kubernetes

Pre-reading

https://kubernetes.io/docs/concepts/overview/

Creating a 3 node Cluster

We need 3 Ubuntu Server 22.04 VMs: master, worker1, and worker2. For all VMs use:

- 2 CPU cores.
- 4 GB of RAM.
- 200 GB HDD.
- Bridged Network adapter.

After creating the VMs use the following guide to setup your cluster [use putty to facilitate using pasting commands]:

https://www.cherryservers.com/blog/install-kubernetes-on-ubuntu

A pdf copy of this guide is available on Moodle.

https://moodle.najah.edu/mod/resource/view.php?id=660758

Note: if you use any other version of ubuntu or any other Linux distribution please do your own research on how to setup kubernetes on your VMs.

Web app with mongo database

All work will be done on the master node.

git clone https://github.com/IbrahimTalaatAh/kubernetes.git

cd kubernetes/

kubectl apply -f mongo-config.yaml

kubectl apply -f mongo-secret.yaml

kubectl apply -f mongo.yaml

kubectl apply -f webapp.yaml

Open the webapp website from your host machine using any node IP with port 30100.

Managing your deployment

Use the following commands to get information about your deployment details:
kubectl get deployments
kubectl get pods
kubectl get services
kubectl describe <type> <type name>
<type> could be deployment, service, pod, ...

Edit your deployment to get 6 webapp pods, use the **kubectl edit deployment webapp-deployment** command for that.

Check how many pods are running in the cluster.

Remove all deployments with the **kubectl delete deployment** command and do the same for the services.

Open mongo.yaml and change the replicas to 4 and do the same for webapp.yaml setting the replicas to 10 and redeploy mongo then webapp.

Get running pods with the **-o wide** option that can be used with any kubectl get command.

Another way to edit your deployment is using the following commands:
kubectl scale deployment/webapp-deployment --replicas=5
kubectl scale deployment/mongo-deployment --replicas=2
show running pods after that.

Use **kubectl get rss** to check the ReplicaSet of your deployments. Notice desired and current.
We decided to update mongo to version 5.0.14 from the 5.0 we are using now. To do that use the following command:
kubectl set image deployment/mongo-deployment mongodb=mongo:5.0.14

Check the status using the commands:
kubectl rollout status deployment/mongo-deployment

kubectl describe deployments notice the mongo version in the output.

You can also rollback and return to a previous version of the container image. Let's rollback mongo to version 5.0 by editing the deployment using the command **kubectl edit deployment mongo-deployment** and at the same time increasing the number of pods by 2. run **kubectl get pods** and **kubectl get rs** to see things in action (be fast).

To check the history of rollouts use the command:
kubectl rollout history deployment/mongo-deployment
you can check the details of a specific revision using the command:
kubectl rollout history deployment/mongo-deployment --revision=2

References:

https://kubernetes.io/docs/home/

https://www.youtube.com/watch?v=s_o8dwzRlu4&t=1528s

https://www.youtube.com/watch?v=U1VzcjCB_sY

https://www.youtube.com/watch?v=_WW16Sp8-Jw

END