

1. Use the previous deployment
2. Create a service of type NodePort for NGINX deployment
3. Check the NodePort service on a browser to verify

vi nginx-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

13, 21 All

kubectl apply -f nginx-service.yaml

```
[ec2-user@ip-172-31-93-114 ~]$ kubectl apply -f nginx-service.yaml
service/nginx-service created
[ec2-user@ip-172-31-93-114 ~]$
```

kubectl get services nginx-service

The screenshot shows the AWS Management Console for the us-east-1 region. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays the 'Instances (1/3)' page with a search bar and filters. A filter for 'Instance state = running' is applied. The table lists three instances: kub-master (I-042ec85885260c244), kub-slave1 (I-0e85835b18a623c9c), and kub-slave2 (I-07692e6a4f9f7b12f). All instances are in the 'Running' state. The details for 'Instance: i-0e85835b18a623c9c (kub-slave1)' are shown, including the public IPv4 address 44.201.153.44 and private IPv4 address 172.31.91.218.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
kub-master	I-042ec85885260c244	Running	t2.medium	2/2 checks passed	View alarms
kub-slave1	I-0e85835b18a623c9c	Running	t2.medium	2/2 checks passed	View alarms
kub-slave2	I-07692e6a4f9f7b12f	Running	t2.medium	2/2 checks passed	View alarms

Instance: i-0e85835b18a623c9c (kub-slave1)

Details	Status and alarms	Monitoring	Security	Networking	Storage	Tags
<p>Instance summary</p> <p>Instance ID: I-0e85835b18a623c9c (kub-slave1)</p> <p>Public IPv4 address: 44.201.153.44 [open address]</p> <p>Private IPv4 addresses: 172.31.91.218</p>						

public ip of slave1:30080 on url

The screenshot shows a web browser window with the address bar displaying '44.201.153.44:30080'. The page content is as follows:

## 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](https://nginx.org). Commercial support is available at [nginx.com](https://nginx.com).

Thank you for using nginx.

slave 2

The screenshot shows a web browser window with the address bar displaying '44.211.208.212:30080'. The page content is as follows:

## 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](https://nginx.org). Commercial support is available at [nginx.com](https://nginx.com).

Thank you for using nginx.

After applying the manifest, you can get the NodePort assigned to the service by running `kubectl get services nginx-service`. Clients can access the Nginx service using any of the Kubernetes nodes' IP addresses and the NodePort.

