

# 任务六：滚动更新实验报告

## 一 实验准备

### 1.1 修改 product Deployment 配置

在 `gomall/k8s/microservices/product-deployment.yaml` 中添加：

#### 健康检查配置

```
livenessProbe:  
  tcpSocket:  
    port: 8881  
  initialDelaySeconds: 10 # 容器启动后10秒开始检查  
  periodSeconds: 10      # 每10秒检查一次  
  timeoutSeconds: 5      # 检查超时时间5秒  
  failureThreshold: 3    # 连续失败3次判定为失败  
  
readinessProbe:  
  tcpSocket:  
    port: 8881  
  initialDelaySeconds: 5 # 容器启动后5秒开始检查  
  periodSeconds: 5      # 每5秒检查一次  
  timeoutSeconds: 3      # 检查超时时间3秒  
  failureThreshold: 3    # 连续失败3次判定为失败
```

说明：

- **Liveness Probe ( 存活探针 )** : 检查容器是否存活，失败则重启容器
- **Readiness Probe ( 就绪探针 )** : 检查容器是否就绪，失败则从 Service 的负载均衡中移除
- 使用 **TCP Socket** 方式检查 gRPC 端口 8881 是否可连接

#### 滚动更新策略

```
strategy:  
  type: RollingUpdate  
  rollingUpdate:  
    maxUnavailable: 1 # 更新过程中最多1个 Pod 不可用  
    maxSurge: 1       # 更新过程中最多新增1个 Pod
```

说明：

- **maxUnavailable: 1** : 确保至少有 1 个 Pod 提供服务
- **maxSurge: 1** : 控制资源使用，避免同时创建过多 Pod

#### 副本数调整

```
replicas: 2
```

## 1.2 应用配置

```
kubectl apply -f gomall\k8s\microservices\product-deployment.yaml
```

## 二、实验步骤与结果

### 步骤 1：查看更新前状态

命令：

```
kubectl get deployment product -n gomall  
kubectl get pods -n gomall -l app=product
```

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl get pods -n gomall -l app=product  
NAME           READY   STATUS    RESTARTS   AGE  
product-88449446f-nxl8j   1/1     Running   0          6m7s  
product-88449446f-pc4xt   1/1     Running   0          13m
```

说明：

- 当前有 2 个 Pod 运行正常
- 使用的镜像版本：`buwandocker/product:lab2`
- 所有 Pod 状态为 Running, READY 为 1/1

---

### 步骤 2：更新到故障镜像

命令：

```
kubectl set image deployment/product product=buwandocker/product:lab2-unhealthy -n gomall
```

说明：

- 故障镜像 `lab2-unhealthy` 不会监听 gRPC 端口 8881
- 这将导致健康检查失败

---

### 步骤 3：观察滚动更新过程

命令

```
kubectl get pods -n gomall -l app=product
```

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl get pods -n gomall -l app=product
NAME           READY   STATUS    RESTARTS   AGE
product-56d55f84c6-5drwb   0/1     Running   0          12s
product-56d55f84c6-7kgrd   0/1     Running   0          12s
product-88449446f-pc4xt   1/1     Running   0          14m
```

分析：

- 创建了 2 个新 Pod ( 故障镜像 )
- 新 Pod 状态为 **0/1**，表示健康检查失败
- 保留了 1 个旧 Pod ( 正常镜像 )，继续提供服务
- 滚动更新被阻止**，因为新 Pod 健康检查未通过

#### 步骤 4：查看健康检查失败详情

命令：

```
kubectl describe pod product-56d55f84c6-5drwb -n gomall
```

关键信息：

```
Warning Unhealthy 8s (x4 over 68s) kubelet      Liveness probe failed:
dial tcp 10.244.0.92:8881: connect: connection refused
Warning Unhealthy 1s (x15 over 72s) kubelet      Readiness probe
failed: dial tcp 10.244.0.92:8881: connect: connection refused
```

Events:				
Type	Reason	Age	From	Message
Normal	Scheduled	78s	default-scheduler	Successfully assigned gomall/product-56d55f84c6-5drwb to minikube
Normal	Killing	48s	kubelet	Container product failed liveness probe, will be restarted
Normal	Pulled	18s (x2 over 78s)	kubelet	Container image "buwandocker/product:lab2-unhealthy" already present on machine
Normal	Created	18s (x2 over 78s)	kubelet	Created container: product
Normal	Started	18s (x2 over 78s)	kubelet	Started container product
Warning	Unhealthy	8s (x4 over 68s)	kubelet	Liveness probe failed: dial tcp 10.244.0.92:8881: connect: connection refused
Warning	Unhealthy	1s (x15 over 72s)	kubelet	Readiness probe failed: dial tcp 10.244.0.92:8881: connect: connection refused

说明：

- Liveness probe failed**：存活探针失败，连接被拒绝
- Readiness probe failed**：就绪探针失败，Pod 不会接收流量
- 原因：故障镜像未监听 8881 端口

#### 步骤 5：查看 Deployment 更新状态

命令：

```
kubectl get deployment product -n gomall
kubectl rollout status deployment/product -n gomall --timeout=5s
```

NAME	READY	UP-TO-DATE	AVAILABLE
product	1/2	2	1

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl rollout status deployment/product -n gomall
Waiting for deployment "product" rollout to finish: 1 old replicas are pending termination...
```

分析：

- **READY: 1/2**：只有 1 个 Pod 是健康的（旧版本）
- **AVAILABLE: 1**：只有 1 个 Pod 可用
- **更新被阻止**：因为新 Pod 健康检查失败，旧 Pod 无法终止

步骤 6：查看所有 Pod 详细状态

命令：

```
kubectl get pods -n gomall -l app=product -o wide
```

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl get pods -n gomall -l app=product -o wide
NAME          READY   STATUS    RESTARTS   AGE      IP           NODE   NOMINATED NODE   READINESS GATES
product-56d55f84c6-5drwb  0/1     Running   3 (50s ago)  3m51s  10.244.0.92  minikube  <none>        <none>
product-56d55f84c6-7kgrd  0/1     Running   3 (50s ago)  3m51s  10.244.0.91  minikube  <none>        <none>
product-88449446f-pc4xt   1/1     Running   0            18m     10.244.0.83  minikube  <none>        <none>
```

说明：

- 新 Pod（故障版本）：0/1 · 健康检查失败
- 旧 Pod（正常版本）：1/1 · 继续服务
- **服务未中断**：至少有 1 个健康的 Pod 提供服务

步骤 7：执行回滚操作

命令：

```
kubectl rollout undo deployment/product -n gomall
```

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl rollout undo deployment/product -n gomall
deployment.apps/product rolled back
```

## 步骤 8：观察回滚过程

命令（回滚5秒后）：

```
kubectl get pods -n gomall -l app=product
```

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl get pods -n gomall -l app=product
NAME             READY   STATUS      RESTARTS   AGE
product-56d55f84c6-5drwb  0/1    Terminating   4 (33s ago)  4m34s
product-56d55f84c6-7kgrd  0/1    Terminating   4 (33s ago)  4m34s
product-88449446f-h9mnr   1/1    Running     0          13s
product-88449446f-pc4xt   1/1    Running     0          18m
```

分析：

- 故障 Pod 正在终止 ( Terminating )
  - 新的健康 Pod 已创建并运行 ( 正常版本 )
  - 回滚过程正在进行
- 

## 步骤 9：等待回滚完成

命令：

```
kubectl rollout status deployment/product -n gomall
```

结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl rollout status deployment/product -n gomall
deployment "product" successfully rolled out
```

---

## 步骤 10：验证回滚后的最终状态

命令：

```
kubectl get pods -n gomall -l app=product
kubectl get deployment product -n gomall -o
jsonpath='{.spec.template.spec.containers[0].image}'
```

## 结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl get pods -n gomall -l app=product
NAME          READY   STATUS    RESTARTS   AGE
product-88449446f-h9mnr  1/1     Running   0          62s
product-88449446f-pc4xt  1/1     Running   0          19m

C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl get deployment product -n gomall -o jsonpath='{.spec.template.spec.containers[0].image}'
'buwandocker/product:lab2'
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>[]
```

## 验证结果：

- 2 个 Pod 全部健康运行 ( 2/2)
- 镜像版本已回滚到 buwandocker/product:lab2

## 查看滚动更新历史

### 命令：

```
kubectl rollout history deployment/product -n gomall
```

## 结果：

```
C:\Users\slxie\Desktop\cloud-native\cloud-lab2>kubectl rollout history deployment/product -n gomall
deployment.apps/product
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
8          kubectl.exe set image deployment/product product=buwandocker/product:lab2-unhealthy --namespace=gomall --record=true
9          <none>
```

## 实验结论

本实验成功验证了 Kubernetes 滚动更新的以下特性：

1. **零停机更新**：通过健康检查和滚动更新策略，确保服务不中断
2. **自动故障检测**：健康检查失败时，自动阻止更新继续进行
3. **快速回滚**：发现问题后，可以使用一条命令快速回滚
4. **资源优化**：通过 maxUnavailable 和 maxSurge 控制更新过程中的资源使用