Geekbang 极客邦科技 InfoQ

QCon 全球软件开发大会【北京站】2016

OpenStack + Kubernetes: 搭建容器虚拟机组合云平台

qingyuanos 王昕

Intemational Software Development Conference


2402
2302
1201
3301
3302


VM
容器

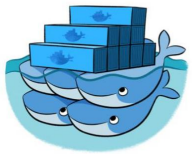
提供虚拟机服务的意义

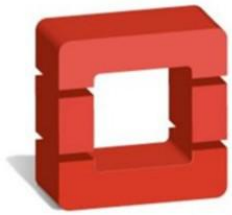
- 客户的需求不仅仅是更多的计算能力
- 安全性：更小的Attack Surface
- 易于提供有状态服务
- 传统应用容易迁移
- Windows应用容易迁移
- 易于部署单体应用
- 用于桌面云
- 多服务单服务器部署

云平台技术的选择


Apache Mesos


Kubernetes


Docker Orchestration



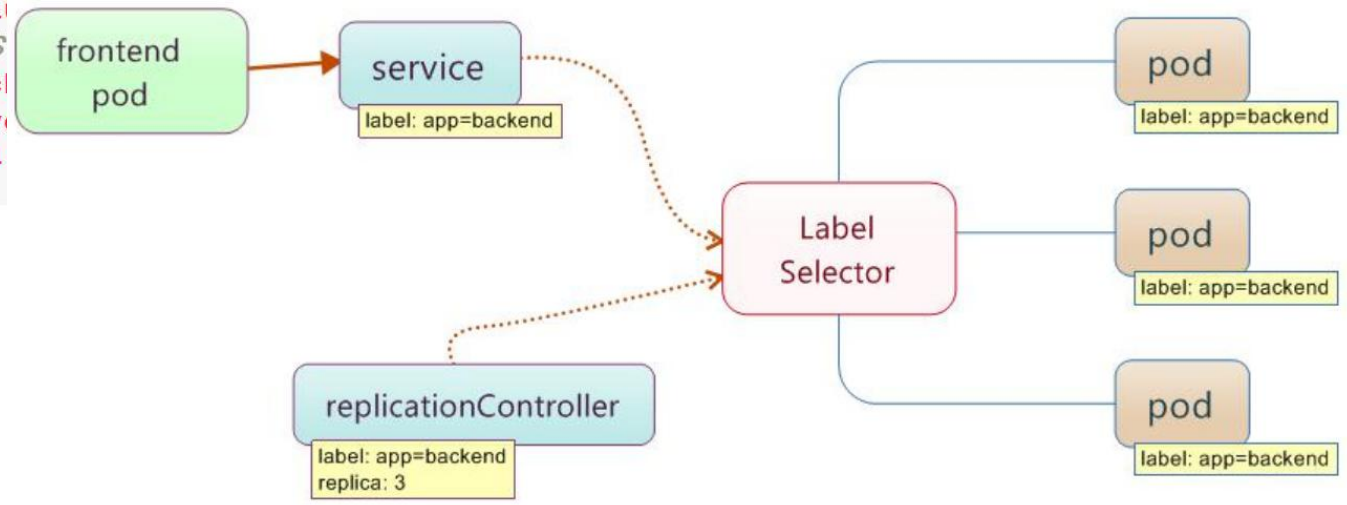
云平台技术选型

容器编排系统的选择
——Kubernetes的优势
vs. Mesos and Swarm

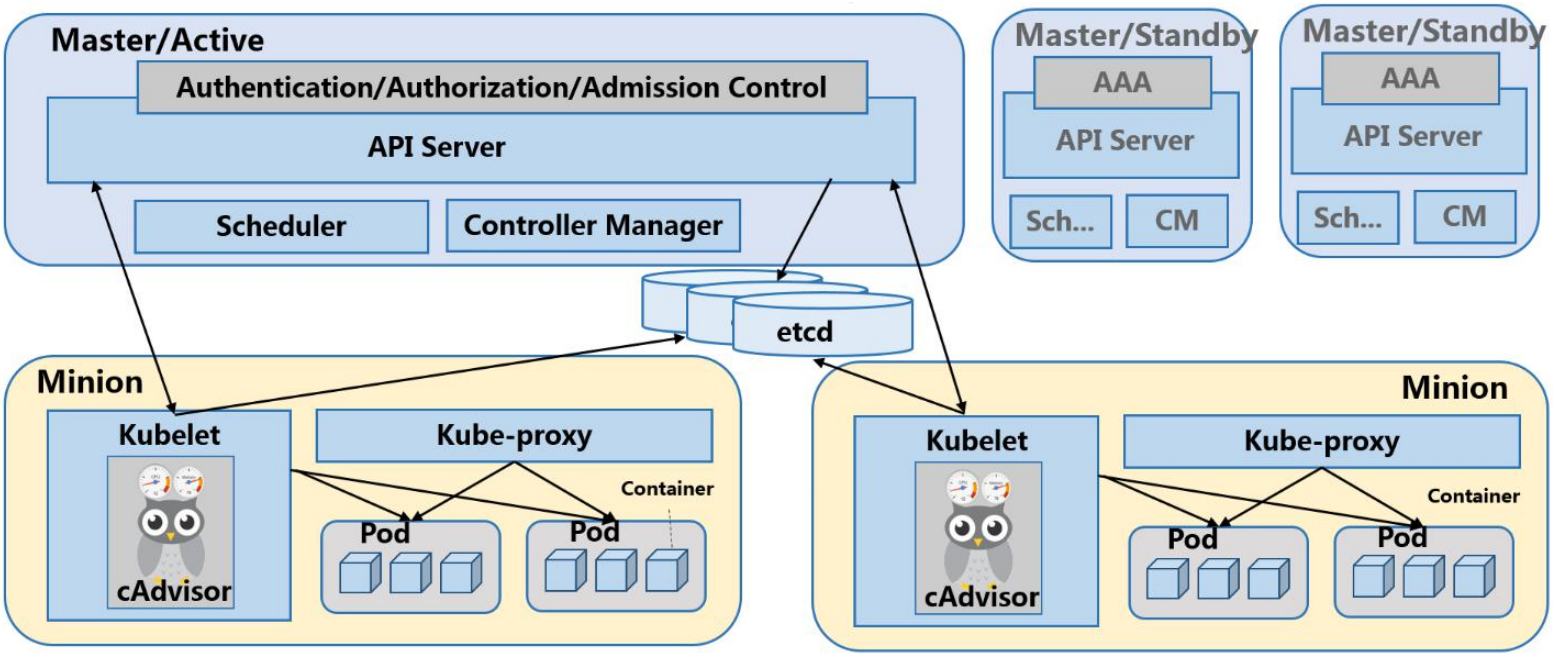
- 来自Google的简单一致的设计理念
- 原生为容器集群打造
- 原生服务发现
  - 统一的资源模型
  - 支持丰富的标签Label发现机制
- 原生负载均衡，高可用方案
- 原生的Rolling Update设计
- 为生产环境专门打造的容器集群
  - 多镜像Pod
  - 多种业务类型：Service+RC/Job/DaemonSet
  - 自动伸缩：AutoScaler
  - 多种Volumn驱动
    - emptyDir, hostPath, gcePersistentDisk, …
    - persistentVolumn, persistentVolumnClaim

Kubernetes的统一资源模型和丰富的标签选择器

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-ebs
spec:
  containers:
  - image: gcr.io/google_containers/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /test-ebs
      name: test-volume
  volumes:
  - name: test-vol
    # This AWS EBS
    awsElasticBloc
      volumeID: <v
      fsType: ext4
```



Kubernetes的架构



- Pod： 最小部署单元，可支持多容器镜像
- RC：控制Pod的个数和Pod的生命周期
- Service：服务入口，由Kube-proxy支持负载均衡
- 原生负载均衡，高可用方案

Kubernetes所支持的存储卷类型

Kubernetes Volumes Support

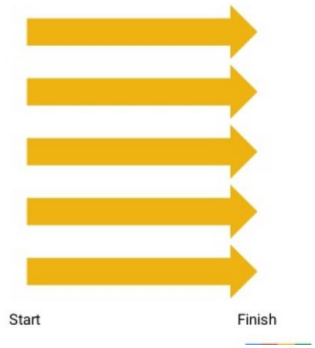
- **emptyDir**
- hostPath
- gcePersistentDisk
- **awsElasticBlockStore**
- **nfs**
- iscsi
- **flocker**
- **glusterfs**
- rbd
- **gitRepo**
- secret
- persistentVolumeClaim

Kubernetes最新支持的资源类型

Jobs

```
apiVersion: extensions/v1beta1
kind: Job
metadata:
  name: ffmpeg
spec:
  selector:
    matchLabels:
      app: ffmpeg
  # run 5 times before done
  completions: 5

...
```

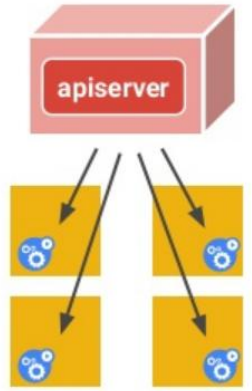


DaemonSets

Solution: Daemon Sets!

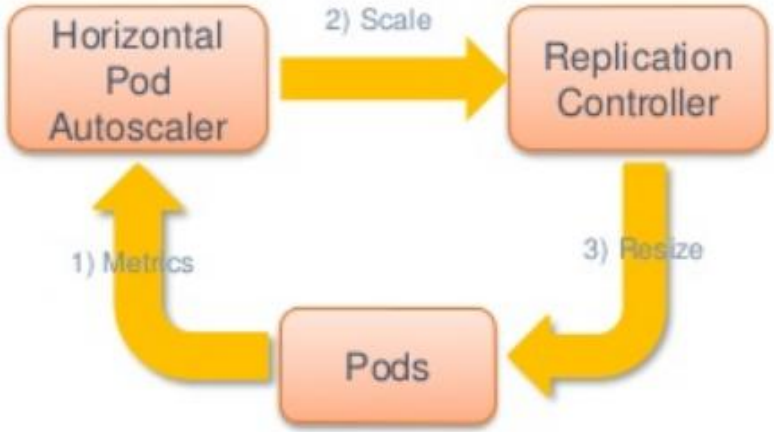
$ kubectl create -f daemon-sharder.yml

```
apiVersion: extensions/v1
kind: DaemonSet
spec:
  template:
    spec:
      nodeSelector:
        app: datastore-node
      containers:
        image: kubernetes/sharded
        ports:
          - containerPort: 9042
```

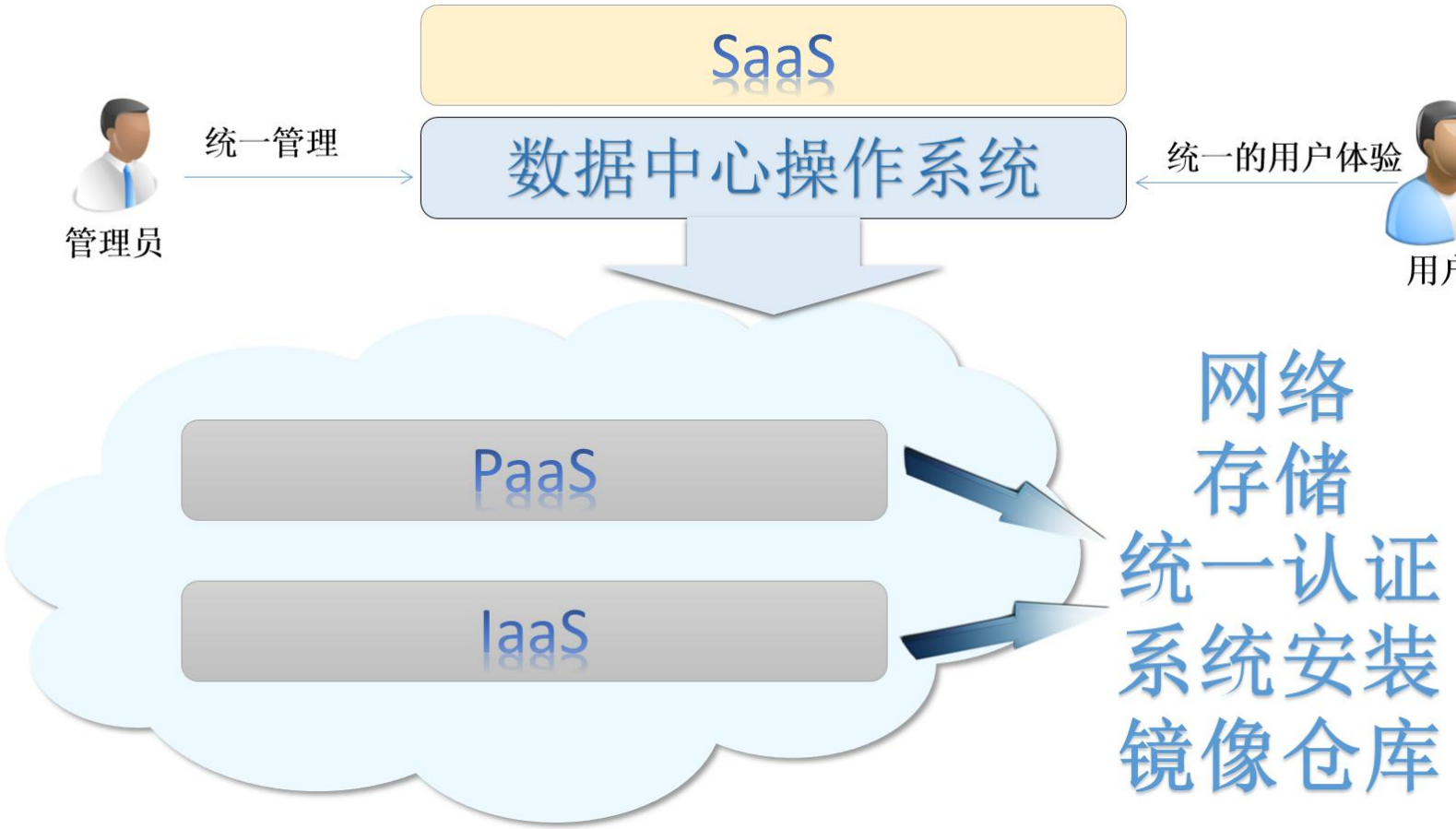


Kubernetes: Autoscale Pod

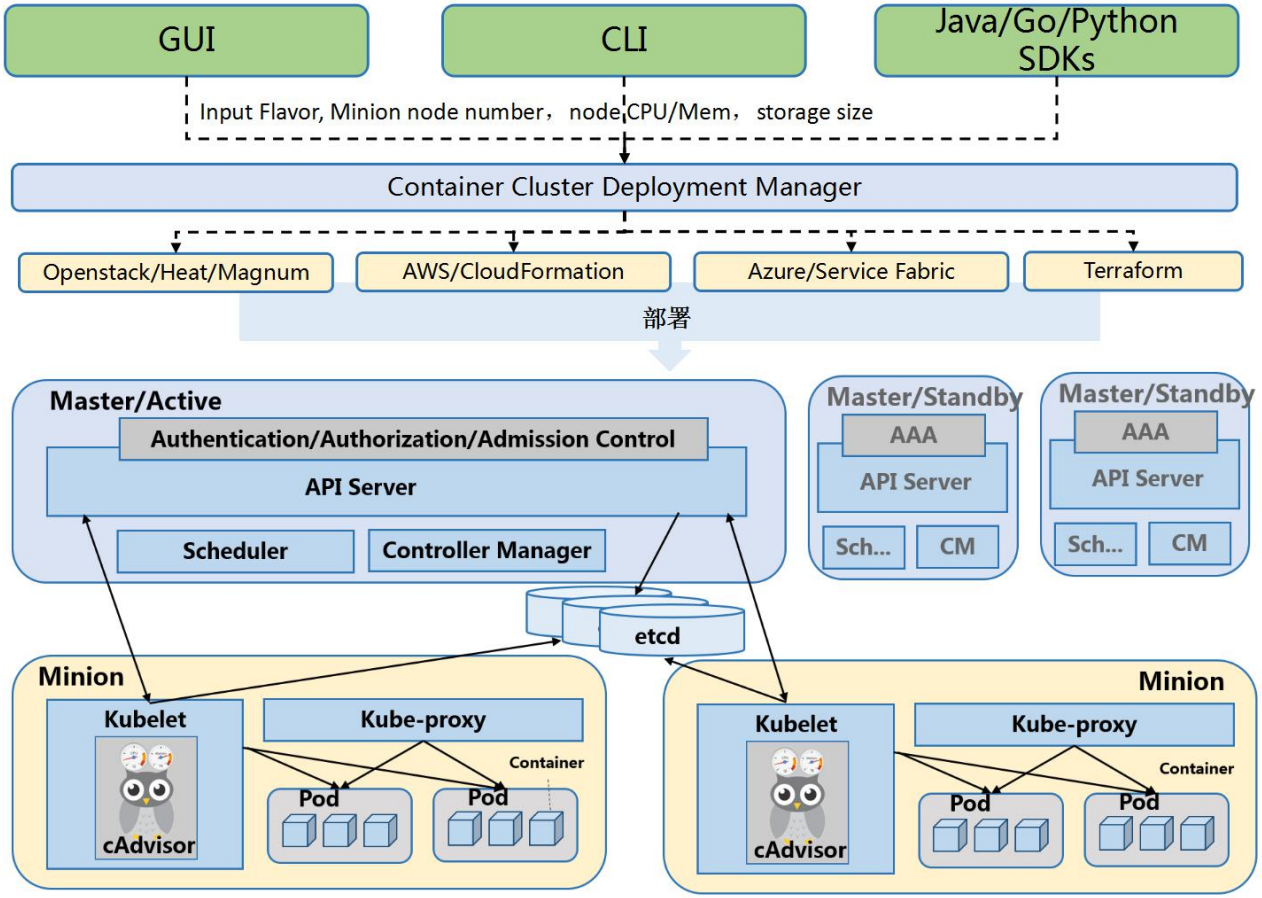
1. **Metrics:** The autoscaler periodically queries pods' metrics.
2. **Scale:** Based on collected metrics, the autoscaler uses a built-in algorithm to trigger the scaling.
3. **Resize:** Replication Controller resize the pods.



OpenStack和K8s的集成系统

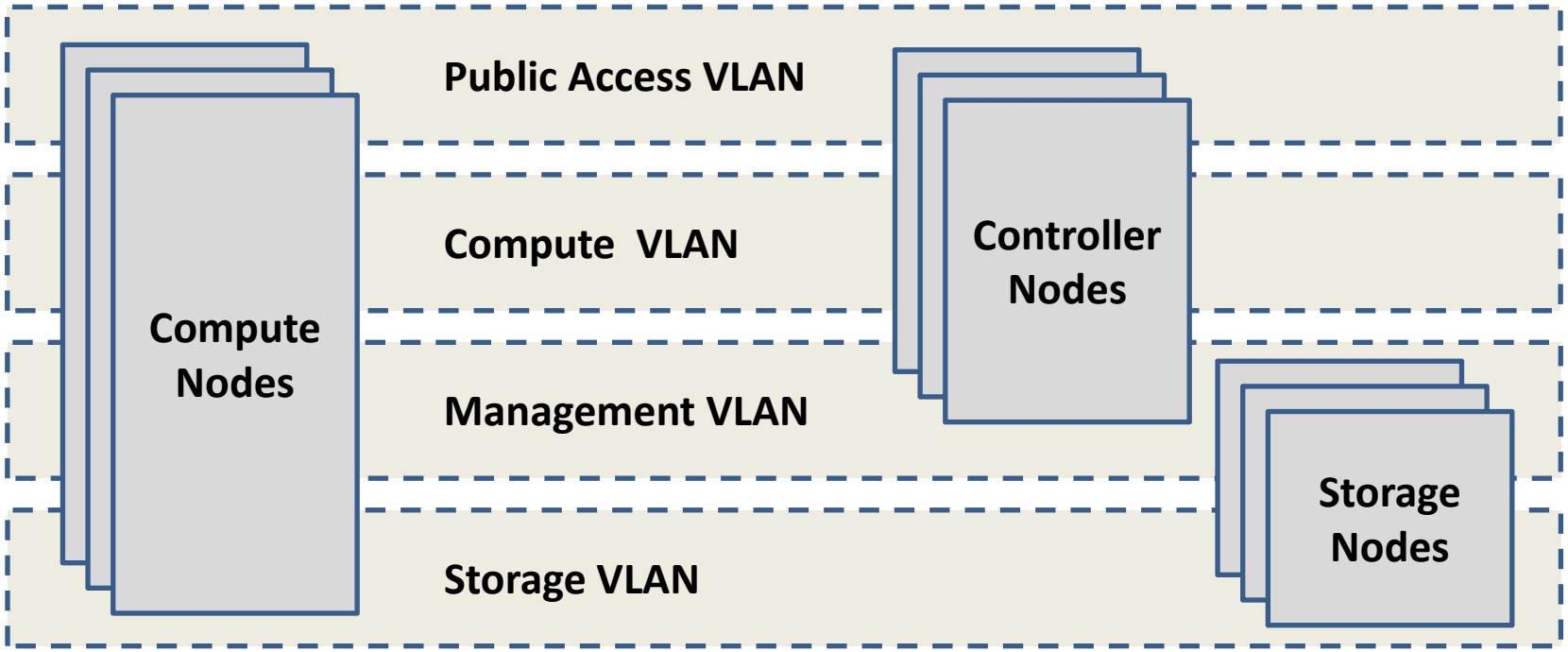


支持跨IaaS部署K8s集群

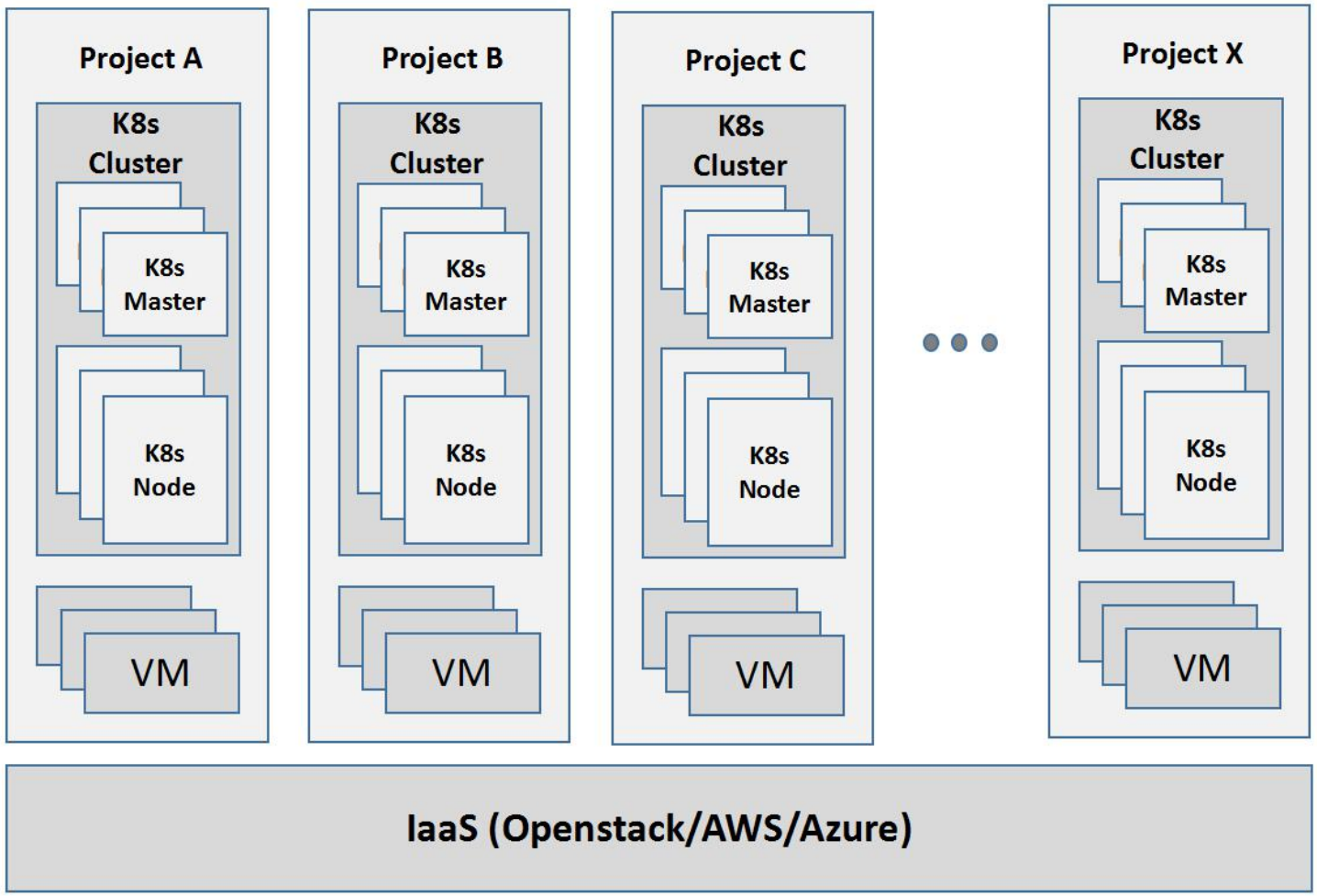


网络集成方案的演变

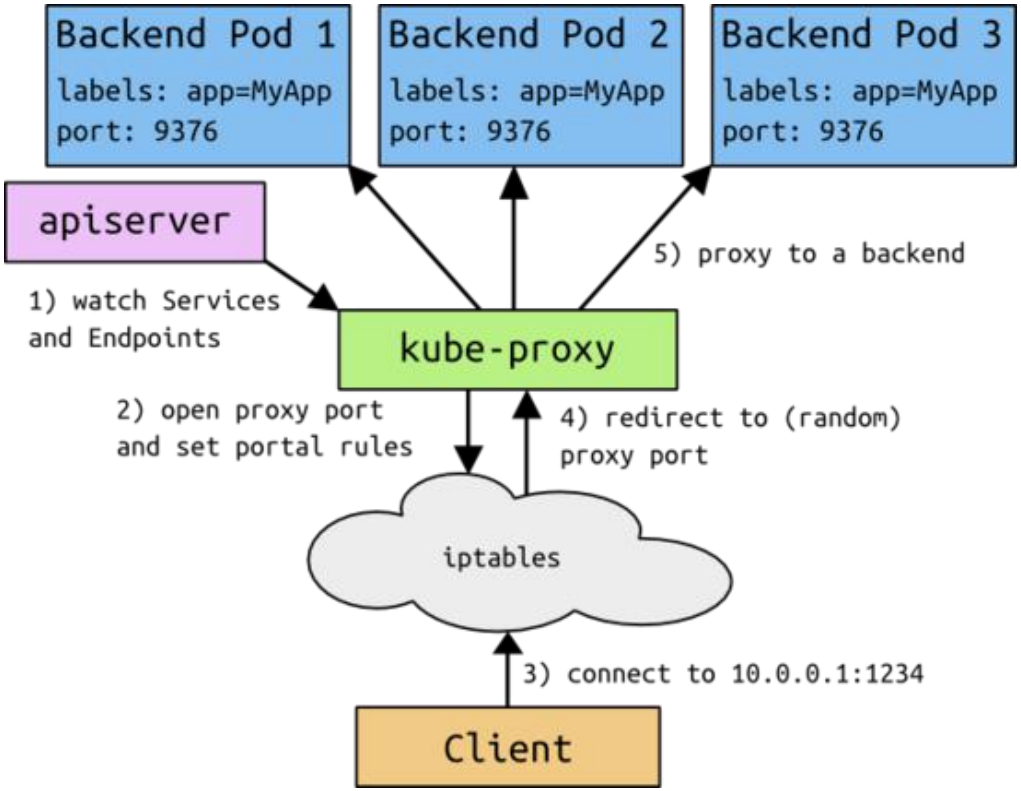
IaaS层的物理网络架构



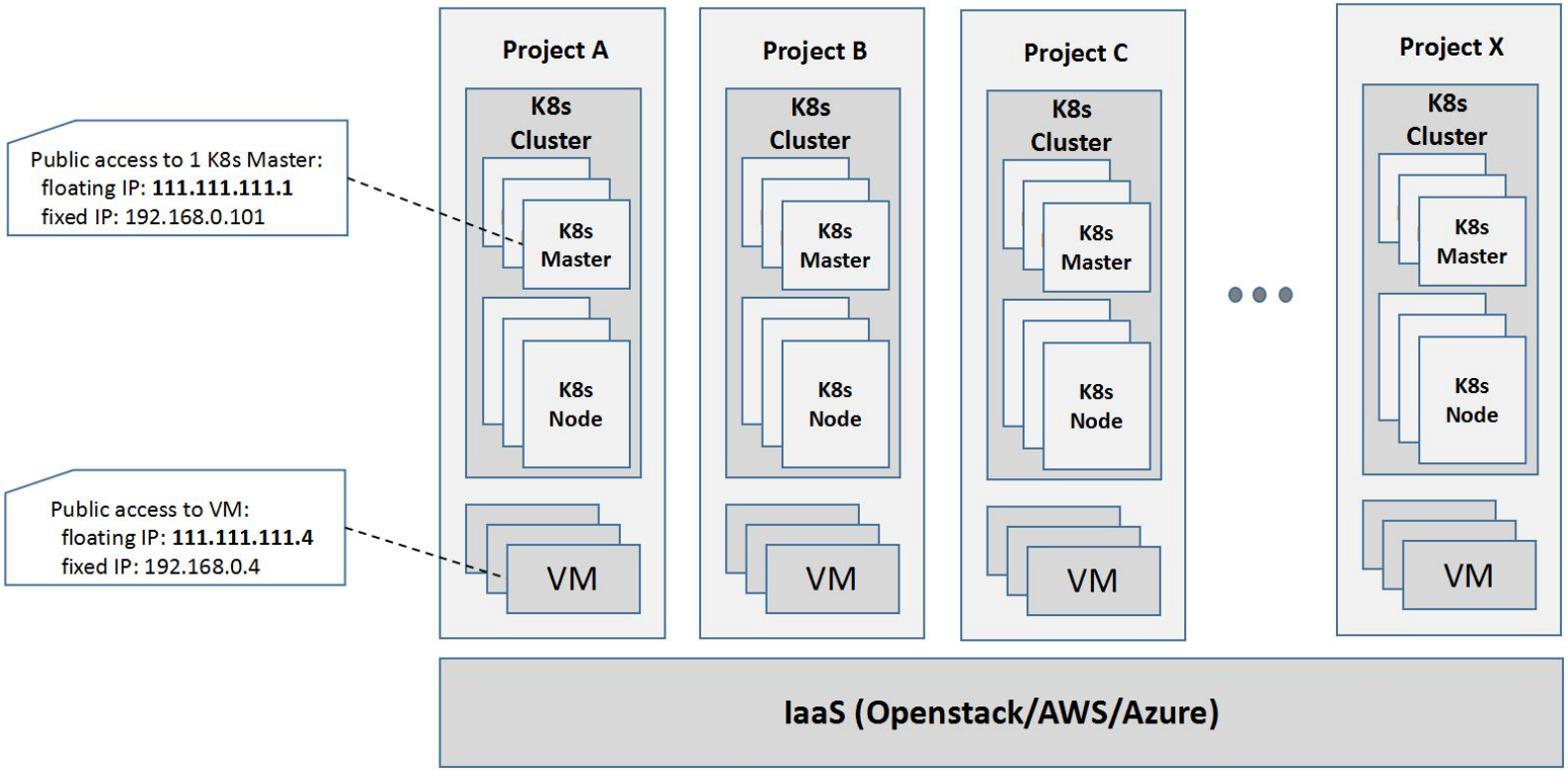
多租户隔离的容器和虚拟机组合网络



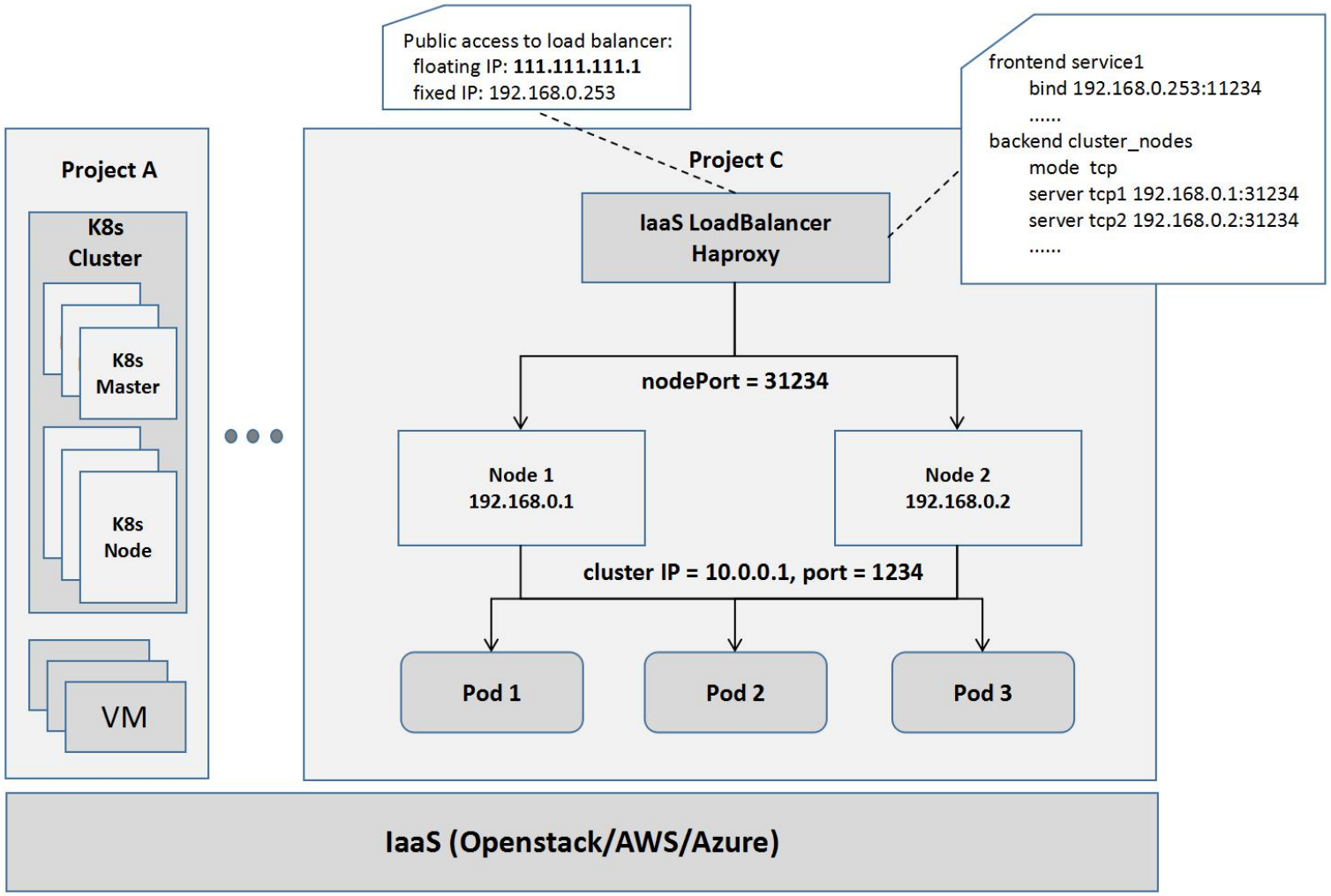
kube-proxy的负载均衡原理



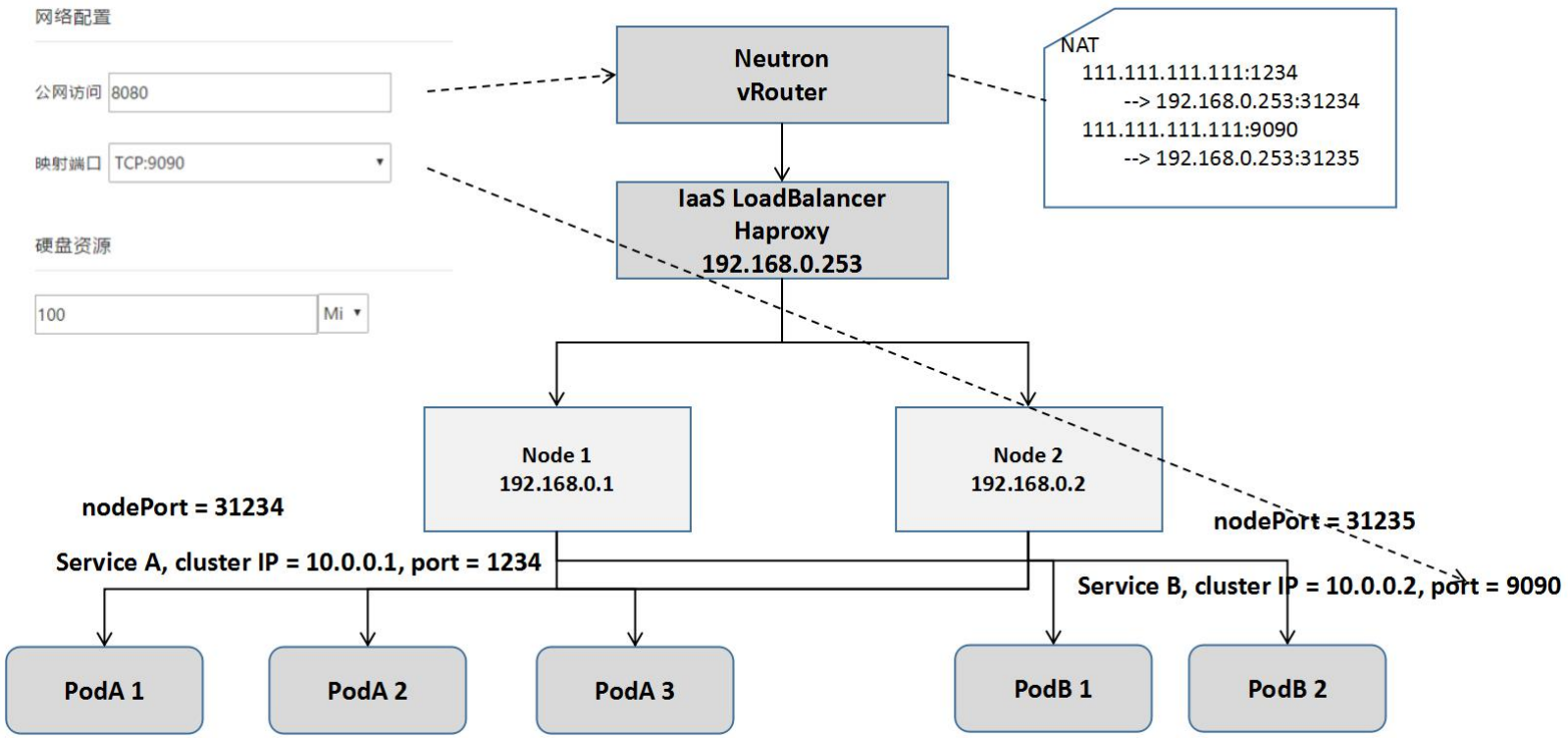
对外发布服务——浮动IP模式



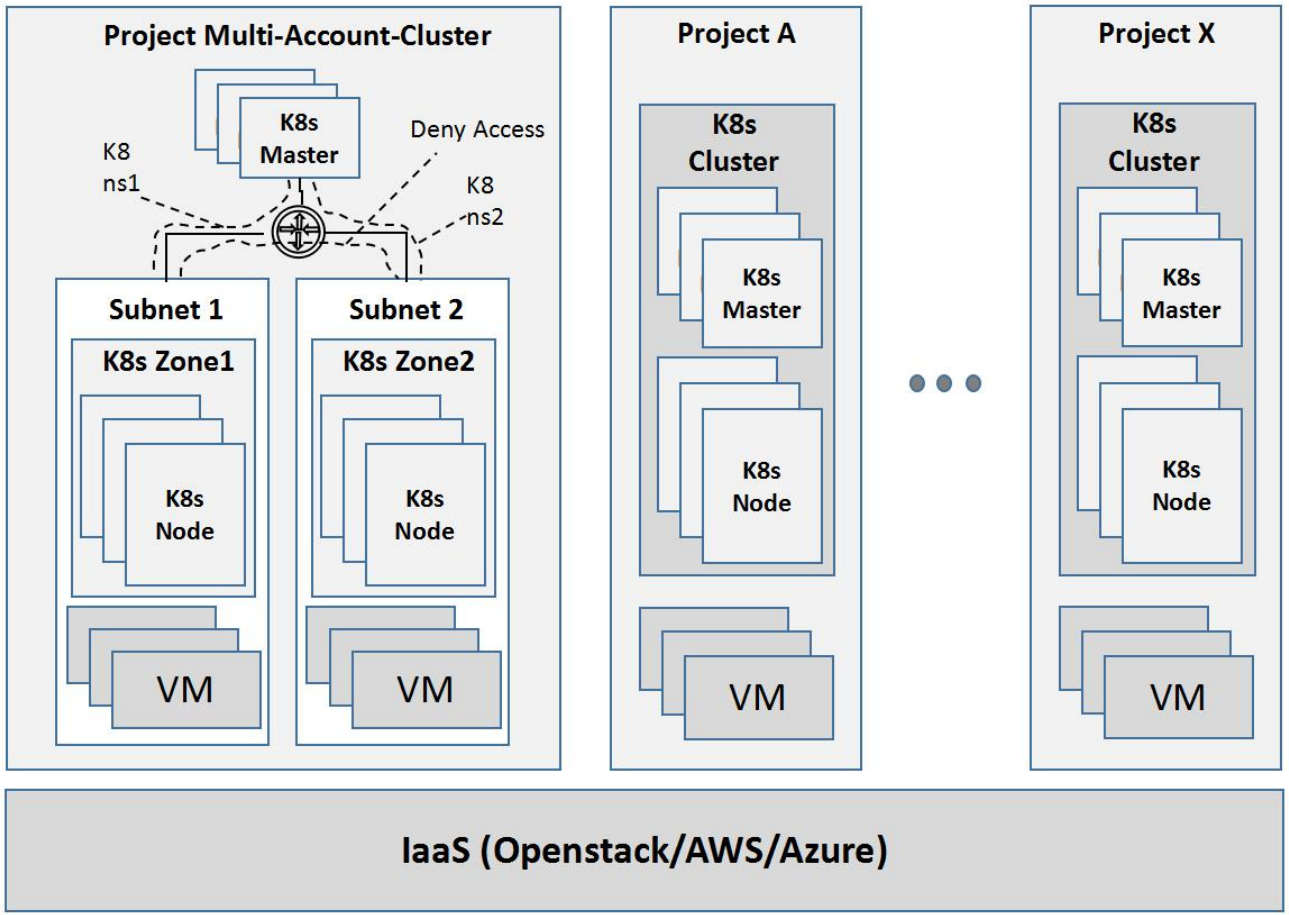
对外发布应用服务


Public access to load balancer:
floating IP: 111.111.111.1
fixed IP: 192.168.0.253
frontend service1
bind 192.168.0.253:11234
.....
backend cluster_nodes
mode tcp
server tcp1 192.168.0.1:31234
server tcp2 192.168.0.2:31234
.....
Project A
K8s Cluster
K8s Master
K8s Node
VM
Project C
IaaS LoadBalancer Haproxy
nodePort = 31234
Node 1 192.168.0.1
Node 2 192.168.0.2
cluster IP = 10.0.0.1, port = 1234
Pod 1
Pod 2
Pod 3
IaaS (Openstack/AWS/Azure)

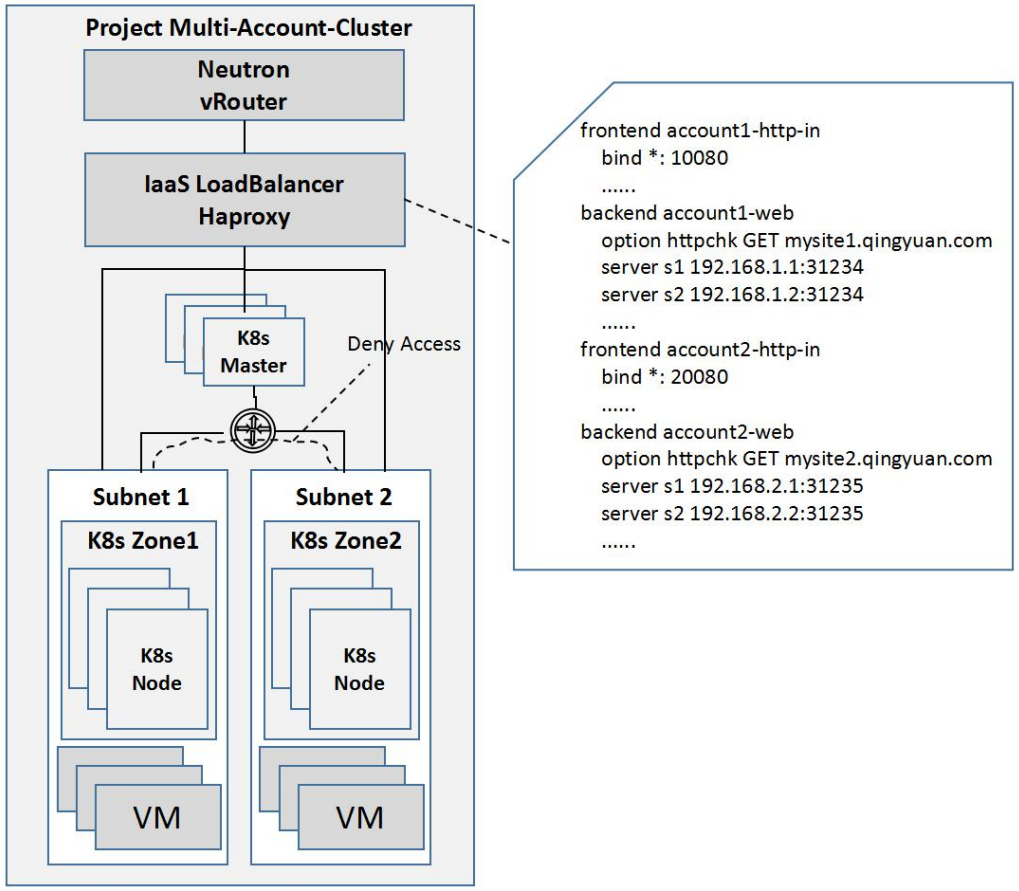
利用端口映射节省IP


网络配置
公网访问 8080
映射端口 TCP:9090
硬盘资源
100 Mi
Neutron
vRouter
NAT
111.111.111.111:1234
--> 192.168.0.253:31234
111.111.111.111:9090
--> 192.168.0.253:31235
IaaS LoadBalancer
Haproxy
192.168.0.253
Node 1
192.168.0.1
Node 2
192.168.0.2
nodePort = 31234
nodePort = 31235
Service A, cluster IP = 10.0.0.1, port = 1234
Service B, cluster IP = 10.0.0.2, port = 9090
PodA 1
PodA 2
PodA 3
PodB 1
PodB 2

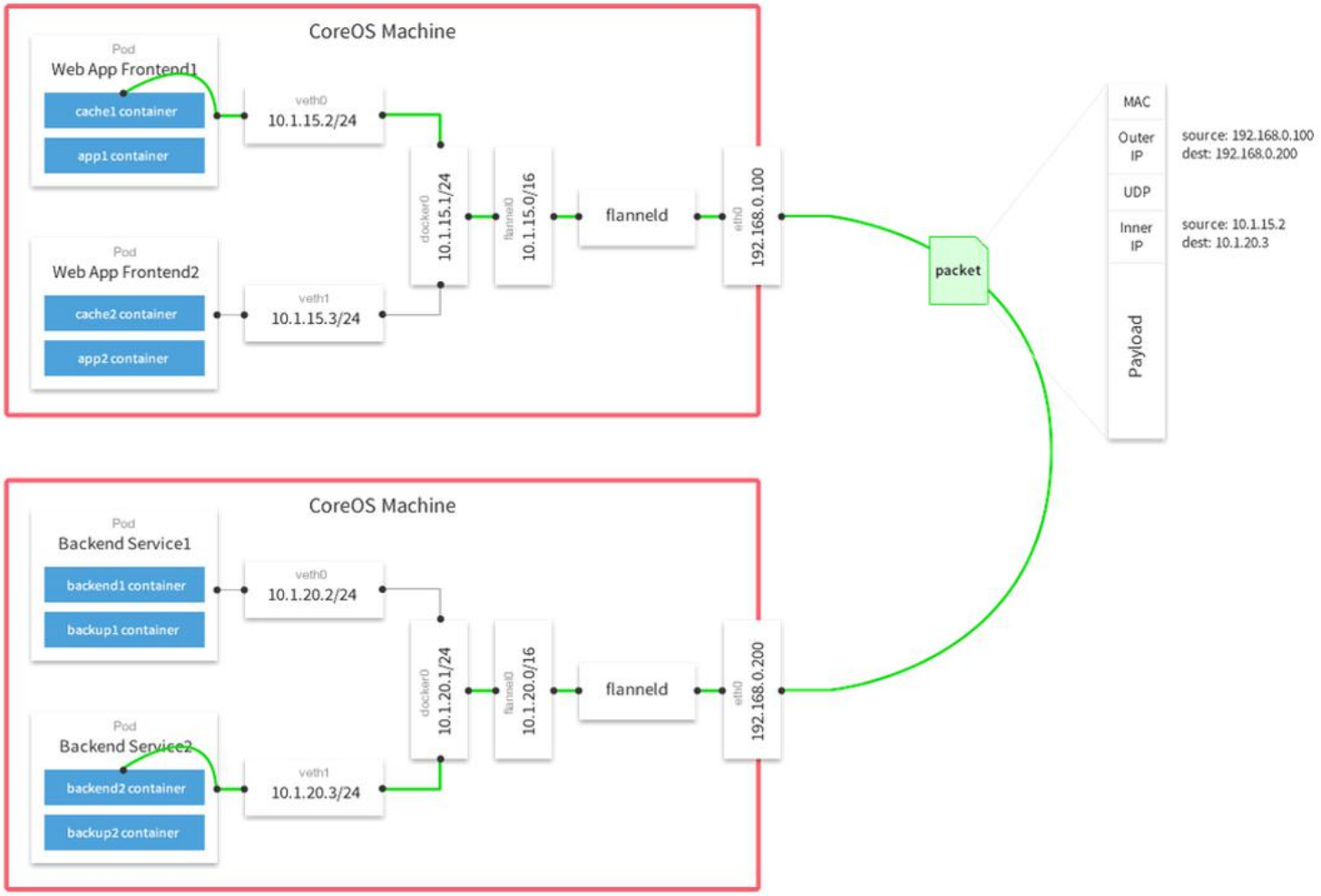
多用户共享Kubernetes集群



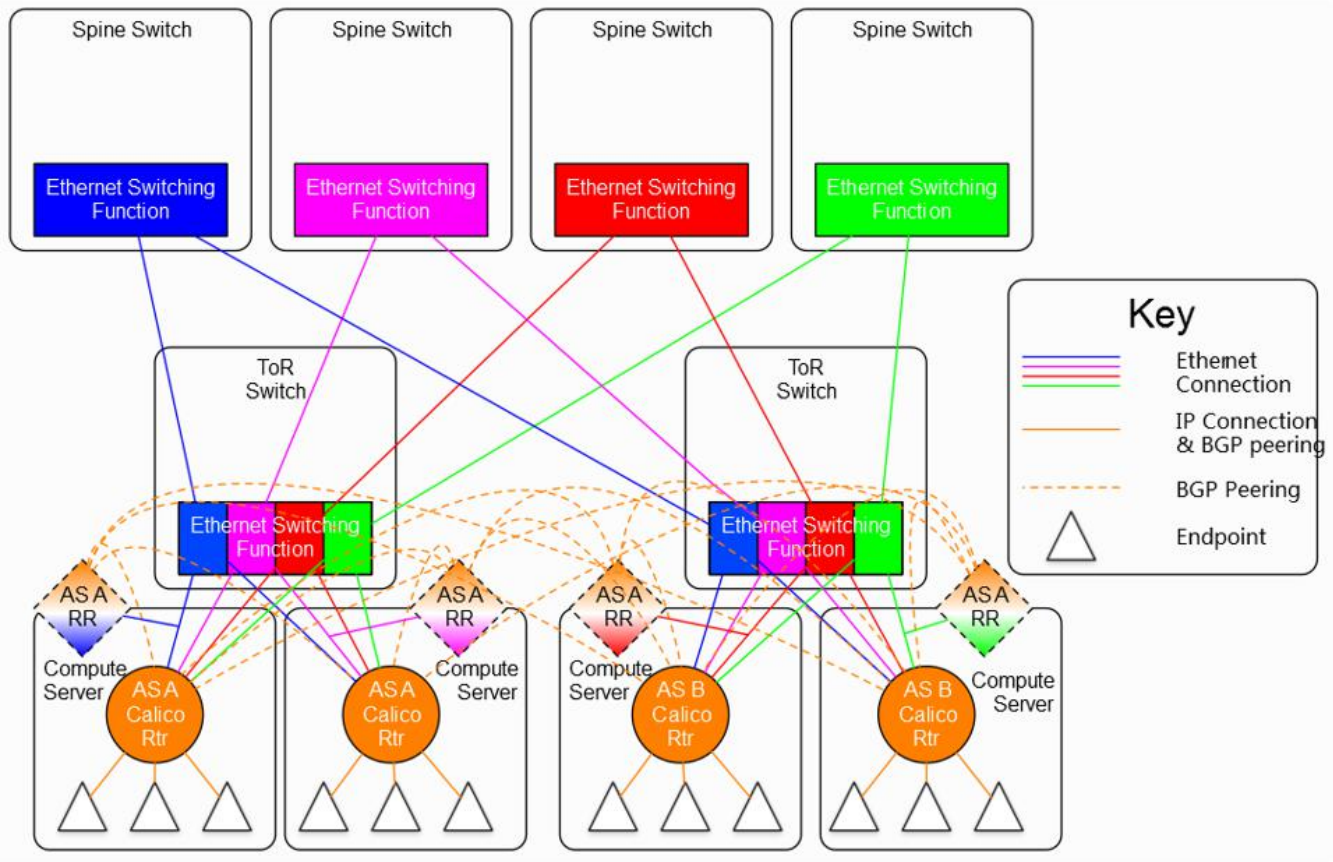
通过二级域名发布服务


Project Multi-Account-Cluster
Neutron
vRouter
IaaS LoadBalancer
Haproxy
K8s
Master
Deny Access
Subnet 1
K8s Zone1
K8s
Node
VM
Subnet 2
K8s Zone2
K8s
Node
VM
frontend account1-http-in
bind *: 10080
.....
backend account1-web
option httpchk GET mysite1.qingyuan.com
server s1 192.168.1.1:31234
server s2 192.168.1.2:31234
.....
frontend account2-http-in
bind *: 20080
.....
backend account2-web
option httpchk GET mysite2.qingyuan.com
server s1 192.168.2.1:31235
server s2 192.168.2.2:31235
.....

覆盖网络（Overlay）
Flannel



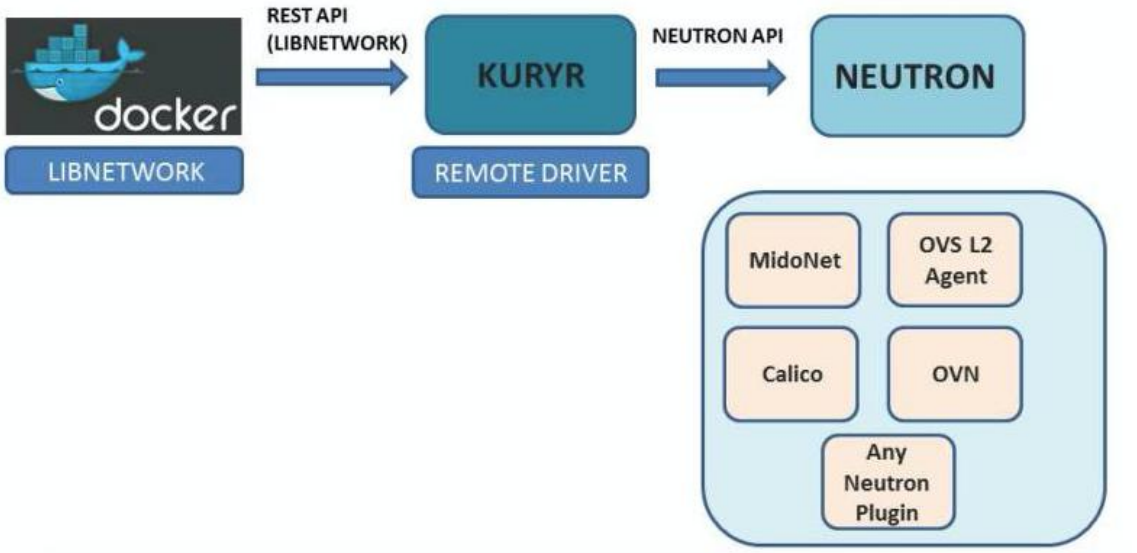
利用路由协议的SDN方案——calico



Kuryr 一个Docker远程驱动

Libnetwork includes the following driver packages:

- null
- bridge
- overlay
- remote



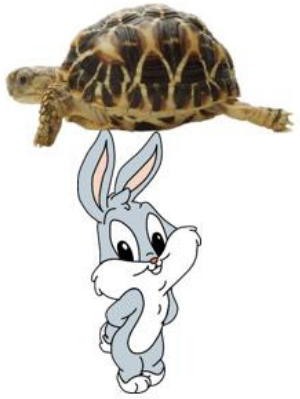
网络集成方案比较

Kuryr

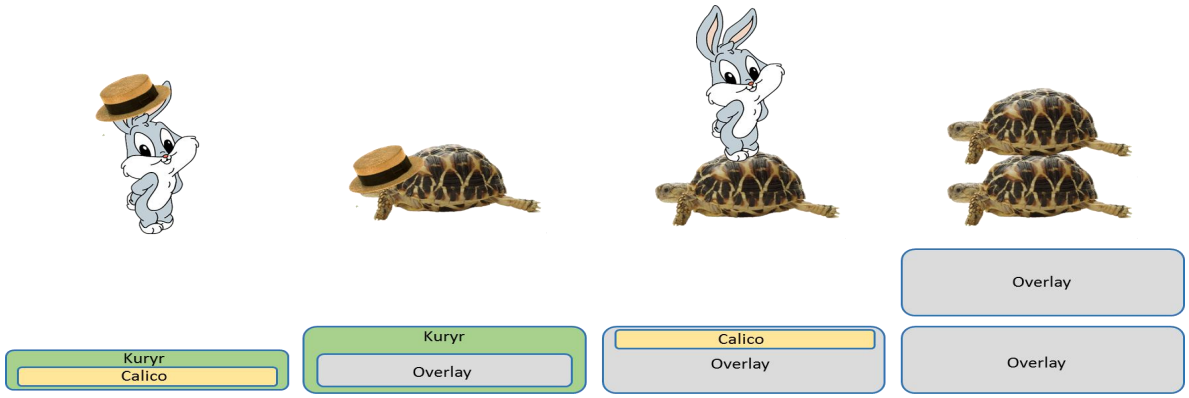
Calico

Overlay

可以排除的组合



网络集成方案比较
可行方案



IaaS层网络	PaaS层网络	适用场景
Calico	Kuryr	不需要多租户隔离，大量使用容器技术，对性能要求很高
Overlay	Kuryr	需要多租户隔离，需要统一管理容器网络和虚拟机网络，将容器用作轻量级虚拟机，对性能要求较高
Overlay	Calico	需要多租户隔离，对容器网络的管理独立于虚拟机网络
Overlay	Overlay	需要多租户隔离，对容器网络的管理独立于虚拟机网络，对性能要求不高；快速集成，用于测试Kubernetes网络方案

曾经遇到的坑和解决方案

覆盖网络造成的MTU Size问题

- 问题
  - Neutron网络做隧道封装时，占用了包头，导致上层网络的最大允许MTU比默认要小，造成虚拟机网络时通时不通
    - 给Linux虚拟机造成问题
    - 给Windows虚拟机造成问题
    - 给虚拟机内的Docker造成问题
- 解决方案
  - 手动改小虚拟机MTU
  - 用CloudInit脚本更改MTU，集成到云平台中

HTTPS的负载均衡

- 问题
  - 如何用HAProxy给HTTPS服务做负载均衡
- 解决方案
  - TCP透传模式
    - 不要用HTTP模式，而要用TCP模式做负载均衡，TCP可以透传TLS连接
    - 后端服务器都要有服务器证书
  - HAProxy认证模式
    - TLS服务器终点为HAProxy，后端连接为明文TCP
    - 要把服务器证书配置到HAProxy上

OpenStack里MySQL Galera集群高可用

- 问题
  - 异步多主多活情况下会出现数据不一致
  - 同步多活情况下容易出现死锁
- 解决方案
  - 改成同步一主两备模式

Kubernetes的PVC绑定问题

- 问题
  - PVC每次申请PV都会占用所有PV容量
- 解决方案
  - 对Kubernetes的PV起初理解偏差，PVC的设计就是占用整个PV
  - 要对每个用户PVC单独开辟PV

Magnum创建baymodel失败

- 问题
  - Baymodel中所使用的镜像没有os-distro属性
- 解决方案
  - 创建虚拟机镜像时一定要指定os-distro属性
    - glance image-create --name fedora-23-atomic --visibility public --disk-format qcow2 --os-distro fedora-atomic --container-format bare --progress --file ./Fedora-Cloud-Atomic-23-20151030.x86_64.qcow2
  - 对于已有的镜像，更新os-distro属性
    - glance image-update --os-distro fedora-atomic fedora-23-atomic

要点总结

- Kubernetes+OpenStack=容器和虚拟机组合服务
- Kubernetes专为生产环境打造的容器集群系统
- 支持多租户的网络解决方案：租户隔离、负载均衡、外网访问、端口映射、二级域名

轻元数据中心操作系统

http://www.qingyuanos.com/opening.html
sales@qingyuanos.com
xwang@qingyuanos.com

Geekbang 极客邦科技 InfoQ

THANKS!