构建工具集成

1. 集成 maven

1.1 先决条件

JDK：在 maven3.3 以上的版本需要 JDK 版本 1.7+。内存：没有最低限制。
磁盘：1G+可用磁盘空间。 操作系统：没有限制。
下载 maven：http://maven.apache.org/download.cgi

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.0-bin.tar.gz	apache-maven-3.6.0-bin.tar.gz.sha512	apache-maven-3.6.0-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.0-bin.zip	apache-maven-3.6.0-bin.zip.sha512	apache-maven-3.6.0-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.0-src.tar.gz	apache-maven-3.6.0-src.tar.gz.sha512	apache-maven-3.6.0-src.tar.gz.asc
Source zip archive	apache-maven-3.6.0-src.zip	apache-maven-3.6.0-src.zip.sha512	apache-maven-3.6.0-src.zip.asc

1.2 安装 maven

```
tar zxf apache-maven-3.6.0-bin.tar.gz -C /usr/local/
#设置全局变量（/etc/profile）
export MAVEN_HOME=/usr/local/apache-maven-3.6.0
export PATH=$PATH:$MAVEN_HOME/bin
source /etc/profile
```

测试:

```
[root@VM_7_14_centos ~]# mvn -v
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5ffb20918da4f719f3; 2018-10-25T02:41:47+08:00)
Maven home: /usr/local/apache-maven-3.6.0
Java version: 1.8.0_201, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-1.el6_10.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "2.6.32-696.el6.x86_64", arch: "amd64", family: "unix"
```

1.3 Jenkins 配置 maven

系统设置->全局工具配置

编写 Jenkinsfile

```
node {
    stage ("build"){
        mavenHome = tool 'M3'
        sh "${mavenHome}/bin/mvn -v"
    }
}
```

构建测试

Console Output

```
Started by user admin
Replayed #4
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/cxy/cxy-wlck-ui_TEST
[Pipeline] {
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] tool
[Pipeline] sh
+ /usr/local/apache-maven-3.6.0/bin/mvn -v
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5ffb20918da4f719f3; 2018-10-25T02:41:47+08:00)
Maven home: /usr/local/apache-maven-3.6.0
Java version: 1.8.0_201, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-
1.el6_10.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "2.6.32-696.el6.x86_64", arch: "amd64", family: "unix"
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

到此 jenkins 集成 maven 就完成了。

1.4 maven 常用命令

clean install -DskipTests
clean package

2. 集成 Ant

2.1 先决条件

下载：https://ant.apache.org/bindownload.cgi

- 1.10.5 .zip archive: apache-ant-1.10.5-bin.zip [PGP] [SHA1] [SHA512]
- 1.9.14 .zip archive: apache-ant-1.9.14-bin.zip [PGP] [SHA512]
- 1.10.5 .tar.gz archive: apache-ant-1.10.5-bin.tar.gz [PGP] [SHA1] [SHA512]
- 1.9.14 .tar.gz archive: apache-ant-1.9.14-bin.tar.gz [PGP] [SHA512]
- 1.10.5 .tar.bz2 archive: apache-ant-1.10.5-bin.tar.bz2 [PGP] [SHA1] [SHA512]
- 1.9.14 .tar.bz2 archive: apache-ant-1.9.14-bin.tar.bz2 [PGP] [SHA512]
- 1.10.5 .tar.xz archive: apache-ant-1.10.5-bin.tar.xz [PGP] [SHA1] [SHA512]

2.2 安装 Ant

```
tar zxf apache-ant-1.10.5-bin.tar.gz -C /usr/local/
#添加全局变量（/etc/profile）
export ANT_HOME=/usr/local/apache-ant-1.10.5
export PATH=$PATH:$MAVEN_HOME/bin:$ANT_HOME/bin
source /etc/profile
```

测试:

```
[root@VM_7_14_centos apache-ant-1.10.5]# ant -version
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-1.8.0-openjdk-
1.8.0.201.b09-1.el6_10.x86_64/lib/tools.jar
Apache Ant(TM) version 1.10.5 compiled on July 10 2018
```

2.3 Jenkins 配置 Ant

系统设置->全局工具配置

编写 Jenkinsfile

```
node {
    stage ("build"){
        antHome = tool 'ANT'
        sh "${antHome}/bin/ant -version"
    }
}
```

构建测试

```
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (antbuild)
[Pipeline] tool
[Pipeline] sh
+ /usr/local/apache-ant-1.10.5/bin/ant -version
Unable to locate tools.jar. Expected to find it in /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.201.b09-
1.el6_10.x86_64/lib/tools.jar
Apache Ant(TM) version 1.10.5 compiled on July 10 2018
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

到此 ant 的集成就完成了

2.4 Ant 常用命令

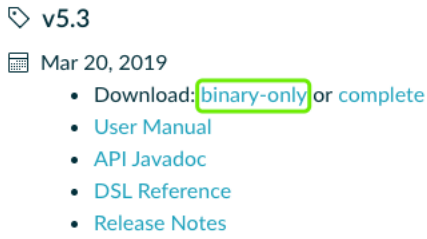
ant -buildfile -f build.xml

3. 集成 Gradle

3.1 先决条件

下载：https://gradle.org/releases/

```
https://downloads.gradle.org/distributions/gradle-5.3-bin.zip
```



3.2 安装 Gradle

```
unzip gradle-5.3-bin.zip -d /usr/local/
#添加全局变量（/etc/profile）
export GRADLE_HOME=/usr/local/gradle-5.3
export PATH=$PATH: $GRADLE_HOME/bin
source /etc/profile
```

测试:

```
[root@VM_7_14_centos ~]# gradle -v

Welcome to Gradle 5.3!

Here are the highlights of this release:
 - Feature variants AKA "optional dependencies"
 - Type-safe accessors in Kotlin precompiled script plugins
 - Gradle Module Metadata 1.0

For more details see https://docs.gradle.org/5.3/release-notes.html


------------------------------------------------------------
Gradle 5.3
------------------------------------------------------------

Build time:   2019-03-20 11:03:29 UTC
Revision:     f5c64796748a98efdbf6f99f44b6afe08492c2a0

Kotlin:       1.3.21
Groovy:       2.5.4
Ant:          Apache Ant(TM) version 1.9.13 compiled on July 10 2018
JVM:          1.8.0_201 (Oracle Corporation 25.201-b09)
OS:           Linux 2.6.32-696.el6.x86_64 amd64
```

3.3 Jenkins 配置 gradle

系统设置->全局工具配置

编写 Jenkinsfile

```
node {
    stage ("gradlebuild"){
        gradleHome = tool 'GRADLE'
        sh "${gradleHome}/bin/gradle -v"
    }
}
```

构建测试：

```
[Pipeline] stage
[Pipeline] { (gradlebuild)
[Pipeline] tool
[Pipeline] sh
+ /usr/local/gradle-5.3/bin/gradle -v
------------------------------------------------------------
Gradle 5.3
------------------------------------------------------------

Build time:   2019-03-20 11:03:29 UTC
Revision:     f5c64796748a98efdbf6f99f44b6afe08492c2a0

Kotlin:       1.3.21
Groovy:       2.5.4
Ant:          Apache Ant(TM) version 1.9.13 compiled on July 10 2018
JVM:          1.8.0_201 (Oracle Corporation 25.201-b09)
OS:           Linux 2.6.32-696.el6.x86_64 amd64

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

到此 jenkins 配置 gradle 完成

3.4 Gradle 常用命令

./gradlew -v 版本号，首次运行，没有 gradle 的要下载的哦。
./gradlew clean 删除 HelloWord/app 目录下的 build 文件夹
./gradlew build 检查依赖并编译打包
./gradlew assembleDebug 编译并打 Debug 包
./gradlew assembleRelease 编译并打 Release 的包
./gradlew installRelease Release 模式打包并安装
./gradlew uninstallRelease 卸载 Release 模式包

4. 集成 Npm

4.1 先决条件

下载 node: https://nodejs.org/en/download/

Windows Installer (.msi)	32-bit	64-bit	
Windows Binary (.zip)	32-bit	64-bit	
macOS Installer (.pkg)	64-bit		
macOS Binary (.tar.gz)	64-bit		
Linux Binaries (x64)	64-bit		
Linux Binaries (ARM)	ARMv6	ARMv7	ARMv8
Source Code	node-v10.15.3.tar.gz		

4.2 安装 Node

```
tar xf node-v10.15.3-linux-x64.tar.xz -C /usr/local/
#添加全局变量（/etc/profile）
export NODE_HOME=/usr/local/node-v10.15.3-linux-x64
export PATH=$PATH: $NODE_HOME/bin
source /etc/profile
```

测试：

```
[root@VM_7_14_centos ~]# node -v
v10.15.3
[root@VM_7_14_centos ~]# npm -v
6.4.1
```

4.3 Jenkins 配置 Npm

在 Jenkins 全局工具配置中并没有 node，可以直接通过 Jenkinsfile 定义使用。

Jenkinsfile：

```
node {
    stage ("npmbuild"){
        sh """
            export npmHome=/usr/local/node-v10.15.3-linux-x64
            export PATH=\$PATH:\$npmHome/bin
            npm -v
            """
    }
}
```

构建测试:

```
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (npmbuild)
[Pipeline] sh
+ export npmHome=/usr/local/node-v10.15.3-linux-x64
+ npmHome=/usr/local/node-v10.15.3-linux-x64
+ export PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/node-v10.15.3-linux-x64/bin
+ PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/node-v10.15.3-linux-x64/bin
+ npm -v
6.4.1
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

4.4 Npm 常用构建命令

npm install && npm run build

总结：