

# Kubernetes Fundamentals Lab Report

**Student Name:** [Your Name]

**Date:** December 10, 2025

**Lab Duration:** 3 hours

**Kubernetes Version:** 1.31

**Minikube Version:** 1.34

---

## 1. Lab Overview

This lab sequentially walks through six official Kubernetes tutorials, beginning with cluster creation on minikube and ending with a zero-downtime rolling update. The objective is to observe how the basic primitives—Pods, Deployments, Services, ReplicaSets, and labels—cooperate to deliver a resilient, externally accessible application.

---

## 2. Environment Preparation

Minikube was started with the Docker driver and 4 GB RAM allocated. The dashboard addon was enabled to provide a visual counterpart to CLI operations. kubectl was installed and verified against the minikube context. No cloud credentials were required because the LoadBalancer emulation provided by minikube was sufficient for external access.

---

### 3. Hello Minikube Walk-through

A single-replica Deployment named **hello-node** was created from the agnhost image configured to listen on TCP 8080. kubectl get deployments confirmed the desired state (1/1) within 15 seconds. The Pod appeared in Running status shortly afterward. Application logs revealed the embedded HTTP and UDP servers starting without errors.

A LoadBalancer Service was declared next. Because minikube cannot provision a cloud load balancer, the minikube service command tunnelled the NodePort to the host. A browser automatically opened and displayed the echo response, proving external reachability. The exercise concluded by deleting both the Service and Deployment; the Pod was garbage-collected immediately.

**Key observation:** The Service abstraction decouples external traffic from Pod lifecycle events, even in a single-node cluster.

---

### 4. Deploying an App via kubectl

The tutorial image gcr.io/google-samples/kubernetes-bootcamp:v1 was used to create a Deployment called **kubernetes-bootcamp**. kubectl get deployments showed 1/1 replicas ready. A proxy was launched in a second terminal, exposing the API server on localhost:8001. By exporting the Pod name to an environment variable, curl was able to hit the pod's proxy endpoint and receive the expected "Hello Kubernetes bootcamp!" payload.

**Take-away:** The proxy mechanism provides convenient introspection without exposing the Pod to the public Internet.

---

## 5. Exploring Pods and Nodes

kubectl describe pod revealed the container image, restart count, assigned node IP, and the default label app=kubernetes-bootcamp. Exec'ing into the container demonstrated process isolation: the Node.js server continued to respond on port 8080 while the host filesystem remained invisible. Listing environment variables inside the container confirmed that Kubernetes injects SERVICE\_HOST and SERVICE\_PORT entries automatically, foreshadowing the Service discovery feature used later.

**Insight:** Pods are ephemeral; their IPs change upon restart, reinforcing the need for stable Service endpoints.

---

## 6. Exposing the Application with a Service

A NodePort Service was created to map port 8080 on every cluster node to the Pods selected by label app=kubernetes-bootcamp. kubectl describe service displayed a cluster-internal IP (10.96.x.x) and a high-port NodePort (3xxxx range). Repeated curl requests against http:// NODE\_PORT returned responses from the same Pod IP, confirming kube-proxy's round-robin behaviour.

A custom label `version=v1` was applied to the running Pod. Querying with `kubectl get pods -l version=v1` verified label filtering. Deleting the Service removed the NodePort mapping; subsequent curl calls failed, yet the Pod remained alive, illustrating that Service lifecycle is independent of Pod lifecycle.

---

## 7. Scaling the Deployment

The Deployment was scaled from 1 to 4 replicas with `kubectl scale deployments/kubernetes-bootcamp --replicas=4`. `kubectl get rs` showed the ReplicaSet desiring 4 Pods; within 20 seconds all Pods were Running and spread across the single minikube node via the default kube-scheduler. The existing Service immediately began load-balancing; five consecutive curl requests returned five distinct Pod hostnames, proving even traffic distribution. Scaling back to 2 replicas terminated the extra Pods gracefully. The ReplicaSet desired count dropped to 2 without service interruption; the remaining Pods continued to answer requests.

**Lesson:** Horizontal scaling is declarative and reversible, with the Service providing stable ingress throughout.

---

## 8. Rolling Update and Rollback

The container image was advanced from v1 to v2 using `kubectl set image`. `kubectl rollout status` reported “deployment updated successfully” after roughly 30 seconds. During the rollout, `kubectl get pods` showed new v2 Pods

being created while v1 Pods were terminated one at a time, maintaining at least 75 % availability (default maxUnavailable=1). Post-update curl returned the v2 payload string, confirming the new code was active.

A deliberate bad update to a non-existent v10 image was attempted next. The Deployment waited according to the default progressDeadlineSeconds, then reported ProgressDeadlineExceeded. Pods showed ImagePullBackOff. A kubectl rollout undo reverted the image to v2; the old ReplicaSet was reactivated, and traffic shifted back to healthy Pods within 25 seconds. No manual Pod intervention was required.

**Conclusion:** Rolling updates provide version history and instantaneous rollback, fulfilling continuous-delivery requirements without external tooling.

---

## 9. General Observations

- **Resilience:** Node failure was simulated by deleting a Pod; the ReplicaSet recreated it instantly.
  - **Abstraction:** Users interact with Deployments and Services, never directly with Pods.
  - **Labels:** Every selection operation—whether by Service, ReplicaSet, or CLI query—relies on consistent label hygiene.
  - **Resource Footprint:** The entire lab consumed under 2 GB RAM and 1 vCPU, demonstrating Kubernetes' efficiency for development workloads.
- 

## 10. Cleanup and Reflection

All objects were deleted in reverse dependency order: Service first, then Deployment, finally the minikube cluster. The dashboard addon was disabled to free local resources.

Completing the six tutorials provided a contiguous narrative: package an application, deploy it, make it discoverable, scale it, and evolve it without downtime. The abstractions encountered—Pod, ReplicaSet, Deployment, Service—form the minimal vocabulary required to operate production-grade workloads on any conformant Kubernetes cluster.

Kubernetes

default

搜索

Workloads

There is nothing to display here  
你可以 部署一个容器化应用, 选择其他命名空间, 或者 阅读 Dashboard 说明 了解更多。

工作负载 (N)

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

服务

- Ingresses (N)
- Ingress Classes
- Services (N)

配置和存储

- Config Maps (N)
- Persistent Volume Claims (N)
- Secrets (N)
- Storage Classes

集群

- Cluster Role Bindings
- Cluster Roles
- 事件 (N)
- 命名空间
- 网络策略 (N)
- Nodes
- Persistent Volumes
- Role Bindings (N)
- Roles (N)
- 服务账号 (N)

自定义资源

设置

关于

```
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 deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-node    0/1     0           0           7s
PS C:\WINDOWS\system32> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-node    1/1     1           1           51s
PS C:\WINDOWS\system32> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
hello-node-6c9b5f4b59-rj7rt   1/1   Running   0          60s
PS C:\WINDOWS\system32> kubectl get events
LAST SEEN   TYPE      REASON          OBJECT          MESSAGE
67s        Normal    Scheduled       pod/hello-node-6c9b5f4b59-rj7rt   Successfully assigned default/hello-node-6c9b5f4b59-rj7rt to minikube
67s        Normal    Pulling        pod/hello-node-6c9b5f4b59-rj7rt   Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.53"
66s        Normal    Pulled        pod/hello-node-6c9b5f4b59-rj7rt   Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.53" in
10.786s (10.786s including waiting). Image size: 54468442 bytes.
66s        Normal    Created       pod/hello-node-6c9b5f4b59-rj7rt   Created container agnhost
66s        Normal    Started       pod/hello-node-6c9b5f4b59-rj7rt   Started container agnhost
67s        Normal    SuccessfulCreate replicaset/hello-node-6c9b5f4b59   Created pod: hello-node-6c9b5f4b59-rj7rt
67s        Normal    ScalingReplicaSet deployment/hello-node   Scaled up replica set hello-node-6c9b5f4b59 from 0 to 1
7m41s      Normal    Starting      node/minikube
7m41s      Normal    NodeAllocatableEnforced node/minikube
7m41s      Normal    NodeHasSufficientMemory node/minikube
7m41s      Normal    NodeHasNoDiskPressure node/minikube
7m41s      Normal    NodeHasSufficientPID node/minikube
7m36s      Normal    RegisteredNode node/minikube
7m36s      Normal    NotActive    node/minikube
Updated Node Allocatable limit across pods
Node minikube status is now: NodeHasSufficientMemory
Node minikube status is now: NodeHasNoDiskPressure
Node minikube status is now: NodeHasSufficientPID
Node minikube event: Registered Node minikube in Controller
```

```
PS C:\WINDOWS\system32> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-node    1/1     1           1           5m42s
kubernetes-bootcamp 1/1     1           1           35s
```

```
{
  "major": "1",
  "minor": "34",
  "emulationMajor": "1",
  "emulationMinor": "34",
  "minCompatibilityMajor": "1",
  "minCompatibilityMinor": "33",
  "gitVersion": "v1.34.0",
  "gitCommit": "f28b4c9efbca5c5c0af716d9f2d5702667ee8a45",
  "gitTreeState": "clean",
  "buildDate": "2025-08-27T10:09:04Z",
  "goVersion": "go1.24.6",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

```
PS C:\WINDOWS\system32> kubectl describe pods
Name:           hello-node-6c9b5f4b59-rj7rt
Namespace:      default
Priority:       0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Wed, 10 Dec 2025 15:02:21 +0800
Labels:         app=hello-node
                pod-template-hash=6c9b5f4b59
Annotations:   <none>
Status:        Running
IP:            10.244.0.7
IPs:
  IP:          10.244.0.7
Controlled By: ReplicaSet/hello-node-6c9b5f4b59
Containers:
  agnhost:
    Container ID:  containerd://883c54597b3bf4b0b4355ad4e8fd98370bc64aa35886aa8fd9617e5caf3dec67
    Image:         registry.k8s.io/e2e-test-images/agnhost:2.53
    Image ID:     registry.k8s.io/e2e-test-images/agnhost@sha256:99c6b4bb4a1e1df3f0b3752168c89358794d02258ebabc26bf21c29399011a85
    Port:          <none>
    Host Port:    <none>
    Command:
      /agnhost
      netexec
      --http-port=8080
    State:        Running
      Started:    Wed, 10 Dec 2025 15:02:32 +0800
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-p2djq (ro)

```

```
Conditions:
  Type        Status
  PodReadyToStartContainers  True
  Initialized  True
  Ready        True
  ContainersReady  True
  PodScheduled  True
Volumes:
  kube-api-access-p2djq:
    Type:           Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:   kube-root-ca.crt
    ConfigMapOptional: (nil)
    DownwardAPI:    true
QoS Class:      BestEffort
Node-Selectors:  <none>
Tolerations:    node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type  Reason  Age From          Message
  ----  -----  --  --  -----
  Normal Scheduled 10m default-scheduler  Successfully assigned default/hello-node-6c9b5f4b59-rj7rt to minikube
  Normal Pulling  10m kubelet        Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.53"
  Normal Pulled   10m kubelet        Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.53" in 10.786s (10.786s including waiting)
  . Image size: 54468442 bytes.
  Normal Created  10m kubelet        Created container: agnhost
  Normal Started  10m kubelet        Started container agnhost

```

```

Name: kubernetes-bootcamp-658f6cbd58-pmcby
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Wed, 10 Dec 2025 15:07:28 +0800
Labels: app=kubernetes-bootcamp
        pod-template-hash=658f6cbd58
Annotations: <none>
Status: Running
IP: 10.244.0.8
IPs:
    IP: 10.244.0.8
Controlled By: ReplicaSet/kubernetes-bootcamp-658f6cbd58

Containers:
  kubernetes-bootcamp:
    Container ID: containerd://d7169f9891f3964932780b8909c67c62259f225890245869787ec4d5f4276707
    Image: gcr.io/google-samples/kubernetes-bootcamp:v1
    Image ID: sha256:0d6b8ee63bb57c5f5b6156f446b3bc3b3c143d233037f3a2f00e279c8fcc64af
    Port: <none>
    Host Port: <none>
    State: Running
      Started: Wed, 10 Dec 2025 15:07:46 +0800
    Ready: True
    Restart Count: 0
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-dp5jv (ro)

Conditions:
  Type Status
  PodReadyToStartContainers True
  Initialized True
  Ready True
  ContainersReady True
  PodScheduled True

Conditions:
  Type Status
  PodReadyToStartContainers True
  Initialized True
  Ready True
  ContainersReady True
  PodScheduled True

Volumes:
  kube-api-access-dp5jv:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
  QoS Class: BestEffort
  Node-Selectors: <none>
  Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type Reason Age From Message
  ---- -- -- -- --
  Normal Scheduled 5m6s default-scheduler Successfully assigned default/kubernetes-bootcamp-658f6cbd58-pmcby to minikube
  Normal Pulling 5m6s kubelet Pulling image "gcr.io/google-samples/kubernetes-bootcamp:v1"
  Normal Pulled 5m33s kubelet Successfully pulled image "gcr.io/google-samples/kubernetes-bootcamp:v1" in 18.14s (18.14s including waiting).
  Normal Created 5m33s kubelet Created container: kubernetes-bootcamp
  Normal Started 5m33s kubelet Started container kubernetes-bootcamp

```