

Exposer une adresse IP externe pour accéder à une application dans un cluster

Cet exercice montre comment créer un objet de service Kubernetes qui expose une adresse IP externe.

Objectifs

- Exécutez cinq instances d'une application Hello World.
- Créez un objet de service qui expose une adresse IP externe.
- Utilisez l'objet Service pour accéder à l'application en cours d'exécution.

Création d'un service pour une application s'exécutant dans cinq pods.

1. Exécutez une **application Hello World** dans votre cluster :

```
sudo kubectl run hello-world --replicas=5 --labels="run=load-balancer-example" --image=gcr.io/google-samples/node-hello:1.0 --port=8080
```

La commande précédente crée un objet Deployment et un objet ReplicaSet associé. Le ReplicaSet dispose de cinq pods, chacun exécutant l'application Hello World.

2. Affichez les informations sur le déploiement :

```
sudo kubectl get deployments hello-world
```

```
sudo kubectl describe deployments hello-world
```

3. Affichez des informations sur vos objets ReplicaSet :

```
sudo kubectl get replicaset
```

```
sudo kubectl describe replicaset
```

4. Créez un objet Service qui expose le déploiement :

```
sudo kubectl expose deployment hello-world --type=LoadBalancer --name=my-service
```

5. Afficher les informations sur le service :

```
sudo kubectl get services my-service
```

Le résultat est similaire à ceci:

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|------------|-----------|--------------|----------------|----------|-----|
| my-service | ClusterIP | 10.3.245.137 | 104.198.205.71 | 8080/TCP | 54s |

Remarque : Si l'adresse IP externe est indiquée comme <en attente> (pending), attendez une minute et entrez à nouveau la même commande.

6. Affichez des informations détaillées sur le service :

sudo kubectl describe services my-service

Le résultat est similaire à ceci :

```
Name:      my-service
Namespace:  default
Labels:     run=load-balancer-example
Annotations: <none>
Selector:   run=load-balancer-example
Type:       LoadBalancer
IP:         10.3.245.137
LoadBalancer Ingress: 104.198.205.71
Port:       <unset> 8080/TCP
NodePort:   <unset> 32377/TCP
Endpoints:  10.0.0.6:8080,10.0.1.6:8080,10.0.1.7:8080 + 2 more...
Session Affinity: None
Events:     <none>
```

Notez l'adresse IP externe (LoadBalancer Ingress) exposée par votre service. Dans cet exemple, l'adresse IP externe est 104.198.205.71. Notez également la valeur de Port et NodePort. Dans cet exemple, le port est 8080 et le port NodePort est 32377.

7. Dans la sortie précédente, vous pouvez voir que le service a plusieurs extrémités : 10.0.0.6:8080,10.0.1.6:8080,10.0.1.7:8080 + 2 autres. Il s'agit des adresses internes des pods exécutant l'application Hello World. Pour vérifier que ce sont des adresses de pod, entrez cette commande :

sudo kubectl get pods --output=wide

Le résultat est similaire à ceci :

```
NAME                ... IP      NODE
hello-world-2895499144-1jaz9 ... 10.0.1.6  gke-cluster-1-default-pool-e0b8d269-1afc
hello-world-2895499144-2e5uh ... 10.0.1.8  gke-cluster-1-default-pool-e0b8d269-1afc
hello-world-2895499144-9m4h1 ... 10.0.0.6  gke-cluster-1-default-pool-e0b8d269-5v7a
```

```
hello-world-2895499144-o4z13 ... 10.0.1.7 gke-cluster-1-default-pool-e0b8d269-1afc  
hello-world-2895499144-segjf ... 10.0.2.5 gke-cluster-1-default-pool-e0b8d269-cpuc
```

8. Utilisez l'adresse IP externe (LoadBalancer Ingress) pour accéder à l'application Hello World :

`sudo curl http://<external-ip>:<port>`

où <ip-external> est l'adresse IP externe (LoadBalancer Ingress) de votre service, et <port> la valeur de Port dans la description de votre service.

Si vous utilisez minikube, taper **minikube service my-service** ouvrira automatiquement l'application Hello World dans un navigateur. La réponse à une demande réussie est un message hello :

Hello Kubernetes!

Nettoyer

Pour supprimer le service, entrez cette commande :

`sudo kubectl delete services my-service`

Pour supprimer le déploiement, le ReplicaSet et les pods exécutant l'application Hello World, entrez la commande suivante :

`sudo kubectl delete deployment hello-world`