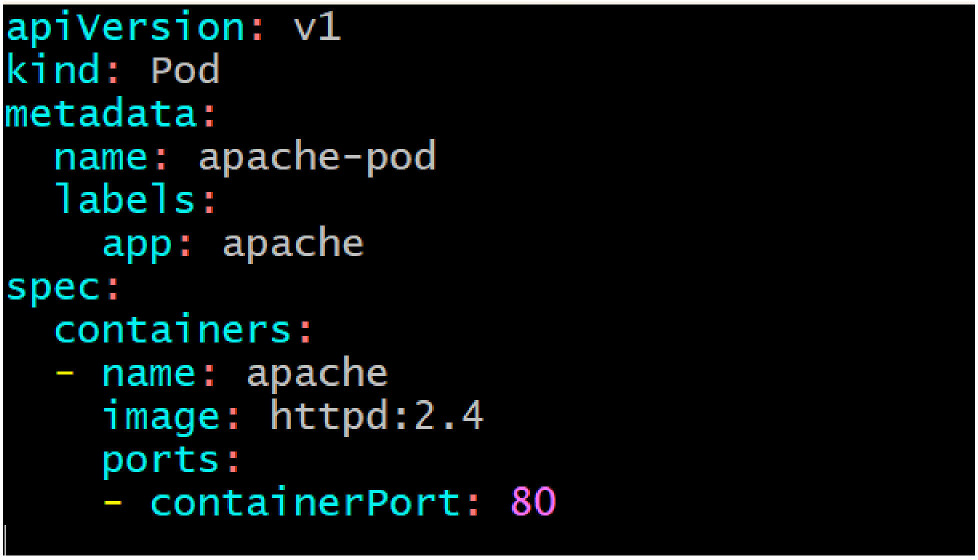
1. Create a ClusterIP service for an Apache web server pod.

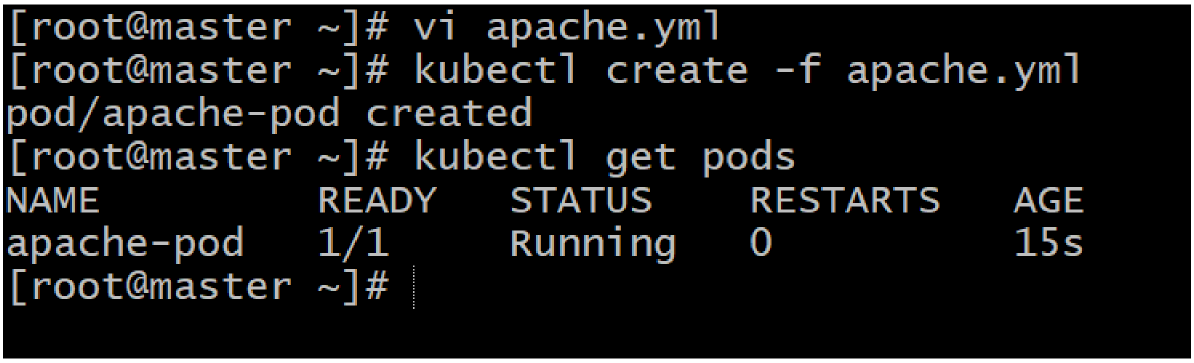
Create a httpd pod with this script.

```
apiVersion: v1
kind: Pod
metadata:
  name: apache-pod
  labels:
    app: apache
spec:
  containers:
  - name: apache
    image: httpd:2.4
    ports:
    - containerPort: 80
```



```
apiVersion: v1
kind: Pod
metadata:
  name: apache-pod
  labels:
    app: apache
spec:
  containers:
  - name: apache
    image: httpd:2.4
    ports:
    - containerPort: 80
```

- **kubectl apply -f apache.yml**

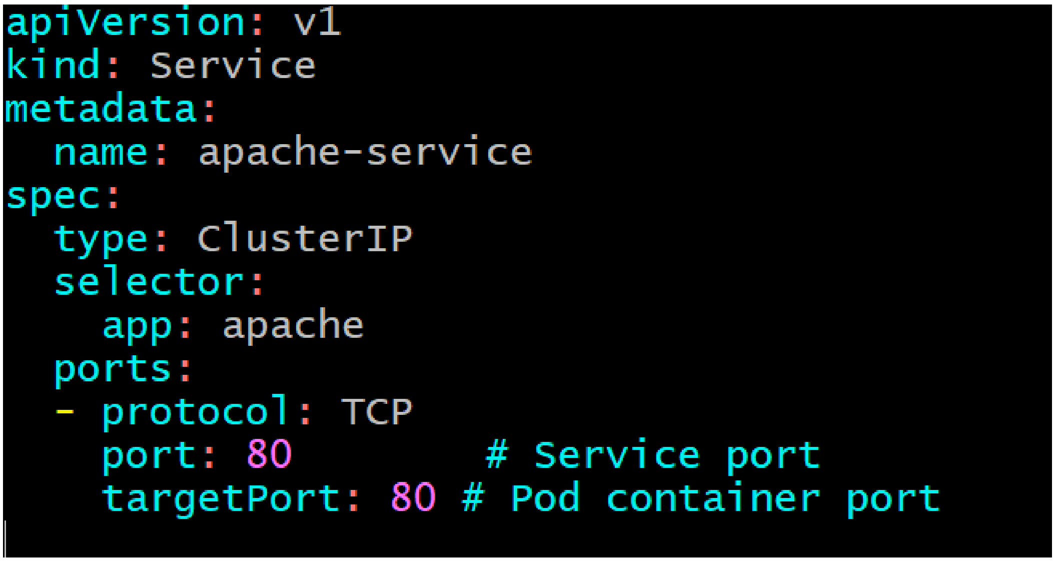


```
[root@master ~]# vi apache.yml
[root@master ~]# kubectl create -f apache.yml
pod/apache-pod created
[root@master ~]# kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
apache-pod   1/1     Running   0          15s
[root@master ~]#
```

Create a service for apache

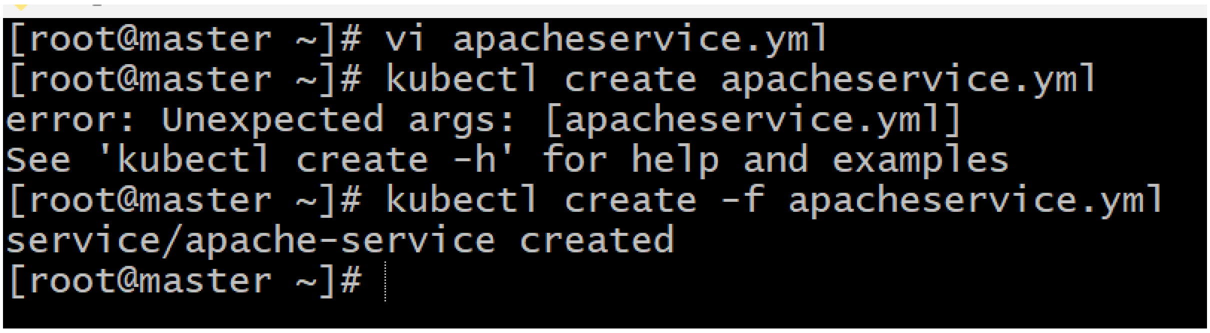
- **vi apacheservice.yml**

```
apiVersion: v1
kind: Service
metadata:
  name: apache-service
spec:
  type: ClusterIP
  selector:
    app: apache
  ports:
  - protocol: TCP
    port: 80       # Service port
    targetPort: 80 # Pod container port
```



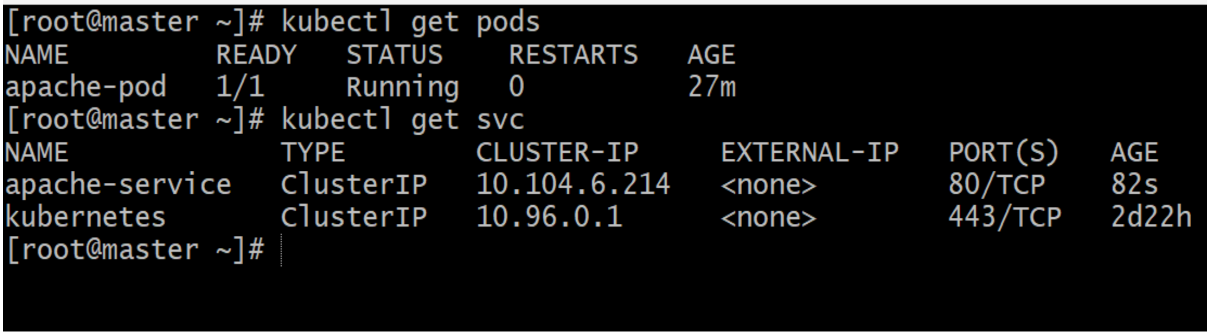
```
apiVersion: v1
kind: Service
metadata:
  name: apache-service
spec:
  type: ClusterIP
  selector:
    app: apache
  ports:
  - protocol: TCP
    port: 80          # Service port
    targetPort: 80 # Pod container port
```

- **kubectl apply -f apacheservice.yml**



```
[root@master ~]# vi apacheservice.yml
[root@master ~]# kubectl create apacheservice.yml
error: Unexpected args: [apacheservice.yml]
See 'kubectl create -h' for help and examples
[root@master ~]# kubectl create -f apacheservice.yml
service/apache-service created
[root@master ~]#
```

- **kubectl get pods**
- **kubectl get svc**

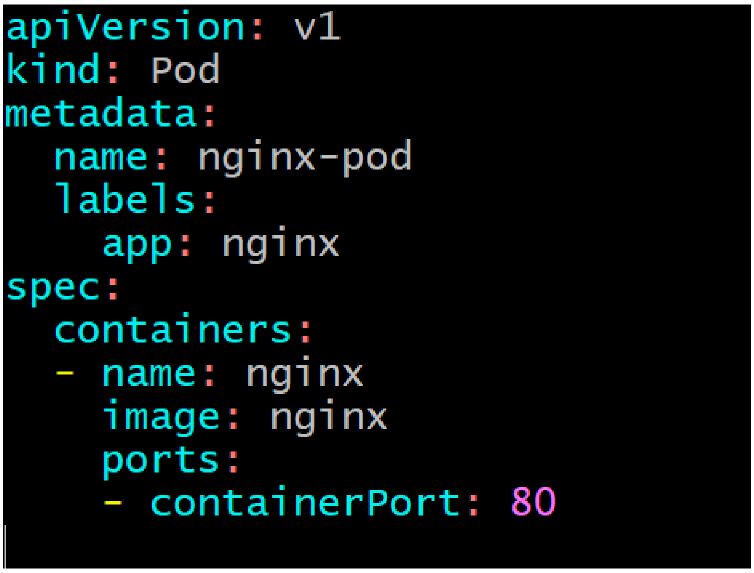


```
[root@master ~]# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
apache-pod    1/1     Running   0          27m
[root@master ~]# kubectl get svc
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
apache-service   ClusterIP   10.104.6.214    <none>        80/TCP    82s
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP   2d22h
[root@master ~]#
```

2. Expose an Nginx pod externally using a NodePort service.

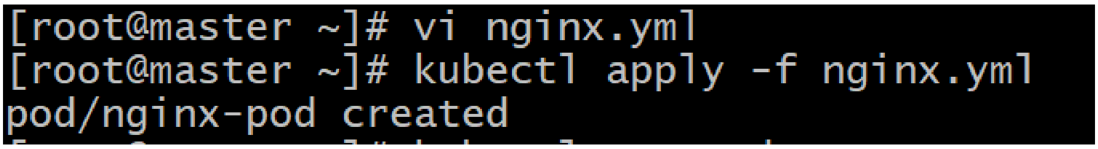
- vi nginx.yml

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
```



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx
    ports:
    - containerPort: 80
```

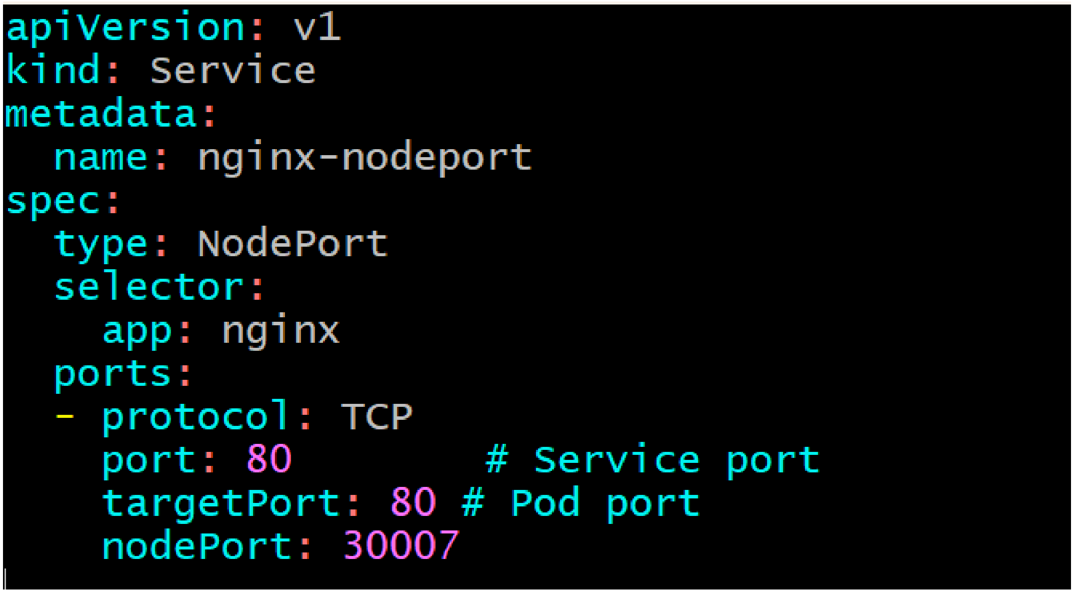
- kubectl apply -f nginx.yml



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

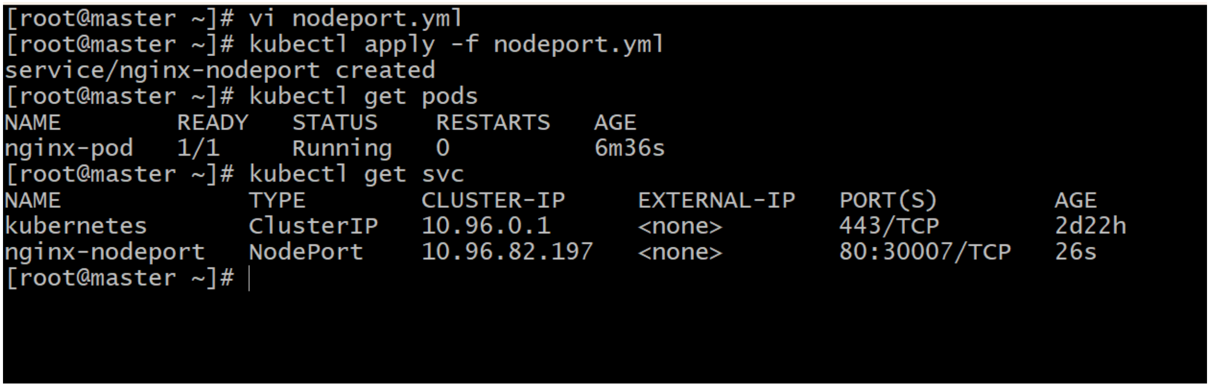
- **vi nodeport.yml**

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80       # Service port
    targetPort: 80 # Pod port
    nodePort: 30007
```



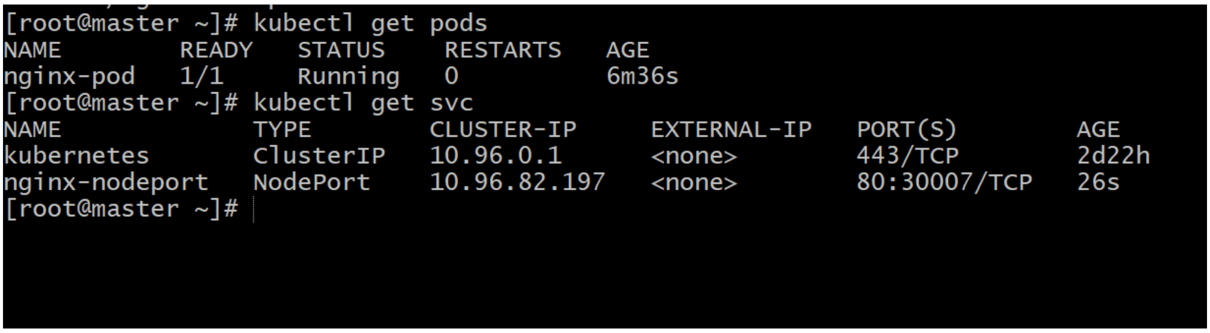
```
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
  - protocol: TCP
    port: 80        # Service port
    targetPort: 80 # Pod port
    nodePort: 30007
```

- **kubectl apply -f nodeport.yml**

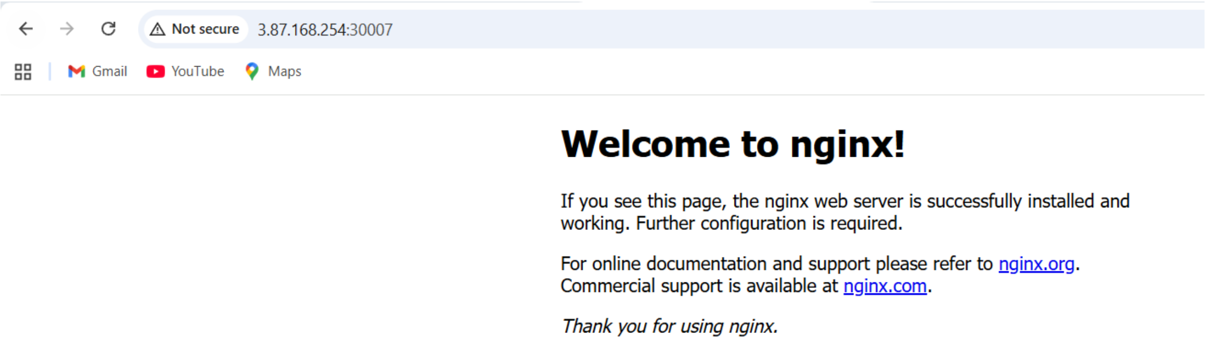


```
[root@master ~]# vi nodeport.yml
[root@master ~]# kubectl apply -f nodeport.yml
service/nginx-nodeport created
[root@master ~]# kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          6m36s
[root@master ~]# kubectl get svc
NAME             TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1      <none>        443/TCP        2d22h
nginx-nodeport   NodePort    10.96.82.197   <none>        80:30007/TCP   26s
[root@master ~]#
```

- **kubectl get pods**
- **kubectl get svc**



```
[root@master ~]# kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          6m36s
[root@master ~]# kubectl get svc
NAME             TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1      <none>        443/TCP        2d22h
nginx-nodeport   NodePort    10.96.82.197   <none>        80:30007/TCP   26s
[root@master ~]#
```



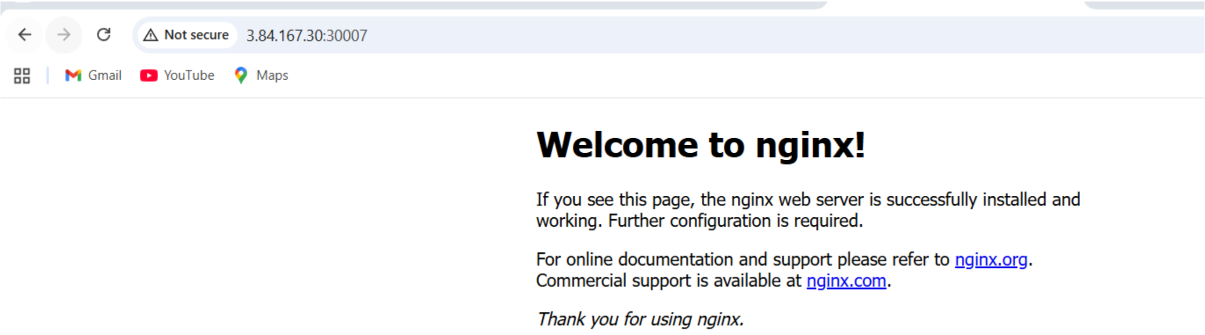
Not secure 3.87.168.254:30007

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.



Not secure 3.84.167.30:30007

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.

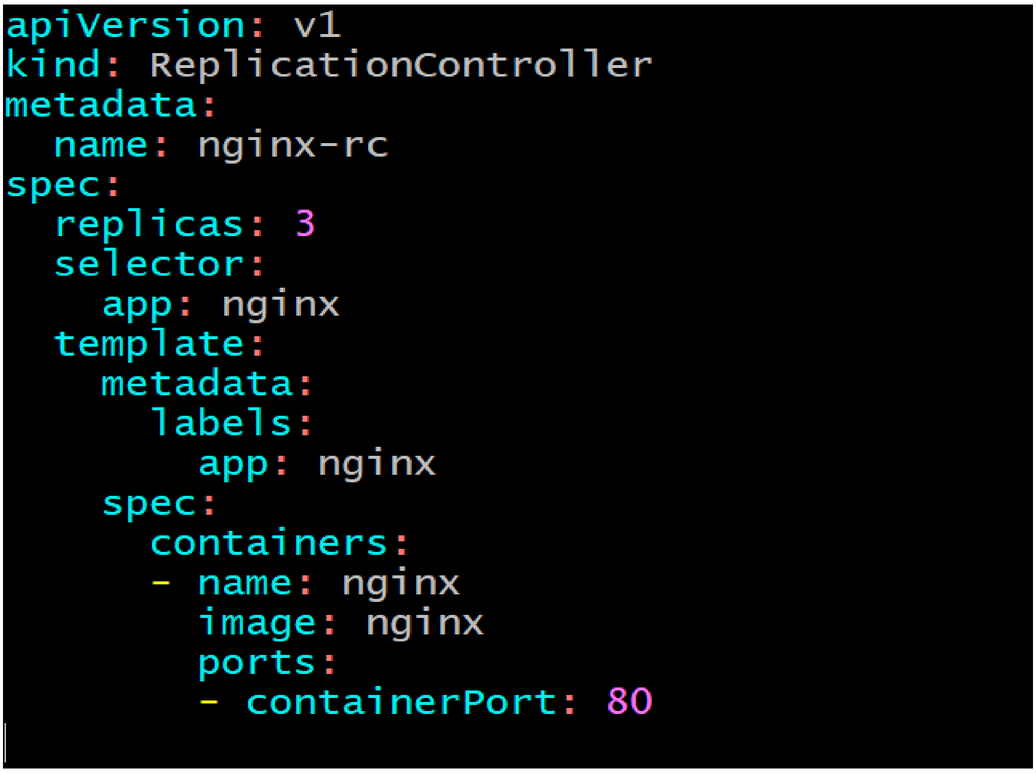
3. Deploy a ReplicationController to maintain 3 replicas of an Nginx pod.

Create a yaml file and write the

- **vi nginxrc.yml**

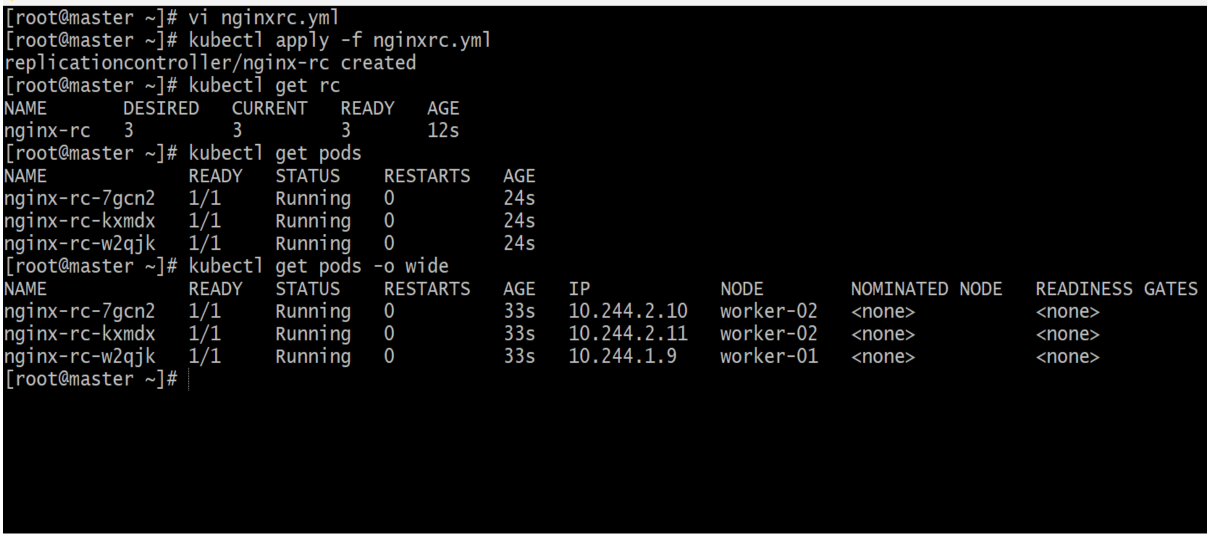
```
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-rc
spec:
  replicas: 3
```

```yaml
selector:
  app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
    - name: nginx
      image: nginx
      ports:
      - containerPort: 80
```



```yaml
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
```

- **kubectl get rc**
- **kubectl get pods**

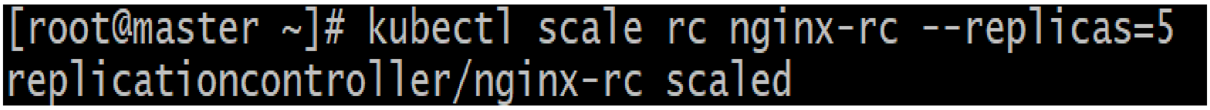


```
[root@master ~]# vi nginxrc.yml
[root@master ~]# kubectl apply -f nginxrc.yml
replicationcontroller/nginx-rc created
[root@master ~]# kubectl get rc
NAME       DESIRED   CURRENT   READY   AGE
nginx-rc   3         3         3       12s
[root@master ~]# kubectl get pods
NAME             READY   STATUS    RESTARTS   AGE
nginx-rc-7gcn2   1/1     Running   0          24s
nginx-rc-kxmdx   1/1     Running   0          24s
nginx-rc-w2qjk   1/1     Running   0          24s
[root@master ~]# kubectl get pods -o wide
NAME             READY   STATUS    RESTARTS   AGE   IP            NODE        NOMINATED NODE   READINESS GATES
nginx-rc-7gcn2   1/1     Running   0          33s   10.244.2.10   worker-02   <none>           <none>
nginx-rc-kxmdx   1/1     Running   0          33s   10.244.2.11   worker-02   <none>           <none>
nginx-rc-w2qjk   1/1     Running   0          33s   10.244.1.9    worker-01   <none>           <none>
[root@master ~]#
```

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

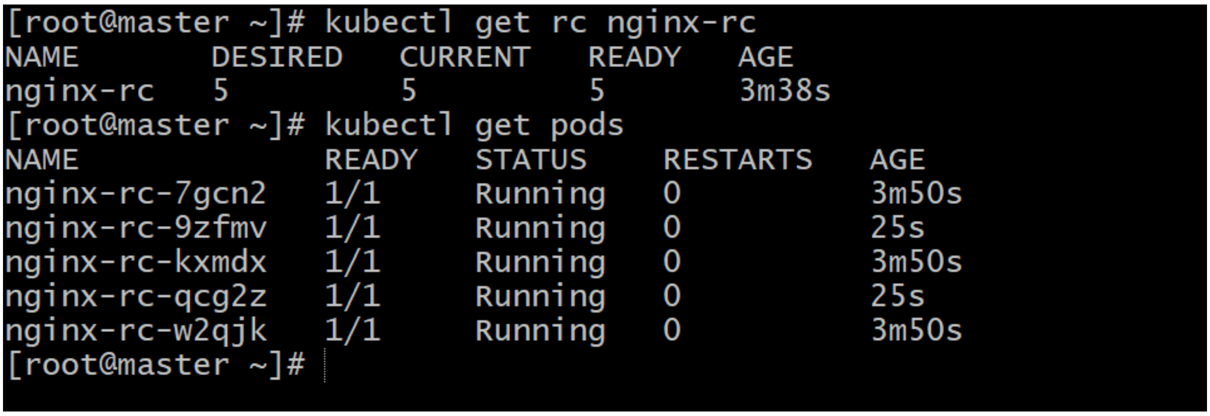
execute this command

- **kubectl scale rc nginx-rc --replicas=5**



```
[root@master ~]# kubectl scale rc nginx-rc --replicas=5
replicationcontroller/nginx-rc scaled
```

- **kubectl get rc nginx-rc**
- **kubectl get pods**



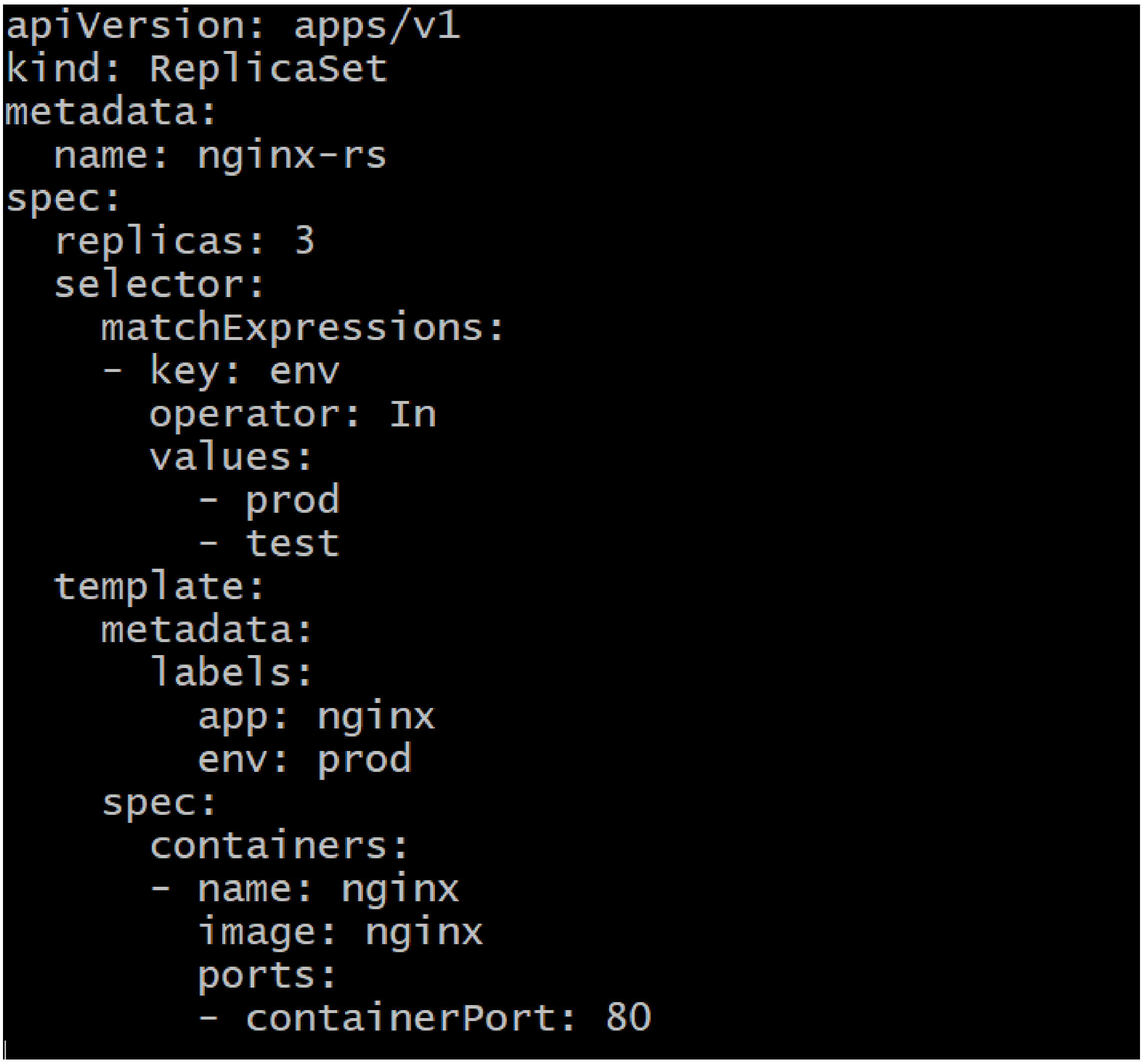
```
[root@master ~]# kubectl get rc nginx-rc
NAME       DESIRED   CURRENT   READY   AGE
nginx-rc   5         5         5       3m38s
[root@master ~]# kubectl get pods
NAME             READY   STATUS    RESTARTS   AGE
nginx-rc-7gcn2   1/1     Running   0          3m50s
nginx-rc-9zfmv   1/1     Running   0          25s
nginx-rc-kxmdx   1/1     Running   0          3m50s
nginx-rc-qcg2z   1/1     Running   0          25s
nginx-rc-w2qjk   1/1     Running   0          3m50s
[root@master ~]#
```

Or you can do another way by changing the yaml file keep the replicas as 5.

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

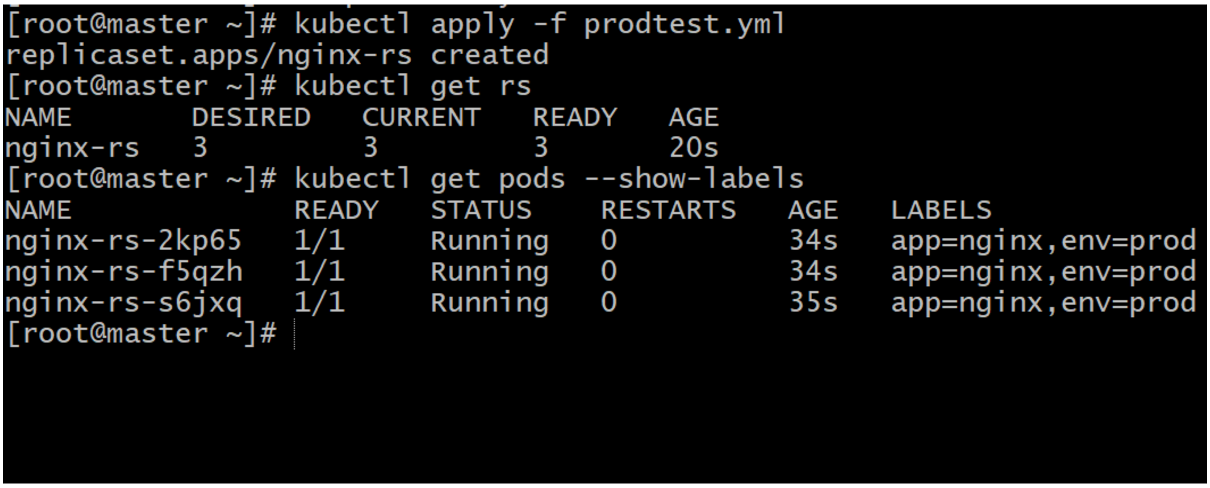
```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
spec:
  replicas: 3
  selector:
    matchExpressions:
    - key: env
      operator: In
      values:
        - prod
        - test
  template:
    metadata:
      labels:
```

```
      app: nginx

      env: prod

    spec:

     containers:

     - name: nginx

      image: nginx

      ports:

      - containerPort: 80
```



```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
spec:
  replicas: 3
  selector:
    matchExpressions:
    - key: env
      operator: In
      values:
        - prod
        - test
  template:
    metadata:
      labels:
        app: nginx
        env: prod
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

- **kubectl apply -f prodtest.yml**
- **kubectl get rs**
- **kubectl get pods --show-labels**



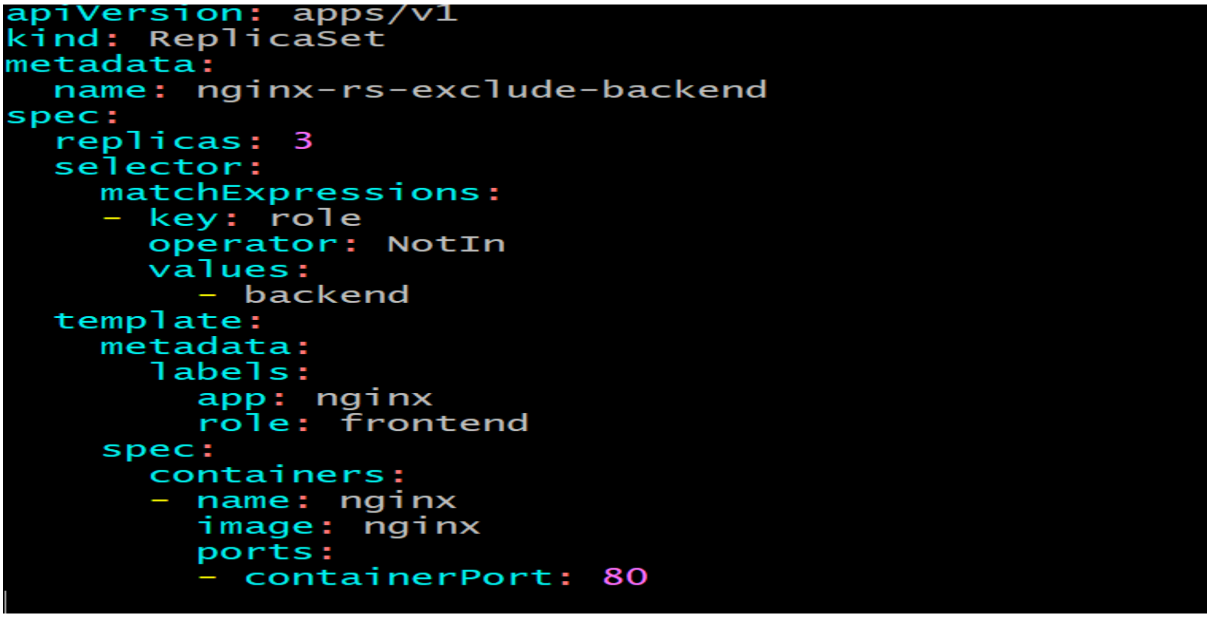
```
[root@master ~]# kubectl apply -f prodtest.yml
replicaset.apps/nginx-rs created
[root@master ~]# kubectl get rs
NAME       DESIRED   CURRENT   READY   AGE
nginx-rs   3         3         3       20s
[root@master ~]# kubectl get pods --show-labels
NAME             READY   STATUS    RESTARTS   AGE   LABELS
nginx-rs-2kp65   1/1     Running   0          34s   app=nginx,env=prod
nginx-rs-f5qzh   1/1     Running   0          34s   app=nginx,env=prod
nginx-rs-s6jxq   1/1     Running   0          35s   app=nginx,env=prod
[root@master ~]#
```

6. Deploy a ReplicaSet that excludes pods with the label backend.

- **vi backend.yml**

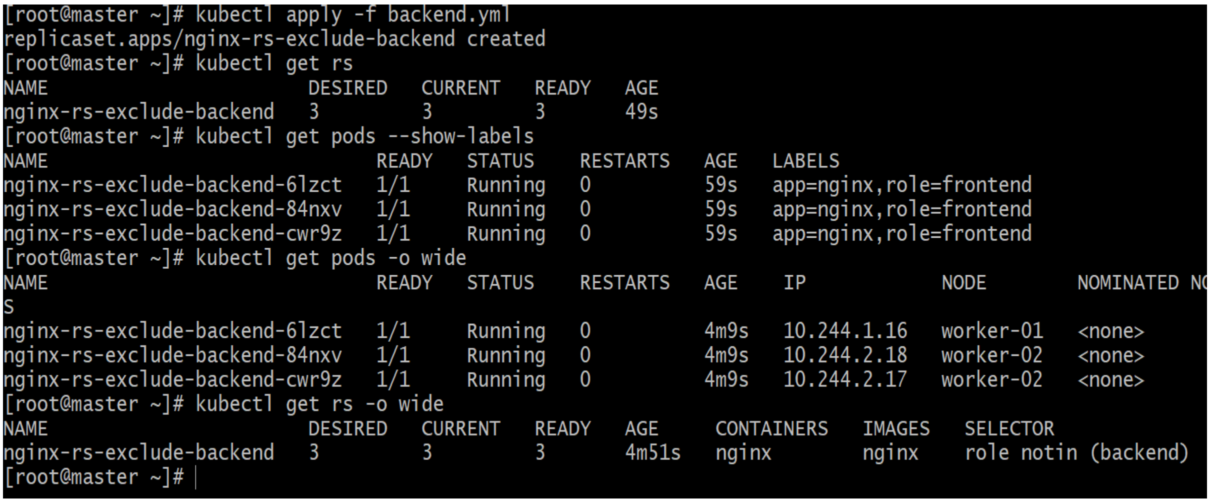
```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs-exclude-backend
spec:
  replicas: 3
  selector:
    matchExpressions:
    - key: role
```

```
      operator: NotIn
      values:
        - backend
  template:
    metadata:
      labels:
        app: nginx
        role: frontend
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```



```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs-exclude-backend
spec:
  replicas: 3
  selector:
    matchExpressions:
    - key: role
      operator: NotIn
      values:
        - backend
  template:
    metadata:
      labels:
        app: nginx
        role: frontend
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

- **kubectl apply -f backend.yml**
- **kubectl get pods --show -labels**

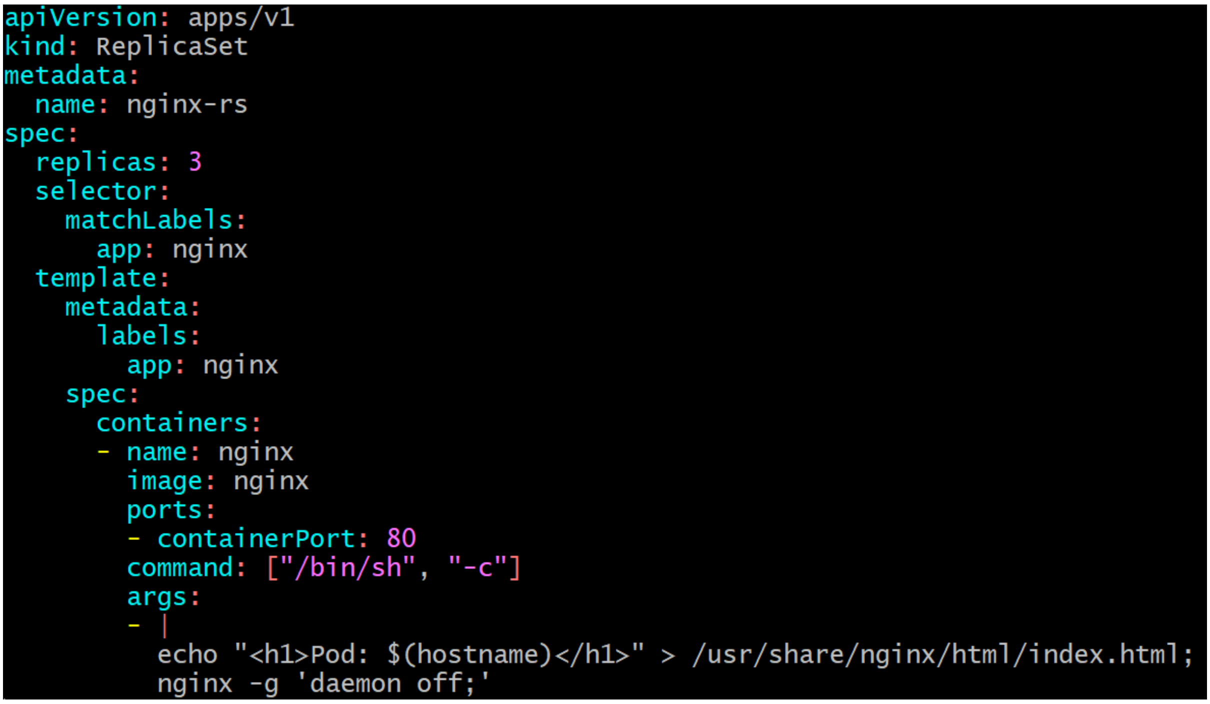


```
[root@master ~]# kubectl apply -f backend.yml
replicaset.apps/nginx-rs-exclude-backend created
[root@master ~]# kubectl get rs
NAME                       DESIRED   CURRENT   READY   AGE
nginx-rs-exclude-backend   3         3         3       49s
[root@master ~]# kubectl get pods --show-labels
NAME                             READY   STATUS    RESTARTS   AGE   LABELS
nginx-rs-exclude-backend-6lzct   1/1     Running   0          59s   app=nginx,role=frontend
nginx-rs-exclude-backend-84nxv   1/1     Running   0          59s   app=nginx,role=frontend
nginx-rs-exclude-backend-cwr9z   1/1     Running   0          59s   app=nginx,role=frontend
[root@master ~]# kubectl get pods -o wide
NAME                             READY   STATUS    RESTARTS   AGE    IP            NODE        NOMINATED NO
S
nginx-rs-exclude-backend-6lzct   1/1     Running   0          4m9s   10.244.1.16   worker-01   <none>
nginx-rs-exclude-backend-84nxv   1/1     Running   0          4m9s   10.244.2.18   worker-02   <none>
nginx-rs-exclude-backend-cwr9z   1/1     Running   0          4m9s   10.244.2.17   worker-02   <none>
[root@master ~]# kubectl get rs -o wide
NAME                       DESIRED   CURRENT   READY   AGE     CONTAINERS   IMAGES   SELECTOR
nginx-rs-exclude-backend   3         3         3       4m51s   nginx        nginx    role notin (backend)
[root@master ~]#
```

7. Test load balancing across multiple pods using a NodePort service.

- **vi nginxrs.yml**

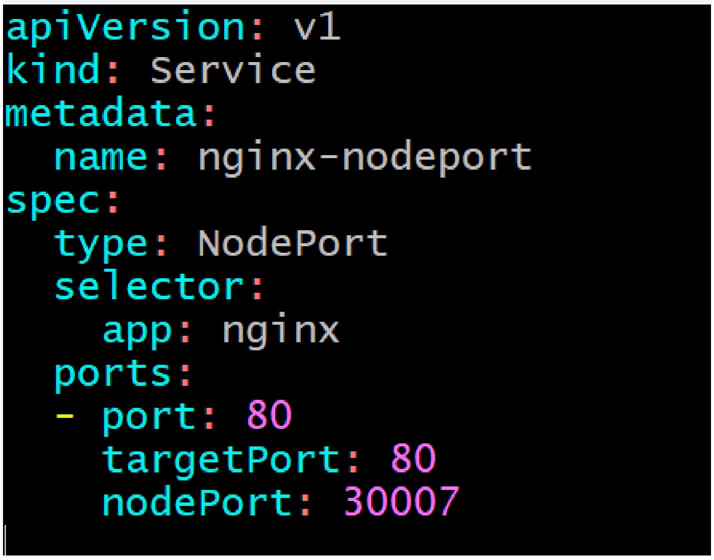
```
apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: nginx-rs

spec:

  replicas: 3

  selector:

    matchLabels:

      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
        command: ["/bin/sh", "-c"]
        args:
        - |
          echo "<h1>Pod: $(hostname)</h1>" >
/usr/share/nginx/html/index.html;
          nginx -g 'daemon off;'
```



```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx-rs
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
        command: ["/bin/sh", "-c"]
        args:
        - |
          echo "<h1>Pod: $(hostname)</h1>" > /usr/share/nginx/html/index.html;
          nginx -g 'daemon off;'
```

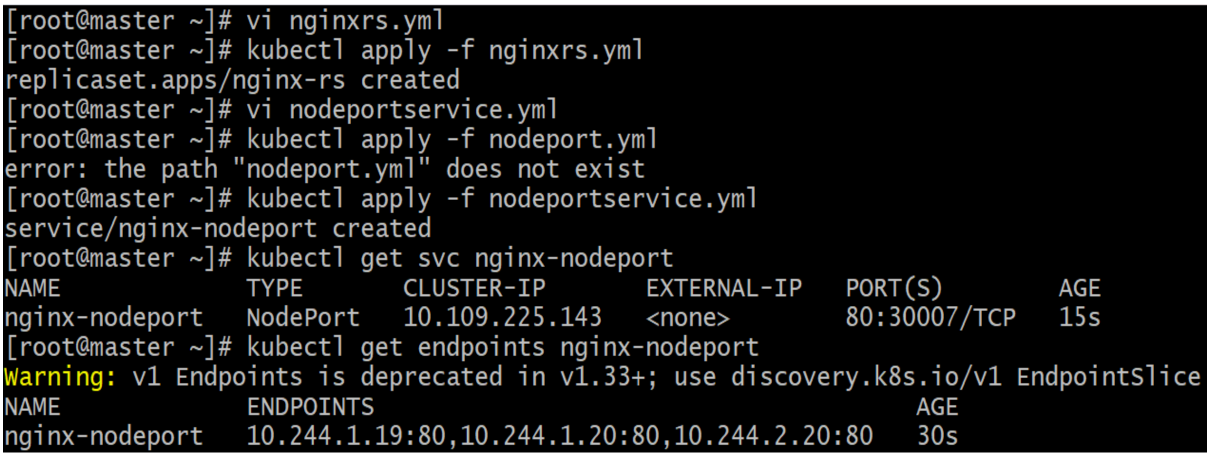
- **vi nodeportservice.yml**

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
  - port: 80
    targetPort: 80
    nodePort: 30007
```



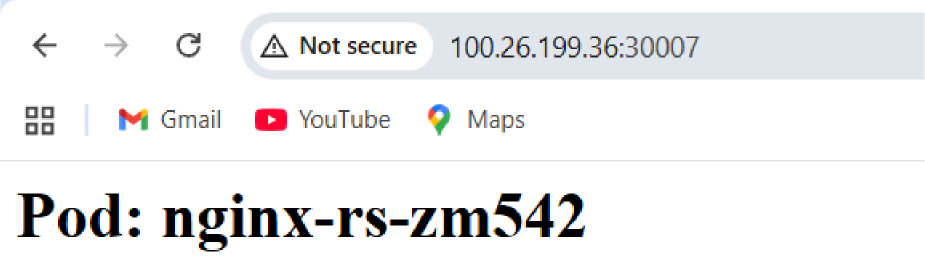
```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
  - port: 80
    targetPort: 80
    nodePort: 30007
```

- **kubectl apply -f nodeportservice.yml**
- **kubectl get svc nginx-nodeport**
- **kubectl get endpoints nginx-nodeport**



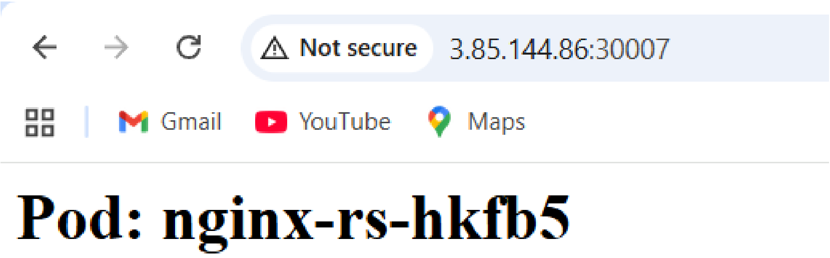
```
[root@master ~]# vi nginxrs.yml
[root@master ~]# kubectl apply -f nginxrs.yml
replicaset.apps/nginx-rs created
[root@master ~]# vi nodeportservice.yml
[root@master ~]# kubectl apply -f nodeport.yml
error: the path "nodeport.yml" does not exist
[root@master ~]# kubectl apply -f nodeportservice.yml
service/nginx-nodeport created
[root@master ~]# kubectl get svc nginx-nodeport
NAME             TYPE       CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
nginx-nodeport   NodePort   10.109.225.143   <none>        80:30007/TCP   15s
[root@master ~]# kubectl get endpoints nginx-nodeport
Warning: v1 Endpoints is deprecated in v1.33+; use discovery.k8s.io/v1 EndpointSlice
NAME             ENDPOINTS                                         AGE
nginx-nodeport   10.244.1.19:80,10.244.1.20:80,10.244.2.20:80    30s
```

Search on browser with 3 node machines public ip and port number 30007.

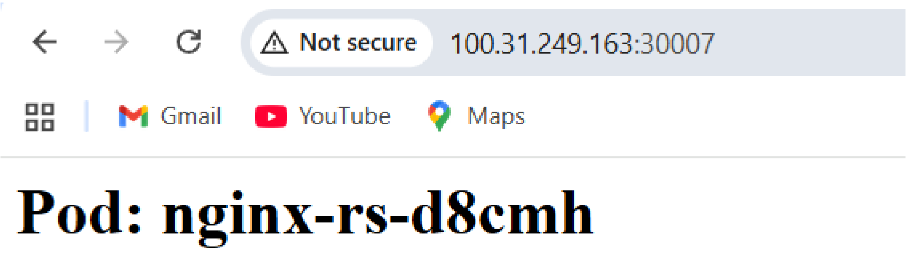


100.26.199.36:30007

Pod: nginx-rs-zm542



Pod: nginx-rs-hkfb5



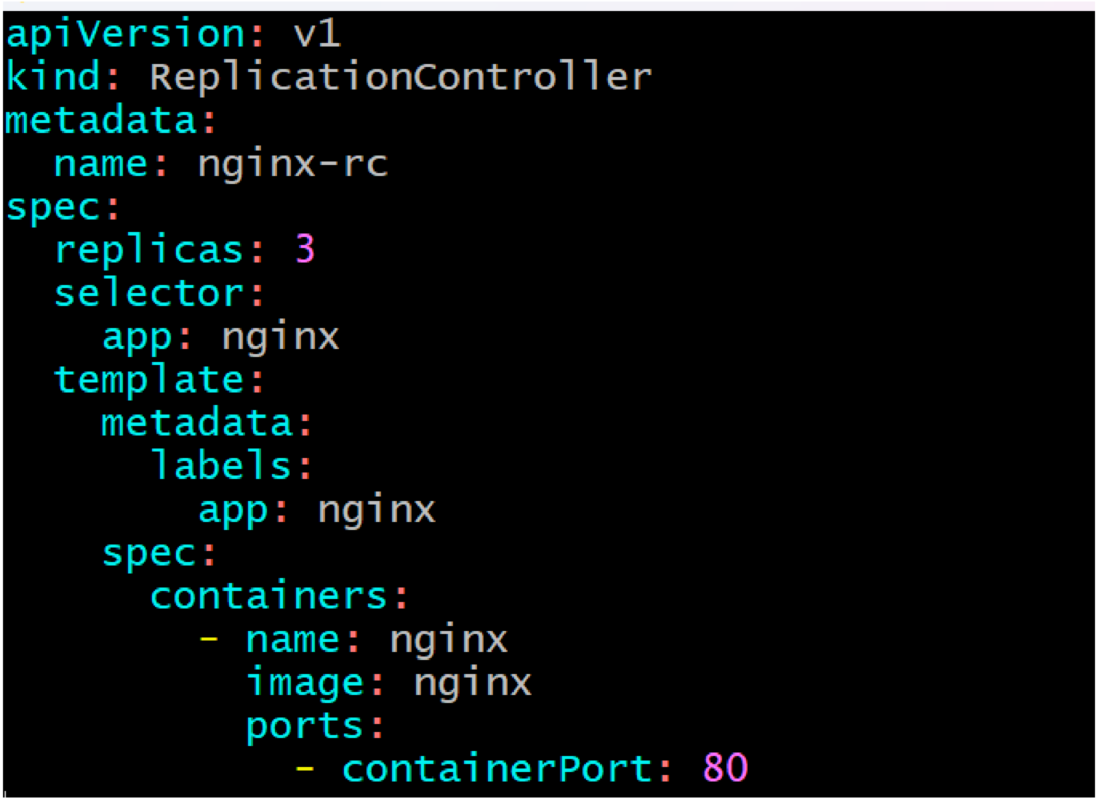
Pod: nginx-rs-d8cmh

8. Delete a ReplicationController without affecting the running pods.

Create a replication controller file or use the file in 3rd question.

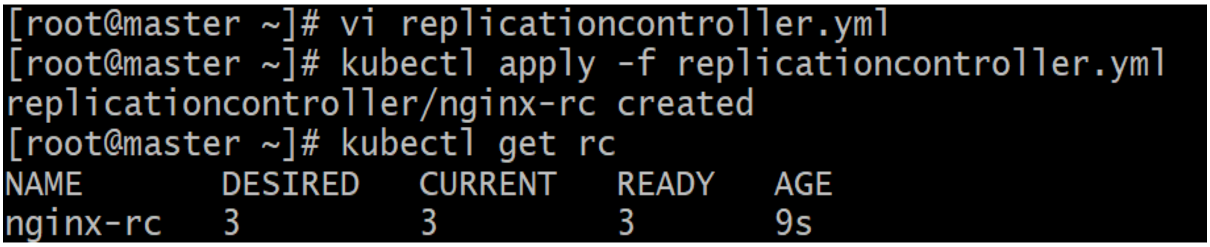
- **vi replicationcontroller.yml**

```
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
```



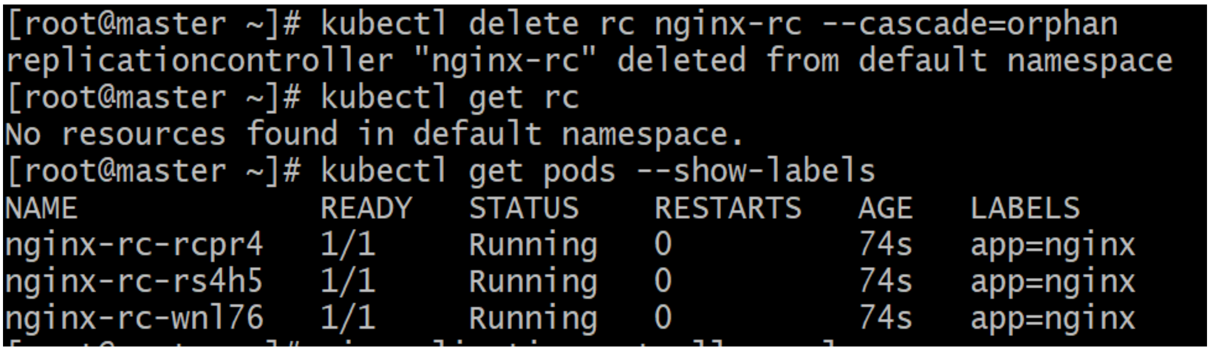
```
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx-rc
spec:
  replicas: 3
  selector:
    app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          ports:
            - containerPort: 80
```

- **kubectl apply -f replicationcontroller.yml**



```
[root@master ~]# vi replicationcontroller.yml
[root@master ~]# kubectl apply -f replicationcontroller.yml
replicationcontroller/nginx-rc created
[root@master ~]# kubectl get rc
NAME        DESIRED   CURRENT   READY   AGE
nginx-rc    3         3         3       9s
```

- **kubectl delete rc nginx-rc --cascade=orphan**
- **kubectl get rc**
- **kubectl get pods --show-labels**



```
[root@master ~]# kubectl delete rc nginx-rc --cascade=orphan
replicationcontroller "nginx-rc" deleted from default namespace
[root@master ~]# kubectl get rc
No resources found in default namespace.
[root@master ~]# kubectl get pods --show-labels
NAME             READY   STATUS    RESTARTS   AGE   LABELS
nginx-rc-rcpr4   1/1     Running   0          74s   app=nginx
nginx-rc-rs4h5   1/1     Running   0          74s   app=nginx
nginx-rc-wnl76   1/1     Running   0          74s   app=nginx
```