Kubernetes那些事儿

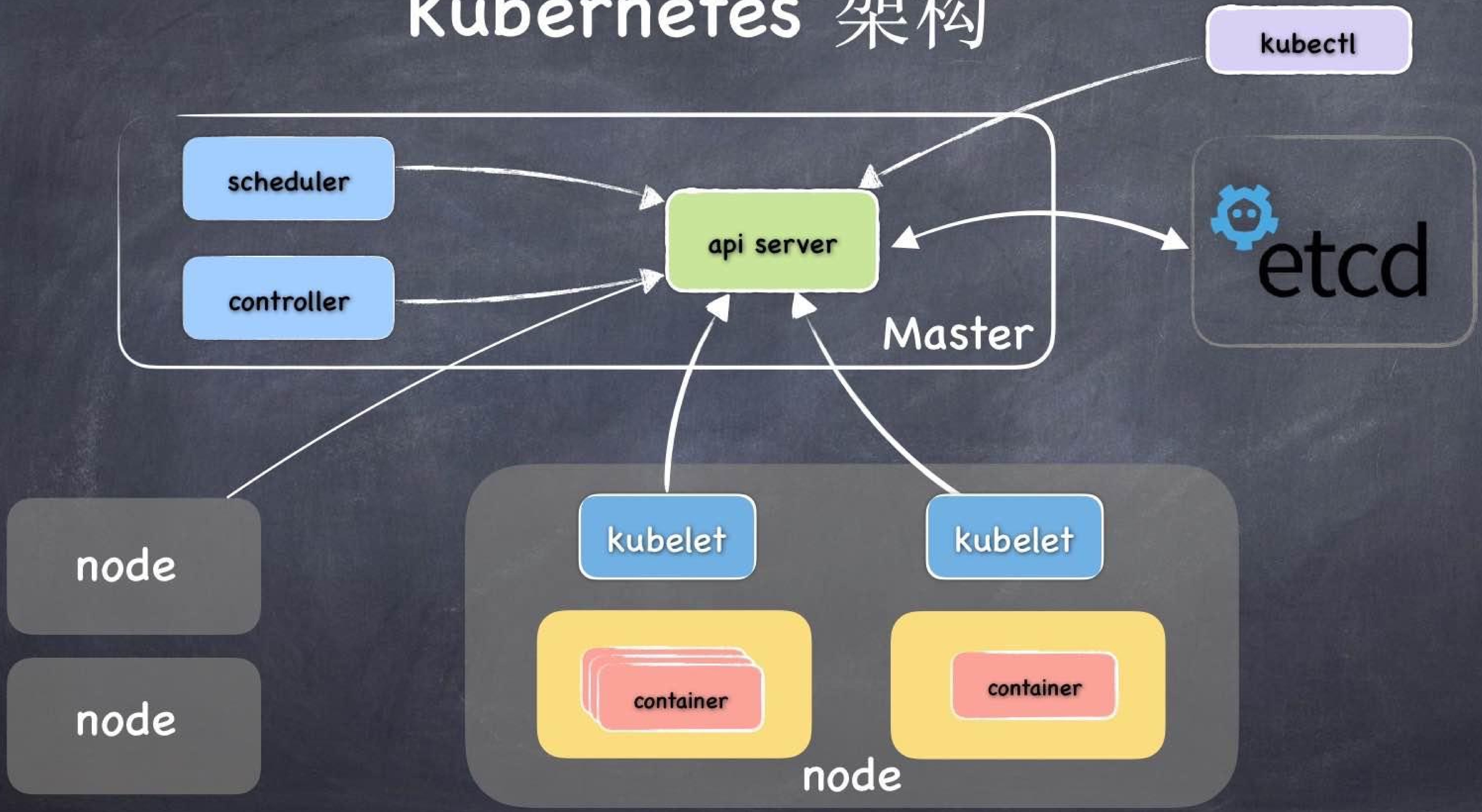
- xiaorui.cc

what is kubernetes ?

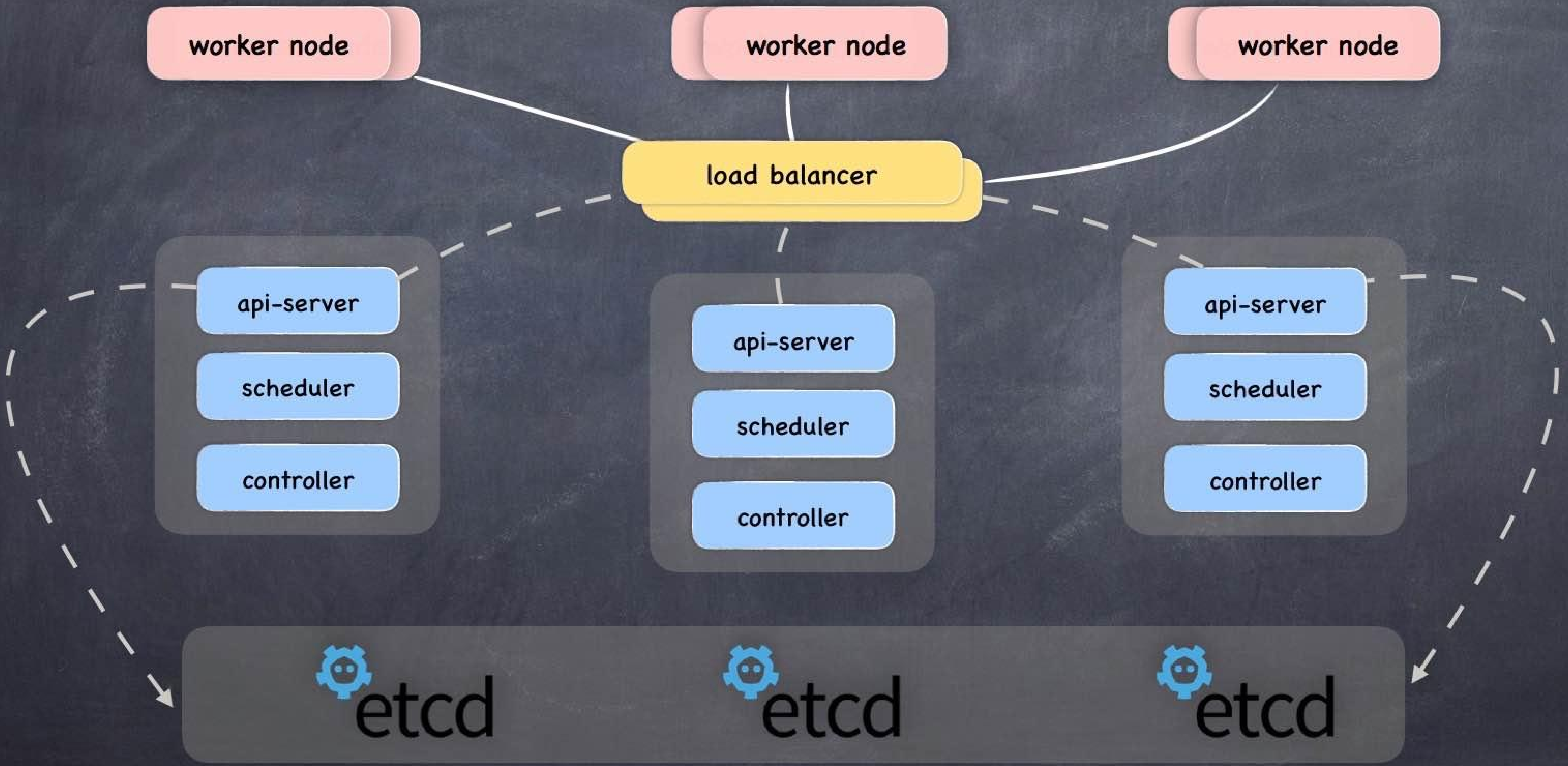
- 基于容器的集群编排引擎

- 扩展集群
- 滚动升级回退
- 弹性伸缩服务
- 自动治愈
- 服务发现
- 资源配额
- 灵活扩展API

kubernetes 架构



kubernetes ha



kubernetes 架构

- maser
  - api server
    - 总操作入口
  - controller
    - 控制中心
  - scheduler
    - pod调度器

- node
  - kubelet
    - 管理容器的生命周期
    - 监控
    - 上报节点状态
  - kube-proxy
    - 管理service

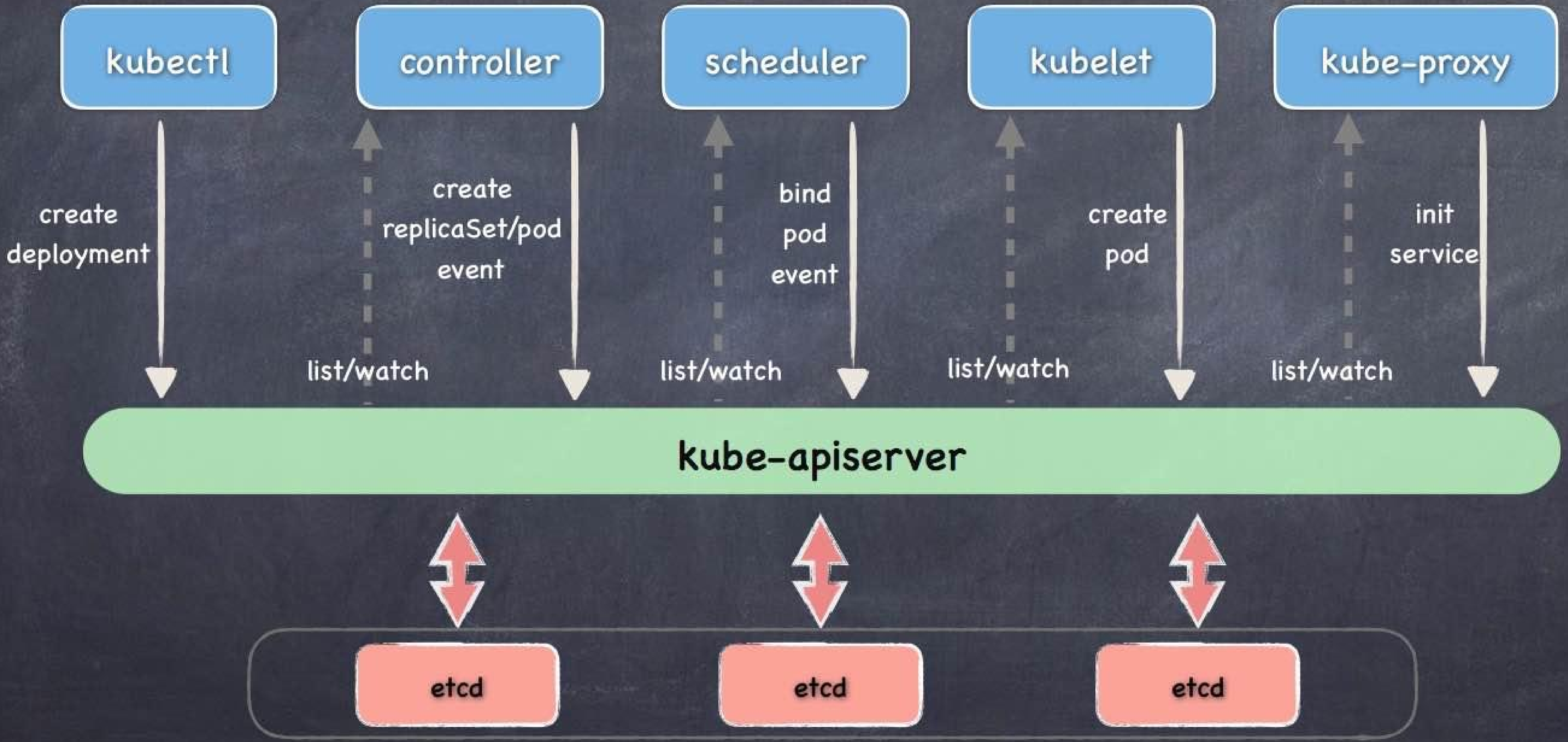
kubernetes notion

- Pod 最小单位
- Deployment
- Service
- RepliaSet
- StatefulSet
- DaemonSet

- Crontab
- Job
- ConfigMap

- Label
  - node
    - disktype=ssd
    - gpu=true
  - pod
    - app
    - version
  - ...

cretae deployment process

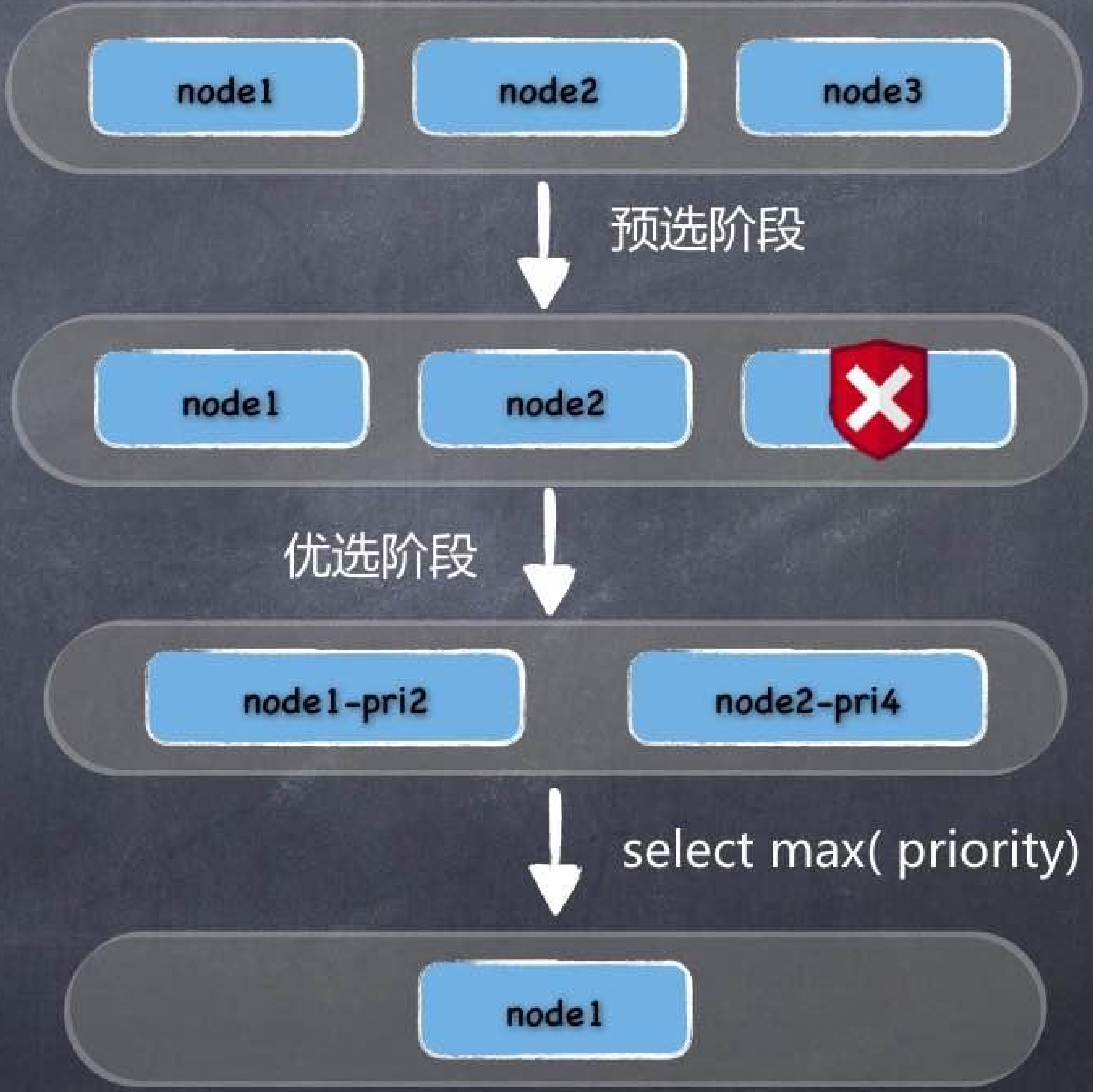


scheduler

- predicates 预选过程
    - 过滤掉不满足条件的节点
    - PodFitsResources
    - PodFitsHostPorts
    - PodSelectorMatches
    - CheckNodeDiskPressure
    - CheckNodeMemoryPressure

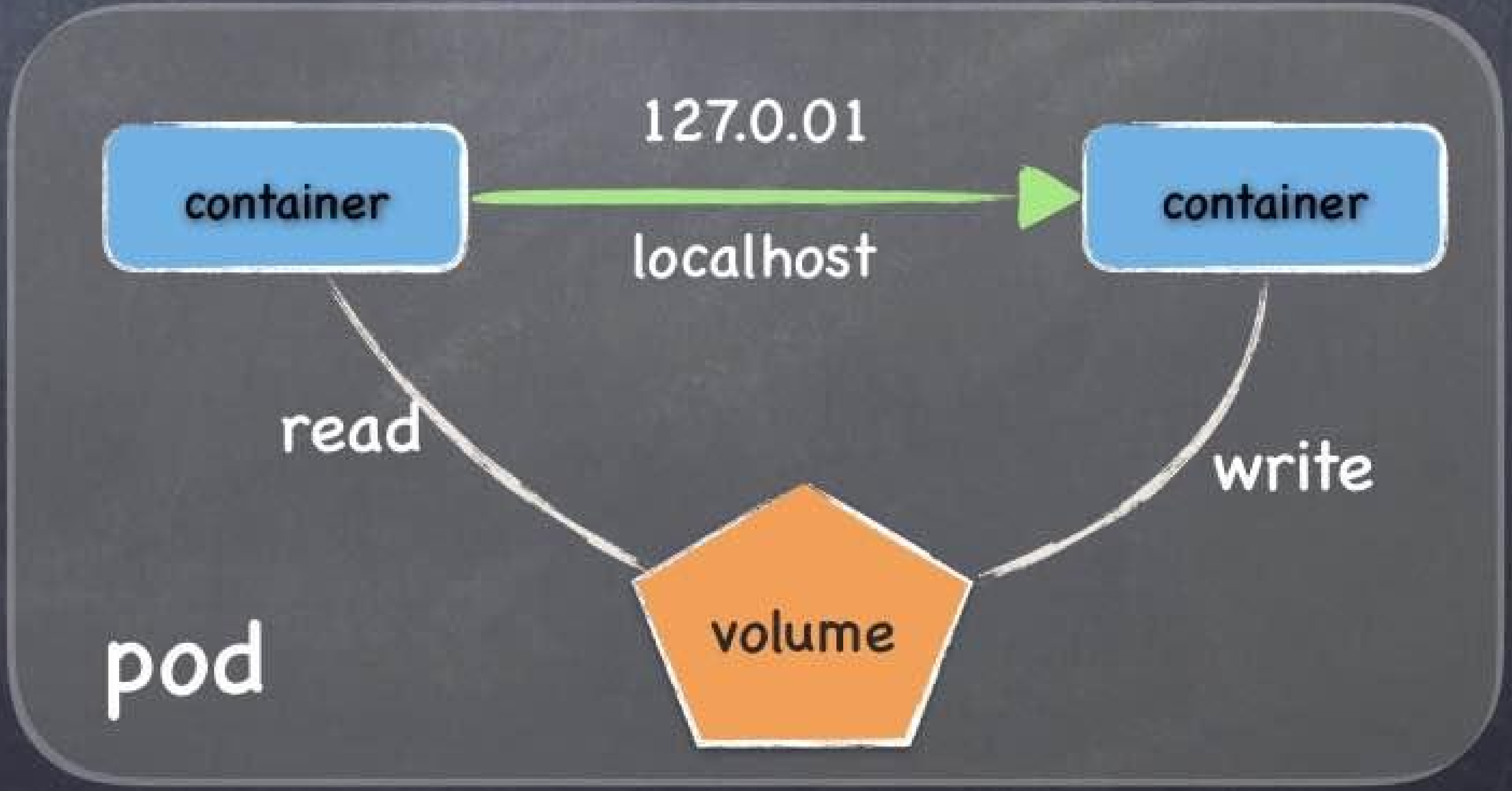
- priorities 优选过程
    - 对节点按照优先级排序
    - LeastRequestedPriority
    - SelectorSpreadPriority
    - ImageLocalityPriority
    - NodeAffinityPriority

- algorithmprovider
    - 选择优先级最高的节点



pod

- 一个pod可以有多个容器
- pod之间容器共享网络namespace (127.0.0.1)
- pod之间容器通过Volume来共享目录 (emptyDir and hostPath)



Service Detail

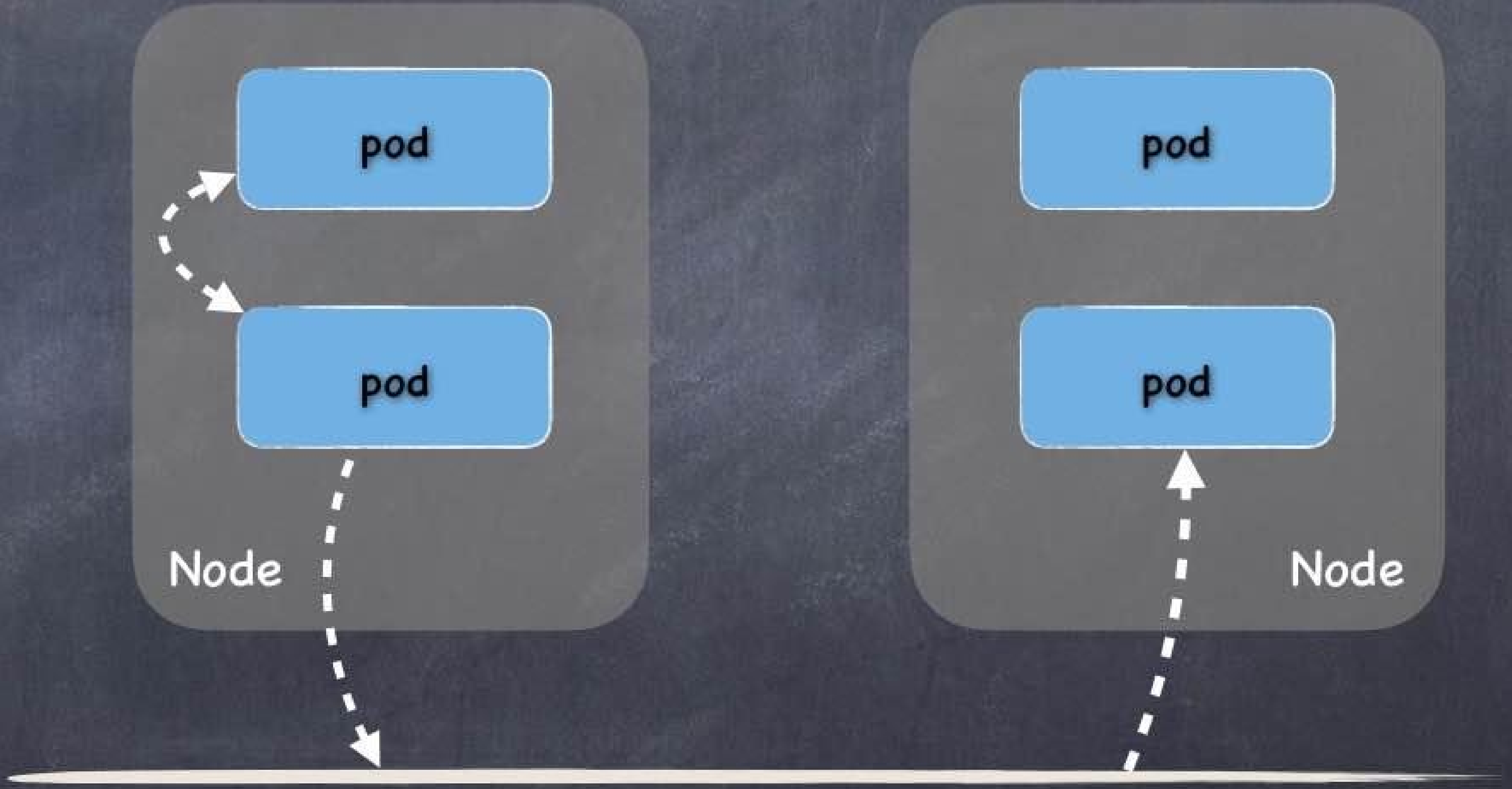
- iptables做转发
    - 匹配延迟
        - 线性匹配
    - 更新延迟
        - 不能增量

- type
    - clusterIP
    - nodePort
    - HeadLess
        - clusterIP: None
    - lb
    - …

- ipvs做转发
    - 算法更灵活
        - 最小负载
        - 最少连接
        - session
    - hash 匹配
    - 可控的更新延迟

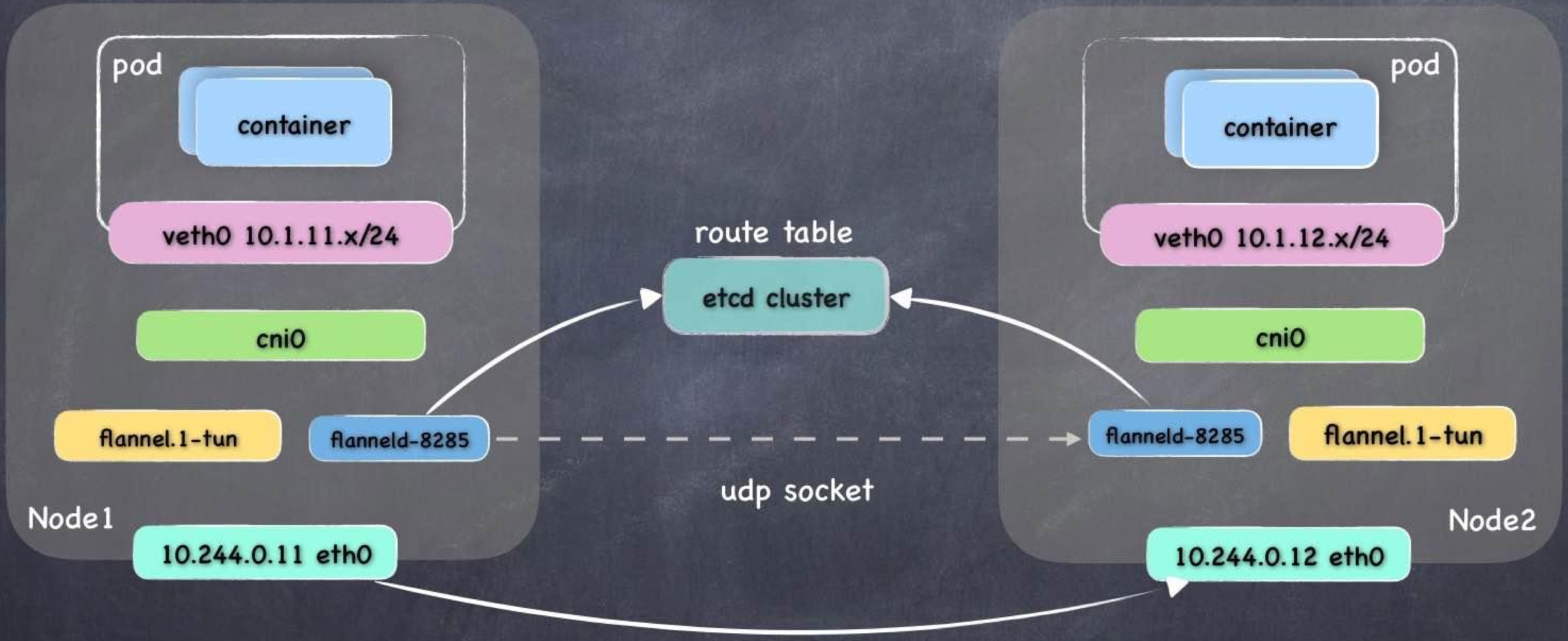
kubernetes network

- Pod
  - 宿主机到pod可以通
  - 宿主机的pod之间可以互通
    - docker/cni0 网桥
  - 不同node的pod也可以互通
    - cni 接口



kubernetes cni

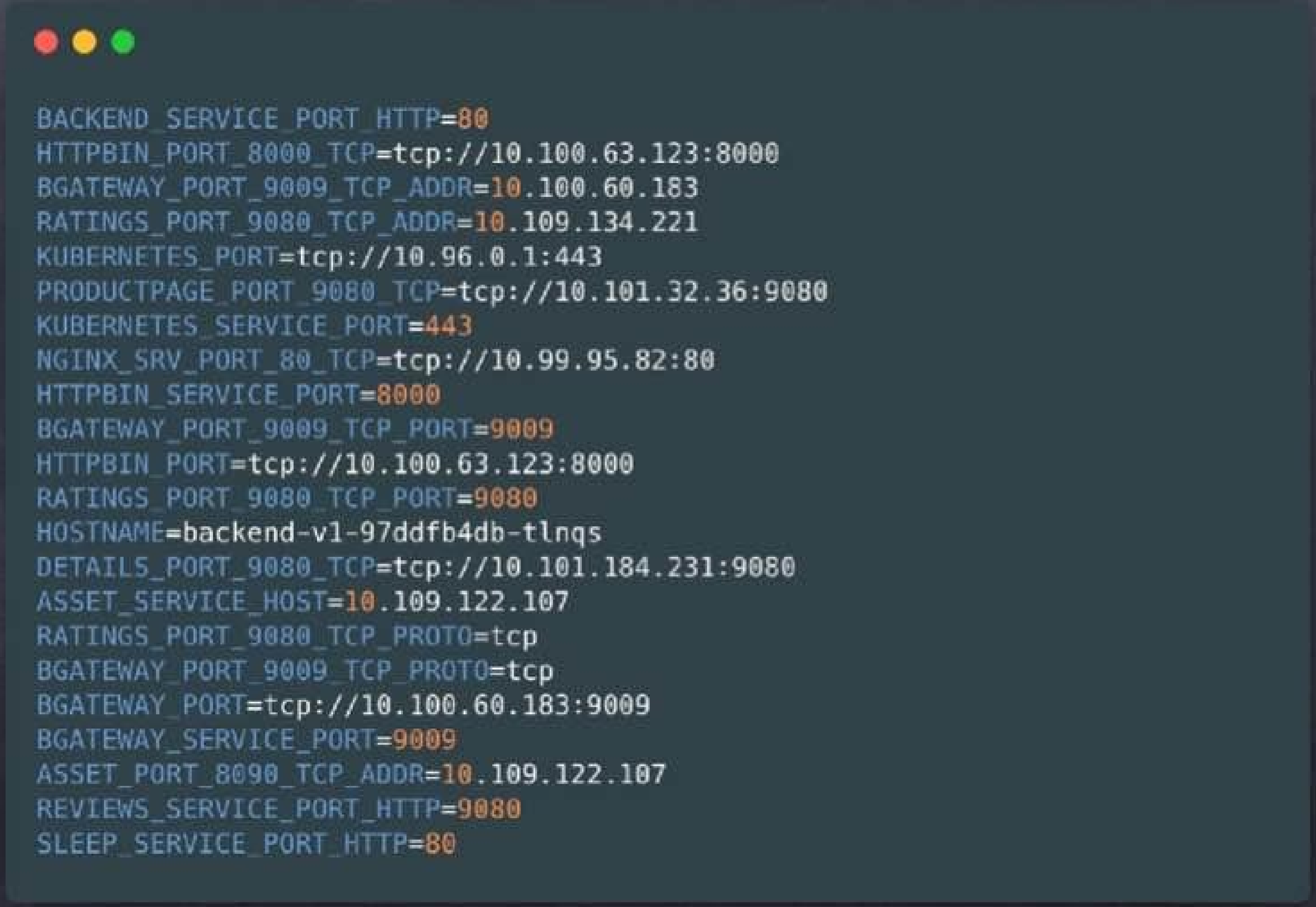
flannel



Packet	mac	outer-ip	udp	inner-ip	payload

服务发现

- 环境变量 env
  - Get ClusterIP, Port
- 使用service name
  - 经过coredns解析拿到clusterIP



```
BACKEND_SERVICE_PORT_HTTP=80
HTTPBIN_PORT_8000_TCP=tcp://10.100.63.123:8000
BGATEWAY_PORT_9009_TCP_ADDR=10.100.60.183
RATINGS_PORT_9080_TCP_ADDR=10.109.134.221
KUBERNETES_PORT=tcp://10.96.0.1:443
PRODUCTPAGE_PORT_9080_TCP=tcp://10.101.32.36:9080
KUBERNETES_SERVICE_PORT=443
NGINX_SRV_PORT_80_TCP=tcp://10.99.95.82:80
HTTPBIN_SERVICE_PORT=8000
BGATEWAY_PORT_9009_TCP_PORT=9009
HTTPBIN_PORT=tcp://10.100.63.123:8000
RATINGS_PORT_9080_TCP_PORT=9080
HOSTNAME=backend-v1-97ddfb4db-tlnqs
DETAILS_PORT_9080_TCP=tcp://10.101.184.231:9080
ASSET_SERVICE_HOST=10.109.122.107
RATINGS_PORT_9080_TCP_PROTO=tcp
BGATEWAY_PORT_9009_TCP_PROTO=tcp
BGATEWAY_PORT=tcp://10.100.60.183:9009
BGATEWAY_SERVICE_PORT=9009
ASSET_PORT_8090_TCP_ADDR=10.109.122.107
REVIEWS_SERVICE_PORT_HTTP=9080
SLEEP_SERVICE_PORT_HTTP=80
```

对于业务来说, 使用 Service Name 就可以了

kubernetes 外部访问

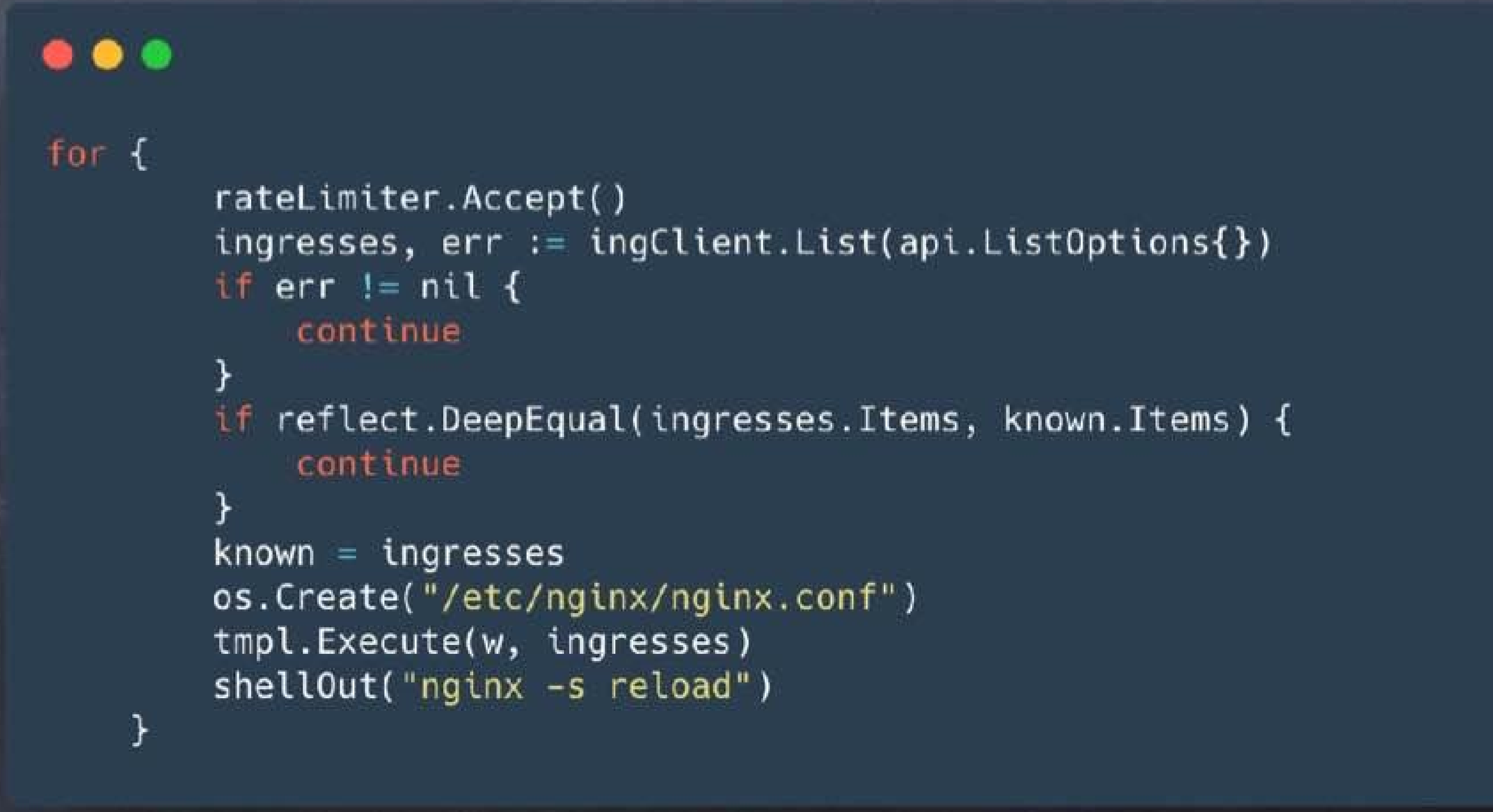
- hostNetwork = true
- hostPort
- Ingress (nginx, haproxy, traefix, envoy)
- NodePort (iptables nat)
- 公有云Load Balancer (aws, azure, gce ...)

kubernetes 外部访问

ingress
NodePort 30000,32767
pod N
inside kubernets cluster
Node
Service A
pod A
pod B
Node
Service B
pod A
pod B

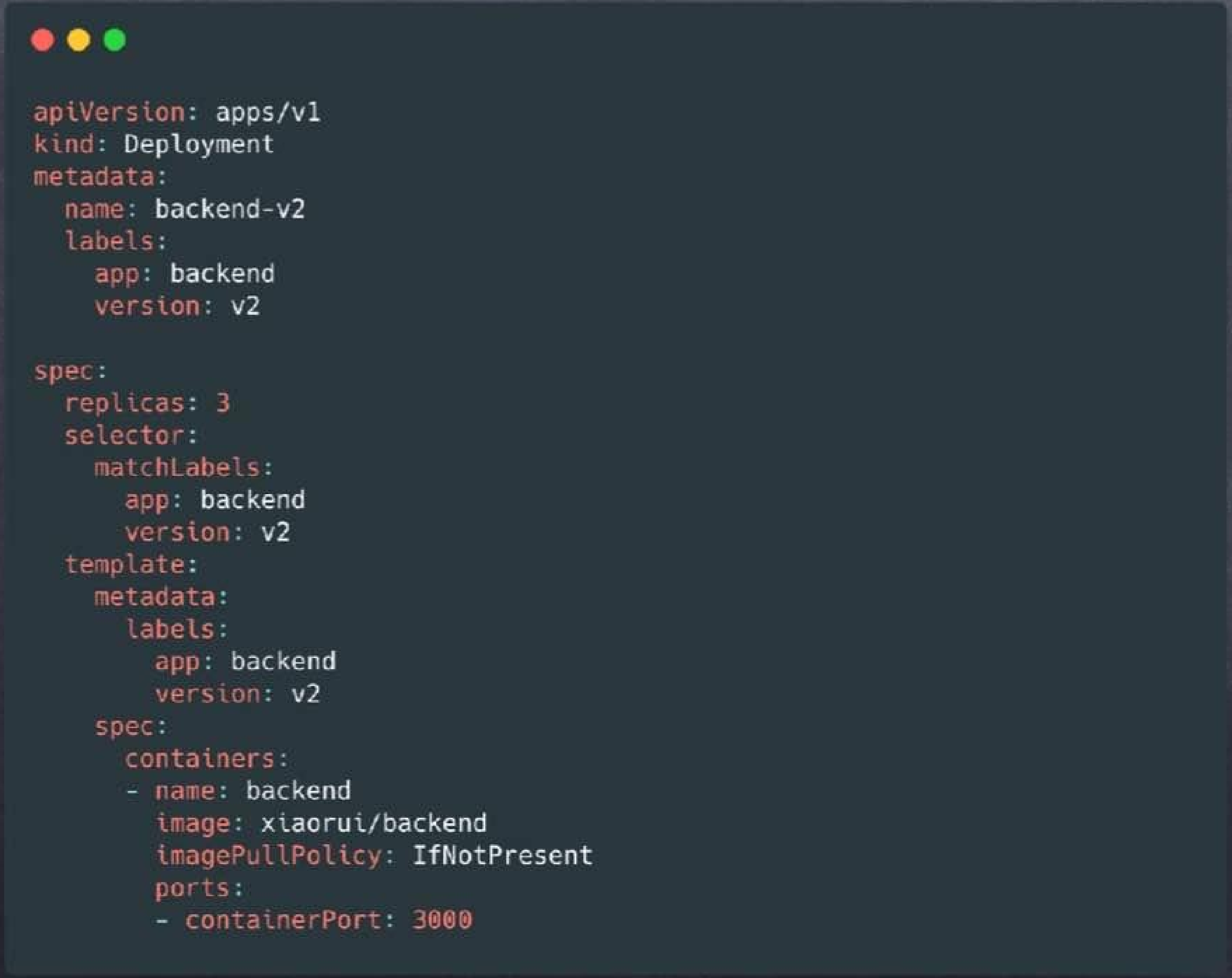
ingress design

- skip kube-proxy
  - direct upstream endpoint
- hostPort
  - bind node port
- daemonSet
  - one pod each node



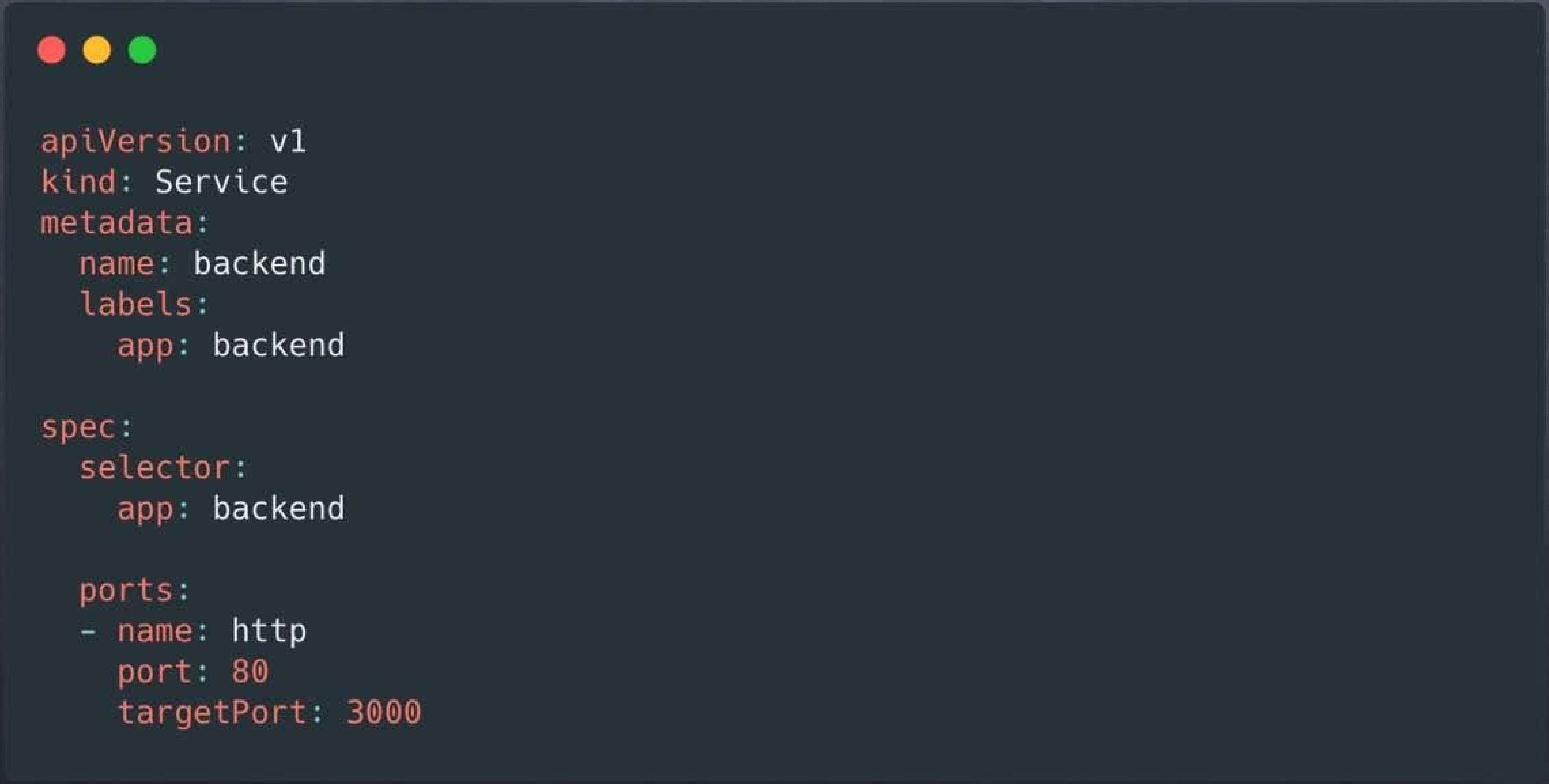
```
for {
      rateLimiter.Accept()
      ingresses, err := ingClient.List(api.ListOptions{})
      if err != nil {
         continue
      }
      if reflect.DeepEqual(ingresses.Items, known.Items) {
         continue
      }
      known = ingresses
      os.Create("/etc/nginx/nginx.conf")
      tmpl.Execute(w, ingresses)
      shellOut("nginx -s reload")
   }
```

Deployment



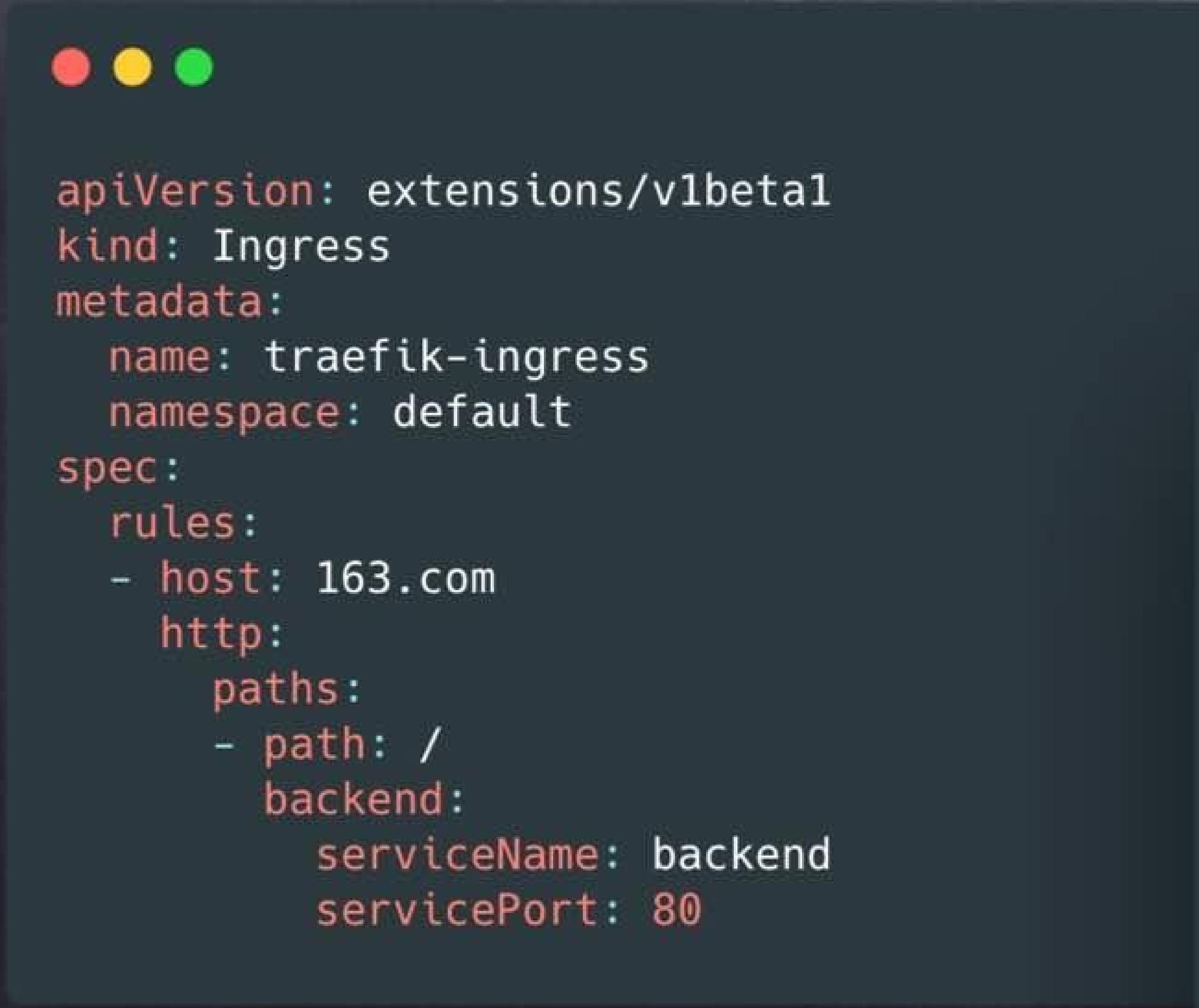
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend-v2
  labels:
    app: backend
    version: v2

spec:
  replicas: 3
  selector:
    matchLabels:
      app: backend
      version: v2
  template:
    metadata:
      labels:
        app: backend
        version: v2
    spec:
      containers:
      - name: backend
        image: xiaorui/backend
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 3000
```

Service

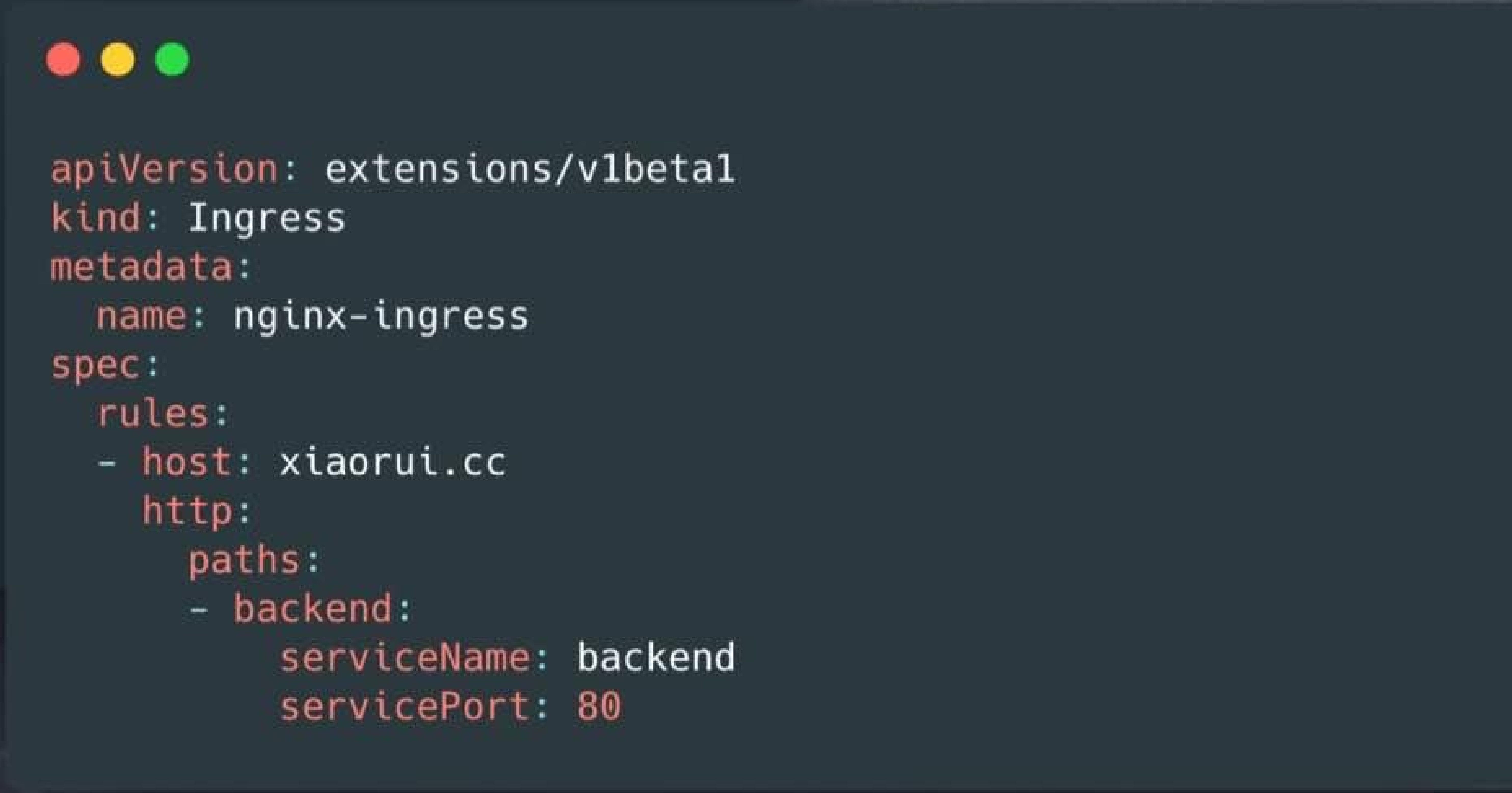


```yaml
apiVersion: v1
kind: Service
metadata:
  name: backend
  labels:
    app: backend

spec:
  selector:
    app: backend

  ports:
  - name: http
    port: 80
    targetPort: 3000
```

Ingress

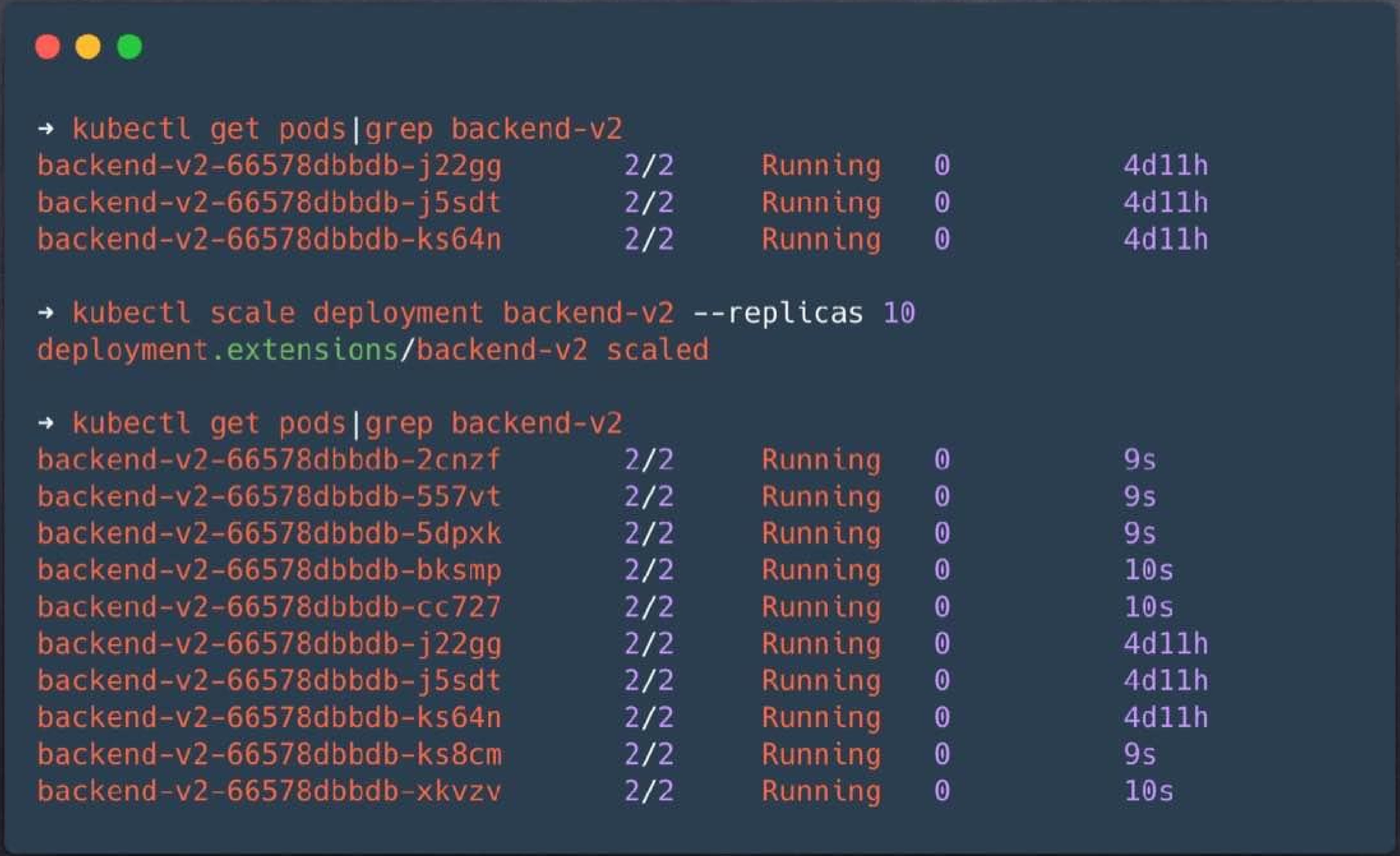


```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: traefik-ingress
  namespace: default
spec:
  rules:
  - host: 163.com
    http:
      paths:
      - path: /
        backend:
          serviceName: backend
          servicePort: 80
```



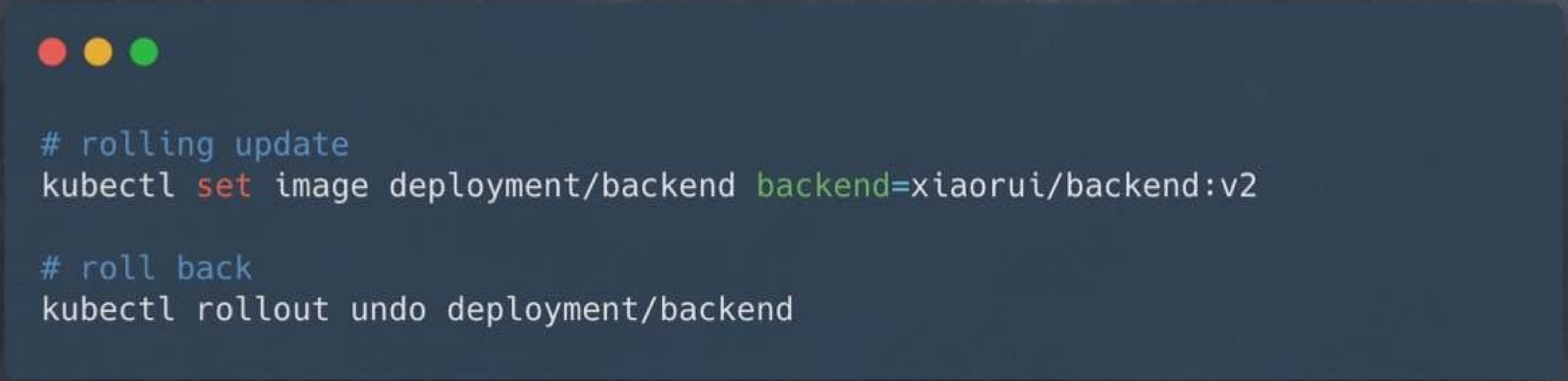
```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx-ingress
spec:
  rules:
  - host: xiaorui.cc
    http:
      paths:
      - backend:
          serviceName: backend
          servicePort: 80
```

快速扩容



```
➜ kubectl get pods|grep backend-v2
backend-v2-66578dbbdb-j22gg       2/2     Running   0          4d11h
backend-v2-66578dbbdb-j5sdt       2/2     Running   0          4d11h
backend-v2-66578dbbdb-ks64n       2/2     Running   0          4d11h

➜ kubectl scale deployment backend-v2 --replicas 10
deployment.extensions/backend-v2 scaled

➜ kubectl get pods|grep backend-v2
backend-v2-66578dbbdb-2cnzf       2/2     Running   0          9s
backend-v2-66578dbbdb-557vt       2/2     Running   0          9s
backend-v2-66578dbbdb-5dpxk       2/2     Running   0          9s
backend-v2-66578dbbdb-bksmp       2/2     Running   0          10s
backend-v2-66578dbbdb-cc727       2/2     Running   0          10s
backend-v2-66578dbbdb-j22gg       2/2     Running   0          4d11h
backend-v2-66578dbbdb-j5sdt       2/2     Running   0          4d11h
backend-v2-66578dbbdb-ks64n       2/2     Running   0          4d11h
backend-v2-66578dbbdb-ks8cm       2/2     Running   0          9s
backend-v2-66578dbbdb-xkvzv       2/2     Running   0          10s
```

升级回滚



```
# rolling update
kubectl set image deployment/backend backend=xiaorui/backend:v2

# roll back
kubectl rollout undo deployment/backend
```

- maxUnavailable:
  - 更新过程中不可用的pod数量
  - default: 25%

- maxSurge:
  - 更新中pod总数的最大值
  - default: 25%

也可使用servcie selector version规避

升级回滚

Deployment
Rs (old)
app-v1
Deployment
Rs (old)
Rs (new)
app-v1
app-v2
Deployment
Rs (old)
Rs (new)
app-v1
app-v2
Deployment
Rs (new)
app-v2

“ Q&A ”

- xiaorui.cc