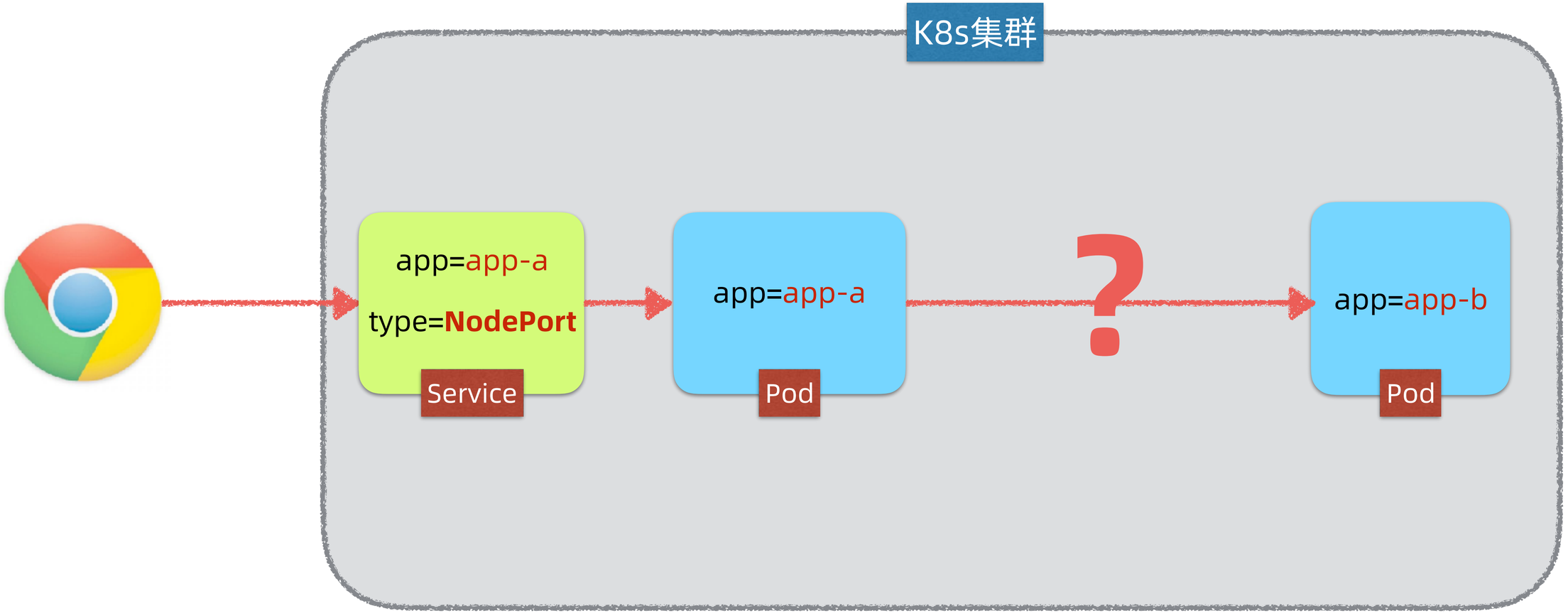
第 8 节

K8s内部反向代理 ~ ClusterIP Service

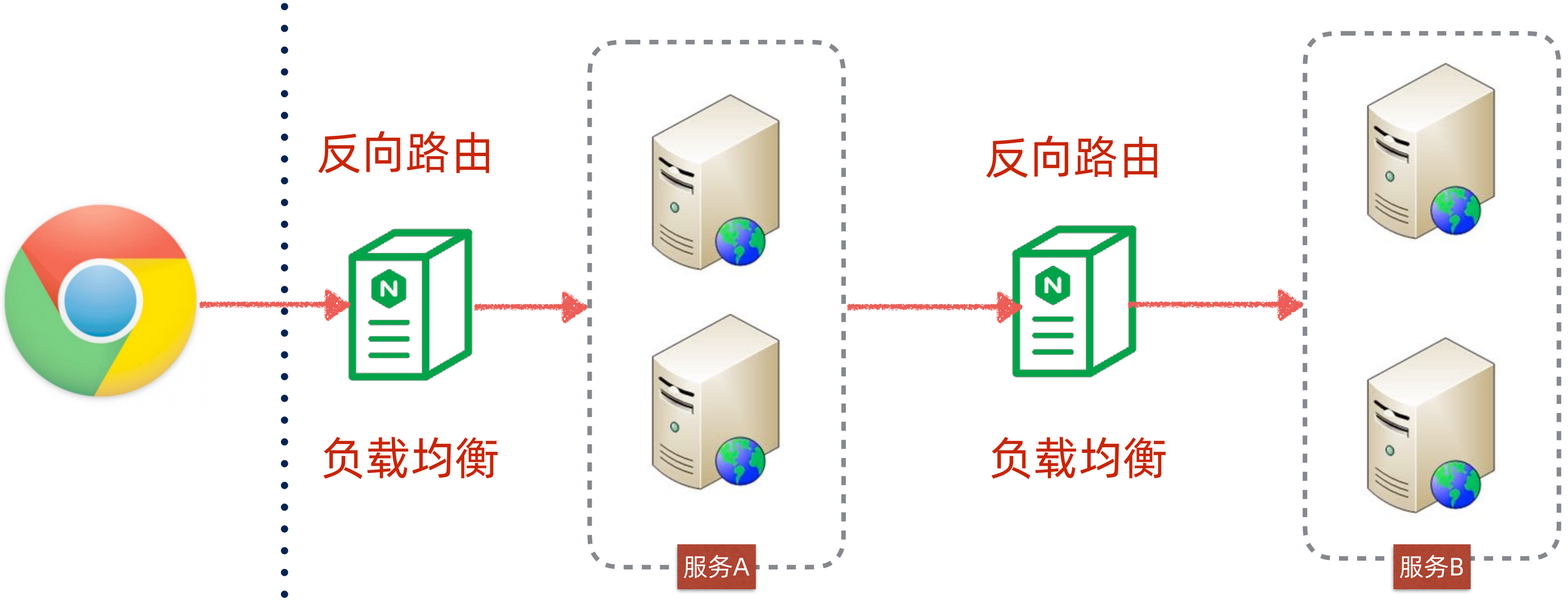
本课内容

- K8s内部反向代理ClusterIP Service原理
- 演示Petclinic通过ClusterIP Service访问内部MySql Pod

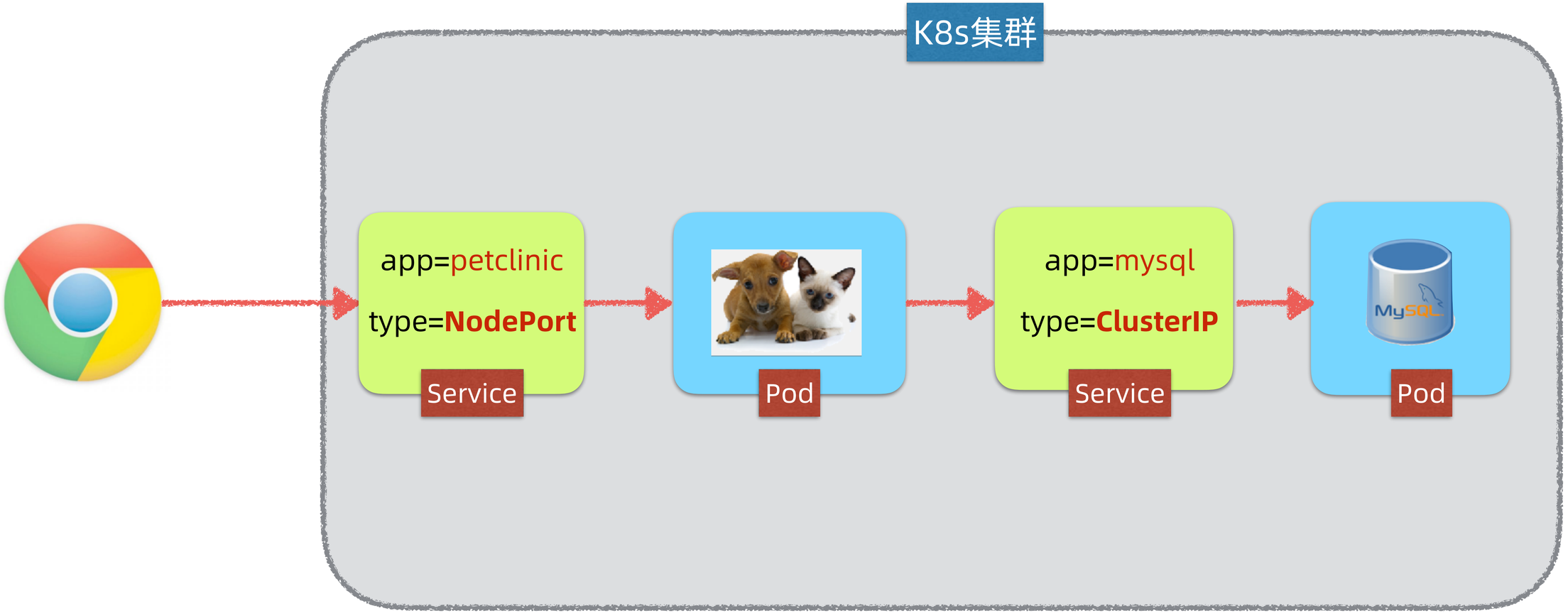
内部服务之间如何相互访问？



传统内部反向代理



K8s内部反向代理


K8s集群
app=petclinic
type=NodePort
Service
Pod
app=mysql
type=ClusterIP
Service
MySQL
Pod

Mysql Pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: mysql
  labels:
    app: mysql
spec:
  containers:
    - name: mysql
      image: mysql:5.7
      env:
        - name: MYSQL_ROOT_PASSWORD
          value: petclinic
        - name: MYSQL_DATABASE
          value: petclinic
```

Mysql Service

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  selector:
    app: mysql
  ports:
    - name: tcp
      port: 3306
      targetPort: 3306
  type: ClusterIP
```

PetClinic Deployment

spring-petclinic-mono项目源码

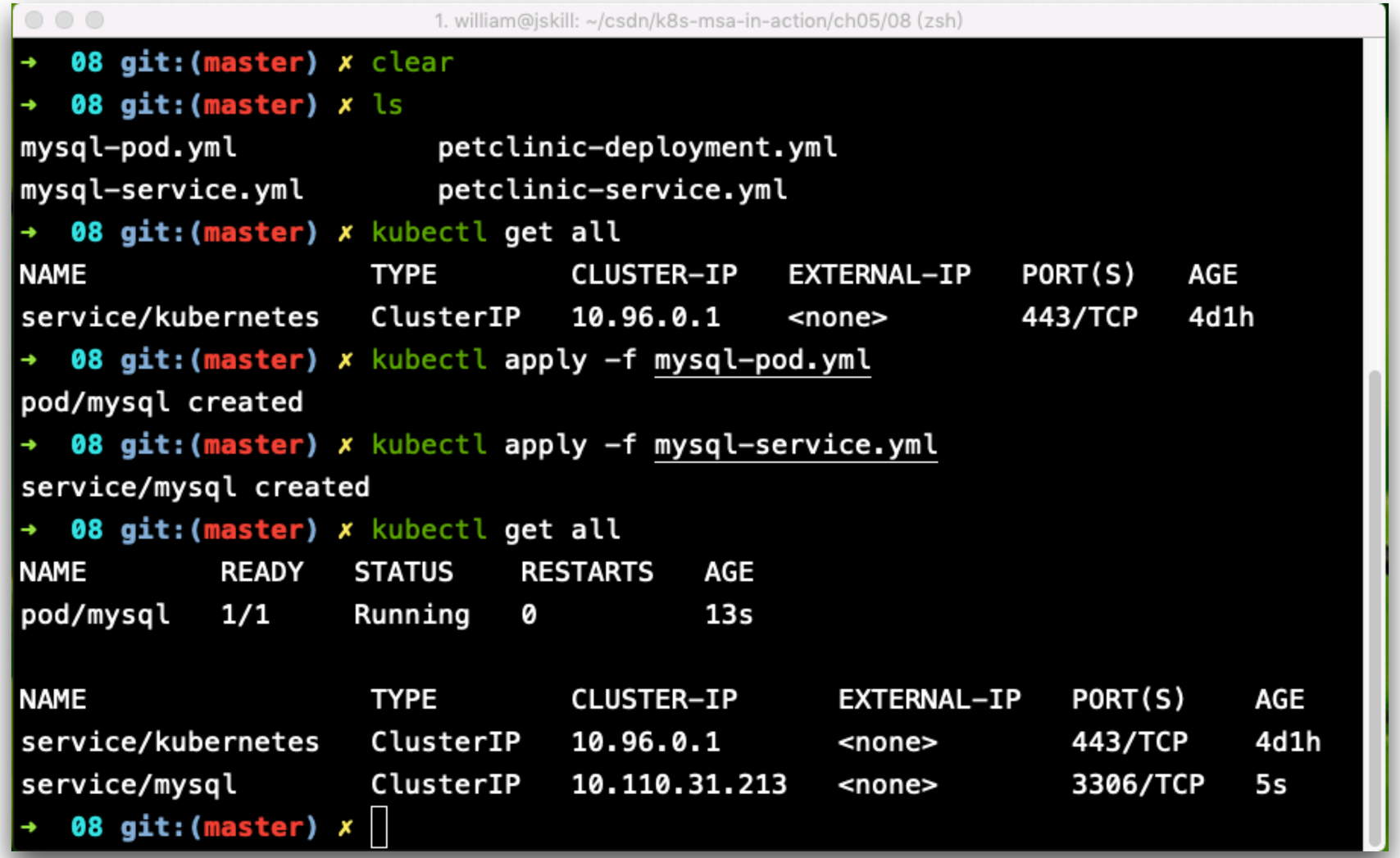
https://github.com/spring2go/spring-petclinic-mono

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: petclinic
spec:
  selector:
    matchLabels:
      app: petclinic
  replicas: 1
  template:
    metadata:
      labels:
        app: petclinic
    spec:
      containers:
        - name: petclinic
          image: spring2go/spring-petclinic:1.0.0.RELEASE
          env:
            - name: SPRING_PROFILES_ACTIVE
              value: mysql
            - name: DATASOURCE_URL
              value: jdbc:mysql://mysql/petclinic
            - name: DATASOURCE_USERNAME
              value: root
            - name: DATASOURCE_PASSWORD
              value: petclinic
            - name: DATASOURCE_INIT_MODE
              value: always
```

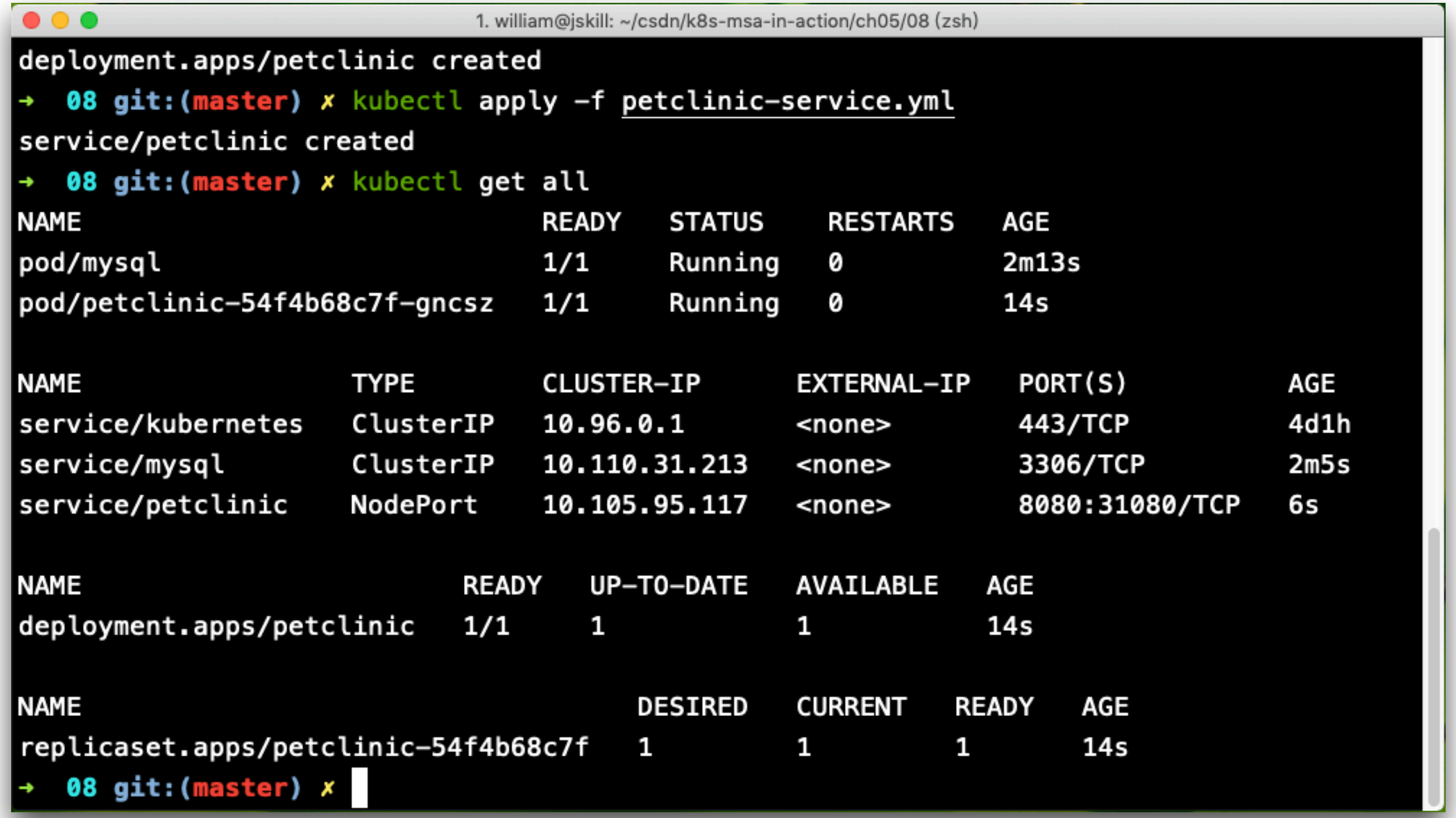
PetClinic Service

```yaml
apiVersion: v1
kind: Service
metadata:
  name: petclinic
spec:
  ports:
    - name: http
      port: 8080
      targetPort: 8080
      nodePort: 31080
  selector:
    app: petclinic
  type: NodePort
```

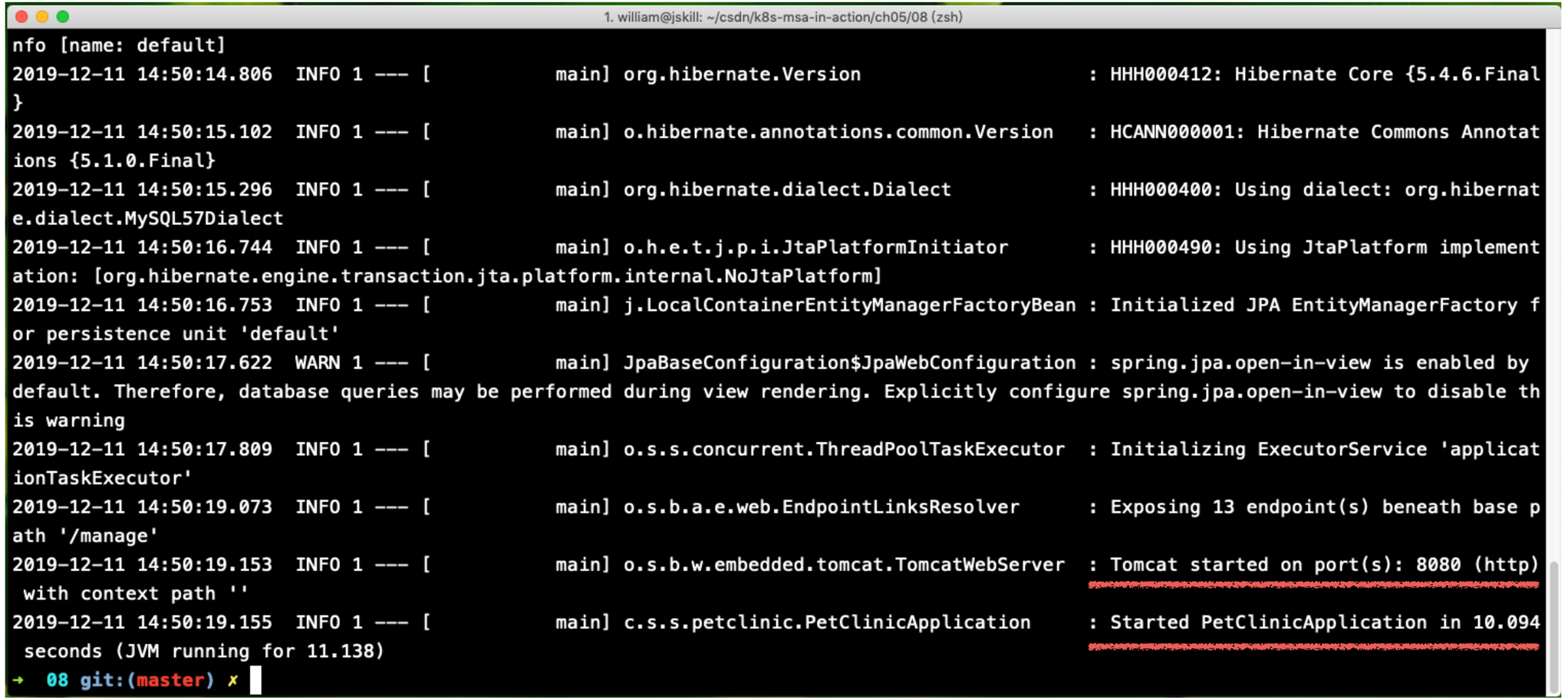
运行mysql pod和service



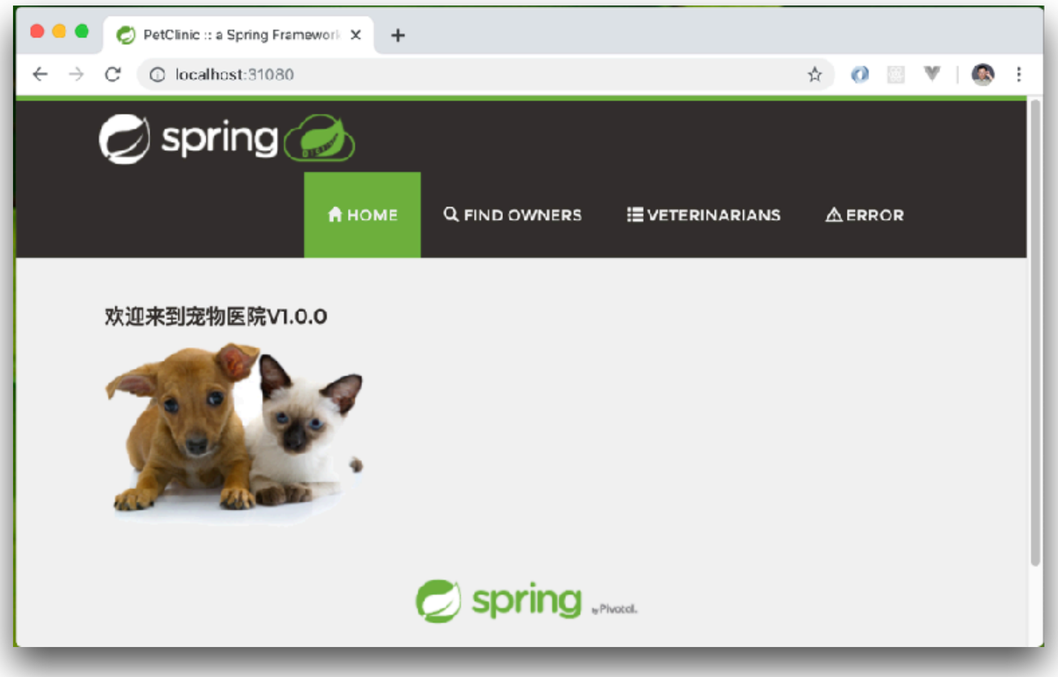
运行petclinic deployment和service

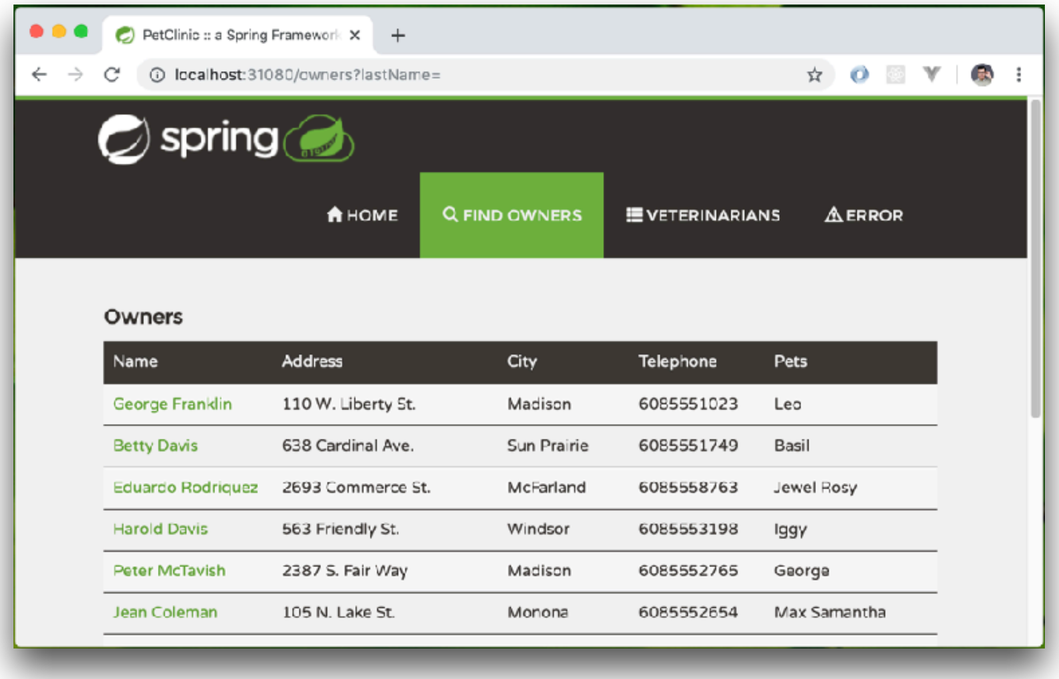


校验PetClinic应用正常启动



校验PetClinic应用





本课小结

- ClusterIP Service是K8s提供的一种内部反向代理机制
- 复习Service的三种类型
  - ClusterIP ~ 内部服务访问
  - NodePort ~ 对外暴露服务
  - LoadBalancer ~ 对外暴露服务(公有云)
- Deployment/Pod发布规范中可以添加环境变量，给Pod传递启动参数
- 有状态服务一般只能部署1个Pod实例