

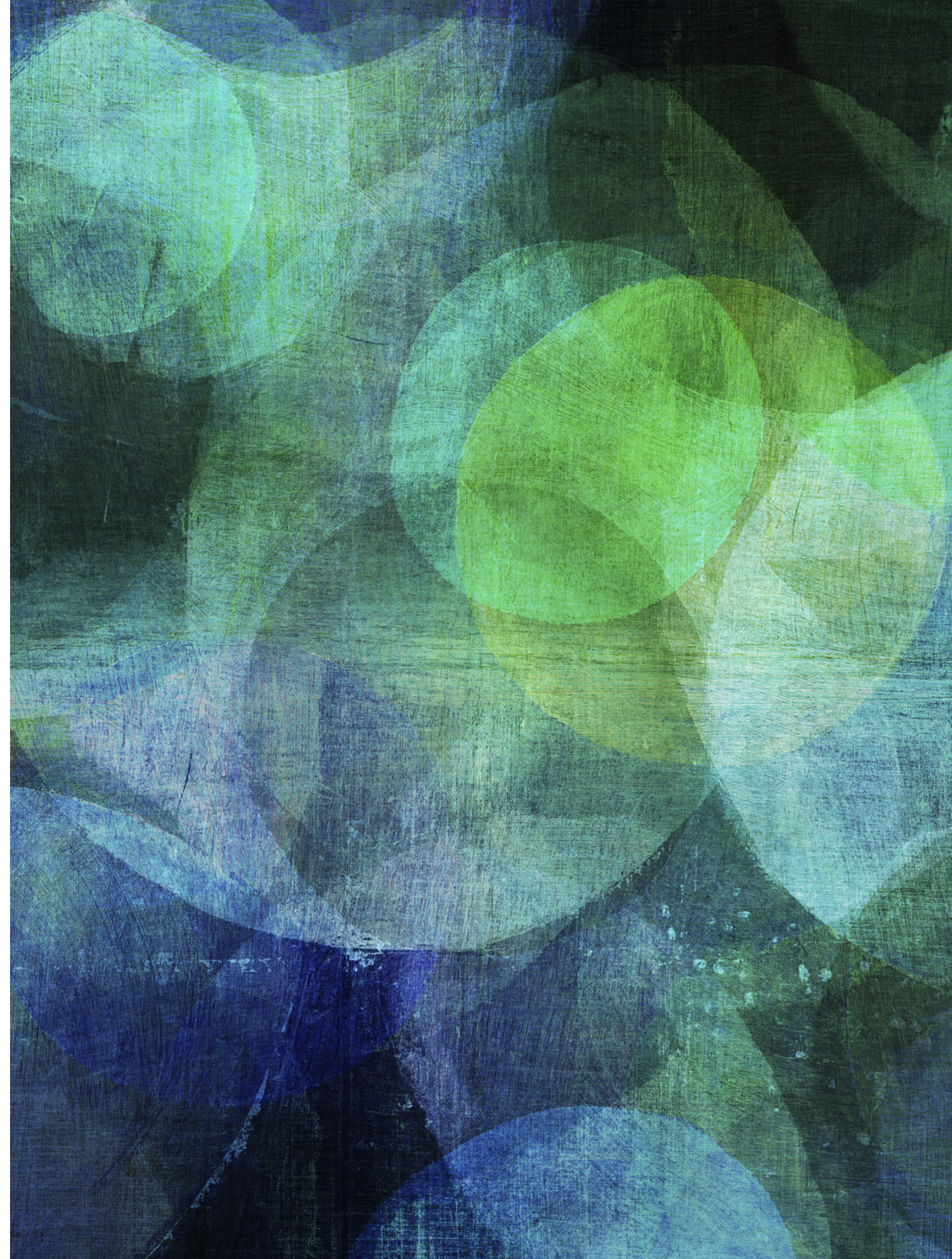
INTRODUCTION TO KUBERNETES



INTRODUCTION TO KUBERNETES

External Service

AGENDA



AGENDA

– Service Types



AGENDA

- Service Types
- Node Port



AGENDA

- Service Types
 - Node Port
- Load Balancer



AGENDA

- Service Types
 - Node Port
- Load Balancer
- ExternalName/IP



PUBLISHING SERVICES (SERVICETYPES)

.....

TYPE OF SERVICES

ClusterIP

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

NodePort

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: NodePort
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

LoadBalancer

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```


TYPE OF SERVICES

ClusterIP

Internal Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

NodePort

External Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: NodePort
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

LoadBalancer

External Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```


TYPE OF SERVICES

ClusterIP

Internal Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

NodePort

External Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: NodePort
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

LoadBalancer

External Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```

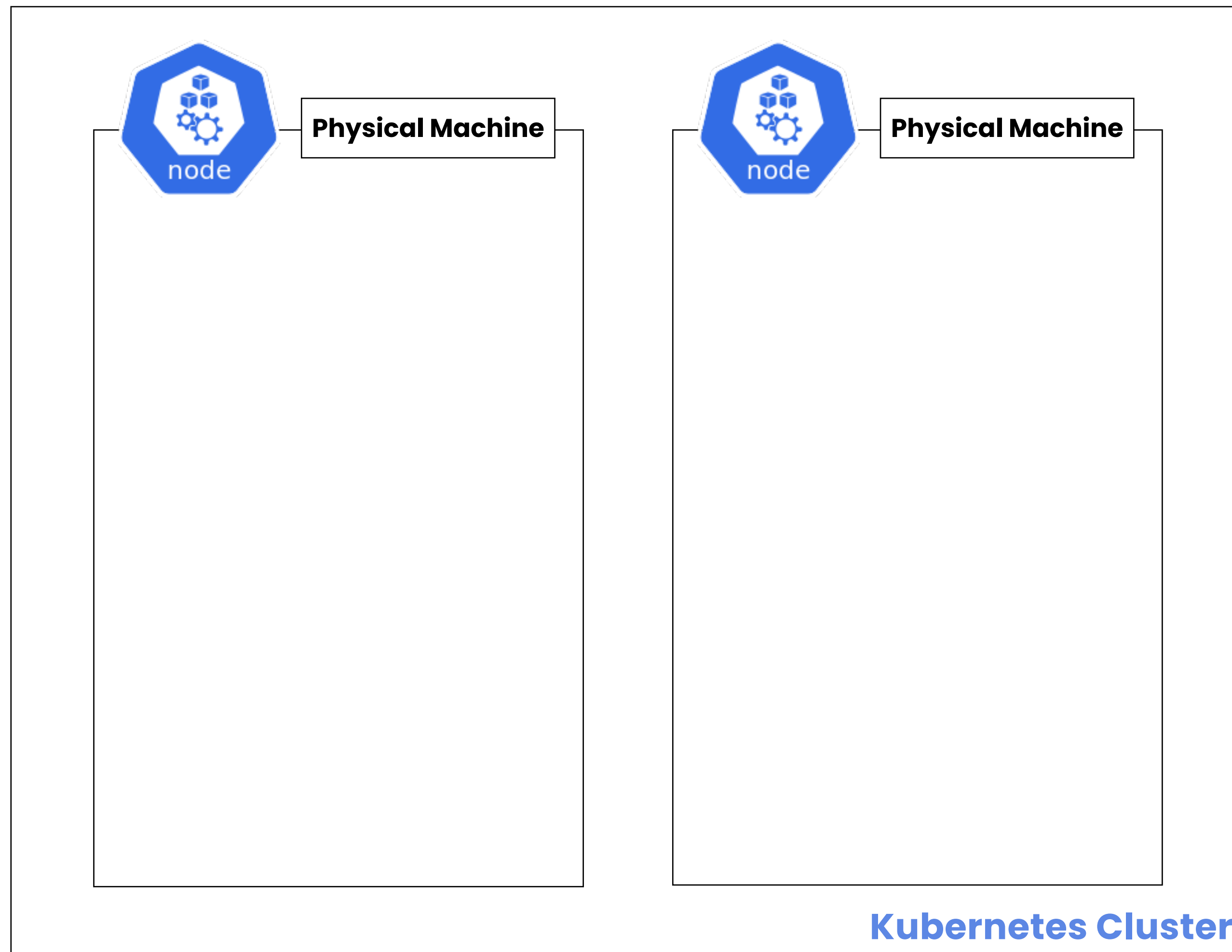
ExternalName

External Service

NODEPORT SERVICES

.....

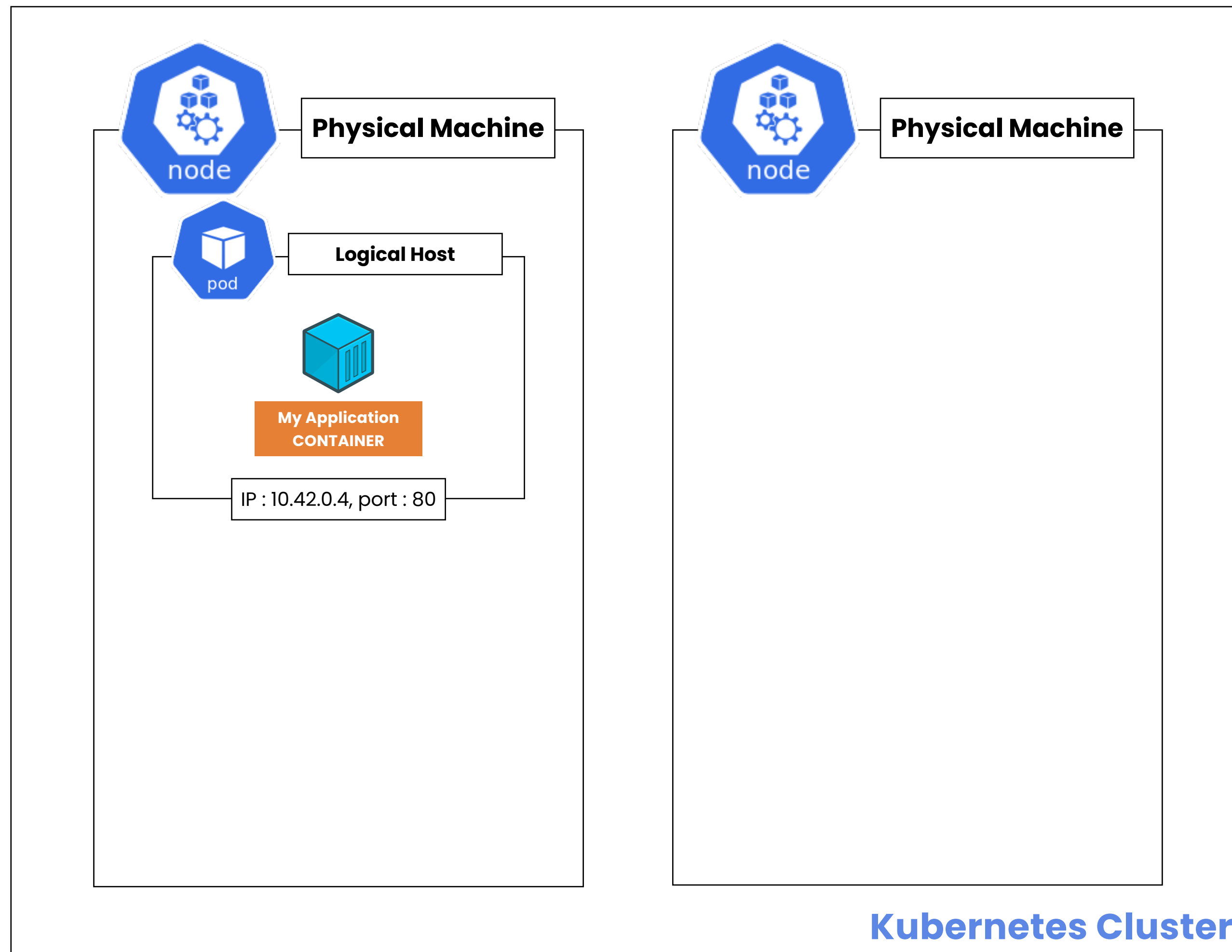
NODEPORT SERVICE



NodePort: Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting `<NodeIP>:<NodePort>`.

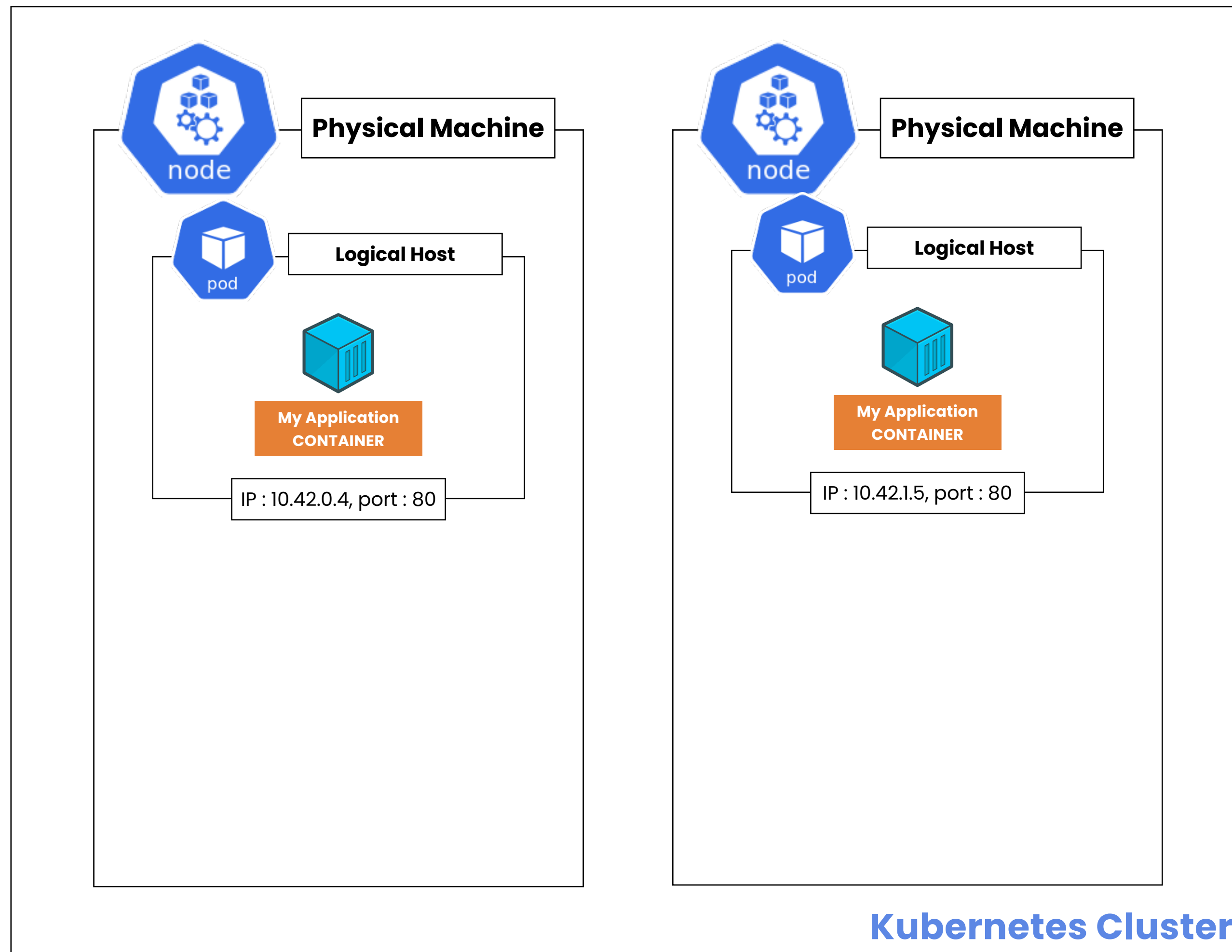
NODEPORT SERVICE



NodePort : **Exposes** the Service on each **Node's IP** at a **static port** (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the *NodePort* Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

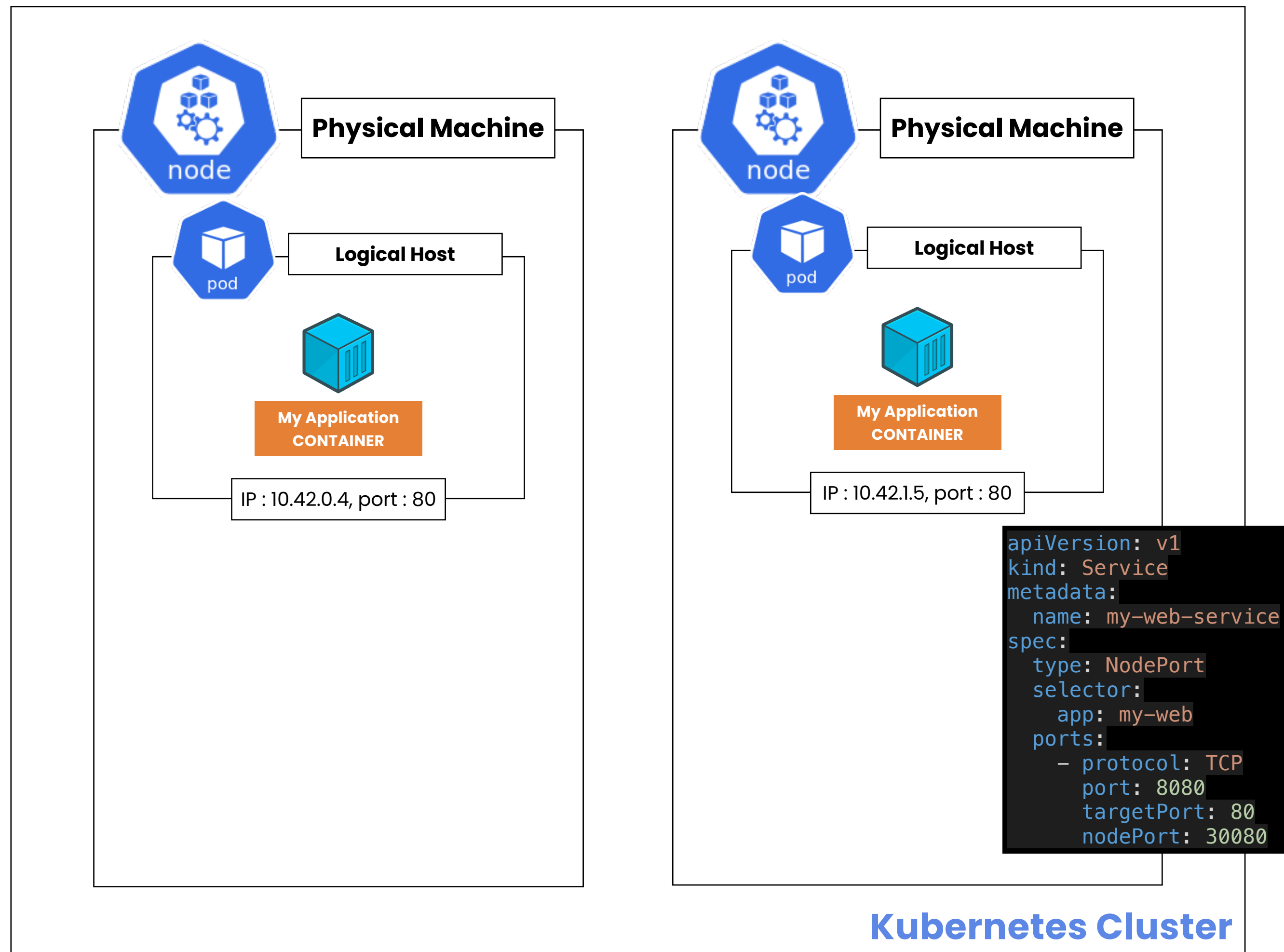
NODEPORT SERVICE



NodePort : Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting `<NodeIP>:<NodePort>`.

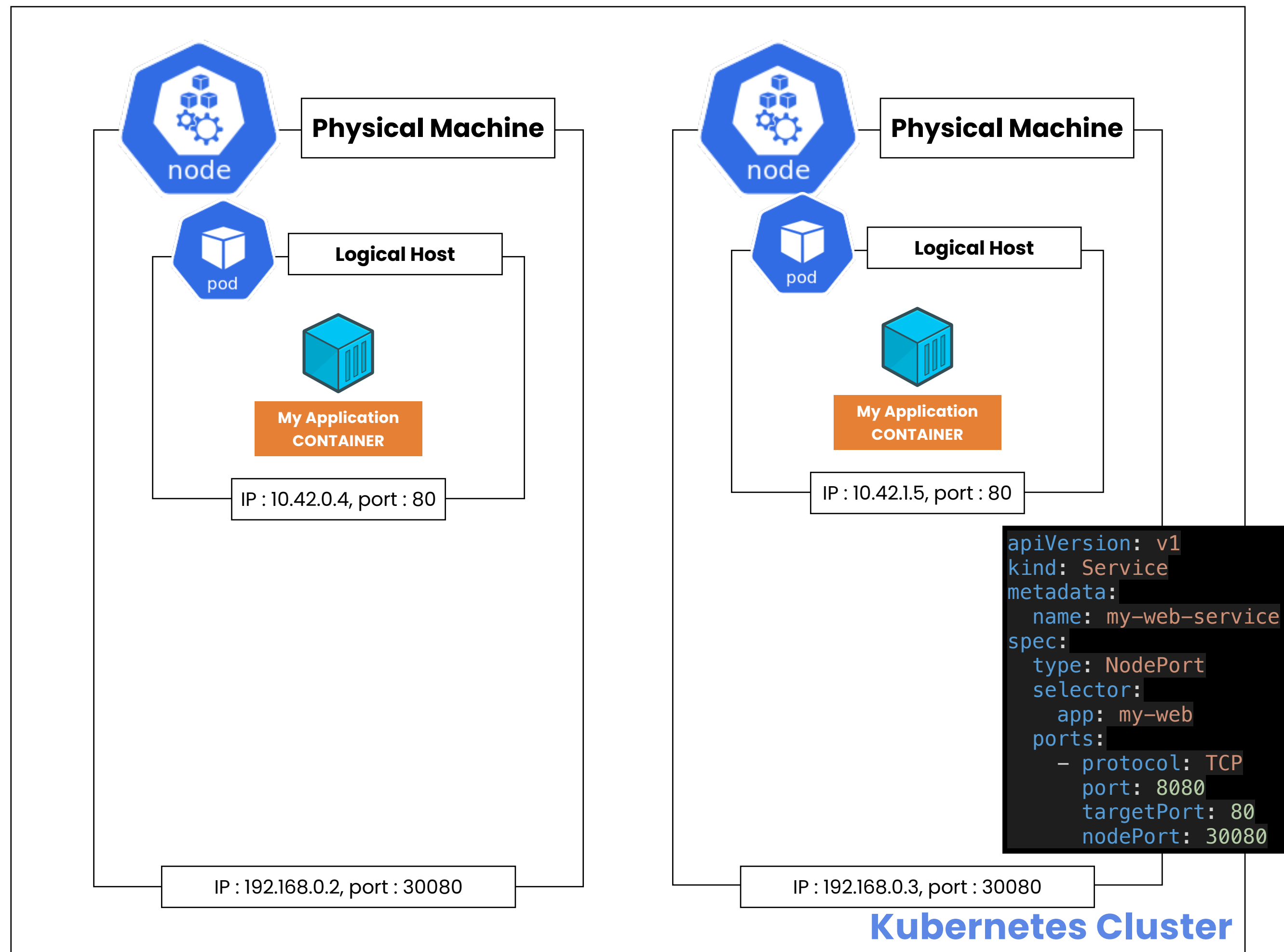
NODEPORT SERVICE



NodePort : Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

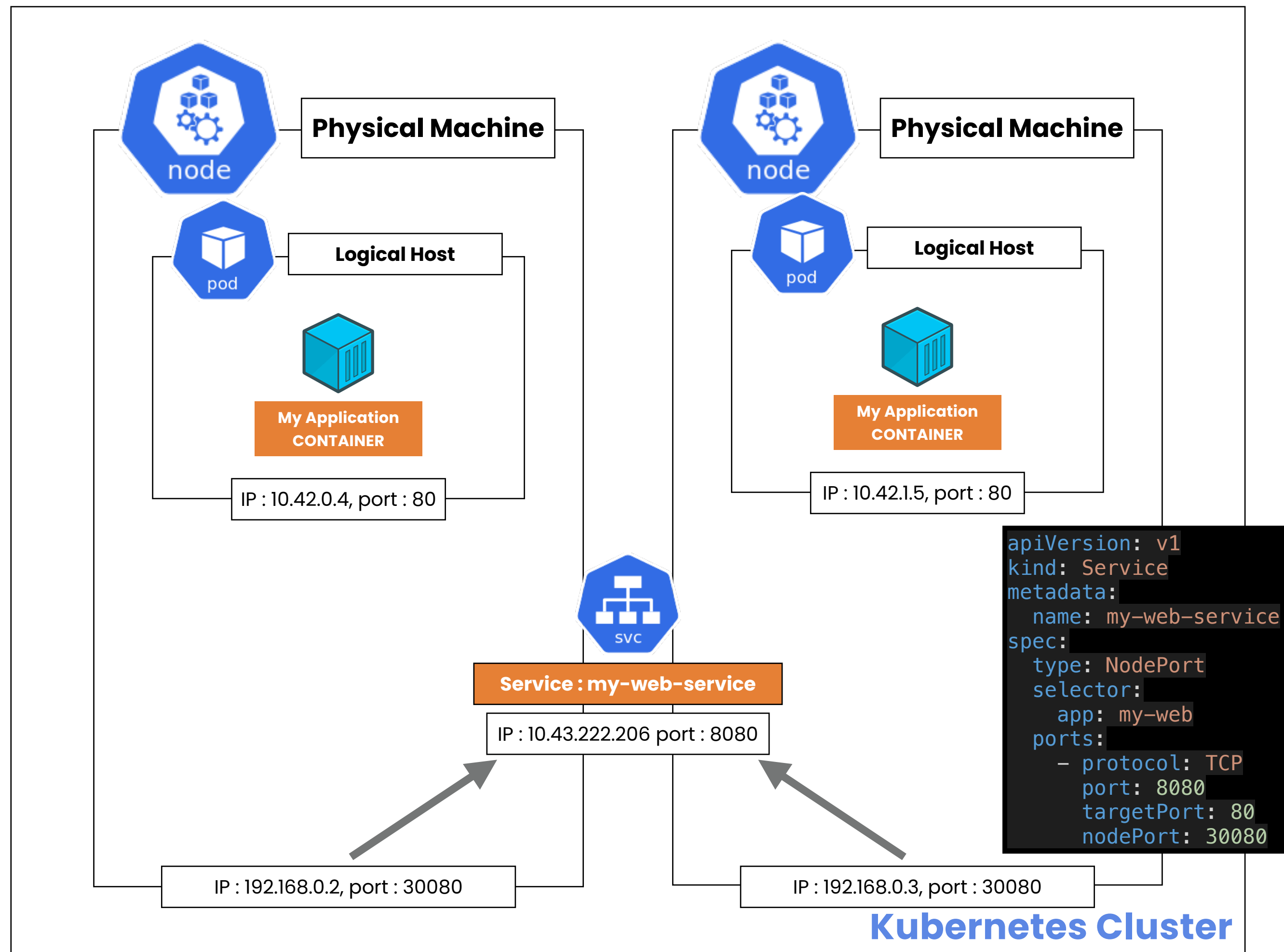
NODEPORT SERVICE



NodePort : Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

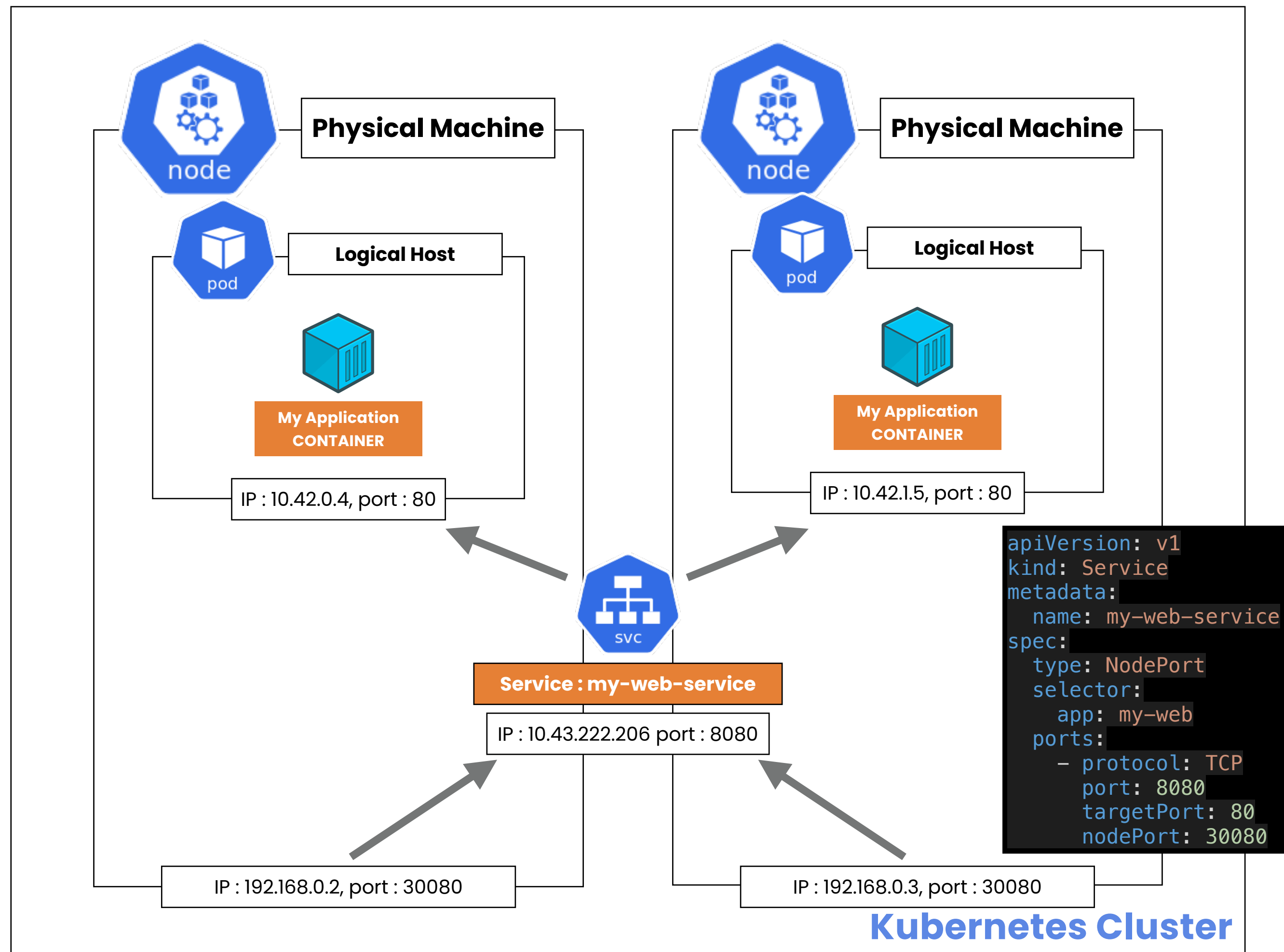
NODEPORT SERVICE



NodePort : Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

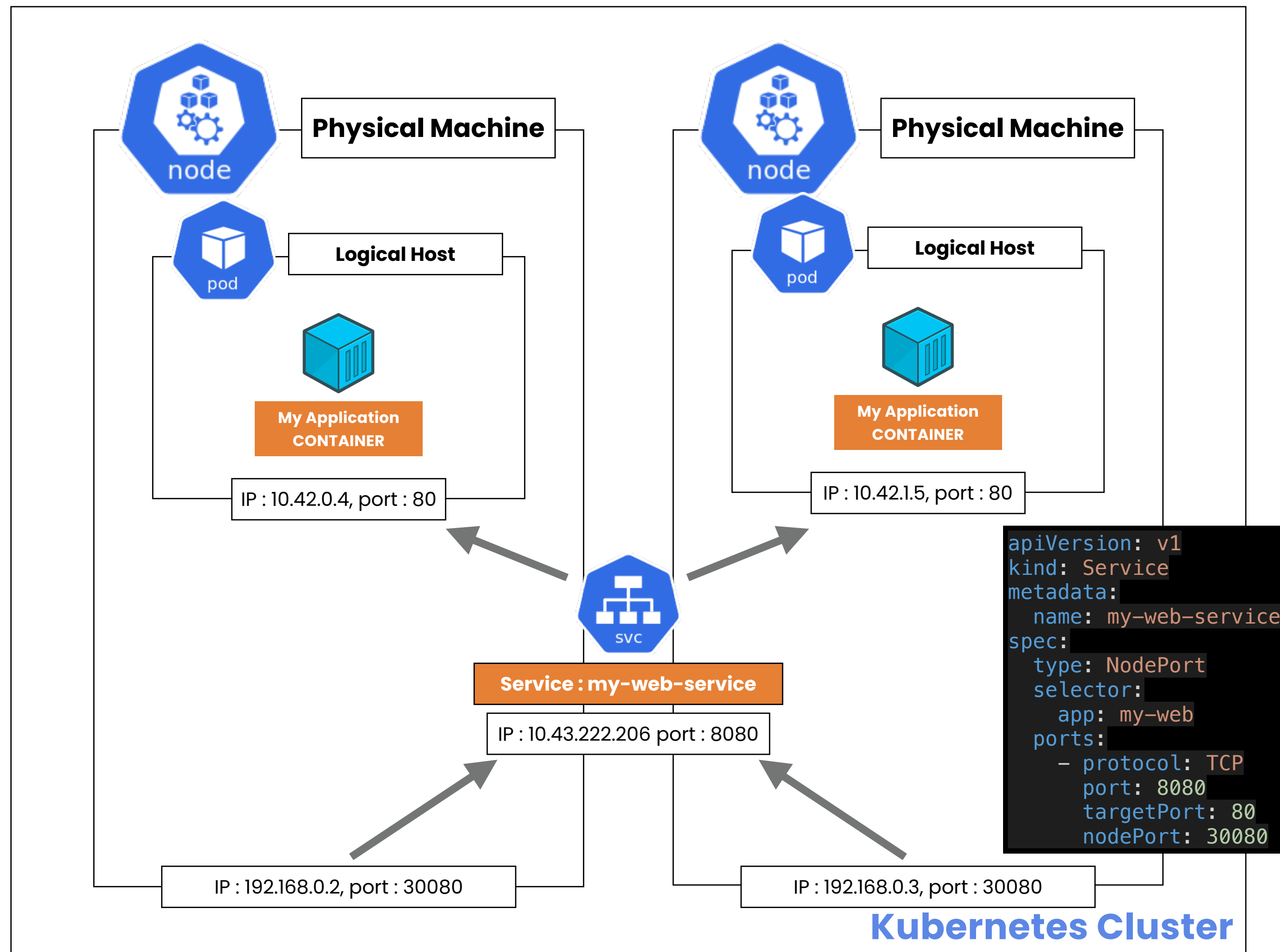
NODEPORT SERVICE



NodePort : Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

NODEPORT SERVICE

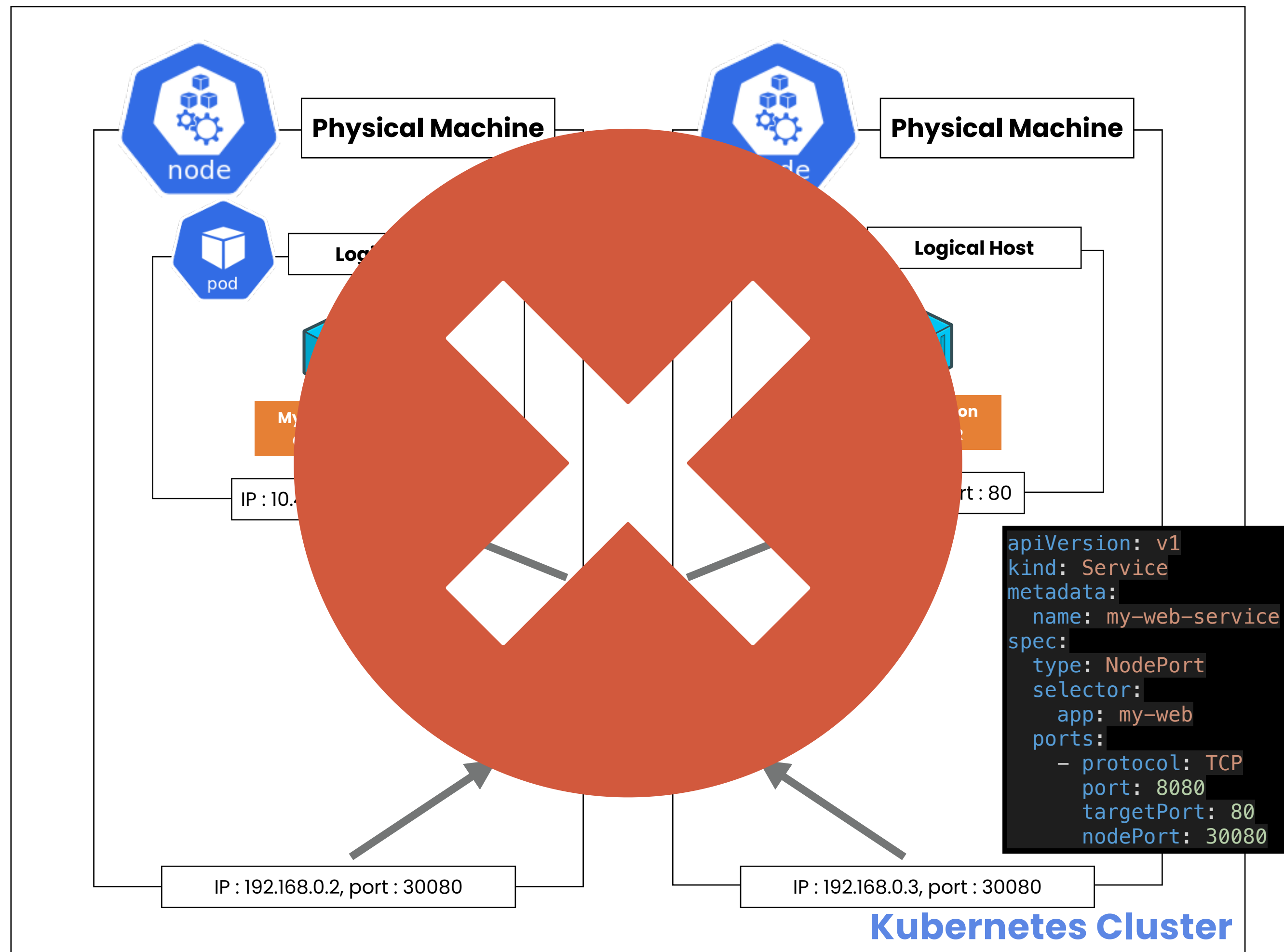


NodePort : Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

ALLOWED 30,000 - 32767

NODEPORT SERVICE



NodePort: Exposes the Service on each Node's IP at a static port (the *NodePort*).

A *ClusterIP* Service, to which the *NodePort* Service routes, is automatically created. You'll be able to contact the NodePort Service, from outside the cluster, by requesting **<NodeIP>:<NodePort>**.

ALLOWED 30,000 - 32767

LOADBALANCER SERVICES

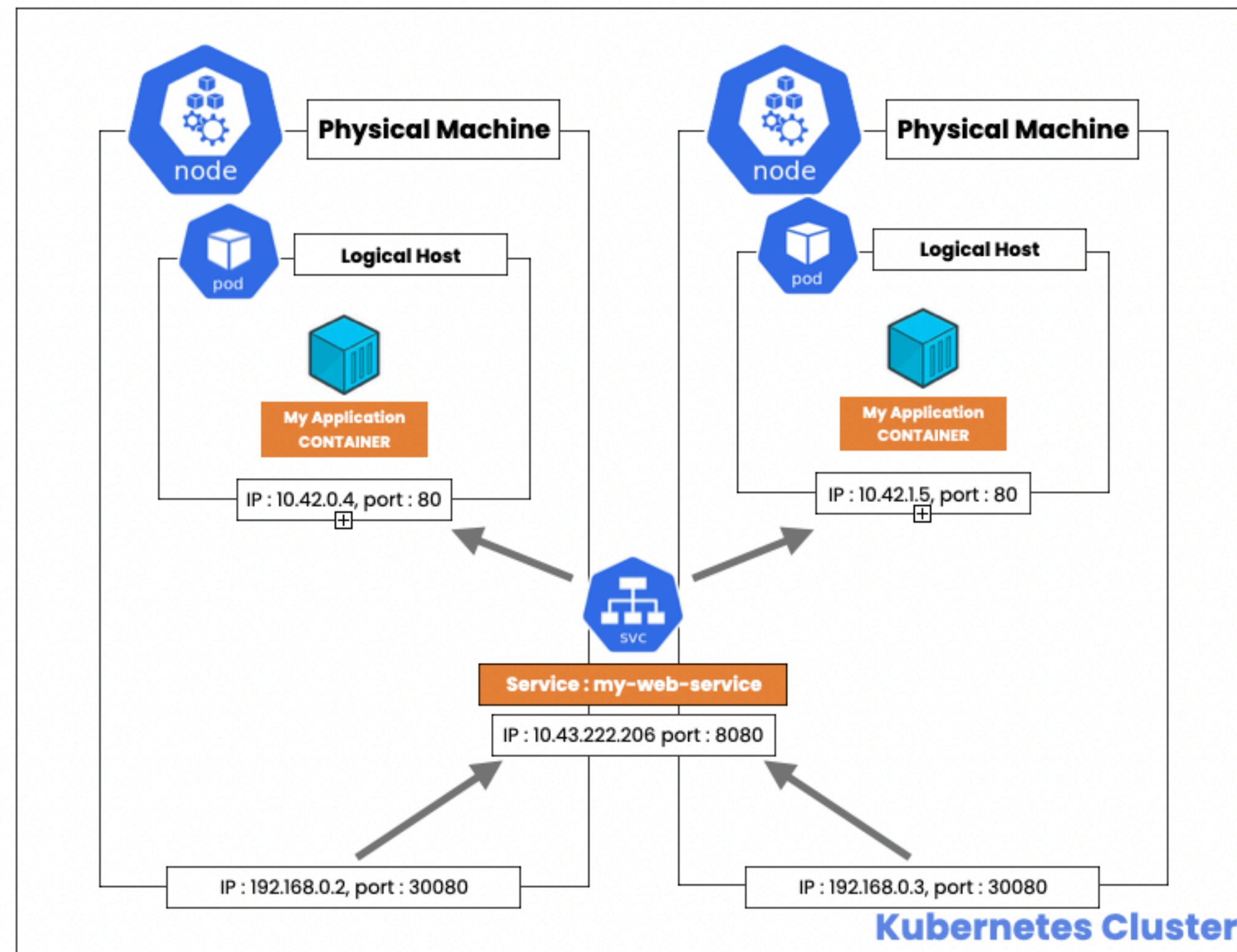
.....

LOADBALANCER SERVICE

LoadBalancer : Exposes the Service externally **using a cloud provider's load balancer**.

NodePort and *ClusterIP* Services, to which the external load balancer routes, are automatically created.

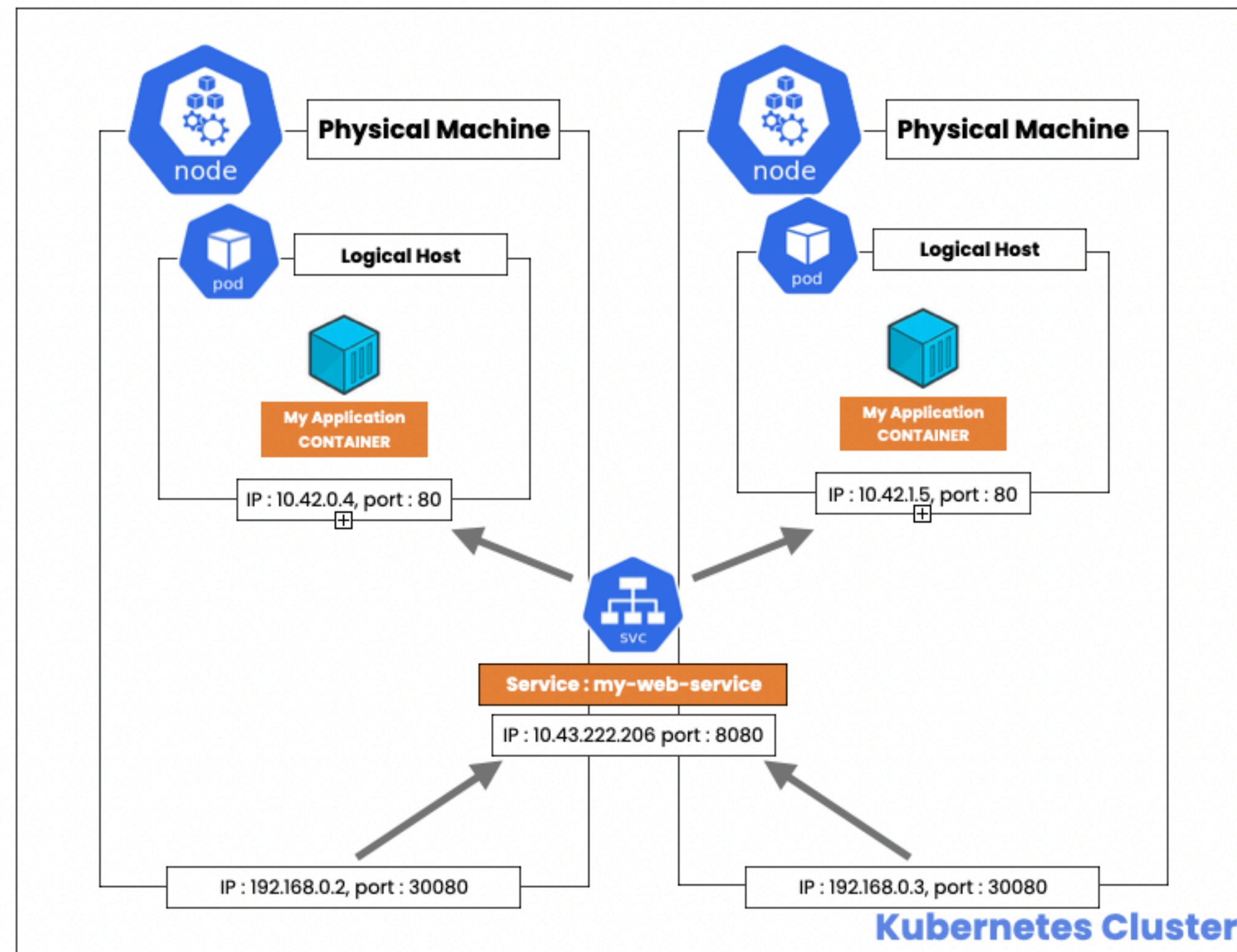
LOADBALANCER SERVICE



LoadBalancer : Exposes the Service externally using a cloud provider's load balancer.

NodePort and *ClusterIP* Services, to which the external load balancer routes, are automatically created.

LOADBALANCER SERVICE

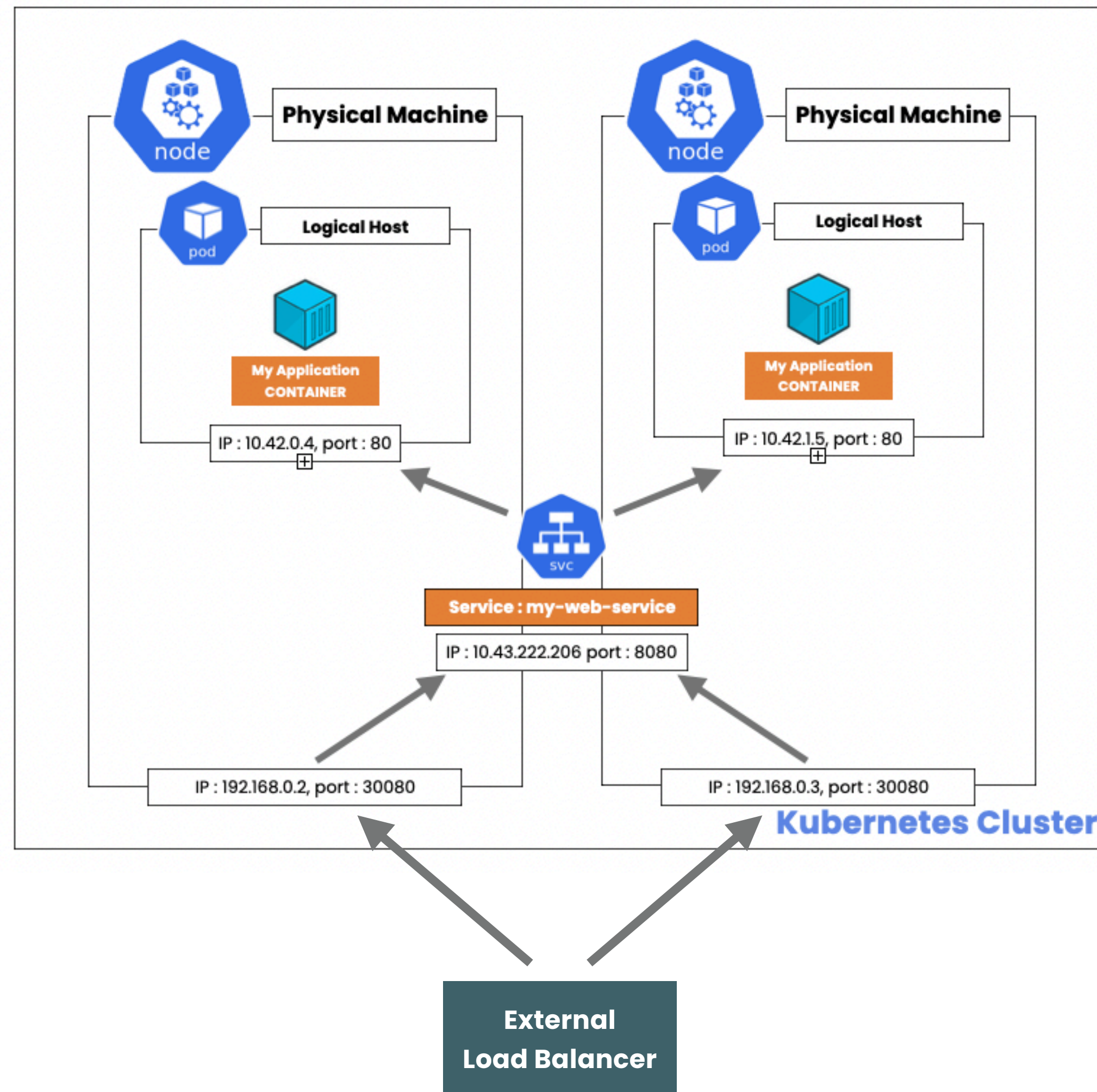


LoadBalancer : Exposes the Service externally using a cloud provider's load balancer.

NodePort and *ClusterIP* Services, to which the external load balancer routes, are automatically created.

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```


LOADBALANCER SERVICE

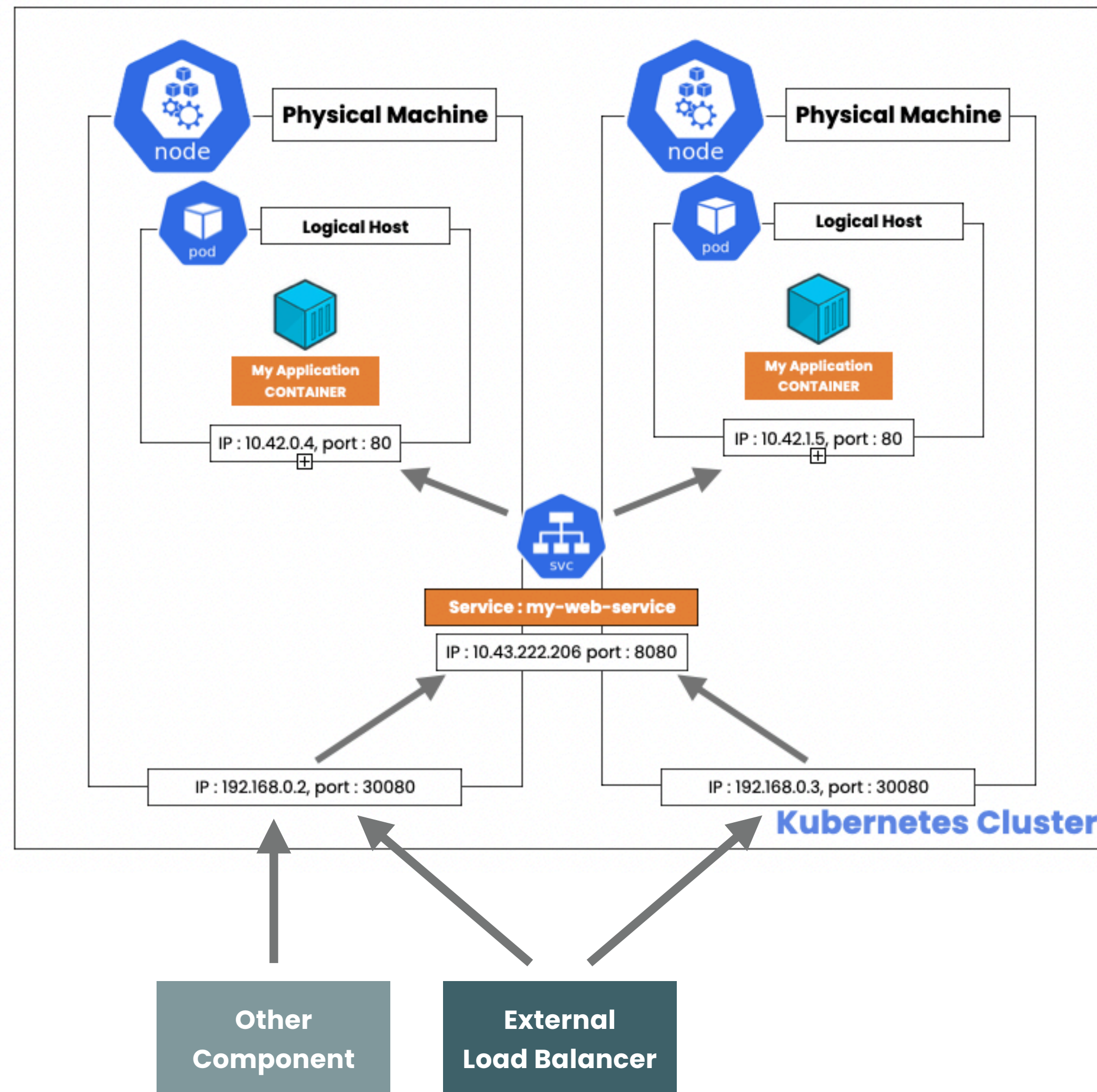


LoadBalancer : Exposes the Service externally using a cloud provider's load balancer.

NodePort and *ClusterIP* Services, to which the external load balancer routes, are automatically created.

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```


LOADBALANCER SERVICE

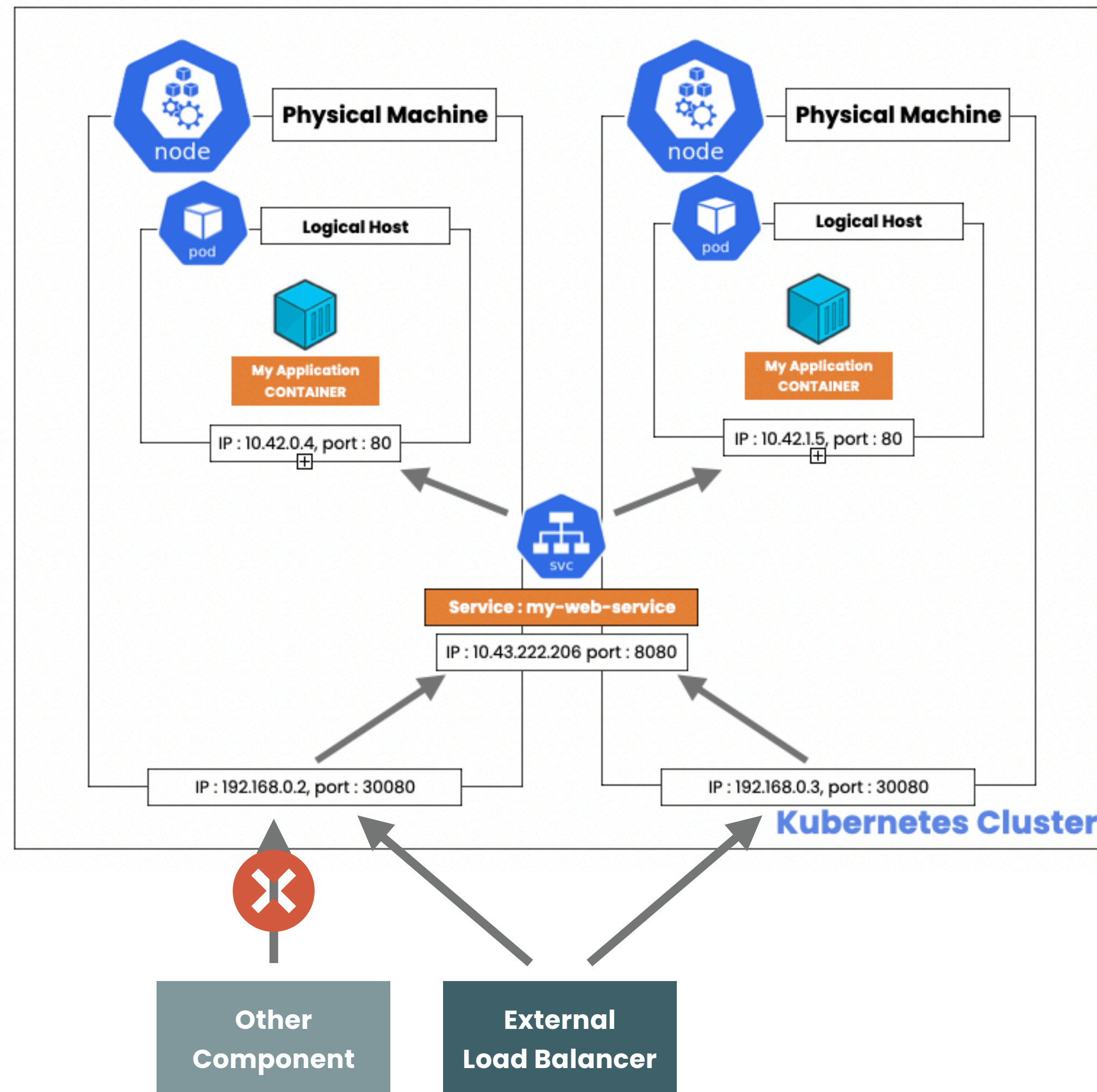


LoadBalancer : Exposes the Service externally using a cloud provider's load balancer.

NodePort and *ClusterIP* Services, to which the external load balancer routes, are automatically created.

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```


LOADBALANCER SERVICE



LoadBalancer : Exposes the Service externally using a cloud provider's load balancer.

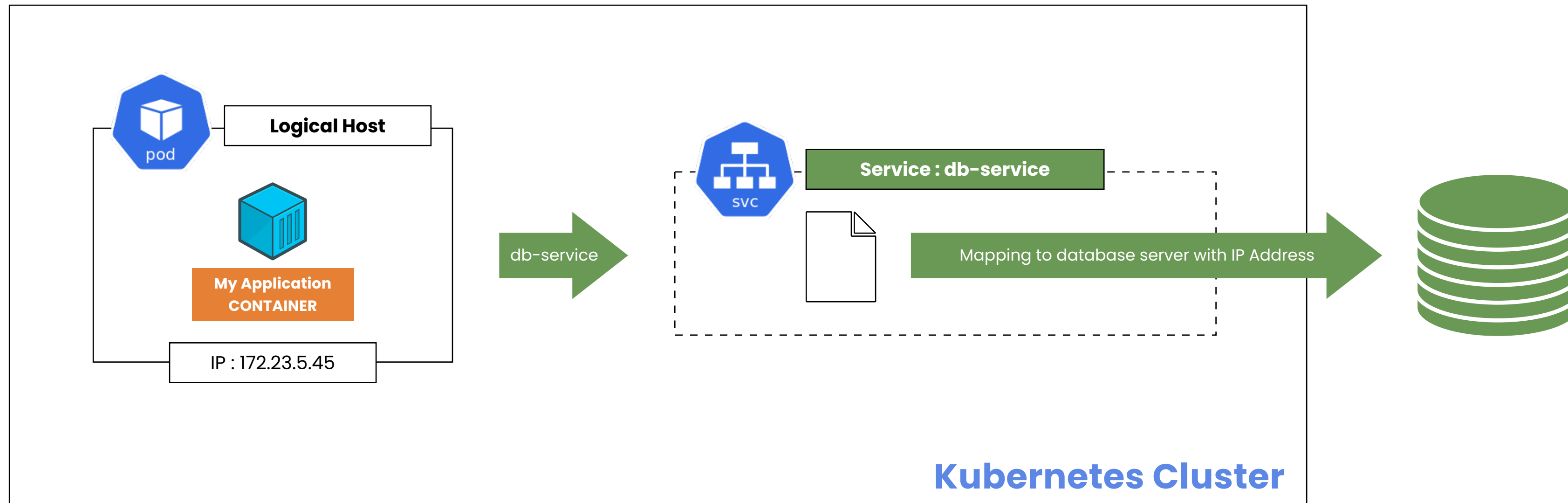
NodePort and *ClusterIP* Services, to which the external load balancer routes, are automatically created.

```
apiVersion: v1
kind: Service
metadata:
  name: my-web-service
spec:
  type: LoadBalancer
  selector:
    app: my-web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30080
```


EXTERNAL NAME/IP

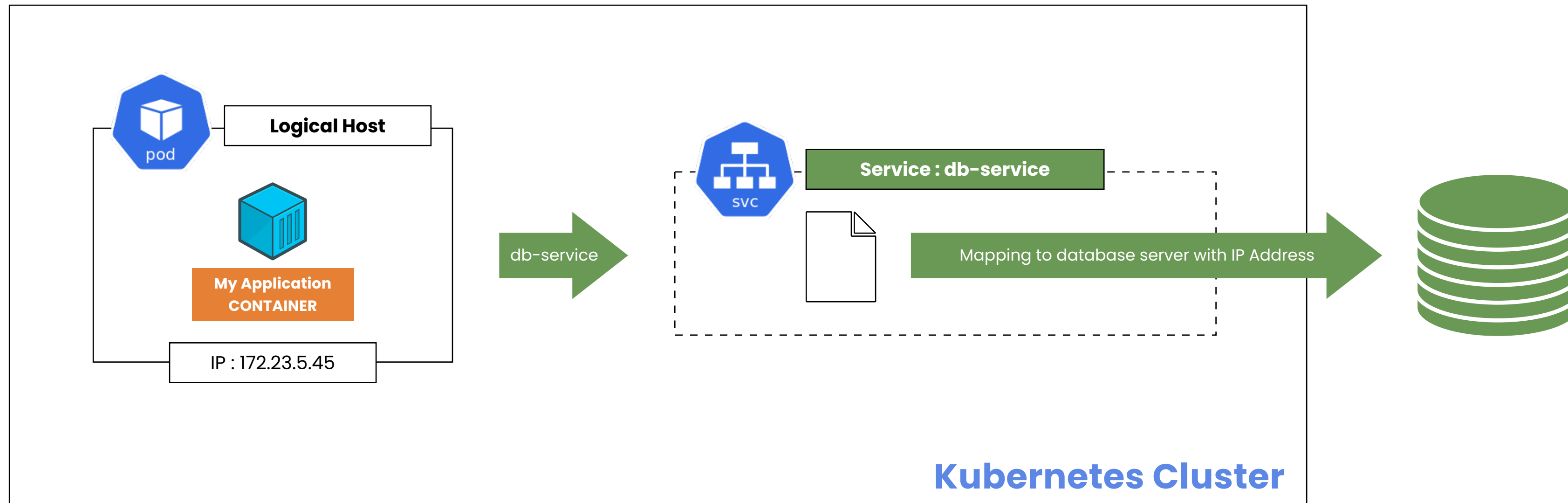
.....

EXTERNAL NAME/IP



ExternalName : Maps the Service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a `CNAME` record with its value. No proxying of any kind is set up.

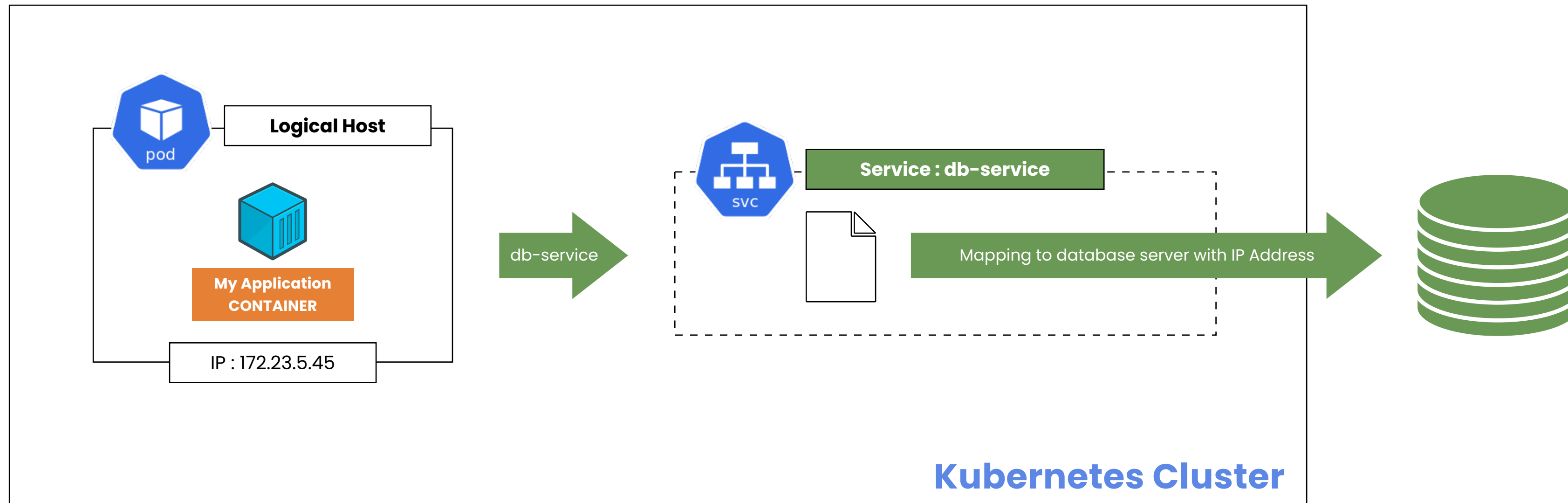
EXTERNAL NAME/IP



ExternalName : Maps the Service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a `CNAME` record with its value. No proxying of any kind is set up.

```
apiVersion: v1
kind: Service
metadata:
  name: ex-json-service
  namespace: prod
spec:
  type: ExternalName
  externalName: jsonplaceholder.typicode.com
```


EXTERNAL NAME/IP



ExternalName : Maps the Service to the contents of the `externalName` field (e.g. `foo.bar.example.com`), by returning a CNAME record with its value. No proxying of any kind is set up.

```
apiVersion: v1
kind: Service
metadata:
  name: ex-json-service
  namespace: prod
spec:
  type: ExternalName
  externalName: jsonplaceholder.typicode.com
```

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: 9376
  externalIPs:
    - 80.11.12.10
```


LET TRY !

THANK YOU