Configuring and Managing Application Access with Ingress

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

Ingress overview and architecture

- Ingress
- Ingress Controller

Common use cases

Ingress Architecture

Ingress Object

Ingress Controller

Ingress Overview

API Object

Defines rules for external access to Services

Load balancing to Endpoints

Name-based virtual hosts

Path-based routing

TLS termination

Ingress Controller

- Implements the rules defined in the Ingress Resource
- Many types of Ingress Controllers
  - Pods in a cluster - nginx
  - Hardware external to the cluster - Citrix and F5
  - Cloud Controllers - AppGW, Google Load Balancer and AWS ALB Ingress
- Ingress controllers have a defined spec

Why Ingress Rather Than Load Balancers?

Layer 7

Path-based routing

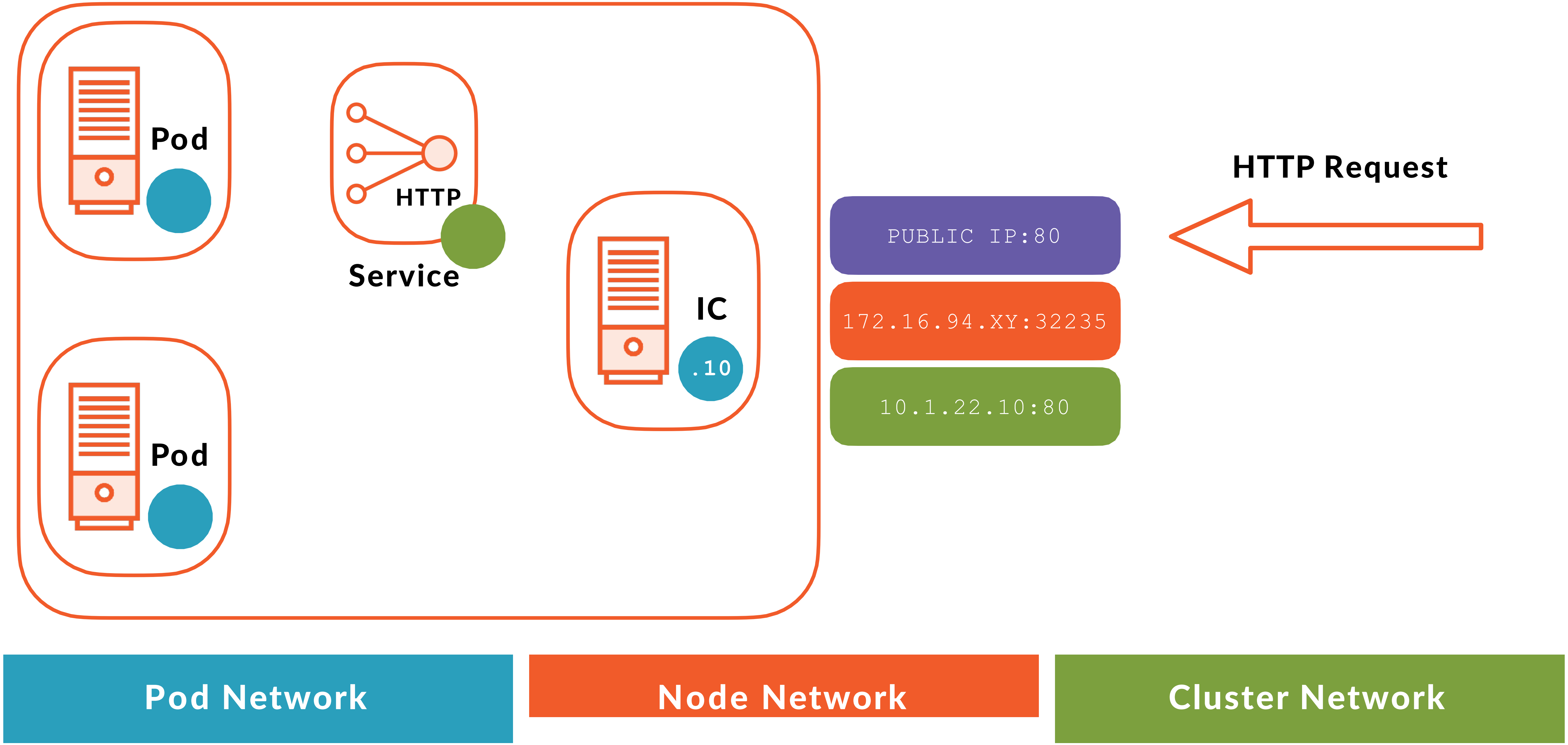
Name-based virtual hosts

Higher level capabilities

Single resource

Reduced latency

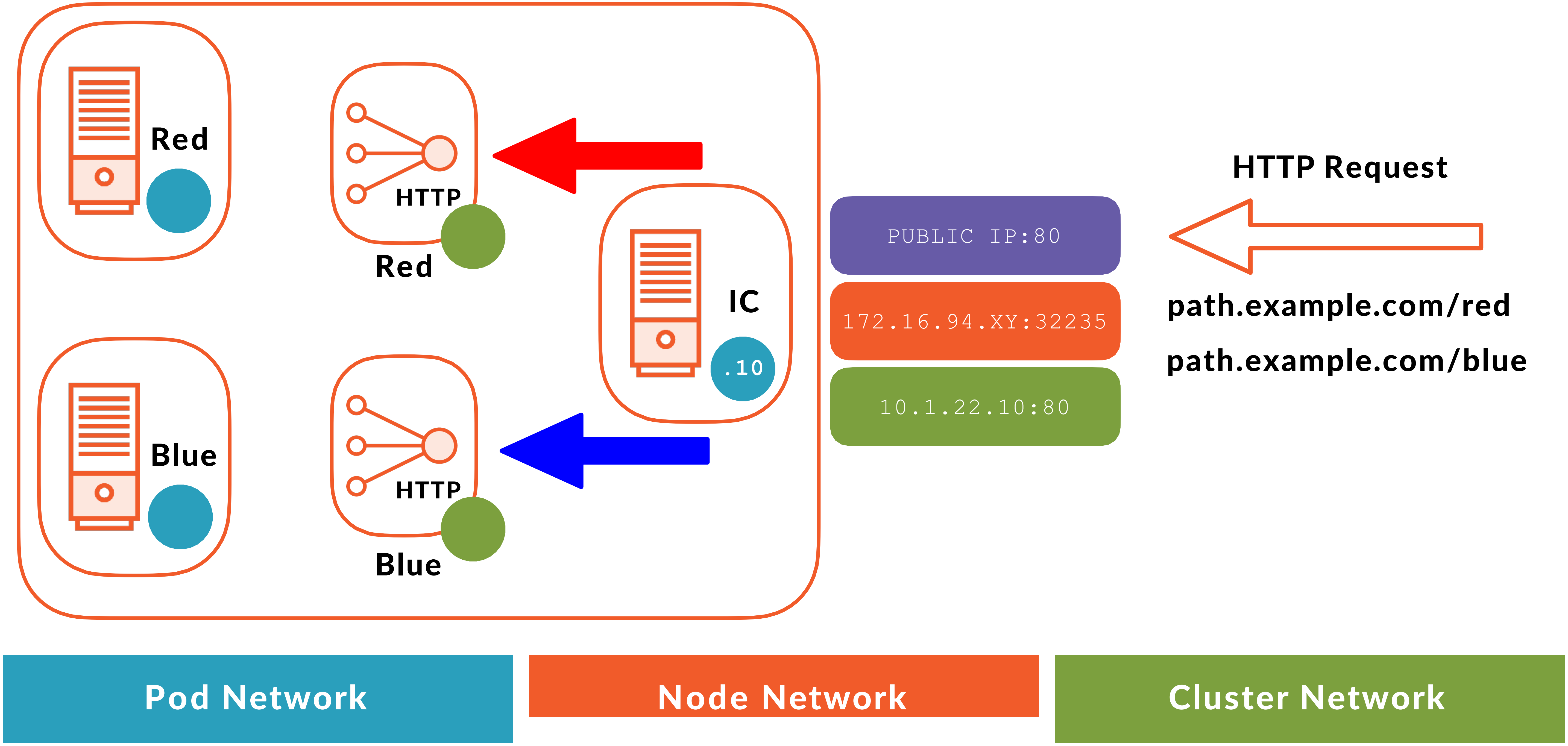
Exposing a Single Service with Ingress



Exposing a Single Service with Ingress

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: ingress-single
spec:
  backend:
    serviceName: hello-world-service-single
    servicePort: 80
```

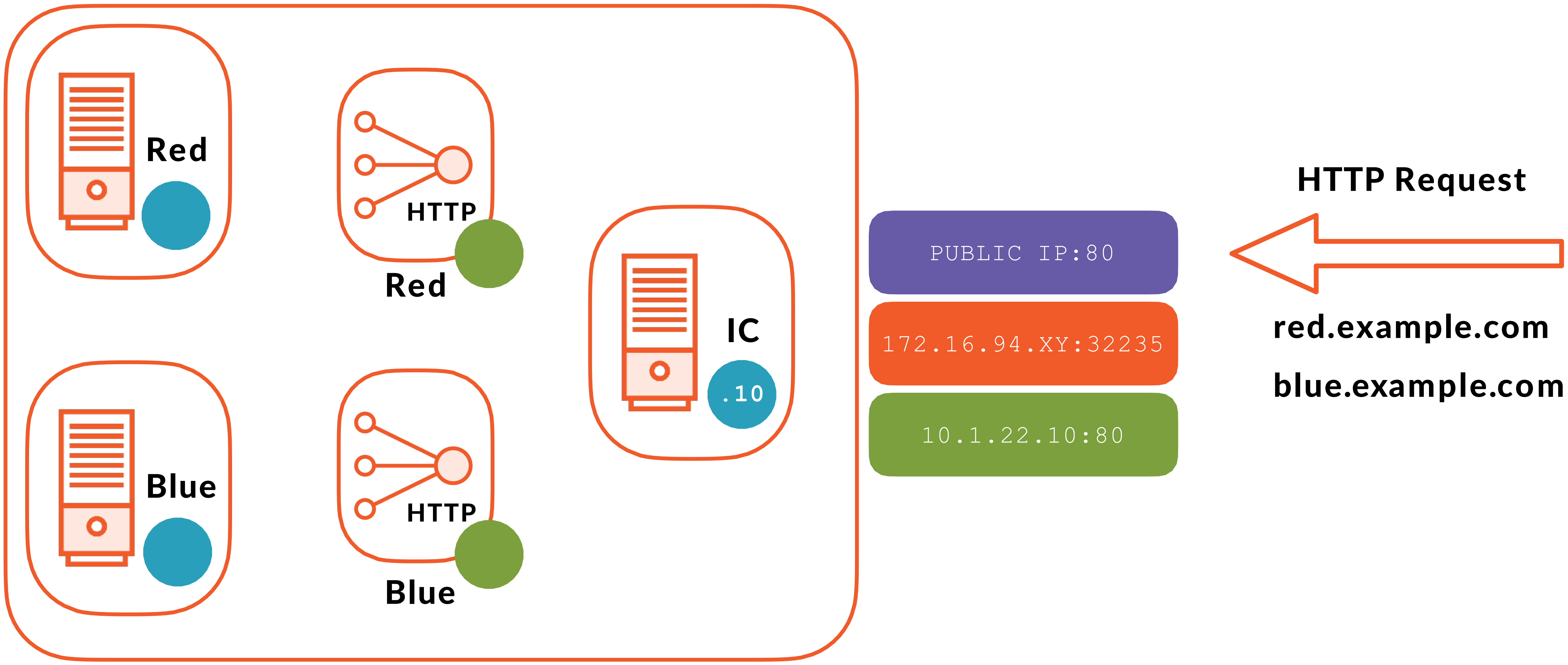
Exposing Multiple Services with Ingress



Exposing Multiple Services with Ingress

```
spec:
  rules:
    - host: path.example.com
      http:
        paths:
        - path: /red
          backend:
            serviceName: hello-world-service-red
            servicePort: 4242
        - path: /blue
          backend:
            serviceName: hello-world-service-blue
            servicePort: 4343
  backend:
    serviceName: hello-world-service-single
    servicePort: 80
```

Name Based Virtual Hosts with Ingress



Pod Network

Node Network

Cluster Network

Name Based Virtual Hosts with Ingress

```
spec:
  rules:
    - host: red.example.com
      http:
        paths:
        - backend:
            serviceName: hello-world-service-red
            servicePort: 4242
    - host: blue.example.com
      http:
        paths:
        - backend:
            serviceName: hello-world-service-blue
            servicePort: 4343
```

Using TLS certificates for HTTPS Ingress

```
spec:
  tls:
  - hosts:
    - tls.example.com
    secretName: tls-secret
  rules:
  - host: tls.example.com
    http:
      paths:
       backend:
          serviceName: hello-world-service-single
          servicePort: 80
```

Demo

Using Ingress to provide access to

- Single service
- Multiple services
- Services using a default backen
- Services using name based virt
- TLS configuration

Review

Ingress Overview and Architecture

- Ingress
- Ingress Controller

Common Use Cases