제76회 OpenTechnet
공개SW가 테스팅을 만나다
지속적인 통합 환경에서의 SW품질 시각화를 통한 효과적인 코드 리뷰 수행 방안(부록)

2015.04.29(수)

김모세

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

1 우분투 Ubuntu 설치

이 챕터에서는 버추얼박스VirtualBox에 우분투Ubuntu 운영체제를 설치하는 법을 설명한다.

1.1 버추얼박스에 가상 이미지 생성

버추얼박스 홈페이지[1]에서 여러분이 사용하는 시스템에 적합한 버추얼박스 및 확장팩 패키지를 다운로드 해서 설치한다.

- 버추얼박스 플랫폼 패키지VirtualBox platform packacges
- 버추얼박스 오라클 가상머신 확장팩VirtualBox Oracle VM VirtualBox Extension Pack

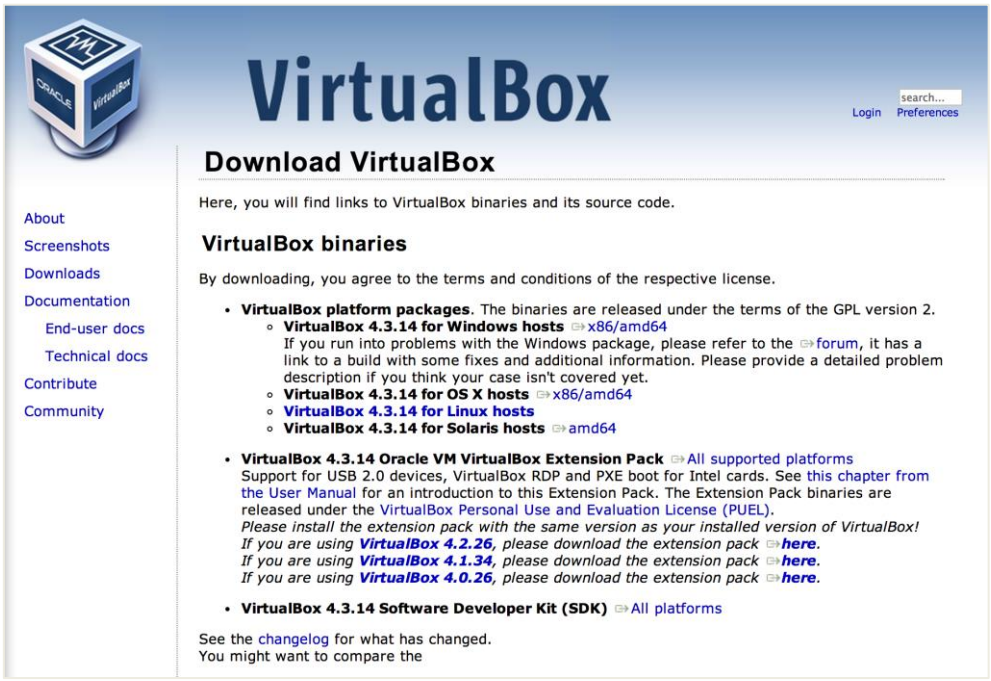


그림 1-1 버추얼박스 다운로드 페이지

버추얼박스 설치가 완료되었다면 다음 과정에 따라 우분투를 설치할 가상 디스크 이미지를 생성한다.

---

[1] https://www.virtualbox.org/wiki/Downloads

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

1. 버추얼박스를 실행 후 ‘새로 만들기(N)’를 클릭한다. 생성할 운영체제의 이름을 입력하고 운영체제의 종류와 버전을 선택한다(운영체제 종류와 버전은 설치 후 변경이 불가능하다).

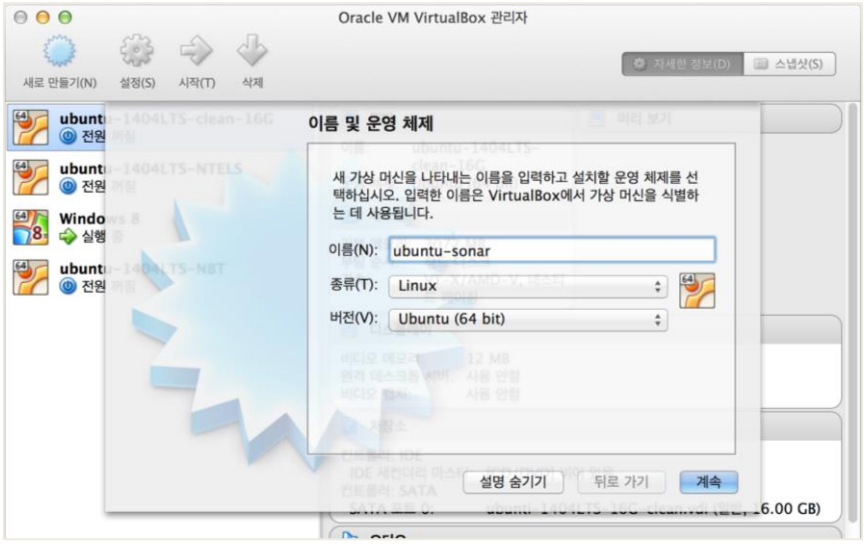


그림 1-2 가상 이미지 이름 및 운영체제 선택

2. 가상 운영체제에서 사용할 메모리 크기를 설정한다(메모리 크기는 운영체제 설치 후에도 변경할 수 있다).

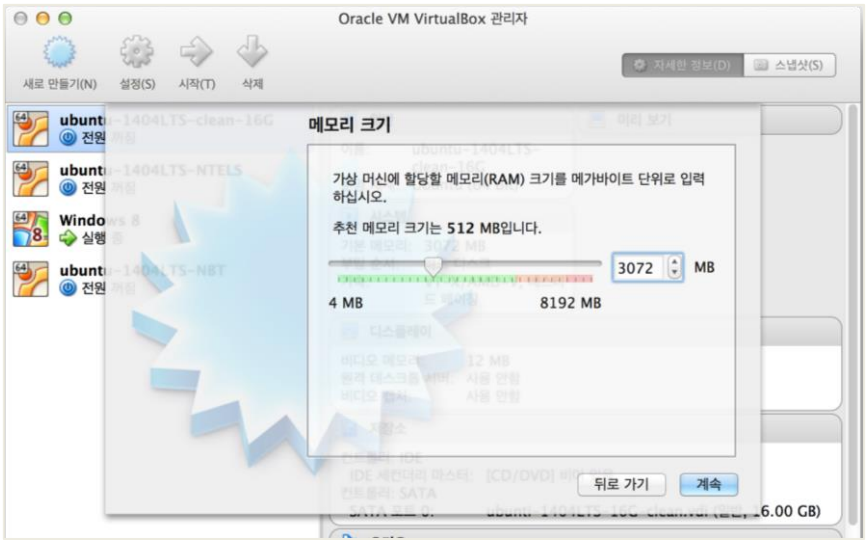


그림 1-3 가상 운영체제 메모리 크기 설정

3. ‘지금 가상 하드 드라이브 만들기(C)’를 선택하고 ‘만들기’를 클릭한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

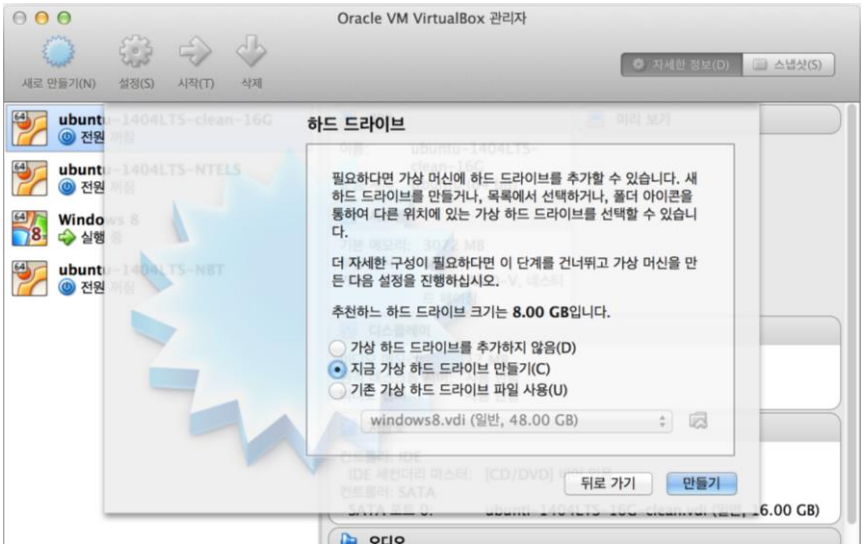


그림 1-4 가상 디스크 하드 드라이브 생성 옵션 선택

4. ‘VDI(VirtualBox 디스크 이미지)’를 선택하고 ‘계속’을 클릭한다.

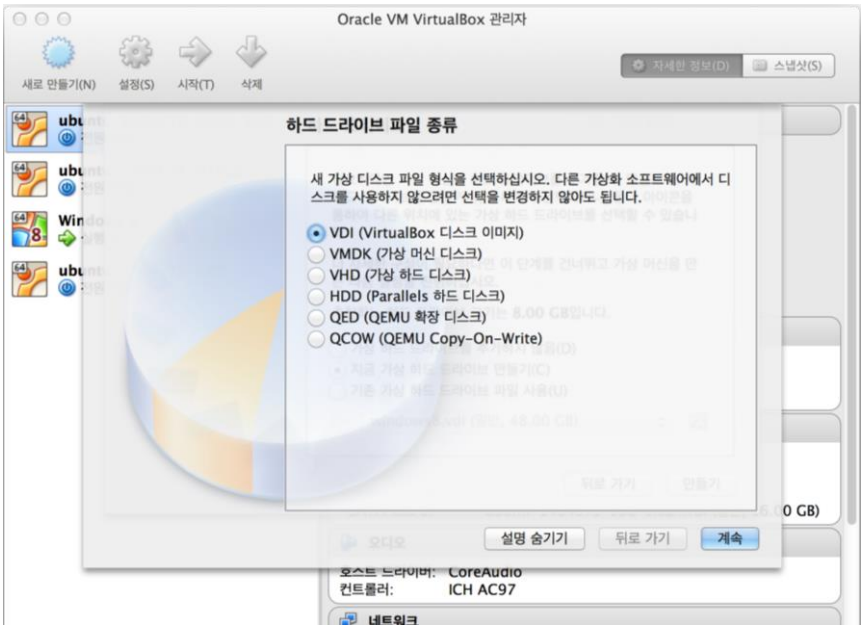


그림 1-5 가상 디스크 하드 드라이브 파일 옵션 선택

5. 물리적 하드 드라이브 저장 옵션을 선택한다. 하드 드라이브에 여유가 있다면 가능한 ‘고정 크기(F)’를 선택하는 것이 좋다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

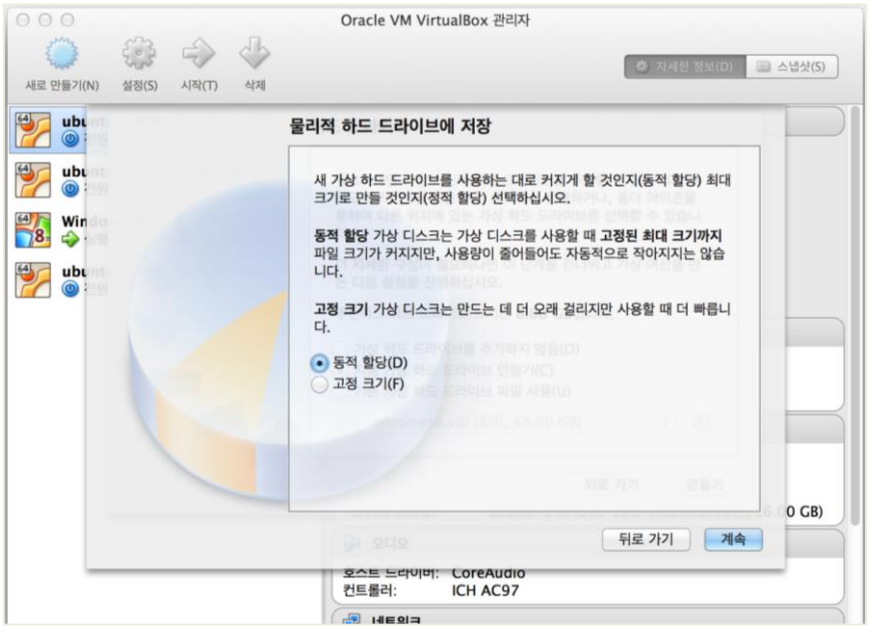


그림 1-6 물리적 하드 드라이브 저장 옵션 선택

6. 파일 위치 및 크기를 설정한다. 원활한 사용을 위해 16GB 이상의 공간을 할당할 것을 권장한다. ‘만들기’를 클릭하면 가상 이미지가 생성된다.

그림 1-7 가상 이미지의 위치 및 크기 설정

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

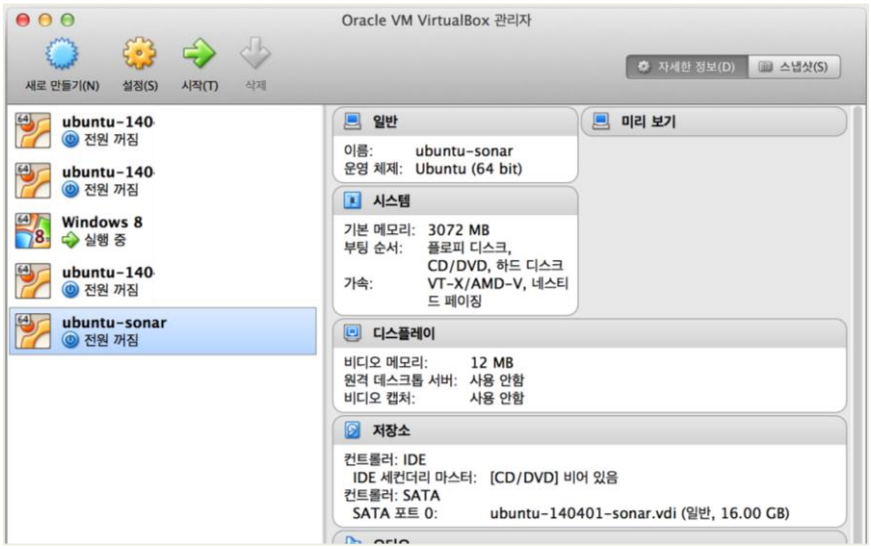


그림 1-8 가상 이미지 설치 완료

1.2 우분투 설치

앞서 생성한 가상 이미지에 우분투를 설치한다. 설치할 우분투는 데스크톱Desktop 버전이며 GUI를 제공하기 때문에 유닉스/리눅스를 처음 접하는 사용자들도 쉽게 접근할 수 있다.

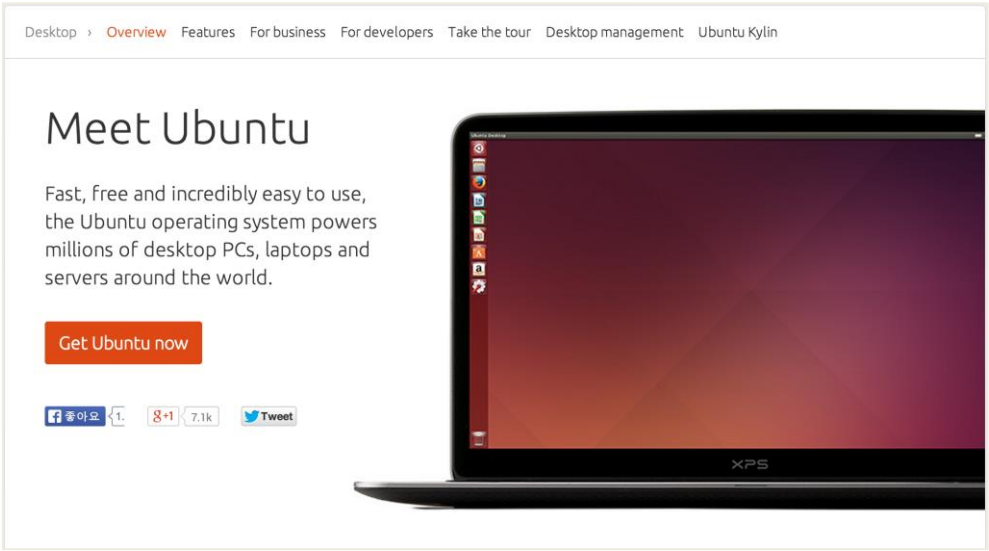


그림 1-9 우분투 홈페이지

다음 과정을 따라 우분투를 설치한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

1. 우분투 홈페이지의 다운로드 페이지[2]에서 설치할 우분투 이미지(.iso)를 다운로드한다. 다운로드받을 파일은 다음과 같다.

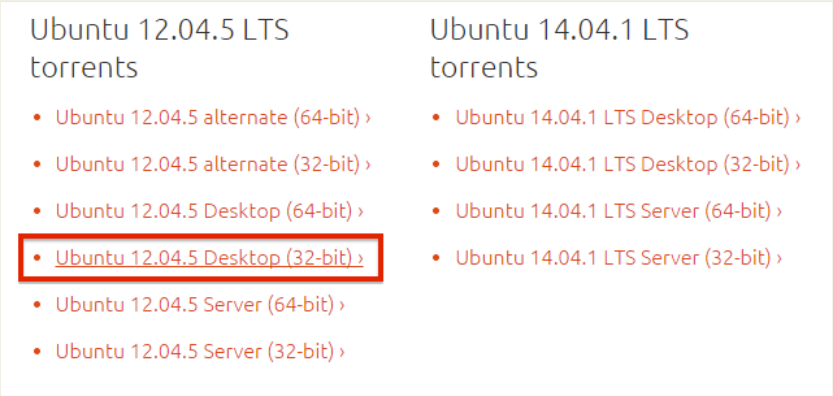
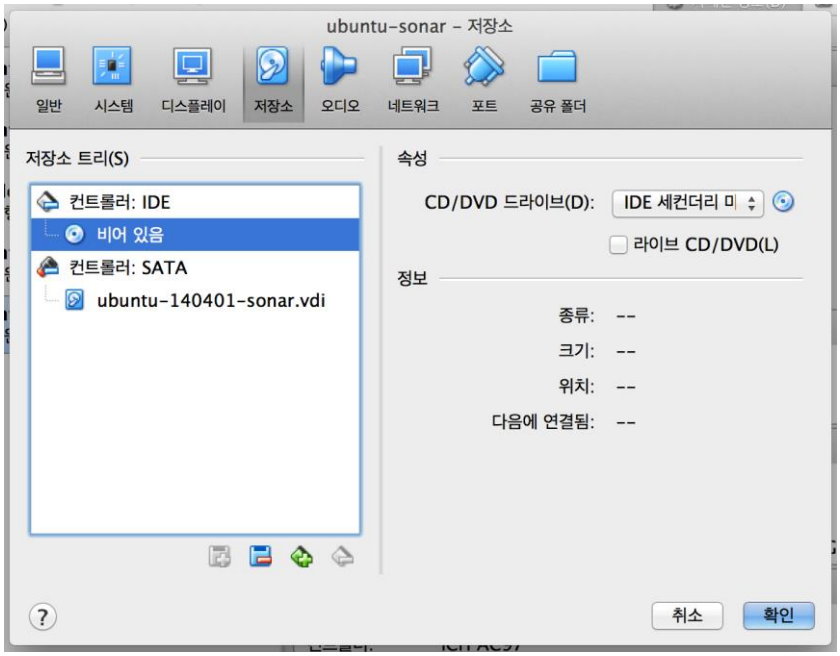


그림 1-10 이 섹션에서 설치할 12.04.x 버전은 토렌트로 제공된다(토렌트 다운로드를 위한 프로그램은 개인이 별도로 설치한다).

2. 버추얼박스를 실행하고 이전 섹션에서 생성한 우분투 설치용 가상 이미지를 선택한다. ‘설정(S) > 저장소 > 컨트롤러: IDE > 비어 있음’을 클릭한다. ‘CD/DVD 드라이브(D)’ 오른쪽의 디스크 버튼을 클릭하고 다운로드받은 우분투 이미지(.iso)를 선택한다. ‘확인’을 클릭한다.



[2] 우분투 다운로드 페이지: http://www.ubuntu.com/download/alternative-downloads

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

그림 1-11 다운로드받은 우분투 이미지 파일을 삽입한다.

3. 버추얼박스에서 우분투 설치용 가상 이미지를 선택하고 ‘시작(T)’을 클릭하면 우분투 설치가 시작된다.

4. 설치 언어를 선택한다. 가급적이면 영어를 선택한다.

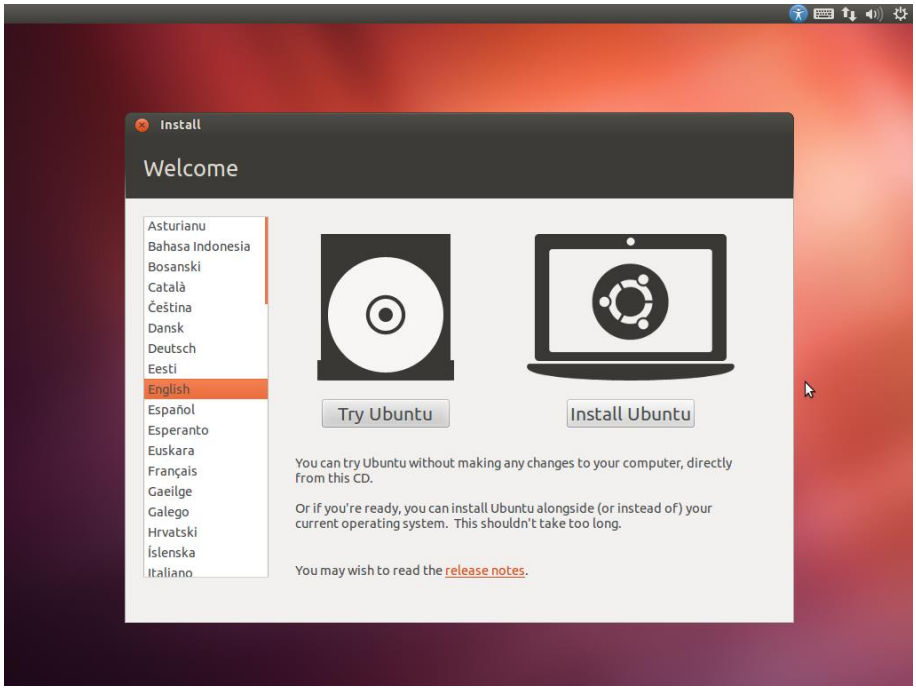


그림 1-12 설치 언어 선택

5. 우분투를 최소 6.5GB 이상의 하드 드라이브 공간을 필요로 한다. 설치과정에서 최신 업데이트 파일을 다운로드 할 수 있다(‘설치하는 동안 업데이트 다운로드Download updates while installing’ 옵션에 체크). ‘Continue’를 클릭한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

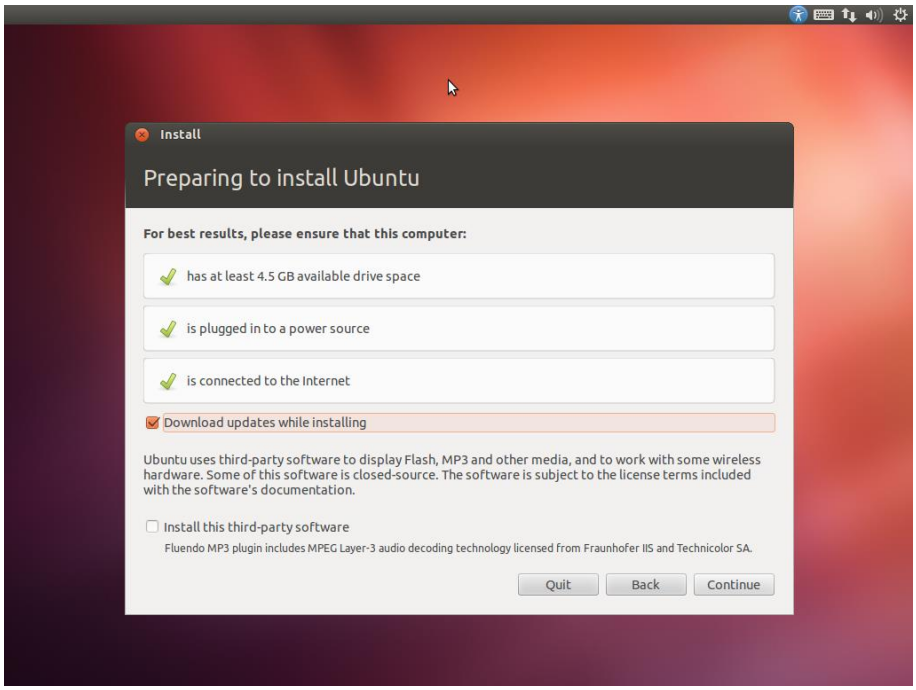


그림 1-13 업데이트 파일 다운로드

6. ‘디스크 삭제 후 우분투 설치Erase disk and install Ubuntu’를 선택하고 ‘Install Now’를 클릭한다.

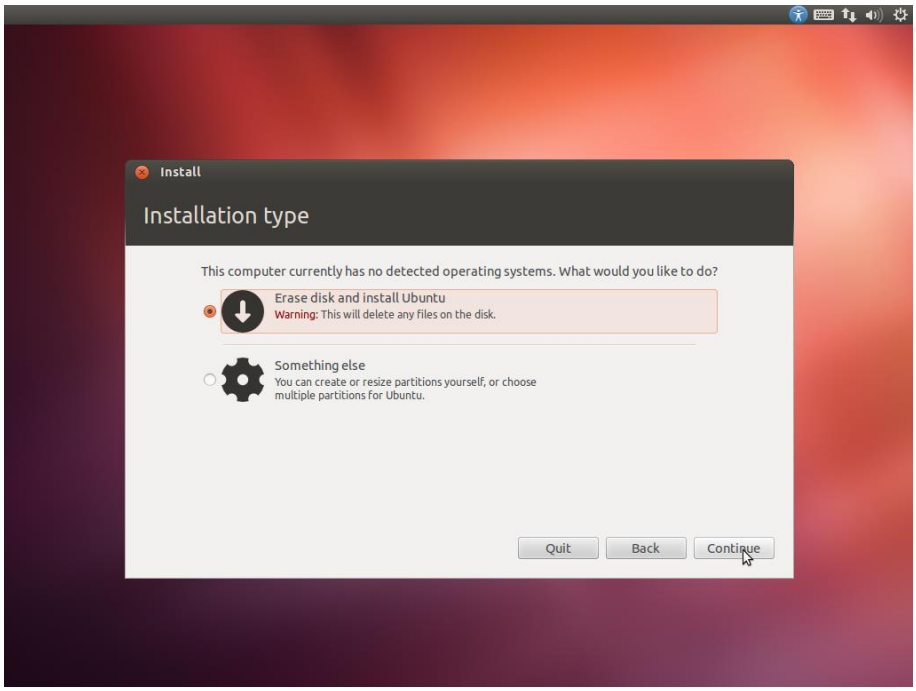


그림 1-14 디스크 삭제 후 우분투 설치

7. 타임존을 선택한다. 네트워크에 연결되어 있다면 현재 위치에 따라 타임존이 자동으로 선택된다. 원하는 타임존을 선택한 후 ‘Continue’를 클릭한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

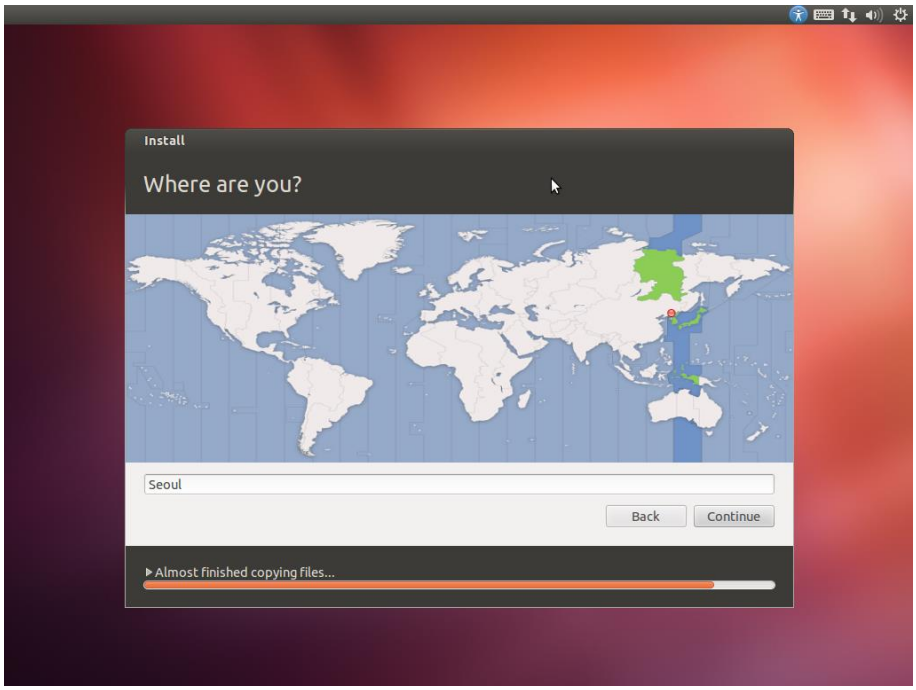


그림 1-15 타임존 설정

8. 키보드 레이아웃을 선택한다. 키보드 레이아웃을 선택한 후 ‘Type here to test your keyboard’ 항목에 문자열을 입력해서 키보드를 확인할 수 있다. ‘Continue’를 클릭한다.

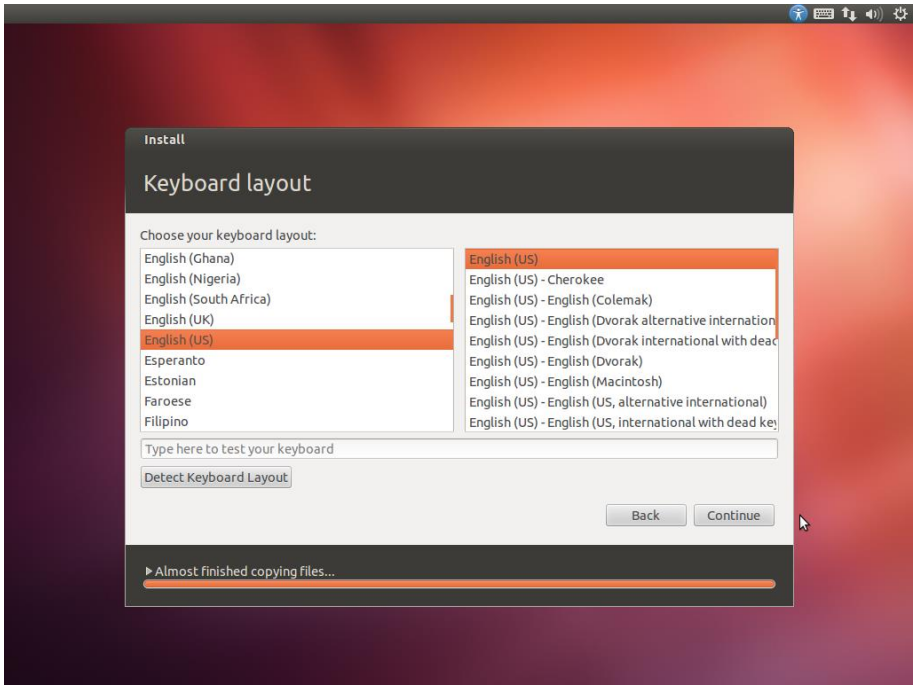


그림 1-16 키보드 레이아웃 선택

9. 우분투 로그인 계정 정보를 입력한다. ‘Continue’를 클릭하면 우분투 설치가 마무리된다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

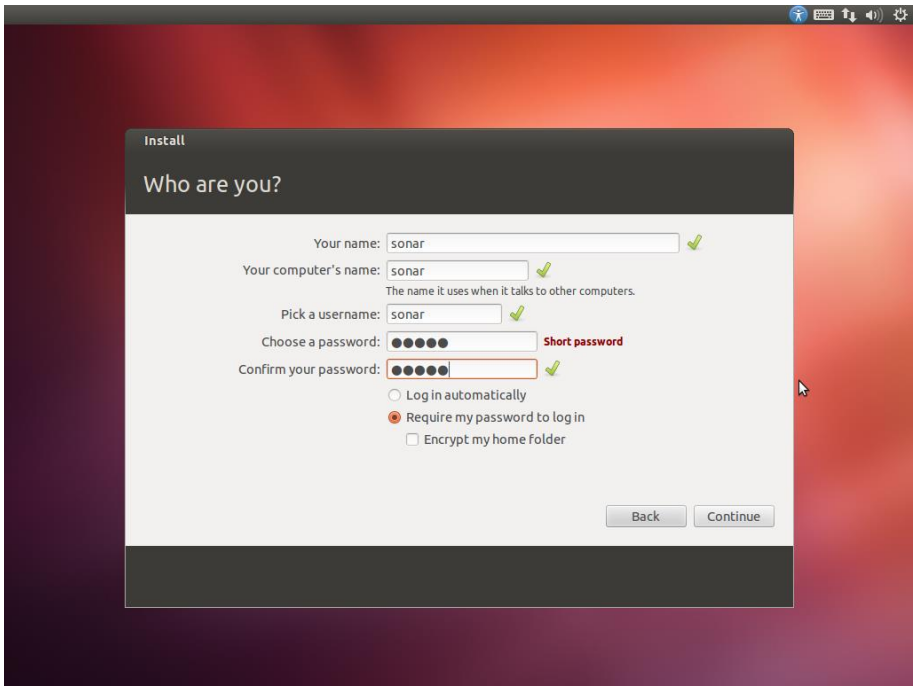


그림 1-17 로그인 계정 정보 입력

10. 우분투 설치가 완료되면 ‘시스템 재시작Restart Now’을 클릭한다.

11. 시스템 재시작 후 잠시 기다려면 업데이트 매니저Update Manager가 활성화 된다. ‘업데이트 설치 Install updates’를 클릭하면 설치 과정에서 다운로드한 업데이트 파일들이 설치된다.

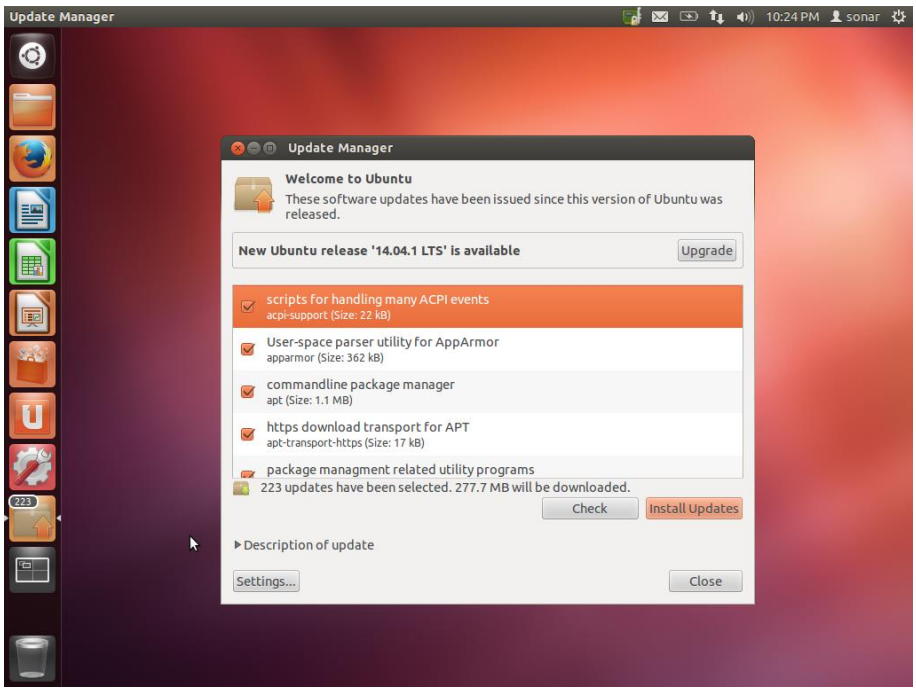


그림 1-18 업데이트 파일 설치

12. 버추얼박스를 설치한 시스템에 최적화된 시스템 요소(각종 시스템 드라이버 등)를 사용하기 위

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

해서는 ‘버추얼박스 게스트 추가팩VirtualBox Guest Addition’을 설치해야 한다. ‘버추얼박스 관리자 메뉴 > Devices…’를 선택하고, ‘Insert Guest Additions CD Image…’를 클릭한다.

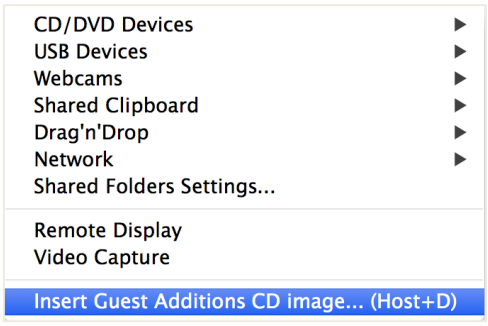


그림 1-19 게스트 추가팩은 기본 제공된다.

13. ‘Run’을 클릭해서 설치를 진행한다.

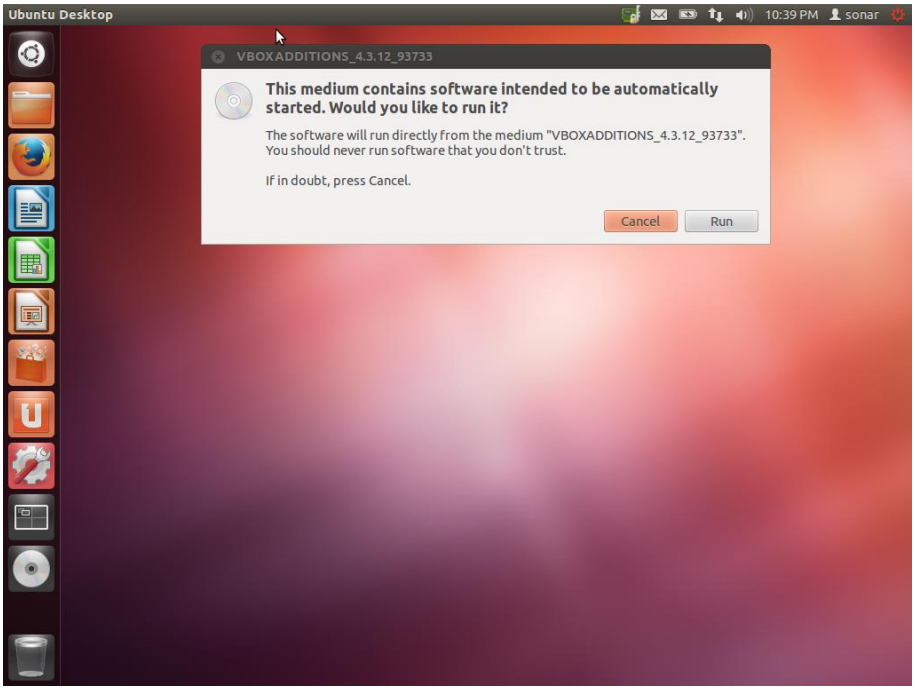


그림 1-20 게스트 추가팩 설치

14. 설치가 완료되면 ‘Restart’를 클릭해서 우분투를 재시작한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

2 소나큐브 설치

이 챕터에서는 소나큐브 서버 및 데이터베이스를 설치하는 방법에 대해 설명한다. 설치 환경은 다음과 같다.

- 오라클Oracle JDK: v1.7.x
- 소나큐브 서버SonarQube Server: v4.4
- 데이터베이스Database: 오라클Oracle MySQL v5.x

2.1 오라클 JDK 설치

소나큐브 서버는 자바 런타임 환경Java Runtime Environment(JRE) v6.x 이상의 자바 가상 머신Java Virtual Machine(JVM)에서 동작한다. 이 책에서는 확장성을 고려해 오라클 자바 개발자 키트Java Development Kit(JDK)를 설치해서 사용한다(JDK에는 JRE가 포함되어 있다).

1. 오라클 홈페이지에서 JDK를 다운로드 받는다.[3]

[3] http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Java SE Development Kit 7u67

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	119.67 MB	jdk-7u67-linux-i586.rpm
Linux x86	136.94 MB	jdk-7u67-linux-i586.tar.gz
Linux x64	120.98 MB	jdk-7u67-linux-x64.rpm
Linux x64	135.78 MB	jdk-7u67-linux-x64.tar.gz
Mac OS X x64	186.01 MB	jdk-7u67-macosx-x64.dmg
Solaris SPARC (SVR4 package)	138.77 MB	jdk-7u67-solaris-sparc.tar.Z
Solaris SPARC	98.61 MB	jdk-7u67-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.99 MB	jdk-7u67-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.39 MB	jdk-7u67-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	24.74 MB	jdk-7u67-solaris-x64.tar.Z
Solaris x64	16.35 MB	jdk-7u67-solaris-x64.tar.gz
Solaris x86 (SVR4 package)	139.39 MB	jdk-7u67-solaris-i586.tar.Z
Solaris x86	95.47 MB	jdk-7u67-solaris-i586.tar.gz
Windows x86	127.98 MB	jdk-7u67-windows-i586.exe
Windows x64	129.7 MB	jdk-7u67-windows-x64.exe

그림 2-1 리눅스용 오라클 JDK 다운로드

2. 우분투에서 터미널을 실행한다.

3. 자바 설치 디렉터리를 생성한다.

```
$ sudo mkdir /usr/local/share/java
```

4. 다운로드 폴더로 이동하여 다운로드받은 파일의 압축을 푼다.

```
$ cd ~/Downloads
$ gzip -d jdk-7u67-linux-i586.tar.gz
$ tar -xvf jdk_7u67_linux_i586.tar
```

5. JDK 디렉터리를 자바 설치 디렉터리로 이동한다.

```
$ sudo mv jdk1.7.0_67 /usr/local/share/java
```

6. 프로파일 스크립트(/etc/profile 혹은 ~/.profile) 파일에 다음 내용을 추가한다. 편집하기 전 프로파일 스크립트 백업 파일을 생성한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

```
$ cp ~/.profile ~/.profile.bak
$ vi ~/.profile

JAVA_HOME=/usr/local/java/jdk1.7.0_67
JRE_HOME=/usr/local/java/jdk1.7.0_67/jre
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
PATH=$PATH:$HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
```

NOTE!! 프로파일 스크립트(/etc/profile 혹은 ~/.profile)는 시스템에서 가장 먼저 실행되는 스크립트 파일이다. 해당 파일을 잘못 수정하는 경우 시스템이 정상적으로 동작하지 않을 수 있다. 리눅스 시스템에 익숙하지 않다면 수정하기 전 반드시 백업 파일을 생성한다.

7. 기본 JDK의 위치를 오라클 JDK 디렉터리로 설정한다.

```
$ sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/local/java/jdk1.7.0_67/jre/bin/java" 1
$ sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/local/java/jdk1.7.0_67/bin/javac" 1
$ sudo update-alternatives --install "/usr/bin/javaws" "javaws"
"/usr/local/java/jre1.7.0_45/bin/javaws" 1
```

8. 오라클 JDK를 기본 JDK로 설정한다.

```
$ sudo update-alternatives --set java
/usr/local/java/jdk1.7.0_67/jre/bin/java
$ sudo update-alternatives --set javac
/usr/local/java/jdk1.7.0_67/bin/javac
$ sudo update-alternatives --set javaws
/usr/local/java/jdk1.7.0_67/bin/javaws
```

9. 추가한 환경 변수를 활성화하기 위해 프로파일 스크립트를 실행한다.

```
$ source ~/.profile
```

10. Java, Javac 버전을 확인한다.

```
$ java -version
java version "1.7.0_67"
$ javac -version
javac 1.7.0_67
```

2.2 소나큐브 서버 설치

다음 과정을 따라 소나큐브 서버를 설치한다,

1. 우분투에서 터미널을 실행한다.

2. 소나큐브 서버를 설치할 디렉터리를 생성한다.

```
$ sudo mkdir /usr/local/share/sonarqube
```

3. 다운로드 폴더로 이동해 소나큐브 서버 배포 패키지를 다운로드[4] 한다.

```
$ cd ~/Downloads
$ wget http://dist.sonar.codehaus.org/sonarqube-4.4.zip
```

4. 다운로드받은 파일에 쓰기 권한을 추가하고 압축을 푼다.

```
$ chmod a+x sonar sonarqube-4.4.zip
$ unzip sonarqube-4.4.zip
```

5. 압축을 푼 소나큐브 서버 디렉터리를 소나큐브 설치 디렉터리로 이동한다.

```
$ sudo mv sonarqube-4.4 /usr/local/share/sonarqube
```

6. 프로파일 스크립트(/etc/profile 혹은 ~/.profile) 파일에 다음 내용을 추가한다. 프로파일 스크립트를 편집하기 전 백업 파일을 만들어 둔다.

[4] http://www.sonarqube.org/downloads/

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

```
$ cp ~/.profile ~/.profile.bak
$ vi ~/.profile

SONAR_HOME=/usr/local/share/sonarqube/sonarqube-4.4
PATH=$PATH:$HOME/bin:$SONAR_HOME/bin/linux-x86-32
export SONAR_HOME
export PATH
```

```
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

JAVA_HOME=/usr/local/share/java/jdk1.7.0_67
JRE_HOME=/usr/local/share/java/jdk1.7.0_67/jre
SONAR_HOME=/usr/local/share/sonarqube/sonarqube-4.4
PATH=$PATH:$HOME/bin:JAVA_HOME/bin
PATH=$PATH:$HOME/bin:JRE_HOME/bin
PATH=$PATH:$HOME/bin:$SONAR_HOME/bin/linux-x86-32
export JAVA_HOME
export JRE_HOME
export SONAR_HOME
export PATH
```

(JAVA_HOME, JRE_HOME은 자바 설치와 관련된 환경 변수임)

7. 추가한 환경 변수를 활성화하기 위해 프로파일 스크립트를 실행한다.

```
$ source ~/.profile
```

8. 소나큐브 서버를 시작시킨다.

```
$ sonar.sh start
```

9. 웹 브라우저를 열고 http://localhost:9000을 방문한다. 소나큐브 화면이 보인다면 소나큐브 서버가 정상적으로 설치된 것이다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

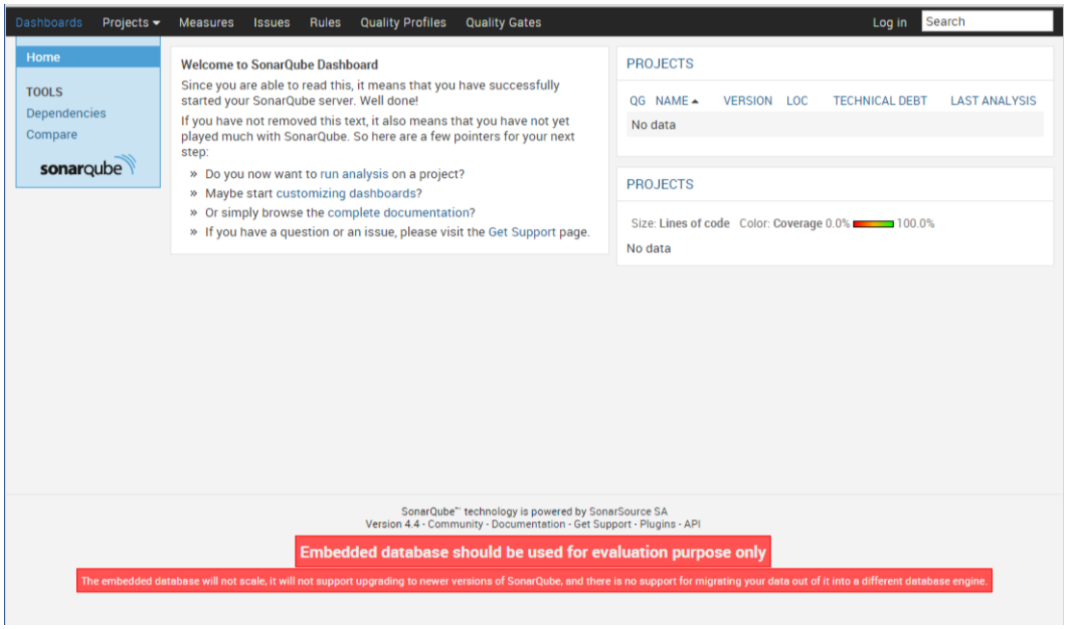


그림 2-2 소나큐브 접속 화면

2.3 소나큐브 서버 환경 설정

소나큐브 서버를 사용하는 필요한 주요 설정값—데이터베이스 접속 정보, 웹 서버 접속 정보, 시스템 업데이트 설정 등—들은 sonar.properties 파일에 정의되어 있다. sonar.properties 파일은 $SONAR_HOME/conf에 위치하고 있다.

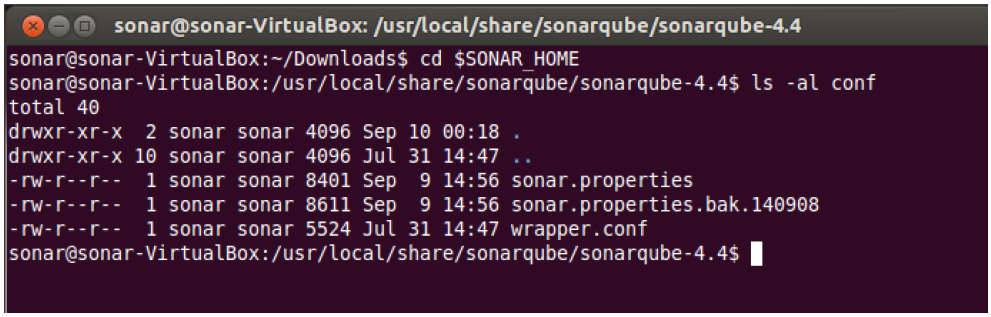


그림 2-3 sonar.properties 파일

이외에도 환경 속성 파일에서 설정할 수 있는 값은 매우 다양하며 간략한 내용을 표 13-1에 표시했다. 환

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

경 속성 파일 설정값과 관련된 보다 자세한 내용은 소나큐브 홈페이지의 도움말[5]을 참조한다.

표 2-1 sonar.properties 파일 속성 리스트

속성값	설명	기본값
sonar.jdbc.username	소나큐브 데이터베이스 접속 계정	sonar
sonar.jdbc.password	소나큐브 데이터베이스 접속 비밀번호	sonar
sonar.jdbc.url	소나큐브가 분석 결과 저장 시 사용할 데이터베이스의 URL	jdbc:h2:tcp://localhost:9092/sonar
sonar.embeddedDatabase.port	소나큐브 내장 데이터베이스의 사용 포트	없음
sonar.web.host	소나큐브 웹 서버의 바인딩 IP 주소. 하나 이상의 IP를 사용하는 서버에 소나큐브가 설치된 경우 소나큐브에 할당할 IP 주소를 입력해야 한다. 기본적으로 소나큐브는 설치된 머신의 모든 IP 주소와 포트를 리스닝한다.	없음
sonar.web.context	웹 서버 컨텍스트 디렉터리 경로	없음
aonar.web.https.port	HTTPS 프로토콜 사용 시, 지정 포트	없음
sonar.web.https.keyAlias	HTTPS - 키스토어Keystore의 서버 인증을 위해 사용. 설정하지 않은 경우 키스토어 리스트의 첫 번째 키 값을 사용한다.	
sonar.web.https	HTTPS - 프로토콜 접속 URL	없음
sonar.web.https.keyPass	HTTPS - 프로토콜의 키를 설정한다.	없음
sonar.web.https.keystoreFile	HTTPS - 서버 인증 키 값을 저장하고 있는 키스토어 파일 경로. 기본적으로 사용자 홈 디렉터리의 “.keystore”를 참조한다.	없음
sonar.web.https.keystorePass	HTTPS - 지정된 키스토어 파일 접근을 위한 비밀번호	없음
sonar.web.https.keystoreType	HTTPS - 서버 인증용 키스토어 파일 종류	JKS(JavaKeyStore)
sonar.web.https.keystoreProvider	HTTPS - 서버 인증용 키스토어 파일 공급자 명칭. 별도로 설정하지 않는 경우 이미 등록된 공급자 리스트를 순서대로 참조하여 해당 키스토어 종류를 지원하는 첫 번째 공급자를 사용한다.	없음

[5] http://docs.codehaus.org/display/SONAR/Analysis+Parameters#AnalysisParameters-Server

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

sonar.web.https.trustStoreFile	HTTPS - 신뢰 인증 권한Trust certificate authority을 포함하고 있는 트러스트스토어Truststore 파일 경로. 기본적으로 JRE에 저장된 파일을 참조한다.	없음
sonar.web.https.trustStorePass	HTTPS - 지정된 트러스트스토어 파일 접근 비밀번호	없음
sonar.web.https.trustStoreType	HTTPS - 사용할 트러스트스토어 파일 종류	JKS
sonar.web.https.trustStoreProvider	HTTPS - 서버 인증 시 사용할 트러스트스토어 공급자 명칭. 별도로 명시하지 않는 경우 기본 등록된 제공자 리스트 순서대로 검색하, 해당 트러스트스토어 파일을 지원하는 첫 번째 공급자를 사용한다.	없음
sonar.web.https.clientAuth	HTTPS - 클라이언트 인증 활성화 여부 . false: 클라이언트 인증 미사용 . want: 인증을 요청하나 선택 사항임 . true: 클라이언트 인증 사용	false
sonar.web.http.maxThreads sonar.web.https.maxThreads	서버에서 최대 수용 가능한 컨넥션 수. 운영체제는 sonar.web.connection.acceptCount 설정값을 기준으로 컨넥션을 허용한다.	50
sonar.web.http.minThreads sonar.web.https.minThreads	항상 실행 상태에 있는 최소한의 스레드 수	5
sonar.web.http.acceptCount sonar.web.https.acceptCount	가용한 모든 스레드가 실행 상태인 경우 외부로부터 들어온 요청을 보관할 수 있는 최대 대기 큐의 값. 대기 한도를 초과한 요청은 버려진다.	25
sonar.web.accessLogs.enable	액세스 로그는 logs/accesslog 디렉터리에 저장되며, 로그 파일은 5MB 단위로 갱신된다.	true
sonar.ajp.port	AJP 인커밍 컨넥션 용도의 TCP 포트 번호. 미사용 시 ‘-1’로 설정한다.	9009
sonar.updatecenter.activate	업데이트 센터 접속 여부. 업데이트 센터를 사용하기 위해서는 http://update.sonarsource.org로 접속 요청을 성공해야 한다.	true
http.proxyHost http.proxyPort	HTTP 프록시 호스트 및 포트	없음
http.auth.ntlm.domain	NTLM 프록시 사용 시, NT 도메인 명	없음
socksProxyHost socksProxyPort	SOCKS 프록시 호스트 및 포트	없음
http.proxyUser http.proxyPassword	프록시 인증 계정 및 비밀번호 HTTP/SOCKS 프록시 모두에 공통으로 적용됨	없음

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

sonar.notifications.delay	알림 큐 처리 사이의 알림 지연 시간(초 단위)	60
sonar.log.profilingLevel	로그에 표시할 정보의 프로파일링 레벨 . NONE(기본) . BASIC(기능 정보) . FULL(기능 및 기술 세부 정보)	없음
sonar.rails.dev	개발자 모드 설정 여부 디버깅 시에만 사용하며, 루비 온 레일즈Ruby on Rails 코드 변경 사항을 디버깅할 경우 true로 설정함	false

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

3 소스 코드 분석

이 챕터에서는 다양한 클라이언트를 사용해서 실제 소스 코드를 분석하는 방법에 대해 설명한다. 소나큐브의 분석 클라이언트는 고유의 특성을 가지고 있으므로 프로젝트 성격이나 목적에 맞게 선택하여 사용한다.

3.1 소나큐브 러너

소나큐브 러너SonarQube Runner는 소나큐브 제작사인 소나소스에서 권장하는 분석 클라이언트이다.

3.1.1 소나큐브 러너 설치

다음 과정을 따라 소나큐브 러너를 설치한다.

1. 우분투에서 터미널을 실행한다.
2. 소나큐브 러너를 설치할 디렉터리를 생성한다.

```
$ sudo mkdir /usr/local/share/sonarqube-runner
```

3. 다운로드 폴더로 이동하여 소나큐브 러너 설치 패키지를 다운로드한다.

```
$ cd ~/Downloads
$ wget
http://repo1.maven.org/maven2/org/codehaus/sonar/runner/sonar-
runner-dist/2.4/sonar-runner-dist-2.4.zip
```

4. 다운로드받은 설치 패키지에 쓰기 권한을 설정한다.

```
$ chmod a+x sonar-runner-dist-2.4.zip
```

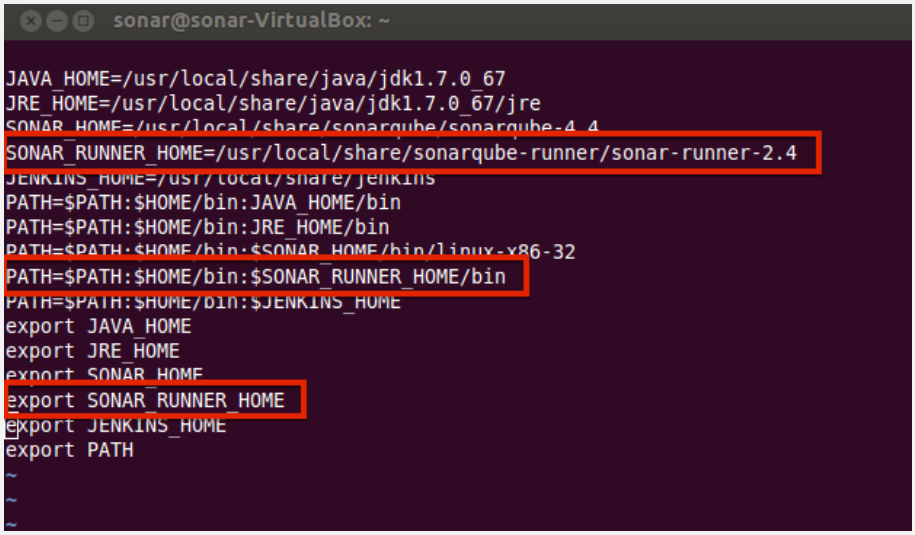
5. 다운로드받은 설치 패키지의 압축을 푼 뒤 소나큐브 설치 디렉터리로 이동시킨다. 소나큐브 러너는 /usr/local/share/sonarqube-runner/sonar-runner-2.4에 설치되며, 이 디렉터리를 {SONAR_RUNNER_HOME}으로 표기한다.

```
$ unzip –d sonar-runner-dist-2.4.zip
$ sudo mv sonar-runner-2.4 /usr/local/share/sonarqube-runner
```

6. 프로파일 스크립트에 소나큐브 러너 설치 디렉터리(SONAR_RUNNER_HOME) 정보를 추가한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

```
$ vi ~/.profile
SONAR_RUNNER_HOME=/usr/local/share/sonarqube-runner/sonar-runner-2.4
PATH=$PATH:$HOME/bin:$SONAR_RUNNER_HOME/bin
export SONAR_RUNNER_HOME
```



7. 프로파일 스크립트를 재적용한다.

```
$ source ~/.profile
```

3.1.2 소나큐브 러너 환경 설정

소나큐브 러너를 사용하기 위해서는 환경 설정 파일인 sonar-runner.properties의 속성값을 설정해야 한다. 이 파일은 {SONAR_RUNNER_HOME}/conf에 위치하고 있다. 소나큐브 서버 환경 파일sonar.properties의 속성 중 서버 주소, 데이터베이스 접속 설정, 기본 인코딩 등을 설정할 수 있다[6].

표 3-1 sonar-runner.properties 파일에 정의 가능한 속성 리스트

속성값	설명	기본값
sonar.host.url	분석 시 사용할 소나큐브 서버의 url 주소	없음
sonar.jdbc.url	분석 시 사용할 데이터베이스 url	없음
sonar.jdbc.username	소나큐브 데이터베이스에 접속할 계정	없음
sonar.jdbc.password	소나큐브 데이터베이스에 접속할 비밀번호	없음
sonar.sourceEncoding	소나큐브 서버가 사용할 소스 코드 인코딩	없음
sonar.login	소나큐브 서버에서 분석 수행 권한을 부여한 사용자의 계정	없음

[6] http://docs.codehaus.org/display/SONAR/Analysis+Parameters#AnalysisParameters-Server

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

sonar.password	소나큐브 서버에서 분석 수행 권한을 부여한 사용자의 비밀번호	없음

3.1.3 소나큐브 분석 프로젝트 환경 설정 sonar-project.properties

소나큐브 러너를 사용하기 위해서는 분석 대상 소스 코드 루트 폴더에 프로젝트 환경 설정 파일(sonar-project.properties)을 생성해 주어야 한다. 표 3-2를 참고하여 프로젝트 분석과 관련된 속성을 설정해 준다.

표 3-2 sonar-project.properties 파일에 정의 가능한 속성 리스트

속성값	설명
sonar.projectKey	소나큐브가 각 프로젝트를 식별하는 고유값. 메이븐 프로젝트인 경우 <groupId>:<artifactId> 속성을 통해 설정한다.
sonar.projectName	소나큐브 웹 인터페이스에 표시할 프로젝트 이름. 메이븐 프로젝트의 경우 <name> 속성을 통해 설정한다.
sonar.projectVersion	프로젝트 버전. 메이븐 프로젝트인 경우 <version> 속성을 통해 설정한다.
sonar.language	분석 대상 언어. 설정값이 없는 경우 다중 언어Multi-language 분석을 수행한다.
sonar.sources	분석 대상 소스 코드를 포함하고 있는 디렉터리. 복수의 디렉터리를 지정하는 경우 각 디렉터리를 콤마(,)로 구분한다.
sonar.projectDescription	프로젝트 설명. 메이븐 스크립트의<description> 속성과는 호환되지 않는다.
sonar.binaries	컴파일된 바이너리 파일을 저장하는 디렉터리. 콤마(,)로 구분된 리스트로 표기할 수 있으며 메이븐 프로젝트에서는 사용할 수 없다.
sonar.tests	테스트 코드를 포함하고 있는 디렉터리. 콤마(,)로 구분된 리스트로 표기할 수 있으며 메이븐 프로젝트에서는 사용할 수 없다.
sonar.libraries	서드파티 라이브러리를 포함하고 있는 디렉터리. 디렉터리를 지정하기 위한 패턴을 사용할 수 있으며 와일드카드 문자(*)는 파일명에만 사용할 수 있다. 예) sonar.libraries=path/to/specific/library/myLibrary.jar,path/to/library/*.jar
sonar.analysis.mode	분석 모드 .analysis(전체 분석 모드) .preview(프리뷰 분석 모드) .incremental(점층 분석 모드)
sonar.preview.readTimeout	프리뷰 분석 모드 사용 시, 특정한 데이터를 서버로부터 로컬 데이터베이스로 읽어 들이는 경우 해당 데이터 읽기가 완료되기까지 대기하는 시간.
sonar.sourceEncoding	소스 코드 파일 인코딩

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

	메이븐 프로젝트인 경우 project.build.sourceEncoding 속성을 사용해 정의할 수 있다. 소나큐브 서버가 설치된 시스템의 자바 가상 머신[7]에 따라 사용 가능한 인코딩이 달라진다.
sonar.importSources	소스 코드 분석 시 소스 코드 복사 여부. 보안상의 이유로 소나큐브 서버에 소스 코드를 저장하지 않거나 컴포넌트 뷰에 소스 코드를 표시하지 않아야 할 경우 false로 설정한다.
sonar.projectDate	분석 결과 날짜 지정. 데이터베이스 마이그레이션 등의 작업을 하면서 이전 프로젝트 정보를 백업하지 않은 경우, 이 속성을 사용해 과거 데이터 히스토리를 남길 수 있다.
sonar.branch	SCM 브랜치 관리. 하나의 소스 프로젝트에 둘 이상의 브랜치가 존재하는 경우 각 브랜치는 별도의 프로젝트로 취급되므로 동일한 이슈가 두 브랜치에 동시에 발생할 수 있다. Developer Cockpit 플러그인[8]을 사용하면 이러한 중복 이슈 발생을 효과적으로 관리할 수 있다.
sonar.profile	소나큐브 서버 환경 파일
sonar.skipDesign	디자인Design/의존도Dependencies 메트릭 분석 여부. Java 언어만 지원한다.
sonar.projectBaseDir	분석 대상 소스 파일 루트 디렉터리. sonar-project.properties 파일의 위치와 소스 코드 파일의 위치가 다른 경우 사용할 수 있다. 분석 과정에서 이 디렉터리에는 sonar.working.diretory가 생성된다.
sonar.working.directory	소스 코드 분석 수행 시 사용하는 작업 디렉터리. 지정된 디렉터리의 내용은 분석 시작과 함께 모두 삭제되므로 각 프로젝트는 고유한 워킹 디렉터리를 사용해야 한다.

3.1.4 프로젝트 분석[9]

다음 과정을 따라 소나큐브 러너를 사용해 프로젝트를 분석한다.

1. 우분투에서 터미널을 실행한다.
2. 프로젝트 형태에 따라 sonar-project.properties 파일을 설정한다. 프로젝트 디렉터리 구조에 따른 sonar-project.properties 파일 기본 설정은 표 3-3을 참조한다.

표 3-3 디렉터리 구조에 따른 sonar--project.properties 설정

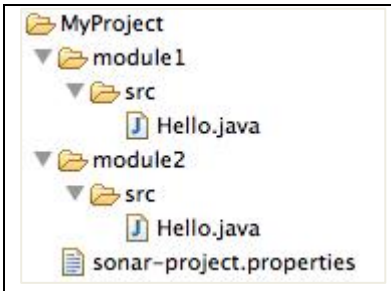
디렉터리 구조	project.properties 파일

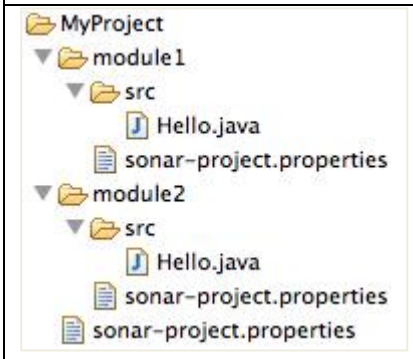
[7] http://docs.oracle.com/javase/1.5.0/docs/guide/intl/encoding.doc.html

[8] http://www.sonarsource.com/products/plugins/developer-tools/developer-cockpit/

[9] http://docs.codehaus.org/display/SONAR/Analyzing+with+SonarQube+Runner

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

	[MyProject/sonar-project.properties]
	
	# 루트 프로젝트 정보
	sonar.projectKey=org.mycompany.myproject
	sonar.projectName=My Project
	sonar.projectVersion=1.0
	
	# 하위 모듈에서 상속할 속성들
	sonar.sources=src
	
	# 모듈 식별자 리스트
	sonar.modules=module1,module2
	
	# 각 모듈에서 오버라이드할 속성. 모듈명을 가장 앞에 붙여야 함
	module1.sonar.projectName=Module 1
	module2.sonar.projectName=Module 2

	[MyProject/sonar-project.properties]
	
	# 루트 프로젝트 관련 정보
	sonar.projectKey=org.mycompany.myproject
	sonar.projectName=My Project
	sonar.projectVersion=1.0
	
	# 하위 모듈에서 상속할 속성들
	sonar.sources=src
	
	# 모듈 식별자 리스트
	sonar.modules=module1,module2
	
	[MyProject/module1/sonar-project.properties]
	# 속성 재정의. 모듈명을 prefix로 사용할 필요는 없음
	sonar.projectName=Module 1
	
	[MyProject/module2/sonar-project.properties]
	# 속성 재정의. 모듈명을 prefix로 사용할 필요는 없음
	sonar.projectName=Module 2

3. 소나큐브 러너를 실행한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

```
$ sonar-runner
```

3.2 메이븐

메이븐을 사용하여 프로젝트를 관리하고 있다면 몇 가지 간단한 설정을 추가하는 것만으로 소나큐브를 사용할 수 있다. 메이븐과 관련된 자세한 설명은 메이븐 웹사이트[10]를 참조한다.

3.2.1 메이븐 설치

메이븐을 처음 사용하는 경우 다음 과정을 따라 메이븐을 설치한다. 이미 메이븐을 사용하고 있다면 다음 섹션인 메이븐 환경 설정으로 바로 넘어간다.

1. 우분투에서 터미널을 실행한다.
2. 메이븐을 설치할 디렉터리를 생성한다.

```
$ sudo mkdir /usr/local/share/maven
```

3. 다운로드 디렉터리로 이동하여 메이븐 설치 파일을 다운로드한다.

```
$ cd ~/Downloads
$ wget http://mirror.apache-kr.org/maven/maven-
3/3.2.3/binaries/apache-maven-3.2.3-bin.zip
```

4. 다운로드받은 메이븐 설치 파일에 쓰기 접근 권한을 추가하고 압축을 해제한다.

```
$ chmod a+x apache-maven-3.2.3-bin.zip
$ unzip apache-maven-3.2.3-bin.zip
```

5. 압축을 해제한 메이븐 설치 파일을 메이븐 설치 디렉터리로 이동한다.

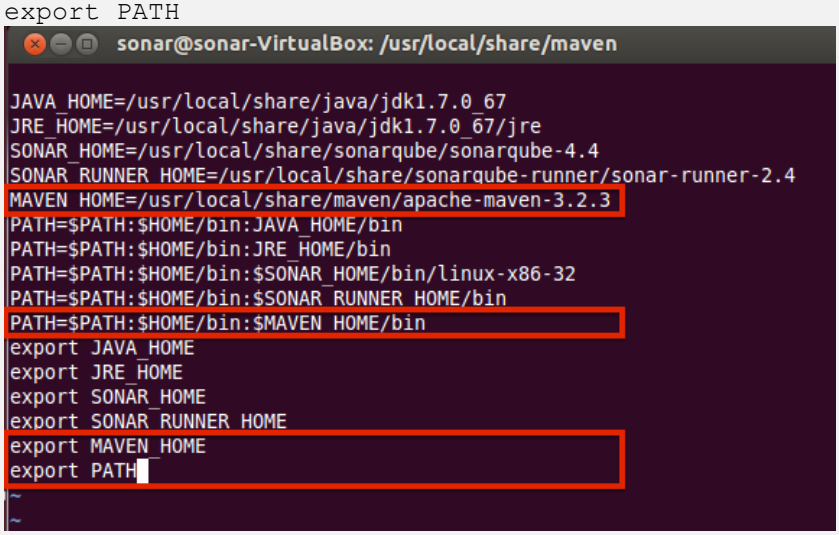
```
$ sudo mv apache-maven-3.2.3 /usr/local/share/maven
```

6. 메이븐 설치 관련 정보를 스크립트 프로파일에 추가한다.

[10] 아파치 메이븐 웹사이트: http://maven.apache.org/

```
$ vi ~/.profile

MAVEN_HOME=/usr/local/maven/apache-maven-3.2.3
PATH=$PATH:$HOME/bin:$MAVEN_HOME/bin
export MAVEN_HOME
export PATH
```

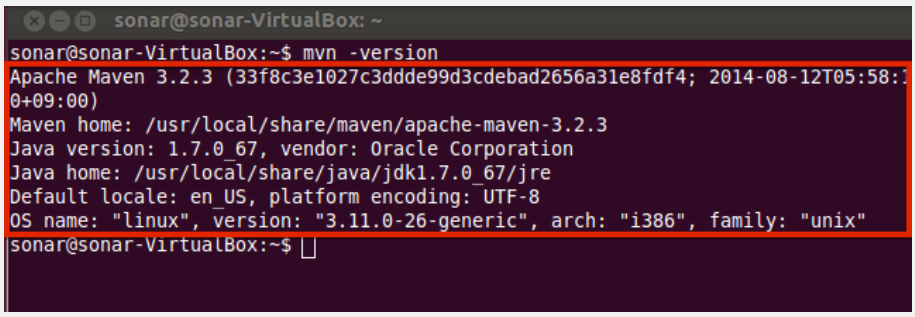


7. 프로파일 스크립트를 재실행한다.

```
$ source ~/.profile
```

8. 설치된 메이븐 버전을 확인한다.

```
$ mvn –version
```



3.2.2 메이븐 환경 설정

메이븐 스크립트에서 소나큐브를 사용하기 위해서는 메이븐 환경 설정 파일(settings.xml)과 프로젝트 설정 파일(pom.xml)에 간단한 설정을 추가해야 한다.

메이븐 환경 설정 파일은 ${MAVEN_HOME}/conf 혹은 ~/.m2 디렉터리에 위치하고 있으며, 프로젝트 설정 파일은 프로젝트 루트 디렉터리에 위치하고 있다. 다음 과정을 따라 소나큐브 연동 환경을 설정한다.

1. 우분투에서 터미널을 실행한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

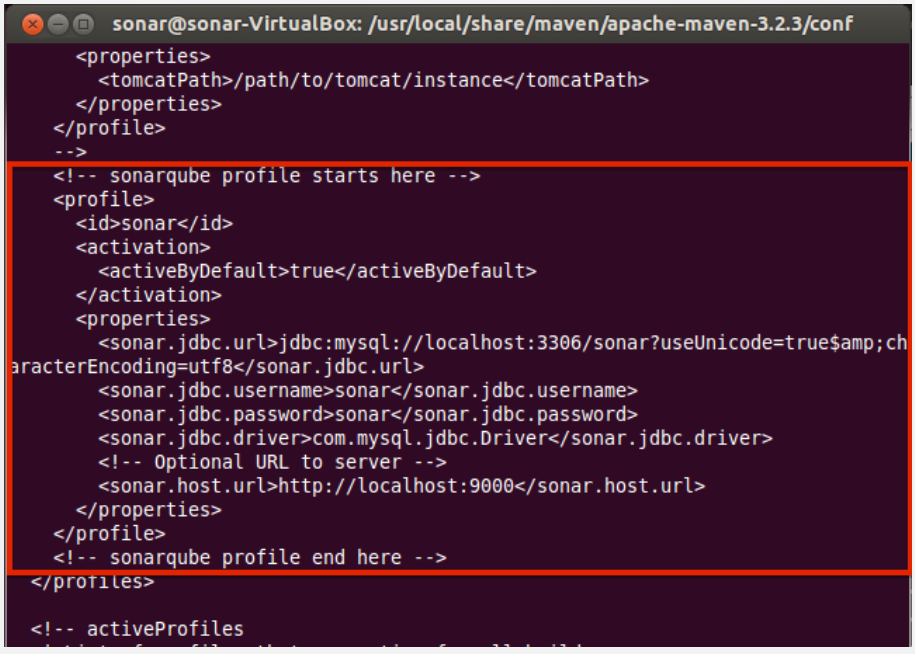
2. 메이븐이 설치된 디렉터리로 이동한다.

```
$ cd /usr/local/share/maven/apache-maven-3.2.3/conf
```

3. 메이븐 환경 설정 파일(settings.xml)에 다음 내용을 추가한다.

```
$ vi settings.xml

<profile>
<id>sonar</id>
<activation>
<activeByDefault>true</activeByDefault>
</activation>
<properties>
<sonar.jdbc.url>jdbc:mysql://localhost:3306/sonar?useUnicode=true&am
p;characterEncoding=utf8</sonar.jdbc.url>
      <sonar.jdbc.username>sonar</sonar.jdbc.username>
<sonar.jdbc.password>sonar</sonar.jdbc.password>
      <!-- Optional URL to server -->
<sonar.host.url>http://localhot:9000</sonar.host.url>
</properties>
</profile>
```



4. 소나큐브로 분석할 프로젝트의 루트 디렉터리로 이동한다(이 섹션에서는 샘플 프로젝트의 java-maven-simple을 분석한다).

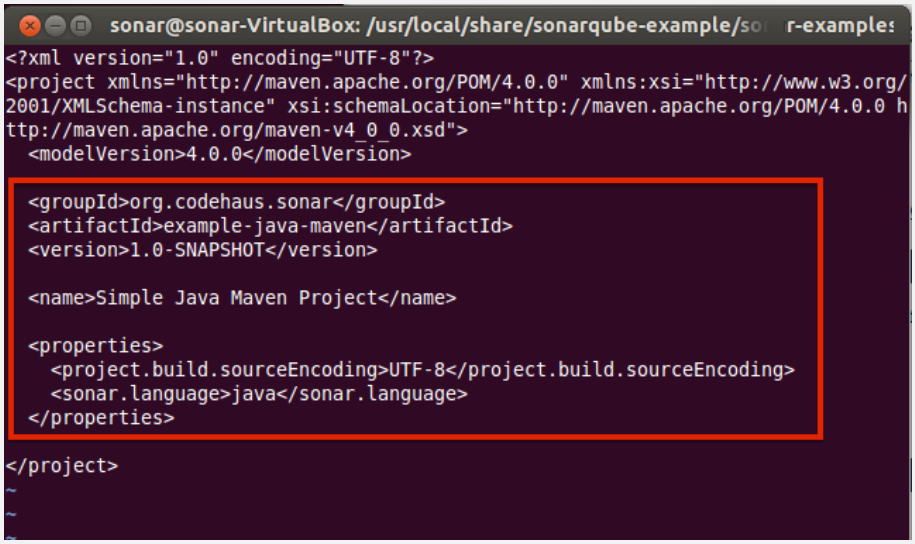
```
$ cd /usr/share/local/sonarqube-examples/sonar-examples-
master/projects/languages/java/maven/java-maven-simple
```

5. 프로젝트 설정 파일(pom.xml)에 다음을 추가한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

```
$ vi pom.xml

<groupId>org.codehaus.sonar</groudId>
<artifactId>example-java-maven</artifactId>
<version>1.0-SNAPSHOT</version>
<name>Simple Java Maven Project</name>
<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<sonar.language>java</sonar.language>
</properties>
```



3.2.3 프로젝트 분석

다음 과정을 따라 소나큐브를 사용해 프로젝트를 분석한다.

1. 우분투에서 터미널을 실행한다.
2. 소나큐브로 분석할 프로젝트의 루트 디렉터리로 이동한다(이 섹션에서는 샘플 프로젝트의 java-maven-simple을 분석한다).

```
$ cd /usr/share/local/sonarqube-examples/sonar-examples-
master/projects/languages/java/maven/java-maven-simple

$ mvn clean install
$ mvn sonar:sonar

혹은

$ mvn clean install sonar:sonar
```

3. 웹 브라우저를 열고 http://localhost:9000에 접속하여 분석 결과를 확인한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

메이븐-소나큐브 연동과 관련된 더 자세한 내용은 메이븐 도움말[11]과 소나큐브 웹사이트 도움말[12]을 참조한다.

[11] http://maven.apache.org/guides/getting-started/index.html

[12] http://docs.codehaus.org/display/SONAR/Installing+and+Configuring+Maven

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

4 지속적인 통합 환경과 소나큐브 연동하기

이 챕터에서는 오픈소스 CI 서버인 젠킨스 소나큐브와 연동하는 방법을 설명한다.

> NOTE!! 빌드 과정 중이나 빌드 작업이 완료된 이후 커맨드 라인 명령어를 실행할 수 있는 CI 서버라면 소나큐브와 연동이 가능하다.[13] 소나큐브 도움말 페이지에서 보다 자세한 정보를 확인할 수 있다.

4.1 젠킨스 설치

젠킨스Jenkins CI 서버(이하 젠킨스)는 JDK가 설치된 운영체제라면 대부분의 운영체제에 설치하여 즉시 사용 가능하다. 소나큐브와 마찬가지로 여러 가지 플러그인을 사용해 빌드 이외에 다양한 기능을 제공한다.

다음 과정을 따라 젠킨스를 설치한다.

1. 우분투에서 터미널을 실행한다.
2. 젠킨스를 설치할 디렉터리를 생성한다.

```
$ sudo mkdir /usr/local/share/jenkins
```

3. 다운로드 폴더로 이동해서 젠킨스 설치 파일을 다운로드한다.

```
$ cd ~/Downloads
$ wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war
```

4. 다운로드받은 젠킨스 설치 파일에 쓰기 권한을 추가하고 젠킨스 설치 디렉터리로 이동시킨다.

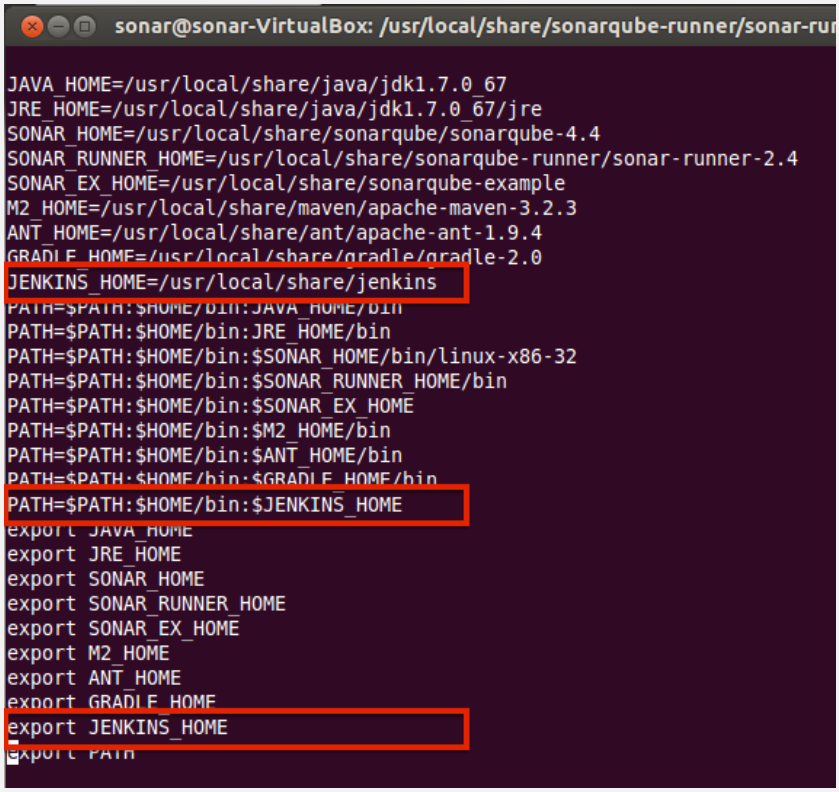
```
$ chmod a+x jenkins.war
$ sudo mv jenkins.war /usr/local/share/jenkins
```

5. 젠킨스 설치 관련 정보를 프로파일 스크립트에 추가한다.

[13] http://mng.bz/43GI

```
$ sudo vi ~/.profile

JENKINS_HOME=/usr/local/share/jenkins
PATH=$PATH:$HOME/bin:$JENKINS_HOME
export JENKINS_HOME
export PATH
```



6. 프로파일 스크립트를 재적용한다.

```
$ source ~/.profile
```

7. 젠킨스를 시작시킨다(기본적으로 8080 포트를 사용하며 기본 포트를 변경하고 싶은 경우 ‘--httpPort={PORT_NUMBER}’ 옵션을 사용한다).

```
$ cd /usr/local/share/jenkins
$ java -jar jenkins.war --httpPort=8080
```

8. 웹 브라우저를 열고 http://localhost:8080로 접속한다. 젠킨스가 정상적으로 설치되면 그림 4-1과 같은 화면이 표시된다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

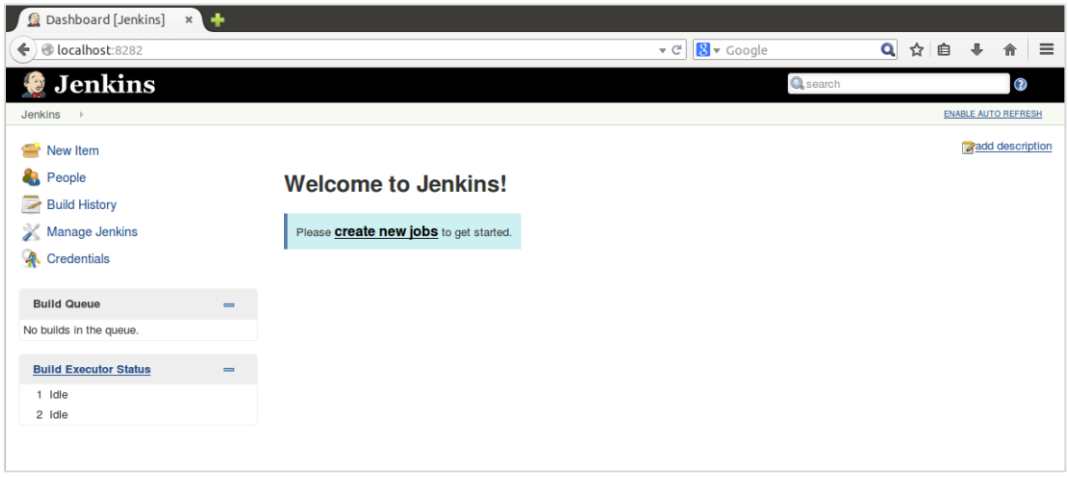


그림 4-1 젠킨스 접속 화면

4.2 젠킨스 설정

젠킨스와 소나큐브를 연동하기 위해서는 몇 가지 추가적인 설정을 해주어야 한다.

- 젠킨스에 소나큐브 플러그인 설치
- 젠킨스에 소나큐브 서버 관련 정보 입력
- 젠킨스에 소나큐브 클라이언트(소나큐브 러너, 메이븐, 앤트 등) 관련 정보 입력
- 젠킨스 Job에 소나큐브 추가

4.2.1 소나큐브 플러그인 설치

젠킨스 서버 및 소나큐브가 모두 정상적으로 동작하고 있다면 다음 과정을 따라 젠킨스에 소나큐브 플러그인을 설치한다.

1. 젠킨스 서버에 접속한다(예: http://localhost:8080).
2. ‘젠킨스 관리Mange Jenkins > 플러그인 관리Manage Plugins > 설치 가능Available’ 으로 이동한다. Filter에 ‘sonar’를 입력하고 ‘Sonar plugin’을 선택한다.
3. ‘ Install’ 체크 박스를 선택하고 ‘다운로드받은 뒤 시스템 재시작 후 적용Download Now and Install After Restart’을 클릭한다. 플러그인 다운로드, 설치 및 시스템 재시작이 진행된다.

4.2.2 소나큐브 서버 정보 입력

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

플러그인 설치를 완료했다면 다음 과정을 따라 소나큐브 서버 설치 정보를 젠킨스에 입력한다.

1. ‘Manage Jenkins > Configure System > Sonar’로 이동한다.
2. ‘Sonar installations’에서 ‘Add sonar’를 클릭하고 다음 항목들을 설정한다(그림 4-2 참조).

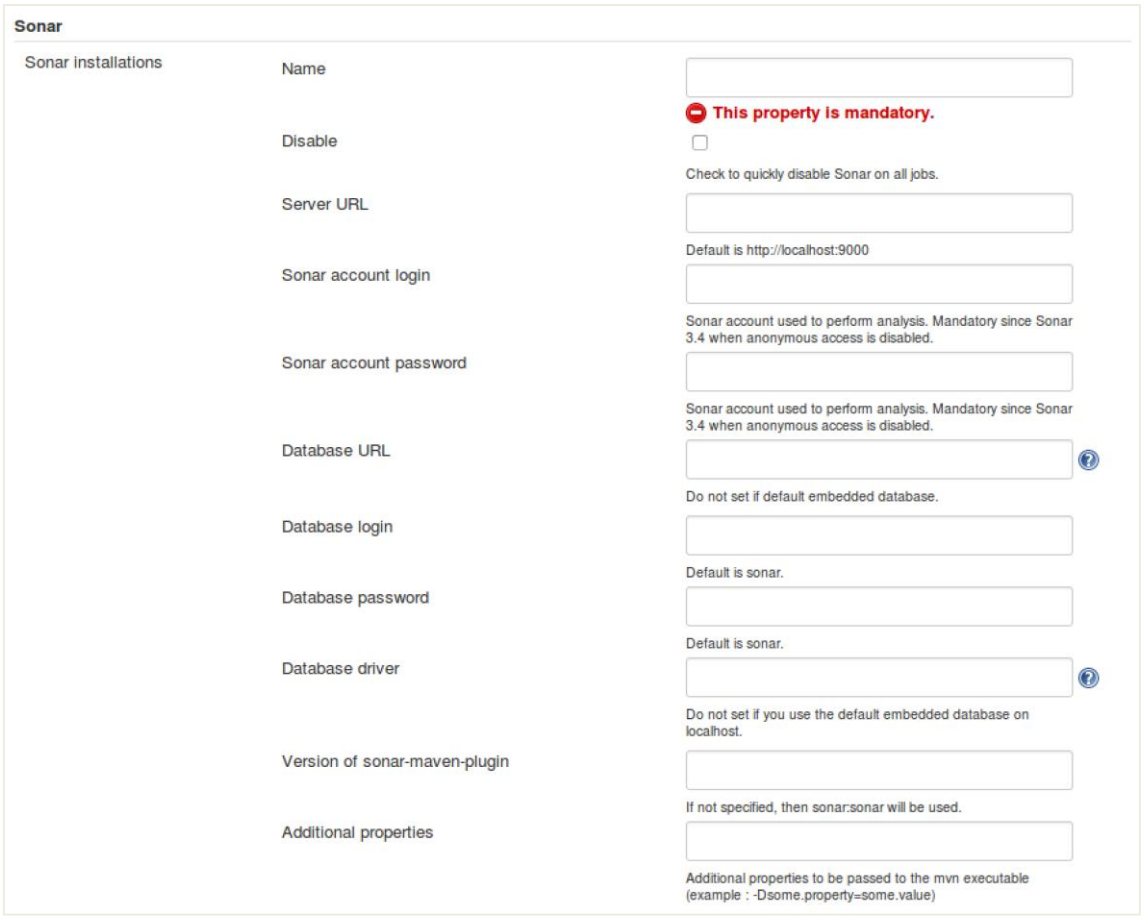


그림 4-2 소나큐브 서버 설정

- Name: 소나큐브 서버의 이름을 입력한다. 이 이름은 이후 젠킨스 내에서 Job 설정 시 사용한다
- Disable: 소나큐브 서버를 비활성화하는 경우 체크한다.
- Server URL: 소나큐브 서버가 설치된 URL을 입력한다(예: http://localhost:9000 등).
- Sonar account login: 소나큐브 접속 계정을 입력한다. 이 계정은 소나큐브 데이터베이스에 저장된 계정으로 소스 코드 분석 실행 권한을 가지고 있어야 한다.
- Sonar account password: 위 계정의 비밀번호를 입력한다.
- Server Public URL: 소나큐브 서버가 퍼블릭 URL을 가진 경우 입력한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

- Database URL: 소나큐브가 사용하는 데이터베이스의 URL을 입력한다.
- Database login: 소나큐브가 사용하는 데이터베이스의 접속 계정을 입력한다.
- Database password: 소나큐브가 사용하는 데이터베이스의 접속 계정 비밀번호를 입력한다.
- Database driver: 소나큐브가 사용하는 데이터베이스의 드라이버를 입력한다.
- version of sonar-maven-plugin: 소나큐브-메이븐 플러그인 버전을 입력한다.
- Additional properties: 기타 소나큐브 서버 설정과 관련된 추가 속성을 설정할 경우 입력한다.

3. 트리거 배제Trigger exclusion 옵션을 설정한다. 이 옵션은 소나큐브 분석으로 포스트 빌드 액션 스텝(메이븐과 함께 사용 가능)에 추가하는 경우에만 유효하다.

그림 4-3 트리거 배제 옵션 설정

- Skip if triggered by SCM changes: SCM 변경 옵션에 의해 빌드가 트리거링되는 경우 코드 품질 분석을 수행하지 않는다.
- Skip if triggered by the build of a dependency: 디펜던시 변경에 의해 빌드가 트리거링되는 경우 코드 품질 분석을 수행하지 않는다.
- Skip if environment variable is defined and set to true: 특정 환경 변수가 true 설정됨에 따라 빌드가 트리거링되는 경우 코드 품질 분석을 수행하지 않는다.

4. ‘Save’ 혹은 ‘Apply’를 클릭해서 설정 내용을 저장한다.

4.2.3 소나큐브 클라이언트 정보 입력

소나큐브 서버 정보를 입력했다면 다음으로 소나큐브 분석 시 사용할 클라이언트 정보를 입력해야 한다.

4.2.3.1 소나큐브 러너

다음 과정을 따라 소나큐브 러너 설치 정보를 입력한다.

1. ‘Manage Jenkins > Configure System > Sonar Runner’로 이동한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

2. ‘Sonar Runner’에서 ‘Add Sonar Runner’를 클릭하고 다음 항목들을 설정한다.

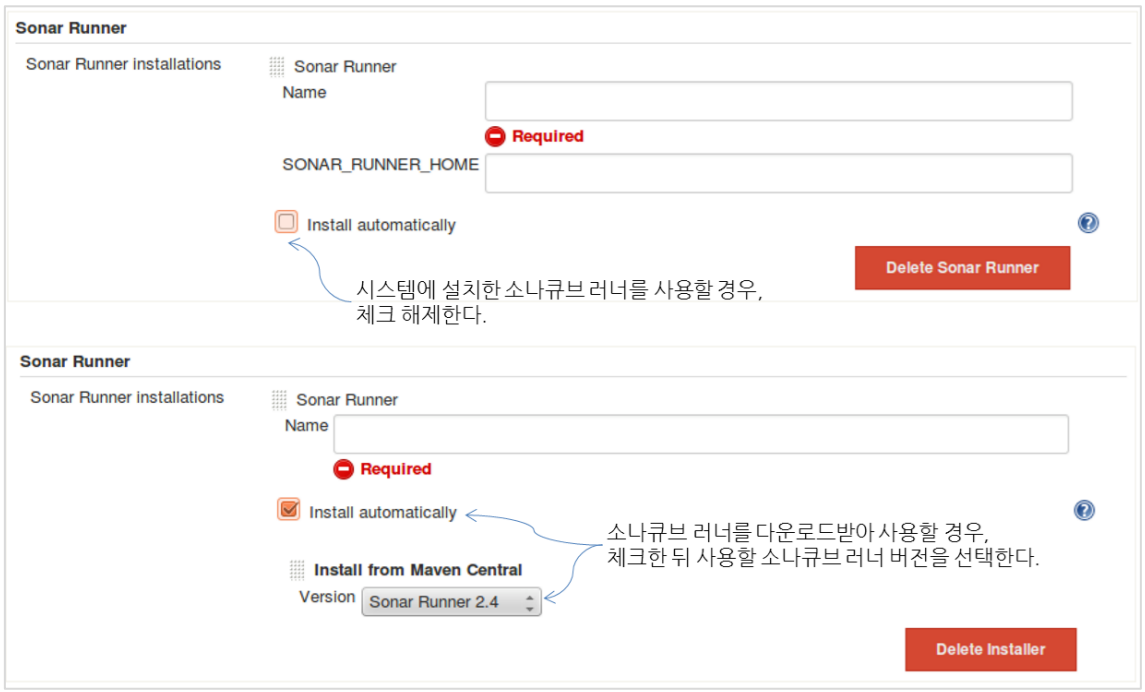


그림 4-4 소나큐브 러너 설정

- Name: 소나큐브 러너 인스턴스의 이름을 입력한다. 이 이름은 이후 젠킨스 내에서 Job 설정 시 사용한다.
- SONAR_RUNNER_HOME: 젠킨스가 설치된 동일 시스템에 설치된 소나큐브 러너를 사용하는 경우 소나큐브 러너가 설치된 경로를 입력한다.
- Install automatically: 코드 품질 분석 수행 시 소나큐브 러너를 다운로드받아 사용한다.
- Install from Maven Central: ‘Install automatically’에 체크한 경우 메이븐 센트럴에서 다운로드 받을 소나큐브 버전을 입력한다.

3. ‘Save’ 혹은 ‘Apply’를 클릭하여 설정 내용을 저장한다.

4.2.3.2 메이븐

다음 과정을 따라 메이븐 설치 관련 정보를 젠킨스에 입력한다.

1. ‘Manage Jenkins > Configure System’으로 이동한다.
2. ‘Maven’에서 ‘Add Maven’를 클릭하고 다음을 항목들을 입력한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

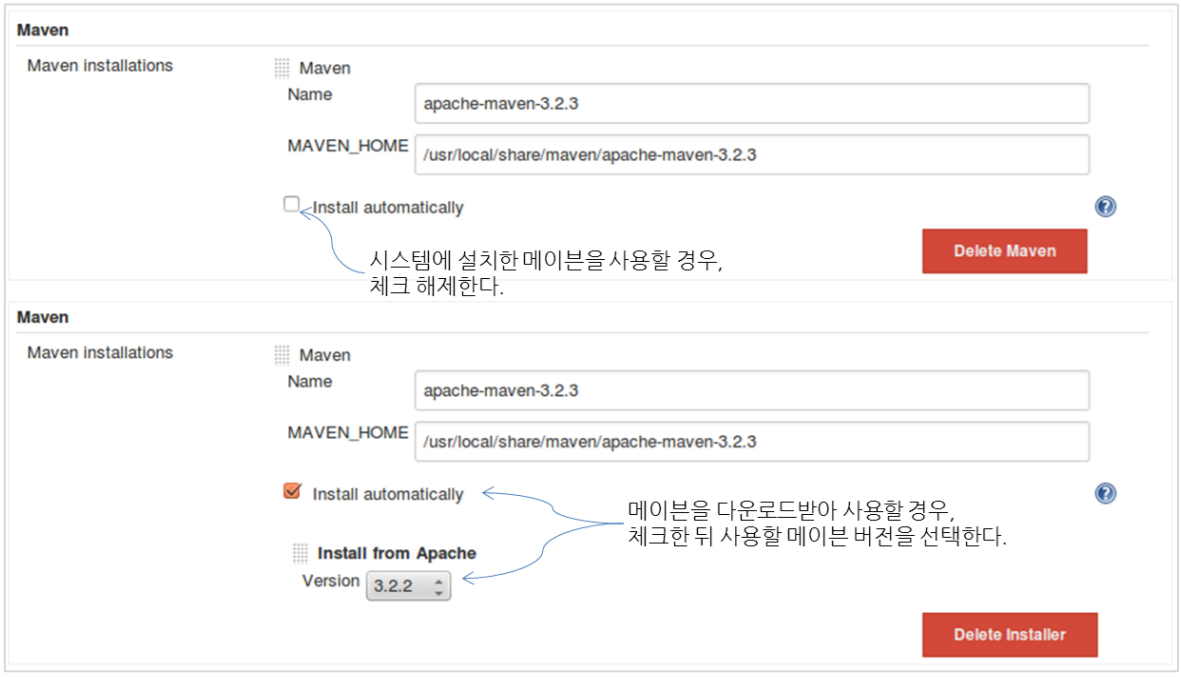


그림 4-5 메이븐 설치 관련 정보 설정하기

- Name: 메이븐 인스턴스 이름을 입력한다. 이 이름은 이후 젠킨스 내에서 Job 설정 시 사용한다
- MAVEN_HOME: 젠킨스가 설치된 동일 시스템에 설치된 메이븐을 사용하는 경우 메이븐이 설치된 경로를 입력한다.
- Install automatically: 코드 품질 분석 수행 시 메이븐을 다운로드받아 사용한다.
- Install from Apache: ‘Install automatically’에 체크한 경우 아파치에서 다운로드받을 메이븐 버전을 설정한다.

3. ‘Save’ 혹은 ‘Apply’를 클릭해서 설정 내용을 저장한다.

4.2.4 젠킨스 Job 설정

소나큐브로 분석을 수행하기 위해 새로운 Job을 만들거나 이미 존재하는 Job에 소스 코드 분석 단계를 추가할 수 있다. 일부 프로그래밍 언어—Java, C# 등—는 바이트 코드Byte-code코드를 품질 분석에 사용하기 때문에 반드시 컴파일을 먼저 수행해야 한다.

4.2.4.1 소나큐브 러너

1. 젠킨스에서 새로운 Job을 생성하거나 소나큐브 분석을 적용할 Job을 선택하고 ‘Configuration’ 메뉴를 선택한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

2. ‘Add build step’에서 ‘Invoke Standalone Sonar Analysis’를 선택한다.

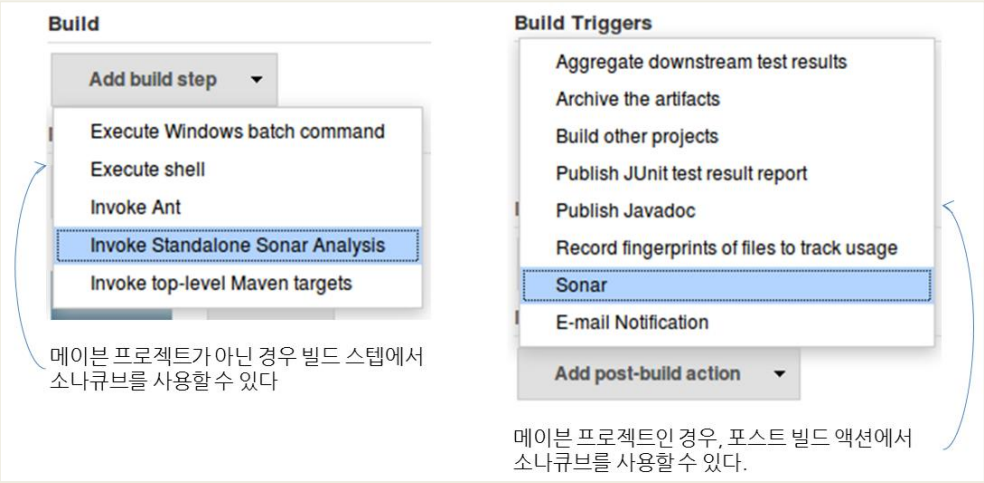


그림 4-6 소나큐브 러너를 추가한다

3. 소나큐브 러너 실행을 위한 옵션을 설정한다.

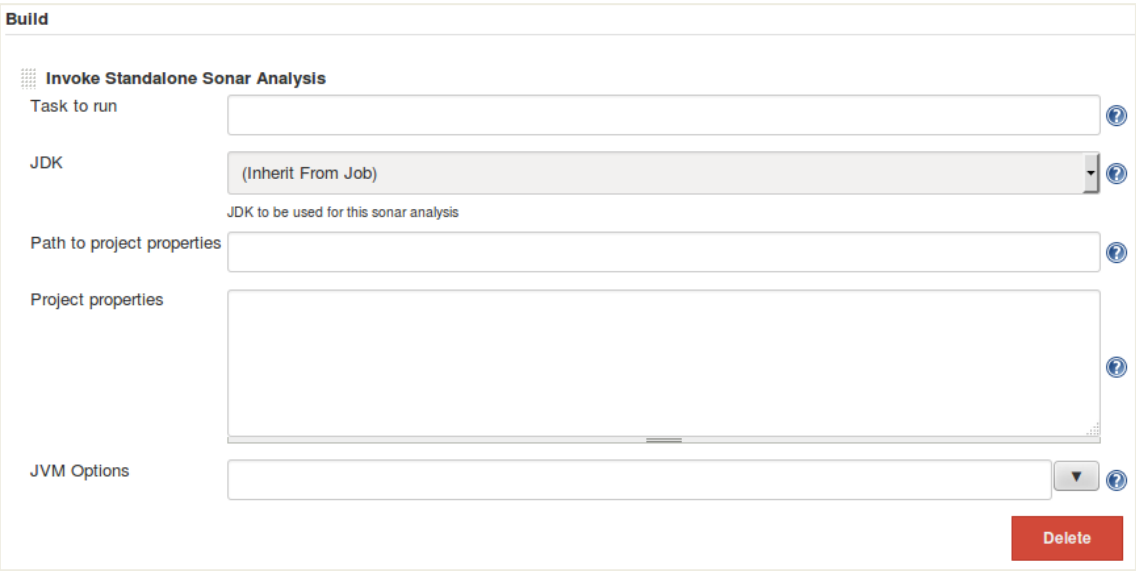


그림 4-7 소나큐브 러너를 사용하여 분석하기

- 실행할 태스크Task to Run: 실행할 태스크. 소나큐브 러너 v2.1 이상에서 지원한다(선택사항)
- JDK: 소나큐브가 사용할 JDK 버전을 선택한다. 젠킨스 CI 서버에 여러 가지 버전의 JDK가 설치되어 있다면 드롭다운 리스트에서 사용할 JDK를 선택할 수 있다.
- 프로젝트 속성 파일 경로Path to project properties: sonar-runner.properties 파일의 경로를 입력한다. 공란으로 둘 경우 선택한 소나큐브 러너에 정의된 기본 속성 파일을 사용한다.
- 프로젝트 속성Project properties: 소나큐브 러너 속성 파일에 설정되어 있는 속성 이외에 추가 속성이 필

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

요한 경우 입력한다. 커맨드 라인 옵션 파라미터와 동일하게 사용할 수 있다.

- 자바 가상 머신JVM Options: 자바 가상 머신과 관련된 추가 파라미터를 입력한다(선택).

4. ‘Save’ 혹은 ‘Apply’를 클릭해서 설정값을 저장한다.

4.2.4.2 메이븐

메이븐을 사용하는 경우 두 가지 방법으로 소나큐브를 사용할 수 있다(그림 4-8 참조).

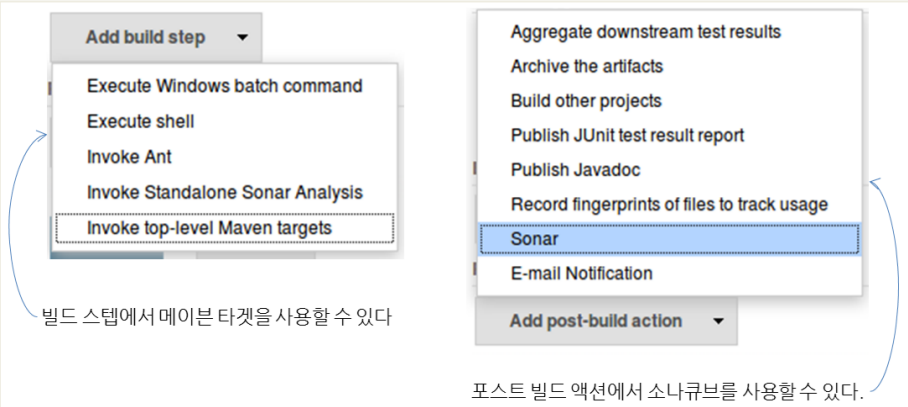


그림 4-8 메이븐을 사용한 소나큐브 분석 추가

빌드 스텝

1. 젠킨스에서 새로운 Job을 생성하거나 소나큐브 분석을 적용할 Job을 선택하고 ‘Configuration’ 메뉴를 선택한다.

2. ‘Add build step’ > ‘Invoke top-level Maven targets’를 클릭하고 다음 항목들을 입력한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

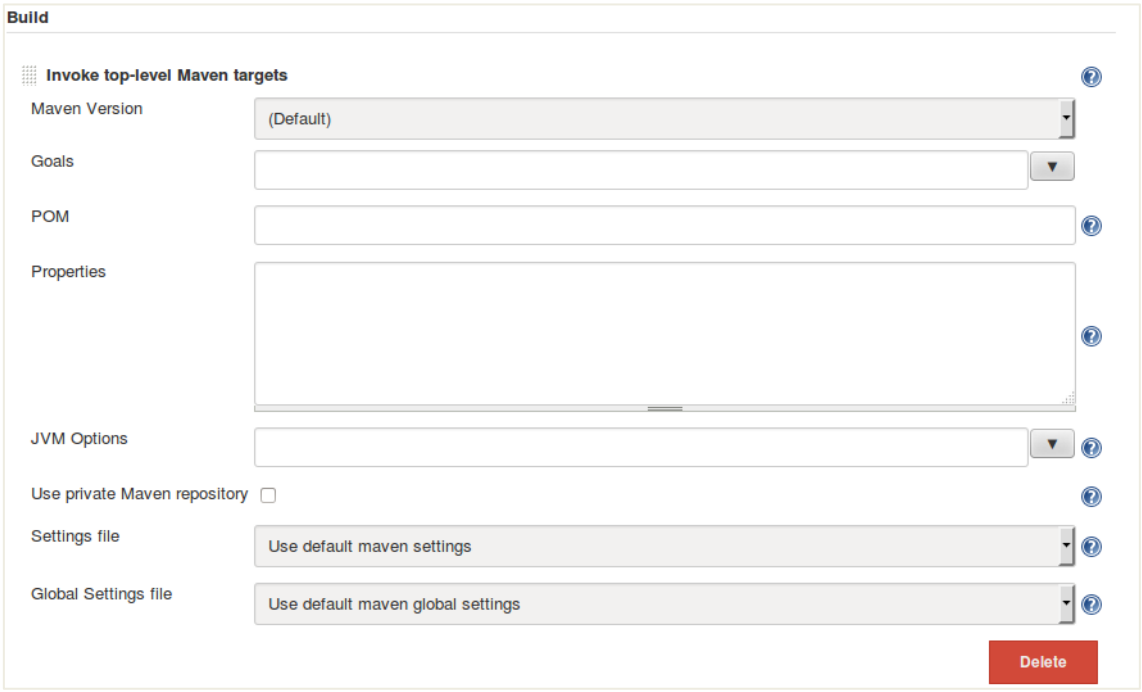


그림 4-9 메이븐을 사용하여 분석하기

- Maven Version: 메이븐 버전을 선택한다.
- Goals: 메이븐 스크립트에서 수행할 골을 선택한다.
- POM: pom.xml 파일의 경로를 입력한다.
- Properties: 추가적인 메이븐 스크립트 옵션을 추가할 수 있다.
- JVM Option: 메이븐과 관련된 자바 가상 머신의 추가 옵션을 설정할 수 있다.
- Use private Maven repository: 개인 메이븐 리파지토리를 사용할 경우 체크한다.
- Settings file: 어떤 settings.xml 파일을 사용할지 선택한다.
- Global settings file: 글로벌 환경 설정 파일로 어떤 settings.xml 파일을 사용할지 선택한다.

3. ‘Save’ 혹은 ‘Apply’를 클릭해서 설정값을 저장한다.

포스트 빌드 스텝

1. 젠킨스에서 새로운 Job을 생성하거나 소나큐브 분석을 적용할 Job을 선택하고 ‘Configuration’ 메뉴를 선택한다.

2. ‘Add post-build action’ > ‘Sonar’를 선택하고 다음 항목들을 입력한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

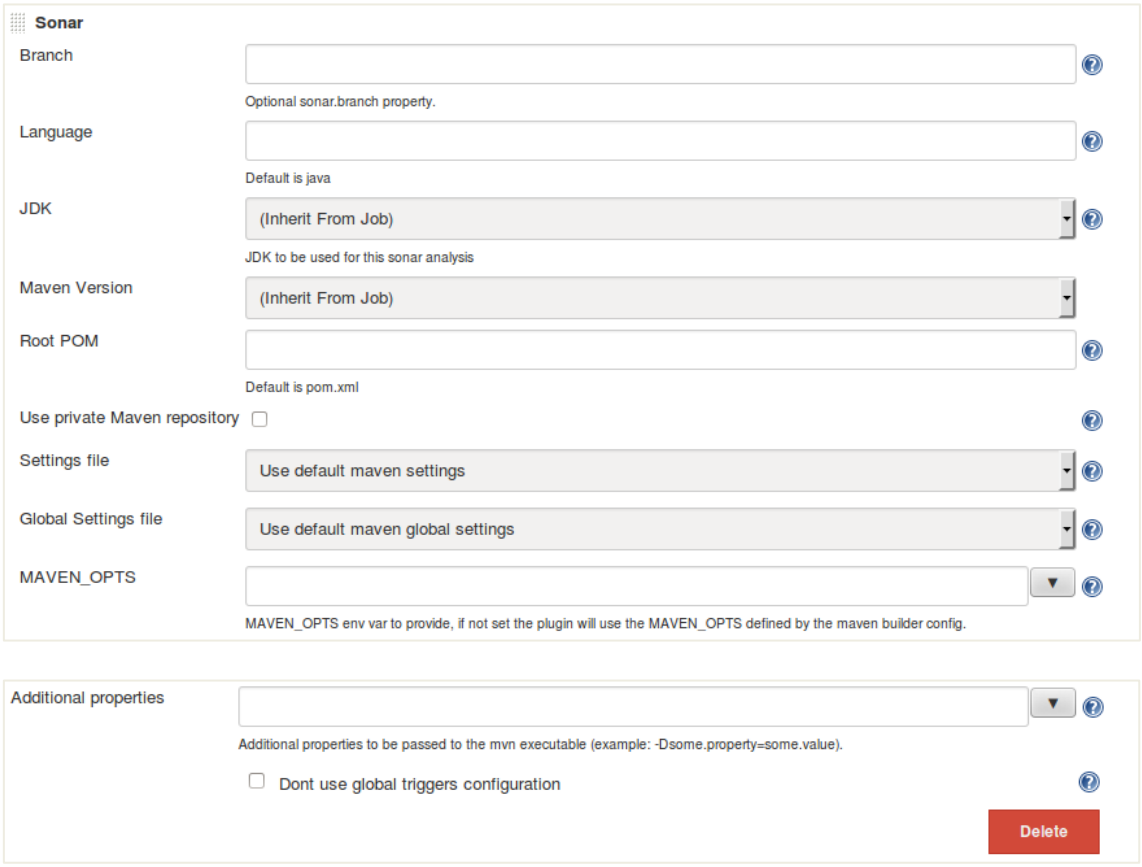


그림 4-10 메이븐 프로젝트 분석 설정

- Branch/Language: SCM 브랜치 혹은 Java가 아닌 프로그래밍 언어를 사용한 경우 설정한다.
- JDK Version: 소나큐브가 사용할 JDK 버전을 선택한다.
- Maven Version: 소나큐브가 사용할 메이븐 버전을 선택한다.
- Root POM: 기본 메이븐 폼 파일(pom.xml)을 사용하지 않을 경우 사용할 폼 파일의 위치를 지정한다.
- User private Maven repository: 개인 메이븐 리파지토리를 사용한다.
- Settings file: 기본 메이븐 환경 설정 파일(settings.xml)을 사용하지 않을 경우 사용할 환경 설정 파일을 선택한다.
- Global settings file: 글로벌 메이븐 환경 설정 파일(settings.xml)을 사용하지 않을 경우 사용할 환경 파일을 선택한다.
- MAVEN_OPTS: 메이븐 커맨드 라인 옵션을 설정한다.
- Additional properties: 소나큐브 분석 시 필요한 추가 속성을 입력한다. ‘-Pintegration-tests -

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)

Dappserver=tomcat -Ddatabase=mysql’과 같은 형태로 입력할 수 있다.

- Don’t use global triggers configuration: 글로벌 트리거 환경을 무시하고 싶은 경우 선택한다.

출처: 코드 품질 시각화의 정석~지속적인 소프트웨어 품질 분석과 관리를 위한 SonarQube 완벽가이드~
(ISBN-13 978-89-93827-93-4)