Lab2 Kubernetes实践报告

小组分工：

> 普昕：任务一二三
>
> 谢升楼：任务四五
>
> 肖羽平：任务六

1.使用minikube搭建kubernetes集群

运行 `minikube start --driver=docker`

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-66bc5c9577-6d2x8           1/1     Running   0          9m8s
kube-system   coredns-66bc5c9577-mggjv           1/1     Running   0          9m8s
kube-system   etcd-minikube                      1/1     Running   0          9m13s
kube-system   kube-apiserver-minikube            1/1     Running   0          9m13s
kube-system   kube-controller-manager-minikube   1/1     Running   0          9m15s
kube-system   kube-proxy-6f5nf                   1/1     Running   0          9m8s
kube-system   kube-scheduler-minikube            1/1     Running   0          9m13s
kube-system   storage-provisioner                1/1     Running   0          9m11s
```

2. 在Kubernetes集群中部署中间件

在gomall/k8s/middlewares中添加 Kubernetes manifests，包括 Redis、NATS 的 Deployment/Service， MySQL 的 ConfigMap、PV/PVC、StatefulSet、Service：

- nats-deployment.yaml -> NATS Deployment
- nats-service.yaml -> NATS Service
- mysqLconfigmap.yaml -> MySQL初始化 SQL
- mysqLpvpvc.yaml -> PersistentVolume 与PersistentVolumeClaim (hostPath)
- mysql-headless-service.yaml -> MySQL headless Service (StatefulSet 使用)
- mysql-service.yaml -> MySQL ClusterIP service
- mysql-statefulset.yaml -> MySQL StatefulSet(挂载 PVC 并使用 ConfigMap 初始化)

在集群中应用这些文件

```
minikube kubectl -- apply -f gomall/k8s/middlewares
```

```
PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- apply -f gomall/k8s/middlewares ; minikube kubectl -- get pods -o wide ; minikube kubectl -- g
et statefulset,deploy,svc,pv,pvc
configmap/mysql-init-sql created
nats-dbfbf4fdb-psmwr     0/1     ContainerCreating   0          1s    <none>   minikube   <none>           <none>
redis-7699f47487-xh8cx   0/1     ContainerCreating   0          1s    <none>   minikube   <none>           <none>
NAME                     READY   AGE
statefulset.apps/mysql   0/1     1s

NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nats    0/1     1            0           1s
deployment.apps/redis   0/1     1            0           1s

NAME                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)             AGE
service/kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP             15m
service/mysql           ClusterIP   10.106.170.172   <none>        3306/TCP            1s
service/mysql-headless  ClusterIP   None             <none>        3306/TCP            1s
service/nats            ClusterIP   10.110.158.2     <none>        4222/TCP,8222/TCP   1s
service/redis           ClusterIP   10.107.167.78    <none>        6379/TCP            1s

NAME                        CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM               STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
persistentvolume/mysql-pv   5Gi        RWO            Retain           Bound    default/mysql-pvc   manual         <unset>                          1s

NAME                                STATUS   VOLUME     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/mysql-pvc     Bound    mysql-pv   5Gi        RWO            manual         <unset>                 1s
```

运行 `minikube kubectl -- get po -A`，查看pod都处于Running状态，说明三个中间件的容器已经成功启动。

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
default       mysql-0                            1/1     Running   1 (40m ago)   3h36m
default       nats-dbfbf4fdb-psmwr               1/1     Running   1 (40m ago)   3h36m
default       redis-7699f47487-xh8cx             1/1     Running   1 (40m ago)   3h36m
kube-system   coredns-66bc5c9577-6d2x8           1/1     Running   1 (40m ago)   3h51m
kube-system   coredns-66bc5c9577-mggjv           1/1     Running   1 (40m ago)   3h51m
kube-system   etcd-minikube                      1/1     Running   1 (40m ago)   3h51m
kube-system   kube-apiserver-minikube            1/1     Running   1 (40m ago)   3h51m
kube-system   kube-controller-manager-minikube   1/1     Running   1 (40m ago)   3h51m
kube-system   kube-proxy-6f5nf                   1/1     Running   1 (40m ago)   3h51m
kube-system   kube-scheduler-minikube            1/1     Running   1 (40m ago)   3h51m
kube-system   storage-provisioner                1/1     Running   1 (40m ago)   3h51m
```

运行 `minikube kubectl -- get svc -n default`，检查 `mysql`, `nats`, 和 `redis`已经创建了 `ClusterIP` 类型的 `Service`，说明它们可以在集群内部被访问。

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- get svc -n default
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)             AGE
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP             3h55m
mysql            ClusterIP   10.106.170.172   <none>        3306/TCP            3h40m
mysql-headless   ClusterIP   None             <none>        3306/TCP            3h40m
nats             ClusterIP   10.110.158.2     <none>        4222/TCP,8222/TCP   3h40m
redis            ClusterIP   10.107.167.78    <none>        6379/TCP            3h40m
```

运行 `minikube kubectl -- get cm -n default`，已经创建了 `mysql-init-sql`。

运行 `minikube kubectl -- describe statefulset mysql -n default` 的输出进一步确认了这个 `ConfigMap` 被正确挂载到了容器的 `/docker-entrypoint-initdb.d/init.sql` 路径，用于初始化。

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- get cm -n default
NAME               DATA   AGE
kube-root-ca.crt   1      3h55m
mysql-init-sql     1      3h40m
```

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- describe statefulset mysql -n default
Name:               mysql
Namespace:          default
CreationTimestamp:  Wed, 26 Nov 2025 13:40:29 +0800
Selector:           app=mysql
Labels:             app=mysql
Annotations:        <none>
Replicas:           1 desired | 1 total
Update Strategy:    RollingUpdate
Pods Status:        1 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=mysql
  Containers:
   mysql:
    Image:      mysql:8.0
    Port:       3306/TCP
    Host Port:  0/TCP
    Environment:
      MYSQL_ROOT_PASSWORD:  root123
      MYSQL_DATABASE:       gomall
      MYSQL_USER:           gomall
      MYSQL_PASSWORD:       gomall123
    Mounts:
      /docker-entrypoint-initdb.d/init.sql from mysql-init-sql (rw,path="init.sql")
      /var/lib/mysql from mysql-data (rw)
  Volumes:
   mysql-init-sql:
    Type:      ConfigMap (a volume populated by a ConfigMap)
    Name:      mysql-init-sql
    Optional:  false
   mysql-data:
    Type:          PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:     mysql-pvc
    ReadOnly:      false
  Node-Selectors:  <none>
  Tolerations:     <none>
Volume Claims:     <none>
Events:
  Type    Reason            Age    From                    Message
  ----    ------            ----   ----                    -------
  Normal  SuccessfulCreate  3h40m  statefulset-controller  create Pod mysql-0 in StatefulSet mysql successful
```

综上，MySQL 已经实现了数据持久化。get pvc 显示已经创建了mysql-pvc 的 PersistentVolumeClaim，它的状态是 Bound，说明它成功绑定到了一个名为 `mysql-pv` 的 PersistentVolume。describe statefulset 确认了这个 mysql-pvc 被用作 mysql-data 卷，并挂载到了容器的 `/var/lib/mysql` 目录。

3. 在Kubernetes集群中部署gomall

总体思路

为每个微服务创建 `ConfigMap`（配置）、`Deployment`、`Service`、以及必要的`Volume`/`PVC`，使得服务在集群内互相发现并稳定运行。

要点

- 微服务通过 `<中间件的服务名>:<端口号>` 访问中间件。在 `ConfigMap` 中把中间件地址设置为服务名：
  - MySQL: `address: mysql`、`port: 3306`
  - Redis: `address: "redis:6379"`
  - NATS: `url: "nats://nats:4222"`
- 使用 `ConfigMap` 存储微服务配置并挂载到容器中。每个微服务 YAML 文件顶部定义了一个 `ConfigMap`，并在 `Deployment.spec.template.spec.volumes` / `volumeMounts` 中以 `subPath` 的方式挂载到 conf.yaml。这样能在不重建镜像的情况下更新配置。

在集群中应用微服务清单

```
minikube kubectl -- apply -f gomall\k8s\microservices
```

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- apply -f gomall/k8s/microservices
configmap/cart-conf created
deployment.apps/cart created
service/cart created
configmap/checkout-conf created
deployment.apps/checkout created
service/checkout created
configmap/email-conf created
deployment.apps/email created
service/email created
configmap/frontend-conf created
deployment.apps/frontend created
service/frontend created
configmap/order-conf created
deployment.apps/order created
service/order created
configmap/payment-conf created
deployment.apps/payment created
service/payment created
configmap/product-conf created
deployment.apps/product created
service/product created
configmap/user-conf created
deployment.apps/user created
service/user created
```

运行 `minikube kubectl -- get pods -o wide`，可能会出现一些pod起不来的情况

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- get pods -o wide
NAME                        READY   STATUS             RESTARTS      AGE     IP            NODE       NOMINATED NODE   READINESS GATES
cart-79f7f79fd9-rcchw       0/1     CrashLoopBackOff   4 (64s ago)   3m23s   10.244.0.14   minikube   <none>           <none>
checkout-6cf865d4bd-h8www   1/1     Running            0             3m23s   10.244.0.13   minikube   <none>           <none>
email-c65b8b4b-9kj9h        1/1     Running            0             3m23s   10.244.0.12   minikube   <none>           <none>
frontend-75d7bd8dbd-8ht8s   1/1     Running            0             3m23s   10.244.0.15   minikube   <none>           <none>
mysql-0                     1/1     Running            1 (77m ago)   4h13m   10.244.0.8    minikube   <none>           <none>
nats-dbfbf4fdb-psmwr        1/1     Running            1 (77m ago)   4h13m   10.244.0.9    minikube   <none>           <none>
order-7985b9787c-8dm62      0/1     Error              4 (55s ago)   3m23s   10.244.0.16   minikube   <none>           <none>
payment-85d74754d-nr86c     0/1     CrashLoopBackOff   3 (15s ago)   3m22s   10.244.0.18   minikube   <none>           <none>
product-56cc57768f-ddgg4    0/1     CrashLoopBackOff   3 (39s ago)   3m22s   10.244.0.17   minikube   <none>           <none>
redis-7699f47487-xh8cx      1/1     Running            1 (77m ago)   4h13m   10.244.0.11   minikube   <none>           <none>
user-7d78cbc4cd-bfb7x       0/1     Error              3 (29s ago)   3m21s   10.244.0.19   minikube   <none>           <none>
```

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- logs cart-79f7f79fd9-rcchw
/app/bin/cart
${Env:dev Kitex:{Service:cart Address::8883 MetricsPort::9993 EnablePprof:false EnableGzip:false EnableAccessLog:false LogLevel:info LogFileName: Lo
gMaxSize:0 LogMaxBackups:0 LogMaxAge:0} MySQL:{DSN:%s:%s@tcp(%s:%d)/%s?charset=utf8mb4&parseTime=True&loc=Local Username:gomall Password:gomall123 A
ddress:mysql Port:3306 Database:cart} Redis:{Address: Username: Password: DB:0} RateLimiter:{Enabled:false Rate:0 BucketSize:0}}

2025/11/26 09:54:53 /gomall/app/cart/biz/dal/mysql/init.go:33
[error] failed to initialize database, got error Error 1044 (42000): Access denied for user 'gomall'@'%' to database 'cart'
panic: Error 1044 (42000): Access denied for user 'gomall'@'%' to database 'cart'

goroutine 1 [running]:
github.com/cloudwego/biz-demo/gomall/app/cart/biz/dal/mysql.Init()
        /gomall/app/cart/biz/dal/mysql/init.go:47 +0x495
github.com/cloudwego/biz-demo/gomall/app/cart/biz/dal.Init(...)
        /gomall/app/cart/biz/dal/init.go:23
main.main()
        /gomall/app/cart/main.go:37 +0x2b
```

运行 `minikube kubectl -- logs cart-79f7f79fd9-rcchw`查看cart 日志，发现MySQL 还没有为 cart 等服务创建数据库并且授予 gomall 对这些数据库的权限。

优化一下

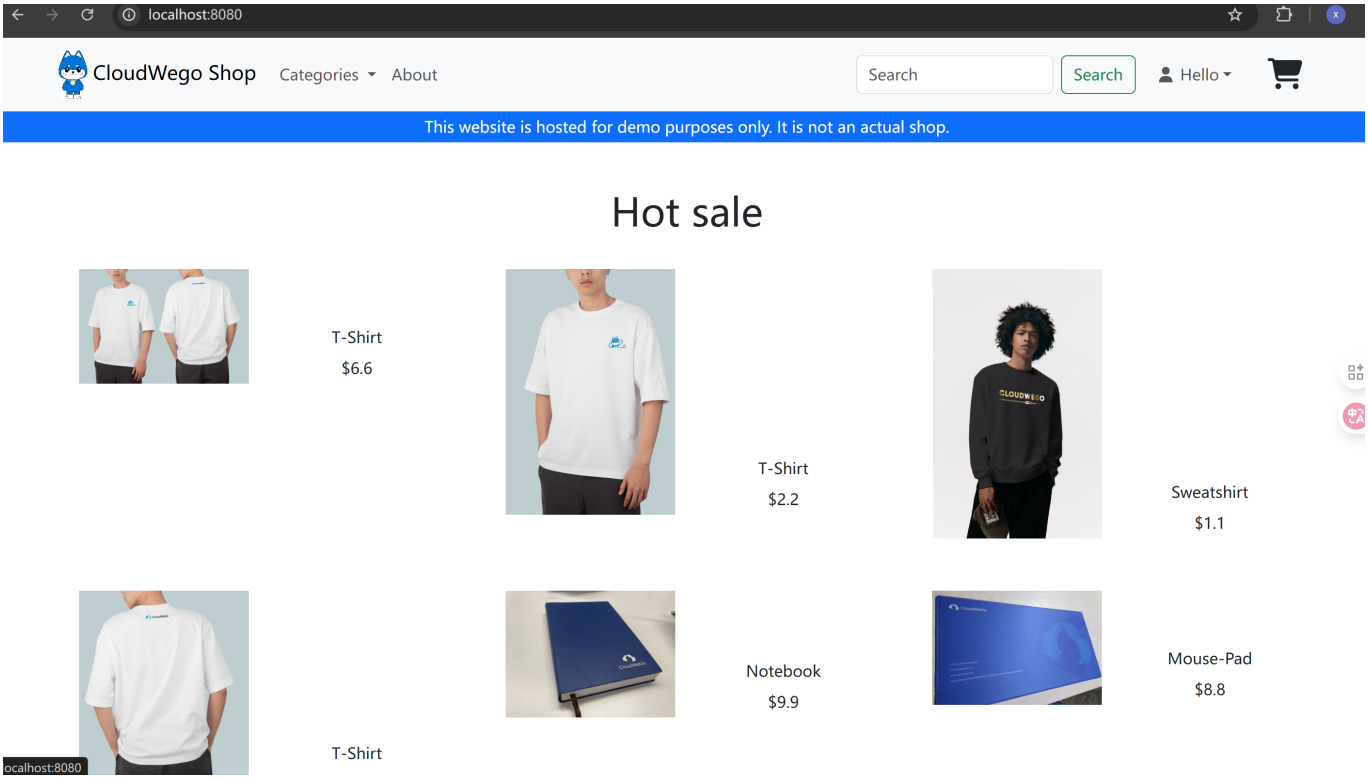
- 添加 mysql-init-job.yaml，等待 MySQL 就绪并执行创建数据库与授权的 SQL。
- 添加 mysql-secret.yaml 保存 MySQL 密码

现在可以了

```
(base) PS E:\Desktop\Junior1\CloudNative\lab\Lab2\cloud-lab2> minikube kubectl -- get pods -A
NAMESPACE     NAME                               READY   STATUS      RESTARTS   AGE
default       cart-675775f99b-9k4l6              1/1     Running     0          7m15s
default       checkout-6cf865d4bd-9mcjf          1/1     Running     0          7m14s
default       email-c65b8b4b-g9qfd               1/1     Running     0          7m14s
default       frontend-75d7bd8dbd-ql8vr          1/1     Running     0          7m14s
default       mysql-0                            1/1     Running     0          7m59s
default       mysql-init-job-559nq               0/1     Completed   0          7m59s
default       nats-dbfbf4fdb-cznhq               1/1     Running     0          7m59s
default       order-849ffc844d-zjmrd             1/1     Running     0          7m14s
default       payment-57c7494997-4q7kg           1/1     Running     0          7m13s
default       product-6f74cd4986-1b7t5           1/1     Running     0          7m12s
default       redis-7699f47487-6qc2x             1/1     Running     0          7m59s
default       user-7d8ddf855f-xpc8q              1/1     Running     0          7m12s
kube-system   coredns-66bc5c9577-n4q96           1/1     Running     0          16m
kube-system   etcd-minikube                      1/1     Running     0          17m
kube-system   kube-apiserver-minikube            1/1     Running     0          17m
kube-system   kube-controller-manager-minikube   1/1     Running     0          17m
kube-system   kube-proxy-rhdrw                   1/1     Running     0          16m
kube-system   kube-scheduler-minikube            1/1     Running     0          17m
kube-system   storage-provisioner                1/1     Running     0          16m
```

运行 `kubectl port-forward service/frontend 8080:8080`

发现可以正常访问在 Kubernetes 集群中运行的 gomall 系统。




```
minikube kubectl -- apply -f .\gomall\k8s\middlewares\mysql-secret.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\mysql-pv-pvc.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\mysql-configmap.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\mysql-headless-service.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\mysql-statefulset.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\redis-deployment.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\redis-service.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\nats-deployment.yaml;
```

```
minikube kubectl -- apply -f .\gomall\k8s\middlewares\nats-service.yaml;
minikube kubectl -- apply -f .\gomall\k8s\middlewares\mysql-init-job.yaml;
```

任务四：扩缩容与负载均衡实验

测试步骤执行记录

1. 初始状态确认

- 初始副本数：1 个 Pod (`product-5454f95b79-7bxj2`)
- 状态：Running

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- get pods -l app=product
NAME                       READY   STATUS    RESTARTS   AGE
product-5454f95b79-7bxj2   1/1     Running   0          56m
```

2. 扩容操作

```
minikube kubectl -- scale deployment/product --replicas=3
```

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- scale deployment/product --replicas=3
deployment.apps/product scaled
```

- 扩容后副本数：3 个 Pod
  - `product-5454f95b79-7bxj2` (原有)
  - `product-5454f95b79-rbdrk` (新增)
  - `product-5454f95b79-wrq6b` (新增)

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- rollout status deployment/product
deployment "product" successfully rolled out
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- get pods -l app=product -o wide
NAME                       READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED NODE   READINESS GATES
product-5454f95b79-7bxj2   1/1     Running   0          16m   10.244.0.15   minikube   <none>           <none>
product-5454f95b79-rbdrk   1/1     Running   0          26s   10.244.0.16   minikube   <none>           <none>
product-5454f95b79-wrq6b   1/1     Running   0          26s   10.244.0.17   minikube   <none>           <none>
```

3. Hey性能测试结果

扩容前

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> hey -c 40 -z 30s http://localhost:8080

Summary:
  Total:        35.7552 secs
  Slowest:      6.4443 secs
  Fastest:      0.0285 secs
  Average:      5.0711 secs
  Requests/sec: 7.2717


Response time histogram:
  0.028 [1]     |
  0.670 [12]    |■■■
  1.312 [6]     |■
  1.953 [3]     |■
  2.595 [5]     |■
  3.236 [4]     |■
  3.878 [5]     |■
  4.520 [4]     |■
  5.161 [7]     |■
  5.803 [191]   |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  6.444 [22]    |■■■■■


Latency distribution:
  10% in 2.5737 secs
  25% in 5.5554 secs
  50% in 5.6887 secs
  75% in 5.7225 secs
  90% in 5.7968 secs
  95% in 5.8190 secs
  99% in 6.4294 secs

Details (average, fastest, slowest):
  DNS+dialup:   0.0113 secs, 0.0285 secs, 6.4443 secs
  DNS-lookup:   0.0111 secs, 0.0000 secs, 0.0743 secs
  req write:    0.0000 secs, 0.0000 secs, 0.0004 secs
  resp wait:    5.0595 secs, 0.0281 secs, 6.4292 secs
  resp read:    0.0001 secs, 0.0001 secs, 0.0007 secs

Status code distribution:
  [200] 260 responses
```

扩容后

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> hey -c 40 -z 30s http://localhost:8080

Summary:
  Total:        30.0888 secs
  Slowest:      0.3913 secs
  Fastest:      0.0055 secs
  Average:      0.0986 secs
  Requests/sec: 405.1012


Response time histogram:
  0.006 [1]     |
  0.044 [1106]  |■■■■
  0.083 [196]   |■
  0.121 [9936]  |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.160 [20]    |
  0.198 [821]   |■■■
  0.237 [98]    |
  0.276 [0]     |
  0.314 [7]     |
  0.353 [0]     |
  0.391 [4]     |


Latency distribution:
  10% in 0.0808 secs
  25% in 0.0939 secs
  50% in 0.0993 secs
  75% in 0.1036 secs
  90% in 0.1148 secs
  95% in 0.1840 secs
  99% in 0.1980 secs

Details (average, fastest, slowest):
  DNS+dialup:   0.0001 secs, 0.0055 secs, 0.3913 secs
  DNS-lookup:   0.0001 secs, 0.0000 secs, 0.0238 secs
  req write:    0.0000 secs, 0.0000 secs, 0.0009 secs
  resp wait:    0.0984 secs, 0.0054 secs, 0.3685 secs
  resp read:    0.0001 secs, 0.0000 secs, 0.0049 secs

Status code distribution:
  [200] 12189 responses
```

4. 负载均衡验证结果

通过查看三个 Pod 的日志，确认所有 Pod 都在处理请求：

Pod 1 (7bxj2): 日志显示大量 `ListProductsService:` 请求

Pod 2 (rbdrk):日志显示大量 `ListProductsService:` 请求

Pod 3 (wrq6b):日志显示大量 `ListProductsService:` 请求

Service 端点验证:

```
Endpoints: 10.244.0.15:8881,10.244.0.17:8881,10.244.0.16:8881
```

三个端点都已正确注册到 Service 中，负载均衡配置生效。

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- logs product-5454f95b79-7bxj2 --tail=20
Defaulted container "product" out of: product, init-db (init)
2025/11/28 08:49:31.069807 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.143212 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.162717 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242825 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242905 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242951 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257692 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257943 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258426 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258665 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258803 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258979 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259057 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259348 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259377 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259576 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259788 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.260376 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.260423 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.343917 list_products.go:36: [Info] ListProductsService:
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- logs product-5454f95b79-rbdrk --tail=20
Defaulted container "product" out of: product, init-db (init)
2025/11/28 08:49:31.243360 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257052 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257097 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257565 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258695 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258863 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259111 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259476 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259513 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259535 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259566 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259887 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.260582 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.260579 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.261675 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.262700 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.343255 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.343752 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.344355 list_products.go:36: [Info] ListProductsService:
```

```
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- logs product-5454f95b79-wrq6b --tail=20
Defaulted container "product" out of: product, init-db (init)
2025/11/28 08:49:31.154154 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.154219 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.160542 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242165 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242416 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242636 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.242691 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.243535 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.243593 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257718 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.257914 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258004 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.258871 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259099 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259177 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259487 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259663 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259824 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.259973 list_products.go:36: [Info] ListProductsService:
2025/11/28 08:49:31.263971 list_products.go:36: [Info] ListProductsService:
PS C:\Users\slxie\Desktop\cloud-native\cloud-lab2> minikube kubectl -- describe svc product
Name:                     product
Namespace:                default
Labels:                   app=product
Annotations:              <none>
Selector:                 app=product
Type:                     ClusterIP
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.111.190.221
```

```
IPs:                      10.111.190.221
Port:                     http  8881/TCP
TargetPort:               8881/TCP
Endpoints:                10.244.0.15:8881,10.244.0.17:8881,10.244.0.16:8881
Session Affinity:         None
Internal Traffic Policy:  Cluster
Events:                   <none>
```

任务五：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

```yaml
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

```yaml
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
```

结果：

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

```
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

6.使用Helm Chart打包部署

使用scoop安装helm

按照任务二的实验过程部署中间件（要求中没有涉及中间件的打包）

将microservices中的每一个yaml文件的内容提取出一个模板写成对应的template

e.g. checkout.yaml将其中需要配置的参数抽取出来，值写在value.yaml中，在template中引用values.yaml的值。

每个资源都需要提取出一个模板（也可以只提取出一个通用的，但可能反而会变得更复杂）

模板提取完成后，将所有的值都分门别类的写入values.yaml.

然后在项目根目录下运行命令 `helm install gomall ./helm`

运行结果：

```
PS C:\Users\xiaoyiyong\Desktop\大学课件\云原生\lab2\gomall> helm install gomall ./gomall-helm
NAME: gomall
LAST DEPLOYED: Fri Nov 28 18:28:15 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
PS C:\Users\xiaoyiyong\Desktop\大学课件\云原生\lab2\gomall> kubectl get pods
NAME                        READY   STATUS    RESTARTS   AGE
cart-fff58774-lgklk         1/1     Running   0          10s
checkout-b695799c9-rtnd8    1/1     Running   0          10s
email-5bd75dff5f-gt222      1/1     Running   0          10s
frontend-5748d9b76c-c7bbq   1/1     Running   0          10s
mysql-0                     1/1     Running   0          25h
nats-dbfbf4fdb-tmdh7        1/1     Running   0          25h
order-7bb55bb44-qzxkh       1/1     Running   0          10s
payment-cb46854bf-qb6gn     1/1     Running   0          10s
product-5b998d958b-wn42d    1/1     Running   0          10s
redis-7699f47487-8dfsd      1/1     Running   0          25h
user-586758644c-99pts       1/1     Running   0          10s
PS C:\Users\xiaoyiyong\Desktop\大学课件\云原生\lab2\gomall>
```

端口转发之后访问前端，服务都正常运行

```
PS C:\Users\xiaoyiyong\Desktop\大学课件\云原生\lab2\gomall> kubectl port-forward service/frontend 8080:8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080
Handling connection for 8080
```