Configuring and Managing Application Access with Services

Kien Bui
DevOps & Platform Engineer

Course Overview

Kubernetes Networking Fundamentals

Configuring and Managing Application Access with Services

Configuring and Managing Application Access with Ingress

Summary

Understanding Services

Types of Services

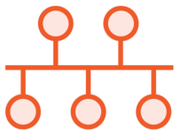
Service network internals

Service discovery

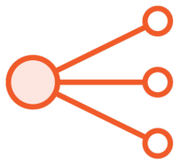
Understanding Services

Persistent endpoint access for clients

Adds persistency to the ephemerality of Pods



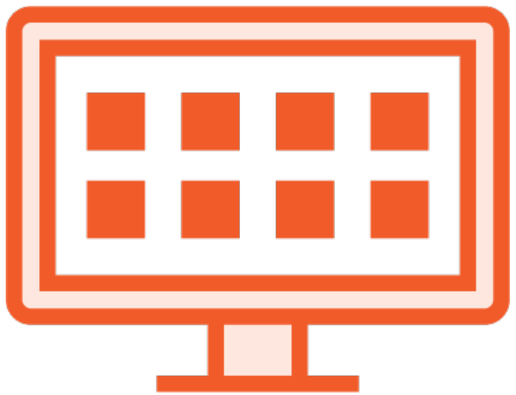
Networking abstraction providing persistent virtual IP and DNS



Load balances to the backend Pods

Automatically updated during Pod controller operations

How Services Work



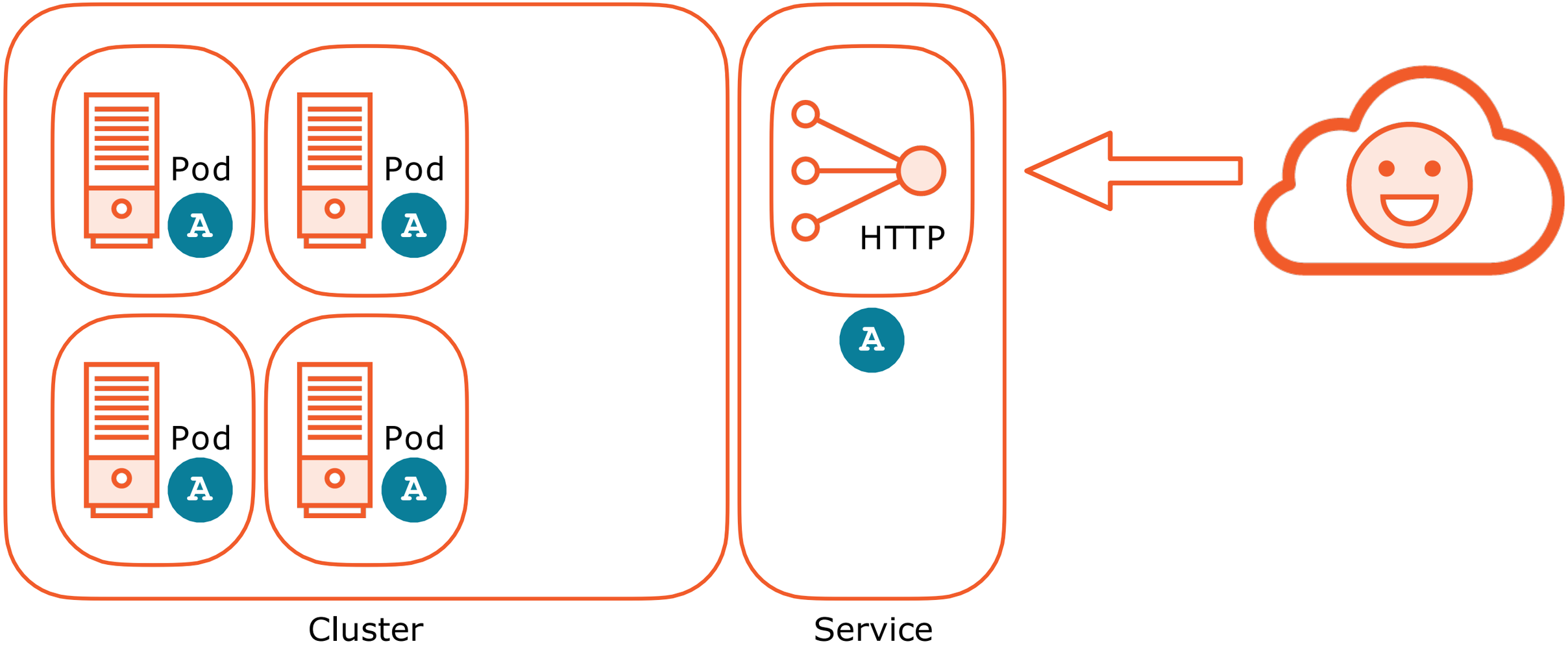
Services match Pods using Labels and Selectors

Creates and registers Endpoints in the Service (Pod IP and Port pair)

Implemented in the `kube-proxy` on the Node in iptables

kube-proxy watches the API Server and the Endpoints

Services



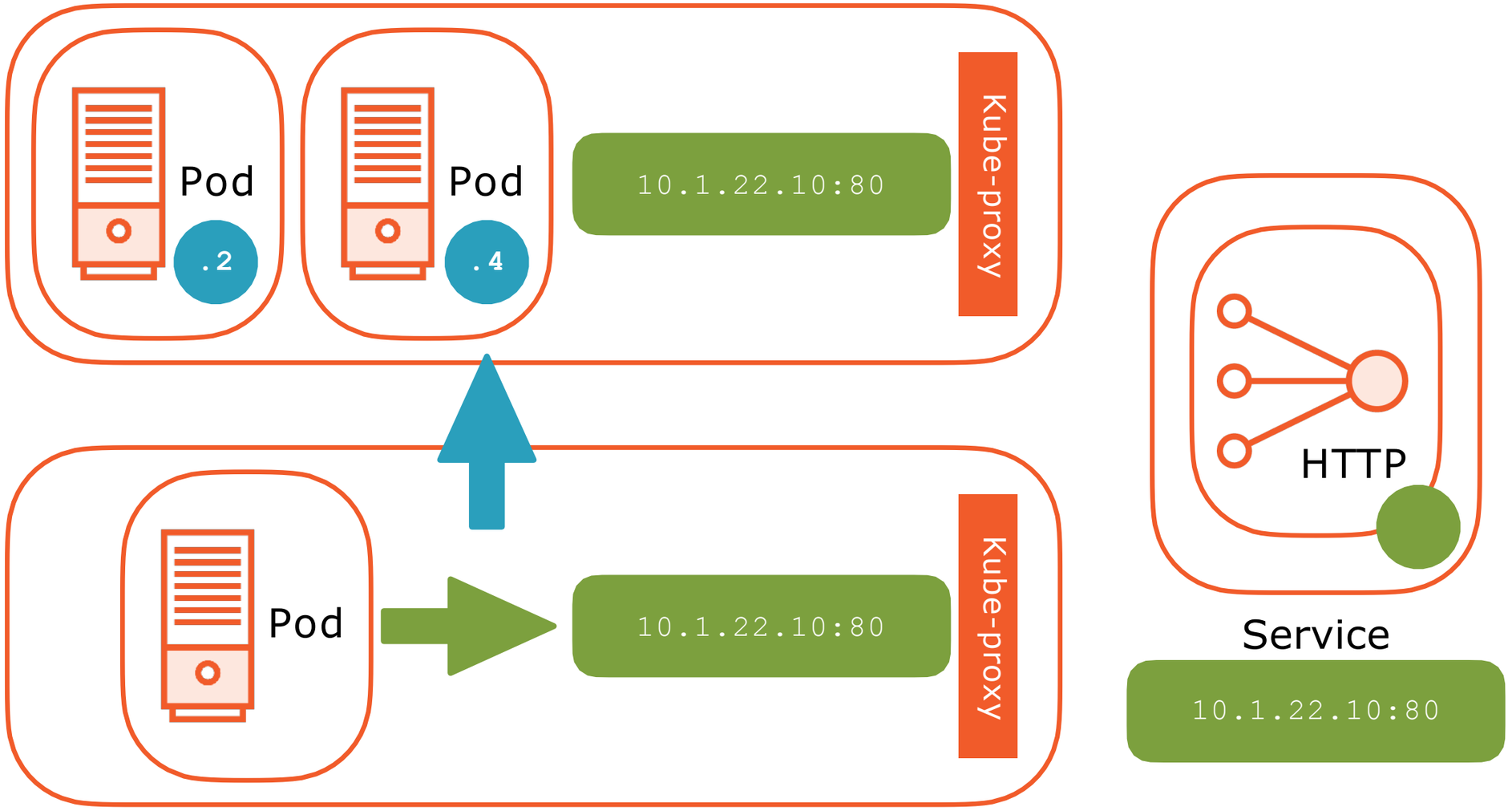
Service Types

ClusterIP

NodePort

LoadBalancer

ClusterIP

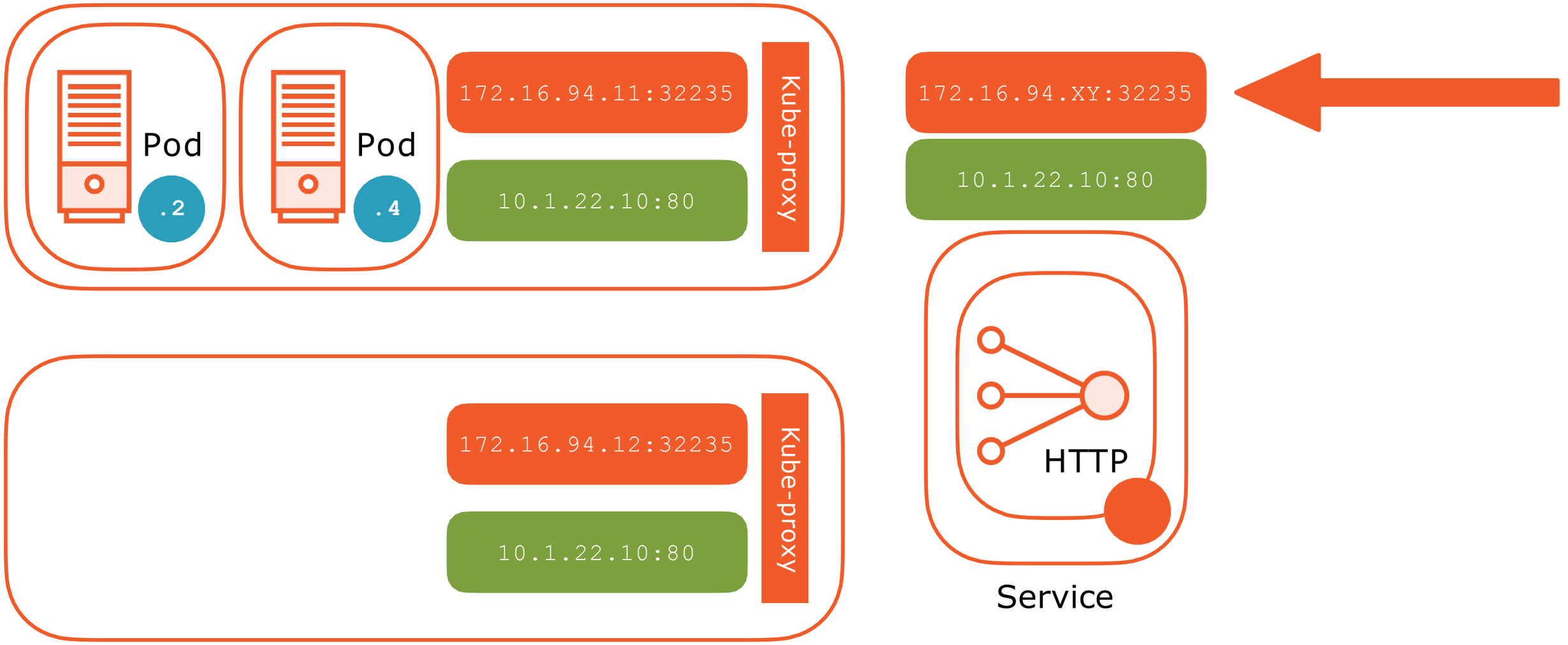


Pod Network

Node Network

Cluster Network

NodePort

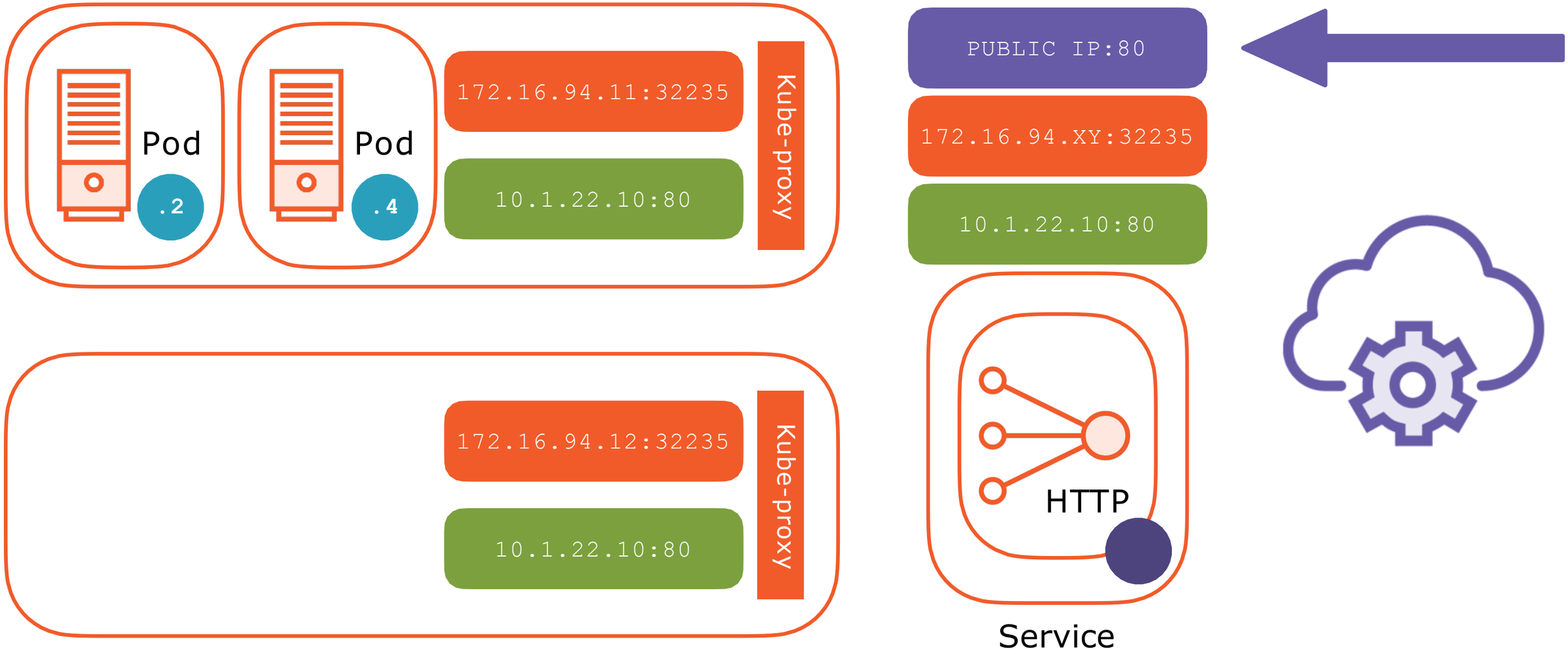


Pod Network

Node Network

Cluster Network

LoadBalancer



Defining Deployments and Services

```
kind: Deployment
...
  template:
    metadata:
      labels:
        run: hello-world
    spec:
      containers:
...
```


```
kind: Service
...
spec:
  type: ClusterIP
  selector:
    run: hello-world
  ports:
  - port: 80
    protocol: TCP
    targetPort: 8080
```

```
kubectl create deployment hello-world --image=gcr.io/google-samples/hello-app:1.0
```

```
kubectl expose deployment hello-world --port=80 --target-port=8080 --type NodePort
```

Demo

Exposing and accessing applications with Services

- ClusterIP
- NodePort
- LoadBalancer

Service Discovery

Infrastructure independence

Static configuration

DNS

Environment variables

Service Discovery

`Services` get DNS records in Cluster DNS

- ‘Normal’ `Services` get `A/AAAA` records
  - `<svcname>.<ns>.svc.<clusterdomain>`
  - `hello-world.default.svc.cluster.local`
- **`Namespaces`** get DNS subdomains
  - `<ns>.svc.<clusterdomain>`
  - `ns1.svc.cluster.local`
- Environment variables
  - Defined in Pods for each Service available at Pod start up

Other Types of Services

ExternalName	Headless	Without Selectors
Service discovery for external services	DNS but NO `ClusterIP`	Map to specific Endpoints
CNAME to resource	DNS Record for Each Endpoint	Manually create Endpoint objects
	Stateful applications	Point to any IP inside or outside cluster

Demo

Service Discovery

- DNS
- Environment Variables

Creating an ExternalName Service

Review

Understanding Services

Types of Services

Service Network Internals

Service Discovery

Up Next:
Configuring and Managing Application Access with Ingress