<https://www.ibm.com/cloud/garage/category/courses>

kubernetes cheat sheet

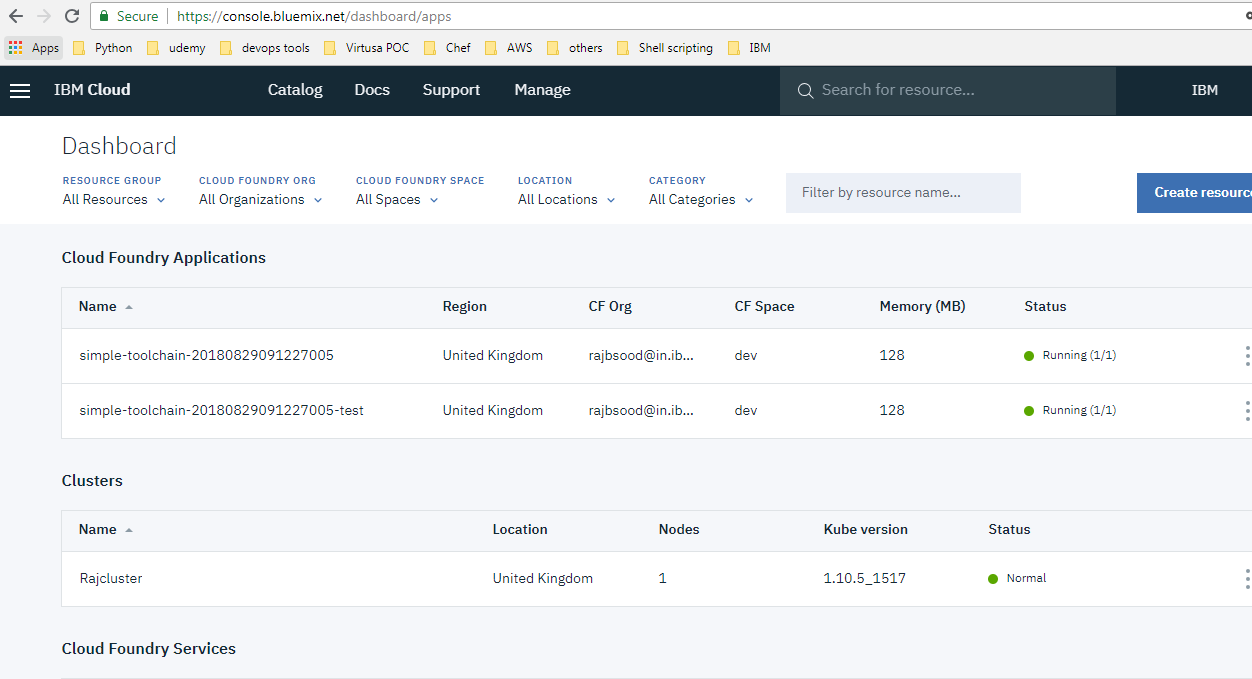
<https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

To create the kubernetes cluster

<https://console.bluemix.net/containers-kubernetes/catalog/cluster?bss_account=90488ffecf50473ba57279d415e24aef>

to see existing cluster -> go to IBM blue-Mix dashboard

<https://console.bluemix.net/dashboard/apps>



# Getting started with IBM Cloud Kubernetes Service

https://console.bluemix.net/docs/containers/container\_index.html#container\_index

**Introduce toolchains by using the "Develop a Cloud Foundry app" toolchain**

[**https://www.ibm.com/cloud/garage/tutorials/introduce-develop-cloud-foundry-app-toolchain**](https://www.ibm.com/cloud/garage/tutorials/introduce-develop-cloud-foundry-app-toolchain)

bx login -a "[https://api.eu-gb.bluemix.net](https://api.eu-gb.bluemix.net/)" --apikey

PS C:\Users\rajbir\_sood> **bx cs clusters // to check cluster**

Using default API endpoint: https://containers.bluemix.net

OK

OK

Name ID State Created Workers Location Version

Rajcluster 0e6355d3f82b49fca64b14e8b5db5c2c normal 2 days ago 1 London 1.10.5\_1517

**bx plugin list**

PS C:\Users\rajbir\_sood> bx plugin list

Listing installed plug-ins...

Plugin Name Version

container-service/kubernetes-service 0.1.575

// to check cluster

PS C:\Users\rajbir\_sood> **bx cs clusters**

Using default API endpoint: https://containers.bluemix.net

OK

If you have clusters that run Kubernetes versions 1.5, 1.7 or 1.8, update them now to continue receiving important secur

ity updates and support. Kubernetes version 1.8 is deprecated and will be unsupported 22 Sept 2018. Versions 1.5 and 1.7

are already unsupported. For more information and update actions, see <https://ibm.biz/iks-versions>

OK

Name ID State Created Workers Location Version

Rajcluster 0e6355d3f82b49fca64b14e8b5db5c2c normal 6 days ago 1 London 1.10.5\_1517

PS C:\Users\rajbir\_sood>

**To check workers in cluster**

$ bx cs workers Rajcluster

Using default API endpoint: https://containers.bluemix.net

OK

OK

ID Public IP Private IP Machine Type State Status Zone Version

kube-mil01-pa0e6355d3f82b49fca64b14e8b5db5c2c-w1 159.122.181.190 10.144.185.165 free normal Ready mil01 1.10.5\_1519

Running on minikube locally

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ minikube start --insecure-registry localhost:5000

Starting local Kubernetes v1.10.0 cluster...

Starting VM...

Getting VM IP address...

Moving files into cluster...

Setting up certs...

Connecting to cluster...

Setting up kubeconfig...

Starting cluster components...

Kubectl is now configured to use the cluster.

Loading cached images from config file.

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ minikube docker-env

export DOCKER\_TLS\_VERIFY="1"

export DOCKER\_HOST="tcp://192.168.99.101:2376"

export DOCKER\_CERT\_PATH="C:\Users\rajbir\_sood\.minikube\certs"

export DOCKER\_API\_VERSION="1.35"

# Run this command to configure your shell:

# eval $(minikube docker-env)

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ eval $(minikube docker-env)

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ minikube dashboard

Opening kubernetes dashboard in default browser...

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ ls

app.js deployment.yml Dockerfile License.txt local.yml package.json README.md

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ ls

app.js deployment.yml Dockerfile License.txt local.yml package.json README.md

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ kubectl apply -f local.yml

deployment.extensions/hello-app created

service/hello-service created

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ kubectl get pods

NAME READY STATUS RESTARTS AGE

hello-app-5d484c6959-f64gv 1/1 Running 0 37s

hello-app-5d484c6959-qsq8s 1/1 Running 0 38s

hello-minikube-c8b6b4fdc-vd7fz 1/1 Running 3 4d

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ kubectl get deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

hello-app 2 2 2 2 49s

hello-minikube 1 1 1 1 4d

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

hello-minikube NodePort 10.103.195.85 <none> 8080:31015/TCP 4d

hello-service NodePort 10.104.165.108 <none> 80:32648/TCP 53s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5d

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ ^C

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ minikube service hello-service --url

http://192.168.99.101:32648

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

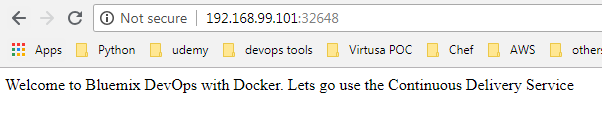
$ curl http://192.168.99.101:32648

% Total % Received % Xferd Average Speed Time Time Time Current

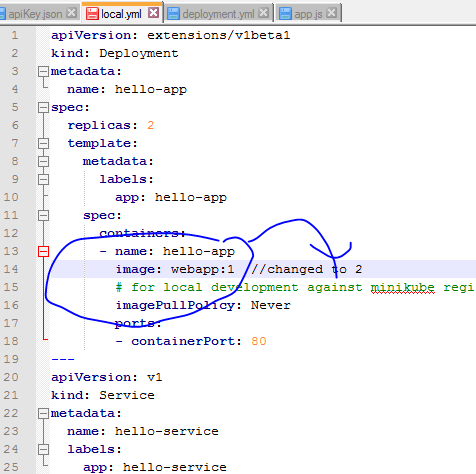
Dload Upload Total Spent Left Speed

100 82 100 82 0 0 2645 0 --:--:-- --:--:-- --:--:-- 5466Welcome to Bluemix DevOps with Docker. Lets go use the Continuous Delivery Service

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)



Now update version



rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ docker build . -t webapp:2

Sending build context to Docker daemon 93.18kB

Step 1/8 : FROM node:6.12.0-alpine

---> fa74995b3b27

Step 2/8 : MAINTAINER Philippe Mulet "philippe\_mulet@fr.ibm.com"

---> Using cache

---> bd991032b22d

Step 3/8 : ADD package.json /app/package.json

---> Using cache

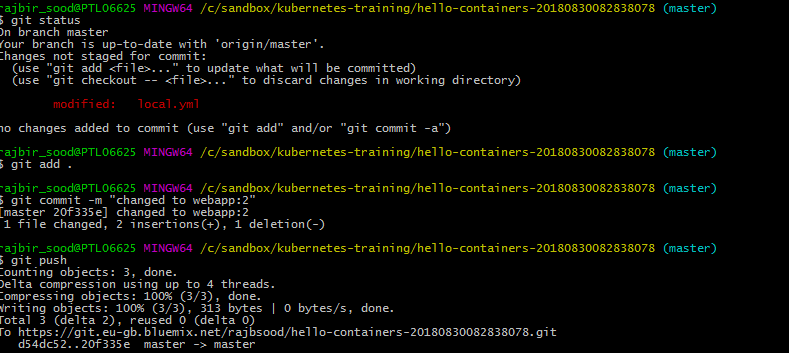
---> 949c0ab9cc56

Step 4/8 : RUN cd /app && npm install

---> Using cache

---> bf8c135d7736

Step 5/8 : ADD app.js /app/app.js



rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: local.yml

no changes added to commit (use "git add" and/or "git commit -a")

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ git add .

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ git commit -m "changed to webapp:2"

[master 20f335e] changed to webapp:2

1 file changed, 2 insertions(+), 1 deletion(-)

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ git push

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 313 bytes | 0 bytes/s, done.

Total 3 (delta 2), reused 0 (delta 0)

To https://git.eu-gb.bluemix.net/rajbsood/hello-containers-20180830082838078.git

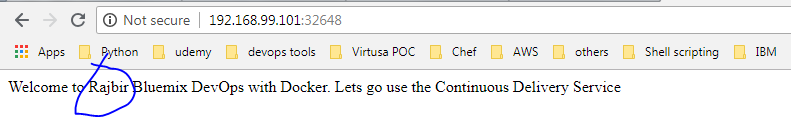
d54dc52..20f335e master -> master

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

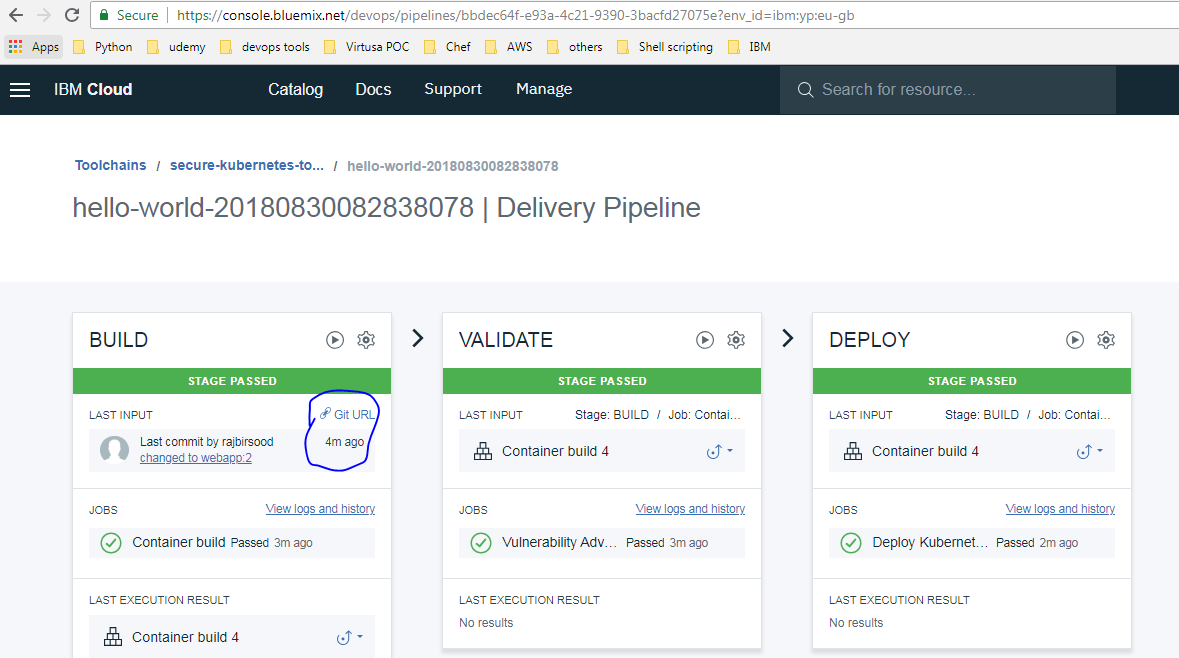
$ kubectl apply -f local.yml

deployment.extensions/hello-app configured

service/hello-service unchanged



On the toolchain's Overview page, click **Delivery Pipeline** to see your toolchain as it is built and deployed. Your pipeline might still be running.



# Installing Istio on IBM Cloud Kubernetes Service

<https://console.bluemix.net/docs/containers/cs_tutorials_istio.html#istio_tutorial>

rajbir\_sood@PTL06625 MINGW64 ~

$ cd Downloads/

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ ibmcloud login --apikey ggZAdoNYcqUJfJljBL\_3NkRxSyYalUjS4CCIFE5EKoo\_

API endpoint: https://api.eu-gb.bluemix.net

Authenticating...

OK

List all of the clusters in the account to get the name of the cluster.

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ ibmcloud ks clusters

Using default API endpoint: https://containers.bluemix.net

OK

If you have clusters that run Kubernetes versions 1.5, 1.7 or 1.8, update them now to continue receiving important security updates and support. Kubernetes version 1.8 is deprecated and will be unsupported 22 Sept 2018. Versions 1.5 and 1.7 are already unsupported. For more information and update actions, see <https://ibm.biz/iks-versions>

OK

Name ID State Created Workers Location Version

Rajcluster 0e6355d3f82b49fca64b14e8b5db5c2c normal 6 days ago 1 London 1.10.7\_1520

Set the cluster you created as the context for this session. Complete these configuration steps every time that you work with your cluster.

Get the command to set the environment variable and download the Kubernetes configuration files.

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ ibmcloud ks cluster-config Rajcluster

Using default API endpoint: https://containers.bluemix.net

OK

OK

The configuration for Rajcluster was downloaded successfully. Export environment variables to start using Kubernetes.

export KUBECONFIG=/C/Users/rajbir\_sood/.bluemix/plugins/container-service/clusters/Rajcluster/kube-config-mil01-Rajcluster.yml

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ echo $KUBECONFIG

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ export KUBECONFIG=/C/Users/rajbir\_sood/.bluemix/plugins/container-service/clusters/Rajcluster/kube-config-mil01-Rajcluster.yml

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ echo $KUBECONFIG

/C/Users/rajbir\_sood/.bluemix/plugins/container-service/clusters/Rajcluster/kube-config-mil01-Rajcluster.yml

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ export KUBECONFIG=/C/Users/rajbir\_sood/.bluemix/plugins/container-service/clusters/Rajcluster/kube-config-mil01-Rajcluster.yml

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ kubectl get nodes

NAME STATUS ROLES AGE VERSION

10.144.185.165 Ready <none> 7d v1.10.5+IKS

Update the IBM Cloud Kubernetes Service plug-in.

1. Install the update from the IBM Cloud plug-in repository.

ibmcloud plugin update container-service

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ ibmcloud plugin list

Listing installed plug-ins...

Plugin Name Version

container-registry 0.1.329

container-service/kubernetes-service 0.1.575

Initialize the CLI.

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ ibmcloud ks init

Using default API endpoint: https://containers.bluemix.net

OK

Update the IBM Cloud Container Registry plug-in.

ibmcloud plugin update container-registry

rajbir\_sood@PTL06625 MINGW64 ~/Downloads

$ ibmcloud plugin list

Listing installed plug-ins...

Plugin Name Version

container-registry 0.1.329

container-service/kubernetes-service 0.1.575

## **Download and install Istio**

Before you use Helm charts with IBM Cloud Kubernetes Service, you must install and initialize a Helm instance in your cluster. You can then add the IBM Cloud Helm repository to your Helm instance.

Repeat above steps to install ibm cli and cluster and kubeconfig

Now, Install the Helm Cli

<https://docs.helm.sh/using_helm/#installing-helm>

rajbir\_sood@PTL06625 MINGW64 /c/sandbox

$ mkdir helm\_script

rajbir\_sood@PTL06625 MINGW64 /c/sandbox

$ cd helm\_script/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm\_script

$ curl https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get > get\_helm.sh

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 6709 100 6709 0 0 7967 0 --:--:-- --:--:-- --:--:-- 8969

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm\_script

$ ls

get\_helm.sh\*

rajbir\_sood@PTL06625 MINGW64 /c/minikube

$ tar -xvzf helm-v2.10.0-windows-amd64.tar.gz

windows-amd64/

windows-amd64/LICENSE

windows-amd64/helm.exe

windows-amd64/README.md

rajbir\_sood@PTL06625 MINGW64 /c/minikube

$ ls

helm-v2.10.0-windows-amd64.tar.gz kubectl.exe\* minikube.exe\* windows-amd64/

rajbir\_sood@PTL06625 MINGW64 /c/minikube

$ cd windows-amd64/

rajbir\_sood@PTL06625 MINGW64 /c/minikube/windows-amd64

$ ls

helm.exe\* LICENSE README.md

## INSTALLING TILLER

Tiller, the server portion of Helm, typically runs inside of your Kubernetes cluster. But for development, it can also be run locally, and configured to talk to a remote Kubernetes cluster.

PS C:\sandbox\helm> **kubectl create -f .\rbac-config.yaml**

serviceaccount/tiller created

clusterrolebinding.rbac.authorization.k8s.io/tiller created

helm init --service-account tiller

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ **helm init --service-account tiller**

Creating C:\Users\rajbir\_sood\.helm

Creating C:\Users\rajbir\_sood\.helm\repository

Creating C:\Users\rajbir\_sood\.helm\repository\cache

Creating C:\Users\rajbir\_sood\.helm\repository\local

Creating C:\Users\rajbir\_sood\.helm\plugins

Creating C:\Users\rajbir\_sood\.helm\starters

Creating C:\Users\rajbir\_sood\.helm\cache\archive

Creating C:\Users\rajbir\_sood\.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 C:\Users\rajbir\_sood\.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

Happy Helming!

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ **kubectl get pods -n kube-system -l app=helm**

NAME READY STATUS RESTARTS AGE

tiller-deploy-895d57dd9-zq86l 1/1 Running 0 46s

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ **helm repo add ibm https://registry.bluemix.net/helm/ibm**

"ibm" has been added to your repositories

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ **helm search ibm**

NAME CHART VERSION APP VERSION DESCRIPTION

ibm/ibm-istio 1.0.3 1.0.0 Helm chart for all istio components

ibm/ibm-istio-remote 1.0.3 1.0.0 Helm chart needed for remote Kubernetes clusters to join ...

ibm/ibm-worker-recovery 1.10.13 IBM Autorecovery system allows automatic recovery of unhe...

ibm/ibmcloud-block-storage-plugin 1.0.2 A Helm chart for installing ibmcloud block storage plugin

ibm/ibmcloud-object-storage-plugin 1.0.0 A Helm chart for installing ibmcloud object storage plugin

ibm/strongswan 2.2.2 18.06.18 A strongSwan IPSec VPN service to securely connect your K...

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ **helm install ibm/ibm-istio --name=istio --namespace istio-system**

NAME: istio

LAST DEPLOYED: Thu Sep 6 15:28:00 2018

NAMESPACE: istio-system

STATUS: DEPLOYED

NOTES:

Thank you for installing ibm-istio.

Your release is named istio.

To get started running application with Istio, execute the following steps:

1. Label namespace that application object will be deployed to by the following command (take default namespace as an example)

$ kubectl label namespace default istio-injection=enabled

$ kubectl get namespace -L istio-injection

2. Deploy your applications

$ kubectl apply -f <your-application>.yaml

For more information on running Istio, visit:

https://istio.io/

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ kubectl get pods

NAME READY STATUS RESTARTS AGE

hello-app-77d8d9cf9-78j2w 1/1 Running 0 1d

hello-app-77d8d9cf9-g88fq 1/1 Running 0 1d

rajbir\_sood@PTL06625 MINGW64 ~/.bluemix/plugins/container-service/clusters/Rajcluster

$ kubectl get pods -n istio-system

NAME READY STATUS RESTARTS AGE

istio-citadel-79fc7bb775-8tp25 1/1 Running 0 30m

istio-egressgateway-65c657f6f4-x4hsb 1/1 Running 0 30m

istio-galley-6567b78c96-gjzcl 1/1 Running 0 30m

istio-ingressgateway-86f5b48447-c95zz 1/1 Running 0 30m

istio-pilot-768dc59b79-zmp5q 2/2 Running 0 30m

istio-policy-76c9cc7f89-n5ftj 2/2 Running 0 30m

istio-sidecar-injector-5f4b7665b7-wc7r9 1/1 Running 0 30m

istio-statsd-prom-bridge-5b99cb8dcd-h8w4n 1/1 Running 0 30m

istio-telemetry-74f5895546-h45br 2/2 Running 0 30m

prometheus-8657544dcf-c4cd9 1/1 Running 0 30m

Good work! You successfully installed Istio into your cluster. Next, deploy the BookInfo sample app into your cluster.

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training

$ bx cr images

Listing images...

REPOSITORY TAG DIGEST NAMESPACE CREATED SIZE SECURITY STATUS

registry.eu-gb.bluemix.net/mykube/hello-containers-20180830082838078 1 ad35581a11fb mykube 1 week ago 32 MB No Issues

registry.eu-gb.bluemix.net/mykube/hello-containers-20180830082838078 2 799c9d520e10 mykube 2 days ago 32 MB No Issues

registry.eu-gb.bluemix.net/mykube/hello-containers-20180830082838078 3 b1d4f60f0ac9 mykube 2 days ago 32 MB No Issues

registry.eu-gb.bluemix.net/mykube/hello-containers-20180830082838078 4 b1d4f60f0ac9 mykube 2 days ago 32 MB No Issues

registry.eu-gb.bluemix.net/mykube/hello-helm-rajbirsood 1 55a430bc0cf1 mykube 2 hours ago 32 MB No Issues

OK

============helm chart==========

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-containers-20180830082838078 (master)

$ **helm init --upgrade**

$HELM\_HOME has been configured at C:\Users\rajbir\_sood\.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

Happy Helming!

Clone repo

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training

$ ls

hello-containers-20180830082838078/ hello-helm-rajbirsood/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training

$ cd hello-helm-rajbirsood/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ ls

app.js chart/ Dockerfile License.txt package.json README.md

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ cd chart/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood/chart (master)

$ ls

hello/

Build image

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ cp ./chart/hello/values.yaml to local.yaml

cp: target 'local.yaml' is not a directory

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ cp ./chart/hello/values.yaml local.yaml

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ ls

app.js chart/ Dockerfile License.txt local.yaml package.json README.md

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

minikube start --insecure-registry localhost:5000

$ minikube start --insecure-registry localhost:5000

$ minikube docker-env

Execute the last command:

$ eval $(minikube docker-env)

And bring up the Minikube dashboard

$ minikube dashboard

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ ls

app.js chart/ Dockerfile License.txt local.yaml package.json README.md

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ docker build . -t webapp:2

Sending build context to Docker daemon 135.7kB

Step 1/8 : FROM node:6.12.0-alpine

---> fa74995b3b27

Step 2/8 : MAINTAINER Philippe Mulet "philippe\_mulet@fr.ibm.com"

---> Using cache

---> bd991032b22d

Deploy the Helm chart.

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ helm upgrade --install release chart/hello --set image.repository=webapp,image.tag=2

Release "release" does not exist. Installing it now.

NAME: release

LAST DEPLOYED: Thu Sep 6 19:34:20 2018

NAMESPACE: default

STATUS: DEPLOYED

RESOURCES:

==> v1/Service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

release-hello NodePort 10.111.173.51 <none> 80:31146/TCP 4s

==> v1beta1/Deployment

NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE

release-hello 2 0 0 0 3s

==> v1/Pod(related)

NAME READY STATUS RESTARTS AGE

release-hello-67fc949bdf-9hzqq 0/1 Pending 0 1s

release-hello-67fc949bdf-pl88t 0/1 Pending 0 1s

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ kubectl get pod

NAME READY STATUS RESTARTS AGE

hello-app-8f9bb4ccb-46g76 1/1 Running 5 2d

hello-app-8f9bb4ccb-vjmjz 1/1 Running 5 2d

hello-minikube-c8b6b4fdc-vd7fz 1/1 Running 8 6d

release-hello-67fc949bdf-9hzqq 1/1 Running 0 1m

release-hello-67fc949bdf-pl88t 1/1 Running 0 1m

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

hello-minikube NodePort 10.103.195.85 <none> 8080:31015/TCP 6d

hello-service NodePort 10.104.165.108 <none> 80:32648/TCP 2d

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 8d

release-hello NodePort 10.111.173.51 <none> 80:31146/TCP 2m

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ minikube service release-hello

Opening kubernetes service default/release-hello in default browser...

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ minikube service release-hello --url

http://192.168.99.101:31146

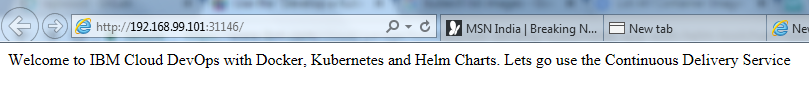
rajbir\_sood@PTL06625 MINGW64 /c/sandbox/kubernetes-training/hello-helm-rajbirsood (master)

$ curl http://192.168.99.101:31146

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 112 100 112 0 0 7000 0 --:--:-- --:--:-- --:--:-- 7000Welcome to IBM Cloud DevOps with Docker, Kubernetes and Helm Charts. Lets go use the Continuous Delivery Service



Make changes to app.js and rebuild with webapp:3

helm upgrade --install release chart/hello --set image.repository=webapp,image.tag=3

==============

Helm chart

<http://tech.paulcz.net/blog/getting-started-with-helm/>

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm

$ cd my-first-helm-chart/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ ls

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ mkdir manifests

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ kubectl run example --image=nginx:1.13.5-alpine \

> -o yaml > manifests/deployment.yaml

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ ls

manifests/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ cd manifests/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart/manifests

$ ls

deployment.yaml

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart/manifests

$ cd ..

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ ls

manifests/

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ kubectl expose deployment example --port=80 --type=NodePort \

> -o yaml > manifests/service.yaml

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ minikube service example --url

Waiting, endpoint for service is not ready yet...

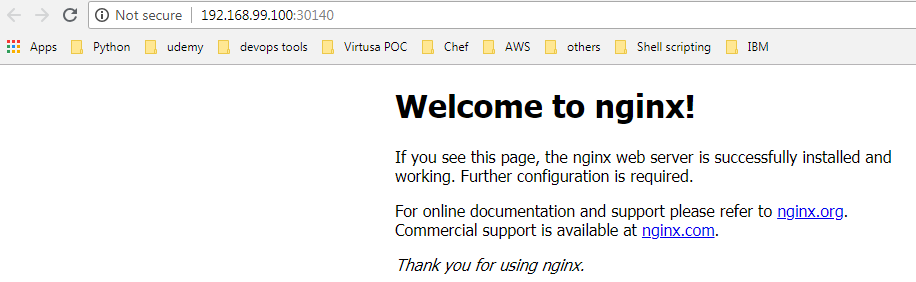
rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$

rajbir\_sood@PTL06625 MINGW64 /c/sandbox/helm/my-first-helm-chart

$ minikube service example --url

http://192.168.99.100:30140



## **Create and Deploy a Basic Helm Chart**