

5. ssh+docker

// docker로 ssh 컨테이너 생성 후, jenkins에서 빌드한 파일을 ssh 서버로 copy
ssh 컨테이너 내부에서 docker로 tomcat 컨테이너를 생성하여 복사해온 war 파일로 배포

☞

1. publish over ssh 플러그인 설치 및 설정

- 설치

Download progress

준비

- Checking internet connectivity
- Checking update center connectivity
- Success

| | |
|--|-----------|
| Oracle Java SE Development Kit Installer | ✓ 성공 |
| Command Agent Launcher | ✓ 성공 |
| Infrastructure plugin for Publish Over X | ✓ 성공 |
| Publish Over SSH | ✓ 성공 |
| Loading plugin extensions | ✓ Success |

→ [메인 페이지로 돌아가기](#)

(설치된 플러그인을 바로 사용하실 수 있습니다.)

→ ☐ 설치가 끝나고 실행중인 작업이 없으면 Jenkins 재시작.

- 설정 (2번의 ssh docker 생성 후 진행하여도 됨)
 - jenkins 관리 > System > Publish over SSH

Name ?

docker-server

Hostname ?

192.168.56.1

Username ?

root

Remote Directory ?

/root

☐ Avoid sending files that have not changed ?

고급 ^ Edited

☒ Use password authentication, or use a different key ?

Passphrase / Password ?

Path to key ?

Key ?

Jump host ?

Port ?

10022

- Hostname에는 windows(local) IP를 입력해준다

2. docker로 SSH 서버 실행

- Windows1) SSH 서버 (with 도커 + dind) 실행 명령어 (방법1)
 - `docker run --privileged --name docker-server -itd -p 10022:22 -p 8081:8080 -e container=docker -v /sys/fs/cgroup:/sys/fs/cgroup edowon0623/docker:latest /usr/sbin/init`
- Windows2) SSH 서버 (with 도커 + dood) 실행 명령어 (방법2, 위 명령어로 실행되지 않을 경우)
 - `docker run -itd --name docker-server -p 10022:22 -e container=docker --tmpfs /run --tmpfs /tmp -v /sys/fs/cgroup:/sys/fs/cgroup:ro -v /var/run/docker.sock:/var/run/docker.sock edowon0623/docker:latest /usr/sbin/init`
- Windows3) SSH 서버 (with 도커 + no_iptables) 실행 명령어 (방법3, 위 명령어로 실행되지 않을 경우, iptables 사용하지 않고 실행)
 - `docker run -itd --name docker-server -p 10022:22 -e container=docker --tmpfs /run --tmpfs /tmp -v /sys/fs/cgroup:/sys/fs/cgroup:ro -v /var/run/docker.sock:/var/run/docker.sock edowon0623/docker:no_iptables /usr/sbin/init`
- configuration 테스트

SSH Servers

SSH Server

Name ?

docker-server

Hostname ?

192.168.56.1

Username ?

root

Remote Directory ?

/root

☐ Avoid sending files that have not changed ?

고급 ▾

✎ Edited

Success

Test Configuration


3. item copy

- 4. 두번째 **maven-project** item copy


Enter an item name

My-Docker-Project


» 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.




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.




Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

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



Copy from

Third-Project

OK

- Poll SCM 해제

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 Branches
- ☐ GitHub Pull Requests ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

- 빌드 후 조치 추가

빌드 후 조치

Send build artifacts over SSH ?

SSH Publishers

SSH Server

Name ?

docker-server

고급 ▾

Transfers

Transfer Set

Source files ?

target/*.war

Remove prefix ?

target

Remote directory ?

-

Exec command ?

❗

Either Source files, Exec command or both must be supplied

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

고급 ▾

4. 빌드 후 파일 copy 확인

- ssh 서버 내부에서 파일 생성되었는지 확인

```
[root@173e33f43667 ~]# ls -al
total 48
dr-xr-x--- 1 root root 4096 Sep 23 11:55 .
drwxr-xr-x 1 root root 4096 Sep 23 11:36 ..
-rw----- 1 root root 2361 Sep 15 2021 anaconda-ks.cfg
-rw-r--r-- 1 root root 608 Sep 15 2021 anaconda-post.log
-rw----- 1 root root 39 Sep 23 12:00 .bash_history
-rw-r--r-- 1 root root 18 May 11 2019 .bash_logout
-rw-r--r-- 1 root root 176 May 11 2019 .bash_profile
-rw-r--r-- 1 root root 176 May 11 2019 .bashrc
-rw-r--r-- 1 root root 100 May 11 2019 .cshrc
-rw-r--r-- 1 root root 129 Sep 20 2022 Dockerfile
-rw----- 1 root root 2059 Sep 15 2021 original-ks.cfg
-rw-r--r-- 1 root root 129 May 11 2019 .tcshrc
[root@173e33f43667 ~]#
[root@173e33f43667 ~]# ls -al
total 7888
dr-xr-x--- 1 root root 4096 Sep 23 12:41 .
drwxr-xr-x 1 root root 4096 Sep 23 11:36 ..
-rw----- 1 root root 2361 Sep 15 2021 anaconda-ks.cfg
-rw-r--r-- 1 root root 608 Sep 15 2021 anaconda-post.log
-rw----- 1 root root 39 Sep 23 12:00 .bash_history
-rw-r--r-- 1 root root 18 May 11 2019 .bash_logout
-rw-r--r-- 1 root root 176 May 11 2019 .bash_profile
-rw-r--r-- 1 root root 176 May 11 2019 .bashrc
-rw-r--r-- 1 root root 100 May 11 2019 .cshrc
-rw-r--r-- 1 root root 129 Sep 20 2022 Dockerfile
-rw-r--r-- 1 root root 8025930 Sep 23 12:41 hello-world.war
-rw----- 1 root root 2059 Sep 15 2021 original-ks.cfg
-rw-r--r-- 1 root root 129 May 11 2019 .tcshrc
[root@173e33f43667 ~]#
[root@173e33f43667 ~]#
```

5. ssh 서버 내부에서 docker로 tomcat 올리기

- Dockerfile 작성

```
FROM tomcat:9.0
```

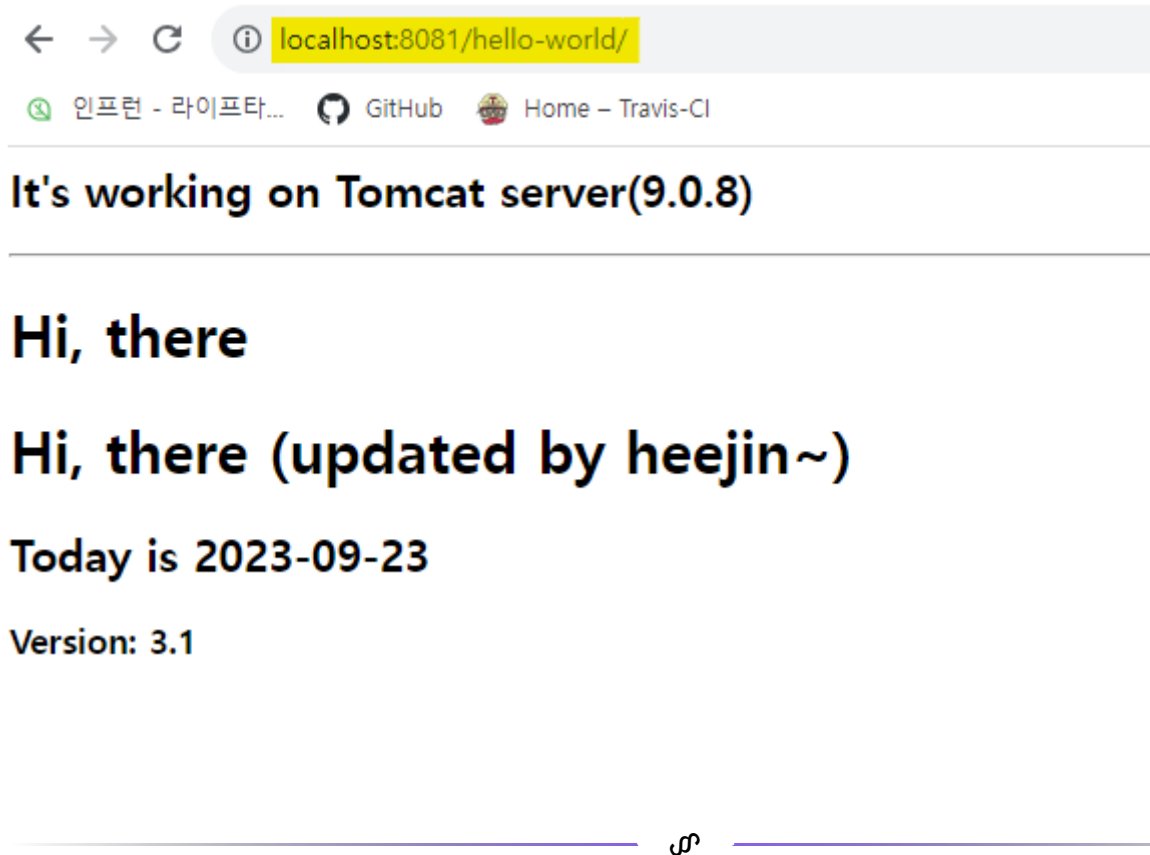
```
LABEL org.opencontainers.image.authors="edowon0623@gmail.com"
```

```
COPY ./hello-world.war /usr/local/tomcat/webapps
```

- 이미지 빌드 명령어 (Container 내에서 테스트)
 - `docker build --tag docker-server -f Dockerfile .`
- 컨테이너 실행 명령어 (Container 내에서 테스트)

- `docker run --privileged -p 8080:8080 --name mytomcat docker-server:latest`
- 컨테이너 중지, 삭제
 - `docker stop mytomcat`
 - `docker rm mytomcat`

6. tomcat 확인



7. 5번에서 수동으로 한 작업을 jenkins에서 수행해보기

- item 설정에서 Exec command 추가

```
docker build --tag=cicd-project -f Dockerfile .;
docker run -d -p 8080:8080 --name mytomcat cicd-project:latest
```


SSH Server

Name ?

docker-server

고급 ▾

Transfers

Transfer Set

Source files ?

target/*.war

Remove prefix ?

target

Remote directory ?

.

Exec command ?

```
docker build --tag=cicd-project -f Dockerfile .;
docker run -d -p 8080:8080 --name mytomcat cicd-project:latest
```

- 빌드 시작 전에 ssh 컨테이너 안의 war 파일 삭제해주기

```
root@173e33f43667 ~]# ls
anaconda-ks.cfg anaconda-post.log Dockerfile hello-world.war original-ks.cfg
root@173e33f43667 ~]#
root@173e33f43667 ~]#
root@173e33f43667 ~]# rm hello-world.war
rm: remove regular file 'hello-world.war'? y
root@173e33f43667 ~]#
root@173e33f43667 ~]# ls
anaconda-ks.cfg anaconda-post.log Dockerfile original-ks.cfg
root@173e33f43667 ~]#
```

docker run --privileged --name docker-server -itd -p 10022:22 -p 8088:8080 -e container=docker -v /sys/fs/cgroup:/sys/fs/cgroup edowon0623/docker:latest /usr/sbin/init