Kubernetes Day-2

Replicasets - is used to created defined number of pods in the manifest file

Replication Controller - Is used to manage the replicas

StatefulSets - maintains the state and identity of each pod using a stable sequence. Each pod gets a unique and persistent identity and is created or terminated in a defined order.

Deployment - provides advanced features like rolling updates, rollbacks for zero downtime deployments.

Labels - are key-value pairs attached to Kubernetes objects like Pods, Nodes, or Services. They are used to organize, group, and select Kubernetes resources.

Selectors - are used to find and operate on resources based on their labels.

Manifest files

Namespace - Is like a group that provides isolation between applications or resources.

- **To run pod in namespace with yaml file**

vim namespace.yaml

```
kind: Namespace
apiVersion: v1
metadate:
  name: nginx
  namespace: nginx
spec:
  containers:
  name: nginx
  image: nginx: latest
  ports:
    - containerPort: 80
```

- **To apply manifest file**

kubectl apply -f namespace.yml

Pod - will have multiple containers

- **To create pod**

vi pod.yaml

kind: Namespace
apiVersion: v1
metadate:
  name: nginx
  namespace: nginx
spec:
  containers:
  name: nginx
  image: nginx: latest
  ports:
    - containerPort: 80

kubectl apply -f pod.yml

- **To go inside the pod**

$ kubectl exec -it pod/ nginx-pod -n nginx -- bash

Debugging and Troubleshooting: kubectl Debugging, Logs, Resource Usage Analysis.

$ kubectl describe pod/ nginx-pod -n nginx

```
ubuntu@ip-172-31-12-22:~/kubernetes-in-one-shot/nginx$ kubectl describe pod/
nginx-pod -n nginx
Name:             nginx-pod
Namespace:        nginx
Priority:         0
Service Account:  default
Node:             my-cluster-worker/172.18.0.2
Start Time:       Wed, 11 Dec 2024 10:10:34 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.1.3
IPs:
  IP:  10.244.1.3
Containers:
```

```
                          node.kubernetes.io/unreachable:NoExecute op=Exi
sts for 300s
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  3m11s  default-scheduler  Successfully assigned nginx/n
ginx-pod to my-cluster-worker
  Normal  Pulling    3m11s  kubelet            Pulling image "nginx:latest"
  Normal  Pulled     3m9s   kubelet            Successfully pulled image "ng
inx:latest" in 1.407s (1.407s including waiting). Image size: 72099501 bytes
.
  Normal  Created    3m9s   kubelet            Created container nginx
  Normal  Started    3m9s   kubelet            Started container nginx
```

Deployment - Ensures that the desired number of Pod replicas are always running.

vi deployment.yaml

```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nginx-deployment
  namespace: nginx
spec:
  replica: 2
  selector:
    matchLabels: app: nginx
template:
  metadata:
    labels:
      app: nginx
spec:
containers:
- name: nginx
  image: nginx:latest
  ports:
  - containerPort: 80
```

- **To scale deployment**

$ kubectl scale deployment/nginx-deployment -n nginx --replicas=5

```
ubuntu@ip-172-31-12-22:~/kubernetes-in-one-shot/nginx$ kubectl scale deploym
ent/nginx-deployment -n nginx --replicas=5
deployment.apps/nginx-deployment scaled
```

- **To reduce number of replicas**

kubectl scale deployment/nginx-deployment -n nginx --replicas=1

- **To get more information like in which node the pod is running.**

$kubectl get pods -n nginx -o wide

```
ubuntu@ip-172-31-12-22:~/kubernetes-in-one-shot/nginx$ kubectl get pods -n n
ginx -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
        NODE                   NOMINATED NODE   READINESS GATES
nginx-deployment-55fb85df89-44mfr   1/1     Running   0          14s   10.24
4.2.8   tws-cluster-worker2   <none>           <none>
nginx-deployment-55fb85df89-cgkg2   1/1     Running   0          14s   10.24
4.1.8   tws-cluster-worker    <none>           <none>
nginx-deployment-55fb85df89-ch8kd   1/1     Running   0          14s   10.24
4.3.7   tws-cluster-worker3   <none>           <none>
nginx-deployment-55fb85df89-pbpg5   1/1     Running   0          74s   10.24
4.2.5   tws-cluster-worker2   <none>           <none>
nginx-deployment-55fb85df89-wfztn   1/1     Running   0          14s   10.24
4.3.8   tws-cluster-worker3   <none>           <none>
```

Rolling Updates - Is a strategy by default for Deployments. It gradually replaces old pods with new ones to ensure zero downtime.

Got an Image ImagePullBackOff - While do rolling updates

```
ubuntu@ip-172-31-12-22:~/kubernetes-in-one-shot/nginx$ kubectl set image dep
loyment/nginx-deployment -n nginx nginx=1.27.3
deployment.apps/nginx-deployment image updated
ubuntu@ip-172-31-12-22:~/kubernetes-in-one-shot/nginx$ kubectl get pods -n n
ginx
NAME                                READY   STATUS         RESTARTS   AGE
nginx-deployment-55fb85df89-44mfr   1/1     Running        0          3m29s
nginx-deployment-55fb85df89-cgkg2   1/1     Running        0          3m29s
nginx-deployment-55fb85df89-ch8kd   1/1     Running        0          3m29s
nginx-deployment-55fb85df89-pbpg5   1/1     Running        0          4m29s
nginx-deployment-5c485557b7-fp6f6   0/1     ErrImagePull   0          5s
nginx-deployment-5c485557b7-hwwsm   0/1     ErrImagePull   0          5s
nginx-deployment-5c485557b7-qcrl8   0/1     ErrImagePull   0          5s
```

To resolve this error

$ kubectl set image deployment/nginx-deployment -n nginx nginx=nginx:1.27.3

```
ubuntu@ip-172-31-12-22:~/kubernetes-in-one-shot/nginx$ kubectl set image dep
loyment/nginx-deployment -n nginx nginx=nginx:1.27.3
deployment.apps/nginx-deployment image updated
```

```
nginx-deployment-55fb85df89-cgkg2   1/1     Running             0          5
m19s
nginx-deployment-55fb85df89-ch8kd   1/1     Running             0          5
m19s
nginx-deployment-55fb85df89-pbpg5   1/1     Running             0          6
m19s
nginx-deployment-774db5cc57-7r5rr   0/1     ContainerCreating   0          2
s
nginx-deployment-774db5cc57-cwpdb   0/1     ContainerCreating   0          2
s
nginx-deployment-774db5cc57-kskqd   0/1     ContainerCreating   0          2
s
```

But when I was getting this error some containers were still running that is known as rolling updates means even during the error some pods are always running while some are getting updated.


```
nginx-deployment-55fb85df89-cgkg2   1/1     Running             0          5
m19s
nginx-deployment-55fb85df89-ch8kd   1/1     Running             0          5
m19s
nginx-deployment-55fb85df89-pbpg5   1/1     Running             0          6
m19s
nginx-deployment-774db5cc57-7r5rr   0/1     ContainerCreating   0          2
s
nginx-deployment-774db5cc57-cwpdb   0/1     ContainerCreating   0          2
s
nginx-deployment-774db5cc57-kskqd   0/1     ContainerCreating   0          2
s
```


$ kubectl set image deployment/nginx-deployment -n nginx nginx=nginx:1.26.2

```
nginx-deployment-774db5cc57-kskqd   1/1     Terminating         0          6
0s
nginx-deployment-79c97d7849-ccnqg   1/1     Running             0          1
1s
nginx-deployment-79c97d7849-dxxks   0/1     ContainerCreating   0          1
s
nginx-deployment-79c97d7849-g8djr   1/1     Running             0          1
1s
nginx-deployment-79c97d7849-g9qjz   1/1     Running             0          1
1s
nginx-deployment-79c97d7849-gljzl   0/1     ContainerCreating   0          1
s
```

When I again update the image to previous version again I noticed that some pods are running while some are getting updated.

This is because Kubernetes uses a **RollingUpdate** strategy by default for Deployments. It gradually replaces old pods with new ones to ensure **zero downtime**.

- **To Delete deployment**

$ kubectl delete -f deployment.yml