Kubernates Up and Running

Alexander Ibrahim Following
May 3 · 3 min read

When the first time I learned about kubernates, I got overwhelmed with terminologies. After trying sometimes, kubernates have three basic building blocks: `pod` , `deployment` and `service` . The other feature we can learn it along the way. I believe that if you have master this, You will have more confident to learn other feature.

Pod

> Pods *are the smallest deployable units of computing that can be created and managed in Kubernetes.*

In a pod we may have one or more containers, but it is best practice if we have one container per pod. Kubernetes supports several container runtimes: Docker, containerd, CRI-O, and any implementation of the Kubernetes CRI (Container Runtime Interface). But for now we use docker.

Deployment

> *A* Deployment *provides declarative updates for Pods and ReplicaSets.*

If we want to update container image and scale the pods, Deployment will help use with kind of case.

Service

> *An abstract way to expose an application running on a set of Pods as a network service.*

In other word, if you want to expose our set of pods to external or internal cluster so they can interact with other pods we need service. Service will load-balancing our set of pods. although pods have IP but they are mortal.If a pod get updated it will change IP address. It is hard to interact with dynamic IP address.

Let us take a look practical example

Prerequisite:

- Minikube

`nginx-deployment`

```
# nginx-deployment.yml

apiVersion: apps/v1
kind: Deploymentmetadata:
  name: nginx
  labels:
    app: frontend
spec:
  replicas: 1
  selector:
```

```
      matchLabels:
        app: frontend
    template:
      metadata:
        name: nginx
        labels:
          app: frontend
      spec:
        containers:
          - name: nginx
            image: nginx:mainline-alpine
            ports:
              - containerPort: 80
```

nginx-service.yml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: frontend
spec:
  selector:
      app: frontend
  type: NodePort
  ports:
    - nodePort: 32000
      port: 80
      targetPort: 80
```

after that we deploy our our pod

```
projects/kubernates kubectl create -f nginx-deploy.yml
deployment.apps/nginx created
projects/kubernates kubectl get pods -l app=frontend -o wide
NAME                     READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
nginx-7cb8874f9f-vcx6f   1/1     Running   0          13s   172.17.0.6   minikube   <none>           <none>
```

pod information

after that we can create our service

```
projects/kubernates kubectl create -f nginx-service.yml
service/nginx created
projects/kubernates kubectl get service -l app=frontend
```

```
NAME    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
nginx   NodePort    10.111.39.136   <none>        80:32000/TCP   11s
```

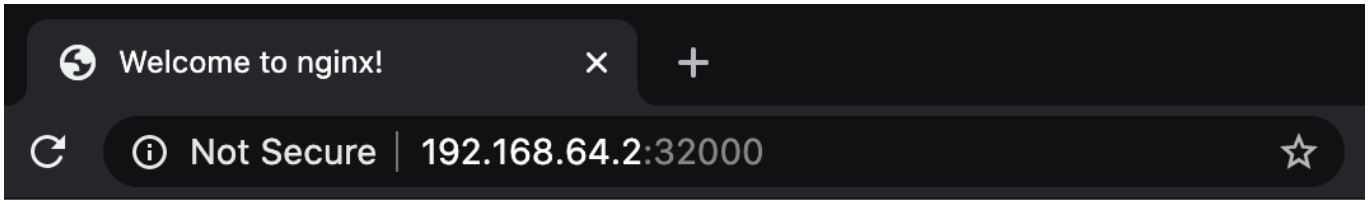
service information

Now we need node IP from minikube

```
projects/kubernates minikube ip
192.168.64.2
```

node ip information

open your browser, open http://192.168.64.2:32000



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Now our kubernates is up and running

If you wonder how this works, I try to breakdown the process with picture below.

breakdown service, deployment and pod

Kubernetes