

# 2630.Jenkins Master-Slave 연동 구성하기

본 문서에서는 Jenkins Slave 를 Master 와 연동하는 방법을 설명합니다.

Jenkins Master-Slave 구성에 대한 내용은 [2620. Jenkins 구성](#) 페이지를 참고합니다.

Jenkins Master-Slave 구성 시 분산된 환경에서 빌드와 테스트를 수행할 수 있으며, 각 프로젝트마다 필요한 작업을 병렬로 처리할 수 있습니다.

Jenkins Master는 중앙 서버의 역할을 하여 전역 설정, Job 정보, 빌드 이력 등을 관리하며, **실제로 빌드와 테스트는 Jenkins Slave 에서 실행**됩니다.



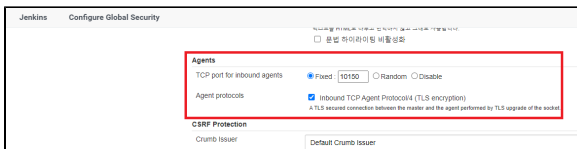
- 각 업무팀 빌드서버에 Jenkins Agent를 설치하시는 경우, [3.2. Jenkins Slave Agent 설치하기](#) 항목을 보시면 됩니다.
- NT서버에 Agent를 설치하시는 경우에는 하위문서 [NT서버 Jenkins Master-Slave 연동 구성하기](#)를 참조하시기 바랍니다

## 1. Jenkins Master Node

Node는 Job이 실행 가능한 서버를 말하며, Master와 Agent로 구분됩니다.

### 1.1. Jenkins Master

- Jenkins가 설치되어 있는 서버이며, Jenkins 설정, 빌드 이력 등을 관리합니다.
- Master와 Agent 간의 연결이 사용하는 포트는 'Jenkins 관리 > Configure Global Security' 에서 확인 가능합니다.(10150 포트를 Fixed로 사용합니다.)



### 1.2. Slave Agent

- 분산된 환경에서 개별 Job의 실행이 이루어지는 서버입니다.
- 별도의 VM에서 구동하여 부하를 분산합니다.
- Slave Agent별 구동 상황은 'Jenkins 관리 > 노드 관리' 에서 확인 가능합니다.

## 2. Jenkins Master에 Node 구성하기

1. 'Jenkins 관리 > 노드 관리 > 신규 노드'를 선택합니다.
2. 노드명을 입력하고, 노드복사에 체크한 후, 이미 생성되어 있는 노드명을 입력하여 생성합니다.

### 2.1. Node 명명규칙

- 기본 명명규칙 : {시스템식별자}\_{환경}\_{서버호스트명} (예) DVS\_STG\_SKT-DOPSPAP01
  - 시스템식별자 : JIRA 프로젝트 생성 시 부여된 식별자를 사용합니다.
  - 환경
    - DEV - 개발 환경
    - STG - 스테이징 환경
    - QA - QA 환경
    - PRD - 운영환경
  - 서버호스트명 : Jenkins Slave Agent 가 설치되는 서버의 호스트명을 작성합니다.



- 운영/개발/스테이징 환경별 Slave Agent 를 설치하는 것을 권장합니다.

#### 목차

- 1. Jenkins Master Node
  - 1.1. Jenkins Master
  - 1.2. Slave Agent
- 2. Jenkins Master에 Node 구성하기
  - 2.1. Node 명명규칙
  - 2.2. Jenkins Master에 Node 생성하기
- 3. Jenkins Slave Agent 설치하기
  - 3.1. Jenkins Slave Agent 설치를 위한 환경준비
    - 3.1.1. Jenkins Slave Agent 설치 및 실행을 위한 Application 계정 생성
    - 3.1.2. JDK 1.8 이상 설치
    - 3.1.3. Git 최신 버전 설치
    - 3.1.4. 빌드 Tool 설치
    - 3.1.5. 방화벽 오픈 신청
  - 3.2. Jenkins Slave Agent 설치하기
    - 3.2.1. Jenkins Slave Agent 구동하기
    - 3.2.2. Jenkins Slave Agent 구동시 오류 조치하기
    - 3.2.3. Jenkins Slave Agent 중지하기

## 2.2. Jenkins Master에 Node 생성하기

Jenkins 관리 > 노드 관리 > 신규 노드 를 선택합니다.

Permanent Agent 를 선택하고 노드명에 Node 명명규칙에 따라 노드명을 입력하고 OK 버튼을 클릭 합니다.

노드 설정 화면에서 필요한 정보들을 입력합니다.

항목	설명
# of executors	동시에 실행 가능한 빌드 개수. Pipeline Job이 실행될 경우, Pipeline과 Child Job이 동시에 실행되기 때문에 이를 고려하여 설정합니다.
Remote root directory	해당 노드의 Root directory를 설정합니다. 이 디렉토리를 기준으로 원격 서버에 설치된 Jenkins Slave Agent 의 Working Directory 가 생성됩니다. (예) /app/myslave 로 설정시 working directory /app/myslave/workspace 가 생성됩니다.
Labels	레이블을 활용하여 Slave Agent 를 그룹핑할 수 있습니다. 레이블 기능을 활용하여 Slave Agent 를 로드밸런싱으로 구축할 수 있습니다.  스페이스를 구분값으로 여러개의 레이블을 지정할 수 있습니다.
Usage	Jenkins 가 빌드할 때 노드의 활용범위를 설정할 수 있습니다. <b>Only build jobs with label expressions matching this node</b> 를 지정하여 다른 업무에서 활용하지 못하도록 합니다.
Launch method	노드를 구동하는 방법을 설정합니다. <b>JNLP 통신방식을 사용하므로 Launch agent by connecting it to the master</b> 을 선택합니다.
Availability	Jenkins 가 Slave Agent 를 시작, 중지 및 사용하는 방식을 정의합니다. <b>Keep this agent online as much as possible</b> 을 선택하여 최대한 온라인 상태를 유지합니다. 일시적인 네트워크 장애 등으로 Slave Agent 가 오프라인이 된다면 Jenkins 는 주기적으로 Slave Agent 를 재시작 합니다.

Jenkins Master-Slave 통신은 JNLP(Java Network Launching Protocol) 에 의해 통신이 이루어집니다.

고급버튼을 클릭하면 나타나는 항목인 Tunnel connection through 에 Jenkins Master 서버정보를 입력합니다.

업무구분	Jenkins Master URL	Tunnel connection through
PoC	/mybuild	61.250.19.35:10150
스윙	/mybuildswg	61.250.19.36:10150
과금	/mybuildrater	61.250.19.34:10150



- Source : 빌드서버 (주로 개발서버), 재기동 대상 WAS 서버 (개발/운영)
- Destination : DevOps Inbound Proxy L4 VIP, Jenkins Master (SKT-DOPSPAP02~04) → 아래 표 참조

구분	Source			Destination			
	Host Name	IP	용도	Host Name	IP	Port	용도
개발				DEVOPS L4 VIP	61.250.19.57	80 443	DevOps Tool 접속 (Bitbucket, Nexus 등)
				SKT-DOPSPAP02	61.250.19.34	10150	Jenkins Master-Slave 통신 (과금영역 업무)
				SKT-DOPSPAP03	61.250.19.35	10150	Jenkins Master-Slave 통신 (POC영역 업무)
				SKT-DOPSPAP04	61.250.19.36	10150	Jenkins Master-Slave 통신 (Swing영역 업무)
운영	(운영 빌드 필요시 가입)			DEVOPS L4 VIP	61.250.19.57	80 443	DevOps Tool 접속 (Bitbucket, Nexus 등)
				SKT-DOPSPAP02	61.250.19.34	10150	Jenkins Master-Slave 통신 (과금영역 업무)
				SKT-DOPSPAP03	61.250.19.35	10150	Jenkins Master-Slave 통신 (POC영역 업무)
				SKT-DOPSPAP04	61.250.19.36	10150	Jenkins Master-Slave 통신 (Swing영역 업무)

#### <배포 방화벽>

- Source : 빌드서버
- Destination : 배포대상서버 ( **FTP Port: 21** )

구분	Source			Destination			
	Host Name	IP	용도	Host Name	IP	Port	용도
개발				SKT-XXXDWAS 등		20 21	Application WAR 또는 Web Resources 배포
운영				SKT-XXXPWAS1		20 21	Application WAR 또는 Web Resources 배포
				SKT-XXXPWAS2			
				SKT-XXXPWEB1 등			

## 3.2. Jenkins Slave Agent 설치하기

Jenkins Slave Agent를 설치할 서버에 Jenkins Slave Agent 용 계정으로 접속 합니다. Jenkins Master 에서 Node 를 생성할때 입력한 Remote Root Directory 로 이동합니다.

아래 4개의 파일을 해당 디렉토리에 저장한 후 스크립트에 변수들을 세팅합니다.

설치파일
<a href="#">agent.jar</a> <a href="#">prepare_SAMPLE.sh</a> <a href="#">startup_SAMPLE.sh</a> <a href="#">shutdown_SAMPLE.sh</a>

수정할 변수값 목록은 아래와 같습니다.

```
> JAVA_HOME= #서버 내 환경에 맞는 경로로 수정
> JENKINS_HOME= #http://devops.sktelecom.com/mybuild, mybuildswg, mybuildrater 중 선택
> NODE_HOME=
> NODE_NAME=
> SECRET=
```

그리고 스크립트 파일명을 수정합니다. 아래 스크립트 파일명의 SAMPLE 문자열을 DevOps 담당자로부터 전달 받은 NODE\_NAME 의 값으로 치환합니다.

> ex) prepare\_SAMPLE.sh → prepare\_\${NODE\_NAME}.sh



NODE\_HOME, NODE\_NAME, SECRET 변수는 Jenkins Master 에 Node 를 생성할 때 입력/생성 되는 값으로 DevOps 담당자에게 메일 또는 쪽지로 요청하여 관련 정보를 회신 받습니다.

### Jenkins Slave Agent 디렉토리 구조

```
$NODE_HOME
+--- scripts
+--- logs // Script
+--- pid // Jenkins Slave Agent pid
+--- workspace // Jenkins JOB , ,
+--- remoting // Jenkins Slave Agent ,
    +--- logs // Jenkins Slave Agent
    +--- jarCache // Jenkins Slave Agent Master jar
```

### prepare\_SAMPLE.sh

```
#!/bin/bash
#Jenkins Slave Agent Home
#DevOps .
NODE_HOME=

SCRIPT_HOME=$NODE_HOME/scripts
FILE_AGENT=agent.jar
FILE_START=startup_SAMPLE.sh
FILE_SHUTDOWN=shutdown_SAMPLE.sh

echo "Jenkins Agent Pre-requisites : [$FILE_AGENT, $FILE_START, $FILE_SHUTDOWN]"
echo "==> 1. make directories "
mkdir -p $SCRIPT_HOME
mkdir -p $NODE_HOME/pid
mkdir -p $NODE_HOME/logs
echo "          : SCRIPT_HOME, NODE_HOME/pid, NODE_HOME/logs are created."

echo "==> 2. move start/stop script into $SCRIPT_HOME/"
mv $FILE_START $SCRIPT_HOME/
mv $FILE_SHUTDOWN $SCRIPT_HOME/
mv $FILE_AGENT $SCRIPT_HOME/
chmod 755 $SCRIPT_HOME/$FILE_START
chmod 755 $SCRIPT_HOME/$FILE_SHUTDOWN
chmod 755 $SCRIPT_HOME/$FILE_AGENT

echo "==> Ready to start Jenkins Agent."
echo "To proceed next step, type command : cd $SCRIPT_HOME"
exit 0
```

#### startup\_SAMPLE.sh

```
#!/bin/bash
export JAVA_HOME= #JAVA_HOME . /bin .
export PATH=$JAVA_HOME/bin:$HOME/.local/bin:$HOME/bin:$PATH:

#Jenkins master URL . mybuild, mybuildswg, mybuildrater
JENKINS_HOME=http://devops.sktelecom.com/mybuild

#Jenkins Slave Agent Home
#DevOps .
NODE_HOME=

#Jenkins Master Node
#DevOps .
NODE_NAME=

#Jenkins Slave Agent log
LOG_DIR=$NODE_HOME/logs
LOG_PATH=$LOG_DIR/jenkins-slave-nohup-$NODE_NAME.log

DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"

#Jenkins Master Node Secret
#DevOps .
SECRET=

PID_PATH=$NODE_HOME/pid/$NODE_NAME.pid
CMD="java -jar agent.jar -jnlpUrl $JENKINS_HOME/computer/$NODE_NAME/slave-agent.jnlp -secret $SECRET -workDir=$NODE_HOME"
#
#CMD="java -jar -Dfile.encoding=UTF-8 agent.jar -jnlpUrl $JENKINS_HOME/computer/$NODE_NAME/slave-agent.jnlp -secret $SECRET -workDir=$NODE_HOME"

echo "Jenkins Agent Started."
cd $DIR
nohup $CMD &> $LOG_PATH&
echo $! > $PID_PATH
```

젠킨스 웹화면에서 한글 로그를 확인하고자 한다면, 위 스크립트의 CMD 변수에 file encoding을 java 명령행 인자로 명시적으로 지정할 수 있습니다.

Agent를 실행할 서버의 운영체제 encoding이 UTF-8인 경우에는 java 명령행 인자를 추가하지 않아도 한글이 출력됩니다.

#### shutdown\_SAMPLE.sh

```
#!/bin/bash
#Jenkins Master Node
#DevOps .
NODE_NAME=

pid="$(ps -fu $USER | grep java | grep $NODE_NAME | grep slave | awk '{print $2}')"
if [ "$pid" == "" ]
then
echo "Slave not running"
exit 1
fi
kill -9 $pid
```

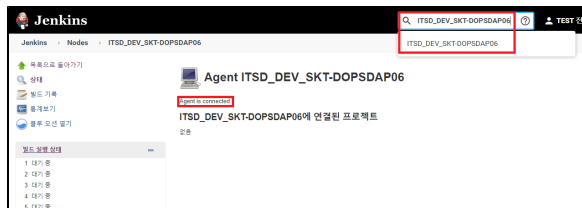
### 3.2.1. Jenkins Slave Agent 구동하기

아래 명령어로 Jenkins Slave Agent 를 실행합니다.  
 prepare\_\${NODE\_NAME}.sh는 최초에 한번만 실행하면 되고, 그 이후에는 startup/shutdown 만 수행하시면 됩니다.

```
$ cd $NODE_HOME
$ ./prepare_${NODE_NAME}.sh
$ cd scripts
$ ./startup_${NODE_NAME}.sh
```

구동 후, Jenkins에 접속하여 우측 상단 검색박스에 Agent명을 입력하여 검색된, Agent명을 선택.

"Agent is connected."메시지가 출력되어있는지 확인 하시면 정상 구동되었는지 확인이 가능합니다.



또는 아래 그림과 같이 좌측 "빌드 실행 상태" 박스에서 확인하고자하는 업무팀 소유의 Agent명을 찾습니다.



아래와 같이 Agent명 옆에 (오프라인)이라는 표시가 없고 대기중인 큐의 정보가 표시되면, Jenkins Master와의 연동이 성공한 것입니다.



### 3.2.2. Jenkins Slave Agent 구동시 오류 조치하기

Error Message	해결방법
java.net.ConnectionException : Connection timed out	방화벽 오픈 여부 또는 DNS 통신 여부를 확인합니다.
java.lang.UnsupportedClassVersionError: hudson/remoting/Launcher : Unsupported major.minor version	JDK 버전 또는 JDK PATH를 확인합니다.

### 3.2.3. Jenkins Slave Agent 중지하기

아래 명령어로 Jenkins Slave Agent 를 중지합니다.

```
$ cd $NODE_HOME/scripts
$ ./shutdown_${NODE_NAME}.sh
```