K8 -03

1. Create a ClusterIP service for an Apache web server pod.

- Create a apache-pod.yml file
- vi apache-pod.yml
- apiVersion: v1
- kind: Pod
- metadata:
- name: apache-pod
- labels:
- app: apache
- spec:
- containers:
- - name: apache
- image: httpd
- ports:
- - containerPort: 80
-

```
[ec2-user@ip-172-31-4-234 ~]$ cat apache-pod.yml
apiVersion: v1
kind: Pod
metadata:
  name: apache-pod
  labels:
    app: apache
spec:
  containers:
  - name: apache
    image: httpd
    ports:
    - containerPort: 80

[ec2-user@ip-172-31-4-234 ~]$
```

- And deploy the pod
- Kubectl apply -f apache-pod.yml

```
[ec2-user@ip-172-31-4-234 ~]$ vi apache-pod.yml
[ec2-user@ip-172-31-4-234 ~]$ kubectl apply -f apache-pod.yml
pod/apache-pod created
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME           READY   STATUS             RESTARTS        AGE
apache-pod     1/1     Running            0               5s
demo-pod       1/1     Running            1 (88m ago)     92m
firstpod       1/1     Running            0               17m
initpod        2/2     Running            2 (47m ago)     108m
nginx          1/1     Running            0               98m
resource-pod   1/2     CrashLoopBackOff   5 (2m24s ago)   5m50s
[ec2-user@ip-172-31-4-234 ~]$ vi apache-service.yml
```

- Now write a service .yml file for cluister ip
- Vi clusaterip.yml
- apiVersion: v1
- kind: Service
- metadata:
- name: apache-service
- spec:
- type: ClusterIP
- selector:
- app: apache
- ports:
- - protocol: TCP
- port: 80
- targetPort: 80
- Niow deploy it and
- Check for ip by
- Kubectl get svc

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl apply -f apache-service.yml
service/apache-service created
[ec2-user@ip-172-31-4-234 ~]$ kubectl get svc
NAME              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
apache-service    ClusterIP   10.100.68.113   <none>        80/TCP     7s
firstpod-service  ClusterIP   10.108.23.171   <none>        80/TCP     11m
kubernetes        ClusterIP   10.96.0.1       <none>        443/TCP    147m
```

- kubectl exec -it testpod -- sh

```
[ec2-user@ip-172-31-4-234 ~]$
[ec2-user@ip-172-31-4-234 ~]$ kubectl exec -it testpod -- sh
/ # wget -qO- apache-service
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>It works! Apache httpd</title>
</head>
<body>
<p>It works!</p>
</body>
</html>
/ #
/ #
```

1. 2 . Expose an Nginx pod externally using a NodePort service.
    - Write 2 yml file
    - Nginx1.yml
    - apiVersion: v1
    - kind: Pod
    - metadata:
    - name: nginx-pod
    - labels:
    - app: nginx
    - spec:
    - containers:
    - - name: nginx

- image: nginx
- ports:
- - containerPort: 80
- Nginx-noideport.yyml
- Vi nginx-nodeport.yml
- apiVersion: v1
- kind: Service
- metadata:
- name: nginx-nodeport
- spec:
- type: NodePort
- selector:
- app: nginx
- ports:
- - port: 80 # Service port
- targetPort: 80 # Container port
- nodePort: 30080 # External port (must be 30000–32767)
- Now run the comman
- kubectl apply -f nginx.yml
- and write nginx-nodeport.yml
- pods created
- now
- get the nodeport ofnginx by
- kubectl get svc

```
Last login: Fri Dec 26 09:43:15 2025 from 13.52.6.115
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME           READY   STATUS             RESTARTS      AGE
apache-pod     1/1     Running            0             11m
demo-pod       1/1     Running            1 (99m ago)   103m
firstpod       1/1     Running            0             28m
initpod        2/2     Running            2 (59m ago)   119m
nginx          1/1     Running            0             109m
nginx-pod      1/1     Running            0             3m2s
resource-pod   1/2     CrashLoopBackOff   8 (22s ago)   17m
testpod        1/1     Running            0             6m44s
[ec2-user@ip-172-31-4-234 ~]$ kubectl get svc
NAME               TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
apache-service     ClusterIP   10.100.68.113    <none>        80/TCP         10m
firstpod-service   ClusterIP   10.108.23.171    <none>        80/TCP         21m
kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP        158m
nginx-nodeport     NodePort    10.101.239.218   <none>        80:30080/TCP   2m20s
[ec2-user@ip-172-31-4-234 ~]$
```

- 

-

1. 3 . Deploy a ReplicationController to maintain 3 replicas of an Nginx pod.
   - Replication controller yaml file
   - nginx-rc.yml
   - apiVersion: v1
   - kind: ReplicationController
   - metadata:
   - name: nginx-rc
   - spec:
   - replicas: 3
   - selector:

- app: nginx
- template:
- metadata:
- labels:
- app: nginx
- spec:
- containers:
- - name: nginx
- image: nginx
- ports:
- - containerPort: 80
- Now create a pod

```
[ec2-user@ip-172-31-4-234 ~]$ cat nginx-rc.yml
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-rc
spec:
  replicas: 3
  selector:
    app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80

[ec2-user@ip-172-31-4-234 ~]$
```

- **i-00bbf6e27330e6872 (master)**
- PublicIPs: 3.101.25.169 PrivateIPs: 172.31.4.234
- Kubectl apply -f nginx-rc.yml
- Now
- Verify controller

- Kubectl get rc

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rc
NAME        DESIRED   CURRENT   READY   AGE
nginx-rc    3         3         3       94s
[ec2-user@ip-172-31-4-234 ~]$
```

1. 4. Scale the ReplicationController from 3 replicas to 5 replicas.

- Scasling up replicas
- kubectl scale rc nginx-rc --replicas=5

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl scale rc nginx-rc --replicas=5
replicationcontroller/nginx-rc scaled
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rc
NAME       DESIRED   CURRENT   READY   AGE
nginx-rc   5         5         5       3m
```

- kubectl scale rc nginx-rc --replicas=6

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl scale rc nginx-rc --replicas=6
replicationcontroller/nginx-rc scaled
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rc
NAME       DESIRED   CURRENT   READY   AGE
nginx-rc   6         6         6       3m10s
[ec2-user@ip-172-31-4-234 ~]$
```

- **i-00bbf6e27330e6872 (master)**

PublicIPs: 3.101.25.169 PrivateIPs: 172.31.4.234

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME             READY   STATUS             RESTARTS        AGE
apache-pod       1/1     Running            0               20m
demo-pod         1/1     Running            1 (108m ago)    112m
firstpod         1/1     Running            0               37m
initpod          2/2     Running            4 (7m58s ago)   128m
nginx            1/1     Running            0               118m
nginx-pod        1/1     Running            0               11m
nginx-rc-527b4   1/1     Running            0               88s
nginx-rc-gcfwh   1/1     Running            0               4m23s
nginx-rc-w25w5   1/1     Running            0               88s
nginx-rc-x7bzg   1/1     Running            0               75s
resource-pod     1/2     CrashLoopBackOff   9 (4m ago)      25m
testpod          1/1     Running            0               15m
[ec2-user@ip-172-31-4-234 ~]$
```

- **i-00bbf6e27330e6872 (master)**

1. 5 . Create a ReplicaSet to manage pods based on multiple labels (prod and test).

- Create a nginx-rs.yml file
- apiVersion: apps/v1
- kind: ReplicaSet
- metadata:
- name: nginx-rs
- spec:
- replicas: 3
- selector:
- matchLabels:
- env: prod
- tier: test
- template:
- metadata:
- labels:
- env: prod
- tier: test
- spec:
- containers:
- - name: nginx
- image: nginx
- ports:
- - containerPort: 80
- Now deploy it

- Kubectl apply -f nginx-rs.yml
- So , now
- Check the replication sets
- Kubectl get rc

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rs
NAME       DESIRED   CURRENT   READY   AGE
nginx-rs   3         3         3       10s
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods -
```

- Check labels
- kubectl get pods --show-labels

```
pod/test-pod created
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods --show-labels
NAME             READY   STATUS             RESTARTS        AGE     LABELS
apache-pod       1/1     Running            0               28m     app=apache
demo-pod         1/1     Running            1 (116m ago)    120m    <none>
firstpod         1/1     Running            0               45m     app=nginx
initpod          2/2     Running            4 (16m ago)     136m    <none>
nginx            1/1     Running            0               126m    app=web,env=dev
nginx-pod        1/1     Running            0               20m     app=nginx
nginx-rc-4lxrj   1/1     Running            0               5m24s   app=nginx
nginx-rc-gcfwh   1/1     Running            0               12m     app=nginx
nginx-rc-rdp8r   1/1     Running            0               5m35s   app=nginx
nginx-rc-xq99n   1/1     Running            0               5m9s    app=nginx
nginx-rs-97rqr   1/1     Running            0               3m25s   env=prod,tier=test
nginx-rs-9p9t2   1/1     Running            0               3m25s   env=prod,tier=test
nginx-rs-w8v9f   1/1     Running            0               3m25s   env=prod,tier=test
resource-pod     1/2     CrashLoopBackOff   11 (111s ago)   34m     <none>
test-pod         1/1     Running            0               97s     env=prod
testpod          1/1     Running            0               23m     run=testpod
[ec2-user@ip-172-31-4-234 ~]$
```

-]
- Test-label.yml file
- apiVersion: v1
- kind: Pod
- metadata:
- name: test-pod
- labels:
- env: prod
- spec:
- containers:

- - name: nginx
- image: nginx
- Now it will cresate new

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl apply -f test-label.yml
pod/test-pod created
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods --show-labels
NAME             READY   STATUS             RESTARTS        AGE     LABELS
apache-pod       1/1     Running            0               28m     app=apache
demo-pod         1/1     Running            1 (116m ago)    120m    <none>
firstpod         1/1     Running            0               45m     app=nginx
initpod          2/2     Running            4 (16m ago)     136m    <none>
nginx            1/1     Running            0               126m    app=web,env=dev
nginx-pod        1/1     Running            0               20m     app=nginx
nginx-rc-4lxrj   1/1     Running            0               5m24s   app=nginx
nginx-rc-gcfwh   1/1     Running            0               12m     app=nginx
nginx-rc-rdp8r   1/1     Running            0               5m35s   app=nginx
nginx-rc-xq99n   1/1     Running            0               5m9s    app=nginx
nginx-rs-97rqr   1/1     Running            0               3m25s   env=prod,tier=test
nginx-rs-9p9t2   1/1     Running            0               3m25s   env=prod,tier=test
nginx-rs-w8v9f   1/1     Running            0               3m25s   env=prod,tier=test
resource-pod     1/2     CrashLoopBackOff   11 (111s ago)   34m     <none>
test-pod         1/1     Running            0               97s     env=prod
```

- i-00bbf6e27330e6872 (master)

1. 6 . Deploy a ReplicaSet that excludes pods with the label backend.
   - Vi exclude-backend.yml
   - apiVersion: apps/v1
   - kind: ReplicaSet
   - metadata:
   - name: exclude-backend-rs
   - spec:
   - replicas: 3
   - selector:
   - matchExpressions:
   - - key: backend
   - operator: DoesNotExist

- template:
- metadata:
- labels:
- app: nginx
- tier: frontend
- spec:
- containers:
- - name: nginx
- image: nginx
- ports:
- - containerPort: 80
- Backend = extension wont appear when we list the pods

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl apply -f exclude-backend-
replicaset.apps/exclude-backend-rs created
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME              READY   STATUS        RESTARTS       AGE
demo-pod          1/1     Running       1 (120m ago)   124m
firstpod          1/1     Running       0              49m
initpod           2/2     Terminating   4 (20m ago)    140m
nginx             1/1     Running       0              130m
nginx-pod         1/1     Running       0              24m
nginx-rc-4lxrj    1/1     Running       0              9m16s
nginx-rc-gcfwh    1/1     Running       0              16m
nginx-rc-rdp8r    1/1     Running       0              9m27s
nginx-rc-xq99n    1/1     Running       0              9m1s
nginx-rs-97rqr    1/1     Running       0              7m17s
nginx-rs-9p9t2    1/1     Running       0              7m17s
nginx-rs-w8v9f    1/1     Running       0              7m17s
testpod           1/1     Running       0              27m
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rs
NAME                DESIRED   CURRENT   READY   AGE
exclude-backend-rs  3         3         3       56s
nginx-rs            3         3         3       8m8s
```

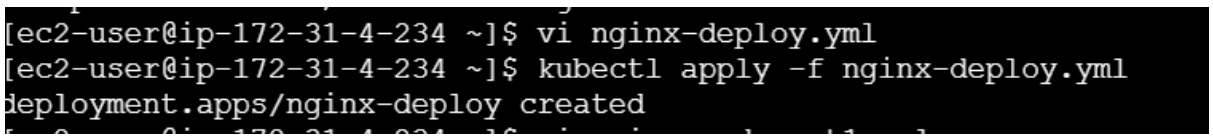
- Kubectl get rs
- Ow

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rs
NAME                 DESIRED   CURRENT   READY   AGE
exclude-backend-rs   3         3         3       56s
nginx-rs             3         3         3       8m8s
[ec2-user@ip-172-31-4-234 ~]$
```

- N
- Now createing a pod witrh baqckend=true
- apiVersion: v1
- kind: Pod
- metadata:
- name: backend-pod
- labels:
- backend: "true"
- spec:
- containers:
- - name: nginx
- image: nginx
- Now deploy the pod
- Kubectl apply -f backend-pod.yml

```
[ec2-user@ip-172-31-4-234 ~]$ vi bakcend-pod.yml
[ec2-user@ip-172-31-4-234 ~]$ kubectl apply -f bakcend-pod.yml
pod/backend-pod created
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME          READY   STATUS    RESTARTS      AGE
backend-pod   1/1     Running   0             6s
```

- 7 . Test load balancing across multiple pods using a NodePort service
- Vi nginx-deploy.yml
- apiVersion: apps/v1
- kind: Deployment

- metadata:
- name: nginx-deploy
- spec:
- replicas: 3
- selector:
- matchLabels:
- app: nginx-lb
- template:
- metadata:
- labels:
- app: nginx-lb
- spec:
- containers:
- - name: nginx
- image: nginx
- ports:
- - containerPort: 80
- Now deploy the pod
- Kubectl apply -f nginx-deploy.yml
- And
- 

- Now
- Write a yaml file for
- Nginx-nideport1.yml
- apiVersion: v1

- kind: Service
- metadata:
- name: nginx-nodeport-lb
- spec:
- type: NodePort
- selector:
- app: nginx-lb
- ports:
- - port: 80
- targetPort: 80
- nodePort: 30081

```
[ec2-user@ip-172-31-4-234 ~]$ cat nginx-nodeport1.yml
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport-lb
spec:
  type: NodePort
  selector:
    app: nginx-lb
  ports:
  - port: 80
    targetPort: 80
    nodePort: 30081

[ec2-user@ip-172-31-4-234 ~]$
```

Deploy the nignx-nodeport1.yml

Now check for load balancer

By

Kubectl get svc ngionx-nodeport1.yml

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get svc nginx-nodeport-lb
NAME                TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
nginx-nodeport-lb   NodePort   10.99.187.84   <none>        80:30081/TCP   31s
```

1. 8 . Delete a ReplicationController without affecting the running pods.

- My replication controller

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME                          READY   STATUS    RESTARTS        AGE
backend-pod                   1/1     Running   0               10m
demo-pod                      1/1     Running   1 (133m ago)    137m
firstpod                      1/1     Running   0               62m
nginx                         1/1     Running   0               144m
nginx-deploy-67677797-4w686   1/1     Running   0               6m50s
nginx-deploy-67677797-djk9z   1/1     Running   0               6m50s
nginx-deploy-67677797-pqs4m   1/1     Running   0               6m50s
nginx-pod                     1/1     Running   0               37m
nginx-rc-4lxrj                1/1     Running   0               22m
nginx-rc-gcfwh                1/1     Running   0               30m
nginx-rc-rdp8r                1/1     Running   0               22m
nginx-rc-xq99n                1/1     Running   0               22m
nginx-rs-97rqr                1/1     Running   0               20m
nginx-rs-9p9t2                1/1     Running   0               20m
nginx-rs-w8v9f                1/1     Running   0               20m
testpod                       1/1     Running   0               41m
[ec2-user@ip-172-31-4-234 ~]$
```

- **i-00bbf6e27330e6872 (master)**
- Now
- To delete the rc pods without deleting the running pods
- kubectl delete rc nginx-rc --cascade=orphan

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rc
NAME       DESIRED   CURRENT   READY   AGE
nginx-rc   6         6         6       32m
[ec2-user@ip-172-31-4-234 ~]$ kubectl delete rc nginx-rc --cascade=orphan
replicationcontroller "nginx-rc" deleted
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rc
No resources found in default namespace.
```

- Verified

```
[ec2-user@ip-172-31-4-234 ~]$ kubectl get rc
No resources found in default namespace.
[ec2-user@ip-172-31-4-234 ~]$ kubectl get pods
NAME                          READY   STATUS        RESTARTS       AGE
backend-pod                   1/1     Running       0              13m
demo-pod                      1/1     Running       1 (136m ago)   140m
firstpod                      1/1     Terminating   0              65m
nginx                         1/1     Running       0              146m
nginx-deploy-67677797-4w686   1/1     Running       0              9m22s
nginx-deploy-67677797-djk9z   1/1     Running       0              9m22s
nginx-deploy-67677797-pqs4m   1/1     Running       0              9m22s
nginx-pod                     1/1     Running       0              40m
nginx-rs-97rqr                1/1     Running       0              23m
nginx-rs-9p9t2                1/1     Running       0              23m
nginx-rs-w8v9f                1/1     Running       0              23m
testpod                       1/1     Terminating   0              43m
[ec2-user@ip-172-31-4-234 ~]$
```

- **i-00bbf6e27330e6872 (master)**