KUBECTL:

*# POD Manifest*

*apiVersion: v1*

*kind: Pod*

*metadata:*

*name: <PodName>*

*labels:*

*<Key>: <value>*

*namespace: <nameSpaceName>*

*spec:*

*containers:*

*- name: <NameOfTheCotnainer>*

*image: <imagaName>*

*ports:*

*- containerPort: <portOfContainer>*

***Example:***

*apiVersion: v1*

*kind: Pod*

*metadata:*

*name: mavenwebapppod*

*labels:*

*app: mavenwebapp*

*namespace: test-ns*

*spec:*

*containers:*

*- name: mavenwebappcontainer*

*image: dockerhandson/maven-web-application:1*

*ports:*

*- containerPort: 8080*

* By default Pods will not scheduled in master-node and master-node will place that pods to worker node.
* Pod-IP will keep on change
* If Pod-IP change we need to re-deploy application one more time
* kubectl apply -f <fileName.yml>
* kubectl get all
* kubectl get pods
* kubectl get po
* kubectl get pods --show-labels
* kubectl get pods -o wide
* kubectl get pods -o wide --show-labels
* kubectl describe pod <podName>
* kubectl describe pod <podName> -n <namespace>
* kubectl describe po -l name=myLabel **# Describe pods by label name=myLabel**
* kubeclt describe node <nodeip> **#node is nothing but server**

kubectl describe node ip-172-31-3-34

**# ip-172-31-3-34 :** node-IP which the pod is scheduled.

* curl -vL <podIP:PORT>/context-root

curl -vL 10.44.0.3:8080/maven-web-application/

* kubectl get pods -n kube-system
* kubectl get pods -o json
* kubectl get pods -o yaml
* kubectl run <image-name> --image=<Name of the image> --dry-run=client –o yaml
* kubectl api-resources
* kubectl get pods **--show-labels -n test-ns**
* kubectl logs <pod-name> **# if only one pod**
* kubectl logs <pod-name> -c <container-name> **# if multiple container in one pod**
* kubectl exec <pod-name>
* kubectl exec <pod-name> -c <container-name> <command>

**kubectl exec javawebapp-pod -c nginxcontainer ls**

* kubectl exec -it <pod-name> /bin/sh

*#Multi-Container Pods Manifest file*

*apiVersion: v1*

*kind: Pod*

*metadata:*

*name: <PODName>*

*namespace: <nameSpaceName>*

*labels:*

*<labelKey>: <labelValue>*

*spec:*

*containers:*

*- name: <nameOftheCotnainer>*

*image: <imageName>*

*ports:*

*- containerPort: <portNumberOfContainer>*

*- name: <nameOftheCotnainer>*

*image: <imageName>*

*ports:*

*- containerPort: <portNumberOfContainer>*

***Example:***

*apiVersion: v1*

*kind : Pod*

*metadata:*

*name: mvnpod*

*namespace: test-ns*

*labels:*

*author: Anil*

*spec:*

*containers:*

*- name: mvnwebcontainer*

*image: dockerhandson/maven-web-application:1*

*ports:*

*- containerPort: 8080*

*- name: httpdimg*

*image: venki22/imghttpd*

*ports:*

*- containerPort: 9090*

**K8's Service**

* In Kubernetes Service makes our pods accessible/discoverable with in the cluster or exposing them to internet.
* service will identify pods using it's labels And Selector. Whenever we create a service a Cluster-IP (virtual IP) Address will be allocated for that serivce and DNS will be created for that IP. So internally we can access using service name (DNS).
* To access the pods(within the cluster or outside of the cluster) we need to create the service
* Service identify the pods using labels and selectors
* **Service Type:**
* 1.Cluster-IP **(default type within the cluster)**
* 2.NodePort
* 3.LoadBalancer(External Load Balance)
* 4.Headless

***Service***

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: <serviceName>*

*namespace: <nameSpace>* ***#logically grouping of kubernates resources***

*spec:*

*type: <ClusterIP/NodePort>*

*selector:*

*<key>: <value>*

*ports:*

*- port: <servciePort> # default It to 80*

*targetPort: <containerPort>*

***With in Cluster ClusterIP***

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: mavenwebappservice*

*namespace: test-ns*

*spec:*

*type: ClusterIP*

*selector:*

*app: mavenwebapp*

*ports:*

*- port: 80*

*targetPort: 8080*

* kubectl apply -f <file.yml>
* kubectl get svc
* kubectl get all
* kubectl get ns
* kubectl get namespace
* kubectl describe service <serviceName>
* kubectl describe service <serviceName> -n <namespace>
* kubectl describe svc <serviceName> -n <namespace>
* kubectl describe service <serviceName> o wide -n <namespace>

kubectl describe service mavenwebappservice o wide -n test-ns

* curl -vL <service-ip>/maven-web-application

curl -vL 10.100.95.105/maven-web-application

