

DOIS | 2018 · 深圳站  
DevOps 落地，从这里开始

# DevOps 国际峰会

暨 DevOps 金融峰会

指导单位： 云计算开源产业联盟  
Open Source Cloud Alliance for Industry (OSCAR)

主办单位： DevOps时代

 高效运维社区  
GreatOPS Community

时间：2018年11月2日-3日

地址：深圳市南山区圣淘沙大酒店（翡翠店）



# 多云 devops 实践

邱见

# 目录

- ➔ **1** 什么是多云？
- 2** 多云环境下的IBM Cloud Private
- 3** 我们遇到的挑战
- 4** ICP devops 实践

# 什么是多云？

多云代表了依赖于多个云提供商的云环境，如OpenStack, IBM Cloud, Azure, AWS.  
比如在私有云(OpenStack)中部署应用，同时使用公有云实现灵活的扩展

## 多云和混合云的区别

### 多云

依赖多个公有云提供商来承载多个应用。



### 混合云

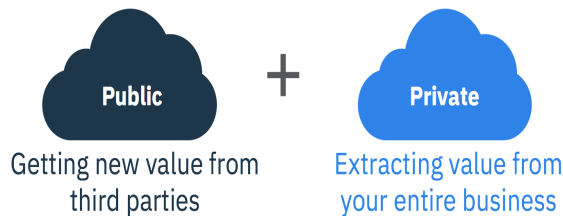
把一个应用同时部署在公有云，私有云，单独服务器上面。



# 为什么要使用多云？

- 把应用部署在最适合的平台之上
- 避免传统应用改造的痛点来适应新的平台
- 避免被公有云厂商绑定

*8 out of 10 committing to Multi-Cloud*  
*71% use 3 or more clouds*



# 目录

1 什么是多云？

➔ 2 多云环境下的IBM Cloud Private

3 我们遇到的挑战

4 ICP devops 实践

# 多云环境下的IBM Cloud Private

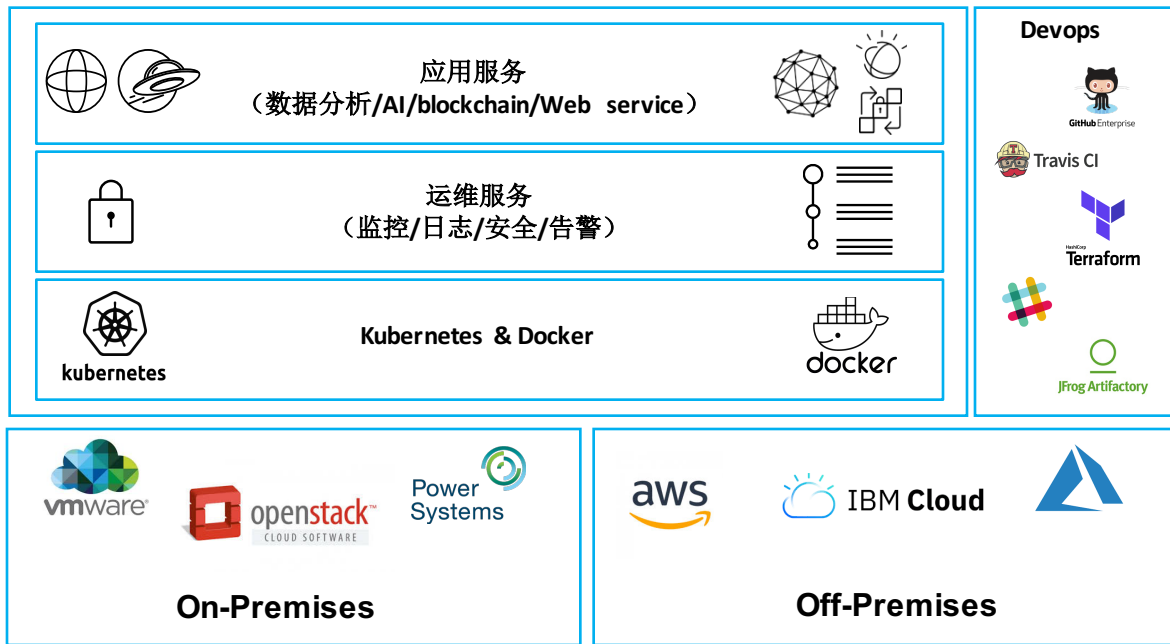


## 多云环境下的Kubernetes

- IBM Container Service
- IBM Cloud Private on AWS
- IBM Cloud Private on Openshift

## DevOps工具集

- Terraform
- Travis CI
- Slack
- Github Enterprise
- Artifactory



# 目录

1 什么是多云？

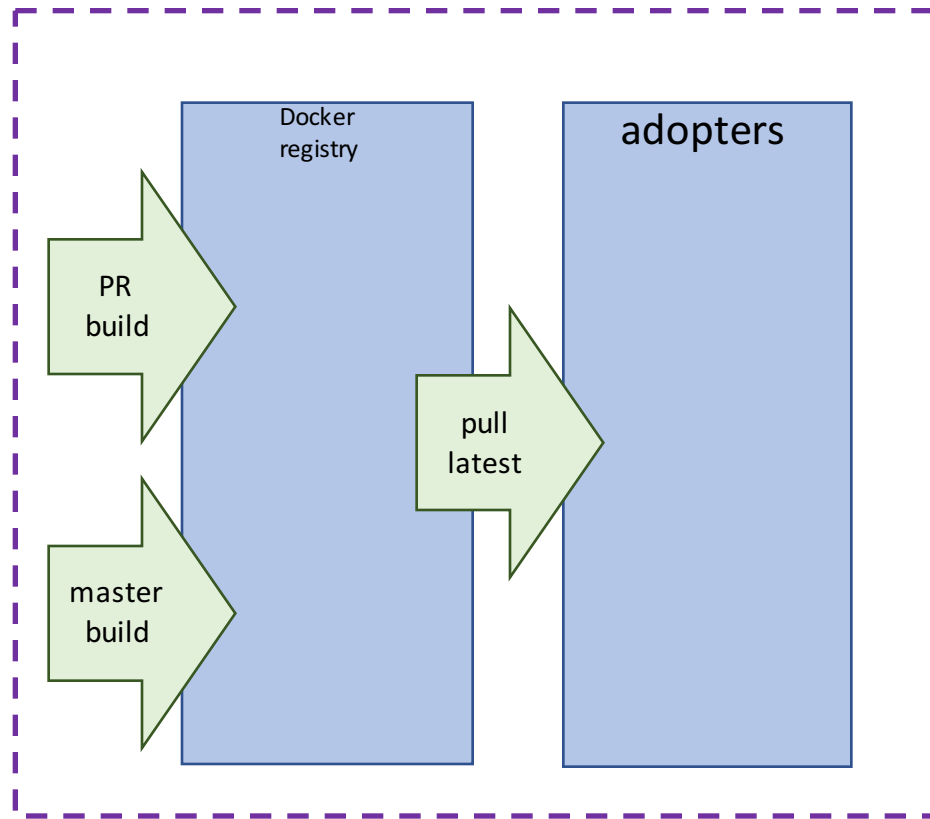
2 多云环境下的IBM Cloud Private

➔ 3 我们遇到的挑战

4 ICP devops 实践



# Devops 1.0



# 我们遇到的挑战

- 组件众多，开发团队众多
- 多云环境下平台测试异常复杂耗时
- 即是消费者又是生产者

# 组件众多



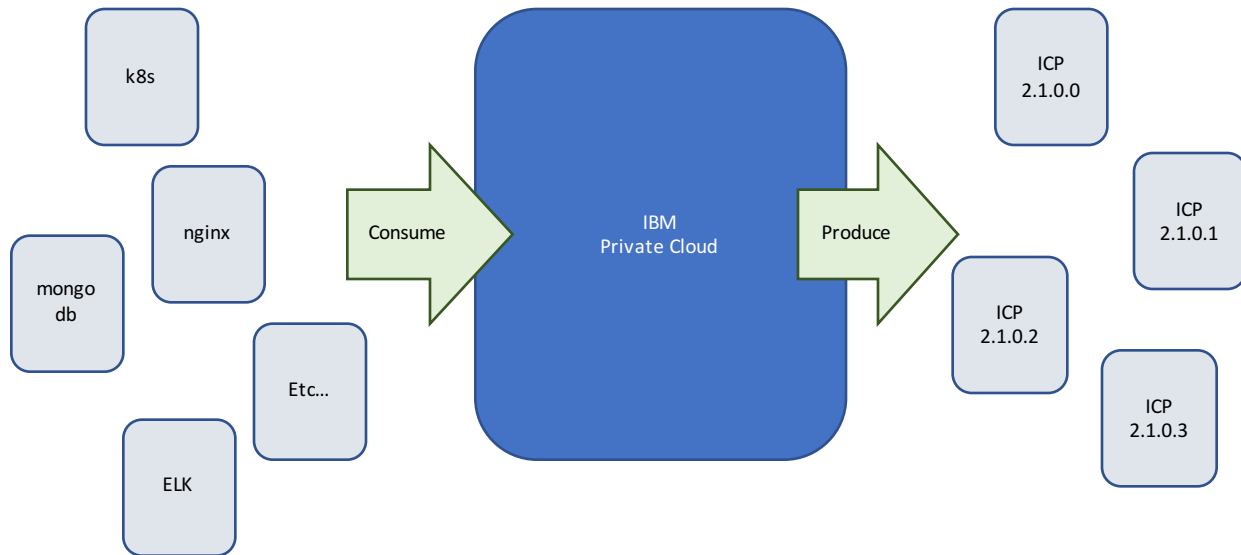
- 组件由**code/镜像/helm chart**构成
- 每个组件开发进度不同
- 组件之间的互相依赖

# 测试异常复杂耗时

- 一个**PR**的测试通常需要**1**个多小时
- 需要进行多种环境的测试
  - **X86 & power**
  - 多个公有云平台
- 多个团队同时开发导致**PR**不断**rebase**不断测试

# 即是消费者也是生产者

- 消费者
  - 开源软件
  - 现有IBM中间件
- 生产者
  - 平台用来承载其他团队的应用服务



# 目录

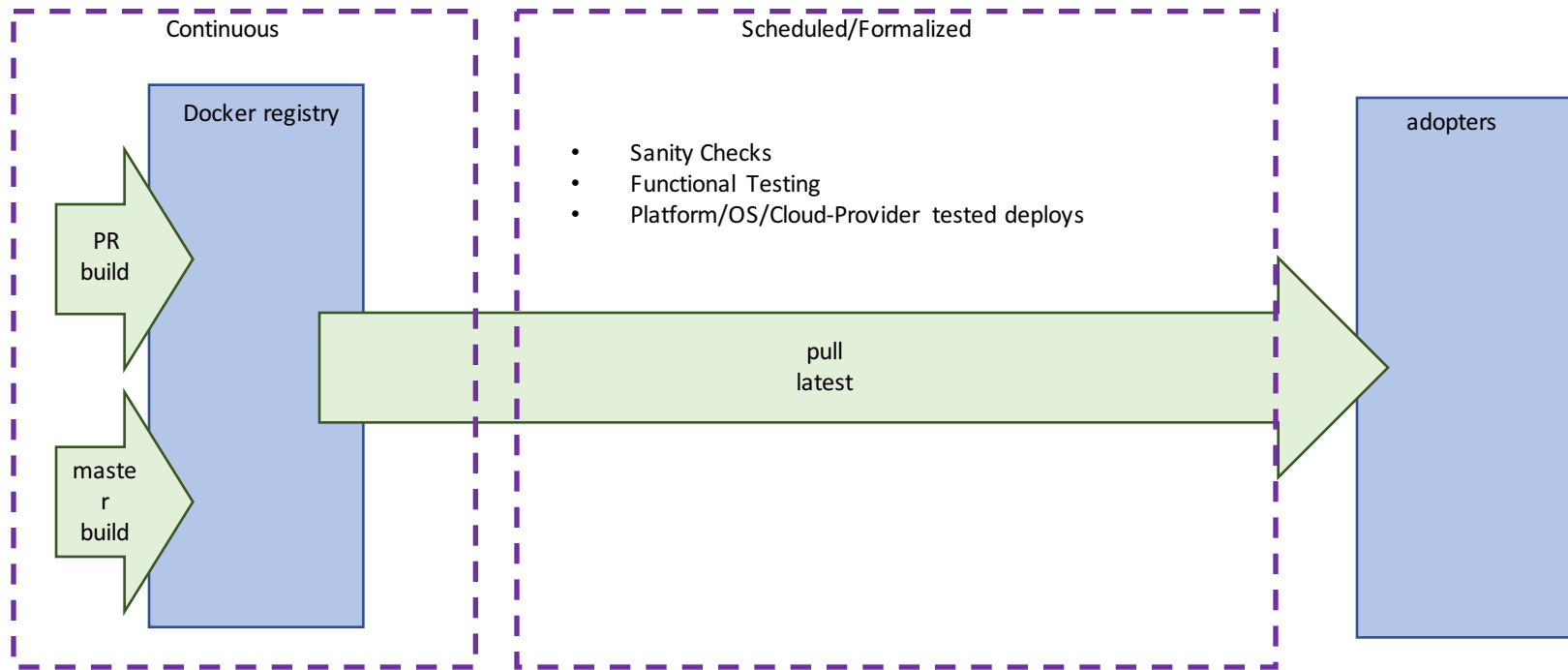
1 什么是多云？

2 多云环境下的IBM Cloud Private

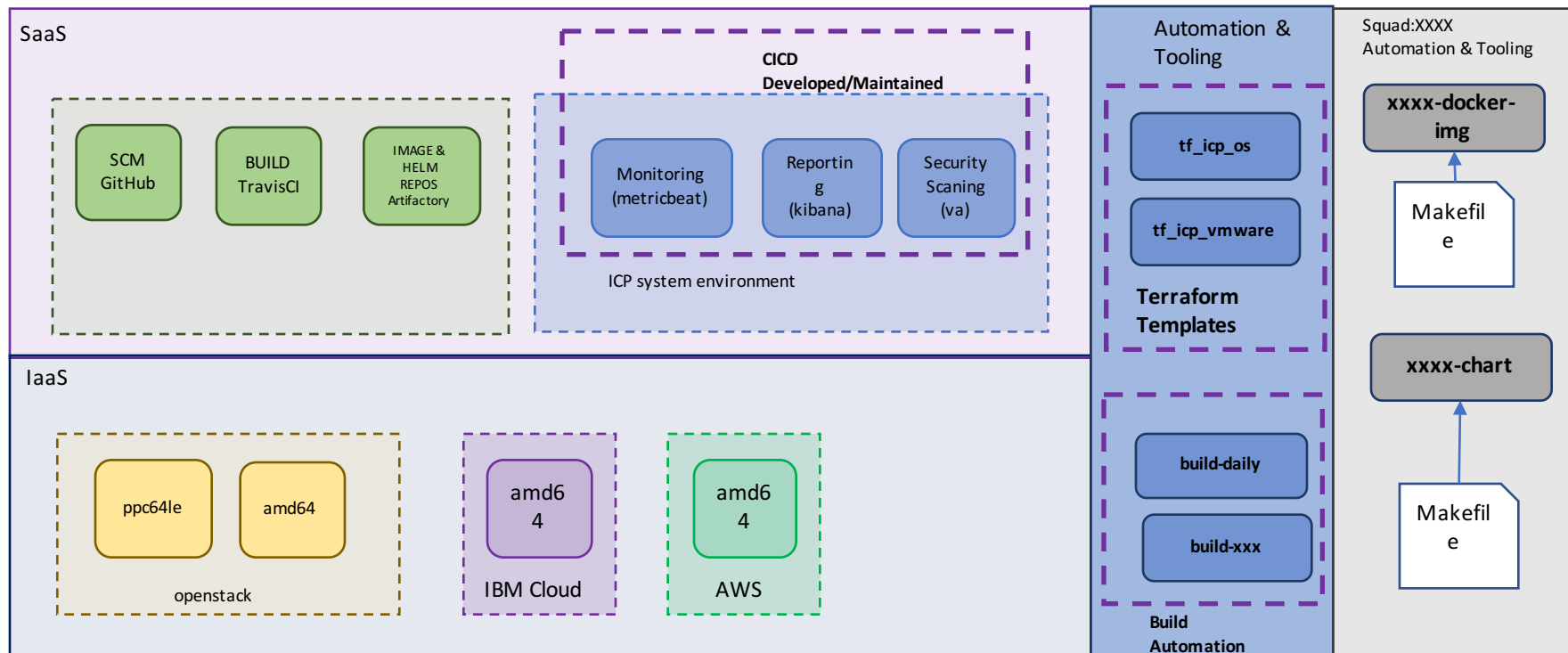
3 我们遇到的挑战

➔ 4 ICP devops 实践

# 增长CICD pipeline

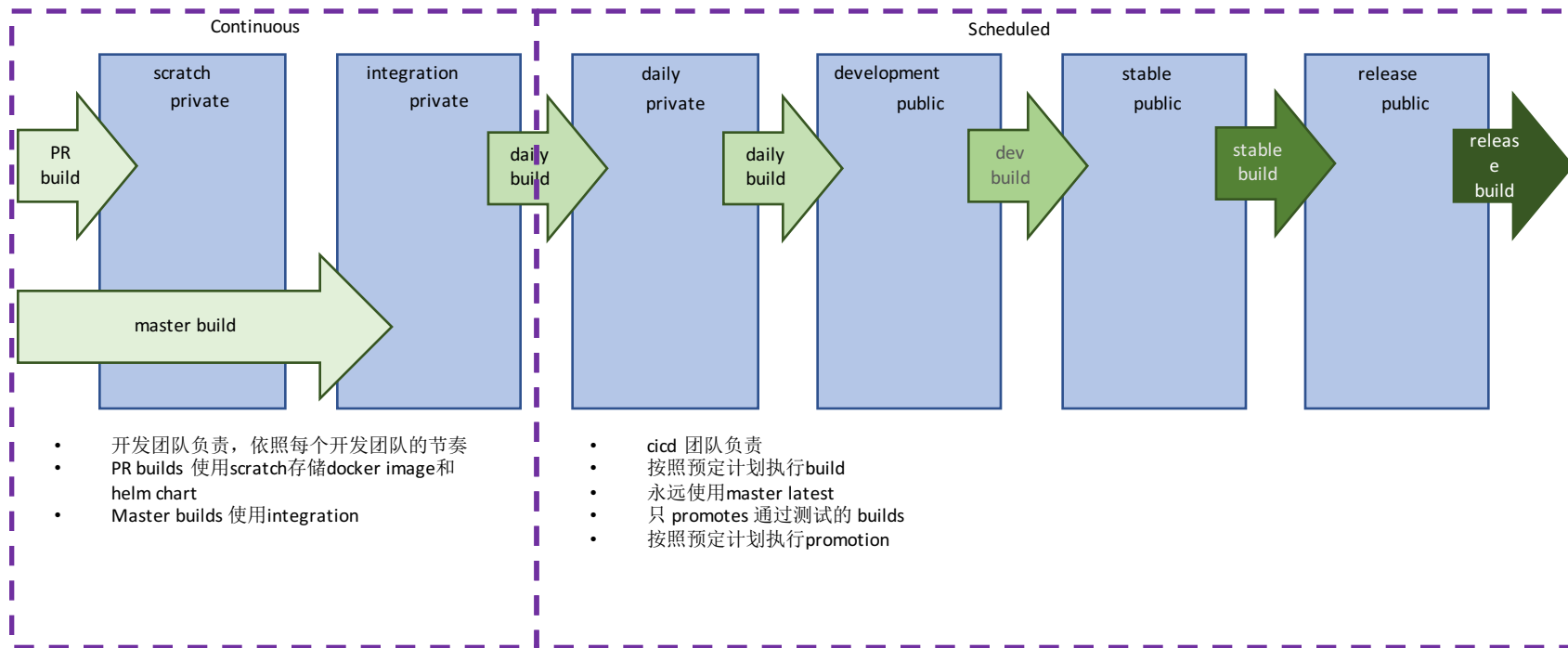


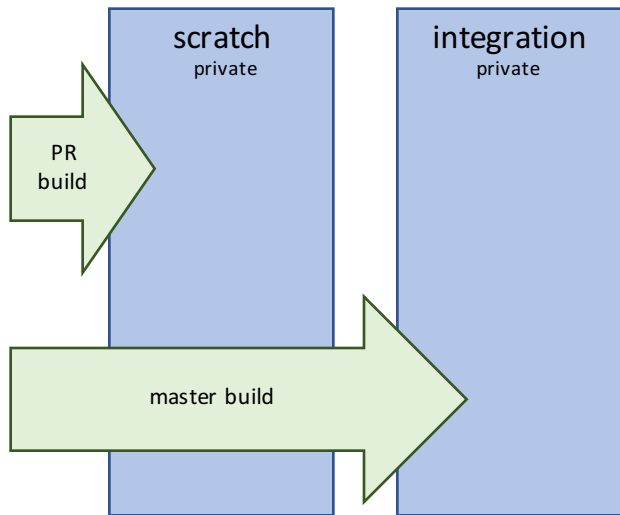
# Devops架构



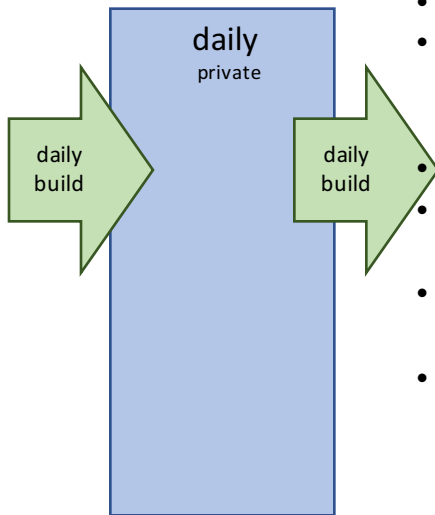


# Devops 2.0



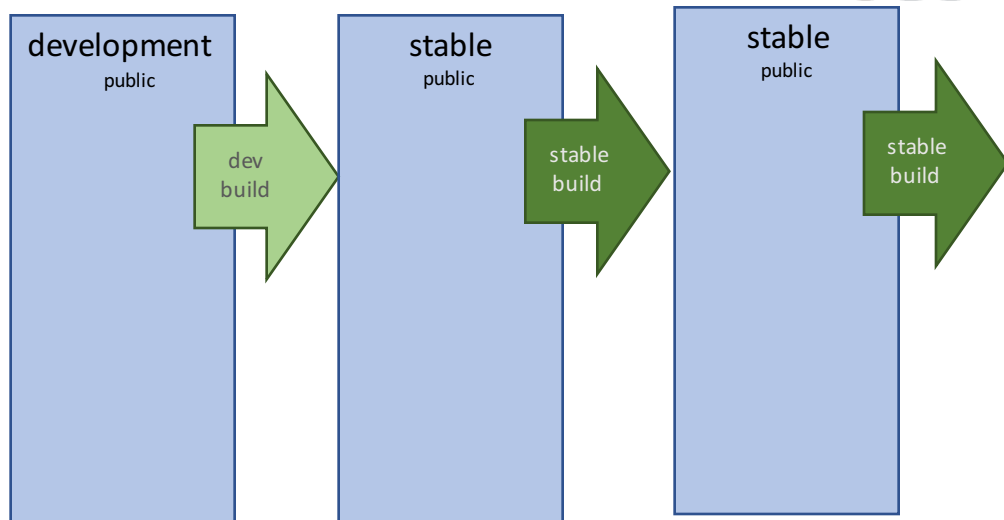


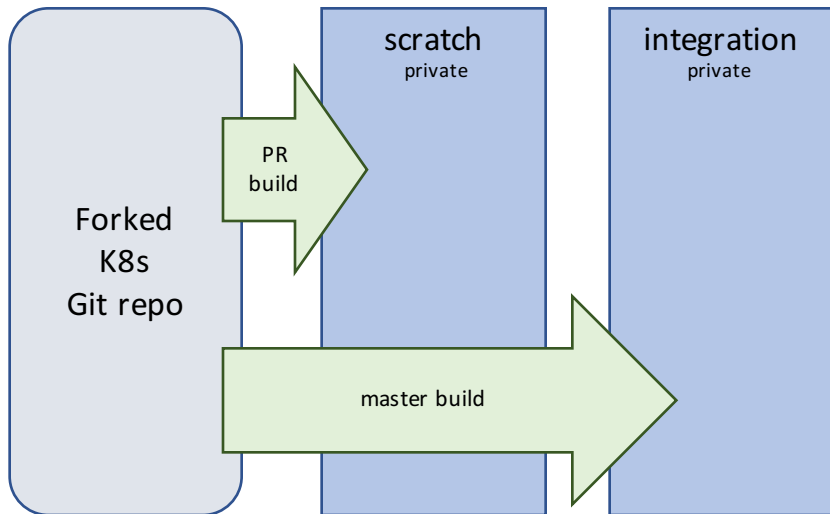
- Scratch 用于存放pr builds 的镜像和chart
- Integration 用于存放master build的镜像和chart
- PR & Master builds are the last builds that will produce artifacts (all subsequent builds are deployments of images/charts previously produced by master builds)
- 开发这只对scratch和integration有写权限
- Docker 镜像tag由git commit sha 或者 “latest”组成
- Helm charts 版本由 git commit sha组成



- Daily build 每日从integration拉取latest
- Daily builds 预期会经常失败
  - 开发团队件组件集成问题
  - 平台兼容性，服务和API升级
- 失败的daily builds 需要在integration中修复
- 通过的daily build会标明所有的的镜像和chart来promote到下一个阶段
- **Daily builds** 执行跨平台，跨操作的系统和云平台的所有测试案例
- **Daily builds** 执行所有开发团队定义的功能测试案例并记录失败日志

- Development stage 包含所有最新的功能
- Development stage 根据daily build情况每日更新
- Stable stage 代表所有完成并测试过的功能
- 使用stable stage的消费者可以从changelog获取所有特性
- Stable stage的特性，API等变化较小
- Release stage 代表所有发布的功能特性
- Release stage 确保稳定的API和特性
- Release stage根据发布周期更新

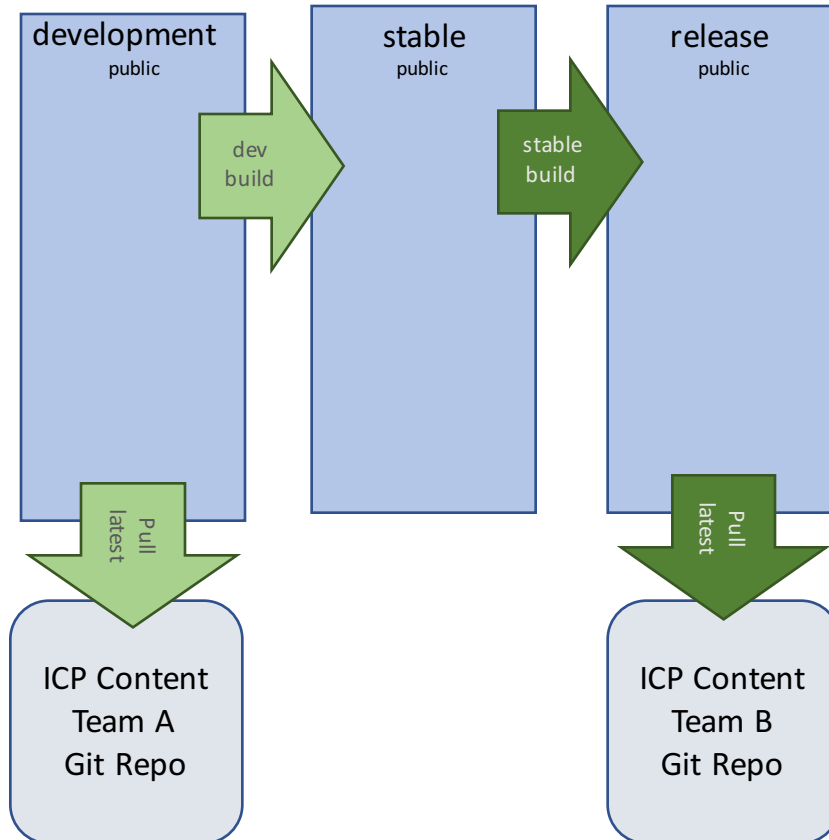




- 遵照 “promotion” 流程
- 可以使用forked repo的任意版本
- 可以通过pipeline构建ppc64le和其他环境的image

# 生产者

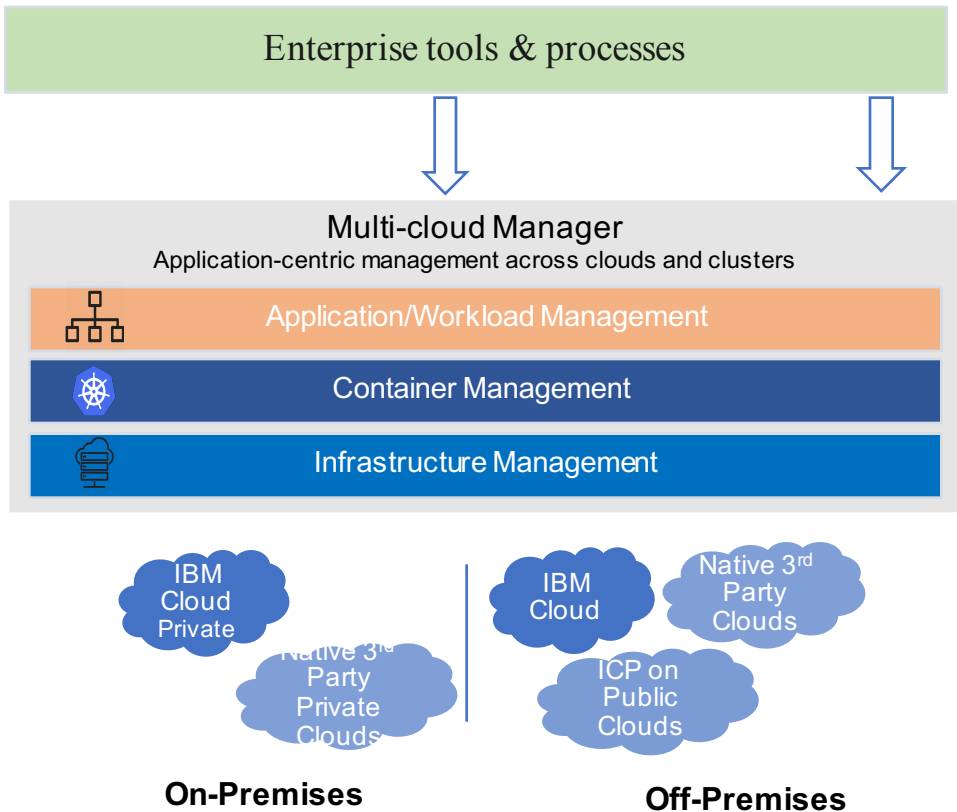
- 类似 Google Chrome's Stream Chooser
- 每个消费者（开发团队）根据稳定性等因素选择特定的版本
- 方便消费者快速的反馈
- 方便消费者快速使用新的功能



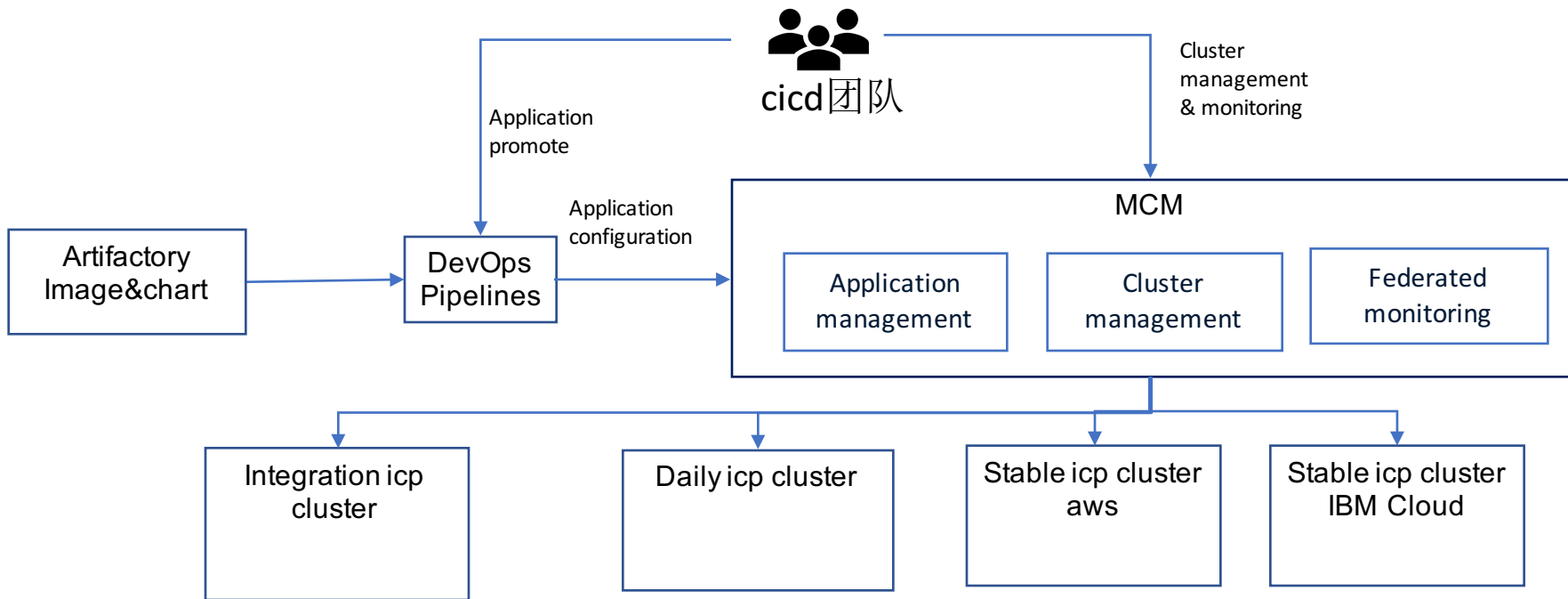
# 多云Devops



- 跨kubernetes集群promotion
  - integration集群测试
  - Daily集群部署daily build
  - Stable集群部署stable build
- 多云管理系统multiple-cloud manager
  - 定义多云环境的应用
  - 定义应用在每个集群的配置
  - 定义应用的placement



# 多集群devops







# Thanks

DevOps 时代社区 荣誉出品

想第一时间看到高效运维社区的  
最新动态吗？

