

制品仓库

最佳实践

文档版本 02
发布日期 2024-06-14



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 通过编译构建任务发布 Maven 组件并按照版本归档至私有依赖库.....	1
2 批量迁移 Maven/npm/PyPI 组件至私有依赖库.....	4

1 通过编译构建任务发布 Maven 组件并按照版本归档至私有依赖库

背景介绍

相对于开发过程中的“源代码”，制品仓库服务关注和管理开发产生的待部署的软件包。软件包通常是由源码编译构建或打包而成，其中涉及生命周期的元数据（如名称、大小等基本属性、代码库地址、代码分支信息、构建任务、构建者、构建时间）。在开发过程中，软件包会根据不同版本不断生成改进。

软件包及其属性的管理是发布过程管理的基础，也是软件开发过程中的重要资产，而能够及时查看软件包的版本记录也成为开发者面临的诉求。

前提条件

1. 已有可用项目。如果没有项目，请先[新建项目](#)。
2. 请添加当前账号对当前私有库的权限，请参考[管理用户权限](#)。

创建 Maven 类型私有依赖库

步骤1 登录软件开发生产线首页，单击项目卡片进入项目。

步骤2 单击菜单栏“制品仓库 > 私有依赖库”。

步骤3 单击 **+ 新建**，选择“本地仓”，输入仓库名称，选择“Maven制品类型”。

步骤4 单击“确定”。新建成功的Maven私有依赖库将显示在仓库视图中。

步骤5 在仓库视图中，单击对应的仓库名称，单击“设置仓库”。

步骤6 选择“项目关联权限”页签。

确认Maven私有依赖库是否关联项目：单击对应项目名所在操作列的  图标，在弹框中勾选目标私有依赖库的名称，单击“确定”。

----结束

在代码仓库中设置组件的版本

步骤1 登录软件开发生产线，进入已创建的项目。单击顶部菜单“服务 > 代码托管”，进入代码托管服务。

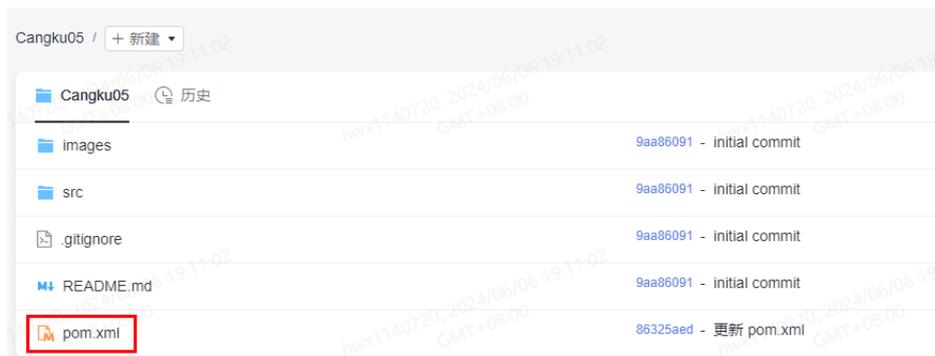
步骤2 单击“新建仓库”。

步骤3 选择“归属项目”，单击“模板仓库”，单击“下一步”。

步骤4 搜索并勾选“Java Maven Demo”类型模板，单击“下一步”。

步骤5 输入“代码仓库名称”，单击“确定”。

步骤6 进入代码仓库，单击“pom.xml”，查看组件配置。



步骤7 在组件配置页面中，<version>代码行中为当前组件的版本号（默认为1.0）。

单击页面右上方 ，可以修改版本号，修改完成后单击“确定”。

----结束

通过编译构建发布 Maven 私有组件到私有依赖库

步骤1 在代码仓库完成设置组件版本后，单击页面右上角“设置构建”，页面跳转至“新建编译构建任务”页面。

步骤2 在页面中选择“空白构建模板”，单击“下一步”。

步骤3 单击“点击添加构建步骤”。搜索并添加步骤“Maven构建”。

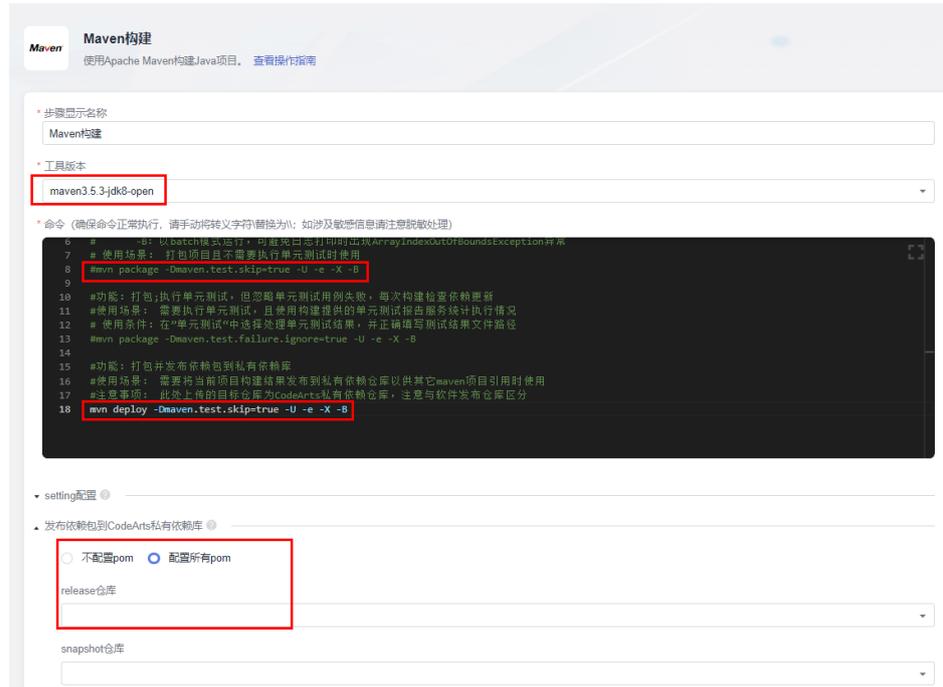


步骤4 编辑步骤“Maven构建”。

- 工具版本按照实际选择，本文中选择“maven3.5.3-jdk8-open”。
- 找到以下命令行，删除命令行前的#。

```
#mvn deploy -Dmaven.test.skip=true -U -e -X -B
```
- 找到以下命令行，在命令行前添加#。

```
mvn package -Dmaven.test.skip=true -U -e -X -B
```
- 在“发布依赖包到CodeArts私有依赖库”一栏勾选“配置所有pom”，并在下拉列表中选择与已项目关联的Maven私有依赖库。



步骤5 单击“新建并执行”，启动构建任务执行。

----结束

在 Maven 私有依赖库的版本视图中查看归档的组件

步骤1 进入目标私有依赖库，找到通过构建任务上传的Maven私有组件。

参考上述步骤，在代码仓库中[设置组件版本](#)，可将多个版本组件归档至私有依赖库。

步骤2 单击“版本视图”。

在包列表中，可以查看从编译构建中获取软件包的版本数和最新版本。

步骤3 单击“包名”，页面将显示该软件包最新版本的概览信息。

步骤4 选择“文件列表”页签，在列表中可以单击目标组件操作列中的↓，可将组件下载到本地。

步骤5 用户在本地对组件修改并设置新的版本号后，在目标私有依赖库中，单击“上传组件”，可将最新版本的组件上传至私有依赖库。

版本视图中的包列表显示对应组件最新上传的版本并统计版本归档过的数量。

----结束

2 批量迁移 Maven/npm/PyPI 组件至私有依赖库

背景介绍

制品仓库服务的私有依赖库支持通过页面手动上传下载私有组件；还支持与本地开发环境对接，通过本地开发环境上传下载私有组件。通过私有依赖库上传组件的操作请参考[上传私有组件](#)。

如果待上传的包太多，单个上传会非常麻烦。私有依赖库提供的批量迁移工具可以提高上传效率，在易用性上能够更加便捷地从nexus或其他类型仓库批量上传组件至私有依赖库。

准备工作

1. 已创建对应格式的私有依赖库。
2. 运行环境为python3。
3. 若使用Nexus，迁移工具必须和Nexus运行在同一台Linux主机上，且必须和CodeArts服务网络连通，该主机必须安装Python3。

迁移 Maven 组件

步骤1 从Maven本地仓库（例如：C:\Users\xxxxx\.m2\repository）找到需迁移的组件，复制到指定目录(用户自己指定)。

步骤2 进入私有依赖库，在左侧边栏中选择目标Maven私有依赖库。

步骤3 单击仓库名称，页面中仓库的详细信息显示“仓库地址”。单击即可复制该地址。

步骤4 单击页面右上方“操作指导”，在弹框中单击“下载配置文件”，将配置文件settings.xml下载到本地。

在本地打开配置文件，在文件中搜索并找到用户名与密码。

```
<server>
  <id>releases</id>
  <username>[redacted]</username>
  <password>[redacted]</password>
</server>
<server>
  <id>snapshots</id>
  <username>[redacted]</username>
  <password>[redacted]</password>
</server>
<server>
  <id>z_mirrors</id>
</server>
```

步骤5 单击页面右上方 ，单击“下载迁移工具”将迁移工具压缩包（脚本 uploadArtifact2.py、配置文件 artifact.conf）下载到本地。

步骤6 配置 artifact.conf。

```
[artifact]
packageType = 组件类型，设置为Maven
userInfo = username:password（步骤4中获取的用户名与密码）
repoRelease = 仓库类型为 Release地址（步骤3中获取的仓库地址）
repoSnapshot = 仓库类型为 Snapshot地址（步骤3中获取的仓库地址）
srcDir = 组件的目录路径（用户自己指定），如存放在步骤1下载组件的目标路径。

[nexus]
nexusAddr=nexus地址
nexusPort=nexus端口
repoName=待迁移的nexus仓库名称
userName=nexus用户名
passwd=nexus密码
```

步骤7 执行迁移脚本 `python uploadArtifact2.py`。

步骤8 进入对应的私有依赖库，查看组件包是否上传成功。

---结束

迁移 npm 组件

步骤1 进入私有依赖库，在左侧边栏中选择目标npm私有依赖库。

步骤2 单击仓库名称，页面中仓库的详细信息显示“仓库地址”。单击  即可复制该地址。

步骤3 单击页面右上方“操作指导”，在弹框中单击“下载配置文件”，将npmrc文件下载到本地。

在本地打开配置文件，在文件中找到“_auth”字段的值并进行base64解码。

步骤4 单击页面右上方 ，单击“下载迁移工具”将迁移工具压缩包（脚本 uploadArtifact2.py、配置文件 artifact.conf）下载到本地。

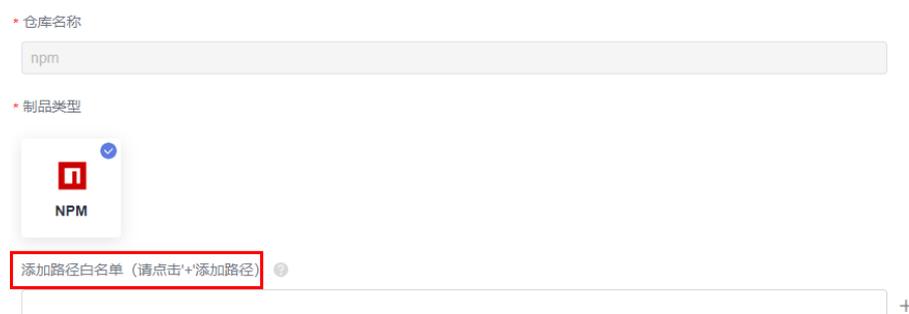
步骤5 配置 artifact.conf。

```
[artifact]
packageType = 组件类型，设置为npm
userInfo = npm仓库下的配置文件npmrc中通过base64解密后的_auth字段的值（参考步骤3）
repoRelease = 私有依赖库地址（步骤2中获取的仓库地址）
repoSnapshot = 保留为空
srcDir = 组件的目录路径，例如：C:\Users\xxxxxx\repository，用户自己指定

[nexus]
nexusAddr=nexus地址
nexusPort=nexus端口
repoName=待迁移的nexus仓库名称
userName=nexus用户名
passwd=nexus密码
```

步骤6 检查对应npm仓是否配置了路径白名单。

请确认package.json中私有二进制包是否在白名单内，只有与白名单内匹配的二进制包才能上传成功；没有配置白名单，则package.json中私有二进制包都可以上传成功。

**步骤7** 执行迁移脚本python uploadArtifact2.py。**步骤8** 进入对应的私有依赖库，查看组件包是否上传成功。

----结束

迁移 PyPI 组件

步骤1 进入私有依赖库，在左侧边栏中选择目标PyPI私有依赖库。**步骤2** 单击仓库名称，页面中仓库的详细信息显示“仓库地址”。单击即可复制该地址。**步骤3** 单击页面右上方“操作指导”，在弹框中单击“下载配置文件”，将配置文件pypirc下载到本地。

在本地打开配置文件，在文件中搜索并找到用户名与密码。

步骤4 单击页面右上方, 单击“下载迁移工具”将迁移工具压缩包（脚本uploadArtifact2.py、配置文件artifact.conf）下载到本地。**步骤5** 配置artifact.conf。

```
[artifact]
packageType = 组件类型，设置为pypi
userInfo = username:password (步骤3中获取的用户名与密码)
repoRelease = 私有依赖库地址 (步骤2中获取的仓库地址)
repoSnapshot = 保留为空
srcDir = 组件的目录路径，例如：C:\Users\xxxxxx\repository，用户自己指定

[nexus]
nexusAddr=nexus地址
nexusPort=nexus端口
repoName=待迁移的nexus仓库名称
userName=nexus用户名
passwd=nexus密码
```

步骤6 执行迁移脚本python uploadArtifact2.py。**步骤7** 进入私有库页面查看二进制包是否上传成功。

----结束