

Travaux Pratiques: Kubernetes - Helm

Introduction

Dans ce TP, nous allons déployer un Ingress Controller en utilisant un template Helm

Prérequis

- Un cluster Kubernetes (> 1.16)

Installation de Helm

Télécharger la dernière version release par la communauté :

<https://github.com/helm/helm/releases> (<https://github.com/helm/helm/releases>)

```
$ curl -fsSL https://get.helm.sh/helm-v3.3.1-linux-amd64.tar.gz -o helm-v3.3.1-linux-amd64.tar.gz
$ tar -zxvf helm-v3.3.1-linux-amd64.tar.gz
$ chmod +x linux-amd64/helm
# mv linux-amd64/helm /usr/bin/helm

$ helm version
version.BuildInfo{Version:"v3.3.1",
GitCommit:"249e5215cde0c3fa72e27eb7a30e8d55c9696144", GitTreeState:"clean",
GoVersion:"go1.14.7"}
```

Préparation de notre Release

Le fichier de values à utiliser se trouve dans tp/containers/helm/values.yml

Chargeons le repository Helm de notre release :

```
$ helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
$ helm repo update
```

Nous devons ensuite trouver le nom du chart que l'on souhaite utiliser ainsi que sa version :

```
$ helm search repo nginx
NAME                                CHART VERSION   APP VERSION   DESCRIPTION
ingress-nginx/ingress-nginx        2.15.0          0.35.0        Ingress controller for
Kubernetes using NGINX a...
```

On choisit le premier qui est

Le Hub Helm <https://hub.helm.sh> (<https://hub.helm.sh>) peut être recherché avec la commande `helm search hub <keyword>`.

On retient le nom du chart et sa version pour le déploiement de la Release.

Installation d'une Release

```
$ helm install ingress-nginx --version 2.15.0 --values values.yml ingress-nginx/ingress-nginx
```

Upgrade d'une Release

On se propose de changer les CPU Requests et les Memory Requests. Passons les respectivement à 500m et 500Mi et déclenchons un update :

```

$ helm upgrade ingress-nginx --values values.yml ingress-nginx/ingress-nginx
Release "ingress-nginx" has been upgraded. Happy Helming!
NAME: ingress-nginx
LAST DEPLOYED: Wed Sep  9 09:38:20 2020
NAMESPACE: default
STATUS: deployed
REVISION: 3
TEST SUITE: None
NOTES:
The ingress-nginx controller has been installed.

[...]

$ kubectl describe pod ingress-nginx-controller-n9drn
Name:          ingress-nginx-controller-n9drn
Namespace:     default
Priority:       0
Node:          master/10.42.42.42
Start Time:    Wed, 09 Sep 2020 09:43:05 +0000
Labels:        app.kubernetes.io/component=controller
               app.kubernetes.io/instance=ingress-nginx
               app.kubernetes.io/name=ingress-nginx
               controller-revision-hash=64fbd68584
               pod-template-generation=3
Annotations:   <none>
Status:        Running
IP:            10.42.42.42
IPs:
  IP:          10.42.42.42
Controlled By: DaemonSet/ingress-nginx-controller
Containers:
  controller:
    Container ID:
containerd://161ad1a470427448953aad69685b5c773fb822cd889690b9f2be881933390455
    Image:      k8s.gcr.io/ingress-
nginx/controller:v0.35.0@sha256:fc4979d8b8443a831c9789b5155cded454cb7de737a8b72
7bc2ba0106d2eae8b
  [...]
  Limits:
    cpu:        1
    memory:     1Gi
  Requests:
    cpu:        500m
    memory:     500Mi

```

Rollback

Si finalement nous considérons que les anciennes CPU Requests et les Memory Requests étaient mieux adaptées, nous pouvons décider de revenir en arrière avec l'aide de la fonction rollback de Helm.

```

$ helm history ingress-nginx
REVISION    UPDATED                               STATUS    CHART                                APP
VERSION DESCRIPTION
1           Mon Sep  7 18:02:11 2020    superseded ingress-nginx-2.15.0
0.35.0      Install complete
2           Mon Sep  7 18:06:06 2020    superseded ingress-nginx-2.15.0
0.35.0      Upgrade complete
3           Wed Sep  9 09:38:20 2020    deployed  ingress-nginx-2.15.0
0.35.0      Upgrade complete

$ helm rollback ingress-nginx 2
Rollback was a success! Happy Helming!

$ kubectl describe pod ingress-nginx-controller-99kio
Name:          ingress-nginx-controller-99kio
Namespace:     default
Priority:       0
Node:          master/10.42.42.42
Start Time:    Wed, 09 Sep 2020 09:43:05 +0000
Labels:        app.kubernetes.io/component=controller
               app.kubernetes.io/instance=ingress-nginx
               app.kubernetes.io/name=ingress-nginx
               controller-revision-hash=64po976425
               pod-template-generation=4
Annotations:   <none>
Status:        Running
IP:            10.42.42.42
IPs:
  IP:          10.42.42.42
Controlled By: DaemonSet/ingress-nginx-controller
Containers:
  controller:
    Container ID:
containerd://161ad1a470427448953aad69685b5c773fb822cd889690b9f2be881933390455
    Image:      k8s.gcr.io/ingress-
nginx/controller:v0.35.0@sha256:fc4979d8b8443a831c9789b5155cded454cb7de737a8b72
7bc2ba0106d2eae8b
[...]
```

```

Limits:
  cpu:    1
  memory: 1Gi
Requests:
  cpu:    300m
  memory: 300Mi

```

Conclusion

Vous avez à votre disposition un cluster de deux nœuds déployés avec kubeadm. Si vous avez le temps, reprenez les TPs minikube et testez les sur ce cluster.

Copyright

- Licence : [Creative Commons BY-SA 4.0 \(https://creativecommons.org/licenses/by-sa/4.0/deed.fr\)](https://creativecommons.org/licenses/by-sa/4.0/deed.fr)
- Copyright © 2014-2019 alter way Cloud Consulting
- Copyright © 2020 particule.
- Depuis 2020, tous les commits sont la propriété de leurs auteurs respectifs