

CI/CD 환경 구축 - Jenkins를 이용한 빌드 자동배포

MSBuild + SVN

사설

- Jenkins 빌드 세팅을 진행하게된 이유?
 - 기존 Jenkins 빌드서버의 노후화 (os, 젠킨스, 플러그인, 기타 툴등의 버전 노후화)로 인하여 기존 Jenkins 빌드서버를 폐기하고 새로운 빌드서버 세팅이 필요해서 해당작업 진행
-

참고 사이트

- <https://www.lesstif.com/continuous-integration/>
-

1. CI / CD란?

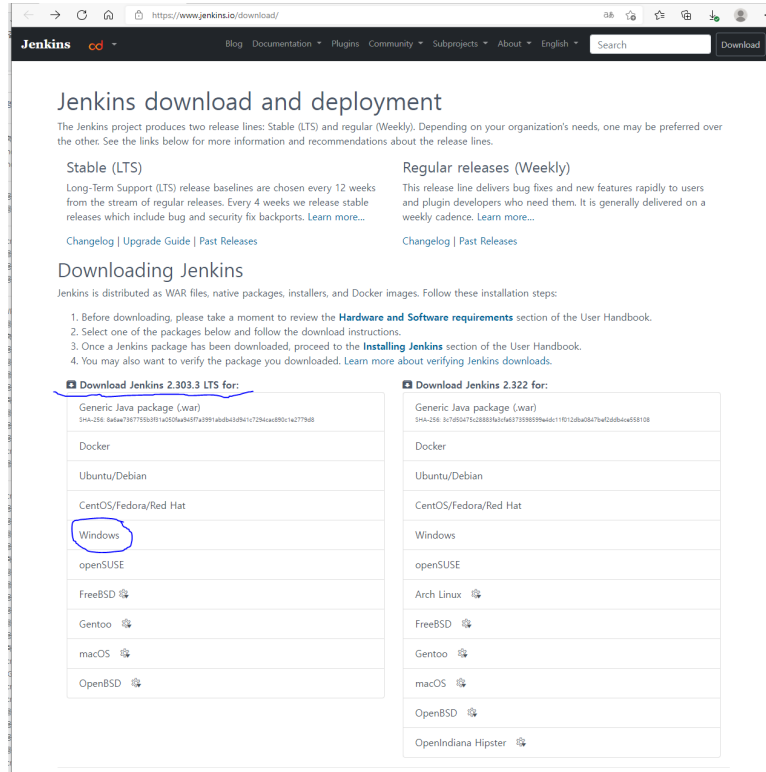
- 개발환경을 사용자가 이용가능한 서비스로 전달하는 모든 과정을 지속 가능한 형태로 만들어 개발자와 사용자간의 격차를 줄이는 것 => 코드를 빌드, 테스트 하고 배포하는 활동이 있다.
-

2. Jenkins 사용 이유

- 1. 기존에 빌드 자동화 배포 툴로 Jenkins를 사용 중이었음
 - 2. Jenkins의 경우 무료 소프트웨어
 - 3. 다양한 플러그인 설치를 통해 여러 기능을 사용할 수 있다
 - 4. 환경 구축이 용이하다
-

3. Jenkins

- 1. Windows 환경에서 구축할 것이므로 <https://www.jenkins.io/download/> 에서 LTS(안정적인 버전) Windows 선택하여 젠킨스 설치



- 2. 젠킨스 설치가 완료되면 인터넷 주소창에 **localhost:8080** 을 입력하여 젠킨스 페이지로 이동 (* 젠킨스 기본포트는 8080이며 변경하여 사용할 경우 localhost:port 에 맞는 port를 입력)
- 3. **localhost:8080** 이동 시 아래와 같은 페이지가 노출되며 **빨간색 음영처리**된 경로에서 패스워드를 복사하여 **Administrator password**에 기입한다

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

`C:\ProgramData\Jenkins\jenkins\secrets\initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

- 4. **Skip Plugin Installations** 를 클릭하여 넘어간다. 젠킨스 플러그인의 경우 추후 설치한다.

Offline

Offline

This Jenkins instance appears to be offline.

For information about installing Jenkins without an internet connection, see the [Offline Jenkins Installation Documentation](#).

You may choose to continue by configuring a proxy or skipping plugin installation.

[Configure Proxy](#)
[Skip Plugin Installations](#)

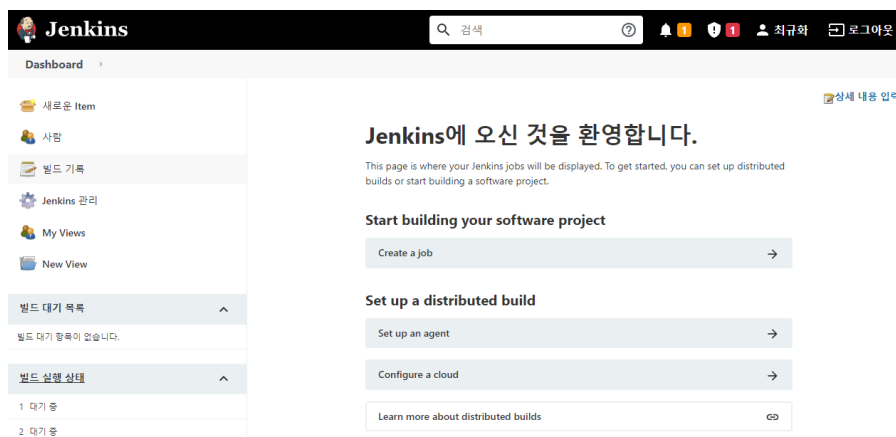
- 5. 젠킨스 로그인시 사용할 계정을 세팅한다. 팀에서 사용할 경우 해당 계정을 공유해서 젠킨스에 접속가능

Getting Started

Create First Admin User

계정명: 암호: 암호 확인: 이름:

- 6. 추후 젠킨스 접속주소를 세팅하고 젠킨스를 시작하면 아래와 같이 메인화면이 노출된다
 젠킨스 접속주소가 localhost:8080으로 되어있는 경우 변경할 수 있으며 젠킨스가 설치된 빌드서버의 주소로 변경한다
 추가적으로 젠킨스 홈 디렉토리(<http://localhost:8080/configure>)의 경로가 실제 젠킨스가 설치된 경로와 다른 경우 변경한다



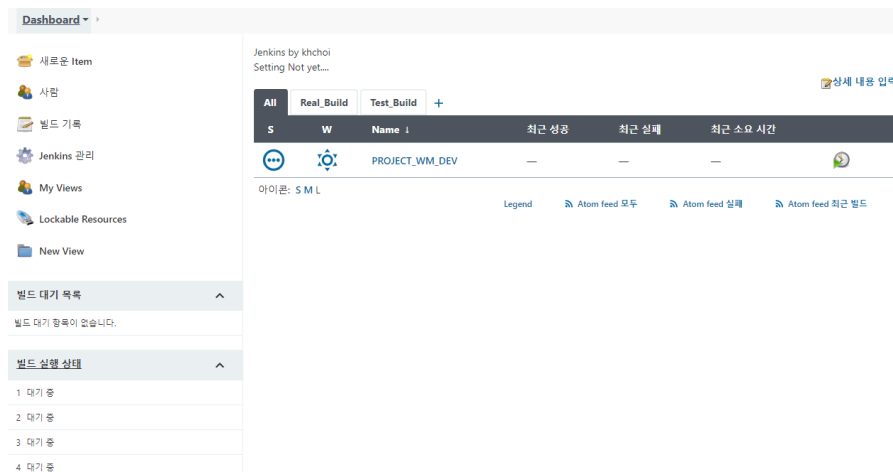
- [젠킨스 접속주소 변경 <http://localhost:8080/configure>]
 Dashboard -> Jenkins관리 -> 시스템 설정 -> Jenkins Location 항목에서 Jenkins URL에 빌드서버 주소 및 포트를 기입한다 (ex:http://10.101.xx.xx:8080/)
- [젠킨스 홈 디렉토리 변경 <http://localhost:8080/configure>]
 실제 젠킨스가 설치된 위치(<C:\Program Files\Jenkins>)에서 [jenkins.xml](#)의 편집으로 들어가

JENKINS_HOME의 value 값을 실제 젠킨스가 설치된 경로로 변경 및 젠킨스 restart 진행 Dashboard
-> Jenkins관리 -> 시스템 설정 -> 홈 디렉토리 항목에서 경로가 바뀌었는지 확인

(* 참고)

젠킨스 restart: (젠킨스 주소)/resart 진행 ex) <http://localhost:8080/restart>

- 7. 젠킨스 뷰 생성
 - 아래와 같이 ALL 버튼 옆의 + 버튼을 통해서 빌드 대상의 사용성격에 따른 뷰 생성이 가능
(ex:Real_Build 뷰 => 실제 배포용 빌드에 사용되는 대상 노출) Test_Build 뷰 => 테스트에 사용되는 대상 노출



4. Jenkins Plugins

- Jenkins에서 특정 기능을 사용하기 위해서는 관련된 플러그인 설치가 필요하다(MSBuild, SVN 등...)
- <http://localhost:8080/pluginManager/> 에서 현재 설치된 플러그인목록, 업데이트 가능한 플러그인 목록 등을 볼 수 있다.
- 플러그인이 설치되는 위치는 <%ProgramFiles%\Jenkins\plugins>
- 플러그인이 설치된 이후에 젠킨스에 반영하기 위해서는 젠킨스를 restart 해야한다
- [플러그인 오프라인 설치]
 - <https://plugins.jenkins.io/> 페이지에서 설치하고자 하는 플러그인 검색 및 설치
 - 설치된 항목을 개발망PC로 가져와 젠킨스 플러그인이 설치된 위치에 저장
 - 젠킨스 restart 진행
 - <http://localhost:8080/pluginManager/installed> (Dashboard -> Jenkins관리 -> 플러그인관리 -> 설치된 플러그인 목록)에 해당 플러그인이 설치되었는지 확인
- [플러그인 온라인 설치]
 - <http://localhost:8080/pluginManager/available> (Dashboard -> Jenkins관리 -> 플러그인관리 -> 설치가능)페이지에서 설치하고자 하는 플러그인 검색 및 설치

- [플러그인 설치 대상]
 1. Subversion
 - Subversion을 사용하기 위해서 필요
 2. MSBuild
 - Visual Studio를 MSBuild로 빌드하기 위해서 필요
 - 설치 시, 구성 > Build > Build a Visual Studio project or solution using MSBuild 항목 생성
 3. Credentials
 - SVN 레퍼지토리 접속에 사용되는 계정 세팅을 위해서 필요
 4. Nuget
 - 빌드 타겟이되는 솔루션에서 Nuget 패키지를 사용하는 경우, Nuget 패키지 복원을 위해서 설치
 5. Pipeline
 - 젠킨스 파이프라인을 설치하여 Job들을 순차 or 병렬로 실행, 스크립트로 이벤트를 연속적으로 실행시키는 기능을 지원

5. Jenkins 빌드를 위한 세팅

1) Global Tool Configuration

- 전역으로 사용되는 옵션 세팅 (ex:JDK 버전, MSBuild 옵션)

1. JDK

- JAVE_HOME 위치 및 이름 설정
- Install automatically 설정 시 실행하는 job마다 사용되는 Java 버전이 다른 경우 수동으로 설치하지 않아도된다. **오라클 Java의 경우 일정버전이상 사용하기 위해서는 계정설정이 필요**

JDK

JDK installations

Add JDK

JDK	Name	JAVA_HOME	Install automatically
	JDK_1_8_0_202	C:\Program Files\Java\jdk1.8.0_202	<input type="checkbox"/>

Delete JDK

2. MSBuild

- MSBuild 이름, 사용할 MSBuild.exe 가 위치한 경로 설정
- VS2019 솔루션 빌드시 경로=> **C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\Bin**

MSBuild

MSBuild installations

[Add MSBuild](#)

MSBuild

Name
VS2019

Path to MSBuild ?

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\Bin

Default parameters ?

☐ Install automatically ?

[Delete MSBuild](#)

2) 시스템 설정

- 환경변수 및 경로 정보등을 설정하는 메뉴로 젠킨스 동작에 필요한 주요 설정

1. 홈 디렉토리

- 젠킨스가 설치된 경로 표시(그렇지 않은경우, 경로를 재설정하여 젠킨스 restart 필요)

홈 디렉토리 ?

C:\Program Files\Jenkins

2. 시스템 메시지

- 젠킨스 Dashboard에 표시되는 메시지
- 일반적으로 팀원들이 알아야할 공지사항이 있는 경우 기재

3. of executors

- 젠킨스가 동시에 빌드를 진행할 수 있는 스레드 갯수(기본값=2) 일반적으로 CPU 코어갯수보다 적게하여 세팅

of executors

4

4. Jenkins Location

- 젠킨스 주소 설정 일반적으로 젠킨스가 설치된 PC IP/설정된 PORT 사용

Jenkins Location

Jenkins URL ?

http://localhost:8080/

⚠ Please set a valid host name, instead of localhost

5. Quiet period

- 값이 설정되면 새로운 빌드가 있을 때 설정한 초만큼 대기한다.
- 버전관리로 cvs를 사용할 경우 필요, 동시에 수 많은 파일을 커밋할 경우 CVS의 특성상 전체 커밋이 완료되지 않더라도 업데이트가 가능하며 이런경우 빌드 실패가능
- 빌드가 많고 바쁜 젠킨스 서버일 경우 해당 값을 늘려 빌드의 수를 조절할 수 있다

6. SCM checkout retry count

- 체크아웃 실패 시 재시도 여부(기본값=0), 0일 경우 재시도 하지않는다.

7. Subversion

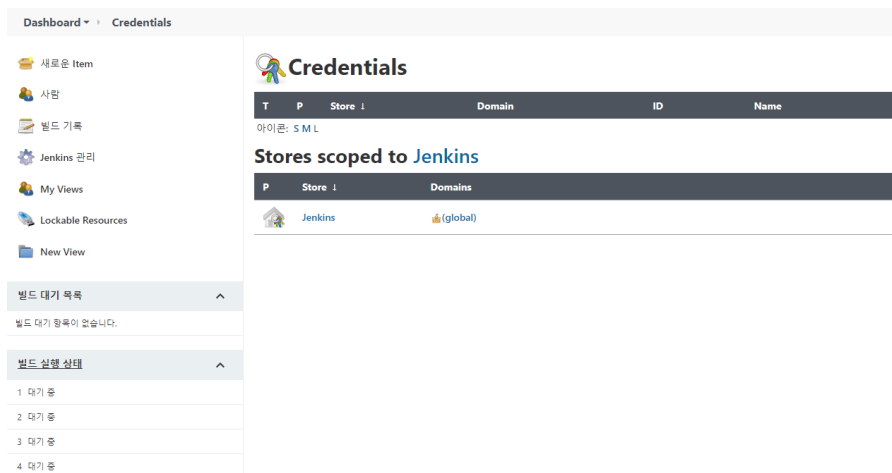
- SVN을 사용하여 빌드 시 세팅
- Subversion Workspace Version: 사용할 SVN 버전, 1.7이상 사용권장
- Exclusion revprop name: 여기에 프로퍼티 이름을 설정하여 특정 커밋은 빌드하지 않고 무시할 수 있다

3) 플러그인 관리

- Jenkins 기능 확장을 위해서 새로운 플러그인을 설치하거나 기존 플러그인을 업데이트하거나 설치된 플러그인 목록을 확인할 수 있다

4) Manage Credentials

- 외부 서비스에 로그인하기 위해서 필요한 정보를 입력하고 관리하는 메뉴
- Dashboard > Jenkins 관리 > Manage Credentials > global(Domains) > Add Credentials 로 계정추가 가능



- 아래와 같이 설정할 수 있다.

1. 유저이름 및 패스워드 사용
2. 전역으로 사용

3. 유저이름, 패스워드, 아이디, 설명 입력

Kind

Username with password

Scope

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

Username

khchoi

☐ Treat username as secret

Password

.....

ID

khchoi

Description

khchoi Credential

OK

6. Job 생성

- Job을 생성하여 빌드를 진행할 수 있다
- Dashboard > 새로운 Item > Freestyle project 를 진행하여 Job 생성

1) General

- 설명
 - 해당 Job에 대한 설명기재
- 오래된 빌드 삭제
 - 빌드 진행 시 이력이 남게되는데, 이를 유지할 기간을 설정한다
- 사용자 빌드 경로사용
 - 빌드하고자 하는 대상이 있는 경로 설정
- 표시이름
 - Dashboard에 표시되고자 하는 해당 job의 이름

2) 소스 코드 관리

- Repository URL
 - 빌드하고자 하는 대상의 SVN 경로 설정
- Credentials

- SVN 접근에 사용하고자 하는 계정 설정
- 해당 계정이 나타나지 않는 경우 **Manage Credentials**을 통해서 계정 생성후 진행
- Repository depth
 - 기본값인 infinity로 설정

3) 빌드 유발

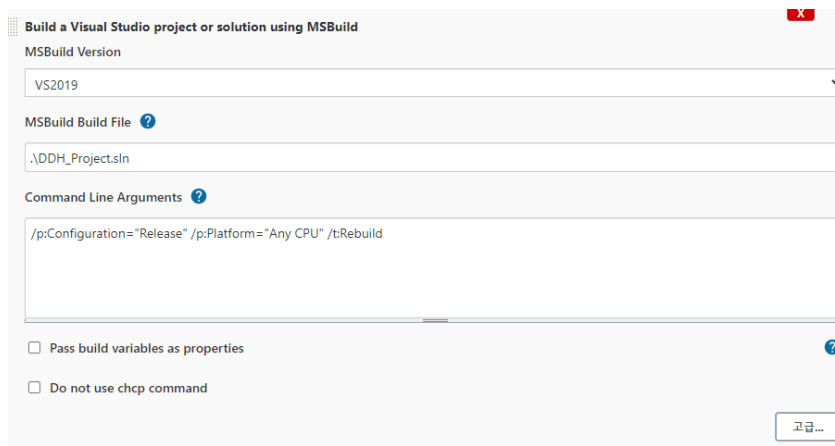
- 빌드를 원격으로 유발하거나 (Credential 토큰생성을 통해서 스크립트로 외부에서 빌드진행 가능) 빌드를 주기적으로 진행하거나 등에 대한 설정 사용 시 세팅

4) 빌드 환경

5) Build

- 빌드 진행에 사용되는 MSBuild, Command, shell, trigger 등을 설정
 - **Trigger** => 해당 job 진행 시 trigger에 설정된 다른 job도 같이 빌드진행 (ex:서드파티 라이브러리 등을 설정하여 같이 빌드가 진행될 수 있도록 할 수 있다)

- **MSBuild** => **Add build step** 버튼에서 **Build a Visual Studio project or solution using MSBuild** 선택 => **Global Tool Configuration**에서 설정한 MSBuild 옵션 사용
 - MSBuild Version => **Global Tool Configuration**에서 세팅한 "VS2019" 선택
 - MSBuild Build File => 빌드가 진행될 솔루션 지정
 - Command Line Arguments => 빌드 옵션 설정



The screenshot shows the Jenkins configuration page for building a Visual Studio project using MSBuild. The title is "Build a Visual Studio project or solution using MSBuild". The "MSBuild Version" is set to "VS2019". The "MSBuild Build File" is ".\DDH_Project.sln". The "Command Line Arguments" field contains the command: `/p:Configuration="Release" /p:Platform="Any CPU" /t:Rebuild`. There are two checkboxes: "Pass build variables as properties" (unchecked) and "Do not use chcp command" (unchecked). A "고급..." (Advanced...) button is at the bottom right.

- **command => Add build step** 버튼에서 **Excute Windows batch command** 선택 => 아래와 같이 빌드진행 시 실행될 배치파일을 기입하여 실행되도록 세팅



The screenshot shows the Jenkins configuration page for executing a Windows batch command. The title is "Execute Windows batch command". The "Command" field contains the command: `D:\visual studio 2019\BatFile.cmd %JOB_NAME% %SYN_VERSION%`. A link "See the list of available environment variables" is provided below the command field. A "고급..." (Advanced...) button is at the bottom right.