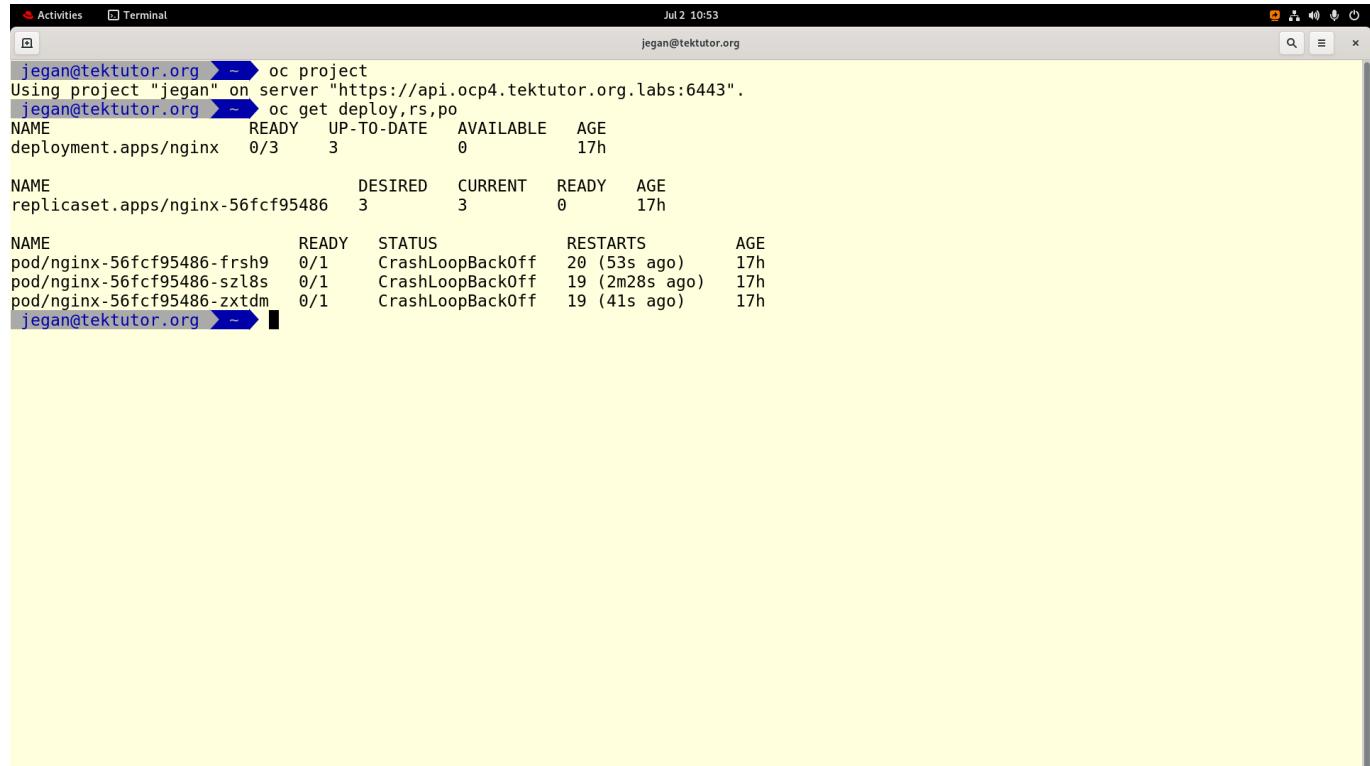


# Day 2

## Lab - Listing multiple resource in your project using a single command

```
oc get deploy,rs,po  
oc get all
```

### Expected output



The screenshot shows a terminal window titled 'Terminal' with the command 'jegan@tektutor.org' and the date 'Jul 2 10:53'. The session starts with 'oc project' followed by 'Using project "jegan" on server "https://api.ocp4.tektutor.org.labs:6443".' Then, the user runs 'oc get deploy,rs,po' which lists deployment resources:

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/nginx	0/3	3	0	17h

Next, the user runs 'oc get replicaset' which lists replicaset resources:

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/nginx-56fcf95486	3	3	0	17h

Finally, the user runs 'oc get pod' which lists pod resources:

NAME	READY	STATUS	RESTARTS	AGE
pod/nginx-56fcf95486-frsh9	0/1	CrashLoopBackOff	20 (53s ago)	17h
pod/nginx-56fcf95486-sz18s	0/1	CrashLoopBackOff	19 (2m28s ago)	17h
pod/nginx-56fcf95486-zxtdm	0/1	CrashLoopBackOff	19 (41s ago)	17h

## Lab - Delete a nginx deployment

```
oc get deploy,rs,po  
oc delete deploy/nginx  
oc get deploy,rs,po
```

## Expected output

```
jegan@tektutor.org ➔ oc project
Using project "jegan" on server "https://api.ocp4.tektutor.org.labs:6443".
jegan@tektutor.org ➔ oc get deploy,rs,po
NAME           READY  UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx  0/3      3           0          17h
NAME          DESIRED  CURRENT  READY   AGE
replicaset.apps/nginx-56fcf95486  3         3         0       17h
NAME          READY  STATUS    RESTARTS  AGE
pod/nginx-56fcf95486-frsh9  0/1    CrashLoopBackOff  20 (53s ago)  17h
pod/nginx-56fcf95486-sz18s  0/1    CrashLoopBackOff  19 (2m28s ago)  17h
pod/nginx-56fcf95486-zxtdm  0/1    CrashLoopBackOff  19 (41s ago)   17h
jegan@tektutor.org ➔ oc delete deploy/nginx
deployment.apps "nginx" deleted
jegan@tektutor.org ➔ oc get deploy,rs,po
No resources found in jegan namespace.
jegan@tektutor.org ➔
```

## Lab - Creating a nginx deployment with bitnami/nginx image

### Points to remember

- the bitnami container images follow the Openshift best practices and conventions
- the bitnami container images are root-less, i.e the applications runs with non-root privilege as per openshift conventions
- hence, mostly all bitnami images are safer to use in Openshift

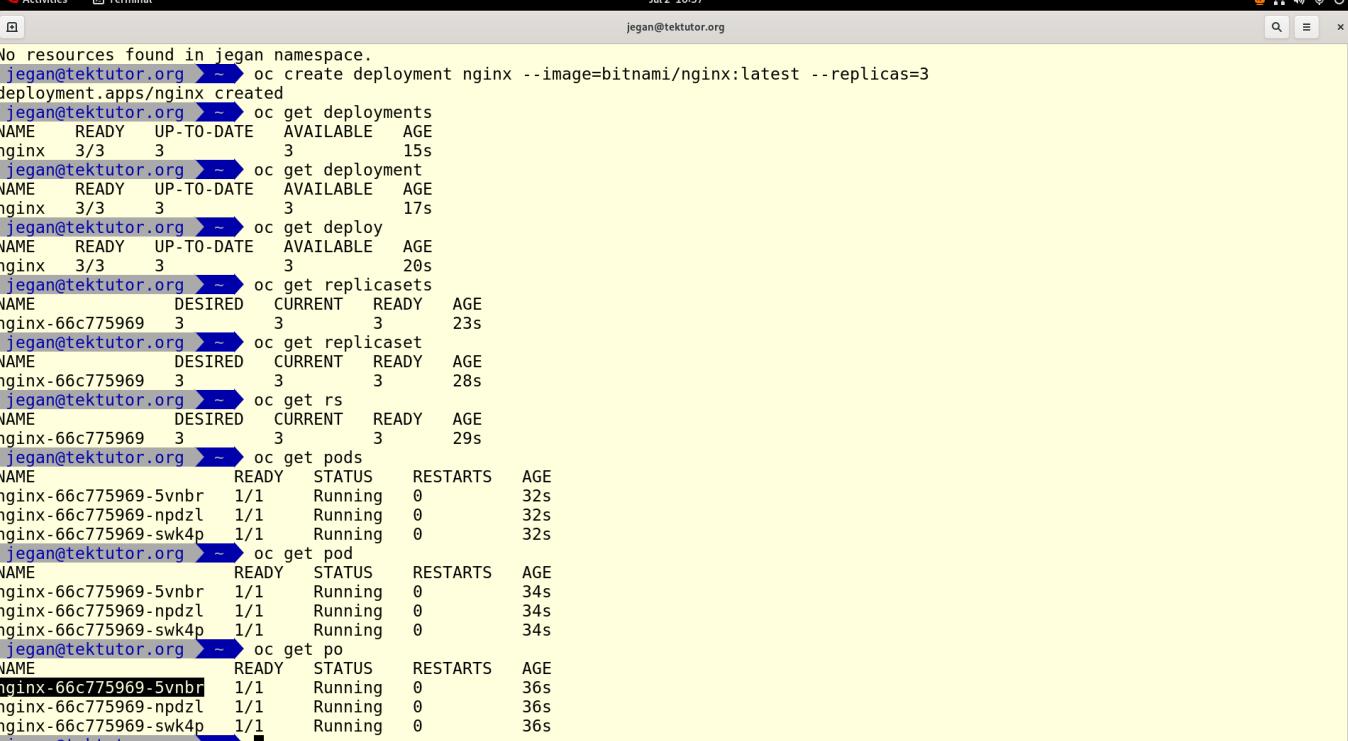
```
oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
oc get deployments
oc get deployment
oc get deploy

oc get replicasesets
oc get replicaset
oc get rs

oc get pods
oc get pod
oc get po

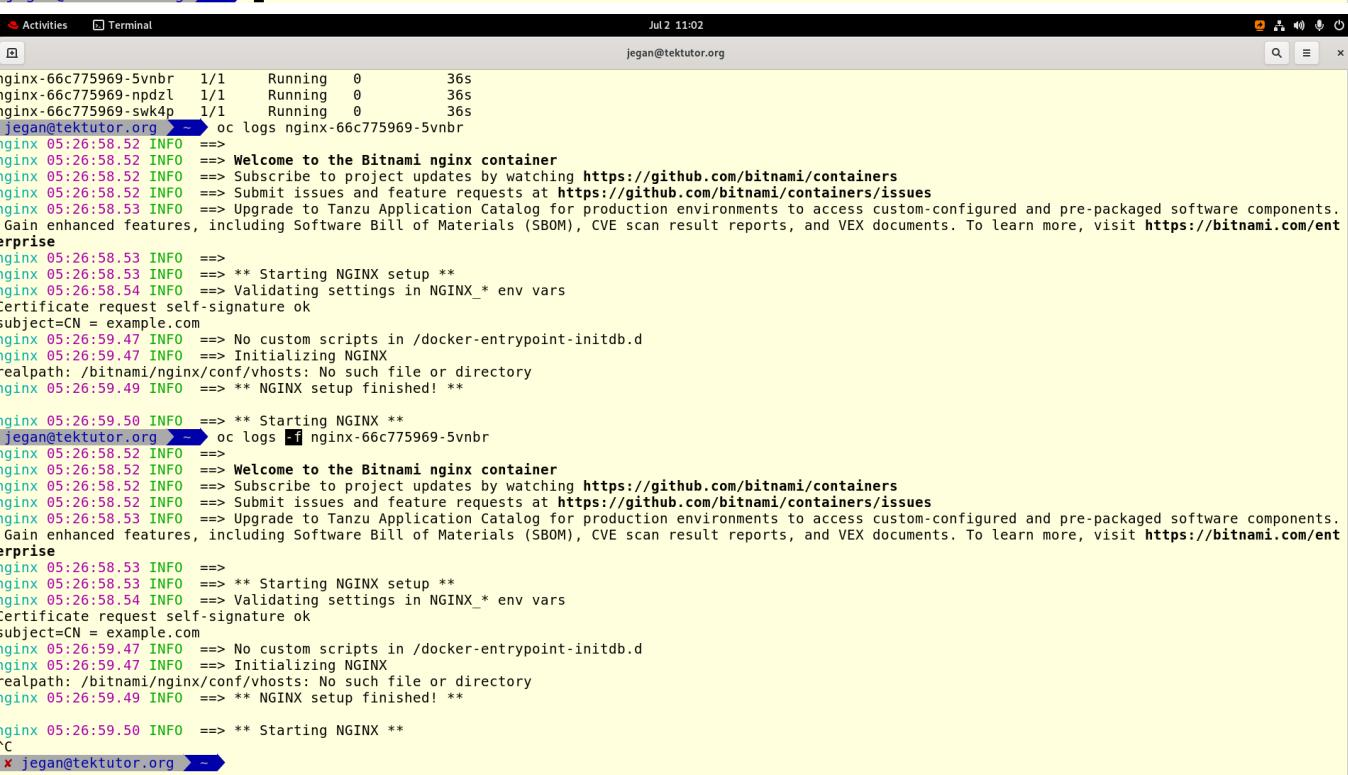
oc logs nginx-66c775969-5vnbr
oc logs -f nginx-66c775969-5vnbr
```

## Expected output



```
jegan@tektutor.org ➤ oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
deployment.apps/nginx created
jegan@tektutor.org ➤ oc get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx    3/3     3            3           15s
jegan@tektutor.org ➤ oc get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx    3/3     3            3           17s
jegan@tektutor.org ➤ oc get deploy
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
nginx    3/3     3            3           20s
jegan@tektutor.org ➤ oc get replicases
NAME      DESIRED  CURRENT  READY   AGE
nginx-66c775969 3        3        3       23s
jegan@tektutor.org ➤ oc get replicaset
NAME      DESIRED  CURRENT  READY   AGE
nginx-66c775969 3        3        3       28s
jegan@tektutor.org ➤ oc get rs
NAME      DESIRED  CURRENT  READY   AGE
nginx-66c775969 3        3        3       29s
jegan@tektutor.org ➤ oc get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx-66c775969-5vnbr 1/1     Running   0          32s
nginx-66c775969-npdzl 1/1     Running   0          32s
nginx-66c775969-swk4p  1/1     Running   0          32s
jegan@tektutor.org ➤ oc get pod
NAME      READY   STATUS    RESTARTS   AGE
nginx-66c775969-5vnbr 1/1     Running   0          34s
nginx-66c775969-npdzl 1/1     Running   0          34s
nginx-66c775969-swk4p  1/1     Running   0          34s
jegan@tektutor.org ➤ oc get po
NAME      READY   STATUS    RESTARTS   AGE
nginx-66c775969-5vnbr 1/1     Running   0          36s
nginx-66c775969-npdzl 1/1     Running   0          36s
nginx-66c775969-swk4p  1/1     Running   0          36s
jegan@tektutor.org ➤

```

```
jegan@tektutor.org ➤ oc logs nginx-66c775969-5vnbr
nginx 05:26:58.52 INFO ===>
nginx 05:26:58.52 INFO ==> Welcome to the Bitnami nginx container
nginx 05:26:58.52 INFO ==> Subscribe to project updates by watching https://github.com/bitnami/containers
nginx 05:26:58.52 INFO ==> Submit issues and feature requests at https://github.com/bitnami/containers/issues
nginx 05:26:58.53 INFO ==> Upgrade to Tanzu Application Catalog for production environments to access custom-configured and pre-packaged software components.
Gain enhanced features, including Software Bill of Materials (SBOM), CVE scan result reports, and VEX documents. To learn more, visit https://bitnami.com/enterprise
nginx 05:26:58.53 INFO ==>
nginx 05:26:58.53 INFO ==> ** Starting NGINX setup **
nginx 05:26:58.54 INFO ==> Validating settings in NGINX_* env vars
Certificate request self-signature ok
subject=CN = example.com
nginx 05:26:59.47 INFO ==> No custom scripts in /docker-entrypoint-initdb.d
nginx 05:26:59.47 INFO ==> Initializing NGINX
realpath: /bitnami/nginx/conf/vhosts: No such file or directory
nginx 05:26:59.49 INFO ==> ** NGINX setup finished! **

nginx 05:26:59.50 INFO ==> ** Starting NGINX **
jegan@tektutor.org ➤ oc logs nginx-66c775969-5vnbr
nginx 05:26:58.52 INFO ===>
nginx 05:26:58.52 INFO ==> Welcome to the Bitnami nginx container
nginx 05:26:58.52 INFO ==> Subscribe to project updates by watching https://github.com/bitnami/containers
nginx 05:26:58.52 INFO ==> Submit issues and feature requests at https://github.com/bitnami/containers/issues
nginx 05:26:58.53 INFO ==> Upgrade to Tanzu Application Catalog for production environments to access custom-configured and pre-packaged software components.
Gain enhanced features, including Software Bill of Materials (SBOM), CVE scan result reports, and VEX documents. To learn more, visit https://bitnami.com/enterprise
nginx 05:26:58.53 INFO ==>
nginx 05:26:58.53 INFO ==> ** Starting NGINX setup **
nginx 05:26:58.54 INFO ==> Validating settings in NGINX_* env vars
Certificate request self-signature ok
subject=CN = example.com
nginx 05:26:59.47 INFO ==> No custom scripts in /docker-entrypoint-initdb.d
nginx 05:26:59.47 INFO ==> Initializing NGINX
realpath: /bitnami/nginx/conf/vhosts: No such file or directory
nginx 05:26:59.49 INFO ==> ** NGINX setup finished! **

nginx 05:26:59.50 INFO ==> ** Starting NGINX **
^C
x jegan@tektutor.org ➤

```

## Lab - Finding more details about a running pod

```
oc get po
oc describe pod/nginx-66c775969-5vnbr
```

## Expected output

```
jegan@tektutor.org ➔ ~ oc get po
NAME           READY   STATUS    RESTARTS   AGE
nginx-66c775969-5vnbr  1/1    Running   0          6m8s
nginx-66c775969-npdzl  1/1    Running   0          6m8s
nginx-66c775969-swk4p  1/1    Running   0          6m8s
jegan@tektutor.org ➔ ~ oc get po -o wide
NAME           READY   STATUS    RESTARTS   AGE   IP                                NODE           NOMINATED NODE   READINESS
GATES
nginx-66c775969-5vnbr  1/1    Running   0          6m15s  10.131.0.13   worker-2.ocp4.tektutor.org.labs <none>        <none>
nginx-66c775969-npdzl  1/1    Running   0          6m15s  10.128.2.7   worker-1.ocp4.tektutor.org.labs <none>        <none>
nginx-66c775969-swk4p  1/1    Running   0          6m15s  10.129.0.6   master-2.ocp4.tektutor.org.labs <none>        <none>
jegan@tektutor.org ➔ ~ oc describe pod/nginx-66c775969-5vnbr
Name:           nginx-66c775969-5vnbr
Namespace:      jegan
Priority:       0
Service Account: default
Node:          worker-2.ocp4.tektutor.org.labs/192.168.122.192
Start Time:     Tue, 02 Jul 2024 10:56:55 +0530
Labels:         app=nginx
               pod-template-hash=66c775969
Annotations:    k8s.ovn.org/pod-networks:
                  {"default":{"ip_addresses":["10.131.0.13/23"],"mac_address":"0a:58:0a:83:00:0d","gateway_ips":["10.131.0.1"]}, "routes": [{"dest": "10.128.0.0..."}], k8s.v1.cni.cncf.io/network-status: [{"name": "ovn-kubernetes", "interface": "eth0", "ips": ["10.131.0.13"], "mac": "0a:58:0a:83:00:0d", "default": true, "dns": {}}], openshift.io/scc: restricted-v2, seccomp.security.alpha.kubernetes.io/pod: runtime/default}
Status:        Running
```

```

Activities Terminal Jul 2 11:04 jegan@tektutor.org
        "10.131.0.13"
    ],
    "mac": "0a:58:0a:83:00:0d",
    "default": true,
    "dns": {}
}
],
openshift.io/scc: restricted-v2
seccomp.security.alpha.kubernetes.io/pod: runtime/default
Status: Running
SeccompProfile: RuntimeDefault
IP: 10.131.0.13
IPs:
IP: 10.131.0.13
Controlled By: ReplicaSet/nginx-66c775969
Containers:
nginx:
  Container ID: cri-o://8bfe9d0cd3c48ebe854a1e5ffde8ef442d42d86b200665a55762e0a33227d838
  Image: bitnami/nginx:latest
  Image ID: docker.io/bitnami/nginx@sha256:0f6c0a48e874645b7c2732ebc5f196787318050de9334e03a457813fd80115cf
  Port: <none>
  Host Port: <none>
  State: Running
    Started: Tue, 02 Jul 2024 10:56:58 +0530
  Ready: True
  Restart Count: 0
  Environment: <none>
  Mounts:
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6xf4n (ro)
Conditions:
  Type Status
  Initialized True
  Ready True
  ContainersReady True
  PodScheduled True
Volumes:
  kube-api-access-6xf4n:
    Type: Projected (a volume that contains injected data from multiple sources)
      State: Running
      Started: Tue, 02 Jul 2024 10:56:58 +0530
      Ready: True
      Restart Count: 0
      Environment: <none>
      Mounts:
        /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-6xf4n (ro)
Conditions:
  Type Status
  Initialized True
  Ready True
  ContainersReady True
  PodScheduled True
Volumes:
  kube-api-access-6xf4n:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
    ConfigMapName: openshift-service-ca.crt
    ConfigMapOptional: <nil>
  QoS Class: BestEffort
  Node-Selectors: <none>
  Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type Reason Age From Message
  ---- ---- - - - -
  Normal Scheduled 6m35s default-scheduler Successfully assigned jegan/nginx-66c775969-5vnbr to worker-2.ocp4.tektutor.org.labs
  Normal AddedInterface 6m35s multus Add eth0 [10.131.0.13/23] from ovn-kubernetes
  Normal Pulling 6m35s kubelet Pulling image "bitnami/nginx:latest"
  Normal Pulled 6m32s kubelet Successfully pulled image "bitnami/nginx:latest" in 2.478s (2.478s including waiting)
  Normal Created 6m32s kubelet Created container nginx
  Normal Started 6m32s kubelet Started container nginx
jegan@tektutor.org ➜

```

Info - What happens when we deploy an application in Openshift with the below command

```
oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
```

- the oc client tool makes a REST call to API Server requesting it to create nginx deployment with image bitnami/nginx:latest with 3 Pods in it
- API Server receives the request from oc client tool, it then creates a new

Deployment record in etcd database

- API Server then sends a broadcasting event saying a New Deployment is created
- the event is received by Deployment Controller, it then sends a REST API call to API Server, requesting it to create a ReplicaSet for nginx deployment
- API Server receives the request from Deployment Controller and it creates a ReplicaSet record in etcd database
- API Server sends a broadcasting event saying a New ReplicaSet is created
- the event is received by ReplicaSet Controller, it then understands 3 Pods are mentioned in the Desired count, hence it makes REST call to API server to create 3 Pods
- the API Server creates 3 Pod records in etcd database and it sends broadcasting events say new Pod created. One such event will be broadcasted for every New Pod created.
- the scheduler receives the event and it sends scheduling recommendation for each Pod to the API Servers
- API Server receives the REST call from Scheduler, it then retrieves the existing Pod records from etcd and it updates the Pod records with the node details as recommended by Scheduler
- API Server then sends broadcasting events that Pod scheduler to so an do nodes
- the kubelet container agent that runs in node where the Pod is scheduled receives the event, it then downloads the container images, creates the container and starts the container
- the kubelet then sends the status of the container running on that nodes to API Server via REST calls
- the API Servers updates the status of the Pod based on the status it received from kubelet
- the kubelet keeps sending this kind of container status updates to API Server like a heart beat fashion
- the API keeps the Pod status updated based on the status reported by kubelet

## Lab - Pod Port forwarding to access the web page served by a single pod

### Points to remember

- port forwarding must be used only for testing purpose
- generally used by developers
- should not be used in production
- for production use we must service

```
oc get pods  
oc port-forward nginx-66c775969-zxz5f 9090:8080
```

From another terminal, you may access the web page served by the pod

```
curl http://localhost:9090
```

## Expected output

```
jegan@tektutor.org ~
jegan@tektutor.org ~
jegan@tektutor.org ~
jegan@tektutor.org ~
jegan@tektutor.org ~

nginx-66c775969-w8zt2 1/1  Running  0          46m  10.131.0.43  worker-2.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-wbwst 1/1  Running  0          46m  10.128.0.54  master-1.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-wf9cr 1/1  Running  0          46m  10.128.0.61  master-1.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-wj7t9  1/1  Running  0          46m  10.131.0.44  worker-2.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-wjdz6  1/1  Running  0          46m  10.131.0.26  worker-2.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-wvwdw7 1/1  Running  0          46m  10.129.0.21  master-2.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-x7wvp  1/1  Running  0          46m  10.129.0.7   master-2.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-x885l  1/1  Running  0          46m  10.129.0.16  master-2.ocp4.tektutor.org.labs <none>
e>           <none>
nginx-66c775969-zxz5f  1/1  Running  0          46m  10.129.0.17  master-2.ocp4.tektutor.org.labs <none>
e>           <none>

jegan@tektutor.org ~ ➔ ping 10.129.0.16
PING 10.129.0.16 (10.129.0.16) 56(84) bytes of data.
^C
--- 10.129.0.16 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1038ms

x jegan@tektutor.org ~ ➔ oc port-forward nginx-66c775969-zxz5f 9090:8080
Forwarding from 127.0.0.1:9090 -> 8080
Forwarding from [::1]:9090 -> 8080
Handling connection for 9090

jegan@tektutor.org ~
jegan@tektutor.org ~
jegan@tektutor.org ~
jegan@tektutor.org ~
jegan@tektutor.org ~

jegan@tektutor.org ~ ➔ curl http://localhost:9090
!DOCTYPE html
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

## Info - Kubernetes/OpenShift Service

- Service represents a group of pods from a single Deployment
- Service can be accessed by its Name - Service discovery
- Types of Services

1. ClusterIP - Internal Service ( Supports inbuilt load-balancing )
2. NodePort - External Service ( supports inbuilt load-balancing )
3. LoadBalancer - External Service ( spins-off an external Load Balancer in AWS/Azure/GCP )

## Lab - Creating an internal service for nginx deployment

- In the below command, type=ClusterIP indicates we want to create an internal clusterIP service
- this type of service is practically used for databases which are accessed only within the cluster by frontend pods
- ClusterIP service is accessible only within the openshift cluster
- Pod running with the same cluster can access this type of service
- For clusterIP service, kube-proxy which runs in every nodes supports the load-balancing
- as this load-balancing is an internal implementation of Kubernetes/OpenShift there will not be any extra charge even if our OpenShift runs in public cloud like AWS/Azure/GCP for the service we created
- the port 8080 is the port where nginx web server is listening internally with the Pod container
- the endpoints for the service are created by a controller called Endpoint controller
- the endpoint controller will be watching for new service, deployment scale up/down, redeployment deletion, pod replaced with another pod

Let's create an internal service for nginx deployment

```
oc get deploy
oc expose deploy/nginx --type=ClusterIP --port=8080
oc get endpoints
oc get services
oc get service
oc get svc
oc describe svc/nginx
oc scale deploy/nginx --replicas=3
oc get endpoints
oc get svc
oc describe svc/nginx
```

## Expected output

```
jegan@tektutor.org ➤ oc expose deploy/nginx --type=ClusterIP --port=8080
service/nginx exposed
jegan@tektutor.org ➤ oc get services
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)    AGE
nginx    ClusterIP  172.30.241.122 <none>       8080/TCP   3s
jegan@tektutor.org ➤ oc get service
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)    AGE
nginx    ClusterIP  172.30.241.122 <none>       8080/TCP   5s
jegan@tektutor.org ➤ oc get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP  PORT(S)    AGE
nginx    ClusterIP  172.30.241.122 <none>       8080/TCP   7s
jegan@tektutor.org ➤ oc describe svc/nginx
Name:            nginx
Namespace:       jegan
Labels:          app=nginx
Annotations:    <none>
Selector:        app=nginx
Type:           ClusterIP
IP Family Policy: SingleStack
IP Families:    IPv4
IP:              172.30.241.122
IPs:             172.30.241.122
Port:            <unset>  8080/TCP
TargetPort:      8080/TCP
Endpoints:      10.128.0.53:8080,10.128.0.54:8080,10.128.0.57:8080 + 47 more...
Session Affinity: None
Events:          <none>
jegan@tektutor.org ➤ oc get endpoints
NAME      ENDPOINTS                                     AGE
nginx   10.128.0.53:8080,10.128.0.54:8080,10.128.0.57:8080 + 47 more...  25s
jegan@tektutor.org ➤
```

```
jegan@tektutor.org ➤ oc scale deploy/nginx --replicas=3
deployment.apps/nginx scaled
jegan@tektutor.org ➤ oc get endpoints
NAME      ENDPOINTS                                     AGE
nginx   10.128.0.53:8080,10.128.0.54:8080,10.128.0.57:8080 + 47 more...  25s
jegan@tektutor.org ➤ oc get endpoints
NAME      ENDPOINTS                                     AGE
nginx   10.129.0.21:8080,10.129.0.6:8080,10.130.0.53:8080  7m57s
jegan@tektutor.org ➤ oc describe svc/nginx
Name:            nginx
Namespace:       jegan
Labels:          app=nginx
Annotations:    <none>
Selector:        app=nginx
Type:           ClusterIP
IP Family Policy: SingleStack
IP Families:    IPv4
IP:              172.30.241.122
IPs:             172.30.241.122
Port:            <unset>  8080/TCP
TargetPort:      8080/TCP
Endpoints:      10.129.0.21:8080,10.129.0.6:8080,10.130.0.53:8080
Session Affinity: None
Events:          <none>
jegan@tektutor.org ➤
```

Accessing the clusterip internal service from a Pod that runs inside the same cluster

```
oc create deployment test --image=tektutor/spring-ms:1.0
oc rsh deploy/test
curl http://nginx:8080
exit
```

## Expected output

```

Port: <unset> 8080/TCP
TargetPort: 8080/TCP
Endpoints: 10.129.0.21:8080,10.129.0.6:8080,10.130.0.53:8080
Session Affinity: None
Events: <none>
jegan@tektutor.org ➤ curl http://nginx:8080
^C
x jegan@tektutor.org ➤ oc create deployment test --image=tektutor/spring-ms:1.0
deployment.apps/test created
jegan@tektutor.org ➤ oc get po -l app=test
NAME READY STATUS RESTARTS AGE
test-544c5f8667-qcjlk 0/1 ContainerCreating 0 6s
jegan@tektutor.org ➤ oc get po -l app=test -w
NAME READY STATUS RESTARTS AGE
test-544c5f8667-qcjlk 0/1 ContainerCreating 0 8s
test-544c5f8667-qcjlk 1/1 Running 0 21s
^C
x jegan@tektutor.org ➤ oc rsh deploy/test
sh-4.4$ curl http://nginx:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a><br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
sh-4.4$ exit
sh: exit: command not found
sh-4.4$ exit
exit
command terminated with exit code 127
x jegan@tektutor.org ➤

```

## Info - How the Service Discovery works in Kubernetes/OpenShift?

- When the Pod containers are created by kubelet container agent, it also configures the /etc/resolv.conf file in every Pod container
- the /etc/resolv.conf file has a nameserver with IP 172.30.0.10
- the dns-default service has Pod running in each node
- they will resolve the nginx service name to its respective service IP

Let's get inside the test pod that we created as part of test deployment

```
oc rsh deploy/test  
cat /etc/resolv.conf
```

On the local machine, you can list and see dns service

```
oc get svc -n openshift-dns
```

Expected output

The screenshot shows a terminal window with four tabs, all titled 'jegan@tektutor.org'. The current tab is active and displays the following command and its output:

```
jegan@tektutor.org ➤ oc get svc -n openshift-dns  
NAME      TYPE    CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE  
dns-default  ClusterIP  172.30.0.10 <none>        53/UDP,53/TCP,9154/TCP  8d  
jegan@tektutor.org ➤ oc describe svc/dns-default -n openshift-dns  
Name:            dns-default  
Namespace:       openshift-dns  
Labels:          dns.operator.openshift.io/owning-dns=default  
Annotations:     service.alpha.openshift.io/serving-cert-signed-by: openshift-service-serving-signer@1719194974  
                  service.beta.openshift.io/serving-cert-secret-name: dns-default-metrics-tls  
                  service.beta.openshift.io/serving-cert-signed-by: openshift-service-serving-signer@1719194974  
Selector:        dns.operator.openshift.io/daemonset-dns=default  
Type:            ClusterIP  
IP Family Policy: SingleStack  
IP Families:    IPv4  
IP:              172.30.0.10  
IPs:             172.30.0.10  
Port:            dns 53/UDP  
TargetPort:      dns/UDP  
Endpoints:       10.128.0.17:5353,10.128.2.6:5353,10.129.0.13:5353 + 2 more...  
Port:            dns-tcp 53/TCP  
TargetPort:      dns-tcp/TCP  
Endpoints:       10.128.0.17:5353,10.128.2.6:5353,10.129.0.13:5353 + 2 more...  
Port:            metrics 9154/TCP  
TargetPort:      metrics/TCP  
Endpoints:       10.128.0.17:9154,10.128.2.6:9154,10.129.0.13:9154 + 2 more...
```

```
jegan@tektutor.org ➔ oc get pods --all-namespaces | grep dns
openshift-dns-operator dns-operator-6f9bcd6678-zq9ff 2/2 Running 12 8d
openshift-dns dns-default-9qnjg 2/2 Running 12 8d
openshift-dns dns-default-cvmd5 2/2 Running 12 8d
openshift-dns dns-default-djtj6 2/2 Running 12 8d
openshift-dns dns-default-fbp1p 2/2 Running 12 8d
openshift-dns dns-default-h6rx5 2/2 Running 12 8d
openshift-dns node-resolver-62fgx 1/1 Running 6 8d
openshift-dns node-resolver-8rnqv 1/1 Running 6 8d
openshift-dns node-resolver-gnkjm 1/1 Running 6 8d
openshift-dns node-resolver-gvg9h 1/1 Running 6 8d
openshift-dns node-resolver-tnt5j 1/1 Running 6 8d

jegan@tektutor.org ➔ oc get pods -n openshift-dns | grep dns-default
dns-default-9qnjg 2/2 Running 12 8d
dns-default-cvmd5 2/2 Running 12 8d
dns-default-djtj6 2/2 Running 12 8d
dns-default-fbp1p 2/2 Running 12 8d
dns-default-h6rx5 2/2 Running 12 8d
jegan@tektutor.org ➔ oc get pods -n openshift-dns -o wide | grep dns-default
dns-default-9qnjg 2/2 Running 12 8d 10.129.0.13 master-2.ocp4.tektutor.org.labs <none> <none>
dns-default-cvmd5 2/2 Running 12 8d 10.128.0.17 master-1.ocp4.tektutor.org.labs <none> <none>
dns-default-djtj6 2/2 Running 12 8d 10.130.0.39 master-3.ocp4.tektutor.org.labs <none> <none>
dns-default-fbp1p 2/2 Running 12 8d 10.128.2.6 worker-1.ocp4.tektutor.org.labs <none> <none>
dns-default-h6rx5 2/2 Running 12 8d 10.131.0.6 worker-2.ocp4.tektutor.org.labs <none> <none>

jegan@tektutor.org ➔ oc get deploy -n openshift-dns
No resources found in openshift-dns namespace.

jegan@tektutor.org ➔ oc get svc -n openshift-dns
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
dns-default ClusterIP 172.30.0.10 <none>        53/UDP,53/TCP,9154/TCP 8d

jegan@tektutor.org ➔
```

## Lab - Creating an external NodePort service for nginx deployment

We need to first delete the clusterip service we created earlier

```
oc get svc
oc delete svc/nginx
oc get svc
```

Let's create the nodeport service

```
oc expose deploy/nginx --type=NodePort --port=8080
oc get svc
oc describe svc/nginx
```

As this is an external service, we can directly access using the node name and node port

```
oc get nodes
curl http://master-1.ocp4.tektutor.org.labs:32326
curl http://master-2.ocp4.tektutor.org.labs:32326
curl http://master-3.ocp4.tektutor.org.labs:32326
curl http://worker-1.ocp4.tektutor.org.labs:32326
curl http://worker-2.ocp4.tektutor.org.labs:32326
```

## Expected output

```
jegan@tektutor.org ➤ oc get svc
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)      AGE
nginx    NodePort  172.30.72.59 <none>        8080:32326/TCP  10s

jegan@tektutor.org ➤ oc get nodes
NAME           STATUS  ROLES      AGE   VERSION
master-1.ocp4.tektutor.org.labs  Ready   control-plane,master,worker  8d   v1.28.10+a2c84a5
master-2.ocp4.tektutor.org.labs  Ready   control-plane,master,worker  8d   v1.28.10+a2c84a5
master-3.ocp4.tektutor.org.labs  Ready   control-plane,master,worker  8d   v1.28.10+a2c84a5
worker-1.ocp4.tektutor.org.labs  Ready   worker      8d   v1.28.10+a2c84a5
worker-2.ocp4.tektutor.org.labs  Ready   worker      8d   v1.28.10+a2c84a5

jegan@tektutor.org ➤ curl http://master-1.ocp4.tektutor.org.labs:32326
!DOCTYPE html
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```
jegan@tektutor.org ➤ curl http://master-2.ocp4.tektutor.org.labs:32326
!DOCTYPE html
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```
jegan@tektutor.org ➔ curl http://master-3.ocp4.tektutor.org.labs:32326
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 333
Date: Mon, 02 Jul 2024 15:12:00 GMT
Connection: keep-alive
Keep-Alive: timeout=5

<p>For online documentation and support please refer to <a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at <a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
jegan@tektutor.org ➔ curl http://master-3.ocp4.tektutor.org.labs:32326
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to <a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at <a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
jegan@tektutor.org ➔
```

```
jegan@tektutor.org ➔ curl http://worker-1.ocp4.tektutor.org.labs:32326
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 333
Date: Mon, 02 Jul 2024 15:12:00 GMT
Connection: keep-alive
Keep-Alive: timeout=5

<p>For online documentation and support please refer to <a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at <a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
jegan@tektutor.org ➔ curl http://worker-1.ocp4.tektutor.org.labs:32326
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to <a href="http://nginx.org/">nginx.org</a>. <br/>
Commercial support is available at <a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
jegan@tektutor.org ➔
```

## Info - NodePort Service gotchas

- NodePort is neither user friendly nor developer friendly
  - the end-user must know the openshift node details like its hostname, node IP, how many nodes are there in the cluster
  - for nodeport service we create one port in the range 30000-32767 will be opened on every node in the cluster
  - the more nodeport services we create, more number of ports has to be opened up in the firewall
  - opening too many ports leads to security issue
  - Just ClusterIP and LoadBalancer service, NodePort service is a Kubernetes feature
  - Openshift has a better alternate called Route

Lab - Creating a LoadBalancer external service for nginx deployment

For detailed instructions, you may refer my medium article

<https://medium.com/tektutor/using-metallb-loadbalancer-with-bare-metal-openshift-onprem-4230944bfa35>

## Points to remember

- Load balancer will not work out of the box in a bare-metal(on-prem) openshift setup
  - We need install metallb operator as an Administrator

- Once installed we need configure the address-pool i.e range of IP addresses we need allocate so that when we create lb service, the metall controller will be able to assign an IP from the address-pool we reserved
  - We also need to create a metallb controller instance which will monitor for loadbalancer services created in the cluster

Let's delete the nginx service we created earlier!

```
oc get svc  
oc delete svc/nginx  
oc get svc
```

## Let's create the loadbalancer service

```
oc get deploy  
oc expose deploy/nginx --type=LoadBalancer --port=8080  
oc get svc  
oc describe svc/nginx
```

## Accessing the loadbalancer external service

```
curl http://192.168.122.90:8080
```

## Expected output

```
jegan@tektutor.org ~ oc get svc
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
nginx    NodePort    172.30.72.59   <none>          8080:32326/TCP  17m
jegan@tektutor.org ~ oc delete svc/nginx
service "nginx" deleted
jegan@tektutor.org ~ oc expose deploy/nginx --type=LoadBalancer --port=8080
service/nginx exposed
jegan@tektutor.org ~ oc get svc
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
nginx    LoadBalancer 172.30.242.202  192.168.122.90  8080:31806/TCP  2s
jegan@tektutor.org ~ oc describe svc/nginx
Name:           nginx
Namespace:      jegan
Labels:         app=nginx
Annotations:   metallb.universe.tf/ip-allocated-from-pool: tektutor-metallb-addresspool
Selector:       app=nginx
Type:          LoadBalancer
IP Family Policy: SingleStack
IP Families:   IPv4
IP:             172.30.242.202
IPs:            172.30.242.202
LoadBalancer Ingress: 192.168.122.90
Port:           <unset> 8080/TCP
TargetPort:     8080/TCP
NodePort:       <unset> 31806/TCP
Endpoints:     10.129.0.21:8080,10.129.0.6:8080,10.130.0.53:8080
Session Affinity: None
External Traffic Policy: Cluster
Events:
  Type      Reason     Age      From            Message
  ----      ----     ----      ----            -----
  Normal    IPAllocated 11s      metallb-controller  Assigned IP ["192.168.122.90"]
  Normal    nodeAssigned 11s      metallb-speaker   announcing from node "worker-2.ocp4.tektutor.org.labs" with protocol "layer2"
jegan@tektutor.org ~ curl http://192.168.122.90:8080
<!DOCTYPE html>
<html>
```

The screenshot shows a terminal window with four tabs, all titled 'jegan@tektutor.org'. The current tab displays the configuration of a service named 'nginx'. The configuration includes:

- TargetPort: 8080/TCP
- NodePort: <unset> 31806/TCP
- Endpoints: 10.129.0.21:8080, 10.129.0.6:8080, 10.130.0.53:8080
- Session Affinity: None
- External Traffic Policy: Cluster

Events section:

Type	Reason	Age	From	Message
Normal	IPAllocated	11s	metallb-controller	Assigned IP ["192.168.122.90"]
Normal	nodeAssigned	11s	metallb-speaker	announcing from node "worker-2.ocp4.tektutor.org.labs" with protocol "layer2"

Terminal command:

```
jegan@tektutor.org ➔ curl http://192.168.122.90:8080
```

Output:

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```
jegan@tektutor.org ➔
```

## Lab - Exposing service to external access using Openshift route

```
oc new-project jegan
oc create deployment nginx --image=bitnami/nginx:1.18 --replicas=3
oc get po
oc expose deploy/nginx --port=8080
oc get svc
oc describe svc/nginx
```

Now, let's create http route

```
oc expose svc/hello
oc get route
curl http://http://nginx-jegan.apps.ocp4.tektutor.org.labs
```

## Expected output

```
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ oc create deployment nginx --image=bitnami/nginx:1.18 --replicas=3
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ oc get po
NAME          READY   STATUS    RESTARTS   AGE
nginx-4dkjv   1/1     Running   0          36m
nginx-566b5879cb-749sw   1/1     Running   0          2m10s
nginx-566b5879cb-gwdg5   1/1     Running   0          2m10s
nginx-566b5879cb-xg8wv   1/1     Running   0          2m10s
nginx-9gh7h   1/1     Running   0          36m
nginx-fd99g   1/1     Running   0          36m
nginx-nrk6m   1/1     Running   0          36m
nginx-tf65d   1/1     Running   0          36m
test-544c5f8667-f86n5   1/1     Running   0          8m29s
test-544c5f8667-md499   1/1     Running   0          8m27s
test-544c5f8667-qcj1k   1/1     Running   0          151m
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ oc expose deploy/nginx --port=8080
service/nginx exposed
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ oc get svc
NAME        CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
nginx       172.30.132.131  <none>           8080/TCP    4s
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ oc expose svc/nginx
route/nginx exposed
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ oc get route
NAME          HOST/PORT      PATH  SERVICES  PORT  TERMINATION  WILDCARD
nginx         nginx-jegan.apps.ocp4.tektutor.org.labs      nginx      8080      None
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx!</em></p>
</body>
</html>
jegan@tektutor.org ~ -/openshift-july-2024/Day2/daemonset [?] main ➤ curl http://nginx-jegan.apps.ocp4.tektutor.org.labs

```

## Info - What is Kubernetes/OpenShift Ingress?

- Ingress is not a service
- Ingress is a set of forwarding rules
- The Ingress Controller picks the rules defined in the Ingress and it configures Load balancer either HAProxy or Nginx
- the 2 most popular ingress controllers are
  1. Nginx Ingress Controller
  2. HAProxy Ingress Controller

## Lab - Creating an ingress to forward calls to two different services based on path based routing

You can find your openshift base domain using the below command

```
oc describe ingresscontroller/default -n openshift-ingress-operator | grep
Domain:
```

As per the output you get for the above command, you need to edit the host url in the ingress.yml.

You need to edit the ingress.yml and replace the host url from tektutor.apps.ocp4.tektutor.org.labs to tektutor.apps.ocp4.rps.com

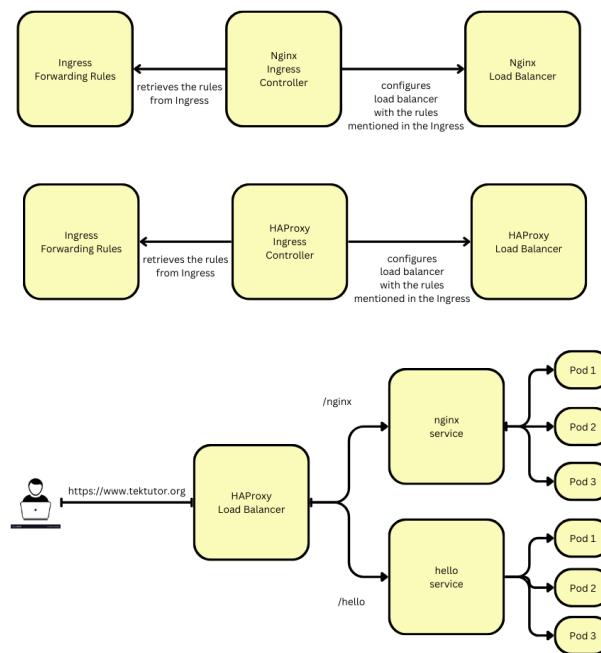
```
cd ~/openshift-july-2024
git pull
```

```
cd Day2/ingress
oc new-project jegan
oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
oc create deployment hello --image=tektutor/spring-ms:1.0 --replicas=3

oc expose deploy/nginx --port=8080
oc expose deploy/hello --port=8080

cat ingress.yml
oc apply -f ingress.yml
oc get ingress
oc describe ingress/tektutor
curl http://tektutor.apps.ocp4.tektutor.org.labs/nginx
curl http://tektutor.apps.ocp4.tektutor.org.labs/hello
```

## Expected output



```
Activities Terminal Jul 2 17:34 jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > oc new-project jegan
Already on project "jegan" on server "https://api.ocp4.tektutor.org.labs:6443".
You can add applications to this project with the 'new-app' command. For example, try:
  oc new-app rails-postgresql-example
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
  kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.43 -- /agnhost serve-hostname
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > oc create deployment nginx --image=bitnami/nginx:latest --replicas=3
deployment.apps/nginx created
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > oc create deployment hello --image=tektutor/spring-ms:1.0 --replicas=3
deployment.apps/hello created
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > oc get deploy
NAME READY UP-TO-DATE AVAILABLE AGE
hello 3/3 3 3 3s
nginx 3/3 3 3 23s
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > oc expose deployment/nginx --port=8080
service/nginx exposed
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > oc expose deployment/hello --port=8080
service/hello exposed
jegan@tektutor.org > ~/openshift-july-2024/Day2 [?] main > cd ingress
jegan@tektutor.org > ~/openshift-july-2024/Day2/ingress [?] main > vim ingress.yml
jegan@tektutor.org > ~/openshift-july-2024/Day2/ingress [?] main > cat ingress.yml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tektutor
  annotations:
    haproxy.router.openshift.io/rewrite-target: /
spec:
  rules:
  - host: www.tektutor.org
    http:
      paths:
        - path: /nginx
          pathType: Prefix
        - path: /hello
          pathType: Prefix
# Added by CRC
192.168.130.11  api.crc.testing canary-openshift-ingress-canary.apps-crc.testing console-openshift-console.apps-crc.testing default-route-openshift-image-registry.apps-crc.testing downloads-openshift-console.apps-crc.testing oauth-openshift.apps-crc.testing
# End of CRC section
jegan@tektutor.org > ~/openshift-july-2024/Day2/ingress [?] main > vim ingress.yml
jegan@tektutor.org > ~/openshift-july-2024/Day2/ingress [?] main > cat ingress.yml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: tektutor
  annotations:
    haproxy.router.openshift.io/rewrite-target: /
spec:
  rules:
  - host: tektutor.apps.ocp4.tektutor.org.labs
    http:
      paths:
        - backend:
            service:
              name: nginx
              port:
                number: 8080
            path: /nginx
            pathType: Prefix
        - backend:
            service:
              name: hello
              port:
                number: 8080
            path: /hello
            pathType: Prefix
jegan@tektutor.org > ~/openshift-july-2024/Day2/ingress [?] main >
```

```

Activities Terminal Jul 2 17:36 jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org
jegan@tektutor.org >~/openshift-july-2024/Day2/ingress ⌘ main ➔ oc apply -f ingress.yml
ingress.networking.k8s.io/tektutor configured
jegan@tektutor.org >~/openshift-july-2024/Day2/ingress ⌘ main ➔ oc get ingress
NAME CLASS HOSTS ADDRESS PORTS AGE
tektutor <none> tektutor.apps.ocp4.tektutor.org.labs router-default.apps.ocp4.tektutor.org.labs 80 86s
jegan@tektutor.org >~/openshift-july-2024/Day2/ingress ⌘ main ➔ oc describe ingress/tektutor
Name: tektutor
Labels: <none>
Namespace: jegan
Address: router-default.apps.ocp4.tektutor.org.labs
Ingress Class: <none>
Default backend: <default>
Rules:
Host Path Backends
--- -----
tektutor.apps.ocp4.tektutor.org.labs /nginx nginx:8080 (10.128.2.27:8080,10.129.0.90:8080,10.131.0.72:8080)
/hello hello:8080 (10.128.2.28:8080,10.130.0.118:8080,10.131.0.73:8080)
Annotations: haproxy.router.openshift.io/rewrite-target: /
Events: <none>
jegan@tektutor.org >~/openshift-july-2024/Day2/ingress ⌘ main ➔ curl http://tektutor.apps.ocp4.tektutor.org.nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

```

```

Activities Terminal Jul 2 17:36 jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org jegan@tektutor.org
jegan@tektutor.org >~/openshift-july-2024/Day2/ingress ⌘ main ➔ curl http://tektutor.apps.ocp4.tektutor.org.nginx
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>

```

Greetings from Spring Boot!

```

jegan@tektutor.org >~/openshift-july-2024/Day2/ingress ⌘ main ➔ curl http://tektutor.apps.ocp4.tektutor.org.labs/hello

```

The screenshot shows a terminal window with four tabs, all titled 'jegan@tektutor.org'. The tabs are arranged horizontally at the top of the window.

The first tab contains the command:

```
jegan@tektutor.org ~ -/openshift-july-2024 [main] oc get nodes
```

The output of this command is:

NAME	STATUS	ROLES	AGE	VERSION
master-1.ocp4.tektutor.org.labs	Ready	control-plane,master,worker	8d	v1.28.10+a2c84a5
master-2.ocp4.tektutor.org.labs	Ready	control-plane,master,worker	8d	v1.28.10+a2c84a5
master-3.ocp4.tektutor.org.labs	Ready	control-plane,master,worker	8d	v1.28.10+a2c84a5
worker-1.ocp4.tektutor.org.labs	Ready	worker	8d	v1.28.10+a2c84a5
worker-2.ocp4.tektutor.org.labs	Ready	worker	8d	v1.28.10+a2c84a5

The second tab contains the command:

```
jegan@tektutor.org ~ -/openshift-july-2024 [main] oc describe ingresscontroller/default -n openshift-ingress-operator | grep Domain
```

The output of this command is:

```
:  
Domain: apps.ocp4.tektutor.org.labs
```

The third tab contains the command:

```
jegan@tektutor.org ~ -/openshift-july-2024 [main]
```