

Jenkins

- description : jenkins 관련 내용 기술
- author : 도봉산핵주먹
- email : hylee@repia.com
- lastupdate : 2020-06-10

Jenkins

Jenkins란?

- 젠킨스(Jenkins)는 소프트웨어 개발 시 지속적 통합(CI : continuous integration) 서비스를 제공하는 툴이다.
- 다수의 개발자들이 하나의 프로그램을 개발할 때 버전 충돌을 방지하기 위해 각자 작업한 내용을 공유 영역에 있는 Git등의 저장소에 빈번히 업로드함으로써 지속적 통합이 가능하도록 해 준다.
- 젠킨스는 오라클과의 분쟁 이후 허드슨으로부터 분기되었다.
- [CI/CD](#) 도구 플러그인으로 정적분석도구(SpotBugs, ...)를 추가할 수 있음

출처

- [위키백과](#)

설치

1. 다운로드

Jenkins download

아래 표시된 Past Releases 로 들어갑니다.

제일 높은 버전을 받으면 됩니다.

Tip : ex) 2.50, 2.100 이 있으면 2.100이 최신입니다. (.뒤에 숫자가 높은게 최신)

Tip : jenkins는 2.54버전 부터 자바 8을 사용해야 합니다.



2. 구동 및 화면 띄우기

.sh 파일을 만들어줍니다..

아래 code에 보면 알다싶이 #{jenkins 경로}/log 폴더를 만들어줘야 합니다.

```
#jenkins.sh
#!/bin/bash
JAVA_HOME=#{java 경로}
HTTP_PORT=9090
JENKINS_HOME=#{jenkins 경로}

export JAVA_HOME HTTP_PORT

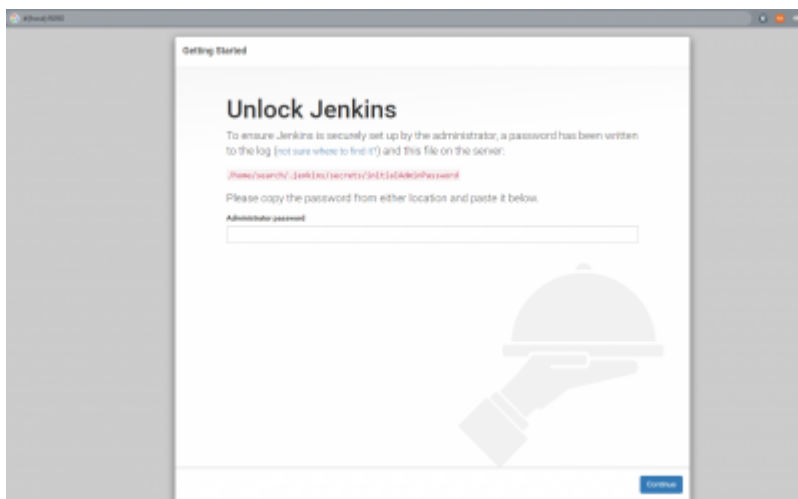
nohup $JAVA_HOME/bin/java -jar $JENKINS_HOME/jenkins.war --
httpPort=$HTTP_PORT --sessionTimeout=120 -XX:+AggressiveOpts >>
./logs/jenkins.log 2>&1 &
```

jenkins.sh 파일을 만든 후 port 방화벽을 풀고 실행해 줍니다.

```
[search@localhost jenkins]$ sudo firewall-cmd --zone=public --add-
port=9090/tcp --permanent
success
[search@localhost jenkins]$ sudo firewall-cmd --reload
success
[search@localhost jenkins]$ sudo firewall-cmd --list-ports
22/tcp 3306/tcp 8080/tcp 80/tcp 9090/tcp

[search@localhost jenkins]$ ./jenkins.sh
```

그럼 아래와 같이 나옵니다.



3. 로그인

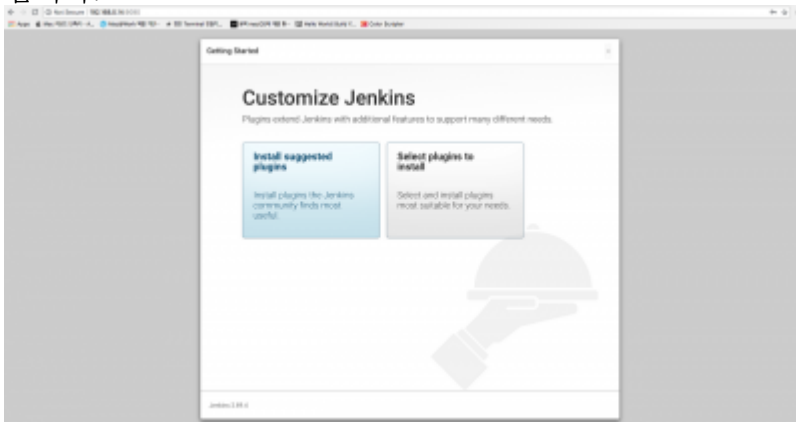
위 화면을 보면 /home/search/.jenkins/secrets/initialAdminPassword 이런 문구가 있습니다.

이건 search 계정으로 ./jenkins.sh를 실행해서 사용자가 search가 된거고 그 경로안에 패스워드가 있다

는 겁니다.

```
vi /home/search/.jenkins/secrets/initialAdminPassword
```

이렇게 치면 한줄로된 문자열이있는데 그걸 복사해서 Administrator password에 붙여넣고 로그인하면 됩니다.



플러그인 설치화면이 나옵니다.

그럼 **suggested**를 클릭하면 플러그인을 설치합니다.

Tip : 플러그인 설치화면에서 다 false이 되면 jenkins 낮은 버전을 받은것입니다.

4. 설정 (svn + war배포)



Jenkins

Jenkins >

-  새로운 Item
-  사람
-  빌드 기록
-  Jenkins 관리
-  Credentials
-  Terminal
-  New View

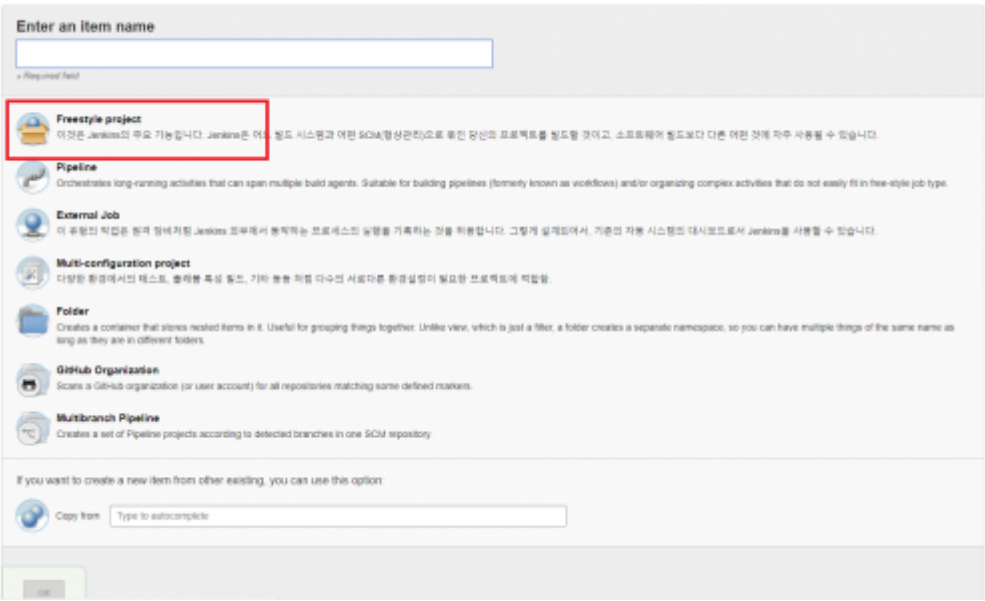
빌드 대기 목록

빌드 대기 항목이 없습니다.

빌드 실행 상태

- 1 대기 중
- 2 대기 중

로그인을 한다음 왼쪽 위에 새로운 Item을 클릭 합니다.



Enter an item name

* Required field

 **Freestyle project**
이것은 Jenkins의 주요 기능입니다. Jenkins은 이 빌드 시스템과 어떤 SCM(항상관리)으로 묶인 장인의 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.

 **External Job**
이 유형의 작업은 원격 장에 있는 Jenkins 코우에서 통치하는 프로세스의 실행을 기록하는 것을 허용합니다. 그렇게 설계되어서, 기존의 자동 시스템의 대시보드에서 Jenkins를 사용할 수 있습니다.

 **Multi-configuration project**
다양한 환경에서의 테스트, 실행용 특성, 기타 응용 작업 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.

If you want to create a new item from other existing, you can use this option:

 Copy from:

Item name을 입력한 다음 Freestyle project를 클릭합니다.

General 소스 코드 관리 빌드 방법 빌드 환경 Build 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: ☒

Add module...

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:

그림과 같이 네모 설정해주면 됩니다.

Credentials은 계정정보이며 Add를 클릭해서 계정정보를 입력해주면 됩니다.

Tip : Repository URL은 현재 우리 Apps에 대한 Url (eclips에 설정되어 있는 Url)이며 꼭 /trunk까지 지정해야 됩니다.

여기까지 저장하고 Jenkins > Jenkins 관리 > Global Tool Configuration 로 들어가서 maven 설정을 먼저 해줍니다.

Jenkins > Global Tool Configuration

Maven

Maven installations

Maven

Name:

☒ Install automatically

☐ Install from Apache

Version:

그 후에 다시 설정하고 있던 Item으로 돌아 갑니다.

Build

Invoke top-level Maven targets

Maven Version:

Goals:

Maven Version : 방금 위에서 설정한 값이 나오면 선택해 주면 됩니다.

Goals : clean install 을 입력해줍니다.

5. 빌드 후 조치

5-1. jenkins 설치 서버내에 배포

4.번까지 완료후에 아래 Add build step을 클릭해서 Execute shell을 클릭하면 Command가 나오는데

이 셋팅은 jenkins를 설치한 서버에 .war를 풀어 배포를 하는 것이라면 추천합니다.

평소에 .war를 풀던 .sh 파일을 실행만 시키면 됩니다.

5-2. jenkins 설치 서버외에 배포

5.1과는 달리 jenkins서버 말고 다른서버나 여러 서버를 설정해서 다발적으로 또는 효율적으로 쓰려면 빌드 후 조치에서 Send build artifacts over SSH 를 선택해야 합니다.

Send build artifacts over SSH가 안보인다면 Publish Over SSH Plugin를 설치해야 합니다.

Plugin이 설치가 되었다면 Jenkins > 환경설정에 들어가서 맨 하단에 SSH Servers가 생겼을 겁니다. 그럼 아래 사진과 같이 설정해 주면됩니다.(Apps 기준 설정)

그 후에 다시 설정하고 있던 Item으로 돌아 갑니다.

Remove prefix : 설정 안해주면 ROOT 경로에 ROOT/trunk/target/apps-1.0.0.war가 복사됩니다.
Exec command : 설정한 대로 빌드하고 끝나면 bash에 사용할 명령어를 쓰면 됩니다.

Tip

Troubleshooting

Ref

- [CentOS 7 Jenkins 설치 :: 마이자몽](#)
- [Jenkins 설치 가이드](#)
- [CentOS 6 Jenkins로 SVN + Spring Maven 프로젝트 자동 빌드 배포 셋팅](#)
- [CI - 젠킨스 원격지 배포하기\(SSH\)](#)
- [젠킨스\(Jenkins\)에서 원격\(Remote\)으로 배포하기](#)
- [젠킨스 빌드 도중 Cannot run program "mvn" error=2, No such file or directory](#)

[도봉산핵주먹, jenkins](#)

From:

<http://125.132.25.164/dokuwiki/> - 문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12

Permanent link:

<http://125.132.25.164/dokuwiki/doku.php?id=wiki:3party:jenkins&rev=1671409007>

Last update: **2022/12/19 09:16**

