Jenkins를 이용한 빌드 자동화와 배포

작성자 : 강경미([carami@nate.com)](mailto:carami@nate.com))

root 사용자로 해야할 일

* 일반 계정 생성
* 일반 계정이 접근할 수 있는 디렉토리 생성 및 권한 주기(tomcat, jenkins등 설치 폴더)
* JDK & Maven설치 및 환경설정
* 포트포워딩(80에서 tomcat운영 포트)

일반 계정으로 해야할 일

* Jenkins 설치 및 실행
* mysql에 사용자 계정 생성 및 데이터 베이스 생성
* tomcat 설치 및 실행
* Jenkins 설정

1. 사용자 계정 생성

서버에 root 계정으로 접속하여 사용자 계정을 생성한다.

- 조에서 함께 사용할 계정을 하나 만든다. adduser명령을 통해 만들 수 있다.

- 개인 계정을 만든다.

- 접속 후 암호변경은 passwd 명령을 통여 변경할 수 있다. 대문자, 소문자, 특수문자등을 잘 섞어서 만든다. 1234 이런 암호는 해킹당함.

|  |
| --- |
| adduser 아이디 |

1. 일반 계정이 접근할 수 있는 디렉토리 생성 및 권한 주기(tomcat, jenkins등 설치 폴더)

root 계정으로 application 이 설치될 디렉토리를 생성한 후, 해당 사용자에게 권한을 준다.

- 디렉토리를 생성한다. mkdir 명령

- 조에서 함께 사용할 아이디에게 권한을 준다. chown 명령.

- 목록을 조회한다. ls 명령

|  |
| --- |
| mkdir /apps  chown 아이디.아이디 /apps  ls -la |

1. JDK & Maven 설치

3-1. JDK설치

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

- jdk-8u141-linux-x64.tar.gz 를 다운로드한다.

ftp 를 이용하여 업로드 한다. windows 사용자는 https://filezilla-project.org/ 를 사용한다.

맥의 경우는 다음과 같이 접속하여 업로드 할 수 있다.

|  |
| --- |
| sftp 아이디@ip주소  sftp> bin  sftp> hash  sftp> jdk-8u141-linux-x64.tar.gz |

업로드 이후 다음과 같이 jdk를 설치한다.

- '~/' 는 홈 디렉토리를 말한다. '/home/아이디' 와 같다.

- /apps디렉토리로 이동한 후, 홈디렉토리에 업로드한 파일을 복사해온다. cp는 복사명령

- apps 폴더에서 압축을 해제한다. xvfz옵션은 gz압축을 해제한 후 tar를 풀라는 의미를 가진다.

- 압축이 풀린 jdk1.8.0\_141 폴더를 jdk 로 심볼릭링크를 건다. /apps/jdk는 JAVA\_HOME 경로로 지정할 예정이다.

- ls -la 명령으로 확인한다.

|  |
| --- |
| cd /apps  cp ~/jdk-8u141-linux-x64.tar.gz .  tar xvfz jdk-8u141-linux-x64.tar.gz  ln -s jdk1.8.0\_141/ jdk  ls -la |

3-2. maven 설치

|  |
| --- |
| cd /apps  wget http://mirror.navercorp.com/apache/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz  tar xvfz apache-maven-3.5.0-bin.tar.gz  ln -s apache-maven-3.5.0 maven |

3-3. JDK & maven 환경설정

JAVA\_HOME, CLASSPATH, PATH 환경설정을 추가한다. root 계정으로 수정해야한다.

'su -' 명령은 root사용자로 전환하는 것이다.

|  |
| --- |
| su -  nano /etc/profile |

Nano 에디터 명령

저장 : ctrl+o 후에 엔터

나가기 : ctrl+x 후에 엔터

profile의 맨 아래에 다음을 추가한다.

|  |
| --- |
| JAVA\_HOME=/apps/jdk  CLASSPATH=.:$JAVA\_HOME/lib/tools.jar  PATH=$PATH:$HOME/bin:$JAVA\_HOME/bin  export PATH CLASSPATH JAVA\_HOME |

수정된 내용을 적용하려면 다음과 같이 명령을 내린다. 혹은 재접속한다.

|  |
| --- |
| source /etc/profile |

다음과 같이 jdk가 잘 설치되었는지 확인한다.

|  |
| --- |
| java -version  echo $JAVA\_HOME  echo $PATH  echo $CLASSPATH |

1. Jenkins설치

4-1. 설치

조에서 함께 사용할 아이디(필자는 carami란 아이디를 사용)로 접속, Jenkins 설치

- jenkins 를 다운로드 한다. Generic Java package (.war) 를 다운로드 한다.

- jenkins는 독립적으로도 실행가능하다.

- java -jar 명령으로 실행할 수 있다. '&'는 리눅스에서 백그라운드로 실행하라는 명령이다. 로그아웃해도 실행되고 있다.

|  |
| --- |
| cd /apps  wget http://mirrors.jenkins.io/war-stable/latest/jenkins.war  java -jar jenkins.war & |

4-2. 젠킨스 실행

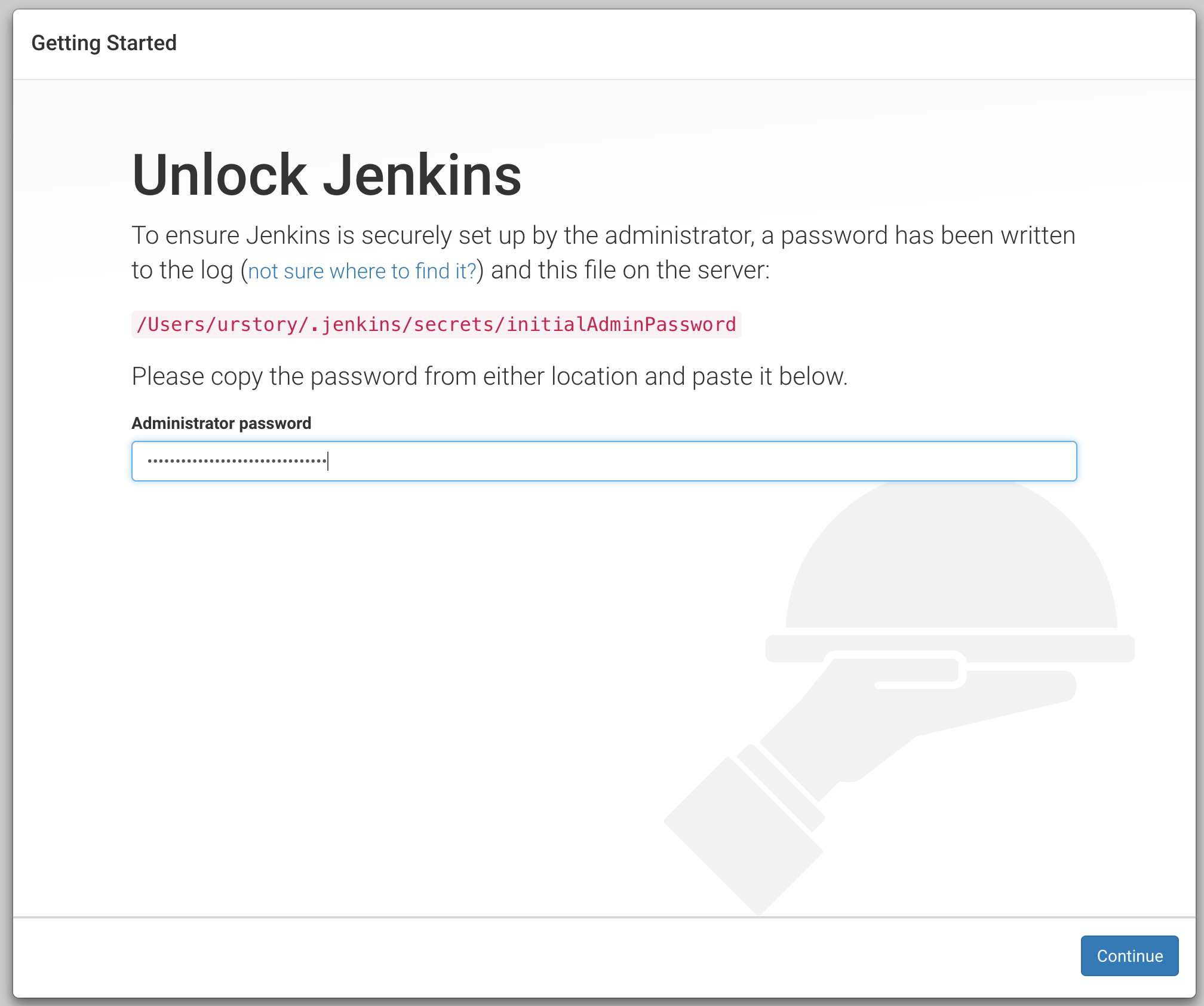
|  |
| --- |
| <http://ip:8080> |

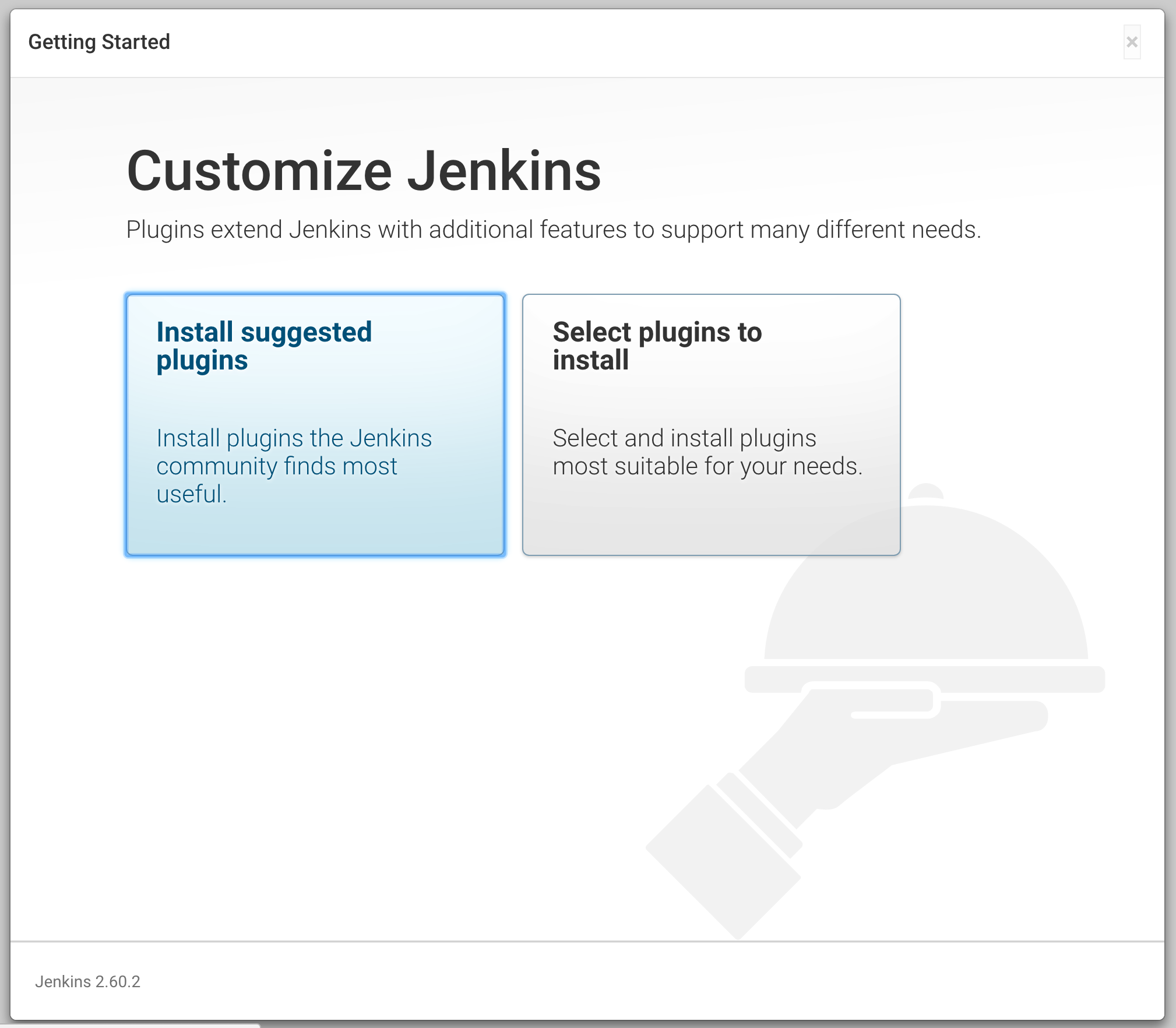
으로 접속할 수 있다.

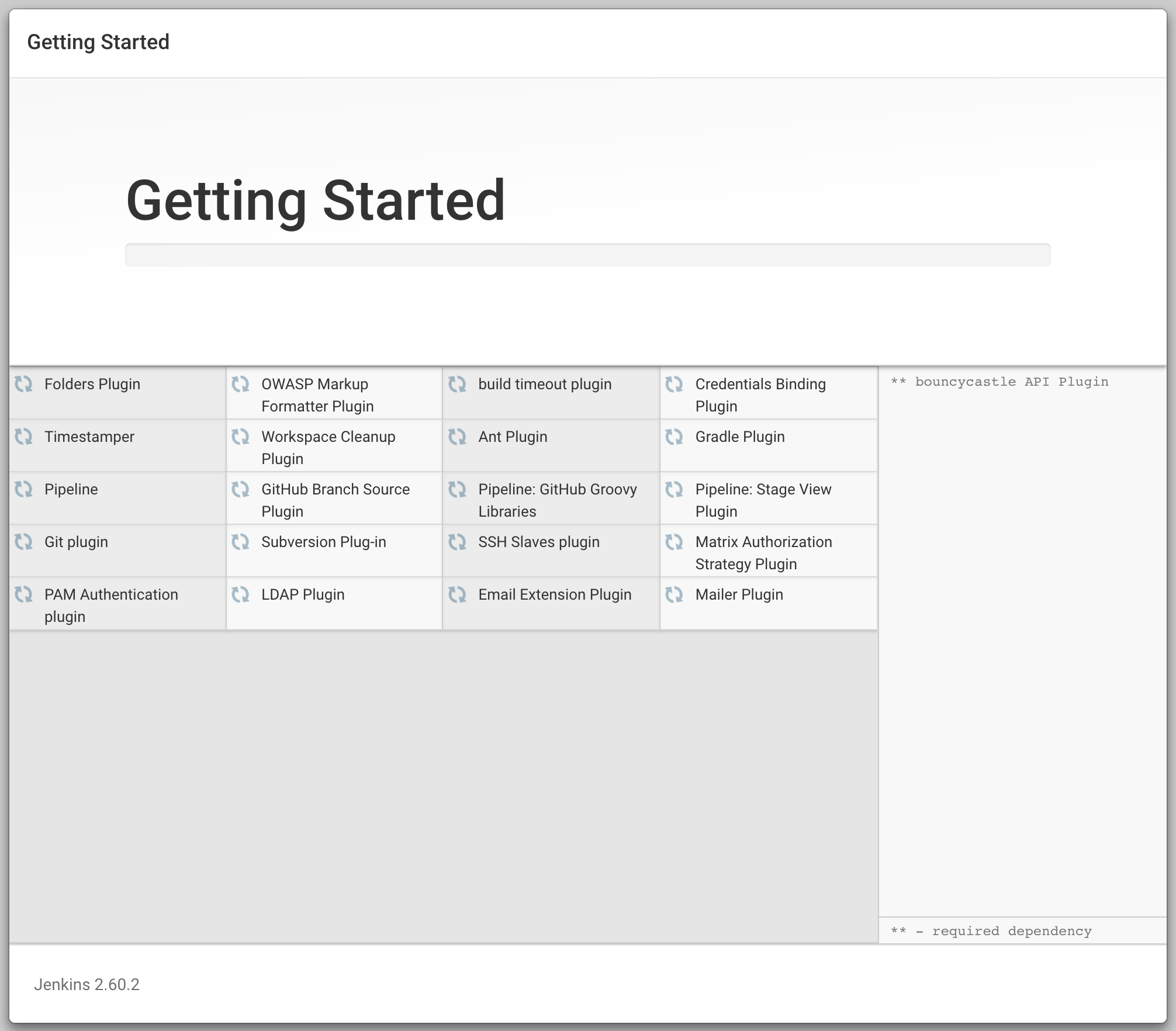
- /home/계정/.jenkins/secrets/initialAdminPassword 에 있는 암호를 입력해야 한다.

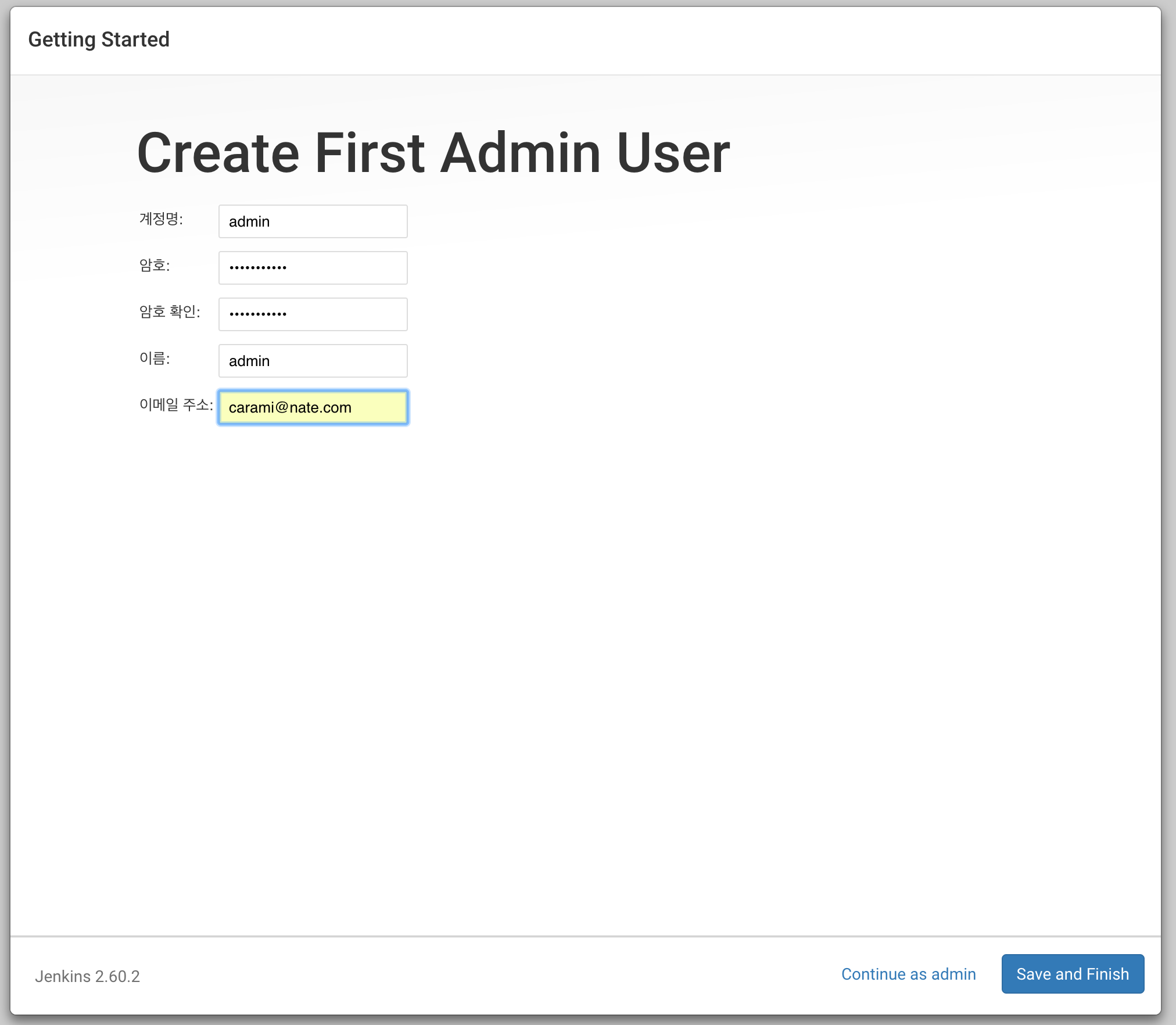
- cat 명령은 파일의 내용을 보는 명령

|  |
| --- |
| cat /home/계정/.jenkins/secrets/initialAdminPassword |

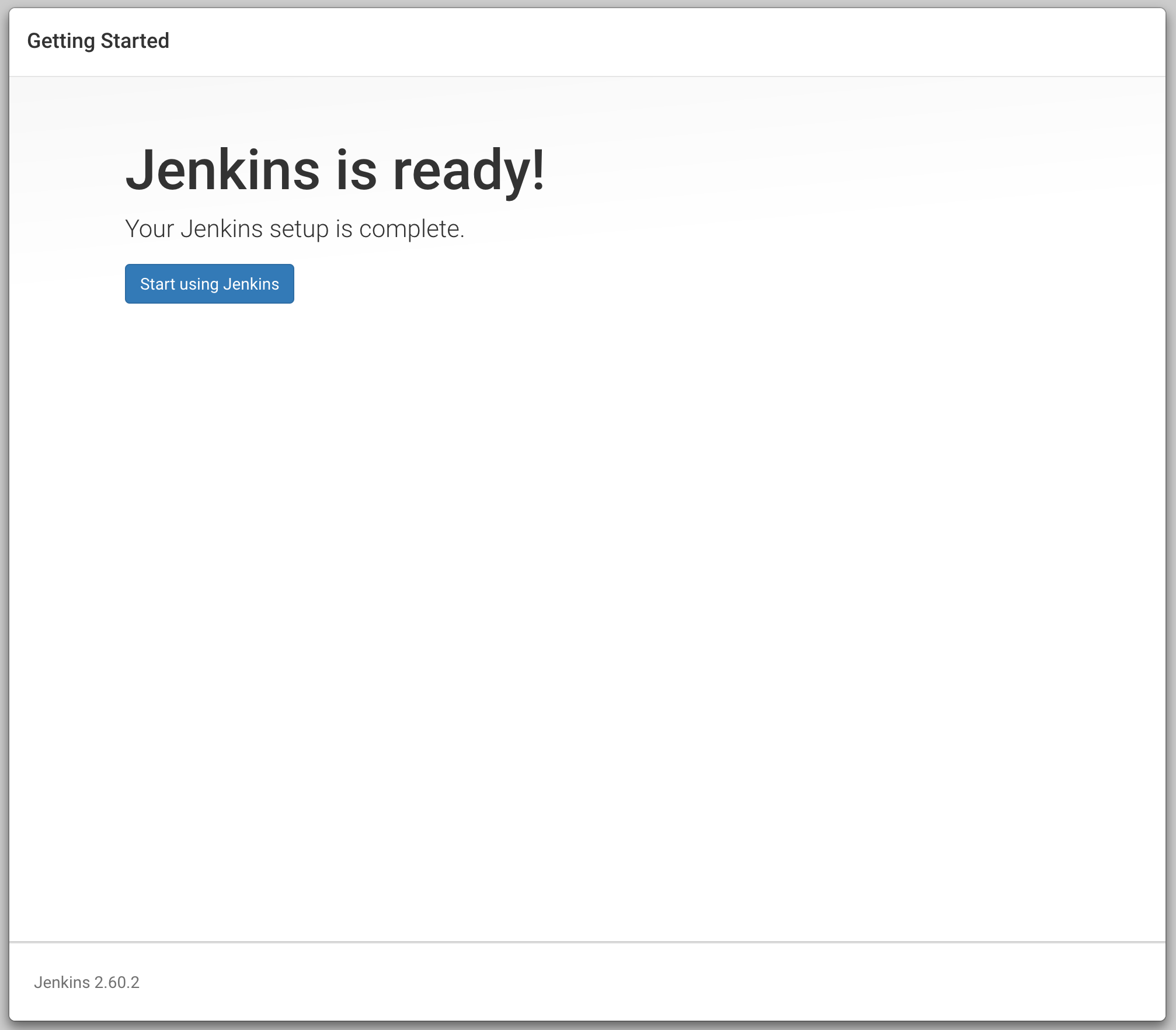


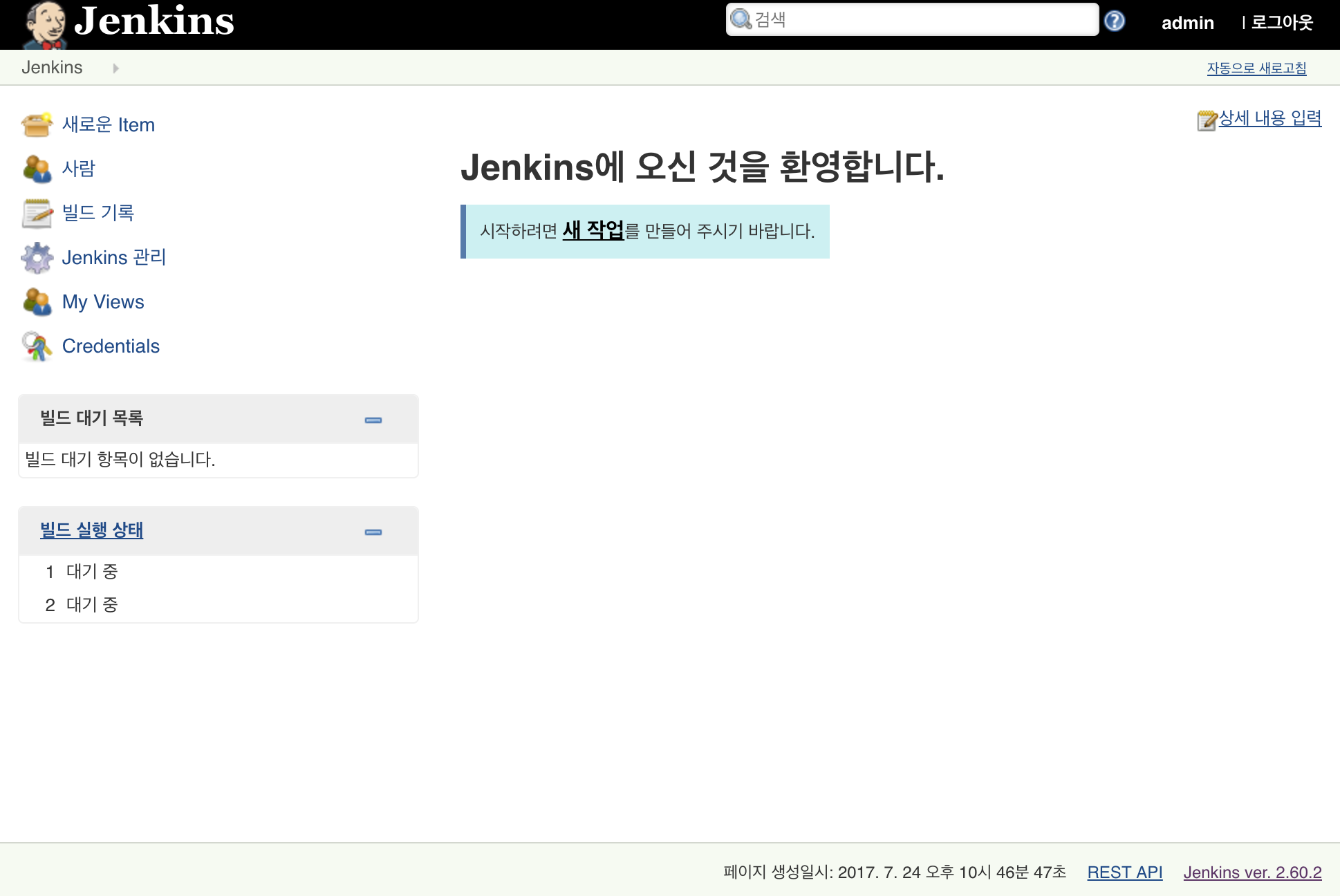






save and finish 버튼을 클릭한다.





4-3. 젠킨스 종료

- ps 명령을 프로세스 목록을 보는 명령

- '|'는 파이프 라고 읽는다. ps의 결과를 파이프 기호 뒷편으로 전달하는 목적으로 사용된다. 표준 출력을 우측프로그램의 표준입력으로 제공

- grep 명령은 입력받은 내용중에서 특정 문자열이 포함된 것을 찾는다.

|  |
| --- |
| ps -lef | grep jenkins |

|  |
| --- |
| [carami@boostcamp-001 ~]$ ps -lef | grep jenkins  0 S carami 1427 871 6 80 0 - 903049 futex\_ 16:23 pts/4 00:00:59 java -jar jenkins.war  0 S carami 2551 2530 0 80 0 - 25814 pipe\_w 16:39 pts/6 00:00:00 grep jenkins |

위와 같이 나올경우 1427, 2551이 프로세스 id이다. 1427을 종료시킨다.

|  |
| --- |
| kill -9 1427 |

1. mysql에 사용자 계정 생성 및 데이터 베이스 생성

- mysql 에 접속한다.

|  |
| --- |
| mysql -uroot mysql |

암호를 대문자,소문자,특수문자, 숫자등으로 잘 만들지 않을 경우 다음과 같은 오류가 발생한다.

Your password does not satisfy the current policy requirements

|  |
| --- |
| create user 'carami'@localhost identified by '암호';  create user 'carami'@'%' identified by '암호';  create database tododb;  GRANT ALL on tododb.\* TO 'carami'@'localhost';  GRANT ALL on tododb.\* TO 'carami'@'%';  flush privileges; |

1. tomcat 설치

|  |
| --- |
| cd /apps  wget http://apache.mirror.cdnetworks.com/tomcat/tomcat-8/v8.5.16/bin/apache-tomcat-8.5.16.tar.gz  tar xvfz apache-tomcat-8.5.16.tar.gz  ln -s apache-tomcat-8.5.16 tomcat |

- jenkins 배포를 위한 tomcat 설정 파일 수정

tomat의 실행 포트를 수정하기 위해 tomcat/conf/server.xml파일을 수정한다.

nano server.xml

|  |
| --- |
| <Connector port="8080" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" /> |

위와 같은 부분을 찾아 다음과 같이 수정한다.

|  |
| --- |
| <Connector port="9090" protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443" URIEncoding="UTF-8"/> |

- webapps 폴더에 있는 기본 웹 어플리케이션들을 삭제한다. 빌드한 웹 어플리케이션을 jenkins에서 ROOT.war 파일로 복사하여 설치할 예정이다.

|  |
| --- |
| cd /apps/tomcat/webapps  rm –rf \*  ls -la |

- tomcat의 실행

현재 경로의 파일을 실행하려면 리눅스에서는 './' 를 앞에 써줘야 한다.

|  |
| --- |
| cd /apps/tomcat/bin  ./startup.sh |

- tomcat의 종료

|  |
| --- |
| cd /apps/tomcat/bin  ./shutdown.sh |

1. 80 -> 9090 으로 포트포워딩

참고문서 : https://blog.outsider.ne.kr/580

1024 이하로 동작하는 어플리케이션을 실행하기 위해서는 root권한이 필요하다. tomcat을 일반 사용자 계정으로 실행하기 위하여 9090포트로 설정하였다. 그런데 <http://ip> 로 웹 어플리케이션이 호출되기 위해서는 80포트로 동작해야한다. 이런 문제를 해결하기 위해서 root사용자로 포트포워딩 설정을 한다. 80으로 오는 요청을 9090으로 전달하는 것이다.

아래의 명령은 root권한으로 실행해야 한다.

|  |
| --- |
| iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080 |

확인은 다음 명령을 이용한다.

|  |
| --- |
| iptables -t nat -L |

1. jenkins를 이용하여 빌드 및 배포하기

참고문서 : http://handcoding.tistory.com/25

http://jojoldu.tistory.com/139

먼저 nono /apps/tomcat/deploy.sh 명령을 이용하여 shell 스크립트를 작성한다.

/apps/tomcat/bin 으로 시작하는 tomcat프로세스가 없다면, /home/carami/.jenkins/workspace/todo/target/todo-0.0.1-SNAPSHOT.war 파일을 /apps/tomcat/webapps/ROOT.war 로 복사한다. (아직 todo-0.0.1-SNAPSHOT.war파일이 존재하지 않다.)

그리고 나서 /apps/tomcat/bin/startup.sh 을 실행하여 톰캣을 실행한다. BUILD\_ID=dontKillMe에 대한 설명은 <http://lng1982.tistory.com/178> 를 참고.

이미 tomcat이 실행중이라면 tomcat프로세스를 종료한 후, 위의 작업을 진행한다. /home/carami 경로를 본인에 맞게 알맞게 수정한다.

|  |
| --- |
| #!/bin/sh  if [ -z "`ps -eaf | grep java|grep /apps/tomcat/bin`" ]; then  cp /home/carami/.jenkins/workspace/todo/target/todo-0.0.1-SNAPSHOT.war /apps/tomcat/webapps/ROOT.war  BUILD\_ID=dontKillMe /apps/tomcat/bin/startup.sh  else  ps -eaf | grep java | grep /apps/tomcat/bin | awk '{print $2}' |  while read PID  do  echo "Killing $PID ..."  kill -9 $PID  echo  echo "Tomcat is being shutdowned."  done  cp /home/carami/.jenkins/workspace/todo/target/todo-0.0.1-SNAPSHOT.war /apps/tomcat/webapps/ROOT.war  BUILD\_ID=dontKillMe /apps/tomcat/bin/startup.sh  fi |

해당 스크립트가 실행될 수 있도록 퍼미션을 설정한다.

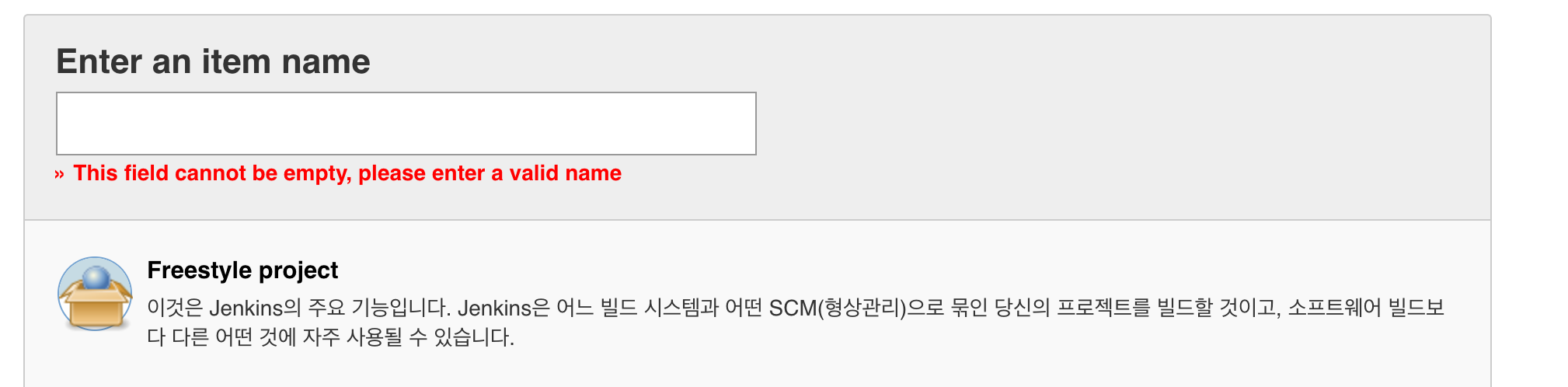
* chmod는 퍼미션을 변경하는 명령이다. 자세한 내용은 찾아보세요.

|  |
| --- |
| chmod 755 deploy.sh |

* Jenkins에서 Item 추가하기

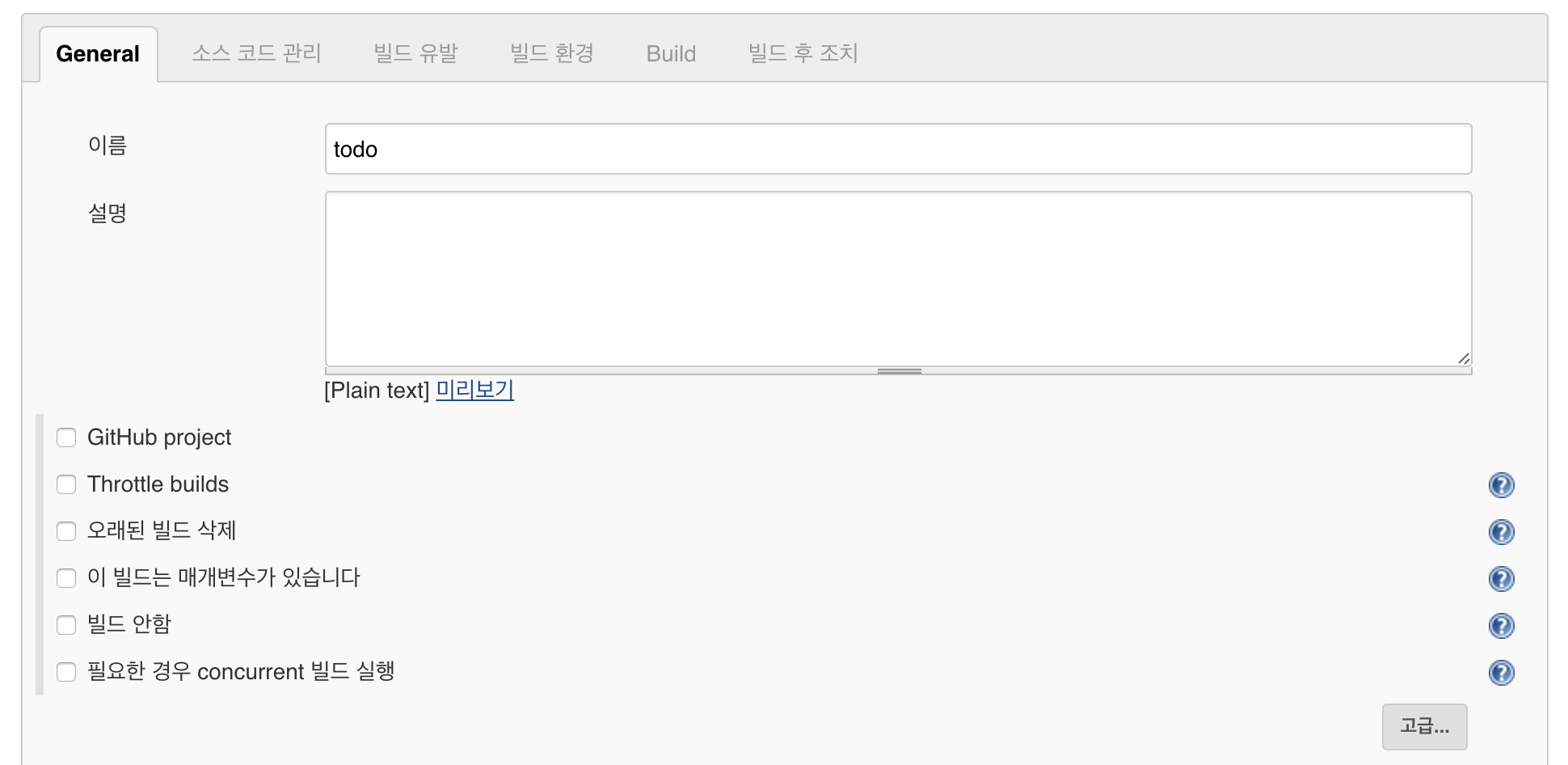


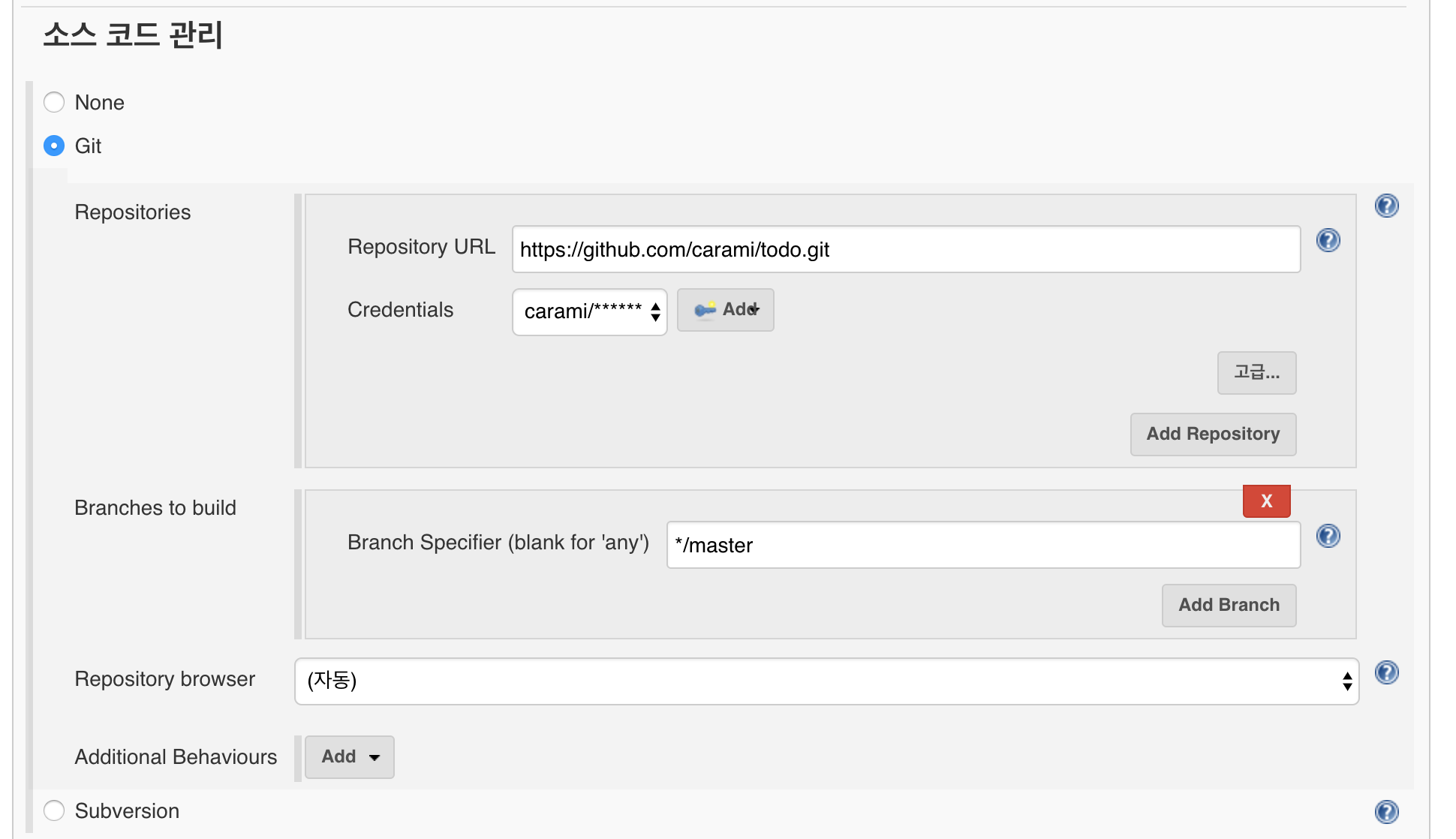
새로운 Item을 클릭한다.



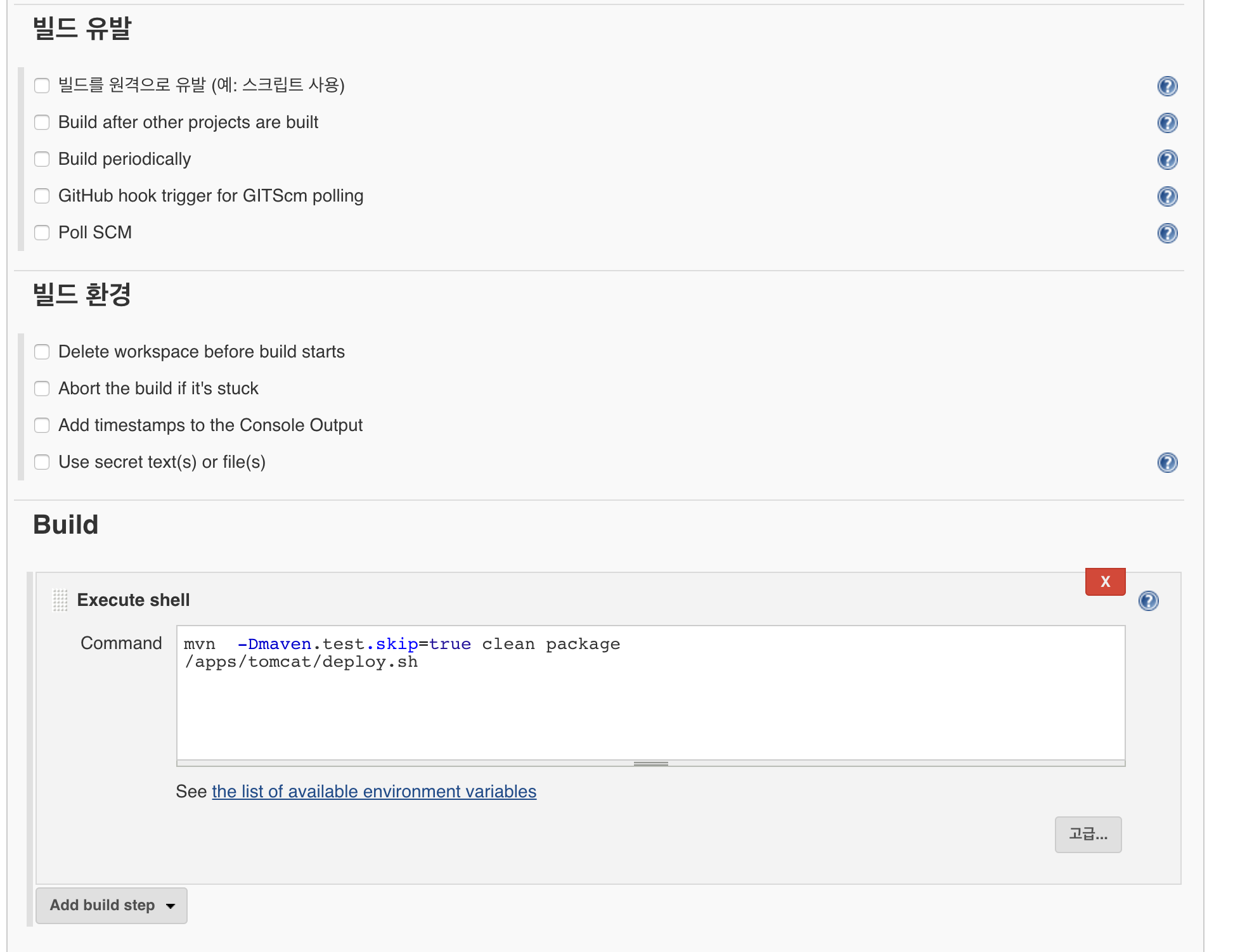
item name에 프로젝트 이름을 적는다. 필자는 todo 라고 입력함. 그리고 나서 아래에 있는 Freestyle project를 클릭한 후 하단의 OK버튼을 클릭한다.

아래의 이미지와 같이 설정한다.





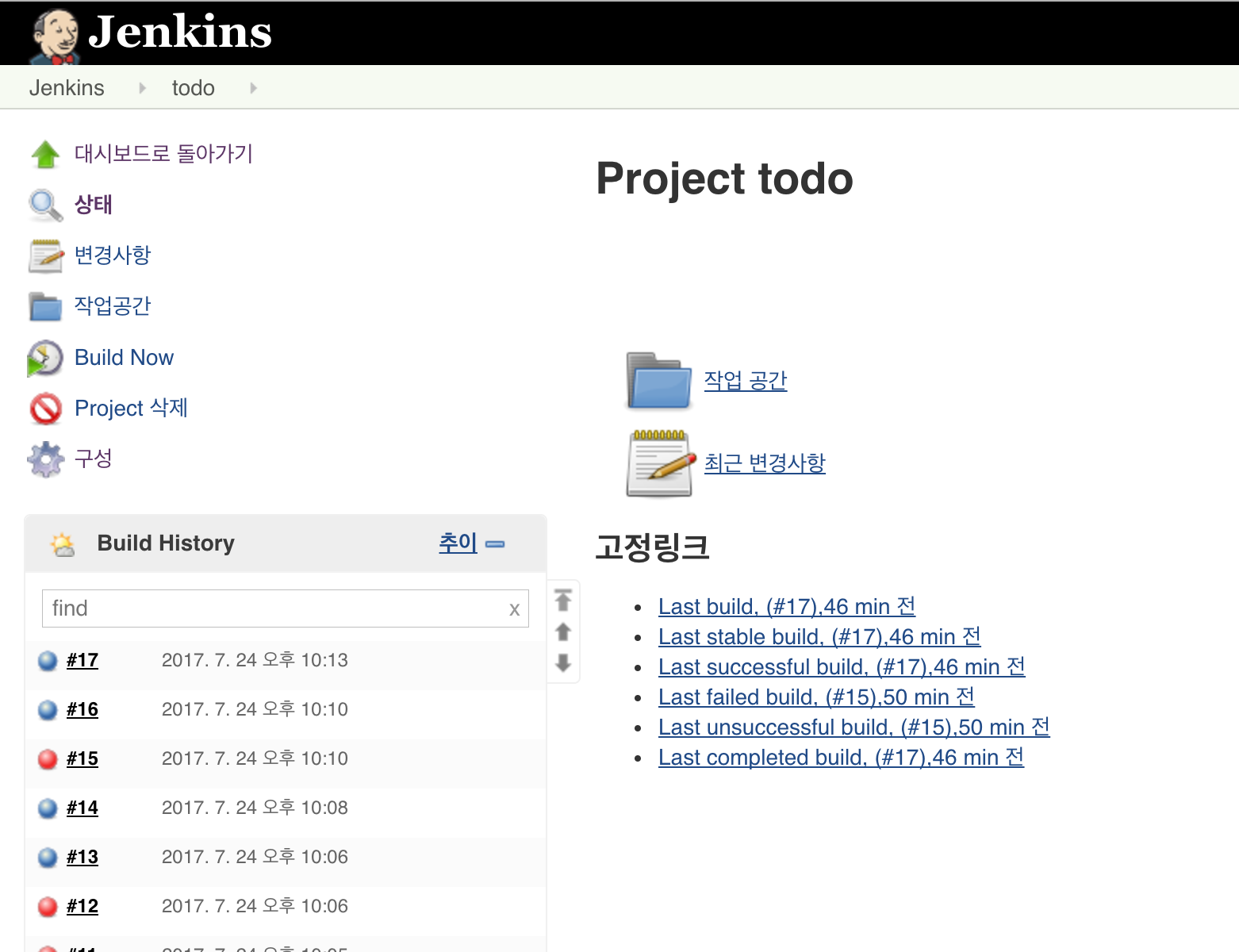
github URL을 적고 하단의 Add버튼을 클릭하여 github id/password를 입력한다. checkout할 branch를 적는다. 여기에서는 master를 적었다.



빌드를 위해 maven을 실행하였다. –Dmaven.test.skip=true는 test를 실행하지 말라는 옵션이다. 필요에 따라서 true를 false로 설정하도록 한다.

앞에서 작성한 deploy.sh를 실행하도록 하였다.

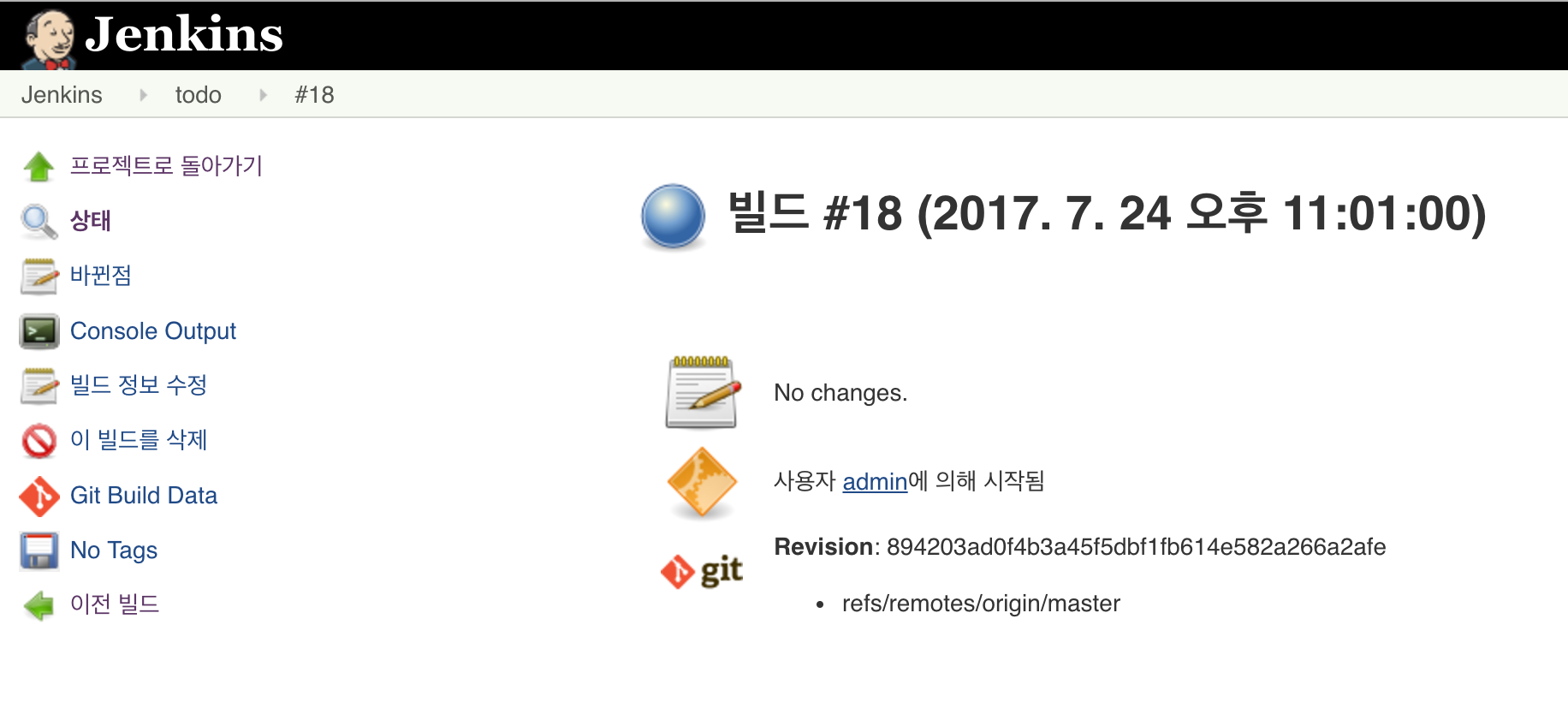
cp /home/carami/.jenkins/workspace/todo/target 에서 jenkins는 빌드를 수행한다.



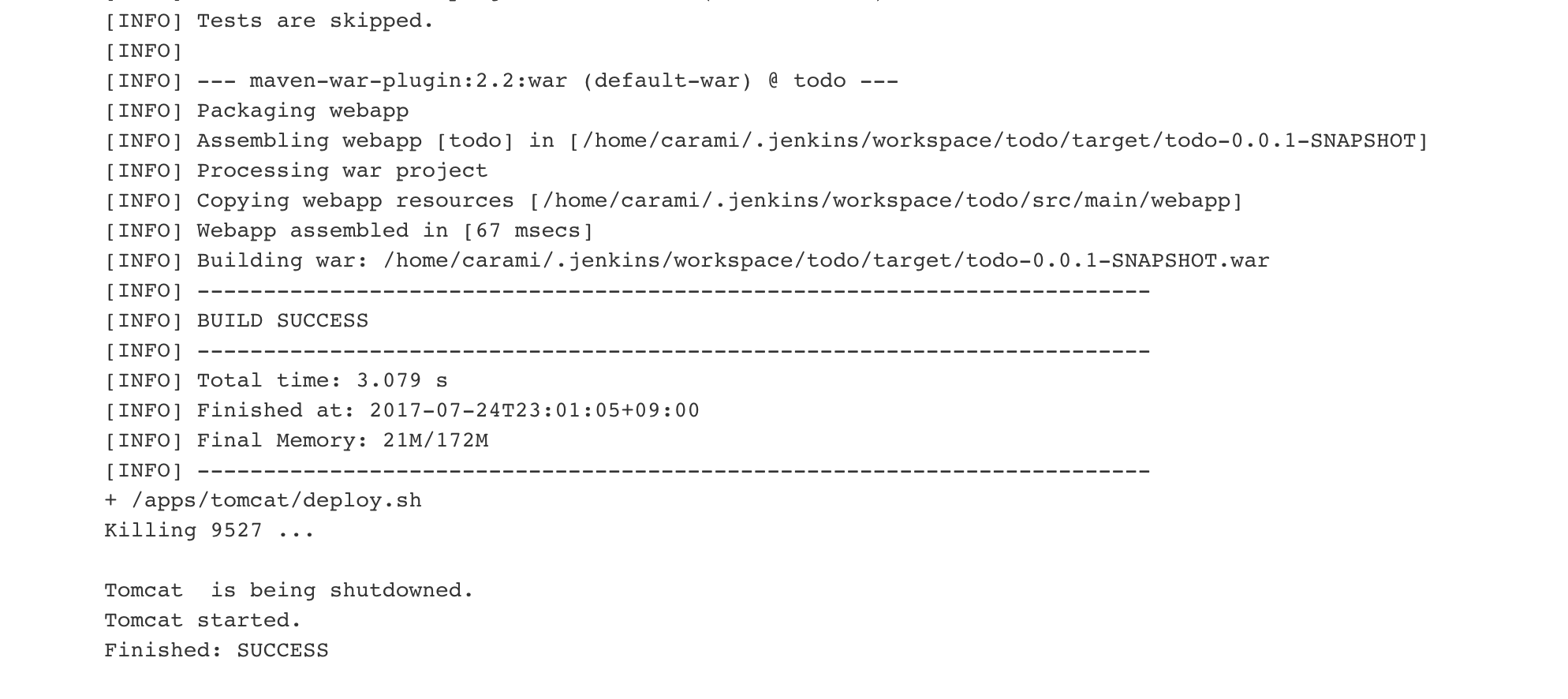
위의 화면은 17번 빌드를 수행한 화면이다. 빌드를 수행하고 싶다면 “Build Now”버튼을 클릭한다.



빌드를 수행하면 위와 같이 #18이라는 새로운 빌드가 수행된 것을 알 수 있다. #18 이라는 링크를 클릭해보자.



위와 같이 어떤 브랜치의 내용이 빌드되었는지 확인된다. Console Output을 클릭한다.



빌드 과정이 출력된 것을 알 있다. 하단에 보면 9527 pid를 가지고 있는 프로세스가 종료되었고, Tomcat이 다시 시작된 것을 알 수 있다. 소스코드를 수정하고 git에 push한 후 빌드를 수행하여 제대로 반영되는지 확인해 본다. <http://ip> 로 확인