



KubeCon



CloudNativeCon

China 2018

# Kubernetes Native DevOps Practice

— 王磊 @TenxCloud



- Our DevOps Expectations
- Kubernetes Capabilities/Advantages to Build DevOps Solution
- Architecture and Features
  - CRD and operator design
  - Pipeline / Stage/ Task / Task Template / Version Control
  - Logging, monitoring, autoscaling, high availability
  - Extensibility / Integration
  - CI/CD examples
- Future plan

- Build a platform and easy to integrate with other DevOps/third-party tools
- Easy to be customized as user requirements are diverse
- Easy to setup, maintain, extend and scale
- Reduce the learning curve for customer and ourselves
- Get consistent user experience and data, leverage with PaaS capability
- Facilitate our PaaS and micro-service product

**WHAT  
DO  
YOU  
EXPECT**



# Kubernetes Capabilities/Advantages to Build DevOps Solution

- k8s itself is NOT a PaaS or DevOps platform, but ...
- k8s resources that can be used to build DevOps solution

Pod



Job



CronJob



Volumes



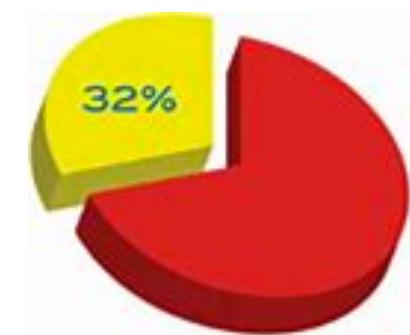
ConfigMap



Secret



ResourceQuota / LimitRanges



- Scheduler / Affinity
- And more ...

## Pod Spec

|                          |   |   |
|--------------------------|---|---|
| [] InitContainers        |    | Initialize the build environment  |
| Affinity / SchedulerName |    | Configure scheduler policy  |
| [] Volumes               |  | Share files between containers, or cache build files                                  |
| ActiveDeadlineSeconds    |  | Timeout of build task   |
| [] Containers            |  | Build tasks and the dependent environments (sidecar)                                  |
| Container                |   |   |
| Image                    | -   | Image of build / dependent environment  |
| [] Command               | -   | Command to execute  |
| [] Args                  | -   | Argument  |
| [] Env                   | -   | Environment variable  |
|                          |   |  |
| VolumeMounts             | -   | Files to be shared or persisted   |
| [] Resources             | -   | Resource requirement  |
| Lifecycle                | -   | Actions defined for postStart/preStop   |

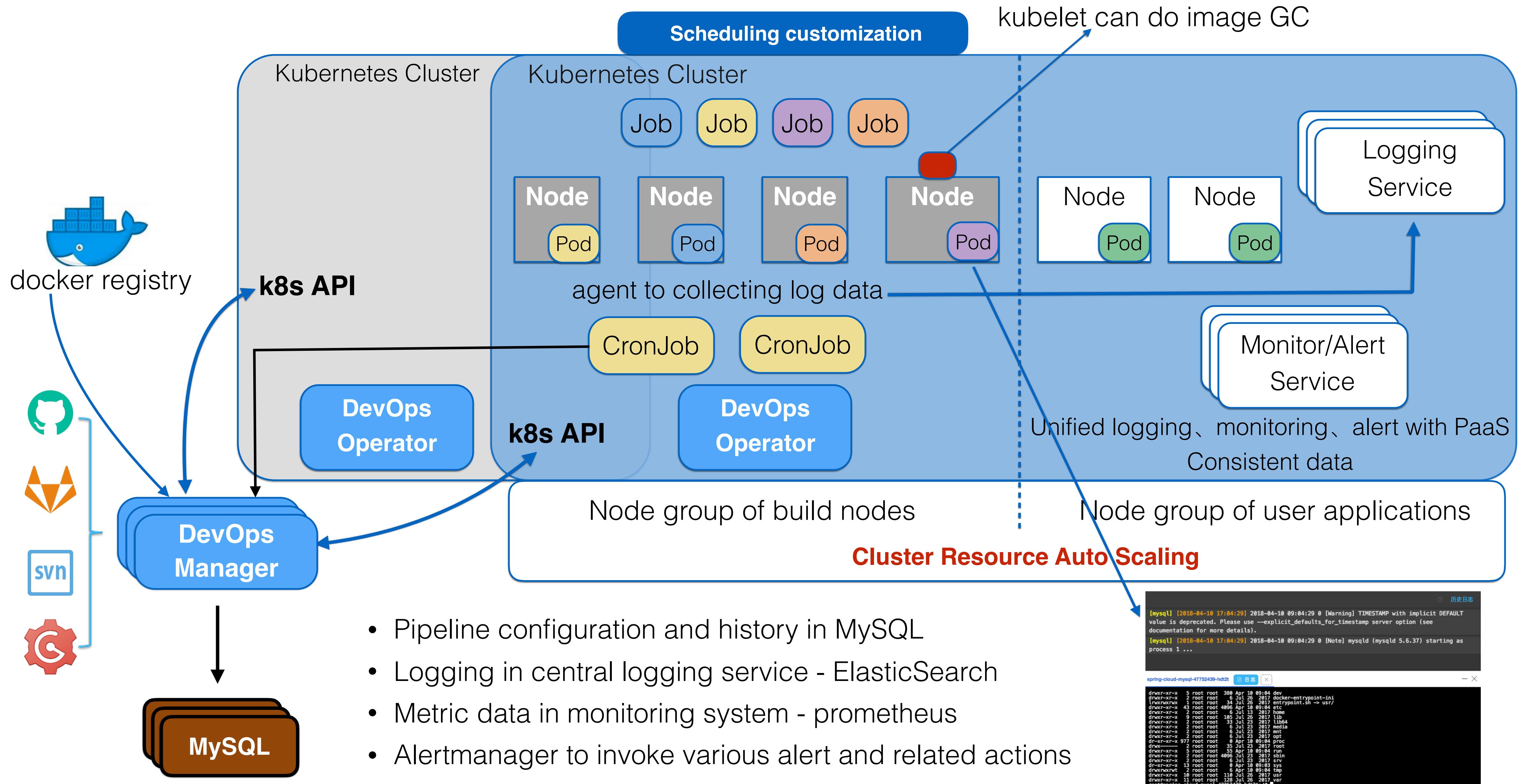
# Kubernetes Capabilities/Advantages to Build DevOps Solution



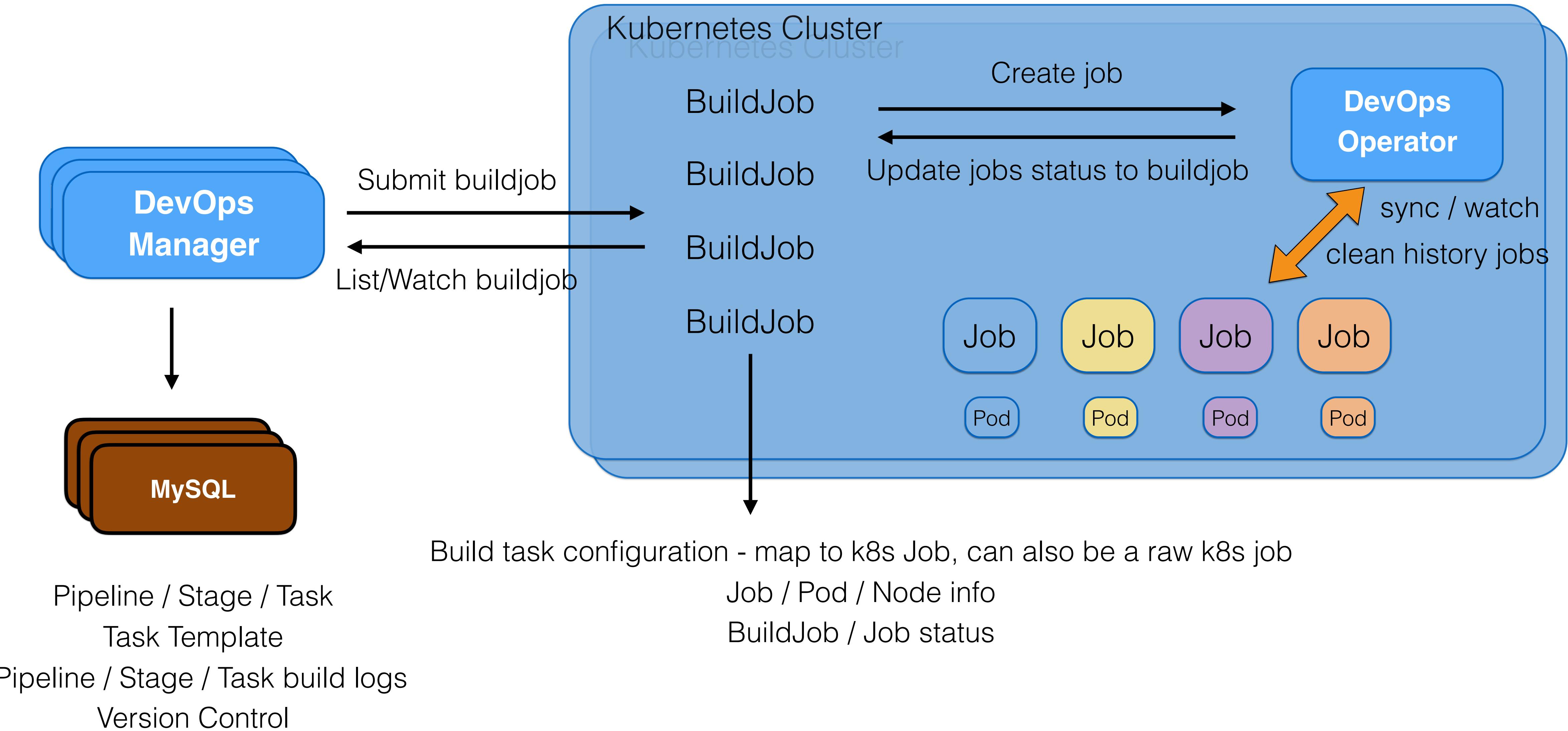
|                     |                            |  |
|---------------------|----------------------------|--|
| <b>Job Spec</b>     | parallelism                | Expected maximum number of parallel build tasks                      |
|                     | completions                | Expected number of completed build tasks                             |
|                     | activeDeadlineSeconds      | The timeout of the running build tasks, default: 12 hours            |
|                     | backoffLimit               | Maximum retry count before mark the build task as failed, default: 6 |
|                     | ttlSecondsAfterFinished    | Time to clean up finished build tasks after it finishes              |
|                     | Pod Template               |  |
| <b>CronJob Spec</b> | schedule                   | Cron style scheduler configuration                                   |
|                     | concurrencyPolicy          | Concurrency policy of CronJob  |
|                     | suspend                    | Whether suspend latter jobs if the previous job is still running     |
|                     | successfulJobsHistoryLimit |  |
|                     | failedJobsHistoryLimit     | Number of successful/failed history jobs to keep                     |
|                     | Job Template               |  |

- Our DevOps Expectations
- Kubernetes Capabilities and Advantages to Build DevOps Solution
- **Architecture and Features**
  - CRD and operator design
  - Pipeline/Stage/Task/Task Template/Version Control/UI generation/Volume...
  - Logging, monitoring, autoscaling, high availability
  - Extensibility/Integration
  - CI/CD examples
- Future plan

# Overall Architecture

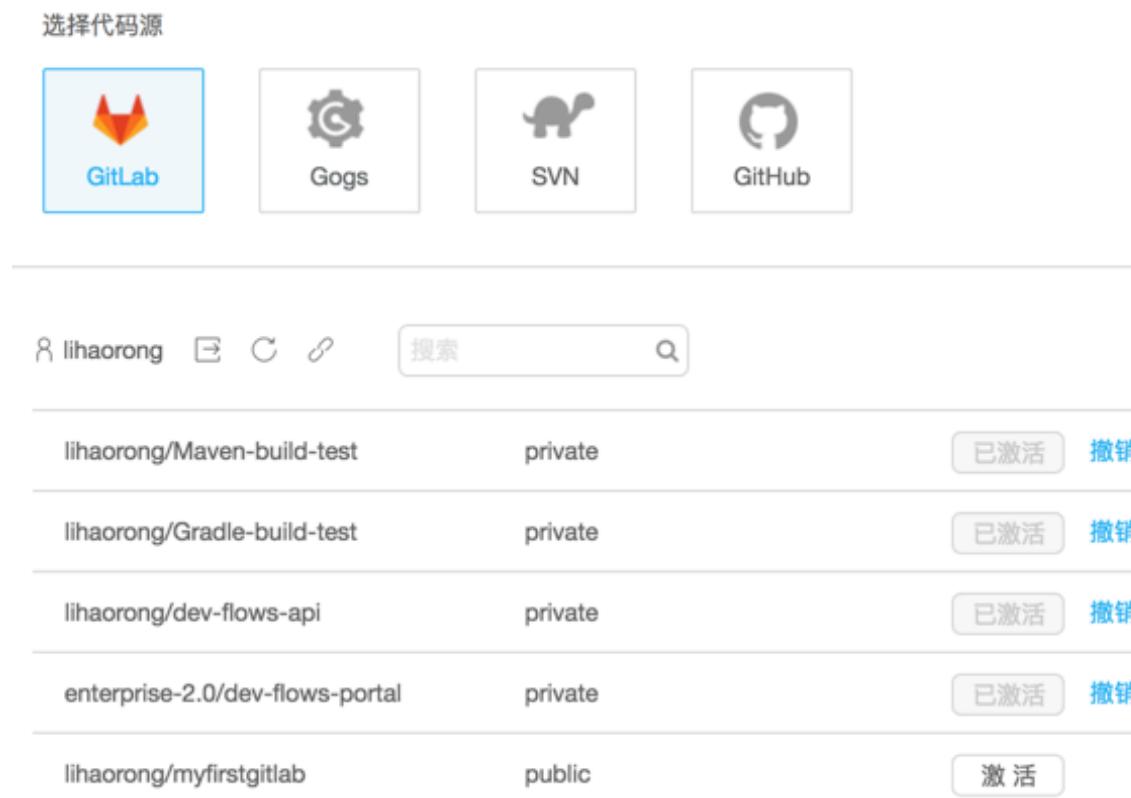


# CRD and Operator Design



# Basic Concepts (partial)

## Repository



选择代码源

- GitLab
- Gogs
- SVN
- GitHub

lihaorong

搜索

| 项目                              | 类型      | 状态  | 操作 |
|---------------------------------|---------|-----|----|
| lihaorong/Maven-build-test      | private | 已激活 | 撤销 |
| lihaorong/Gradle-build-test     | private | 已激活 | 撤销 |
| lihaorong/dev-flows-api         | private | 已激活 | 撤销 |
| enterprise-2.0/dev-flows-portal | private | 已激活 | 撤销 |
| lihaorong/myfirstgitlab         | public  | 激活  |    |

## Managed Project



| 名称                  | ID             | 属性      | 代码源  | 操作   |
|---------------------|----------------|---------|--|------|
| svn                 | MPID-HykHg06o7 | private | http://192.168.2.19:3343/csvn                      | 解除激活 |
| yihoyoung/test      | MPID-SJZwYhajm | public  | http://localhost:3000/yihoyoung/test.git           | 解除激活 |
| yihoyoung/1024      | MPID-rJNMa2Lj7 | public  | https://github.com/yihoyoung/1024.git              | 解除激活 |
| enterprise-2.0/d... | MPID-H1Vz-gKKX | private | git@gitlab.tenxcloud.com:enterprise-2.0/dev-flo... | 解除激活 |

## Data Volume



独享型存储 共享型存储

缓存卷：为流水线的任务提供缓存功能，用户可以将执行任务常用的依赖保存在缓存卷内，避免浪费网络资源，并显示读写操作

刷新 删除 搜索

| 缓存卷   | 状态  | 类型            | 大小   | 格式   | 任务 | 执行环境    |
|-------|-----|---------------|------|------|----|---------|
| test1 | 未绑定 | 块存储 (ceph886) | 512M | ext4 | -  | DevTest |

PVC



执行记录 流水线定义 Yaml 统计记录 设置

CI已开启 gitlab

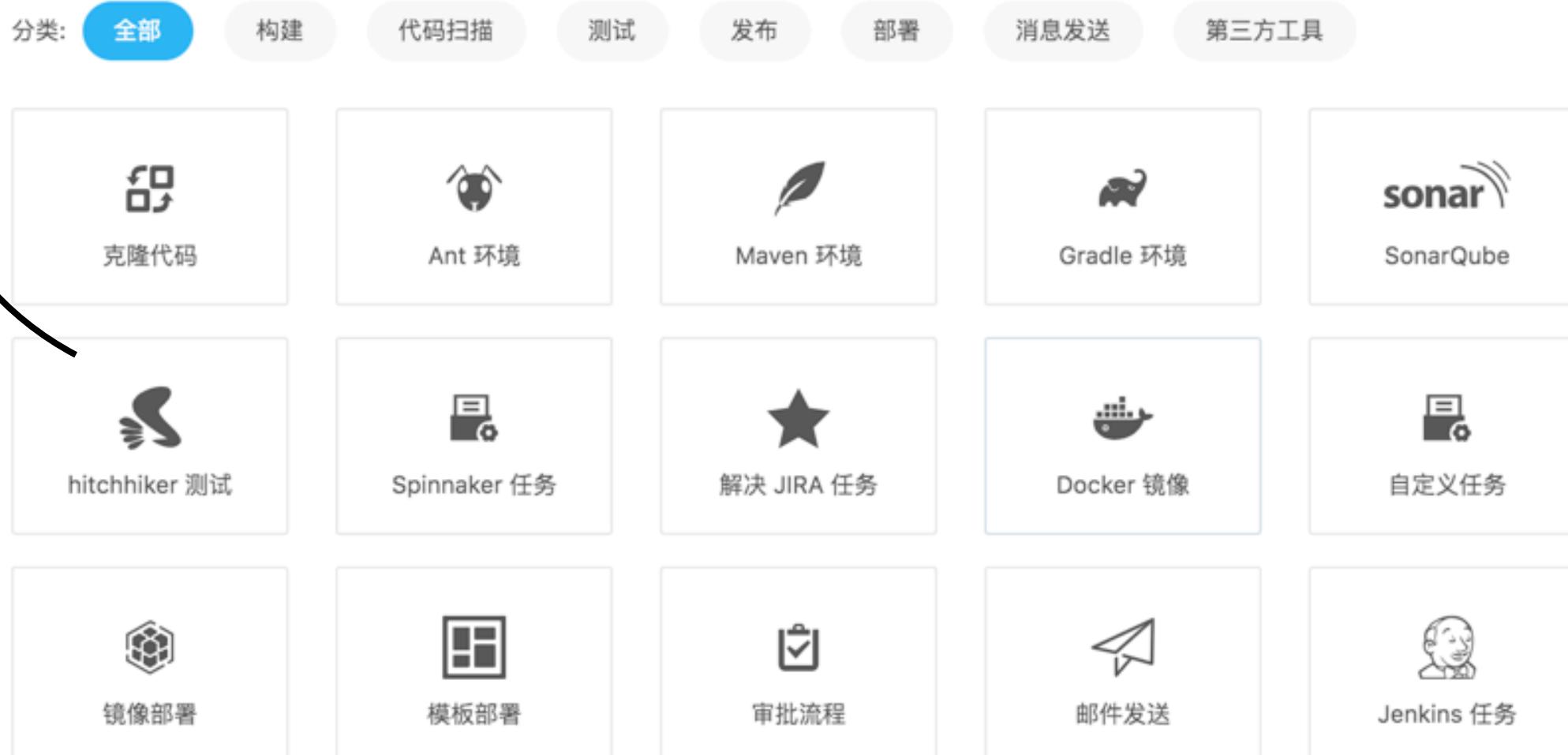
任务示例：

- 代码分析、测试
- 编译构建
- 第三方测试
- 部署交付

Tasks in same stage can run sequentially or in parallel

Task created from template

Task Template



分类: 全部 构建 代码扫描 测试 发布 部署 消息发送 第三方工具

|               |              |            |           |            |
|---------------|--------------|------------|-----------|------------|
| 克隆代码          | Ant 环境       | Maven 环境   | Gradle 环境 | SonarQube  |
| hitchhiker 测试 | Spinnaker 任务 | 解决 JIRA 任务 | Docker 镜像 | 自定义任务      |
| 镜像部署          | 模板部署         | 审批流程       | 邮件发送      | Jenkins 任务 |

Dockerfile / Scripts  
Common Configuration



普通配置 加密配置

「(lihaorong-test4)lihaorong-test」项目在 DevTest 中「服务配置」

配置分类 配置组

请输入分类名搜索

+ 创建配置组

未分类配置组 (0)

ConfigMap/Secret

# Logging, Monitoring, Autoscaling, High Availability

vCenter openstack

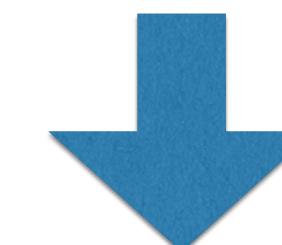


remove / add nodes

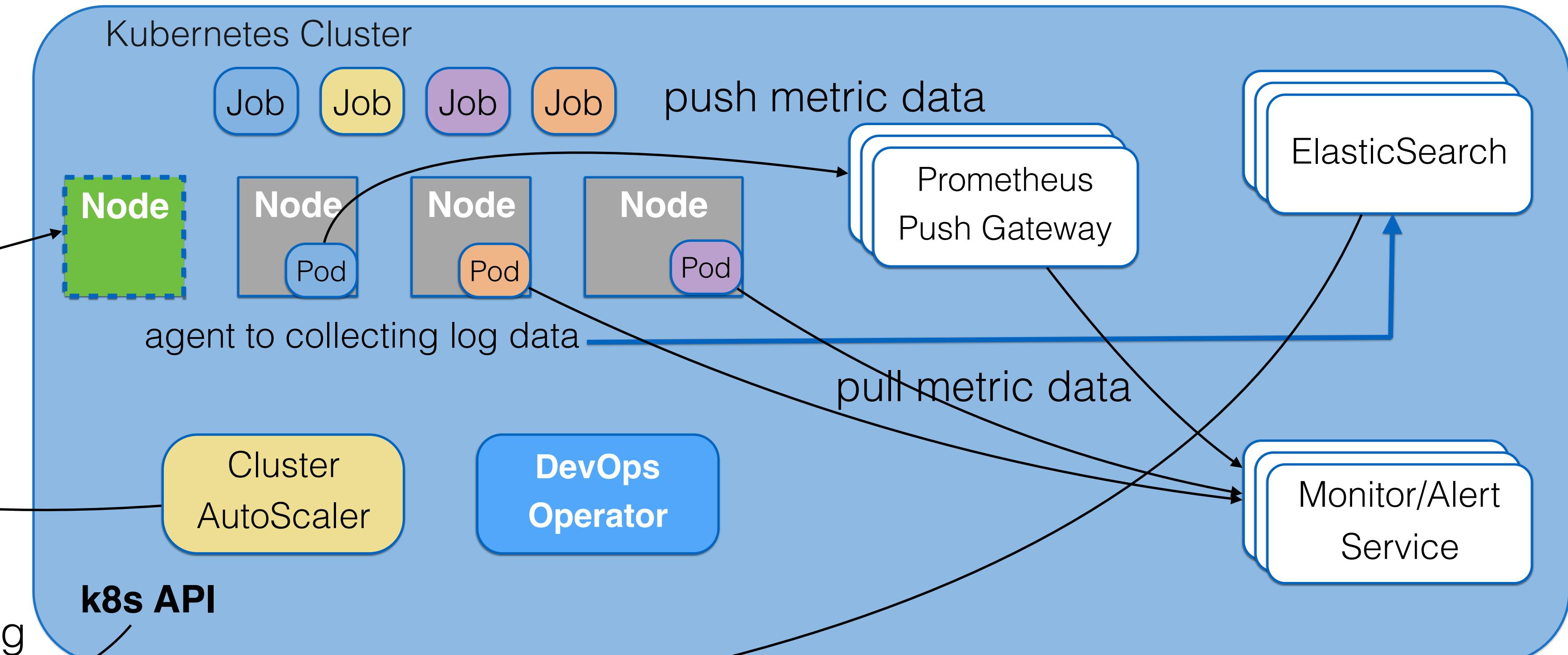
Infrastructure Layer

Cloud provider

insufficient resource



Restful API

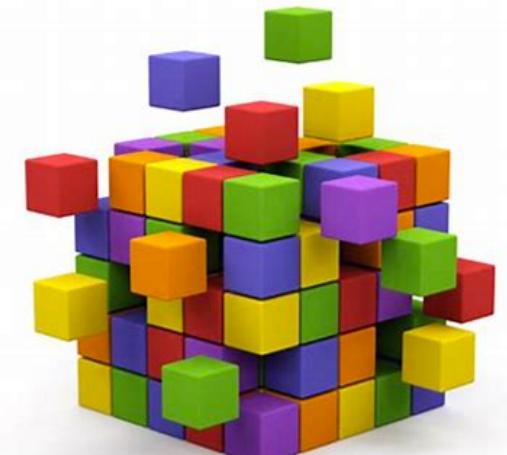


- Build task can also expose custom metric data
- Ephemeral build task can push metric to gateway if needed
- Cluster autoscaler will add/remove node from build group for scaling
- HA is guaranteed by cluster HA, k8s Job controller and cluster autoscaler, can also use container probe if needed

- Easy to extend task template, just docker images
- Use can define raw k8s job, more features can be enabled
- Leverage all k8s resources and extensibility to fulfill various requirements, prompt innovation
- New features of k8s may help enhance or improve DevOps



- Help integration test - use sidecar container as dependent environment
- Encapsulate API / SDK of other tools using image for better integration/collaboration
- Leverage k8s integration capabilities, such as external service registration, service catalog, integrate using CRD and controller, etc...



# CI/CD Examples - Build Docker Image

发布 | Docker 镜像

\* 任务名称 build image

使用流水线默认代码源 (若取消勾选, 当前任务将不再克隆默认代码)  
① 当前任务容器的 /app 目录

Dockerfile  使用云端 Dockerfile  
  
/ ① 系统默认构建目录为 /app, 制品可通过缓存卷挂载为子目录, 配合 Dc

构建缓存  关

发布镜像

\* 仓库组类型  镜像仓库组  自定义仓库  
http://192.168.1.52  
集群: DevTest  
cicd

\* 镜像名称 booking-service

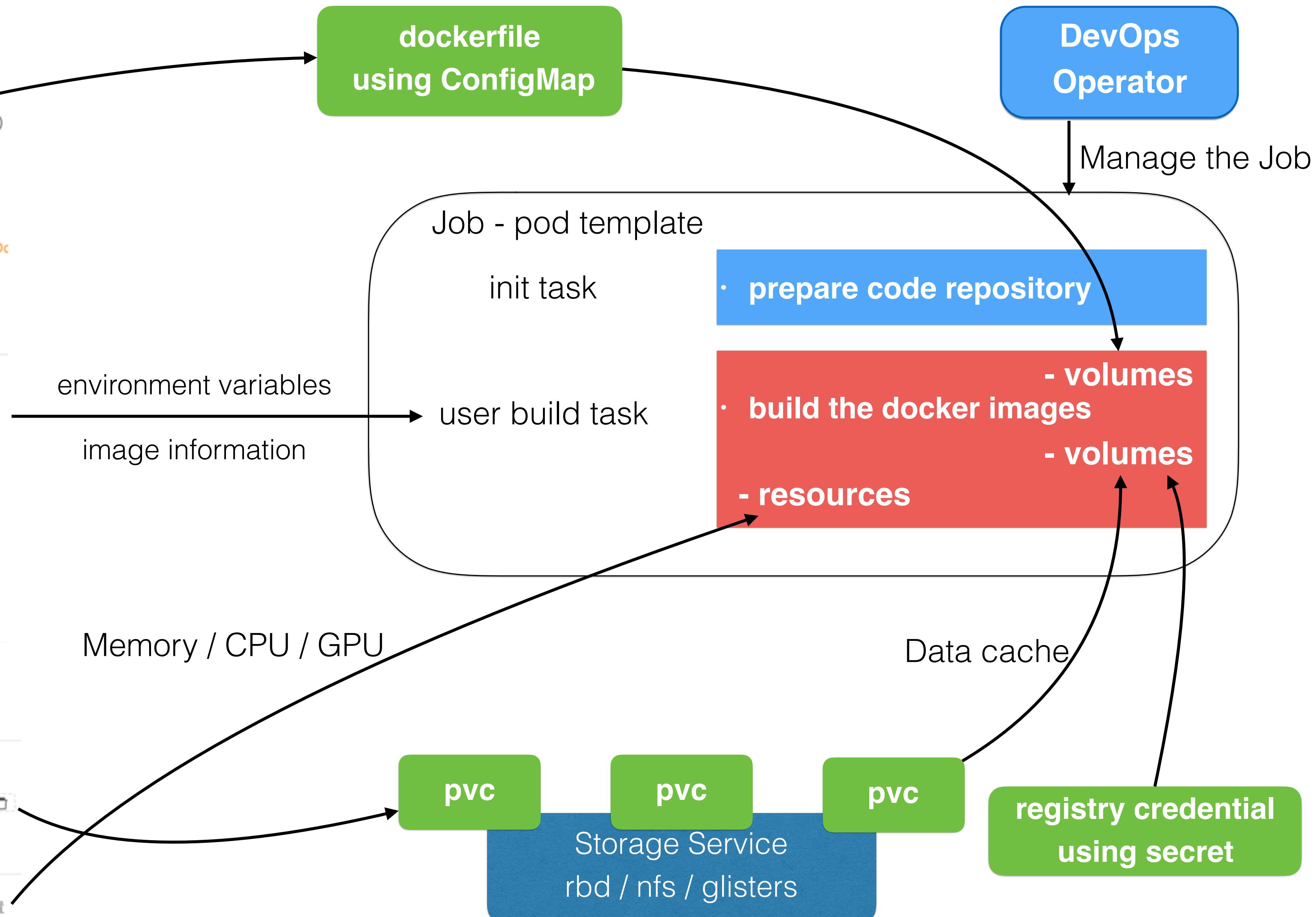
\* 镜像版本  以代码分支名为 tag  时间戳为 tag  自定义 tag  
选择构建生成的 Docker 镜像的 tag 命名规范, 支持以上三种命名规则

\* 执行环境 DevTest

允许失败  关 ① 该任务失败, 将不影响流水线继续执行

缓存卷  开  
独享型 (rbd) volume1 /root/.m2   
+添加一个容器目录

自定义配置  关 ① 关闭状态时, 代表不限制资源; 开启状态时, 代表可自定义资源配置



# CI/CD Examples - Artifact Management

执行 CMD 脚本文件 上传文件或直接输入文件内容 (仅支持上传 .sh .py .rb .php 格式文件, 建议文件大小 2M)  
脚本执行路径/app

上传脚本文件

Shell Python Ruby PHP

shell (读写)

```
1 #!/bin/sh
2 # CMD 脚本执行的工作目录为 pipeline 基础镜像的文件系统目录, 包含 clone
3 mvn clean package -dSkipTests -dSkipDocker
4
```

添加一个参数

人工制品

提取人工制品 仅支持提取构建容器/app目录下内容

/app/\* jar X

添加提取

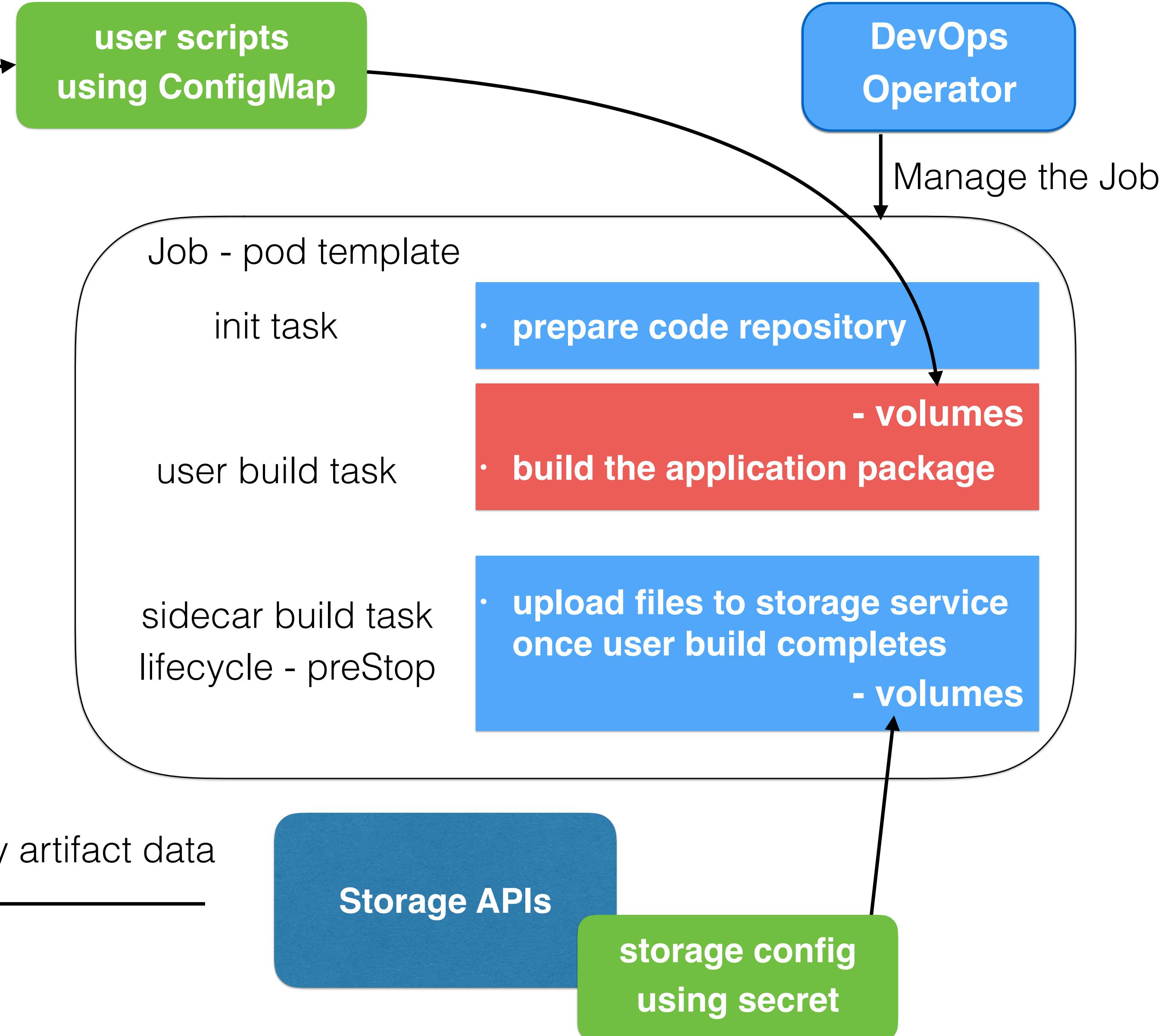
执行记录 流水线定义 文件管理 Yaml 统计记录 设置

请输入版本搜索

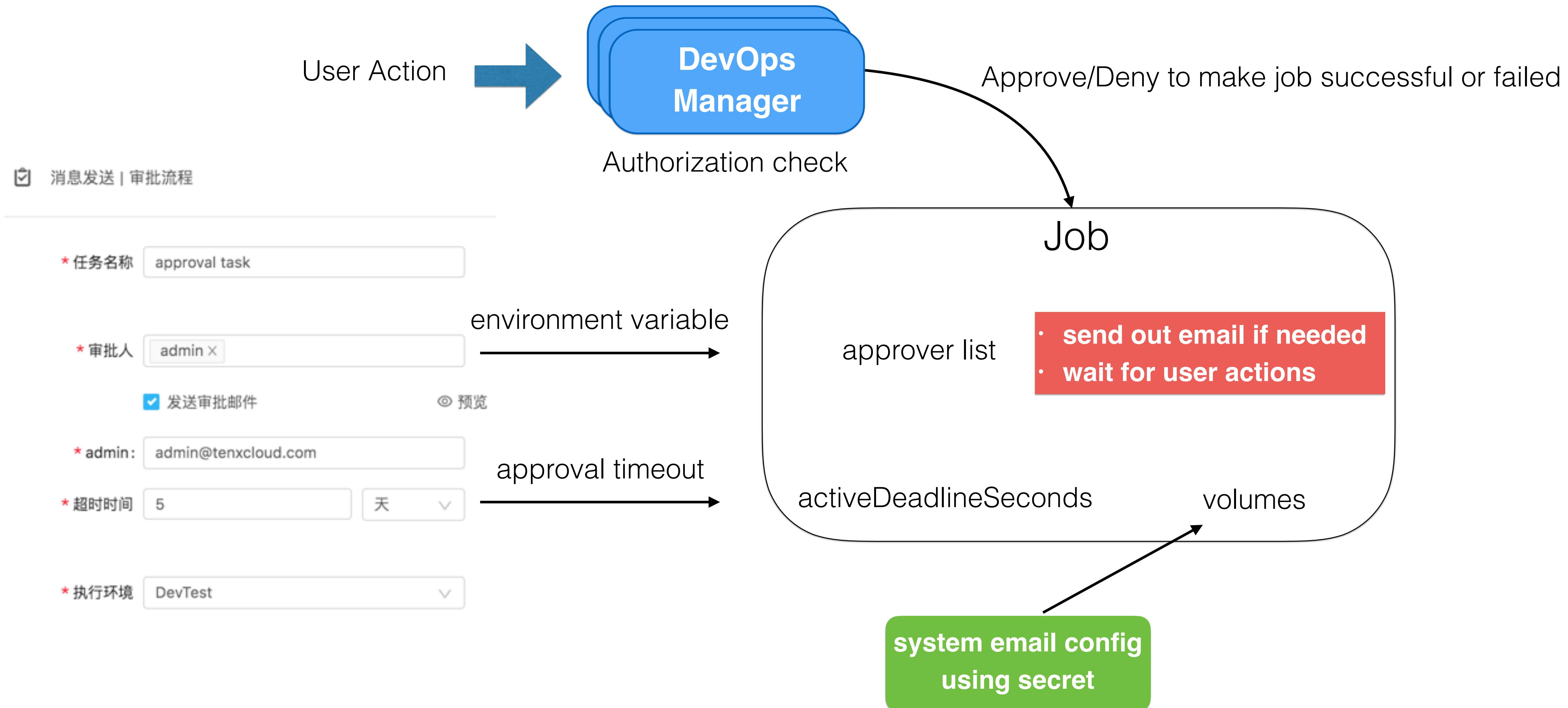
C 刷新

共计3条 < 1 >

| <input type="checkbox"/> | 名称               | 更新时间                | 大小  | 下载数(次) | 操作  |
|--------------------------|------------------|---------------------|-----|--------|---|
| <input type="checkbox"/> | store-v1.0.jar   | 2018-10-27 10:39:20 | 16B | 0      | <input type="button" value="下载"/> <input type="button" value="删除"/> |
| <input type="checkbox"/> | booking-v1.0.jar | 2018-10-27 10:39:20 | 16B | 0      | <input type="button" value="下载"/> <input type="button" value="删除"/> |
| <input type="checkbox"/> | billing-v1.0.jar | 2018-10-27 10:39:20 | 16B | 0      | <input type="button" value="下载"/> <input type="button" value="删除"/> |



# CI/CD Examples - Human/Manual Task



# CI/CD Examples - Continuous Deployment

Upgrade an existing service

\* 任务名称 滚动升级

\* 选择 Docker 镜像任务 (1) 发布阶段 / nginx镜像

镜像名称: 192.168.1.52/public/nginx

\* 要部署的服务 DevTest / nginx

普通升级  滚动升级  滚动升级采用RS扩容方式, 不会中断服务

Deploy using helm template

\* 任务名称 deploy template

\* 应用模板 mynginx

\* 选择集群 DevTest

\* 应用名称 myapp

新建应用

nginx 使用模板镜像

镜像名称: 192.168.1.52/public/nginx:20181016.094004.86

普通升级  滚动升级  滚动升级采用RS扩容方式, 不会中断服务

Invoke Spinnaker web hook of specified deployment pipeline

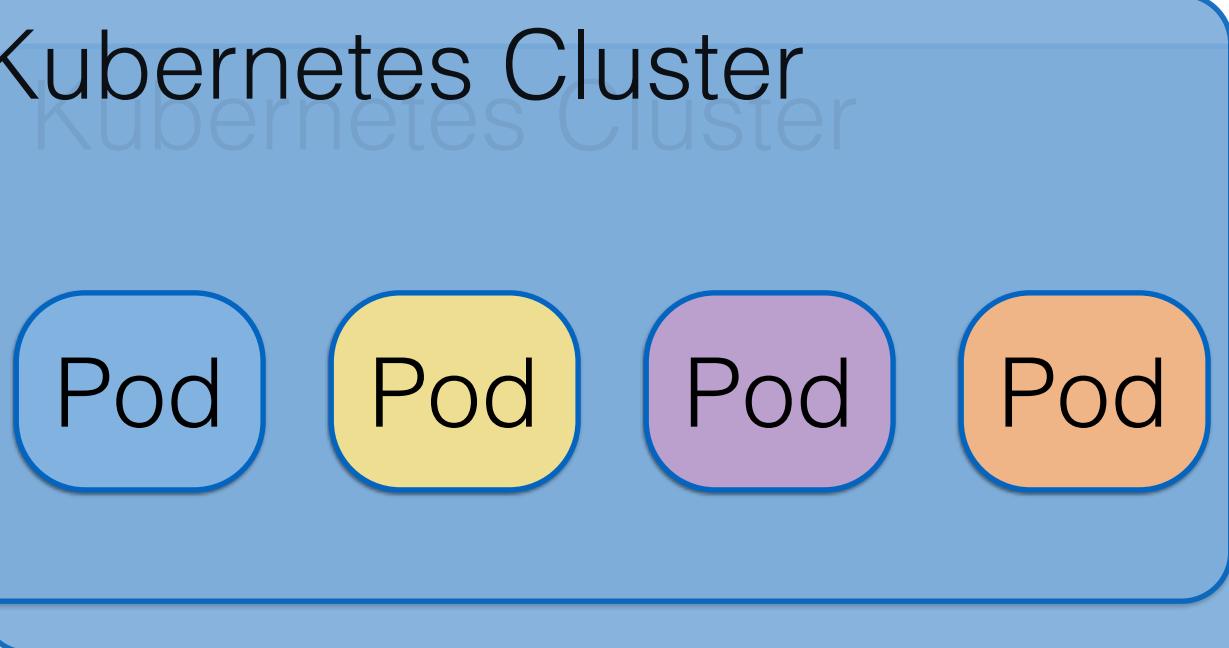
environment variables  
→  
deployment strategy

serviceaccount

Security policy for the build task

Job

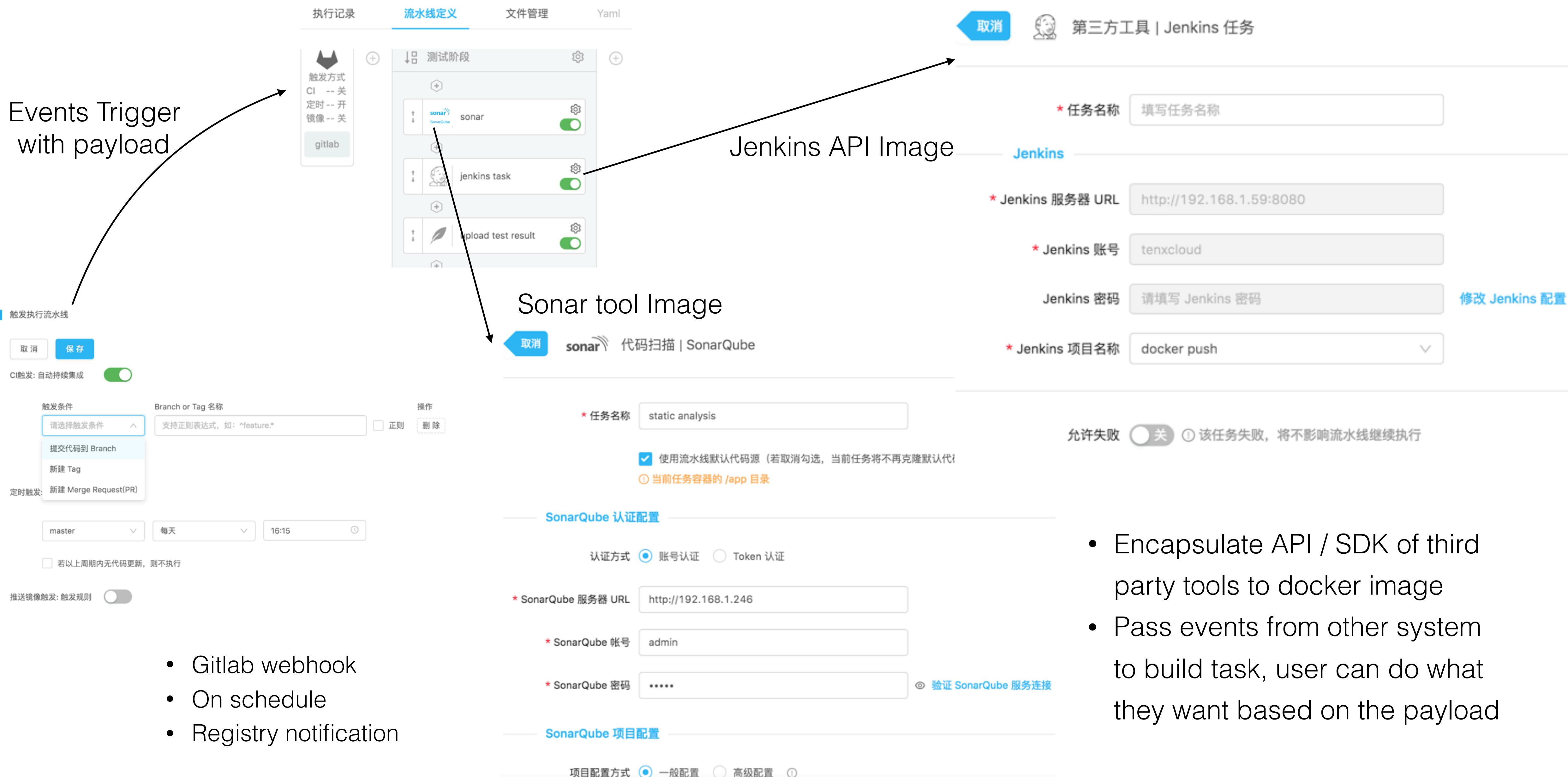
- Image with kubectl or call k8s API
- Image with helm client
- Call Spinnaker webhook



deployment pipeline



# CI/CD Examples - Gitlab/Sonar/Jenkins Integration



The diagram illustrates the CI/CD integration flow between three tools:

- Gitlab:** Triggered by events (Events Trigger with payload) via webhook or scheduled triggers.
- SonarQube:** Triggered by Gitlab events, specifically "sonar" events.
- Jenkins:** Triggered by SonarQube events via Jenkins API Image.

**Gitlab Pipeline Configuration:**

- Triggered by events (Events Trigger with payload).
- Triggers a "sonar" task.
- Triggers a "jenkins task".
- Triggers an "upload test result" task.

**SonarQube Configuration:**

- 任务名称: static analysis
- 使用流水线默认代码源 (若取消勾选, 当前任务将不再克隆默认代码源)
- 当前任务容器的 /app 目录

**Jenkins Task Configuration:**

- 任务名称: 填写任务名称
- Jenkins 服务器 URL: http://192.168.1.59:8080
- Jenkins 账号: tenxcloud
- Jenkins 密码: 请填写 Jenkins 密码
- Jenkins 项目名称: docker push
- 允许失败: 关 (该任务失败, 将不影响流水线继续执行)

**Summary:**

- Encapsulate API / SDK of third party tools to docker image
- Pass events from other system to build task, user can do what they want based on the payload

- Gitlab webhook
- On schedule
- Registry notification

# CI/CD Examples - Gitlab/Harbor/Jira Integration

Event payload can be passed to build task if needed



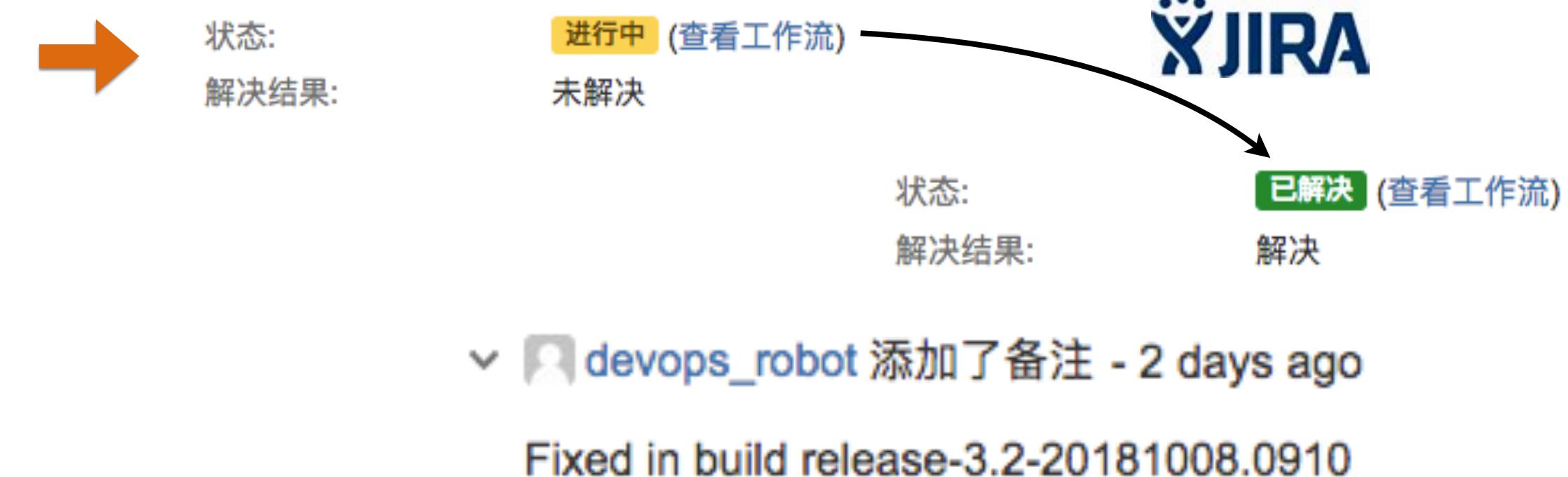
- Call harbor API to add/remove label of image, can also lock the passed image

| 操作          | 版本  | 推送时间                | 标签 |
|-------------|-----|---------------------|----|
| Test Passed | 2.0 | 2018-11-13 14:14:19 |    |
| No Test     | 1.0 | 2018-11-08 16:38:58 |    |

- Get the commits between two builds
- Invoke Jira API to mark the status and add the build info

```
git log --pretty=oneline c5eff7ea..3211901e
3211901e9b877c92ab059a6f25180469dcf1629 Merge branch 'dev-branch' into 'dev-branch'
5d6cbb9d09a8e8a34cad8a27df036c07f3b4c291 [KK-134] Fix xxx
b10192764026df3a1eac57d69019032eeb722ceb [LOT-3372] Fix xxx
7797990da64929112a90e9f6839bd3420e64c99b [KK-9] Fix xxx
7797990da64929112a90e9f6839bd3420e64c99b [LOT-3213] Fix xxx
96ce85fdecd50aaafafca2eae6a2a1fe4b1aef72d Merge branch 'LOT-3033' into 'dev-branch'
```

git commit should have naming convention



- Our DevOps Expectations
- Kubernetes Capabilities and Advantages to Build DevOps Solution
- Architecture and Features
  - CRD and operator design
  - Pipeline/Stage/Task/Task Template/Version Control/UI generation/Volume...
  - Logging, monitoring, autoscaling, high availability
  - Extensibility/Integration
  - CI/CD examples
- **Future plan**

# Our Future Plan

- More task templates to be added, integrate more CI/CD and project management tools
- Optimize UI generation methodology
- Improve development experience, such as CLI, plugin for IDE, dev on Cloud
- Move forward to better DevOps under micro-service architecture
- Consolidate the consistent events and data, and create k8s native ChatOps tool
- Contribute to open source community



***Automation***



***Integration***



***Collaboration***

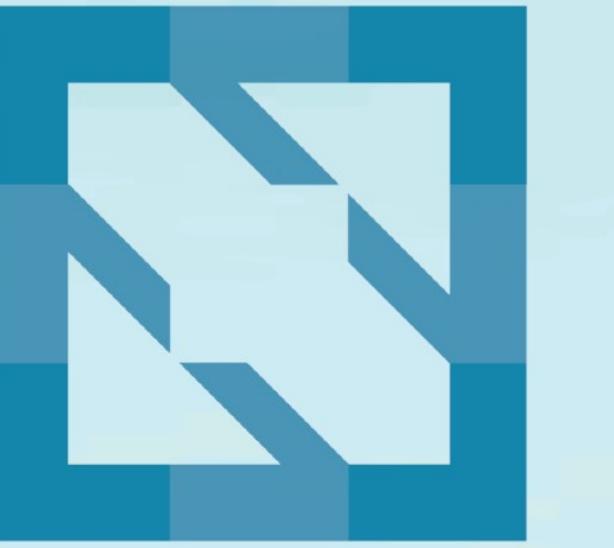


***Quantification***





KubeCon



CloudNativeCon

China 2018

Thank you !

