Kubernetes Service's

ClusterIP

Exposes the service on a cluster-internal IP. Choosing this value makes the service only reachable from within the cluster. This is the default ServiceType. Only Pods within the K8 Cluster can communicate using this service. You cannot access directly from the browser using Service IP.

Deplyment yaml file

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

Create a Deployment based on YAML file

```
ubuntu@master:~$ kubectl get deploy -A
NAMESPACE      NAME               READY   UP-TO-DATE   AVAILABLE   AGE
default        nginx-deployment   3/3     3            3           3d2h
developement   nginx-deployment   3/3     3            3           10h
kube-system    coredns            2/2     2            2           4d5h
ubuntu@master:~$ kubectl delete deploy nginx-deployment
deployment.apps "nginx-deployment" deleted
ubuntu@master:~$ kubectl apply -f nginxdepleyement.yaml
deployment.apps/nginx-deployment unchanged
ubuntu@master:~$ kubectl get po -n developement
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-7848d4b86f-8qclv   1/1     Running   1          10h
nginx-deployment-7848d4b86f-np7d7   1/1     Running   1          10h
nginx-deployment-7848d4b86f-q5qhl   1/1     Running   1          10h
ubuntu@master:~$
```

Display information about all deployments

```
ubuntu@master:~$ kubectl get deployments -n developement
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           10h
ubuntu@master:~$
```

Display information about Pods with labels

```
ubuntu@master:~$ kubectl get pods --show-labels  -n developement
NAME                                READY   STATUS    RESTARTS   AGE   LABELS
nginx-deployment-7848d4b86f-8qclv   1/1     Running   1          10h   app=nginx,pod-template-hash=7848d4b86f
nginx-deployment-7848d4b86f-np7d7   1/1     Running   1          10h   app=nginx,pod-template-hash=7848d4b86f
nginx-deployment-7848d4b86f-q5qhl   1/1     Running   1          10h   app=nginx,pod-template-hash=7848d4b86f
ubuntu@master:~$
```

Create ClusterIP Service using YAML manifest

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: developement
spec:
  type: ClusterIP
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
```

Create a Service using YAML file

```
ubuntu@master:~$ cat nginxservice.yml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
  - port: 8080
    targetPort: 80
ubuntu@master:~$ kubectl apply -f nginxservice.yaml
service/nginx-service unchanged
ubuntu@master:~$ kubectl get svc -n developement
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
nginx-service   ClusterIP   10.103.134.147   <none>        8080/TCP   10h
ubuntu@master:~$
```

Display information about Services

```
ubuntu@master:~$ kubectl get services -n developement
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
nginx-service   ClusterIP   10.103.134.147   <none>        8080/TCP   10h
ubuntu@master:~$
```

K8 Endpoint object will be created automatically when the Service is created with same service name. It will track all the Pod IPs where label as app:nginx

```
ubuntu@master:~$ kubectl get endpoints -n developement
NAME            ENDPOINTS                                      AGE
nginx-service   10.36.0.3:80,10.44.0.2:80,10.44.0.3:80         11h
ubuntu@master:~$
```

By default, Kubernetes Service IP address (i.e. Cluster-IP) and Port will be accessed within the cluster only. So, enter into one of the nginx Pod and make curl using service IP address/name and Port to view nginx webpage

Enter into nginx Pod shell

```
ubuntu@master:~$ kubectl get services
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP    4d6h
nginx-service   ClusterIP   10.106.66.83    <none>        8080/TCP   7m40s
ubuntu@master:~$ kubectl get endpoints
NAME            ENDPOINTS                     AGE
kubernetes      172.31.1.216:6443             4d6h
nginx-service   10.36.0.1:80,10.44.0.1:80     7m51s
ubuntu@master:~$ kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-66b6c48dd5-s2lll   1/1     Running   0          9m18s
nginx-deployment-66b6c48dd5-w85wk   1/1     Running   0          9m18s
ubuntu@master:~$ kubectl exec -it nginx-deployment-66b6c48dd5-s2lll -- /bin/bash
root@nginx-deployment-66b6c48dd5-s2lll:/#
```

```
root@nginx-deployment-66b6c48dd5-s2lll:/# curl -v -k http://10.106.66.83:8080 | grep -w "200"
* Rebuilt URL to: http://10.106.66.83:8080/
*   Trying 10.106.66.83...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to 10.106.66.83 (10.106.66.83) port 8080 (#0)
> GET / HTTP/1.1
> Host: 10.106.66.83:8080
> User-Agent: curl/7.52.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: nginx/1.14.2
< Date: Thu, 09 Dec 2021 14:45:34 GMT
< Content-Type: text/html
< Content-Length: 612
< Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT
< Connection: keep-alive
< ETag: "5c0692e1-264"
< Accept-Ranges: bytes
<
{ [612 bytes data]
* Curl_http_done: called premature == 0
100   612  100   612    0     0   338k      0 --:--:-- --:--:-- --:--:--  597k
* Connection #0 to host 10.106.66.83 left intact
root@nginx-deployment-66b6c48dd5-s2lll:/#
```

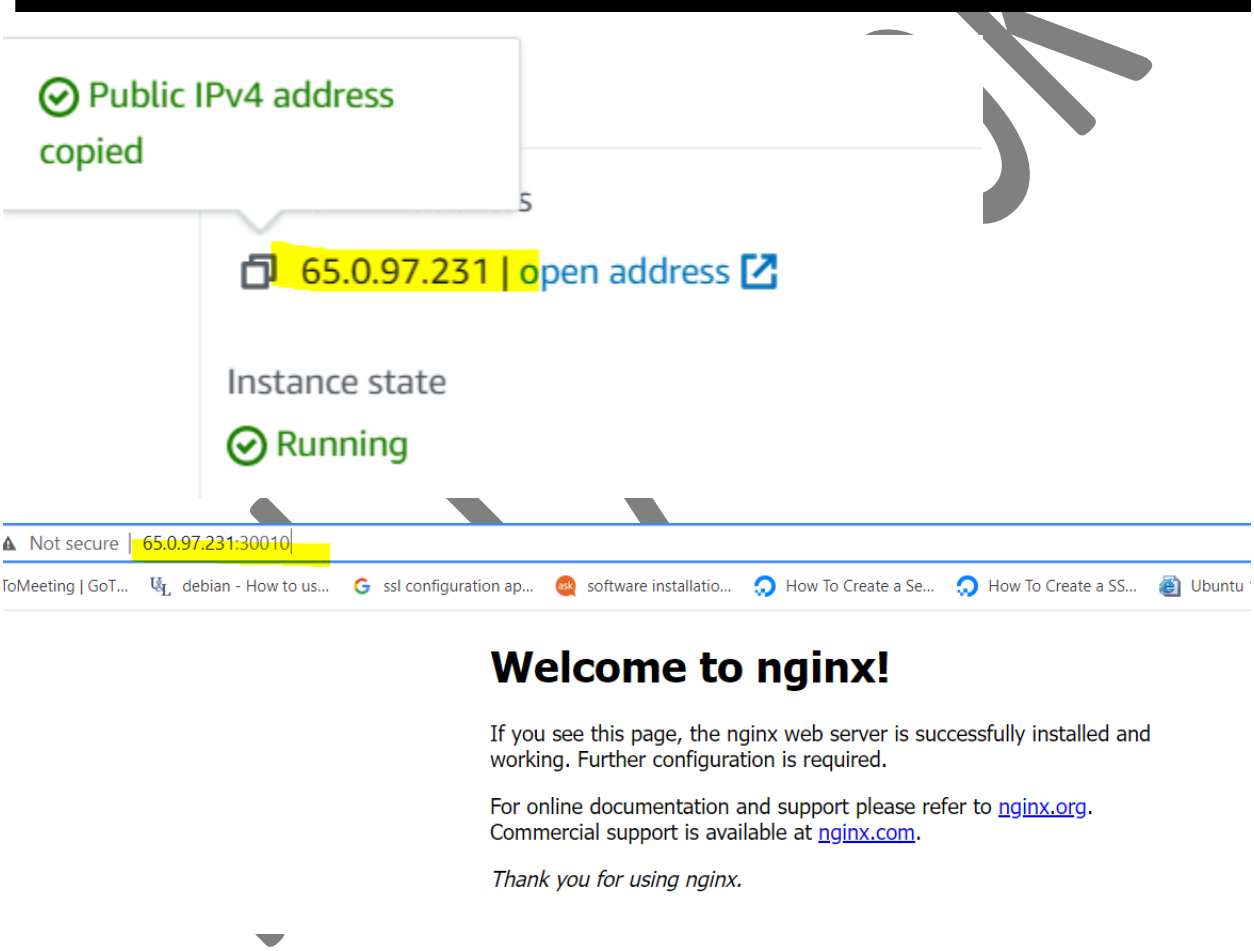

NodePort

A NodePort service is the most primitive way to get external traffic directly to your service. NodePort, as the name implies, opens a specific port on all the Nodes (the VMs), and any traffic that is sent to this port is forwarded to the service.

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - protocol: TCP
```

```
    port: 8080
    targetPort: 80
    nodePort: 30010
```

```
ubuntu@master:~$ vi nodeportservice.yaml
ubuntu@master:~$ kubectl apply -f nodeportservice.yaml
service/nginx-service configured
ubuntu@master:~$ kubectl get services
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP   10.96.0.1      <none>        443/TCP          4d6h
nginx-service   NodePort    10.106.66.83   <none>        8080:30010/TCP   15m
ubuntu@master:~$
```



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.

LoadBalancer

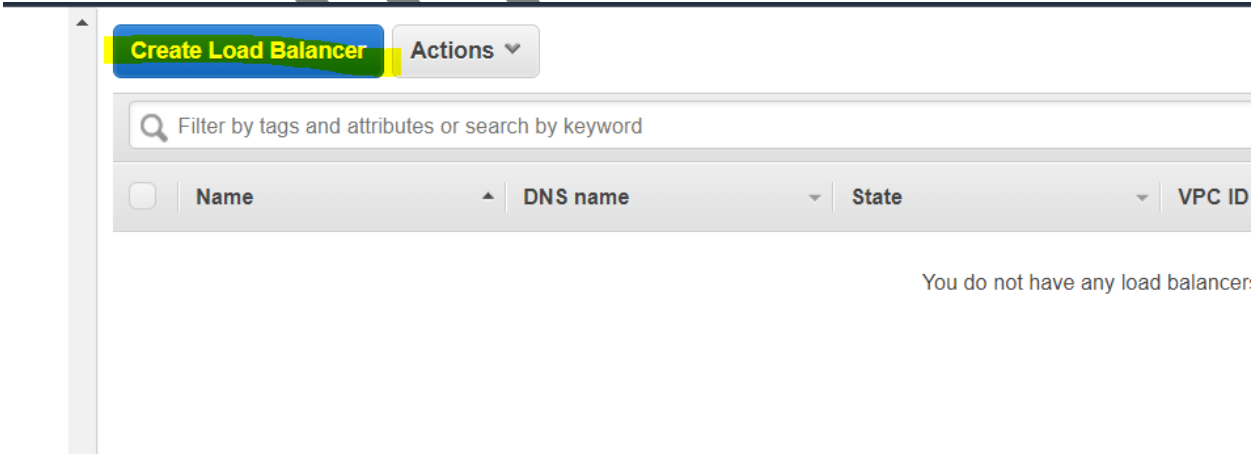
A LoadBalancer service is the standard way to expose a service to the internet. On aws, this will spin up a Network Load Balancer that will give you a single IP address that will forward all traffic to your service.

Create LoadBalancer Service using YAML manifest

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  type: LoadBalancer
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 8081
      targetPort: 80
      nodePort: 30010
```

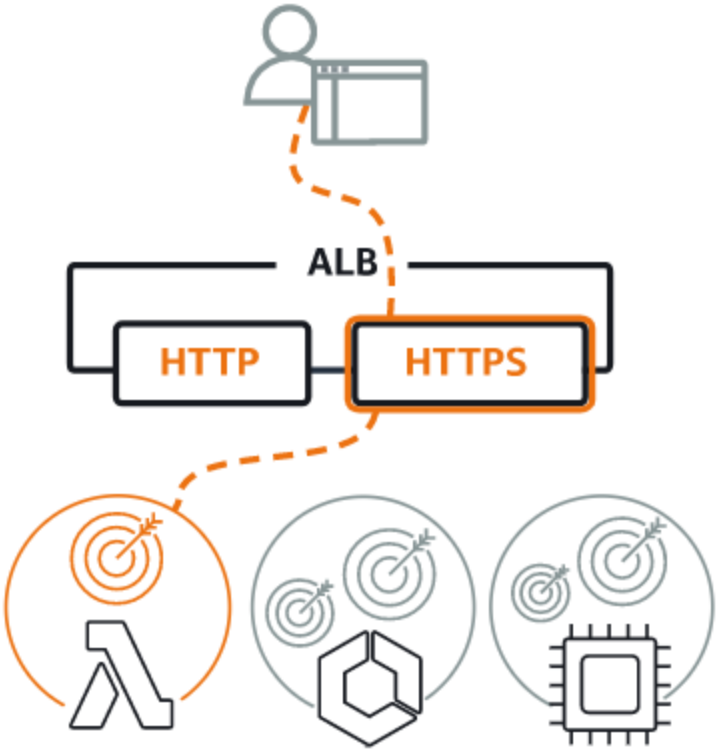
```
ubuntu@master:~$ kubectl get svc
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.96.0.1       <none>        443/TCP          4d6h
nginx-service   LoadBalancer   10.106.66.83    <pending>     8080:30010/TCP   25m
ubuntu@master:~$
```

Create a load balancer on aws



Create a application load balancer

Application Load Balancer Info



Choose an Application Load Balancer when

Go to create

How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

test-loadbalncer-svc

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme Info

Scheme cannot be changed after the load balancer is created.

- (•) Internet-facing
  An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
- ( ) Internal
  An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type Info

Select the type of IP addresses that your subnets use.

IP address type Info

Select the type of IP addresses that your subnets use.

- (•) IPv4
  Recommended for internal load balancers.
- ( ) Dualstack
  Includes IPv4 and IPv6 addresses.

Network mapping Info

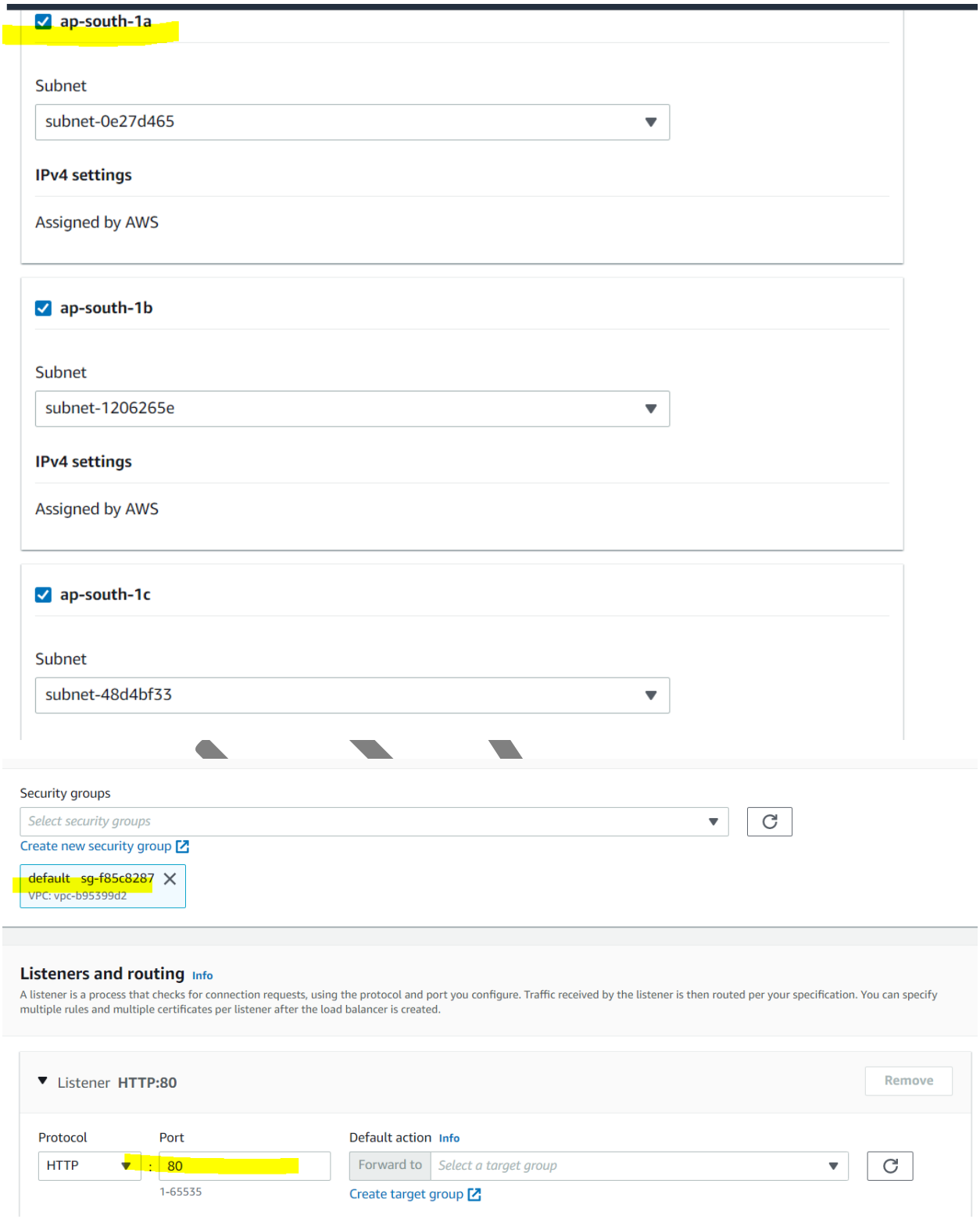
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is create
confirm the VPC for your targets, view your target groups.

-
vpc-b95399d2
IPv4: 172.31.0.0/16

Mappings Info


ap-south-1a
Subnet
subnet-0e27d465
IPv4 settings
Assigned by AWS
ap-south-1b
Subnet
subnet-1206265e
IPv4 settings
Assigned by AWS
ap-south-1c
Subnet
subnet-48d4bf33
Security groups
Select security groups
Create new security group
default sg-f85c8287
VPC: vpc-b95399d2
Listeners and routing Info
A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.
Listener HTTP:80
Remove
Protocol
Port
Default action Info
HTTP
80
1-65535
Forward to
Select a target group
Create target group

Choose a target type

- (•) **Instances**
  - Supports load balancing to instances within a specific VPC.
- ( ) **IP addresses**
  - Supports load balancing to VPC and on-premises resources.
  - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
  - Offers flexibility with microservice based architectures, simplifying inter-application communication.
- ( ) **Lambda function**
  - Facilitates routing to a single Lambda function.
  - Accessible to Application Load Balancers only.
- ( ) **Application Load Balancer**
  - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
  - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

test-targetgroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol Port

Target group name

test-targetgroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol Port

HTTP : 30010

VPC

Select the VPC with the instances that you want to include in the target group.

-
vpc-b95399d2
IPv4: 172.31.0.0/16

Protocol version

HTTP1

Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/3)

Filter resources by property or value

1

Instance ID	Name	State	Security groups	Zone	Subnet ID
i-08dddb88571a05360	kube master	running	kube	ap-south-1b	subnet-1206265e
i-09f37834dfe8eb92b	kuber worker	running	kube1	ap-south-1b	subnet-1206265e
i-0d4b9dd624142a288	kuber worker 1	running	Resize EBS	ap-south-1b	subnet-1206265e

2 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

30010

1-65535 (separate multiple ports with commas)

```
ubuntu@master:~$ kubectl get svc
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.96.0.1       <none>        443/TCP          4d6h
nginx-service   LoadBalancer   10.106.66.83    <pending>     8080:30010/TCP   25m
ubuntu@master:~$
```

Create Load Balancer | Actions

search : test | Add filter

1 to

Name	DNS name	State	VPC ID	Availability Zones	Type
test	test-1923761215.ap-south-1....	Active	vpc-b95399d2	ap-south-1b, ap-south-1c	application

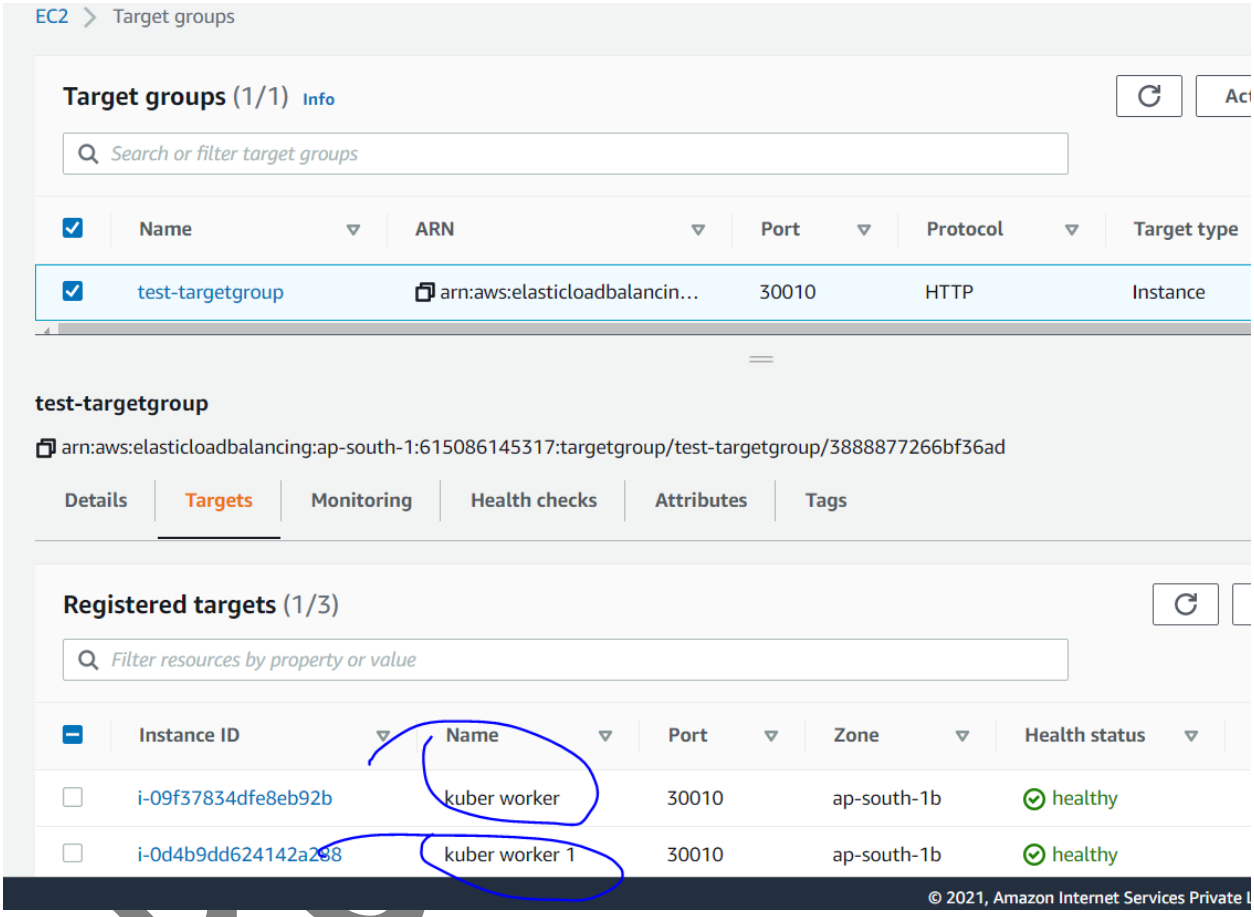
Load balancer: test

Description | Listeners | Monitoring | Integrated services | Tags

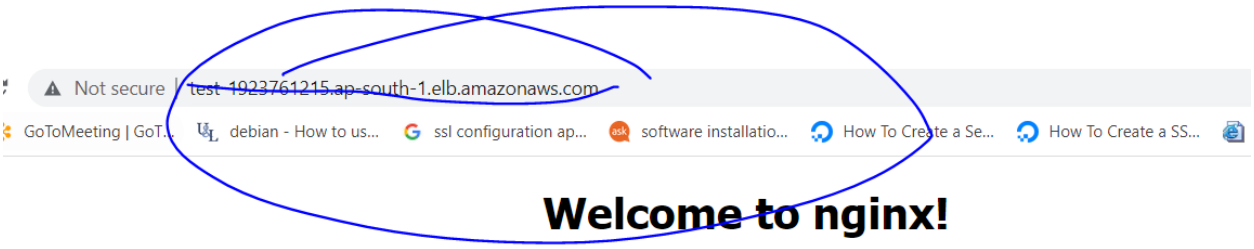
Basic Configuration

Name	test
ARN	arn:aws:elasticloadbalancing:ap-south-1:615086145317:loadbalancer/app/test/fe0929f0dbe784ee
DNS name	test-1923761215.ap-south-1.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	internet-facing
IP address type	ipv4

```
Last login: Thu Dec  9 15:42:29 2021 from 49.37.219.198
ubuntu@master:~$ kubectl get svc
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.96.0.1       <none>        443/TCP          4d7h
nginx-service   LoadBalancer   10.106.66.83    <pending>     8080:30010/TCP   88m
ubuntu@master:~$
```




```
ubuntu@master:~$ kubectl get nodes
NAME      STATUS   ROLES                  AGE    VERSION
master    Ready    control-plane,master   4d7h   v1.21.0
worker    Ready    <none>                 3d5h   v1.21.0
worker1   Ready    <none>                 3d5h   v1.21.0
ubuntu@master:~$
```



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.