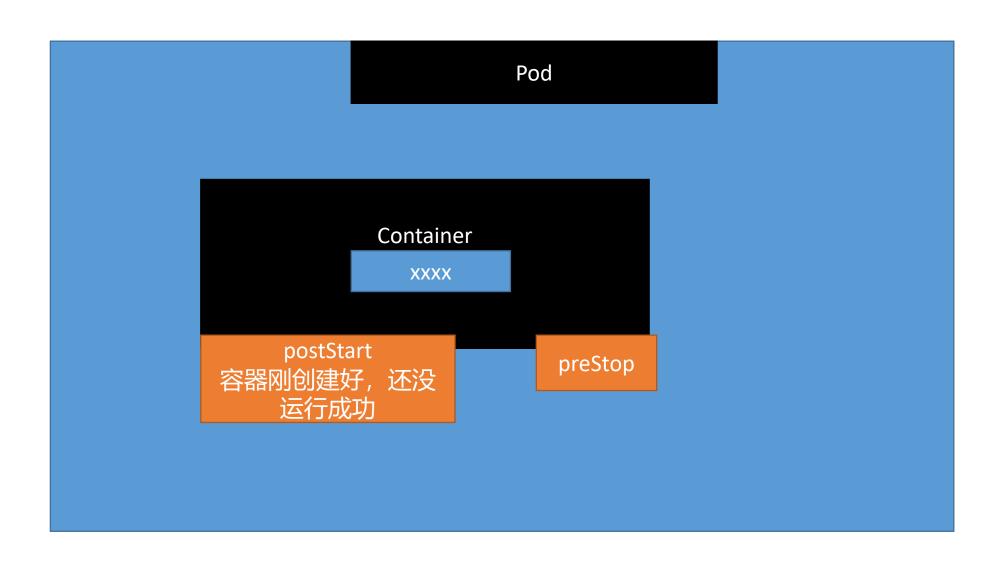
K8S图例

容器

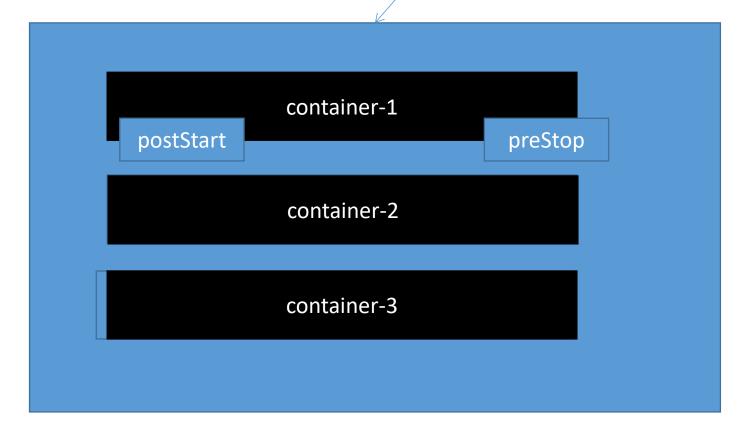


kubelet: 控制容器生命周期

Pod

kubectl delete pod xxx

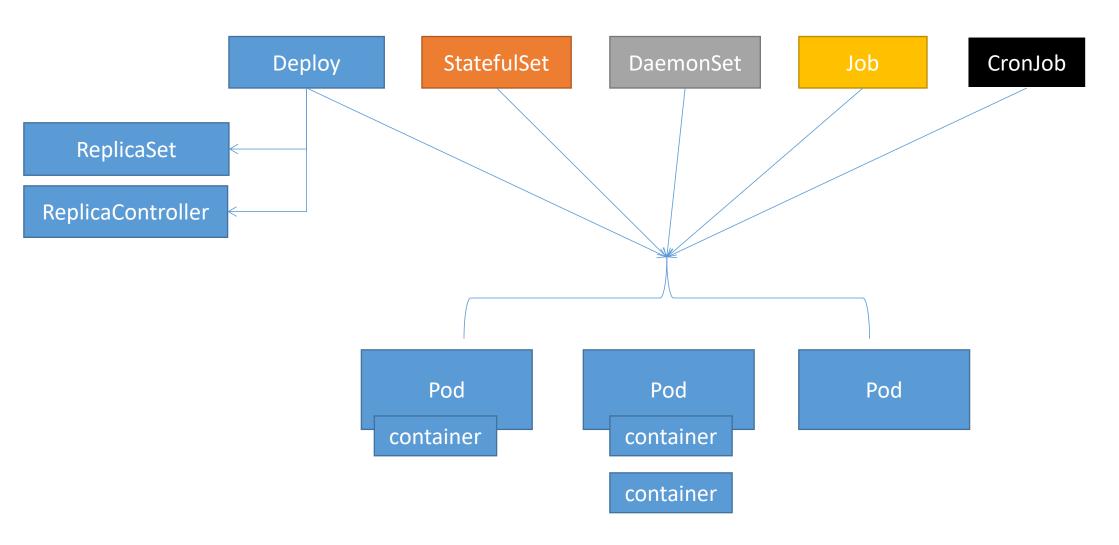
containers



Pod



工作负载



控制器

Master节点

controller-manager

控制器管理器

api-server

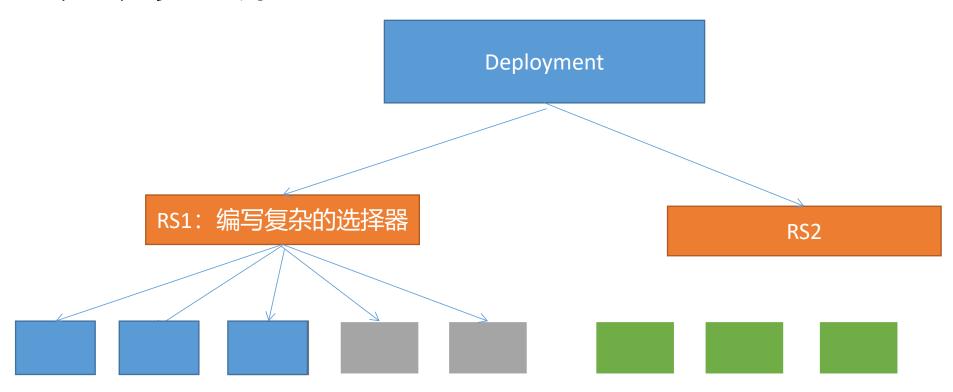
etcd

kubectl apply -f deploy.yaml (kind: Deployment) Deployment 控制器 (Controller) Job控制器 (Controller)

kubectl apply -f job.yaml (kind: Job)

kubelet

滚动更新



金丝雀-Service版

Deployment v3

app=nginx v=3

app=nginx v=3

Service app=nginx

Deployment v1 app=nginx, v=1

app=nginx v=1

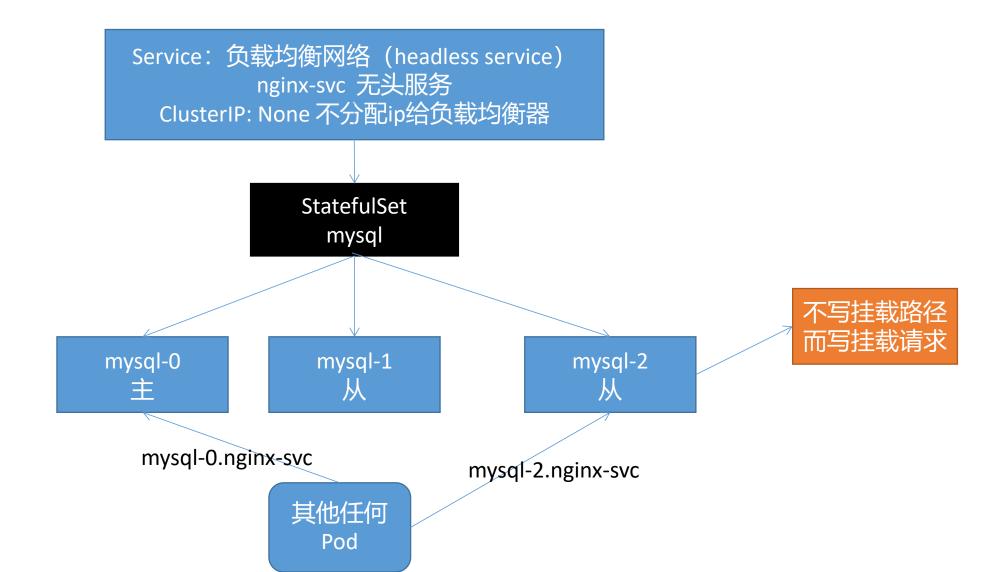
app=nginx v=1 app=nginx v=1 Deployment v2 给他扩缩容能调整这个版本接 受的流量多少

app=nginx v=2 app=nginx v=2

当V2 Ok以后,Deployment V1就被删除

StatefulSet

全地址
pod-specific-string.serviceName.default.svc.cluster.local
pod名.service名.namespace名.后面一串默认的



CronJob

startingDeadlineSeconds:启动的超时时间 600s。设置超大concurrencyPolicy:并发策略。设置为Allow "Allow"(允许, defail "Forbid"(禁止):forbids;前个任务没执行完,要并发下一个的话,个会被跳过

"Replace"(替换): 新任务,替换当前运行的任务 能保证最起码有一个任务在运行。



Service整个端口问题^{curl 10.170.11.11:6379 访问不到}

port: 80
targetPort: 8080

Service: 10.170.11.88 cluster-service-02

port: 80

port: 99

Service: 10.170.11.11

cluster-service-test

port: 80

port: 99

- name: abc port: 80

targetPort: 8080

name: redis

port: 99

targetPort: 6379

targetPort: 8080

container-03 不能占用8080

Pod: 也有ip。只要有ip就认为是一个新主机

app: canary-tomcat

containerPort: 8080

tomcat-container

containerPort: 6379

redis-container

targetPort: 8080

Pod

app: canary-tomcat

targetPort: 8080

Pod

app: canary-tomcat

curl 10.170.11.11:80

port: 80

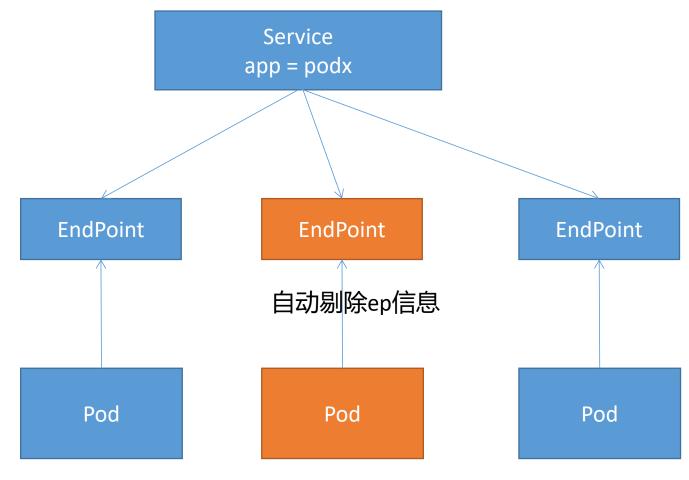
targetPort: 8080

以上所有端口和Node的端口没有任何冲突

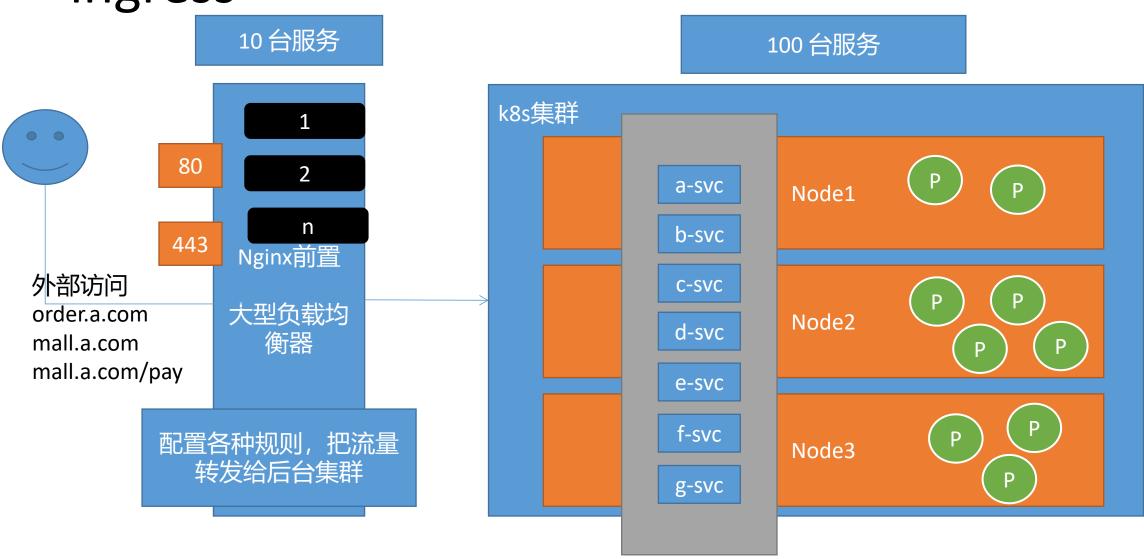
网络层次 -- 默认全是通的

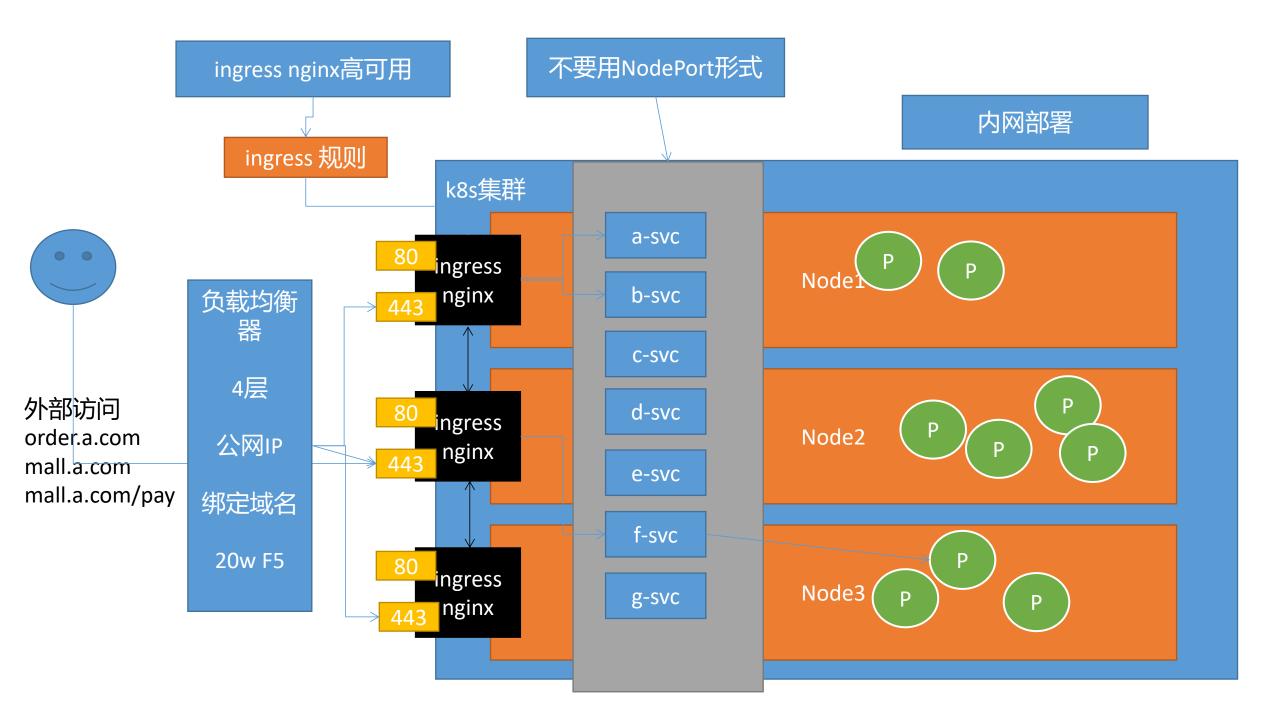


Service原理

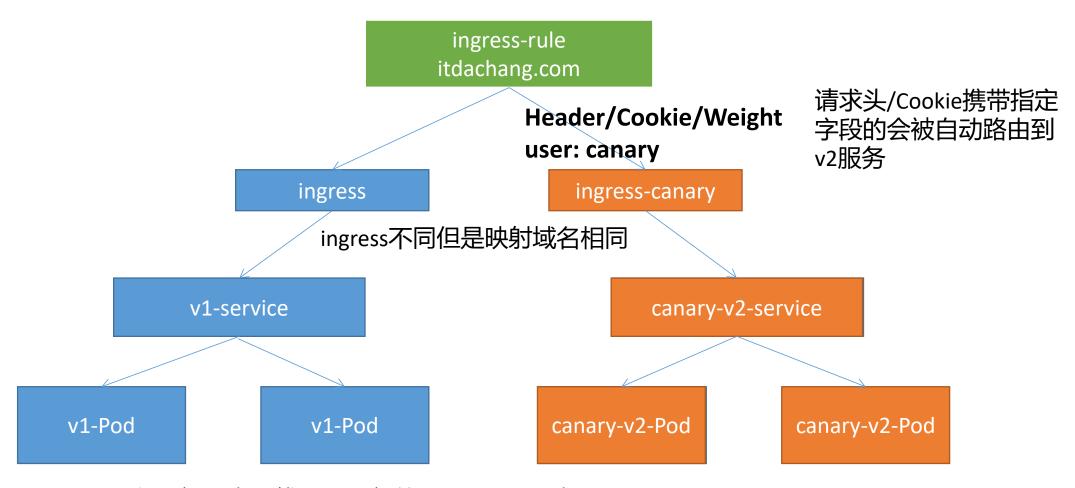


ingress





金丝雀-Ingress版



以后新版本上线,配置新的ingress-canary规则即可。 canary验证通过以后,移除旧的ingress和service。 取消当前ingress-canary的annotation,变为普通的ingress