kaikeba

开课吧

Kubernetes 上手实践

@ 张晋涛

01 · 初识 Kubernetes

02 · 准备 Kubernetes 环境

03 · 上手实践

| 01 |

初识 Kubernetes

Kubernetes 发展历程

- 2014 年， Google 开源
- 大规模场景下 Docker 容器编排
- 隔离性 / 标准化
- 滚动更新
- 故障自愈
- 扩 / 缩容

Kubernetes 发展历程

- CNCF 首个毕业项目
- 生产中应用 Kubernetes 的比例达 83%
- GitHub star 数达到 75.3K

注：上述数据来自 CNCF 2020 调查报告

理解 Kubernetes

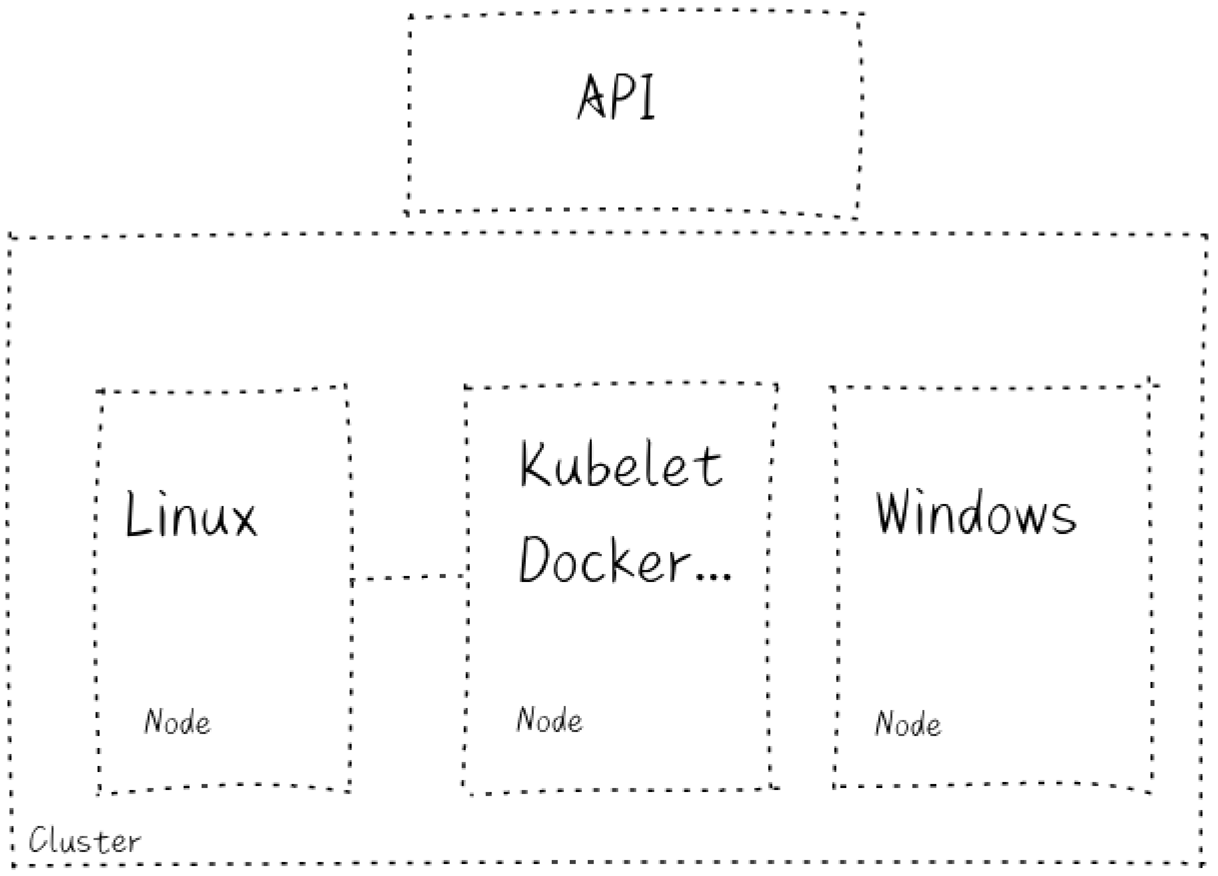
●API（控制面）

- 定义应用
- 多个组件协作

●Cluster

- 运行应用
- 一台或多台服务器
- Kubelet
- 容器运行时（ Docker 、 containerd 等）

理解 Kubernetes



如何在 K8S 中部署应用

- **按规范定义应用程序**
  - 通常使用 YAML 格式（为了易读性）
- **通过 API 发布应用**
  - 通常使用 kubectl 命令行工具
- **Cluster 运行应用**
  - 多副本
  - 高可用
  - 可通过网络交互

K8S 的其他主要资源类型

- Configmap
- Secret
- Service
- Volume
- …

| 02 |

准备 Kubernetes 环境

本地开发环境方案

- Docker Desktop 内置
- KIND （Kubernetes In Docker）
- minikube
- microk8s
- …

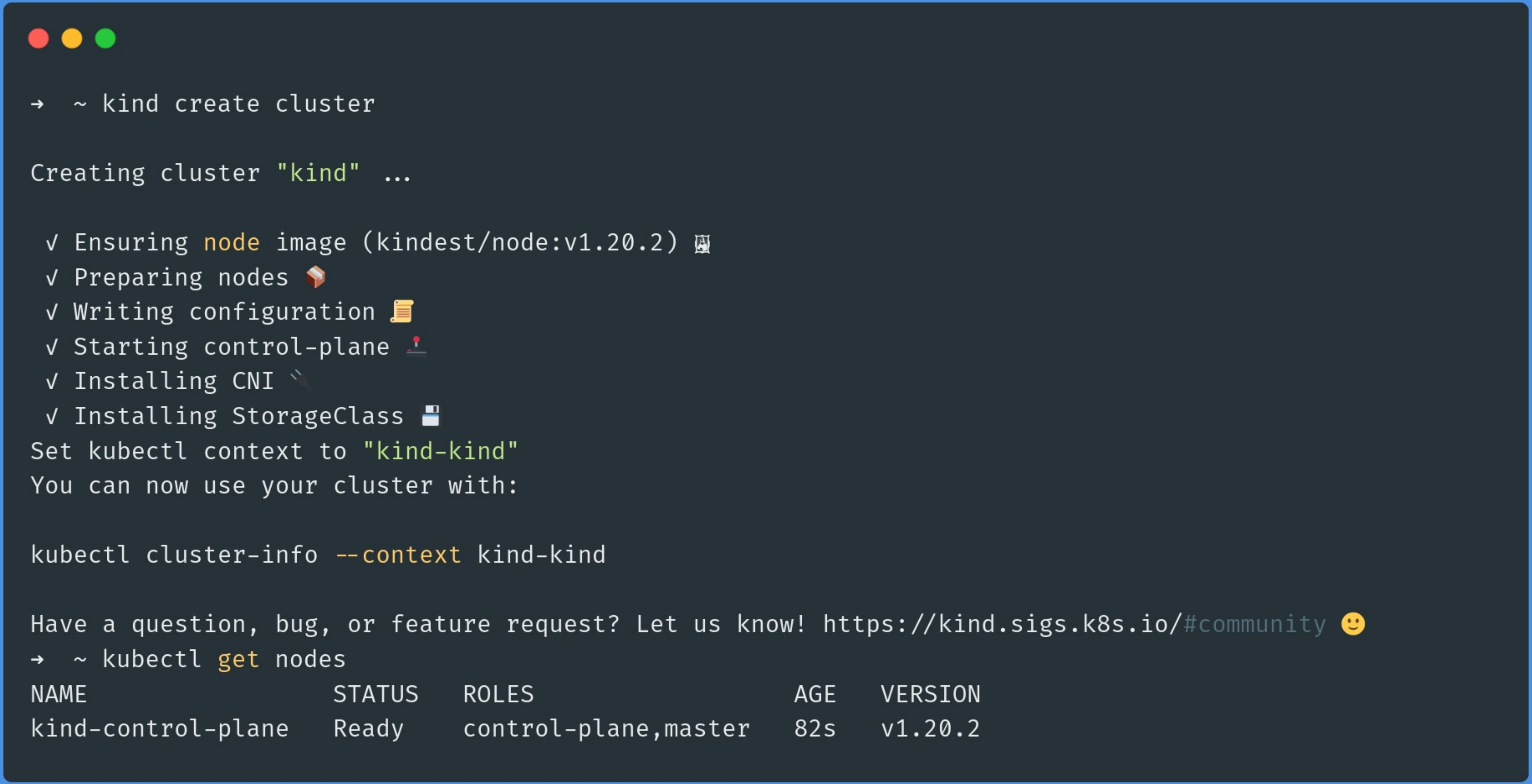
在线测试环境

- katakoda
- Play with Kubernetes
- …

使用 KIND 创建本地集群

- 下载 kind 二进制文件
- 安装 Docker 环境
- 文档：https://kind.sigs.k8s.io/

- kind create cluster



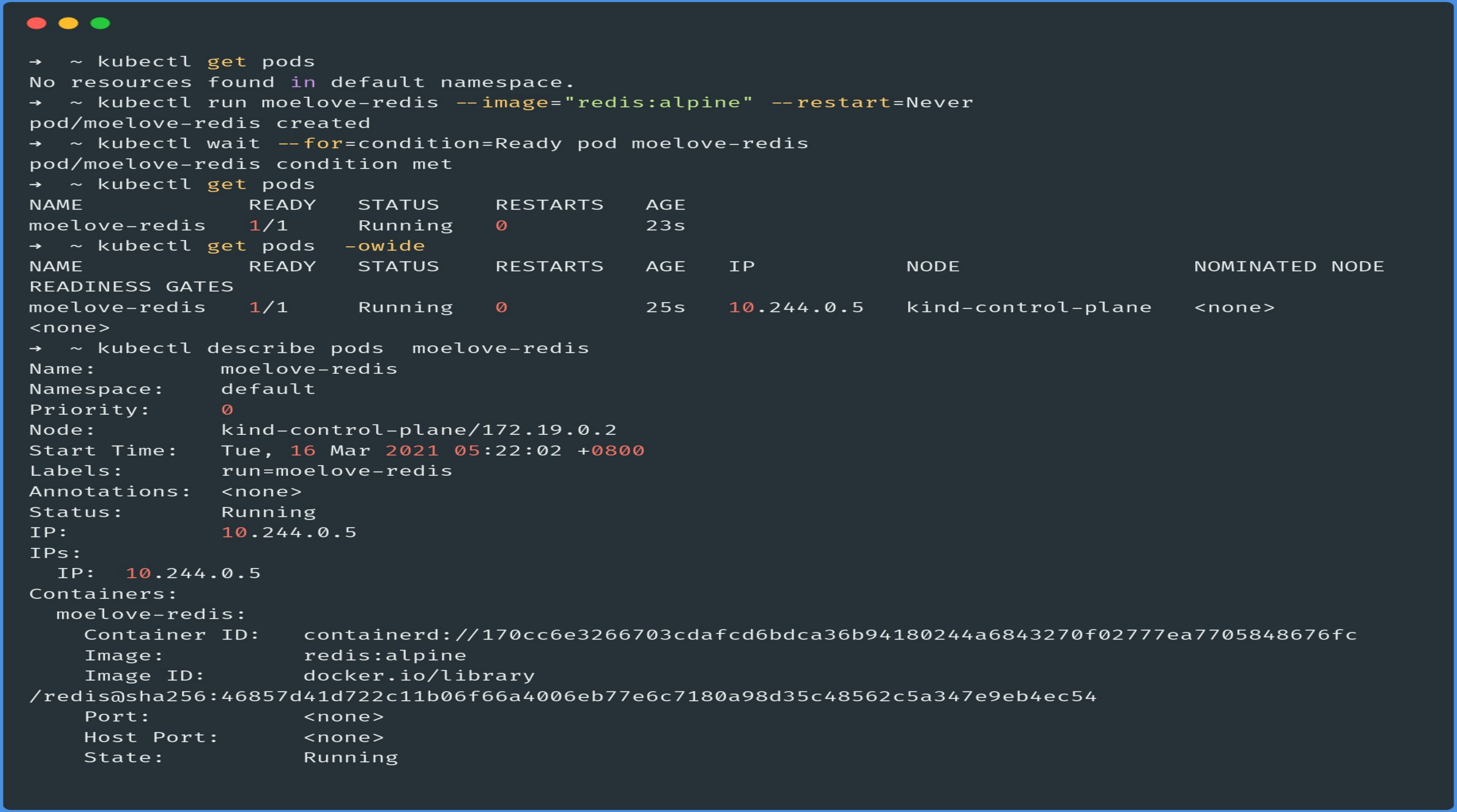
```
➜  ~ kind create cluster

Creating cluster "kind" ...

 ✓ Ensuring node image (kindest/node:v1.20.2) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 🙂
➜  ~ kubectl get nodes
NAME                 STATUS   ROLES                  AGE   VERSION
kind-control-plane   Ready    control-plane,master   82s   v1.20.2
```

| 03 |

Kubernetes 上手实践

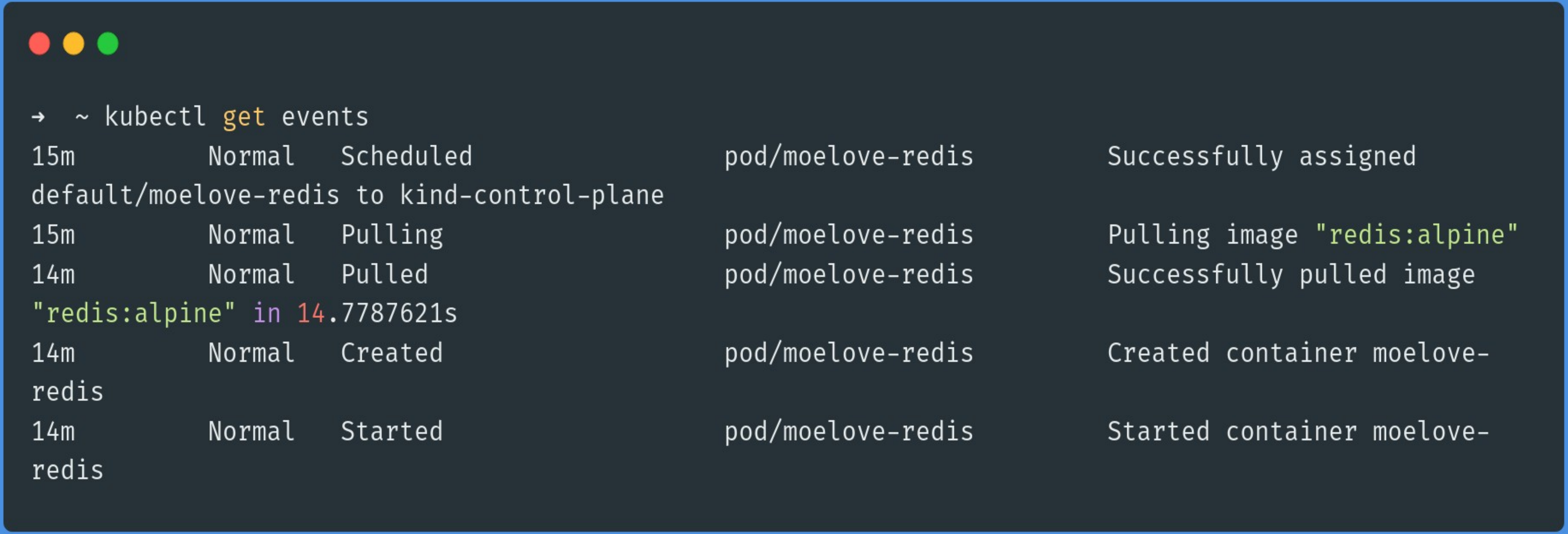
Kubernetes 如何管理容器

- K8S 中最小的单元是 Pod
- Pod 可以由多个容器组成
- 每个 Pod 有自己的 IP
- 可通过集群网络与其他 Pod 通信
- Pod 内容器共享 network namespace



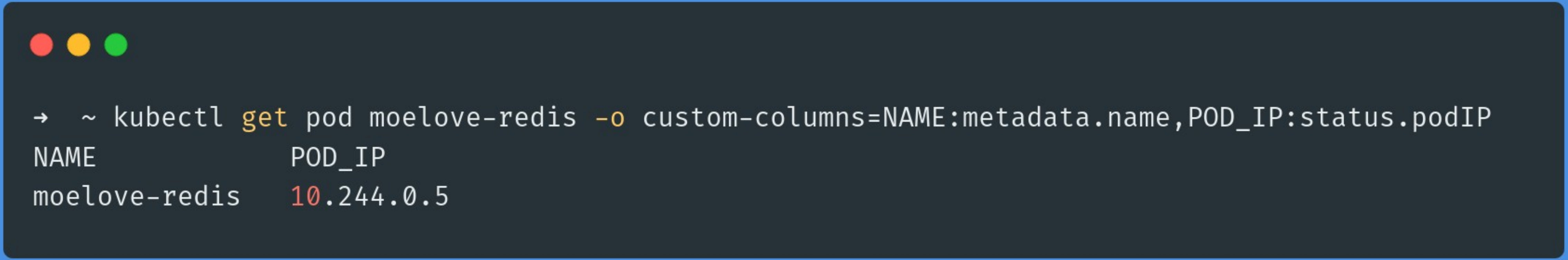
```
➜  ~ kubectl get pods
No resources found in default namespace.
➜  ~ kubectl run moelove-redis --image="redis:alpine" --restart=Never
pod/moelove-redis created
➜  ~ kubectl wait --for=condition=Ready pod moelove-redis
pod/moelove-redis condition met
➜  ~ kubectl get pods
NAME            READY   STATUS    RESTARTS   AGE
moelove-redis   1/1     Running   0          23s
➜  ~ kubectl get pods  -owide
NAME            READY   STATUS    RESTARTS   AGE   IP           NODE                 NOMINATED NODE
READINESS GATES
moelove-redis   1/1     Running   0          25s   10.244.0.5   kind-control-plane   <none>
<none>
➜  ~ kubectl describe pods  moelove-redis
Name:         moelove-redis
Namespace:    default
Priority:     0
Node:         kind-control-plane/172.19.0.2
Start Time:   Tue, 16 Mar 2021 05:22:02 +0800
Labels:       run=moelove-redis
Annotations:  <none>
Status:       Running
IP:           10.244.0.5
IPs:
  IP:  10.244.0.5
Containers:
  moelove-redis:
    Container ID:   containerd://170cc6e3266703cdafcd6bdca36b94180244a6843270f02777ea7705848676fc
    Image:          redis:alpine
    Image ID:       docker.io/library
/redis@sha256:46857d41d722c11b06f66a4006eb77e6c7180a98d35c48562c5a347e9eb4ec54
    Port:           <none>
    Host Port:      <none>
    State:          Running
```

K8S 启动 Pod 的过程

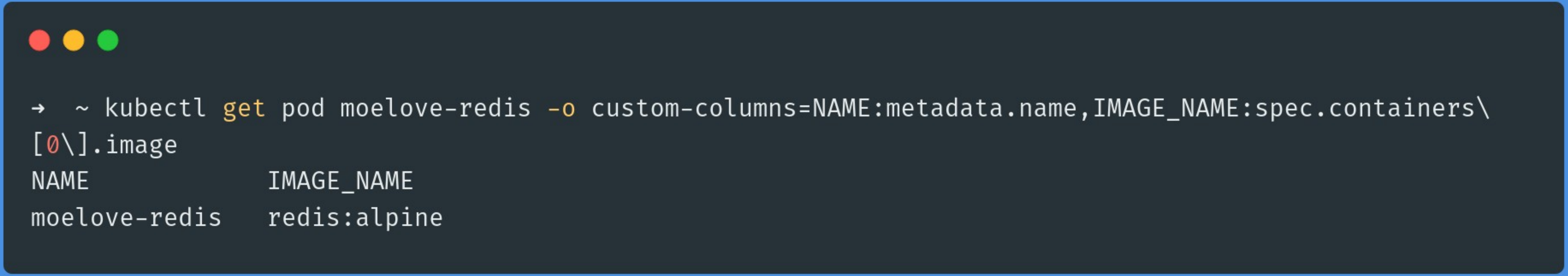


```
→  ~ kubectl get events
15m         Normal   Scheduled                 pod/moelove-redis         Successfully assigned
default/moelove-redis to kind-control-plane
15m         Normal   Pulling                   pod/moelove-redis         Pulling image "redis:alpine"
14m         Normal   Pulled                    pod/moelove-redis         Successfully pulled image
"redis:alpine" in 14.7787621s
14m         Normal   Created                   pod/moelove-redis         Created container moelove-
redis
14m         Normal   Started                   pod/moelove-redis         Started container moelove-
redis
```

Kubectl 的自定义输出

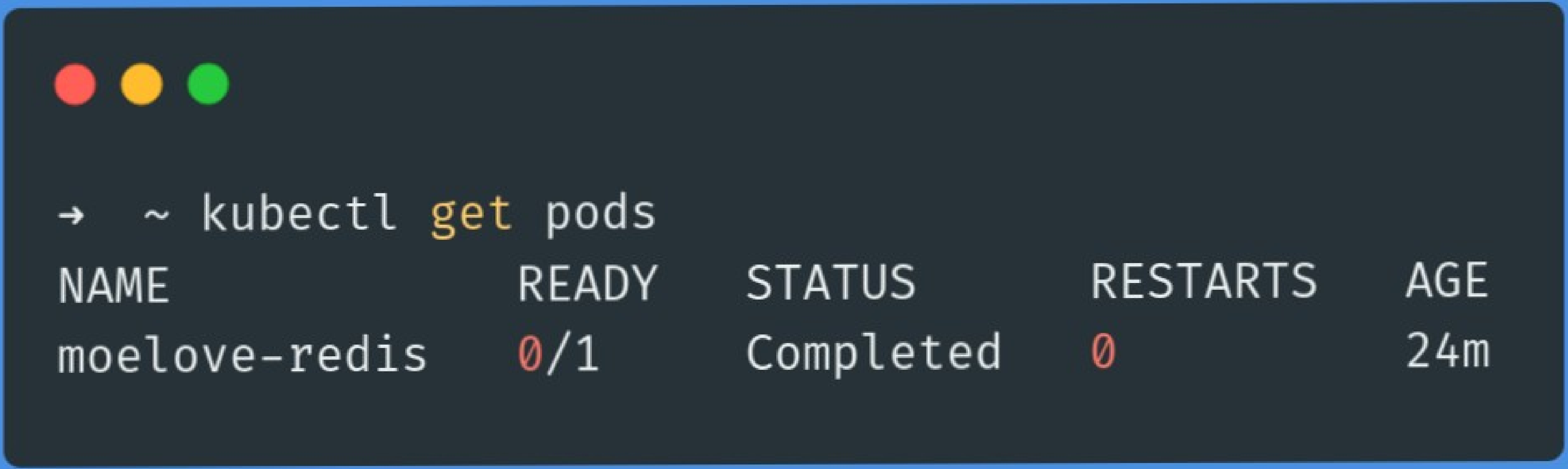


```
➜  ~ kubectl get pod moelove-redis -o custom-columns=NAME:metadata.name,POD_IP:status.podIP
NAME            POD_IP
moelove-redis   10.244.0.5
```



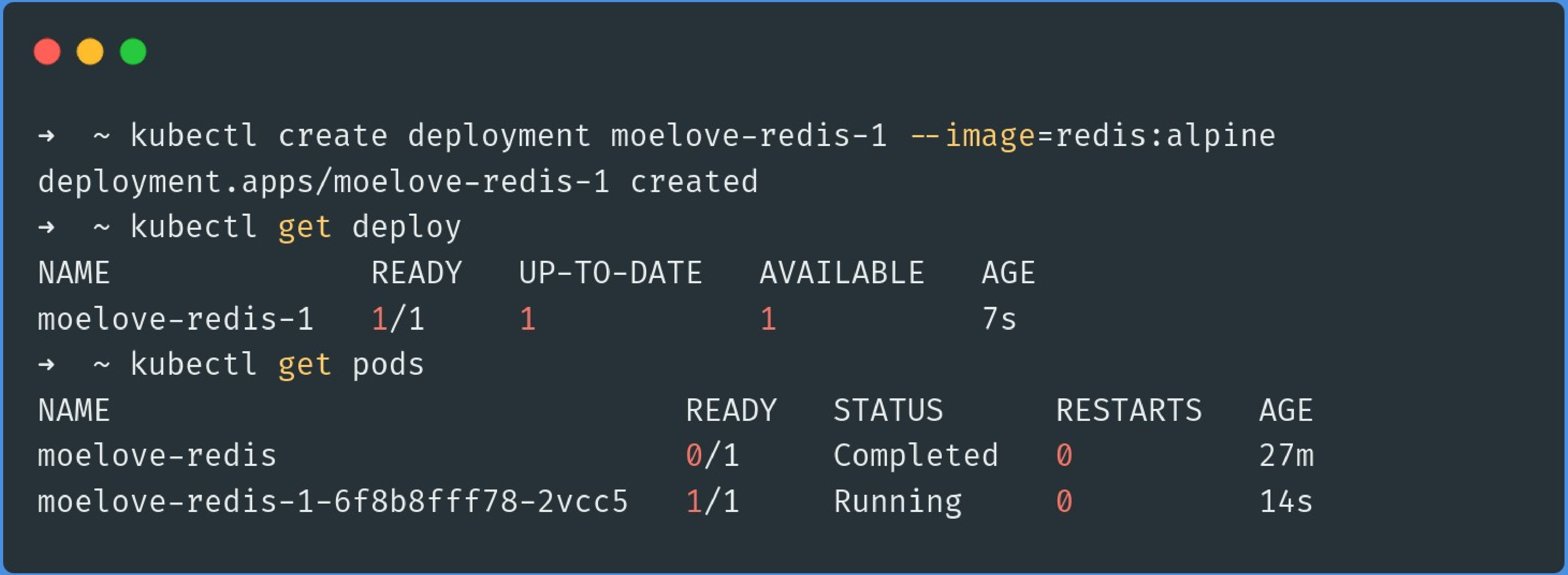
```
➜  ~ kubectl get pod moelove-redis -o custom-columns=NAME:metadata.name,IMAGE_NAME:spec.containers\
[0\].image
NAME            IMAGE_NAME
moelove-redis   redis:alpine
```

停止 container 后的状态



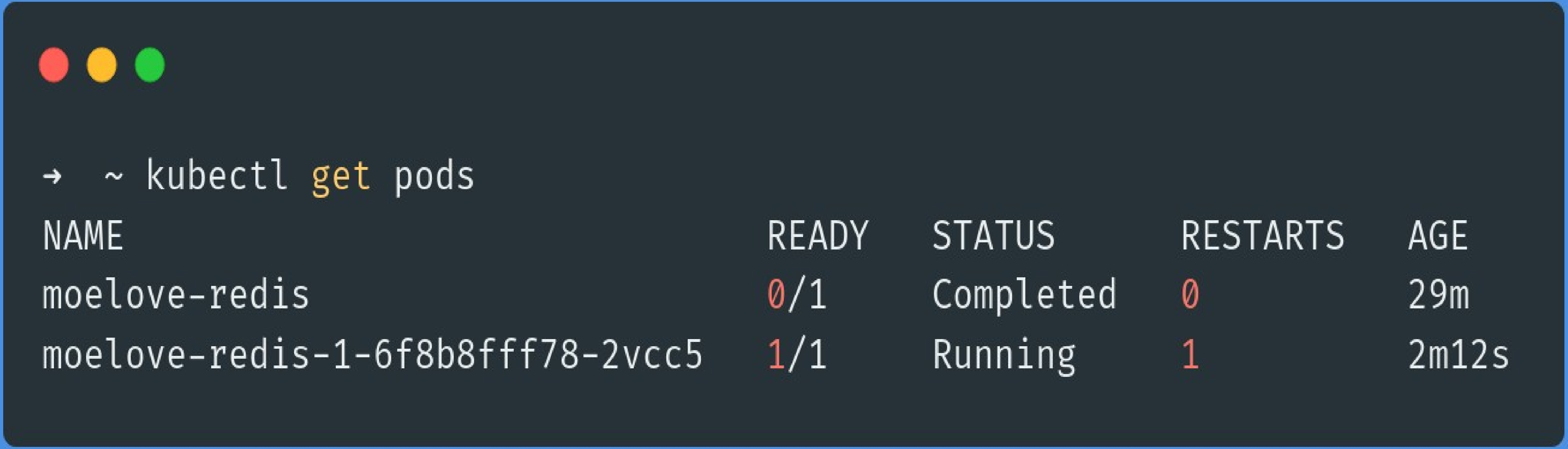
```
➜  ~ kubectl get pods
NAME            READY   STATUS      RESTARTS    AGE
moelove-redis   0/1     Completed   0           24m
```

使用 Deployment 部署



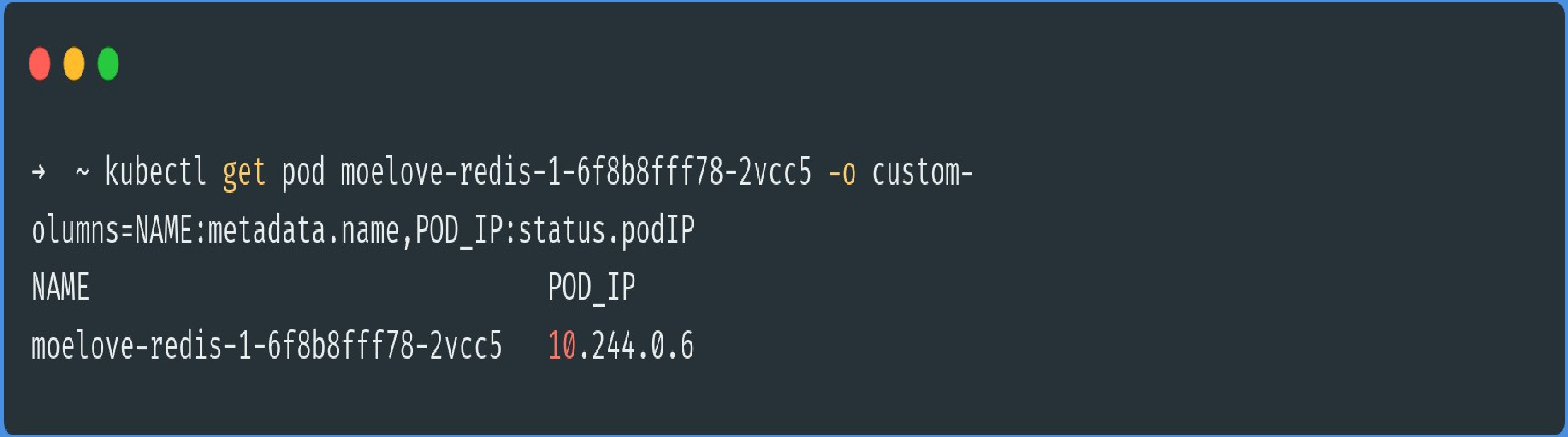
```
➜  ~ kubectl create deployment moelove-redis-1 --image=redis:alpine
deployment.apps/moelove-redis-1 created
➜  ~ kubectl get deploy
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
moelove-redis-1   1/1     1            1           7s
➜  ~ kubectl get pods
NAME                               READY   STATUS      RESTARTS   AGE
moelove-redis                      0/1     Completed   0          27m
moelove-redis-1-6f8b8fff78-2vcc5   1/1     Running     0          14s
```

停止 Deployment 的 Pod 中的容器



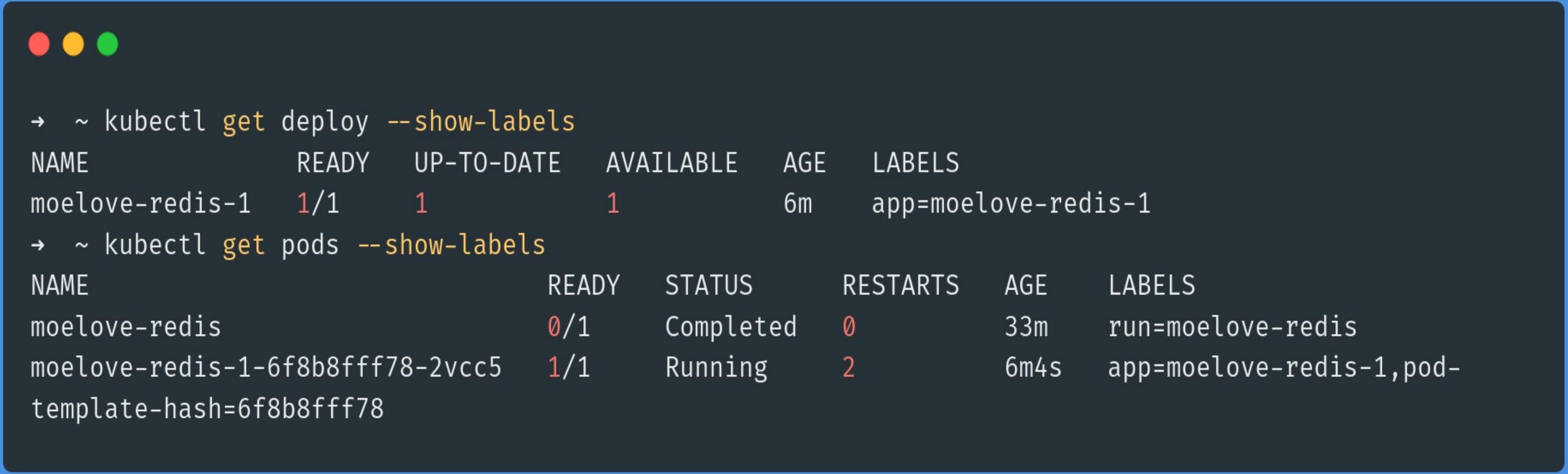
```
➜  ~ kubectl get pods
NAME                              READY   STATUS      RESTARTS   AGE
moelove-redis                     0/1     Completed   0          29m
moelove-redis-1-6f8b8fff78-2vcc5  1/1     Running     1          2m12s
```

只停止业务容器， IP 未变化



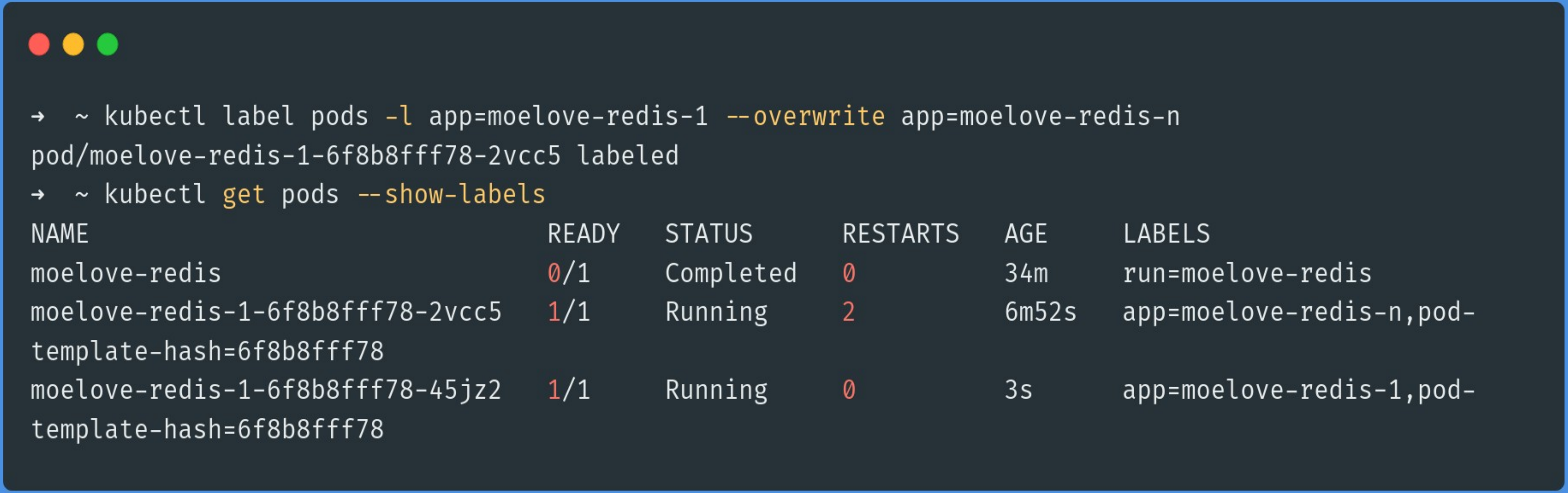
```
➜  ~ kubectl get pod moelove-redis-1-6f8b8fff78-2vcc5 -o custom-
olumns=NAME:metadata.name,POD_IP:status.podIP
NAME                               POD_IP
moelove-redis-1-6f8b8fff78-2vcc5   10.244.0.6
```

标签



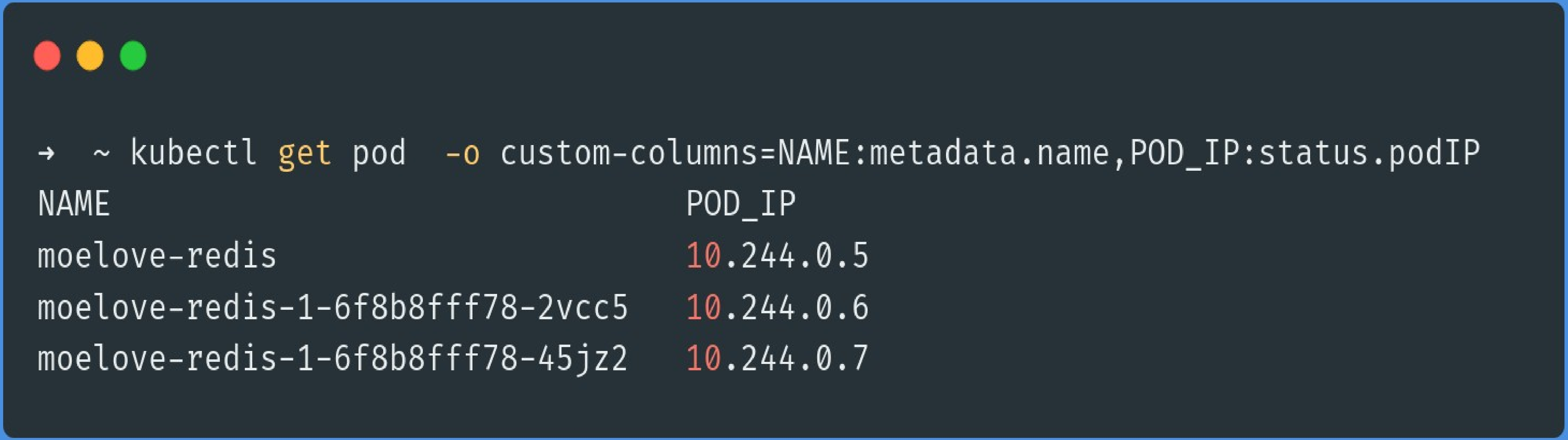
```
➜  ~ kubectl get deploy --show-labels
NAME              READY   UP-TO-DATE   AVAILABLE   AGE   LABELS
moelove-redis-1   1/1     1            1           6m    app=moelove-redis-1
➜  ~ kubectl get pods --show-labels
NAME                               READY   STATUS      RESTARTS   AGE    LABELS
moelove-redis                      0/1     Completed   0          33m    run=moelove-redis
moelove-redis-1-6f8b8fff78-2vcc5   1/1     Running     2          6m4s   app=moelove-redis-1,pod-
template-hash=6f8b8fff78
```

覆盖标签



```
➜  ~ kubectl label pods -l app=moelove-redis-1 --overwrite app=moelove-redis-n
pod/moelove-redis-1-6f8b8fff78-2vcc5 labeled
➜  ~ kubectl get pods --show-labels
NAME                               READY   STATUS      RESTARTS   AGE     LABELS
moelove-redis                      0/1     Completed   0          34m     run=moelove-redis
moelove-redis-1-6f8b8fff78-2vcc5   1/1     Running     2          6m52s   app=moelove-redis-n,pod-
template-hash=6f8b8fff78
moelove-redis-1-6f8b8fff78-45jz2   1/1     Running     0          3s      app=moelove-redis-1,pod-
template-hash=6f8b8fff78
```

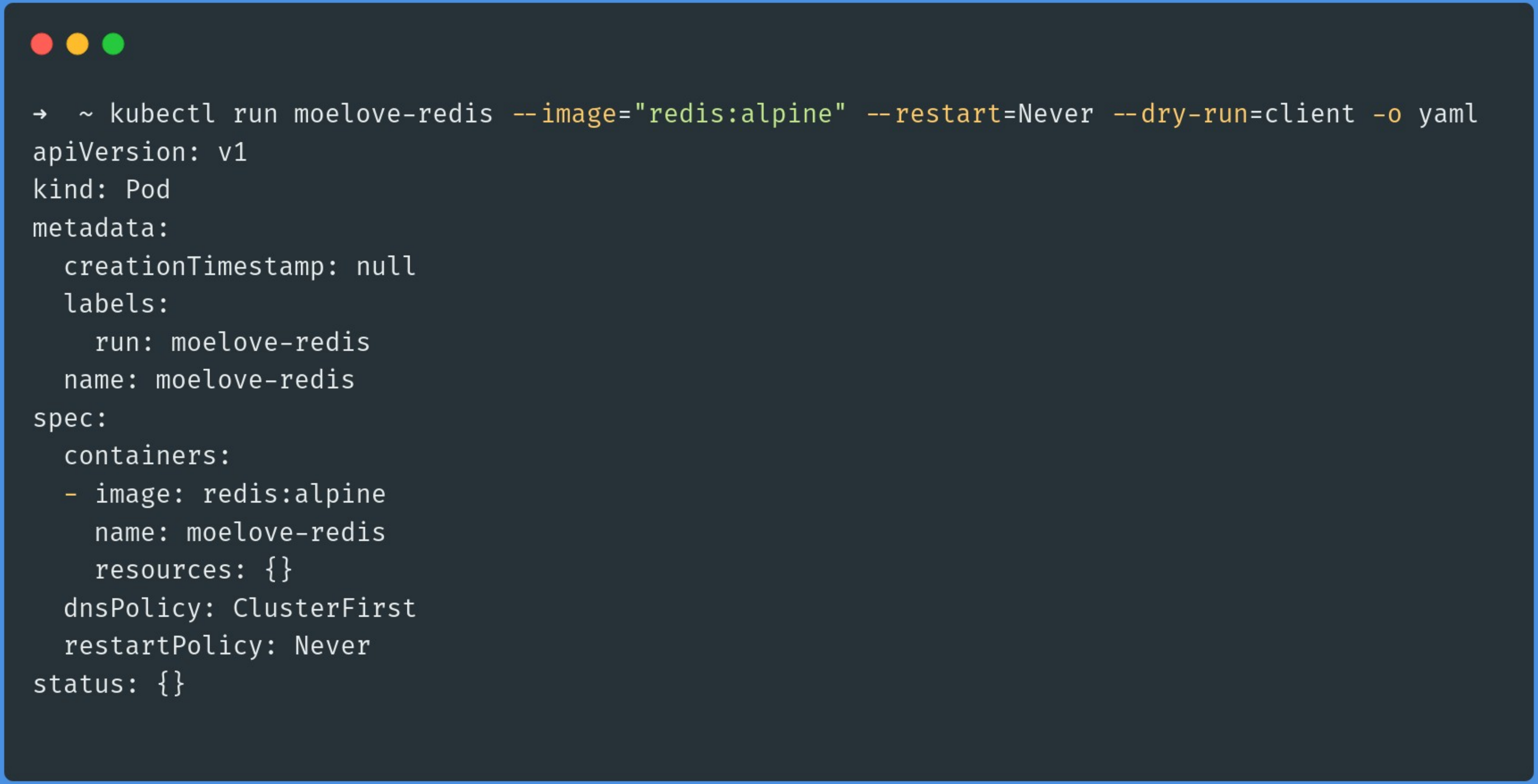
新启动 Pod IP 发生变更



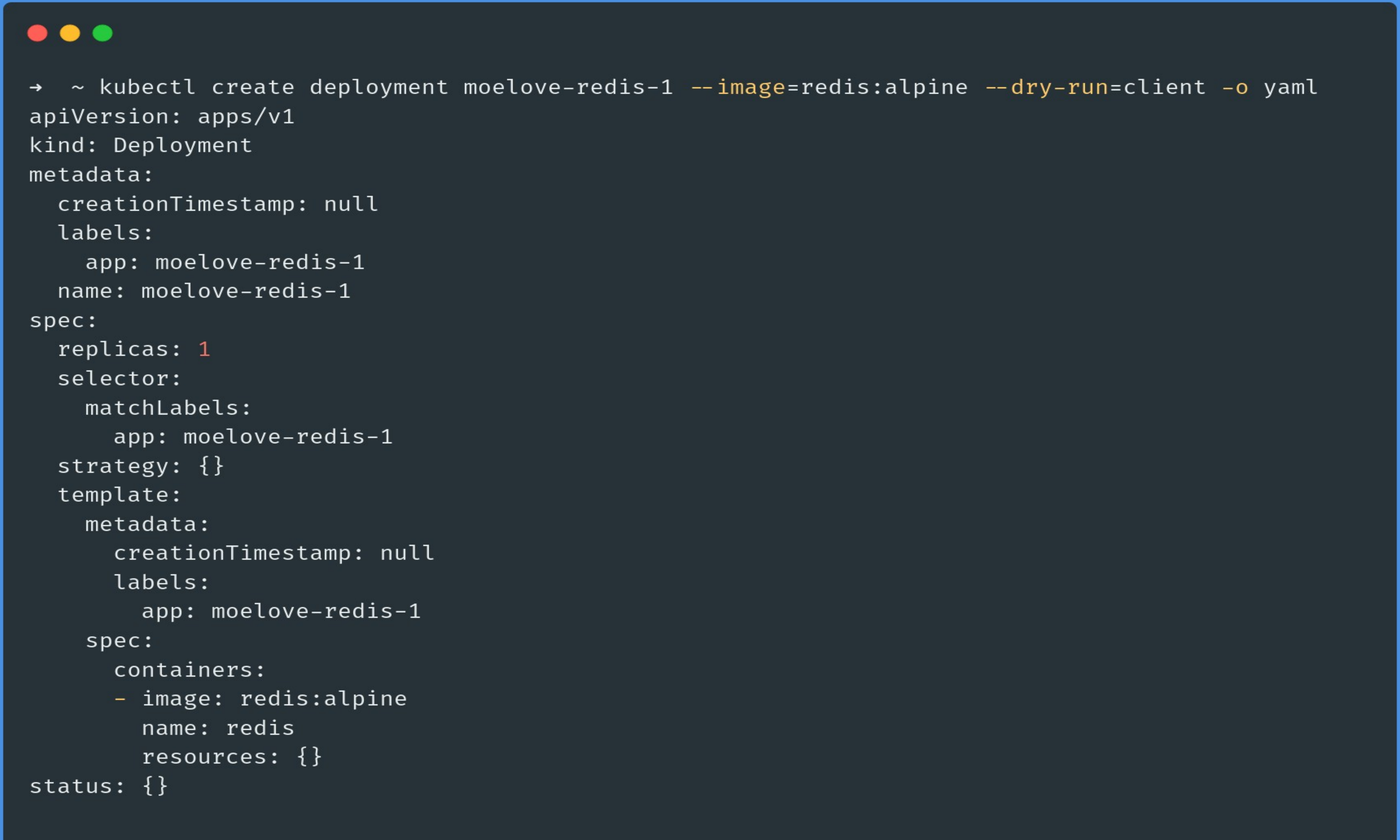
```
➜  ~ kubectl get pod  -o custom-columns=NAME:metadata.name,POD_IP:status.podIP
NAME                            POD_IP
moelove-redis                   10.244.0.5
moelove-redis-1-6f8b8fff78-2vcc5   10.244.0.6
moelove-redis-1-6f8b8fff78-45jz2   10.244.0.7
```

| 04 |

用配置文件部署

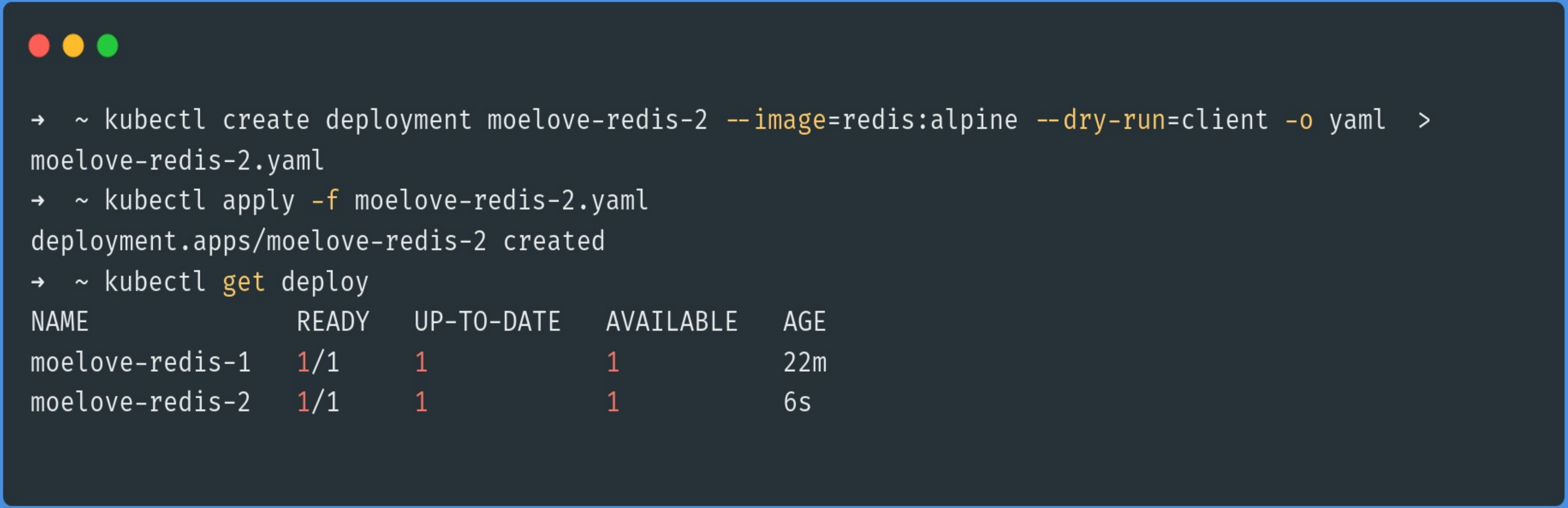


```
➜  ~ kubectl run moelove-redis --image="redis:alpine" --restart=Never --dry-run=client -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: moelove-redis
  name: moelove-redis
spec:
  containers:
  - image: redis:alpine
    name: moelove-redis
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Never
status: {}
```



```
➜  ~ kubectl create deployment moelove-redis-1 --image=redis:alpine --dry-run=client -o yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: moelove-redis-1
  name: moelove-redis-1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: moelove-redis-1
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: moelove-redis-1
    spec:
      containers:
      - image: redis:alpine
        name: redis
        resources: {}
status: {}
```

使用配置文件部署



```
➜  ~ kubectl create deployment moelove-redis-2 --image=redis:alpine --dry-run=client -o yaml  >
moelove-redis-2.yaml
➜  ~ kubectl apply -f moelove-redis-2.yaml
deployment.apps/moelove-redis-2 created
➜  ~ kubectl get deploy
NAME              READY   UP-TO-DATE   AVAILABLE   AGE
moelove-redis-1   1/1     1            1           22m
moelove-redis-2   1/1     1            1           6s
```

总结

- Kubernetes 概览
- 本地 Kubernetes 环境搭建
- 理解 Pod 和 Deployment 资源及其行为
- 使用配置文件完成容器的部署

Reference

- Kubernetes **官网：** https://kubernetes.io/
- KIND **官网：** https://kind.sigs.k8s.io/
- **使用** KIND **搭建本地环境：** https://zhuanlan.zhihu.com/p/105173589

kaikeba
开课吧

扫码关注开课吧

官方网站
www.kaikeba.com

联系电话
400-996-0826