

표준프레임워크
CI 및 Nexus 환경 구축 가이드
Version 4.3

Index

1. 표준프레임워크 EGOVCI 패키지 설치	3
1.1 개요.....	3
1.2 EGOVCI 압축풀기.....	3
1.3 EGOVCI시스템 구성.....	3
1.4 CI 시스템 구동 (START/STOP)	4
2. NEXUS 설정 정보.....	7
2.1 NEXUS 서버구동(START/STOP)	7
2.2 NEXUS 저장소 설정.....	7
2.3 NEXUS 저장소에 라이브러리 추가.....	8
2.4 NEXUS 실행 및 추가된 라이브러리 확인.	8
2.5 NEXUS저장소를 프로젝트에 반영.....	9
2.6 NEXUS 저장소에 3 RD PARTY 라이브러리 추가.....	10
3. CI (SVN & JENKINS) 설정.....	12
3.1 SUBVERSION 설치 (선택사항)	12
3.2 SVN REPOSITORIES 설정 및 사용자 등록.	12
3.2.1 저장소 설정 및 등록	13
3.2.2 사용자 등록	14
3.2.3 SVN 서버 시작/종료.....	15
3.2.4 Eclipse 프로젝트를 SVN서버에 등록.	15
3.3 JENKINS 설정.....	17
3.3.1 Jenkins 기본 시스템 설정	18
3.3.2 새 작업 생성	21
3.3.3 작업 Bulid.....	25

1. 표준프레임워크 eGovCI 패키지 설치

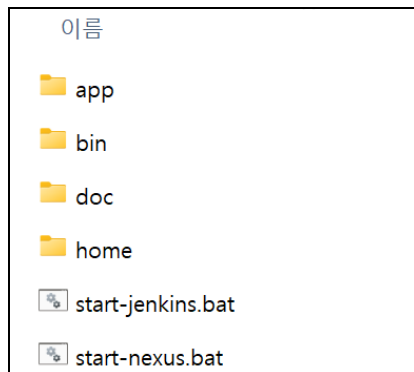
1.1 개요

본 가이드는 전자정부 표준프레임워크 센터에서 제공하는 eGovCI 구성 및 설치 가이드이며, 아래와 같은 구성으로 이루어져 있다.

- eGovCI 패키지 (윈도우용 exe 압축파일 또는 리눅스용 tar.gz 압축파일)
- Nexus repository 구성 정보.
- CI 관련 구성 정보 (SVN 포함, 리눅스용 패키지는 미포함 - 별도 설치필요).
- 기본 접속 설정 URL
- JENKINS : http://<ip_address>:8082
- NEXUS : http://<ip_address>:8081/nexus

1.2 eGovCI 압축풀기

전자정부 표준프레임워크에서 제공하는 eGovCI ("eGovCI-X.X.X_XXbit") 파일을 "C:\\"에 압축을 푼다. 압축을 푼 구조는 아래와 같다. (리눅스의 경우 /home/<사용자> 디렉토리 권장)



1.3 eGovCI 시스템 구성

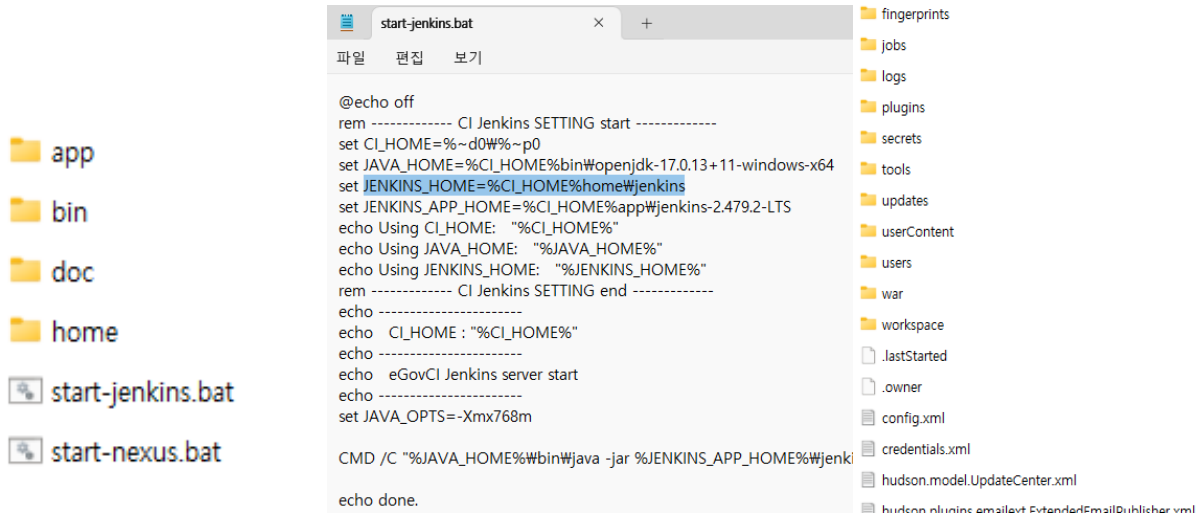
eGovCI 패키지는 별도의 모듈이 필요하지 않도록, 아래와 같은 구성으로 제공하고 있다. 주요 모듈로는 JAVA JDK, Tomcat configurations, Maven 등으로 구성되어 있다.

- Apache Ant 1.10.15 (under /bin dictory)
- Apache Maven 3.9.9 (under /bin dictory)
- OpenJDK.1.8 (under /bin dictory) - 리눅스 제외
- OpenJDK.17 (under /bin dictory) - 리눅스 제외

- Nexus Repository 2.15.2 (under /app dictory)
- Jenkins 2.479.2 LTS (under /app dictory)
- VisualSVN-Server-4.1.3.msi (under /app dictory) - 별도 설치형 파일 제공 (리눅스 제외)

1.4 CI 시스템 구동

CI 시스템 구성 이후, “start-jenkins.bat” 파일로 Jenkins 서버를 구동 할 수 있다. (리눅스의 경우 동일한 이름파일의 .sh 파일 제공)



```

start-jenkins.bat
파일 편집 보기
@echo off
rem ----- CI Jenkins SETTING start -----
set CI_HOME=%~d0%~p0
set JAVA_HOME=%CI_HOME%bin\openjdk-17.0.13+11-windows-x64
set JENKINS_HOME=%CI_HOME%home\jenkins
set JENKINS_APP_HOME=%CI_HOME%app\jenkins-2.479.2-LTS
echo Using CI_HOME: "%CI_HOME%"
echo Using JAVA_HOME: "%JAVA_HOME%"
echo Using JENKINS_HOME: "%JENKINS_HOME%"
rem ----- CI Jenkins SETTING end -----
echo -----
echo CI_HOME : "%CI_HOME%"
echo -----
echo eGovCI Jenkins server start
echo -----
set JAVA_OPTS=-Xmx768m

CMD /C "%JAVA_HOME%\bin\java -jar %JENKINS_APP_HOME%\jenkins.war %JAVA_OPTS%"
echo done.
  
```

Linux 의 경우 다음 내용을 추가로 작성한다.

- setEnvJDK8.sh 파일과 setEnvJDK17.sh 파일의 “export JAVA_HOME=” 뒤에 각 파일별로 알맞게 JDK 8, 17 경로를 입력한다.
- ex) export JAVA_HOME= /usr/lib/jvm/java-17-openjdk-amd64

구동 이후, 콘솔 창에서 아래와 같은 내용을 참조할 수 있으며, 해당 포트 번호를 확인 할 수 있다.

```

Tomcat
21-Jan-2019 13:34:38.777 INFO org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
21-Jan-2019 13:34:38.788 INFO org.apache.catalina.startup.Catalina.start Server startup in 4263 ms
21-Jan-2019 13:34:39.772 INFO [pool-6-thread-2] jenkins.InitReactorRunner$1.onAttained Started initialization
21-Jan-2019 13:34:39.967 INFO [pool-6-thread-1] jenkins.InitReactorRunner$1.onAttained Listed all plugins
21-Jan-2019 13:34:45.273 INFO [pool-6-thread-9] jenkins.InitReactorRunner$1.onAttained Prepared all plugins
21-Jan-2019 13:34:45.285 INFO [pool-6-thread-5] jenkins.InitReactorRunner$1.onAttained Started all plugins
21-Jan-2019 13:34:45.926 INFO [GitSCM.onLoaded] hudson.ExtensionFinder$GuiceFinder$FaultTolerantScope$1.error Failed to instantiate optional component hudson.plugins.build_timeout.operations.AbortAndRestartOperation$DescriptorImpl; skipping
21-Jan-2019 13:34:46.907 INFO [pool-6-thread-12] jenkins.InitReactorRunner$1.onAttained Augmented all extensions
21-Jan-2019 13:34:47.194 INFO [pool-6-thread-2] jenkins.InitReactorRunner$1.onAttained Loaded all jobs
21-Jan-2019 13:34:47.264 INFO [Download metadata thread] hudson.model.AsyncPeriodicWork$1.run Started Download metadata
21-Jan-2019 13:34:47.270 INFO [Download metadata thread] hudson.model.AsyncPeriodicWork$1.run Finished Download metadata. 6 ms
21-Jan-2019 13:34:47.719 INFO [Finalizing set up] org.springframework.context.support.AbstractApplicationContext.prepareRefresh Refreshing org.springframework.web.context.support.StaticWebApplicationContext@243fd89d: display name [Root WebApplicationContext]; startup_date [Mon Jan 21 13:34:47 KST 2019]; root of context hierarchy
21-Jan-2019 13:34:47.719 INFO [Finalizing set up] org.springframework.context.support.AbstractApplicationContext.obtainFreshBeanFactory Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@243fd89d]: org.springframework.beans.factory.support.DefaultListableBeanFactory@21a173ca
21-Jan-2019 13:34:47.737 INFO [Finalizing set up] org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@21a173ca: defining beans [authenticationManager]; root of factory hierarchy
21-Jan-2019 13:34:47.974 INFO [Finalizing set up] org.springframework.context.support.AbstractApplicationContext.prepareRefresh Refreshing org.springframework.web.context.support.StaticWebApplicationContext@2e86f090: display name [Root WebApplicationContext]; startup_date [Mon Jan 21 13:34:47 KST 2019]; root of context hierarchy
21-Jan-2019 13:34:47.975 INFO [Finalizing set up] org.springframework.context.support.AbstractApplicationContext.obtainFreshBeanFactory Bean factory for application context [org.springframework.web.context.support.StaticWebApplicationContext@2e86f090]: org.springframework.beans.factory.support.DefaultListableBeanFactory@14a0401a
21-Jan-2019 13:34:47.977 INFO [Finalizing set up] org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@14a0401a: defining beans [filter, legacy]; root of factory hierarchy
21-Jan-2019 13:34:48.042 INFO [pool-6-thread-9] jenkins.InitReactorRunner$1.onAttained Completed initialization
21-Jan-2019 13:34:48.080 INFO [Jenkins initialization thread] hudson.WebAppMain$3.run Jenkins is fully up and running
  
```

** 서버 성능에 따라 구동 시간이 다소 소요될 수 있다.

구동후, 아래와 같은 주소를 CI 웹 페이지를 확인 할 수 있다.

<http://localhost:8082>

계정

- ID : admin
- Password : test1234

*** 기본 제공되는 비밀번호는 반드시 다른 것으로 변경하여 사용하시기 바랍니다.**



Sign in to Jenkins

사용자 이름

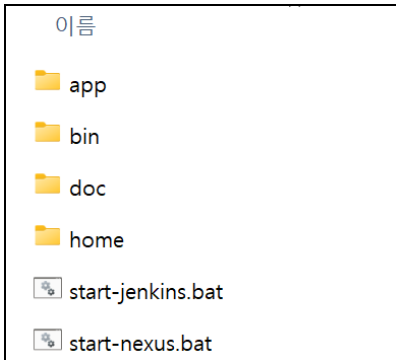
비밀번호

로그인 상태 유지

2. Nexus Repository 설정 정보

2.1 Nexus Repository 서버구동

CI 시스템 구성 이후, “start-nexus.bat”을 사용하여 Nexus 서버 구동 할 수 있으며, 실행된 콘솔창을 닫으면 Nexus Repository 서버를 멈출 수 있다.



구동 이후, 아래와 같은 콘솔 창으로 nexus 을 시작을 확인 할 수 있다.

기본 접속 URL : <http://127.0.0.1:8081/nexus/>

```
Using CI_HOME: "C:\\eGovCI-4.3.0_64bit\"
Using JAVA_HOME: "C:\\eGovCI-4.3.0_64bit\\bin\\openjdk-8u432-b06-windows-x64"
Using PATH: "C:\\eGovCI-4.3.0_64bit\\bin\\openjdk-8u432-b06-windows-x64\\bin;C:\\ProgramData\\anaconda3\\condabin;C:\\Windows"
Using NEXUS_HOME: "C:\\eGovCI-4.3.0_64bit\\app\\nexus-2.15.2-03-bundle\\nexus-2.15.2-03"
-----
CI_HOME : "C:\\eGovCI-4.3.0_64bit\"
-----
eGovCI Nexus server start
-----
```

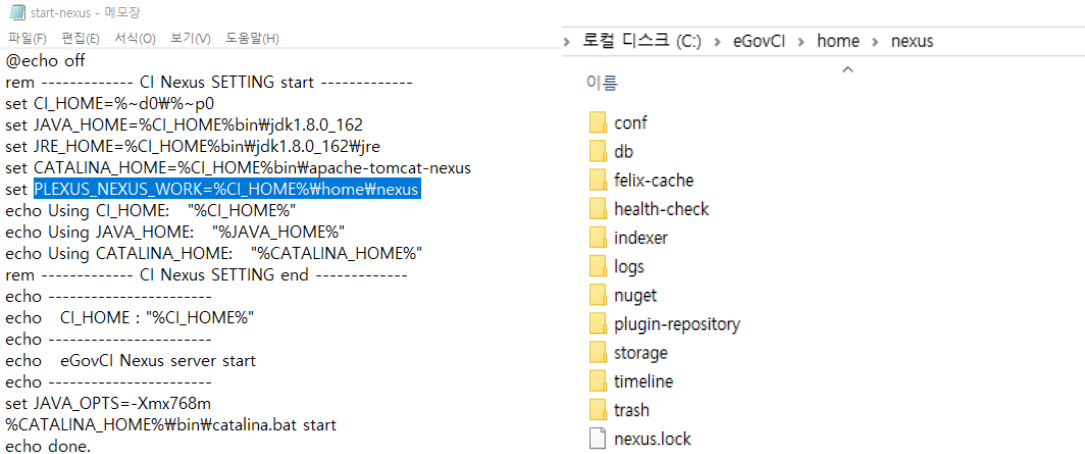
포트(Port)변경을 아래와 같은 경로에서 nexus.properties 파일 내에 application-port=8081 을 변경할 수 있다

파일 경로 : <eGovCI_home>\app\nexus-2.15.2-03-bundle\nexus-2.15.2-03\conf

```
# Jetty section
application-port=8081
application-host=0.0.0.0
nexus-webapp=${bundleBasedir}/nexus
nexus-webapp-context-path=/nexus
```

2.2 Nexus Repository 설정

Nexus 의 저장소는 기본으로 <eGovCI_home>\home\nexus 로 설정되어 있으며, 아래와 같이 별도 설정이 가능하다.



2.3 Nexus Repository 에 라이브러리 일괄 추가.

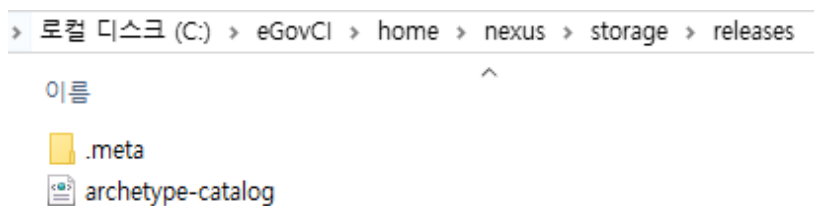
Nexus 저장소에 라이브러리 추가 위치

<eGovCI_home>\home\nexus\storage

- Storage 는 Nexus 에 등록된 central repository 들을 포함하고 있으며, 각 repository 는 등록된 URL 기반으로 웹 상의 해당 repository 에 접속하여 라이브러리를 내려 받아 관리함

또한, Nexus storage 에 필요한 라이브러리를 아래와 같이 한 번에 추가가능하다.

- Target 저장소 : <eGovCI_home>/home/nexus/storage/releases
- 직접 또는 FTP 환경을 설정하여 Target 저장소에 위치 시킨다.

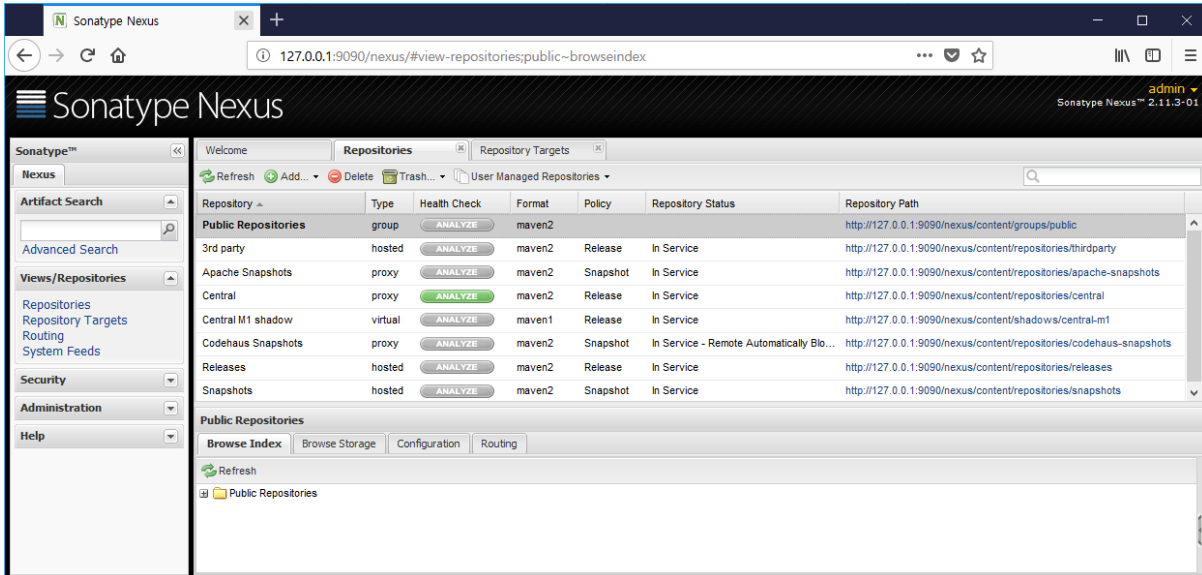


2.4 Nexus Repository 실행 및 추가된 라이브러리 확인.

"start-nexus.bat" 실행하여, Nexus Repository 를 실행가능하며, 웹브라우저에서 아래 주소로 Nexus Repository 확인이 가능하다.

<http://127.0.0.1:8081/nexus>

Nexus 를 접속 후 “repositories” 선택하고 “public repositories” 또는 release 저장소에서 추가한 라이브러리 목록을 확인 할 수 있다.



2.5 Nexus Repository URL 을 프로젝트에 반영.

프로젝트의 pom.xml file 을 열고 아래와 같이 <repositories> 변경하여 반영할 수 있다.

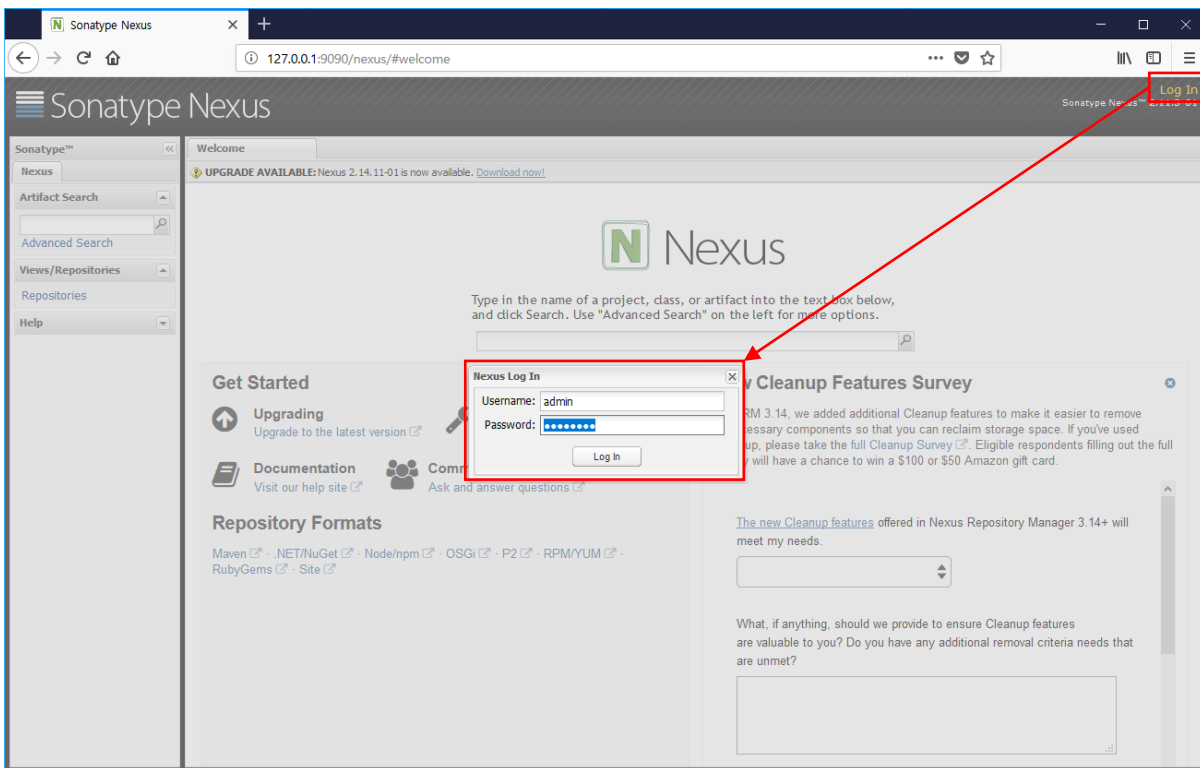
```
<repositories>
  <repository>
    <id>egovci</id>
    <url>http://127.0.0.1:8081/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>projectRepository</id>
    <name>Project Repository</name>
    <url>file://${project.basedir}/src/main/webapp/WEB-INF/lib</url>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <releases>
      <updatePolicy>never</updatePolicy>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
    <id>central</id>
    <name>Plugin Repository</name>
    <url>http://127.0.0.1:8081/nexus/content/groups/public/</url>
  </pluginRepository>
</pluginRepositories>
```

2.6 Nexus Repository 에 3rd Party 라이브러리 추가.

Nexus 저장소에 별도로 3rd Party 라이브러리를 별개로 등록 가능하다.

1. 관리자 로 로그인
 - ID : **admin**
 - Password : **admin123**

*** 기본 제공되는 비밀번호는 반드시 다른 것으로 변경하여 사용하시기 바랍니다.**



2. 저장소 목록에서 “3rd party” 저장소를 선택 한다.
3. 이후, Artifact Upload -> GA Definition -> GAV Parameters 를 차례대로 선택한다.
4. 추가할 라이브러리 Group, Artifact, Version 정보를 입력한다.
5. 정보 입력후 "Select Artifact(s) to Upload" 통해 라이브러리를 추가 한다.
6. 추가한 정보를 확인 후 "Add artifact" 통해 적용한다.

The screenshot shows the Sonatype Nexus web interface. The browser address bar indicates the URL: `127.0.0.1:9090/nexus/#view-repositories/thirdparty--uploadPanel`. The page title is "Sonatype Nexus" and the version is "2.11.3-01".

The main content area is titled "3rd party" and contains the "Artifact Upload" configuration panel. The panel includes the following sections:

- Select GAV Definition Source:** A dropdown menu is set to "GAV Parameter".
- Auto Guest:** A checkbox is checked.
- Group:** An empty text input field.
- Artifact:** An empty text input field.
- Version:** An empty text input field.
- Packaging:** A dropdown menu set to "Select...".
- Select Artifact(s) for Upload:** A section with a "Select Artifact(s) to Upload" button and three input fields for "Filename:", "Classifier:", and "Extension:". Below these is an "Add Artifact" button.
- Artifacts:** An empty list area with "Remove" and "Remove All" buttons.
- Buttons:** "Upload Artifact(s)" and "Reset" buttons are located at the bottom of the panel.

A table of repositories is visible at the top of the page:

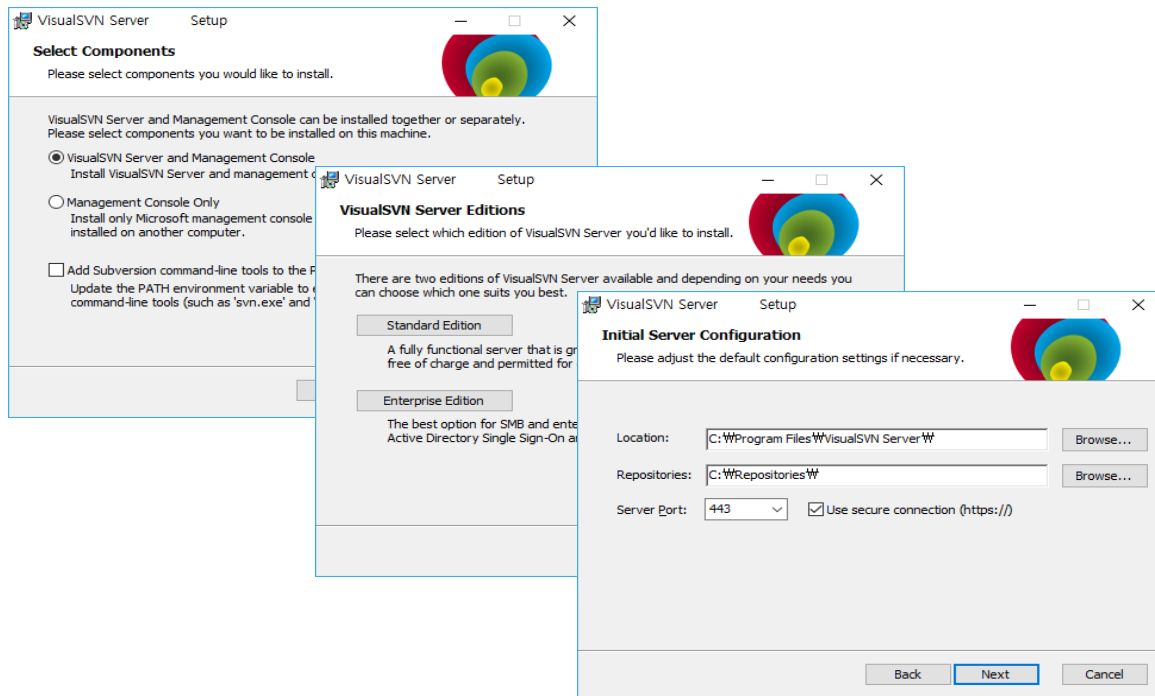
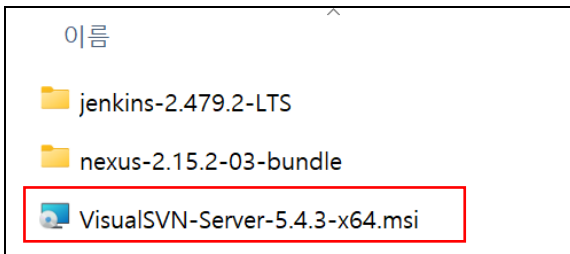
Repository	Type	Health Check	Format	Policy	Repository Status	Repository Path
Public Repositories	group	ANALYZE	maven2		In Service	http://127.0.0.1:9090/nexus/content/groups/public
3rd party	hosted	ANALYZE	maven2	Release	In Service	http://127.0.0.1:9090/nexus/content/repositories/thirdparty
Apache Snapshots	proxy	ANALYZE	maven2	Snapshot	In Service	http://127.0.0.1:9090/nexus/content/repositories/apache-snapshots
Central	proxy	ANALYZE	maven2	Release	In Service	http://127.0.0.1:9090/nexus/content/repositories/central
Central/M1 shadow	virtual	ANALYZE	maven1	Release	In Service	http://127.0.0.1:9090/nexus/content/shadows/central-m1
Codehaus Snapshots	proxy	ANALYZE	maven2	Snapshot	In Service - Remote Automatically Blo...	http://127.0.0.1:9090/nexus/content/repositories/codehaus-snapshots
Releases	hosted	ANALYZE	maven2	Release	In Service	http://127.0.0.1:9090/nexus/content/repositories/releases
Snapshots	hosted	ANALYZE	maven2	Snapshot	In Service	http://127.0.0.1:9090/nexus/content/repositories/snapshots

3. CI (SVN & JenkinsNS) 설정

3.1 Subversion 설치 (선택사항)

CI 설정에 앞서 소스 저장소(Subversion)을 설치한다. eGovCI에는 VisualSVN의 설치파일을 별도로 제공하고 있으며, 필요 시 설치하여 사용한다. (별도로 SVN을 구성하여 사용하더라도 무방하다.)

eGovCI의 VisualSVN-server-XXX.msi를 설치한다.



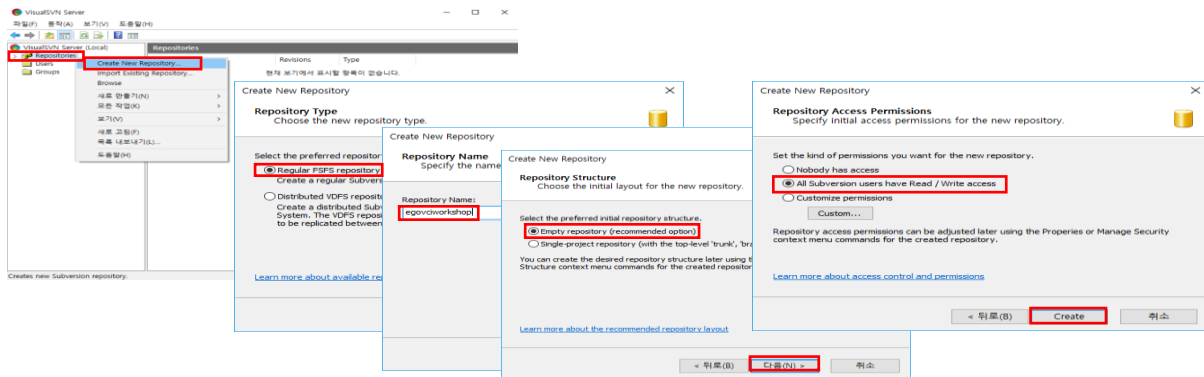
3.2 SVN Repositories 설정 및 사용자 등록.

SVN 저장소 설정 및 사용자 등록 관련 사항을 설정한다.

3.2.1 저장소 설정 및 등록

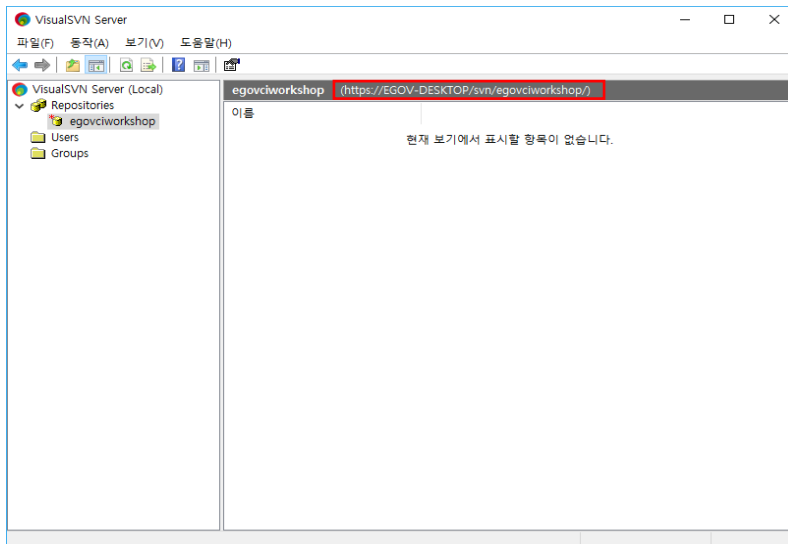
VisualSVN 서버의 구동을 확인 하고, "Repositories"의 오른쪽을 클릭하여 "Create New Repository..." 선택, 새로운 저장소를 만든다.

1. 임의 저장소 설정. (ex: egovciworkshop)
2. 필요한 사항을 설정하고 Create 를 한다.

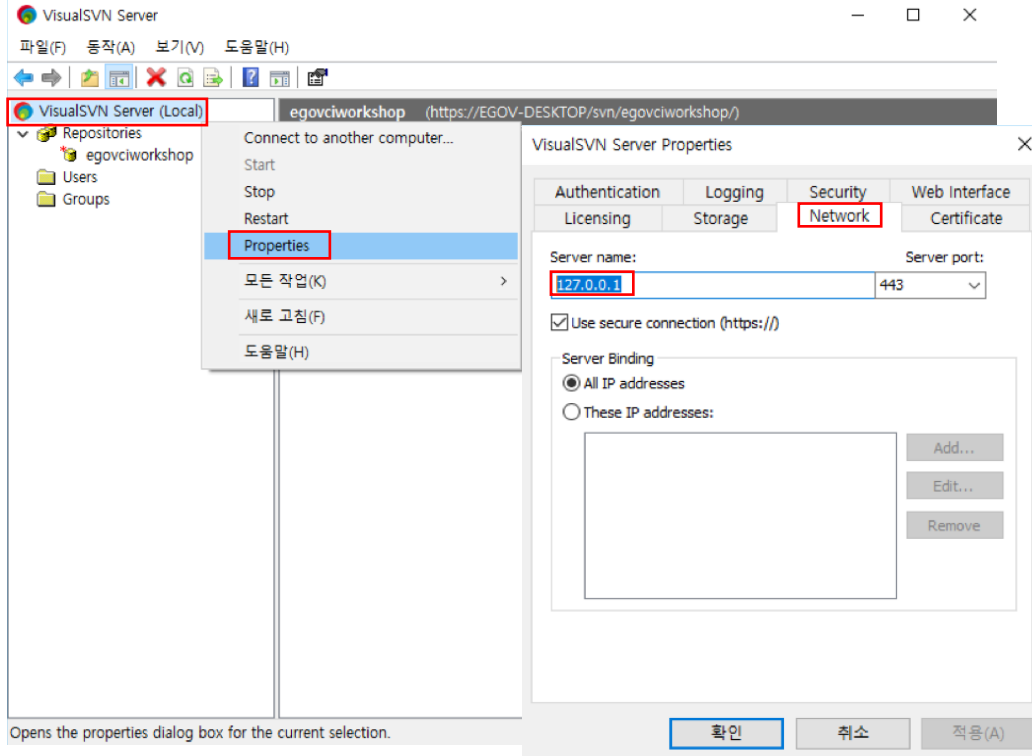


설정을 완료 후, 아래와 같이 저장소 위치를 확인 할 수 있다.

Repository location : "http://<localhost>/svn/egovciworkshop/"



** 참고로 <localhost> 주소를 아래와 같이 실제 IP 주소를 대치하여 사용 가능하다.

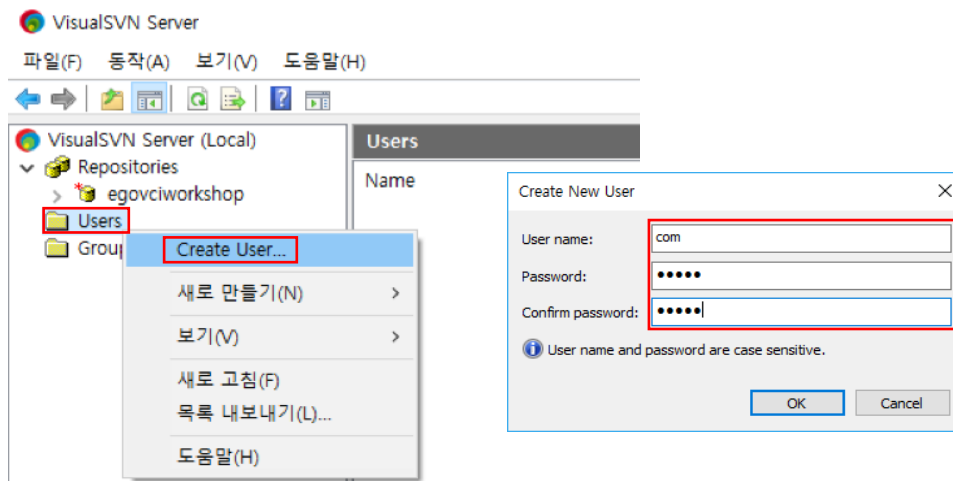


3.2.2 사용자 등록

아래와 같이 "User" 메뉴에서 새로운 사용자를 설정 가능하다.

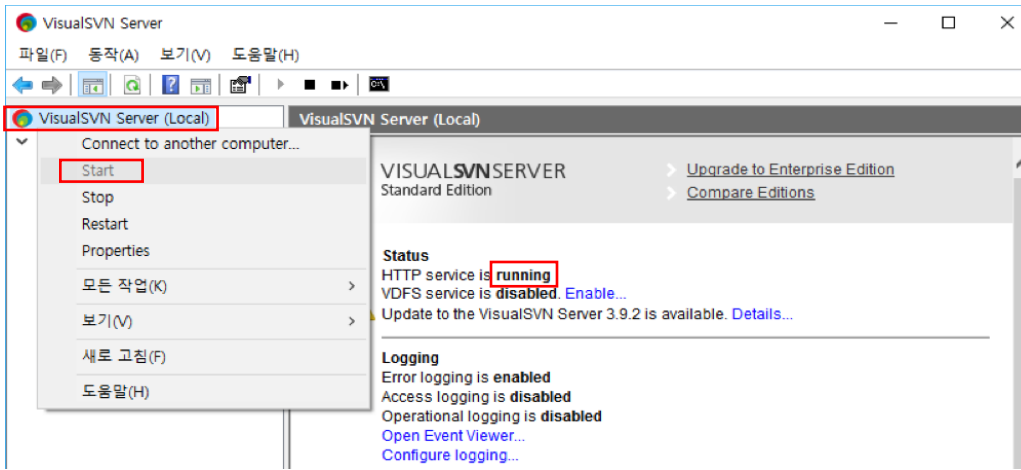
Ex) 사용자 이름: "com"

비밀번호 : "*****"



3.2.3 SVN 서버 시작/종료.

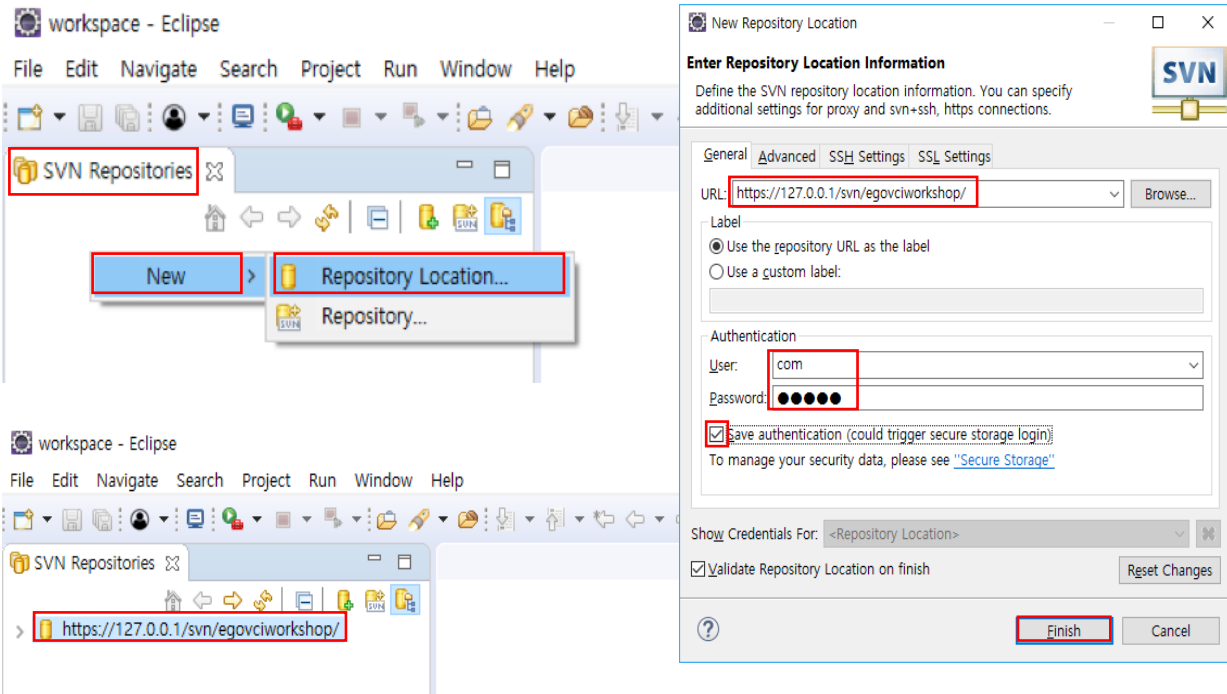
SVN 왼쪽 메뉴에서 아래와 같이 서버를 시작/종료를 할 수 있으며, 확인이 가능하다.



3.2.4 Eclipse 프로젝트를 SVN 서버에 등록.

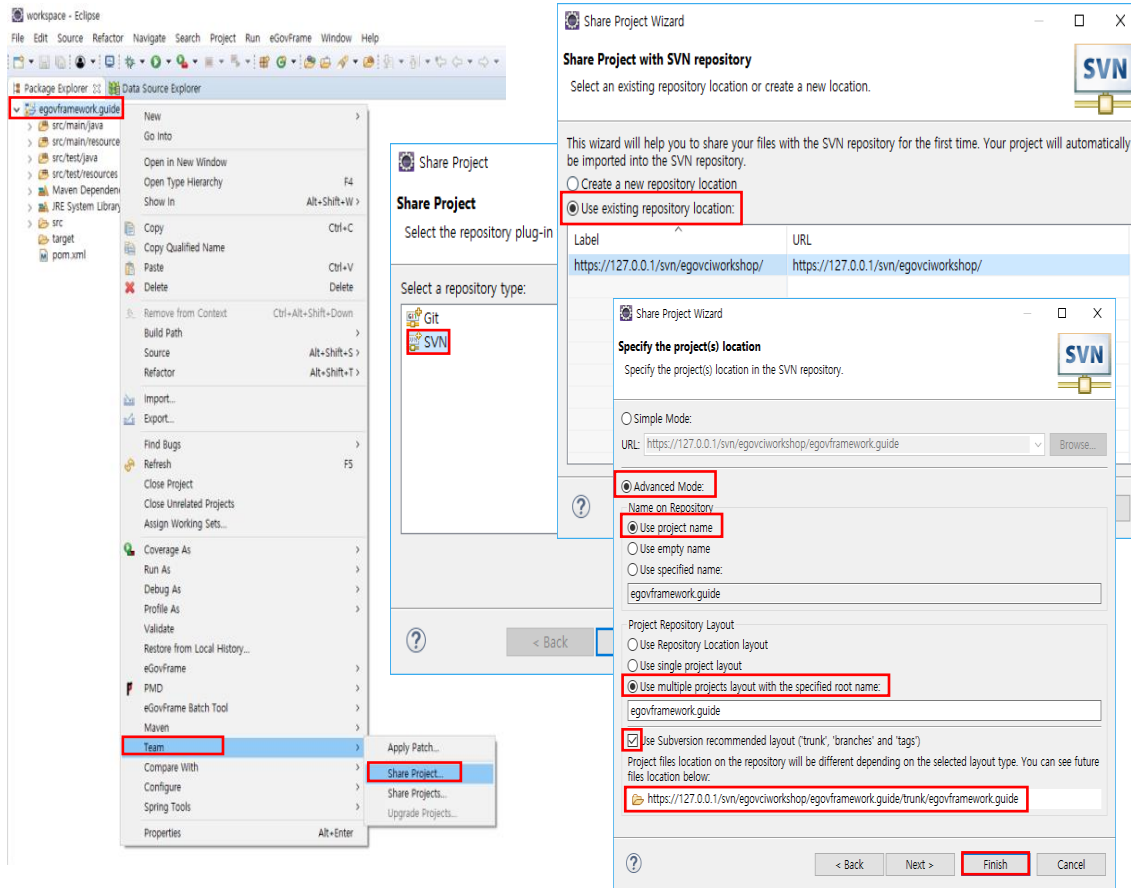
SVN 서버의 정보를 활용하여 Eclipse 내의 프로젝트를 등록하고 소스를 반영한다.

1. Eclipse 에서 SVN Repositories 를 연다.
2. 새로운 저장소 정보를 생성 및 SVN 정보를 등록한다.

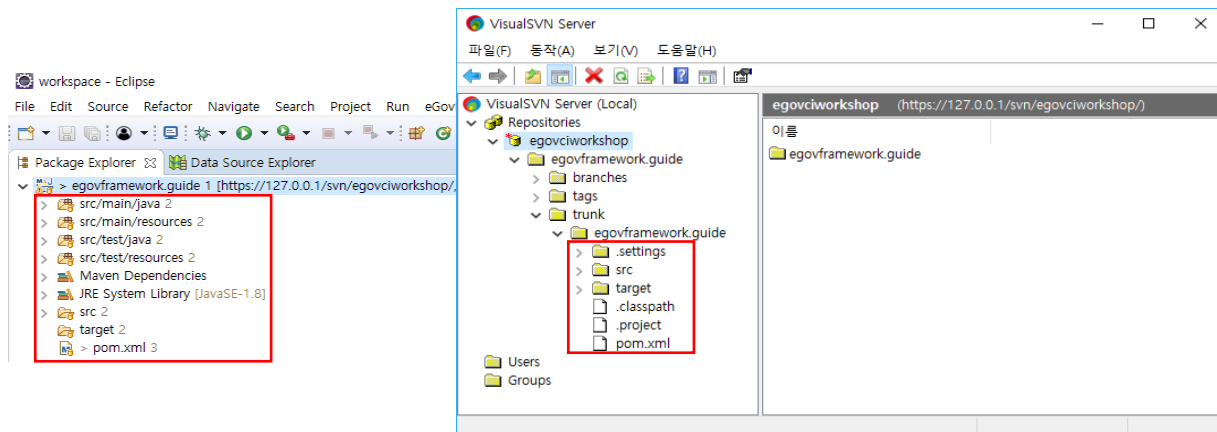


SVN 저장소 정보 등록 후, 프로젝트를 SVN 에 등록 하여야 한다.

1. 프로젝트 선택 후, 오른쪽 마우스를 클릭하여 “Team -> share Project” 로 프로젝트를 SVN 에 공유한다.



등록된 프로젝트는 SVN 서버 Manager 에서 다시 확인 가능하다.

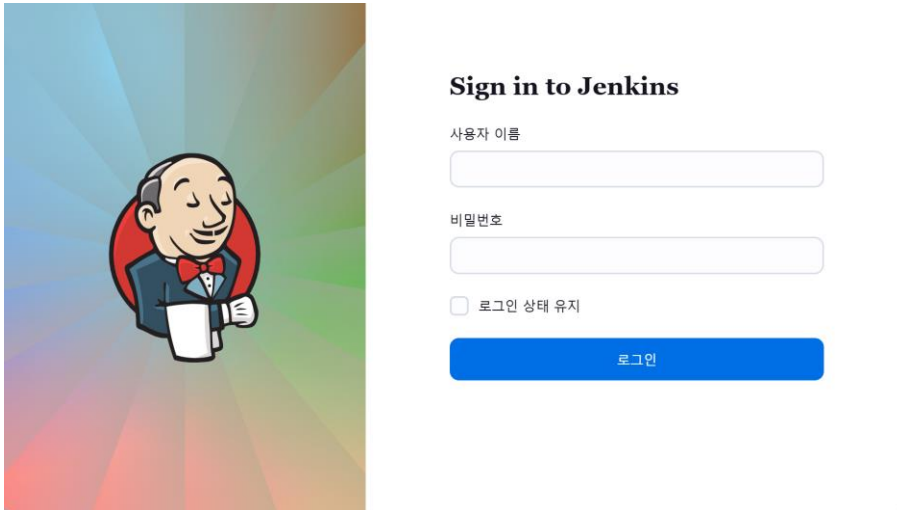


3.3 Jenkins 설정.

다음은 Jenkins 설정으로 아래와 같은 주소를 통해 접속이 가능하다.

<http://<localhost>:8082>

** 접속이 되지 않은 경우, "start-jenkins.bat"을 통해 Tomcat 를 실행한다.

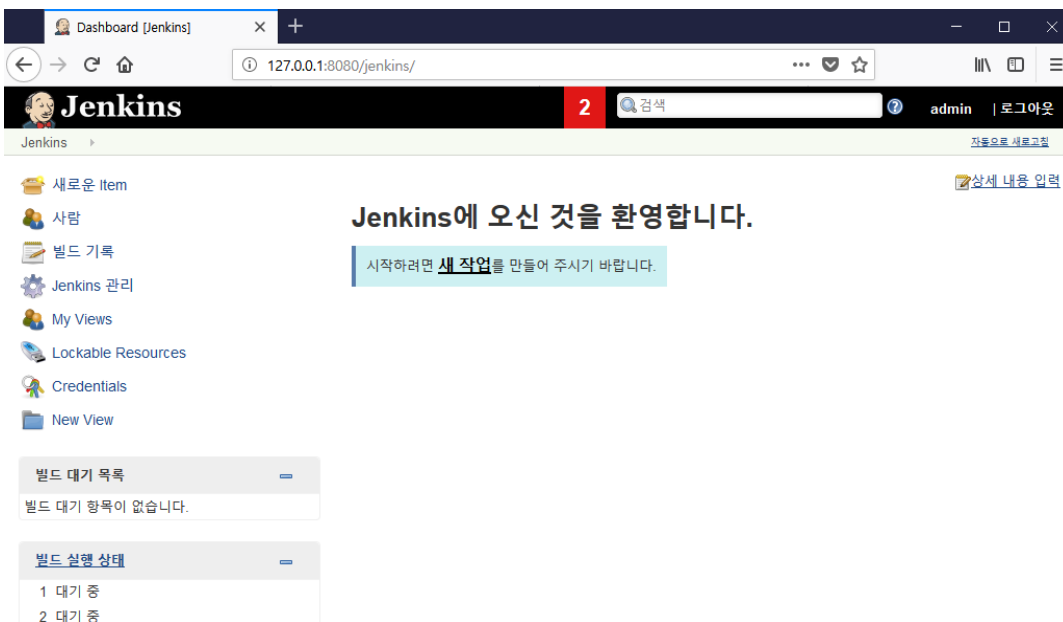


관리자로 로그인

- ID : admin
- Password : test1234

*** 기본 제공되는 비밀번호는 반드시 다른 것으로 변경하여 사용하시기 바랍니다.**

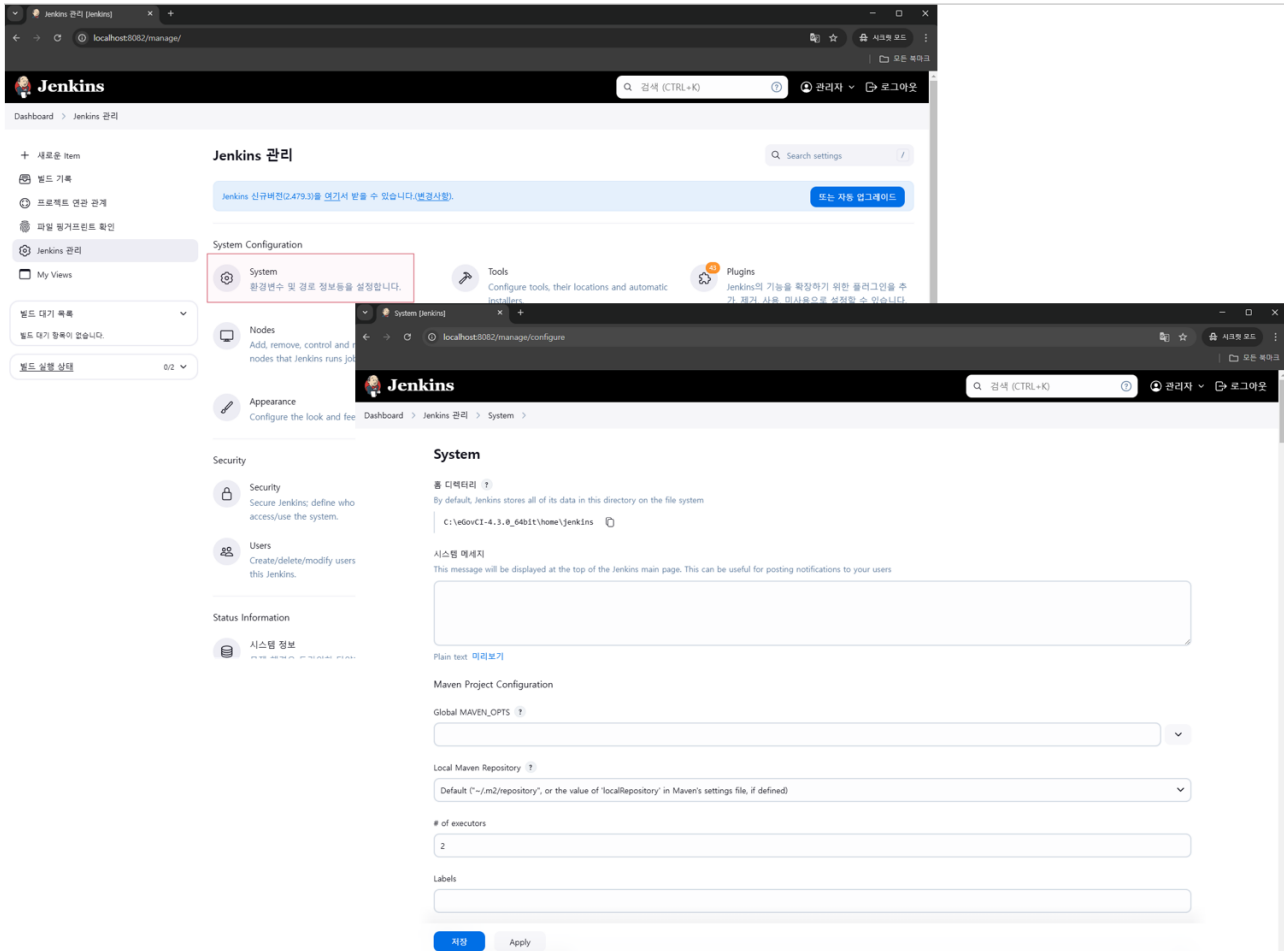
생성한 계정으로 로그인 한다.



3.3.1 Jenkins 기본 시스템 설정

Jenkins의 새 작업 등록 전, Jenkins의 기본 시스템 설정한다. 설정을 통하여 JDK, ANT, Maven, SVN 서버 정보를 등록하여 프로젝트의 컴파일, Deploy을 할 수 있다.

시스템 설정은 Jenkins 관리 -> System Configure 메뉴를 통해서 할 수 있다.



아래와 같이 ANT, MAVEN home 디렉토리를 설정한다.

Dashboard > Jenkins 관리

Jenkins 관리

System Configuration

- System**
환경변수 및 경로 정보등을 설정합니다.
- Plugins**
Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로 설정할 수 있습니다.
- Tools**
Configure tools, their locations and automatic installers.
- Nodes**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Maven installations

Maven installations ^ Edited

Add Maven

Maven

Name
Maven 3.9.9

MAVEN_HOME
C:\WeGovCI-4.3.0_64bit\bin\wapache-maven-3.9.9

Install automatically ?

Add Maven

인터넷 외부망과의 연결이 허용된 환경이라면, Install automatically 를 클릭하여 설치함으로써 Jenkins 가 Maven 을 직접 관리하도록 하는 방법도 가능하다.

Maven installations

Add Maven

≡ Maven ✕

Name

 Install automatically ?

≡ Install from Apache ✕

Version

Add Installer ▾

이후 Maven 의 사용자 정의 settings.xml 파일의 위치를 지정해줄 수 있다.

Tools

Maven Configuration

Default settings provider

Settings file in filesystem ?

File path ?

Default global settings provider

Use default maven global settings ?

Maven 대신에 Ant 로 빌드하고자 한다면 아래 화면과 같이 설정하면 된다.

또한 인터넷 외부망과의 연결이 허용된 환경이라면, **Install automatically** 를 클릭하여 설치함으로써 Jenkins 가 Ant 를 직접 관리하도록 하는 방법도 가능하다.

Ant installations

Ant installations ^

 Edited

Add Ant

☰ **Ant** ✕

Name

Ant 1.10.15

ANT_HOME

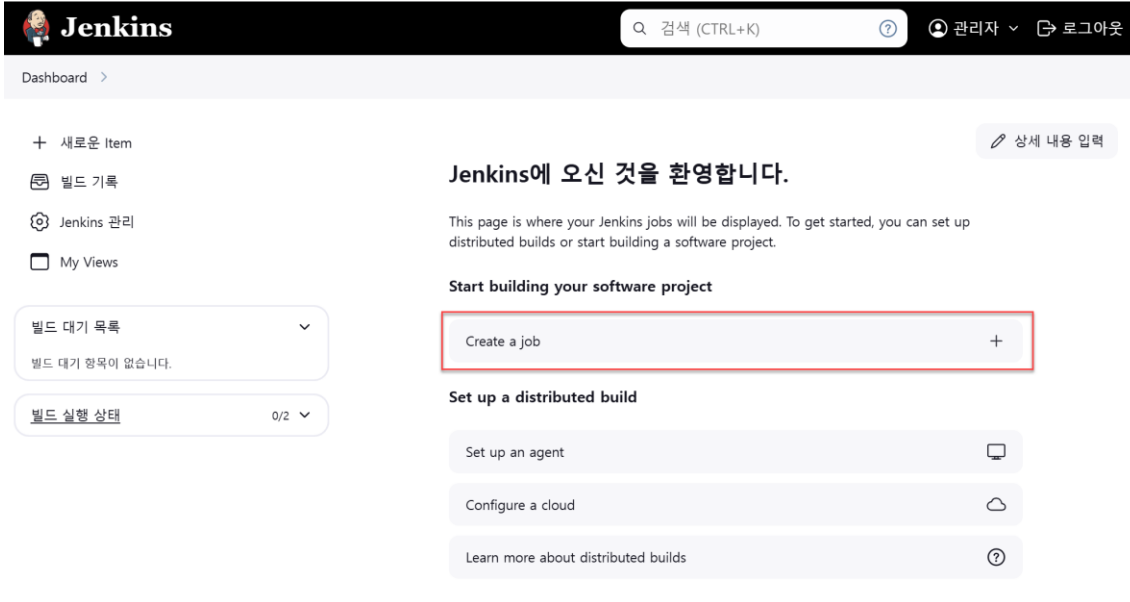
C:\WeGovCI-4.3.0_64bit\bin\apache-ant-1.10.15

Install automatically ?

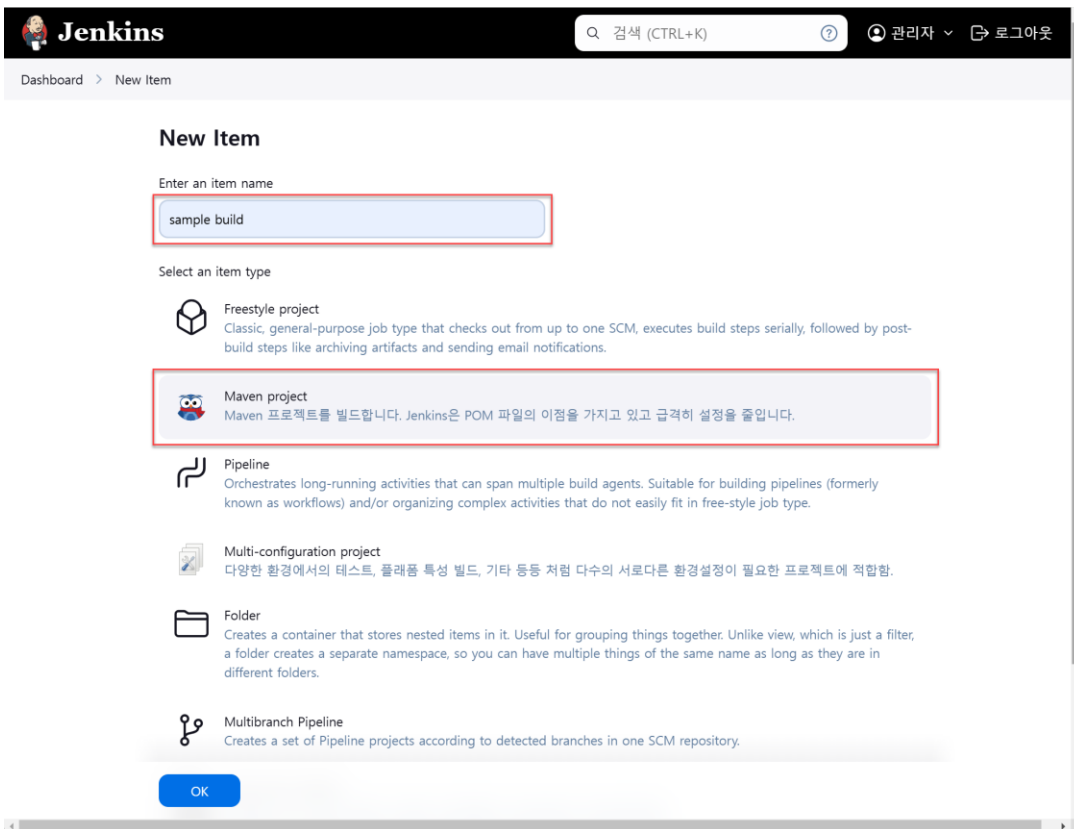
Add Ant

3.3.2 새 작업 생성

통합 환경에서 자동으로 프로젝트 **complie** 을 위해 아래와 같이 새 작업을 생성한다.



“Maven project”를 선택한다.



Source Code Management 에서 SVN 에 등록된 프로젝트 정보 및 사용자 정보를 아래와 같이 설정한다.

Configure

General

소스 코드 관리

빌드 유발

빌드 환경

Pre Steps

Build

Post Steps

Build Settings

빌드 후 조치

소스 코드 관리

None

Git ?

Subversion ?

Modules

Repository URL ?

https://192.168.100.153/svn/egovci/sample/trunk/test.bbs

Credentials

egov/*****

+ Add

Local module directory ?

.

Repository depth ?

infinity

Ignore externals ?

Cancel process on externals fail ?

빌드 유발

Build whenever a SNAPSHOT dependency is built ?

Schedule build when some upstream has no successful builds ?

빌드를 원격으로 유발 (예: 스크립트 사용) ?

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

빌드 환경

Delete workspace before build starts

고급

Use secret text(s) or file(s) ?

Add timestamps to the Console Output

Inspect build log for published build scans

Terminate a build if it's stuck

With Ant ?

Goals: type "clean install" (which means of MVN clean and install command)

Pre Steps

Add pre-build step ▾

Build

Root POM ?

pom.xml

Goals and options ?

clean install

만약 프로젝트(아이템) 별로 Maven 설정을 개별적으로 관리하고 싶다면, 고급(Advanced)을 클릭한 후 아래 이미지와 같이 설정하면 된다.

Advanced ^

MAVEN_OPTS ?

- Incremental build - only build changed modules ?
- Disable automatic artifact archiving ?
- Disable automatic site documentation artifact archiving ?
- Disable automatic fingerprinting of consumed and produced artifacts ?
- Enable triggering of downstream projects ?
 - Block downstream trigger when building ?
- Build modules in parallel ?
- Use private Maven repository ?
- Resolve Dependencies during Pom parsing
- Run Headless ?
- Process Plugins during Pom parsing
- Use custom workspace ?

Maven Validation Level

DEFAULT ▾

Settings file ?

Settings file in filesystem ▾ ?

File path ?

Global Settings file ?

Use default maven global settings ▾ ?

이후 저장을 누르면 끝난다.

빌드 후 조치

빌드 후 조치 추가 ▾

저장

Apply

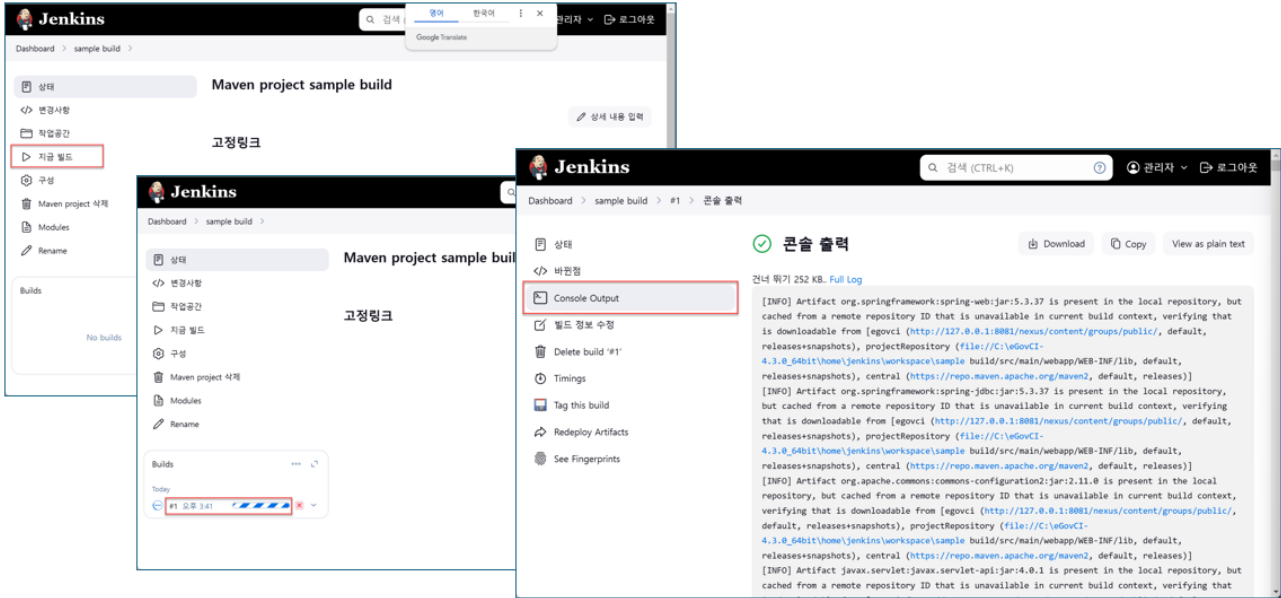
The screenshot shows the Jenkins dashboard interface. At the top, there is a search bar and user management options. The main content area displays a table of build history for a project named 'sample build'. The table has columns for days of the week (S, W), build name, success/failure status, duration, and a play button for details. Below the table, there are configuration options for '아이콘' (icon) and 'My Views'. On the left sidebar, there are navigation links for '새로운 Item', '빌드 기록', '프로젝트 연관 관계', '파일 핑거프린트 확인', 'Jenkins 관리', and 'My Views'. At the bottom right, it shows 'REST API' and 'Jenkins 2.479.2'.

S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
✓	☀	sample build	1 hr 13 min #1	—	28 sec

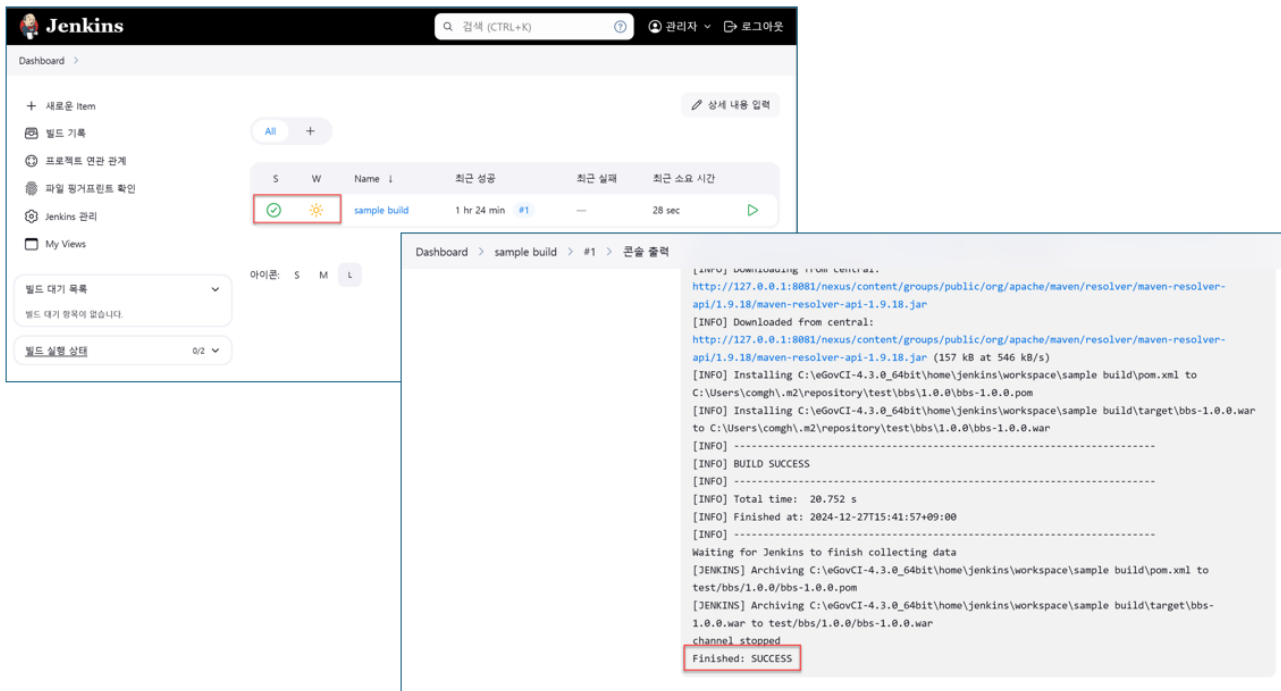
3.3.3 작업 Bulid

모든 설정 이후, 최초 수동으로 **Bulid** 하여 설정을 확인 할 수 있다.

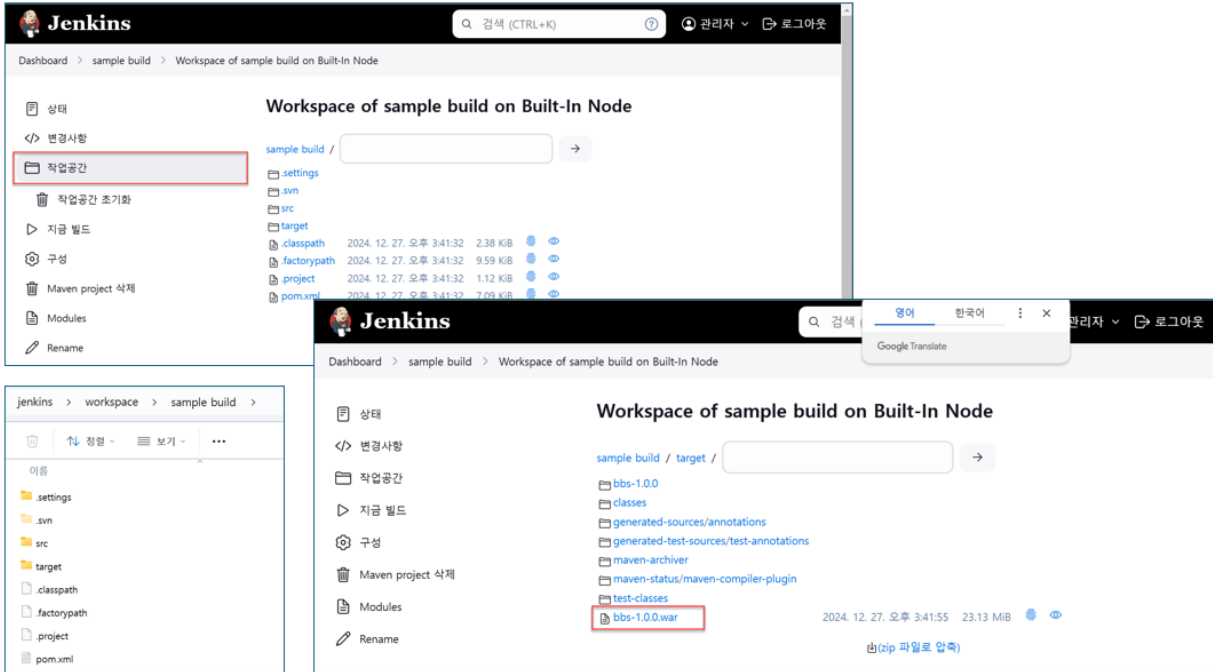
아래와 같이 **Build Now** 메뉴를 통하여, **Maven build** 를 실행한다. 빌드 실행 시 콘솔을 통하여 빌드 프로세스를 확인할 수 있다.



아래와 같이 작업 아이콘 또는 콘솔의 정보로 빌드를 확인한다.



정상적으로 빌드가 끝나면, Workspace 에서 빌드를 통하여 만들어진 war(servlet 웹프로젝트의 경우) / jar(자바프로젝트의 경우) 파일을 확인할 수 있다.



* 기본 제공되는 “초기 비밀번호”는 반드시 다른 것으로 변경하여 사용하시기 바랍니다.