Day 2: æ— çŠ¶æ€•åº”ç”¨ç®¡ç•† (Deployment, R

ðŸŽ¯ å¦ä¹ ç›®æ ‡
- **æŠ€èƒ½ç›®æ ‡**: ç•†è§£å¹¶æŽŒæ•¡ Kubernetes ä¸ç®¡ç•†æ— çŠ¶æ€•åº”ç”¨çš„æ ¸å¿ƒæŽ§åˆ¶å™¨ `Deployment` å’Œ `ReplicaSet`ã€‚
- **æ ¸å¿ƒæ¦‚å¿µ**: æ·±å…¥ç•†è§£å£°æ˜Žå¼• APIã€•æŽ§åˆ¶å™¨æ¨¡å¼•ï¼Œä»¥å•Š `Deployment` å¦‚ä½•å®žçŽ°åº”ç”¨çš„æ»šåŠ¨æ›´æ–°å’Œå›žæ»šã€‚
- **å…·ä½“æˆžæžœ**:èƒ½å¤Ÿç‹¬ç«‹ç¼–å†™ä¸€ä¸ª `Deployment` YAML æ–‡ä»¶æ¥éƒ¨ç½²ã€•æ‰©å®¹å’Œç¼©å®¹ä¸€ä¸ªæ— çŠ¶æ€•åº”ç”¨ï¼ˆå¦‚ Nginxï¼‰ã€‚èƒ½å¤Ÿæ‰§è¡ŒåŠŸèƒ½å¯¹ä¸€ä¸ªå·²éƒ¨ç½²çš„åº”ç”¨æ‰§è¡Œæ»šåŠ¨æ›´æ–°ï¼Œå°†å…¶å‡çº§åˆ°æ–°ç‰ˆæœ¬ã€‚èƒ½å¤ŸæŸ¥çœ‹æ›´æ–°åŽ†å•²ï¼Œå¹¶åœ¨éœ€è¦•æ—¶å°†åº”ç”¨å›žæ»šåˆ°æŒ‡å®šçš„æ—§ç‰ˆæœ¬ã€‚èƒ½å¤Ÿè§£é‡Š `Deployment`, `ReplicaSet`, `Pod` ä¸‰è€…ä¹‹é—´çš„å…³ç³»ã€‚

ðŸ“š ç•†è®ºåŸºç¡€ (30%)

1. å£°æ˜Žå¼• API ä¸ŽæŽ§åˆ¶å™¨æ¨¡å¼•
Kubernetes çš„å·¥ä½œæ¨¡å¼•æ˜¯ **å£°æ˜Žå¼• (Declarative)** çš„ï¼Œè€Œé•žå‘½ä»¤å¼• (Imperative)ã€‚

- **å‘½ä»¤å¼•**: ä½ å‘Šè¯‰ç³»ç»Ÿâ€œå•šä»€ä¹ˆâ€•ï¼Œä¾‹å¦‚ `è¿•è¡Œä¸€ä¸ªå®¹å™¨`ã€•`å•œæ¢é‚£ä¸ªå®¹å™¨`ã€‚
- **å£°æ˜Žå¼•**: ä½ å‘Šè¯‰ç³»ç»Ÿâ€œæˆ‘æƒ³è¦•ä»€ä¹ˆçŠ¶æ€•â€•ï¼Œä¾‹å¦‚ `æˆ‘æƒ³è¦•ä¸€ç›´æœ‰3ä¸ªNginxå®¹å™¨åœ¨è¿•è¡Œ`ã€‚

ä½ é€šè¿‡ YAML æ–‡ä»¶å•‘ API Server å£°æ˜Žä½ çš„â€œææœŸæœ›çŠ¶æ€•â€•ã€‚è€Œ Kubernetes å†…éƒ¨çš„å•„ç§• **æŽ§åˆ¶å™¨ (Controllers)** å°±ä¼šä¸•å•œåœ°å·¥ä½œï¼ŒæŒ•ç»åœ°å°†é›†ç¾¤çš„â€œå½“å‰çŠ¶æ€•â€•èº«æ•´ä¸ºä½ çš„â€œææœŸæœ›çŠ¶æ€•â€•ã€‚è¿™æ£æ˜¯ Kubernetes å¼ºå¤§è‡ªæ„ˆèƒ½åŠ›çš„æ ¹æºã€‚

2. ReplicaSet: å‰¯æœ¬çš„å®ˆæŠ¤è€…
- **è•Œè´£**: `ReplicaSet` çš„å”¯ä¸€è•Œè´£å°±æ˜¯ç¡®ä¿åœ¨ä»»ä½•æ—¶å€™éƒ½æœ‰æŒ‡å®šæ•°é‡çš„ã€•ç¬¦å•ˆç‰¹å®šæ¨¡æ¿çš„ Pod å‰¯æœ¬åœ¨è¿•è¡Œã€‚
- **å·¥ä½œåŽŸç•†**: å®ƒé€šè¿‡ä¸€ä¸ª **æ ‡ç¾é€‰æ‹©å™¨ (Label Selector)** æ•¥è¯†åˆ«å®ƒåº”è¯¥ç®¡ç•†çš„ Podã€‚å¦‚æžœå•‘çŽ°è¿•è¡Œä¸çš„ Pod æ•°é‡å°‘äºŽæœŸæœ›å€¼ï¼Œå®ƒå°±ä¼šæ ¹æ® **Pod æ¨¡æ¿ (Pod Template)** åˆ›å»ºæ–°çš„ Podã€‚å¦‚æžœæ•°é‡å¤šäºŽæœŸæœ›å€¼ï¼Œå®ƒå°±ä¼šéš•æœºåˆ é™¤å¤šä½™çš„ Podã€‚
- **ä½¿ç”¨**: ä½ é€šå¸¸ä¸•ä¼šç›´æŽ¥åˆ›å»º `ReplicaSet`ï¼Œè€Œæ˜¯é€šè¿‡ `Deployment` æ¥é—´æŽ¥ç®¡ç•†å®ƒã€‚

3. Deployment: æ›´é«˜çº§çš„åº”ç”¨ç®¡ç†å™¨

`Deployment` æ˜¯ä¸€ä¸ªæ¯” `ReplicaSet` æ›´é«˜ä¸€å±‚çš„æŽ§åˆ¶å™¨ï¼Œå®ƒæ•´ä¾›äº†æ›´å¤šç®¡ç†åº”ç”¨æ‰€éœ€çš„åŠŸèƒ½ï¼Œæ˜¯éƒ¨ç½²æ— çŠ¶æ€åº”ç”¨çš„é¦–é€‰æ–¹å¼ã€‚

- **æ ¸å¿ƒåŠŸèƒ½**:**ç®¡ç† ReplicaSet å’Œ Pod**: ä½ åˆ›å»ºä¸€ä¸ª Deploymentï¼Œå®ƒä¼šè‡ªåŠ¨ä¸ºä½ åˆ›å»ºä¸€ä¸ª ReplicaSetï¼Œç„¶åŽç”± ReplicaSet æ¥åˆ›å»º Podã€‚**æ»šåŠ¨æ›´æ–° (Rolling Update)**: è¿™æ˜¯ Deployment æœ€æ ¸å¿ƒçš„åŠŸèƒ½ä¹‹ä¸€ã€‚å½“ä½ æ›´æ–°åº”ç”¨çš„é•œåƒæˆ–é…ç½®æ—¶ï¼ŒDeployment ä¼šä»¥ä¸€ç§å¹³æ»‘çš„æ–¹å¼ï¼Œé€æ¥åœ°ç”¨æ–°ç‰ˆæœ¬çš„ Pod æ›¿æ¢æ—§ç‰ˆæœ¬çš„ Podï¼Œä»Žè€Œå®žçŽ°é›¶åœæœºæ›´æ–°ã€‚**ç‰ˆæœ¬å›žæ»š (Rollback)**: Deployment ä¼šè®°å½•ä¸‹æ¯ä¸€æ¬¡æ›´æ–°çš„åŽ†å²ç‰ˆæœ¬ã€‚å¦‚æžœå‘çŽ°æ–°ç‰ˆæœ¬æœ‰é—®é¢˜ï¼Œä½ å¯ä»¥è½»æ¾åœ°å°†åº”ç”¨ä¸€é”®å›žæ»šåˆ°ä¹‹å‰çš„æŸä¸ªç¨³å®šç‰ˆæœ¬ã€‚

4. ä¸‰è€…å…³ç³»

Deployment â†' ReplicaSet â†' Pod

- ä½ å®šä¹‰ä¸€ä¸ª `Deployment`ã€‚
- `Deployment` æ ¹æ®è‡ªå·±çš„å®šä¹‰ï¼Œåˆ›å»ºä¸€ä¸ª `ReplicaSet`ã€‚
- `ReplicaSet` æ ¹æ®è‡ªå·±çš„å®šä¹‰ï¼Œåˆ›å»ºå‡ºæŒ‡å®šæ•°é‡çš„ `Pod`ã€‚
- å½“ä½ æ›´æ–° `Deployment` æ—¶ï¼Œå®ƒä¼šåˆ›å»ºä¸€ä¸ª**æ–°çš„** `ReplicaSet`ï¼Œç„¶åŽé€æ¥åœ°å°† Pod ä»Žæ—§ `ReplicaSet` çš„ç®¡ç†ä¸‹è½¬ç§»åˆ°æ–° `ReplicaSet`ï¼Œä»Žè€Œå®žçŽ°æ»šåŠ¨æ›´æ–°ã€‚æ—§çš„ `ReplicaSet` ä¸•ä¼šè¢«ç«‹å³åˆ é™¤ï¼Œä»¥ä¾¿æ”¯æŒå›žæ»šã€‚

`![Deployment Relationship](https://i.stack.imgur.com/kflbS.png)`

ðŸ› ï¸ å®žè·µæ“•ä½œ (50%)

1. åˆ›å»ºä¸€ä¸ª Deployment

åˆ›å»ºä¸€ä¸ªæ–‡ä»¶ `nginx-deployment.yaml`:

```
apiVersion: apps/v1 # æ³¨æ„è¿™é‡Œçš„ apiVersion æ˜¯ apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3 # å£°æ˜ŽæœŸæœ›çŠ¶æ€ï¼šéœ€è¦ 3 ä¸ªå‰¯æœ¬
  selector:
    matchLabels:
      app: nginx # æ ‡ç¾é€‰æ‹©å™¨ï¼šç®¡ç†é‚£äº›å¸¦æœ‰ app=nginx æ ‡ç¾çš„ Pod
  template: # Pod æ¨¡æ¿ï¼šå¦‚ä½•åˆ›å»º Pod
    metadata:
      labels:
```

```
        app: nginx # Pod çš„æ ‡ç-¾ï¼Œå¿…é¡»ä¸Žä¸Šé•¢çš„ selector åŒ¹é…•
    spec:
      containers:
      - name: nginx
        image: nginx:1.24 # ä½¿ç”¨ 1.24 ç‰ˆæœ¬
        ports:
        - containerPort: 80
```

éƒ¨ç½²å®ƒ:

```
kubectl apply -f nginx-deployment.yaml
```

2. è§å¯Ÿåˆ›å»ºçš„èµ„æº

```
# æŸ¥çœ‹ Deployment çŠ¶æ€•
kubectl get deployment nginx-deployment
# NAME               READY   UP-TO-DATE   AVAILABLE   AGE
# nginx-deployment   3/3     3            3           30s

# æŸ¥çœ‹ ReplicaSetï¼Œæ³¨æ„„å®ƒçš„å••å-—æ˜¯ç”± Deployment å••ç§°åŠ ä¸€ä¸ª hash æž„æˆ•çš„
kubectl get rs
# NAME                          DESIRED   CURRENT   READY   AGE
# nginx-deployment-6b6c47b5b6   3         3         3       45s

# æŸ¥çœ‹ Podsï¼Œæ³¨æ„„å®ƒä»¬éƒ½æ‹¥æœ‰ app=nginx æ ‡ç-¾
kubectl get pods --show-labels
# NAME                                READY   STATUS    RESTARTS   AGE   LABELS
# nginx-deployment-6b6c47b5b6-abcde   1/1     Running   0          60s
app=nginx,pod-template-hash=6b6c47b5b6
# nginx-deployment-6b6c47b5b6-fghij   1/1     Running   0          60s
app=nginx,pod-template-hash=6b6c47b5b6
# nginx-deployment-6b6c47b5b6-klmno   1/1     Running   0          60s
app=nginx,pod-template-hash=6b6c47b5b6
```

3. æ‰©å®¹å’Œç¼©å®¹

```
# ä½¿ç”¨ scale å‘½ä»¤å°†å‰¯æœ¬æ•°æ©±å••å^° 5
kubectl scale deployment nginx-deployment --replicas=5
# deployment.apps/nginx-deployment scaled

# è§‚å¯Ÿ Pod æ•°é‡•å•˜åŒ–
kubectl get pods -l app=nginx # ä½¿ç”¨æ ‡ç-¾é€‰æ‹©å™¨æ•¥æŸ¥çœ‹

# ç¼©å®¹å›ž 2 ä¸ª
kubectl scale deployment nginx-deployment --replicas=2
```

4. æ‰§è¡Œæ»šåŠ¨æ›´æ–°

çŽ°åœ¨ï¼Œæ^‘ä»¬å°† Nginx çš„ç‰ˆæœ¬ä»Ž `1.24` å•‡çº§å^° `1.25`ã€‚

æœ€ç®€å••çš„æ–¹å¼•æ˜¯ä½¿ç”¨ `kubectl set image` å‘½ä»¤ï¼š

```
kubectl set image deployment/nginx-deployment nginx=nginx:1.25
# deployment.apps/nginx-deployment image updated
```

ä½ ä¹Ÿå•¯ä»¥ç›´æŽ¥ä¿®æ”¹ YAML æ–‡ä»¶ä¸çš„ `image` å—æ®µï¼Œç„¶å•Žå†•æ¬¡æ‰§è¡Œ `kubectl apply -f nginx-deployment.yaml`ï¼Œæ•ˆæžœæ˜¯ä¸€æ ·çš„ã€‚

è§‚å¯Ÿæ»šåŠ¨æ›´æ–°çš„è¿‡ç¨‹ï¼š

```
# ä½¿ç”¨ -w å•‚æ•°æŒ•ç»è§‚å¯Ÿ Pod çš„å•˜åŒ–
kubectl get pods -l app=nginx -w
# ä½ ä¼šçœ‹åˆ°æ–°çš„ Pod è¢«åˆ›å»º (terminating æ—§çš„ï¼Œcreating æ–°çš„)

# æŸ¥çœ‹æ›´æ–°çŠ¶æ€•
kubectl rollout status deployment/nginx-deployment
# Waiting for deployment "nginx-deployment" rollout to finish: 2 of 3 updated pods are available...
# deployment "nginx-deployment" successfully rolled out
```

æ›´æ–°å®Œæˆ•å•Žï¼ŒæŸ¥çœ‹ ReplicaSetï¼Œä½ ä¼šå•‘çŽ°å¤šäº†ä¸€ä¸ªæ–°çš„ RSï¼Œè€Œæ—§çš„ RS çš„å‰¯æœ¬æ•°å•˜ä¸ºäº† 0ã€‚

```
kubectl get rs
# NAME                                DESIRED   CURRENT   READY   AGE
# nginx-deployment-6b6c47b5b6   0         0         0       10m   <-- æ—§çš„ RS
# nginx-deployment-7d7c58c6c7   3         3         3       2m    <-- æ–°çš„ RS
```

5. å›žæ»šåº”ç”¨

å•‡è®¾æ–°ç‰ˆæœ¬ `1.25` æœ‰ bugï¼Œæˆ‘ä»¬éœ€è¦•å›žæ»šã€‚

```
# æŸ¥çœ‹æ›´æ–°åŽ†å•²
kubectl rollout history deployment/nginx-deployment
# REVISION  CHANGE-CAUSE
# 1         <none>
# 2         <none>

# æ‰§è¡Œå›žæ»šï¼Œå›žåˆ°ä¸Šä¸€ä¸ªç‰ˆæœ¬ (REVISION 1)
kubectl rollout undo deployment/nginx-deployment
# deployment.apps/nginx-deployment rolled back

# å†•æ¬¡è§‚å¯Ÿ Pod å•˜åŒ–ï¼Œå®ƒä»¬ä¼šå›žåˆ°ä½¿ç”¨ 1.24 é•œåƒ•
kubectl get pods -l app=nginx -w
```

ðŸ’» Go ç¼–ç¨‹å®žçŽ° (20%)

é¡¹ç›®: k8s-deployment-manager

ç›®æ ‡: ç¼–å†™ä¸€ä¸ª Go ç¨‹åº•ï¼Œä½¿ç”¨ `client-go` æ•¥èŽ·å•–æŒ‡å®š

Deployment çš„ä¿¡æ•¯ï¼Œå¹¶æ••ä¾›æ‰©å®¹/ç¼©å®¹çš„åŠŸèƒ½ã€‚

1. åˆ•å§‹åŒ–é¡¹ç›®

```
mkdir k8s-deployment-manager
cd k8s-deployment-manager
go mod init deployment.manager.dev/me
go get k8s.io/client-go@v0.28.2 k8s.io/api@v0.28.2 k8s.io/apimachinery@v0.28.2
```

2. ç¼–å†™ä»£ç (`main.go`)

```
package main

import (
 "context"
 "fmt"
 "log"
 "os"
 "path/filepath"
 "strconv"
 appsv1 "k8s.io/api/apps/v1"
 metav1 "k8s.io/apimachinery/pkg/apis/meta/v1"
 "k8s.io/client-go/kubernetes"
 "k8s.io/client-go/tools/clientcmd"
)

func main() {
 // ç”¨æ³•: go run main.go <namespace> <deployment-name> [replicas]
 if len(os.Args) < 3 {
  fmt.Println("ç”¨æ³•: go run main.go <namespace> <deployment-name>
[replicas]")
  os.Exit(1)
 }
 namespace := os.Args[1]
 deploymentName := os.Args[2]

 // --- é…ç½®åŠ è½½å®¢æˆ·ç«¯ clientset ---
 userHomeDir, _ := os.UserHomeDir()
 kubeconfig := filepath.Join(userHomeDir, ".kube", "config")
 config, _ := clientcmd.BuildConfigFromFlags("", kubeconfig)
 clientset, _ := kubernetes.NewForConfig(config)

 // --- å¦‚果提供了副本数，则更新 Deployment ---
 if len(os.Args) < 4 {
  fmt.Printf("èŽ·å–‐ Deployment '%s' ä¿¡æ¯...\n", deploymentName)
  deployment, err :=
clientset.AppsV1().Deployments(namespace).Get(context.TODO(), deploymentName,
metav1.GetOptions{})
  if err != nil {
   log.Fatal(err)
  }
  fmt.Printf(" - Replicas: %d\n", *deployment.Spec.Replicas)
```

```
  fmt.Printf(" - Image: %s\n",
deployment.Spec.Template.Spec.Containers[0].Image)
  return
 }

 // --- å¦‚果æ••ä¾›äº†å‰¯æœ¬æ•°å•,æ•°ï¼Œå™™æ‰§è¡Œæ‰©/ç¼©å®¹ ---
 replicas, err := strconv.Atoi(os.Args[3])
 if err != nil {
  log.Fatalf("å‰¯æœ¬æ•°å¿…é¡»æ˜¯æ•´æ•°: %v", err)
 }

 fmt.Printf("å°† Deployment '%s' çš„å‰¯æœ¬æ•°è°ƒæ•´ä¸º %d...\n",
deploymentName, replicas)

 // ä½¿ç”¨ Get-Update çš„æ–¹å¼•æ•¥æ›´æ–°å¯¹è±¡
 retryErr := clientcmd.RetryOnConflict(clientcmd.DefaultRetry, func() error {
  // 1. Get: èŽ·å•–æœ€æ–°ç‰ˆæœ¬çš„ Deployment å¯¹è±¡
  deployment, getErr :=
clientset.AppsV1().Deployments(namespace).Get(context.TODO(), deploymentName,
metav1.GetOptions{})
  if getErr != nil {
   return getErr
  }

  // 2. Update: ä¿®æ”¹å‰¯æœ¬æ•°
  *deployment.Spec.Replicas = int32(replicas)

  // 3. Commit: æ••äº¤æ›´æ–°
  _, updateErr :=
clientset.AppsV1().Deployments(namespace).Update(context.TODO(), deployment,
metav1.UpdateOptions{})
  return updateErr
 })

 if retryErr != nil {
  log.Fatalf("æ›´æ–°å¤±è´¥: %v", retryErr)
 }
 fmt.Println("æ›´æ–°æˆ•åŠŸ!")
}
```

3. è¿•è¡Œ

```
# èŽ·å•– nginx-deployment çš„ä¿¡æ•¯
go run main.go default nginx-deployment

# å°†å‰¯æœ¬æ•°è°ƒæ•´ä¸º 5
go run main.go default nginx-deployment 5

# å°†å‰¯æœ¬æ•°è°ƒæ•´ä¸º 1
go run main.go default nginx-deployment 1
```

ðŸ”• æ•…éšœæŽ’æŸ¥ä¸Žä¼˜åŒ–

- **æ»šåŠ¨æ›´æ–°å•¡ä½•**: `kubectl rollout status deployment/<name>` æŸ¥çœ‹çŠ¶æ€ã€`kubectl describe deployment <name>` æŸ¥çœ‹è¯¦ç»†ä¿¡æ¯ã€`kubectl describe rs <new-rs-name>` æŸ¥çœ‹æ–°° ReplicaSet çš„äº‹ä»¶ã€`kubectl describe pod <new-pod-name>` æŸ¥çœ‹æ–°° Pod çš„äº‹ä»¶ã€å¸¸è§•åŽŸå› ï¼šæ–°é•œåƒæ‹‰å–å¤±è´¥ (`ImagePullBackOff`)ã€•å•¥ºæ£€æŸ¥å¤±è´¥å¯¼è‡´ Pod ä¸• Readyã€•èµ„æºä¸•è¶³æ— æ³•å¯åŠ¨æ–° Podã€,
- **ä¼˜åŒ–**: åœ¨ Deployment çš„ `spec.strategy.rollingUpdate` ä¸-å•¯ä»¥è®¾ç½® `maxSurge` å’Œ `maxUnavailable` å•,æ•°æ•¥ç²¾ç»†æŽ§åˆ¶æ»šåŠ¨æ›´æ–°°çš„è¿‡ç¨‹ã€`maxSurge`: æ›´æ–°°è¿‡ç¨‹ä¸-ï¼Œå…•è®¸æ¯”æœŸæœ›å‰¯æœ¬æ•°å¤šå‡ºçš„ Pod æ•°é‡ã€`maxUnavailable`: æ›´æ–°°è¿‡ç¨‹ä¸-ï¼Œå…•è®¸çš„ä¸•å•¯ç”¨ Pod çš„æœ€å¤§æ•°é‡ã€,

ðŸ• è¯¾å•Žä½œä¸š

1. **ç”¨ Deployment æ›´æ–°°ç-–ç•¥**: åˆ›å»ºä¸€ä¸ª Deploymentï¼Œå°†å…¶ `spec.strategy.type` è®¾ç½®ä¸º `Recreate`ã€ç„¶å•Žå°•è¯•æ›´æ–°°é•œåƒ•ï¼Œè§‚å¯Ÿå…¶è¡Œä¸ºä¸Ž `RollingUpdate` æœ‰ä½•ä¸•å•Œã€,
2. **å¸¦å‘½ä»¤çš„æ›´æ–°°åŽ†å•²**: åœ¨æ‰§è¡Œ `kubectl set image` æˆ– `kubectl apply` æ—¶ï¼Œä½¿ç”¨ `--record` æ ‡å¿— (è™½ç„¶å·²å¼ƒç¼ƒä½†ä¸ºäº†è§£å…¶åŠŸèƒ½å•¯ä»¥ä¸€è¯•) æˆ–è€…åœ¨ YAML ä¸-ä½¿ç”¨ `annotations` æ•¥è®°å½•æ¯•æ¡¡å•˜æ›´çš„åŽŸå› ã€ç„¶å•Žæ‰§è¡Œ `kubectl rollout history` æŸ¥çœ‹ `CHANGE-CAUSE` åˆ—ã€,