



# EulerDevOps 原生开发服务

sig-CICD

Ethan-Zhang 张以正

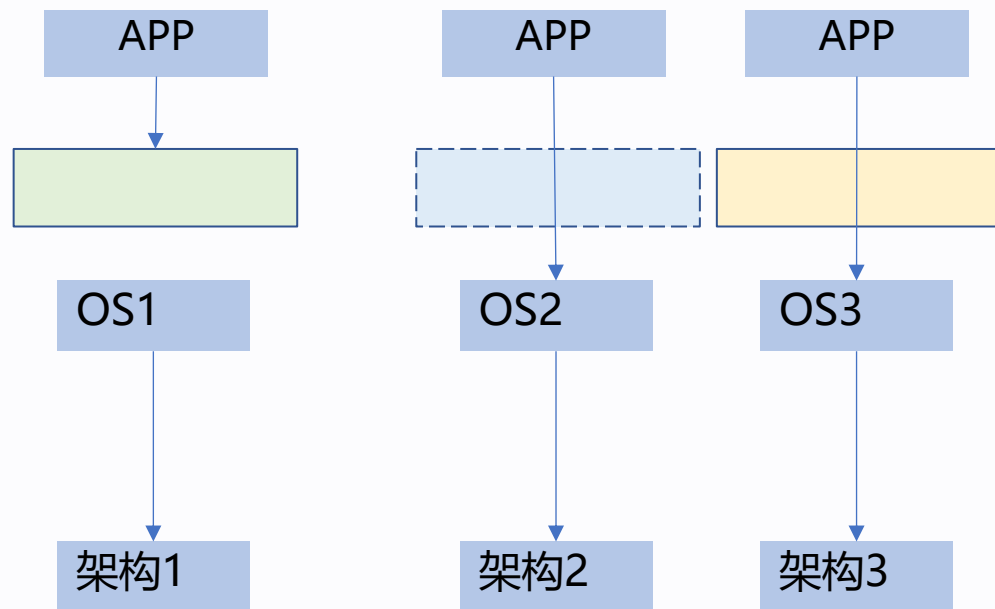
# 目录

---

1. 原生开发是什么?
2. 原生开发服务架构 ( $1+1+N+M$ )
3. 1套灵活强大的描述语言
4. N种预定义模板
5. M种原子任务
6. Intel-lkvs流水线开发实例
7. 快速定义自用流水线
8. 一键复现与结对调试

# 原生开发是什么？

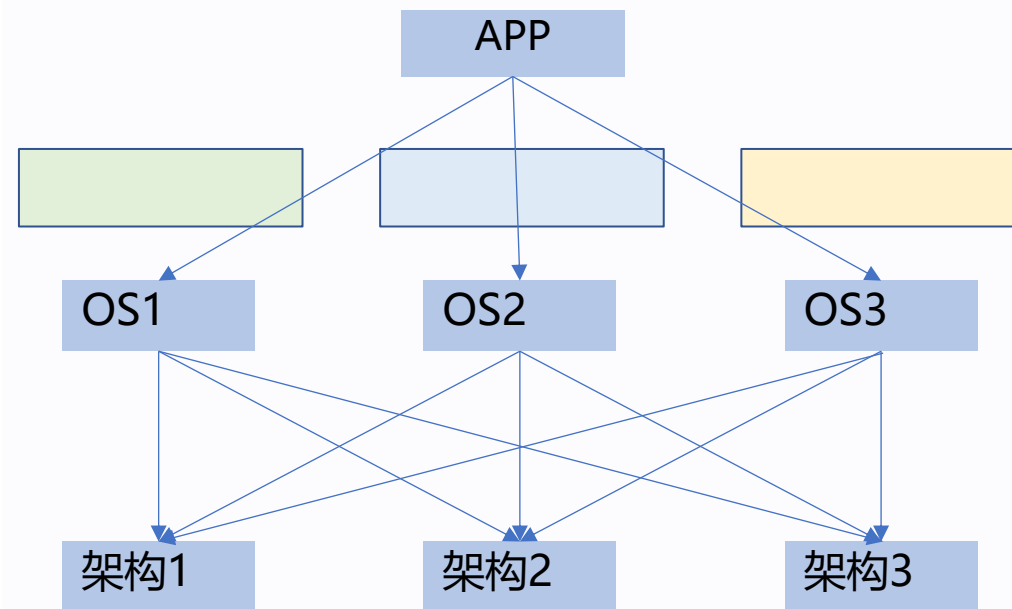
## 以应用软件为中心开发



单算力开发

多算力移植&适配

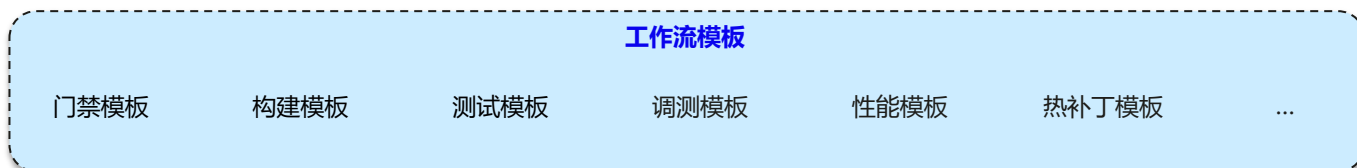
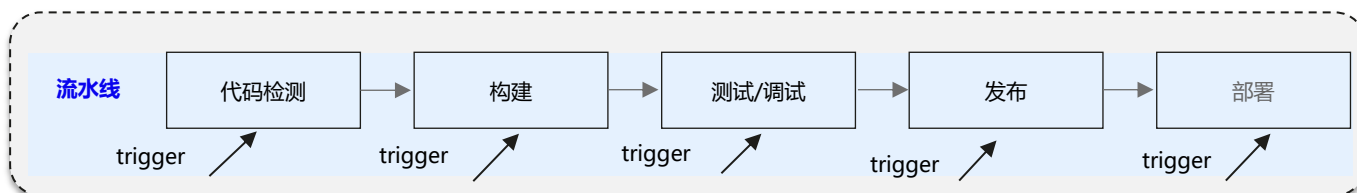
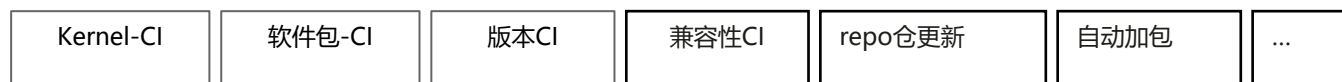
## 以多样性算力为中心开发



使能应用软件在多样性算力并行开发，原生支持

- openEuler 操作系统及工具链融入到开发者的日常编码、编译、构建、测试、调测、调优、发布等活动，使开发软件原生支持欧拉生态。
- 通过CI将开发活动通过代码固化，实现开发活动自动化，减少环境搭建，开发环境切换部署的等待时间，软件在多样算力支持矩阵式开发，提高工作效率

# 原生开发服务架构 (1+1+N+M)

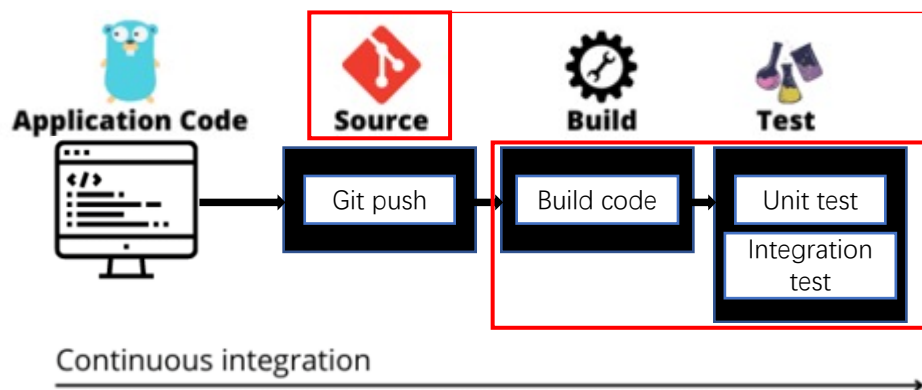


## 1条CICD流水线

```
1 version: v1.0
2
3 name: qcow2-ci
4
5 on:
6   - type: cron
7     day:
8       - Monday
9       - Tuesday
10      - Wednesday
11      - Thursday
12      - Friday
13      - Saturday
14      - Sunday
15     time: 00:00
16     start_date: 2023-11-06
17
18 vars:
19   # openEuler社区账号密码
20   # 请确保目标构建工程对应用户具备权限触发构建
21   openeuler_account:
22   openeuler_password:
23   # 待构建OS
24   os:
25   os_version:
26   # 镜像流水线分组
27   image_pipeline_group:
28
29 jobs.eulermaker-build-project:everything:
30   overrides:
31     project_name: ${vars.os}-${vars.os_version}:everything
32     # 构建类型 (必填)
33     # full - 全量构建
34     # incremental - 增量构建
```



# 1套灵活强大的描述语言



## 流水线yaml: webhook触发声明

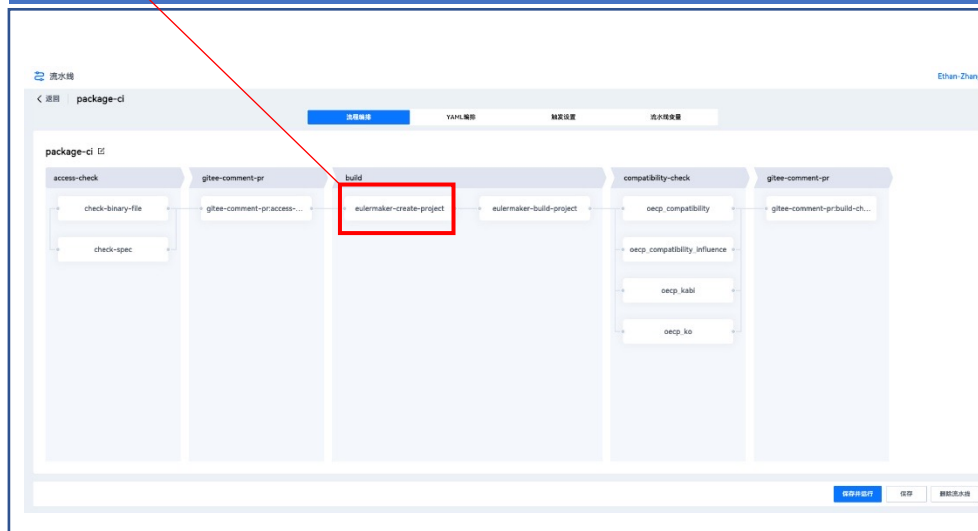
```
on:
  # PR提交触发
  - type: webhook/pr
    # 仓库地址
    git_repo: https://gitee.com/src-openeuler/kernel.git
    # 仓库分支
    branch: master
  # PR评论触发
  - type: webhook/note
    # 仓库地址
    git_repo: https://gitee.com/src-openeuler/kernel.git
    # 源码仓分支
    branch: master
    notable_type: PullRequest
    # 触发钩子 (可自定义增减)
    note:
      - /retest
```

## 原子任务yaml: eulermaker-create-project模板

```
master | lkp-tests / programs / eulermaker-create-project / jobs / eulermaker-create-project.yaml |
eulermaker-create-project.yaml 2.33 KB
张以正 提交于 1个月前 · Feat: add access-check, etc.

1 # This job is a place holder, you must set necessary params before running it.
2
3 suite: eulermaker-create-project
4 category: functional
5
6 secrets:
7   # openEuler 社区账号信息
8   ACCOUNT:
9   PASSWORD:
10
11 project_params:
12   # 工程名 (必填)
13   os_project:
14   # 工程分支, 作为软件包仓库分支备选项
15   spec_branch: openEuler-22.03-LTS-SP2
16   # 软件包
17   package_repos:
18   # 软件包名 (必填)
19   - spec_name: bcc
20   # 软件包仓库url (必填)
21   spec_url: https://gitee.com/src-openeuler/bcc.git
22   # 软件包仓库分支 (选填, 默认master分支)
23   spec_branch: master
24   - spec_name: redis
25   spec_url: https://gitee.com/src-openeuler/redis.git
```

## 流水线yaml: 控制流结构声明(v1.0语法)



```
sequence:
  parallel:access-check:
    check-binary-file:
    check-spec:
  gitee-comment-pr:access-check:
  matrix:
    arch:
      - aarch64
      - x86_64
  sequence:build:
    eulermaker-create-project:
    eulermaker-build-project:
  parallel:compatibility-check:
    oecp_compatibility:
    oecp_compatibility_influence:
    oecp_kabi:
    oecp_ko:
  gitee-comment-pr:build-check:
```

# 1套灵活强大的描述语言

## matrix矩阵声明(v1.0语法)

```
sequence:
  parallel:access-check:
    check-binary-file:
    check-spec:
  gitee-comment-pr:access-check:
  matrix:
    os|os_version:
      - openeuler | 22.03-LTS-SP1
      - openeuler | 22.03-LTS-SP2
      - openeuler | 22.03-LTS-SP3
    arch:
      - aarch64
      - x86_64
  sequence:build:
    eulermaker-create-project:
    eulermaker-build-project:
  parallel:compatibility-check:
    oecp_compatibility:
    oecp_compatibility_influence:
    oecp_kabi:
    oecp_ko:
  gitee-comment-pr:build-check:
```



- 一次声明，多种组合并行运行
- 提供矩阵总览视图，多os多架构运行结果一眼掌握

# 1套灵活强大的描述语言



## 上下文变量引用

```
jobs.eulermaker-build-project:everything:
  overrides:
    project_name: ${vars.os}-${vars.os_version}:everything
    # 构建类型 (必填)
    # full - 全量构建
    # incremental - 增量构建
    build_type: full
    build_arch: ${matrix.arch}
    secrets:
      ACCOUNT: ${vars.openeuler_account}
      PASSWORD: ${vars.openeuler_password}
    testbox: vm-2p8g

jobs.eulermaker-build-project:epol:
  overrides:
    project_name: ethan:${vars.os}-${vars.os_version}:epol
    # 构建类型 (必填)
    # full - 全量构建
    # incremental - 增量构建
    build_type: full
    build_arch: ${matrix.arch}
    secrets:
      ACCOUNT: ${vars.openeuler_account}
      PASSWORD: ${vars.openeuler_password}
    testbox: vm-2p8g
```

➤ jobs均具备vars(流水线变量)、matrix(矩阵)、jobs(前置jobs输出)、event(事件)等上下文, 在声明原子任务时, 可以通过`\${expr}`从这些上下文引用变量, 获取提交时的实际值

## python表达式支持

```
jobs.majun-codecheck:
  overrides:
    pr_url: ${event.pr_url.split('#')[0]}
    testbox: vm-2p8g

jobs.eulermaker-create-image:
  overrides:
    image_project_params:
      pipeline_info:
        pipeline_name: ${datetime.datetime.now().year} ...
        group: ${vars.image_pipeline_group}
        category: standard
        scene: cloud
        image_format: qcow2
        arch: ${matrix.arch}
      image_config:
        release_image_config:
          repo_url: >
            http://192.168.137.90/api/${jobs.eulermaker-build...}
            http://192.168.137.90/api/${jobs.eulermaker-build...}
        product: ${vars.os.lower() }
```

➤ 在声明原子任务时, 也可以通过`\${expr}`的方式对引用的变量使用python表达式。如本示例, pr\_url的实际取值为此job事件上下文中pr\_url实际值按'#'字符分割数组的第一位元素。

# N种预定义模板

## 欧拉开发活动定义

### □方案级CI，上游/社区开发者

- 欧拉系统植入到上游CI中，与上游交互
- 直接关联上游，与上游交互
- 关联社区项目，与社区开发者交互

### □版本级CI，release/qa

- 日构建，每日版本持续集成，与release/qa
- Beta、版本测试，版本发布活动集成
- Update 维护版本发布

### □项目级CI，上游/社区开发者

- 上游版本、补丁、特性回合等活动，持续集成
- 社区bug修复
- 覆盖SIG组

### □个人级CI，社区/本地开发者

- 代码ci，按照语言进行代码静态检测，编译
- 上库前开发自测

## 开发者服务

### 智能服务

#### □ upStream-订阅

- 更新补丁
- 更新特性
- 版本演进

#### □ issue-订阅

- 论坛问题
- 仓库问题
- 测试问题

#### □ 交互服务

- 复现
- 登录

#### □ 高阶-CI

- 性能看护
- 调优

### PR提交

#### □ Code-CI

- 防投毒
- 片段引用
- 敏感信息排查
- 合规性
- 安全函数
- 编译选项

#### □ Kernel-CI

- CheckPatch
- 随机配置
- 安装启动
- KABI

#### □ Package-CI

- Spec 规范
- 构建
- ABI检测
- 反向API检测
- 安装/校验

### 持续集成

#### □ Repo-CI

- 工程创建
- 工程配置
- 工程触发(全量/增量单包)

#### □ ISO-CI

- OS构建
- AT-含Ltp-smoke
- 软件包安装校验
- 异构版本号校验
- 版本后缀正确性
- 签名校验

#### □ Image CI

- 容器AT
- 启动测试

#### □ Qcow2 CI

- 虚拟机AT

#### □ App-CI

- 构建、部署
- 应用AT测试

#### □ 性能-CI

- 反馈优化

### 版本发布

#### □ 单包test-CI

- 升级/回退
- 命令/服务启停

#### □ 版本test-CI

- 功能测试
- 安全测试
- 可靠性
- 性能测试
- 兼容性

#### □ update-CI

- 漏洞、病毒扫描
- 功能测试

#### □ Release-CI

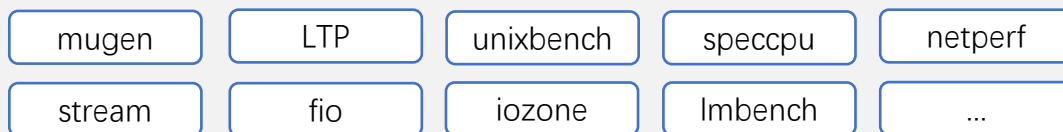
- 官方仓
- 审核
- 发布



# M种原子任务

## 原子任务框架：lkp-tests

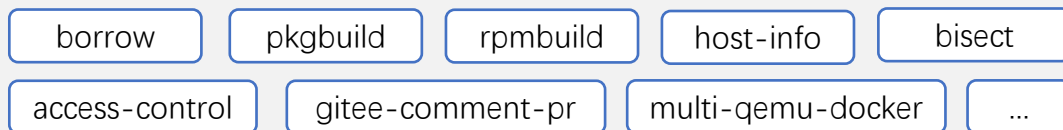
### 测试套件适配



### 第三方服务API调用



### 原生支持



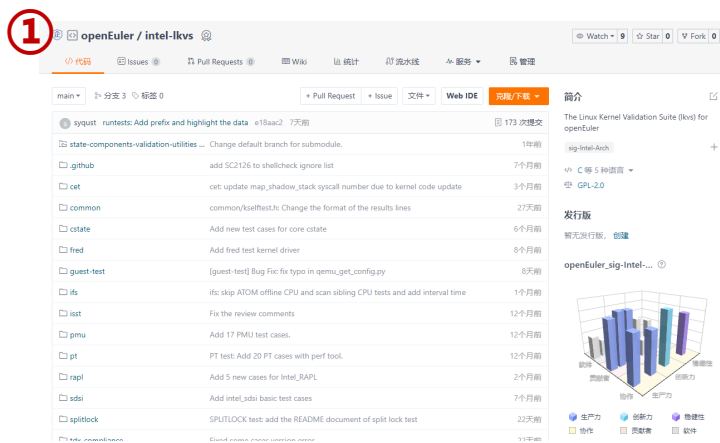
- 原子任务(jobs)是能被原生开发流水线服务编排的最小单位
- 原子任务(jobs)按一定先后顺序串行或并行地组合在一起即为一条流水线(workflows)
- 原子任务(jobs)在流水线模板(workflows)中靠任务名声明，合法与否取决于lkp-tests中是否存在
- 原子任务(jobs)和流水线模板(workflows)默认来源于<https://gitee.com/compass-ci/lkp-tests.git>，也支持来源于不同的lkp-tests fork仓或者符合规定的独立仓库



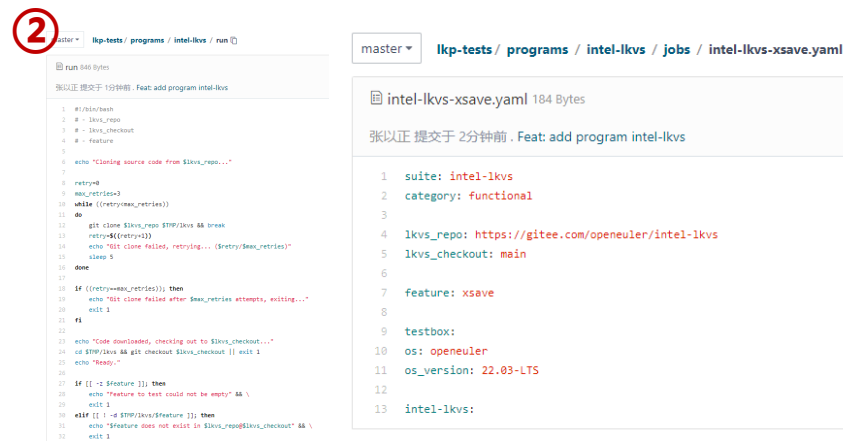
# Intel-Ikvs流水线开发实例



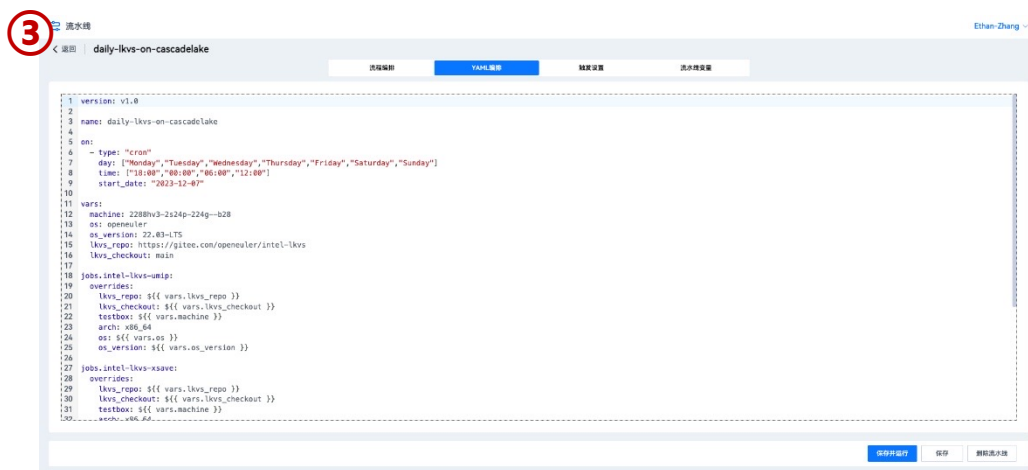
## Intel-Ikvs测试套件代码仓



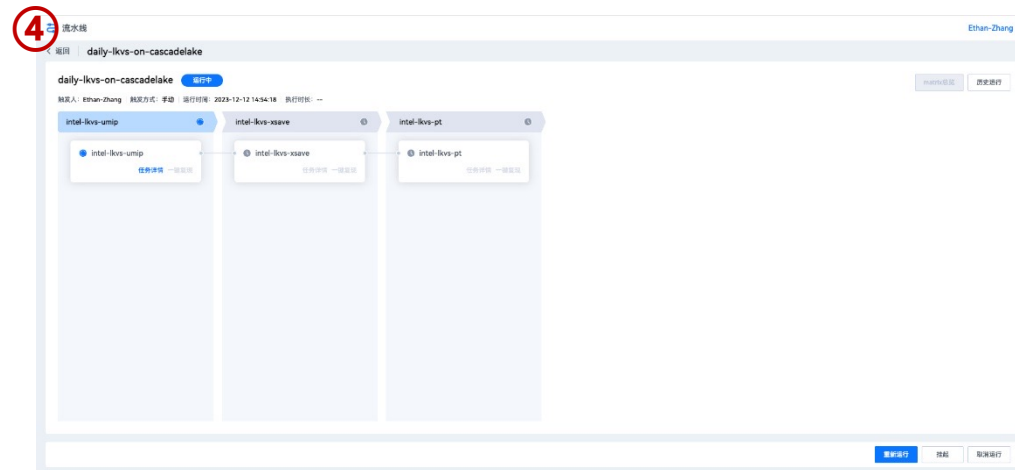
## lkp-tests添加原子任务Intel-Ikvs



## 编排intel-Ikvs流水线



## 每日运行于搭载intel cascadelake的服务器



# 快速定义自用流水线



## 代码仓门禁流水线

选择access-control/code-ci模板

设置webhook触发  
指定触发仓库、分支与事件

复制webhook url与签名密钥到git仓库

git仓库触发对应事件  
流水线开始运行

## Linux发行版每日构建AT流水线

选择continuous-integration/iso-ci模板

设置定时触发  
设定流水线每天00:00准点触发

配置EulerMaker帐号密码  
根据发行版构建工程填写发行版信息

确保EulerMaker存在发行版构建工程

定时触发  
流水线开始运行



# 欢迎大家申请试用



流水线

user001

我的流水线

全部流水线

我的项目

我参与的 200

新建流水线 运行 删除

流水线名称	最近运行状态	最近运行阶段	触发信息	开始时间	执行时长	创建人	操作
流水线0001	测试中	构建	定时触发	2023-5-09 13:06	2m55s	zhangsan	运行 删除 更多
流水线0002	运行中	构建	定时触发	2023-5-09 11:06	3m26s	zhangsan	运行 删除 更多
流水线0003	未开始	构建	--	--	--	zhangsan	运行 删除 更多
流水线0004	运行失败	测试	定时触发	2023-5-07 10:06	7m33s	zhangsan	运行 删除 更多
流水线0005	运行成功	构建	定时触发	2023-5-06 10:06	8m42s	zhangsan	运行 删除 更多
流水线0006	运行超时	测试	定时触发	2023-5-05 13:06	2m55s	zhangsan	运行 删除 更多
流水线0007	运行中	构建	定时触发	2023-5-05 11:06	3m26s	zhangsan	运行 删除 更多
流水线0008	已挂起	构建	--	--	--	zhangsan	运行 删除 更多
流水线0009	运行失败	构建	定时触发	2023-5-04 11:06	7m33s	zhangsan	运行 删除 更多
流水线0010	运行成功	构建	定时触发	2023-5-03 11:06	7m33s	zhangsan	运行 删除 更多
流水线0011	阻塞	构建	定时触发	2023-5-04 10:06	7m33s	zhangsan	运行 删除 更多
流水线0012	运行中	构建	定时触发	2023-5-05 11:06	3m26s	zhangsan	运行 删除 更多
流水线0013	未开始	构建	--	--	--	zhangsan	运行 删除 更多
流水线0014	运行失败	测试	webhook触发	2023-5-04 11:06	7m33s	zhangsan	运行 删除 更多
流水线0015	已取消	构建	webhook触发	2023-5-03 11:06	7m33s	zhangsan	运行 删除 更多
流水线0016	运行成功	构建	webhook触发	2023-5-04 10:06	7m33s	zhangsan	运行 删除 更多

共 200 条 40条/页 < 1 2 3 4 5 > 前往 2 页

服务官网: <https://eulerpipeline.compass-ci.openeuler.openatom.cn>

lkp-tests公仓: <https://gitee.com/compass-ci/lkp-tests>

参与方式: sig-CICD





