젠킨스 설정하기

1. Jenkins란?

1) CI

지속적 통합(Continuous Integration) - 지속적으로 품질 관리를 적용하는 프로세스를 실행하는 것

2) Jenkins

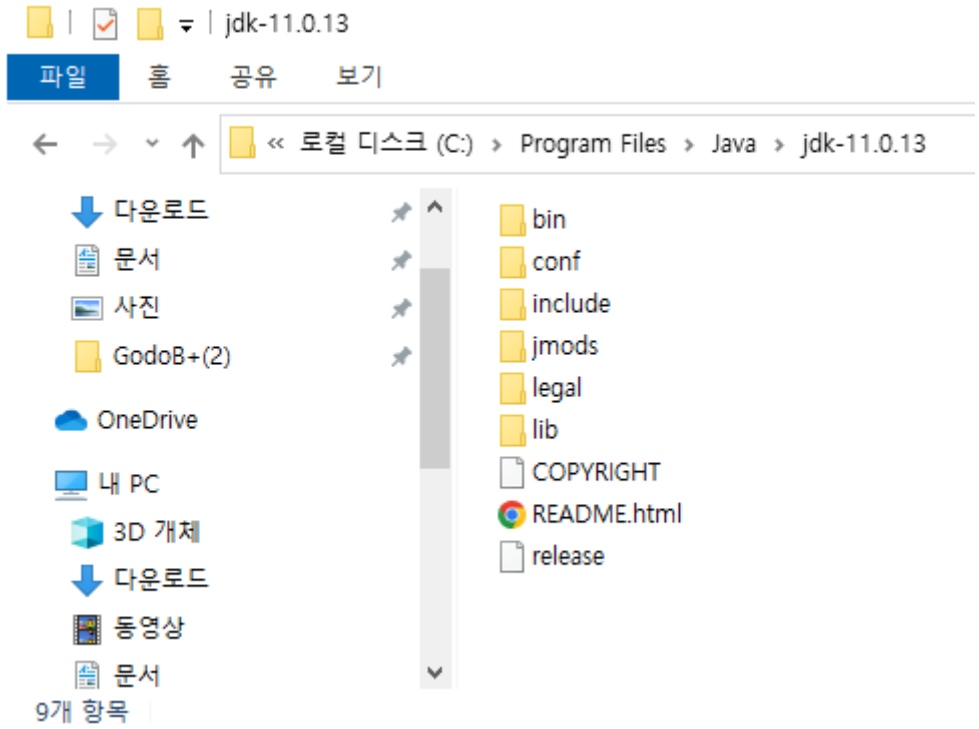
지속적 통합 서비스를 제공하는 툴(tool)로 허드슨(hudson) 프로젝트에서 시작

다수의 개발자들이 하나의 애플리케이션을 개발할 때
버전 충돌을 방지하기 위해서 github을 활용하는데
젠킨스는 github에 있는 애플리케이션을
자동으로 빌드(build)하고 톰캣 서버에 배포(deploy)해 줌

2. Java

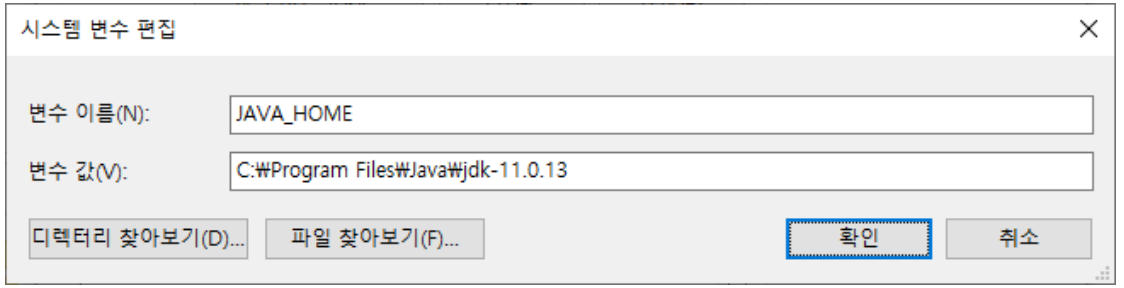
1) 자바 설치 경로 확인

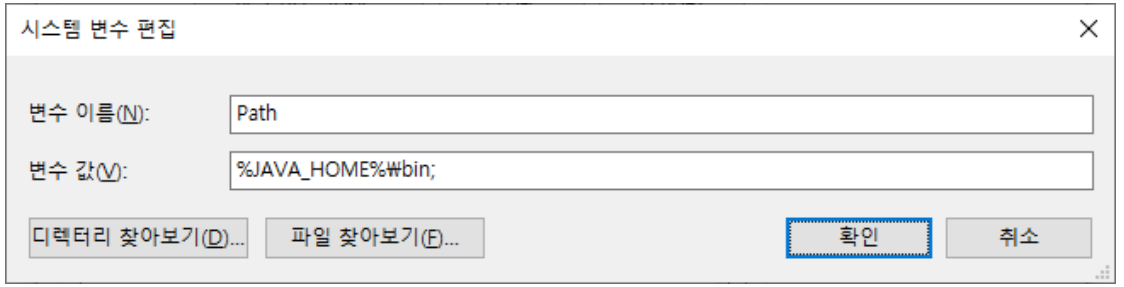
C:₩Program Files₩Java₩jdk-11.0.13



2) 자바 환경변수

제어판 - 시스템



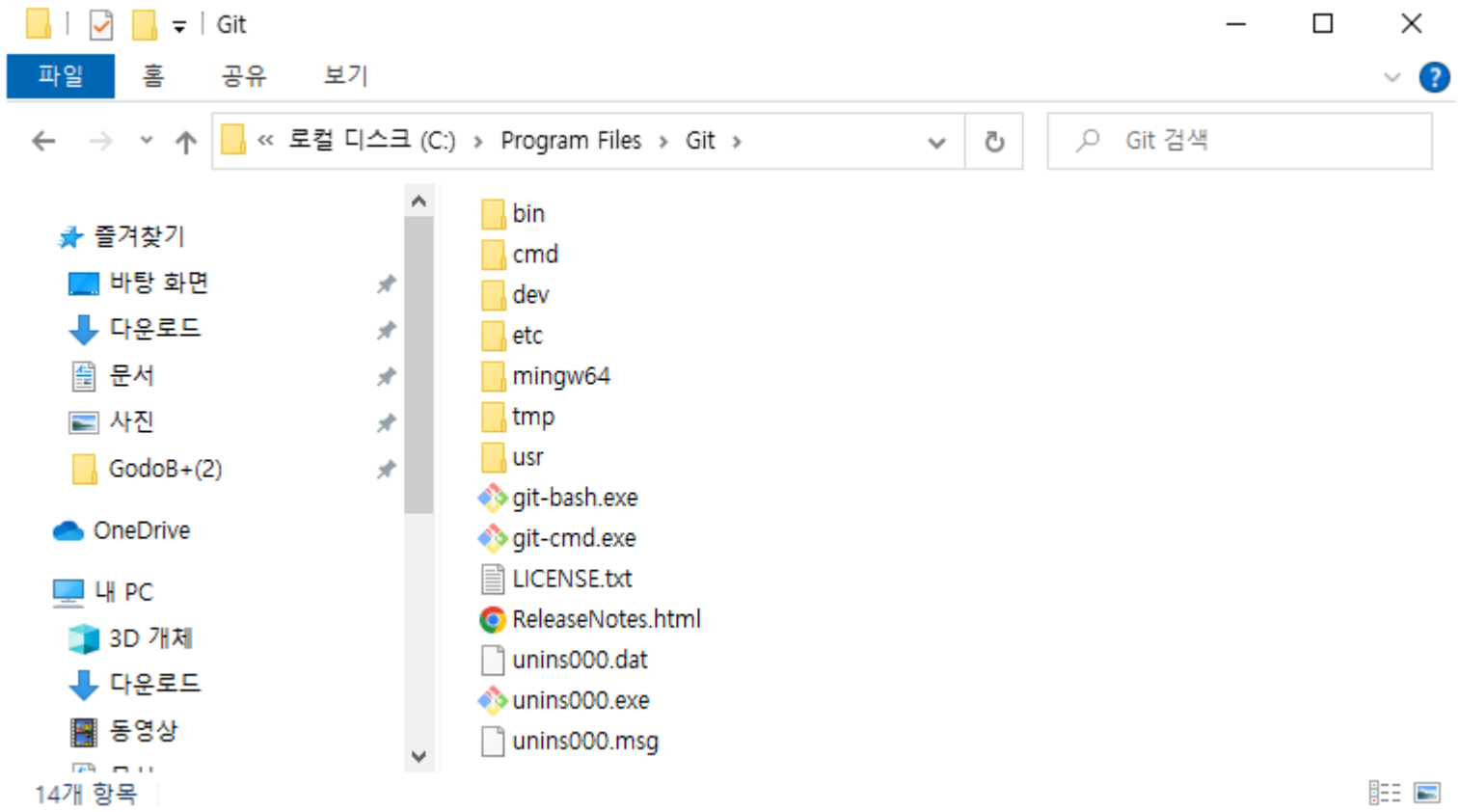


Jenkins

3. Git

1) Git 설치 경로 확인

C:₩Program Files₩Git

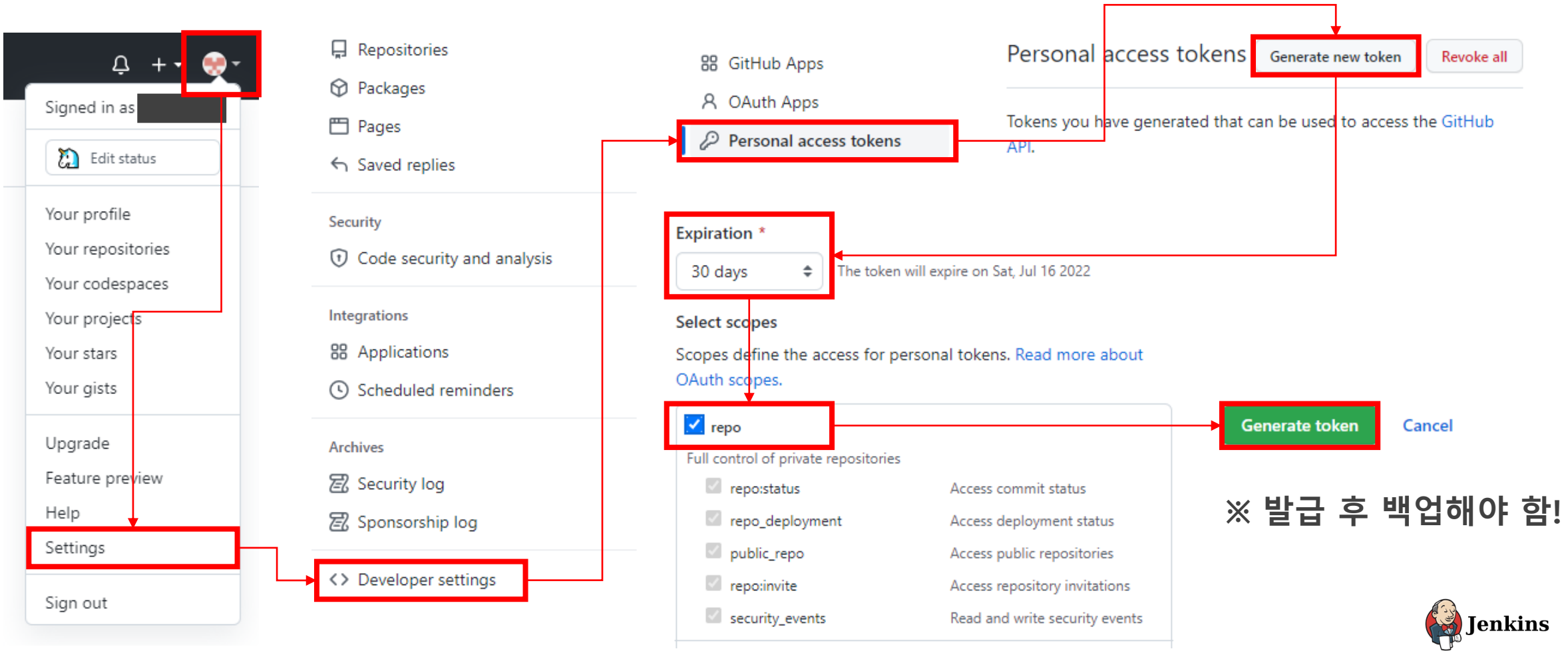


Jenkins

3. Git

2) Personal Access Token

github 로그인 - [계정 아이콘] - [Settings] - [Developer settings] - [Personal access token]



Jenkins

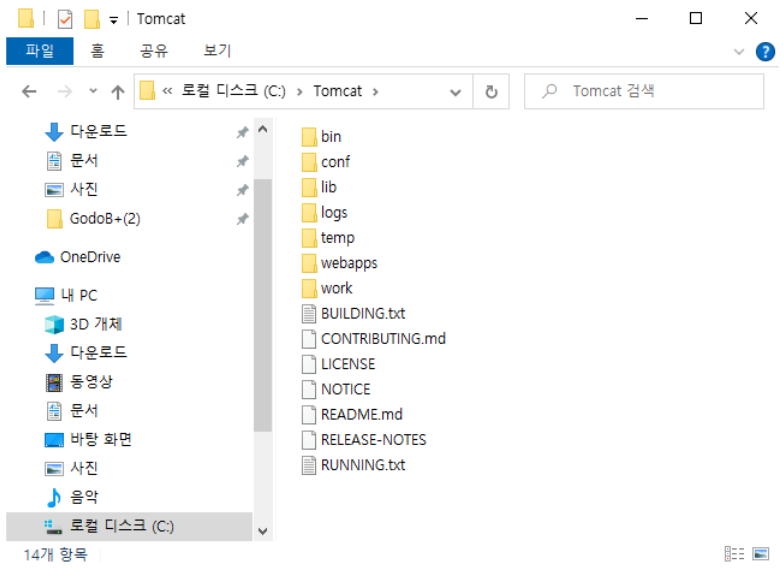
4. Tomcat

1) 톰캣 다운로드 (버전 9.0.x)

https://tomcat.apache.org/

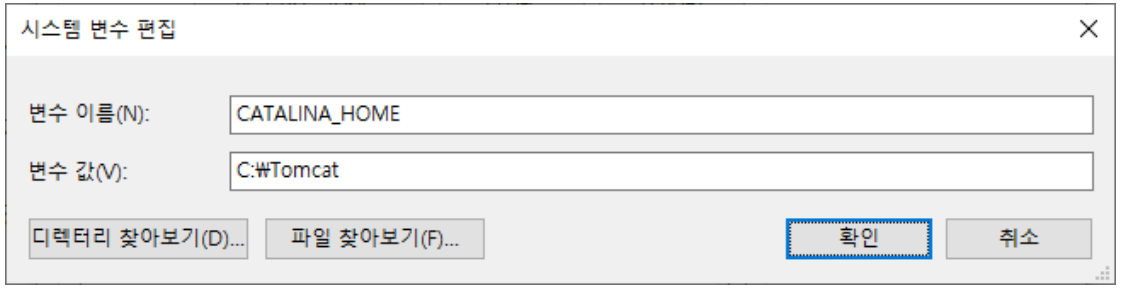
2) 톰캣 설치

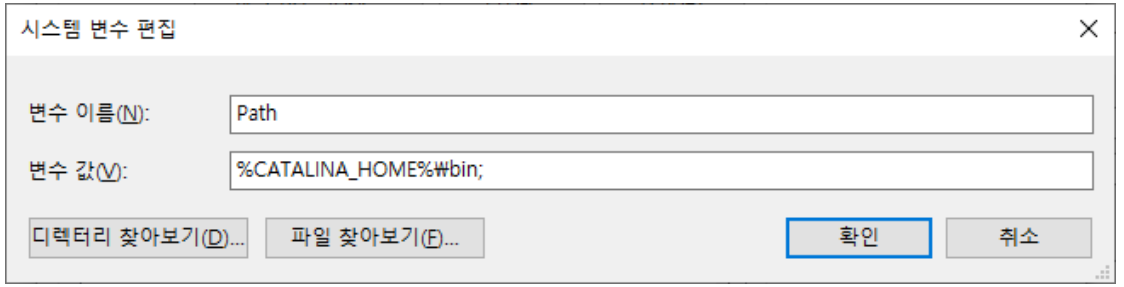
C:₩Tomcat



3) 톰캣 환경변수

제어판 - 시스템





Jenkins

4. Tomcat

4) 포트번호 수정 (Oracle DB인 경우)

C:₩Tomcat₩conf₩server.xml

태그 내용 대신 주석을 수정하지 않도록 주의!

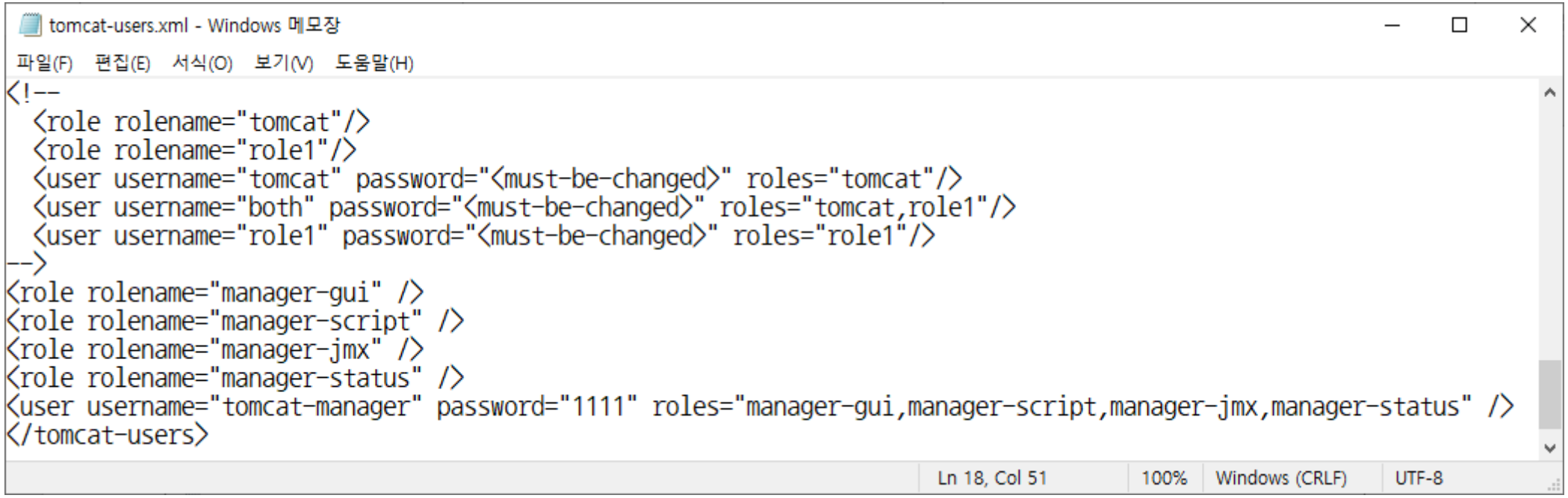
Jenkins

4. Tomcat

5) 톰캣 관리자 등록

C:₩Tomcat₩conf₩tomcat-users.xml

관리자명 : tomcat-manager
비밀번호 : 1111



Jenkins

5. Maven

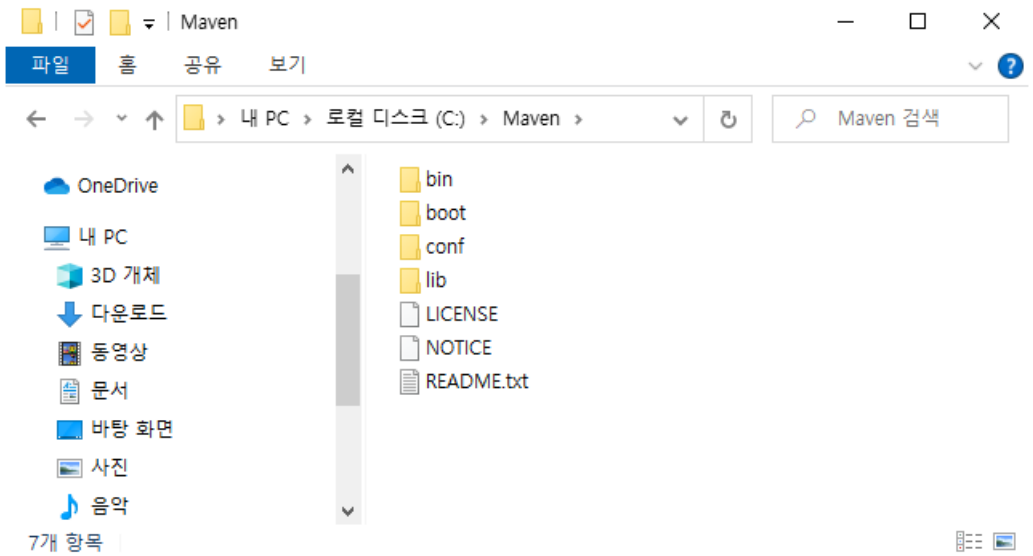
1) 메이븐 다운로드 (버전 3.8.1)

https://maven.apache.org/

* mvnrepository.com에 있는 버전을 사용해야 함

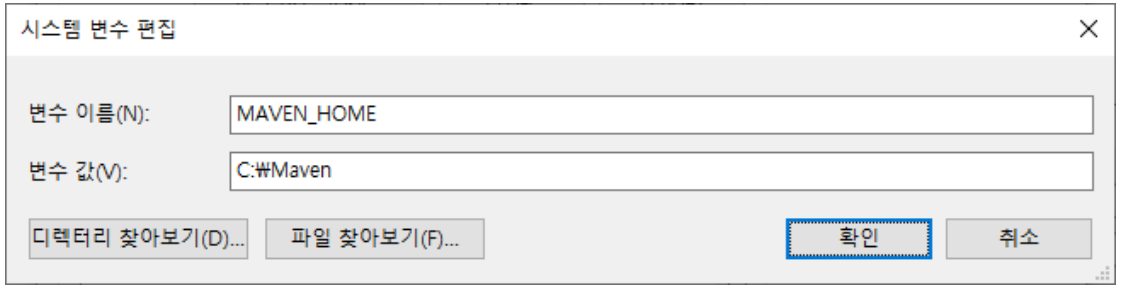
2) 메이븐 설치

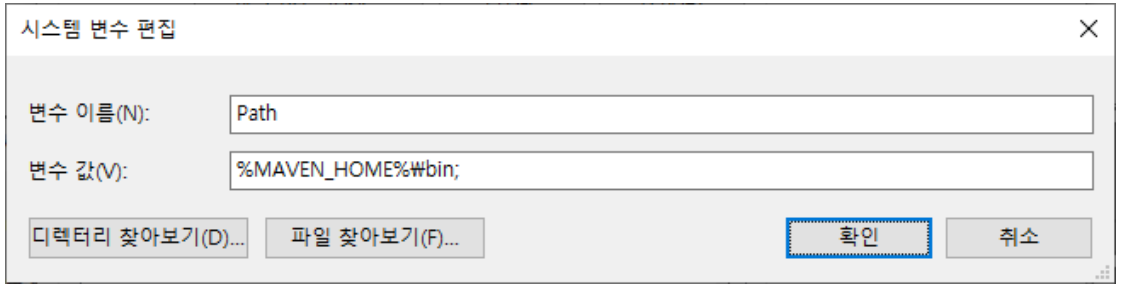
C:₩Maven



3) 메이븐 환경변수

제어판 - 시스템





Jenkins

6. Application

pom.xml

```
<groupId>com.company</groupId>
<artifactId>app</artifactId>
<name>project</name>
<packaging>war</packaging>
<version>1.0.0</version>
```

빌드 결과로 app-1.0.0.war 파일이 생성됨.
파일명의 버전을 없애려면 finalName 태그를 추가함.
finalName 태그가 추가되면 app.war 파일이 생성됨.

```
<build>
    <finalName>${artifactId}</finalName>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>${maven-version}</version>
            <configuration>
                <source>${java-version}</source>
                <target>${java-version}</target>
                <compilerArgument>-Xlint:all</compilerArgument>
                <showWarnings>true</showWarnings>
                <showDeprecation>true</showDeprecation>
            </configuration>
        </plugin>
    </plugins>
</build>
```

```
<properties>
    <java-version>11</java-version>
    <org.springframework-version>5.3.3</org.springframework-version>
    <org.aspectj-version>1.9.6</org.aspectj-version>
    <org.slf4j-version>1.7.30</org.slf4j-version>
    <maven-version>3.8.1</maven-version>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
</properties>
```

설치한 Maven 버전과 젠킨스 빌드에서 사용할 Encoding

Jenkins

7. Jenkins Start

1) 젠킨스 다운로드
https://www.jenkins.io/

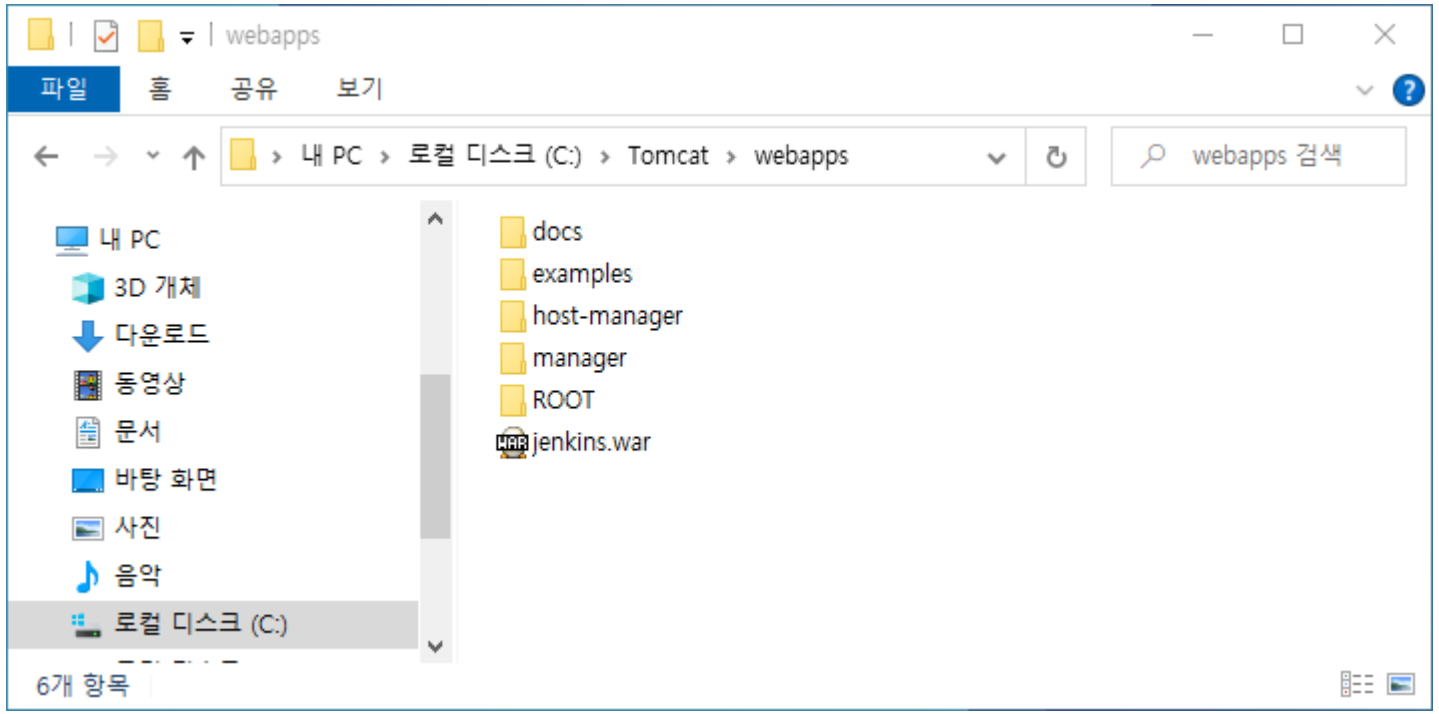
LTS 버전
Generic Java Package(.war)

Jenkins

7. Jenkins Start

2) 젠킨스 배포

C:₩Tomcat₩webapps 디렉터리에
다운로드 받은 jenkins.war 파일을 복사해 둠



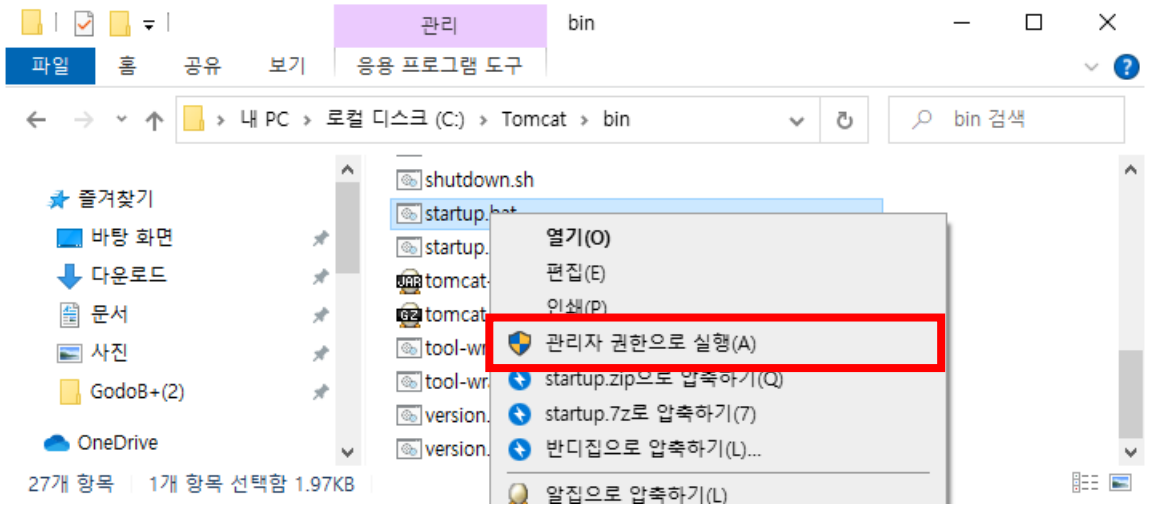
Jenkins

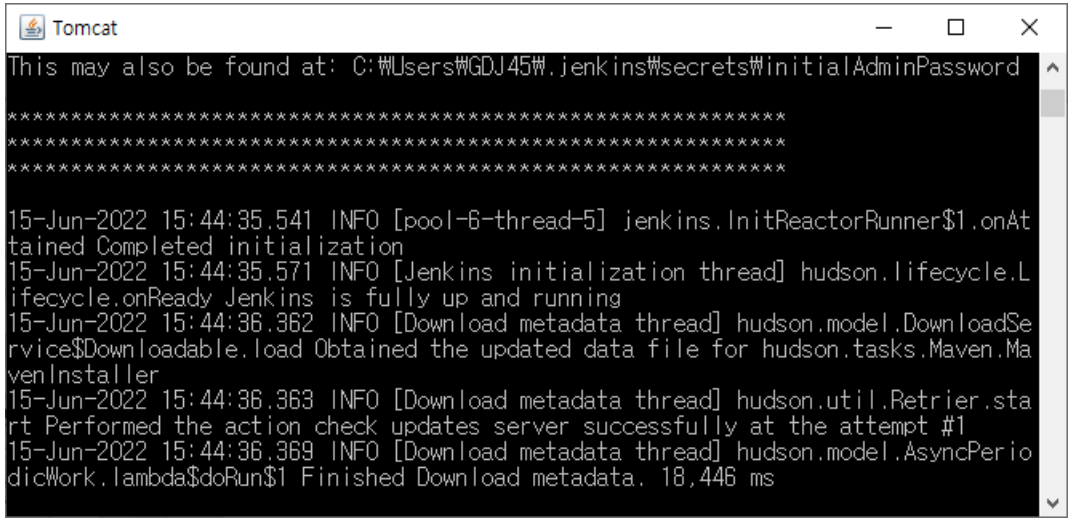
7. Jenkins Start

3) 톰캣 실행

C:₩Tomcat₩bin₩startup.bat 파일을
[관리자 권한]으로 실행

* 리눅스에서는 startup.sh 파일 실행





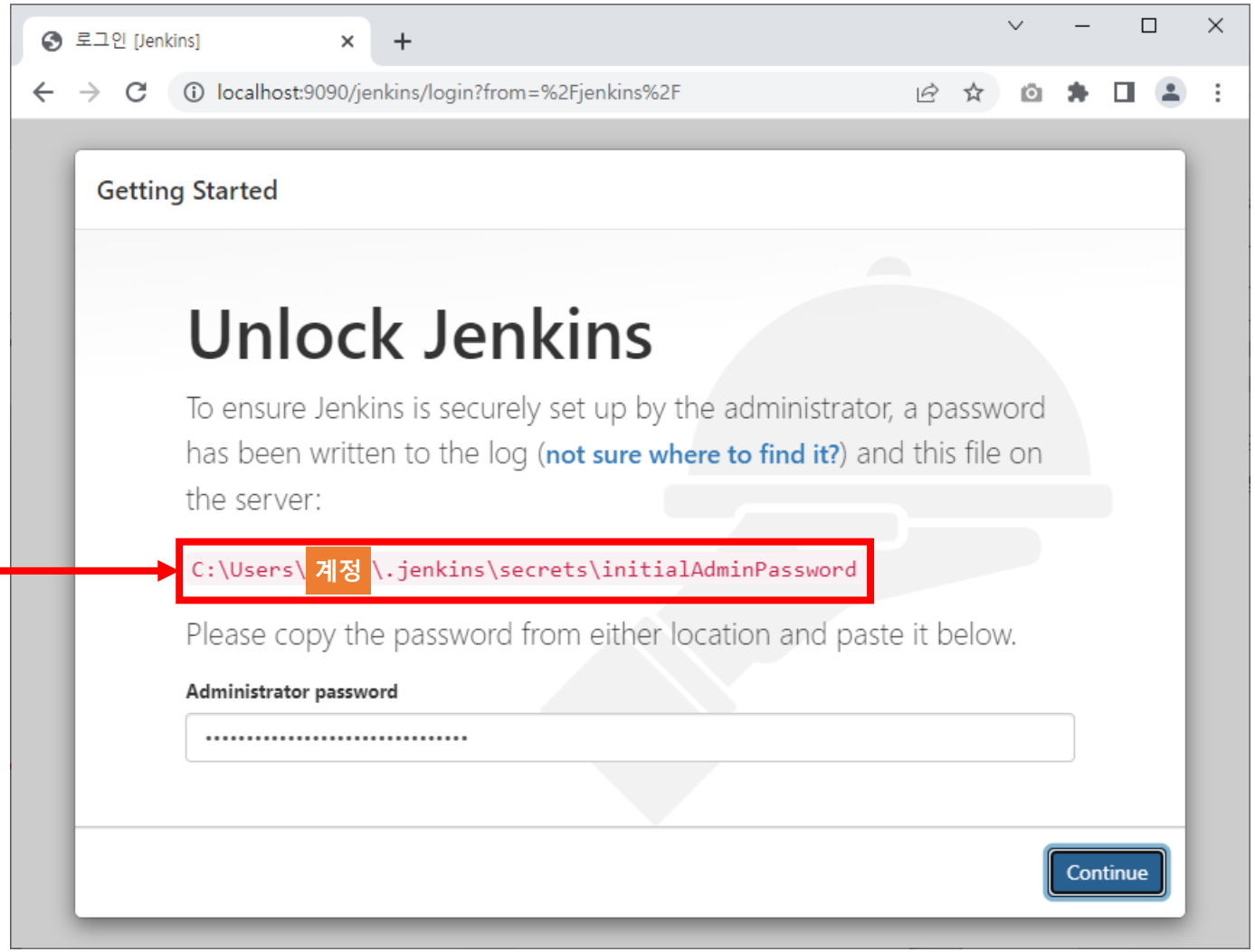
끄지 말 것!

Jenkins

7. Jenkins Start

4) 젠킨스 실행
http://localhost:9090/jenkins 접속

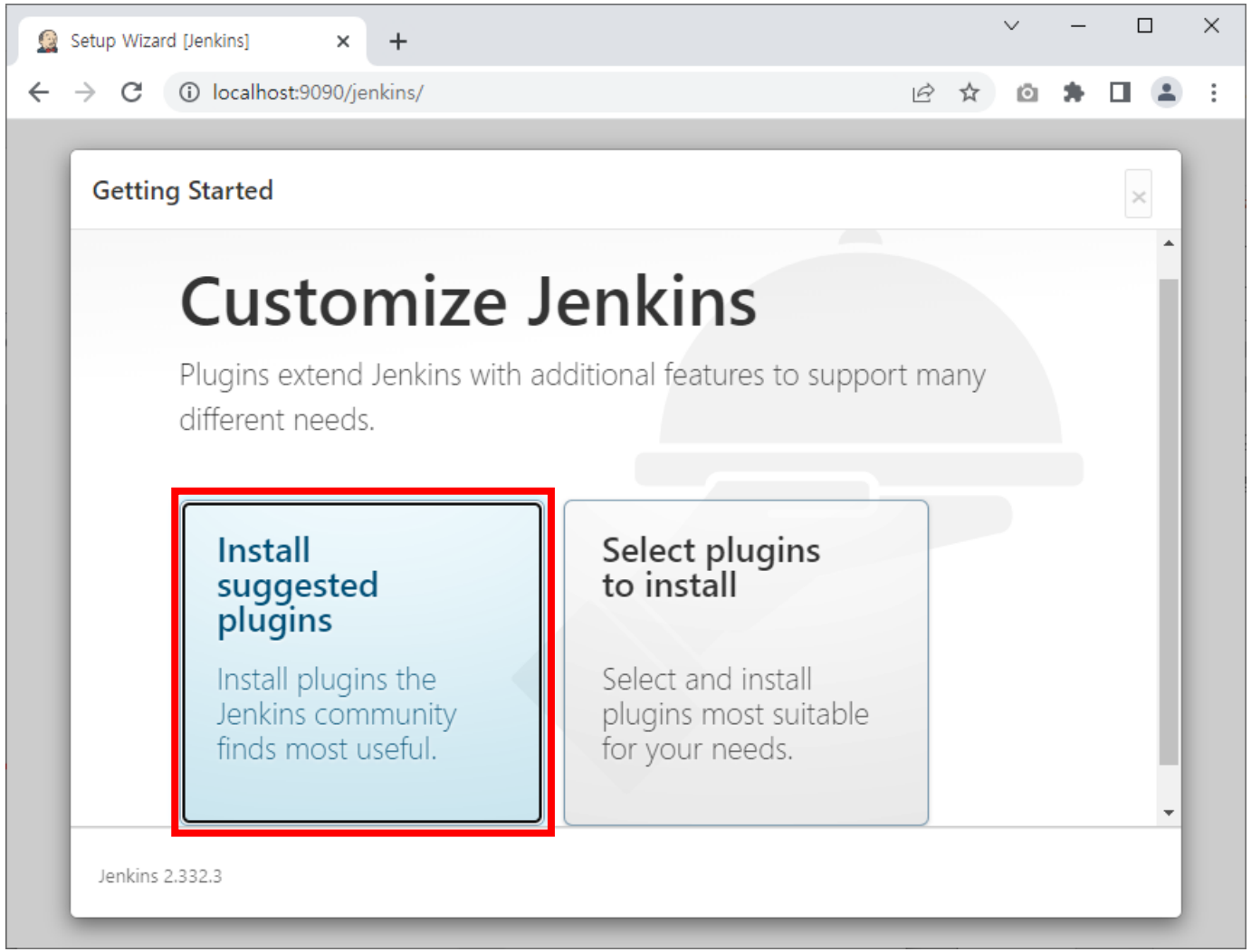
초기비밀번호 입력
(초기비밀번호 위치)



Jenkins

7. Jenkins Start

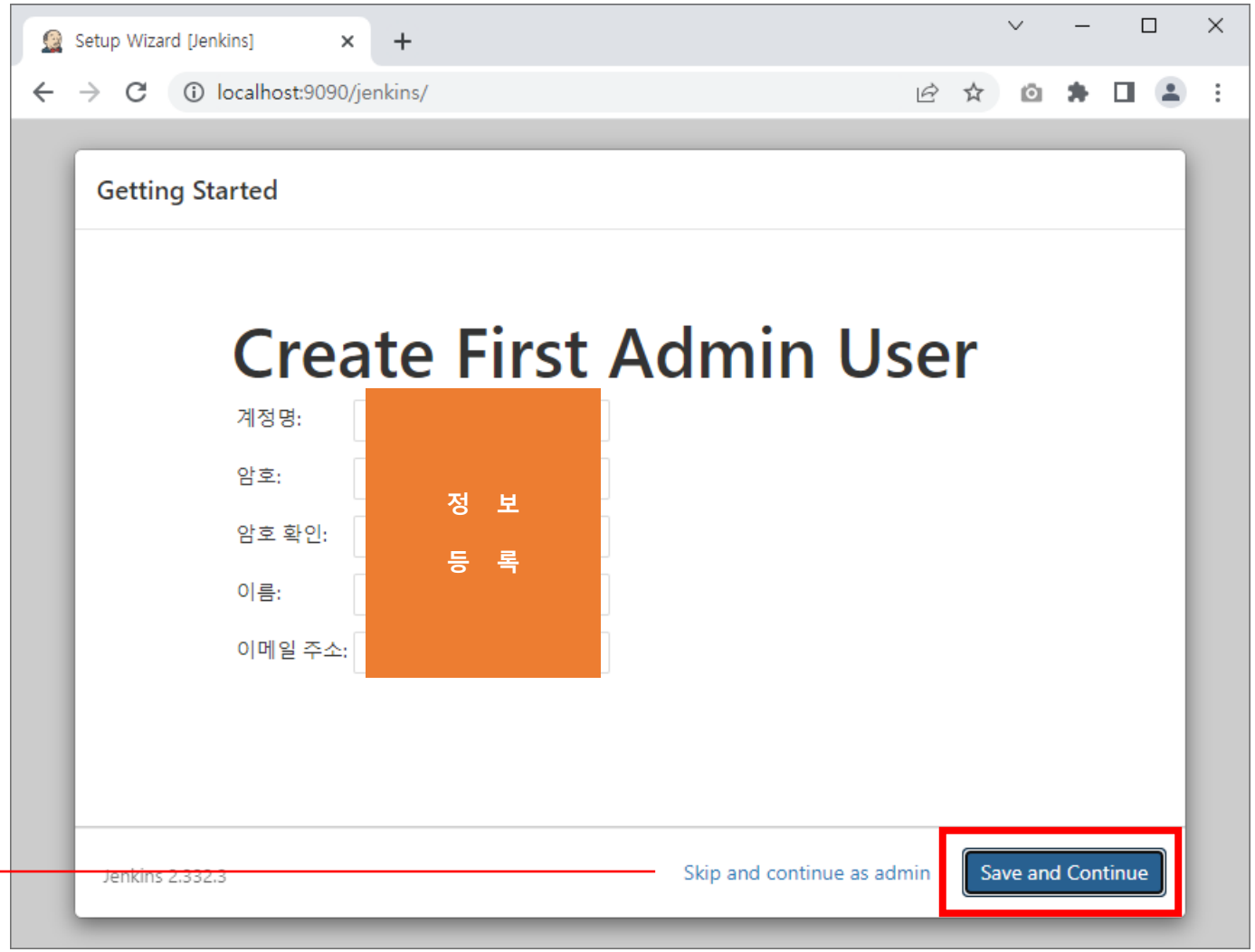
5) 젠킨스 플러그인 설치
Install suggested plugins



Jenkins

7. Jenkins Start

6) 젠킨스 관리자 등록
관리자 등록



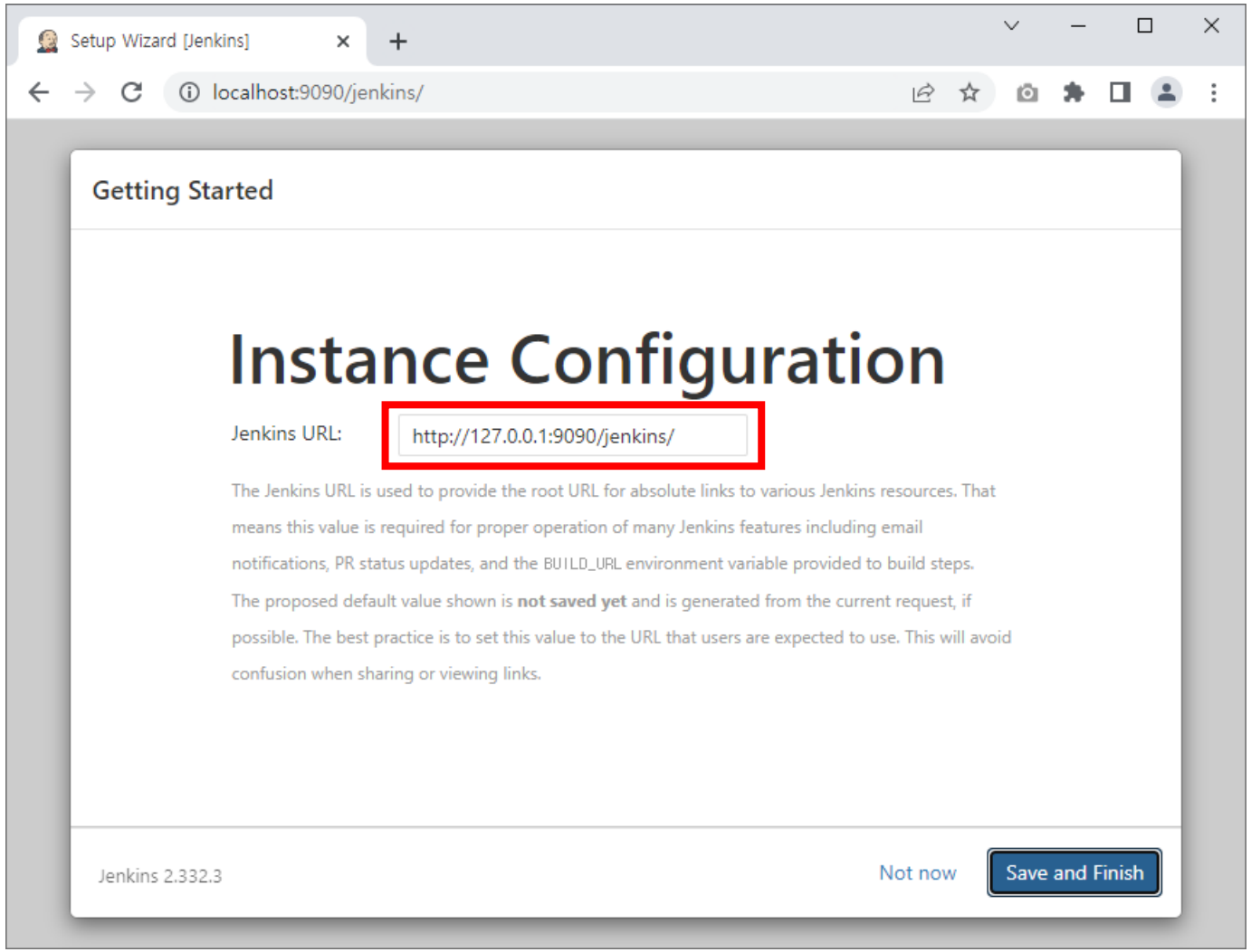
* Skip and continue as admin
관리자 등록을 생략하는 경우 admin 계정과 초기비밀번호를 이용해서 계속 로그인 가능

Jenkins

7. Jenkins Start

7) 젠킨스 주소 등록
http://127.0.0.1:9090/jenkins/

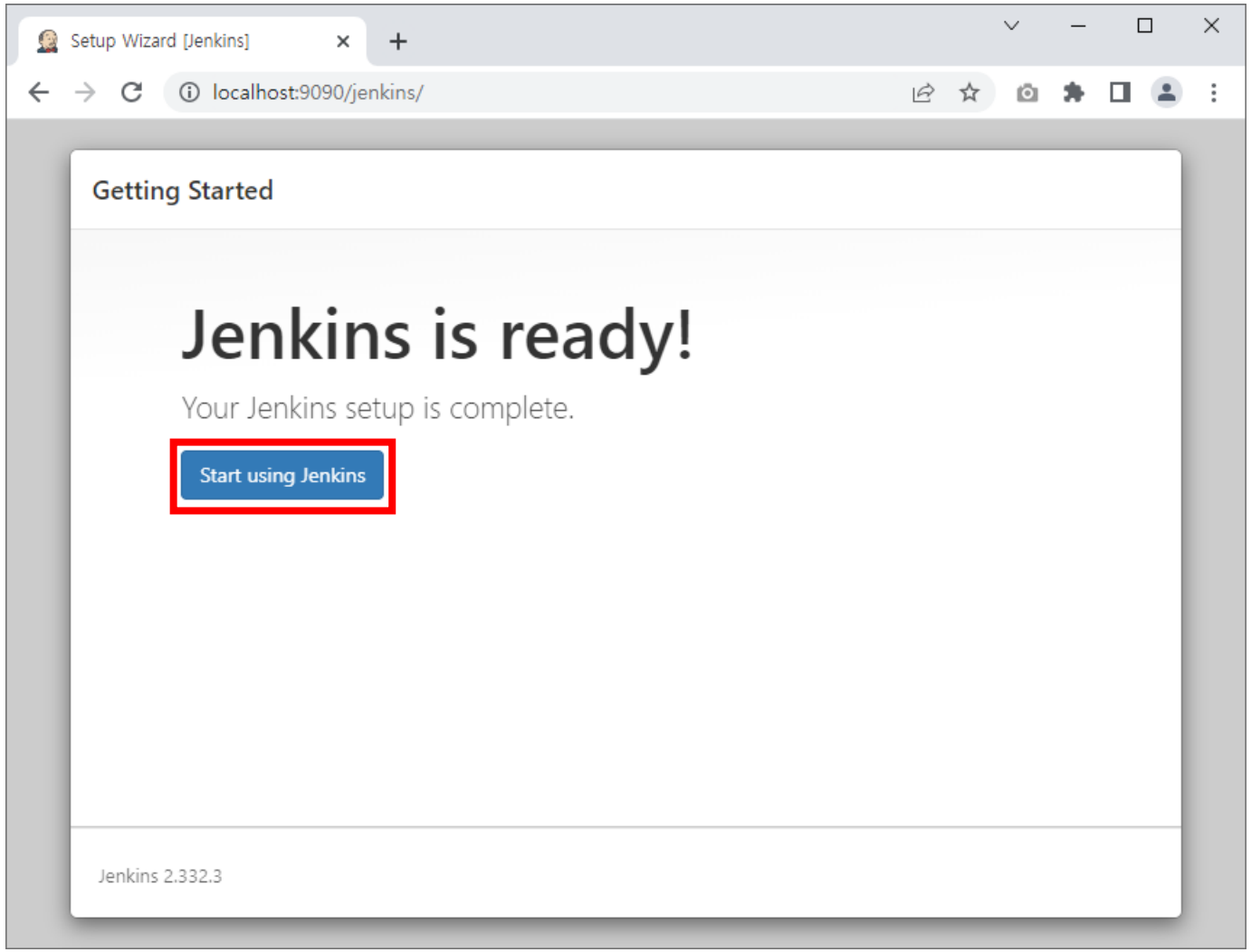
* 젠킨스 주소는 localhost 사용 불가



Jenkins

7. Jenkins Start

8) 젠킨스 준비 완료

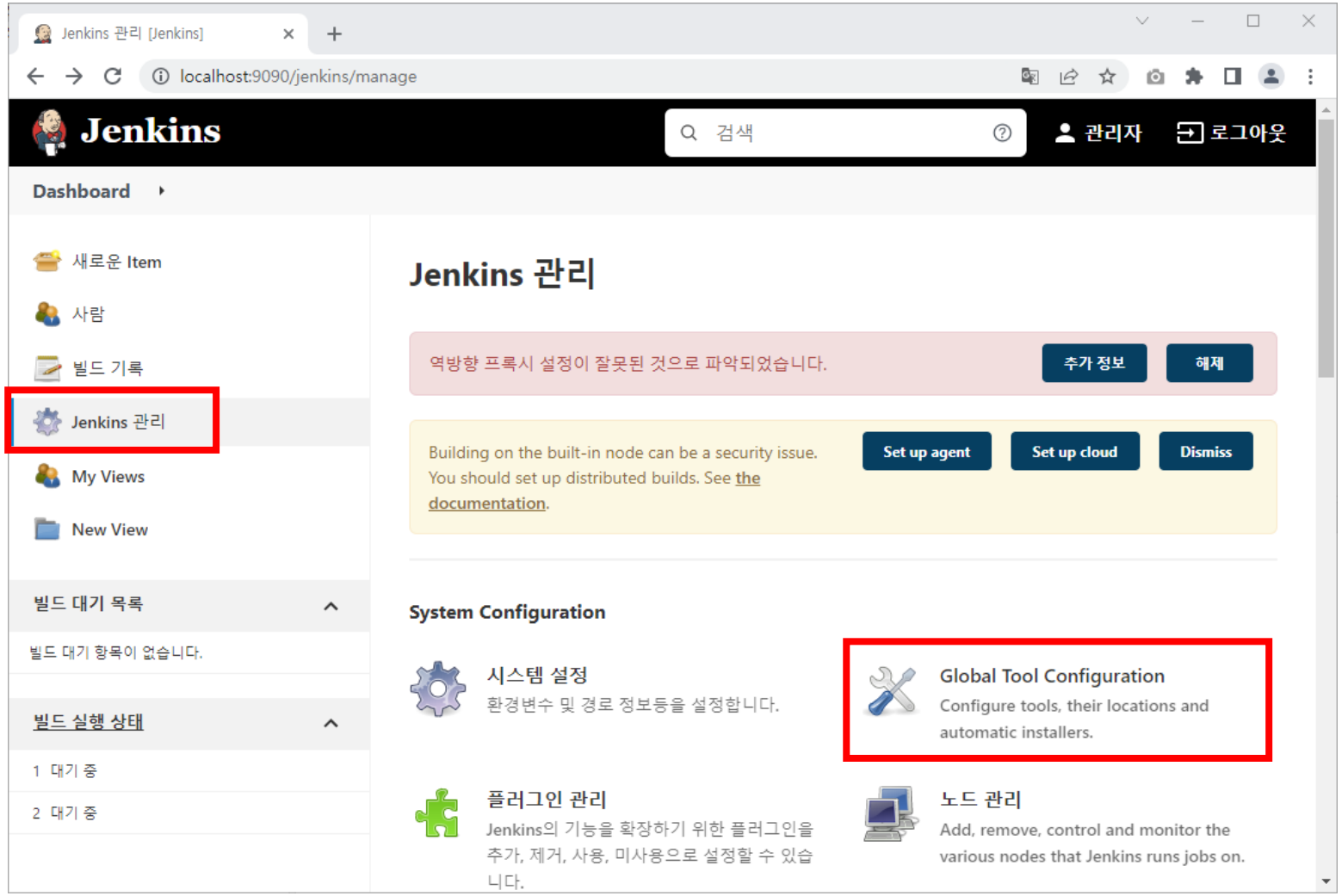


Jenkins

8. Jenkins Setting

1) [Jenkins 관리]

   ㄴ [Global Tool Configuration]



Jenkins

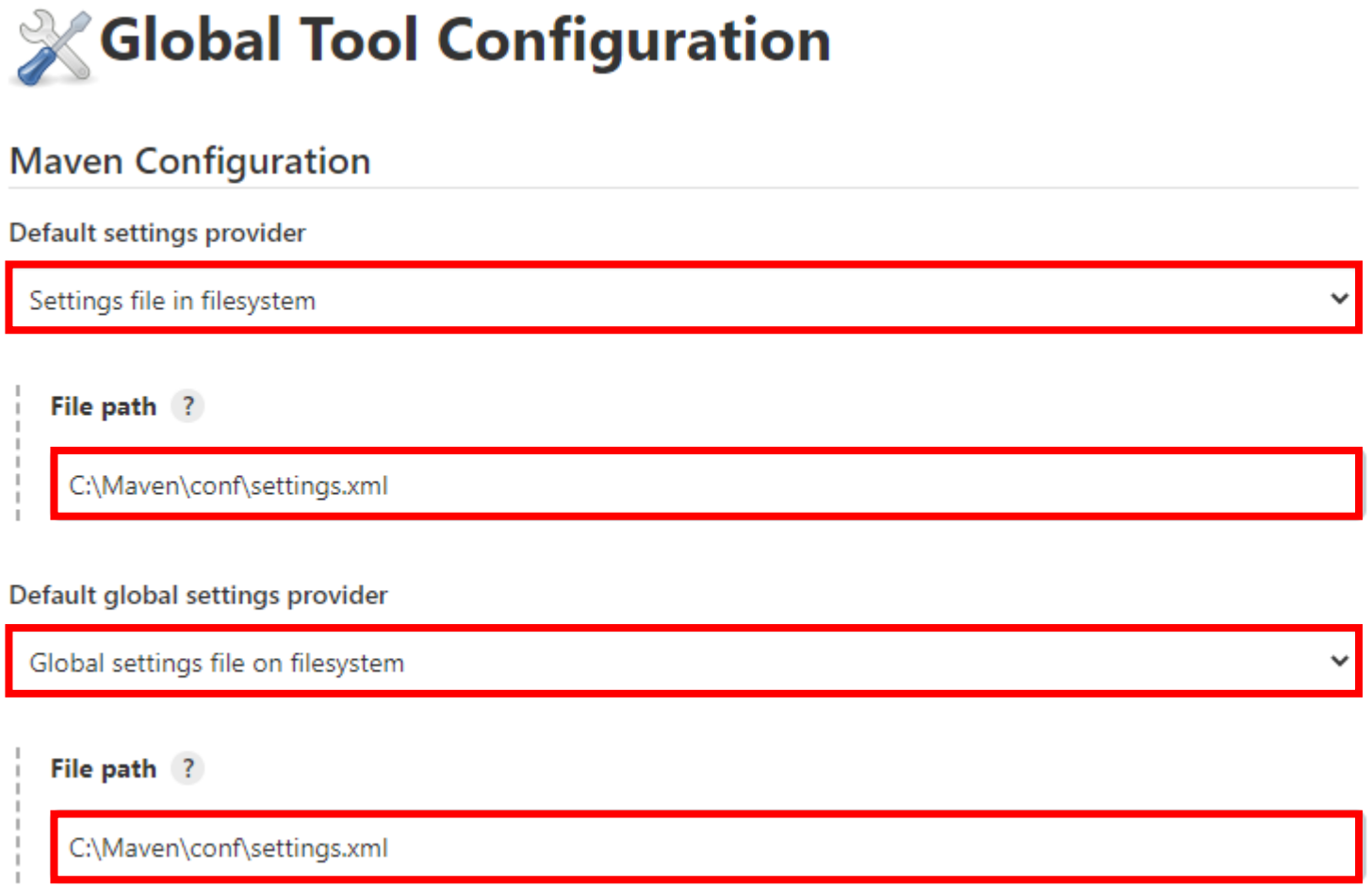
8. Jenkins Setting

1) [Jenkins 관리]

   ㄴ [Global Tool Configuration]

   ㄴ [Maven Configuration]

- Maven 환경설정파일을 등록하는 작업임



Jenkins

8. Jenkins Setting

1) **[Jenkins 관리]**

   ㄴ **[Global Tool Configuration]**

   ㄴ **[JDK]**

- Java가 설치된 경로를 등록하는 작업임

Jenkins

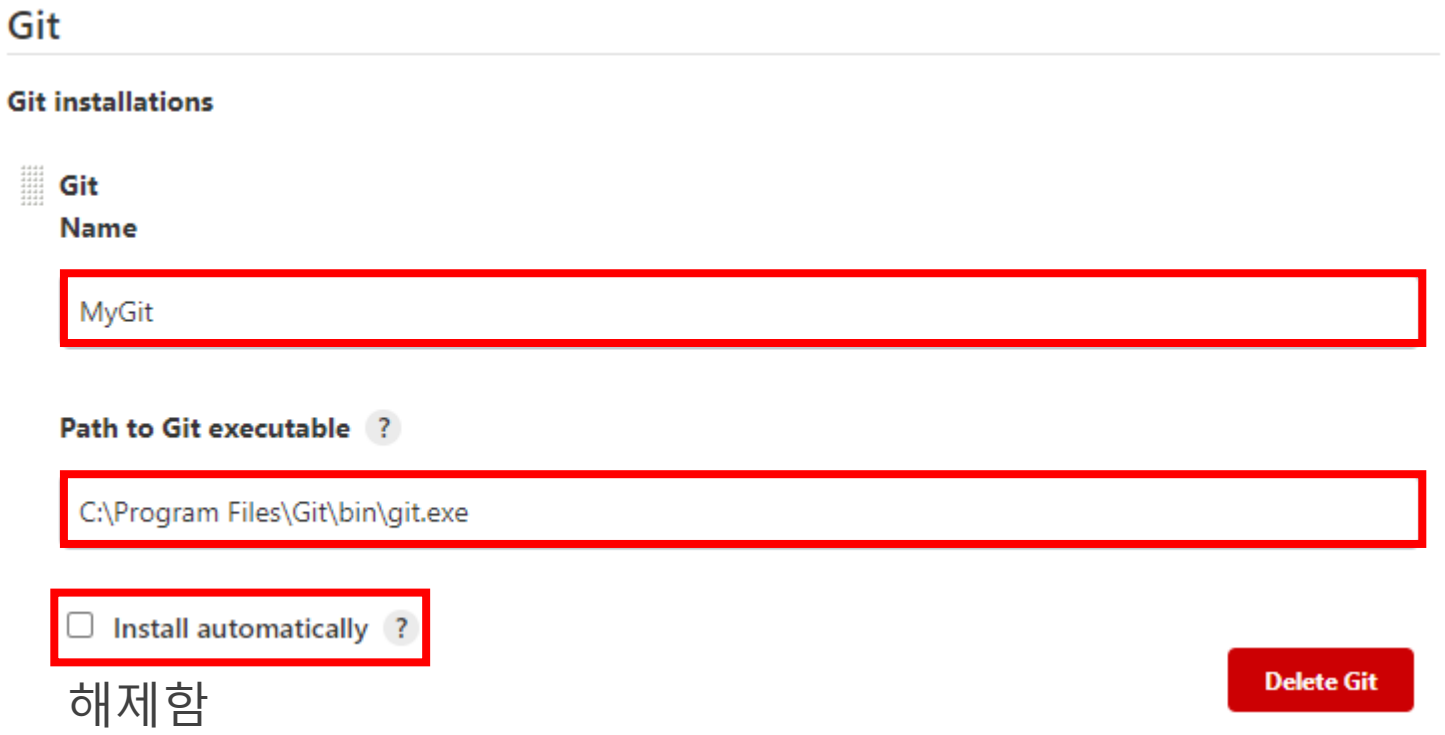
8. Jenkins Setting

1) [Jenkins 관리]

   ㄴ [Global Tool Configuration]

   ㄴ [Git]

- Git이 설치된 경로를 등록하는 작업임



해제함

Jenkins

8. Jenkins Setting

1) **[Jenkins 관리]**

   ㄴ **[Global Tool Configuration]**

   ㄴ **[Maven]**

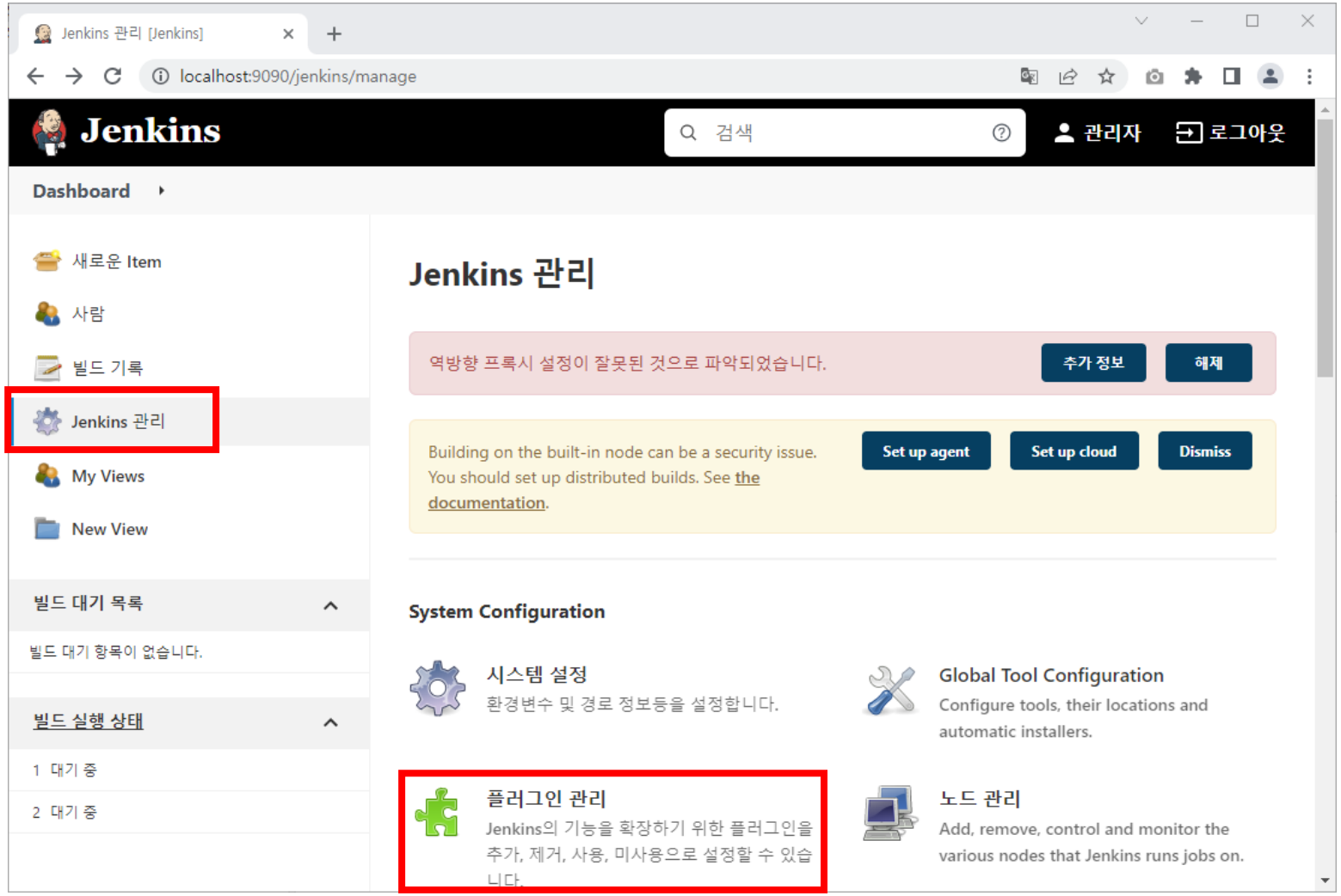
- Maven이 설치된 경로를 등록하는 작업임

Jenkins

8. Jenkins Setting

1) [Jenkins 관리]

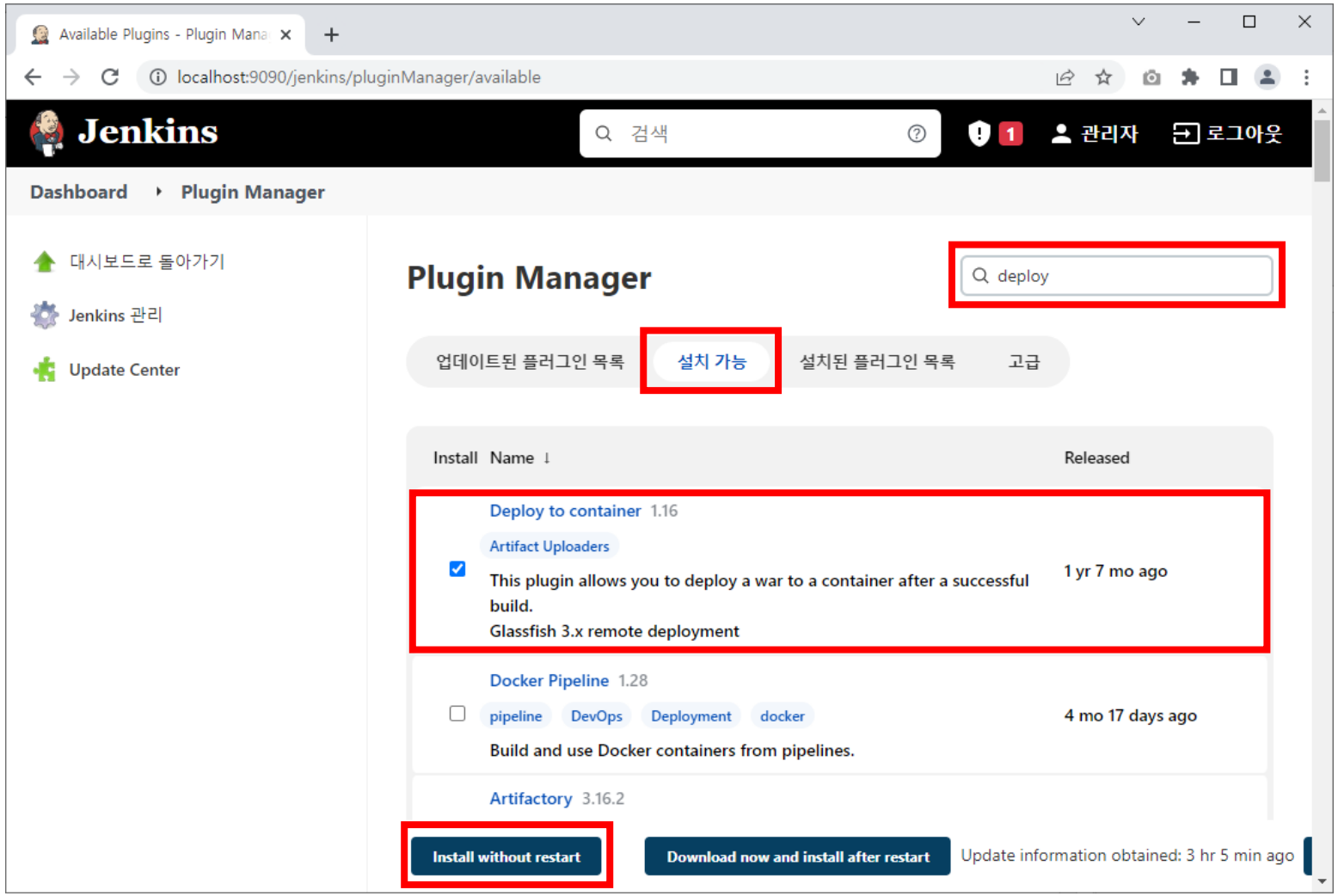
   ㄴ [플러그인 관리]



Jenkins

8. Jenkins Setting

1) [Jenkins 관리]
   - [플러그인 관리]
   - 설치 가능
   - deploy 검색
   - Deploy to container 선택
   - Install without restart

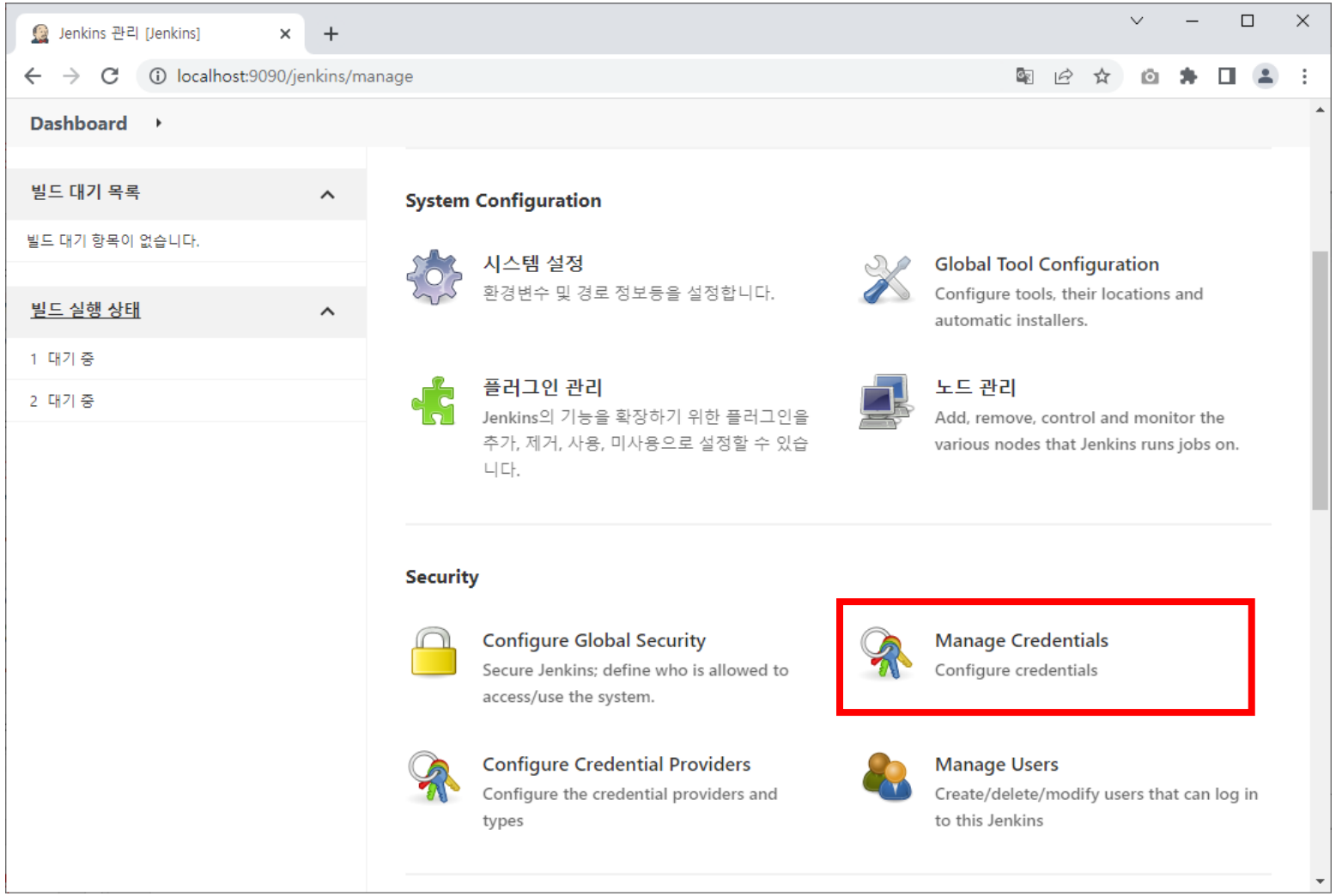


Jenkins

8. Jenkins Setting

1) [Jenkins 관리]

   ㄴ [Manage Credentials]



Jenkins

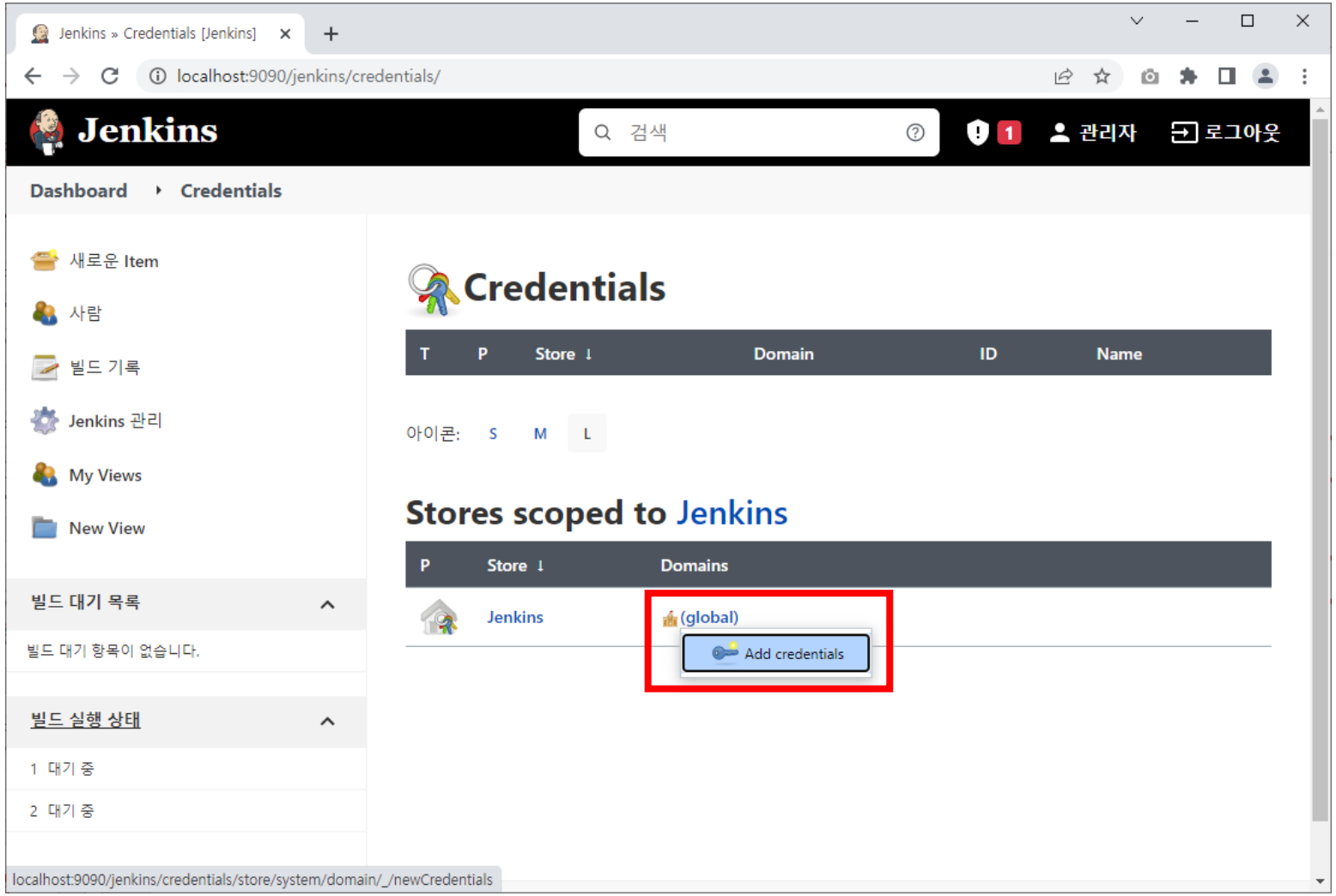
8. Jenkins Setting

1) [Jenkins 관리]

   ㄴ [Manage Credentials]

   ㄴ [Add credentials]

   총 2번 수행해야 함



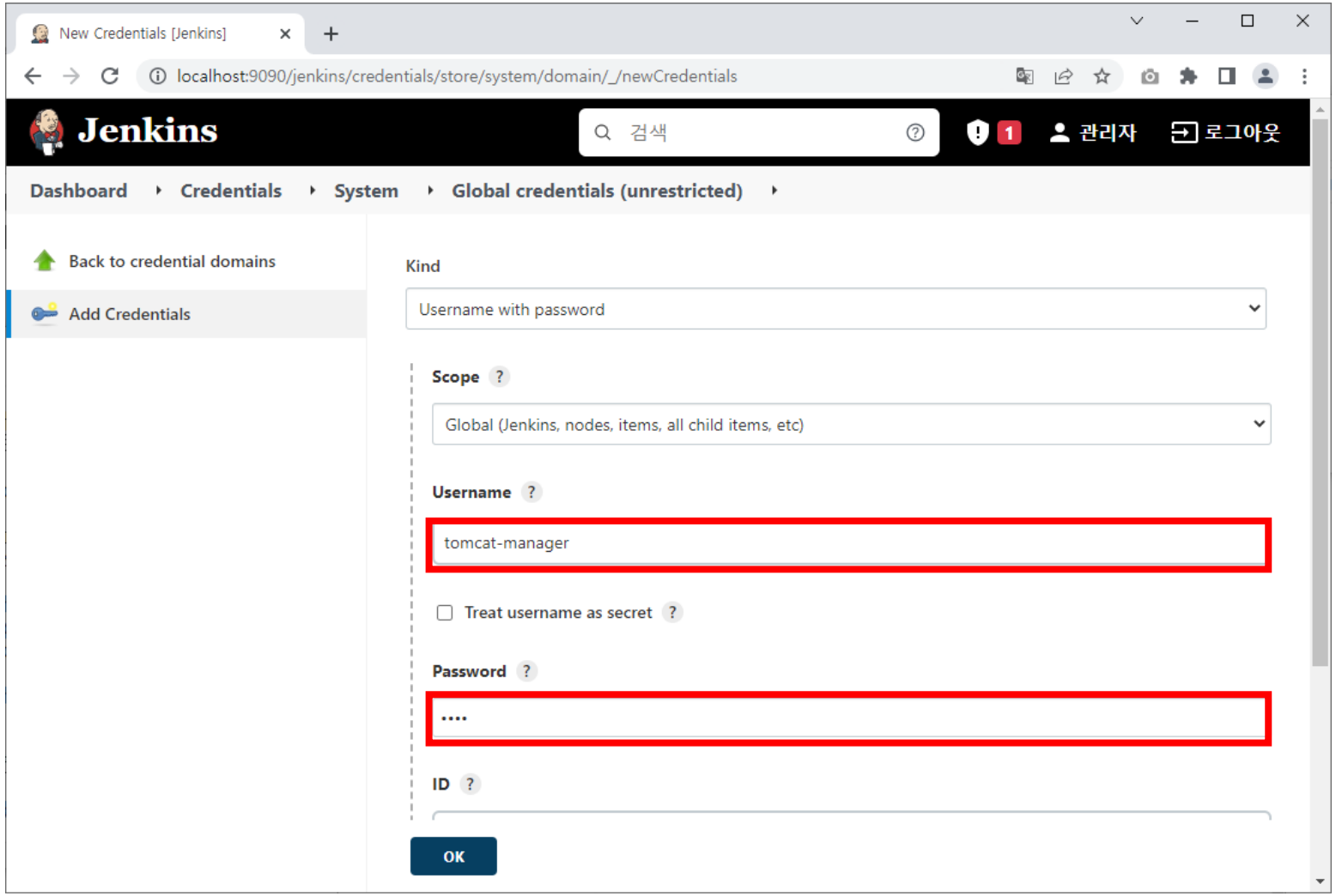
Jenkins

8. Jenkins Setting

(1) 톰캣 관리자의
Username과 Password 등록

Username : tomcat-manager
Password : 1111

* tomcat-users.xml 에 등록한 정보임

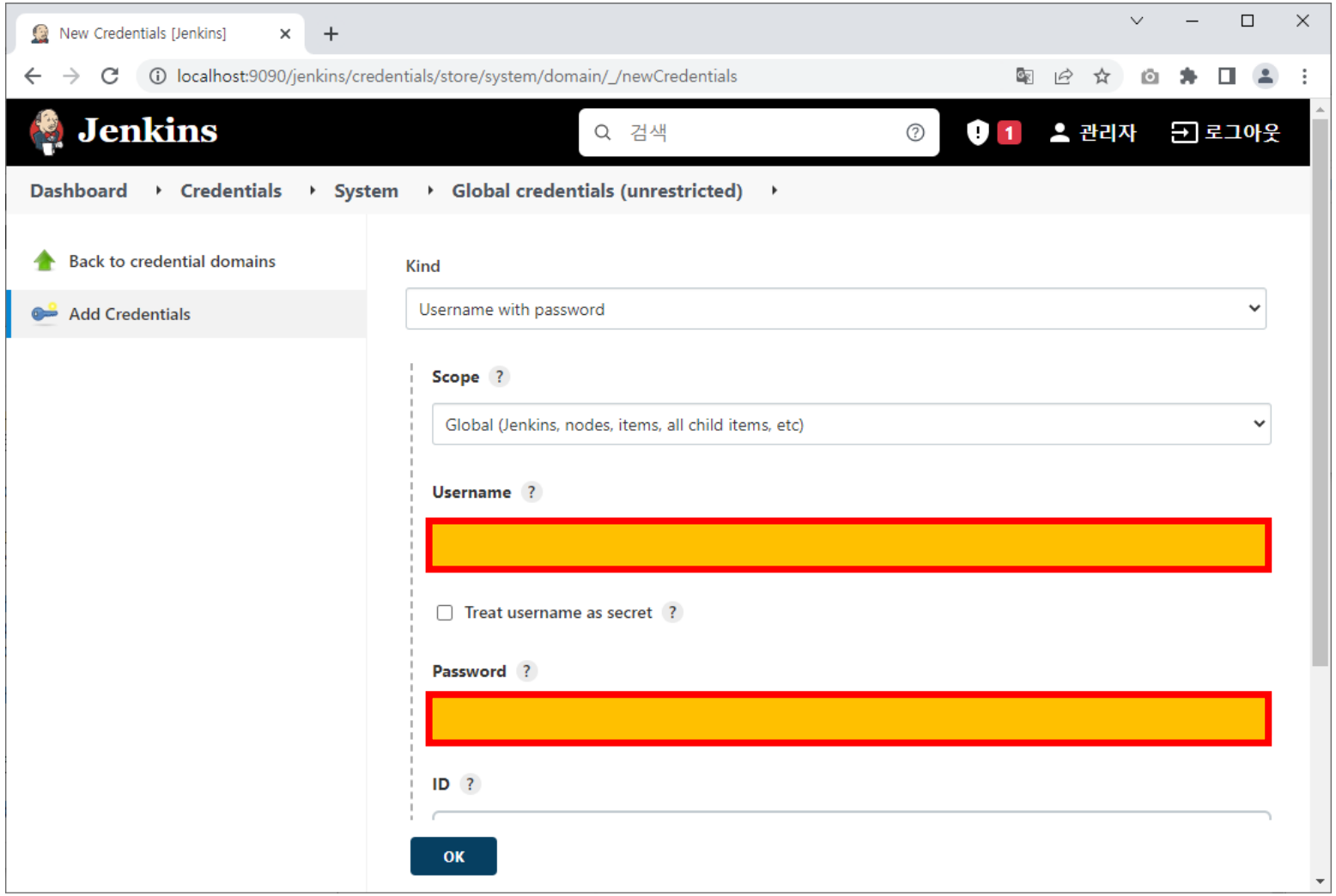


Jenkins

8. Jenkins Setting

(2) github 관리자의
Username과 Password 등록

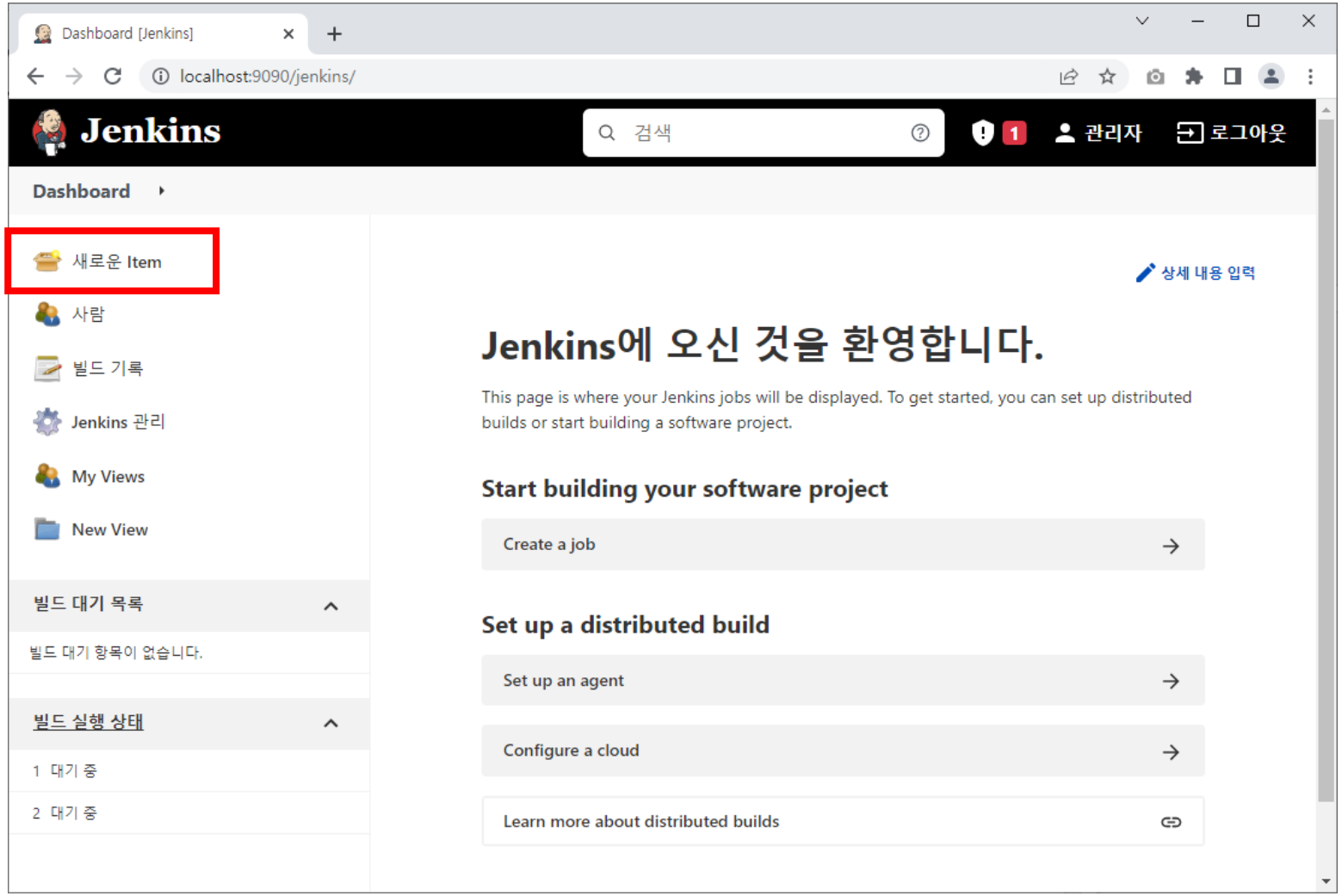
Username : 본인 github 아이디
Password : Personal access token 값



Jenkins

9. Project Setting

[새로운 Item]

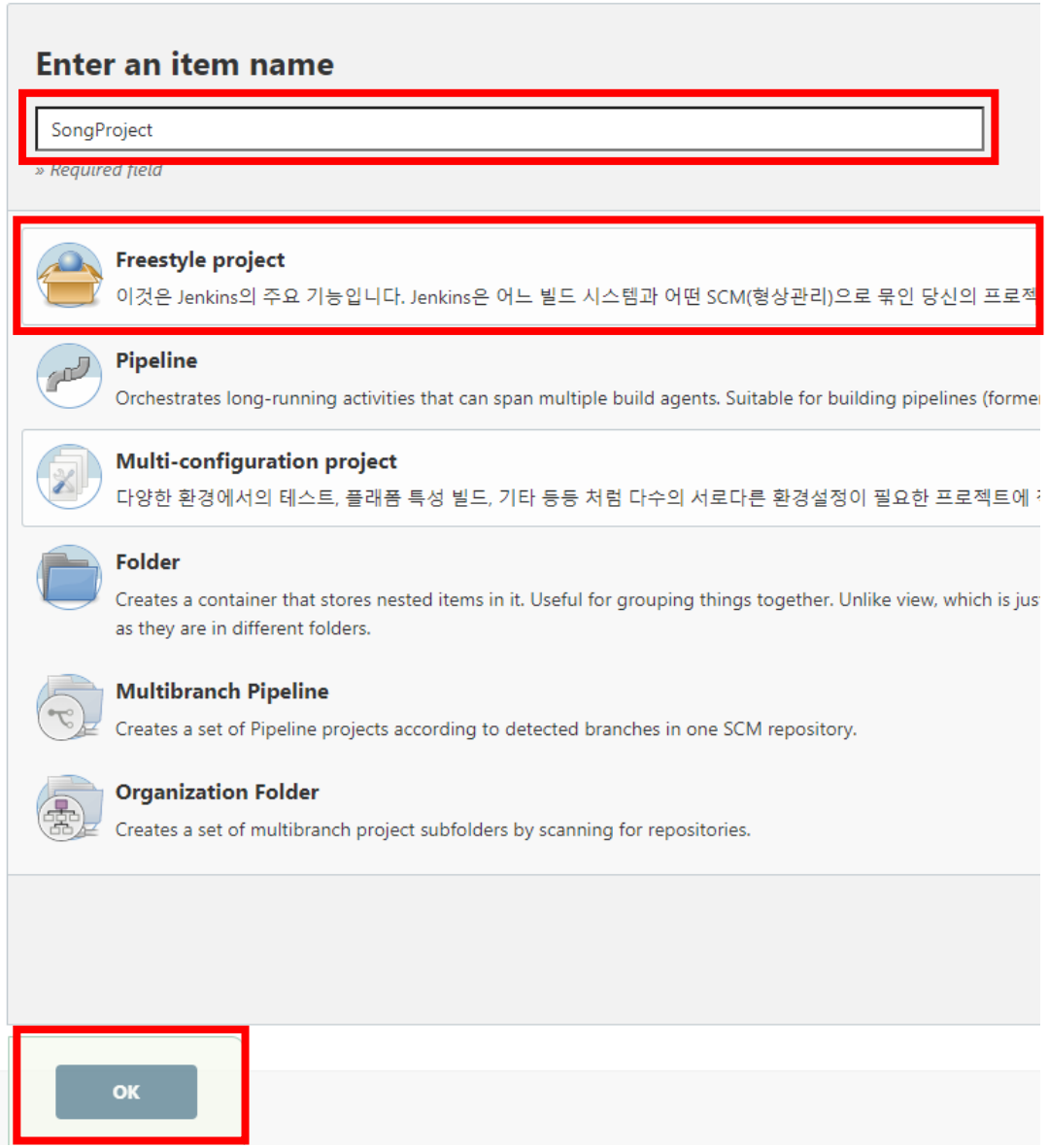


Jenkins

9. Project Setting

[프로젝트명] 입력

Freestyle project 선택

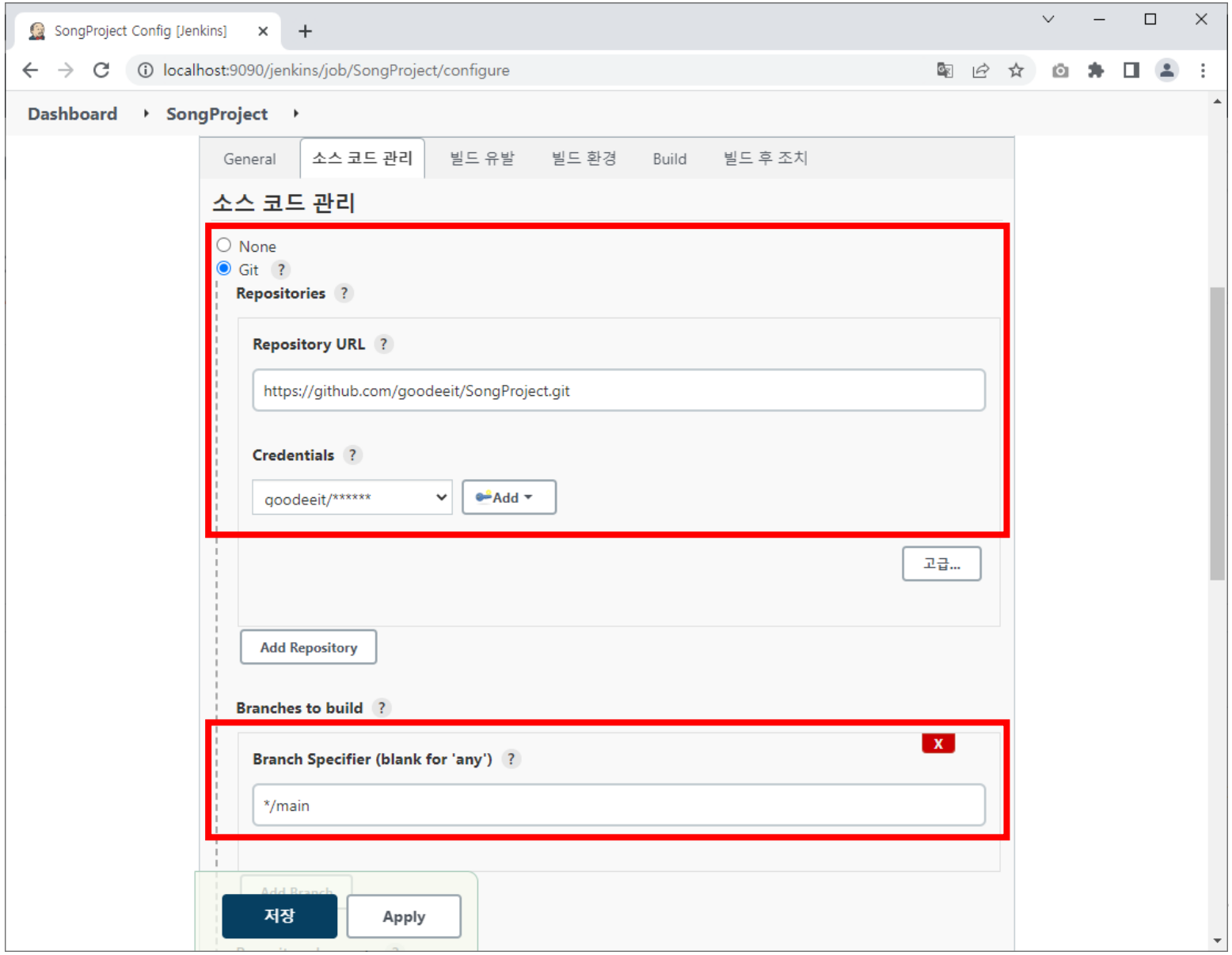


Jenkins

9. Project Setting

[소스 코드 관리]

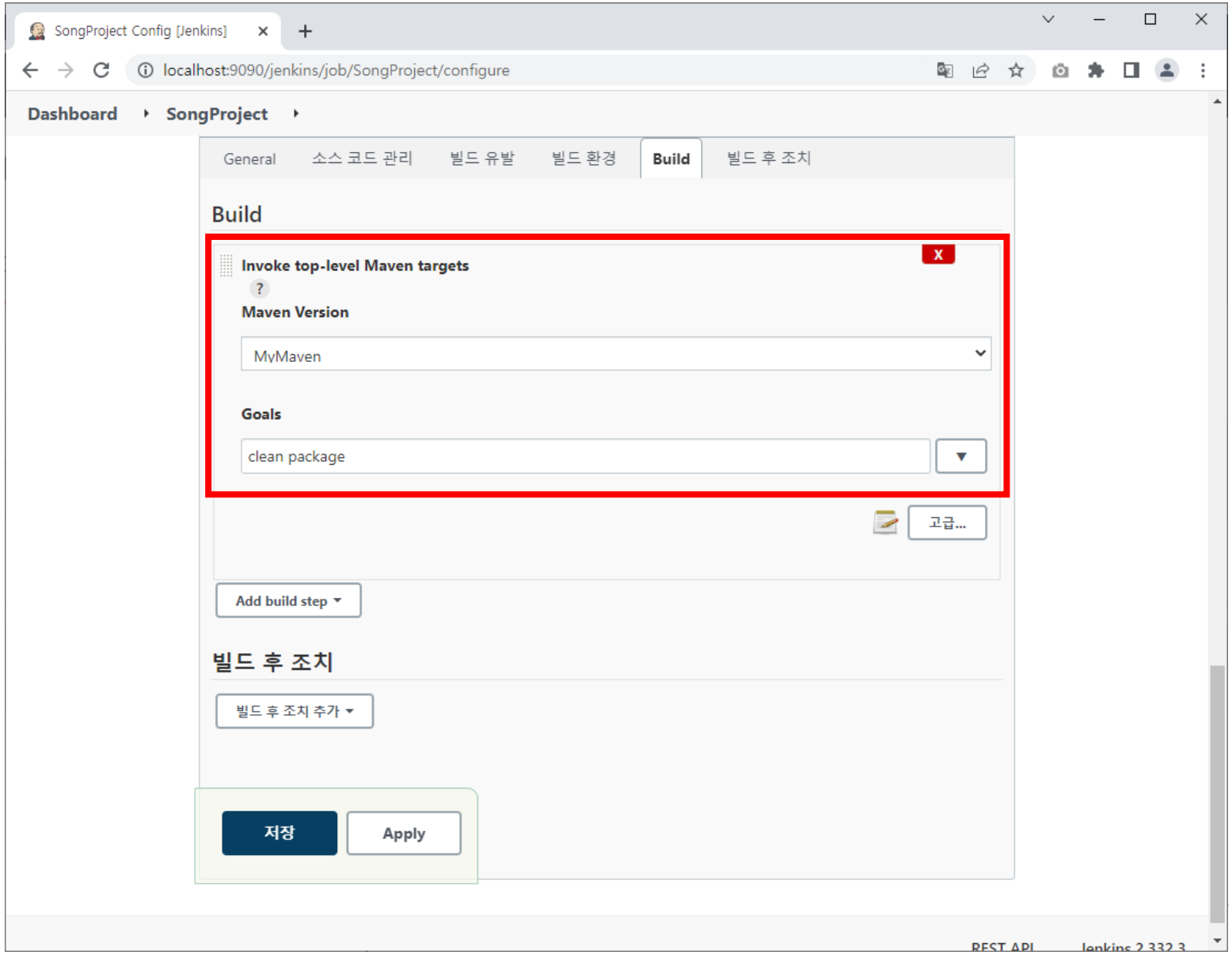
- **[Git]**
  - **[Repository URL]**
    애플리케이션github주소
  - **[Credentials]**
    github관리자계정
  - **[Branch Specifier]**
    .main



9. Project Setting

[Build]

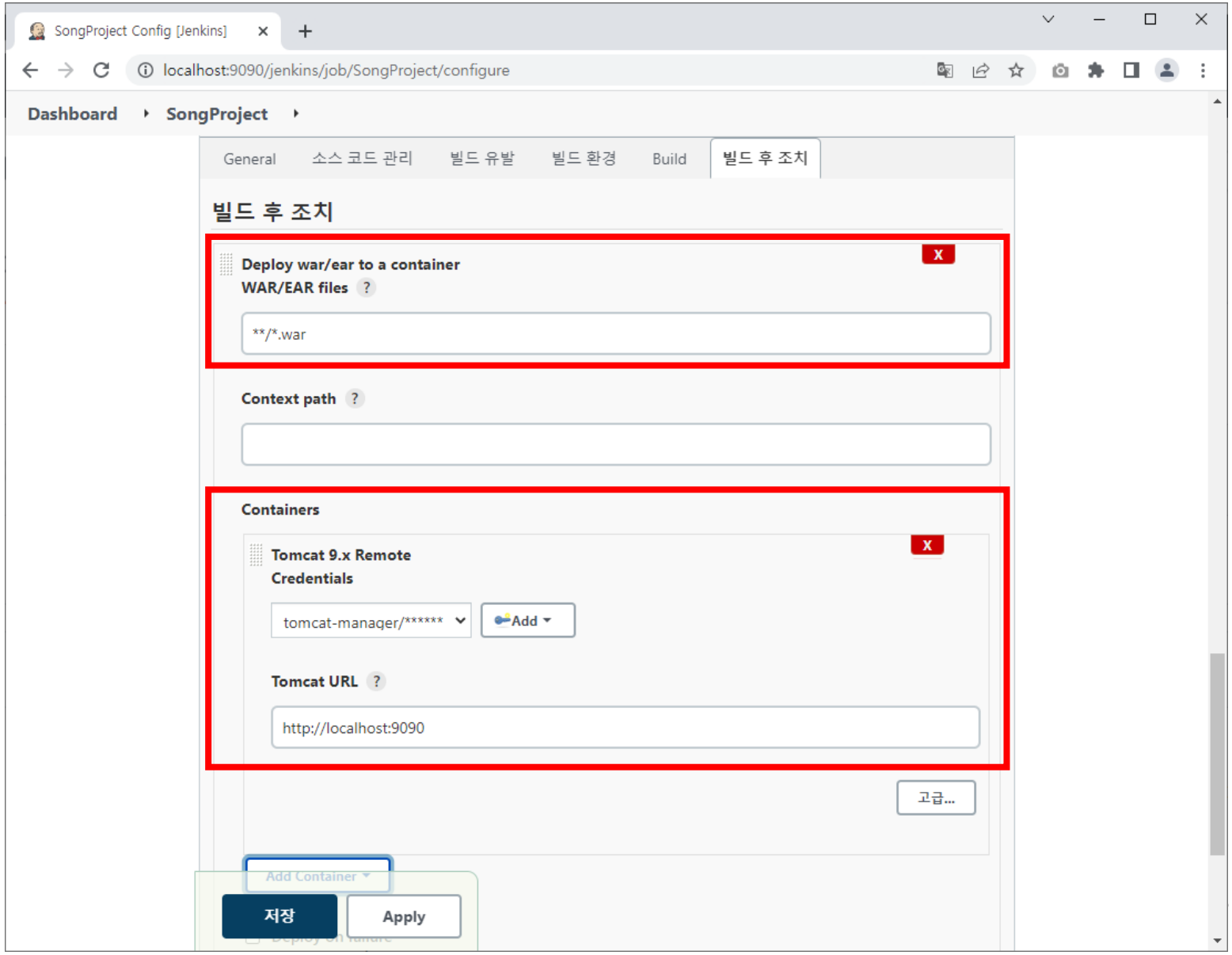
- **[Maven Version]**
  MyMaven
- **[Goals]**
  clean package



9. Project Setting

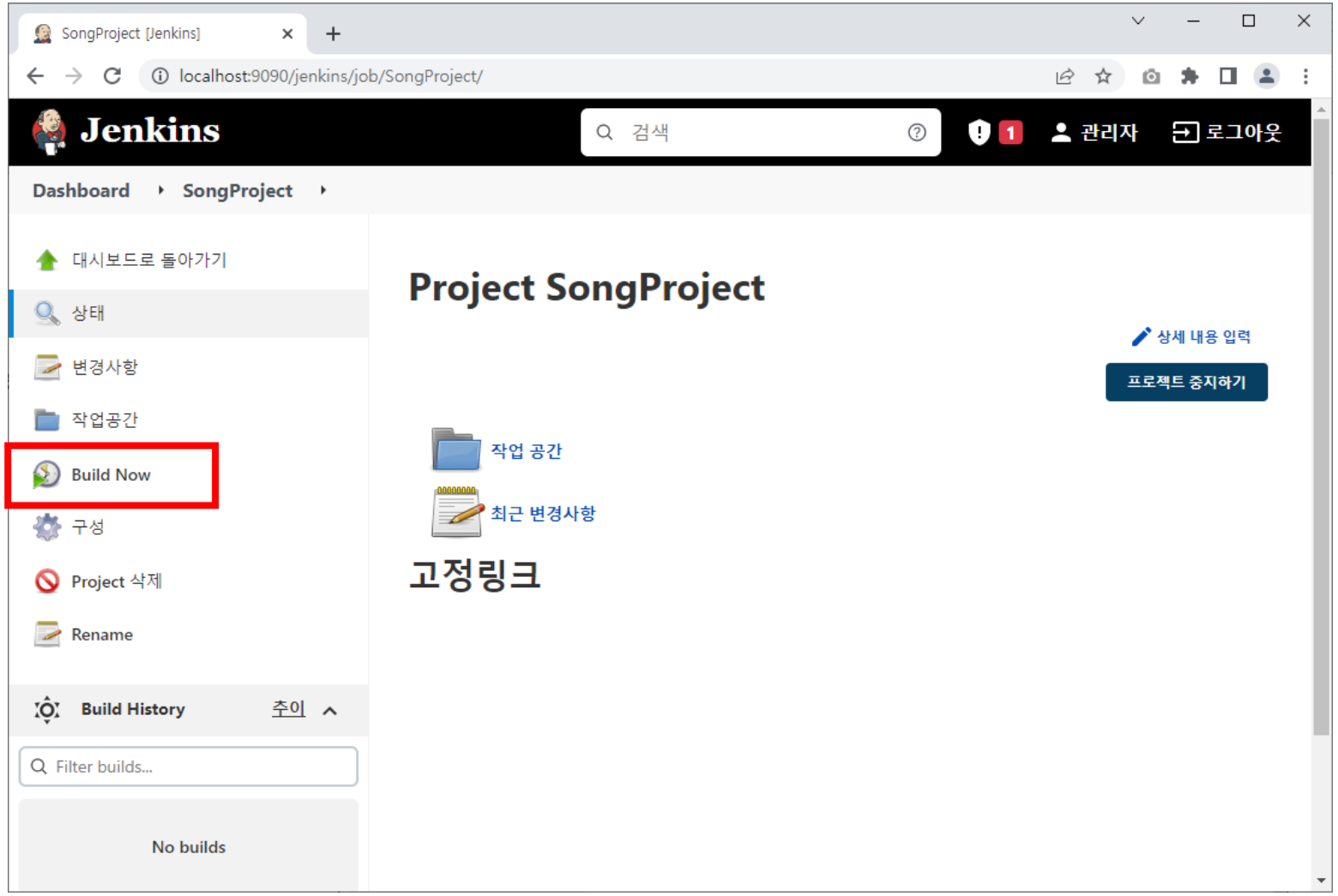
[빌드 후 조치]

- [Deploy war/ear to a container]
  - [WAR/EAR files]
    **/*.war
- [Containers]
  - [Tomcat 9.x Remote]
  - [Credentials]
    Tomcat관리자계정
  - [Tomcat URL]
    http://localhost:9090



10. Project Build & Deploy

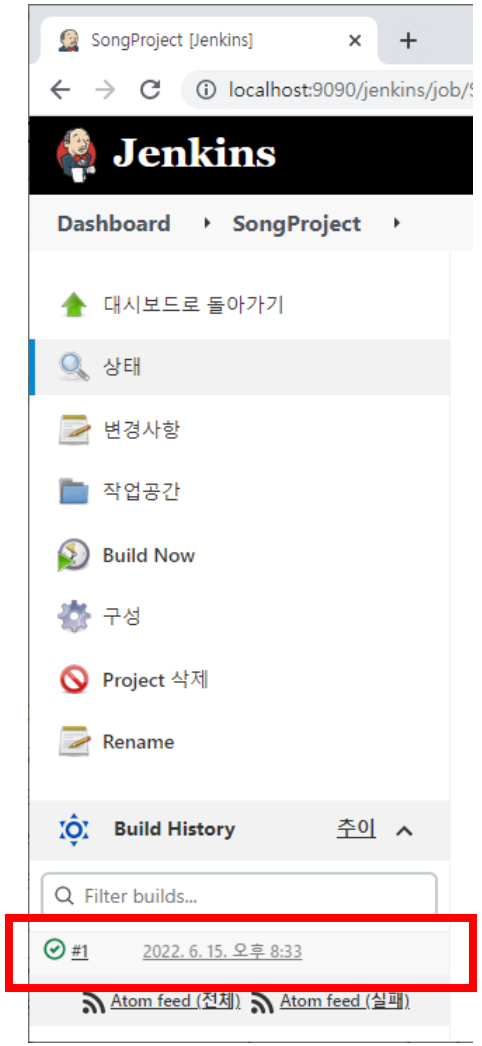
[Build Now]



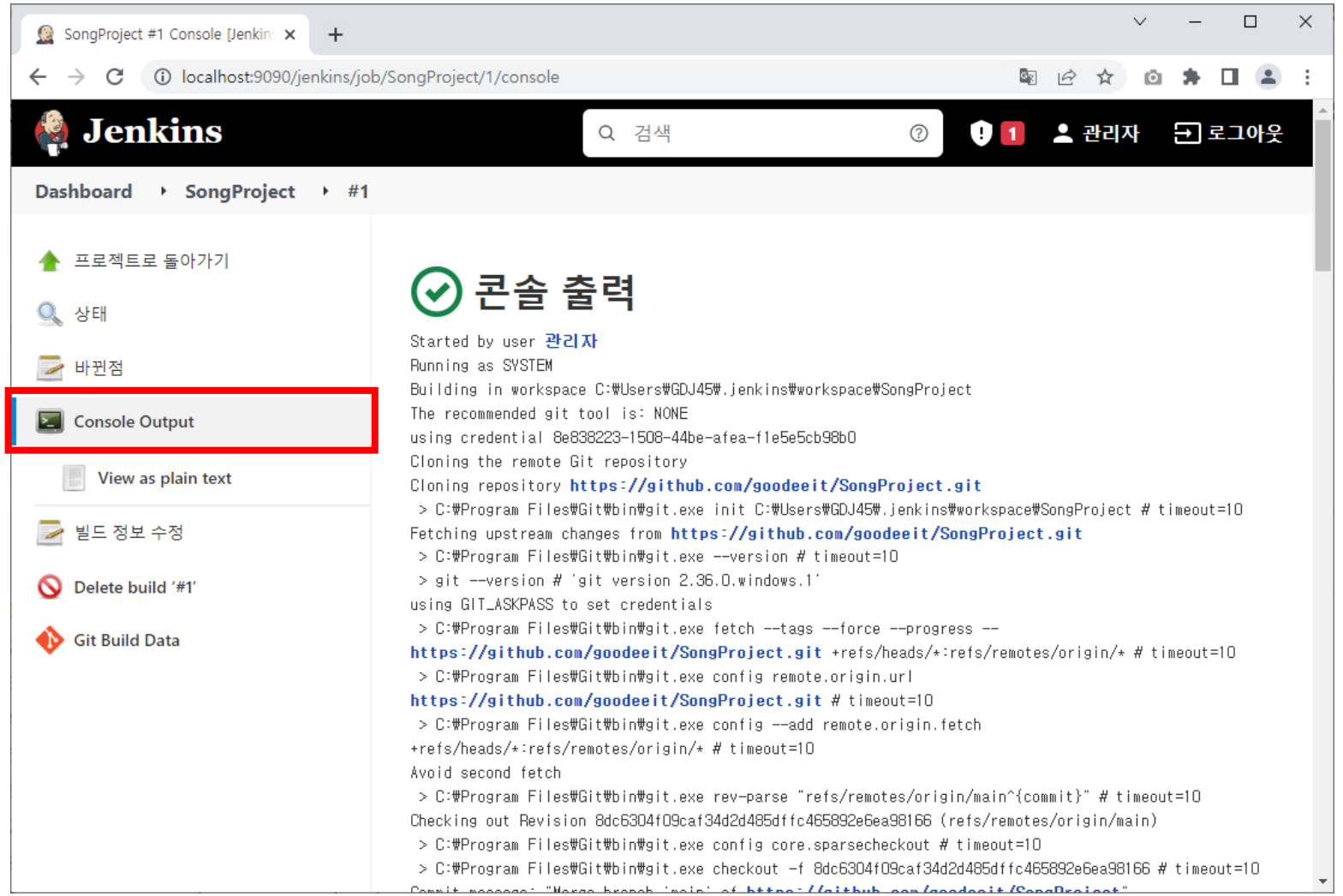
Jenkins

10. Project Build & Deploy

[Build History]



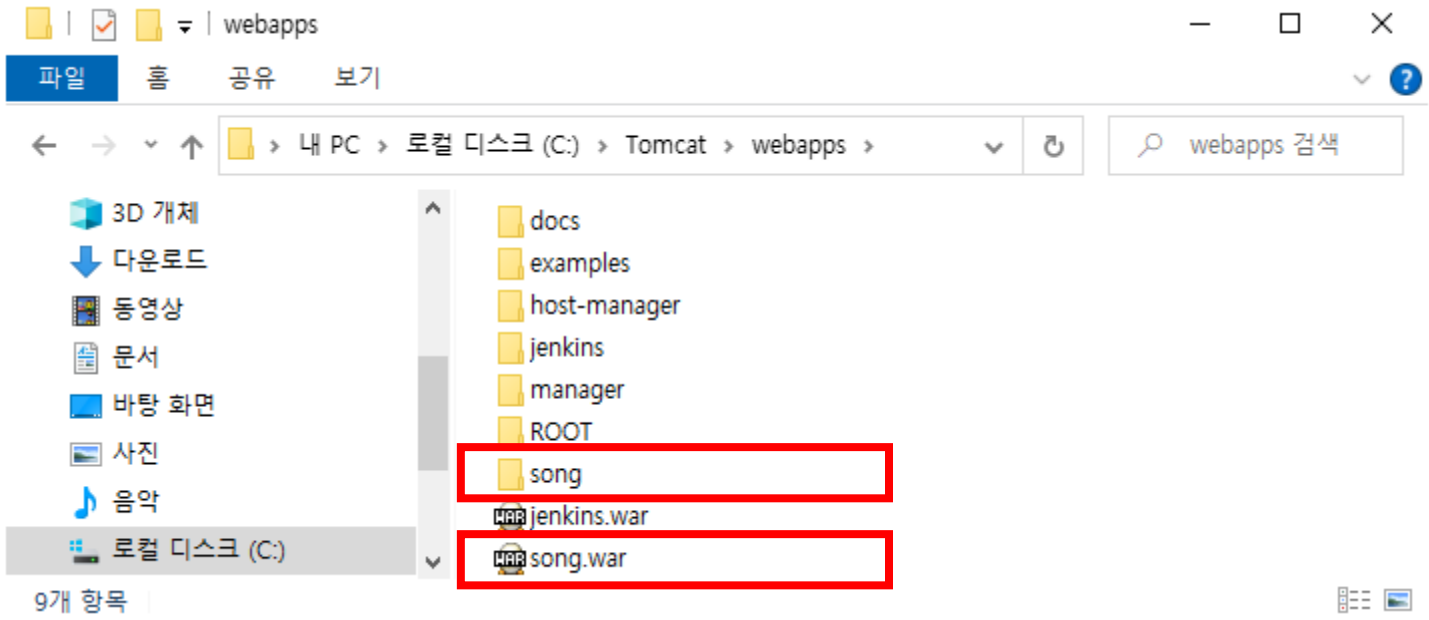
[Console Output]



Jenkins

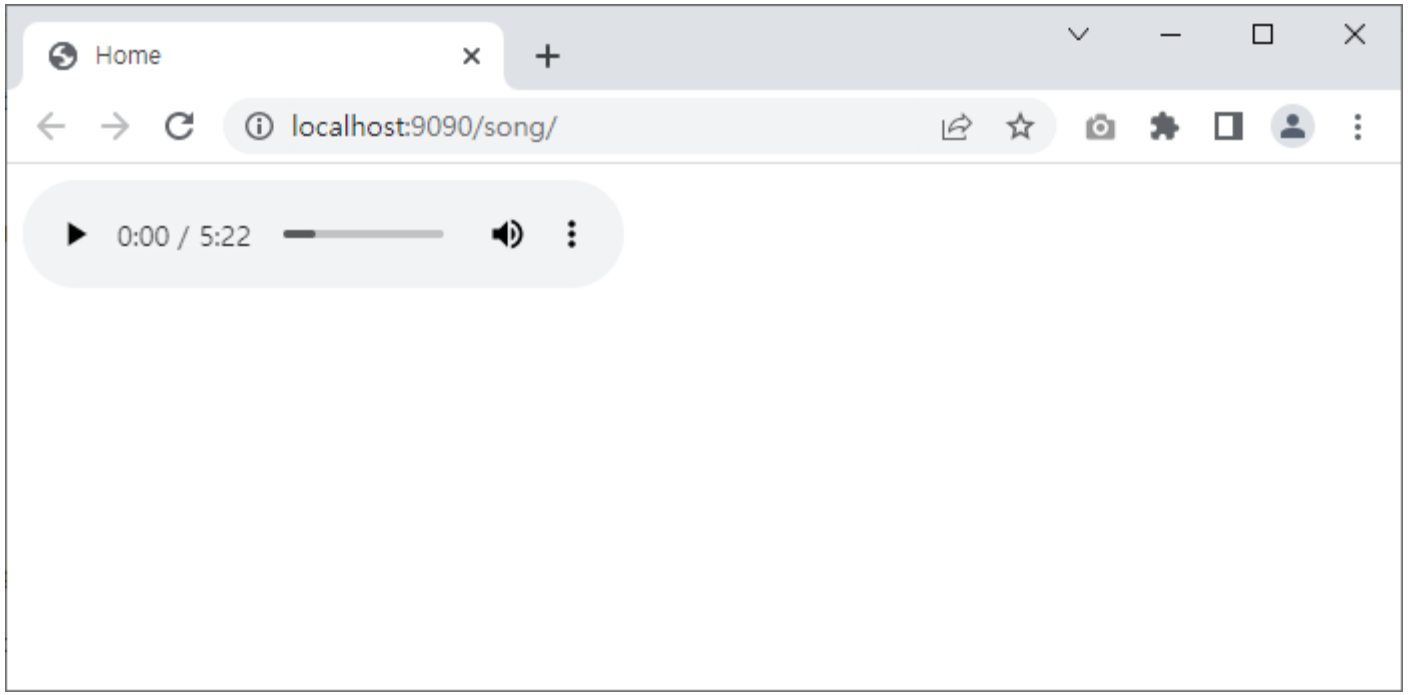
10. Project Build & Deploy

톰캣에 배포되었는지 확인



Jenkins

10. Project Build & Deploy



Jenkins