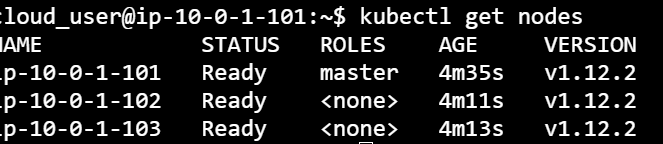
Install Helm and Tiller in the Existing Cluster



Download the helm binary release:

curl https://storage.googleapis.com/kubernetes-helm/helm-v2.12.3-linux-amd64.tar.gz > ./helm.tar.gz

If we run a quick ls, we'll see that new helm.tar.gz sitting in our directory. Extract the archive:

tar -xvf ./helm.tar.gz

Now we can navigate to the linux-amd64 directory:

cd linux-amd64

If we run another ls, we'll see two executable files, helm and tiller. Let's move them to the /usr/local/bin directory:

sudo mv ./helm /usr/local/bin

sudo mv ./tiller/usr/local/bin

To ensure that the helm command is available, let's get back into our home directory and run the helm version command:

cd ~

helm version

We get a version, meaning Helm is good to go, but we get an error that Helm couldn't find Tiller. So we can install it with this:

helm init

Wait for a bit for things to start up, then run the version command again to ensure that Tiller is available:

helm version

Now we can see that there's a Tiller server.

Deploy the Example nginx Chart

Ensure that you are in the cloud\_user home directory. An ls should show a directory named nginx that contains the chart we need:

cd ~

ls

ls nginx

This last command should return a directory listing containing Chart.yaml. Install the nginx chart with this:

helm install ./nginx

This command should error with a "name" error, due to the tiller service account missing.

Correct Errors

Our problem is that missing service account, so let's create one:

kubectl create serviceaccount --namespace kube-system tiller

Now we'll create a Tiller cluster rule:

kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller

Let's patch the Tiller deployment:

kubectl patch deploy --namespace kube-system tiller-deploy -p'{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'

If we wait for the tiller pod to restart with the update, we can try installing the nginx example chart again:

helm install ./nginx

Verify That the Container is Running

Once the release has completed and we see the release output, let's locate the cluster IP for the service that has been created. It was in the output of the helm install command. We're going to need that IP to confirm that the nginx pod has been deployed correctly:

curl <CLUSTER-IP>:8888

This should return an HTML message:

<h1>Hello</h1> <p>This is a test</p>

Finishing Up

Conclusion

Well, we've installed Helm from our binary, corrected an issue with the service account, and now we can actually deploy Helm releases.