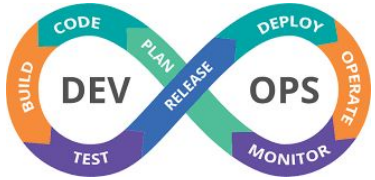
基于 Tekton 的 CICD 平台

chenshaowen @ 2021-07-31

个人介绍

- 陈少文
- www.chenshaowen.com
- 目前在金山办公
- 主要负责 CICD、PaaS 平台研发，云原生技术落地

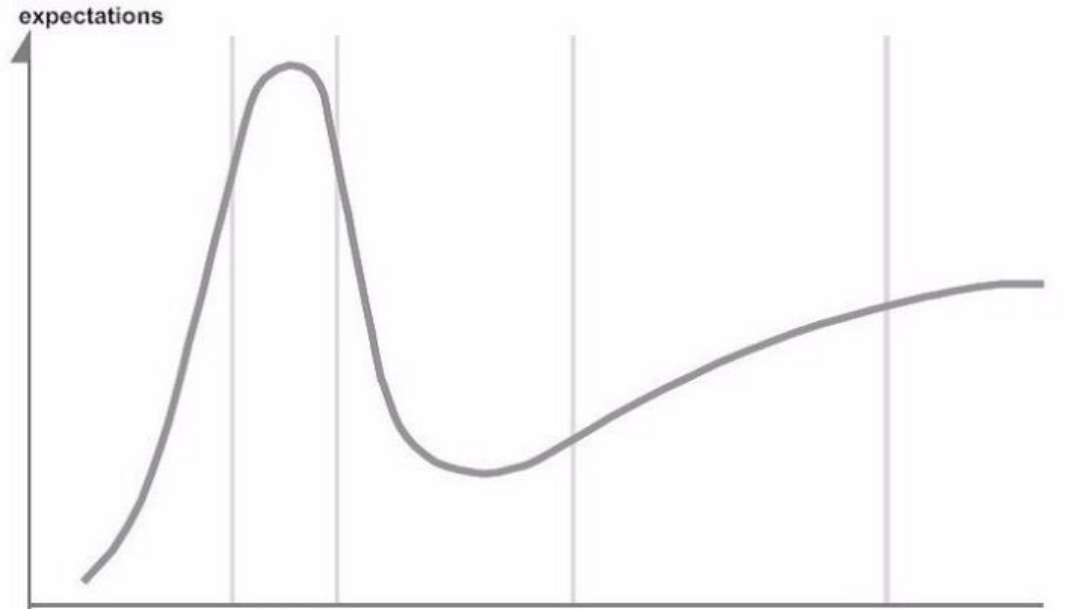


我们眼里的新技术

老板眼里的新技术

新技术落地

- 发现新技术
  - 很兴奋
  - 干翻世界
- 被干翻
  - 有风险
  - 有 Block
- 解决核心问题
  - 起死回生
- 持续迭代
  - 打造精品



目录

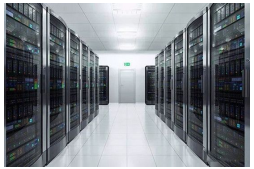
业务背景 - 跨云跨区

- 国内(两区)
- 海外(六区)
- 金山云、AWS、华为云、自建机房

金山云

HUAWEI

AWS

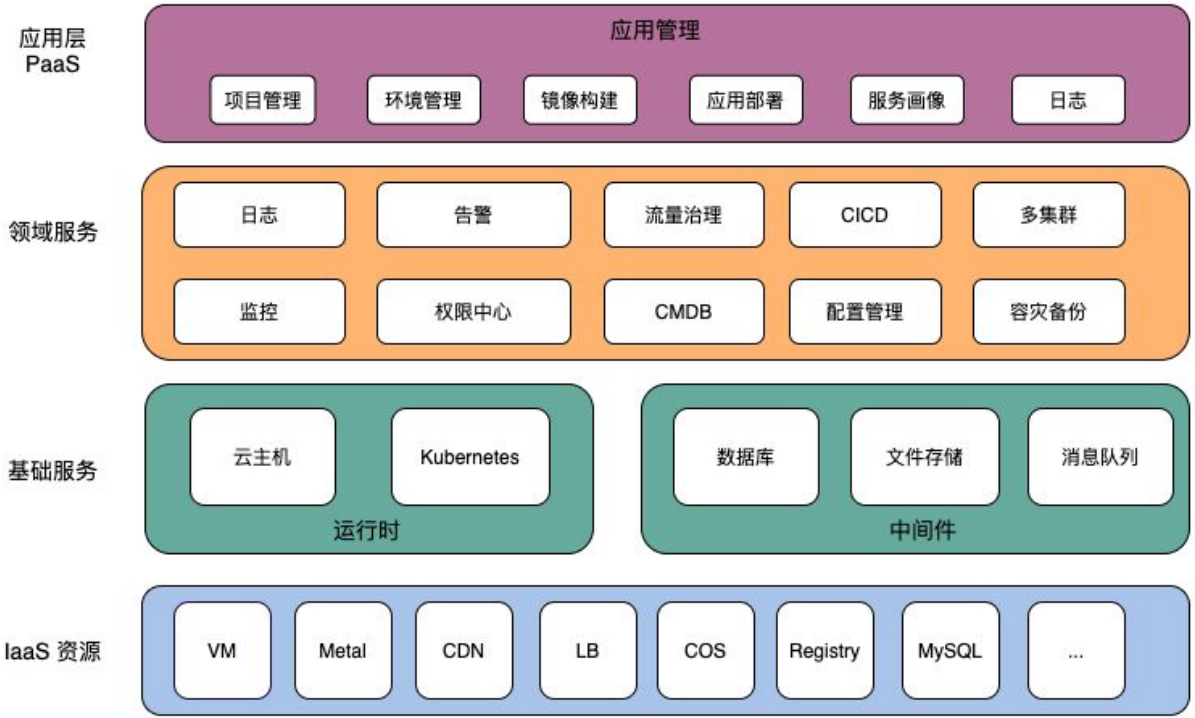


业务背景 - 超 80%的容器化率

- 金山办公成立于 1988 年
- 2015 年开始落地 Kubernetes
- 超过百个 Kubernetes 集群
- 孵化出的应用平台已经运行上数千应用，每天处理数百亿请求

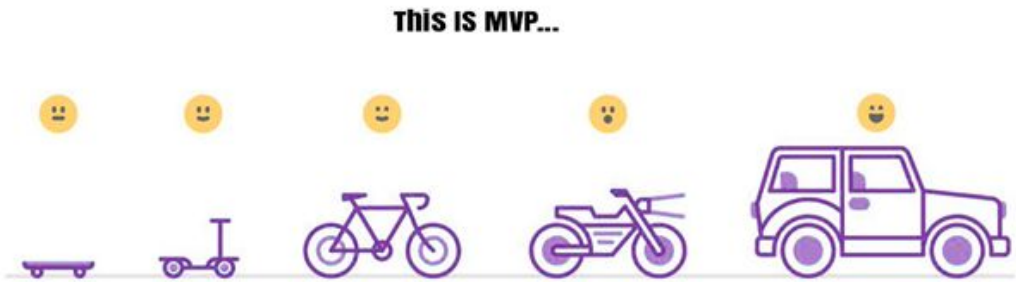
业务背景 - 面临的挑战

- 业务增长
- 容器化率高
- 体系正在建设



业务背景 - 业务对 CICD 的需求

- 跨网络。服务上云，但代码不能出公司。需要在云上组装，而在内网进行构建容器镜像。
- 可大规模执行流水线。CICD 是自动化系统，执行次数越多，意味着节省的人力时间越多。在未来，CICD 会承载越来越多的场景。集群安装、证书巡检....CICD 提供的是一次性运行时。
- 零停机运维。
- 较短时间交付，持续迭代。



选型比较 - Jenkins VS Tekton

- 我使用过的一些 CI 工具

  Gitlab CI/ Jenkins/ Tekton/ GitHub Action/ ArgoCD

- User - yaml vs groovy

  Outer DSL 简单易掌握


- Developer - go vs java

  Inner DSL 高效易维护

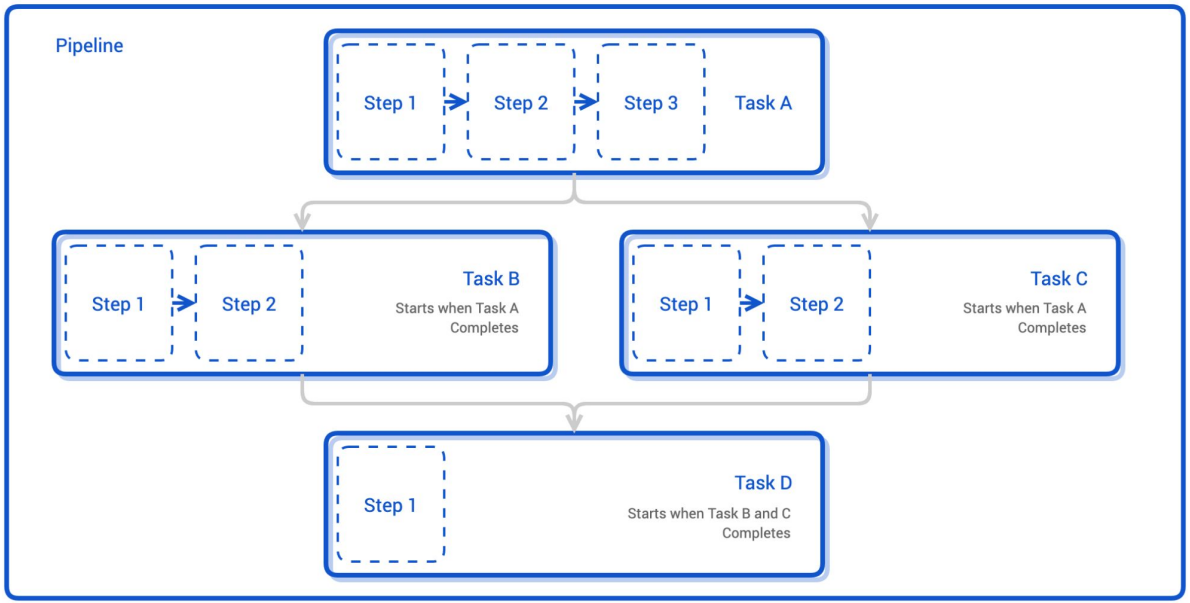

- Ecosystem - 100+ vs 1500+

  生态，能复用的原子要多


功能	Jenkins	Tekton
编程语言	Java	Golang
开发插件语言	Java	Shell、Yaml
流水线描述语言	Groovy、Shell	Yaml、Shell
插件生态	很多插件，LDAP、GitLab	不足
插件数量	1500+	100+
插件之间的兼容性	可能会有冲突，不能随便升级	完全兼容
二次开发	封装 Api	组合 Task
是否高可用	集成 Gearman、主从模式	依赖 Kuberntes 的高可用
单实例并发构建规模	几百并发	依赖 Kuberntes 的 Pod 管理能力，可以很大
数据存储	本地磁盘	Etcd
是否支持自动触发	支持	支持
是否有商业支持	无	无

但 Tekton 开发 Task 门槛并不高，掌握基本 Shell 、Yaml 就行

选型比较- Tekton 的 Pipeline



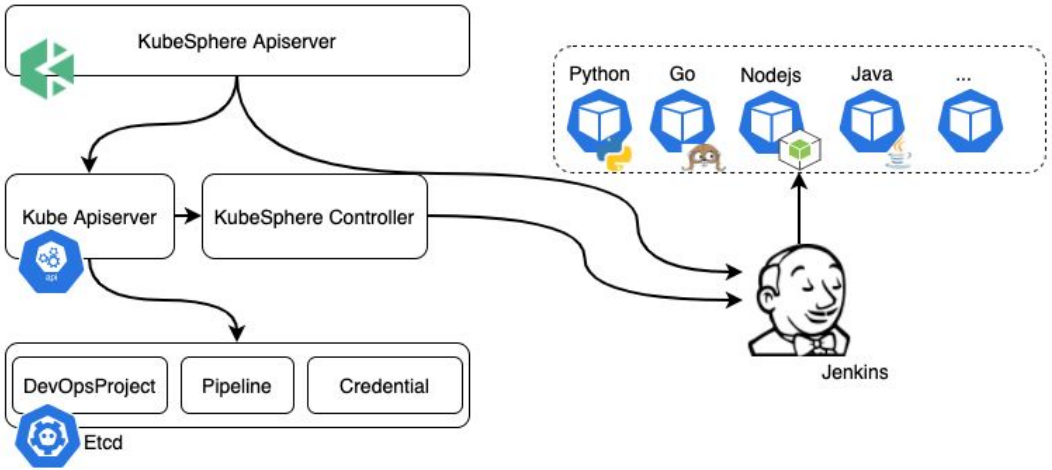

```
apiVersion: tekton.dev/v1beta1
kind: PipelineRun
metadata:
  name: test-demo
spec:
  pipelineSpec:
    tasks:
      - name: test-1
        taskSpec:
          steps:
            - name: run
              image: alpine
              script: |
                sleep 100
```

https://hub.tekton.dev/ 提供了大量 Task(拉取代码、构建、推送通知、发送邮件、代码检查)

选型比较 - 基于 Jenkins 的 CICD 平台

- Operator 连接万物, CRD+Controller
- 外部资源 -> 声明式描述
- 并没有解决核心问题
  - 可维护性
  - 可用性
  - 并发性能

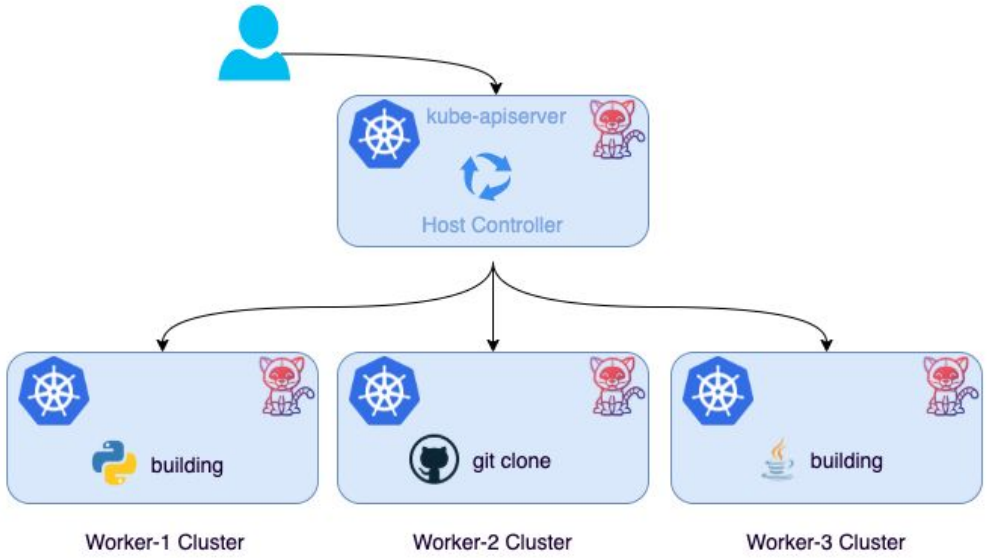
产品概念	Kubernetes 对象	Jenkins 对象
DevOps 工程	DevopsProject	文件夹
流水线	Pipeline	流水线/多分支流水线
凭证	Credential	文件夹下的凭据



KubeSphere DevOps 3.0

选型比较 - 多集群下的 Tekton CICD 方案

- 借助开源多集群管理项目
  - KubeFed v2
  - Karmada
  - Open Cluster Management
- 多集群能带来的优势
  - 扩展性
  - 可维护性
- 服务在云上，代码不能出内网
  - 内网穿透？有风险



设计落地 - 最终实现

- web

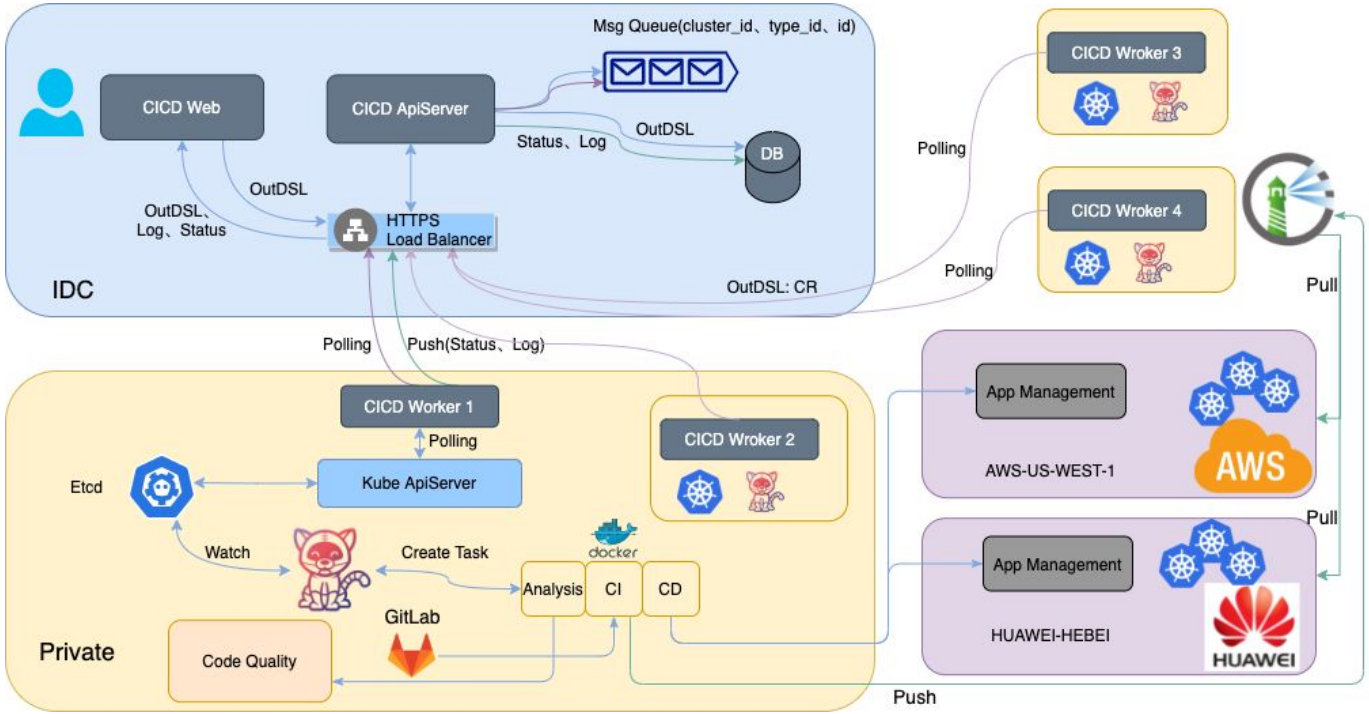
提供用户操作界面

- apiserver

提供用户操作、worker拉取任务的接口

- worker

拉取当前集群的任务、执行并推送结果



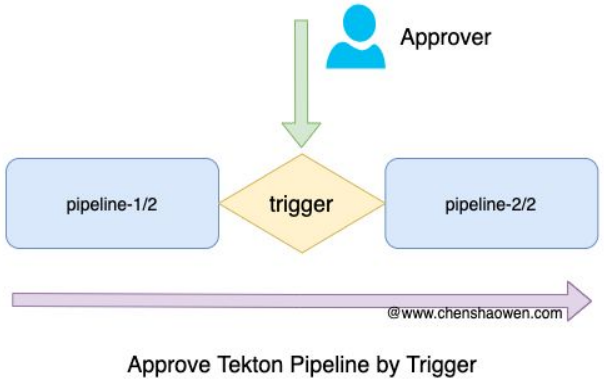
设计落地 - 不同角色视角

- 运维 - 添加集群
  - 准备 Kubernetes 1.18+
  - 安装 Tekton 0.24
  - 部署 worker
- 用户
  - 根据 Pipeline 模板、Task，编辑流水线
  - 保存到 DB、生成事件消息
- Woker
  - 使用 Token 鉴权、过滤拉取当前集群的 Resource/Pipelinerun
  - 将 Resource Status/ Container log 回写到 DB

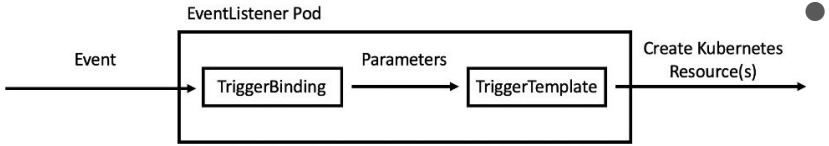
```
containers:
- name: worker
    image: 'harbor.domain/sre/worker:latest'
    command:
    - /worker
    - '--token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'
    - '--serverUrl=http://10.10.10.10:30000/api/v1'
```

设计落地 - Tekton 的审批方案

- 当流水线 pipeline-1/2 执行完成时，通知审批者。
- 审批者审批通过后，触发 pipeline-2/2 执行。
- pipeline-2/2 执行结束，完成整条流水线。



Approve Tekton Pipeline by Trigger

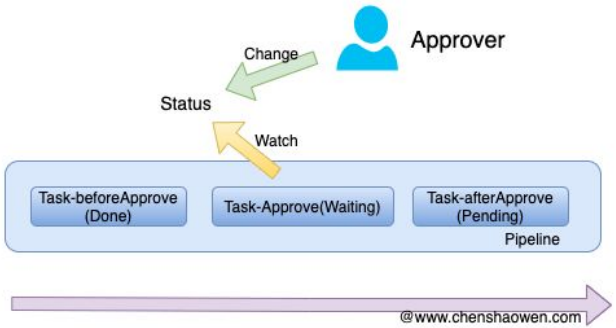


- 审批之后，推送一个触发事件 Event
- EventController 收到这个事件之后，从 TriggerBinding 提取出事件内的参数 Parameters
- TriggerTemplate 利用传递过来的参数 Parameters，创建流水线 pipeline-2/2 。

设计落地 - Tekton 的审批方案

在一条流水线中，插入一个用于 审批控制的 Task-Approve。

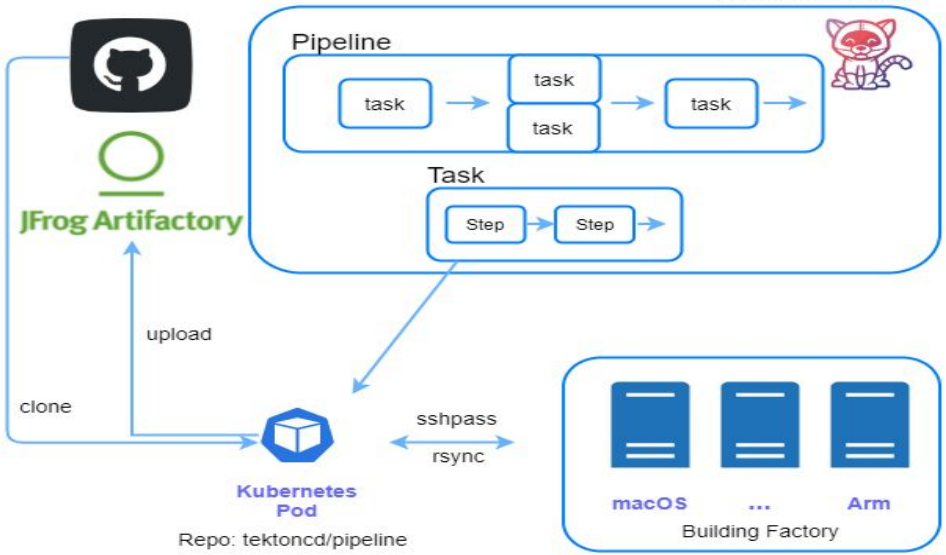
- 在使用审批原子时，需要同步 创建一个 ConfigMap，用于保存审批的状态 Status=init
- 当流水线执行完成 Task-beforeApprove 任务时，启动 Task-Approve 任务，修改状态 Status=notifying。Task-Approve 任务一直处于等待状态。
- 发送通知给 Approver，修改状态 Status=notified
- 审批者审批流水线，允许执行，修改状态 Status=success
- Task-Approve 检测到 Status=success，立即结束等待状态，完成当前 Task
- 流水线继续执行审批后的任务 Task-afterApprove，直至结束



Approve Tekton Pipeline by Task

落地设计 - Tekton 的物理机接入方案

- 克隆代码
- 执行 rsync 将代码同步到构建机
- 执行 sshpass 在构建机上执行构建命令
- 执行 rsync 将构建机中的构建产物同步到容器
- 归档构建产物(示例中, 这一步会被省略, 仅验证能拿到构建产物)

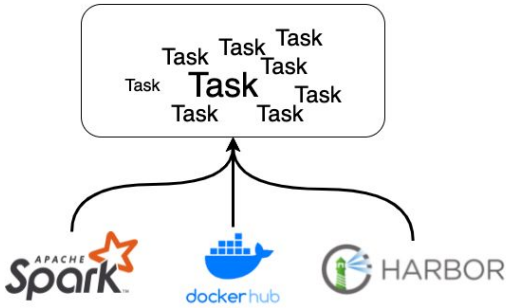


后期展望 - 优化

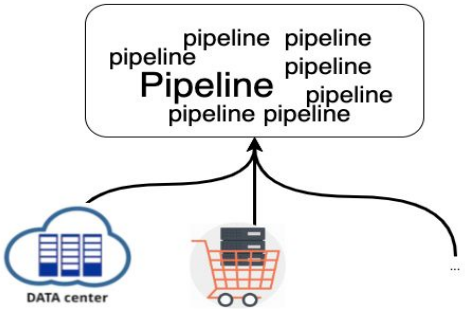
- 通过开发插件集成更多外部系统
- 推动社区解决审批、暂停、物理机构建
- 启动速度
  - Pod Create -> Ready 时间太长
  - Virtual-kubelet + Serverless

后期展望 - 更多的 CICD 应用场景

- 交付
  - Helm 面向的是 K8s，面向 VM/BMS 的服务如何交付？
- 自动化运维
  - 故障处理
  - 增删节点
  - 申请资源
  - 服务变更
  - ...



面向服务提供商



面向集成、交付厂商

总结回顾

- 介绍业务背景
- 为什么选择 Tekton 而不是 Jenkins
- 几个 CICD 场景下的 Tekton 解决方案
- 后期的计划和承载更多场景的 CICD

QA & Thanks

Don’t Be The Same.
Be Better.