Installation via Maven WAR Overlay

This page describes how to install Jenkins using a Maven WAR Overlay method.

Why?

While Jenkins provides extremely easy installation and execution (e.g., `java -jar jenkins.war`), sometimes more complicated setups are required. It is very common to want to deploy the `jenkins.war` into an existing Apache Tomcat installation. This is the exact scenario that will be described here.

How?

The preferred mechanism is to create a Maven project which can be used to build and deploy Jenkins to your Apache Tomcat. It utilizes the `maven-war-plugin` and `cargo-maven2-plugin` to build and deploy.

You need to create a project hierarchy with the `pom.xml` and `context.xml` as shown below. Once this is in place, you need to update the `JENKINS_URL` to be the URL for your Tomcat instance. Now, the only thing you have to do is run `mvn clean package cargo:redeploy` and watch the Maven magic.

Directory Hierarchy

```
project/
  pom.xml
  src/
    main/
      webapp/
        META-INF/
          context.xml
```

pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.acme.infra</groupId>
  <artifactId>jenkins-war</artifactId>
  <version>1.414</version>
  <packaging>war</packaging>

  <repositories>
    <repository>
      <id>m.g.o-public</id>
      <url>http://maven.glassfish.org/content/groups/public/</url>
    </repository>
  </repositories>

  <pluginRepositories>
    <pluginRepository>
      <id>m.g.o-public</id>
      <url>http://maven.glassfish.org/content/groups/public/</url>
    </pluginRepository>
  </pluginRepositories>

  <dependencies>
    <dependency>
      <groupId>org.jenkins-ci.main</groupId>
      <artifactId>jenkins-war</artifactId>
      <version>${project.version}</version>
      <type>war</type>
      <scope>runtime</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.codehaus.cargo</groupId>
        <artifactId>cargo-maven2-plugin</artifactId>
        <version>1.1.0</version>
        <configuration>
          <container>
            <containerId>tomcat7x</containerId>
            <type>remote</type>
          </container>
          <type>runtime</type>
          <properties>
            <cargo.tomcat.manager.url>http://JENKINS_URL/manager/text</cargo.tomcat.manager.url>
            <cargo.remote.username>tomcatadmin</cargo.remote.username>
            <cargo.remote.password>tomcatadmin</cargo.remote.password>
          </properties>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
Troubleshooting

- This manual presumes you are using Tomcat 7.x, if not, you should update the cargo plugin configuration accordingly.
- Make sure that `mvn clean package` builds the war file, if not, you probably have some sort of network issue or you're not using the right Jenkins version number.
- Make sure that you have configured the Tomcat 7.x manager application, and the updated `tomcat-users.xml` which split out roles very different than Tomcat 6.x.
- Make sure that you have the correct tomcat manager username and password.