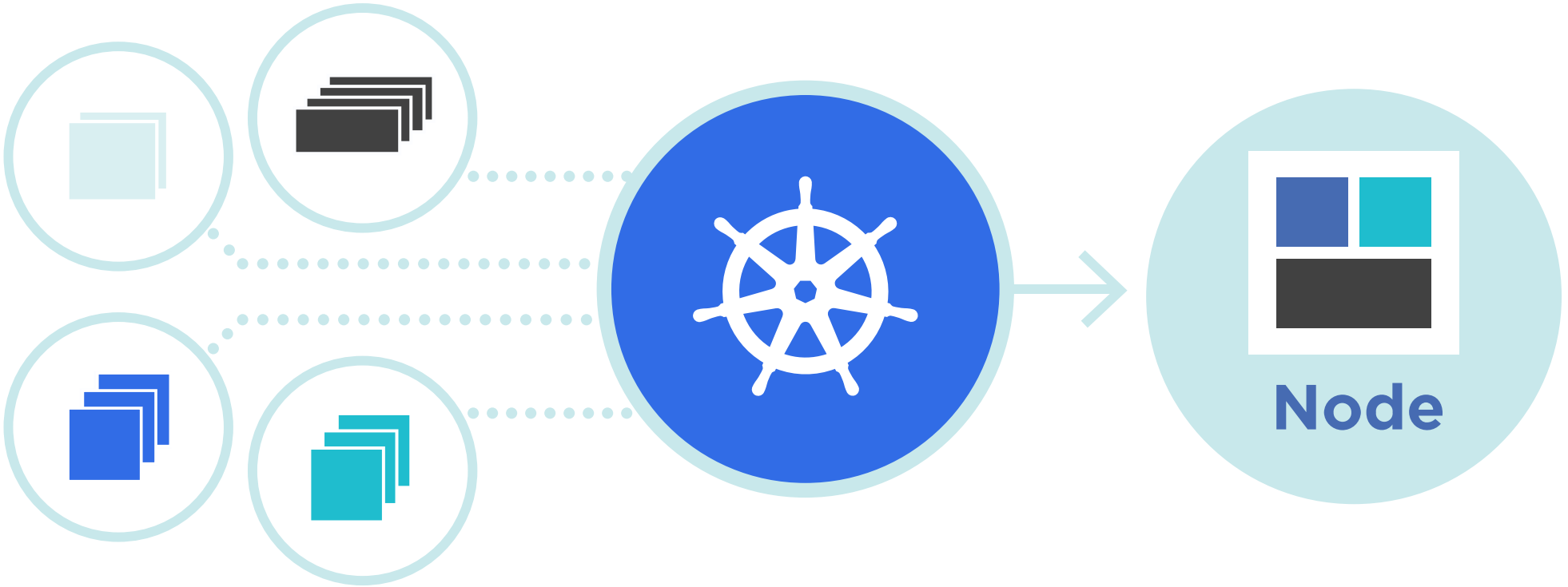
Kubernetes

实战与原理



K8S 介绍

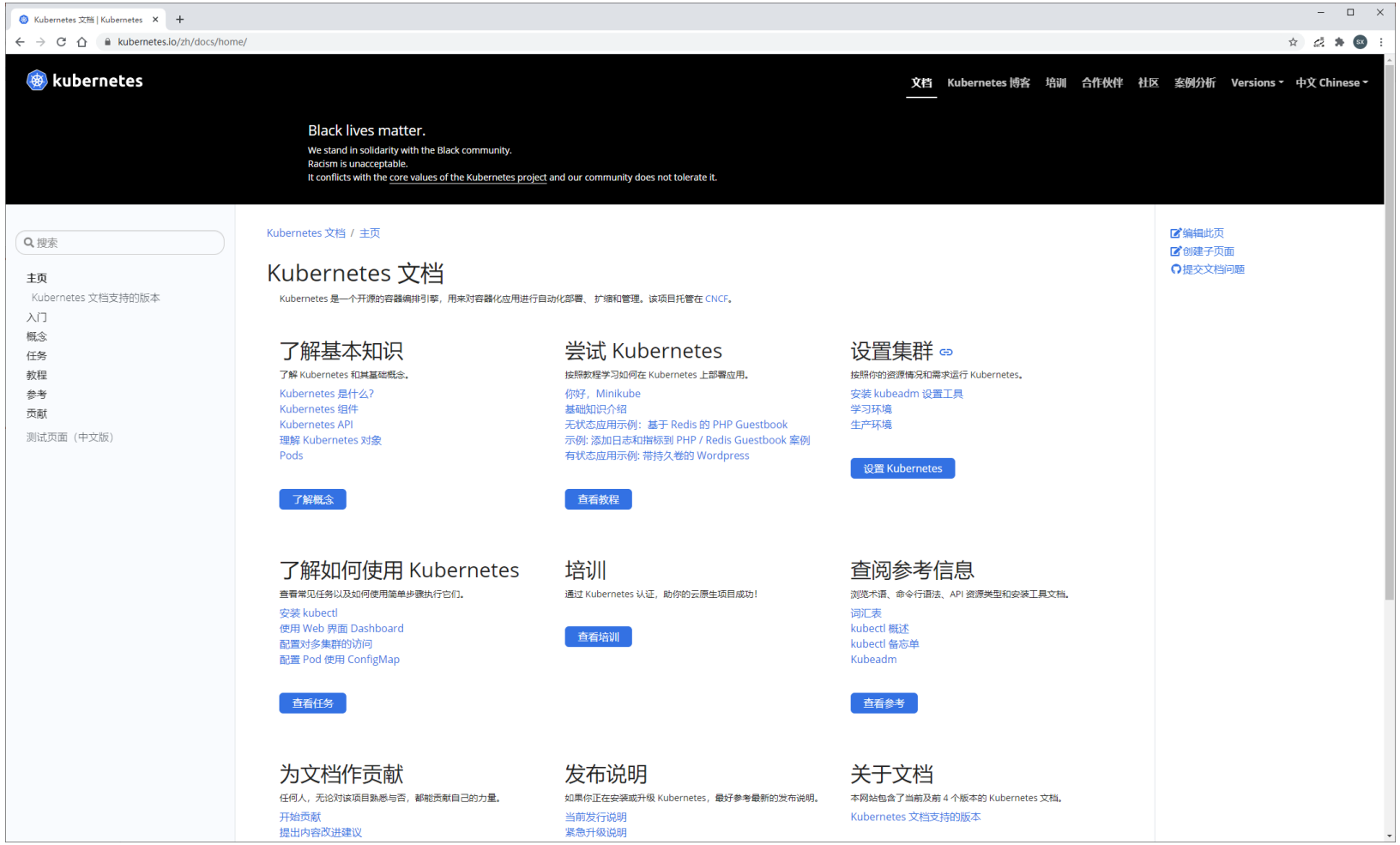
从他的名字开始

Kubernetes又名K8S
为什么？他中间有八个字母

K-ubernete-s

K8S 介绍

centos



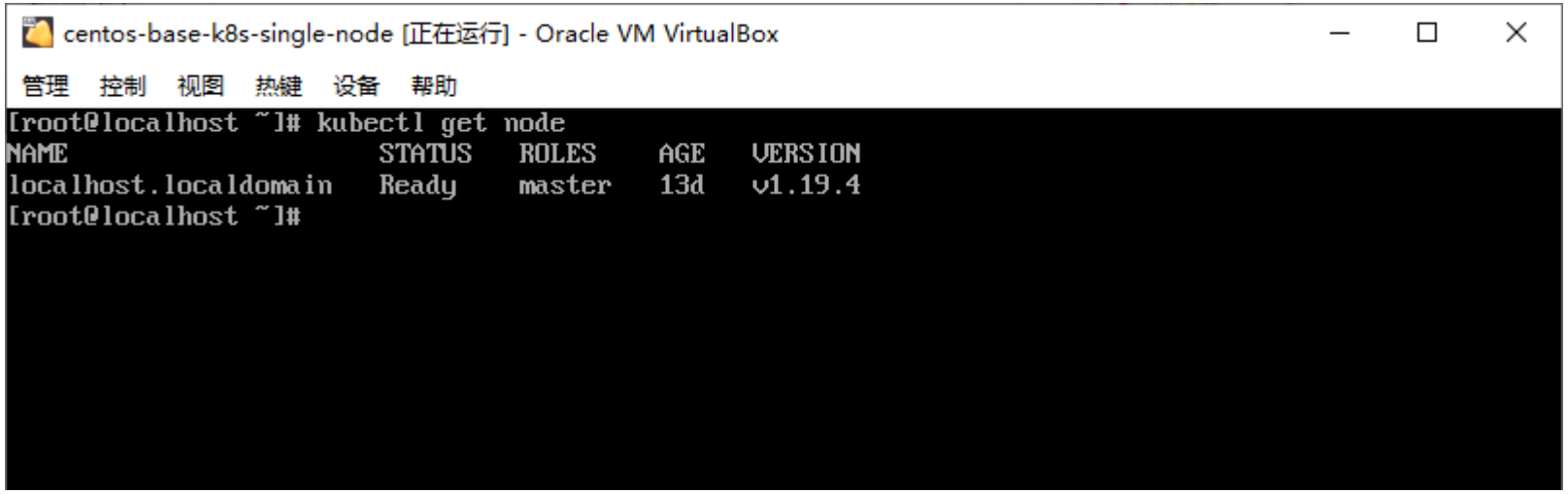
[中文官网链接]

这里几乎可以满足我们所有的要求。

K8S 安装

学习环境

基于虚拟机的安装，准备一台centos8，执行脚本，当发现以下输出中STATUS为Ready的时候，k8s单点安装完成



K8S 术语

Master

Master 是集群的控制节点，默认不负责工作负载，每个集群中至少有一个Master节点来负责整个集群的管理和控制。

进程	作用	备注
Kubernetes API Server	提供HTTP Rest接口，是所有资源CRUD唯一的入口	
Kubernetes Controller Manager	资源自动化控制中心	
Kubernetes Scheduler	负责Pod的调度	

K8S 术语

Node

一个节点不是Master就是Node了，他是集群的工作负载节点，会被Master分配一些容器，当Node宕机的时候，Master会把他的工作负载转移到其他Node

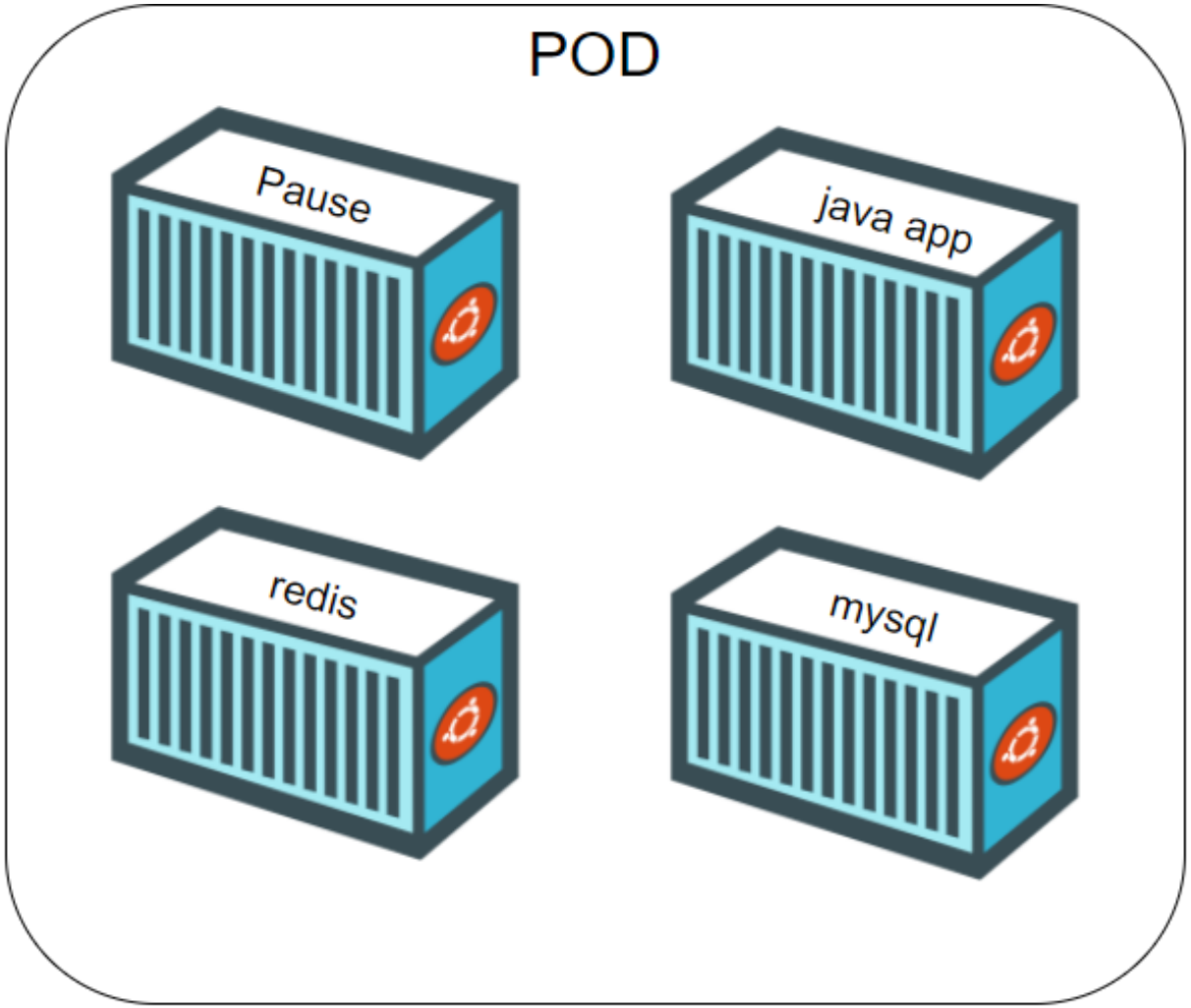
进程	作用	备注
Kubelet	负责Pod的创建、启停，与Master通信	
Kube-proxy	负责Kubernetes Service通信和负载均衡	
Docker Engine	-	

K8S 术语

Pod

前面我们一直提及到了Pod这个概念，他是一个抽象的概念，是一组容器构成的集合

右图是一个POD的例子，这个POD包含了4个容器，分别是Pause，Java App， Redis 和 Mysql



K8S 术语

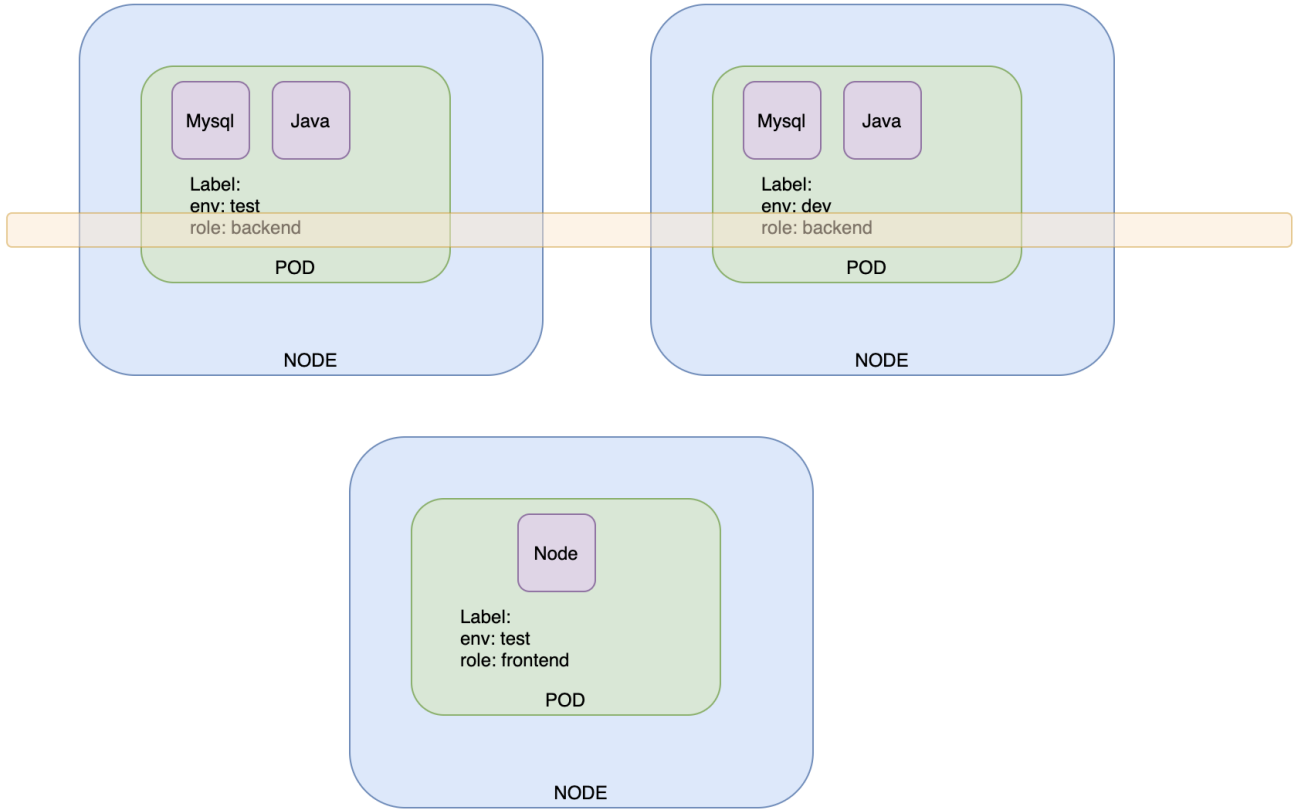
Label

Label : 标签

我们可以给K8S的所有资源打上标签，然后通过标签选择器来选择他们

右图是一个标签的例子，我们可以通过标签选择器选择role=backend的POD来获取上面的两个POD，

当然也可以选择env=test的情况，来选择其他POD

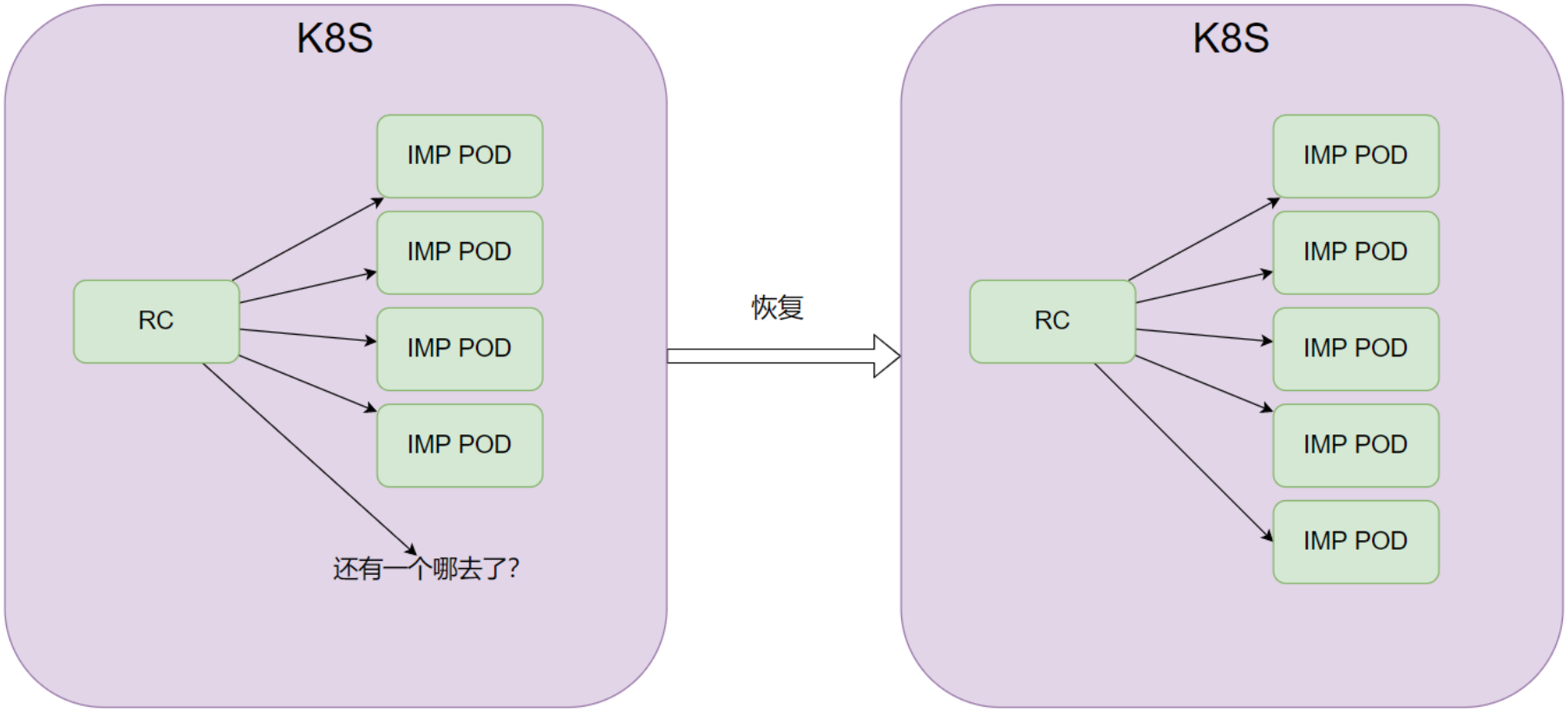


K8S 术语

RC/RS

Replication Controller（RC）： 副本控制器，只要我们定义了一个RC并提交到了K8S集群，当这个RC会通过Label选择器来监测某些指定的POD，当他们的数量少于预期值的时候，会触发副本重新创建。

Replication Set(RS) : RS是RC的改进，RC只可以选择K=V，RS可以使用K in [V1,V2...]



K8S 术语

Deployment

Deployment是高级资源，他指定了一个POD应该如何构建，由哪些Container组合而成，定义这个POD的副本数，并通过RS来管理，定义POD的滚动更新策略，执行回滚等操作，

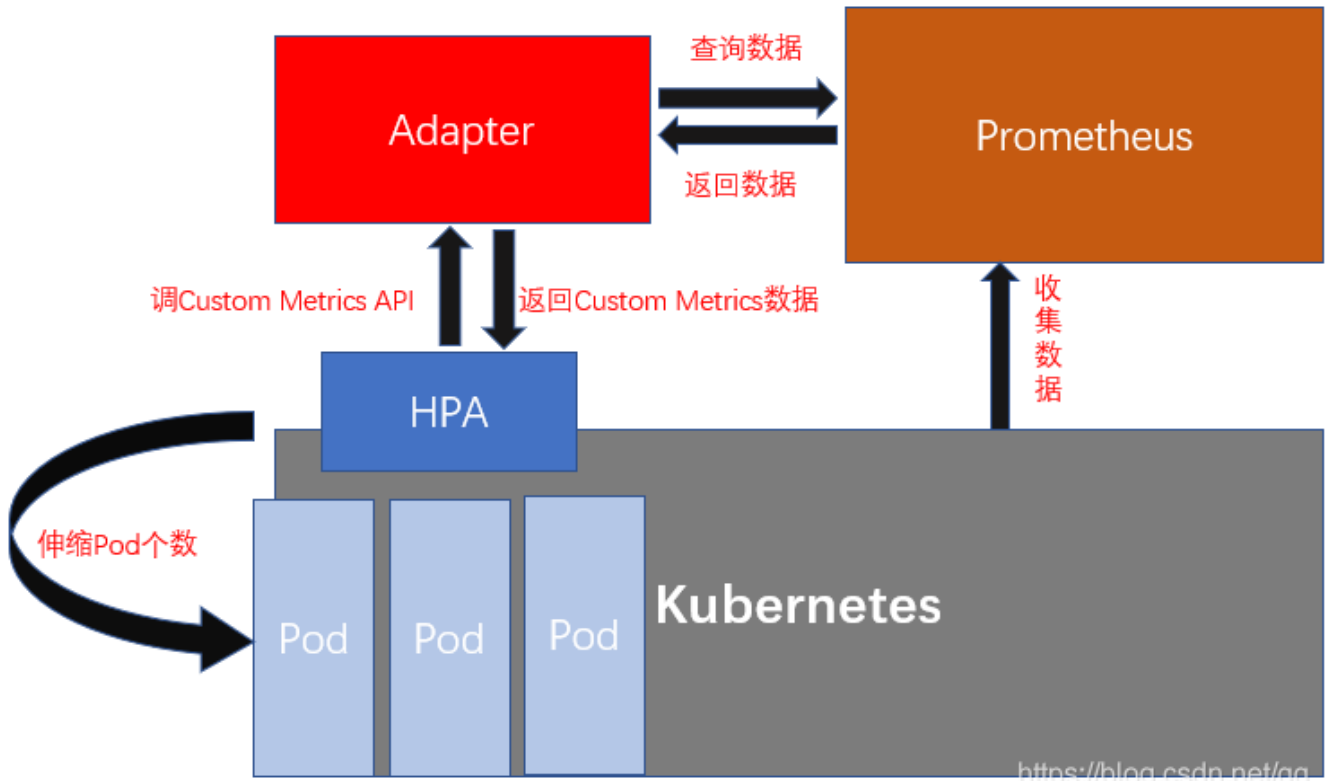
下图是使用K8S创建一个简单的Deployment，我们看到他创建了一个pod，一个deployment，以及一个replicaset

```
[→  ~ kubectl create deployment nginx --image=nginx:1.14
deployment.apps/nginx created
[→  ~ kubectl get all
NAME                         READY   STATUS              RESTARTS   AGE
pod/nginx-5658bdf5d4-vqrr4   0/1     ContainerCreating   0          5s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   2d9h

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx   0/1     1            0           6s

NAME                               DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-5658bdf5d4   1         1         0       6s
```

K8S 术语

HPA

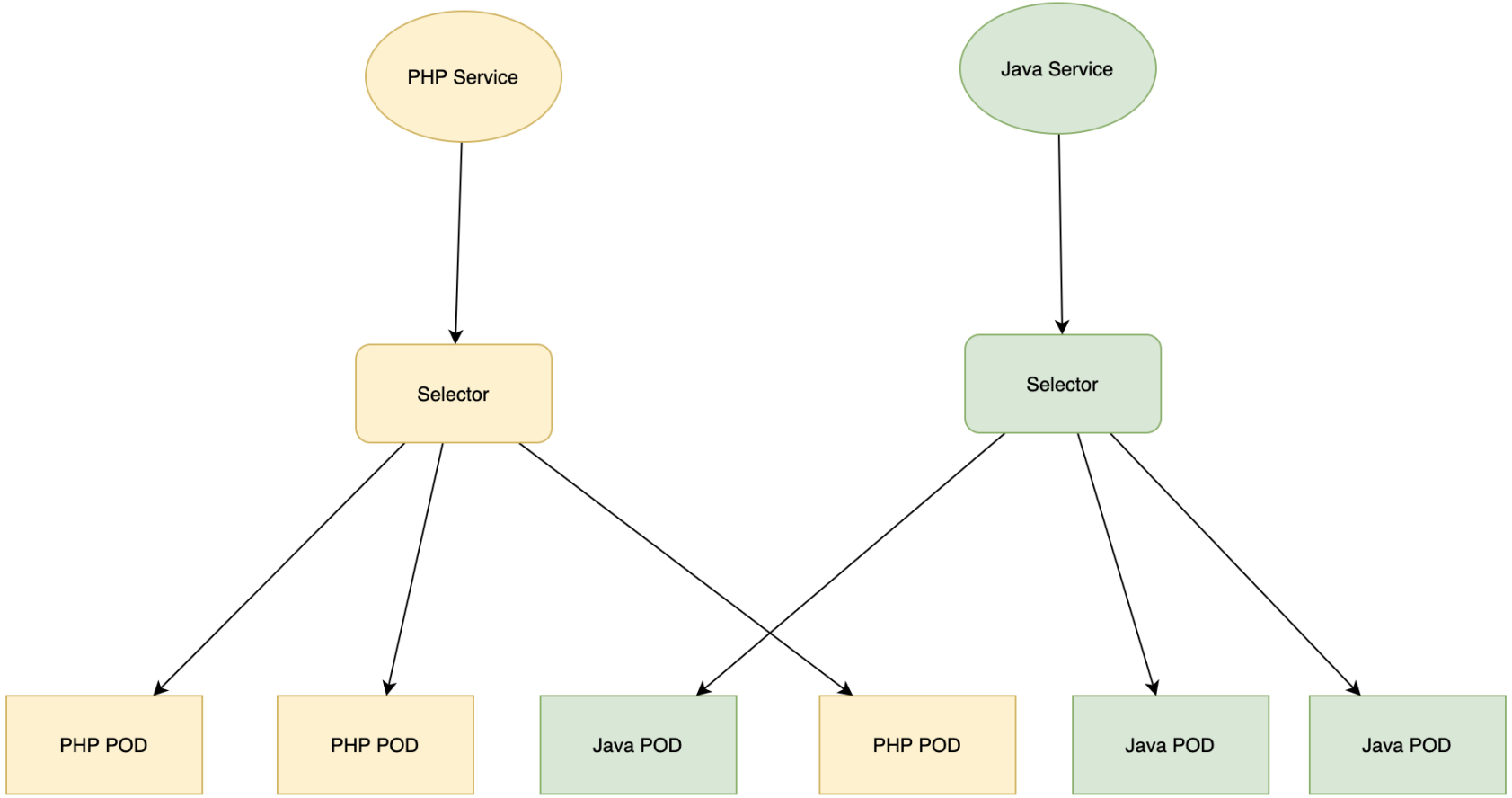
HPA是K8S中负责自动扩缩容的资源，会根据POD的CPU，内存，qps等信息进行弹性伸缩



https://blog.csdn.net/qq_17305249

K8S 术语

Service



K8S中一个应用可能会起很多个POD，我们在K8S外，要如何访问呢？

Service出现了，他负责暴露这些POD

K8S 实践

Pod实践

右边是K8S模板，几乎所有的K8S资源都要写成右边的形式。这里包含版本apiVersion， kind类型，metadata元数据（可以在里面为Deployment打标签）以及spec，是一些Deployment的详细信息

```
3   apiVersion: apps/v1
2   kind: Deployment
1   metadata:
4     name: vdata-imp-deployment
1     labels:
2       app: vdata-imp
3   spec: ···
```

下面是spec中的内容， replicas代表RS需要维护的副本数，selector代表RS需要监听的POD，template是副本不足的时候创建新POD的模板，stargetgy是POD升级的策略，

```
37  spec:
36    replicas: 1
35    selector:
34      matchLabels:
33        app: vdata-imp
32    template: ···
5     strategy: # 策略
4       type: RollingUpdate # 也可以是Recreate
3       rollingUpdate:
2         maxUnavailable: 50% # 滚动更新的时候的最大不可用pod数量， 可以是绝对数字或者比例10%
1         maxSurge: 50% # 动更新的时候的溢出的pod数量，也可以是绝对数字
44    progressDeadlineSeconds: 150 # 进度期限秒数，不懂是什么
1     minReadySeconds: 100 # 最短就绪时间， 容器创建多久以后被视为就绪
2     revisionHistoryLimit: 3 # 历史修订限制， 保留的rs的数量，这个数量会消耗etcd资源，rs删除了就不能回滚刀那个版本的Deployment了
```

Pod实践

右边是template， 定义了一个POD，包含标签+容器。

livenessProbe是存活探针，只要返回200-400，均视为存活，

```yaml
template:
  metadata:
    labels:
      app: vdata-imp
  spec:
    containers:
    - name: vdata-imp
      image: vdata-imp:98
      imagePullPolicy: IfNotPresent
      env:
      - name: ENV
        value: "env"
      ports:
      - containerPort: 80
      resources:
        limits:
          cpu: 0.3
          memory: 400Mi
        requests:
          cpu: 0.3
          memory: 300Mi
      livenessProbe:
        httpGet:
          path: /swagger-ui/
          port: 80
        initialDelaySeconds: 100
        periodSeconds: 3
```

Service实践

右边是Service的全部， 通过选择器选择POD，然后将port暴露到Service的targetPort， 暴露到Node的NodePort

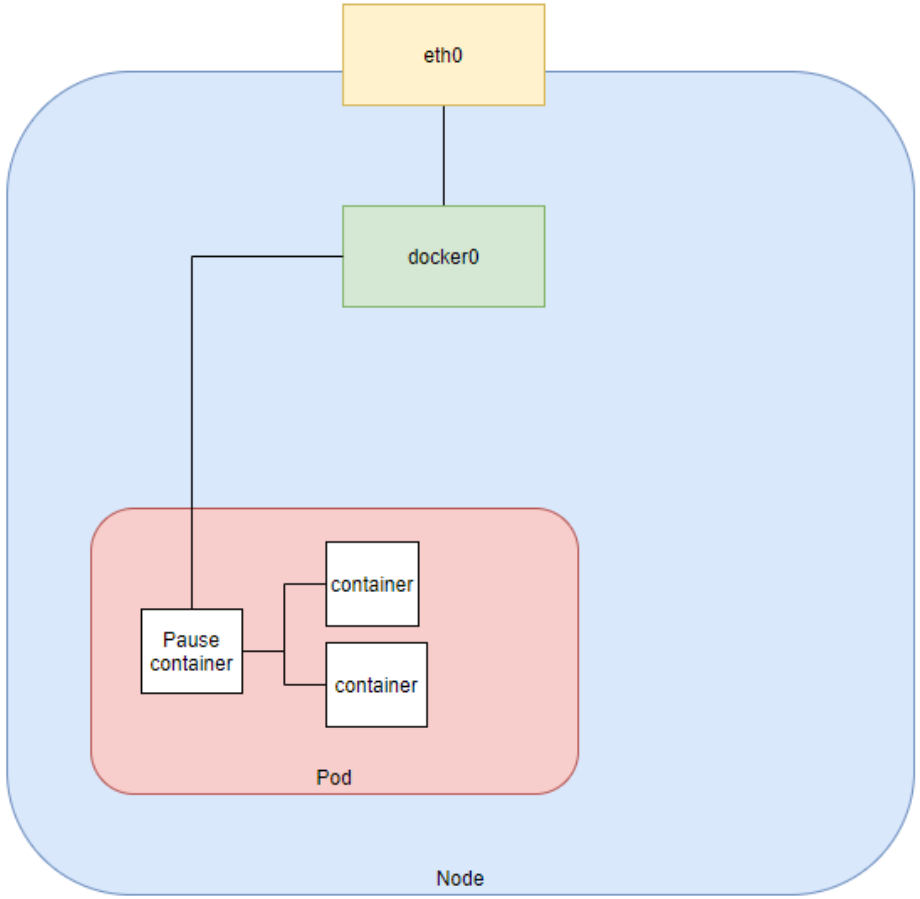
我们可以通过访问任何K8S任何一个Node的nodePort，来访问到vdata-imp

```yaml
apiVersion: v1
kind: Service
metadata:
  name: vdata-imp-service
spec:
  type: NodePort
  selector:
    app: vdata-imp
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30001
```

K8S 网络

POD网络

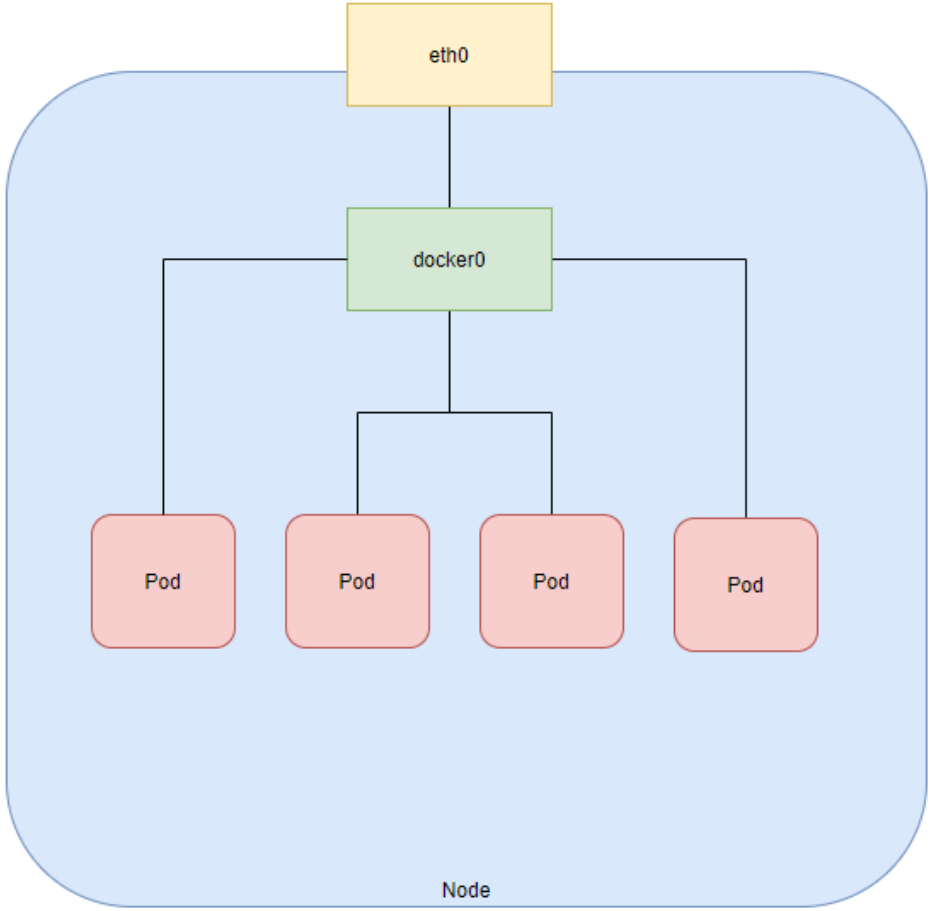
POD是容器的集合，POD中一定包含了一个Pause容器， 其他的容器共享Pause的网络命名空间，从而容器间可以通过localhost互相访问



K8S 网络

POD网络

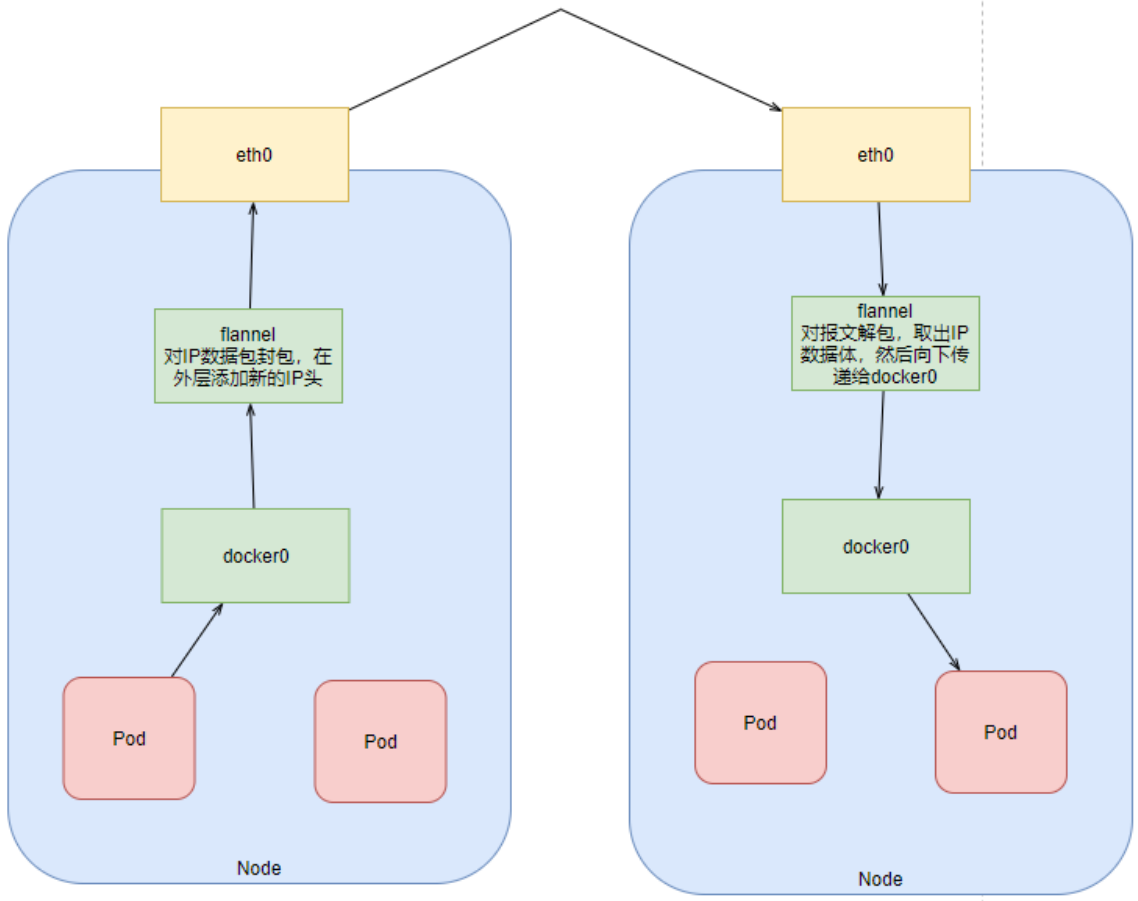
POD间通信， 如果POD在同一个Node，他们通过docker0网桥进行通信，这是docker提供的能力



K8S 网络

POD网络

POD间通信， 如果POD不在同一个Node，我们需要构建一个flannel覆盖网络，使用隧道的方式传递数据包

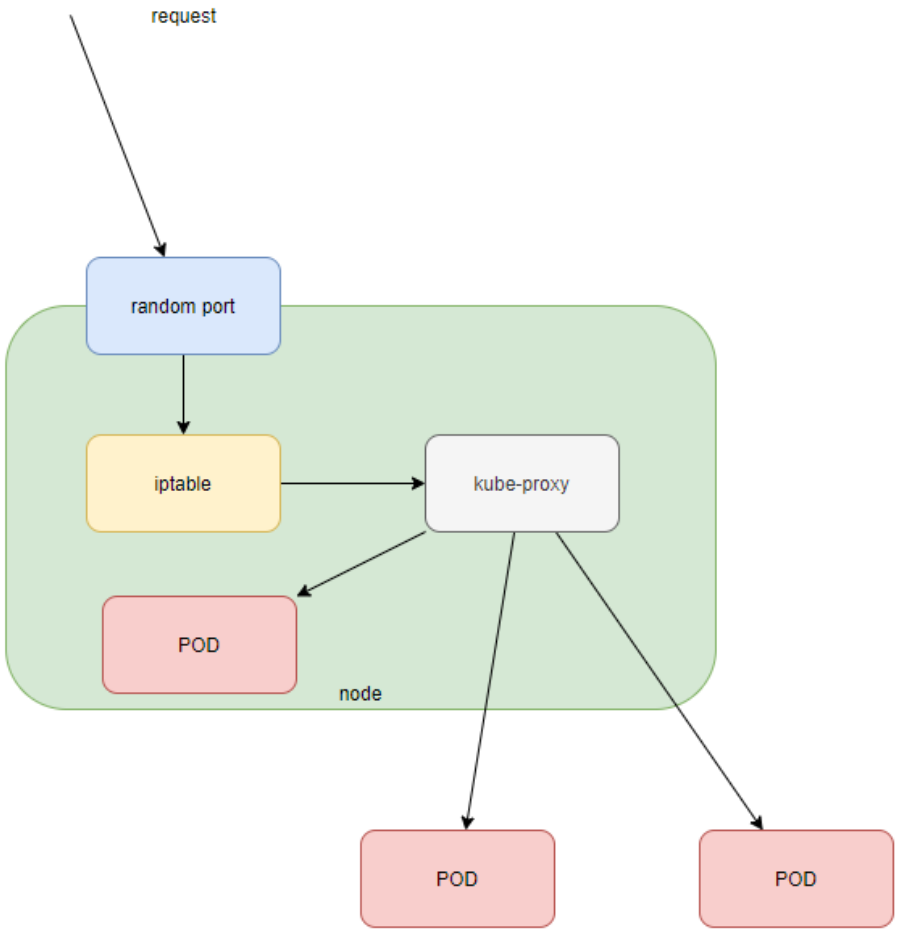


K8S 网络

Service网络

Userspace 模式:

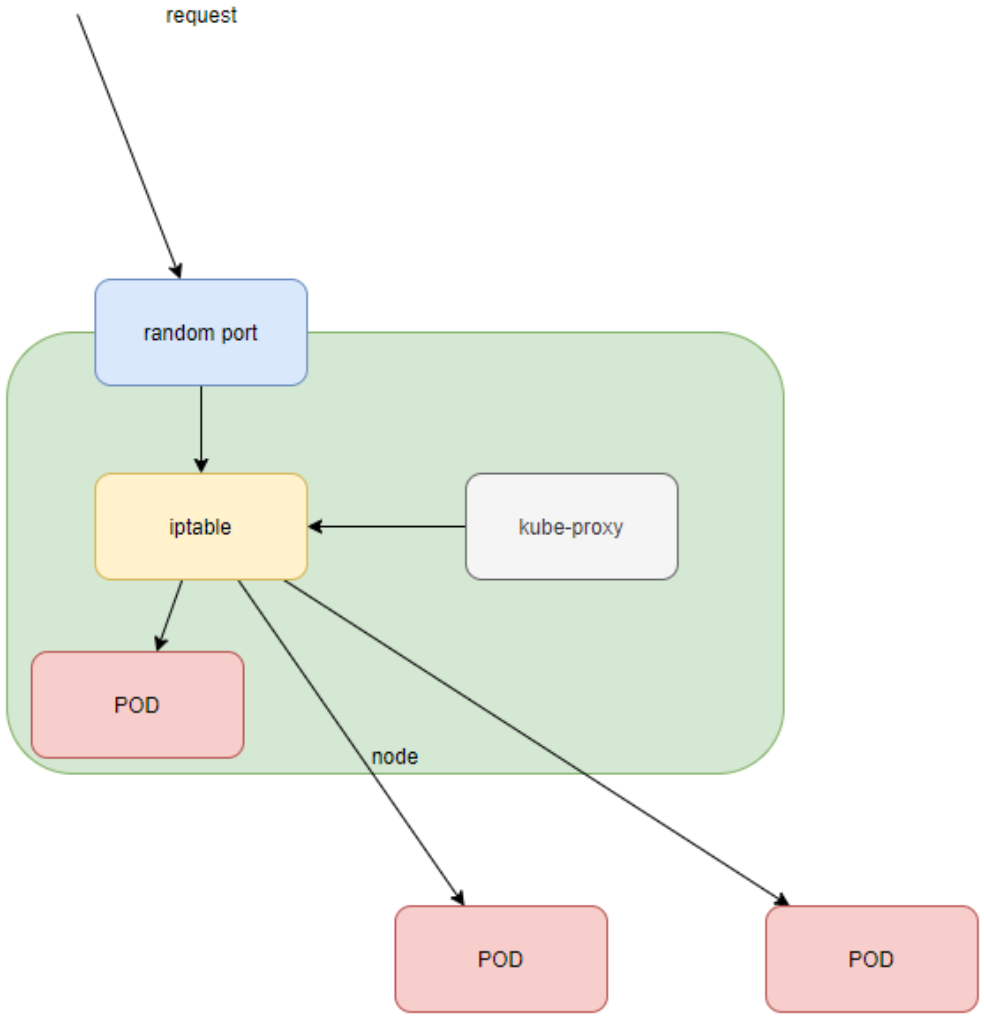
Node开启一个随机端口进行监听请求，使用iptable转发到kube-proxy, kube-proxu反向代理转发给POD， kube-proxy负责向Kube Api Server 查询，监听POD的变化



Service网络

Iptable 模式：

Kube-proxy不再负责转发，而是负责监控POD变化，然后直接写iptable，让iptable直接转发到对应的POD



K8S 安全

K8S 组件

Kube Scheduler

调度器的责任是分配POD到指定的Node，并将相关信息写入etcd，其中涉及到了预选策略和优选策略， 满足所有预选条件后使用任意一种优选策略来进行调度

预选条件
磁盘卷不可冲突
CPU内存要满足要求
标签选择器要满足
…

优选策略
资源利用率最低的Node得分高
手动指定的某些Node得分高
资源利用率最均衡的Node得分高
…

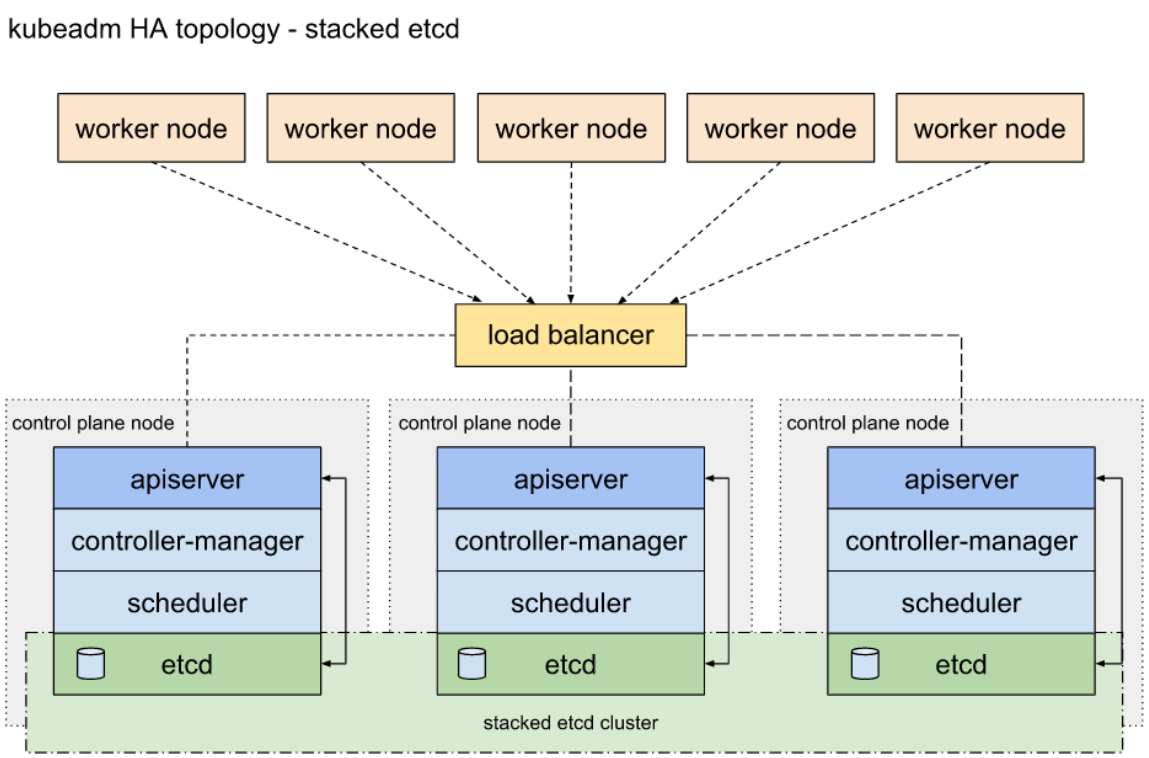
$$-\left(\frac{TotalCPU}{NodeCPU}+\frac{TotalMemery}{NodeMemery}\right)$$

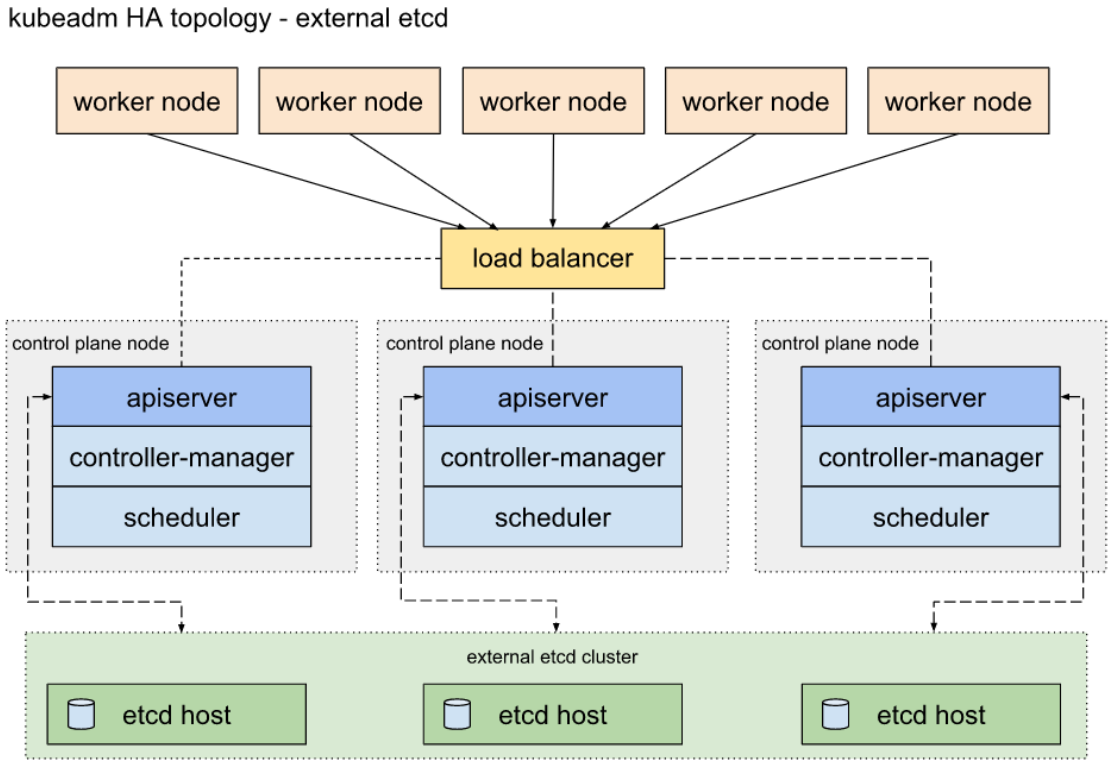
$$\left|\frac{TotalCPU}{NodeCPU}-\frac{TotalMemery}{NodeMemery}\right|$$

K8S 高可用

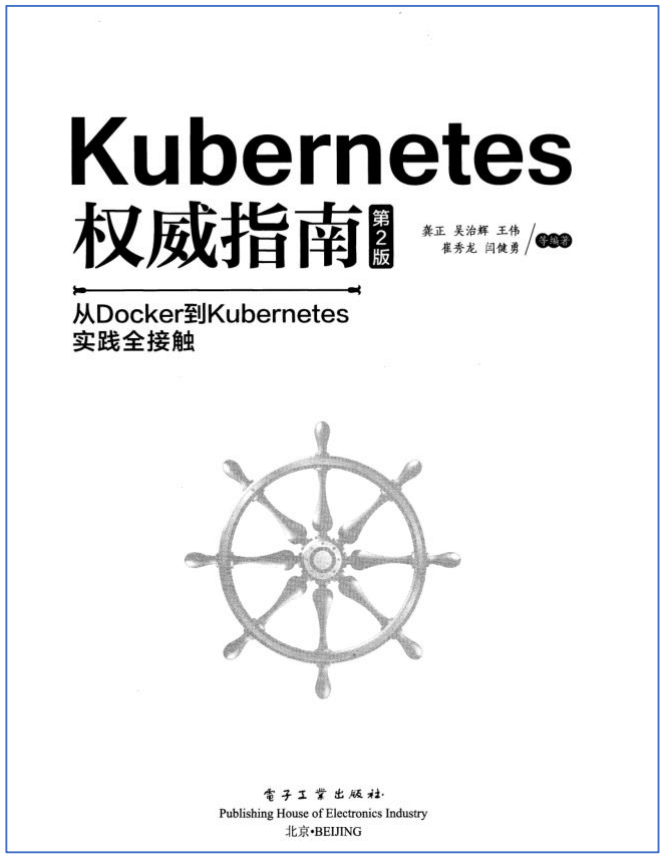
高可用

同一时刻只允许一个Master节点提供服务，当Master宕机以后，会依赖etcd触发简单的选主， 第一个在etcd创建leader信息的就是主节点，只有他提供服务，所有的Master都会定期检查etcd，如果发现没有主节点就会申请自己成为主节点。





参考文献


Kubernetes
权威指南 第2版
龚正 吴治辉 王伟
崔秀龙 闫健勇 著
从Docker到Kubernetes
实践全接触
電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING


Broadview
www.broadview.com.cn
Kubernetes
实战
吴龙辉 著
中国工信出版集团
電子工業出版社