Kubernetes cluster Administration notes

Kubectl

Show yaml

```
kubectl get deployments/bookstack -o yaml
```

Scale

```
kubectl scale deployment/name --replicas=2
```

Show all resources

```
for i in $(kubectl api-resources --verbs=list --namespaced -o name | grep -v "events.events.k8s.io" | grep -v "events" | sort | uniq)

do echo "Resource:" $i
kubectl get $i

done
```

Drain nodes

Drain node

```
kubectl drain host.name.local --ignore-daemonsets
```

Put node back to ready

```
kubectl uncordon host.name.local
```

Replace a new node

Delete a node

```
kubectl delete node [node_name]
```

Generate a new token:

```
kubeadm token generate
```

List the tokens:

```
kubeadm token list
```

Print the kubeadm join command to join a node to the cluster:

```
kubeadm token create [token_name] --ttl 2h --print-join-command
```

Create etcd snapshot

Get the etcd binaries:

```
wget https://github.com/etcd-io/etcd/releases/download/v3.3.12/etcd-v3.3.12-linux-amd64.tar.gz
```

Unzip the compressed binaries:

```
tar xvf etcd-v3.3.12-linux-amd64.tar.gz
```

Move the files into `/usr/local/bin`:

```
mv etcd-v3.3.12-linux-amd64/etcd* /usr/local/bin
```

Take a snapshot of the etcd datastore using etcdctl:

```
ETCDCTL_API=3 etcdctl snapshot save snapshot.db --cacert /etc/kubernetes/pki/etcd/server.crt --cert /etc/kubernetes/pki/etcd/ca.crt --key /etc/kubernetes/pki/etcd/ca.key
```

View the help page for etcdctl:

```
ETCDCTL_API=3 etcdctl --help
```

Browse to the folder that contains the certificate files:

```
cd /etc/kubernetes/pki/etcd/
```

View that the snapshot was successful:

```
ETCDCTL_API=3 etcdctl --write-out=table snapshot status snapshot.db
```

Backup etcd snapshot

Zip up the contents of the etcd directory:

```
tar -zcvf etcd.tar.gz /etc/kubernetes/pki/etcd
```

Create pods on specific node(s) :

Create a DaemonSet from a YAML spec :

```
apiVersion: apps/v1beta2
kind: DaemonSet
metadata:
  name: ssd-monitor
spec:
  selector:
    matchLabels:
      app: ssd-monitor
  template:
    metadata:
      labels:
        app: ssd-monitor
    spec:
      nodeSelector:
        disk: ssd
      containers:
```

```
    - name: main
      image: linuxacademycontent/ssd-monitor
```

```
kubectl create -f ssd-monitor.yaml
```

Label a node to identify it and create a pod on it :

```
kubectl label node node02.myhypervisor.ca disk=ssd
```

Remove a label from a node:

```
kubectl label node node02.myhypervisor.ca disk-
```

Change the label on a node from a given value to a new value :

```
kubectl label node node02.myhypervisor.ca disk=hdd --overwrite
```

> If you override an existing label, pods running with the previous label will be terminated

Migration notes

Connect to bash

```
kubectl exec -it pod/nextcloud /bin/bash
```

Restore MySQL data

```
kubectl exec -it nextcloudsql-0 -- mysql -u root -pPASSWORD nextcloud_db < backup.sql
```

Revision #10
Created 19 April 2019 18:54:36 by Dave
Updated 19 March 2020 19:15:34 by Dave