

总而言之，Apache Maven 作为管理和构建软件项目的健壮、通用且可靠的工具脱颖而出。其约定优于配置的方法、高效的依赖管理和基于插件的体系结构使其成为 Ubuntu 用户极具吸引力的选择。该指南将演示如何使用 CLI 命令在 Ubuntu 22.04 Jammy Jellyfish 或 Ubuntu 20.04 Focal Fossa 上安装 Apache Maven。

## 第 1 部分：使用 APT 安装 Apache Maven

在本节中，我们将引导您完成在 Ubuntu 上使用高级打包工具 (APT) 安装 Apache Maven 的过程。此方法简单方便，因为 Ubuntu 在其默认存储库中包含 Apache Maven。但是，值得注意的是存储库中的版本可能并不总是最新的。如果您需要最新版本的 Maven，您可能需要考虑手动安装，我们将在第 2 节中介绍。

### 第 1 步：更新您的 Ubuntu 系统

在我们深入安装过程之前，确保您的 Ubuntu 系统是最新的至关重要。此步骤有助于避免潜在的冲突并确保您拥有最新的安全补丁和系统改进。要更新系统，您需要在终端中执行以下命令：

```
sudo apt update && sudo apt upgrade
```

这个命令做了两件事。 `sudo apt update` 更新需要升级的包的升级包列表，以及刚刚进入存储库的新包。

### 第 2 步：安装 Apache Maven

更新系统后，您就可以安装 Apache Maven 了。如前所述，Ubuntu 在其默认存储库中包含 Apache Maven，这简化了安装过程。要安装 Apache Maven，请执行以下命令：

```
sudo apt install maven
```

这个命令告诉 APT 安装这个 `maven` 包。该 `sudo` 前缀确保该命令以超级用户权限运行，这对于系统范围的安装是必需的。

### 第 3 步：验证安装

安装完成后，最好验证 Apache Maven 是否已正确安装并检查安装的版本。此步骤有助于确保安装过程顺利进行，并确保您拥有满足您需要的正确版本的 Maven。要确认安装并检查安装的版本，请使用以下命令：

```
mvn -version
```

此命令要求 Maven 显示其版本号，该版本号应输出有关已安装的 Maven 版本的信息。

如果您发现安装的 Maven 版本不满足您的需要，或者您更喜欢使用最新版本，您需要参考第 2 部分，我们将在其中指导您完成手动安装最新版本的过程 Apache Maven 版本。

## 第 2 部分：Apache Maven 的手动安装

在本节中，我们将指导您完成手动安装最新版本的 Apache Maven 的过程。这种方法对于那些需要 Maven 最新功能的人或那些希望对安装过程有更多控制权的人特别有用。

### 步骤 1：安装默认 Java 开发工具包 (JDK)

在我们安装 Apache Maven 之前，我们需要确保您的 Ubuntu 系统上安装了 Java。Apache Maven 需要 Java 才能运行，因此这是必要的步骤。要在 Ubuntu 上安装默认 JDK，请使用以下命令：

```
sudo apt install default-jdk -y
```

此命令指示 APT 安装 default-jdk 软件包。该 -y 标志会自动确认安装，使您不必手动批准它。

### 第 2 步：下载 Apache Maven 档案

安装 Java 后，我们现在可以继续下载 Apache Maven 档案。wget 为此，我们将使用该命令，这是一个用于从 Web 非交互式下载文件的实用程序。

要查找最新版本的 Apache Maven，您可以访问[Apache Maven 下载页面](#)。出于本指南的目的，我们假设最新版本为 3.9.2。要下载此版本，请使用以下命令：

```
wget https://dlcdn.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-bin.tar.gz
```

### 第 3 步：提取 Apache Maven 存档

下载 Apache Maven 归档文件后，我们需要解压缩它。我们会将存档解压缩到 /opt/ 目录，这是用于存储可选或附加软件的标准目录。要提取存档，请使用以下命令：

```
sudo tar xzf apache-maven-3.9.2-bin.tar.gz -C /opt/
```

此命令使用该 `tar` 实用程序来提取存档。标志 `-xzf` 告诉 `tar` 提取 (`-x`) 压缩 (`-z`) 文件, `-C` 标志指定提取的目标目录。

```
cd /opt/
```

#### 第 4 步: 创建指向 Apache Maven 目录的符号链接

提取归档文件后, 我们将创建指向 Apache Maven 目录的符号链接。此步骤是可选的, 但建议这样做, 因为它可以让您更方便地引用目录。要创建符号链接, 请使用以下命令:

```
sudo ln -s apache-maven-3.9.2 maven
```

此命令创建一个指向 `maven` 该 `apache-maven-3.9.2` 目录的符号链接。

#### 第 5 步: 设置 Apache Maven 环境

接下来, 我们需要通过定义一些环境变量来设置 Apache Maven 环境。我们将通过在目录中创建一个新的脚本文件来完成此操作 `/etc/profile.d/`, 该目录是用于添加设置环境变量的脚本的标准目录。要创建脚本文件, 请使用以下命令:

```
sudo nano /etc/profile.d/maven.sh
```

此命令打开一个 `maven.sh` 在 `nano` 文本编辑器中命名的新文件。在此文件中, 添加以下行:

```
export M2_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

这些行定义了两个环境变量: `M2_HOME` 指向 Apache Maven 目录的, 以及 `PATH` 包含 Apache Maven 二进制文件路径的。要保存文件并退出编辑器, 请按 `CTRL+O`, 然后按 `CTRL+X`。

**或者**，我们可以使用 `echo` 命令连同 `sudo tee` 命令来实现这一点。该 `echo` 命令将生成必要的行，并将 `sudo tee` 这些行写入文件。以下是您的操作方法：

```
echo -e "export M2_HOME=/opt/maven\nexport PATH=\${M2_HOME}/bin:\${PATH}"  
| sudo tee /etc/profile.d/maven.sh
```

此命令执行以下操作：

该 `echo -e` 命令启用对反斜杠转义的解释并生成要添加到文件中的行。

(`|` 管道) 获取其左侧命令的输出，并将其用作右侧命令的输入。

该 `sudo tee /etc/profile.d/maven.sh` 命令将从命令接收到的输入写入 `echo` 文件 `/etc/profile.d/maven.sh`。该 `tee` 命令与 `结合使用 sudo` 以写入 `root` 用户拥有的文件。

此命令有效地创建 `maven.sh` 具有必要环境变量的文件，无需手动打开和编辑文件。

## 第 6 步：启用 Apache Maven 环境

定义了环境变量后，我们需要启用它们。我们将通过采购我们刚刚创建的脚本文件来做到这一点。获取脚本文件意味着在当前 `shell` 中执行该文件，在这种情况下允许环境变量在当前会话中可用。要获取脚本文件，请使用以下命令：

```
source /etc/profile.d/maven.sh
```

此命令 `maven.sh` 在当前 `shell` 中读取并执行文件。该 `source` 命令是一个 `shell` 内置命令，它从指定为当前 `shell` 环境中的参数的文件中读取并执行命令。

## 第 7 步：验证安装

最后，在设置 Apache Maven 环境后，最好验证一切设置是否正确。和以前一样，我们将通过检查已安装的 Apache Maven 版本来完成此操作。要验证安装，请使用以下命令：

```
mvn -version
```

此命令应输出有关已安装的 Apache Maven 版本的信息。如果一切设置正确，版本号应该与您下载的版本相匹配。

## 第 3 部分：创建和测试 Maven 项目

在本节中，我们将指导您完成创建简单 Maven 项目和运行基本测试的过程。这将帮助您验证 Maven 是否已正确设置并在您的 Ubuntu 系统上按预期运行。

## 第 1 步：创建一个新的 Maven 项目

创建 Maven 项目的第一步是生成一个新的项目结构。Maven 为此提供了一个命令，该命令创建一个带有默认 pom.xml 文件（项目对象模型文件）的新目录结构，这是 Maven 中的基本工作单元。

要创建新的 Maven 项目，请导航到要创建项目的目录并运行以下命令：

```
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

此命令告诉 Maven 从原型生成一个新项目。原型是项目的模板，在本例中是 maven-archetype-quickstart 一个基本的 Java 项目。groupId 和参数 artifactId 用于在所有项目中唯一标识您的项目。

## 第 2 步：导航到项目目录

创建项目后，使用以下命令导航到项目目录：

```
cd my-app
```

此命令将当前目录更改为该目录 my-app，该目录是您刚刚创建的项目的根目录。

## 第 3 步：构建项目

创建项目并设置项目目录后，您现在可以构建项目。构建项目会编译源代码并将其打包成可分发的格式，例如 JAR 文件。要构建项目，请使用以下命令：

```
mvn package
```

此命令告诉 Maven 执行 package 构建生命周期的阶段。该 package 阶段编译源代码，运行任何测试，并将编译后的代码打包成可分发的格式。

## 第 4 步：运行项目

最后，在构建项目之后，您可以运行它来验证一切是否按预期工作。要运行该项目，请使用以下命令：

```
java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App
```

此命令运行由命令生成的 JAR 文件中包含的 Java 应用程序 `mvn package`。该 `-cp` 标志指定类路径，命令的参数 `java` 是应用程序的主类。

如果一切设置正确，运行此命令应该输出“Hello World!”，表明您的 Maven 项目已正确设置并按预期运行。

## 关于在 Ubuntu Linux 上安装 Apache Maven 的结束思考

在本指南中，我们已经完成了在 Ubuntu 上安装 Apache Maven 的过程，涵盖了直接的 APT 方法和更实际的手动安装。我们还演示了如何创建和测试一个简单的 Maven 项目，确保您的 Maven 设置按预期工作。

在 Ubuntu 上安装 Apache Maven 是一个简单的过程，无论您选择从默认存储库安装还是手动安装最新版本。每种方法都有其优点，APT 方法更简单、更快捷，而手动方法可让您更好地控制和访问 Maven 的最新功能。

请记住，软件开发成功的关键是了解您的工具。Apache Maven 凭借其约定优于配置的方法、高效的依赖关系管理和基于插件的体系结构，是任何开发人员工具包中的强大工具。

## 其他资源

如需进一步阅读并加深您对 Apache Maven 的理解，请考虑以下官方资源：

**Apache Maven 官方网站**：这是 Apache Maven 的官方网站。它提供了丰富的信息，包括全面的 Maven 指南、文档以及最新消息和更新。

**Apache Maven 文档**：这是 Apache Maven 的官方文档。它涵盖了从 Maven 入门到更高级主题的所有内容，使其成为初学者和有经验的用户的宝贵资源。

**GitHub 上的 Apache Maven**：这是 Apache Maven 的官方 GitHub 存储库。这是探索源代码、为项目做贡献以及了解最新进展的好地方。