**### COMMANDS ###**

=======================================================================

**# # Cluster Information**

========================================================================

#ICluster

kubectl cluster-info

lsof -i

kubectl get nodes #Show available nodes list

kubectl get pods #Show available pods list

kubectl get cs #Show cluster information

kubectl get pods --all-namespaces #Show namespace information

kubectl get service -o wide

#Show pods and exposing port no showing, it will show load balancer DNS

kubectl config view #Display username and pass word

#Deployment

kubectl get deployments #Show available Deployments list

kubectl get deploy #show the deployments details

kubectl edit deploy/deployment-name #Edit the text only it will upgrade automatically

#Replicas

kubectl get rs #show the replicas details

#Services

kubectl get service (OR) kubectl get svc #show the exposed port no

# Run & Create Pod Commands line

kubectl run --image=nginx <Container Name> # Create Pod

kubectl run sample-nginx –image=nginx –replicas=2 –port=80 # Create Pod

# Logs information

kubectl logs <Container name> #Show pod logs

# Expose service after creation of Pod

kubectl expose pod PodName --port=80 --target-port=80 --type=NodePort

### Manual Expose the container port outside world

kubectl expose pod PodName --port=80 --target-port=80 --type=ClusterIP

### Manual Expose the container port inside cluster only

kubectl expose pod PodName --port=80 --target-port=80 --type=LoadBalancer

### Manual Expose the container port Load balancer

# Show the information

kubectl describe pod PodName ## Show the web pod details

kubectl describe svc servicename ## It will show node pod exposed port no

#Scale the pods

kubectl scale rs/web-rs --replicas=10 ##### increase the replicas 10 no

kubectl scale rs/web-rs --replicas=2 ##### decrease the replicas from 10 to 2 no

kubectl scale deploy/deploymentname --replicas=5 ##### increase the replicas 5 no

# Create POD, RS, DEPLOY, through YAML File

kubectl create -f db-pod.yml -f db-service.yml -f web-pod.yml -f web-service.yml ## Create Pods & Services

# Delete

kubectl delete pod myweb-999d5fcbc-n5k55 #### delete the pods

kubectl delete deployments <Name> ### delete the all deployments, replicas, pods

kubectl delete svc <Name> ### delete services

kubectl get pods -w ### it will comes back undo recently deletion pods

# Volumes

cat /etc/export #### show the NFS Data in Master

/opt/data 10.245.1.2/24 rw, sync, no\_root\_squash, no\_all\_squash

cat /etc/fstab #### show the NFS path in node

/opt/data 10.245.1.2/24 rw, sync, no\_root\_squash, no\_all\_squash

vi /etc/fstab #### Mount NFS Volume from Master to Node

10.245.1.2:/opt/data /usr/backup nfs rw, sync, hard, user 0 0

#Rolling update

kubectl set image deployment deployment-name container-name=imagename:2

#update the container imagename:1 to imagename:2

kubectl rollout history deployment/deployment-name

#show the rolling update history revisions

kubectl rollout undo deployment/deployment-name

#undo option

kubectl rollout resume deployment/deployment-name

#resume option

kubectl rollout status deployment/deployment-name

#it will show status of deployment

# Port No

3306 = MySQL # Change location:

8080 = Jenkins # Change location

80 = HTTP # Change location

22 = SSH # Change location

443 = HTTPS # Change location

# Config-Maps

Kubectl get configmap # Show the config

Kubectl get configmaps -o yaml # Show yaml file parameters.

enu

# Use-full Scripts

While true; do curl node1:$NODE\_PORT; echo “ ” ; sleep1 : done #aaa

Snapshot etcd data

The **snapshot** action of the etcd charm allows the operator to snapshot a running cluster’s data for use in cloning, backing up, or migrating to a new cluster.

**juju run-action etcd/0 snapshot**

This will create a snapshot in **/home/ubuntu/etcd-snapshots** by default.

Restore etcd data

The etcd charm is capable of restoring its data from a cluster-data snapshot via the **restore** action. This comes with caveats and a very specific path to restore a cluster: The cluster must be in a state of only having a single member. So it’s best to deploy a new cluster using the etcd charm, without adding any additional units.

**juju deploy etcd new-etcd**

The above code snippet will deploy a single unit of etcd, as ‘new-etcd’

**juju run-action etcd/0 restore target=/mnt/etcd-backups**

Once the restore action has completed, evaluate the cluster health. If the unit is healthy, you may resume scaling the application to meet your needs.

* **param** target: destination directory to save the existing data.
* **param** skip-backup: Don’t backup any existing data.

## Migrating an etcd cluster

Using the above snapshot and restore operations, migrating etcd is a fairly easy task.

**Step 1:** Snapshot your existing cluster. This is encapsulated in the **snapshot** action.

**juju run-action etcd/0 snapshot**

Results:

**Action queued with id: b46d5d6f-5625-4320-8cda-b611c6ae580c**

**Step 2:** Check the status of the action so you can grab the snapshot and verify the sum. The **copy.cmd**result output is a copy/paste command for you to download the exact snapshot that you just created.

Download the snapshot archive from the unit that created the snapshot and verify the sha256 sum

**juju show-action-output b46d5d6f-5625-4320-8cda-b611c6ae580c**

Results:

**results:**

**copy:**

**cmd: juju scp etcd/0:/home/ubuntu/etcd-snapshots/etcd-snapshot-2016-11-09-02.41.47.tar.gz**

**.**

**snapshot:**

**path: /home/ubuntu/etcd-snapshots/etcd-snapshot-2016-11-09-02.41.47.tar.gz**

**sha256: 1dea04627812397c51ee87e313433f3102f617a9cab1d1b79698323f6459953d**

**size: 68K**

**status: completed**

Copy the snapshot to the local disk and then check the sha256sum.

**juju scp etcd/0:/home/ubuntu/etcd-snapshots/etcd-snapshot-2016-11-09-02.41.47.tar.gz .**

**sha256sum etcd-snapshot-2016-11-09-02.41.47.tar.gz**

**Step 3:** Deploy the new cluster leader, and attach the snapshot:

**juju deploy etcd new-etcd --resource snapshot=./etcd-snapshot-2016-11-09-02.41.47.tar.gz**

**Step 4:** Reinitialize the master with the data from the resource we just attached in step 3.

**juju run-action new-etcd/0 restore**

please schedule at 5.00Pm