

1. Install Maven manually on the Jenkins server and the “mav” issue.

- Download the Maven binary from Apache Maven downloads.
- Extract it somewhere like `/opt/maven` or `C:\Tools\maven`.
- Add it to the system PATH or configure it directly in Jenkins:
 - Go to **Manage Jenkins** → **Global Tool Configuration** → **Maven**.
 - Uncheck *Install automatically*.
 - Point it to the manually installed Maven home (`/opt/maven/apache-maven-3.x.x`).

☐ Works even if the Jenkins version is too old to auto-install Maven.

2. Bundle Maven inside your Jenkins job/workspace

If you can't (or don't want to) install Maven globally:

- Add Maven as part of your repository (or fetch it with a script).
- Example in a pipeline:
- ```
pipeline {
 agent any
 stages {
 stage('Setup Maven') {
 steps {
 sh '''
 if [! -d maven]; then
 curl -s https://downloads.apache.org/maven/maven-
3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz | tar xz
 mv apache-maven-3.9.9 maven
 fi
 '''
 }
 }
 stage('Build') {
 steps {
 sh './maven/bin/mvn clean install'
 }
 }
 }
}
```

☐ Makes builds portable and independent of Jenkins' Maven plugin.

---

## 3. Use a Maven Docker image (if you have Docker agents)

- Instead of relying on Jenkins to have Maven installed:
- ```
pipeline {  
  agent {  
    docker {  
      image 'maven:3.9.9-eclipse-temurin-17'    }  
  }  
}
```

```

•     args '-v $HOME/.m2:/root/.m2'
•     }
•     }
•     stages {
•         stage('Build') {
•             steps {
•                 sh 'mvn clean install'
•             }
•         }
•     }
• }

```

□ Clean, reproducible builds, avoids old Jenkins tool limitations.

4. Upgrade just the Maven Plugin (not Jenkins itself)

Sometimes the **Maven Integration plugin** is too old.

- You can manually download a newer `.hpi` file from the Jenkins plugin archive.
- Drop it into `$JENKINS_HOME/plugins` and restart Jenkins.
 - Risk: might break if Jenkins is **too** old, so safer to use manual Maven installs instead.

Quick advice:

If this is a **long-lived Jenkins**, I'd recommend **manual Maven install (option 1)**.

If it's **per-project and flexible**, go with **pipeline-managed Maven (option 2 or 3)**.

Common Jenkins + Maven Issues Jenkins can't find nav

1. **Maven not installed or not on PATH**
 - Jenkins can't find `mvn` command.
 - Fix: Install Maven manually on the server and configure it in **Manage Jenkins** → **Global Tool Configuration** → **Maven**.
2. **Old Jenkins Maven Plugin**
 - The *Maven Integration* plugin doesn't support newer Maven versions.
 - Fix: Download a newer `maven-plugin.hpi` and drop it in `$JENKINS_HOME/plugins`, or bypass the plugin by using pipeline steps with a custom Maven.
3. **Pipeline job can't resolve Maven tool**
 - Example error:
 - No tool named 'Maven 3.8.6' found
 - Fix: Either define it under **Global Tools**, or bundle Maven yourself inside the job.
4. **Freestyle jobs missing Maven project option**
 - Happens if the Maven plugin is missing.
 - Fix: Install **Maven Integration plugin** (but be careful with Jenkins version compatibility).
5. **Maven build fails due to JAVA_HOME mismatch**
 - Maven needs a compatible JDK, but Jenkins is using the wrong Java.
 - Fix: In Jenkins, configure both JDK and Maven properly.

Workarounds

- **Option 1: Manual install (freestyle jobs)**

Install Maven manually and configure Jenkins to point to it.

- **Option 2: Pipeline-managed Maven**

```
• pipeline {
•   agent any
•   tools {
•     jdk 'jdk17'
•     maven 'maven-3.9.9'
•   }
•   stages {
•     stage('Build') {
•       steps {
•         sh 'mvn clean install'
•       }
•     }
•   }
• }
```

(This requires Maven & JDK to be defined in Global Tools.)

- **Option 3: Self-contained Maven in pipeline (no plugin needed)**

```
• pipeline {
•   agent any
•   stages {
•     stage('Build with custom Maven') {
•       steps {
•         sh '''
•           curl -s https://downloads.apache.org/maven/maven-
• 3/3.9.9/binaries/apache-maven-3.9.9-bin.tar.gz | tar xz
•           ./apache-maven-3.9.9/bin/mvn clean install
•         '''
•       }
•     }
•   }
• }
```

- **Option 4: Dockerized Maven agent**

```
• pipeline {
•   agent {
•     docker {
•       image 'maven:3.9.9-eclipse-temurin-17'
•       args '-v $HOME/.m2:/root/.m2'
•     }
•   }
•   stages {
•     stage('Build') {
•       steps {
•         sh 'mvn clean package'
•       }
•     }
•   }
• }
```