

EPAM Systems, RD Dep., RD Dep.
MTN.*NIX.09
Continuous Delivery with Jenkins

MTN.*NIX.09 Continious Delivery with Jenkins
Software build principles

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
<1.0>	Initial revision	Aleksandr Mazurenko	<25-Jun-2017>		

Legal Notice

This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM Systems.

Task 1. Maven build

Review

Implement Maven build script for sample project.

Task

1. Install Maven

```
[root@jenkins home-task]# mvn -v
Apache Maven 3.6.2 (40f52333136460af0dc0d7232c0dc0bcf0d9e117; 2019-08-27T15:06:16Z)
Maven home: /root/maven/apache-maven-3.6.2
Java version: 1.8.0_232, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.232.b09-0.el7_7.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-957.27.2.el7.x86_64", arch: "amd64", family: "unix"
[root@jenkins home-task]#
```

2. Clone GitHub project

[git@github.com:MNT-Lab/build-t00ls.git](https://github.com/MNT-Lab/build-t00ls.git)

3. Create your own branch based on master by pattern:

First char of name + surname

<https://github.com/MNT-Lab/build-t00ls/tree/ayanchuk>

4. Implement pom.xml script to clean, compile, test, create jar file and distribution artifact. (source code in 'src' repo).

```
<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/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.test</groupId>
```

```
<artifactId>home-task</artifactId>
```

```
<packaging>jar</packaging>
```

```
<version>1.0-SNAPSHOT</version>
```

```
<properties>
```

```
<maven.compiler.release>1.8</maven.compiler.release>
```

```
</properties>
```

```
<build>
```

```
<plugins>
```

```
<plugin>
```

```
<groupId>org.apache.maven.plugins</groupId>
```

```
<artifactId>maven-jar-plugin</artifactId>
```

```
<version>3.2.0</version>
```

```
<configuration>
```

```
<archive>
```

```
<manifest>
```

```
<addClasspath>true</addClasspath>
```

```
<classpathPrefix>lib/</classpathPrefix>
```

```
<mainClass>com.test.Project</mainClass>
```

```
</manifest>
```

```

    </archive>
  </configuration>
</plugin>
<plugin>
  <artifactId>maven-clean-plugin</artifactId>
  <version>3.1.0</version>
  <executions>
    <execution>
      <id>auto-clean</id>
      <phase>initialize</phase>
      <goals>
        <goal>clean</goal>
      </goals>
    </execution>
  </executions>
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <version>2.3.1</version>
  <configuration>
    <outputDirectory>distribution</outputDirectory>
  </configuration>
</plugin>
</plugins>
</build>
<name>maven-home-task</name>
<dependencies>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-collections4</artifactId>
    <version>4.1</version>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
</dependencies>
</project>

```

5. All changes should be tracked by commits to own branch

```
commit b638b37710eaabbd554a135b9dce15b72dd9b381
Author: Anton Yanchuk <anton_yanchuk@epam.com>
Date: Tue Nov 26 17:21:28 2019 +0300

    change distribution

commit 8a10e9e12e97fd60e8fa8315ac0f85d12f49ef3b
Author: Anton Yanchuk <anton_yanchuk@epam.com>
Date: Tue Nov 26 17:15:14 2019 +0300

    improve maven tasks

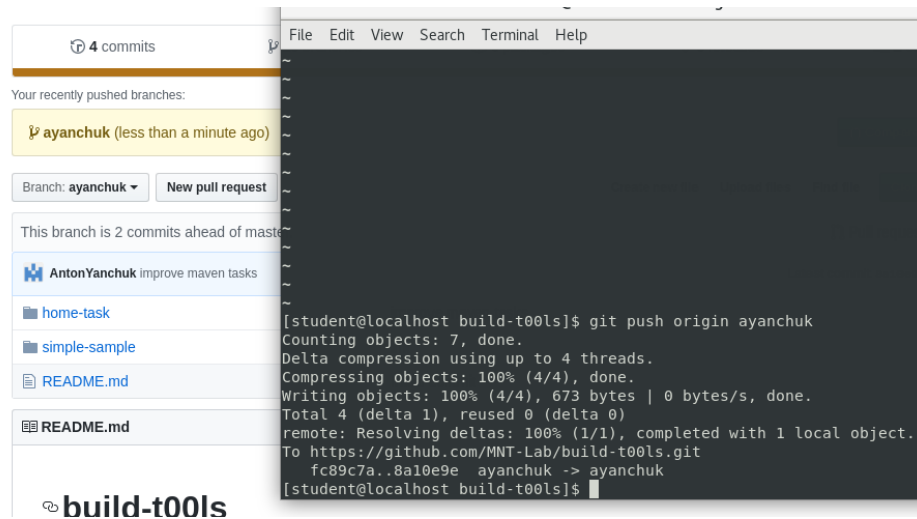
commit fc89c7ab9863774e25c076650af8f04ebaae00f1
Author: Anton Yanchuk <anton_yanchuk@epam.com>
Date: Tue Nov 26 14:42:03 2019 +0300

    second commit

commit 3e91c30b78aed6620b5059660a7e6c0bb53a28ad
Author: Aleksandr Mazurenko <neo.katso@gmail.com>
Date: Tue Nov 26 10:53:21 2019 +0300

    init
```

6. Push to GitHub



The screenshot shows a GitHub repository interface on the left and a terminal window on the right. The GitHub page displays the repository 'build-t00ls' with a branch named 'ayanchuk'. It shows the commit history, including a commit titled 'improve maven tasks' by AntonYanchuk. The terminal window shows the command 'git push origin ayanchuk' being executed, with output indicating successful push to GitHub.

7. Setup Jenkins job ("helloworld-sivanou-maven-build"):

a) build from repository (poll SCM)

Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Repository browser (Auto)

Additional Behaviours

☐ Multiple SCMs

☐ Subversion

Build Triggers

☒ Build whenever a SNAPSHOT dependency is built

☐ Schedule build when some upstream has no successful builds

☐ Trigger builds remotely (e.g., from scripts)

☐ Build after other projects are built

☒ Build periodically

☐ GitHub hook trigger for GITScm polling

☒ Poll SCM

Schedule

b) use maven plugin

Available Jenkins plugins (test results to be published)			
<input checked="" type="checkbox"/>	Mailer Plugin	1.29	<input type="button" value="Uninstall"/>
This plugin allows you to configure email notifications for build results			
<input checked="" type="checkbox"/>	Maven Integration plugin	3.4	<input type="button" value="Uninstall"/>
This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (JUnit, ...).			

c) run jar and output save to file 'output.log'

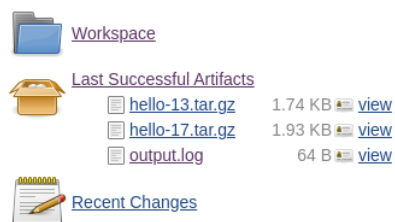
d) archive and attach jar file

ARTEFACT_NAME=hello-\$BUILD_NUMBER.tar.gz

The screenshot shows the 'Post Steps' section of a Jenkins configuration page. At the top, there are fields for 'Root POM' (home-task/pom.xml) and 'Goals and options' (clean install). Below these is a 'Post Steps' section with three radio button options: 'Run only if build succeeds', 'Run only if build succeeds or is unstable', and 'Run regardless of build result'. The second option is selected. Below the options is a section for 'Execute shell' with a text area containing the command: `java -jar home-task/distribution/home-task-1.0-SNAPSHOT.jar >> output.log`. Below the text area is a link to 'See the list of available environment variables'. At the bottom of the 'Execute shell' section is an 'Advanced...' button. Below this is another 'Execute shell' section with a text area containing the command: `tar -czvf hello-$BUILD_NUMBER.tar.gz home-task/distribution/home-task-1.0-SNAPSHOT.jar`. Below the text area is another 'Advanced...' button.

e) attach log file to build (archive artifact)

The screenshot shows the 'Build Settings' and 'Post-build Actions' sections of a Jenkins configuration page. In the 'Build Settings' section, the 'E-mail Notification' checkbox is checked. Below it is a 'Recipients' field with the value 'anton_yanchuk@epam.com'. There are three unchecked checkboxes: 'Send e-mail for every unstable build', 'Send separate e-mails to individuals who broke the build', and 'Send e-mail for each failed module'. In the 'Post-build Actions' section, there is an 'Archive the artifacts' section with a 'Files to archive' field containing the value '*log, *.tar.gz'. Below the field is an 'Advanced...' button.



Report:

Email pattern: [MNT-CD-3]-FirstName-LastName

Contains: GitHub link, Plugin config, Job config, output (screenshot)