

Helm

Contents

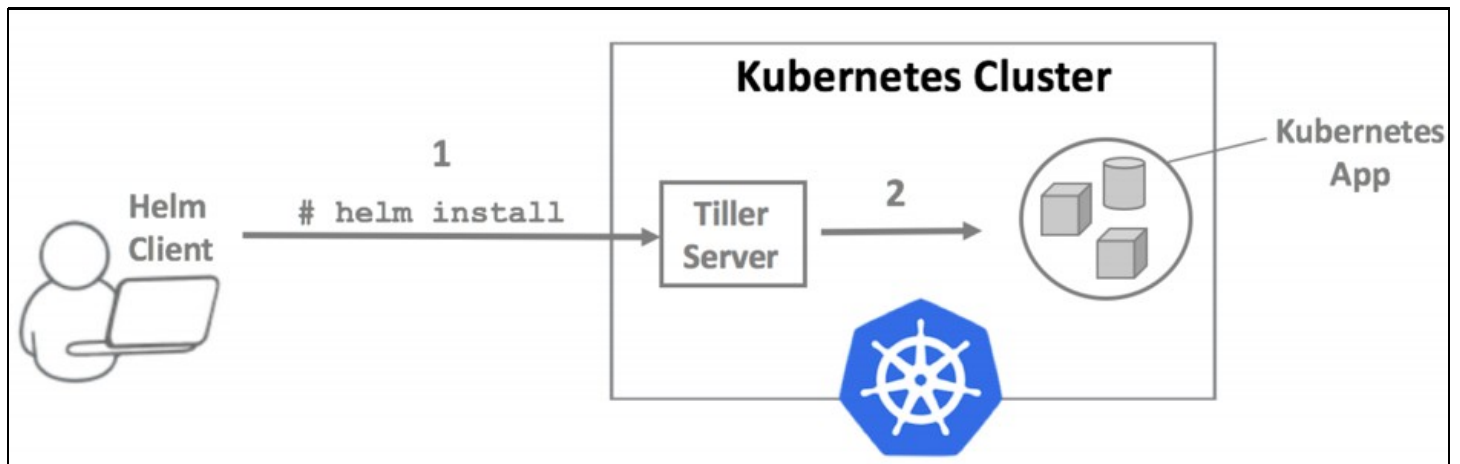
- 1 Bevezet?
- 2 Telepítés
 - ♦ 2.1 Telepítés saját névtérbe
 - ◊ 2.1.1 Klaszter role
- 3 Prometheus telepítése

Bevezet?

A Helm a Kubernetes-hez készült csomag kezelő, mint pl a RedHat világában a yum ill dnf parancsok. A Helm mögé olyan nagy cégek sorakoztak föl mint pl a Google vagy a Microsoft. A Helm alapja az úgynevezett Helm chart, mely egy telepítési leírója egy adott szoftver csomagnak. Egy adott Helm chart telepítésével egy mozdulattal tudunk el?re gyártott, összetett szoftver konfigurációt telepíteni egy Kubernetes klaszterbe. Pl a Prometheus Helm-chart tartalmazza magát a Prometheus szervert, az Alert manager-t, a node-exporter-t, meg még jó pár komponenst, amiket a chart varázsütésre ftelepít a Kubernetes klaszterbe, úgy hogy az összes komponens m?ködésre készen konfigurálva van, úgy hogy a Helm az összes szükséges Kubernetes objektumot létrehozza, beleértve a RoleBinding-okat és Ingress komponenseket is.

A Helm rendszer (3 verzió el?tt) négy f? komponensb?l áll.

- Helm repository: Ez a docker hub-hoz hasonlatos repository, ahol a Docker-hub-hoz hasonlóan tudunk Helm-chart-okat keresni, letölteni. Ez az els?dleges forrása a chart-ok beszerzésének.
- Helm parancssor:
- Helm chart:
- Tiller:



A Helm templéteket használ a Kubernetes yaml objektum leíró fájlok elkészítésére. A Helm telepítés el?tt rendereli a templéteket, ebb?l el?állnak a Kubernetes objektum leírók, majd ezeket a Tiller-en keresztül telepíti a Kubernetes klaszterbe. A chart-ok a `<user-home>/charts/<chart-név>` mappában vannak. Ezen belül a **templates** mappában vannak a yaml templétek.

```
# cat server-pvc.yaml
{{- if not .Values.server.statefulSet.enabled -}}
{{- if .Values.server.persistentVolume.enabled -}}
{{- if not .Values.server.persistentVolume.existingClaim -}}
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  {{- if .Values.server.persistentVolume.annotations }}
  annotations:
  ...
```

A placeholdereket a chart mappában található values.yaml fájl alapján tölti ki a Helm. Ez minden chart kötelező eleme:

```
serviceAccounts:
  alertmanager:
    create: true
    name:
  kubeStateMetrics:
    create: true
    name:
  nodeExporter:
    create: true
  ...
```

A values.yaml fájl átírásával lehet testre szabni a chart-okat.

Telepítés

Helm letöltése: https://helm.sh/docs/using_helm/#installing-helm

A helm binárisra létre kell hozni egy sim-linket a bin mappában:

```
# ln -s /home/adam/Programs/helm/helm2/helm /usr/bin/helm
```

A Tiller-t a helm

```
# helm init
$HELM_HOME has been configured at /root/.helm.
```

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users' policy.
To prevent this, run 'helm init' with the --tiller-tls-verify flag.
For more information on securing your installation see: https://docs.helm.sh/using_helm/#securing-your-helm-installation



Note A Tiller

```
# kubectl get all -n kube-system
```

NAME	READY	STATUS	RESTARTS	AGE
pod/tiller-deploy-775c9dcf94-x429k	1/1	Running	0	5m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/tiller-deploy	ClusterIP	172.30.26.107	<none>	44134/TCP	5m

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/tiller-deploy	1	1	1	1	5m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/tiller-deploy-775c9dcf94	1	1	1	5m

Telepítés saját névtérbe

Az a cél, hogy a Tiller-t a saját névtérünkbe telepítsük föl és korlátozzuk a hozzáférését, hogy kizárólag ehhez a névtérhez legyen joga, semmi másához ne férjen hozzá:

Hozzunk létre egy service-account-ot a tiller-nek:

```
# kubectl create serviceaccount tiller --namespace mynamespace
serviceaccount/tiller created
```

Készítünk egy role-t a saját névtérünkhöz, amiben minden m?veletet megengedünk a mynamespace névtérben:
role-tiller.yaml

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tiller
  namespace: mynamespace
rules:
- apiGroups: ["", "batch", "extensions", "apps"]
  resources: ["*"]
  verbs: ["*"]

# kubectl apply -f role-tiller.yaml
role.rbac.authorization.k8s.io/tiller created
```

Majd összekötjük a role-t a service-account-al:
rb-tiller.yaml

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: tiller-binding
  namespace: mynamespace
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: mynamespace
roleRef:
  kind: Role
  name: tiller
  apiGroup: rbac.authorization.k8s.io

# kubectl apply -f rb-tiller.yaml
rolebinding.rbac.authorization.k8s.io/tiller-binding created
```

A tiller init -ben meg kell adni a service-account-ot és a névtérét:

```
# helm init --service-account tiller --tiller-namespace mynamespace
$HELM_HOME has been configured at /root/.helm.
```

Tiller (the Helm server-side component) has been installed into your Kubernetes Cluster.

Ellen?rizzük hogy a jó névtérben fut e a tiller pod:

```
# kubectl get pod -n mynamespace
NAME                                READY   STATUS    RESTARTS   AGE
tiller-deploy-6cddf5788b-hcpr8     1/1     Running   0           1m
```

Klaszter role

Ha a névtéren kívül is létre akarunk hozatni elemeket akkor klaszter szint? jogokat kell adni a tiller sa-nak:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: tiller-clusterrole
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: mynamespace
```

Prometheus telepítése

<https://hub.helm.sh/>

```
# helm search prometheus
NAME                                CHART VERSION   APP VERSION     DESCRIPTION
stable/prometheus                  8.14.0          2.11.1          Prometheus is a monitoring system and time series database.
```

```
# helm inspect values stable/prometheus
rbac:
  create: true
imagePullSecrets:
# - name: "image-pull-secret"
## Define serviceAccount names for components. Defaults to component's fully qualified name.
##
serviceAccounts:
  alertmanager:
    create: true
....
```

```
PROM_ADDR=mon.$(minishfit ip).nip.io
AM_ADDR=alertmanager.$(minishfit ip).nip.io
```

```
# helm list --tiller-namespace mynamespace
NAME          REVISION   UPDATED              STATUS    CHART              APP VERSION   NAMESPACE
prometheus    1          Tue Jul 16 00:06:52 2019  DEPLOYED  prometheus-8.14.0  2.11.1        mynamespace
```

```
# helm delete prometheus --tiller-namespace mynamespace
release "prometheus" deleted
```