Kubernetes Basics

Concepts

Pod

- group of **one or more containers**
- tied together for the purpose of networking and administration.
- **By default, a pod is only accessible by its internal IP address within the kubernetes cluster**
    - **To make a container, say *hello-node* container, accessible outside kubernetes cluster, the pod has to be exposed as a kubernetes service**

Kubernetes Deployment

- checks on the health of your pod and restarts the Pod's container if it terminates.
- Deployments are the recommended way to manage the creation and scaling of pods

---

Hands on

- Use the kubectl create command to create a Deployment that manages a Pod. The Pod runs a Container based on the provided Docker image.
    - `kubectl create deployment hello-node --image=gcr.io/hello-minikube-zero-install/hello-node`
- **View the Deployment:**
    - `kubectl get deployments`
        - Output

```
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-7676b5fb8d-rmh97   1/1     Running   0          6m7s
```

- **View pods:**
    - `kubectl get pods`
        - Output

```
NAME                          READY   STATUS    RESTARTS   AGE
hello-node-7676b5fb8d-rmh97   1/1     Running   0          8m22s
```

- **View cluster events**
    - `kubectl get events`

- Output

```
LAST SEEN   TYPE     REASON                    OBJECT
MESSAGE
<unknown>   Normal   Scheduled                 pod/hello-node-
7676b5fb8d-rmh97    Successfully assigned default/hello-node-
7676b5fb8d-rmh97 to minikube
11m         Normal   Pulling                   pod/hello-node-
7676b5fb8d-rmh97    Pulling image "gcr.io/hello-minikube-zero-
install/hello-node"
8m11s       Normal   Pulled                    pod/hello-node-
7676b5fb8d-rmh97    Successfully pulled image "gcr.io/hello-
minikube-zero-install/hello-node"
8m10s       Normal   Created                   pod/hello-node-
7676b5fb8d-rmh97    Created container hello-node
8m10s       Normal   Started                   pod/hello-node-
7676b5fb8d-rmh97    Started container hello-node
11m         Normal   SuccessfulCreate          replicaset/hello-
node-7676b5fb8d   Created pod: hello-node-7676b5fb8d-rmh97
11m         Normal   ScalingReplicaSet         deployment/hello-
node              Scaled up replica set hello-node-7676b5fb8d to 1
37m         Normal   Starting                  node/minikube
Starting kubelet.
37m         Normal   NodeHasSufficientMemory   node/minikube
Node minikube status is now: NodeHasSufficientMemory
37m         Normal   NodeHasNoDiskPressure     node/minikube
Node minikube status is now: NodeHasNoDiskPressure
37m         Normal   NodeHasSufficientPID      node/minikube
Node minikube status is now: NodeHasSufficientPID
37m         Normal   NodeAllocatableEnforced   node/minikube
Updated Node Allocatable limit across pods
36m         Normal   RegisteredNode            node/minikube
Node minikube event: Registered Node minikube in Controller
36m         Normal   Starting                  node/minikube
Starting kube-proxy.
```

- **Create a service**

1. Expose the Pod to the public internet using the kubectl expose command:

```
> kubectl expose deployment hello-node --type=LoadBalancer --port=8080

Output:
service/hello-node exposed
```

The `--type=LoadBalancer` flag indicates that you want to **expose your Service outside of the cluster**.

2. View the Services

```
kubectl get services
```

```
 Output:
 NAME         TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)
AGE
 hello-node   LoadBalancer   10.96.69.90   <pending>     8080:31950/TCP
53s
 kubernetes   ClusterIP      10.96.0.1     <none>        443/TCP
50m
```

Note: On cloud providers that support load balancers, an external IP address would be provisioned to access the Service. On Minikube, the LoadBalancer type makes the Service accessible through the minikube service command.

3. On Minikube, the LoadBalancer type makes the Service accessible through the minikube service command.

```
minikube service hello-node
```

Sample output:

NAMESPACE	NAME	TARGET PORT	URL
default	hello-node		http://192.168.99.100:31950

- **Cleanup**

```
$ kubectl delete service hello-node
```

```
service "hello-node" deleted
```

```
$ kubectl get services
```

```
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   76m
```

```
$ kubectl delete deployment hello-node
```

```
deployment.apps "hello-node" deleted
```

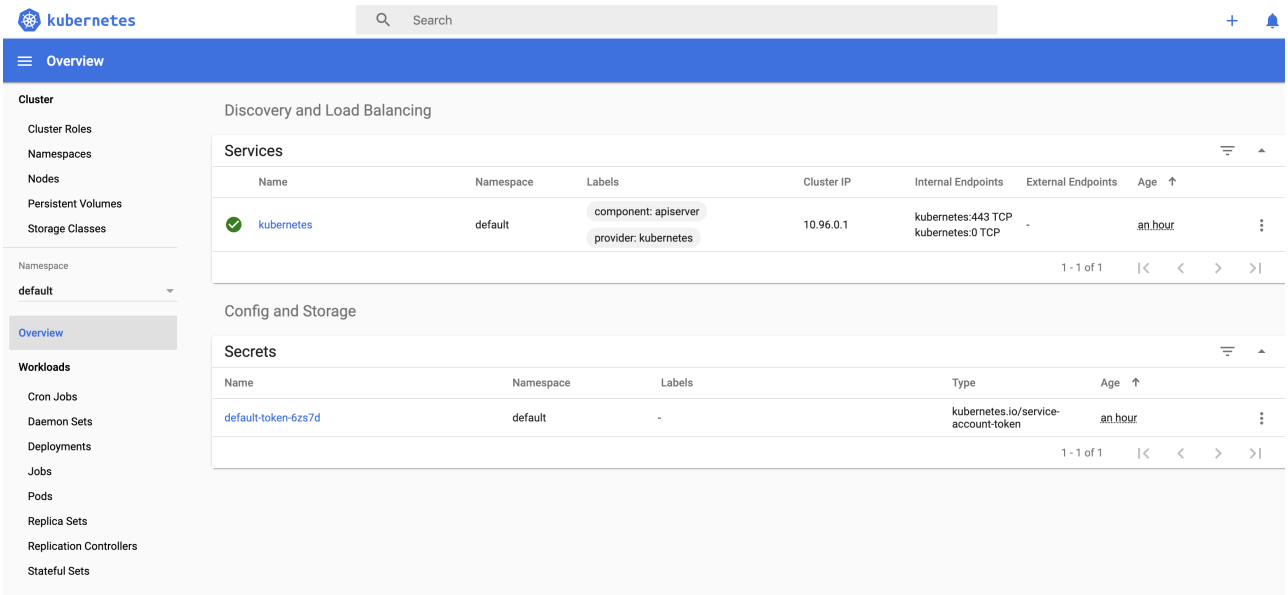
```
$ kubectl get deployments
```

```
No resources found.
```

- **Kubernetes Dashboard**

> minkube dashboard

```
🔌  Enabling dashboard ...
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:61643/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```



Quick Commands

- To check **if virtualization is supported on the underlying machine**
  - `sysctl -a | grep -E --color 'machdep.cpu.features|VMX'`
  - The output of above command **should be having VMX**
- **Install `minikube`**
  - Pre-requisites
    - Install `kubectl`
    - Install a Hypervisor {HyperKit or VirtualBox or VMWareFusion}
  - `brew cask install minikube`
    - ALternatively
      - `curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-`

`darwin-amd64 \ && chmod +x minikube`

- AND `sudo mv minikube /usr/local/bin`

- **Cleanup Local State**
  - If you have previously installed minikube, and run:
    - `minikube start`
    - **Output**:

```
   😄  minikube v1.4.0 on Darwin 10.14.5
   💿  Downloading VM boot image ...
       > minikube-v1.4.0.iso.sha256: 65 B / 65 B [--------------]
100.00% ? p/s 0s
       > minikube-v1.4.0.iso: 135.73 MiB / 135.73 MiB [] 100.00%
2.00 MiB p/s 1m8s
   🔥  Creating virtualbox VM (CPUs=2, Memory=2000MB, Disk=20000MB)
...
   🐳  Preparing Kubernetes v1.16.0 on Docker 18.09.9 ...
   💾  Downloading kubeadm v1.16.0
   💾  Downloading kubelet v1.16.0
   🚜  Pulling images ...
   🚀  Launching Kubernetes ...
   ⌛  Waiting for: apiserver proxy etcd scheduler controller dns
   🏄  Done! kubectl is now configured to use "minikube"
```

  - And this command returns an error:
    - `machine does not exist`
    - You need to clear minikube's local state:
      - `minikube delete`

References

- kubectl commands
  - https://kubernetes.io/docs/reference/kubectl/overview/
-