

# CI/CD 구축 & GHE 연동



Tangunsoft Co., Ltd.

Eric.Kim

Tel : 02-6206-2545

E-Mail : kch@tangunsoft.com

# Agenda

- CI/CD와 Jenkins
  - Build&Deploy Architecture
  - Jenkins 설치
  - Jenkins 환경설정
  - 배포서버 구성
  - 송출서버 구성
  - Jenkins 프로젝트 구성
  - CI/CD에서 DevOps로

# CI/CD와 Jenkins

# CI/CD란 무엇인가

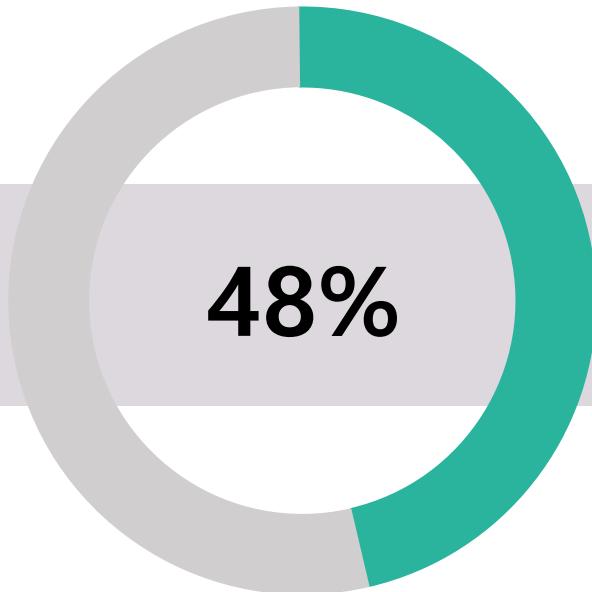
**Continuous Integration / Continuous Delivery & Deployment**

애플리케이션 개발 단계를 자동화하여 애플리케이션을 보다 짧은 주기로 고객에게 제공하는 방법



## CI/CD 구축의 필요성

## 개발자들이 업무시간 중 코딩에 투자하는 시간은..?

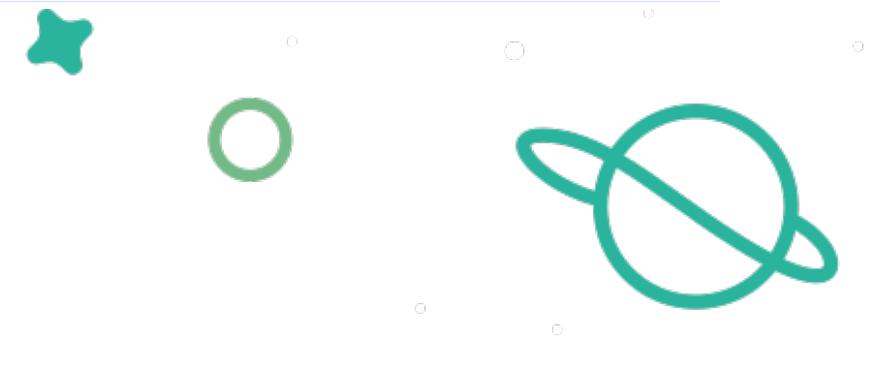


# 업무 시간 중 약 48% 에만 코딩

그럼 대체, 어디에 시간을 투자하고 있을까?

- 툴 세팅
  - 미팅
  - 인프라 문제
  - 관리자 승인 대기

## 여러분의 하루 일과는 어떤가요?



# Jenkins란?

# Jenkins

- OpenSource CI Tool
  - 2004년부터 Sun Microsystems의 가와구치 코스케에 의해 개발
  - Sun Microsystems가 Oracle에 인수된 후 라이선스 이슈로

Jenkins라는 별명의 프로젝트로 분기됨

- 약 1700개(2019년 11월 기준) 의 다양한 Plug-In을 기반으로

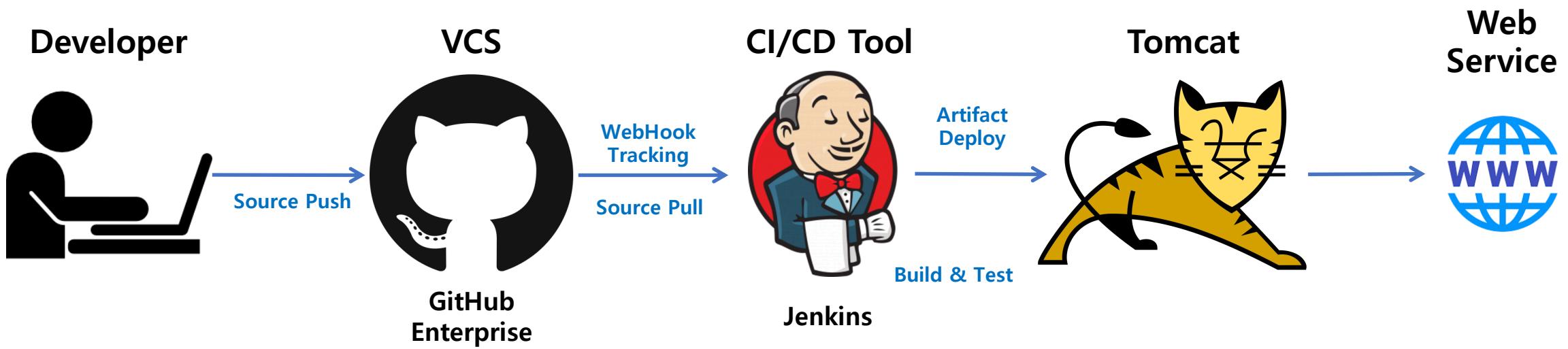
사용자가 원하는 형태로 CI서버 구성 가능



# Jenkins

# Build&Deploy Architecture

# Build & Deploy Automation Architecture

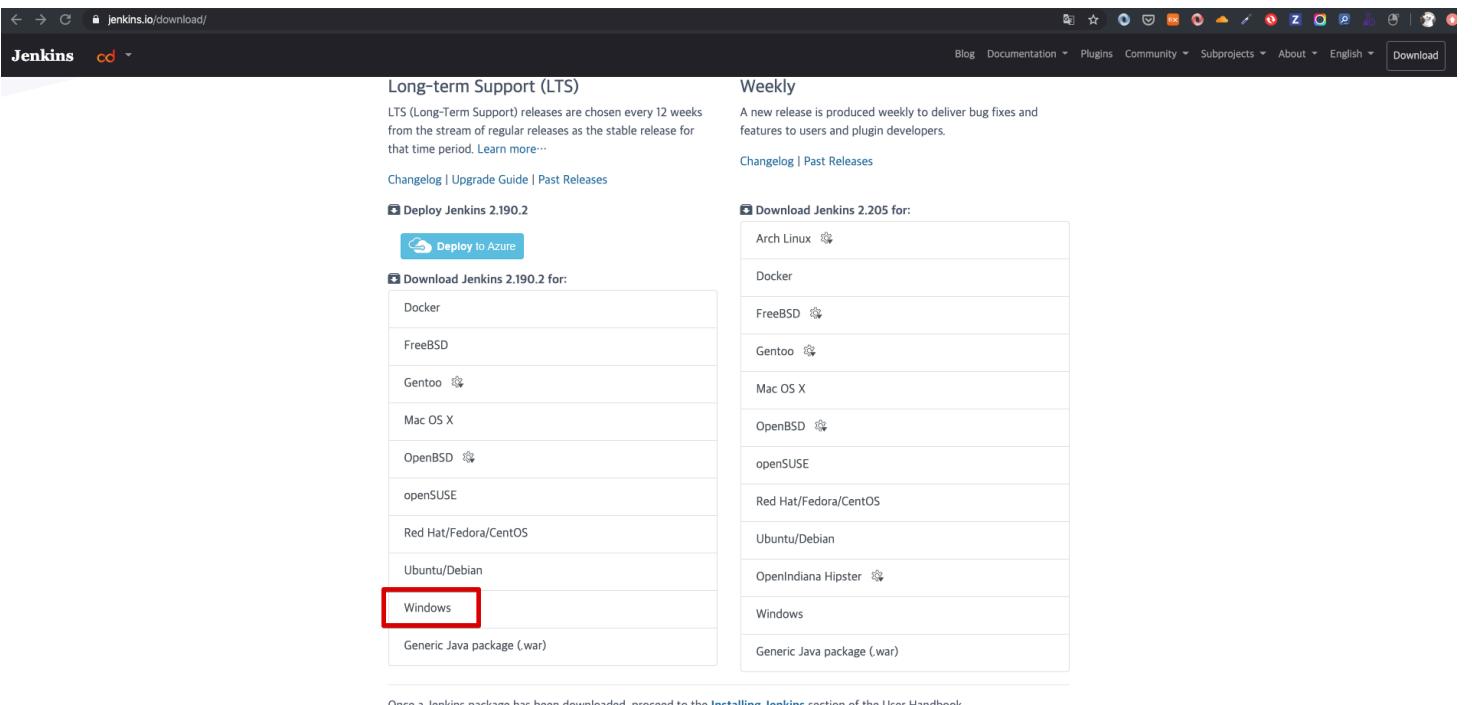


# Jenkins 설치

# Jenkins 설치

# Jenkins Download

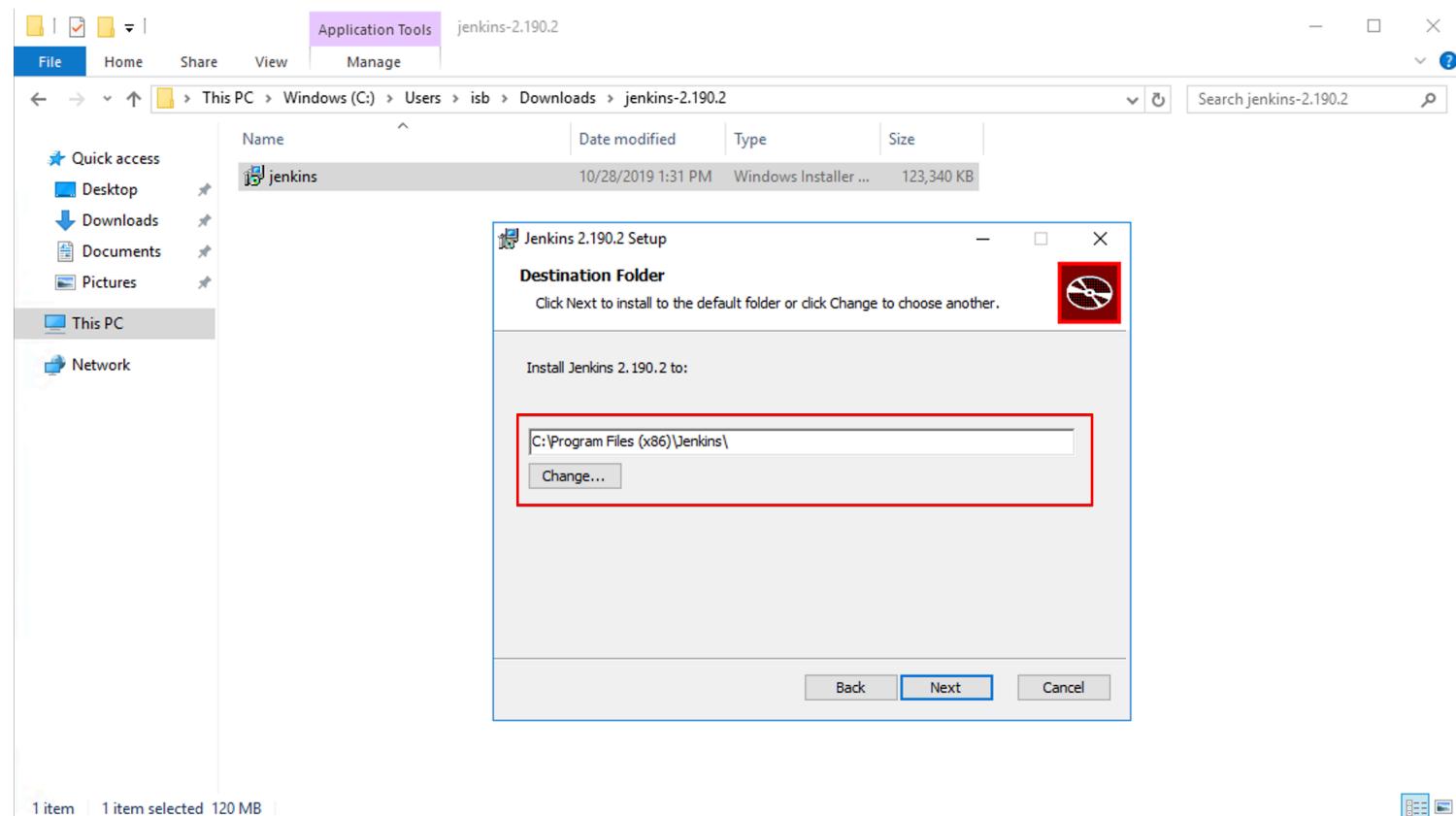
- <https://jenkins.io/download/>
  - LTS Version의 Windows OS Installer File Download



# Jenkins 설치

# Jenkins Install

- Download 받은 파일의 압축을 풀고 설치 파일 실행
  - 설치 경로 설정하고 Install 진행



# Jenkins 설치

## Jenkins 초기 설정

- Jenkins 설치경로 내 initialAdminPassword 파일을 열어 Password Text를 복사.
  - Administrator password란에 복사한 Password를 삽입 후 Continue 버튼 클릭

## Getting Started

# Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

1

C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword

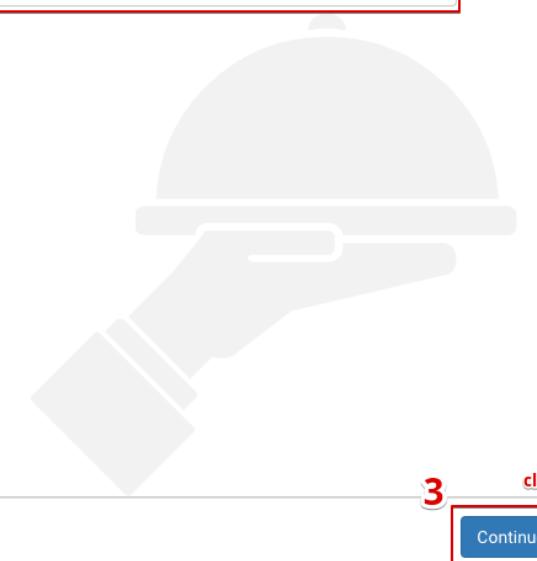
## File Open & Copy

Please copy the password from either location and paste it below.

2

## Administrator password

Paste



# Jenkins 설치

## Jenkins 초기 설정

- Install Suggested Plugin – Most Plugin Default 설치
  - Select Plugins to Install - Most Plugin Select 설치
  - Install Suggested Plugin 클릭

Getting Started

X

# Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

click

## Install suggested plugins

Install plugins the Jenkins community finds most useful.

## Select plugins to install

Select and install plugins most suitable for your needs.

A large, semi-transparent grayscale icon of a hand holding a wrench, positioned in the lower right quadrant of the page. The icon is slightly faded, allowing the background content to be visible through it.

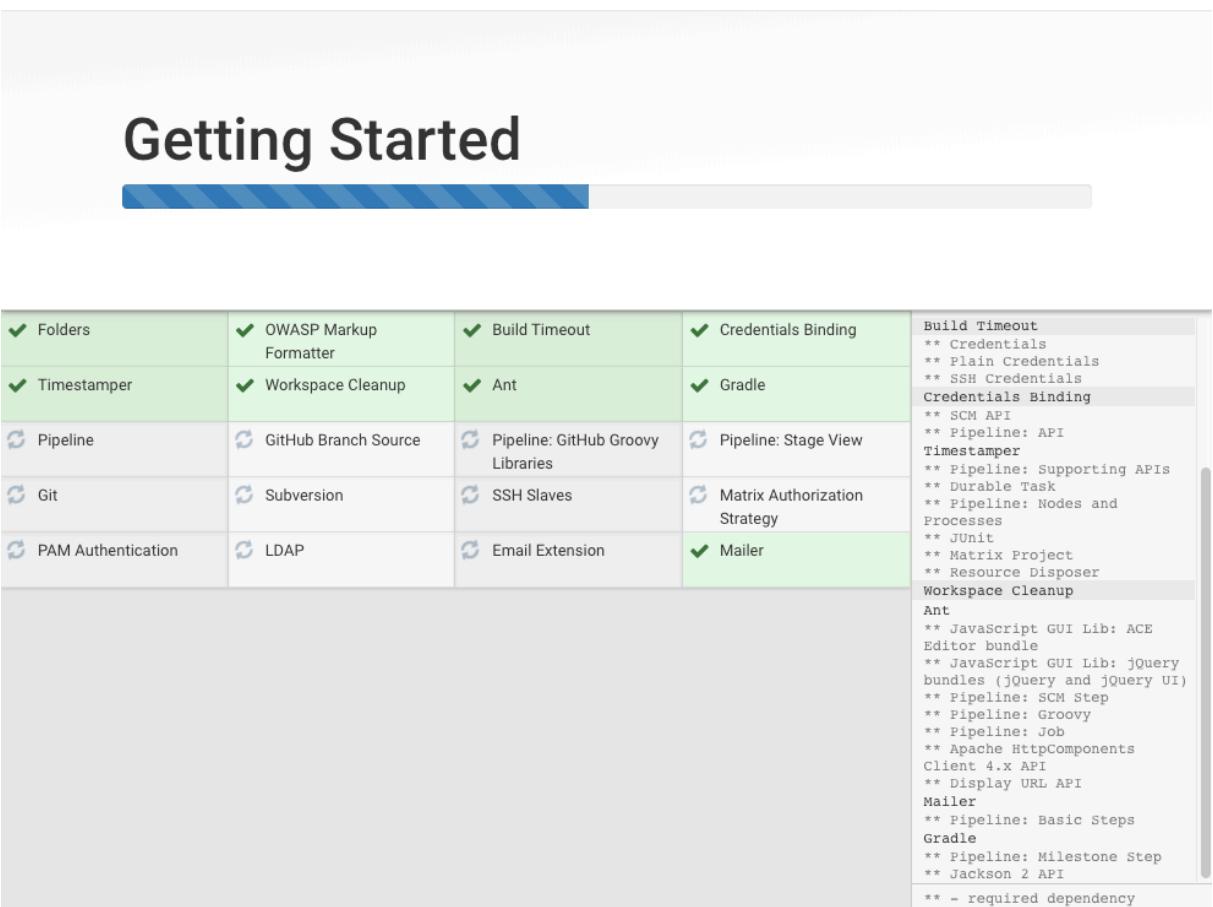
Jenkins 2.190.2

# Jenkins 설치

## Jenkins 초기 설정

- Most Plugin 설치 진행

### Getting Started



Jenkins 2.190.2

# Jenkins 설치

## Jenkins 초기 설정

- Jenkins 관리자 계정 생성
  - 관리자 계정 정보 입력 후 Save and Continue 버튼 클릭

Getting Started

# Create First Admin User

계정명:	<input type="text"/>
암호:	<input type="password"/>
암호 확인:	<input type="password"/>
이름:	<input type="text"/>
이메일 주소:	<input type="text"/>

Jenkins 2.190.2

[Continue as admin](#)

[Save and Continue](#)

# Jenkins 설치

## Jenkins 초기 설정

- Jenkins Service URL 확인 후 Save and Finish 버튼 클릭

Getting Started

# Instance Configuration

Jenkins URL:

http://:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.190.2

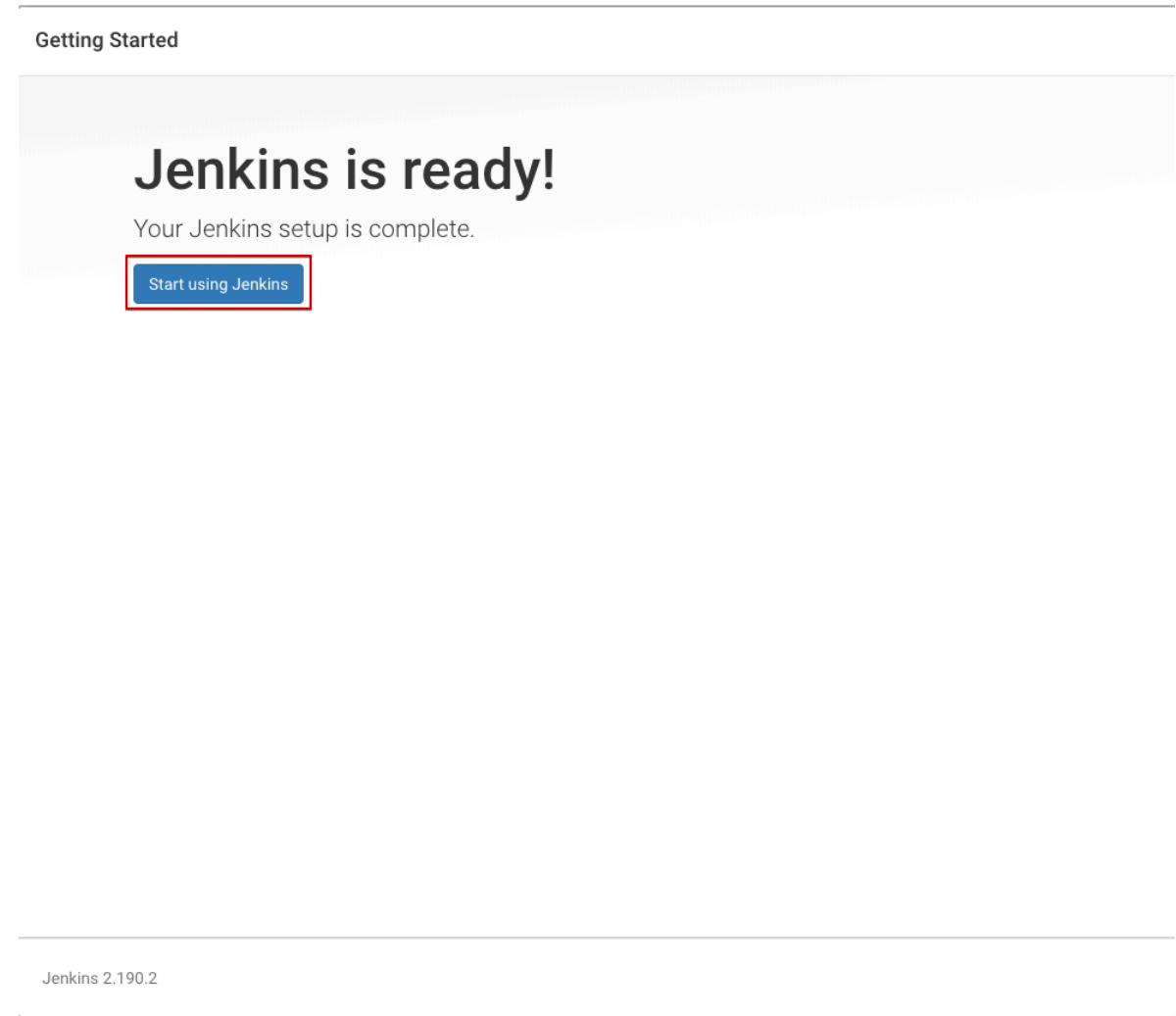
Not now

Save and Finish

# Jenkins 설치

## Jenkins 초기 설정

- 초기 설정 완료. Start Using Jenkins 버튼 클릭



# Jenkins 설치

## Jenkins 초기 설정

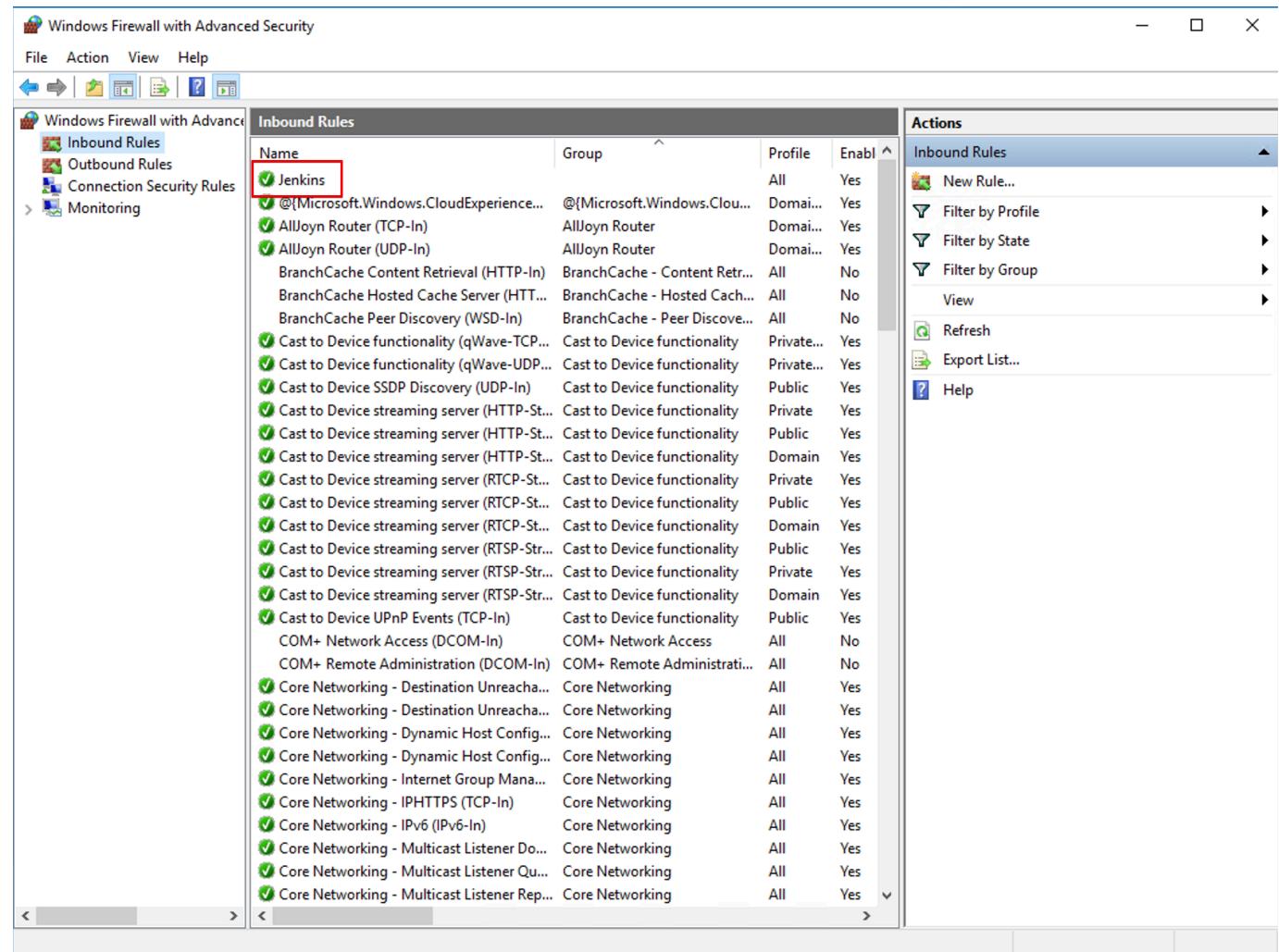
- Jenkins 초기 화면

The screenshot shows the Jenkins dashboard. At the top left is the Jenkins logo and the word "Jenkins". On the right are search, help, and user account links. The main content area has a large green header "Jenkins에 오신 것을 환영합니다." (Welcome to Jenkins). Below it is a teal box with the text "시작하려면 새 작업을 만들어 주시기 바랍니다." (Please create a new job). To the left is a sidebar with icons for "새로운 Item", "사람", "빌드 기록", "Jenkins 관리", "My Views", "Credentials", "Lockable Resources", and "New View". Below the sidebar are two collapsed sections: "빌드 대기 목록" (Build Queue List) which says "빌드 대기 항목이 없습니다." (No build items in queue), and "빌드 실행 상태" (Build Execution Status) which shows "1 대기 중" (1 pending) and "2 대기 중" (2 pending).

# Jenkins 설치

## Jenkins 초기 설정

- Server Local의 시스템 검색에서 "Windows Firewall with Advanced Security" 검색
- Inbound Rules에서 Jenkins 등록 확인



# Jenkins 환경 설정

## Jenkins 환경설정

JDK 설치 확인

- 검색 -> cmd
  - 명령 프롬프트 창에서 java -version

```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.  
  
C:\Users\isb>java -version  
'java' is not recognized as an internal or external command,  
operable program or batch file.  
  
C:\Users\isb>
```

# Jenkins 환경설정

## JDK 설치

- <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Java SE 11.0.5 (LTS) Download
- Download 시 Oracle 계정 필요

The screenshot shows the Oracle Java SE Downloads page. It features three main download sections:

- Java SE 13.0.1**: Described as the latest release for the Java SE Platform. It includes links for Installation Instructions, Release Notes, Oracle JDK License, Java SE Licensing Information User Manual (which includes Third Party Licenses), Certified System Configurations, and Readme. A blue "Oracle JDK DOWNLOAD" button is located on the right.
- Java SE 11.0.5 (LTS)**: Described as the latest release for the Java SE 11 Platform. It includes links for Installation Instructions, Release Notes, Oracle JDK License, Java SE Licensing Information User Manual (which includes Third Party Licenses), Certified System Configurations, and Readme. A blue "Oracle JDK DOWNLOAD" button is located on the right. The "Java SE 11.0.5 (LTS)" section is highlighted with a red border and has a "Click" label above it.
- Java SE 8u231**: Described as including important bug fixes. It includes links for Installation Instructions, Release Notes, Oracle License, Java SE Licensing Information User Manual (which includes Third Party Licenses), and Certified System Configurations. A blue "JDK DOWNLOAD" button is located on the right.

Below the sections, there is a note: "To Learn more about these options visit Oracle JDK Releases for Java 11 and Later".

# Jenkins 환경설정

## JDK 설치

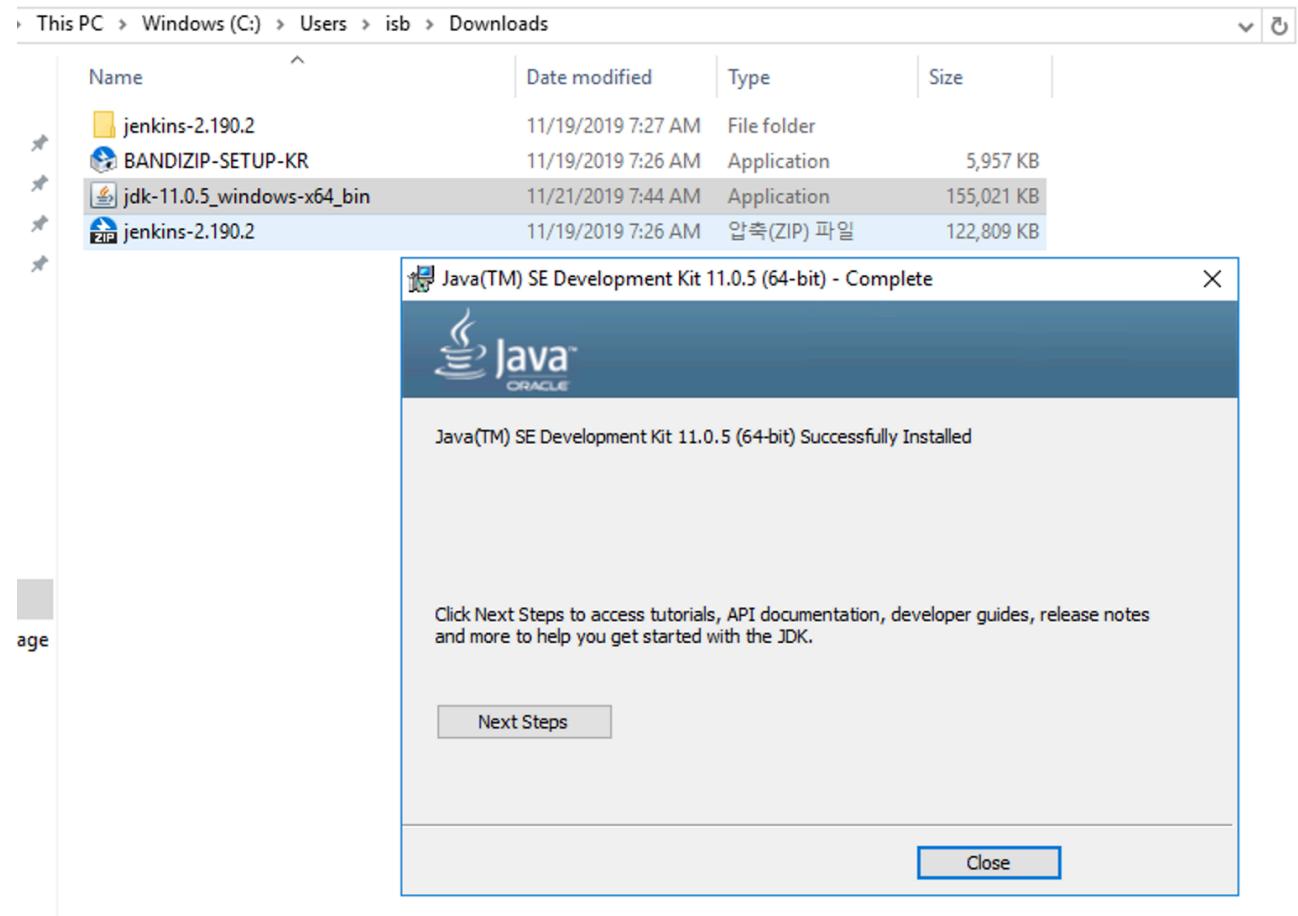
- download this software. Accept License Agreement  
라디오 버튼 항목 선택
- jdk-11.0.5\_windows-x64\_bin.exe 클릭하여 설치 파일  
Download



## Jenkins 환경설정

## JDK 설치

- JDK 설치 파일 실행
  - Default 설정으로 설치 진행

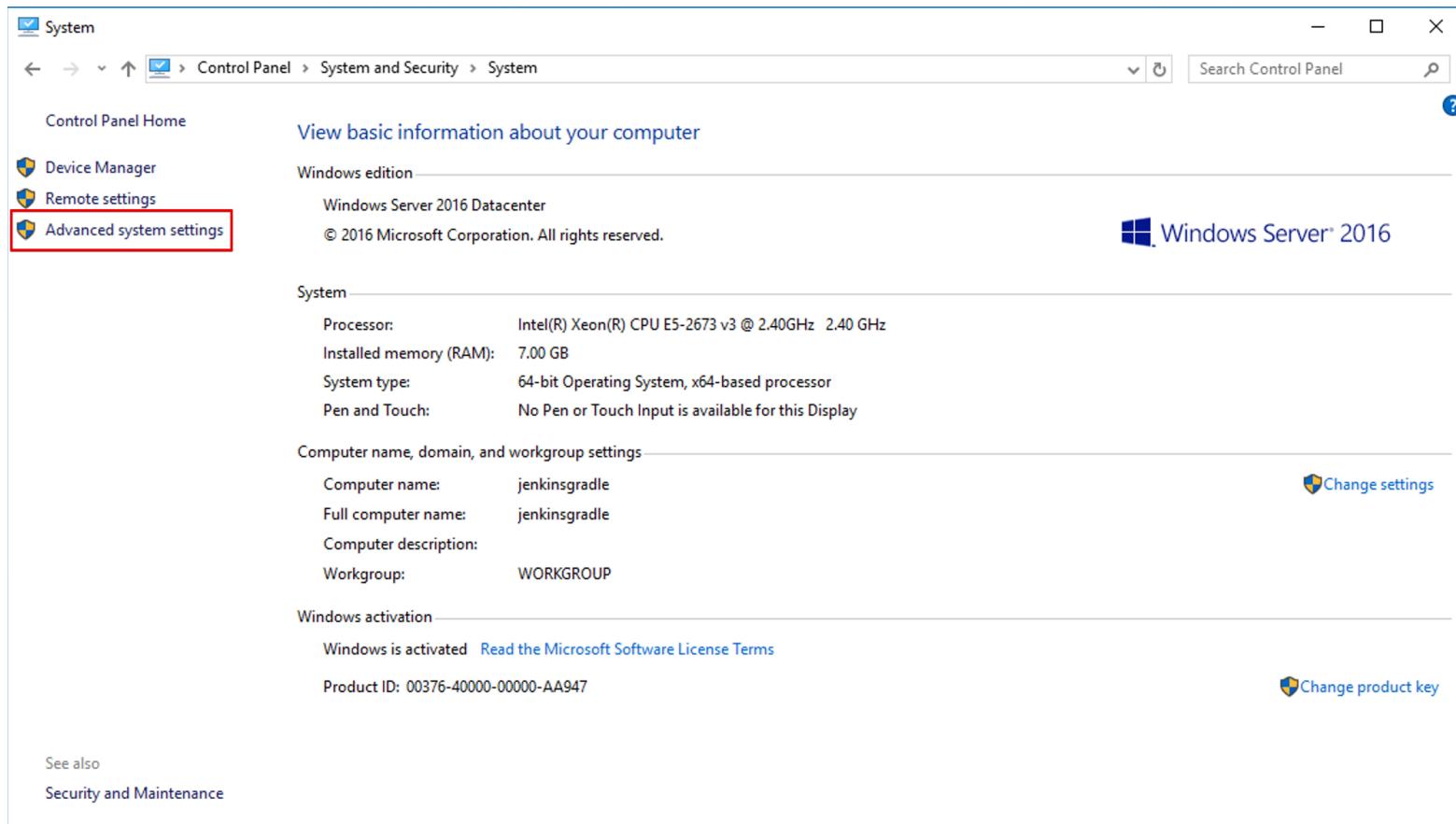


# Jenkins 환경설정

## JDK 설치

- 제어판 -> 시스템 및 보안 -> 시스템 ->

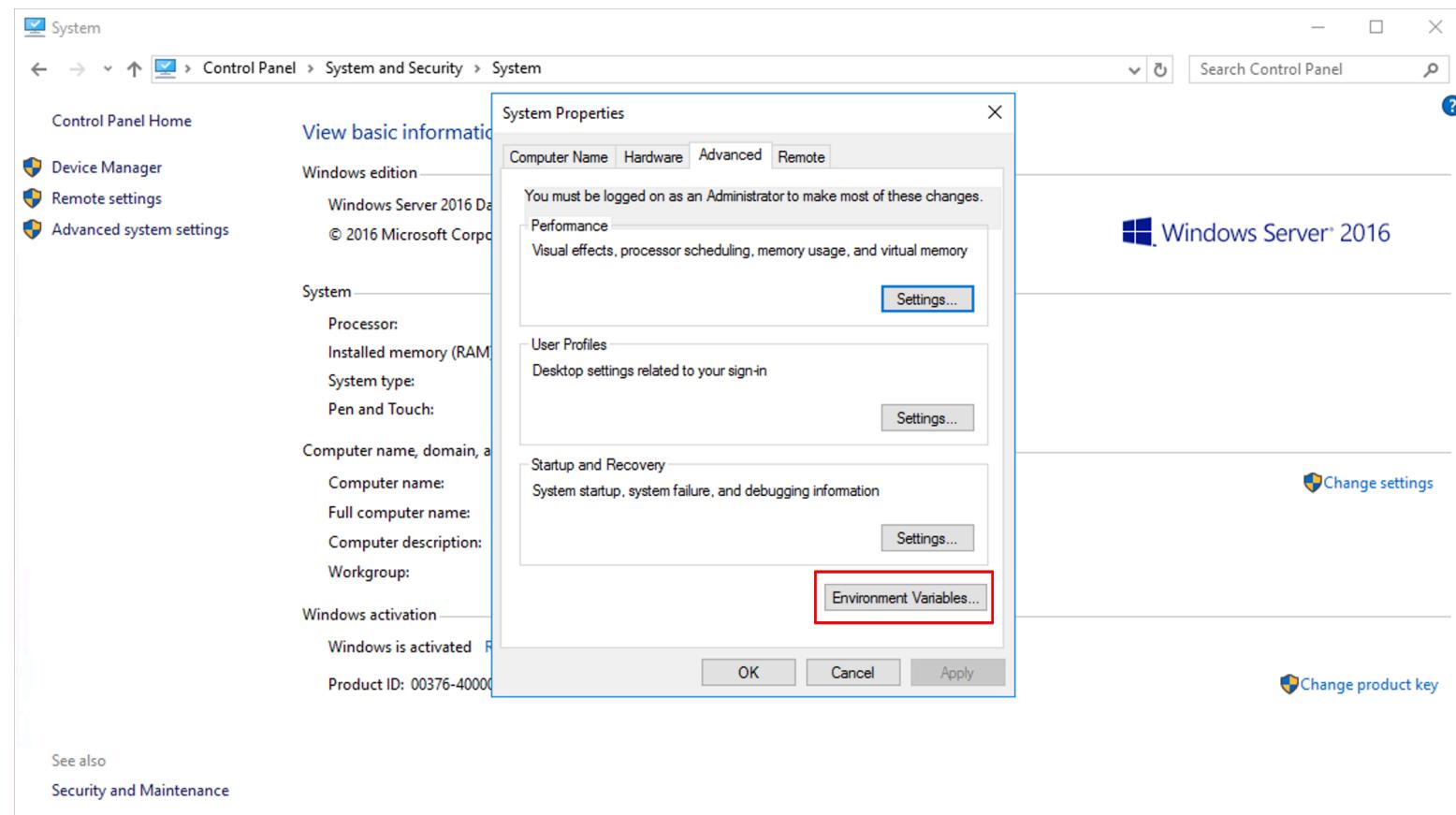
## 고급 시스템 설정 클릭



## Jenkins 환경설정

## JDK 설치

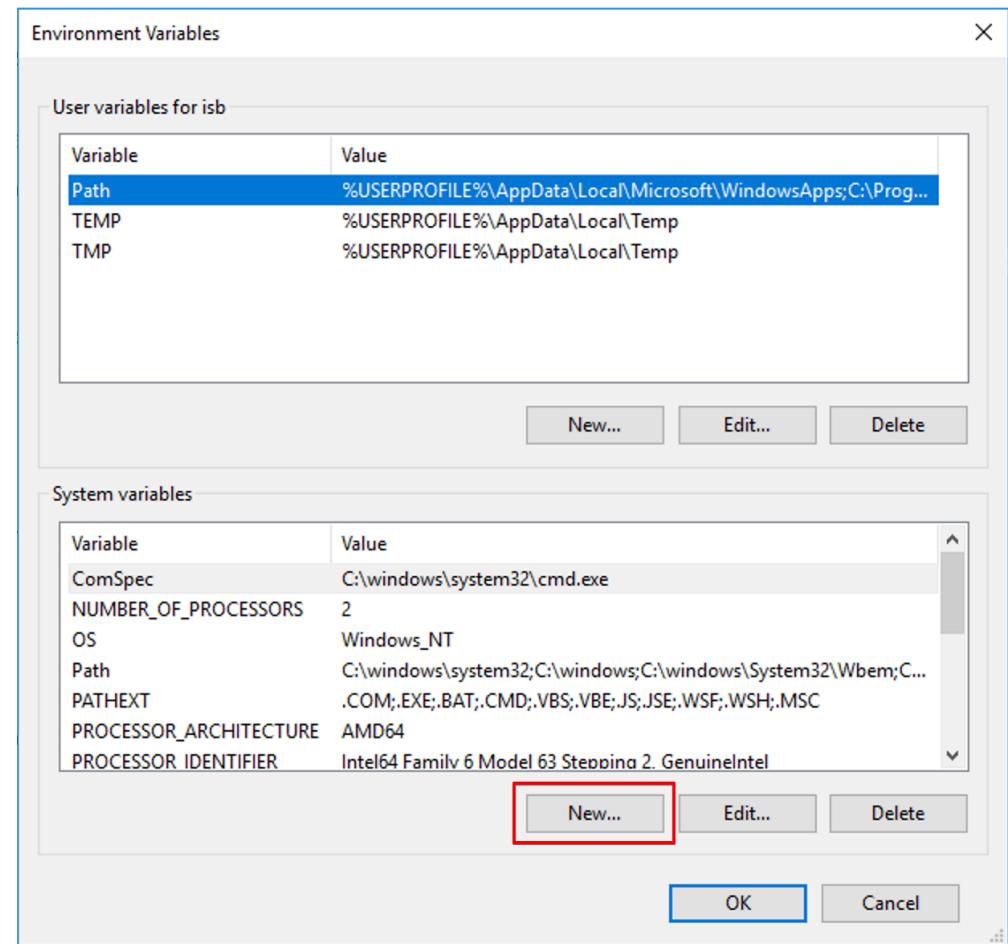
- 제어판 -> 시스템 및 보안 -> 시스템 ->  
고급 시스템 설정 -> 고급 -> 환경변수



## Jenkins 환경설정

## JDK 설치

- 환경변수 창에서 시스템 변수의 "새로 만들기" 버튼 클릭



# Jenkins 환경설정

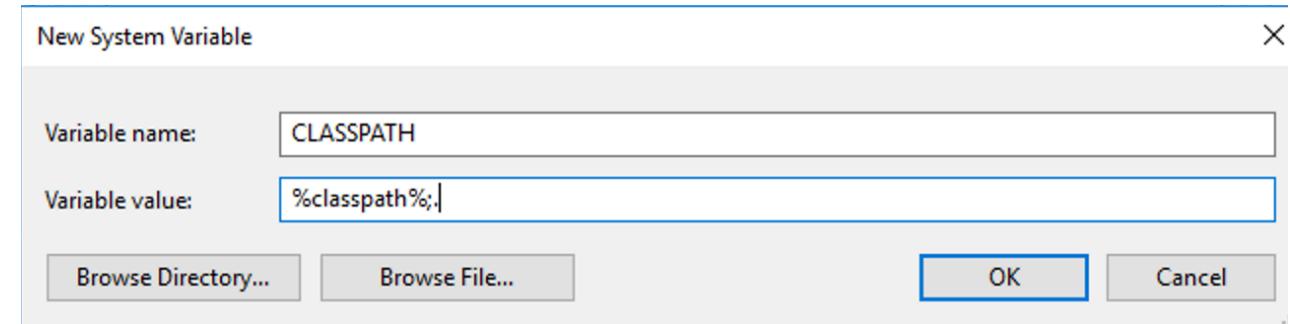
## JDK 설치

- 시스템 변수 추가

Virable1

Variable name : CLASSPATH

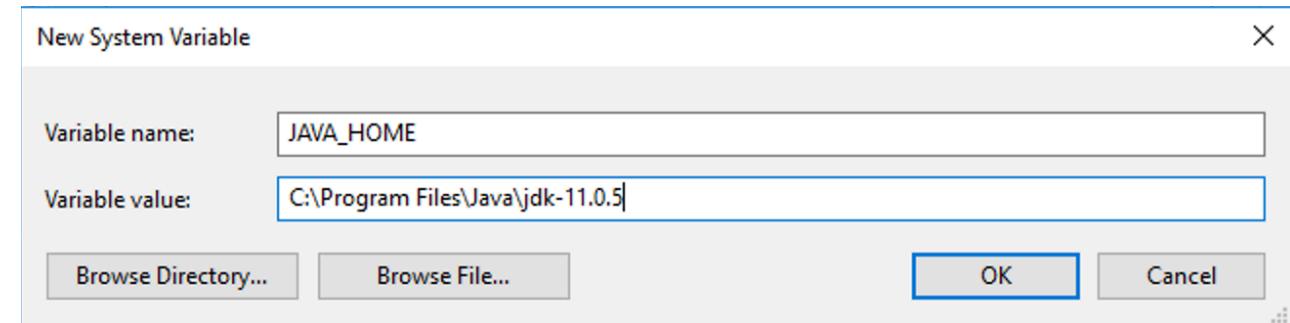
Variable value : %classpath%;;.



Virable1

Variable name : JAVA\_HOME

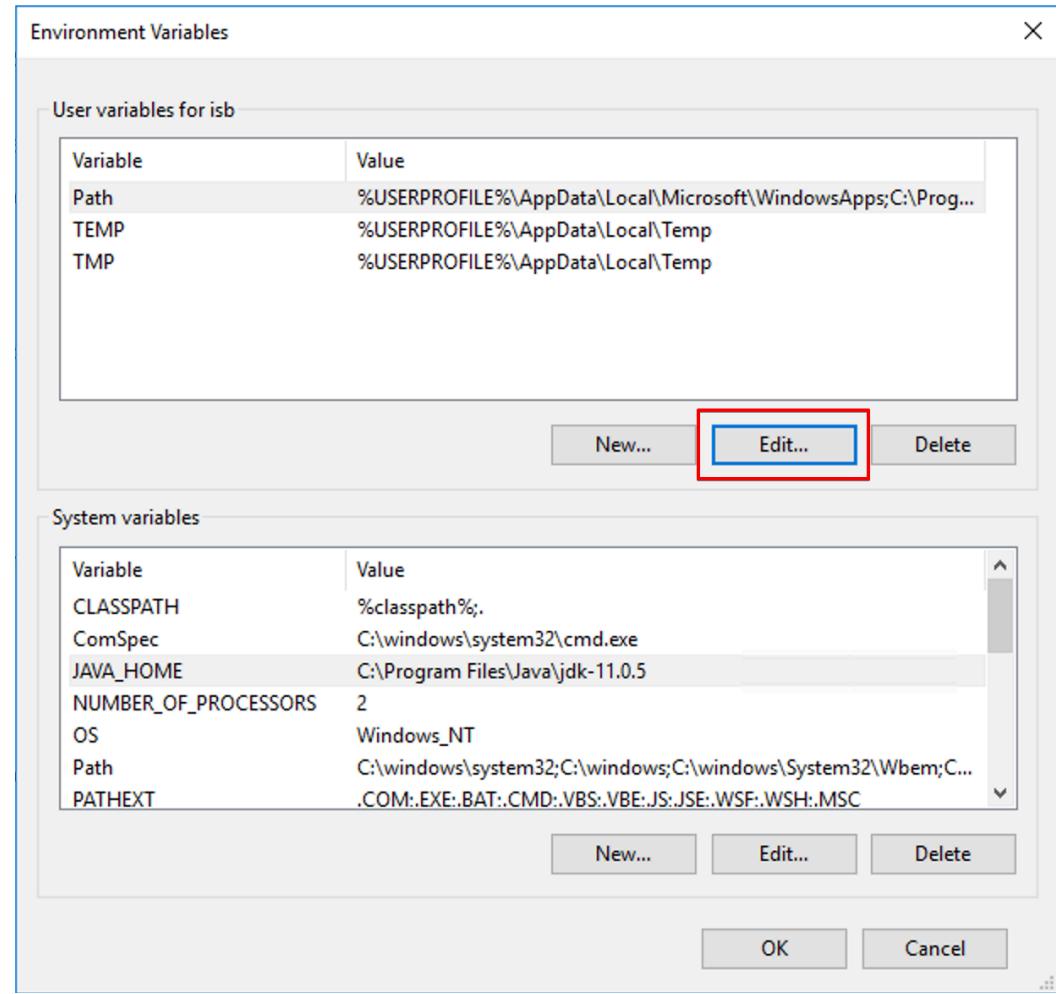
Variable value : C:\Program File\Java\Java Version



## Jenkins 환경설정

## JDK 설치

- 유저 변수의 Path 선택 후 편집 버튼 클릭

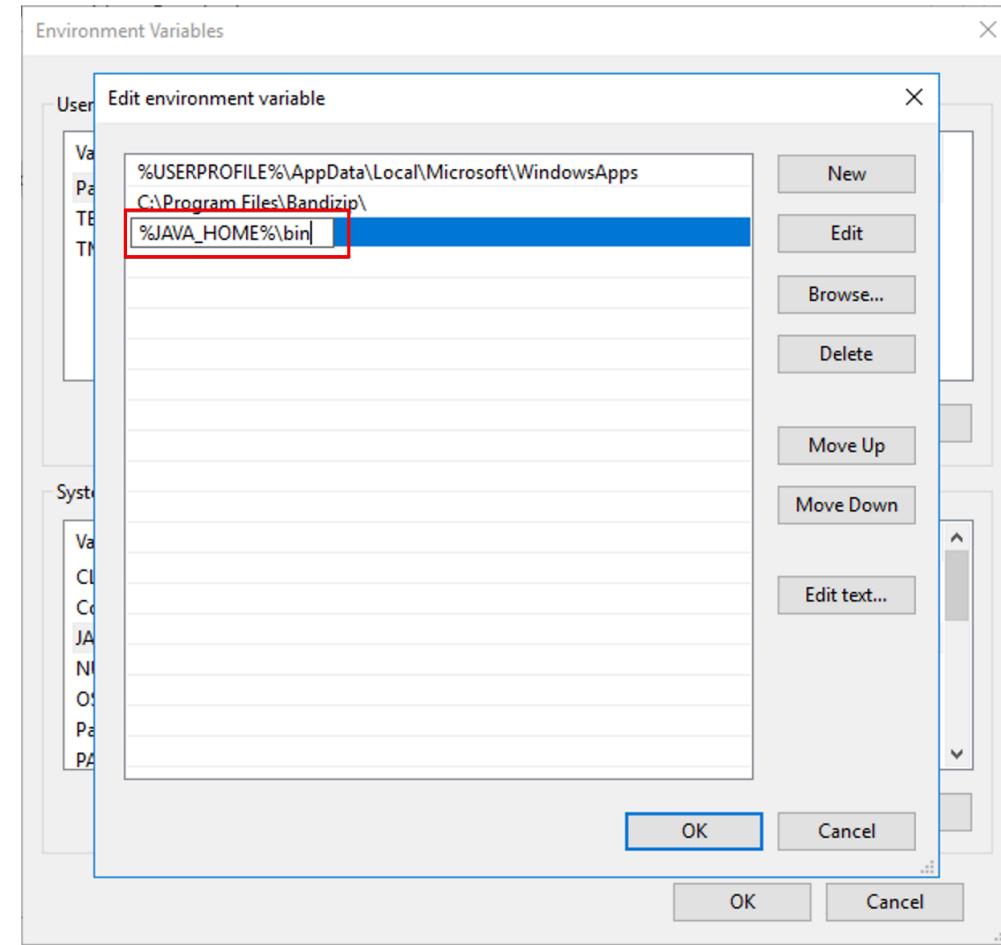


## Jenkins 환경설정

## JDK 설치

- “새로 만들기” 버튼 클릭 후 변수값 저장

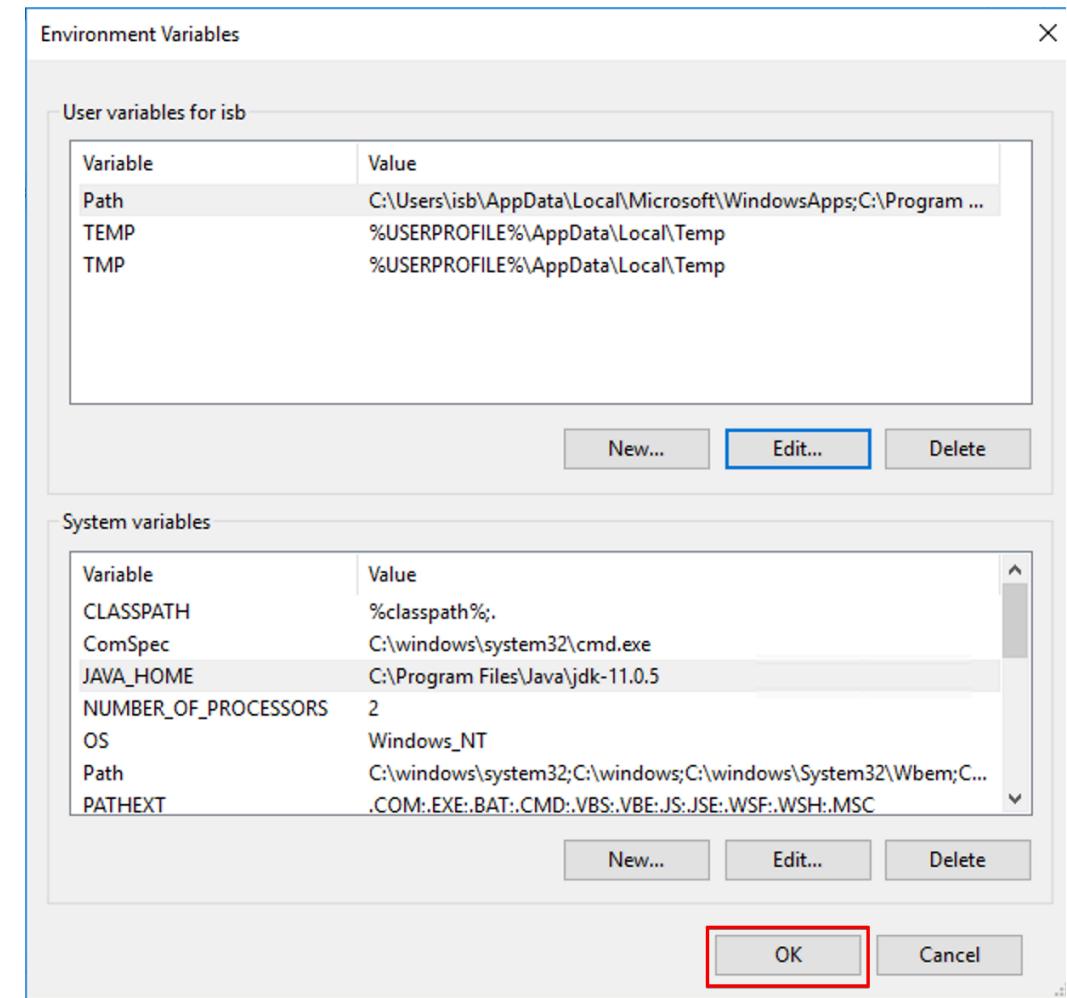
Variable : %JAVA\_HOME%\bin



## Jenkins 환경설정

## JDK 설치

- 추가내용 확인 후 “확인” 버튼 클릭하여 설정 저장



# Jenkins 환경설정

## JDK 설치

- 명령 프롬프트창에서 환경변수 정상 적용되었는지 확인

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

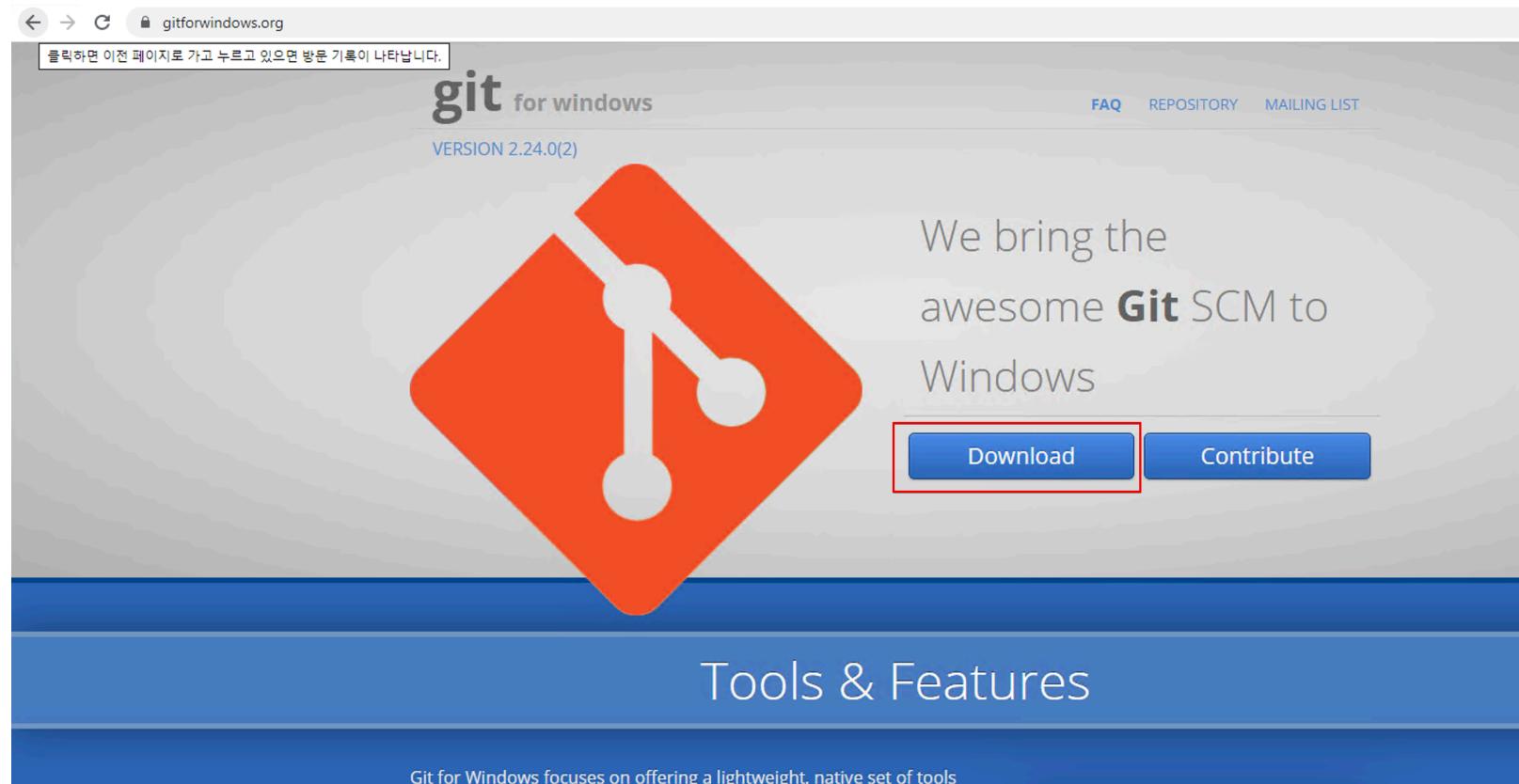
C:\Users\isp>java -version
java version "11.0.5" 2019-10-15 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.5+10-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.5+10-LTS, mixed mode)

C:\Users\isp>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                                Read options and filenames from file
  -Akey[=value]                                 Options to pass to annotation processors
  --add-modules <module>(<module>)*
    Root modules to resolve in addition to the initial modules, or all modules
    on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
    Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
    Specify where to find user class files and annotation processors
  -d <directory>
```

# Jenkins 환경설정

## Git 설치

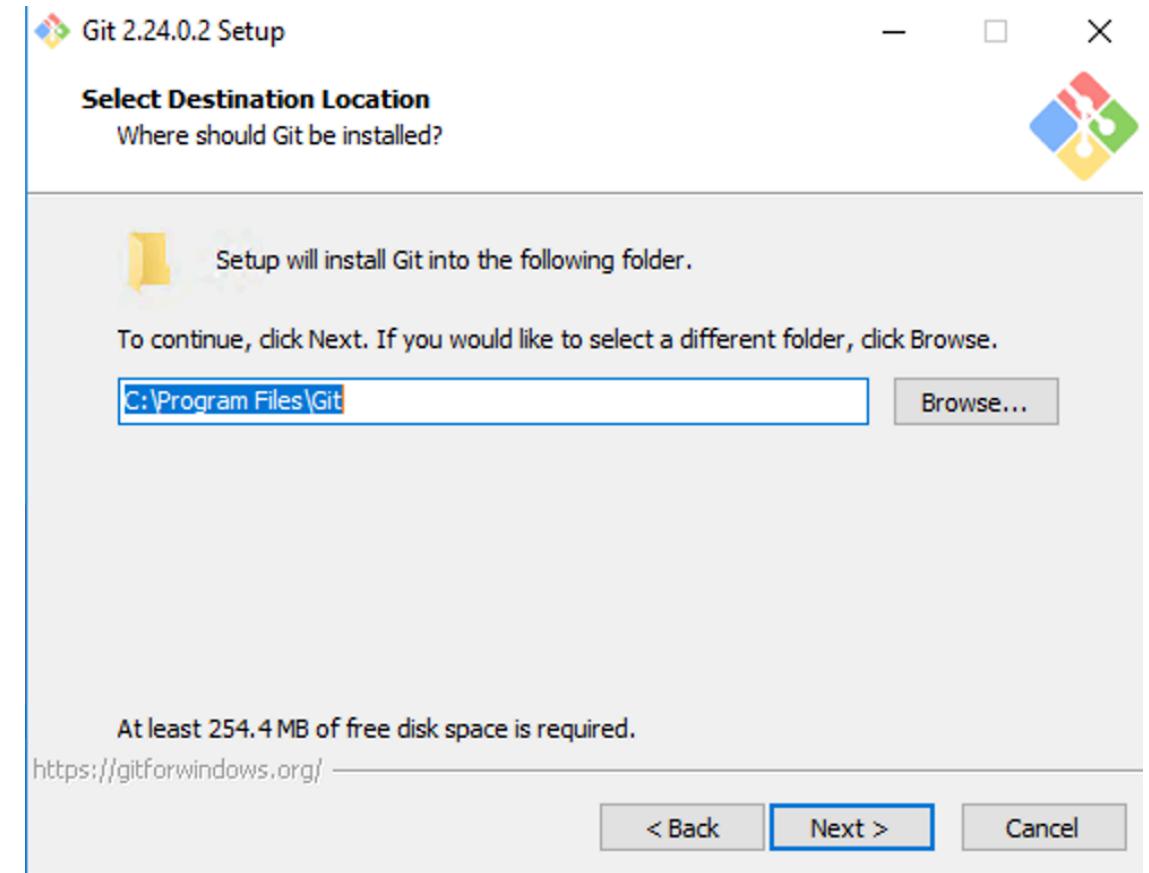
- <https://gitforwindows.org/>에서 설치파일 Download



# Jenkins 환경설정

## Git 설치

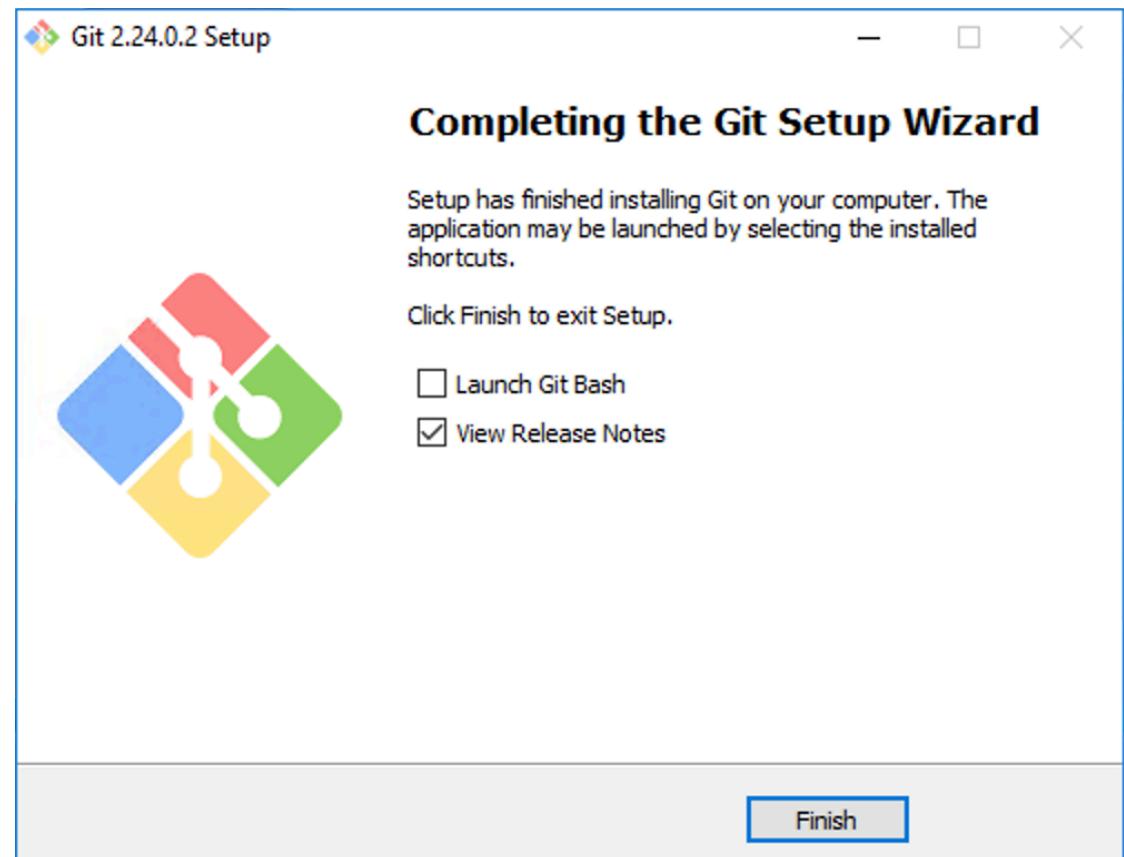
- <https://gitforwindows.org/>에서 설치파일 Download
- Default 설정으로 Next> 설치 진행



# Jenkins 환경설정

## Git 설치

- 설치 완료



# Jenkins 환경설정

## Gradle 설치

- <https://gradle.org/releases/>에서 최신버전의 설치파일 다운로드

### Getting Started Resources

The Gradle team offers free [training](#) courses each month.

There are many [Gradle tutorials](#) available to help you get started quickly. Source distributions also include many [working samples](#) for which guides are not yet written.

#### ↳ v6.0.1

📅 Nov 18, 2019

- Download: [binary-only](#) or [complete](#)
- [User Manual](#)
- [API Javadoc](#)
- [DSL Reference](#)
- [Release Notes](#)

#### ↳ v6.0

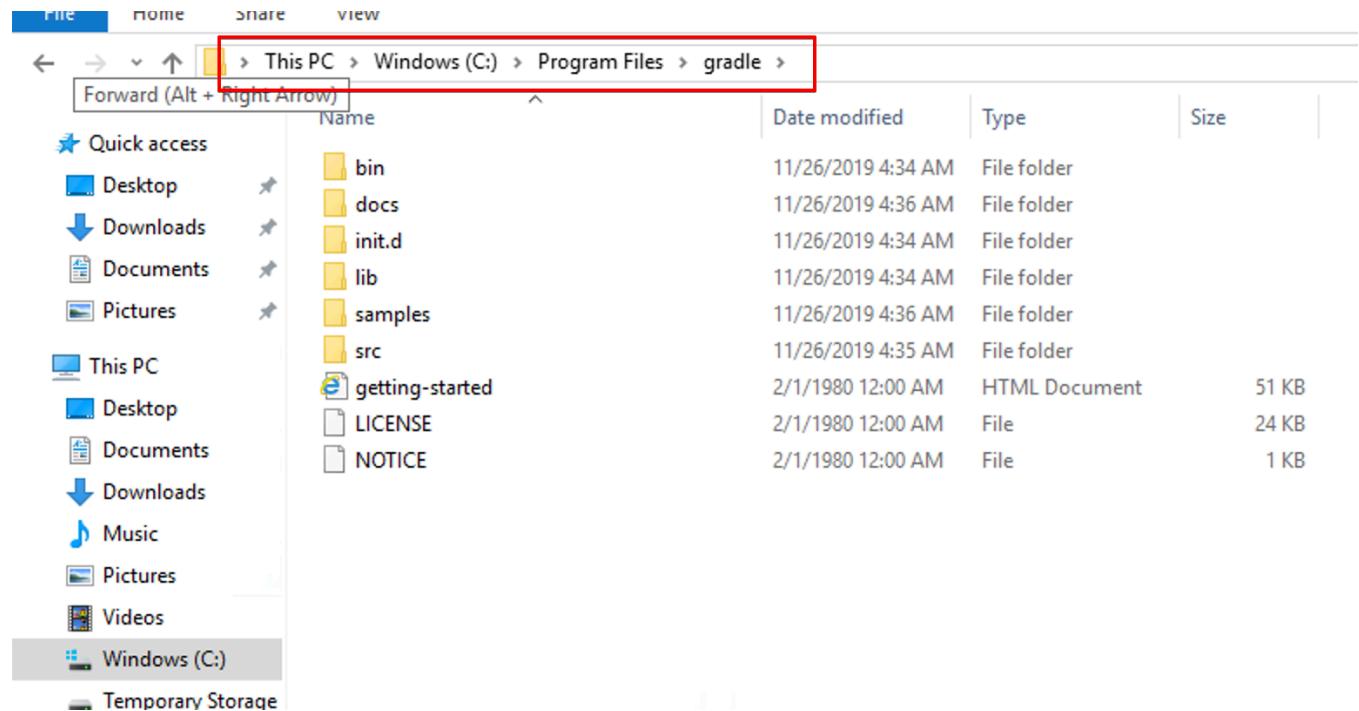
📅 Nov 08, 2019

- Download: [binary-only](#) or [complete](#)
- [User Manual](#)
- [API Javadoc](#)

# Jenkins 환경설정

# Gradle 설치

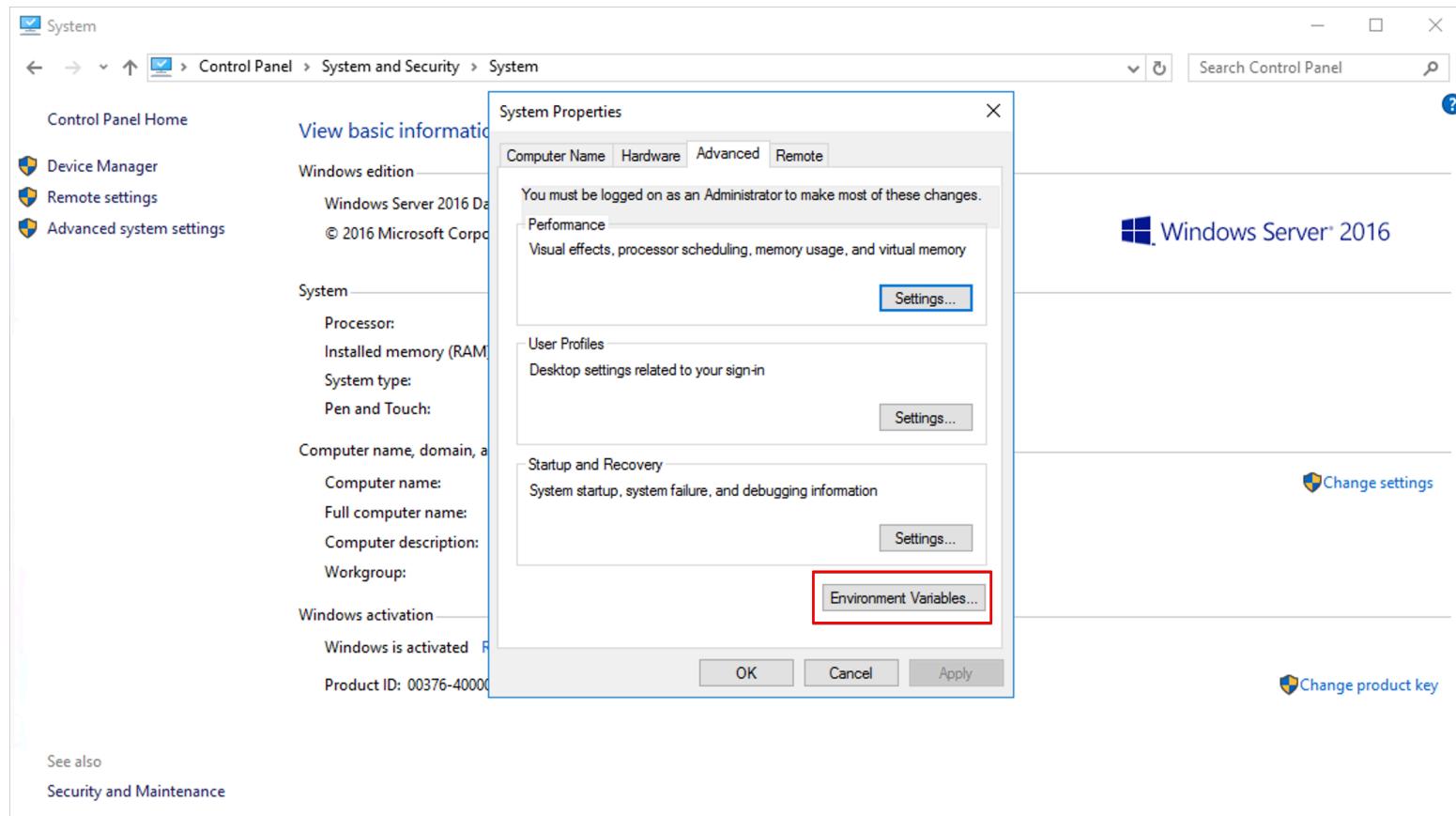
- Download 받은 Gradle 설치파일의 압축을 해제한 후 디렉토리명을 gradle로 변경
  - 변경한 디렉토리를 C:\Program Files 경로에 복사



## Jenkins 환경설정

# Gradle 설치

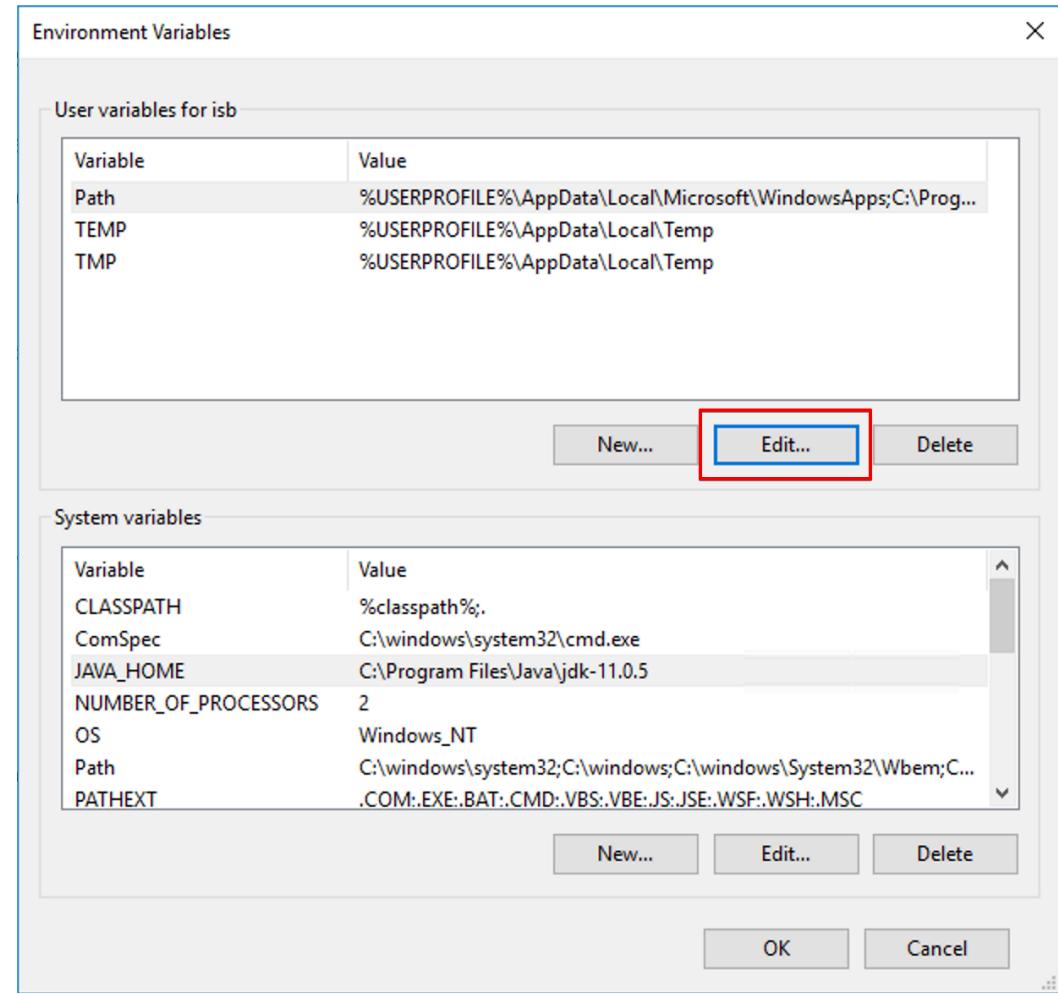
- 제어판 -> 시스템 및 보안 -> 시스템 ->  
고급 시스템 설정 -> 고급 -> 환경변수



## Jenkins 환경설정

# Gradle 설치

- 유저 변수의 Path 선택 후 편집 버튼 클릭

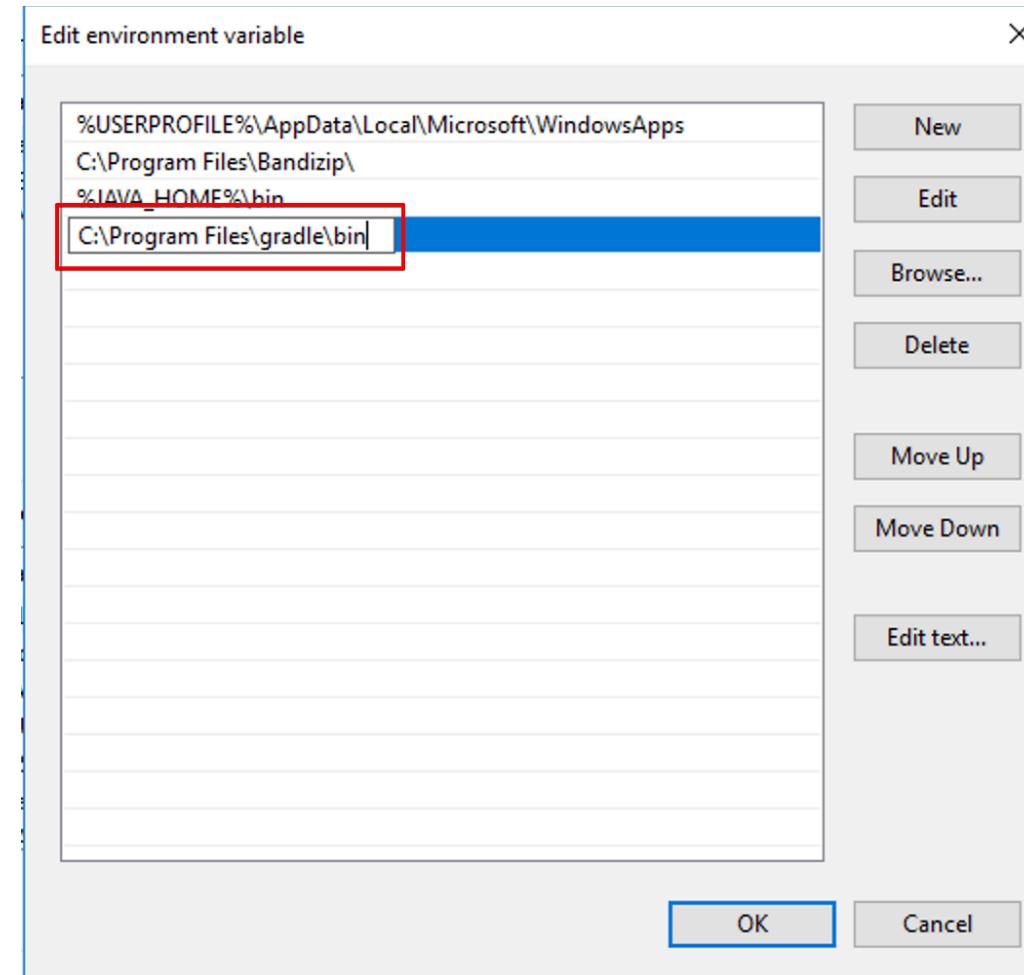


# Jenkins 환경설정

## JDK 설치

- “새로 만들기” 버튼 클릭 후 변수값 저장

Variable : C:\Program Files\gradle\bin



## Jenkins 환경설정

## Gradle 설치 확인

- 검색 -> cmd
  - 명령 프롬프트 창에서 gradle -v

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\isb>gradle -v

-----
Gradle 6.0.1
-----
Build time: 2019-11-18 20:25:01 UTC
Revision: fad121066a68c4701acd362daf4287a7c309a0f5

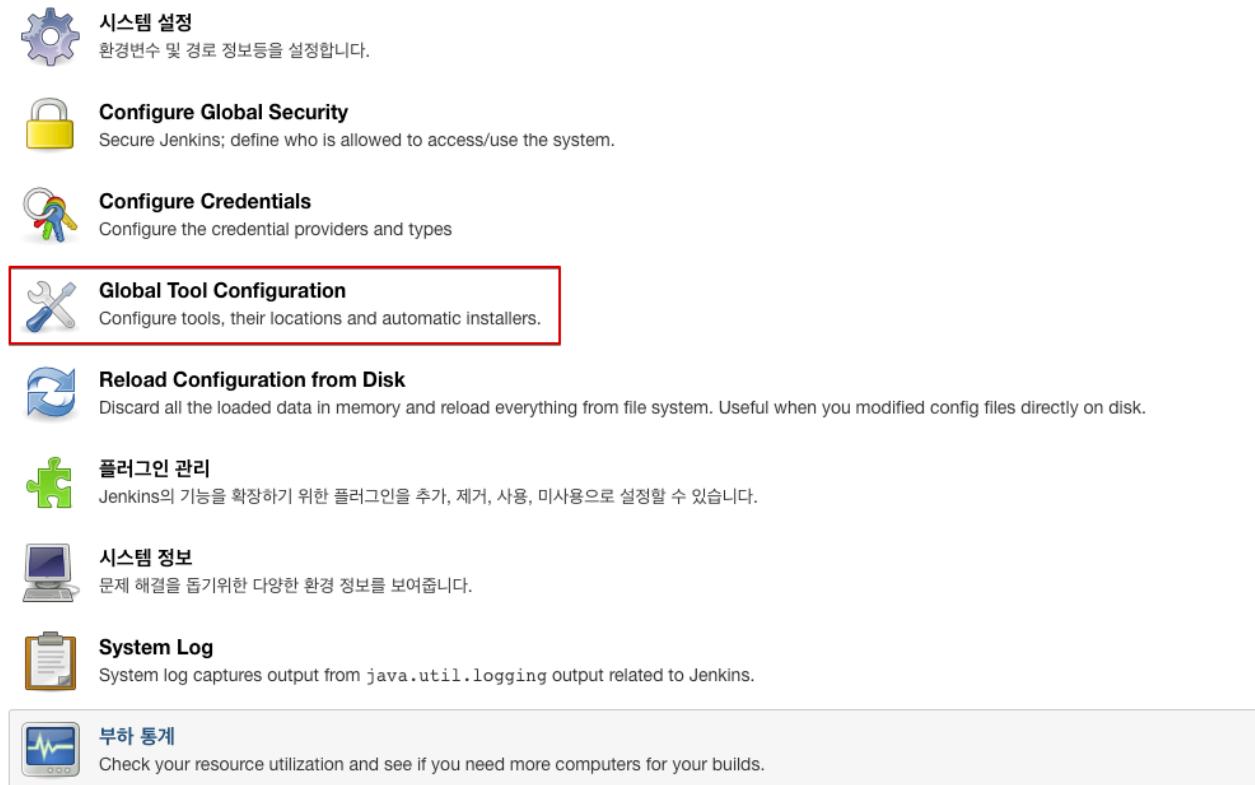
Kotlin: 1.3.50
Groovy: 2.5.8
Ant: Apache Ant(TM) version 1.10.7 compiled on September 1 2019
JVM: 11.0.5 (Oracle Corporation 11.0.5+10-LTS)
OS: Windows Server 2016 10.0 amd64

C:\Users\isb>
```

# Jenkins 환경설정

## Jenkins 관리툴 설정

- Jenkins 메뉴에서 Global Tool Configuration 클릭



# Jenkins 환경설정

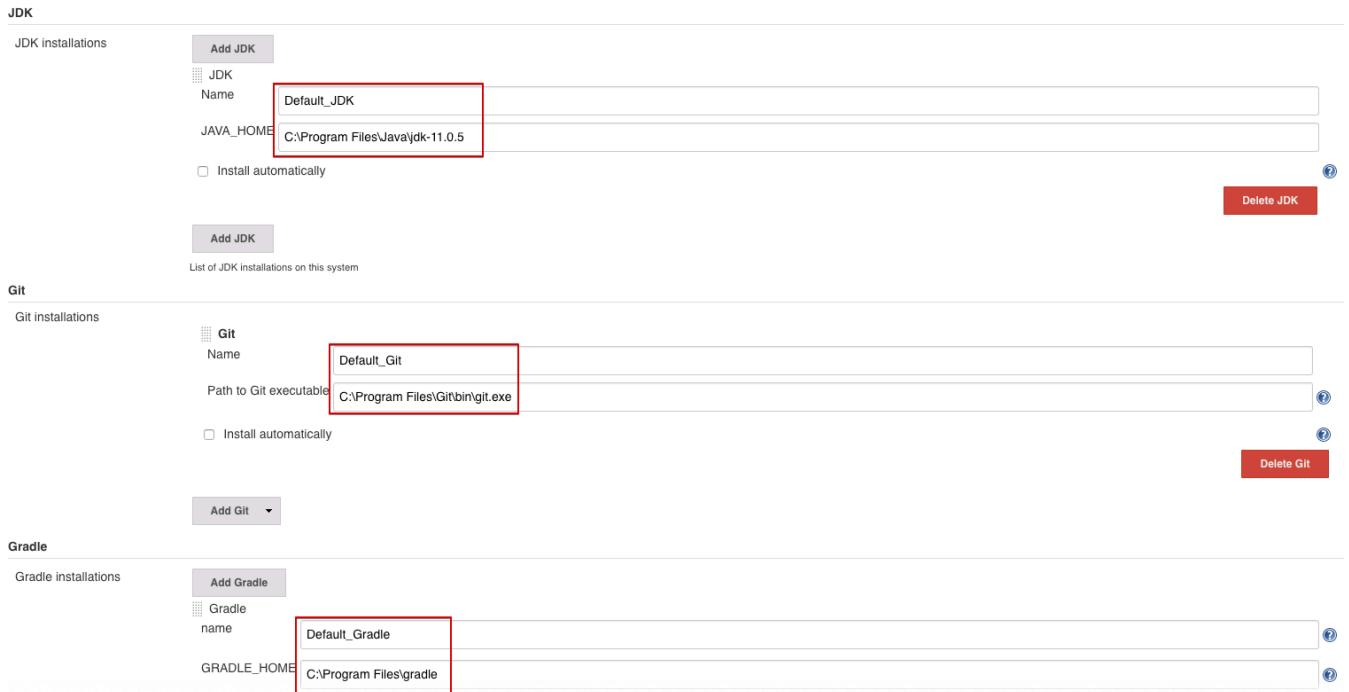
## Jenkins 관리툴 설정

- JDK installations
    - Name : Default\_JDK
    - JAVA\_HOME : C:\Program Files\Java\jdk-11.0.5

- Git installations
    - Name : Default\_Git
    - Path to Git executable : C:\Program

- Files\Git\bin\git.exe

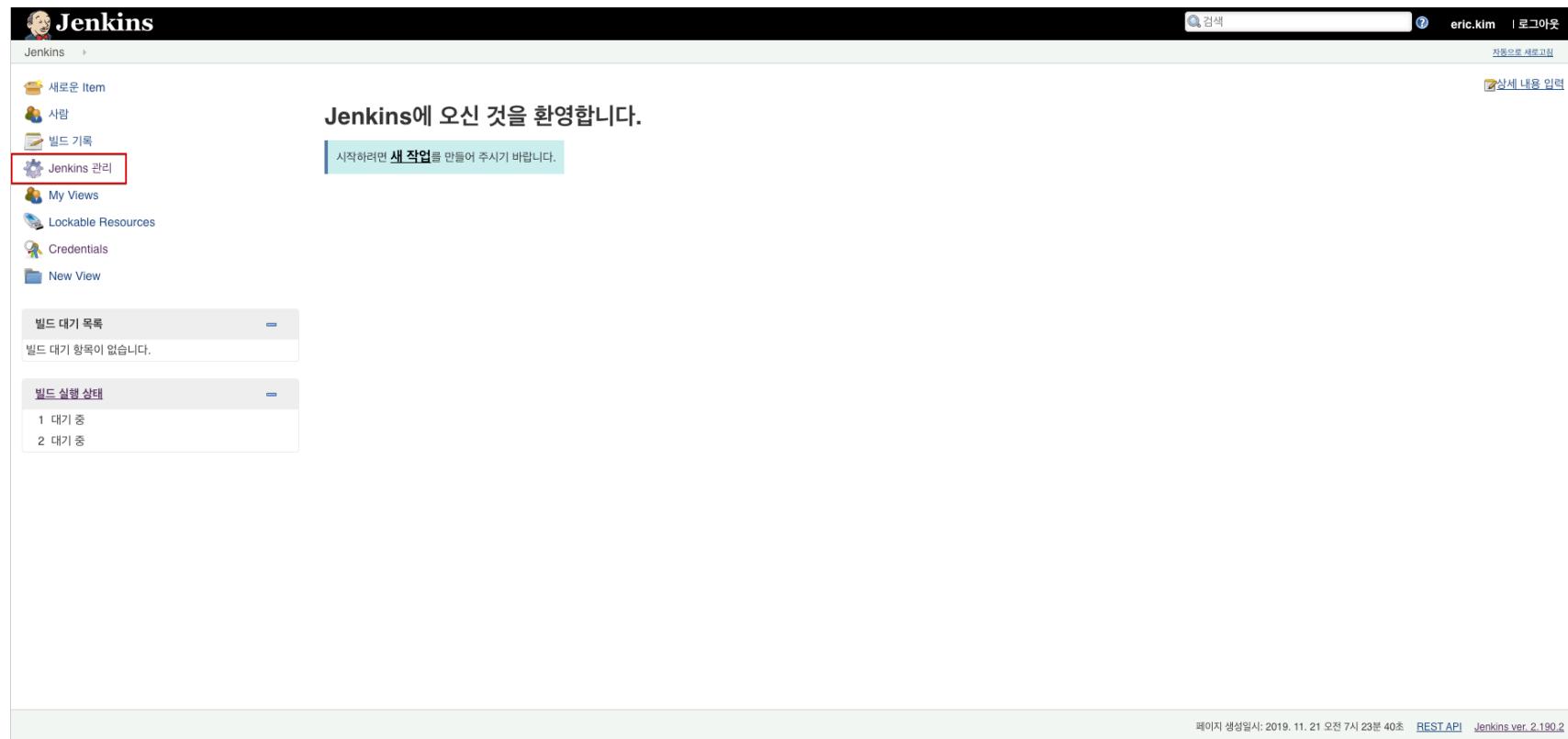
- Gradle installations
    - Name : Default\_Gradle
    - GRADLE\_HOME : C:\Program Files\gradle
  - 설정 변경 후 저장



## Jenkins 환경설정

# Jenkins 플러그인 설치

- “Jenkins 관리” 메뉴 클릭



# Jenkins 환경설정

## Jenkins 플러그인 설치

- “플러그인 관리” 클릭

 Jenkins

Jenkins >

 새로운 Item

 사람

 빌드 기록

 Jenkins 관리

 My Views

 Lockable Resources

 Credentials

 New View

**빌드 대기 목록**

빌드 대기 항목이 없습니다.

**빌드 실행 상태**

1 대기 중  
2 대기 중

## Jenkins 관리

이전 버전의 Jenkins 로 복원

 **시스템 설정**  
환경변수 및 경로 정보들을 설정합니다.

 **Configure Global Security**  
Secure Jenkins; define who is allowed to access/use the system.

 **Configure Credentials**  
Configure the credential providers and types

 **Global Tool Configuration**  
Configure tools, their locations and automatic installers.

 **Reload Configuration from Disk**  
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

 **플러그인 관리**  
Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.

 **시스템 정보**  
문제 해결을 돋기위한 다양한 환경 정보를 보여줍니다.

 **System Log**  
System log captures output from java.util.logging output related to Jenkins.

 **도움말**

# Jenkins 환경설정

## Jenkins 플러그인 설치

- GitHub 검색하여 다음 Plug-In 설치

## GitHub Authentication

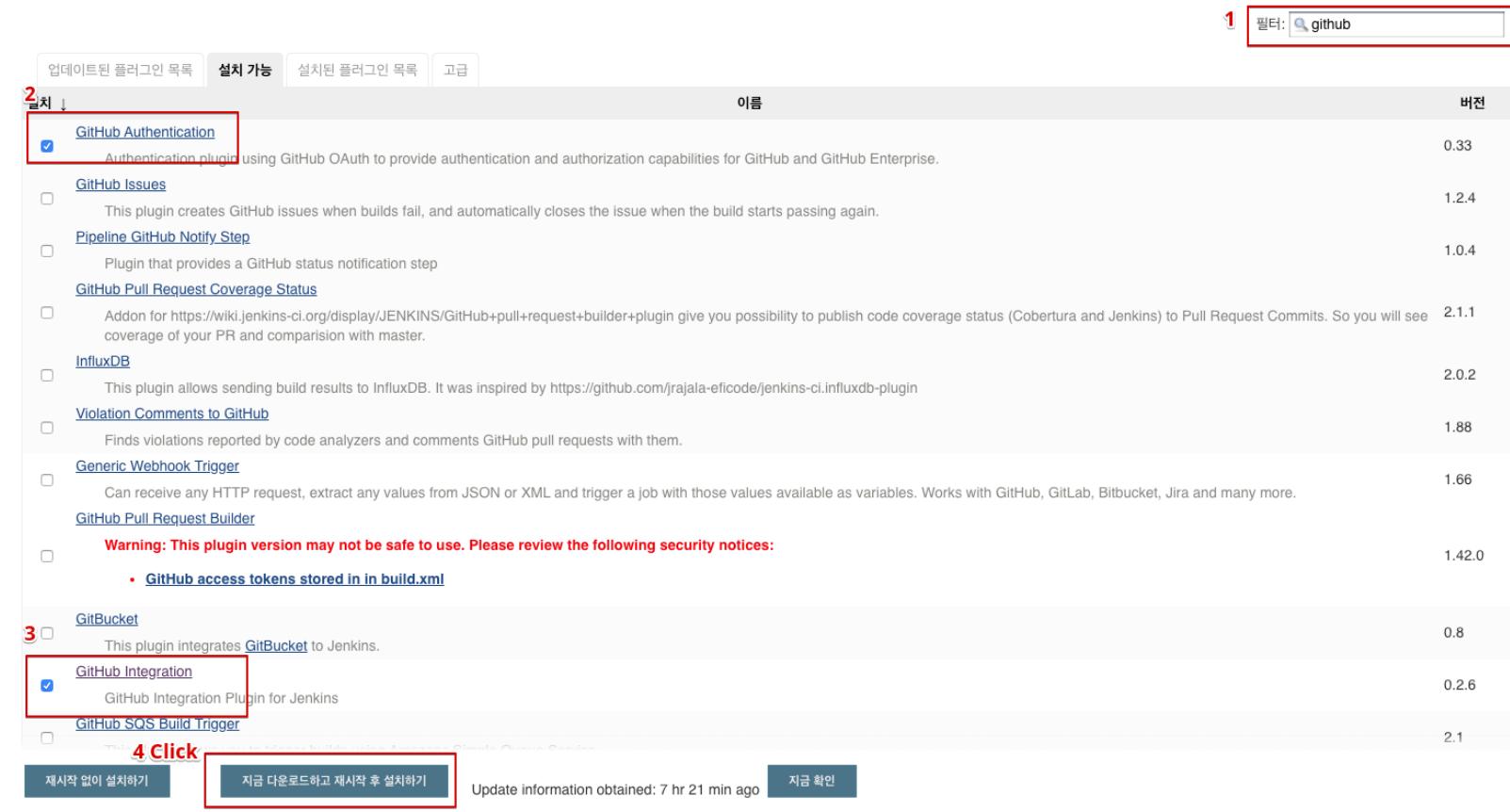
## GitHub Integration

- Plug-In 설치 후 Jenkins 자동 재시작
  - 반응 없는 경우 Browser Refresh
  - 재시작이 적용되지 않은 경우

(jenkins\_url)/safeRestart

- Server Local에서는 윈도우 서비스에서

## Jenkins stop & start



# Jenkins 환경설정

# Jenkins – GitHub Enterprise 연동

- “Jenkins 관리” 메뉴에서 시스템 설정 클릭

Jenkins 관리

## 이전 버전의 Jenkins 로 복원



## 시스템 설정

환경변수 및 경로 정보등을 설정합니다.



## Configure Global Security

Secure Jenkins; define who is allowed to access/use the system.



## Configure Credentials

Configure the credential providers and types



## Global Tool Configuration

Configure tools, their locations and automatic installers.



### **Reload Configuration from Disk**

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.



플러그인 관리

Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.

# Jenkins 환경설정

## Jenkins – GitHub Enterprise 연동

- GitHub Enterprise Servers 입력란에 다음 내용 등록

API endpoint : <http://githubbedu.tangunsoft.com/api/v3>

Name : GitHub Enterprise Edu

- 입력 후 저장

### GitHub Enterprise Servers

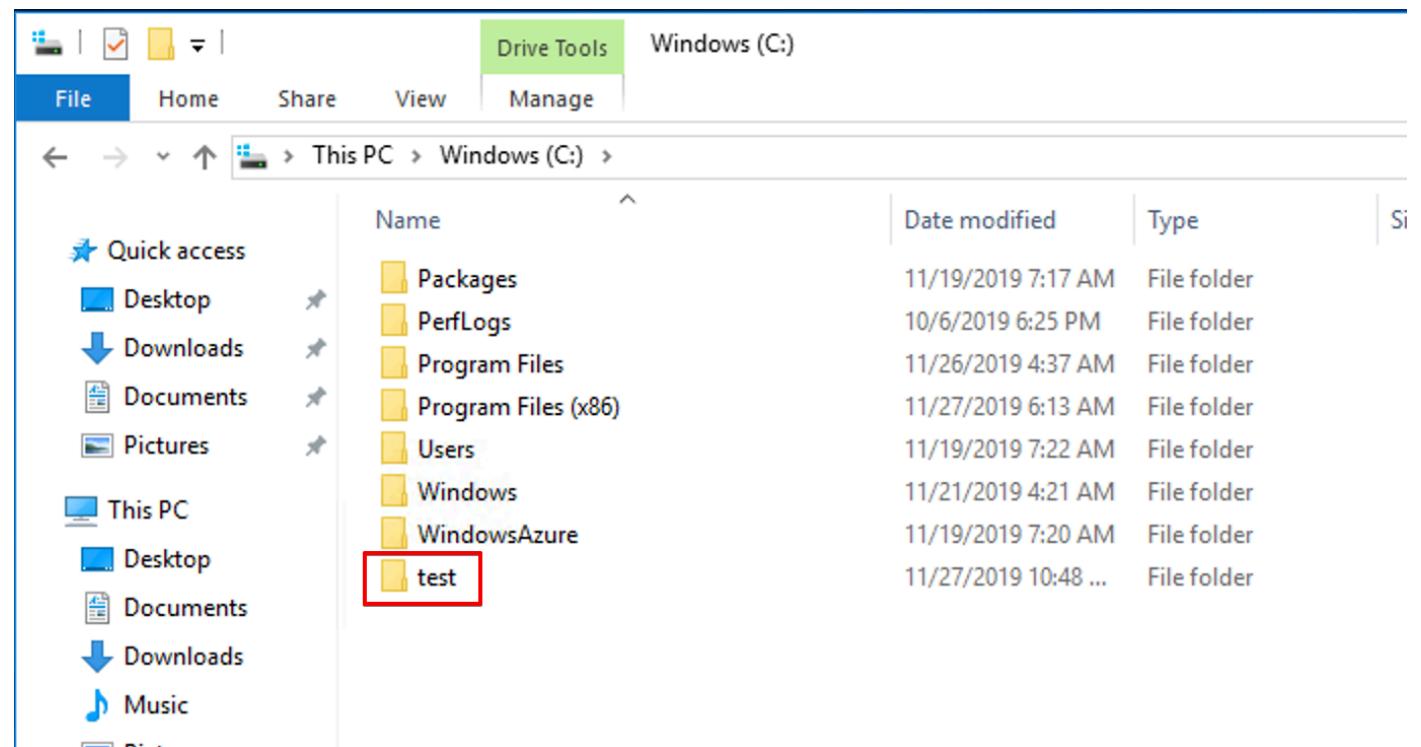
API endpoint	<input type="text" value="http://githubbedu.tangunsoft.com/api/v3"/>
GitHub Enterprise server verified	
Name	<input type="text" value="GitHub Enterprise Edu"/>

# 배포서버 구성

## 배포서버 구성

## SSH 서버 구성

- C:\₩에 배포 경로로 사용될 test 디렉토리 생성



# 배포서버 구성

## SSH 서버 구성

- <http://www.freesshd.com/?ctt=download>에서

freeFTPd.exe 설치파일 다운로드

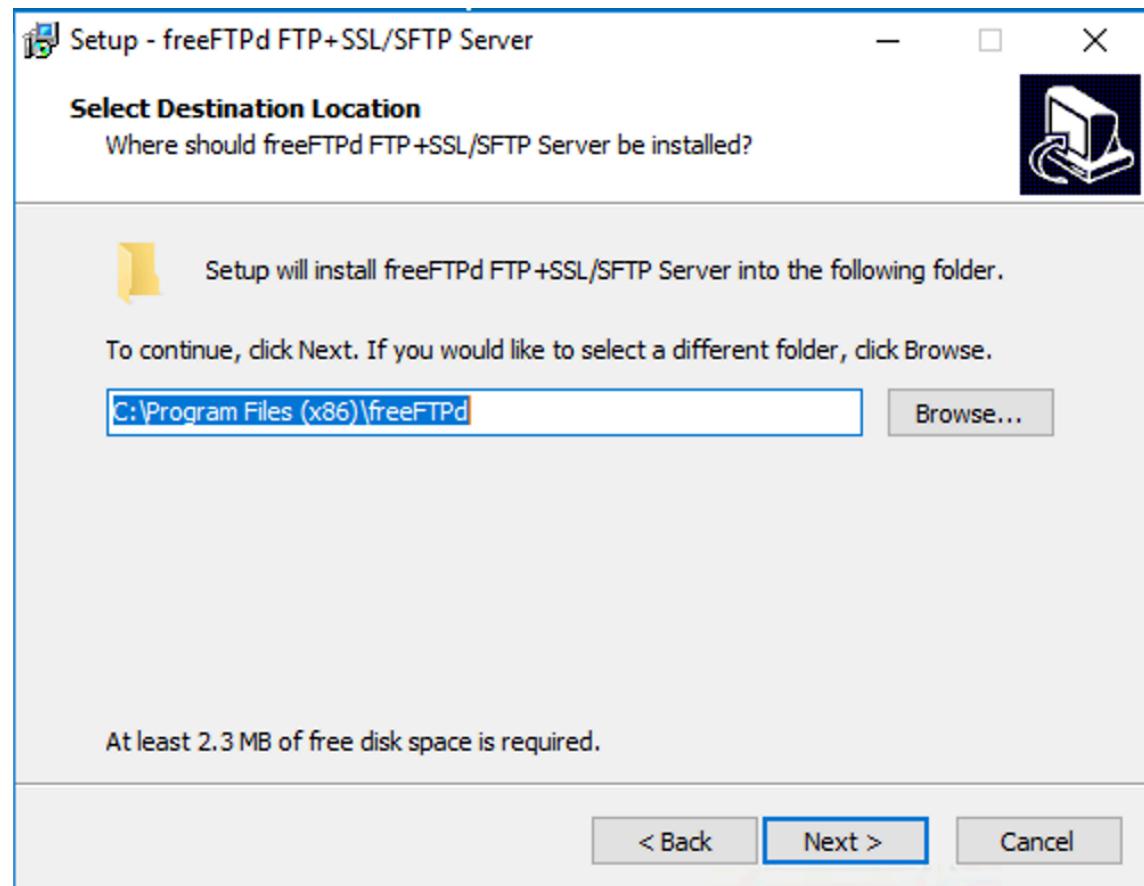
The screenshot shows a web browser displaying the [freeSSHD and freeFTPd](http://www.freesshd.com/?ctt=download) download page. The page has a blue header with the title and navigation links for Home, Overview, and Download. Below the header is a section titled "Downloads" with a sub-instruction: "Here you can download the latest version of freeSSHD and freeFTPd. Please have in mind that you must have administrator privileges to install both applications." A table below lists three files:

name	version
freeSSHD.exe	1.3.1
freeFTPd.exe	1.0.13
freeUserImport.exe	1.0.0

# 배포서버 구성

## SSH 서버 구성

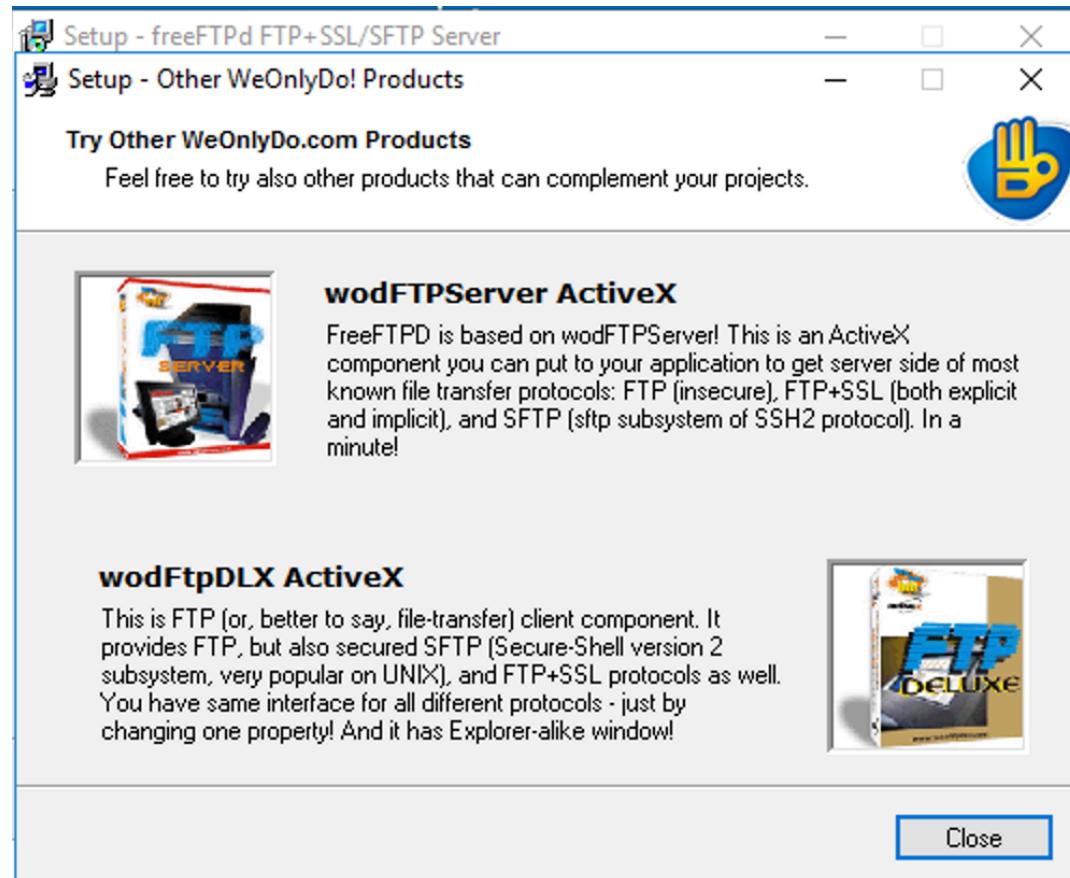
- freeFTPd.exe 설치파일 실행하여 설치 진행
- 설치 설정 기본으로 하여 설치 진행



# 배포서버 구성

## SSH 서버 구성

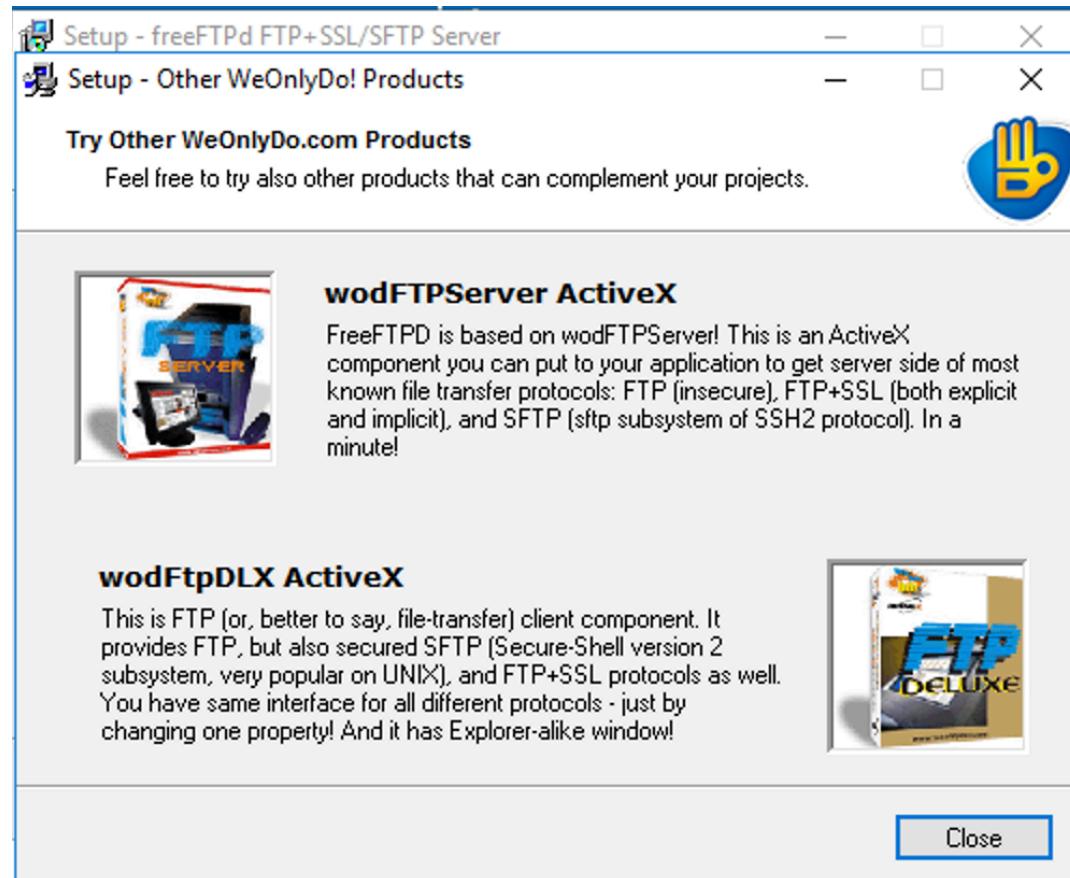
- freeFTPD 설치 완료



# 배포서버 구성

## SSH 서버 구성

- freeFTPD 설치 완료

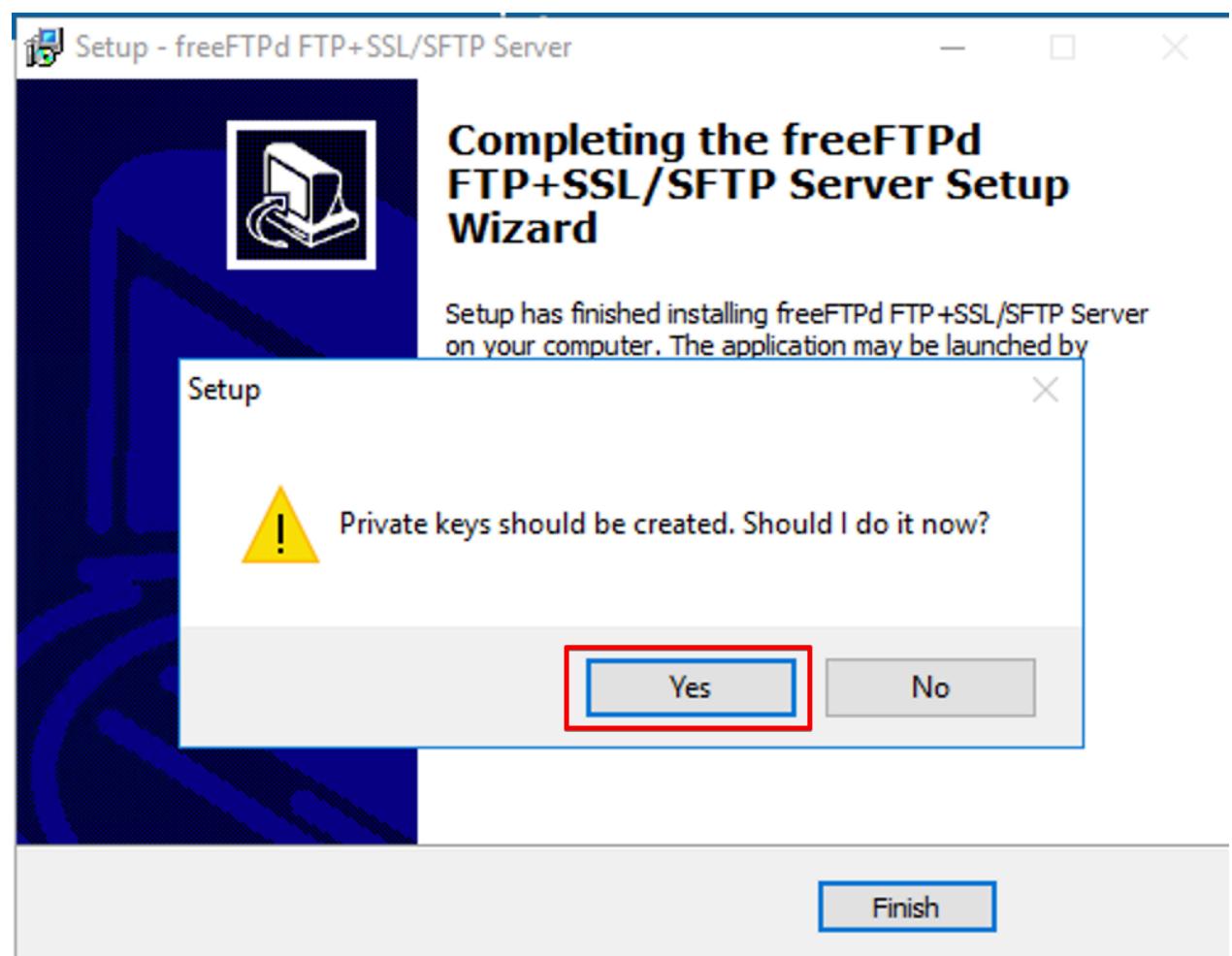


# 배포서버 구성

## SSH 서버 구성

- 개인키 생성 Yes 클릭

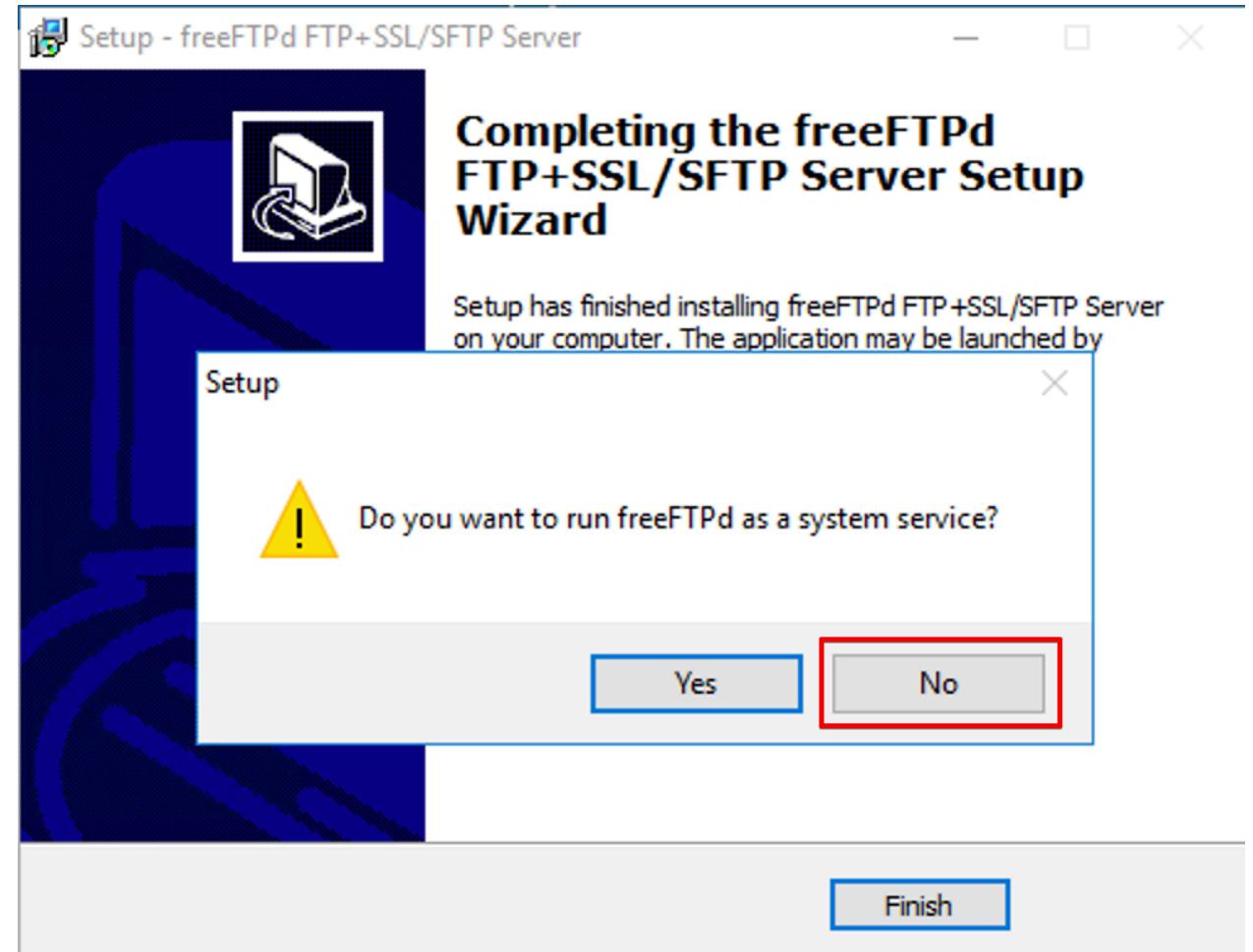
(개인키를 생성하나 시나리오에서는 ID / PW 이용)



# 배포서버 구성

## SSH 서버 구성

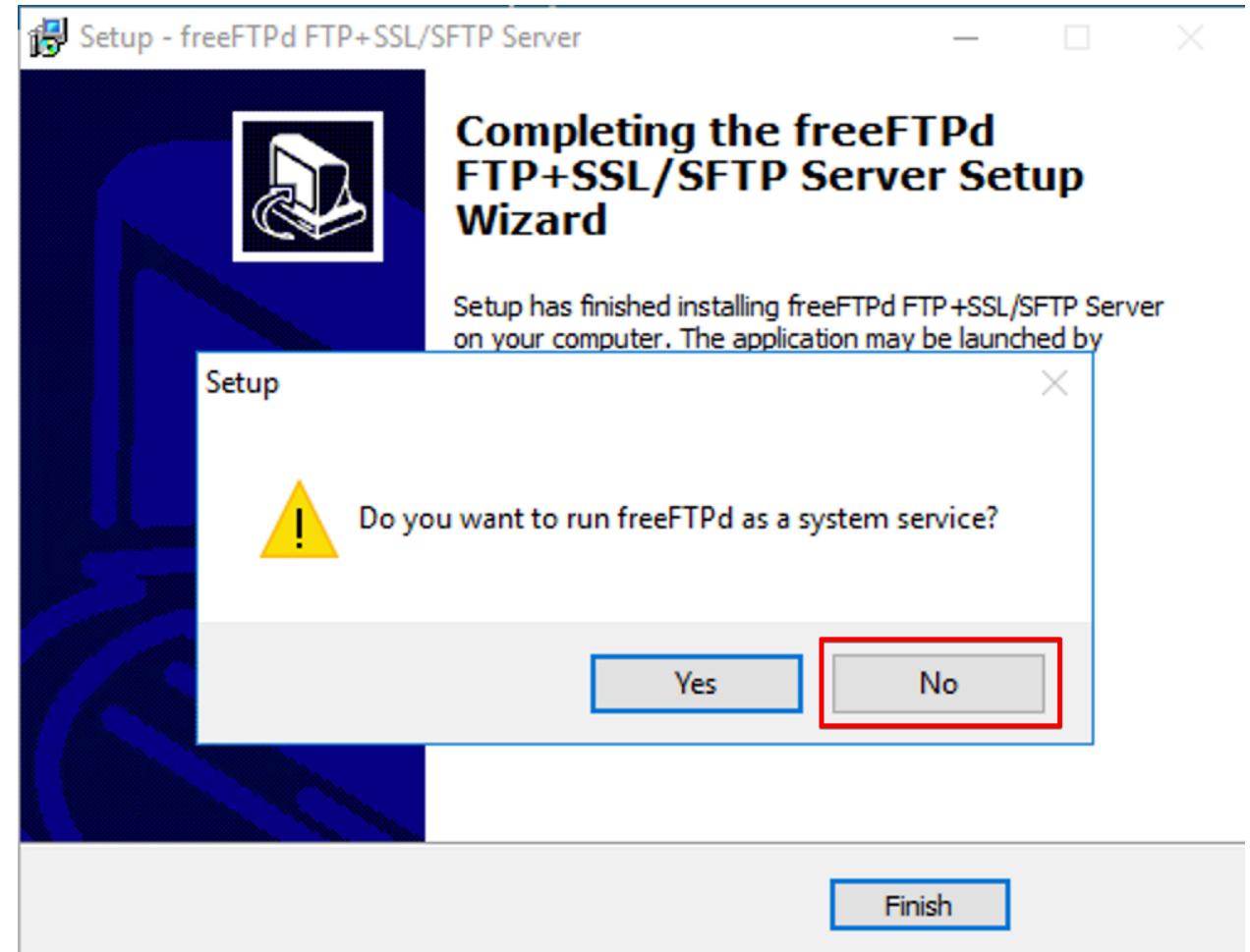
- freeFTPD를 시스템 서비스로 구동할 것인지에 문의
- 프로그램 불안정으로 인하여 시스템 서비스에서 정상 작동되지 않으므로 No 클릭



# 배포서버 구성

## SSH 서버 구성

- freeFTPD를 시스템 서비스로 구동할 것인지에 문의
- 프로그램 불안정으로 인하여 시스템 서비스에서 정상 작동되지 않으므로 No 클릭



# 배포서버 구성

## SSH 서버 구성

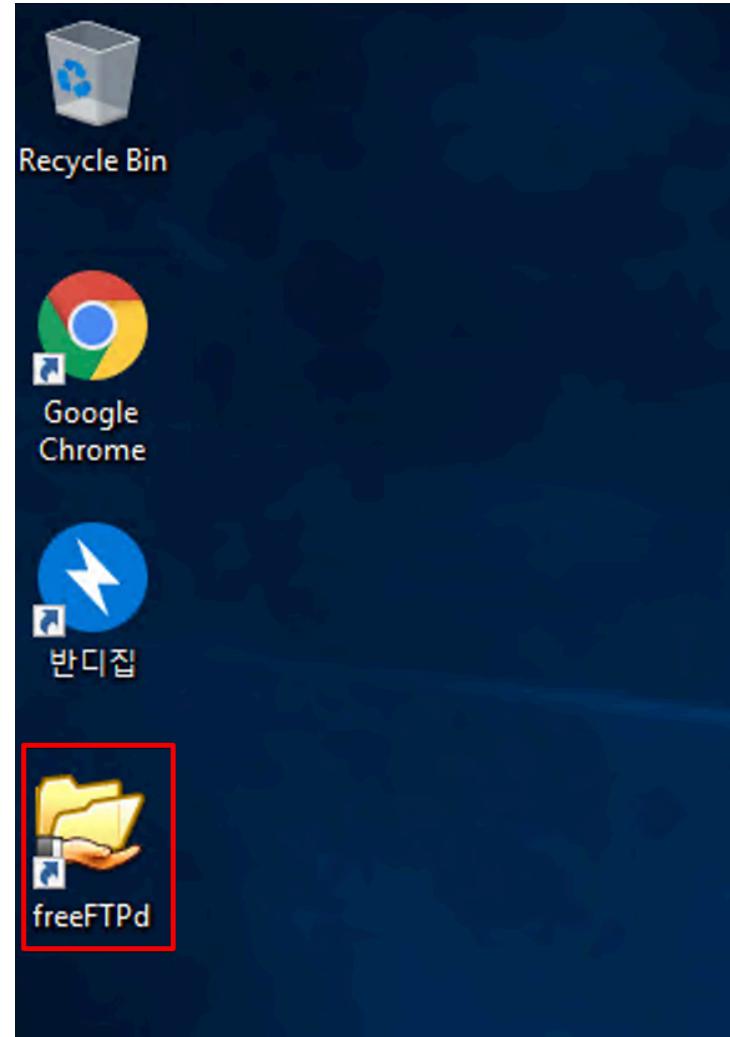
- 초기 설정 완료
- Finish 버튼 클릭



# 배포서버 구성

## SSH 서버 구성

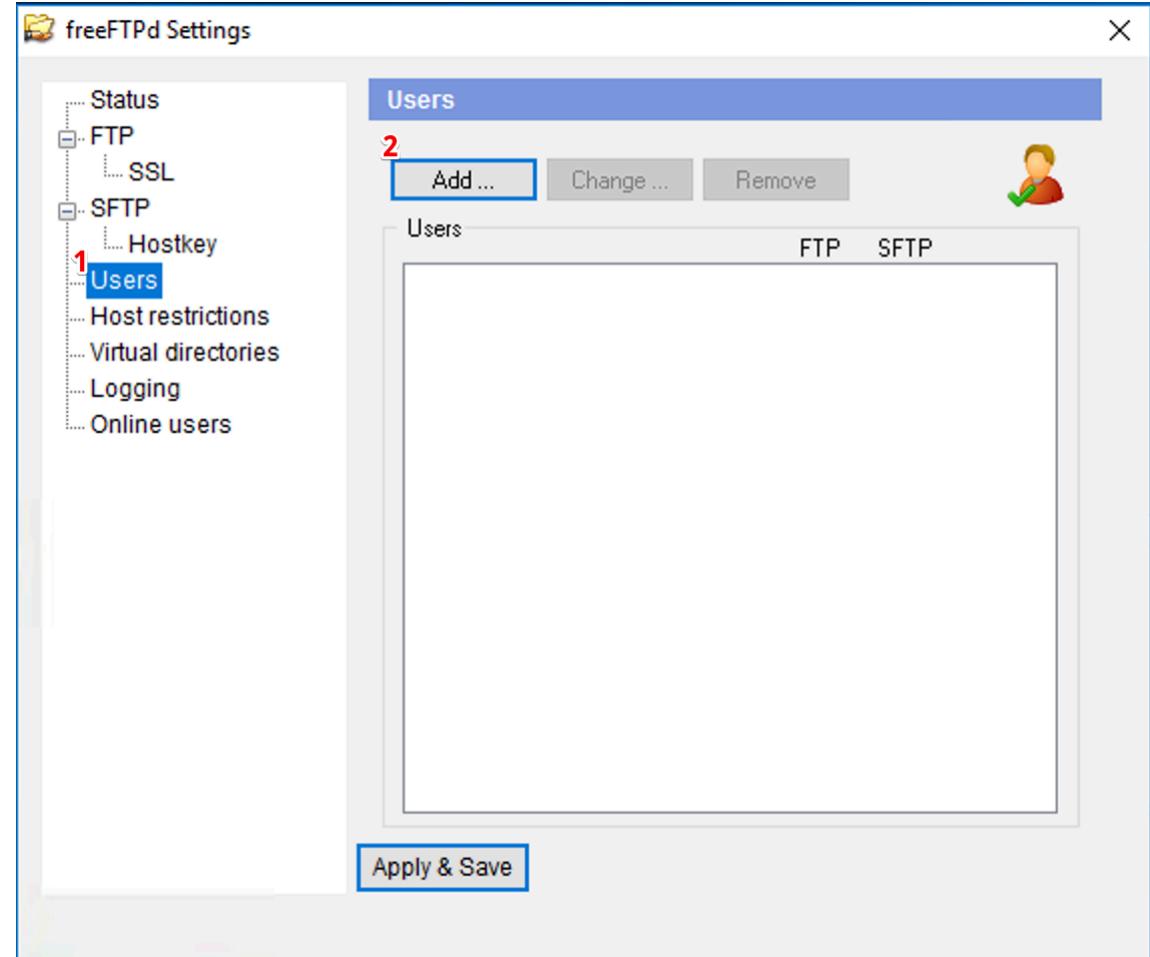
- FreeFTPD 실행



# 배포서버 구성

## SSH 서버 구성

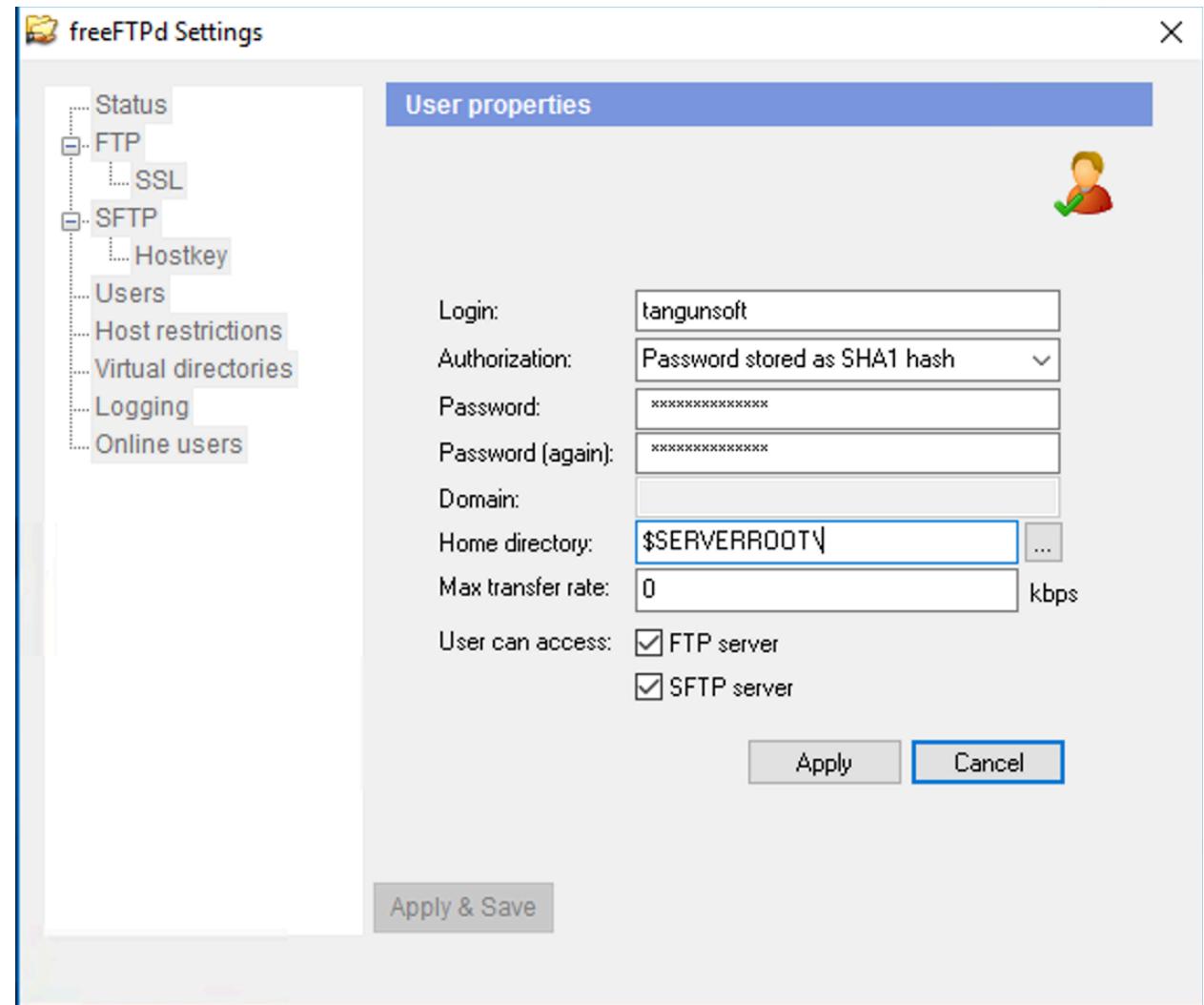
- 좌측 메뉴란에서 “Users” 클릭
- Users 란에서 “Add” 버튼 클릭



# 배포서버 구성

## SSH 서버 구성

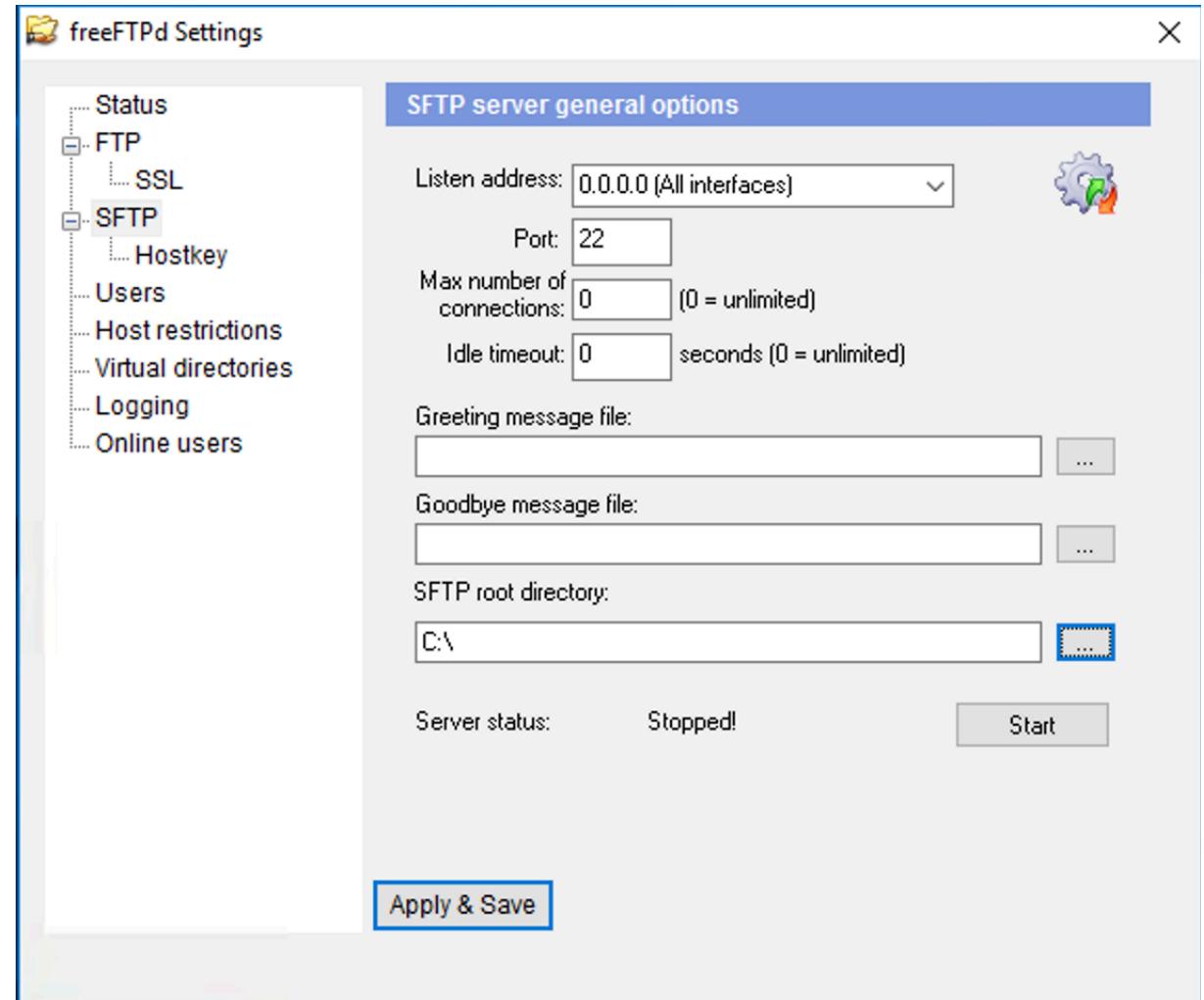
- Login : SSH로 접속할 ID 설정
- Authorization : Password 방식 설정
- Password : SSH로 접속할 PW 설정
- Home Directory : \$SERVERROOT 설정
- 모든 설정 입력 후 "Apply" 버튼 클릭



# 배포서버 구성

## SSH 서버 구성

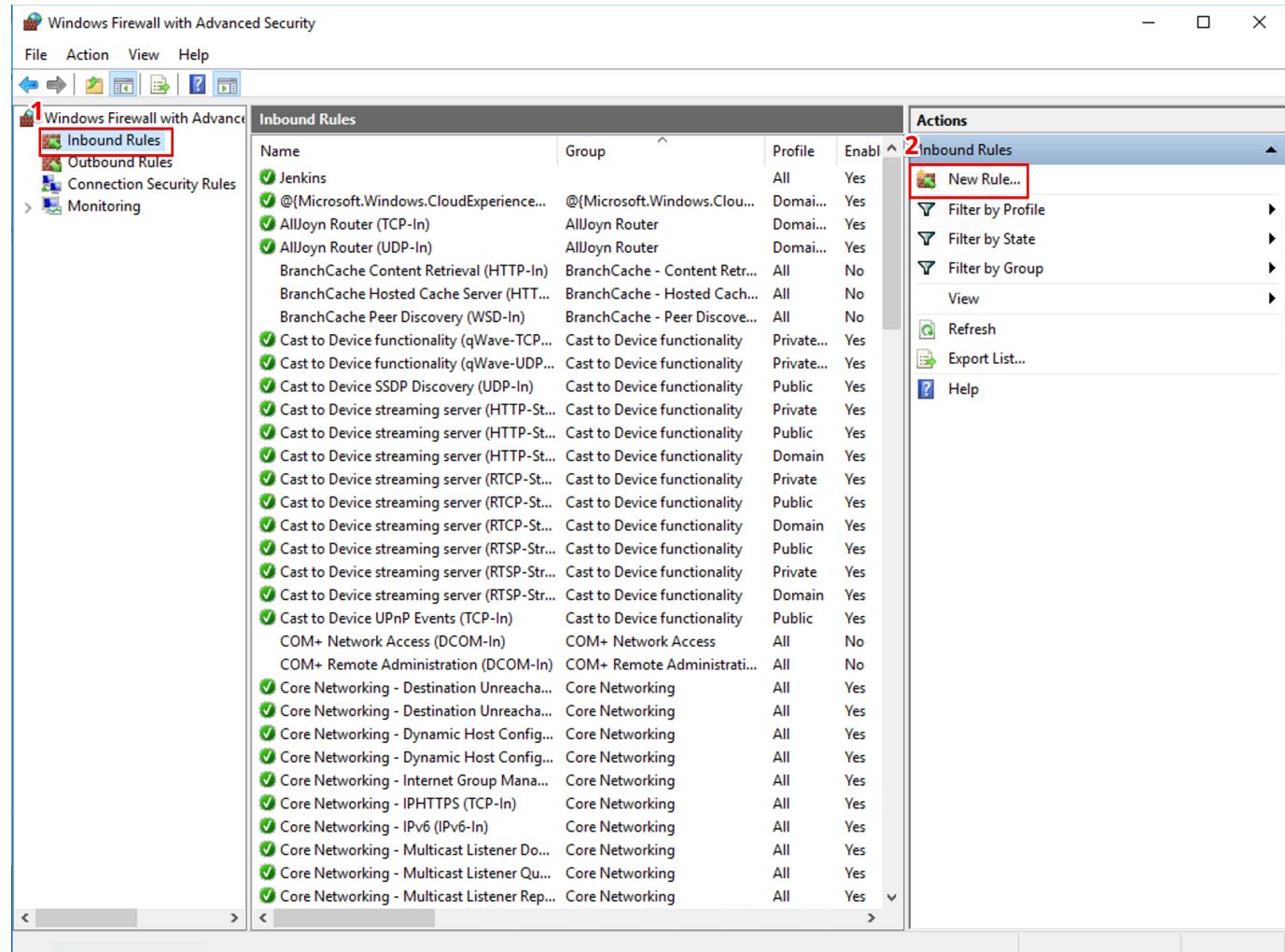
- SFTP Root directory : C:\WWW 로 변경
- Server Status : Start로 변경
- Apply & Save 버튼 클릭



## 배포서버 구성

## SSH 서버 구성

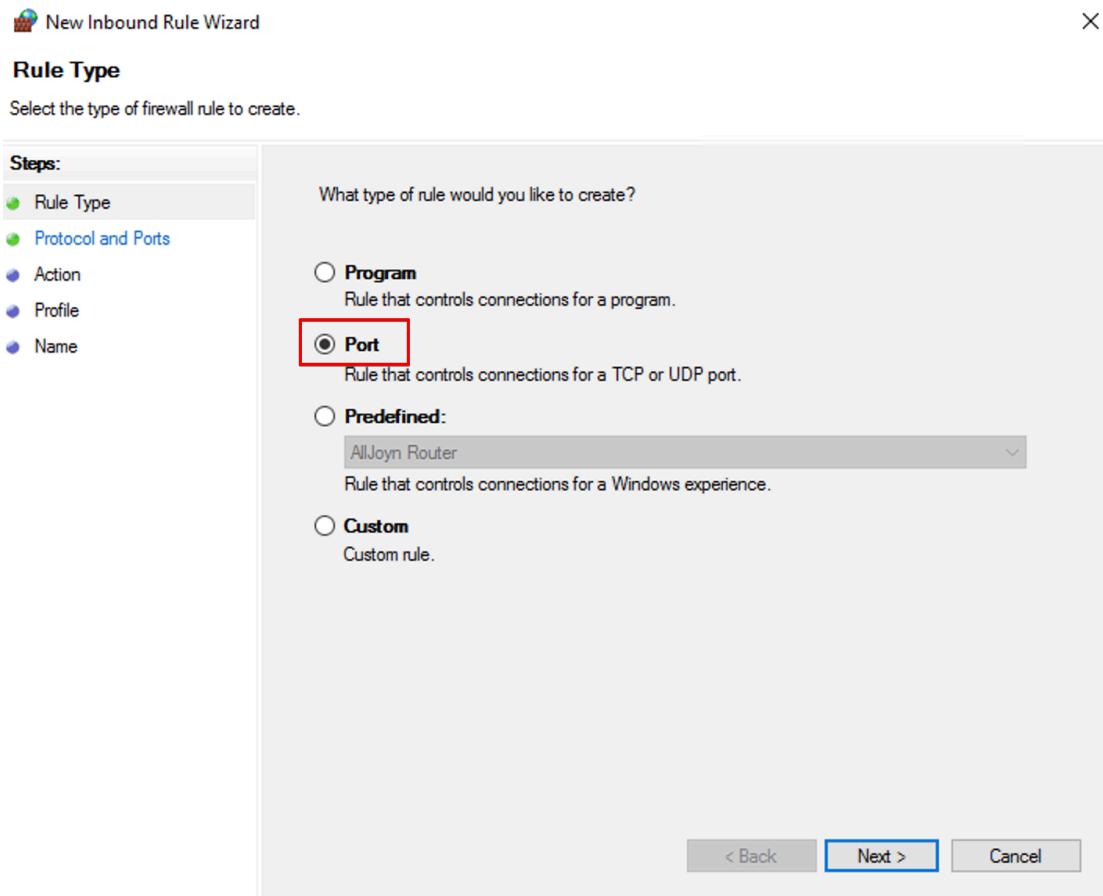
- 서버 시스템 검색에서 "Windows Firewall with Advanced Security" 검색
  - Inbound Rules 선택
  - New Rules 클릭



# 배포서버 구성

## SSH 서버 구성

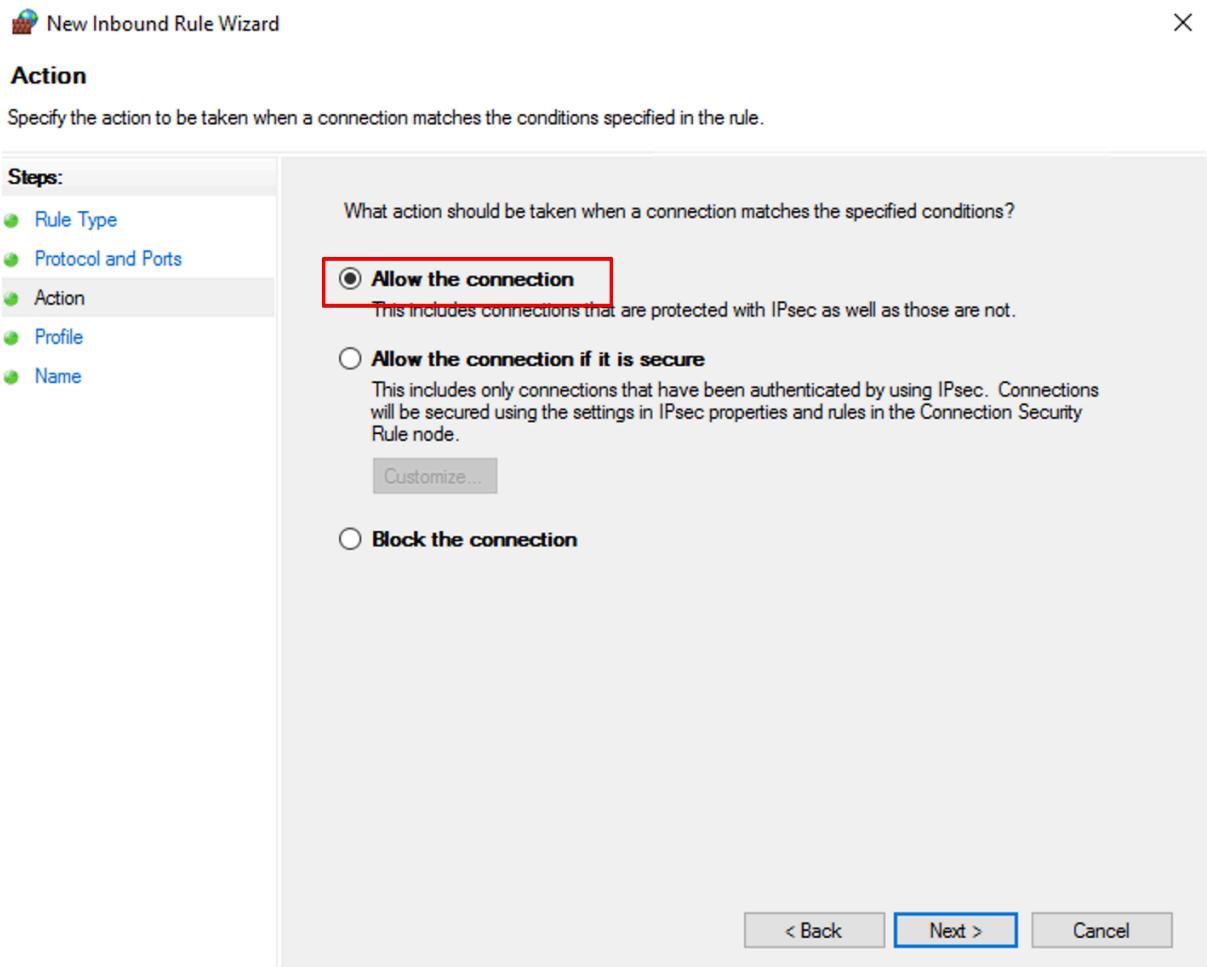
- Rule Type "Port" 선택



## 배포서버 구성

## SSH 서버 구성

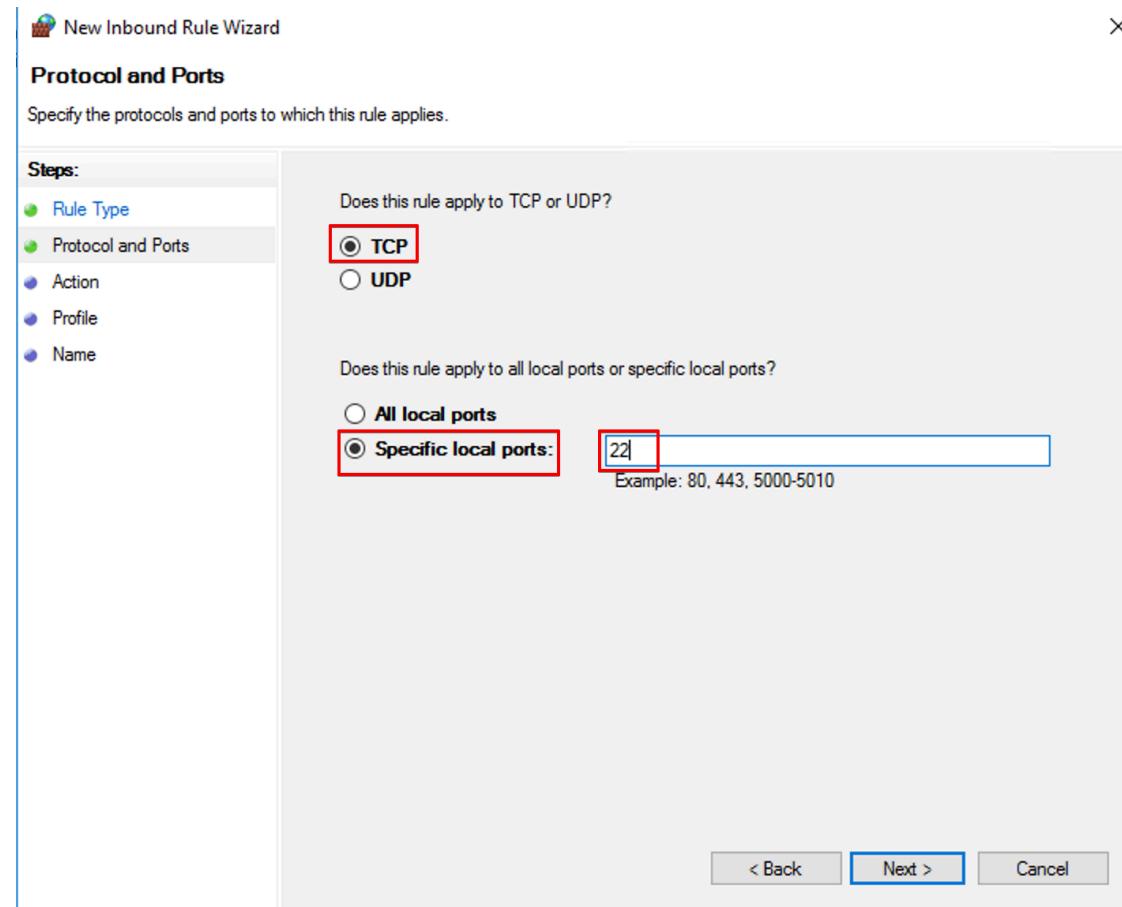
- Allow the connection 선택



# 배포서버 구성

## SSH 서버 구성

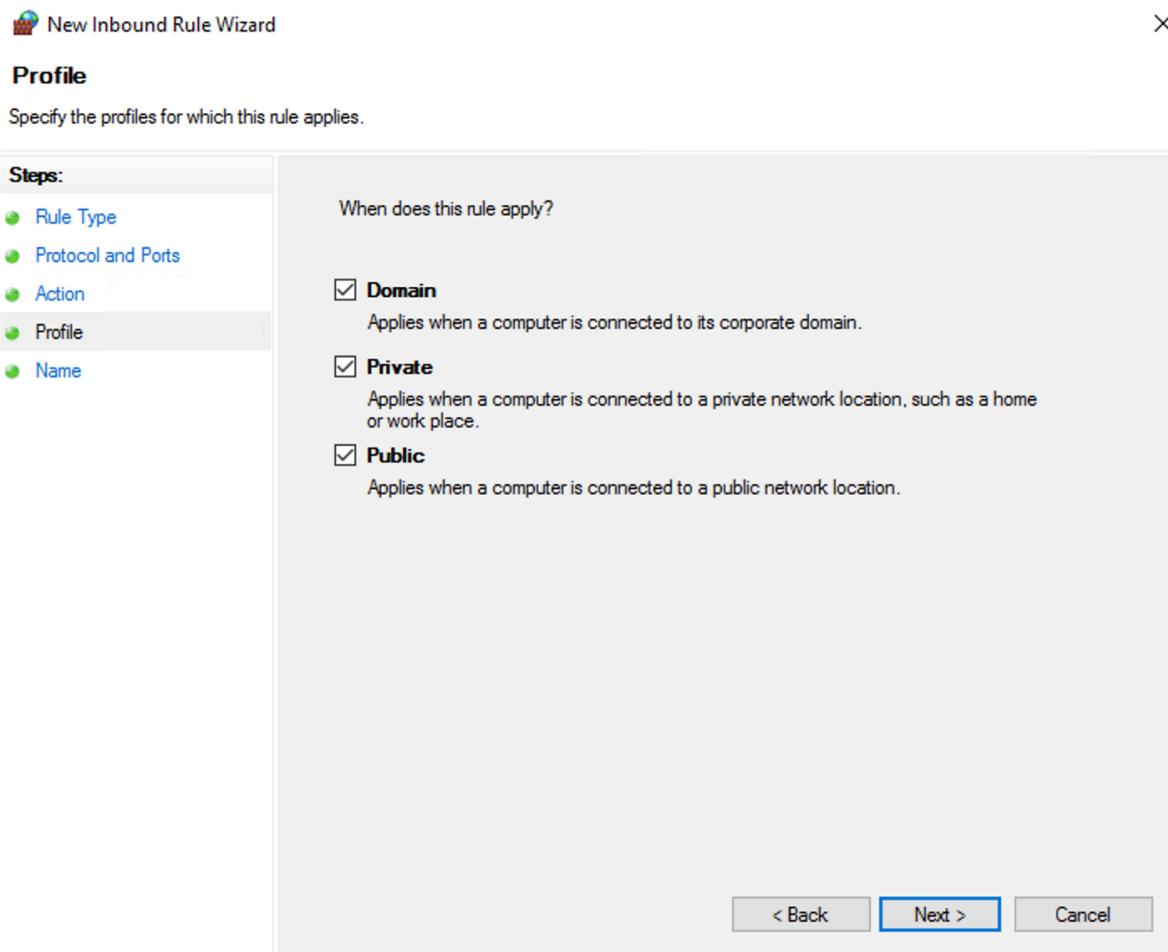
- TCP 선택
- Specific local ports 선택
- Port 번호 22 지정



# 배포서버 구성

## SSH 서버 구성

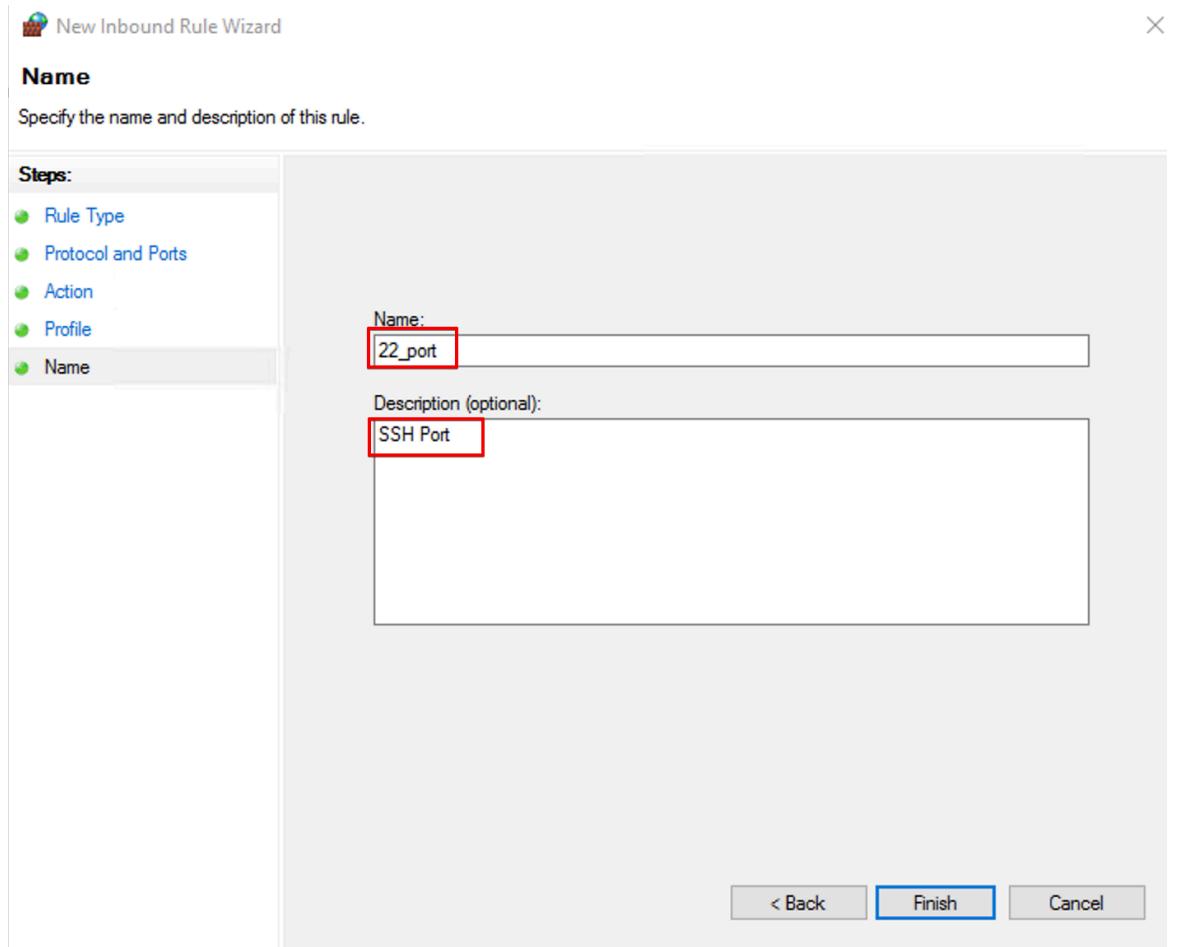
- Domain, Private, Public 선택



# 배포서버 구성

## SSH 서버 구성

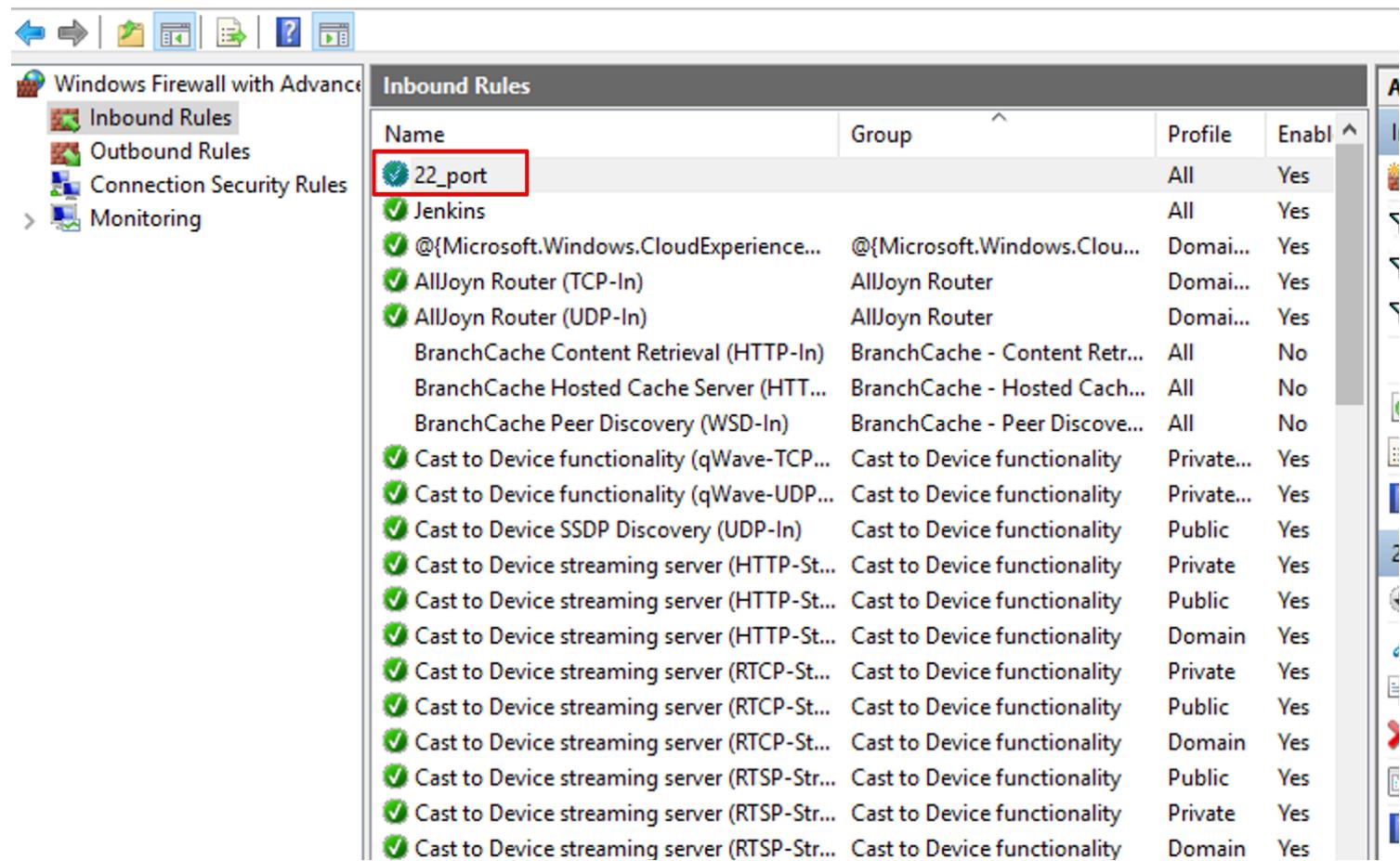
- Name : 22\_port
- Description : SSH Port
- “Finish” 버튼 클릭하여 설정 저장



## 배포서버 구성

# SSH 서버 구성

- 방화벽 설정 추가 확인



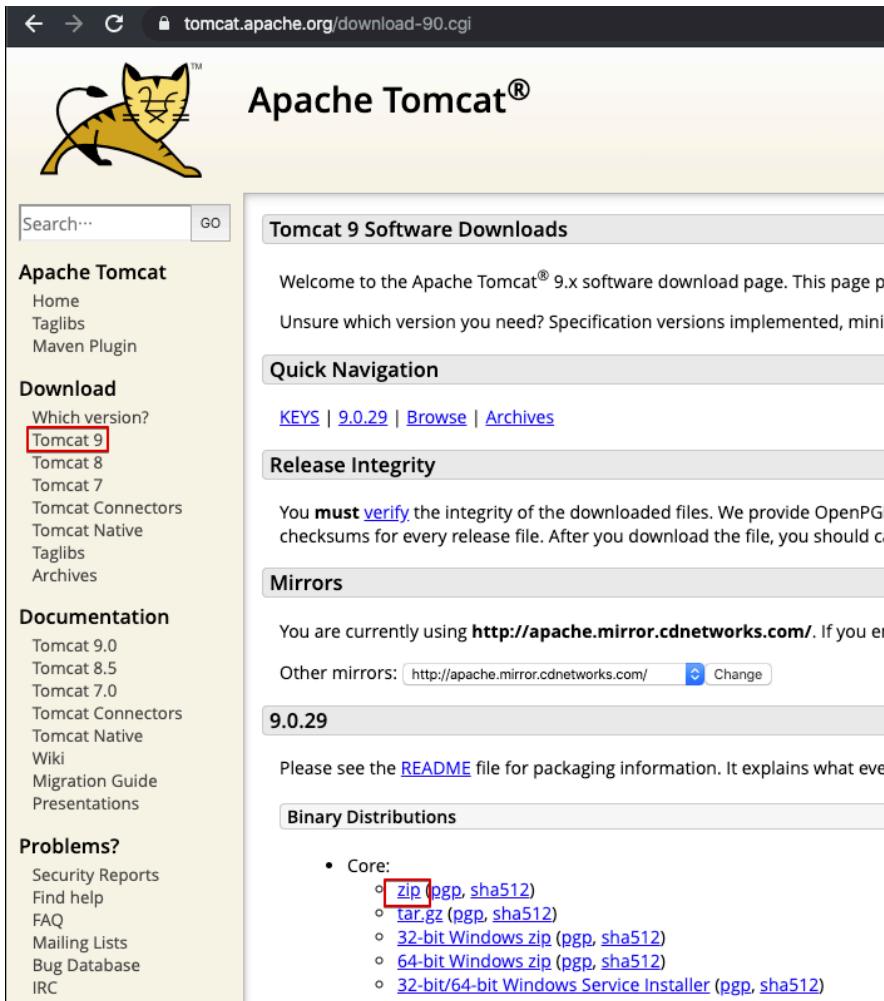
# 송출서버 구성

# 송출서버 구성

## Tomcat 설치

- <https://tomcat.apache.org/> 에 접속
  - “Tomcat 9” 선택
  - Binary Distributions -> Core -> zip 클릭하여 설치파일

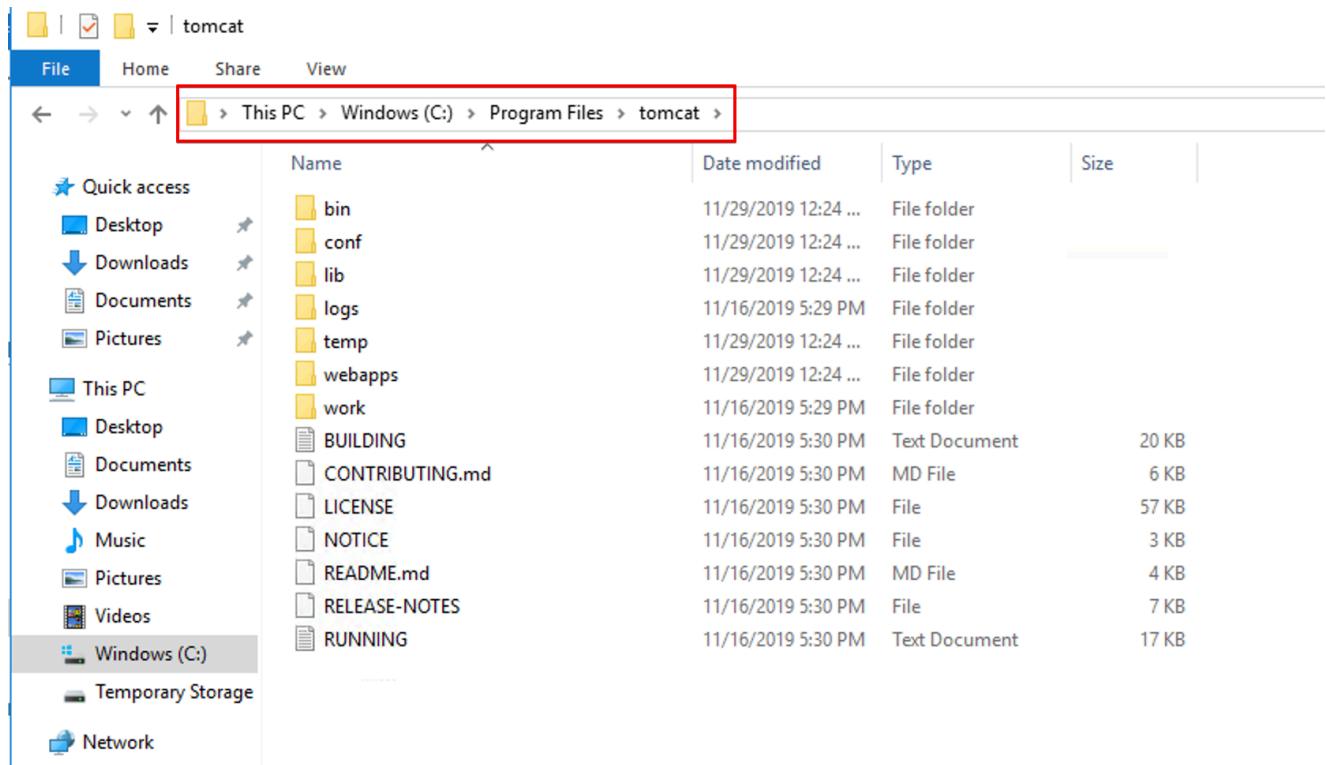
## Download



# 송출서버 구성

## Tomcat 설치

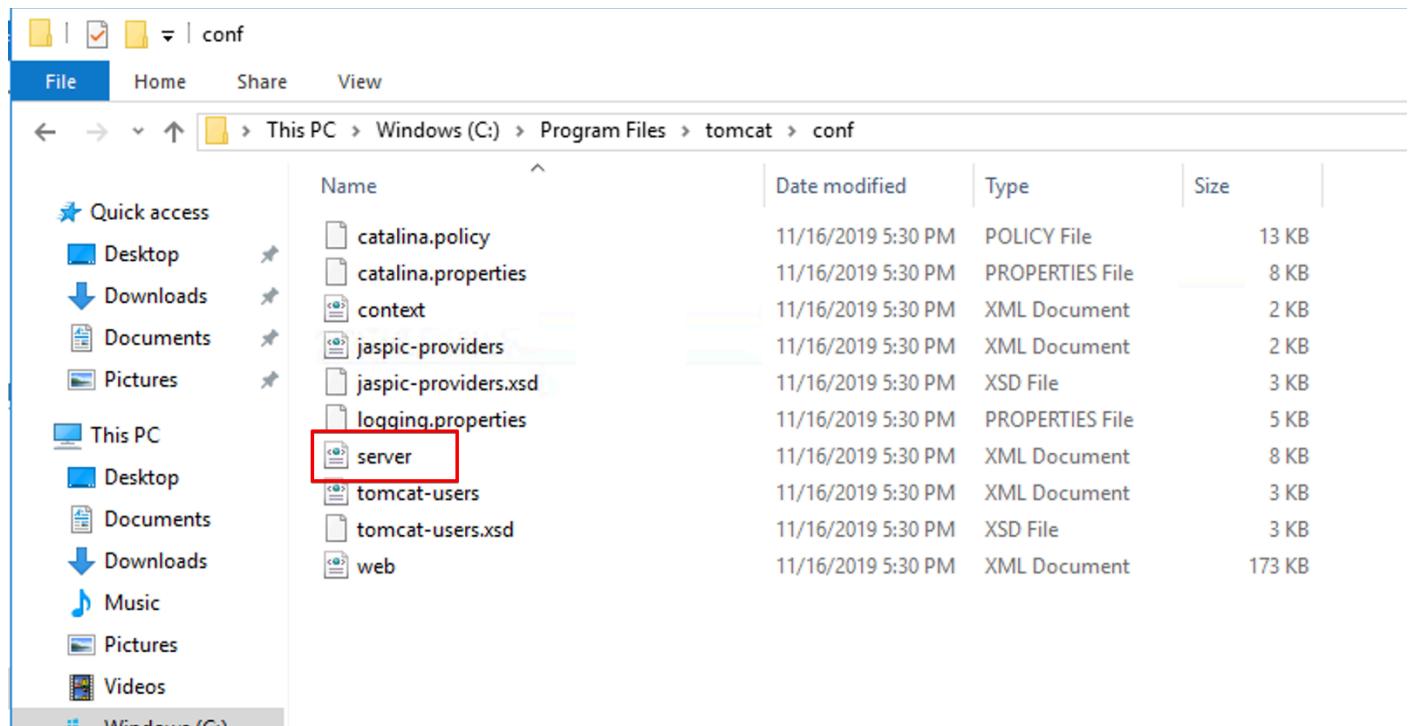
- 설치파일 압축 해제
- 설치파일 디렉토리명 tomcat으로 변경
- Tomcat 디렉토리 C:\Program Files 로 이동



## 송출서버 구성

# Tomcat 설치

- C:\Program Files\Tomcat\conf로 이동
  - Server.xml 파일을 에디터로 열기



# 송출서버 구성

## Tomcat 설치

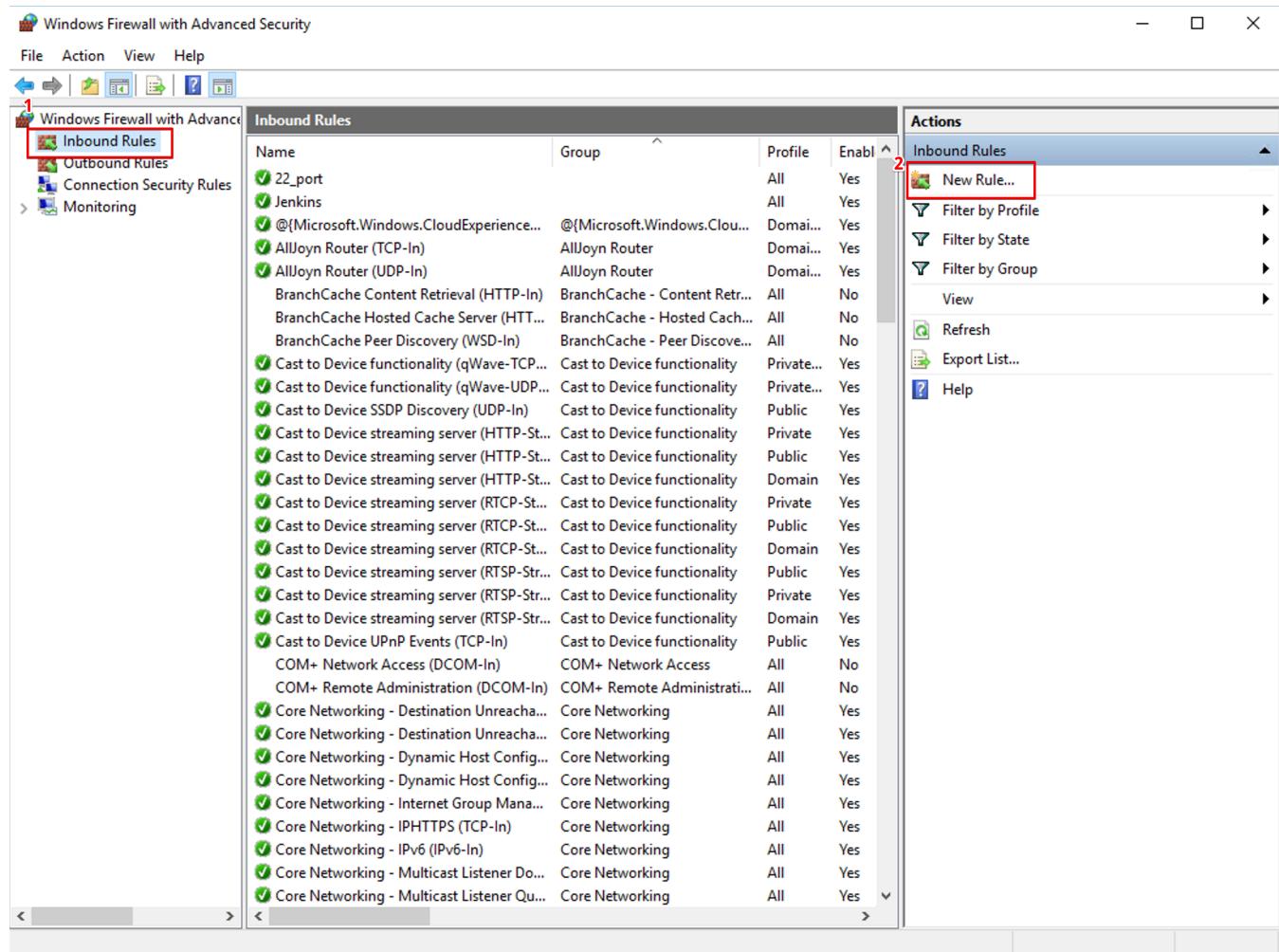
- Tomcat Service Port 8080 -> 80 으로 변경
  - 변경 후 저장

```
-->
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

# 송출서버 구성

## Tomcat 설치

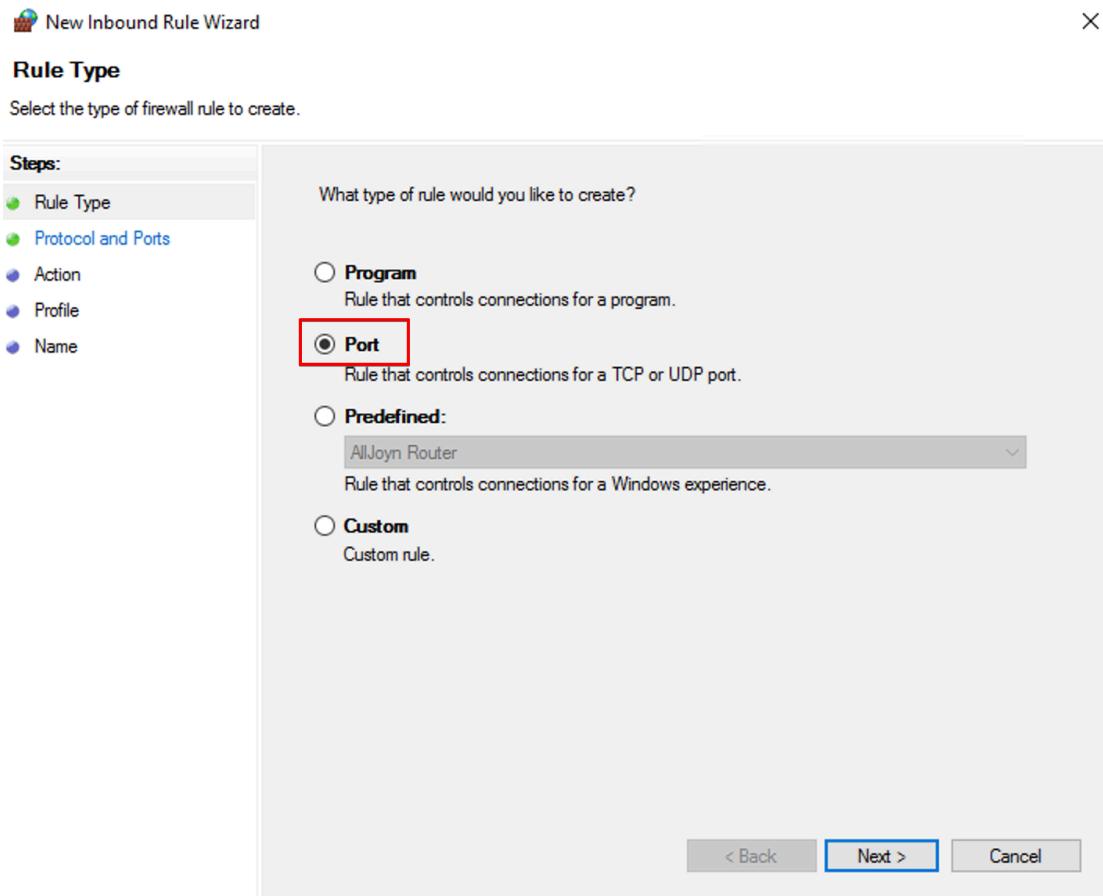
- 서버 시스템 검색에서 "Windows Firewall with Advanced Security" 검색
- Inbound Rules 선택
- New Rules 클릭



# 송출서버 구성

## Tomcat 설치

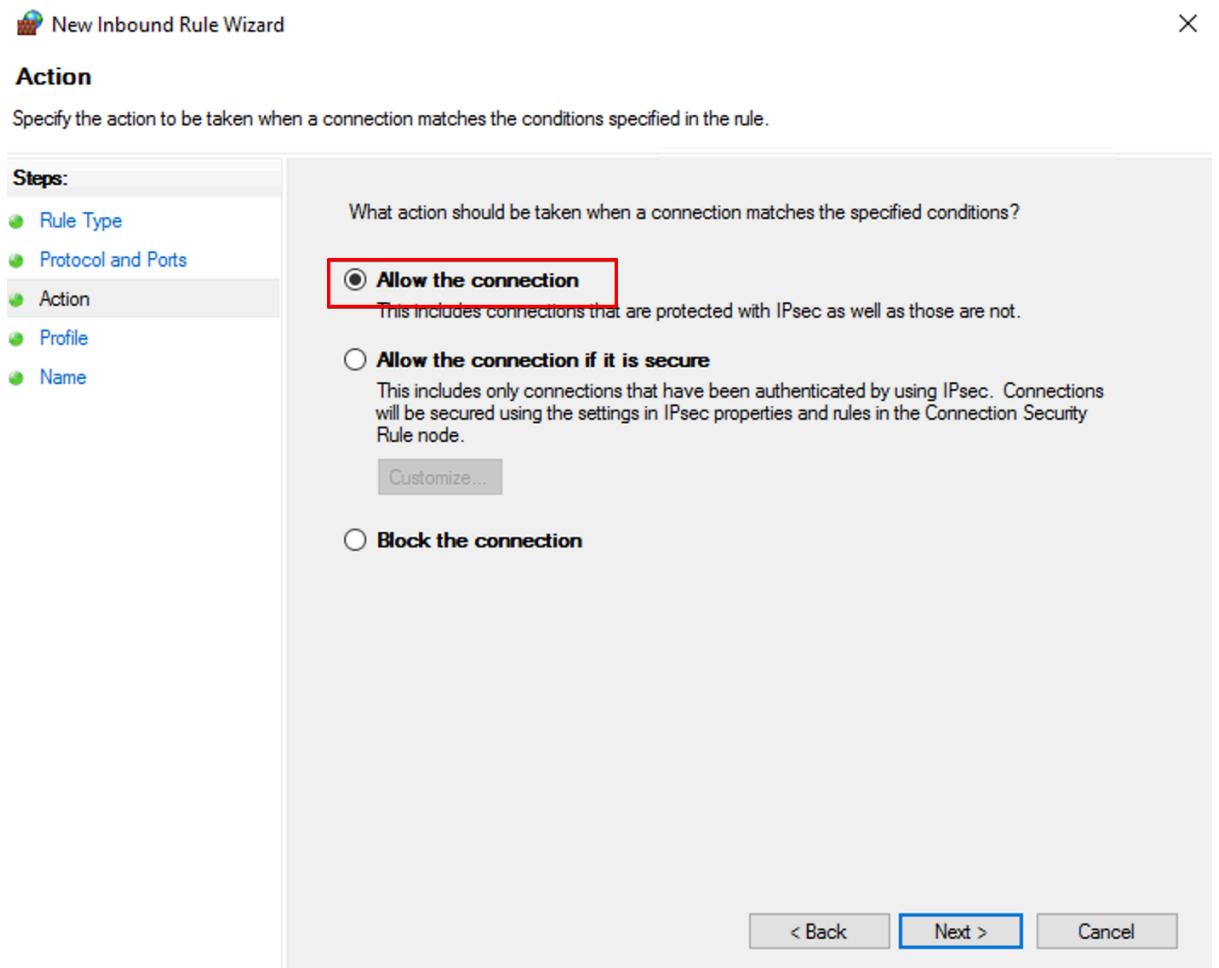
- Rule Type “Port” 선택



# 송출서버 구성

## Tomcat 설치

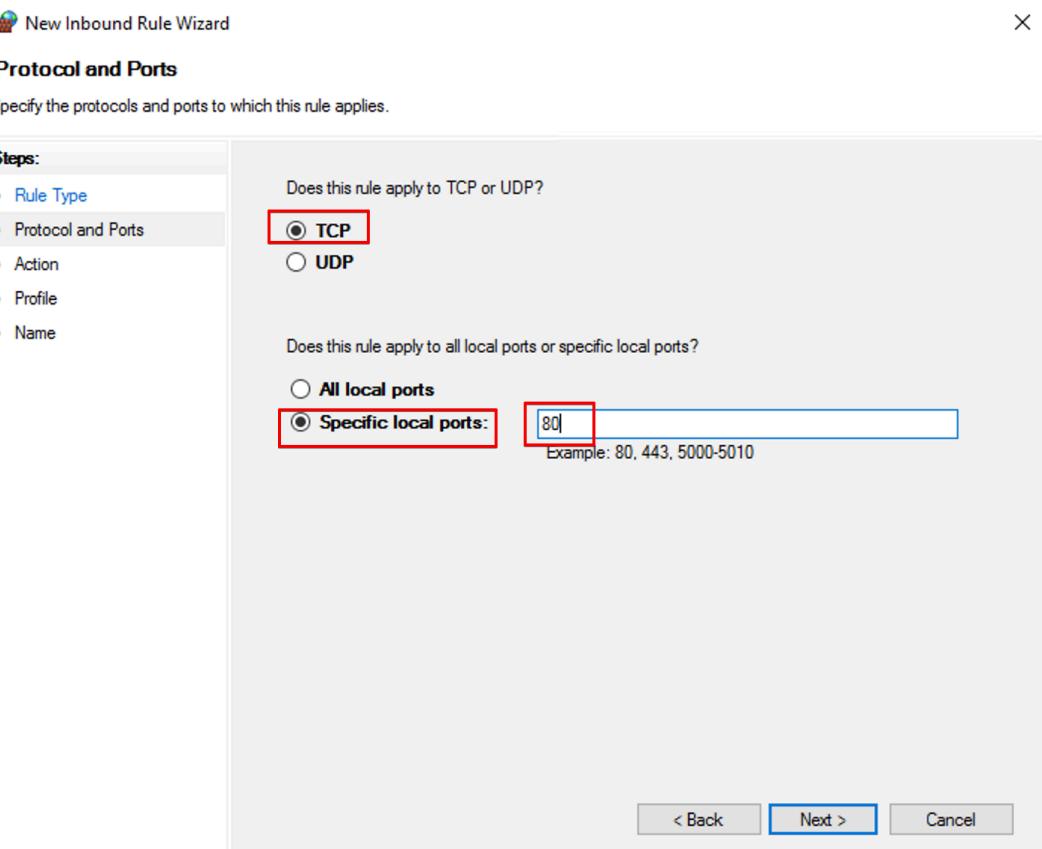
- Allow the connection 선택



# 송출서버 구성

## Tomcat 설치

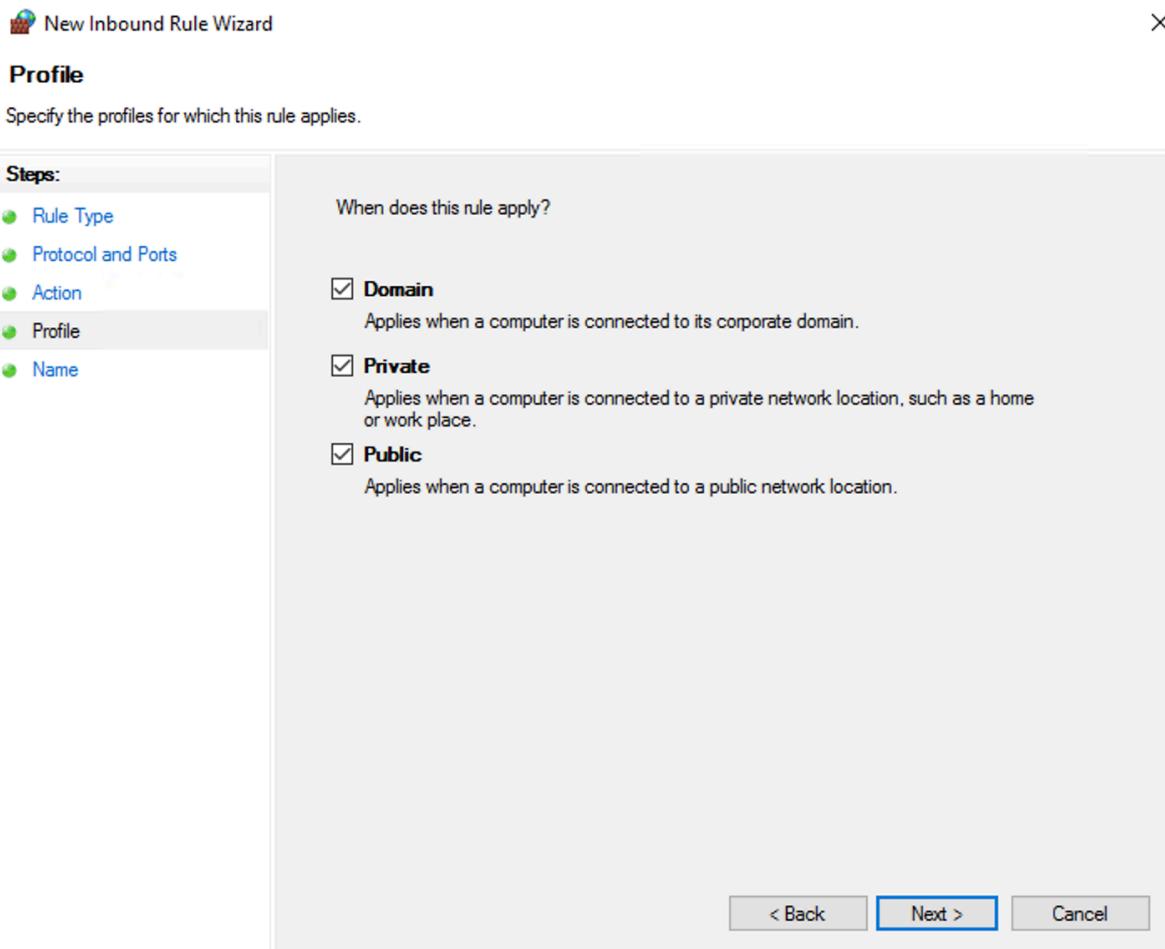
- TCP 선택
- Specific local ports 선택
- Port 번호 80 지정



# 송출서버 구성

## Tomcat 설치

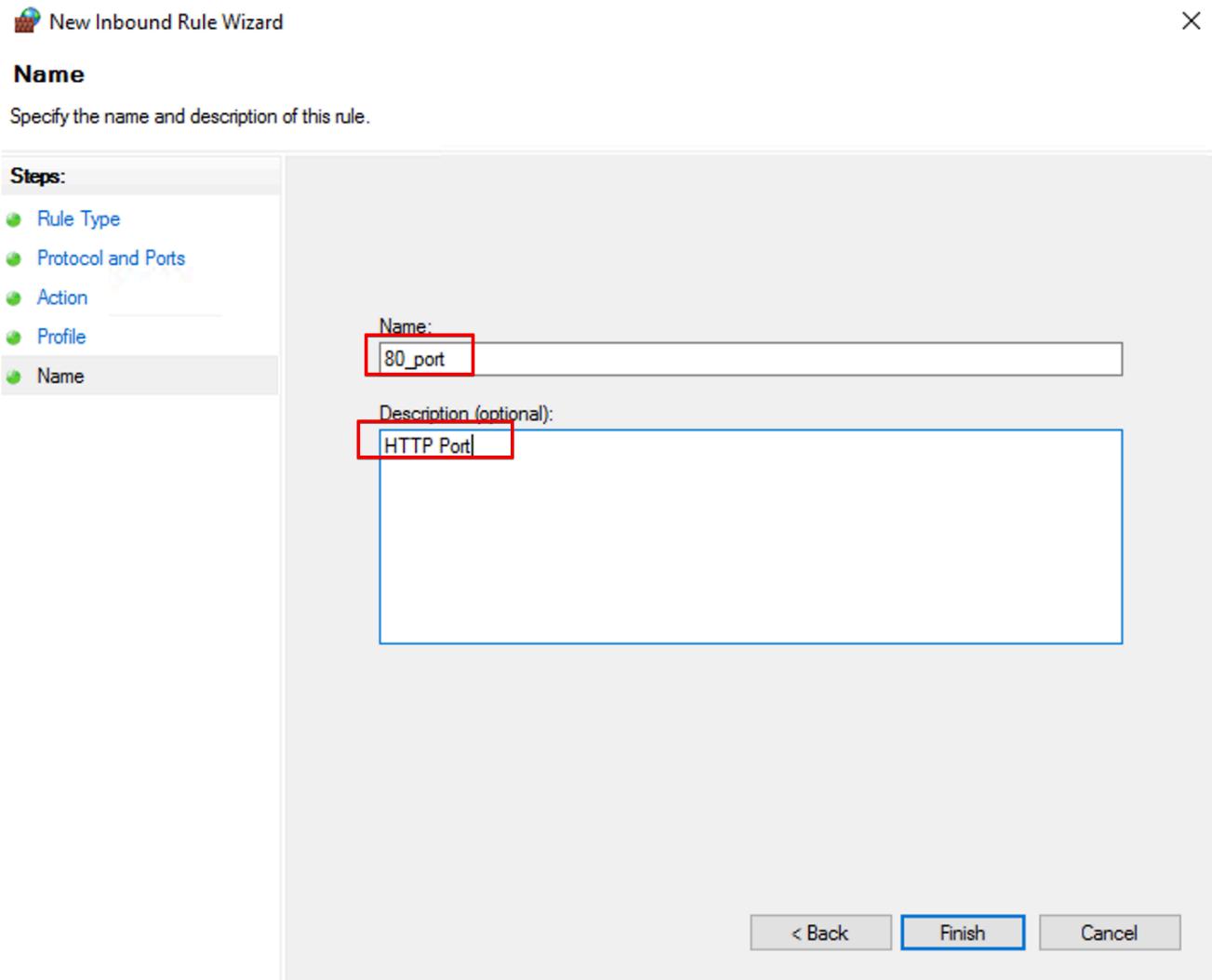
- Domain, Private, Public 선택



# 송출서버 구성

## Tomcat 설치

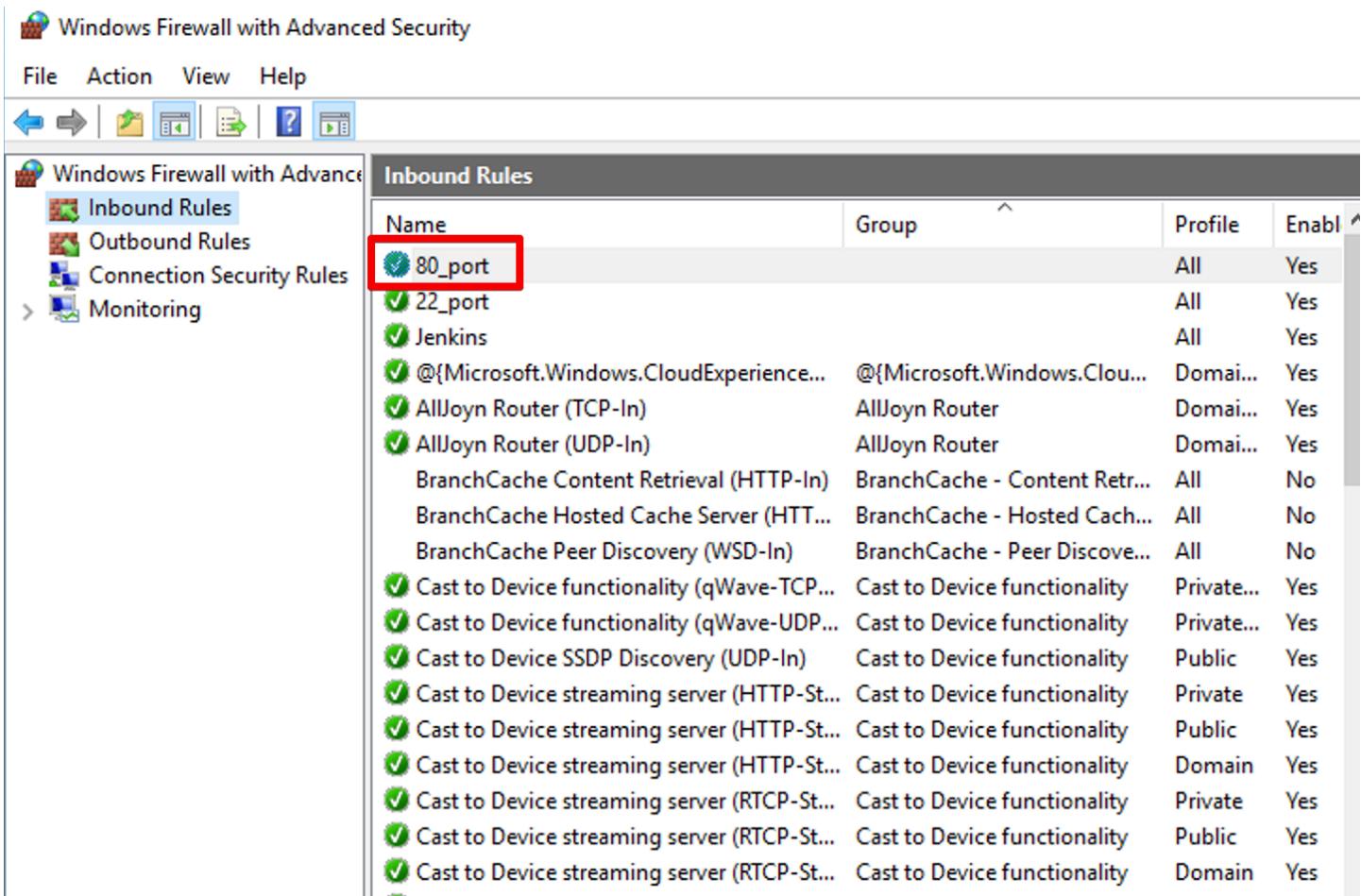
- Name : 80\_port
- Description : HTTP Port
- "Finish" 버튼 클릭하여 설정 저장



# 송출서버 구성

## Tomcat 설치

- 방화벽 설정 추가 확인



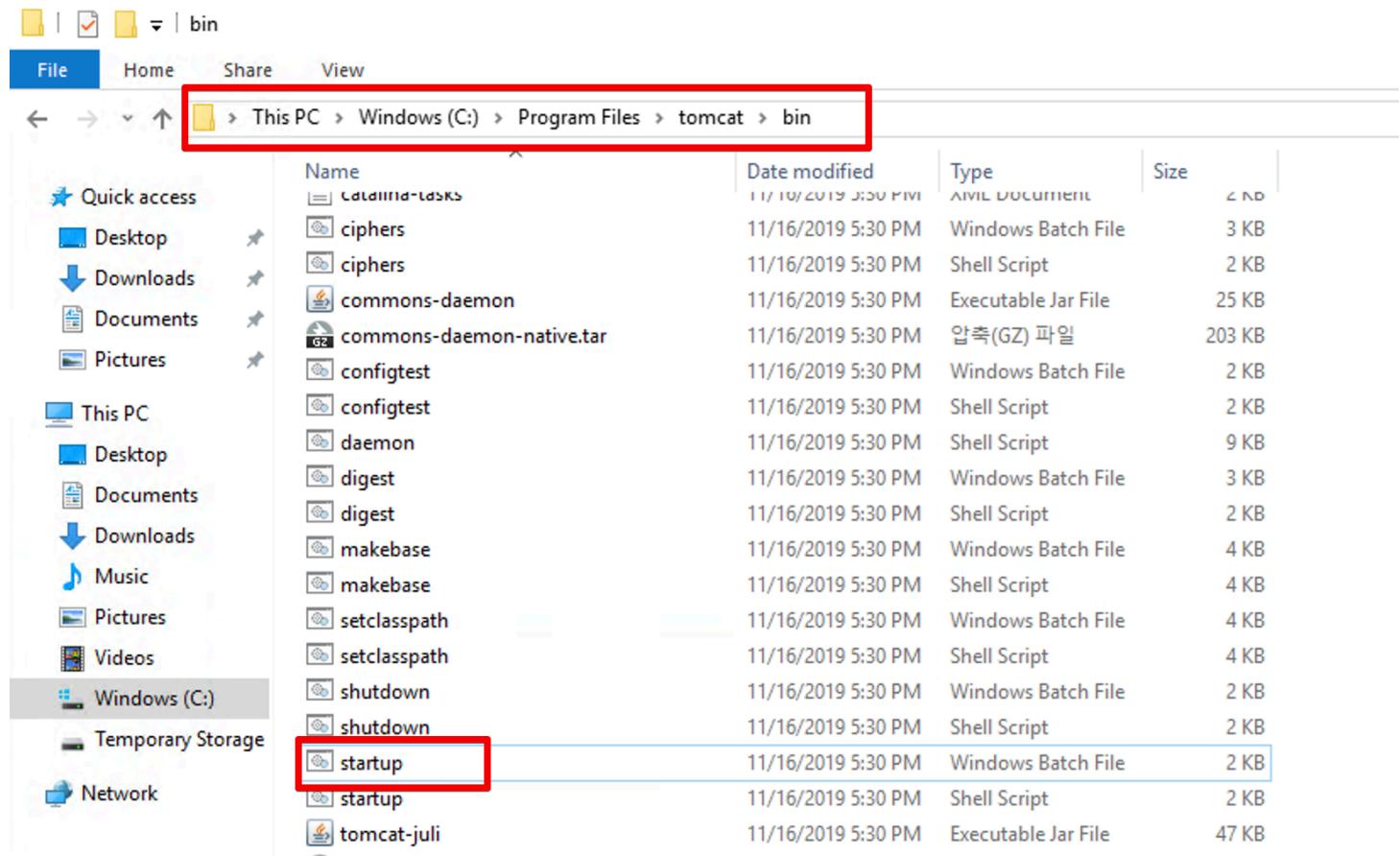
# 송출서버 구성

## Tomcat 설치

- C:\Program Files\Tomcat\bin

디렉토리로 이동

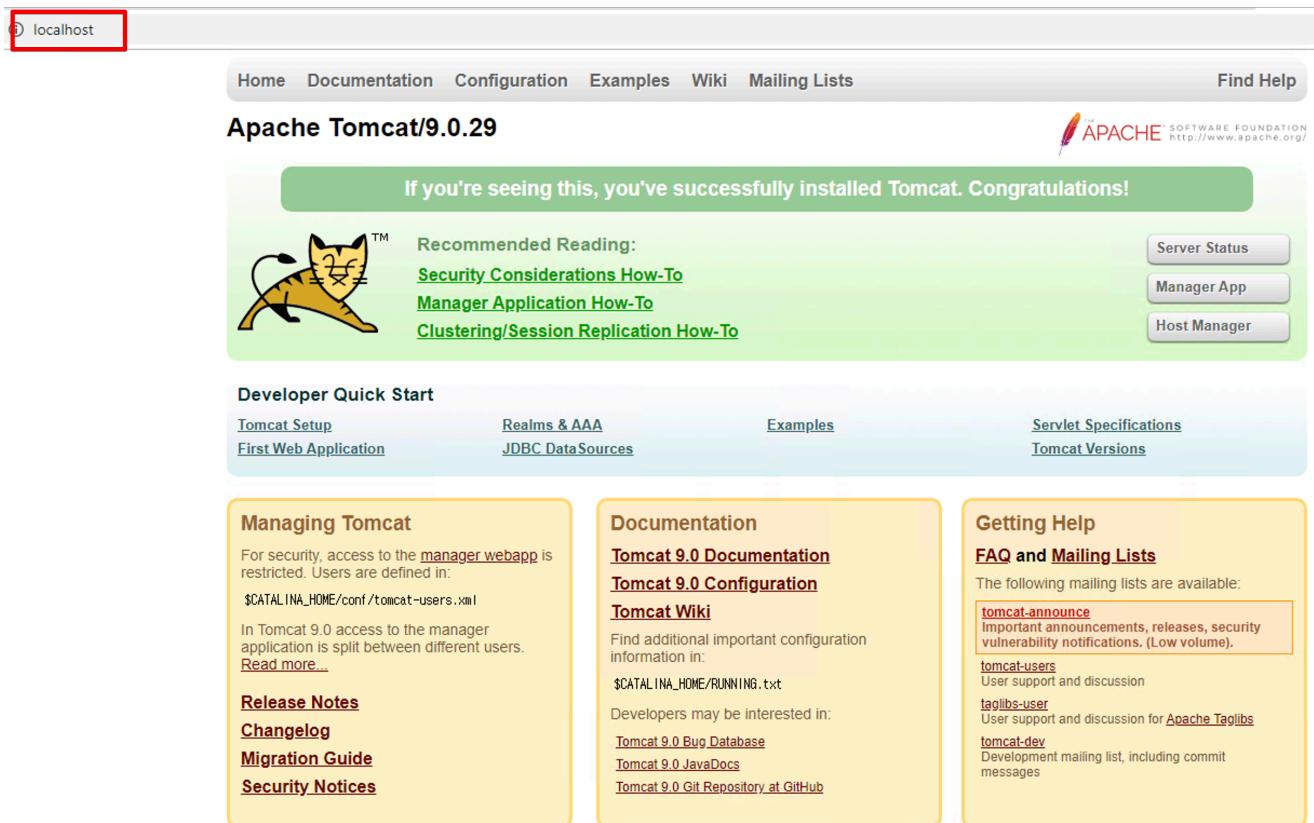
- Tomcat 실행



# 송출서버 구성

## Tomcat 설치

- 브라우저에서 localhost 입력하여 웹페이지 정상적으로 뜨는지 확인

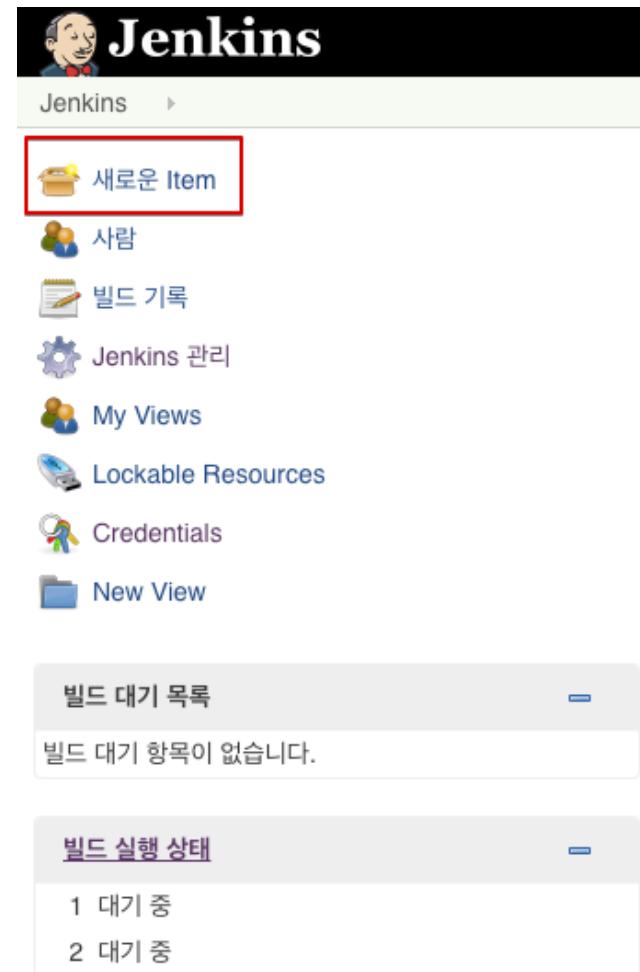


# Jenkins 프로젝트 구성

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 생성

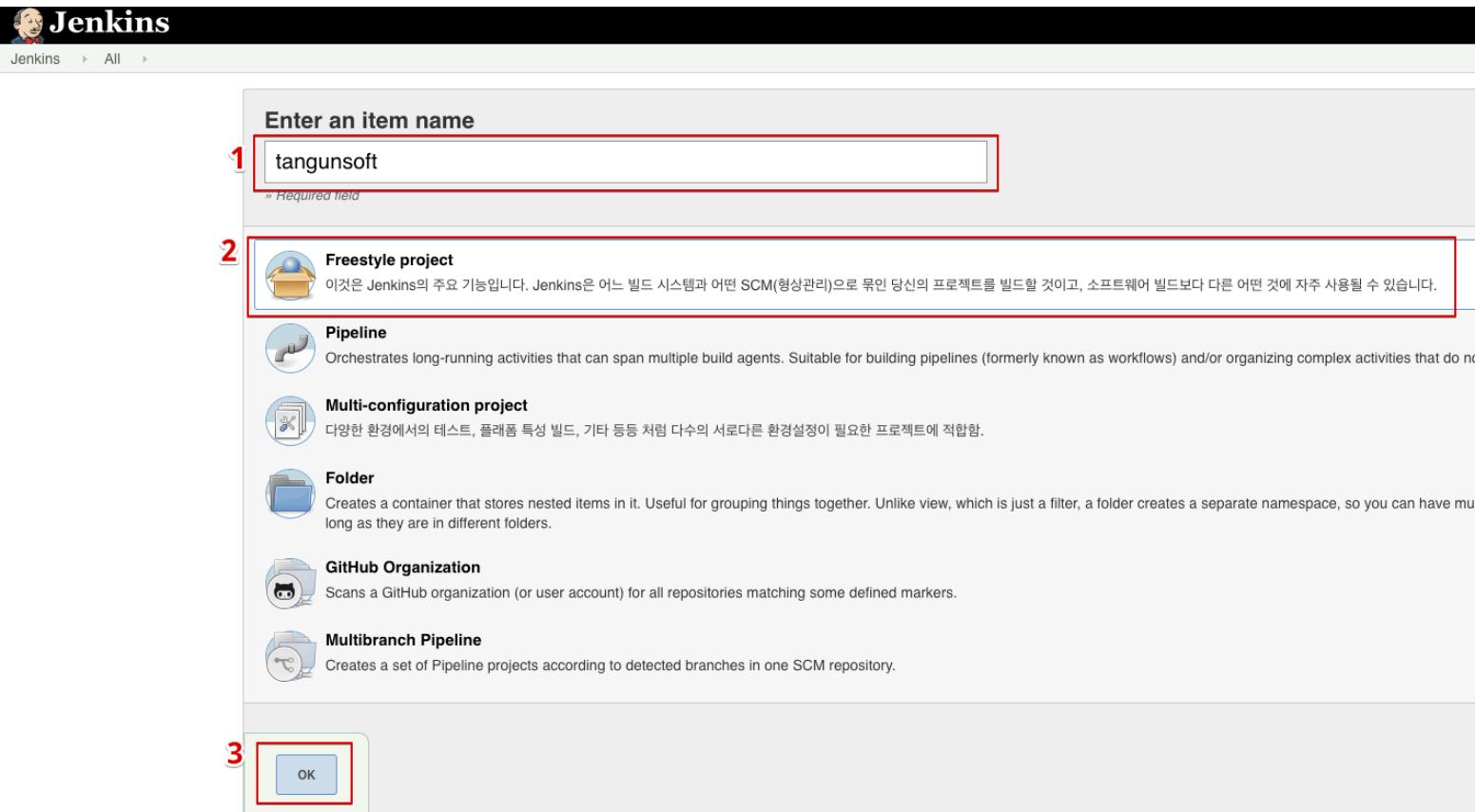
- Jenkins 신규 프로젝트 생성을 위하여 메뉴에서 “새로운 Item” 클릭



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 설정

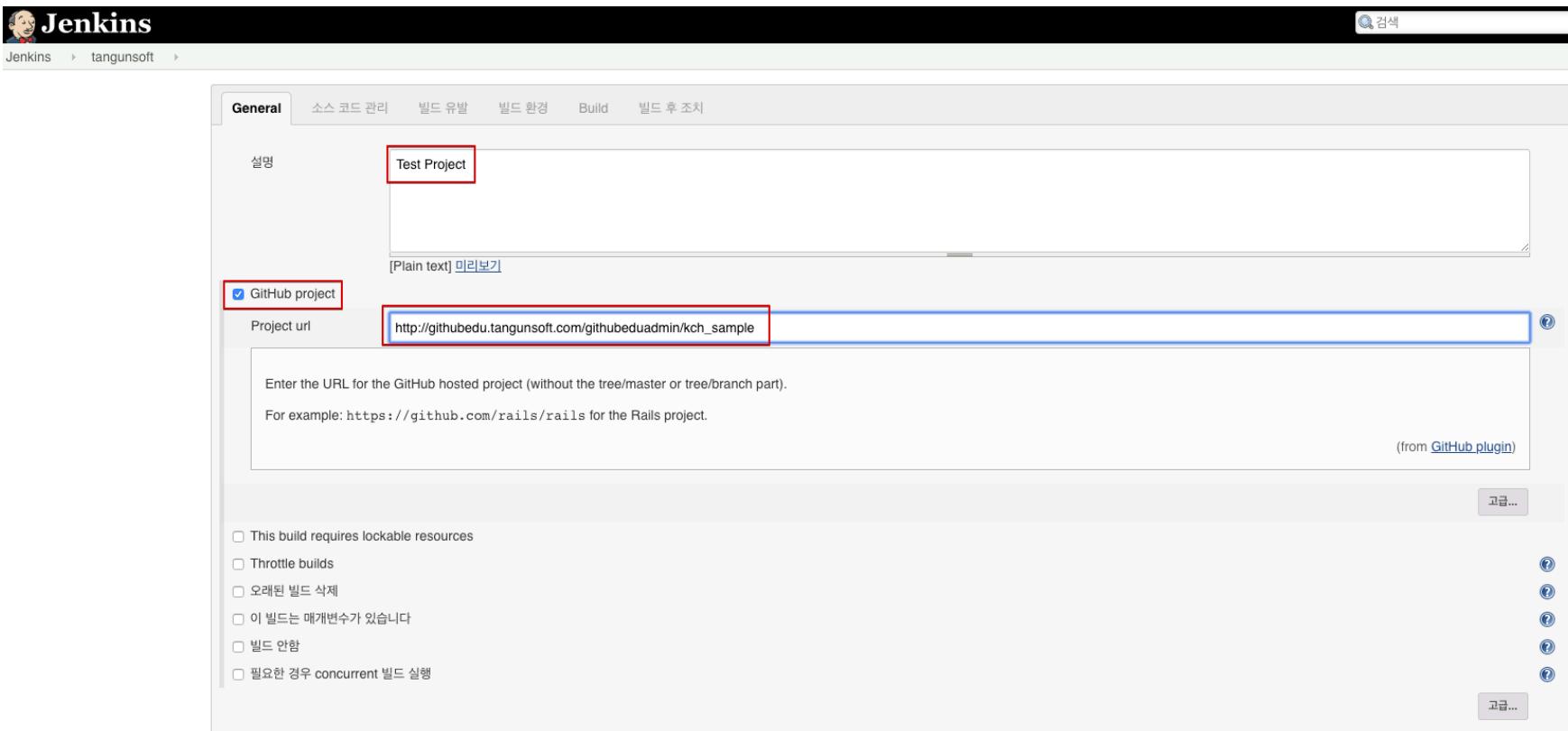
- Jenkins 프로젝트명을 입력
- 프로젝트 타입을 Freestyle Project로 선택
- OK 버튼 클릭



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 설정

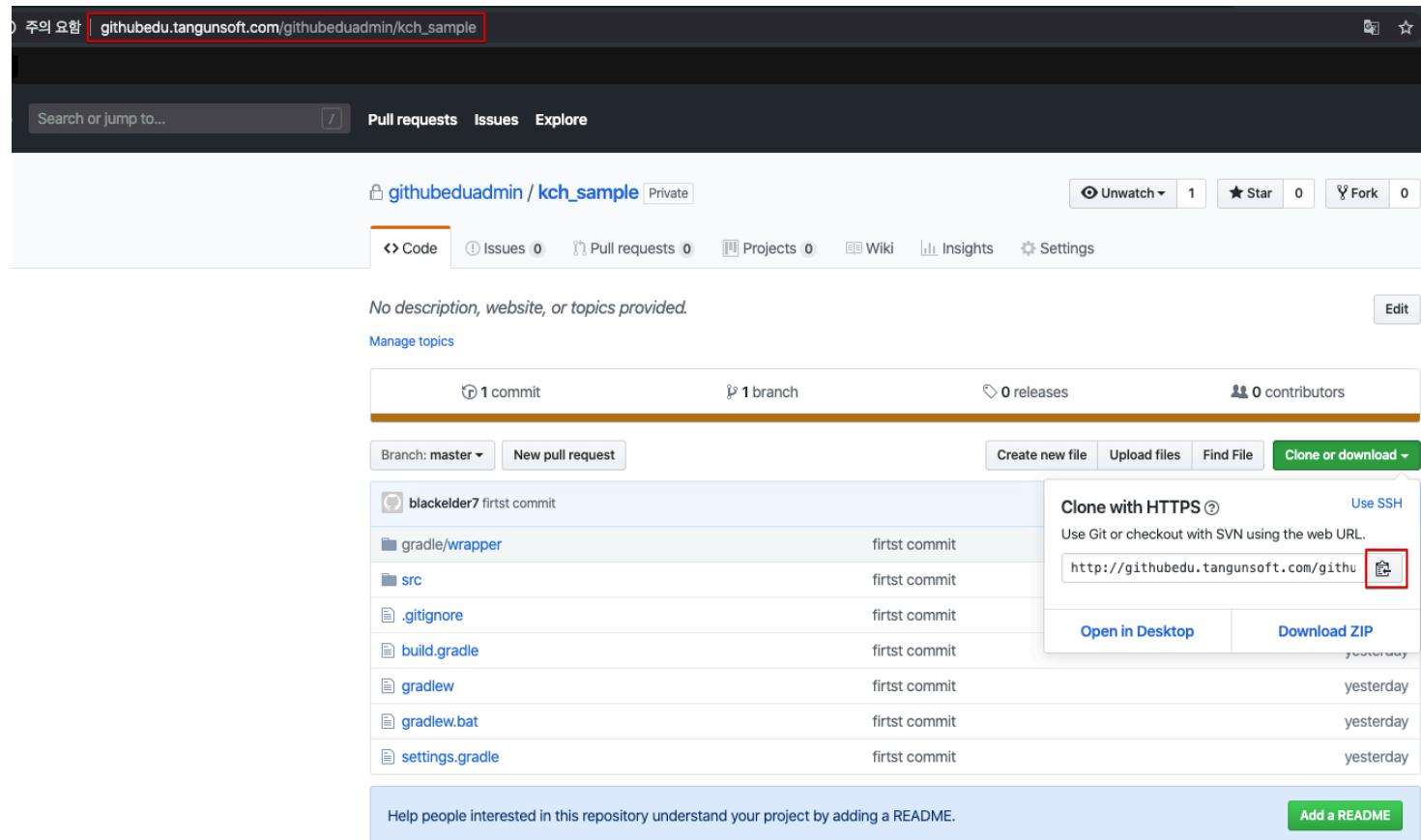
- 프로젝트의 General 탭에서 설명 입력
- GitHub Project 체크
- Github Project Url 입력



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- GitHub Enterprise의 Sample 프로젝트에 접근
- Clone or download 버튼 클릭
- Url 클립보드 버튼 클릭



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- GitHub 계정정보 입력
- 입력후 “Add” 버튼 클릭

 Jenkins Credentials Provider: Jenkins

 Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: githubbeduadmin

Password: .....

ID: githubbeduadmin

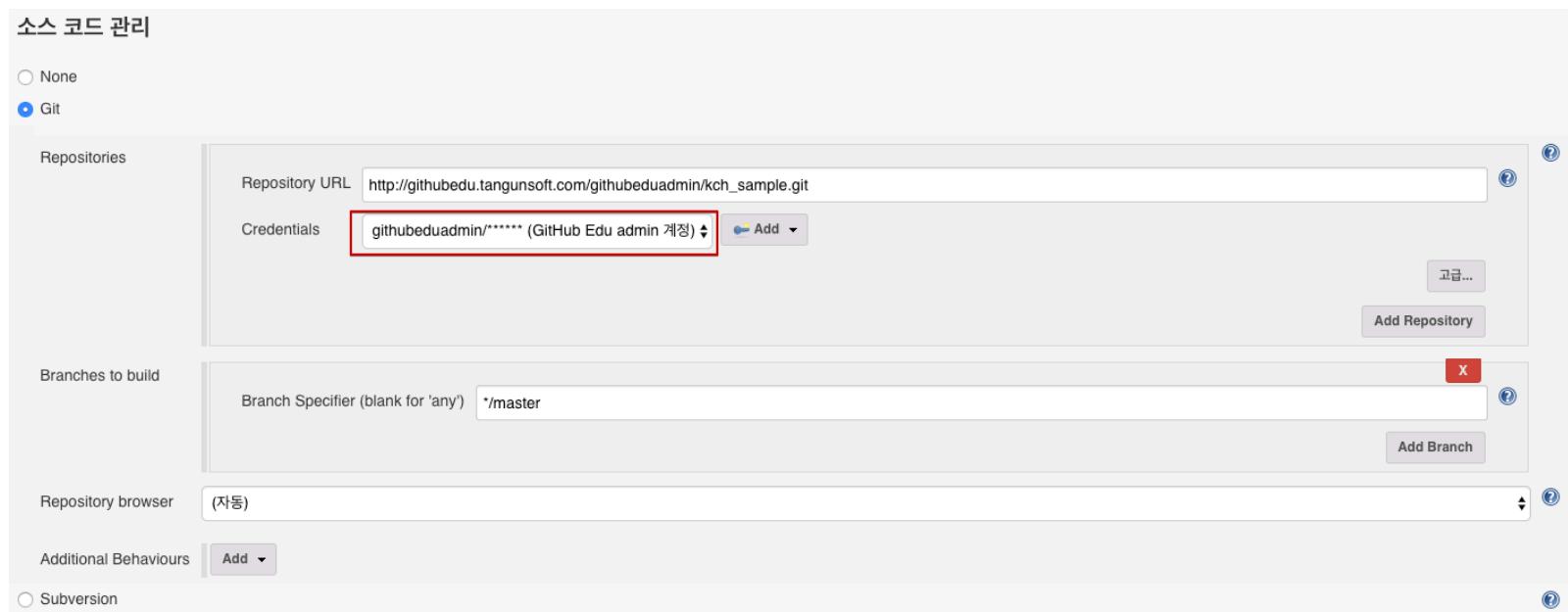
Description: GitHub Edu admin 계정

Add Cancel

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- Credentials에서 생성한 GitHub 계정 항목 선택
- 선택 후 Jenkins 프로젝트 저장



## Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

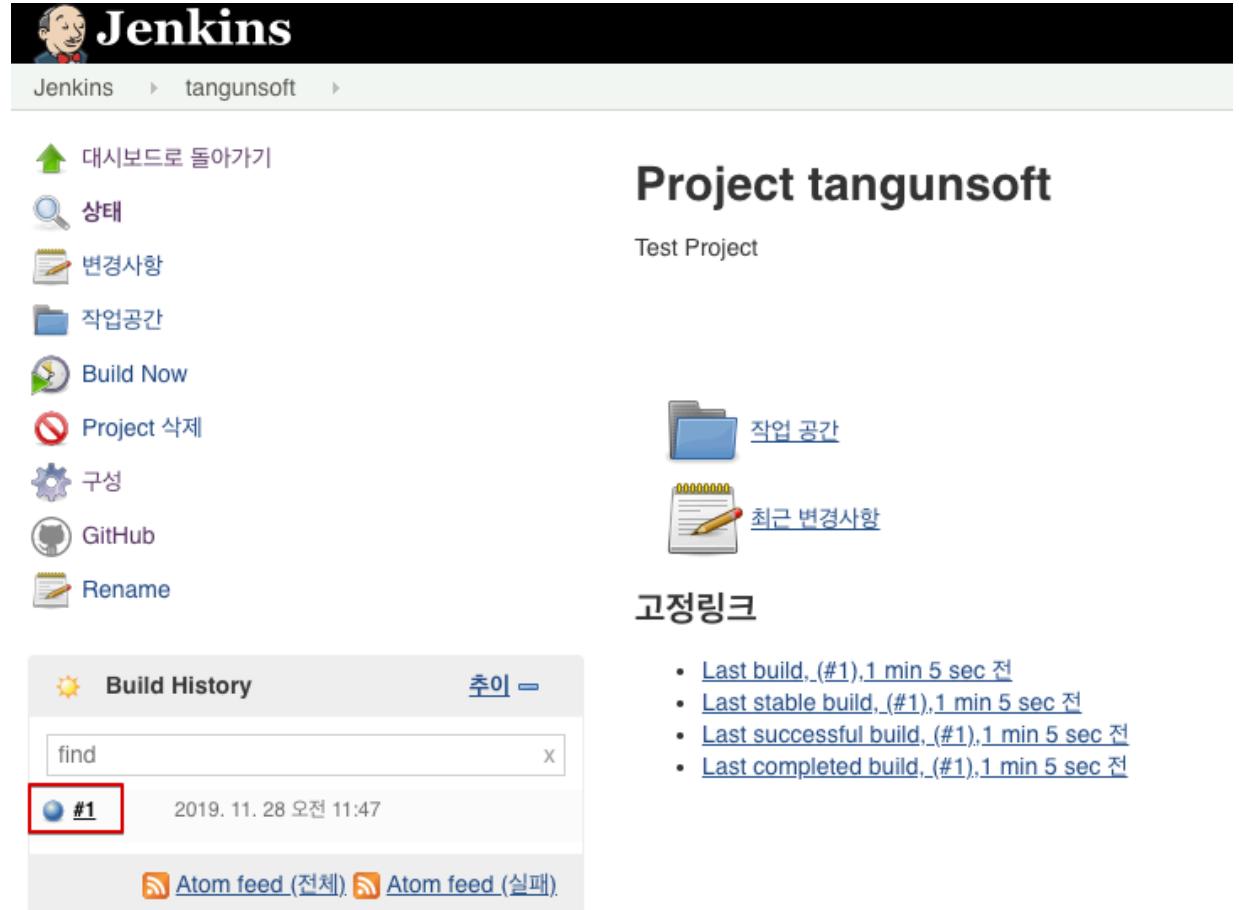
- 프로젝트 메뉴에서 Build Now 실행

The screenshot shows the Jenkins interface for the 'tangunsoft' project. The top navigation bar includes the Jenkins logo and the project name 'Jenkins > tangunsoft'. On the left, there's a sidebar with various icons and labels: '대시보드로 돌아가기' (Dashboard), '상태' (Status), '변경사항' (Changes), '작업공간' (Workspace), 'Build Now' (highlighted with a red box), 'Project 삭제' (Delete Project), '구성' (Configure), 'GitHub', and 'Rename'. Below this is a 'Build History' section with a search bar labeled 'find' and two RSS feed links: 'Atom feed (전체)' and 'Atom feed (실패)'. The main content area displays the project name 'Project tangunsoft' and its status as 'Test Project'. It also features a 'workspace' icon and a '최근 변경사항' (Recent Changes) icon. At the bottom right, there's a link labeled '고정링크' (Pin Link).

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- 텔드 번호 확인
  - 텔드 번호 클릭



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- 빌드 로그 확인을 위해 Console Output 클릭

 Jenkins

Jenkins > tangunsoft > #1

 [프로젝트로 돌아가기](#)

 [상태](#)

 [바뀐점](#)

 [Console Output](#)

 [빌드 정보 수정](#)

 [Delete build '#1'](#)

 [Git Build Data](#)

 [No Tags](#)

 **빌드 #1 (2019. 11. 28 오전 11:47:29)**

 No changes.

 사용자 [eric.kim](#)에 의해 시작됨

 **Revision:** 0ca094c5a98f675dea461f3bf304ec0c74e1838e

- refs/remotes/origin/master

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- 빌드 로그 확인

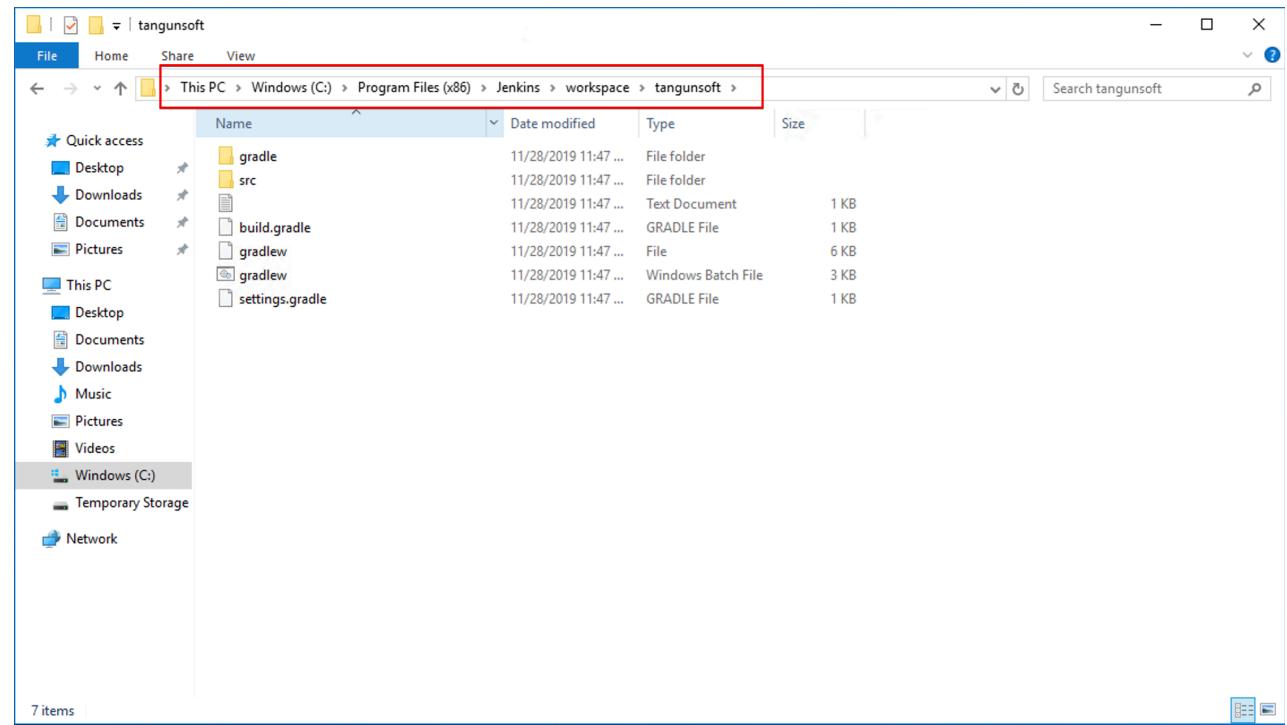
The screenshot shows the Jenkins interface with a successful build log. The log details the cloning of a GitHub repository, configuration of Git credentials, and a successful commit message. A red box highlights the URL of the cloned repository and the final 'Finished: SUCCESS' message.

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 – GHE 연동

- C:\Program Files (x86)\Jenkins\workspace\tangunsoft

경로에 정상적으로 GitHub 소스 내려 받았는지 확인



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 빌드 구성

- 프로젝트 메뉴로 돌아가 “구성” 클릭



# Jenkins

Jenkins > tangunsoft >

 대시보드로 돌아가기

 상태

 변경사항

 작업공간

 Build Now

 Project 삭제

 **구성**

 GitHub

 Rename

**Project tangunsoft**

Test Project

 **작업 공간**

 [최근 변경사항](#)

**고정링크**

- [Last build. \(#1\), 18 min 전](#)
- [Last stable build. \(#1\), 18 min 전](#)
- [Last successful build. \(#1\), 18 min 전](#)
- [Last completed build. \(#1\), 18 min 전](#)

**Build History** [초이 -](#)

find X

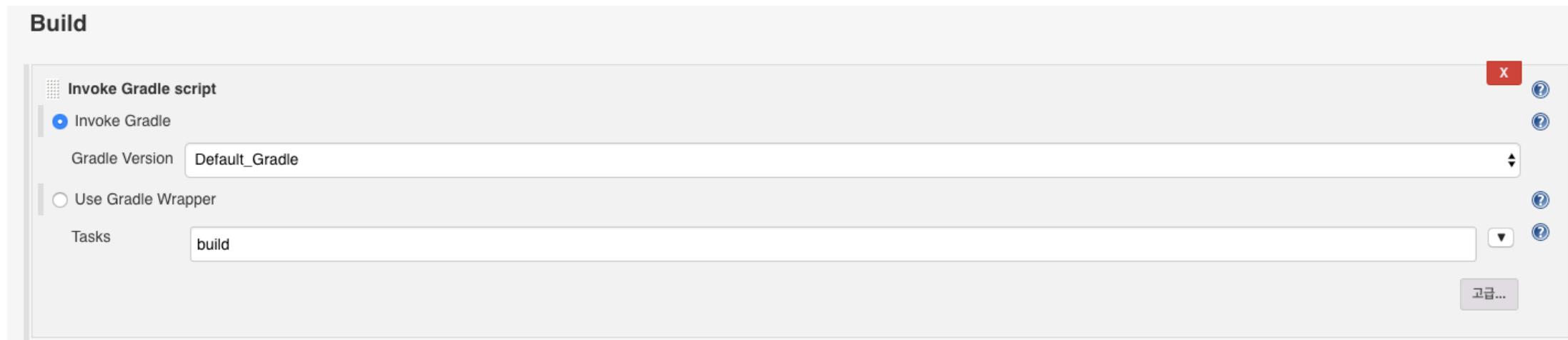
#1 2019. 11. 28 오전 11:47

 [Atom feed \(전체\)](#)  [Atom feed \(실패\)](#)

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 빌드 구성

- Build 항목에서 Invoke Gradle script 선택
  - Invoke Gradle 라디오버튼 클릭
  - Gradle Version : Default\_Gradle 선택
  - Tasks : build 입력
  - 프로젝트 설정 저장



## Jenkins 프로젝트 구성

## Jenkins 프로젝트 빌드 구성

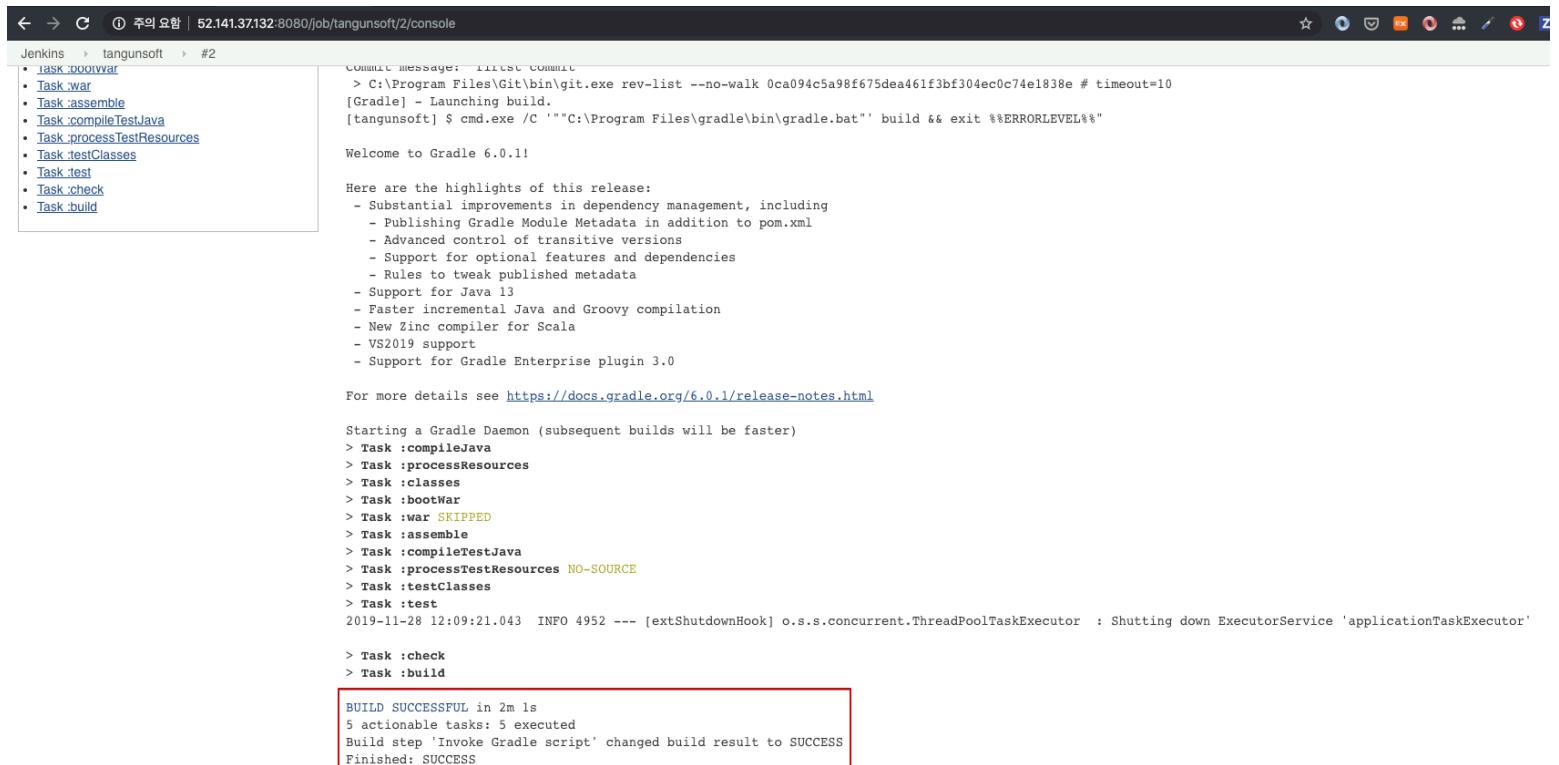
- #### • 프로젝트 빌드 실행

The screenshot shows the Jenkins interface for the project 'tangunsoft'. The top navigation bar includes the Jenkins logo and the project name. On the left, there's a sidebar with various options: '대시보드로 돌아가기', '상태', '변경사항', '작업공간', 'Build Now' (which is highlighted with a red box), 'Project 삭제', '구성', 'GitHub', and 'Rename'. Below this is a 'Build History' section with a search bar labeled 'find' and a single build entry for '#1' from November 28, 2019, at 11:47. At the bottom, there are links for 'Atom feed (전체)' and 'Atom feed (실패)'. To the right of the sidebar, the main content area displays the project name 'Project tangunsoft' and its description 'Test Project'. It also features two links: '작업 공간' and '최근 변경사항', each accompanied by an icon.

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 빌드 구성

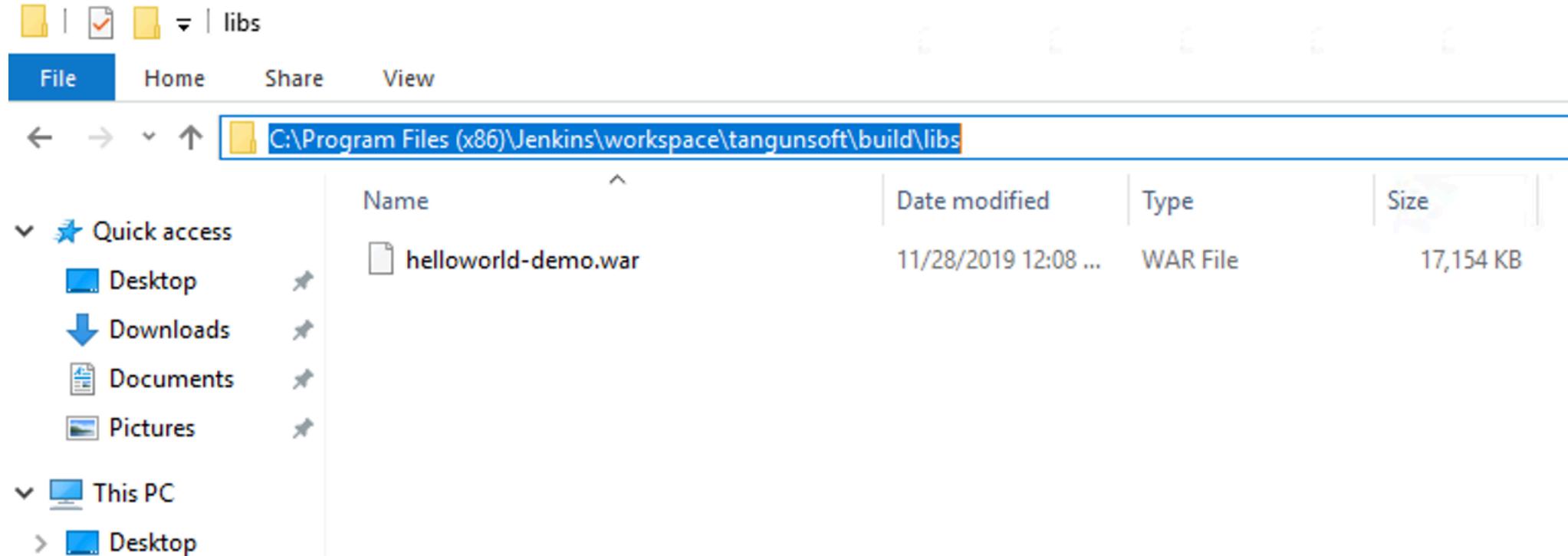
- 해당 빌드의 콘솔 로그 확인
  - 빌드 내용 확인



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 빌드 구성

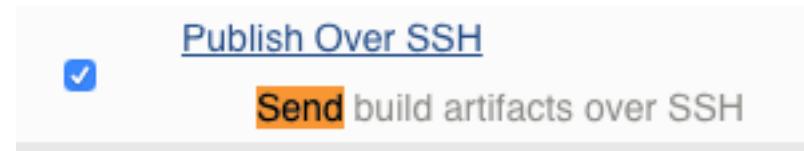
- War 파일이 정상적으로 생성되었는지 확인



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

- 젠킨스 관리 -> 플러그인 관리 -> 설치가능 탭에서 Publish Over SSH 플러그인 선택하여 설치
  - 플러그인 설치 후 Jenkins 재시작



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

- Jenkins 관리 -> 시스템 설정으로 이동
  - Publish over SSH 항목에서 빌드된 파일 전송할 서버 정보 입력
  - “Test Configuration” 클릭하여 Connection 정상인지 확인
  - 설정 저장

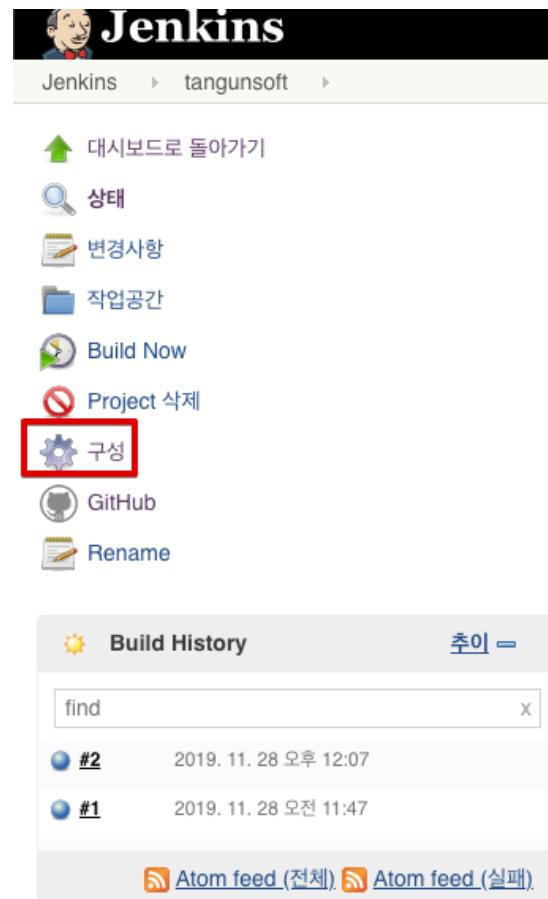
**Publish over SSH**

Jenkins SSH Key	<input type="text"/>	(?)															
Passphrase	<input type="password"/> .....  .....	(?)															
Path to key	<input type="text"/>	(?)															
Key	<input type="text"/>	(?)															
Disable exec	<input type="checkbox"/>	(?)															
SSH Servers	<table><tr><td><input type="checkbox"/> SSH Server</td><td><input type="text"/> Test Web Server  .....</td><td>(?)</td></tr><tr><td>Name</td><td><input type="text"/> Test Web Server  .....</td><td>(?)</td></tr><tr><td>Hostname</td><td><input type="text"/> 10.0.0.5  .....</td><td>(?)</td></tr><tr><td>Username</td><td><input type="text"/> tangunsoft  .....</td><td>(?)</td></tr><tr><td>Remote Directory</td><td><input type="text"/></td><td>(?)</td></tr></table>	<input type="checkbox"/> SSH Server	<input type="text"/> Test Web Server  .....	(?)	Name	<input type="text"/> Test Web Server  .....	(?)	Hostname	<input type="text"/> 10.0.0.5  .....	(?)	Username	<input type="text"/> tangunsoft  .....	(?)	Remote Directory	<input type="text"/>	(?)	(?)
<input type="checkbox"/> SSH Server	<input type="text"/> Test Web Server  .....	(?)															
Name	<input type="text"/> Test Web Server  .....	(?)															
Hostname	<input type="text"/> 10.0.0.5  .....	(?)															
Username	<input type="text"/> tangunsoft  .....	(?)															
Remote Directory	<input type="text"/>	(?)															

## Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

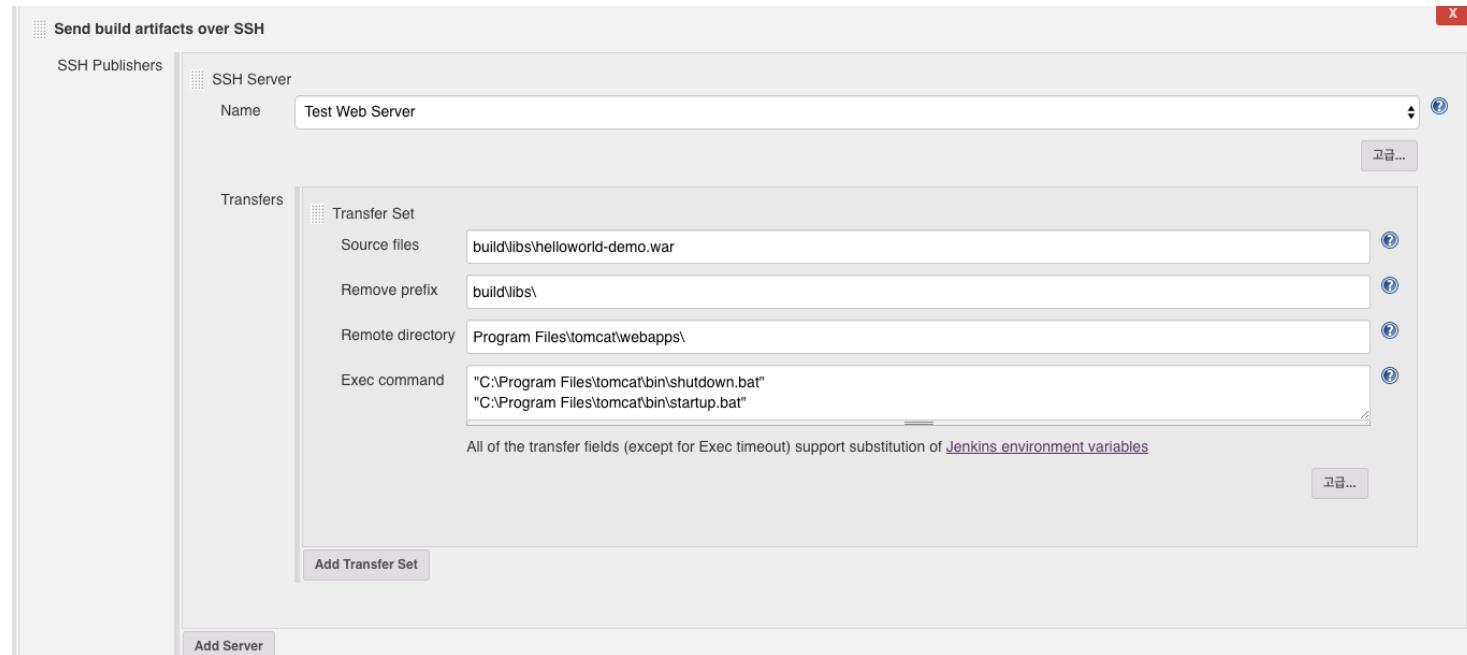
- 프로젝트 페이지로 이동하여 프로젝트 메뉴 중  
“구성” 항목 클릭



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

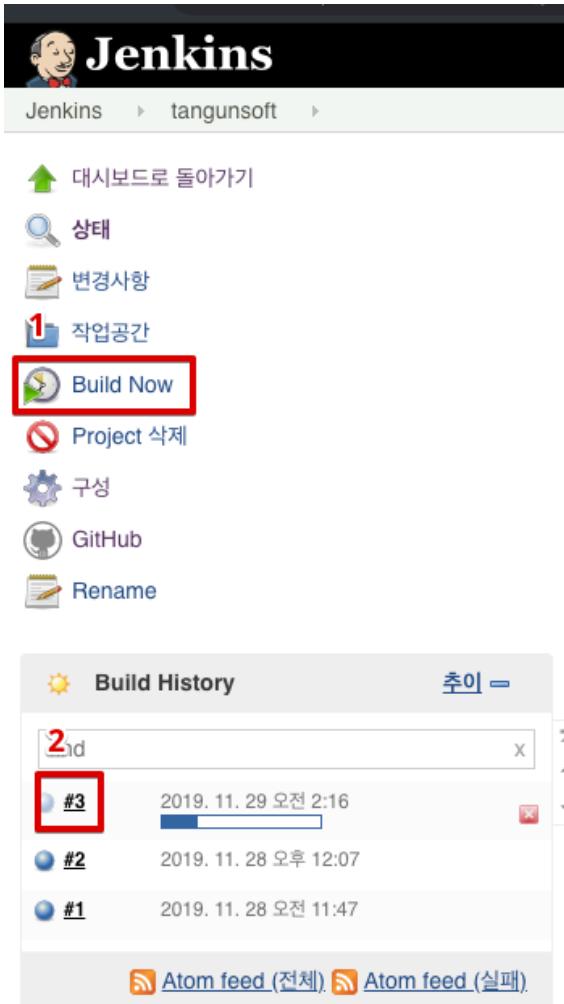
- 프로젝트 항목 중 빌드 후 조치 항목에서 "Send build artifacts over SSH 선택
  - 배포할 SSH 서버 지정
  - 배포할 소스파일과 원격서버 배포경로, 실행 커멘드 입력
  - 입력 후 설정 저장



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

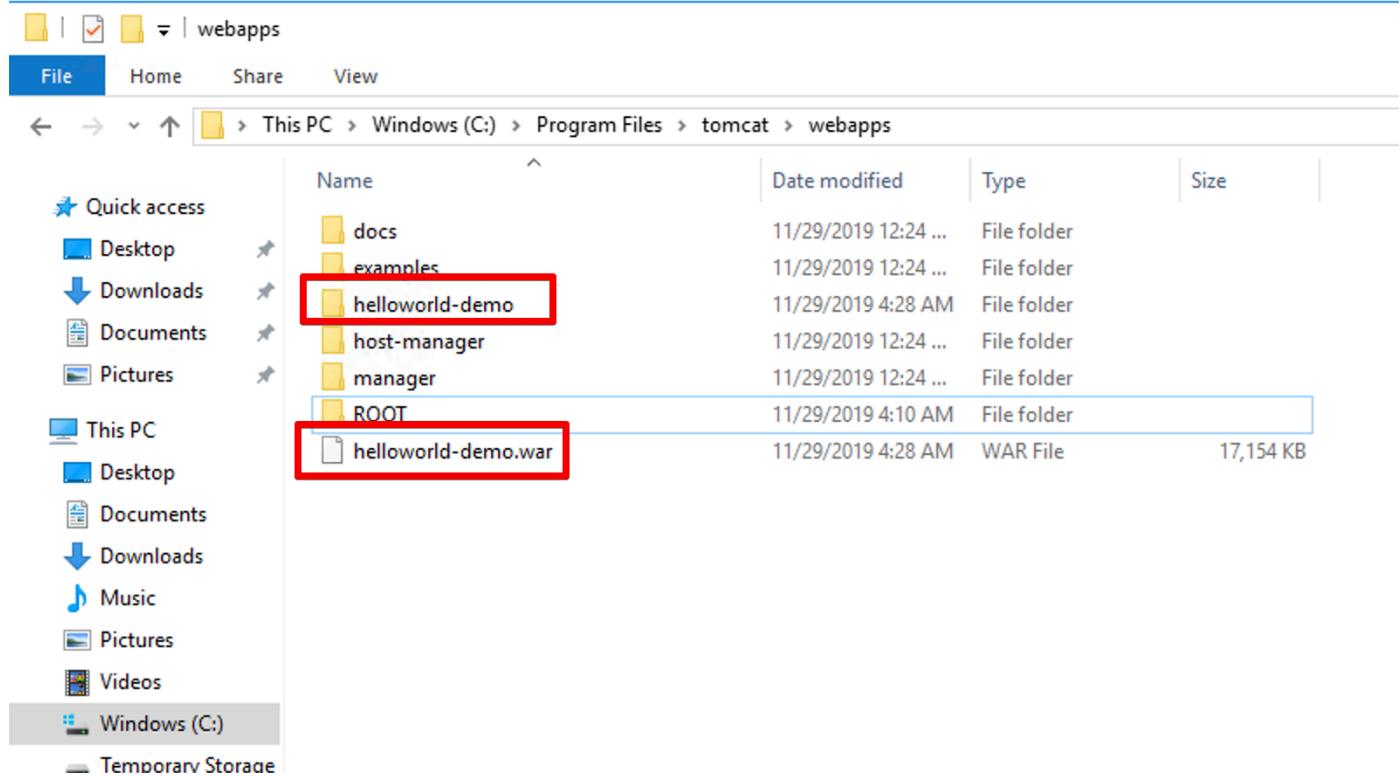
- 프로젝트 빌드 실행
  - 빌드 로그 확인



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

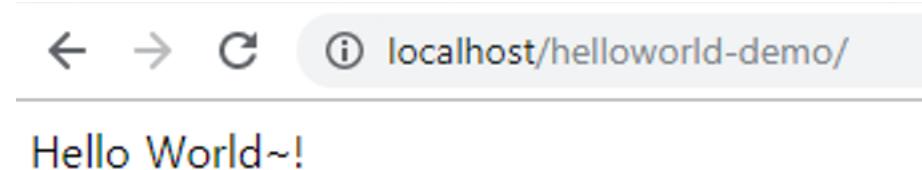
- 파일 전송 내역 확인



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 배포 구성

- [Http://localhost/helloworld-demo](http://localhost/helloworld-demo) 페이지 접근
  - 서비스 반영 내역 확인



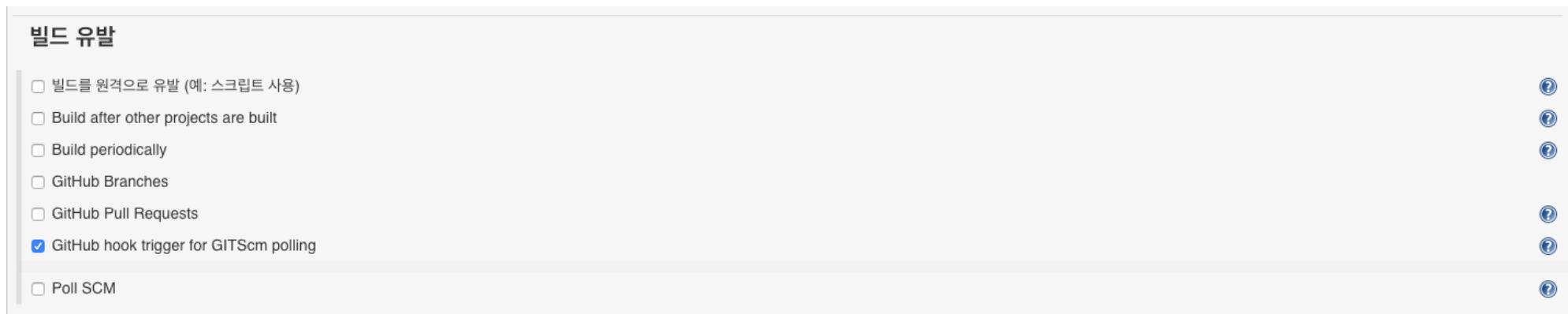
# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- Jenkins 프로젝트에서 빌드 유발 항목

## “GitHub hook trigger for GITScm polling”

선택 후 저장

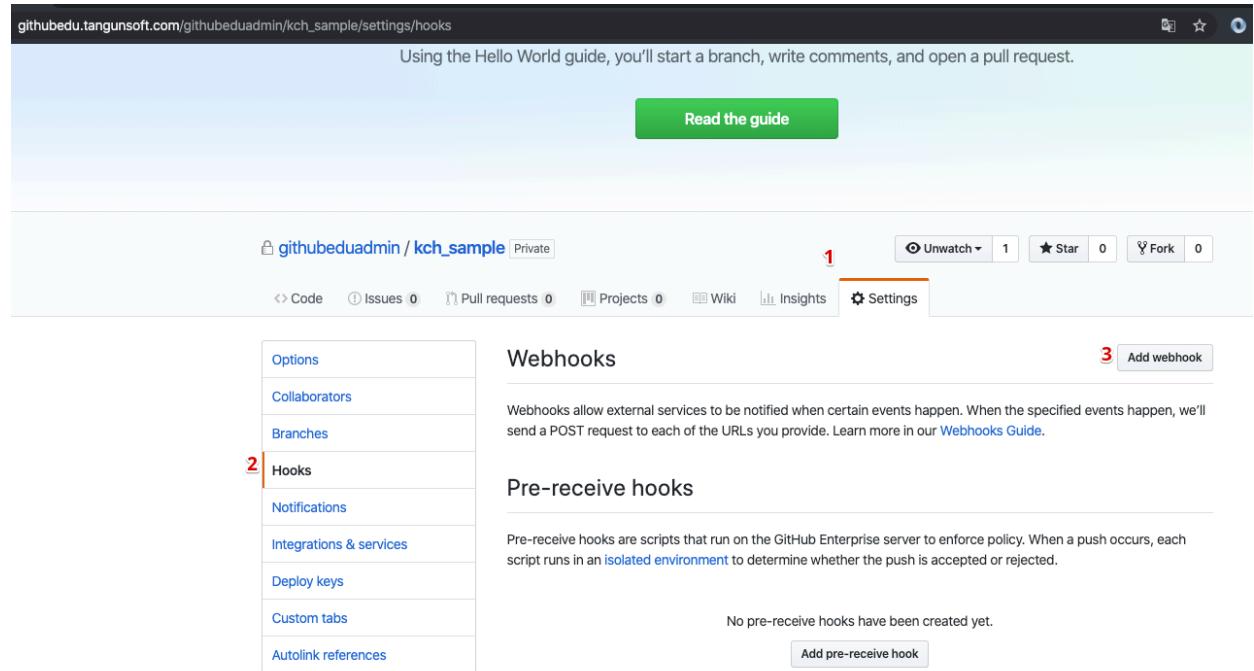


# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- GitHub Enterprise Repo에서

Settings Tab -> Hooks -> Add webhook 클릭



# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- Payload URL에 다음 주소를 입력 후 저장

http(s)://{{Jenkins Url}}/github-webhook/

The screenshot shows the GitHub settings page for a repository named 'githubeduadmin / kch\_sample'. The 'Hooks' tab is selected in the sidebar. A new webhook is being configured with the following details:

- Payload URL:** http://52.141.37.132:8080/github-webhook/ (highlighted with a red box)
- Content type:** application/x-www-form-urlencoded
- Secret:** (empty field)
- Which events would you like to trigger this webhook?**
  - Just the push event.
  - Send me everything.
  - Let me select individual events.
- Active:** (checkbox checked, with a note: "We will deliver event details when this hook is triggered.")

A green 'Add webhook' button is at the bottom right of the form.

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- Webhooks 설정 저장 후 Status 확인
- Webhooks 상세 Status 를 확인하고자 하는 경우 WebHooks Url 클릭

### Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).



[http://52.141.37.132:8080/github-webhook/ \(push\)](http://52.141.37.132:8080/github-webhook/)

Edit

Delete

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- Webhooks 상세 Status 확인

### Recent Deliveries

The screenshot shows a Jenkins interface for managing webhook deliveries. At the top, there's a summary bar with a green checkmark, the delivery ID '19fa0030-14fc-11ea-862a-1e11eb6a2697', the date '2019-12-02 21:05:58', and a 'Completed in 0.11 seconds' message. Below this, there are tabs for 'Request' (selected) and 'Response' (status 200). A 'Redeliver' button is also present. The 'Headers' section displays the request details, including the URL, method (POST), content type (application/x-www-form-urlencoded), and various GitHub-specific headers like X-GitHub-Delivery, X-GitHub-Enterprise-Host, and X-GitHub-Event. The 'Payload' section shows the JSON payload sent by GitHub, which includes information about the repository hook, its configuration (URL, content type, etc.), and its status.

```
Request URL: http://52.141.37.132:8080/github-webhook/
Request method: POST
content-type: application/x-www-form-urlencoded
Expect:
User-Agent: GitHub-Hookshot/df81e86
X-GitHub-Delivery: 19fa0030-14fc-11ea-862a-1e11eb6a2697
X-GitHub-Enterprise-Host: githubbedu.tangunsoft.com
X-GitHub-Enterprise-Version: 2.19.2
X-GitHub-Event: ping

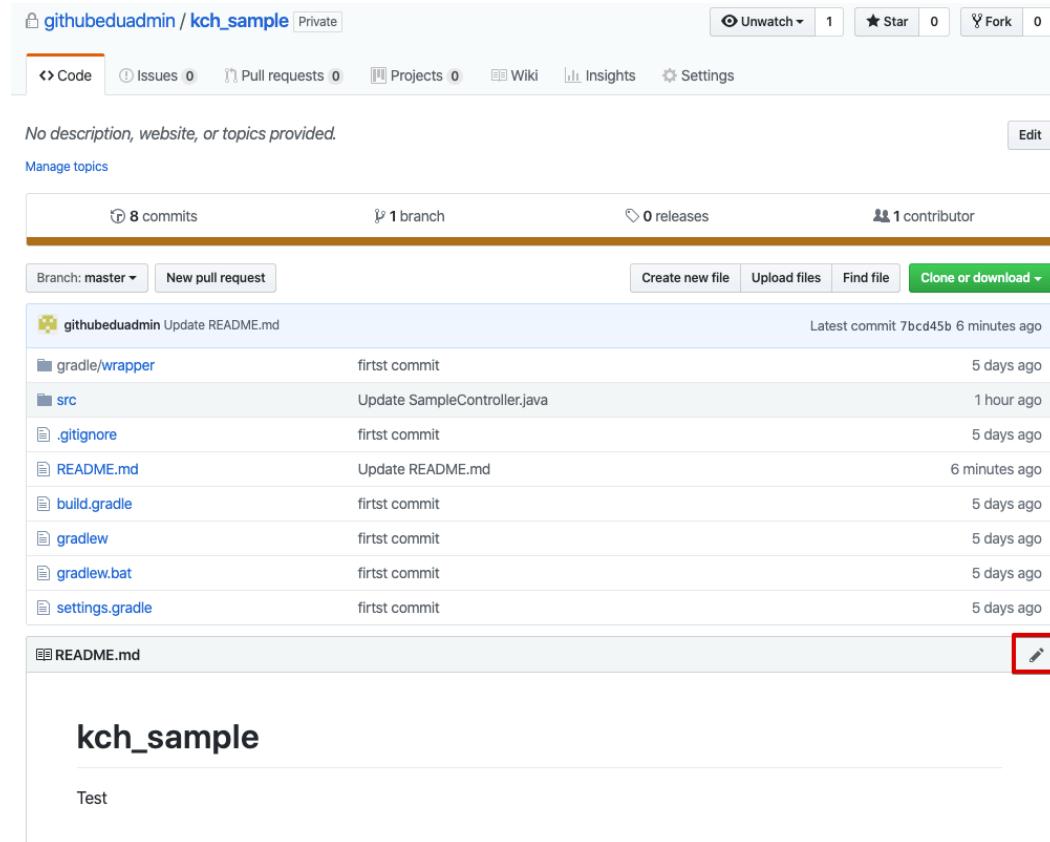
{
  "zen": "Mind your words, they are important.",
  "hook_id": 4,
  "hook": {
    "type": "Repository",
    "id": 4,
    "name": "web",
    "active": true,
    "events": [
      "push"
    ],
    "config": {
      "url": "http://52.141.37.132:8080/github-webhook/",
      "content_type": "form",
      "insecure_ssl": "0"
    },
    "updated_at": "2019-12-02T12:05:58Z",
    "created_at": "2019-12-02T12:05:58Z",
    "url": "http://githubbedu.tangunsoft.com/api/v3/repos/githubbeduadmin/kch_sample/hooks/4"
  }
}
```

# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- GitHub Repo에서 README 파일 생성 후

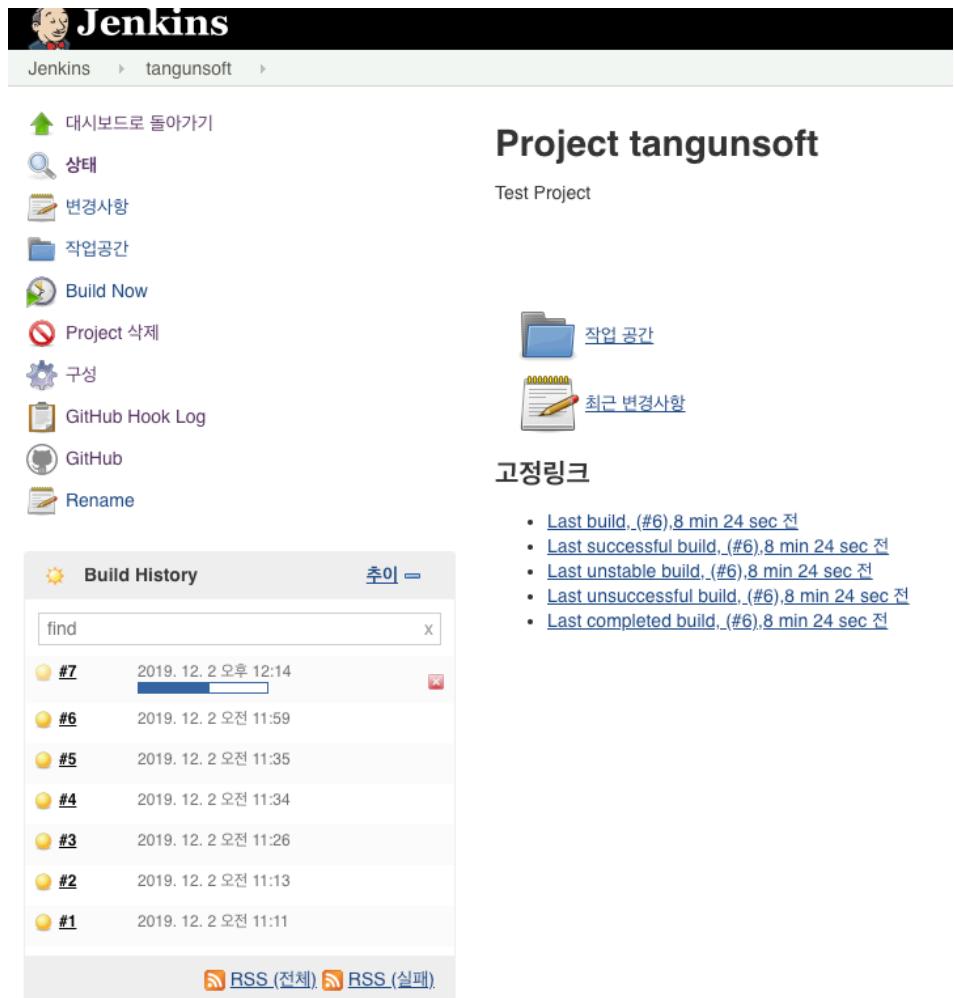
## 수정하여 Webhook 유발



## Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- Jenkins Project Build 목록에서 Webhook을 통한 빌드가 실행되는지 확인



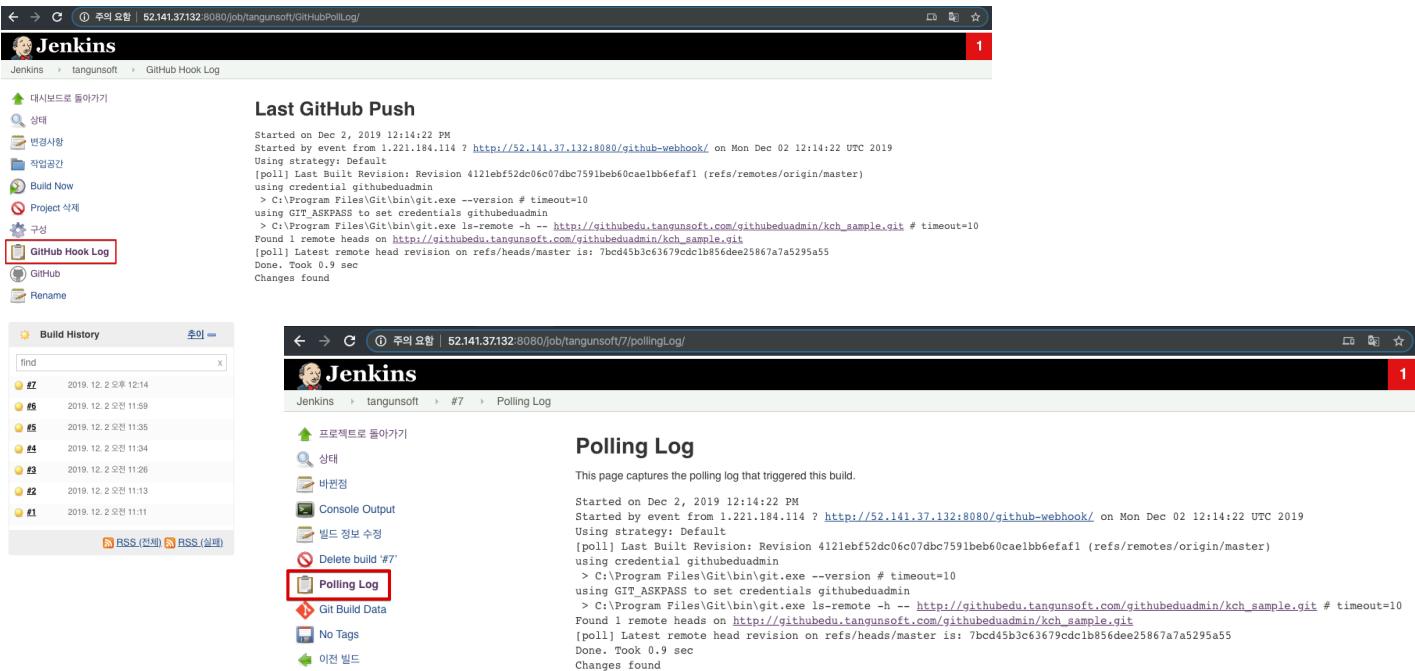
# Jenkins 프로젝트 구성

## Jenkins 프로젝트 webhook 구성

- Jenkins Project 메뉴의 GitHub Hook Log 또는

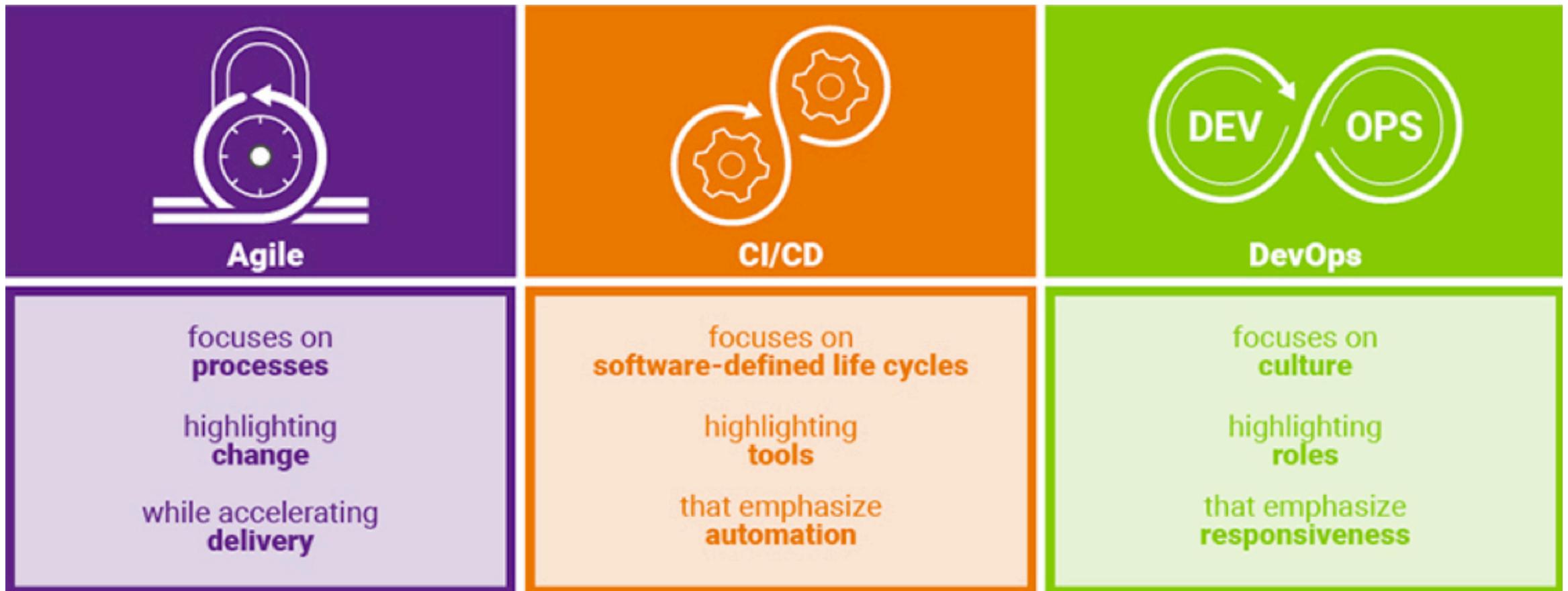
Build History에서 확인한 빌드 넘버 선택 후

## Polling Log에서 WebHook Log 확인



# CI/CD에서 DevOps로

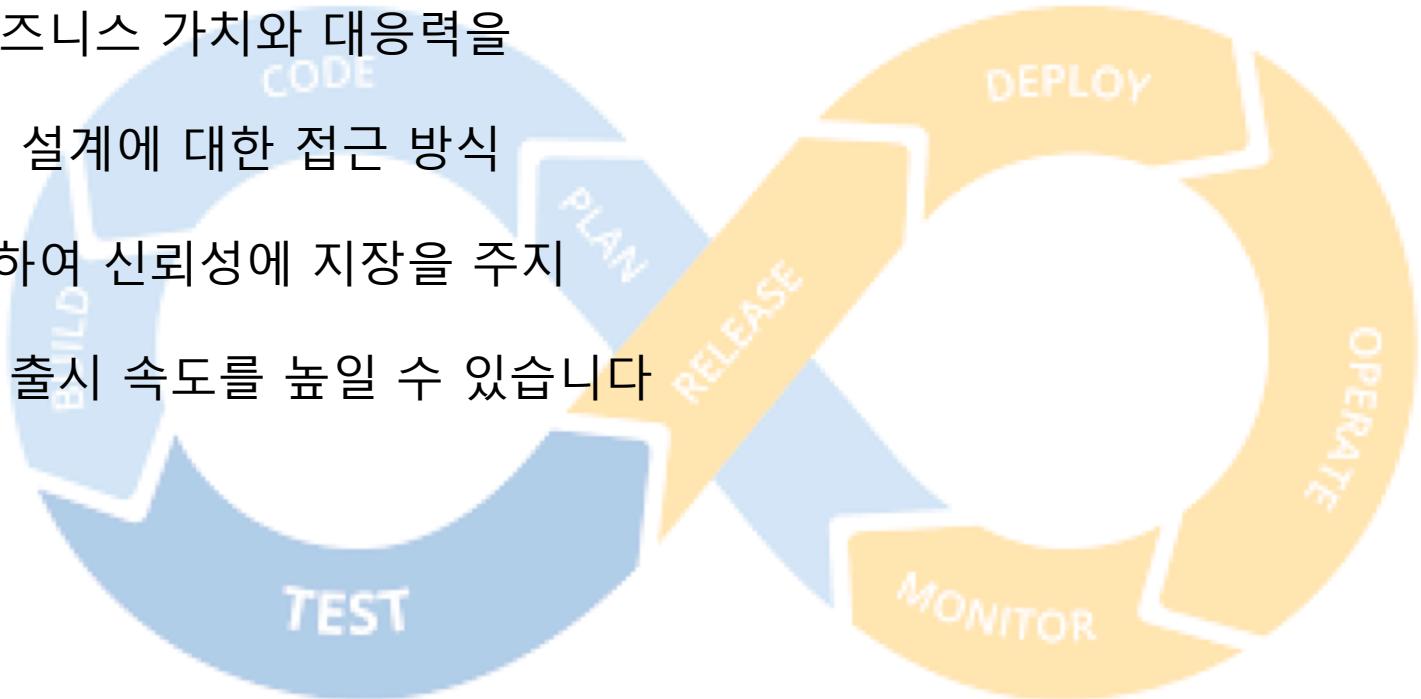
# CI/CD에서 DevOps로



# DevOps란?

## DevOps

- development(개발) + operations(운영) 합성어
- 신속한 고품질 서비스 제공을 통해 비즈니스 가치와 대응력을 향상시키기 위한 문화, 자동화, 플랫폼 설계에 대한 접근 방식
- 개발자와 IT 운영 팀이 밀접하게 협력하여 신뢰성에 지장을 주지 않으면서 소프트웨어 구축, 테스트 및 출시 속도를 높일 수 있습니다



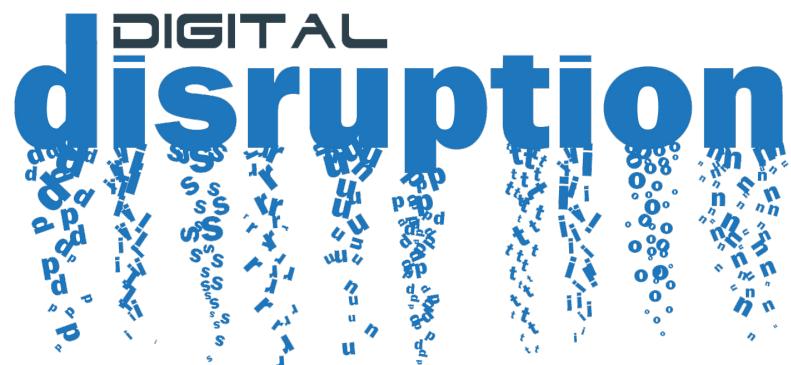
# 기업들이 DevOps에 투자하는 이유

## FORTUNE 선정 100 대 기업 분석결과

2000년 **100대** 기업 중 2018년 100대 기업으로 선정된 기업은 단 **31개** 업체!

2000년 **500대** 기업 중 2018년에 남아있는 기업은 **약 250개** 업체로 **50%** 이상이 **부도** 또는 인수합병

## 개발자는 기업 혁신의 중심!



### ✓ Digital Disruption 이란 ?

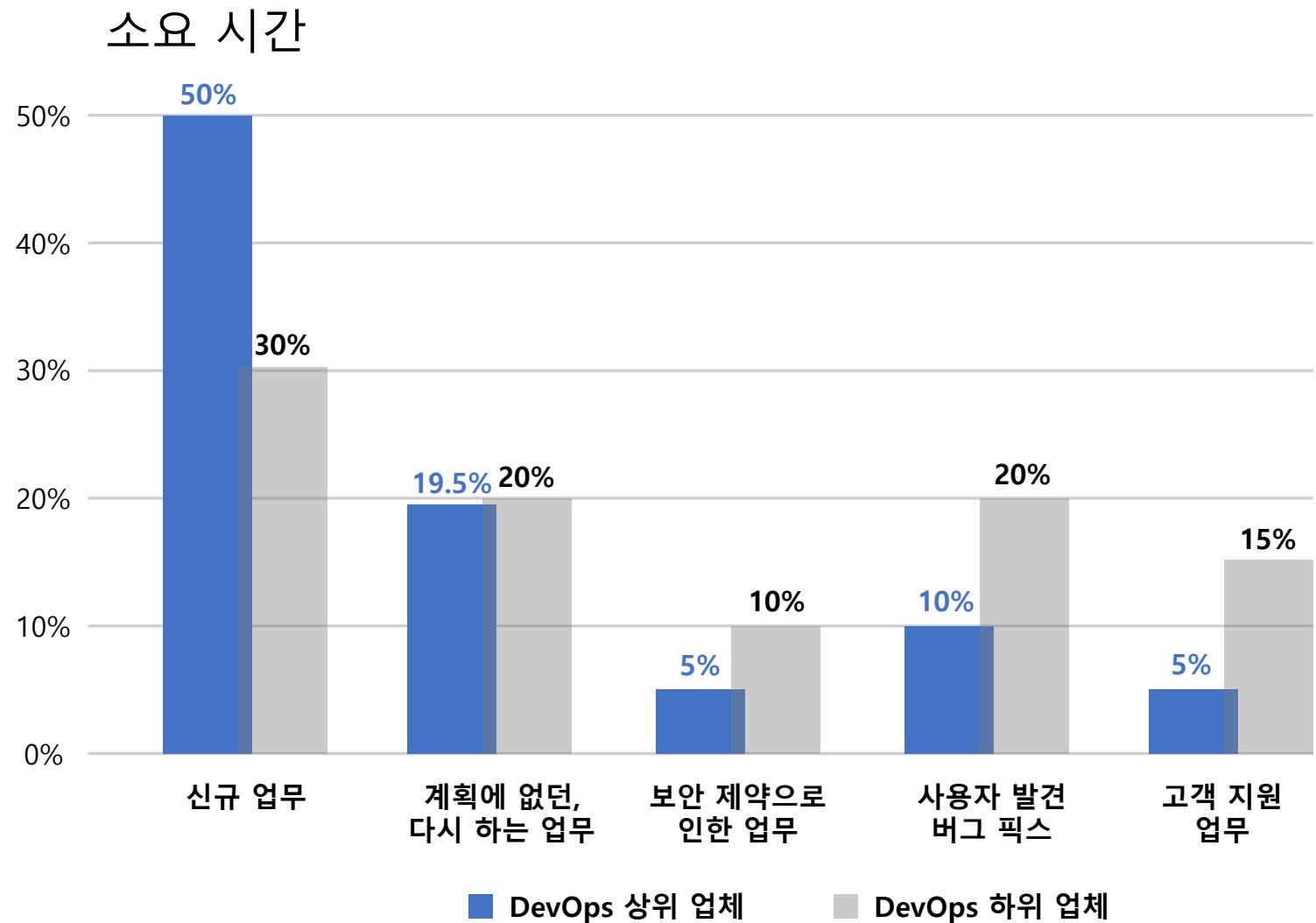
디지털 혁신이 일으키는 변화에 적응하지 못하는 기업들이 사라지는 현상을 일컫는 용어이다.

모든 산업은 디지털화 되어 있고, 디지털화로 변화하는 속도에 따라 경쟁에서 우위를 차지하거나, 뒤처질 수 있다.

# 기업들이 DevOps에 투자하는 이유

**혁신에  
더 많은 시간을  
투자해야 합니다.**

소프트웨어 분야 상위 업체는  
**혁신에 더 많은 시간을 투자**하고  
단순 작업에 적은 시간을  
소비하고 있습니다.

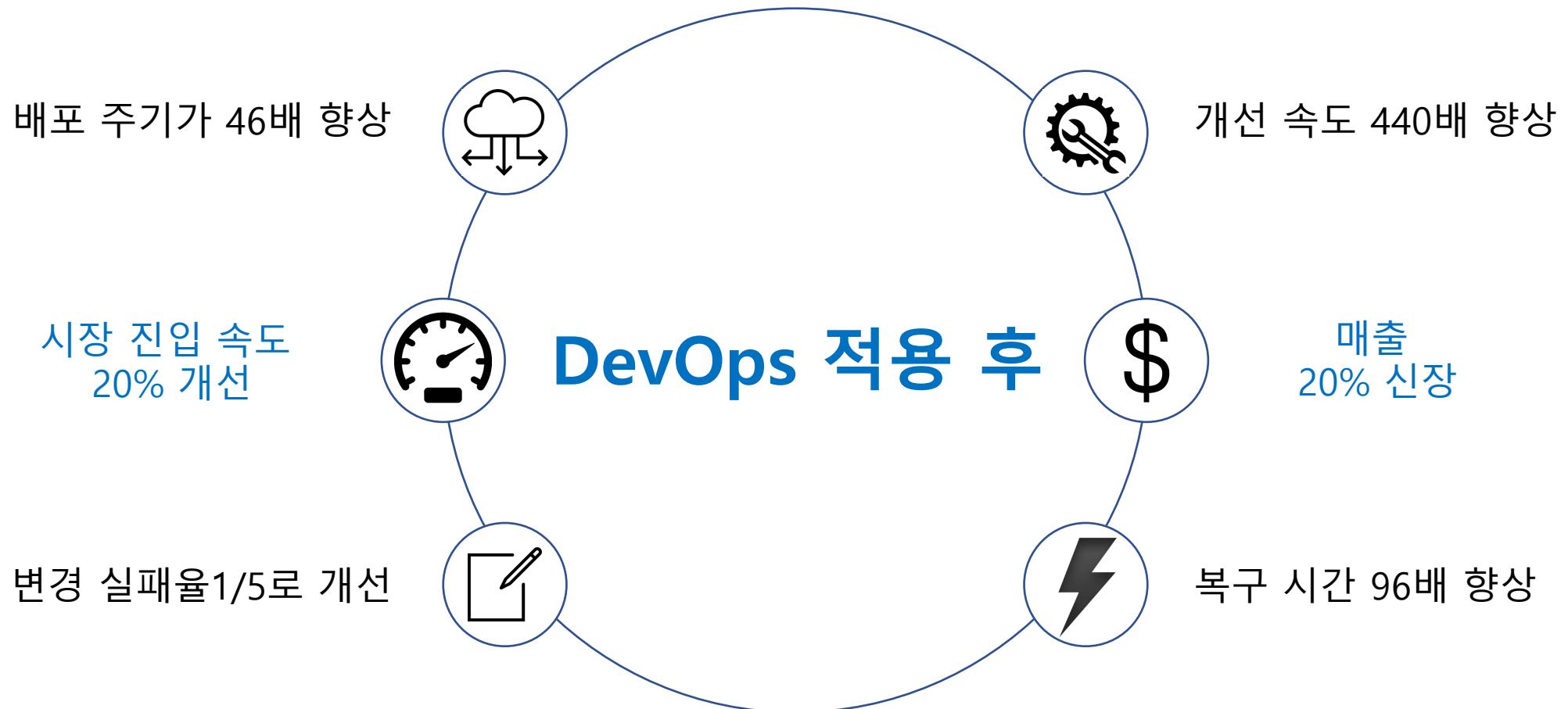


출처: Source 2018 State of DevOps Report, presented by DORA

# 기업들이 DevOps에 투자하는 이유



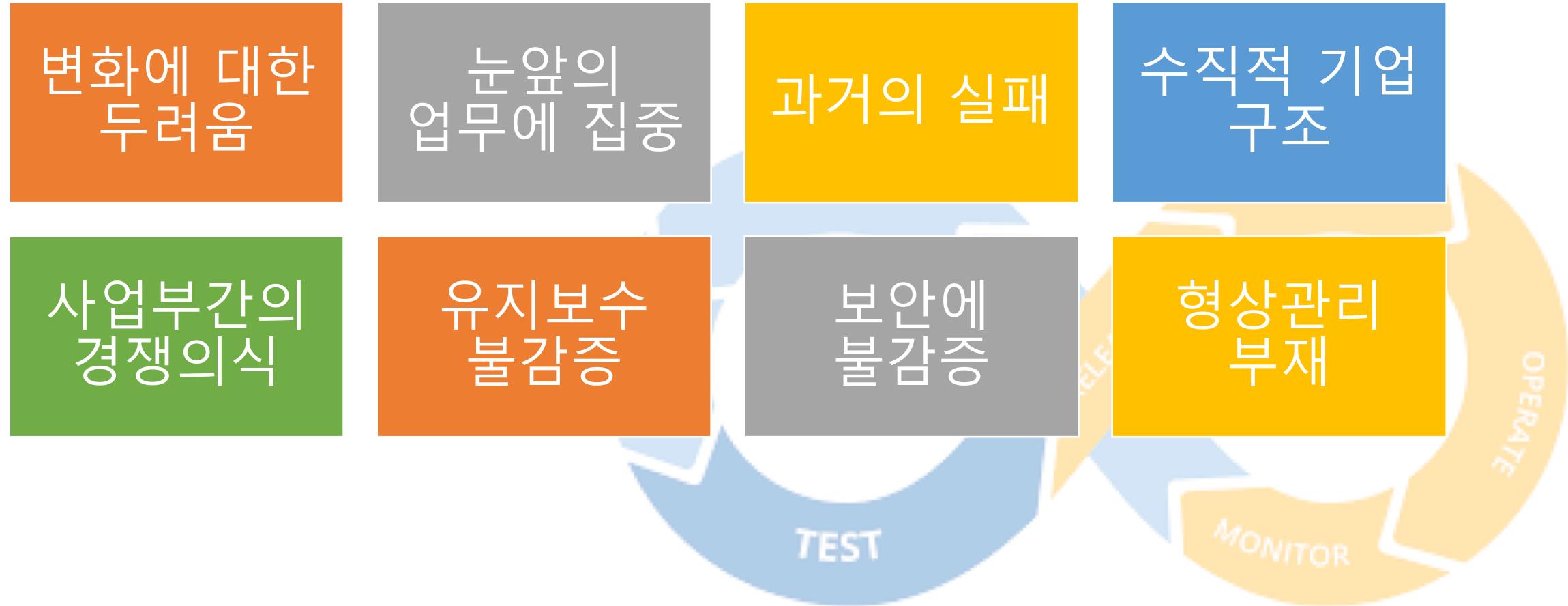
# 기업들이 DevOps에 투자하는 이유



출처: Source 2017 State of DevOps Report, presented by DORA

# DevOps 진입장벽

## DevOps 로의 진입 장벽



# 회사 소개

## 일반현황

회사명	(주)단군소프트
설립일	2002년 1월 21일
직원수	52명
사업분야	컨설팅, 소프트웨어 유통, 기술지원 서비스
소재지	서울 강남구 언주로 620 7층 (논현동, 현대인텔렉스빌딩)
자회사	(주)케이투소프트
해외지사	말레이시아
교육센터	단군소프트 교육장 (15인석)

## 자회사 현황

회사명	(주)케이투소프트
설립일	2007년 10월 19일
직원수	11명
사업분야	소프트웨어 개발
소재지	서울 강남구 언주로 620 7층 (논현동, 현대인텔렉스빌딩)

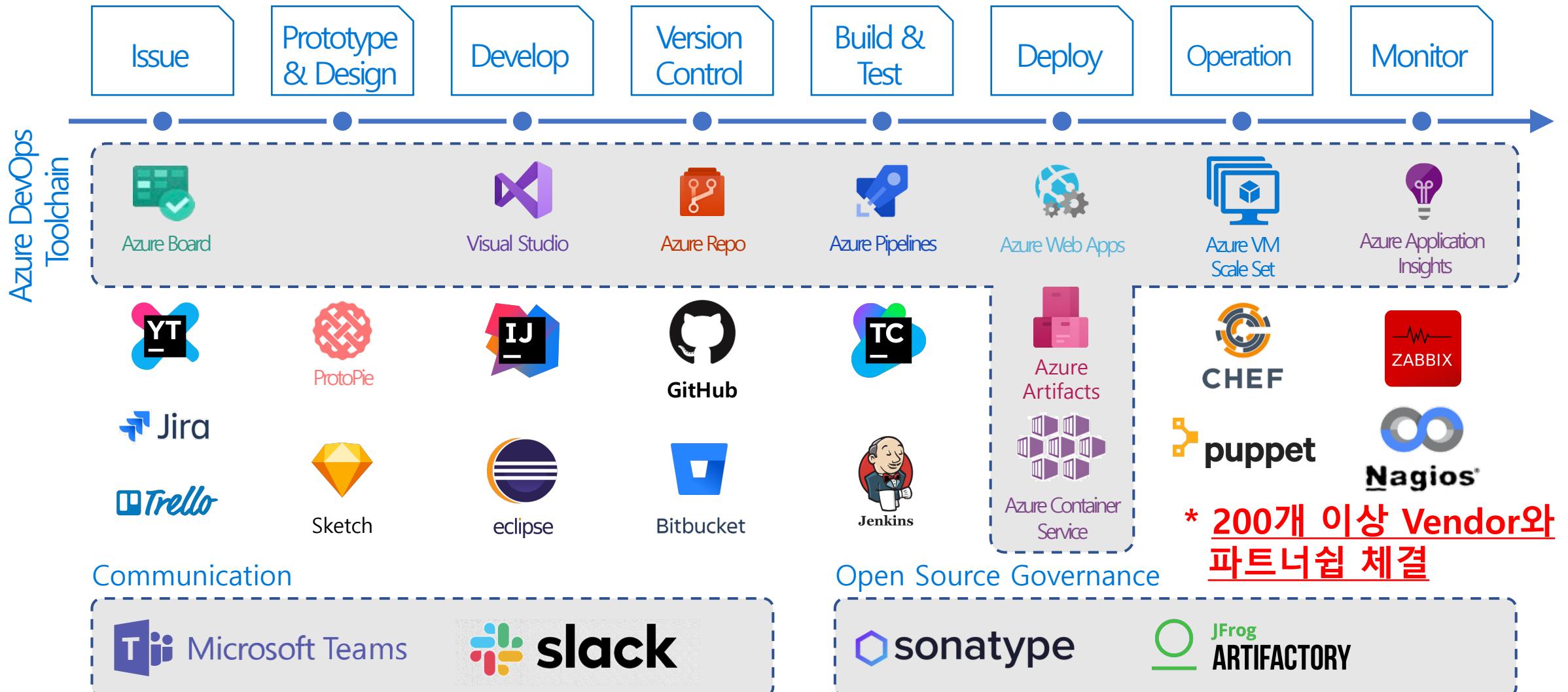
# 단군소프트의 강점(DevOps)

DevOps

- 단군소프트는 17년 이상 해외 소프트웨어 **개발툴과 오픈소스 및 컴포넌트를 취급한 노하우**와 고객사를 확보하고 있습니다.
- 다양한 분야의 고객층에 대한 요구사항에 대한 **경험으로** 스타트-업부터 Enterprise 까지 **고객사 환경에 필요한 다양한 분야의 툴체인(Toolchain)을 보유하고 있습니다.**

# 단군소프트 DevOps Enterprise Toolchain

DevOps



- 단군소프트는 고객에 맞춤형 DevOps 구현을 위한 단계별 고객서비스를 제공합니다.
- DevOps 구현을 위한 오픈소스 활용 시 전문화된 분석을 통한 거버넌스 확보와 통합 및 자동화를 통한 DevOps 최적화를 제공합니다.



# 단군소프트 DevOps 컨설팅팀 구성

DevOps



## 기술 컨설팅

- 고객사 상황에 맞는 솔루션 가이드 및 관리자 교육
- 맞춤형 Agile 방법론 제시
- 장애 또는 요청 사항에 대한 전담 인력 지원
- 서비스 자동화 및 유지 관리

## 영업 컨설팅

- 비즈니스 요구 사항에 충족하는 라이선스 컨설팅
- 라이선스 서비스 유지 관리
- 온/오프라인 제품 소개 및 교육 가이드

# - 감사합니다 -