**K8**

**What is pod?**

**Kubernetes could not deploy the container directly to the nodes. In Kubernetes the containers are encapsulated inside the pod and the pods are deployed to the nodes. So, Pod is the basic atomic unit of scheduling for Kubernetes. So, by adding or deleting the pods we can scale up and scale down the app. If the load increase further, we can add new node and start deploying pods onto the new node.**

**How to deploy pod?**

**To deploy pod, we need to write pod manifest file in json or yaml format. The pod manifest file contains the container images which we want to deploy. The manifest file submits to the API server on the master node. After that API server and the scheduler components on master nodes decides and deploys these pods onto**

**appropriate worker nodes.**

**What is multi-container pod?**

**Mostly there will be one container inside one pod. But sometimes we can see there will be multiple containers inside a pod.**

**Sometimes you will come across a scenario where you have a helper container that might be doing some kind of supporting task. For our main web app, such as processing a user entered data or a processing a file uploaded by user etc. In that case you need both helper and app container inside one pod. So, in this case if the app container dies then the helper container also dies as they are part of same container.**

**Multi container inside the pod communicate with each other directly by referencing using a local host because they share the same network namespace. Also easily share the storage space as well.**

**So all these containers within the same pod share, same common IP address, volumes, network namespace , and IPC.**

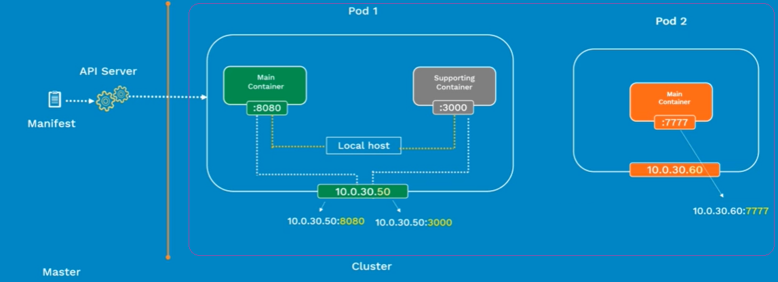
**How pod communicate with another pod?**

**When we deploy the pod onto the worker node inside the Kubernetes cluster, it will get its own IP address. That is known as pod IP address. And by using the pod ip one pod communicate with another pod.**

**How container inside pod communicate with the outside world?**

**All these containers inside a pod operate within network namespace. which means all these containers inside the pod will have the same pod IP address with container port. Using pod ip and container port the container inside the pod communicates outside world.**

**If there is different container inside single pod, then there will be different container ports for each container. In that case the containers inside the single pod can communicate directly with each other's port on localhost ( that is localhost interface ) within the pod.**

****

**How one pod communicates with another pod / Interpol communication?**

**In Kubernetes one pod communicate with another pod using pod network. When we install pod network plugin ( flannel ) and configure Cidr IP range then we will get unique ip address for pod. That's routable within that cluster. This means every pod can directly talk with every other pod and there is no need to mess with any port mapping or anything.**

**Explain the life cycle of pod?**

**Write pod manifest file 🡪 Submit manifest file to API server in k8 master 🡪 pod get scheduled onto a worker node in k8 cluster 🡪 once scheduled pod go to pending state 🡪 In pending state container image mention in pod manifest file get downloaded and start running the container 🡪 once container is up and running pod go to running stage else go to failed stage 🡪 Once the pod achieve its goal get shut down and go to succeeded stage.**

**Note -: if pod dies you can replace it with a new pod, but you cannot bring that pod back.**

**What is Api version?**

**API version defines the version number of Kubernetes object belongs to which stable release of Kubernetes API.**

****

**Write a pod manifest file / pod config file ?**

**vi nginx-pod.yaml**

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: nginx-pod**

**labels:**

**app: nginx**

**tier: dev**

**spec:**

**containers:**

* **name: nginx-container**

**image: nginx**

**How to create\deploy a pod ?**

**Kubectl create -f pod\_Yaml\_file\_name**

**Ex🡪 Kubectl create -f nginx-prod.yaml**

**How to list all the pod in k8 cluster? How to list all pod status in k8 cluster?**

**Kubectl get pod**

**How to list specific pod status in k8 cluster?**

**Kubectl get pod pod\_name**

**Ex 🡪 Kubectl get pod nginx-pod**

**How to get the pod ip address and node in k8 cluster?**

**Kubectl get pod -o wide**

**How to get the pod config detail in yaml format ?**

**Kubectl get pod pod\_name -o yaml**

**Ex🡪 Kubectl get pod nginx-pod -o yaml**

**What are the details you will get when execute Kubectl describe?**

**Syntax 🡪 Kubectl describe pod pod-name**

**The Kubectl describe give list of all events from the time pod is assigned to the node till the current status of the pod. As well as it gives label of the pod , node and ip address of the pod on which it is running.**

**How to check the pod is up and running?**

**Go to the master node and ping the pod ip ( which you can get from Kubectl get command ).**

**How to get inside the pod?**

**Kubectl exec -it pod-name -- /bin/sh**

**Ex 🡪 Kubectl exec -it nginx-pod -- /bin/sh**

**How to delete the pod?**

**Kubectl delete pod pod-name**

**Ex 🡪 Kubectl delete pod nginx-pod**

**How to expose pod using NodePort service?**

**Kubectl expose pod pod\_name --type=NodePort --port=Port\_no**

**Ex 🡪 Kubectl expose pod nginx-pod --type=NodePort --port=80**

**How to display the service an node port ?**

**Kubectl describe svc pod\_name**

**Ex 🡪 Kubectl describe svc nginx-pod**

**How to delete service for a pod?**

**Kubectl delete svc pod-name**

**Ex 🡪 Kubectl delete svc nginx-pod**

**How to make the containerize app portable in k8?**

**Using ConfigMap, we can created containerize app portable in k8. With the help of config maps, we can containerize the image and share it with other teammate or open source community.**

**What is ConfigMap in k8?**

**ConfigMap is a Kubernetes object which allows you to separate your configuration from your pods and components. ConfigMap is an API object which contain non-sensitive data as key-value pairs and help to configure k8 pods.**

**So, if you reference a config map inside the podspec that doesn't exist, then pod won't start.**

**How to create ConfigMap? / how do we apply this configuration to the container?**

**We can apply the configuration in three ways.**

1. **Configuration files**
2. **Command line arguments**
3. **Environment variables**

**How to create a ConfigMap?**

**Create ConfigMap using file contain key value pair**

**Kubectl create configmap map\_name --from-file=directory path/file name**

**Create ConfigMap using literal**

**Kubectl create configmap map\_name --from-literal=literal\_vairble=value**

**How to create a configmap from a directory?**

**mkdir -p configure\_pod/configmaps/k8kubectl**

**vi game.properties vi ui.properties**

**enemies=aliens color.good=purple**

**lives=3 color.bad=yellow**

**Now create the configmap from directory**

**Kubectl create configmap my-game-cfg --from-file= configure\_pod/configmaps/k8kubectl/**

**How to display the configmaps in k8?**

**Kubectl get configmaps -o wide**

**How to verify data present in the configmap?**

**Kubectl get configmaps configmap\_name -o yaml**

**Ex🡪 Kubectl get configmaps my-game-cfg -o yaml**

**How to create a configmap from multiple files?**

**Kubectl create configmap map\_name --from-file=1st-file-name --from-file=2nd-file-name**

**How to create a configmap from single file?**

**cat redis-conf**

**maxmemory 2mb**

**maxmemory-policy allkeys-lru**

**Now create the configmap**

**Kubectl create configmap ex-redis-cfg --from-file=redis-conf**

**How to use configmap in pod?**

**We can use configmap in pod in two ways**

1. **Using volume**
2. **Using environment variable**

**How to use configmap in pod using volume?**

**Vi redis-cofigMap-pod.yaml**

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: redis**

**spec:**

**containers:**

**- name: redis**

**Image: Kubernetes/redis:v1**

**vloumeMounts:**

**- mountPath: /redis-master under this path configMap path present**

**name: config**

**volumes:**

**- name: config**

**configMap:**

**name: ex-redis-cfg name of the configmap created using a file**

**items:**

**- key: redis-config Key is nothing but file name created when volume created**

**path: redis.conf path is the location when data will present**

**Now create the pod**

**Kubectl create -f redis-cofigMap-pod.yaml**

**How to check the configmap file in a pod?**

**Kubectl exec pod-name cat volume-path**

**Ex 🡪 Kubectl exec redis cat /redis-master/redis.conf**

**To verify the redis config has applied successfully we can use below command as well**

**Kubectl exec redis redis-CLI**

**How to create a configmap using literal values / using key value pair variable?**

**Kubectl create configmap config-map-name --from-literal=literal-key=value**

**Ex🡪 Kubectl create configmap special-conf --from-literal=special.how=very**

**To use this configmap inside the pod we need to use the key ( in ex special.how is the key )**

**How to use configmap inside pod using environment variable?**

**Under spec section we need to use env section and under env we have to refer the configmap key which we want to use.**

**Vi busyboxTest.yaml**

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: test-pod**

**spec:**

**containers:**

**- name: test-container**

**Image: k8.gcr.io/busybox**

**Command: [ “/bin/sh”, “-c” , “env” ]**

**env:**

**- name: SPECIAL\_USE\_KEY**

**valueFrom:**

**configMapKeyRef:**

**name: special-conf**

**key: special.how**

**restartPolicy: Never**

**Now create the pod Kubectl create -f busyboxTest.yaml**

**To test / validate the environment variable we can use Kubectl log command ( and we will get the value assigned to the variable name that is SPECIAL\_USE\_KEY=very ).**

**Kubectl log test-pod | grep SPECIAL**

**How to validate the configmaps in k8?**

**Kubectl get configmaps**

**How to delete the configmap?**

**Kubectl delete configmaps configmap-name1 configmap-name2 …….**

**What is secrets in k8?**

**Secrets is a k8 object that contain sensitive data which includes passwords, token or SSH keys.**

**Using secrets, we can reduce the risk of accidental exposure of confidential information while deploying the pod.**

**What is the size of secrets file and where is stored in k8?**

**Size of secrets file is 1MB and it is stored in ETCD. Generally secret is stored in tmpfs so that will restrict access to the application in the node. Another important thing is , as it is stored in tmpfs an application running another container can add access.**

**How will you create secrets in k8?**

**There are two ways to create secretes in k8.**

1. **Using manifest file**
2. **Using Kubectl command line**

**How to create secrets using command line ?**

**Kubectl create secret secret\_type name\_of\_secrete --from-file secrete\_data**

**Or**

**Kubectl create secret secret\_type name\_of\_secrete --from-literal secrete\_data**

**Generic**

**File**

**Directory**

**Literal value**

**Docker-registry**

**tls**