



货拉拉移动端devops实践



自我介绍

尹伊博

- 2013年毕业，主要从事Android客户端研发，先后就职于盛大集团、饿了么。
- 2020年9月加入货拉拉，主要负责货拉拉移动端devops平台的从 0 到 1 的建设。

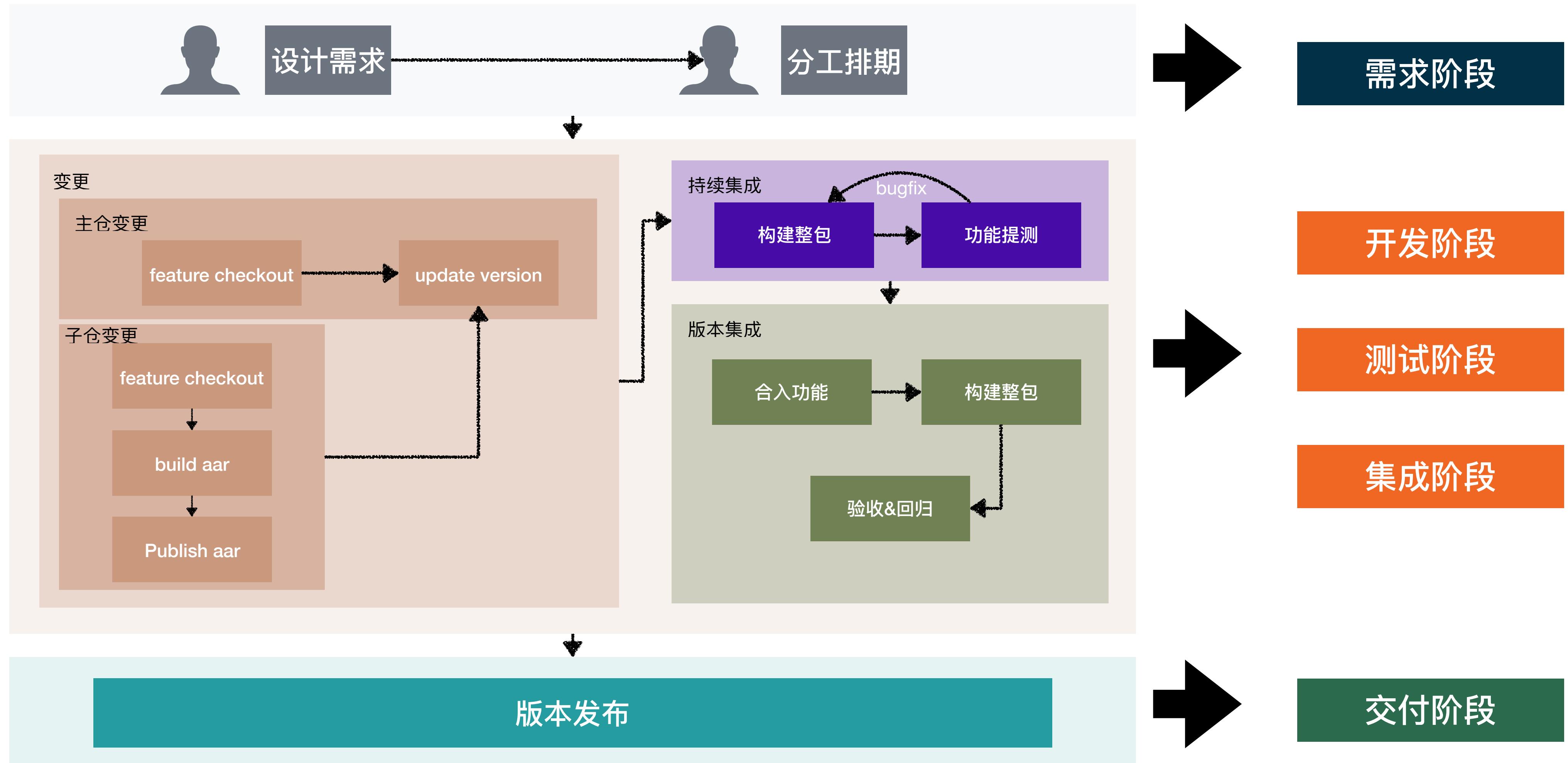
拉货就找货拉拉





- 1 货拉拉移动端devops的背景
- 2 货拉拉移动端devops介绍
- 3 货拉拉移动端devops一些核心建设
- 4 总结与展望

货拉拉移动端devops的背景 – 研发流程



拉货就找货拉拉

理想情况：各阶段高效扭转，达到稳定交付的一个良性循环



货拉拉移动端devops的背景 – 现状



打包难

- 本地打包提测，**环境配置各异**
- **打包时间长**，经常发布打包到凌晨
- 研发模式和分支管理不统一，**没有统一标准**
- 线下交流，**各个环节割裂**



发布难

- 没有灰度能力，**线上频繁冒烟**
- 没有精准发布到点的能力，导致产品**验证难，反馈慢**
- 缺少统一的**强更能力**



止损难

- 端上出现问题，只能通过重新发布上市场的方式，**止损周期长，体验差**
- 缺少**降级和热修的能力**

拉货就找货拉拉





货拉拉移动端devops介绍



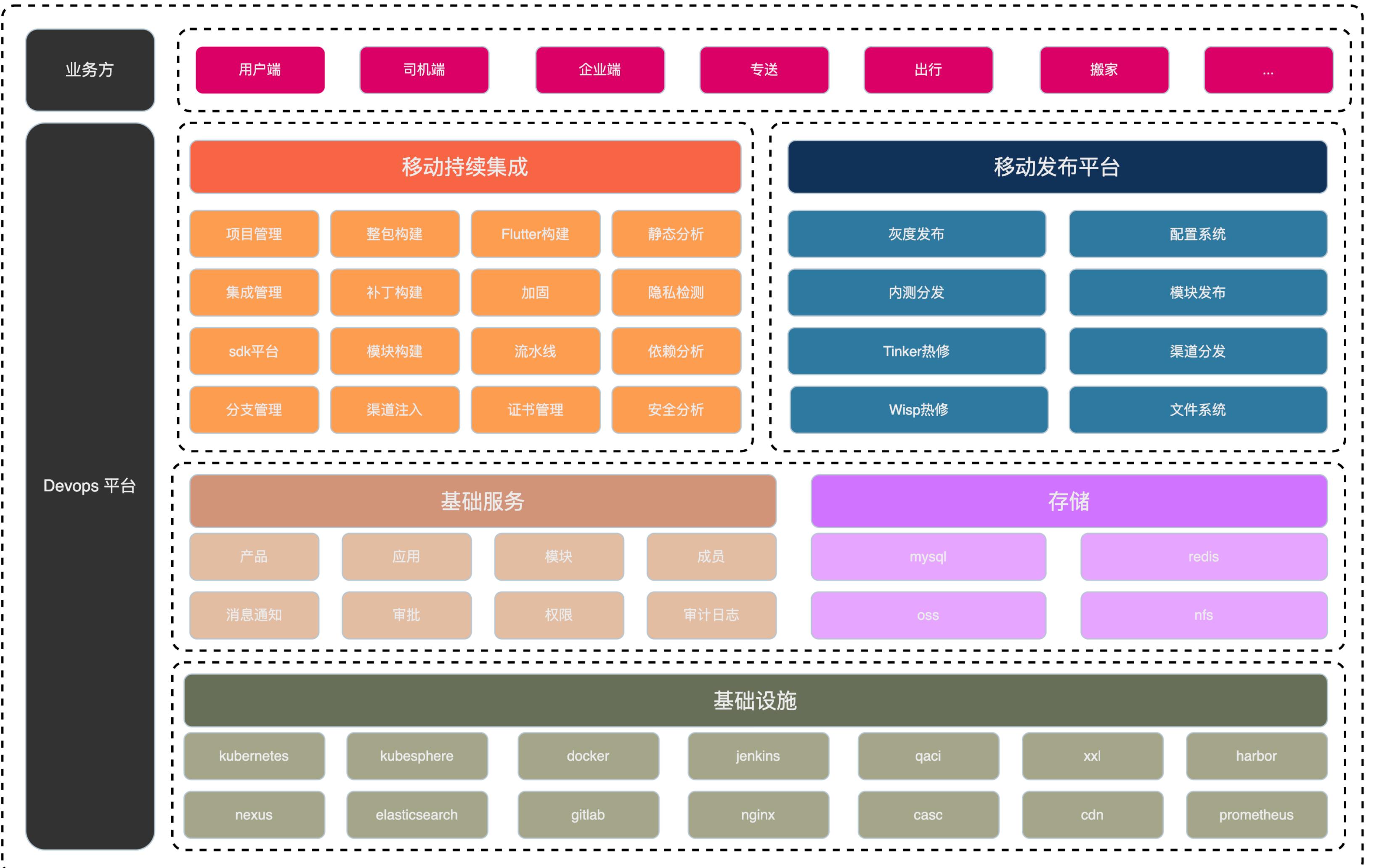
货拉拉移动端devops介绍



拉货就找货拉拉



货拉拉移动端devops介绍



拉货就找货拉拉





货拉拉移动端devops一些核心建设 — 集群建设

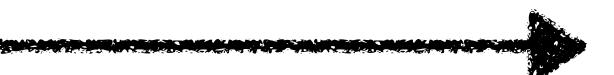


集群建设 — 机器管理

早期核心目标：远端构建能力

- 机器较少
- 任务种类较少

解法



- 人工维护构建容器
- 容器持久运行
- 资源分配手动维护

拉货就找货拉拉

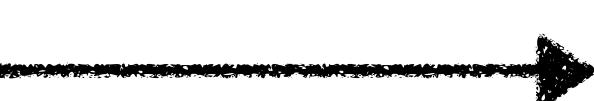


集群建设 — 机器管理

现阶段目标：流水线能力

- 机器多 ↑
- 任务种类多 ↑

新解法



- k8s 管理节点和容器
- 容器按需创建
- 资源动态分配

```
- name: android
image: 'harbor.mdap-devops.huolala.work/mdap-devops/android:0.0.5'
workingDir: /home/mdapdevops
resources:
limits:
cpu: '12'
memory: 30Gi
requests:
cpu: '8'
memory: 16Gi
```

名称	状态	角色	CPU 用量	内存用量	容器组	已分配 CPU	已分配内存
node8 192.168.106.113	运行中	工作节点	3% 0.49/16 核	20% 6.26/31.26 GiB	6% 7/110	4.522 核 (28%) 资源预留	16.679 GiB (57%) 资源预留
node9 192.168.106.114	运行中	工作节点	3% 0.34/16 核	6% 1.85/31.26 GiB	5% 6/110	1.472 核 (9%) 资源预留	1.265 GiB (4%) 资源预留
node7 192.168.106.52	运行中	工作节点	14% 2.14/16 核	9% 2.88/31.26 GiB	7% 8/110	0.472 核 (3%) 资源预留	0.288 GiB (0%) 资源预留
node6 192.168.106.51	运行中	工作节点	15% 2.53/16 核	18% 5.64/31.36 GiB	18% 20/110	1.447 核 (9%) 资源预留	1.911 GiB (6%) 资源预留
node5 192.168.106.53	运行中	工作节点	20% 3.07/16 核	36% 11.13/31.26 GiB	15% 17/110	1.502 核 (9%) 资源预留	1.704 GiB (5%) 资源预留
node4 192.168.106.159	运行中	工作节点	17% 2.6% 3.07/16 核	20% 11.13/31.26 GiB	14.62 核 (9%) 资源预留	1.431 GiB (4%) 资源预留	
node3 192.168.106.158	运行中 1	工作节点					
node2 192.168.146.103	运行中	工作节点					
node1 192.168.146.102	运行中	工作节点					

名称	状态	节点	容器组 IP 地址	更新时间
devops-27881880-fb5d5 已完成	已完成	node7 (192.168.106.52)	10.233.95.189	2023-01-05 18:00:00
base-lr06m 运行中	运行中	node4 (192.168.106.159)	10.233.105.177	2023-01-05 17:51:59
base-8l3gs 运行中	运行中	node4 (192.168.106.159)	10.233.105.237	2023-01-05 17:51:59
analysis-7ndsp 运行中	运行中	node6 (192.168.106.51)	10.233.108.189	2023-01-05 17:47:14
analysis-lwz8h 运行中	运行中	node9 (192.168.106.114)	10.233.98.13	2023-01-05 17:40:01
devops-27881850-5k9vx 已完成	已完成	node4 (192.168.106.159)	10.233.105.172	2023-01-05 17:30:00
analysis-xmv99 运行中	运行中	node2 (192.168.146.103)	10.233.96.47	2023-01-05 17:27:27
base-ckrn8 运行中	运行中	node2 (192.168.146.103)	10.233.96.53	2023-01-05 17:18:47
base-ft7zh 运行中	运行中	node2 (192.168.146.103)	10.233.96.48	2023-01-05 17:18:18
devops-27881820-6h8sf 已完成	已完成	node9 (192.168.106.114)	10.233.98.3	2023-01-05 17:00:00

拉货就找货拉拉

机器统一管理

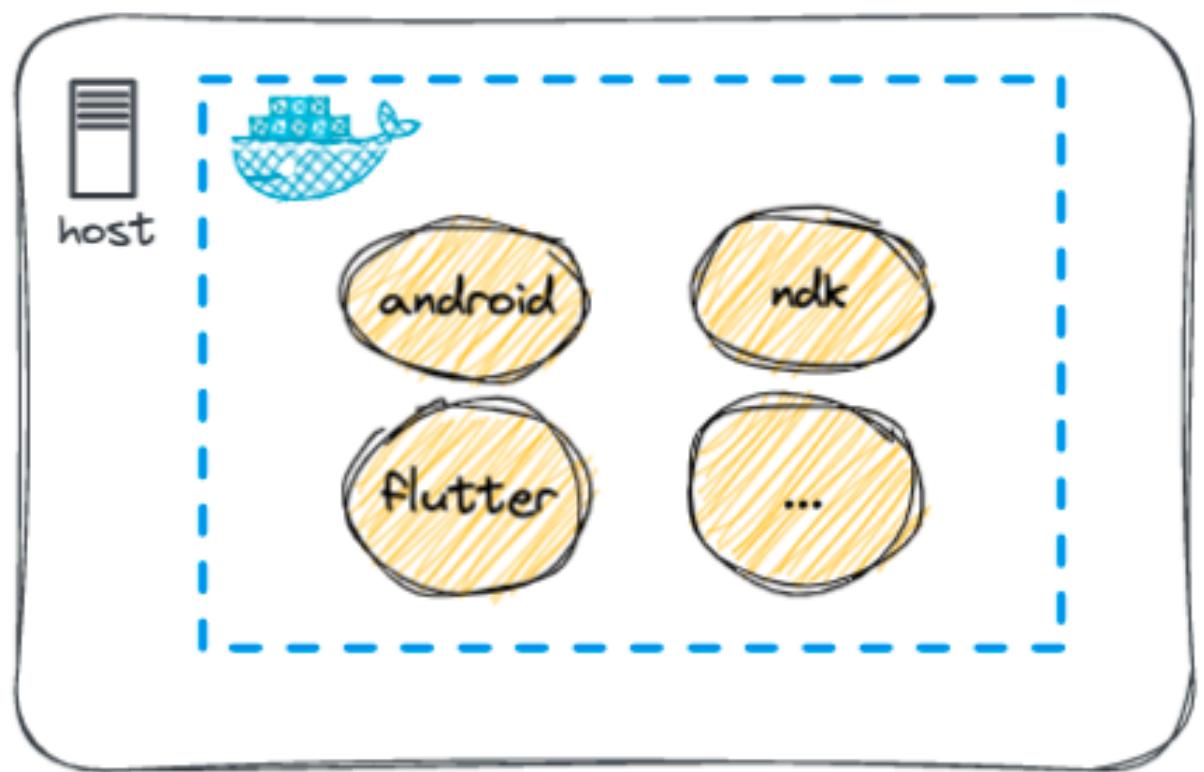
执行容器按需创建



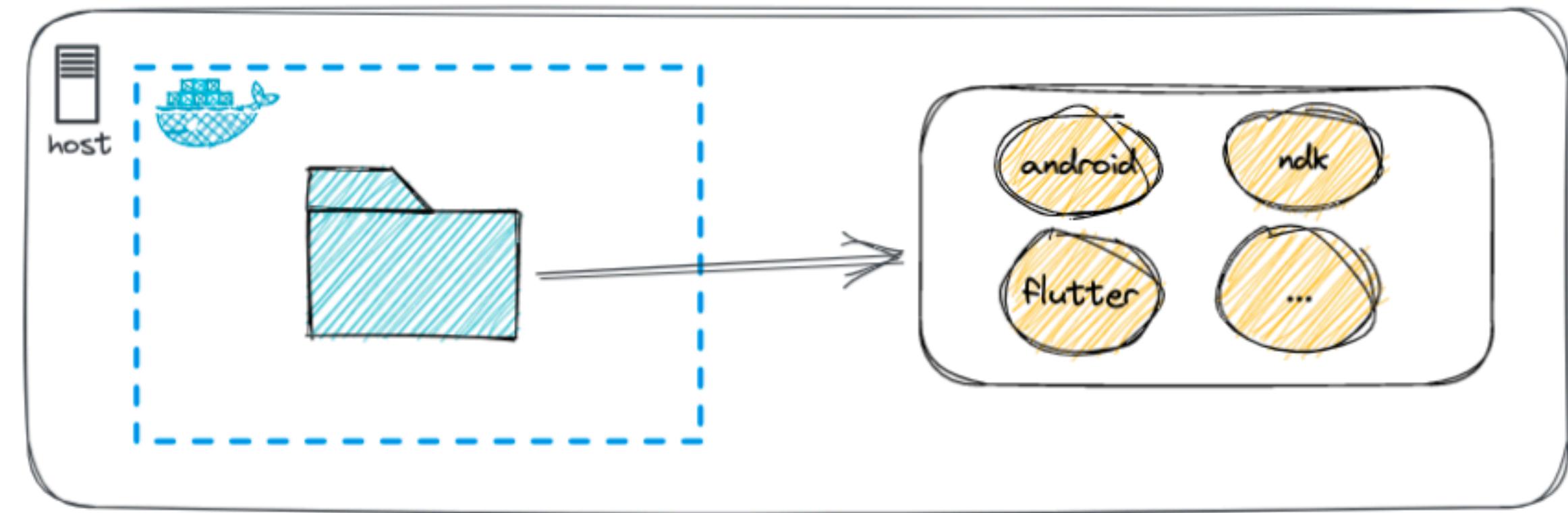
集群建设 – 存储管理

问题：如何管理构建相关工具？

1



2



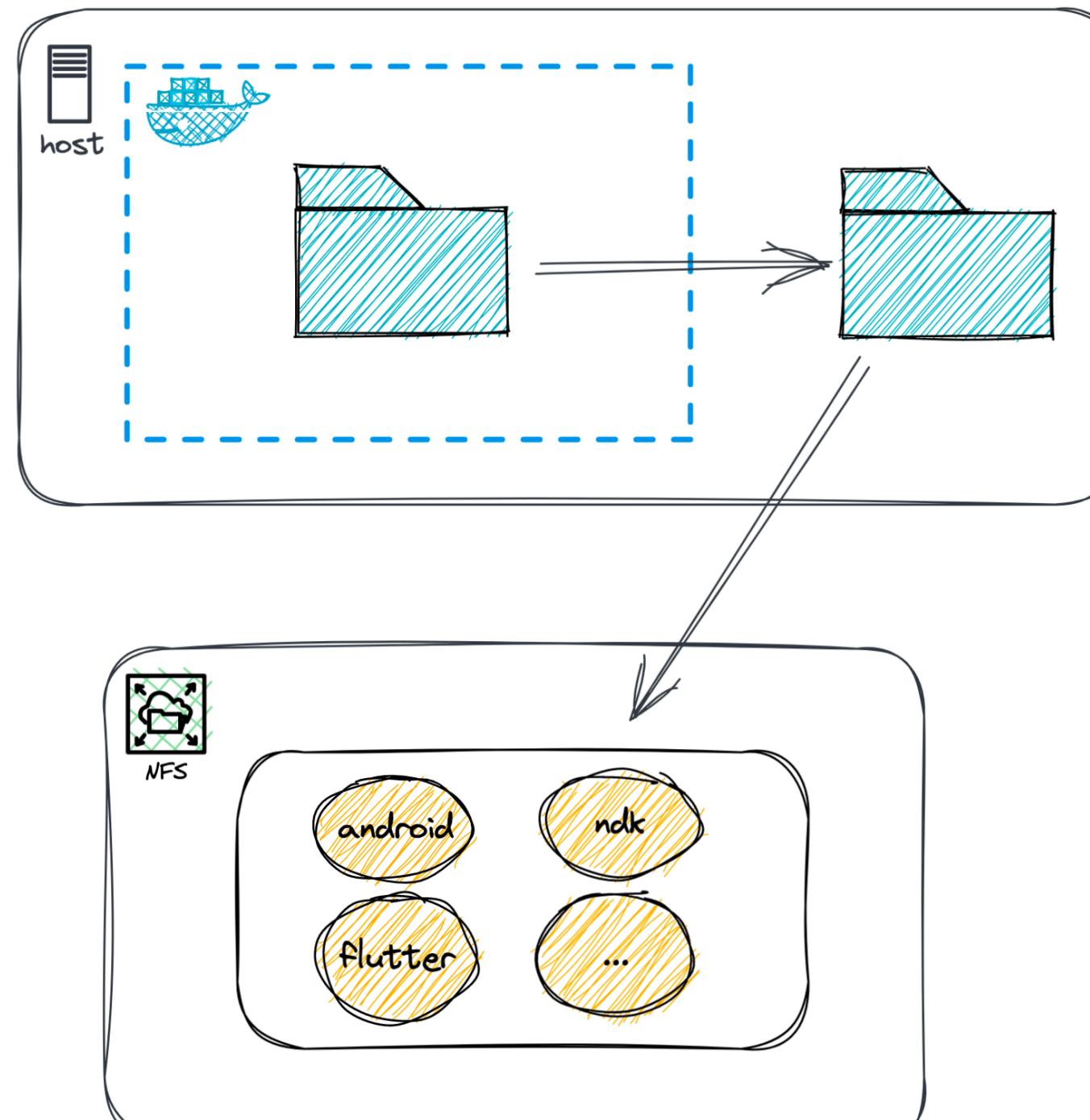
灵活性低

维护成本高

拉货就找货拉拉



集群建设 – 存储管理

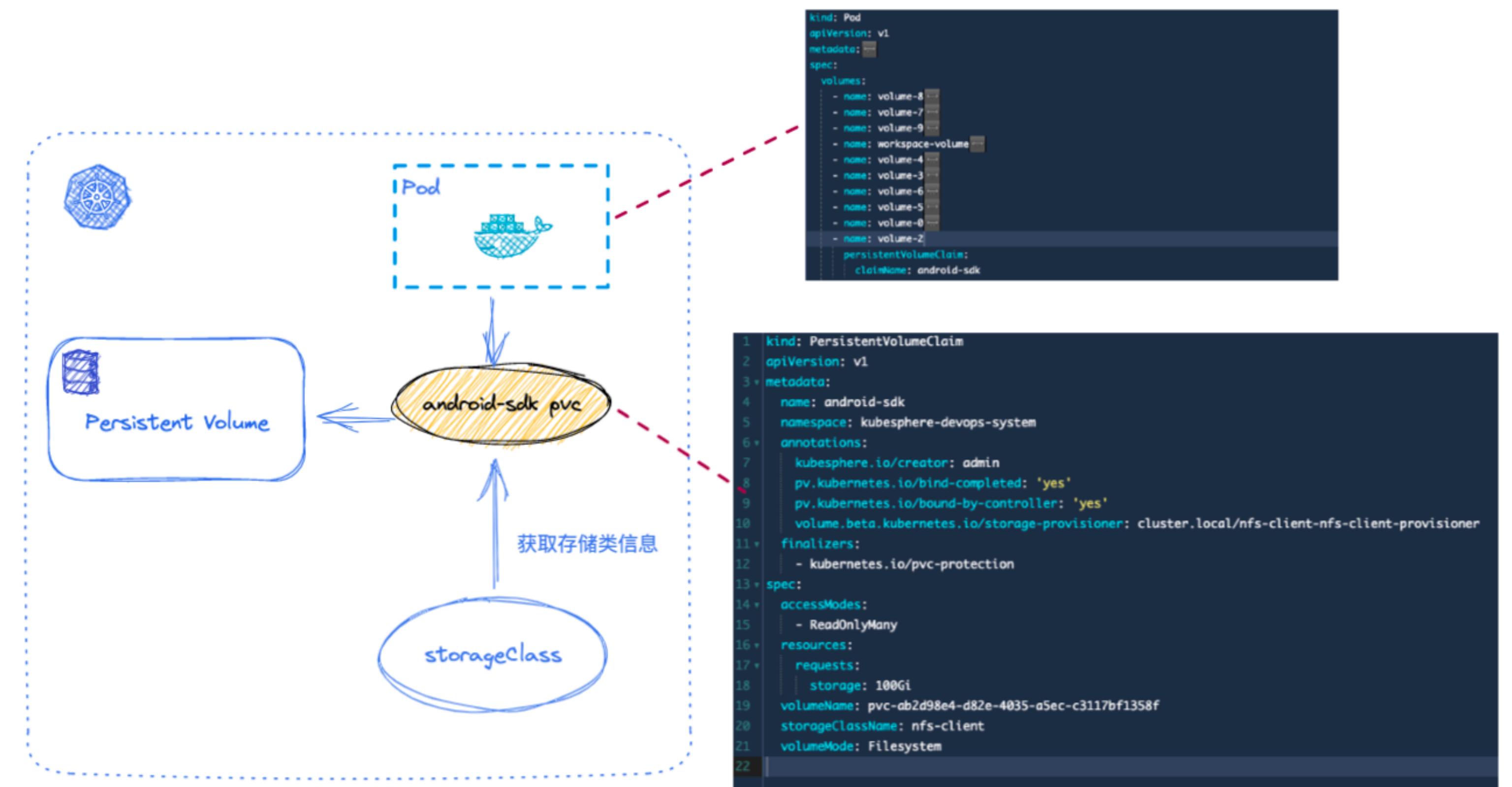


拉货就找货拉拉

易维护

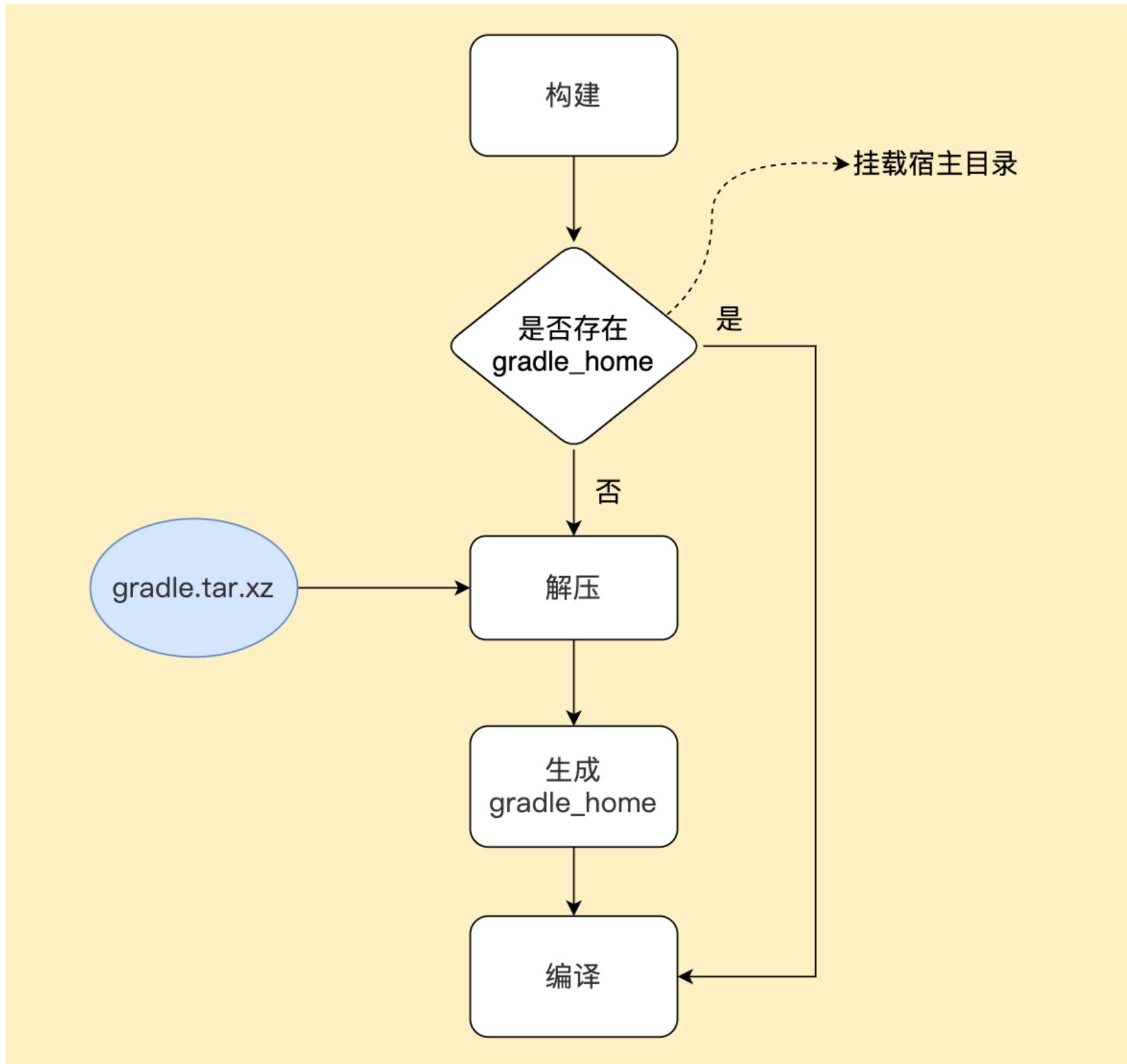
节省空间

ReadOnlyMany: 只适用部分工具



集群建设 – 存储管理

针对多节点读写场景



GRADLE_RO_DEP_CACHE

REMOTE_CACHE

拉货就找 货拉拉



集群建设 — 节点调度

问题：如何进行节点选择？

Jenkins label ?

节点亲和性

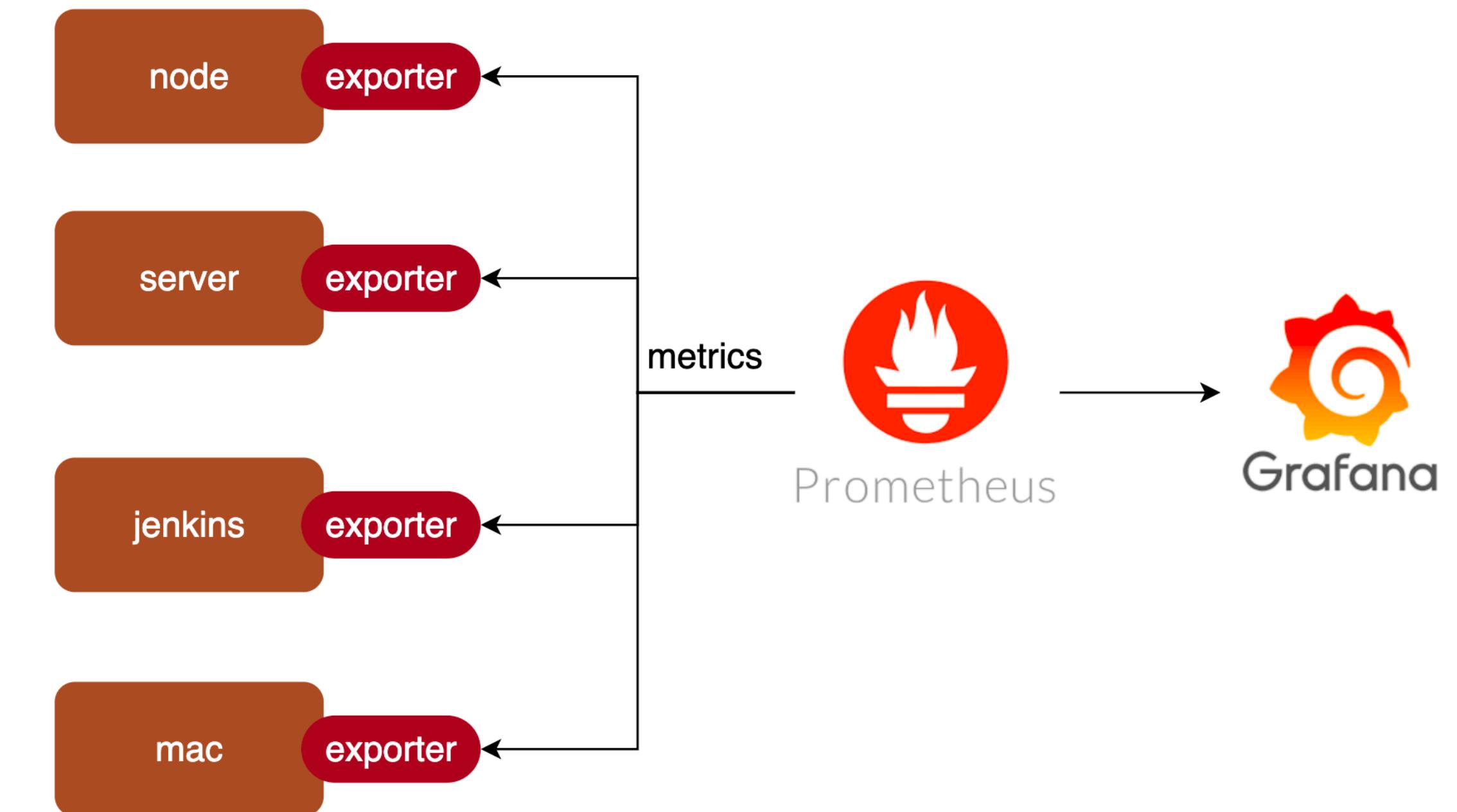
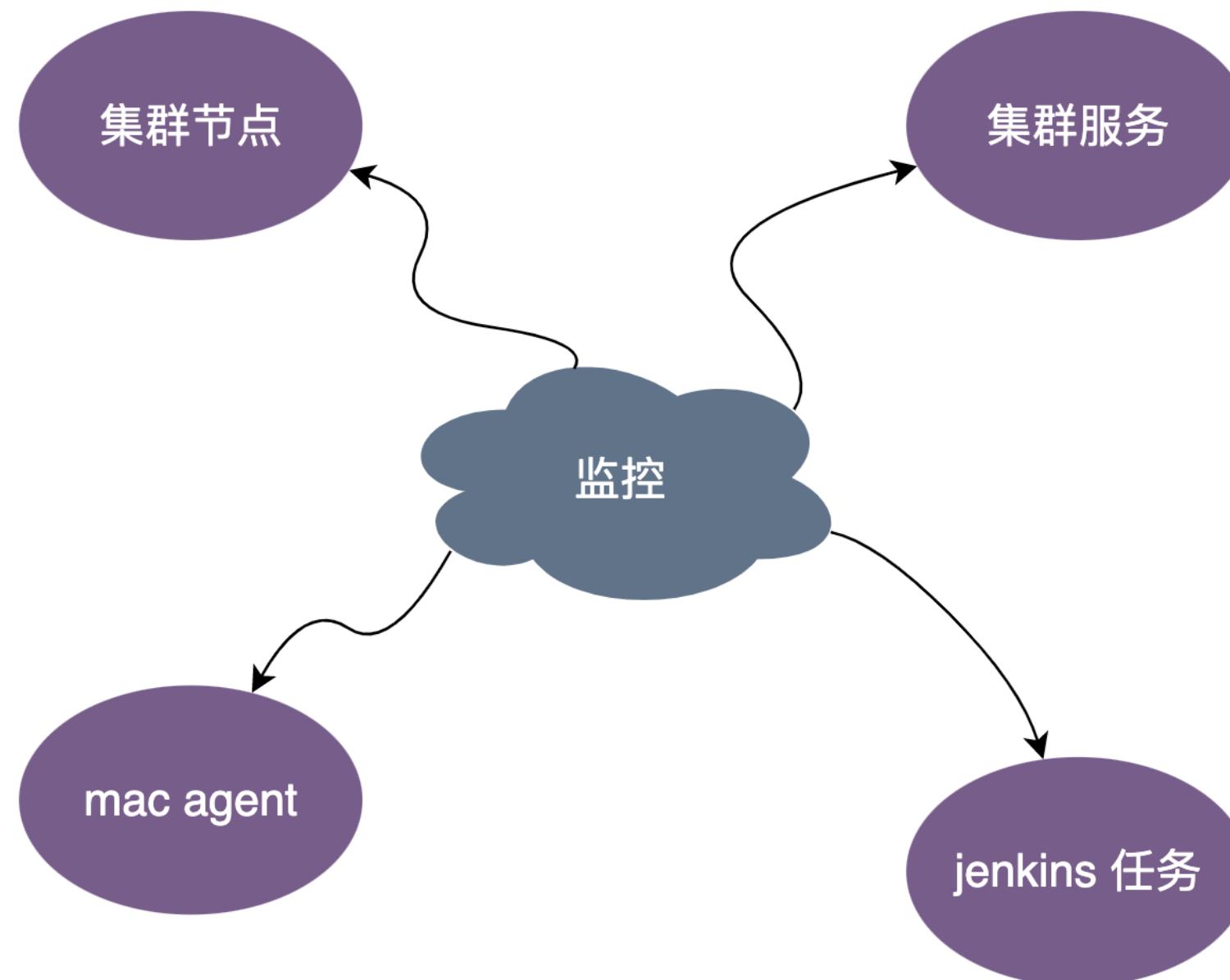
拉货就找货拉拉

```
spec:  
  affinity:  
    nodeAffinity:  
      requiredDuringSchedulingIgnoredDuringExecution:  
        nodeSelectorTerms:  
          - matchExpressions:  
              - key: mdap/ci  
                operator: In  
                values:  
                  - android  
      preferredDuringSchedulingIgnoredDuringExecution:  
        - preference:  
            matchExpressions:  
              - key: "mdap/ssd"  
                operator: "Exists"  
            weight: 1  
    containers:  
      - args:  
        env:  
          - name: "MDAP_NODE_NAME"  
            valueFrom:  
              fieldRef:  
                fieldPath: spec.nodeName  
          - name: "MDAP_NODE_IP"  
            valueFrom:  
              fieldRef:  
                fieldPath: status.hostIP
```



集群建设 – 监控

问题：如何监控稳定性



拉货就找货拉拉



集群建设 — 监控

1. Mac物理机安装node_exporter (ansible)
2. 外部资源需要定义 Endpoints
3. 设置 Service 和 ServiceMonitor

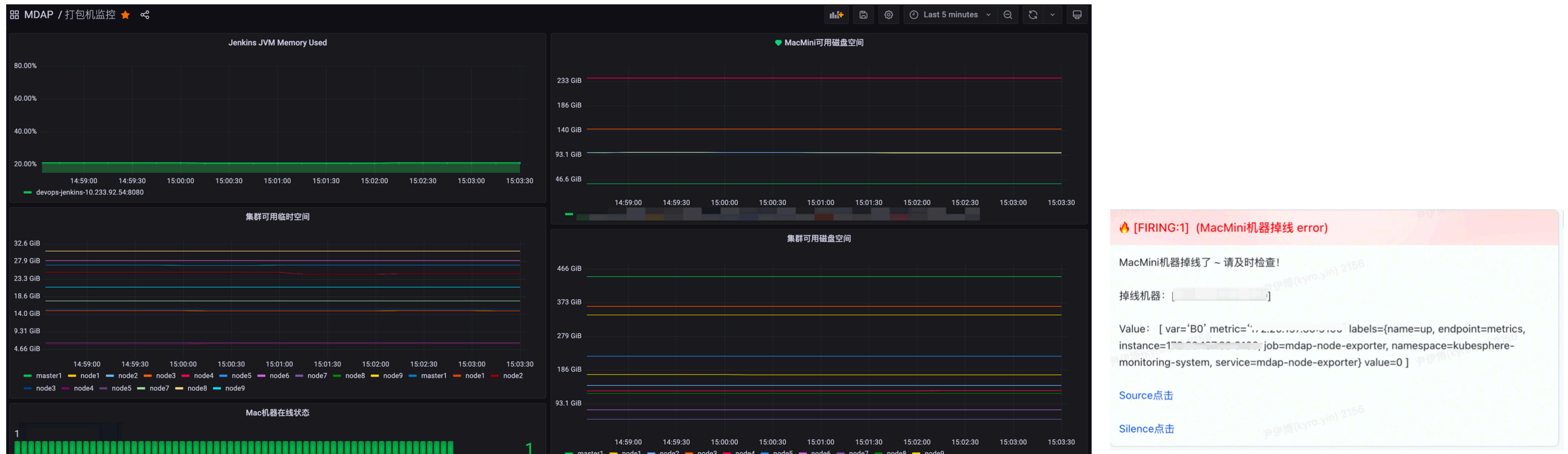
```
apiVersion: v1
kind: Service
metadata:
  name: mdap-node-exporter
  namespace: mdap-monitoring-system
  labels:
    app: mdap-node-exporter
    app.kubernetes.io/name: mdap-node-exporter
spec:
  type: ClusterIP
  ports:
  - name: metrics
    port: 9100
    protocol: TCP
    targetPort: 9100
---
apiVersion: v1
kind: Endpoints
metadata:
  name: mdap-node-exporter
  namespace: mdap-monitoring-system
  labels:
    app: mdap-node-exporter
    app.kubernetes.io/name: mdap-node-exporter
subsets:
- addresses:
  - ip: [REDACTED]
  - ip: [REDACTED]
  - ip: [REDACTED]
  - ip: [REDACTED]
```

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: mdap-node-exporter
  namespace: mdap-monitoring-system
  labels:
    app: mdap-node-exporter
    release: prometheus
spec:
  selector:
    matchLabels:
      app: mdap-node-exporter
  namespaceSelector:
    matchNames:
    - mdap-monitoring-system
  endpoints:
  - port: metrics
    interval: 10s
    path: /metrics
```

拉货就找货拉拉



集群建设 — 监控



拉货就找货拉拉

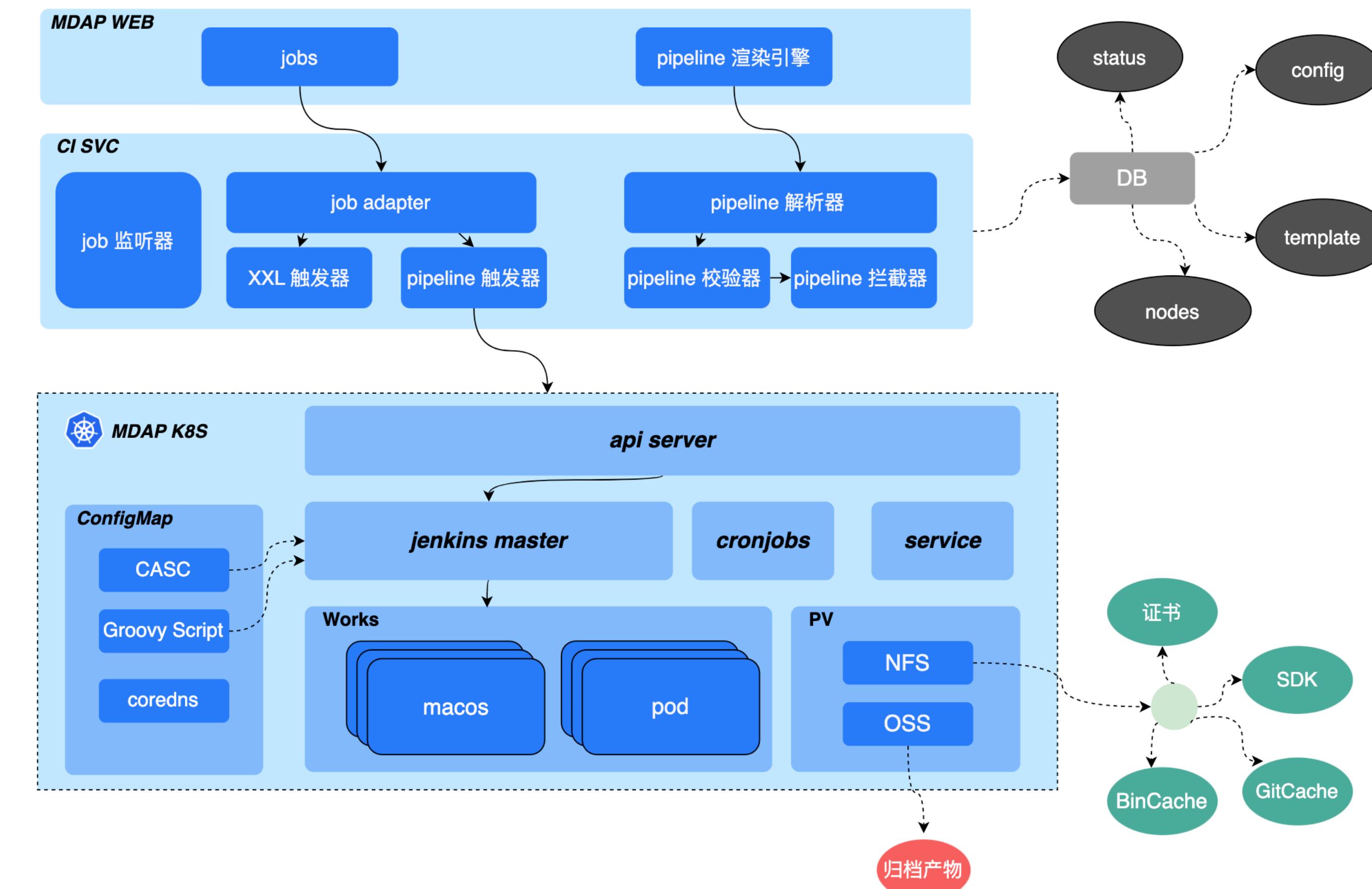


集群建设 — 总结

CI 的稳定性

减少人工运维

提高调度能力



拉货就找货拉拉



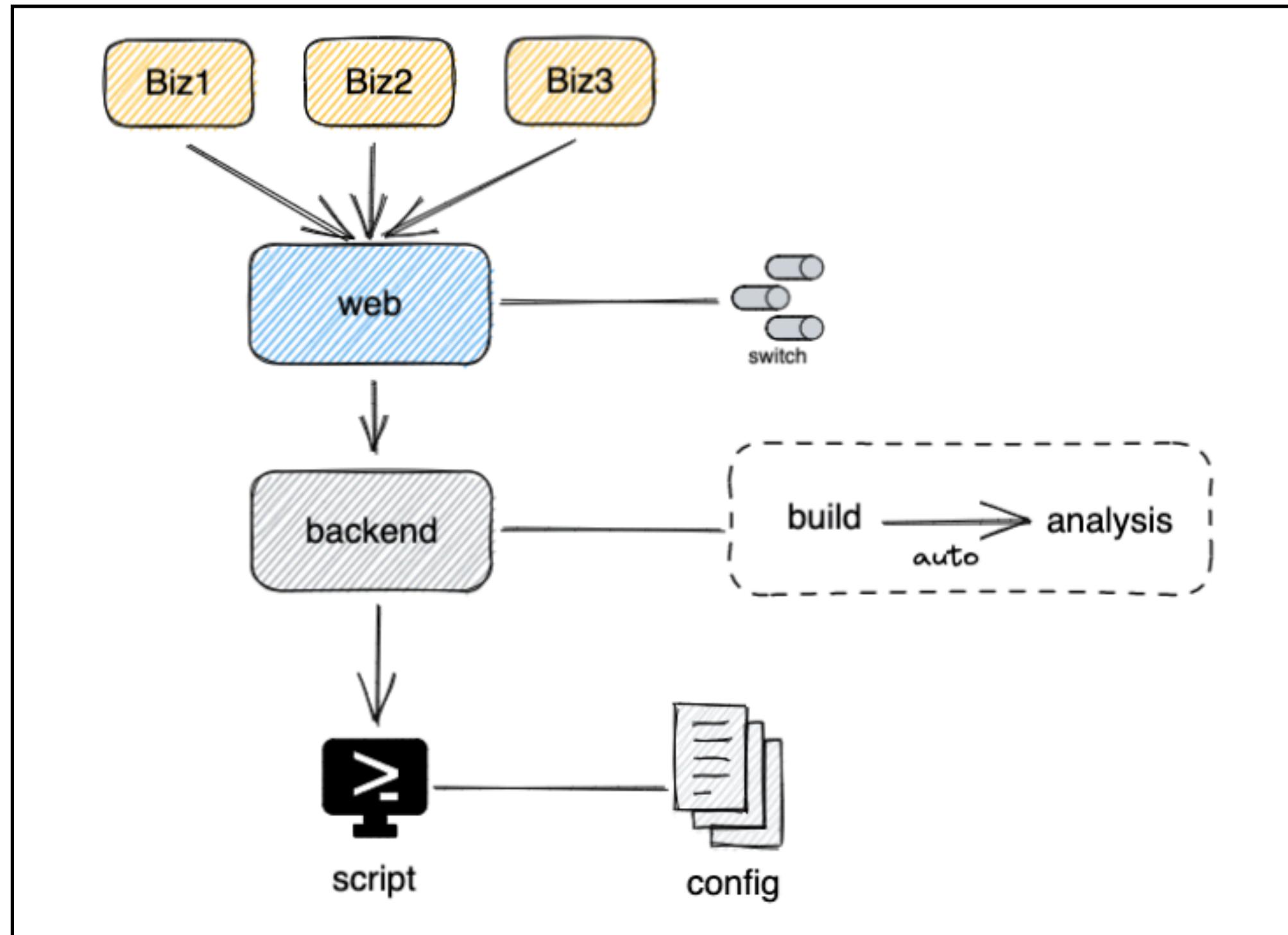


货拉拉移动端devops一些核心建设 — 流水线建设



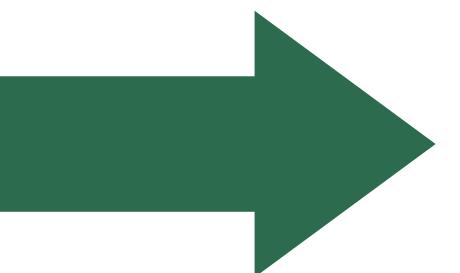
流水线建设

痛点：



- 阶段流程不一样
- 执行流程不透明
- 扩展性较低
- 改动风险大
- 使用成本较高

目标



脚本独立

自定义组合

易用

拉货就找货拉拉



流水线建设

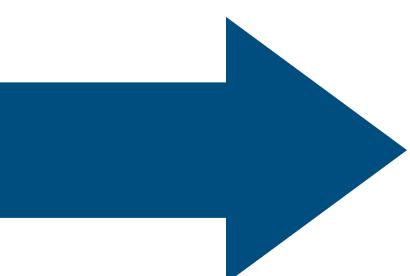
如何使用?



脚本独立

自定义组合

易用



原子服务

流水线

实现

可视化编排

拉货就找货拉拉

流程清晰

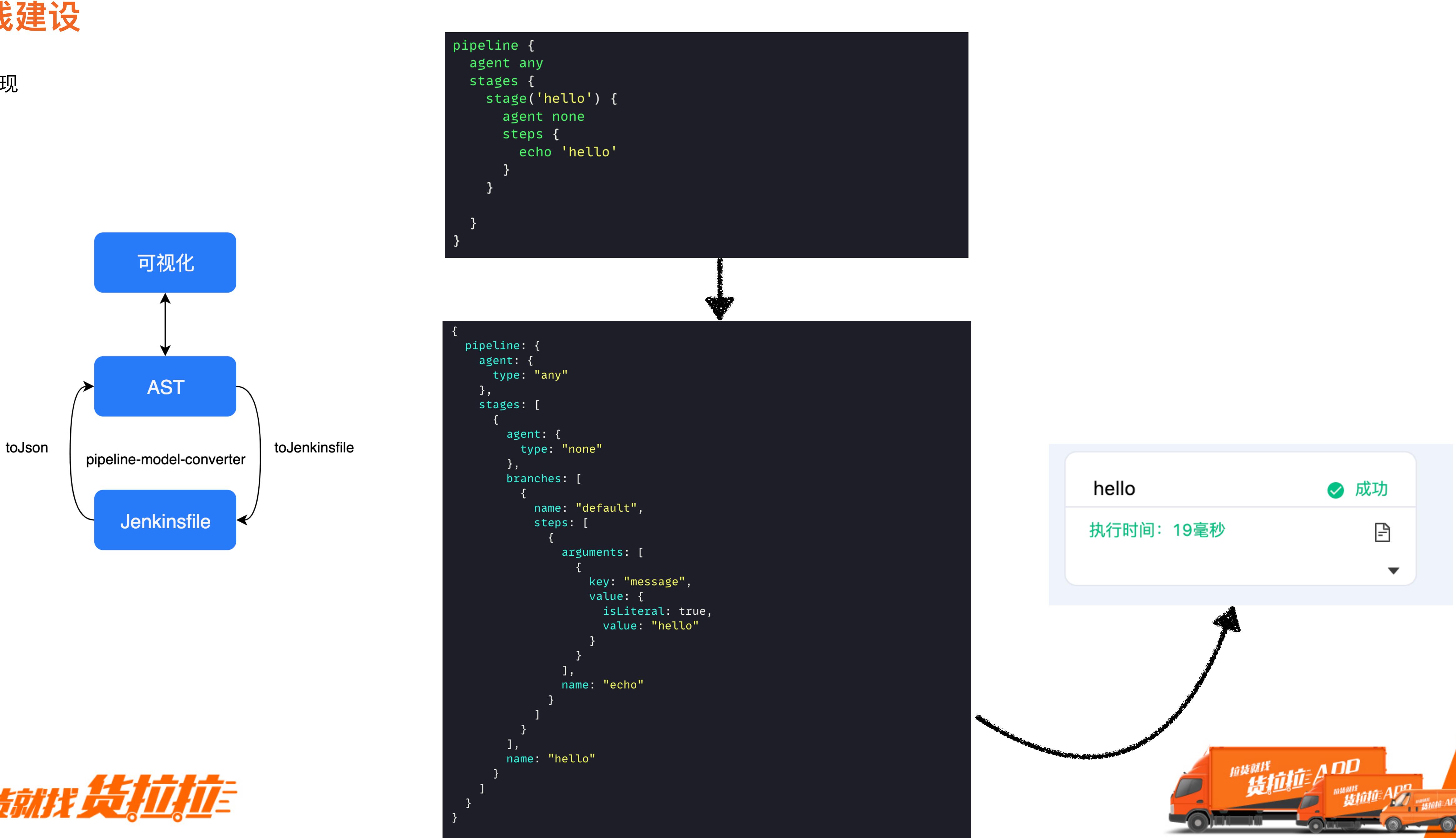
配置灵活

可复用



流水线建设

可视化实现



流水线建设

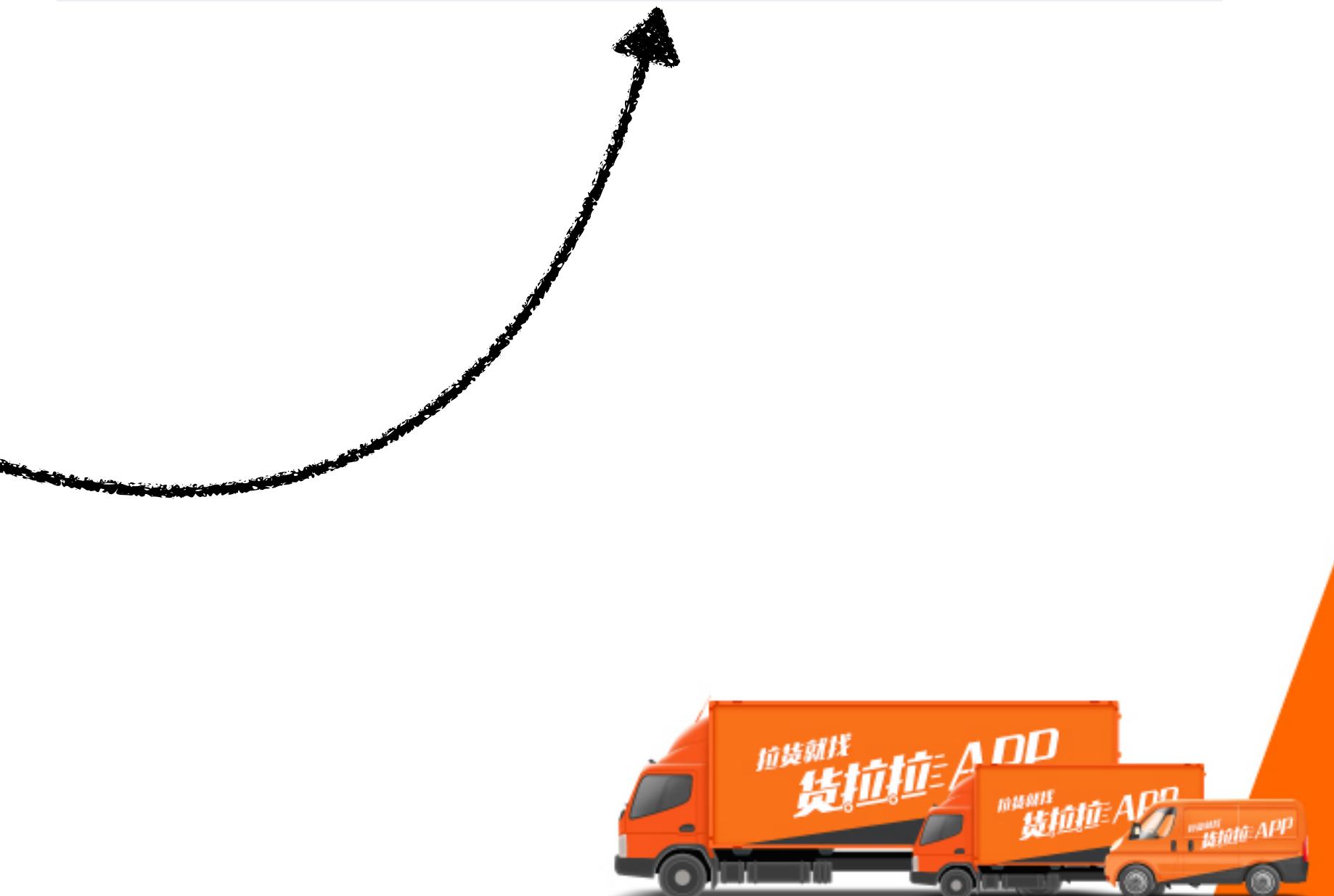
卡口&重试

```
def <R> R inquiredAction(final Closure<R> action, final int timeOut = 30) {  
  
    try {  
        return action.call()  
    } catch (final exception) {  
        def userChoice  
        timeout(time: timeOut, unit: 'MINUTES') {  
            userChoice = input(  
                message: 'error',  
                id: uuid,  
                parameters: [  
                    choice(  
                        name: 'action',  
                        choices: ['retry', 'continue', 'abort'].join('\n'),  
                        description: 'desc'  
                    )  
                ]  
            )  
        }  
        switch (userChoice) {  
            case 'Retry':  
                return inquiredAction(action)  
            case 'Continue':  
                return null  
            default:  
                throw exception  
        }  
    }  
}
```



```
inquiredAction {  
    // TODO  
}
```

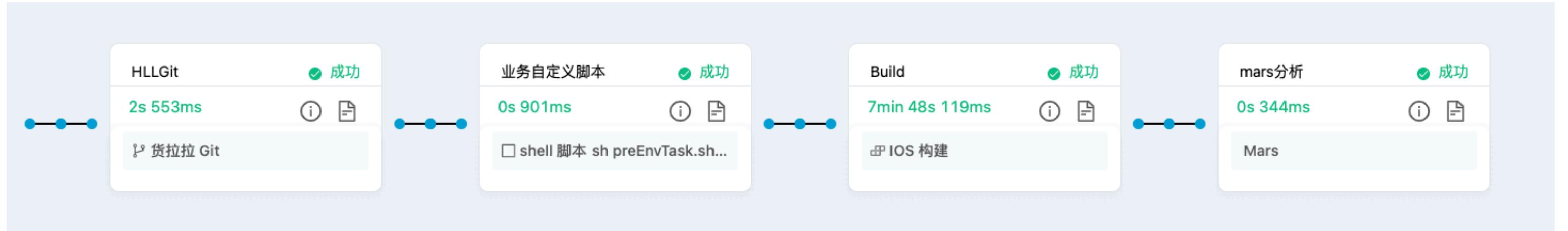
拉货就找货拉拉



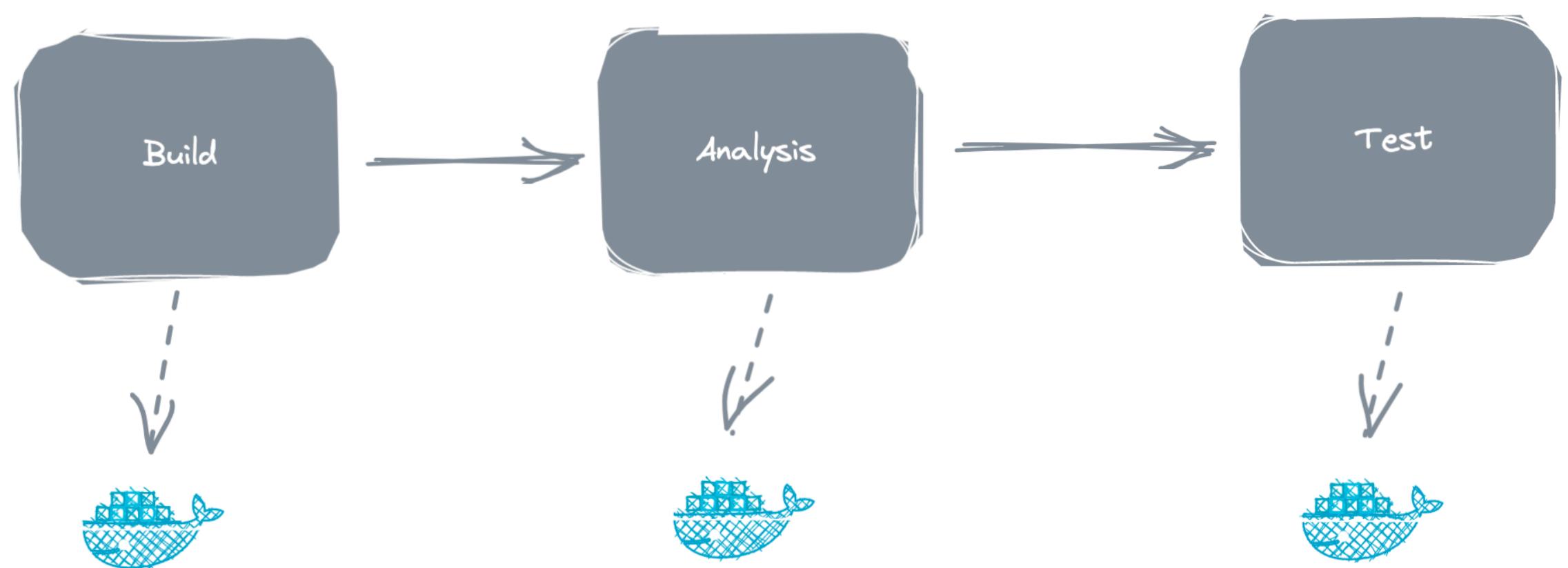
流水线建设

流水线的规范

早期



1. 原子任务相互耦合
2. Node节点资源浪费



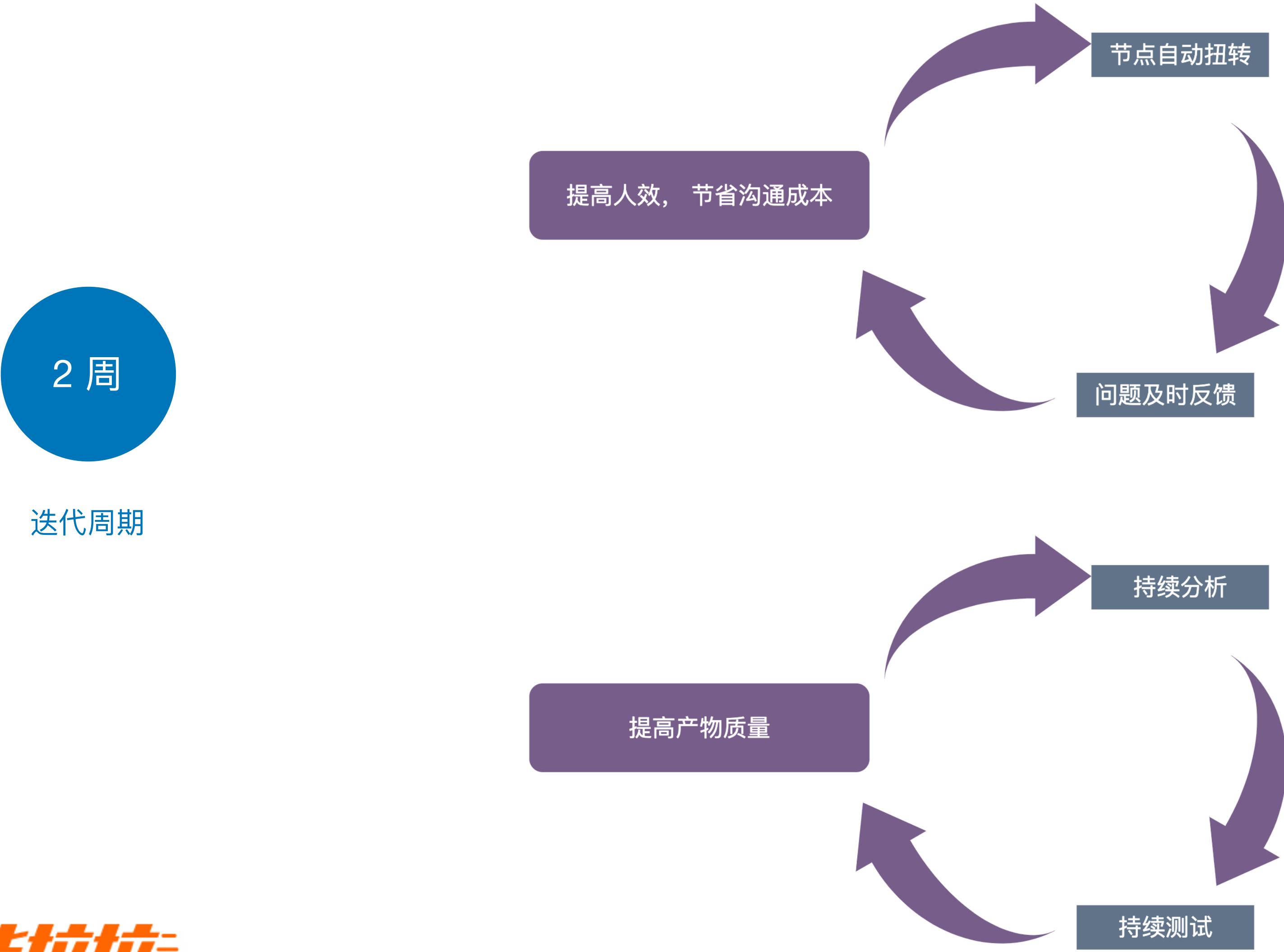
1. 原子任务相互独立
2. 尽量减少前后依赖
3. 避免使用全局Agent

拉货就找货拉拉



基于流水线的实践

如何在具体的业务场景中去落地来达到最好的效果呢?



拉货就找货拉拉



基于流水线的实践

第一步 感知研发阶段



新建项目

* 项目类型: 需求 hotfix 临时项目

* 所属应用: 货拉拉司机端-Android

* pmis需求: 请输入您在pmis创建的需求名称

* 项目名称: 请输入您要创建项目的名称

展开更多信息 ▾

取消 确定

新建集成区

* 所属应用: 货拉拉司机端-Android

* 版本号: 6.3.24 迭代版本 状态: 未使用

* 集成区名称: 【Android】6.3.24版本集成区

发布时间: 请选择日期

集成区描述: 请输入集成区描述

展开更多信息 ▾

取消 确定

基础信息

集成区名称: 【Android】6.3.9版本集成区 版本号: 6.3.9

所属应用: 货拉拉司机端-Android 集成分支:

集成详情 整包构建 依赖管理 **发布单**

发布单列表

ID	基础信息	构建号	构建产物	创建人/创建时间	操作
暂无数据					

删除集成区 编辑成员 编辑集成区 Web IDE 展开

新建发布单

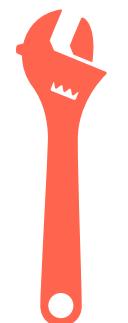
拉货就找货拉拉



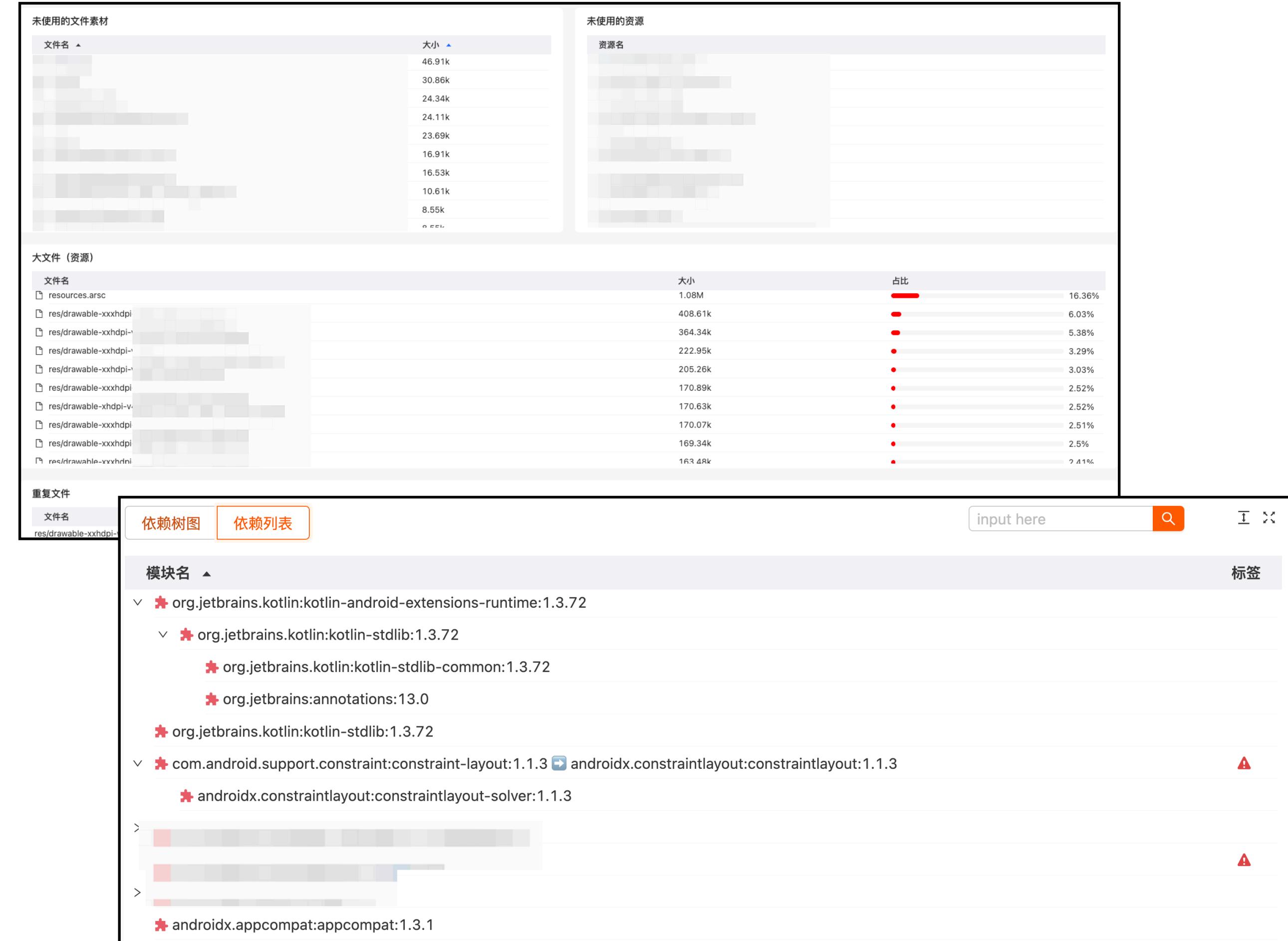
基于流水线的实践

第二步 扩充原子服务能力

工具型



- 加固
- 依赖图
- 包瘦身
- 构建耗时
- 上传AppStore



拉货就找 货拉拉



基于流水线的实践

静态分析型



- Infra 分析
- 安全扫描
- 隐私API检查
- TCA代码分析

安全扫描

深度分析

问题详细列表

风险等级 空指针引用 错误类型 错误

共 209 条 50条/页 1 2 3 4 5 >

规则描述

GPSReceiver.java onReceive object 'action' last assigned on line 15 could be null and is dereferenced at line 16.

行数 0 Bug追踪轨迹

Line 14 lib_third_party/src/main/java/com/alamove/huolala/lib_third_party/location/GPSReceiver.java start of procedure onReceive(...)

Line 15 lib_third_party/src/main/java/com/alamove/huolala/lib_third_party/location/GPSReceiver.java Skipping getAction(): unknown method

Line 16

OrderUtil.java genWebParamForShowCancelOrderNoStickerFee object returned by 'getInstance().getLoginConfigInfo()' could be null and is dereferenced at line 16.

OrderUtil.java isNativeShowCancelOrderNoStickerFee object returned by 'getInstance().getLoginConfigInfo()' could be null and is dereferenced at line 23.

SharedUtil.java getEditor object 'mPrefs' last assigned on line 33 could be null and is dereferenced at line 34.

TaskDialog.java onCreate object returned by 'getWindow()' could be null and is dereferenced at line 36.

AppConfig.java getDevVersionName object returned by 'getSignatureDate()' could be null and is dereferenced at line 50.

【Kotlin】基础规则包

基础

Kotlin官方基础规则包，提供常见代码问题扫描。

适用于 1 种语言 kotlin

查看详细规则 >

自定义规则 1 条

适用于 1 种语言 java

适用于 10 种语言 cpp、cs、Go、java、js、php、pytho...

查看详细规则 >

【Android】隐私合规检查

安全

检查Android项目中的隐私API使用是否合规。

适用于 1 种语言 java

查看详细规则 >

【Android】动态权限调用分析

安全

分析Android项目中的动态权限调用。

适用于 1 种语言 java

查看详细规则 >

【Android】危险权限分析

安全

分析Android项目中的危险权限使用。

适用于 1 种语言 java

查看详细规则 >

【Java】强化安全规则

安全

安全规则，需要申请license使用

适用于 1 种语言 java

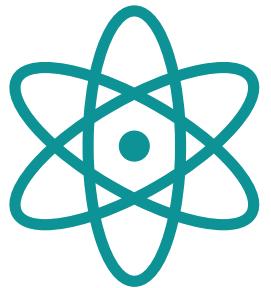
查看详细规则 >

拉货就找货拉拉



基于流水线的实践

动态分析型



- UI自动化测试
- 稳定性冒烟
- 启动时间
- 日志规范检查



拉货就找货拉拉



基于流水线的实践

第三步 基于三个阶段提供不同的流水线模板

添加构建模板

基础信息

* 名称: 请输入名称, 例如: debug包
* 英文名: 请输入英文名, 例如: debug-build

应用: 货拉拉司机端-Android
模板类型: 普通构建类型 发布类型
 补丁构建类型

描述: 请输入
* 生效范围: 项目 集成区

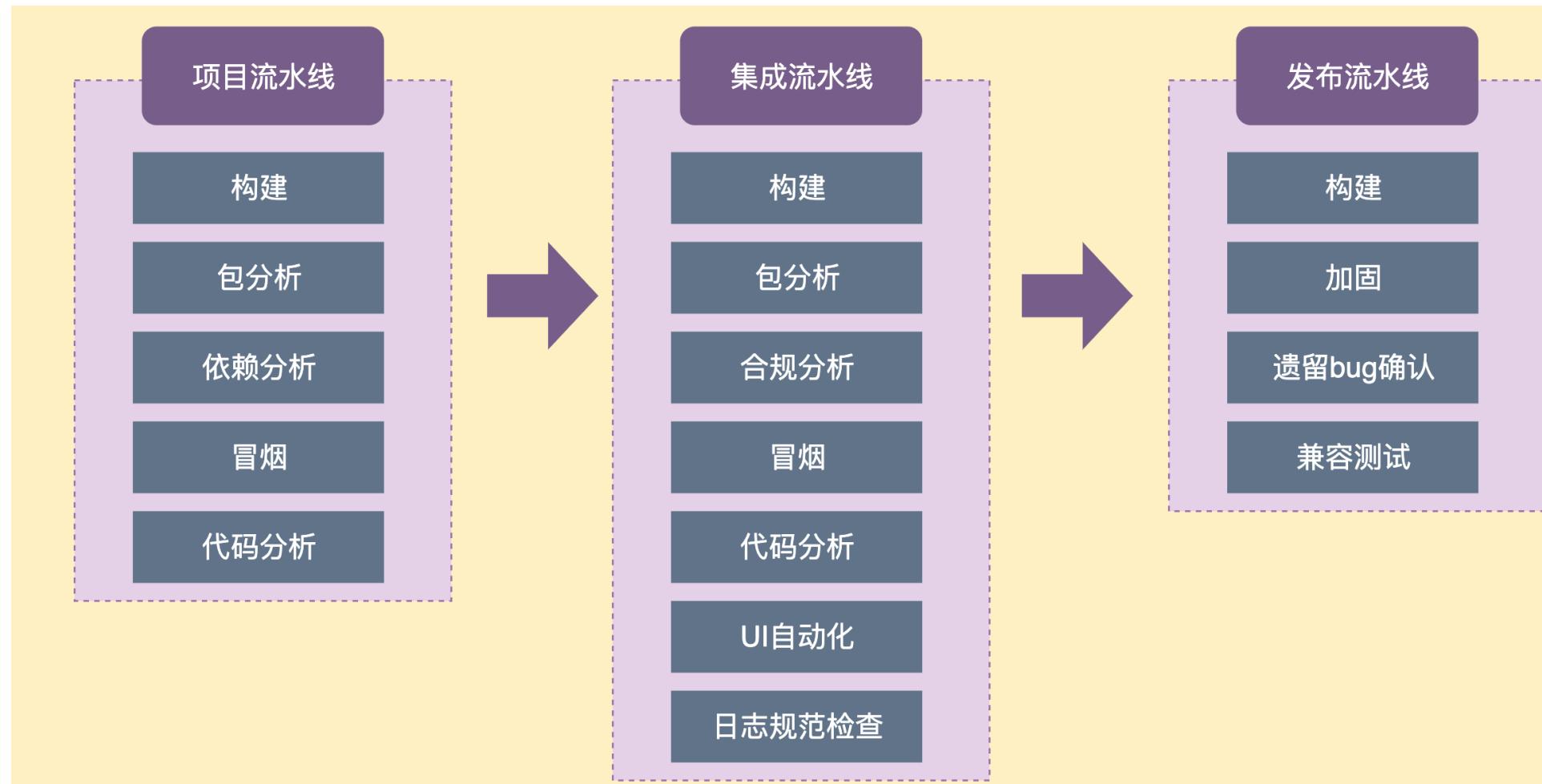
参数化构建

暂无参数

添加参数

取消 保存

发布模板名称	模板英文名	类型	生效范围	状态	操作
release发布	publish	发布类型	<input type="button" value="项目"/> <input type="button" value="集成区"/>	<input checked="" type="radio"/> 已启用	<input type="button" value="编辑模板信息"/> <input type="button" value="禁用"/> <input type="button" value="删除"/>
构建模板名称	模板英文名	类型	生效范围	状态	操作
debug包-stg	debug-build	普通构建类型	<input type="button" value="项目"/>	<input checked="" type="radio"/> 已启用	<input type="button" value="编辑模板信息"/> <input type="button" value="禁用"/> <input type="button" value="删除"/>
release包	release-package	普通构建类型	<input type="button" value="项目"/>	<input checked="" type="radio"/> 已启用	<input type="button" value="编辑模板信息"/> <input type="button" value="禁用"/> <input type="button" value="删除"/>
debug包-pre	debug-build-pre	普通构建类型	<input type="button" value="项目"/>	<input checked="" type="radio"/> 已启用	<input type="button" value="编辑模板信息"/> <input type="button" value="禁用"/> <input type="button" value="删除"/>
integration-release-ci	integration-release-ci	普通构建类型	<input type="button" value="集成区"/>	<input checked="" type="radio"/> 已启用	<input type="button" value="编辑模板信息"/> <input type="button" value="禁用"/> <input type="button" value="删除"/>
integration-debug-ci	integration-debug-ci	普通构建类型	<input type="button" value="集成区"/>	<input checked="" type="radio"/> 已启用	<input type="button" value="编辑模板信息"/> <input type="button" value="禁用"/> <input type="button" value="删除"/>



拉货就找货拉拉



基于流水线的实践

实践结果

The screenshot shows a detailed view of a CI/CD pipeline execution. At the top, it displays configuration details: '配置项模版: integration-debug-ci', '修改时间: 2022-11-16 10:59:13', '创建/修改: [redacted]', '构建分支: [redacted]', and a large orange '立即构建' button. Below this, the pipeline status is shown with several stages: '分支: [redacted]' (手工触发), 'commitId: be378a54', 'commitMsg: [redacted]' (触发时间: 2022-12-23 15:01:46), '备注: --', '流水线耗时: 1小时32分钟6秒', '构建产物: 品3', 'APP版本(BuildNo): 6.3.8(6308)', '代码分析: 查看'. The pipeline itself consists of several steps: 构建 (成功, 655毫秒), 5分钟冒烟 (成功, 排队时间: 9秒, 执行时间: 7分43秒), ApkChecker (成功, 排队时间: 12秒, 执行时间: 37秒), 核心场景UI回归 (成功, 执行时间: 35分6秒), 8小时稳定性测试 (成功, 执行时间: 31秒), 代码扫描 (成功, 排队时间: 8秒, 执行时间: 41分13秒). On the right, there's a thumbnail of the APK file and a download link: '安装包:d_stg_20221223...'. At the bottom, a table summarizes the build: 构建号 30220, commitId be378a54, 状态 流水线执行成功, 触发人 [redacted], 触发时间 2022-12-23 15:01:46, 构建耗时 7分钟8秒, 日志 日志, 产物信息 品3, 操作 提测 分析详情 查看 删除. A pagination bar at the bottom right indicates '总共1条 < 1 > 4条/页'.

自定义流水线 **100+**

提供了原子服务 **10+**

提供了流水线模板 **10+**

拉货就找 货拉拉

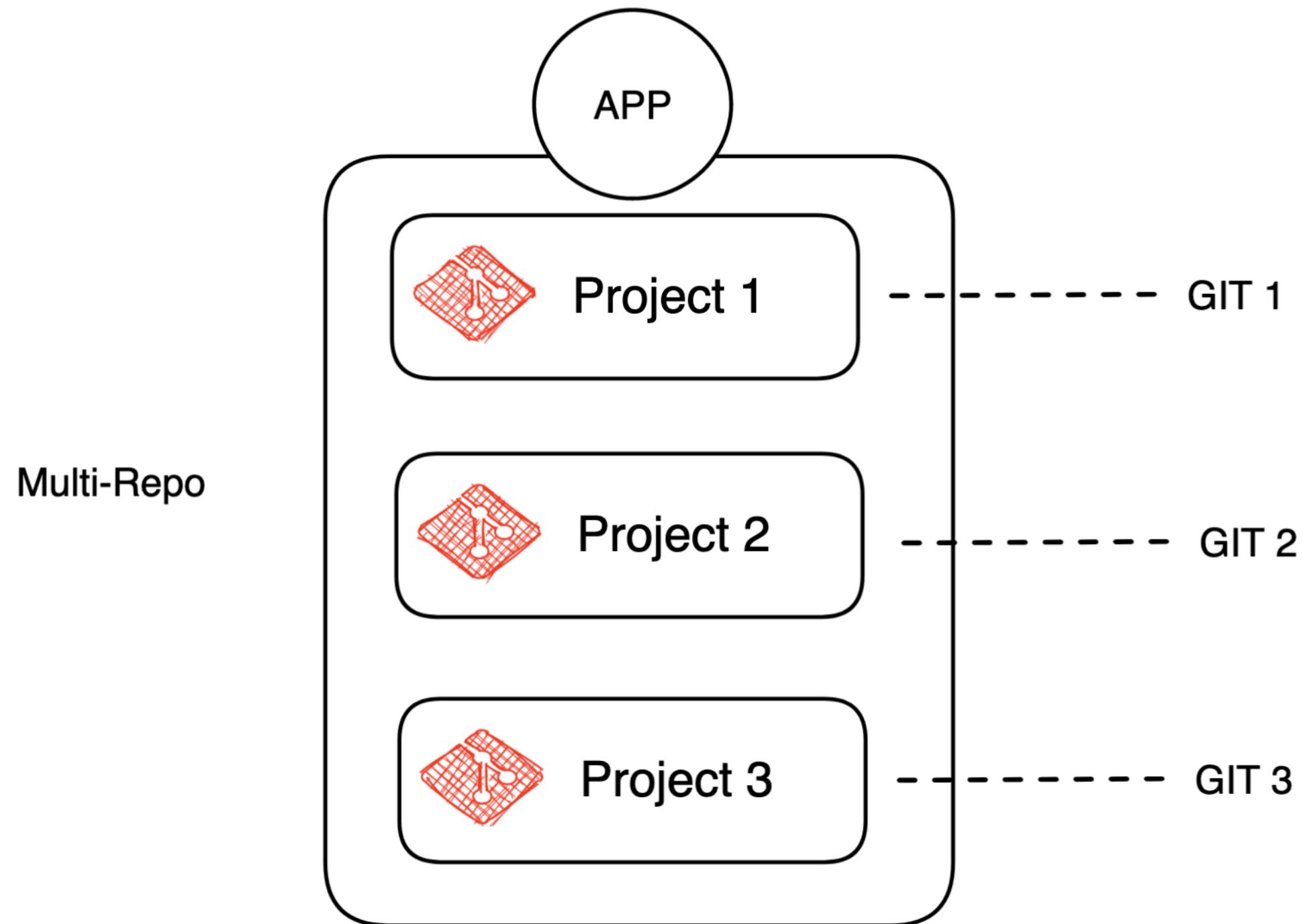




货拉拉移动端devops一些核心建设 — 多仓集成建设



多仓集成建设 – 背景



优点:

- 编译速度快
- 模块间相互独立，可以进行独立开发
- 代码安全性高

缺点:

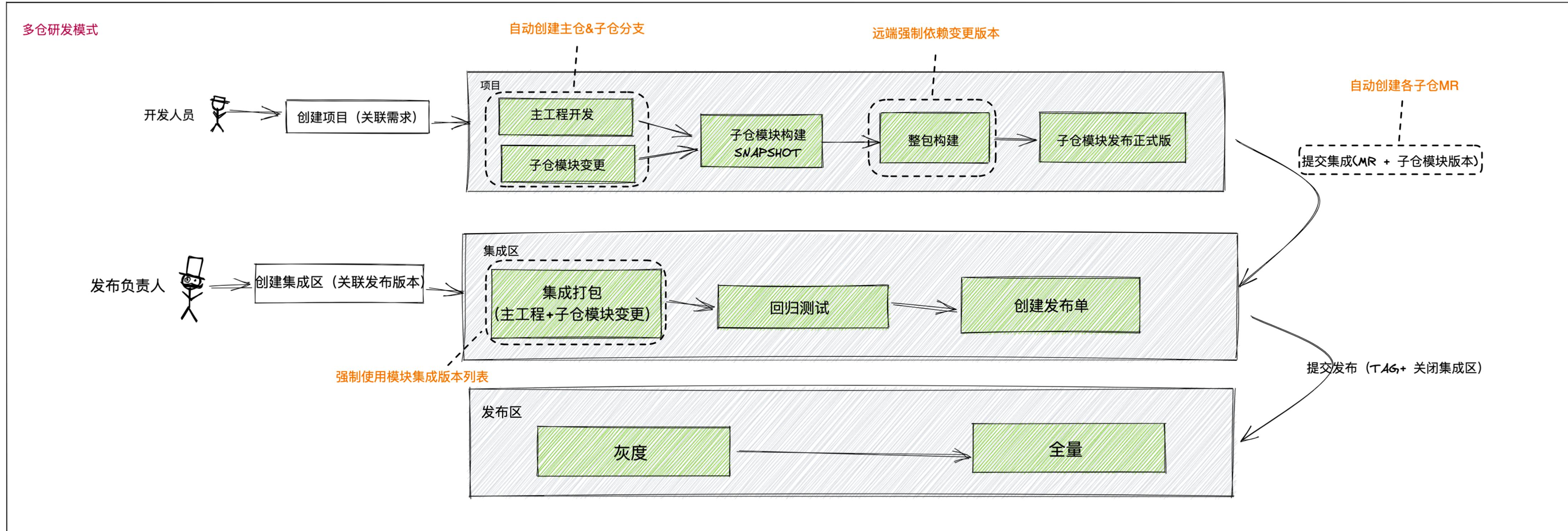
- 版本管理复杂
- 分支管理操作成本较高

拉货就找货拉拉



多仓集成建设

提供多仓研发模式的平台支持



引入一个新的问题：如何去管理以及发布我们的模块？

拉货就找货拉拉



多仓集成建设

模块平台管理

The screenshot shows a web-based module management interface. At the top left, it says '模块专区 / 模块详情'. On the right, there are buttons for '+ 新建' and a user profile. The main content area displays a module named 'Mdap更新SDK-Android 0.4.2'. It includes a rating of 1 star, a release date of 2022-10-18 17:37:10, and categories like 'Android', 'public', and '基础能力'. Below this are tabs for 'Readme', 'Changelog', 'Example', '依赖分析', '版本', '构建', and '构建配置'. The 'Readme' tab is currently selected. To the right of the tabs is a large, heavily blurred text area. On the far right, there are several sections: '基础信息' (with fields for groupId, artifactId, key, and outputType), 'Gitlab' (with a link), '模块描述' (with the text 'MDAP更新SDK'), '安装组件' (with code snippets for '# 引用' and 'implementation'), '成员' (with fields for 负责人 and 开发人员), '模块标签' (with a '基础能力' button), and '高级操作' (with a '删除' button). A red '编辑' button is located at the top right of each section.

管理 公有模块 和 私有模块

- 模块Readme
- 查看变更历史
- 引用依赖
- 版本管理
- 二进制发布

拉货就找货拉拉



多仓集成建设

多仓分支创建

设置项目分支

* 操作: 创建分支 选择已有分支

* 项目分支名: test * 基线分支: master

* 模块: hll-user * 基线分支: master

* 模块分支名: test * 基线分支: master

* 模块: hll-common * 基线分支: master

* 模块分支名: test * 基线分支: master

+ 新增模块分支创建

取消 确定

集成单创建

新建集成单

* 选择集成区: 【iOS】6.7.16版本集成区

* 集成原因: 正常集成需求

项目源工程

* 源分支: [redacted]

自动发起MR:

* 目标分支: v6.7.16-release 初审人: [请搜索选择用户]

* 终审人: [redacted]

依赖变更

* 集成方式: 版本依赖 2.0.8

取消 确定

审核通知



集成审核

集成详情--

v 2022-11-18 04:37:23 集成通过

开发人员: [redacted] 测试人员: [redacted]

提测状态: 已提测

源分支: feature/connor/addr... 目标分支: v6.7.4-release 已通过

初审人: [redacted] 终审人: [redacted]

集成方式: 分支依赖 集成分支: v6.7.4-release 查看变更

源分支: feature/connor/addr... 目标分支: v6.7.4-release 已通过

初审人: [redacted] 终审人: [redacted]

集成方式: 分支依赖 集成分支: v6.7.4-release 查看变更

源分支: feature/connor/addr... 目标分支: v6.7.4-release 已通过

初审人: [redacted] 终审人: [redacted]

集成方式: 分支依赖 集成分支: v6.7.4-release 查看变更

取消 确定

依赖管理

集成详情 整包构建 依赖管理 发布单

当前依赖列表

属性	模块名称	依赖KEY	version	更新时间	操作
私有属性	[redacted]	[redacted]	版本依赖 6.6.9 → 6.7.0	2022-11-27 09:52:33	变更历史
私有属性	[redacted]	[redacted]	版本依赖 6.6.7 → 6.7.0	2022-11-23 14:32:11	变更历史
私有属性	[redacted]	[redacted]	版本依赖 6.6.9 → 6.7.0	2022-11-27 10:02:59	变更历史
私有属性	[redacted]	[redacted]	版本依赖 6.6.9 → 6.7.0	2022-11-26 19:47:03	变更历史
私有属性	[redacted]	[redacted]	版本依赖 6.6.9 → 6.7.0	2022-11-24 17:11:26	变更历史

拉货就找 货拉拉





总结与展望



总结与展望

我的专区

欢迎使用
打包之余记得泡杯枸杞茶哦

正在进行项目 6个 已完成项目 58个 参与集成 214个 一共构建 717次

我的项目 新建项目

查看更多

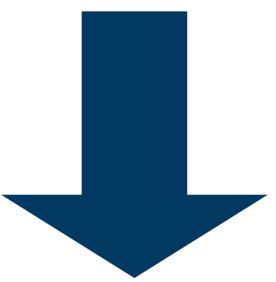
我的关注

项目 集成区 应用 产品线

应用名称	平台	操作
货拉拉Android	Android	★
搬家小哥iOS	iOS	★
搬家小哥Android	Android	★
货拉拉客户端-iOS (AppStore)	iOS	★
货拉拉司机端-Android	Android	★
货拉拉专送司机端-Android	Android	★

查看更多

从 0 到 1



从 1 到 10

拉货就找货拉拉



总结与展望

工作流升级

原子能力扩充

智能扭转

调度引擎升级

专注线下流程

本地构建监控

人工流程梳理

IDE 工具

产物健康度

持续分析/测试

可量化

分配到人

持续发布

灰度监控

智能止损

智能下发

拉货就找货拉拉



拉货就找 货拉拉

Q & A



拉货就找 货拉拉



扫码加入官方微信群，与讲师继续交流技术

