

编译构建

快速入门

文档版本 01
发布日期 2023-11-15



版权所有 © 华为技术有限公司 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

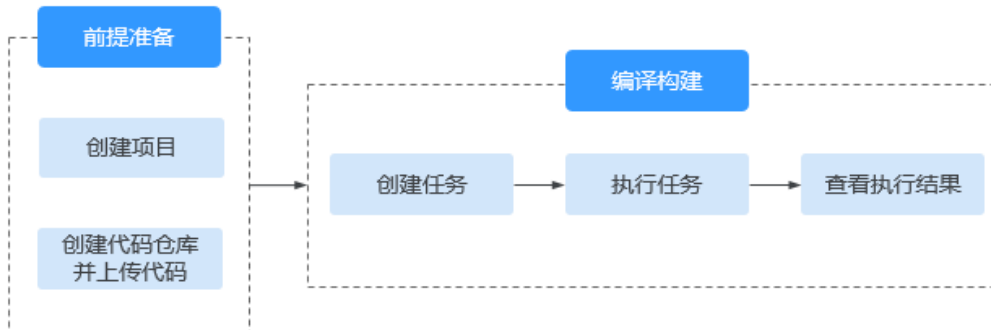
1 图形化构建.....	1
1.1 快速上手编译构建.....	1
1.2 快速上手 Maven 构建.....	4
1.3 快速上手 Npm 构建.....	7
1.4 快速上手 Android 构建.....	10
1.5 快速上手 CMake 构建.....	12
1.6 使用鲲鹏环境构建.....	15
2 代码化构建.....	18
2.1 Maven 构建任务.....	18
2.2 Npm 构建任务.....	24
3 入门实践.....	30

1 图形化构建

1.1 快速上手编译构建

编译构建为开发者提供配置简单的混合语言构建平台，实现编译构建云端化，支撑企业实现持续交付，缩短交付周期，提升交付效率。支持编译构建任务一键创建、配置和执行，实现获取代码、构建、打包等活动自动化，实时监控构建状态，让您更加快速、高效地进行云端编译构建。

本节为您介绍编译构建服务的基本操作流程，帮助您快速上手编译构建。



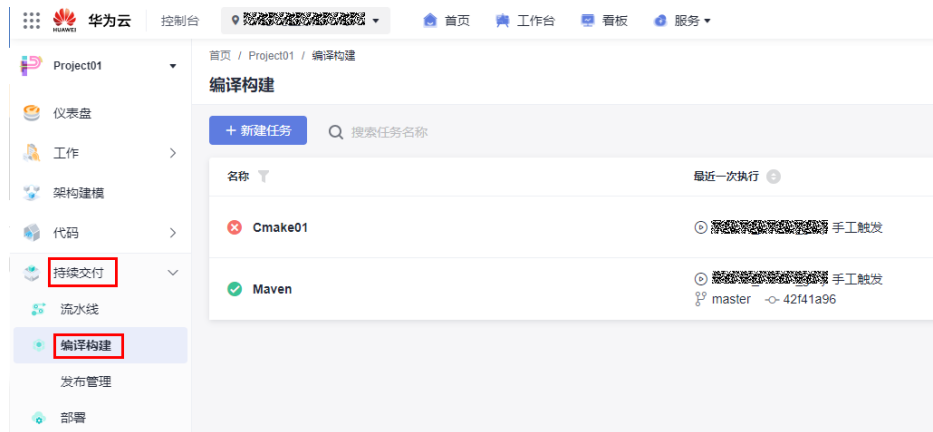
前提条件

- 已注册华为账号并开通华为云。
- 已有可用项目，如果没有，请先新建项目。
- 已在项目中新建代码仓库并上传代码，如果没有，请按如下步骤新建。
 - a. 使用华为云账号登录软件开发生产线首页。
 - b. 单击需要创建代码仓的项目名称。
 - c. 在页面导航栏中选择“代码 > 代码托管”。
 - d. 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。
 - e. 单击“下一步”，在“按模板新建 > 选择仓库模板”页面，搜索并选择“Java Maven Demo”模板。
 - f. 单击“下一步”，填写仓库名称等信息，然后单击“确定”完成仓库创建。

- g. [上传代码至代码仓库](#)。

新建编译构建任务

步骤1 在页面导航栏选择“持续交付 > 编译构建”，进入编译构建服务。



步骤2 单击“新建任务”，进入配置“基本信息”页面。

步骤3 填写参数后，单击“下一步”。

表 1-1 基本信息配置参数

参数名称	参数说明
任务名称	自定义任务名称。
源码源	选择“Repo”。
源码仓库	选择 前提条件 中创建的代码仓库名称。
分支	选择 前提条件 创建代码仓时创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 选择“Maven”模板。单击“下一步”。

步骤5 进入“构建步骤”页签，根据需要自定义配置[构建步骤](#)。

步骤6 配置完成后，单击“新建”，完成任务创建。

----结束

执行编译构建任务

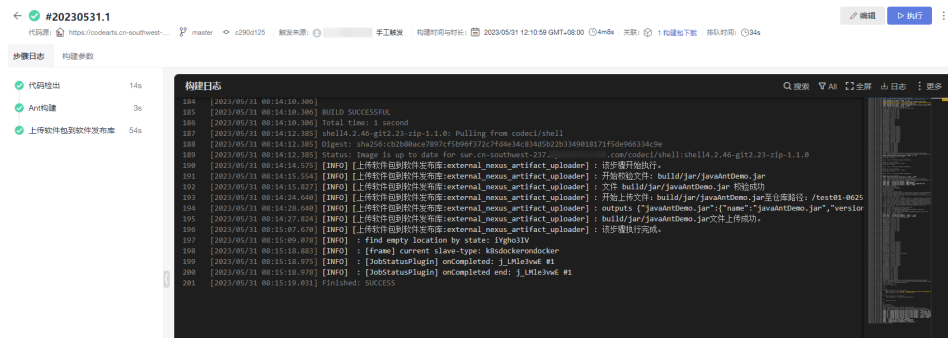
介绍如何执行构建任务以及查看构建结果。

步骤1 单击构建任务名称。

步骤2 进入“构建历史”页面，单击右上角“执行”，启动构建任务。

- 若出现如[图1-1](#)页面，表示任务执行成功。

图 1-1 任务构建成功



- 若任务执行失败，可通过页面提示信息或分析日志排查解决。

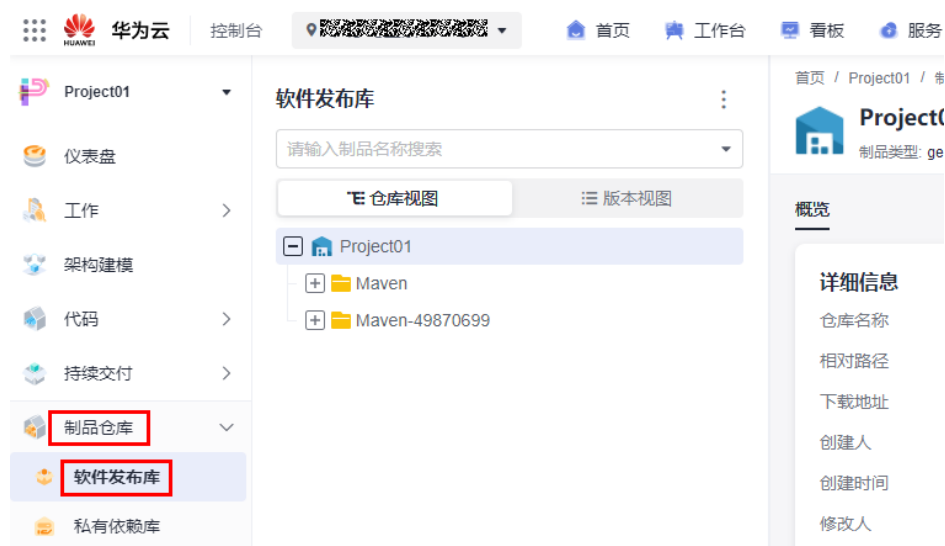


----结束

查看构建包

使用默认配置构建生成的构建包，存储在软件发布库“构建名/构建时间”目录。本节介绍如何查看构建包、验证发布结果。

步骤1 选择页面导航栏“制品仓库 > 软件发布库”。



步骤2 在软件发布库查看发布的软件包。软件包与新建编译构建任务时的任务名称一致，如图 1-2 所示。

图 1-2 查看软件包



----结束

资源清理

为了避免不必要的费用产生，完成本示例体验后，建议释放构建相关资源。

- 代码托管：删除代码仓库。
- 软件发布库：[删除软件包](#)，并[清空回收站](#)。

须知

资源释放后无法恢复，请谨慎操作。

1.2 快速上手 Maven 构建

编译构建提供多种构建模板，其中“Maven”模板可以对Java Maven项目进行构建打包。

本节为您介绍使用代码托管提供的代码仓库模板“Java Maven Demo”、编译构建提供的构建模板“Maven”快速完成Maven构建的基本操作流程。

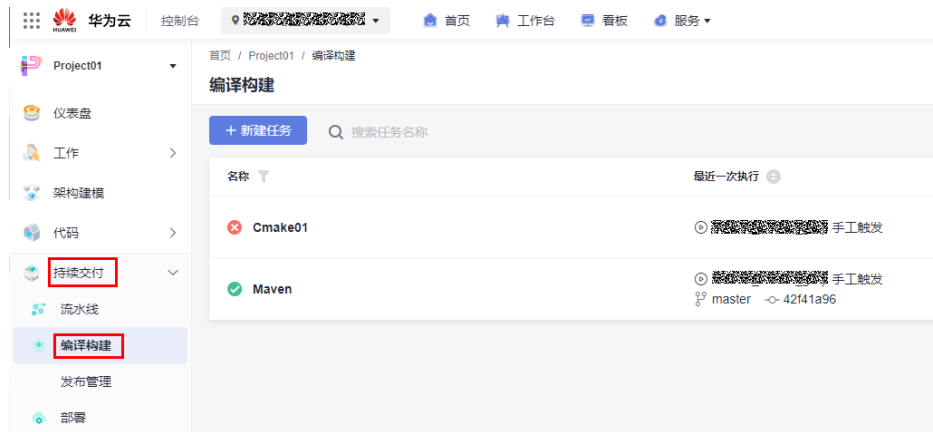
前提条件

- 已[注册华为账号并开通华为云](#)。
- 已有可用项目，如果没有，请先[新建项目](#)。
- 已在项目中新建代码仓库并上传代码，如果没有，请按如下步骤新建。
 - a. 使用华为云账号登录软件开发生产线首页。
 - b. 单击需要创建代码仓的项目名称。
 - c. 在页面导航栏中选择“代码 > 代码托管”。
 - d. 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。
 - e. 单击“下一步”，在“按模板新建 > 选择仓库模板”页面，搜索并选择“Java Maven Demo”模板。
 - f. 单击“下一步”，填写仓库名称等信息，然后单击“确定”完成仓库创建。

- g. [上传代码至代码仓库](#)。

新建编译构建任务

步骤1 在页面导航栏选择“持续交付 > 编译构建”，进入编译构建服务。



步骤2 单击“新建任务”，进入配置“基本信息”页面。

步骤3 填写参数后，单击“下一步”。

表 1-2 基本信息配置参数

参数名称	参数说明
任务名称	自定义任务名称。
源码源	选择“Repo”。
源码仓库	选择 前提条件 中创建的代码仓库名称。
分支	选择 前提条件 创建代码仓时创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 选择“Maven”模板。单击“下一步”。

步骤5 进入“构建步骤”页签，根据需要自定义配置[构建步骤](#)。

步骤6 配置完成后，单击“新建”，完成任务创建。

----结束

执行编译构建任务

介绍如何执行构建任务以及查看构建结果。

步骤1 单击构建任务名称。

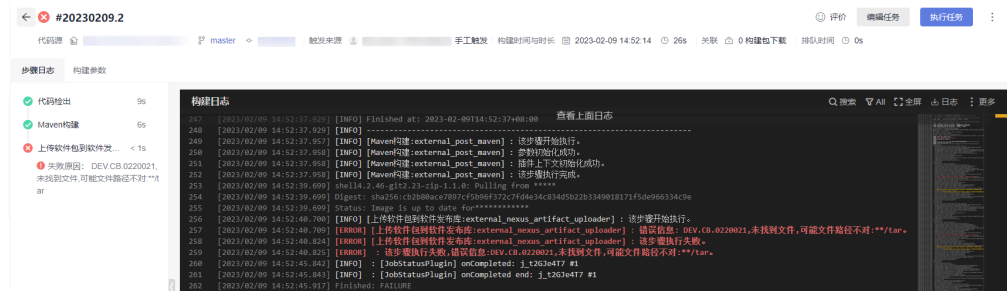
步骤2 进入“构建历史”页面，单击右上角“执行”，启动构建任务。

- 若出现如[图1-3](#)页面，表示任务执行成功。

图 1-3 任务构建成功



- 若任务执行失败，可通过页面提示信息或分析日志排查解决。

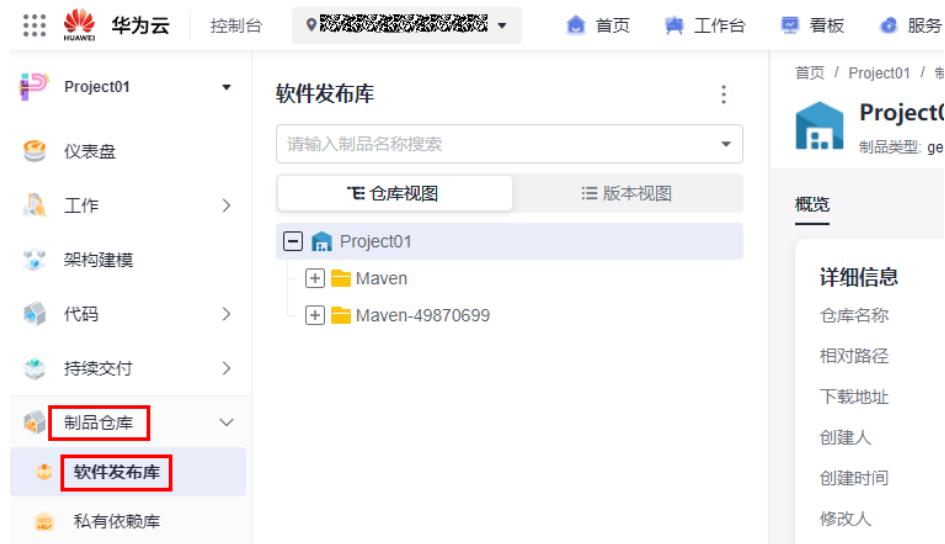


----结束

查看构建包

使用默认配置构建生成的构建包，存储在软件发布库“构建名/构建时间”目录。本节介绍如何查看构建包、验证发布结果。

步骤1 选择页面导航栏“制品仓库 > 软件发布库”。



步骤2 在软件发布库查看发布的软件包。软件包与新建编译构建任务时的任务名称一致，如图1-4所示。

图 1-4 查看软件包



---结束

资源清理

为了避免不必要的费用产生，完成本示例体验后，建议释放构建相关资源。

- 代码托管：删除代码仓库。
- 软件发布库：[删除软件包](#)，并[清空回收站](#)。

须知

资源释放后无法恢复，请谨慎操作。

1.3 快速上手 Npm 构建

编译构建提供多种构建模板，其中“Npm”模板可以对Node.JS项目进行构建打包。

本节为您介绍使用系统提供的代码仓库模板“Nodejs Webpack Demo”、构建模板“Npm”快速完成Npm构建的基本操作流程。

前提条件

- [已注册华为账号并开通华为云](#)。
- 已有可用项目，如果没有，请先[新建项目](#)。
- 已在项目中创建模板代码仓库，如果没有，请按如下步骤创建。
 - a. 使用华为云账号登录软件开发生产线首页。
 - b. 单击需要创建代码仓的项目名称。
 - c. 在页面导航栏中选择“代码 > 代码托管”。
 - d. 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。
 - e. 单击“下一步”在“按模板新建 > 选择模板”页面，搜索并选择“Nodejs Webpack Demo”模板。
 - f. 单击“下一步”，填写仓库名称等信息，然后单击“确定”完成仓库创建。

新建编译构建任务

步骤1 在页面导航栏选择“持续交付 > 编译构建”，进入编译构建服务。



步骤2 单击“新建任务”，进入配置“基本信息”页面。

步骤3 填写参数后，单击“下一步”。

表 1-3 基本信息配置参数

参数名称	参数说明
任务名称	自定义任务名称。
源码源	选择“Repo”。
源码仓库	选择前提条件中创建的代码仓库名称。
分支	选择前提条件创建代码仓时创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 构建模板选择“Npm”，单击“下一步”。

步骤5 进入“构建步骤”页签，根据需要自定义配置构建步骤。

步骤6 配置完成后，单击“新建”，完成任务创建。

----结束

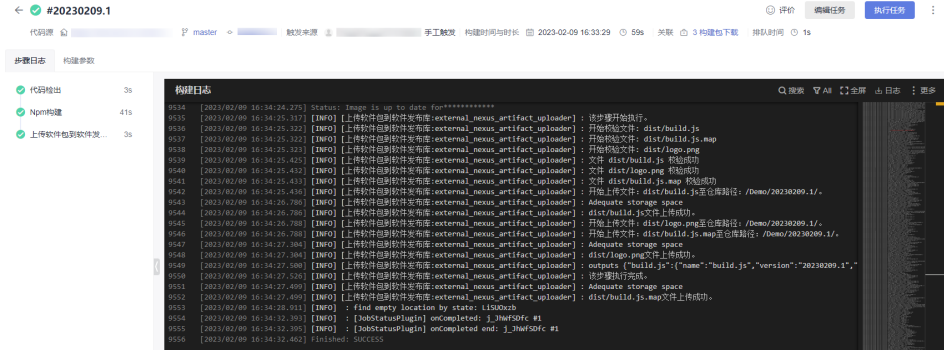
执行编译构建任务

介绍如何执行Npm模板构建任务以及查看构建结果。

步骤1 单击构建任务名称。

步骤2 进入“构建历史”页面，单击右上角“执行任务”，启动构建任务。

- 若出现如下页面，表示任务执行成功。



- 若任务执行失败，可通过页面提示信息或分析日志排查解决。

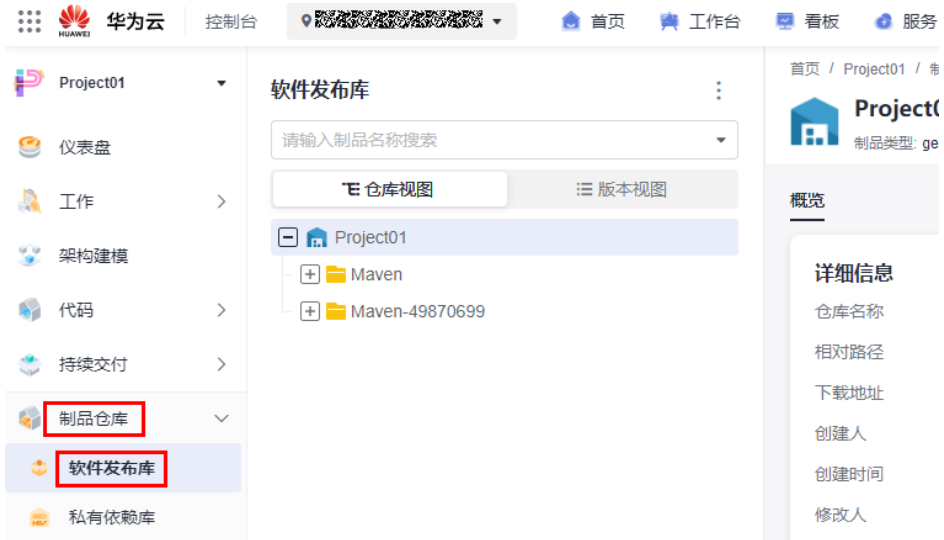


----结束

查看构建包

使用Npm模板默认配置构建生成的构建包，存储在软件发布库“构建名/构建时间”目录。本节介绍如何查看构建包、验证发布结果。

步骤1 选择页面导航栏“制品仓库 > 软件发布库”。



步骤2 在软件发布库查看发布的软件包。软件包与新建编译构建任务时的任务名称一致。

----结束

资源清理

为了避免不必要的费用产生，完成本示例体验后，建议释放构建相关资源。

- 代码托管：删除代码仓库。
- 软件发布库：删除软件包，并清空回收站。

须知

资源释放后无法恢复，请谨慎操作。

1.4 快速上手 Android 构建

编译构建提供多种构建模板，其中“Android APK”模板可以对Android项目进行构建打包。

本节为您介绍使用系统提供的代码仓库模板“Android Gradle Demo”、构建模板“Android APK”快速完成Android构建的基本操作流程。

前提条件

- 已注册华为账号并开通华为云。
- 已有可用项目，如果没有，请先新建项目。
- 已在项目中创建模板代码仓库，如果没有，请按如下步骤创建。
 - a. 使用华为云账号登录软件开发生产线首页。
 - b. 单击需要创建代码仓的项目名称。
 - c. 在页面导航栏中选择“代码 > 代码托管”。
 - d. 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。
 - e. 单击“下一步”，在“按模板新建 > 选择仓库模板”页面，搜索并选择“Android Gradle Demo”模板。
 - f. 单击“下一步”，填写仓库名称等信息，然后单击“确定”完成仓库创建。

新建编译构建任务

步骤1 在页面导航栏选择“持续交付 > 编译构建”，进入编译构建服务。



步骤2 单击“新建任务”，进入配置“基本信息”页面。

步骤3 填写参数后，单击“下一步”。

表 1-4 基本信息配置参数

参数名称	参数说明
任务名称	自定义任务名称。
源码源	选择“Repo”。
源码仓库	选择前提条件中创建的代码仓库名称。
分支	选择前提条件创建代码仓时创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 选择实际需要使用的“构建模板”。单击“下一步”。

步骤5 进入“构建步骤”页签，根据需要自定义配置构建步骤。

步骤6 配置完成后，单击“新建”，完成任务创建。

----结束

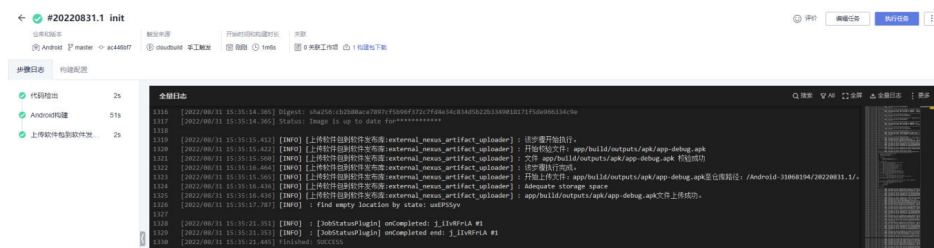
执行编译构建任务

介绍如何执行Android APK模板构建任务以及查看构建结果。

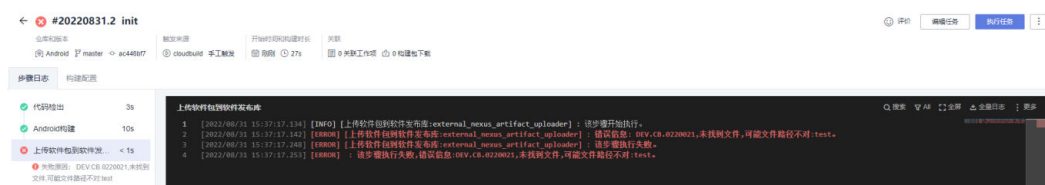
步骤1 单击构建任务名称。

步骤2 在“构建历史”页面，单击右上方“执行”，启动构建任务。

- 若出现如下页面，表示任务执行成功。



- 若任务执行失败，可通过页面提示信息或分析日志排查解决。

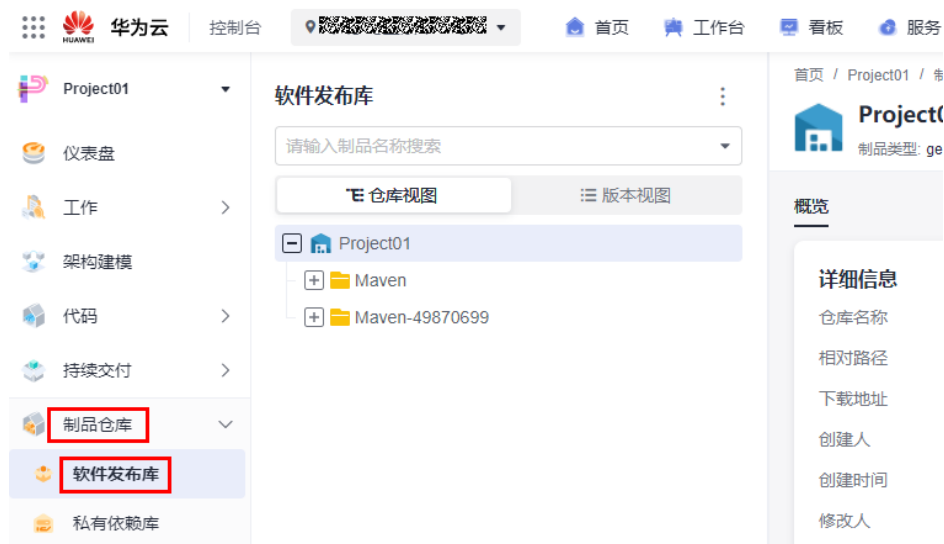


----结束

查看构建包

使用Android APK模板默认配置构建生成的构建包，存储在软件发布库“构建名/构建时间”目录。本节介绍如何查看构建包、验证发布结果。

步骤1 选择页面导航栏“制品仓库 > 软件发布库”。



步骤2 在软件发布库查看发布的软件包。软件包与**新建编译构建任务**时的任务名称一致。

---结束

资源清理

为了避免不必要的费用产生，完成本示例体验后，建议释放构建相关资源。

- 代码托管：删除代码仓库。
- 软件发布库：**删除软件包**，并**清空回收站**。

须知

资源释放后无法恢复，请谨慎操作。

1.5 快速上手 CMake 构建

编译构建提供多种构建模板，其中“CMake”模板可以对跨平台项目工程进行构建打包。

本节为您介绍使用系统提供的代码仓库模板“Cpp Demo”、构建模板“CMake”快速完成CMake构建的基本操作流程。

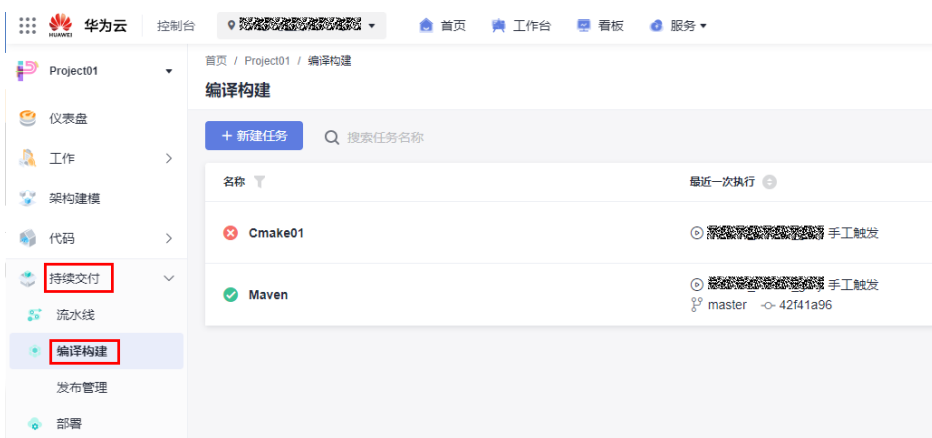
前提准备

- 已**注册华为账号并开通华为云**。
- 已有可用项目，如果没有，请先**新建项目**。

- 已在项目中创建模板代码仓库，如果没有，请按如下步骤创建。
 - a. 使用华为云账号登录软件开发生产线首页。
 - b. 单击需要创建代码仓的项目名称。
 - c. 在页面导航栏中选择“代码 > 代码托管”。
 - d. 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。
 - e. 单击“下一步”，在“按模板新建 > 选择仓库模板”页面，搜索并选择“Cpp Demo”模板。
 - f. 单击“下一步”，填写仓库名称等信息，然后单击“确定”完成仓库创建。

新建编译构建任务

步骤1 在页面导航栏选择“持续交付 > 编译构建”，进入编译构建服务。



步骤2 单击“新建任务”，进入配置“基本信息”页面。

步骤3 填写参数后，单击“下一步”。

表 1-5 基本信息配置参数

参数名称	参数说明
任务名称	自定义任务名称。
源码源	选择“Repo”。
源码仓库	选择前提条件中创建的代码仓库名称。
分支	选择前提条件创建代码仓时创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 构建模板选择“CMake”，单击“下一步”。

步骤5 进入“构建步骤”页签，根据需要自定义配置构建步骤。

步骤6 配置完成后，单击“新建”，完成任务创建。

----结束

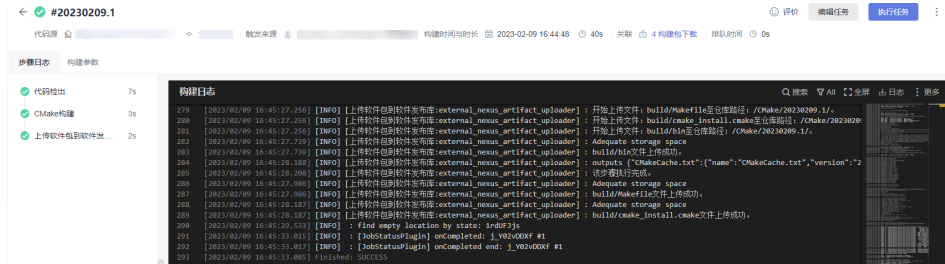
执行编译构建任务

介绍如何执行CMake模板构建任务以及查看构建结果。

步骤1 单击构建任务名称。

步骤2 进入“构建历史”页面，单击右上角“执行”，启动构建任务。

- 若出现如下页面，表示任务执行成功。



- 若任务执行失败，可通过页面提示信息或分析日志排查解决。

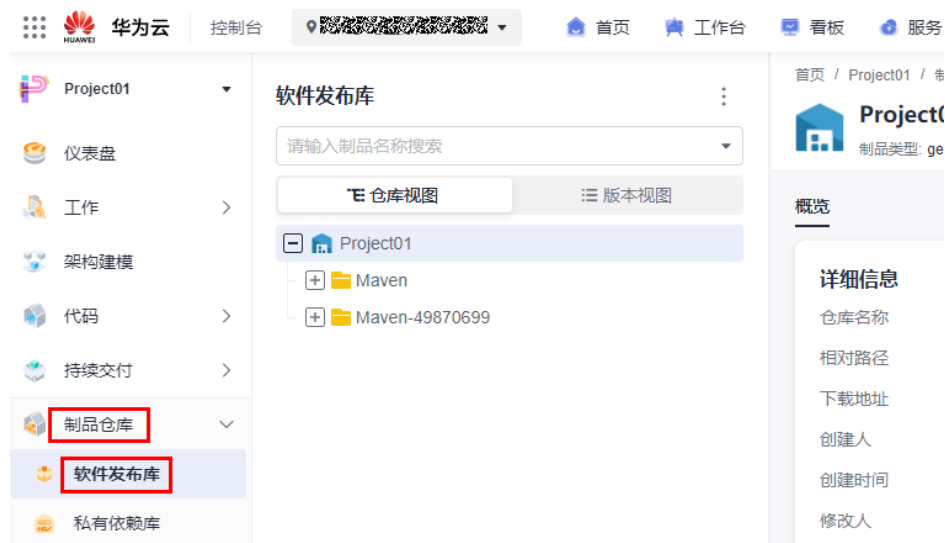


----结束

查看构建包

使用CMake模板默认配置构建生成的构建包，存储在软件发布库“构建名/构建时间”目录。本节介绍如何查看构建包、验证发布结果。

步骤1 选择页面导航栏“制品仓库 > 软件发布库”。



步骤2 在软件发布库查看发布的软件包。软件包与**新建编译构建任务**时的任务名称一致。

----结束

资源清理

为了避免不必要的费用产生，完成本示例体验后，建议释放构建相关资源。

- 代码托管：删除代码仓库。
- 软件发布库：[删除软件包](#)，并[清空回收站](#)。

须知

资源释放后无法恢复，请谨慎操作。

1.6 使用鲲鹏环境构建

以Cmake构建为例，介绍如何使用编译构建服务鲲鹏环境构建项目，并上传软件包到软件发布仓库。

说明

“鲲鹏（ARM）服务器”目前只有华北-北京四区域支持。

前提准备

- 已[注册华为账号并开通华为云](#)。
- 已有可用项目，如果没有，请先[新建项目](#)。
- 已在项目中创建模板代码仓库，如果没有，请按如下步骤创建。
 - a. 使用华为云账号登录软件开发生产线首页。
 - b. 单击需要创建代码仓的项目名称。
 - c. 在页面导航栏中选择“代码 > 代码托管”。
 - d. 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。
 - e. 单击“下一步”，在“按模板新建 > 选择仓库模板”页面，搜索并选择“Cpp Demo”模板。
 - f. 单击“下一步”，填写仓库名称等信息，然后单击“确定”完成仓库创建。

操作步骤

步骤1 在页面导航栏选择“持续交付 > 编译构建”，进入编译构建服务。



步骤2 单击“新建任务”，进入配置“基本信息”页面。

步骤3 填写参数后，单击“下一步”。

表 1-6 基本信息配置参数

参数名称	参数说明
任务名称	自定义任务名称。
源码源	选择“Repo”。
源码仓库	选择前提条件中创建的代码仓库名称。
分支	选择前提条件创建代码仓时创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 构建模板选择“CMake”，单击“下一步”。

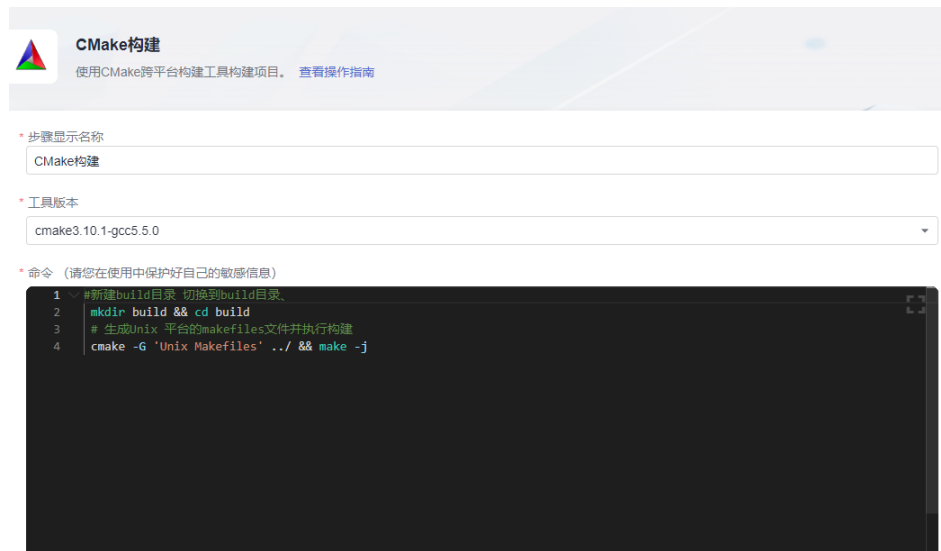
步骤5 进入“构建步骤”页签，配置“构建环境主机类型”为“鲲鹏（ARM）服务器”。

说明

在不同芯片架构上运行的软件，需要选择对应的环境主机。如软件最终在鲲鹏服务器上运行，则选择鲲鹏服务器。

步骤6 配置构建步骤“Cmake构建”。

- 根据需要选择“工具版本”。
- 配置Cmake命令，一般使用系统默认给出的命令即可。

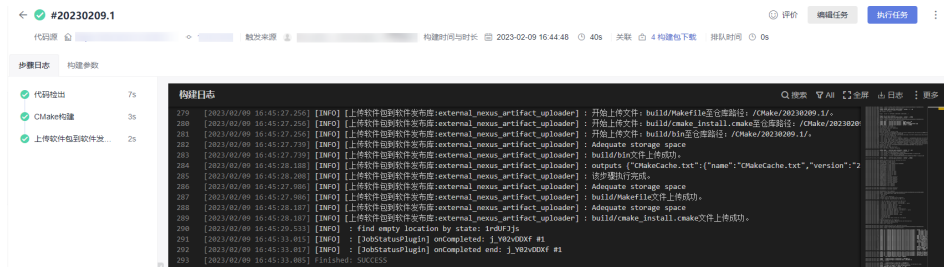


步骤7 配置构建步骤“上传软件包到软件发布库”。

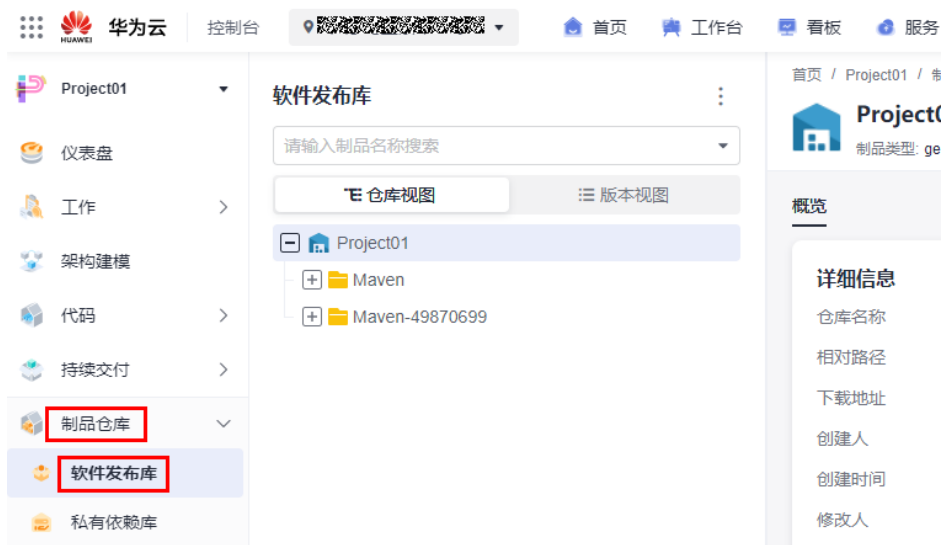
- 输入要上传到软件发布仓的构建包路径、版本号和包名。
- 构建包路径默认是通配符“build/*”格式，可以选择默认通配符格式，也可根据实际情况填写准确的路径，详细指导请参考[如何配置上传软件包到软件发布库](#)。

步骤8 配置完成后，单击“新建”，完成任务创建。

步骤9 出现以下页面时，表示构建成功。



步骤10 选择页面导航栏“制品仓库 > 软件发布库”。



步骤11 在软件发布库查看发布的软件包。软件包与**创建构建任务**时的任务名称一致
----结束

2 代码化构建

2.1 Maven 构建任务

场景概述

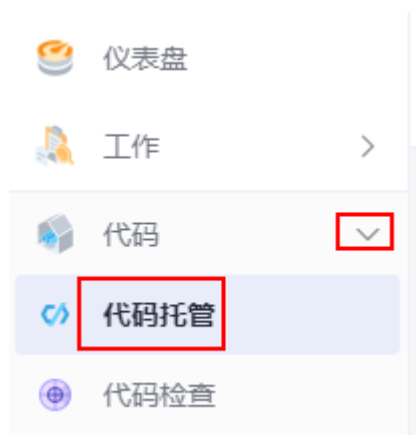
编译构建服务支持通过yaml文件配置构建脚本，用户可以将构建时需要配置的构建环境、构建参数、构建命令、构建步骤等操作，通过yaml语法编写成build.yml文件实现，并且将build.yml文件和被构建的代码一起存储到代码仓库。执行构建任务时，系统会以build.yml文件作为构建脚本执行构建任务，使构建过程可追溯、可还原，安全可靠。本节已使用Maven构建为例。

前提条件

- 已[注册华为账号并开通华为云](#)。
- 已有可用项目，如果没有，请[新建项目](#)。

新建代码仓

- 步骤1** 使用华为云账号登录软件开发生产线首页。
- 步骤2** 单击需要创建代码仓的项目名称。
- 步骤3** 在页面左侧目录中选择“代码 > 代码托管”。



步骤4 进入代码托管页面，单击“新建仓库”，选择“普通仓库”。

步骤5 根据表2-1填写参数，单击“确定”。

表 2-1 新建代码仓

参数名称	参数说明
代码仓库名称	自定义代码仓名称。例如：maven_yml_build。 <ul style="list-style-type: none">以数字、字母或者“_”开头。可包含“.”和“-”。不能以“.git”、“.atom”或者“.”结尾。
描述	对代码仓的描述。
选择 gitignore	根据编程语言选择“.gitignore”，例如：Java。
权限设置	勾选全部。 <ul style="list-style-type: none">允许项目内人员访问仓库：选择后会自动将项目中的项目经理设为仓库管理员，开发人员设为仓库普通成员。当项目新增这两个角色时，也会自动同步到已经存在的仓库中。允许生成README文件：可以通过编辑README文件，记录项目的架构、编写目的等信息，相当于对整个仓库的一种注释。自动创建代码检查任务（免费）：仓库创建完成后在代码检查任务列表中，可看到对应仓库的检查任务。
是否公开	设置为“私有”。 <ul style="list-style-type: none">私有：仓库仅对仓库成员可见，仓库成员可访问仓库或者提交代码。公开只读：仓库对所有访客公开只读，但不出现在访客的仓库列表及搜索中，您可以选择开源许可证作为备注。

----结束

创建 build.yml 文件

步骤1 在页面导航中选择“代码 > 代码托管”。

步骤2 单击新建代码仓中创建的代码仓名称。

步骤3 单击“新建 > 新建目录”，如图2-1所示。

图 2-1 新建目录



步骤4 根据表2-2填写参数信息，单击“确定”。

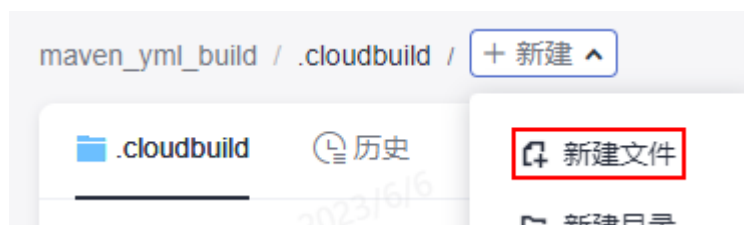
表 2-2 新建目录

参数名称	参数说明
目录名称	可自定义，例如“.cloudbuild”。
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。

步骤5 单击步骤4创建的目录名称。

步骤6 单击“新建 > 新建文件”，如图2-2所示。

图 2-2 新建文件



步骤7 文件命名为“build.yml”，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
      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8 单击“确定”。

----结束

创建 Java 文件

步骤1 参考步骤4，创建名为“src/main/java”的目录。

步骤2 在“src/main/java”的目录下，参考步骤6和步骤7，创建名为“HelloWorld.java”的文件。文件中代码如下：

```
/**
 * Hello world
```



```
*  
*/  
  
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

步骤3 单击“确定”。

----结束

创建 Dockerfile 文件

步骤1 在根目录下，参考**步骤6**和**步骤7**，创建名为“Dockerfile”的文件。文件中代码如下：

```
FROM swr.cn-north-5.myhuaweicloud.com/codeci/special_base_image:centos7-base-1.0.2-in  
MAINTAINER <devcloud@demo.com>  
USER root  
RUN mkdir /demo  
COPY ./target/server-1.0.jar /demo/app.jar
```

步骤2 单击“确定”。

----结束

创建 pom.xml 文件

步骤1 在根目录下，参考**步骤6**和**步骤7**，创建名为“pom.xml”的文件。文件中代码如下：

```
<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/maven-  
v4_0_0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
    <groupId>com.huawei.demo</groupId>  
    <artifactId>server</artifactId>  
    <packaging>jar</packaging>  
    <version>1.0</version>  
    <name>server</name>  
    <url>http://maven.apache.org</url>  
    <dependencies>  
        <dependency>  
            <groupId>junit</groupId>  
            <artifactId>junit</artifactId>  
            <version>3.8.1</version>  
            <scope>test</scope>  
        </dependency>  
    </dependencies>  
  
    <build>  
        <pluginManagement>  
            <plugins>  
                <plugin>  
                    <groupId>org.apache.maven.plugins</groupId>  
                    <artifactId>maven-jar-plugin</artifactId>  
                    <version>2.6</version>  
                    <configuration>  
                        <archive>  
                            <manifest>  
                                <addClasspath>true</addClasspath>  
                            </manifest>  
                            <manifestEntries>  
                                <Main-Class>  
                                    HelloWorld  
                                </Main-Class>  
                            </manifestEntries>  
                        </archive>  
                    </configuration>  
                </plugin>  
            </plugins>  
        </pluginManagement>  
    </build>  
</project>
```

```
</Main-Class>  
</manifestEntries>  
</archive>  
</configuration>  
</plugin>  
</plugins>  
</pluginManagement>  
</build>  
</project>
```

步骤2 单击“确定”。

----结束

创建构建任务

步骤1 在页面导航中选择“持续交付 > 编译构建”，如图2-3所示。

图 2-3 编译构建首页



步骤2 单击“新建任务”。

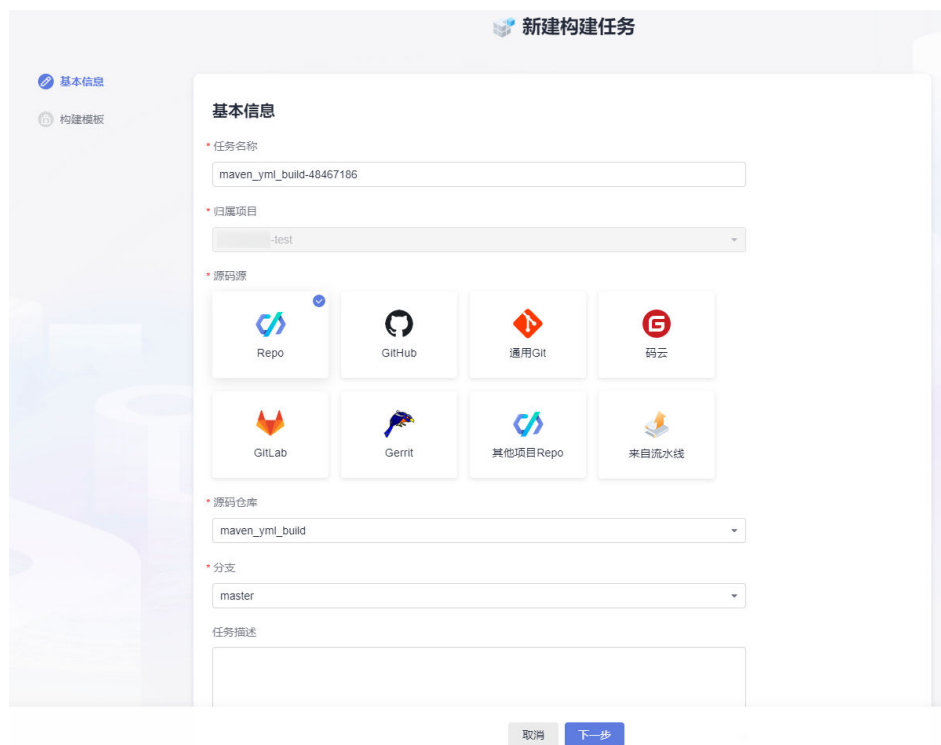
步骤3 根据表2-3填写参数信息，如图2-4所示。

表 2-3 基本信息配置

参数名称	参数说明
任务名称	自定义任务名称，例如：maven_yml_build。
源码源	选择“Repo”。
源码仓库	选择新建代码仓中创建的代码仓库名称。
分支	选择新建代码仓中创建的分支，若没有创建，选择默认“master”即可。

参数名称	参数说明
任务描述	对该构建任务的描述。

图 2-4 创建构建任务



步骤4 单击“下一步”。

步骤5 选择“空白构建模板”，单击“下一步”。

步骤6 单击“代码化”页签，可查看到导入的构建脚本，如图2-5所示。

图 2-5 代码化页签



步骤7 单击页面右上角的“新建并执行”。

----结束

查看并验证构建结果

- 查看上传的软件包。
 - 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。软件包与**创建构建任务**时的任务名称一致，如**图2-6**所示。

图 2-6 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**创建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**创建build.yml文件**时代码中填写的“镜像名称”，如：maven_demo，如**图2-7**所示。

图 2-7 筛选镜像

2.2 Npm 构建任务

场景概述

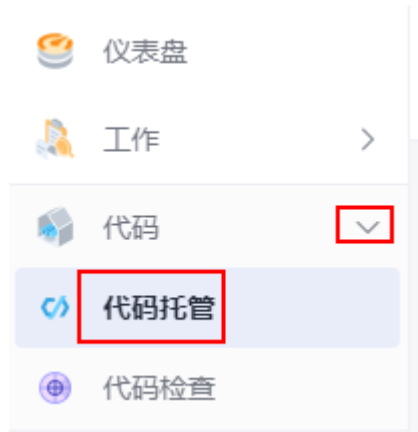
编译构建服务支持通过yaml文件配置构建脚本，用户可以将构建时需要配置的构建环境、构建参数、构建命令、构建步骤等操作，通过yaml语法编写成build.yml文件实现，并且将build.yml文件和被构建的代码一起存储到代码仓库。执行构建任务时，系统会以build.yml文件作为构建脚本执行构建任务，使构建过程可追溯、可还原，安全可靠。本节已使用Npm构建为例。

前提条件

- 已**注册华为账号并开通华为云**。
- 已有可用项目，如果没有，请**新建项目**。

新建代码仓

- 步骤1** 使用华为云账号登录软件开发生产线首页。
- 步骤2** 单击需要创建代码仓的项目名称。
- 步骤3** 在页面左侧目录中选择“代码 > 代码托管”。



步骤4 进入代码托管页面，单击“新建仓库”，选择“普通仓库”。

步骤5 根据表2-4填写参数，单击“确定”。

表 2-4 新建代码仓

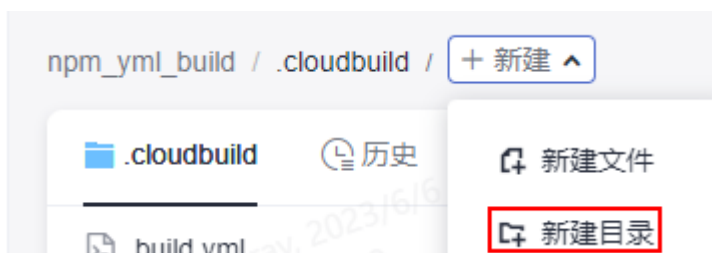
参数名称	参数说明
代码仓库名称	自定义代码仓名称。例如：npm_yml_build。 <ul style="list-style-type: none">以数字、字母或者“_”开头。可包含“.”和“-”。不能以“.git”、“.atom”或者“.”结尾。
描述	对代码仓的描述。
选择gitignore	根据编程语言选择“.gitignore”，例如：Java。
权限设置	勾选全部。 <ul style="list-style-type: none">允许项目内人员访问仓库：选择后会自动将项目中的项目经理设为仓库管理员，开发人员设为仓库普通成员。当项目新增这两个角色时，也会自动同步到已经存在的仓库中。允许生成README文件：可以通过编辑README文件，记录项目的架构、编写目的等信息，相当于对整个仓库的一种注释。自动创建代码检查任务（免费）：仓库创建完成后在代码检查任务列表中，可看到对应仓库的检查任务。
是否公开	设置为“私有”。 <ul style="list-style-type: none">私有：仓库仅对仓库成员可见，仓库成员可访问仓库或者提交代码。公开只读：仓库对所有访客公开只读，但不出现在访客的仓库列表及搜索中，您可以选择开源许可证作为备注。

----结束

创建 build.yml 文件

- 步骤1 在页面导航中选择“代码 > 代码托管”。
- 步骤2 单击新建代码仓中创建的代码仓名称。
- 步骤3 单击“新建 > 新建目录”，如图2-8所示。

图 2-8 新建目录



- 步骤4 根据表2-5填写参数信息，单击“确定”。

表 2-5 新建目录

参数名称	参数说明
目录名称	可自定义，例如“.cloudbuild”。
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。

- 步骤5 单击步骤4创建的目录名称。
- 步骤6 单击“新建 > 新建文件”，如图2-9所示。

图 2-9 新建文件



- 步骤7 文件命名为“build.yml”，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:
    - npm:
      inputs:
        #check:
        #project_dir: .
      command: |
        export PATH=$PATH:-/.npm-global/bin
        #设置缓存目录
        npm config set cache /npmcache
        npm config set registry http://mirrors.tools.huawei.com/npm/
        npm config set disturl http://mirrors.tools.huawei.com/nodejs
        npm config set sass_binary_site http://mirrors.tools.huawei.com/node-sass/
        npm config set phantomjs_cdnurl http://mirrors.tools.huawei.com/phantomjs
```

```
npm config set chromedriver_cdnurl http://mirrors.tools.huawei.com/chromedriver
npm config set operadriver_cdnurl http://mirrors.tools.huawei.com/operadriver
npm config set electron_mirror http://mirrors.tools.huawei.com/electron/
npm config set python_mirror http://mirrors.tools.huawei.com/python
npm config set prefix '~/.npm-global'
npm install --verbose
zip -r ./nodeserver.zip ./
- upload_artifact:
  inputs:
    path: "./nodeserver.zip"
```

步骤8 单击“确定”。

----结束

创建 package.json 文件

步骤1 在根目录下，参考**步骤6**和**步骤7**，创建名“package.json”的文件。文件中代码如下：

```
{
  "name": "docker_web_app",
  "version": "1.0.0",
  "description": "Node.js on Docker",
  "author": "First Last <first.last@example.com>",
  "main": "server.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^4.16.1"
  }
}
```

步骤2 单击“确定”。

----结束

创建 server.js 文件

步骤1 在根目录下，参考**步骤6**和**步骤7**，创建名为“server.js”的文件。文件中代码如下：

```
'use strict';
const express =require('express');
// Constants
const PORT=8080;
const HOST='127.0.0.1';
// App
const app =express();
app.get('/',(req, res)=>{
  res.send('Hello world\n');
});
app.listen(PORT,HOST);
console.log(`Running on http://${HOST}:${PORT}`);
```

步骤2 单击“确定”。

----结束

创建构建任务

步骤1 在页面导航中选择“持续交付 > 编译构建”，如**图2-10**所示。

图 2-10 编译构建首页



步骤2 单击“新建任务”。

步骤3 根据表2-6填写参数信息。

表 2-6 基本信息配置

参数名称	参数说明
任务名称	自定义任务名称，例如：npm_yml_build。
源码源	选择“Repo”。
源码仓库	选择新建代码仓中创建的代码仓库名称。
分支	选择新建代码仓中创建的分支，若没有创建，选择默认“master”即可。
任务描述	对该构建任务的描述。

步骤4 单击“下一步”。

步骤5 选择“空白构建模板”，单击“下一步”。

步骤6 单击“代码化”页签，可查看到导入的构建脚本，如图2-11所示。

图 2-11 代码化页签



步骤7 单击页面右上角的“新建并执行”。

----结束

查看并验证构建结果

步骤1 选择页面导航栏“制品仓库 > 软件发布库”。

步骤2 在软件发布库查看发布的软件包。软件包与**创建构建任务**时的任务名称一致，如图 2-12所示。

图 2-12 查看软件包



----结束

3 入门实践

当用户开通编译构建服务后，可以根据不同的业务场景，使用图形化或者代码化的方式进行构建。

本文介绍常见的图形化构建实践。

图形化构建

实践	描述
使用Node.js构建包制作Docker镜像	使用软件开发生产线编译构建服务打包Node.js项目及制作构建包的Docker镜像。