K8S

1. What is Pod?

ANS: Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

A Pod (as in a pod of whales or pea pod) is a group of one or more [containers](https://kubernetes.io/docs/concepts/containers/), with shared storage and network resources, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context. A Pod models an application-specific "logical host": it contains one or more application containers which are relatively tightly coupled. In non-cloud contexts, applications executed on the same physical or virtual machine are analogous to cloud applications executed on the same logical host.

Pods creation?

============

2 ways

1. Single line command
2. Manifeast files

Top of Form



Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | Kubernetes Notes: |
|  |  |
|  | # kubectl get nodes |
|  |  |
|  | Scenario1: Creation of Pod definition file |
|  |  |
|  | # mkdir mynewfiles |
|  | #cd mynewfiles |
|  | # ls |
|  | # vim pod-definition1.yml |
|  | I |
|  | --- |
|  | apiVersion: v1 |
|  | kind: Pod |
|  | metadata: |
|  | name: nginx-pod |
|  | labels: |
|  | type: reverse-proxy |
|  | author: Shobha |
|  | spec: |
|  | containers: |
|  | - name: mynginx |
|  | image: nginx |
|  |  |
|  |  |
|  |  |
|  | # kubectl get pods |
|  |  |
|  | // pod will be ready and running |
|  |  |
|  | # kubectl get pods -o wide |
|  |  |
|  | // will give more information about pods with name of slave on which pod is available. |
|  |  |
|  | # kubectl get nodes -o wide |
|  |  |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | Create another Jenkins pod: |
|  |  |
|  | # vim pod-definition2.yml |
|  |  |
|  | i |
|  |  |
|  | --- |
|  | apiVersion: v1 |
|  | kind: Pod |
|  | metadata: |
|  | name: jenkins-pod |
|  | labels: |
|  | author: edureka |
|  | type: CI-CD |
|  | spec: |
|  | containers: |
|  | - name: myjenkins |
|  | image: jenkins |
|  |  |
|  |  |
|  | # kubectl get pods |
|  | #kubectl get pods -o wide |
|  | # kubectl get nodes - o wide |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | Create replica Set |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | kubectl delete -f pod-definition1.yml |
|  | kubectl delete -f pod-definition2.yml |
|  | kubectl get pods |
|  | # vim rc-definition.yml |
|  | i |
|  |  |
|  | --- |
|  | apiVersion: apps/v1 |
|  | kind: ReplicaSet |
|  | metadata: |
|  | name: tomcat-rc |
|  | labels: |
|  | author: edureka |
|  | type: webserver |
|  | spec: |
|  | replicas: 3 |
|  | selector: |
|  | matchLabels: |
|  | type: webserver |
|  | template: |
|  | metadata: |
|  | name: tomcat-pod |
|  | labels: |
|  | type: webserver |
|  | spec: |
|  | containers: |
|  | - name: mytomcat |
|  | image: tomcat |
|  | ports: |
|  | - containerPort: 8080 |
|  | hostPort: 9090 |
|  |  |
|  | # kubectl get pods |
|  | # kubectl get pods -o wide |
|  | # kubectl get all |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | Scaling of Pods using replica set |
|  |  |
|  | Method1: |
|  |  |
|  | Open the replica set and change replicas to 5 |
|  |  |
|  | save the file |
|  |  |
|  | execute this command: |
|  |  |
|  | # kubectl replace -f rc-definition.yml |
|  |  |
|  |  |
|  | # kubectl get pods |
|  |  |
|  | now desired replicas of tomcat will be running |
|  |  |
|  |  |
|  | Method2: direct command |
|  |  |
|  | increase or decrese the replica count by using this command |
|  |  |
|  | # kubectl scale --replicas=2 -f rc-definition.yml |
|  |  |
|  | # kubectl get pods |
|  |  |
|  | //only 2 pods will be running now |
|  |  |
|  |  |
|  | For opening desired ports: |
|  | gcloud compute firewall-rules create rule1 --allow tcp:8080 |
|  | gcloud compute firewall-rules create rule1 --allow tcp:9090 |
|  | gcloud compute firewall-rules create rule1 --allow tcp:30008 |
|  |  |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | Deployment: ROLLING UPDATE |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | Highlevel kubernetes object called Deployment: |
|  |  |
|  |  |
|  | For this we use an object called Deployment Object |
|  |  |
|  | Create a new yaml file |
|  |  |
|  | # vim deployment.yml |
|  | i |
|  |  |
|  | apiVersion: apps/v1 |
|  | kind: Deployment |
|  | metadata: |
|  | name: nginx-deployment |
|  | labels: |
|  | author: edureka |
|  | type: proxyserver |
|  | spec: |
|  | replicas: 2 |
|  | selector: |
|  | matchLabels: |
|  | type: proxyserver |
|  | template: |
|  | metadata: |
|  | name: nginx-pod |
|  | labels: |
|  | type: proxyserver |
|  | spec: |
|  | containers: |
|  | - name: mynginx |
|  | image: nginx:1.7.9 |
|  | ports: |
|  | - containerPort: 80 |
|  | hostPort: 8888 |
|  |  |
|  |  |
|  | # kubectl get all |
|  |  |
|  | //will show deployment created. |
|  |  |
|  | # kubectl get pods |
|  |  |
|  | // will show all 3 pods created |
|  |  |
|  | # kubectl get deployment |
|  |  |
|  | // will show deployment which we just created information |
|  |  |
|  |  |
|  | Lets do rolling update now: |
|  |  |
|  | # kubectl get all |
|  | # kubectl --record deployment.apps/nginx-deployment set image deployment.v1.apps/nginx-deployment mynginx=nginx:1.9.1 |
|  |  |
|  |  |
|  | # kubectl get pods |
|  |  |
|  | will give all pods, here you will see a duplicate rpelica is created for which evrsion is being updated and after which it will delete the earlier version replica. so users will not face any downtime |
|  |  |
|  | here copy the name of pod only begining : nginx-deployment |
|  |  |
|  | # kubectl describe pods nginx-deployment | less |
|  |  |
|  | Clean Up |
|  |  |
|  | # kubectl delete -f deployment.yml |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | Creation of SERVICE OBJECTS |
|  |  |
|  | NODEPORT: |
|  |  |
|  | Create a jenkins pod definition file |
|  | # vim pod-definition3.yml |
|  | --- |
|  | apiVersion: v1 |
|  | kind: Pod |
|  | metadata: |
|  | name: jenkins-pod1 |
|  | labels: |
|  | author: edureka |
|  | type: CI |
|  | spec: |
|  | containers: |
|  | - name: myjenkins1 |
|  | image: jenkins |
|  | :wq! |
|  |  |
|  | # kubcectl create -f pod-definition3.yml |
|  | # kubectl get pods |
|  | # kubectl get pods -o wide |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  | Create nodeport service object |
|  |  |
|  | # vim myservice.yml |
|  | i |
|  | --- |
|  | apiVersion: v1 |
|  | kind: Service |
|  | metadata: |
|  | name: jenkins-service |
|  | spec: |
|  | type: NodePort |
|  | ports: |
|  | - targetPort: 8080 |
|  | port: 8080 |
|  | nodePort: 30008 |
|  | selector: |
|  | author: edureka |
|  | type: CI-CD |
|  |  |
|  | # kubectl create -f myservice.yml |
|  | # kubcectl get pods |
|  | # kubectl get pods -o wide |
|  | # kubectl get nodes -o wide |
|  | # kubectl get all // will give port information |
|  |  |
|  | go to browser with an ip:30008 |
|  |  |
|  | # kubectl delete -f myservice.yml |
|  | # kubectl delete -f pod-definition3.yml |
|  |  |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | Service Object: LoadBalancer |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  | Create a new deployment definition file for nginx pod |
|  |  |
|  | # vim deployment.yml |
|  | i |
|  |  |
|  | --- |
|  | apiVersion: apps/v1 |
|  | kind: Deployment |
|  | metadata: |
|  | name: nginx-deployment |
|  | labels: |
|  | author: edureka |
|  | type: proxyserver |
|  | spec: |
|  | replicas: 2 |
|  | selector: |
|  | matchLabels: |
|  | type: proxyserver |
|  | template: |
|  | metadata: |
|  | name: nginx-pod |
|  | labels: |
|  | type: proxyserver |
|  | spec: |
|  | containers: |
|  | - name: mynginx |
|  | image: nginx:1.7.9 |
|  | ports: |
|  | - containerPort: 80 |
|  | hostPort: 8888 |
|  |  |
|  |  |
|  | # kubectl get pods |
|  | # kubectl get pods -o wide |
|  | # kubectl get nodes -o wide |
|  |  |
|  |  |
|  | We will create a new definition file of kind Service object and type as LoadBalancer, |
|  | here take the label same as that of pod given in above deployment under template: |
|  | metadata: |
|  | name: nginx-pod |
|  | labels: |
|  | type: proxyserver -- take this label information |
|  |  |
|  |  |
|  | # vim myservice2.yml |
|  | i |
|  | --- |
|  | apiVersion: v1 |
|  | kind: Service |
|  | metadata: |
|  | name: nginx-myservice |
|  | labels: |
|  | type: proxy1 |
|  | spec: |
|  | type: LoadBalancer |
|  | ports: |
|  | - port: 80 |
|  | targetPort: 80 |
|  | selector: |
|  | type: proxyserver |
|  |  |
|  |  |
|  | # kubectl create -f myservice2.yml |
|  |  |
|  | # kubectl get all |
|  |  |
|  | // you will see a service with name as service/nginx-myservice |
|  | and external ip address as pending |
|  | re execute the command to see if external ip is generated |
|  |  |
|  | OR |
|  |  |
|  | # kubectl describe service/nginx-myservice | less |
|  |  |
|  | OR |
|  |  |
|  | # kubectl describe service nginx-myservice | less |
|  |  |
|  |  |
|  | These all commands will give complete information about service and we can find the load balancer ingress ip |
|  |  |
|  |  |
|  | Clean up |
|  |  |
|  | # kubectl delete -f myservice2.yml |
|  | # kubectl get all |
|  |  |
|  | # kubectl delete -f deployment2.yml |
|  |  |
|  | # kubectl get pods |
|  |  |
|  | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |