



## Maven Build Life Cycle

- `mvn clean` : to clean local repo
- `mvn compile` : to compile the code
- `mvn test` : to apply test cases
- `mvn package` : to prepare a Package
- `mvn install` : to install a file local repo [.m2]
- `mvn deploy` : to deploy a file remote server / Global repos (nexus)

## Example POM

```
<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>BalajiApp</groupId>
  <artifactId>BalajiApp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>samvit</name>
</project>
```

- It should be noted that there should be a single POM file for each project
- All POM files require the project element and three mandatory fields : **group id, artifact id, version.**
- Projects notation in repository is **groupid:artifactid:version**
- Root element of pom.xml is project and it has three major sub-nodes: properties, dependences & build

## Maven Install

- `#apt-get update`
- `#apt-get install openjdk-7-jdk -y`
- `Apt-get install tomcat7 -y`
- `#apt-get install maven -y (or)`
- `#apt-get install maven2 -y`
- `#mvn -version`

# Distribution Installation

← → ↻ 🔒 https://maven.apache.org/repository/ ☆ S O | 🌐



Apache / Maven / Maven Central Repository 📄

Download | Get Sources | Last Published: 2018-12-24

Project Roles

How to Contribute

Getting Help

Issue Management

Getting Maven Source

The Maven Team

PROJECT DOCUMENTATION

Project Information ➤

MAVEN PROJECTS

Archetype

Artifact Resolver

Doxia

JXR

Maven

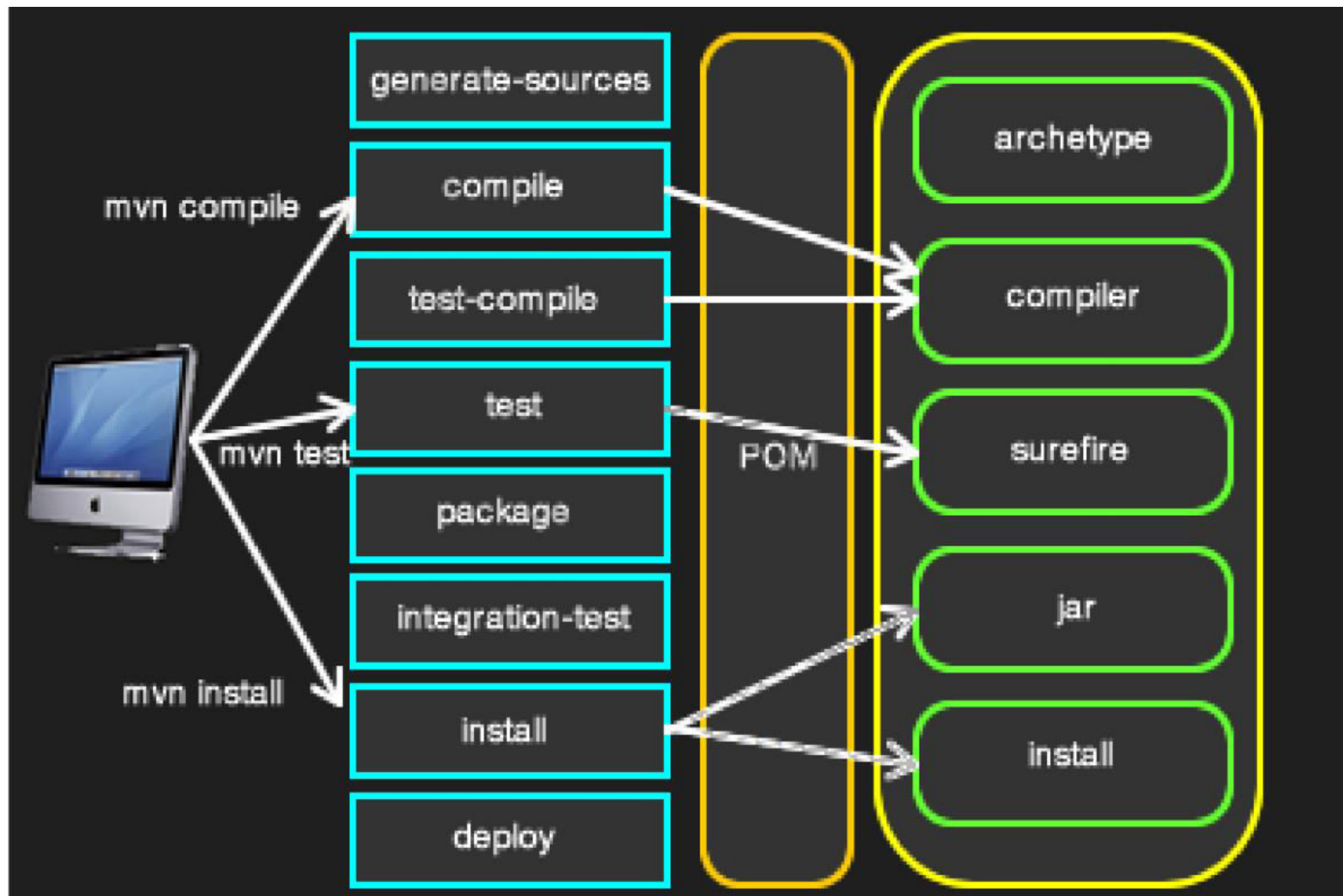
In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	<a href="#">apache-maven-3.6.0-bin.tar.gz</a>	<a href="#">apache-maven-3.6.0-bin.tar.gz.sha512</a>	<a href="#">apache-maven-3.6.0-bin.tar.gz.asc</a>
Binary zip archive	<a href="#">apache-maven-3.6.0-bin.zip</a>	<a href="#">apache-maven-3.6.0-bin.zip.sha512</a>	<a href="#">apache-maven-3.6.0-bin.zip.asc</a>
Source tar.gz archive	<a href="#">apache-maven-3.6.0-src.tar.gz</a>	<a href="#">apache-maven-3.6.0-src.tar.gz.sha512</a>	<a href="#">apache-maven-3.6.0-src.tar.gz.asc</a>
Source zip archive	<a href="#">apache-maven-3.6.0-src.zip</a>	<a href="#">apache-maven-3.6.0-src.zip.sha512</a>	<a href="#">apache-maven-3.6.0-src.zip.asc</a>

- [Release Notes](#)
- [Reference Documentation](#)
- [Apache Maven Website As Documentation Archive](#)
- All current release sources (plugins, shared libraries,...) available at <https://www.apache.org/dist/maven/>
- latest source code from source repository
- Distributed under the [Apache License](#), version 2.0

## Distribution Installation

- `#vi ~/.profile or vi ~/.bashrc`  
    `export JAVA_HOME="/opt/jdk7"`  
    `export PATH=$JAVA_HOME/bin:$PATH`  
  
    `export CATALINA_HOME="/opt/apache-tomcat"`  
    `export PATH=$CATALINA_HOME/bin:$PATH`  
  
    `export M2_HOME="/opt/apache-maven-3.5"`  
    `export PATH=$M2_HOME/bin:$PATH`  
  
`#source .profile`



Life Cycle Phases

Plugins