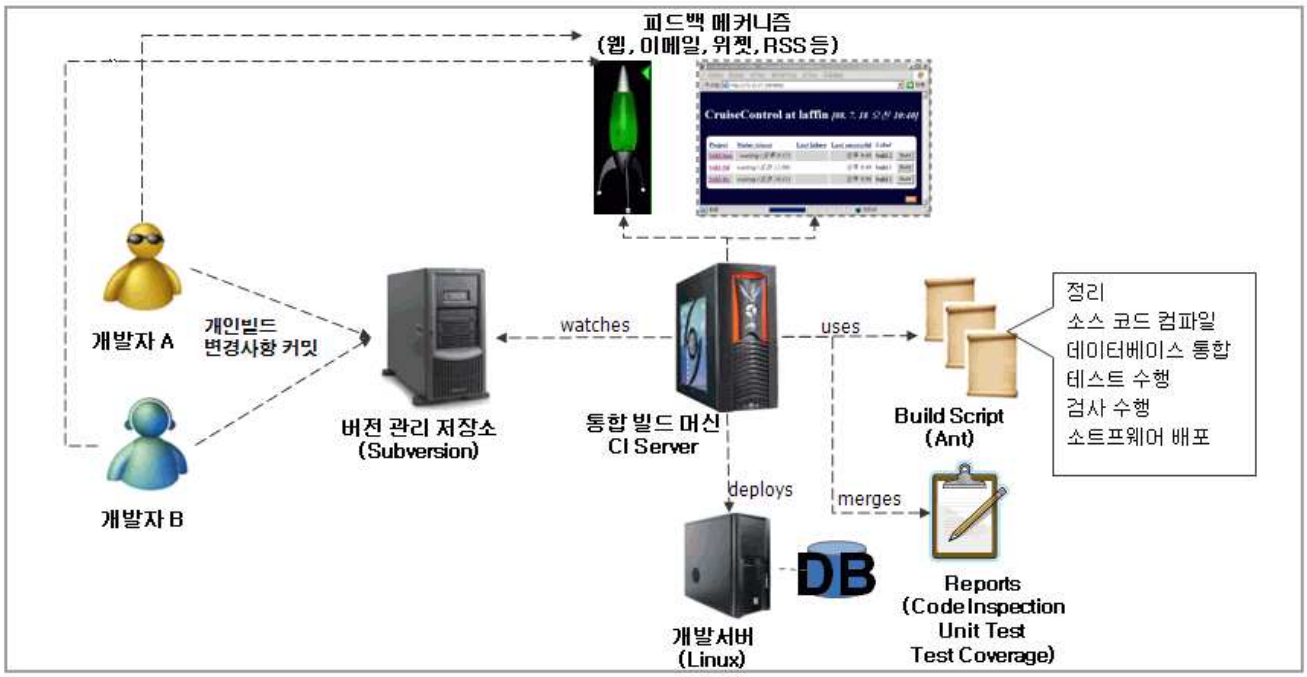
1. CI 서버

소프트웨어 개발 시 지속적 통합(continuous integration) 서비스를 제공하는 툴이다. 다수의 개발자들이 하나의 프로그램을 개발할 때 버전 충돌을 방지하기 위해 각자 작업한 내용을 공유 영역에 있는 저장소에 빈번히 업로드함으로써 지속적 통합이 가능하도록 해 준다.



2. 프로그램 설치

가. 젠킨스(jenkins3)

1) 젠킨스란

허드슨(Hudson)과 비슷한 자동배포 웹 애플리케이션입니다. 허드슨 개발자들이 나와서 제작한 툴로서, 허드슨 보다 버전이 낮지만 강력한 기능을 자랑합니다.

소프트웨어 프로젝트 빌드, cron Job과 같은 반복 작업을 모니터 하기 위한 웹 어플리케이션으로 웹 어플리케이션의 경우 SVN update, Maven 빌드, Tomcat deploy의 과정을 원하는 시간대에 주기적으로 실행 가능.

2) Jenkins 주요 기능

- 소프트웨어 자동 빌드
  빌드 주기에 따른 일일 빌드 또는 주간 빌드 기능 제공
- 지속적이고 자동화된 빌드 검증
  SCM 폴링(polling) 기능을 통한 최신 코드 기반의 빌드 수행
- 지속적이고 자동화된 빌드 테스트
  테스트 슈트 실행을 통한 코드 품질 검증
- 빌드 후속 절차 자동화
  컴파일 된 코드의 패키징 및 테스트 리포팅 기능 제공

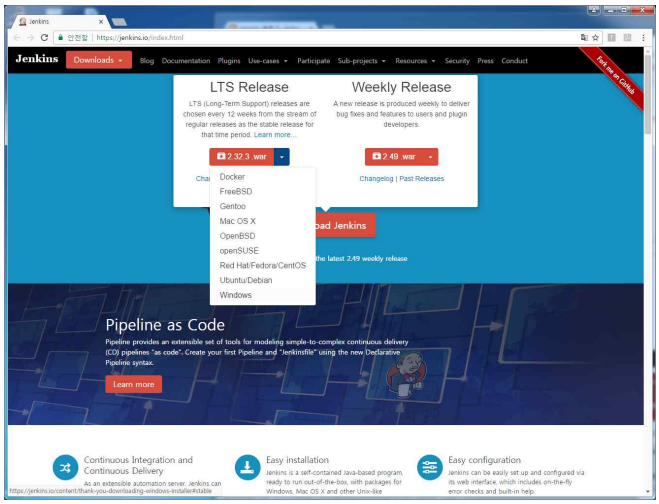
3) 참고사이트

설치방법 : http://sayingublog.blogspot.kr/2015/06/jenkin.html

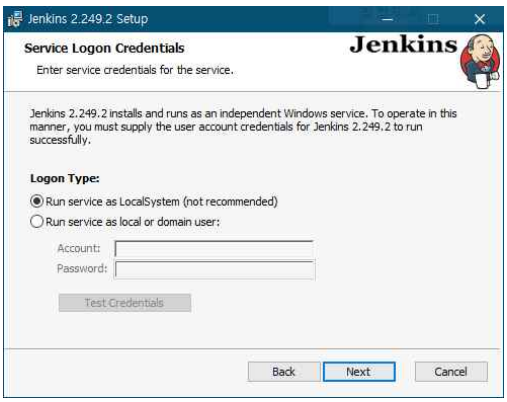
4) 다운로드

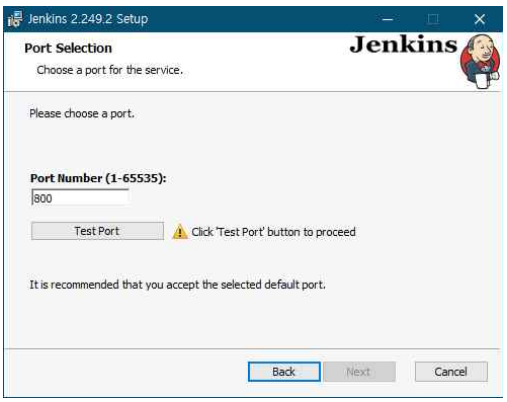
다운로드: https://jenkins-ci.org/

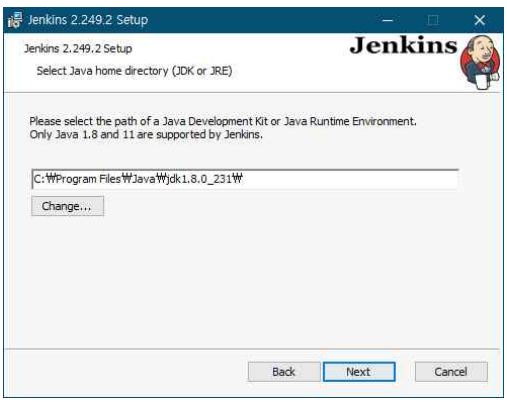
파일명 : jenkins-2.131.zip (Windows)

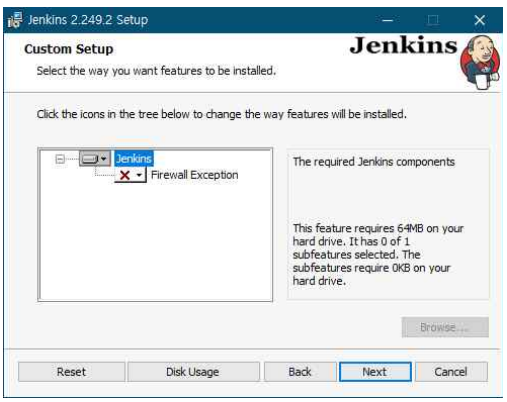


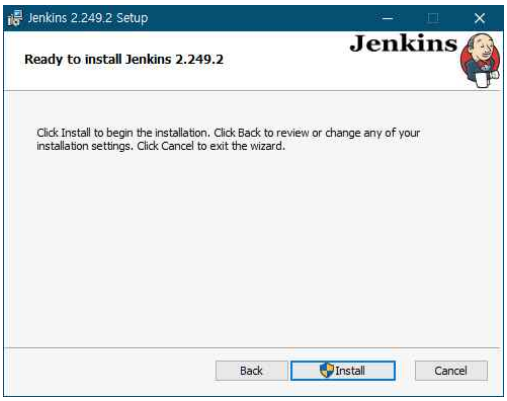
5) 설치하기

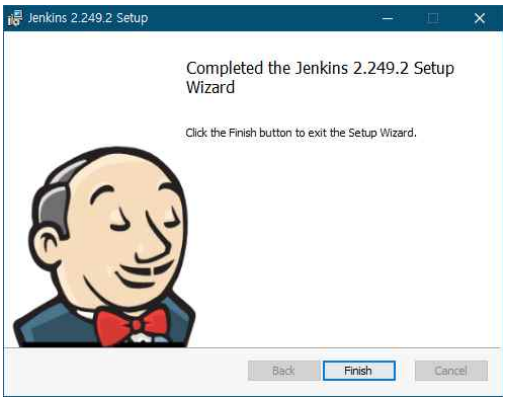








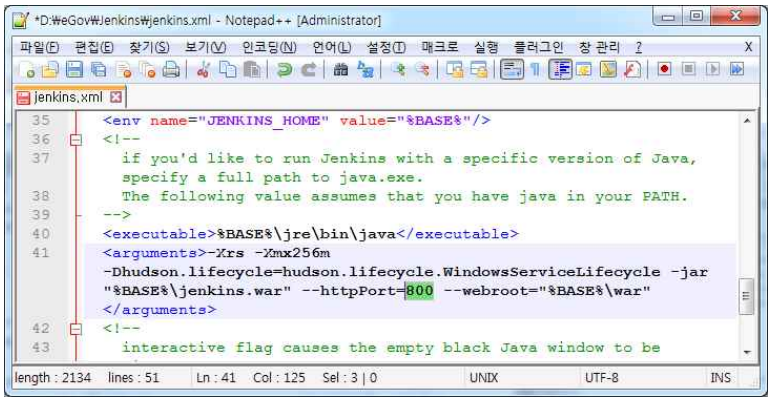




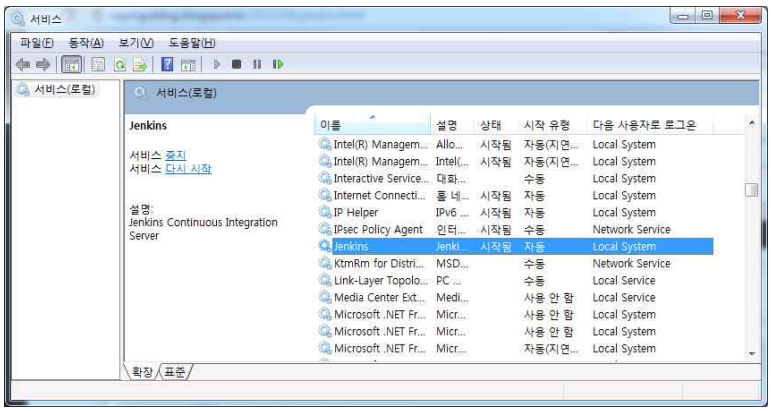
6) Jenkins 포트변경

파일위치: d:₩dev₩Jenkins₩Jenkins.xml

포트 : 800



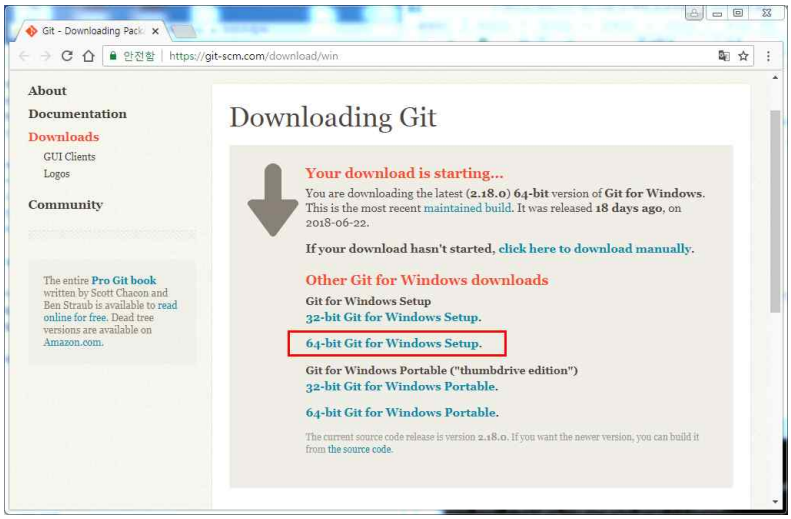
7) Jenkins 서비스 시작



나. git 설치

사이트: https://git-scm.com/download/win

파일: Git-2.18.0-64-bit.exe



다. maven 설치

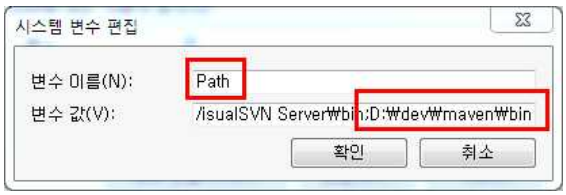
설치 가이드: https://zetawiki.com/wiki/윈도우_메이븐_설치

사이트 : http://maven.apache.org/download.cgi

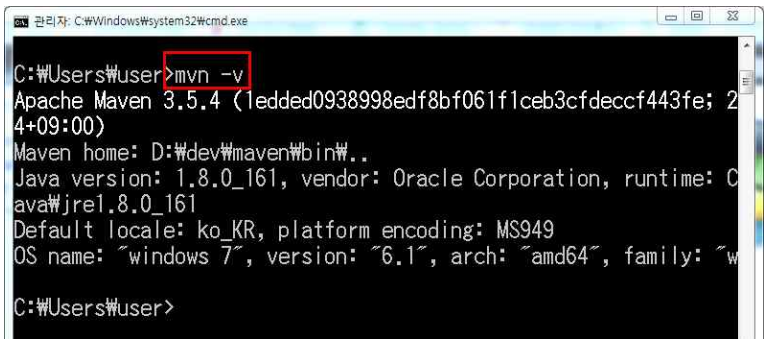
파일: apache-maven-3.5.4-bin.zip

1) 압축 해제하여 설치하기
d:₩dev에 압축 해제.

2) 환경변수 path 수정
시스템 환경변수 path에 maven 실행경로를 추가한다.

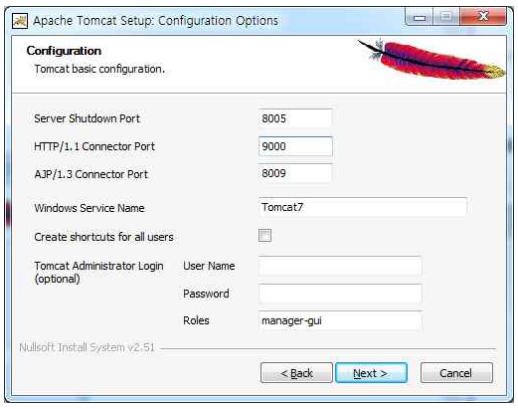


3) 설치 후 확인

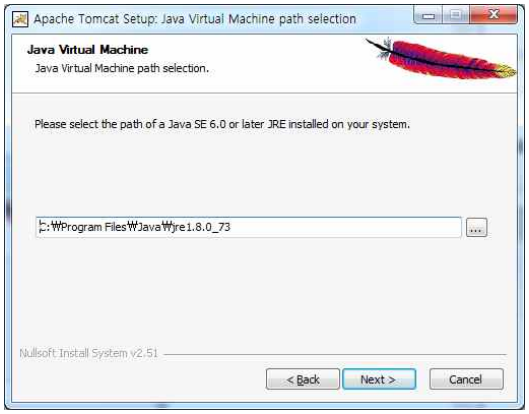


라. 톰캣 설치

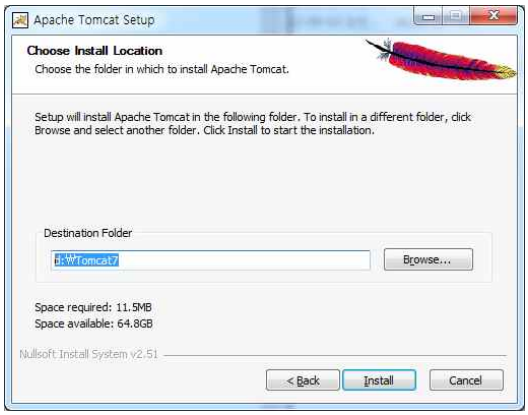
1) 설치하기
- 포트 변경 : 80



- jre 경로 확인



- 설치경로 변경 : d:₩dev₩Tomcat8 (공백이 없도록 함)



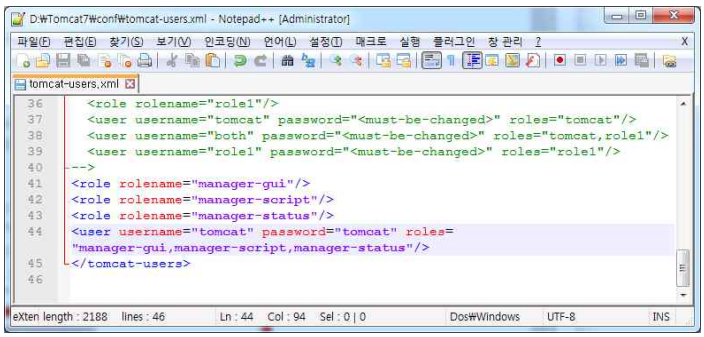
2) 사용자 권한 변경

- 톰캣을 외부 또는 웹에서 deploy하기 위해서 deploy관련 서비스에 대해서 권한설정을 해줘야 한다. 원격서버로 배포하는 것이 목적이므로 manager-script 권한만 설정해도 된다.

참고사이트: http://tomcat.apache.org/tomcat-8.5-doc/manager-howto.html

- 파일위치 : **d:₩dev₩Tomcat8₩conf₩tomcat-users.xml**

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager-status"/>
<user username="tomcat" password="tomcat" roles="manager-gui,manager-script,manager-status"/>
```



manager-gui — Access to the HTML interface.

manager-status — Access to the "Server Status" page only.

manager-script — Access to the tools-friendly plain text interface that is described in this document, and to the "Server Status" page.

manager-jmx — Access to JMX proxy interface and to the "Server Status" page.

참고사항

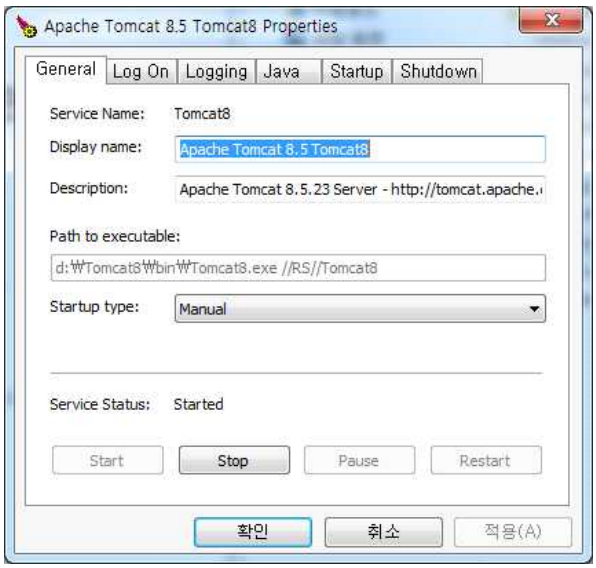
에러 메시지 : The username you provided is not allowed to use the text-based Tomcat Manager (error 403) when deploying on remote Tomcat8 using Jenkins

조치사항 : Edit the file d:/tomcat/webapps/manager/META-INF/context.xml 파일에서 RemoteAddrValve 설정을 주석으로 막는다

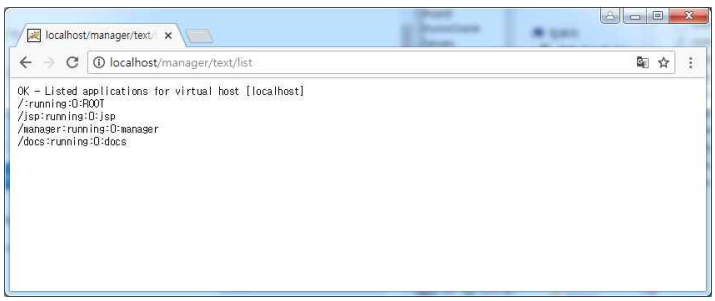
```
<Context antiResourceLocking="false" privileged="true">
  <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127₩.₩d+₩.₩d+₩.₩d+|::1|0:0:0:0:0:0:0:1" />
</Context>
```

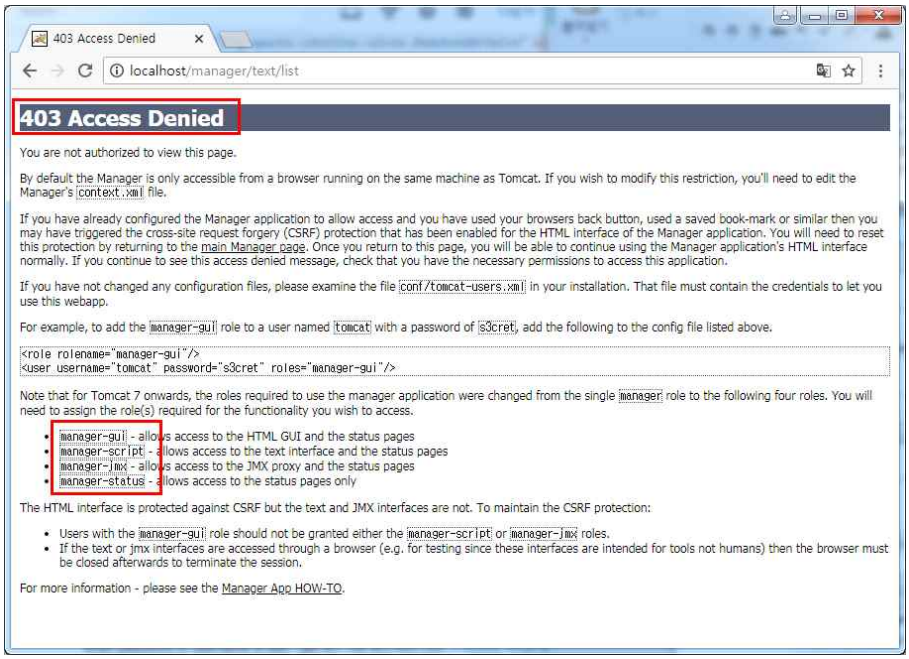
```
<Context antiResourceLocking="false" privileged="true">
  <!--  <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127₩.₩d+₩.₩d+₩.₩d+|::1|0:0:0:0:0:0:0:1" />  -->
</Context>
```

3) tomcat 서비스 시작



4) 권한이 없는 경우 에러




403 Access Denied
localhost/manager/text/list
403 Access Denied
You are not authorized to view this page.
By default the Manager is only accessible from a browser running on the same machine as Tomcat. If you wish to modify this restriction, you'll need to edit the Manager's context.xml file.
If you have already configured the Manager application to allow access and you have used your browsers back button, used a saved book-mark or similar then you may have triggered the cross-site request forgery (CSRF) protection that has been enabled for the HTML interface of the Manager application. You will need to reset this protection by returning to the main Manager page. Once you return to this page, you will be able to continue using the Manager application's HTML interface normally. If you continue to see this access denied message, check that you have the necessary permissions to access this application.
If you have not changed any configuration files, please examine the file conf/tomcat-users.xml in your installation. That file must contain the credentials to let you use this webapp.
For example, to add the manager-gui role to a user named tomcat with a password of s3cret, add the following to the config file listed above.
<role rolename="manager-gui"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
Note that for Tomcat 7 onwards, the roles required to use the manager application were changed from the single manager role to the following four roles. You will need to assign the role(s) required for the functionality you wish to access.
manager-gui - allows access to the HTML GUI and the status pages
manager-script - allows access to the text interface and the status pages
manager-jmx - allows access to the JMX proxy and the status pages
manager-status - allows access to the status pages only
The HTML interface is protected against CSRF but the text and JMX interfaces are not. To maintain the CSRF protection:
Users with the manager-gui role should not be granted either the manager-script or manager-jmx roles.
If the text or jmx interfaces are accessed through a browser (e.g. for testing since these interfaces are intended for tools not humans) then the browser must be closed afterwards to terminate the session.
For more information - please see the Manager App HOW-TO.

3. Jenkins 시작하기

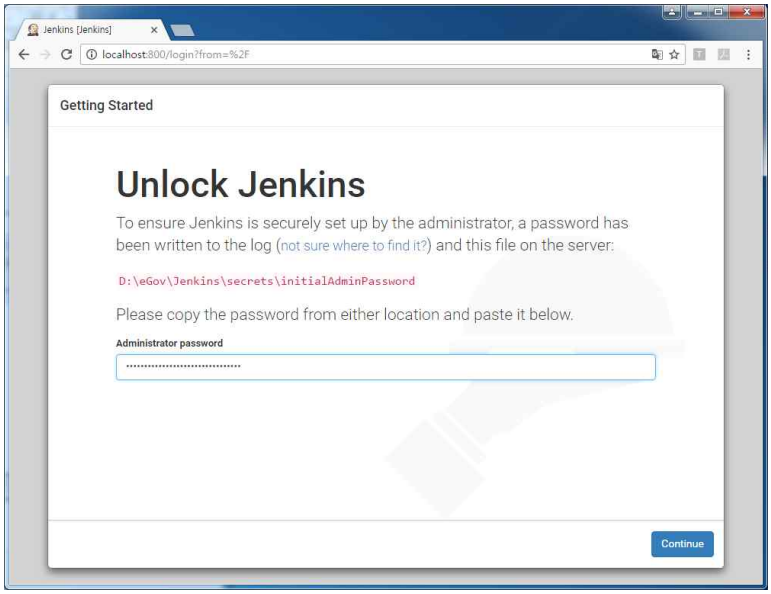
1) Jenkins 접속

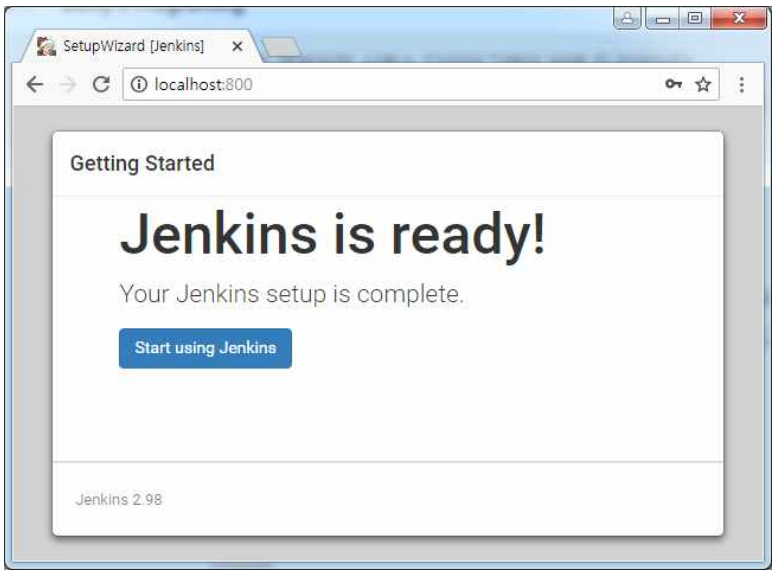
URL: localhost:800

2) 관리자 패스워드 입력

-> 패스워드파일 참조: d:₩dev₩Jenkins₩secrets₩initialAdminPassword

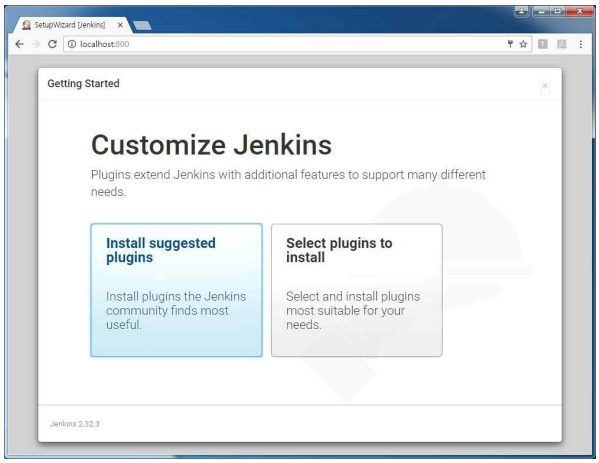
패스워드 파일의 패스워드를 복사하여 붙여 넣고 continue 버튼 클릭하여 다음으로 넘어간다.

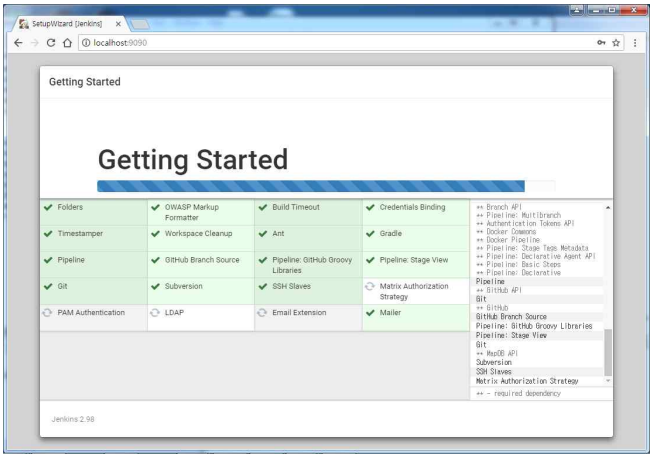




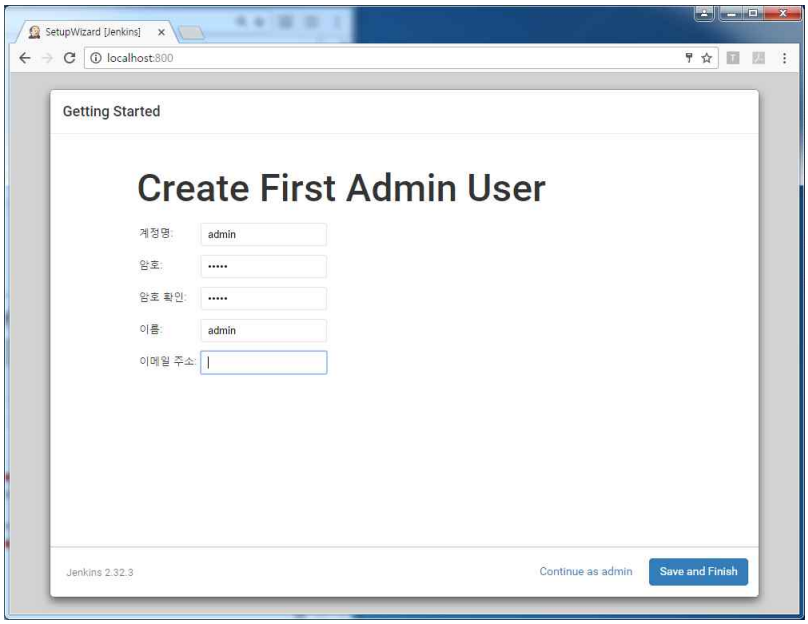
3) 플러그인 설치

Install suggested Plugins 버튼을 클릭하여 추천 플러그인을 설치한다.





4) 관리자 계정 지정



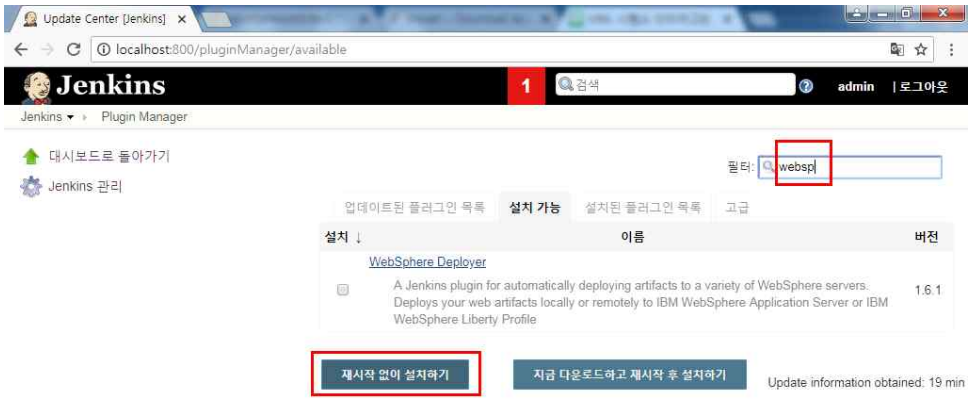
5) 젠킨스 시작

나. 젠킨스 관리

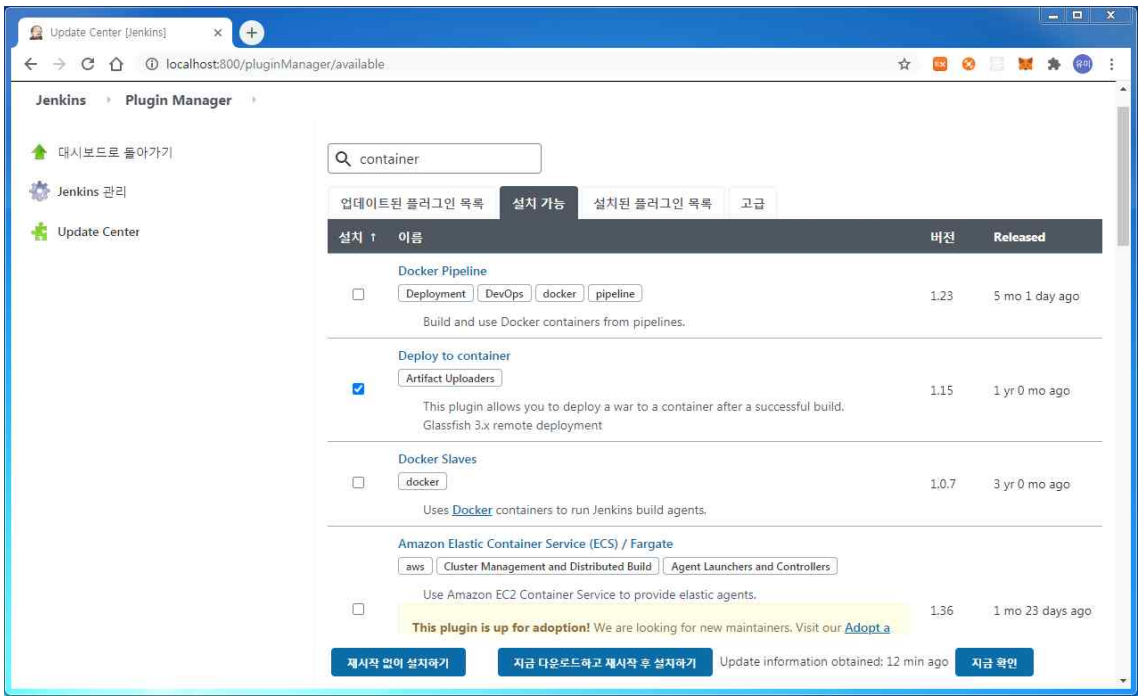
1) 플러그인 추가 설치

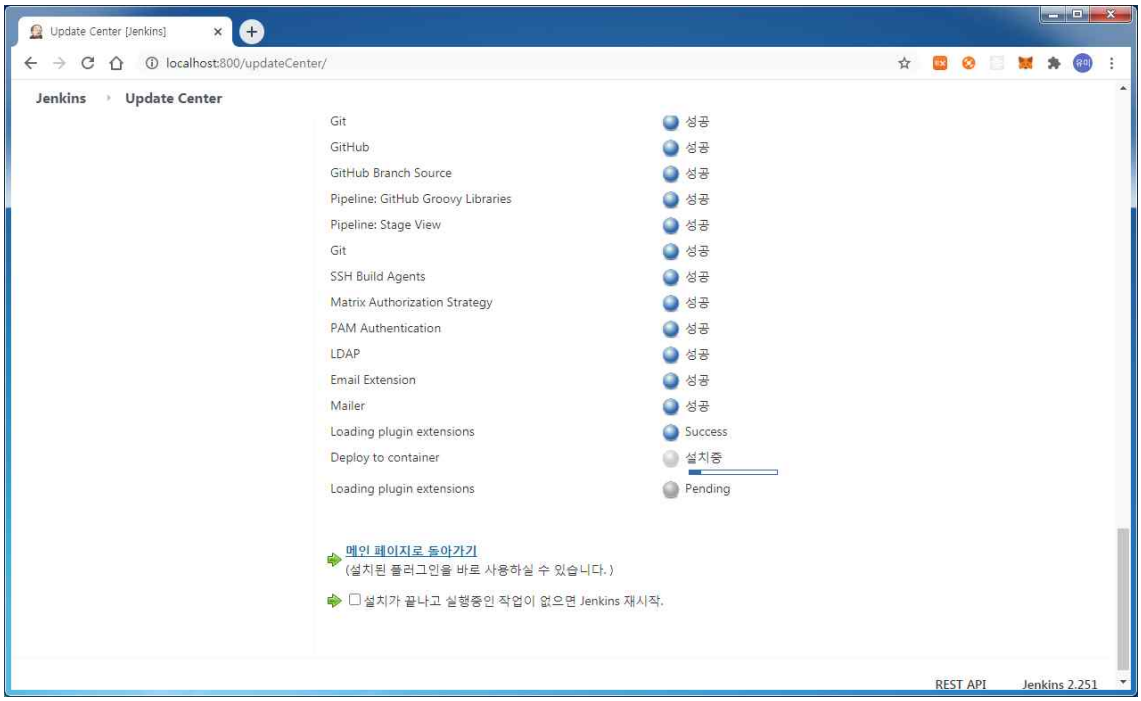
Jenkins관리 -> 플러그인 관리 -> 설치가능 탭

- "websphere" 검색하여 설치



"deploy" 검색하여 Deploy to container Plugin 항목 선택 -> [재시작 없이 설치하기] 버튼 클릭 --> [빌드후 조치] 메뉴에 "Deploy war/ear to a container" 체크박스 메뉴가 생긴다.

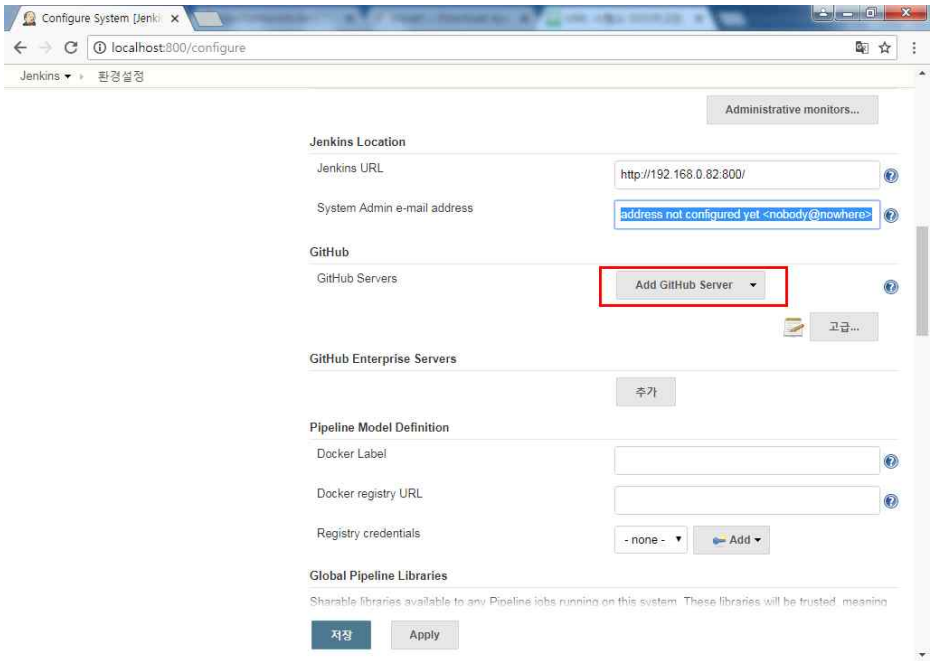




2) 시스템 설정

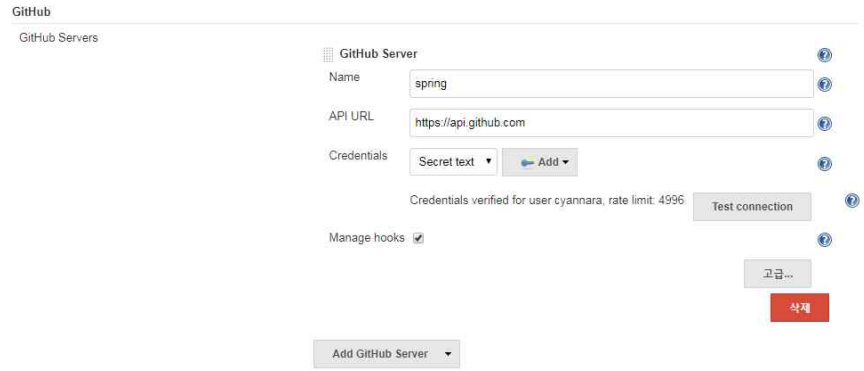
Jenkins관리 -> 시스템 설정 -> gitHub server 설정

- Add Github Server 클릭

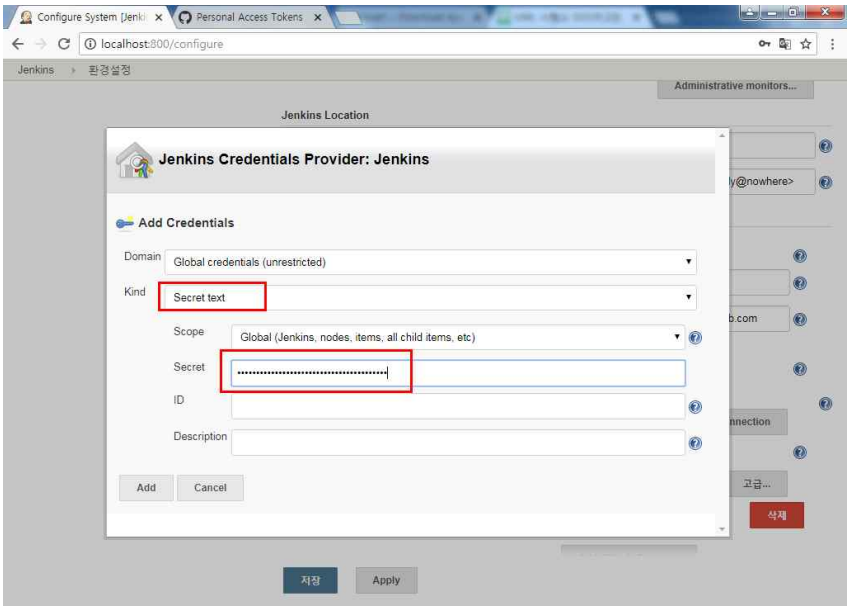


- git 접속계정 등록

Credentilas -> Add



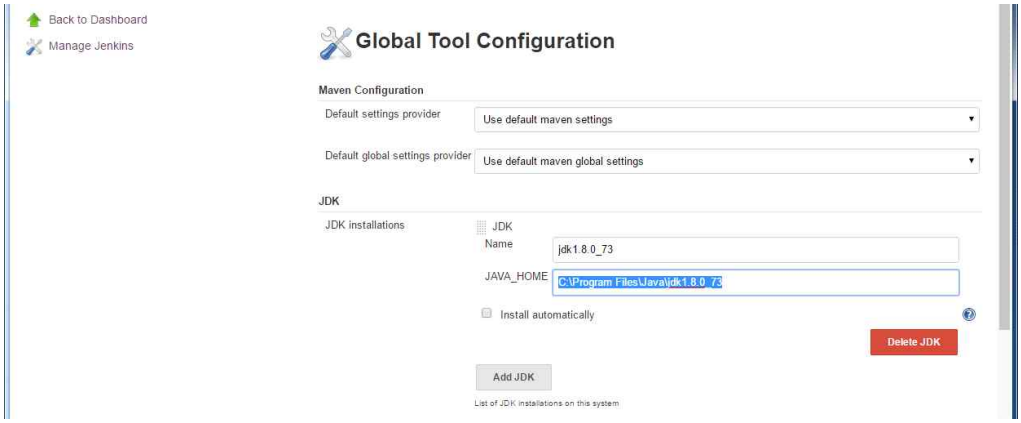
kind는 Sceret Text 선택
Secret에는 git token 입력
[add] 클릭하여 저장



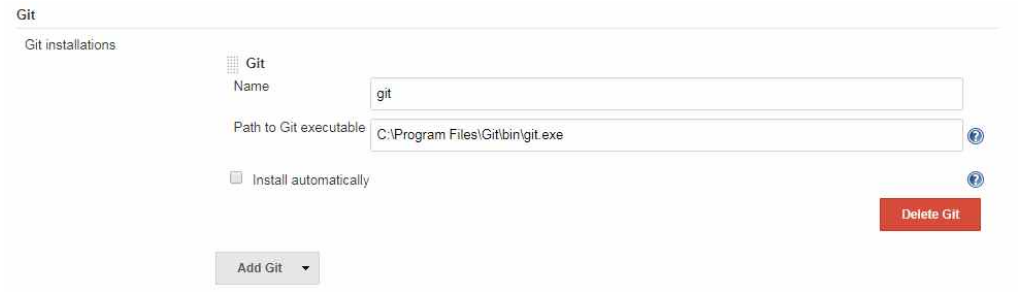
- "Test connection" 버튼 클릭하여 연결 확인.
저장 버튼 클릭하여 저장하기

3) 환경설정
Jenkins관리 -> Global Tool Configuration -> **JDK, GIT, MAVEN 설정**

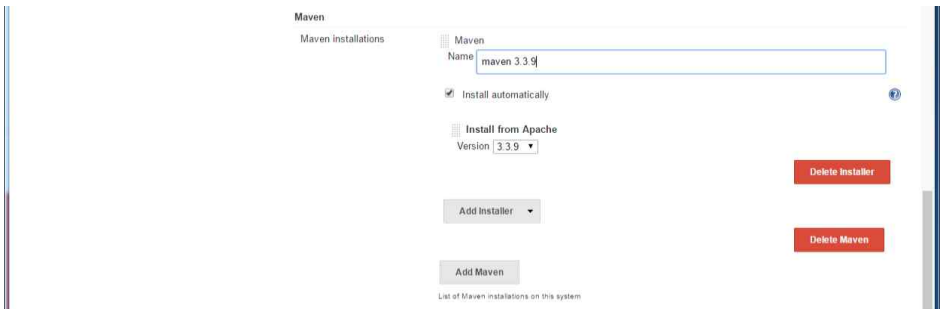
- JDK : Name 과 JAVA_HOME 지정
Install aotumatically 체크박스 해제하고 자바 설치 경로를 지정한다.



- git



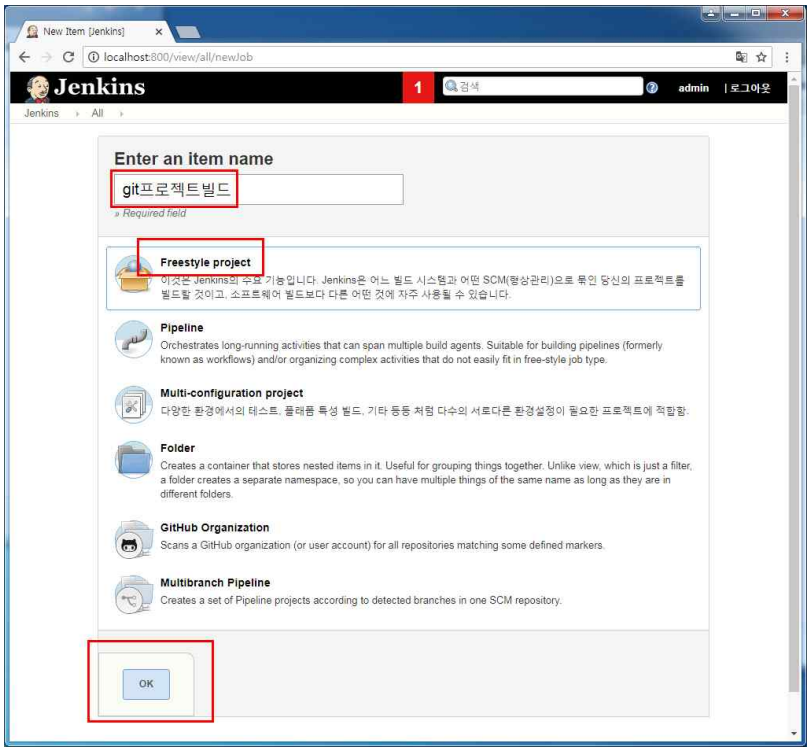
- Maven
- maven 설치 안 된 경우 : version 선택하고 Install automatically 체크하여 설치되도록 함



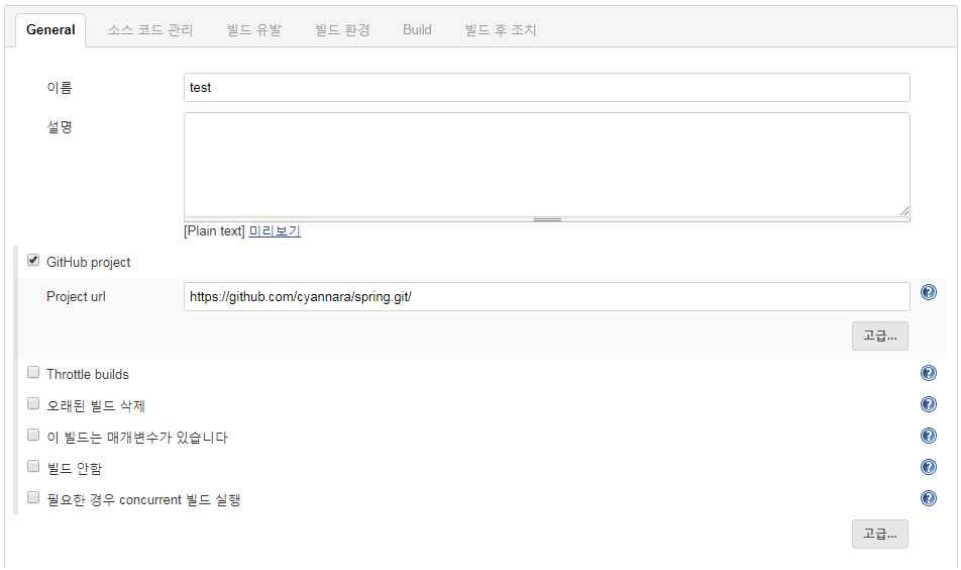
- maven 설치 된 경우에는 Install automatically 체크 해제하고 설치경로 지정

다. 새작업 추가(GitHub 연동)

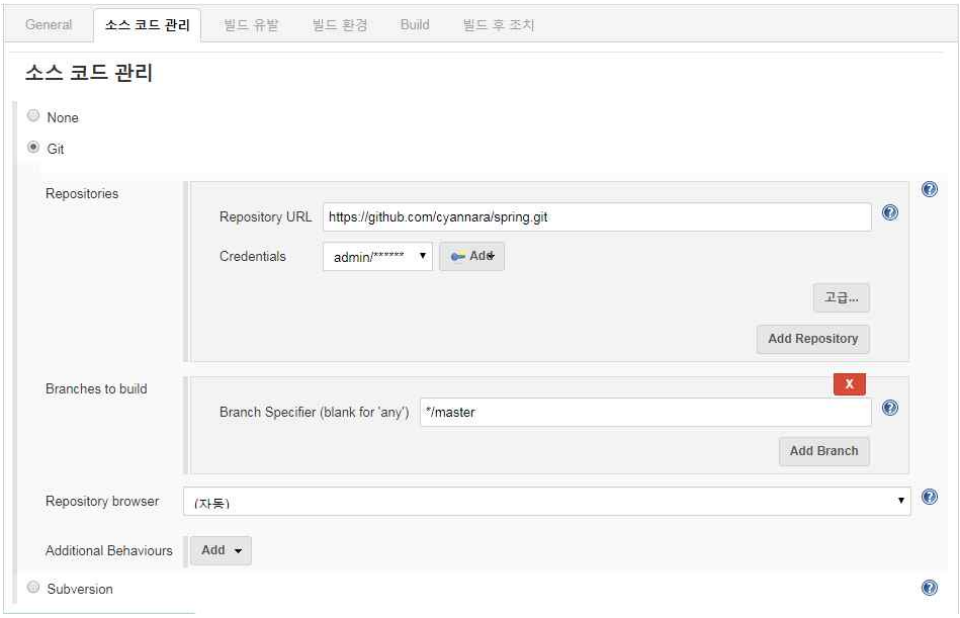
1) 새작업


New Item [Jenkins]
localhost:800/view/all/newJob
Jenkins
admin | 로그아웃
Jenkins › All ›
Enter an item name
git프로젝트빌드
» Required field
Freestyle project
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.
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, 플래폼 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.
GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.
OK

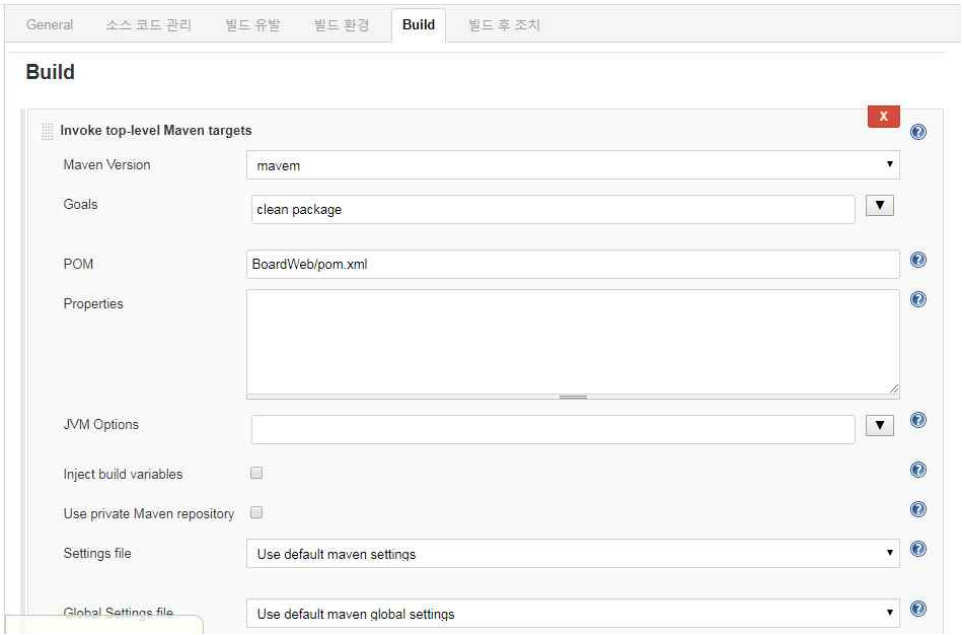
2) General


General
소스 코드 관리
빌드 유발
빌드 환경
Build
빌드 후 조치
이름
test
설명
[Plain text] 미리보기
GitHub project
Project url
https://github.com/cyannara/spring.git/
고급...
Throttle builds
오래된 빌드 삭제
이 빌드는 매개변수가 있습니다
빌드 안함
필요한 경우 concurrent 빌드 실행
고급...

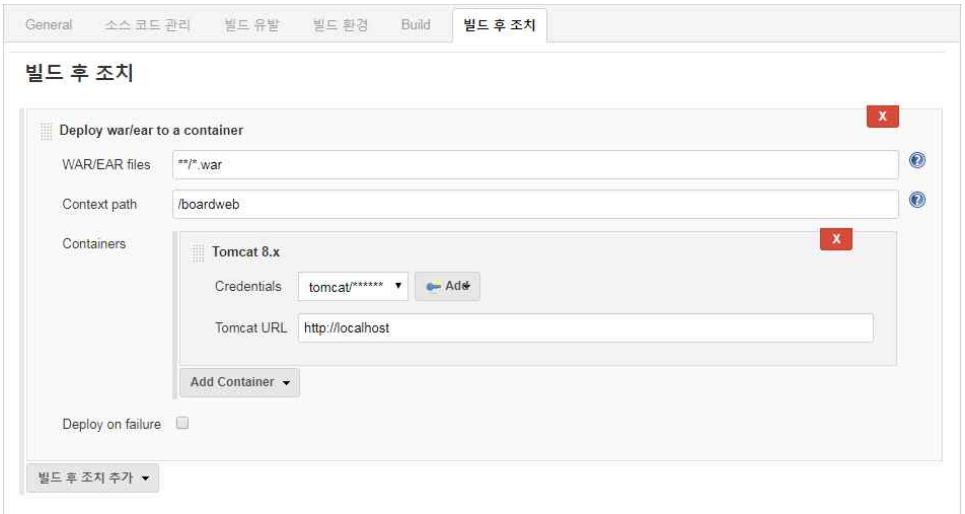
3) 소스 코드 관리



4) 빌드



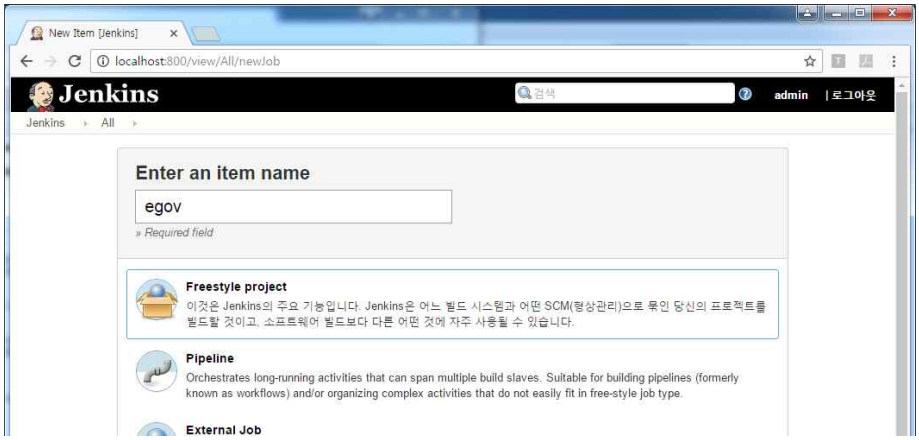
5) 빌드 후 조치



라. 새작업 추가(SVN 리포지토리 연결인 경우)

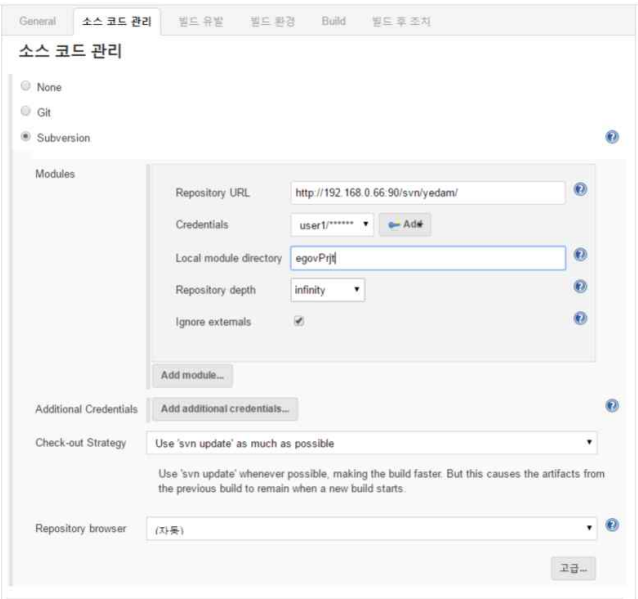
1) [새로운 Item] 메뉴 선택

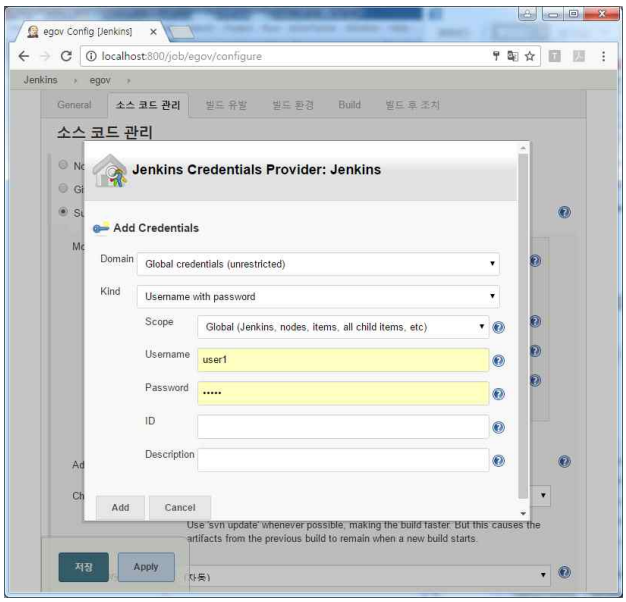
- item name 입력하고 "Freestyle project" 선택 하고 "OK" 버튼 클릭



2) 소스 코드 관리

- Subversion 선택
- Repository URL 입력 : SVN url
- Credential : SVN 계정

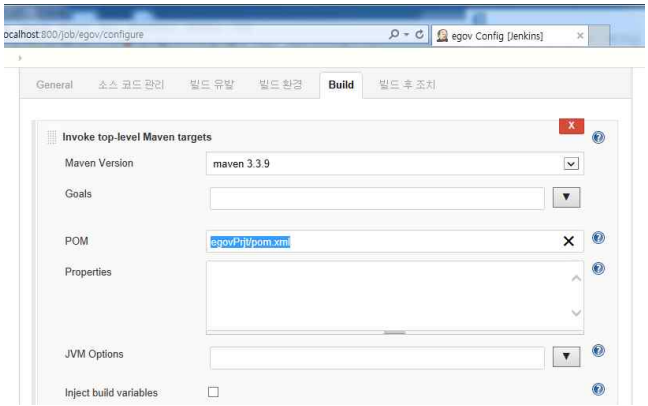




- Local module directory : 빌드된 war 파일이 저장될 위치
  D:₩eGov₩Jenkins₩jobs₩egov₩workspace 에 폴더가 생성

3) Build

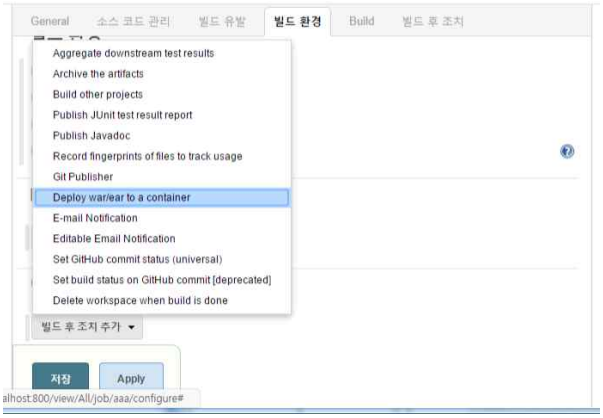
고급버튼 클릭한 후 POM 파일 위치 입력



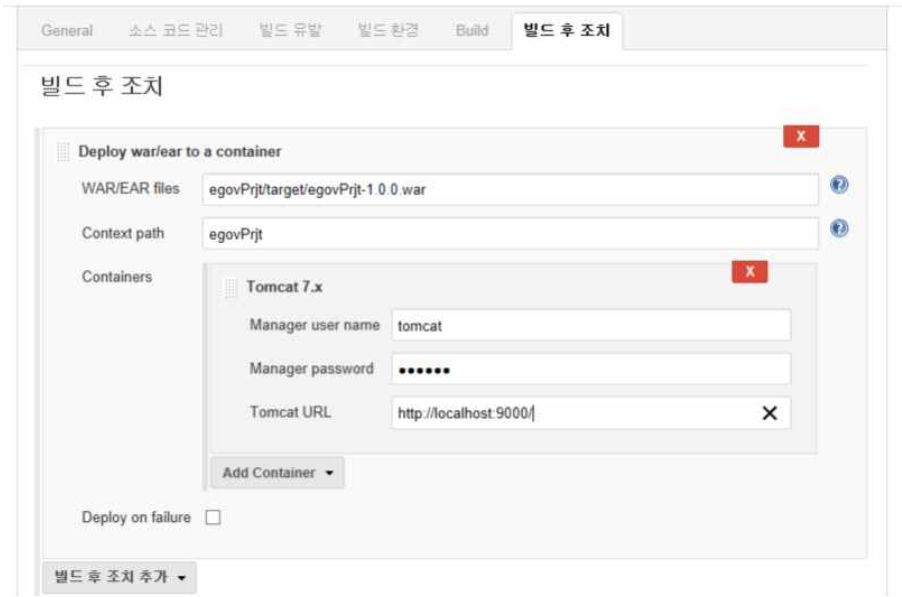
4) 빌드 후 조치

빌드 후 조치 부분에 [빌드 후 조치 추가] 버튼을 클릭 후 Deploy war/ear to a conatiner 선택한 후 Tomcat Deploy 설정 후 저장

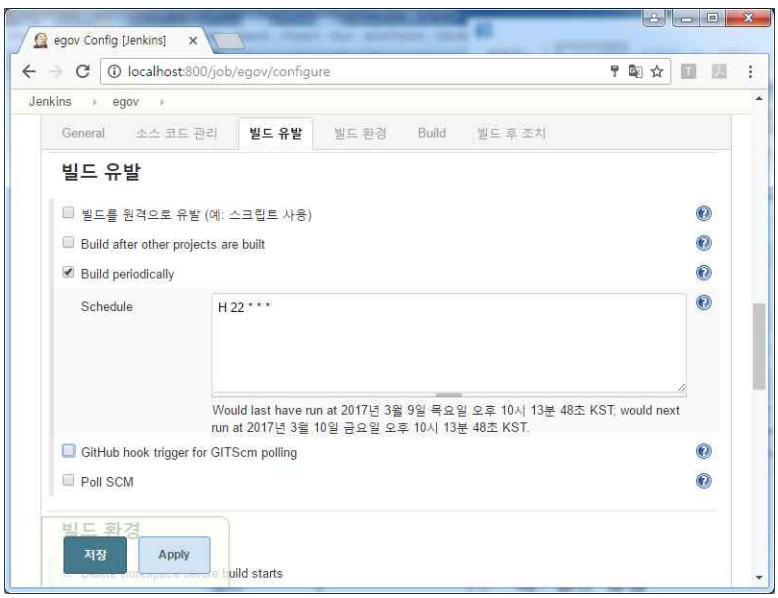
- Deploy War 선택



- tomcat URL, 계정, war 경로명, context path 입력



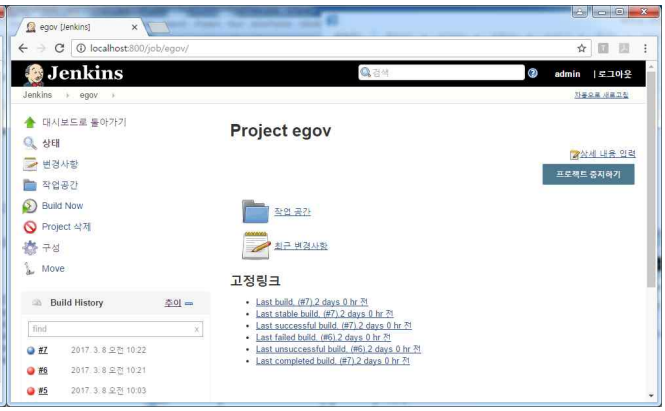
5) 빌드 유발



마. 작업 생성 완료 확인

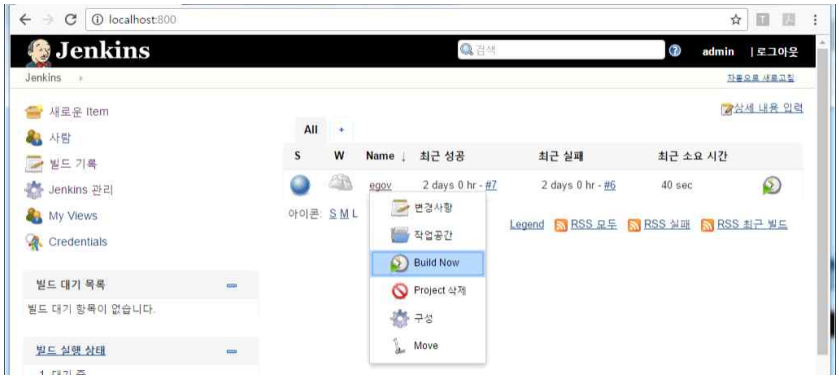
- 작업 목록

- 작업 상세



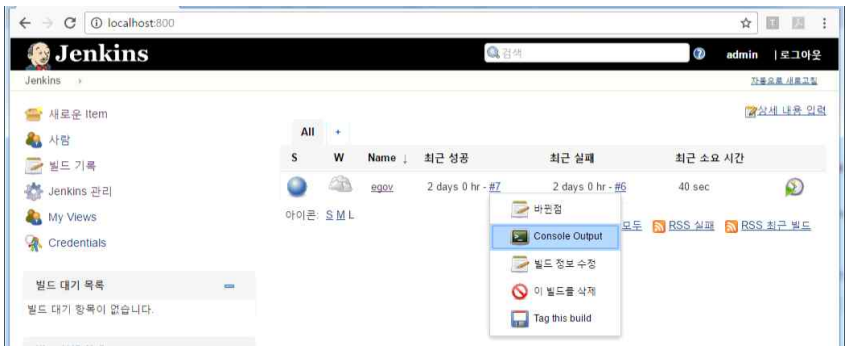
바. 빌드

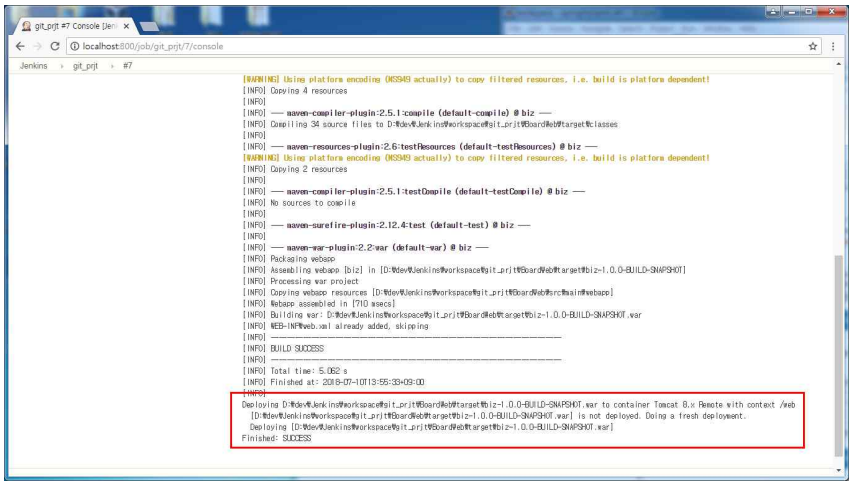
1) 빌드



2) console output

- 빌드 메시지 확인





- build success

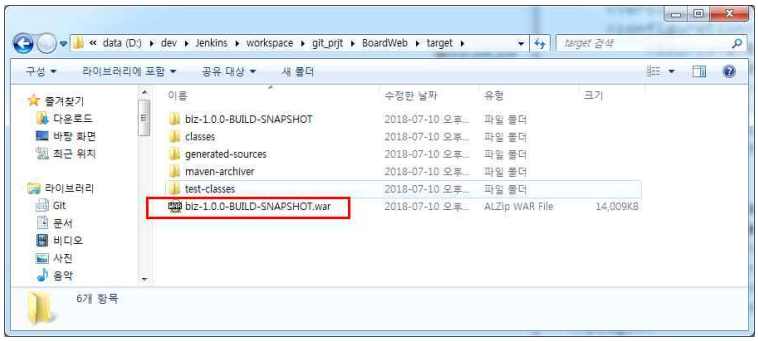
```
Downloaded: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar (226 KB at 262.3 KB/sec)
[INFO] Installing D:₩eGov₩Jenkins₩jobs₩egov₩workspace₩egovPrjt₩egovPrjt₩target₩egovPrjt-1.0.0.war to
C:₩Windows₩system32₩config₩systemprofile₩.m2₩repository₩yedam₩com₩egovPrjt₩1.0.0₩egovPrjt-1.0.0.war
[INFO] Installing D:₩eGov₩Jenkins₩jobs₩egov₩workspace₩egovPrjt₩egovPrjt₩pom.xml to
C:₩Windows₩system32₩config₩systemprofile₩.m2₩repository₩yedam₩com₩egovPrjt₩1.0.0₩egovPrjt-1.0.0.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 04:04 min
[INFO] Finished at: 2017-03-08T09:45:34+09:00

[INFO] Final Memory: 32M/241M
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

- 빌드후 조치 success

```
[INFO] ------------------------------------------------------------------------
Deploying D:₩eGov₩Jenkins₩jobs₩egov₩workspace₩egovPrjt₩egovPrjt₩target₩egovPrjt-1.0.0.war to container Tomcat 7.x Remote
  [D:₩eGov₩Jenkins₩jobs₩egov₩workspace₩egovPrjt₩egovPrjt₩target₩egovPrjt-1.0.0.war] is not deployed. Doing a fresh deployment.
  Deploying [D:₩eGov₩Jenkins₩jobs₩egov₩workspace₩egovPrjt₩egovPrjt₩target₩egovPrjt-1.0.0.war]

Finished: SUCCESS
```



빌드 결과로 war 파일이 생성됨

```
[INFO] Building war: D:₩eclipse-oxygen₩jenkins₩workspace₩test₩BoardWeb₩target₩biz-1.0.0-BUILD-SNAPSHOT.war
[INFO] WEB-INF₩web.xml already added, skipping
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.719 s
[INFO] Finished at: 2017-12-29T06:49:14+09:00
[INFO] Final Memory: 18M/179M
[INFO] ------------------------------------------------------------------------
Deploying D:₩eclipse-oxygen₩jenkins₩workspace₩test₩BoardWeb₩target₩biz-1.0.0-BUILD-SNAPSHOT.war to container Tomcat 8.x Remote with context /boardweb
  [D:₩eclipse-oxygen₩jenkins₩workspace₩test₩BoardWeb₩target₩biz-1.0.0-BUILD-SNAPSHOT.war] is not deployed. Doing a fresh deployment.
  Deploying [D:₩eclipse-oxygen₩jenkins₩workspace₩test₩BoardWeb₩target₩biz-1.0.0-BUILD-SNAPSHOT.war]
Finished: SUCCESS
```

- 빌드 후 조치 오류

```
ERROR: Step 'Deploy war/ear to a container' aborted due to exception:
java.io.IOException: Expecting Ant GLOB pattern, but saw 'D:/eGov/Jenkins/jobs/egov/workspace/egovPrjt/egovPrjt/target/egovPrjt-1.0.0.war'. See http://ant.apache.org/manual/Types/fileset.html for syntax
        at hudson.FilePath.glob(FilePath.java:1741)
        at hudson.FilePath.access$1700(FilePath.java:195)
        at hudson.FilePath$32.invoke(FilePath.java:1722)
        at hudson.FilePath$32.invoke(FilePath.java:1719)
        at hudson.FilePath.act(FilePath.java:996)
        at hudson.FilePath.act(FilePath.java:974)
        at hudson.FilePath.list(FilePath.java:1719)
        at hudson.FilePath.list(FilePath.java:1704)
        at hudson.FilePath.list(FilePath.java:1690)
        at hudson.plugins.deploy.DeployPublisher.perform(DeployPublisher.java:59)
        at hudson.tasks.BuildStepMonitor$3.perform(BuildStepMonitor.java:45)
        at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:779)
        at hudson.model.AbstractBuild$AbstractBuildExecution.performAllBuildSteps(AbstractBuild.java:720)
        at hudson.model.Build$BuildExecution.post2(Build.java:185)
        at hudson.model.AbstractBuild$AbstractBuildExecution.post(AbstractBuild.java:665)
        at hudson.model.Run.execute(Run.java:1753)
        at hudson.model.FreeStyleBuild.run(FreeStyleBuild.java:43)
        at hudson.model.ResourceController.execute(ResourceController.java:98)
        at hudson.model.Executor.run(Executor.java:404)
Finished: FAILURE
```

사. 오류 발생 시 조치 방법

1) pom.xml 파일 위치가 지정 안 된 경우

```
[egovPrjt] $ cmd.exe /C "D:₩eGov₩Jenkins₩tools₩hudson.tasks.Maven_MavenInstallation₩maven_3.3.9₩bin₩mvn.cmd && exit %%ERRORLEVEL%%"
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.114 s
[INFO] Finished at: 2017-03-08T09:36:10+09:00
[INFO] Final Memory: 5M/123M
[INFO] ------------------------------------------------------------------------
[ERROR] No goals have been specified for this build. You must specify a valid lifecycle phase or a goal in the format <plugin-prefix>:<goal> or <plugin-group-id>:<plugin-artifact-id>[:<plugin-version>]:<goal>. Available lifecycle phases are: validate, initialize, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes, generate-test-sources, process-test-sources, generate-test-resources, process-test-resources, test-compile, process-test-classes, test, prepare-package, package, pre-integration-test, integration-test, post-integration-test, verify, install, deploy, pre-clean, clean, post-clean, pre-site, site, post-site, site-deploy. -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/NoGoalSpecifiedException
Build step 'Invoke top-level Maven targets' marked build as failure
```

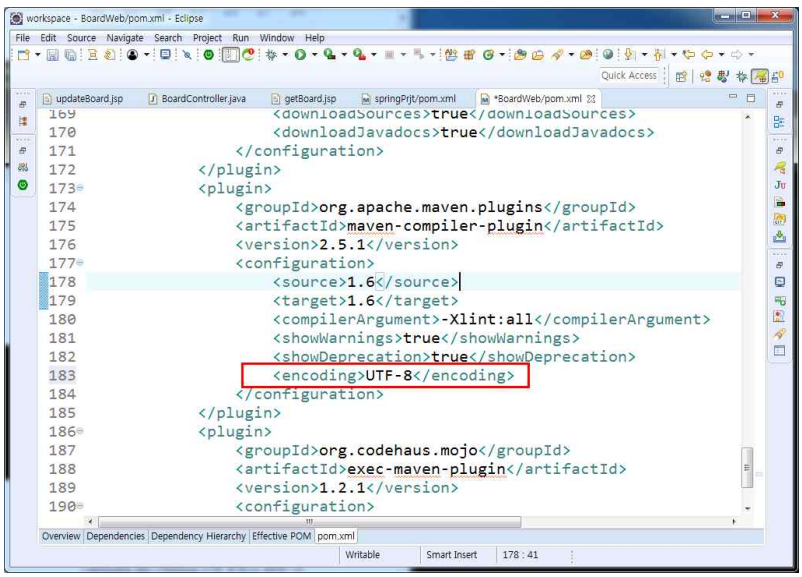
Finished: FAILURE

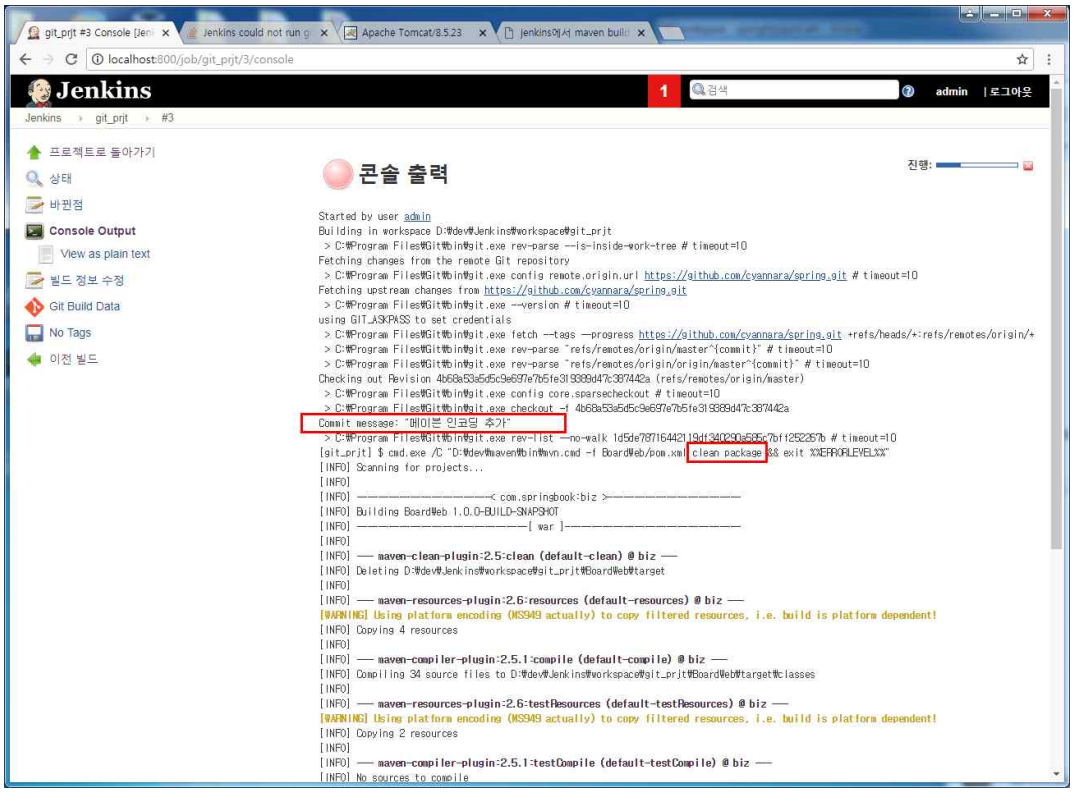
2) encoding 오류가 나는 경우

```
<build>
  <plugins>

    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.5.1</version>
      <configuration>
          <source>1.6</source>
          <target>1.6</target>
          <encoding>utf-8</encoding>
          <compilerArgument>-Xlint:all</compilerArgument>
          <showWarnings>true</showWarnings>
          <showDeprecation>true</showDeprecation>
      </configuration>
    </plugin>
  </plugins>
</build>
```

pom.xml에서 compiler-plugin 설정에 encoding 설정 추가





참고사이트: https://dukeom.wordpress.com/2017/03/20/jenkinsgithubmaven-으로-빌드배포하기-24/