

**표준프레임워크**  
**CI 및 Nexus 환경 구축 가이드**  
**Version 5.0**

---

# Index

<b>1. 표준프레임워크 EGOVCI 패키지 설치</b> .....	<b>3</b>
<b>1.1 개요</b> .....	<b>3</b>
<b>1.2 EGOVCI 압축풀기</b> .....	<b>3</b>
<b>1.3 EGOVCI 시스템 구성</b> .....	<b>4</b>
<b>2. NEXUS REPOSITORY 3</b> .....	<b>5</b>
<b>2.1 NEXUS REPOSITORY 서버 구동</b> .....	<b>5</b>
<b>2.2 NEXUS REPOSITORY 설정</b> .....	<b>7</b>
2.2.1 Nexus data directory 변경 .....	7
2.2.2 Port 및 Context-path 변경 .....	8
<b>2.3 NEXUS REPOSITORY 에 라이브러리 추가</b> .....	<b>8</b>
2.3.1 Nexus Repository 3 라이브러리 관리 방식 변경 .....	8
2.3.2 소량의 라이브러리 Upload .....	9
2.3.3 대량의 라이브러리 Upload .....	10
<b>2.4 NEXUS REPOSITORY URL 을 POM.XML 에 반영</b> .....	<b>13</b>
<b>2.5 NEXUS REPOSITORY 2 → 3 마이그레이션(업그레이드)</b> .....	<b>14</b>
<b>3. CI - SVN</b> .....	<b>15</b>
<b>3.1 SVN 설치 (선택사항)</b> .....	<b>15</b>
<b>3.2 SVN REPOSITORIES 설정 및 사용자 등록</b> .....	<b>15</b>
3.2.1 저장소 설정 및 등록 .....	16
3.2.2 사용자 등록 .....	17
<b>3.3 SVN 서버 시작/종료</b> .....	<b>18</b>
<b>3.4 ECLIPSE 프로젝트를 SVN 서버에 등록</b> .....	<b>18</b>
<b>4. CI - JENKINS</b> .....	<b>20</b>
<b>4.1 JENKINS 서버 구동</b> .....	<b>20</b>
<b>4.2 JENKINS 플러그인 목록</b> .....	<b>22</b>
<b>4.3 JENKINS 설정</b> .....	<b>24</b>
4.3.1 Jenkins 설정 : Port .....	24
4.3.2 Jenkins 설정 : System .....	24
4.3.3 Jenkins 설정 : Tools .....	25
<b>4.4 JENKINS 작업</b> .....	<b>27</b>
4.4.1 작업 생성 .....	27
4.4.2 작업 Bulid .....	31

# 1. 표준프레임워크 eGovCI 패키지 설치

## 1.1 개요

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

- eGovCI 패키지 (Windows용 zip 파일 또는 Linux 용 tar.gz 파일)
  - Nexus repository 3
  - CI - SVN (Windows용 VisualSVN Server 포함, Linux는 미포함 - 별도 설치 필요)
  - CI - Jenkins
  - 기본 접속 설정 URL
    - NEXUS : http://<ip\_address>:8081/
    - JENKINS : http://<ip\_address>:8082/

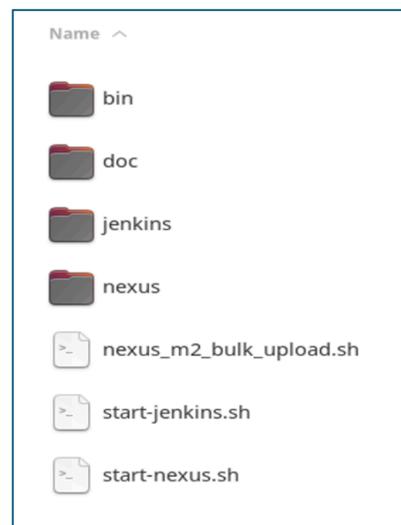
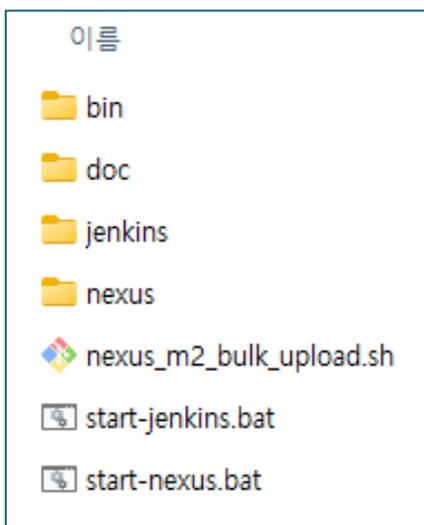
## 1.2 eGovCI 압축풀기

전자정부 표준프레임워크에서 제공하는 eGovCI ("eGovCI-<Version>-<OS>-<bit>") 파일을 다음 권장 경로에 압축을 풉니다.

- Windows : C:\₩
- Linux : /home/<사용자>

압축을 푼 구조는 아래와 같습니다.

< Windows : C:\₩eGovCI-5.0.0-Windows-64bit >    < Linux : /home/<사용자>/eGovCI-5.0.0-Linux-64bit >



## 1.3 eGovCI 시스템 구성

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

- Apache Ant 1.10.15 (under /bin dictory)
- Apache Maven 3.9.9 (under /bin dictory)
- Eclipse Temurin OpenJDK 17.0.17+10 (under /bin dictory)
- Linux용 패키지에는 Linux/x64 버전의 JDK가 설치되어 있습니다. PC가 x64가 아니라 AArch64 등 다른 버전이라면, 해당 버전과 호환되는 JDK를 설치해 주시기 바랍니다.
- Nexus Repository 3.86.2-01 (under /nexus/nexus-app dictory)
- Jenkins 2.528.3 LTS (under /jenkins/jenkins-app dictory)
- VisualSVN-Server-5.4.4-x64.msi (under /bin dictory)
  - 별도 설치형 파일 제공 (Linux용 서버 개발환경에는 제외)

## 2. Nexus Repository 3

### 2.1 Nexus Repository 서버 구동

CI 시스템 구성 이후, start-nexus.bat 또는 start-nexus.sh 스크립트 파일을 이용하여 Nexus 서버를 구동하고 종료할 수 있습니다.

#### Windows 운영체제 주의사항

- 구동 전에 Nexus 서비스를 Windows에 등록하는 사전 절차가 필요합니다. Windows에 Nexus 서비스를 등록하기 위해서는 관리자의 권한으로 실행된 cmd 또는 PowerShell에서 서비스 등록 명령을 실행해야 합니다.
- Nexus 서비스 등록은 Nexus 서비스 미등록 상태에서 Nexus 서버 구동 전 최초 1회에 한해 필요합니다.
- Nexus 서비스 등록/삭제 방법
  - 관리자의 권한으로 실행된 PowerShell일 경우 명령어
    - 서비스 등록 : .\start-nexus.bat install
    - 서비스 삭제 : .\start-nexus.bat uninstall
  - 관리자의 권한으로 실행된 cmd일 경우 명령어
    - 서비스 등록 명령어 : start-nexus.bat install
    - 서비스 삭제 명령어 : start-nexus.bat uninstall

#### Linux 운영체제 주의사항

- 구동 전에 JDK 확인이 필요합니다. Linux용 패키지에는 Linux/x64 버전의 JDK가 설치되어 있습니다.
- PC가 x64가 아니라 AArch64 등 다른 버전이라면 해당 버전과 호환되는 JDK를 설치해 주시기 바랍니다.
- 그 후 start-nexus.sh 스크립트 파일에서 INSTALL4J\_JAVA\_HOME\_OVERRIDE의 값을 위 JDK 설치 경로로 수정해 주시기 바랍니다.

```
# CI_HOME should be anchored to the script directory.
SCRIPT_DIR="$(cd "$(dirname "$0")" && pwd)"
export CI_HOME="$SCRIPT_DIR"

# JDK 17: use the CI-bundled JDK (shared with Jenkins) instead of Nexus embedded JDK.
export INSTALL4J_JAVA_HOME_OVERRIDE="$CI_HOME/bin/jdk-17.0.17+10"

# bundled Nexus location
export NEXUS_APP_HOME="$CI_HOME/nexus/nexus-app"
NEXUS_BIN="$NEXUS_APP_HOME/bin/nexus"

# 입력 인자 개수($#)가 0이면(아무 명령도 안 주면) 기본 동작으로 "start-nexus.sh start" 실행
if [ $# -eq 0 ]; then
    DEFAULTED_CMD=1
    set -- start
fi
```

## Nexus 서버 구동 방법

- Windows
  - 실행 명령어 : start-nexus.bat start (또는 start-nexus.bat 파일 더블클릭)
  - 종료 명령어 : start-nexus.bat stop
- Linux
  - 실행 명령어 : ./start-nexus.sh start
  - 종료 명령어 : ./start-nexus.sh stop

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

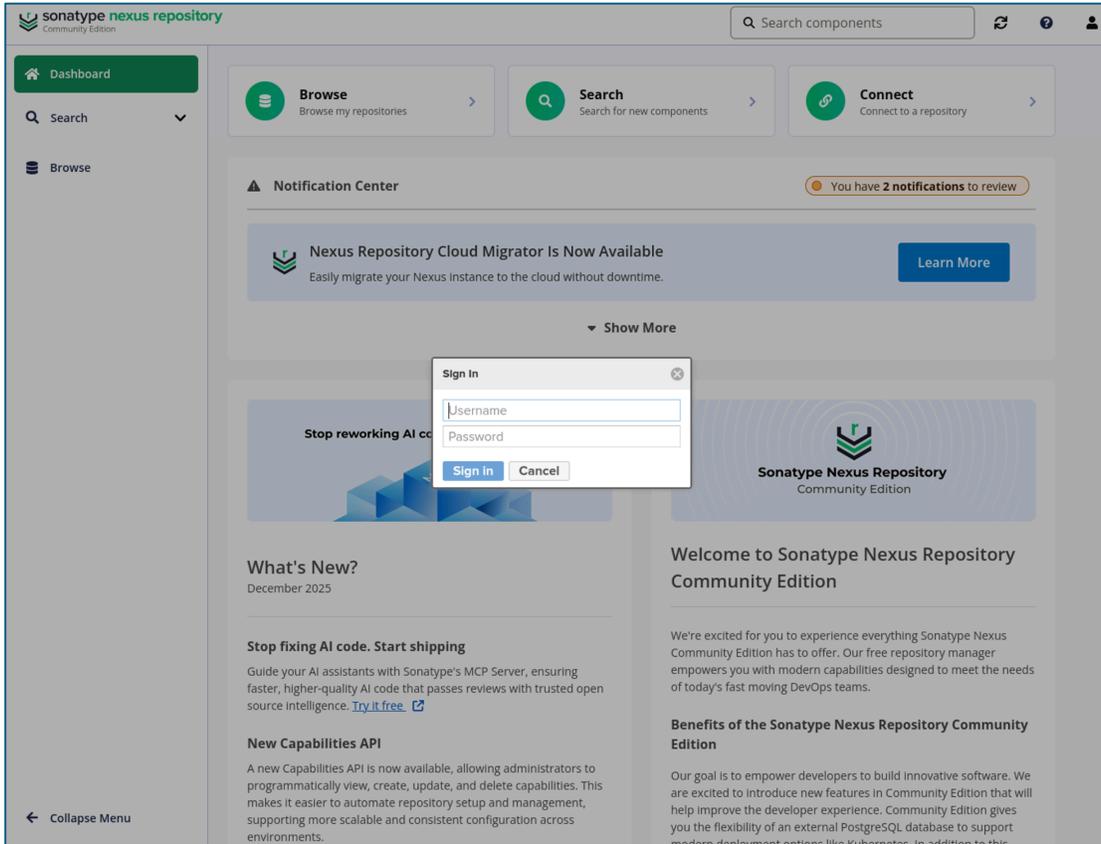
```
egov69@egov69:~/eGovCI-5.0.0-Linux-64bit$ ./start-nexus.sh start
-----
Nexus runner
-----
Using CI_HOME: "/home/egov69/eGovCI-5.0.0-Linux-64bit"
Using NEXUS_APP_HOME: "/home/egov69/eGovCI-5.0.0-Linux-64bit/nexus/nexus-app"
Using NEXUS_BIN: "/home/egov69/eGovCI-5.0.0-Linux-64bit/nexus/nexus-app/bin/nexus"
Using INSTALL4J_JAVA_HOME_OVERRIDE: "/home/egov69/eGovCI-5.0.0-Linux-64bit/bin/jdk-17.0.17+10"

Usage:
./start-nexus.sh start      (run in background)
./start-nexus.sh stop      (stop)
./start-nexus.sh run       (run in foreground for debugging; stop with Ctrl+C)

Access:
http://127.0.0.1:8081/ (default port)

$CI_HOME = /home/egov69/eGovCI-5.0.0-Linux-64bit
$INSTALL4J_JAVA_HOME_OVERRIDE = /home/egov69/eGovCI-5.0.0-Linux-64bit/bin/jdk-17.0.17+10
$NEXUS_APP_HOME = /home/egov69/eGovCI-5.0.0-Linux-64bit/nexus/nexus-app
-----
Starting nexus
```

기본 접속 URL : <http://localhost:8081/>



## 사용자 계정

- User Name : admin
- Password : admin123

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

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

## 2.2 Nexus Repository 설정

### 2.2.1 Nexus data directory 변경

Nexus 의 Data 저장소는 기본적으로 "<CI\_HOME>/nexus/sonatype-work/nexus3"으로 설정되어 있으며, nexus.vmoptions 파일에 Dkaraf.data 값 수정을 통해 별도 설정도 가능합니다.

- 파일 경로 : <CI\_HOME>/nexus/nexus-app/bin/nexus.vmoptions

```
-Xms2703m  
-Xmx2703m  
-XX:+UnlockDiagnosticVMOptions  
-XX:+LogVMOutput  
-XX:LogFile=./sonatype-work/nexus3/log/jvm.log  
-XX:-OmitStackTraceInFastThrow  
-Dkaraf.home=.  
-Dkaraf.base=.  
-Djava.util.logging.config.file=etc/spring/java.util.logging.properties  
-Dkaraf.data=./sonatype-work/nexus3  
-Dkaraf.log=./sonatype-work/nexus3/log  
-Djava.io.tmpdir=./sonatype-work/nexus3/tmp  
-Djdk.tls.ephemeralDHKeySize=2048  
-Dfile.encoding=UTF-8
```

## 2.2.2 Port 및 Context-path 변경

Nexus의 Port는 기본적으로 “8081,” nexus-context-path는 “/”으로 설정되어 있으며, nexus.properties 파일에 application-port와 nexus-context-path 값 수정을 통해 별도 설정도 가능합니다.

- 파일 경로 : <CI\_HOME>/nexus/sonatype-work/nexus3/etc/nexus.properties

```
# Jetty section  
application-port=8081  
# application-host=0.0.0.0  
# nexus-args=${jetty.etc}/jetty.xml,${jetty.etc}/jet  
# nexus-context-path=/  

```

## 2.3 Nexus Repository 에 라이브러리 추가

### 2.3.1 Nexus Repository 3 라이브러리 관리 방식 변경

이전 버전인 Nexus Repository 2는 File Storage 기반 라이브러리 저장 방식이었습니다. 따라서 라이브러리를 추가하기 위해 Nexus의 Upload 기능을 사용하는 것 이외에 maven 라이브러리를 직접 이동하는 방식으로 라이브러리를 추가할 수 있었습니다.

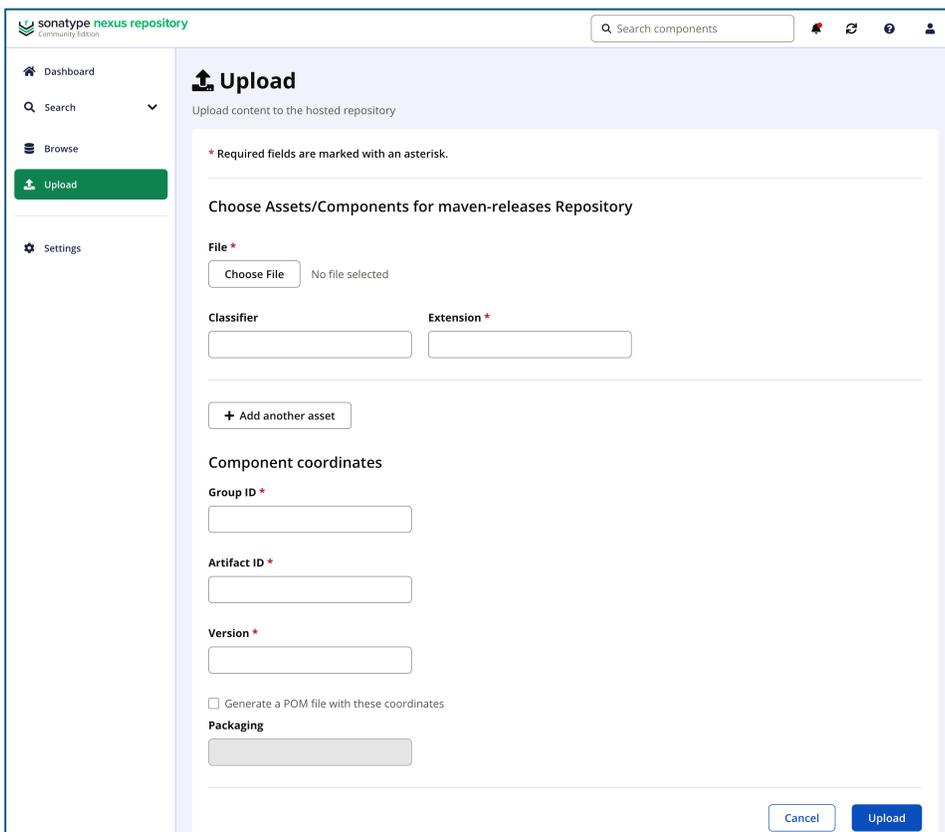
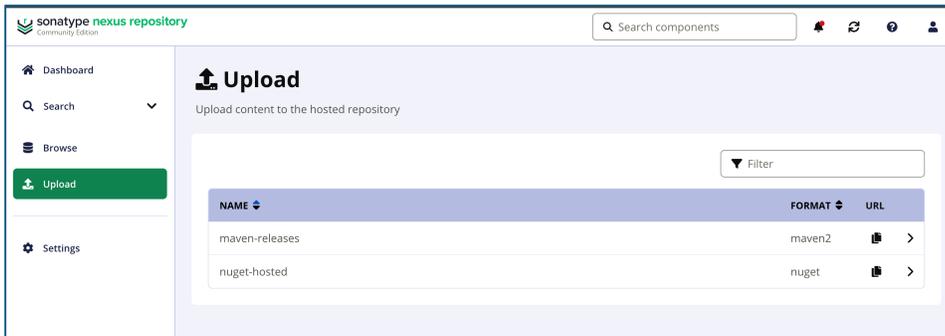
반면 신규 버전인 Nexus Repository 3는 Blob store 기반 라이브러리 저장 방식으로 변경되었습니다. 그 결과 라이브러리를 추가하기 위해 반드시 Nexus의 Upload 기능을 활용해야 합니다.

이 목차에서 안내할 내용은 다음 3가지입니다.

- 소량의 라이브러리를 Upload 하는 방법
- 대량의 라이브러리를 Upload 하는 방법
- Nexus Repository 2의 Repository를 Nexus Repository 3로 마이그레이션 하는 방법

## 2.3.2 소량의 라이브러리 Upload

소량의 라이브러리를 Upload 하고자 한다면 다음 Upload 기능을 활용해 주시기 바랍니다.

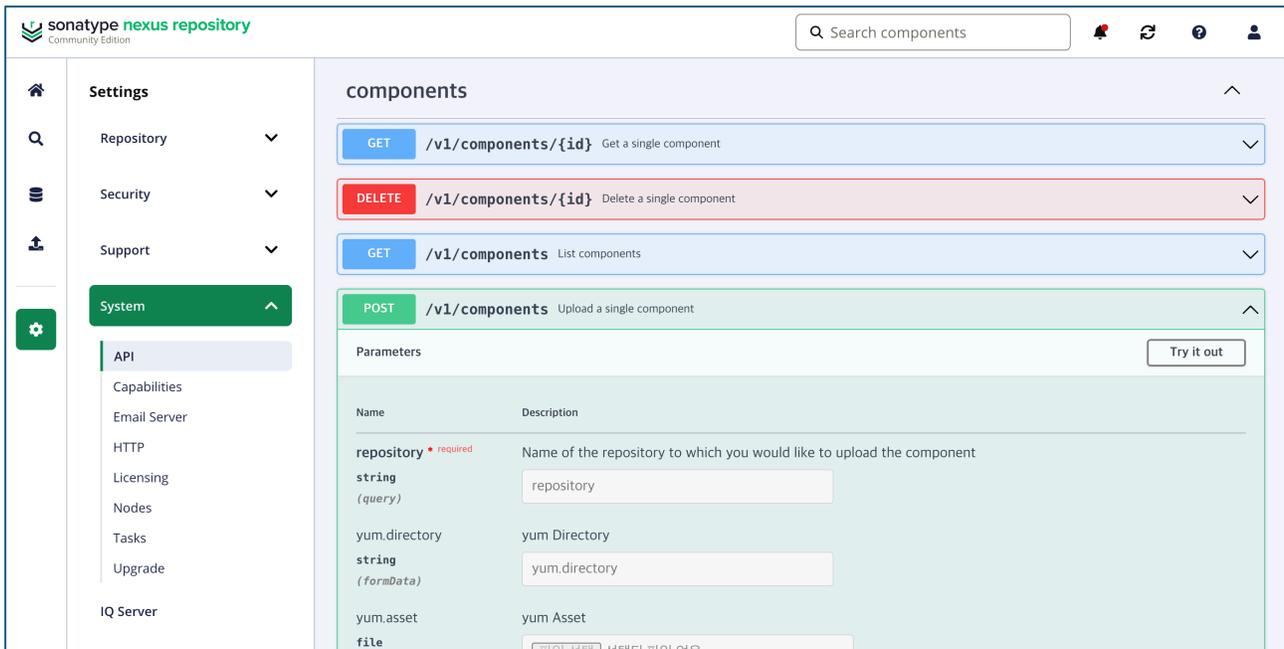


### 2.3.3 대량의 라이브러리 Upload

Nexus Repository 3 는 대량의 라이브러리를 한 번에 Upload 하는 기능이 없습니다. 현실적으로 위 Upload 기능으로 대량의 라이브러리들을 업로드하는 데에는 많은 시간이 소모됩니다.

위 불편함을 해소하고자 eGovCI 패키지 내에 스크립트 파일(nexus\_m2\_bulk\_upload.sh)을 제공하고 있습니다. 이 스크립트 파일은 Nexus Repository 3 의 POST /service/rest/v1/components 를 활용합니다.

메뉴 : Settings > System > API > components



Nexus\_m2\_bulk\_upload.sh 스크립트 실행을 위한 인자는 필수 5 가지(root, nexus, repo, user, pass)와 옵션(dry-run)이 있습니다.

실행 명령 코드를 설명드리겠습니다.

- 1) 실제 업로드 전 드라이런(권장)

```
./nexus_m2_bulk_upload.sh ₩
--root "<MAVEN-REPOSITORY>" ₩
--nexus "http://<NEXUS-HOST>:8081" ₩
--repo "<NEXUS-REPOSITORY-NAME>" ₩
--user "<NEXUS-USER-NAME>" ₩
--pass "<NEXUS-USER-PASSWORD>" ₩
--dry-run
```

- 2) 실제 업로드

```
./nexus_m2_bulk_upload.sh ₩  
--root "<MAVEN-REPOSITORY>" ₩  
--nexus "<NEXUS-HOST:PORT>" ₩  
--repo "<NEXUS-REPOSITORY-NAME>" ₩  
--user "<NEXUS-USER-NAME>" ₩  
--pass "<NEXUS-USER-PASSWORD>" ₩  
* 권한 문제 발생시 : chmod +x nexus_m2_bulk_upload.sh
```

위 코드에서 적혀 있는 변수는 다음과 같습니다.

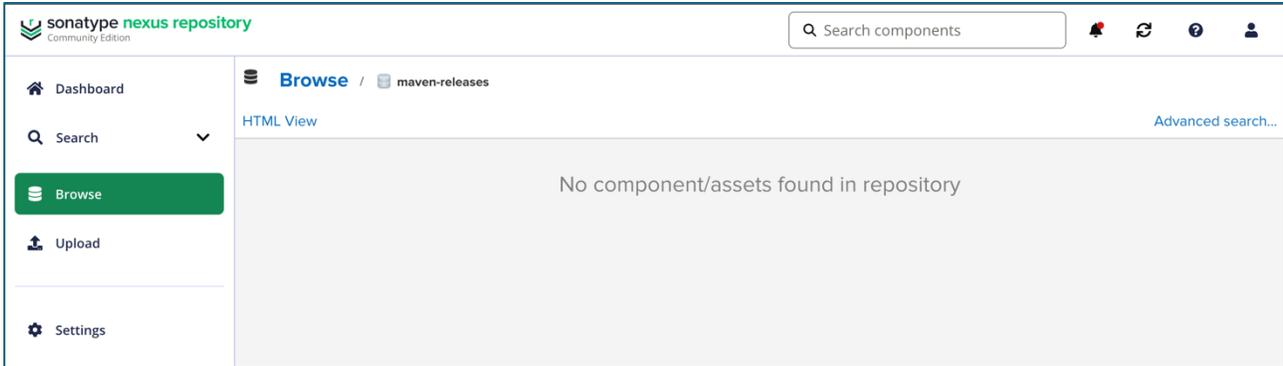
- <MAVEN-REPOSITORY> : 로컬 Maven 저장소입니다. groupId/artifactId/version 방식으로 폴더 디렉토리가 관리되는 곳이어야 합니다. 기본적으로 <USER-HOME>/.m2/repository 경로가 로컬 Maven 저장소입니다. 이 값에는 절대경로 또는 스크립트 파일 기준 상대경로가 들어갈 수 있습니다.
- <NEXUS-HOST:PORT> : Nexus가 실행중인 서버의 URL입니다. nexus.properties 파일을 수정하지 않았다면, http://127.0.0.1:8081을 입력하시면 됩니다.
- <NEXUS-REPOSITORY-NAME> : 라이브러리를 업로드할 Nexus Repository 이름입니다.
- <NEXUS-USER-NAME> : Nexus 사용자명입니다. 관리자 계정의 사용자명 기본값은 admin입니다.
- <NEXUS-USER-PASSWORD> : 위 Nexus 사용자명의 비밀번호입니다. 관리자 계정의 비밀번호 기본값은 admin123입니다.

실제 업로드 명령 코드 예시는 다음과 같습니다.

```
- Windows  
.\nexus_m2_bulk_upload.sh ₩  
--root "C:\Users\₩<USER-NAME>\₩.m2\₩repository" ₩  
--nexus "http://127.0.0.1:8081" ₩  
--repo "maven-releases" ₩  
--user "admin" ₩  
--pass "admin123" ₩  
- Linux  
./nexus_m2_bulk_upload.sh ₩  
--root "/home/<USER-NAME>/.m2/repository" ₩  
--nexus "http://127.0.0.1:8081" ₩  
--repo "maven-releases" ₩  
--user "admin" ₩  
--pass "admin123" ₩
```

이제 스크립트를 실행해 보겠습니다.

스크립트 실행 전 Nexus 의 maven-releases 레포지토리는 비어 있는 상태입니다.



스크립트를 실행한 터미널 화면입니다.

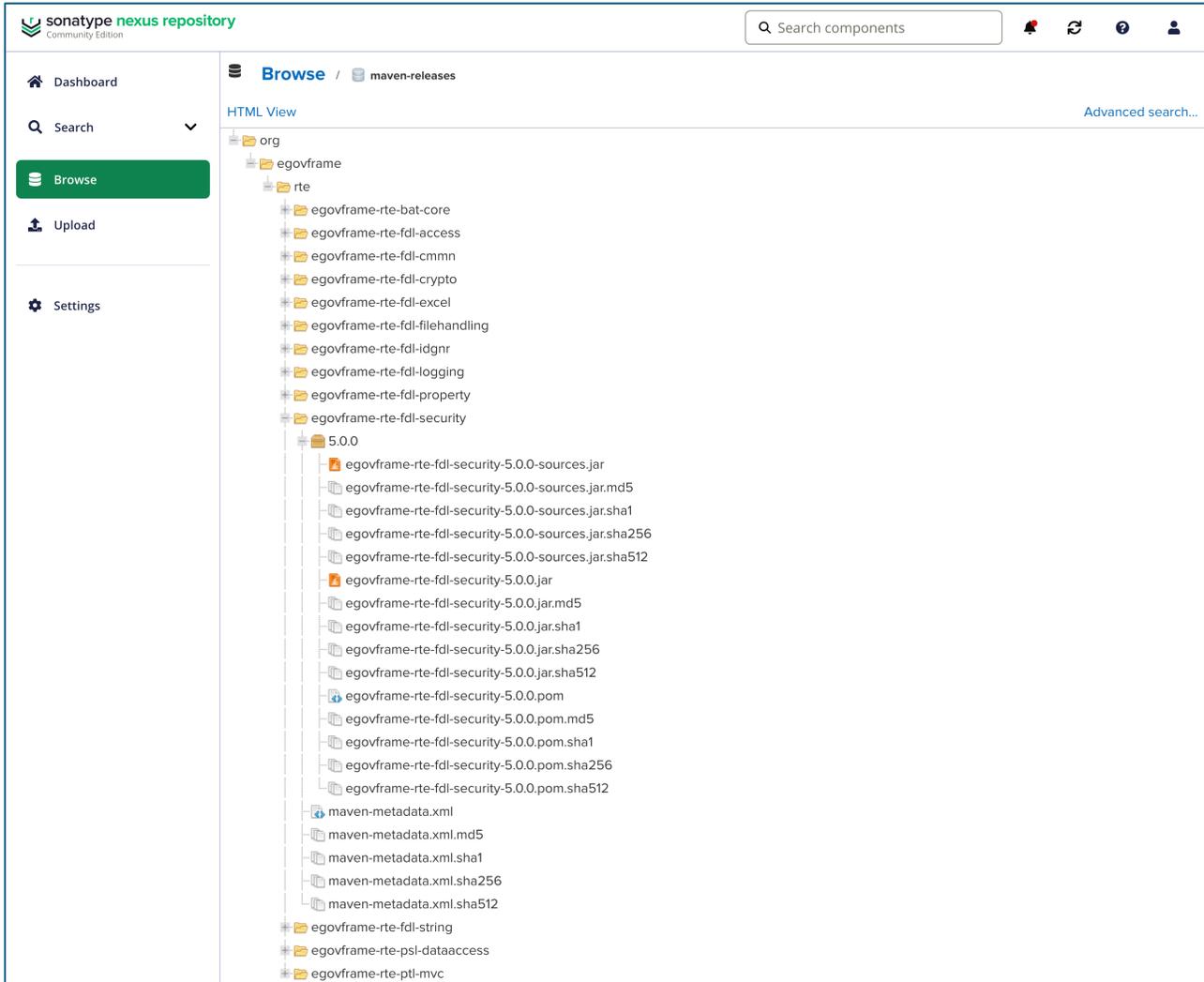
```

egov69@egov69:~/Documents/maven-upload-test/maven-upload-test$ cd /home/egov69/Documents/maven-upload-test/maven-upload-test
egov69@egov69:~/Documents/maven-upload-test/maven-upload-test$ chmod +x nexus_m2_bulk_upload.sh
egov69@egov69:~/Documents/maven-upload-test/maven-upload-test$ ./nexus_m2_bulk_upload.sh \
--root "./repository" \
--nexus "http://127.0.0.1:8081" \
--repo "maven-releases" \
--user "admin" \
--pass "admin123"
[OK ] org.egovframe.rte:egovframe-rte-ptl-mvc:5.0.0 assets=[egovframe-rte-ptl-mvc-5.0.0.pom, egovframe-rte-ptl-mvc-5.0.0.jar, egovframe-rte-ptl-mvc-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-psl-dataaccess:5.0.0 assets=[egovframe-rte-psl-dataaccess-5.0.0.pom, egovframe-rte-psl-dataaccess-5.0.0.jar, egovframe-rte-psl-dataaccess-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-property:5.0.0 assets=[egovframe-rte-fdl-property-5.0.0.pom, egovframe-rte-fdl-property-5.0.0.jar, egovframe-rte-fdl-property-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-logging:5.0.0 assets=[egovframe-rte-fdl-logging-5.0.0.pom, egovframe-rte-fdl-logging-5.0.0.jar, egovframe-rte-fdl-logging-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-bat-core:5.0.0 assets=[egovframe-rte-bat-core-5.0.0.pom, egovframe-rte-bat-core-5.0.0.jar, egovframe-rte-bat-core-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-string:5.0.0 assets=[egovframe-rte-fdl-string-5.0.0.pom, egovframe-rte-fdl-string-5.0.0.jar, egovframe-rte-fdl-string-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-filehandling:5.0.0 assets=[egovframe-rte-fdl-filehandling-5.0.0.pom, egovframe-rte-fdl-filehandling-5.0.0.jar, egovframe-rte-fdl-filehandling-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-idgnr:5.0.0 assets=[egovframe-rte-fdl-idgnr-5.0.0.pom, egovframe-rte-fdl-idgnr-5.0.0.jar, egovframe-rte-fdl-idgnr-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-access:5.0.0 assets=[egovframe-rte-fdl-access-5.0.0.pom, egovframe-rte-fdl-access-5.0.0.jar, egovframe-rte-fdl-access-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-crypto:5.0.0 assets=[egovframe-rte-fdl-crypto-5.0.0.pom, egovframe-rte-fdl-crypto-5.0.0.jar, egovframe-rte-fdl-crypto-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-cmmn:5.0.0 assets=[egovframe-rte-fdl-cmmn-5.0.0.pom, egovframe-rte-fdl-cmmn-5.0.0.jar, egovframe-rte-fdl-cmmn-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-excel:5.0.0 assets=[egovframe-rte-fdl-excel-5.0.0.pom, egovframe-rte-fdl-excel-5.0.0.jar, egovframe-rte-fdl-excel-5.0.0-sources.jar]
[OK ] org.egovframe.rte:egovframe-rte-fdl-security:5.0.0 assets=[egovframe-rte-fdl-security-5.0.0.pom, egovframe-rte-fdl-security-5.0.0.jar, egovframe-rte-fdl-security-5.0.0-sources.jar]

Done. processed=13 ok=13 fail=0
egov69@egov69:~/Documents/maven-upload-test/maven-upload-test$ █

```

스크립트 실행 결과 Nexus 의 maven-releases 레포지토리에 라이브러리가 채워졌습니다.



## 2.4 Nexus Repository URL 을 pom.xml 에 반영

프로젝트의 pom.xml 파일을 열고 아래와 같이 <repositories> 변경하여 반영할 수 있습니다.

```
<repositories>
  <repository>
    <id>egovci</id>
    <url>http://127.0.0.1:8081/repository/maven-releases/</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>true</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>mvn2s</id>
    <url>https://repo1.maven.org/maven2/</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>true</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>egovframe</id>
    <url>https://maven.egovframe.go.kr/maven/</url>
    <releases>
      <enabled>>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

## 2.5 Nexus Repository 2 → 3 마이그레이션(업그레이드)

doc/[표준프레임워크 v5.0]Nexus 마이그레이션가이드.docx 문서를 참조하시기 바랍니다.

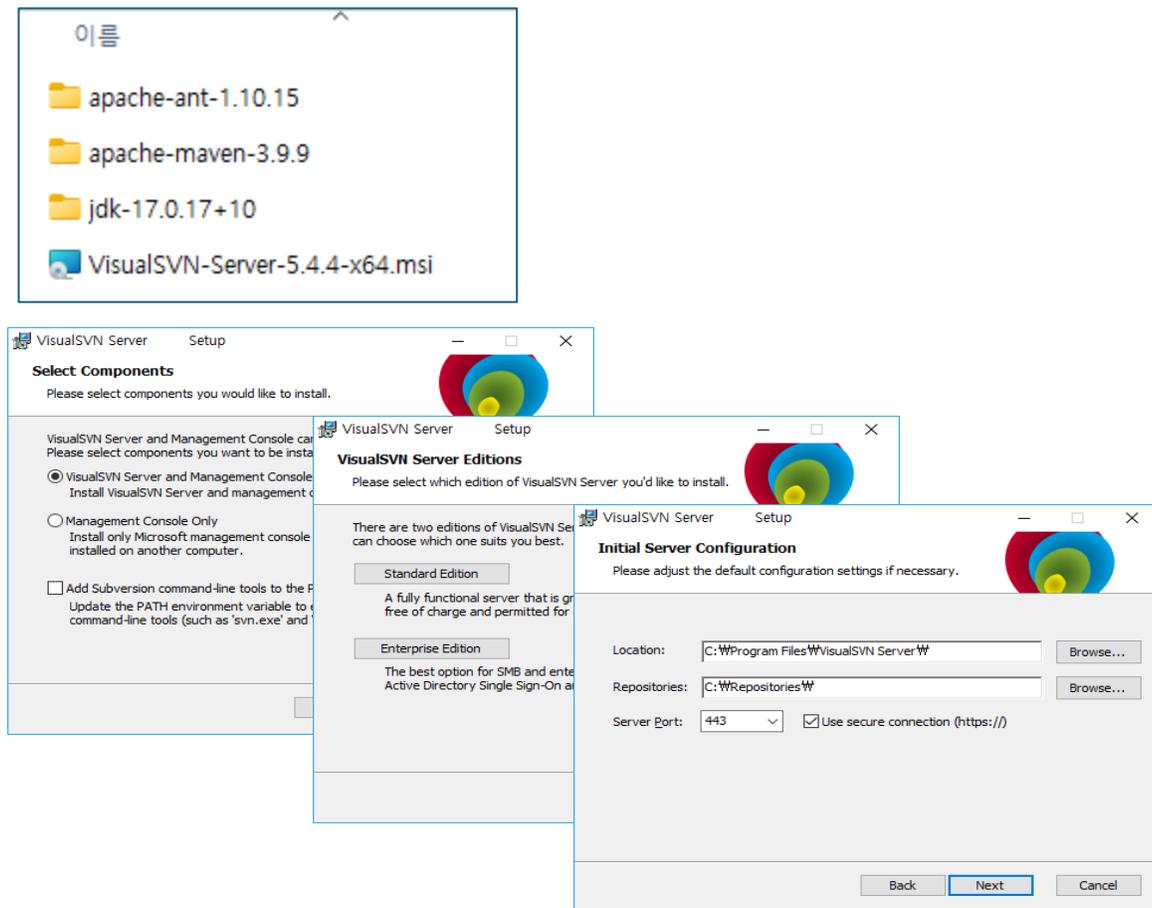
## 3. CI - SVN

### 3.1 SVN 설치 (선택사항)

CI 설정에 앞서 소스 저장소인 SVN(Subversion)을 설치합니다. eGovCI 는 VisualSVN Server 의 설치파일을 별도로 제공하고 있으므로, UI 형태의 SVN 이 필요하다면 이를 설치하여 사용하기 바랍니다.

VisualSVN Server 이외에 별도로 SVN 을 구성하여 사용하셔도 무방합니다. VisualSVN Server 는 Windows 만 지원하므로, Linux 용 eGovCI 패키지에는 VisualSVN Server 가 제공되지 않습니다.

<eGovCI-HOME>\#bin\#VisualSVN-server-XXX.msi 를 설치합니다.



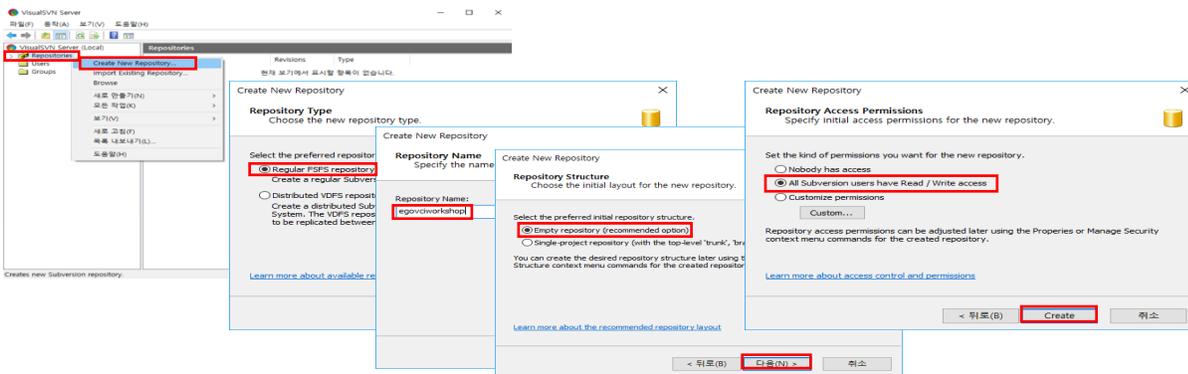
### 3.2 SVN Repositories 설정 및 사용자 등록

SVN 저장소 설정 및 사용자 등록 관련 사항을 설정합니다.

### 3.2.1 저장소 설정 및 등록

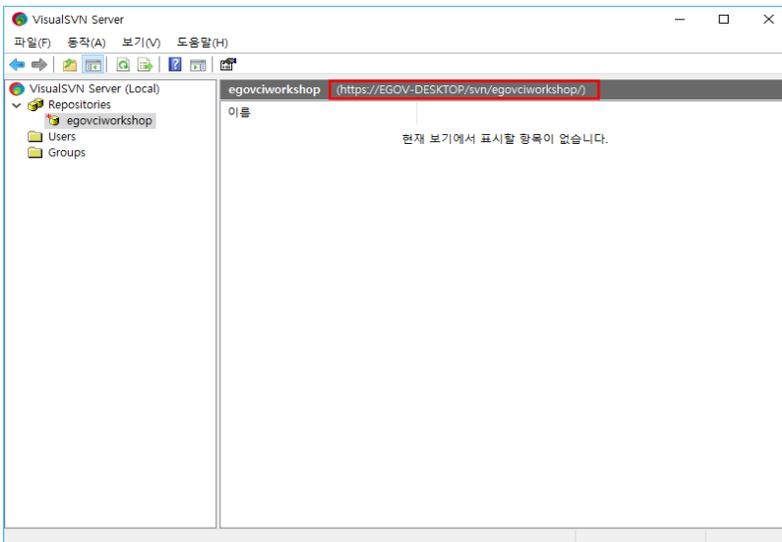
VisualSVN 서버의 구동을 확인 하고, "Repositories"의 오른쪽을 클릭하여 "Create New Repository..." 선택하면, 새로운 저장소를 만들 수 있습니다.

- 임의의 저장소 설정 (ex: egovciworkshop)
- 필요한 사항을 설정하고 Create

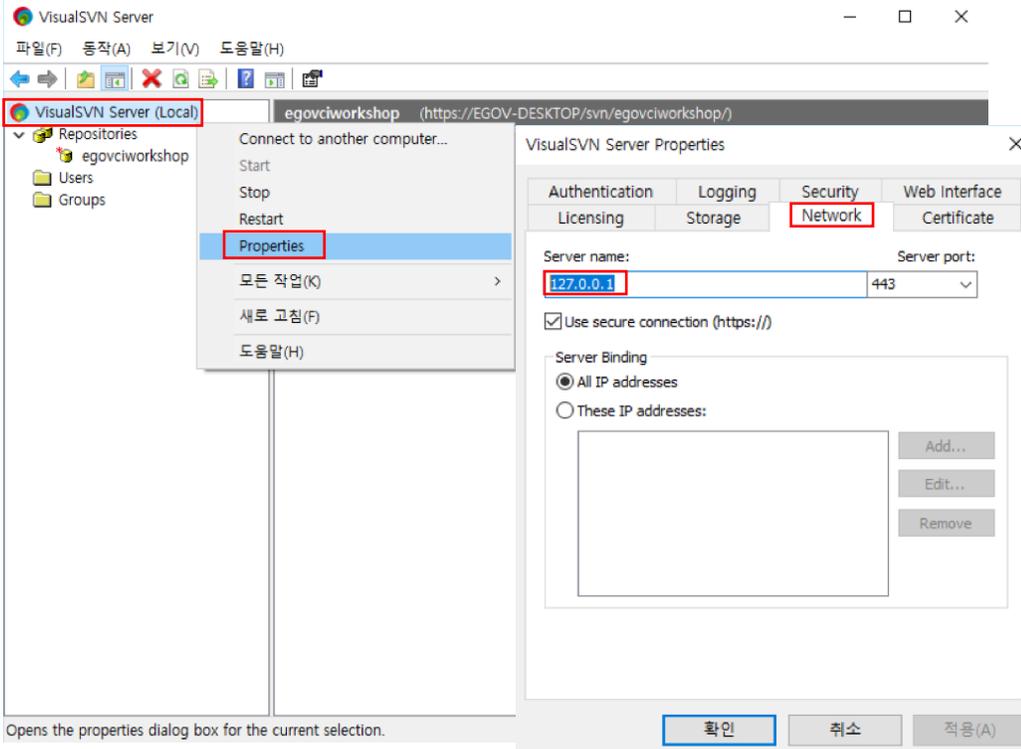


설정을 완료 후, 아래와 같이 저장소 위치를 확인 할 수 있습니다.

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



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



### 3.2.2 사용자 등록

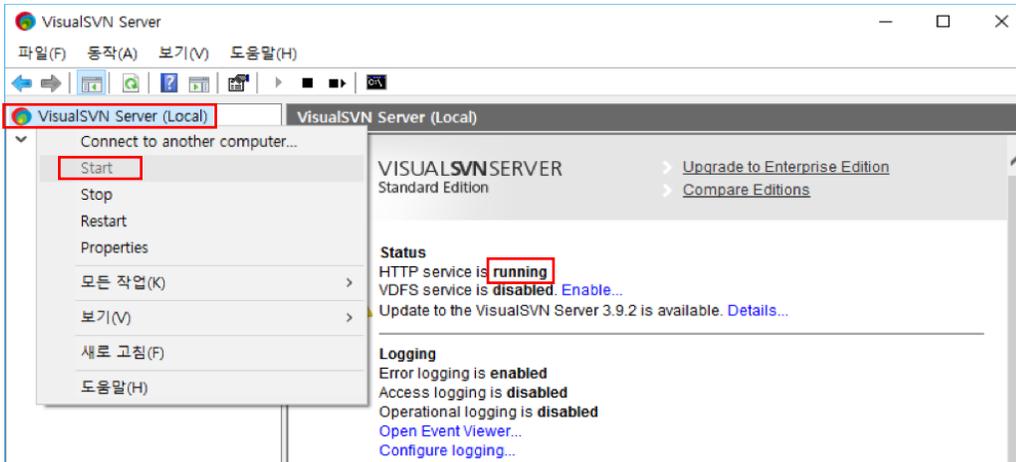
아래와 같이 "User" 메뉴에서 새로운 사용자를 설정 가능합니다.

- 사용자 이름(예시) : "com"
- 비밀번호 : com01



### 3.3 SVN 서버 시작/종료

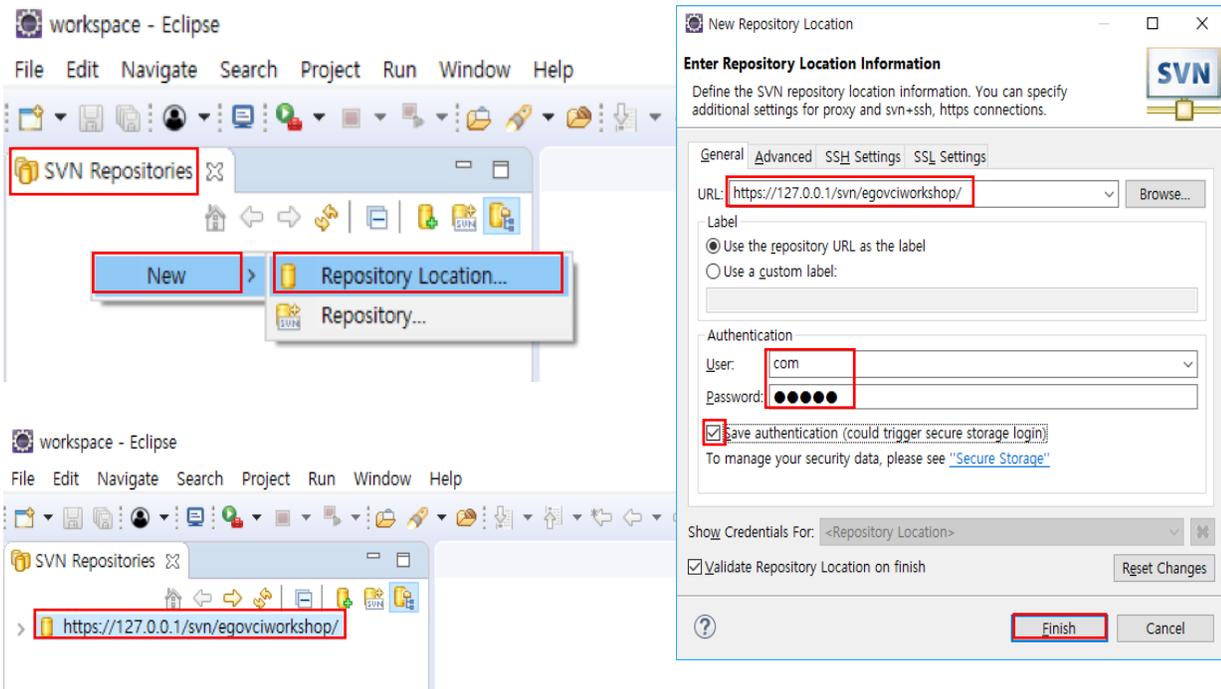
SVN 왼쪽 메뉴에서 아래와 같이 서버를 시작/종료를 할 수 있으며, 서비스 상태를 확인 가능합니다.



### 3.4 Eclipse 프로젝트를 SVN 서버에 등록

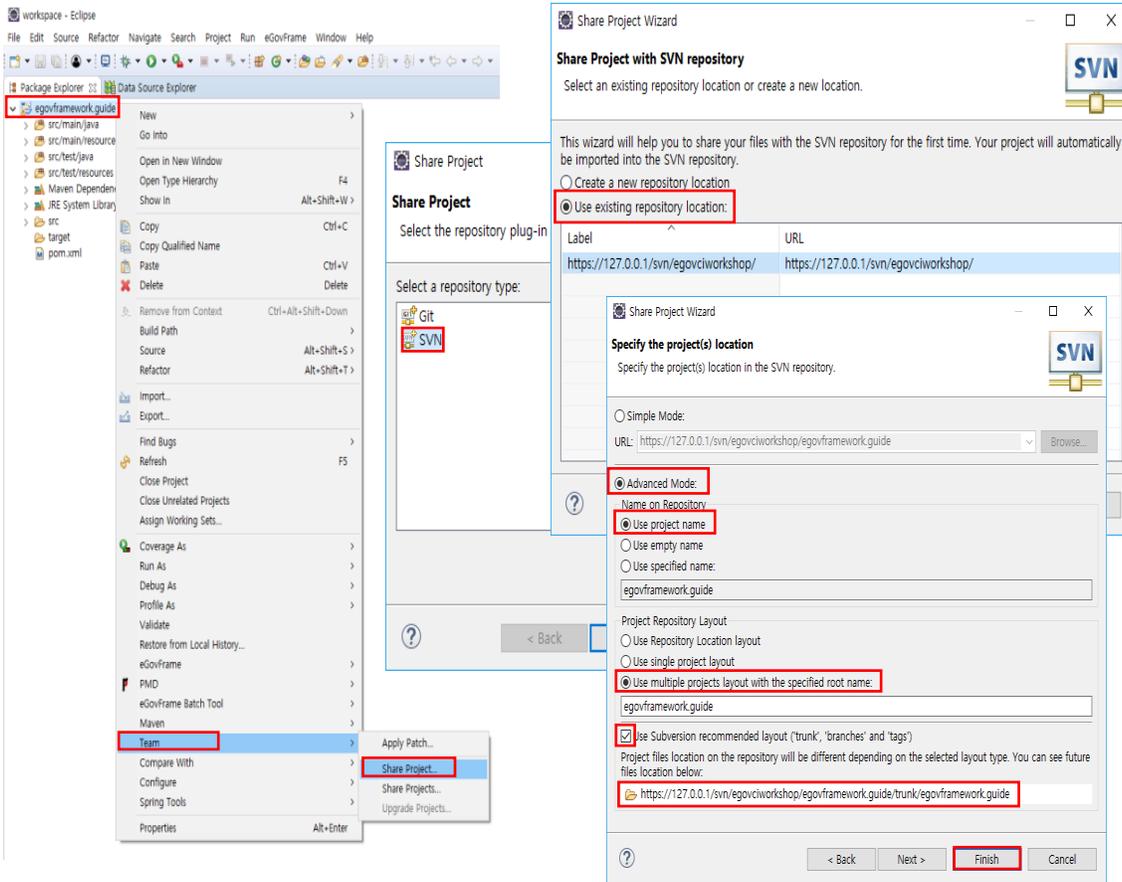
SVN 서버의 정보를 활용하여 Eclipse 내의 프로젝트를 등록하고 소스를 반영합니다.

- Eclipse에서 SVN Repositories를 엽니다.
- 새로운 저장소 정보를 생성하고, SVN정보를 등록합니다.

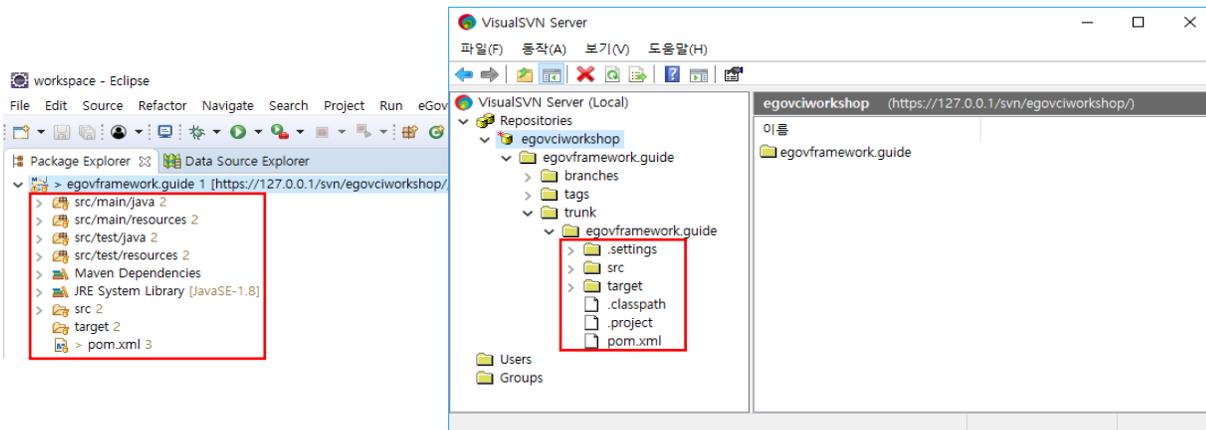


SVN 저장소 정보 등록 후, 프로젝트를 SVN 에 등록합니다..

- 프로젝트 선택 후, 오른쪽 마우스를 클릭하여 "Team -> share Project" 로 프로젝트를 SVN에 공유



등록된 프로젝트는 SVN 서버 Manager 에서 다시 확인 가능합니다.



## 4. CI - Jenkins

### 4.1 Jenkins 서버 구동

CI 시스템 구성 이후, 패키지에 포함된 스크립트 파일로 Jenkins 서버를 구동할 수 있습니다.

#### Linux 운영체제 주의사항

- 구동 전에 JDK 확인이 필요합니다. Linux용 패키지에는 Linux/x64 버전의 JDK가 설치되어 있습니다.
- PC가 x64가 아니라 AArch64 등 다른 버전이라면 해당 버전과 호환되는 JDK를 설치해 주시기 바랍니다.
- 그 후 start-jenkins.sh 스크립트 파일에서 JAVA\_HOME의 값을 위 JDK 설치 경로로 수정해 주시기 바랍니다.

```

Open  start-jenkins.sh
~/eGovCI-5.0.0-Linux-64bit

#!/bin/sh

# ----- CI Jenkins SETTING start -----

# CI_HOME: 스크립트가 위치한 디렉토리 기준
SCRIPT_DIR=$(CDPATH= cd "$(dirname "$0")" && pwd)
export CI_HOME="$SCRIPT_DIR"

export JAVA_HOME="$CI_HOME/bin/jdk-17.0.17+10"
export PATH="$JAVA_HOME/bin:$PATH"
export JAVA_OPTS="-Xmx768m"

export JENKINS_APP_HOME="$CI_HOME/jenkins/jenkins-app"
export JENKINS_HOME="$CI_HOME/jenkins/jenkins-home"

```

#### Jenkins 서버 구동 방법

- Windows
  - 실행 명령어 : start-jenkins.bat (또는 start-nexus.bat 파일 더블클릭)
  - 종료 명령어 : Jenkins가 실행 중인 cmd 종료 (ctrl + c)
- Linux
  - 실행 명령어 : ./start-jenkins.sh
  - 종료 명령어 : Jenkins가 실행 중인 터미널 종료 (ctrl + c)

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

```

egov69@egov69: ~/eGovCI-5.0.0-Linux-64bit$ ./start-jenkins.sh
Information: this Jenkins is a Generic Java Package(jenkins.war) 2.528.3 LTS Version.
-----
Using CI_HOME:           "/home/egov69/eGovCI-5.0.0-Linux-64bit"
Using JAVA_HOME:        "/home/egov69/eGovCI-5.0.0-Linux-64bit/bin/jdk-17.0.17+10"
Using JAVA_OPTS:        "-Xmx768m"
Using JENKINS_APP_HOME: "/home/egov69/eGovCI-5.0.0-Linux-64bit/jenkins/jenkins-app"
Using JENKINS_HOME:     "/home/egov69/eGovCI-5.0.0-Linux-64bit/jenkins/jenkins-home"
-----
eGovCI Jenkins server start
Enter http://127.0.0.1:8082/ to access Jenkins(default port).
Running from: /home/egov69/eGovCI-5.0.0-Linux-64bit/jenkins/jenkins-app/jenkins.war
webroot: /home/egov69/eGovCI-5.0.0-Linux-64bit/jenkins/jenkins-home/war
2026-02-02 07:27:40.790+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
2026-02-02 07:27:40.844+0000 [id=1] WARNING o.e.j.ee9.nested.ContextHandler#setContextPath: Empty contextPath
2026-02-02 07:27:40.905+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-12.0.25; built: 2025-08-11T2
2026-02-02 07:27:41.286+0000 [id=1] INFO o.e.j.e.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, c
2026-02-02 07:27:41.345+0000 [id=1] INFO o.e.j.s.DefaultSessionIdManager#doStart: Session workerName=node0
2026-02-02 07:27:41.658+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /home/egov69
perty("JENKINS_HOME")
2026-02-02 07:27:41.745+0000 [id=1] INFO o.e.j.s.handler.ContextHandler#doStart: Started oeje9n.ContextHandler$CoreE
nux-64bit/jenkins/jenkins-home/war/,a=AVAILABLE,h=oeje9n.ContextHandler$CoreContextHandler$CoreToNestedHandler@1051817b{STA
2026-02-02 07:27:41.759+0000 [id=1] INFO org.e.j.server.AbstractConnector#doStart: Started ServerConnector@2970a5bc{HT
2026-02-02 07:27:41.768+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started oejs.Server@d706f19{STARTI
2026-02-02 07:27:41.769+0000 [id=26] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=d
2026-02-02 07:27:41.901+0000 [id=25] INFO jenkins.model.Jenkins#<init>: Starting version 2.528.3
2026-02-02 07:27:42.005+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2026-02-02 07:27:42.141+0000 [id=37] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2026-02-02 07:27:45.120+0000 [id=41] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2026-02-02 07:27:45.135+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2026-02-02 07:27:45.146+0000 [id=42] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2026-02-02 07:27:45.476+0000 [id=34] INFO h.p.b.g.GlobalTimeoutConfiguration#load: global timeout not set
2026-02-02 07:27:46.262+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2026-02-02 07:27:46.263+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2026-02-02 07:27:46.527+0000 [id=42] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2026-02-02 07:27:46.546+0000 [id=37] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2026-02-02 07:27:46.589+0000 [id=38] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2026-02-02 07:27:46.629+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running

```

\*\* 서버 성능에 따라 구동 시간이 다소 소요될 수 있습니다.

구동 후, 아래와 같은 주소에서 CI 를 위한 Jenkins 웹 페이지를 확인 할 수 있습니다.

- <http://localhost:8082/>



## Sign in to Jenkins

사용자 이름

비밀번호

로그인 상태 유지

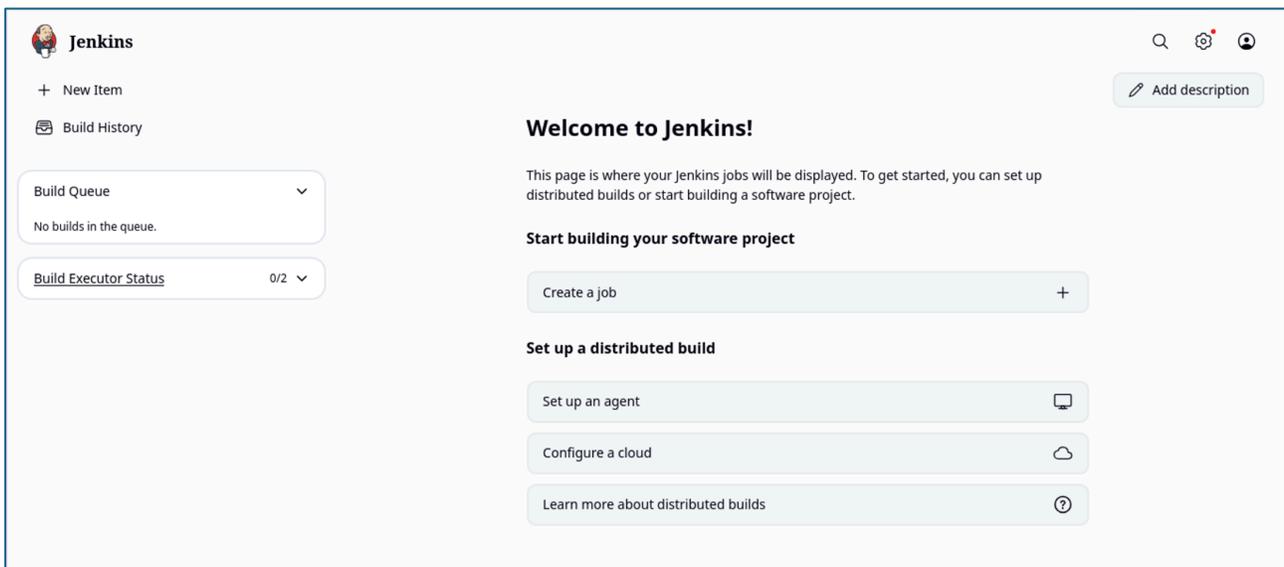
로그인

## 관리자 계정

- ID : admin
- Password : admin123

\* 기본 제공되는 비밀번호는 반드시 다른 것으로 변경하여 사용하기 바랍니다.

\* 기본 제공되는 비밀번호는 반드시 다른 것으로 변경하여 사용하기 바랍니다.



## 4.2 Jenkins 플러그인 목록

eGovCI 패키지 내 Jenkins 에 설치되어 있는 플러그인 목록입니다.

1. Organization and Administration
  - A. Folders
  - B. OWASP Markup Formatter
2. Build Features
  - A. Build Timeout
  - B. Credentials Binding
  - C. Timestamper
  - D. Workspace Cleanup
3. Build Tools

- A. Ant
- B. Gradle
- 4. Build Analysis and Reporting
  - A. Coverage
  - B. HTML Publisher
  - C. JUnit
- 5. Pipelines and Continuous Delivery
  - A. Pipeline
  - B. GitHub Branch Source
  - C. Pipeline: Github Groovy Libraries
  - D. Pipeline: Graph View
- 6. Source Code Management
  - A. Git
  - B. Git Parameter
  - C. GitHub
  - D. GitLab
  - E. Subversion
- 7. Distributed Builds
  - A. SSH Build Agents
- 8. User Management and Security
  - A. Matrix Authorization Strategy
  - B. LDAP
- 9. Notifications and Publishing
  - A. Email Extension
  - B. Mailer
- 10. Appearance
  - A. Dark Theme
- 11. Etc
  - A. Maven Integration
  - B. Maven Invoker

## 4.3 Jenkins 설정

### 4.3.1 Jenkins 설정 : Port

start-jenkins.bat 또는 start-jenkins.sh 파일에서 마지막 줄에 `--httpPort` 의 값을 수정하면 Jenkins 서버의 포트를 변경할 수 있습니다.

```

Open  start-jenkins.sh
~/eGovCI-5.0.0-Linux-64bit

port)."
echo "-----"

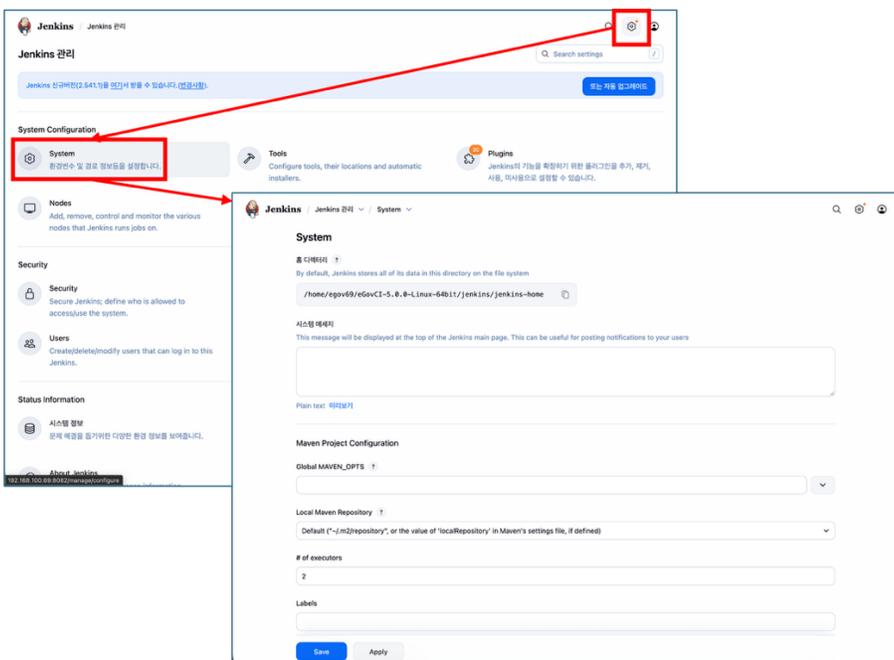
# NOTE: -D(JVM system properties) must come before -jar to take
effect.
"$JAVA_HOME/bin/java" $JAVA_OPTS -DJENKINS_HOME="$JENKINS_HOME"
-jar "$JENKINS_APP_HOME/jenkins.war" --httpPort=8082

echo "done."
    
```

### 4.3.2 Jenkins 설정 : System

Jenkins 의 새 작업 등록 전, Jenkins 관리에 System 에서 기본 시스템 설정합니다. JDK, ANT, Maven, SVN 서버 정보를 등록하여 프로젝트를 컴파일 및 Deploy 할 수 있습니다.

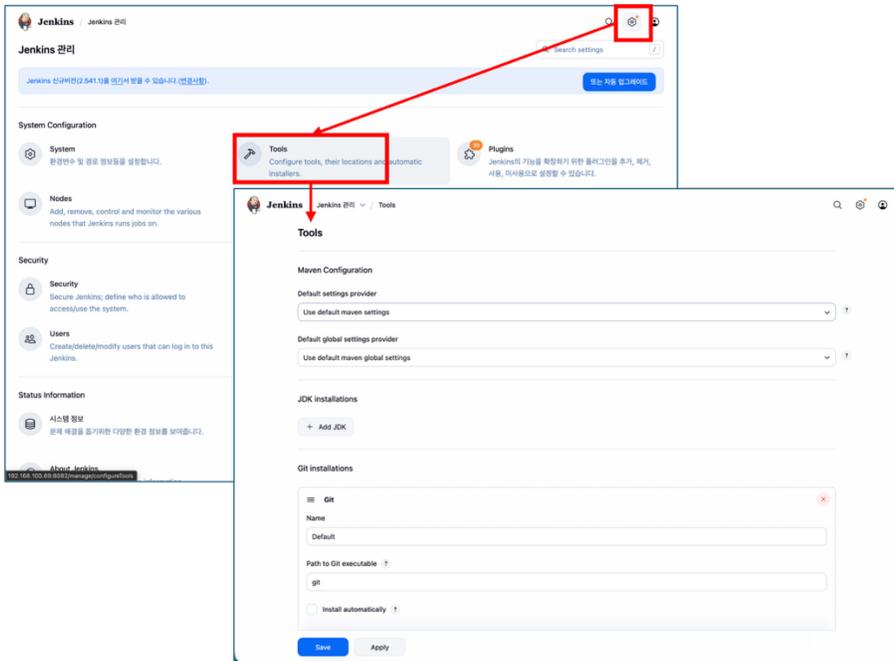
시스템 설정은 Jenkins 관리 → System Configure → System 메뉴를 통해서 확인할 수 있습니다.



### 4.3.3 Jenkins 설정 : Tools

Jenkins 관리에 Tools 에서 JDK, ANT, MAVEN 등을 설정합니다.

Tools 설정은 Jenkins 관리 → System Configure → Tools 메뉴를 통해서 확인할 수 있습니다.



JDK 는 eGovCI 에서 제공하는 JDK 17 로 설정되어 있으며, 빌드할 프로젝트에 맞춰 JDK 를 추가/삭제 하여도 무방합니다. eGovCI 에서 기본으로 설정한 JAVA\_HOME 경로는 다음과 같습니다.

- Windows – JAVA\_HOME : C:\eGovCI-5.0.0-Windows-64bit\bin\jdk-17.0.17+10
- Linux – JAVA\_HOME : /home/egov69/eGovCI-5.0.0-Linux-64bit\bin\jdk-17.0.17+10

JDK installations

JDK installations ^ Edited

+ Add JDK

≡ **JDK** ×

Name

JDK 17

JAVA\_HOME

/home/egov69/eGovCI-5.0.0-Linux-64bit/bin/jdk-17.0.17+10

Install automatically ?

Ant 는 eGovCI 에서 제공하는 Ant 1.10.15 로 설정되어 있으며, 빌드할 프로젝트에 맞춰 Ant 를 추가/삭제 하여도 무방합니다. eGovCI 에서 기본으로 설정한 ANT\_HOME 경로는 다음과 같습니다.

- Windows – ANT\_HOME : C:\WeGovCI-5.0.0-Windows-64bit/bin/ant-1.10.15
- Linux – ANT\_HOME : /home/egov69/eGovCI-5.0.0-Linux-64bit/bin/ant-1.10.15

Ant installations

Ant installations ^ Edited

+ Add Ant

≡ **Ant** ×

Name

Ant 1.10.15

ANT\_HOME

/home/egov69/eGovCI-5.0.0-Linux-64bit/bin/apache-ant-1.10.15

Install automatically ?

Maven 은 eGovCI 에서 제공하는 Maven 3.9.9 로 설정되어 있으며, 빌드할 프로젝트에 맞춰 Maven 을 추가/삭제 하여도 무방합니다. eGovCI 에서 기본으로 설정한 MAVEN\_HOME 경로는 다음과 같습니다.

- Windows – ANT\_HOME : C:\WeGovCI-5.0.0-Windows-64bit/bin/maven-3.9.9
- Linux – ANT\_HOME : /home/egov69/eGovCI-5.0.0-Linux-64bit/bin/maven-3.9.9

Maven installations

Maven installations ^ Edited

+ Add Maven

≡ Maven ×

Name

Maven 3.9.9

MAVEN\_HOME

/home/egov69/egovCI-5.0.0-Linux-64bit/bin/apache-maven-3.9.9

Install automatically ?

인터넷 외부망과의 연결이 허용된 환경이라면, Install automatically 버튼을 클릭하여 설치할 수도 있습니다.

추가적으로 Maven 의 사용자 정의 settings.xml 파일도 지정할 수 있습니다.

## Tools

### Maven Configuration

#### Default settings provider

Settings file in filesystem

File path ?

#### Default global settings provider

Use default maven global settings

## 4.4 Jenkins 작업

### 4.4.1 작업 생성

통합 환경에서 자동으로 프로젝트를 빌드하기 위해 아래와 같이 새 작업을 생성합니다.

**Jenkins**

+ 새로운 Item

빌드 기록

빌드 대기 목록

빌드 대기 항목이 없습니다.

빌드 실행 상태 0/2

### Jenkins에 오신 것을 환영합니다.

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

#### Start building your software project

Create a job +

#### Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

Item(Job)의 이름을 입력하고, Maven project 를 선택합니다.

**Jenkins** / New Item

## New Item

Enter an item name

sample build

Select an item type

- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Maven project**  
Maven 프로젝트를 빌드합니다. Jenkins은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
다양한 환경에서의 테스트, 플랫폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Source Code Management(소스 코드 관리)에서 SVN URL 과 사용자 정보를 입력합니다. 사용자 정보 Credentials 를 입력하기 위해 Add 버튼을 클릭합니다.

The screenshot shows the Jenkins configuration page for 'sample build' under 'Configuration'. The 'Source Code Management' section is active. Under 'Subversion', the 'Repository URL' field is filled with 'https://192.168.100.153/svn/egovci/sample/trunk/test.bbs'. The 'Credentials' dropdown menu is set to '- none -', and the '+ Add' button is highlighted with a red box.

Credentials 추가 화면에서 자신이 사용할 SVN 계정의 Username, Password 를 입력하고 추가합니다.

The screenshot shows the 'Add Credentials' dialog. The 'Domain' is set to 'Global credentials (unrestricted)'. The 'Kind' is 'Username with password'. The 'Scope' is 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'com' and the 'Password' field contains '.....'. Both fields are highlighted with red boxes. At the bottom, there are 'Cancel' and 'Add' buttons.

빌드 유발(Triggers)와 빌드 환경(Environment)을 설정합니다.

Build 항목에서 최상위 레벨의 pom.xml 파일 경로와 빌드 목표를 입력합니다. Root POM 의 기본 값은 pom.xml 이고, Goals 는 "clean install", "clean deploy" 등 원하는 빌드 지점을 입력합니다.

The screenshot shows a configuration interface for a build process. It is divided into two main sections: 'Pre Steps' and 'Build'. Under 'Pre Steps', there is a button labeled 'Add pre-build step'. The 'Build' section contains two input fields. The first is labeled 'Root POM' and contains the text 'pom.xml'. The second is labeled 'Goals and options' and contains the text 'clean install'. Both input fields are highlighted with a red border.

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

The screenshot shows the 'Advanced' configuration page for Maven. It features a list of checkboxes for various options, including 'Incremental build', 'Disable automatic artifact archiving', and 'Enable triggering of downstream projects'. The 'File path' field under the 'Settings file' section is highlighted with a red border. Other fields include 'MAVEN\_OPTS', 'Maven Validation Level', and 'Global Settings file'.

이후 저장을 누르면 작업 생성이 완료됩니다.

빌드 후 조치

빌드 후 조치 추가 ▾

저장

Apply

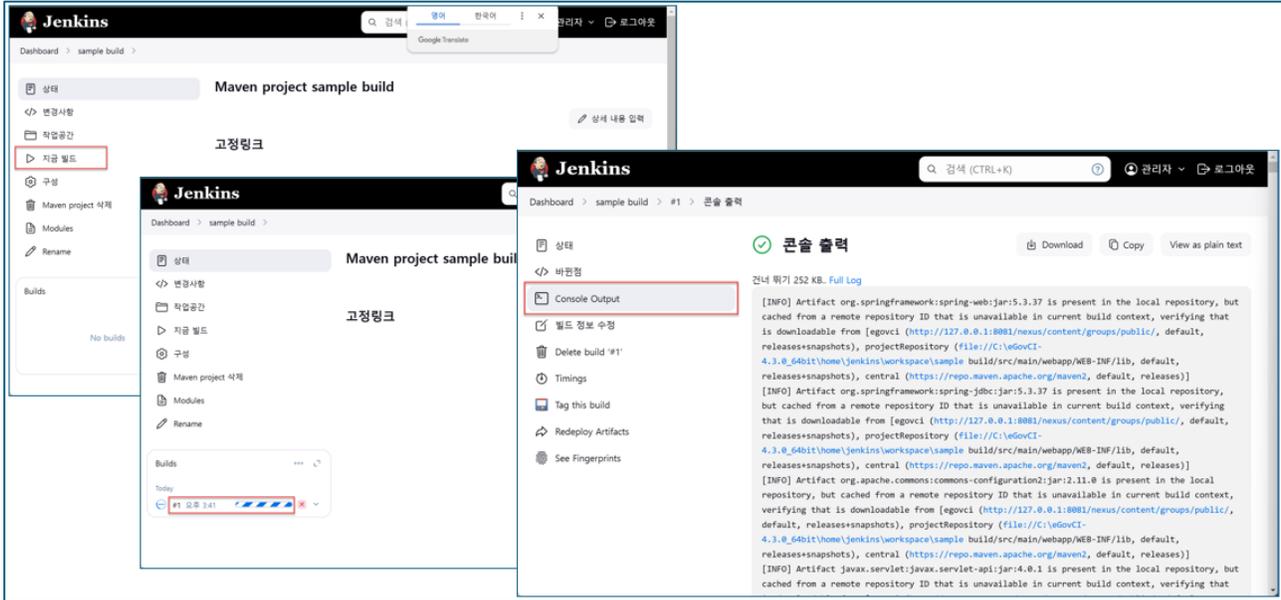
The screenshot shows the Jenkins dashboard with a search bar at the top right containing '검색 (CTRL+K)'. The main content area displays a table of build records. The table has columns for 'S', 'W', 'Name', '최근 성공', '최근 실패', and '최근 소요 시간'. A single row is visible for 'sample build' with a success status (green checkmark), a duration of '1 hr 13 min', and a build number of '#1'. Below the table, there are filters for '아이콘: S M L' and a '모든 항목' button. On the left sidebar, there are menu items like '새로운 Item', '빌드 기록', '프로젝트 연관 관계', '파일 핑거프린트 확인', 'Jenkins 관리', and 'My Views'. At the bottom right, it says 'REST API Jenkins 2.479.2'.

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

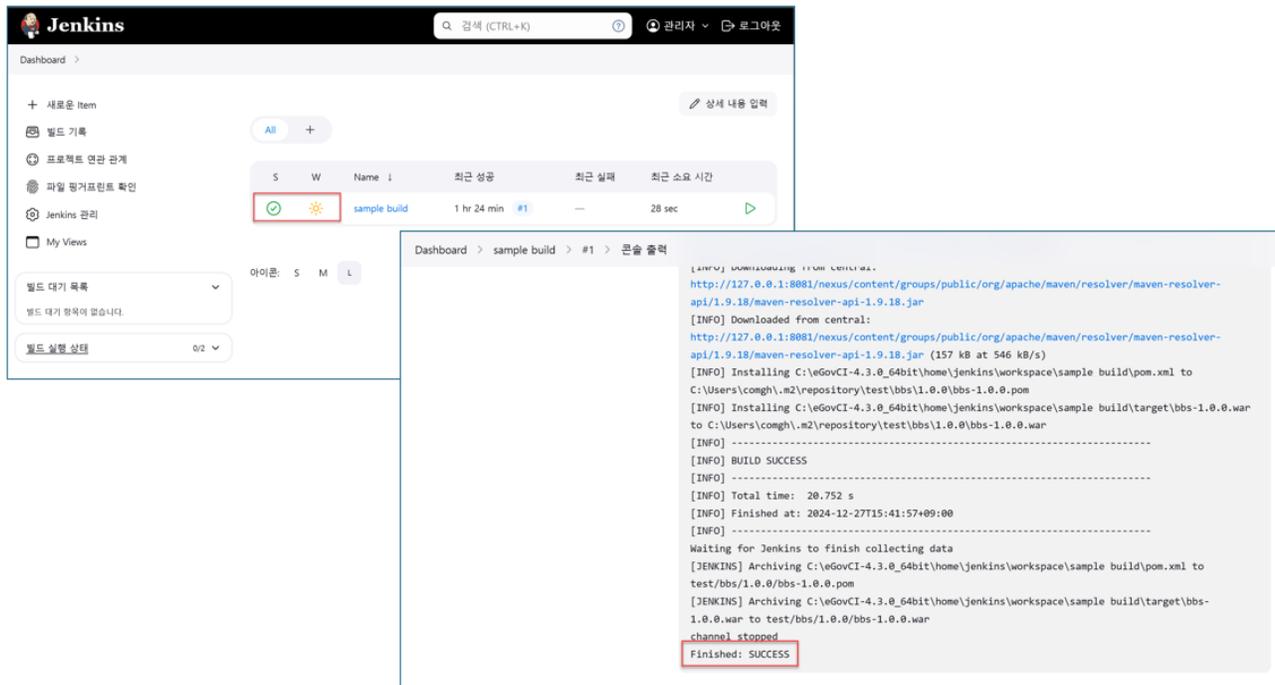
## 4.4.2 작업 Build

모든 설정 이후, 최초 수동으로 Build 하여 설정을 확인 할 수 있습니다.

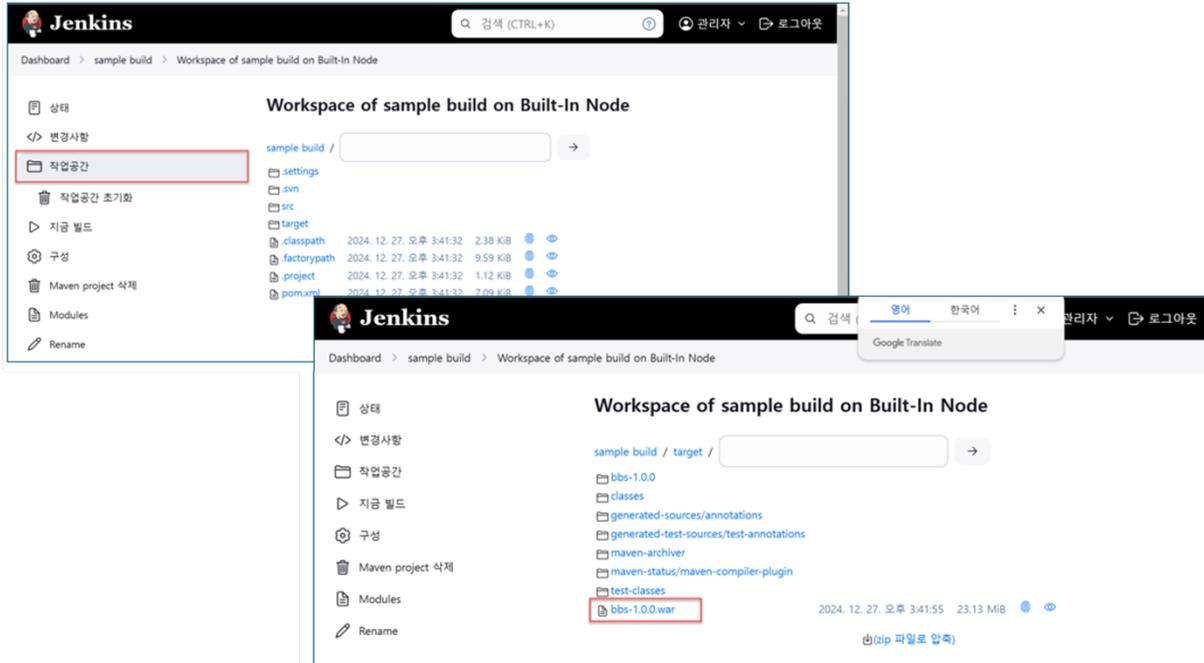
아래와 같이 Build Now"메뉴를 통하여, Maven build 를 실행합니다. 빌드 실행 시 콘솔을 통하여 빌드 프로세스를 확인할 수 있습니다.



아래와 같이 작업 아이콘 또는 콘솔의 정보로 빌드를 확인할 수 있습니다.



정상적으로 빌드가 끝나면, Workspace 에서 빌드를 통하여 만들어진 artifact 를 확인할 수 있습니다.



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