

표준프레임워크  
CI 및 Nexus 환경 구축 가이드  
Version 3.10

---

## Index

1. 표준프레임워크 EGOVCI 패키지 설치 .....	3
1.1 개요 .....	3
1.2 EGOVCI 압축풀기 .....	3
1.3 EGOVCI 시스템 구성 .....	3
1.4 CI 시스템 구동 (START/STOP) .....	4
2. NEXUS 설정 정보 .....	6
2.1 NEXUS 서버구동(START/STOP) .....	6
2.2 NEXUS 저장소 설정 .....	7
2.3 NEXUS 저장소에 라이브러리 추가 .....	7
2.4 NEXUS 실행 및 추가된 라이브러리 확인 .....	8
2.5 NEXUS 저장소를 프로젝트에 반영 .....	8
2.6 NEXUS 저장소에 3 <sup>RD</sup> PARTY 라이브러리 추가 .....	9
3. CI (SVN & JENKINS) 설정 .....	11
3.1 SUBVERSION 설치 (선택사항) .....	11
3.2 SVN REPOSITORIES 설정 및 사용자 등록 .....	12
3.2.1 저장소 설정 및 등록 .....	12
3.2.2 사용자 등록 .....	13
3.2.3 SVN 서버 시작/종료 .....	14
3.2.4 Eclipse 프로젝트를 SVN 서버에 등록 .....	14
3.3 JENKINS 설정 .....	16
3.3.1 Jenkins 기본 시스템 설정 .....	17
3.3.2 새 작업 생성 .....	20
3.3.3 작업 Bulid .....	24

# 1. 표준프레임워크 eGovCI 패키지 설치

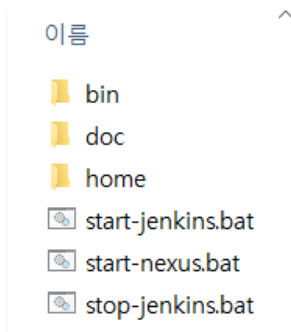
## 1.1 개요

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>:8080/jenkins](http://<ip_address>:8080/jenkins)
- NEXUS : [http://<ip\\_address>:8081/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.

- Ant-1.10.9 (under /bin dictory)
- Maven-3.6.3 (under /bin dictory)
- Tomcat-8.5.61 (under /bin dictory)
- OpenJDK.1.8 (under /bin dictory)
- VisualSVN-Server-4.3.1.msi (under /bin dictory) - 별도 설치형 파일 제공 (리눅스 제외)

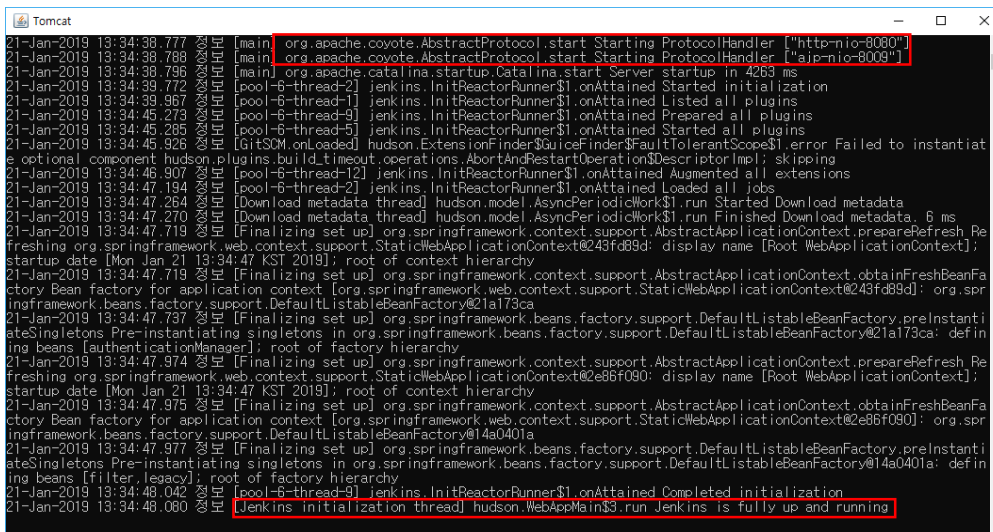
## 1.4 CI 시스템 구동 (Start/Stop)

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

(! 윈도우의 경우 반드시 관리자 권한으로 실행)



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



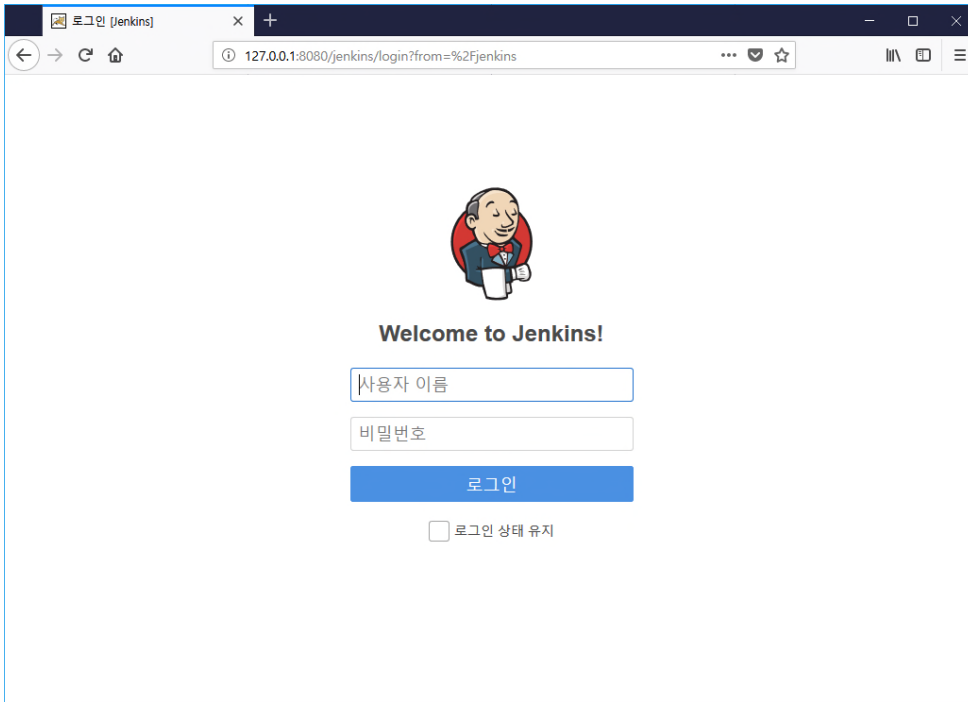
\*\* 서버 성능에 따라 구동 시간이 다소 소요될 수 있다.

구동후, 아래와 같은 주소를 CI 웹 페이지를 확인 할 수 있다.

(<http://localhost:8080/jenkins/>)

계정 ID : admin ,

계정 비밀번호 : <eGovCI\_경로>/home/jenkins/secrets/initialAdminPassword 파일 확인



\*\* CI 서버 기본 포트는 '8080'으로 설정되어 있으며, 아래와 같이 별도 포트로 변경하여 사용가능 하다. (<eGovCI\_home>\bin\apache-tomcat-jenkins\conf\server.xml)

```

server - 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 (blocking & non-blocking)
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="8080" 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" />
-->
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation that requires the JSSE
-->

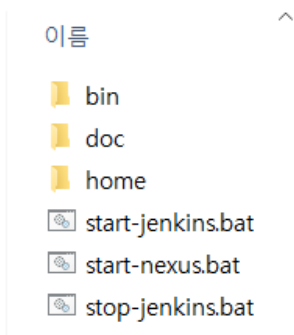
server - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
    
```

## 2. Nexus 설정 정보

### 2.1 Nexus 서버구동(Start/Stop)

CI 시스템 구성 이후, “start-nexus.bat”을 사용하여 Nexus 서버 구동 할 수 있으며, 실행된 콘솔창을 닫으면 Nexus 서버를 멈출 수 있다. (별도의 stop nexus 파일 없음)

(! 반드시 관리자 권한으로 실행)



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

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

```
C:\WINDOWS\System32\cmd.exe
Using CI_HOME: "D:\eGovCI-3.9.0_64bit#"
Using JAVA_HOME: "D:\eGovCI-3.9.0_64bit\bin\jdk8u242-b08"
Using NEXUS_HOME: "D:\eGovCI-3.9.0_64bit\bin\nexus-2.14.15-01-bundle\nexus-2.14.15-01"
-----
CI_HOME : "D:\eGovCI-3.9.0_64bit#"
-----
eGovCI Nexus server start
wrapper | Starting the nexus service...
wrapper | Waiting to start...
```

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

파일 경로 : <eGovCI\_home>\bin\nexus-2.14.20-02-bundle\nexus-2.14.20-02\conf

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

## 2.2 Nexus 저장소 설정

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

```

start-nexus - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
@echo off
rem ----- CI Nexus SETTING start -----
set CI_HOME=%~d0#%~p0
set JAVA_HOME=%CI_HOME%\bin\jdk1.8.0_162
set JRE_HOME=%CI_HOME%\bin\jdk1.8.0_162\jre
set CATALINA_HOME=%CI_HOME%\bin\apache-tomcat-nexus
set PLEXUS_NEXUS_WORK=%CI_HOME%\home\nexus
echo Using CI_HOME: "%CI_HOME%"
echo Using JAVA_HOME: "%JAVA_HOME%"
echo Using CATALINA_HOME: "%CATALINA_HOME%"
rem ----- CI Nexus SETTING end -----
echo -----
echo CI_HOME : "%CI_HOME%"
echo -----
echo eGovCI Nexus server start
echo -----
set JAVA_OPTS=-Xmx768m
%CATALINA_HOME%\bin\catalina.bat start
echo done.
    
```

로컬 디스크 (C:) > eGovCI > home > nexus

이름 ^

- conf
- db
- felix-cache
- health-check
- indexer
- logs
- nuget
- plugin-repository
- storage
- timeline
- trash
- nexus.lock

## 2.3 Nexus 저장소에 라이브러리 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.

로컬 디스크 (C:) > eGovCI > home > nexus > storage > releases

이름 ^

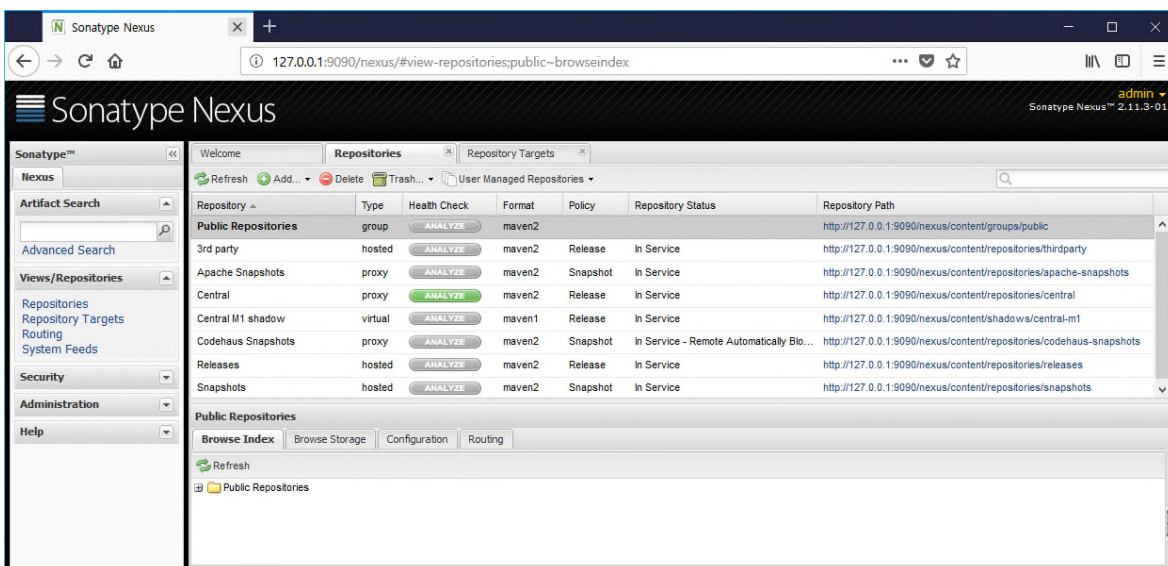
- .meta
- archetype-catalog

## 2.4 Nexus 실행 및 추가된 라이브러리 확인.

"start-nexus.bat" 실행하여, Nexus 를 실행가능하며, 브라우저에서 아래 주소로 Nexus 을 확인 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 저장소를 프로젝트에 반영.

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



```

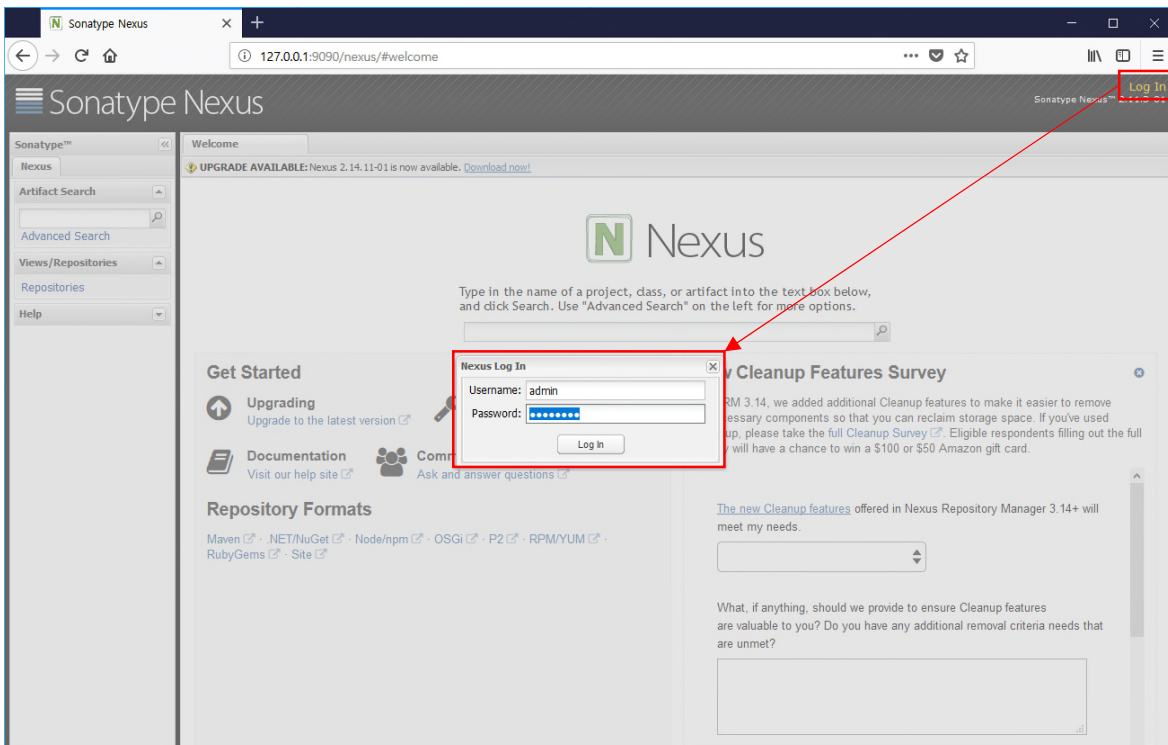
<repositories>
  <repository>
    <id>public</id>
    <url>http://127.0.0.1:8081/nexus/content/groups/public</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
  <repository>
    <id>egovframe</id>
    <url>http://maven.egovframe.kr:8080/maven/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>

```

## 2.6 Nexus 저장소에 3<sup>rd</sup> Party 라이브러리 추가.

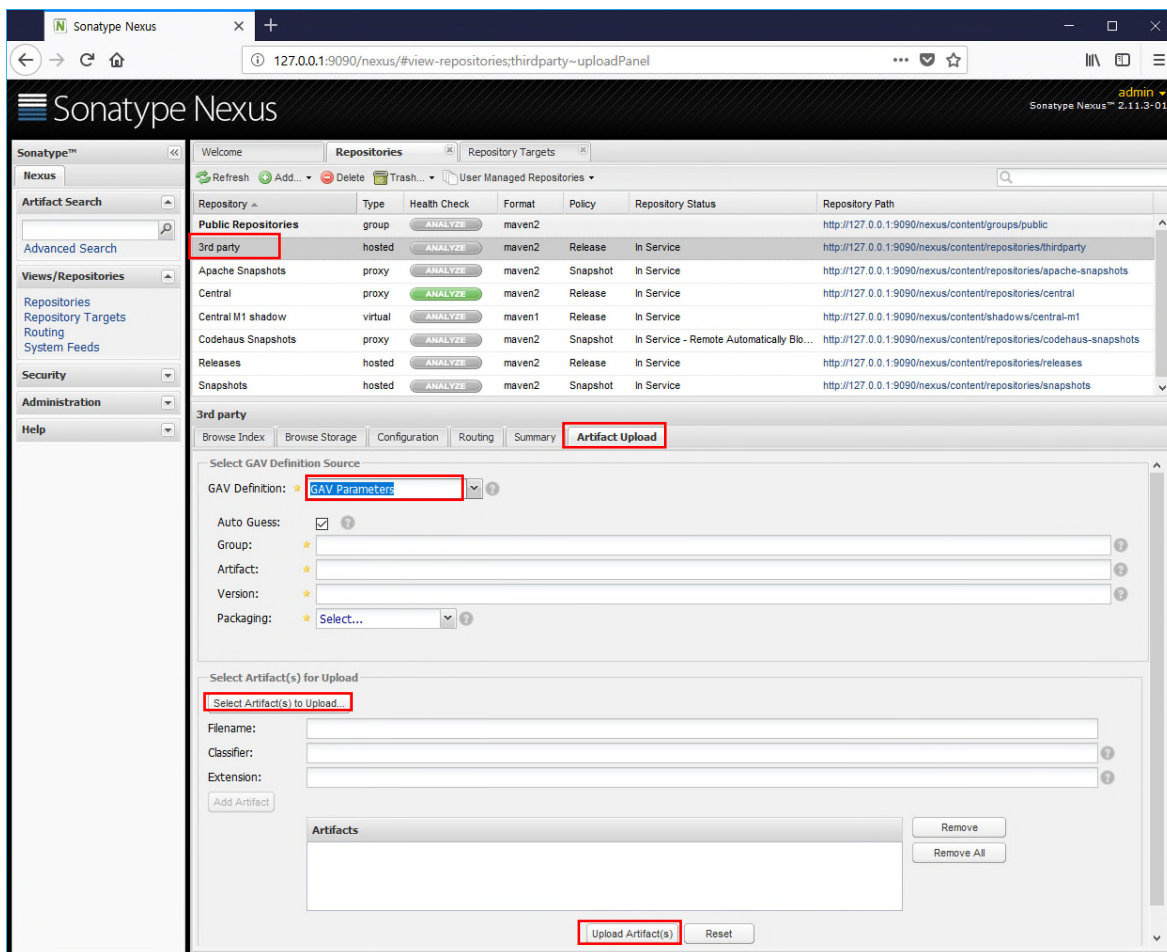
Nexus 저장소에 별도로 3rd Party 라이브러리를 별개로 등록 가능하다.

### 1. 관리자로 로그인 : admin / admin123



### 2. 저장소 목록에서 “3<sup>rd</sup> 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.









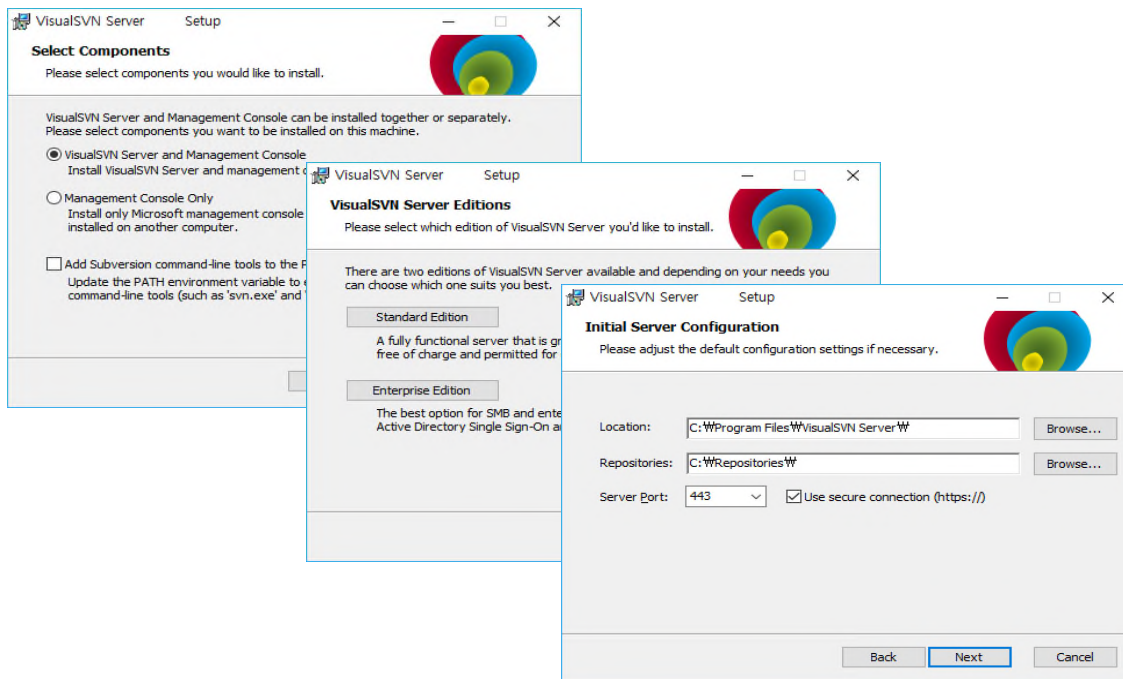
### 3. CI (SVN & JENKINS) 설정

#### 3.1 Subversion 설치 (선택사항)

CI 설정에 앞서 소스 저장소(Subversion)을 설치 한다. eGovCI 에는 VisualSVN 의 설치파일을 별도로 제공하고 있으며, 필요 시 설치하여 사용한다. (별도로 SVN 을 구성하여 사용하여도 무방하다.)

eGovCI 의 VisualSVN-server-XXX.msi 를 설치한다.

-  apache-ant-1.10.9
-  apache-maven-3.6.3
-  apache-tomcat-8.5.61-jenkins
-  jdk8u242-b08
-  nexus-2.14.20-02-bundle
-  **VisualSVN-Server-4.3.1-x64.msi**



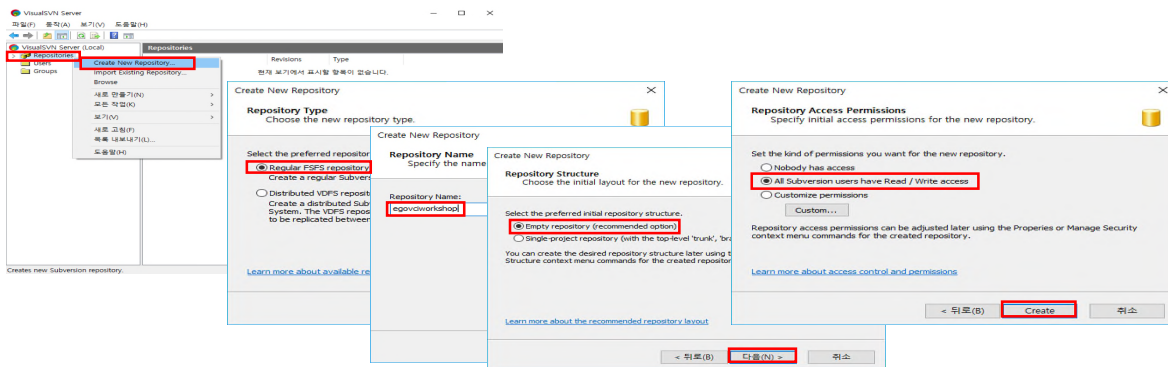
## 3.2 SVN Repositories 설정 및 사용자 등록.

SVN 저장소 설정 및 사용자 등록 관련 사항을 설정한다.

### 3.2.1 저장소 설정 및 등록

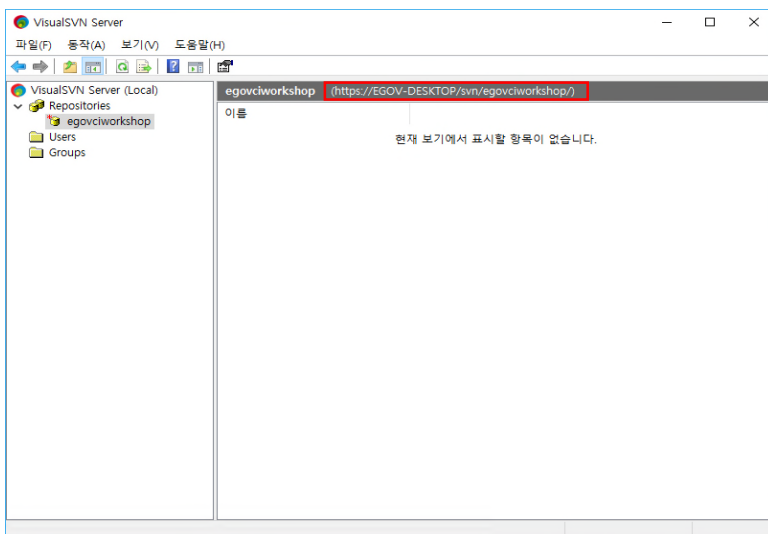
VisualSVN 서버의 구동을 확인 하고, "Repositories"의 오른쪽을 클릭하여 “Create New Repository...” 선택, 새로운 저장소를 만든다.

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

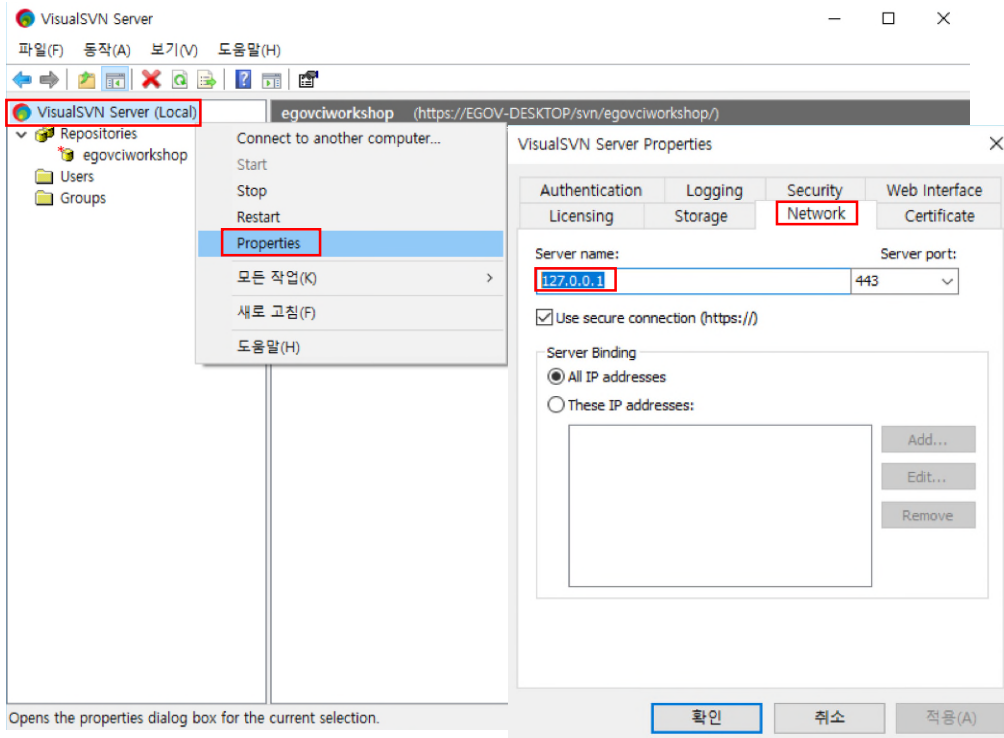


설정을 완료 후, 아래와 같이 저장소 위치를 확인 할 수 있다.

Repository location : “http://<localhost>/svn/egovciworkshop/”



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

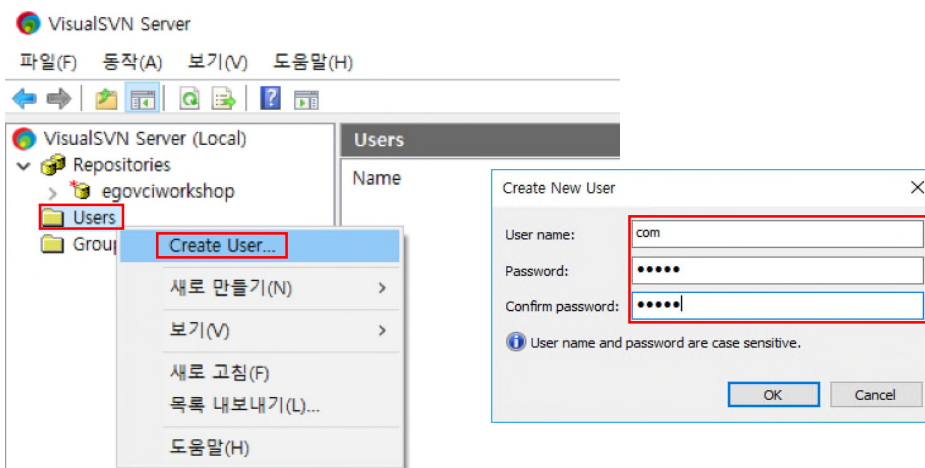


### 3.2.2 사용자 등록

아래와 같이 "User" 메뉴에서 새로운 사용자를 설정 가능하다.

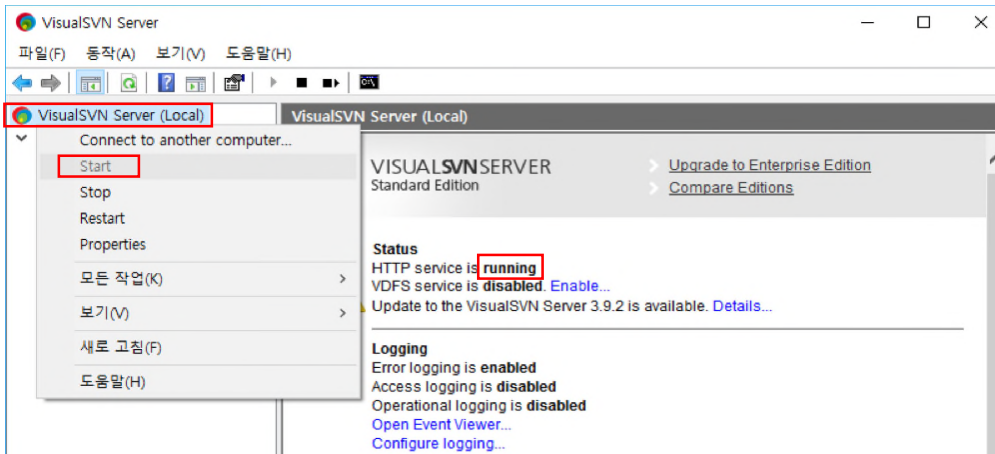
Ex) 사용자 이름: "com"

비밀번호 : "com01"



### 3.2.3 SVN 서버 시작/종료.

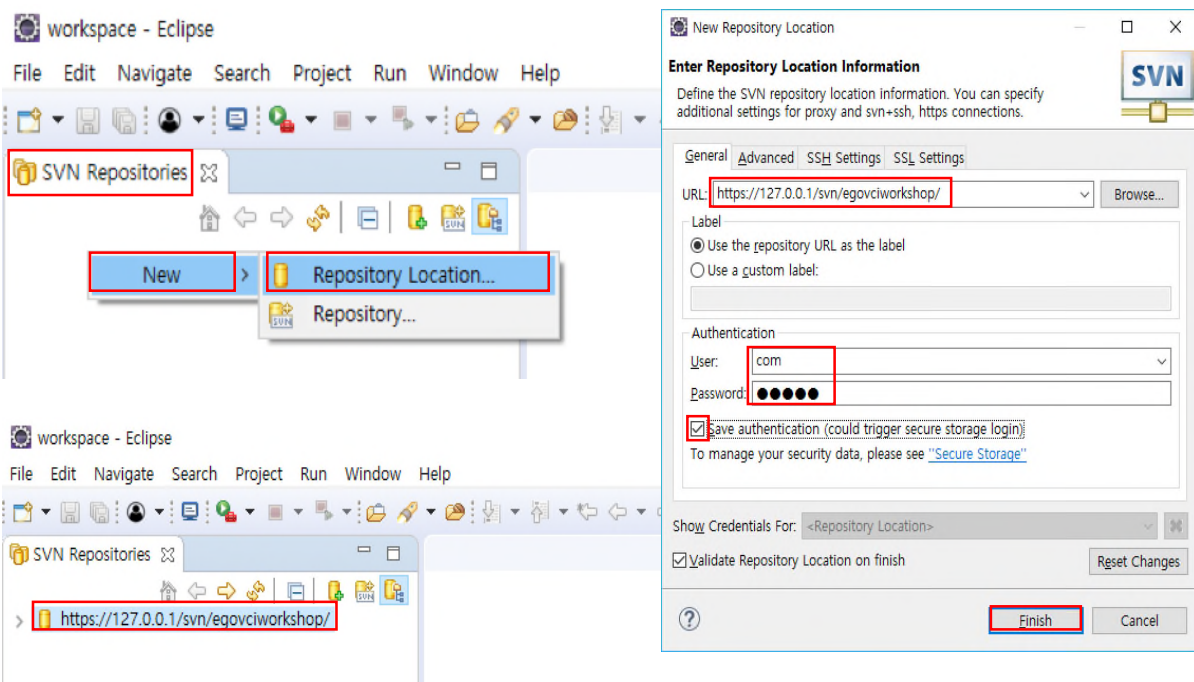
SVN 왼쪽 메뉴에서 아래와 같이 서버를 시작/종료를 할 수 있으며, 확인이 가능하다.



### 3.2.4 Eclipse 프로젝트를 SVN 서버에 등록.

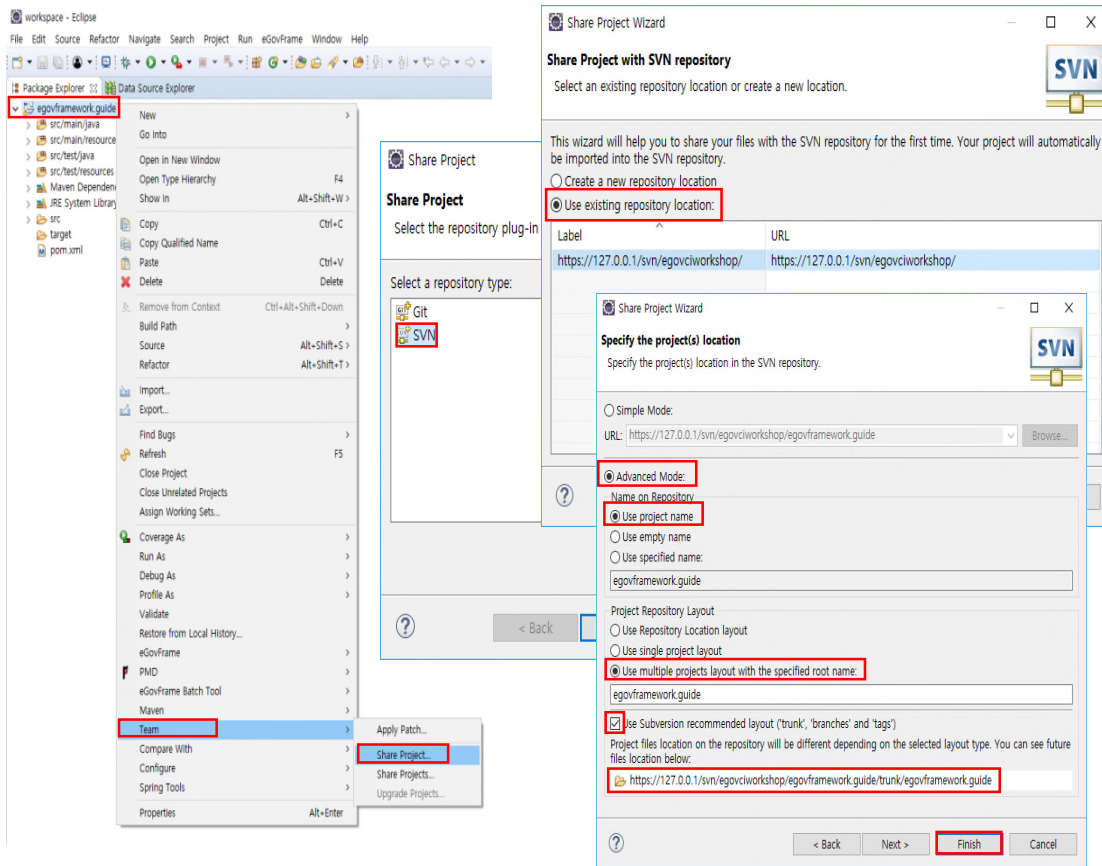
SVN 서버의 정보를 활용하여 Eclipse 내의 프로젝트를 등록하고 소스를 반영한다.

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

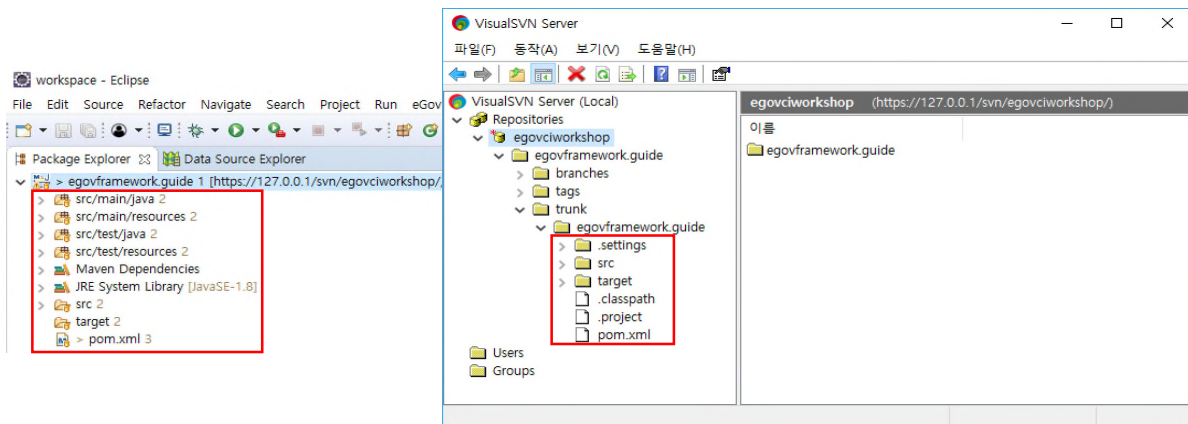


SVN 저장소 정보 등록 후, 프로젝트를 SVN 에 등록 하여야 한다.

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



등록된 프로젝트는 SVN 서버 Manager 에서 다시 확인 가능하다.

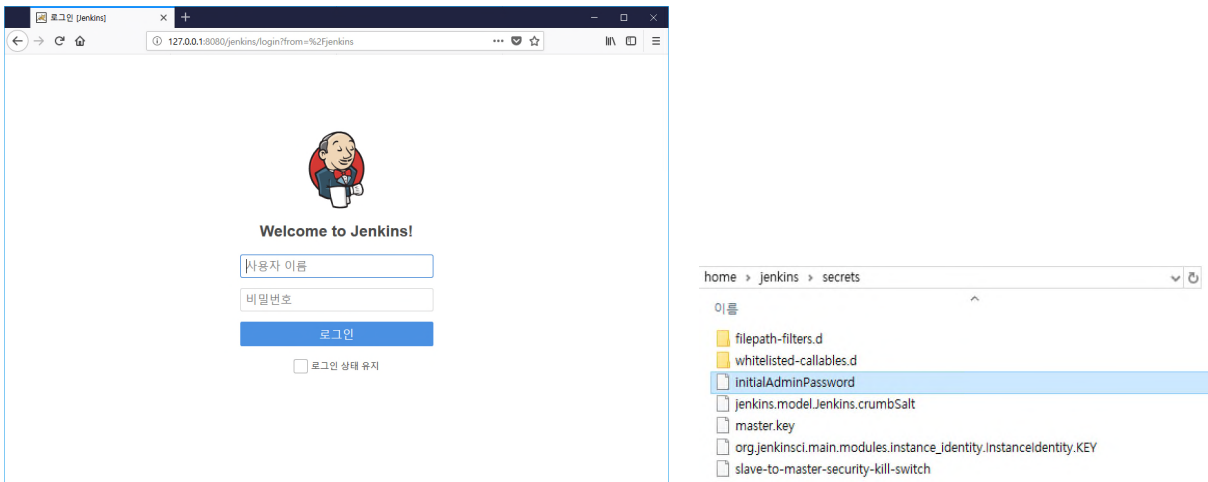


### 3.3 Jenkins 설정.

다음은 Jenkins 설정으로 아래와 같은 주소를 통해 접속이 가능하다.

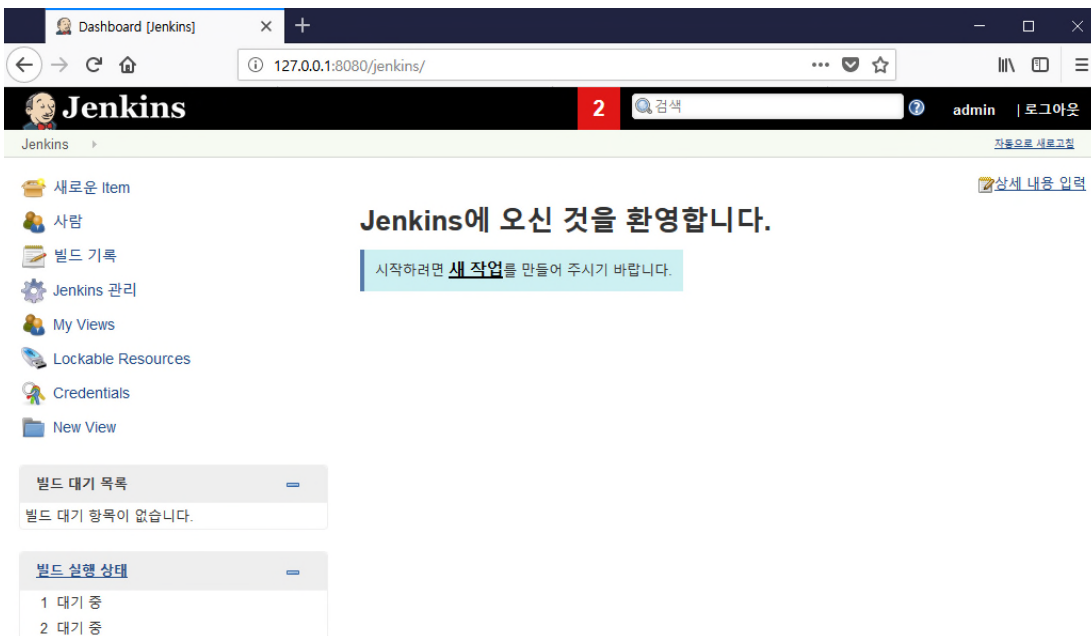
<http://<localhost>:8080/jenkins/>

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



Ex) 사용자 이름: "admin"

비밀번호 : <initialAdminPassword> 파일 확인

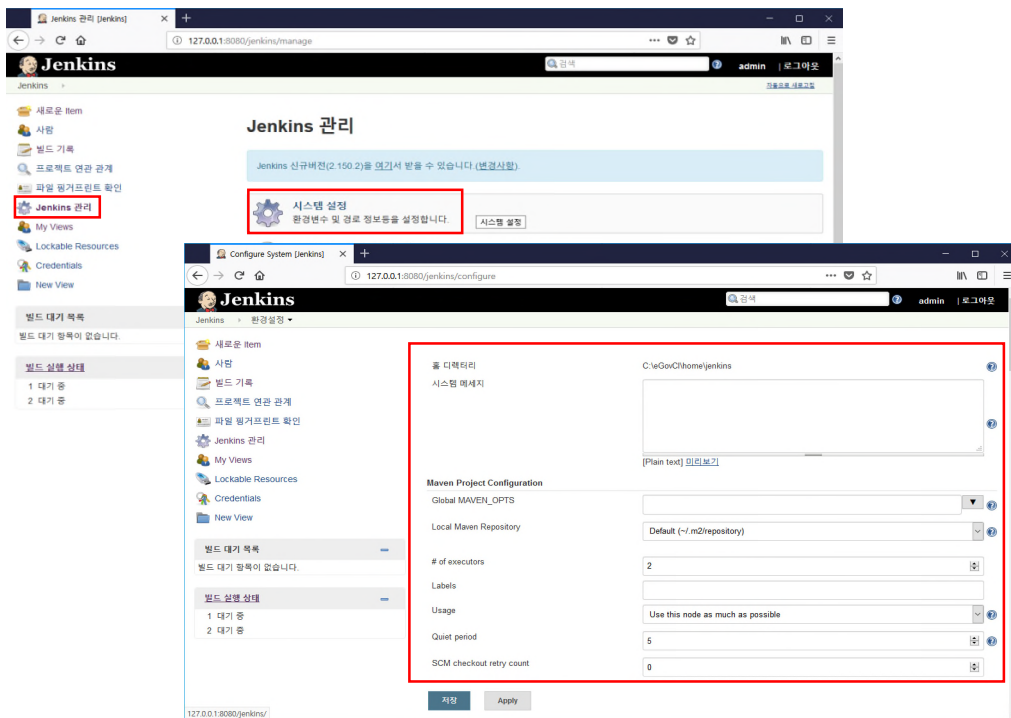




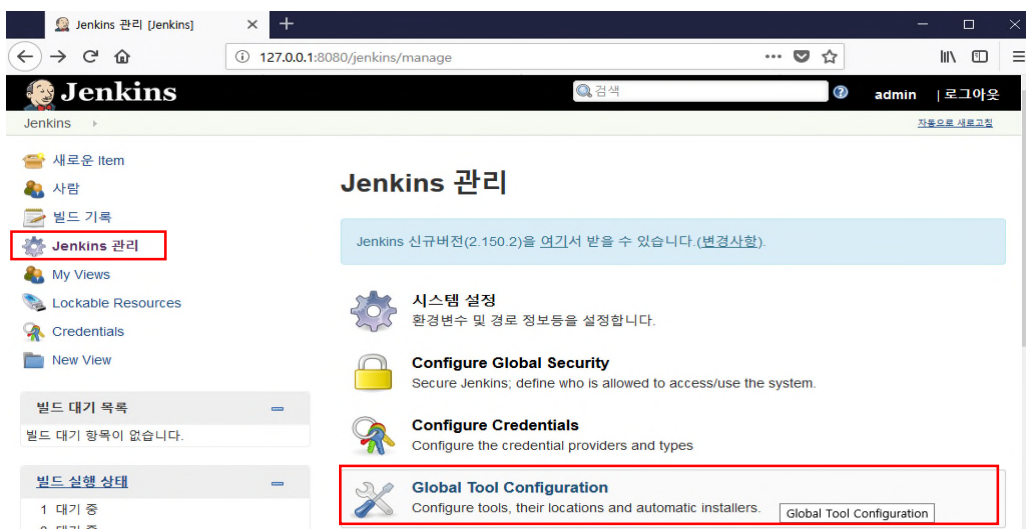
### 3.3.1 Jenkins 기본 시스템 설정

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 -> Configure System 메뉴를 통해서 할 수 있다..



아래와 같이 JDK, ANT, MAVEN home 디렉토리를 설정한다.



- apache-ant-1.10.9
- apache-maven-3.6.3
- apache-tomcat-8.5.61-jenkins
- jdk&u242-b08
- nexus-2.14.20-02-bundle
- VisualSVN-Server-4.3.1-x64.msi

**JDK**

JDK installations

Add JDK

JDK Name: jdk1.8.0\_162

JAVA\_HOME: C:\eGovCli\bin\jdk1.8.0\_162

Install automatically

Delete JDK

---

**Ant**

Ant installations

Add Ant

Ant Name: apache-ant-1.9.5

ANT\_HOME: C:\eGovCli\bin\apache-ant-1.9.5

Install automatically

Delete Ant

---

**Maven**

Maven installations

Add Maven

Maven Name: apache-maven-3.3.3

MAVEN\_HOME: C:\eGovCli\bin\apache-maven-3.3.3

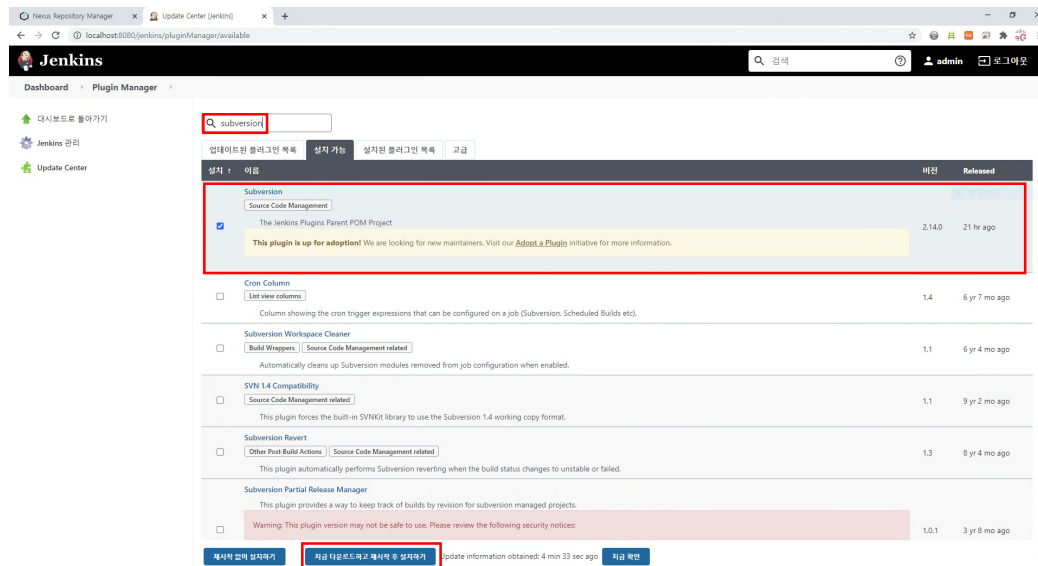
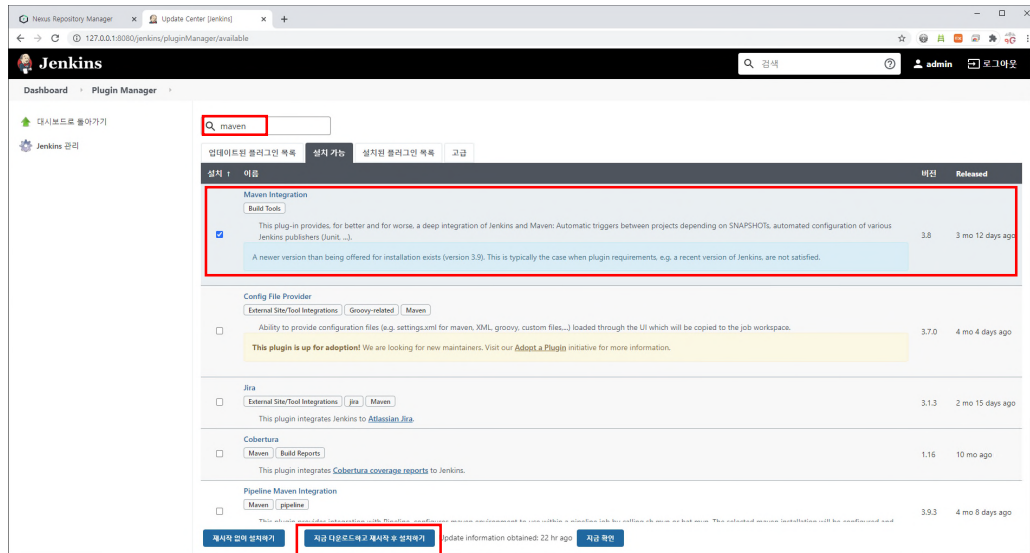
Install automatically

Delete Maven

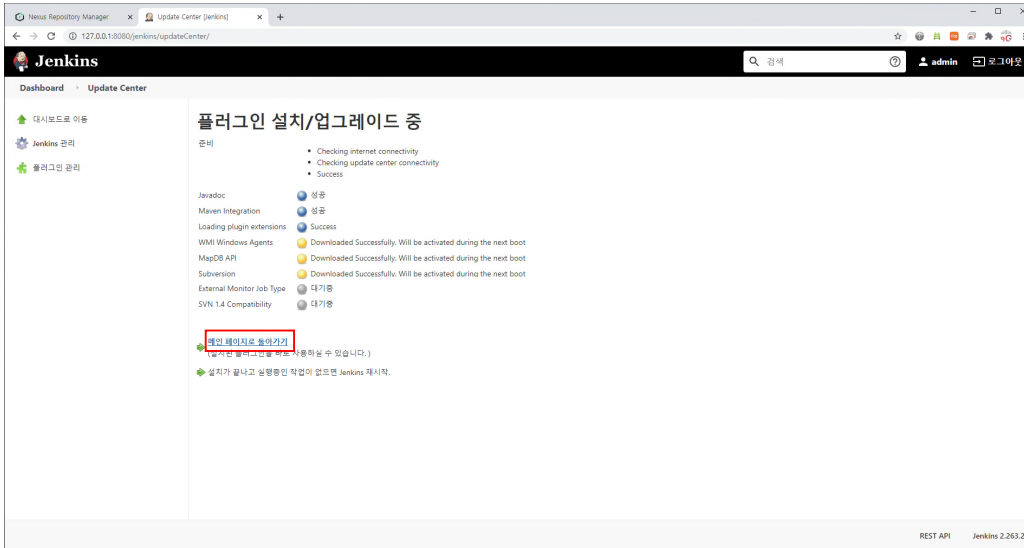
아래와 같이 플러그인을 다운로드 받을 수 있다.

The screenshot shows the Jenkins management interface. The 'Jenkins 관리' (Jenkins Management) page is displayed, with a red box highlighting the '플러그인 관리' (Plugin Management) option under the 'System Configuration' section. The page also includes sections for Security, Status Information, Troubleshooting, and Tools and Actions.

maven 과 svn 플러그인을 설치한다.

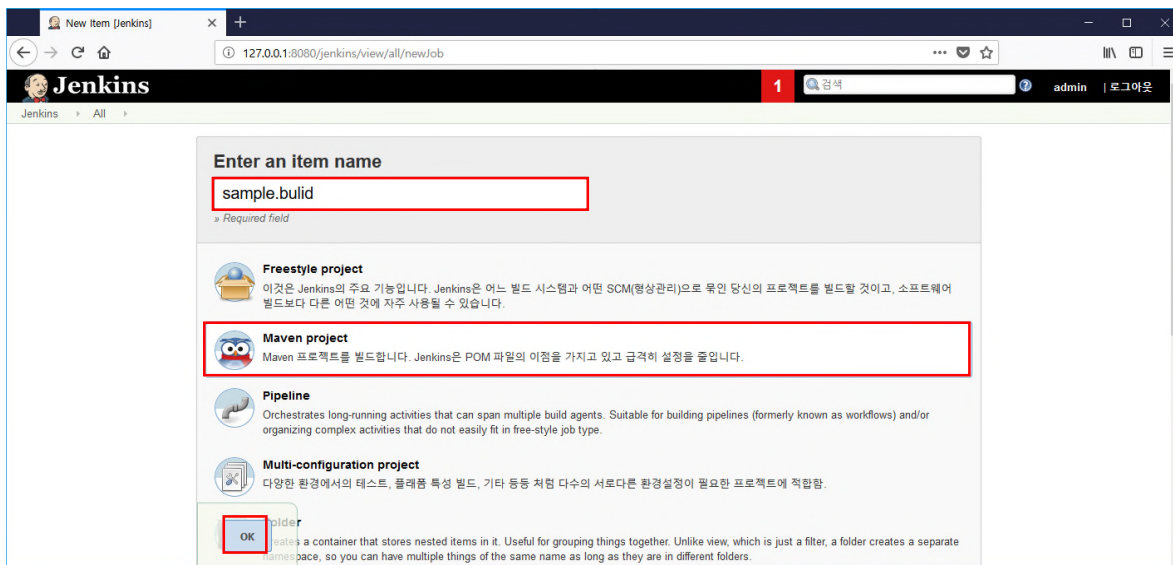


설치완료 후 메인 페이지로 돌아간다. (설치가 반영이 안되었을 경우, jenkins 를 재시작한다.)



### 3.3.2 새 작업 생성

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



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

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:

Credentials:

Local module directory:

Repository depth:

Ignore externals:

Cancel process on externals fail:

Additional Credentials:

Check-out Strategy:

Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

Quiet check-out:

Repository browser:

General 소스 코드 관리 **빌드 유발** 빌드 환경 Pre Steps Build Post Steps Build Settings 빌드 후 조치

### 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

General 소스 코드 관리 빌드 유발 **빌드 환경** Pre Steps Build Post Steps Build Settings 빌드 후 조치

### 빌드 환경

Delete workspace before build starts

고급...

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

Abort the build if it's stuck

Add timestamps to the Console Output

With Ant ?

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

General 소스 코드 관리 빌드 유발 빌드 환경 **Pre Steps** Build Post Steps Build Settings 빌드 후 조치

### Pre Steps

Add pre-build step ▾

### Build

Root POM pom.xml ?

Goals and options clean install ?

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 ?

사용자 빌드 경로 사용 ?

Maven Validation Level DEFAULT ▾

Settings file Use default maven settings ▾ ?

Global Settings file Use default maven global settings ▾ ?

General   소스 코드 관리   빌드 유발   빌드 환경   Pre Steps   Build   **Post Steps**   Build Settings   빌드 후 조치

### Post Steps

Run only if build succeeds  
 Run only if build succeeds or is unstable  
 Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Add post-build step ▾

---

### Build Settings

E-mail Notification

---

### 빌드 후 조치

빌드 후 조치 추가 ▾

저장   Apply

Dashboard [Jenkins] x +   127.0.0.1:8080/jenkins/   admin | 로그아웃

Jenkins > [자동으로 새로고침](#)   [상세 내용 입력](#)

- 새로운 Item
- 사람
- 빌드 기록
- 프로젝트 연관 관계
- 파일 핑거프린트 확인
- Jenkins 관리
- My Views
- Lockable Resources
- Credentials
- New View

**빌드 대기 목록** ▾  
 빌드 대기 항목이 없습니다.

**빌드 실행 상태** ▾  
 1 대기 중  
 2 대기 중

All +

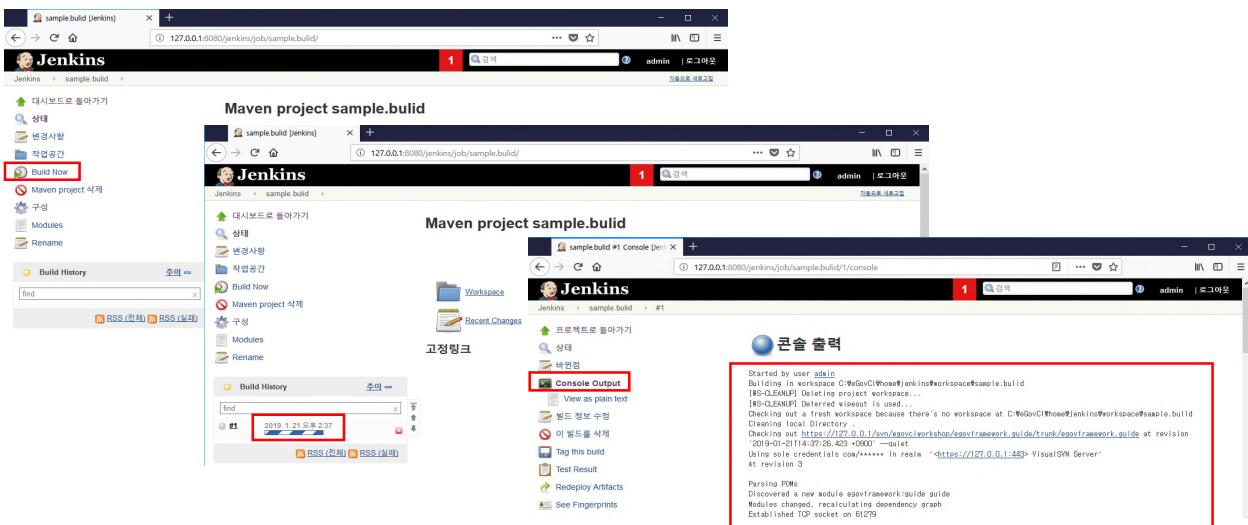
S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간
●	☀	sample.build	—	—	—

아이콘: S M L   [Legend](#)   [RSS 모두](#)   [RSS 실패](#)   [RSS 최근 빌드](#)

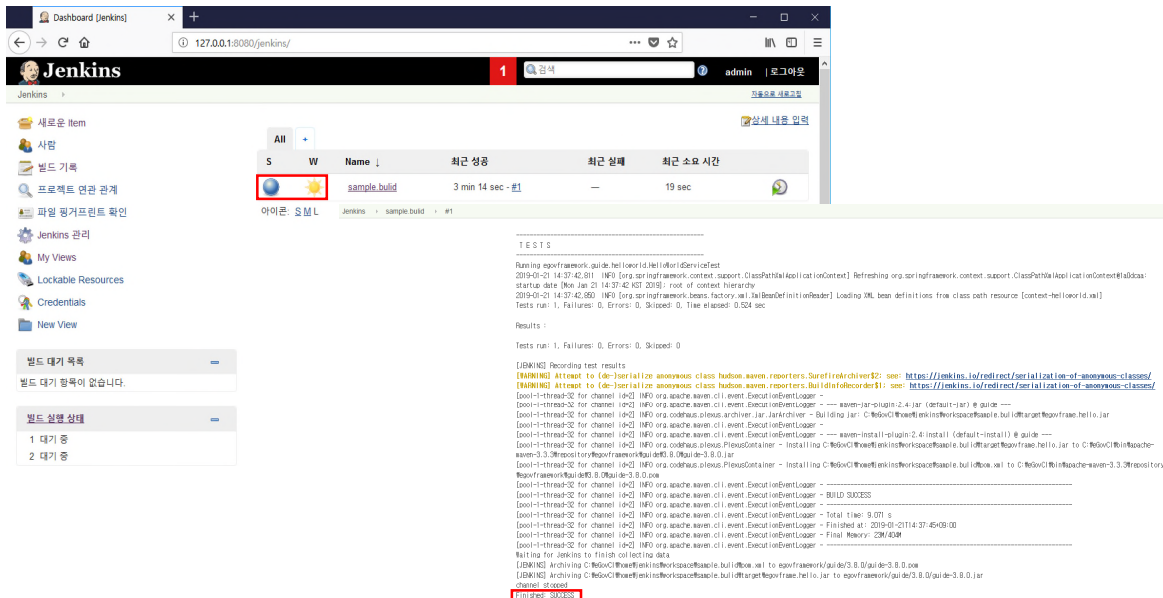
### 3.3.3 작업 Build

모든 설정 이후, 최초 수동으로 Build 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.

