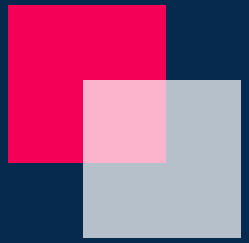


k8s和新版本预研介绍

Wen Zhenglin
2018-6-14

概览

1. K8s介绍
2. K8s新版本介绍

k8s趋势

预测1：Kubernetes项目将在企业中获得成功，不过过程仍然坎坷。

预测2：构建和运行Kubernetes应用的复杂性将由日益崛起的Kubernetes平台解决。

预测3：到2018年底，我们将看到近50家Kubernetes认证服务供应商。

预测4：70%的客户将从其云提供商中选择Kubernetes平台。

https://thenewstack.io/predictions-2018-rise-kubernetes-platform/（中文版）

什么是CNCF

CNCF is an open source software foundation dedicated to making cloud native computing universal and sustainable.

CNCF serves as the vendor-neutral home for many of the fastest-growing projects on GitHub, including Kubernetes, Prometheus and Envoy, fostering collaboration between the industry's top developers, end users, and vendors.

https://www.cncf.io/announcement/2018/03/06/cloud-native-computing-foundation-announces-kubernetes-first-graduated-project/

https://www.cncf.io/

CNCF成员


220+ Members and Growing
Platinum Members
Alibaba Cloud
aws
Azure
cisco
DELL Technologies
docker
FUJITSU
Google Cloud
HUAWEI
IBM Cloud
intel
JD.COM
MESOSPHERE
ORACLE
Pivotal
redhat
SAMSUNG SDS
SAP
vmware
Gold Members
AT&T
Baidu 百度
DigitalOcean
JFrog
NEC
NetApp
salesforce
sumologic
SUSE
ZTE
End User Members
Bloomberg
CapitalOne
COMCAST
ebay
DENSO
DiDi
GitHub
Goldman Sachs
indeed
intuit
JD.COM
Morgan Stanley
NAIC
ncsoft
Pinterest
salesforce
Spotify
twilio
twitter
YAHOO! JAPAN
End User Supporters
ADS ON TOP
box
cruise
Layer
Nasdaq
The New York Times
Clark
PUSHER
ricardo.ch
SAP Concur
shopify
showmax
stix
textkernel
THREDUP
ticketmaster
vevo
woorank
Workspot
Academic/Nonprofit
INTERNET.
NAIC
NIPR
SEI
WIKIMEDIA
16
© 2018 Cloud Native Computing Foundation

什么是Docker容器

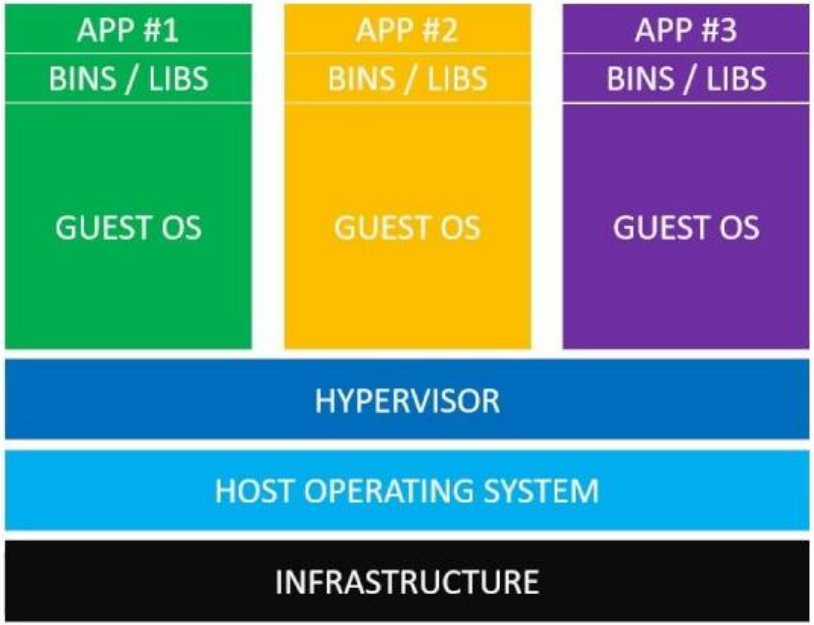
容器的行为就像一台虚拟机，看起来像有他们自己的完整系统。

与虚拟机不同，容器不需要复制整个操作系统，只需要复制它们需要的各个组件即可运行。

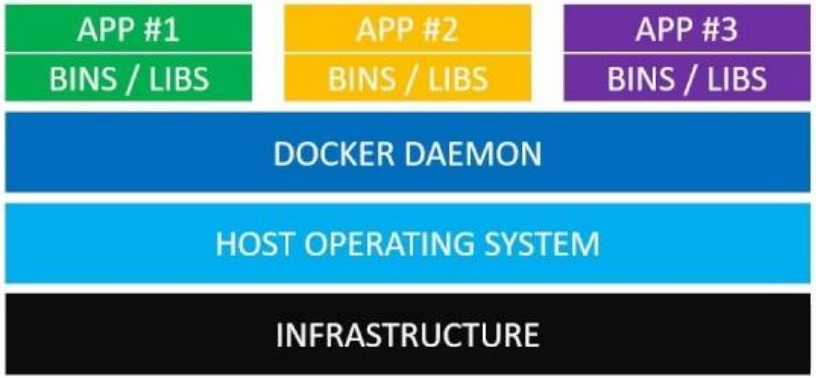
相对于虚拟机，容器可以提高性能并缩小应用程序的大小，同时运行速度更快。

注：docker是容器的一种技术实现，见容器标准 https://www.opencontainers.org/

Docker vs VM



Virtual Machines



Docker Containers

什么是Kubernetes

Kubernetes（简称k8s) 是一个生产级的开源容器编排平台，管理容器的工作负载和服务，支持跨主机群集部署，扩展和管理容器，使用声明式配置描述最终状态，通过自动化来真正实现最终状态。

Kubernetes的创建是基于在谷歌使用Borg系统大规模地运行生产工作负载，长达十五年的经验的基础上，并结合社区的最佳创意和实践来实现的。

Demo of UI

http://172.28.137.7:9090

Kubernetes特性1

自动包装

根据资源需求和其他约束自动放置容器，同时不会牺牲可用性，混合关键和最大努力的工作负载，以提高资源利用率并节省更多资源。

自我修复

重新启动失败的容器，在节点不可用时，替换和重新编排节点上的容器，终止不对用户定义的健康检查做出响应的容器，并且不会在客户端准备投放之前将其通告给客户端。

横向缩放

使用简单的命令或 UI，或者根据 CPU 的使用情况自动调整应用程序副本数。

服务发现和负载均衡

不需要修改您的应用程序来使用不熟悉的服务发现机制，Kubernetes 为容器提供了自己的 IP 地址和一组容器的单个 DNS 名称，并可以在它们之间进行负载均衡。

Kubernetes特性2

自动部署和回滚

Kubernetes 逐渐部署对应用程序或其配置的更改，同时监视应用程序运行状况，以确保它不会同时终止所有实例。 如果出现问题，Kubernetes会为您恢复更改，利用日益增长的部署解决方案的生态系统。

密钥 和 配置 管理

部署和更新密钥和应用程序配置，不会重新编译您的镜像，不会在堆栈配置中暴露密钥(secrets)。

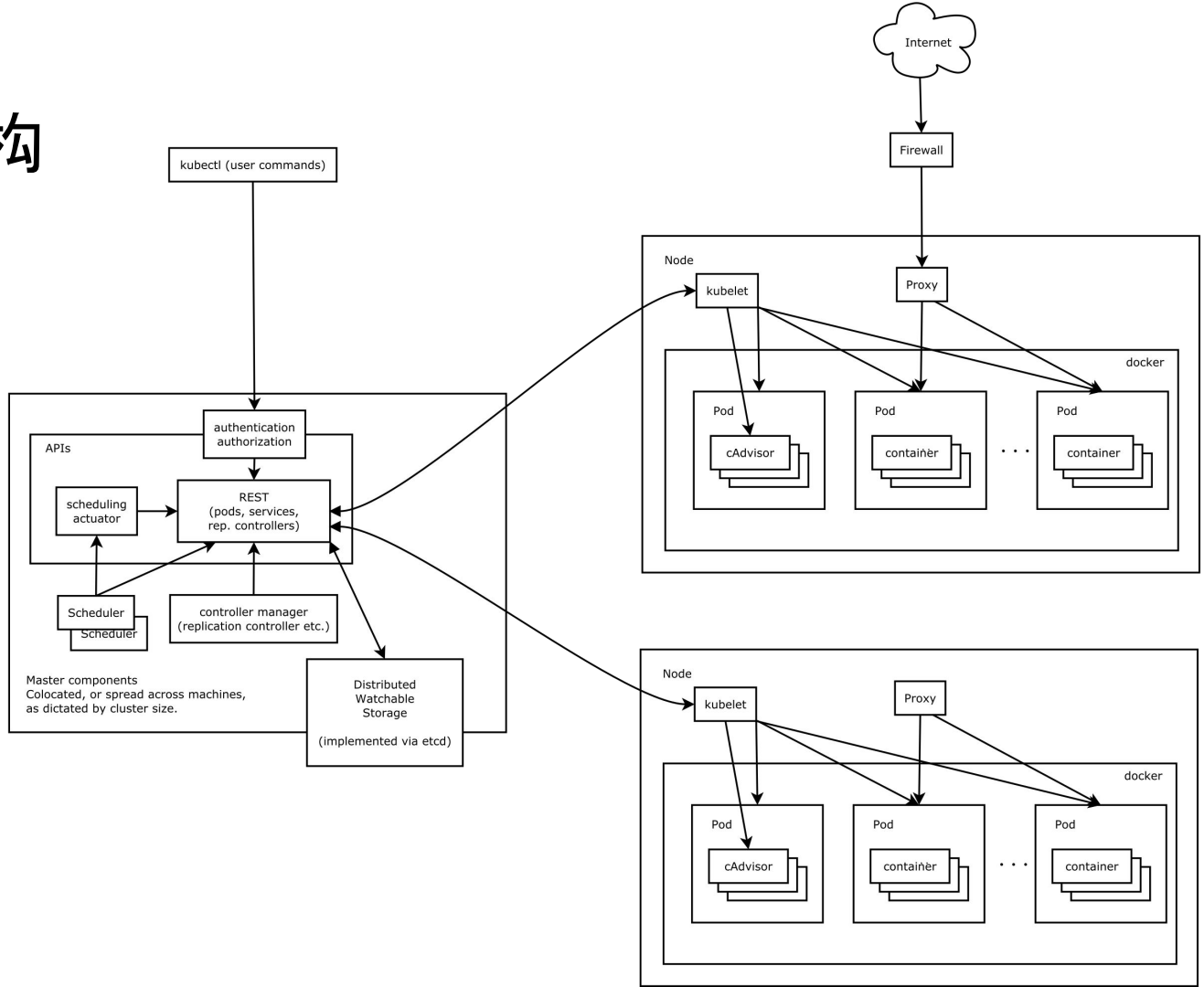
存储编排

自动安装您所选择的存储系统，无论是本地存储，如公有云提供商 GCP 或 AWS, 还是网络存储系统 NFS, iSCSI, Gluster, Ceph, Cinder, 或 Flocker。

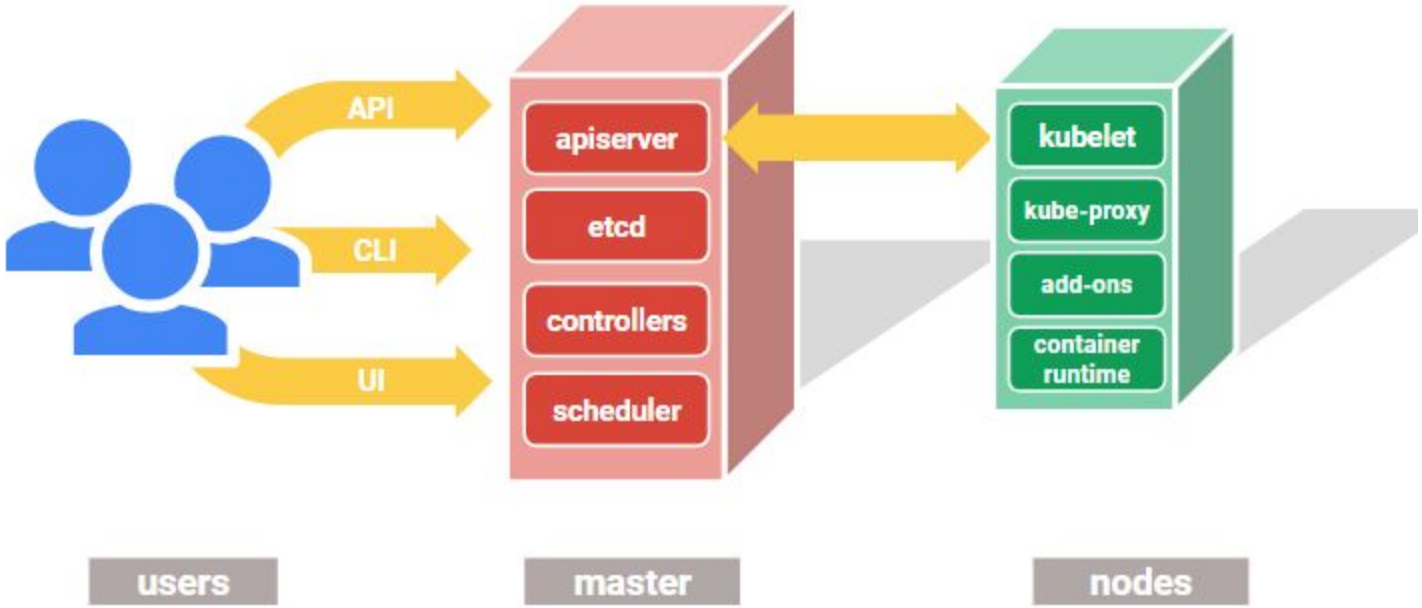
批处理

除了服务之外，Kubernetes还可以管理您的批处理和 CI 工作负载，如果需要，替换出现故障的容器。

k8s架构


Internet
Firewall
kubectl (user commands)
Node
kubelet
Proxy
docker
Pod
cAdvisor
container
Pod
container
Pod
container
APIs
authentication
authorization
scheduling
actuator
REST
(pods, services,
rep. controllers)
Scheduler
Scheduler
controller manager
(replication controller etc.)
Master components
Colocated, or spread across machines,
as dictated by cluster size.
Distributed
Watchable
Storage
(implemented via etcd)
Node
kubelet
Proxy
docker
Pod
cAdvisor
container
Pod
container
Pod
container

k8s系统组件

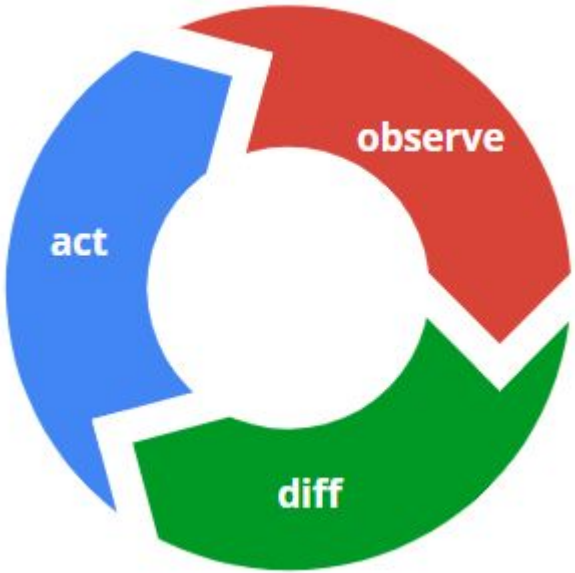

API
CLI
UI
apiserver
etcd
controllers
scheduler
kubelet
kube-proxy
add-ons
container runtime
users
master
nodes

核心概念：控制环

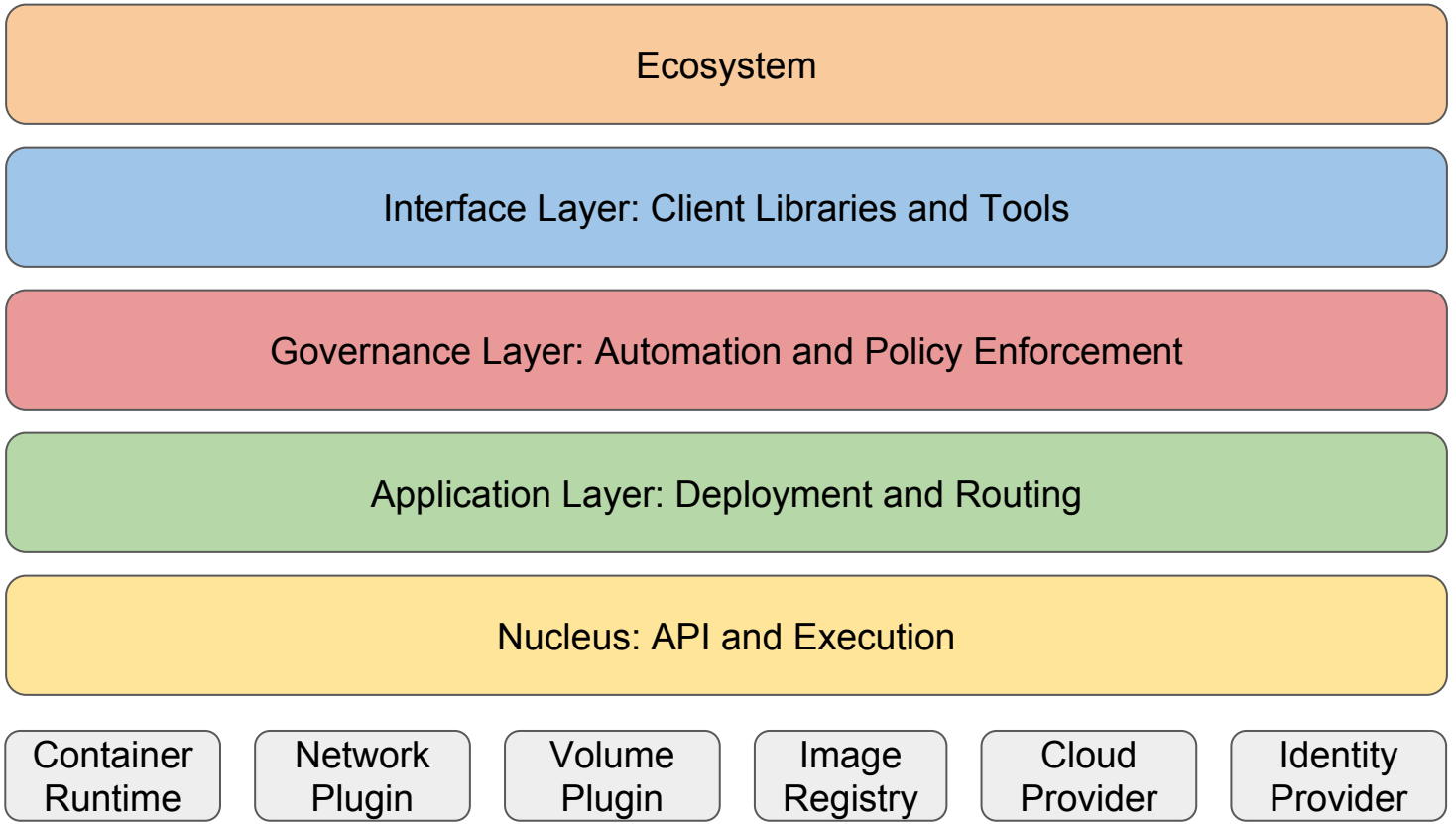
驱动“**当前状态 → 最终状态**”

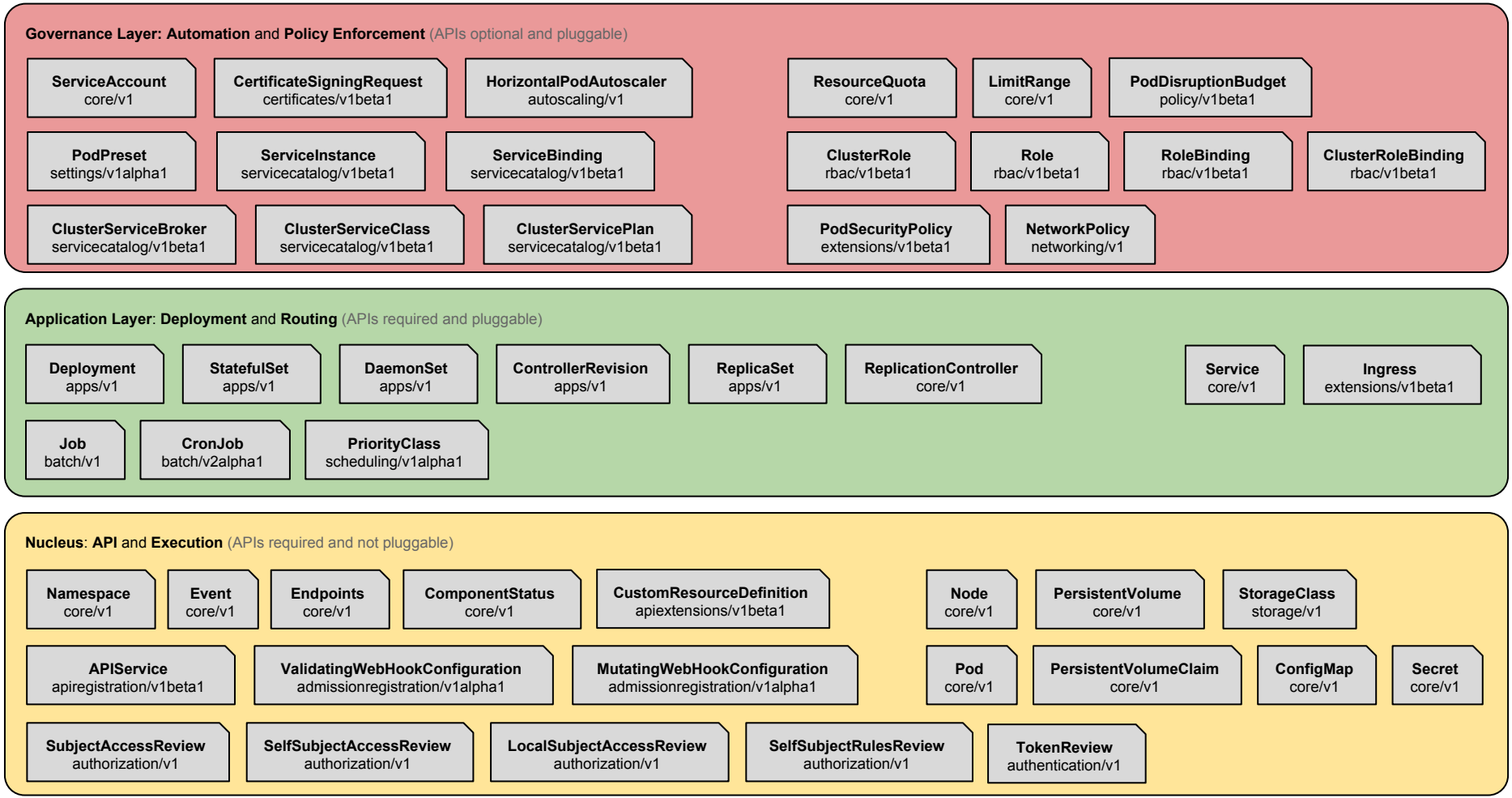
事件驱动，而非中心化编排。

其它方式：编排，状态机，共享存储，事件总线，点对点。



K8s层级


Ecosystem
Interface Layer: Client Libraries and Tools
Governance Layer: Automation and Policy Enforcement
Application Layer: Deployment and Routing
Nucleus: API and Execution
Container Runtime
Network Plugin
Volume Plugin
Image Registry
Cloud Provider
Identity Provider


Governance Layer: Automation and Policy Enforcement (APIs optional and pluggable)
ServiceAccount core/v1
CertificateSigningRequest certificates/v1beta1
HorizontalPodAutoscaler autoscaling/v1
ResourceQuota core/v1
LimitRange core/v1
PodDisruptionBudget policy/v1beta1
PodPreset settings/v1alpha1
ServiceInstance servicecatalog/v1beta1
ServiceBinding servicecatalog/v1beta1
ClusterRole rbac/v1beta1
Role rbac/v1beta1
RoleBinding rbac/v1beta1
ClusterRoleBinding rbac/v1beta1
ClusterServiceBroker servicecatalog/v1beta1
ClusterServiceClass servicecatalog/v1beta1
ClusterServicePlan servicecatalog/v1beta1
PodSecurityPolicy extensions/v1beta1
NetworkPolicy networking/v1
Application Layer: Deployment and Routing (APIs required and pluggable)
Deployment apps/v1
StatefulSet apps/v1
DaemonSet apps/v1
ControllerRevision apps/v1
ReplicaSet apps/v1
ReplicationController core/v1
Service core/v1
Ingress extensions/v1beta1
Job batch/v1
CronJob batch/v2alpha1
PriorityClass scheduling/v1alpha1
Nucleus: API and Execution (APIs required and not pluggable)
Namespace core/v1
Event core/v1
Endpoints core/v1
ComponentStatus core/v1
CustomResourceDefinition apiextensions/v1beta1
Node core/v1
PersistentVolume core/v1
StorageClass storage/v1
APIService apiregistration/v1beta1
ValidatingWebHookConfiguration admissionregistration/v1alpha1
MutatingWebHookConfiguration admissionregistration/v1alpha1
Pod core/v1
PersistentVolumeClaim core/v1
ConfigMap core/v1
Secret core/v1
SubjectAccessReview authorization/v1
SelfSubjectAccessReview authorization/v1
LocalSubjectAccessReview authorization/v1
SelfSubjectRulesReview authorization/v1
TokenReview authentication/v1

K8s工作负载类型

Deployments
StatefulSets
DaemonSet

Jobs - Run to Completion
CronJob

Deployments

A Deployment controller provides declarative updates for Pods and ReplicaSets.

You describe a desired state in a Deployment object, and the Deployment controller changes the actual state to the desired state at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

Service

A Kubernetes Service is an abstraction which defines a logical set of Pods and a policy by which to access them - sometimes called a micro-service.

The set of Pods targeted by a Service is (usually) determined by a Label Selector.

Resource - Pod

```
kubectl apply -f - <<eof
apiVersion: v1
kind: Pod
metadata:
  name: busybox
  labels:
    env: test
spec:
  containers:
  - name: busybox
    image: reg.qianbao-inc.com/k8s/busybox
    command: [ "/bin/sh", "-c", "ls /etc/config/; sleep 36000" ]
    imagePullPolicy: IfNotPresent
eof
```

Resource - Deployment

```
kubectl create -f - <<eof
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: busybox
  labels:
    app: busybox
spec:
  replicas: 3
  selector:
    matchLabels:
      app: busybox
  template:
    metadata:
      labels:
        app: busybox
    spec:
      containers:
        - name: busybox
          image: reg.qianbao-inc.com/k8s/busybox
          command: [ "/bin/sh", "-c", "ls /etc/config/; sleep 36000" ]
eof
```

Demo of usage

v1.6版本的不足

1. 早期得出结论，考虑过升级k8s版本来解决遇到的问题，两个方案：
   a. 升小版本实现了（当时为了更快速的得到一个可用的集群，并且已 经用了一段时间，完成当时的上线需求）
   b. 升大版本（目前考 虑从根基上提供一个更 稳定的版本，同时作为初步备用方案，以及支持更多功能和周边新的方案）
2. Kube-dns问题
3. 新集群搭建复杂

k8s新版本v1.10

为什么v1.10

1. 已有多个厂商使用v1.10
2. v1.10系列功能进入稳定状态

新版本优势

1. 新增系列主要新功能
2. 多个主要特性默认开启（进入beta稳定状态）
3. 相关Bug修复

主要新功能1

ExternalName

https://kubernetes.io/docs/tutorials/kubernetes-basics/expose-intro/
v1.7 ExternalName - Exposes the Service using an arbitrary name (specified by externalName in the spec) by returning a CNAME record with the name. No proxy is used.

This type requires v1.7 or higher of kube-dns

HostAlias

https://kubernetes.io/docs/concepts/services-networking/add-entries-to-pod-etc-hosts-with-host-aliases/
HostAlias is only supported in 1.7+.
HostAlias support in 1.7 is limited to non-hostNetwork Pods because kubelet only manages the hosts file for non-hostNetwork Pods.
In 1.8, HostAlias is supported for all Pods regardless of network configuration.

主要新功能2

Custom-resource-definitions

https://kubernetes.io/docs/tasks/access-kubernetes-api/extend-api-custom-resource-definitions/

custom-resource-definitions

Make sure your Kubernetes cluster has a master version of 1.7.0 or higher

RBAC stable

https://kubernetes.io/docs/admin/authorization/rbac/

As of 1.8, RBAC mode is stable and backed by the rbac.authorization.k8s.io/v1 API.

To enable RBAC, start the apiserver with --authorization-mode=RBAC

Local volume

https://kubernetes.io/docs/concepts/storage/volumes/#local

local

FEATURE STATE: Kubernetes v1.7 alpha

This alpha feature requires the PersistentLocalVolumes feature gate to be enabled.

主要新功能3

Cron-jobs (v1.8)

Note: CronJob resource in batch/v2alpha1 API group has been deprecated starting from cluster version 1.8.
https://kubernetes.io/docs/concepts/workloads/controllers/cron-jobs/
For previous versions of cluster (< 1.8) you need to explicitly enable batch/v2alpha1 API by passing --runtime-config=batch/v2alpha1=true to the API server You need a working Kubernetes cluster at version >= 1.8 (for CronJob)

https://kubernetes.io/docs/search/?q=feature%20state

Proxy-mode: ipvs (v1.9)

FEATURE STATE: Kubernetes v1.9 beta https://kubernetes.io/docs/concepts/services-networking/service/#proxy-mode-ipvs

主要新功能4

Pod Priority and Preemption (v1.9)

FEATURE STATE: Kubernetes v1.9 alpha https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/

Persistent Volume Claim Protection (v1.9)

FEATURE STATE: Kubernetes v1.9 alpha https://kubernetes.io/docs/concepts/storage/persistent-volumes/

Pod dns config (v1.9)(beta in v1.10)

https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/#a-records
Pod's DNS Config
Kubernetes v1.9 introduces an Alpha feature (Beta in v1.10) that allows users more control on the DNS settings for a Pod. This feature is enabled by default in v1.10. To enable this feature in v1.9, the cluster administrator needs to enable the CustomPodDNS feature gate on the apiserver and the kubelet, for example, "--feature-gates=CustomPodDNS=true,...

K8s v1.10系列功能进入稳定状态

Feature	Default	Stage	Since
CSIPersistentVolume	true	Beta	1.10
CustomPodDNS	true	Beta	1.10
CustomResourceSubresources	false	Alpha	1.10
DevicePlugins	true	Beta	1.10
DynamicVolumeProvisioning	true	GA	1.8
Initializers	false	Alpha	1.7
LocalStorageCapacityIsolation	true	Beta	1.10
MountPropagation	true	Beta	1.10
PersistentLocalVolumes	true	Beta	1.10
StorageObjectInUseProtection	true	Beta	1.10
SupportIPVSProxyMode	true	Beta	1.10
VolumeScheduling	true	Beta	1.10

https://kubernetes.io/docs/reference/command-line-tools-reference/#feature-gates

Bug修复数

版本	关闭issue数
v1.7	358
v1.8	194
v1.9	99
v1.10	118

k8s新版本劣势

1. 没有足够的试验
2. 可能需要平台做接口版本变化调整(及多集群支持)

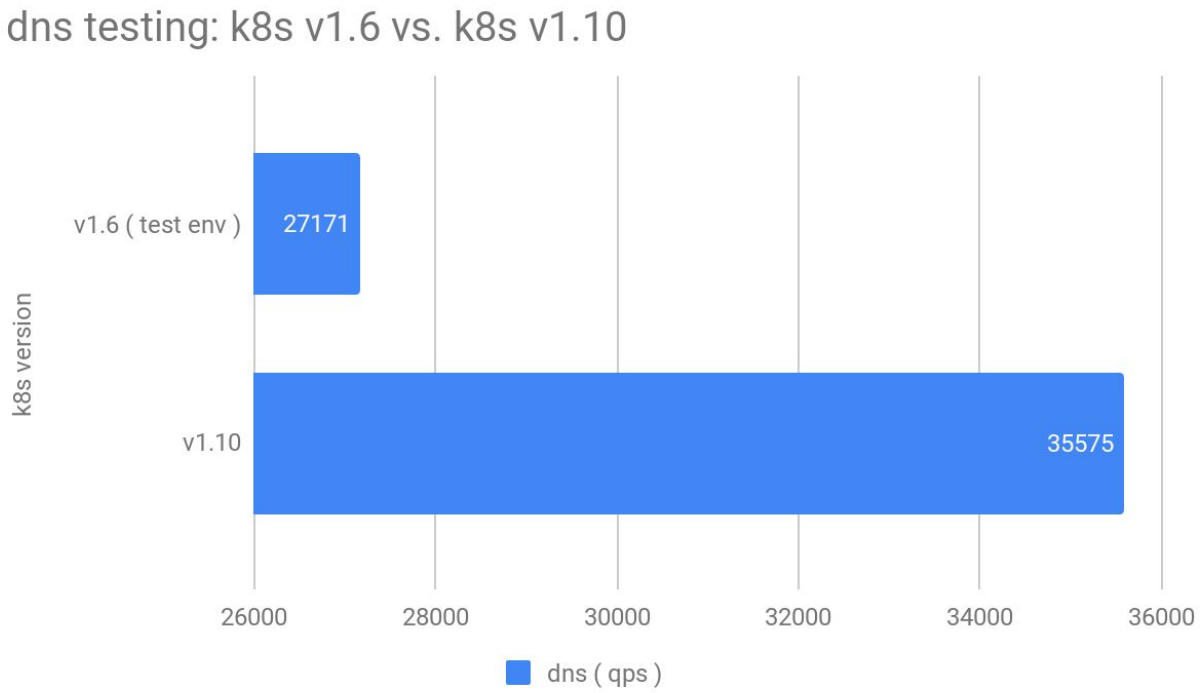
DNS解决方案

1. 采用CoreDNS替代原来的KubeDNS
2. DNS性能提升23%

注:由于版本升级，不仅是性能的提升也包含bug的修复。

k8s version	dns解决方案
v1.6	KubeDNS-1.14.3
v1.10	CoreDNS-1.0.6

DNS性能提升


dns testing: k8s v1.6 vs. k8s v1.10
k8s version
v1.6 (test env)
27171
v1.10
35575
26000
28000
30000
32000
34000
36000
dns (qps)

网络解决方案--Kube-Router1

All things from a single DaemonSet/Binary. It doesn't get any easier.

Kube-router is also a purpose built solution for Kubernetes.

Use standard Linux networking stack and toolset.

There is no overlays or SDN pixie dust, but just plain good old Linux networking. So its lot leaner.

Kube-router is being used in several production clusters by diverse set of users ranging from financial firms, gaming companies to universities.

https://github.com/cloudnativelabs/kube-router#project-status

网络解决方案--Kube-Router2

IPVS/LVS Service Proxy

Kube-router uses battle-tested Linux LVS/IPVS to provide a service proxy and provides rich set of scheduling options and enables advanced use-cases like DSR.

Pod Networking

kube-router handles Pod networking efficiently with direct routing thanks to the BGP protocol and the GoBGP Go library.

Network Policy

Kube-router fully support Network Policy semantics. It uses ipsets with iptables to enforce network policies but have as little performance impact on your cluster as possible.

Network Load Balancer

Kube-router has the ability to advertise service VIP's to L3 fabric BGP peers. So you can do network load balancing with ECMP.

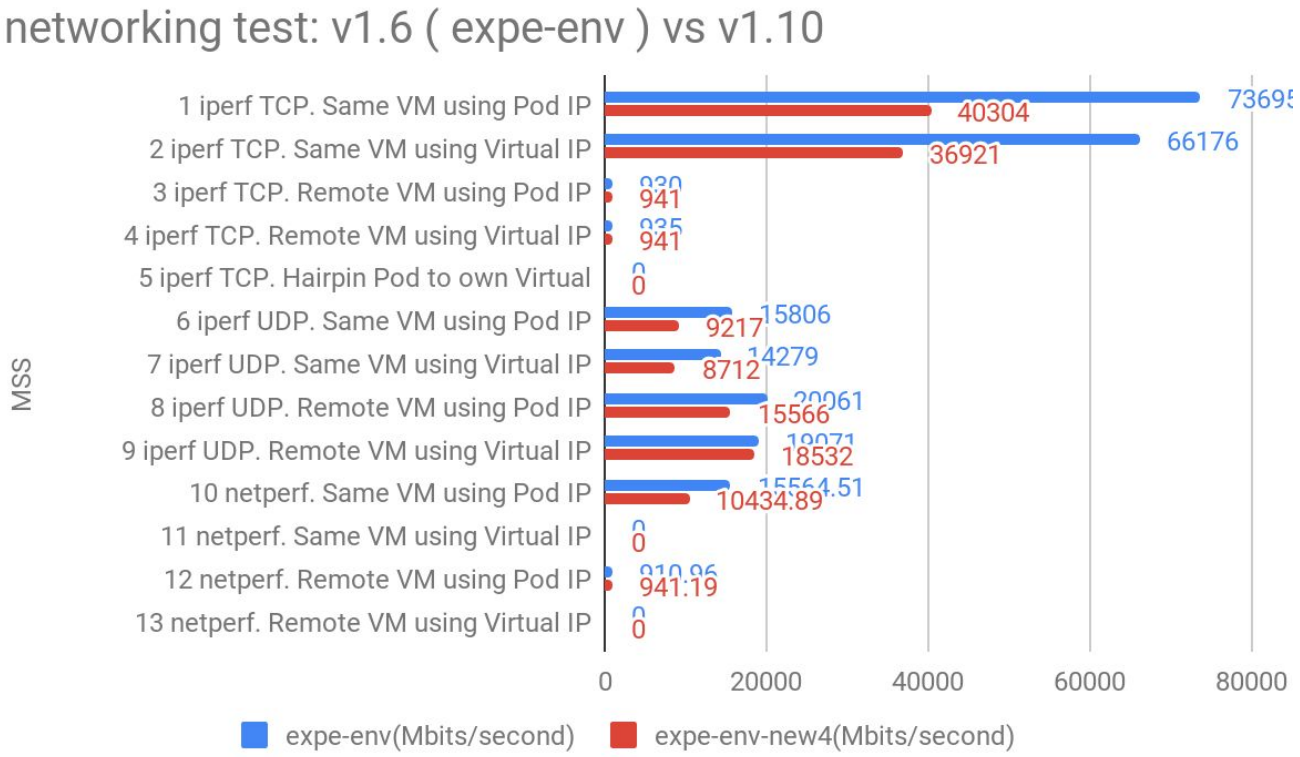
Small footprint

Although it does the work of several of its peers in one binary, kube-router does it all with a relatively tiny code base. Easy to hack-up and maintain.

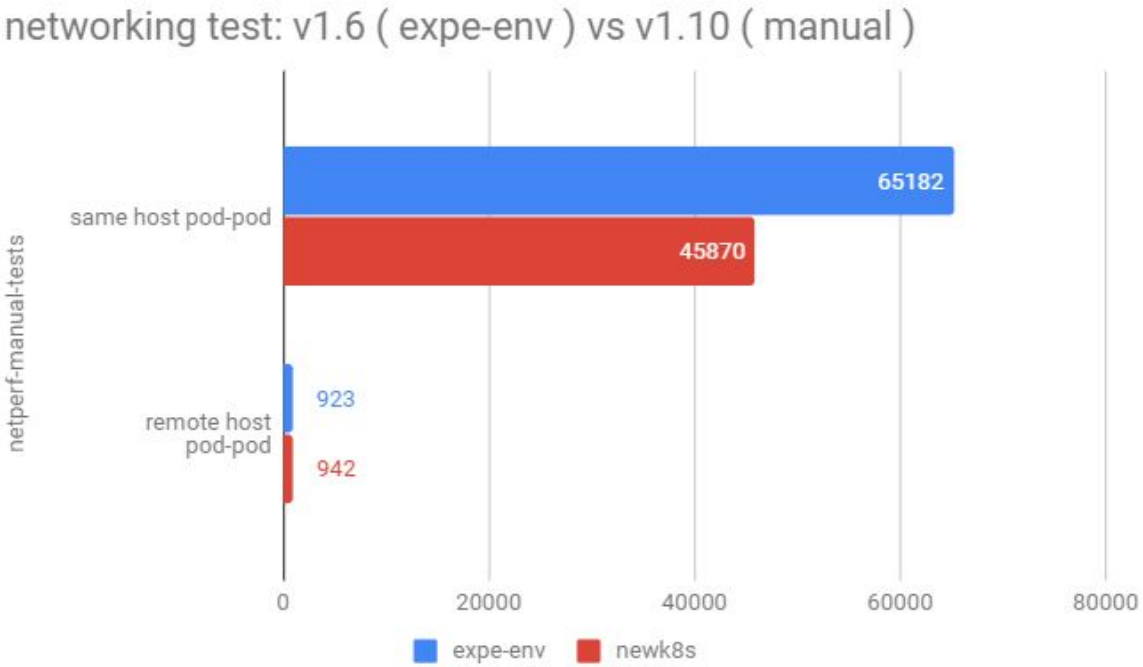
High performance

A primary motivation for kube-router is performance. The combination of BGP for inter-node Pod networking and IPVS for load balanced proxy Services is a perfect recipe for high-performance cluster networking at scale.

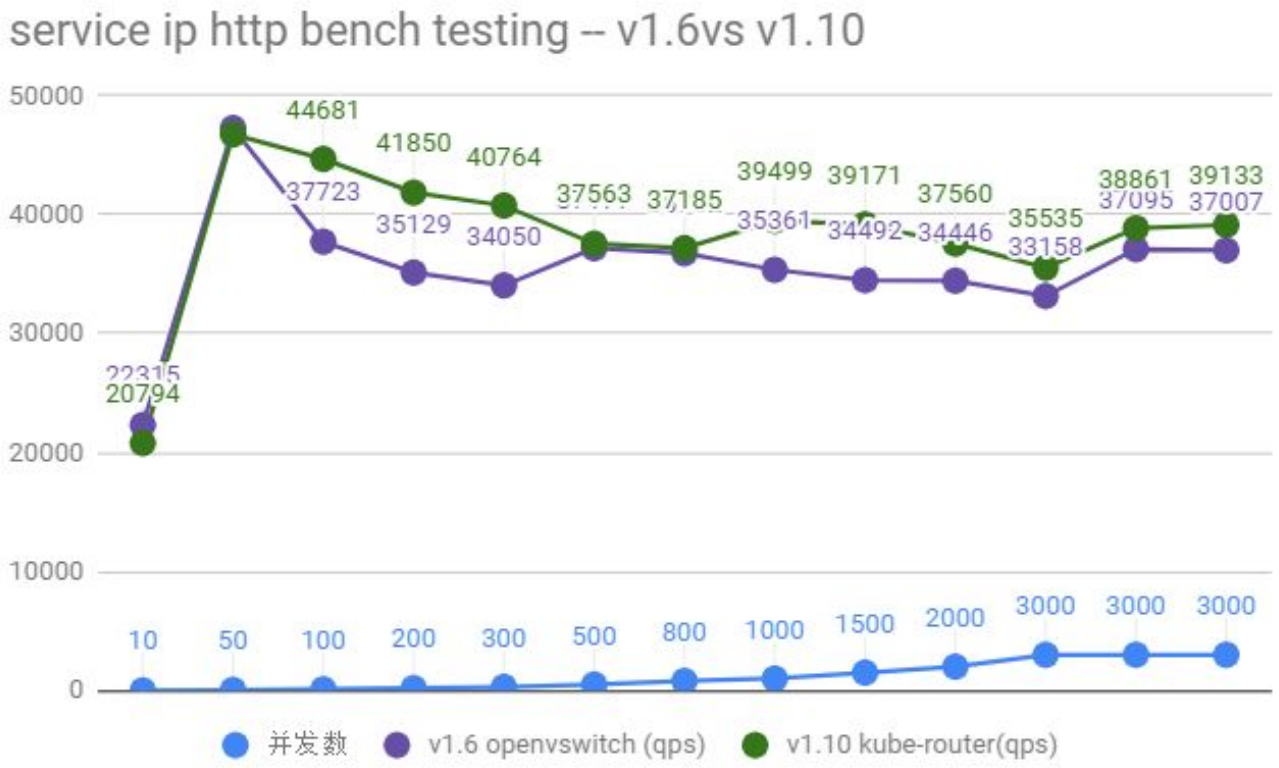
Networking测试结果1



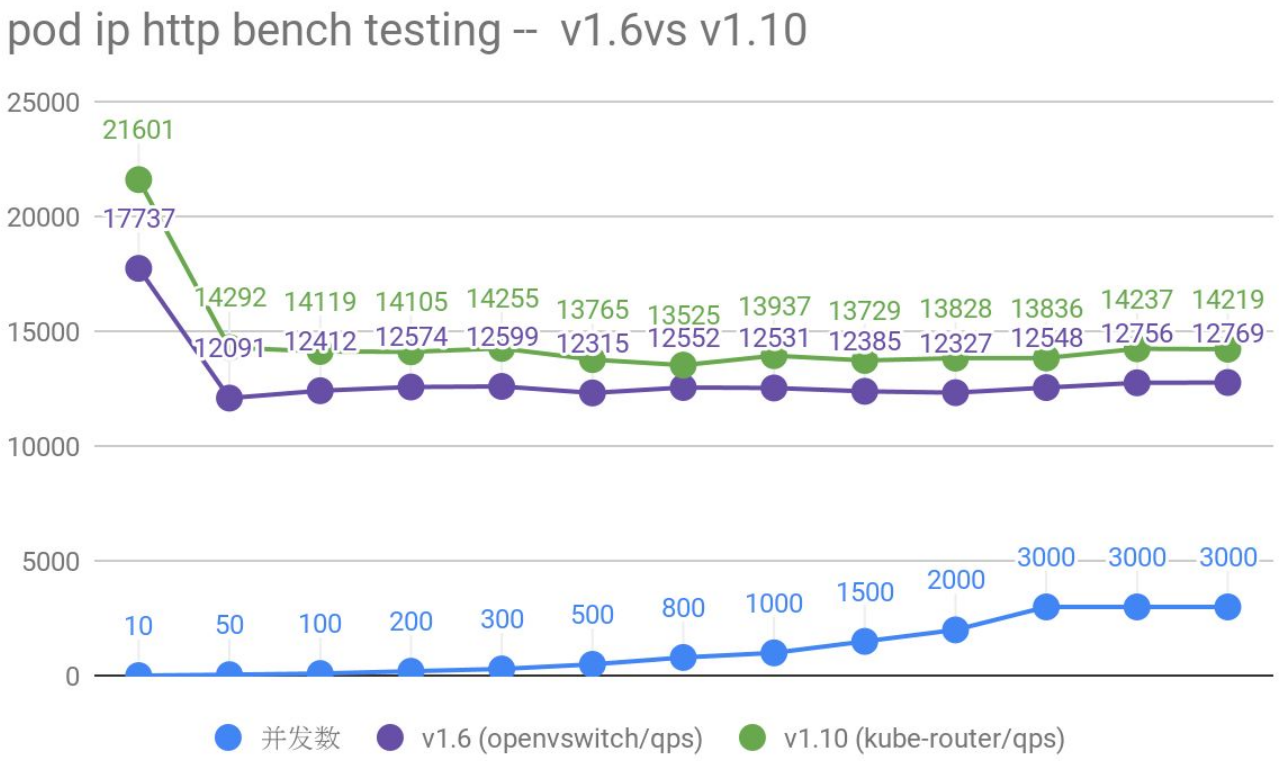
Networking测试结果2



HTTP测试结果1


service ip http bench testing -- v1.6vs v1.10
50000
40000
30000
20000
10000
0
44681
41850
40764
37723
35129
34050
37563
37185
39499
39171
37560
35361
34492
34446
35535
33158
38861
37095
39133
37007
22315
20794
10
50
100
200
300
500
800
1000
1500
2000
3000
3000
3000
并发数
v1.6 openvswitch (qps)
v1.10 kube-router(qps)

HTTP测试结果2


pod ip http bench testing -- v1.6vs v1.10
25000
20000
15000
10000
5000
0
21601
17737
14292
14119
14105
14255
13765
13525
13937
13729
13828
13836
14237
14219
12091
12412
12574
12599
12315
12552
12531
12385
12327
12548
12756
12769
10
50
100
200
300
500
800
1000
1500
2000
3000
3000
3000
并发数
v1.6 (openvswitch/qps)
v1.10 (kube-router/qps)

Networking测试结果总结

1. 跨节点网络通信有提升20M左右
2. 同主机不同Pod通信有下降（变成现在的45gps，目前考虑暂不会成为瓶颈，有待进一步调查？）
3. 新版本service访问性能提升5%（kube-router替换了kube-proxy的功能，同是服务访问 采用了ipvs替换了原来的iptables）

K8s 相关领域成果

Networking（kube-router更简化的方案）

http://issue.qianbao-inc.com/SRE/k8s/src/branch/master/networking

Ingress (kong 初步)

http://issue.qianbao-inc.com/SRE/k8s/src/branch/master/ingress

Storage (rook+ceph 初步)

http://issue.qianbao-inc.com/SRE/k8s/src/branch/master/storage

Testing, Monitoring, UI etc…（see http://issue.qianbao-inc.com/SRE/k8s ）

k8s新版本部署

减少了安装复杂度和时间

k8s version	method	tools	docs length
v1.6	manual	manual	20 pages
v1.10	half-automatic	kubeadm	about 1 page

k8s version	reference
v1.6	kubernetes 安装-new.pdf
v1.10	issue.qianbao-inc.com/SRE/k8s/install/kubeadm

Demo of creating cluster

当前k8s进度

以下项目已上至k8s生产平台（thanks to xigui）

1. SSO

一个新k8s版本（更稳定及支持更多功能）正在进行中，准备测试环境或预发布环境部署。

k8s未来工作

Workflow-Smooth

External Tolerance

Kubeadm-HA

Storage

And more…
http://issue.qianbao-inc.com/SRE/k8s/src/branch/master/co-work-design.md

总结

本次介绍了k8s的特性和架构及核心概念，及相应新版本的预研成果，同时提供了k8s当前的进度及相关的未来工作。

期望后续同事加入到k8s工作中来，让我们通过k8s为公司带来更大的效益。

参考链接

1. https://kubernetes.io/docs/reference/command-line-tools-reference/#feature-gates
2. https://github.com/kubernetes/community/blob/master/contributors/design-proposals/architecture/architecture.md
3. https://docs.google.com/presentation/d/1BoxFeENJcINgHbKfygXpXROchiRO2LBT-pzdaOFr4Zg/edit#slide=id.g3b76446f7d_0_210 CNCF Overview
4. https://www.cncf.io/cncf-annual-report-2017/
5. https://docs.google.com/presentation/d/1oPZ4rznkBe86O4rPwD2CWgqgMuaSXguIBHIE7Y0TKVc/edit#slide=id.p Kubernetes Architecture
6. https://www.kube-router.io/

Thank you!