



젠킨스 설정하기

# 1. Jenkins란?



## 1) CI

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

## 2) Jenkins

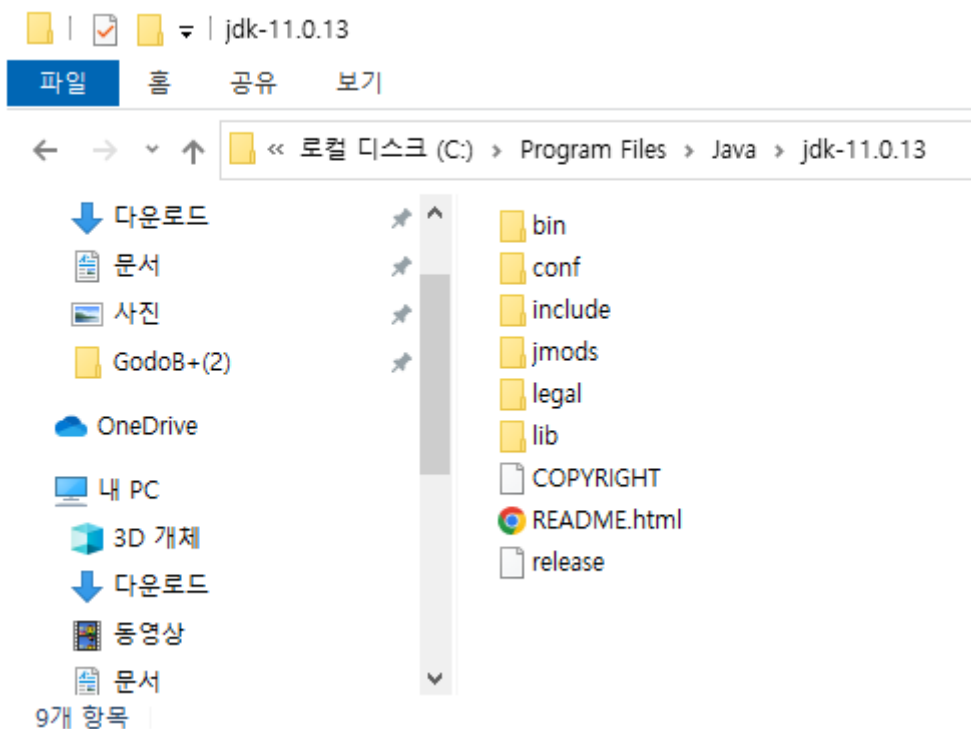
지속적 통합 서비스를 제공하는 툴(tool)로 허드슨(hudson) 프로젝트에서 시작

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

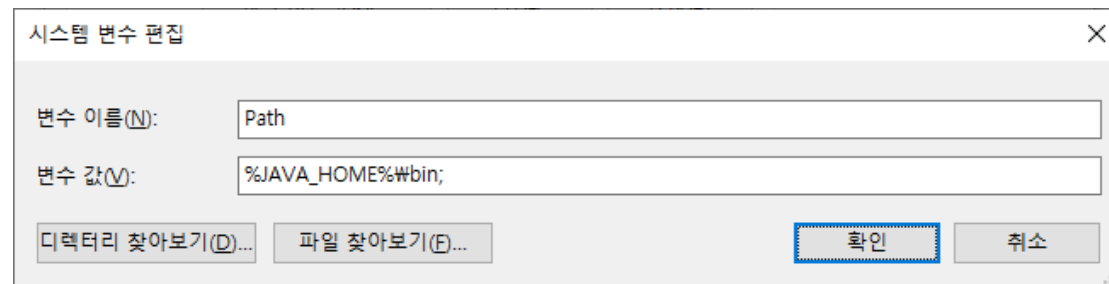
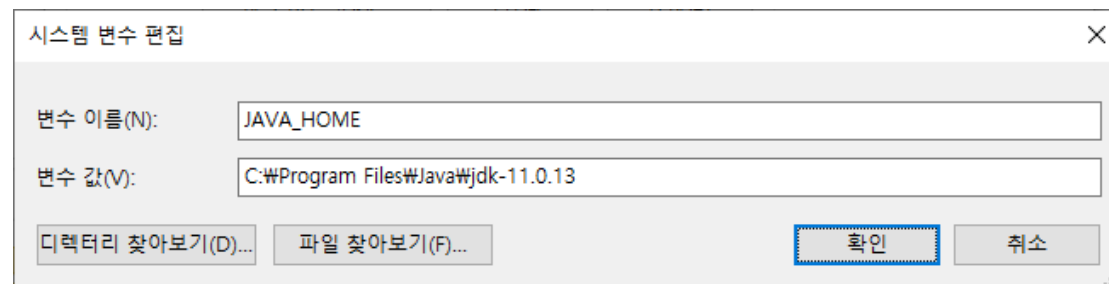
# 2. Java

## 1) 자바 설치 경로 확인

C:\Program Files\Java\jdk-11.0.13



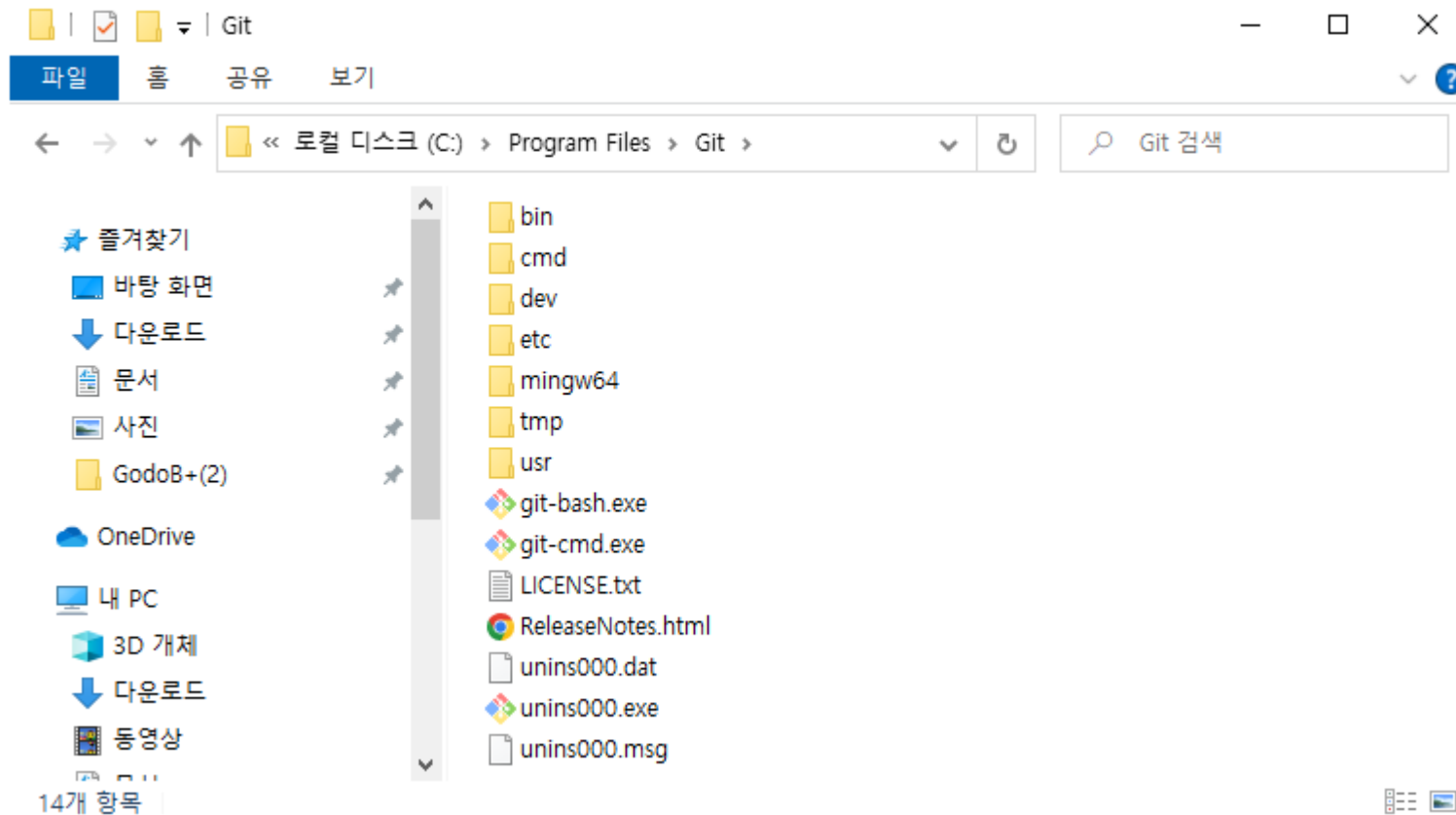
## 2) 자바 환경변수 제어판 - 시스템



# 3. Git

## 1) Git 설치 경로 확인

C:\Program Files\Git



# 3. Git

## 2) Personal Access Token

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

The screenshot illustrates the steps to generate a Personal Access Token on GitHub. Red boxes and arrows highlight the navigation path:

- Profile icon**: Located in the top right corner of the GitHub interface.
- Settings**: Located in the left sidebar menu.
- Developer settings**: Located in the left sidebar menu, under the 'Settings' section.
- Personal access tokens**: Located in the 'Developer settings' section.
- Generate new token**: A button located next to the 'Personal access tokens' section.
- Expiration**: A dropdown menu showing '30 days'.
- Select scopes**: A section where you can choose the permissions for the token. The 'repo' scope is selected.
- Generate token**: A green button located at the bottom right of the 'Select scopes' section.

Additional text and elements visible in the screenshot:

- Personal access tokens**: Tokens you have generated that can be used to access the [GitHub API](#).
- Generate new token**: A button located next to the 'Personal access tokens' section.
- Revoke all**: A button located next to the 'Generate new token' button.
- Expiration**: A dropdown menu showing '30 days'. The token will expire on Sat, Jul 16 2022.
- Select scopes**: Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)
- repo**: A checkbox that is checked, indicating that the token will have full control of private repositories.
- Generate token**: A green button located at the bottom right of the 'Select scopes' section.
- Cancel**: A button located next to the 'Generate token' button.

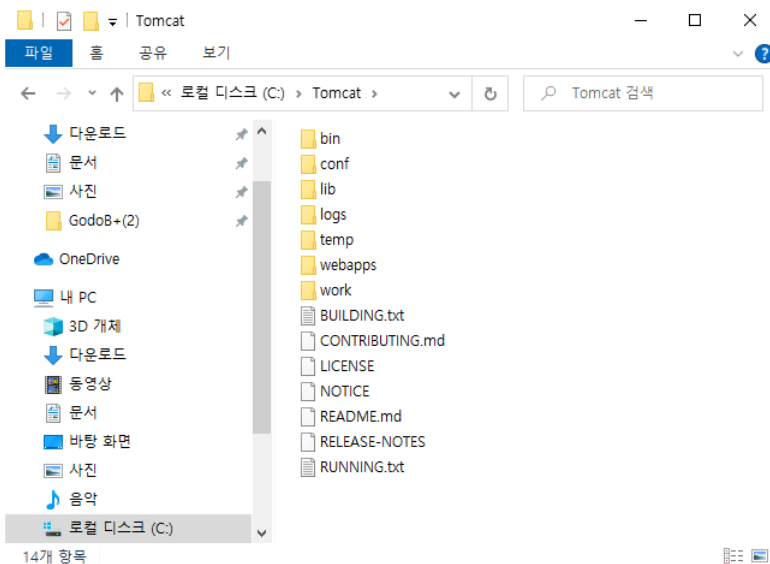
※ 발급 후 백업해야 함!

# 4. Tomcat

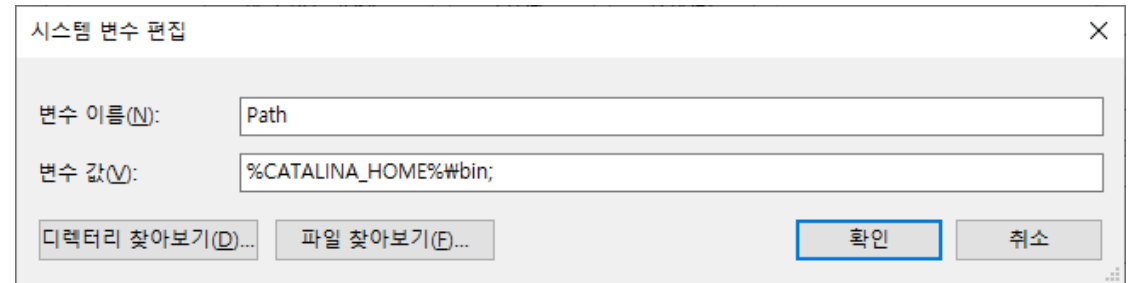
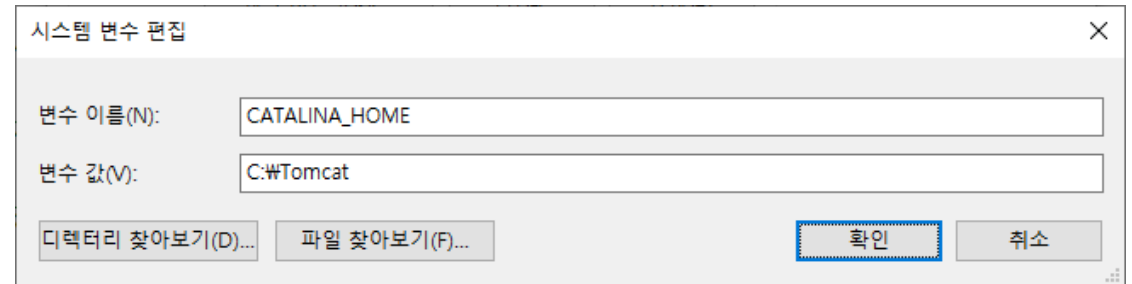
## 1) 톰캣 다운로드 (버전 9.0.x)

<https://tomcat.apache.org/>

## 2) 톰캣 설치 C:\Tomcat



## 3) 톰캣 환경변수 제어판 - 시스템



# 4. Tomcat

## 4) 포트번호 수정 (Oracle DB인 경우)

C:\Tomcat\conf\server.xml



```
server.xml - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html
Java AJP  Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="9090" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
-->
```

태그 내용 대신 주석을 수정하지 않도록 주의!

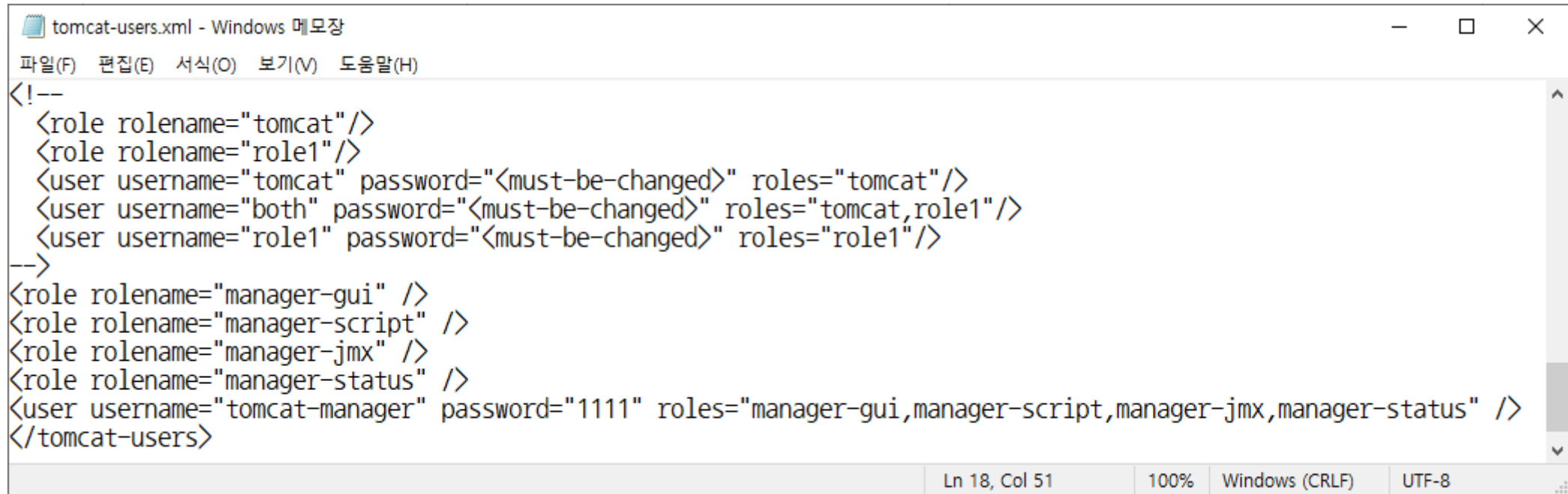
# 4. Tomcat

## 5) 톰캣 관리자 등록

C:\Tomcat\conf\tomcat-users.xml

관리자명 : tomcat-manager

비밀번호 : 1111



```
tomcat-users.xml - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!--
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="<must-be-changed>" roles="tomcat"/>
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
<role rolename="manager-gui" />
<role rolename="manager-script" />
<role rolename="manager-jmx" />
<role rolename="manager-status" />
<user username="tomcat-manager" password="1111" roles="manager-gui,manager-script,manager-jmx,manager-status" />
</tomcat-users>
```



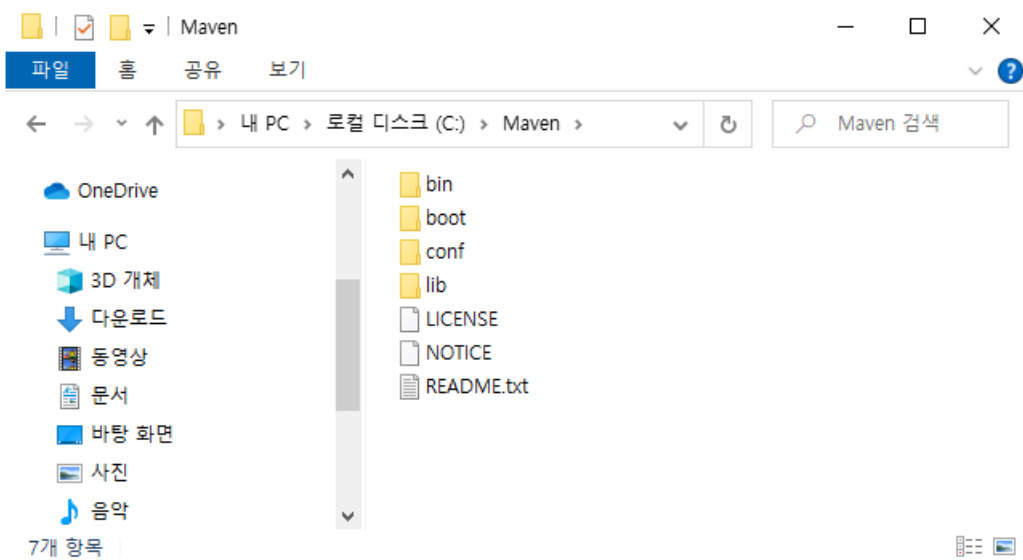
# 5. Maven

## 1) 메이븐 다운로드 (버전 3.8.1)

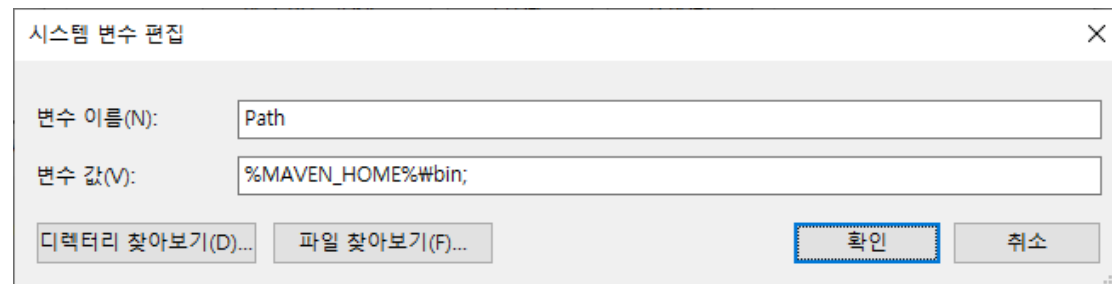
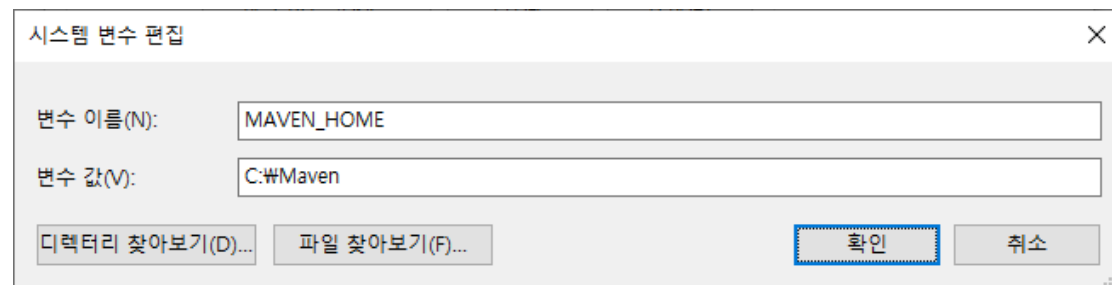
<https://maven.apache.org/>

\* mvnrepository.com에 있는 버전을 사용해야 함

## 2) 메이븐 설치 C:\Maven



## 3) 메이븐 환경변수 제어판 - 시스템



# 6. Application

## pom.xml

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

```
<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

```
<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>
```

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

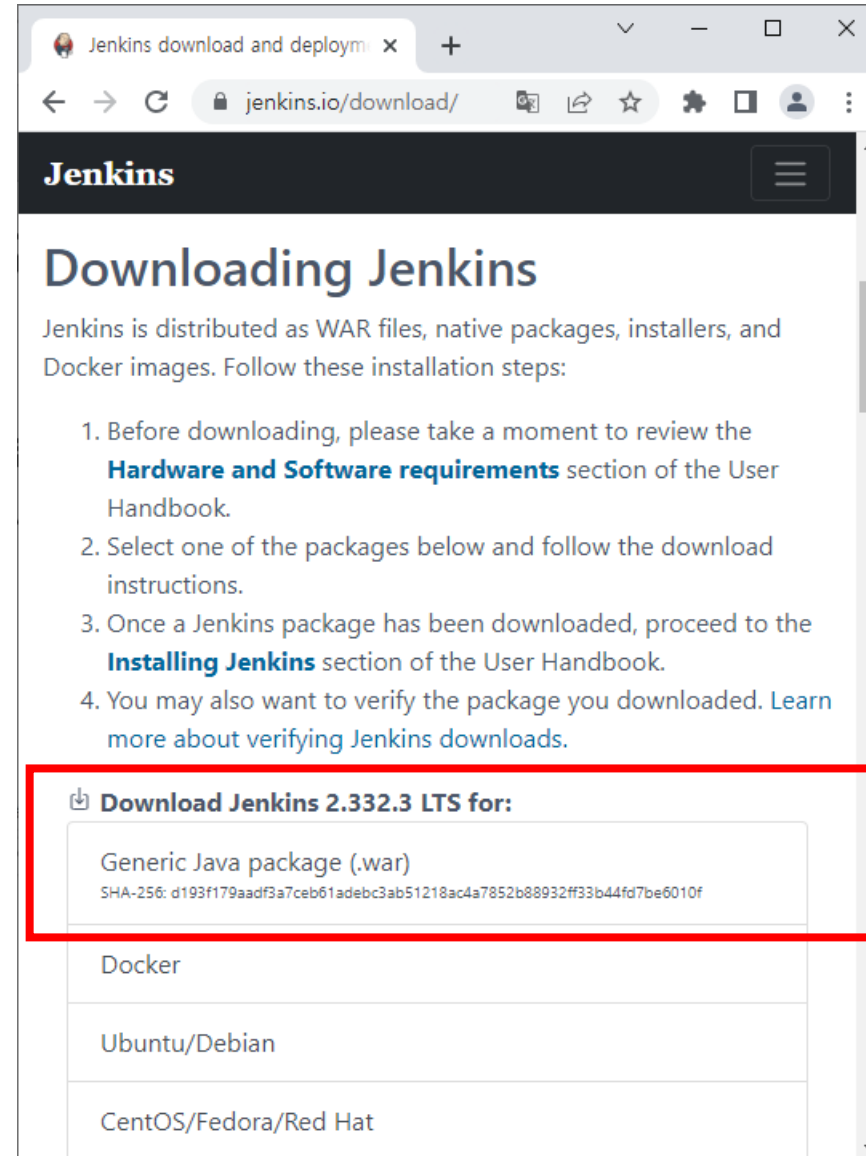
# 7. Jenkins Start

## 1) 젠킨스 다운로드

<https://www.jenkins.io/>

LTS 버전

Generic Java Package(.war)

A screenshot of a web browser showing the Jenkins download page. The browser's address bar shows 'jenkins.io/download/'. The page has a dark header with the 'Jenkins' logo. The main heading is 'Downloading Jenkins'. Below it, a paragraph states: 'Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:'. A numbered list of four steps follows. At the bottom, a section titled 'Download Jenkins 2.332.3 LTS for:' contains a table of download options. The first option, 'Generic Java package (.war)', is highlighted with a red rectangular box. Below it are 'Docker', 'Ubuntu/Debian', and 'CentOS/Fedora/Red Hat'.

Jenkins download and deployment x +

jenkins.io/download/

## Jenkins

### Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the **Hardware and Software requirements** section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the **Installing Jenkins** section of the User Handbook.
4. You may also want to verify the package you downloaded. [Learn more about verifying Jenkins downloads.](#)

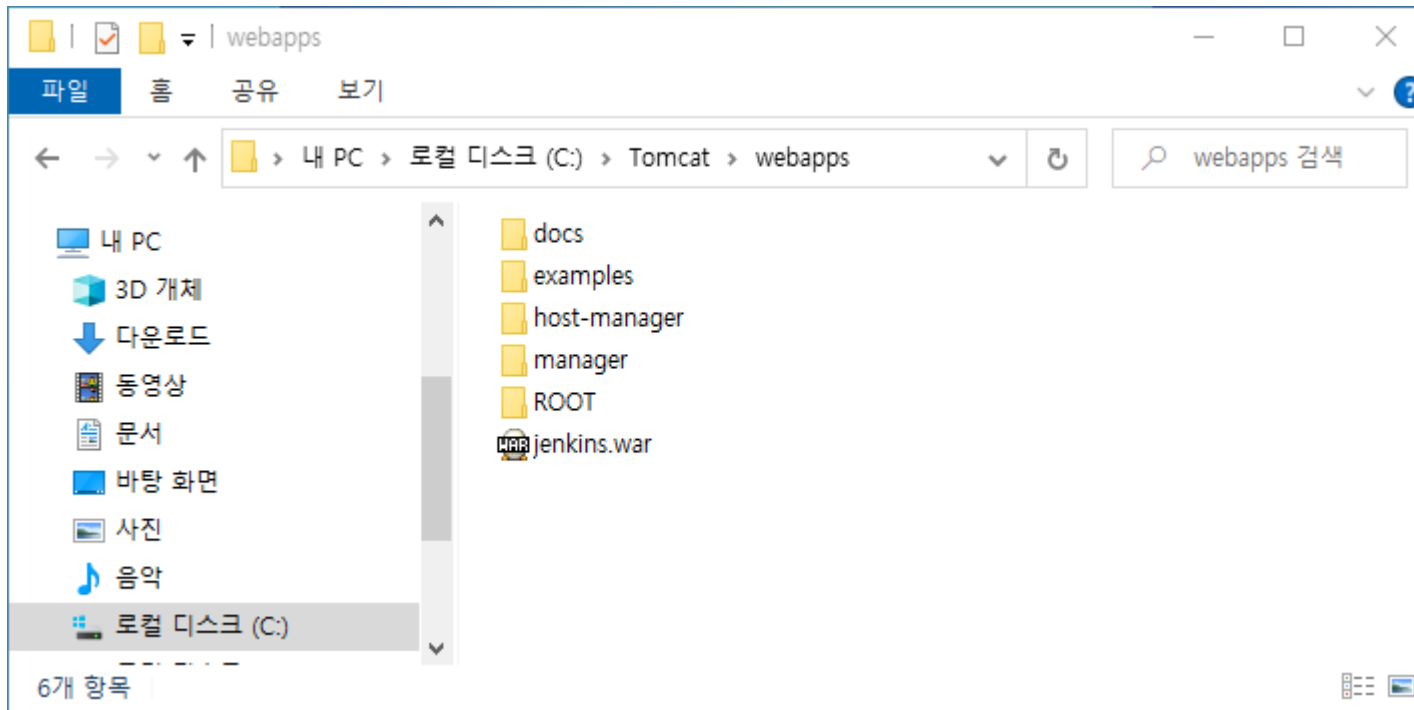
Download Jenkins 2.332.3 LTS for:

Generic Java package (.war) SHA-256: d193f179aadf3a7ceb61adebc3ab51218ac4a7852b88932ff33b44fd7be6010f
Docker
Ubuntu/Debian
CentOS/Fedora/Red Hat

# 7. Jenkins Start

## 2) 젠킨스 배포

C:\Tomcat\webapps 디렉터리에  
다운로드 받은 jenkins.war 파일을 복사해 둬

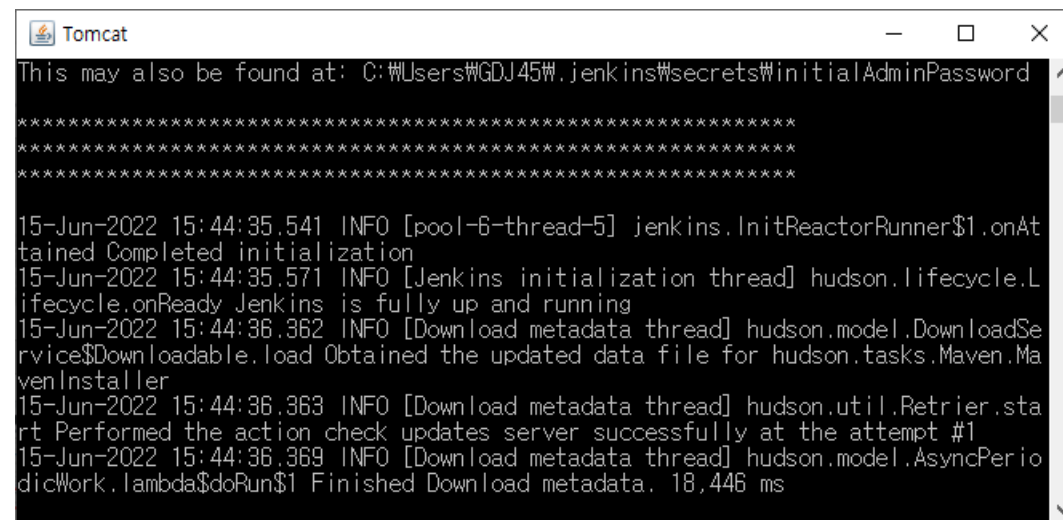
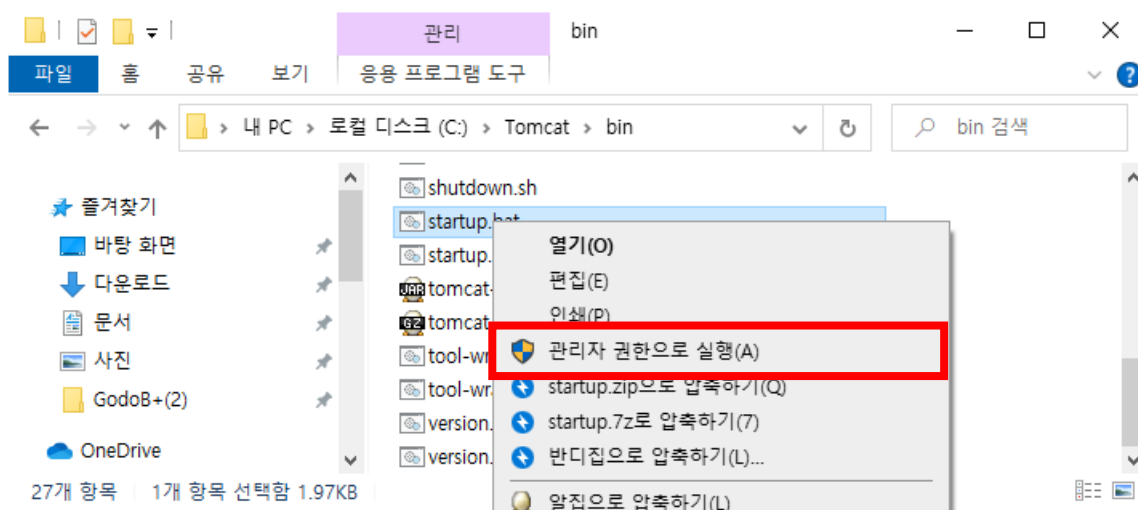


# 7. Jenkins Start

## 3) 톰캣 실행

C:\WTomcat\bin\startup.bat 파일을  
[관리자 권한]으로 실행

\* 리눅스에서는 startup.sh 파일 실행



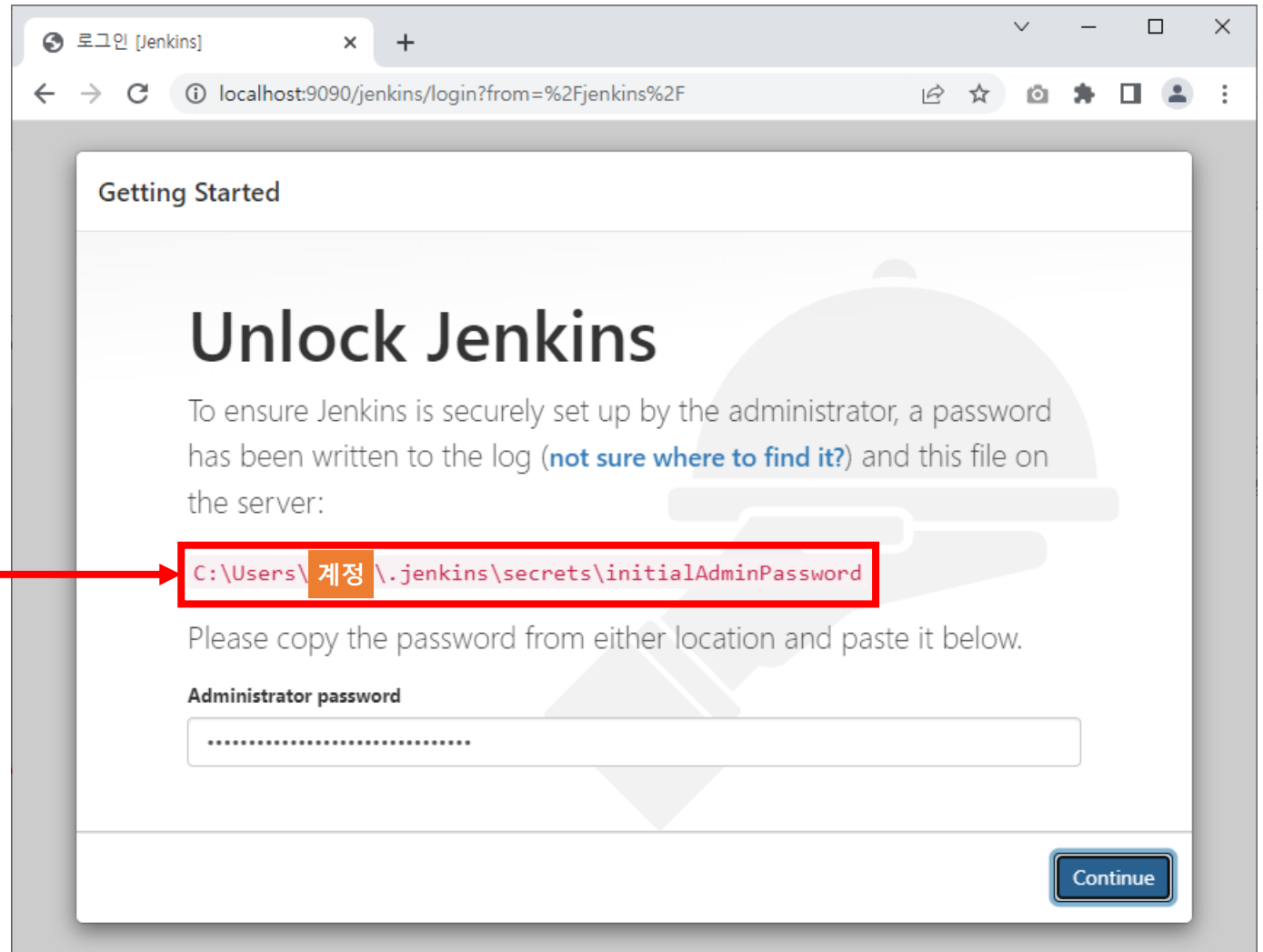
끄지 말 것!

# 7. Jenkins Start

## 4) 젠킨스 실행

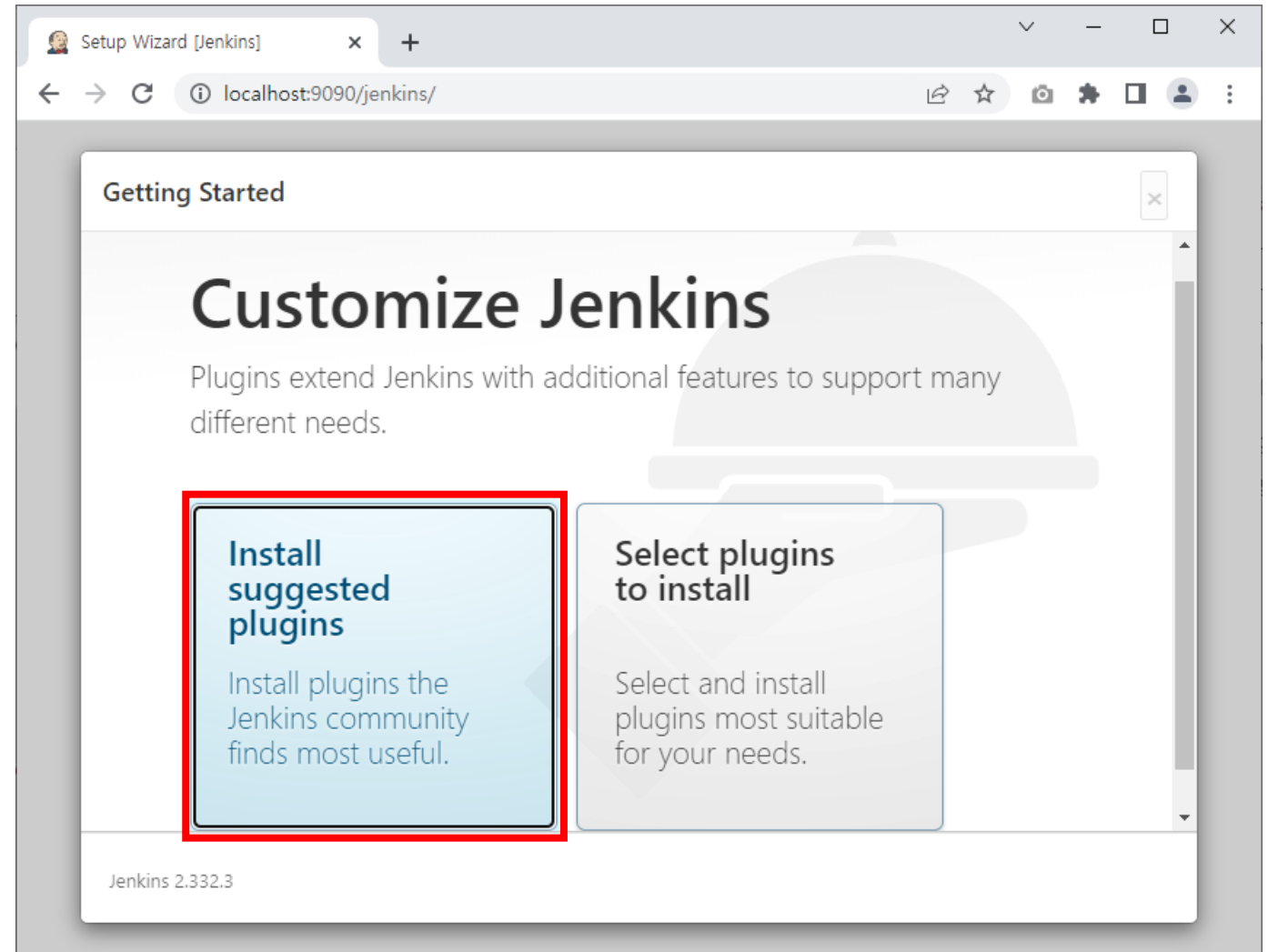
http://localhost:9090/jenkins 접속

초기비밀번호 입력  
(초기비밀번호 위치)



# 7. Jenkins Start

## 5) 젠킨스 플러그인 설치 Install suggested plugins



# 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

Setup Wizard [Jenkins]

localhost:9090/jenkins/

### Getting Started

## Create First Admin User

계정명:

암호:

암호 확인:

이름:

이메일 주소:

Jenkins 2.332.3

[Skip and continue as admin](#)

[Save and Continue](#)

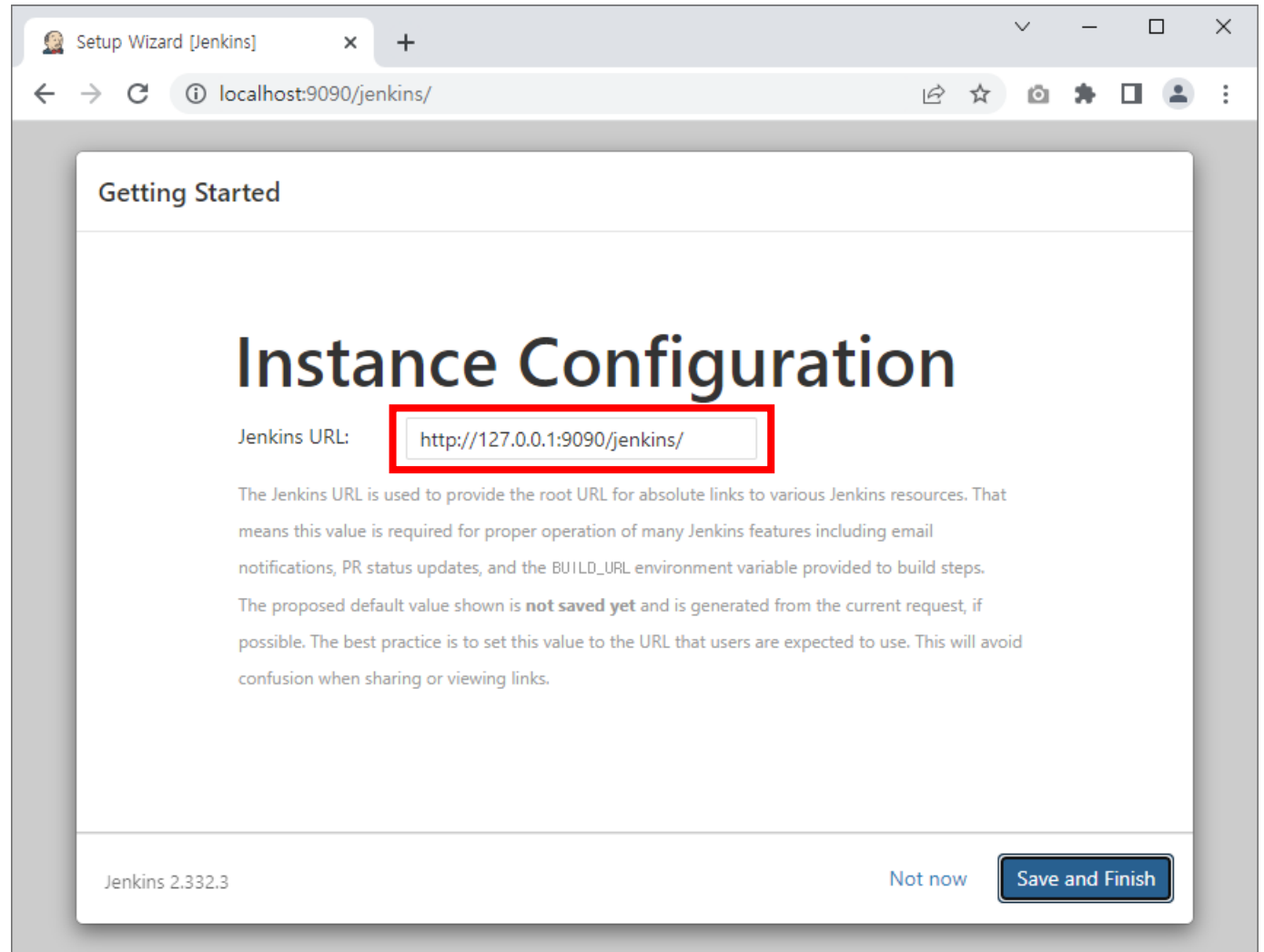


# 7. Jenkins Start

## 7) 젠킨스 주소 등록

http://127.0.0.1:9090/jenkins/

\* 젠킨스 주소는 localhost 사용 불가



The screenshot shows a web browser window titled "Setup Wizard [Jenkins]" with the address bar displaying "localhost:9090/jenkins/". The main content area is titled "Getting Started" and "Instance Configuration". It features a text input field for "Jenkins URL:" containing the value "http://127.0.0.1:9090/jenkins/", which is highlighted with a red rectangular box. Below the input field, there is explanatory text about the Jenkins URL and its importance for various features. At the bottom of the configuration page, there are two buttons: "Not now" and "Save and Finish". The version "Jenkins 2.332.3" is displayed in the bottom left corner of the page.

Setup Wizard [Jenkins]

localhost:9090/jenkins/

Getting Started

## Instance Configuration

Jenkins URL:

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.

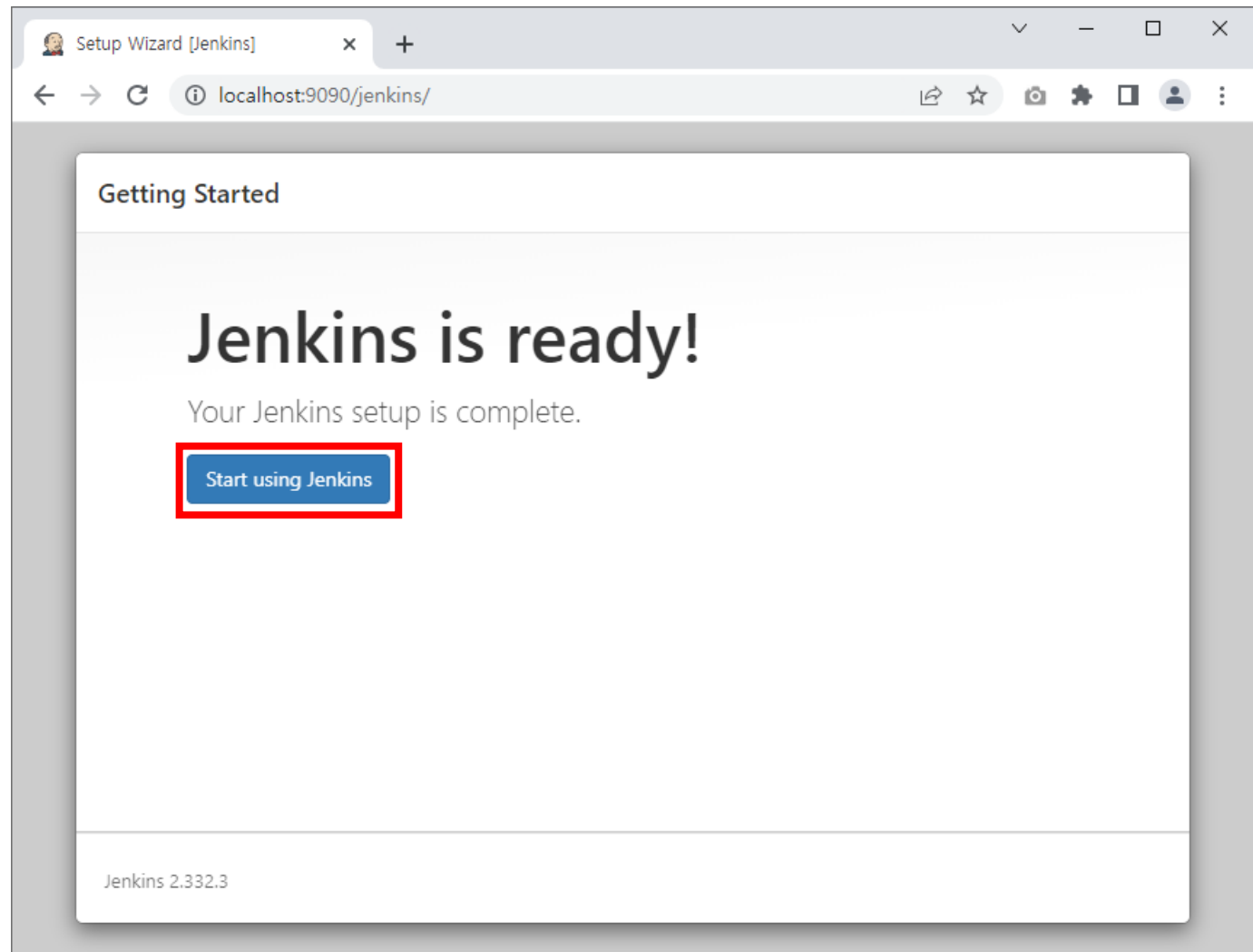
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.332.3

Not now **Save and Finish**

# 7. Jenkins Start

## 8) 젠킨스 준비 완료



# 8. Jenkins Setting

## 1) [Jenkins 관리]

### └ [Global Tool Configuration]

The screenshot shows the Jenkins web interface in a browser window. The address bar indicates the URL is `localhost:9090/jenkins/manage`. The top navigation bar includes the Jenkins logo, a search bar, and links for '관리자' (Admin) and '로그아웃' (Logout). The left sidebar contains a 'Dashboard' menu with several options: '새로운 Item' (New Item), '사람' (People), '빌드 기록' (Build History), 'Jenkins 관리' (Jenkins Management), 'My Views', and 'New View'. The 'Jenkins 관리' option is highlighted with a red box. The main content area is titled 'Jenkins 관리' and contains several informational messages and configuration options. A red box highlights the 'Global Tool Configuration' option, which includes a wrench icon and the text 'Configure tools, their locations and automatic installers.' Other options visible include '시스템 설정' (System Configuration), '플러그인 관리' (Plugin Management), and '노드 관리' (Node Management).

# 8. Jenkins Setting

## 1) [Jenkins 관리]

### └ [Global Tool Configuration]

### └ [Maven Configuration]

- Maven 환경설정파일을 등록하는 작업임



## Global Tool Configuration

### Maven Configuration

Default settings provider

Settings file in filesystem

File path ?

C:\Maven\conf\settings.xml

Default global settings provider

Global settings file on filesystem

File path ?

C:\Maven\conf\settings.xml

# 8. Jenkins Setting

## 1) [Jenkins 관리]

### └ [Global Tool Configuration]

### └ [JDK]

- Java가 설치된 경로를 등록하는 작업임

#### JDK

##### JDK installations

Add JDK



JDK  
Name

MyJDK

##### JAVA\_HOME

C:\Program Files\Java\jdk-11.0.13

☐ Install automatically ?

해제함

Delete JDK

# 8. Jenkins Setting

## 1) [Jenkins 관리]

### └ [Global Tool Configuration]

### └ [Git]

- Git이 설치된 경로를 등록하는 작업임

## Git

### Git installations



Git  
Name

MyGit

Path to Git executable ?

C:\Program Files\Git\bin\git.exe

☐ Install automatically ?

해제함

Delete Git

# 8. Jenkins Setting

## 1) [Jenkins 관리]

### └ [Global Tool Configuration]

### └ [Maven]

- Maven이 설치된 경로를 등록하는 작업임

## Maven

### Maven installations

Add Maven

Maven  
Name

MyMaven

MAVEN\_HOME

C:\Maven

☐ Install automatically ?

해제함

Delete Maven

# 8. Jenkins Setting

## 1) [Jenkins 관리]

### ↳ [플러그인 관리]

The screenshot shows the Jenkins management interface in Korean. The browser address bar indicates the URL is `localhost:9090/jenkins/manage`. The left sidebar contains a list of navigation items: '새로운 Item' (New Item), '사람' (People), '빌드 기록' (Build History), 'Jenkins 관리' (Jenkins Management), 'My Views', and 'New View'. The 'Jenkins 관리' item is highlighted with a red box. The main content area is titled 'Jenkins 관리' and contains several sections:

- A warning message: '역방향 프록시 설정이 잘못된 것으로 파악되었습니다.' (Reverse proxy settings are suspected to be incorrect.) with buttons for '추가 정보' (More Info) and '해제' (Dismiss).
- A security notice: 'Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).' with buttons for 'Set up agent', 'Set up cloud', and 'Dismiss'.
- A 'System Configuration' section with a gear icon and the text '시스템 설정' (System Settings) and '환경변수 및 경로 정보등을 설정합니다.' (Set environment variables and path information, etc.).
- A 'Global Tool Configuration' section with a wrench icon and the text 'Global Tool Configuration' and 'Configure tools, their locations and automatic installers.'
- A '노드 관리' (Node Management) section with a computer icon and the text '노드 관리' and 'Add, remove, control and monitor the various nodes that Jenkins runs jobs on.'
- A '플러그인 관리' (Plugin Management) section with a puzzle piece icon and the text '플러그인 관리' and 'Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.' (You can add, remove, use, or disable plugins to extend Jenkins's functionality.). This section is highlighted with a red box.



# 8. Jenkins Setting

## 1) [Jenkins 관리]

└ [플러그인 관리]

└ 설치 가능

└ deploy 검색

└ Deploy to container 선택

└ Install without restart

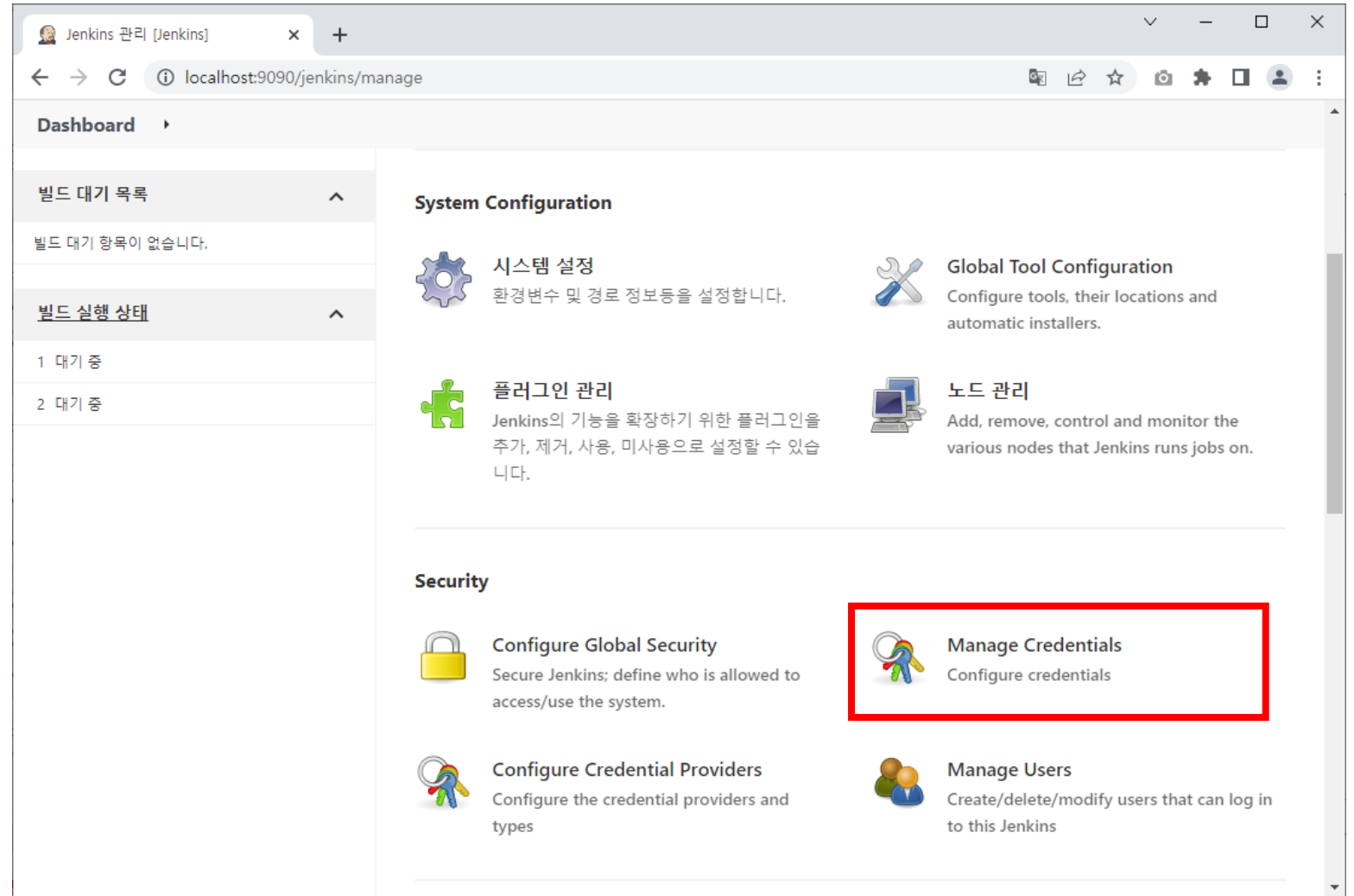
The screenshot shows the Jenkins Plugin Manager interface. The browser address bar indicates the URL is `localhost:9090/jenkins/pluginManager/available`. The Jenkins logo and navigation links are visible at the top. The left sidebar contains links for '대시보드로 돌아가기', 'Jenkins 관리', and 'Update Center'. The main content area is titled 'Plugin Manager' and features a search bar with the text 'deploy'. Below the search bar, there are tabs for '업데이트된 플러그인 목록', '설치 가능' (highlighted with a red box), '설치된 플러그인 목록', and '고급'. A table lists available plugins. The 'Deploy to container' plugin (version 1.16) is highlighted with a red box. It has a checked checkbox in the 'Install' column and a description: 'This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment'. Below the table, there are two buttons: 'Install without restart' (highlighted with a red box) and 'Download now and install after restart'. The bottom right corner shows the update information: 'Update information obtained: 3 hr 5 min ago'.

Install	Name ↓	Released
<input checked="" type="checkbox"/>	<b>Deploy to container</b> 1.16 Artifact Uploaders This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment	1 yr 7 mo ago
<input type="checkbox"/>	<b>Docker Pipeline</b> 1.28 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	4 mo 17 days ago
	<b>Artifactory</b> 3.16.2	

# 8. Jenkins Setting

## 1) [Jenkins 관리]

### └ [Manage Credentials]



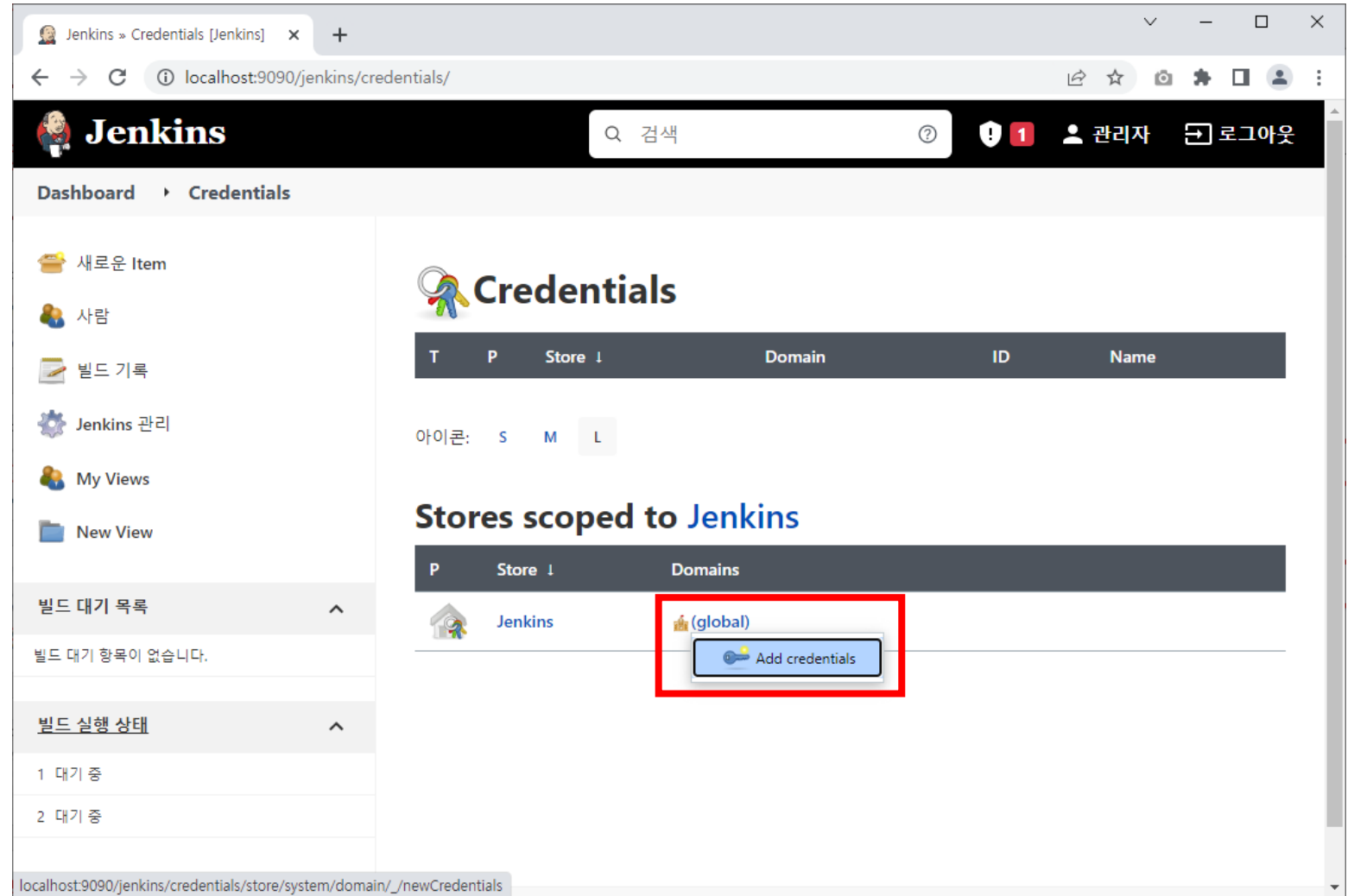
# 8. Jenkins Setting

## 1) [Jenkins 관리]

└ [Manage Credentials]

└ [Add credentials]

총 2번 수행해야 함

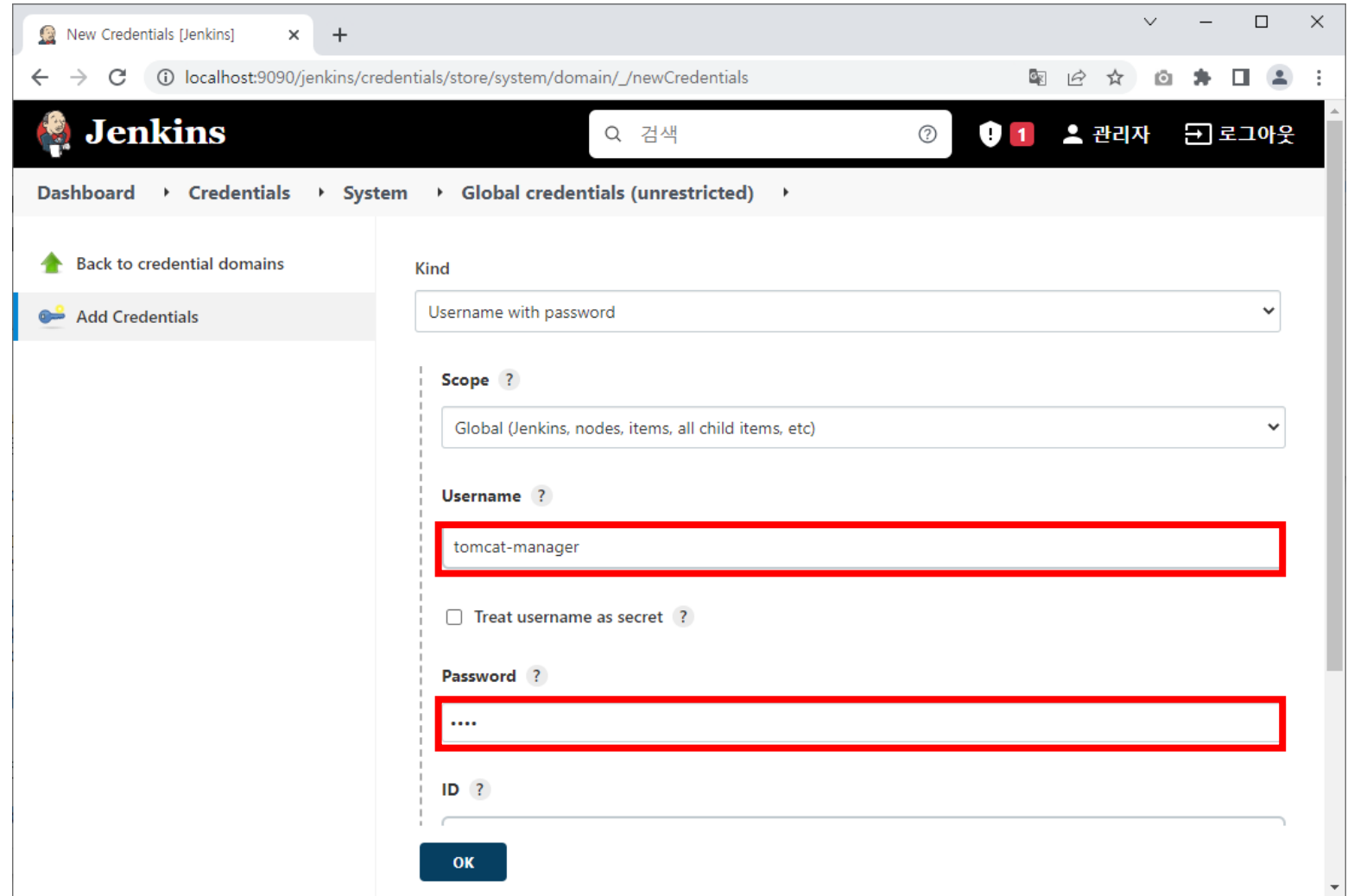


# 8. Jenkins Setting

(1) 톰캣 관리자의  
Username과 Password 등록

Username : tomcat-manager  
Password : 1111

\* tomcat-users.xml 에 등록된 정보임



New Credentials [Jenkins]

localhost:9090/jenkins/credentials/store/system/domain/\_/newCredentials

Jenkins

Dashboard > Credentials > System > Global credentials (unrestricted)

Back to credential domains

Add Credentials

Kind

Username with password

Scope ?

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

Username ?

tomcat-manager

☐ Treat username as secret ?

Password ?

....

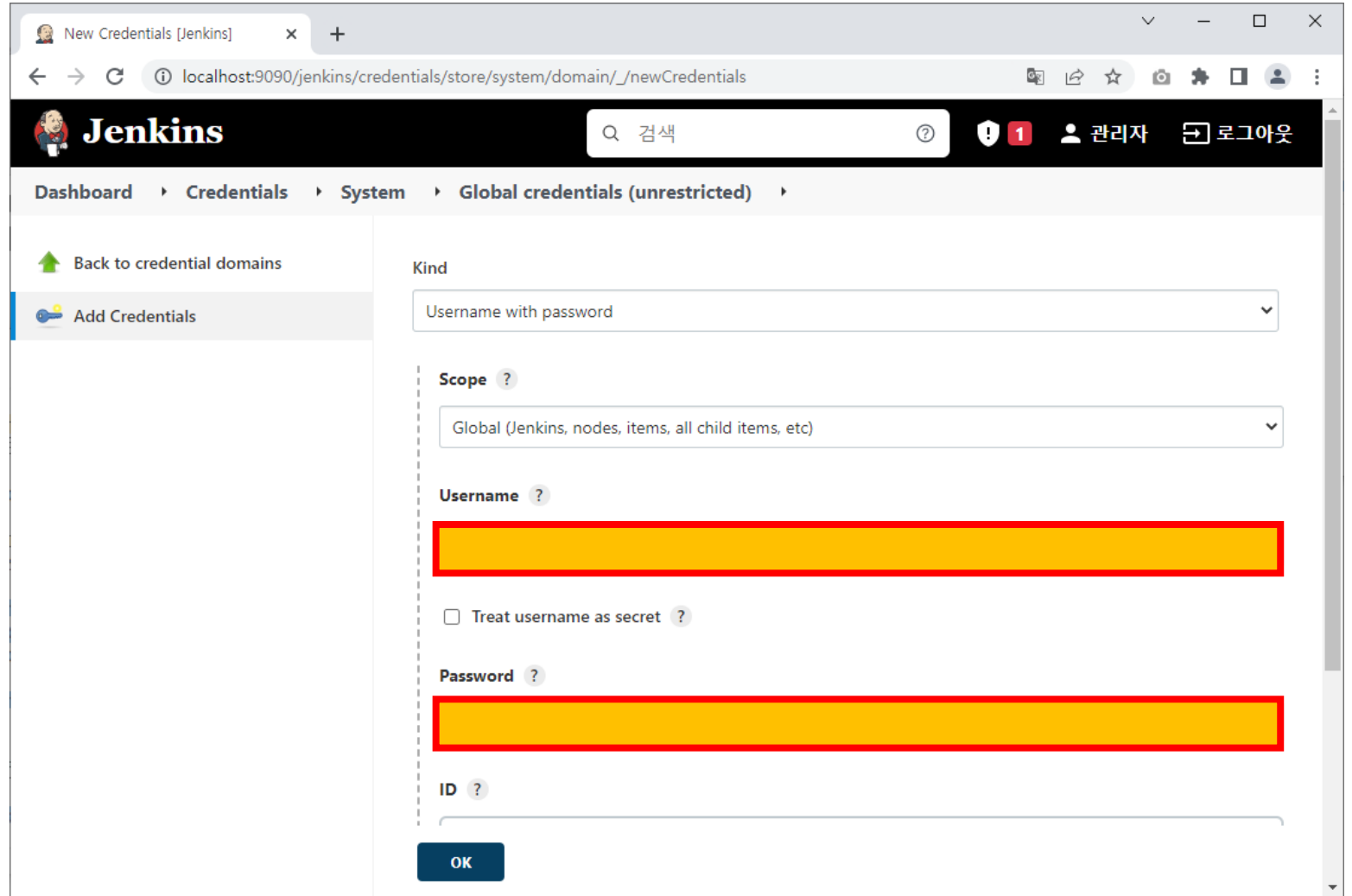
ID ?

OK

# 8. Jenkins Setting

(2) github 관리자의  
Username과 Password 등록

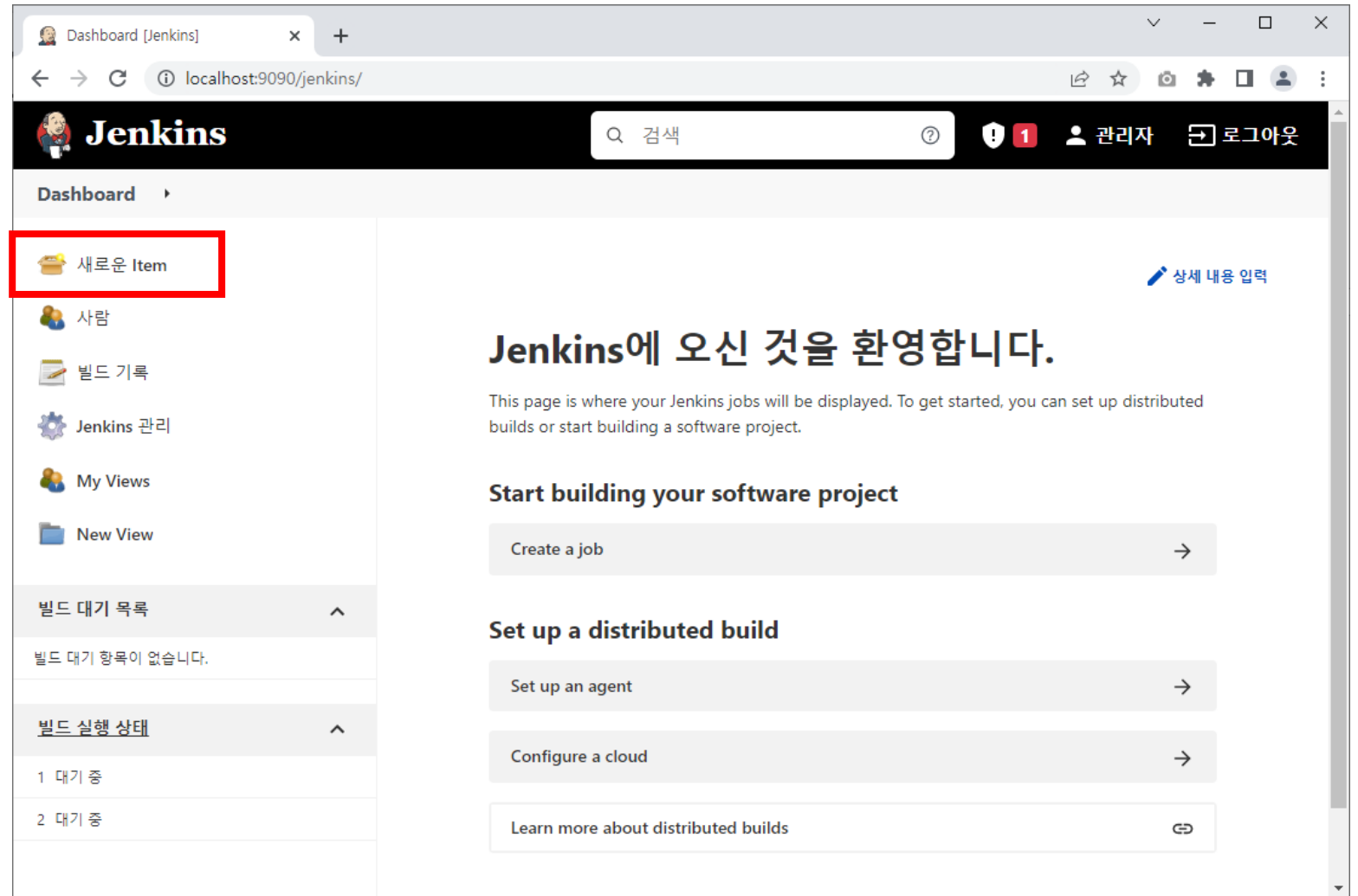
Username : 본인 github 아이디  
Password : Personal access token 값



The screenshot shows the Jenkins web interface for creating a new credential. The browser address bar indicates the URL is `localhost:9090/jenkins/credentials/store/system/domain/_/newCredentials`. The Jenkins logo and navigation menu are visible at the top. The breadcrumb trail is `Dashboard > Credentials > System > Global credentials (unrestricted)`. On the left sidebar, there are links for `Back to credential domains` and `Add Credentials`. The main form is titled `Kind` and has a dropdown menu set to `Username with password`. Below this, the `Scope` dropdown is set to `Global (Jenkins, nodes, items, all child items, etc)`. The `Username` field is a text input, and the `Password` field is a password input, both highlighted with a red border. There is a checkbox labeled `Treat username as secret` which is currently unchecked. The `ID` field is a text input. At the bottom of the form is an `OK` button.

# 9. Project Setting

[새로운 Item]





# 9. Project Setting


[프로젝트명] 입력


Freestyle project 선택


**Enter an item name**  
  
» Required field


**Freestyle project**  
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (former

**Multi-configuration project**  
다양한 환경에서의 테스트, 플랫폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is jus  
as they are in different folders.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**  
Creates a set of multibranch project subfolders by scanning for repositories.

# 9. Project Setting

## [소스 코드 관리]

### └ [Git]

#### └ [Repository URL]

애플리케이션github주소

#### └ [Credentials]

github관리자계정

#### └ [Branch Specifier]

.main

SongProject Config [Jenkins] x +

localhost:9090/jenkins/job/SongProject/configure

Dashboard ▸ SongProject ▸

General 소스 코드 관리 빌드 유발 빌드 환경 Build 빌드 후 조치

소스 코드 관리

☐ None  
☒ Git ?

Repositories ?

Repository URL ?  
https://github.com/goodeeit/SongProject.git

Credentials ?  
goodeeit/\*\*\*\*\* Add

고급...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? X  
\*/main

Add Branch

저장 Apply



# 9. Project Setting

## [Build]

### └ [Maven Version]

MyMaven

### └ [Goals]

clean package

The screenshot shows the Jenkins configuration page for 'SongProject'. The browser address bar indicates the URL is 'localhost:9090/jenkins/job/SongProject/configure'. The page has a breadcrumb trail: 'Dashboard > SongProject >'. Below this, there are tabs for 'General', '소스 코드 관리', '빌드 유발', '빌드 환경', 'Build', and '빌드 후 조치'. The 'Build' tab is currently selected. Inside the 'Build' tab, there is a section titled 'Build' with a red border. This section contains the following elements: a checkbox labeled 'Invoke top-level Maven targets' (which is checked), a dropdown menu for 'Maven Version' set to 'MyMaven', and a text input field for 'Goals' containing 'clean package'. Below the 'Goals' field is a '고급...' (Advanced...) button. Below the 'Build' section is an 'Add build step' button. At the bottom of the page, there are two buttons: '저장' (Save) and 'Apply'. The footer of the page shows 'REST API' and 'Jenkins 2.332.3'.

# 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**

SongProject Config [Jenkins] x +

localhost:9090/jenkins/job/SongProject/configure

Dashboard ▸ SongProject ▸

General 소스 코드 관리 빌드 유발 빌드 환경 Build **빌드 후 조치**

### 빌드 후 조치

Deploy war/ear to a container X

WAR/EAR files ?

\*\*/\*.war

Context path ?

Containers X

Tomcat 9.x Remote

Credentials

tomcat-manager/\*\*\*\*\* Add

Tomcat URL ?

http://localhost:9090

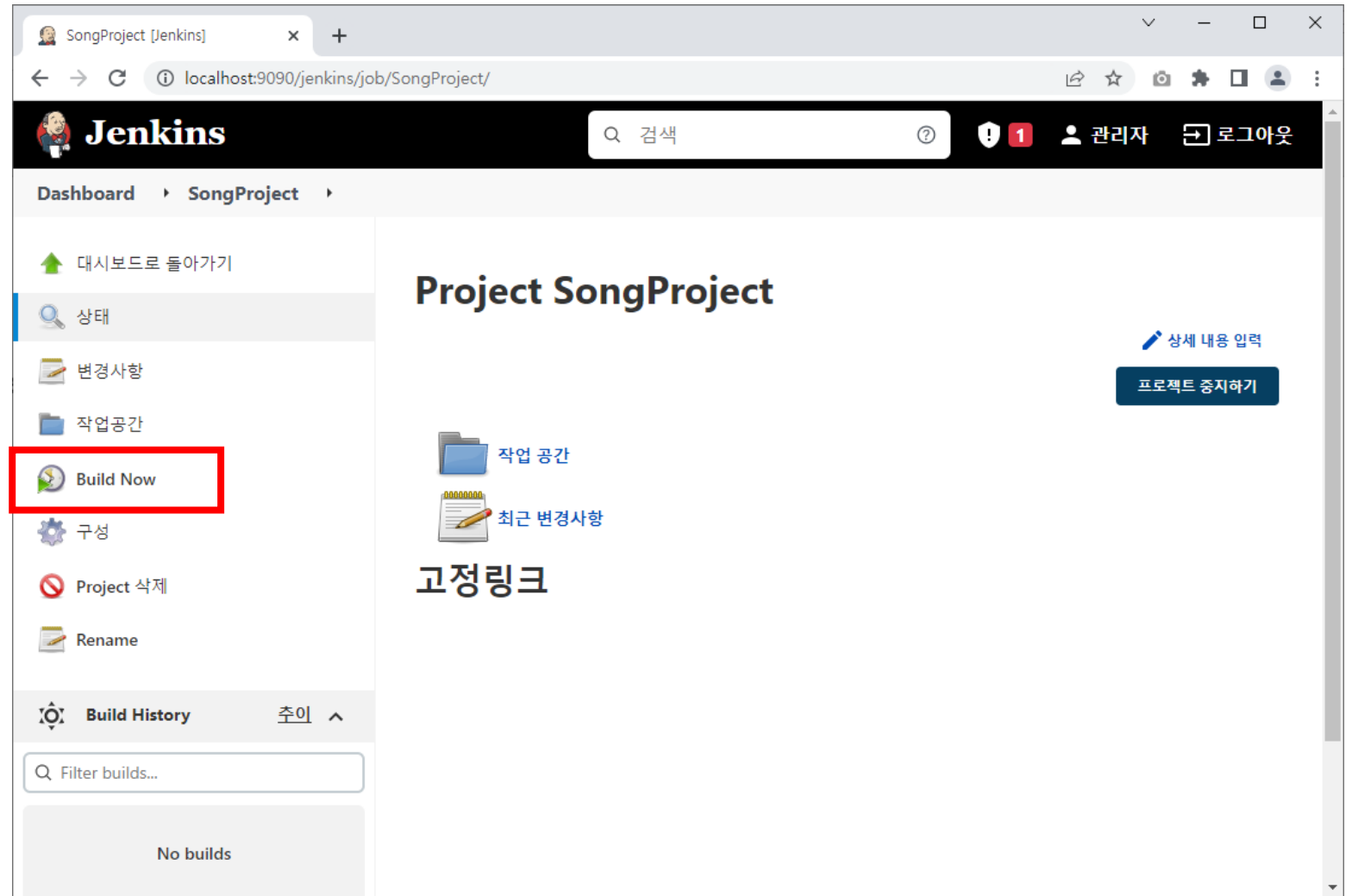
고급...

Add Container ▾

저장 Apply

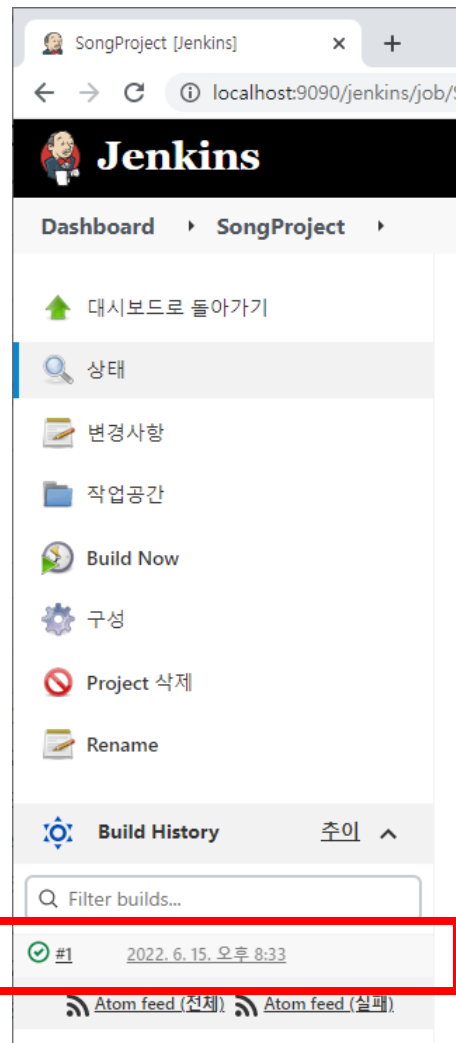
# 10. Project Build & Deploy

[Build Now]



# 10. Project Build & Deploy

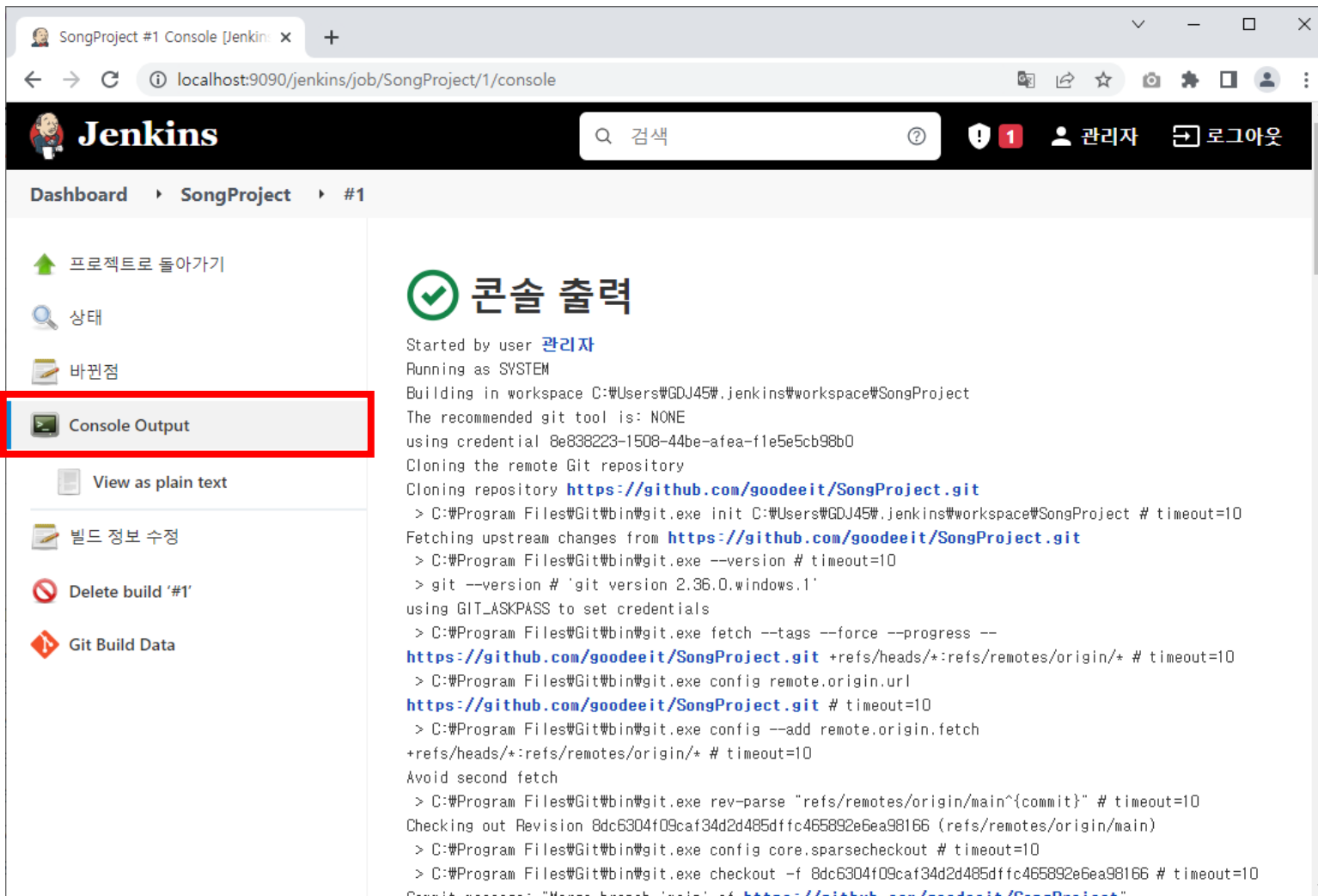
## [Build History]



The screenshot shows the Jenkins 'Build History' page for the 'SongProject' job. The left sidebar contains navigation links: '대시보드로 돌아가기', '상태', '변경사항', '작업공간', 'Build Now', '구성', 'Project 삭제', and 'Rename'. The main area shows a table of builds. The first build, labeled '#1', is highlighted with a red box and shows a green status icon and the timestamp '2022. 6. 15. 오후 8:33'. Below the table are links for 'Atom feed (전체)' and 'Atom feed (실패)'.

Build	Status	Timestamp
#1	Success	2022. 6. 15. 오후 8:33

## [Console Output]

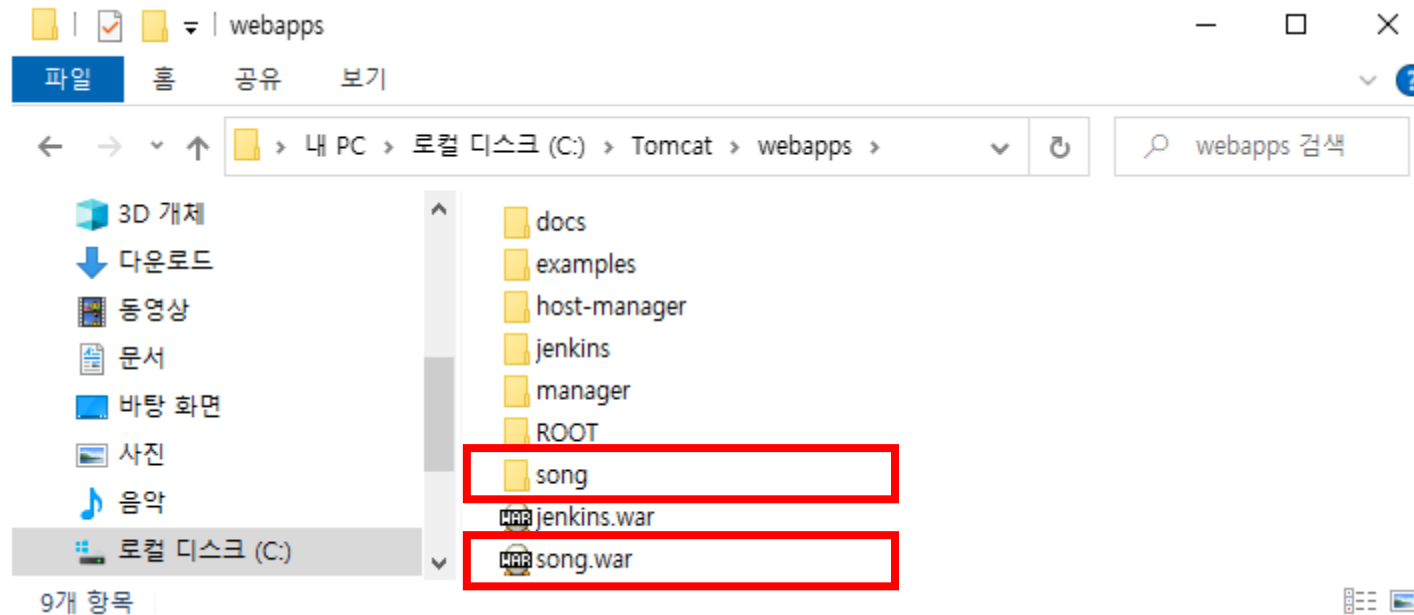


The screenshot shows the Jenkins 'Console Output' page for the 'SongProject' job, build #1. The left sidebar contains navigation links: '프로젝트로 돌아가기', '상태', '바뀐점', 'Console Output' (highlighted with a red box), 'View as plain text', '빌드 정보 수정', 'Delete build '#1'', and 'Git Build Data'. The main area displays the console output, which includes the following text:

```
Started by user 관리자
Running as SYSTEM
Building in workspace C:\Users\GDJ45\jenkins\workspace\SongProject
The recommended git tool is: NONE
using credential 8e838223-1508-44be-afea-f1e5e5cb98b0
Cloning the remote Git repository
Cloning repository https://github.com/goodeeit/SongProject.git
> C:\Program Files\Git\bin\git.exe init C:\Users\GDJ45\jenkins\workspace\SongProject # timeout=10
Fetching upstream changes from https://github.com/goodeeit/SongProject.git
> C:\Program Files\Git\bin\git.exe --version --version # timeout=10
> git --version # 'git version 2.36.0.windows.1'
using GIT_ASKPASS to set credentials
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress --
https://github.com/goodeeit/SongProject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe config remote.origin.url
https://github.com/goodeeit/SongProject.git # timeout=10
> C:\Program Files\Git\bin\git.exe config --add remote.origin.fetch
+refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 8dc6304f09caf34d2d485dfc465892e6ea98166 (refs/remotes/origin/main)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f 8dc6304f09caf34d2d485dfc465892e6ea98166 # timeout=10
Commit message: "Merge branch 'main' of https://github.com/goodeeit/SongProject"
```

# 10. Project Build & Deploy

## 톰캣에 배포되었는지 확인



# 10. Project Build & Deploy

