编译构建

最佳实践

文档版本	01
发布日期	2023-11-15

华为技术有限公司

商标声明


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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 基于 Maven 构建产物制作 Docker 镜像并发布到镜像仓（内置执行机/图形化构建）

应用场景

本实践为您介绍如何使用CodeArts Build将构建产物通过Dockerfile文件制作成Docker镜像，并发布到容器镜像服务的镜像仓库，您可以使用容器镜像中的构建产物进行编译或者部署。

约束限制

- 已在容器镜像服务中**创建组织**，组织名称为“codeci_gray”。
- 需已具备CodeArts Repo服务的操作权限。

操作流程

表 1-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建CodeArts Repo代码仓	为本实践创建构建过程中使用的代码文件。
新建构建任务	新建本实践中需要使用的构建任务并按照本实践场景配置任务并执行。
查看构建结果	查看本实践的构建结果，包括查看构建日志和结果文件。

新建项目

步骤1 使用华为云账号**登录华为云控制台页面**。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建 CodeArts Repo 代码仓

步骤1 在页面导航栏选择“代码 > 代码托管”。

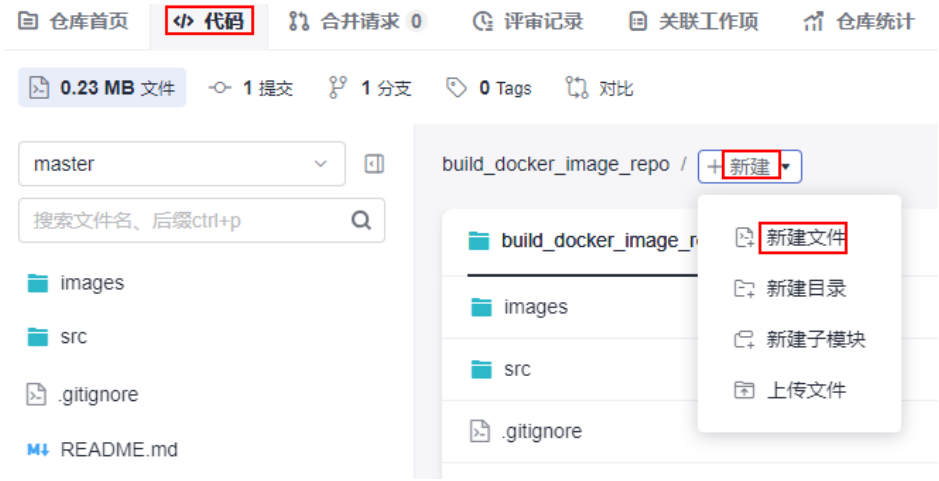
步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 填写代码仓库名称为“build_docker_image_repo”，其他参数保持默认即可。单击“确定”，代码仓创建完成，跳转到代码仓详情页面。

步骤5 在代码仓根目依次单击“新建 > 新建文件”。

图 1-1 新建文件



步骤6 文件名命名为“Dockerfile”，复制如下代码，粘贴到文件内容，如图1-3所示，单击“提交”。

```
FROM ubuntu:latest

# set maintainer
LABEL maintainer=build

RUN mkdir /release_app
COPY ./target/javaMavenDemo-1.0.jar /release_app/maven_app.jar

USER build
```

其中“javaMavenDemo-1.0.jar”为“pom.xml”文件里定义的“${artifactId}-${version}.${packaging}”，如图1-2所示。

图 1-2 pom.xml 文件

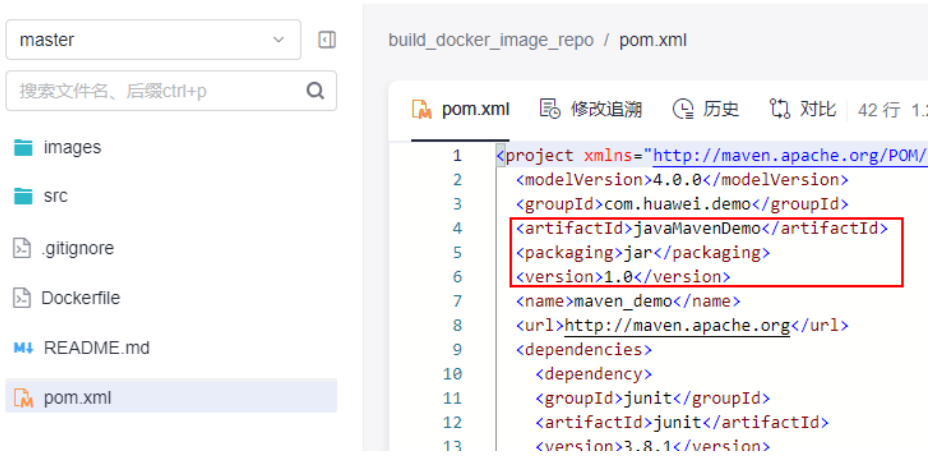
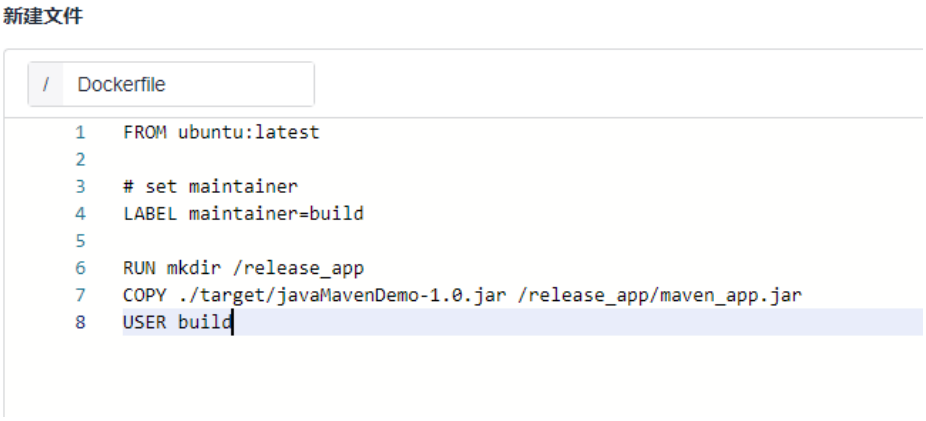


图 1-3 Dockerfile 文件内容



----结束

新建构建任务

步骤1 在页面导航中选择“持续交付 > 编译构建”。

步骤2 单击“新建任务”，根据表1-2填写参数信息，单击“下一步”。

表 1-2 基本信息配置

参数	说明
任务名称	自定义任务名称，例如：build_docker_image_task。
代码源	选择构建时拉取的代码源，这里选择“Repo”。
代码仓	选择新建CodeArts Repo代码仓中新建的代码仓库名称“build_docker_image_repo”。
默认分支	选择默认“master”即可。

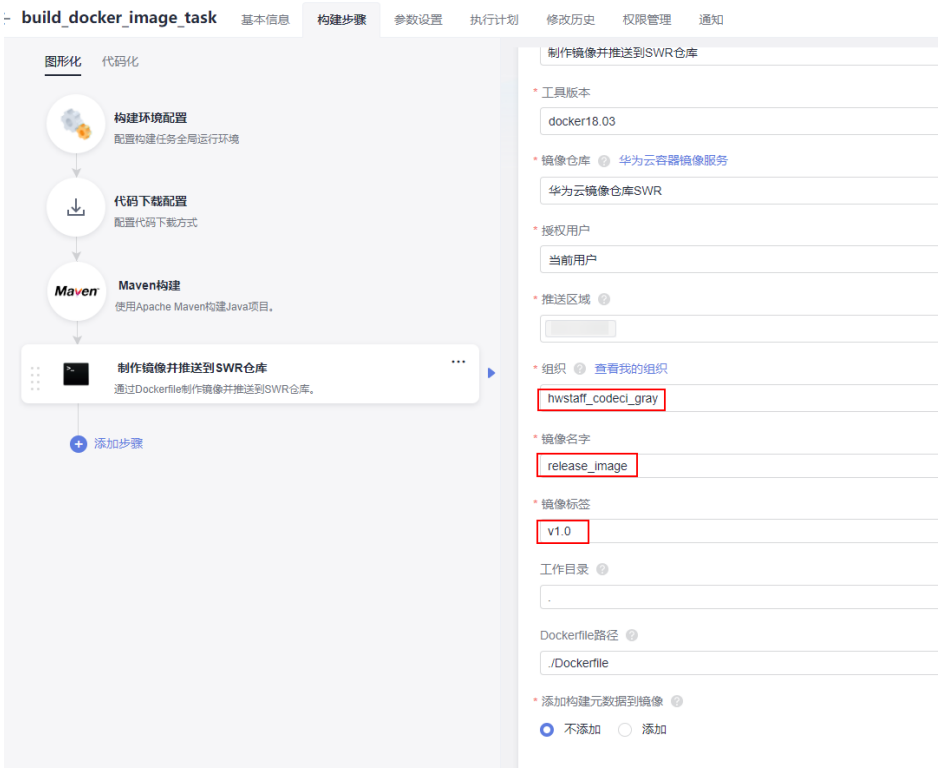
步骤3 选择“空白构建模板”，单击“确定”按钮，构建任务创建完成，自动跳转至构建步骤配置页面。

步骤4 在“构建步骤”页签，单击“图形化”，单击左侧“点击添加构建步骤”，添加“Maven构建”，参数保持默认即可。

步骤5 单击“添加步骤”，在右侧区域“容器类”页签中，单击“制作镜像并推送到SWR仓库”所在行的“添加”，按照图1-4配置参数。

其中“组织”选择约束限制中创建的组织名称“hwstaff_codeci_gray”，“镜像名称”输入“release_image”，“镜像标签”输入“v1.0”，其他参数保持默认即可。

图 1-4 配置构建步骤



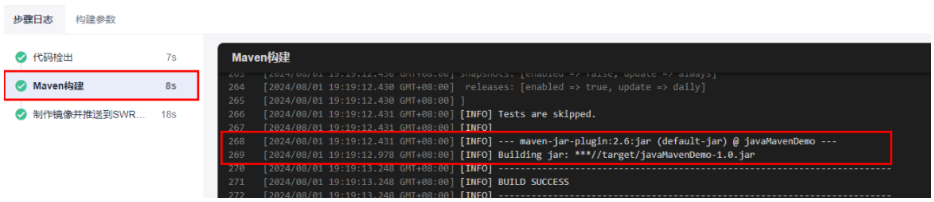
步骤6 单击页面右上角“保存并执行”，在弹出的窗口中单击“确定”，自动跳转到构建任务执行页面。

----结束

查看构建结果

步骤1 “步骤日志”页签中，“构建日志”控制台会滚动打印构建任务执行日志信息。如图1-5所示，构建日志控制台打印了Maven构建产物信息。

图 1-5 Maven 构建产物信息



步骤2 待构建任务成功执行完成后，跳转到“容器镜像服务”控制台页面，单击“我的镜像”，单击“自有镜像”页签，单击**步骤5**中制作的镜像名称“release_image”，即可查看镜像详情。

图 1-6 镜像列表页

----结束

2 使用 Maven 构建上传软件包至私有依赖库（内置执行机/图形化构建）

应用场景

当CodeArts Build提供的默认依赖库不满足业务要求时，您可使用自己搭建的私有依赖库进行构建。本实践以Maven构建为例，为您介绍构建完后如何上传构建产物到私有依赖库，以便后续构建使用，其他构建语言操作类似。

本实践需要依赖使用的其他服务如下：

- **代码托管服务**，用于存储实践中项目所使用的代码。
- **制品仓库服务**，用于存储实践中使用的私有依赖包。

约束限制

- 需已具备CodeArts Artifact服务的操作权限。
- 需已具备CodeArts Repo服务的操作权限。

操作流程

表 2-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建私有依赖库	新建本实践中使用的私有依赖库。
新建CodeArts Repo代码仓	新建本实践需要使用的代码仓。
新建构建任务并执行	新建本实践中需要使用的构建任务并按照本实践场景配置任务并执行。
查看构建结果	查看本实践的构建结果，包括查看构建日志和结果文件。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建私有依赖库

步骤1 选择导航栏“制品仓库 > 私有依赖库”。

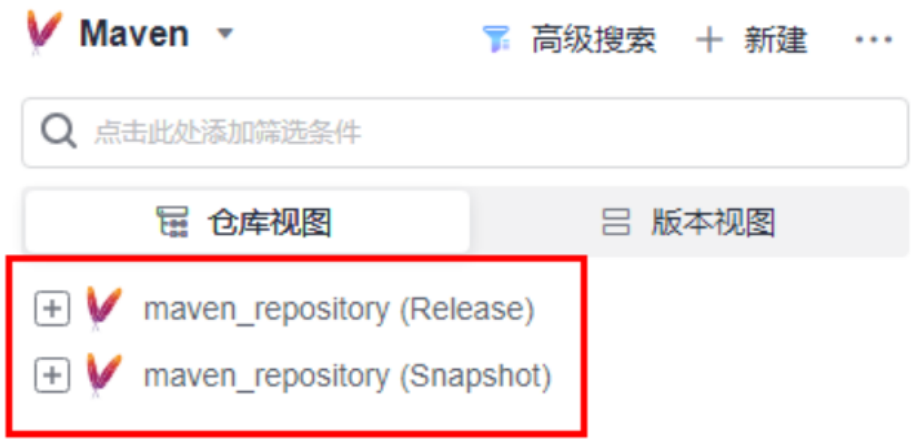
步骤2 单击“新建”，按照表2-2配置参数。

表 2-2 新建私有依赖库参数说明

参数	说明
仓库类型	选择“本地仓”。
仓库名称	自定义仓库名称，例如“maven_repository”。
制品类型	选择“Maven”。
归属项目	默认填写为“build-bestpractice”，无需手动填写。
添加路径白名单	本实践不涉及，无需填写。
版本策略	选择发布的版本，Release（功能稳定的发行版本）或者Snapshot（功能不稳定、处于开发阶段中的快照版本）。本实践选择“Release”和“Snapshot”。
描述	自定义描述信息。最多200个字符。

步骤3 单击“确定”，进入到“maven_repository”依赖库的详情页面。创建完成的私有依赖库如图2-1所示。

图 2-1 私有依赖库



----结束

新建 CodeArts Repo 代码仓

步骤1 在选择导航栏“代码 > 代码托管”。

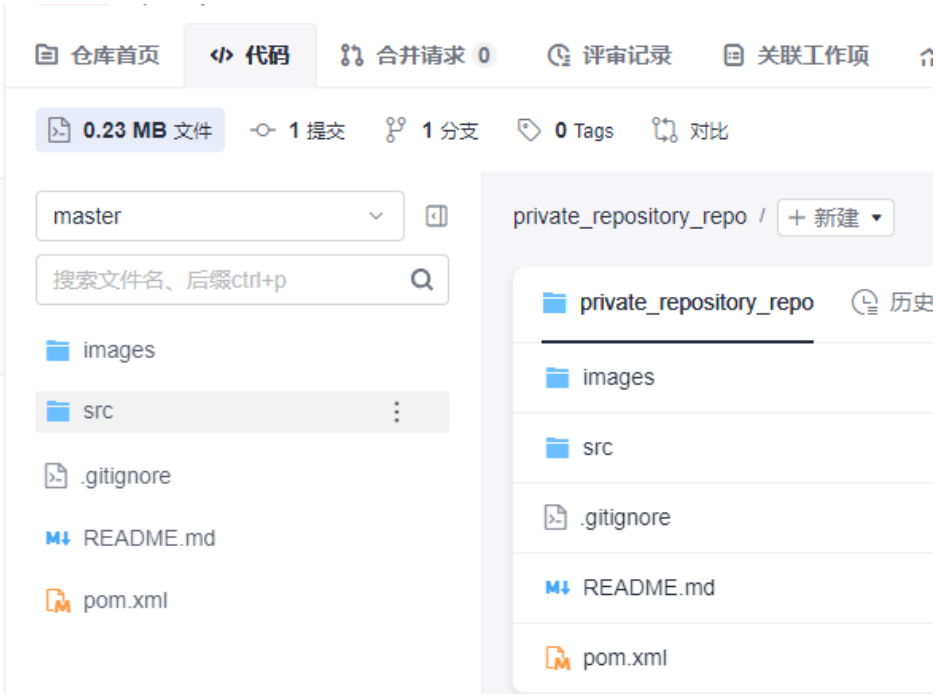
步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 在按模板新建页面，“代码仓库名称”命名为“maven_private_repository_repo”，其他参数保持默认即可。

步骤5 单击“确定”，自动跳转到“代码仓详情”页面。新建后代码仓文件目录如图2-2所示。

图 2-2 文件目录



----结束

新建构建任务并执行

步骤1 在编译构建服务页面，单击“新建任务”，按照如下参数说明配置参数，其他参数保持默认即可。

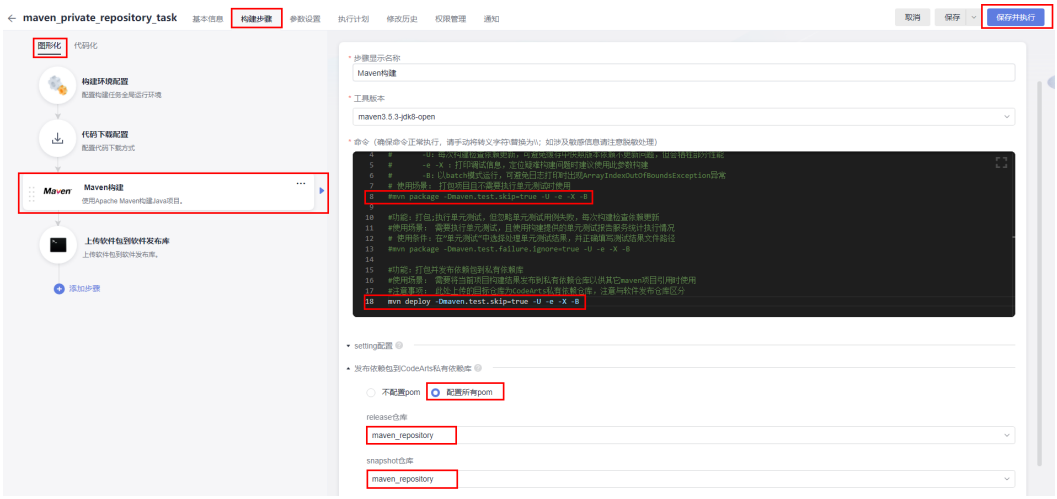
- 名称：自定义构建任务名称，例如“maven_private_repository_task”。
- 代码源：选择本次构建拉取的代码源，这里选择“Repo”。
- 代码仓：选择**新建CodeArts Repo代码仓**中新建的代码仓“maven_private_repository_repo”。

步骤2 单击“下一步”，选择“Maven”模板。然后单击“确定”，自动跳转到构建步骤配置页面。

步骤3 在“构建步骤”页签，单击“图形化”，单击左侧“Maven构建”，按照如下说明配置构建步骤，其他参数保持默认即可。

- 命令：**mvn package -Dmaven.test.skip=true -U -e -X -B**命令前加“#”，删除**#mvn deploy -Dmaven.test.skip=true -U -e -X -B**前的“#”。
- 发布依赖包到CodeArts私有依赖库：选择“配置所有pom”。
- “release仓库”和“snapshot仓库”选择**新建私有依赖库**中仓库名“maven_repository”。

图 2-3 配置 Maven 构建步骤



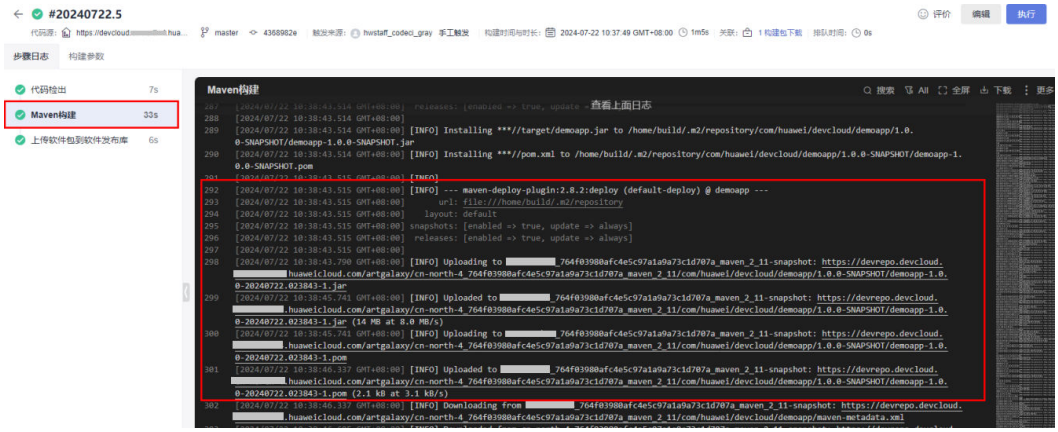
步骤4 单击页面右上角“保存并执行”，在弹出的窗口中单击“确定”，自动跳转到构建任务执行页面。

----结束

查看构建结果

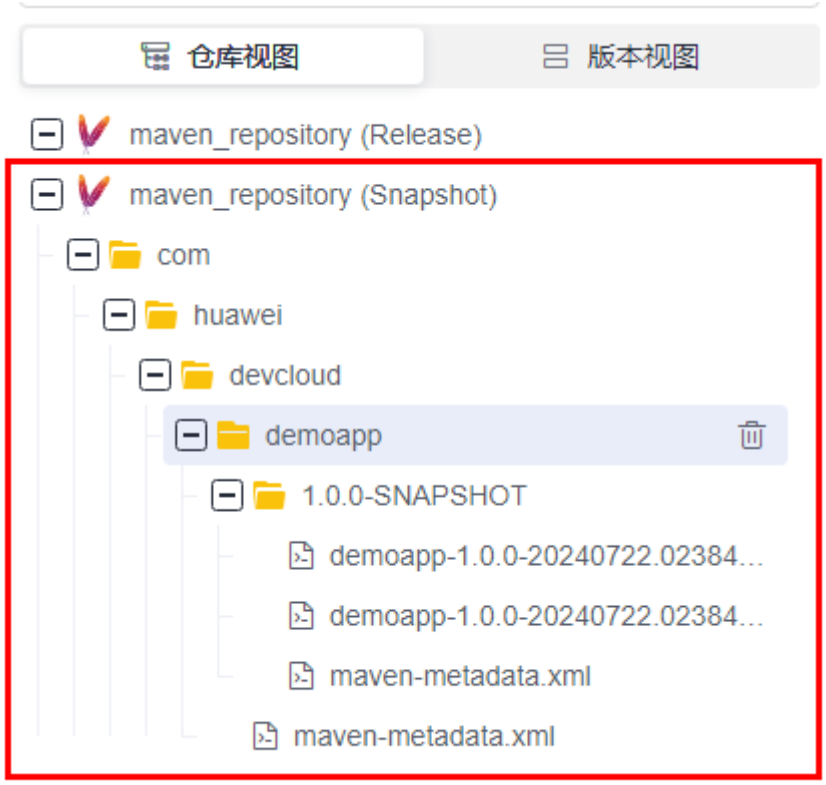
步骤1 待构建任务成功执行后，在“步骤日志”页签单击“Maven构建”，控制台打印了Maven构建产物上传到私有依赖库的日志信息。

图 2-4 查看构建日志



步骤2 选择页面导航栏“制品仓库 > 私有依赖库”，在“私有依赖库”展开“maven_repository (Snapshot)”目录，可查看发布的依赖包，如图2-5所示。

图 2-5 查看软件包



----结束

相关信息

当前实践展示的是归档“Snapshot”快照版本，如果要归档正式的“Release”发布版本，可以修改**新建CodeArts Repo代码仓**中的代码仓“pom.xml”文件中的“version”内容，将“1.0.0-SNAPSHOT”修改为“1.0.0”，提交文件改动，重新执行构建任务即可。

Maven构建会根据模块的版本号，即“pom.xml”文件中的“version”内容是否带有“-SNAPSHOT”来判断是快照版本还是正式版本。

3 使用 Maven 构建实现私有依赖包的上传及下载引用（内置执行机/图形化构建）

应用场景

本实践案例将为您介绍，如何在CodeArts Build构建工程中引用私有依赖库中的二方或三方依赖包，实现应用的编译构建。本实践案例为您演示依赖包的发布和下载引用两个环节，共涉及2个Maven构建工程，1个私有依赖库。

- 构建工程“dependency_task”：用于发布工具包，将“pom.xml”文件定义的“dependencyProject-1.0.jar”工具包发布到私有依赖库，为构建工程“release_task”提供依赖工具包引用。
- 构建工程“release_task”：用于发布应用，构建时依赖构建工程“dependency_repo”发布到私有依赖库的工具包“dependencyProject-1.0.jar”。
- 私有依赖库“dependency_libs”：存放构建工程“dependency_task”发布的工具包“dependencyProject-1.0.jar”，为构建工程“release_task”提供依赖工具包下载。

  本实践需要依赖使用的其他服务如下：
  - **代码托管服务**，用于存储实践中项目所使用的代码。
  - **制品仓库服务**，用于存储实践中使用的私有依赖包。

约束限制

- 需已具备CodeArts Artifact服务的操作权限。
- 需已具备CodeArts Repo服务的操作权限。

操作流程

表 3-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建私有依赖库	新建本实践中使用的私有依赖库。

流程	说明
新建dependency_repo代码仓	新建发布“dependencyProject-1.0.jar”工具包使用的代码仓。
新建dependency_task构建任务	新建发布“dependencyProject-1.0.jar”工具包的构建任务。
新建release_repo代码仓	新建发布应用使用的代码仓。
新建release_task构建任务	新建发布应用使用的构建任务，该构建任务依赖“dependencyProject-1.0.jar”工具包。
查看构建结果	查看本实践的的构建结果。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建私有依赖库

步骤1 选择导航栏“制品仓库 > 私有依赖库”。

步骤2 单击“新建”，按照表3-2配置参数。

表 3-2 新建私有依赖库参数说明

参数	说明
仓库类型	选择“本地仓”。
仓库名称	自定义仓库名称，例如“dependency_libs”。
制品类型	选择“Maven”。
归属项目	默认填写为“build-bestpractice”，无需手动填写。

参数	说明
添加路径白名单	本实践不涉及，无需填写。
版本策略	选择发布的版本，Release（功能稳定的发行版本）或者Snapshot（功能不稳定、处于开发阶段中的快照版本）。本实践选择“Release”。
描述	自定义描述信息。最多200个字符。

步骤3 单击“确定”，进入到“dependency_libs”依赖库的详情页面。创建完成的私有依赖库如图3-1所示。

图 3-1 私有依赖库

----结束

新建 dependency_repo 代码仓

步骤1 在选择导航栏“代码 > 代码托管”。

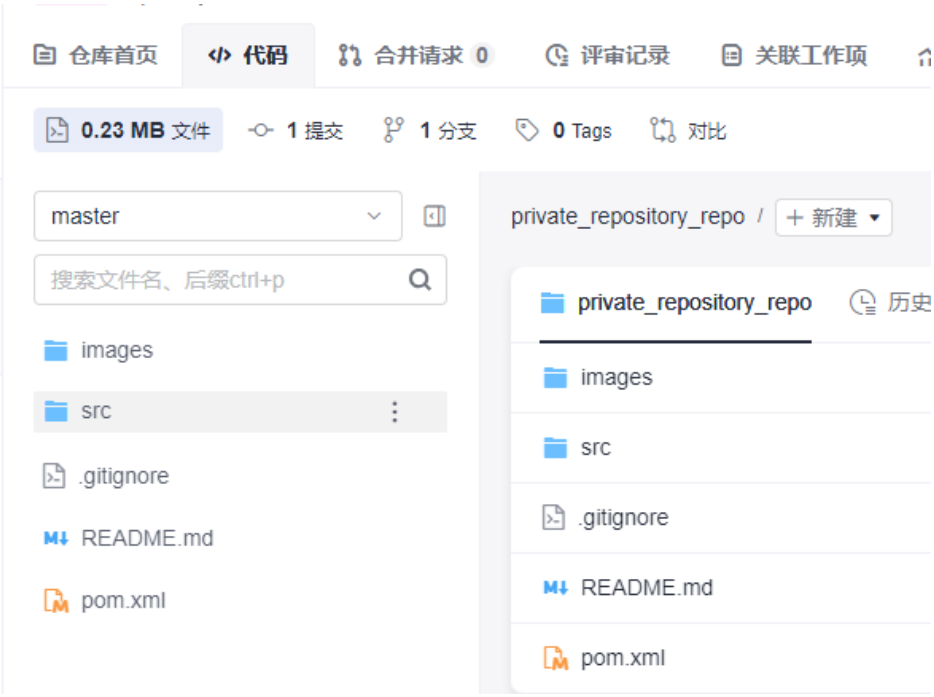
步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 在按模板新建页面，“代码仓库名称”命名为“dependency_repo”，其他参数保持默认即可。

步骤5 单击“确定”，自动跳转到“代码仓详情”页面。新建后代码仓文件目录如图3-2所示。

图 3-2 文件目录



步骤6 单击“pom.xml”文件，在右侧区域单击 ，进入“pom.xml”文件编辑页面。

图 3-3 编辑 pom.xml 文件

步骤7 “groupId”修改为“com.huawei.dependency”，“artifactId”修改为“dependencyProject”，“name”修改为“dependency_project”，参考图3-4所示。单击“确定”，保存修改后的“pom.xml”文件。

图 3-4 修改 pom.xml 文件

----结束

新建 dependency_task 构建任务

步骤1 选择导航栏“持续交付 > 编译构建”。

步骤2 单击“新建任务”，按照如下参数说明配置参数，其他参数保持默认即可。

- 名称：自定义，例如“private_repository_task”。
- 代码源：选择“Repo”。
- 代码仓：选择**新建dependency_repo代码仓**中新建的代码仓“private_repository_repo”。

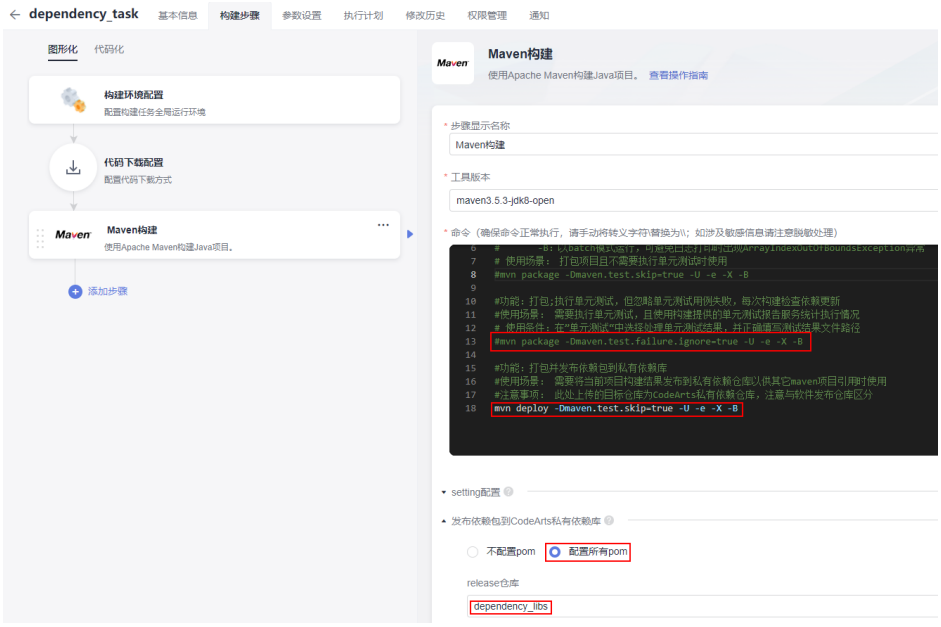
步骤3 单击“下一步”，选择“空白构建模板”。单击“确定”，进入到构建步骤配置页面。

步骤4 在“构建步骤”页签，单击“图形化”，单击左侧“点击添加构建步骤”，添加“Maven构建”。

步骤5 单击“Maven构建”，按照如下说明配置构建步骤，其他参数保持默认即可。

- 命令：**mvn package -Dmaven.test.skip=true -U -e -X -B**命令前加“#”，删除**#mvn deploy -Dmaven.test.skip=true -U -e -X -B**前的“#”。
- 发布依赖包到CodeArts私有依赖库：选择“配置所有pom”。
- “release仓库”选择**新建私有依赖库**中仓库名“dependency_libs”。

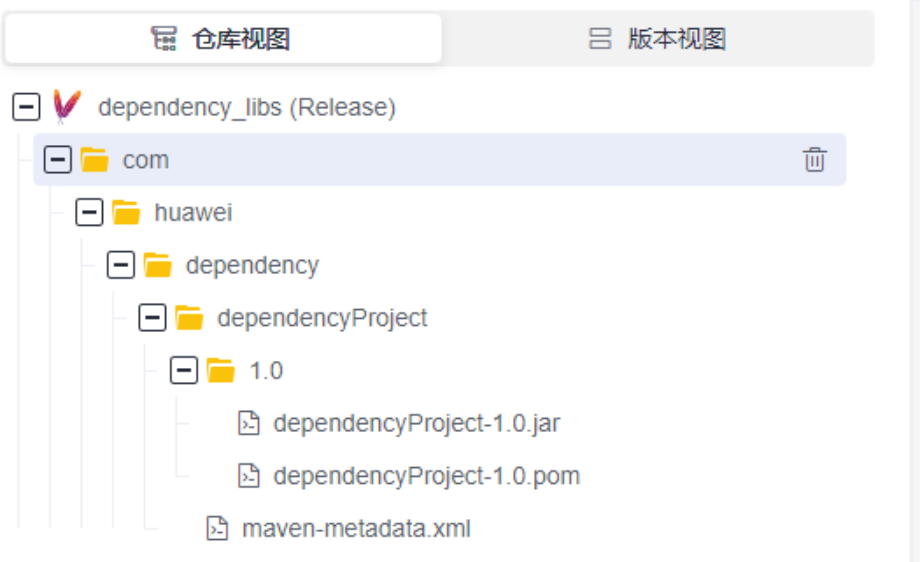
图 3-5 配置 Maven 构建步骤



步骤6 单击页面右上角“保存并执行”，在弹出的窗口中单击“确定”，自动跳转到构建任务执行页面。

步骤7 待构建任务成功执行完成后，选择页面导航栏“制品仓库 > 私有依赖库”，在“私有依赖库”展开“dependency_libs（Release）”目录，可查看发布的依赖包，如图3-6所示，为本次的构建产物。

图 3-6 查看依赖包



----结束

新建 release_repo 代码仓

步骤1 在选择导航栏“代码 > 代码托管”。

步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 在按模板新建页面，“代码仓库名称”命名为“release_repo”，其他参数保持默认即可。

步骤5 单击“确定”，自动跳转到“代码仓详情”页面。

步骤6 单击“pom.xml”文件，在右侧区域单击 ，进入“pom.xml”文件编辑页面，将如下代码复制到**图3-7**中红框处。单击“确定”，保存修改后的“pom.xml”文件。

以下示例代码表示“dependency”节点新增了对**新建dependency_task构建任务**中生成的依赖包“dependencyProject-1.0.jar”的引用。

```
<modelVersion>4.0.0</modelVersion>
  <groupId>com.huawei.release</groupId>
  <artifactId>releaseProject</artifactId>
  <packaging>jar</packaging>
  <version>1.0</version>
  <name>release_roject</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
    </dependency>
    <dependency>
    <groupId>com.huawei.dependency</groupId>
    <artifactId>dependencyProject</artifactId>
    <version>1.0</version>
    </dependency>
  </dependencies>
```

图 3-7 修改 pom.xml 文件

----结束

新建 release_task 构建任务

步骤1 选择导航栏“持续交付 > 编译构建”。

步骤2 单击“新建任务”，按照如下参数说明配置参数，其他参数保持默认即可。

- 名称：自定义，例如“release_task”。
- 代码源：选择“Repo”。
- 代码仓：选择**新建release_repo代码仓**中新建的代码仓“release_repo”。

步骤3 单击“下一步”，选择“空白构建模板”。单击“确定”，进入到构建步骤配置页面。

步骤4 在“构建步骤”页签，单击“图形化”，单击左侧“点击添加构建步骤”，添加“Maven构建”，参数保持默认配置即可。

步骤5 单击页面右上角“保存并执行”，在弹出的窗口中单击“确定”，自动跳转到构建任务执行页面。

图 3-8 执行构建任务

----结束

查看构建结果

在“步骤日志”页签中，“构建日志”控制台会滚动打印构建任务执行日志信息。如图所示，日志控制台输出信息表示从私有依赖库“dependency_libs”拉取依赖工具包“dependencyProject-1.0.jar”，“dependencyProject-1.0.jar”为**新建dependency_task构建任务**中发布到私有依赖库的工具包。

图 3-9 查看构建日志

4 使用 NPM 构建上传软件包至软件发布库（内置执行机/图形化构建）

应用场景

本实践帮助您了解如何通过编译构建服务的内置执行机，并以图形化构建的方式编译Node.js项目并上传软件包至软件发布库。

本实践需要依赖使用的其他服务如下：

- 代码托管服务，用于存储实践中项目所使用的代码。
- 制品仓库服务，用于存储实践中使用的私有依赖包。

约束限制

- 需已具备CodeArts Artifact服务的操作权限。
- 需已具备CodeArts Repo服务的操作权限。

操作流程

表 4-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建CodeArts Repo代码仓	为本实践新建Repo代码仓。
新建构建任务	为本实践新建编译构建任务。
配置构建步骤并执行构建任务	按照本实践场景配置构建步骤并执行构建任务。
查看并验证构建结果	查看并验证构建结果。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建 CodeArts Repo 代码仓

步骤1 在导航栏选择“代码 > 代码托管”。

步骤2 进入代码托管页面，单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Nodejs Webpack Demo”模板，单击“下一步”。

步骤4 在新建仓库页面将“代码仓库名称”命名为“nodesource”，其他参数保持默认即可，单击“确定”。

----结束

新建构建任务

步骤1 在导航栏选择“持续交付 > 编译构建”。

步骤2 单击“新建任务”，根据表4-2填写参数信息，单击“下一步”。

表 4-2 基本信息配置

参数	说明
任务名称	自定义任务名称，例如：npm_yml_build。
代码源	选择“Repo”。
代码仓	选择新建CodeArts Repo代码仓中新建的代码仓名称，选择“nodesource”。
默认分支	保持默认“master”即可。
任务描述	对该构建任务的描述。

步骤3 选择“npm”模板，单击“确定”，进入构建步骤配置页面。

----结束

配置构建步骤并执行构建任务

步骤1 配置“Npm构建”。

在命令编辑器里，**npm run build**命令前加“#”，新增**zip -r ./nodeserver.zip ./**命令，用来将代码打包成“nodeserver.zip”，如图4-1所示。其他参数保持默认即可。

图 4-1 命令示例

```
23    #npm run build
24    zip -r ./nodeserver.zip ./
25    #tar -zcvf demo.tar.gz ./**
```

步骤2 按图4-2所示配置“上传软件包到软件发布库”。

图 4-2 配置上传软件包到软件发布库

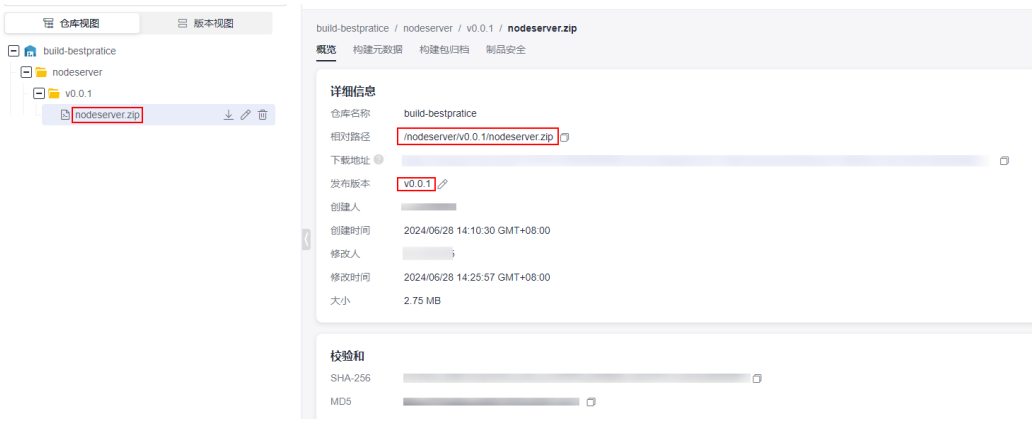
步骤3 配置完所有构建步骤，单击“保存并执行”，执行编译构建任务。

----结束

查看并验证构建结果

在导航栏选择“制品仓库 > 软件发布库”，查看上传的软件包，如图4-3所示。

图 4-3 查看上传的软件包



包名和发布版本与**步骤2**中配置的包名和发布版本号一致。

5 使用自定义执行机执行 Maven 构建（自定义执行机/图形化构建）

应用场景

当编译构建服务提供的内置执行机构建环境不满足业务要求时，您可接入自行提供的计算资源，通过注册的方式托管到编译构建服务中，委托编译构建服务进行调度并执行构建任务。本实践我们通过“Maven构建”和“上传软件包到软件发布库”两个构建步骤来演示使用自定义执行机的构建场景。

本实践需要依赖使用的其他服务如下：

- **代码托管服务**，用于存储实践中项目所使用的代码。
- **制品仓库服务**，用于存储实践中使用的私有依赖包。

约束限制

- 需已具备CodeArts Artifact服务的操作权限。
- 需已具备CodeArts Repo服务的操作权限。

前提准备

已参考**自定义购买ECS**购买自定义执行机使用的弹性云服务器。

操作流程

表 5-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建自定义执行机资源池	为本实践新建所需的自定义执行机资源池。
新建CodeArts Repo代码仓	为本实践新建存储代码的代码仓。

流程	说明
新建并执行编译构建任务	为本实践新建构建任务，包括“Maven构建”和“上传软件包到软件发布库”构建步骤。
查看构建任务和构建结果	为您介绍通过查看构建日志确认该实践的构建任务使用的执行机和在制品仓中查看上传的软件包。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角  ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建自定义执行机资源池

步骤1 在导航栏中单击用户名  ，选择“租户设置”。

步骤2 选择“资源池管理 > 资源池”。

步骤3 单击“新建资源池”，在弹出的窗口中参考表5-2配置参数后，单击“保存”。

表 5-2 资源池配置参数说明

参数名称	参数说明
资源池名称	资源池的名称，根据需要自定义。例如：custom_pool。
资源池类型	选择LINUX_DOCKER。执行任务时将拉起一个Linux docker容器，任务在容器中运行。
资源池描述	根据需要输入资源池描述。可不填写。
资源池可以被租户下所有子用户使用	勾选后，此资源池可以被当前租户下所有子用户使用。可不勾选。

步骤4 单击新建的资源池名称“custom_pool”，进入到资源池配置页面。

步骤5 单击“新建代理”，在弹出的窗口中，参考表5-3配置代理信息，其他参数项保持默认即可。

表 5-3 新建代理参数说明

参数	说明
是否安装Docker	勾选此项，配需安装Docker。
自动安装Docker	打开开关，自动安装Docker。
AK	参考**获取AK/SK**获取。
SK	参考**获取AK/SK**获取。
代理名称	自定义代理名称。例如：agent_test_custom。
代理工作空间	填写代理工作空间，需符合标准的linux目录格式。例如：/opt/agent_test_custom。

步骤6 勾选协议，依次单击“生成命令”和“复制命令”。单击“关闭”。

图 5-1 新建代理

步骤7 根据“步骤三”提示，在弹性云服务器列表页，单击**前提准备**中购买的服务器所在行的“远程登录”按钮，执行**步骤6**中复制的命令。

步骤8 在代理列表页面，单击“刷新列表”，后台自动同步信息后，代理列表中会增加一条代理执行机信息。代理执行机的代理别名为“agent_test_custom-mwlye1NlLG”。

图 5-2 代理执行机

----结束

新建 CodeArts Repo 代码仓

步骤1 在导航栏中选择“服务 > 代码托管”，进入代码托管服务首页。

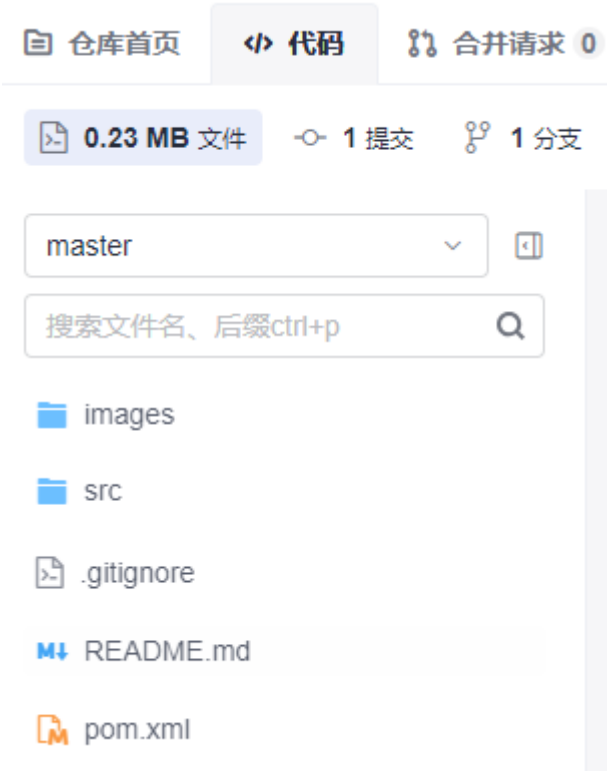
步骤2 单击“新建仓库”，在新建仓库页面“归属项目”选择新建项目中创建的项目名称，“仓库类型”选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”仓库模板，单击“下一步”。

步骤4 “代码仓库名称”填写为“custom_repo”，其他参数保持默认即可。单击“确定”，完成代码仓的创建。

创建完成后的代码仓文件目录如图5-3所示。

图 5-3 代码仓文件目录



----结束

新建并执行编译构建任务

步骤1 在页面导航栏中选择“持续交付 > 编译构建”。

步骤2 单击“新建任务”，根据表5-4填写参数信息，单击“下一步”。

表 5-4 基本信息配置

参数	说明
任务名称	自定义任务名称，例如：custom_task。
代码源	选择“Repo”。拉取CodeArts Repo代码仓中的代码进行编译构建。
代码仓	选择新建CodeArts Repo代码仓中新建的代码仓库名称“custom_repo”。
默认分支	选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤3 选择“Maven”模板，单击“确定”，进入构建步骤配置页面。

步骤4 参考配置“构建环境配置”步骤，其他参数保持默认即可，单击“保存并执行”。

表 5-5 构建环境配置参数说明

参数	说明
执行主机	选择自定义执行。
选择代理资源池	在下拉框中选择新建自定义执行机资源池中新建的资源池“custom_pool”。

图 5-4 构建环境配置

步骤5 在弹出的窗口中单击“确定”，跳转到构建任务运行页面。

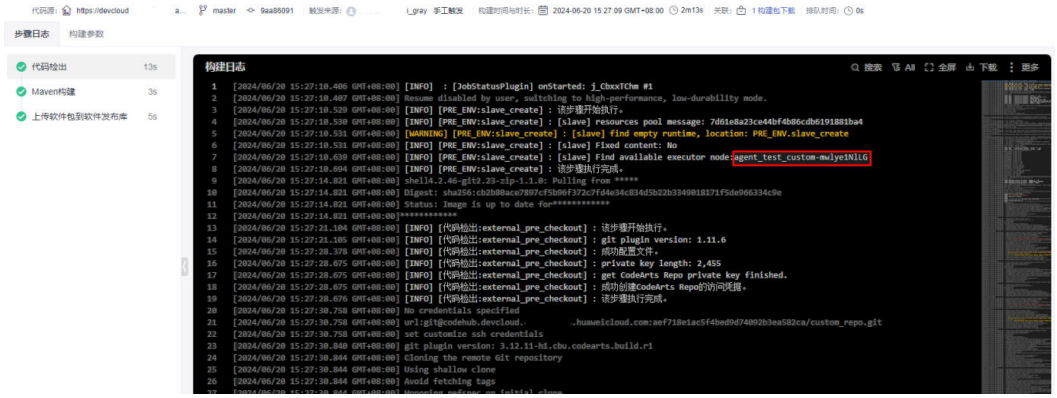
----结束

查看构建任务和构建结果

步骤1 在构建日志中，控制台会滚动打印构建任务执行日志信息。如图5-5所示，使用的执行机为新建自定义执行机资源池中新建的代理资源池“custom_pool”下的执行机

“agent_test_custom-mwlye1NlLG”，表示当前构建任务是在该代理执行机中运行的。

图 5-5 构建日志



步骤2 待构建任务执行完成后，每个构建步骤标记 表示构建任务执行成功。

图 5-6 构建任务执行成功

步骤3 单击步骤日志的“上传软件包至软件发布库”，在日志中查看上传到软件发布库的路径为“/custom_task/20240620.19/”，如图5-7所示。

图 5-7 查看上传的软件包路径

步骤4 在导航栏选择“制品仓库 > 软件发布库”，文件路径为“/custom_task/20240620.19/javaMavenDemo-1.0.jar”。

图 5-8 软件包信息

----结束

6 使用 Maven 构建上传软件包和推送镜像到 SWR（内置执行机/代码化构建）

应用场景

编译构建服务支持通过yaml文件配置构建脚本，用户可以将构建时需要配置的构建环境、构建参数、构建命令、构建步骤等操作，通过yaml语法编写成build.yml文件实现，并且将build.yml文件和被构建的代码一起存储到代码仓库。执行构建任务时，系统会以build.yml文件作为构建脚本执行构建任务，使构建过程可追溯、可还原，安全可信。本实践以使用Maven构建为例，为您演示上传软件包至软件发布库和推送镜像到SWR。

本实践需要依赖使用的其他服务如下：

- SWR，即**容器镜像服务**。SWR镜像仓库用于存储用户上传的Docker镜像，可以在构建、部署或其他场景使用。
- **代码托管服务**，用于存储实践中项目所使用的代码。
- **制品仓库服务**，用于存储实践中使用的私有依赖包。

约束限制

- 已在容器镜像服务中**创建组织**，组织名称为“codeci_gray”。
- 需已具备CodeArts Artifact服务的操作权限。
- 需已具备CodeArts Repo服务的操作权限。
- 代码化构建仅支持使用CodeArts Repo中的代码。

操作流程

表 6-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建CodeArts Repo代码仓	为本实践新建Repo代码仓用于存储代码文件。

流程	说明
新建build.yml文件	通过“build.yml”定义整个构建的流程。
新建Dockerfile文件	通过修改Dockerfile文件实现自定义镜像。
新建编译构建任务	新建本实践的编译构建任务。
查看并验证构建结果	查看并验证构建结果。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建 CodeArts Repo 代码仓

步骤1 在导航栏选择“代码 > 代码托管”。

步骤2 单击“新建仓库”，选择“模板仓库 ”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 在新建代码仓页面，“代码仓库名称”命名为“maven_yml_build”，其他参数保持默认即可。

步骤5 单击“确定”，进入代码仓详情页。

----结束

新建 build.yml 文件

步骤1 在代码仓详情页，单击“新建 > 新建目录”，如图6-1所示。

图 6-1 新建目录

步骤2 在新建目录页面，根据表6-2填写参数信息，单击“确定”。

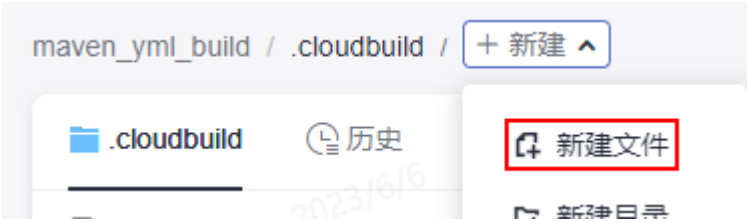
表 6-2 新建目录

参数	说明
目录名称	填写目录名称，例如“.cloudbuild”。文件目录名称仅支持中文，英文字母，数字，单斜杠“/”，下划线“_”，中横线“-”和点“.”，字符长度范围为1~100。
提交信息	目录的备注信息，用于记录该文件夹文件的描述信息。字符长度范围1~2000。

步骤3 单击步骤2中新建的目录名称。

步骤4 单击“新建 > 新建文件”，如图6-2所示。

图 6-2 新建文件



步骤5 文件命名为“build.yml”，并将如下代码拷贝到文件中。

```
# This YAML is the default template and can be modified based on this
---
version: 2.0
steps:
  BUILD:
  - maven:
      image: cloudbuild@maven3.5.3-jdk8-open # 可以自定义镜像地址
      inputs:
        settings:
          public_repos:
            - https://mirrors.huawei.com/maven
        cache: true # 是否开启缓存
        command: mvn package -Dmaven.test.failure.ignore=true -U -e -X -B
  - upload_artifact:
      inputs:
        path: "**/target/*.?ar"
  - build_image:
      inputs:
        organization: codeci_gray # 组织名称
        image_name: maven_demo # 镜像名称
```



```
    image_tag: 1.0 # 镜像版本
    dockerfile_path: ./Dockerfile
```

步骤6 单击“提交”。

----结束

新建 Dockerfile 文件

步骤1 在根目录下，参考**步骤4**新建名为“Dockerfile”的文件。文件中代码如下：

```
FROM swr.regionID.myhuaweicloud.com/codeci/special_base_image:centos7-base-1.0.2
MAINTAINER <devcloud@demo.com>
USER root
RUN mkdir /demo
COPY ./target/server-1.0.jar /demo/app.jar
```

其中**server-1.0.jar**为“pom.xml”文件中**artifactId**、**packaging**和**version**的参数值组合。

步骤2 单击“提交”。

----结束

新建编译构建任务

步骤1 在导航栏选择“持续交付 > 编译构建”。

步骤2 单击“新建任务”，根据**表6-3**填写参数信息。

表 6-3 基本信息配置

参数	说明
任务名称	自定义任务名称，例如：maven_yml_build。
代码源	选择“Repo”。
代码仓	选择**新建CodeArts Repo代码仓**中创建的代码仓库名称“Repo01”。
默认分支	保持默认“master”即可。
任务描述	对该构建任务的描述。

步骤3 单击“下一步”，选择“空白构建模板”。单击“确定”，进入构建步骤配置页面。

步骤4 单击“代码化”页签，可查看到导入的构建脚本，如**图6-3**所示。

图 6-3 代码化页签

步骤5 单击页面右上角的“保存并执行”。

----结束

查看并验证构建结果

- 查看上传的软件包。
  a. 选择页面导航栏“制品仓库 > 软件发布库”。
  b. 在软件发布库查看发布的软件包。软件包所在目录与**新建编译构建任务**时的任务名称一致，如**图6-4**所示。

  图 6-4 查看软件包

- 查看推送的镜像。
  a. 进入**容器镜像服务SWR**。
  b. 单击导航栏“我的镜像”，在组织中筛选**新建build.yml文件**时代码中填写的“组织名称”，如：codeci_gray。
  c. 在筛选结果中单击**新建build.yml文件**时代码中填写的“镜像名称”，如：maven_demo。

7 使用 Maven 构建执行多任务构建工程（内置执行机/代码化构建）

应用场景

在编译构建中，构建任务是构建的最小单元，适用于业务比较简单的场景，但是在有些复杂的构建场景下，构建任务可能并不能满足复杂的构建要求。例如，用户希望更模块化、更加细粒度的拆分构建任务，并按照构建任务之间的依赖顺序进行构建。

为此，编译构建服务支持使用BuildFlow将多个存在依赖关系的构建任务按照有向无环图（DAG）的方式组装起来，BuildFlow将会按照构建的依赖关系并发进行构建。

本实践为您演示构建任务Job3依赖于构建任务Job1和构建任务Job2的构建工程，实践中依赖使用**代码托管服务**，用于存储实践中项目所使用的代码。

约束限制

- 使用BuildFlow构建仅支持使用CodeArts Repo中的代码。
- 需已具备CodeArts Repo服务的操作权限。

操作流程

表 7-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建CodeArts Repo代码仓	为本实践新建Repo代码仓用于存储代码文件。
新建build.yml文件	通过“build.yml”定义整个构建的流程。
新建build.yml中使用的子任务执行脚本	新建整个构建过程中依赖的构建任务的执行脚本。
新建并执行编译构建任务	新建BuildFlow编译构建任务并执行。
查看编译构建结果	查看编译构建结果。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建 CodeArts Repo 代码仓

步骤1 在页面导航栏选择“代码 > 代码托管”。

步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 填写代码仓库名称为“Repo01”，其他参数保持默认即可。

步骤5 单击“确定”。

----结束

新建 build.yml 文件

步骤1 在代码仓详情页，选择“新建 > 新建目录”。

步骤2 目录名称填写“.cloudbuild”，描述信息自定义即可，单击“确定”。

步骤3 在“.cloudbuild”目录下，选择“新建 > 新建文件”，文件名命名为“build.yml”，文件中代码内容如下。

```
version: 2.0 # 必须是2.0，该版本号必填且唯一
params: # 构建参数，可在构建过程中引用
  - name: condition_param
    value: 1
# envs配置为非必填项。
envs:
  - condition: condition_param == 0 # 主机规格与类型的判断条件，不满足条件则不使用以下主机规格与类型
    resource:
      type: docker
      arch: ARM
  - condition: condition_param == 1 # 主机规格与类型的判断条件，满足条件会使用以下主机规格与类型
    resource:
      type: docker
      arch: X86

buildflow:
  jobs: # 构建任务
    - job: Job3 # 子任务的名称，可自定义
      depends_on: # 定义job的依赖,此处表示Job3依赖Job1和Job2
```



```
      - Job1
      - Job2
    build_ref: .cloudbuild/build_job3.yml # 定义Job3在构建过程中需要运行的yaml构建脚本
  - job: Job1
    build_ref: .cloudbuild/build_job1.yml # 定义Job1在构建过程中需要运行的yaml构建脚本
  - job: Job2
    build_ref: .cloudbuild/build_job2.yml # 定义Job2在构建过程中需要运行的yaml构建脚本
```

“build.yml”定义了整个构建的流程，当前定义了3个构建任务，Job3依赖于Job1和Job2，即构建优先级Job1、Job2 > Job3，且Job1和Job2优先级相同会同步触发。
“build_ref”定义了构建子任务所需运行的构建脚本。

----结束

新建 build.yml 中使用的子任务执行脚本

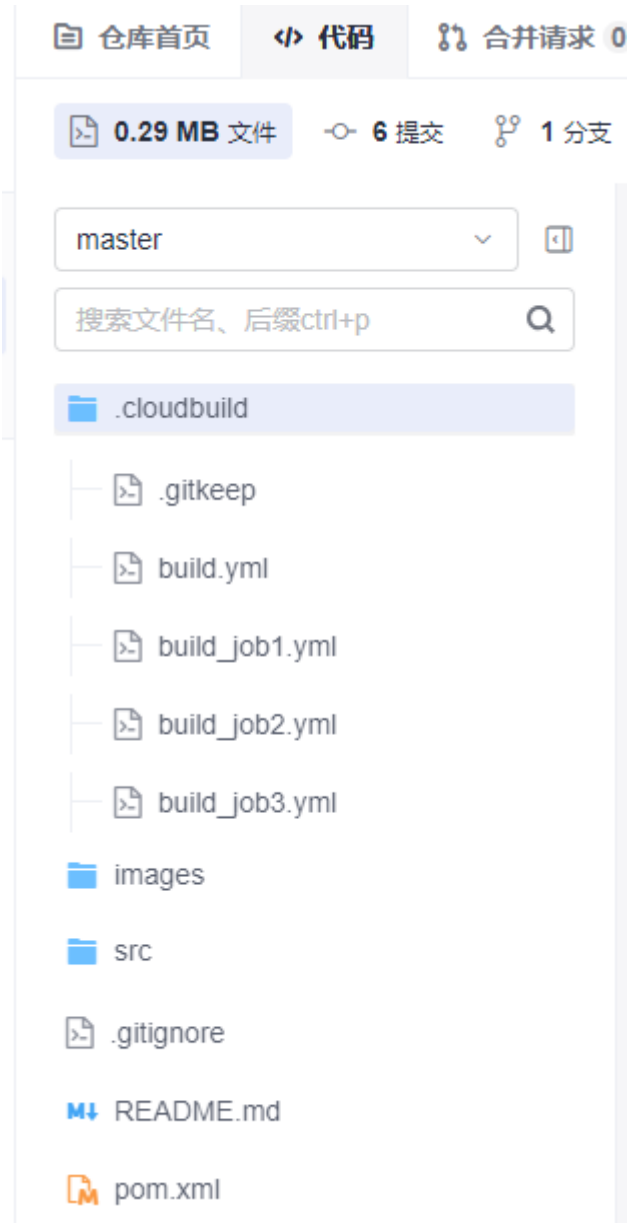
步骤1 在“.cloudbuild”目录下，选择“新建 > 新建文件”，文件名命名为“build_job1.yml”，文件中代码示例如下。

```
version: 2.0
steps:
  BUILD:
  - maven:
      image: cloudbuild@maven3.5.3-jdk8-open # 使用的构建镜像，用户可以自定义镜像
      inputs:
        settings:
          public_repos:
            - https://mirrors.huawei.com/maven # 配置依赖仓
        cache: true # 是否开启缓存
        command: mvn package -Dmaven.test.failure.ignore=true -U -e -X -B # 执行命令
```

步骤2 参考**步骤1**新建“build_job2.yml”和“build_job3.yml”，代码示例保持一致即可。

步骤3 文件新建完成后，代码仓文件目录如下图所示。

图 7-1 文件目录



----结束

新建并执行编译构建任务

步骤1 在页面导航中选择“持续交付 > 编译构建”。

步骤2 单击“新建任务”，根据表7-2填写参数信息。

表 7-2 基本信息配置

参数	说明
任务名称	自定义任务名称，例如：BuildFlow。
代码源	选择“Repo”。

参数	说明
代码仓	选择**新建CodeArts Repo代码仓**中新建的代码仓库名称“Repo01”。
默认分支	保持默认“master”即可。
任务描述	对该构建任务的描述。

步骤3 单击“下一步”，选择“Maven”模板。单击“确定”，进入构建步骤配置页面。

步骤4 单击“代码化”页签，会自动加载“Repo01”代码仓中的构建运行脚本。

步骤5 单击“保存并执行”，在弹出的窗口中单击“确定”即可跳转到构建任务运行页面。

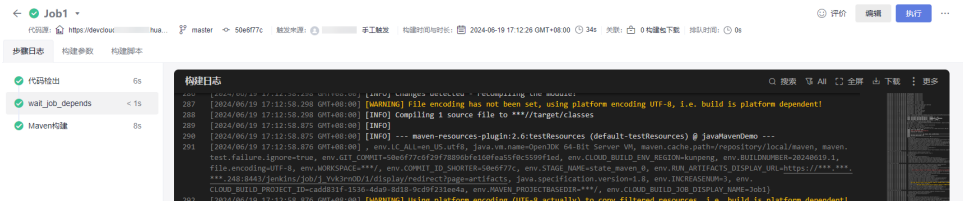
----结束

查看编译构建结果

“构建流程”页签中展示了当前构建任务运行的全量流程图，在构建任务未执行完成时，可以清晰的看到Job1和Job2是并行执行，Job3是等待Job1和Job2执行完成后才执行。

步骤1 单击“构建流程”页签中左侧菜单的“Job1”或右侧界面的绿色矩形图形的“Job1”，进入Job1构建任务的执行详情页面，可以查看Job1构建任务的构建日志，如**图7-2**所示。

图 7-2 查看构建结果



其中，

- “步骤日志”页签展示了当前构建任务运行顺序和资源调度情况。
- “构建参数”页签展示了当前构建任务全局的参数信息。
- “构建脚本”页签展示了当前构建任务执行的脚本内容。

步骤2 参考**步骤1**可查看“Job2”和“Job3”的构建任务执行详情。

----结束

8 基于私有依赖库使用 Maven 构建并上传软件包（内置执行机/图形化构建）

应用场景

当CodeArts Build提供的默认依赖库不满足业务要求时，用户可使用自己搭建的私有依赖库进行Maven构建。

本实践需要依赖使用的其他服务如下：

- **代码托管服务**，用于存储实践中项目所使用的代码。
- **制品仓库服务**，用于存储实践中使用的私有依赖包。

约束限制

- 需已具备CodeArts Artifact服务的操作权限。
- 需已具备CodeArts Repo服务的操作权限。

操作流程

表 8-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建私有依赖库	新建本实践中使用的私有依赖库。
查询私有依赖仓库信息	查询私有依赖库的id和url信息，用于配置在代码仓的“pom.xml”文件中。
上传settings.xml文件至编译构建	上传“settings.xml”文件到编译构建服务的“文件管理”中。
新建CodeArts Repo代码仓	新建本实践需要使用的代码仓。

流程	说明
配置Maven构建产物发布的私有依赖库地址	配置构建产物上传的私有依赖库的地址。
新建编译构建任务	新建本实践需要使用的编译构建任务。
配置构建步骤并执行构建任务	配置“下载文件管理的文件”和“Maven构建”步骤并执行构建任务。
查看编译构建结果	在私有依赖库中查看编译构建结果。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建私有依赖库

步骤1 选择导航栏“制品仓库 > 私有依赖库”。

步骤2 单击“新建”，按照如下表格配置参数。

表 8-2 新建私有依赖库参数说明

参数	说明
仓库类型	选择“本地仓”。
仓库名称	自定义仓库名称，例如“private_repository”。
制品类型	选择“Maven”。
归属项目	默认填写为“build-bestpractice”，无需手动填写。
添加路径白名单	本实践不涉及，无需填写。
版本策略	选择发布的版本，Release（功能稳定的发行版本）或者Snapshot（功能不稳定、处于开发阶段中的快照版本）。本实践选择“Release”。

参数	说明
描述	自定义描述信息。最多200个字符。

步骤3 单击“确定”，进入到private_repository依赖库的详情页面。

----结束

查询私有依赖仓库信息

步骤1 单击页面右上角“操作指导”。

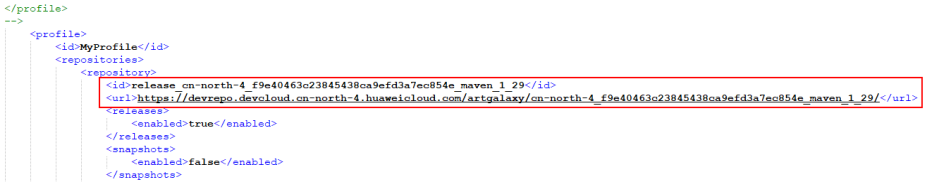
步骤2 在弹出的窗口中保持默认选项，单击“下载配置文件”。

步骤3 在弹出的窗口中单击“下载”。

图 8-1 下载配置文件

步骤4 打开下载到本地的“settings.xml”文件，找到“<profile>”节点下定义的仓库信息<repository>中的“id”和“url”，并记录。

图 8-2 查看仓库的 id 和 url



----结束

上传“settings.xml”文件至编译构建

步骤1 选择导航栏“持续交付 > 编译构建”。

步骤2 在编译构建任务列表页选择“更多 > 文件管理”。

步骤3 单击“上传文件”。

步骤4 在弹出的窗口中，上传**查询私有依赖仓库信息**中下载的“settings.xml”文件，勾选协议后单击“保存”。

----结束

新建 CodeArts Repo 代码仓

步骤1 选择导航栏“代码 > 代码托管”。

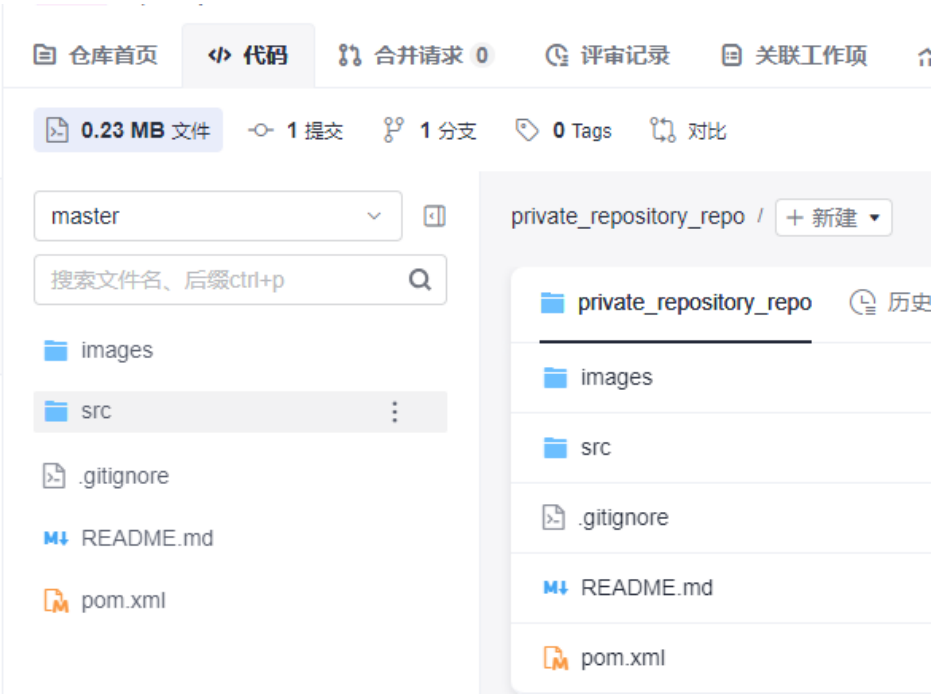
步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 在按模板新建页面，“代码仓库名称”命名为“private_repository_repo”，其他参数保持默认即可。

步骤5 单击“确定”，新建后代码仓文件目录如**图8-3**所示。

图 8-3 文件目录



----结束

配置 Maven 构建产物发布的私有依赖库地址

步骤1 单击“pom.xml”文件，在右侧区域单击 进入到文件编辑模式。

步骤2 将如下代码示例粘贴到build标记下方。

图 8-4 pom 文件代码示例


```
<distributionManagement>
  <repository>
    <id>ID</id>
    <url>https://example/</url>
  </repository>
 </distributionManagement>
```

其中“id”和“url”为**步骤4**中查看到的“id”和“url”。

步骤3 单击“确定”。

----结束

新建编译构建任务

步骤1 选择导航栏“持续交付 > 编译构建”。

步骤2 单击“新建任务”，按照如下参数说明配置参数，其他参数保持默认即可。

- 名称：自定义，例如“private_repository_task”。
- 代码源：选择“Repo”。
- 代码仓：选择**新建CodeArts Repo代码仓**中新建的代码仓“private_repository_repo”。

步骤3 单击“下一步”，选择“空白构建模板”。

步骤4 单击“确定”，进入到构建步骤配置页面。

----结束

配置构建步骤并执行构建任务

步骤1 单击“点击添加构建步骤”，添加“下载文件管理的文件”构建步骤，“步骤显示名称”和“工具版本”保持默认，“下载文件”选择**上传settings.xml文件至编译构建**中上传的文件“settings.xml”。

步骤2 单击“添加步骤”，添加“Maven构建”构建步骤，“命令”窗口中**mvn package -Dmaven.test.skip=true -U -e -X -B**命令前加“#”注释，删除**mvn deploy -Dmaven.test.skip=true -U -e -X -B**前的“#”，并将**mvn deploy -Dmaven.test.skip=true -U -e -X -B**改为**mvn deploy -Dmaven.test.skip=true -s settings.xml -U -e -X -B**，其他参数保持默认即可。

图 8-5 打包命令

```
# 使用场景： 打包项目且不需要执行单元测试时使用
#mvn package -Dmaven.test.skip=true -U -e -X -B
```

图 8-6 发布依赖包命令

```
#功能：打包并发布依赖包到私有依赖库
#使用场景： 需要将当前项目构建结果发布到私有依赖仓库以供其它maven项目引用时使用
#注意事项： 此处上传的目标仓库为CodeArts私有依赖仓库，注意与软件发布仓库区分
mvn deploy -Dmaven.test.skip=true -s settings.xml -U -e -X -B
```

步骤3 单击“保存并执行”。在弹出的窗口中单击“确定”，等待构建任务执行完成。

----结束

查看编译构建结果

步骤1 单击构建任务名称“private_repository_task”。

步骤2 在“构建历史”页签单击构建编号，查看步骤日志如下信息，其中“com/huawei/demo/javaMavenDemo/1.0”为构建产物在私有依赖库“private_repository”中上传的路径。

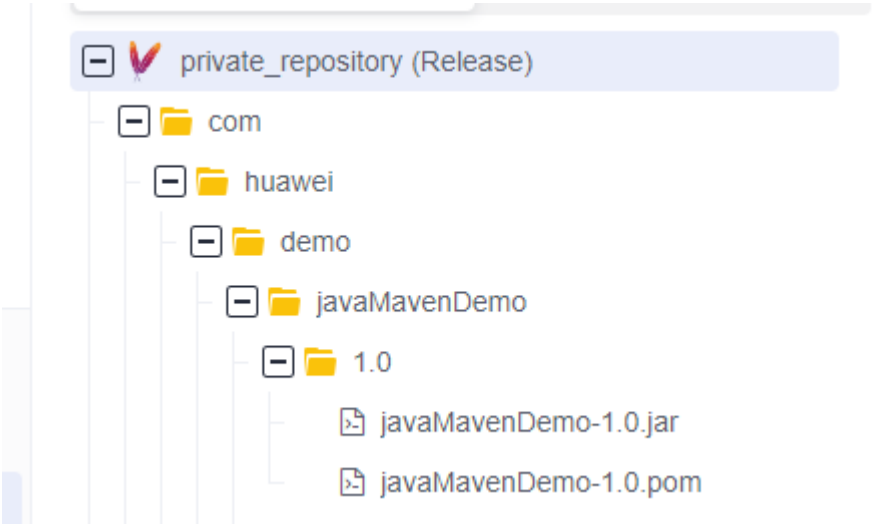
图 8-7 依赖包上传路径

```
257  [2024/06/26 10:31:50.908 GMT+08:00] [INFO] Uploaded to release_          _f9e40463c2384538ca9efd3a7ec854e_maven_1_29: https://devrepo.devcloud.
          .huaweicloud.com/artgalaxy/          _f9e40463c2384538ca9efd3a7ec854e_maven_1_29/com/huawei/demo/javaMavenDemo/1.0/javaMavenDemo-1.0.jar
     (2.4 kB at 3.3 kB/s)
```

步骤3 选择导航栏“制品仓库 > 私有依赖库”。

步骤4 展开“private_repository”，打开“com/huawei/demo/javaMavenDemo/1.0”，可查看到本实践上传的软件包。

图 8-8 查看上传的软件包



----结束

9 使用自定义构建环境执行构建任务（内置执行机/图形化构建）

应用场景

在构建过程中，通常会遇到以下这样的场景：

- CodeArts Build默认构建环境中支持的Java版本是1.8，而实际用户需要使用java 21。
- 构建时需要使用企业专有工具，CodeArts Build平台未支持。

针对于以上构建场景，本实践将为您介绍如何使用自定义构建环境执行构建任务。

约束限制

- 已在容器镜像服务中**创建组织**，组织名称为“hwstaff_codeci_gray”。
  - 需已具备CodeArts Artifact服务的操作权限。
  - 需已具备CodeArts Repo服务的操作权限。

操作流程

表 9-1 操作流程

流程	说明
新建项目	为本实践新建项目。
新建CodeArts Repo代码仓	为本实践创建构建过程中使用的代码文件。
制作自定义构建环境镜像	制作自定义环境使用的镜像。
新建构建任务并执行	新建本实践中需要使用的构建任务并按照本实践场景配置任务并执行。
查看构建结果	查看本实践的构建结果，包括查看构建日志和结果文件。

新建项目

步骤1 使用华为云账号登录华为云控制台页面。

步骤2 单击页面左上角 ☰ ，在服务列表中选择“开发与运维 > 软件开发生产线 CodeArts”。

步骤3 单击“立即使用”，进入CodeArts服务首页。

步骤4 在首页单击“新建项目”，选用“Scrum”项目模板。

步骤5 项目名称填写“build-bestpractice”，其他保持默认即可。

步骤6 单击“确定”后，进入到“build-bestpractice”项目下。

----结束

新建 CodeArts Repo 代码仓

步骤1 在页面导航栏选择“代码 > 代码托管”。

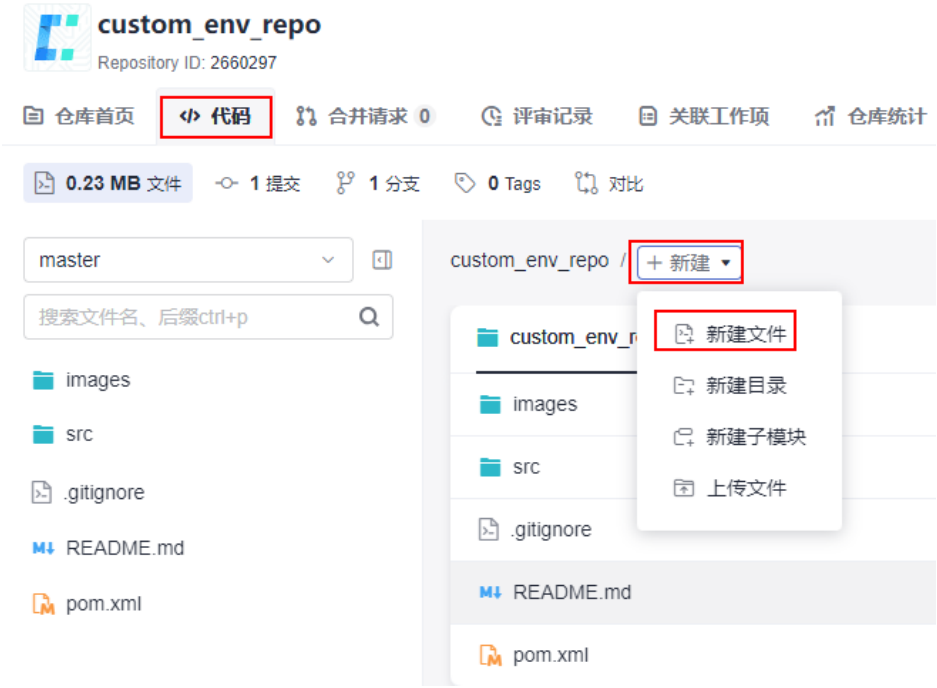
步骤2 单击“新建仓库”，选择“模板仓库”，单击“下一步”。

步骤3 选择“Java Maven Demo”模板，单击“下一步”。

步骤4 填写代码仓库名称为“custom_env_repo”，其他参数保持默认即可。单击“确定”，代码仓创建完成，跳转到代码仓详情页面。

步骤5 在代码仓根目依次单击“新建 > 新建文件”。

图 9-1 新建文件



步骤6 文件名命名为“Dockerfile”，复制如下代码，粘贴到文件内容，单击“提交”。

```
FROM ubuntu:latest
```



```
# set maintainer
LABEL maintainer=custom_image

RUN apt-get update && apt-get install -y wget

RUN mkdir /usr/java && \
  cd /usr/java && \
  wget "https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz" && \
  tar -xvf jdk-17_linux-x64_bin.tar.gz && \
  rm -rf jdk-17_linux-x64_bin.tar.gz

RUN mkdir /usr/maven && \
  cd /usr/maven && \
  wget "https://dlcdn.apache.org/maven/maven-3/3.9.8/binaries/apache-maven-3.9.8-bin.tar.gz" && \
  tar -xvf apache-maven-3.9.8-bin.tar.gz && \
  rm -rf apache-maven-3.9.8-bin.tar.gz

ENV JAVA_HOME /usr/java/jdk-17.0.12
ENV MAVEN_HOME /usr/maven/apache-maven-3.9.8
ENV PATH $PATH:$JAVA_HOME/bin:$MAVEN_HOME/bin

RUN java -version && mvn -v

USER build
```

图 9-2 Dockerfile 文件内容

“Dockerfile”文件为制作容器镜像环境的主体，文件中定义了如下内容。

表 9-2 Dockerfile 文件指令说明

指令	说明
FROM	指定基础镜像，必须为第一个命令。当前案例指定基础镜像为官方的最新ubuntu镜像。
LABEL	用于为镜像添加元数据。

指令	说明
RUN	构建镜像docker build时执行的命令。当前案例安装了三个工具：wget、jdk17、maven 3.9.8。并在设置了环境变量后，执行了jdk、maven对应的版本命令，确认jdk、maven是否正常安装。
ENV	设置环境变量。当前案例设置了jdk、maven的环境变量，并加入到PATH环境变量里面去，方便用户使用jdk、maven的快捷命令。
USER	设置运行容器时的用户。当前案例设置启动容器时的用户为“build”用户。

----结束

制作自定义构建环境镜像

步骤1 在页面导航中选择“持续交付 > 编译构建”。

步骤2 单击“新建任务”，根据表9-3填写参数信息，单击“下一步”。

表 9-3 基本信息配置

参数	说明
任务名称	自定义任务名称，例如：custom_env_task。
代码源	选择构建时拉取的代码源，这里选择“Repo”。
代码仓	选择新建CodeArts Repo代码仓中新建的代码仓库名称“custom_env_repo”。
默认分支	选择默认“master”即可。

步骤3 选择“空白构建模板”，单击“确定”按钮，构建任务创建完成，自动跳转至构建步骤配置页面。

步骤4 在“构建步骤”页签，单击“图形化”，单击“点击添加构建步骤”。

图 9-3 添加构建步骤

步骤5 在右侧区域“容器类”页签中，单击“制作镜像并推送到SWR仓库”所在行的“添加”，按照图9-4配置参数。其中“组织”选择约束限制中创建的组织名称

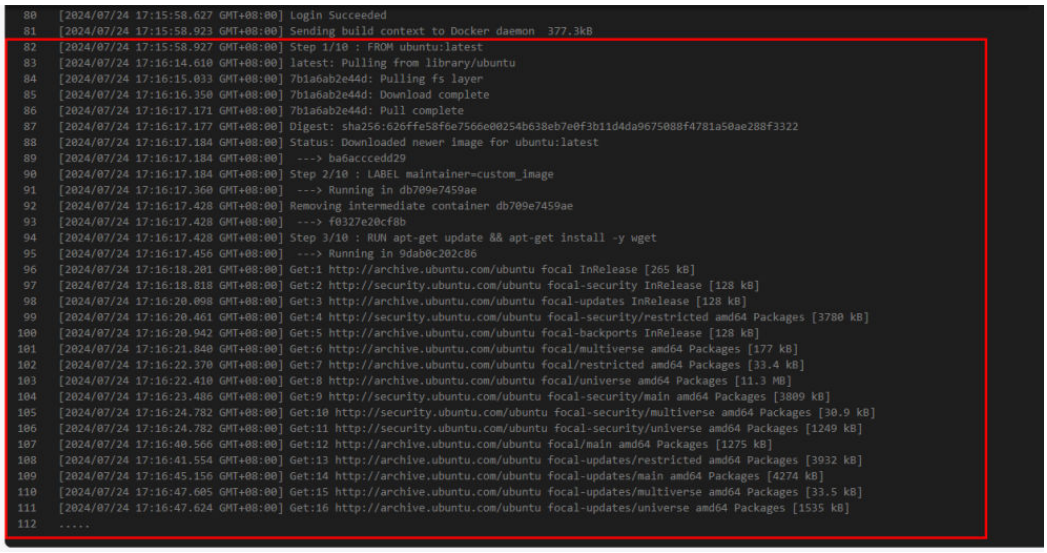
“hwstaff_codeci_gray”，“镜像名称”输入“custom_ubuntu_image”，“镜像标签”输入“v1.0”，其他参数保持默认即可。

图 9-4 配置构建步骤

步骤6 单击页面右上角“保存并执行”，在弹出的窗口中单击“确定”，自动跳转到构建任务执行页面。

步骤7 “步骤日志”页签中，“构建日志”控制台会滚动打印构建任务执行日志信息。如图9-5所示，构建日志控制台打印了根据代码仓的“Dockerfile”文件制作容器镜像的日志。

图 9-5 构建任务执行日志



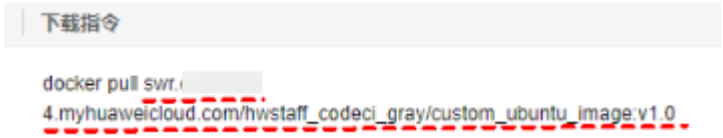
步骤8 待构建任务成功执行完成后，跳转到“容器镜像服务”控制台页面，单击“我的镜像”，单击“自有镜像”页签，单击步骤5中制作的镜像名称“custom_ubuntu_image”，跳转到镜像详情页面。

步骤9 在镜像详情页面单击“编辑”，在弹出的窗口中“类型”设置为“公开”，单击“确定”。

图 9-6 镜像详情

步骤10 在“下载指令”列，复制docker pull后面的完整镜像名称“swr.{regionID}.myhuaweicloud.com/hwstaff_codeci_gray/custom_ubuntu_image:v1.0”并保存到本地，便于后续的构建步骤使用。其中“{regionID}”为当前使用的区域的ID。

图 9-7 完整镜像名称



----结束

新建构建任务并执行

步骤1 在编译构建服务页面，单击“新建任务”，按照如下参数说明配置参数，其他参数保持默认即可。

- 名称：自定义构建任务名称，例如“custom_env_build_task”。
- 代码源：选择本次构建拉取的代码源，这里选择“Repo”。
- 代码仓：选择**新建CodeArts Repo代码仓**中新建的代码仓“custom_env_repo”。

步骤2 单击“下一步”，选择“空白构建模板”。然后单击“确定”，自动跳转到构建步骤配置页面。

步骤3 在“构建步骤”页签，单击“图形化”，单击“点击添加构建步骤”。

步骤4 在右侧区域“容器类”页签中，单击“使用SWR公共镜像”所在行的“添加”，按照**图9-4**配置参数。其中“镜像地址”填写**步骤10**中保存的完整镜像名称“swr.{regionID}.myhuaweicloud.com/hwstaff_codeci_gray/custom_ubuntu_image:v1.0”，将如下代码示例拷贝至“命令”中，其余参数保持默认即可。

```
java -version  # 打印当前镜像里安装的jdk版本号
mvn -v   # 打印当前镜像里安装的maven版本号
mvn package -Dmaven.test.skip=true -U -e -X -B  # 执行maven构建命令
```


图 9-8 配置使用 SWR 公共镜像

步骤5 单击“添加步骤”，添加“上传软件包至软件发布库”构建步骤，“构建包路径”输入“**/target/*.?ar”，其他参数保持默认即可。

步骤6 单击页面右上角“保存并执行”，在弹出的窗口中单击“确定”，跳转到构建任务执行页面。

----结束

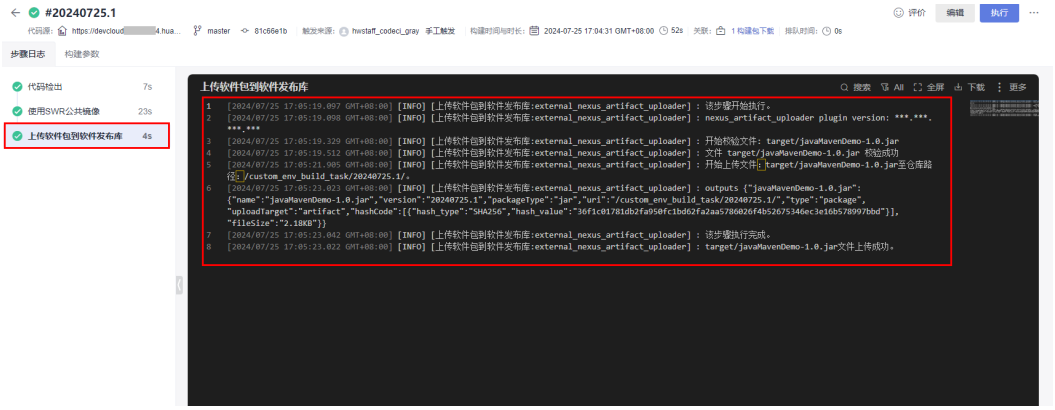
查看构建结果

步骤1 待构建任务成功执行后，在“步骤日志”页签，单击“构建日志”控制台上的“查看上面日志”，将日志向上滚屏。在日志中找到“Status: Downloaded newer image for swr.{regionID}.myhuaweicloud.com/hwstaff_codeci_gray/custom_ubuntu_image:v1.0”行，说明当前构建环境是基于自定义镜像进行构建的。

图 9-9 查看构建日志

步骤2 单击左侧“上传软件包到软件发布库”构建步骤，在右侧日志控制台打印了本次构建产物上传到软件发布库的信息。

图 9-10 构建产物上传到软件发布库的信息



步骤3 选择页面导航栏“制品仓库 > 软件发布库”，在软件发布库查看发布的软件包。

软件包所在目录与**构建任务**的名称一致，如**图9-11**所示。

图 9-11 查看软件包

----结束

