Lab 6 Report

Course: Cloud_Computing
Student: ZhengYang
GitHub Repo: y*angzheng.github.io — Repository for personal website project using GitHub Pages*
Date: 2025.11.3

1. Objectives

Install and run a local single-node Kubernetes cluster using Minikube.

Deploy, explore, expose, scale, and update an application using `kubectl`.

Document the steps with screenshots and explanations.

2. Environment & Prerequisites

Operating System: windows11

Minikube Version: minikube version: v1.37.0

Kubectl Version:

Client Version: v1.34.0

Kustomize Version: v5.7.1

Server Version: v1.34.0

Driver: docker

Hardware: ≥ 2 CPUs, ≥ 2 GB RAM, ≥ 20 GB disk space

3. Lab Steps & Observations

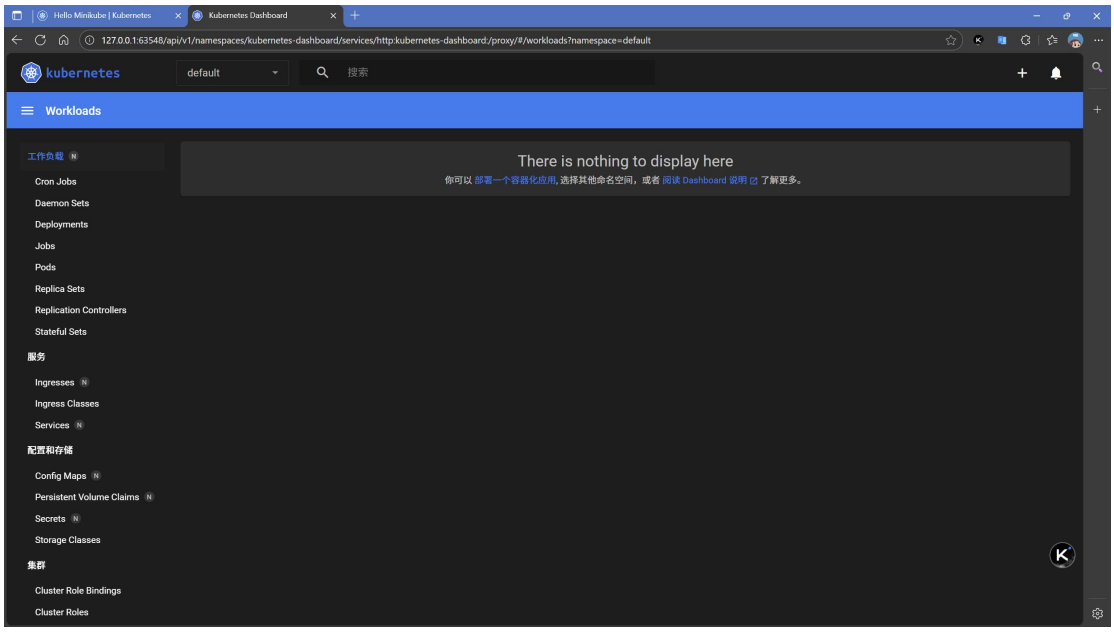
For each section below, include your **commands**, **screenshots**, **observations**, and **notes**.

3.1 Start Minikube Cluster

Commands:

```
minikube start
Minikube dashboard
kubectl version
kubectl get nodes
```

Screenshots:




```
PS C:\Windows\system32> minikube dashboard
* 正在验证 dashboard 运行情况 ...
* 正在启动代理...
* 正在验证 proxy 运行状况 ...
* 正在使用默认浏览器打开 http://127.0.0.1:63973/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ ...
```

Observations:

Cluster started successfully

Dashboard started successfully

3.2 Deploy Application

Commands:

```
kubectl create deployment hello-minikube
--image=kicbase/echo-server:1.0
kubectl get deployments
kubectl get pods
```

Screenshots:

```
PS C:\Windows\system32> kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
```

```
PS C:\Windows\system32> kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-6c9b5f4b59-jfgbg   1/1     Running   0          60s
```

Observations:

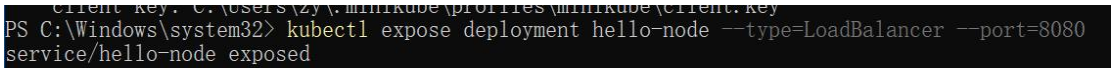
Deployment created and pod(s) are running.

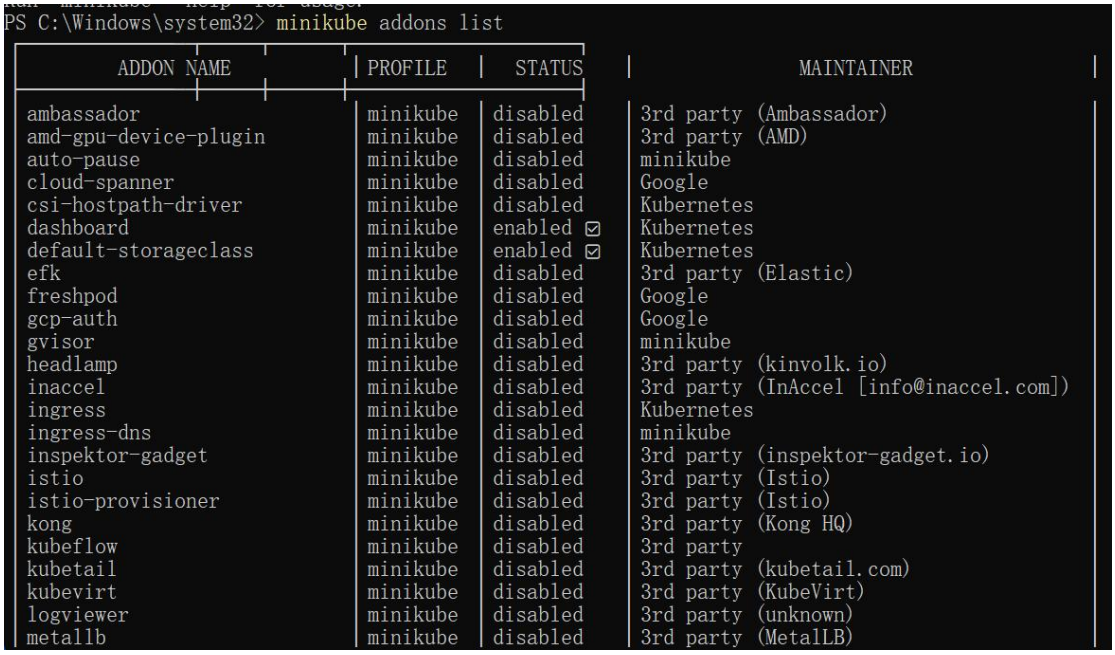
3.3 Explore Resources

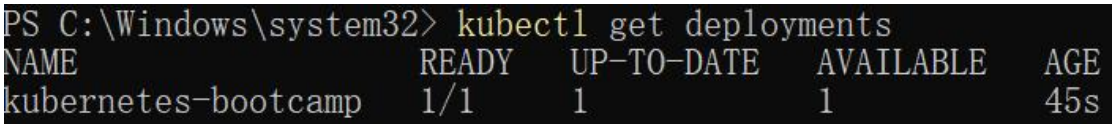
Commands:

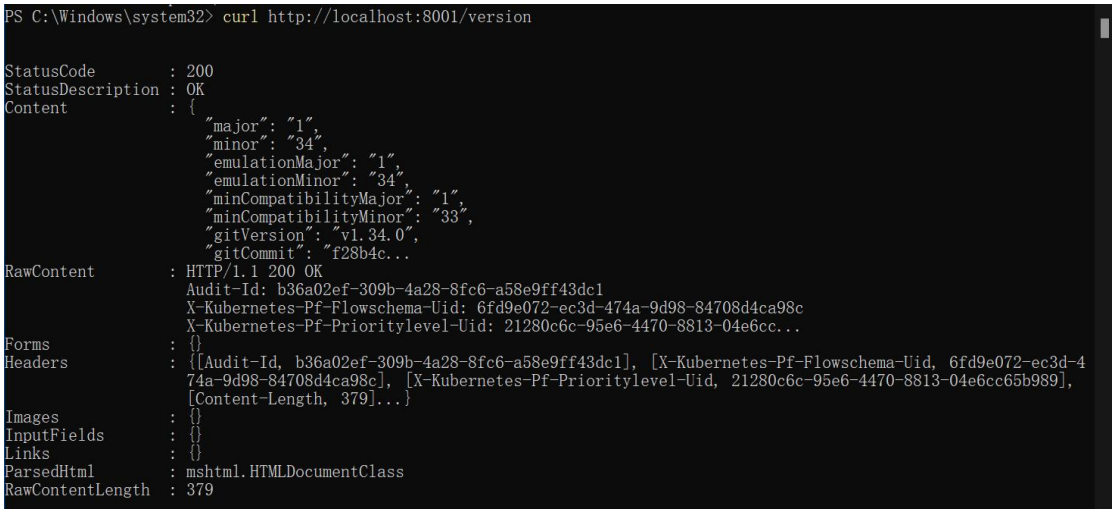
```
kubectl get pods
kubectl describe pod <pod-name>
kubectl logs <pod-name>
```

Screenshot:









Observations:

Pod details and logs retrieved successfully.

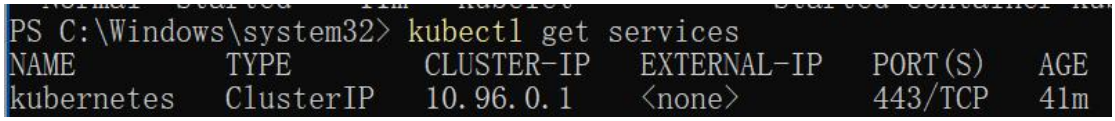
Notes: Add findings or unexpected log behavior.

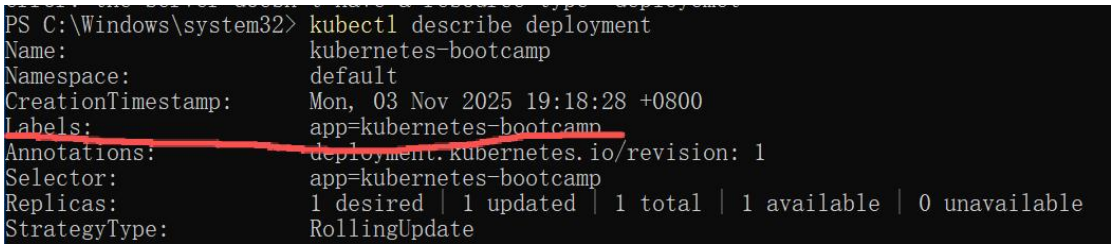
3.4 Expose Service

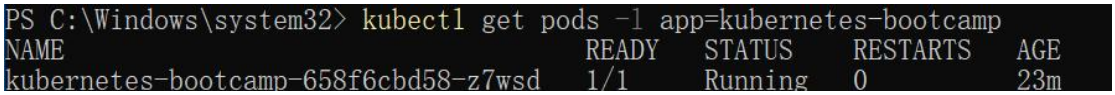
Commands:

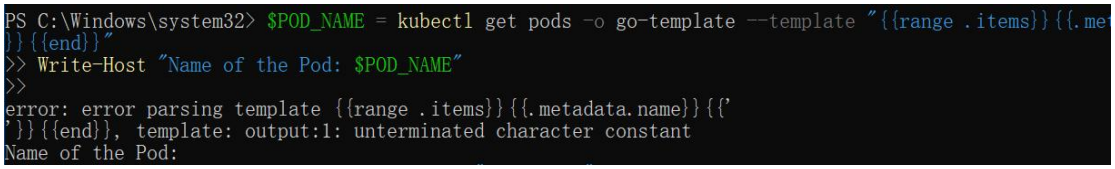
```
kubectl expose deployment hello-minikube --type=NodePort --port=8080
kubectl get services
minikube service hello-minikube --url
```

Screenshot:









Observations:

Service exposed successfully.

Accessible via NodePort URL.

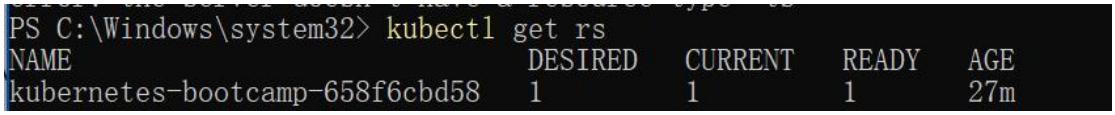
Notes: Record NodePort number or accessibility issues.

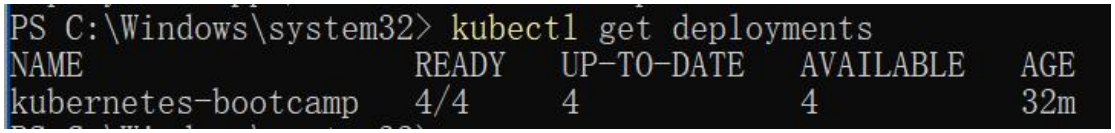
3.5 Scale Application

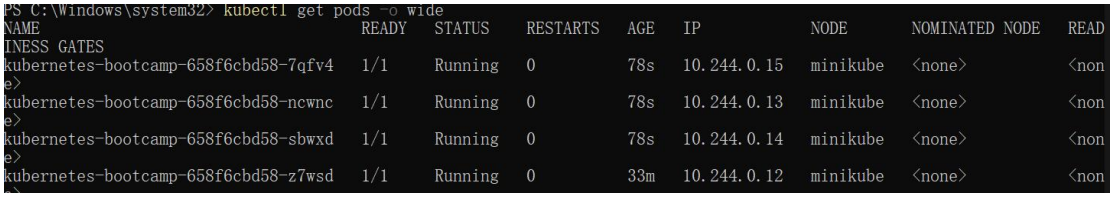
Commands:

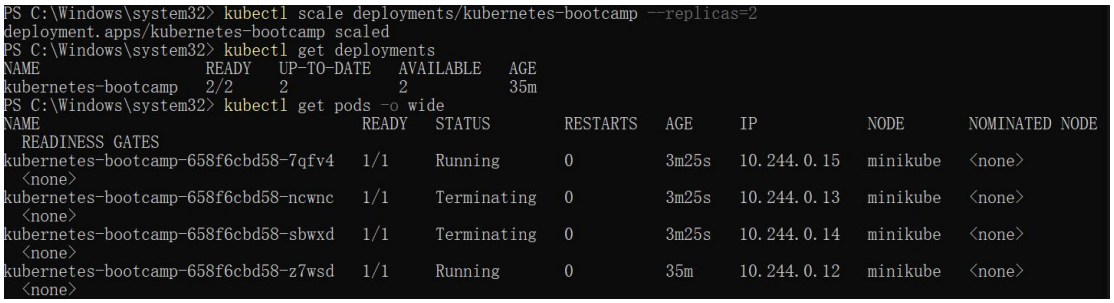
```
kubectl scale deployment hello-minikube --replicas=3
kubectl get pods
```

Screenshot:









Observations:

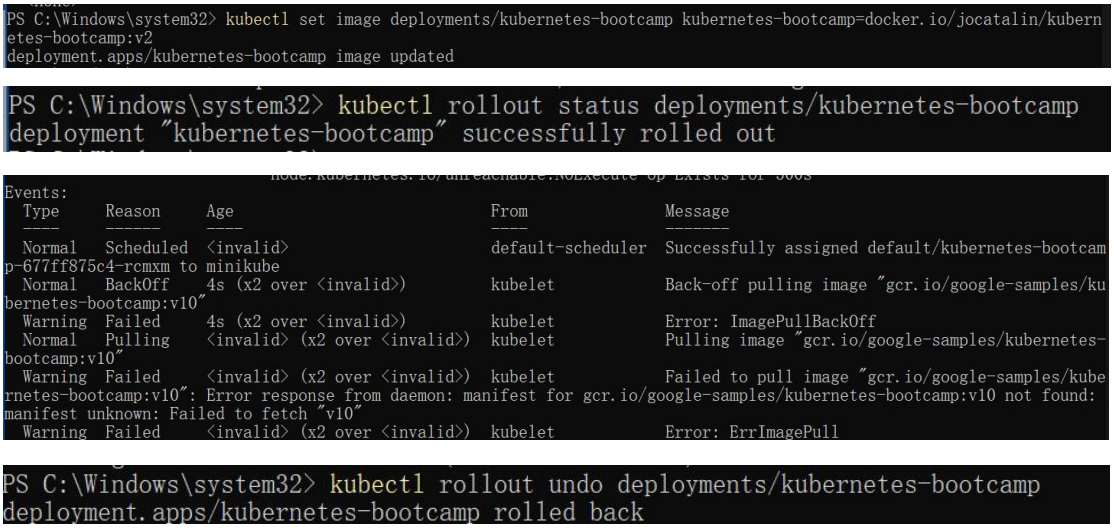
Deployment scaled up to 4 pods and then scaled down to 2 pods.
Notes: Mention any differences in load balancing or startup times.

3.6 Update Application

Commands:

```
kubectl set image deployment/hello-minikube
hello-minikube=kicbase/echo-server:2.0
kubectl rollout status deployment/hello-minikube
kubectl get pods
```

Screenshot:



Observations:

Deployment updated or rolled back successfully.
Notes: Record whether the update was smooth or required rollback.

4. Summary & Lessons Learned

What went well: *Smooth deployment, quick Minikube startup.*

Challenges: *Driver or network issues.*

Key Takeaways:

Kubernetes resource hierarchy and relationships.

How deployments manage pods and rollouts.

How scaling and updates maintain app availability.