K8S Introduction

Basic Objects

- Container
  Containerized application (usually a single process per Container)
- Pod
  A set of Containers sharing the same network namespace (one IP per Pod)
- Replicaset
  A set of Pods running the same application (instances of the same class of application)

Deployment

- Deployment
  A definition of a set of Containers forming a Pod and optionally a Replicaset

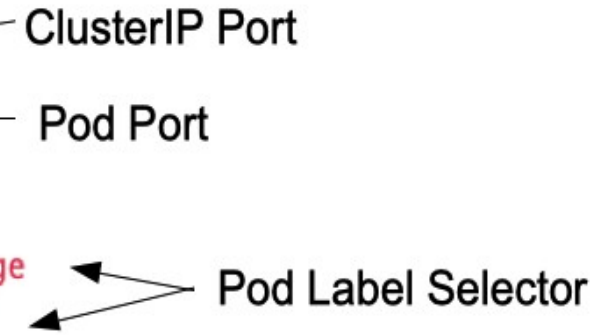
Deployment Example

Service

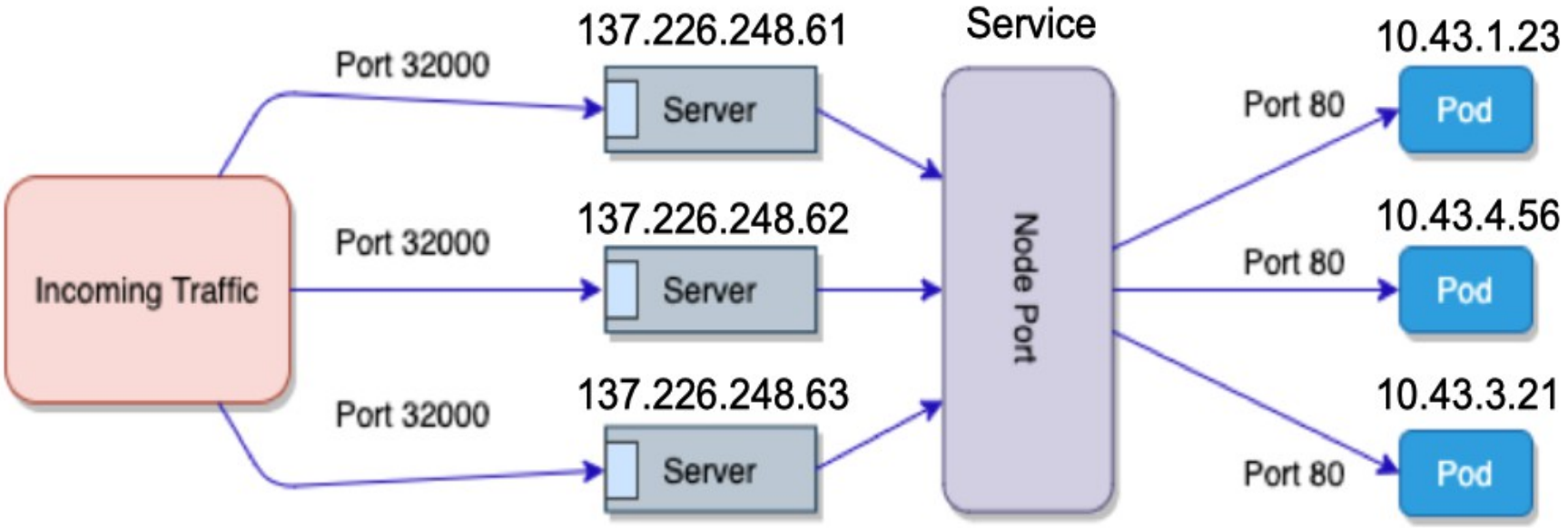
- Service
  A (load balanced) endpoint of the Pod's application which is either of type
  - ClusterIP, reachable inside the k8s cluster (10.43.0.0/16)
  - NodePort, reachable inside OpenStack cluster via any k8s node (192.168.1.0/24)
    also reachable via 137.226.248.61, 137.226.248.62 and 137.226.248.63 (master.acs-infra-k8s.osc.eonerc.rwth-aachen.de)
    for all TCP/UDP services on Ports 30000-32767 inside RWTH Network

Service ClusterIp Example

```yaml
apiVersion: v1
kind: Service
metadata:
  name: kubernetes-emptypage
  namespace: kubernetes-entrypage
spec:
  ports:
  - port: 80
    targetPort: 80
    name: http
  selector:
    app: kubernetes-emptypage
    role: nginx
```

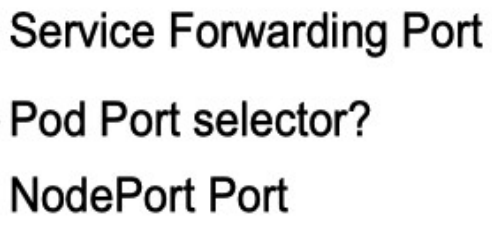


Service NodePort



Service NodePort Example

```yaml
apiVersion: v1
kind: Service
metadata:
  name: kubernetes-entrypage
  namespace: kubernetes-entrypage
  labels:
    app: kubernetes-entrypage
spec:
  type: NodePort
  ports:
  - port: 80
    protocol: TCP
    name: http
    nodePort: 30077
  selector:
    app: kubernetes-entrypage
    role: nginx
```



Ingress

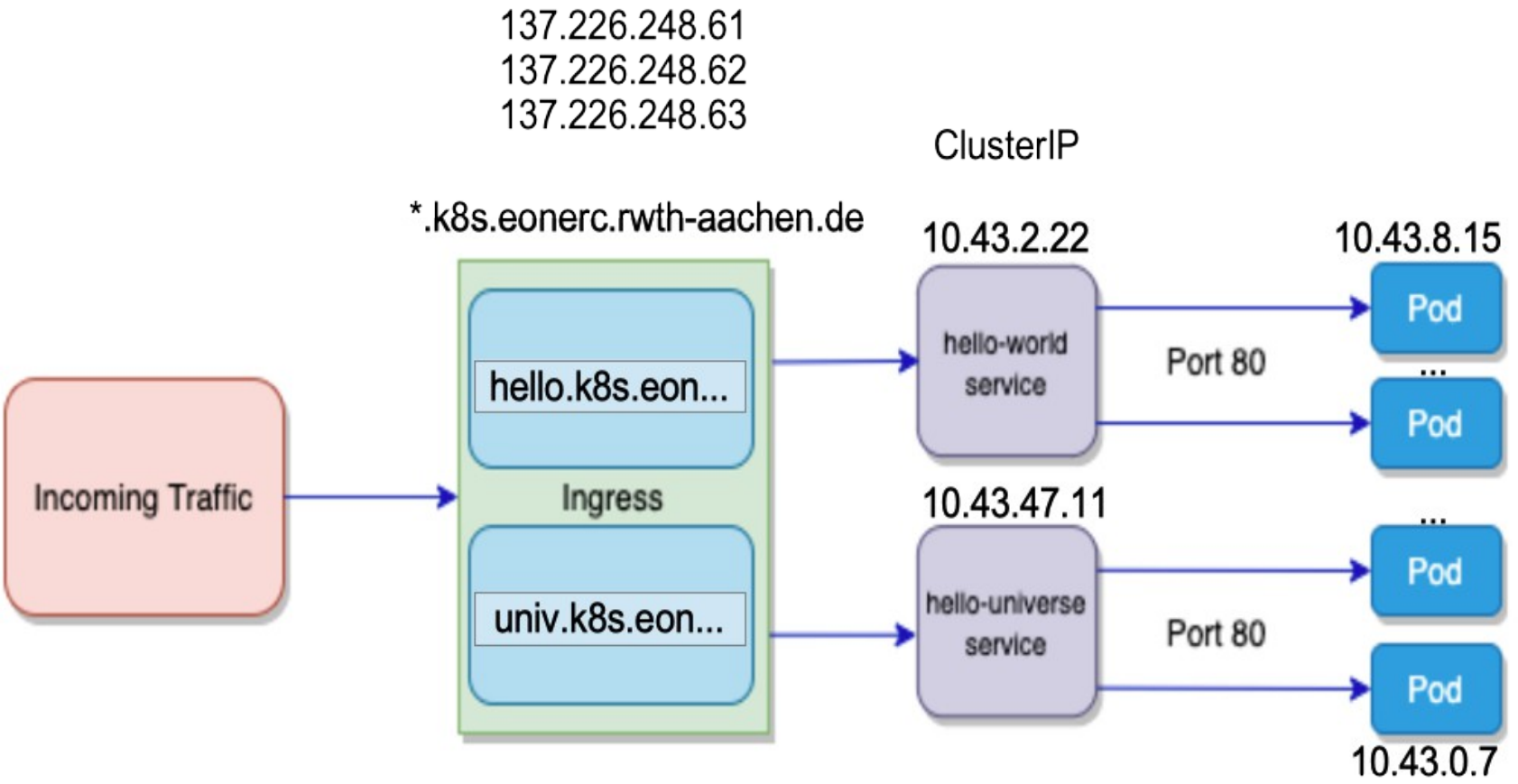
- Ingress
  A set of rules that define a virtual hostname and it's proxy connection to a Service (ClusterIP or NodePort).

  Virtual hosts are also provided a let's encrypt certificate.
  DNS names, ip addresses and ports (80 HTTP and 443 HTTPS only) are reachable worldwide and can be formed as subdomains of

  - *.k8s.eonerc.rwth-aachen.de
  - *.k8s.fein-aachen.org
  - *.acs-infra-k8s.osc.eonerc.rwth-aachen.de

  completely automatically.

Ingress


137.226.248.61
137.226.248.62
137.226.248.63
*.k8s.eonerc.rwth-aachen.de
ClusterIP
10.43.2.22
10.43.8.15
Incoming Traffic
hello.k8s.eon...
Ingress
univ.k8s.eon...
hello-world service
hello-universe service
Port 80
Port 80
Pod
...
Pod
10.43.47.11
...
Pod
Pod
10.43.0.7

Ingress Example

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: kubernetes-entrypage
  namespace: kubernetes-entrypage
  labels:
    app: kubernetes-entrypage
  annotations:
    kubernetes.io/ingress.class: nginx
    cert-manager.io/cluster-issuer: letsencrypt-prod
spec:
  tls:
  - hosts:
    - k8s.eonerc.rwth-aachen.de
    secretName: letsencrypt-prod-kubernetes-entrypage
  rules:
  - host: k8s.eonerc.rwth-aachen.de
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: kubernetes-entrypage
            port:
              number: 80
```

Certificate Settings

Certificate DNS Name

Certificate Store Location

Proxy Mappings

Hostname

Path

Service

The End

Thanks for your Attention

Option: Introduction to Rancher Dashboard