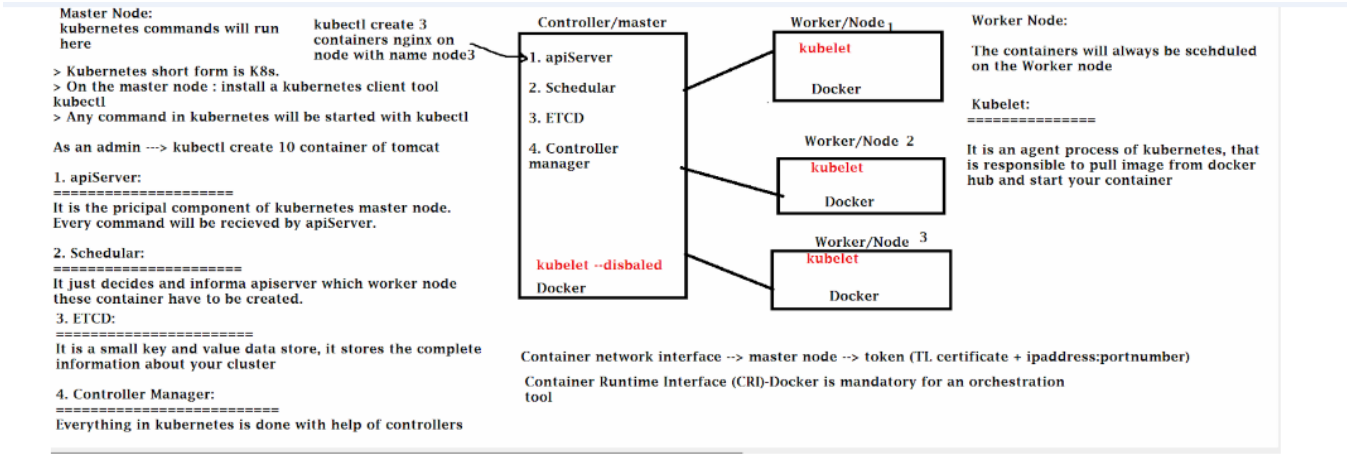
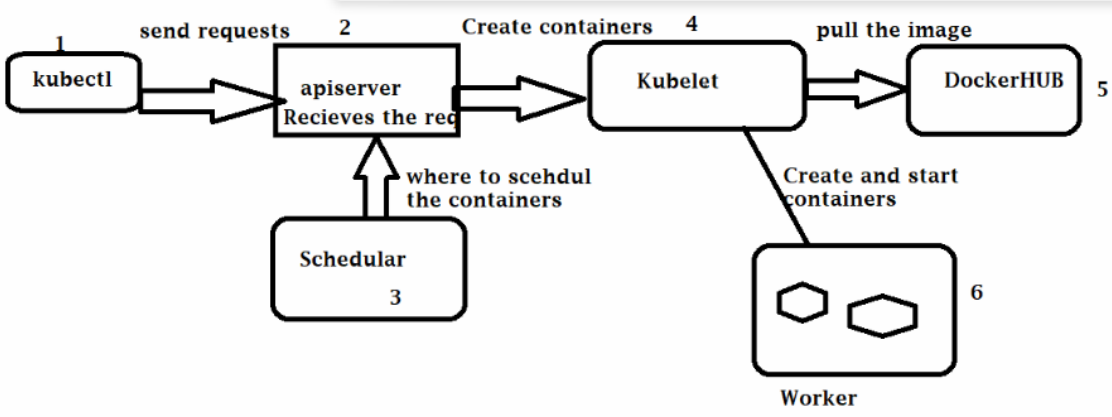
KUBERNETES - POD file, Replica's Scale Up/Down

Kubernetes is a Container Orchestration tool. It Manages multiple containers on multiple VMs. Deletion, creation, movements of containers, and everything else to make the resource reliable and cost effective. Makes load balancing possible

- ➔ Creating multiple container from one single image
- ➔ Desired number of containers are managed by kubernetes. Scale up and scale down the containers
- ➔ Provisioning and deployment of containers across various VMs





1. Perform these Installations steps on the linux server
   → INSTALL/Upgrade DOCKER
   # sudo su -
   # sudo apt-get update
   # sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common
   # curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
   # sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

   # sudo apt-get update ; clear
   # sudo apt-get install -y docker-ce
   # sudo vi /etc/docker/daemon.json
   Press i and insert below content
   {
   "exec-opts": ["native.cgroupdriver=systemd"]
   }
   Save the file.
   sudo service docker restart

   → INSTALL KUBEADM,KUBELET,KUBECTL
   echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
   curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
   sudo apt-get update ; clear
   sudo apt-get install -y kubelet kubeadm kubectl

   → This step is only needed to be done on a master node
   ##### Initiate Kube cluster

   sudo kubeadm init --ignore-preflight-errors=all

   → #### Setup the environment Variables
   mkdir -p $HOME/.kube
   sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
   sudo chown $(id -u):$(id -g) $HOME/.kube/config

→ set up the Calico network
sudo kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/v3.24.1/manifests/calico.yaml

→ #### check if Kube Master is ready or not
kubectl get nodes

2. PODs
PODs are the layer on the top of the container used to run containers on the node servers. Pods defines the abstraction over the containers. Running the pod means running the container

→ Running nginx pod on the master node
We can see that the status of pod1 created is pending

```
root@ip-172-31-76-130:/# kubectl run pod1 --image nginx
pod/pod1 created
root@ip-172-31-76-130:/# kubectl get pods
NAME   READY   STATUS    RESTARTS   AGE
pod1   0/1     Pending   0          14s
```

This is because there is no worker node given to the POD on which it can be running the containers. So we will have to perform some function, which let the master node acts like a worker node.

→ First, we will note the ip of the node

```
pod1   0/1     Pending   0          14s
root@ip-172-31-76-130:/# kubectl get nodes
NAME               STATUS   ROLES                  AGE   VERSION
ip-172-31-76-130   Ready    control-plane,master   67m   v1.23.2
root@ip-172-31-76-130:/# ^C
```

Then we will get the information of node via #describe command by giving the ip address

```
root@ip-172-31-76-130:/# ^C
root@ip-172-31-76-130:/# kubectl describe nodes ip-172-31-76-130
```

Here we can see the taint column, which sets the master node and needs to be changed

```
root@ip-172-31-76-130:/# ^C
root@ip-172-31-76-130:/# kubectl describe nodes ip-172-31-76-130
Name:               ip-172-31-76-130
Roles:              control-plane,master
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ip-172-31-76-130
                    kubernetes.io/os=linux
                    node-role.kubernetes.io/control-plane=
                    node-role.kubernetes.io/master=
                    node.kubernetes.io/exclude-from-external-load-balancers=
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 172.31.76.130/20
                    projectcalico.org/IPv4IPIPTunnelAddr: 192.168.132.192
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Thu, 26 Oct 2023 12:36:33 +0000
Taints:             node-role.kubernetes.io/master:NoSchedule
Unschedulable:      false
Lease:
  HolderIdentity:  ip-172-31-76-130
```

To remove the taint, give this command

kubectl taint node <node_name> <taintname>-

```
root@ip-172-31-76-130:/# kubectl taint node ip-172-31-76-130 node-role.kubernetes.io/master:NoSchedule
node/ip-172-31-76-130 tainted
```

Taint will be removed

→ Lets create a new pod and try to run on the master node

```
error: node ip-172-31-76-130 already has node-role.kuberne
root@ip-172-31-76-130:/# kubectl run pod2 --image nginx
pod/pod2 created
root@ip-172-31-76-130:/# kubectl get pods
NAME   READY   STATUS    RESTARTS   AGE
pod1   1/1     Running   0          26m
pod2   0/1     Pending   0          9s
root@ip-172-31-76-130:/#
```

It will be successfully running

To know the information of the nodes for the each pod

```
root@ip-172-31-76-130:/# kubectl get pods -o wide
NAME   READY   STATUS    RESTARTS   AGE   IP                NODE
pod1   1/1     Running   0          27m   192.168.132.196   ip-172-31-76-130
pod2   0/1     Pending   0          69s   <none>            <none>
root@ip-172-31-76-130:/#
```

Delete the pod

```
pod2   0/1     Pending   0          69s   <none>
root@ip-172-31-76-130:/# kubectl delete pod pod1
pod "pod1" deleted
root@ip-172-31-76-130:/# kubectl get pods
NAME   READY   STATUS    RESTARTS   AGE
pod2   0/1     Pending   0          3m18s
root@ip-172-31-76-130:/#
```

$ kubectl delete pods –all → deletes all pods

3. PODs creation file

Lets create a pod file with just 1 container. It is written in YAML

In kubernetes the code written in YAML
The file in which we write the code for kubernetes --> Object defintion file OR Manifest file

in this file we have 4 main sections:

apiVersion:

It is the schema that k8s uses to create the deired object
The apiVersion value is given by kubernetes

kind:

indicates the kind of object that we have to create
The name of the object is also given by kuberentes

metadata:

name: here we give name of the object (given by user)
labels:
labels are tags whicha re used to query pods of same type/group
userKey: uservalue

spec:
you specify what you want on the object

```
root@ip-172-31-76-130:/# mkdir mykubfiles
root@ip-172-31-76-130:/# cd mykubfiles/
root@ip-172-31-76-130:/mykubfiles# vim myPodFile.yml
```

```yaml
apiVersion: v1

kind: Pod

metadata:

 name: mypod

 labels:

  author: akshit

  app: java

  type: webserver

spec:

 containers:

 - name: c1

   image: nginx
```

```
error: the path "pod-definition.yml" does not exist
root@ip-172-31-76-130:/mykubfiles#  kubectl create -f myPodFile.yml
pod/mypod created
root@ip-172-31-76-130:/mykubfiles# kubectl describe pod myPodFile.yml
Error from server (NotFound): pods "myPodFile.yml" not found
root@ip-172-31-76-130:/mykubfiles# kubectl describe pod mypod
Name:         mypod
Namespace:    default
Priority:     0
Node:         <none>
Labels:       app=java
              author=sonal
              type=webserver
Annotations:  <none>
Status:       Pending
IP:
IPs:          <none>
Containers:
  c1:
    Image:        nginx
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-swfnp (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-swfnp:
    Type:                     Projected (a volume that contains injected data from mu
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    ConfigMapOptional:        <nil>
    DownwardAPI:              true
QoS Class:                    BestEffort
Node-Selectors:               <none>
```

kubectl create –f means creating an object which is in this file, i.e create the container in this case- #describe command describes all the stages happening in the pod by the kubectl.

Pod has been successfully created and running

```
root@ip-172-31-76-130:/mykubfiles# kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
mypod   1/1     Running   0          8m26s
pod2    1/1     Running   0          36m
root@ip-172-31-76-130:/mykubfiles#
```

*if not running, check the taint mode again

→ Now we will build pod file with multiple containers.

```
apiVersion: v1
kind: Pod
metadata:
 name: multipod
 labels:
  role: dev
spec:
 containers:
 - name: c1
   image: nginx
 - name: c2
   image: tomcat
 - name: c3
   image: ubuntu
   command: ["bash", "-c", "sleep 6000"]
```

command : sleep 6000 – lets the Ubuntu container keep running, by default it stops, once after it runs.

```
root@ip-172-31-76-130:~# kubectl get pods
NAME       READY   STATUS    RESTARTS        AGE
multipod   3/3     Running   0               3m20s
mypod      1/1     Running   1 (3h27m ago)   5h8m
pod2       1/1     Running   1 (3h27m ago)   5h36m
```

All 3 pod containers are running on the node

→ Seeing logs of a container from pod
#kubectl logs “pod name” –c “container name”

```
root@ip-172-31-76-130:~# kubectl logs multipod
error: a container name must be specified for pod multipod, choose one of: [c1 c2 c3]
root@ip-172-31-76-130:~# kubectl logs multipod -c c2
26-Oct-2023 19:36:44.920 INFO [main] org.apache.catalina.startup.VersionLoggerListener.lo
version name:   Apache Tomcat/10.1.15
26-Oct-2023 19:36:44.939 INFO [main] org.apache.catalina.startup.VersionLoggerListener.lo
built:          Oct 11 2023 23:38:40 UTC
26-Oct-2023 19:36:44.939 INFO [main] org.apache.catalina.startup.VersionLoggerListener.lo
version number: 10.1.15.0
26-Oct-2023 19:36:44.939 INFO [main] org.apache.catalina.startup.VersionLoggerListener.lo
:               Linux
```

4. Replica set in pods:
   Create multiple pods with same container.

```
vim replicaset.yml

apiVersion: apps/v1
kind: ReplicaSet
metadata:
 name: myrs
spec:
 replicas: 3
 selector: # This is used to query the cluster before creating the replicas
  matchLabels:
   type: webserver
 template: # this is pod template using which replicas will be created
  metadata:
   labels: # purpose of labels is to group replicas for quering in cluster
    type: webserver
    author: sonal
  spec: # this is specification of pod/replica
   containers:
    - name: c1
      image: nginx
```

Run the pod - # kubectl create -f <filename>

```
root@ip-172-31-76-130:~# kubectl get all
NAME             READY   STATUS    RESTARTS        AGE
pod/multipod     3/3     Running   0               55m
pod/mypod        1/1     Running   1 (4h20m ago)   6h1m
pod/myrs-gjlvm   1/1     Running   0               18m
pod/myrs-zs6xb   1/1     Running   0               18m
pod/pod2         1/1     Running   1 (4h20m ago)   6h29m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   7h55m

NAME                   DESIRED   CURRENT   READY   AGE
replicaset.apps/myrs   3         3         3       18m
root@ip-172-31-76-130:~#
```

All 3 replica set pods are running with the name multipod

→ We can scale up the pods

```
root@ip-172-31-76-130:~#  kubectl scale replicaset myrs --replicas=4
replicaset.apps/myrs scaled
root@ip-172-31-76-130:~# kubectl get all
NAME             READY   STATUS    RESTARTS        AGE
pod/multipod     3/3     Running   0               57m
pod/mypod        1/1     Running   1 (4h21m ago)   6h2m
pod/myrs-4b5pf   1/1     Running   0               16s
pod/myrs-gjlvm   1/1     Running   0               19m
pod/myrs-zs6xb   1/1     Running   0               19m
pod/pod2         1/1     Running   1 (4h21m ago)   6h30m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   7h57m

NAME                   DESIRED   CURRENT   READY   AGE
replicaset.apps/myrs   4         4         4       19m
```

→ Similarly we can scale down the pods

```
root@ip-172-31-76-130:~# kubectl scale replicaset myrs --replicas=2
replicaset.apps/myrs scaled
root@ip-172-31-76-130:~# kubectl get all
NAME             READY   STATUS    RESTARTS        AGE
pod/multipod     3/3     Running   0               58m
pod/mypod        1/1     Running   1 (4h23m ago)   6h4m
pod/myrs-zs6xb   1/1     Running   0               21m
pod/pod2         1/1     Running   1 (4h23m ago)   6h32m

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   7h58m

NAME                   DESIRED   CURRENT   READY   AGE
replicaset.apps/myrs   2         2         2       21m
```

s