

SW V&V

CTIP 환경 구축

3조

박동현

박성호

박현우

안찬우

Contents

Java IDE

Auto Build

Unit Testing

Issue Tracking

Communication

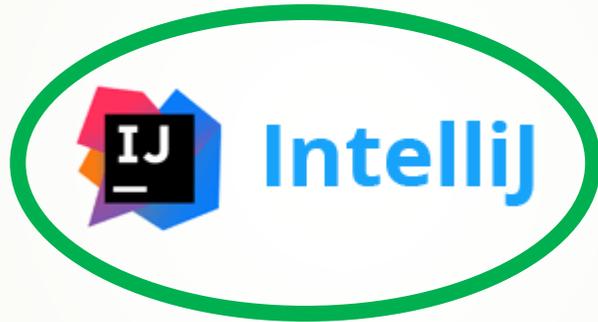
CI

Slack Notification

Overview

Java IDE

- ▶ 가독성 좋은 UI
- ▶ Eclipse에 비해 비교적 가벼움
- ▶ 유료 vs 무료 (but 학생 무료)



VS



Auto Build



VS

MavenTM

- ▶ 가독성
- ▶ 편리한 관리
- ▶ 성능

Gradle Implement

```
plugins {
    id 'java'
}

group 'com.sample.gradletest'
version '1.0-SNAPSHOT'

repositories {
    mavenCentral()
}

jar {
    manifest {
        attributes 'Title': 'GradleTest', 'Version': '0.1', 'Main-Class': 'Main'
    }
    archiveName 'GradleTest.jar'
    from {
        configurations.compile.collect { file(it)
            it.isDirectory() ? it : zipTree(it)
        }
    }
}

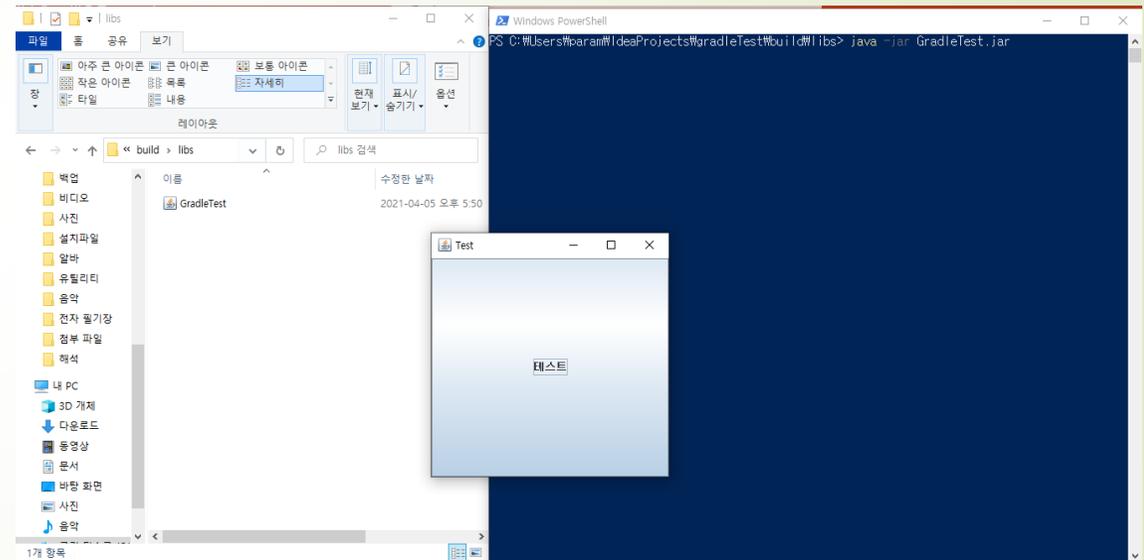
sourceCompatibility = 1.8
targetCompatibility = 1.8

dependencies {
    testImplementation 'org.junit.jupiter:junit-jupiter-api:5.6.0'
    testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine'
    // https://mvnrepository.com/artifact/org.openjfx/javafx-swing
    implementation group: 'org.openjfx', name: 'javafx-swing', version: '17-aa+6'
}

test {
    useJUnitPlatform()
}

compileJava.options.encoding = 'UTF-8'
tasks.withType(JavaCompile) {
    options.encoding = 'UTF-8'
}
```

테스트 코드



결과

Gradle을 통해 라이브러리 버전 정의 및 jar 생성 확인

Unit Testing – JUnit5



- ▶ 단위모듈(Method)이 정확히 구현되었는 지를 확인 Testcase 생성 및 실행, 오류추적
- ▶ 단위모듈별 테스트를 가능케 함으로써 코드품질을 보장
- ▶ 단위테스팅으로 통합 테스트 시의 회귀결함을 감소
- ▶ 다른 모듈에 의존하지 않고 원하는 모듈만 임의의 순서대로 테스트 가능
- ▶ JFeature(요구사항 개발도구)와 통합되어 요구사항의 정확한 구현 비율을 알 수 있음

JUnit5 Implement



```
@BeforeAll
public static void setClass(){
    junitTest = new JunitTest( name: "Team3", age: 24, tall: 180.0f);
}

@Test
void testGetName() {
    assertEquals( expected: "Team3", junitTest.getName());
}

@Test
void testGetAge() {
    assertEquals( expected: 24, junitTest.getAge());
}

@org.junit.jupiter.api.Test
void testGetTall() {
    assertEquals( expected: 180.0, junitTest.getTall(), delta: 0.0);
}
```

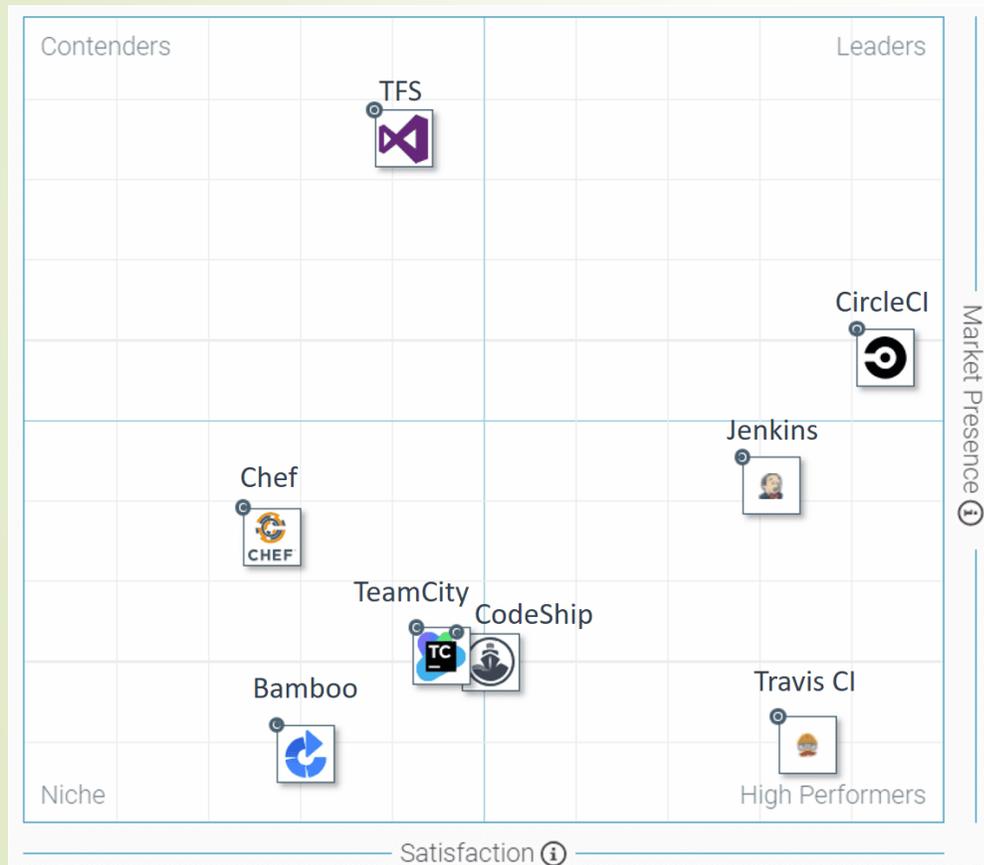
테스트 코드



✓ Test Results	19 ms
✓ JunitTestTest	19 ms
✓ printInfo()	16 ms
✓ testGetName()	2 ms
✓ testGetTall()	
✓ testGetAge()	
✓ testSetName()	
✓ testSetTall()	
✓ testSetAge()	1 ms

결과

CI Comparison



	Jenkins	TeamCity	Bamboo	Travis
Pricing	Free	\$299-\$1999	\$10-\$800	\$69-\$489
Operating system	Windows, Linux, macOS, any Unix-like OS	Windows, Linux, macOS, Solaris, FreeBSD, and more	Windows, Linux, macOS, Solaris	Linux, macOS
Hosting	On premise/ cloud	On premise	On premise/ Bitbucket as cloud	On premise/ cloud
Container support	✓	✓	✓	✓
Plugins	*****	****	**	****
Docs and support	Adequate	Good	Good	Poor
Learning curve and usability	Easy	Medium	Medium	Easy

Issue Tracking



vs



REDMINE

flexible project management

Communication

▶ Slack

- ▶ Github, Jenkins 연동 플러그인
 - Push/merge, build fail 알람 등 가능
- ▶ 디테일한 마크다운
 - 가독성/편리성



▶ Zoom

- ▶ 학교 이메일로 무료 이용 가능
- ▶ 팀원 간 실시간 화상채팅



CI Tool - Jenkins

- ▶ 무료이다.
- ▶ Docker와 Kubernetes와 같은 container사용에 용이
- ▶ Git, Gradle, Slack 등 library plugin이 많다.

(Cover : platforms, UI, administration, source code management, and, most frequently, build management.)

- ▶ 범용적으로 가장 많이 사용되고 있기 때문에, 정보 얻기가 상대적으로 쉬움.
- ▶ Master-Slave Architecture를 사용한다.

(Master -> main server , Slaves- remote machines used for build and test.)



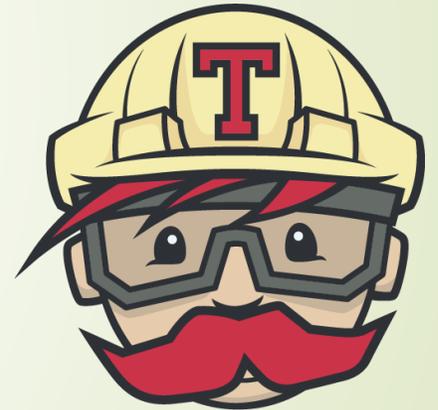
CI Tool - Team City

- ▶ JetBrains에 의해 개발된 CI
- ▶ 현존하는 거의 대다수의 OS를 지원 (Solaris 등등)
- ▶ History reports, instant feedback on test failure 등 고유의 편의기능이 다수 존재
- ▶ Code coverage analysis, 여러가지 .NET testing framework, static code analysis tool 등 많은 .NET tool들을 가장 잘 지원하고 있는 CI도구 중 하나
- ▶ Git 뿐만 아니라 정말 다양한 VCS를 따로 지원 (AccruRev, ClearCase, Mercurial, Perforce 등)



CI Tool - Travis

- ▶ 새로 빌드를 할 때 기존 Github repository를 clone함으로써 항상 backup을 유지할 수 있도록 제공
- ▶ 설치 불필요
- ▶ 간단하게 프로젝트를 등록하여 Yaml file로 관리 가능
 - > 전체적으로 사용하기가 매우 편리

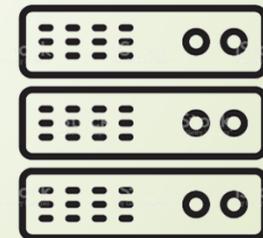
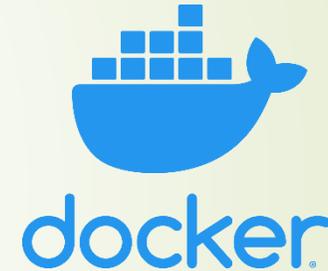


Server

- ▶ 팀원의 개인 Linux 서버 사용
- ▶ Docker를 통해 Jenkins와 Sonarqube 서비스
- ▶ 가정용 공유기의 포트포워딩으로 빌드용 서버의 포트들만 사용 가능



sonarqube



Server

Jenkins/Sonarqube Install on Docker

- ▶ Docker Hub에서 제공해주는 repository들을 활용
- ▶ Auto hacking bot들을 회피하기 위해 공인된 port number 외의 port number를 outbound로 사용

```
dtwxdtwx 1 root root 0 Apr 2 15:57 jenkins
(base) └─ dev sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED    STATUS    PORTS                                     NAMES
175396eb2b48   sonarqube     "bin/run.sh bin/sona..." 2 days ago Up 4 minutes 0.0.0.0:9111->9000/tcp   sonarqube
c0303a6ec1d7   jenkins/jenkins:lts "/sbin/tini -- /usr/..." 2 days ago Up 4 minutes 50000/tcp, 0.0.0.0:8181->8080/tcp   jenkins
(base) └─ dev
```

Docker Containers on Server

Jenkins Project Setting

- ▶ Jenkins에 Plugin 설치
- ▶ Github repository&credential
- ▶ Github web hook
 - ▶ 원하는 브랜치에 push가 발생하면 특정 커밋을 기준으로 자동 build
 - ▶ main 브랜치의 최신 커밋을 기준으로 자동 build 하도록 세팅
- ▶ Github Issues Plugin
 - ▶ build failure 내용에 대해 github issue에 자동으로 보고서 작성
- ▶ Slack notifications
 - ▶ slack에 원하는 정보를 포스트

Jenkins Project Setting – Plugins Install

- Github integration plugin
- Github issues plugin

<input checked="" type="checkbox"/>	GitHub Integration Plugin GitHub Integration Plugin for Jenkins	0.2.8	설치 제거
<input checked="" type="checkbox"/>	GitHub Issues Plugin This plugin creates GitHub issues when builds fail, and automatically closes the issue when the build starts passing again.	1.2.4	설치 제거

Plugins

Jenkins Project Setting – Repository&credential

- Github repository 접근 권한 부여
 - 접근 위해 github에서 access token 발행
- Repository url 입력
- Credentials에 Token 입력
- Build할 branch 설정 – main

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

소스 코드 관리

None

Git

Repositories

Repository URL

https://github.com/naem1023/software-v-v

Credentials

naem1023/*****

Add

고급...

Add Repository

Branches to build

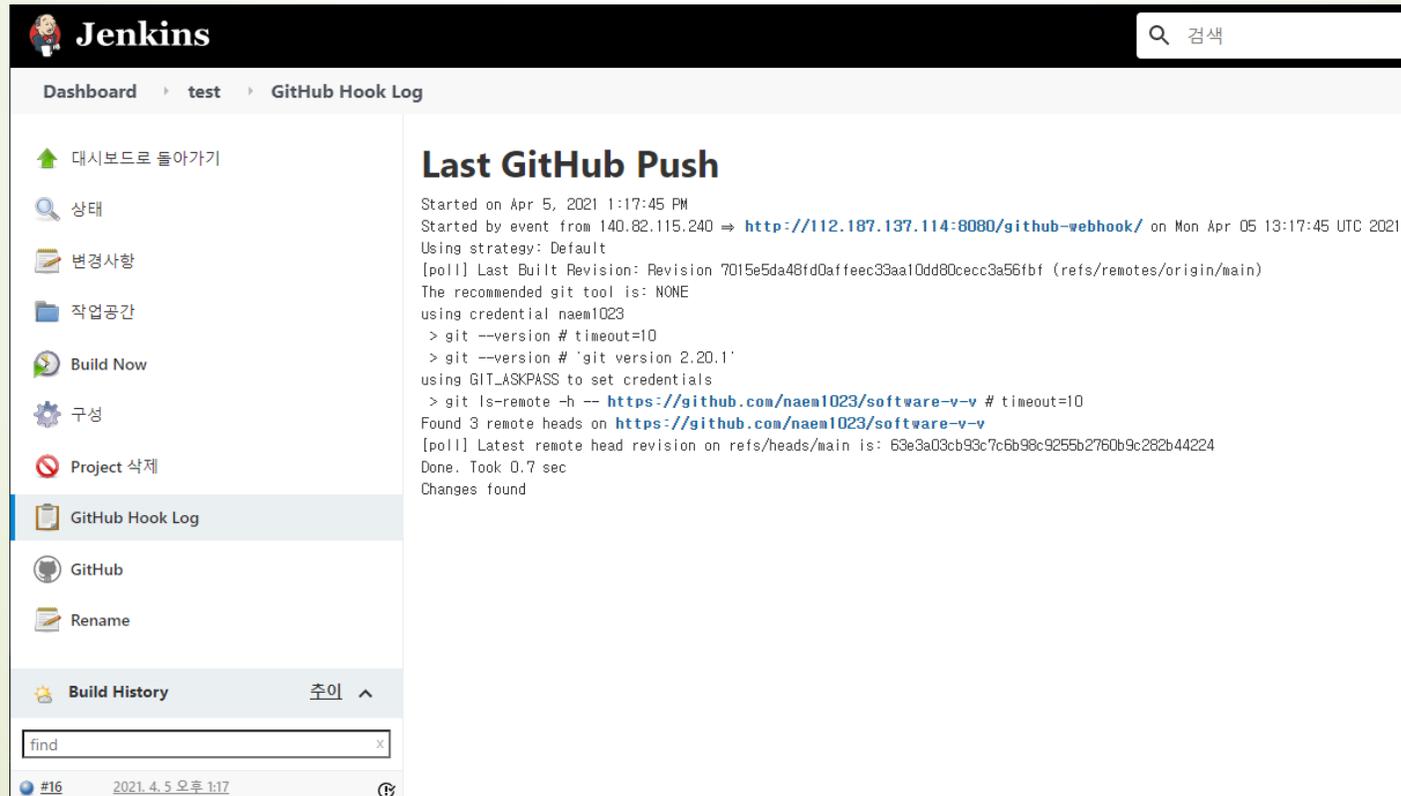
Branch Specifier (blank for 'any')

*/main

Add Branch

Jenkins Project Setting – Github Webhook & Jenkins

- ▶ 브랜치에 커밋 push 시에 github webhook가 성공적으로 작동하는 것을 확인



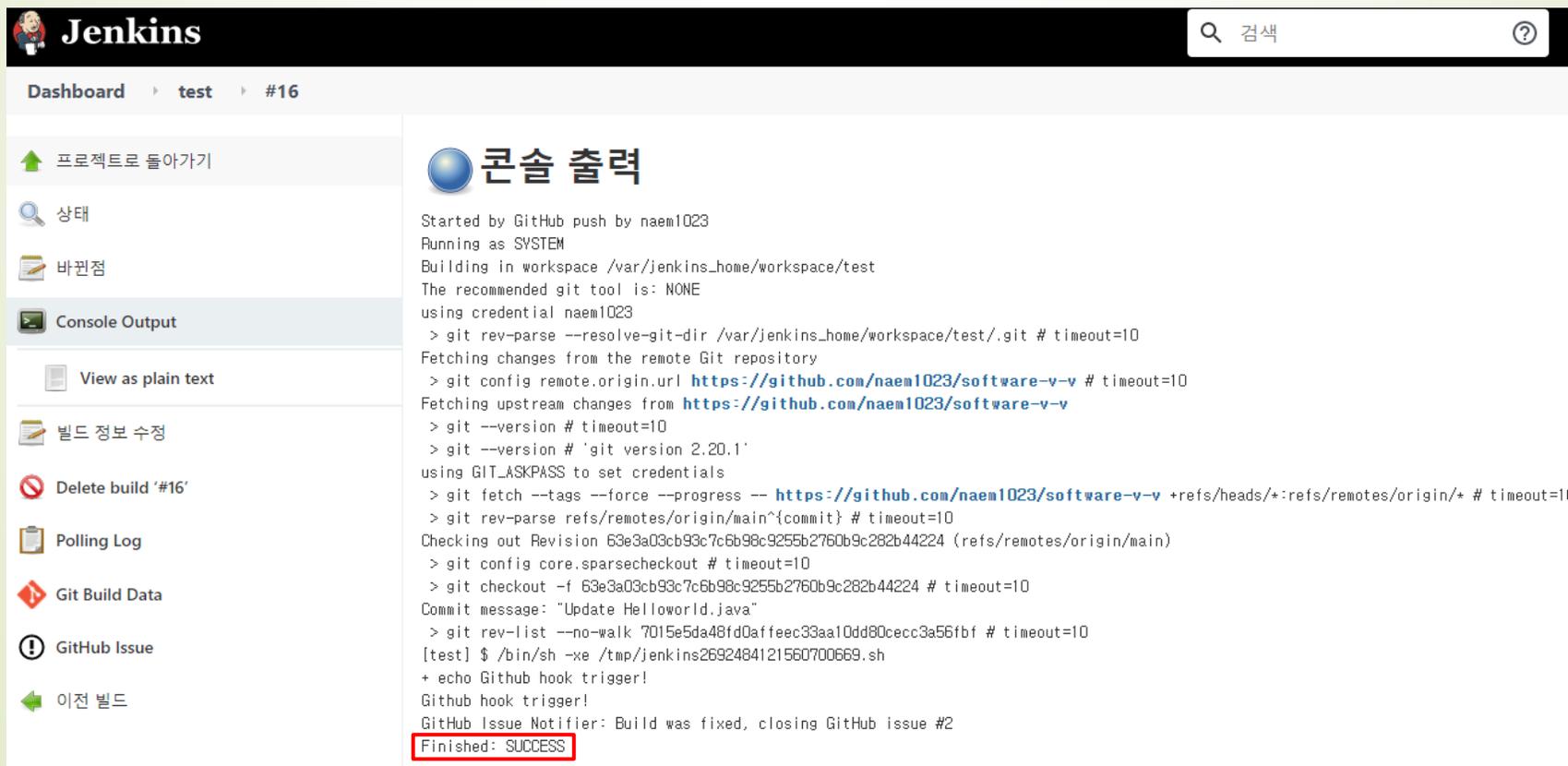
The screenshot displays the Jenkins web interface. At the top, the Jenkins logo and a search bar are visible. The breadcrumb navigation shows 'Dashboard > test > GitHub Hook Log'. On the left sidebar, there are several menu items: '대시보드로 돌아가기', '상태', '변경사항', '작업공간', 'Build Now', '구성', 'Project 삭제', 'GitHub Hook Log' (which is highlighted), 'GitHub', 'Rename', 'Build History', and a search box containing 'find'. The main content area is titled 'Last GitHub Push' and contains the following text:

```
Started on Apr 5, 2021 1:17:45 PM
Started by event from 140.82.115.240 => http://112.187.137.114:8080/github-webhook/ on Mon Apr 05 13:17:45 UTC 2021
Using strategy: Default
[pool] Last Built Revision: Revision 7015e5da48fd0affeec33aa10dd80cecc3a556fbf (refs/remotes/origin/main)
The recommended git tool is: NONE
using credential naem1023
> git --version # timeout=10
> git --version # 'git version 2.20.1'
using GIT_ASKPASS to set credentials
> git ls-remote -h -- https://github.com/naem1023/software-v-v # timeout=10
Found 3 remote heads on https://github.com/naem1023/software-v-v
[pool] Latest remote head revision on refs/heads/main is: 63e3a03cb93c7c6b98c9255b2760b9c282b44224
Done. Took 0.7 sec
Changes found
```

At the bottom of the page, the build number '#16' and the timestamp '2021. 4. 5 오후 1:17' are displayed.

Jenkins Project Setting – Github Webhook & Jenkins

- ▶ main 브랜치에 커밋 push시에 자동으로 build 되는 것을 확인



The screenshot shows the Jenkins web interface for a build named 'test #16'. The left sidebar contains navigation options like '프로젝트로 돌아가기', '상태', '바뀐점', 'Console Output', 'View as plain text', '빌드 정보 수정', 'Delete build '#16'', 'Polling Log', 'Git Build Data', 'GitHub Issue', and '이전 빌드'. The main area displays the console output for the build, which is titled '콘솔 출력'. The output shows the build was started by a GitHub push, and the console log includes the following commands and their outputs:

```
Started by GitHub push by naem1023
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/test
The recommended git tool is: NONE
using credential naem1023
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/naem1023/software-v-v # timeout=10
Fetching upstream changes from https://github.com/naem1023/software-v-v
> git --version # timeout=10
> git --version # 'git version 2.20.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/naem1023/software-v-v +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 63e3a03cb93c7c6b98c9255b2760b9c282b44224 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 63e3a03cb93c7c6b98c9255b2760b9c282b44224 # timeout=10
Commit message: "Update HelloWorld.java"
> git rev-list --no-walk 7015e5da48fd0affeec33aa10dd80cecc3a56fbf # timeout=10
[test] $ /bin/sh -xe /tmp/jenkins2692484121560700669.sh
+ echo Github hook trigger!
Github hook trigger!
GitHub Issue Notifier: Build was fixed, closing GitHub issue #2
Finished: SUCCESS
```

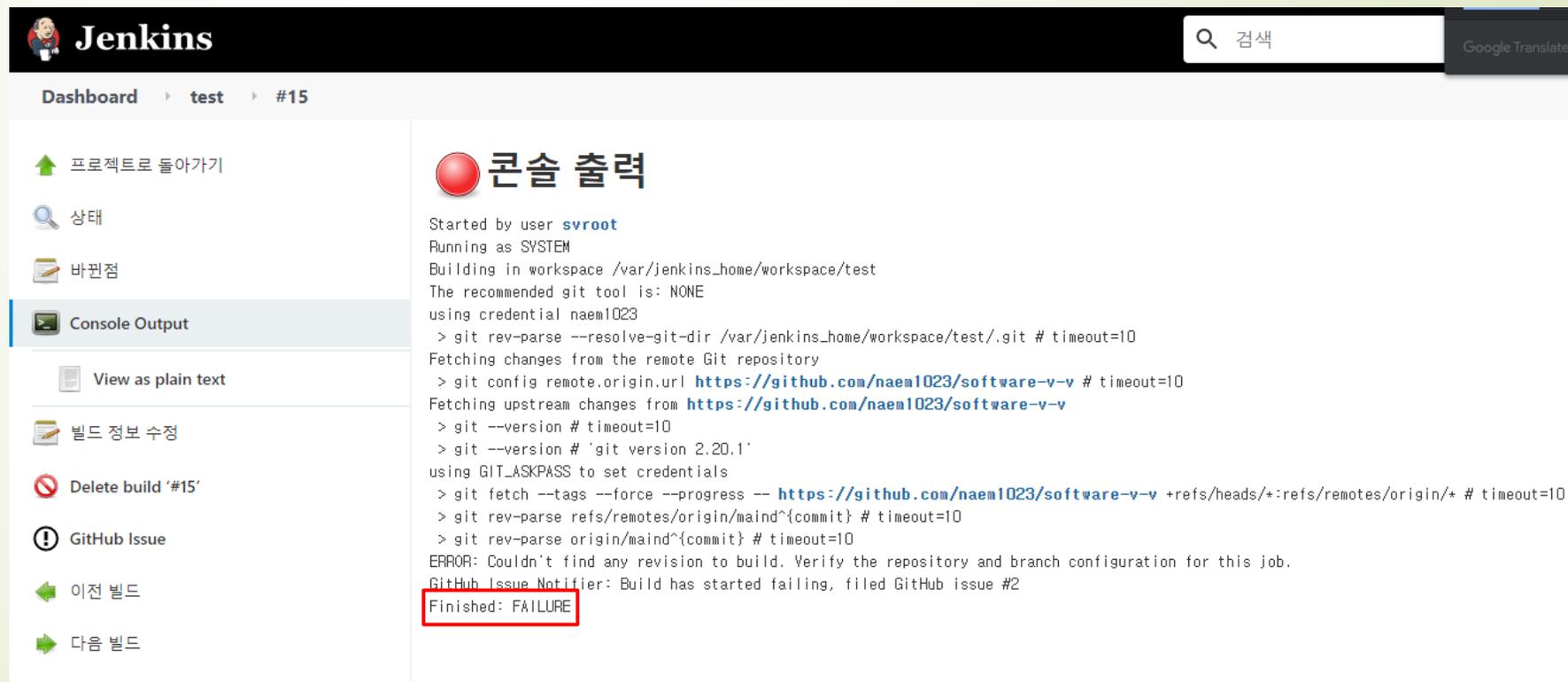
Jenkins Project Setting – Github Webhook & Jenkins - Test

- ▶ Main 브랜치에 커밋 push 시에 임의로 build 오류가 발생하도록 Jenkins 세팅을 변경
- ▶ Build failure 발생 확인
- ▶ Github Issue Plugin 작동 확인
- ▶ 성공적으로 build가 되도록 Jenkins 세팅 변경
- ▶ main 브랜치에 push
- ▶ Github Issue Plugin 작동 확인

Jenkins Project Setting – Github Webhook & Jenkins - Test

- ▶ Main 브랜치에 커밋 push 시에 임의로 build 오류가 발생하도록 Jenkins 세팅을 변경

- ▶ Build failure 발생 확인



The screenshot shows the Jenkins web interface for a job named 'test' (build #15). The 'Console Output' tab is selected, displaying the following text:

```
Started by user svroot
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/test
The recommended git tool is: NONE
using credential naem1023
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/naem1023/software-v-v # timeout=10
Fetching upstream changes from https://github.com/naem1023/software-v-v
> git --version # timeout=10
> git --version # 'git version 2.20.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/naem1023/software-v-v +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
> git rev-parse origin/main^{commit} # timeout=10
ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job.
GitHub Issue Notifier: Build has started failing, filed GitHub issue #2
Finished: FAILURE
```

The final line, 'Finished: FAILURE', is highlighted with a red box. The left sidebar contains navigation options such as '프로젝트로 돌아가기', '상태', '바뀐점', 'Console Output', 'View as plain text', '빌드 정보 수정', 'Delete build '#15'', 'GitHub Issue', '이전 빌드', and '다음 빌드'.

Jenkins Project Setting – Github Webhook & Jenkins - Test

Issue Report

- ▶ Jenkins의 Github Issues Plugin이 failure issue 작성
- ▶ 성공적으로 build가 되도록 Jenkins 세팅 변경
- ▶ main 브랜치에 push
- ▶ Build가 성공하고 Github issues plugin이 failure issue에 "Build was fixed!"를 출력 후 issue를 close하는 것을 확인

The screenshot shows a GitHub issue titled "test #15 failed #2" which has been closed. The issue was opened by user "naem1023" 5 hours ago. A comment from "naem1023" 5 hours ago states "Build 'test' is failing!" and includes the last 50 lines of build output. The output shows a Jenkins build process that failed with the error: "ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job." A subsequent comment from "naem1023" 8 minutes ago states "Build was fixed!", which is highlighted with a red box in the image.

```
test #15 failed #2
Closed naem1023 opened this issue 5 hours ago · 1 comment

naem1023 commented 5 hours ago
Build 'test' is failing!

Last 50 lines of build output:

Started by user svroot
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/test
The recommended git tool is: NONE
using credential naem1023
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/naem1023/software-v-v # timeout=10
Fetching upstream changes from https://github.com/naem1023/software-v-v
> git --version # timeout=10
> git --version # 'git version 2.28.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/naem1023/software-v-v +refs/heads/*:refs/remotes/origin/* # t
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
> git rev-parse origin/main^{commit} # timeout=10
ERROR: Couldn't find any revision to build. Verify the repository and branch configuration for this job.

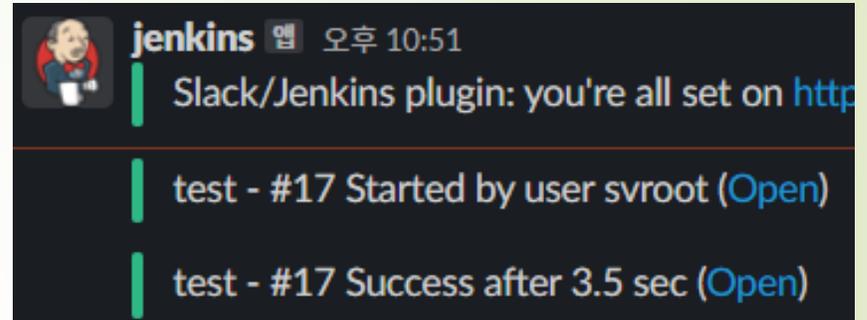
Changes since last successful build:
No changes

View full output

naem1023 commented 8 minutes ago
Build was fixed!
```

Slack Notification - Jenkins

- ▶ Jenkins에서 빌드 발생 시
- > Project Setting에 설정한 정보들이 slack에 자동으로 포스트



빌드 후 조치

Create GitHub issue on failure

Advanced settings

Slack Notifications

- Notify Build Start
- Notify Success
- Notify Aborted
- Notify Not Built
- Notify Unstable
- Notify Regression
- Notify Every Failure
- Notify Back To Normal

Dashboard > test > #17

- 프로젝트로 돌아가기
- 상태
- 바뀐점
- Console Output
- View as plain text
- 빌드 정보 수정
- Delete build '#17'
- Git Build Data
- 이전 빌드

콘솔 출력

Started by user **svroot**
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/test
The recommended git tool is: NONE
using credential naem1023

```
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/naem1023/software-v-v # timeout=10
Fetching upstream changes from https://github.com/naem1023/software-v-v
> git --version # timeout=10
> git --version # 'git version 2.20.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/naem1023/software-v-v +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 63e3a03cb93c7c6b98c9255b2760b9c282b44224 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 63e3a03cb93c7c6b98c9255b2760b9c282b44224 # timeout=10
Commit message: "Update HelloWorld.java"
> git rev-list --no-walk 63e3a03cb93c7c6b98c9255b2760b9c282b44224 # timeout=10
[test] $ /bin/sh -xe /tmp/jenkins7654641660870042025.sh
+ echo Github hook trigger!
Github hook trigger!
[Slack Notifications] found #16 as previous completed, non-aborted build
[Slack Notifications] will send OnSuccessNotification because build matches and user preferences allow it
Finished: SUCCESS
```

Slack Notification - Github

- ▶ 새 push , merge 등 발생 시 포스트
- ▶ Issue 포스트

Friday, April 2nd

GitHub APP 4:00 PM
was added to this conversation by Hyunwoo Park.

GitHub APP 4:32 PM
Subscribed to [naem1023/software-v-v](#)

Pull request opened by fous0327

fous0327
#1 Update README.md
slack test
naem1023/software-v-v | Apr 2nd

GitHub APP 5:27 PM
naem1023
2 new commits pushed to `main`
`379099b9` - Update README.md
`0292dac9` - Merge pull request #1 from naem1023/fous0327-test
naem1023/software-v-v

Pull request merged by naem1023

#1 Update README.md
naem1023/software-v-v | Apr 2nd

github-pages[bot]

Type @ to notify a teammate. Try it

Overview

