

POC : Create the nginx pod with a yaml file

Introduction

Kubernetes

- Kubernetes also known as K8s, is an open-source container orchestration platform that for automating deployment, scaling, and management of containerized applications.
- Google created this kubernetes.
- Converts Isolated dockerized containers running on different hardwares into a cluster.

Features of kubernetes

- **Orchestration:**

Clustering any no of containers on different hardwares

- **Auto scaling :**

It is a feature in which the cluster is capable of increasing the number of nodes as the demand for service response increases and vice-versa.

- **Auto healing :**

New containers in place of crashed containers

- **Load balancing :** Distributing Incoming Traffic across nodes
- **Rollback :** Going to previous versions

Kuberntes uses various types of objects.

1. **Pod:** This is a layer of abstraction on top of a container. This is the smallest object that kubernetes can work on. In the pod, we have the container. kubectl commands will work on the pod and pod communicates there instructions to the container.
2. **Service Object:** This is used for port mapping and network load balancing.
3. **NameSpace:** This is used for creating partitions in the cluster. Pods running in a namespace cannot communicate with other pods running in other namespace.
4. **Secrets :** This is used for passing encrypted data to the pods.
5. **ReplicaSet / Replication Controller:** This is used for managing multiple replicas of a pod to perform activities like load balancing and autoscaling.=
6. **Deployment:** This is used for performing all activites that a ReplicaSet can do. It can also handle rollling updates.

Command to create a pod

```
kubectl run --image nginx webserver (webserver is pod name )
```

To see list of pod

```
kubectl get pods
```

To delete the pod

```
kubectl delete pods webserver
```

This one way of creating Pods by defining all parameteres in the command line itself.

Also Kubernetes performs container orchestration by using **Definition files or Manifest**. Definition files are yaml files.

To create a Kubernetes pod with YAML, you first create an empty file, then define the necessary key-value pairs. The important ones are the apiVersion, the kind (pod), name, and the containers within the pod.

Kubernetes Manifest

When defining a Kubernetes manifest, YAML gives you a number of advantages,

including:

- **Convenience**: You'll no longer have to add all of your parameters to the command line
- **Maintenance**: YAML files can be added to source control, such as a Github repository so you can track changes
- **Flexibility**: You'll be able to create much more complex structures using YAML than you can on the command line

4 top level elements in Definition file

1. apiVersion:

Depending on kubernetes object we want to create, there is corresponding code library we want to use.

apiVersion refers to code library

Kind	apiVersion
Pod	v1
Service	v1
Namespace	v1
Secrets	v1
ReplicaSet	apps/v1
Deployment	apps/v1

2. kind:

Refers to kubernetes object which we want to create.

Ex: Pod, Replicaset, service etc

3. metadata:

Additional information about the kubernetes object
like name, labels etc

4. spec:

Contains docker container related information like image name,

environment variables, port mapping etc.

Before Starting With this Exercise we should ready with

1. Master Node
2. Worker Node
3. Docker should be Installed on both Nodes
4. Installed Kubeadm in both master and Worker Nodes.
5. Initialization of Master Node.
6. And connect Worker Node with Master Node.

To create a nginx pod with definition file or Yaml file

1. create empty yaml file

```
touch pod-definition1.yml
```

2. Now need to define necessary Key value pairs in that Yaml file

```
vim pod-definition1.yml
```

```
---
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: nginx-pod
```

```
  labels:
```

author: arun

version : v1

spec:

containers:

- name: appserver

image: nginx

:wq

```
[root@master ~]# cat pod-definition1.yml
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    author: arun
    version : v1
spec:
  containers:
    - name: appserver
      image: nginx
[root@master ~]#
```

3. To get the list of pods

kubectl get pods (no pod is running)

```
[root@master ~]# kubectl get pods
No resources found in default namespace.
[root@master ~]#
```

4. Now run the definition file

kubectl create -f pod-definition1.yml (Pod is created)

```
[root@master ~]# kubectl create -f pod-definition1.yml
pod/nginx-pod created
```

5. Now get the list of pods

kubectl get pods

```
[root@master ~]# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-pod     1/1     Running   0           51s
[root@master ~]#
```

6. To get the list of pods on which node the pod is running

kubectl get pods -o wide

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
nginx-pod	1/1	Running	0	2m20s	192.168.171.69	worker	<none>	<none>

7. To Describe the pod

kubectl describe pods nginx-pod

All the informations about the Pod is available

```
[root@master ~]# kubectl describe pods nginx-pod
Name:          nginx-pod
Namespace:     default
Priority:       0
Service Account: default
Node:          worker/172.31.28.216
Start Time:    Wed, 12 Oct 2022 06:52:55 +0000
Labels:        author=arun
               version=v1
Annotations:    cni.projectcalico.org/containerID: 85757fc29211c23ad6f3fa27b2f9e848c1aa50d68f3a8a28b1d7184071bc8852
               cni.projectcalico.org/podIP: 192.168.171.69/32
               cni.projectcalico.org/podIPs: 192.168.171.69/32
Status:        Running
IP:            192.168.171.69
IPs:           IP: 192.168.171.69
Containers:
  appserver:
    Container ID:   containerd://e86aeb0a1882d97b1ef4f7b51ce4945d0f09d3aadb53e1a03ed806012c92e004
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:2f770d2fe27bc85f68fd7fe6a63900ef7076bc703022fe81b980377fe3d27b70
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 12 Oct 2022 06:52:56 +0000
    Ready:          True
    Restart Count:   0
    Environment:    <none>
```

8. To access the containers in the pod, enter the following command

```
-----
kubectl exec -it podname -c containername bash
```

```
kubectl exec -it nginx-pod -c appserver bash
```

```
[root@master ~]# kubectl exec -it nginx-pod -c appserver bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
root@nginx-pod:/#
```

enter exit

to get out of the container.

```
root@nginx-pod:/# exit
exit
[root@master ~]#
```

Kubadm -- This is an application that is responsible for creating master node. It also stores information about the slaves.

kubectl -- This is an application that triggers the kubernetes commands