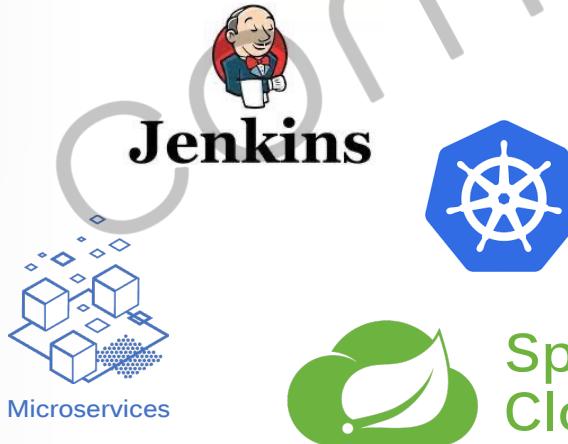


# JenKins 를 이용한 CI/CD Pipeline 구축



```
public static void main(String[] args)
{
    < servlet >
        < servlet-name >BoardController</servlet-name>
        < servlet-class >com.joneconsulting.controller.BoardController</servlet-class>
        < init-param >
            < param-name >user_name</param-name>
            < param-value >Kenneth Lee</param-value>
        </init-param >
    </servlet >
```

```
    < servlet >
        < servlet-name >BookController</servlet-name>
        < servlet-class >com.joneconsulting.controller.BookController</servlet-class>
        < init-param >
            < param-name >self_title, price, author</param-name>
            < param-value >self_title
                self_price
                self_author = author
            </param-value>
        </init-param >
    </servlet >
```

```
    < servlet >
        < servlet-name >UIApplicationDelegate</servlet-name>
        < servlet-class >com.joneconsulting.delegate.UIApplicationDelegate</servlet-class>
        < init-param >
            < param-name >NEXT_INNOVATION_DELEGATE</param-name>
            < param-value >@Interface NextInnovationDelegate : NSObject</param-value>
        </init-param >
    </servlet >
```



# 프로필

Dowon Lee



지식공유자 인증

5452 ★ 4.8(420)

멘토링 활성



- 홈
- 강의
- 로드맵
- 수강후기
- 블로그

최신순 ▾

## 강의 (3)



Spring Cloud로 개발하는 마이크로서비스 애플리케이션(MSA)

Dowon Lee

★★★★★ (174)

학습중

+3000명 독점 할인중



Spring Boot를 이용한 RESTful Web Services 개발

Dowon Lee

★★★★★ (308)

학습중

+2700명 독점 할인중



웹 애플리케이션 개발을 위한 IntelliJ IDEA 설정

Dowon Lee

★★★★★ (228)

학습중

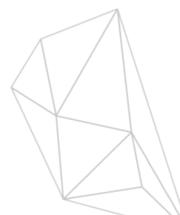
+3500명



## 목차

---

- Section 1: DevOps와 CI/CD
- Section 2: Jenkins를 이용한 CI/CD 사용
- Section 3: Jenkins + Infrastructure as Code
- Section 4: Jenkins + Ansible + Kubernetes 연동
- Section 5: Advanced Jenkins 사용
- Section 6: Public Cloud에 배포
- Appendix



## Section 2.

# Jenkins를 이용한 CI/CD 사용

- CI/CD를 위한 Git & Maven 연동
- CI/CD를 위한 Tomcat Server 연동
- PollSCM 설정
- Docker를 이용한 실습 환경 구성

# Setup Git plugin

njone company

- Manage Jenkins → Jenkins Plugins → available → github

The screenshot shows the Jenkins management interface. On the left, a sidebar lists various management options: '새로운 Item', '사람', '빌드 기록', **Jenkins 관리**, 'My Views', 'Lockable Resources', and 'New View'. The 'Jenkins 관리' option is selected. The main content area is titled 'Jenkins 관리' and 'System Configuration'. It contains three sections: '시스템 설정' (Configure build triggers and environments), 'Global Tool Configuration' (Configure tools, their locations and automatic installers), and '노드 관리' (Manage Jenkins nodes). A red dashed box highlights the 'Global Tool Configuration' section. Another red dashed box highlights the 'GitHub plugin' entry in the 'Available' section of the Jenkins Plugins page.

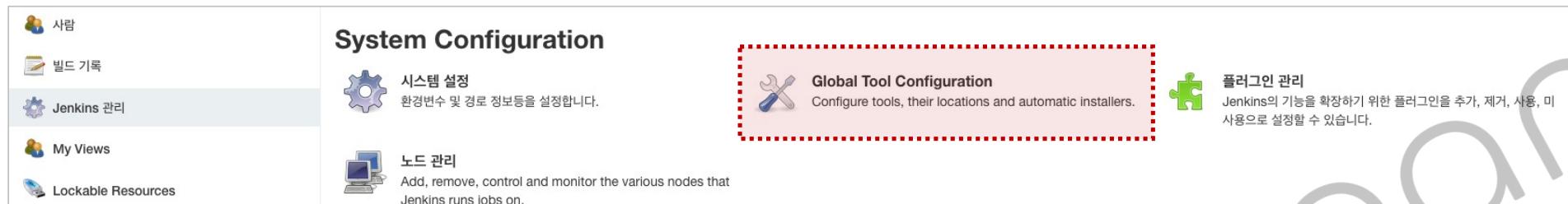
The screenshot shows the Jenkins Plugins page. It lists several available plugins, including 'GitHub plugin'. This plugin is described as integrating GitHub with Jenkins. A red dashed box highlights the 'GitHub plugin' entry. The page also shows the version '1.34.1' and a '설치' (Install) button.



# Setup Git plugin

njone company

- Manage Jenkins → Global Tool Configuration → git



The screenshot shows the Jenkins System Configuration page. On the left, there is a sidebar with icons for People, Build History, Jenkins Management (which is selected), My Views, and Lockable Resources. The main content area has three sections: 'System Configuration' (with a gear icon), 'Global Tool Configuration' (with a wrench and screwdriver icon, highlighted with a red dashed box), and 'Plugins' (with a puzzle piece icon). The 'Global Tool Configuration' section contains the text: 'Configure tools, their locations and automatic installers.'



The screenshot shows the Jenkins Git configuration page. It displays a single Git installation named 'Default'. The configuration includes:

- Name: Default
- Path to Git executable: git
- A checkbox for 'Install automatically' which is unchecked.

A red dashed box highlights the 'Git' section. At the bottom right, there is a red button labeled 'Delete Git'.

# Setup Maven plugin

njone company

- Manage Jenkins → Jenkins Plugins → available → maven

The screenshot shows the Jenkins Plugin Manager interface. A search bar at the top contains the text "maven". Below the search bar are four buttons: "업데이트된 플러그인 목록" (Updated Plugins List), "설치 가능" (Installable), "설치된 플러그인 목록" (Installed Plugins List), and "고급" (Advanced). The "설치 가능" button is highlighted. The main area displays a list of available plugins. One plugin, "Maven Integration", is highlighted with a red dashed box. It has a "Build Tools" badge. A description below the plugin reads: "This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...)." To the right of the plugin details are columns for "Version" (3.12) and "Released" (2 mo 18 days ago).

## 플러그인 설치/업그레이드 중

준비

- Checking internet connectivity
- Checking update center connectivity
- Success





# Setup Maven plugin

njone company

- Manage Jenkins → Global Tool Configuration → maven

**Maven**

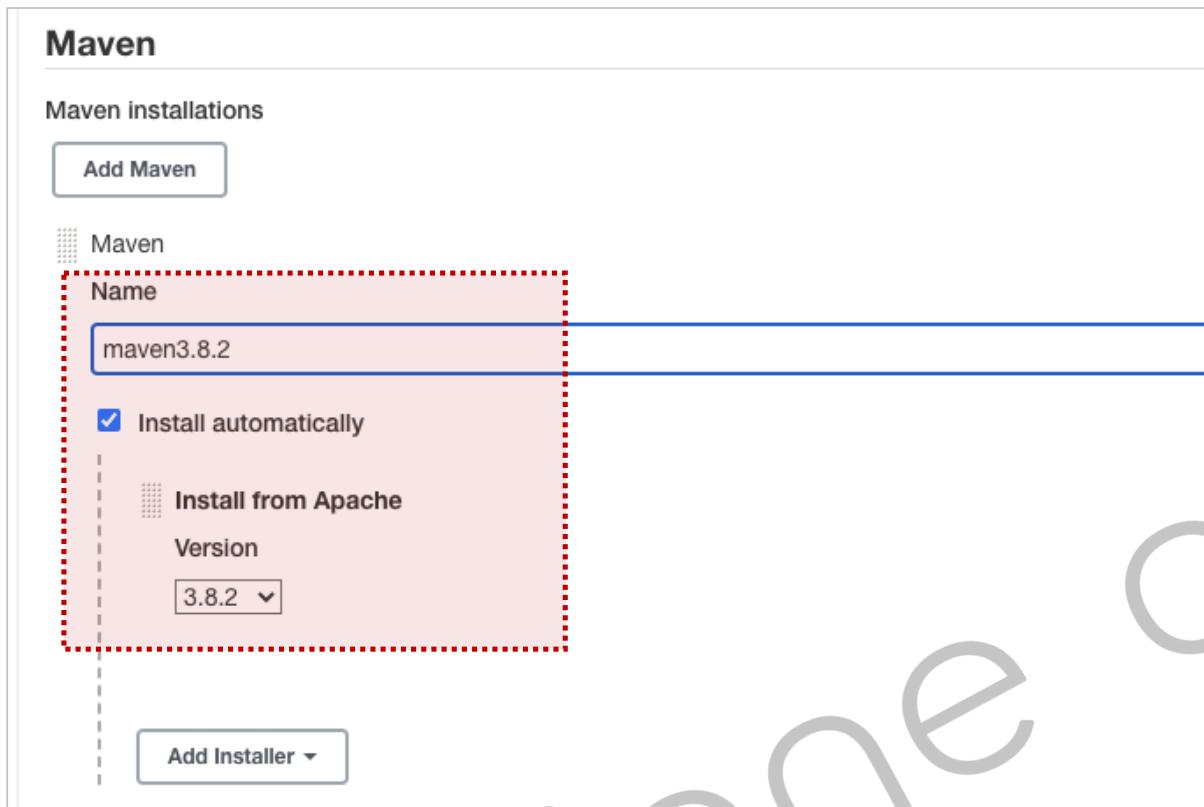
Maven installations

Add Maven

Maven

Name	maven3.8.2
<input checked="" type="checkbox"/> Install automatically	
Install from Apache	
Version	3.8.2 ▾

Add Installer ▾





# Exercise #2 JenKins Job 1/4

njone company

- Item name → My-Second-Project
  - *Maven project*

Enter an item name

» 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.

 **Maven project**  
Maven 프로젝트를 빌드합니다. Jenkins은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.

 **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.

# Exercise #2 JenKins Job 2/4

njone company

## ■ General

- ***My maven project build***

The screenshot shows the Jenkins General configuration page. The 'General' tab is selected. A red dashed box highlights the 'Name' field, which contains the text 'My maven project build'.

## ■ Source Code Management

- Repository URL

<https://github.com/joneconsulting/cicd-web-project>

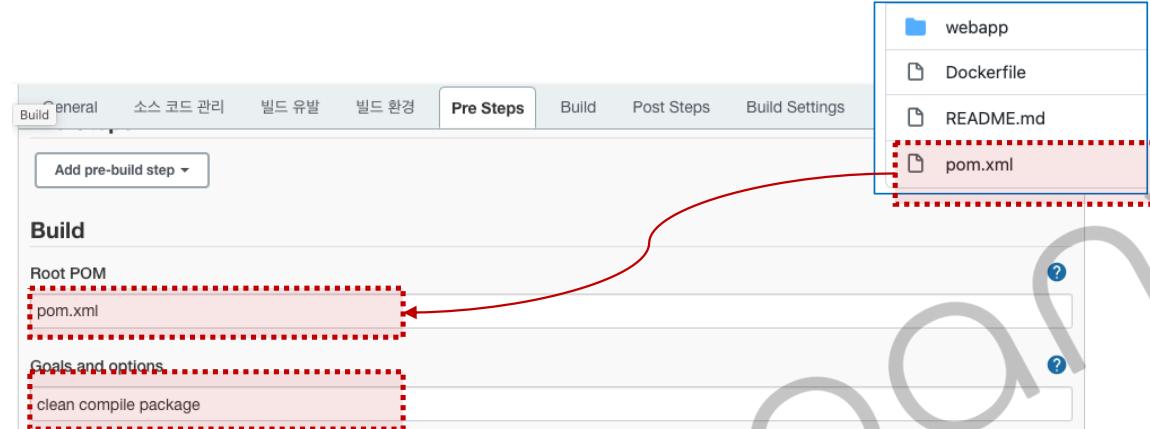
The screenshot shows the Jenkins Source Code Management configuration page for a Git repository. The 'Source Code Management' tab is selected. A red dashed box highlights the 'Repository URL' field, which contains the value 'https://github.com/joneconsulting/hello-world.git'. Below it, the 'Branches to build' section shows a 'Branch Specifier' field containing '/master'.

# Exercise #2 JenKins Job 3/4

njone company

## ■ Build

- Root POM: *pom.xml*
- Goals: *clean compile package*



## ■ Save > Build Now

```
[INFO] Maven Project ..... SUCCESS [ 7.118 s]
[INFO] Server ..... SUCCESS [ 50.442 s]
[INFO] Webapp ..... SUCCESS [ 7.227 s]
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:06 min
[INFO] Finished at: 2021-09-02T15:54:24Z
[INFO]
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/jenkins_home/workspace/My-Second-Project/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/jenkins_home/workspace/My-Second-Project/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/jenkins_home/workspace/My-Second-Project/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/jenkins_home/workspace/My-Second-Project/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/jenkins_home/workspace/My-Second-Project/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
Finished: SUCCESS
```

# Exercise #2 JenKins Job 4/4

njone company

## ■ Workspace

- 대시보드로 돌아가기
- 상태
- 변경사항
- 작업공간
- Build Now
- 구성
- Maven project 삭제



## ■ Jenkins on Docker

- HOST PC의 VOLUME MOUNT를 사용하지 않고 실행하였을 경우, jenkins\_home의 내용을 확인하기 위해서는 jenkins docker container의 terminal로 접속하여 확인

```
$ docker container exec -it [container_id or container_name] bash
```

```
root@XXXXXX:~$ cd /var/jenkins_home/workspace
```

# Setup Tomcat Plugin

njone company

- Manage Jenkins → Jenkins Plugins → available → deploy to container plugin

The screenshot shows the Jenkins management interface. On the left, there's a sidebar with various links: '+ 새로운 Item', '사람', '빌드 기록', '프로젝트 연관 관계', '파일 팔거프린트 확인', 'Jenkins 관리' (which is highlighted), 'My Views', and '새로운 뷰'. The main area is titled 'Jenkins 관리' and contains sections for 'System Configuration' and 'Global Tool Configuration'. A large red box highlights the 'Plugin Manager' section. The 'Plugin Manager' section has tabs for '업데이트된 플러그인 목록' (selected), '설치 가능', '설치된 플러그인 목록', and '고급'. A search bar shows 'deploy'. Below it, a card for the 'Deploy to container' plugin is displayed, which is version 1.16, released 1 year 8 months ago. The card includes a brief description: 'This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment'. A red box highlights this card.

+ 새로운 Item

사람

빌드 기록

프로젝트 연관 관계

파일 팔거프린트 확인

Jenkins 관리

My Views

새로운 뷰

## Jenkins 관리

### System Configuration

시스템 설정  
환경변수 및 경로 정보등을 설정합니다.

Global Tool Configuration  
Configure tools, their locations and automatic installers.

### Plugin Manager

업데이트된 플러그인 목록 설치 가능 설치된 플러그인 목록 고급

검색: deploy

Install Name ↓

Released

Deploy to container 1.16

Artifact Uploaders

This plugin allows you to deploy a war to a container after a successful build.  
Glassfish 3.x remote deployment

1 yr 8 mo ago



# Exercise #3 JenKins Job 1/7

njone company

- Item name → *My-Third-Project*
  - *Maven project*

Enter an item name

» Required field

 **Freestyle project**  
이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 다른 어떤 것에 자주 사용될 수 있습니다.

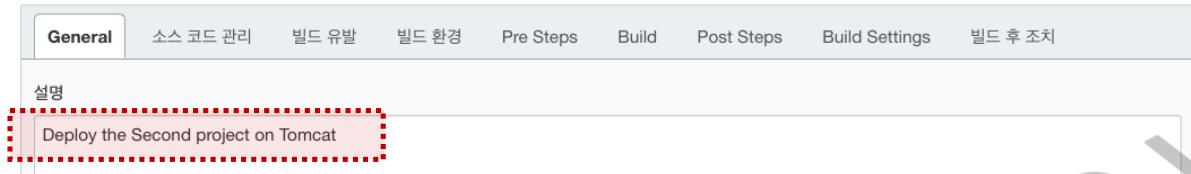
 **Maven project**  
Maven 프로젝트를 빌드합니다. Jenkins은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.

# Exercise #3 JenKins Job 2/7

njone company

## ■ General

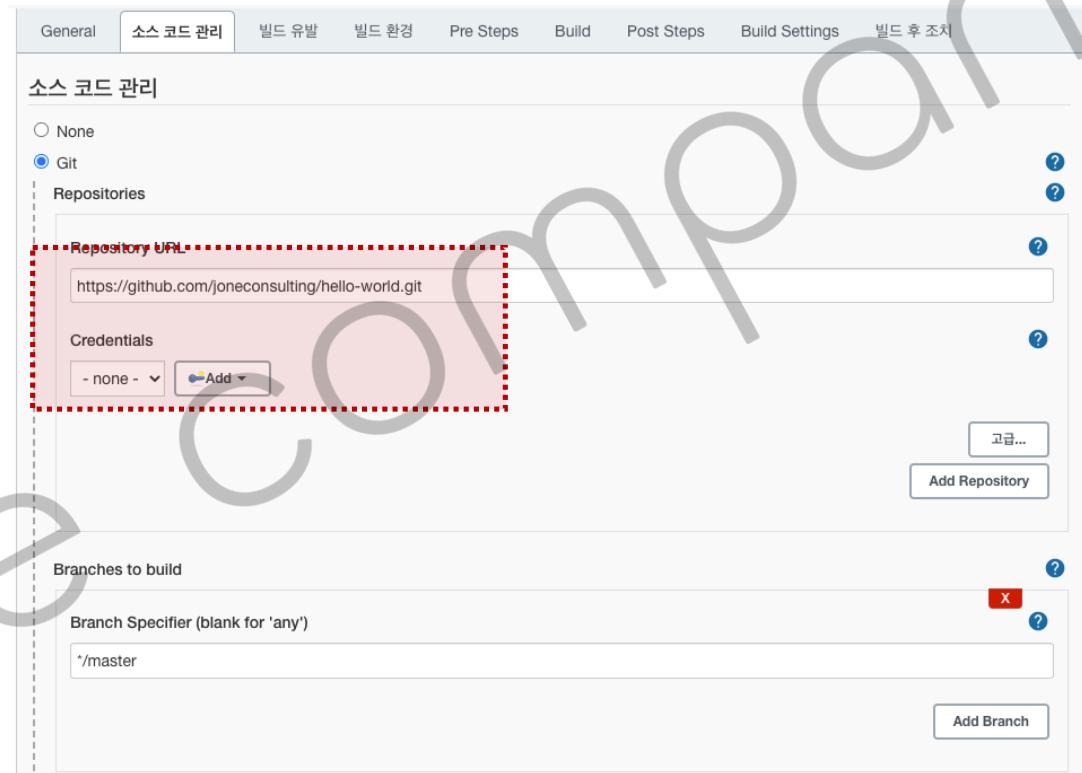
- ***Deploy the Second project on Tomcat***



## ■ Source Code Management

- Repository URL

<https://github.com/joneconsulting/cicd-web-project>



# Exercise #3 JenKins Job 3/7

njone company

## ■ Build

- Root POM: *pom.xml*
- Goals: *clean compile package*



## ■ Post-build Actions

- Deploy war/ear to a container
- *\*\*/\*.war*



# Exercise #3 JenKins Job 4/7

njone company

## ■ Post-build Actions

- Container: **Tomcat 9.x Remote**

빌드 후 조치

Deploy war/ear to a container  
WAR/EAR files ?  
\*\*/\*.war

Context path ?

Containers

- Tomcat 9.x Remote
- Credentials
- none -
- Tomcat URL ?

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: Username with password

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

Username: deployer

Treat username as secret

Password:

ID: deployer\_user

Description: user to deploy on tomcat VM

60 <user username="admin" password="admin"  
61 roles="manager-gui, manager-script, manager-jmx, manager-status"/>  
62 <user username="deployer" password="deployer"  
63 roles="manager-script"/>  
64 <user username="tomcat" password="tomcat"  
65 roles="manager-gui"/>

username: **deployer**

password: **deployer**

ID: **deployer\_user1**

Description: **user to deploy on tomcat**



# Exercise #3 JenKins Job 5/7

njone company

- Post-build Actions

- Credentials: *deployer*
- Tomcat URL: *http://192.168.0.8:8080/*



# Exercise #3 JenKins Job 6/7

njone company

## ■ Save > Build Now

Dashboard > My-Third-Project > #1

▶ 프로젝트로 돌아가기  
🔍 상태  
📝 바뀐점  
Console Output  
View as plain text  
Build 정보 수정  
Delete build '#1'  
Git Build Data

### ✓ 콘솔 출력

```
Started by user Administrator
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/My-Third-Project
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/joneconsulting/hello-world
> git init /var/jenkins_home/workspace/My-Third-Project # timeout=10
Fetching upstream changes from https://github.com/joneconsulting/hello-world
> git --version # timeout=10
> git --version # 'git version 2.30.2'
> git fetch --tags --force --progress -- https://github.com/joneconsulting/hello-world +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/joneconsulting/hello-world # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
[INFO] Maven Project ..... SUCCESS [ 0.664 s]
[INFO] Server ..... SUCCESS [ 4.339 s]
[INFO] Webapp ..... SUCCESS [ 1.277 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.977 s
[INFO] Finished at: 2021-09-03T05:10:40Z
[INFO] -----
```

```
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/jenkins_home/workspace/My-Third-Project/webapp/target/webapp.war to container Tomcat 9.x Remote with
context null
[/var/jenkins_home/workspace/My-Third-Project/webapp/target/webapp.war] is not deployed. Doing a fresh deployment.
Deploying [/var/jenkins_home/workspace/My-Third-Project/webapp/target/webapp.war]
Finished: SUCCESS
```

# Exercise #3 JenKins Job 7/7

njone company

- <http://192.168.0.8:8080/>

```
▶ ls -l ./webapps
total 8
drwxr-xr-x@ 13 dowonlee staff 416 7 31 06:12 ROOT
drwxr-xr-x@ 60 dowonlee staff 1920 7 31 06:12 docs
drwxr-xr-x@ 8 dowonlee staff 256 7 31 06:12 examples
drwxr-xr-x@ 7 dowonlee staff 224 7 31 06:12 host-manager
drwxr-xr-x@ 9 dowonlee staff 288 7 31 06:12 manager
drwxr-x--- 5 dowonlee staff 160 9 3 14:10 webapp
-rw-r----- 1 dowonlee staff 2529 9 3 14:10 webapp.war
```

The screenshot shows the Tomcat Manager Application interface at <http://192.168.0.8:8080/manager/html>. The page title is "Tomcat 웹 애플리케이션 매니저". It displays a list of applications:

경로	버전	표시 이름	실행 중	세션들
/manager	지정 안됨	Tomcat Manager Application	true	1
/webapp	지정 안됨	Webapp	true	0

The screenshot shows a web browser window with the URL <http://192.168.0.8:8080/webapp/>. The page content is "Hello, Welcome to Simple DevOps Project !!".

# Setup PollSCM

njone company

## ■ Project > Configure > Build Triggers

- **Build periodically** → cron job
- **Poll SCM** → cron job



### cron

위키백과, 우리 모두의 백과사전.

소프트웨어 유틸리티 cron은 유닉스 계열 컴퓨터 운영 체제의 시간 기반 잡 스케줄러이다. 소프트웨어 환경을 설정하고 관리하는 사람들은 작업을 고정된 시간, 날짜, 간격에 주기적으로 실행할 수 있도록 스케줄링하기 위해 cron을 사용한다.

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
  - Schedule
  - \*\*\*\*\*

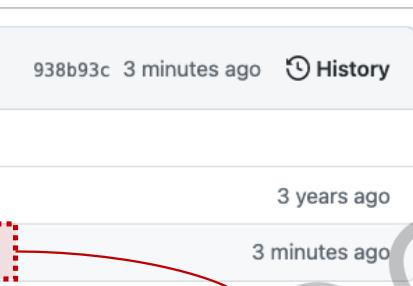
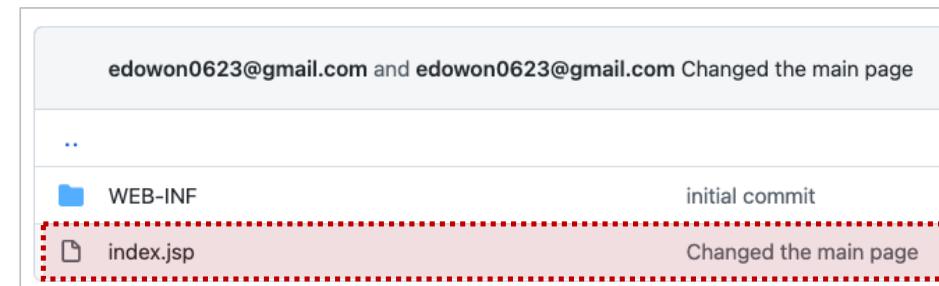
⚠ Do you really mean "every minute" when you say "\*\*\*\*\*"? Perhaps you meant "H \* \* \* \*" to poll once per hour  
Would last have run at Friday, September 3, 2021 at 5:28:06 AM Coordinated Universal Time; would next run at Friday, September 3, 2021 at 5:28:06 AM Coordinated Universal Time.

# Setup PollSCM

njone company

## ■ New commit

- 코드 수정
- git add → git commit → git push



A screenshot of a build history panel. It lists three builds: #3 (2021. 9. 3. 오전 5:46), #2 (2021. 9. 3. 오전 5:37), and #1 (2021. 9. 3. 오전 5:10). Each build entry has a green checkmark icon. A red dashed box highlights the first build entry.

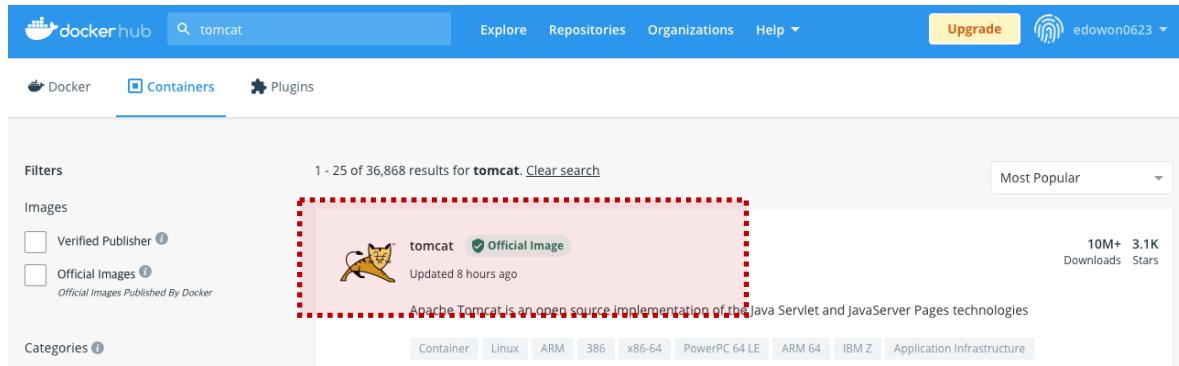
build	date
#3	2021. 9. 3. 오전 5:46
#2	2021. 9. 3. 오전 5:37
#1	2021. 9. 3. 오전 5:10



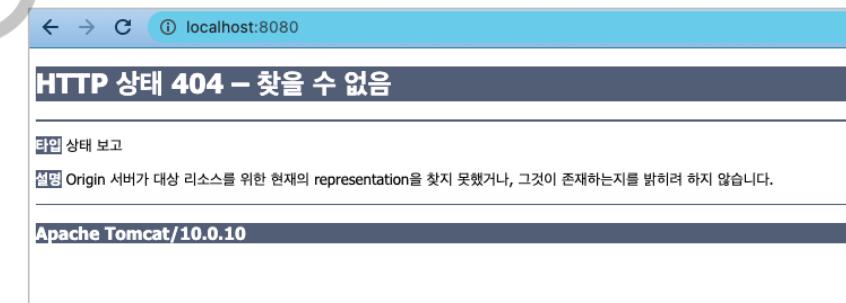
# Install Tomcat Container on Docker

njone company

- <https://hub.docker.com>



- **\$ docker pull tomcat:latest**
- **\$ docker run -d --name tomcat -p 8080:8080 tomcat:9.0**



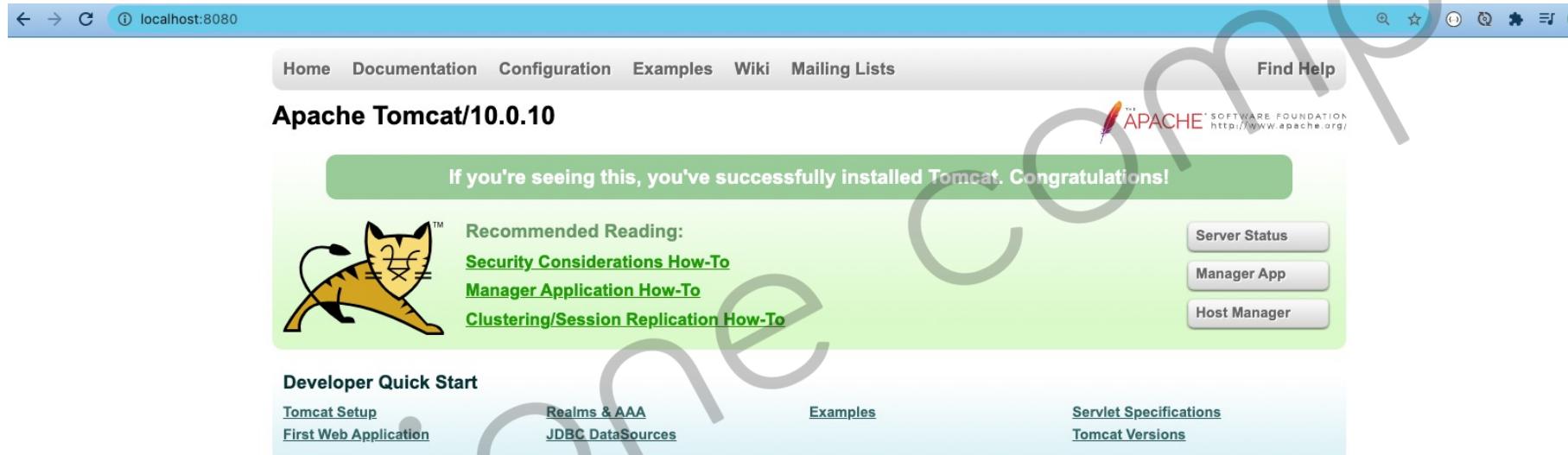


# Setup Tomcat Container on Docker

njone company

- `$ docker stop [CONTAINER_ID] && docker rm [CONTAINER_ID]`
- `$ docker run -d --name tomcat -p 8080:8080 tomcat:9.0`
- `$ docker exec -it [CONTAINER_ID] bash`

`root@[CONTAINER_ID]:/usr/local/tomcat# cp -R ./webapps.dist/* ./webapps/`



# Setup Publish Over Plugin

njone company

- Manage Jenkins → Jenkins Plugins → available → publish over ssh

The screenshot shows the Jenkins System Configuration page. On the left, there is a sidebar with various management options: '새로운 Item', '사람', '빌드 기록', 'Jenkins 관리' (which is selected), 'My Views', 'Lockable Resources', and 'New View'. The main content area is titled 'Jenkins 관리' and 'System Configuration'. It contains three sections: '시스템 설정' (Configure build number and cron information), '노드 관리' (Add, remove, control and monitor nodes), and 'Global Tool Configuration' (Configure tools, their locations and automatic installers). A red dashed box highlights the '플러그인 관리' section on the right, which is described as a place to manage plugins to enhance Jenkins' functionality.

The screenshot shows the Jenkins Plugin Manager interface. A search bar at the top contains the text 'publish over'. Below it, there are tabs: '업데이트된 플러그인 목록' (Updated plugin list), '설치 가능' (Installable), '설치된 플러그인 목록' (Installed plugin list), and '고급' (Advanced). A red dashed box highlights the 'Install' column. The results list includes the 'Infrastructure plugin for Publish Over X' and the 'Publish Over SSH' plugin. The 'Publish Over SSH' plugin is highlighted with a red dashed box. It has two tabs: 'Build Tools' and 'Artifact Uploaders'. A message below the list states: 'This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.'

# Setup Publish Over Plugin

njone company

- Manage Jenkins → Configure System → Publish over SSH

- Add SSH Servers
  - Name: ***docker-host***
  - Hostname: **[Remote IP] ex)192.168.0.8**
  - Username: ***root***
  - Passphrase/Password: **P@ssw0rd**
  - Port: **10022**
- Test Configuration

SSH Servers

<input type="checkbox"/> SSH Server
Name
dind-host
Hostname
192.168.0.8
Username
root
Remote Directory
/root
<input checked="" type="checkbox"/> Use password authentication, or use a different key
Passphrase / Password
 Concealed
Port
10022

Success

**Test Configuration**

# Exercise 4# JenKins Job 1/8

njone company

- Item name → ***My-Docker-Project***
  - Copy from: ***My-Third-Project***

Enter an item name

» 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.

**Maven project**  
Maven 프로젝트를 빌드합니다. Jenkins은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.

**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.

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

**Copy from**

**OK**



# Exercise 4# JenKins Job 2/8

njone company

- Build Triggers
  - Poll SCM → **Disable**
- Post-build Actions
  - Deploy war/ear to a container → **Delete**
  - ***Send build artifacts over SSH***

- ***SSH Server***

- Name: [Publish over SSH에서 설정한 이름] ex) docker-host
- ***Transfer Set***
  - Source files: ***target/\*.war***
  - Remove prefix: ***target***
  - Remote directory: ***.***

# Exercise 4# JenKins Job 3/8

njone company

## ■ Save > Build Now

```
[INFO] Maven Project ..... SUCCESS [ 0.617 s]
[INFO] Server ..... SUCCESS [ 3.748 s]
[INFO] Webapp ..... SUCCESS [ 1.167 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.019 s
[INFO] Finished at: 2021-09-03T13:26:06Z
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/webapp/target/webapp.war to com.example.maven-project/webapp/1.0-SNAPSHOT/webapp-1.0-SNAPSHOT.war
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/server/pom.xml to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.pom
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/server/target/server.jar to com.example.maven-project/server/1.0-SNAPSHOT/server-1.0-SNAPSHOT.jar
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/pom.xml to com.example.maven-project/maven-project/1.0-SNAPSHOT/maven-project-1.0-SNAPSHOT.pom
channel stopped
SSH: Connecting from host [512076bd5890]
SSH: Connecting with configuration [docker-host] ...
SSH: Disconnecting configuration [docker-host] ...
SSH: Transferred 1 file(s)
Finished: SUCCESS
```

```
16b4daa4cf99:~# ls -l
total 8
-rw-r--r--    1 root      root          127 Sep  3 14:37 Dockerfile
-rw-r--r--    1 root      root        2443 Sep  5 14:04 webapp.war
```

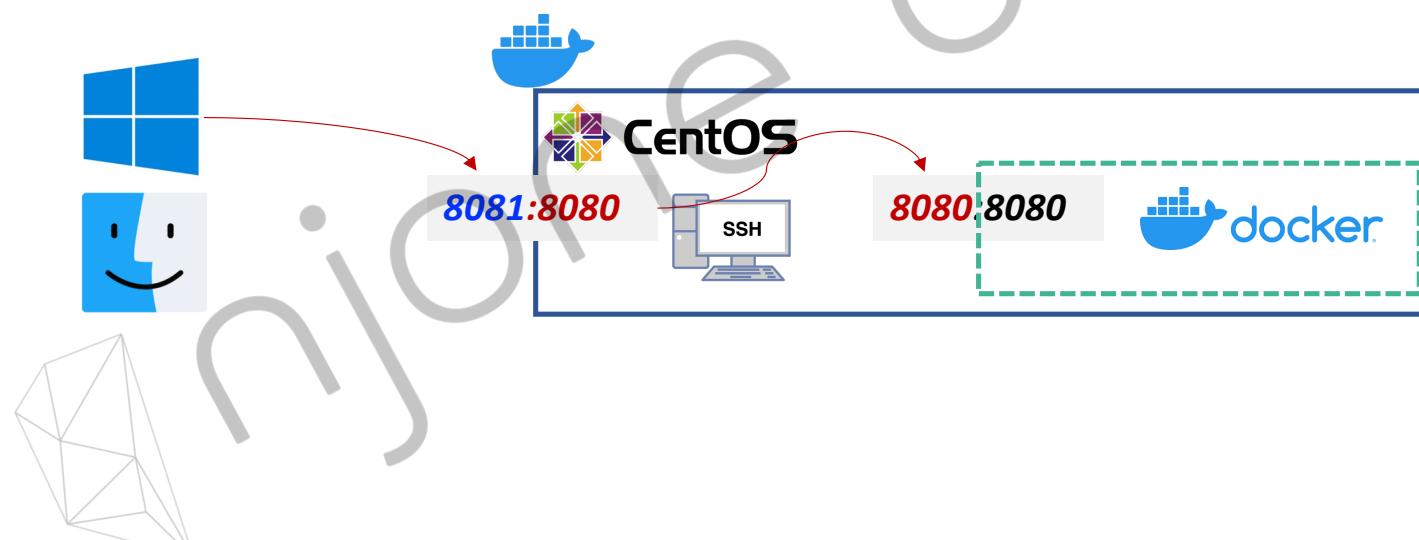
# Exercise 4# JenKins Job 4/8

njone company

- Create a Dockerfile on Docker server

```
1 FROM tomcat:9.0  
2  
3 COPY ./hello-world.war /usr/local/tomcat/webapps
```

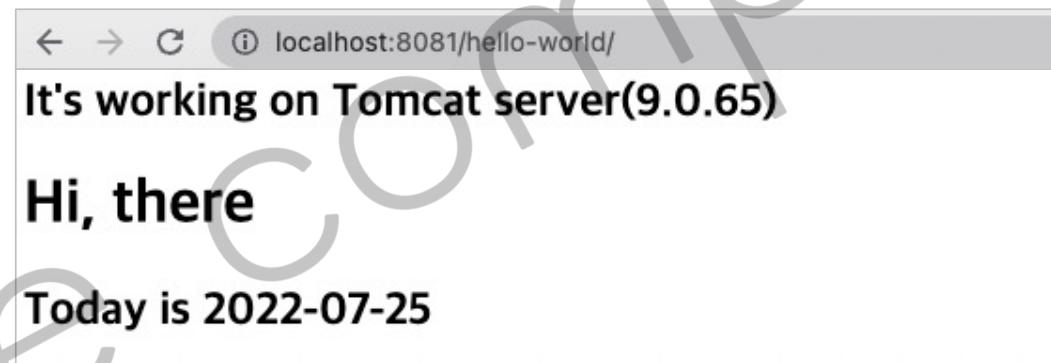
- **\$ docker build --tag=cicd-project -f Dockerfile .**
- **\$ docker images**
- **\$ docker image inspect cicd-project:latest**
- **\$ docker run -p 8080:8080 --name mytomcat cicd-project:latest**



# Exercise 4# Jenkins Job 5/8

njone company

```
▶ docker run -d -p 8081:8080 --name mytomcat cicd-project:latest  
adecc32eb40825fff90de64a037792a1017b2fa157d1f3962bdb2551f826b597  
(base) downonlee ~/Desktop/Work/14.online/devops▶  
▶ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS  
adecc32eb408 cicd-project:latest "catalina.sh run"  
mytomcat 3 seconds ago Up 2 seconds 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp  
512076bd5890 jenkins/jenkins:lts-jdk11 "/sbin/tini -- /usr/..." 7 hours ago Up 7 hours 0.0.0.0:50000->50000/tcp, :::50000->50000/tcp,  
tcp, 0.0.0.0:8088->8080/tcp, :::8088->8080/tcp nervous_wing  
f48fbfb0d1968 tomcat:latest "catalina.sh run"  
tomcat 7 hours ago Up 7 hours 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
```



- **\$ docker stop mytomcat**
- **\$ docker rm mytomcat**
- **\$ docker rmi cicd-project:latest**
- **\$ rm -rf webapp.war**



# Exercise 4# JenKins Job 6/8

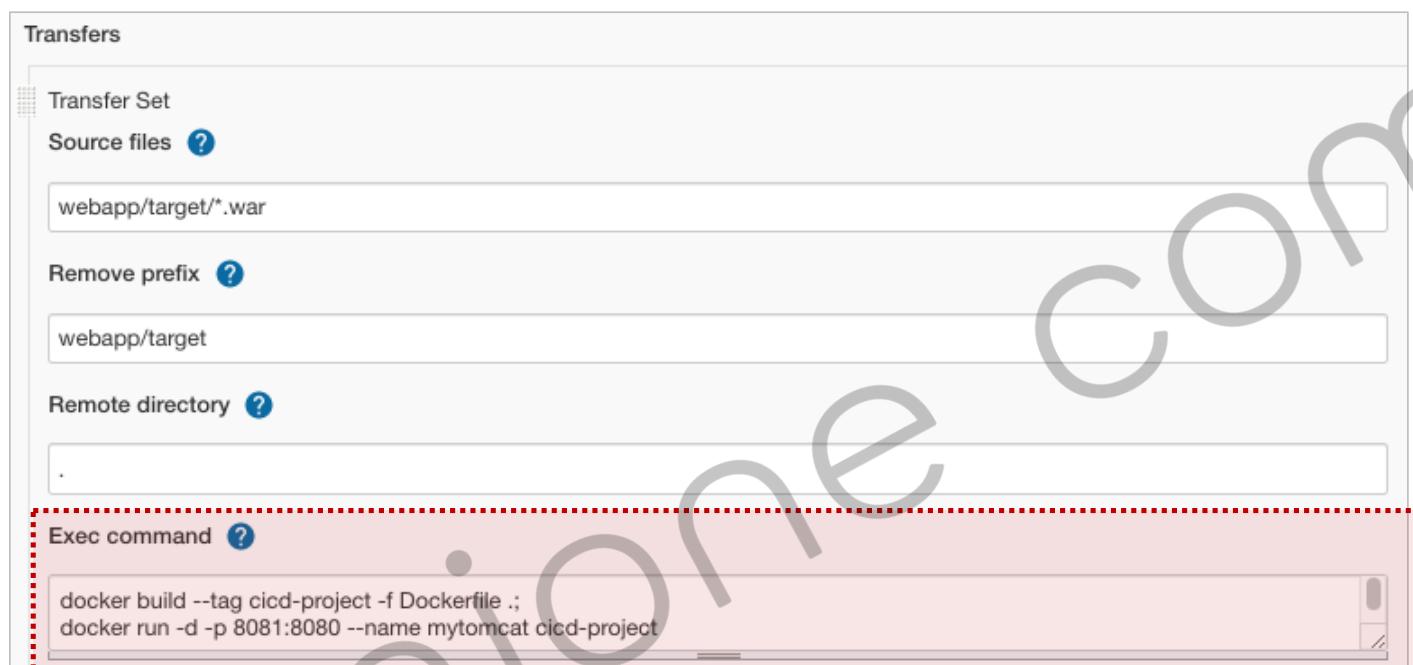
njone company

- Project > Configure > Post Build Actions

- *Exec command*

```
docker build --tag=cicd-project -f Dockerfile .;
```

```
docker run -d -p 8080:8080 --name mytomcat cicd-project:latest
```





# Exercise 4# Jenkins Job 7/8

njone company

- Save > Build Now

```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 6.874 s  
[INFO] Finished at: 2021-09-03T14:54:14Z  
[INFO] -----  
Waiting for Jenkins to finish collecting data  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/webapp/pom.xml to com.example.maven-project  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/webapp/target/webapp.war to com.example.maven-project  
SNAPSHOT.war  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/server/pom.xml to com.example.maven-project  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/server/target/server.jar to com.example.maven-project  
SNAPSHOT.jar  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/pom.xml to com.example.maven-project  
SNAPSHOT.pom  
channel stopped  
SSH: Connecting from host [512076bd5890]
```

```
[root@ae2b6f08d2dc ~]# docker images  
REPOSITORY          TAG           IMAGE ID        CREATED         SIZE  
cicd-project        latest        362a7c69e220   53 minutes ago  680 MB  
docker.io/tomcat     latest        ab1f0e1bb1a1    41 hours ago   680 MB  
[root@ae2b6f08d2dc ~]# docker ps  
CONTAINER ID        IMAGE          COMMAND        CREATED        STATUS          PORTS          NAMES  
e6011dc8d047        cicd-project   "catalina.sh run"  53 minutes ago  Up 53 minutes  0.0.0.0:8081->8080/tcp  mytomcat
```



# Exercise 4# JenKins Job 8/8

njone company

- Build Now (after changed sources)

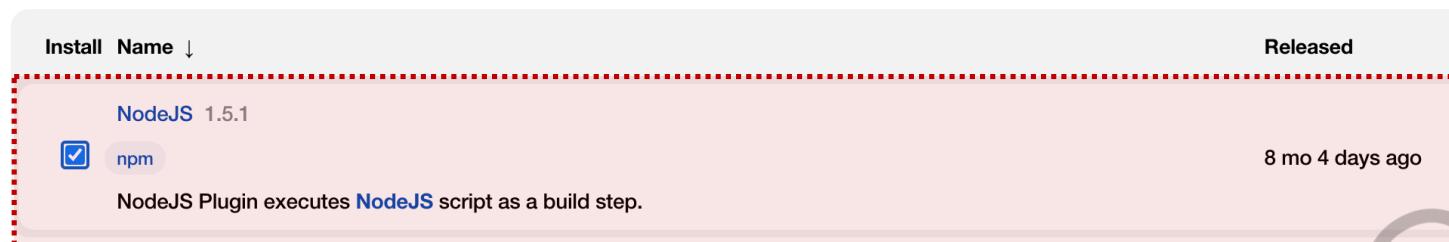
```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time:  7.138 s  
[INFO] Finished at: 2021-09-03T14:57:49Z  
[INFO] -----  
Waiting for Jenkins to finish collecting data  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/webapp/pom.xml to com.example.maven-project/webapp/1.0-SNAPS  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/webapp/target/webapp.war to com.example.maven-project/webapp  
SNAPSHOT.war  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/server/pom.xml to com.example.maven-project/server/1.0-SNAPS  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/server/target/server.jar to com.example.maven-project/server  
SNAPSHOT.jar  
[JENKINS] Archiving /var/jenkins_home/workspace/My-Docker-Project/pom.xml to com.example.maven-project/maven-project/1.0-SNAPS  
SNAPSHOT.pom  
channel stopped  
SSH: Connecting from host [512076bd5890]  
SSH: Connecting with configuration [docker-host] ...  
SSH: EXEC: completed after 1,005 ms  
SSH: Disconnecting configuration [docker-host] ...  
ERROR: Exception when publishing, exception message [Exec exit status not zero. Status [125]]  
Build step 'Send build artifacts over SSH' changed build result to UNSTABLE  
Finished: UNSTABLE
```



# Exercise 4-2# JenKins Job 1/5

njone company

- Jenkins Server) Nodejs 설치
  - Plugin Manager > Available > nodejs



- Global Tool Configuration > NodeJS 설치





# Exercise 4-2# JenKins Job 2/5

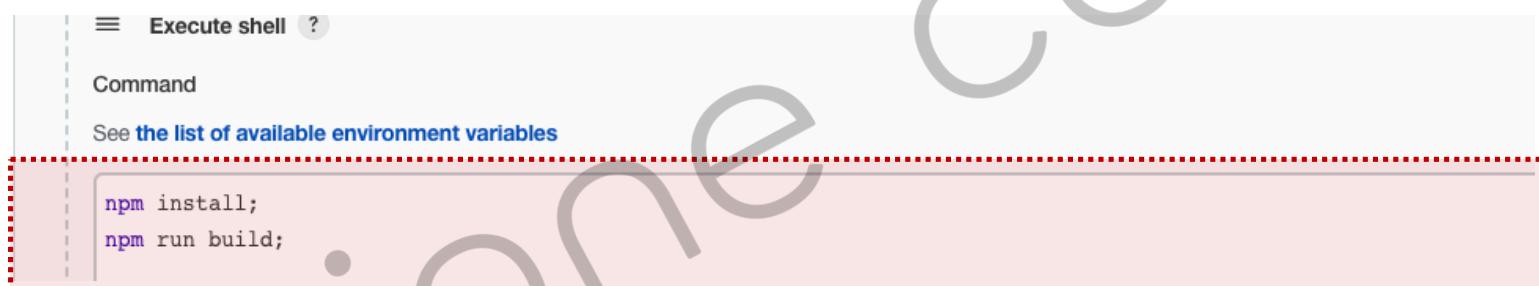
njone company

- Item name → ***My-Nodejs-Project***

- Build > Execute NodeJS script > ***Nodejs-16.17.0 선택***



- Build > Execute shell





# Exercise 4-2# JenKins Job 3/5

njone company

- Save > Build Now

```
Commit message: "upload project files"
> git rev-list --no-walk c9c23da81a9b98a5a53f5d8122086b035a0e7f3b # timeout=10
[My-Nodejs-Project] $ /var/jenkins_home/tools/jenkins.plugins.nodejs.tools.NodeJSInstallation/Nodejs-16.17.0/bin/node /tmp/jenkins14481587423122626179.js
[My-Nodejs-Project] $ /bin/sh -xe /tmp/jenkins16441738717920205999.sh
+ npm install

added 309 packages, and audited 310 packages in 2s

48 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
+ npm run build

> example@1.0.0 build
> babel app.js -d dist

Successfully compiled 1 file with Babel (148ms).
Finished: SUCCESS
```

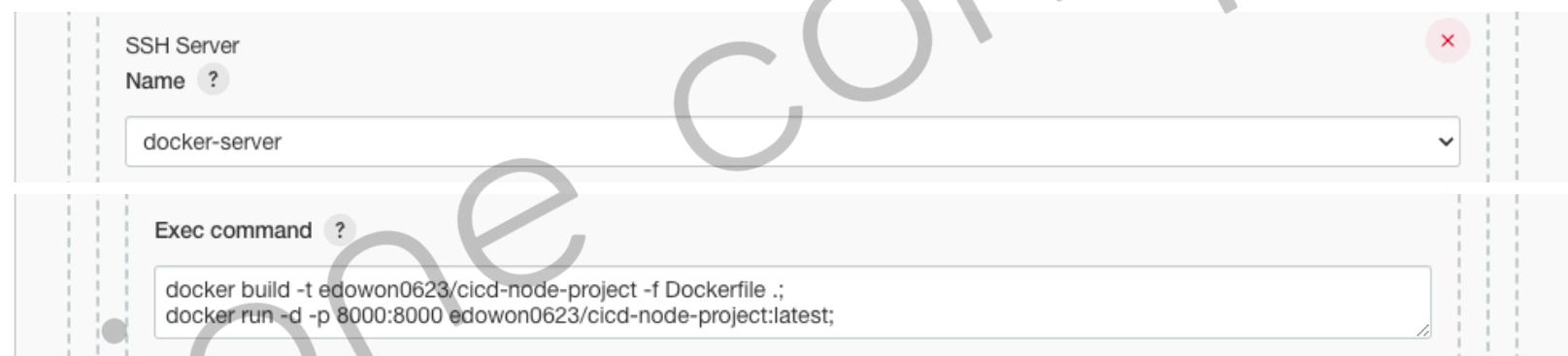
```
jenkins@e1253d619fc1:~/workspace/My-Nodejs-Project$ ls -al
total 360
drwxr-xr-x  5 jenkins jenkins  4096 Sep 18 15:32 .
drwxr-xr-x 11 jenkins jenkins  4096 Sep 16 04:28 ..
drwxr-xr-x  8 jenkins jenkins  4096 Sep 18 15:32 .git
-rw-r--r--  1 jenkins jenkins   478 Sep 16 04:28 Dockerfile
-rw-r--r--  1 jenkins jenkins   252 Sep 16 04:28 app.js
-rw-r--r--  1 jenkins jenkins    42 Sep 16 04:28 babel.config.json
drwxr-xr-x  2 jenkins jenkins  4096 Sep 18 15:32 dist
-rw-r--r--  1 jenkins jenkins    92 Sep 16 04:28 ecosystem.config.js
drwxr-xr-x 190 jenkins jenkins 12288 Sep 18 15:32 node_modules
-rw-r--r--  1 jenkins jenkins 317553 Sep 18 15:32 package-lock.json
-rw-r--r--  1 jenkins jenkins   431 Sep 16 04:28 package.json
jenkins@e1253d619fc1:~/workspace/My-Nodejs-Project$ █
```

# Exercise 4-2# JenKins Job 4/5

njone company

- Delete → Build > Execute NodeJS script, Execute Shell
- Post Build > Send build artifacts over SSH → ***docker-server***
  - Source files: \*\*
  - Remote directory: .
  - Exec command:

```
docker build -t edowon0623/cicd-node-project -f Dockerfile .;  
docker run -d -p 8000:8000 edowon0623/cicd-node-project:latest;
```





# Exercise 4-2# JenKins Job 5/5

njone company

- Save > Build Now

```
jenkins@e1253d619fc1:~/workspace/My-Nodejs-Project$ ls -al
total 32
drwxr-xr-x  3 jenkins jenkins 4096 Sep 19 02:07 .
drwxr-xr-x 11 jenkins jenkins 4096 Sep 16 04:28 ..
drwxr-xr-x  8 jenkins jenkins 4096 Sep 19 03:23 .git
-rw-r--r--  1 jenkins jenkins  271 Sep 19 02:07 Dockerfile
-rw-r--r--  1 jenkins jenkins  252 Sep 19 02:07 app.js
-rw-r--r--  1 jenkins jenkins   42 Sep 19 02:07 babel.config.json
-rw-r--r--  1 jenkins jenkins   92 Sep 19 02:07 ecosystem.config.js
-rw-r--r--  1 jenkins jenkins 431 Sep 19 02:07 package.json
jenkins@e1253d619fc1:~/workspace/My-Nodejs-Project$
```

```
[root@3bd341fe9bdd ~]# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
edowon0623/cicd-node-project    latest   21109ecd9821  31 minutes ago  934MB
tomcat              latest   d9c12d68a96f  2 weeks ago   469MB
node                10      8951689bc94c  17 months ago  859MB
[root@3bd341fe9bdd ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS              NAMES
a31278417596        edowon0623/cicd-node-project:latest    "docker-entrypoint.s..."   32 minutes ago   Up 32 minutes
0.0.0.0:8000->8000/tcp      zealous_tharp
```



## 후속 강의 소개

- ~~Spring Cloud로 개발하는 마이크로서비스 애플리케이션~~
- ~~Jenkins를 이용한 CI/CD Pipeline 구축~~
- **Microservice Architecture 와 Patterns**
- **Spring Boot 와 WebFlux를 이용한 Reactive RESTful API 개발**