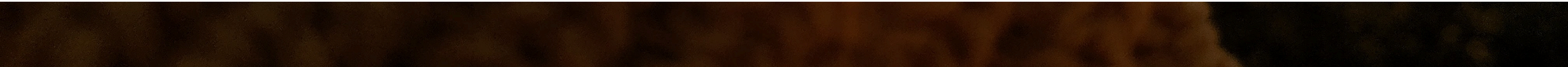
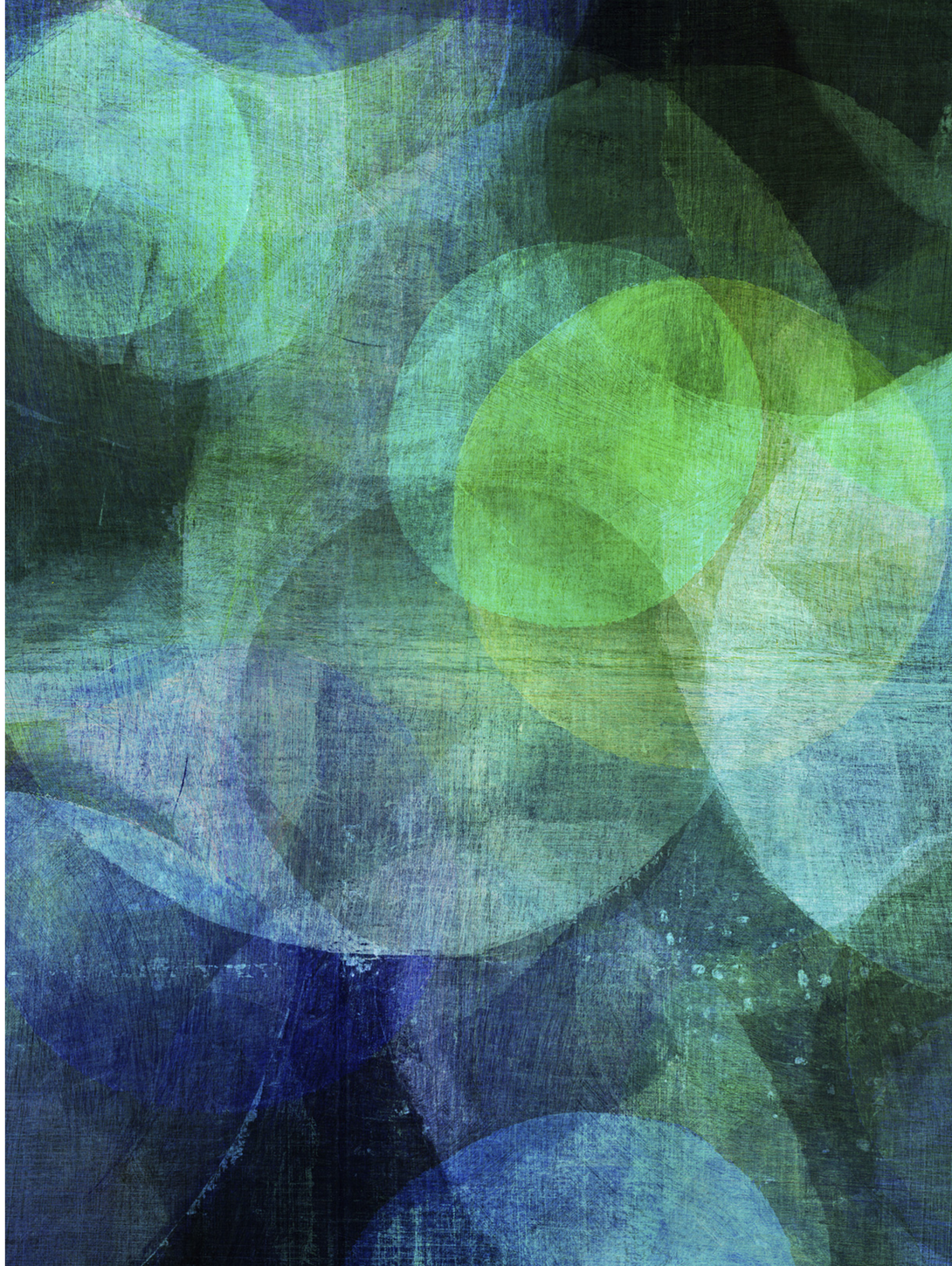
INTRODUCTION TO KUBERNETES

INTRODUCTION TO KUBERNETES

Kubernetes Services

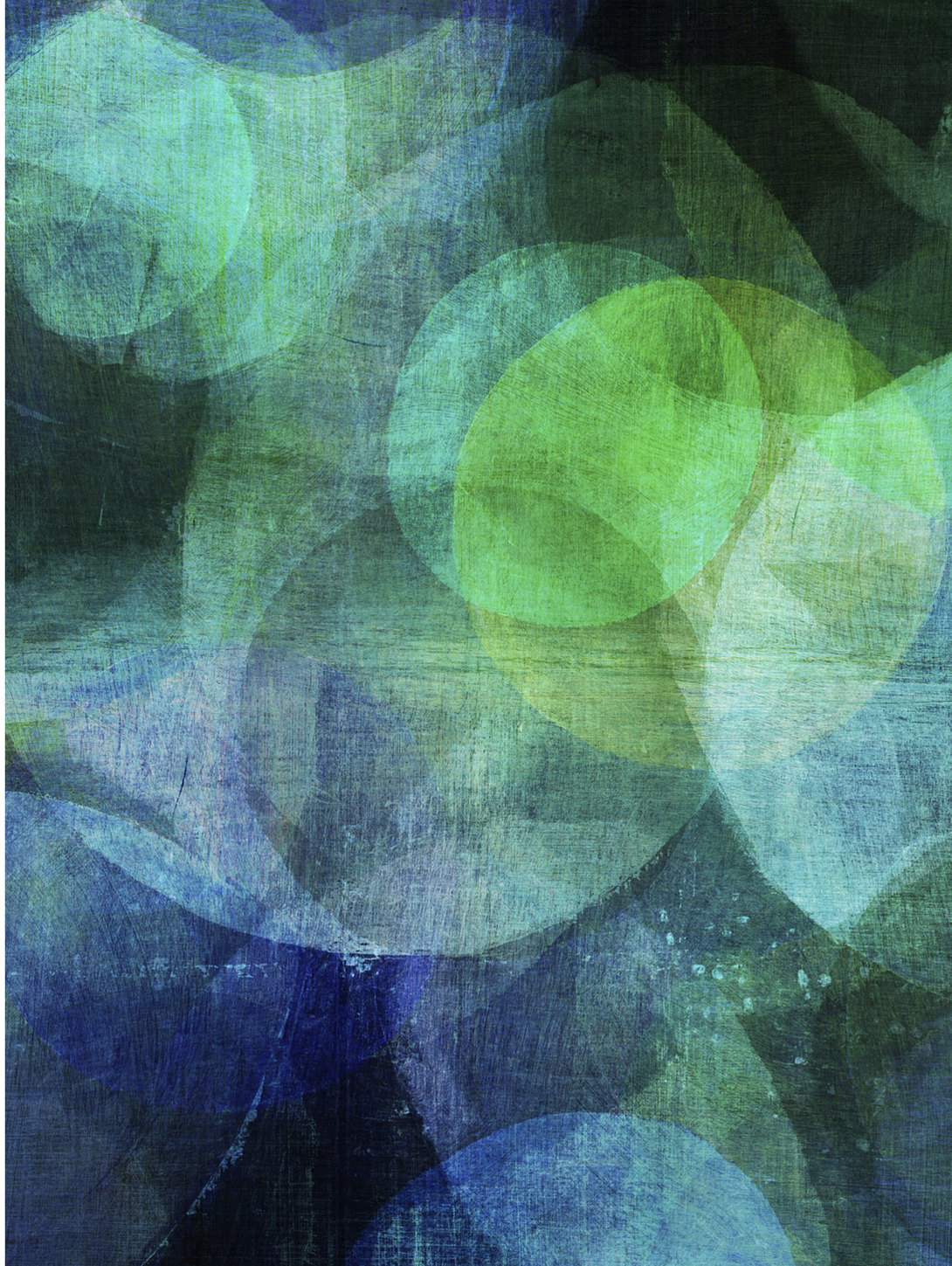


AGENDA



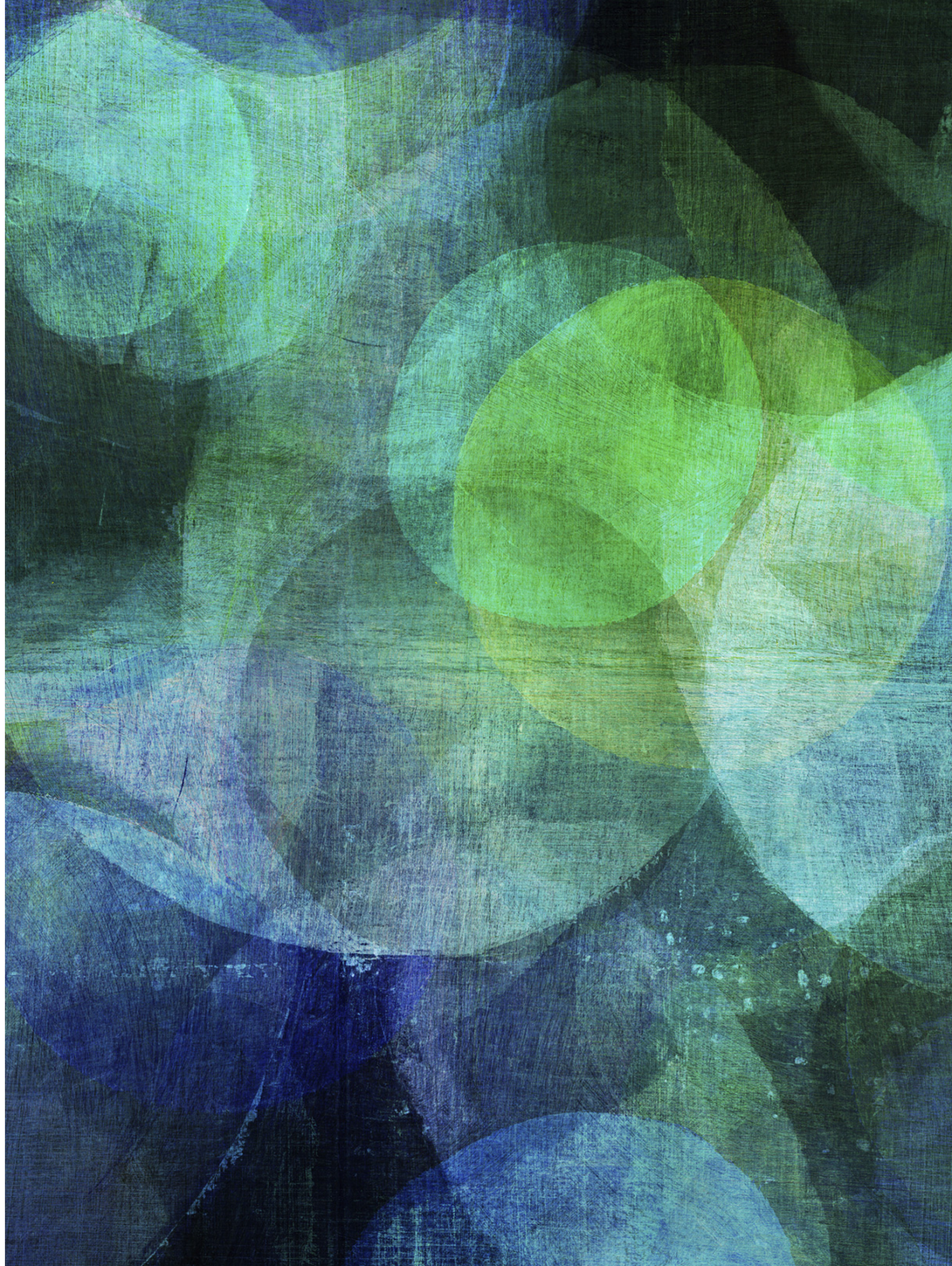
AGENDA

- What/Why Service



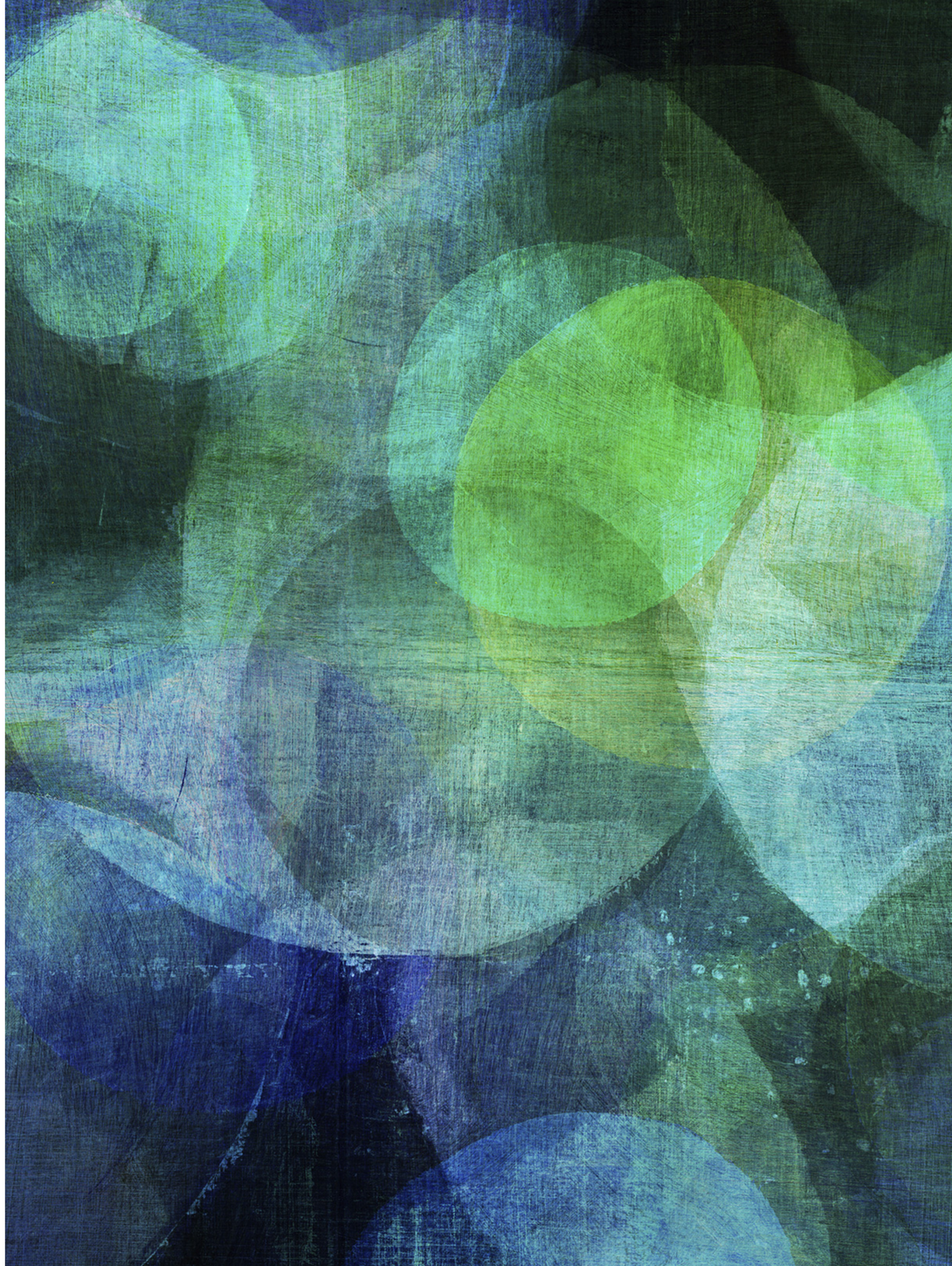
AGENDA

- What/Why Service
- Cluster IP



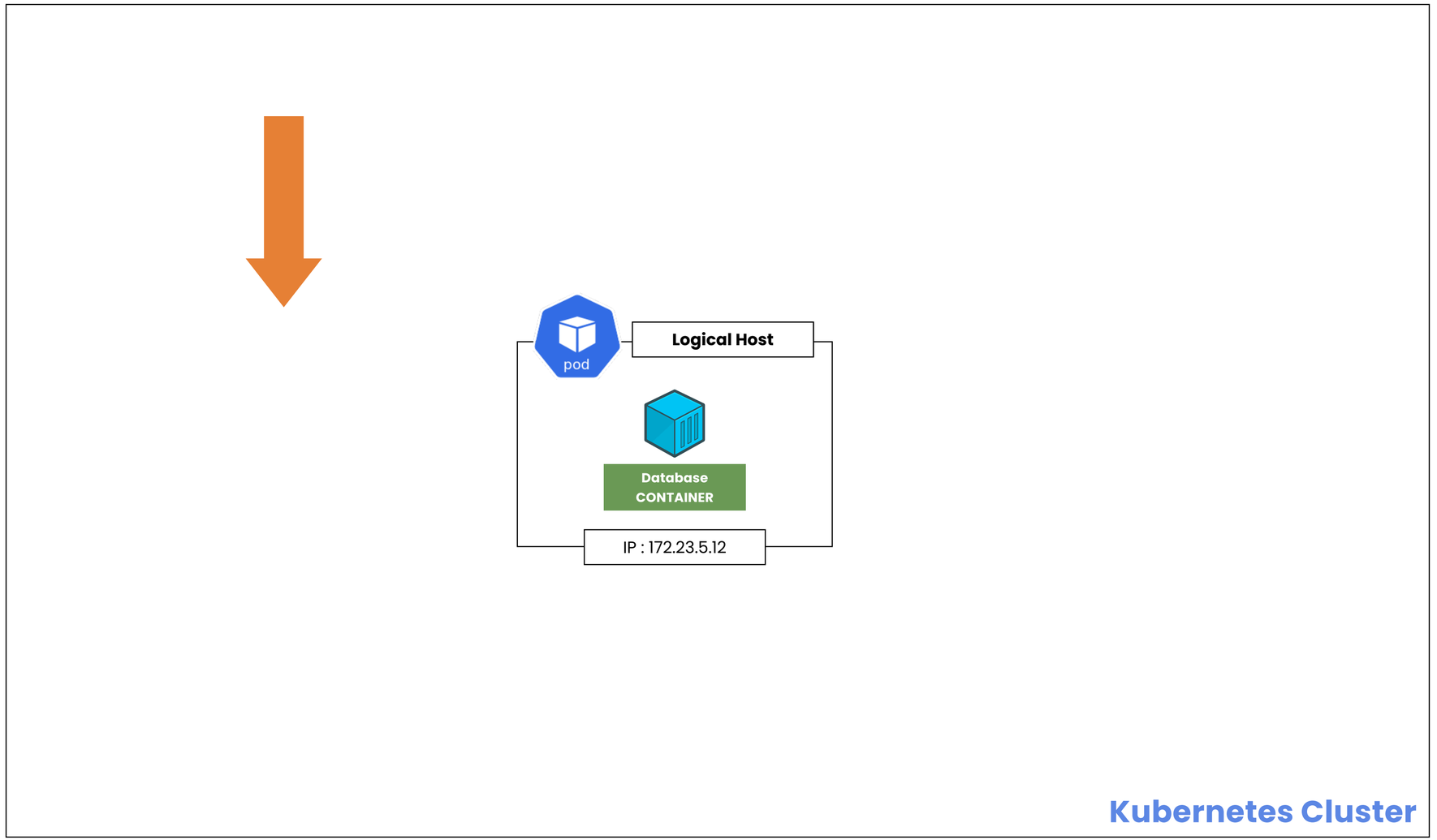
AGENDA

- What/Why Service
- Cluster IP
- Headless Service

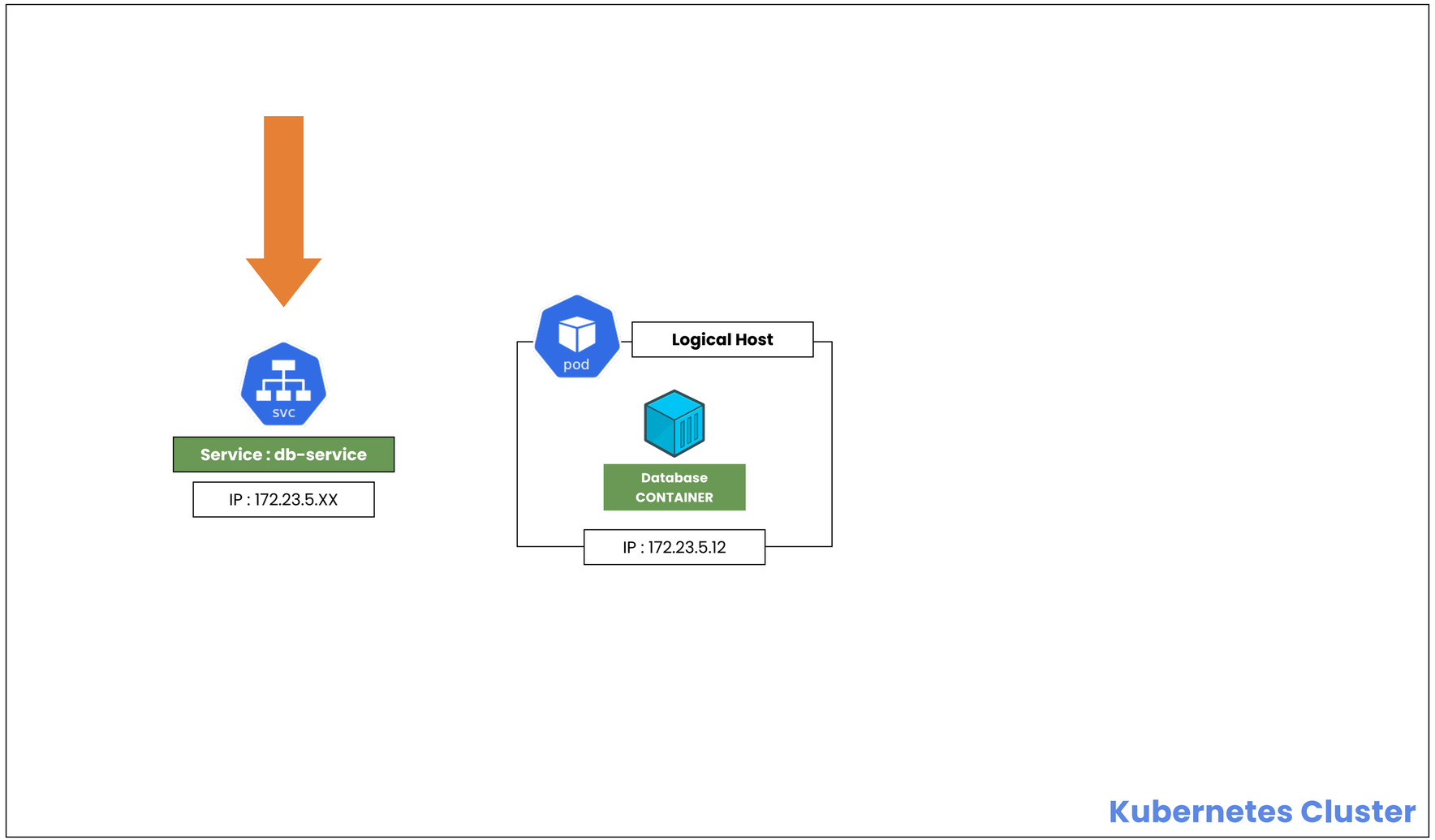


WHAT/WHY SERVICE

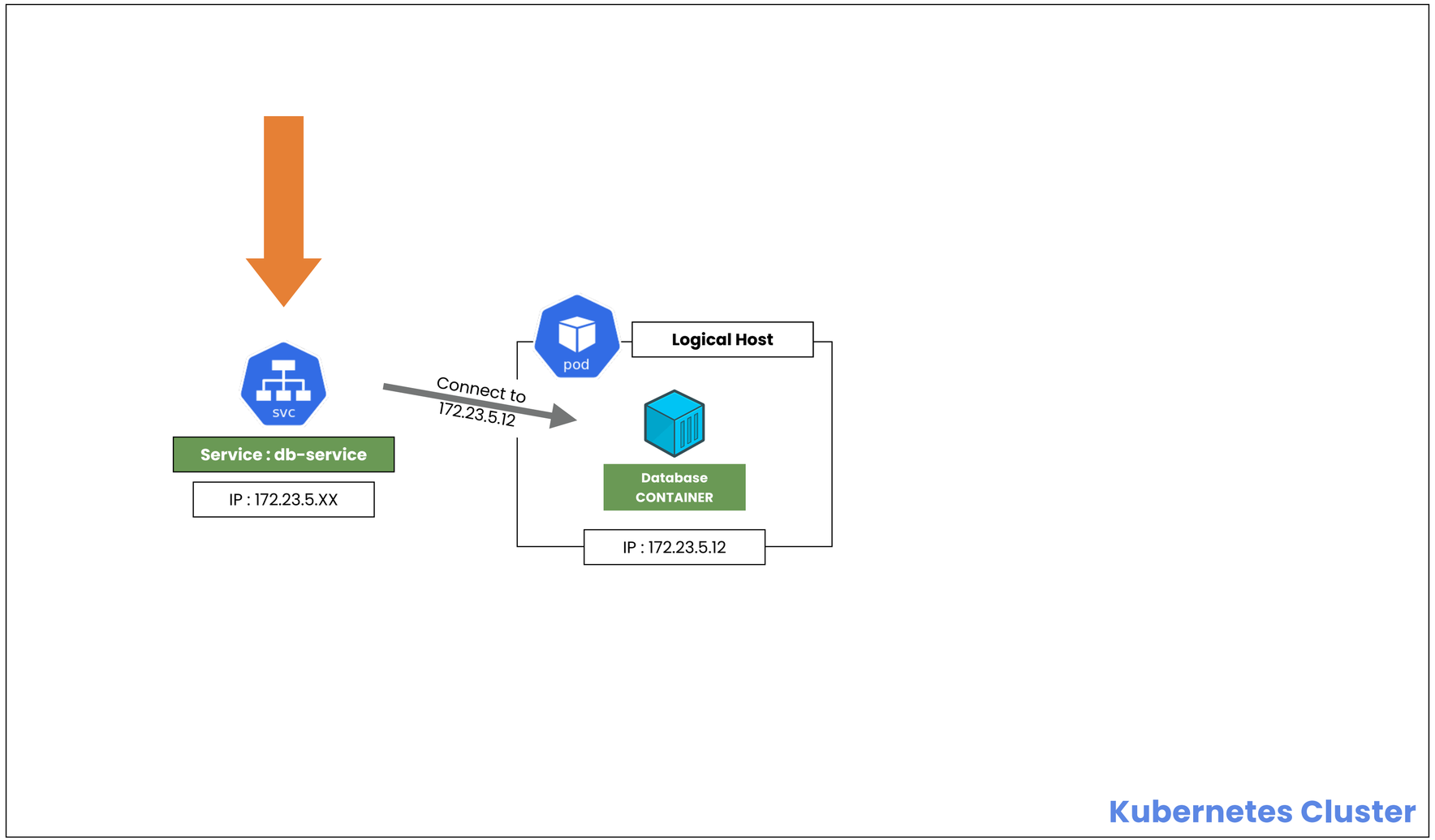
SERVICE - STABLE IP



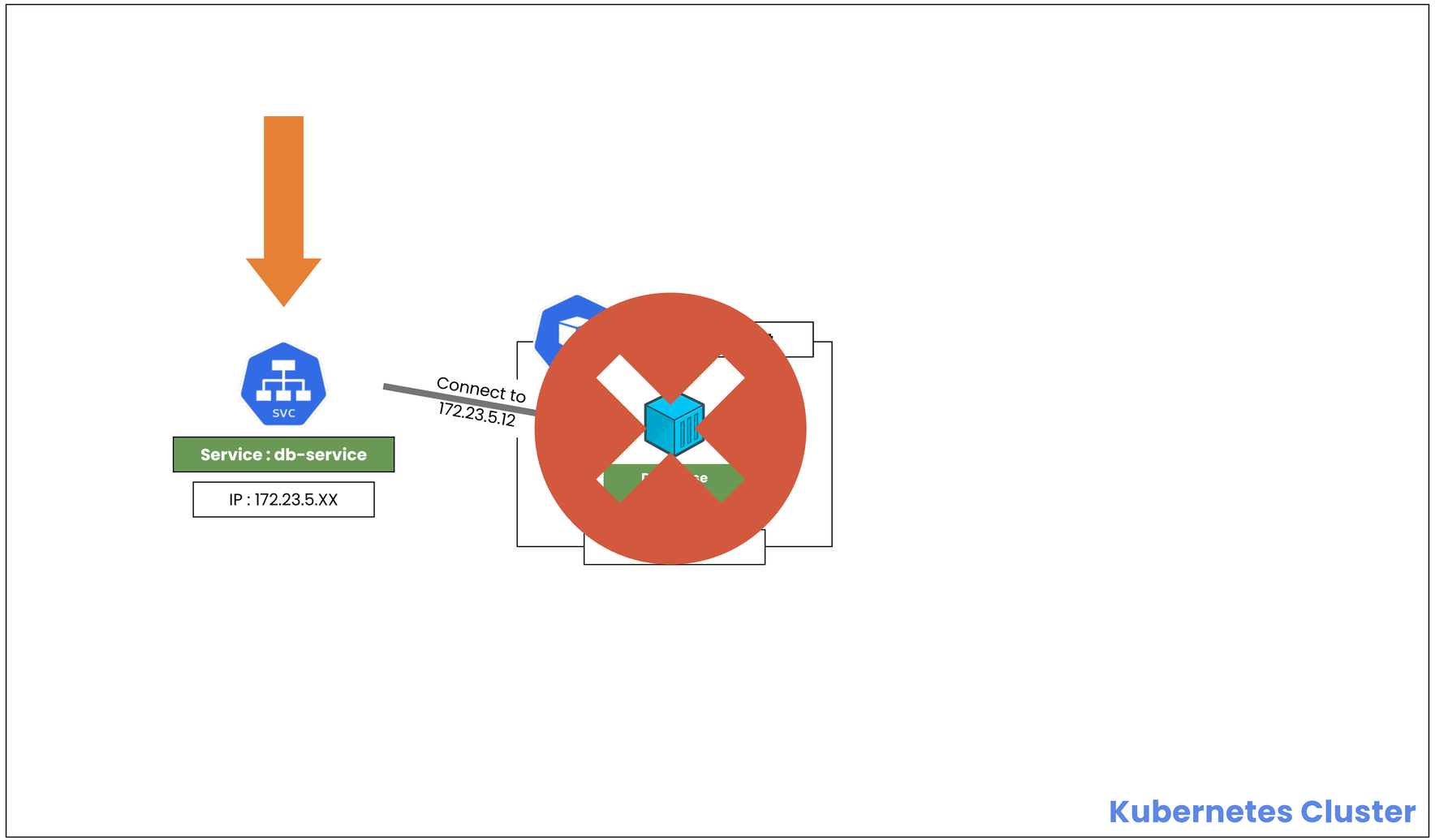
SERVICE - STABLE IP



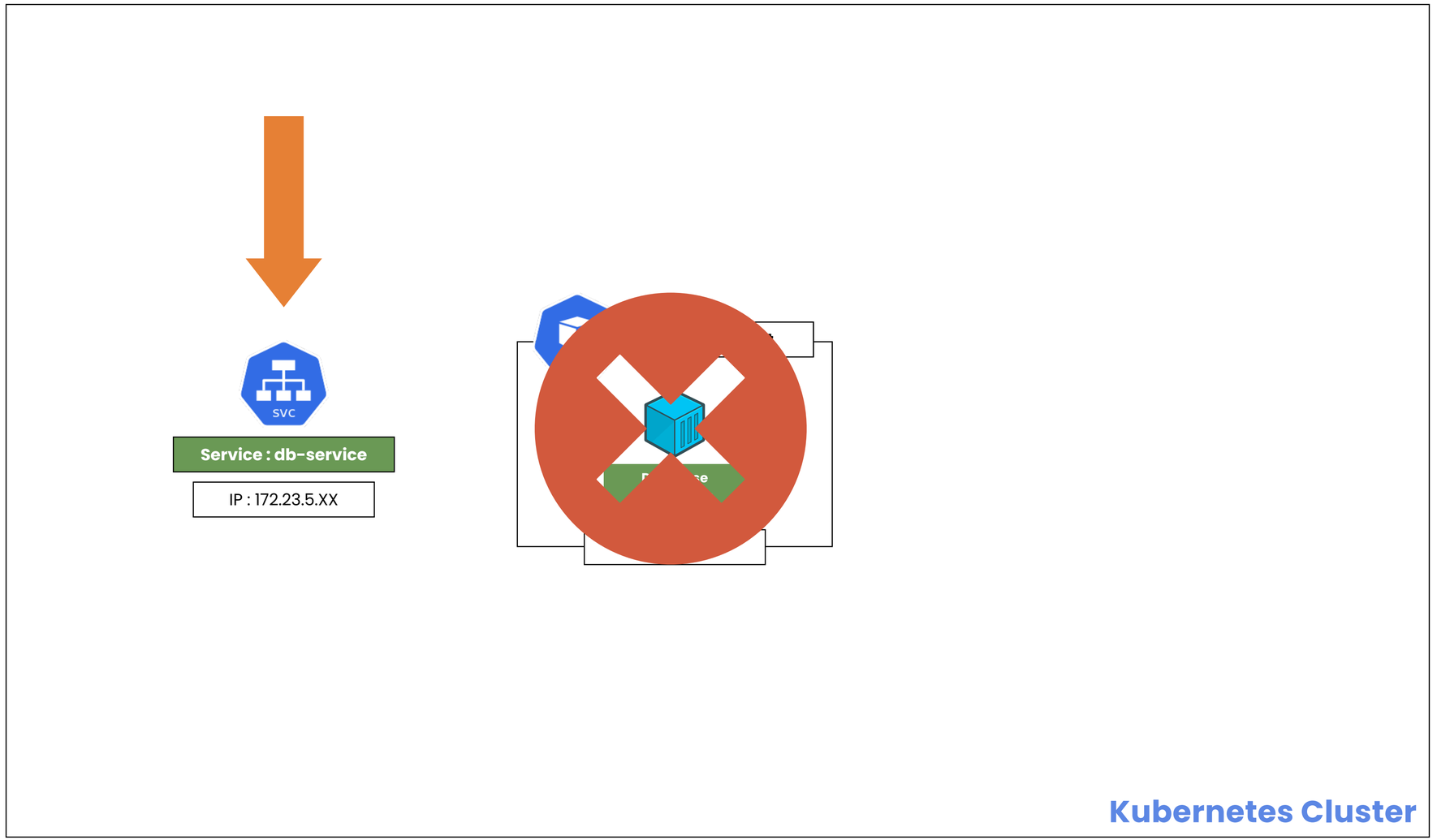
SERVICE - STABLE IP



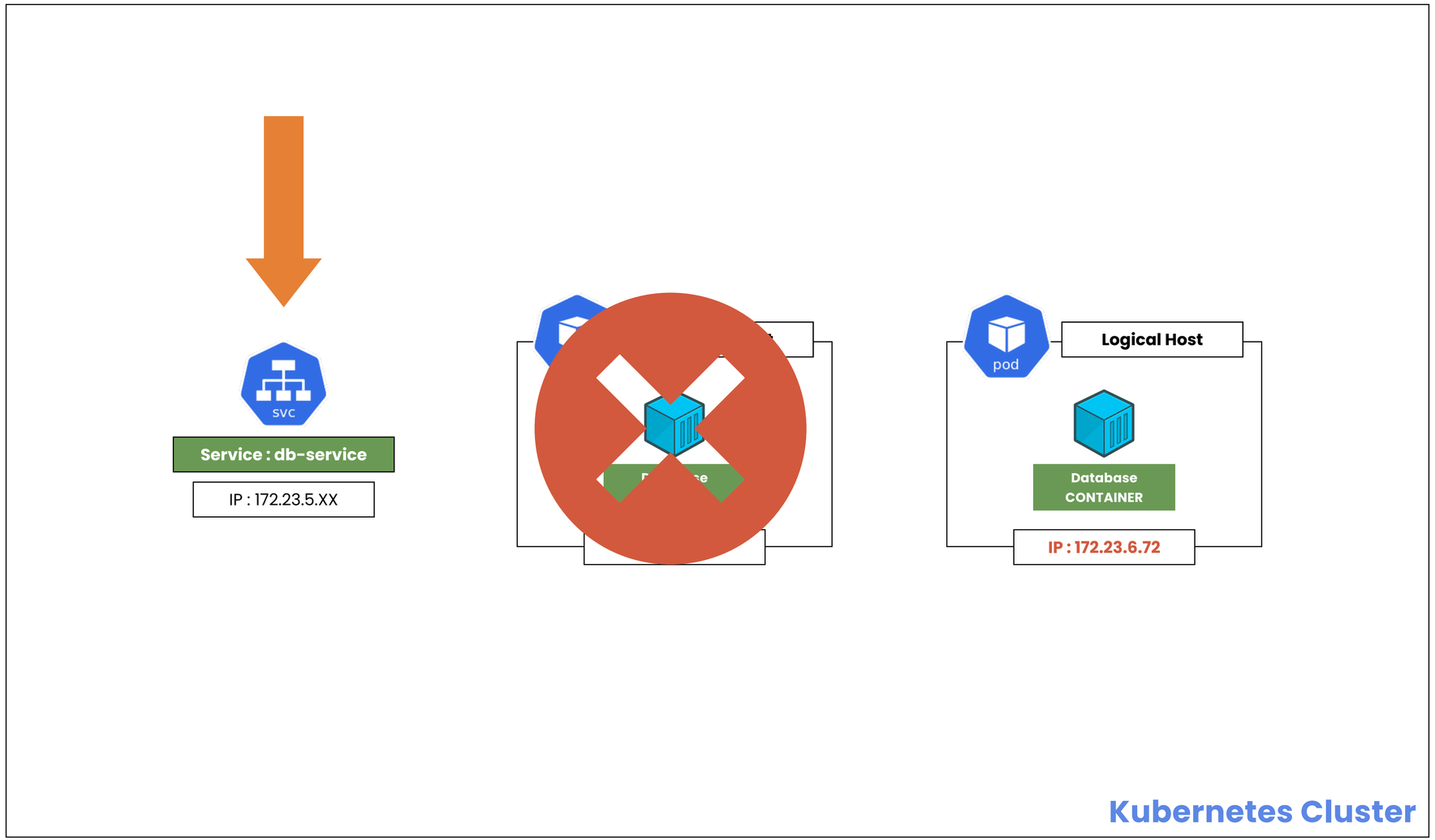
SERVICE - STABLE IP



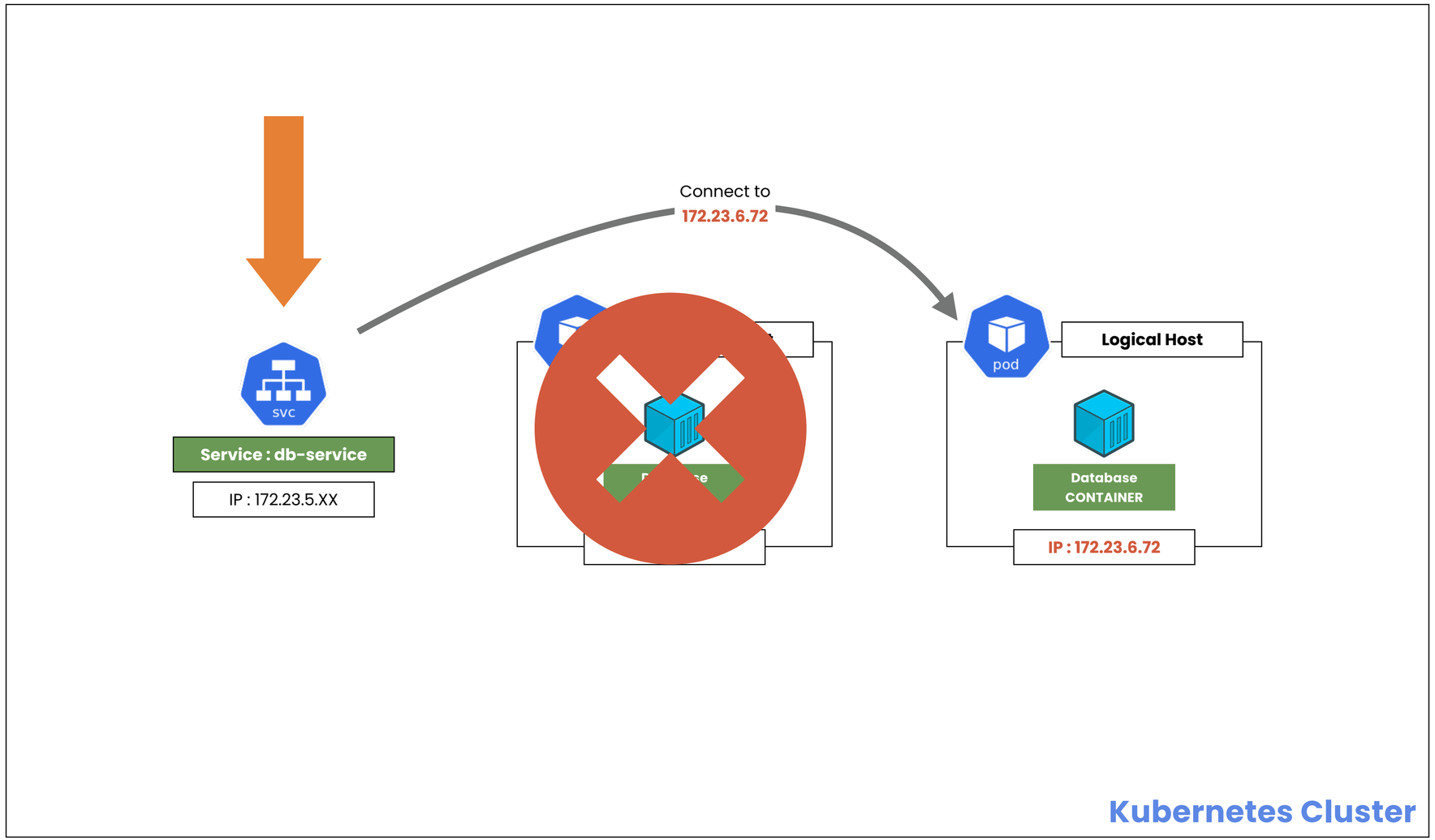
SERVICE - STABLE IP



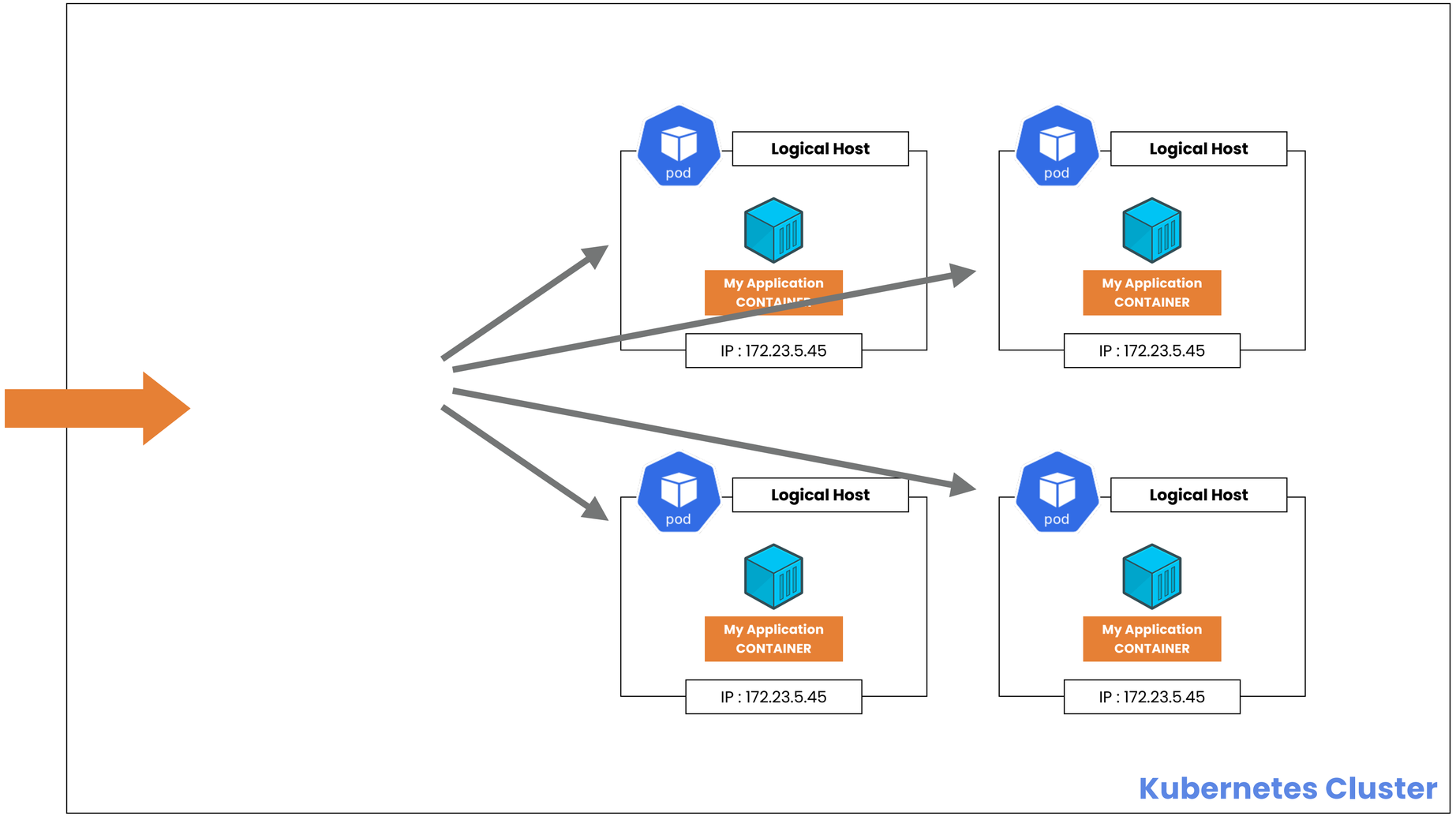
SERVICE - STABLE IP



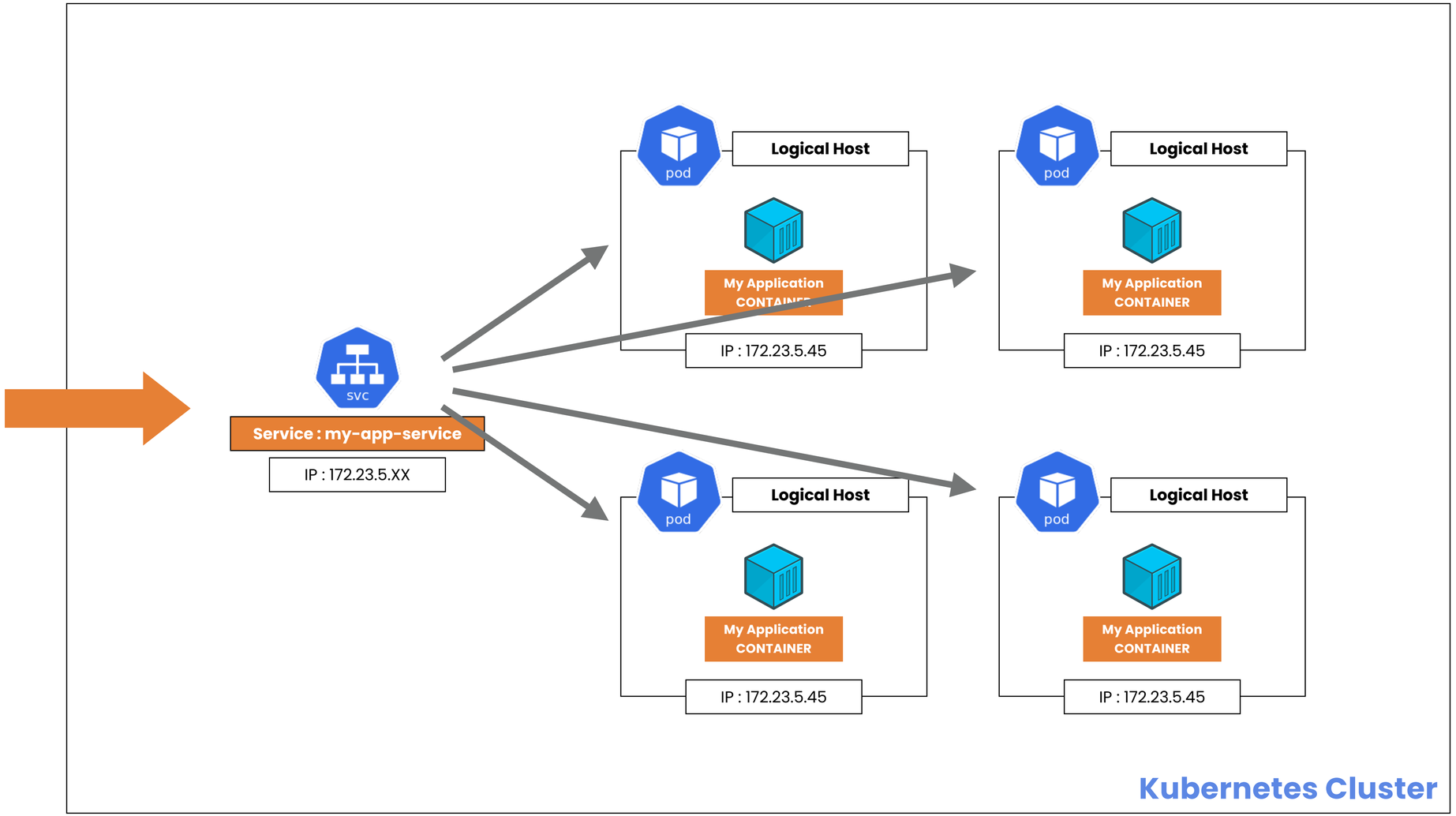
SERVICE - STABLE IP



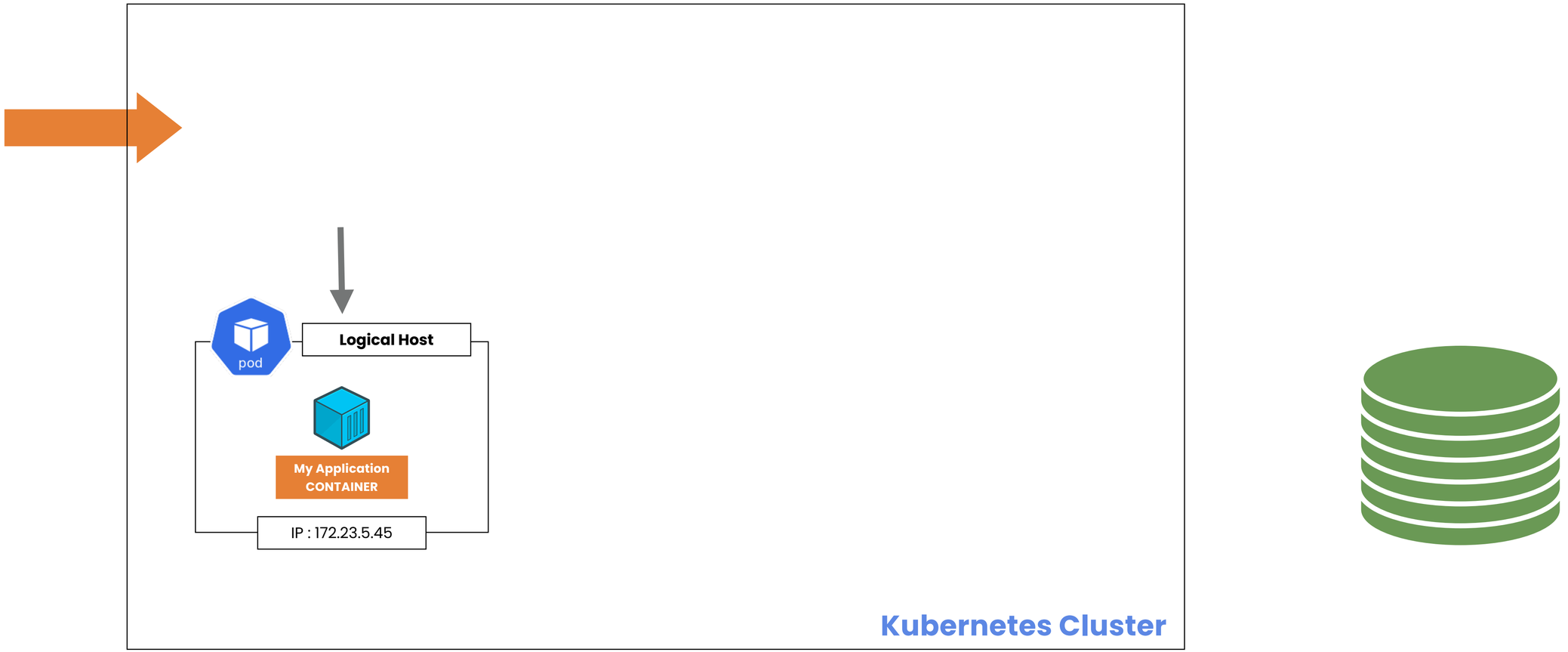
SERVICE - LOADBALANCER



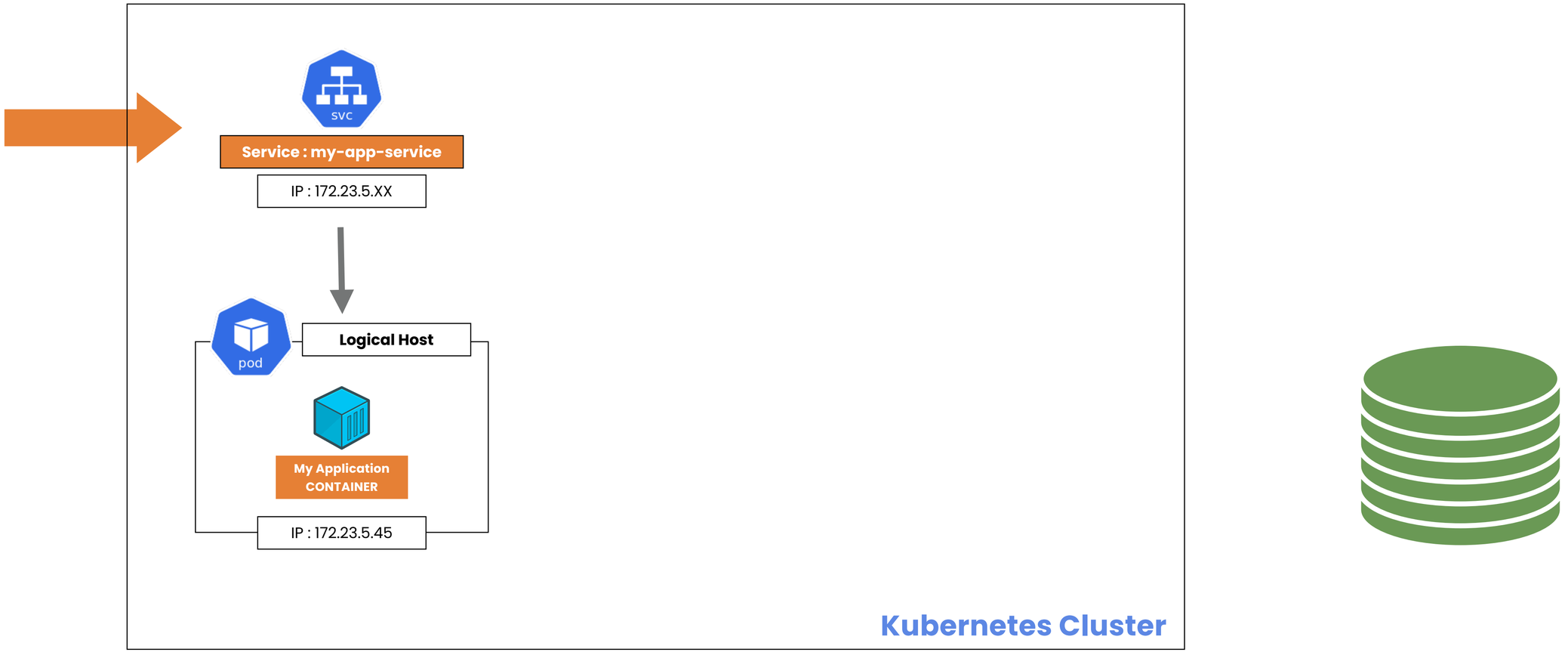
SERVICE - LOADBALANCER



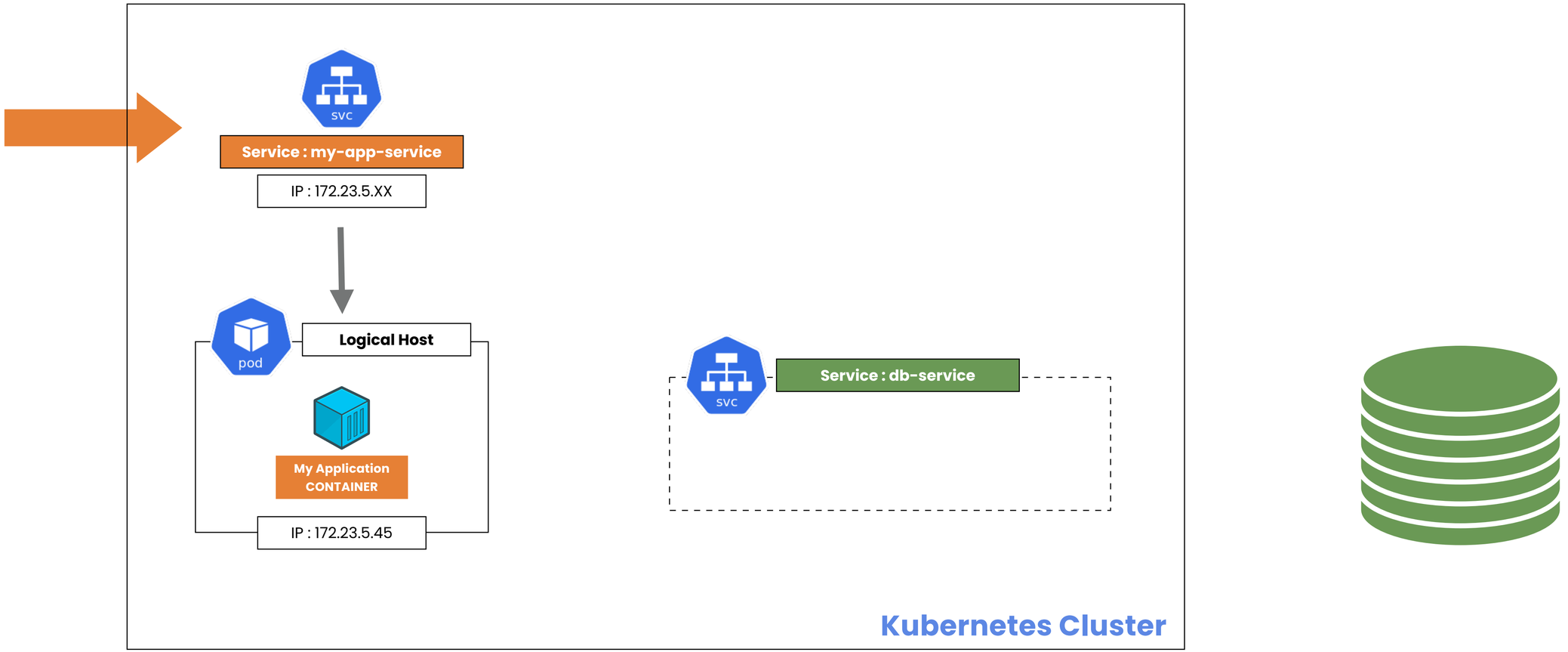
SERVICE - USING FOR INSIDE AND OUTSIDE OF CLUSTER



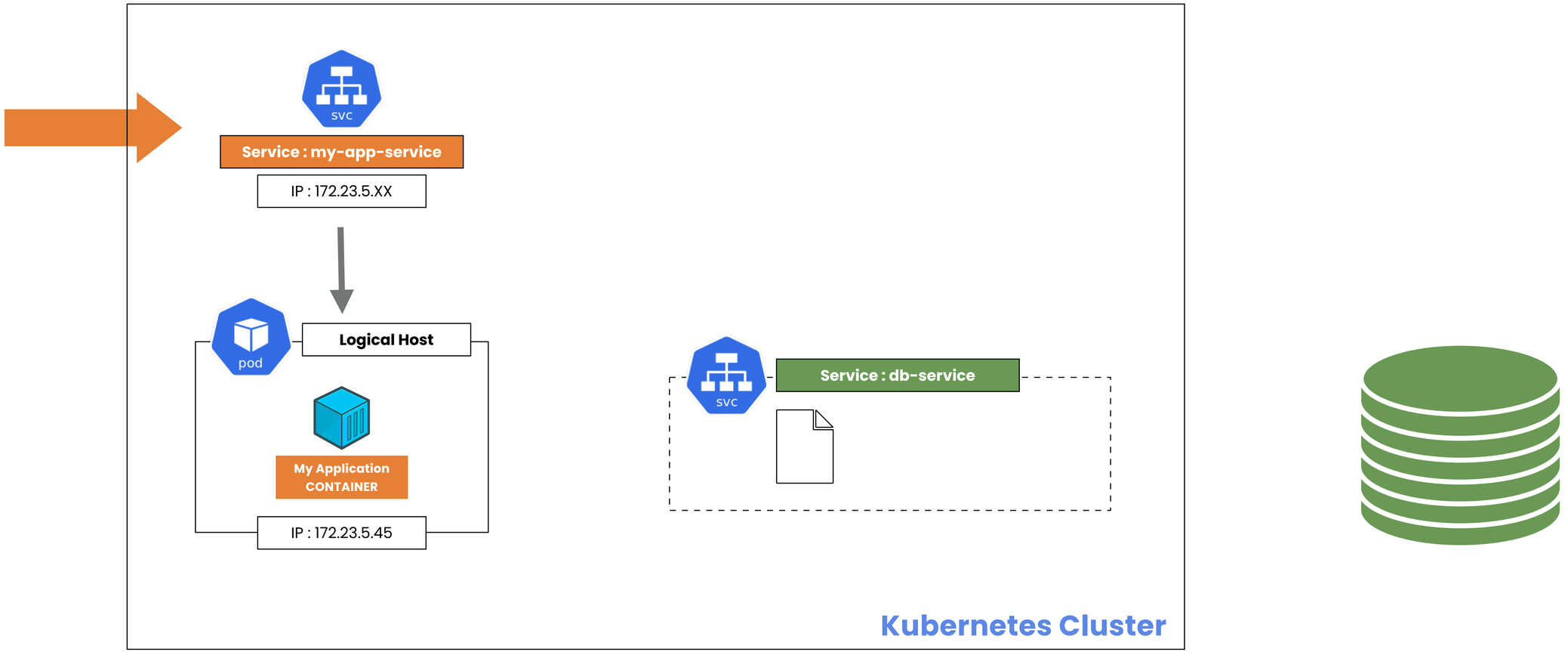
SERVICE - USING FOR INSIDE AND OUTSIDE OF CLUSTER



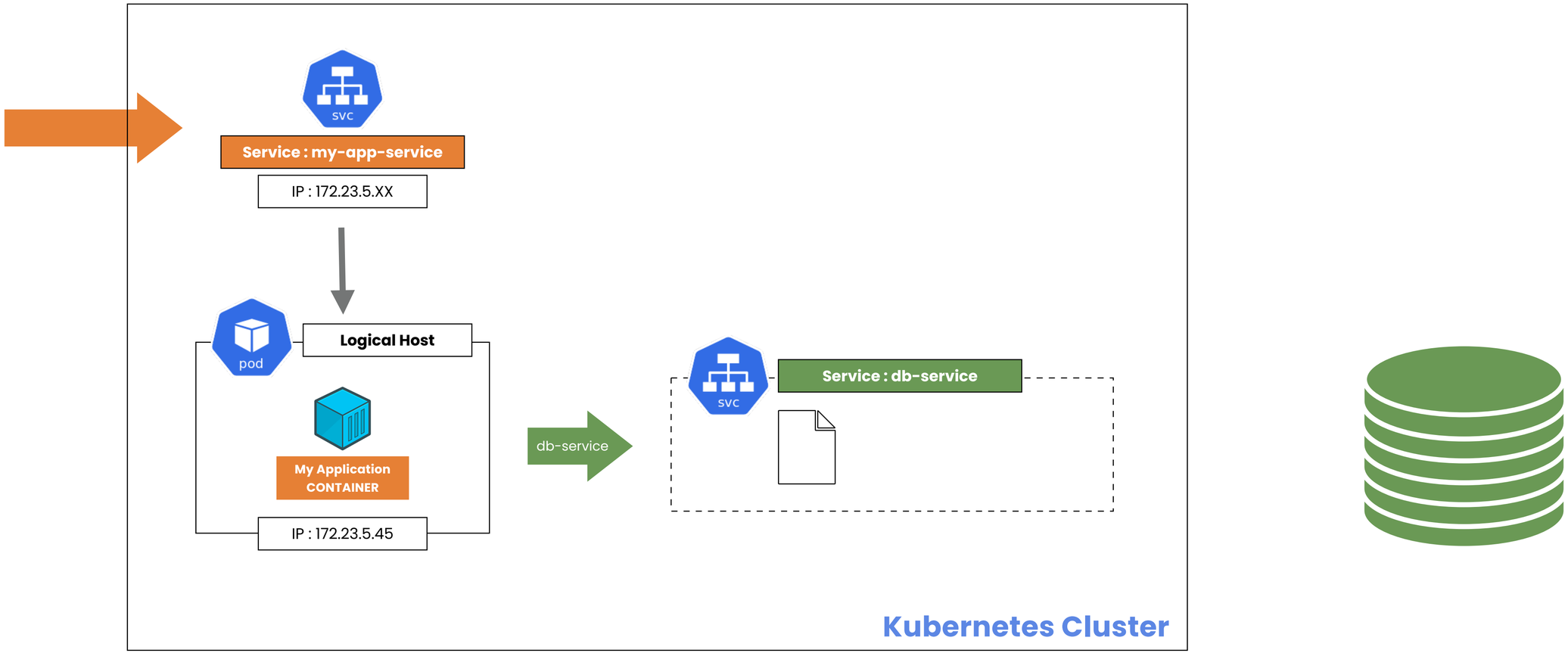
SERVICE - USING FOR INSIDE AND OUTSIDE OF CLUSTER



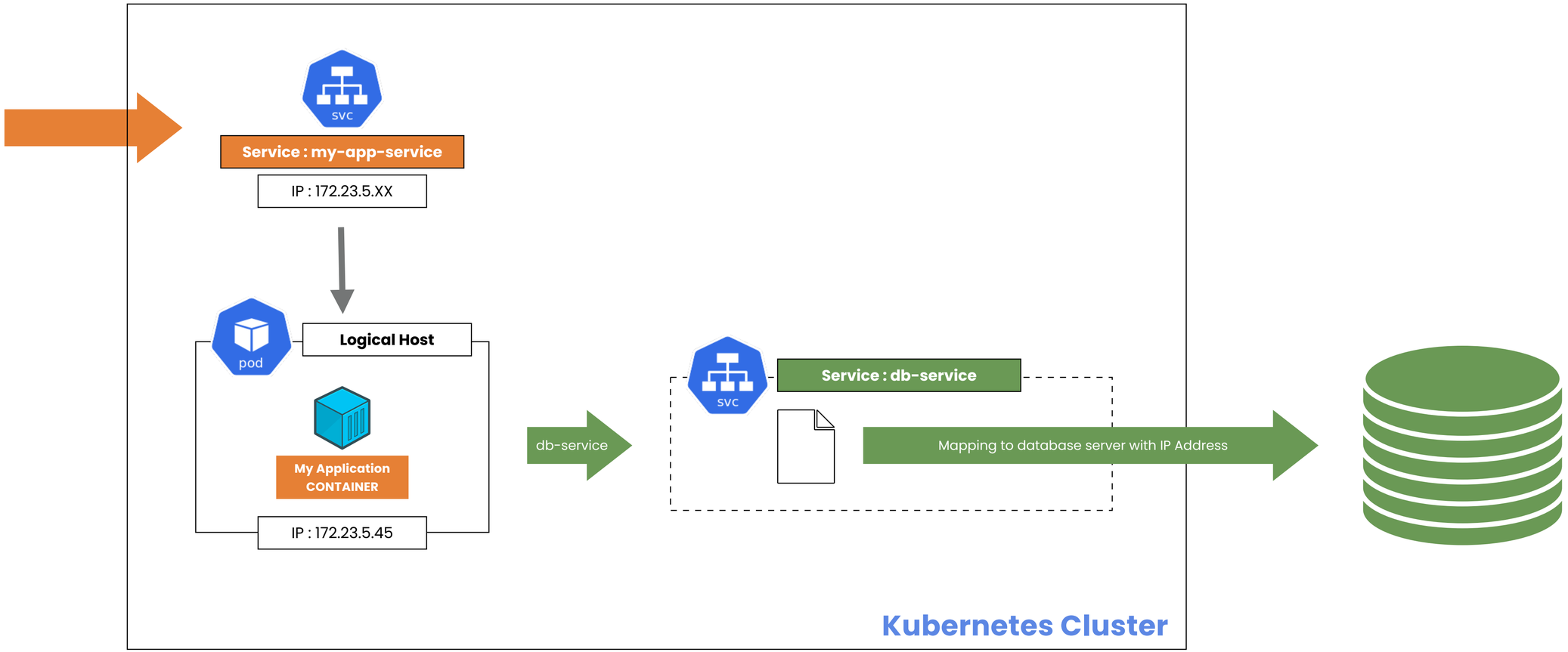
SERVICE - USING FOR INSIDE AND OUTSIDE OF CLUSTER



SERVICE - USING FOR INSIDE AND OUTSIDE OF CLUSTER



SERVICE - USING FOR INSIDE AND OUTSIDE OF CLUSTER



CLUSTERIP SERVICES

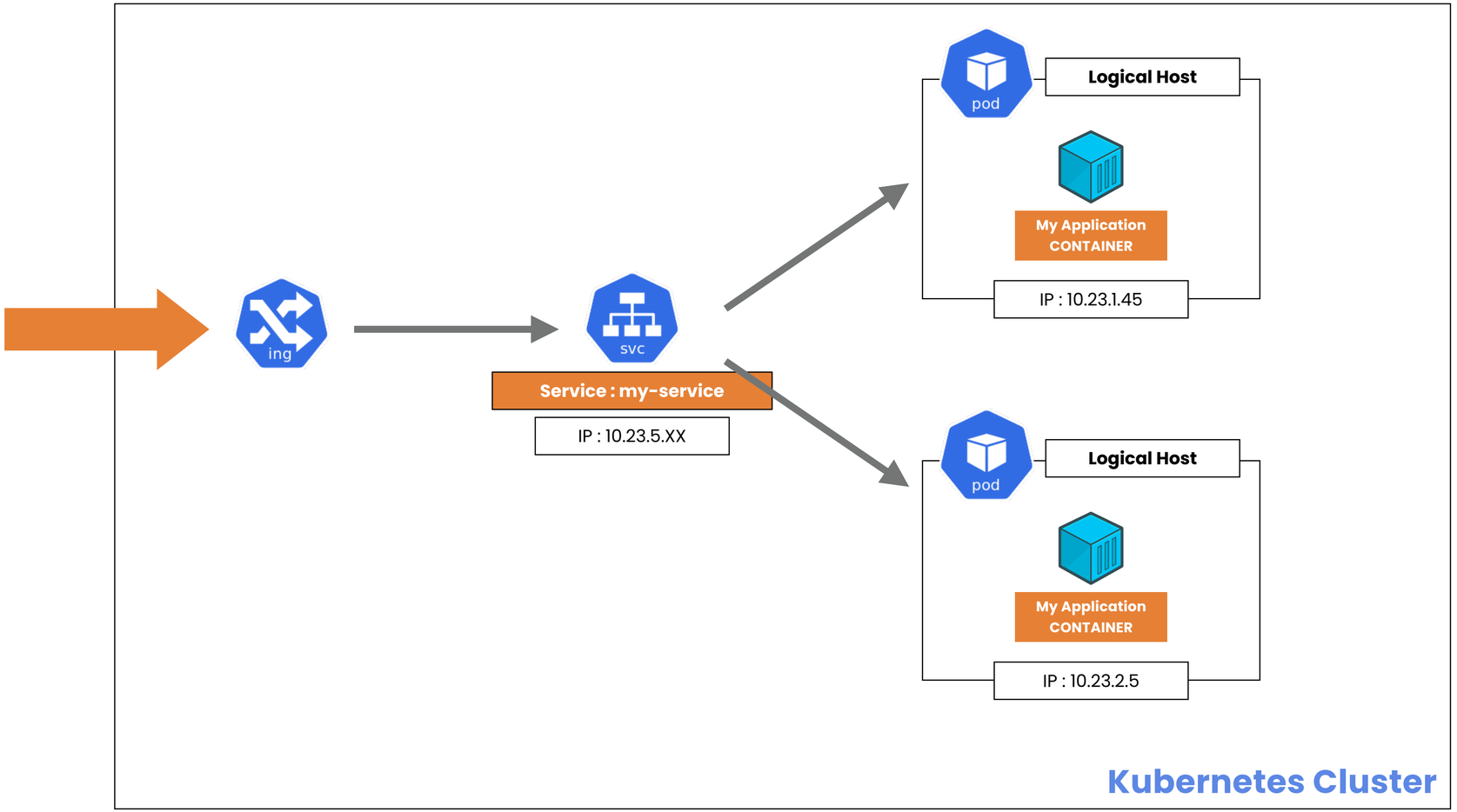
CLUSTERIP SERVICE

ClusterIP : Exposes the Service on a cluster-internal IP. Choosing this value makes the Service only reachable from within the cluster.

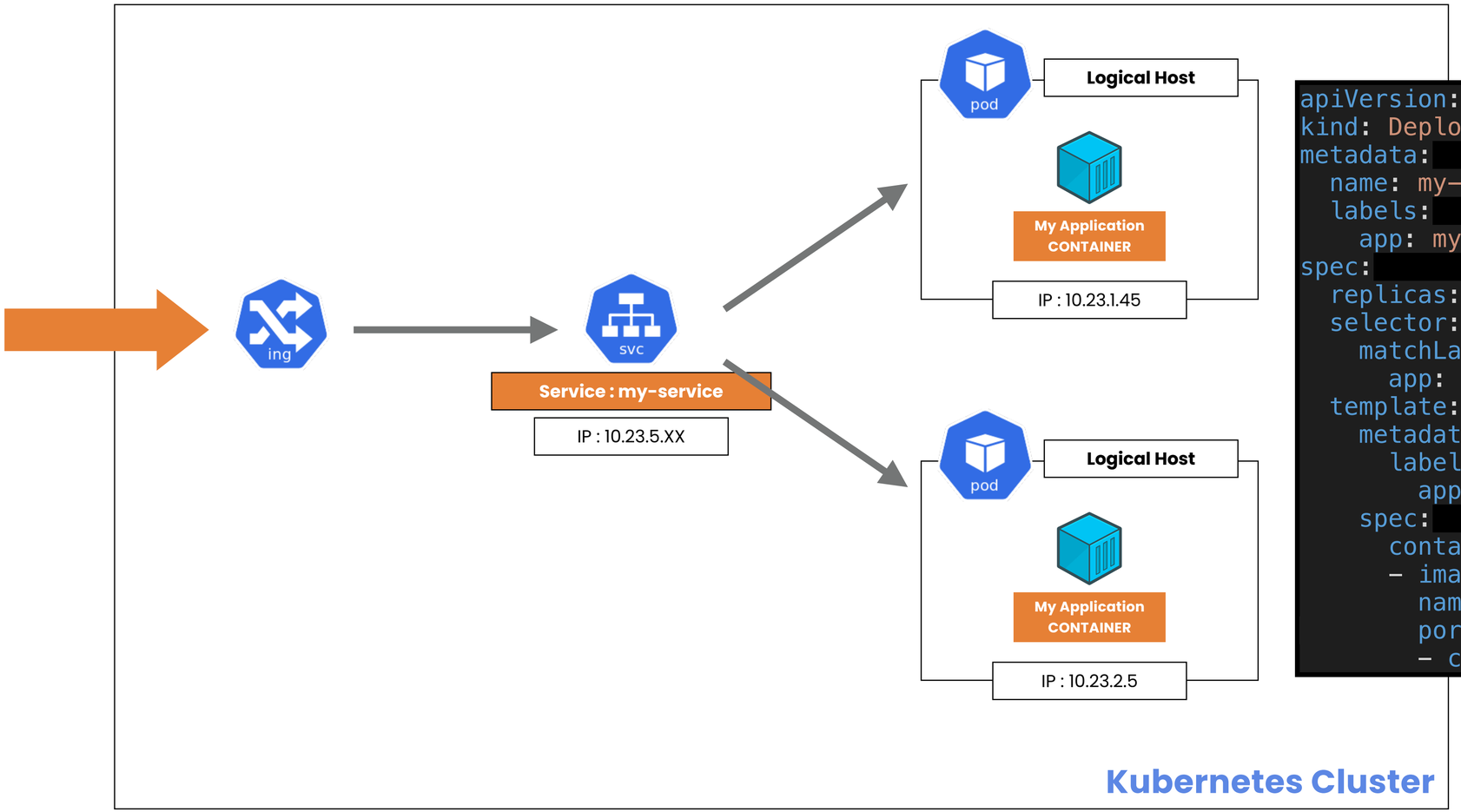
This is the default

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

CLUSTERIP

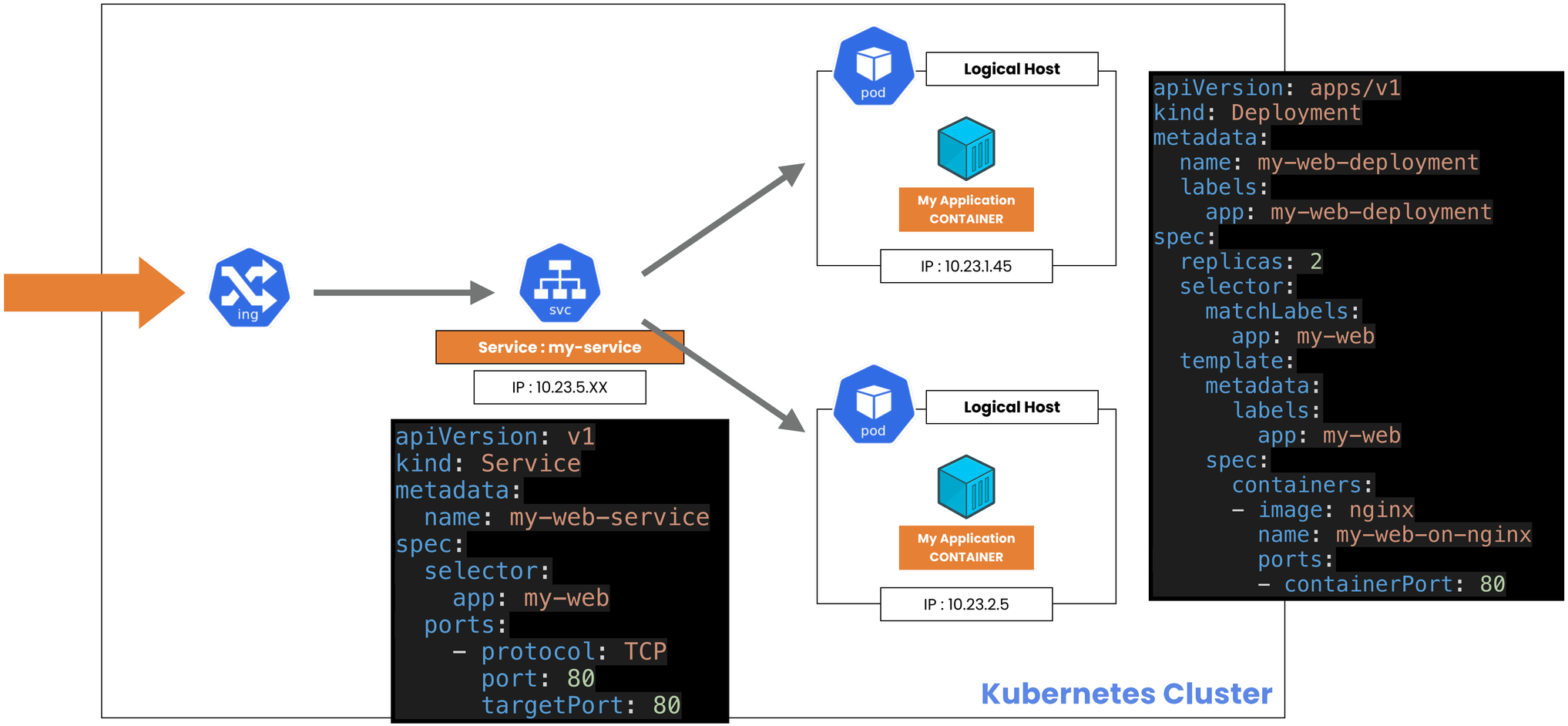

pod
Logical Host
My Application
CONTAINER
IP : 10.23.1.45
ing
svc
Service : my-service
IP : 10.23.5.XX
pod
Logical Host
My Application
CONTAINER
IP : 10.23.2.5
Kubernetes Cluster

CLUSTERIP

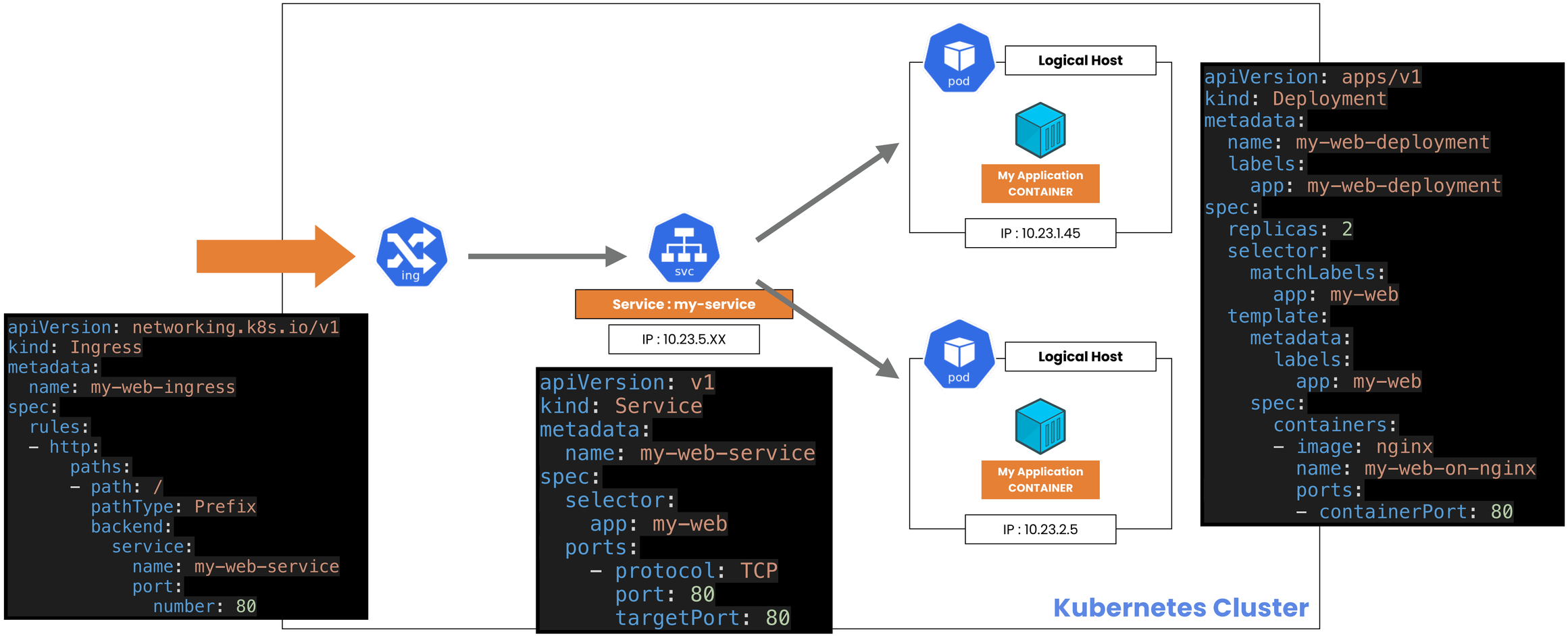



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
        - containerPort: 80
```

CLUSTERIP


ing
svc
Service : my-service
IP : 10.23.5.XX
pod
Logical Host
My Application
CONTAINER
IP : 10.23.1.45
pod
Logical Host
My Application
CONTAINER
IP : 10.23.2.5
Kubernetes Cluster
apiVersion: v1
kind: Service
metadata:
name: my-web-service
spec:
selector:
app: my-web
ports:
- protocol: TCP
port: 80
targetPort: 80
apiVersion: apps/v1
kind: Deployment
metadata:
name: my-web-deployment
labels:
app: my-web-deployment
spec:
replicas: 2
selector:
matchLabels:
app: my-web
template:
metadata:
labels:
app: my-web
spec:
containers:
- image: nginx
name: my-web-on-nginx
ports:
- containerPort: 80

CLUSTERIP

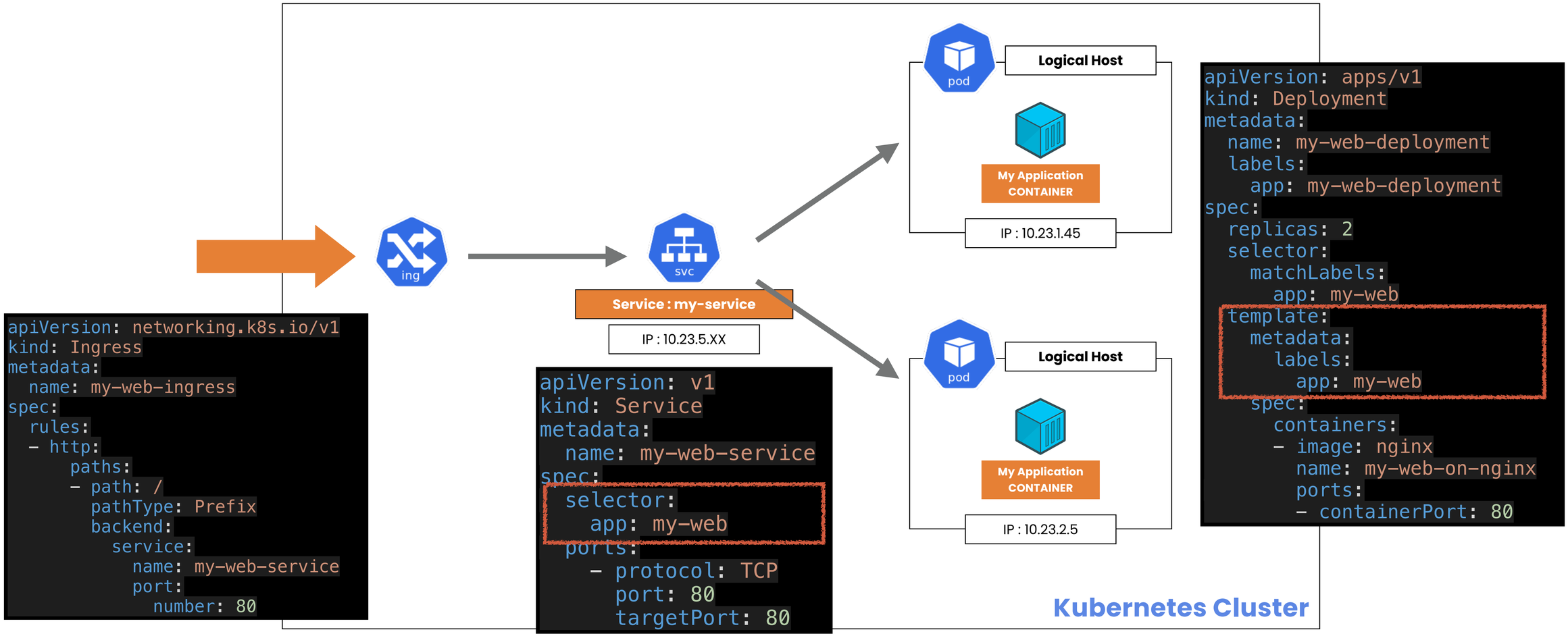



```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-web-ingress
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-web-service
            port:
              number: 80
```

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

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
        - containerPort: 80
```

CLUSTERIP

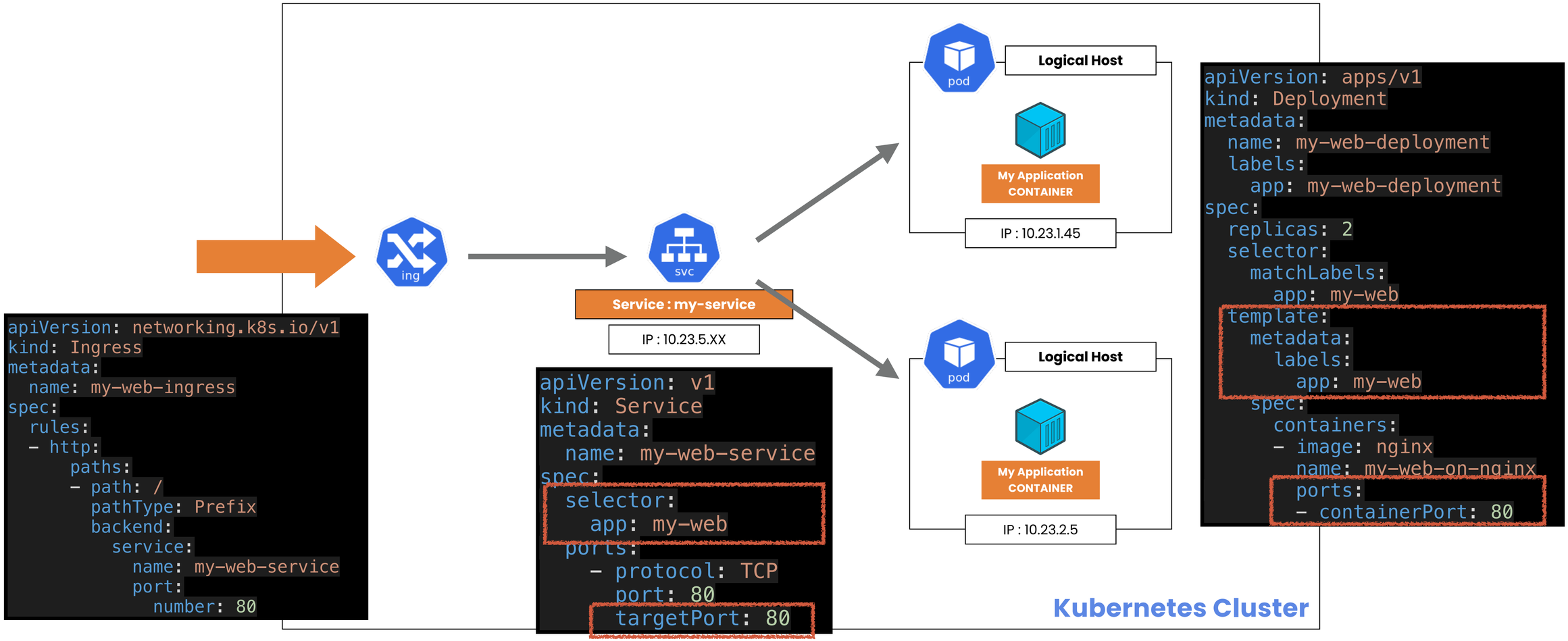



```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-web-ingress
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-web-service
            port:
              number: 80
```

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

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
        - containerPort: 80
```

CLUSTERIP




```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-web-ingress
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-web-service
            port:
              number: 80
```

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

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-web-deployment
  labels:
    app: my-web-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: my-web
  template:
    metadata:
      labels:
        app: my-web
    spec:
      containers:
      - image: nginx
        name: my-web-on-nginx
        ports:
        - containerPort: 80
```

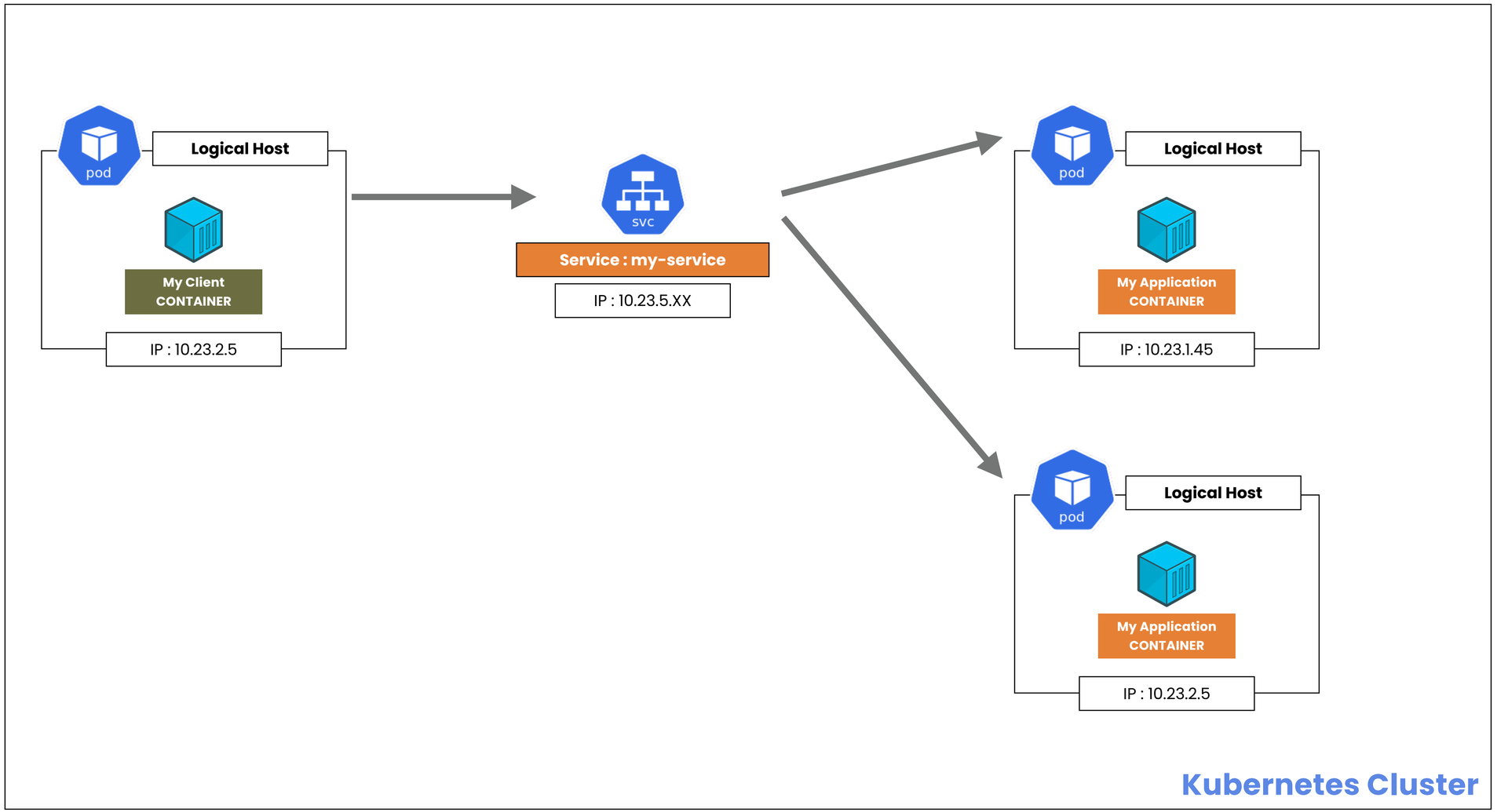
HEADLESS SERVICES

HEADLESS SERVICE

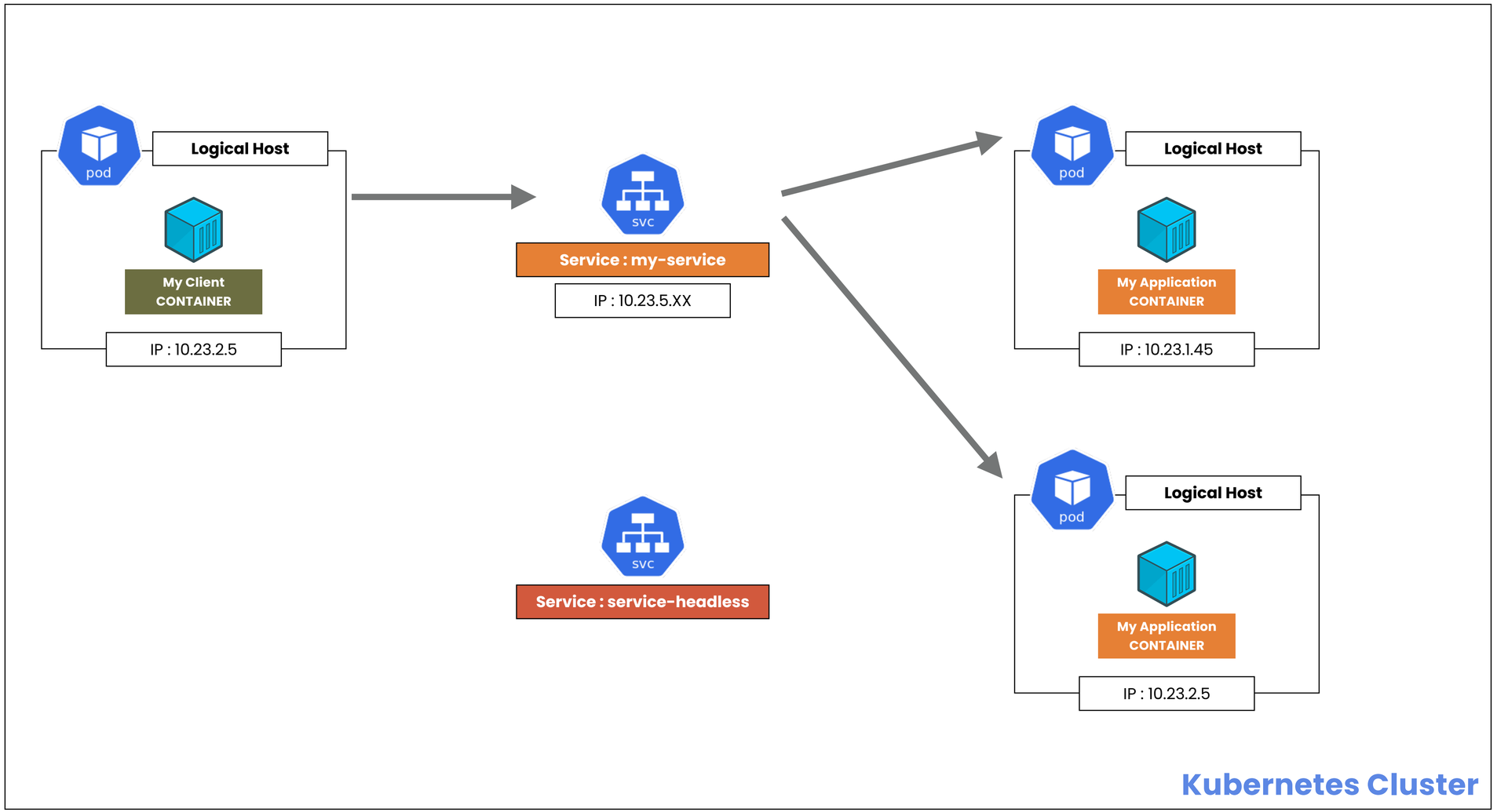
Headless : Exposes the Service that you don't need load-balancing and a single Service IP. In this case, you can create what are termed "headless" Services, by explicitly specifying `"None"` for the cluster IP (`.spec.clusterIP`).

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-web-service-headless
spec:
  clusterIP: None
  ports:
  - port: 80
    name: web
  selector:
    app: nginx
```

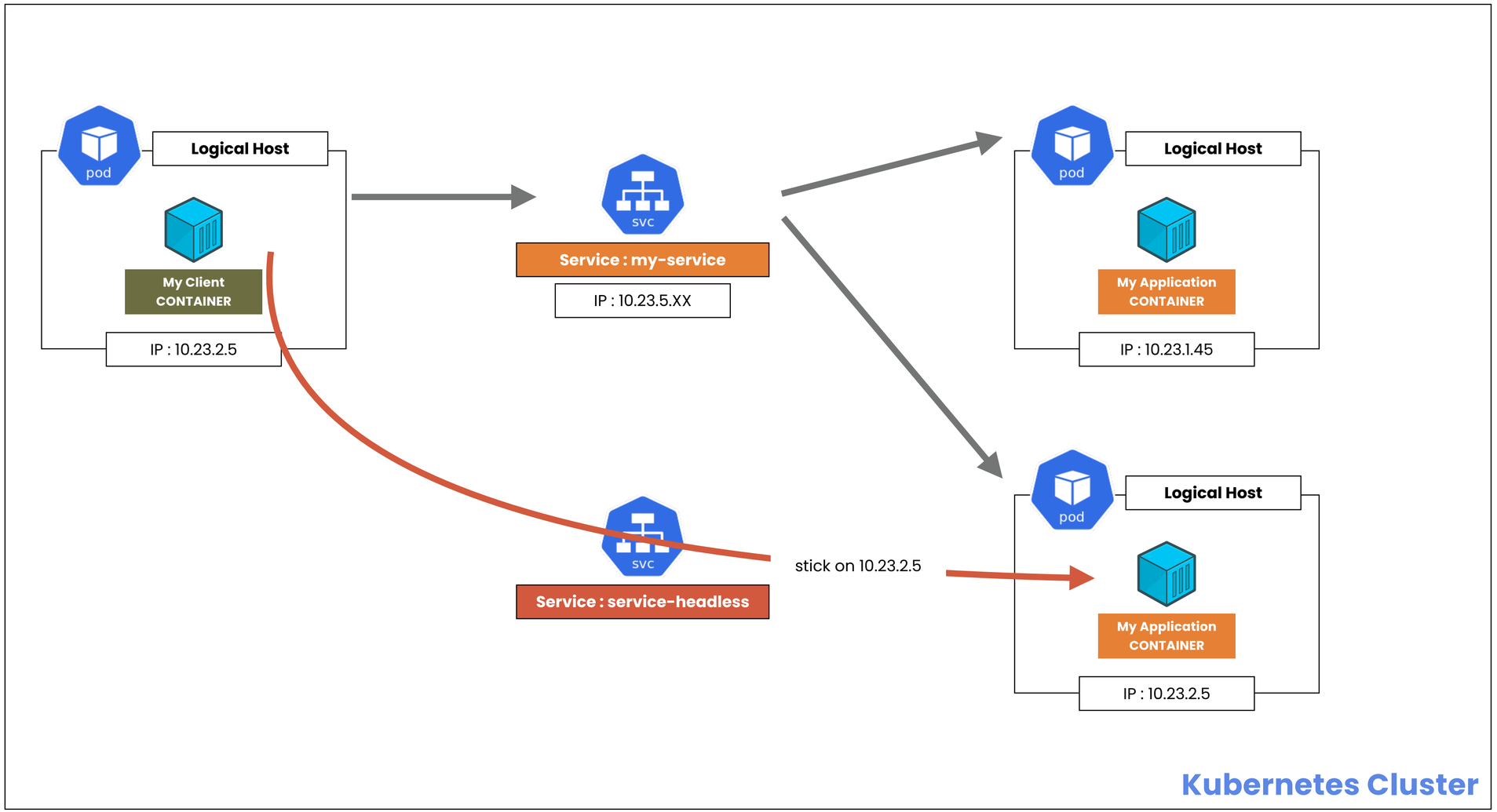
HEADLESS


pod
Logical Host
My Client
CONTAINER
IP : 10.23.2.5
svc
Service : my-service
IP : 10.23.5.XX
pod
Logical Host
My Application
CONTAINER
IP : 10.23.1.45
pod
Logical Host
My Application
CONTAINER
IP : 10.23.2.5
Kubernetes Cluster

HEADLESS


pod
Logical Host
My Client
CONTAINER
IP : 10.23.2.5
SVC
Service : my-service
IP : 10.23.5.XX
SVC
Service : service-headless
pod
Logical Host
My Application
CONTAINER
IP : 10.23.1.45
pod
Logical Host
My Application
CONTAINER
IP : 10.23.2.5
Kubernetes Cluster

HEADLESS


pod
Logical Host
My Client
CONTAINER
IP : 10.23.2.5
SVC
Service : my-service
IP : 10.23.5.XX
pod
Logical Host
My Application
CONTAINER
IP : 10.23.1.45
SVC
Service : service-headless
stick on 10.23.2.5
pod
Logical Host
My Application
CONTAINER
IP : 10.23.2.5
Kubernetes Cluster

LET TRY !

THANK YOU