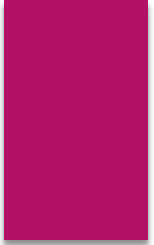


# 게임 프로젝트 (휴디) Jenkins + Github 사용법

NHN NEXT 이선희

- 
1. 젠킨스 소개
  2. 젠킨스 설치방법
  3. Github와 연동방법
  4. MSBuild 사용하기
  5. 빌드하기
  6. 빌드 예약하기
  7. 젠킨스 계정 관리
  8. 다른 서버에 옮기기

# 젠킨스 소개

- ▶ 젠킨스란?
  - ▶ 자동배포 웹 어플리케이션
  - ▶ 클릭만으로 빌드, 배포가 가능하다
  - ▶ 빌드 예약을 통하여 모두가 잠든 새벽 시간에 자동 빌드하여 다음 날 아침에 확인해 볼 수 있다.
  - ▶ 빌드 시간이 오래걸리는 대규모 프로젝트에 적합하다

# 젠킨스 설치방법 - 1

- ▶ 젠킨스 공식 홈페이지 (<https://jenkins-ci.org/>)



클릭하여 윈도우 용을 설치한다.

# 젠킨스 설치방법 - 2

- ▶ 다운로드한 젠킨스 압축파일을 압축해제한다.

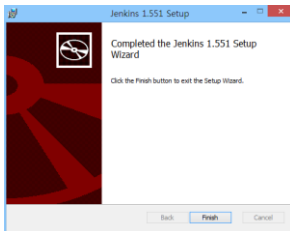


- ▶ setup.exe 파일을 실행하여 설치한다.

jenkins-1.551.msi



- ▶ next를 계속 누른다.
- ▶ 설치 끝



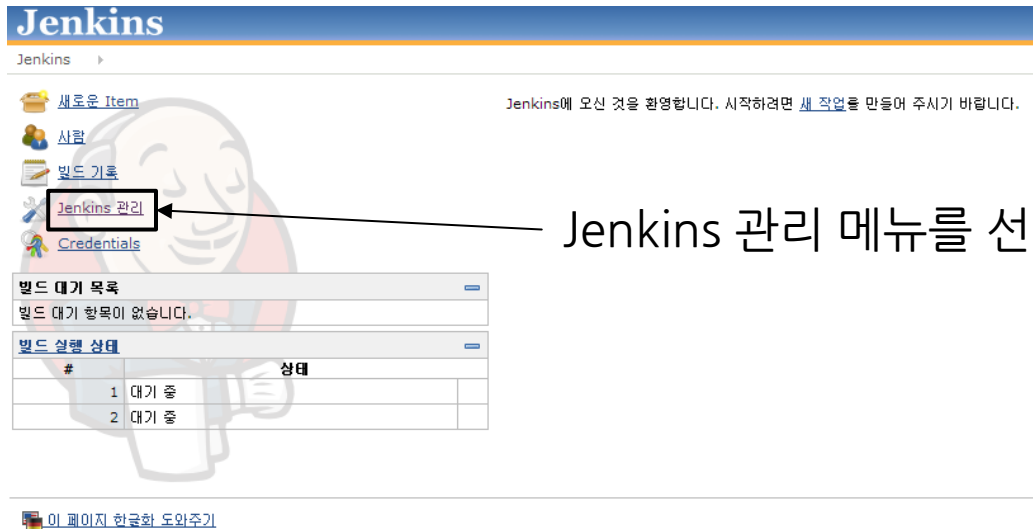
# Github와 연동방법 - 1

- ▶ 젠킨스를 설치하면 다음과 같은 화면이 등장한다.



- ▶ 위와 같은 화면이 이미 톰캣 서버 등 여러가지 이유로 8080포트를 사용하고 있어서 나오지 않는다면 (<http://jinstudio.tistory.com/16>) 이 링크를 참조하여 사용 포트를 바꿔주자
- ▶ 당연히 Windows용 GIT을 미리 설치해두자

# Github와 연동방법 - 2



The screenshot shows the Jenkins web interface. At the top is the 'Jenkins' header. Below it, a navigation sidebar on the left contains icons and labels for '새로운 Item', '사람', '빌드 기록', 'Jenkins 관리' (highlighted with a red box and a callout arrow), and 'Credentials'. The main content area displays a message: 'Jenkins에 오신 것을 환영합니다. 시작하려면 새 작업을 만들어 주시기 바랍니다.' Below this, there are sections for '빌드 대기 목록' (Build Queue) and '빌드 실행 상태' (Build Execution Status). The '빌드 대기 목록' section shows a table with two rows, both labeled '대기 중' (Waiting). The '빌드 실행 상태' section shows a table with two rows, both labeled '대기 중' (Waiting). At the bottom left, there is a link for '이 페이지 한글화 도와주기' (Help with Korean translation of this page).

#	상태
1	대기 중
2	대기 중

Jenkins 관리 메뉴를 선택한다.

# Github와 연동방법 - 3

## Jenkins 관리

⚠ Unsecured Jenkins allows anyone on the network to launch process



[시스템 설정](#)  
환경변수 및 경계 정보를 설정합니다.



[Configure Global Security](#)  
Secure Jenkins; define who is allowed to access/use the system.



[Reload Configuration from Disk](#)  
Discard all the loaded data in memory and reload everything from disk.



[플러그인 관리](#)  
Jenkins의 기능을 확장하기 위한 플러그인을 추가, 제거, 사용, 미사용으로

플러그인 관리 메뉴를 선택한다



[시스템 정보](#)  
문제 해결을 돕기 위한 다양한 환경 정보를 보여줍니다.



[System Log](#)  
System log captures output from java.util, logging output related to Jenkins.



[부하 통계](#)  
Check your resource utilization and see if you need more compute resources.



[Jenkins CLI](#)  
Access/manage Jenkins from your shell, or from your script.



[Script Console](#)  
Executes arbitrary script for administration/trouble-shooting/diagnosis.



[노드 관리](#)  
Add, remove, control and monitor the various nodes that Jenkins uses.



[Manage Credentials](#)  
Create/delete/modify the credentials that can be used by Jenkins.



[About Jenkins](#)  
See the version and license information.



[Manage Old Data](#)  
Scrub configuration files to remove remnants from old plugins and builds.



[실행 중지](#)  
Stops executing new builds, so that the system can be eventually updated.



# Github와 연동방법 - 4

- ▶ 설치 가능 탭에서 필터를 통하여 Github Oauth Plugin, Github Plugin, Git Plugin을 설치한다.

업데이트된 플러그인 목록

설치 가능

설치된 플러그인 목록

고급

선택

이름

버전

Artifact Uploaders

☐

[Appaloosa Plugin](#)

Publish your mobile applications (Android, iOS, ...) to the [appaloosa-store.com](#) platform.

☐

[ArtifactDeployer Plugin](#)

This plugin makes it possible to copy artifacts to remote locations.

☐

[Artifact Plugin](#)

This plugin allows deploying Maven 2, Maven 3, Ivy and Gradle artifacts and build info to the Artifactory artifacts manager.

1.4.0

0.28

2.2.1

- ▶ 설치 가능 목록이 나오지 않았을 경우 고급 탭에서 지금 확인 버튼을 누른다. 확인이 되지 않을 경우 방화벽 문제일 가능성이 크다.

변경정보 획득시기 : 2 min 34 sec 전 [지금 확인](#)

# Github와 연동방법 - 5

- ▶ 새로운 Item 메뉴를 클릭한다.



- ▶ Item 이름을 적고 리스트 중 가장 첫 번째 체크 박스를 클릭한 후 OK한다.

Item 이름

☒ **Build a free-style software 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.

# Github와 연동방법 - 6

- ▶ 다음과 같은 방법으로 작성한다.  
혹시 Repository URL에서 git이 존재하지 않는다고 한다면  
Jenkins 관리 -> 시스템 환경 -> Git 부분에서 Path를 확인해보자

The screenshot shows the Jenkins 'New Project' configuration page for a GitHub repository. The 'Name' field is set to 'Test'. The 'Description' field is empty. The 'GitHub project' field is set to 'https://github.com/kciter/test'. The 'Repository browser' dropdown is set to 'Git'. The 'Repository URL' field is set to 'https://github.com/kciter/test'. The 'Credentials' dropdown is set to 'none'. The 'Branches to build' field is set to '/master'. The 'Repository browser' dropdown is set to 'Git'. The 'Additional Behaviours' section is empty.

이름: Test

설명: [Raw HTML] [리보기]

☐ 오래된 빌드 삭제

GitHub project: https://github.com/kciter/test

☐ 이 빌드는 매개변수가 있습니다

☐ 빌드 안함 (프로젝트가 다시 빌드될 때 태깅된 새로운 빌드가 실행되지 않습니다.)

☐ 필요한 경우 concurrent 빌드 실행

관련 프로젝트 옵션

소스 코드 관리

☐ CVS

☐ CVS Projectset

☒ Git

Repositories

Repository URL: https://github.com/kciter/test

Credentials: none

Add Repository Delete Repository

Branches to build

Branch Specifier (blank for 'any'): \*/master

Add Branch Delete Branch

Repository browser: Git

Additional Behaviours: Add

# MSBuild 사용하기 - 1

- ▶ Jenkins에서 msbuild plugin을 설치한다

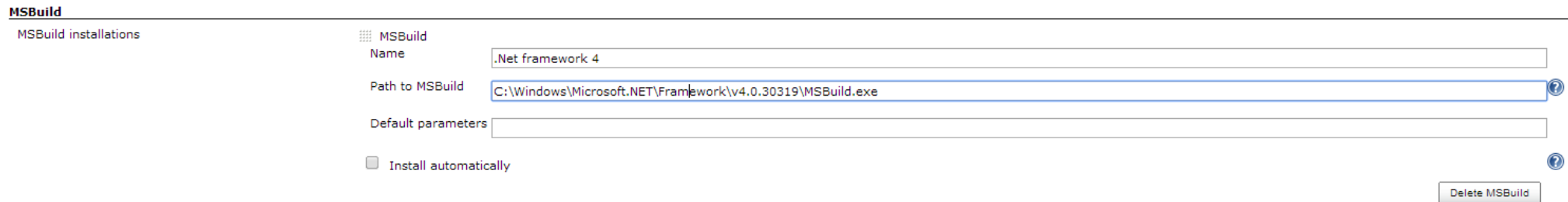


- ▶ Jenkins 관리 -> 시스템 설정 -> MSBuild 부분에서 Add MSBuild 클릭



# MSBuild 사용하기 - 2

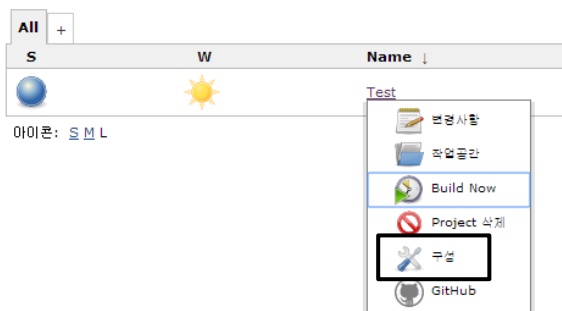
- ▶ 다음과 같이 MSBuild를 추가한다. 이름은 마음대로 정해도 상관없다.  
\* 환경에 따라 MSBuild의 경로가 다를 수 있습니다.



The screenshot shows the 'MSBuild' dialog box with the following fields and options:

- MSBuild Name:** .Net framework 4
- Path to MSBuild:** C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
- Default parameters:** (empty)
- Install automatically:** (unchecked checkbox)
- Delete MSBuild:** (button)

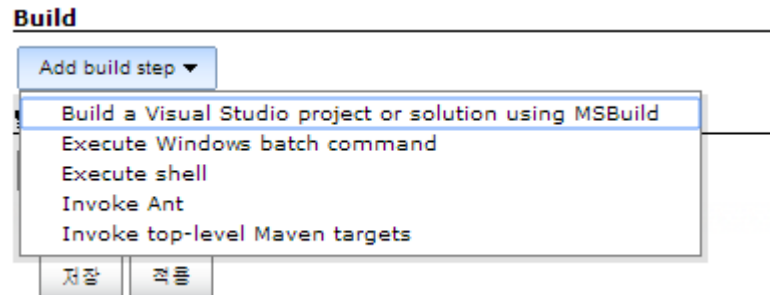
- ▶ 기존에 추가한 Item을 수정해야한다.



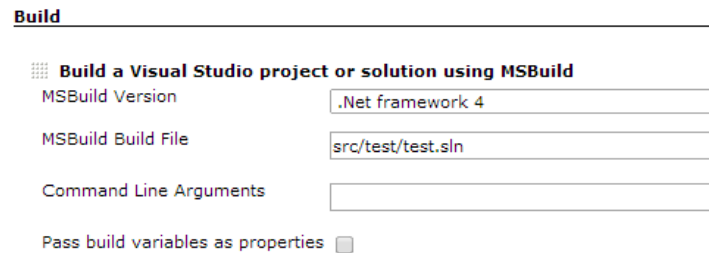
VS2013의 경우 경로는 C:\Program Files (x86)\MSBuild\12.0\Bin

# MSBuild 사용하기 - 3

- ▶ Build 부분에서 MSBuild로 빌드하도록 지정해준다.

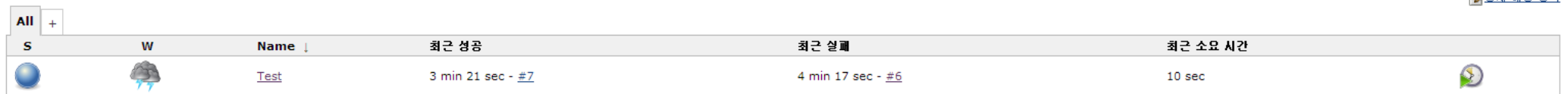





- ▶ 전에 만들어둔 MSBuild를 적용하고 빌드해야할 솔루션 파일의 경로를 적는다.



# 빌드하기


- ▶ 대시보드에서 가장 오른쪽 아이콘인 즉시 빌드하기를 클릭하면 빌드에 성공한다.



All	S	W	Name ↓	최근 성공	최근 실패	최근 소요 시간	
			Test	3 min 21 sec - <a href="#">#7</a>	4 min 17 sec - <a href="#">#6</a>	10 sec	

- ▶ 빌드된 내용을 commit하고 싶다면 빌드 후 조치에서 다음 항목을 설정한다.  
만약 Debug, Release, .exe 등이 ignor되어 있다면 커밋되지 않는다.

## 빌드 후 조치

 Set build status on GitHub commit

빌드 후 조치 추가 ▼

# 빌드 예약하기 - 1

- ▶ Build when a change is pushed to Github 체크박스를 선택하면 Github에 푸시 요청이 들어올경우 빌드한다.  
Poll SCM은 빌드 시간을 지정하여 그 시간에 빌드하도록 지정 할 수 있다.

☒ Build when a change is pushed to GitHub

☒ Poll SCM

Schedule



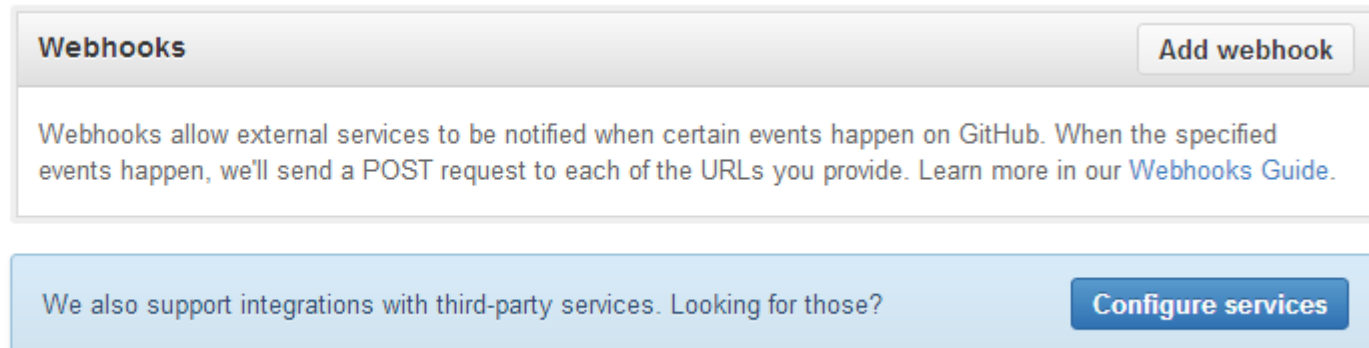
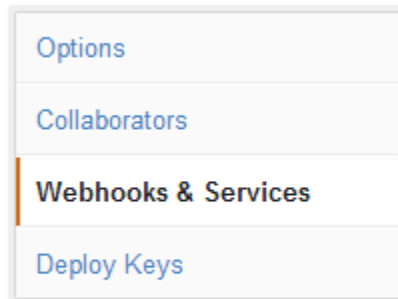
## 빌드 예약하기 - 2

- ▶ Build when a change is pushed to Github 기능을 사용하기 위해서는 github에서 webhook 기능을 사용할 수 있도록 설정해야한다.  
젠킨스와 연동된 프로젝트에서 세팅 메뉴를 클릭하자.



# 빌드 예약하기 - 3

- ▶ 다음 메뉴에서 Webhooks & Services 메뉴를 클릭한 후 Configure services 메뉴를 클릭한다.



# 빌드 예약하기 - 4

- Services 중 Jenkins (Github plugin) 항목을 클릭한 후 다음과 같이 설정한다.  
URL은 자신의 서버 주소 + /github-webhook/을 붙인다.  
ex) `http://example.com/github-webhook/`

**Services**

ActiveCollab

Acunote

AgileBench

AgileZen

AmazonSNS

Apiary

Apoio

AppHarbor

Apropos

Asana

AWS OpsWorks

Backlog

Bamboo

Bazaarvoice

Jenkins Hook Url

`http://10.73.45.140/github-webhook/`

☒ Active

Test Hook

Update settings

Jenkins is a popular continuous integration server.

Using the Jenkins GitHub Plugin you can automatically trigger build jobs when pushes are made to GitHub.

**Install Notes**

1. "Jenkins Hook Url" is the URL of your Jenkins server's webhook endpoint. For example:  
`http://ci.jenkins-ci.org/github-webhook/`.

For more information see <https://wiki.jenkins-ci.org/display/JENKINS/GitHub+plugin>.

# 빌드 예약하기 - 5

- ▶ 그 다음 젠킨스 설정에서 GitHub Web Hook 설정에서 Username과 Password를 설정해주면 푸시가 올 경우 자동으로 빌드를 해준다.

## GitHub Web Hook

☒ Let Jenkins auto-manage hook URLs

☐ Override Hook URL

### GitHub Credentials

API URL

Username

kciter

Password

OAuth token

Test Credential

Add 삭제

☐ Manually manage hook URLs

- ▶ 학교에서는 내부에서만 접속이 가능하기 때문에 GitHub Hook이 무시됨

# 빌드 예약하기 - 7

- ▶ Poll SCM의 표현방법은 다음과 같다.
  - ▶ 다섯 개의 문자열로 이루어져 있다.
  - ▶ 각 문자열은 한 칸의 공백으로 분리된다.
  - ▶ 왼쪽 문자열부터 차례대로 [분(minuth), 시(hour), 일(day), 월(month), 주(week)] 이다.
  - ▶ 여기서 week는 0~7로 나타낼 수 있고 0과 7은 일요일이다.

# 빌드 예약하기 - 8

- ▶ Poll SCM의 문법은 다음과 같다.
  - ▶ 여러 라인을 입력할 수 있다.
  - ▶ \* 은 모든 경우를 만족한다.
  - ▶ M-N은 (여기서 M, N은 정수) 값의 범위를 나타낸다.
  - ▶ M-N/X는 (여기서 M, N, X는 정수) M과 N사이의 스텝을 나타낸다.
    - ▶ ex) 0-10/2는 2, 4, 6, 8, 10
  - ▶ ,(콤마)를 이용해 여러 값을 선택할 수 있다.
  - ▶ H 심볼은 랜덤을 나타낸다.
    - ▶ ex) H 12 \* \* \* 의 경우 12시 0분에 실행될 수도 12시 30분에 실행될 수도 있다.
  - ▶ #이 맨 앞에 붙은 라인은 주석이다.

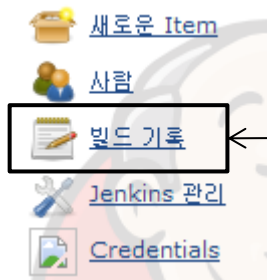
# 빌드 예약하기 - 9

## ▶ 예제

- ▶ 매일 15분 마다 빌드하기 [ex) :07, :22, :37, :52]
  - ▶ H/15 \* \* \* \*
- ▶ 매일 한 시간 중 30분 전 까지만 10분 마다 빌드하기 [ex) :04, :14, :24]
  - ▶ H(0-29)/10 \* \* \* \*
- ▶ 주말을 제외한 날에만 두 시간 마다 빌드하기 [ex) 10:38 AM, 12:38 PM, 2:39 PM...]
  - ▶ H 9-16/2 \* \* 1-5
- ▶ 12월을 제외하고 매월 1일, 15일에 빌드하기
  - ▶ H H 1,15 1-11 \*

# 빌드 확인하기 - 1

- ▶ 빌드 기록을 클릭한다.



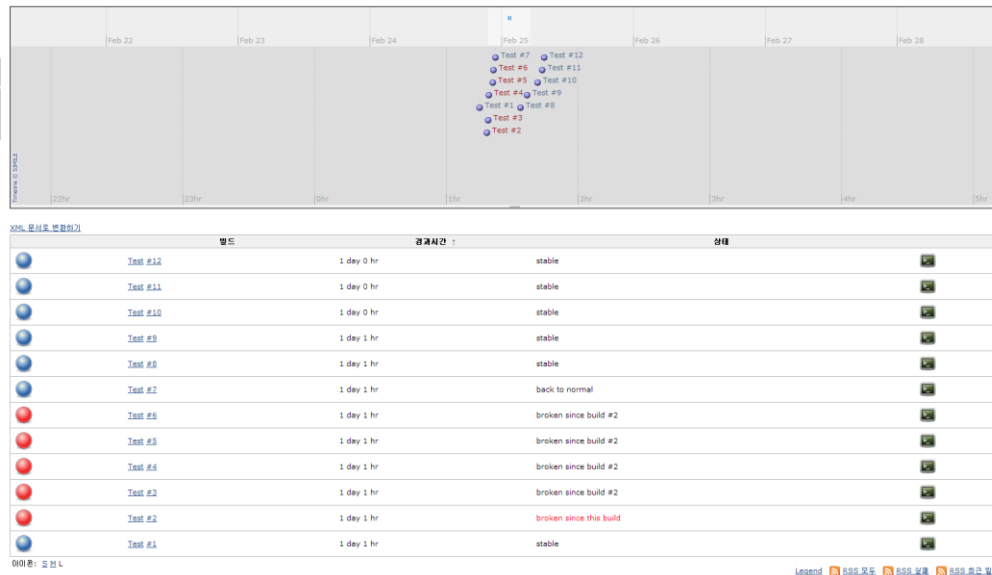
← 클릭



# 빌드 확인하기 - 2

- ▶ 드래그 혹은 마우스 휠을 이용하여 시간별 빌드 기록을 확인할 수 있다.  
맨 오른쪽 아이콘을 클릭하면 커맨드 창에 출력된 결과를 확인 할 수 있다.  
커맨드 창에 출력된 결과는 평소에 자주보던 Visual Studio의 빌드 기록과 같다.

Jenkins의 빌드 기록



# 젠킨스 계정 관리 - 1

- ▶ 젠킨스를 첫 설치한 후 Jenkins 관리 탭에 들어가면 다음과 같은 Warning이 표시된다.

## Jenkins 관리

⚠ Unsecured Jenkins allows anyone on the network to launch processes on your behalf. Consider at least enabling authentication to discourage misuse.

보안 설정 Dismiss

- ▶ 여기서 보안 설정을 클릭하거나 다음 메뉴를 클릭하면 다음과 같은 설정이 나타난다.



# 젠킨스 계정 관리 - 2

- ▶ 다음과 같이 설정하면 이제 회원가입/로그인해야만 이용할 수 있다.

☒ Jenkins' own user database

☒ 사용자의 가입 허용

## Authorization

☐ Anyone can do anything

☐ Legacy mode

☒ Logged-in users can do anything

☐ Matrix-based security

☐ Project-based Matrix Authorization Strategy

# 젠킨스 계정 관리 - 3

- ▶ 다음과 같이 계정별로 권한을 부여해 줄 수 있다.  
만약 권한부여가 안된 상태로 저장이 된다면 재설치해야 할 수 있다.

## Authorization

- ☐ Anyone can do anything
- ☐ Legacy mode
- ☐ Logged-in users can do anything
- ☒ Matrix-based security

User/group	Overall										Credentials					Slave					Job					View								
	Administer	Read	Run	Scripts	Upload	Plugins	Configure	Update	Center	Create	Update	View	Delete	Manage	Domains	Configure	Delete	Create	Disconnect	Connect	Build	Create	Delete	Configure	Read	Discover	Build	Workspace	Cancel	Create	Delete	Configure	Read	
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 kciter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

User/group to add:

- ☐ Project-based Matrix Authorization Strategy

# 다른 서버에 옮기기 - 1

- ▶ Publish Over FTP Plugin을 설치한다.

업데이트된 플러그인 목록		설치 가능	설치된 플러그인 목록	고급
설치 ↓				
<input type="checkbox"/>	<a href="#">FTP-Publisher Plugin</a>		This plugin can be used to upload project artifacts and whole directories to an ft	
<input type="checkbox"/>	<a href="#">Publish Over FTP Plugin</a>		Publish files over FTP	
<input type="checkbox"/>	<a href="#">Publish Over SSH Plugin</a>		Publish files and/or execute commands over SSH (SCP using SFTP)	
<input type="checkbox"/>	<a href="#">Over-the-Air Ad Hoc Deployment Plugin For iOS</a>		This plugin help you to build OTA installation packages for your XCode builds on	

# 다른 서버에 옮기기 - 2

- ▶ 업로드 해야하는 서버 설정을 한다.

## **Publish over FTP**

FTP Servers

FTP Server  
Name

game server

Hostname

10.73.45.140

Username

sunhyoup

Password

.....

Remote Directory

# 다른 서버에 옮기기 - 3

- ▶ 아까 설정한 서버를 선택한 후 옮겨야 하는 파일과 삭제할 폴더를 지정한다.
- ▶ Remote directory는 해당 FTP에서 만들어질 디렉토리를 설정한 후 그 안에 옮길 파일이 전송된다.

## 빌드 후 조치

### Send build artifacts over FTP

#### FTP Publishers

##### FTP Server

Name

#### Transfers

##### Transfer Set

###### Source files

###### Remove prefix

###### Remote directory

All of the transfer fields support substitution of [Jenkins environment variables](#)

###### Exclude files

###### Pattern separator

]+

No default excludes ☒

Make empty dirs ☐

Flatten files ☐

Remote directory is a date format ☒

Clean remote ☐



ASCII mode ☐

Add Transfer Set

# 다른 서버에 옮기기 - 4

- ▶ 빌드를 한 후 서버를 확인해보면 잘 생성되어있다.

/21\_2014-55-10/의 색인

이름	크기	수정된 날짜
 [상위 디렉토리]		
 test.exe	30.5 kB	14. 3. 10. 오전 11:52:00



# TODO

- ▶ 내일 정오까지 제출
  - ▶ 빌드 상태를 확인 할 수 있는 Jenkins URL
  - ▶ 빌드 파일을 받아볼 수 있는 FTP URL
    - ▶ Anonymous로 받을 수 있어야 함

