Day 1: Kubernetes æ ¸å¿ƒæž¶æž„ä¸ŽåŸºæœ¬æ

ðŸŽ¯ å¦ä¹ ç›®æ ‡

- **æŠ€èƒ½ç›®æ ‡**: ç•†è§£å®¹å™¨ç¼–æŽ’çš„å¿…è¦æ€§ï¼ŒæŽŒæ•¡ Kubernetes çš„æ ¸å¿ƒæž¶æž„å’Œç»„ä»¶ä½œç”¨ã€‚
- **æ ¸å¿ƒæ¦‚å¿µ**: æ·±å…¥ç•†è§£ Cluster, Node, Pod è¿™ä¸‰ä¸ªæœ€åŸºæœ¬çš„æ¦‚å¿µã€‚
- **å…·ä½“æˆæžœ**:èƒ½å¤Ÿç‹¬ç«‹æ•å»ºä¸€ä¸ªæœ¬åœ°çš„ Kubernetes æµ‹è¯•çŽ¯å¢ƒ (ä¾‹å¦‚ minikube)ã€‚èƒ½å¤Ÿç†Ÿç»ƒä½¿ç”¨ `kubectl` æ‰§è¡Œé›†ç¾¤ä¿¡æ•¯æŸ¥çœ‹ã€•èŠ‚ç‚¹çŠ¶æ€•æŸ¥è¯¢ç‰åŸºæœ¬å‘½ä»¤ã€‚èƒ½å¤Ÿç‹¬ç«‹ç¼–å†™ä¸€ä¸ªç®€å••çš„ Pod YAML æ–‡ä»¶ï¼Œå¹¶æˆåŠŸéƒ¨ç½²åˆ°é›†ç¾¤ä¸ã€‚èƒ½å¤Ÿä½¿ç”¨ `kubectl` å¯¹è¿è¡Œä¸çš„ Pod è¿›è¡ŒæŸ¥çœ‹ã€•æ£€æŸ¥ã€•äº¤äº’å’Œåˆ é™¤ã€‚

ðŸ“š ç•†è®ºåŸºç¡€ (40%)

1. ä»Žå®¹å™¨åˆ°å®¹å™¨ç¼–æŽ’

æˆ'ä»¬å·²ç»•çŸ¥é•"ï¼Œå®¹å™¨ï¼ˆå¦‚ Dockerï¼‰ä¸ºåº"ç"¨æ••ä¾›äº†ä¸€ä¸ªè½»é‡•çº§ã€•å•¯ç§»æ¤ã€•è‡ªåŒ…å«çš„è¿•è¡ŒçŽ¯å¢ƒã€‚ä½†è¿™å•ªè§£å†³äº†å••ä¸ªåº"ç"¨çš„æ‰"åŒ…å'Œè¿•è¡Œé—®é¢˜ã€‚å½"åº"ç"¨å•˜å¾—å¤•æ•‚ï¼Œç"±å‡ å••ä¸Šç™¾ä¸ªå¾®æœ•åŠ¡æž„æˆ•æ—¶ï¼Œæ–°çš„é—®é¢˜å‡ºçŽ°äº†ï¼š

- **éƒ¨ç½²**: å¦‚ä½•ä¸€æ¬¡æ€§éƒ¨ç½²å'Œç®¡ç•†æˆ•ç™¾ä¸Šå•ƒä¸ªå®¹å™¨ï¼Ÿ
- **ä¼¸ç¼©**: å¦‚ä½•æ ¹æ•®è´Ÿè½½è‡ªåŠ¨å¢žåŠ æˆ–å‡•å°'å®¹å™¨å®žä¾‹ï¼Ÿ
- **æœ•åŠ¡å•'çŽ°**: ä¸€ä¸ªå®¹å™¨å¦‚ä½•æ‰¾åˆ°å¹¶ä¸Žå•¦ä¸€ä¸ªå®¹å™¨é€šä¿¡ï¼Ÿ
- **è‡ªæ„ˆ**: å¦‚æžœä¸€ä¸ªå®¹å™¨æˆ–å®ƒæ‰€åœ¨çš„æœºå™¨å®•æœºäº†ï¼Œå¦‚ä½•è‡ªåŠ¨æ•¢å¤•æœ•åŠ¡ï¼Ÿ
- **å•‡çº§**: å¦‚ä½•åœ¨ä¸•ä¸æ–æœ•åŠ¡çš„æƒ…å†µä¸‹æ›´æ–°åº"ç"¨ç‰ˆæœ¬ï¼Ÿ

å®¹å™¨ç¼–æŽ’ (Container Orchestration)
æ£æ˜¯ä¸ºè§£å†³è¿™äº›é—®é¢˜è€Œç"Ÿã€‚Kubernetes å°±æ˜¯ç›®å‰ä¸šç•Œæœ€ä¸»æµã€•æœ€å¼ºå¤§çš„å®¹å™¨ç¼–æŽ'ç³»ç»Ÿã€‚ä½ å•¯ä»¥æŠŠå®ƒæƒ³è±¡æˆä¸€ä¸ªç®¡ç•†æµ·é‡•å®¹å™¨çš„â€œåˆ†å¸ƒå¼æ"•ä½œç³»ç»Ÿâ€•ã€‚

2. Kubernetes æ ¸å¿ƒæž¶æž„

Kubernetes é›†ç¾¤ç"±ä¸¤ç§•ä¸»è¦•ç±»åž‹çš„èŠ‚ç‚¹ç»„æˆï¼š**æŽ§åˆ¶å¹³é•¢èŠ‚ç‚¹ (Control Plane Nodes)** å'Œ **å·¥ä½œèŠ‚ç‚¹ (Worker Nodes)**ã€‚

`![K8s Architecture](https://d33wubrfki0l68.cloudfront.net/2475489eaf2033f99d5581e45a50668853a73252/a7143/images/docs/components-of-kubernetes.svg)`

a. æŽ§åˆ¶å¹³é•¢ (Control Plane) - é›†ç¾¤çš„å¤§è„'

æŽ§åˆ¶å¹³é•¢è´Ÿè´£å•šå‡ºå…¨å±€å†³ç–ï¼Œä¾‹å¦‚è°ƒåº¦ Podã€•æ£€æµ‹å'Œå"•åº"é›†ç¾¤äº‹ä»¶ç‰ã€‚å®ƒç"±ä»¥ä¸‹å‡ ä¸ªå…³é"®ç»„ä»¶æž„æˆï

¼š

- **`kube-apiserver`**: **é›†ç¾¤çš„ç»Ÿä¸€å…¥å•£**ã€‚å®ƒæš´éœ² Kubernetes APIï¼Œæ˜¯æ‰€æœ‰ç»„ä»¶ï¼ˆåŒ…æ‹¬ `kubectl`ï¼‰ä¸Žé›†ç¾¤çŠ¶æ€äº¤äº’çš„å”¯ä¸€é€”å¾„ã€‚å®ƒè´Ÿè´£å¤„ç† REST è¯·æ±‚ã€•éªŒè¯•è¯·æ±‚ã€•å¹¶æ›´æ–° `etcd` ä¸çš„å¯¹è±¡çŠ¶æ€ã€‚
- **`etcd`**: ä¸€ä¸ªé«˜å¯ç”¨çš„é”®å€¼å¯¹å˜å‚¨ç³»ç»Ÿã€‚**å®ƒä¿•å˜äº†æ•´ä¸ªé›†ç¾¤çš„å®Œæ•´çŠ¶æ€æ•°æ®**ï¼Œæ˜¯é›†ç¾¤çš„å”¯ä¸€â€œäº‹å®žæ¥æº• (Source of Truth)â€•ã€‚æ‰€æœ‰å¯¹é›†ç¾¤çŠ¶æ€çš„æ”¹å˜éƒ½å¿…é¡»é€šè¿‡ `apiserver` å†™å…¥ `etcd`ã€‚
- **`kube-scheduler`**: **Pod çš„è°ƒåº¦å™¨**ã€‚å®ƒç›‘è§†æ–°åˆ›å»ºçš„ã€•ä½†å°šæœªåˆ†é…•åˆ°èŠ‚ç‚¹çš„ Podï¼Œç„¶åŽæ ¹æ•®ä¸€ç³»åˆ—å¤æ•‚çš„è§„åˆ™ï¼ˆå¦‚èµ„æºéœ€æ±‚ã€•äº²å’Œæ€§ã€•ç–ç•¥ç‰ï¼‰ä¸ºå…¶é€‰æ‹©ä¸€ä¸ªæœ€å•ˆé€‚çš„å·¥ä½œèŠ‚ç‚¹ã€‚
- **`kube-controller-manager`**: **é›†ç¾¤çŠ¶æ€çš„ç»´æŠ¤è€…**ã€‚å®ƒè¿è¡Œç•€å¤šä¸ªæŽ§åˆ¶å™¨è¿›ç¨‹ï¼ˆå¦‚èŠ‚ç‚¹æŽ§åˆ¶å™¨ã€•å‰¯æœ¬æŽ§åˆ¶å™¨ç‰ï¼‰ã€‚æ¯•ä¸ªæŽ§åˆ¶å™¨è´Ÿè´£ç›‘è§†ä¸€ç§•ç‰¹å®šèµ„æºçš„çŠ¶æ€ï¼Œå¹¶åŠªåŠ›å°†å½“å‰çŠ¶æ€è°ƒæ•´ä¸ºåœ¨ `etcd` ä¸å®šä¹‰çš„æœŸæœ›çŠ¶æ€ã€‚

b. å·¥ä½œèŠ‚ç‚¹ (Worker Node) - é›†ç¾¤çš„åŠ³åŠ¨åŠ›

å·¥ä½œèŠ‚ç‚¹è´Ÿè´£è¿è¡Œç”¨æˆ·çš„åº”ç”¨ç¨‹åº•ï¼ˆå•³å®¹å™¨ï¼‰ã€‚å®ƒåŒ…å•«ä»¥ä¸‹ç»„ä»¶ï¼š

- **`kubelet`**: **èŠ‚ç‚¹ä¸Šçš„ä»£ç†**ã€‚å®ƒç›´æŽ¥ä¸Žå®¹å™¨è¿è¡Œæ—¶ï¼ˆå¦‚ containerdï¼‰äº¤äº’ï¼Œç¡®ä¿ Pod ä¸æ••è¿°çš„å®¹å™¨éƒ½å¤„äºŽæ£ç¡®åœ°å¯åŠ¨ã€•è¿è¡Œå’Œå•œæ¢ã€‚å®ƒä¹Ÿå®šæ—¶å‘ `apiserver` æ±‡æŠ¥æœ¬èŠ‚ç‚¹çš„çŠ¶æ€ã€‚
- **`kube-proxy`**: **ç½‘ç»œä»£ç†**ã€‚å®ƒè´Ÿè´£ç»´æŠ¤èŠ‚ç‚¹ä¸Šçš„ç½‘ç»œè§„åˆ™ï¼Œå®žçŽ°äº† Kubernetes Service çš„æ¦‚å¿µï¼Œå…•è®¸ç½‘ç»œæµ•é‡•åœ¨ Pod ä¹‹é—´è¿›è¡Œè·¯ç”±å’Œè´Ÿè½½å•‡è¡¡ã€‚
- **`Container Runtime`**: **å®¹å™¨è¿è¡Œæ—¶**ã€‚è¿™æ˜¯çœŸæ£è´Ÿè´£è¿è¡Œå®¹å™¨çš„è½¯ä»¶ï¼Œä¾‹å¦‚ `containerd`, `CRI-O`ï¼Œæˆ–è€…æ—©æœŸçš„ `Docker`ã€‚

3. æ ¸å¿ƒæ¦‚å¿µï¼šCluster, Node, Pod

- **Cluster (é›†ç¾¤)**: ç”±ä¸€ä¸ªæˆ–å¤šä¸ªæŽ§åˆ¶å¹³é•¢èŠ‚ç‚¹å’Œå¤šä¸ªå·¥ä½œèŠ‚ç‚¹ç»„æˆçš„å®Œæ•´ Kubernetes çŽ¯å¢ƒã€‚
- **Node (èŠ‚ç‚¹)**: ä¸€ä¸ªå·¥ä½œæœºå™¨ï¼Œå•¯ä»¥æ˜¯ç‰©ç†æœºæˆ–è™šæ‹Ÿæœºã€‚å®ƒæ˜¯ Pod è¿è¡Œçš„è½½ä½“ã€‚
- **Pod**: **Kubernetes ä¸æœ€å°•ã€•æœ€åŸºæœ¬çš„éƒ¨ç½²å••å…ƒ**ã€‚ä¸€ä¸ª Pod å°•è£…äº†ä¸€ä¸ªæˆ–å¤šä¸ªç´§å¯†å…³è•”çš„å®¹å™¨ã€•å˜å‚¨èµ„æº•ã€•ä»¥å•Šä¸€ä¸ªå”¯ä¸€çš„ç½‘ç»œ IPã€‚Pod å†…çš„å®¹å™¨å…±äº«å•Œä¸€ä¸ªç½‘ç»œå‘½å••ç©ºé—´å’Œå˜å‚¨å·ï¼Œå•¯ä»¥é€šè¿‡ `localhost` ç›¸äº’é€šä¿¡ã€‚

ðŸ› ï¸• å®žè·µæ“•ä½œ (50%)

1. å®‰è£…æœ¬åœ° Kubernetes çŽ¯å¢ƒ (minikube)
Minikube æ˜¯ä¸€ä¸ªå••ä»¥åœ¨æœ¬åœ°å¿«é€Ÿå••åŠ¨ Kubernetes é›†ç¾¤çš„å·¥å…·ï¼Œé•žå¸¸é€‚å•ˆå¦ä¹ ã€‚

```
# æ ¹æ•®ä½ çš„æ“•ä½œç³»ç»Ÿï¼Œè€ƒå®˜æ–‡æ¡£å®‰è£… minikube
# https://minikube.sigs.k8s.io/docs/start/

# å•¯åŠ¨ä¸€ä¸ª minikube é›†ç¾¤
minikube start --driver=docker
```

2. å®‰è£…å¹¶é…•ç½® kubectl
`kubectl` æ˜¯ä¸Z Kubernetes é›†ç¾¤äº¤äº’çš„å‘½ä»¤è¡Œå·¥å…·ã€‚

```
# å•,è€ƒå®˜æ–¹æ¡£å®‰è£… kubectl
# https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/

# minikube start ä¼šè‡ªåŠ¨é…•ç½® kubectl çš„ä¸Šä¸‹æ–‡
# éªŒè¯• kubectl æ˜¯å•¦é…•ç½®æ-£ç¡®
kubectl cluster-info
# è¾“å‡ºå°”æ˜¾ç¤º Master å’Œ CoreDNS çš„åœ°å•€

# æŸ¥çœ‹é›†ç¾¤ä¸-çš„èŠ‚ç‚¹
kubectl get nodes
# NAME       STATUS   ROLES           AGE   VERSION
# minikube   Ready    control-plane   10m   v1.28.3
```

3. åˆ›å»ºä½ çš„ç¬¬ä¸€ä¸ª Pod
å˜›å»ºä¸€ä¸ªæ–‡ä»¶ `my-first-pod.yaml`:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:1.25
    ports:
    - containerPort: 80
```

- `apiVersion`: å®šä¹‰ä°†ä½¿ç”¨å“ªä¸ªç‰ˆæœ¬çš„ K8s API æ•¥å˜›å»è¿™ä¸ªå¯¹è±¡ã€‚
- `kind`: å®šä¹‰è¦•å˜›å»çš„èµ„æº•ç±»åž‹ï¼Œè¿™æ˜¯ä¸€ä¸ª `Pod`ã€‚
- `metadata`: åŒ…å«ä°†å¯¹è±¡çš„å…ƒæ•°æ•®ï¼Œå••ç§° (`name`) å’Œæ ‡ç-¾ (`labels`)ã€‚

- `spec`: å®šä¹‰äº†å¯¹è±¡çš„æœŸæœ›çŠ¶æ€•ï¼Œä¾‹å¦‚ Pod ä¸åº”è¯¥åŒ…å«å“ªäº›å®¹å™¨ã€‚

ä½¿ç”¨ `kubectl` åˆ›å»ºè¿™ä¸ª Pod:

```
kubectl apply -f my-first-pod.yaml
# pod/nginx-pod created
```

4. è§‚å¯Ÿå’Œæ£€æŸ¥ Pod

```
# æŸ¥çœ‹æ‰€æœ‰ Pod çš„åˆ—è¡¨å’ŒåŸºæœ¬çŠ¶æ€•
kubectl get pods
# NAME        READY   STATUS    RESTARTS   AGE
# nginx-pod   1/1     Running   0          30s

# æŸ¥çœ‹æ›´è¯¦ç»†çš„çŠ¶æ€•ï¼ŒåŒ…æ‹¬è¢«åˆ†é…çš„ IP å’Œæ‰€åœ¨èŠ‚ç‚¹
kubectl get pods -o wide

# æŸ¥çœ‹ Pod çš„è¯¦ç»†ä¿¡æ¯ï¼ŒåŒ…æ‹¬äº‹ä»¶å¿—ï¼Œè¿™å¯¹äºŽæŽ’é”™è‡³å…³é‡è¦
kubectl describe pod nginx-pod

# æŸ¥çœ‹ Pod ä¸å®¹å™¨çš„æ ‡å‡†è¾“å‡ºæ—¥å¿—
kubectl logs nginx-pod

# åœ¨è¿è¡Œä¸çš„ Pod å†…æ‰§è¡Œå‘½ä»¤ (ç±»ä¼¼äºŽ docker exec)
kubectl exec -it nginx-pod -- /bin/bash
# root@nginx-pod:/# ls
# root@nginx-pod:/# exit
```

5. åˆ é™¤ Pod

```
kubectl delete -f my-first-pod.yaml
# pod "nginx-pod" deleted

# æˆ–è€…æŒ‰åç§°åˆ é™¤
kubectl delete pod nginx-pod
```

ðŸ’» Go ç¼–ç¨‹å®žçŽ° (10%)

é¡¹ç›®: k8s-cluster-info

ç›®æ ‡: ç¼–å†™ä¸€ä¸ªç®€å•çš„ Go ç¨‹åºï¼Œä½¿ç”¨ `client-go` è¿žæŽ¥åˆ°é›†ç¾¤å¹¶æ‰“å°å‡ºæ‰€æœ‰èŠ‚ç‚¹çš„åç§°å’Œç‰ˆæœ¬ä¿¡æ¯ã€‚

1. åˆå§‹åŒ–é¡¹ç›®

```
mkdir k8s-cluster-info
cd k8s-cluster-info
go mod init cluster.info.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"

 metav1 "k8s.io/apimachinery/pkg/apis/meta/v1"
 "k8s.io/client-go/kubernetes"
 "k8s.io/client-go/tools/clientcmd"
)

func main() {
 // 1. åŠ è½½ kubeconfig æ–‡ä»¶
 userHomeDir, err := os.UserHomeDir()
 if err != nil {
  log.Fatalf("æ— æ³•èž·å•–ç”¨æˆ·å®¶ç›®å½•: %v", err)
 }
 kubeconfigPath := filepath.Join(userHomeDir, ".kube", "config")

 // 2. æž„å»ºé…•ç½®
 config, err := clientcmd.BuildConfigFromFlags("", kubeconfigPath)
 if err != nil {
  log.Fatalf("åŠ è½½ kubeconfig å¤±è´¥: %v", err)
 }

 // 3. åˆ›å»º clientset
 clientset, err := kubernetes.NewForConfig(config)
 if err != nil {
  log.Fatalf("åˆ›å»º clientset å¤±è´¥: %v", err)
 }

 // 4. ä½¿ç”¨ clientset ä¸Ž API Server äº¤äº’
 fmt.Println("--- Kubernetes Nodes ---")
 nodes, err := clientset.CoreV1().Nodes().List(context.TODO(),
metav1.ListOptions{})
 if err != nil {
  log.Fatalf("åˆ—å‡ºèŠ‚ç‚¹å¤±è´¥: %v", err)
 }

 for _, node := range nodes.Items {
  fmt.Printf("- Name: %s\n", node.Name)
  fmt.Printf("  Kubelet Version: %s\n", node.Status.NodeInfo.KubeletVersion)
  fmt.Printf("  OS: %s\n", node.Status.NodeInfo.OperatingSystem)
  fmt.Println("--------------------")
 }
}
```

3. è¿•è¡Œ

```
go run main.go
# --- Kubernetes Nodes ---
# - Name: minikube
#   Kubelet Version: v1.28.3
#   OS: linux
# --------------------
```

ðŸ”• æ•…éšœæŽ’æŸ¥ä¸Žä¼˜åŒ–

- **`kubectl` å‘½ä»¤æ— æ³•è¿žæŽ¥**:æ£€æŸ¥ `~/.kube/config` æ–‡ä»¶æ˜¯å¦å˜åœ¨ä¸”å†…å®¹æ£ç¡®ã€‚è¿•è¡Œ `minikube status` ç¡®ä¿•é›†ç¾¤æ£åœ¨è¿•è¡Œã€‚
- **Pod çŠ¶æ€•ä¸º `Pending`**:`kubectl describe pod <pod-name>` æŸ¥çœ‹äº‹ä»¶ã€‚å¸¸è§•åŽŸå› ï¼šèµ„æº不足ï¼ˆå¦‚å†…å˜æˆ–CPUï¼‰ï¼Œæˆ–æ²¡æœ‰å¯ç”¨èŠ‚ç‚¹ã€‚
- **Pod çŠ¶æ€•ä¸º `ImagePullBackOff` æˆ– `ErrImagePull`**:`kubectl describe pod <pod-name>` æŸ¥çœ‹äº‹ä»¶ã€‚å¸¸è§•åŽŸå› ï¼šé•œåƒ•å••ç§°é”™è¯¯ã€•Tag ä¸•å˜åœ¨ã€•æˆ–æ— æ³•è®¿é—®ç§•æœ‰é•œåƒ•ä»“åº“ã€‚

ðŸ• è¿›å•Žä½œä¸š

1. **ç ”ç©¶ Pod ç”Ÿå‘½å‘¨æœŸ**: é˜…è¯»å®˜æ–¹æ–‡æ¡£ï¼Œè¯¦ç»†äº†è§£ Pod ä»Žåˆ›å»ºåˆ°é”€æ¯ç»•åŽ†çš„å„ä¸ªé˜¶æ®µï¼ˆ`Pending`, `Running`, `Succeeded`, `Failed`, `Unknown`ï¼‰åŠå…¶å«ä¹‰ã€‚
2. **å¤šå®¹å™¨ Pod**: ä¿®æ”¹ `my-first-pod.yaml`ï¼Œåœ¨å•Œä¸€ä¸ª Pod ä¸å¢žåŠ ä¸€ä¸ª `busybox` å®¹å™¨ï¼ˆimage: busybox`ï¼‰ï¼Œè®©å®ƒæ¯•5ç§’æ‰“å•°ä¸€æ¬¡æ—¥æœŸ (`command: ["/bin/sh", "-c", "while true; do date; sleep 5; done"]`ï¼‰ã€‚éƒ¨ç½²å•Žï¼Œä½¿ç”¨ `kubectl logs nginx-pod -c busybox-container` æŸ¥çœ‹ busybox å®¹å™¨çš„æ—¥å¿—ã€‚æ€•è€ƒè¿™ç§•æ¨¡å¼•çš„åº”ç”¨åœºæ™¯ã€‚