

2018 | 中国·北京站
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 (OSCAI)

主办单位： DevOps时代

 高效运维社区
GreatOPS Community

2018年6月29日-30日

地址：北京悠唐皇冠假日酒店

K8S功能实践

郭宏泽 技术总监

目录

- ➔ **1** K8S自动化部署
- 2** K8S核心概念
- 3** K8S功能实践

Kubernetes 介绍

DOIS

- 2014年，Google发起了开源容器集群管理系统Kubernetes，原生支持了Docker。
- Kubernetes与较早的集群管理系统Mesos和YARN相比，对容器尤其是Docker的支持更加原生，同时提供了更强大的机制实现资源调度，自动管理容器生命周期，负载均衡，高可用等底层功能，使开发者可以专注于开发应用。
- Kubernetes是为生产环境而设计的容器调度管理系统，对于负载均衡、服务发现、高可用、滚动升级、自动伸缩等容器云平台的功能要求有原生支持



docker



版本选择



组件	版本	说明
Kubernetes	1.10.2	主程序
Docker	18.03.0-ce	容器
Flannel	0.10.0	网络组件
Etcd	3.3.4	数据库
Kubernetes-Dashboard	1.8.3	界面
Coredns	1.0.6	DNS组件
Harbor	1.2.0+	私有镜像库
Heapster	1.5.1	监控

使用心得

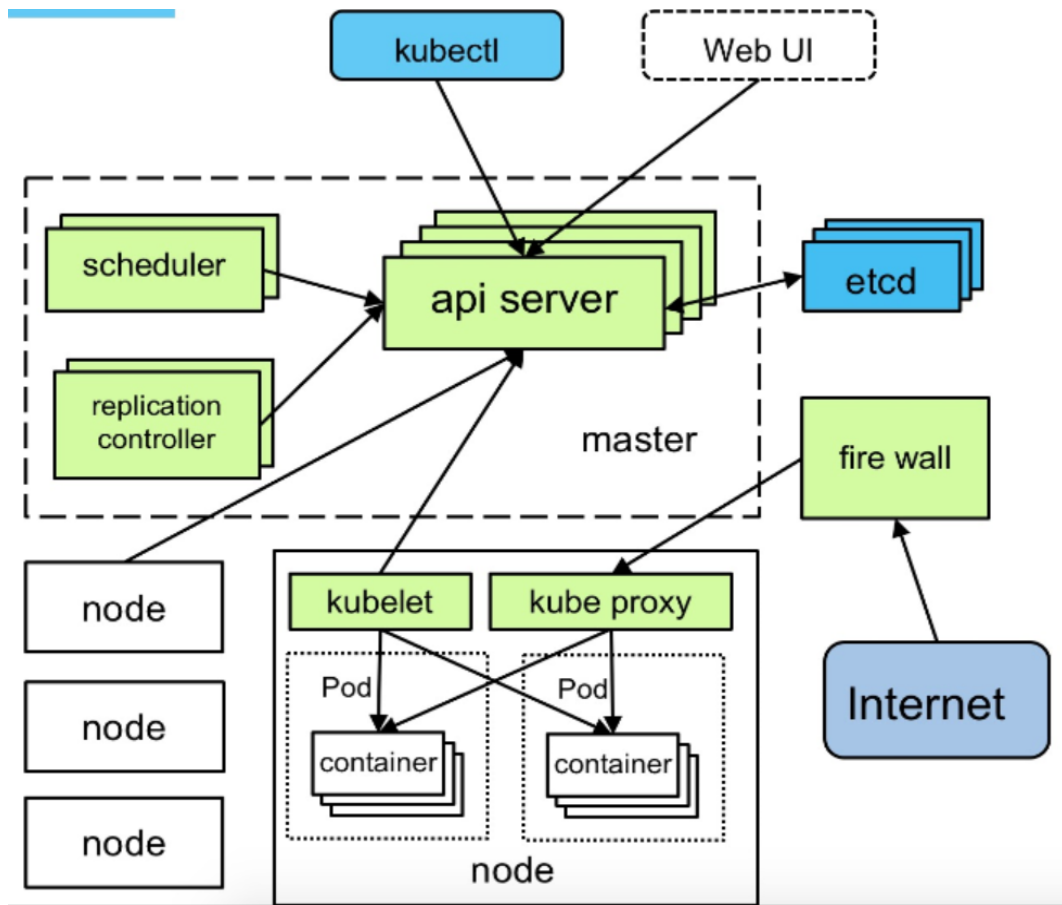
- K8s每个版本将会发布三种功能：
 - Alpha功能，预览功能，不稳定。
 - Beta功能，测试中的功能，不保证稳定。
 - 正式功能,已经经过测试的稳定功能，不保证无BUG。
- 尽可能减少对功能的使用，保持简单。
- 不要轻易升级你的生产系统。
- K8S的设计是面向基础平台，而非用户应用。

常用资源



- 官网
 - kubernetes.io
- 命令行参考
 - <https://kubernetes.io/docs/reference/generated/kubect/kubectl-commands#>
- Ansible 自动化部署
 - <https://github.com/gjmzj/kubeasz>
- SaltStack 自动化部署
 - <https://github.com/unixhot/salt-kubernetes>

生产部署



目录

1 K8S自动化部署

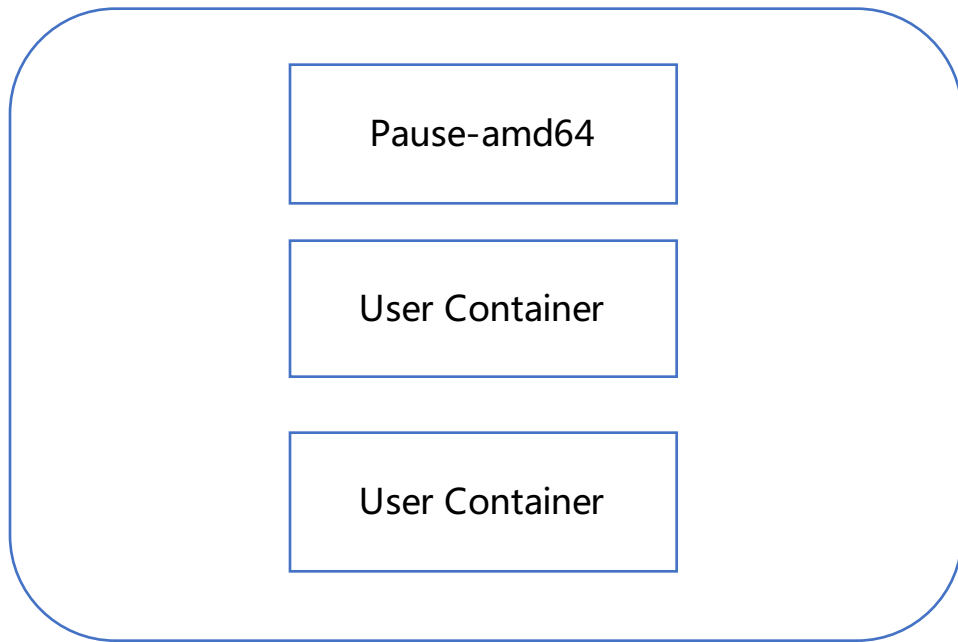


2 K8S核心概念

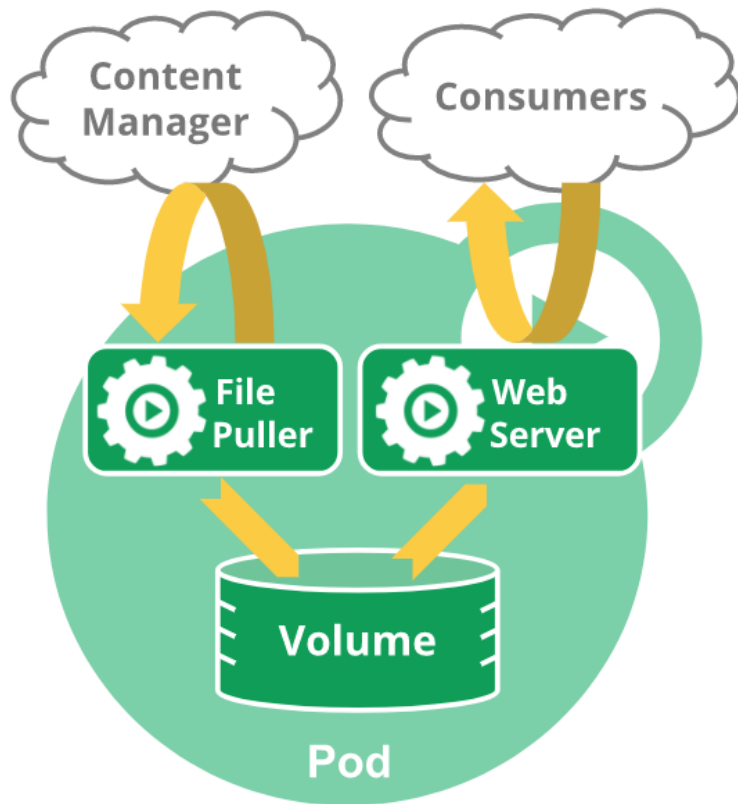
3 K8S功能实践

- Pod是K8s集群中所有业务类型的基础
- Pod是在K8s集群中运行部署应用或服务的最小单元，它是可以支持多容器的。
- Pod的设计理念是支持多个容器在一个Pod中共享网络地址和文件系统。
- POD控制器Deployment、Job、DaemonSet和PetSet

POD



POD

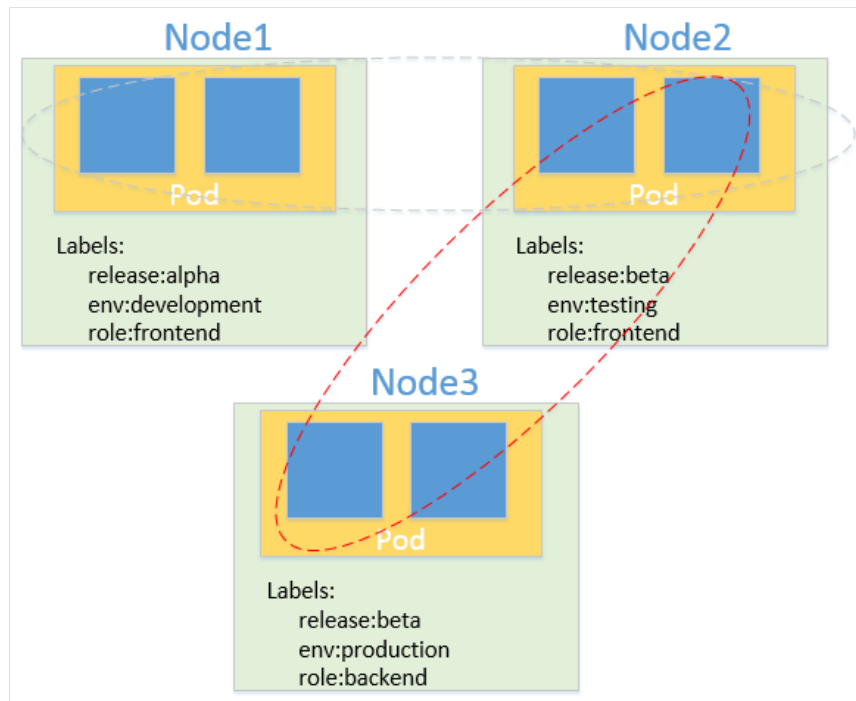


Label



- Label是一个 `key=value` 的键值对，由用户指定，可以附加到K8S资源之上。
- 给某个资源定义一个标签，随后可以通过label进行查询和筛选，类似SQL的where语句。
- Label可以给对象创建多组标签

Label



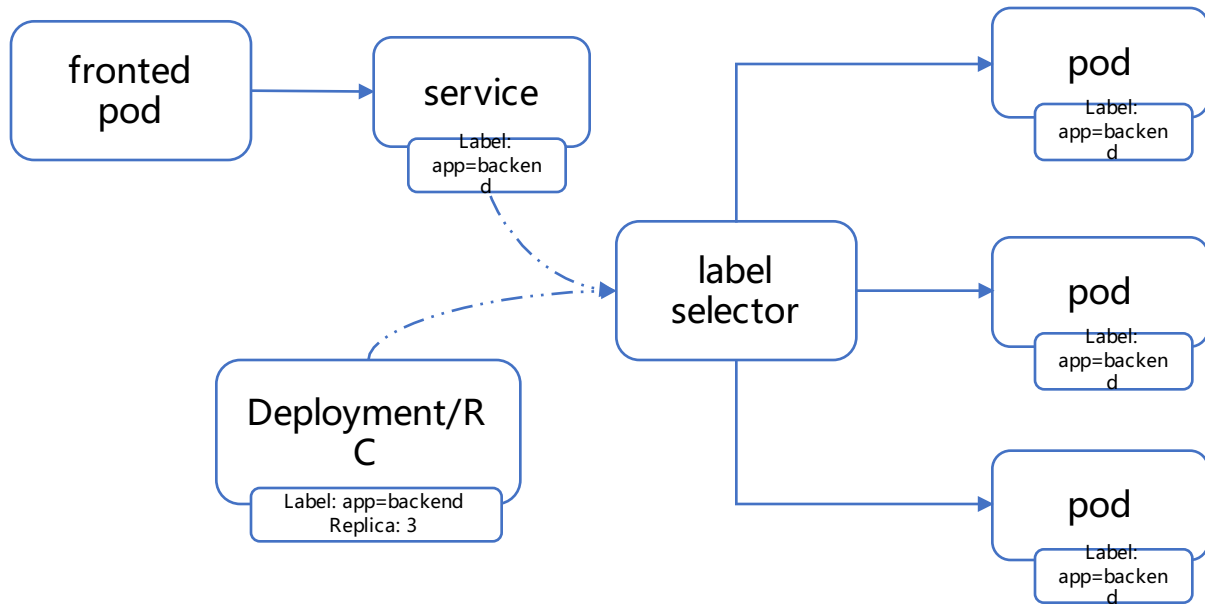
Deployment



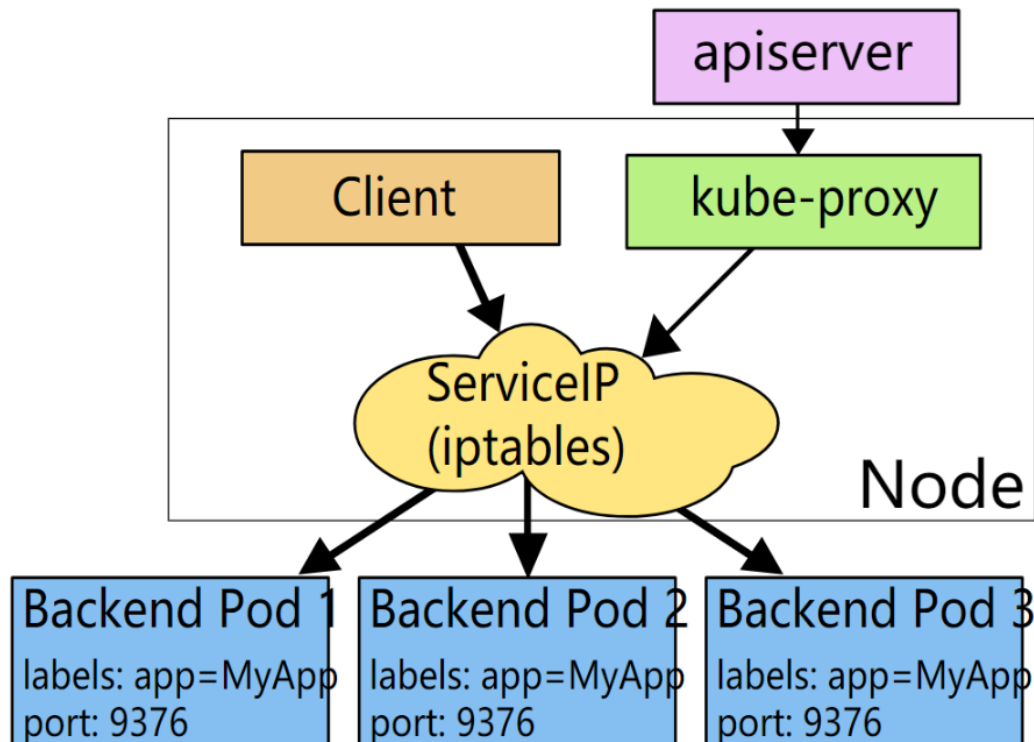
- 部署表示用户对K8s集群的一次更新操作。部署是一个比RS应用模式更广的API对象，
- 可以是创建一个新的服务，更新一个新的服务，也可以是滚动升级一个服务。滚动升级一个服务，实际是创建一个新的RS，然后逐渐将新RS中副本数增加到理想状态，将旧RS中的副本数减小到0的复合操作；
- 这样一个复合操作用一个RS是不太好描述的，所以用一个更通用的Deployment来描述。
- 以K8s的发展方向，未来对所有长期伺服型的业务的管理，都会通过Deployment来管理。
- Deployment是对RC的升级，与RC的相似度超过90%。

- RC、RS和Deployment只是保证了支撑服务的POD的数量，但是没有解决如何访问这些服务的问题。一个Pod只是一个运行服务的实例，随时可能在一个节点上停止，在另一个节点以一个新的IP启动一个新的Pod，因此不能以确定的IP和端口号提供服务。
- 要稳定地提供服务需要服务发现和负载均衡能力。服务发现完成的工作，是针对客户端访问的服务，找到对应的后端服务实例。
- 在K8集群中，客户端需要访问的服务就是Service对象。每个Service会对应一个集群内部有效的虚拟IP，集群内部通过虚拟IP访问一个服务。
- 在K8s集群中微服务的负载均衡是由Kube-proxy实现的。Kube-proxy是K8s集群内部的负载均衡器。它是一个分布式代理服务器，在K8s的每个节点上都有一个；
- 这一设计体现了它的伸缩性优势，需要访问服务的节点越多，提供负载均衡能力的Kube-proxy就越多，高可用节点也随之增多。

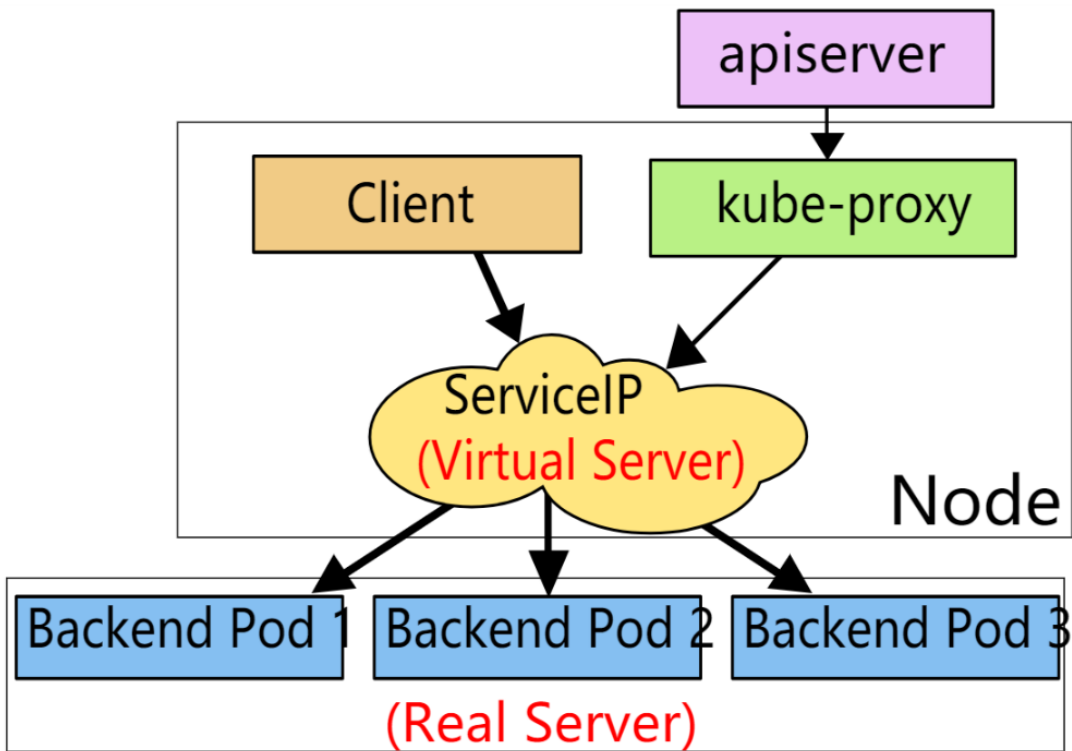
服务调用



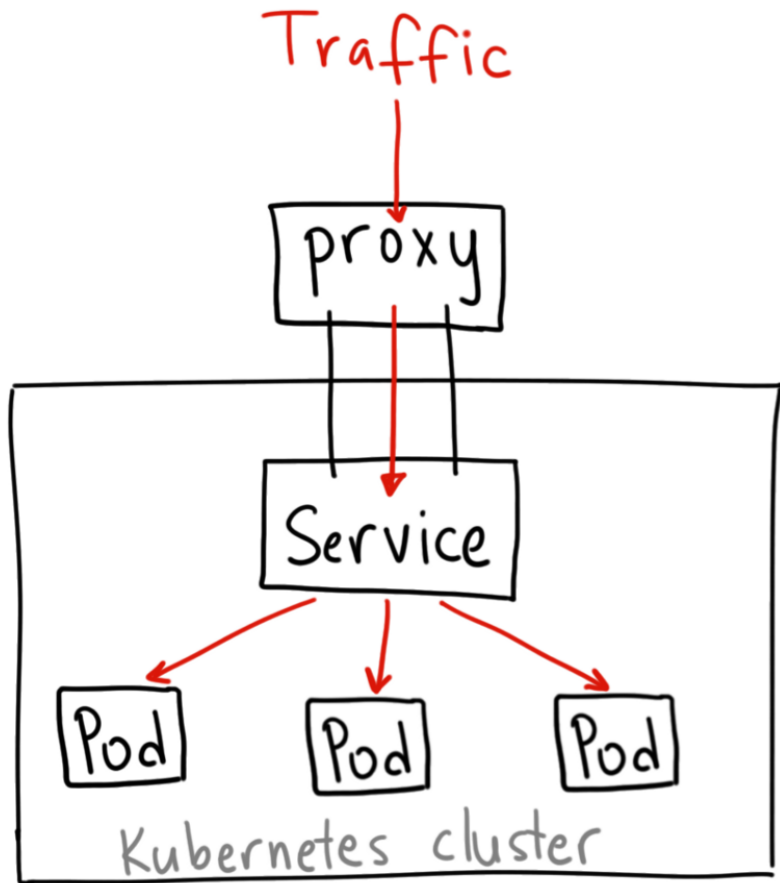
Service实现-Iptables



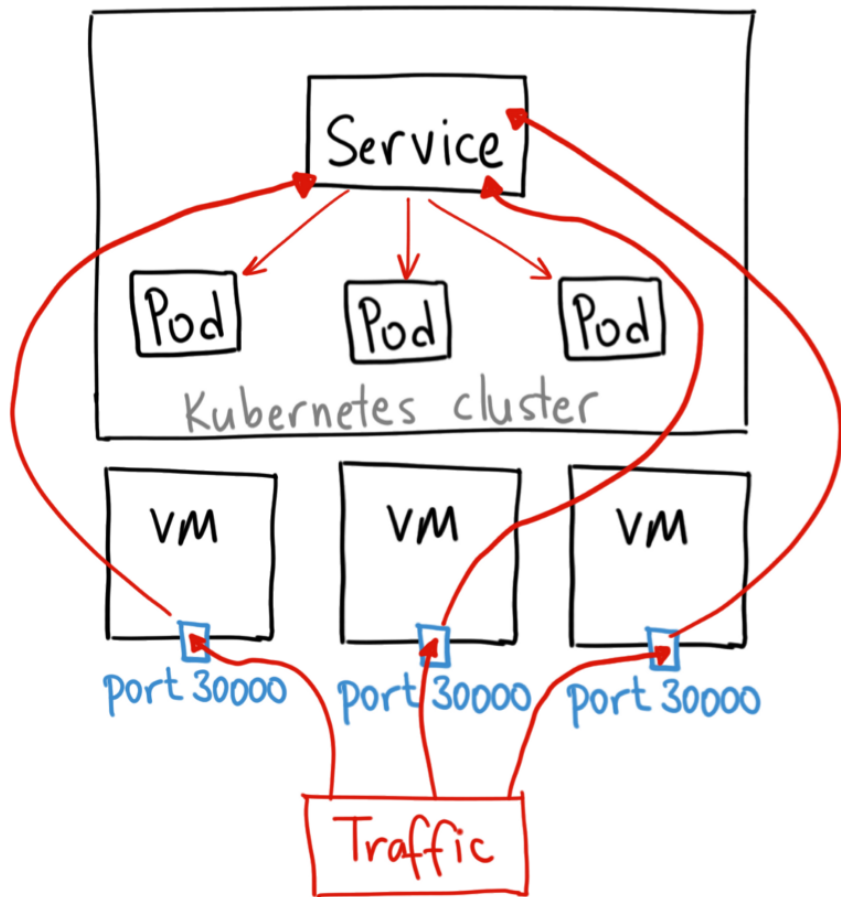
Service实现-LVS



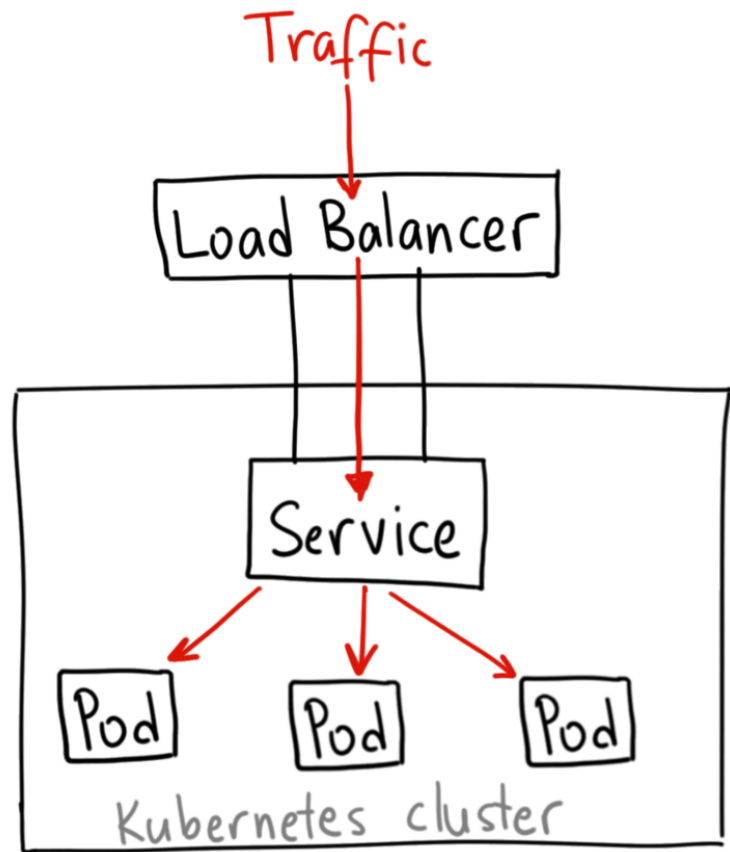
Service入口ClusterIP



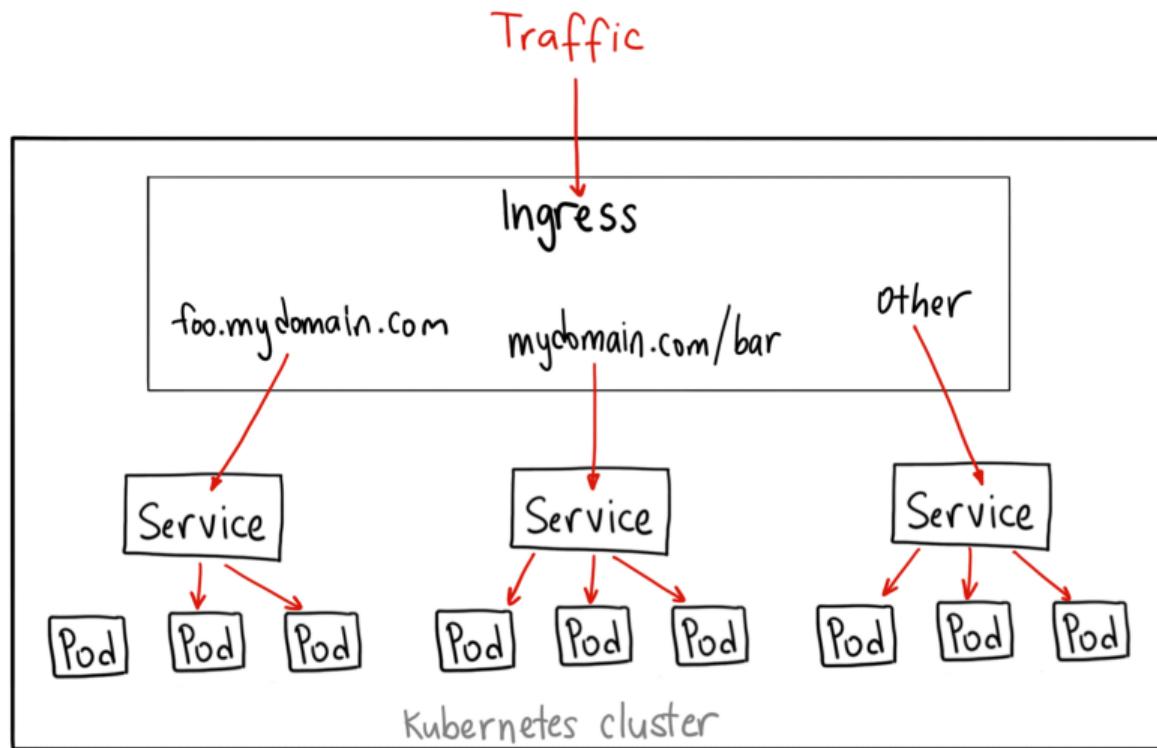
Service入口NodePort



Service入口LoadBalancer



Service入口Ingress



K8S存储

- Volume是Pod中能够被多个容器访问的共享目录
- Kubernetes中的Volume与Pod生命周期相同，但与容器的生命周期不相关
- Kubernetes支持多种类型的Volume，并且一个Pod可以同时使用任意多个Volume
- Volume类型包括:

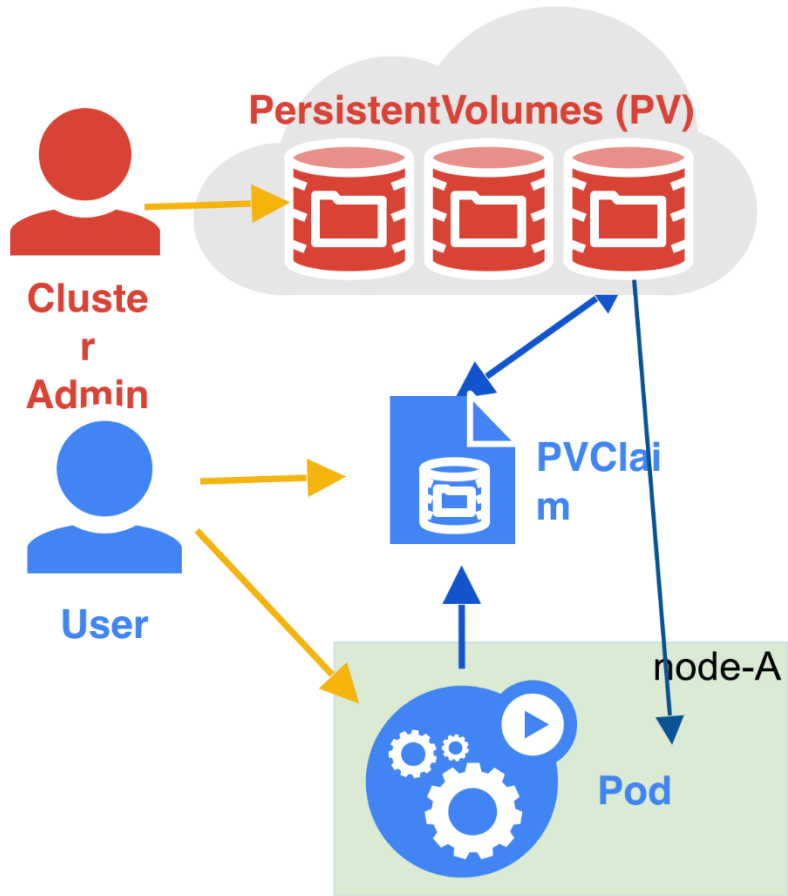
EmptyDir：Pod分配时创建，K8S自动分配，当Pod被移除数据被清空。用于临时空间等。

hostPath：为Pod上挂载宿主机目录。用于持久化数据。

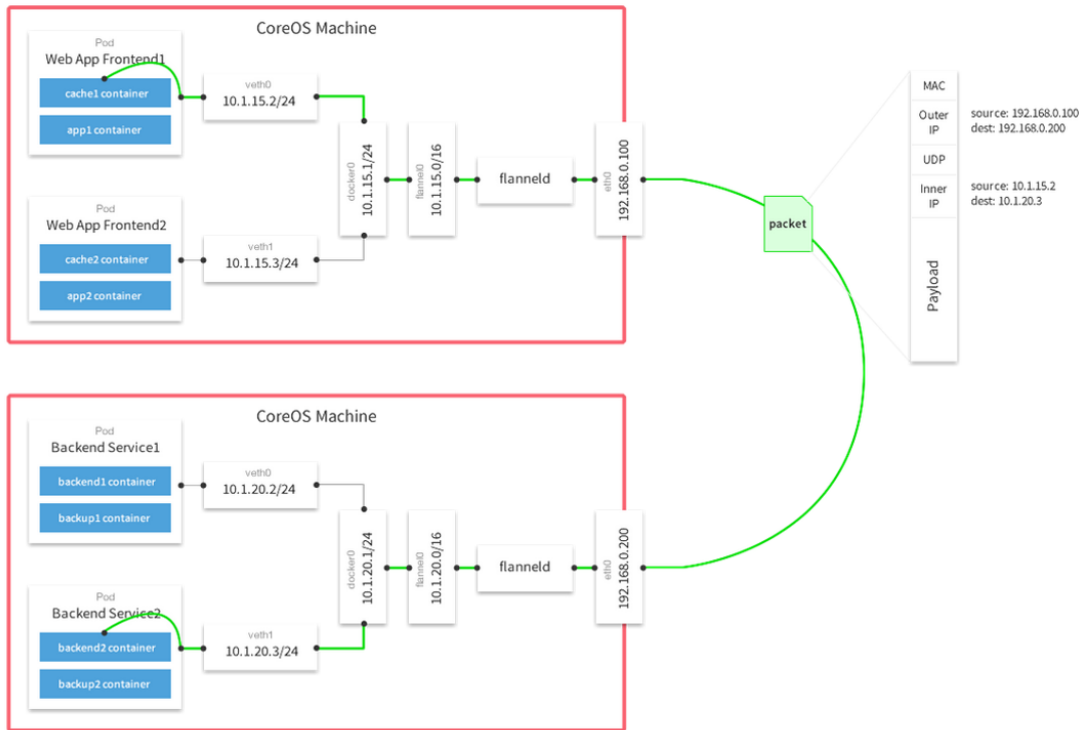
gcePersistentDisk、awsElasticBlockStore：挂载公有云盘。

nfs、iscsi、glusterfs、rbd、gitRepo：挂载相应磁盘资源。

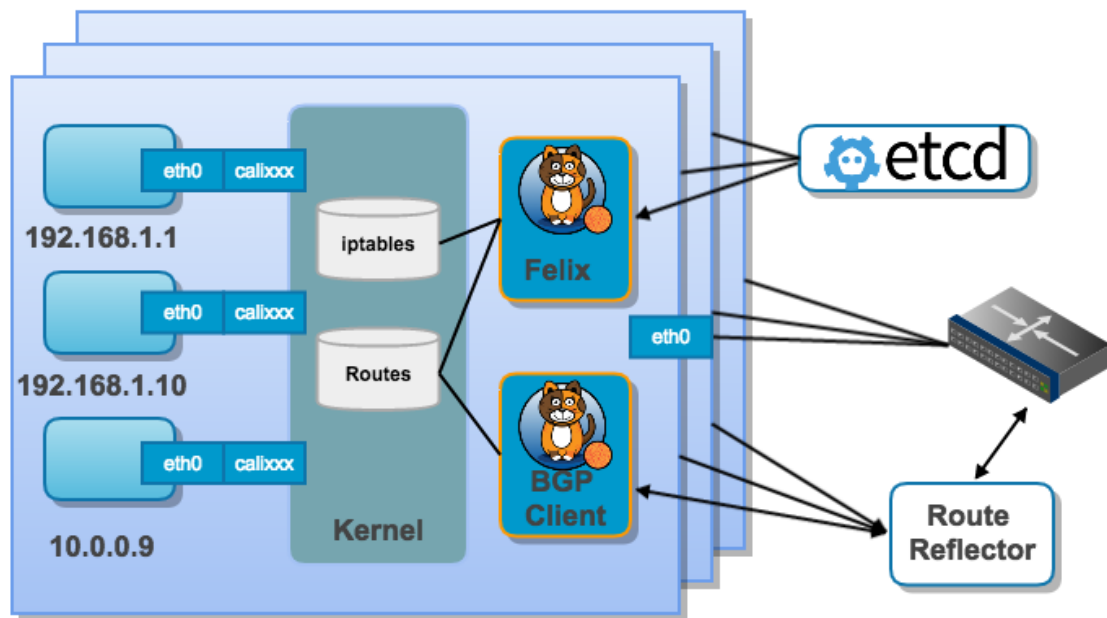
静态绑定模式 PVC



K8S网络flannel



K8S网络 Calico



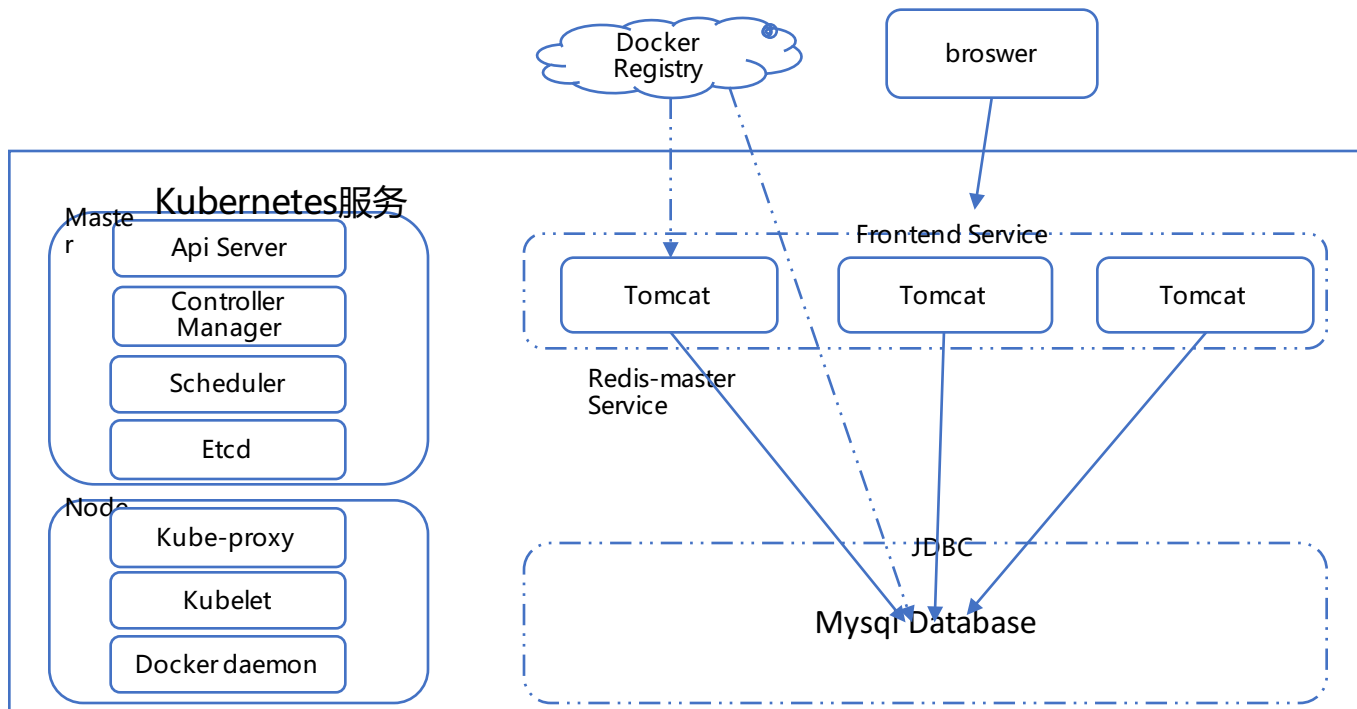
目录

1 K8S自动化部署

2 K8S核心概念

➔ 3 K8S功能实践

JAVA-TOMCAT应用





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？

