

Imagine that, using nodePort service type
you exposed your web app to outside world on the internet

which node IP and nodePort will you provide to end users?



LoadBalancer Service

Concept

Objectives

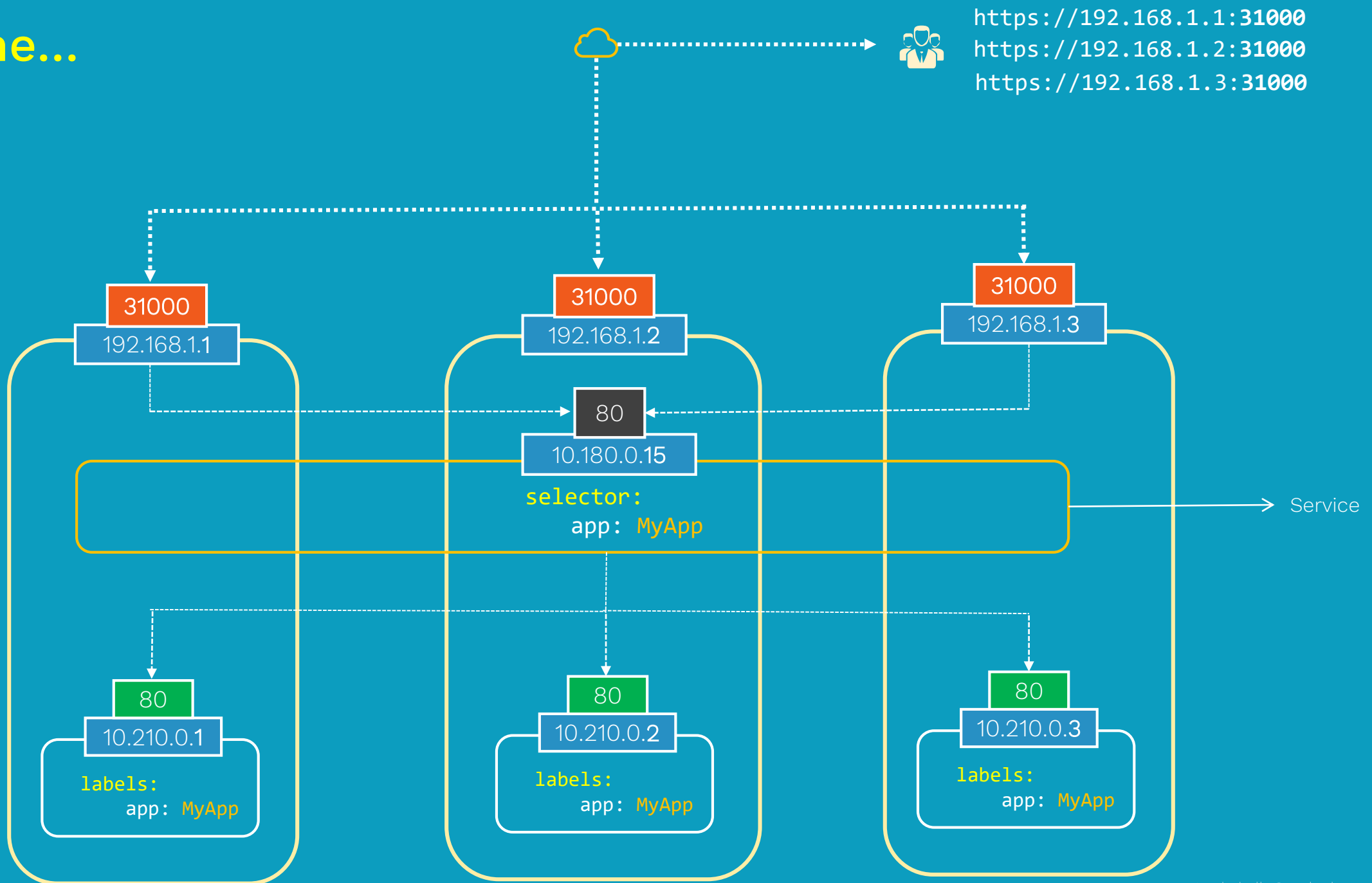
Concept

- a. Why we need?
- b. LoadBalancer - Overview

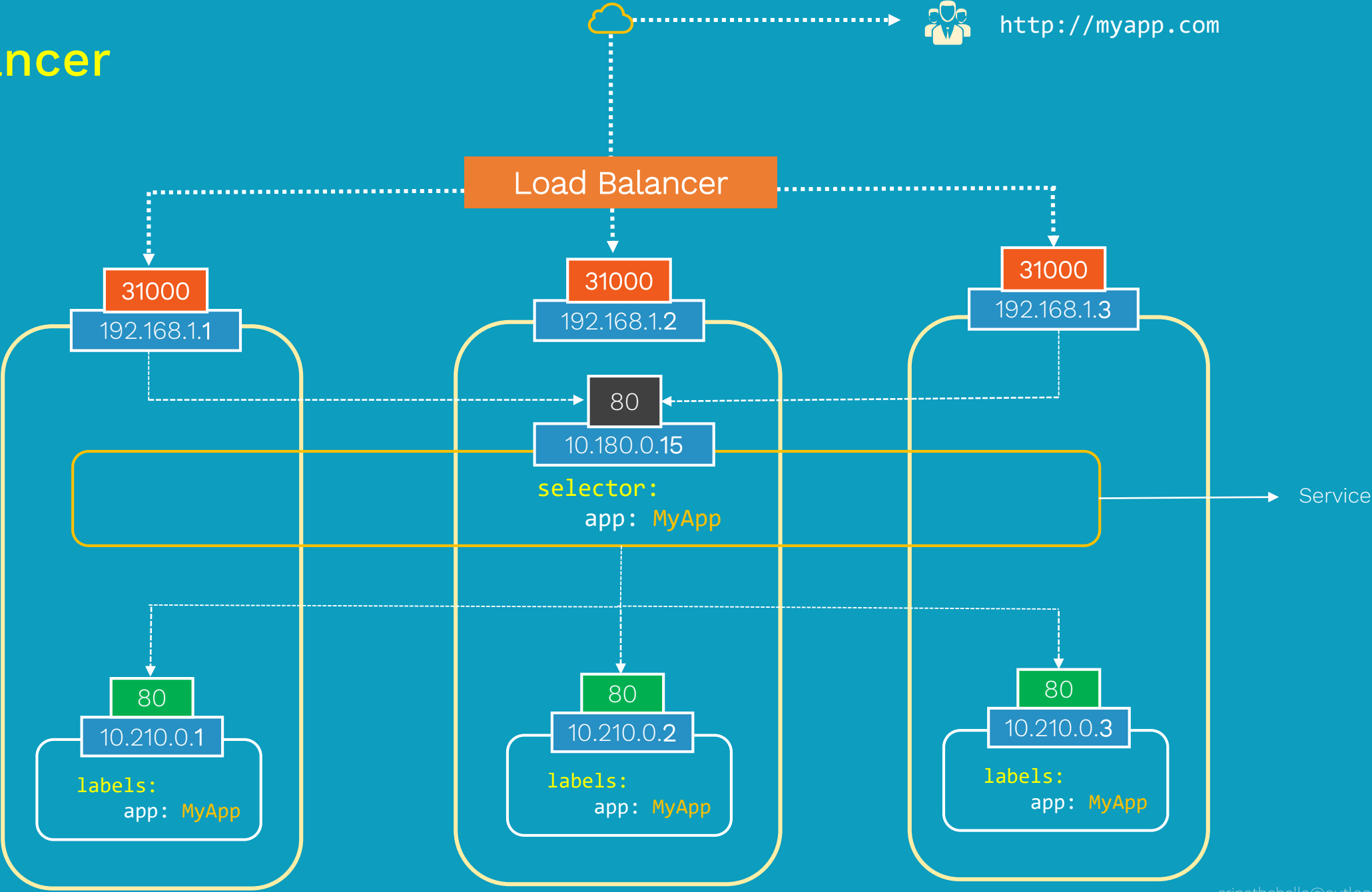
Review Demo

- a. Manifest file
- b. Create and display
- c. Test use cases
- d. Clean up

Think time...



LoadBalancer



Review Demo

- a. Manifest file
- b. Create objects



- d. Display & Validate
- e. Clean up

LoadBalancer – Config

```
# Service - LoadBalancer
# nginx-service -lb.yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: nginx-app
spec:
  selector:
    app: nginx-app
  type: LoadBalancer
  ports:
  - nodePort: 31000
    port: 80
    targetPort: 80
```

```
# kubectl expose deploy nginx-deployment --name=nginx-
service --port=80 --target-port=80 --type=LoadBalancer
```

```
# Deployment
# controllers/nginx-deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-app
  template:
    metadata:
      labels:
        app: nginx-app
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```

LoadBalancer – Create & Display

```
srinath@master:~ $ kubectl create -f nginx-deploy.yaml  
deployment.apps/nginx-deployment created
```

```
srinath@master:~ $ kubectl create -f nginx-service -lb.yaml  
service/my-service created
```

```
srinath@master:~ $ kubectl get service -l app=nginx-app
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
my-service	LoadBalancer	10.11.241.216	<pending>	80:31000/TCP	26s

LoadBalancer – LoadBalancer - Display

```
srinath@master:~ $ kubectl get service -l app=nginx-app
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
my-service	LoadBalancer	10.11.241.216	104.198.43.137	80:31000/TCP	26s

Google Cloud Platform

My First Project

Services

REFRESH

Kubernetes services

Brokered services

BETA

Services are sets of pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to services.

Is system object : False

Filter resources

Name ^	Status	Service Type	Endpoints	Pods	Namespace	Cluster
my-service	Ok	Load balancer	104.198.43.137:80	1 / 1	default	cluster-1

LoadBalancer – LoadBalancer - Display

```
srinath@master:~ $ kubectl describe service my-service
```

```
Name: my-service
Namespace: default
Labels: app=nginx-app
Annotations: <none>
Selector: app=nginx-app
Type: LoadBalancer
IP: 10.11.241.216
LoadBalancer Ingress: 104.198.43.137
Port: <unset> 80/TCP
TargetPort: 80/TCP
NodePort: <unset> 31000/TCP
Endpoints: 10.8.0.12:80
Session Affinity: None
External Traffic Policy: Cluster
```

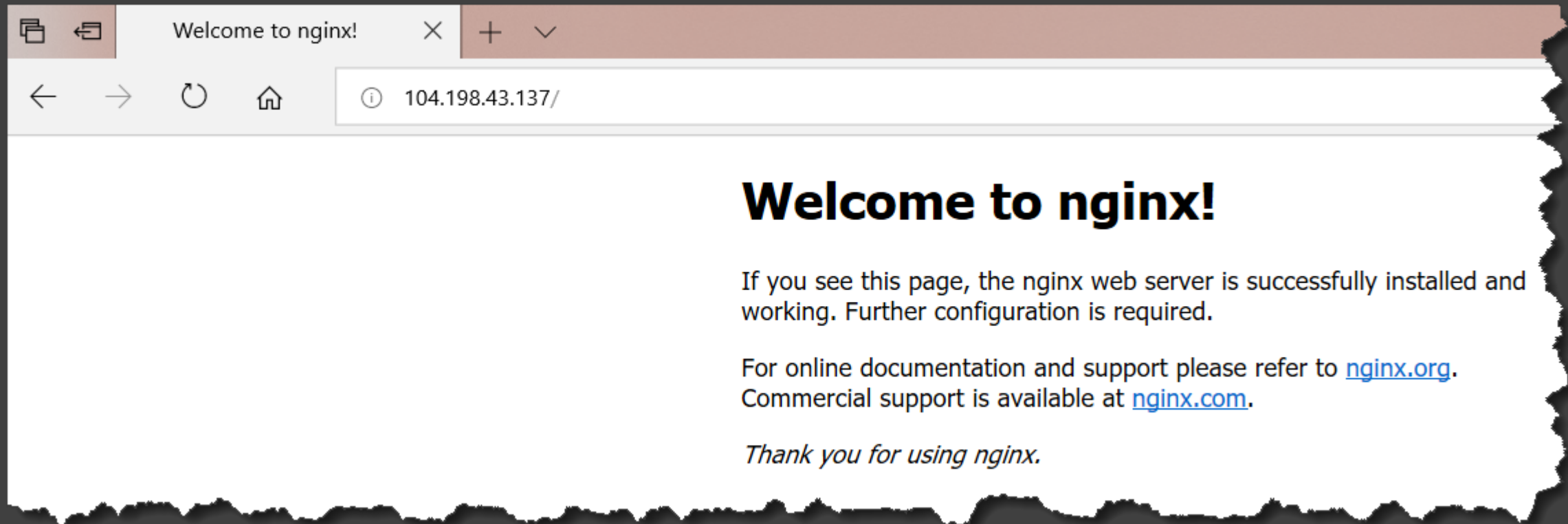
```
Events:
```

Type	Reason	Age	From	Message
----	-----	----	----	-----
Normal	EnsuringLoadBalancer	15m	service-controller	Ensuring load balancer
Normal	EnsuredLoadBalancer	14m	service-controller	Ensured load balancer

LoadBalancer – Accessing using LoadBalancer IP

```
srinath@master:~ $ kubectl describe service my-service | grep Load
```

```
Type: LoadBalancer  
LoadBalancer Ingress: 104.198.43.137
```



LoadBalancer – Deleting Service

```
[srinath@master ~]$ kubectl delete service my-service  
service "my-service" deleted
```

```
[srinath@master ~]$ kubectl get pods  
No resources found
```

Summary

Concept

- a. Why we need?
- b. LoadBalancer

Review Demo

- a. Manifest file
- b. Create and display
- c. Test use cases
- d. Clean up

Coming up...

Demo LoadBalancer