IBM Cloud Private 3.1.2

Lab Exercise #2

Prepare Helm chart, deploy , upgrade and rollback.

Duration: 60 mins.

Table of Contents

Objective

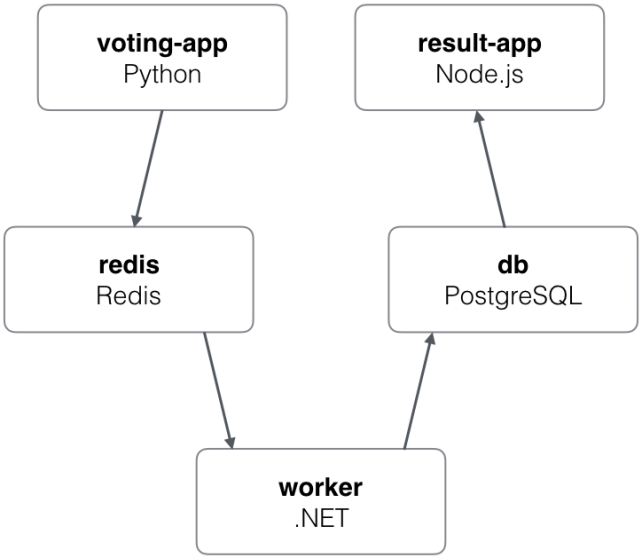
1. To create a helm chart for an application having dependencies which are packaged together
2. Learn how to create the helm chart.
3. Deploy the chart from command line.
4. Access the deployed application.
5. Modify the application and upgrade the existing release with new version of the chart
6. Rollback the release to the older version.

Assumptions

1. User has access to ICP cluster and helm cli has been downloaded and configured to connect with the given ICP cluster.
2. User has access to his own namespace.
3. User namespace can be associated with pod security policy - 'ibm-anyuid-psp'
4. Catalog CLI and Helm CLI have been configured to work with the ICP instance.
5. User has access to storageClass (e.g. glusterfs or ceph) and its set to default sc.

Instructions

1. Walk through of the voting app



Vote app : UI to vote for cats/dogs.
Redis db : Stores the vote from vote app.
Worker app: Pulls from redis and updates Postgress db
Postgress db: Stores the results of voting for the result app.
Result app: UI to show results of voting

2. Deploy the sample app as is

Lab can be found at this folder on your machine (or on github **https://github.com/sachinkj1982/All-Labs/tree/master/Lab-03/example-voting-app**)

$ C:\Users\Administrator\labs\All-Labs\Lab-03\example-voting-app

Go to the above folder and then follow the steps given below

Replace **<your-namespace>** with the namespace allocated for you for the duration of the lab exercises.

2.1 Login

$ `cloudctl login -a https://174.37.17.188:8443`

2.2 Set the target namespace to <your-namespace>

$ kubectl config set-context mycluster-context --namespace=<your-namespace>

2.3 Create docker secret

Replace <user-id> with your user id.

$ kubectl create secret docker-registry registry-secret --docker-server=optumera.icp:8500 --docker-username=<user-id> --docker-password=passw0rd --docker-email=null

2.4 Patch default service account to use the imagePullSecret

```
$ kubectl patch serviceaccount default -p
'{"imagePullSecrets": [{"name": "registry-secret" }]}'
```

This is required since the images have been pushed to the default namespace and by default the scope of images is 'namespace'. So for deployment in other namespace, the default service account in the namespace need to use imagePullSecret.

2.5 Deploy using the individual k8s deployment and service definition files

```
$ kubectl create -f k8s-specifications/
```

```
niladris-MacBook-Pro:example-voting-app niladri$ kubectl create -f k8s-specifications/
deployment.extensions/db created
service/db created
deployment.extensions/redis created
service/redis created
deployment.extensions/result created
service/result created
deployment.extensions/vote created
service/vote created
deployment.extensions/worker created
```

2.6 Access the vote app at given NodePorts

2.6.1 Option 1 – From console

a. Login to web console https://174.37.17.188:8443

b. **Launch from workloads > Deployment tab**

2.6.2 Option 2 – Identify the Host and port through kubectl

Issue the following command to get the service port

```
kubectl get svc -n <your-namespace>
```

```
niladris-MacBook-Pro:example-voting-app niladri$ kubectl get svc -o wide -n user2
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE   SELECTOR
db        ClusterIP   20.0.13.152     <none>        5432/TCP         22m   app=db
redis     ClusterIP   20.0.197.188    <none>        6379/TCP         22m   app=redis
result    NodePort    20.0.207.162    <none>        5001:30412/TCP   22m   app=result
vote      NodePort    20.0.130.232    <none>        5000:32106/TCP   22m   app=vote
```

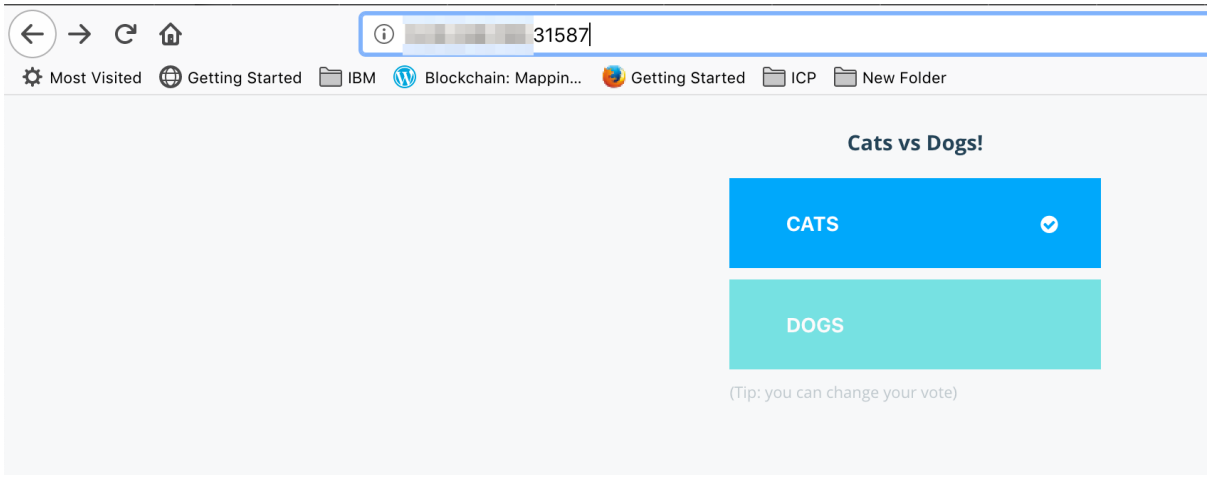
Issue the following command to get the Node IP of the POD

```
kubectl get pod -n <your-namespace>
```

```
niladris-MacBook-Pro:example-voting-app niladri$ kubectl get pods -o wide -n user2
NAME                      READY   STATUS    RESTARTS   AGE   IP            NODE              NOMINATED NODE
db-788bc87dd7-jjgwc       1/1     Running   0          21m   20.1.142.18   10.186.255.207    <none>
redis-bb9d87fcd-6mxsm     1/1     Running   0          21m   20.1.51.81    10.186.255.229    <none>
result-68d589f669-drv4l   1/1     Running   0          21m   20.1.46.211   10.186.255.247    <none>
vote-79c6f88cbd-dd9bd     1/1     Running   0          21m   20.1.142.19   10.186.255.207    <none>
worker-5cbccb5d9d-b7ml8   1/1     Running   0          21m   20.1.51.82    10.186.255.229    <none>
```

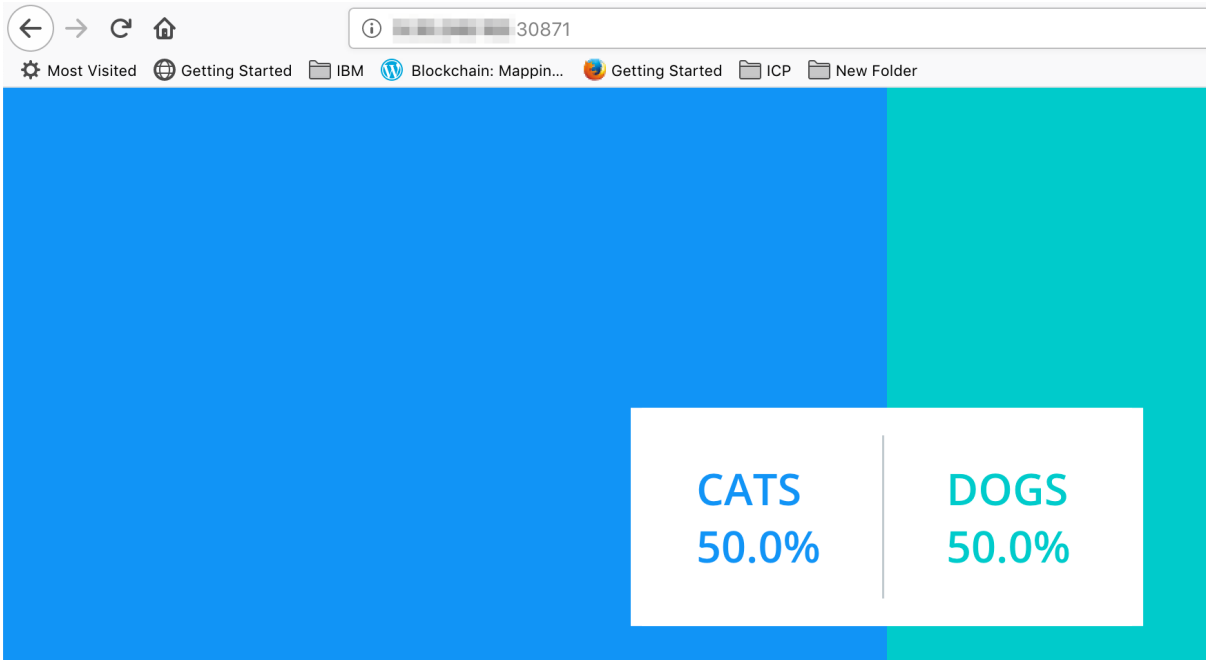
This is the internal IP of the worker node. Ask your Lab instructor for the external IP corresponding to this internal IP.

Vote app: http://<worker_ip>:<voteServiceNodePort>



Repeat the same steps to get the worker node IP and service port for the result service.

Result app: http://<worker_ip>:<resultServiceNodePort>



3. Create helm Chart

3.1 Create default helm chart

cd to <project root directory>

$ helm create `voting-app-helm-charts`

This creates the folder structure for a helm chart.

Follow the steps below. In case of doubts refer the modified chart in the final-charts folder

3.2 Create required charts for micro services.

Go to <projectroot>/voting-app-helm-charts/charts folder

$ helm create db

$ helm create redis

$ helm create result

$ helm create worker

$ helm create vote

3.3 Modify the db chart templates

3.3.1 Just replace the current deployment.yaml and service.yaml files under db/templates/ with the contents from k8s-specification/db-deployment.yaml and k8s-specification/db-service.yaml

3.3.2 Delete ingress.yaml, and NOTES.txt

$ *del db/templates/ingress.yaml*
$ del db/templates/NOTES.txt
$ del db/templates/deployment.yaml
$ del db/templates/service.yaml
$ copy ..\..\k8s-specifications\db-deployment.yaml db\templates
$ copy ..\..\k8s-specifications\db-service.yaml db\templates

3.4 Modify the redis chart templates

Just replace the current deployment.yaml and service.yaml files under redis/templates/ with the contents from k8s-specification/redis-deployment.yaml and k8s-specification/redis-service.yaml

Delete ingress.yaml, NOTES.txt

$ *del redis/templates/ingress.yaml*
$ del redis/templates/NOTES.txt
$ del redis/templates/deployment.yaml
$ del redis/templates/service.yaml
$ copy ..\..\k8s-specifications\redis-deployment.yaml redis\templates
$ copy ..\..\k8s-specifications\redis-service.yaml redis\templates

3.5 Modify the vote chart

Delete ingress.yaml, NOTES.txt

$ *del vote/templates/ingress.yaml*

$ del vote/templates/NOTES.txt

$ del vote/templates/deployment.yaml

$ del vote/templates/service.yaml

Copy deployment.yaml and service.yaml from gitbub
https://github.com/sachinkj1982/All-Labs/tree/master/Lab-03/example-voting-app/final-charts/voting-app-helm-charts/charts/vote/templates

Modify the <projectroot>/voting-app-helm-charts/charts/vote/values.yaml as follows:

```
1. replicaCount: 1
2.
3. image:
4.   repository: optumera.icp:8500/default/vote
5.   tag: 0.1.0
6.   pullPolicy: IfNotPresent
7.
8. service:
9.   type: NodePort
10.      port: 80
```

3.6 Modify the result chart

Just replace the current deployment.yaml and service.yaml files under result/templates/ with the contents from k8s-specification/ result-deployment.yaml and k8s-specification/result-service.yaml

Delete ingress.yaml, NOTES.txt

$ del result/templates/ingress.yaml

$ del result/templates/NOTES.txt

$ del result/templates/deployment.yaml

$ del result/templates/service.yaml

3.7 Modify the worker chart

Delete files templates/service.yaml and templates/ingress.yaml.

$ *del worker/templates/ingress.yaml*

$ del worker/templates/NOTES.txt

$ del worker/templates/deployment.yaml

$ del worker/templates/service.yaml

$ copy ..\..\k8s-specifications\worker-deployment.yaml worker\templates

3.8 Update the top level values.yaml

Add parameters so that any of included chart parameters can be configured during install.

Update the <projectroot>/voting-app-helm-charts/values.yaml as follows:

```
# Default values for voting-app-helm-charts.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

global:
  serviceAccountName: default

worker:
  replicaCount: 1

  image:
```

```
    repository: optumera.icp:8500/default/worker
    tag: 0.1.0
    pullPolicy: IfNotPresent

vote:
  replicaCount: 1

  image:
    repository: optumera.icp:8500/default/vote
    tag: 0.1.0
    pullPolicy: IfNotPresent

  service:
    type: NodePort
    port: 80

result:
  replicaCount: 1

  image:
    repository: optumera.icp:8500/default/result
    tag: 0.1.0
    pullPolicy: IfNotPresent

  service:
    type: NodePort
    port: 80

db:
  dataPVC:
    name: db-pvc
    storageClassName:
    useDynamicProvisioning: true
    accessMode: ReadWriteOnce
    size: 5Gi
```

The parameters are same as in the included chart's values.yaml but they are nested under the chart name.

3.9 Update top level templates

Go to <projectroot>/voting-app-helm-charts/

$ del templates/ingress.yaml
$ del templates/NOTES.txt
$ del templates/deployment.yaml
$ del templates/service.yaml

3.10 Validate the charts

Go to <projectroot>/voting-app-helm-charts/

$ helm lint charts/db
$ helm lint charts/result
$ helm lint charts/vote
$ helm lint charts/worker
$ helm lint charts/redis
$ cd .. (move to the project root folder)
$ helm lint voting-app-helm-charts

There should be no errors during validation.

3.11 Package the helm chart for distribution (optional)

In case you want to add the chart to a repository or share it with someone, there is a step to package the chart which creates a .tgz file

$ helm package voting-app-helm-charts

3.12 Install the chart

Delete the existing deployment and services created from step 2

$ kubectl delete -f ./k8s-specifications -n <your-namespace>

$ helm install ./voting-app-helm-charts --name voting-app-<user-id> --namespace <your-namespace> --tls

```
Sachins-MacBook-Pro:example-voting-app sachinkumarjha$ helm install ./voting-app-helm-charts --tls
NAME:   quieting-gorilla
LAST DEPLOYED: Tue May 21 17:07:13 2019
NAMESPACE: vote
STATUS: DEPLOYED

RESOURCES:
==> v1beta2/Deployment
NAME                     DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
quieting-gorilla-result  1        1        1           0          5s
quieting-gorilla-vote    1        1        1           0          4s
quieting-gorilla-worker  1        1        1           0          3s

==> v1/Pod(related)
NAME                                     READY  STATUS             RESTARTS  AGE
db-66967bd56d-zksw4                      0/1    ContainerCreating  0         5s
redis-5684f8d55c-bhwz4                   0/1    ContainerCreating  0         5s
quieting-gorilla-result-56676544cf-2958j 0/1    ContainerCreating  0         4s
quieting-gorilla-vote-6b569dff88-kssf2   0/1    ContainerCreating  0         4s
quieting-gorilla-worker-7575854cd6-tgf8z 0/1    ContainerCreating  0         3s

==> v1/Service
NAME                   TYPE       CLUSTER-IP     EXTERNAL-IP  PORT(S)       AGE
db                     ClusterIP  10.0.188.114   <none>       5432/TCP      5s
redis                  ClusterIP  10.0.93.133    <none>       6379/TCP      5s
result                 NodePort   10.0.192.254   <none>       80:31555/TCP  5s
quieting-gorilla-vote  NodePort   10.0.195.226   <none>       80:31587/TCP  5s

==> v1beta1/Deployment
NAME   DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
db     1        1        1           0          5s
redis  1        1        1           0          5s
```

Launch the vote and result application from Workloads > Deployment on the ICP web console.

4. Update charts to new version of image and upgrade to a new version.

4.1 Updated image:

For the purpose of this demo, an updated image of vote app has been added to icp image repository. We will just modify the image version in values.yaml

4.2 Update the vote image version in different files , in helm chart

4.2.1 Update the image tag in <project root>/voting-app-helm-charts/charts/vote/values.yaml

```
image:
  repository: mycluster.icp:8500/default/vote
  tag: 0.1.1
  pullPolicy: IfNotPresent
```

4.2.2 Update the image tag in <project root>/voting-app-helm-charts/values.yaml

```
vote:
  replicaCount: 1

  image:
    repository: mycluster.icp:8500/default/vote
    tag: 0.1.1
    pullPolicy: IfNotPresent
```

4.2.3 Update the chart version to 0.1.1 in the <project root>/voting-app-helm-charts/Chart.yaml file

```
apiVersion: v1
appVersion: "1.0"
description: A Helm chart for Kubernetes
name: voting-app-helm-charts
version: 0.1.1
```

4.3 Upgrade existing helm release with new version of the chart.

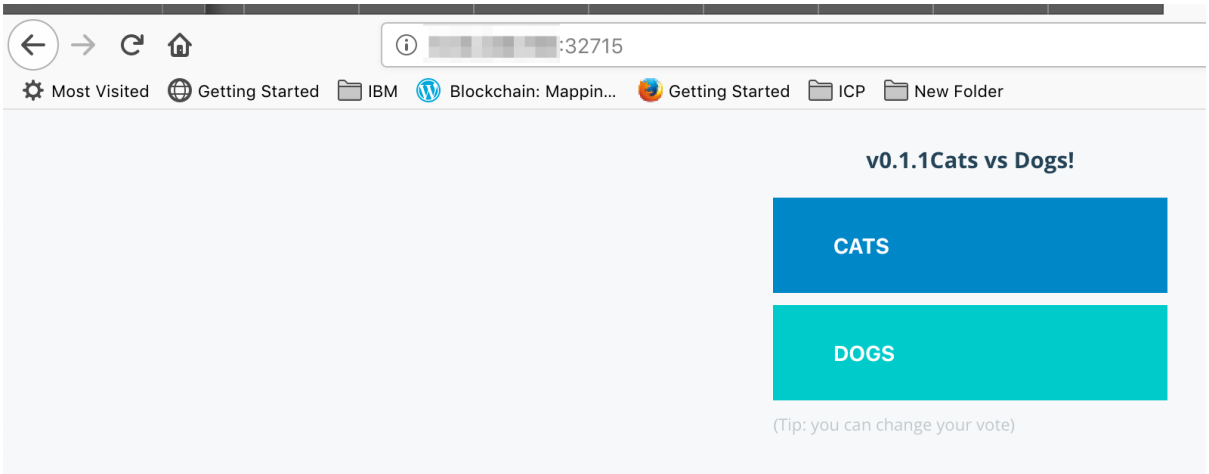
```
$ helm upgrade voting-app-<user-id> ./voting-app-helm-charts
--tls --namespace <your namespace>
```

```
Sachins-MacBook-Pro:example-voting-app sachinkumarjha$ helm upgrade voting-app ./voting-app-helm-charts --tls
Release "voting-app" has been upgraded. Happy Helming!
E0522 11:25:49.609588    1526 portforward.go:303] error copying from remote stream to local connection: readfrom t
27.0.0.1:50622: write: broken pipe
LAST DEPLOYED: Wed May 22 11:25:48 2019
NAMESPACE: vote
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                                READY  STATUS             RESTARTS  AGE
db-66967bd56d-7l6d6                 1/1    Running            0         11h
redis-5684f8d55c-gg6cw              1/1    Running            0         11h
voting-app-result-7bfbcf5b77-r5nwr  1/1    Running            0         19m
voting-app-vote-5b7844b57c-k92tn    0/1    ContainerCreating  0         1s
voting-app-vote-5c48d76f88-p6qg9    1/1    Running            0         19m
voting-app-worker-8467fddd7c-x2t6z  1/1    Running            0         19m

==> v1/PersistentVolumeClaim
NAME          STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS       AGE
postgres-pvc  Bound   pvc-ae26549b-7bf1-11e9-b37d-00163e01d870  5Gi       RWO           rbd-storage-class  11h

==> v1/Service
NAME             TYPE       CLUSTER-IP    EXTERNAL-IP  PORT(S)       AGE
db               ClusterIP  10.0.176.55   <none>       5432/TCP      11h
redis            ClusterIP  10.0.45.101   <none>       6379/TCP      11h
result           NodePort   10.0.126.86   <none>       80:31831/TCP  11h
voting-app-vote  NodePort   10.0.190.181  <none>       80:32715/TCP  11h

==> v1beta1/Deployment
NAME   DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
db     1        1        1           1          11h
redis  1        1        1           1          11h

==> v1beta2/Deployment
voting-app-result  1  1  1  1  11h
voting-app-vote    1  2  1  1  11h
voting-app-worker  1  1  1  1  11h
```

Notice that the new voting-app pod is getting created and in some time the existing one will be deleted.

4.4 Access the app now (URL is still the same)

Observe that the voting app now shows version v0.1.1



5. Rollback to older version

5.1 Check the history of the versions available.

$ helm history voting-app-<user-id> --tls

5.2 Rollback to the desired version.

$ helm rollback voting-app-<user-id> 1 --tls

5.3 Observe the update in application

$ helm list --tls

UI would show the voting app page without version.

6. Clean up

$ helm delete --purge voting-app-<user-id> --tls

Summary

We have gone through the following steps:

1) Looked at the existing voting app as is.
2) Deployed the existing app using individual deployment files.
3) Created the helm chart with dependencies
4) Validate the helm chart via Lint
5) Installed the initial version of the chart
6) Upgraded the helm release to a new version and rolled back to older version.