Create volume

```
docker volume create <volume name >
```

Run the image

```
docker container run -d   --name akshata   -p 8080:3306   -e MYSQL_ROOT_PASSWORD="manager"   -v /Users/tejas/volume/ram:/var/lib/mysql   mysql
```

```
docker container exec -it   <containername > bash
```

For httpd

1st create mkdir
Take. 1 index.html
And 1 Dockerfile
And put the data

```
{  FROM httpd
COPY index.html /usr/local/apache2/htdocs
EXPOSE 80  }
```

Then build

```
docker image build -t myimage .
```

And run

```
docker container run -itd --name web -p 8085:80 myimage
```

## pods

```
# get the list of pods

> kubectl get pod

# keep watching the status of pods

> kubectl get pods --watch

# get more details of every pod
```

```
> kubectl get pods -o wide

# create a pod by using a YAML file


> kubectl create -f <yaml file name>

# get details of selected pod

# > kubectl describe pod mypod

> kubectl describe pod <pod name>

# delete a selected pod
# > kubectl delete pod mypod
> kubectl delete pod <pod name>
```

Pod yaml {

```
apiVersion: v1
kind: Pod
metadata:
  name: pod3
  labels:
    type: frontend
spec:
  containers:
    - name: container1
      image: httpd
      ports:
        - containerPort: 80
```

}

ReplicaSet

```
# get the list of replica sets

> kubectl get rs
>  kubectl create -f <file name>
```

# create a rs
> kubectl create -f <file name>

#delete

 kubectl delete replicaset --all -n default

ReplicaSet yaml {

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rs1
spec:
  # no of replicas to be managed
  replicas: 3

  # criteria to find the pod(s) which belong to the RS
  selector:
    matchLabels:
      type: frontend

  # if pod does not exist, then create them using following template
  template:
    metadata:
      labels:
        type: frontend
    spec:
      containers:
        - name: container1
          image: httpd
          ports:
            - containerPort: 80
```

}

Create service

kubectl create -f <yaml file name>

// for list :-

1) kubectl get service

2) kubectl delete svc <service name>

Last command for access

minikube service <service name>

service.yml

{

```
apiVersion: v1
kind: Service
metadata:
  name: service1
spec:
  type: NodePort
  selector:
    type: frontend
  ports:
    - port: 80
      targetPort: 80
```

}

Dockerfile for node and react

FROM node WORKDIR /srcCOPY . .EXPOSE 3000 CMD node server.js

FROM node WORKDIR /srcCOPY . .EXPOSE 3000 CMD npm start