

# 云原生构建 操作指南



腾讯云

## 【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

# 文档目录

## 操作指南

个人中心

组织管理

代码仓库

云原生开发

云原生构建

制品管理

任务集



进入 CNB，单击右上角用户头像，选择**个人设置**。查看个人资料，设置账号用户名等。

## 个人资料

- 昵称、头像获取自登录的微信。可在微信扫码登录，选择身份时，自行选择。
- 个人签名/个人网站/微信公众号处可自定义填写，填写后，将展示在个人主页。
- 赞赏码处可上传微信赞赏码，具体操作请参见 [赞赏码配置](#)。

**个人资料**

昵称

个人签名

个人网站

微信公众号

**个人头像**

**赞赏码**

## 账号设置

用户名为用户在 CNB 的唯一标识，一年仅可修改一次。

**账号设置**

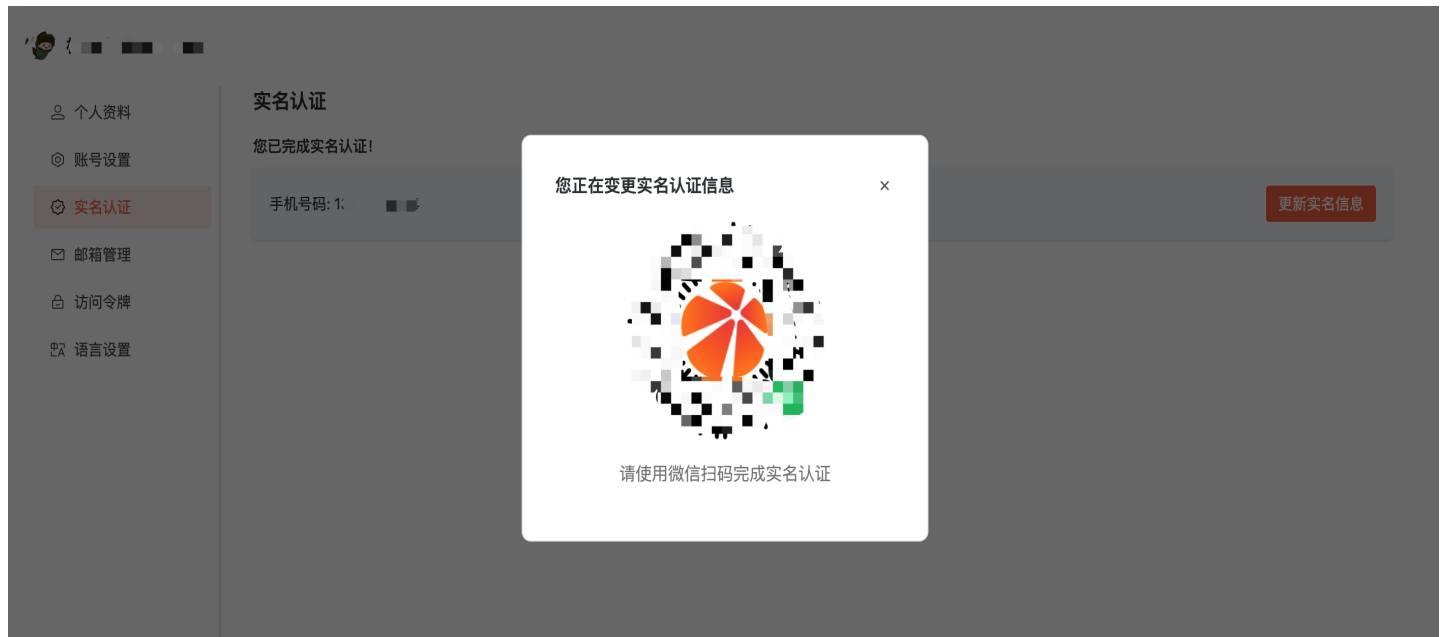
用户名

距离上次变更一年以内，将禁止变更用户名，下方会提示下次可修改的时间。



## 实名认证

由于产品存在 UGC 社区属性，根据相关法律规定，未经过实名认证的用户，将禁止写行为，例如创建 Issue。在个人设置 > 实名认证中，微信扫码授权云原生构建获取您微信关联的手机号码，即可完成实名认证。



## 邮箱管理

可以登记自己的邮箱，以便于平台匹配 commit author 的 email，正确渲染 commit 的作者信息。此外，还可以配置提交邮箱，配置后，在页面或者云原生开发的新建 commit 操作时，将使用该邮箱作为 commit author 的 email 信息。

通知邮箱用于接收系统通知邮件，配置后，系统通知将同步发送至该通知邮箱。



个人资料

账号设置

实名认证

邮箱管理

访问令牌

语言设置

## 邮箱管理

### 邮箱列表

所有登记的邮箱，都可用于匹配 Git 提交的 author

AJFORrKqorwI4VHjhwaU. [redacted]@noreply.cnb.cool 系统邮箱

添加邮箱

### 提交邮箱

该邮箱将作为页面或云原生开发中 Git 操作的 author

### 通知邮箱

接收系统通知邮件

## 访问令牌

云原生构建为微信扫码登录注册，没有登录密码。

## 创建令牌

在 Git、制品库、OpenApi 的鉴权场景，使用访问令牌可以完成鉴权行为。

单击添加访问令牌，在创建时，可以通过常见场景快速初始化令牌授权范围，也可以在授权范围中逐个勾选。

个人资料

账号设置

实名认证

邮箱管理

访问令牌

语言设置

## 访问令牌 / 添加访问令牌

## 令牌名

## 到期时间

永久有效

此令牌永不过期

## 使用范围

 仅公开仓库/制品库  全部仓库/制品库  指定仓库

## 常见场景

勾选常见场景, 快速初始化授权范围

## Git 客户端凭据

勾选后, 将授予令牌仓库代码的读写权限, 适用于 Git https 操作时的密码

## ISSUE 管理

勾选后, 将授予令牌仓库 ISSUE 的读写权限, 适用于授权给流水线或机器人, 完成 ISSUE 的管理行为

## 合并请求管理

勾选后, 将授予令牌仓库合并请求的读写权限, 适用于授权给流水线或机器人, 完成合并请求的管理行为

## 制品库

勾选后, 将授予令牌仓库制品的读写权限, 适用于授权给流水线或机器人, 完成制品的推送

## 授权范围

授予令牌在账号下的可操作权限

&gt; repo

&gt; registry

&gt; account

&gt; group

创建

取消

## 令牌鉴权逻辑

在使用令牌鉴权时, 会同时校验以下两种, 两者同时满足, 才会鉴权通过, 执行此行为。

- 令牌绑定用户的权限是否可以完成此行为。
- 该令牌的授权范围是否包含此行为。

例如:

当使用 `用户 A` 的令牌, 拉取 `repo-A` 时, 将同时校验 `用户 A` 是否可以拉取 `repo-A`, 以及该令牌授权范围是否包含拉取代码仓库。

同时满足, 才会允许该令牌拉取此仓库。

## 语言设置

平台已适配简体中文和英文, 用户可按需自行切换。



- 个人资料
- 账号设置
- 实名认证
- 邮箱管理
- 访问令牌
- 语言设置**

### 语言设置

语言

简体中文

跟随浏览器语言

简体中文

English

# 组织管理

最近更新时间：2025-02-26 17:57:23

组织是用于管理团队成员和资源的命名空间。在创建仓库资源之前，需要先创建一个组织来管理成员和仓库。

## 创建组织

进入 [CNB](#)，单击右上角的“+”，选择**创建组织**，填写组织名称及相关描述后，单击**创建**即可完成组织创建。

The screenshot shows the '创建组织' (Create Organization) page. At the top right, there is a search bar and a '+' button. A dropdown menu is open, showing several options: '创建仓库', '创建 Maven 制品库', '创建 npm 制品库', '创建 ohpm 制品库', '创建组织' (highlighted with a red box), and '创建任务集'. Below the dropdown, there are three input fields: '组织名称 \*' (Organization Name), '组织别名' (Organization Alias), and '组织描述' (Organization Description). The '组织名称' field contains 'https://cnb.cool/' and '请输入组织名称'. The '组织别名' field contains '请输入组织别名'. The '组织描述' field contains '请输入组织描述' and a character count '0/200'. At the bottom, there are two buttons: '创建' (Create) and '取消' (Cancel).


## 子组织

CNB 支持多级组织，方便映射大型研发团队的组织架构，隔离不同业务部门，便于各部门自行管理成员及资源。进入组织后，选择**子组织** > **创建子组织**可以创建子组织。

**说明：**  
组织的最大层级为10级。

## 创建组织

所属组织 \*

 git-use ▼

组织名称 \*

https://cnb.cool/git-use/ 请输入组织名称 

组织名称必填

组织别名

请输入组织别名 0/50

组织描述

请输入组织描述 0/200

创建

取消

## 成员

CNB 按角色授权，固定为五种角色，子组织、仓库、制品库将自动继承父级组织的成员权限关系。

Teewee

组织概览 < 代码仓库 2 子组织 任务集 制品库 **成员 1** 组织设置

组织成员(1)  + 邀请成员

成员	角色	加入时间	操作
	负责人	2024-08-21 11:27:37	...

共 1 条数据

**负责人**

拥有组织和仓库的全部管理权限

**管理员**

可管理组织、仓库及成员，但不能执行删除组织/仓库等危险操作

**开发者**

可提交代码、推送制品，处理 Issue、合并请求

**助手**

可访问代码，管理 Issue 和合并请求，但不能执行合并操作

**访客**

不可访问代码，可创建 Issue、可拉取制品

© 2025 CNB, Inc. 深圳市南山区粤海街道麻岭社区科技中一路腾讯大厦35楼 cnb\_support@tencent.com +86-755-86013388

关于我们 隐私声明 服务协议 API Docs 最佳实践 使用反馈 京ICP备11018762号-111

角色	说明
负责人	拥有组织和仓库的全部管理权限。
管理员	可管理组织、仓库及成员，但不能执行删除组织/仓库等危险操作。
开发者	可提交代码、推送制品，处理 Issue、合并请求。
助手	可访问代码，管理 Issue 和合并请求，但不能执行合并操作。
访客	不可访问代码，可创建 Issue、可拉取制品。

## CNB 权限逻辑

组织可以添加成员并配置成员角色，进行授权。

子组织或仓库，继承上层组织的成员角色，同时也可以直接添加成员。

在进行鉴权时，将对用户在所有父级组织中的权限取并集，即采用“最大角色”。

## 继承逻辑举例

组织的结构如下：

A (根组织)

|- B (A 的子组织)

|- C (所属于 B 的仓库)

用户 user-1 被添加至 A，并拥有负责人权限。未在 B、C 中单独添加该用户。

那么 user-1 在 A、B、C 均拥有负责人权限，因为子组织或仓库会继承父级成员，所以 user-1 在 B、C 也拥有权限。

## 组织设置

### 基础信息

基础信息主要为组织的公开信息，配置后将展示在组织主页中。

The screenshot shows the 'Teewee' organization settings interface. The top navigation bar includes the Teewee logo, a search bar, and a user profile icon. The main navigation menu contains: 组织概览, 代码仓库 (2), 子组织, 任务集, 制品库, 成员 (1), and 组织设置 (selected). The '组织设置' page is divided into a left sidebar and a main content area. The sidebar lists settings: 基础设置 (selected), 高级设置, 变更组织名, 计费管理, 管控, 组织安全, and 水印设置. The main content area is titled '组织信息' and includes: 组织别名 (input: Tee 站, 6/50), 组织描述 (input: 请输入组织描述, 0/200), 组织网站 (input: 请输入组织自有网站, 如: https://cnb.cool), and 微信公众号 (input: 请输入公众号的微信号或原始ID). Below the WeChat input is a note: '登录微信公众平台 (https://mp.weixin.qq.com), 设置与开发-公众号设置-注册信息-微信号/原始ID (例: cnb-\_-cool)'. A '更新组织信息' button is at the bottom. To the right, the '组织 LOGO' section shows a placeholder image with a '更改图片' button.

### 高级设置

管理组织危险操作，如转移组织和删除组织。注意需谨慎修改。

**说明:**

- 顶级组织不支持转移。
- 当前组织存在子组织或仓库时，无法删除。

组织概览 < 代码仓库 1 子组织 任务集 制品库 成员 1 **组织设置**

基础设置

**高级设置**

变更组织名

计费管理

管控

组织安全

水印设置

**危险操作****转移组织**

将该组织转移至另一个父级组织中，这会导致

- 所有子组织及所属仓库的变更
- 当前组织的公开性可能因适配新的父级组织，而发生变更
- 组织的继承成员，将以新的父级组织为准

转移组织

顶级组织不支持转移

**删除组织**

当前组织存在子组织或仓库，禁止删除！

删除组织

**变更组织名**

需注意，变更组织名会导致所有子组织及所属仓库的访问 URL 改变，请谨慎操作。

组织概览 < 代码仓库 1 子组织 任务集 制品库 成员 1 **组织设置**

基础设置

高级设置

**变更组织名**

计费管理

管控

组织安全

水印设置

**变更组织名**

! 变更组织名会导致所有子组织及所属仓库的访问 URL 改变，请谨慎操作！

原组织名

https://cnb.cool/ testqq



新组织名 \*

https://cnb.cool/ 请输入新组织名



变更

# 计费管理

统计整个组织的用量消耗，以及管理用量预算上限。

组织概览 > 代码仓库 1 > 子组织 > 任务集 > 制品库 > 成员 1 > **组织设置**

- 基础设置
- 高级设置
- 变更组织名
- 计费管理**
- 管控
- 组织安全
- 水印设置

计费管理 2025-02

云原生构建	用量详情
<b>免费配额</b> 已用 0.00 核时 / 上限 160.00 核时	0.00%
<b>公测限免配额</b> 已用 0.00 核时 / 上限 1,600.00 核时	0.00%
<b>付费用量</b> 已用 0.00 核时 / 上限 ∞ 核时	0.00%
云原生开发	用量详情
<b>免费配额</b> 已用 70.75 核时 / 上限 1,600.00 核时	4.42%
<b>公测限免配额</b> 已用 0.00 核时 / 上限 16,000.00 核时	0.00%
<b>付费用量</b> 已用 0.00 核时 / 上限 ∞ 核时	0.00%
代码仓库	用量详情
<b>免费配额</b> 已用 20.00 KiB / 上限 100.00 GiB	0.00%
<b>公测限免配额</b> 已用 0.00 B / 上限 100.00 GiB	0.00%
<b>付费用量</b> 已用 0.00 B / 上限 ∞ B	0.00%
对象存储 ①	用量详情

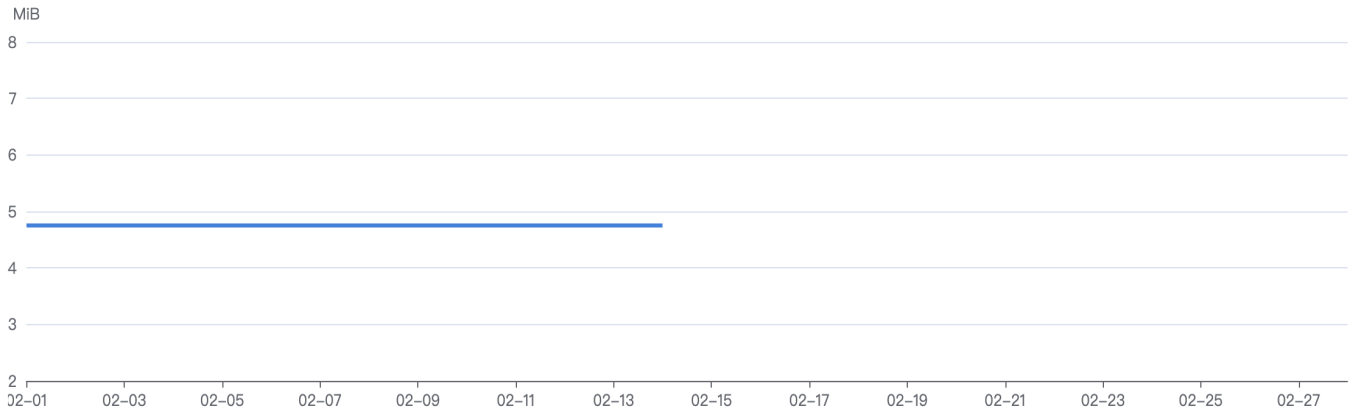
单击右上角用量详情，可查看该资源类型的用量走势和具体分布。

计费管理 / 对象存储用量详情

2025-02

已使用

**4.75 MiB**



Top 50 仓库

2025-02-14

仓库路径	总量	占比
xl. st	4.75 MiB	100.00%

## 组织安全

可以管理外部成员对子组织列表和成员列表的可见性。

- 组织概览
- <> 代码仓库 1
- 子组织
- 任务集
- 制品库
- 成员 1
- 组织设置**

- 基础设置
- 高级设置
- 变更组织名
- 计费管理
- 管控
- 组织安全**
- 水印设置

### 组织安全

隐藏子组织

开启后，外部成员无法查看子组织列表，子组织本身的访问不受影响。

隐藏组织成员

开启后，外部成员无法查看成员列表，且外部成员无法从成员主页查看到组织。



## 水印设置

支持设置组织下的私有仓库自动显示水印。水印内容为个人账号设置的用户名。

[组织概览](#)[代码仓库 1](#)[子组织](#)[任务集](#)[制品库](#)[成员 1](#)[组织设置](#)[基础设置](#)[高级设置](#)[变更组织名](#)[计费管理](#)[管控](#)[组织安全](#)[水印设置](#)

### 水印设置

私仓显示水印

开启后，组织下私有仓库将自动显示水印

# 代码仓库

最近更新时间：2025-02-26 17:57:23

## 创建仓库

### 步骤一：创建代码仓库

进入 **CNB**，单击右上角的“+”，选择**创建仓库**，选择仓库的归属组织、填写仓库名称，并按需选择仓库的公开性后，单击**创建**即可完成仓库创建。

创建仓库 按 下 搜索 +

### 创建仓库

仓库归属 \*

仓库名称 \*

URL预览

仓库主页 <https://cnb.cool/>

HTTPS <https://cnb.cool/>.git

仓库描述

添加仓库描述

0/350

公开性

公开 (任何人都可以查看, 适用于开源协作)

私有 (仅限仓库成员访问, 适用于团队内部协作)

密钥仓库 (仅允许页面查看和修改, 适用于云原生构建加载密钥)

- <> 创建仓库
- ✓ 创建 Maven 制品库
- 创建 npm 制品库
- 创建 ohpm 制品库
- 👤 创建组织
- 📁 创建任务集

### 步骤二：初始化仓库

您可以选择以下任一方式完成仓库的初始化。

#### 云端快速初始化（推荐）

您可以在云原生开发中执行相关命令迁移仓库，或直接新建文件来完成仓库的初始化。

## 云端快速初始化 👍 推荐

您可以在云原生开发中执行以下命令迁移仓库，或直接 [新建文件](#) 来完成仓库的初始化

```
cnb-init-from https://your-git.com/your-repo.git
```

[云原生开发](#)[官方示例](#)

## 本地初始化

可以通过裸库迁移、分支迁移、空仓初始化中任一方式来完成初始化。具体命令可在目标仓库查看，界面如下：

### 本地初始化

HTTPS

<https://cnb.cool/git-use/test.git>

#### 方式 1: 裸库迁移

在本地可以通过下列命令迁移 全部分支和 TAG 到本仓库，点击创建 [访问令牌](#)

```
mkdir empty && cd empty
git clone --bare https://your-git.com/group/name.git .
git lfs fetch origin --all
git push --mirror https://cnb.cool/git-use/test.git
```

#### 方式 2: 分支迁移

在本地可以通过下列命令迁移 当前分支 到本仓库，点击创建 [访问令牌](#)

```
git remote -v
git remote add cnb https://cnb.cool/git-use/test.git
git remote -v
git push -u cnb HEAD:main
```

#### 方式 3: 空仓初始化

在本地可以通过下列命令创建 全新仓库，点击创建 [访问令牌](#)

```
git init .
git remote add origin https://cnb.cool/git-use/test.git
git config --local user.name xiaohuan
git config --local user.email "your.email@example.com"
git config credential.helper store
```

## 克隆仓库

### 获取仓库克隆地址

进入目标仓库，单击克隆，可以获取仓库克隆地址。



## 编辑文件 .cnb.yml #3

🔧 开启中
👤 创建于 1 月前
for-... → main

[编辑](#)
云原生开发

[概览](#)
[提交历史 1](#)
[文件更改 1](#)

👤 创建于 1 月前
...

在 .cnb.yml 文件中新增了 main 部分的 push 配置

1. 新增了 main 部分
2. 在 push 中引用了 aosp\_build\_config 配置

[评审者](#)  
 暂未指定

[处理人](#)  
 暂未指定

[标签](#)  
 暂未指定

[参与者](#)  
👤

🔧
✓ 状态检查: 已通过 (6/6)
↻

- cnb/pull\_request/pipeline-1 success [34s]
- cnb/pull\_request/pipeline-2 success [43s]
- cnb/pull\_request/pipeline-3 success [33.7s]
- cnb/pull\_request/pipeline-4 success [43.6s]
- cnb/pull\_request/pipeline-5 success [44.6s]
- cnb/pull\_request/pipeline-6 success [33.8s]

git-clone-yyds 24.8s, 124.42 GB
Pipeline .cnb.yml
pull\_request success, 48.3s

✓ 源分支与目标分支没有冲突

直接合并 (Merge)
▼

## 仓库设置

### 基础设置

配置仓库的基础信息，以及配置仓库全局的推送限制。

基础设置

高级设置

权限

仓库成员

功能

分支保护

合并请求

云原生开发

云原生构建

用量

用量统计

基础设置

仓库名

monorepo 修改

URL预览

仓库主页 <https://cnb.cool/aosp/monorepo>  
 HTTPS <https://cnb.cool/aosp/monorepo.git>

默认分支

main

默认分支被视为仓库中的基本分支，除非您指定其他分支，否则将基于默认分支自动生成所有拉取请求和代码提交。

推送限制 (PUSH)

单次可以更新的分支和标签的个数  
 如果单次推送超过设置的分支和标签数量，推送将被拒绝

仅仓库负责人和管理员可以推送 Tag 及创建 Release

高级设置

管理仓库的危险操作，需谨慎配置。

[基础设置](#)[高级设置](#)[权限](#)[仓库成员](#)[功能](#)[分支保护](#)[合并请求](#)[云原生开发](#)[云原生构建](#)[用量](#)[用量统计](#)

## 危险操作

### 修改仓库可见性

本仓库当前为公开仓库，公开仓库对整个社区可见。可变更为私有仓库，仅对仓库成员可见。

[修改仓库可见性](#)

### 归档代码仓库

归档代码仓库后，开发者将无法继续访问仓库及对仓库进行任何写操作

[归档仓库](#)

### 转移代码仓库

将此代码仓库资源转移到其他组织

[转移仓库](#)

### 删除代码仓库

代码仓库删除后将无法恢复，操作需谨慎

[删除仓库](#)

## 仓库成员

仓库默认继承上级组织的所有成员权限关系。此外，还可单击[邀请成员](#)直接邀请成员加入当前仓库成为“仓库成员”或“外部协作者”。其中，外部协作者适用于一些临时参与协作的用户。

基础设置

高级设置

权限

仓库成员

功能

分支保护

合并请求

云原生开发

云原生构建

用量



用量统计

### 仓库成员

仓库成员 (2) | 继承成员 (4) | 外部协作者

搜索成员

+ 邀请成员

成员	角色	加入时间	操作
 [模糊]	负责人	2024-09-20 10:17	
 [模糊]	管理员	2024-12-17 20:03	...
共 2 条数据		20 条/页	< 1 >

## 分支保护

单击**新建保护规则**，通过设置分支保护规则，可配置指定分支的推送及合并策略，以防止未经报备的代码被更改。



◎ 基础设置

🔒 高级设置

权限

👤 仓库成员

功能

🛡️ 分支保护

🔄 合并请求

🔗 云原生开发

🔗 云原生构建

用量

📊 用量统计

## 分支保护 / 新建

您可以通过定义分支保护规则来限制推送及合并，防止分支被删除

## 分支名称

请输入希望保护的分支名称正则表达式

示例: \*-stable 或 production/\*

## 保护规则设置

 提交前需创建合并请求

启用后，所有提交都必须提交到不受保护的分支并通过合并请求提交，然后才能合并到与此规则匹配的分支中。

 需要评审人批准

启用后，保护分支的合并请求必须通过代码评审

合并前需要通过 1 个评审人批准

有效评审通过率需 &gt;50%

 需至少一个仓库管理员批准 需要通过状态检查 仅允许线性提交

不允许直接合并到保护分支

 仅允许自动合并

代码合并时仅允许没有冲突，能够自动完成的合并

 禁止创建保护分支 仓库管理员及负责人除外

启用后，仓库管理员及负责人可以创建符合本规则的保护分支

 禁止推送 仓库管理员及负责人除外

启用后，仓库管理员及负责人可以推送到保护分支

## 合并请求

配置仓库全局的合并请求策略，包括合并方式、默认评审人。

合并方式包括：

- 允许直接提交（Merge）：源分支所有的提交都会合并到目标分支，并产生一个新的提交。
- 允许压缩合并（Squash）：源分支中的多个提交会打包成一个，提交合并到目标分支。
- 允许变基合并（Rebase）：来自源分支的多个提交将被重新定位并提交到目标分支。

- ◎ 基础设置
- ▣ 高级设置
- 权限
- 👤 仓库成员
- 功能
- 🔒 分支保护
- 🔗 合并请求
- 🌐 云原生开发
- 🌐 云原生构建
- 用量
- 📊 用量统计

### 合并请求

#### 合并方式

选择仓库允许的合并方式

允许直接提交 (Merge)

源分支所有的提交都会合并到目标分支，并产生一个新的提交

需要评审人批准

默认消息 ▼

允许压缩合并 (Squash)

源分支中的多个提交会打包成一个提交合并到目标分支

合并操作时默认生成的提交信息内容

默认消息 ▼

允许变基合并 (Rebase)

来自源分支的多个提交将被重新定位并提交到目标分支。

#### 默认评审人

创建合并请求后，自动添加仓库管理员为代码评审人

## 云原生开发

了解如何配置当前仓库的云上开发空间。

- ◎ 基础设置
- ▣ 高级设置
- 权限
- 👤 仓库成员
- 功能
- 🔒 分支保护
- 🔗 合并请求
- 🌐 云原生开发
- 🌐 云原生构建
- 用量
- 📊 用量统计

### 云原生开发

当前仓库使用默认的 `cnbcool/default-dev-env:latest` 镜像，作为云原生开发环境

#### 自定义云原生开发环境

如果仓库根目录包含 `.ide/Dockerfile` 文件，则启动开发环境时，会优先使用该文件构建一个镜像，作为云原生开发环境 [了解更多用法](#)

## 云原生构建

配置仓库全局的触发管控。

- 允许自动触发：选中后，仓库将按照 `.cnb.yml` 配置自动触发云原生构建。
- Fork 的仓库默认允许自动触发：选中后，由本仓 Fork 出来的仓库将按照 `.cnb.yml` 配置自动触发云原生构建。

<> 代码 [ISSUE](#) [合并请求 1](#) [云原生构建](#) [制品库](#) [设置](#)

- ◎ 基础设置
- ◎ 高级设置
- 权限
- 仓库成员
- 功能
- 分支保护
- 合并请求
- 云原生开发
- 云原生构建**
- 用量
- 用量统计

### 云原生构建

- 允许自动触发  
选中后，仓库将按照 `.cnb.yml` 配置自动触发云原生构建
- Fork 的仓库默认允许自动触发  
选中后，由本仓 Fork 出来的仓库将按照 `.cnb.yml` 配置自动触发云原生构建

## 用量统计

统计当前仓库的用量情况。支持修改相关资源的使用上限。

基础设置

高级设置

权限

仓库成员

功能

分支保护

合并请求

云原生开发

云原生构建

用量

用量统计

### 用量统计

云原生构建

9.92 核时

云原生开发

0.00 核时

#### 代码仓库

仓库容量

21.67%

已用 21.67 GiB / 上限 100.00 GiB

[修改上限](#)

#### 对象存储

LFS 对象

0.00%

已用 0.00 B / 上限 500.00 GiB

[修改上限](#)

Docker 制品

0.33%

已用 333.57 MiB / 上限 100.00 GiB

[修改上限](#)

[展开详情](#)

Helm 制品

0.00%

已用 0.00 B / 上限 1.00 GiB

[修改上限](#)

图片及附件

0.00%

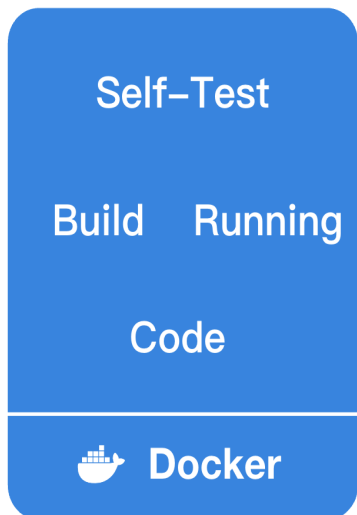
已用 0.00 B / 上限 ∞ B

# 云原生开发

最近更新时间：2025-02-26 17:57:23

## 云上开发空间

提供秒级启动的云端容器环境，为开发者提供编码、编译、运行和自测的能力，能够随时随地进行云上开发。



```

.dockerfile
FROM node:20

# 以及按需安装其他软件
# RUN apt-get update && apt-get install -y git

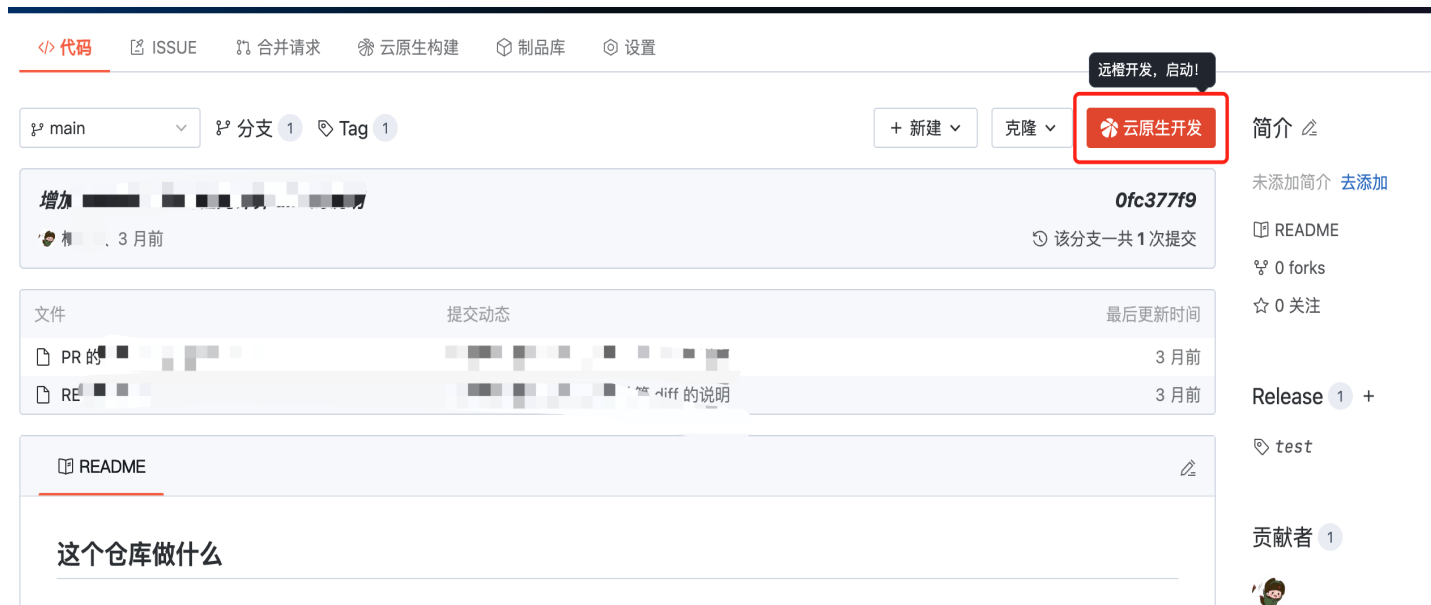
# 安装 code-server 和 vscode 常用插件
RUN curl -fsSL https://code-server.dev/install.sh | sh \
  && code-server --install-extension redhat.vscode-yaml \
  && code-server --install-extension dbaeumer.vscode-eslint \
  && code-server --install-extension eamodio.gitlens \
  && code-server --install-extension tencent-cloud.coding-copilot \
  && echo done

# 安装 ssh 服务，用于支持 VSCode 客户端通过 Remote-SSH 访问开发环境
RUN apt-get update && apt-get install -y wget unzip openssh-server

# 指定字符集支持命令行输入中文（根据需要选择字符集）
ENV LANG C.UTF-8
ENV LANGUAGE C.UTF-8
    
```

## 分支即环境

在仓库首页，单击云原生开发，即可为当前用户 + 该仓库分支，分配关联的云端容器环境。



## 多种打开方式

支持 web 页面直接打开云端容器环境，也支持本地 IDE 通过 Remote-SSH 连接云端容器环境。

 **云原生开发**  
工作空间创建成功 耗时 5.6s

 **使用 WebIDE 打开** →  
使用 WebIDE 体验在线代码编辑

 无需安装  配置漫游  代码自动保存

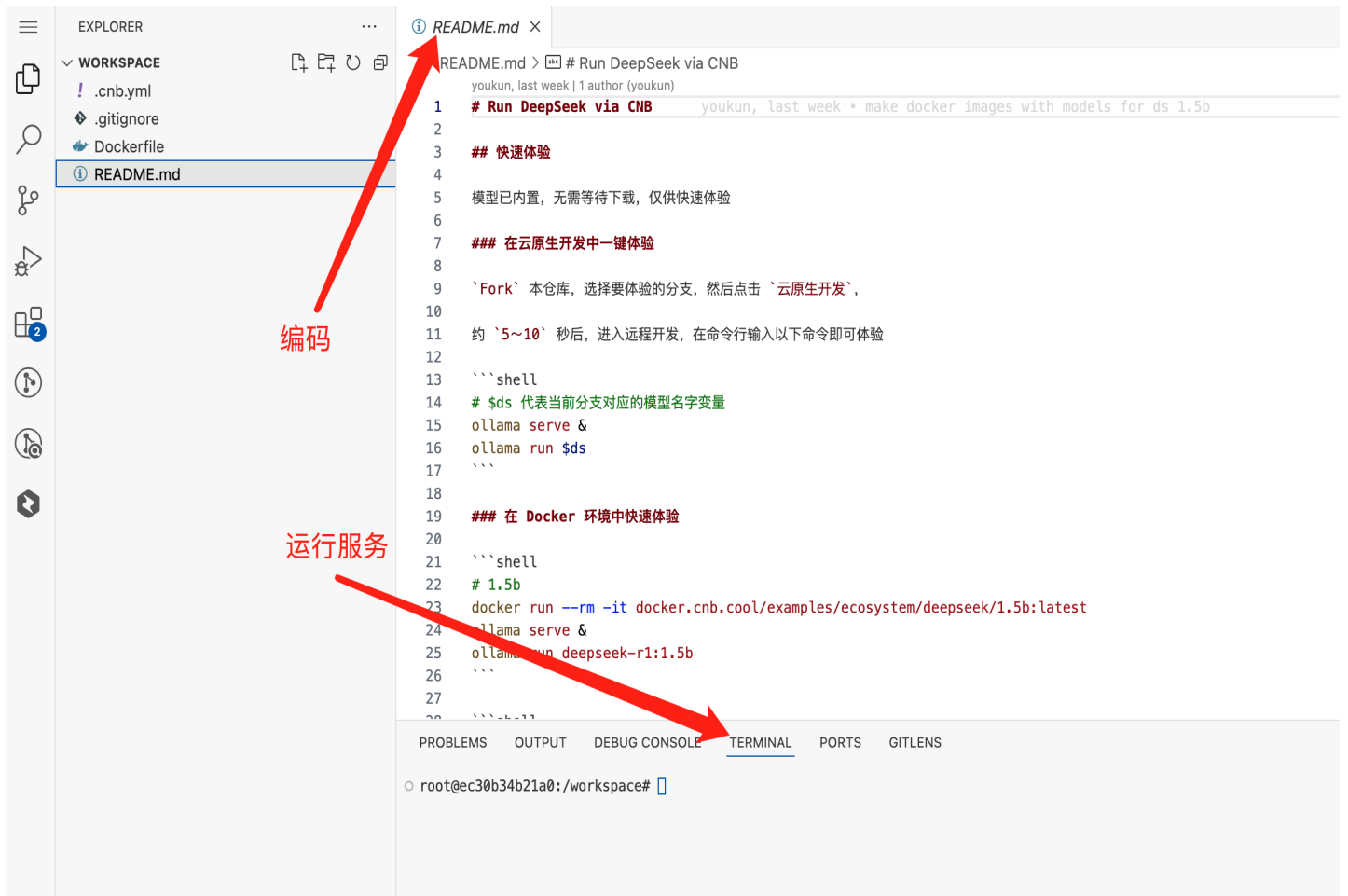
 **使用 VSCode 打开** →  
使用 VSCode 客户端打开(需先安装 Remote-SSH 插件)

 **使用 Cursor 打开** →  
使用 Cursor 客户端打开(需先安装 Remote-SSH 插件)

仓库 `undefined` `dek`  
分支 `undefined`  
日志 `cnb-npo-`

## 云端容器中编码、编译和自测

进入环境后，支持进行编码并运行体验。



## 进阶玩法

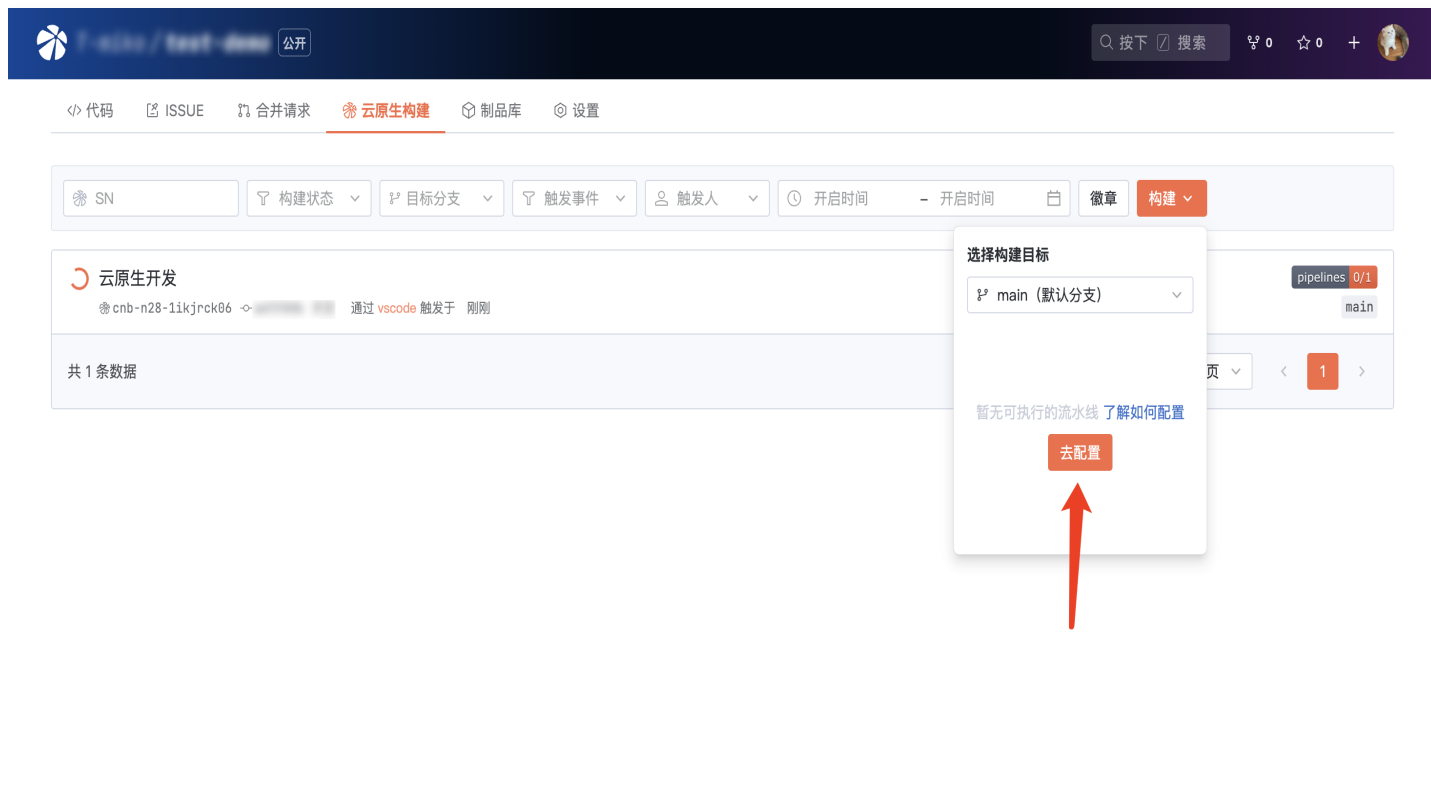
自定义云端容器环境，请参见 [.ide/dockerfile 自定义开发环境](#)。


更多使用技巧，请参见 [使用技巧](#)。

# 云原生构建

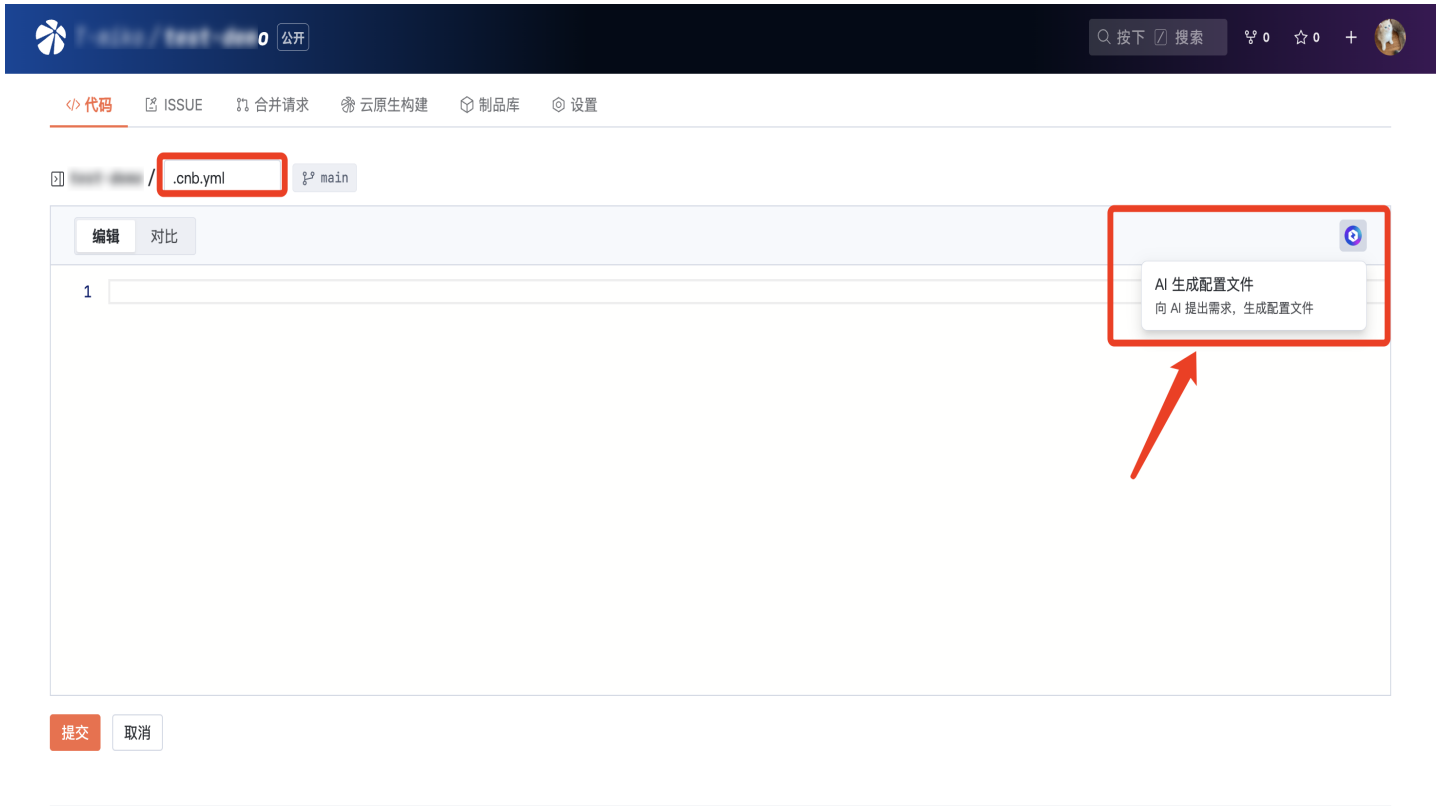
最近更新时间：2025-02-26 17:57:23

CNB 流水线配置采用声明式代码管理方式，通过在代码仓库根目录下的 `.cnb.yml` 文件进行配置。进入目标仓库后，选择云原生构建 > 构建，单击去配置进入文件配置页面。



您可以自行编辑相关内容，也可以单击右上角的  来让 AI 生成配置文件示例。进入 AI 对话页面后，输入您的需求，即可快速生成配置文件。





© 2025 CNB, Inc.

深圳市南山区粤海街道麻岭社区科技中一路腾讯大厦35楼 cnb\_support@tencent.com +86-755-86013388

关于我们 隐私声明 服务协议 API Docs 最佳实践 使用反馈

京ICP备11018762号-111



© 2025 CNB, Inc.

深圳市南山区粤海街道麻岭社区科技中

协议 API Docs 最佳实践 使用反馈

京ICP备11018762号-111

## 典型场景案例

下文以流水线的一个场景作为示例，根据这些场景要素编写 CI 配置文件并查看执行结果。

### 场景

典型的流水线使用场景：

主干分支有 Pull Request 时，触发流水线进行 lint 和 test 检测，未通过则发出通知。

分析这个需求，拆解构建流程的大致思路：

1. 监听的主干分支，比如 main。
2. 触发的仓库事件，即 pull\_request 事件。
3. 构建流程需要执行的 step：
  - lint
  - test
4. 边界条件 – 未通过则触发通知：
  - notify

### 编写流水线配置

1. 第一层属性为分支名：

```
#分支名
main:
```

2. 分支下有 pull\_request 事件时触发构建：

```
#分支名
main:
  # 事件名
  pull_request:
```

3. 事件可以执行多条流水线（并行），流水线有多个任务（串行或并行）。这里我们简化事件下只有一条流水线：

```
#分支名
main:
  # 事件名
  pull_request:
    # 数组类型表示可以有多个流水线
    - name: pr-check
      stages:
```

包括两个串行的任务，分别是 lint 和 test。

```
#分支名
main:
  # 事件名
  pull_request:
    # 数组类型表示可以有多个流水线
    - name: pr-check
      # 流水线下的多个任务
      stages:
        # 要执行的任务
        - name: lint
          script: echo "lint job"
        - name: test
          script: echo "test job"
```

4. 当流水线任务失败时，需要发送通知。流水线除 `stages` 表示期望执行的任务外，还有个 `failStages` 表示 `stages` 任务执行失败时要执行的任务。如下：

```
#分支名
main:
  # 事件名
  pull_request:
    # 数组类型表示可以有多个流水线
    - name: pr-check
      # 流水线下的多个任务
      stages:
        - name: lint
          script: echo "lint job"
        - name: test
          script: echo "test job"
      # stages 失败时执行的任务
      failStages:
        - name: lint
          script: echo "notify to some chat group"
```

## 查看流水线执行结果

在仓库页面单击云原生构建可以看到构建列表。最新一条即是刚刚触发的 `push` 事件流水线，下方有加载标识的记录是云原生开发对应的构建。



单击 **push** 事件流水线名称进入可以看到构建详情。如下图：



## 云原生构建语法

- 云原生构建详细语法，请查阅 [语法](#)。
- 配置文件更多用法，请查阅 [配置文件](#)。

## 进阶实践

如需编写出更符合自身需求的流水线配置，可以参见 [实践示例](#) 选择具体场景的仓库，`fork` 仓库或复制其中配置文件来使用。

- 自定义构建环境，请查阅 [Docker 即环境](#)。
- 自定义构建插件，请查阅 [Docker 即插件](#)。

# 制品管理

最近更新时间：2025-02-26 17:57:23

## 概述

制品库可用于存储、管理和分发软件构建过程中生成的制品。

现云原生构建支持如下类型制品：

- Docker
- Helm
- Maven
- npm
- ohpm

其中，Docker 和 Helm 容器类型的制品，可直接发布至源码仓库，进行存储、管理和分发。

cnb / cool / 公开 按下 搜索 0 0 +

[代码](#) [ISSUE](#) [合并请求](#) [云原生构建](#) [制品库](#) [设置](#)

**欢迎使用制品库**

制品库用于存储、管理和分发软件构建过程中生成的制品。 [如何创建制品](#)

推送您的制品

**Docker**

Docker 可以让开发者打包他们的应用以及依赖包到一个轻量级、可移植的容器中。

[推送制品 >](#)

**Helm**

Helm 使用的包格式称为 Chart，是一个描述 Kubernetes 相关资源的文件集合。

[推送制品 >](#)

© 2025 CNB, Inc. [关于我们](#) [隐私声明](#) [服务协议](#) [API](#) [Docs](#) [最佳实践](#) [使用反馈](#)

深圳市南山区粤海街道麻岭社区科技中一路腾讯大厦35楼 cnb\_support@tencent.com +86-755-86013388 京ICP备11018762号-111

考虑到 Maven、npm 和 ohpm 会涉及二方依赖、三方依赖的统一管理，其需单独创建制品库用于存储、管理和分发。



### 欢迎使用制品库

制品库用于存储、管理和分发软件构建过程中生成的制品。容器类制品，可直接发布至源码仓库



用于管理、分发、加速 JAVA 项目的依赖和插件。

[创建制品库 >](#)



用于管理、分发、加速 JavaScript 和 Node.js 的开源包和模块。

[创建制品库 >](#)



用于管理、分发、加速鸿蒙包。

[创建制品库 >](#)

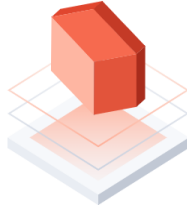
## Docker

### 推送制品

Docker 制品无需创建制品库，可直接发布至源码仓库，按照指定命名规则推送即可：

- 同名制品：制品路径与仓库路径一致，例如：`docker.cnb.cool/{repository-path}`。
- 非同名制品：仓库路径作为制品的命名空间，制品路径就是仓库路径/制品名称，例如：`docker.cnb.cool/{repository-path}/{artifact-name}`。

进入目标仓库后，选择制品库，然后单击推送制品可查看具体的推送方式。界面如下：



### 欢迎使用制品库

制品库用于存储、管理和分发软件构建过程中生成的制品。 [如何创建制品](#)

### 推送您的制品

**Docker**

Docker 可以让开发者打包他们的应用以及依赖包到一个轻量级、可移植的容器中。

推送制品 >

**HelM**

Helm 使用的包格式称为 Chart，是一个描述Kubernetes相关资源的文件集合。

推送制品 >

点击推送制品，可查看操作指引

## 制品列表

在源码仓库 > 制品库，可查看发布至当前源码仓库的制品。可自由切换选择 Docker 或 Helm 制品类型来查看对应制品。

<> 代码
ISSUE 52
合并请求 22
云原生构建
制品库
设置

Docker
Helm

更新时间

请输入制品名搜索

+ 推送制品

<div style="display: flex; justify-content: space-between;"> <span>cnb/mo</span> <span>下载量 333</span> </div> <div style="margin-top: 5px;"> <span style="background-color: #eee; padding: 2px 5px;">linux/amd64</span> <span style="background-color: #eee; padding: 2px 5px;">linux/arm64/v8</span> </div> <div style="margin-top: 5px;"> <span style="color: blue;">Docker</span> <span style="margin-left: 10px;">24</span> <span style="margin-left: 10px;">在 5 天前 推送</span> </div>
<div style="display: flex; justify-content: space-between;"> <span>cnb/mo</span> <span>下载量 14</span> </div> <div style="margin-top: 5px;"> <span>这是CI专用制品</span> </div> <div style="margin-top: 5px;"> <span style="background-color: #eee; padding: 2px 5px;">linux/amd64</span> </div> <div style="margin-top: 5px;"> <span style="color: blue;">Docker</span> <span style="margin-left: 10px;">1</span> <span style="margin-left: 10px;">在 2024-05-16 推送</span> </div>
<div style="display: flex; justify-content: space-between;"> <span>cnb/mo</span> <span>下载量 2</span> </div> <div style="margin-top: 5px;"> <span style="background-color: #eee; padding: 2px 5px;">linux/amd64</span> </div> <div style="margin-top: 5px;"> <span style="color: blue;">Docker</span> <span style="margin-left: 10px;">1</span> <span style="margin-left: 10px;">在 2024-05-16 推送</span> </div>

## 制品详情

`Docker pull` 制品地址，可默认拉取该 Docker 制品的 `latest` 标签。

单击制品名称进入制品详情，可查看使用指引、制品地址、名称、描述、推送时间、下载量等基础信息。

此外，还可查看近期制品和标签列表。



# monorepo/dockerfile-caches



### 制品地址

registry.cn-hongkong.aliyuncs.com/monorepo/dockerfile-caches

### 制品信息

最后发布  
5 天前

推送

下载总量 近七下载  
333 3

### 使用方式

[了解制品使用](#)

#### 登录

可以使用访问令牌作为密码进行鉴权，登录命令：

```
docker login -u cnb [redacted]
```

#### 通过命令行下载

```
docker pull [redacted] monorepo/dockerfile-caches
```

#### 作为云原生构建上下文

```
main:
  push:
    - docker:
      image: [redacted]
      stages:
        - name: hello world
          script: echo "Hello World"
```

#### 作为云原生开发基础镜像

```
$:
  vscode:
    - docker:
      # 指定云原生开发启动时的基础镜像为当前镜像
      image: [redacted]
  services:
    - vscode
    - docker
```

### 历史标签

**5cc6e6a45d49d0dc36844fc9c61e83c2e7fde826**

## 访问指定标签

下载量 0

```
sha256:9d6c9479957ac1fa6f7e9c009d3619b9c1fe5397b420e8ef97803b0a3b18a661
```

linux/amd64 1.75 GiB

9753935241b8d239edded71d193eac8840094fe0

在 1 月前 推送

下载量 6

```
sha256:daa28e9b4ec90d2d677c6e42be09fb80af6dc4be2c878ee1bb5c48b27de560fd
```

linux/amd64 760.38 MiB

aa372e71c513a2df4fed85b34799106eced42ce2

在 1 月前 推送

下载量 23

```
sha256:b38a1be929564ccf2f6a14db31158dd8fe40b0cb71465f37f164d972ad05473
```

linux/amd64 1.8 GiB

012a2e09997a5b5d7e3f8fafa07a84fbc637d07

在 1 月前 推送

下载量 3

```
sha256:3db7de67d0ec5e77ed08ae4e8fa0278d5558179903cae0cee8a112f98112665
```

linux/arm64/v8 1.78 GiB

99990ec18a86dd6df2482e73e080b41d8644b1d0

在 1 月前 推送

下载量 3

```
sha256:ab0700b7cc24f6d28c3811e33b2a6536c547ceb339f0af72184221c55a7a1456
```

linux/amd64 760.38 MiB

全部标签 >

## 查看标签列表

### 标签详情

访问该制品的指定标签，可查看标签的使用指引、基础信息以及使用数据，页面底部还可查看 layers 信息。此时 `docker pull` 制品地址，可拉取该指定标签。

monorepo/dockerfile-caches / 5cc6e6a45d49d0dc36844fc9c61e83c2e7fde826

monorepo/dockerfile-caches:5cc6e6a45d49d0dc36844fc9c61e83c2e7fde826 ⓘ

amd64 sha256:9d6c9479957ac1fa6f7e9c009d3619b9cf1e5397b420e8ef97803b0a3b18a661



#### 制品地址

#### 制品信息

最后发布

5 天前

推送

制品大小

1.75 GiB

下载总量 近七日内下载

0 0

关联资源

e8701592

关联的 git commit id

#### 使用方式

了解制品使用

##### 登录

可以使用访问令牌作为密码进行鉴权，登录命令：

```
docker login -u cnb ( -- )
```

##### 通过命令行下载

```
docker pull dc...po/dockerfile-caches:5cc6e6a45d49d0dc36844fc9c61e83c2e7fde826
```

##### 作为云原生构建上下文

```
main:
  push:
    - docker:
        image: d...caches:5cc6e6a45d49d0dc36844fc9c61e83c2e7fde826
  stages:
    - name: hello world
      script: echo "Hello World"
```

##### 作为云原生开发基础镜像

```
$.
  vscode:
    - docker:
        # 指定云原生开发启动时的基础镜像为当前镜像
        image: d...com/cnb/monorepo/dockerfile-caches:5cc6e6a45d49d0dc36844fc9c61e83c2e7fde826
  services:
    - vscode
    - docker
```

#### LAYERS 30 该 Docker Tag 的 layers

bin/sh...ADD file:7d9c4f68970e9dc2c4689602970ac610fc65c9b09799fca66a3124f5d47d299 in / 47.27 MiB

### 了解更多

更多命令行使用，可以参见 [Docker 制品库](#)。

# Helm

## 推送制品

helm 制品在发布到某一仓库时，支持两种命名规则

1. 同名制品：制品路径与仓库路径一致。例如：`helm.cnb.cool/{repository-path}`。
2. 非同名制品：以仓库路径作为制品的命名空间，制品路径就是仓库路径/制品名称。例如：`helm.cnb.cool/{repository-path}/{artifact-name}`。

### ⚠ 注意：

helm push remote 时，chart name 不出现在 remote-url 中，而是从 chart 中读取。因此，同名制品的 remote-url 为 `helm.cnb.cool/{group-path}`，非同名制品的 remote-url 为 `helm.cnb.cool/{repository-path}`。

## 查看制品

进入目标仓库后，选择制品库，切换选择 Helm 制品类型来查看对应制品。



## 制品详情

`helm pull` 制品地址，可默认拉取该 Helm 制品的 `latest` 标签。

单击制品名称进入制品详情，可查阅使用指引、制品地址、名称、描述、推送时间、下载量等基础信息。此外，还可查看近期制品和标签列表。

HELM
...

制品地址

oci://hel xxxxxx flow

制品信息

最后发布  
45 分钟前  
推送

下载总量 近七日内下载  
0 0

使用方式

[了解制品使用](#)

**登录**

可以使用您的访问令牌作为密码进行鉴权，登录命令：

```
helm registry login -u cnb -p <cnb_token> helm-oci-registry.com
```

**拉取 helm**

```
helm pull oci://helm-oci-registry.com/x-xxxxxx-flow
```

**查看 Helm 信息**

```
helm show all oci://helm-oci-registry.com/x-xxxxxx-flow
```

**预览 manifest**

```
helm template <my-release> oci://helm-oci-registry.com/x-xxxxxx-flow
```

**安装 helm chart**

```
helm install <my-release> oci://helm-oci-registry.com/x-xxxxxx-flow
```

**升级 helm chart**

```
helm upgrade <my-release> oci://helm-oci-registry.com/x-xxxxxx-flow
```

历史标签

16.1.10
🔗
访问指定 Helm 标签

🗑️

🕒 在 45 分钟前 推送
📄 下载量 0


## 标签详情

访问该制品的指定标签，可查看标签的使用指引、基础信息以及使用数据。

此时 `helm pull` 制品地址，可拉取该指定标签。

ai / w / 16.1.10

a ow: 16.1.10


...

制品地址

```
oci://hel... .n/xi... flow --version 16.1.10
```

使用方式

[了解制品使用](#)

**登录**

可以使用您的访问令牌作为密码进行鉴权，登录命令：

```
helm registry login -u cnb -p <cnb_token> hel...
```

**拉取 helm**

```
helm pull oci://hel... .com/xi... flow --version 16.1.10
```

**查看 Helm 信息**

```
helm show all oci://hel... /a... flow --version 16.1.10
```

**预览 manifest**

```
helm template <my-release> oci://he... flow --version 16.1.10
```

**安装 helm chart**

```
helm install <my-release> oci://h... flow --version 16.1.10
```

**升级 helm chart**

```
helm upgrade <my-release> oci://he... flow --version 16.1.10
```

制品信息

最后发布  
48 分钟前  
推送

制品大小  
210.52 KiB

下载总量 近七日内下载  
0 0

- 依赖制品
- redis 18.x.x
  - postgresql 13.x.x
  - common 2.x.x

当前 helm chart 所依赖的其他 chart

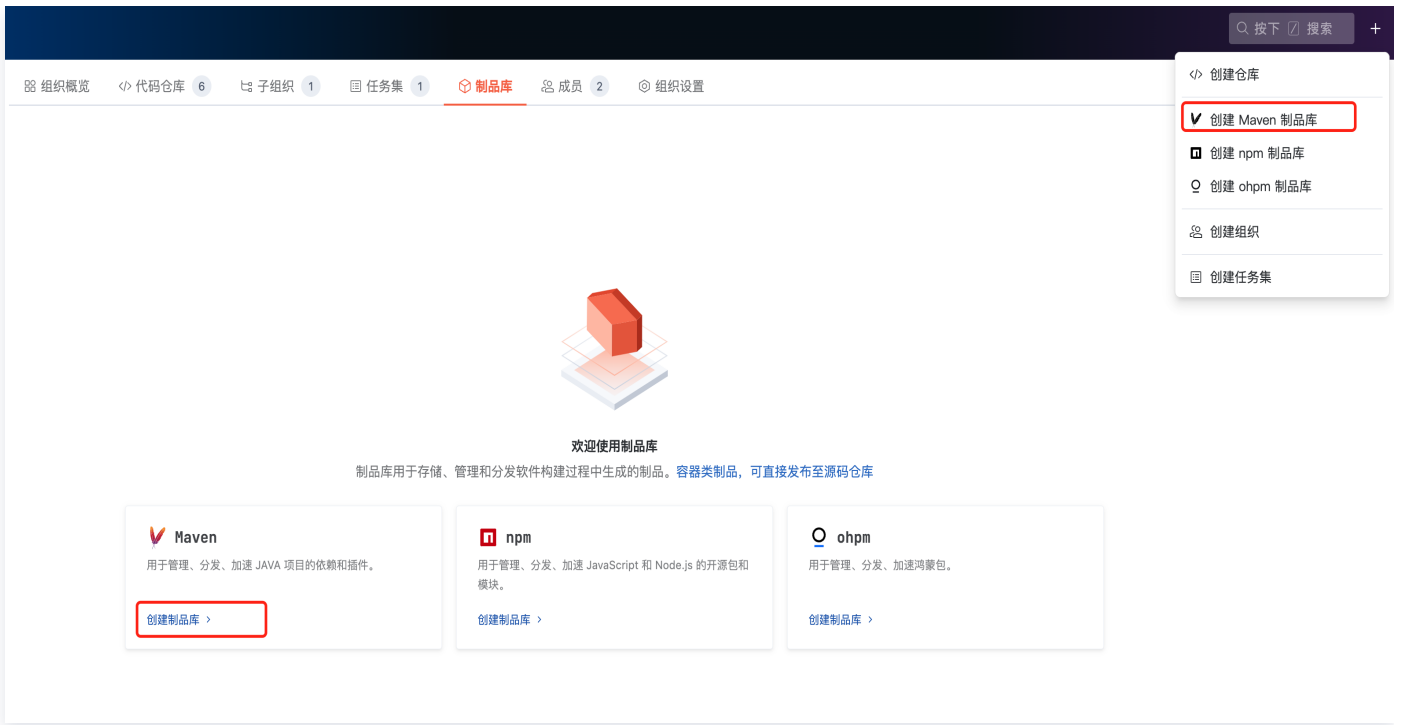
## 了解更多

更多命令行使用，可以参见 [Helm 制品库](#)。

## Maven

### 创建制品库

1. 可在 [cnb.cool](#) 右上角单击 + > **创建 Maven 制品库**快速发起创建，或进入某个组织，在制品库中单击**创建制品库**。



2. 创建 maven 制品库时，除名称、描述、可见性等基础信息外，还支持配置版本类型，可用于管控存储制品的版本策略。

**注意：**

版本策略创建后，不允许修改。

## 创建 Maven 制品库

制品库归属 \*

制品库名称 \*

 0/255

URL预览

制品库主页	https://cnb.cool/x/n/
Maven repository	https://maven.cnb.cool/x/n/ /-/packages/

制品库描述

 0/100

版本类型 \*

- 不限制版本类型（默认禁止版本覆盖）
- 仅 Snapshot 版本（存放 snapshot，默认允许版本覆盖）
- 仅 Release 版本（存放 release，默认禁止版本覆盖）

公开性 \*

- 公开（任何人都可以查看，适用于开源协作）
- 私有（仅限制制品库成员访问，适用于团队内部协作）

创建

取消

## 查看制品库

访问目标组织 > 制品库，可查看该组织下的制品库列表，切换制品类型可查看对应类型的制品库。

组织概览 > 代码仓库 1 > 子组织 1 > 任务集 > **制品库** > 成员 1

全部 Maven npm ohpm 最近创建  [+ 创建制品库](#)

- fengtest/maven/maven-open**  
Maven 制品库 1 6天前更新
- fengtest/feng-maven-open**  
Maven 制品库 0 1周前更新

## 制品包列表

单击某一制品仓库，首页即为仓库内的制品包列表。Maven 包列表中的制品名称由 Maven 包的基础信息组成，规则为：`group-id:artifact-id`。

单击界面右侧的**使用指引**，可查阅配置凭据、拉取制品、推送制品的参考命令。

**Maven 包** 1

更新时间  [使用指引](#)

- group1.0:artifact1.0** 下载量 0  
Maven 1 在 6 天前 推送 1.1.0-SNAPSHOT

**简介**  
未添加简介  
类型  
Maven 制品库  
版本类型  
不限制版本类型

## 制品包详情

在制品包详情中，可查阅使用指引、名称、描述、推送时间、下载量等基础信息。切换使用方式可查看 maven、gradle 不同包管理器的使用指引。

**注意：**  
在包详情的使用指引中，未指明 version 信息，请注意替换。



maven-open / group1.0:artifact1.0

**group1.0:artifact1.0**



使用方式 Maven

配置凭据

制品信息

1.将 dependency 配置添加到 pom.xml 中，前置依赖完成 配置凭据

```

<!-- 请自行替换 groupId 等信息，或进入制品包详情查看 -->
<dependencies>
  <dependency>
    <groupId>group1.0</groupId>
    <artifactId>artifact1.0</artifactId>
    <version>[VERSION]</version>
  </dependency>
</dependencies>
    
```

2.使用命令拉取

```

mvn install
# 如果选择指定位置的 settings.xml 和 pom.xml 作为配置文件，请执行
mvn install -s ./settings.xml -f ./pom.xml
    
```

最后推送

6 天前

推送

下载总量 近七日内下载

0 0

历史版本

1.1.0-SNAPSHOT



创建于 6 天前 推送

下载量 0

## 制品包版本详情

在制品包详情中，单击界面下方的历史版本，进入该版本详情。在文件列表中，可查阅该制品包版本中的文件内容以及文件大小。

📦 / maven-open / group1.0:artifact1.0 / 1.1.0-SNAPSHOT

**group1.0:artifact1.0:1.1.0-SNAPSHOT**



使用方式 Maven

[配置凭据](#) **制品信息**

1.将 dependency 配置添加到 pom.xml 中, 前置依赖完成 [配置凭据](#)

```
<!-- 请自行替换 groupId 等信息, 或进入制品包详情查看 -->
<dependencies>
  <dependency>
    <groupId>group1.0</groupId>
    <artifactId>artifact1.0</artifactId>
    <version>1.1.0-SNAPSHOT</version>
  </dependency>
</dependencies>
```

2.使用命令拉取

```
mvn install
# 如果选择指定位置的 settings.xml 和 pom.xml 作为配置文件, 请执行
mvn install -s ./settings.xml -f ./pom.xml
```

最后推送

6 天前

■ 推送

制品大小

1.87 KiB

下载总量 近七日下载

0 0

文件列表

文件	文件大小
artifact1.0-1.1.0-20250218.071250-1.jar	1.39 KiB
artifact1.0-1.1.0-20250218.071250-1.jar.md5	32 B
artifact1.0-1.1.0-20250218.071250-1.jar.sha1	40 B

## 制品库设置

### 基础设置

进入目标制品库 > **制品库设置** > **基础设置**, 可以修改制品库名称、修改可见性以及删除仓库。

基础设置

- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

基础设置

制品库名

maven 修改

制品库名与主页 URL 地址相关联

制品库主页	https://maven.aliyun.com/maven/maven
Maven repository	https://maven.aliyun.com/maven/maven/-/packages/

制品类型

Maven

版本类型

不限制版本类型

危险操作

修改制品库可见性

当前为公开制品库，公开制品库对整个社区可见。可变更为私有制品库，仅对制品库成员可见。

修改制品库可见性

删除制品库

制品库删除后将无法恢复，请谨慎操作

删除制品库

## 成员管理

默认继承上级组织的所有成员权限关系。此外，还可单击邀请成员直接邀请成员加入当前制品库成为“制品库成员”或“外部协作者”。其中，外部协作者适用于一些临时参与协作的用户。

- 基础设置
- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

制品库成员

制品库成员 (1) 继承成员 (1) 外部协作者  + 邀请成员

成员	角色	加入时间	操作
	负责人	2025-02-08 16:56	...

共 1 条数据 10 条/页 1

## 代理设置

为提升公共制品的拉取效率，云原生构建制品库配置了内置的制品加速源，在使用制品库时，可基于内置的加速源地址直接加速。

基础设置

权限

成员管理

功能

代理设置

策略管理

用量

用量统计

### 代理设置

拉制品时，优先获取本仓库的内容，而后按照优先级逐级检索加速源内容，优先级 1 为最高。

#### 系统加速源

优先级	加速源地址
1	https://repo.maven.apache.org/maven2/
2	https://mirrors.cloud.tencent.com/nexus/repository/maven-public/

## 策略管理

可配置制品包的覆盖策略以及清理策略。

配置清理策略时，可配置最多保留的版本数量，当超出时，将根据配置的策略自动清理推送时间更早的/使用时间更早的，以节省制品库空间。

基础设置

权限

成员管理

功能

代理设置

策略管理

用量

用量统计

### 策略设置

#### 版本覆盖策略

允许覆盖已有版本

启用后，推送已有版本时，将会覆盖更新原有内容

#### 版本清理策略

自动版本清理

启用后，将根据条件自动清理版本以节省制品库空间

最多保留  个版本，超出则自动清理

优先清理  制品

推送时间更早的

使用时间更早的

## 用量统计

汇总当前制品仓库中，所有包占用存储空间的总和，下方可查看每个包的占用情况，并支持下载详细清单。

基础设置

权限

成员管理

功能

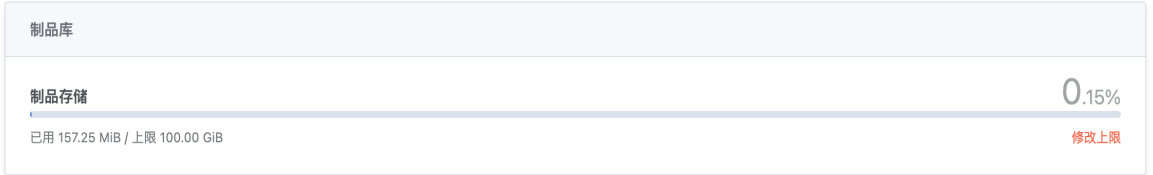
代理设置

策略管理

用量

用量统计

用量统计



用量详情

包名称	占用容量	占比
net.mingsoft:ms-mcms	157.25 MiB	100.00%

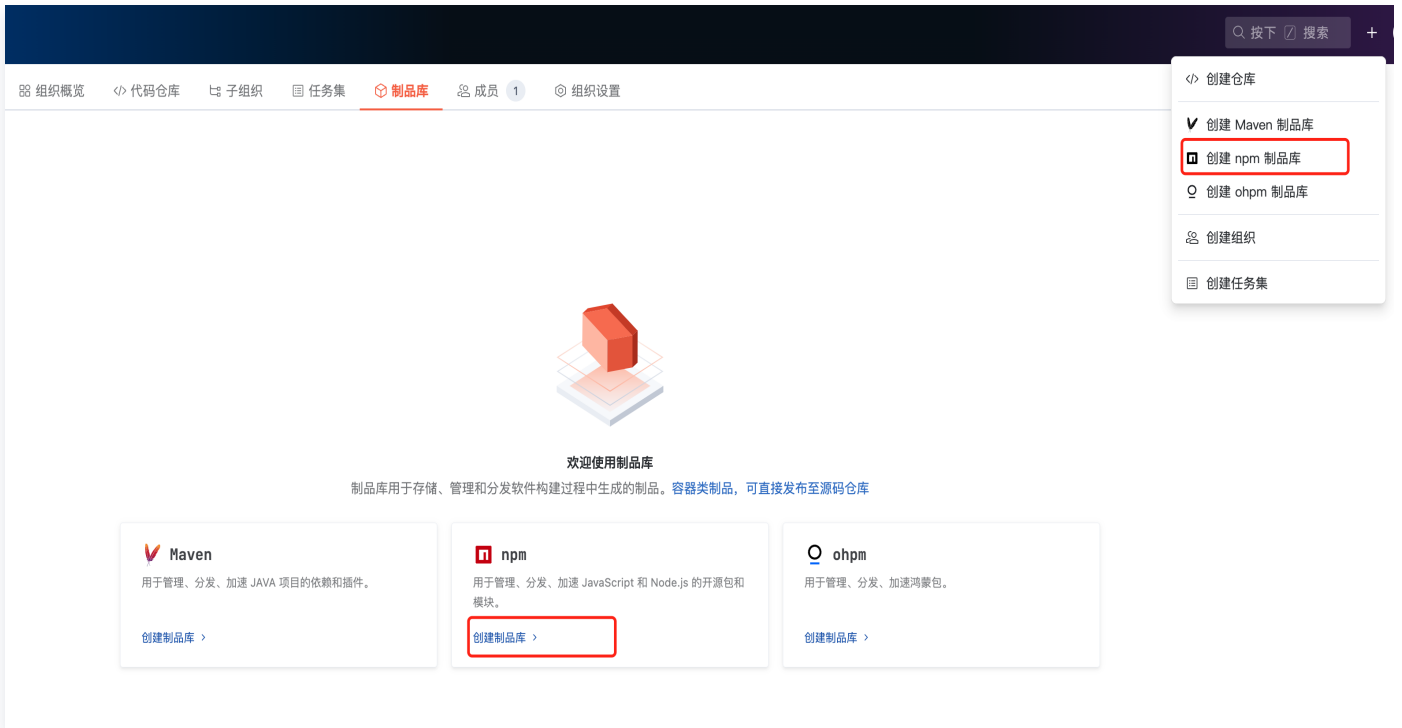
共 1 条数据

10 条/页 < 1 >

# npm

## 创建制品库

- 可在 [cnb.cool](https://cnb.cool) 右上角单击 + > 创建 npm 制品库快速发起创建，或进入某个组织，在制品库中单击创建制品库。



- 创建 npm 制品库需配置名称、描述和可见性基础信息。

## 创建 npm 制品库

制品库归属 \*

 registry-test ▼

制品库名称 \*

支持英文、数字、点、短横线、下划线，如：cloud-native-build 0/255

URL预览

制品库主页 <https://cnb-dev.woa.com/registry-test/>

npm registry <https://npm.cnb-dev.woa.com/registry-test/-/packages/>

制品库描述

添加制品库描述 0/100

公开性 \*

- 公开（任何人都可以查看，适用于开源协作）
- 私有（仅限制制品库成员访问，适用于团队内部协作）

创建 取消

## 查看制品库

访问目标组织 > 制品库，可查看该组织下的制品库列表，切换制品类型可查看对应类型的制品库。

☰ 组织概览
<> 代码仓库 1
🏠 子组织 1
📁 任务集
📦 制品库
👤 成员 1

全部
Maven
npm
ohpm

◇ 最近创建
▼

+ 创建制品库
▼

f

*fengtest/feng-npm-open*

---

npm 制品库 📦 2 🕒 1周前更新

## 制品包列表

单击某一制品仓库，首页即为仓库内的制品包列表。npm 包列表中的制品名称由 `package.json` 中的 `name` 指定。

单击界面右侧的**使用指引**，可查阅配置凭据、拉取制品、推送制品的参考命令。

npm 包 2

更新时间

请输入制品名搜索

使用指引

简介

未添加简介

类型

npm 制品库

<code>types-node</code>	下载量 3
npm 2 在 6 天前 推送 22.3.1	
<code>A/types-node</code>	下载量 0
npm 1 在 6 天前 推送 22.3.0	

## 制品包详情

在制品包详情中，可查阅使用指引、名称、描述、推送时间、下载量等基础信息。切换使用方式可查看 npm、yarn 不同使用方式的命令指引。

### 注意：

npm 不指定版本时，将会拉取 `latest` 标签对应的版本。

npm 包 2

/ feng-npm-open / types-node

types-node



使用方式 npm

配置凭据

制品信息

1.使用命令拉取

```
npm install types-node --registry=https://[...]/fengtest/feng-npm-open/~/packages/
```

最后推送

6 天前

推送

下载总量

近七下载

3

0

历史版本

22.3.1

在 6 天前 推送



下载量 2

22.3.0

在 6 天前 推送



下载量 1

## 制品包版本详情

在制品包详情中，单击界面下方的历史版本，进入该版本详情。

- 在文件列表中，可查阅该制品包版本中的文件内容以及文件大小。
- 在依赖列表中，可查阅该制品包版本所依赖的其他包及版本，取自 `package.json` 中的 `dependencies`。



📦 / feng-npm-open / types-node / 22.3.1

## types-node:22.3.1



使用方式 npm

[配置凭据](#)

### 制品信息

最后推送

6 天前

■ 推送

制品大小

1.58 KiB

下载总量 近七日内下载

2 0

1.使用命令拉取

```
npm install types-node@22.3.1 --registry=https://npm.../fengtest/feng-npm-open/-/packages/
```

### 文件列表

文件	文件大小
types-node-22.3.1.tgz	1.58 KiB

### 依赖列表

包名称	版本
alannfeng0122	^1.0.7
test123	^1.0.0
webpack	^5.90.3

## 制品库设置

### 基础设置

进入目标制品库 > 制品库设置 > 基础设置，可以修改制品库名称、修改可见性以及删除仓库。

- ② 基础设置
- 权限
- 👤 成员管理
- 功能
- 🔗 代理设置
- ⚙️ 策略管理
- 用量
- 📊 用量统计

### 基础设置

#### 制品库名

npm-test 修改

制品库名与主页 URL 地址相关联

制品库主页	https://cnpm.dev/packages/npm-test
npm registry	https://npm.pkg.github.com/npm-test/-/packages/

#### 制品类型

npm

### 危险操作

#### 修改制品库可见性

当前为公开制品库，公开制品库对整个社区可见。可变更为私有制品库，仅对制品库成员可见。

修改制品库可见性

#### 删除制品库

制品库删除后将无法恢复，请谨慎操作

删除制品库

## 成员管理

默认继承上级组织的所有成员权限关系。此外，还可单击**邀请成员**直接邀请成员加入当前制品库成为“制品库成员”或“外部协作者”。其中，外部协作者适用于一些临时参与协作的用户。

- ② 基础设置
- 权限
- 👤 成员管理
- 功能
- 🔗 代理设置
- ⚙️ 策略管理
- 用量
- 📊 用量统计

### 制品库成员

👤 制品库成员 (1)
👤 继承成员 (1)
👤 外部协作者+ 邀请成员

成员	角色	加入时间	操作
👤 成员	负责人	2025-02-24 21:30	...

共 1 条数据

10 条/页
<
1
>

## 代理设置

为提升公共制品的拉取效率，云原生构建制品库配置了内置的制品加速源，在使用制品库时，可基于内置的加速源地址直接加速。

npm 包 制品库设置

- 基础设置
- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

### 代理设置

拉取制品时，优先获取本仓库的内容，而后按照优先级逐级检索加速源内容，优先级 1 为最高。

#### 系统加速源

优先级	加速源地址
1	http://mirrors.cloud.tencent.com/npm

## 策略管理

可配置制品包的覆盖策略以及清理策略。

配置清理策略时，可配置最多保留的版本数量，当超出时，将根据配置的策略自动清理推送时间更早的/使用时间更早的，以节省制品库空间。

npm 包 制品库设置

- 基础设置
- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

### 策略设置

#### 版本覆盖策略

允许覆盖已有版本

启用后，推送已有版本时，将会覆盖更新原有内容

#### 版本清理策略

自动版本清理

启用后，将根据条件自动清理版本以节省制品库空间

最多保留  个版本，超出则自动清理

优先清理  制品

## 用量统计

汇总当前制品仓库中，所有包占用存储空间的总和，下方可查看每个包的占用情况，并支持下载详细清单。

基础设置

权限

成员管理

功能

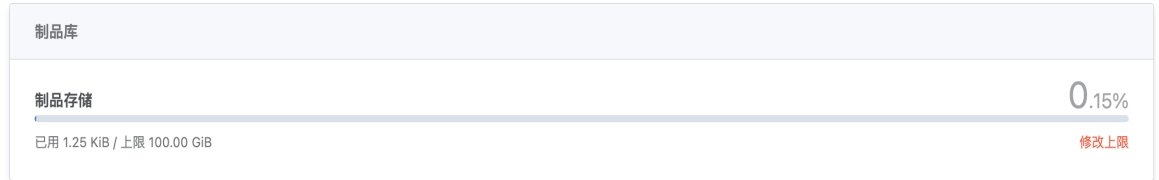
代理设置

策略管理

用量

用量统计

### 用量统计



### 用量详情

包名称	占用容量	占比
types-node	1.25 KiB	100.00%

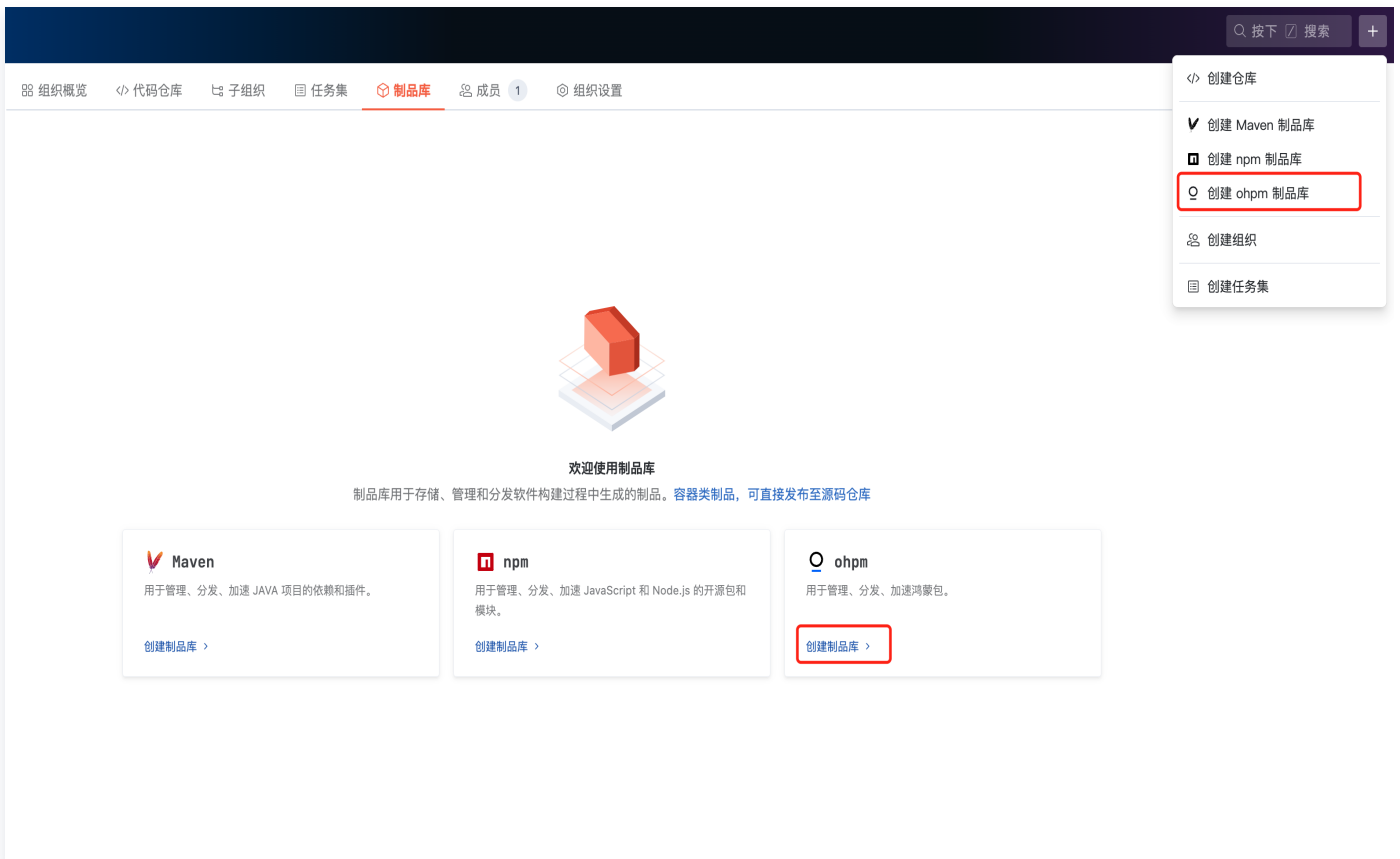
共 1 条数据

10 条/页 < 1 >

# ohpm

## 创建制品库

1. 可在 [cnb.cool](https://cnb.cool) 右上角单击 + > 创建 ohpm 制品库快速发起创建，或进入某个组织，在制品库中单击创建制品库。



2. 创建 ohpm 制品库需配置名称、描述和可见性基础信息。

## 创建 ohpm 制品库

制品库归属 \*

registry-test

制品库名称 \*

支持英文、数字、点、短横线、下划线，如：cloud-native-build 0/255

URL预览

制品库主页 https://cnb-...com/registry-test/  
ohpm registry https://ohpm-...com/registry-test/-/packages/

制品库描述

添加制品库描述 0/100

公开性 \*

- 公开 (任何人都可以查看, 适用于开源协作)
- 私有 (仅限制制品库成员访问, 适用于团队内部协作)

创建 取消

## 查看制品库

访问目标组织 > 制品库, 可查看该组织下的制品库列表, 切换制品类型可查看对应类型的制品库。

组织概览 < 代码仓库 1 子组织 1 任务集 制品库 成员 1

全部 | Maven | npm | ohpm 最近创建

请输入制品库名搜索 + 创建制品库

fengtest/ohpm-open

ohpm 制品库 1 6天前更新

## 制品包列表

单击某一制品仓库，首页即为仓库内的制品包列表。ohpm 包列表中的制品名称由 `oh-package.json5` 中的 `name` 指定。

单击界面右侧的**使用指引**，可查阅配置凭据、拉取制品、推送制品的参考命令。

ohpm 包 1

更新时间

请输入制品名搜索

使用指引

简介

未添加简介

类型

ohpm 制品库

alann-demo 下载量 1

ohpm 在 6 天前 推送 1.0.1

## 制品包详情

在制品包详情中，可查阅使用指引、名称、描述、推送时间、下载量等基础信息。

**注意：**  
ohpm 不指定版本时，将会拉取 `latest` 标签对应的版本。

ohpm 包 1

/ ohpm-open / alann-demo

alann-demo

使用方式

配置凭据

制品信息

最后推送

6 天前

推送

下载总量 近七下载

1 1

历史版本

1.0.1

在 6 天前 推送

下载量 1

```
ohpm install alann-demo --registry=https://..._est/ohpm-open/-/packages/
```

## 制品包版本详情

在制品包详情中，单击界面下方的历史版本，进入该版本详情。

- 在文件列表中，可查阅该制品包版本中的文件内容以及大小。

- 在依赖列表中，可查阅该制品包版本所依赖的其他包及版本，取自 `oh-package.json5` 中的 `dependencies`。

ohpm-open / alann-demo / 1.0.1

**alann-demo:1.0.1**

使用方式

配置凭据

制品信息

1.使用命令拉取

```
ohpm install alann-demo@1.0.1 --registry=https://ohpm.open.tencent.com/fengtest/ohpm-open/--packages/
```

最后推送

6 天前

唯一推送

制品大小

810 KiB

文件列表

下载总量

近七日下载

1

1

文件	文件大小
alann-demo-1.0.1.har	810 KiB

依赖列表

包名称	版本
@pura/harmony-utils	^1.2.8

## 制品库设置

### 基础设置

进入目标制品库 > 制品库设置 > 基础设置，可以修改制品库名称、修改可见性以及删除仓库。

基础设置

- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

基础设置

制品库名

ohpm-test 修改

制品库名与主页 URL 地址相关联

制品库主页 <https://ohpm-test.n/ohpm-test>  
 ohpm registry <https://ohpm-test.n/ohpm-test/-/packages/>

制品类型

ohpm

危险操作

修改制品库可见性

当前为公开制品库，公开制品库对整个社区可见。可变更为私有制品库，仅对制品库成员可见。

修改制品库可见性

删除制品库

制品库删除后将无法恢复，请谨慎操作

删除制品库

## 成员管理

默认继承上级组织的所有成员权限关系。此外，还可单击邀请成员直接邀请成员加入当前制品库成为“制品库成员”或“外部协作者”。其中，外部协作者适用于一些临时参与协作的用户。

- 基础设置
- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

制品库成员

制品库成员 (1) 继承成员 (1) 外部协作者  + 邀请成员

成员	角色	加入时间	操作
	负责人	2025-02-24 21:58	...

共 1 条数据 10 条/页 < 1 >

## 代理设置

为提升公共制品的拉取效率，云原生构建制品库配置了内置的制品加速源，在使用制品库时，可基于内置的加速源地址直接加速。



- 基础设置
- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

### 代理设置

拉取制品时，优先获取本仓库的内容，而后按照优先级逐级检索加速源内容，优先级 1 为最高。

#### 系统加速源

优先级	加速源地址
1	https://ohpm.openharmony.cn/ohpm

## 策略管理

可配置制品包的覆盖策略以及清理策略。

配置清理策略时，可配置最多保留的版本数量，当超出时，将根据配置的策略自动清理推送时间更早的/使用时间更早的，以节省制品库空间。

- 基础设置
- 权限
- 成员管理
- 功能
- 代理设置
- 策略管理
- 用量
- 用量统计

### 策略设置

#### 版本覆盖策略

允许覆盖已有版本  
启用后，推送已有版本时，将会覆盖更新原有内容

#### 版本清理策略

自动版本清理  
启用后，将根据条件自动清理版本以节省制品库空间

最多保留  个版本，超出则自动清理

优先清理  制品

## 用量统计

汇总当前制品仓库中，所有包占用存储空间的总和，下方可查看每个包的占用情况，并支持下载详细清单。

基础设置

权限

成员管理

功能

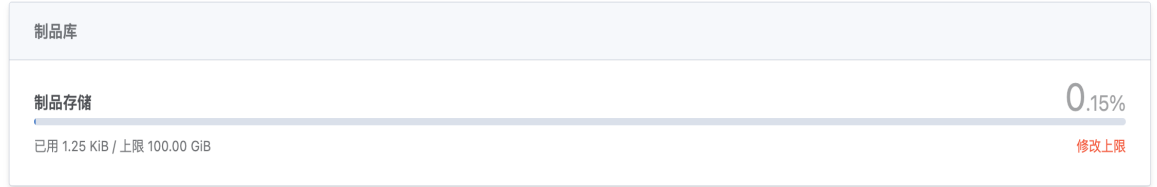
代理设置

策略管理

用量

用量统计

### 用量统计



### 用量详情

下载

包名称	占用容量	占比
aLann-demo	1.25 KiB	100.00%

共 1 条数据

10 条/页 < 1 >

# 任务集

最近更新时间：2025-02-26 17:57:23

任务集可以帮助团队跨组织/仓库集中化视图管理 Issue/PR。

## 创建任务集

进入 **CNB**，单击右上角的“+”，选择**创建任务集**。

- 任务集名称：任务集的标识，也会组成任务集的访问路径。
- 数据范围：管理任务集中 Issue/PR 的数据来源。

按下 搜索 +

### 创建任务集

任务集归属 \*

任务集名称 \*

URL 预览

任务集描述

数据范围 \* <sup>①</sup>

公开性

公开任务集 (任何人都可以访问)

私有任务集 (仅任务集成员和组织成员可访问)

- <> 创建仓库
- ✓ 创建 Maven 制品库
- ▣ 创建 npm 制品库
- 创建 ohpm 制品库
- 👤 创建组织
- 📁 创建任务集**

## 任务集视图

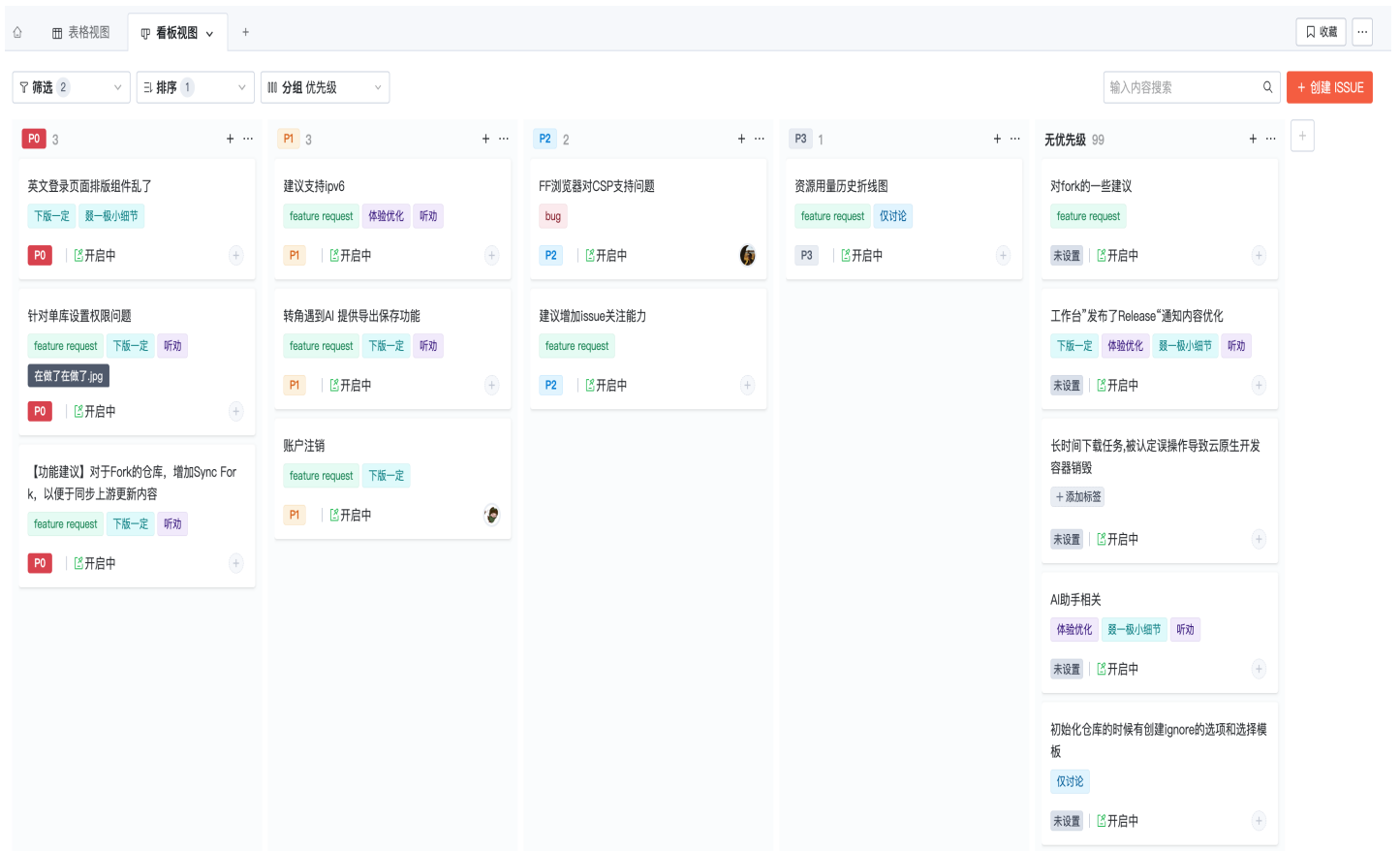
在任务集中，您可以：

- 配置视图，任务集将自动读取 Issue/合并请求的最新状态。
- 自由配置/切换表格视图、看板视图，以不同视角管理 Issue/PR。
- 通过筛选、排序、搜索来查看指定条件下的 Issue/PR。
- 快速创建指定归属仓库的 Issue。
- 进入看板视图，通过移动拖放，快速修改事项的优先级。


表格视图效果如下：

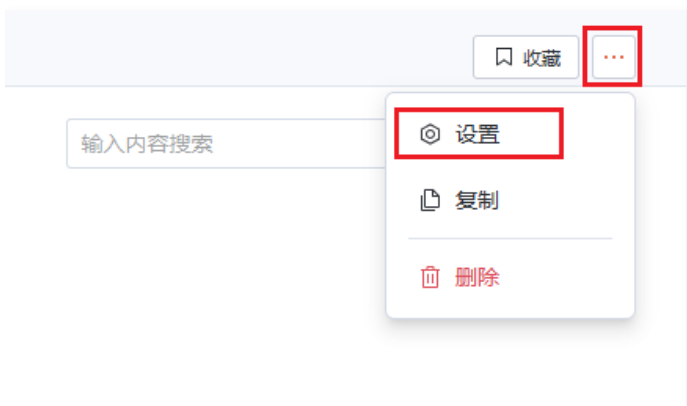
标题	状态	处理人	优先级	标签
对fork的一些建议	开启中			feature request
工作台"发布了Release"通知内容优化	开启中			体验优化 听劝 下版一定 跟一版小细节
长时间下载任务,被认定误操作导致云原生开发容器销毁	开启中			
AI助手相关	开启中			体验优化 听劝 跟一版小细节
初始化仓库的时候有创建ignore的选项和选择模板	开启中			仅讨论
受保护分支合入默认分支后, 页面提示的删除分支按钮是否可以取消	开启中			
需要注销账号功能	开启中			听劝 下版一定 feature request
Linux版, 如已开发, 请提供。	开启中			仅讨论
站内链接加强匹配, 解决无空格的干扰	开启中			听劝 下版一定 跟一版小细节
增加 ssh 支持	开启中			仅讨论 委编拒绝
Private Repo 不能 Fork & PR?	开启中			仅讨论 委编拒绝
docker 制品详情页显示 manifest 文件内容	开启中			跟一版小细节 feature request
fork仓库同步功能的2点体验优化	开启中			
自定义 issue tracker	开启中			仅讨论
仓库墙 支持 排序	开启中			听劝 下版一定 跟一版小细节
仓库未初始化代码时, 无法修改项目简介	开启中			
已合并的 pr, 仍然出现在代办列表里, 无法删除	开启中			bug

看板视图效果如下：



## 任务集设置

在任务集中单击右侧的 ，然后选择**设置**，即可进入任务集设置页。可以进行基础设置、高级设置和管理任务集成员。



## 基础设置

可以修改任务集名称和编辑任务集的数据范围等任务集的基础信息。

**注意：**

修改任务集名称将会变更任务集的访问 URL，需谨慎操作。

[← 任务集设置](#)

基础设置

高级设置

权限

任务集成员

## 基础设置

## 任务集名称 \*

test-missions

13/255

任务集名称与任务集详情 URL 地址相关联

任务集详情 https://[redacted].cn/test-missions

## 任务集描述

请额外添加一个任务集描述

0/100

## 数据范围 \* ①

cnb/feedback

x

cnb/plugins/market

x

cnb/cool/git-clone-yyds

x

cnb/cool/node-plugin-env

x

cnb/sdk/go-sdk-generator

x

[+ 选择数据范围](#)已选择 5 个，最多 10 个 [清空](#)[修改](#)[取消](#)

## 高级设置

可以复制任务集、修改任务集的可见性和删除任务集。

**⚠ 注意：**

修改任务集可见性会影响任务集的访问范围，删除任务集后数据无法恢复，需谨慎操作。

## 任务集设置

基础设置

高级设置

权限

任务集成员

### 高级设置

#### 复制任务集

复制当前任务集内容，创建一个新的任务集

复制任务集

### 危险操作

#### 修改任务集可见性

当前为公开任务集，对整个社区可见。可变更为私有任务集，仅对任务集和组织成员可见。

修改任务集可见性

#### 删除任务集

任务集删除后将无法恢复，请谨慎操作

删除任务集

## 任务集成员

默认继承上级组织的所有成员权限关系。此外，还可单击**邀请成员**直接邀请成员加入当前任务集成为“任务集成员”或“外部协作者”。其中，外部协作者适用于一些临时参与协作的用户。

← 任务集设置

基础设置

高级设置

权限

任务集成员

任务集成员

任务集成员 (1) | 继承成员 (2) | 外部协作者

搜索成员

+ 邀请成员

成员	角色	加入时间	操作
	负责人	2025-02-24 18:53	...

共 1 条数据

20 条/页 < 1 >