

[Get Helm](#)[Blog](#)[Docs](#)

# The package manager for Kubernetes

Helm is the best way to find, share, and use software built for [Kubernetes](#).



---

Helm is supported by and built with a community of over 250 developers.

---



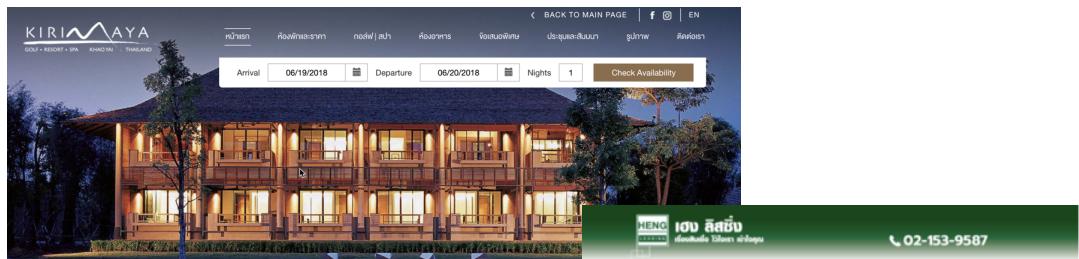
# Who are we ? (Opcellent)



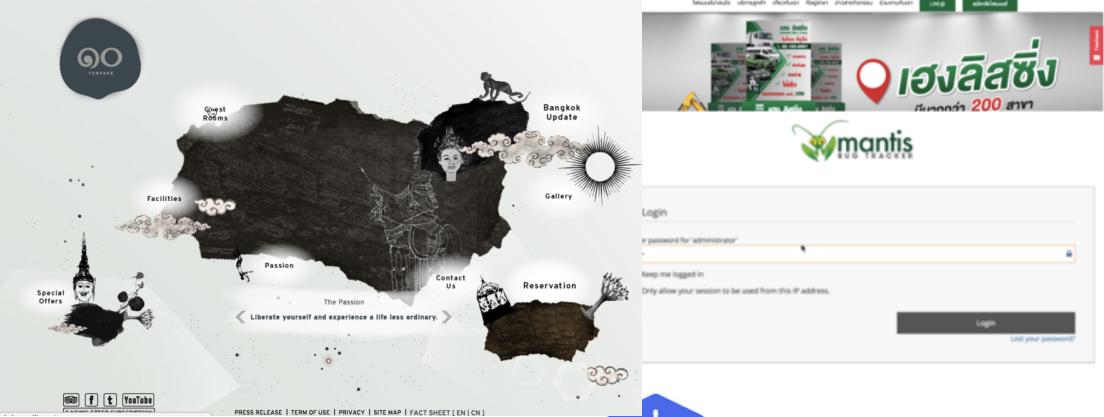
The Opcellent website features a light blue header with the company logo and navigation links for About, Service, Training, and Contact. Below the header is a large teal graphic with a white hand icon. The main content area includes a sub-header "Modern Server Technology Implementer", a "GET DETAILS" button, and a "ABOUT" section with a sub-section "SERVICE".



A screenshot of an event listing for "Advanced Docker and Kubernetes 101" presented by GDG Chiang Mai and Artisan Digital. It shows the event date (Saturday, 12 May 2018), time (6pm - 9pm), location (Punspace Tha Pae Gate), and a note about food and beer. There are also "Interested" and "Going" buttons.



A screenshot of the Kirinaya Golf & Resort Spa Thailand website. It shows a search interface for room availability with fields for arrival (06/19/2018), departure (06/20/2018), nights (1), and a "Check Availability" button. Below the form is a night view of the resort's buildings.



A screenshot of the Mantis Sod Tracker website. It features a map of Thailand with various locations marked and labeled, such as "Guest R80ms", "Facilities", "Passion", "The Passion", "Contact Us", "Reservation", "Bangkok Update", "Gallery", and "Special Offers". To the right is a login form with fields for "username" and "password", and checkboxes for "Keep me logged in" and "Only allow your session to be used from this IP address".

Present by: Praparn L. (eva10409@gmail.com)



**kubernetes**  
by Google

# Agenda

- What is Helm ? And Why Helm ?
- Helm Component
- Chart and Repository
- Workshop



# Lab Resource

- Download on Google Drive
  - <https://goo.gl/oCUayz>
- Download on GitHub
  - git@github.com:praparn/kubernetes\_20170128.git

No description, website, or topics provided.

Branch: master ▾ New pull request

Clone with SSH Use HTTPS  
git@github.com:praparn/kuberneteslab.git

File	Commit	Date
praparn 2017110214055	2017110214055	an hour ago
WorkShop_1.1_Install_Kubernetes	2017110214055	an hour ago
WorkShop_1.2_Pods_Service_Deployment	2017110214055	an hour ago
WorkShop_1.3_Replication_Controller	2017110214055	an hour ago
WorkShop_1.4_Deployment	2017110214055	an hour ago
WorkShop_1.5_Volume	2017110214055	an hour ago
WorkShop_1.6_Liveness_Readiness_Probe	2017110214055	an hour ago
WorkShop_1.7_Resource_Management_and_HPA	2017110214055	an hour ago
WorkShop_2.1_ConfigMap_Secret	2017110214055	an hour ago
WorkShop_2.3_Log_and_Monitoring	2017110214055	an hour ago
WorkShop_2.4_Ingress_Network	2017110214055	an hour ago
WorkShop_2.5_Kubernetes_RealWorld	2017110214055	an hour ago
WorkShop_2.6_Orchestrator_Assignment	20171108235628	2 days ago
WorkShop_2.7_Persistent_Storage	20171110	23 hours ago
Workshop_2.2_Job_CronJob	2017110214055	an hour ago
.DS_Store	2017110003943	22 hours ago
Kubernetes_Training_master.pdf	2017110003943	22 hours ago
Untitled-1	2017110214055	an hour ago

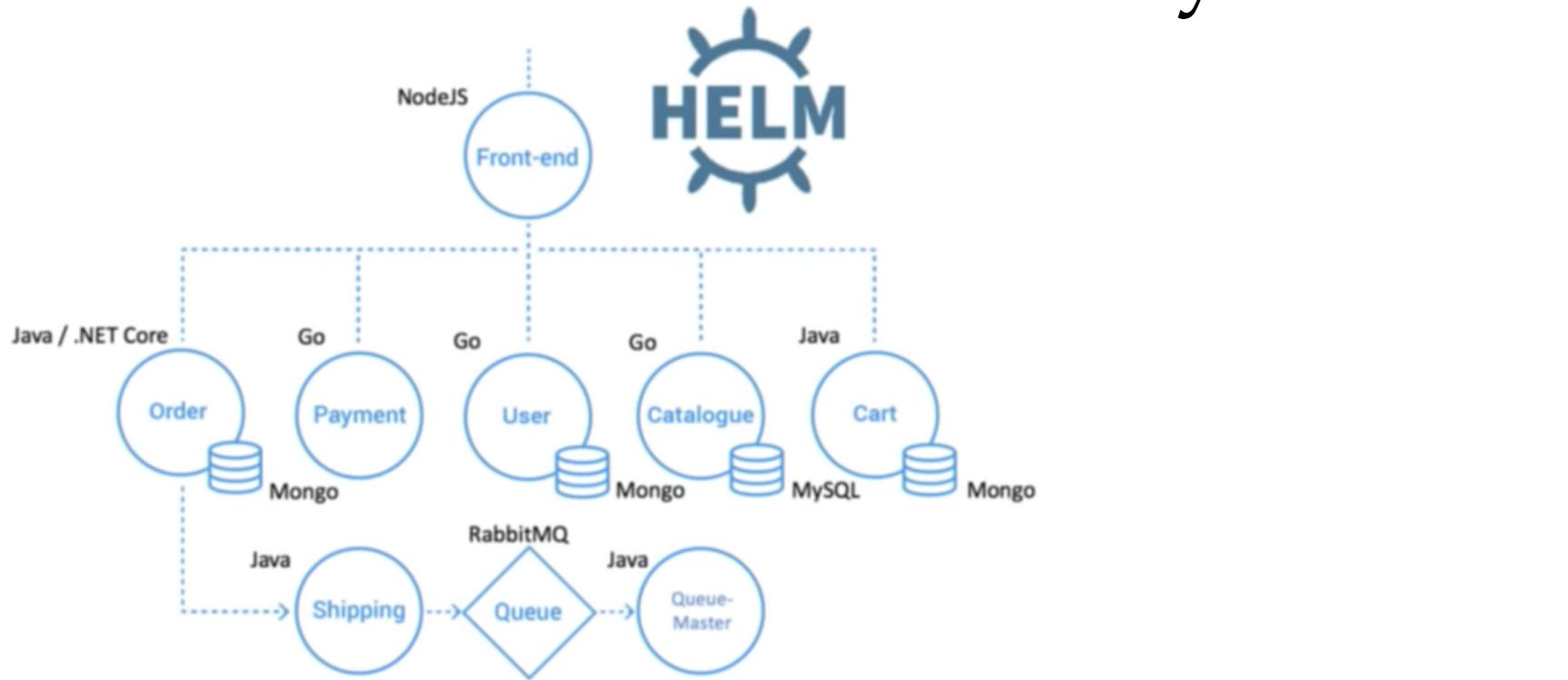


# What is Helm ? And Why Helm ?

- Helm is tools for manage package in K8S
  - Helm (Chart) will all-in-one build tool for support kubernetes
  - Many microservice in realworld
    - Front-end application
    - Back-end application
      - Microservice set A
      - Microservice set B
      - Microservice set C
    - Queuing system management
    - Etc
  - Each microservice need to write-down to kubernetes platform
    - Pods / Daemon Set / Replication Controller / Deployment
    - Service / Ingress / External Load Balance
    - etc



# What is Helm ? And Why Helm ?



## Manage Complexity

Charts describe even the most complex apps; provide repeatable application installation, and serve as a single point of authority.

## Easy Updates

Take the pain out of updates with in-place upgrades and custom hooks.

## Simple Sharing

Charts are easy to version, share, and host on public or private servers.

## Rollbacks

Use `helm rollback` to roll back to an older version of a release with ease.

Ref: [https://www.youtube.com/watch?v=Jj1Ueq\\_Lz6A](https://www.youtube.com/watch?v=Jj1Ueq_Lz6A)

Kubernetes: Production Workload Orchestration



**kubernetes**  
by Google

# What is Helm ? And Why Helm ?

IBM

IBM Cloud Blog Why IBM Products Solutions Garage Pricing Blogs Docs Support

Compute Services

## Speed deployment on Kubernetes with Helm Chart – Quick YAML example from scratch

October 25, 2017 | Written by: [Rick Osowski](#)

Categorized: [Compute Services](#) | [How-tos](#) | [Hybrid Deployments](#)

Share this post:



Are you working with Kubernetes, with all the recent supporting releases on [IBM Cloud Private](#), the [IBM Cloud Container Service](#) on IBM Cloud Platform, or elsewhere? Are you in the middle of containerizing workloads across your portfolio? Have you adopted Kubernetes and looking to speed up deployment and reuse? Or are you simply looking to see how far this whole nautical theme is going to go?

Either way, understanding all the tools at your disposal is a critical step to success here. One of those most advantageous tools is [Helm](#) and [Helm Charts](#). Many projects took to packaging their releases in Docker containers immediately, but that's only one step... the runtime. You still need to know how to connect the dots with all the supporting services, peer containers, and operational characteristics of the project itself. Helm Charts allow you to do just that, with strict templates applied to Kubernetes configuration YAML files, providing the ability to build, package, distribute, and deploy complex containerized applications through simple helm CLI commands.



Kubernetes: Production Workload Orchestration



kubernetes  
by Google

# What is Helm ? And Why Helm ?

IBM

IBM Cloud Blog Why IBM Products Solutions Garage Pricing Blogs Docs Support

Compute Services

## Speed deployment on Kubernetes with Helm Chart – Quick YAML example from scratch

October 25, 2017 | Written by: [Rick Osowski](#)

Categorized: [Compute Services](#) | [How-tos](#) | [Hybrid Deployments](#)

Share this post:



Are you working with Kubernetes, with all the recent supporting releases on [IBM Cloud Private](#), the [IBM Cloud Container Service](#) on IBM Cloud Platform, or elsewhere? Are you in the middle of containerizing workloads across your portfolio? Have you adopted Kubernetes and looking to speed up deployment and reuse? Or are you simply looking to see how far this whole nautical theme is going to go?

Either way, understanding all the tools at your disposal is a critical step to success here. One of those most advantageous tools is [Helm](#) and [Helm Charts](#). Many projects took to packaging their releases in Docker containers immediately, but that's only one step... the runtime. You still need to know how to connect the dots with all the supporting services, peer containers, and operational characteristics of the project itself. Helm Charts allow you to do just that, with strict templates applied to Kubernetes configuration YAML files, providing the ability to build, package, distribute, and deploy complex containerized applications through simple helm CLI commands.

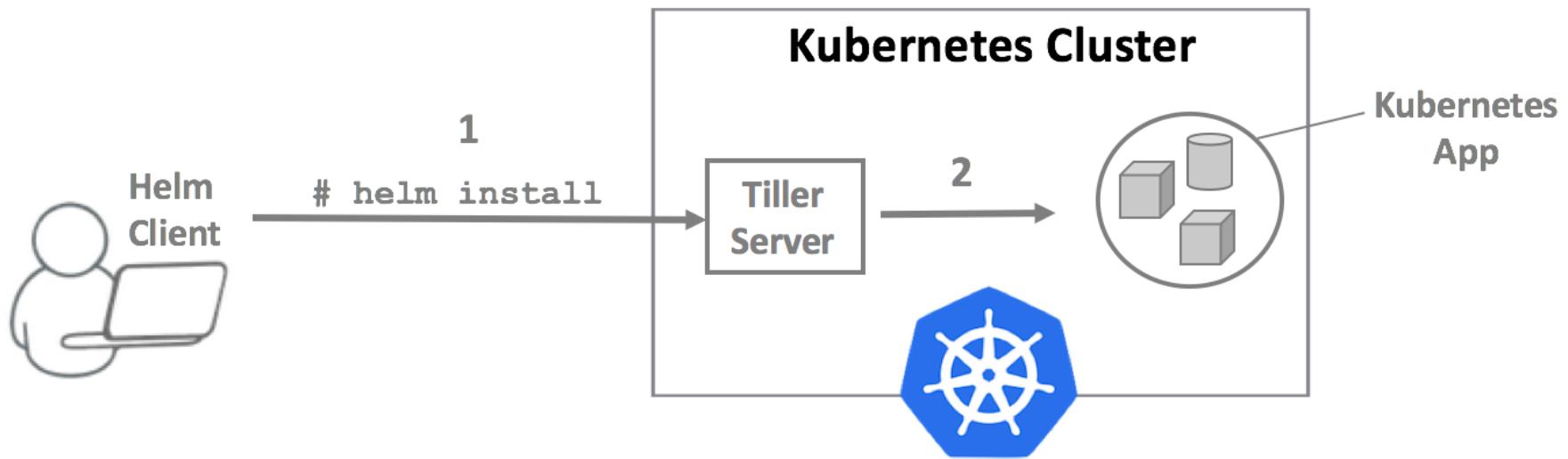


Kubernetes: Production Workload Orchestration



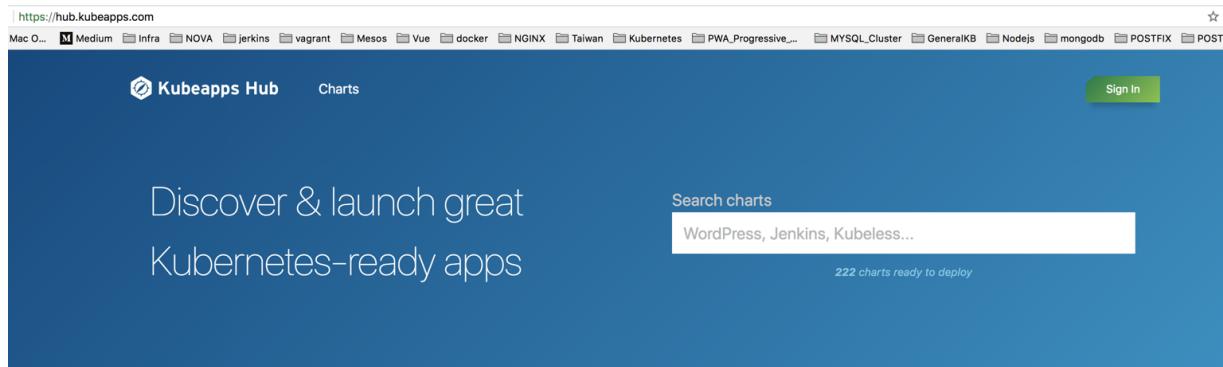
kubernetes  
by Google

# Helm's Component



# Chart and Repository

- Chart is collection for component for K8S
  - Helm (Chart) will all-in-one build tool for support kubernetes
  - Coming with huge repository for install



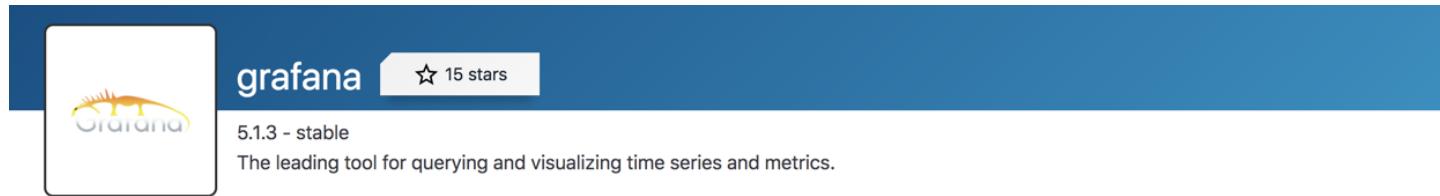
acs-engine-autoscaler ☆ 2      2.1.1 stable	aerospike ☆ 2      v3.14.1.2 stable	anchore ☆ 5      0.1.10 stable	apm-server ☆ 0      6.2.4 stable
ark ☆ 0      0.8.2 stable	artifactory ☆ 23      6.0.0 stable	artifactory-ha ☆ 4      6.0.0 stable	auditbeat ☆ 0      6.2.4 stable
azuremonitor-containers ☆ 0      2.0.0-3 incubator	bitcoind ☆ 2      0.15.1 stable	buildkite ☆ 3      3 stable	burrow ☆ 1      0.17.1 stable

Kubernetes: Production Workload Orchestration



kubernetes  
by Google

# Chart and Repository



The screenshot shows the Grafana Helm Chart page. At the top, there's a logo for Grafana, a star icon indicating 15 stars, and the version 5.1.3 - stable. Below this, a brief description states: "The leading tool for querying and visualizing time series and metrics."

## Grafana Helm Chart

- Installs the web dashboarding system [Grafana](#)

TL;DR;

```
$ helm install stable/grafana
```

## Installing the Chart

To install the chart with the release name `my-release`:

```
$ helm install --name my-release stable/grafana
```

## Uninstalling the Chart

To uninstall/delete the `my-release` deployment:

```
$ helm delete my-release
```

The command removes all the Kubernetes components associated with the chart and deletes the release.

## Install

Add stable repository

```
helm repo add stable https://kubernetes-charts.storage.googleapis.com/
```



Install chart

```
helm install stable/grafana --name my-release
```



[Need Helm?](#)

## Chart Versions

[1.12.0](#) - Jul 10, 2018

[1.11.6](#) - Jun 29, 2018

[1.11.5](#) - Jun 24, 2018

[1.11.4](#) - Jun 22, 2018

[1.11.3](#) - Jun 19, 2018



# Chart and Repository

## Configuration

Parameter	Description	Default
<code>replicas</code>	Number of nodes	1
<code>deploymentStrategy</code>	Deployment strategy	<code>RollingUpdate</code>
<code>image.repository</code>	Image repository	<code>grafana/grafana</code>
<code>image.tag</code>	Image tag. ( Must be >= 5.0.0 ) Possible values listed <a href="#">here</a> .	5.0.4
<code>image.pullPolicy</code>	Image pull policy	<code>IfNotPresent</code>
<code>service.type</code>	Kubernetes service type	<code>ClusterIP</code>
<code>service.port</code>	Kubernetes port where service is exposed	9000
<code>service.annotations</code>	Service annotations	80
<code>service.labels</code>	Custom labels	{}
<code>ingress.enabled</code>	Enables Ingress	false
<code>ingress.annotations</code>	Ingress annotations	{}
<code>ingress.labels</code>	Custom labels	{}
<code>ingress.hosts</code>	Ingress accepted hostnames	[]
<code>ingress.tls</code>	Ingress TLS configuration	[]
<code>resources</code>	CPU/Memory resource requests/limits	{}
<code>nodeSelector</code>	Node labels for pod assignment	{}
<code>tolerations</code>	Toleration labels for pod assignment	[]
<code>affinity</code>	Affinity settings for pod assignment	{}
<code>persistence.enabled</code>	Use persistent volume to store data	false
<code>persistence.size</code>	Size of persistent volume claim	10Gi
<code>persistence.existingClaim</code>	Use an existing PVC to persist data	nil
<code>persistence.storageClassName</code>	Type of persistent volume claim	nil
<code>persistence.accessModes</code>	Persistence access modes	[]

<code>persistence.subPath</code>	Mount a sub directory of the persistent volume if set	...
<code>schedulerName</code>	Alternate scheduler name	nil
<code>env</code>	Extra environment variables passed to pods	{}
<code>envFromSecret</code>	The name of a Kubernetes secret (must be manually created in the same namespace) containing values to be added to the environment	...
<code>extraSecretMounts</code>	Additional grafana server secret mounts	[]
<code>datasources</code>	Configure grafana datasources	{}
<code>dashboardProviders</code>	Configure grafana dashboard providers	{}
<code>dashboards</code>	Dashboards to import	{}
<code>dashboardsConfigMaps</code>	ConfigMaps reference that contains dashboards	{}
<code>grafana.ini</code>	Grafana's primary configuration	{}
<code>ldap.existingSecret</code>	The name of an existing secret containing the <code>ldap.toml</code> file, this must have the key <code>ldap-toml</code> .	...
<code>ldap.config</code>	Grafana's LDAP configuration	...
<code>annotations</code>	Deployment annotations	{}
<code>podAnnotations</code>	Pod annotations	{}
<code>sidecar.dashboards.enabled</code>	Enabled the cluster wide search for dashboards and adds/updates/deletes them in grafana	false
<code>sidecar.dashboards.label</code>	Label that config maps with dashboards should have to be added	false
<code>sidecar.datasources.enabled</code>	Enabled the cluster wide search for datasources and adds/updates/deletes them in grafana	false
<code>sidecar.datasources.label</code>	Label that config maps with datasources should have to be added	false



# WorkShop

- Basic Command
  - Install helm client

```
ubuntu@ip-10-0-1-66:~$ curl https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get > get_helm.sh
  % Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total Spent   Left  Speed
100  6740  100  6740    0     0  14787      0 --:--:-- --:--:-- --:--:-- 14780
ubuntu@ip-10-0-1-66:~$ chmod 700 get_helm.sh
ubuntu@ip-10-0-1-66:~$ ./get_helm.sh
Downloading https://kubernetes-helm.storage.googleapis.com/helm-v2.9.1-linux-amd64.tar.gz
Preparing to install into /usr/local/bin
helm installed into /usr/local/bin/helm
Run 'helm init' to configure helm.
```

- Create clusterrolebinding by tiller

```
ubuntu@ip-10-0-1-66:~$ kubectl create clusterrolebinding tiller-cluster-admin \
>   --clusterrole=cluster-admin \
[>   --serviceaccount=kube-system:default
clusterrolebinding "tiller-cluster-admin" created
```



# WorkShop

- Basic Command
  - Initial helm by command: helm init

```
[ubuntu@ip-10-0-1-66:~$ helm init
Creating /home/ubuntu/.helm
Creating /home/ubuntu/.helm/repository
Creating /home/ubuntu/.helm/repository/cache
Creating /home/ubuntu/.helm/repository/local
Creating /home/ubuntu/.helm/plugins
Creating /home/ubuntu/.helm/starters
Creating /home/ubuntu/.helm/cache/archive
Creating /home/ubuntu/.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    [I]
$HELM_HOME has been configured at /home/ubuntu/.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.  
For more information on securing your installation see: [https://docs.helm.sh/using\\_helm/#securing-your-helm-installation](https://docs.helm.sh/using_helm/#securing-your-helm-installation)  
Happy Helming!



# WorkShop

- Basic Command
  - Watch until tiller pods is running

NAME	READY	STATUS	RESTARTS	AGE
calico-etcd-qzs5b	1/1	Running	0	30m
calico-kube-controllers-5db8f7fd5c-w6ptg	1/1	Running	0	30m
calico-node-kwdp2	2/2	Running	1	24m
calico-node-ksw4	2/2	Running	1	24m
calico-node-s2mrg	2/2	Running	0	30m
etcd-ip-10-0-1-66	1/1	Running	0	33m
heapster-546c9964c6-hqqvh	1/1	Running	0	26m
kube-apiserver-ip-10-0-1-66	1/1	Running	0	33m
kube-controller-manager-ip-10-0-1-66	1/1	Running	0	33m
kube-dns-6f4fd4bdf-p9182	3/3	Running	0	33m
kube-proxy-ccplq	1/1	Running	0	24m
kube-proxy-dbccz	1/1	Running	0	33m
kube-proxy-v2r79	1/1	Running	0	24m
kube-scheduler-ip-10-0-1-66	1/1	Running	0	33m
kubernetes-dashboard-5bd6f767c7-xxlgn	1/1	Running	0	26m
monitoring-influxdb-66946c9f58-s6f5f	1/1	Running	0	26m
tiller-deploy-7ccf99cd64-hjjvc	1/1	Running	0	9m

- Check helm connection via command: helm list

```
[ubuntu@ip-10-0-1-66:~$ helm list
ubuntu@ip-10-0-1-66:~$ ]
```



# WorkShop

- Basic Command
  - Watch until tiller pods is running

NAME	READY	STATUS	RESTARTS	AGE
calico-etcd-qzs5b	1/1	Running	0	30m
calico-kube-controllers-5db8f7fd5c-w6ptg	1/1	Running	0	30m
calico-node-kwdp2	2/2	Running	1	24m
calico-node-ksw4	2/2	Running	1	24m
calico-node-s2mrg	2/2	Running	0	30m
etcd-ip-10-0-1-66	1/1	Running	0	33m
heapster-546c9964c6-hqqvh	1/1	Running	0	26m
kube-apiserver-ip-10-0-1-66	1/1	Running	0	33m
kube-controller-manager-ip-10-0-1-66	1/1	Running	0	33m
kube-dns-6f4fd4bdf-p9182	3/3	Running	0	33m
kube-proxy-ccplq	1/1	Running	0	24m
kube-proxy-dbccz	1/1	Running	0	33m
kube-proxy-v2r79	1/1	Running	0	24m
kube-scheduler-ip-10-0-1-66	1/1	Running	0	33m
kubernetes-dashboard-5bd6f767c7-xxlgn	1/1	Running	0	26m
monitoring-influxdb-66946c9f58-s6f5f	1/1	Running	0	26m
tiller-deploy-7ccf99cd64-hjjvc	1/1	Running	0	9m

- Check helm connection via command: helm list

```
[ubuntu@ip-10-0-1-66:~$ helm list
ubuntu@ip-10-0-1-66:~$ ]
```



# Question & Answer Section



By: Praparn L (eva10409@gmail.com)



**kubernetes**  
by Google