**소개**

1. JAVA 기반 빌드 툴로서 JAVA에 적합하게 만들어져 있고 자동화에 유리함.
2. 자바 소스 컴파일, 파일이나 폴더의 이동 및 복사, 삭제, jar, war, ear, zip 파일의 생성. DB 스크립트 실행, 외부 프로그램 실행 등 언급한 것 외에도 여러가지 작업이 가능.

**선수조건**

1. Ant를 지원하는 툴의 존재
2. 빌드할 소스코드의 존재

**필요도구**

1. Ant
2. Eclipse
3. Tomcat

**목적**

1. **Ant도구를 이용하여 웹프로젝트를 패키지 구조에 맞게 war파일로 생성한다.**
2. **생성한 war파일을 원격지로 FTP를 이용하여 upload.**

**실행방법**

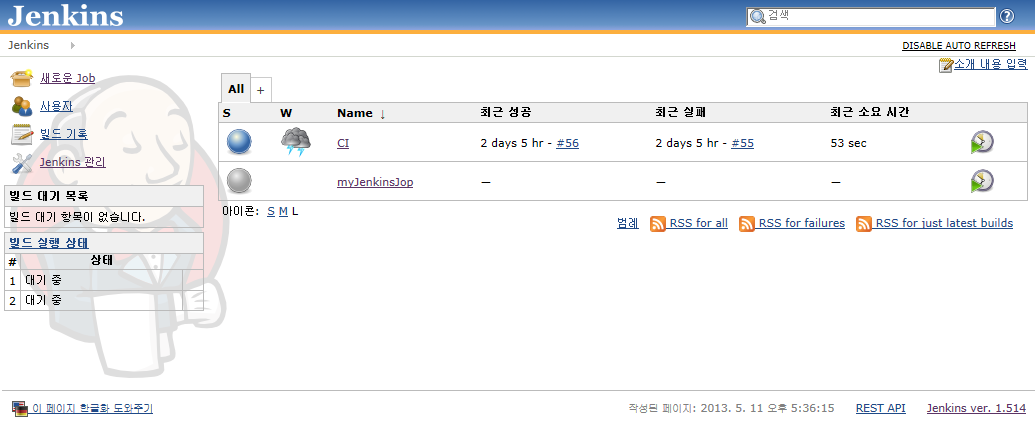


**<http://jenkins-ci.org/>**

**로 접속하여 Jenkins를 다운로드 한다. 본문에서는war버전을 이용하여 테스트 하도록 한다. 인스톨러버전으로 설치할때는 별다른 설정이 없이 Next를 진행하면 Jenkins가 설치되고 Jenkins라는 이름으로 윈도우 서비스에 등록된다. 해당 서비스가 실행될 때 localhost:8080으로 접속하게 되면 젠킨스 홈페이지로 이동한다.**



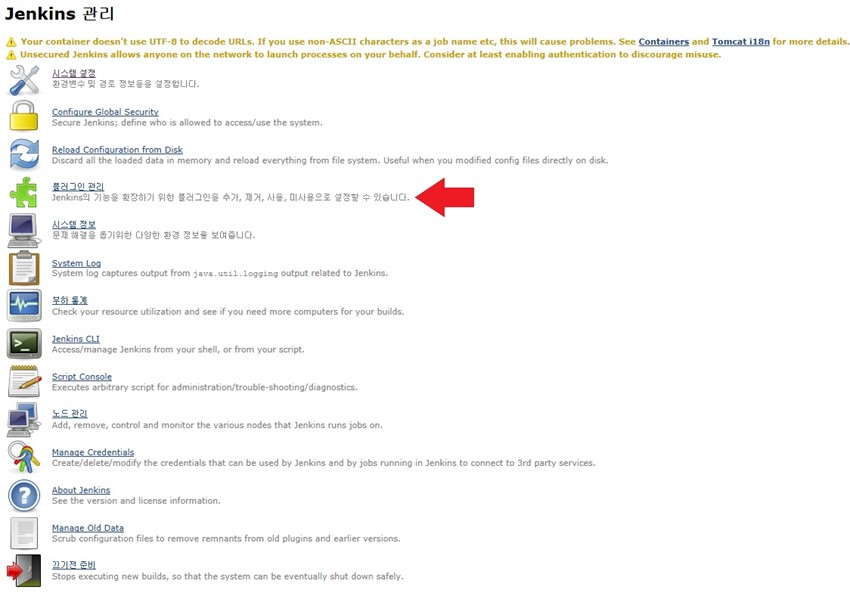
1. Jenkins.war를 톰캣설치경로(이하 ${CATALINA\_HOME})/webapps에 위치 시키고 ${CATALINA\_HOME}/bin/startup.bat을 이용하여 톰캣을 구동한다.



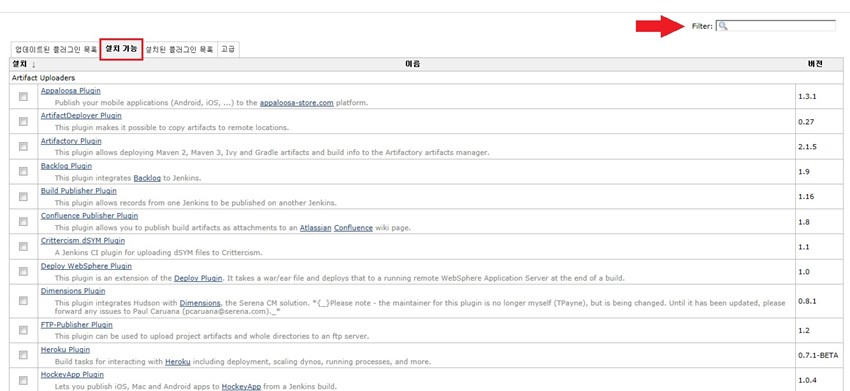
1. Jenkins가 Deploy되고 웹 서비스가 시작되면 localhost:8080/jenkis로 접속하여 정상적으로 구동되는지 확인한다. (필자는 두개의 테스트 Job을 가지고 있어서 표시되었다.)

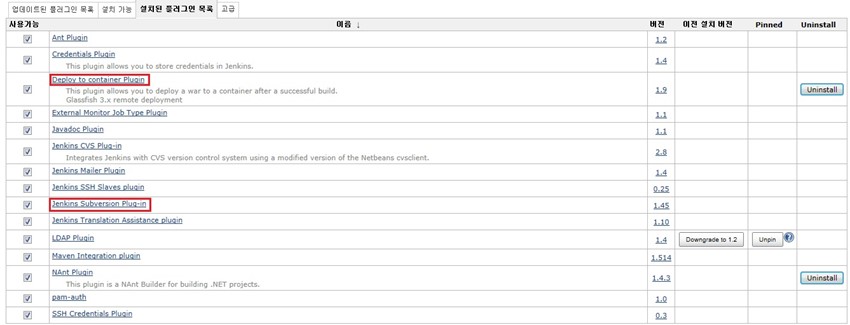


1. 구동이 확인 되었으면 홈페이지에서 **Jenkins 관리**로 이동한다.

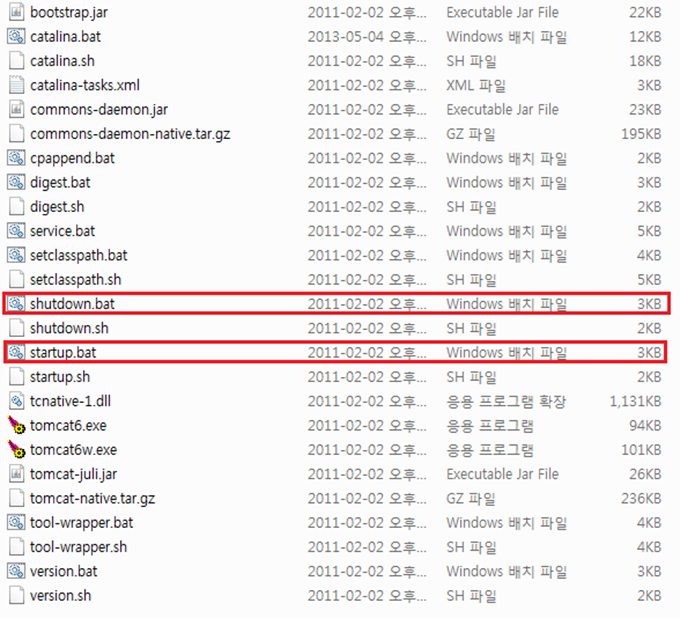


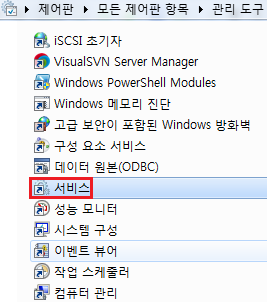
1. **플러그인 관리** 로 이동한다.

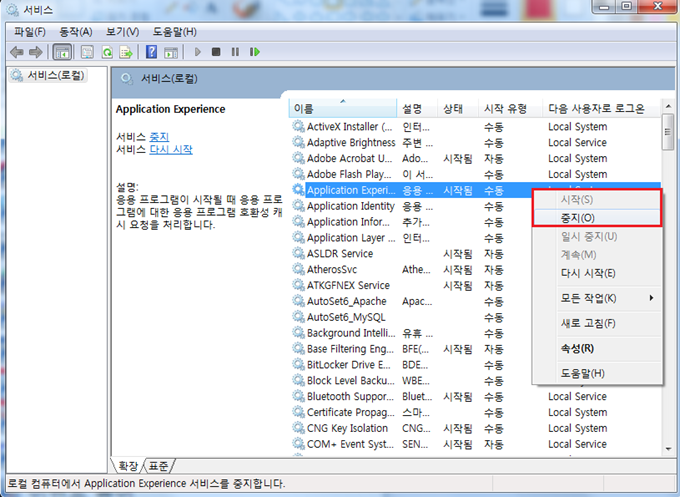




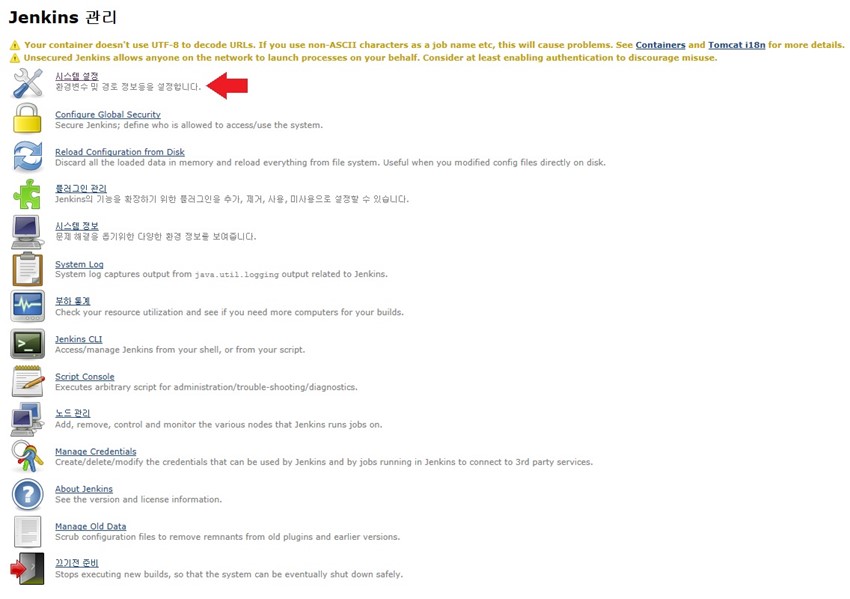
1. SVN에서 업데이트내역을 확인하고Ant와 연동하여 자동으로 빌드 한 후, 원격지의 톰캣서버에 war를 배포하기 위해 몇 가지 플러그인을 설치할 필요가 있다. **설치 가능**탭으로 이동하여 Deploy to container plugin, Jenkins Subversion Plug-in을 설치한다.



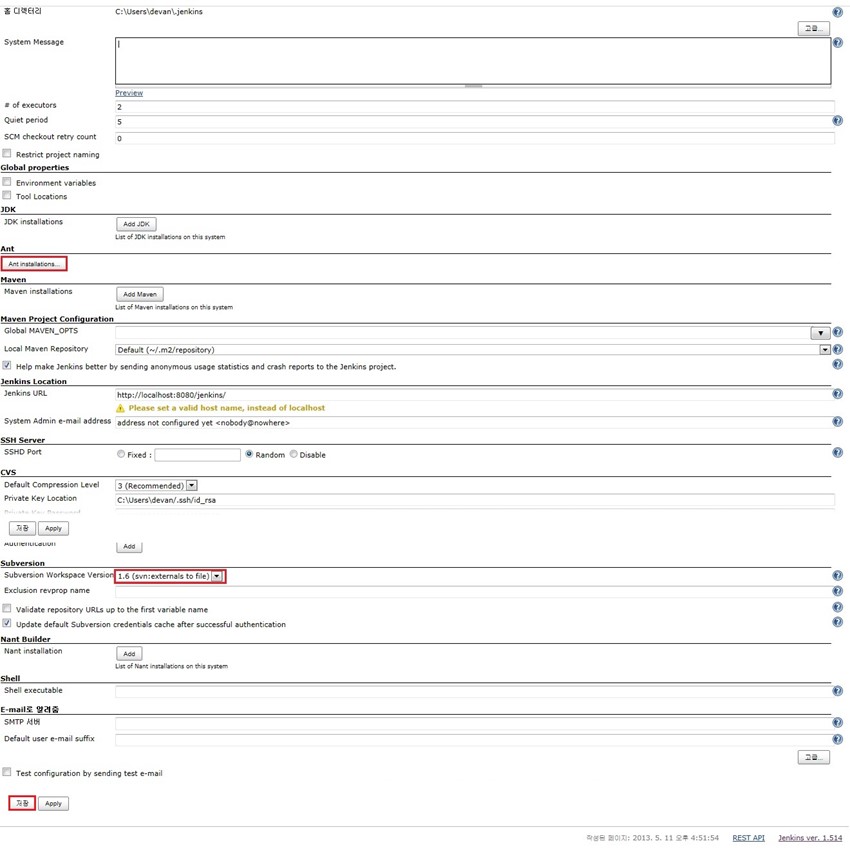




1. 플러그인 설치가 완료되면 Jenkins를 재시작해준다. 톰캣의 경우 ${CATALINA\_HOME}/bin/shutdown.bat을 이용하여 톰캣을 종료한 후 다시 ${CATALINA\_HOME}/bin/startup.bat을 실행시켜 톰캣을 시작한다. 설치형 Jenkins의 경우 **시작 > 제어판 > 관리도구 > 서비스**로 이동하여 **Jenkins 서비스**를 **우클릭 > 중지** 한 후 중지가 확인되었으면 다시 **Jenkins서비스**를 **우클릭 > 시작** 하여 재구동한다. (필자의 경우 Jenkins를 war형으로 설치하였으므로 윈도우 서비스로 등록되지 않아 임의의 서비스를 통해 중지와 시작을 스크린샷하였다. 윈도우 설치형 Jenkins를 사용하는 유저라면 Jenkins라는 서비스가 시작됨 상태로 구동중일것이다.)



1. Jenkins홈으로 이동하여 다시 Jenkins관리 > 시스템 설정으로 이동한다.



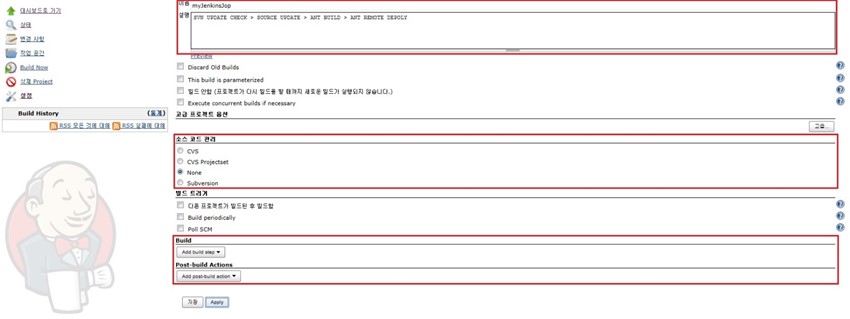
1. 기본 설정을 위해 위의 빈칸들을 채워넣고 저장을한다. 별다른 설정을 할 필요는 없고 Ant설정과 Subversion만 1.6버전으로 되어있는지 확인하면 된다. 안되어 있다면 콤보박스에서 1.6으로 선택하고 저장한다. 위의 Ant Installation버튼을 선택하면 다음과 같은 폼이 추가 된다.



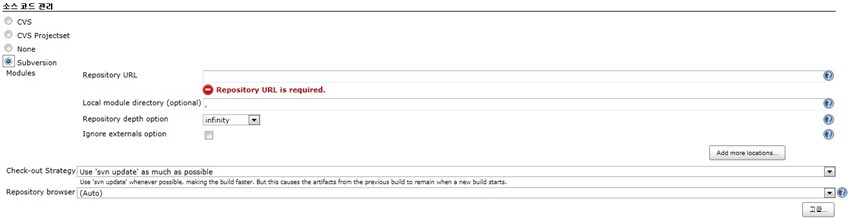
1. Name은 젠킨스 프로젝트 내에서 지정될 해당 안트의 명이다. ANT\_HOME으로 현재 안트가 설치되어있는 경로를 지정하고 저장하면 된다.



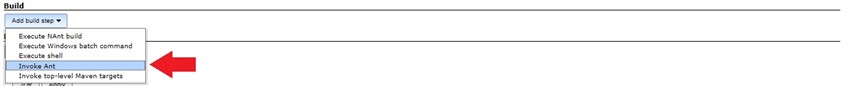
1. 다시 홈(이하 대쉬보드)으로 이동하여 **새로운 Job**으로 이동하면 신규 Job의 명과 형태를 지정하는 화면이 출력된다. Build a free-style software project를 선택하고 명은 자신이 원하는 이름으로 지정한 뒤 OK 한다.



1. OK를 하게 되면 좀 더 상세한 정보들을 입력할 수 있는 폼들이 추가된다. 우리가 필요한 것은 SVN과 Build 그리고 Post-build Actions 이다.



1. 소스 코드 관리 화면에서 Subversion을 선택하면 위 처럼 폼이 추가된다. 빌드 대상이 될 소스가 존재하는 SVN설정 정보들을 입력한다. SVN에 사용자 계정정보가 필요할 경우 로그인 폼으로 이동하게 된다. 어려울 것 없이 SVN계정정보로 로그인하듯이 저장하면 된다. Check-out Strategy의 User 'svn update'as much as possible과 Repository URL을 제외한 부분은 지금 테스트 케이스에선 필요하지 않으므로 생략한다.



1. SVN설정을 완료하였으면 Build설정으로 이동한다. Add build step을 선택하면 콤보리스트가 나오며 우리는 현재 Ant로 build할것이므로 Invoke Ant를 선택한다. 선택하면 아래와같이 폼이 추가된다.

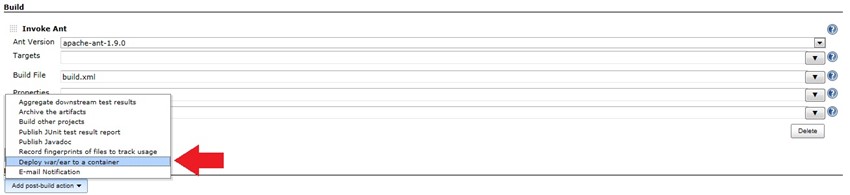




1. Ant의 build정보를 입력하는 부분이다. 우측 하단의 고급을 클릭하여 폼을 추가한다. Ant Version은 시스템설정에서 지정한 Ant이름이 출력된다. 선택하면 된다.



1. 추가된 폼중에서 Build File을 지정하는 부분이 있다. Ant의 빌드스크립트인 build.xml을 지정하는 부분이며 이곳에 들어올 경로는 젠킨스의 Job에서 지정한 workspace의 루트부터이다. 따라서 build.xml만 지정한다면 ${WORKSPACE\_HOME}/build.xml을 참조하게 될것이다. 필자의 경우에는 젠킨스의 Repository인 C:\Users\devan\.jenkins 경로에서 \jobs\${JOB\_NAME}\workspace\build.xml을 참조하게 될것이다.



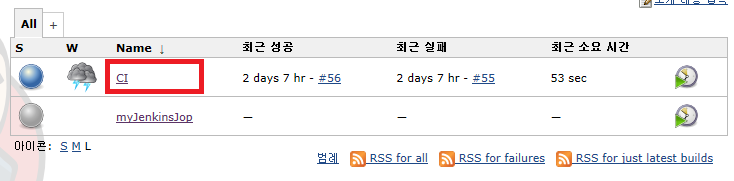
1. Build 정보를 입력하였으면 Build가 끝나고나서 진행할 Post-build Actions를 지정한다. 우리는 원격지에 배포하는 것이 목적이므로 Deploy war/ear to a container를 선택한다. 위에서 플러그인을 설치하지 않았다면 콤보박스 리스트에 나오지 않으므로 반드시 플러그인을 설치하고 진행하여야한다.



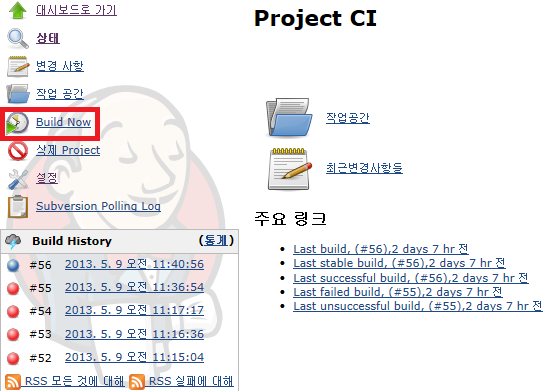
1. WAR/EAR files는 \*\*/\*.war를 선택한다. 모든 .war파일들을 대상으로 원격지에 배포하겠다는 뜻이 된다. 선택 후 추가되는 폼은 위와 같다. 현재 apache-tomcat-6.0.32로 테스트하고 있으므로 Tomcat6.x를 선택해준다.



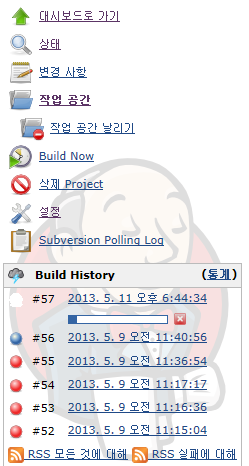
1. Manager user name과 Manager password, Tomcat URL등으로 폼이 변경된다. 해당 username과password는 배포 대상 tomcat 에서 manager-gui와 manager-script권한을 할당받고 있는 유저의 아이디와 패스워드로 기입하여야한다. 배포 대상 톰캣에서 conf/tomcat-users.xml을 참조하면 된다.



1. 설정을 완료하였으면 저장하고 젠킨스 메인페이지 이동한다. 위처럼 새로운 Job이 등록된 것을 확인할 수 있다. 좌측이 가장 최근 build상태이다. 파란색은 성공, 회색은 없음, 붉은색은 실패 등이다. 새로 등록된 Job의 Name을 클릭하여 이동한다.



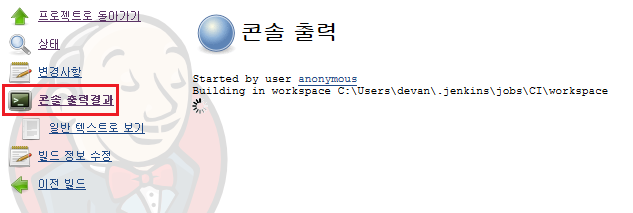
1. 작업의 상태나 수동빌드, 설정등을 지정할 수 있다. 바로 Build Now를 실행해보자.



1. 좌측의 Build History가 추가되면서 뭔가 진행되고 있음을 알려준다.



1. 해당 빌드를 클릭하면 상세정보를 확인할 수 있다. 아직 빌드되고있으므로 정보가 없고 우상단에 진행되고있음을 알리고있다.



1. 좌측의 콘솔 출력결과를 클릭하면 실시간으로 콘솔에 출력되는 상황을 확인할 수 있다.



1. 갱신된 소스로 빌드가 성공적으로 종료되었고 원격 배포도 성공하였다.

**결론**

1. CI를 사용하여 SVN 소스 체크, 빌드, 원격 배포를 자동화 하였다.