

阿里云 × CLOUD NATIVE  
COMPUTING FOUNDATION

# 云原生技术公开课

第 14 讲

## Kubernetes Service

溪恒 阿里巴巴技术专家



关注“阿里巴巴云原生”公众号  
获取第一手技术资料



# 上节回顾

## **Kubernetes Pod网络的基本要素**

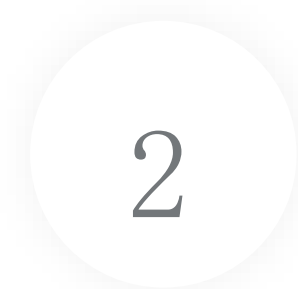
- **Pod**有独立的网络空间和唯一地址
- **Pod**与**Pod**, **Node**, 外界网络互联互通

## **NetworkPolicy**

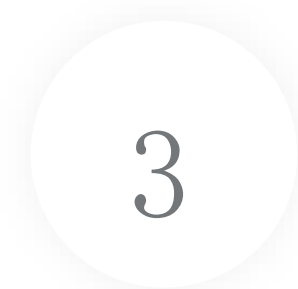
- 控制**Pod**到**Pod**, **Node**, 外接网络的访问限制



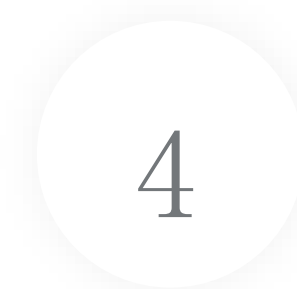
需求来源



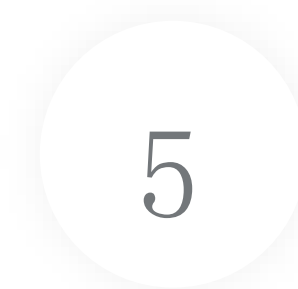
用例解读



操作演示



架构设计



课后思考实践

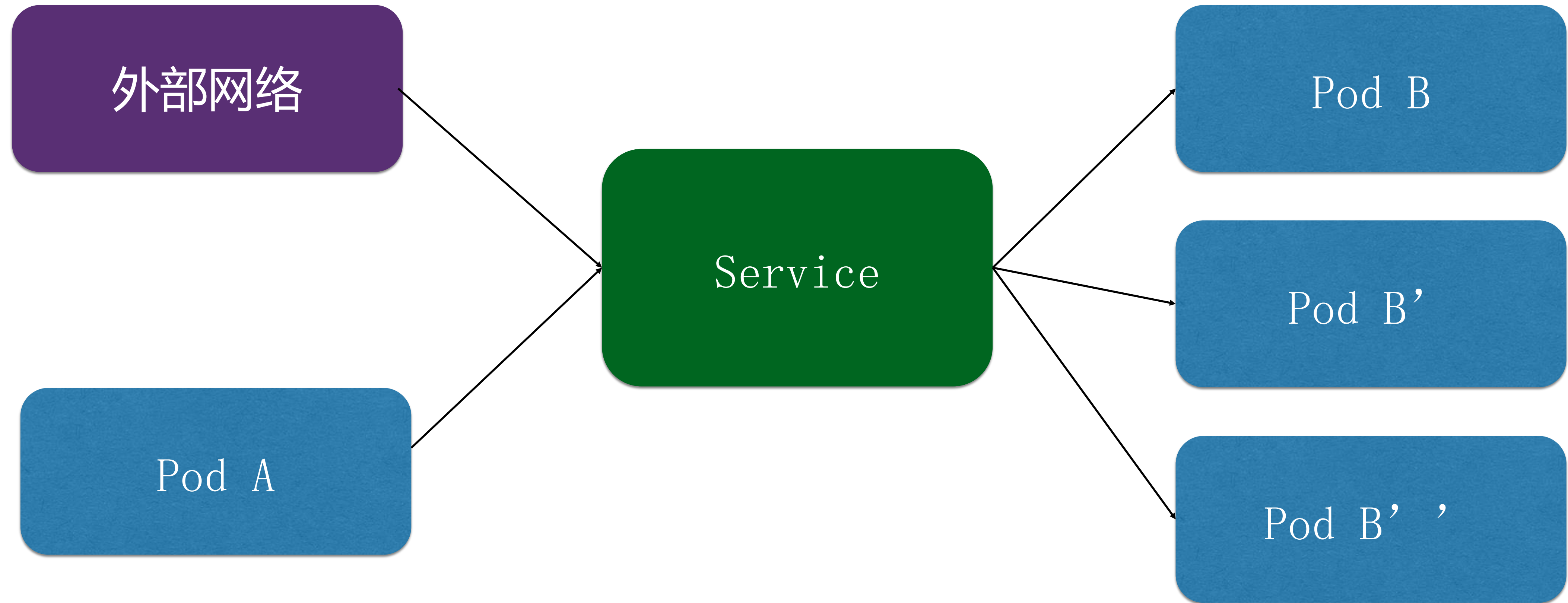
# 为什么需要服务发现

## **Kubernetes应用应如何相互调用？**

- **Pod**生命周期短暂，**IP**地址随时变化
- **Deployment**等的**Pod**组需要统一访问入口和做负载均衡
- 应用间在不同环境部署时保持同样的部署拓扑和访问方式

## **应用服务如何暴露到外部访问和负载均衡？**

# Service: Kubernetes中的服务发现与负载均衡



1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

架构设计

.....

5

课后思考实践



# Service 语法

## 新知识点:

**ports:** 定义服务发现的协议和端口

## 历史知识点回顾:

**labels:** 标签

**selector:** 选择器

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  selector:
    app: MyApp
  ports:
  - protocol: TCP
    port: 80
    targetPort: 9376
```

**Service**元信息

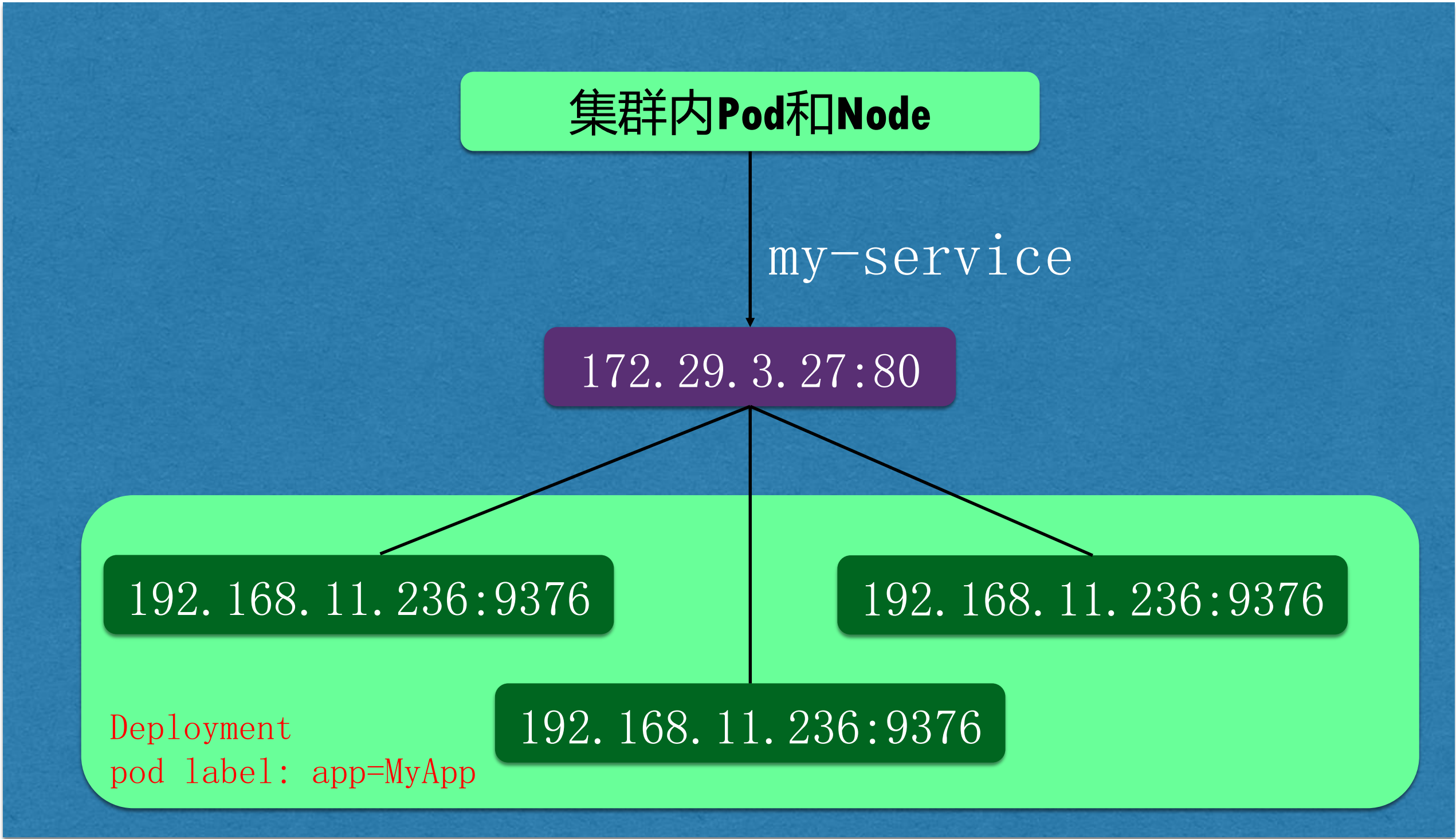
**Pod**选择器

协议和端口

# 创建和查看 Service

```
$ kubectl apply -f service.yaml
service/my-service created

$ kubectl describe service my-service
Name: my-service
Namespace: default
Labels: app=my-service
Selector: app=MyApp
Type: ClusterIP
IP: 172.29.3.27
Port: 80/TCP
TargetPort: 9376/TCP
Endpoints: 192.168.115.236:9376, 192.168.115.237:9376, 192.168.115.243:9376
Session Affinity: None
```



selector 匹配到的后端Pod地址



# 集群内访问Service

- 直接Service的虚拟IP

**kubectl get svc** 可以查到

- 直接访问服务名，依靠**DNS**解析

同一个**Namespace**直接通过**servicename**访问

不同**Namespace**加上**namespace**名访问：

**{servicename}.{namespace}**

- 通过环境变量访问

**MY\_SERVICE\_PORT\_80\_TCP=tcp://172.29.3.27:80**

**MY\_SERVICE\_SERVICE\_HOST=172.29.3.27**

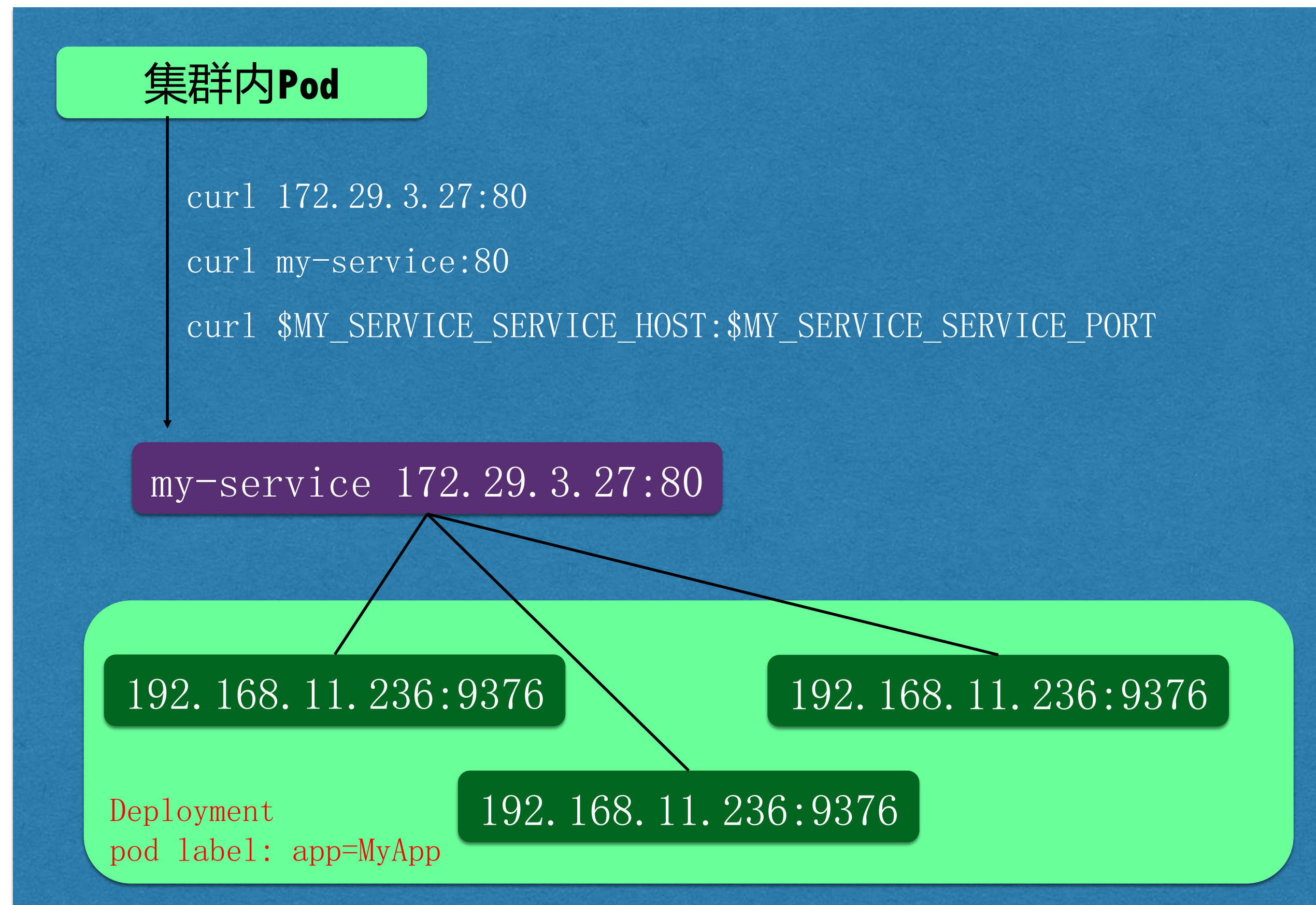
**MY\_SERVICE\_PORT=tcp://172.29.3.27:80**

**MY\_SERVICE\_SERVICE\_PORT=80**

**MY\_SERVICE\_PORT\_80\_TCP\_ADDR=172.29.3.27**

**MY\_SERVICE\_PORT\_80\_TCP\_PORT=80**

**MY\_SERVICE\_PORT\_80\_TCP\_PROTO=tcp**



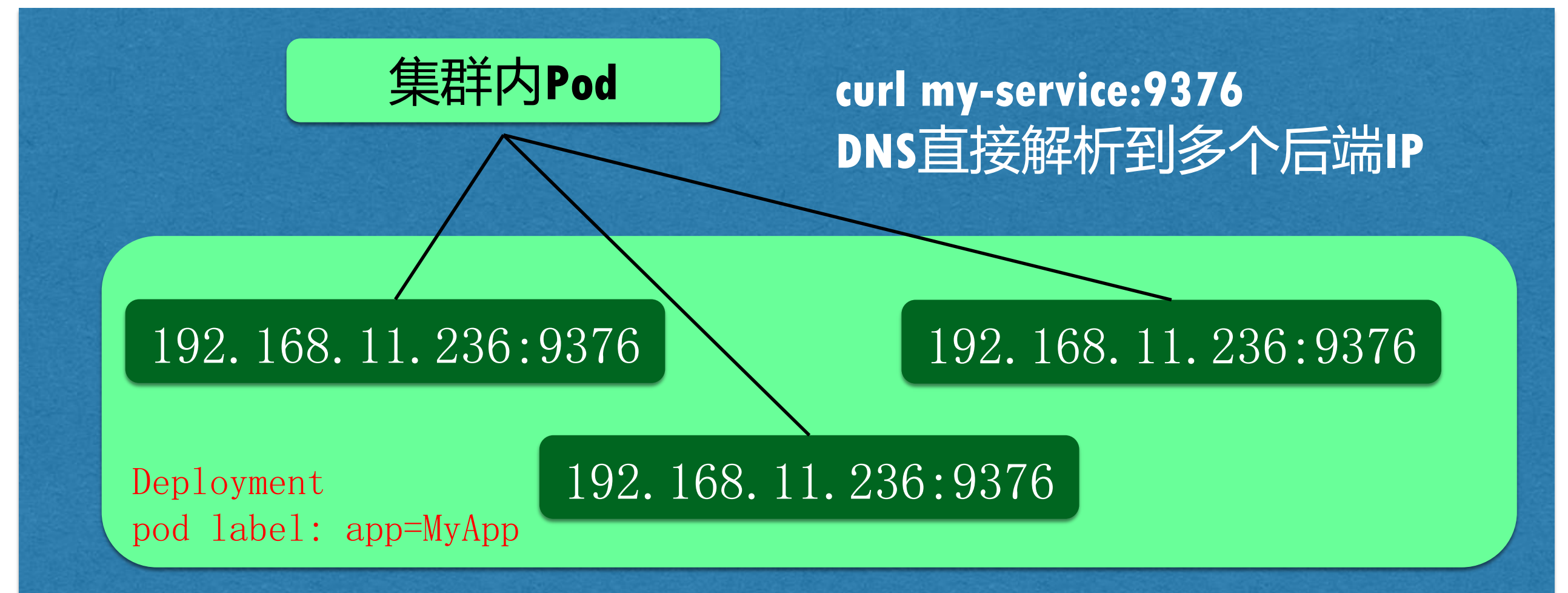
# Headless Service

## Service 指定 clusterIP: None

- Pod 通过 service\_name 方式时直接解析到所有后端 Pod IP
- 客户端应用自主选择需要访问的 Pod

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  selector:
    app: MyApp
  clusterIP: None
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

增加 **clusterIP: None**  
让 **Service** 不再通过虚拟 IP 来负载均衡



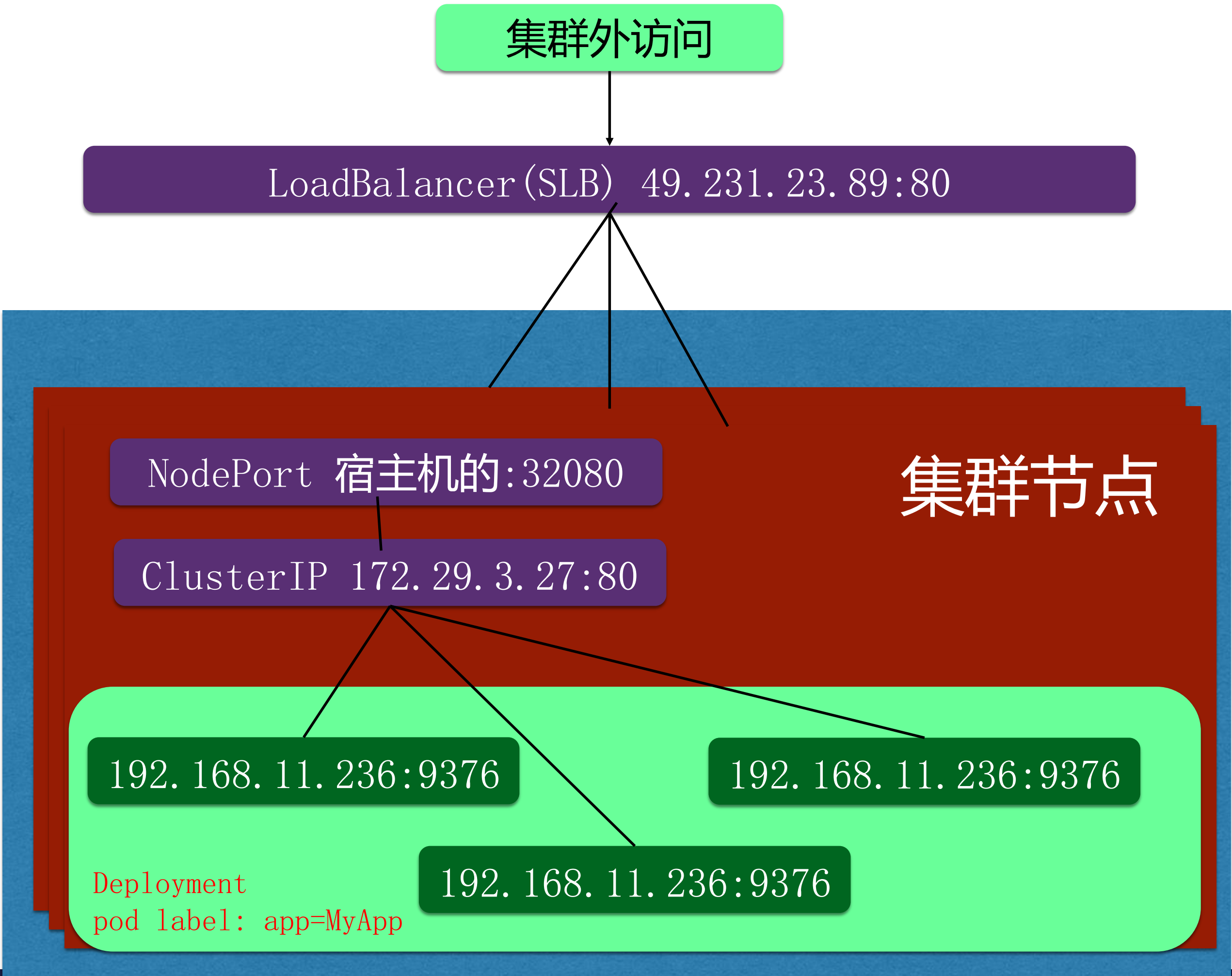
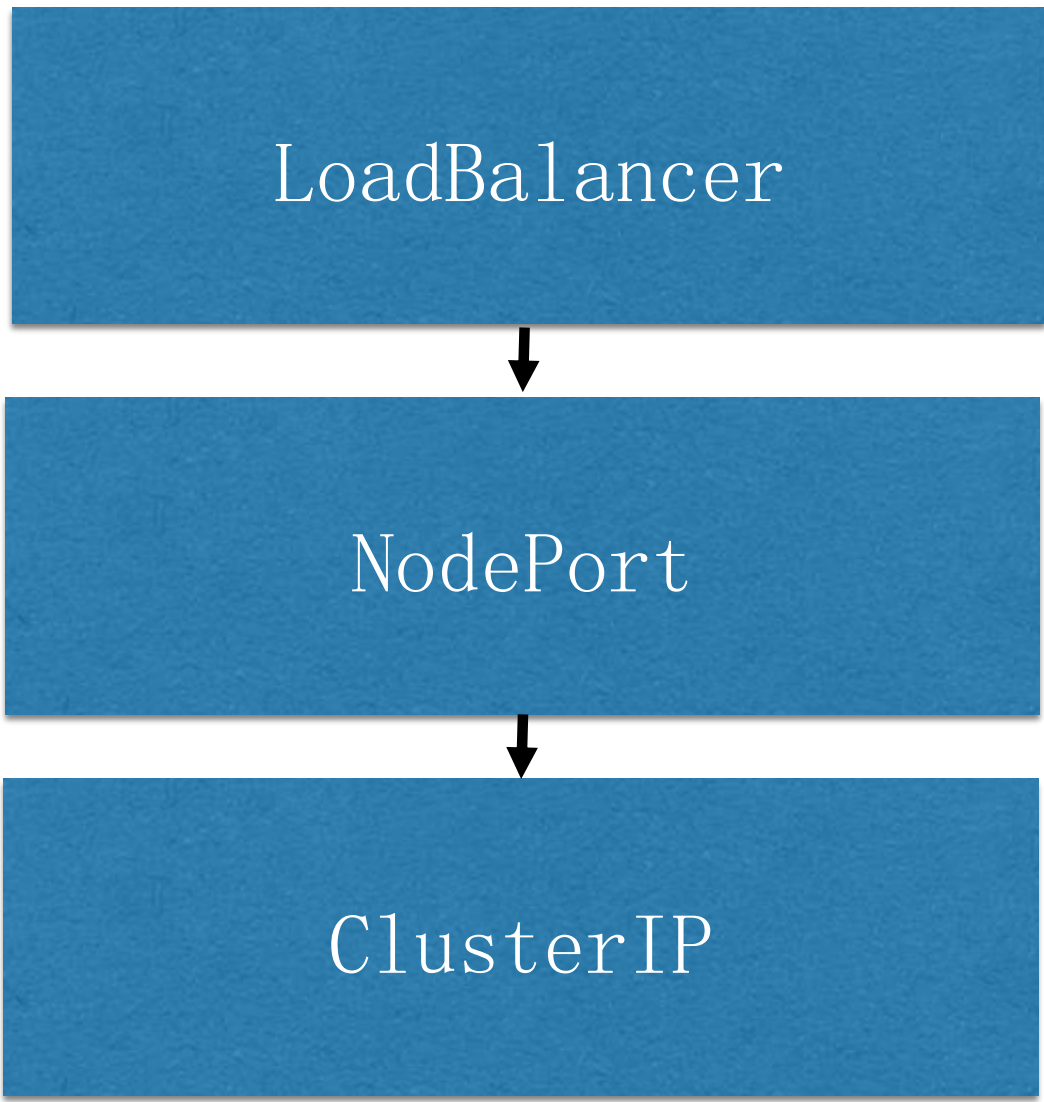


# 向集群外暴露Service

## Service类型

- **ClusterIP**
- **ExternalName**
- **NodePort**
- **LoadBalancer**

} 向外暴露服务



1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

架构设计

.....

5

课后思考实践

# 操作演示

- 创建一个 **Deployment** 来产生一组服务 **Pod**
- 创建一个 **Service** 负载均衡这一组 **Pod**
- 在集群中创建一个 **Pod** 通过不同方式访问 **Service**
- 修改服务类型，通过 **NodePort** 和 **Loadbalancer** 类型来暴露服务到外部



1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

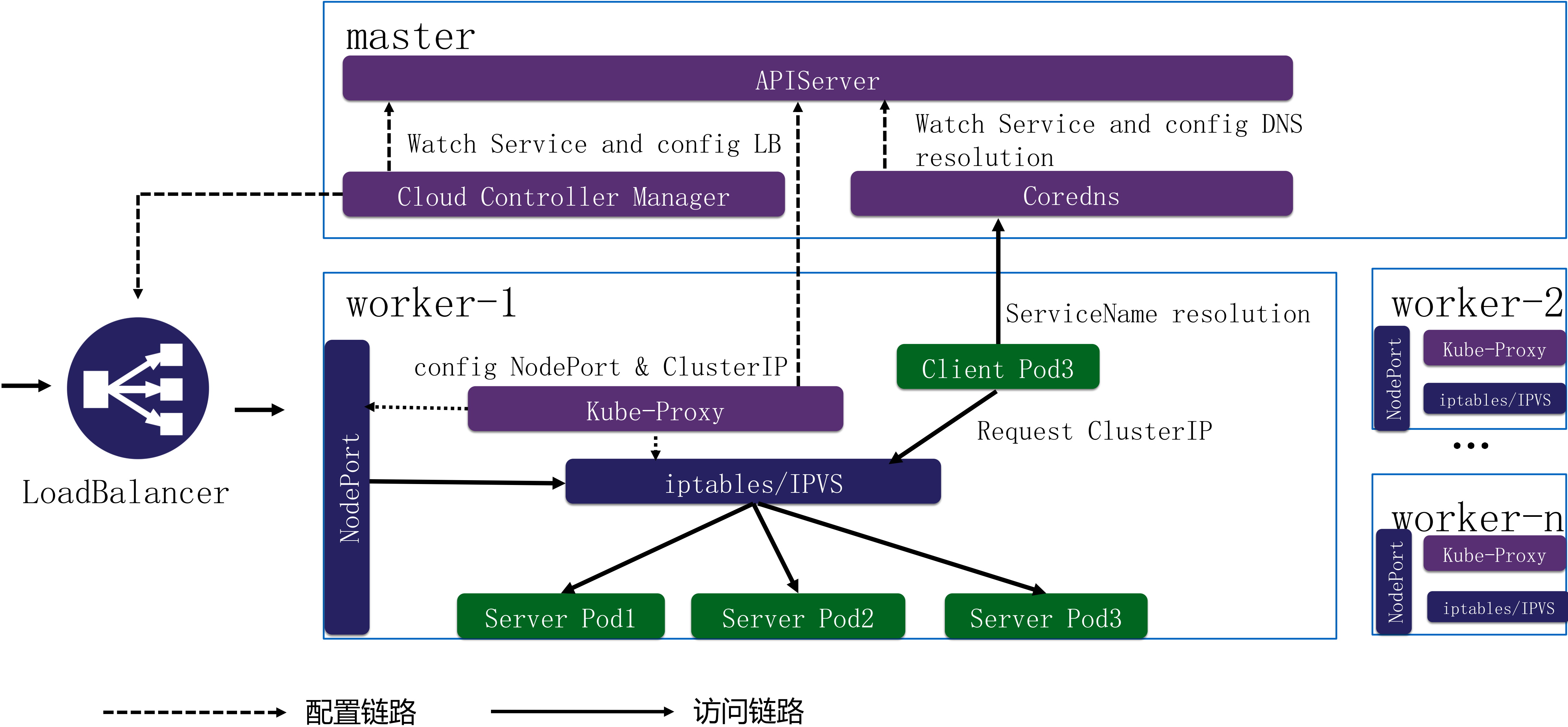
架构设计

.....

5

课后思考实践

# Kubernetes 服务发现架构



# 后续进阶部分预告

- 深入讲解 **Service** 的实现原理
- **Service**网络出现问题时的 **Debug** 技巧

1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

架构设计

.....

5

课后思考实践



关注“阿里巴巴云原生”公众号  
获取第一手技术资料