Kubernetes

Before running commands make sure the following 3 steps

- [ ] To start using Kubernetes:
- [ ] First start Minikube container in docker
- [ ] Then run command minikube start

To start minikube:
minikube start

Create a deployment pod without yaml
kubectl create deployment nginx-depl --image=nginx

To get list of nodes:
Kubectl get nodes

To get status of the nodes
minikube status

To check kubectl version:
kubectl version

Kubernetes create using yaml:
kubectl create -f pod.yaml   (or) kubectl apply -f pod.yaml

So, what is difference b/w using create and apply: kubectl create -f pod.yaml   (vs) kubectl apply -f pod.yaml

kubectl create -f pod.yaml:

- This command creates the resource defined in the YAML file if it doesn't already exist.
- If the resource already exists, it will return an error.
- It is suitable for initial resource creation or when you want to ensure that a resource is created and not modified if it already exists.

kubectl apply -f pod.yaml:

- This command creates the resource if it doesn't exist or updates it if it already exists.
- If the resource doesn't exist, it will be created.
- If the resource exists, Kubernetes will perform a "declarative" update, meaning it will attempt to apply the changes specified in the YAML file to the existing resource without deleting and recreating it.
- It is suitable for ongoing management of resources, allowing you to easily modify and update existing resources without manual intervention.

To get a list of services:
kubectl get services:

Get pods details:
kubectl get pods

To delete a pod if created normally :
Kubectl delete kindofpodname nginx

To delete a pod created using yaml file:
Kubectl delete -f file.yaml

To create a service to run website:
minikube service website-deployment --url

After creating pod expose port:
kubectl expose deployment website-deployment --type=NodePort --port=80

To view details about the containers in a pod:
kubectl describe pod <pod-name>

If there is only one container in a pod, then we can login to that container using:

```
kubectl exec -it <pod-name> -- /bin/bash
```

To login to container in a pod:

```
kubectl exec -it <pod-name> --container <container-name> -- /bin/bash
```

This shows the logs of a container

```
Kubectl logs podname -c containername
```

This command allows you to watch the status of pods in real-time, continuously updating the output as changes occur. Here's how it works:

```
kubectl get pod --watch
```

Expose port 80:

```
kubectl expose deployment nginx-depl --type=NodePort --port=80
```

To get deployment pods:

```
kubectl get deployment
```

To get view browser link:

```
minikube service nginx-depl --url
```

To get the logs of a pod:

```
kubectl logs podname
```

Commands from lecture:

kubectl apply commands in order

```
kubectl apply -f mongo-secret.yaml
kubectl apply -f mongo.yaml
kubectl apply -f mongo-configmap.yaml
kubectl apply -f mongo-express.yaml
```

kubectl get commands

kubectl get pod
kubectl get pod --watch
kubectl get pod -o wide
kubectl get service
kubectl get secret
kubectl get all | grep mongodb

kubectl debugging commands

kubectl describe pod mongodb-deployment-xxxxxx
kubectl describe service mongodb-service
kubectl logs mongo-express-xxxxxx

give a URL to external service in minikube

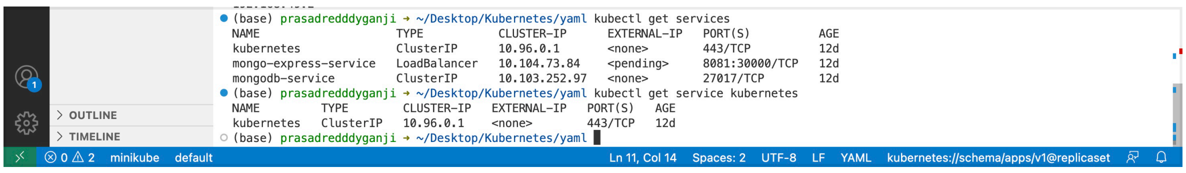
minikube service mongo-express-service

To encrypt a username:
echo -n 'username' | base64

To encrypt passwd:
echo -n 'password' | base64

To get list of services:

kubectl get service <service-name>



```
(base) prasadredddyganji → ~/Desktop/Kubernetes/yaml kubectl get services
NAME                    TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes              ClusterIP      10.96.0.1       <none>        443/TCP          12d
mongo-express-service   LoadBalancer   10.104.73.84    <pending>     8081:30000/TCP   12d
mongodb-service         ClusterIP      10.103.252.97   <none>        27017/TCP        12d
(base) prasadredddyganji → ~/Desktop/Kubernetes/yaml kubectl get service kubernetes
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   12d
(base) prasadredddyganji → ~/Desktop/Kubernetes/yaml
```

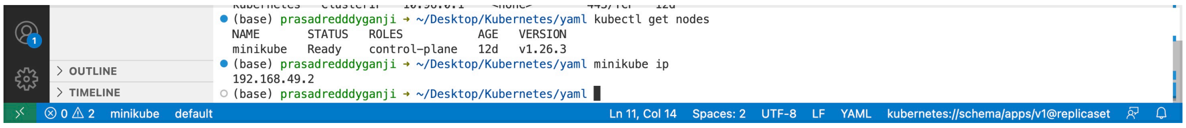
OUTLINE
TIMELINE
0 2 minikube default
Ln 11, Col 14 Spaces: 2 UTF-8 LF YAML kubernetes://schema/apps/v1@replicaset

To get list details about a service

kubectl get service my-service

To get ip of our node ip

minikube ip

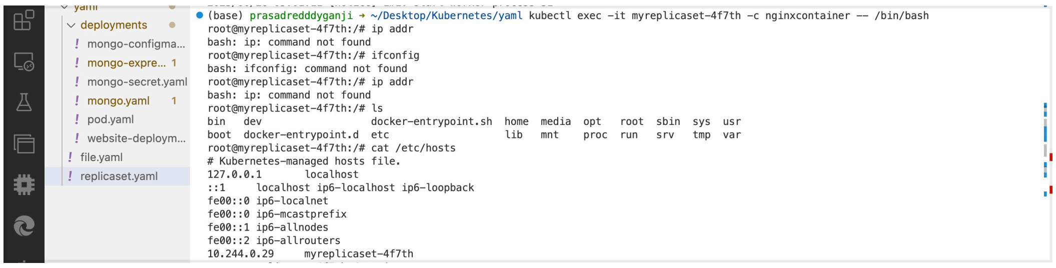


```
(base) prasadredddyganji ➜ ~/Desktop/Kubernetes/yaml kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   12d   v1.26.3
(base) prasadredddyganji ➜ ~/Desktop/Kubernetes/yaml minikube ip
192.168.49.2
(base) prasadredddyganji ➜ ~/Desktop/Kubernetes/yaml
```

OUTLINE
TIMELINE
0 2 minikube default
Ln 11, Col 14 Spaces: 2 UTF-8 LF YAML kubernetes://schema/apps/v1@replicaset

To get ip of container(details of a contain er)

kubectl exec -it myreplicaset-4f7th -c nginxcontainer -- /bin/bash



```
(base) prasadredddyganji ➜ ~/Desktop/Kubernetes/yaml kubectl exec -it myreplicaset-4f7th -c nginxcontainer -- /bin/bash
root@myreplicaset-4f7th:/# ip addr
bash: ip: command not found
root@myreplicaset-4f7th:/# ifconfig
bash: ifconfig: command not found
root@myreplicaset-4f7th:/# ip addr
bash: ip: command not found
root@myreplicaset-4f7th:/# ls
bin   dev                  docker-entrypoint.sh  home  media  opt   root  sbin  sys  usr
boot  docker-entrypoint.d  etc                   lib   mnt    proc  run   srv   tmp  var
root@myreplicaset-4f7th:/# cat /etc/hosts
# Kubernetes-managed hosts file.
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
fe00::0 ip6-mcastprefix
fe00::1 ip6-allnodes
fe00::2 ip6-allrouters
10.244.0.29     myreplicaset-4f7th
```