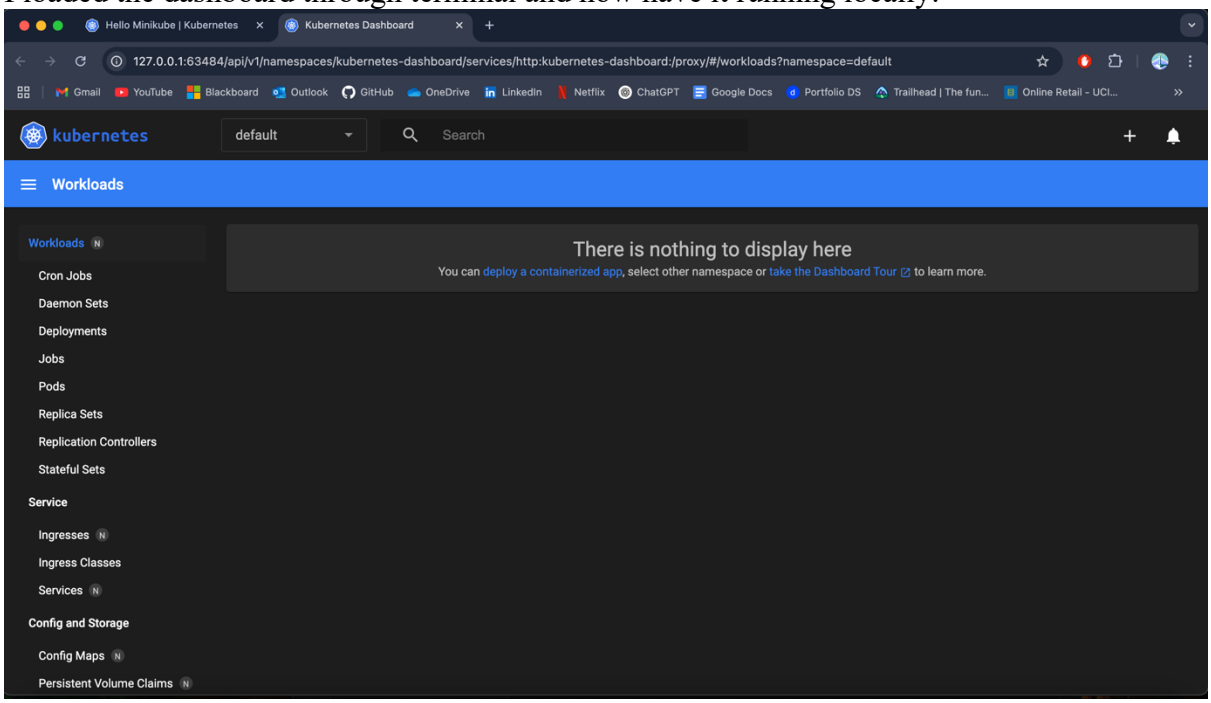
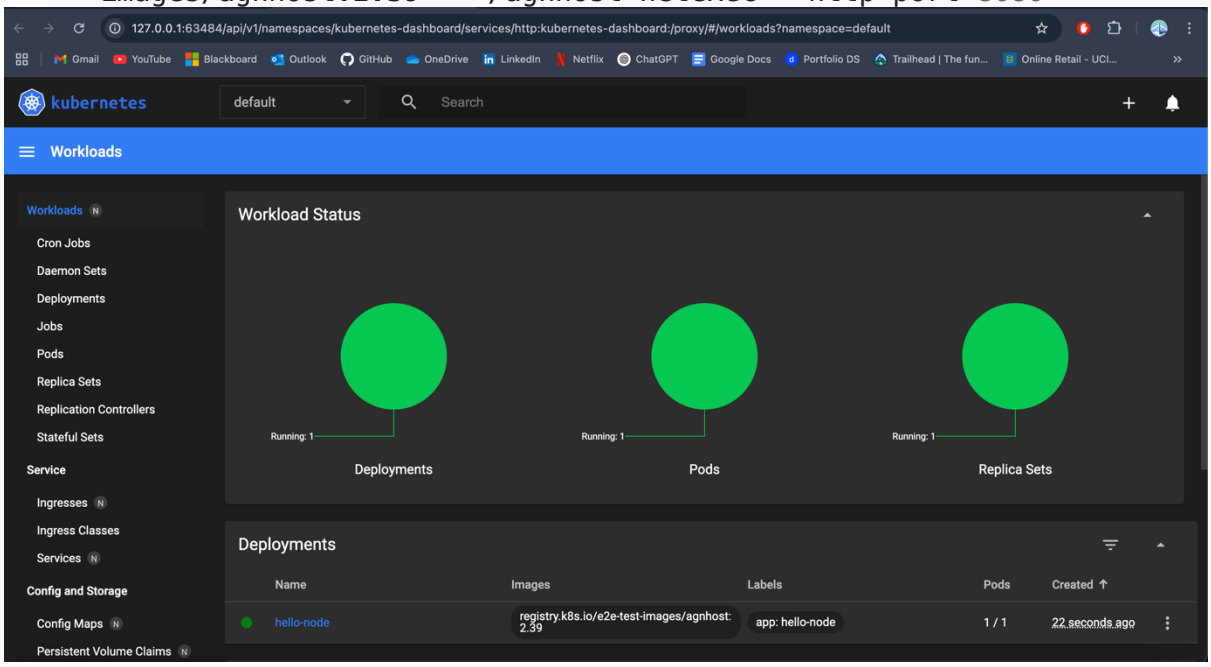
Cloud Data Centres – Lab 2

Hello MiniKube:

I loaded the dashboard through terminal and now have it running locally:



1. ***I then created a deployment using this command:*** `kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080`



Here you can see the deployment and the pod through my terminal

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get deployments

NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           112s
aaron@MacBookAir Cloud-Data-Centres % kubectl get pods

NAME                          READY   STATUS    RESTARTS   AGE
hello-node-c74958b5d-pdnc6    1/1     Running   0          2m4s
aaron@MacBookAir Cloud-Data-Centres %
```

Then I exposed it to the internet

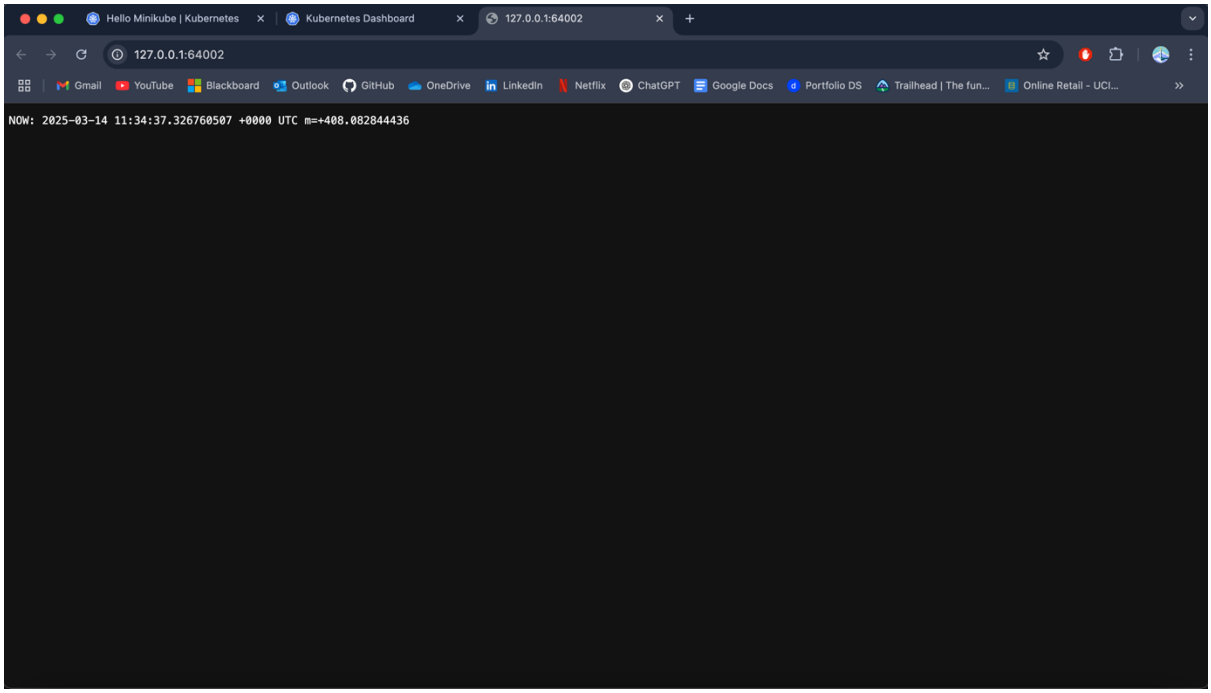
```
aaron@MacBookAir Cloud-Data-Centres % kubectl expose deployment hello-node --type=LoadBalancer --port=8080

service/hello-node exposed
aaron@MacBookAir Cloud-Data-Centres %
```

Here are the services for my cluster

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get services

NAME         TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.107.47.31   <pending>     8080:30181/TCP   101s
kubernetes   ClusterIP      10.96.0.1      <none>        443/TCP          9m21s
aaron@MacBookAir Cloud-Data-Centres %
```

I then ran the following command to get it working in the browser:

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get services

NAME         TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.107.47.31   <pending>     8080:30181/TCP   101s
kubernetes   ClusterIP      10.96.0.1      <none>        443/TCP          9m21s
aaron@MacBookAir Cloud-Data-Centres % minikube service hello-node

|-----------|------------|-------------|--------------------------|
| NAMESPACE |    NAME    | TARGET PORT |           URL            |
|-----------|------------|-------------|--------------------------|
| default   | hello-node |        8080 | http://192.168.49.2:30181 |
|-----------|------------|-------------|--------------------------|
🏃  Starting tunnel for service hello-node.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | hello-node |             | http://127.0.0.1:64002 |
|-----------|------------|-------------|------------------------|
🎉  Opening service default/hello-node in default browser...
❗  Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
```



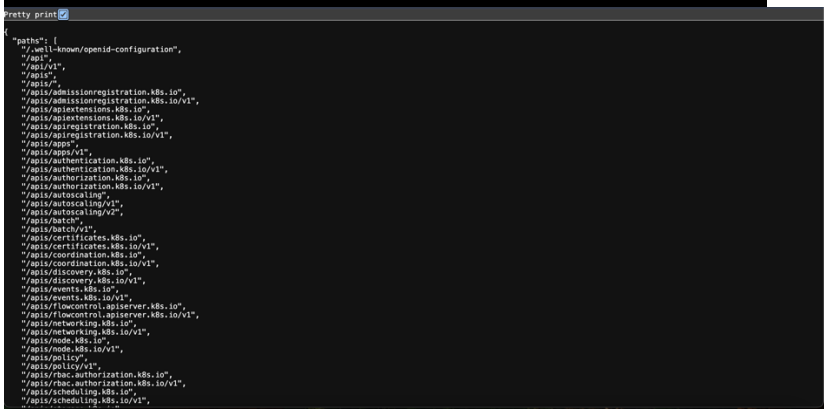
Deploy An App:

I created the bootcamp deployment through terminal

```
aaron@MacBookAir Cloud-Data-Centres % kubectl create deployment kubernetes-bootc
amp --image=gcr.io/google-samples/kubernetes-bootcamp:v1


deployment.apps/kubernetes-bootcamp created
aaron@MacBookAir Cloud-Data-Centres % kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
hello-node            1/1     1            1           10m
kubernetes-bootcamp   1/1     1            1           35s
aaron@MacBookAir Cloud-Data-Centres %
```

I then hosted the proxy

```
aaron@MacBookAir Cloud-Data-Centres % kubectl proxy


Starting to serve on 127.0.0.1:8001
```



Then I got the pod name and stored it:

```
aaron@MacBookAir Cloud-Data-Centres % export POD_NAME=$(kubectl get pods -o go-t
emplate --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')
echo Name of the Pod: $POD_NAME
Name of the Pod: hello-node-c74958b5d-pdnc6
kubernetes-bootcamp-9bc58d867-8dzvc
```

Explore Your App:

I got the pod name here:

```
aaron@MacBookAir Cloud-Data-Centres % export POD_NAME=$(kubectl get pods --no-he
aders -o custom-columns=":metadata.name" | grep 'bootcamp')

aaron@MacBookAir Cloud-Data-Centres %
```

Here is the output of our application:

```
aaron@MacBookAir Cloud-Data-Centres % curl http://localhost:8001/api/v1/namespac
es/default/pods/$POD_NAME:8080/proxy/
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-9bc58d867-8dzvc | v
=1
```

These are the environment variables

```
aaron@MacBookAir Cloud-Data-Centres % kubectl exec "$POD_NAME" -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=kubernetes-bootcamp-9bc58d867-8dzvc
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
HELLO_NODE_SERVICE_HOST=10.107.47.31
HELLO_NODE_SERVICE_PORT=8080
HELLO_NODE_PORT_8080_TCP=tcp://10.107.47.31:8080
KUBERNETES_PORT_443_TCP_PROTO=tcp
HELLO_NODE_PORT=tcp://10.107.47.31:8080
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
HELLO_NODE_PORT_8080_TCP_PROTO=tcp
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP_PORT=443
HELLO_NODE_PORT_8080_TCP_PORT=8080
HELLO_NODE_PORT_8080_TCP_ADDR=10.107.47.31
KUBERNETES_SERVICE_PORT=443
NPM_CONFIG_LOGLEVEL=info
NODE_VERSION=6.3.1
HOME=/root
```

Now I am starting a bash session:

```
aaron@MacBookAir Cloud-Data-Centres % kubectl exec -ti $POD_NAME -- bash
root@kubernetes-bootcamp-9bc58d867-8dzvc:/#
```

Now we can run a node js application

```
root@kubernetes-bootcamp-9bc58d867-8dzvc:/# cat server.js
var http = require('http');
var requests=0;
var podname= process.env.HOSTNAME;
var startTime;
var host;
var handleRequest = function(request, response) {
  response.setHeader('Content-Type', 'text/plain');
  response.writeHead(200);
  response.write("Hello Kubernetes bootcamp! | Running on: ");
  response.write(host);
  response.end(" | v=1\n");
  console.log("Running On:" ,host, "| Total Requests:", ++requests,"| App Uptime
:", (new Date() - startTime)/1000 , "seconds", "| Log Time:",new Date());
}
var www = http.createServer(handleRequest);
www.listen(8080,function () {
    startTime = new Date();;
    host = process.env.HOSTNAME;
    console.log ("Kubernetes Bootcamp App Started At:",startTime, "| Running On:
 " ,host, "\n" );
});
root@kubernetes-bootcamp-9bc58d867-8dzvc:/#
root@kubernetes-bootcamp-9bc58d867-8dzvc:/#
```

Expose your app publicly:

Here I am just listing the current pods

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
hello-node-c74958b5d-pdnc6             1/1     Running   0          27m
kubernetes-bootcamp-9bc58d867-8dzvc    1/1     Running   0          18m
```

Here I'm exposing the bootcamp to external traffic

```
aaron@MacBookAir Cloud-Data-Centres % kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080


service/kubernetes-bootcamp exposed
```

We now have a Kubernetes bootcamp service

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get services


NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
hello-node           LoadBalancer   10.107.47.31    <pending>     8080:30181/TCP   27m
kubernetes           ClusterIP      10.96.0.1       <none>        443/TCP          35m
kubernetes-bootcamp  NodePort       10.98.112.174   <none>        8080:32043/TCP   92s
aaron@MacBookAir Cloud-Data-Centres %
```

This is the Kubernetes bootcamp description

```
aaron@MacBookAir Cloud-Data-Centres % kubectl describe services/kubernetes-bootcamp


Name:                     kubernetes-bootcamp
Namespace:                default
Labels:                   app=kubernetes-bootcamp
Annotations:              <none>
Selector:                 app=kubernetes-bootcamp
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.98.112.174
IPs:                      10.98.112.174
Port:                     <unset>  8080/TCP
TargetPort:               8080/TCP
NodePort:                 <unset>  32043/TCP
Endpoints:                10.244.0.6:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:                   <none>
```

Now I’m creating a variable for the node port

```
aaron@MacBookAir Cloud-Data-Centres % export NODE_PORT="$(kubectl get services/kubernetes-bootcamp -o go-template=
'{{(index .spec.ports 0).nodePort}}')"
echo "NODE_PORT=$NODE_PORT"
NODE_PORT=32043
```

Then I deployed it

```
aaron@MacBookAir Cloud-Data-Centres % curl http://127.0.0.1:$NODE_PORT"
dquote>
```

This is the deployment description

```
Name:                   kubernetes-bootcamp
Namespace:              default
CreationTimestamp:      Fri, 14 Mar 2025 11:37:24 +0000
Labels:                 app=kubernetes-bootcamp
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=kubernetes-bootcamp
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=kubernetes-bootcamp
  Containers:
   kubernetes-bootcamp:
    Image:         gcr.io/google-samples/kubernetes-bootcamp:v1
    Port:          <none>
    Host Port:     <none>
    Environment:   <none>
    Mounts:        <none>
  Volumes:         <none>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   kubernetes-bootcamp-9bc58d867 (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  42m   deployment-controller  Scaled up replica set kubernetes-bootcamp-9bc58d867 from
 0 to 1
```

Here I am getting the pod name and labelling it

```
aaron@MacBookAir Cloud-Data-Centres % export POD_NAME="$(kubectl get pods -o go-
template --template '{{range .items}}{{.metadata.name}}{{"\n"}}{{end}}')"
echo "Name of the Pod: $POD_NAME"
Name of the Pod: kubernetes-bootcamp-9bc58d867-8dzvc
aaron@MacBookAir Cloud-Data-Centres % kubectl label pods "$POD_NAME" version=v1


pod/kubernetes-bootcamp-9bc58d867-8dzvc labeled
```

Now I can query the pod using the new label

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get pods -l version=v1


NAME                                   READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-9bc58d867-8dzvc   1/1     Running   0          50m
aaron@MacBookAir Cloud-Data-Centres %
```

Now I can use the label to delete as well

```
aaron@MacBookAir Cloud-Data-Centres % kubectl delete service -l app=kubernetes-b
ootcamp


service "kubernetes-bootcamp" deleted
aaron@MacBookAir Cloud-Data-Centres %
```

Scale Your App:

We have the bootcamp deployment and replica here

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get deployments


NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   1/1     1            1           55m
aaron@MacBookAir Cloud-Data-Centres % kubectl get rs


NAME                             DESIRED   CURRENT   READY   AGE
kubernetes-bootcamp-9bc58d867   1         1         1       56m
```

We now have 4 replicas of the bootcamp

```
aaron@MacBookAir Cloud-Data-Centres % kubectl scale deployments/kubernetes-bootc
amp --replicas=4


deployment.apps/kubernetes-bootcamp scaled
aaron@MacBookAir Cloud-Data-Centres % kubectl get deployments


NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   4/4     4            4           57m
aaron@MacBookAir Cloud-Data-Centres %
```

Each node now has a different IP address

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get pods -o wide


NAME                                   READY   STATUS    RESTARTS   AGE   IP
       NODE       NOMINATED NODE   READINESS GATES
kubernetes-bootcamp-9bc58d867-8dzvc   1/1     Running   0          58m   10.244.
0.6    minikube   <none>           <none>
kubernetes-bootcamp-9bc58d867-mdv5k   1/1     Running   0          59s   10.244.
0.10   minikube   <none>           <none>
kubernetes-bootcamp-9bc58d867-t4wwc   1/1     Running   0          59s   10.244.
0.8    minikube   <none>           <none>
kubernetes-bootcamp-9bc58d867-zrrdq   1/1     Running   0          59s   10.244.
0.9    minikube   <none>           <none>
```

Here I am getting the node port

```
aaron@MacBookAir Cloud-Data-Centres % export NODE_PORT="$(kubectl get services/k
ubernetes-bootcamp -o go-template='{{(index .spec.ports 0).nodePort}}')"


aaron@MacBookAir Cloud-Data-Centres % echo NODE_PORT=$NODE_PORT


NODE_PORT=30904
aaron@MacBookAir Cloud-Data-Centres %
```

Now I have it running

```
aaron@MacBookAir Cloud-Data-Centres % kubectl get pods

NAME                                   READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-9bc58d867-8dzvc   1/1     Running   0          63m
kubernetes-bootcamp-9bc58d867-mdv5k   1/1     Running   0          6m20s
kubernetes-bootcamp-9bc58d867-t4wwc   1/1     Running   0          6m20s
kubernetes-bootcamp-9bc58d867-zrrdq   1/1     Running   0          6m20s
aaron@MacBookAir Cloud-Data-Centres % curl http://127.0.0.1:49611
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-9bc58d867-8dzvc | v
=1
aaron@MacBookAir Cloud-Data-Centres %
```

Now we can scale deployments back down

```
aaron@MacBookAir Cloud-Data-Centres % kubectl scale deployments/kubernetes-bootc
amp --replicas=2


deployment.apps/kubernetes-bootcamp scaled
```

Update Your App:

Here I am updating the image for the bootcamp

```
aaron@MacBookAir Cloud-Data-Centres % kubectl set image deployments/kubernetes-b
ootcamp kubernetes-bootcamp=docker.io/jocatalin/kubernetes-bootcamp:v2


deployment.apps/kubernetes-bootcamp image updated
```

As can be seen it is running the new image

```
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://c3c17ff2a36f941c01b706e8299bfa949c2080a0d2d2fe94a383c2c8db7c54ab
    Image:          docker.io/jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-bootcamp@sha256:fb1a3ced00cecfc1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
```

Here I'm setting a new image

```
aaron@MacBookAir Cloud-Data-Centres % kubectl set image deployments/kubernetes-b
ootcamp kubernetes-bootcamp=gcr.io/google-samples/kubernetes-bootcamp:v10
deployment.apps/kubernetes-bootcamp image updated
aaron@MacBookAir Cloud-Data-Centres % kubectl get deployments


NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   2/2     1            2           70m
aaron@MacBookAir Cloud-Data-Centres % kubectl get pods
NAME                                  READY   STATUS             RESTARTS   AGE
kubernetes-bootcamp-5c4f7cb664-hkdm7   1/1     Running            0          4m4
8s
kubernetes-bootcamp-5c4f7cb664-jv6k8   1/1     Running            0          4m4
5s
kubernetes-bootcamp-75bd5fd495-5vrpd   0/1     ImagePullBackOff   0          26s
aaron@MacBookAir Cloud-Data-Centres %
```

One of the images didn't update as expected so we rolled it back

```
aaron@MacBookAir Cloud-Data-Centres % kubectl rollout undo deployments/kubernete
s-bootcamp


deployment.apps/kubernetes-bootcamp rolled back
```

Then finally cleaning up and deleting the bootcamp

```
aaron@MacBookAir Cloud-Data-Centres % kubectl delete deployments/kubernetes-boot
camp services/kubernetes-bootcamp


deployment.apps "kubernetes-bootcamp" deleted
service "kubernetes-bootcamp" deleted
```