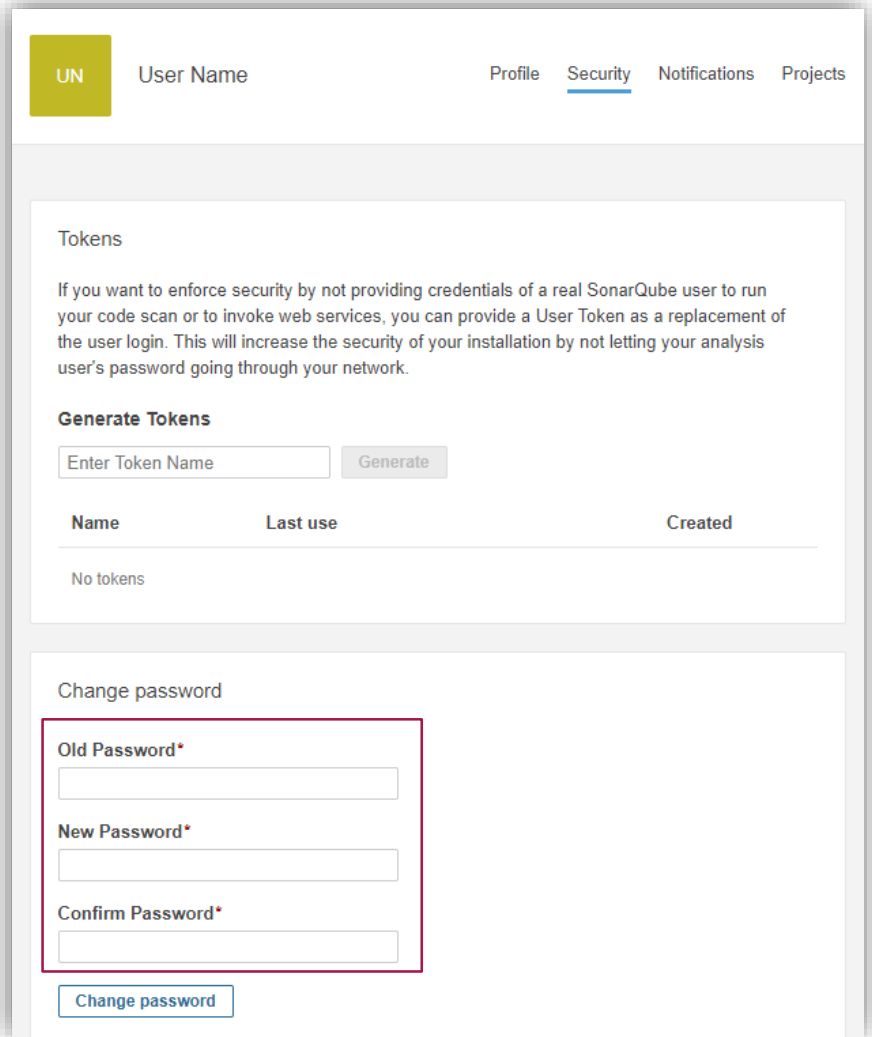
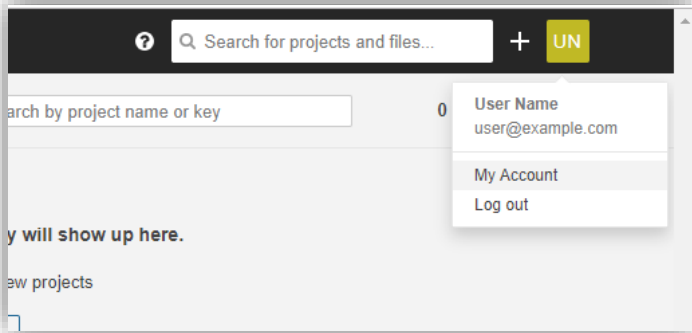
최초 접속



04 기본 설정

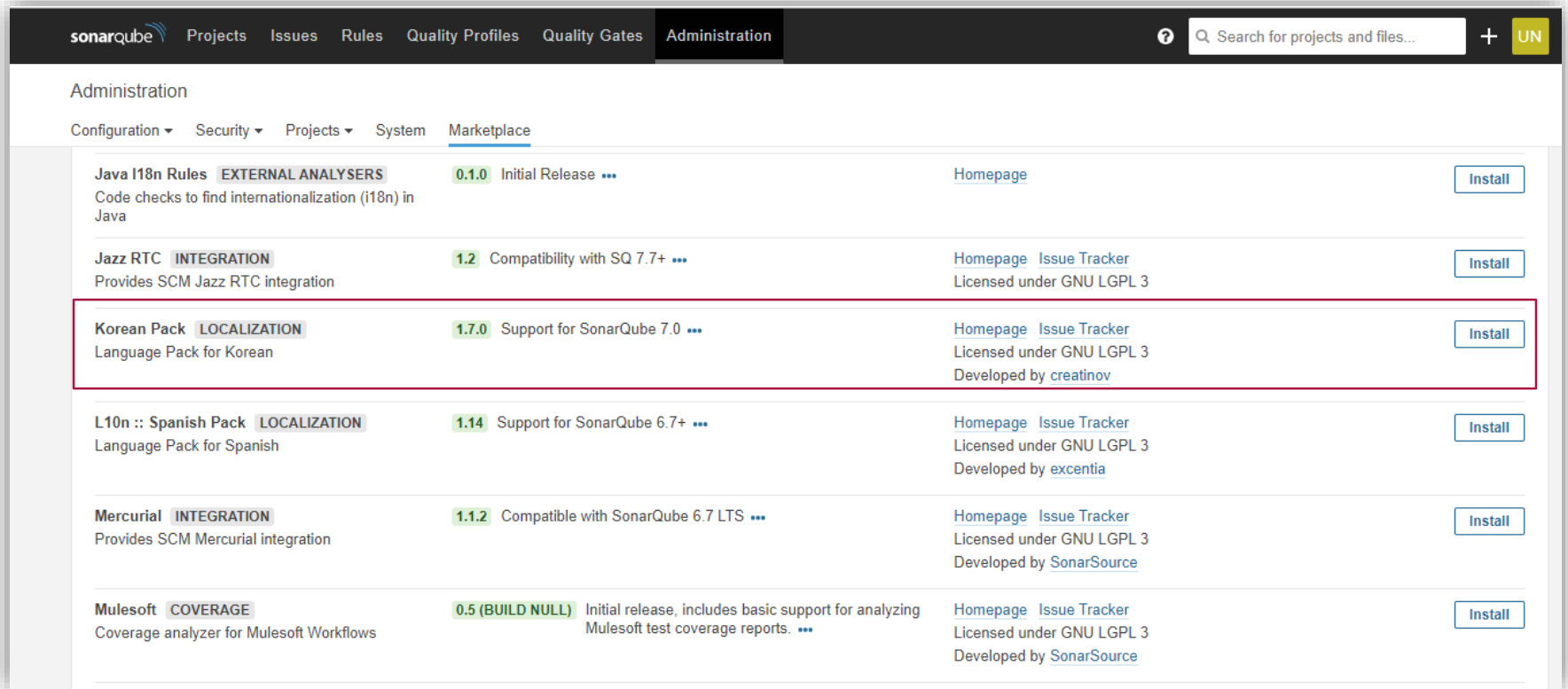


admin 계정의 비밀번호 변경

A screenshot of the SonarQube 'Security' page. The page has a header with a yellow 'UN' button, the user's name 'User Name', and navigation links for 'Profile', 'Security' (which is underlined), 'Notifications', and 'Projects'. The main content area is divided into two sections. The top section is titled 'Tokens' and contains a paragraph explaining that User Tokens can be used instead of real credentials for code scans. Below this is a 'Generate Tokens' section with a text input field for 'Enter Token Name' and a 'Generate' button. The bottom section is titled 'Change password' and contains three text input fields labeled 'Old Password*', 'New Password*', and 'Confirm Password*'. A 'Change password' button is located at the bottom of this section. A red box highlights the three password input fields.

한국어 언어팩 설치

❑ Administration -> Marketplace에서 Korean Pack 선택

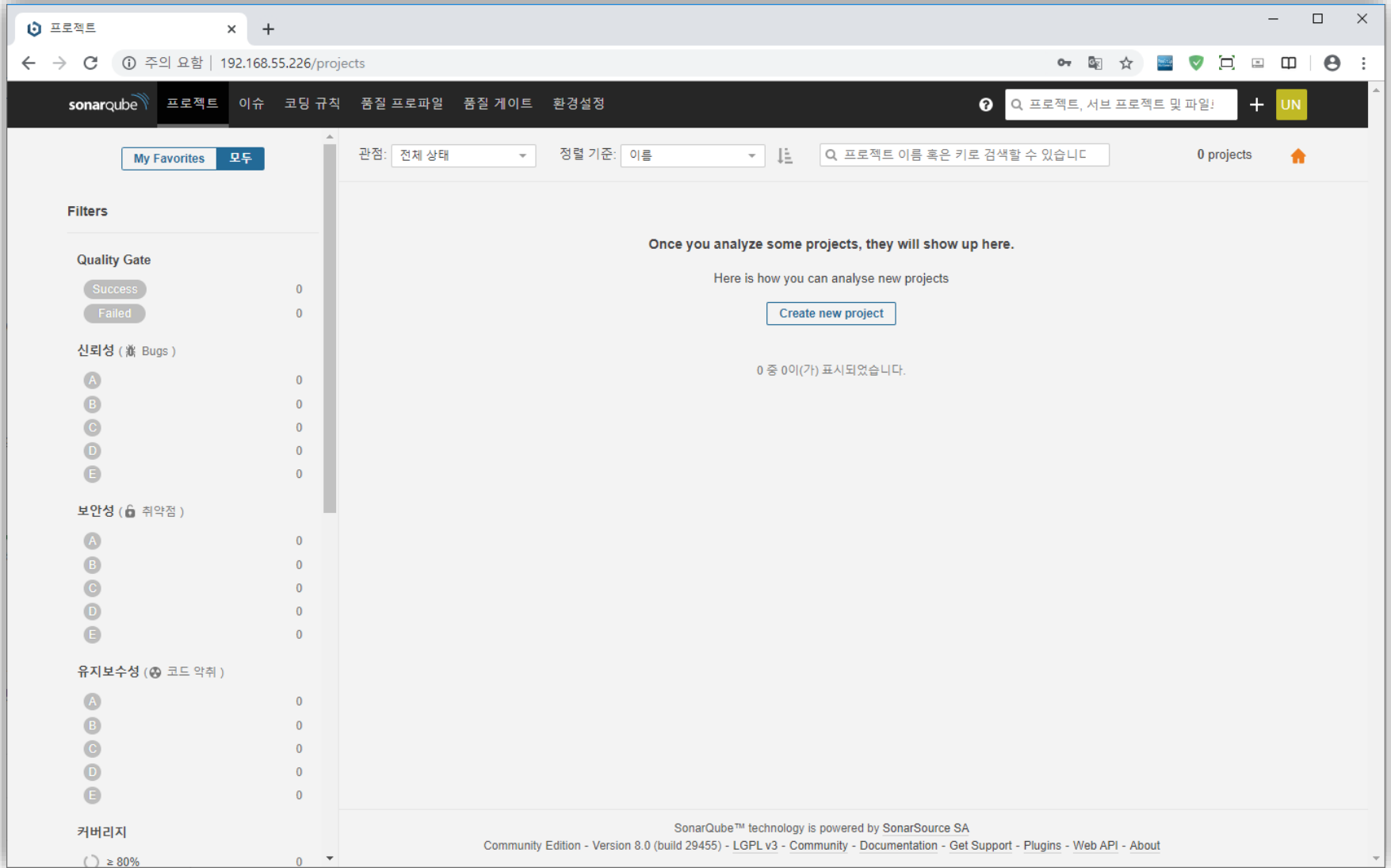


The screenshot shows the SonarQube Administration interface. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. A search bar is on the right. The 'Administration' section is active, and the 'Marketplace' sub-tab is selected. A list of available packs is shown, with the 'Korean Pack' highlighted by a red box. The 'Korean Pack' is categorized under 'LOCALIZATION', version '1.7.0', and supports SonarQube 7.0. It is licensed under GNU LGPL 3 and developed by creatinov. Other packs like 'Java I18n Rules', 'Jazz RTC', 'L10n :: Spanish Pack', 'Mercurial', and 'Mulesoft' are also listed.

Pack Name	Category	Version	Support	License	Developer	Action
Java I18n Rules	EXTERNAL ANALYSERS	0.1.0	Initial Release	GNU LGPL 3		Install
Jazz RTC	INTEGRATION	1.2	Compatibility with SQ 7.7+	GNU LGPL 3		Install
Korean Pack	LOCALIZATION	1.7.0	Support for SonarQube 7.0	GNU LGPL 3	creatinov	Install
L10n :: Spanish Pack	LOCALIZATION	1.14	Support for SonarQube 6.7+	GNU LGPL 3	excentia	Install
Mercurial	INTEGRATION	1.1.2	Compatible with SonarQube 6.7 LTS	GNU LGPL 3	SonarSource	Install
Mulesoft	COVERAGE	0.5 (BUILD NULL)	Initial release, includes basic support for analyzing Mulesoft test coverage reports.	GNU LGPL 3	SonarSource	Install

한국어 언어팩 적용 모습

□ 브라우저 설정에 따라 언어 표시



사용자 생성

□ 환경설정 -> 시큐리티 -> 사용자 -> 사용자 생성

The screenshot shows the SonarQube web interface with the 'Create User' dialog box open. The dialog box contains the following fields and sections:

- 로그인*** (Login): A text input field with a note '최소 3 문자' (Minimum 3 characters).
- 이름*** (Name): A text input field.
- Email**: A text input field.
- 비밀번호*** (Password): A text input field.
- SCM Accounts**: A section with a '추가' (Add) button and a note: '로그인 계정과 이메일 계정은 자동으로 SCM 계정으로 간주됩니다.' (Login and email accounts are automatically considered as SCM accounts).
- Buttons**: '생성' (Create) and '취소' (Cancel) buttons at the bottom right.

The background interface shows the '환경설정' (Settings) menu, the '시큐리티' (Security) sub-menu, and the '사용자' (Users) page. A table of users is visible, showing a user named 'admin' with email 'user@example.com'.

05 정적 분석 룰 설정



룰 설정의 필요성

□ 조직/프로젝트 특성에 맞는 룰 선정 필요

- Java 기준, SonarQube는 500여개 룰이 있으며, 이 중 300여개를 기본 검사함
- 과연, 우리에게 모두 필요한 룰인가?
 - 모든 룰은 유용하나, 우리의 현 특성에 딱 맞지는 않음

□ 룰 선정 시 고려할 항목

- 처음 시작하는 조직인가?
- 반드시 지켜야 하는 룰이 있는가?
 - 보안을 위한 CERT, OWASP 적용
 - 계약 상 발주사의 요청
 - MISRA 등 표준 요건 만족 필요
- 유지보수를 고려해야 하는가?
- QA가 품질 목표(위반 목표)를 제시하는가?

Quality Profile – SonarQube 룰셋

□ 개요

- 우리말: 품질 프로파일
- SonarQube의 룰셋(룰의 모음) 설정
- 언어 별 여러 룰셋을 지정하고, 해당 프로젝트에서 사용할 수 있음
- 특정 룰셋을 지정하지 않을 경우, 기본값 룰셋 적용

□ SonarQube의 기본 룰셋

- 각 언어 별 Sonar Way가 기본 룰셋
- Java의 경우, 약 350개의 룰을 기본 지정

□ 우리만의 룰셋을 지정하기 위한 Tip

- Sonar Way를 복사
- 불필요한 룰을 비활성화 처리
- 우리 룰셋을 기본값(Default) 처리

SonarQube 품질 프로파일 목록

The screenshot displays the SonarQube Quality Profiles page. The browser's address bar shows the URL `104.46.217.69/profiles`. The top navigation bar includes links for '프로젝트' (Projects), '이슈' (Issues), '코딩 규칙' (Coding Rules), '품질 프로파일' (Quality Profile), '품질 게이트' (Quality Gate), '환경설정' (Settings), and '더보기' (More). The '품질 프로파일' tab is highlighted with a red dashed box. The main content area lists quality profiles for various languages, each with a table of associated projects, rules, and update status.

Language	Profile Name	Projects	Rules	Updated	Used
Java, 1 프로파일	Sonar way Built-in	기본값	391	2개월 전	Never
JavaScript, 2 프로파일	Sonar way Built-in	기본값	101	2개월 전	Never
	Sonar way Recommended Built-in	0	141	2개월 전	Never
JSP, 1 프로파일	Sonar way Built-in	기본값	0	2개월 전	Never
Kotlin, 1 프로파일	Sonar way Built-in	기본값	31	2개월 전	Never
PHP, 3 프로파일	Drupal Built-in	0	21	2개월 전	Never
	PSR-2 Built-in	0	20	2개월 전	Never
	Sonar way Built-in	기본값	107	2개월 전	Never

품질 프로파일 복사

❑ Java - Sonar Way의 복사

Java, 1 프로파일	Projects ?	Rules	Updated	Used
Sonar way Built-in	기본값	391	2개월 전	Never
JavaScript, 2 프로파일	Projects ?	Rules	Updated	

- 비교
- 복사
- Extend



다음 프로파일을 복사합니다: **Sonar way**

새 이름*

Synetics Way

복사 취소

복사한 품질 프로파일 Main 화면

The screenshot shows the SonarQube web interface for a quality profile named 'Synetics Way'. The browser address bar shows the URL: 104.46.217.69/profiles/show?language=java&name=Synetics+Way. The SonarQube navigation bar includes links for '프로젝트', '이슈', '코딩 규칙', '품질 프로파일', '품질 게이트', '환경설정', and '더보기'. The main content area is titled '품질 프로파일 / Java' and 'Synetics Way'. It displays a table of rule statistics, a summary of the profile's status, and a section for project assignments.

규칙	활성화	비활성화
Total	391	190
Bugs	113	12
Vulnerabilities	36	8
Code Smells	212	168
Security Hotspots	30	2

Buttons: [더 많은 규칙 활성화](#)

프로파일 상속: Synetics Way, 391개의 규칙 활성화, 0개의 오버라이딩 된 규칙. Button: [부모 프로파일 변경](#)

프로젝트: 이 프로파일을 명시적으로 할당한 프로젝트가 없습니다. Button: [프로젝트 변경](#)

권한: 글로벌 "Manage Quality Profile" 권한을 가진 사용자들은 이 quality profile을 관리할 수 있습니다. Button: [더 많은 사용자에게 권한 부여](#)

Footer: SonarQube™ technology is powered by SonarSource SA. Community Edition - Version 7.9.1 (build 27448) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

불필요 룰 비활성화

❑ Code Smell의 Minor 룰 비활성화

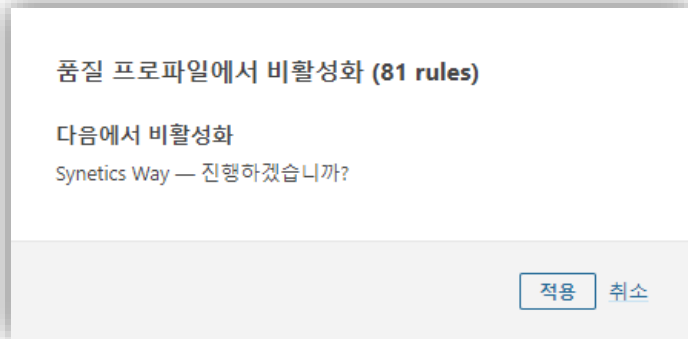
- “대규모 변경”을 통해, 선택된 룰 전체를 한 번에 비활성화 함

The screenshot shows the SonarQube web interface. The browser address bar indicates the URL: `104.46.217.69/coding_rules?activation=true&qprofile=AW6UKrVFcfU0gnjY1pj&severities=MINOR&types=CODE_SMELL`. The left sidebar contains filters for Language (Java), Tag, Repository, Default Severity (Minor), Status, Security Category, Available Since, Template, Quality Profile (Synetics...), and Inheritance. The main area displays a list of Code Smell rules. A tooltip is visible over the '대규모 변경' button, showing options: '다음에서 활성화...' (Activate from here...), '다음에서 비활성화...' (Deactivate from here...), and '다음에서 비활성화 Synetics Way' (Deactivate from here Synetics Way). The list of rules includes:

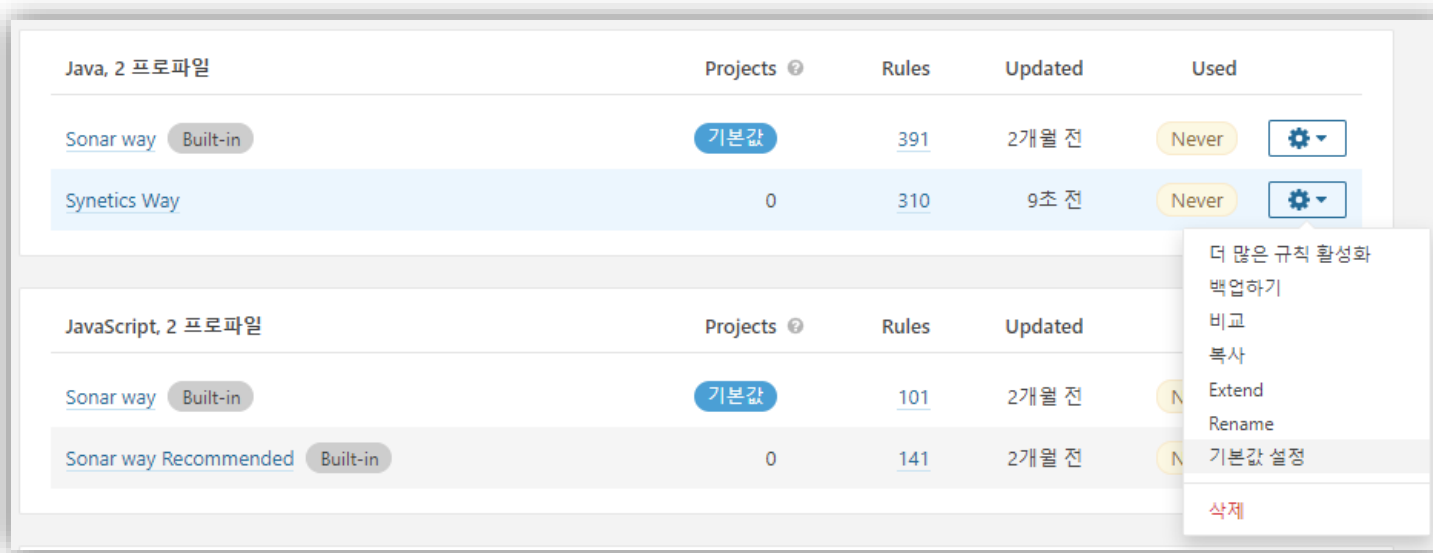
- for "@Nullable" should not be used on primitive types
- "@Deprecated" code should not be used
- "catch" clauses should do more than rethrow
- "close()" calls should not be redundant
- "Collections.EMPTY_LIST", "EMPTY_MAP", and "EMPTY_SET" should not be used
- "equals(Object obj)" should be overridden along with the "compareTo(T obj)" method
- "finalize" should not set fields to "null"
- "indexOf" checks should use a start position
- "private" methods called only by inner classes should be moved to those classes
- "read(byte[],int,int)" should be overridden
- "StandardCharsets" constants should be preferred
- "Stream" call chains should be simplified when possible
- "switch" statements should have at least 3 "case" clauses
- "ThreadLocal.withInitial" should be preferred

불필요 룰 비활성화와 기본값 설정

□ 비활성화 확인



□ 기본값 설정



업계에서 많이 위반하는 룰 No.1

❑ String의 값 비교 방법

- 개요: String의 값 비교 시 ==를 사용하지 말아야 한다. equals()를 이용해야 한다.
- 의미: int의 ==는 값 비교이나, String의 ==는 주소값이 같은지 확인한다.
- SonarQube 룰: Strings and Boxed types should be compared using "equals()"
- 예제

Noncompliant Code Example

```
String firstName = getFirstName(); // String overrides equals
String lastName = getLastName();

if (firstName == lastName) { ... }; // Non-compliant; false even if the strings have the same value
```

Compliant Solution

```
String firstName = getFirstName();
String lastName = getLastName();

if (firstName != null && firstName.equals(lastName)) { ... };
```

업계에서 많이 위반하는 룰 No.2

❑ 중요 정보의 하드 코딩

- 개요: 중요 정보인 IP, ID, PW는 하드코딩 하지 말아야 한다.
- 의미: 중요 정보의 하드 코딩은 보안 취약점이다.
- SonarQube 룰: Credentials should not be hard-coded
- 예제

Noncompliant Code Example

```
Connection conn = null;
try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
        "user=steve&password=blue"); // Noncompliant
    String uname = "steve";
    String password = "blue";
    conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
        "user=" + uname + "&password=" + password); // Noncompliant

    java.net.PasswordAuthentication pa = new java.net.PasswordAuthentication("userName", "1234".toCharArray());
}
```

Compliant Solution

```
Connection conn = null;
try {
    String uname = getEncryptedUser();
    String password = getEncryptedPass();
    conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
        "user=" + uname + "&password=" + password);
}
```

06 분석 실행 – Java Maven 경우



Maven 프로젝트 실행 방법

❑ Maven의 Goal을 이용하여 실행 가능

- sonar:sonar
- 별도의 SonarScanner 설치 및 설정 필요 없음

❑ Goal 실행 시, SonarQube 프로젝트 정보 지정 필요

- 서버 주소
- 프로젝트 ID
- 로그인 토큰

❑ 실행 방법은 프로젝트 생성 시, SoanrQube에서 안내

- 예)

로컬 컴퓨터에서 SonarQube Scanner for Maven 실행하기

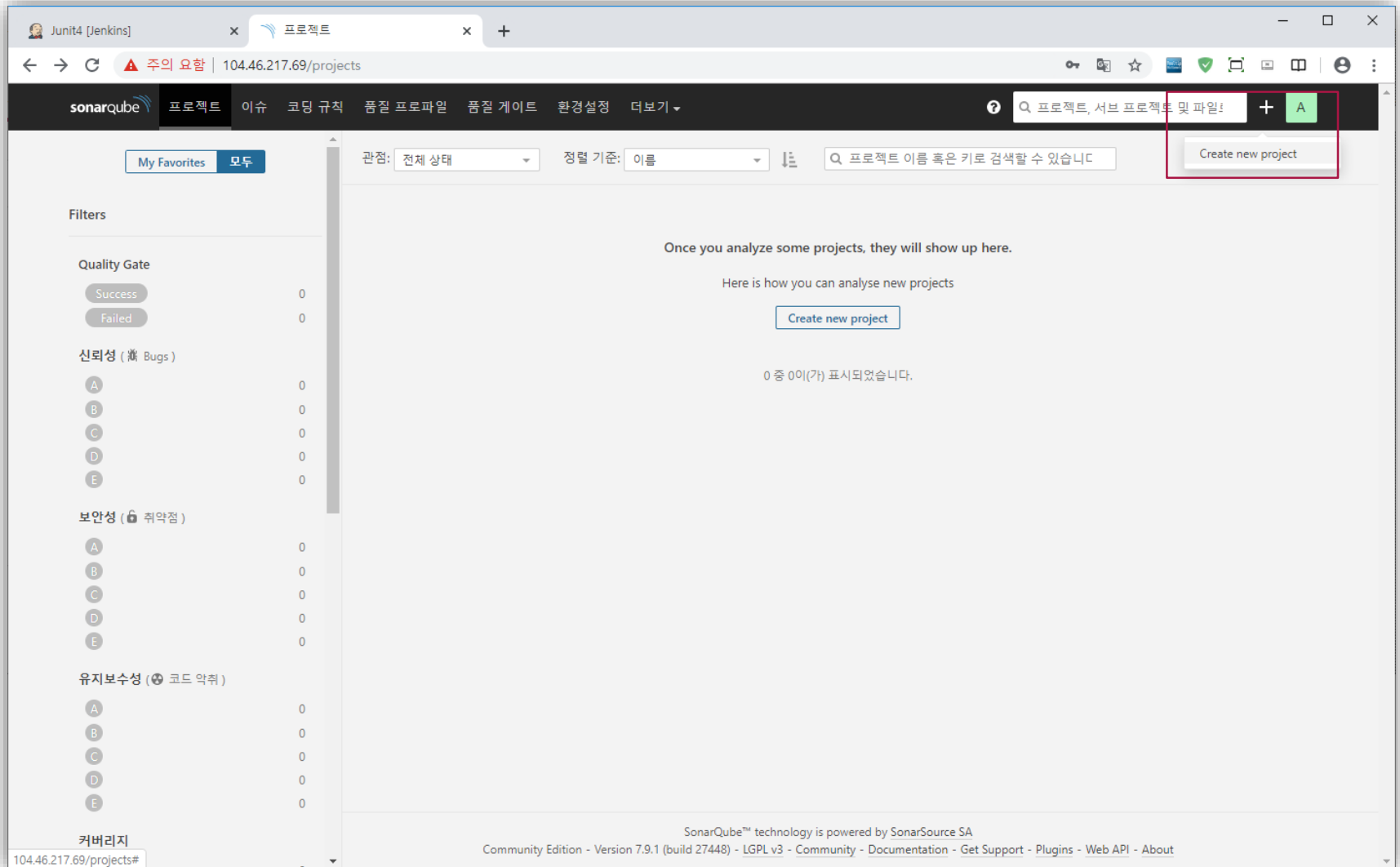
Maven과 함께 SonarQube를 매우 쉽게 실행할 수 있습니다. 프로젝트 폴더에서 다음 명령어를 실행합니다:

```
mvn sonar:sonar \
  -Dsonar.projectKey=JUnit4 \
  -Dsonar.host.url=http://104.46.217.69 \
  -Dsonar.login=088ac21e667933f8d7b5d218af9b1d91b23702f8
```

복사

프로젝트 생성

□ 우측 상단의 + 클릭



프로젝트 생성

❑ 프로젝트 키값과 프로젝트 표시명 지정

- 프로젝트 키값은 분석 실행 시 필수 입력 값임

Create new project

Project key* ?

JUnit4

Up to 400 characters. All letters, digits, dash, underscore, period or colon.

Display name* ?

JUnit4

Up to 255 characters

Set Up

로그인을 위한 토큰 생성

□ 아이디/패스워드 또는 토큰 방식으로 분석에 대한 권한 부여 가능

- 보안을 위해 권한 있는 사람만 분석 실행 및 결과 확인 가능하도록 설정 필요

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!

1 토큰 생성하기

admin: 088ac21e667933f8d7b5d218af9b1d91b23702f8 

분석 수행시, 토큰을 활용해 사용자를 식별합니다. 토큰을 사용하지 않을 경우에는 언제든지 여러분의 계정에서 토큰을 삭제할 수 있습니다.

Continue

2 프로젝트 분석 실행하기

프로젝트 특성에 맞는 분석 선택

□ 예제는 Java - Maven을 선택

2 프로젝트 분석 실행하기

프로젝트는 주로 어떤 언어로 구현되어 있습니까?

Java C# 혹은 VB.NET 기타 (JS, Python, PHP, ...)

주로 Java 언어로 개발을 하고 계십니다: 어떤 빌드 기법을 사용하십니까?

Maven Gradle

로컬 컴퓨터에서 SonarQube Scanner for Maven 실행하기

Maven과 함께 SonarQube를 매우 쉽게 실행할 수 있습니다. 프로젝트 폴더에서 다음 명령어를 실행합니다:

```
mvn sonar:sonar \
  -Dsonar.projectKey=JUnit4 \
  -Dsonar.host.url=http://104.46.217.69 \
  -Dsonar.login=088ac21e667933f8d7b5d218af9b1d91b23702f8
```

복사

Please visit the [official documentation of the Scanner for Maven](#) for more details.

Once the analysis is completed, this page will automatically refresh and you will be able to browse the analysis results.

프로젝트 특성에 맞는 분석 선택

❑ 예제는 기타-Windows를 선택

- SonarScanner가 어느 위치에서도 실행 가능하도록, Path 등록이 필요

2 프로젝트 분석 실행하기

프로젝트는 주로 어떤 언어로 구현되어 있습니까?

Java C# 혹은 VB.NET **기타 (JS, Python, PHP, ...)**

운영체제는 무엇을 사용하십니까?

Linux **Windows** macOS

Windows 용 SonarQube Scanner를 다운로드하고 압축을 풉니다

And add the `bin` directory to the `%PATH%` environment variable

[다운로드](#)

로컬 컴퓨터에서 SonarQube Scanner 실행하기

SonarQube 분석을 매우 쉽게 실행할 수 있습니다. 프로젝트 폴더에서 다음 명령어를 실행합니다:

```
sonar-scanner.bat -D"sonar.projectKey=JS_Test" -D"sonar.sources=." -D"sonar.host.url=http://104.46.217.69" -D"sonar.login=088ac21e66793"
```

[복사](#)

더 자세한 정보는 <http://redirect.sonarsource.com/doc/install-configure-scanner.html> official documentation of the SonarQube Scanner 페이지를 참조하십시오.

[실습] JUnit4의 분석 실행

□ 사전조건

- Maven 설치 및 설정
- Git 설치 및 설정

□ 실행 순서

1. Github에서 JUnit4 Clone
2. Compile 수행
 1. mvn compile
3. SonarQube에서 JUnit4 프로젝트 생성
4. mvn 명령에 따라 분석 실행
5. Build Success 확인
6. SonarQube에서 결과 확인

[실습] JUnit4의 분석 실행

```
C:\WINDOWS\system32\cmd.exe
C:\git\junit4>mvn sonar:sonar -Dsonar.projectKey=JUnit4 -Dsonar.host.url=http://104.46.217.69 -Dsonar.login=088ac21e6
67933f8d7b5d218af9b1d91b23702f8
[INFO] Scanning for projects...
[WARNING] The project junit:junit:jar:4.13-SNAPSHOT uses prerequisites which is only intended for maven-plugin projects
but not for non maven-plugin projects. For such purposes you should use the maven-enforcer-plugin. See https://maven.apa
che.org/enforcer/enforcer-rules/requireMavenVersion.html
[INFO]
[INFO] -----< junit:junit >-----
[INFO] Building JUnit 4.13-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- sonar-maven-plugin:3.7.0.1746:sonar (default-cli) @ junit ---
[INFO] User cache: C:\Users\Dongjoon Han\.sonar\cache
[INFO] SonarQube version: 7.9.1
[INFO] Default locale: "ko_KR", source code encoding: "ISO-8859-1"
[WARNING] SonarScanner will require Java 11+ to run starting in SonarQube 8.x
[INFO] Load global settings
[INFO] Load global settings (done) | time=142ms
[INFO] Server id: 243B8A4D-AW1MEreOnU2WJCLYkDIJ
[INFO] User cache: C:\Users\Dongjoon Han\.sonar\cache
[INFO] Load/download plugins
[INFO] Load plugins index
[INFO] Load plugins index (done) | time=72ms
[INFO] Plugin [l10nko] defines 'l10nen' as base plugin. This metadata can be removed from manifest of l10n plugins since
version 5.2.
[INFO] Load/download plugins (done) | time=118ms
[INFO] Process project properties
[INFO] Execute project builders
[INFO] Execute project builders (done) | time=5ms
[INFO] Project key: JUnit4
```

[실습] JUnit4의 분석 실행

```
C:\WINDOWS\system32\cmd.exe
[INFO] Sensor JaCoCo XML Report Importer [jacoco] (done) | time=8ms
[INFO] Sensor Packages sensor [jdepend]
[INFO] Sensor Packages sensor [jdepend] (done) | time=72ms
[INFO] ----- Run sensors on project
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=197ms
[INFO] Sensor Java CPD Block Indexer
[INFO] Sensor Java CPD Block Indexer (done) | time=183ms
[INFO] SCM provider for this project is: git
[INFO] 467 files to be analyzed
[INFO] 467/467 files analyzed
[INFO] 89 files had no CPD blocks
[INFO] Calculating CPD for 127 files
[INFO] CPD calculation finished
[INFO] Analysis report generated in 627ms, dir size=2 MB
[INFO] Analysis report compressed in 5005ms, zip size=1 MB
[INFO] Analysis report uploaded in 402ms
[INFO] ANALYSIS SUCCESSFUL, you can browse http://104.46.217.69/dashboard?id=JUnit4
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://104.46.217.69/api/ce/task?id=AW6U03lkcfclU0gnjY14h
[INFO] Analysis total time: 41.417 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 46.206 s
[INFO] Finished at: 2019-11-23T02:49:16+09:00
[INFO] -----
C:\git\junit4>mvn compile
```


[실습] JUnit4의 분석 실행

□ 프로젝트 목록

The screenshot displays the SonarQube web interface for a project named 'JUnit4'. The browser tabs show 'JUnit4 [Jenkins]', '프로젝트', and 'junit-team/junit4: A programme'. The address bar indicates the URL '104.46.217.69/projects'. The SonarQube header includes navigation links like '프로젝트', '이슈', '코딩 규칙', '품질 프로파일', '품질 게이트', '환경설정', and '더보기'. A search bar is present with the text '프로젝트, 서브 프로젝트 및 파일'. The main content area shows the project 'JUnit4' with a 'Success' status. Key metrics are displayed: 12 Bugs (C), 5 취약점 (B), 842 코드 악취 (A), 0.0% Coverage, and 0.0% Duplications. The last analysis was performed on 2019년 11월 23일 오전 2:48. The left sidebar contains filters for 'Quality Gate' (Success/Failed), '신뢰성 (Bugs)' (A-E), '보안성 (취약점)' (A-E), and '유지보수성 (코드 악취)' (A-E). The footer mentions 'SonarQube™ technology is powered by SonarSource SA' and provides links for 'Community Edition - Version 7.9.1 (build 27448)', 'LGPL v3', 'Community', 'Documentation', 'Get Support', 'Plugins', 'Web API', and 'About'.

JUnit4 [Jenkins] x 프로젝트 x junit-team/junit4: A programme x +

← → ↺ 주의 요함 | 104.46.217.69/projects

sonarqube 프로젝트 이슈 코딩 규칙 품질 프로파일 품질 게이트 환경설정 더보기

프로젝트, 서브 프로젝트 및 파일

관점: 전체 상태 정렬 기준: 이름

Q 프로젝트 이름 혹은 키로 검색할 수 있습니다

1 projects

☆ JUnit Success

마지막 분석: 2019년 11월 23일 오전 2:48

12 C Bugs | 5 B 취약점 | 842 A 코드 악취 | 0.0% Coverage | 0.0% Duplications | 11k M Java, XML

1 중 1이(가) 표시되었습니다.

Filters

Quality Gate

Success 1 |

Failed 0 |

신뢰성 (Bugs)

A 0 |

B 0 |

C 1 |

D 0 |

E 0 |

보안성 (취약점)

A 0 |

B 1 |

C 0 |

D 0 |

E 0 |

유지보수성 (코드 악취)

A 1 |

B 0 |

C 0 |

D 0 |

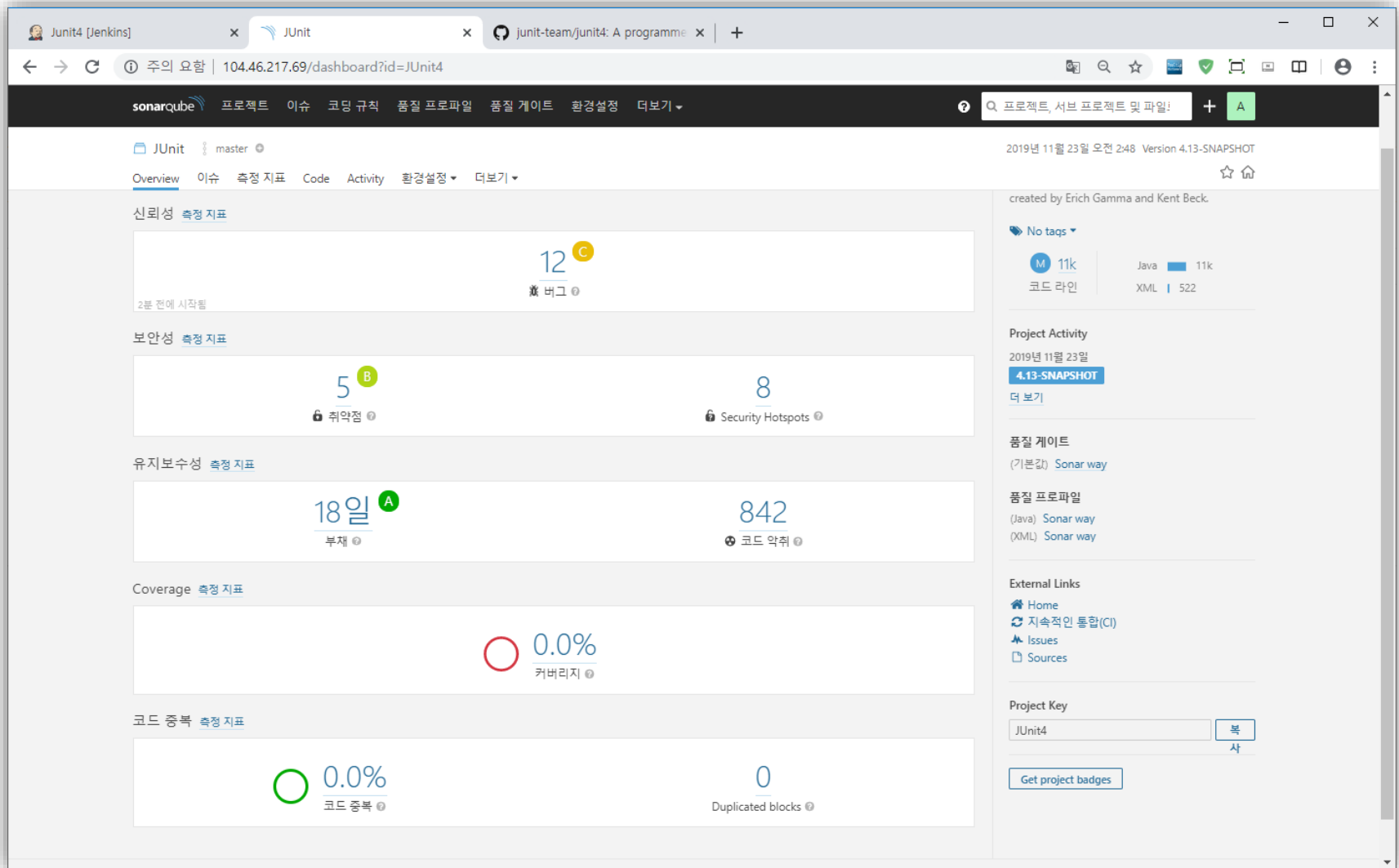
E 0 |

SonarQube™ technology is powered by SonarSource SA

Community Edition - Version 7.9.1 (build 27448) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

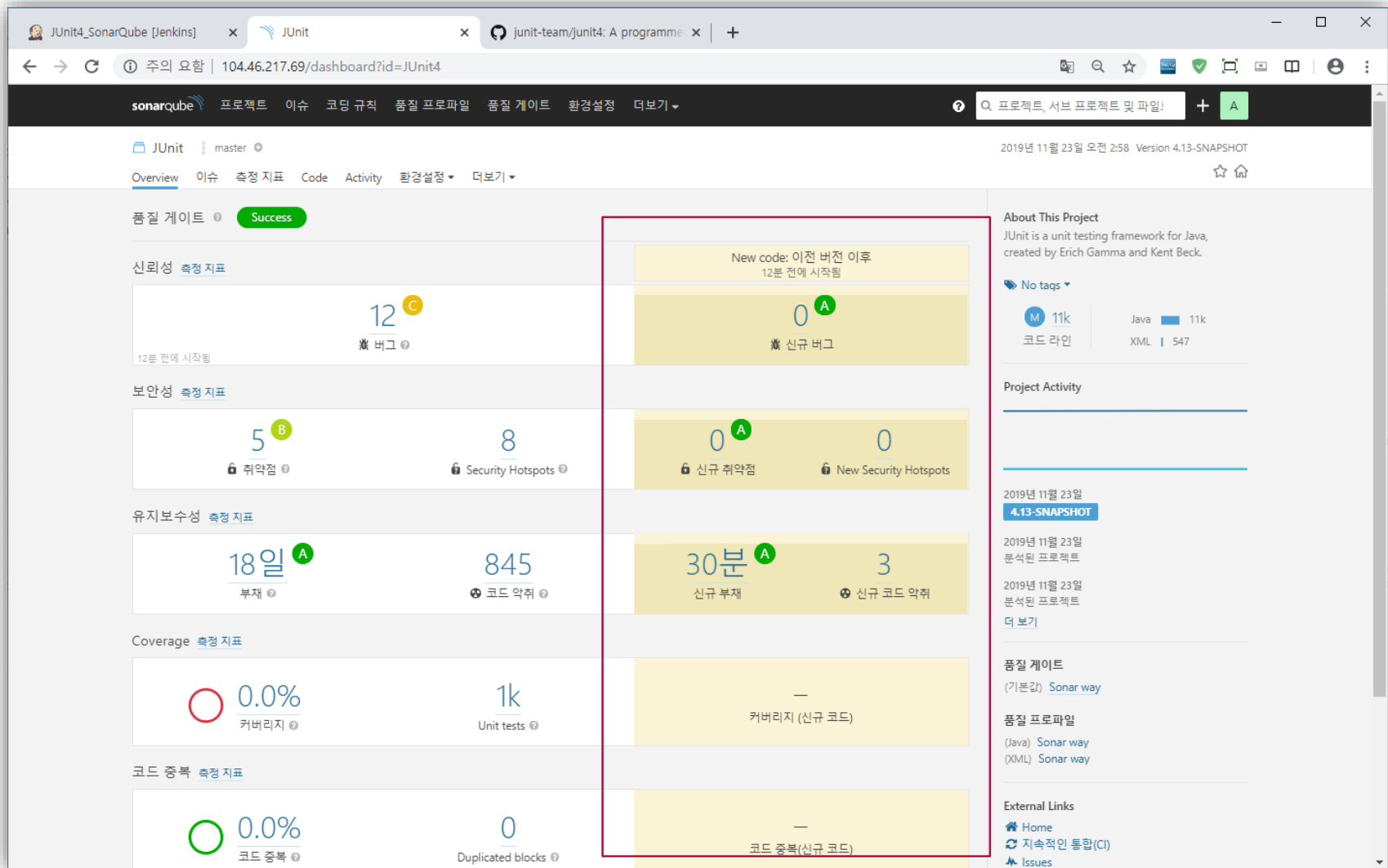
[실습] JUnit4의 분석 실행

❑ 프로젝트 Main 화면



[실습] JUnit4의 분석 실행

□ 지난 분석과의 비교 (2번째 분석부터 표시)



[실습] JUnit4의 분석 실행

❑ 룰 위반 항목의 상세 보기

JUnit4_SonarQube [Jenkins] x A "NullPointerException" could x junit-team/junit4: A programme x +

← → ↺ ① 주의 요약 | 104.46.217.69/project/issues?id=JUnit4&open=AW6UO4IGbA2TWn4WKorT&resolved=false&types=BUG

sonarqube 프로젝트 이슈 코딩 규칙 품질 프로파일 품질 게이트 환경설정 더보기

JUnit master 2019년 11월 23일 오전 2:58 Version 4.13-SNAPSHOT

Overview 이슈 측정 지표 Code Activity 환경설정 더보기

← 2 / 12 issues ↺

src/.../junit/extensions/ActiveTestSuite.java

Either re-interrupt this method or rethrow the "InterruptedException".
Bug

src/.../java/junit/framework/TestCase.java

A "NullPointerException" could be thrown; "runMethod" is nullable here.
Bug +4

- 1 Implies 'runMethod' is null.
- 2 'NoSuchMethodException' is thrown.
- 3 'NoSuchMethodException' is caught.
- 4 'runMethod' is dereferenced.

alt + ⌘ to navigate issue locations

src/.../junit/experimental/ParallelComputer.java

src/main/java/junit/framework/TestCase.java

```
160 egamma protected void runTest() throws Throwable {
161 dsaff    assertNotNull("TestCase.fName cannot be null", fName); // Some VMs crash when calling getMethod(null,null);
162 egamma    1 Method runMethod = null;
163          try {
164              // use getMethod to get all public inherited
165              // methods. getDeclaredMethods returns all
166              // methods of this class but excludes the
167              // inherited ones.
168 egamma    runMethod = 2 getClass().getMethod(fName, (Class[]) null);
169 egamma    } catch ( 3 NoSuchMethodException e) {
170          fail("Method \" + fName + "\" not found");
171          }
172 egamma    if (!Modifier.isPublic( 4 runMethod.getModifiers())) {
173          fail("Method \" + fName + "\" should be public");
174          }
175          }
```

A "NullPointerException" could be thrown; "runMethod" is nullable here. See Rule 15년 전 L172 cert, cwe

Bug Major Open 할당되지 않음 10min effort 코멘트

Null pointers should not be dereferenced squid:S2259

Bug Major Main sources cert, cwe 다음 기간 이후 적용됨 2019년 9월 20일 SonarAnalyzer (Java) 상수/이슈: 10min

A reference to `null` should never be dereferenced/accessed. Doing so will cause a `NullPointerException` to be thrown. At best, such an exception will cause abrupt program termination. At worst, it could expose debugging information that would be useful to an attacker, or it could allow an attacker to bypass security measures.

Note that when they are present, this rule takes advantage of `@CheckForNull` and `@Nonnull` annotations defined in JSR-305 to understand which values are and are not nullable except when `@Nonnull` is used on the parameter to `equals`, which by contract should always work with `null`.

Noncompliant Code Example

```
@CheckForNull
String getName(){...}
```

07 분석 실행 – JS/PHP/Python의 경우

SonarScanner 다운로드

❑ SonarScanner란?

- SonarQube의 기본 분석기
- 빌드를 Hook 형식으로 분석하는 C/C++/C# 등을 제외한 다른 언어에 적용
 - PHP, JS, Python, TS, Ruby 등
- OS 별 실행파일 제공

❑ 동작 방식

- SonarScanner에서 분석을 실행
- 결과를 SonarQube 서버에 지정된 프로젝트로 전송
- SonarQube는 해당 내용을 분석하여 표시

2 프로젝트 분석 실행하기

프로젝트는 주로 어떤 언어로 구현되어 있습니까?

Java

C# 혹은 VB.NET

기타 (JS, Python, PHP, ...)

운영체제는 무엇을 사용하십니까?

Linux

Windows

macOS

Windows 용 SonarQube Scanner를 다운로드하고 압축을 풀니다

And add the `bin` directory to the `%PATH%` environment variable

다운로드

분석 실행

□ 분석 실행 명령 예제

- 프로젝트 root 폴더에서 실행
- 예제: `sonar-scanner.bat -D"sonar.projectKey=JSProject" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=1345f1196ab3ae5d70d5aa0a46b0c851dc349951"`

[실습] JS 프로젝트 다운로드

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

hagopj13 / node-express-boilerplate Template

Notifications Star 1.6k Fork 399

<> Code Issues 3 Pull requests Discussions Actions Projects Security Insights

master 1 branch 14 tags Go to file Code

hagopj13 Merge pull request #87 from iCherya/patch-1 6c88d7f Mar 31, 2021 214 commits

.husky	Upgrade dependencies	Mar 1, 2021
bin	Upgrade dependencies	Mar 1, 2021
src	Remove default values from jwt expiration configs	Mar 30, 2021
tests	Fix verify email endpoints	Mar 30, 2021
.dockerignore	Add docker support	Nov 22, 2019
.editorconfig	Add .editorconfig	Oct 25, 2019
.env.example	Fixed typo	Mar 31, 2021
.eslintignore	Add script to create app using npm init	Dec 25, 2020
.eslintrc.json	Do formatting fixes	May 12, 2020
.gitattributes	Add .gitignore and .gitattributes	Oct 25, 2019
.gitignore	Add script to create app using npm init	Dec 25, 2020
.lintstagedrc.json	Add husky and lint-staged	Oct 25, 2019
.prettiignore	exclude coverage directory when running prettier #20	Jul 6, 2020

About

A boilerplate for building production-ready RESTful APIs using Node.js, Express, and Mongoose

nodejs boilerplate express mongodb es6 jest mongoose rest-api starter express-boilerplate node-boilerplate es2018

Readme MIT License

Releases 14

v1.7.0 Latest Mar 30, 2021 + 13 releases

Packages

[실습] SonarQube 프로젝트 생성

The screenshot shows the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is on the right. The main content area is for the project 'JSProject' (master branch). The 'Overview' tab is selected, showing a step-by-step guide to run the analysis. The first step is 'Run analysis on your project'. It asks for the build system (Maven, Gradle, .NET, or Other) and the operating system (Linux, Windows, or macOS). It then provides instructions to download the scanner for Windows and add the bin directory to the PATH environment variable. A 'Download' button is provided. Next, it instructs to execute the scanner from the computer, showing a command to run in a terminal. A 'Copy' button is next to the command. Finally, it provides a link to the official documentation and states that the page will refresh after the analysis is complete.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration ? Search for projects... A

JSProject ☆ master +

Overview Issues Security Hotspots Measures Code Activity Project Settings ▾ Project Information

2 Run analysis on your project

What option best describes your build?

Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)

What is your OS?

Linux Windows macOS

Download and unzip the Scanner for Windows

And add the `bin` directory to the `%PATH%` environment variable

Download

Execute the Scanner from your computer

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner.bat -D"sonar.projectKey=JSProject" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=1345f1196
```

< > Copy

Please visit the [official documentation of the Scanner](#) for more details.

Once the analysis is completed, this page will automatically refresh and you will be able to browse the analysis results.

[실습] SonarScanner 실행

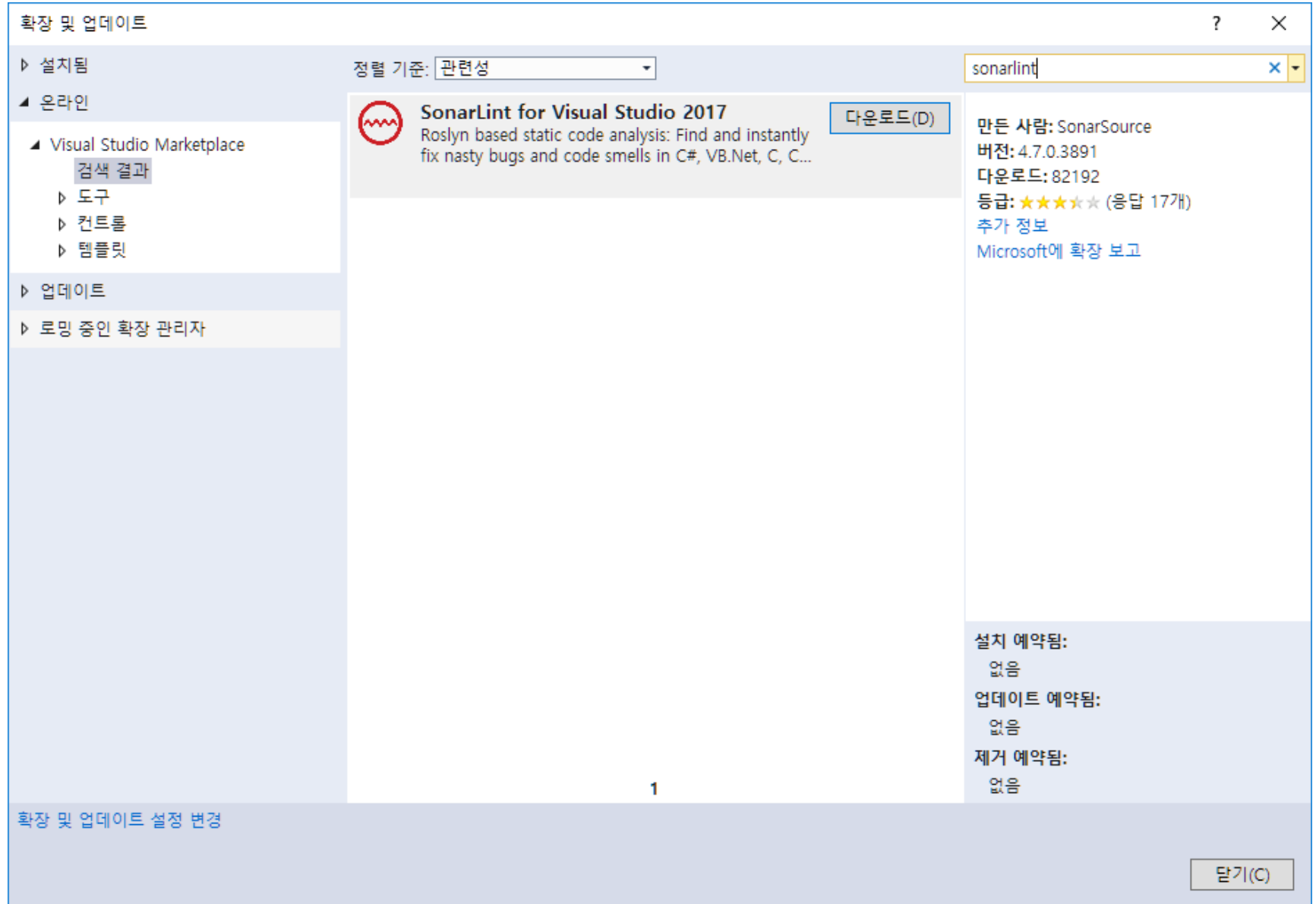
□ 프로젝트 root 폴더에서 실행

```
선택 C:\Windows\system32\cmd.exe - sonar-scanner.bat -D"sonar.projectKey=JSProject" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=1345f1196ab3ae5d70d5aa0a46b0c851dc349951"

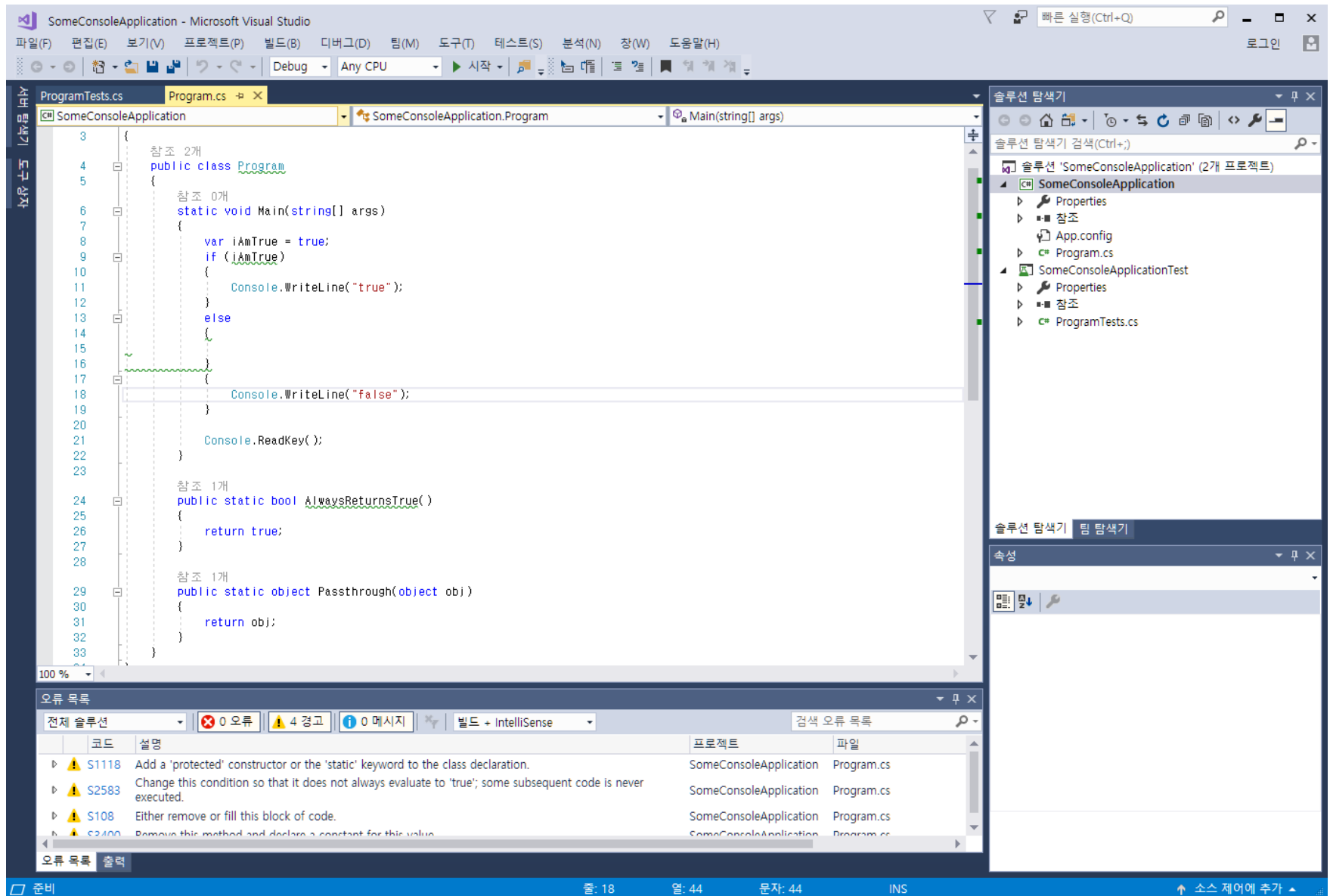
C:\sourcecode\node-express-boilerplate-master>sonar-scanner.bat -D"sonar.projectKey=JSProject" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.login=1345f1196ab3ae5d70d5aa0a46b0c851dc349951"
INFO: Scanner configuration file: C:\sonar-scanner\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.6.0.2311
INFO: Java 11.0.3 AdoptOpenJDK (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\aaaaa\sonar\cache
INFO: Scanner configuration file: C:\sonar-scanner\bin\..\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: Analyzing on SonarQube server 8.8.0
INFO: Default locale: "ko_KR", source code encoding: "x-windows-949" (analysis is platform dependent)
INFO: Load global settings
INFO: Load global settings (done) | time=46ms
INFO: Server id: BF41A1F2-AXistd77oWq1ps9UzqKc
INFO: User cache: C:\Users\aaaaa\sonar\cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=22ms
```

08 주요 IDE 별 sonarlint 설치 방법

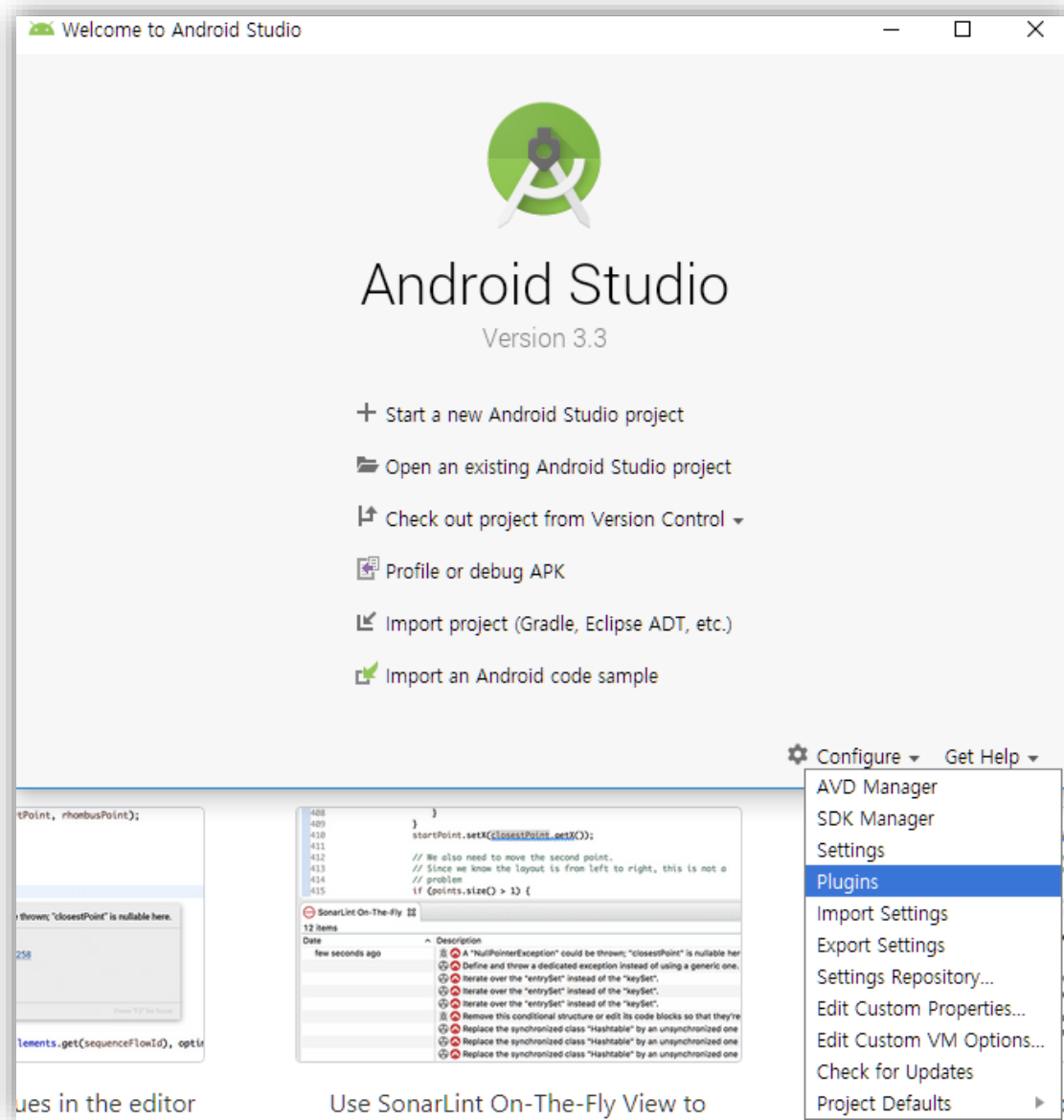
[VS2017] 설치 방법



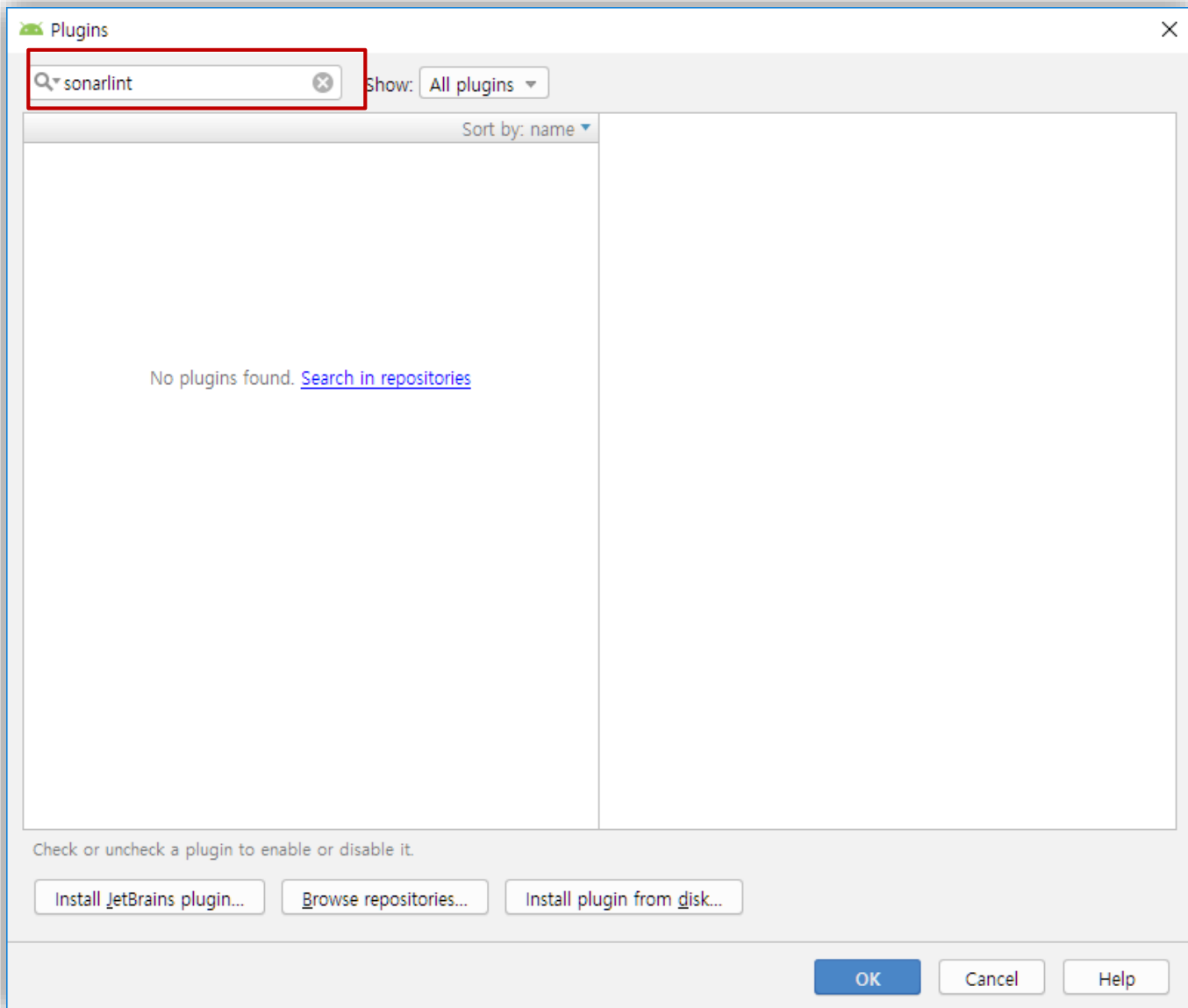
[VS2017] 실행 방법



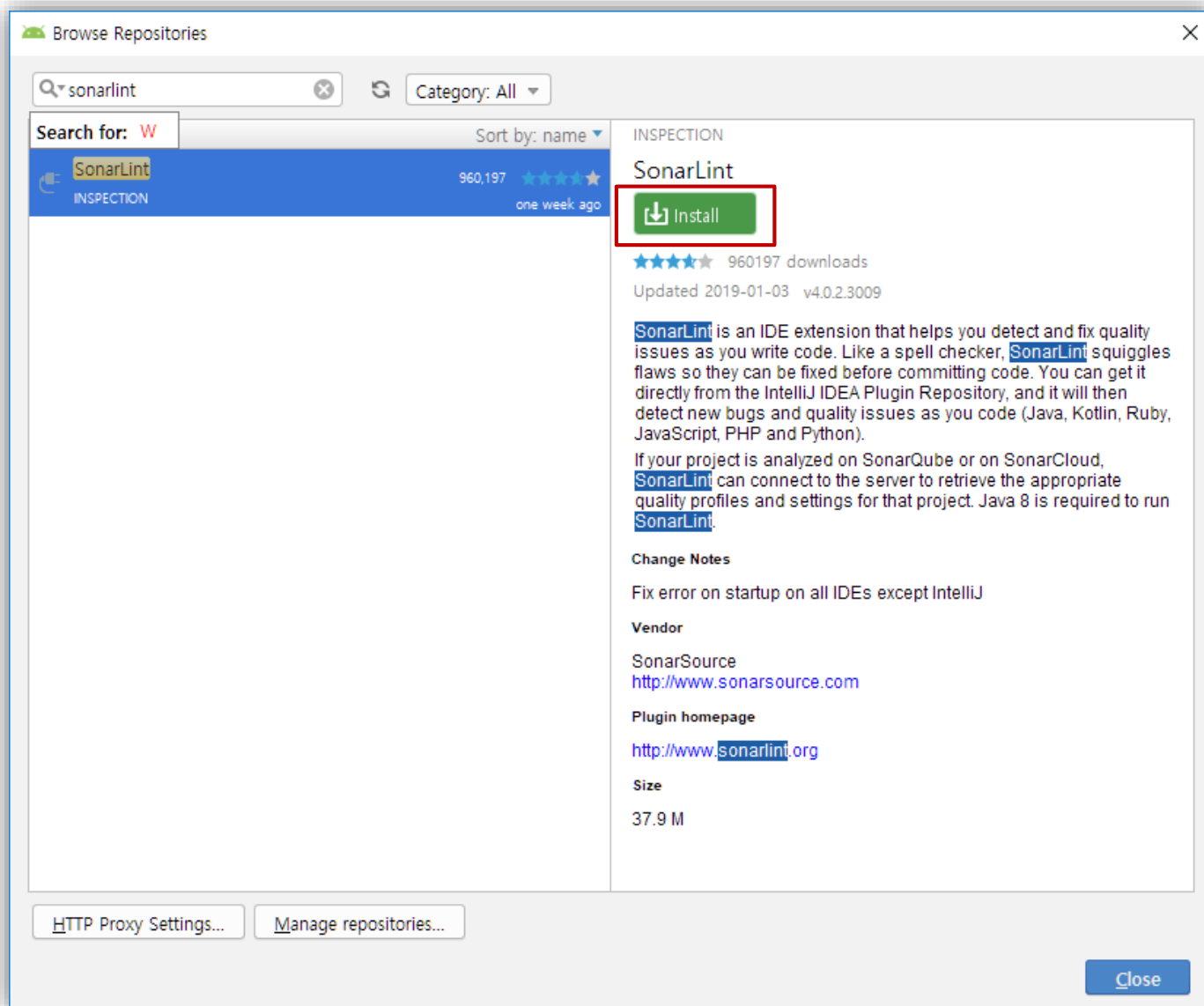
[Android Studio] 설치 방법



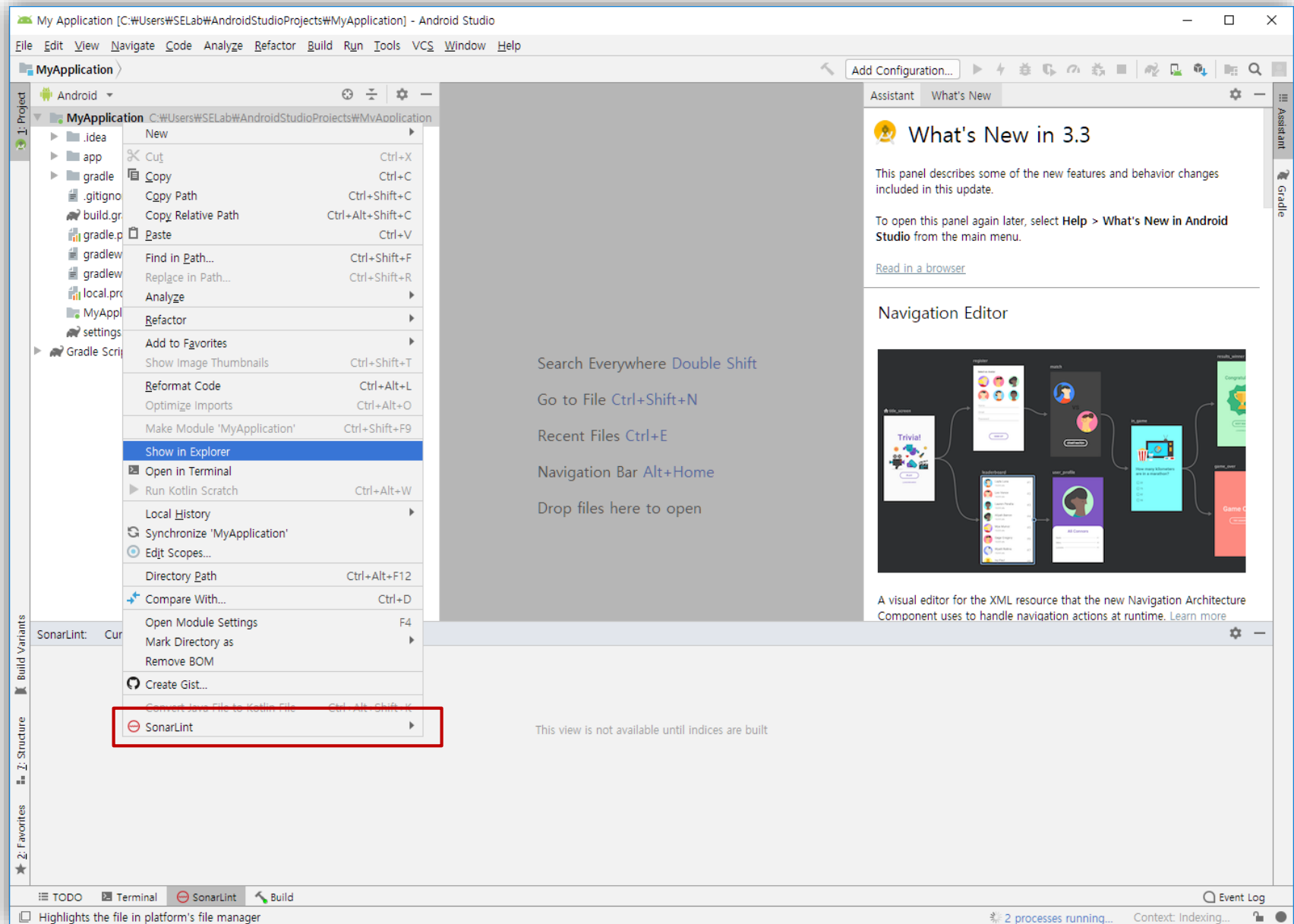
[Android Studio] 설치 방법



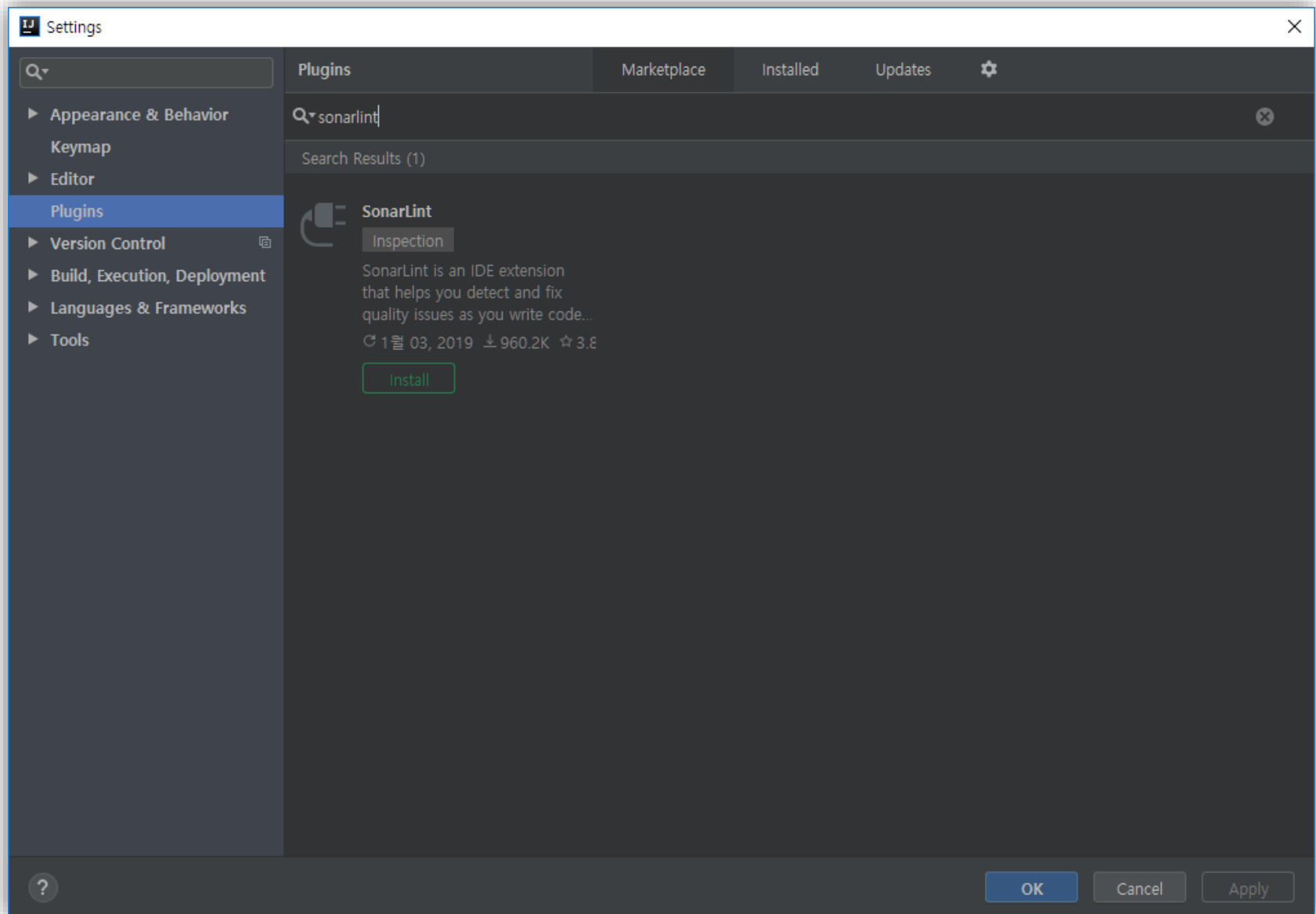
[Android Studio] 설치 방법



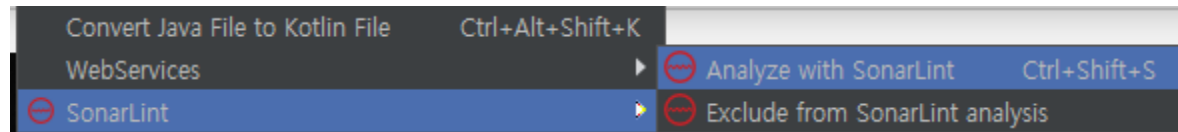
[Android Studio] 실행 방법



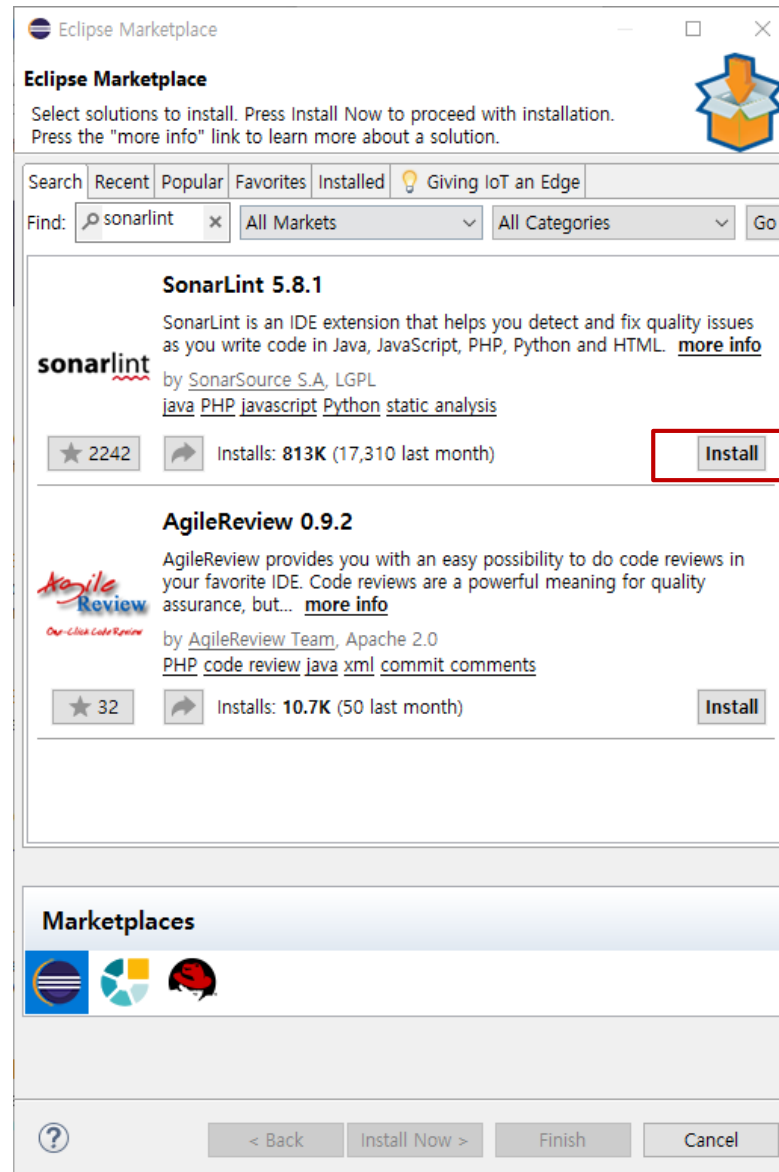
[Intelli J] 설치 방법



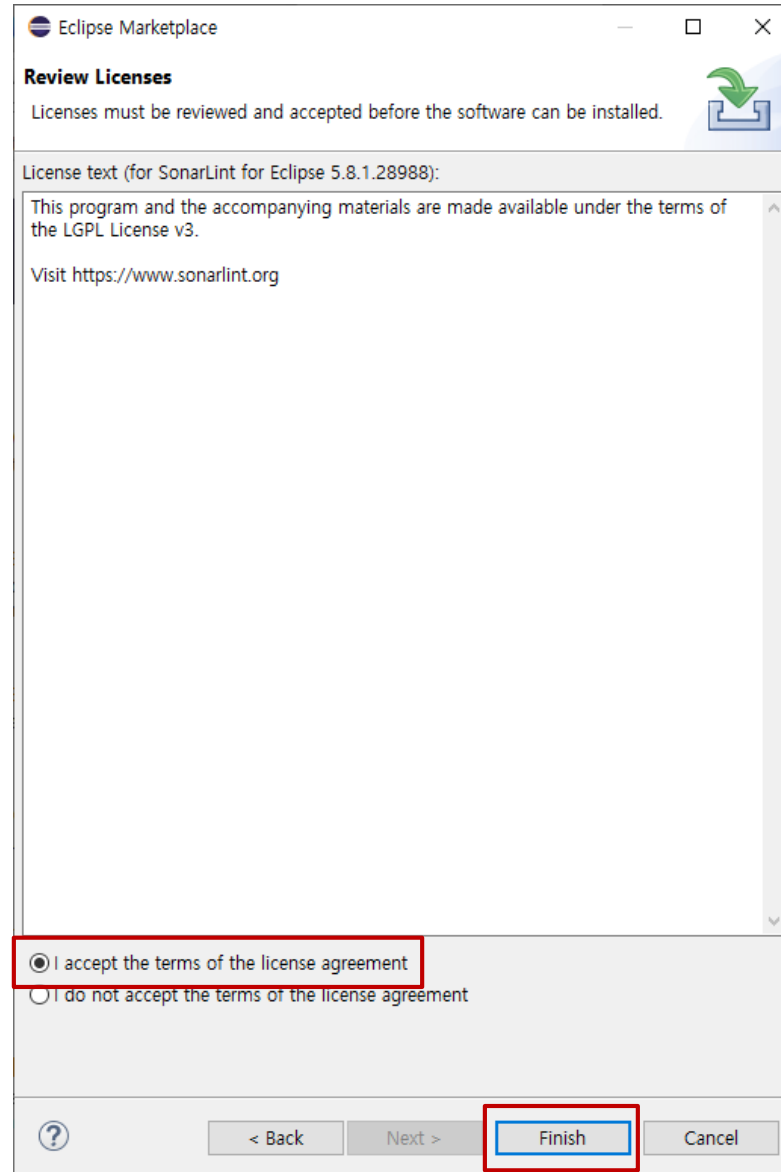
[Intelli J] 실행 방법



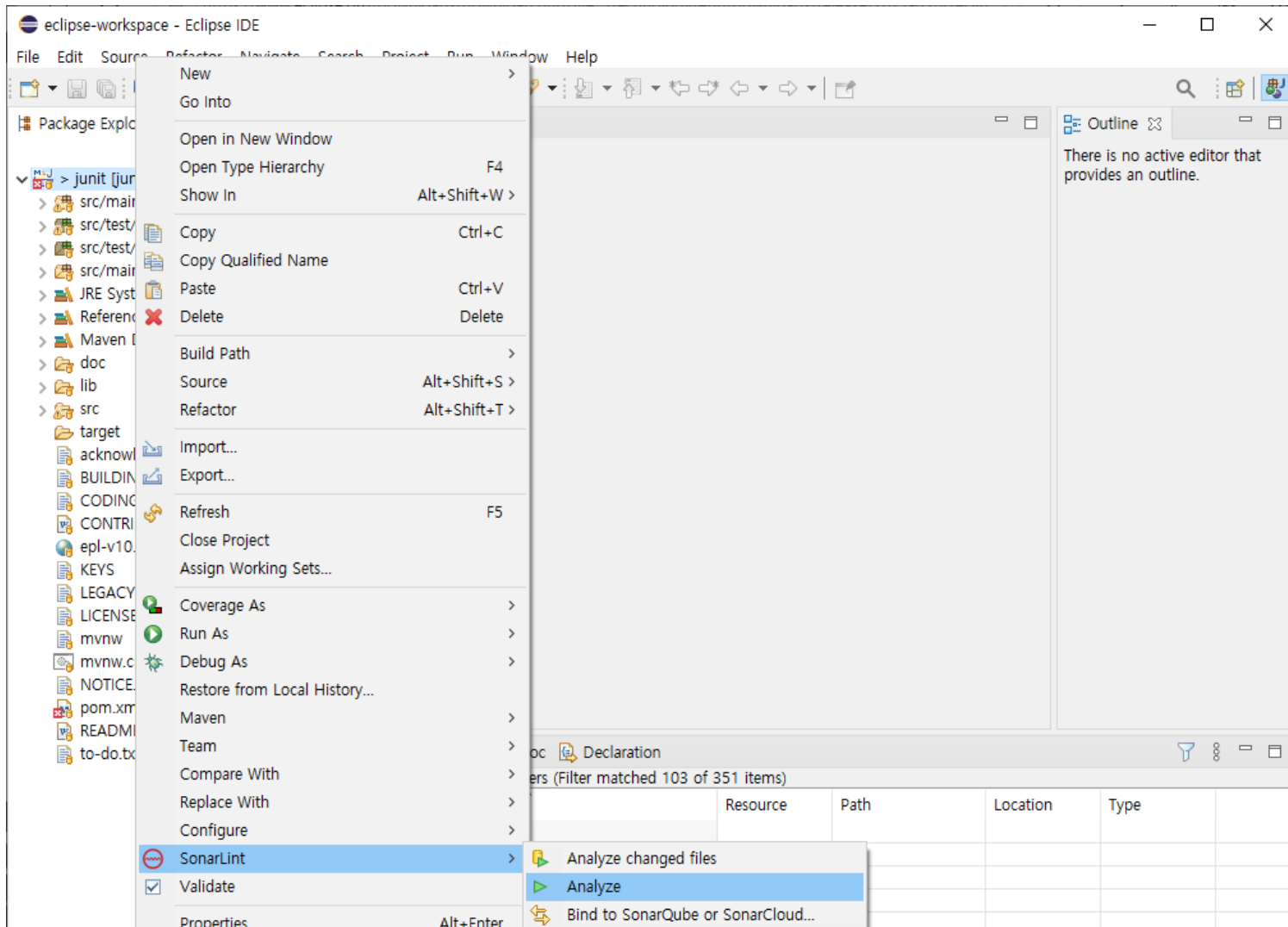
[Eclipse] 설치 방법



[Eclipse] 설치 방법



[Eclipse] 실행 방법



[Eclipse] 실행 결과 확인

Problems @ Javadoc Declaration SonarLint Bindings SonarLint Report			
Filter matched 100 of 1145 items			
Resource	Date	Description	
ActiveTestSu	few seconds ago	Either re-interrupt this method or rethrow the "InterruptedException" that can be caught	
ActiveTestSu	few seconds ago	This block of commented-out lines of code should be removed.	
AllDefaultPo	few seconds ago	Do not forget to remove this deprecated code someday.	
AllMembersS	few seconds ago	Replace the type specification in this constructor call with the diamond operator ("<>").	
AllMembersS	few seconds ago	Replace the type specification in this constructor call with the diamond operator ("<>").	
AllMembersS	few seconds ago	Replace the type specification in this constructor call with the diamond operator ("<>").	
AllMembersS	few seconds ago	Define and throw a dedicated exception instead of using a generic one.	
AllMembersS	few seconds ago	Define and throw a dedicated exception instead of using a generic one.	
AllMembersS	few seconds ago	Define and throw a dedicated exception instead of using a generic one.	
AllMembersS	few seconds ago	Define and throw a dedicated exception instead of using a generic one.	
AllMembersS	few seconds ago	Remove this unused method parameter "sig". [+1 location]	
AllMembersS	few seconds ago	Remove this unused method parameter "sig". [+1 location]	
AllMembersS	few seconds ago	Remove this unused method parameter "sig". [+1 location]	
AllMembersS	few seconds ago	Remove this unused method parameter "sig". [+1 location]	
AllTests.java	few seconds ago	Add some tests to this class.	
AllTests.java	few seconds ago	Add some tests to this class.	
AllTests.java	few seconds ago	Add some tests to this class.	
AllTests.java	few seconds ago	Add some tests to this class.	
AllTestsTest.j	few seconds ago	Add at least one assertion to this test case.	
AllTestsTest.j	few seconds ago	Add at least one assertion to this test case.	
AnnotatedBu	few seconds ago	Define and throw a dedicated exception instead of using a generic one.	
AnnotatedBu	few seconds ago	Add at least one assertion to this test case.	
AnnotatedBu	few seconds ago	Add at least one assertion to this test case.	

[Eclipse] SonarQube 연동

□ 의미

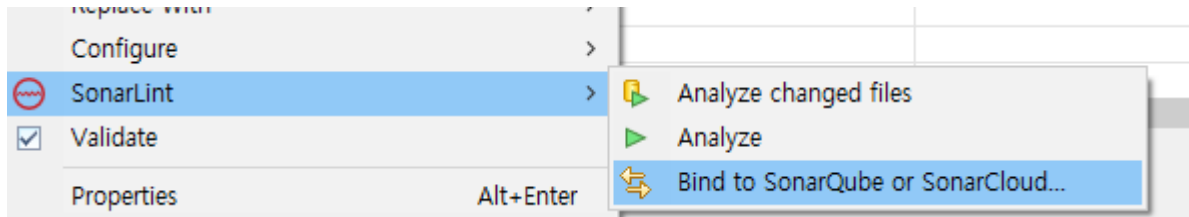
- SonarQube에 등록한 프로젝트의 룰 설정과 Eclipse의 룰 설정을 동기화

□ 필요성

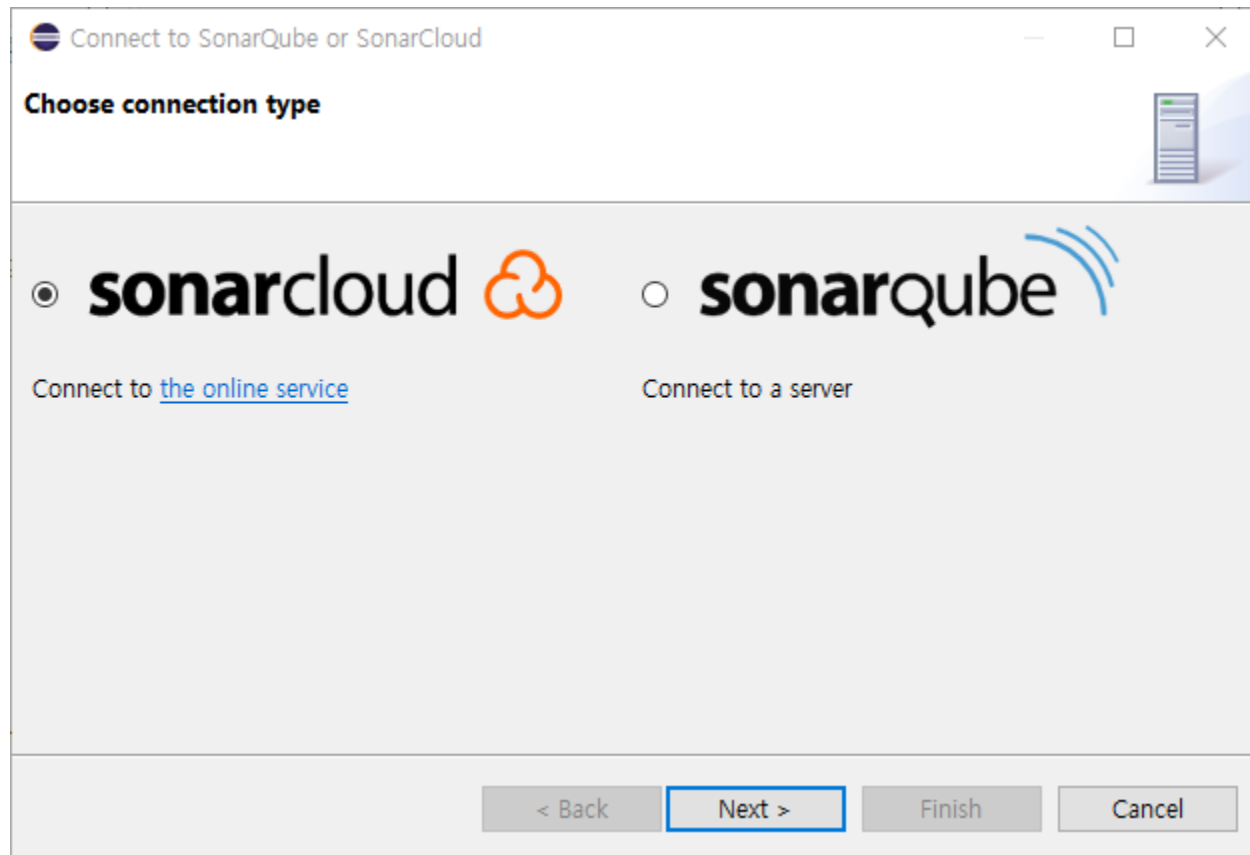
- SonarQube 서버와 Eclipse의 룰 위반을 동일하게 표시
- 당연한 이야기지만, 그렇지 못한 도구도 있음
 - PMD: Maven 실행 결과(Jenkins 실행 결과)와 Eclipse 실행 결과가 상이 (기본 룰 실행 시)

□ 연동 방법

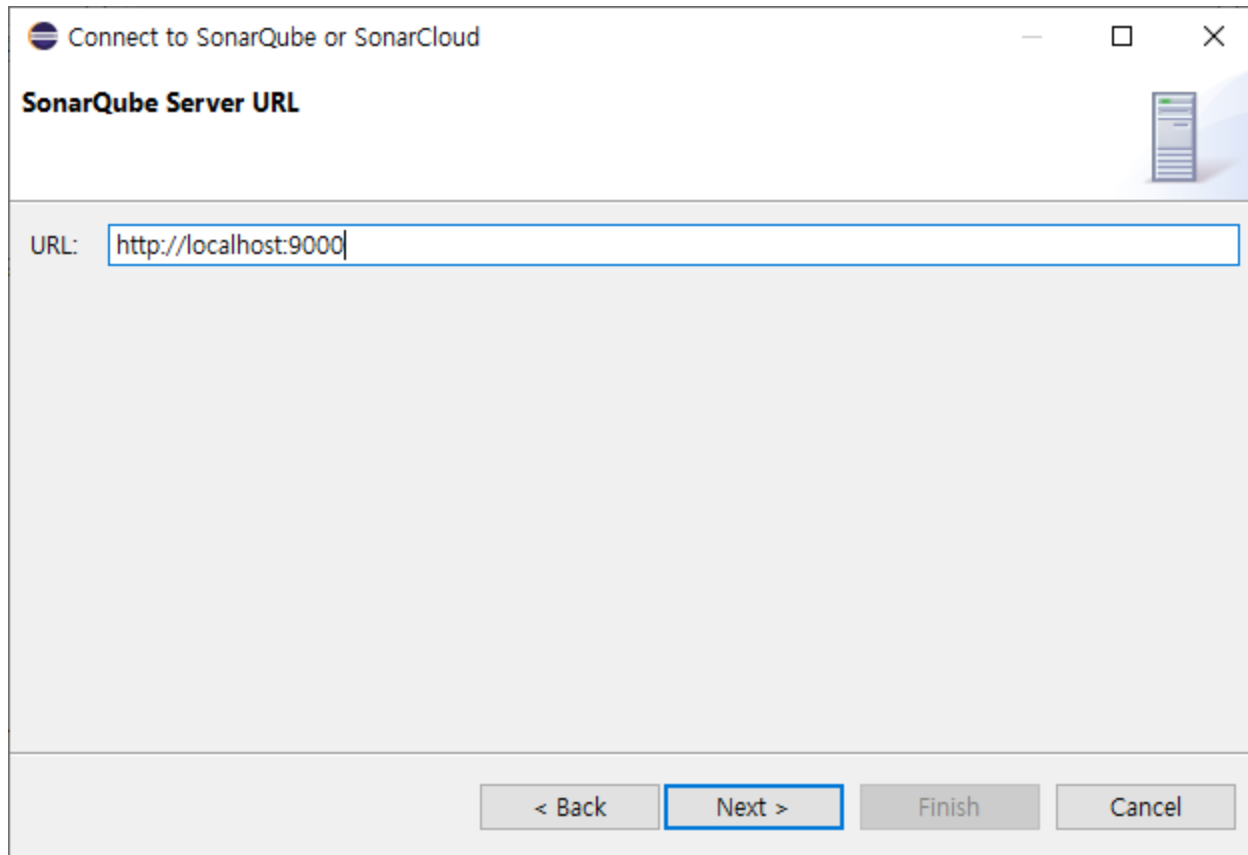
- Bind 기능 이용



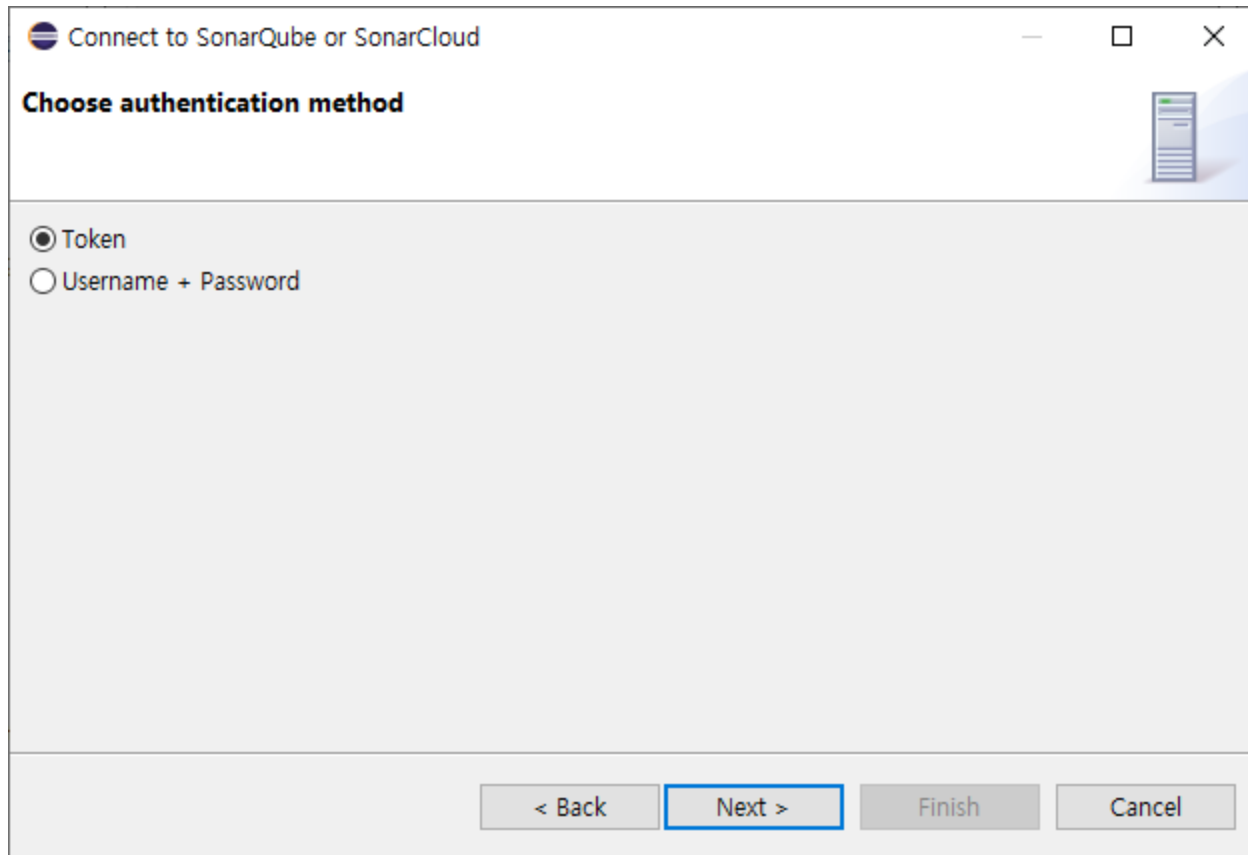
[Eclipse] SonarQube 연동



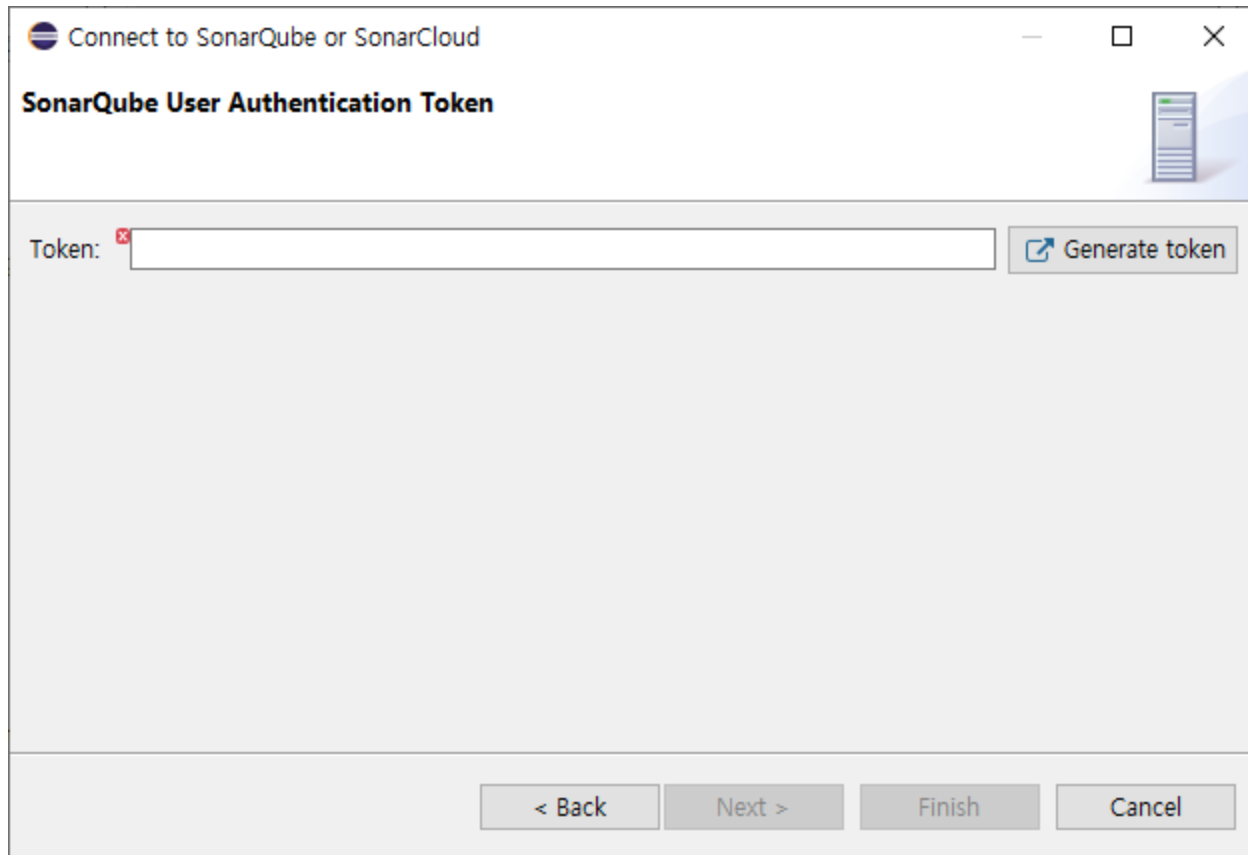
[Eclipse] SonarQube 연동



[Eclipse] SonarQube 연동



[Eclipse] SonarQube 연동



The image shows a dialog box titled "Connect to SonarQube or SonarCloud" with a subtitle "SonarQube User Authentication Token". The dialog box has a standard Windows window frame with minimize, maximize, and close buttons. On the right side, there is a small icon of a server rack. The main area of the dialog box contains a text input field labeled "Token:" with a red 'x' icon to its left. To the right of the input field is a button labeled "Generate token" with a blue icon of a document with a checkmark. At the bottom of the dialog box, there are four buttons: "< Back", "Next >", "Finish", and "Cancel".

Connect to SonarQube or SonarCloud

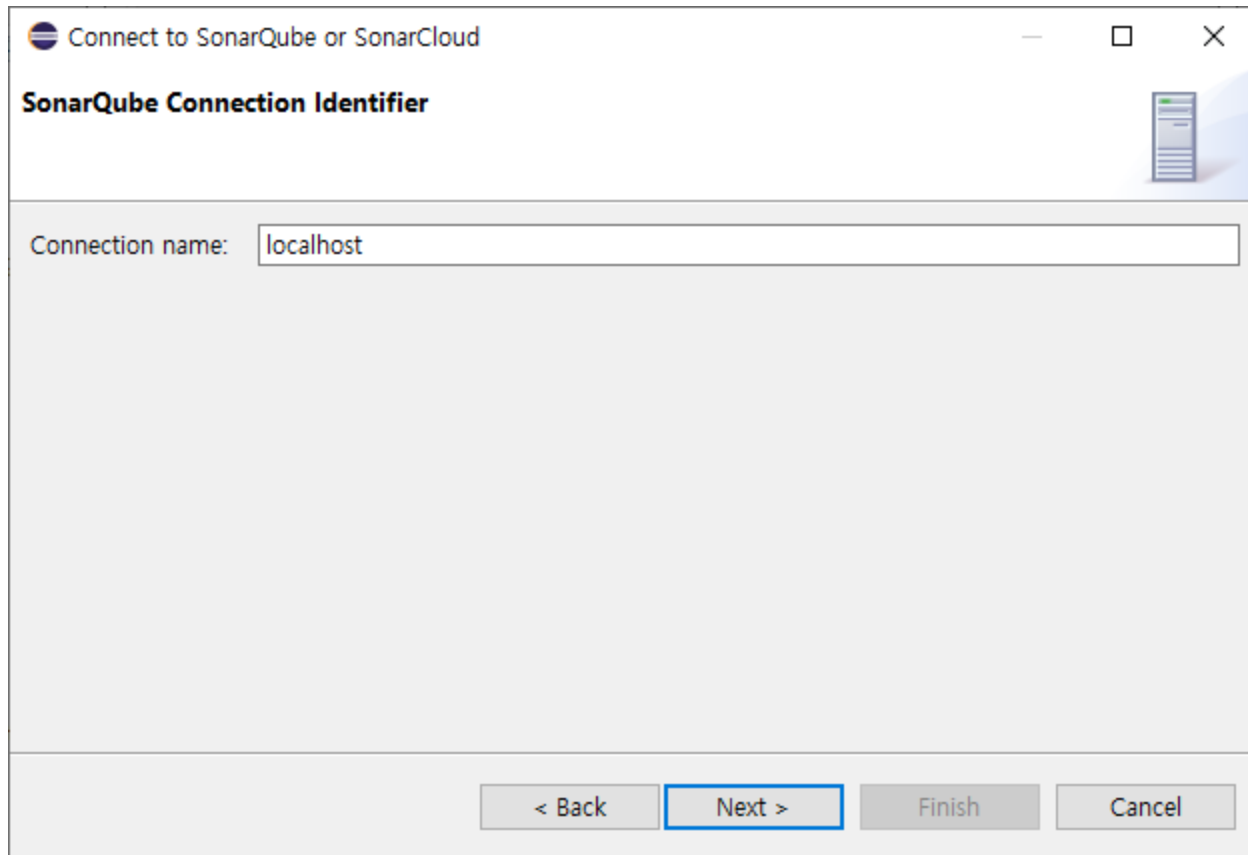
SonarQube User Authentication Token

Token:

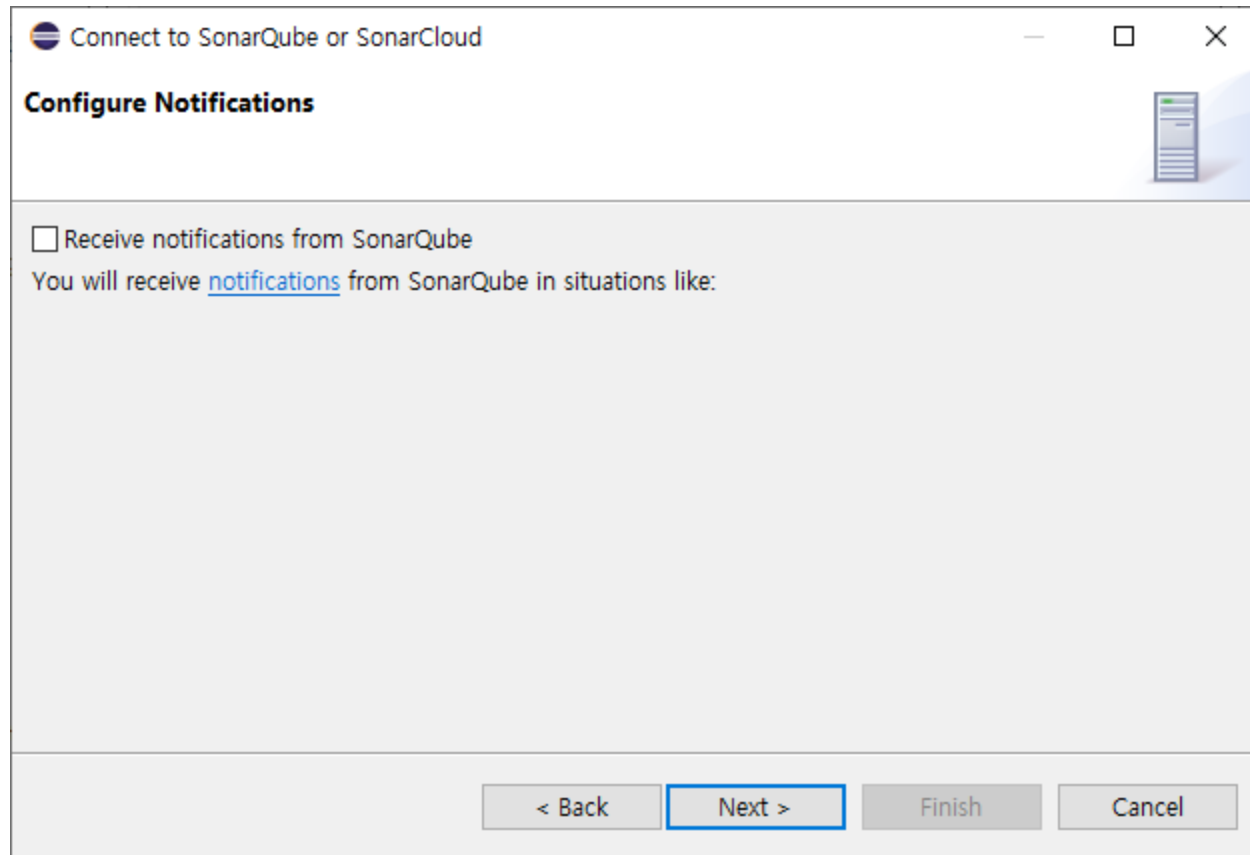
[Generate token](#)

< Back Next > Finish Cancel

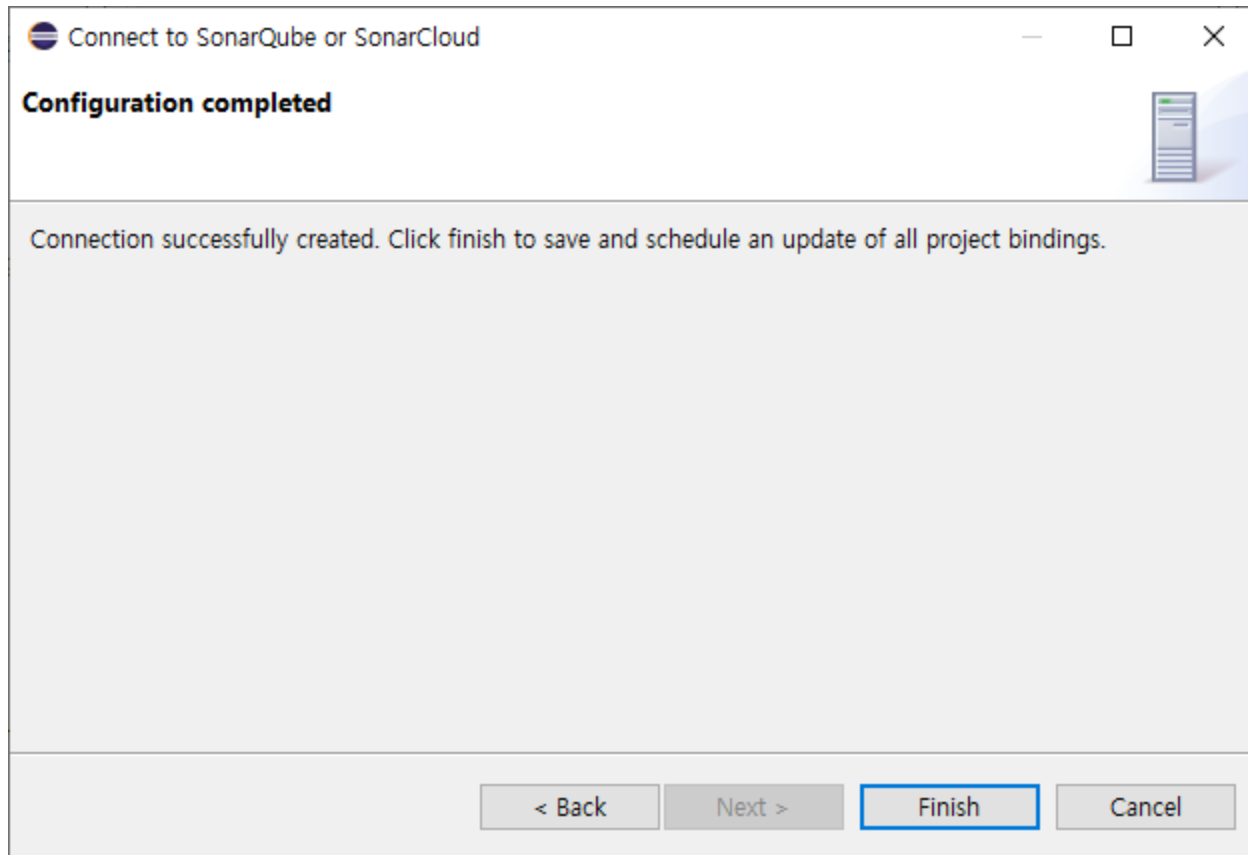
[Eclipse] SonarQube 연동



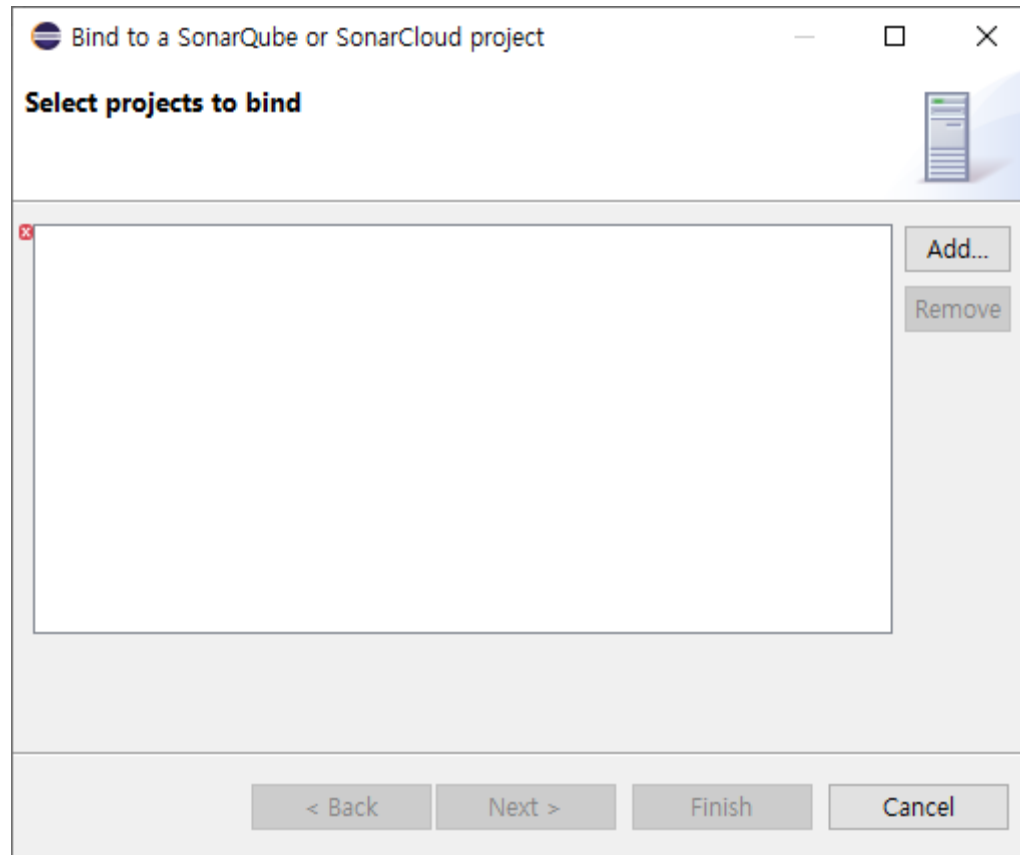
[Eclipse] SonarQube 연동



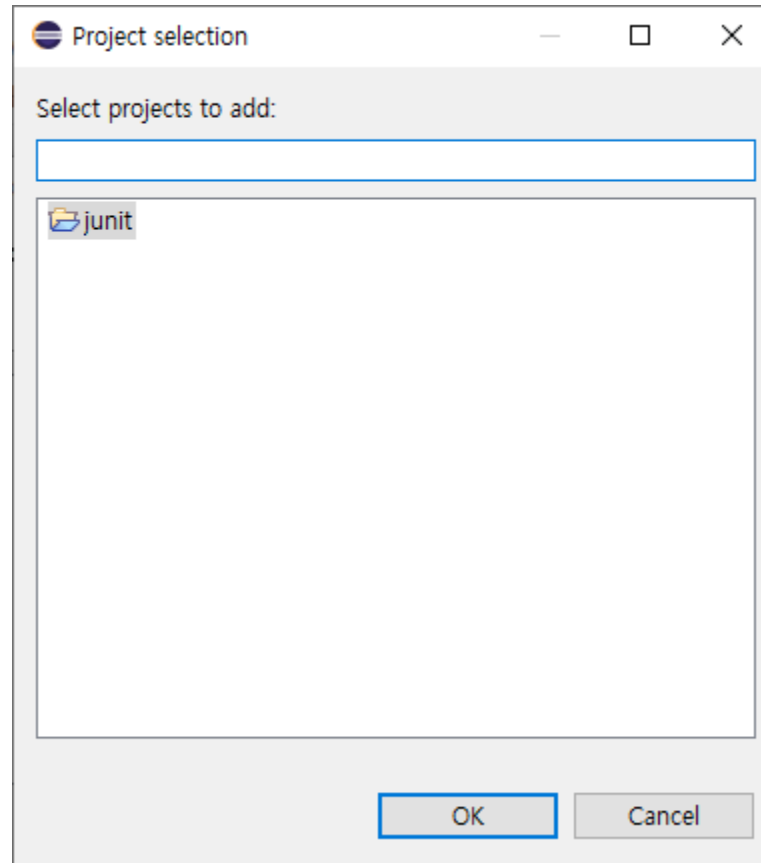
[Eclipse] SonarQube 연동



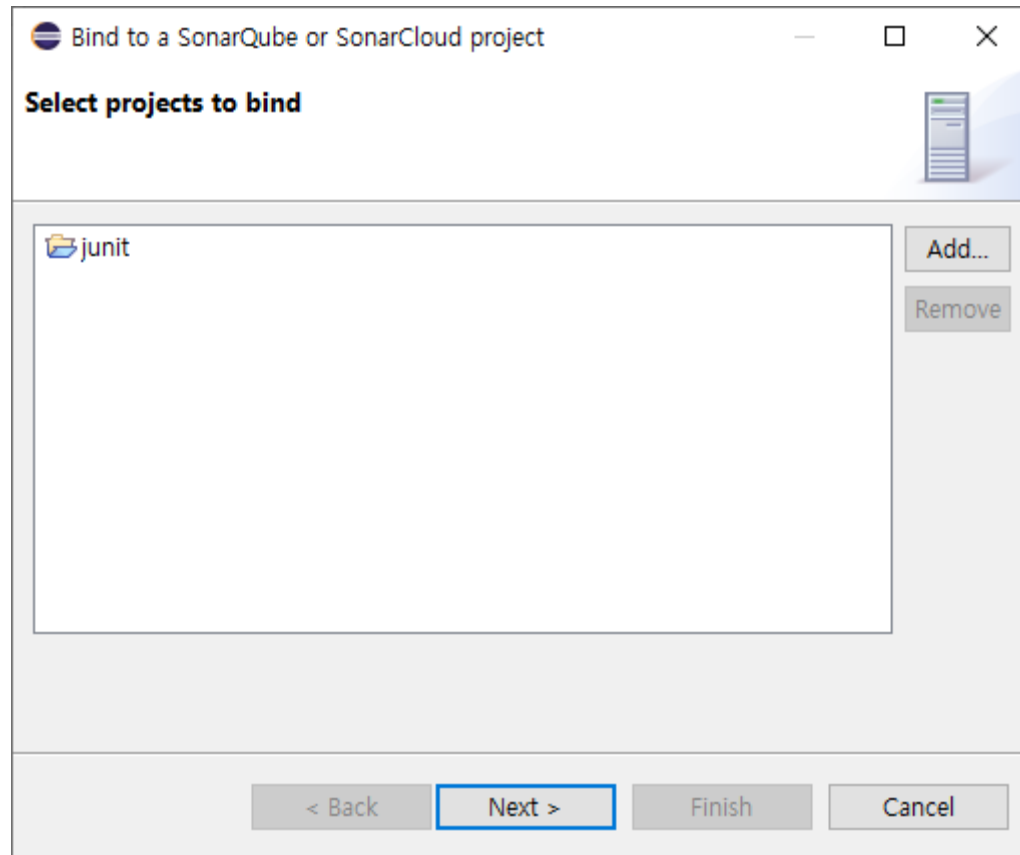
[Eclipse] SonarQube 연동



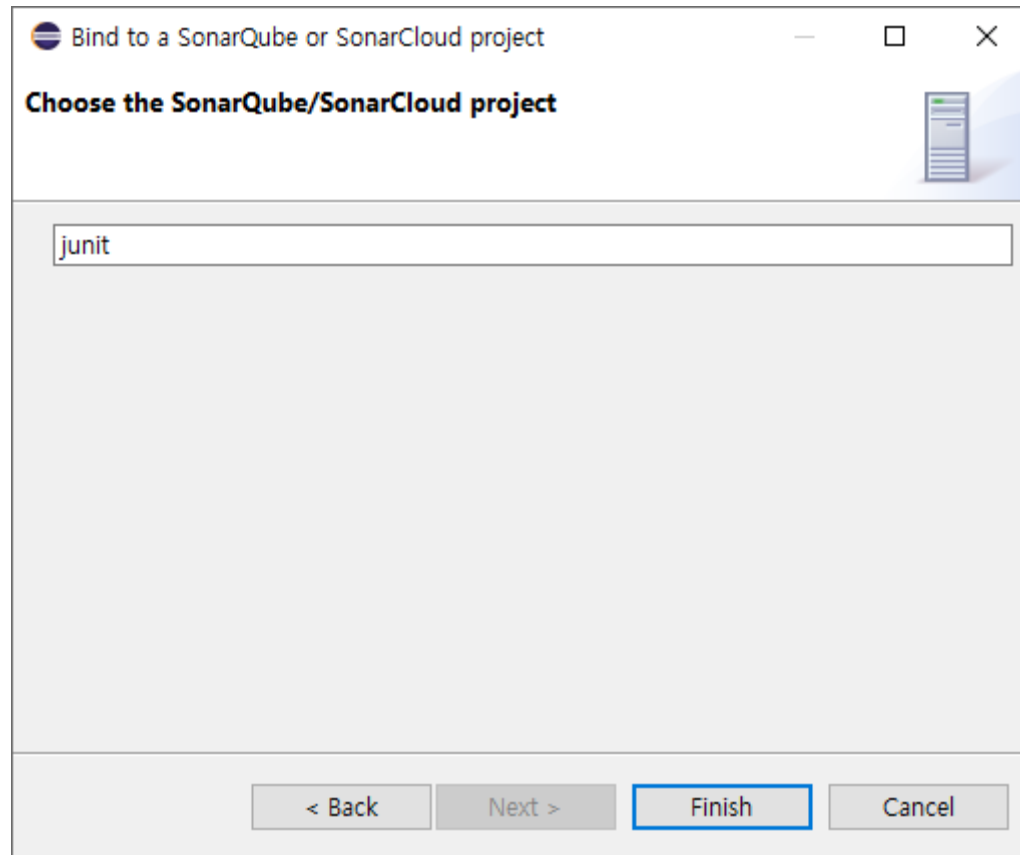
[Eclipse] SonarQube 연동



[Eclipse] SonarQube 연동



[Eclipse] SonarQube 연동



08 SonarQube와 Jenkins 연동



Jenkins와 SonarQube

□ 동작 방식

- Jenkins 빌드 시점 도래
- SonarScanner 실행
- 분석 결과 SonarQube 서버 등록
- Jenkins Job에 SonarQube 프로젝트 링크

SonarQube 프로젝트 링크

SonarScanner



Jenkins

sonarqube 

SonarQube 플러그인 설치

The screenshot shows the Jenkins Update Center interface. The browser address bar indicates the URL is `52.141.2.121:8080/jenkins/pluginManager/available`. The Jenkins logo and name are at the top left. A search bar on the right contains the text "sonarqube". Below the search bar, there are tabs for "업데이트된 플러그인 목록", "설치 가능", "설치된 플러그인 목록", and "고급". The "설치 가능" tab is selected. A table lists available plugins with columns for "설치" (Install), "이름" (Name), and "버전" (Version). The "SonarQube Scanner" plugin is highlighted with a checkmark in the "설치" column. Below the table, there are buttons for "재시작 없이 설치하기", "지금 다운로드하고 재시작 후 설치하기", and "지금 확인". A warning message is displayed for the "Sonar Gerrit" plugin, stating "Warning: This plugin version may not be safe to use. Please review the following security notices: Credentials stored in plain text". The footer of the page shows the page generation time and version information.

Update Center [Jenkins] x +

← → ↻ 주의 요함 | 52.141.2.121:8080/jenkins/pluginManager/available

Jenkins 1 s ADMIN | 로그아웃

Jenkins > Plugin Manager

← 대시보드로 돌아가기

Jenkins 관리

Update Center

필터: sonarqube

업데이트된 플러그인 목록 설치 가능 설치된 플러그인 목록 고급

설치 ↓	이름	버전
<input checked="" type="checkbox"/>	SonarQube Scanner This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality.	2.10
<input type="checkbox"/>	Mashup Portlets Additional Dashboard Portlets: Generic JS Portlet (lets you pull in arbitrary content via JS), Recent Changes Portlet (shows the SCM changes for a given job), SonarQube Portlets (show SonarQube statistics directly in Jenkins) and Test Results Portlet (shows the test results for a given job).	1.1.0
<input type="checkbox"/>	Sonar Gerrit This plugin allows to submit issues from SonarQube to Gerrit as comments directly. Warning: This plugin version may not be safe to use. Please review the following security notices: <ul style="list-style-type: none">Credentials stored in plain text	2.3
<input type="checkbox"/>	SonarQube Generic Coverage TODO	1.0

재시작 없이 설치하기

지금 다운로드하고 재시작 후 설치하기

Update information obtained: 42 min ago

지금 확인

페이지 생성일시: 2019. 11. 22 오후 6시 27분 18초 REST API Jenkins ver. 2.190.2 afonsof.com/jenkins-material-theme v. 1.3.3

Jenkins 시스템 설정

❑ SonarQube 서버 기본 정보 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

Synetics_SonarQube


Server URL

http://138.91.14.94:9000

Default is http://localhost:9000

Server authentication token

- none -

 ADD

SonarQube authentication token. Mandatory when anonymous access is disabled.

고급...

DELETE SONARQUBE

ADD SONARQUBE

List of SonarQube installations

Global Tool Configuration

❑ SonarScanner 정보 등록

SonarQube Scanner

SonarQube Scanner installations

ADD SONARQUBE SCANNER

SonarQube Scanner

Name

Scanner

☒ Install automatically

?

Install from Maven Central

Version

SonarQube Scanner 4.2.0.1873 ▼

DELETE INSTALLER

ADD INSTALLER ▼

DELETE SONARQUBE SCANNER

ADD SONARQUBE SCANNER

List of SonarQube Scanner installations on this system

Job 설정 (JS 등 일반 프로젝트)

❑ SonarScanner분석 옵션 설정

Build

Execute SonarQube Scanner

Task to run

?

JDK

(Inherit From Job)

▼

?

JDK to be used for this SonarQube analysis

Path to project properties

?

Analysis properties

sonar.projectKey=my:project
sonar.projectName=My project
sonar.projectVersion=1.0

?

Additional arguments

▼

?

JVM Options

▼

?

ADD BUILD STEP ▼

Job 설정 (Maven 프로젝트)

❑ SonarScanner분석 옵션 설정

Invoke top-level Maven targets

Maven Version

Maven362

Goals

clean
package

고급...

Execute SonarQube Scanner

Task to run

JDK

(Inherit From Job)

JDK to be used for this SonarQube analysis

Path to project properties

Analysis properties

sonar.projectKey=JUnit4
sonar.host.url=http://104.46.217.69
sonar.login=088ac21e667933f8d7b5d218af9b1d91b23702f8
sonar.sources=src/main/java
sonar.java.binaries=target/classes

Additional arguments

JVM Options

Job 설정 (Maven 프로젝트)

- ❑ Maven Goal에서 sonar:sonar 제거
- ❑ 소스코드 위치와 바이너리 위치를 추가로 지정해야 함

로컬 컴퓨터에서 SonarQube Scanner for Maven 실행하기

Maven과 함께 SonarQube를 매우 쉽게 실행할 수 있습니다. 프로젝트 폴더에서 다음 명령어를 실행합니다:

```
mvn sonar:sonar \
  -Dsonar.projectKey=JUnit4 \
  -Dsonar.host.url=http://104.46.217.69 \
  -Dsonar.login=088ac21e667933f8d7b5d218af9b1d91b23702f8
```

복사



sonar.projectKey=JUnit4
sonar.host.url=http://104.46.217.69
sonar.login=088ac21e667933f8d7b5d218af9b1d91b23702f8
sonar.sources=src/main/java
sonar.java.binaries=target/classes

Job 화면에서 SonarQube 링크 확인

□ 링크 및 Quality Gate 결과 표시

The screenshot shows the Jenkins web interface for a job named 'JUnit4_SonarQube'. The page is divided into a left sidebar with navigation links and a main content area. In the sidebar, the 'SonarQube' link is highlighted with a red box. The main content area shows the 'Project JUnit4_SonarQube' configuration. A red box highlights the 'SonarQube' icon in the top left of the main area. Another red box highlights the 'SonarQube Quality Gate' section, which displays 'JUnit OK' and 'server-side processing: Success'. A third red box highlights the '고정링크' (Fixed Links) section, which lists various build links. The bottom of the page shows the footer with the page generation time and version information.

Jenkins

1 검색 ADMIN | 로그아웃

Jenkins > JUnit4_SonarQube > 자동으로 새로고침

← 대시보드로 돌아가기

상태

변경사항

작업공간

Build Now

Project 삭제

구성

SonarQube

Rename

Build History 추이 ^

find x

Build	Time	Status
#3	2019. 11. 23 오전 3:07	Success
#2	2019. 11. 23 오전 3:04	Failed
#1	2019. 11. 23 오전 2:59	Failed

RSS (전체) RSS (실패)

Project JUnit4_SonarQube

SonarQube

작업 공간

최근 변경사항

SonarQube Quality Gate

JUnit OK

server-side processing: Success

고정링크

- Last build, (#3), 7 min 20 sec 전
- Last stable build, (#3), 7 min 20 sec 전
- Last successful build, (#3), 7 min 20 sec 전
- Last failed build, (#2), 10 min 전
- Last unsuccessful build, (#2), 10 min 전
- Last completed build, (#3), 7 min 20 sec 전

페이지 생성일시: 2019. 11. 23 오전 3시 14분 50초 REST API Jenkins ver. 2.190.2 afonsof.com/jenkins-material-theme v. 1.3.3

10

[참고: C/C++ 정적분석] MISRA와 CppCheck

일반 Vs. Automotive 정적 분석 비교

구분	일반	Automotive
적용 룰	임베디드: MISRA를 기본으로 자체 룰 생성 (보통 50개 내외) 웹/앱: 오픈소스 제공 룰의 Subset 이용	MISRA C 모든 룰 (150여개)
순환 복잡도	임베디드: 20 이하 웹/앱: 10 이하 (유지보수 관점)	ISO 26262의 요건 15 이하 권고
의존성	임베디드: 제어/데이터 흐름 관리 (항공) 웹/앱: 인터페이스를 이용한 의존성 관리	ISO 26262의 요건 Autosar 기반의 의존성 관리

Table 8 — Design principles for software unit design and implementation

Methods		ASIL			
		A	B	C	D
1a	One entry and one exit point in subprograms and functions ^a	++	++	++	++
1b	No dynamic objects or variables, or else online test during their creation ^{a,b}	+	++	++	++
1c	Initialization of variables	++	++	++	++
1d	No multiple use of variable names ^a	+	++	++	++
1e	Avoid global variables or else justify their usage ^a	+	+	++	++
1f	Limited use of pointers ^a	0	+	+	++
1g	No implicit type conversions ^{a,b}	+	++	++	++
1h	No hidden data flow or control flow ^c	+	++	++	++
1i	No unconditional jumps ^{a,b,c}	++	++	++	++
1j	No recursions	+	+	++	++
^a Methods 1a, 1b, 1d, 1e, 1f, 1g and 1i may not be applicable for graphical modelling notations used in model-based development.					
^b Methods 1g and 1i are not applicable in assembler programming.					
^c Methods 1h and 1i reduce the potential for modelling data flow and control flow through jumps or global variables.					

NOTE For the C language, MISRA C^[3] covers many of the methods listed in Table 8.

MISRA C 2012 룰셋

룰셋#	룰셋명	룰 개수
1	C 표준 (A standard C environment)	3
2	사용하지 않는 코드 (Unused code)	7
3	주석 (Comments)	2
4	문자셋과 어휘 요소 (Character sets and lexical conventions)	2
5	식별자 (Identifiers)	9
6	타입 (Types)	2
7	리터럴과 상수 (Literals and constants)	4
8	선언과 정의 (Declarations and definitions)	14
9	초기화 (Initialization)	5
10	에센셜 타입 모델 (The essential type model)	8
11	포인터 타입 변환 (Pointer type conversions)	9
12	표현식 (Expressions)	5
13	부작용 (Side effects)	6
14	제어문 표현식 (Control statements expressions)	4
15	제어 흐름 (Control flow)	7
16	Switch 문 (Switch statements)	7
17	함수 (Functions)	8
18	포인터와 배열 (Pointers and arrays)	8
19	저장 영역 덮어쓰기 (Overlapping storage)	2
20	전처리 지시자 (Preprocessing directives)	14
21	표준 라이브러리 (Standard libraries)	20
22	자원 (Resources)	10
합계		156

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
1	Required	1. 표준 C 환경	1.1	The program shall contain no violations of the standard C syntax and constraints, and shall not exceed the implementation's translation limits (프로그램은 C 표준과 사용하는 컴파일러의 번역 제한을 준수해야 함)
2	Advisory	1. 표준 C 환경	1.2	Language extensions should not be used (언어 확장을 사용하면 안 됨)
3	Required	1. 표준 C 환경	1.3	There shall be no occurrence of undefined or critical unspecified behaviour (정의되지 않거나 명시되지 않은 행동이 발생하면 안 됨)
4	Required	2. 사용하지 않는 코드	2.1	A project shall not contain unreachable code (도달할 수 없는 코드(unreachable code)가 있으면 안 됨)
5	Required	2. 사용하지 않는 코드	2.2	There shall be no dead code (죽은 코드(dead code)는 없어야 함)
6	Advisory	2. 사용하지 않는 코드	2.3	A project should not contain unused type declarations (사용되지 않은 타입 선언은 없어야 함)
7	Advisory	2. 사용하지 않는 코드	2.4	A project should not contain unused tag declarations (사용되지 않은 tag(struct, union, enum) 선언은 없어야 함)
8	Advisory	2. 사용하지 않는 코드	2.5	A project should not contain unused macro declarations (사용되지 않은 매크로 선언은 없어야 함)
9	Advisory	2. 사용하지 않는 코드	2.6	A function should not contain unused label declarations (함수 내에 사용되지 않은 레이블(label) 선언은 없어야 함)
10	Advisory	2. 사용하지 않는 코드	2.7	There should be no unused parameters in functions (함수에서 사용되지 않은 파라미터는 없어야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
11	Required	3. 주석	3.1	The character sequences /* and // shall not be used within a comment (문자열 /*와 //는 주석 안에서 사용하지 말아야 함)
12	Required	3. 주석	3.2	Line-splicing shall not be used in // comments (행 접합(Line-splicing)은 // 주석 내에서 사용하면 안 됨)
13	Required	4. 문자 세트	4.1	Octal and hexadecimal escape sequences shall be terminated (8진수, 16진수 escape 시퀀스에 다른 escape 시퀀스 외에는 붙이면 안 됨)
14	Advisory	4. 문자 세트	4.2	Trigraphs should not be used (Trigraph는 사용하면 안 됨)
15	Required	5. 식별자	5.1	External identifiers shall be distinct (외부 식별자는 구별되어야 함)
16	Required	5. 식별자	5.2	Identifiers declared in the same scope and name space shall be distinct (같은 scope 혹은 name space에 선언된 식별자는 구별되어야 함)
17	Required	5. 식별자	5.3	An identifier declared in an inner scope shall not hide an identifier declared in an outer scope (안쪽 scope의 식별자를 바깥 scope의 식별자가 가리면 안 됨)
18	Required	5. 식별자	5.4	Macro identifiers shall be distinct (매크로의 식별자는 구별되어야 함)
19	Required	5. 식별자	5.5	Identifiers shall be distinct from macro names (식별자는 매크로 이름과 구별되어야 함)
20	Required	5. 식별자	5.6	A typedef name shall be a unique identifier (typedef 이름은 유일한 식별자여야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
21	Required	5. 식별자	5.7	A tag name shall be a unique identifier (tag(struct, union, enum) 이름은 유일한 식별자여야 함)
22	Required	5. 식별자	5.8	Identifiers that define objects or functions with external linkage shall be unique (외부 연결을 갖는 변수나 함수 식별자는 유일해야 함)
23	Advisory	5. 식별자	5.9	Identifiers that define objects or functions with internal linkage should be unique (내부 연결을 갖는 변수나 함수 식별자는 유일해야 함)
24	Required	6. 타입	6.1	Bit-fields shall only be declared with an appropriate type (bit-field는 올바른 타입으로 선언되어야 함)
25	Required	6. 타입	6.2	Single-bit named bit fields shall not be of a signed type (single-bit로 표현된 이름있는 bit-field는 signed면 안 됨)
26	Required	7. 리터럴과 상수	7.1	Octal constants shall not be used (8진수 상수는 사용하면 안 됨)
27	Required	7. 리터럴과 상수	7.2	A “u” or “U” suffix shall be applied to all integer constants that are represented in an unsigned type (unsigned integer 상수에는 접미사 "u"나 "U"를 붙여야 함)
28	Required	7. 리터럴과 상수	7.3	The lowercase character “l” shall not be used in a literal suffix (소문자 접미사 "l" 은 사용하면 안 됨)
29	Required	7. 리터럴과 상수	7.4	A string literal shall not be assigned to an object unless the object’s type is “pointer to const-qualified char” (문자열을 const char * 타입이 아닌 객체에 할당하면 안 됨)
30	Required	8. 선언과 정의	8.1	Types shall be explicitly specified (타입은 명시적으로 입력해야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
31	Required	8. 선언과 정의	8.2	Function types shall be in prototype form with named parameters (함수는 이름 있는 파라미터로 구성된 프로토타입 형태여야 함)
32	Required	8. 선언과 정의	8.3	All declarations of an object or function shall use the same names and type qualifiers (동일한 객체 또는 함수의 모든 선언은 같은 이름과 타입 한정자를 사용해야 함)
33	Required	8. 선언과 정의	8.4	A compatible declaration shall be visible when an object or function with external linkage is defined (객체나 함수의 정의 또는 호출 이전에 호환가능한 선언이 존재해야함)
34	Required	8. 선언과 정의	8.5	An external object or function shall be declared once in one and only one file (외부연결(external linkage)를 갖는 객체나 함수는 오직 하나의 파일에서만 선언되어야함)
35	Required	8. 선언과 정의	8.6	An identifier with external linkage shall have exactly one external definition (외부연결 식별자는 하나의 외부정의(external definition)을 가져야 함)
36	Advisory	8. 선언과 정의	8.7	Functions and objects should not be defined with external linkage if they are referenced in only one translation unit (함수나 객체가 하나의 번역단위에서 참조된다면 외부참조(external linkage)로 정의되면 안됨)
37	Required	8. 선언과 정의	8.8	The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage (내부 연결(internal linkage) 을 갖는 모든 객체 또는 함수는 static 을 이용해야 함)
38	Advisory	8. 선언과 정의	8.9	An object should be defined at block scope if its identifier only appears in a single function (단일 함수에만 쓰이는 객체는 해당 블록범위에서 정의되어야 한다.)
39	Required	8. 선언과 정의	8.10	An inline function shall be declared with the static storage class (인라인 함수는 static으로 선언되어야 한다.)
40	Advisory	8. 선언과 정의	8.11	When an array with external linkage is declared, its size should be explicitly specified (배열에 대한 외부연결(external linkage) 선언 시, 그 크기를 명시적으로 해야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
41	Required	8. 선언과 정의	8.12	Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique (열거자 리스트 내에서 묵시적으로 지정된 열거형 상수의 값은 유일해야 함)
42	Advisory	8. 선언과 정의	8.13	A pointer should point to a const-qualified type whenever possible (가능하다면 포인터는 const 로 한정된 타입을 가리켜야 함)
43	Required	8. 선언과 정의	8.14	The restrict type qualifier shall not be used (restrict 사용금지)
44	Mandatory	9. 초기화	9.1	The value of an object with automatic storage duration shall not be read before it has been set (모든 변수는 읽기 전에 할당되어야 함)
45	Required	9. 초기화	9.2	The initializer for an aggregate or union shall be enclosed in braces (배열, 구조체, 유니온 타입의 초기화는 큰괄호('{ }') 로 둘러쌓여야 함)
46	Required	9. 초기화	9.3	Arrays shall not be partially initialized (배열은 일부분만 초기화 되면 안됨)
47	Required	9. 초기화	9.4	An element of an object shall not be initialized more than once (객체의 요소 두 번 이상 초기화 금지)
48	Required	9. 초기화	9.5	Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly (지정 초기화(designated initializer)가 사용된 배열은 그 크기가 명시적이어야 함)
49	Required	10. 에센셜 타입 모델	10.1	Operands shall not be of an inappropriate essential type (부적절한 essential 타입의 피연산자를 사용하지 않아야 함)
50	Required	10. 에센셜 타입 모델	10.2	Expressions of essentially character type shall not be used in appropriately in addition and subtraction operations (Essential 타입이 character인 표현식은 가감연산자에 부적합하게 사용되지 않아야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
51	Required	10. 에센셜 타입 모델	10.3	The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category (표현식의 값은 더 작은 essential 타입이나 다른 essential 타입 분류에 타입을 갖는 객체에 할당되지 않아야 함)
52	Required	10. 에센셜 타입 모델	10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential typecategory (일반 산술 변환이 수행되는 연산자의 두 피연산자들은 필히 같은 essential 타입 분류에 속하는 타입이어야 함)
53	Advisory	10. 에센셜 타입 모델	10.5	The value of an expression should not be cast to an inappropriate essential type (수식의 값은 적절하지 않은 essential type으로 변환되지 않아야 함)
54	Required	10. 에센셜 타입 모델	10.6	The value of a composite expression shall not be assigned to an object with wider essential type (복합 표현식의 값은 더 큰 essential 타입의 객체에 할당되지 않아야 함)
55	Required	10. 에센셜 타입 모델	10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type (복합 수식이 기본 산술 변환을 수행하는 연산자의 피연산자로 사용된 경우, 다른 피연산자는 해당 수식의 타입보다 큰 essential 타입을 가지지 않아야 함)
56	Required	10. 에센셜 타입 모델	10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type (복합 수식의 값은 다른 essential 타입 분류에 속하는 타입이나 더 큰 essential 타입으로 변환되지 않아야 함)
57	Required	11. 포인터 타입 변환	11.1	Conversions shall not be performed between a pointer to a function and any other type (함수 포인터와 다른 타입 간의 형 변환 금지)
58	Required	11. 포인터 타입 변환	11.2	Conversions shall not be performed between a pointer to an incomplete type and any other type (불완전 포인터(incomplete)타입과 다른 타입 간의 형 변환 금지)
59	Required	11. 포인터 타입 변환	11.3	A cast shall not be performed between a pointer to object type and a pointer to a different object type (한 객체 포인터로와 다른 객체 포인터 간의 변환 금지)
60	Advisory	11. 포인터 타입 변환	11.4	A conversion should not be performed between a pointer to object and an integer type (객체 포인터와 정수형 타입 간의 변환 금지)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
61	Advisory	11. 포인터 타입 변환	11.5	A conversion should not be performed from pointer to void into pointer to object (void 포인터에서 객체 포인터로 변환 금지)
62	Required	11. 포인터 타입 변환	11.6	A cast shall not be performed between pointer to void and an arithmetic type (void 포인터와 산술 타입 간의 변환 금지)
63	Required	11. 포인터 타입 변환	11.7	A cast shall not be performed between pointer to object and a noninteger arithmetic type (객체 포인터와 정수가 아닌 산술 타입 간의 변환금지)
64	Required	11. 포인터 타입 변환	11.8	A cast shall not remove any const or volatile qualification from the type pointed to by a pointer (형 변환 시 포인터의 const 또는 volatile 제거금지)
65	Required	11. 포인터 타입 변환	11.9	The macro NULL shall be the only permitted form of integer null pointer constant (매크로 NULL은 정수 null 포인터 상수만으로 사용해야 함)
66	Advisory	12. 표현식	12.1	The precedence of operators within expressions should be made explicit (수식 내부에 사용된 연산자의 우선순위가 명확한지 검사)
67	Required	12. 표현식	12.2	The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand (shift 연산자의 우측 피연산자가 좌측 피연산자의 essential 타입의 범위 내의 정수여야 함)
68	Advisory	12. 표현식	12.3	The comma operator should not be used (Comma 연산자 사용금지)
69	Advisory	12. 표현식	12.4	Evaluation of constant expressions should not lead to unsigned integer wrap-around (unsigned integer wrap-around를 발생시키는 상수 수식 금지)
70	Mandatory	12. 표현식	12.5	The sizeof operator shall not have an operand which is a function parameter declared as "array of type" (sizeof 연산자는 "배열 유형"으로 선언 된 함수 매개 변수를 가지면 안 됨)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
71	Required	13. 부작용	13.1	Initializer lists shall not contain persistent side effects (초기화 리스트가 영구적인 side effect을 일으키면 안 됨)
72	Required	13. 부작용	13.2	The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders (수식의 값과 영구적인 side effect의 평가결과가 평가 순서에 상관없이 같아야 함)
73	Advisory	13. 부작용	13.3	A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator (증감 연산자(++,-)를 포함한 수식은 해당 증감 연산자를 제외한 잠재적인 side effect 금지)
74	Advisory	13. 부작용	13.4	The result of an assignment operator should not be used (할당 연산자의 결과를 사용 금지)
75	Required	13. 부작용	13.5	The right hand operand of a logical && or operator shall not contain persistent side effects (논리적 &&나 연산자의 우측 항에 영구적인 side effect 가 있으면 안 됨)
76	Mandatory	13. 부작용	13.6	The operand of the sizeof operator shall not contain any expression which has potential side effects (sizeof의 피연산자에 side effect를 발생시킬 수 있는 수식 포함 금지)
77	Required	14. 제어문 표현식	14.1	A loop counter shall not have essentially floating type (loop counter 의 essential 타입이 실수형이면 안 됨)
78	Required	14. 제어문 표현식	14.2	A for loop shall be well-formed (for 문이 잘 짜여져야 함)
79	Required	14. 제어문 표현식	14.3	Controlling expressions shall not be invariant (결과가 항상 같은 제어식 사용 금지)
80	Required	14. 제어문 표현식	14.4	The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type (조건문이나 반복문의 제어식이 essentially Boolean type 인지 검사)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
81	Advisory	15. 제어 흐름	15.1	The goto statement should not be used (goto 문 사용 금지)
82	Required	15. 제어 흐름	15.2	The goto statement shall jump to a label declared later in the same function (goto 문의 label이 같은 함수에서 더 나중에 위치하는지 검사)
83	Required	15. 제어 흐름	15.3	Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement (goto 문이 참조하는 label은 같은 블록이나 인접한 블록에 있는지 검사)
84	Advisory	15. 제어 흐름	15.4	There should be no more than one break or goto statement used to terminate any iteration statement (반복문에는 하나의 break 나 goto 문만 있는지 검사)
85	Advisory	15. 제어 흐름	15.5	A function should have a single point of exit at the end (함수는 마지막에 하나의 return만 가지는지 검사)
86	Required	15. 제어 흐름	15.6	The body of an iteration-statement or a selection-statement shall be a compound-statement (반복문이나 선택문이 복합문인지 검사)
87	Required	15. 제어 흐름	15.7	All if ... else if constructs shall be terminated with an else statement (모든 if ... else if 구조는 else 문으로 끝나는지 검사)
88	Required	16. Switch 문	16.1	All switch statements shall be well-formed (모든 switch 문은 잘 짜여야 함)
89	Required	16. Switch 문	16.2	A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement (switch label 을 포함하는 가장 가까운 문장은 switch 문이어야 함)
90	Required	16. Switch 문	16.3	An unconditional break statement shall terminate every switch-clause (모든 switch 절은 break 로 끝나야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
91	Required	16. Switch 문	16.4	Every switch statement shall have a default label (모든 switch 문에 default label 이 있어야 함)
92	Required	16. Switch 문	16.5	A default label shall appear as either the first or the last switch label of a switch statement (default label 은 switch 문장의 맨처음이나 마지막 switch label 이어야 함)
93	Required	16. Switch 문	16.6	Every switch statement shall have at least two switch-clauses (모든 switch 문에 적어도 둘 이상의 switch 절이 있어야 함)
94	Required	16. Switch 문	16.7	A switch-expression shall not have essentially Boolean type (switch-expression 은 Boolean 타입이면 안 됨)
95	Required	17. 함수	17.1	The features of <stdarg.h> shall not be used (stdarg.h에 정의된 요소들은 사용 금지)
96	Required	17. 함수	17.2	Functions shall not call themselves, either directly or indirectly (직, 간접적 재귀호출 금지)
97	Mandatory	17. 함수	17.3	A function shall not be declared implicitly (함수를 묵시적으로 선언 금지)
98	Mandatory	17. 함수	17.4	All exit paths from a function with non-void return type shall have an explicit return statement with an expression (리턴 타입이 void가 아닌 함수의 모든 경로의 마지막은 수식을 포함한 return 문이어야 함)
99	Advisory	17. 함수	17.5	The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements (배열 타입으로 선언된 함수의 매개변수는 적합한 수의 원소를 가져야 함)
100	Mandatory	17. 함수	17.6	The declaration of an array parameter shall not contain the static keyword between the [] (배열 타입의 매개 변수의 선언은 [] 사이에 static keyword를 포함하지 않아야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
101	Required	17. 함수	17.7	The value returned by a function having non-void return type shall be used (리턴 타입이 void가 아닌 함수의 리턴 값은 필히 사용해야 함)
102	Advisory	17. 함수	17.8	A function parameter should not be modified (함수의 매개변수는 변경되지 않아야 함)
103	Required	18. 포인터와 배열	18.1	A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand (포인터의 연산 결과는 해당 포인터가 가리키는 배열의 요소이어야 함)
104	Required	18. 포인터와 배열	18.2	Subtraction between pointers shall only be applied to pointers that address elements of the same array (포인터 간의 뺄셈은 같은 배열의 요소들을 가리키고 있는 포인터들에만 적용되어야 함)
105	Required	18. 포인터와 배열	18.3	The relational operators >, >=, < and <= shall not be applied to objects of pointer type except where they point into the same object (관계 연산자 >, >=, <, 그리고 <=는 같은 객체를 가리키고 있는 포인터를 제외한 다른 포인터 타입의 객체에 적용 금지)
106	Advisory	18. 포인터와 배열	18.4	The +, -, += and -= operators should not be applied to an expression of pointer type (+, -, +=, -= 연산자는 포인터 타입의 표현식에 적용 금지)
107	Advisory	18. 포인터와 배열	18.5	Declarations should contain no more than two levels of pointer nesting (3차원 이상의 포인터 선언 금지)
108	Required	18. 포인터와 배열	18.6	The address of an object with automatic storage shall not be copied to another object that persists after the first object has ceased to exist (자동으로 값이 할당되는 객체의 주소는, 처음 객체가 소멸된 이후에도 지속되는 다른 객체로 복사 금지)
109	Required	18. 포인터와 배열	18.7	Flexible array members shall not be declared (구조체의 멤버로써 가변 배열 선언금지)
110	Required	18. 포인터와 배열	18.8	Variable-length array types shall not be used (Variable-length(길이를 변수로 지정한) 배열 타입 사용 금지)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
111	Mandatory	19. 저장 영역 덮어쓰기	19.1	An object shall not be assigned or copied to an overlapping object (오버랩되는 객체에 할당 또는 복사 금지)
112	Advisory	19. 저장 영역 덮어쓰기	19.2	The union keyword should not be used (union 키워드 사용 금지)
113	Advisory	20. 전처리 지시자	20.1	#include directives should only be preceded by preprocessor directives or comments (소스코드에서 #include의 상단에는 전처리 지시자 또는 주석만 허용)
114	Required	20. 전처리 지시자	20.2	The ' , " or \ characters and the /* or // character sequences shall not occur in a header file name (문자 ' , " , \ 와 문자열 /* , // 은 헤더파일의 이름에 포함 금지)
115	Required	20. 전처리 지시자	20.3	The #include directive shall be followed by either a <filename> or "filename" sequence (#include 지시자는 또는 "filename"의 형태를 따라야 함)
116	Required	20. 전처리 지시자	20.4	A macro shall not be defined with the same name as a keyword (키워드와 같은 이름으로 매크로 정의 금지)
117	Advisory	20. 전처리 지시자	20.5	#undef should not be used (#undef 사용 금지)
118	Required	20. 전처리 지시자	20.6	Tokens that look like a preprocessing directive shall not occur within a macro argument (매크로 인자 부분에 전처리 지시자 사용금지)
119	Required	20. 전처리 지시자	20.7	Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses (매크로 인자는 괄호로 감싸야 함)
120	Required	20. 전처리 지시자	20.8	The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1 (#if 또는 #elif의 제어 표현식은 0 또는 1로 값이 나와야 함)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
121	Required	20. 전처리 지시자	20.9	All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation (#if 또는 #elif의 제어 표현식에 사용된 모든 식별자는 평가하기 전에 #define으로 정의 되어야 함)
122	Advisory	20. 전처리 지시자	20.10	The # and ## preprocessor operators should not be used (전처리 연산자 # 과 ## 사용금지)
123	Required	20. 전처리 지시자	20.11	A macro parameter immediately following a # operator shall not immediately be followed by a ## operator (전처리 연산자 #의 피연산자 바로 뒤에 ## 연산자 사용금지)
124	Required	20. 전처리 지시자	20.12	A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators (# 또는 ## 의 피연산자이며 추가적인 매크로 치환이 필요한 매크로 인자는 다른 연산자의 피연산자로 사용 금지)
125	Required	20. 전처리 지시자	20.13	A line whose first token is # shall be a valid preprocessing directive (첫번째 토큰이 # 인 행은 유효한 전처리 지시자여야 함)
126	Required	20. 전처리 지시자	20.14	All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related (모든 #else, #elif, #endif 전처리 지시자는 같은 파일 내에 관련된 #if, #ifdef, #ifndef 가 존재해야 한다.)
127	Required	21. 표준 라이브러리	21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name (예약된 식별자 또는 예약된 매크로 이름을 #define 또는 #undef에 사용 금지)
128	Required	21. 표준 라이브러리	21.2	A reserved identifier or reserved macro name shall not be declared (예약된 지시자 또는 예약된 매크로 이름으로 선언 금지)
129	Required	21. 표준 라이브러리	21.3	The memory allocation and deallocation functions of <stdlib.h> shall not be used (stdlib.h의 메모리 할당과 해제 함수 사용 금지)
130	Required	21. 표준 라이브러리	21.4	The standard header file <setjmp.h> shall not be used (setjmp.h 사용 금지)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
131	Required	21. 표준 라이브러리	21.5	The standard header file <signal.h> shall not be used (signal.h 사용 금지)
132	Required	21. 표준 라이브러리	21.6	The Standard Library input/output functions shall not be used (stdio.h 등 표준 입출력 함수 사용 금지)
133	Required	21. 표준 라이브러리	21.7	The Standard Library functions atof, atoi, atol and atoll functions of <stdlib.h> shall not be used (stdlib.h의 atof, atoi, atol, atoll 함수 사용 금지)
134	Required	21. 표준 라이브러리	21.8	The Standard Library functions abort, exit, getenv and system of <stdlib.h> shall not be used (stdlib.h의 라이브러리 함수인 abort, exit, getenv, system 사용 금지)
135	Required	21. 표준 라이브러리	21.9	The Standard Library functions bsearch and qsort of <stdlib.h> shall not be used (stdlib.h의 라이브러리 함수 bsearch, qsort 사용 금지)
136	Required	21. 표준 라이브러리	21.10	The Standard Library time and date functions shall not be used (표준 라이브러리 time, date 함수 사용 금지)
137	Required	21. 표준 라이브러리	21.11	The standard header file <tgmath.h> shall not be used (tgmath.h 사용 금지)
138	Advisory	21. 표준 라이브러리	21.12	The exception handling features of <fenv.h> should not be used (fenv.h의 예외 처리 식별자 사용 금지)
139	Mandatory	21. 표준 라이브러리	21.13	Any value passed to a function in <ctype.h> shall be representable as an unsigned char or be the value EOF (<ctype.h>의 함수에 전달 된 값은 모두 unsigned char로 표현되거나 EOF 값이어야 함)
140	Required	21. 표준 라이브러리	21.14	The Standard Library function memcmp shall not be used to compare null terminated strings (표준 라이브러리 함수 memcmp는 null로 끝나는 문자열을 비교하는 데 사용해서는 안됨)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
141	Required	21. 표준 라이브러리	21.15	The pointer arguments to the Standard Library functions memcpy, memmove and memcmp shall be pointers to qualified or unqualified versions of compatible types (표준 라이브러리 함수 인 memcpy, memmove 및 memcmp에 대한 포인터 인자는 규정 버전 또는 규정되지 않았지만 호환 가능 유형의 포인터여야 함)
142	Required	21. 표준 라이브러리	21.16	The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type (표준 라이브러리 함수 memcmp에 대한 포인터 인자는 포인터 유형, essential signed 유형, essential unsigned 유형, essential boolean 유형 또는 essential enum 유형이어야 함)
143	Mandatory	21. 표준 라이브러리	21.17	Use of the string handling functions from <string.h> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters (<string.h>의 문자열 처리 함수의 사용으로 참조한 포인터 매개 변수의 의 경계를 벗어나는 접근 금지)
144	Mandatory	21. 표준 라이브러리	21.18	The size_t argument passed to any function in <string.h> shall have an appropriate value (<string.h>의 함수에 전달 된 size_t 인자는 적절한 값을 가져야 함)
145	Mandatory	21. 표준 라이브러리	21.19	The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type (표준 라이브러리 함수 localeconv, getenv, setlocale 또는 strerror가 반환 한 포인터는 const 유형에 대한 포인터를 가지고있는 경우만 사용함)
146	Mandatory	21. 표준 라이브러리	21.20	The pointer returned by the Standard Library functions asctime, ctime, gmtime, localtime, localeconv, getenv, setlocale or strerror shall not be used following a subsequent call to the same function (표준 라이브러리 함수 asctime, ctime, gmtime, localtime, localeconv, getenv, setlocale 또는 strerror가 반환한 포인터는 동일한 함수에 대한 후속 호출에 사용 금지)
147	Required	22. 자원	22.1	All resources obtained dynamically by means of Standard Library functions shall be explicitly released (표준 라이브러리를 통해 동적으로 얻은 리소스는 명시적으로 해제되어야 함)
148	Mandatory	22. 자원	22.2	A block of memory shall only be freed if it was allocated by means of a Standard Library function (메모리는 시스템 라이브러리 함수를 통해 할당되었을 때만 해제되어야 함)
149	Required	22. 자원	22.3	The same file shall not be open for read and write access at the same time on different streams (서로 다른 스트림에서 같은 파일을 동시에 읽기, 쓰기로 열면 안 됨)
150	Mandatory	22. 자원	22.4	There shall be no attempt to write to a stream which has been opened as read-only (읽기 전용으로 열린 스트림에 쓰기를 하면 안 됨)

MISRA C 2012 룰 목록

NO.	위험도	룰셋	룰	룰 설명
151	Mandatory	22. 자원	22.5	A pointer to a FILE object shall not be dereferenced (FILE 객체를 가리키는 포인터는 dereference되면 안 됨)
152	Mandatory	22. 자원	22.6	The value of a pointer to a FILE shall not be used after the associated stream has been closed (FILE 객체를 가리키는 포인터를 해당 스트림이 닫힌 후에 사용하면 안 됨)
153	Required	22. 자원	22.7	The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF (매크로 EOF는 EOF를 반환 할 수있는 표준 라이브러리 함수의 수정되지 않은 반환 값과 비교해야 함)
154	Required	22. 자원	22.8	The value of errno shall be set to zero prior to a call to an errno-settingfunction (errno의 값은 errno-setting 함수를 호출하기 전에 0으로 설정해야 함)
155	Required	22. 자원	22.9	The value of errno shall be tested against zero after calling an errnosetting- function (errno 값은 errnosetting 함수를 호출 한 후 0에 대해 검사해야 함)
156	Required	22. 자원	22.10	The value of errno shall only be tested when the last function to be called was an errno-setting-function (errno의 값은 호출 될 마지막 함수가 errno-setting-function 일 때만 시험해야 함)

CppCheck 개요와 실습


□ 사용 도구: CppCheck (오픈소스)


- <http://cppcheck.sourceforge.net/>
- 특징
 - 오픈소스 (무료로 사용 가능)
 - 기본적으로, MISRA 지원하지 않음
 - MISRA를 구매하고 룰 정보를 txt로 변환하였을 경우 MISRA 검증 가능
 - MISRA 구매 비용: 25 파운드
 - MISRA의 모든 룰을 점검하지 않으나, MISRA 지원을 위한 투자 진행


□ 분석 대상 소스코드: Curl


- <https://github.com/curl/curl>
- 커맨드라인에서 http, ftp 등의 접속을 위한 도구 (C로 개발)


[실습] Curl 다운로드


 curl / curl


 Sponsor


 Watch 647


 Star 13,838


 Fork 3,059


 Code

 Issues 24

 Pull requests 32

 Wiki

 Security

 Insights

Join GitHub today


Dismiss


GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together.


Sign up


A command line tool and library for transferring data with URL syntax, supporting HTTP, HTTPS, FTP, FTPS, GOPHER, TFTP, SCP, SFTP, SMB, TELNET, DICT, LDAP, LDAPS, FILE, IMAP, SMTP, POP3, RTSP and RTMP. libcurl offers a myriad of powerful features <https://curl.haxx.se/>


[http](#) [https](#) [ftp](#) [user-agent](#) [client](#) [library](#) [curl](#) [libcurl](#) [c](#) [transfer-data](#) [ldap](#)

 24,463 commits

 12 branches

 186 releases

 519 contributors


 View license

Branch: master ▾


New pull request

Find File

Clone or download ▾


 Gaël PORTAY and bagder curl_multi_wait.3: escape backslash in example ...

Latest commit 44b5468 yesterday

 .github

.github/FUNDING: mention our opencollective "home" [ci skip]

last month

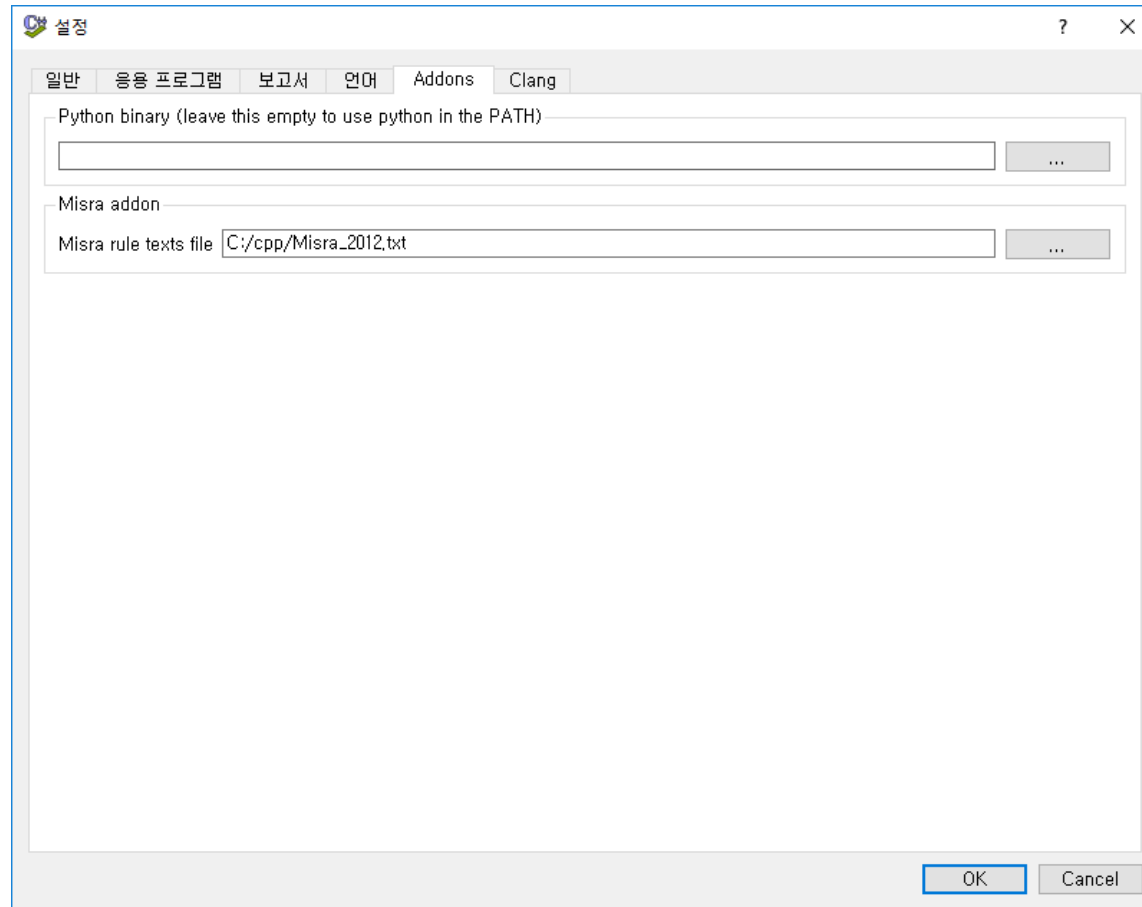
 CMake

build: fix Codacy warnings

22 days ago

[실습] MISRA C 검증을 위한 설정

□ 편집 -> 설정



[실습] 검증 결과 확인

Cppcheck - 프로젝트: C:/Users/Dongjoon Han/Downloads/curl-master/src/test1.cppcheck

파일(F) 편집(E) 보기(V) Analyze 도움말(H)

빠른 필터:

파일	분류	행	Id
..\\include\\curl\\curl.h	스타일	138	misra-c2012-5.4
..\\include\\curl\\curl.h	스타일	951	misra-c2012-5.4
..\\include\\curl\\curl.h	스타일	944	misra-c2012-5.4
..\\include\\curl\\curl.h	스타일	947	misra-c2012-5.4
..\\include\\curl\\curl.h	스타일	948	misra-c2012-5.4
..\\include\\curl\\curl.h	스타일	949	misra-c2012-5.4
..\\include\\curl\\curl.h	스타일	950	misra-c2012-5.4

Id: misra-c2012-5.4
Required Macro identifiers shall be distinct

```
940 #undef CINIT
941 #endif
942
943 #ifdef CURL_ISOCPP
944 #define CINIT(na,t,nu)  CURLOPT_ ## na = CURLOPTTYPE_ ## t + nu
945 #else
946 /* The macro "##" is ISO C, we assume pre-ISO C doesn't support it. */
947 #define LONG            CURLOPTTYPE_LONG
948 #define OBJECTPOINT    CURLOPTTYPE_OBJECTPOINT
949 #define FUNCTIONPOINT  CURLOPTTYPE_FUNCTIONPOINT
950 #define OFF_T          CURLOPTTYPE_OFF_T
951 #define CINIT(name,type,number)  CURLOPT_/**/name = type + number
952 #endif
953
954 /* handy aliases that make no run-time difference */
955 #define CURLOPTTYPE_STRINGPOINT  CURLOPTTYPE_OBJECTPOINT
956 #define CURLOPTTYPE_SLISTPOINT  CURLOPTTYPE_OBJECTPOINT
957
958 /*
959 * This macro-manip below setups the CURLOPT_ {what1 enum} to be used with
```

Analysis Log Warning Details

11 몇 가지 고려사항



고려사항

1. 룰 위반을 개선해도 되는가?
2. 프로젝트 중반, QA가 정적분석 목표치를 강요한다.
적절한 대응 방안은?
3. 안전한 리팩토링 방법은?
4. Dart는?

Q&A

감사합니다.