Experiment: Working with Helm and KPT

In this experiment/lab, we'll grab a Helm chart, link it to a KPT package and allow for the handling of changes in the future for any modifications made or upstream package changes.

Install KPT

If you don't already have KPT installed, you can download a binary, alternatively you can build, use a tap, or several other methods for installation as noted here:

https://googlecontainertools.github.io/kpt/installation/

Download a KPT binary here:

https://googlecontainertools.github.io/kpt/installation/binaries/

Install Helm if not already present

https://github.com/helm/helm/releases

For MacOS

```
$ brew install helm
```

For Windows

```
$ choco install kubernetes-helm
```

```
$ helm version
```

Fetch a Helm chart

Make a repo

```
$ mkdir kpt-repo-msql
```

Change to the repo folder

```
$ cd kpt-repo-msql
```

Grab the MySQL helm chart we'll use for this experiment

```
$ helm fetch stable/mysql
```

Now we've pulled a helm chart for expansion. This may optionally be checked into

git so it can be expanded again in the future.

Expand the Helm chart

Validate which image we have since that can change from day to day

```
$ ls

 Volume in drive C is OS
 Volume Serial Number is 5081-CA53

 Directory of C:\k3d\kpt-repo-msql

09/12/2020  06:10 PM    <DIR>          .
09/12/2020  06:10 PM    <DIR>          ..
09/12/2020  06:10 PM            11,424 mysql-1.6.7.tgz
               1 File(s)         11,424 bytes
               2 Dir(s)  143,234,584,576 bytes free
```

$ helm template mysql-1.6.7.tgz --output-dir .

```
wrote .\mysql/templates/secrets.yaml
wrote .\mysql/templates/tests/test-configmap.yaml
wrote .\mysql/templates/pvc.yaml
wrote .\mysql/templates/svc.yaml
wrote .\mysql/templates/deployment.yaml
wrote .\mysql/templates/tests/test.yaml
```

```
$ tree mysql
Folder PATH listing for volume OS
Volume serial number is 5081-CA53
C:\K3D\KPT-REPO-MSQL\MYSQL
└───templates
    └───tests
```

That shows us the folders, now let’s expand to view the files

```
$ tree /f mysql
Folder PATH listing for volume OS
Volume serial number is 5081-CA53
C:\K3D\KPT-REPO-MSQL\MYSQL
└───templates
    │   deployment.yaml
    │   pvc.yaml
    │   secrets.yaml
    │   svc.yaml
    │
    └───tests
            test-configmap.yaml
            test.yaml
```

Publish the kpt package
Initialize git if you don’t have an existing repository for the application

```
~/projects/kpt-repo-msql $ git init
```

```
Initialized empty Git repository in C:/k3d/kpt-repo-msql/.git/
```

Add the files for this project package

```
~/projects/kpt-repo-msql $ git add .
```

Configure required globals for git if not already set

```
~/projects/kpt-repo-msql $ git config --global user.email "you@example.com"
```

```
~/projects/kpt-repo-msql $ git config --global user.name "Your Name"
```

Commit the package to allow us to use for packaging in KPT

```
~/projects/kpt-repo-msql $ git commit -m "Add mysql package"
```

```
[master (root-commit) 9b44544] Add mysql package
 7 files changed, 209 insertions(+)
 create mode 100644 mysql-1.6.7.tgz
 create mode 100644 mysql/templates/deployment.yaml
 create mode 100644 mysql/templates/pvc.yaml
 create mode 100644 mysql/templates/secrets.yaml
 create mode 100644 mysql/templates/svc.yaml
 create mode 100644 mysql/templates/tests/test-configmap.yaml
 create mode 100644 mysql/templates/tests/test.yaml
```

Optionally we’ll tag this repo.

```
~/projects/kpt-repo-msql $ git tag package-examples/mysql/mysql/templates/v0.1.0
```

Set the package repo and we can use kpt to manage

```
/Users/kubelord/k3d/kpt-repo-msql$ export REPO=https://github.com/GoogleContainerTools/kpt.git
```

Use KPT pkg get to sync a repo. This enables us to capture future changes.

```
/Users/kubelord/k3d/kpt-repo-msql$ kpt pkg get $REPO/package-examples/mysql/mysql/templates@v0.16.0 mysql/
```

```
fetching package /package-examples/mysql/mysql/templates from
https://github.com/GoogleContainerTools/kpt to mysql\templates
```

Now that we’ve completed this operation with “kpt pkg get”, the local package can be modified after it is fetched. This allows us to pull in future upstream changes when the upstream package is regenerated from the chart or other modifications are made.