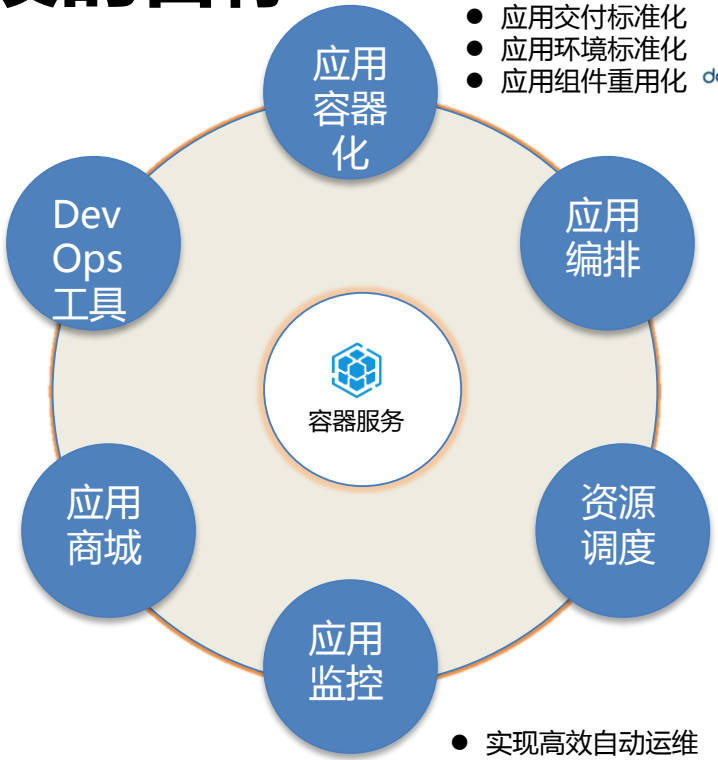
K8S快速构建DevOps流水线及其最佳实践分享

阿里云容器服务—流生
2019年1月23日

PaaS平台建设的目标



- 应用交付标准化
- 应用环境标准化
- 应用组件重用化

docker hub

Bamboo

Travis CI

- 支持敏捷开发
- 实现DevOps
- 支持灰度发布
- 支持A / B Test
- 支持在线升级/蓝绿发布

- 一键部署复杂应用
- 应用自动弹性伸缩
- 应用自动HA和Failover
- 微服务架构管理

kubernetes

docker SWARM

RANCHER

- 新一代CaaS
- 软件能力开放

- 解决测试开发环境不一致
- 异构资源智能调度
- 混合云无缝部署

kubernetes

docker SWARM

- 实现高效自动运维
- 保障业务安全稳定
- 实现APM

Grafana

Nagios

elastic

Ganglia

能带来什么效果？

与传统IT方式相比，减少 80% 的成本

已有应用

容器化交付

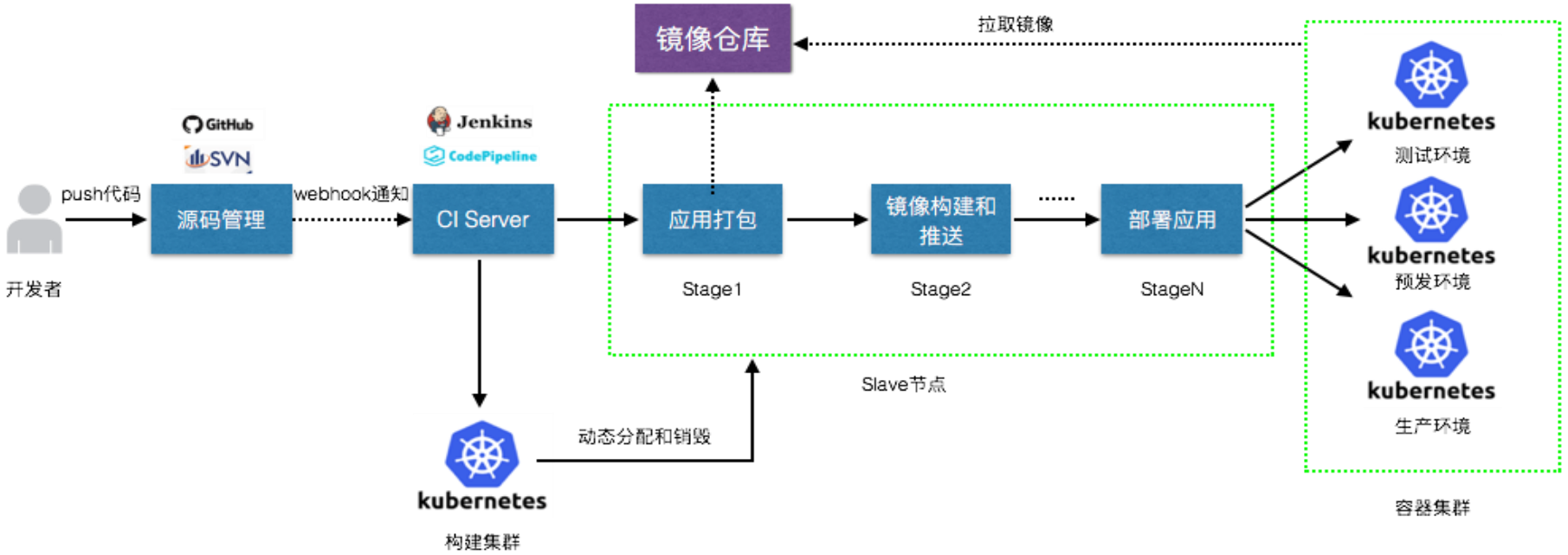
敏捷基础设施
支持专有云、公共云、混合云等环境

敏捷研发流程
整合持续集成、持续交付DevOps能力

敏捷应用架构
采用微服务化架构开发新的业务模块

提高效率，快速迭代

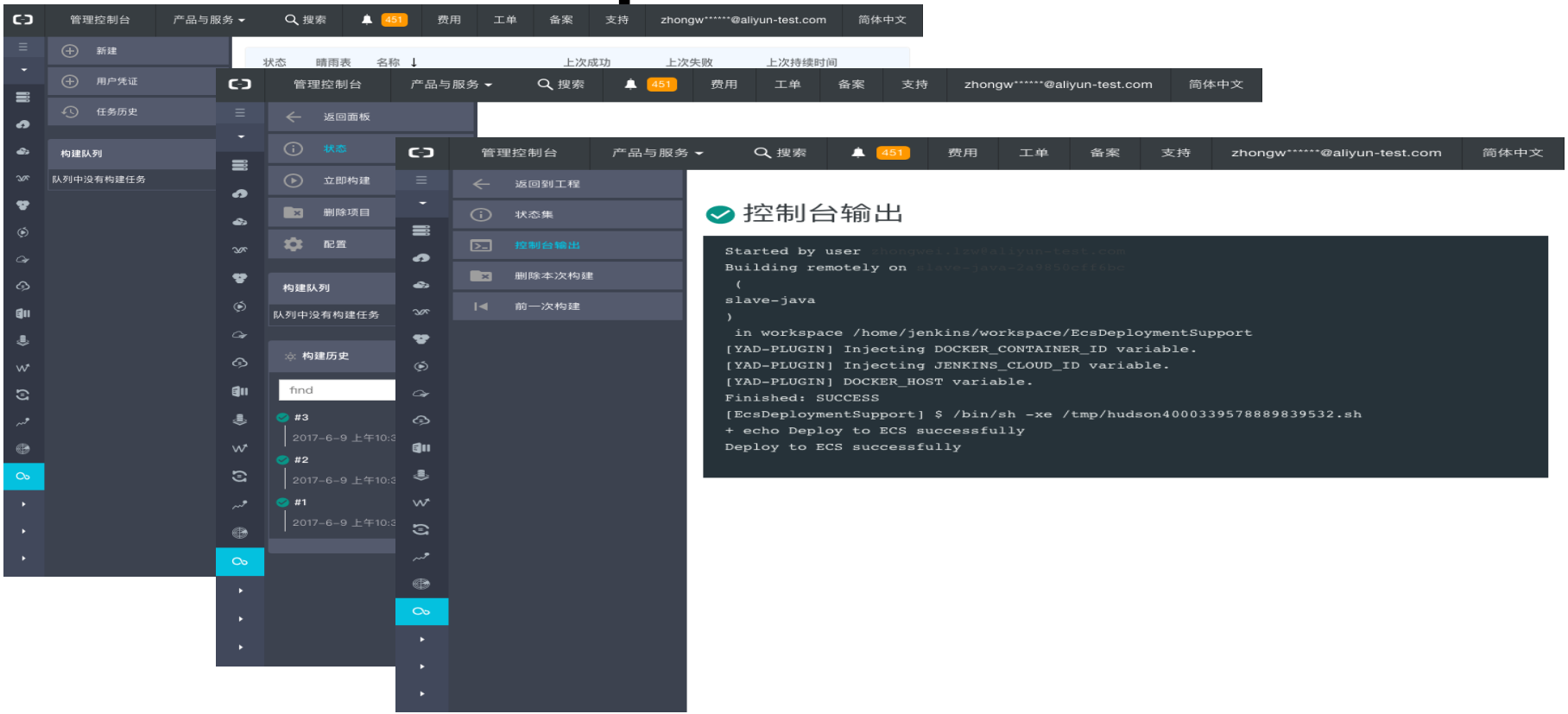
一个通用的CICD流程示意图



几种快速建立CICD流水线的方式

- Jenkins As Service （CodePipeline）
- 开源Jenkins
- 开源Jenkins X Platform

Jenkins As Service -- CodePipeline



CodePipeline的优缺点

优点：

- 继承开源Jenkins的使用习惯
- 无需运维，开箱即用
- 资源按需生成，动态分配动态销毁
- 支持多种部署方式
- 与阿里云容器服务、OSS服务等产品无缝集成
- 多种编译构建环境
- OpenAPI支持任务、凭证的CRUD
- 免费

缺点：

- 不支持Jenkinsfile

开源 Jenkins

使用开源Jenkins快速完成应用编译打包、镜像构建、应用部署的流水线

demo示例

Kaniko工具的使用介绍

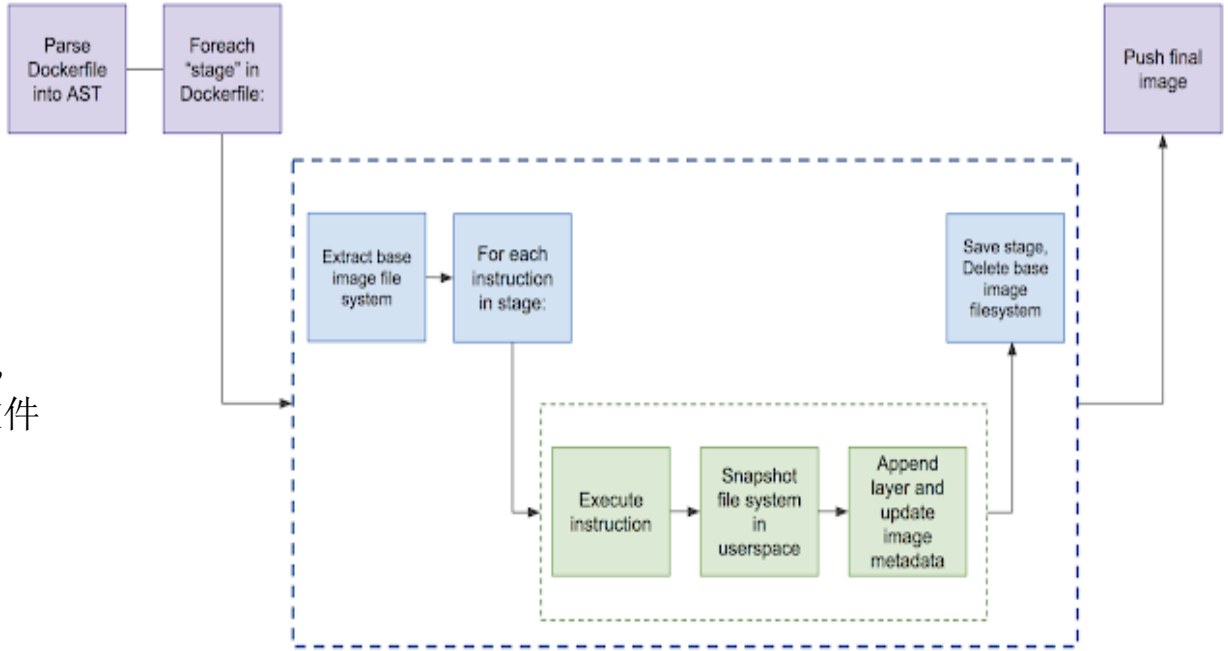
Kaniko是一个无需挂载docker daemon即可在用户空间完成镜像打包和上传的工具。

```
// 添加第四个stage，运行容器镜像构建和推送命令， 用到了environment中定义的groovy环境变量
stage('Image Build And Publish'){
  steps{
      container("kaniko") {
          sh "kaniko -f `pwd`/Dockerfile -c `pwd` --destination=${ORIGIN_REPO}/${REPO}:${IMAGE_TAG}"
      }
  }
}
```

Kaniko工作原理

- 三个参数：
Dockerfile、context、docker registry

- Kaniko解压文件系统，执行命令，在执行器镜像的用户空间中对文件系统做快照



开源 Jenkins

优点：

- 自主可控、灵活
- 构建资源按需生成，动态分配动态销毁
- 支持Jenkinsfile

缺点：

- 运维、安全、构建资源
- 非云原生

Jenkins X Platform

Jenkins X is a CI/CD solution for modern cloud applications on Kubernetes

Jenkins X Platform

- 工具的自动化安装升级和配置

Helm, Skaffold, Kaniko, Jenkins, Monocular, Nexus etc

- 应用在Kubernetes集群中的自动化CICD过程

Docker images， HelmChart， Pipeline

- GitOps管理应用再不同环境中的推进和回滚

Develop -> Staging -> Production

Jenkins X Platform

	**Jenkins** - Fully integrated CI / CD solution with opinionated yet customisable pipelines and environments
	**Nexus** - Artifact repository (pluggable so we can switch with Artifactory)
	**Chartmuseum** - Helm Chart repository (Helm is the most popular Kubernetes package manager used to install and upgrade your applications)
	**Monocular** - Web UI for searching and discovering Helm Charts

Jenkins X Platform

使用开源Jenkins X创建流水线并演示GitOps管理应用发布

demo示例

Draft工具的使用介绍

Draft是一个能自动化帮你生成Dockerfile、Helm chart、Jenkinsfile等文件的开源工具。

工作流程：

- 检测语言类型
- 根据对应语言类型的最佳实践模板生成Dockerfile等文件

```
|-- charts
|   |-- Chart.yaml
|   |-- Makefile
|   |-- README.md
|   |-- templates
|   |   |-- deployment.yaml
|   |   |-- _helpers.tpl
|   |   |-- NOTES.txt
|   |   `-- service.yaml
|   `-- values.yaml
|-- Dockerfile
|-- pipeline.yaml
|-- preview
|   |-- Chart.yaml
|   |-- Makefile
|   |-- requirements.yaml
|   `-- values.yaml
|-- skaffold.yaml
`-- watch.sh
```

DRAFT

Skaffold工具的使用介绍

Skaffold是一款命令行工具，旨在促进kubernetes应用的持续开发，Skaffold可以将构建、推送及向kubernetes集群部署应用程序的过程自动化

- Skaffold使用小示例：
https://github.com/GoogleContainerTools/skaffold/tree/master/examples/getting-started

- skaffold.yaml详解
https://github.com/GoogleContainerTools/skaffold/blob/master/examples/annotated-skaffold.yaml

GitOps

1. 创建Staging环境
- GitHub自动创建environment-bj-staging项目 -> Jenkins自动创建environment-bj-staging任务
- master分支有代码更新 -> 触发Jenkins Job运行 -> 部署应用到Staging环境

2. 创建Production环境
- GitHub自动创建environment-bj-production项目 -> Jenkins自动创建environment-bj-production任务
- master分支有代码更新 -> 触发Jenkins Job运行 -> 部署应用到Production环境

3. 创建jenkins-x-demo应用并推送到master分支
- GitHub自动创建jenkins-x-demo项目 -> Jenkins自动创建jenkins-x-demo任务
- master分支有代码更新 -> 触发Jenkins Job运行 -> 构建镜像、HelmChart并推送 -> 自动merge到environment-bj-staging项目 -> 运行步骤1中的流程

4. 创建新分支feature/add-index并提交PR
- 有新PR生成-> 触发Jenkins Job运行 -> 构建镜像、HelmChart并推送 -> 部署为Preview环境-> 合并到master分支-> 运行步骤3中的流程

5. 推送到Production环境

参考

Jenkins on k8s: https://yq.aliyun.com/articles/683440

Gitlab CI： https://yq.aliyun.com/articles/672645

Jenkins x platform : https://yq.aliyun.com/articles/687938

Jenkins on serverless k8s: https://yq.aliyun.com/articles/685219

为了无法计算的价值 | 阿里云