**How To Install and Configure Helm**



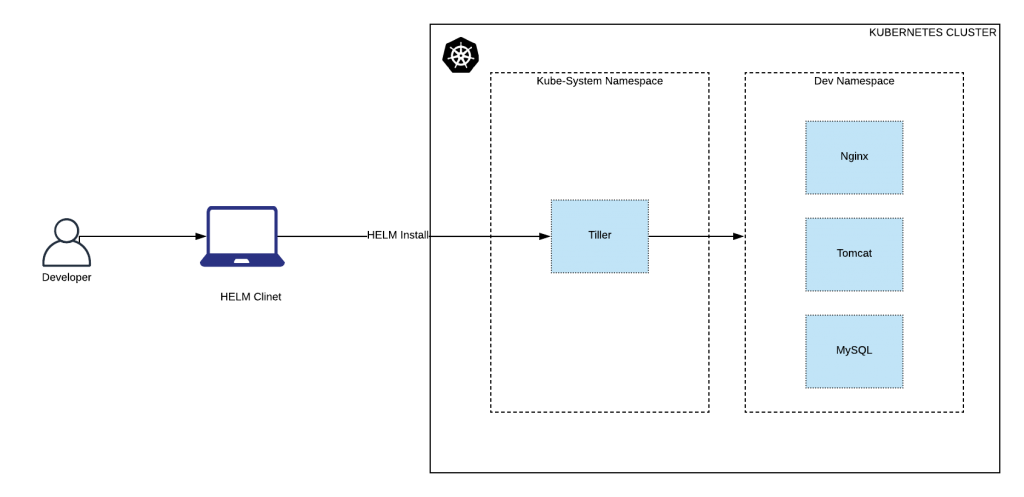
*This post explains how to install and configure helm for deploying and managing applications on a Kubernetes cluster.*

**Prerequisites**

You should have the following before getting started with the helm setup.

1. A running Kubernetes cluster.
2. The Kubernetes cluster API endpoint should be reachable from the machine you are running helm.
3. Authenticate the cluster using kubectl and it should have cluster admin permissions.

Helm Architecture



**Installing helm [Client]**

This installation is on the client side. ie, a personal workstation, a Linux VM, etc. You can install helm using a single liner. It will automatically find your OS type and installs helm on it.

Execute the following from your command line.



|  |  |
| --- | --- |
| 1 | curl -L https://git.io/get\_helm.sh | bash |

**Create Tiller Service Account With Cluster Admin Permissions**

Tiller is the server component for helm. Tiller will be present in the kubernetes cluster and the helm client talks to it for deploying applications using helm charts.

Helm will be managing your cluster resources. So we need to add necessary permissions to the tiller components which resides in the cluster kube-system namespace.

Here is what we will do,

1. Create a service account names tiller
2. Create a ClusterRoleBinding with cluster-admin permissions to the tiller service account.

We will add both service account and clusterRoleBinding in one yaml file.

Create a file named helm-rbac.yaml and copy the following contents to the file.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | 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 |

Lets create these resources using kubectl



|  |  |
| --- | --- |
| 1 | kubectl apply -f helm-rbac.yam |

**Initialize Helm: Deploy Tiller**

Next step is to initialize helm. When you initialize helm, a deployment named tiller-deploy will be deployed in the kube-system namespace.

Initialize helm using the following command.



|  |  |
| --- | --- |
| 1 | helm init --service-account=tiller --history-max 300 |

If you want a specific tiller version to be installed, you can specify the tiller image link in the init command using --tiller-image flag. You can find the all tiller docker images in [public google GCR registry.](https://console.cloud.google.com/gcr/images/kubernetes-helm/GLOBAL/tiller?gcrImageListsize=30)



|  |  |
| --- | --- |
| 1 | helm init --service-account=tiller --tiller-image=gcr.io/kubernetes-helm/tiller:v2.14.1   --history-max 300 |

If you dont mention “–service-account=tiller”, you will get the following error.



|  |  |
| --- | --- |
| 1 | Error: no available release name found |

You can check the tiller deployment in the kube-system namespace using kubectl.



|  |  |
| --- | --- |
| 1 | kubectl get deployment tiller-deploy -n kube-system |

**Deploy a Sample App Using Helm**

Now lets deploy a sample nginx ingress using helm.

Execute the following helm install command to deploy an nginx ingress in the kubernetes cluster. It will download the nginx-ingress helm chart from the [public github helm chart repo](https://github.com/helm/charts/tree/master/stable).



|  |  |
| --- | --- |
| 1 | helm install stable/nginx-ingress --name nginx-ingress |

You can check the install helm chart using the following command.



|  |  |
| --- | --- |
| 1 | helm ls |

You can delete the sample deployment using delete command. For example,



|  |  |
| --- | --- |
| 1 | helm delete nginx-ingress |

**Remove Helm (Tiller) From Kubernetes Cluster**

If you want to remove the tiller installtion from the kubernetes cluster use the following command.



|  |  |
| --- | --- |
| 1 | helm reset |

For some reason, if it throws error, force remove it using the following command.



|  |  |
| --- | --- |
| 1 | helm reset --force |

Also you can use the kubectl command to remove it.



|  |  |
| --- | --- |
| 1 | kubectl delete deployment tiller-deploy --namespace kube-system |