

# Configuring and Managing Application Access with Ingress

---



**Anthony E. Nocentino**

ENTERPRISE ARCHITECT @ CENTINO SYSTEMS

@nocentino [www.centinosystems.com](http://www.centinosystems.com)

# 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 Resource**

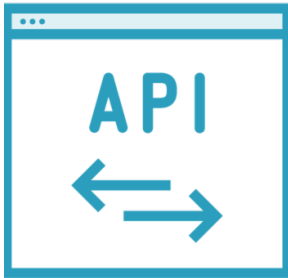


**Ingress Controller**



**Ingress Class**

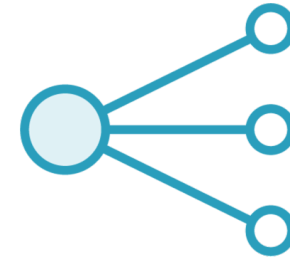
# Ingress Overview



Ingress Resource



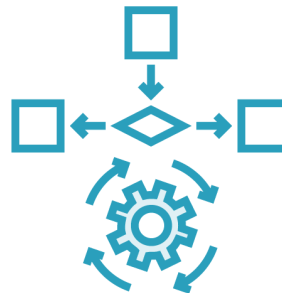
Defines rules for external access to Services



Load balancing to Endpoints



Name-based virtual hosts



Path-based routing



TLS termination

**Managing Ingress Traffic Patterns for Kubernetes Services**

# 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, F5 and more**

**Cloud Ingress 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**

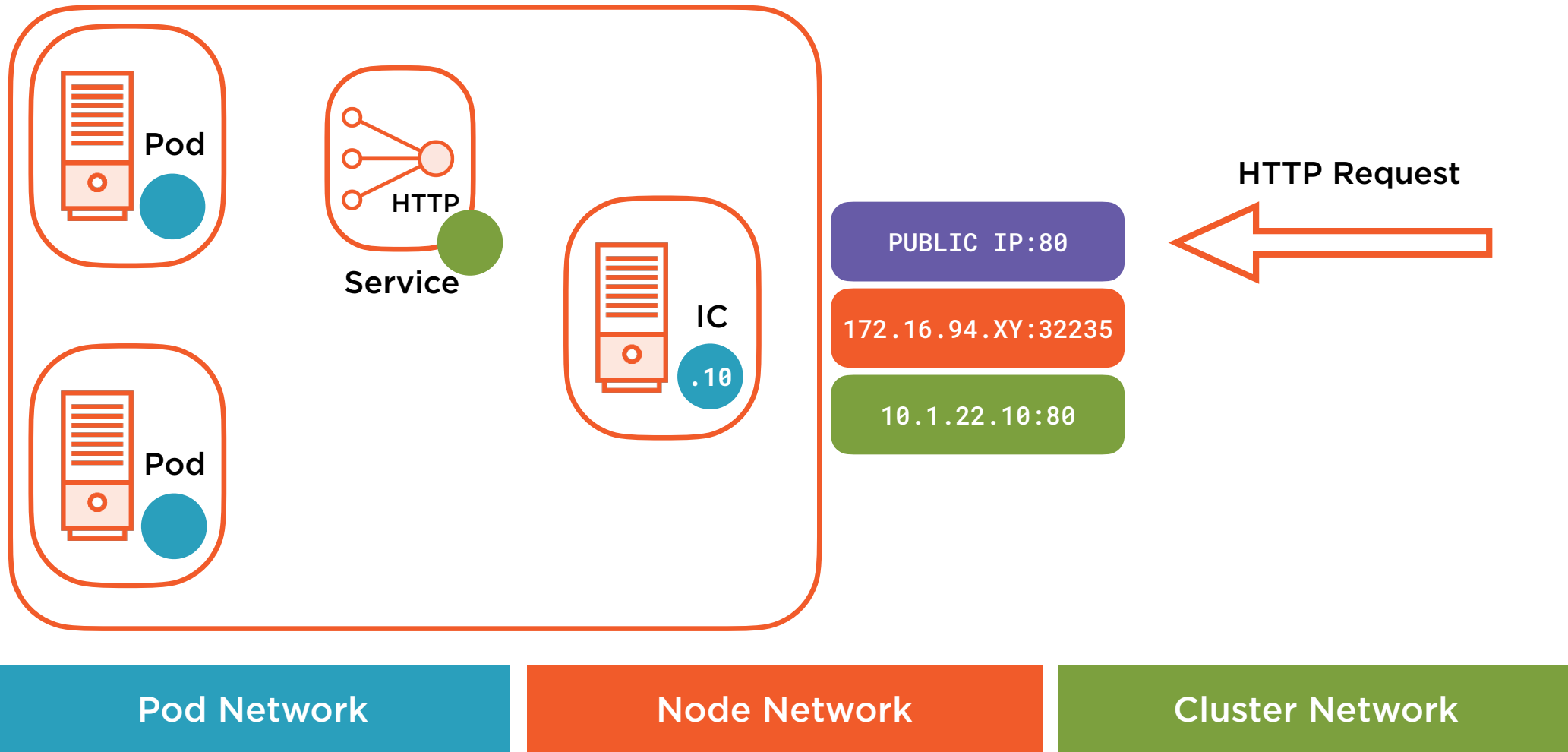
**Name-based virtual  
hosts**

**Higher level  
capabilities**

**Single resource**

**Reduced latency**

# Exposing a Single Service with Ingress

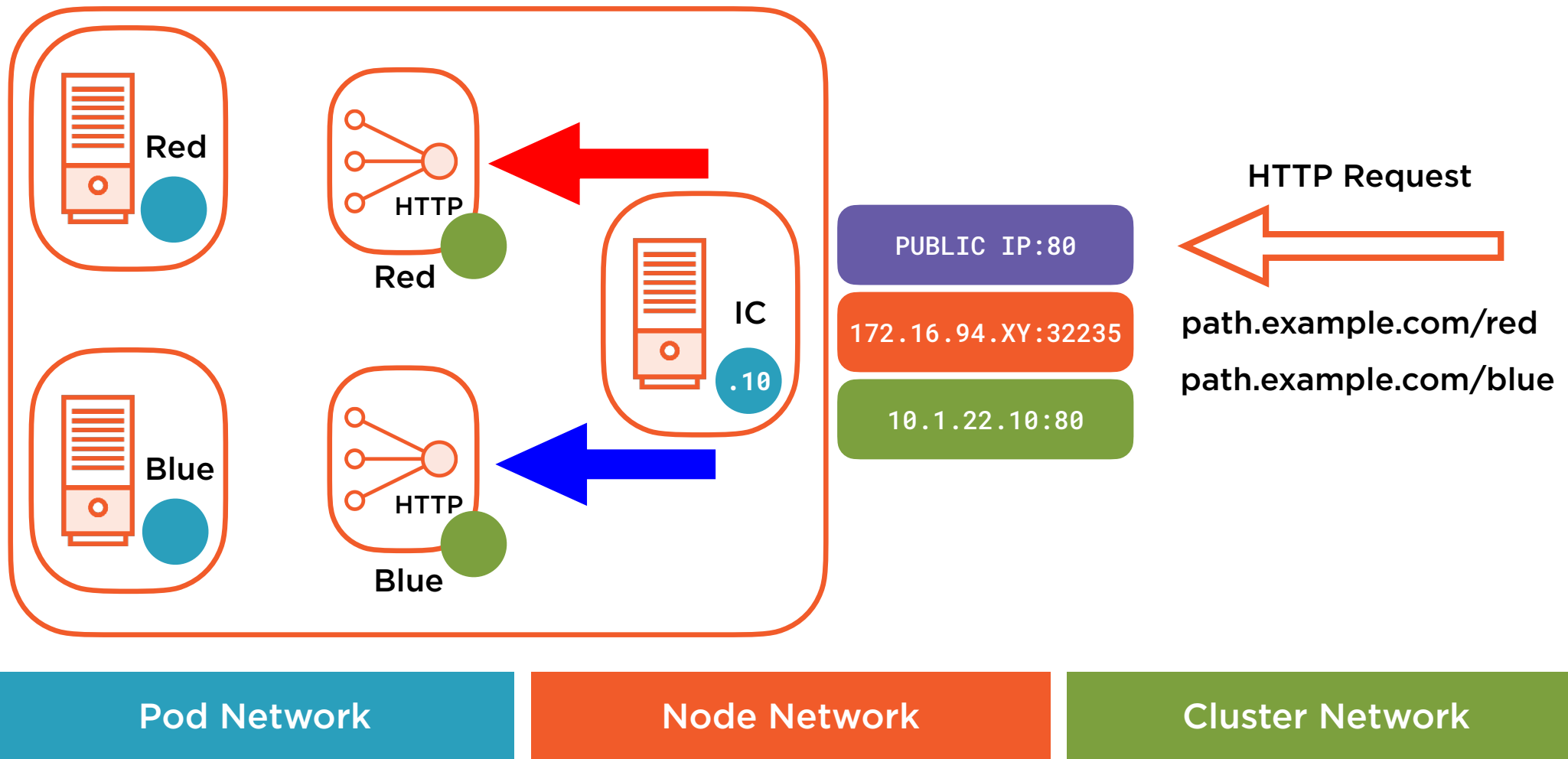




# Exposing a Single Service with Ingress

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: ingress-single
spec:
  ingressClassName: nginx
  defaultBackend:
    service:
      name: hello-world-service-single
      port:
        number: 80
```

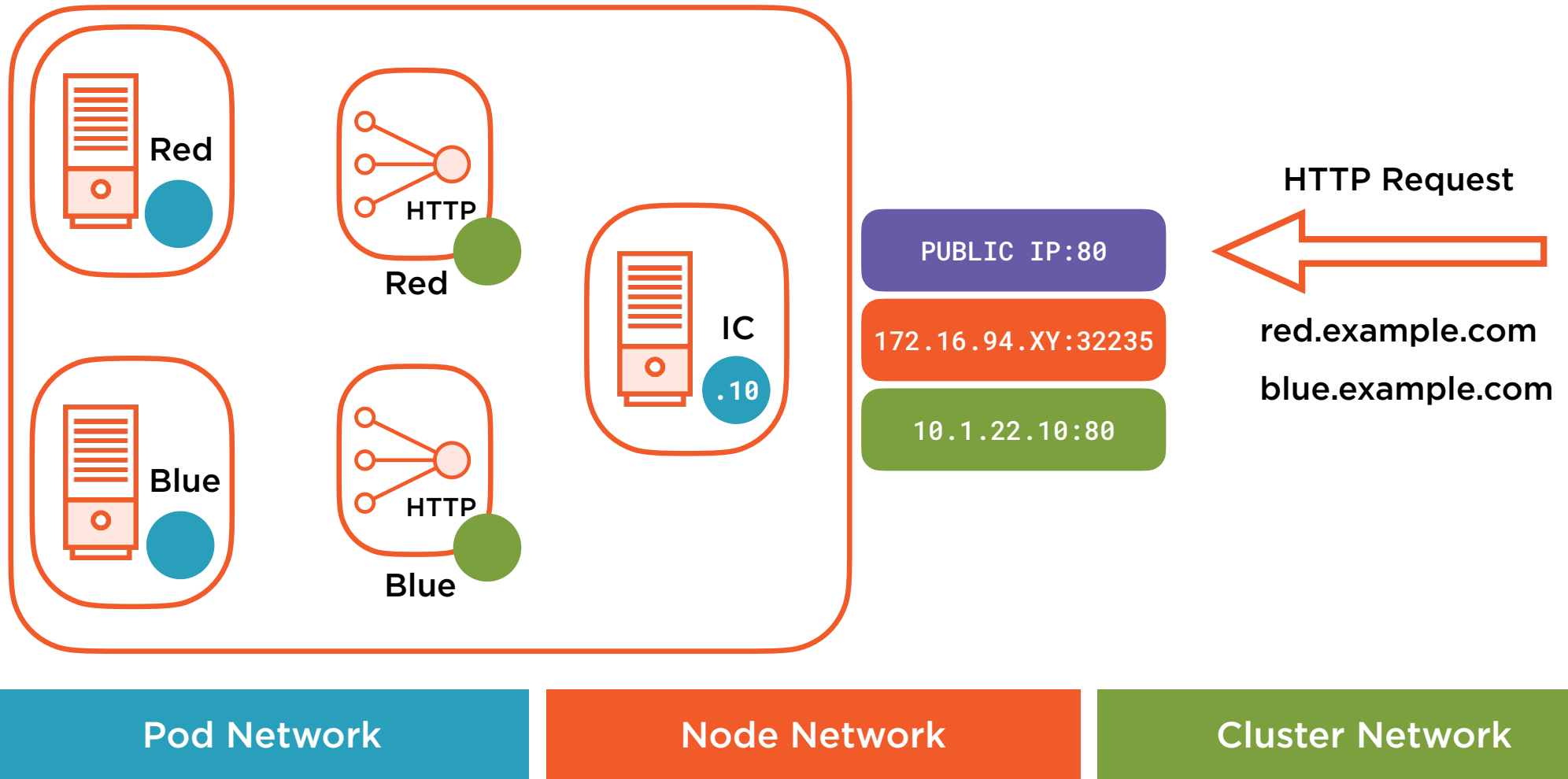
# Exposing Multiple Services with Ingress



# Exposing Multiple Services with Ingress

```
spec:
  ingressClassName: nginx
  rules:
    - host: path.example.com
      http:
        paths:
          - path: /red
            pathType: Prefix
            backend:
              service:
                name: hello-world-service-red
                port:
                  number: 4242
          - path: /blue
            pathType: Exact
            backend:
              service:
                name: hello-world-service-blue
                port:
                  number: 4343
        defaultBackend:
          service:
            name: hello-world-service-single
            port:
              number: 80
```

# Name Based Virtual Hosts with Ingress



# Name Based Virtual Hosts with Ingress

```
spec:
  ingressClassName: nginx
  rules:
    - host: red.example.com
      http:
        paths:
          - pathType: Prefix
            path: "/"
            backend:
              service:
                name: hello-world-service-red
                port:
                  number: 4242
    - host: blue.example.com
      http:
        paths:
          - pathType: Prefix
            path: "/"
            backend:
              service:
                name: hello-world-service-blue
                port:
                  number: 4343
```

# Using TLS certificates for HTTPS Ingress

```
spec:
  ingressClassName: nginx
  tls:
    - hosts:
        - tls.example.com
      secretName: tls-secret
  rules:
    - host: tls.example.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: hello-world-service-single
                port:
                  number: 80
```

# Demo

Using Ingress to provide access to

- Single service
- Multiple services
- Services using a default backend
- Services using name based virtual hosts
- TLS configuration

# Review

## **Ingress Overview and Architecture**

- **Ingress**
- **Ingress Controller**

## **Common Use Cases**



Thank You!  
@nocentino