Day 3: K8s æœ•åŠ¡ä¸Žç½‘ç»œ (Service)

ðŸŽ¯ å¦ä¹ ç›®æ ‡

- **æŠ€èƒ½ç›®æ ‡**: ç•†è§£å¹¶æŽŒæ•¡ Kubernetes ä¸å®žçŽ°æœ•åŠ¡å•‘çŽ°å’Œè´Ÿè½½å•‡è¡¡çš„æ ¸å¿ƒèµ„æº• `Service`ã€‚
- **æ ¸å¿ƒæ¦‚å¿µ**: æ·±å…¥ç•†è§£ Pod IP çš„é•žæŒ•ä¹…æ€§é—®é¢˜ï¼Œä»¥å•Š `Service` å¦‚ä½•é€šè¿‡ä¸€ä¸ªç¨³å®šçš„è™šæ‹Ÿ IP å’Œ DNS å••ç§°æ•¥è§£å†³è¿™ä¸ªé—®é¢˜ã€‚
- **å…·ä½“æˆ•æžœ**:èƒ½å¤Ÿç‹¬ç«‹ä¸ºä¸€ç»„ Pod åˆ›å»ºä¸€ä¸ª `ClusterIP` ç±»åž‹çš„ `Service`ï¼Œå¹¶å®žçŽ°é›†ç¾¤å†…éƒ¨çš„è®¿é—®ã€‚èƒ½å¤Ÿä½¿ç”¨ `NodePort` ç±»åž‹çš„ `Service`ï¼Œå°†åº”ç”¨ç«¯å•£æš´éœ²åˆ°é›†ç¾¤å¤–éƒ¨è¿›è¡Œè®¿é—®ã€‚èƒ½å¤Ÿè§£é‡Š `Service`ã€•`EndpointSlice` (æˆ– `Endpoints`) å’Œ `Pod` ä¹‹é—´çš„å…³è•”å…³ç³»ã€‚èƒ½å¤Ÿè§£é‡Š K8s å†…éƒ¨çš„ DNS æ˜¯å¦‚ä½•å·¥ä½œçš„ã€‚

ðŸ“š ç•†è®ºåŸºç¡€ (40%)

1. ä¸ºä»€ä¹ˆéœ€è¦• Serviceï¼Ÿ

åœ¨ Day 2 æˆ‘ä»¬å¦ä¹ äº† Deploymentï¼Œå®ƒå•¯ä»¥åŠ¨æ€•åœ°åˆ›å»ºå’Œé”€æ¯• Pod æ•¥ç»´æŒ•æœŸæœ›çš„å‰¯æœ¬æ•°ã€‚è¿™å¸¦æ•¥ä¸€ä¸ªæ–°é—®é¢˜ï¼š**Pod çš„ IP åœ°å•€æ˜¯ä¸•å›ºå®šçš„**ã€‚å½“ä¸€ä¸ª Pod æŒ‚æŽ‰å¹¶è¢«é‡•å»ºå•Žï¼Œå®ƒä¼šèŽ·å¾—ä¸€ä¸ªæ–°çš„ IP åœ°å•€ã€‚

è¿™å°±æ„•å‘³ç•€ï¼Œå¦‚æžœä¸€ä¸ªâ€œå‰•ç«¯â€• Pod æƒ³è®¿é—®ä¸€ä¸ªâ€œå•Žç«¯â€• Podï¼Œå®ƒä¸•èƒ½ç¡¬ç¼–ç •å•Žç«¯çš„ IP åœ°å•€ã€‚æˆ‘ä»¬éœ€è¦ï¿½ï¿½ï¿½ä¸€ç§•æœºåˆ¶ï¼Œèƒ½å¤Ÿï¼š

1. ä¸ºä¸€ç»„æ••ä¾›ç›¸å•Œæœ•åŠ¡çš„ Pod æ••ä¾›ä¸€ä¸ª**ç¨³å®šã€•ä¸•å•˜çš„**è®¿é—®å…¥å•£ã€‚
2. è‡ªåŠ¨è¿½è¸ªè¿™ç»„ Pod çš„ IP åœ°å•€å•˜åŒ–ï¼Œæ›´æ–°è·¯ç”±ä¿¡æ•¯ã€‚
3. åœ¨å¤šä¸ª Pod å‰¯æœ¬ä¹‹é—´è¿›è¡Œ**è´Ÿè½½å•‡è¡¡**ã€‚

`Service` å°±æ˜¯ Kubernetes ä¸ºè§£å†³è¿™ä¸ªé—®é¢˜è€Œè®¾è®¡çš„æ ¸å¿ƒèµ„æº•ã€‚

2. Service çš„å·¥ä½œåŽŸç•†

`Service` çš„æ ¸å¿ƒæ€•æƒ³æ˜¯åœ¨å®¢æˆ·ç«¯å’Œ Pod ä¹‹é—´å¢žåŠ ä¸€ä¸ªæŠ½è±¡å±‚ã€‚å®ƒé€šè¿‡**æ ‡ç¾é€‰æ‹©å™¨ (Label Selector)** æ•¥æ‰¾åˆ°å®ƒè¦•ä»£ç•†çš„ä¸€ç»„ Podã€‚

å½“ä¸€ä¸ª `Service` è¢«åˆ›å»ºæ—¶ï¼Œä¼šå•‘ç”Ÿä¸¤ä»¶ä¸»è¦•çš„äº‹æƒ…ï¼š

1. **åˆ†é…•è™šæ‹Ÿ IP (ClusterIP)**: Kubernetes ä¼šä¸ºè¿™ä¸ª Service åˆ†é…•ä¸€ä¸ª**è™šæ‹Ÿçš„ã€•ä»…åœ¨é›†ç¾¤å†…éƒ¨æœ‰æ•ˆçš„ IP åœ°å•€**ã€‚è¿™ä¸ª IP åœ°å•€æ˜¯ç¨³å®šçš„ï¼Œå•ªè¦• Service å˜åœ¨ï¼Œå®ƒå°±ä¸•ä¼šæ”¹å•˜ã€‚
2. **åˆ›å»º Endpoints (æˆ– EndpointSlice)**: Kubernetes ä¼šè‡ªåŠ¨åˆ›å»ºä¸€ä¸ª `EndpointSlice`

å¯¹è±¡ã€‚è¿™ä¸ªå¯¹è±¡ä¼šæŒ•ç»åœ°ã€•è‡ªåŠ¨åœ°å¯—å‡ºæ‰€æœ‰è¢« Service çš„æ ‡ç¾é€‰æ‹©å™¨åŒ¹é…•å°çš„ã€•å¹¶ä¸”å¤„äºŽ `Ready` çŠ¶æ€•çš„ Pod çš„çœŸå®ž IP åœ°å•€å'Œç«¯å•£ã€‚

å½“é›†ç¾¤å†…çš„ä»»ä½•ä¸€ä¸ªå®¢æˆ·ç«¯ï¼ˆä¾‹å¦‚å•¦ä¸€ä¸ª Podï¼‰å°•è¯•è®¿é—® Service çš„ ClusterIP æ—¶ï¼ŒèŠ‚ç‚¹ä¸Šçš„ `kube-proxy` ç»„ä»¶ä¼šæ‹¦æˆªè¿™ä¸ªè¯·æ±‚ï¼Œå¹¶æ ¹æ® `EndpointSlice` ä¸çš„åˆ—è¡¨ï¼Œä»Žå•Žç«¯çš„å¥åº· Pod ä¸é€‰æ‹©ä¸€ä¸ªï¼Œç„¶åŽå°†æµ•é‡•è½¬å•'è¿‡åŽ»ï¼Œä»Žè€Œå®žçŽ°äº†è´Ÿè½½å•‡è¡¡ã€‚

![Service Architecture](https://miro.medium.com/v2/resize:fit:1200/1*OBWhC0b_n6xG_a_msH2uFw.png)

3. Service çš„ç±»åž‹

`Service` æœ‰å¤šç§•ç±»åž‹ï¼Œç"¨äºŽæ»¡è¶³ä¸•å•Œçš„æš´éœ²éœ€æ±‚ï¼š

- **`ClusterIP`**:**é»˜è®¤ç±»åž‹**ã€‚ä¸º Service åˆ†é…•ä¸€ä¸ªé›†ç¾¤å†…éƒ¨çš„è™šæ‹Ÿ IPã€‚**å•ªèƒ½åœ¨é›†ç¾¤å†…éƒ¨è®¿é—®**ã€‚**é€‚ç"¨åœºæ™¯**: å¤§å¤šæ•°é›†ç¾¤å†…éƒ¨æœ•åŠ¡ä¹‹é—´çš„é€šä¿¡ï¼Œä¾‹å¦‚å‰•ç«¯æœ•åŠ¡è®¿é—®å•Žç«¯æœ•åŠ¡ã€•API ç½'å…³è®¿é—®å¾®æœ•åŠ¡ã€‚
- **`NodePort`**:åœ¨ `ClusterIP` çš„åŸºç¡€ä¸Šï¼Œé¢•å¤–åœ¨**æ¯•ä¸€ä¸ªå·¥ä½œèŠ‚ç‚¹**ä¸Šéƒ½æ‰“å¼€ä¸€ä¸ªç›¸å•Œçš„ã€•å›ºå®šçš„ç«¯å•£ï¼ˆèŒƒå›´é€šå¸¸æ˜¯ 30000-32767ï¼‰ã€‚ä»»ä½•å'é€•åˆ° `<NodeIP>:<NodePort>` çš„æµ•é‡•éƒ½ä¼šè¢«è½¬å•'åˆ°è¯¥ Service çš„ ClusterIPï¼Œè¿›è€Œè½¬å•'åˆ°å•Žç«¯çš„ Podã€‚**å•¯ä»¥ä»Žé›†ç¾¤å¤–éƒ¨è®¿é—®**ã€‚**é€‚ç"¨åœºæ™¯**: ç"¨äºŽä¸´æ—¶æš´éœ²æœ•åŠ¡æˆ–åœ¨å¼€å•'çŽ¯å¢ƒä¸å¿«é€Ÿæµ‹è¯•ï¼Œä¸•å»ºè®®åœ¨ç"Ÿäº§çŽ¯å¢ƒä¸ç›´æŽ¥ç"¨äºŽå…³é"®ä¸šåŠ¡ï¼Œå› ä¸ºå®ƒç»•è¿‡äº†äº'æœ•åŠ¡å•†çš„è´Ÿè½½å•‡è¡¡å™¨ã€‚
- **`LoadBalancer`**:åœ¨ `NodePort` çš„åŸºç¡€ä¸Šï¼Œé¢•å¤–è¯·æ±‚äº'æœ•åŠ¡å•†ï¼ˆå¦‚ AWS, GCP, Azureï¼‰åˆ›å»ºä¸€ä¸ª**å¤–éƒ¨è´Ÿè½½å•‡è¡¡å™¨**ã€‚è¿™ä¸ªå¤–éƒ¨è´Ÿè½½å•‡è¡¡å™¨ä¼šæœ‰ä¸€ä¸ªå…¬ç½' IPï¼Œå¹¶å°†æµ•é‡•å¯¼å•'æ‰€æœ‰èŠ‚ç‚¹çš„ `NodePort`ã€‚**æ˜¯å•'å…¬ç½'æš´éœ²æœ•åŠ¡çš„æ ‡å‡†æ–¹å¼**ã€‚**é€‚ç"¨åœºæ™¯**: éœ€è¦•ä»Žäº'è•"ç½'å…¬å¼€è®¿é—®çš„åº"ç"¨ï¼Œå¦‚ç½'ç«™ã€•å¯¹å¤–APIã€‚æ¤ç±»åž‹ä»…åœ¨äº' K8s çŽ¯å¢ƒä¸æœ‰æ•ˆã€‚
- **`Headless`**:é€šè¿‡å°† `spec.clusterIP` è®¾ç½®ä¸º `None` æ•¥åˆ›å»ºã€‚Kubernetes ä¸•ä¼šä¸ºå®ƒåˆ†é…• ClusterIPã€‚å½“æŸ¥è¯¢è¿™ä¸ª Service çš„ DNS å••ç§°æ—¶ï¼Œå®ƒä¸•ä¼šè¿"å›žä¸€ä¸ªè™šæ‹Ÿ IPï¼Œè€Œæ˜¯ç›´æŽ¥è¿"å›ž**æ‰€æœ‰å•Žç«¯ Pod çš„ IP åœ°å•€åˆ—è¡¨**ã€‚**é€‚ç"¨åœºæ™¯**: ç"¨äºŽ StatefulSetï¼Œä¸ºæ¯•ä¸ªæœ‰çŠ¶æ€•çš„ Pod æ••ä¾›ç‹¬ç«‹çš„ã€•ç¨³å®šçš„ DNS è®°å½•ï¼›æˆ–è€…å½“å®¢æˆ·ç«¯å¸Œæœ›è‡ªå·±æ¥å†³å®šè¿žæŽ¥å"ªä¸ª Pod å®žä¾‹æ—¶ã€‚

4. æœ•åŠ¡å•‘çŽ°ä¸Ž DNS

Kubernetes é›†ç¾¤å†…éƒ¨æœ‰ä¸€ä¸ª DNS æœ•åŠ¡ï¼ˆé€šå¸¸æ˜¯ CoreDNSï¼‰ã€‚å½“ä¸€ä¸ª Service è¢«åˆ›å»ºæ—¶ï¼ŒDNS æœ•åŠ¡ä¼šè‡ªåŠ¨ä¸ºå…¶åˆ›å»ºä¸€æ•¡ DNS A è®°å½•ï¼š ` <service-name>.<namespace>.svc.cluster.local` è¿™æ•¡è®°å½•ä¼šè§£æž•åˆ°è¯¥ Service çš„ ClusterIPã€‚

è¿™æ„•å‘³ç•€ï¼Œåœ¨å•Œä¸€ä¸ª `namespace` ä¸‹çš„ Podï¼Œå•¯ä»¥ç›´æŽ¥é€šè¿‡ `<service-name>` æ•¥è®¿é—®å•¦ä¸€ä¸ªæœ•åŠ¡ï¼Œæ— éœ€å…³å¿ƒå…¶ IP åœ°å•€ã€‚ä¾‹å¦‚ï¼Œ`order-service` å•¯ä»¥ç›´æŽ¥é€šè¿‡ `http://user-service` æ•¥è®¿é—® `user-service`ã€‚

ðŸ› ï¸• å®žè·µæ“•ä½œ (50%)

1. ä¸º Deployment åˆ›å»º ClusterIP Service

æˆ‘ä»¬å°†ä¸º Day 2 åˆ›å»ºçš„ `nginx-deployment` åˆ›å»ºä¸€ä¸ª Serviceã€‚ åˆ›å»ºä¸€ä¸ªæ–‡ä»¶ `nginx-service.yaml`:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: ClusterIP # å•¯ä»¥çœ•ç•¥ï¼Œå› ä¸ºæ˜¯é»˜è®¤å€¼
  selector:
    app: nginx # å…³é”®ï¼šé€‰æ‹©æ‰€æœ‰å¸¦æœ‰ app=nginx æ ‡ç¾çš„ Pod
  ports:
    - protocol: TCP
      port: 80 # Service è‡ªèº«æš´éœ²çš„ç«¯å•£
      targetPort: 80 # æµ•é‡•è¦•è½¬å•‘åˆ° Pod çš„å“ªä¸ªç«¯å•£
```

éƒ¨ç½²å®ƒ:

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

2. éªŒè¯• ClusterIP è®¿é—®

```
# æŸ¥çœ‹ Serviceï¼Œæ³¨æ„•å®ƒèŽ·å¾—äº†ä¸€ä¸ª CLUSTER-IP
kubectl get svc nginx-service
# NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
# nginx-service   ClusterIP   10.108.111.222   <none>        80/TCP    15s

# å•¯åŠ¨ä¸€ä¸ªä¸´æ—¶çš„ busybox Podï¼Œç”¨äºŽåœ¨é›†ç¾¤å†…éƒ¨æµ‹è¯•è®¿é—®
kubectl run -it --rm busybox --image=busybox -- /bin/sh

# åœ¨ busybox çš„ shell ä¸ï¼Œä½¿ç”¨ wget è®¿é—® Service çš„ DNS å••ç§°
# wget -q -O - http://nginx-service
# <!DOCTYPE html>
# <html>
# <head>
# <title>Welcome to nginx!</title>
# ...
# </html>

# å¤šæ¬¡è®¿é—®ï¼Œæµ•é‡•ä¼šè¢«è´Ÿè½½å•‡è¡¡åˆ°ä¸•å•Œçš„ Nginx Pod
```

3. æŸ¥çœ‹ Endpoints

`kube-proxy` æ˜¯å¦‚ä½•çŸ¥é•"è¦•å°†æµ•é‡•è½¬å•'åˆ°å"ªä›› Pod çš„å'¢ï¼Ÿç"æ¡ˆæ˜¯ `EndpointSlice`ã€‚

```
# æŸ¥çœ‹ä¸Ž Service å…³è•"çš„ EndpointSlice
kubectl get endpointslice -l kubernetes.io/service-name=nginx-service
# NAME                      ADDRESSTYPE   PORTS   ENDPOINTS
      AGE
# nginx-service-abcde       IPv4          80
10.244.1.10,10.244.2.8,10.244.3.9   5m

# å•¯ä»¥çœ‹åˆ°ï¼Œå®ƒåˆ—å‡ºäº†æ‰€æœ‰å•Žç«¯ Pod çš„çœŸå®ž IP åœ°å•€ã€‚
```

4. å°† Service æš´éœ²åˆ°é›†ç¾¤å¤– (NodePort)

ä¿®æ”¹ `nginx-service.yaml`ï¼Œå°† `type` æ”¹ä¸º `NodePort`ã€‚

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: NodePort # ä¿®æ”¹ç±»åž‹
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      # nodePort: 30080 # å•¯ä»¥æŒ‡å®šä¸€ä¸ªç«¯å•£ï¼Œä½†é€šå¸¸è®© K8s
è‡ªåŠ¨åˆ†é…•
```

é‡•æ–°åº"ç"¨: `kubectl apply -f nginx-service.yaml`

æŸ¥çœ‹ Serviceï¼Œæ³¨æ„ `PORT(S)` åˆ—çš„å•˜åŒ–ï¼š

```
kubectl get svc nginx-service
# NAME            TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
# nginx-service   NodePort   10.108.111.222   <none>        80:31234/TCP   10m
# 80:31234 çš„æ„•æ€•æ˜¯ï¼ŒService çš„ 80 ç«¯å•£è¢«æ˜ å°„åˆ°äº†æ‰€æœ‰èŠ‚ç‚¹çš„
31234 ç«¯å•£ã€‚
```

èŽ·å•– minikube èŠ‚ç‚¹çš„ IP åœ°å•€ï¼Œå¹¶ä»Žä½ çš„ç"µè„‘ä¸Šè®¿é—®å®ƒï¼š

```
minikube ip
```

```
# 192.168.49.2

# åœ¨ä½ çš„æµ•è§ˆå™¨æˆ–ä½¿ç”¨ curl è®¿é—®
curl http://192.168.49.2:31234
# <!DOCTYPE html> ...
```

ðŸ'» Go ç¼–ç¨‹å®žçŽ° (10%)

é¡¹ç›®: k8s-service-lister

ç›®æ ‡: ç¼–å†™ä¸€ä¸ª Go ç¨‹åº•ï¼Œåˆ—å‡ºæŒ‡å®šå'½å••ç©ºé—´ä¸‹çš„æ‰€æœ‰ Service å•Šå…¶ç±»åž‹å'Œ ClusterIPã€‚

```
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() {
 if len(os.Args) < 2 {
  fmt.Println("ç”¨æ³•: go run main.go <namespace>")
  os.Exit(1)
 }
 namespace := os.Args[1]

 // --- é…•ç½®å'Œåˆ›å»º clientset ---
 userHomeDir, _ := os.UserHomeDir()
 kubeconfig := filepath.Join(userHomeDir, ".kube", "config")
 config, _ := clientcmd.BuildConfigFromFlags("", kubeconfig)
 clientset, _ := kubernetes.NewForConfig(config)

 fmt.Printf("--- Services in namespace '%s' ---\n", namespace)
 serviceList, err :=
clientset.CoreV1().Services(namespace).List(context.TODO(),
metav1.ListOptions{})
 if err != nil {
  log.Fatal(err)
 }

 for _, svc := range serviceList.Items {
  fmt.Printf("- Name: %s\n", svc.Name)
  fmt.Printf("  Type: %s\n", svc.Spec.Type)
  fmt.Printf("  ClusterIP: %s\n", svc.Spec.ClusterIP)
  fmt.Println("--------------------")
 }
}
```

è¿•è¡Œ:

```
go run main.go default
# --- Services in namespace 'default' ---
# - Name: kubernetes
#   Type: ClusterIP
#   ClusterIP: 10.96.0.1
# --------------------
# - Name: nginx-service
#   Type: NodePort
#   ClusterIP: 10.108.111.222
# --------------------
```

ðŸ”• æ•…éšœæŽ’æŸ¥ä¸Žä¼˜åŒ–

- **æ— æ³•é€šè¿‡ Service å••ç§°è®¿é—®**:æ£€æŸ¥ DNS æ˜¯å¦æ£å¸¸: `kubectl exec -it <pod-name> -- nslookup <service-name>`ã€‚æ£€æŸ¥ Service çš„ `selector` æ˜¯å¦æ£ç¡®ï¼Œæ˜¯å¦èƒ½åŒ¹é…åˆ° Pod çš„ `labels`ã€‚
- **æ— æ³•é€šè¿‡ Service IP è®¿é—®**:`kubectl describe svc <service-name>` æŸ¥çœ‹ `Endpoints` æ˜¯å¦ä¸ºç©ºã€‚å¦‚æžœ `Endpoints` ä¸ºç©ºï¼Œè¯´æ˜Žæ²¡æœ‰å¥åº·çš„ã€•`Ready` çŠ¶æ€çš„ Pod è¢«é€‰ä¸ã€‚æ£€æŸ¥åŽç«¯ Pod çš„çŠ¶æ€•å'Œå•¥åº·æŽ¢é'ˆï¼ˆDay 5 å†…å®¹ï¼‰ã€‚
- **æ— æ³•é€šè¿‡ NodePort è®¿é—®**:æ£€æŸ¥é˜²ç«å¢™è§„åˆ™ï¼Œç¡®ä¿èŠ‚ç‚¹ä¸Šçš„ç«¯å£æ˜¯å¼€æ”¾çš„ã€‚æ£€æŸ¥ `kube-proxy` Pod æ˜¯å¦åœ¨æ‰€æœ‰èŠ‚ç‚¹ä¸Šéƒ½æ£å¸¸è¿•è¡Œã€‚

ðŸ• è¯¾åŽä½œä¸š

1. **ç ”ç©¶ `EndpointSlice`**: ä½¿ç”¨ `kubectl get endpointslice` å'Œ `kubectl describe endpointslice <name>` å'½ä»¤ï¼Œè¯¦ç»†æŸ¥çœ‹ `EndpointSlice` å¯¹è±¡çš„å†…å®¹ï¼Œç†è§£å®ƒæ˜¯å¦‚ä½•å°† Service ä¸Žä¸€ç»„ Pod IP åœ°å•€å…³è•”èµ·æ•¥çš„ã€‚
2. **Headless Service å®žè·µ**: åˆ›å»ºä¸€ä¸ª `Headless` Service (è®¾ç½® `clusterIP: None`)ï¼Œå¹¶ä¸ºå®ƒå…³è•” Nginx Deploymentã€‚ç„¶åŽåœ¨ä¸€ä¸ªä¸´æ—¶ Pod ä¸ä½¿ç”¨ `nslookup <headless-service-name>`ï¼Œè§‚å¯Ÿè¿”å›žçš„ç»“æžœä¸Žï¿½ï¿½é€š Service æœ‰ä½•ä¸•å•Œã€‚