Using Multi-Node Clusters

Overview

- This tutorial will show you how to start a multi-node clusters on minikube and deploy a service to it.

Prerequisites

- minikube 1.10.1 or higher
- kubectl

Caveat

Default host-path volume provisioner doesn't support multi-node clusters (#12360). To be able to provision or claim volumes in multi-node clusters, you could use CSI Hostpath Driver addon.

Tutorial

- Start a cluster with 2 nodes in the driver of your choice:

```
minikube start --nodes 2 -p multinode-demo
```

```
😄  [multinode-demo] minikube v1.18.1 on Opensuse-Tumbleweed
✨  Automatically selected the docker driver
👍  Starting control plane node multinode-demo in cluster multinode-demo
🔥  Creating docker container (CPUs=2, Memory=8000MB) ...
🐳  Preparing Kubernetes v1.20.2 on Docker 20.10.3 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass

👍  Starting node multinode-demo-m02 in cluster multinode-demo
🔥  Creating docker container (CPUs=2, Memory=8000MB) ...
🌐  Found network options:
    ▪ NO_PROXY=192.168.49.2
🐳  Preparing Kubernetes v1.20.2 on Docker 20.10.3 ...
    ▪ env NO_PROXY=192.168.49.2
🔎  Verifying Kubernetes components...
🏄  Done! kubectl is now configured to use "multinode-demo" cluster and "default" namesp
```

- Get the list of your nodes:

```
kubectl get nodes
```

```
NAME                 STATUS   ROLES                  AGE   VERSION
multinode-demo       Ready    control-plane,master   99s   v1.20.2
multinode-demo-m02   Ready    <none>                 73s   v1.20.2
```

- You can also check the status of your nodes:

```
minikube status -p multinode-demo
```

```
kubelet: Running
apiserver: Running
kubeconfig: Configured

multinode-demo-m02
type: Worker
host: Running
kubelet: Running
```

- Deploy our hello world deployment:

```
kubectl apply -f hello-deployment.yaml
```

```
deployment.apps/hello created
```

```
kubectl rollout status deployment/hello
```

```
deployment "hello" successfully rolled out
```

- Deploy our hello world service, which just spits back the IP address the request was served from:

```
kubectl apply -f hello-svc.yaml
```

```
service/hello created
```

- Check out the IP addresses of our pods, to note for future reference

```
kubectl get pods -o wide
```

```
NAME                     READY   STATUS    RESTARTS   AGE   IP           NODE
hello-695c67cf9c-bzrzk   1/1     Running   0          22s   10.244.1.2   multinode-demo-
hello-695c67cf9c-frcvw   1/1     Running   0          22s   10.244.0.3   multinode-demo
```

- Look at our service, to know what URL to hit

```
minikube service list -p multinode-demo
```

```
|-------------|------------|--------------|---------------------------|
|  NAMESPACE  |    NAME    | TARGET PORT  |            URL            |
|-------------|------------|--------------|---------------------------|
| default     | hello      |           80 | http://192.168.49.2:31000 |
| default     | kubernetes | No node port |                           |
| kube-system | kube-dns   | No node port |                           |
|-------------|------------|--------------|---------------------------|
```

- Let's hit the URL a few times and see what comes back

```
curl  http://192.168.49.2:31000
```

```
Hello from hello-695c67cf9c-frcvw (10.244.0.3)

curl  http://192.168.49.2:31000
Hello from hello-695c67cf9c-bzrzk (10.244.1.2)

curl  http://192.168.49.2:31000
Hello from hello-695c67cf9c-bzrzk (10.244.1.2)

curl  http://192.168.49.2:31000
Hello from hello-695c67cf9c-frcvw (10.244.0.3)
```

- Multiple nodes!

- Referenced YAML files

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 100%
  selector:
    matchLabels:
      app: hello
  template:
    metadata:
      labels:
        app: hello
    spec:
      affinity:
        # ⬇⬇⬇ This ensures pods will land on separate hosts
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions: [{ key: app, operator: In, v
              topologyKey: "kubernetes.io/hostname"
      containers:
        - name: hello-from
          image: pbitty/hello-from:latest
          ports:
            - name: http
              containerPort: 80
      terminationGracePeriodSeconds: 1
```

Last modified February 13, 2023: add multi-node cluster doc for the csi hostpath driver (874de40ca)