

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）

携程持续交付与构建平台实践

周光明 携程

目录

**1****携程持续交付简介****2****统一构建平台设计****3****Jenkins on K8s 实践****4****问题与改进**

携程持续交付背景



8000+ 应用



3000+ 研发



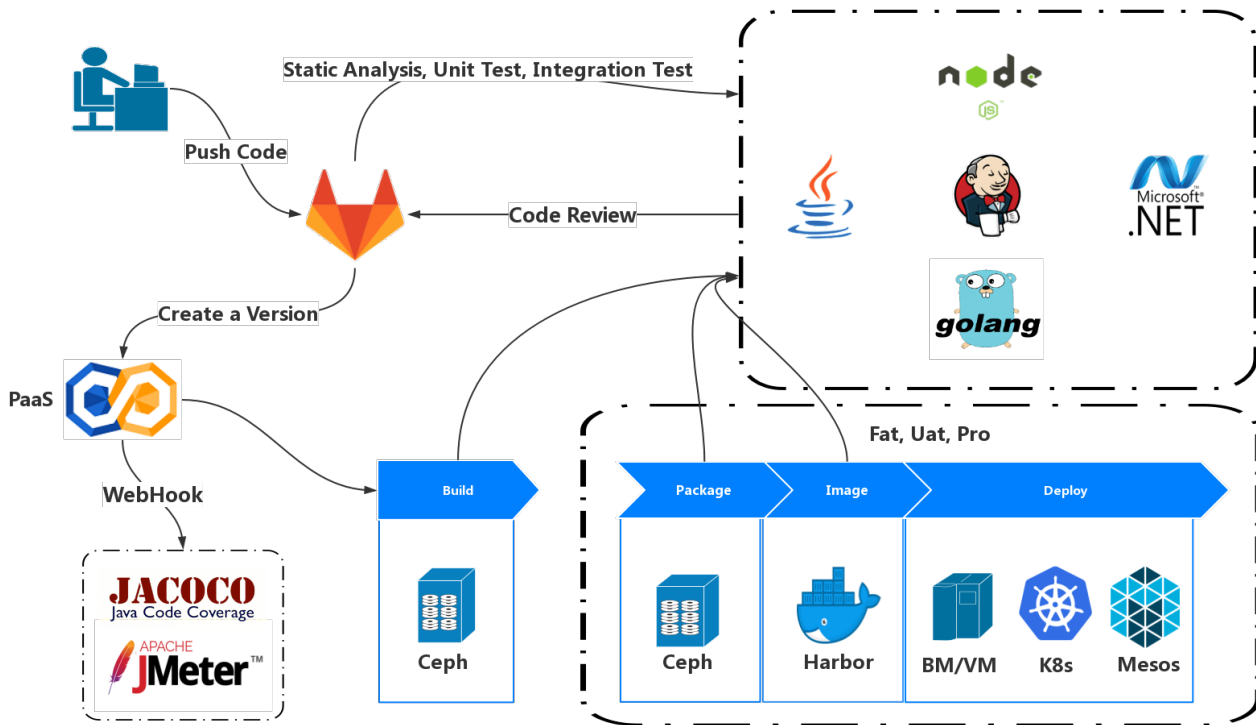
6000+ 发布/天

持续交付的意义

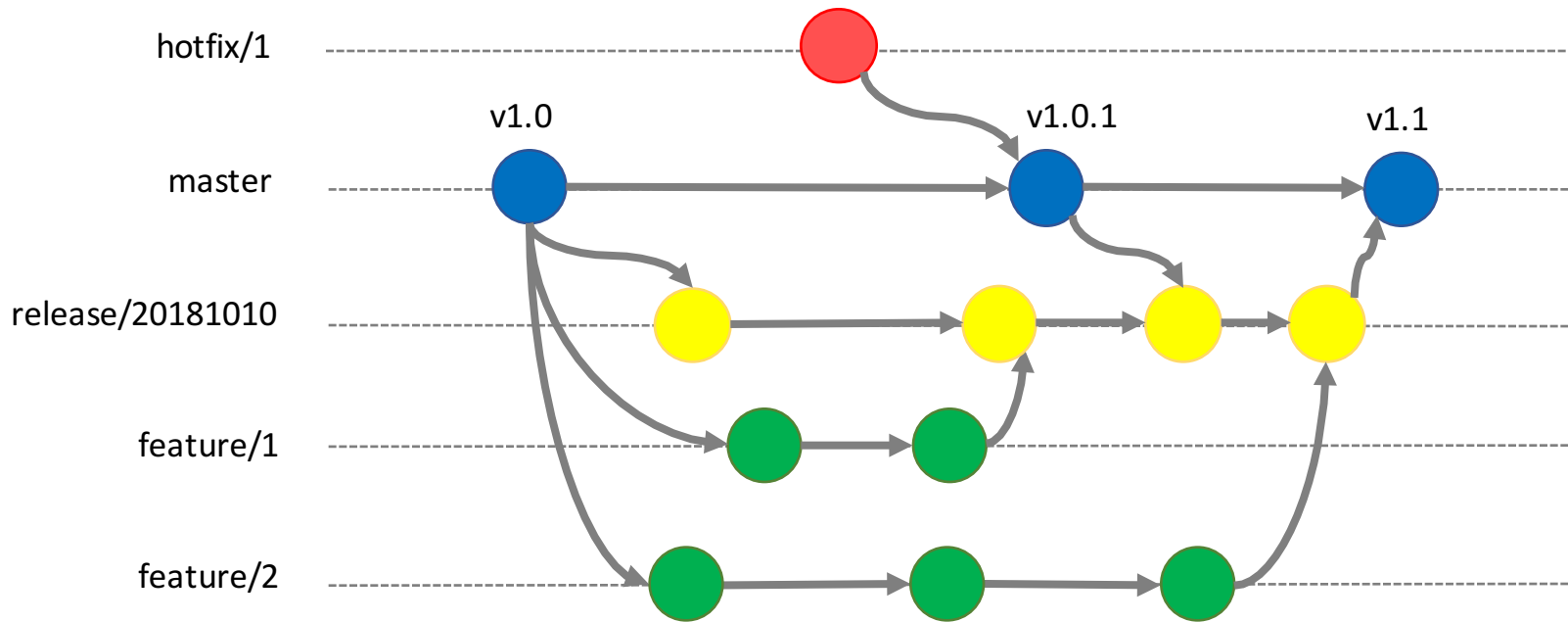
- 效率提升
- 质量保障
- 安全可靠
- 团队协作
- 流程透明



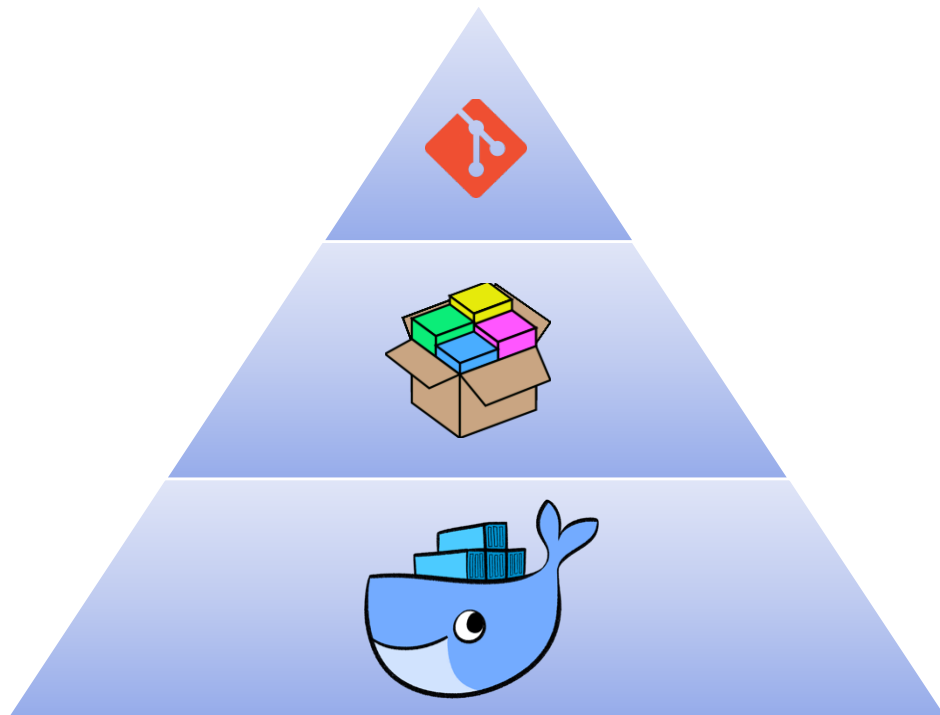
持续交付流程



分支模型管理



什么是版本?



Git Tags

源码

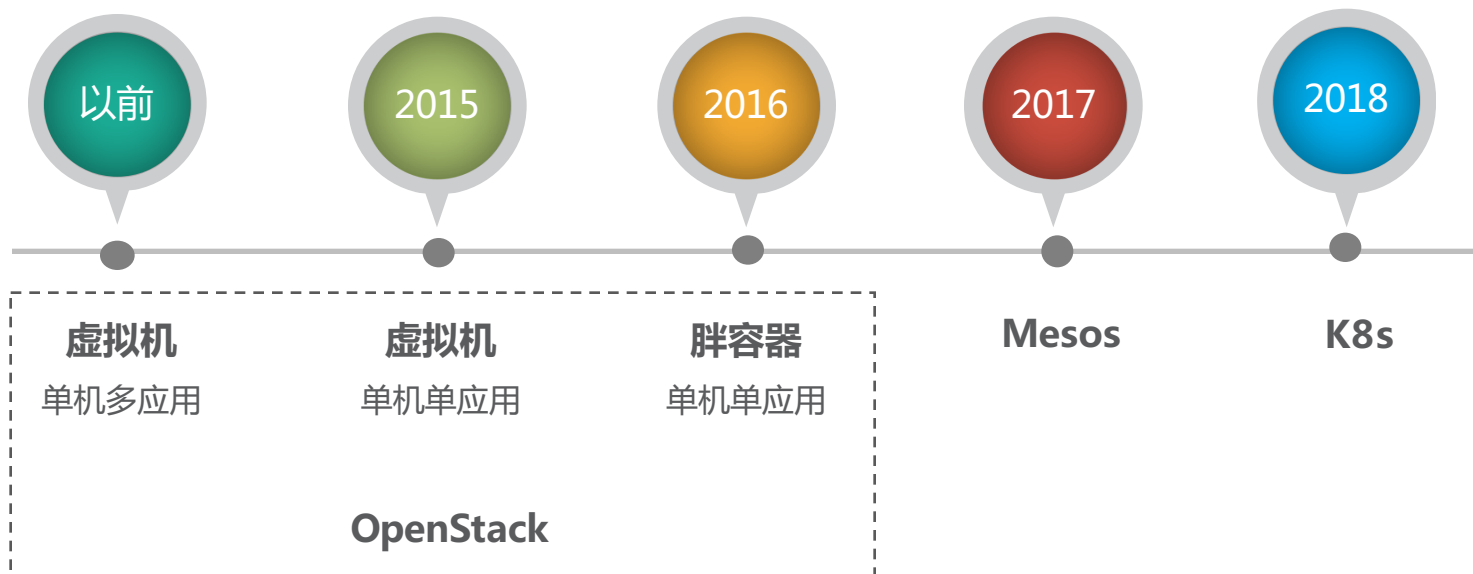
Packages

源码 + 代码依赖

Images

源码 + 代码依赖 + 环境依赖

部署模型演进



部署过程

- Group

一组暴露同一服务的集合

- 拉入拉出

Group 中的某个成员是否接收请求

- 堡垒机

生产环境 Group 中第一台被发布验证的机器

- 点火

应用初始化、预热、加载数据等

- 分批

同一 Group 分成多个批次进行滚动部署

- 降级

部分依赖系统发生故障后发布系统可忽略

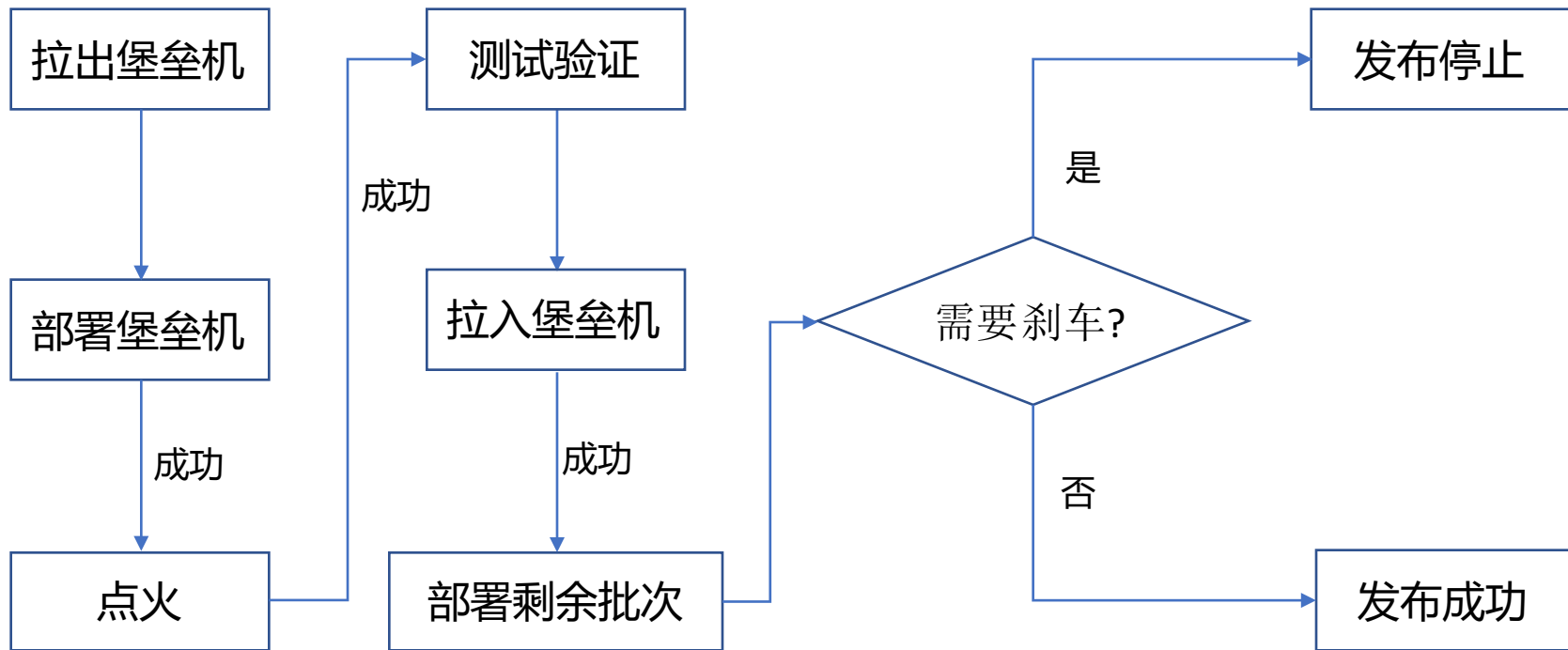
- 刹车

当发布失败机器数量大于一定比例则停止发布

- 回滚

稳定的符合预期的回滚逻辑

部署过程



资源管理

- 多环境

测试环境, 生产环境

- 多类型

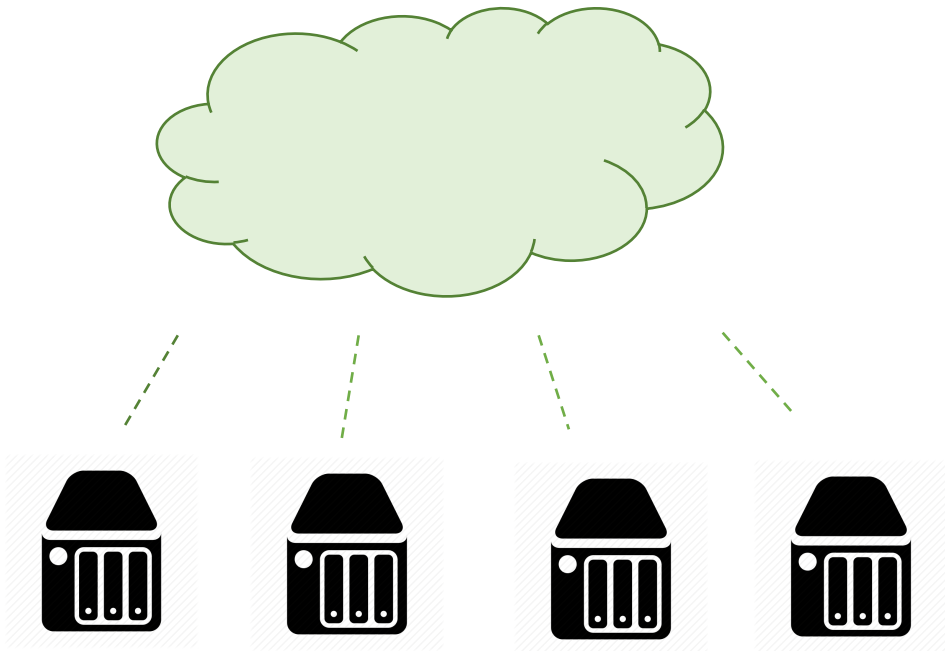
Docker, VM, BM

- 多平台

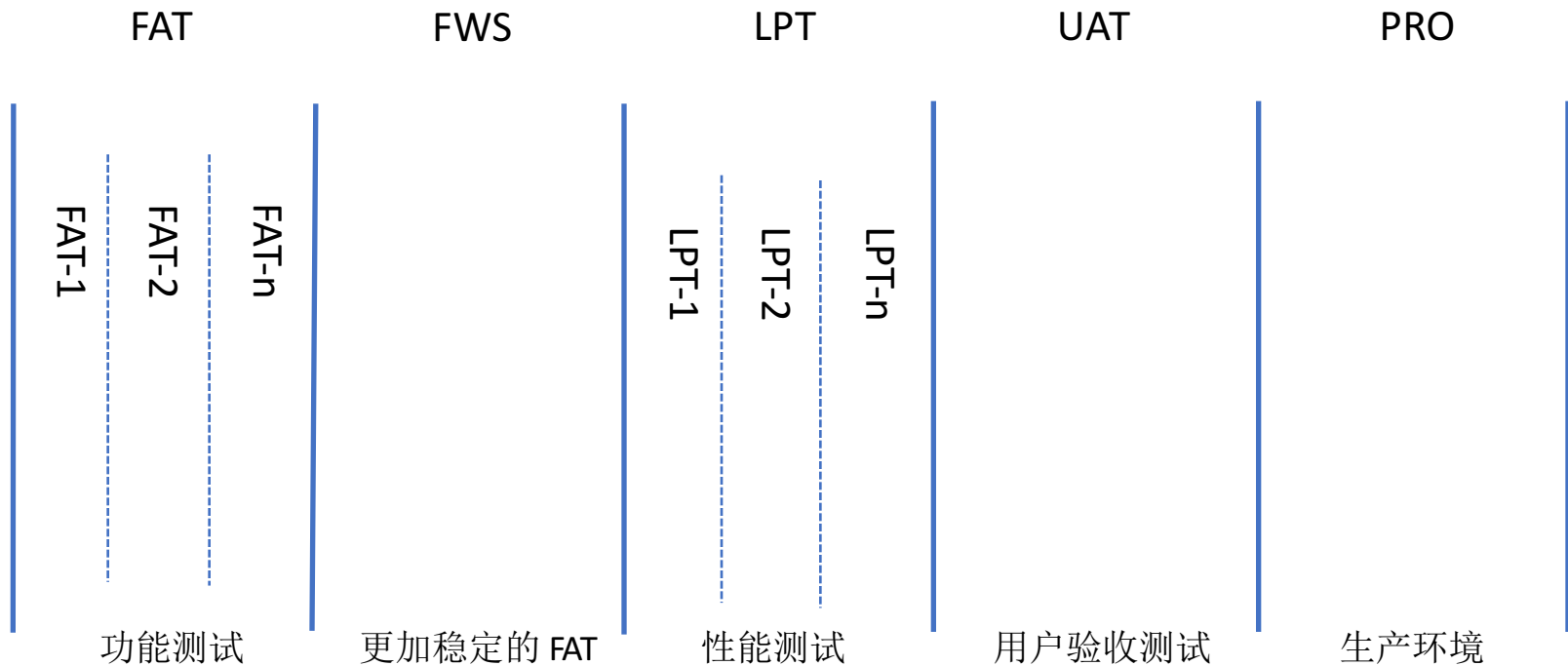
OpenStack, Mesos, K8s

- 多数据中心

私有云, 公有云



环境管理



目录

1 携程持续交付简介



2 统一构建平台设计

3 Jenkins on K8s 实践

4 问题与改进

Jenkins 的优势

- 开箱即用
- 成熟稳定
- 插件丰富
- 社区活跃
- 强大的 Pipeline
-



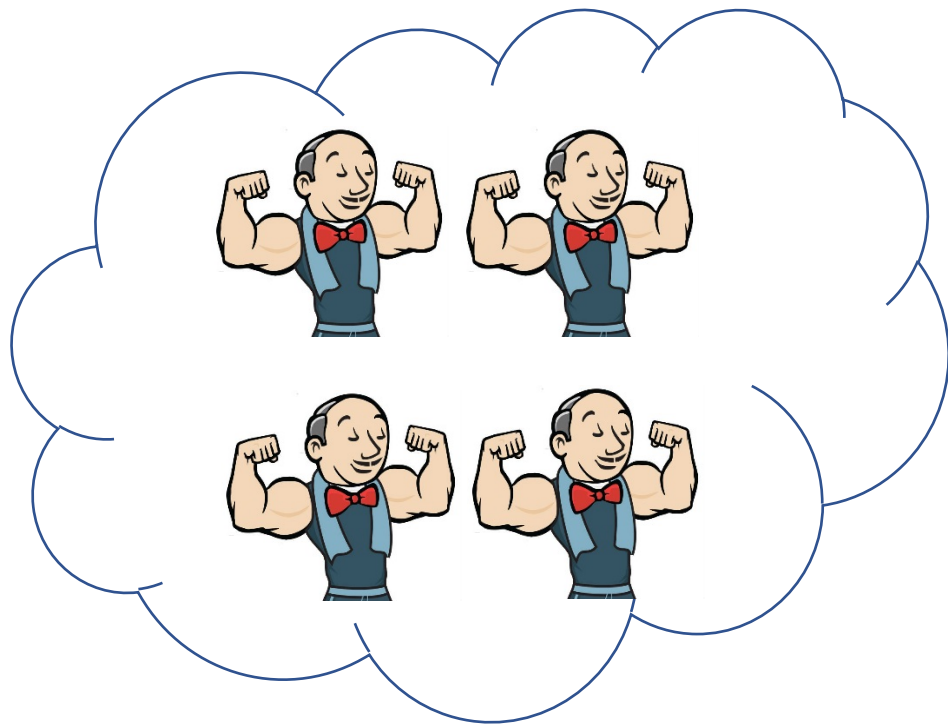
Jenkins 的不足



单点故障与单机性能

Jenkins Master HA 方案

- Keepalived + Virtual IP
- Gearman Plugin
- Jenkins on Mesos/K8s
- CloudBees
-



Master 拆分策略

	By environment	By org chart	By product lines
Pros	<p>Can tailor plugins on masters to be specific to that environment's needs</p> <p>Can easily restrict access to an environment to only users who will be using that environment</p>	<p>Can tailor plugins on masters to be specific to that team's needs</p> <p>Can easily restrict access to a division's projects to only users who are within that division</p>	<p>Entire flows can be visualized because all steps are on one master</p> <p>Reduces the impact of one master's downtime on only affects a small subset of products</p>
Cons	<p>Reduces ability to create pipelines</p> <p>No way to visualize the complete flow across masters</p> <p>Outage of a master will block flow of all products</p>	<p>Reduces ability to create cross-division pipelines</p> <p>No way to visualize the complete flow across masters</p> <p>Outage of a master will block flow of all products</p>	<p>A strategy for restricting permissions must be devised to keep all users from having access to all items on a master.</p>

来源 <https://jenkins.io/doc/book/architecting-for-scale/>

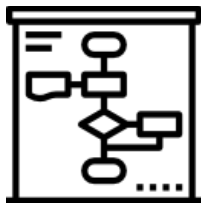
■ 预估 Job, Master 和 Executors 的数量

- Number of Jobs = Number of Developers * 3.333
- Number of Masters = Number of Jobs/500
- Number of Executors = Number of Jobs * 0.03

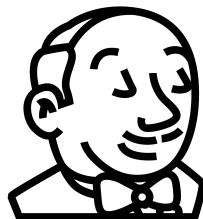


来源 <https://jenkins.io/doc/book/architecting-for-scale/>

构建系统现状



20000+ Jobs



20+ Masters



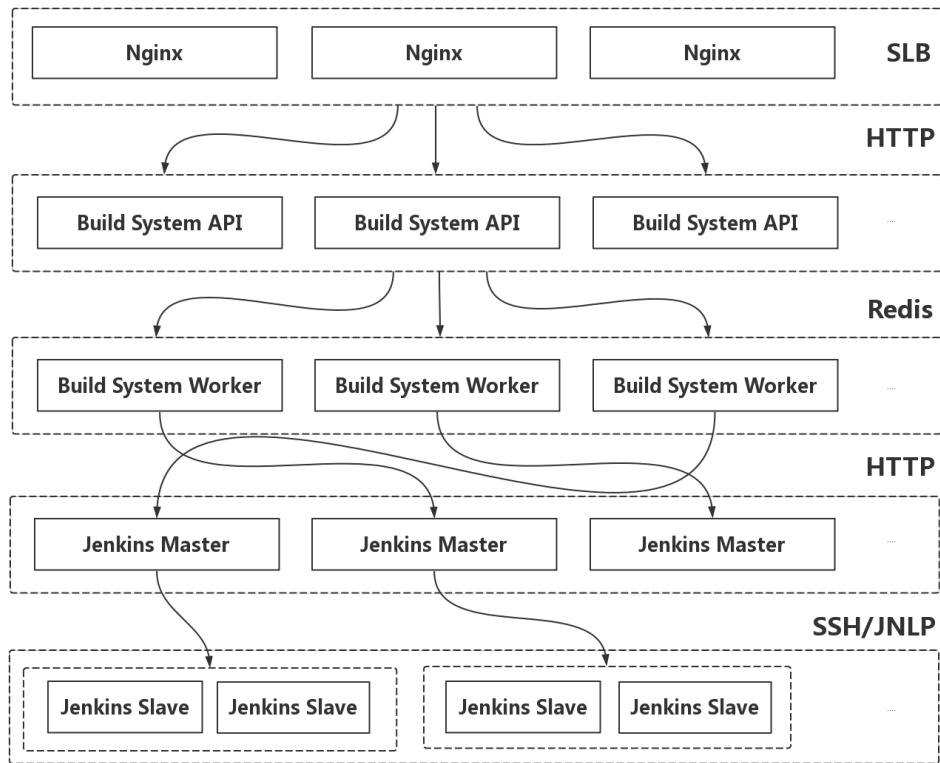
20+ 类型



12000+ 构建/天

自己动手丰衣足食

构建系统架构



API 层负责接收各类
构建请求

Worker 层负责 Job
调度

构建系统特点

- Job Division Strategies

队列按照 app 或者 project 进行分割

- Job Template

每种 Job 都由数据库中的队列模板动态创建与更新，队列模板支持版本化的提交，可进行灰度发布与快速回滚

- Labeling

Job 与 Master 都带有标签，Job 可以通过标签找到符合条件的 Master

- Sharding

每个 Master 可以配置容量，新建 Job 时按照 Master 容量配置比例进行 Sharding

- Failover

每个 Job 至少有两台满足条件的 Master，构建系统实时对 Master 进行健康监测，故障时 Job 自动转移到其他健康的 Master

监控与告警

- 操作系统层面

CPU, Mem, Disk, Network 等指标

- 应用系统层面

API, Worker 与 Jenkins 的可用性与异常情况

- 业务逻辑层面

构建队列是否有堵塞

系统容量是否到达瓶颈

Pipeline 关键 Step 是否超时

The image displays four screenshots of Jenkins email notifications, arranged in a 2x2 grid. Each notification is from 'build@ctrip.com' and includes a recipient address.

- Top Left:** Dated 2018/10/15 (周一) 15:37. Subject: '[BuildPortal] Jenkins 1 [redacted] for job type [redacted]'. The body shows build statistics: 'Current Enqueued Builds Count: 2', 'Current Enqueued on Master Builds Count: 0', 'Current Running on Master Builds Count: 11', and 'Max Building on Master Builds Count: 10'.
- Top Right:** Dated 2018/10/2 (周二) 0:45. Subject: '[BuildPortal] Slave fq-windows-container-11 on 10.[redacted] is abnormal --- production'. The body contains a JSON snippet:

```
{
  "_class": "hudson.slaves.SlaveComputer",
  "actions": [
    ],
    "displayName": "fq-windows-container-11",
    "executors": [
      {
      }
    ]
  },
  ],
```
- Bottom Left:** Dated 2018/10/15 (周一) 18:24. Subject: '[BuildPortal] Notification --- production'. The body shows the message: 'job master server: 1 [redacted] has been membered out'.
- Bottom Right:** Dated 2018/10/12 (周五) 17:57. Subject: '[BuildPortal] Notification --- production'. The body shows the message: 'Current enqueued builds 11'.

构建系统界面

Build				Build ID	vzgm周光明
				当前排队数量: 0 当前运行数量: 0	
Online	Healthy	http://192.168.1.100:80	erlang, go, lang, ares, python, php, java_sonar, jar,	构建队列数量: 460 历史构建数量: 209123 当前排队数量: 0 当前运行数量: 0	编辑 监控 删除 k8s
Online	Healthy	http://192.168.1.100:80	python, php, erlang, go, lang, ares, java_sonar, jar,	构建队列数量: 680 历史构建数量: 161189 当前排队数量: 0 当前运行数量: 0	编辑 监控 删除 k8s
Online	Healthy	http://192.168.1.100:80	java, nodejs, java_sonar, java_mr, jar, python, ares,	构建队列数量: 470 历史构建数量: 120712 当前排队数量: 0 当前运行数量: 0	编辑 监控 删除 k8s
Online	Healthy	http://192.168.1.100:80	java, nodejs, jar, java_sonar, java_mr, ares,	构建队列数量: 530 历史构建数量: 170083 当前排队数量: 0 当前运行数量: 0	编辑 监控 删除 k8s
Offline	Healthy	http://192.168.1.100:80		构建队列数量: 0 历史构建数量: 290 当前排队数量: 0 当前运行数量: 0	编辑 监控 删除
Online	Healthy	http://192.168.1.100:80	java_sonar, java_mr, ares_sonar,	构建队列数量: 545 历史构建数量: 160907 当前排队数量: 0	编辑 监控 删除

目录

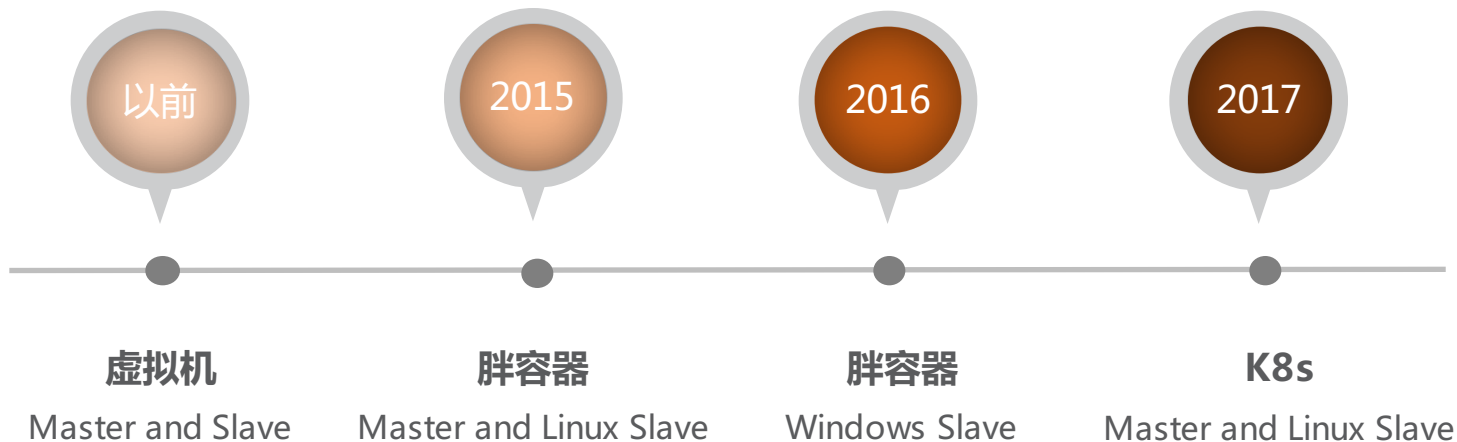
1 | 携程持续交付简介

2 | 统一构建平台设计

➔ **3** | Jenkins on K8s 实践

4 | 问题与改进

Jenkins 集群演进

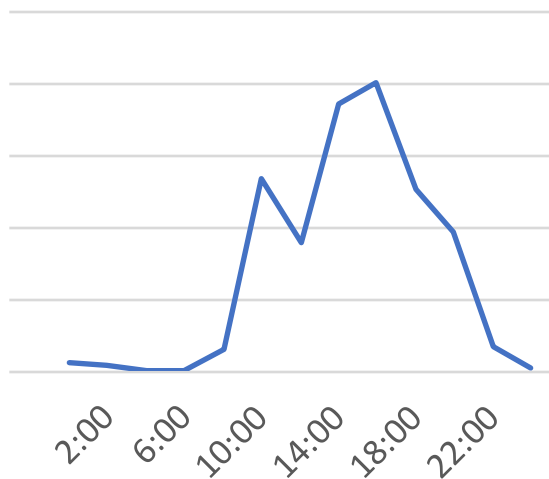




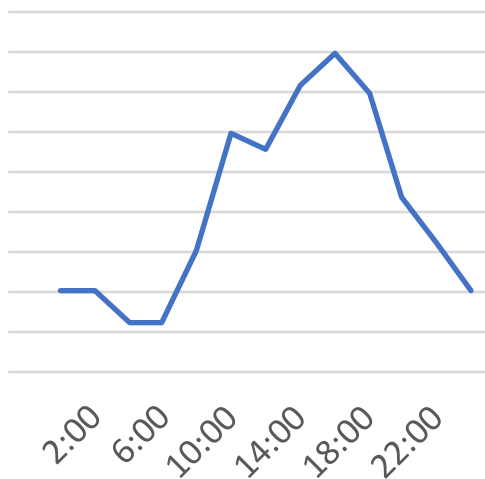
DevOps 国际峰会 2018·深圳站

Slave 弹性调度

构建数量



容器数量



根据构建的频繁程度, 为不同类型的 Slave 配置不同的 PodIdleMinutes 参数控制 Slave 保留时间

Slave Provision 时间优化

```
-Dhudson.slaves.NodeProvisioner.initialDelay=0 //Default 100
-Dhudson.model.LoadStatistics.decay=0.5 //Default 0.9
-Dhudson.slaves.NodeProvisioner.MARGIN=50 //Default 10
-Dhudson.slaves.NodeProvisioner.MARGIN0=0.85 //Default 0.5
-Dhudson.slaves.NodeProvisioner.MARGIN_DECAY=0.5 //Default 0.5
```

initialDelay: Jenkins 启动留给静态 Slave 连接的时间

decay: Jenkins 负载统计公式 “指数移动平均值(EMA)” 中的平滑指数:

$$history[0] * decay + newData * (1 - decay)$$

判断是否应该创建 Slave:

$$excessWorkload > 1 - m \quad (excessWorkload = qlen - plannedCapacity - connectingCapacity)$$

m 的计算公式:

$$MARGIN + (MARGIN0 - MARGIN) * Math.pow(MARGIN_DECAY, totalSnapshot)$$

Slave Provision 时间优化

经常有 Slave 的创建时间超过 5 分钟!



2018/10/22 (周一) 1:34

build@ctrip.com

[BuildPortal] Schedule slave for k8s job: java-scheduler exceed 5 minutes --- production

收件人

Kubernetes slave 调度时间超过五分钟，编译终止

Build ID: 4126227

Job Name: java-scheduler

Job Type: java

[查看详情](#)

遍历 Labels 运行 Provision 轮训逻辑, 而第一次新建 Slave 时不会更新 Labels 集合

```
@Override
protected void doRun() {
    Jenkins h = Jenkins.getInstance();
    h.unlabeledNodeProvisioner.update();
    for( Label l : h.getLabels() )
        l.nodeProvisioner.update();
}
```

jenkins: core/src/main/java/hudson/slaves/NodeProvisioner.java#861

Jenkins 每隔 5 分钟更新一次 Labels 集合

```
Timer.get().scheduleAtFixedRate(new SafeTimerTask() {
    @Override
    protected void doRun() throws Exception {
        trimLabels();
    }
}, TimeUnit2.MINUTES.toMillis(5), TimeUnit2.MINUTES.toMillis(5), TimeUnit.MILLISECONDS);
```

jenkins: core/src/main/java/jenkins/model/Jenkins.java#966

在插件创建 Slave 时主动调用 reset 方法

```
for (LabelAtom label : newTemplate.getLabelSet()) {
    if (label.isEmpty()) {
        LOGGER.log(Level.INFO, "Reset label {0}",
            new Object[] { label.getName() });
        Class cls = LabelAtom.class;
        Method method = cls.getSuperclass().getDeclaredMethod("reset");
        method.setAccessible(true);
        method.invoke(label);
    }
}
```

kubernetes-plugin: src/main/java/org/csanchez/jenkins/plugins/kubernetes/pipeline/PodTemplateStepExecution.java#start

Workspace 保留问题

为什么?

用户需要在 Slave 被回收后依然可以在 Jenkins 上面查看 Workspace, 并且可以减少源码下载次数

如何实现?

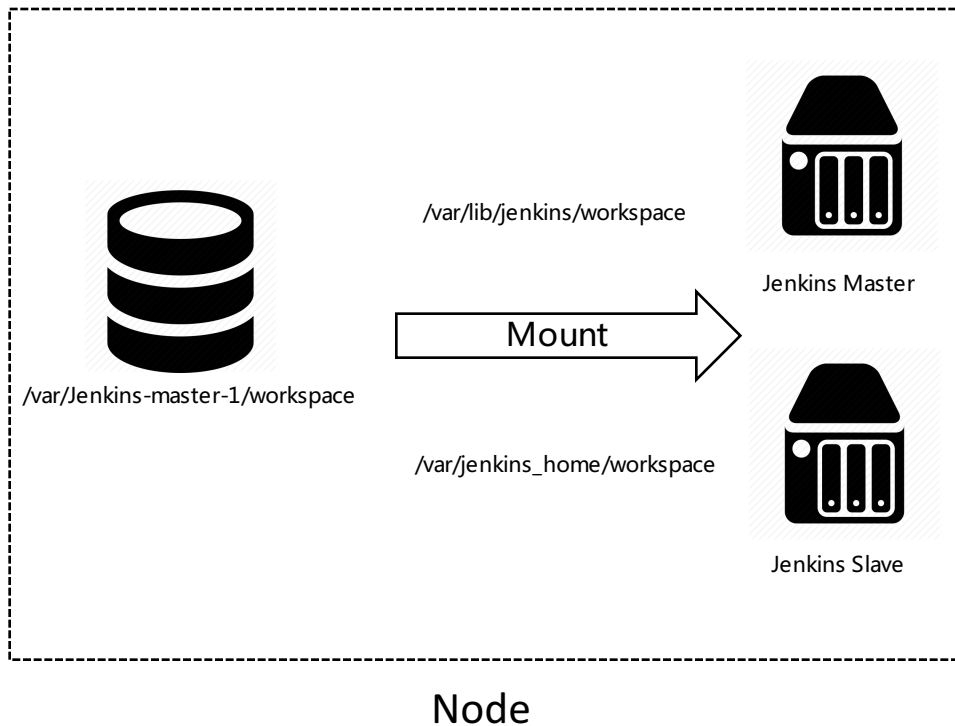
让 Slave 与 Master 在同一个 Node(nodeSelector) 且共享同一 Workspace, 通过 Master 查看 Workspace

遇到问题?

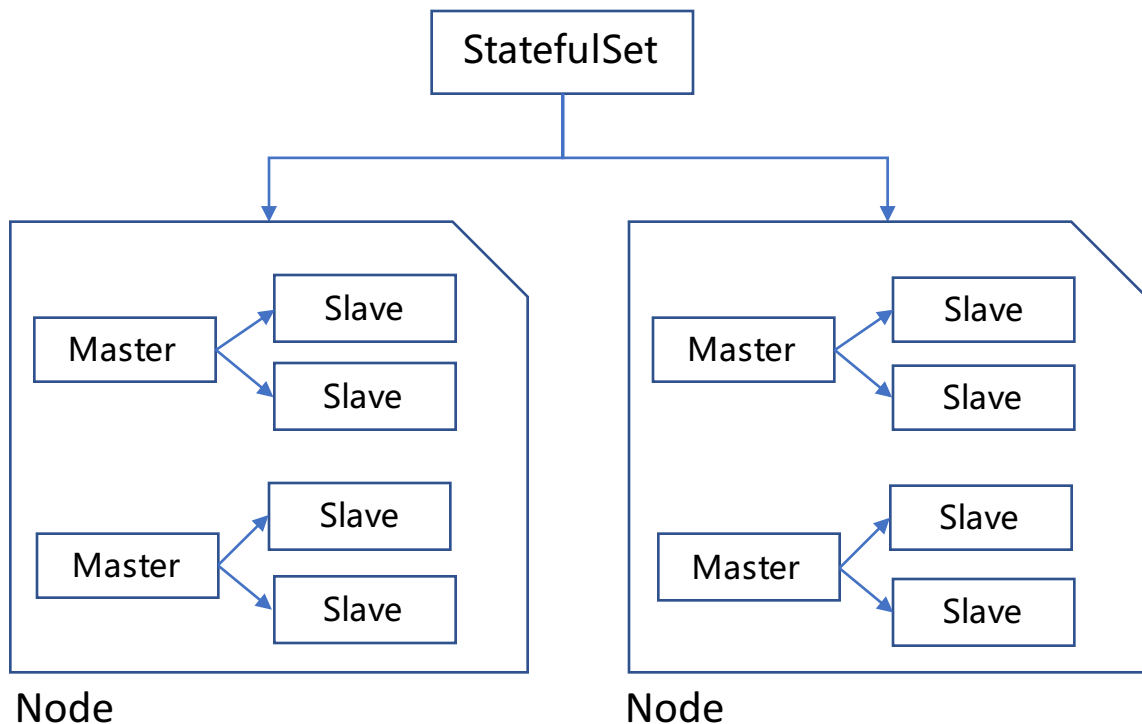
一个 Job 在同一时间只能在一个 Master 中运行

如何解决?

构建系统实现同一个 Job 对于多个 Master 并发构建的调度



StatefulSet 管理 Master



使用 StatefulSet 管理
Jenkins Master 集群降低
维护成本

Sticky Scheduler

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  namespace: build
  name: jenkins-master
  annotations:
    jenkins-master-0: svr01
    jenkins-master-1: svr01
    jenkins-master-2: svr01
    jenkins-master-3: svr02
    jenkins-master-4: svr02
    jenkins-master-5: svr02
spec:
  serviceName: jenkins-master
  replicas: 6
  selector:
    matchLabels:
      app: jenkins-master
```

让 Master Pod 固定分配到指定的 Node 上

CHostpath Volume Driver

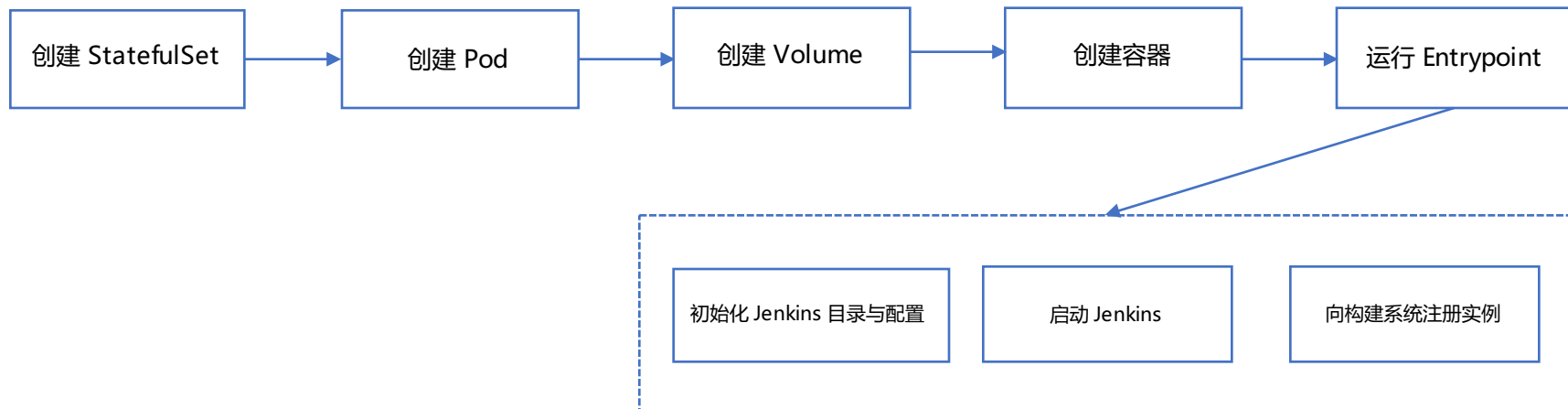
```
deploy@ubuntu:/var/lib/k8s/storage$ tree
```

```
.
├── chostpath
│   ├── build
│   │   └── StatefulSet
│   │       ├── jenkins-master
│   │       │   └── Pod
│   │       │       ├── jenkins-master-1
│   │       │       │   ├── jenkins-home
│   │       │       ├── jenkins-master-2
│   │       │       │   ├── jenkins-home
│   │       │       └── jenkins-master-3
│   │       │           └── jenkins-home
```

按照规则自动创建 Pod
的挂载目录

```
/var/lib/k8s/storage/chostpath/${NAMESPACE}/${RESOURCE_TYPE}/${RESOURCE_NAME}/Pod/${POD_NAME}/${VOLUME_NAME}
```

Jenkins Master 启动过程



目录

1 携程持续交付简介

2 统一构建平台设计

3 Jenkins on K8s 实践

 **4** 问题与改进

I 问题与改进

- 多环境应用镜像

多个环境使用同一个应用镜像, 减少环境差异, 提升发布效率

- 非标准应用容器化

支持用户自定义的 Dockerfile, 运行构建与制作镜像, 减少运维成本

- 资源混部

构建资源与其他资源可以混合部署, 提升资源利用率



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？

