Chapter 15

Helm

15.1 Labs

Exercise 15.1: Working with Helm and Charts

Overview

helm allows for easy deployment of complex configurations. This could be handy for a vendor to deploy a multi-part application in a single step. Through the use of a `Chart`, or template file, the required components and their relationships are declared. Local agents like **Tiller** use the API to create objects on your behalf. Effectively its orchestration for orchestration.

There are a few ways to install **Helm**. The newest version may require building from source code. We will download a recent, stable version. Once installed we will deploy a `Chart`, which will configure **Hadoop** on our cluster.

Install Helm

1. On the `master` node use **wget** to download the compressed tar file. The short URL below is for: `https://storage.googleapis.com/kubernetes-helm/helm-v2.7.0-linux-amd64.tar.gz`

   ```
   student@lfs458-node-1a0a:~$ wget goo.gl/nbEcHn
   ```

2. Uncompress and expand the file.

   ```
   student@lfs458-node-1a0a:~$ tar -xvf nbEcHn
   linux-amd64/
   linux-amd64/README.md
   linux-amd64/helm
   linux-amd64/LICENSE
   ```

3. Copy the **helm** binary to the `/usr/local/bin/` directory, so it is usable via the shell search path.

   ```
   student@lfs458-node-1a0a:~$ sudo cp linux-amd64/helm /usr/local/bin/
   ```

4. Due to new **RBAC** configuration **helm** is unable to run in the `default` namespace, in this version of Kubernetes. During initialization you could choose to create and declare a new namespace. Other RBAC issues may be encountered even then. In this lab we will create a service account for **tiller**, and give it admin abilities on the cluster. More on **RBAC** in another chapter.

   Begin by creating the `serviceaccount` object.

```
student@lfs458-node-1a0a:~$ kubectl create serviceaccount \
 --namespace kube-system tiller
serviceaccount "tiller" created
```

5. Bind the `serviceaccount` to the admin role called `cluster-admin` inside the `kube-system` namespace.

```
student@lfs458-node-1a0a:~$ kubectl create clusterrolebinding \
    tiller-cluster-rule \
    --clusterrole=cluster-admin \
    --serviceaccount=kube-system:tiller
clusterrolebinding "tiller-cluster-rule" created
```

6. We can now initialize **helm**. This process will also configure **tiller** the client process. There are several possible options to pass such as `nodeAffinity`, a particular version of software, alternate storage backend, and even a dry-run option to generate JSON or YAML output. The output could be edited and ingested into **kubectl**. We will use default values in this case.

```
student@lfs458-node-1a0a:~$ helm init

<output_omitted>
```

7. Update the `tiller-deploy` deployment to have the service account.

```
student@lfs458-node-1a0a:~$ kubectl patch deployment \
    tiller-deploy -p \
    '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}' \
    -n kube-system
```

8. Verify the **tiller** pod is running. Examine the logs of the pod. Note that each line of log begins with an tag of the component generating the messages, such as `[main]`, `[storage]`, and `[storage]`.

```
student@lfs458-node-1a0a:~$ kubectl get pods --all-namespaces
<output_omitted>
kube-system   tiller-deploy-84b97f465c-76lvs  1/1  Running 0 30m

student@lfs458-node-1a0a:~$ kubectl logs tiller-deploy-84b97f465c-76lvs\
                -n kube-system
<output_omitted>
```

9. View the available sub-commands for **helm**. As with other Kubernetes tools, expect ongoing change.

```
student@lfs458-node-1a0a:~$ helm help
<output_omitted>
```

10. View the current configuration files, archives and plugins for helm. Return to this directory after you have worked with a `Chart` later in the lab.

```
student@lfs458-node-1a0a:~$ helm home
/home/student/.helm

student@lfs458-node-1a0a:~$ ls -R /home/student/.helm/
/home/student/.helm/:
cache  plugins  repository  starters

/home/student/.helm/cache:
archive
<output_omitted>
```


11. Verify **helm** and **tiller** are responding, also check the current version installed.

```
student@lfs458-node-1a0a:~$ helm version
Client: &version.Version{SemVer:"v2.7.0", GitCommit:"08c1144f5...
Server: &version.Version{SemVer:"v2.7.0", GitCommit:"08c1144f5...
```

12. Ensure both are upgraded to the most recent stable version.

```
student@lfs458-node-1a0a:~$ helm init --upgrade
```

13. A `Chart` is a collection of containers to deploy an application. There is a collection available on https://github.com/kubernetes/charts/tree/master/stable, provided by vendors, or you can make your own. Take a moment and view the current stable `Charts`. Then search for available stable databases.

```
student@lfs458-node-1a0a:~$ helm search database
NAME                    VERSION  DESCRIPTION
stable/cockroachdb      0.5.4    CockroachDB is a scalable, ...
stable/dokuwiki         0.2.1    DokuWiki is a standards-compliant...
stable/mariadb          2.1.3    Fast, reliable, scalable, and eas...
stable/mediawiki        0.6.1    Extremely powerful, scalable soft...
stable/mongodb          0.4.22   NoSQL document-oriented database ...
stable/mongodb-replicaset 2.1.4  NoSQL document-oriented datab...
stable/mysql            0.3.4    Fast, reliable, scalable, and eas...
<output_omitted>
```

14. We will install the **mariadb**. Take a look at install details https://github.com/kubernetes/charts/tree/master/stable/mariadb#custom-mycnf-configuration The **–debug** option will create a lot of output. Note the interesting name for the deployment, like `illmannered-salamander`. The output will typically suggest ways to access the software. As well we will indicate that we do not want persistent storage, which would require use to create an available `PV`.

```
student@lfs458-node-1a0a:~$ helm --debug install stable/mariadb \
       --set persistence.enabled=false
[debug] Created tunnel using local port: '38396'

[debug] SERVER: "localhost:38396"

[debug] Original chart version: ""
[debug] Fetched stable/mariadb to /home/student/.helm/cache/archive/mar...

[debug] CHART PATH: /home/student/.helm/cache/archive/mariadb-2.0.1.tgz

NAME:   illmannered-salamander
<output_omitted>
```

15. Using some of the information at the end of the previous command output we will deploy another container and access the database. We begin by getting the root password for `illmannered-salamander`. Be aware the output lacks a carriage return, so the next prompt will appear on the same line. We will need the password to access the running `MariaDB` database.

```
student@lfs458-node-1a0a:~$ kubectl get secret --namespace default \
        illmannered-salamander-mariadb \
        -o jsonpath="{.data.mariadb-root-password}" \
        | base64 --decode
IFBldzAQfx
```

16. Now we will install another container to act as a client for the database. We will use **apt-get** to install client software.


```
student@lfs458-node-1a0a:~$ kubectl run -i --tty ubuntu --image=ubuntu:16.04 --restart=Never -- bash -il
If you don't see a command prompt, try pressing enter.
root@ubuntu:/#

root@ubuntu:/# apt-get update ; apt-get install -y mariadb-client
<output_omitted>
```

17. Use the client software to access the database. The following command uses the server name and the root password we found in a previous step. Both of yours will be different.

```
root@ubuntu:/# mysql -h illmannered-salamander-mariadb -p
Enter password: IFBldzAQfx
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 153
Server version: 10.1.28-MariaDB Source distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

MariaDB [(none)]>
MariaDB [(none)]> quit
root@ubuntu:/# exit
```

18. View the `Chart` history on the system. The use of the **-a** option will show all `Charts` including `deleted` and `failed` attempts. The output below shows the current running `Chart` as well as a previously deleted **hadoop** `Chart`.

```
student@lfs458-node-1a0a:~$ helm list -a
NAME      REVISION        UPDATED   STATUS  CHART    NAMESPACE
goodly-beetle  1  Wed Nov  8 23:01:24 2017 DELETED hadoop-1.0.1 default
illmannered-salamander 1   Thu Nov  9 05:00:12 2017 DEPLOYED mariadb-...
```

19. Delete the **mariadb** `Chart`. No output should happen from the list.

```
student@lfs458-node-1a0a:~$  helm delete illmannered-salamander
release "illmannered-salamander" deleted

student@lfs458-node-1a0a:~$ helm list
```

20. Add another repository and view the `Charts` available.

```
student@lfs458-node-1a0a:~$ helm repo add stable \
       http://storage.googleapis.com/kubernetes-charts

student@lfs458-node-1a0a:~$ helm search
NAME                            VERSION        DESCRIPTION
stable/acs-engine-autoscaler         2.1.0          Scales worker nodes within...
stable/artifactory                   6.2.0          Universal Repository Manag...
<output_omitted>
```

