

# HELM

## About Helm

helm is third party kubernetes package installer and it will work like a linux yum repo, better to learn helm because in cloud native foundation helm is keep getting developed.

in ansible how playbook will be written, and roles will be defined and placed on ansible-galaxy which will be work as repo and everyone can use it and download it from repo.

helm two components are there

- 1, server components – helm
- 2, client components -- tiller.

helm will be used for version revision and shareable.

## Actual Readme

Helm is a package manager for Kubernetes that allows developers and operators to more easily package, configure, and deploy applications and services onto Kubernetes clusters.

### An Overview of Helm

Most every programming language and operating system has its own package manager to help with the installation and maintenance of software. Helm provides the same basic feature set as many of the package managers you may already be familiar with, such as Debian's apt, or Python's pip.

### Helm can:

Install software.

Automatically install software dependencies.

Upgrade software.

Configure software deployments.

Fetch software packages from repositories.

Helm provides this functionality through the following components:

A command line tool, helm, which provides the user interface to all Helm functionality.

A companion server component, tiller, that runs on your Kubernetes cluster, listens for commands from helm, and handles the configuration and deployment of software releases on the cluster.

The Helm packaging format, called charts.

### PREREQUISITES

The following prerequisites are required for a successful and properly secured use of Helm.

A Kubernetes clusters Deciding what security configurations to apply to your installation, if any Installing and configuring Helm and Tiller, the cluster-side service. #kubectl cluster-info

### ##Install helm by following below steps##

#curl -LO [https://git.io/get\\_helm.sh](https://git.io/get_helm.sh)

```
[root@anskube manifest]# curl -LO https://git.io/get_helm.sh
% Total    % Received % Xferd  Average Speed   Time    Time     Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
100 7034 100 7034    0     0 19091  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[root@anskube manifest]#
```

# chmod 700 get\_helm.sh

# ./get\_helm.sh

```
[root@anskube manifest]# ./get_helm.sh
Downloading https://get.helm.sh/helm-v2.16.0-linux-amd64.tar.gz
Preparing to install helm and tiller into /usr/local/bin
helm installed into /usr/local/bin/helm
tiller installed into /usr/local/bin/tiller
Run 'helm init' to configure helm.
[root@anskube manifest]#
```

# helm version

```
[root@ansikube manifest]# helm version
Client: &version.Version{SemVer:"v2.16.0", GitCommit:"e13bc94621d4ef666270cfbe734aaabf342a49bb", GitTreeState:"clean"}
Error: could not find tiller
```

Note: - ignore above error because still tiller is not installed.

##Create a role base access control##

# helm init --service-account tiller --history-max 200

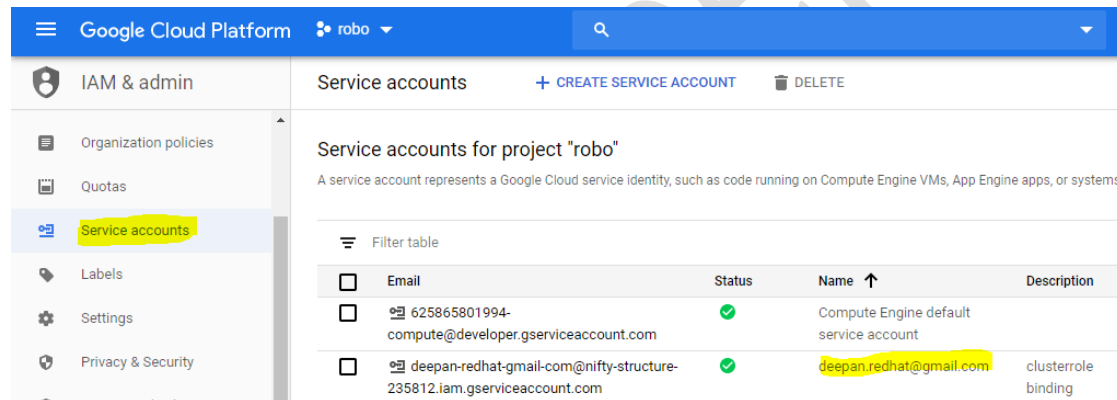
```
[root@ansikube manifest]# helm init --service-account tiller --history-max 200
Creating /root/.helm
Creating /root/.helm/repository
Creating /root/.helm/repository/cache
Creating /root/.helm/repository/local
Creating /root/.helm/plugins
Creating /root/.helm/starters
Creating /root/.helm/cache/archive
Creating /root/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$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
[root@ansikube manifest]#
```

@CREATE SERVICE ACCOUNT on GCP (Add your ID)

# Go to GCP --> IAM & admin --> Service accounts --> Create service account → GIVE YOUR id → select kubernetes engine → Kubernetes engine admin & Kubernetes engine cluster admin



The screenshot shows the Google Cloud Platform IAM & admin console. The left sidebar lists navigation options: Organization policies, Quotas, Service accounts (highlighted), Labels, Settings, and Privacy & Security. The main panel displays 'Service accounts for project "robo"'. Below this, a table lists service accounts:

	Email	Status	Name	Description
<input type="checkbox"/>	625865801994-compute@developer.gserviceaccount.com	✓	Compute Engine default service account	
<input type="checkbox"/>	deepan-redhat-gmail-com@nifty-structure-235812.iam.gserviceaccount.com	✓	deepan.redhat@gmail.com	clusterrole binding

# gcloud auth login

```
You are now logged in as [deepan.redhat@gmail.com].
Your current project is [nifty-structure-235812]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
[root@ansikube manifest]#
```

#export GCP\_USER=\$(gcloud config get-value account | head -n 1)

#kubectl create clusterrolebinding cluster-admin-binding --clusterrole=cluster-admin --user=\$GCP\_USER

```
[root@ansikube manifest]# export GCP_USER=$(gcloud config get-value account | head -n 1)
[root@ansikube manifest]# kubectl create clusterrolebinding cluster-admin-binding --clusterrole=cluster-admin --user=$GCP_USER
clusterrolebinding.rbac.authorization.k8s.io/cluster-admin-binding created
[root@ansikube manifest]#
```

Note: - Kindly use your GCP ID.

```
# cat helm-rbac.yml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: tiller
  namespace: kube-system
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: tiller
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: tiller
  namespace: kube-system
```

```
# kubectl apply -f helm-rbac.yml
# kubectl get ClusterRoleBinding |grep tiller
[root@anskube manifest]# kubectl get ClusterRoleBinding |grep tiller
tiller                                42s
[root@anskube manifest]#
```

```
# helm init --service-account tiller --history-max 200
# kubectl get pods --namespace kube-system |grep tiller
[root@anskube manifest]# kubectl get pods --namespace kube-system |grep tiller
tiller-deploy-695779d66-v17dp        1/1      Running    0          48s
[root@anskube manifest]#
```

```
# helm list
# helm search mysql
```

@Let's download and install the percona package for practice purpose

```
# helm search percona
[root@anskube manifest]# helm search percona
NAME                CHART VERSION  APP VERSION  DESCRIPTION
stable/percona      1.2.0         5.7.17      free, fully compatible, enhanced, open source drop-in rep...
stable/percona-xtradb-cluster 1.0.3         5.7.19      free, fully compatible, enhanced, open source drop-in rep...
```

```
# helm fetch stable/percona
# ls |grep percona
[root@anskube manifest]# ls |grep percona
percona-1.2.0.tgz
[root@anskube manifest]#
# tar -xzf percona-1.2.0.tgz && cd percona && ls
Chart.yaml  README.md  templates  values.yaml
[root@anskube percona]# pwd
/root/manifest/percona
[root@anskube percona]#
```

Note: - View values.yaml, it will have all objects and RESOURCES info. this is the major benefit of helm, we no need configure resource/objects for any apps/db image. Its already customised by someone and kept it on helm repo for public access.

# helm install percona

```
[root@anskube manifest]# helm install percona
NAME: agile-manta
LAST DEPLOYED: Fri Nov 8 03:48:20 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                                AGE
agile-manta-percona-5bb799f89d-vtmxn 0s

==> v1/Secret
NAME                                AGE
agile-manta-percona                  0s

==> v1/Service
NAME                                AGE
agile-manta-percona                  0s

==> v1beta1/Deployment
NAME                                AGE
agile-manta-percona                  0s
```

# kubectl get pods

```
[root@anskube manifest]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
agile-manta-percona-5bb799f89d-vtmxn 1/1     Running   0           118s
```

# helm list

```
[root@anskube manifest]# helm list
NAME          REVISION   UPDATED              STATUS   CHART          APP VERSION   NAMESPACE
agile-manta   1          Fri Nov 8 03:48:20 2019    DEPLOYED percona-1.2.0  5.7.17        default
[root@anskube manifest]#
```

@To check the details of installed package.

# helm list

# helm status agile-manta

@To check the repo.

# helm repo list

```
[root@anskube manifest]# helm repo list
NAME    URL
stable  https://kubernetes-charts.storage.googleapis.com
local   http://127.0.0.1:8879/charts
[root@anskube manifest]#
```

@Install package directly.

# helm install stable/mysql

```
[root@anskube manifest]# helm install stable/mysql
NAME: opining-cheetah
LAST DEPLOYED: Fri Nov 8 03:53:29 2019
NAMESPACE: default
STATUS: DEPLOYED
```

# helm list

```
[root@anskube manifest]# helm list
NAME          REVISION   UPDATED              STATUS   CHART          APP VERSION   NAMESPACE
agile-manta   1          Fri Nov 8 03:48:20 2019    DEPLOYED percona-1.2.0  5.7.17        default
opining-cheetah 1          Fri Nov 8 03:53:29 2019    DEPLOYED mysql-1.4.0    5.7.27        default
```

Note1: - Two way we can install, download package and install or direct install.

@To delete the installed package.

# helm delete agile-manta

# helm delete opining-cheetah

##Create own folder with all objects and install with helm##

# cat Chart.yaml

```
Name: webserver
Description: Nginx Webserver
apiVersion: v1
Version: 1.0
appVersion: 1.0
Maintainer:
  - name: deepan
    mail: deepan.redhat@gmail.com
```

# mkdir templates

# cat templates/deployments.yml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: deploy1
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      name: nginx
      labels:
        app: nginx
    spec:
      containers:
        - name: con1
          image: nginx:1.15
          ports:
            - containerPort: 80
```

# cat templates/service.yml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  type: NodePort
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
      nodePort: 32001
```

# cat templates/ingress.yml

```
kind: Ingress
apiVersion: extensions/v1beta1
metadata:
  name: nginx-ingress
spec:
  backend:
    serviceName: nginx-service
    servicePort: 8080
```

@To check syntax error.

# helm lint webserver

```
[root@anskube manifest]# helm lint webserver
==> Linting webserver
[INFO] Chart.yaml: icon is recommended
[INFO] values.yaml: file does not exist

1 chart(s) linted, no failures
[root@anskube manifest]#
```

# helm install webserver

# helm status moldy-kangaroo

```
[root@anskube manifest]# helm status moldy-kangaroo
LAST DEPLOYED: Fri Nov  8 05:47:16 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                                     AGE
deploy1-5ff46bf668-8kxv4               80s
deploy1-5ff46bf668-jnvkt               80s
deploy1-5ff46bf668-pzdlx               80s

==> v1/Service
NAME          AGE
nginx-service 81s

==> v1beta1/Deployment
NAME      AGE
deploy1   81s

==> v1beta1/Ingress
NAME          AGE
nginx-ingress 81s
```

Note: - just wait for ingress ip to create, then able to access nginx from outside.

##Create own folder with all object by using values.yml##

#mkdir webserver1 && cd webserver1

# cat Chart.yaml

```
Name: webserver1
Description: Nginx Webserver
apiVersion: v1
Version: 1.0
appVersion: 1.2
Maintainer:
  - name: Deepan
    mail: deepan.redhat@gmail.com
```

# cat values.yaml

```
scale: 2
tag: "1.17"
```

# mkdir templates

```
# cat templates/deployments.yml
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: deploy2
spec:
  replicas: {{.Values.scale}}
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      name: nginx
      labels:
        app: nginx
    spec:
      containers:
        - name: con1
          image: nginx:{{ .Values.tag }}
          ports:
            - containerPort: 80
```

```
# cat templates/service.yml
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service1
spec:
  selector:
    app: nginx
  type: NodePort
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
      nodePort: 32010
```

```
# cat templates/ingress.yml
```

```
kind: Ingress
apiVersion: extensions/v1beta1
metadata:
  name: nginx-ingress1
spec:
  backend:
    serviceName: nginx-service1
    servicePort: 8080
```

```
# helm lint webserver1
```

```
[root@ansikube manifest]# helm lint webserver1
==> Linting webserver1
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, no failures
[root@ansikube manifest]#
```

```
# helm install webserver1
```

```
# helm list
```

```
[root@ansikube manifest]# helm list
```

NAME	REVISION	UPDATED	STATUS	CHART	APP VERSION	NAMESPACE
moldy-kangaroo	1	Fri Nov 8 05:47:16 2019	DEPLOYED	webserver-1	1	default
nosy-platypus	1	Fri Nov 8 06:19:15 2019	DEPLOYED	webserver1-1	1.2	default

```
[root@ansikube manifest]#
```

```
# helm status nosy-platypus
```

```
[root@anskube manifest]# helm status nosy-platypus
LAST DEPLOYED: Fri Nov  8 06:19:15 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                                     AGE
deploy1-5ff46bf668-8kxv4               33m
deploy1-5ff46bf668-jnvkt               33m
deploy1-5ff46bf668-pzdlx               33m
deploy2-6bfbbbfb58-gzq6j               100s
deploy2-6bfbbbfb58-mrzs5               100s

==> v1/Service
NAME          AGE
nginx-service1 100s

==> v1beta1/Deployment
NAME          AGE
deploy2       100s

==> v1beta1/Ingress
NAME          AGE
nginx-ingress1 100s
```

Note: - just wait for ingress ip to create, then able to access nginx from outside.

@Try for nginx version upgrade by using helm.

: - change the appVersion : 1.2

```
# kubectl get pods
```

deploy2-67555d454c-ghcz5	1/1	Running	0	3m41s
deploy2-67555d454c-nk24g	1/1	Running	0	3m36s
deploy2-67555d454c-vj5pf	1/1	Running	0	3m46s
deploy2-67555d454c-xj67z	1/1	Running	0	3m46s

```
# kubectl exec -it deploy2-67555d454c-vj5pf -- nginx -v
```

```
[root@anskube webserver1]# kubectl exec -it deploy2-67555d454c-vj5pf -- nginx -v
nginx version: nginx/1.16.1
```

```
# cat webserver1/Chart.yaml
```

```
[root@anskube manifest]# cat webserver1/Chart.yaml
Name: webserver1
Description: Nginx Webserver
apiVersion: v1
Version: 1.0
appVersion: 1.3
Maintainer:
  - name: Deepan
    mail: deepan.redhat@gmail.com
```

```
# helm list
```

NAME	REVISION	UPDATED	STATUS	CHART	APP VERSION	NAMESPACE
moldy-kangaroo	1	Fri Nov  8 05:47:16 2019	DEPLOYED	webserver-1	1	default
nosy-platypus	1	Fri Nov  8 06:19:15 2019	DEPLOYED	webserver1-1	1.2	default



:- Tag (tag=1.17) is version of nginx.

# helm upgrade --set scale=4,tag=1.17 nosy-platypus webserver1

```
[root@anskube manifest]# helm upgrade --set scale=4,tag=1.17 nosy-platypus webserver1
Release "nosy-platypus" has been upgraded.
LAST DEPLOYED: Fri Nov 8 06:47:30 2019
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                                     AGE
deploy1-5ff46bf668-8kxv4               60m
deploy1-5ff46bf668-jnvkt               60m
deploy1-5ff46bf668-pzdlx               60m
deploy2-67555d454c-ghcz5               5m27s
deploy2-67555d454c-nk24g               5m22s
deploy2-67555d454c-vj5pf               5m32s
deploy2-67555d454c-xj67z               5m32s
deploy2-6bfbbbf58-wqmz7                1s

==> v1/Service
NAME          AGE
nginx-service1 28m

==> v1beta1/Deployment
NAME          AGE
deploy2       28m

==> v1beta1/Ingress
NAME          AGE
nginx-ingress1 28m
```

# kubectl get pods

```
[root@anskube manifest]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
deploy1-5ff46bf668-8kxv4           1/1     Running   0           51m
deploy1-5ff46bf668-jnvkt           1/1     Running   0           51m
deploy1-5ff46bf668-pzdlx           1/1     Running   0           51m
deploy2-6bfbbbf58-gzq6j            1/1     Running   0           19m
deploy2-6bfbbbf58-mrzs5            1/1     Running   0           19m
[root@anskube manifest]#
```

# helm list

```
[root@anskube manifest]# helm list
NAME          REVISION    UPDATED              STATUS      CHART          APP VERSION    NAMESPACE
moldy-kangaroo 2            Fri Nov 8 06:36:31 2019    DEPLOYED   webserver-1    1.2            default
nosy-platypus 3            Fri Nov 8 06:47:30 2019    DEPLOYED   webserver1-1   1.3            default
[root@anskube manifest]#
```

# kubectl exec -it deploy2-6bfbbbf58-488qt -- nginx -v

```
[root@anskube manifest]# kubectl exec -it deploy2-6bfbbbf58-488qt -- nginx -v
nginx version: nginx/1.17.5
[root@anskube manifest]#
```

@Try for rollback.

This command needs two arguments: **release name, revision number**

# helm list

```
[root@anskube manifest]# helm list
NAME          REVISION    UPDATED              STATUS      CHART          APP VERSION    NAMESPACE
moldy-kangaroo 2            Fri Nov 8 06:36:31 2019    DEPLOYED   webserver-1    1.2            default
nosy-platypus 3            Fri Nov 8 06:47:30 2019    DEPLOYED   webserver1-1   1.3            default
[root@anskube manifest]#
```

# helm rollback nosy-platypus 2 && helm list

# kubectl get pods

# kubectl exec -it deploy2-6bfbbbf58-b8fw2 -- nginx -v

```
[root@anskube manifest]# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
deploy1-5ff46bf668-8kxv4           1/1     Running   0           67m
deploy1-5ff46bf668-jnvkt           1/1     Running   0           67m
deploy1-5ff46bf668-pzdlx           1/1     Running   0           67m
deploy2-67555d454c-76pj8           1/1     Running   0           46s
deploy2-67555d454c-gzvn1           1/1     Running   0           48s
deploy2-67555d454c-jcdps           1/1     Running   0           48s
deploy2-67555d454c-s2wt7           1/1     Running   0           46s
[root@anskube manifest]# kubectl exec -it deploy2-67555d454c-76pj8 -- nginx -v
nginx version: nginx/1.16.1
[root@anskube manifest]#
```