How to make and share your own Helm package

https://miro.medium.com/max/60/1*ANDxSZMbvvhaxwqdI-6rPw.png?q=20



**Create a package**

What is a helm chart? It is basically a set of templates and a file containing variables used to fill these templates. Let’s have a look at an example. I assume that you already have Helm installed and configured at this point.  
To start working on a chart, Helm uses asimple command create:

$ helm create my-app

After that Helm creates a directory with the following layout:

my-app/  
├── charts  
├── Chart.yaml  
├── templates  
│ ├── deployment.yaml  
│ ├── \_helpers.tpl  
│ ├── ingress.yaml  
│ ├── NOTES.txt  
│ └── service.yaml  
└── values.yaml2 directories, 7 files

It has charts directory with chart dependencies, but we don’t need it at the moment. Next comes Chart.yaml containing global variables for the chart such as version and description:

$ cat my-app/Chart.yaml   
apiVersion: v1  
appVersion: "1.0"  
description: A Helm chart for Kubernetes  
name: my-app  
version: 0.1.0

Then comes templates directory – there you put all the \*.yaml files for Kubernetes. Helm uses Go template markup language to customize these files. Helm creates three default file types: deployment, service and ingress. All the files in this directory are ‘skeletons’ which are filled with the variables from values.yamlwhen you deploy your Helm chart. File \_helpers.tpl contains your custom helper functions for variable calculaton.

By default helm creates an nginx deployment. Let’s customize it a bit. Add new ConfigMap to the templates directory:

$ cat << HELM > my-app/templates/cm.yaml  
apiVersion: v1  
data:  
 nginx.conf: |  
 events {  
 worker\_connections 1024;  
 }  
 http {  
 server {  
 listen 80;  
 location / {  
 return 200 "===============================\n\n This is your helm deploy! \n\n===============================\n";  
 }  
 }  
 }  
kind: ConfigMap  
metadata:  
 name: nginx-config  
HELM

Point our nginx Deployment to that ConfigMap. Add the following lines to the deployment.yaml:

volumes:  
 - name: config  
 configMap:  
 name: nginx-config

and:

volumeMounts:  
 - name: config  
 mountPath: /etc/nginx/nginx.conf  
 subPath: nginx.conf

That’s it! Let’s check if we are doing the right thing:

helm template my-app

This will generate all templates with variables and show the output. Now that we know everything is OK, we can deploy the chart:

helm install my-app --name=my-app-name

Then check that Service and Deploy have been created and curl our Service:

$ kubectl get svc  
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
my-app-name-my-app ClusterIP 10.100.56.254 <none> 80/TCP 44m  
$ curl 10.100.56.254  
=============================== This is your helm deploy! ===============================

Congratulations! We have created and deployed our first Helm chart.  
Additionally, you can create a package:

$ helm package my-app

This command creates an archive like my-app-0.1.0.tgz — now you can share your chart with others. For instance, you can upload this file to Helm repository, which we are going to do now.

**Create Helm repo and publish your chart**

Helm repo is an HTTP server that has file index.yaml and all your chart files. You can use any http-server, but the easiest way to do that is to use GitHub pages.  
First, create a GitHub repo, clone it locally and create a branch (note: it should be namedch-pages) for our charts (I will be using the repo called gree-gorey/helm-example):

$ git clone https://github.com/gree-gorey/helm-example.git  
$ cd helm-example  
$ git checkout -b gh-pages

Then create an empty file and push it to the repo:

$ touch index.yaml  
$ git add index.yaml  
$ git commit -a -m "add index.yaml"  
$ git push --set-upstream origin gh-pages

Then go to [github.com](http://github.com/) to your repo settings and scroll down to “GitHub pages” section. Then choose gh-pages branch for the source. Copy the link above to the clipboard. Mine is <https://gree-gorey.github.io/helm-example/>

Now we are going to add our chart to that repo:

$ helm package my-app  
$ mv my-app-0.1.0.tgz helm-example  
$ helm repo index helm-example/ --url <https://gree-gorey.github.io/helm-example/>

The last command generates index.yaml file. Let’s take a look at it:

$ cat helm-example/index.yaml   
apiVersion: v1  
entries:  
 my-app:  
 - apiVersion: v1  
 appVersion: "1.0"  
 created: 2018-03-30T14:00:56.531328411Z  
 description: A Helm chart for Kubernetes  
 digest: 29089aabaa8aa08a03215098b1982ad38f6cd9de1c9b25ff842003a53cad881d  
 name: my-app  
 urls:  
 - https://gree-gorey.github.io/helm-example/my-app-0.1.0.tgz  
 version: 0.1.0  
generated: 2018-03-30T14:00:56.530846921Z

Now commit & push the changes:

$ git commit -a -m "change index"  
$ git push origin

Check that your server is serving index.yaml:

$ curl https://gree-gorey.github.io/helm-example/index.yaml  
apiVersion: v1  
entries:  
 my-app:  
 - apiVersion: v1  
 appVersion: "1.0"  
 created: 2018-03-30T14:00:56.531328411Z  
 description: A Helm chart for Kubernetes  
 digest: 29089aabaa8aa08a03215098b1982ad38f6cd9de1c9b25ff842003a53cad881d  
 name: my-app  
 urls:  
 - https://gree-gorey.github.io/helm-example/my-app-0.1.0.tgz  
 version: 0.1.0  
generated: 2018-03-30T14:00:56.530846921Z

OK, we did it! Now we can add this repo to another Helm installation:

$ helm repo add helm-example https://gree-gorey.github.io/helm-example  
$ helm repo list  
NAME URL   
helm-example <https://gree-gorey.github.io/helm-example>

Now check it by creating a new deploy from the repo:

$ helm install helm-example/my-app --name=my-app-name