

# 云原生时代下的 CI/CD 变革

张晋涛

# 个人介绍

- 张晋涛
- Container/Docker/Kubernetes
- <https://moelove.info>
- <https://github.com/tao12345666333>
- <https://zhuanlan.zhihu.com/container>



# 目录

- CI/CD
- 云原生
- 变革及挑战
- 探索及实践

# CI/CD

- 是什么
- 为什么
- 怎么做

# CI/CD 是什么

- 持续集成
- 继续交付
- 持续部署

# 为什么需要 CI/CD

- 提升效率
- 发现问题
- 快速部署
- 快速验证

# 怎么做

## 发展阶段

- 手工集成
- 批量工具
- 自动化

# 怎么做

## 发展阶段

- 手工集成
- 批量工具
- 自动化

## CI 服务

- 代码托管(GitHub/GitLab)
- Jenkins/JenkinsX
- Travis CI/Circle CI
- Prow (k8s/Istio)
- Spinnaker/Tekton
- AWS CodeBuild/Azure DevOps



# 云原生

- 微服务
- Container
- Kubernetes
- 12 Factor APP

# 微服务

## 优势

- 可靠
- 可治理
- 降低复杂度
- 发布周期可缩短

# 微服务

## 优势

- 可靠
- 可治理
- 降低复杂度
- 发布周期可缩短

## 挑战

- 服务拆分粒度
- 基础设施支持
- 配置化
- CI/CD 效率和可用性

# Container

## 优势

- 资源隔离
- 消除环境差异
- 标准交付单元 (OCI)

# Container

## 优势

- 资源隔离
- 消除环境差异
- 标准交付单元 (OCI)

## 实现/运行时

- Docker
- containerd
- CRI-O
- runc
- Kata
- rkt

# Kubernetes

## 云原生应用的基石

- 容器编排
- 部署
- 扩容
- 管理
- 云厂商支持
- CNCF 及其生态

# 12 Factor APP

## 主体内容

- 标准化
- 可移植性
- 可扩展
- 减小环境差异
- 配置分离
- 应用无需关注日志存储等

# 12 Factor APP

## 主体内容

- 标准化
- 可移植性
- 可扩展
- 减小环境差异
- 配置分离
- 应用无需关注日志存储等

## 含义

- 应用有基准代码(代码仓库)
- 依赖隔离
- 线下/线上环境尽量一致
- 配置管理
- 基础组件支持



# 变革及挑战

## CI 服务

- 频率变高
- 任务增多
- 快速扩/缩容
- 稳定性
- 耗时

## 配置维护

- 大量配置
- 新增频繁

# 探索和实践

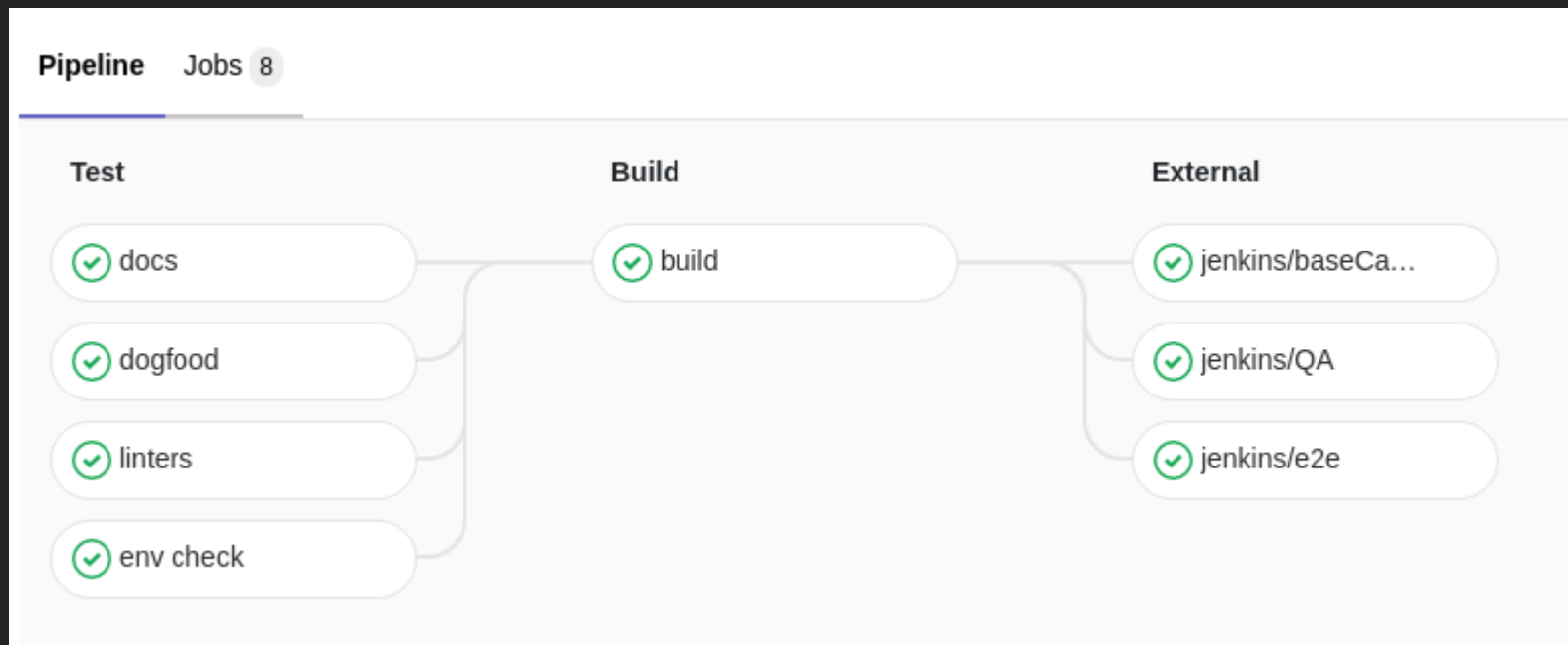
## 探索

- Prow
- Jenkins/Jenkinsx
- 其他

## GitLab CI

- 灵活/易用
- 开放
- C/S 架构
- 原生 Kubernetes 支持
- CI 效率优化

# Pipeline



# 构建镜像

```

image: registry.docker-cn.com/taobeier/docker:18.03
services:
  - name: registry.docker-cn.com/taobeier/docker:stable-dind
    alias: docker
variables:
  DOCKER_DRIVER: overlay2
  IMAGE_NAME: $CI_REGISTRY/$CI_PROJECT_PATH
before_script:
  - docker login -u gitlab-ci-token -p $CI_JOB_TOKEN $CI_REGISTRY
.build_template: &build_definition
  script:
    - docker build -t "$IMAGE_NAME:$IMAGE_TAG" .
    - docker images
    - docker push "$IMAGE_NAME:$IMAGE_TAG"
build-latest:
  variables:
    IMAGE_TAG: latest
  script:
    <<: *build_definition
  
```

# 运行测试

```
image: registry.docker-cn.com/taobeier/docker

variables:
  DOCKER_DRIVER: overlay2      # overlay2 is best but need kernel >= 4.2

services:
  - name: registry.docker-cn.com/taobeier/docker:stable-dind
    alias: docker

build_and_test:
  tags:
    - build
  script:
    - apk add --no-cache py-pip
    - pip install -i https://mirrors.ustc.edu.cn/pypi/web/simple docker-compose
    - docker-compose up -d
    - docker-compose run --rm web pytest -s -v tests/test_session.py
```

# 基础镜像选择

- 尽可能减少镜像体积
- 但并不是越小越好
- Alpine Linux 使用 musl 而不是 glibc
- 最佳实践

# CD 实践

## 交付

- 源码/tag/commit id
- tag 包/zip 包
- war包/wheels包/rpm包/二进制
- Docker 镜像

# CD 实践

## 交付

- 源码/tag/commit id
- tag 包/zip 包
- war包/wheels包/rpm包/二进制
- Docker 镜像

## 部署

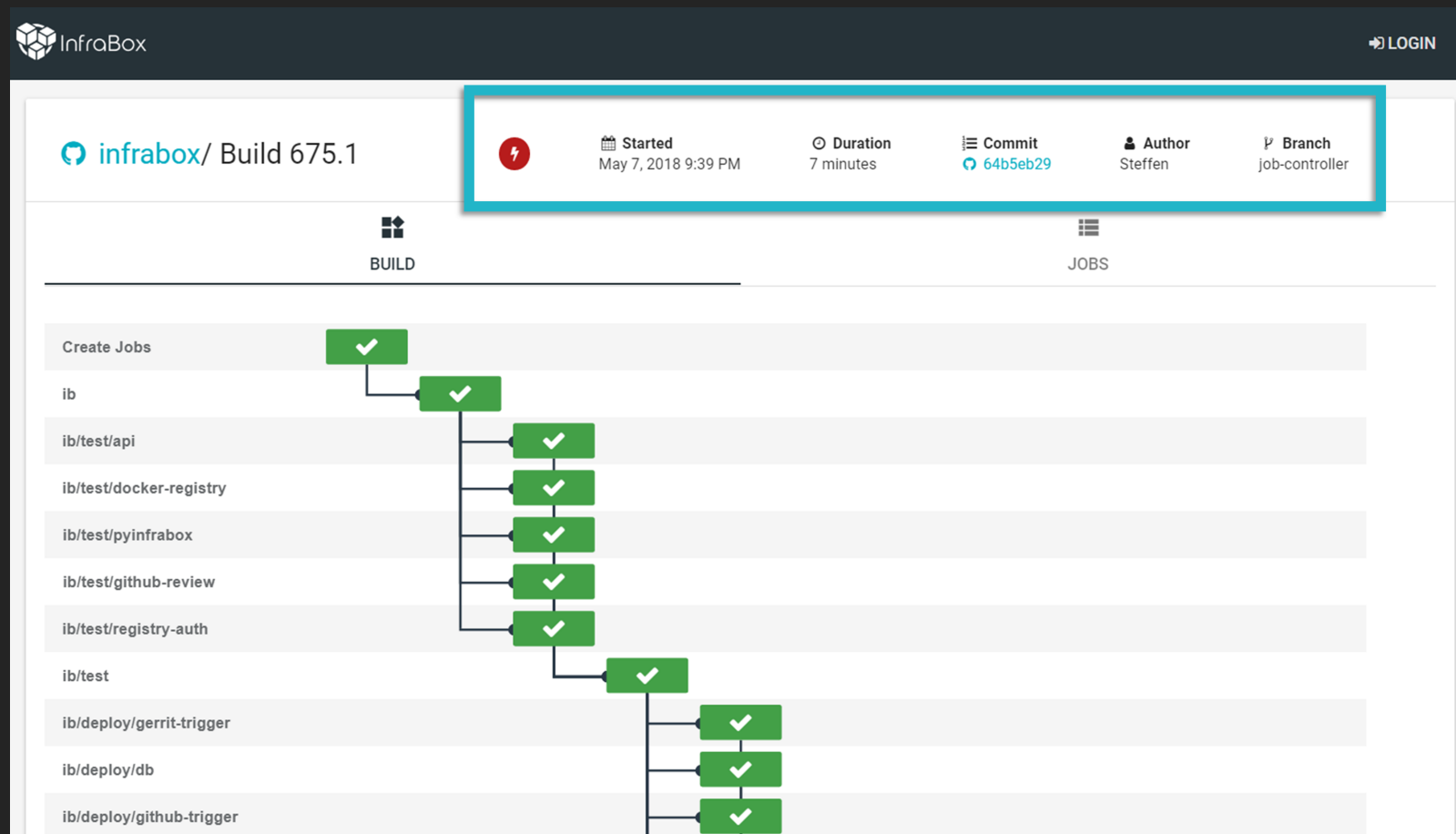
- 手动
- 平台化



# 平台化

## InfraBox

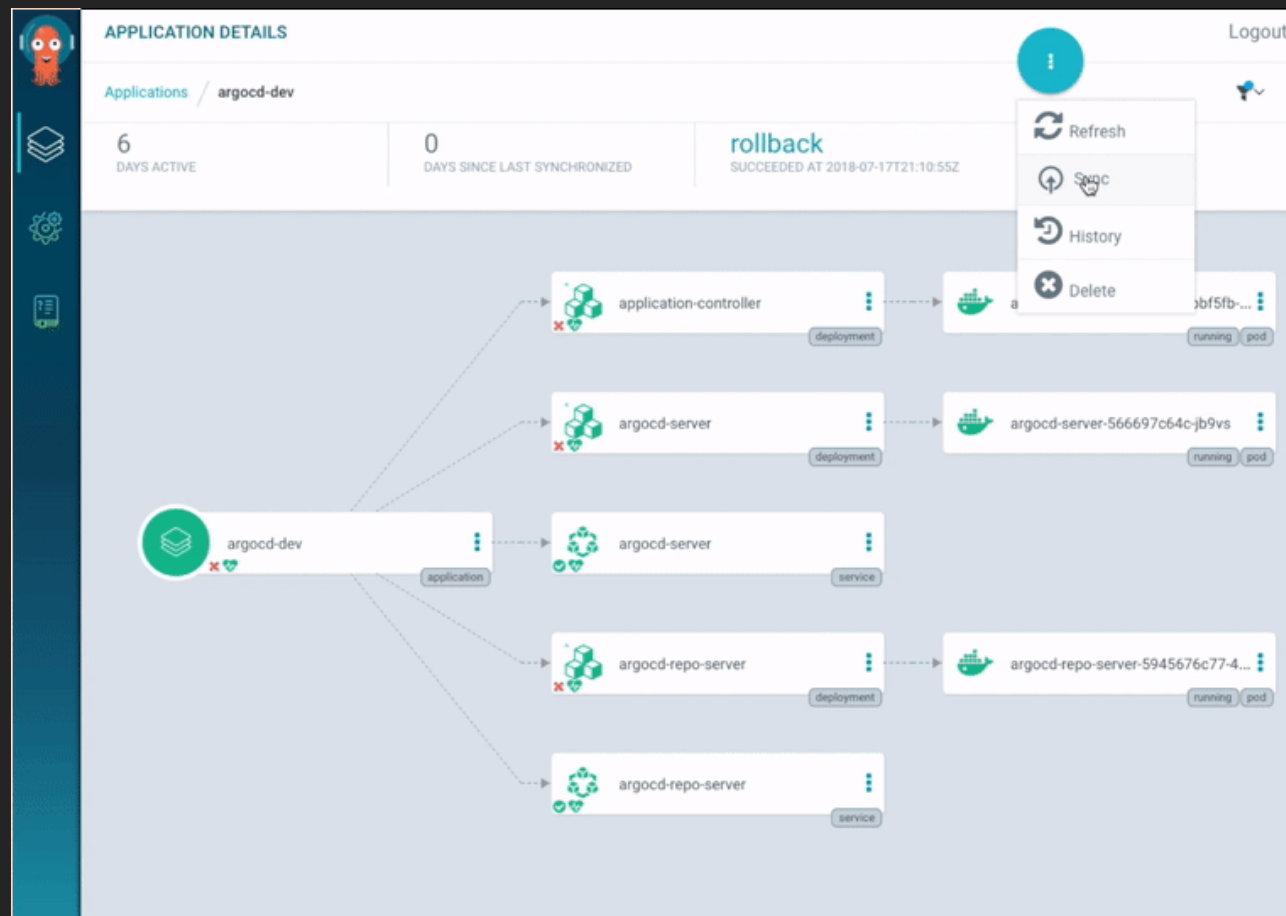
- knative
- 动态 workflow
- 不支持 GitLab



# 平台化

## Argo

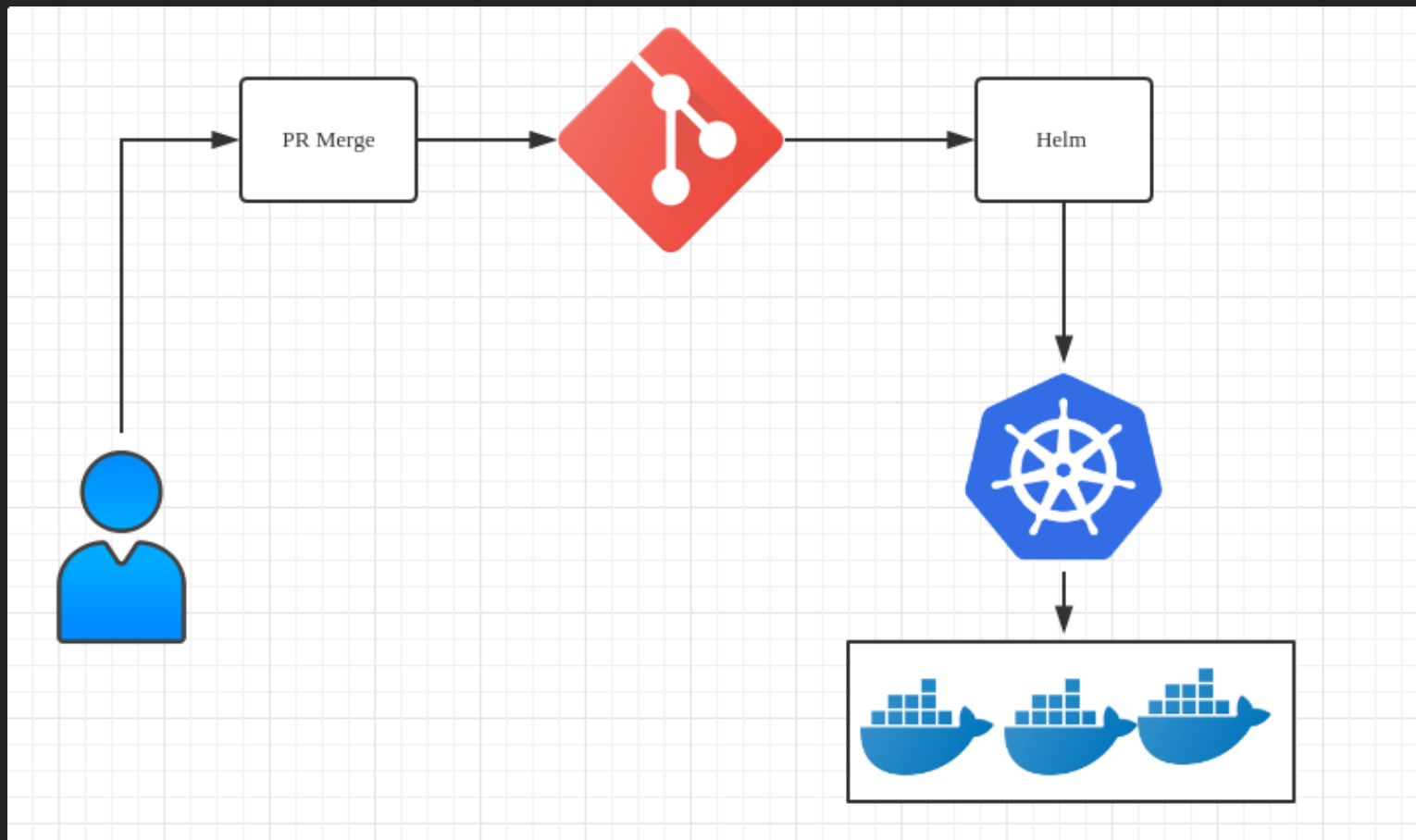
- DAG
- workflow引擎
- CRD
- All in K8S



# 平台化

## GitLab CI

- 灵活
- 易用



# GitLab CI for CD

## 特点

- All in K8S
- Kubectl
- Helm

```
deploy:
  image: $CI_REGISTRY/devops/helm
  stage: deploy
  script:
    - helm init --client-only
    - helm upgrade --install
      --set image.tag=$(cat VERSION)
      --wait
      --namespace $NAMESPACE
  only:
    - master
```

# 发展方向

## 集中式管理

- 数据统计

## GitLab Bot

- Issue 处理
- 授权管理
- 覆盖率
- 配置验证

# Q&A

## Thanks!

