

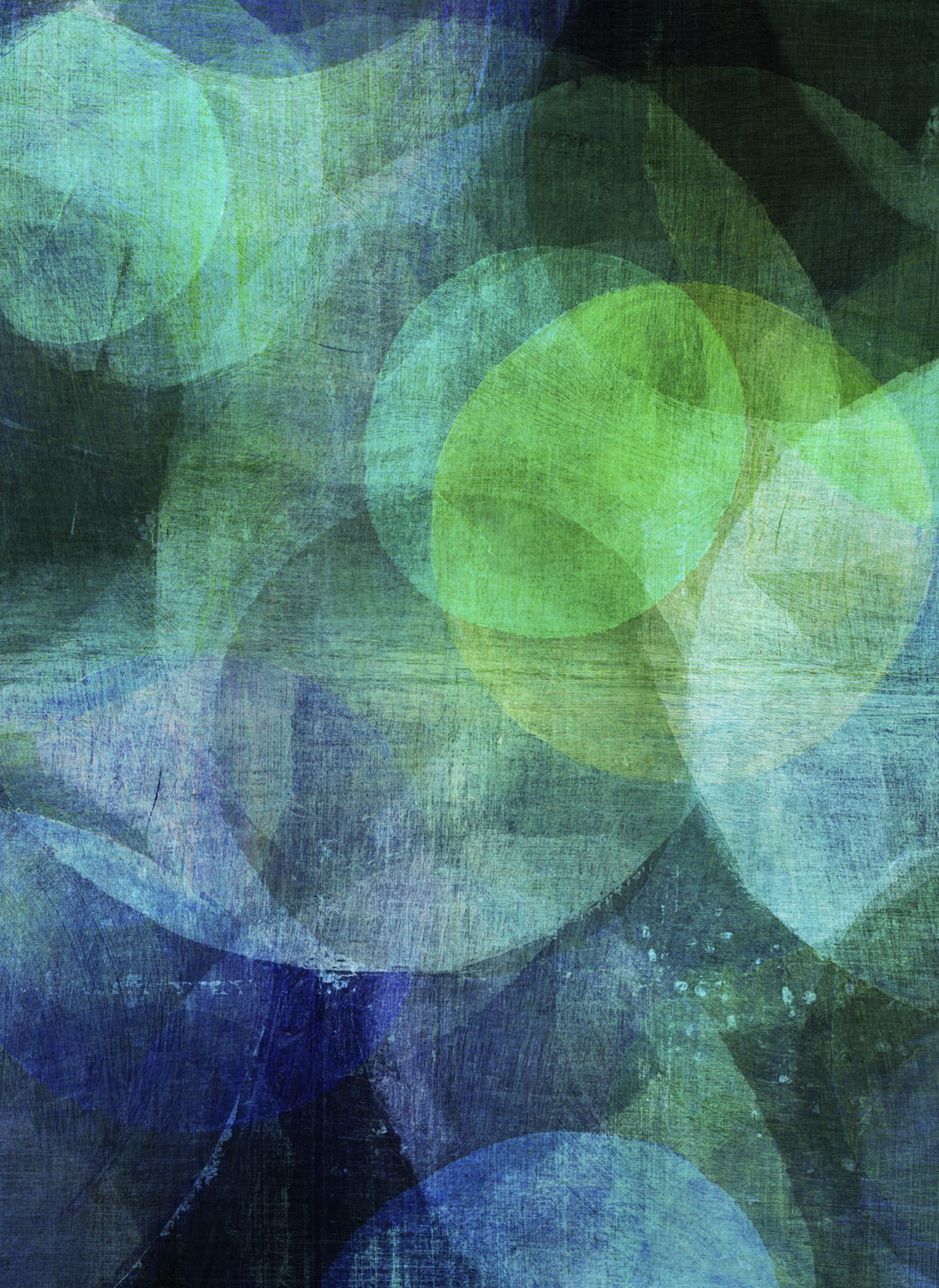
INTRODUCTION TO KUBERNETES



INTRODUCTION TO KUBERNETES

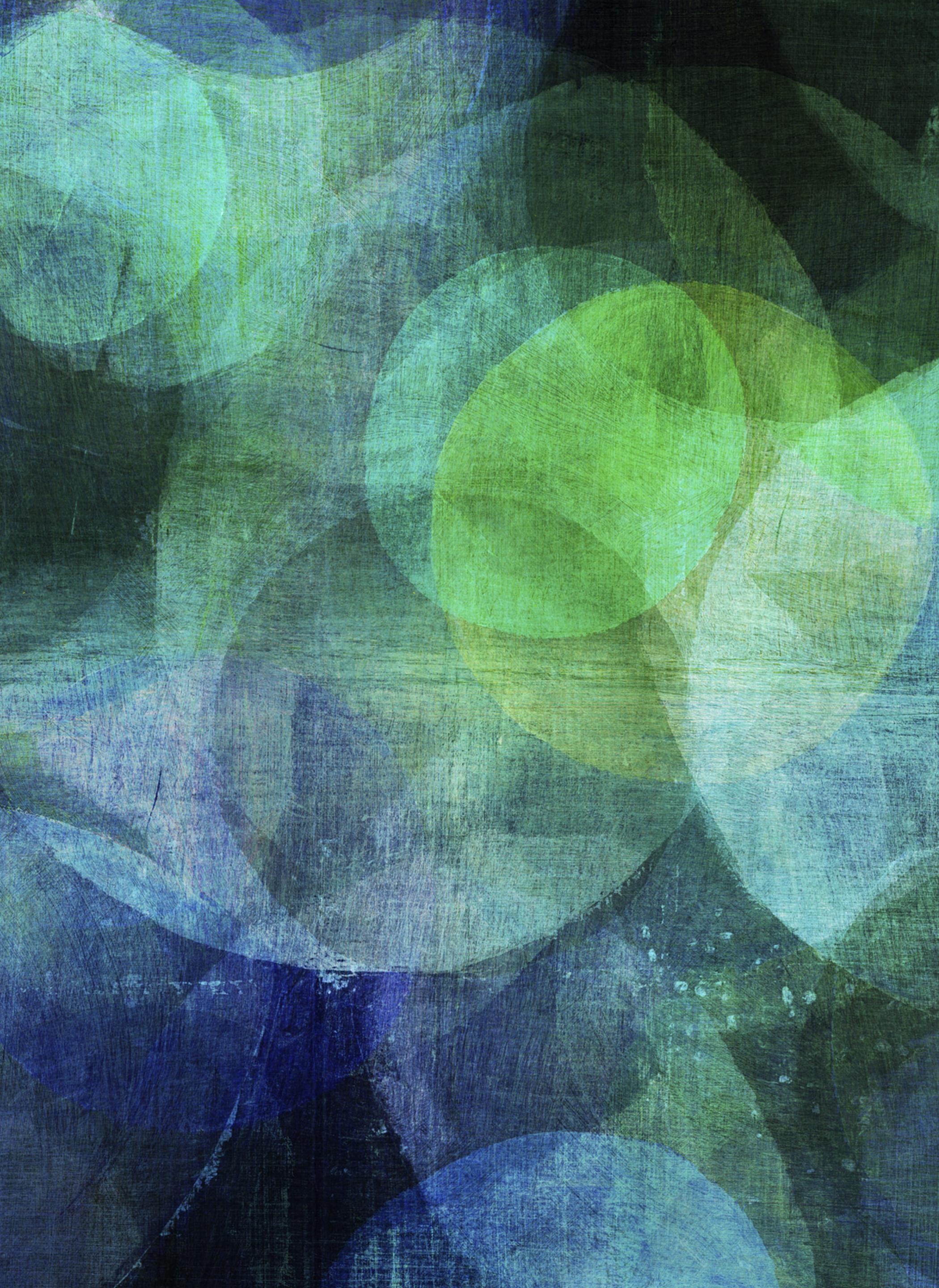
Kubernetes Objects and Components Part 2

AGENDA



AGENDA

- ConfigMap & Secret



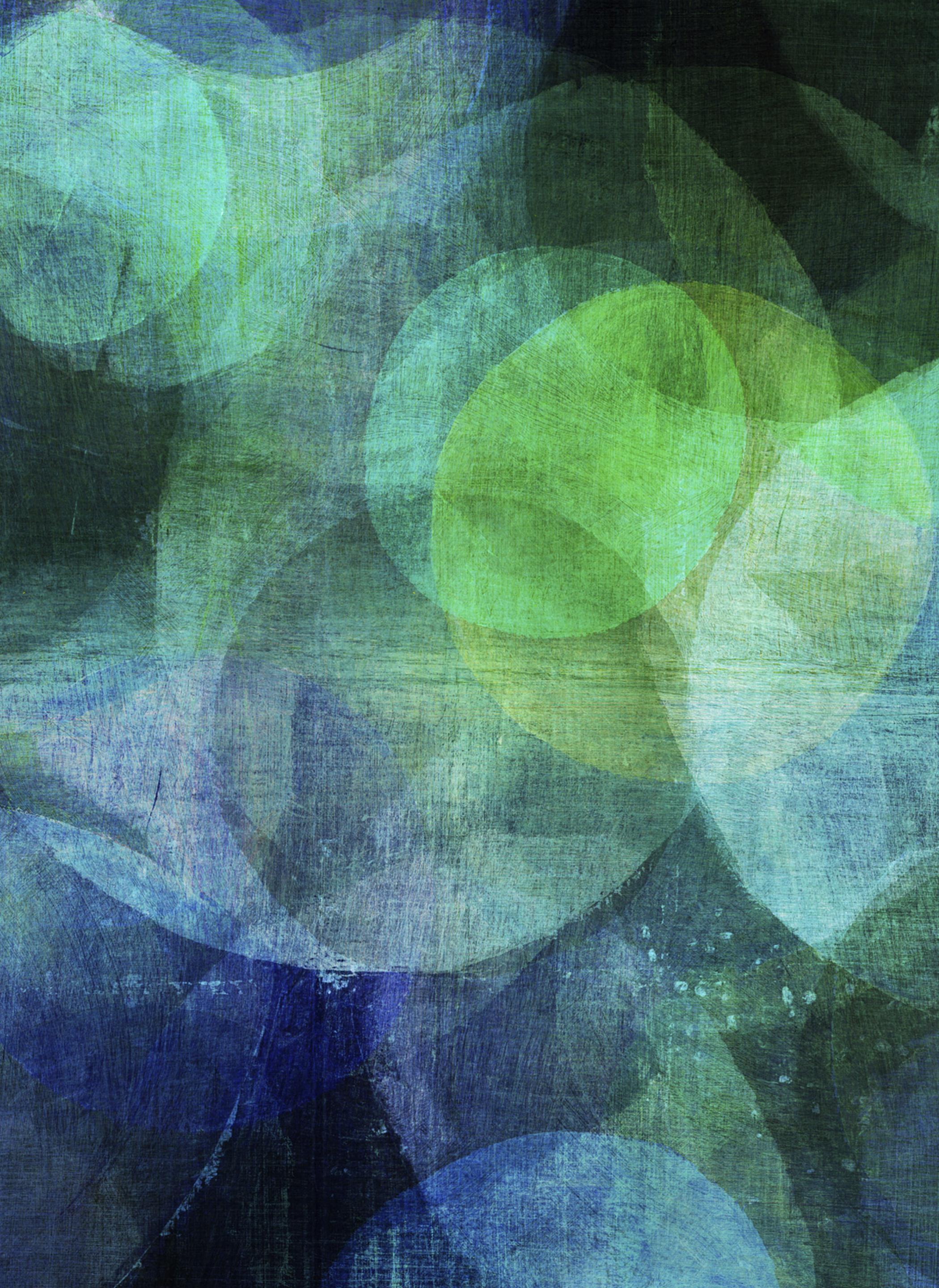
AGENDA

- ConfigMap & Secret
- Volume



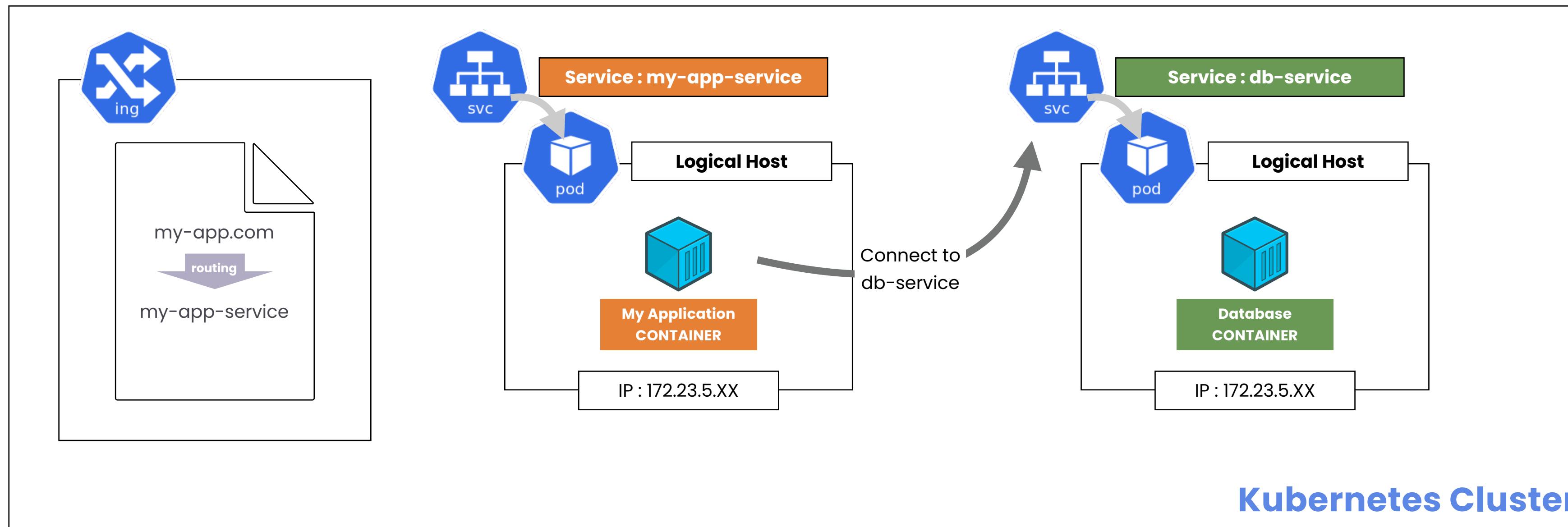
AGENDA

- ConfigMap & Secret
- Volume
- Deployment & StatefulSet

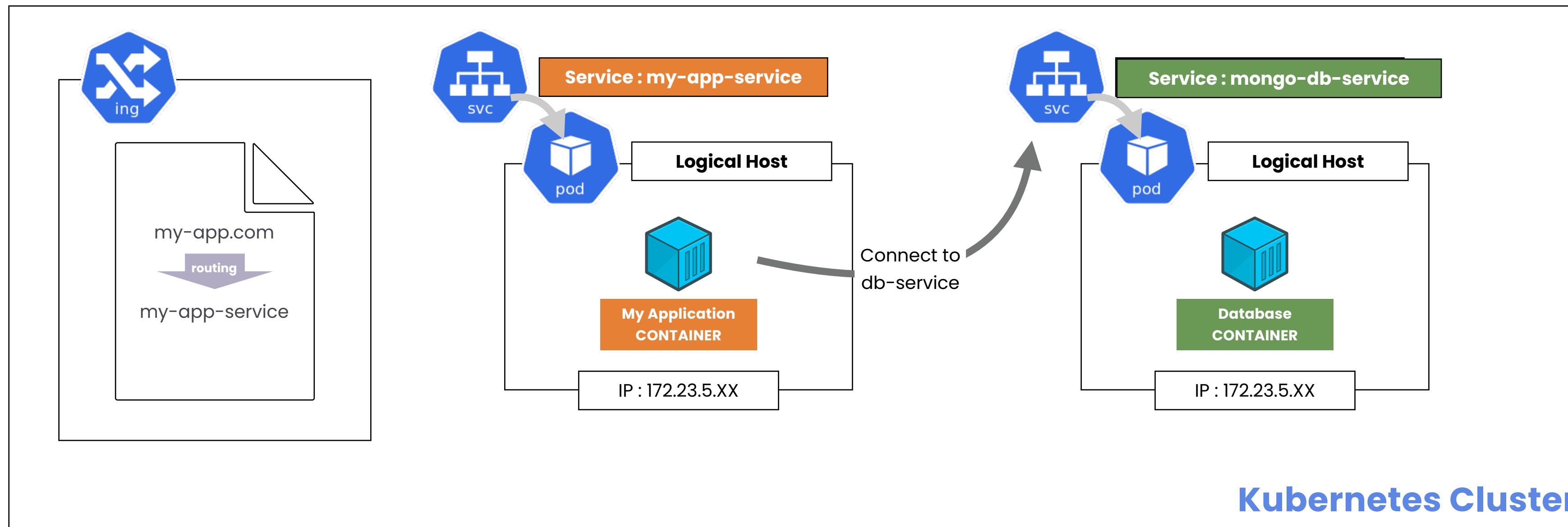


CONFIGMAP & SECRET

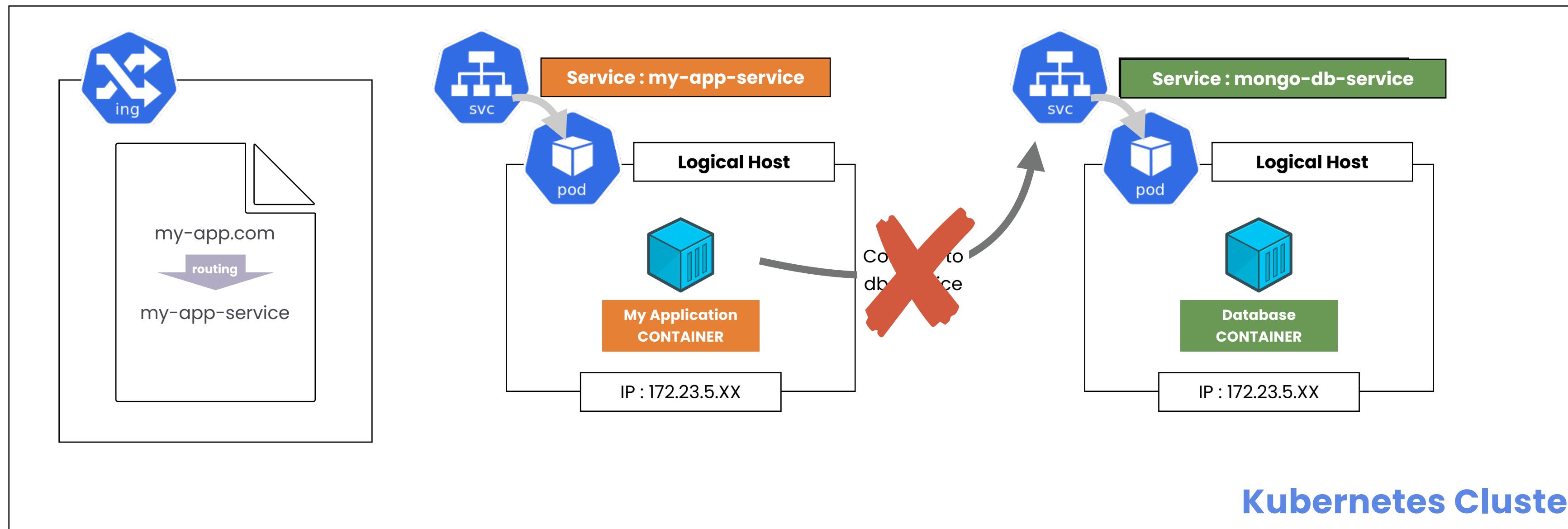
SELF-CONTAINMENT PRINCIPLE



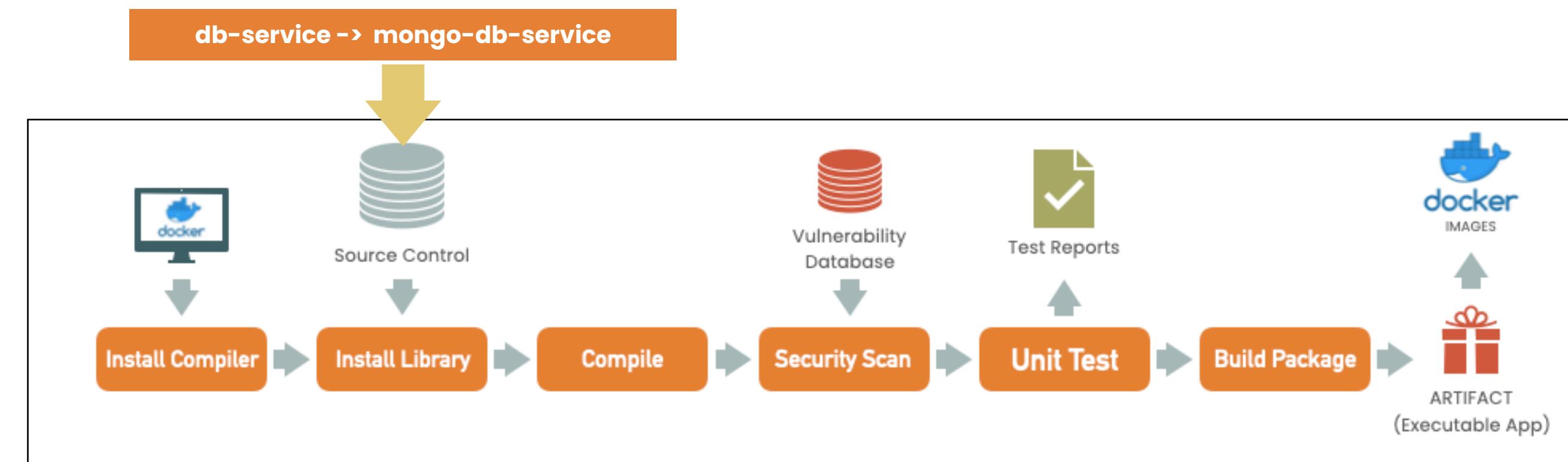
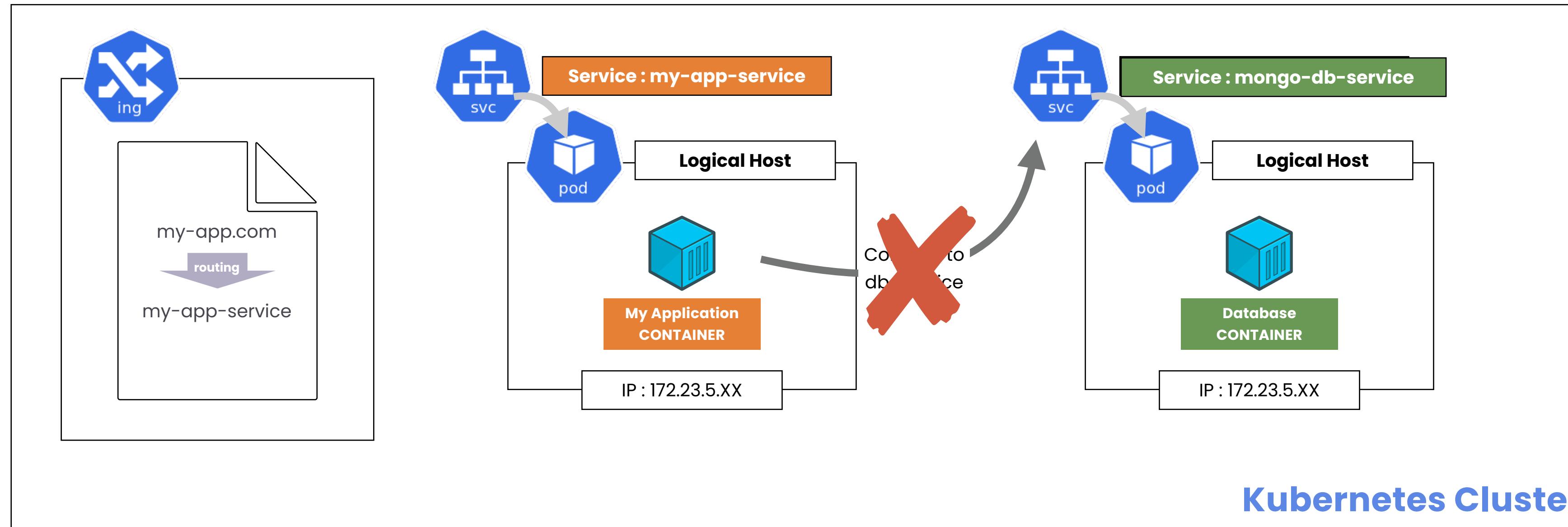
SELF-CONTAINMENT PRINCIPLE



SELF-CONTAINMENT PRINCIPLE



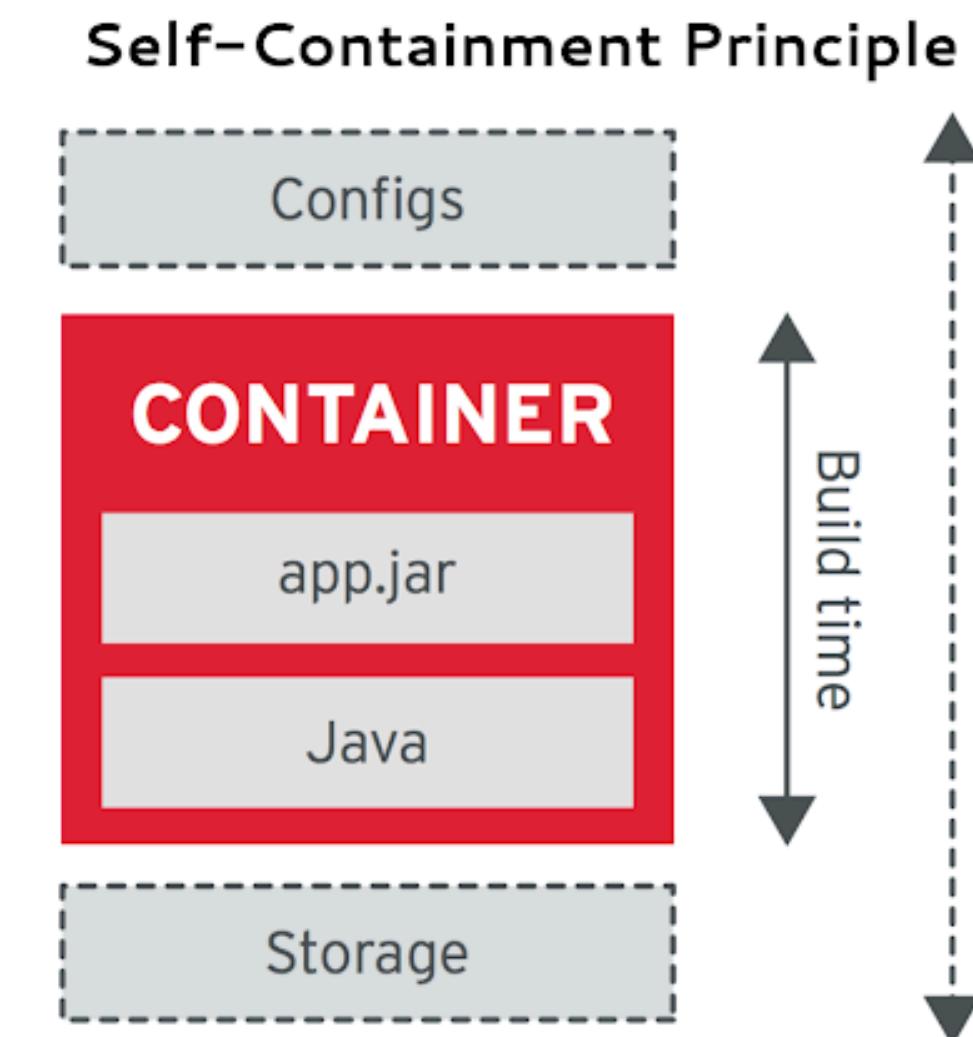
SELF-CONTAINMENT PRINCIPLE



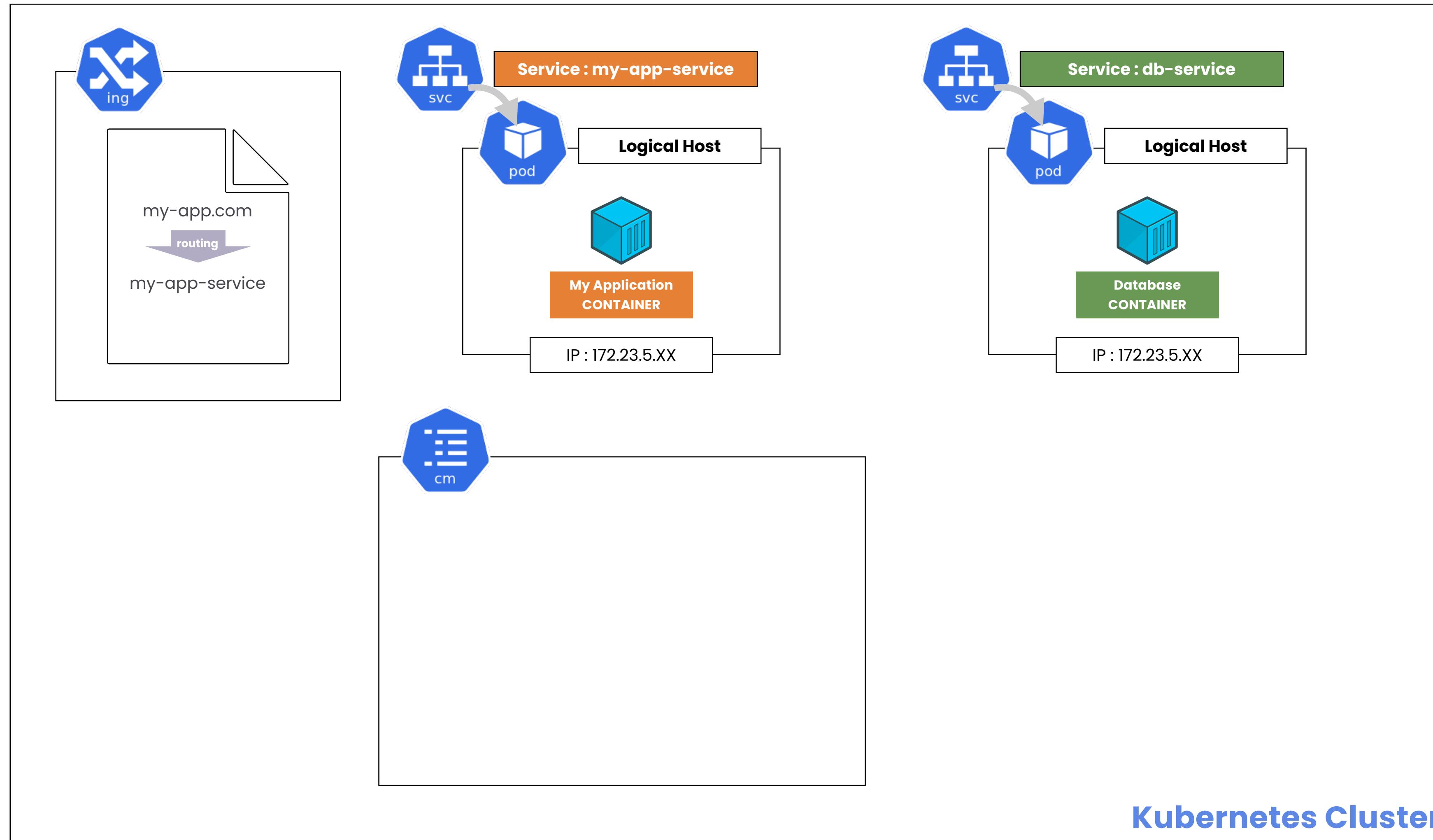
CONFIGMAP

A ConfigMap is an API object used to **store non-confidential data** in key-value pairs. Pods can consume ConfigMaps as **environment variables**, **command-line arguments**, or as **configuration files** in a volume.

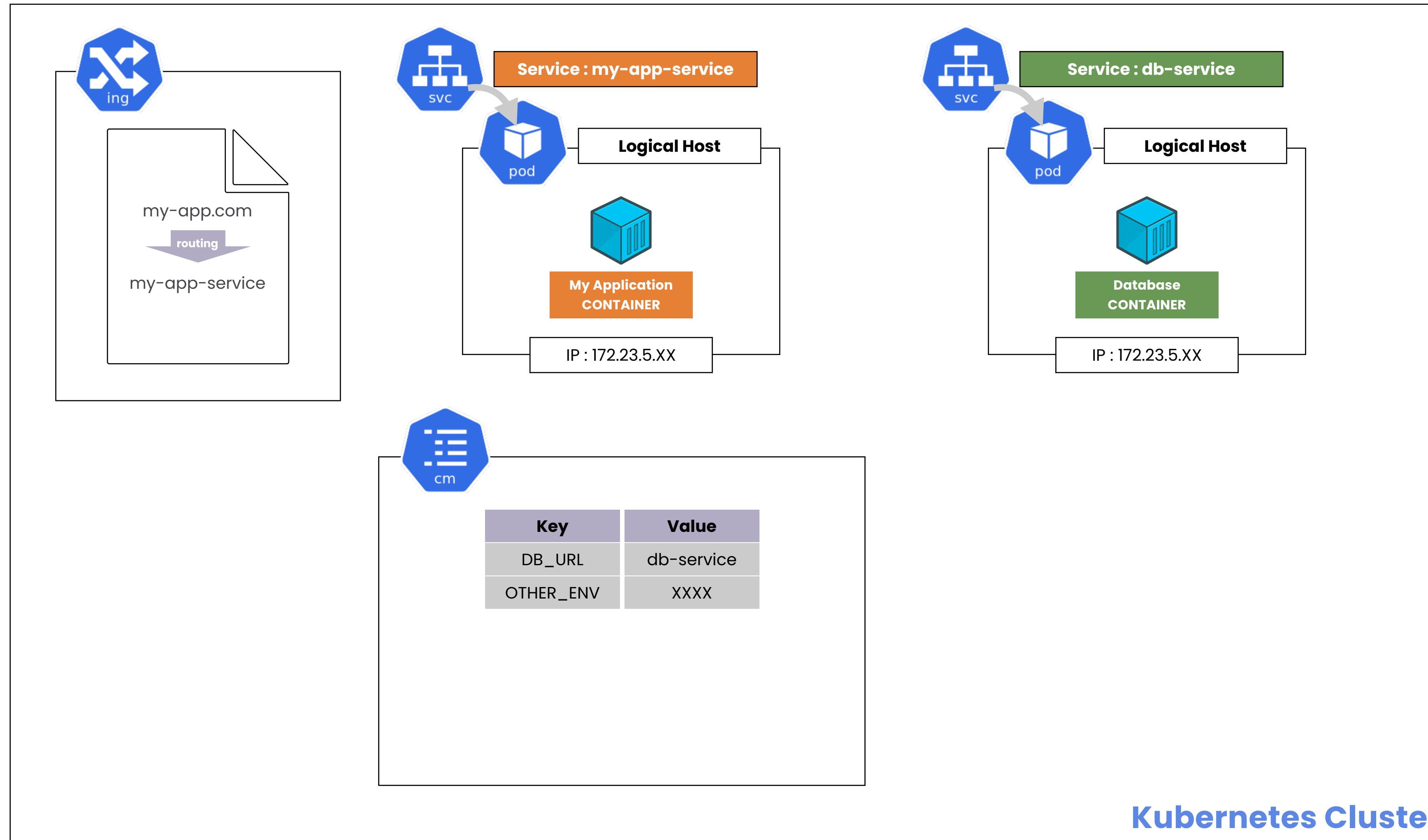
A ConfigMap allows you to decouple environment-specific configuration from your container images, so that your applications are easily portable.



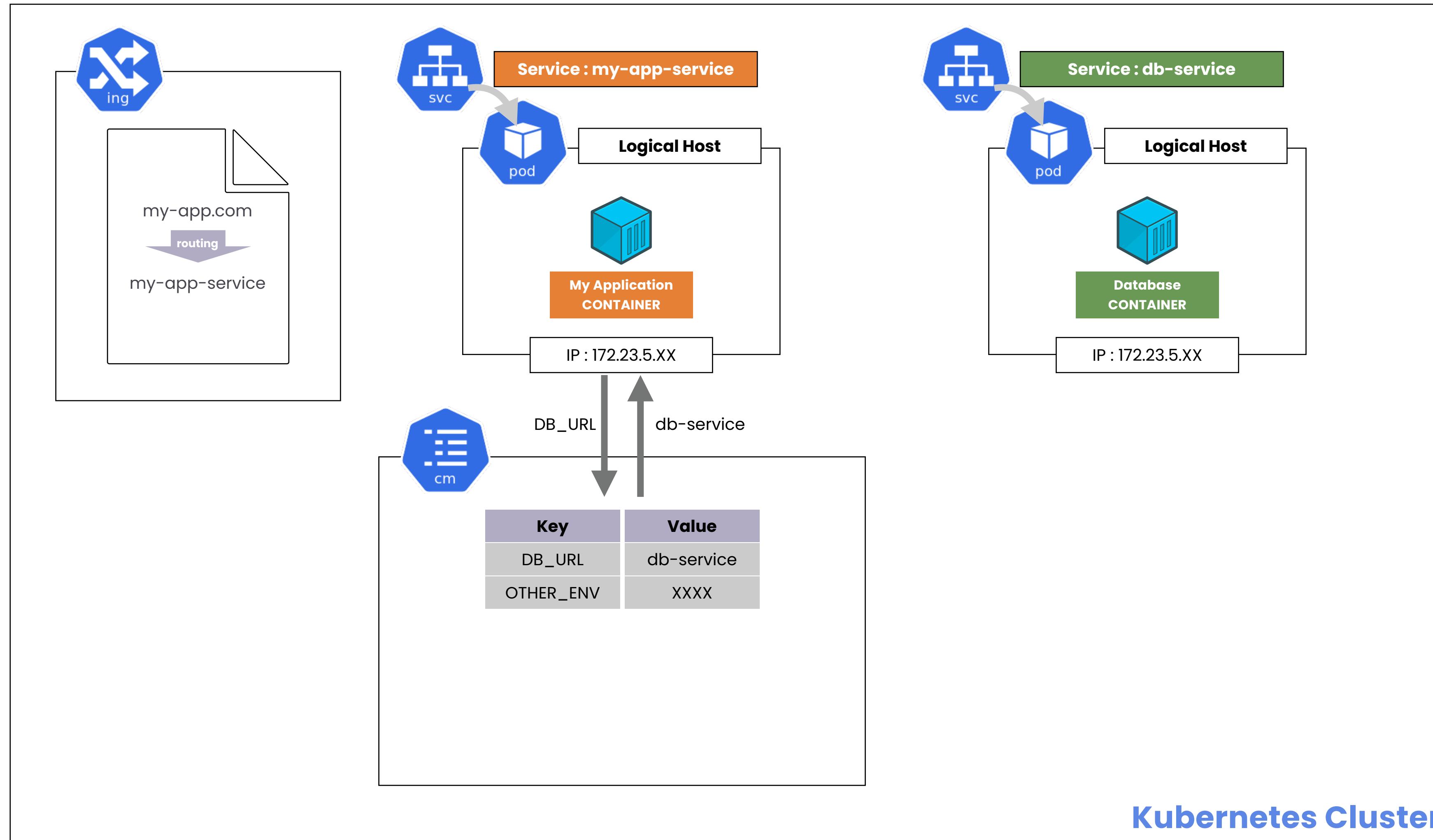
WORKING WITH CONFIGMAP



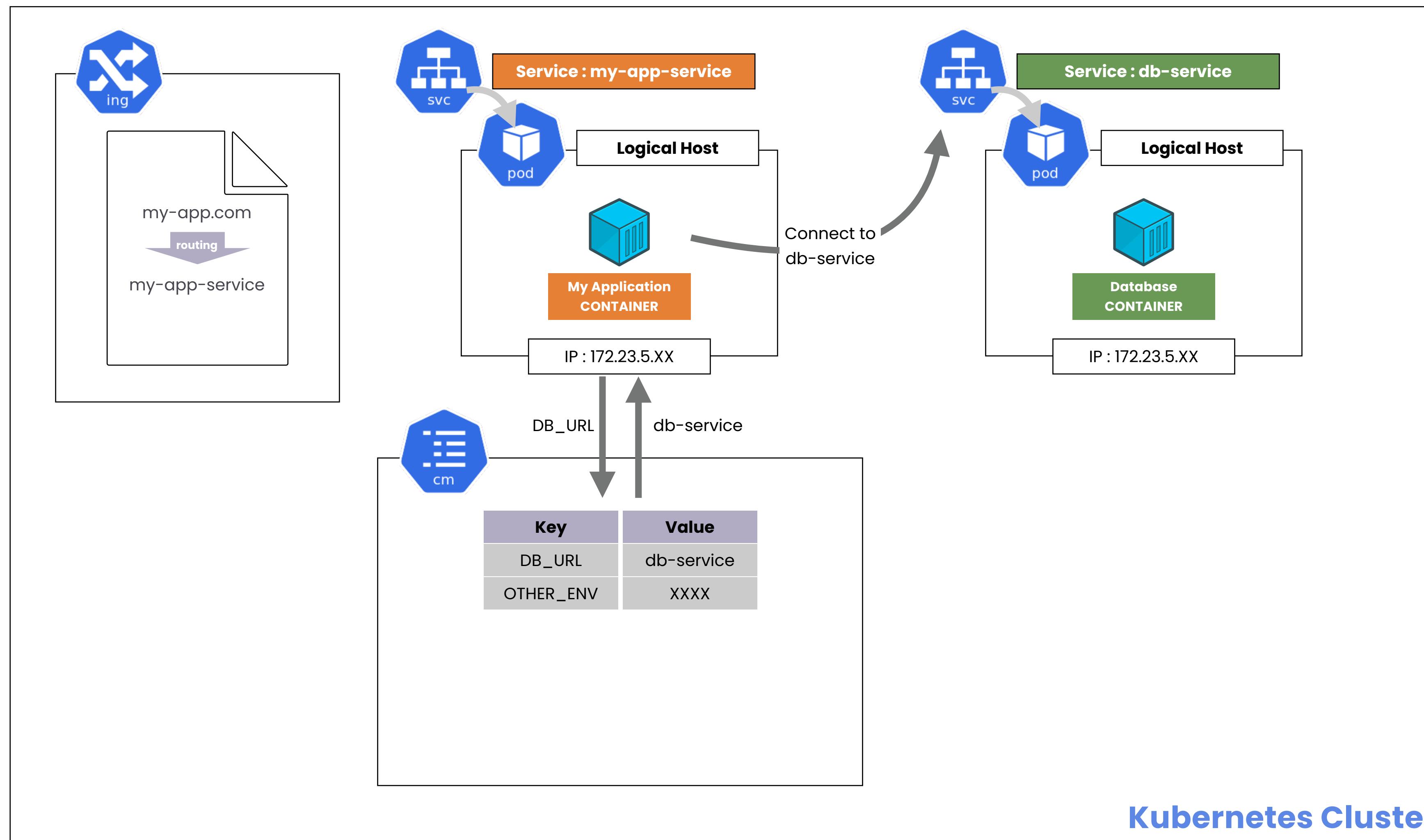
WORKING WITH CONFIGMAP



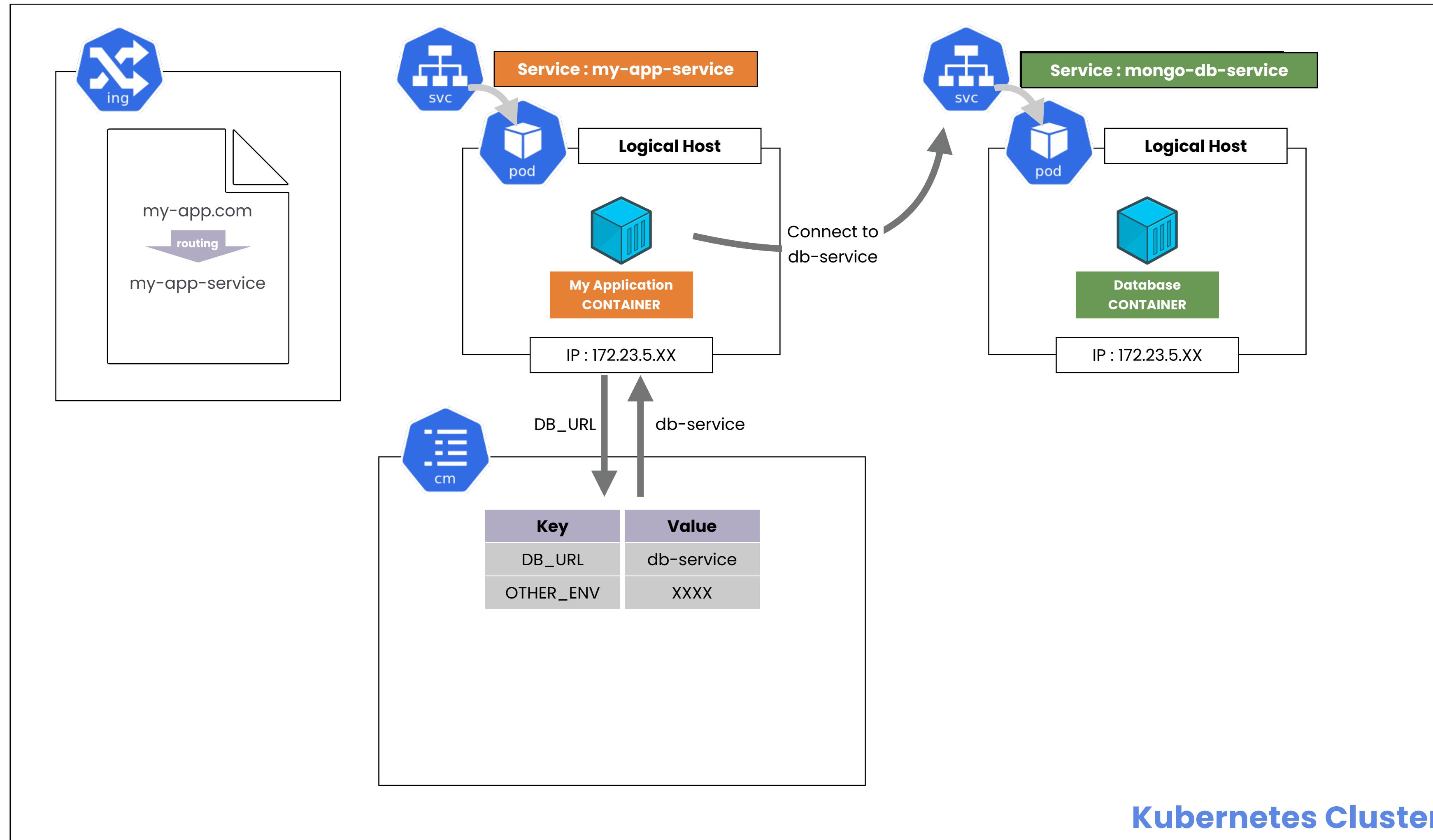
WORKING WITH CONFIGMAP



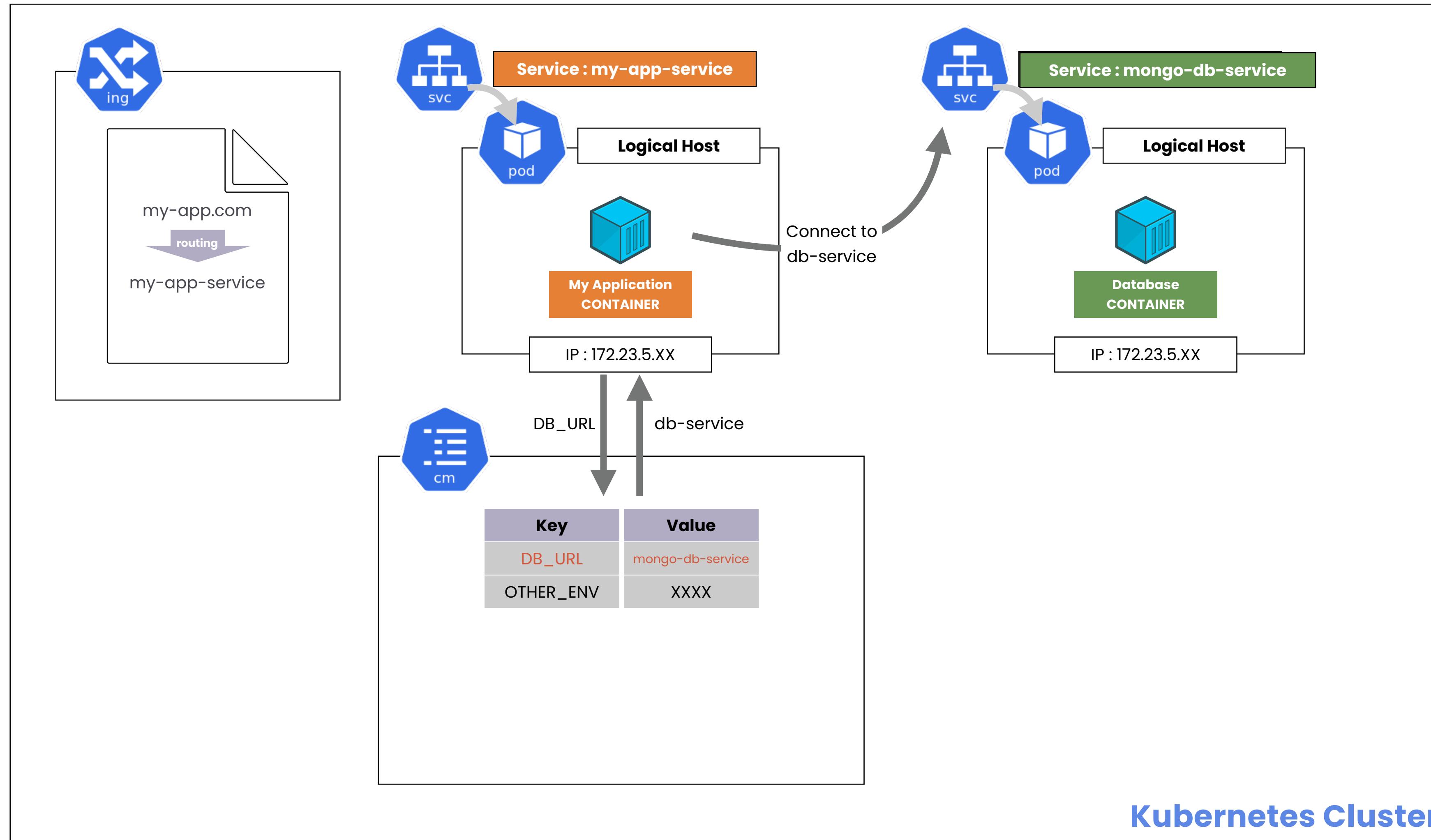
WORKING WITH CONFIGMAP



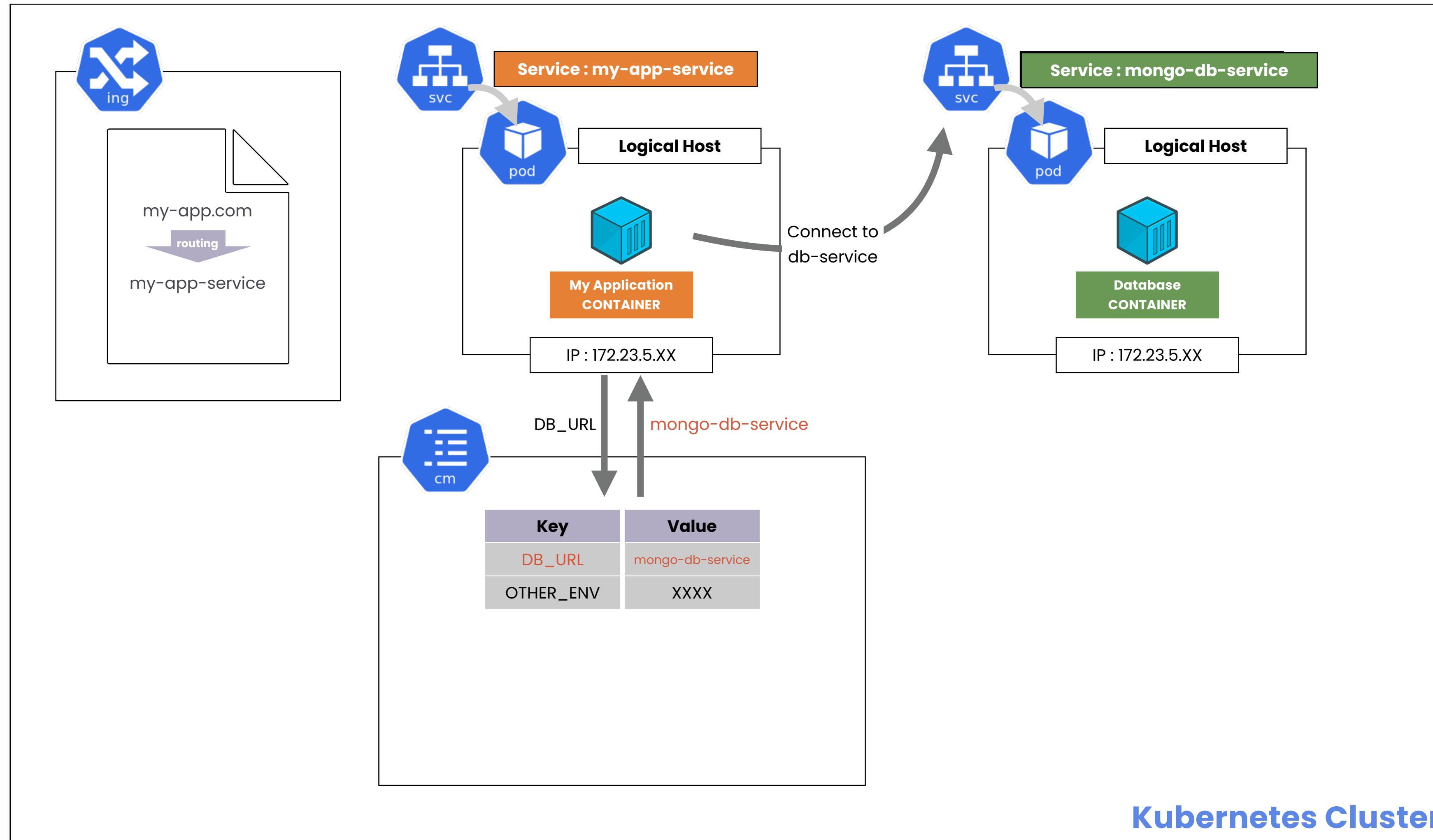
WORKING WITH CONFIGMAP



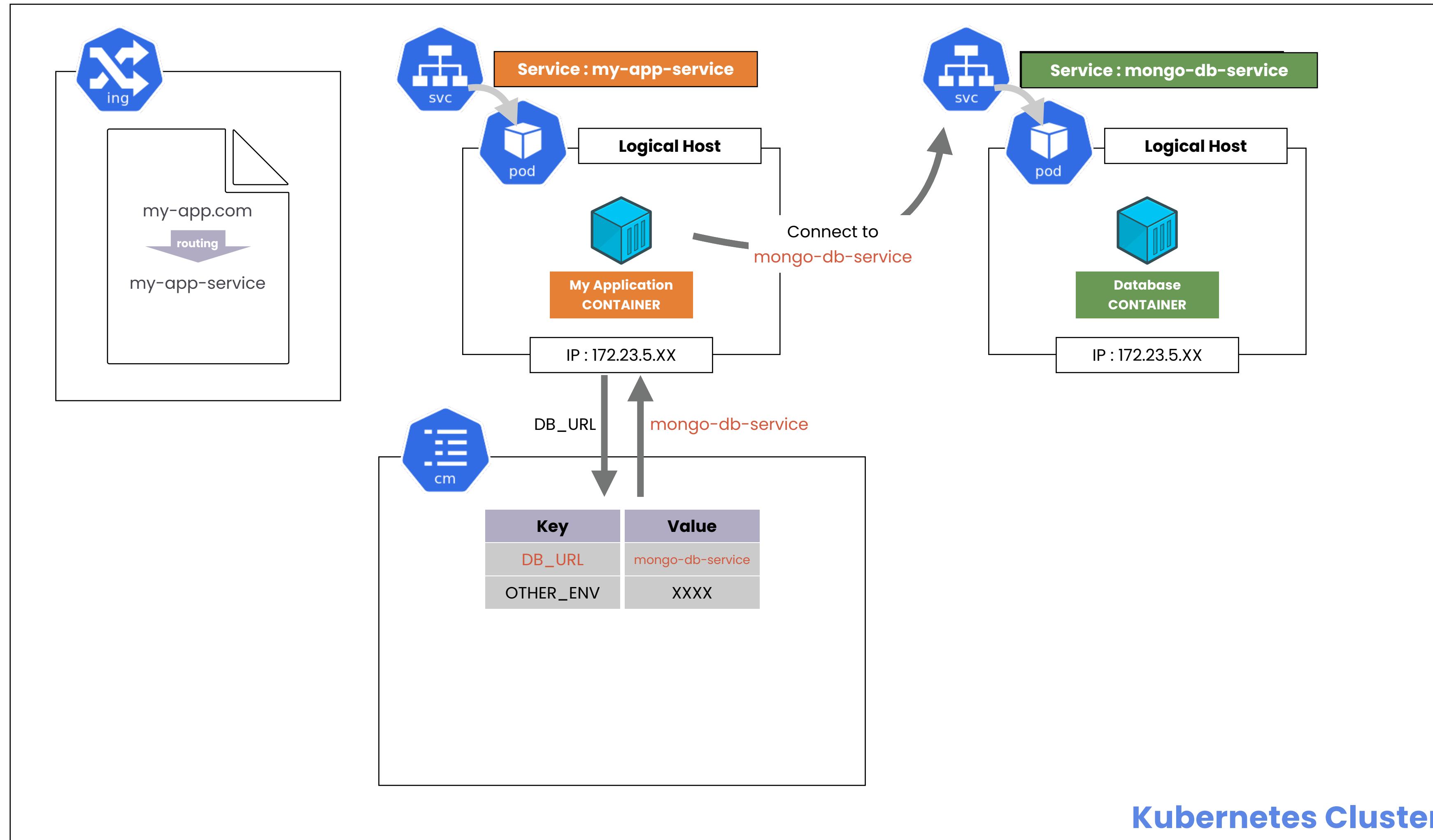
WORKING WITH CONFIGMAP



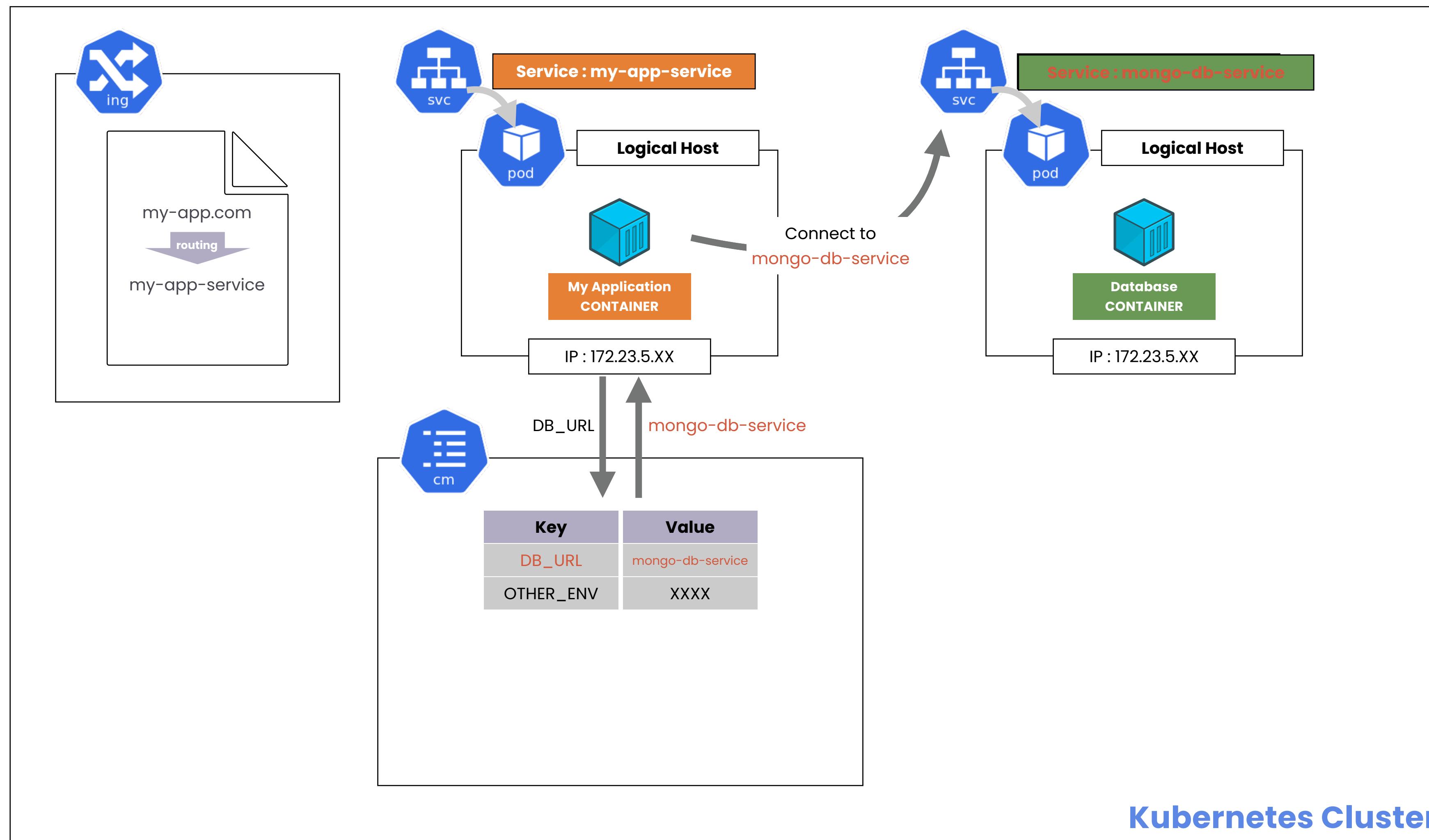
WORKING WITH CONFIGMAP



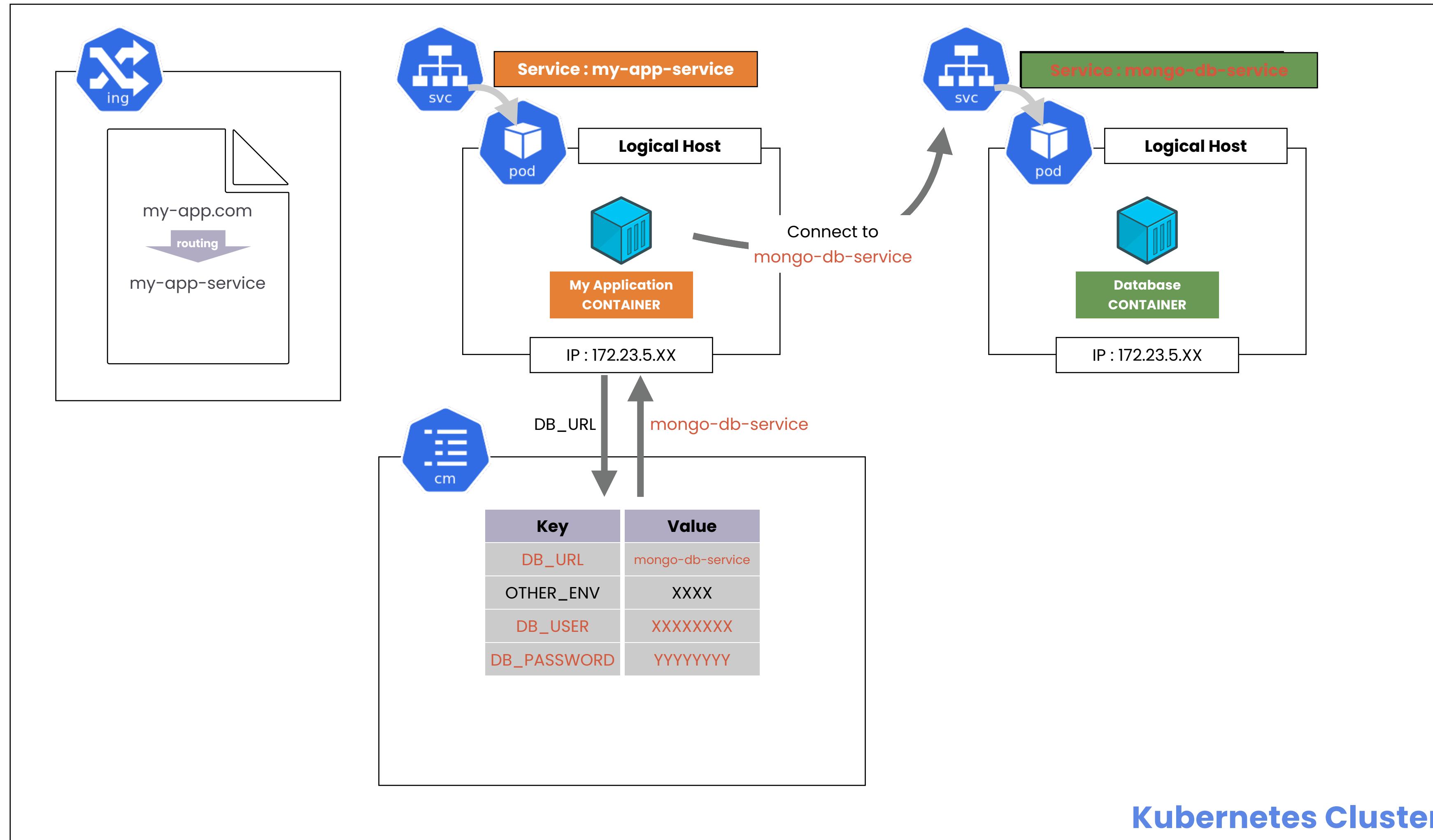
WORKING WITH CONFIGMAP



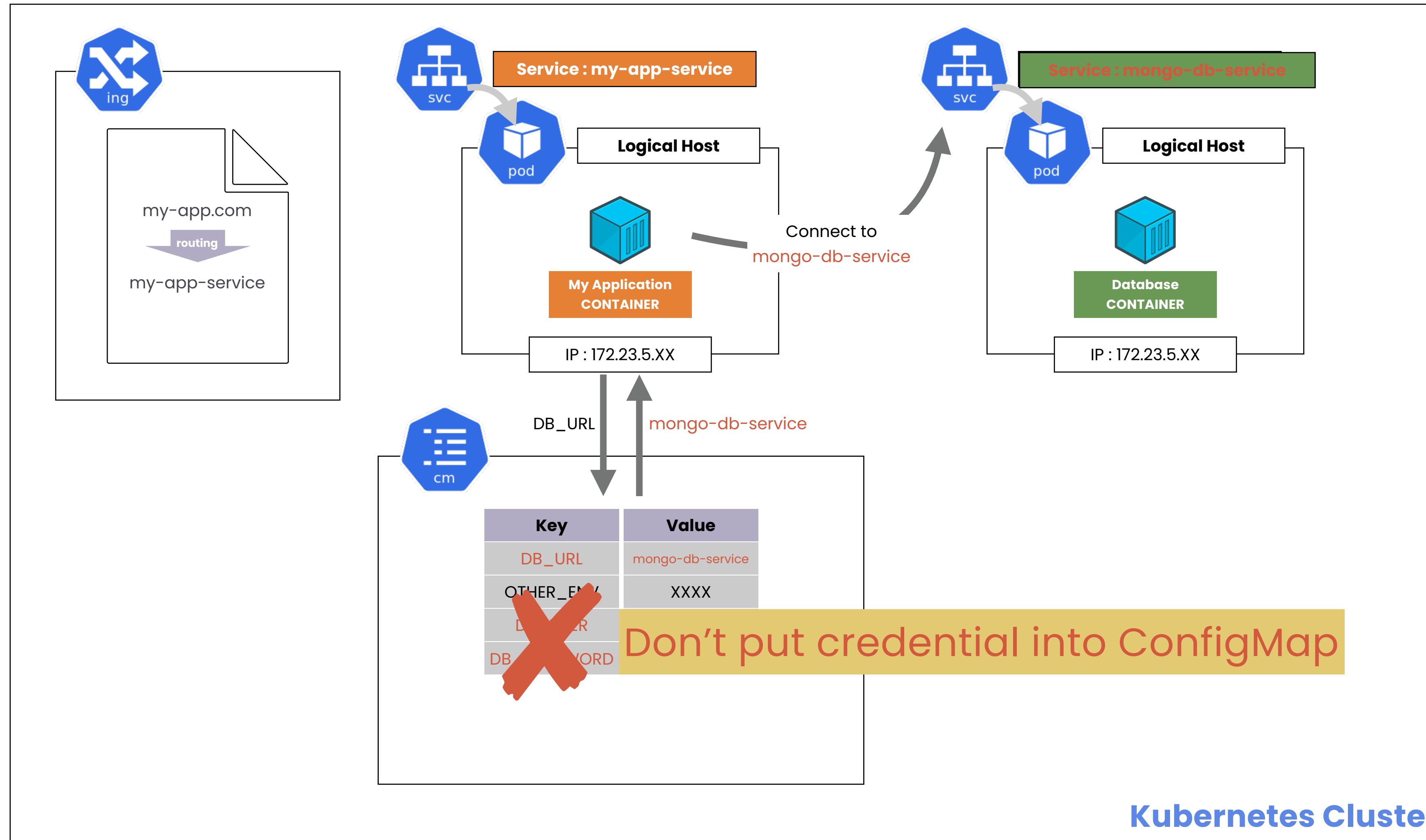
WORKING WITH CONFIGMAP



WORKING WITH CONFIGMAP

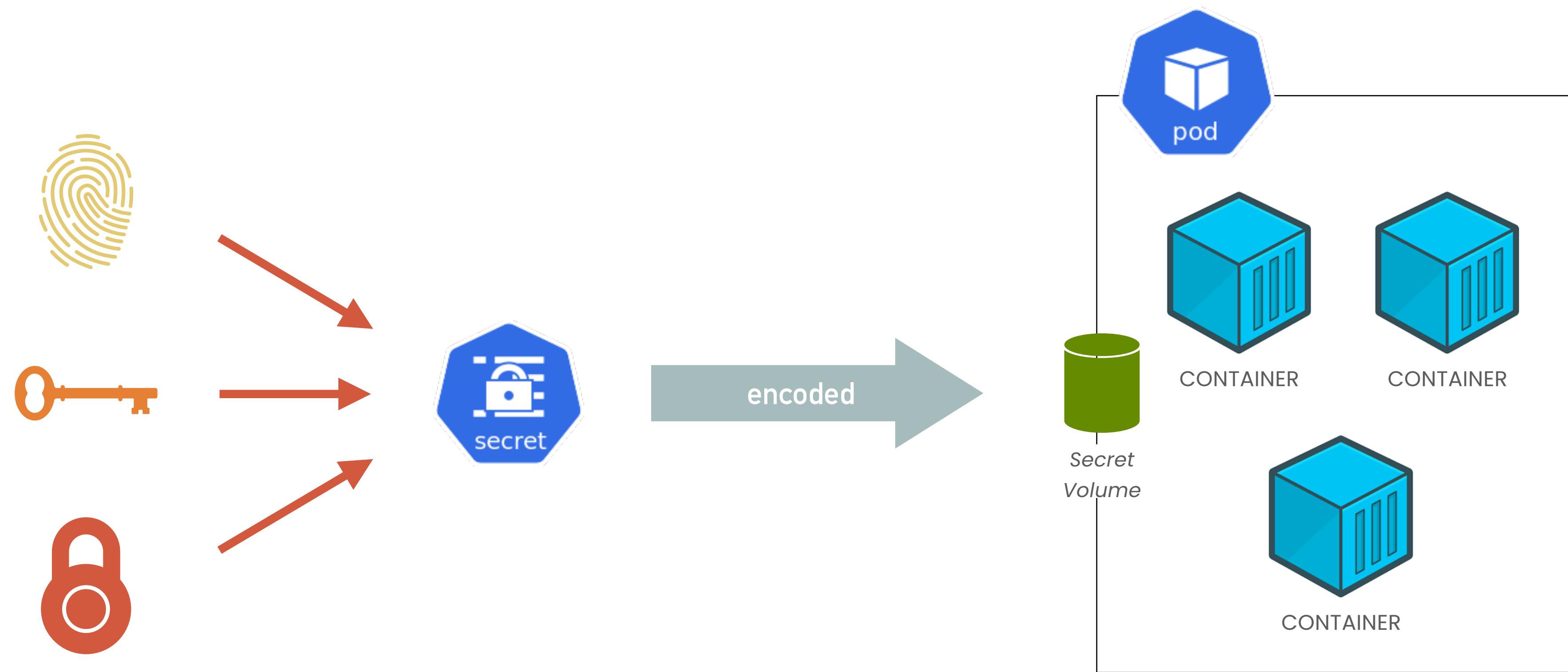


WORKING WITH CONFIGMAP

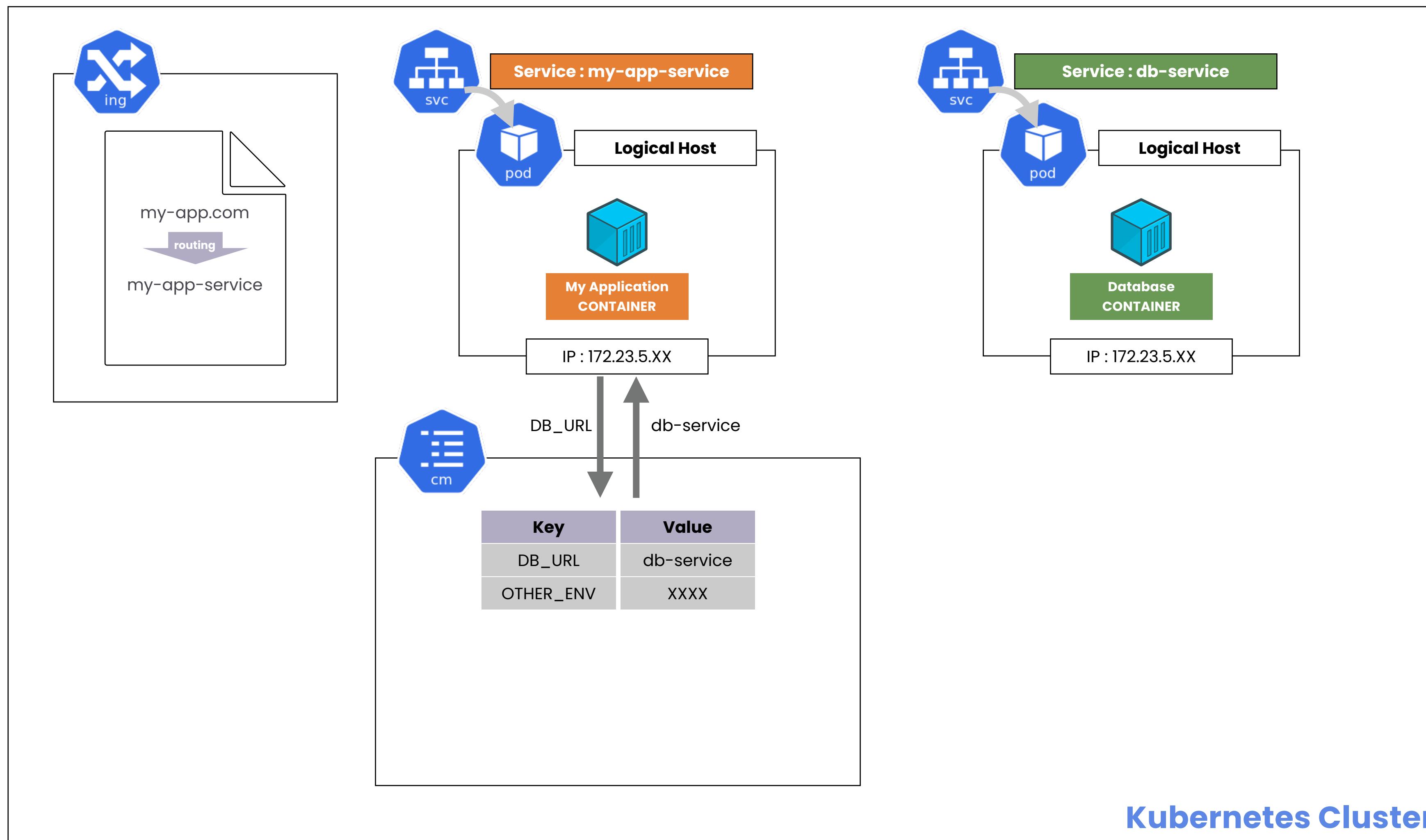


SECRET

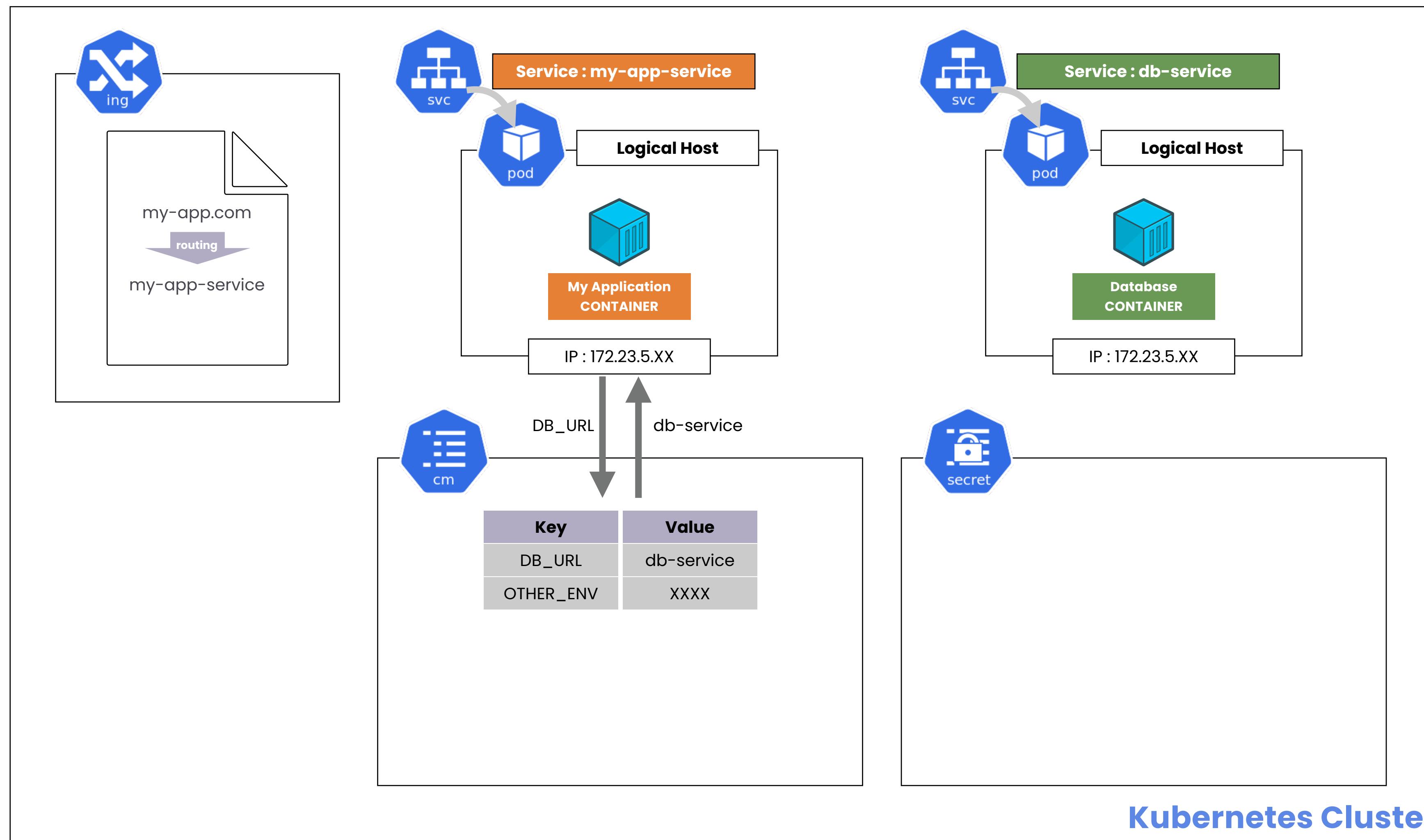
A Secret is an object that contains **a small amount of sensitive data** such as a password, a token, or a key. Such information might otherwise be put in a Pod specification or in a container image. Using a Secret means that you don't need to include confidential data in your application code.



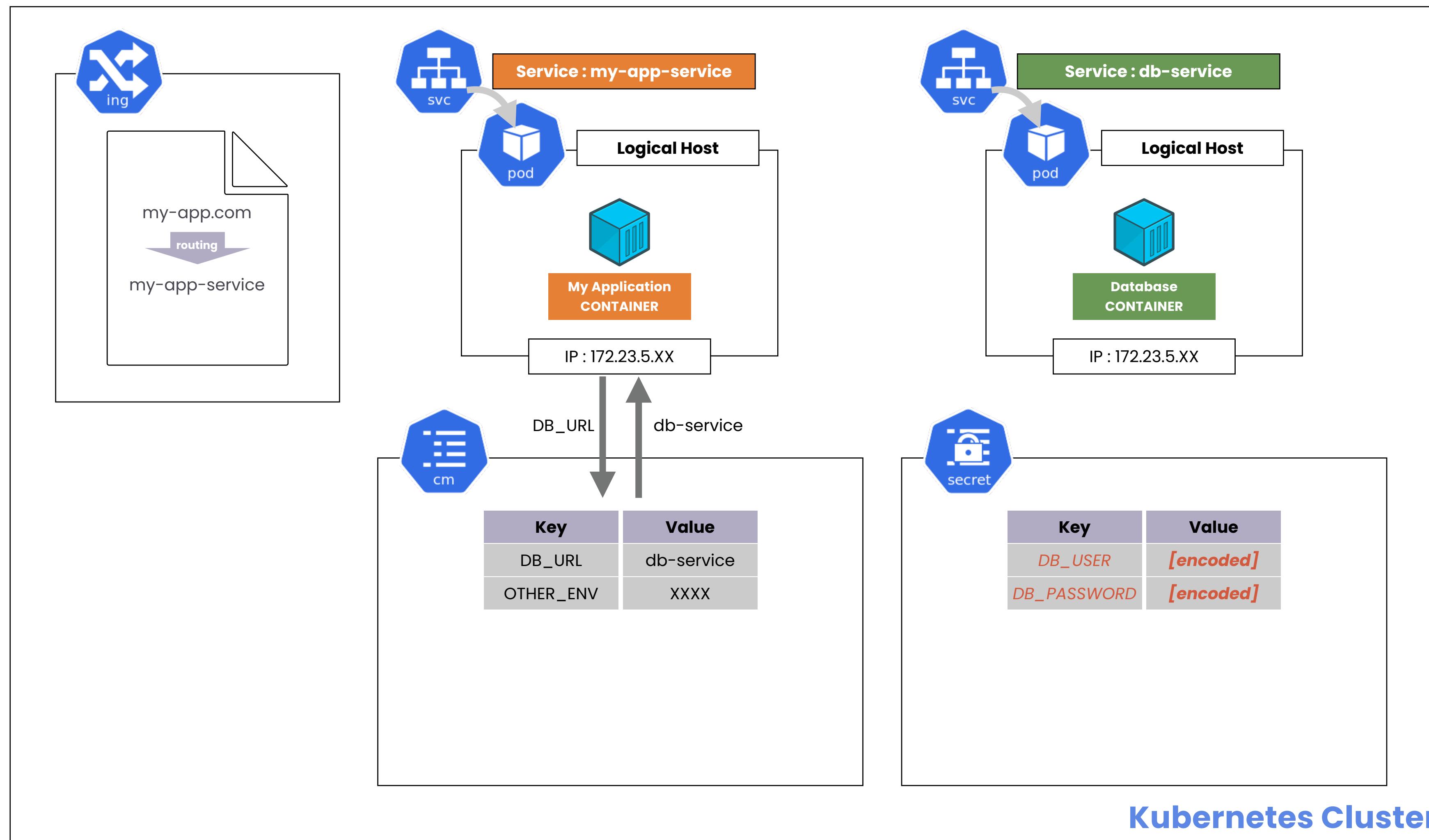
WORKING WITH SECRET



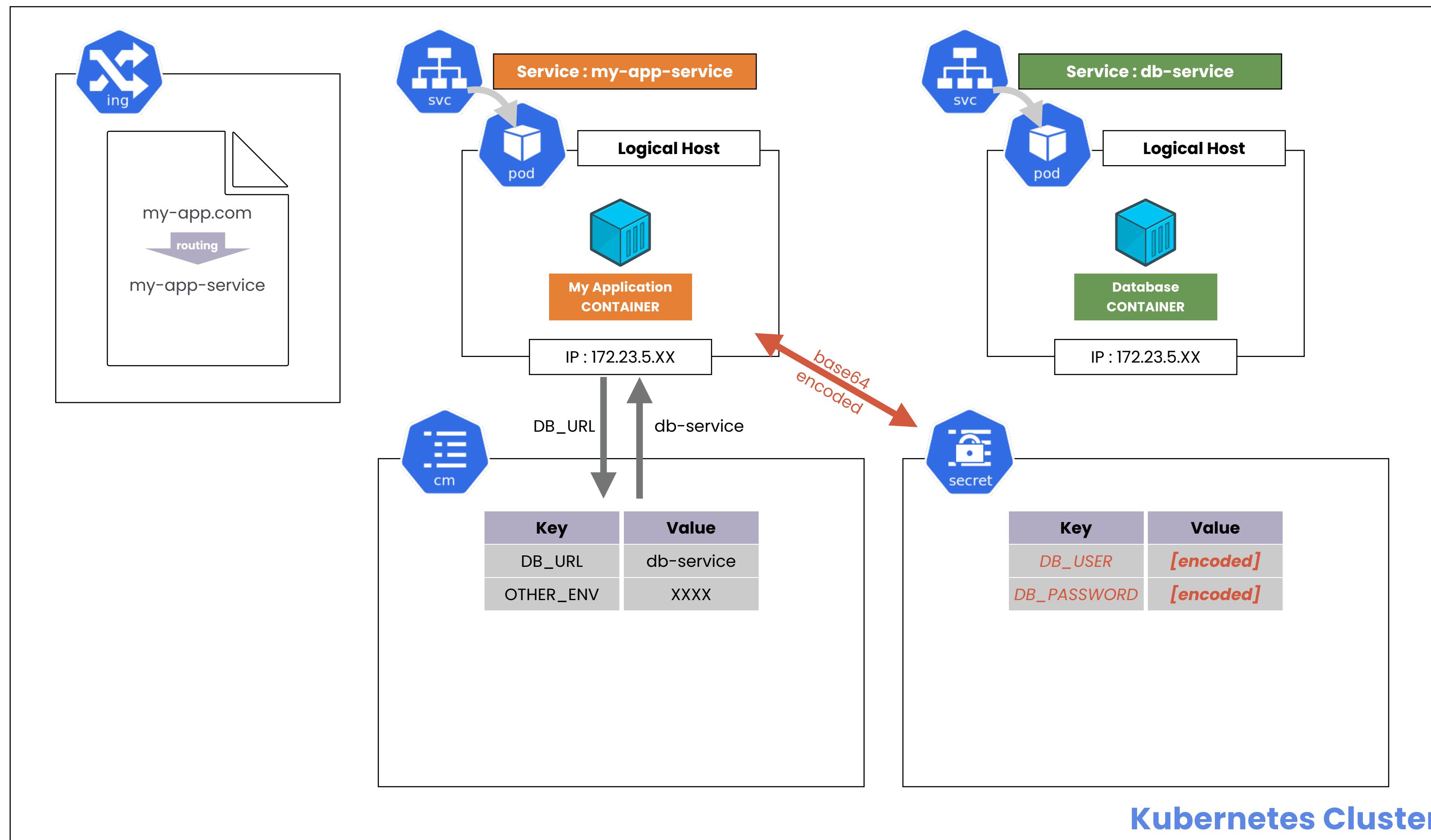
WORKING WITH SECRET



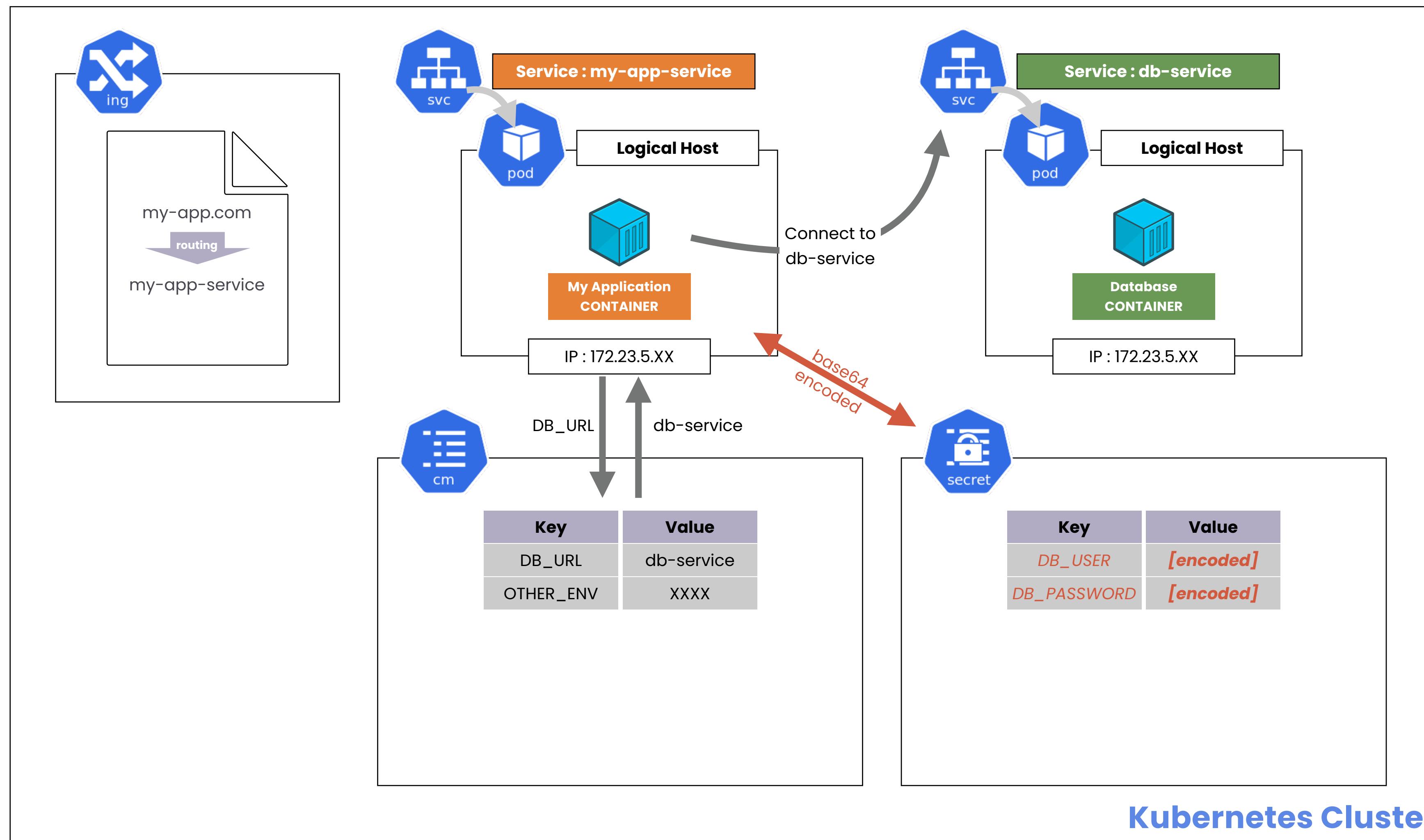
WORKING WITH SECRET



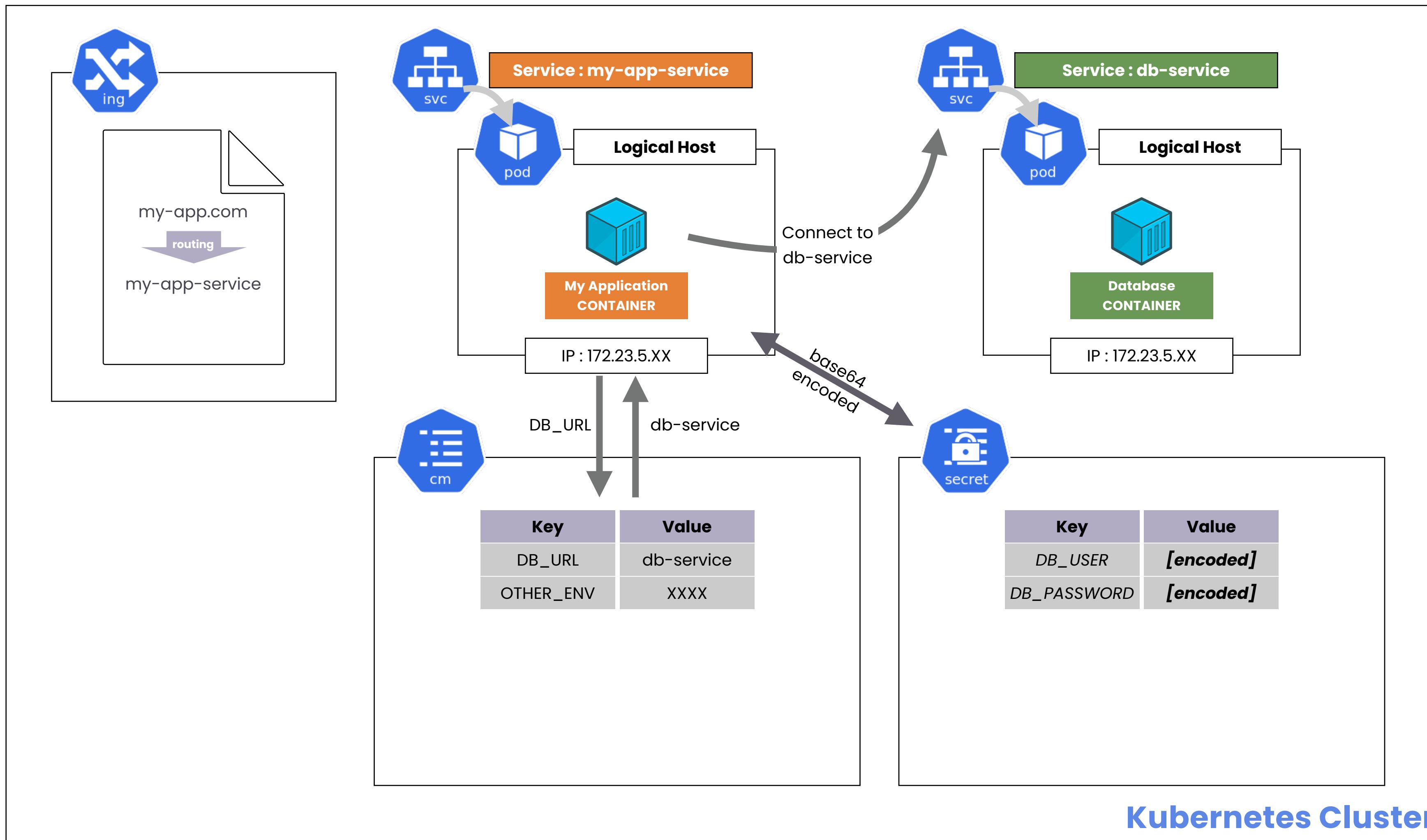
WORKING WITH SECRET



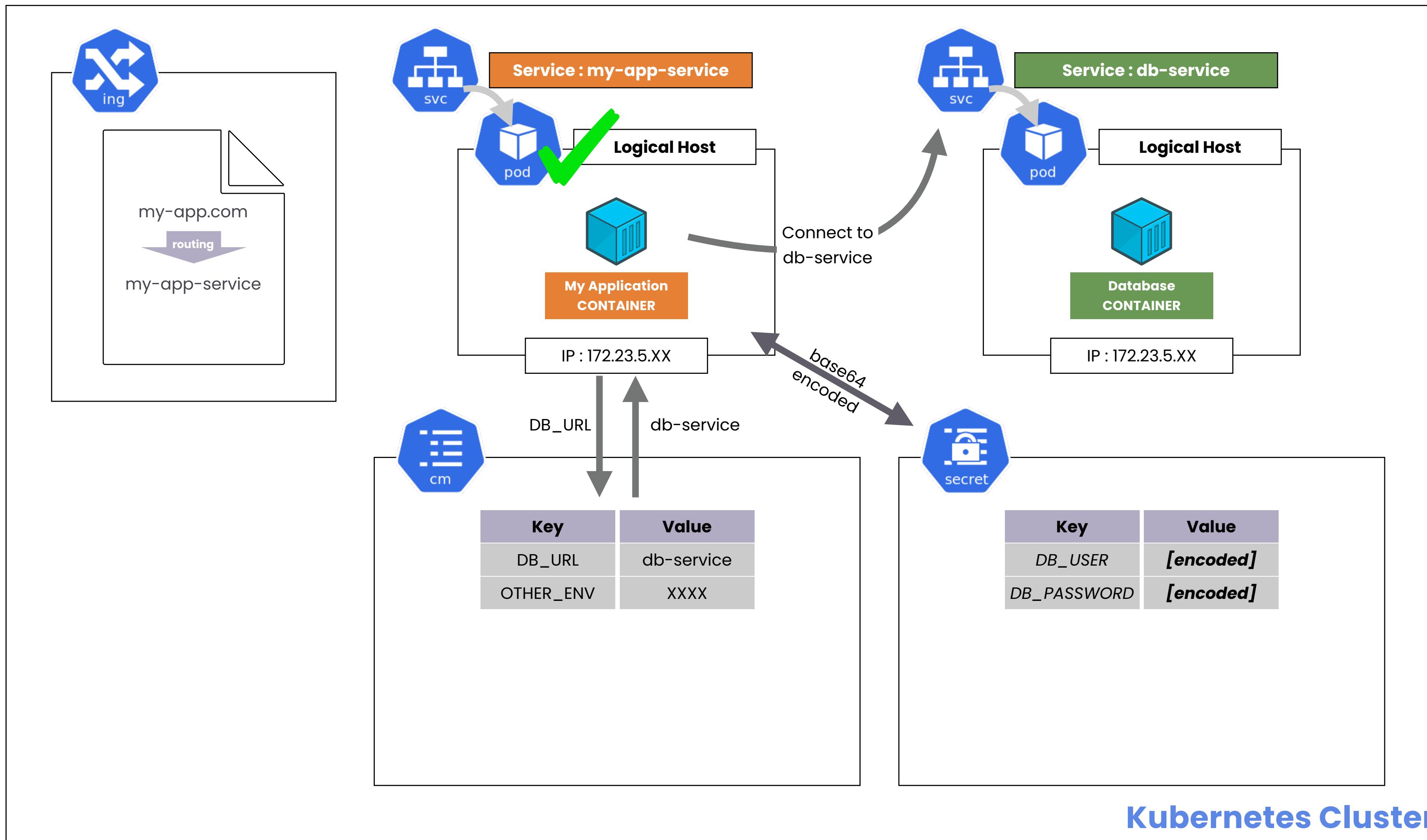
WORKING WITH SECRET



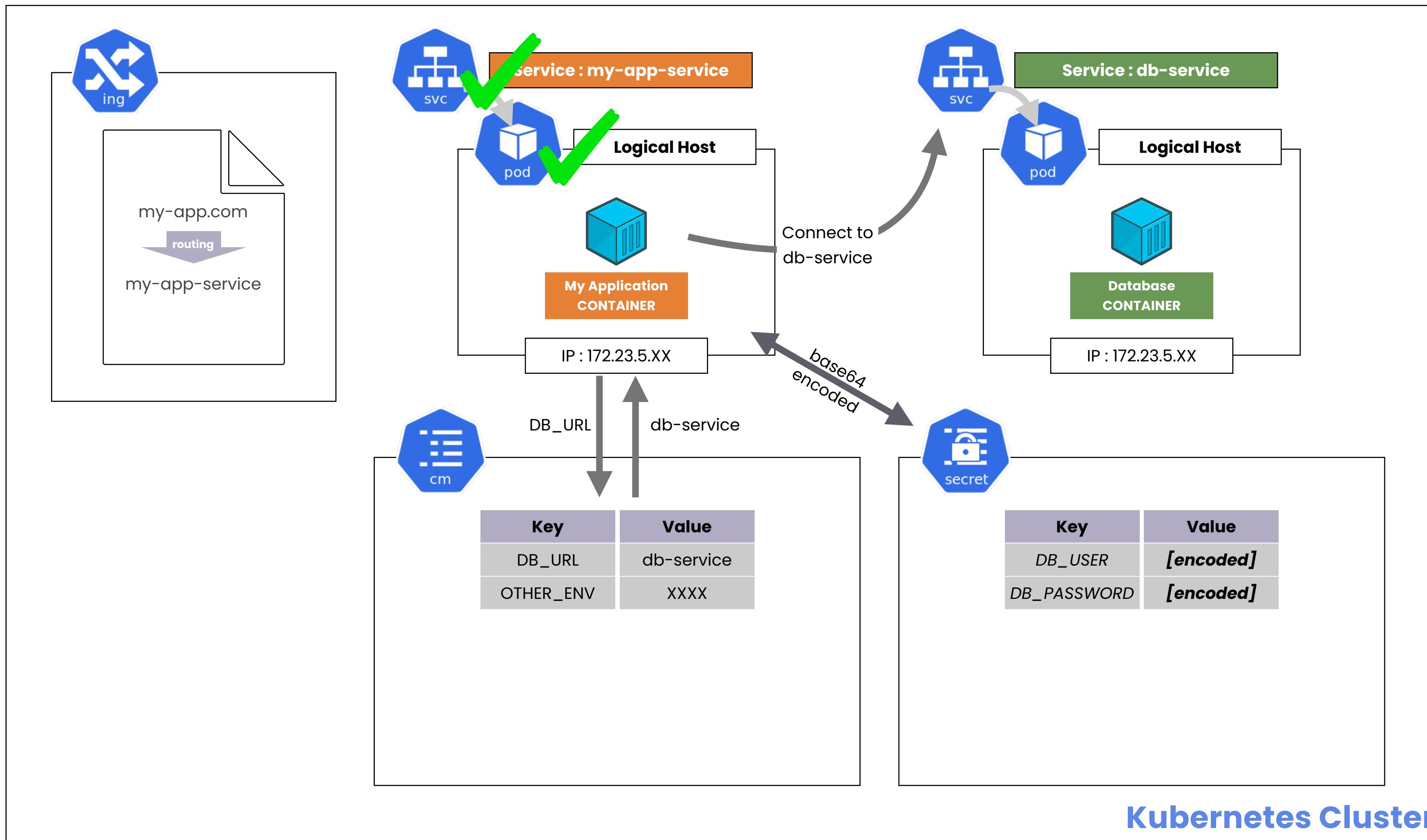
KUBERNETES'S OBJECT



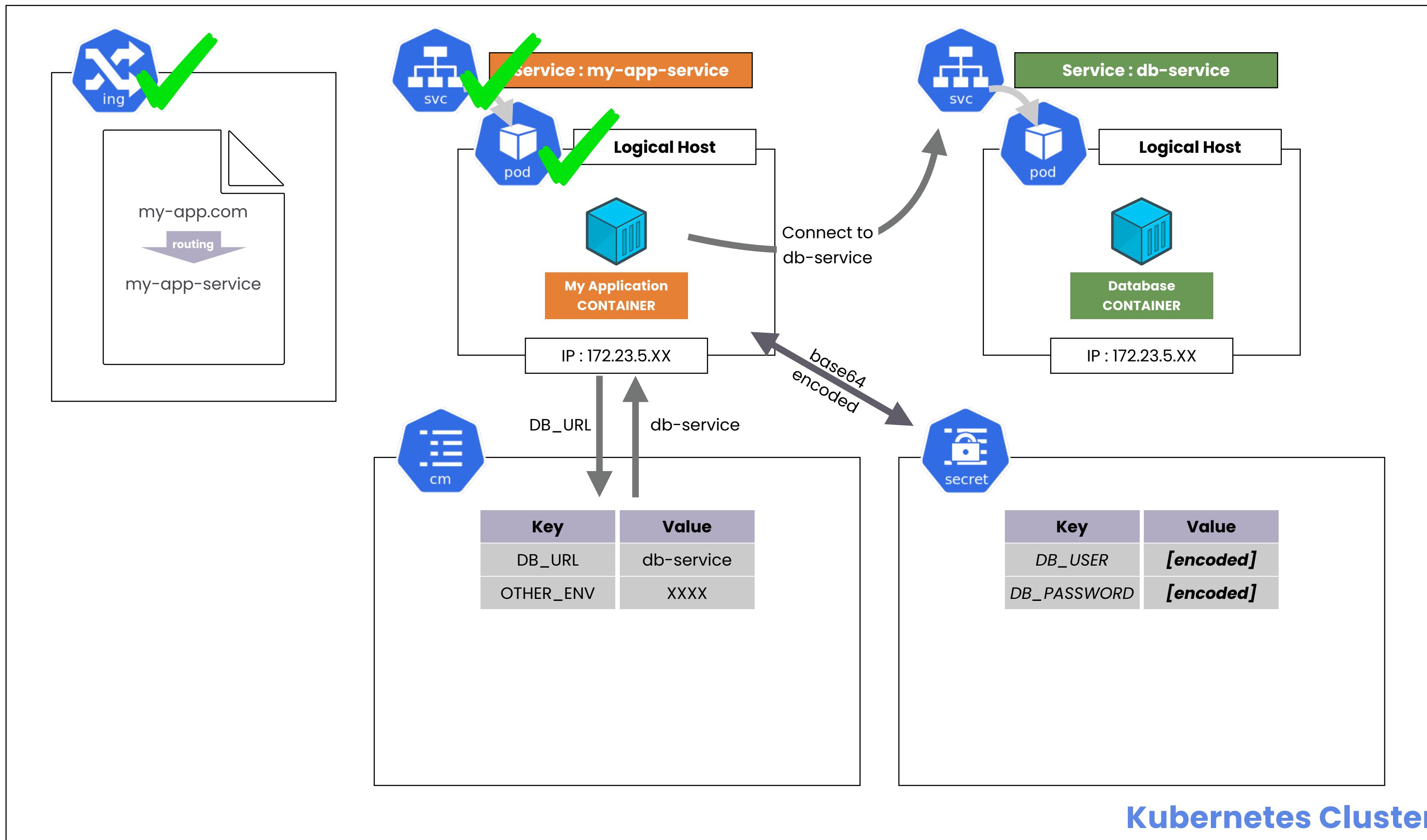
KUBERNETES'S OBJECT



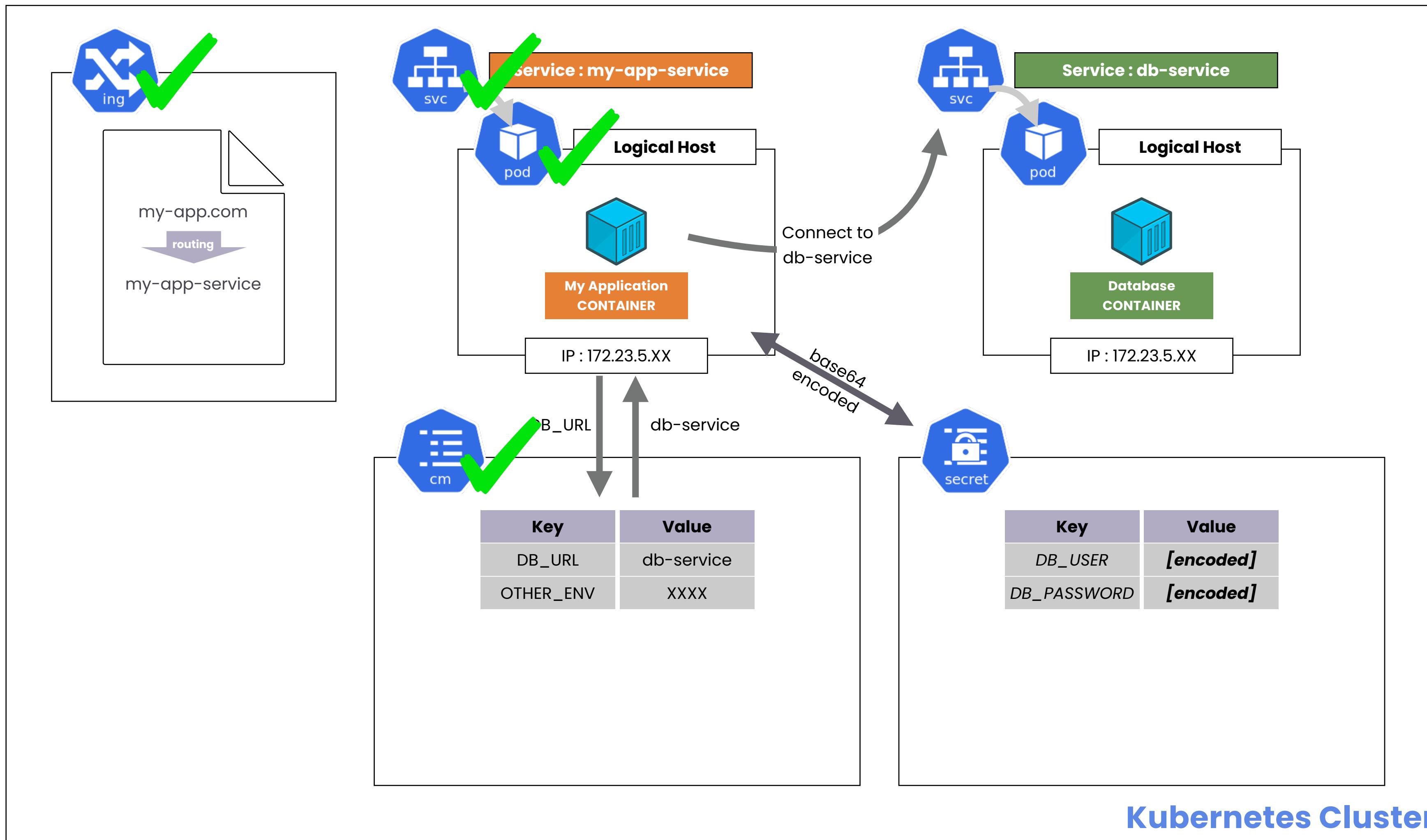
KUBERNETES'S OBJECT



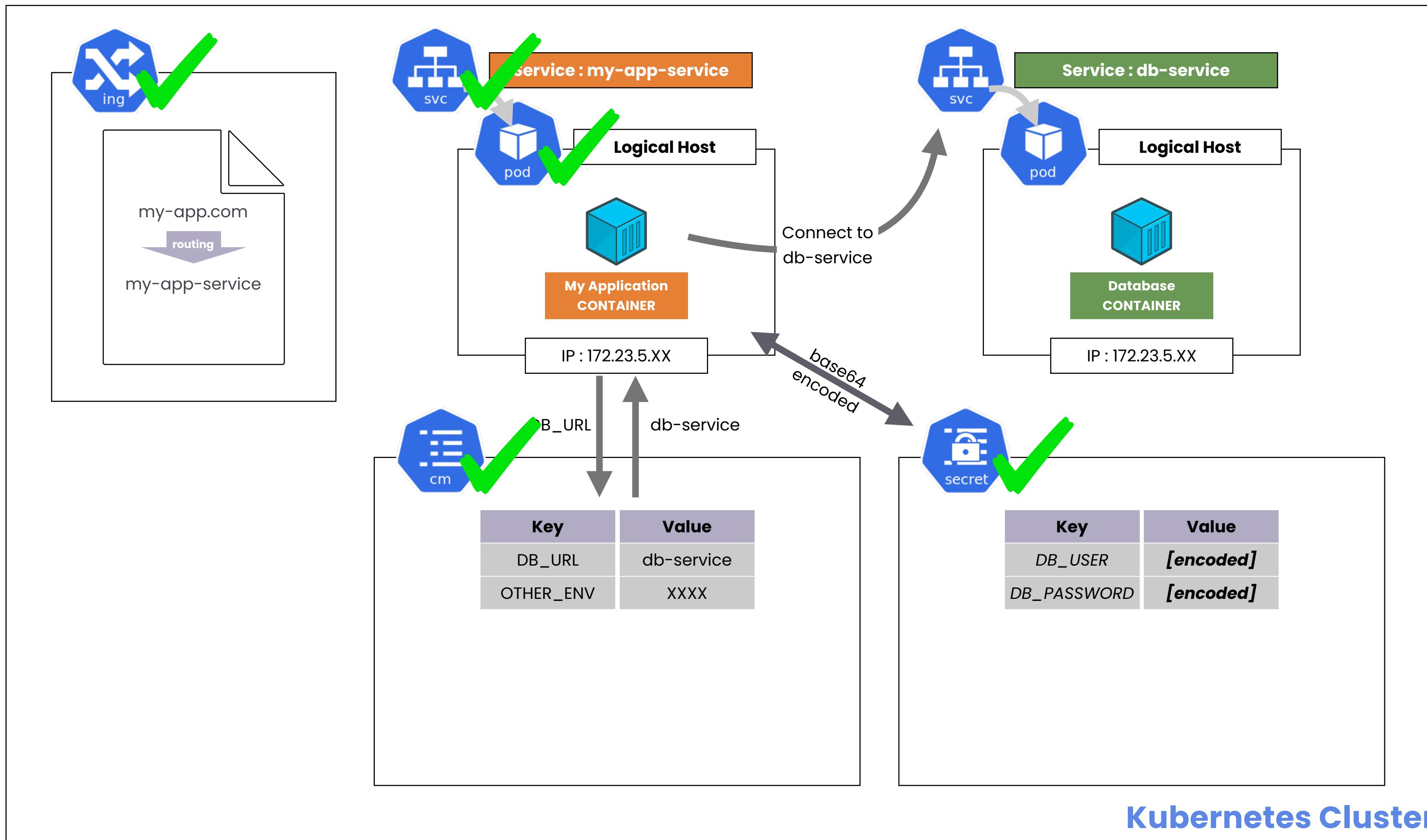
KUBERNETES'S OBJECT



KUBERNETES'S OBJECT



KUBERNETES'S OBJECT



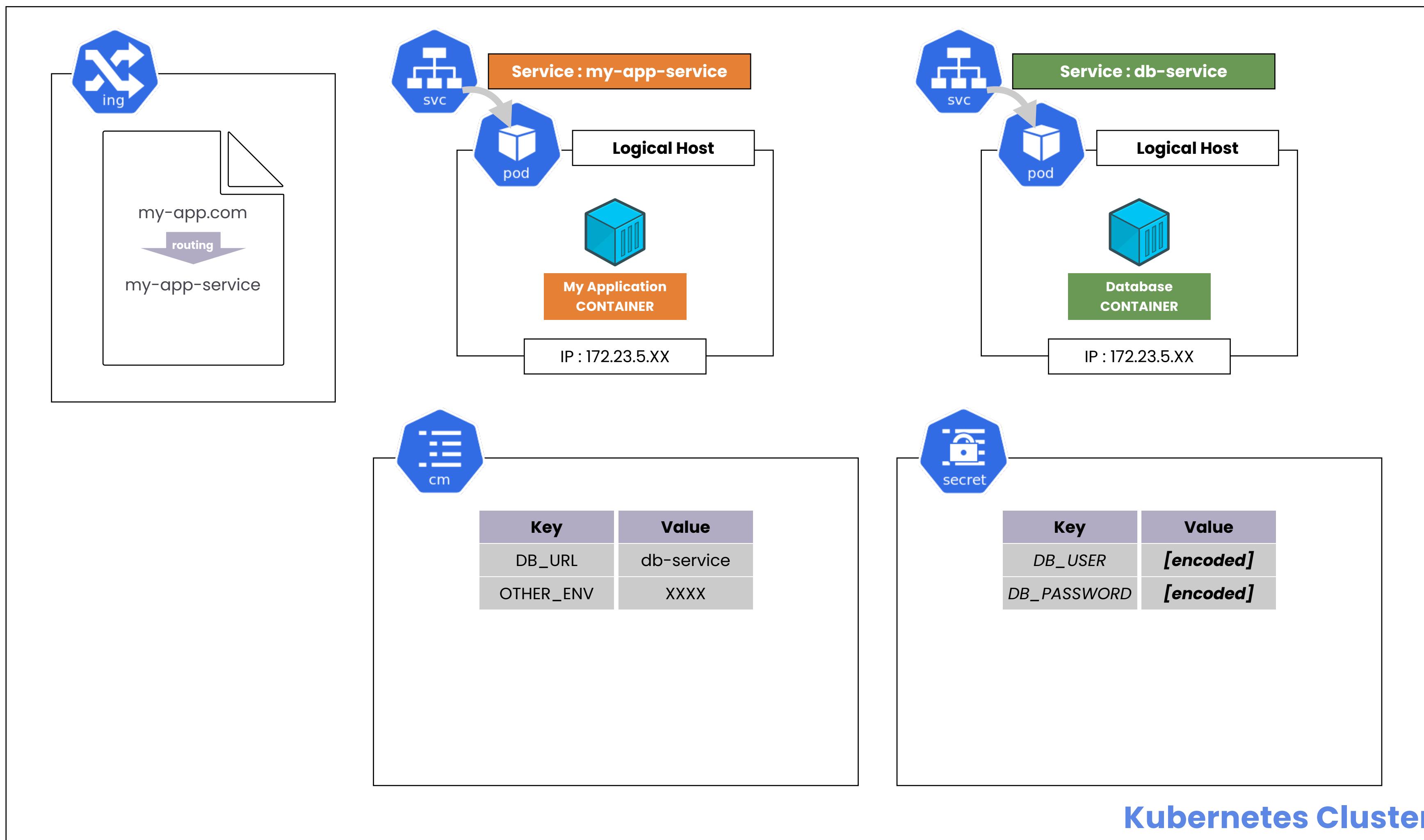
VOLUMES

VOLUME

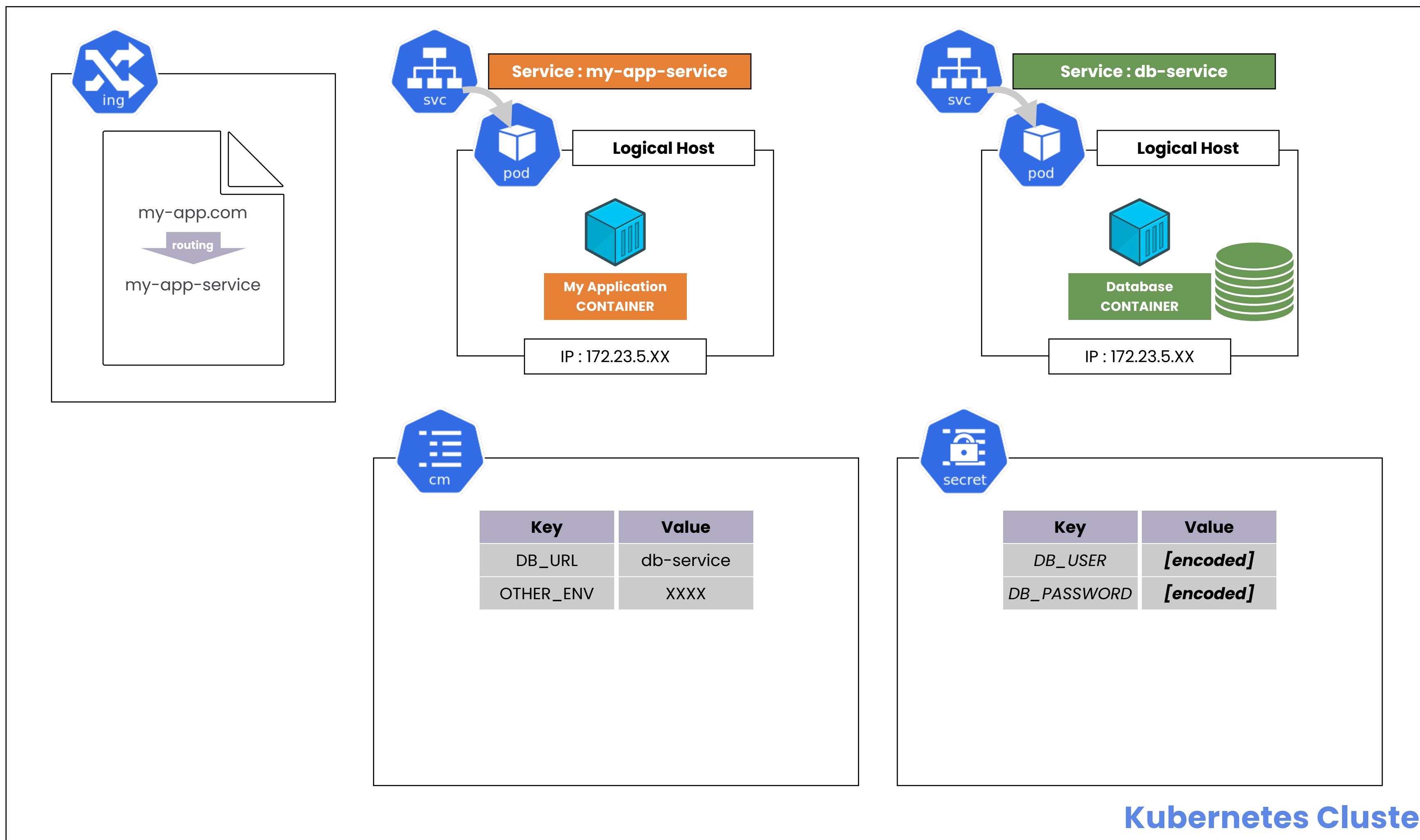
On-disk files in a container are ephemeral, which presents some problems for non-trivial applications when running in containers. One problem is the loss of files when a container crashes. A second problem occurs when sharing files between containers running together in a Pod. The Kubernetes volume abstraction solves both of these problems. Familiarity with Pods is suggested.

Kubernetes supports many types of volumes. A Pod can use any number of volume types simultaneously. Ephemeral volume types have a lifetime of a pod, but persistent volumes exist beyond the lifetime of a pod. When a pod ceases to exist, Kubernetes destroys ephemeral volumes; however, Kubernetes does not destroy persistent volumes. For any kind of volume in a given pod, data is preserved across container restarts.

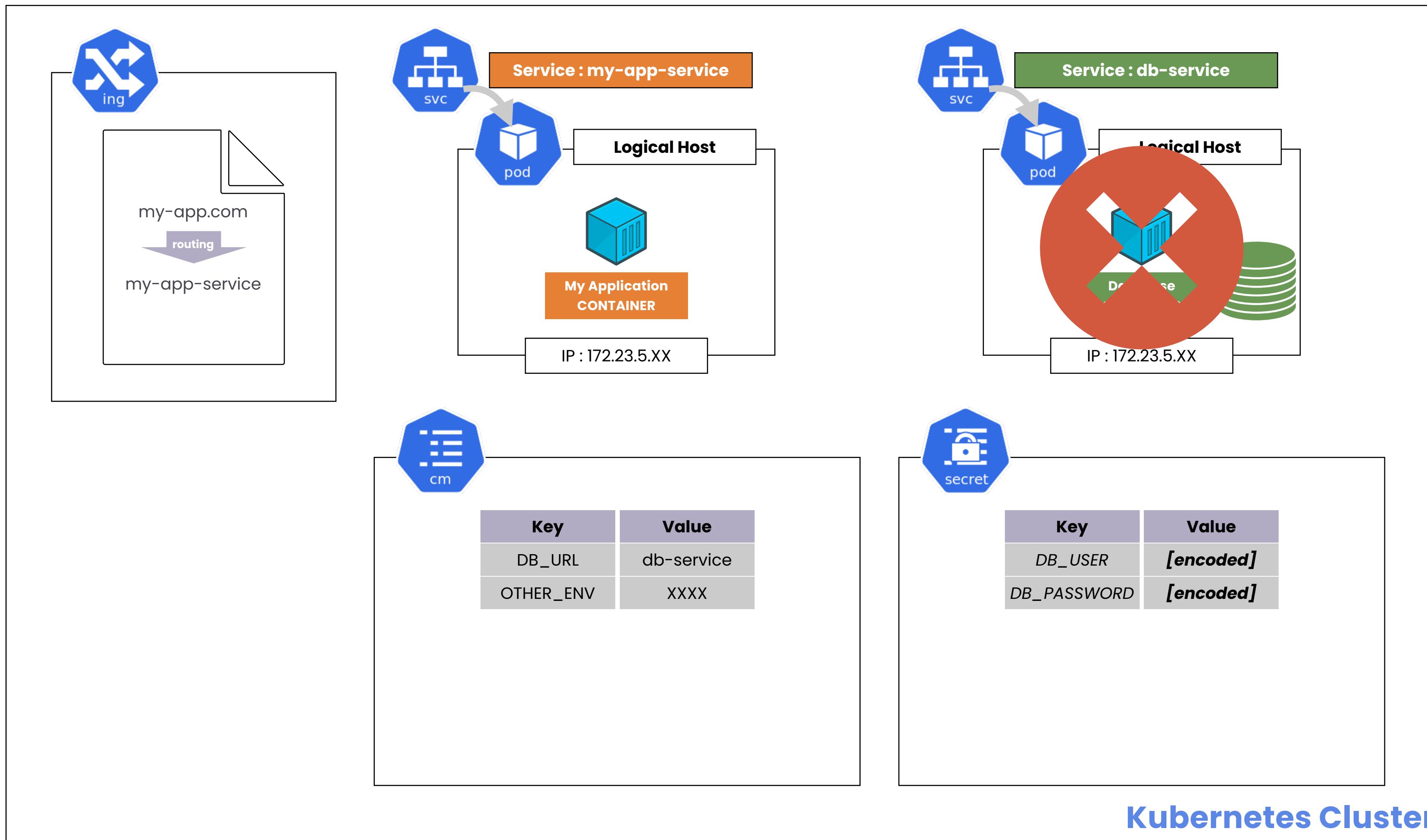
DATA STORAGE IN CONTAINERS



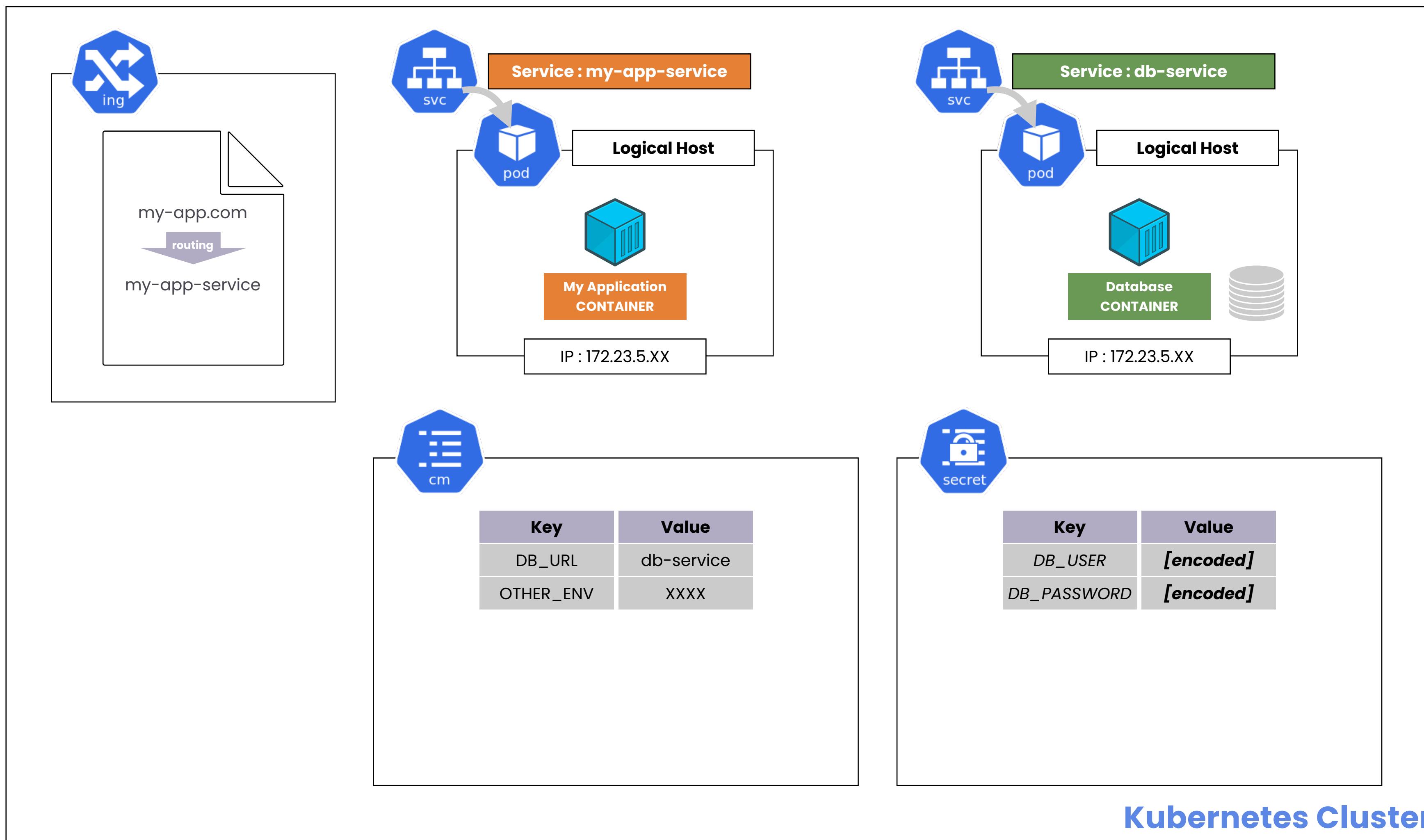
DATA STORAGE IN CONTAINERS



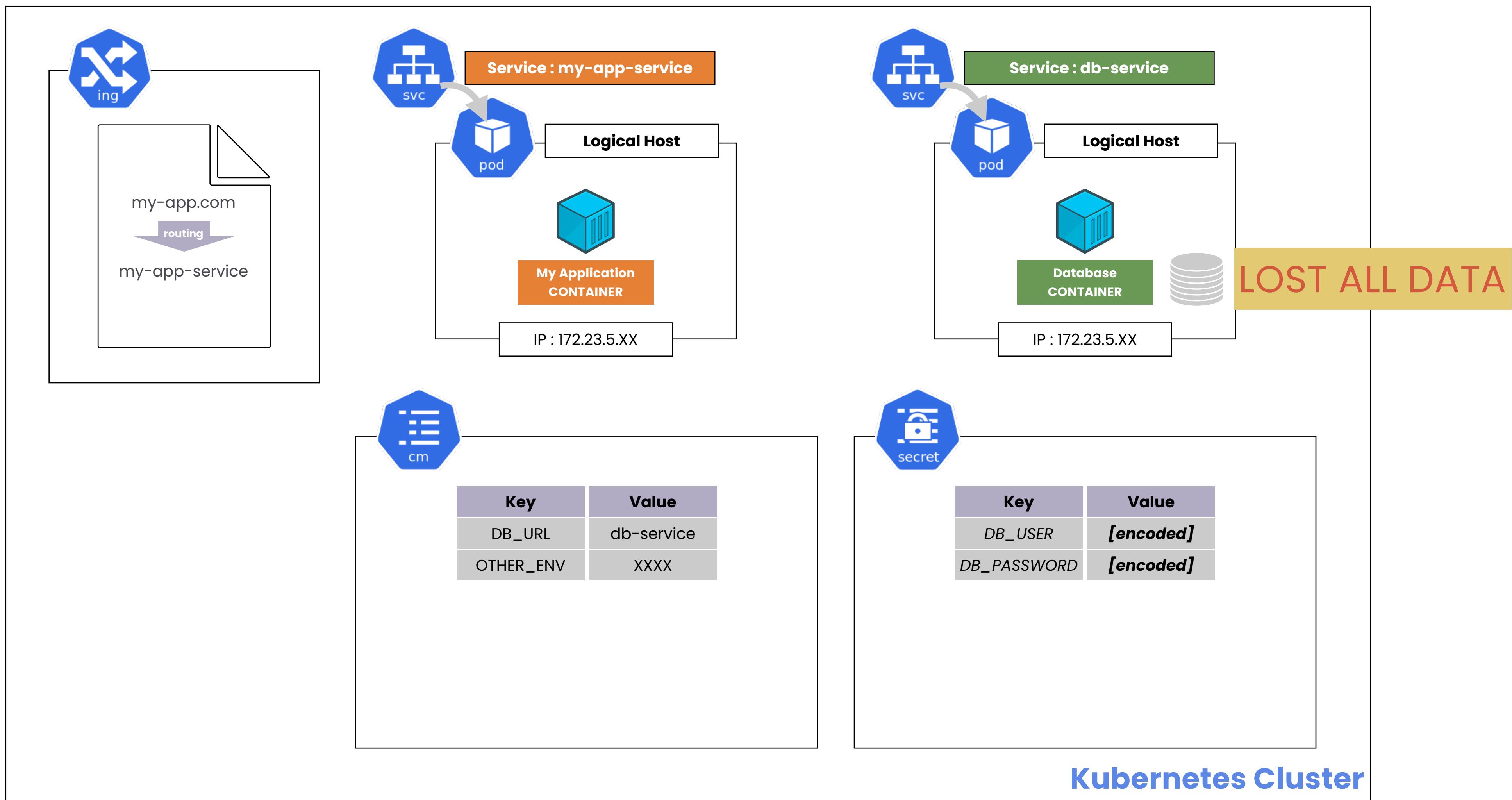
DATA STORAGE IN CONTAINERS



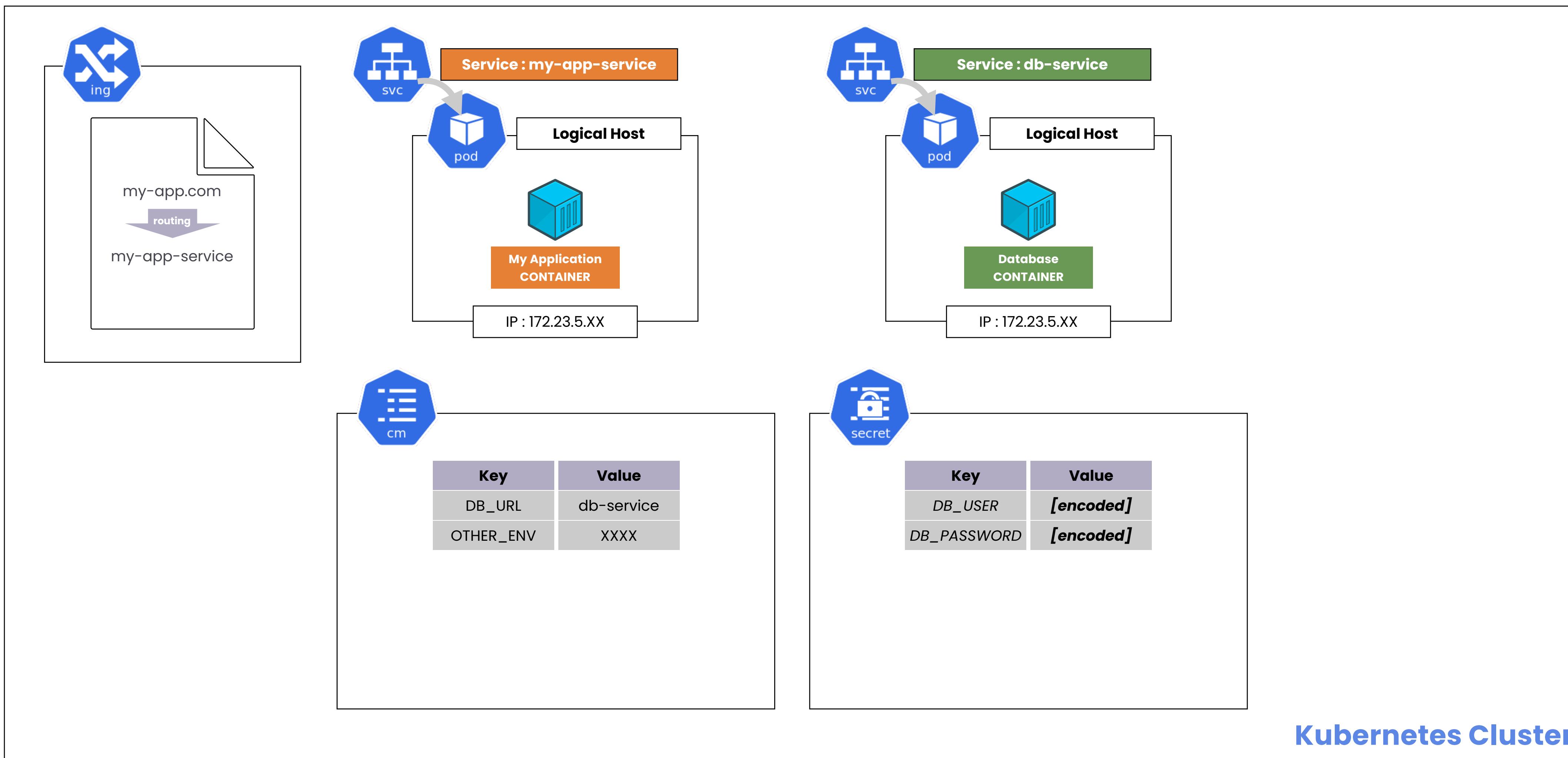
DATA STORAGE IN CONTAINERS



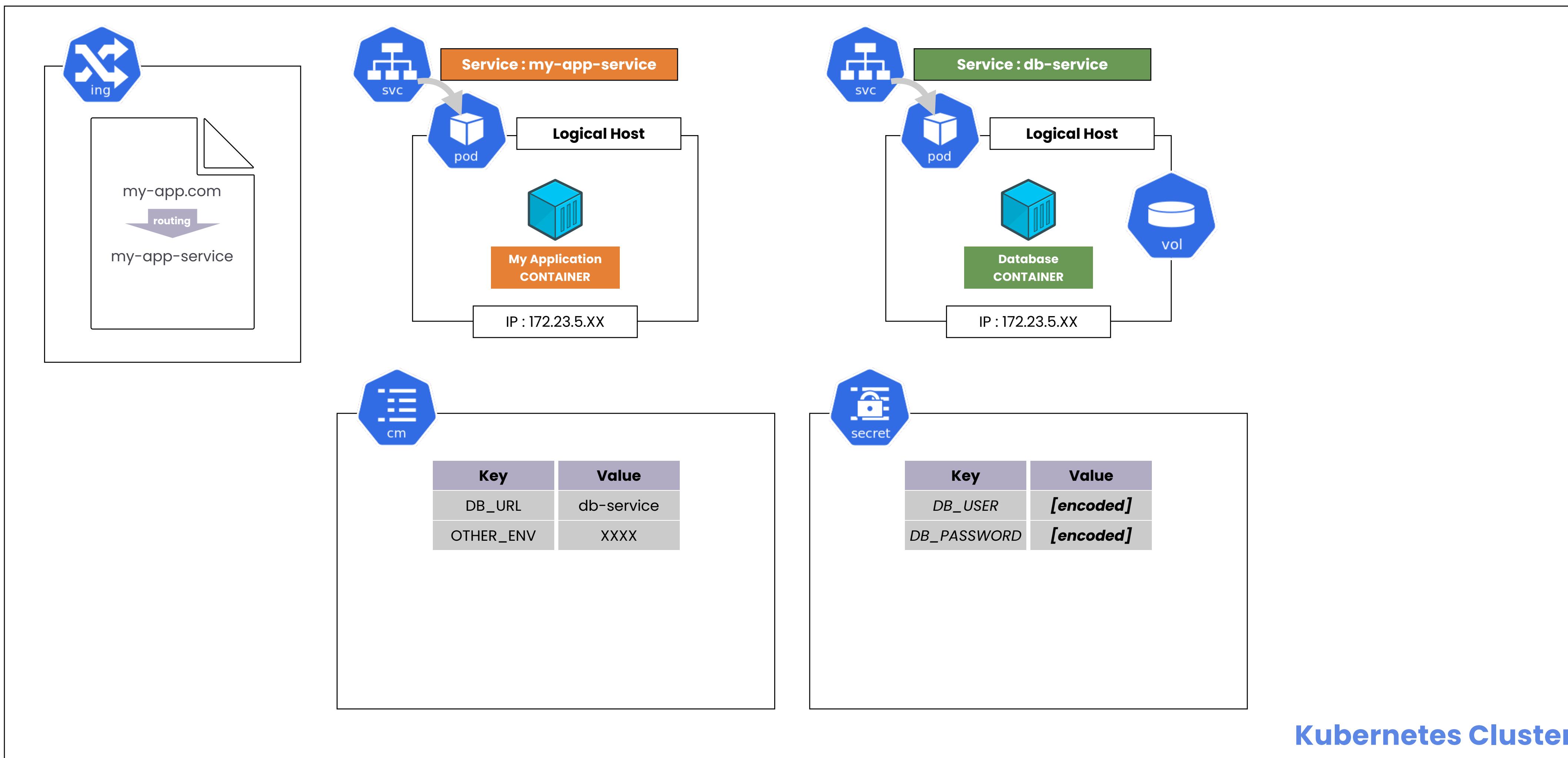
DATA STORAGE IN CONTAINERS



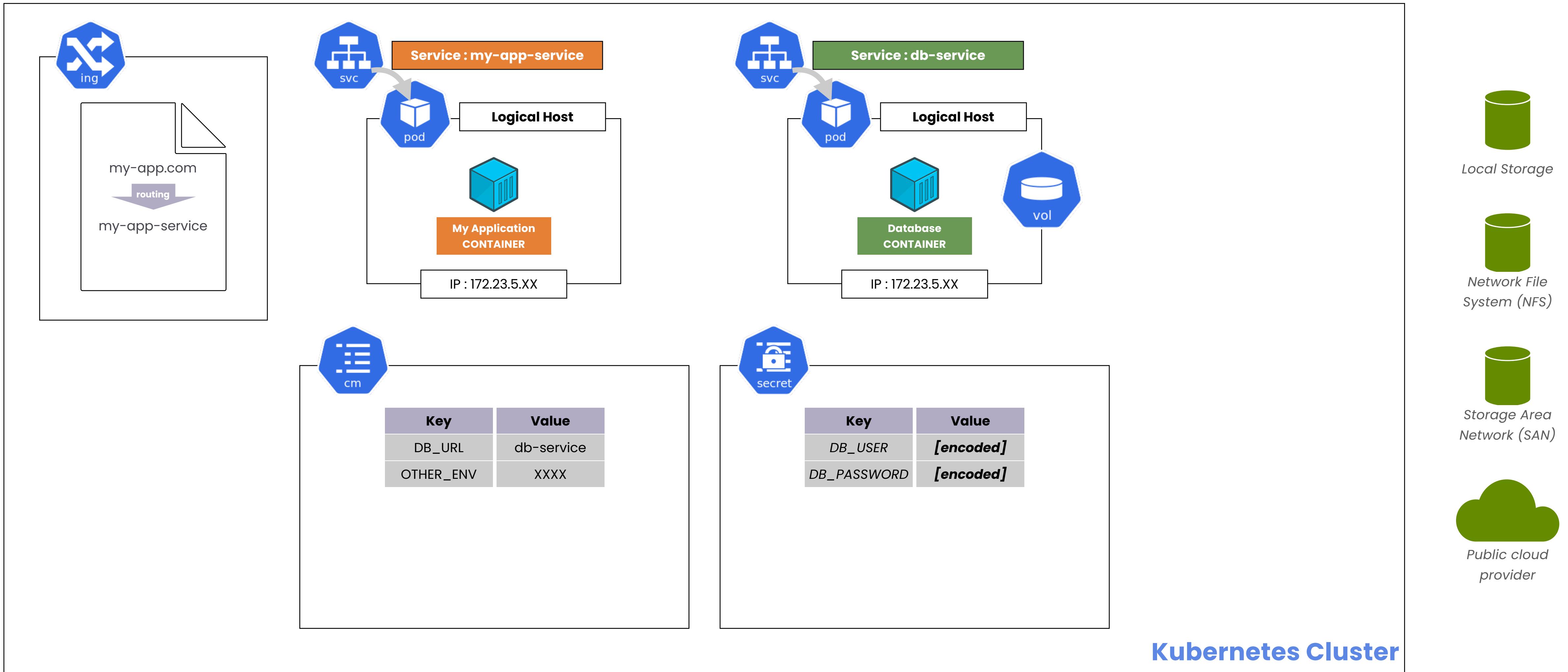
KUBERNETES'S OBJECT



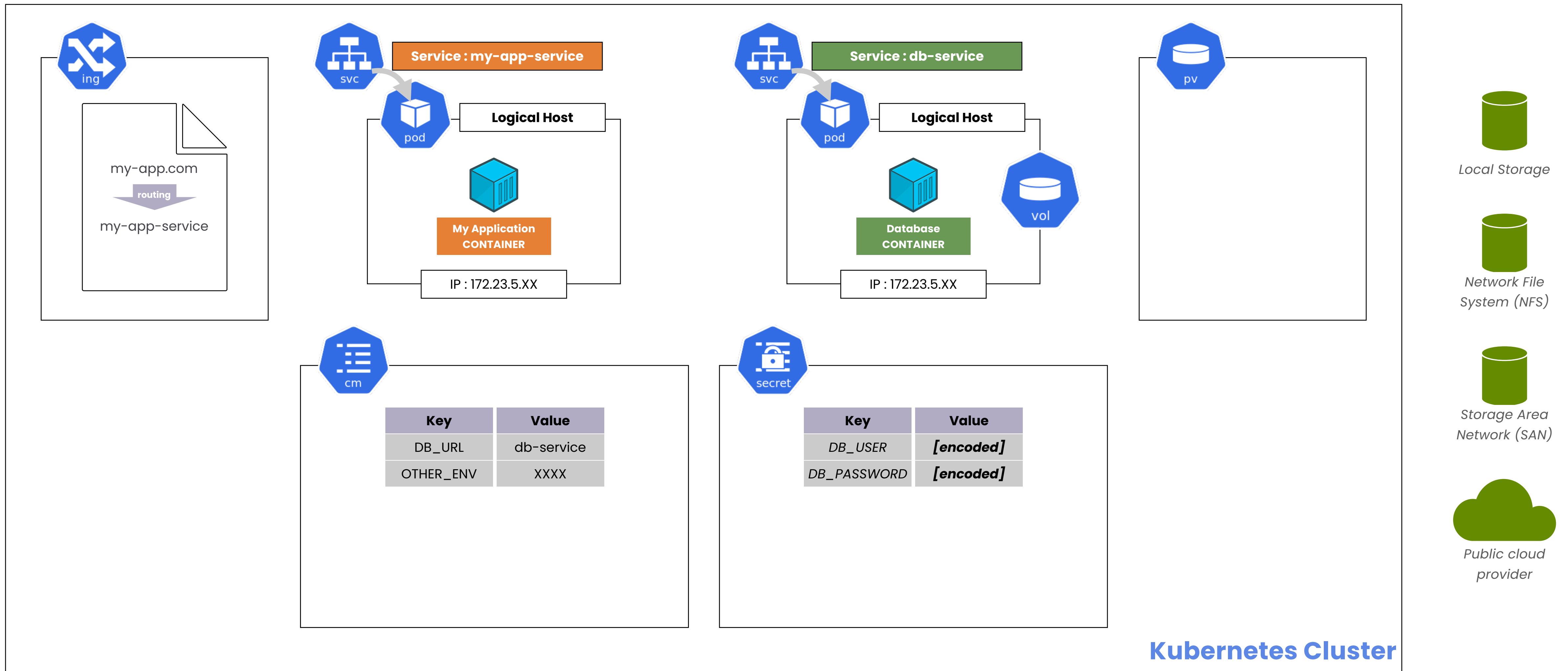
KUBERNETES'S OBJECT



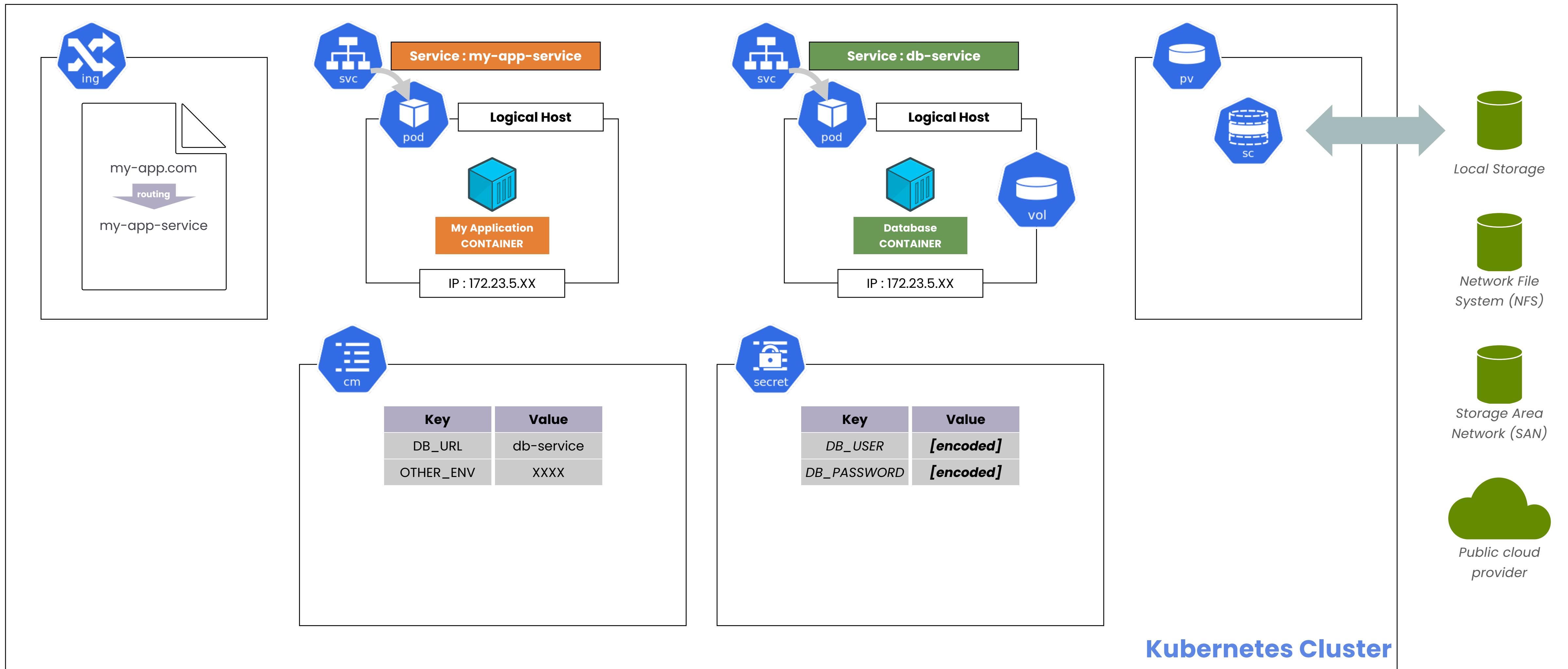
KUBERNETES'S OBJECT



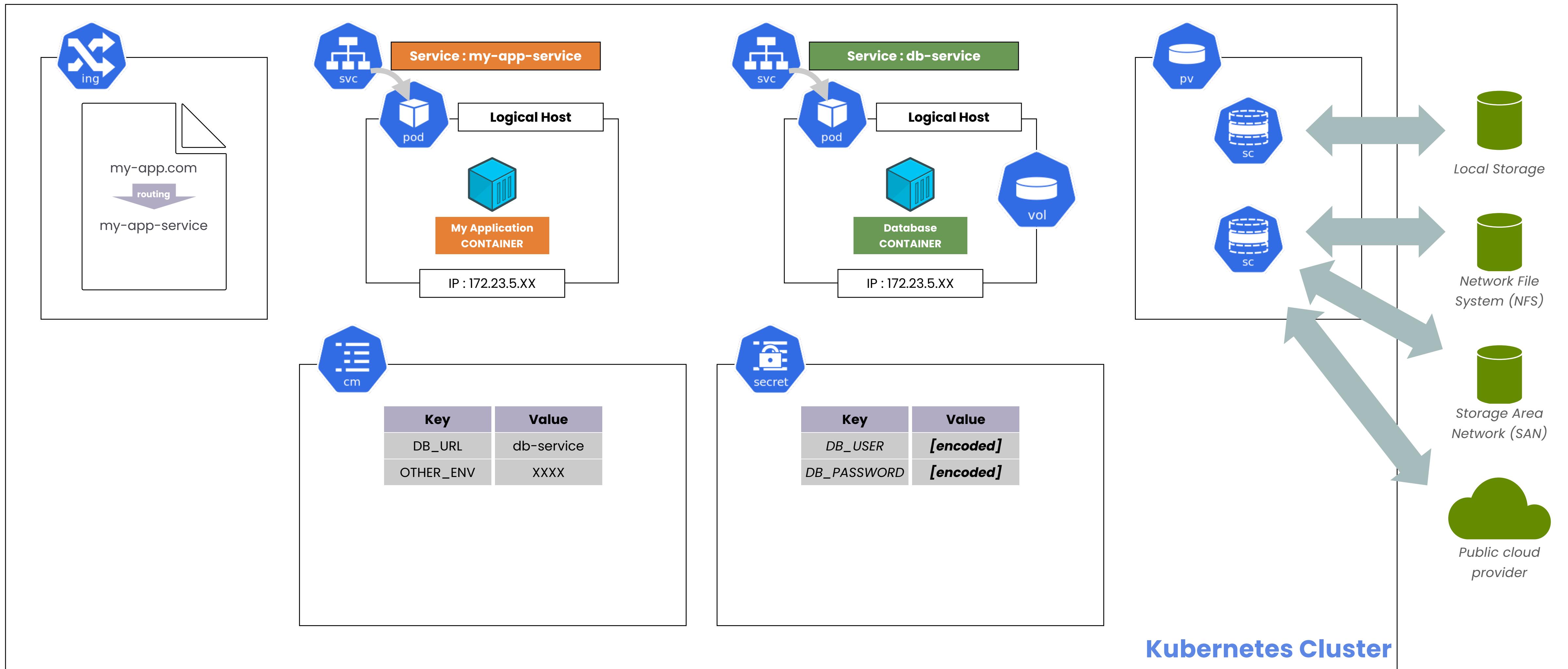
KUBERNETES'S OBJECT



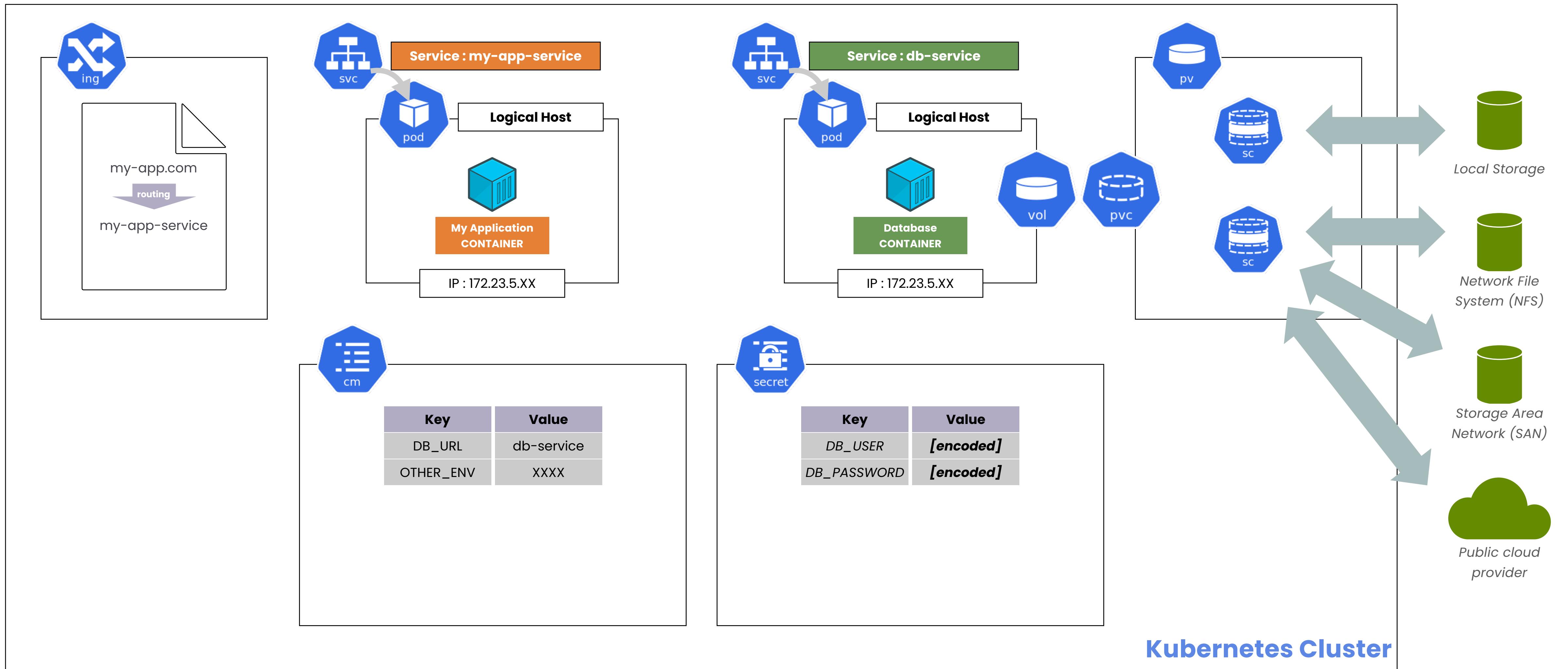
KUBERNETES'S OBJECT



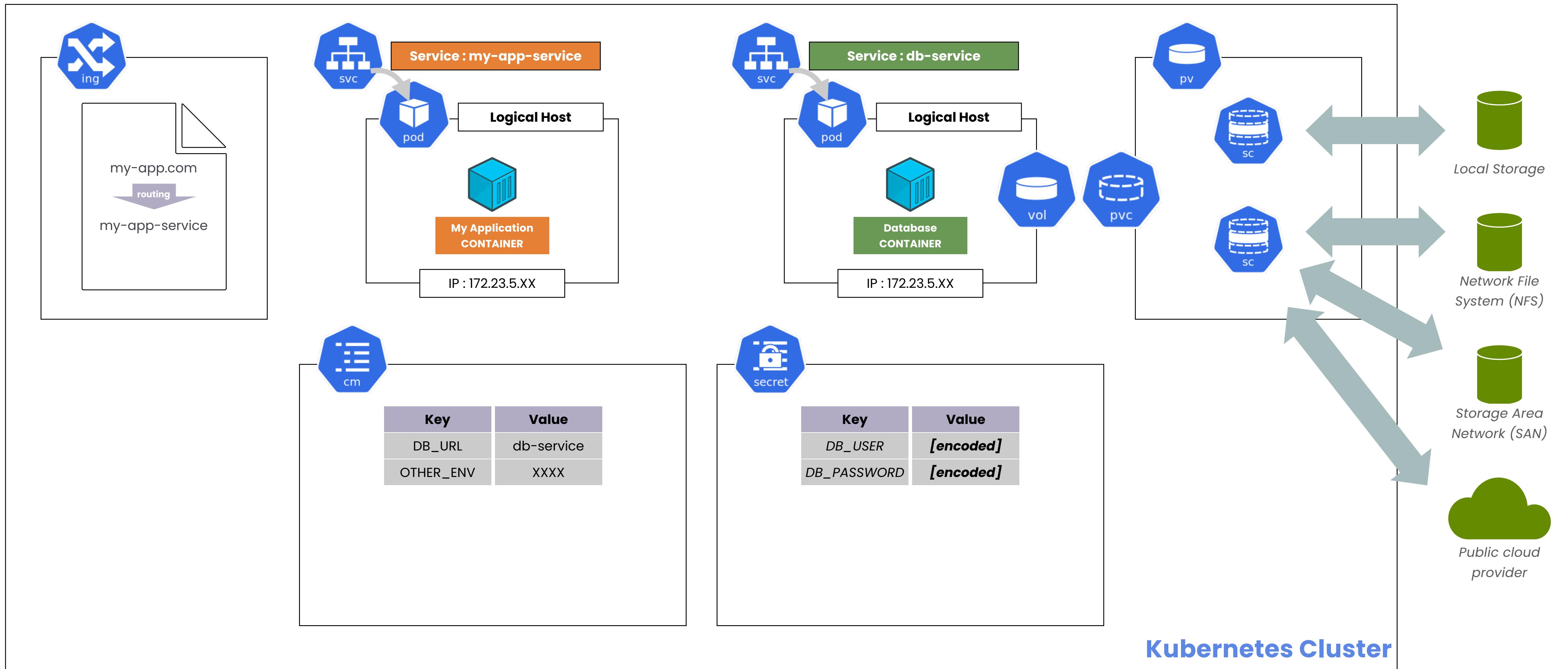
KUBERNETES'S OBJECT



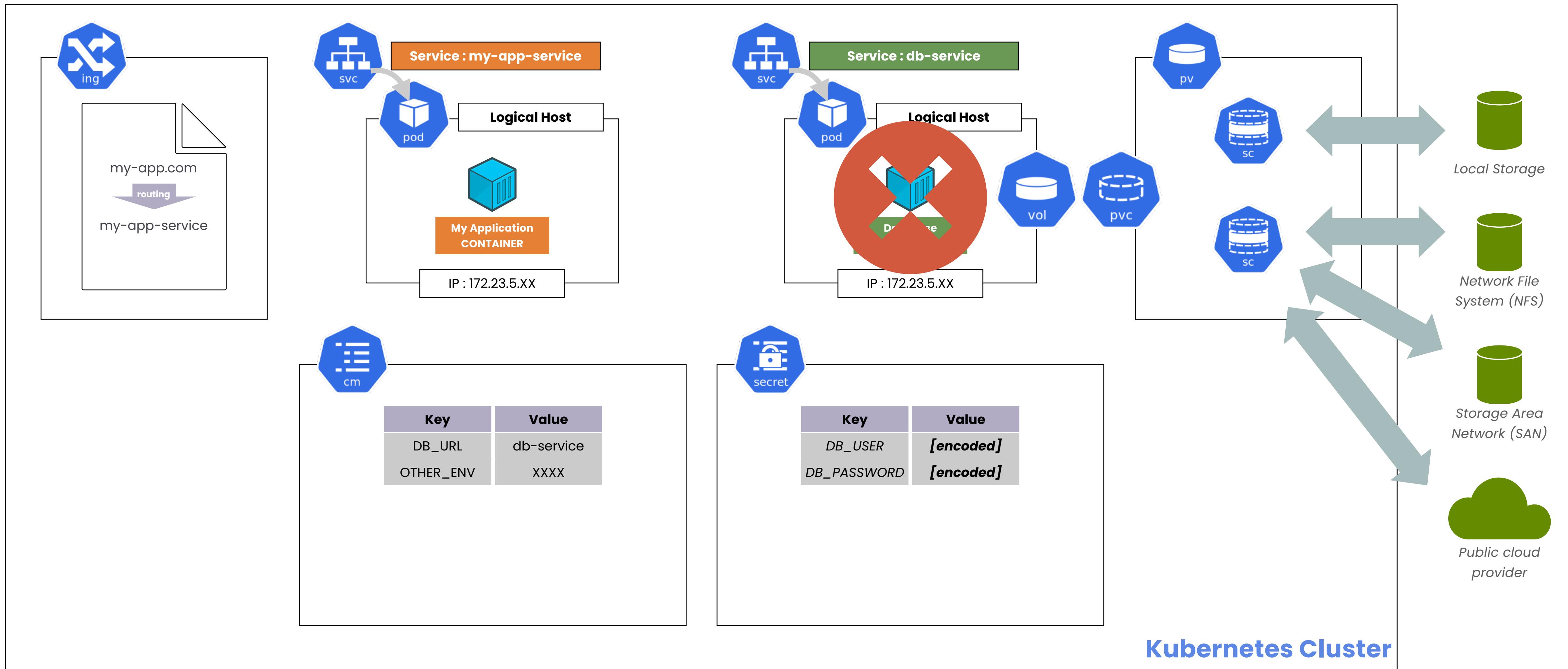
KUBERNETES'S OBJECT



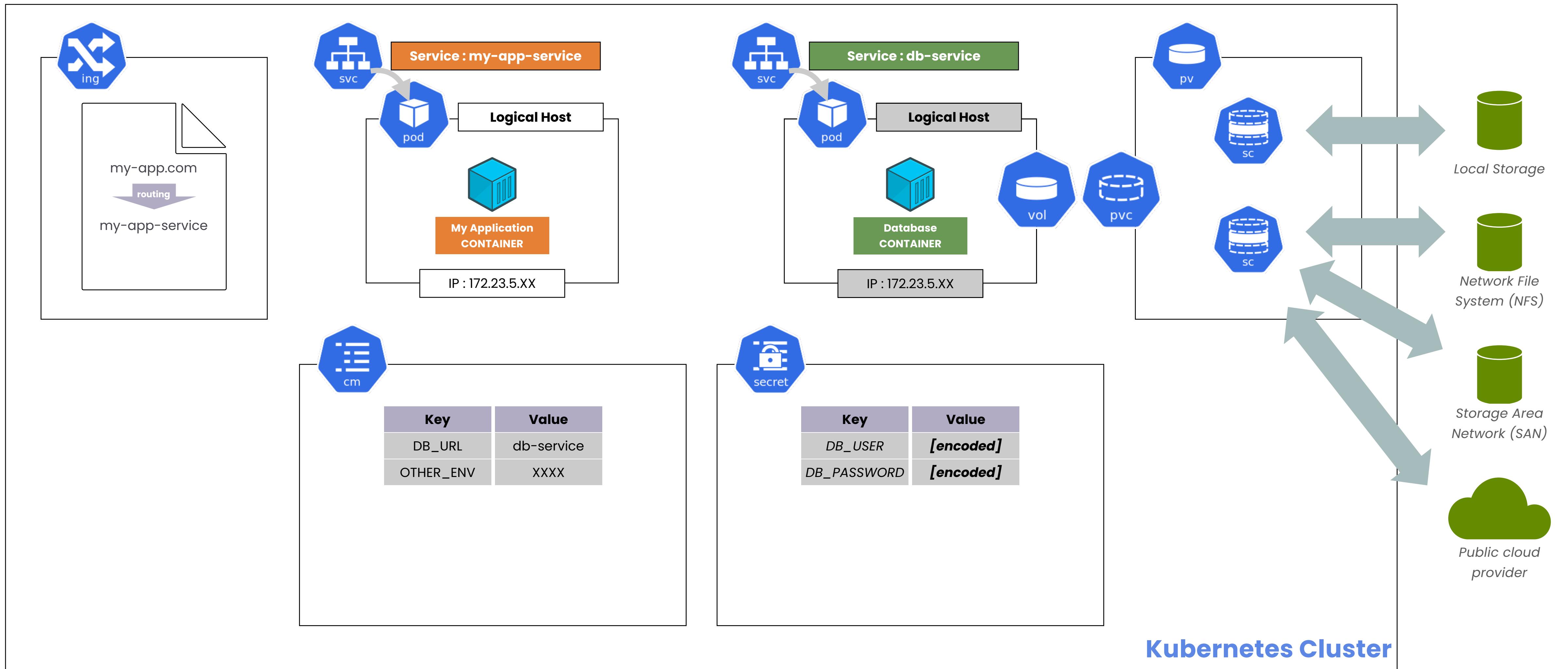
KUBERNETES'S OBJECT



KUBERNETES'S OBJECT

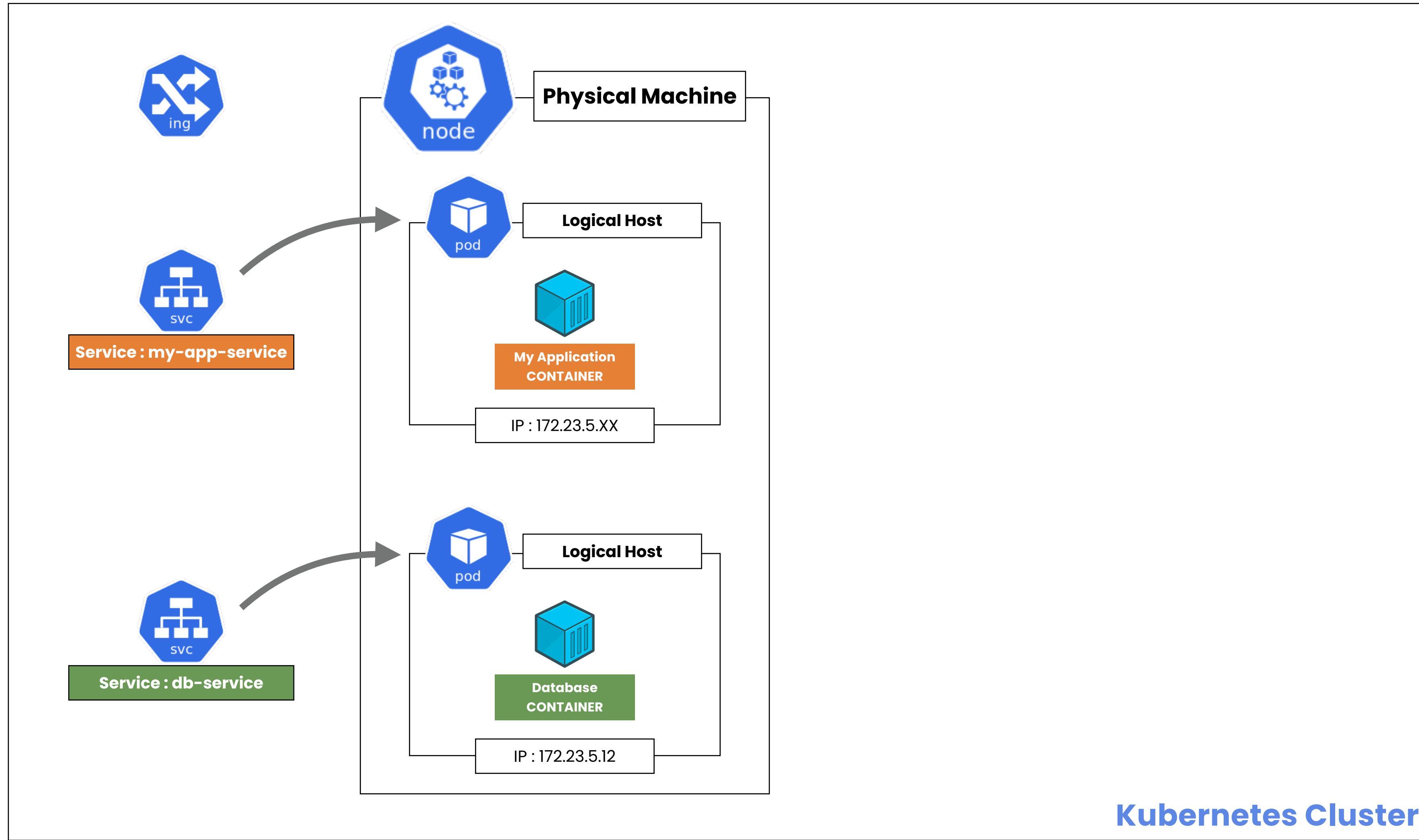


KUBERNETES'S OBJECT

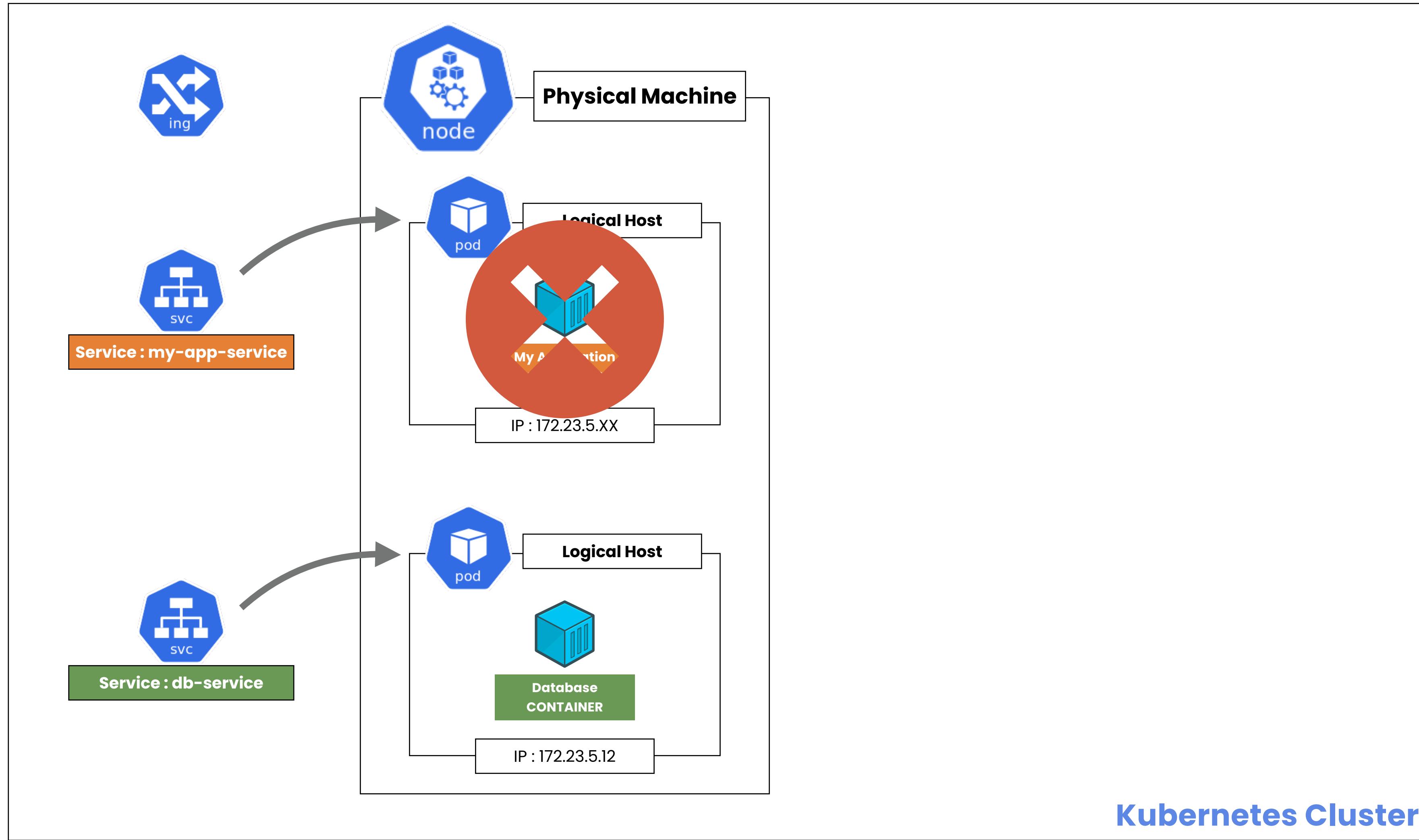


K8S WORKLOADS DEPLOYMENT & STATEFULSET

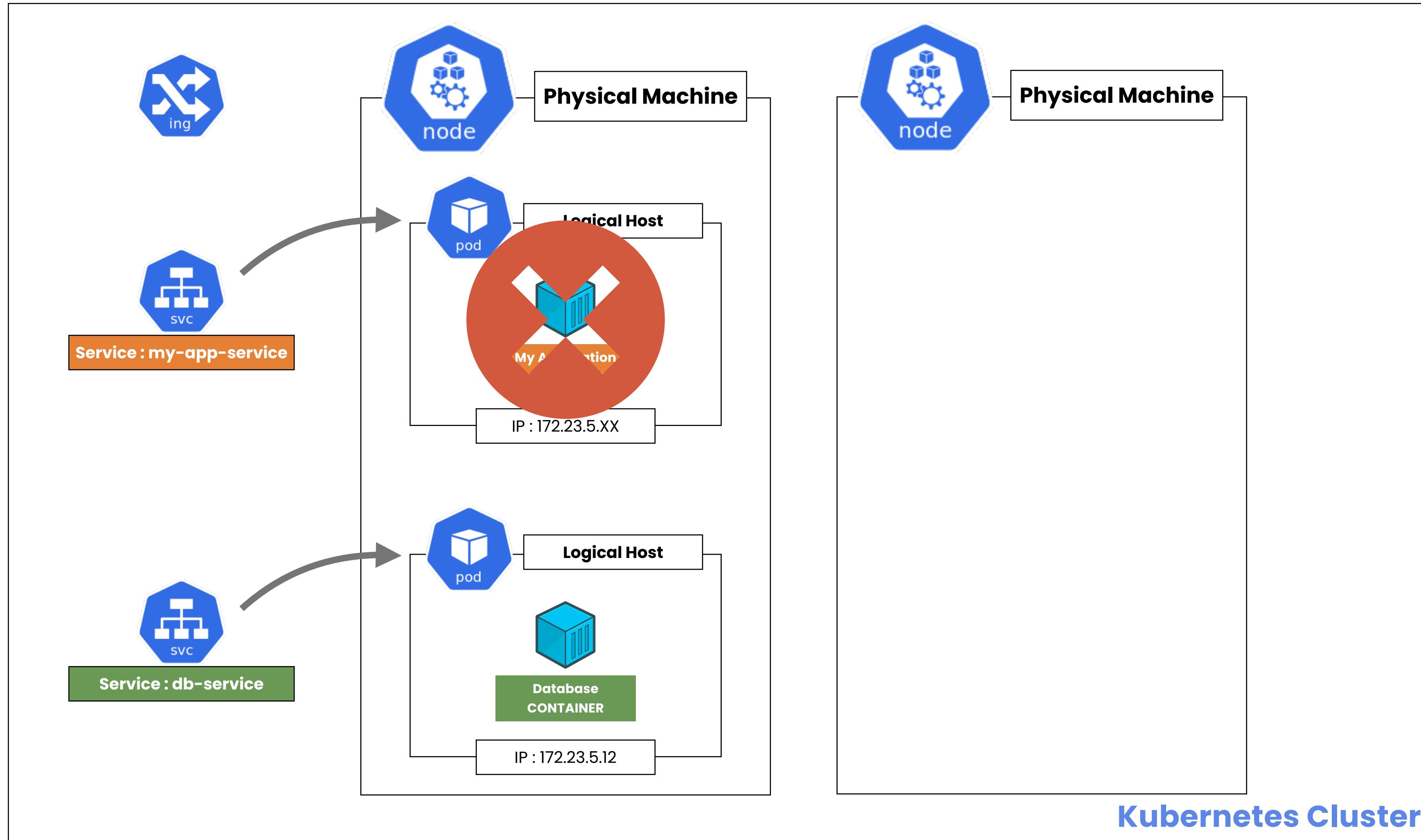
DEPLOYMENTS



DEPLOYMENTS

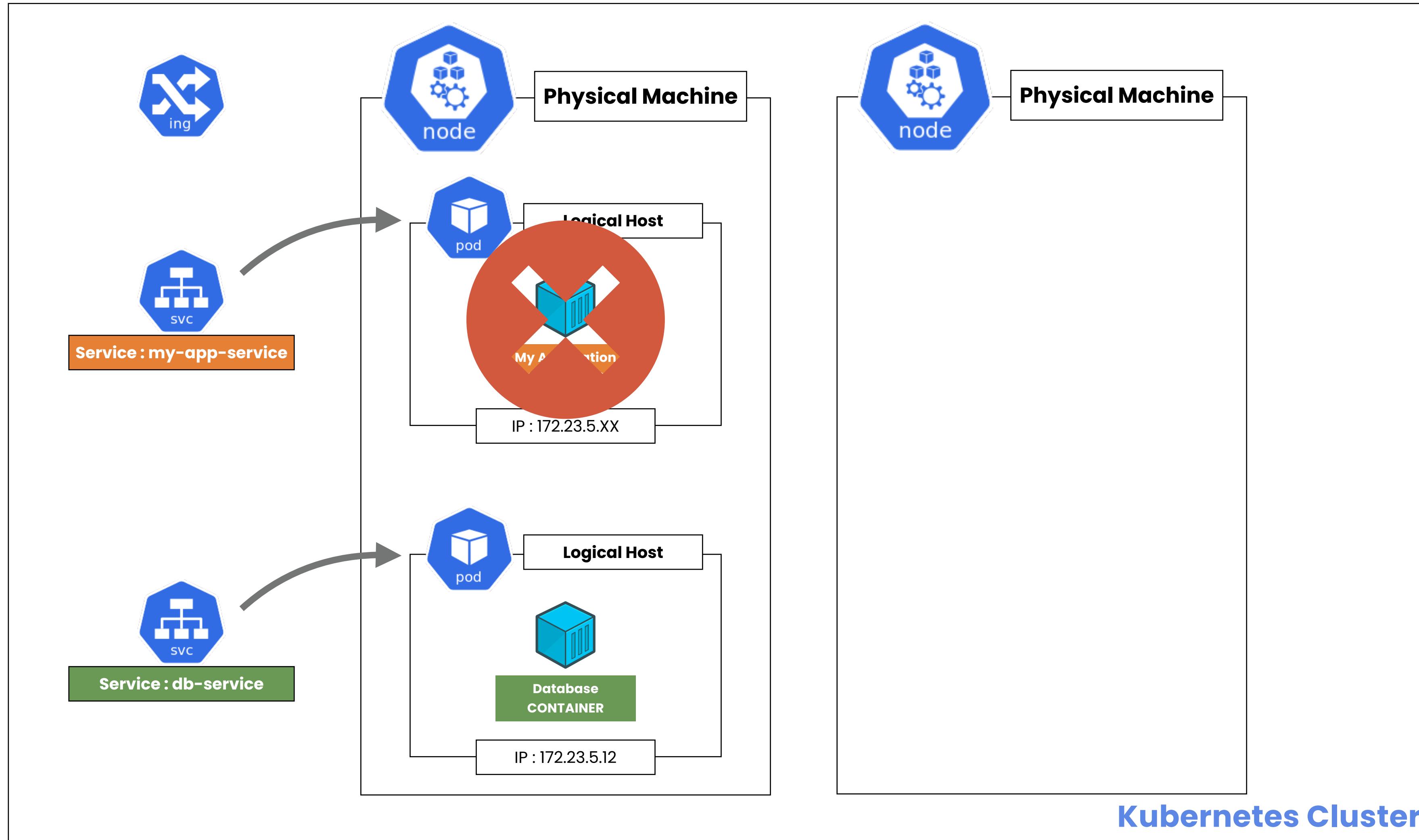


DEPLOYMENTS

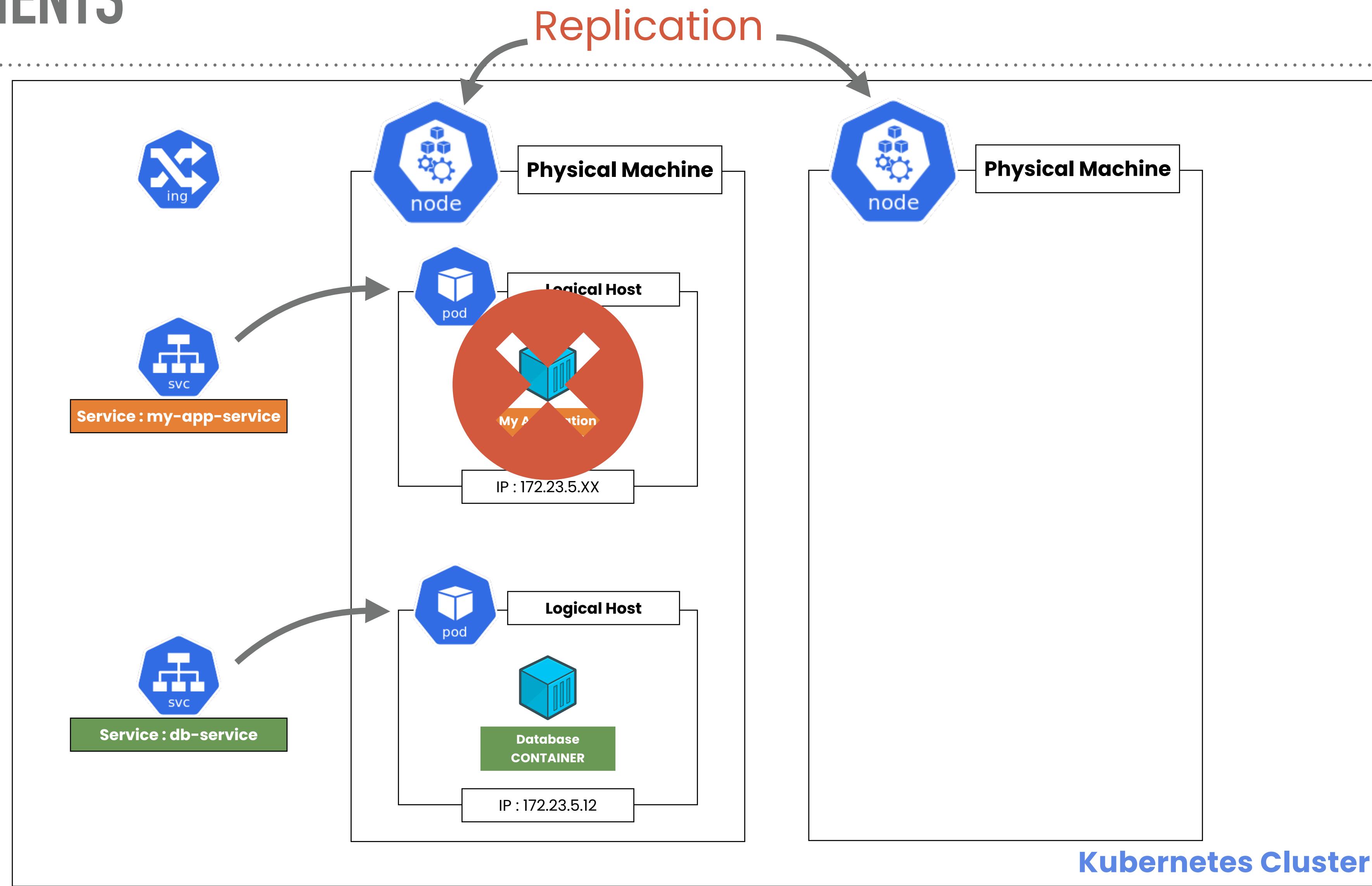


DEPLOYMENTS

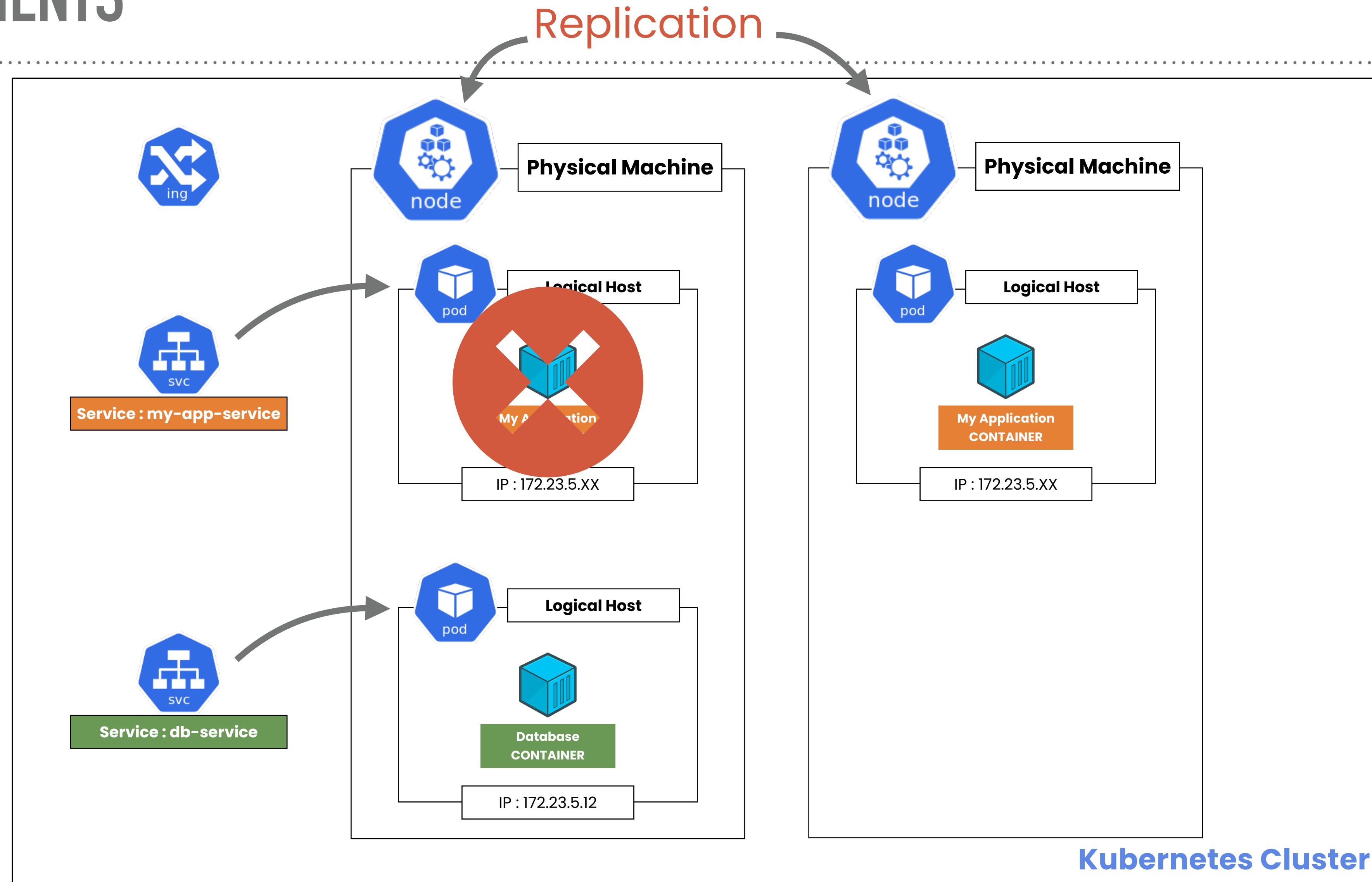
Replication



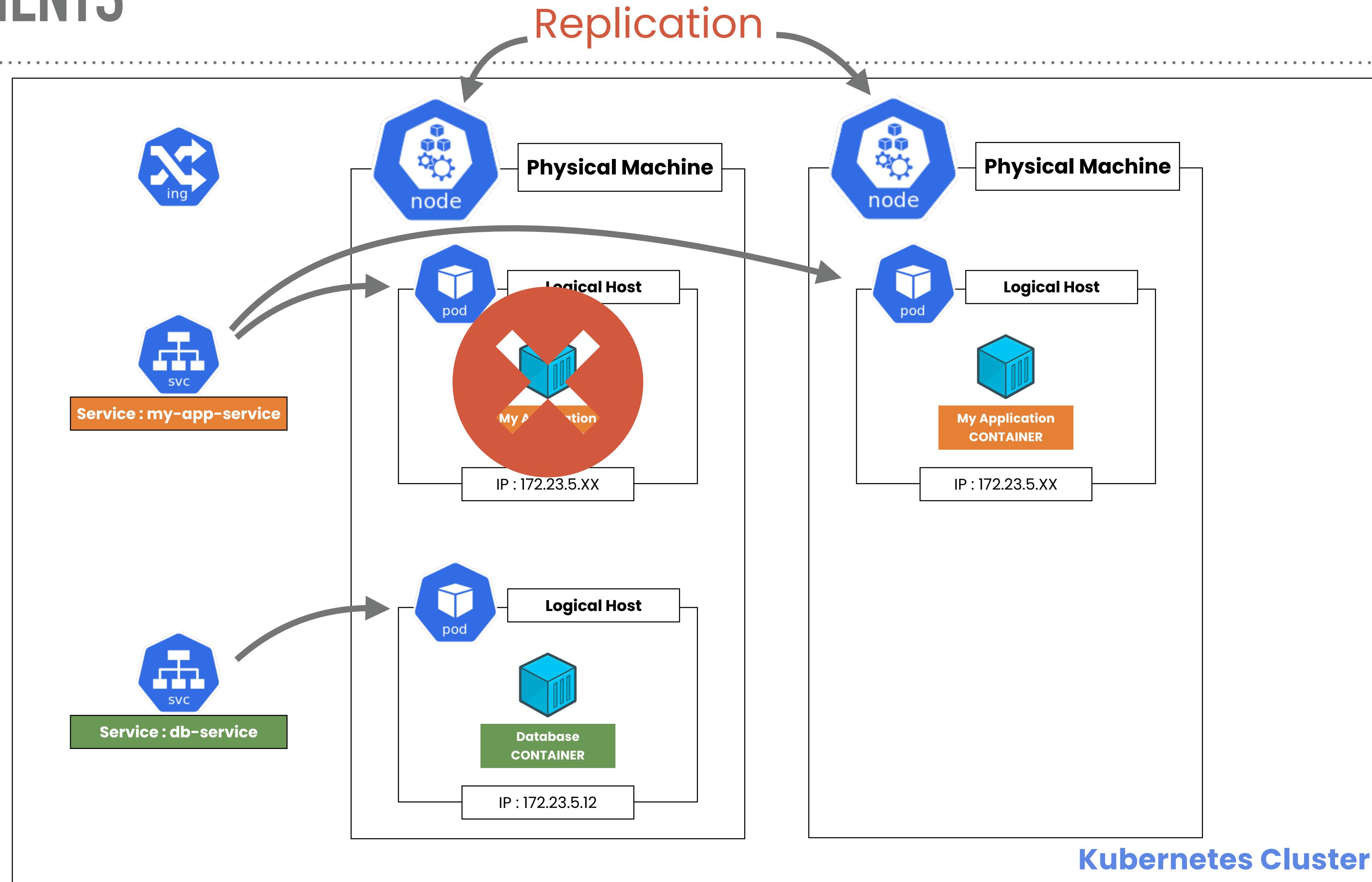
DEPLOYMENTS



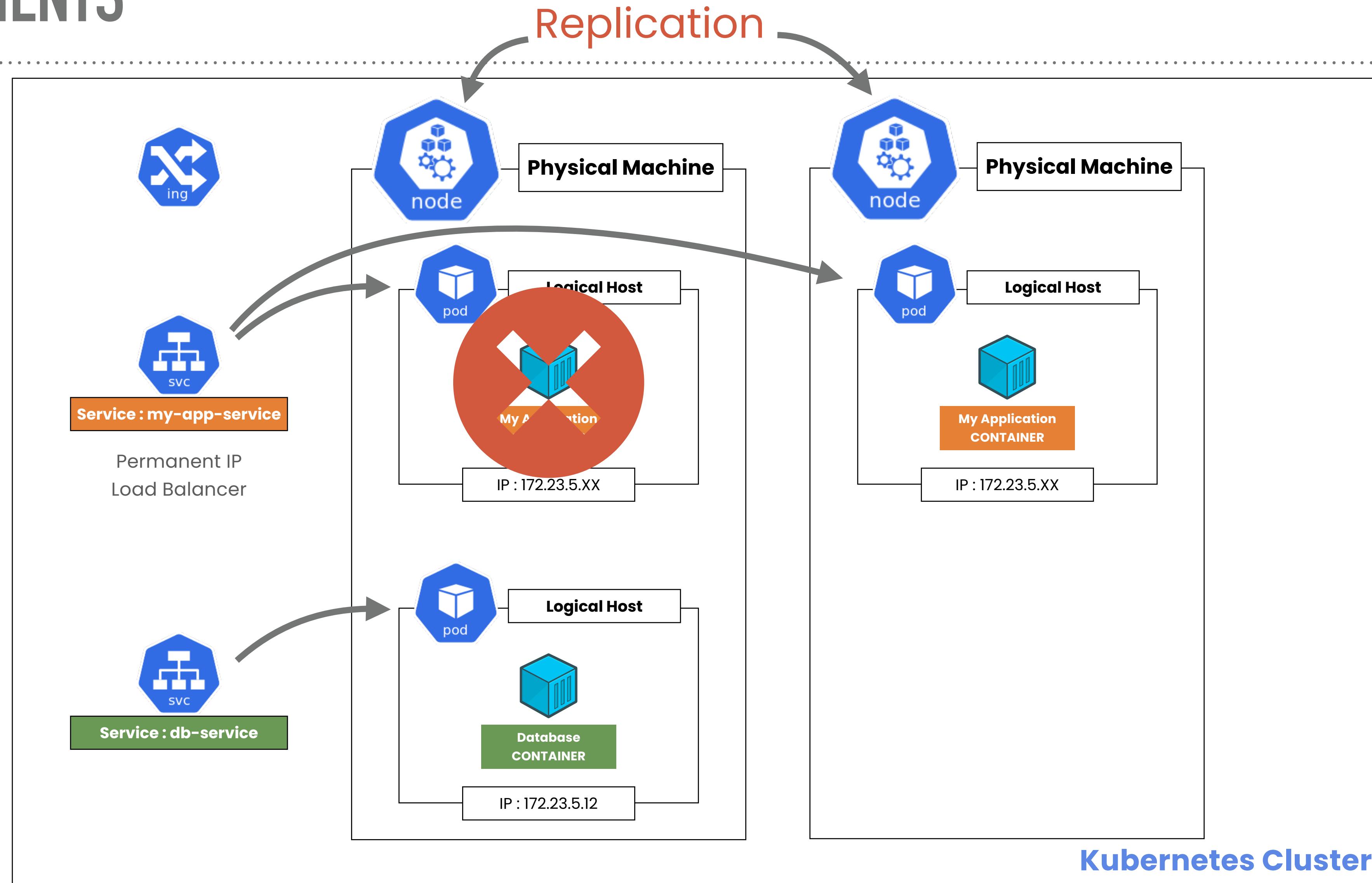
DEPLOYMENTS



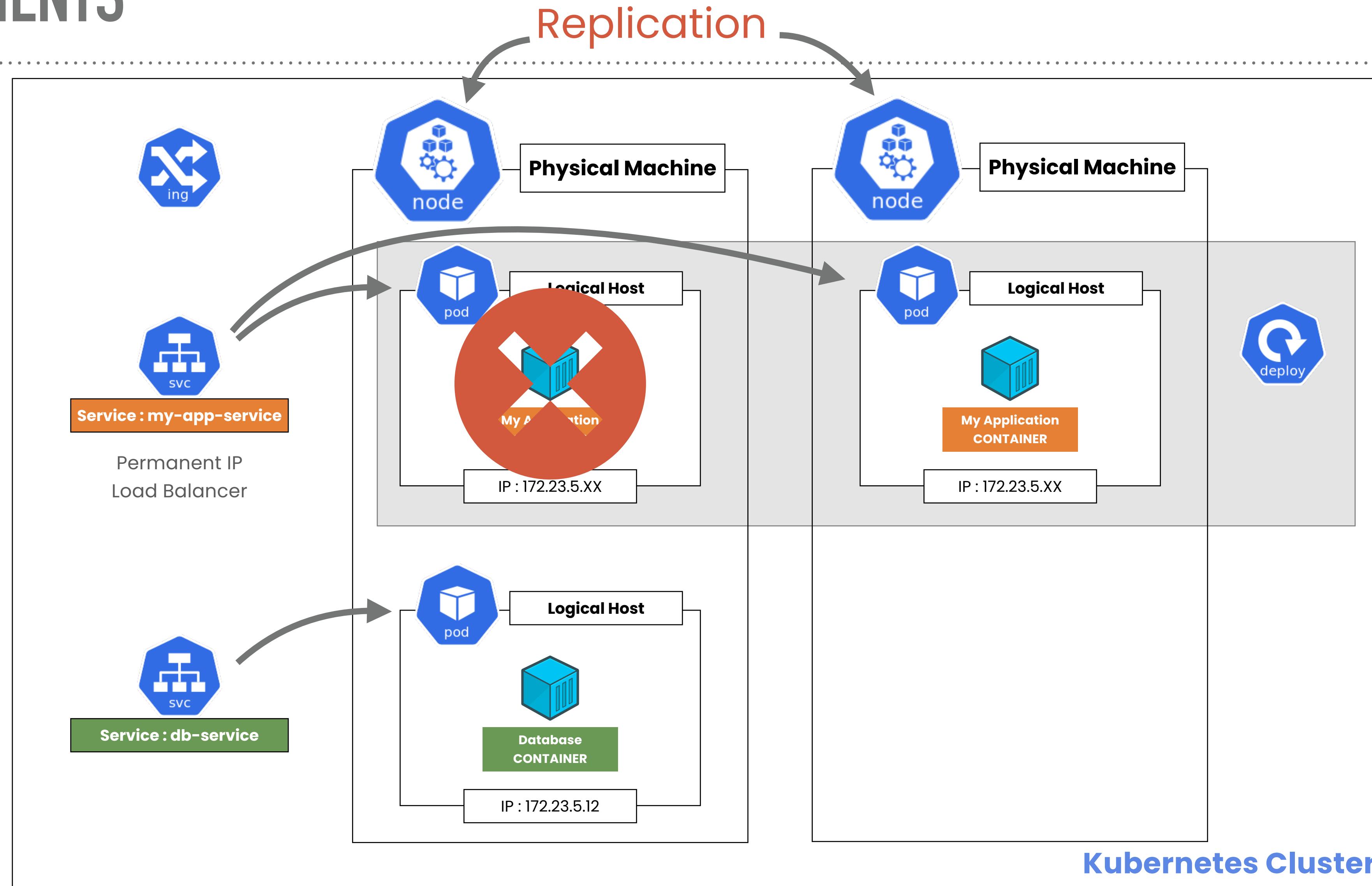
DEPLOYMENTS



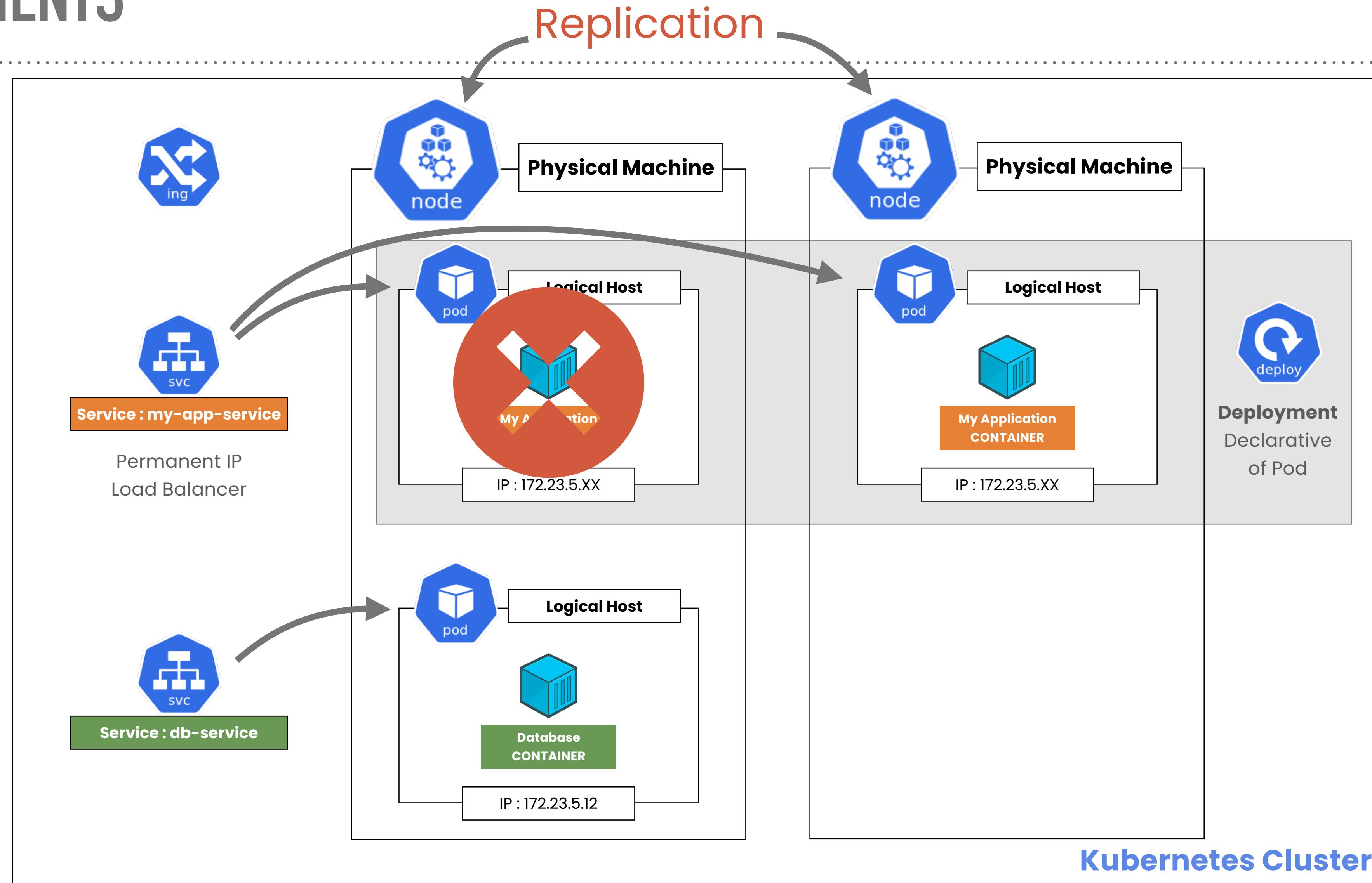
DEPLOYMENTS



DEPLOYMENTS



DEPLOYMENTS

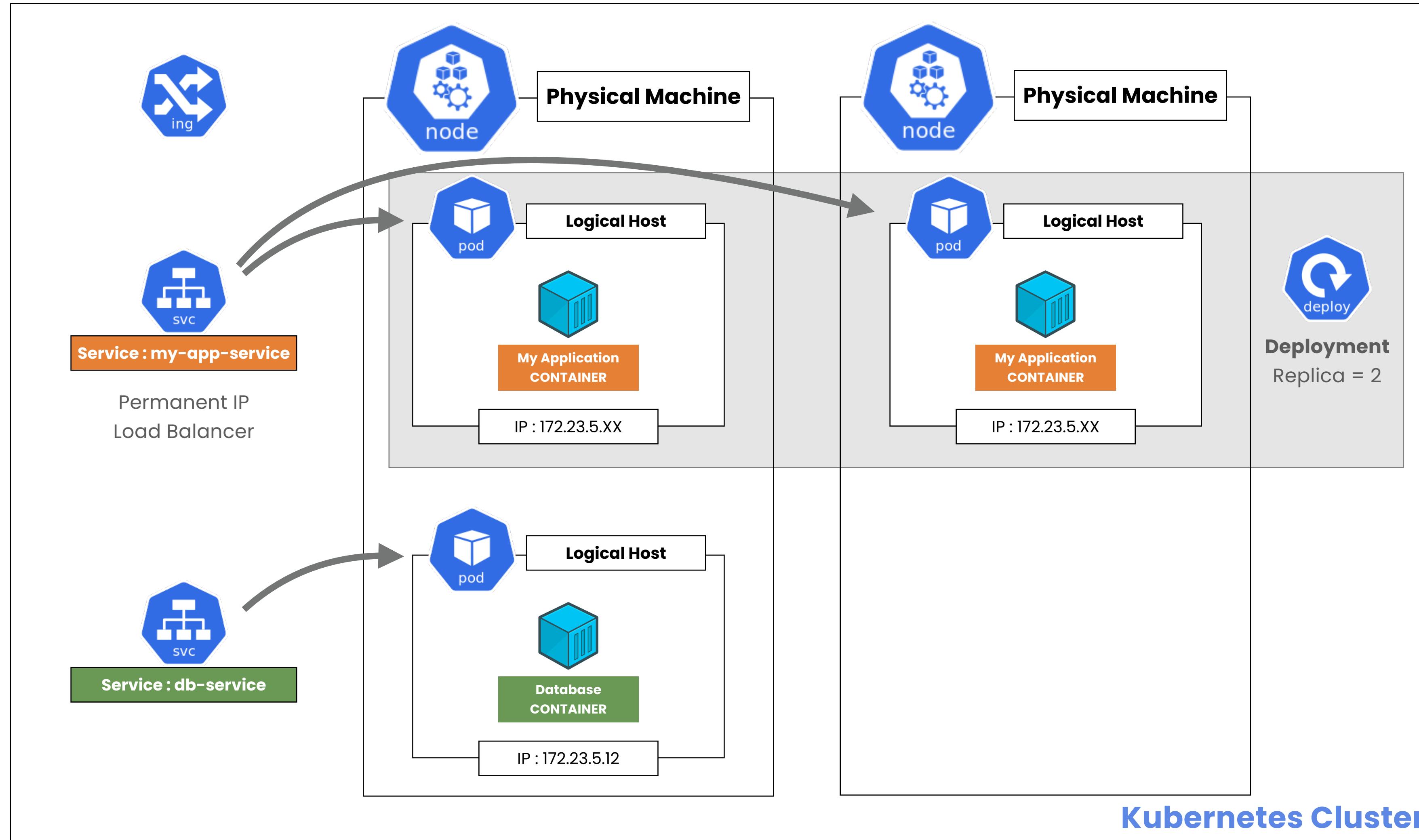


DEPLOYMENTS

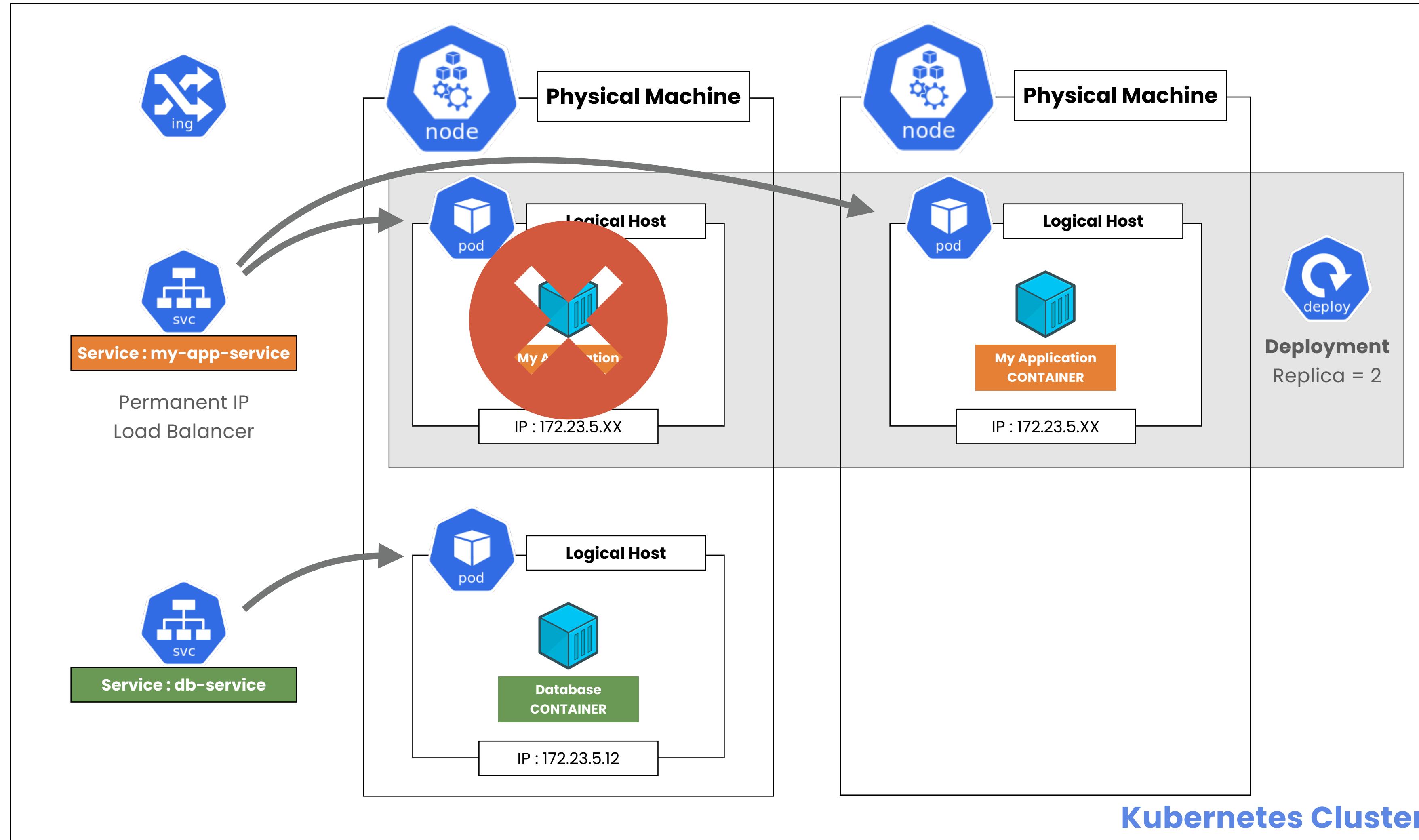
A *Deployment* provides declarative updates for Pods and ReplicaSets.

You describe a *desired state* in a Deployment, and the Deployment Controller changes the *actual state to the desired state* at a controlled rate. You can define Deployments to create new ReplicaSets, or to remove existing Deployments and adopt all their resources with new Deployments.

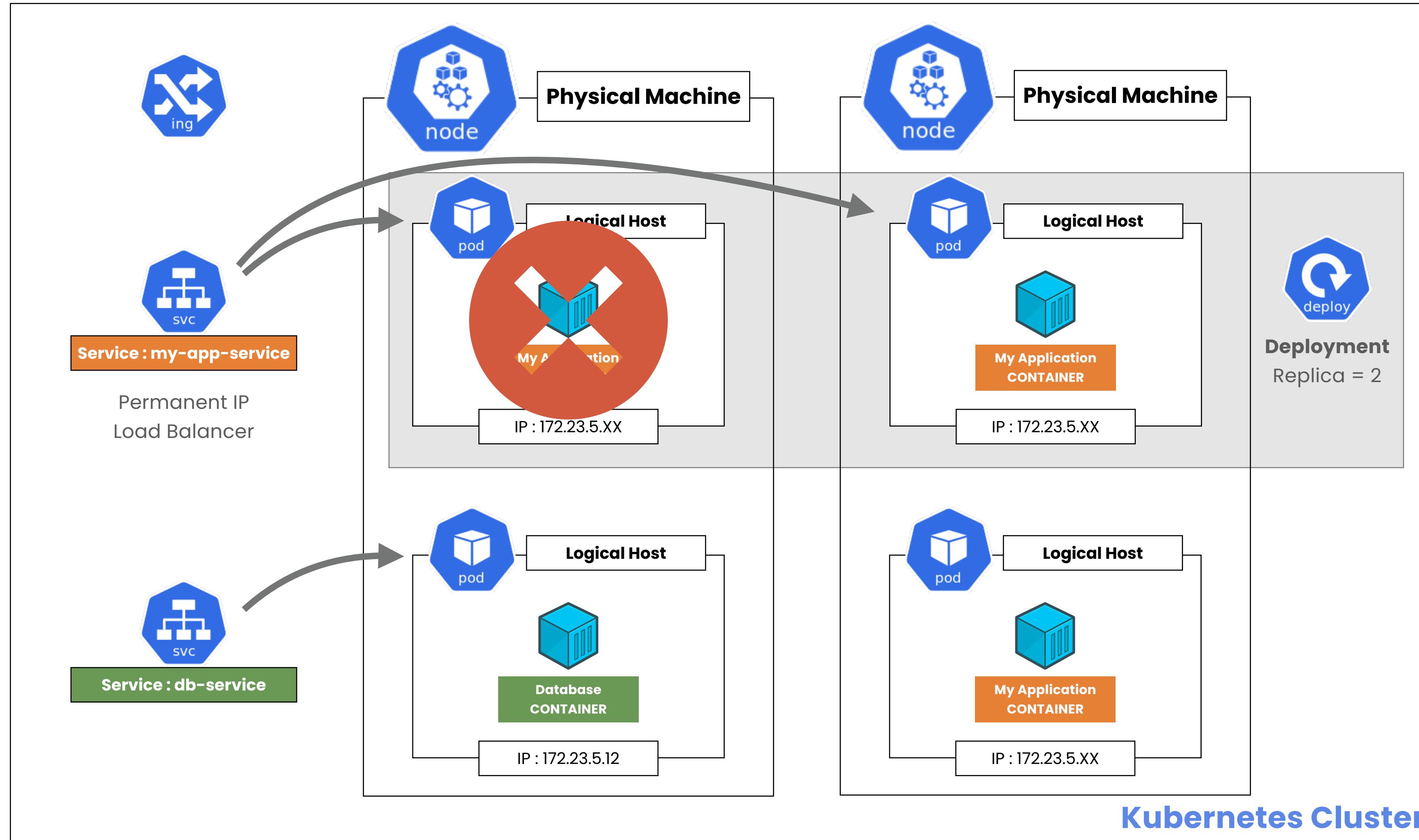
DEPLOYMENTS FOR STATELESS APPLICATION



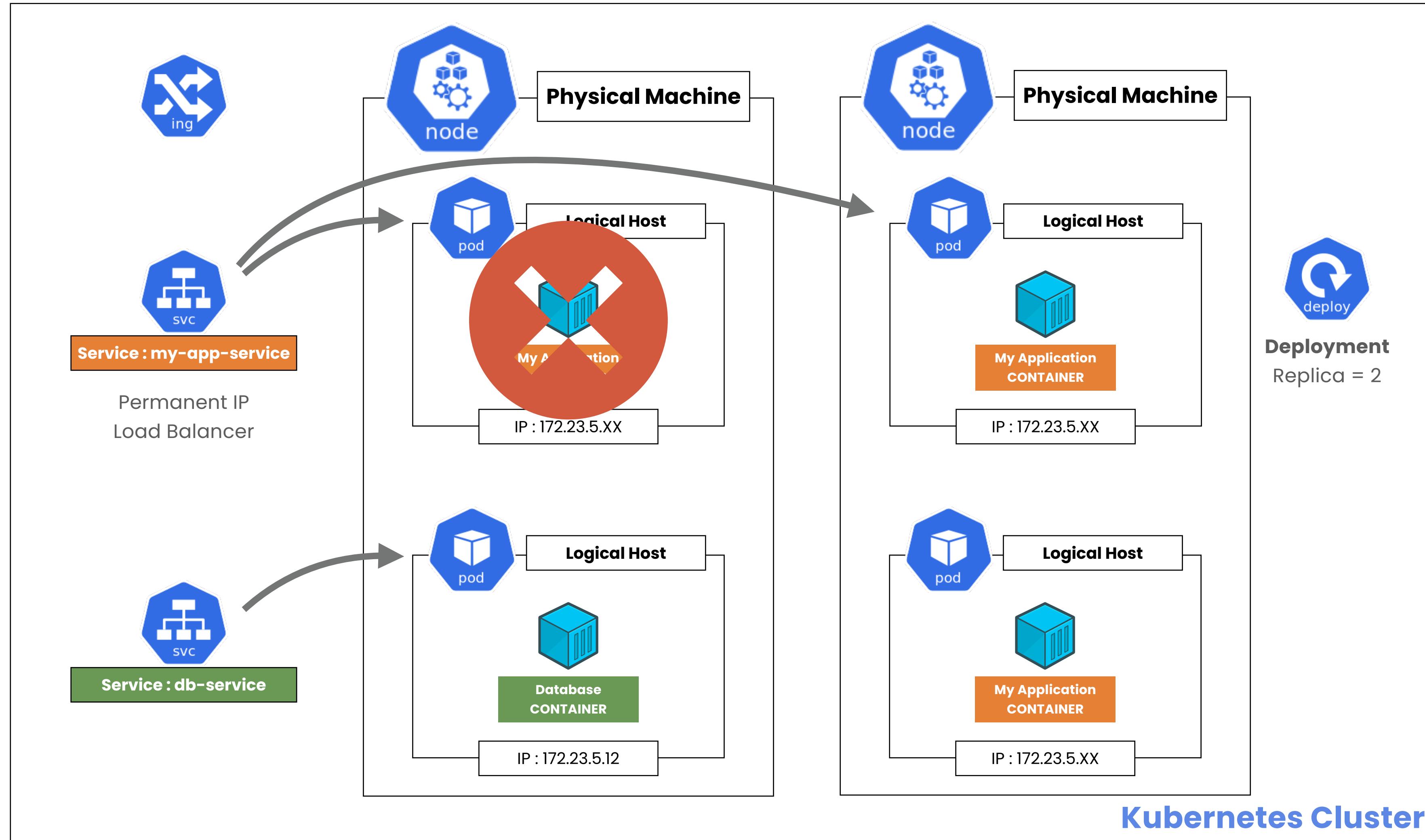
DEPLOYMENTS FOR STATELESS APPLICATION



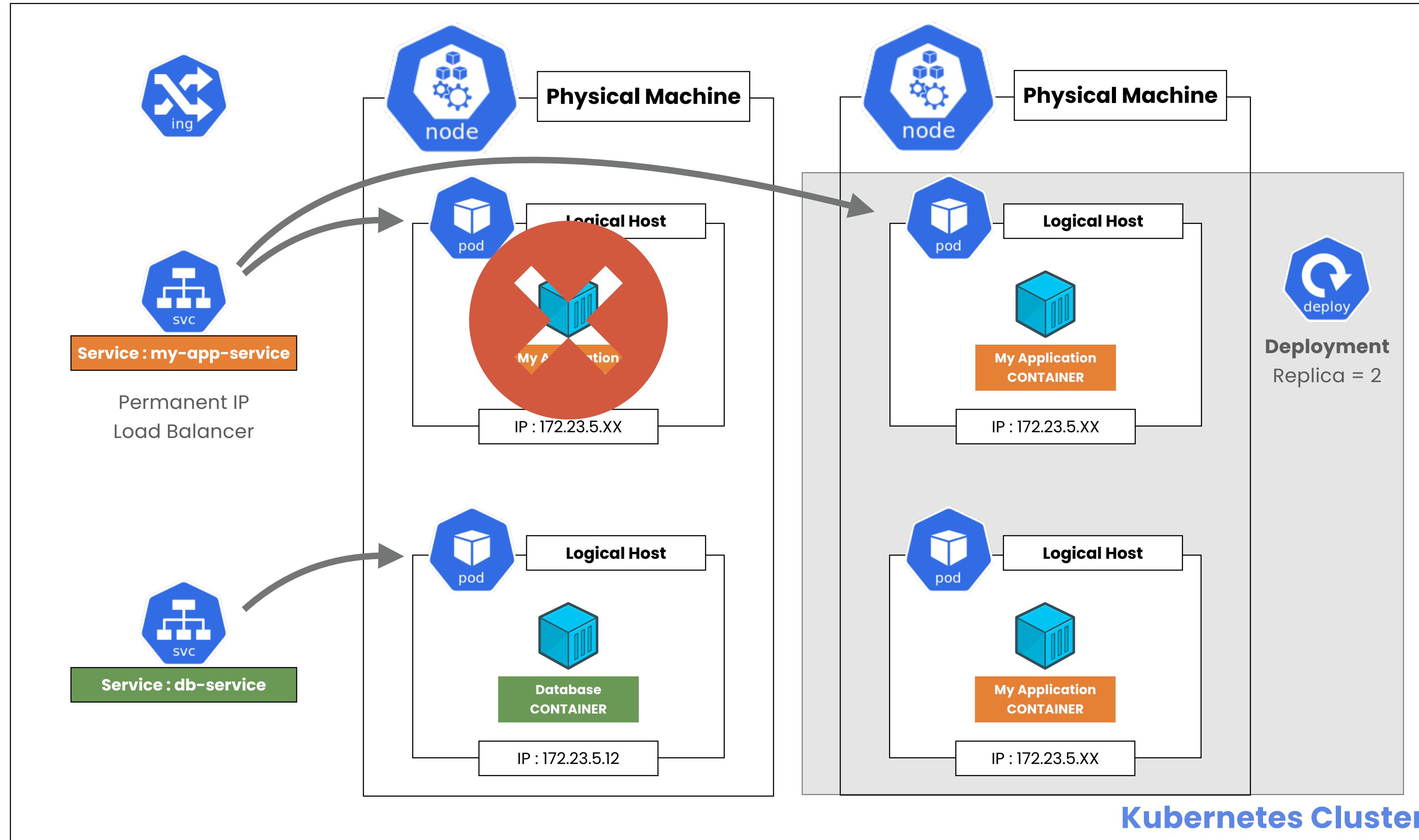
DEPLOYMENTS FOR STATELESS APPLICATION



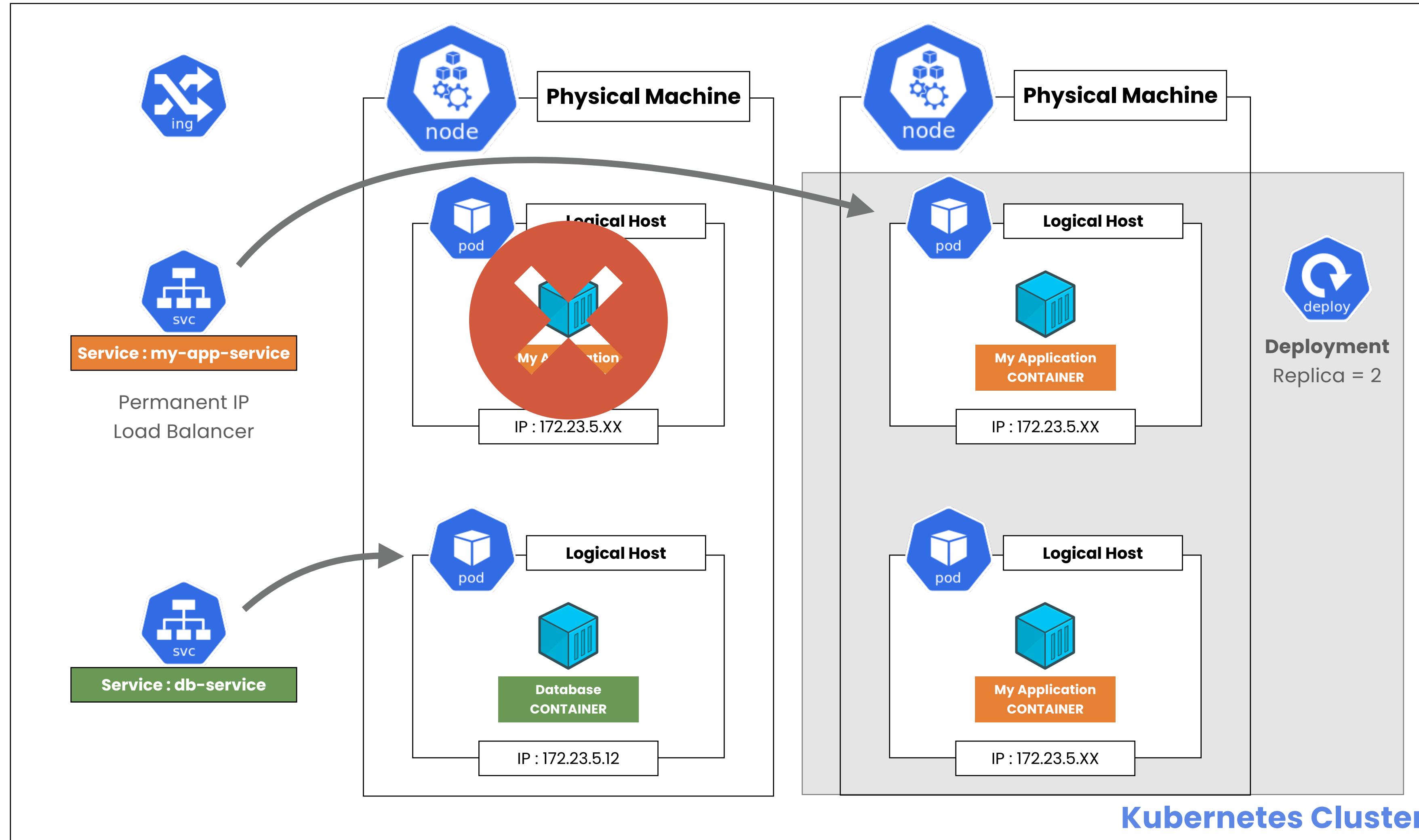
DEPLOYMENTS FOR STATELESS APPLICATION



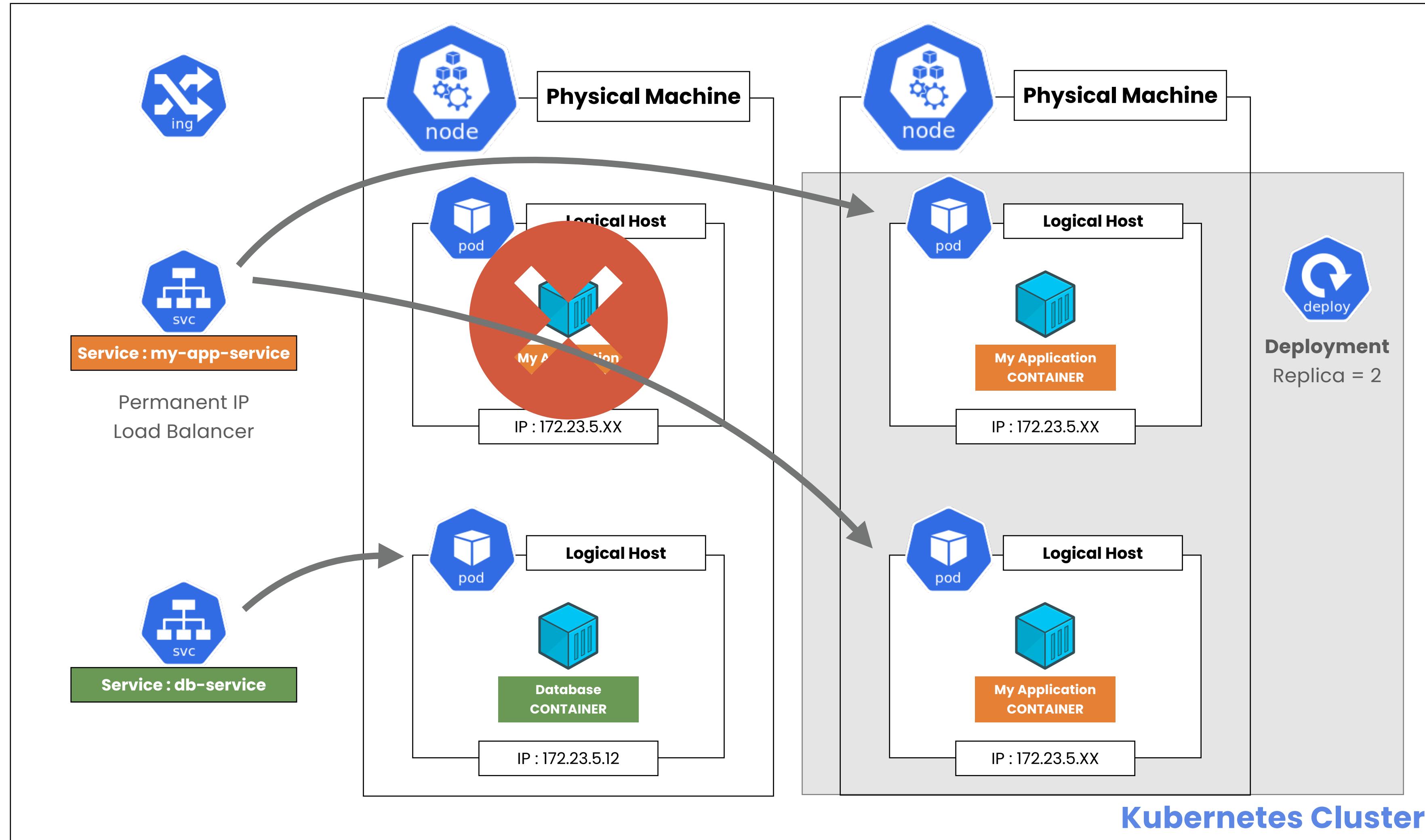
DEPLOYMENTS FOR STATELESS APPLICATION



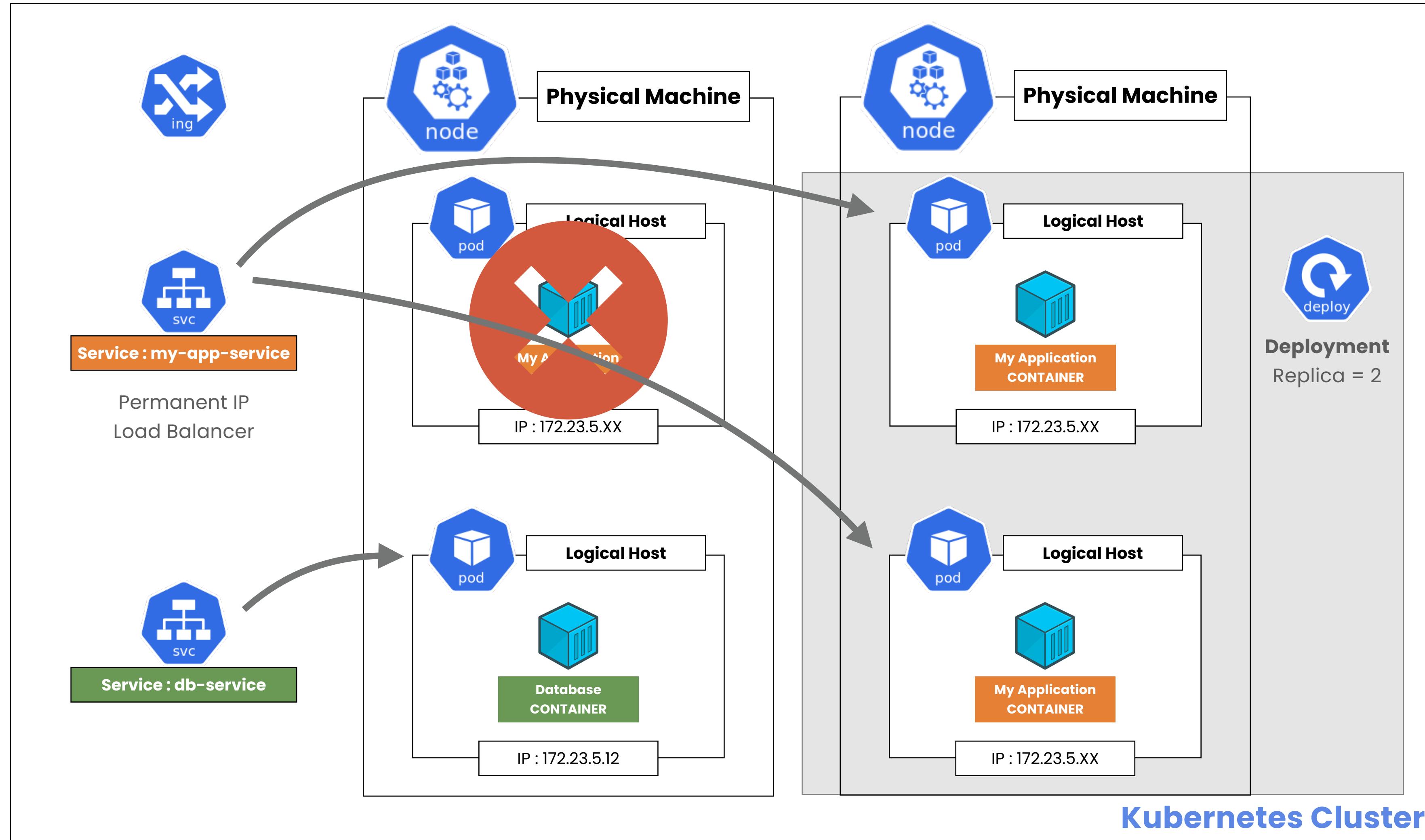
DEPLOYMENTS FOR STATELESS APPLICATION



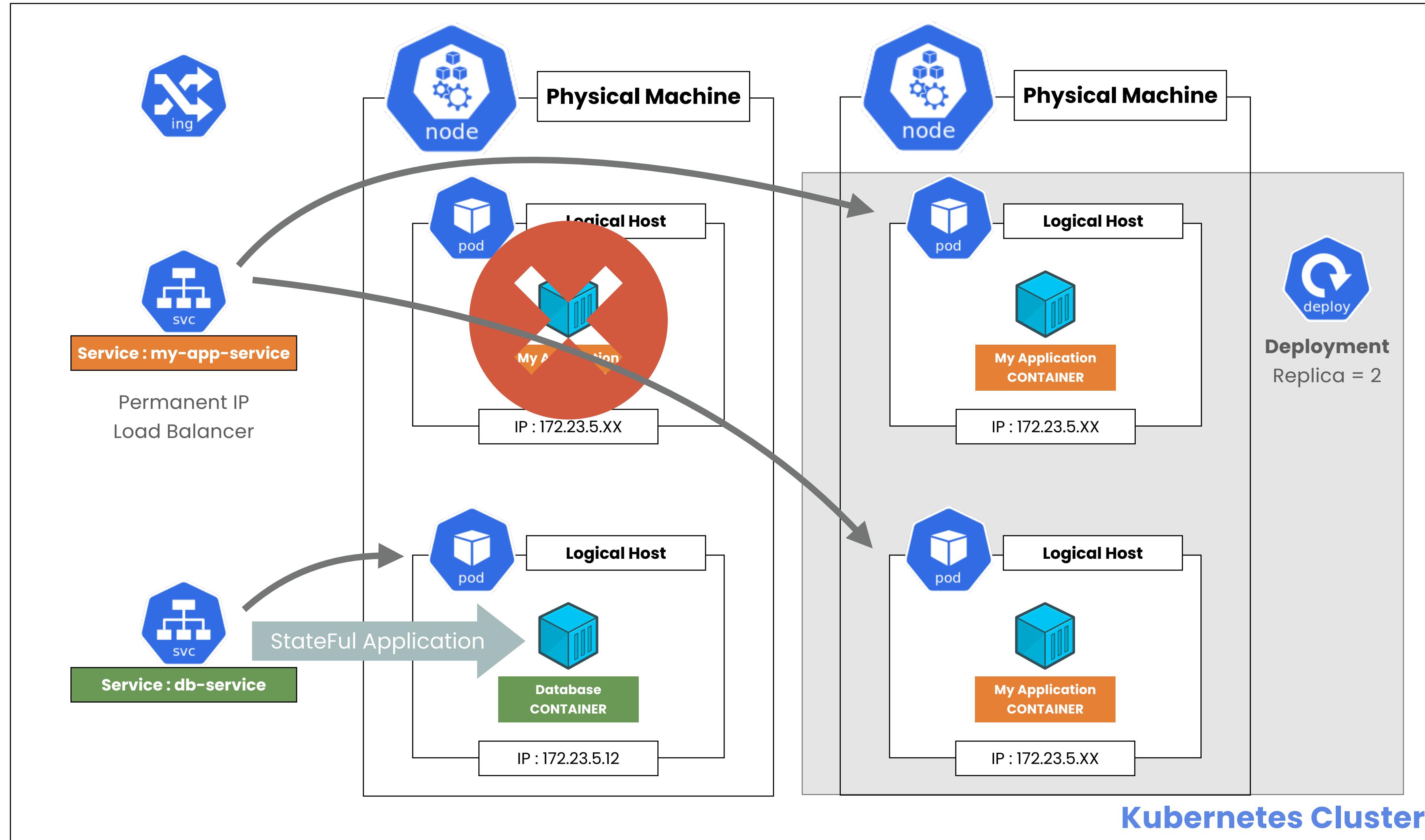
DEPLOYMENTS FOR STATELESS APPLICATION



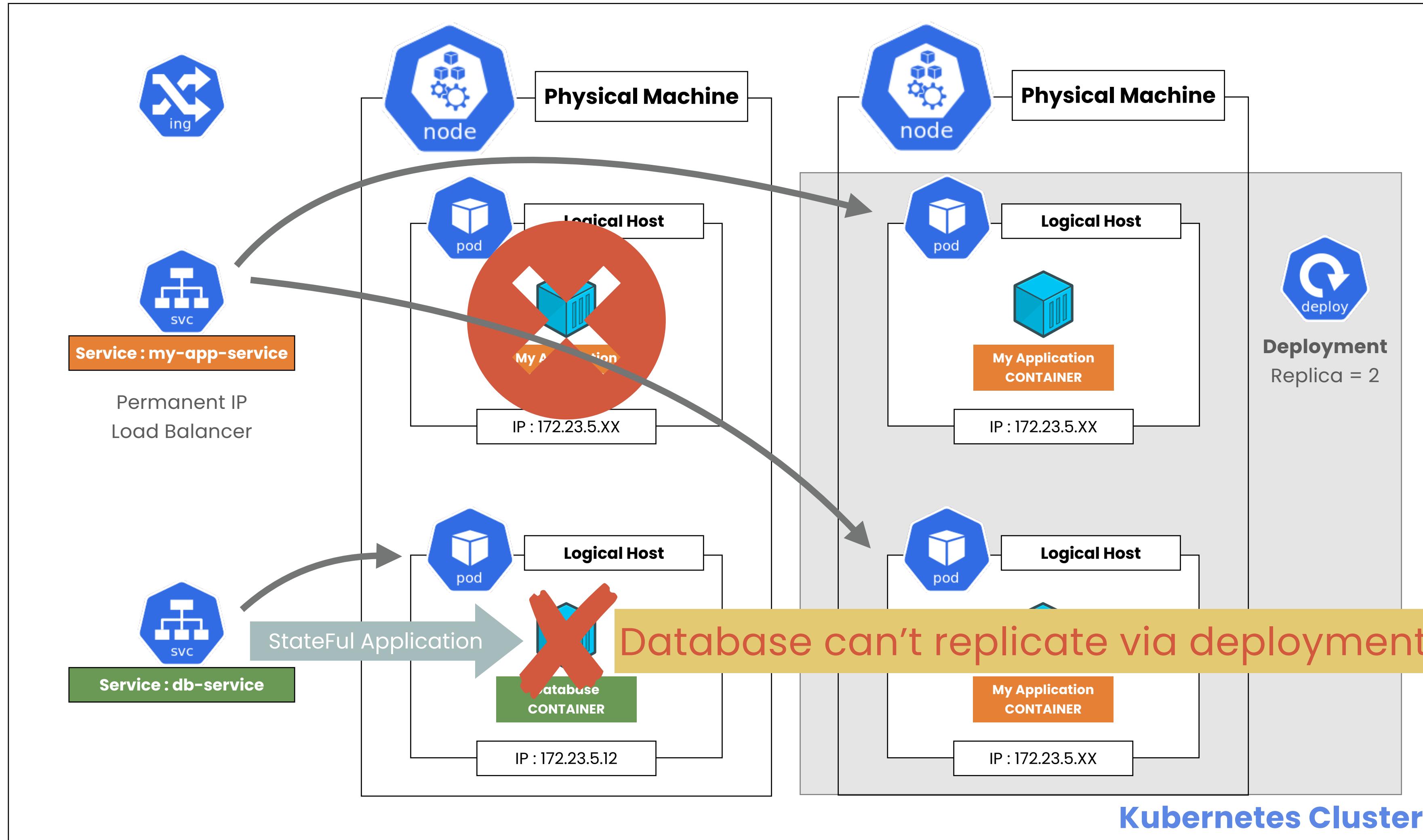
STATEFUL APPLICATION



STATEFUL APPLICATION



STATEFUL APPLICATION



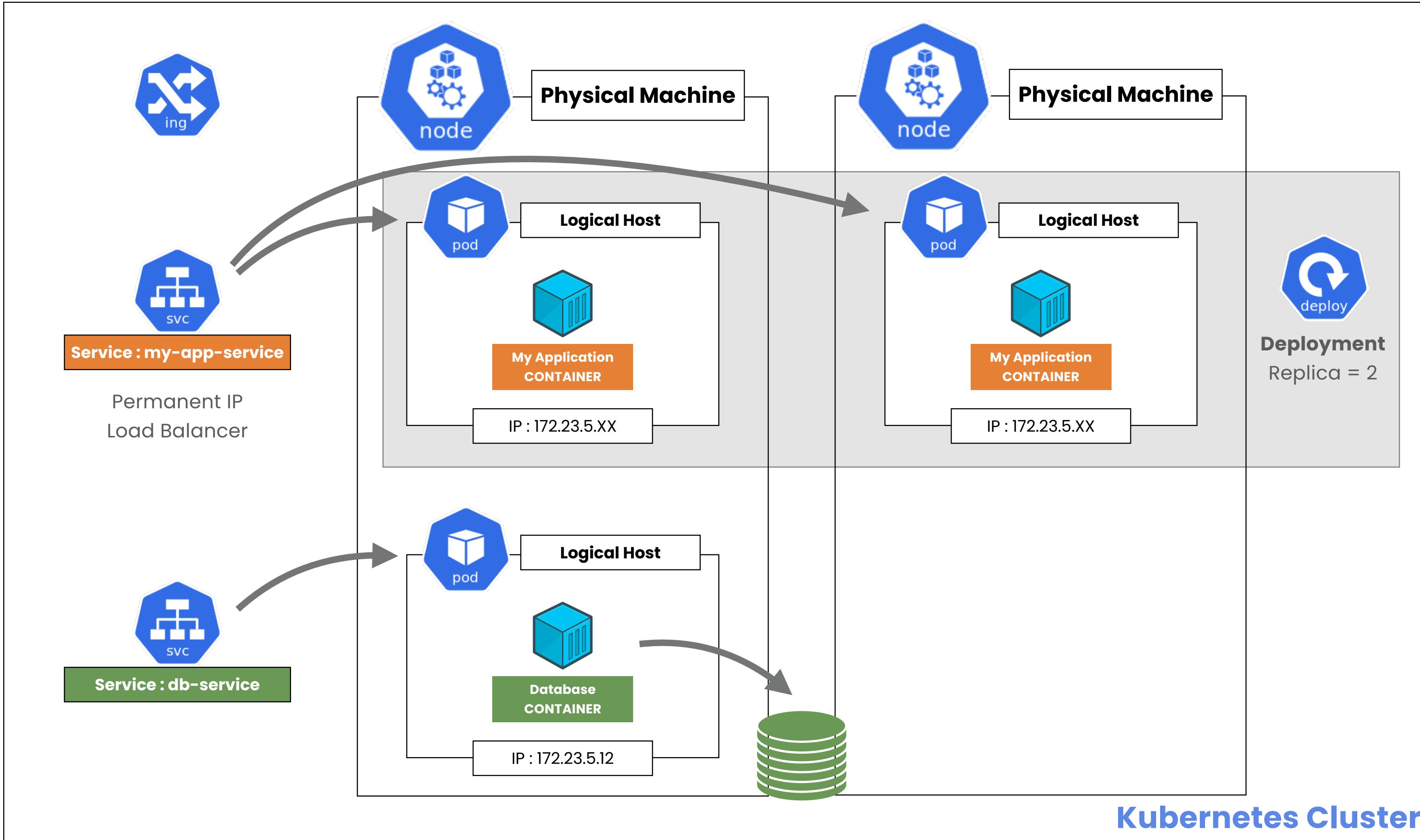
STATEFULSETS

StatefulSet is the workload API object used to **manage stateful applications**.

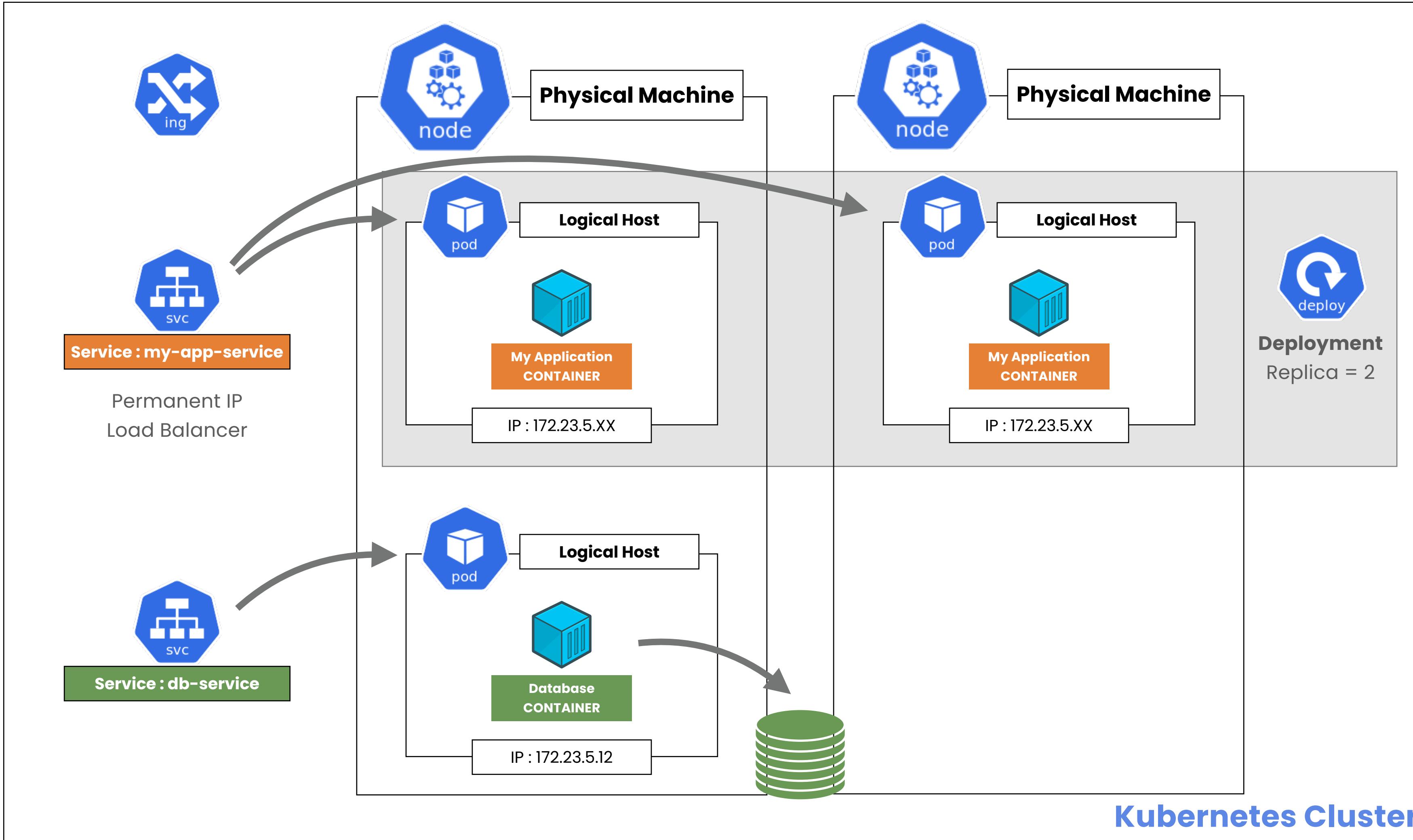
Manages the deployment and scaling of a set of Pods, *and provides guarantees about the ordering and uniqueness* of these Pods.

StatefulSet maintains a **sticky identity** for each of their Pods. These **pods are created from the same spec**, but are not interchangeable: each has a persistent identifier that it maintains across any rescheduling.

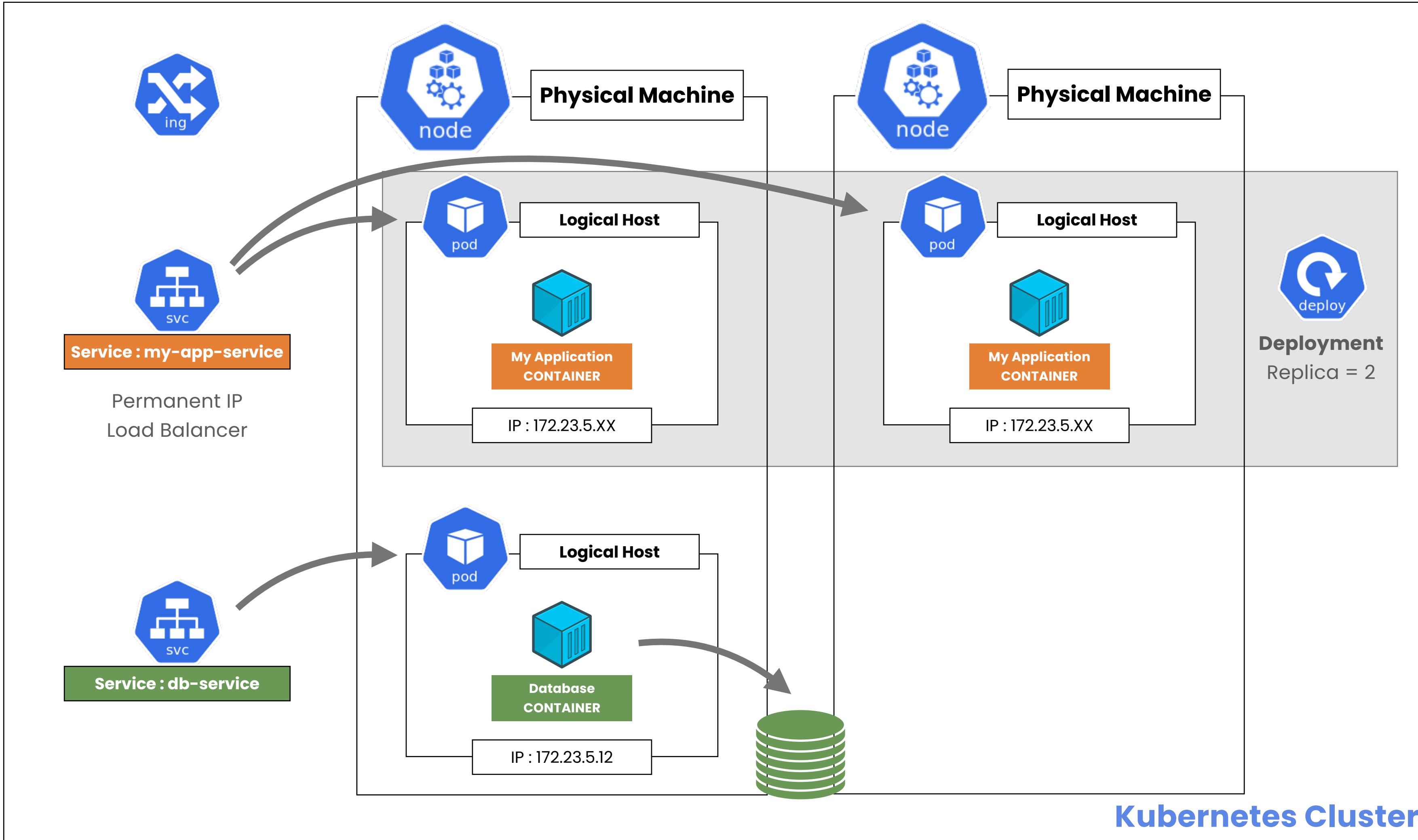
STATEFUL-SET



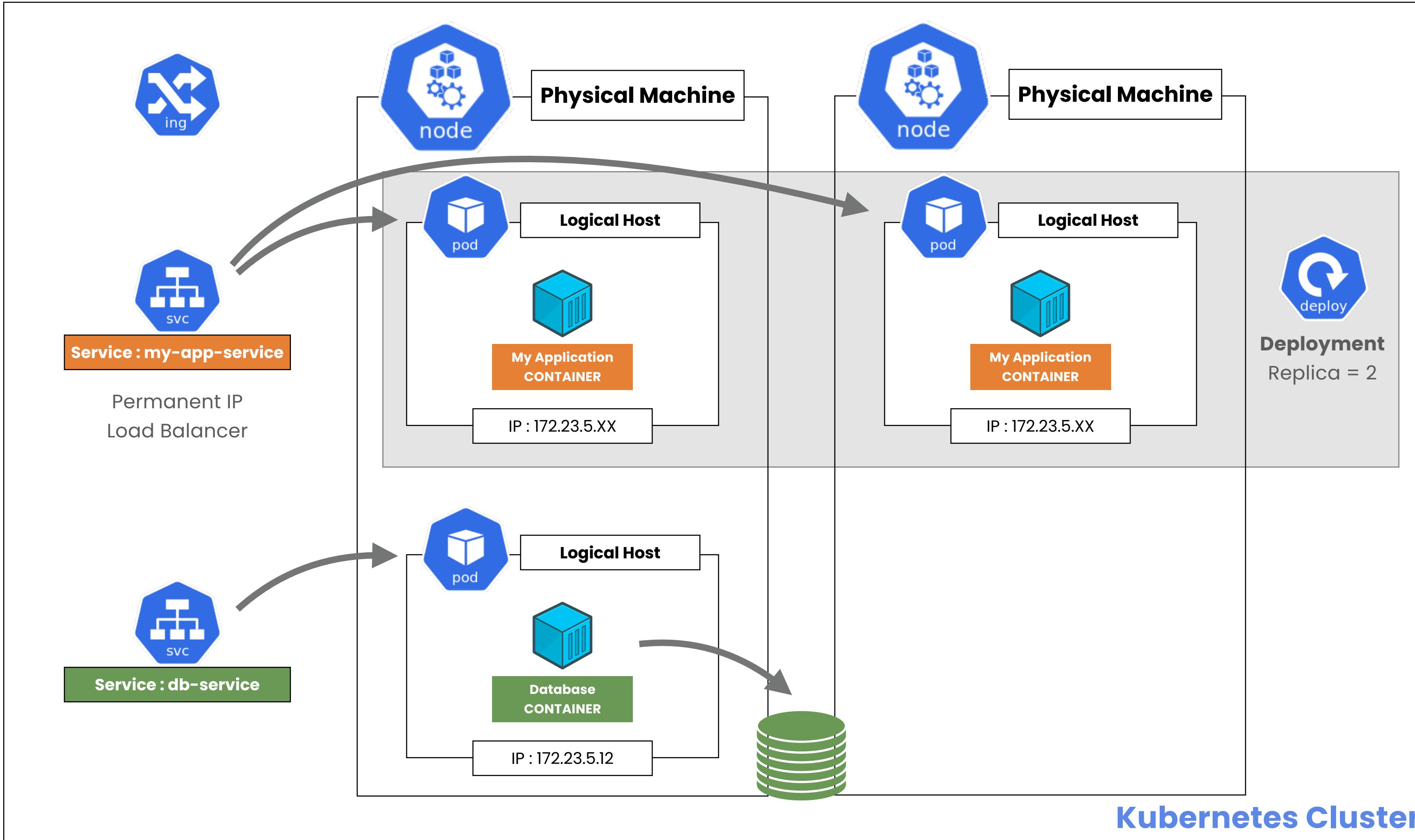
STATEFUL-SET



STATEFUL-SET



STATEFUL-SET

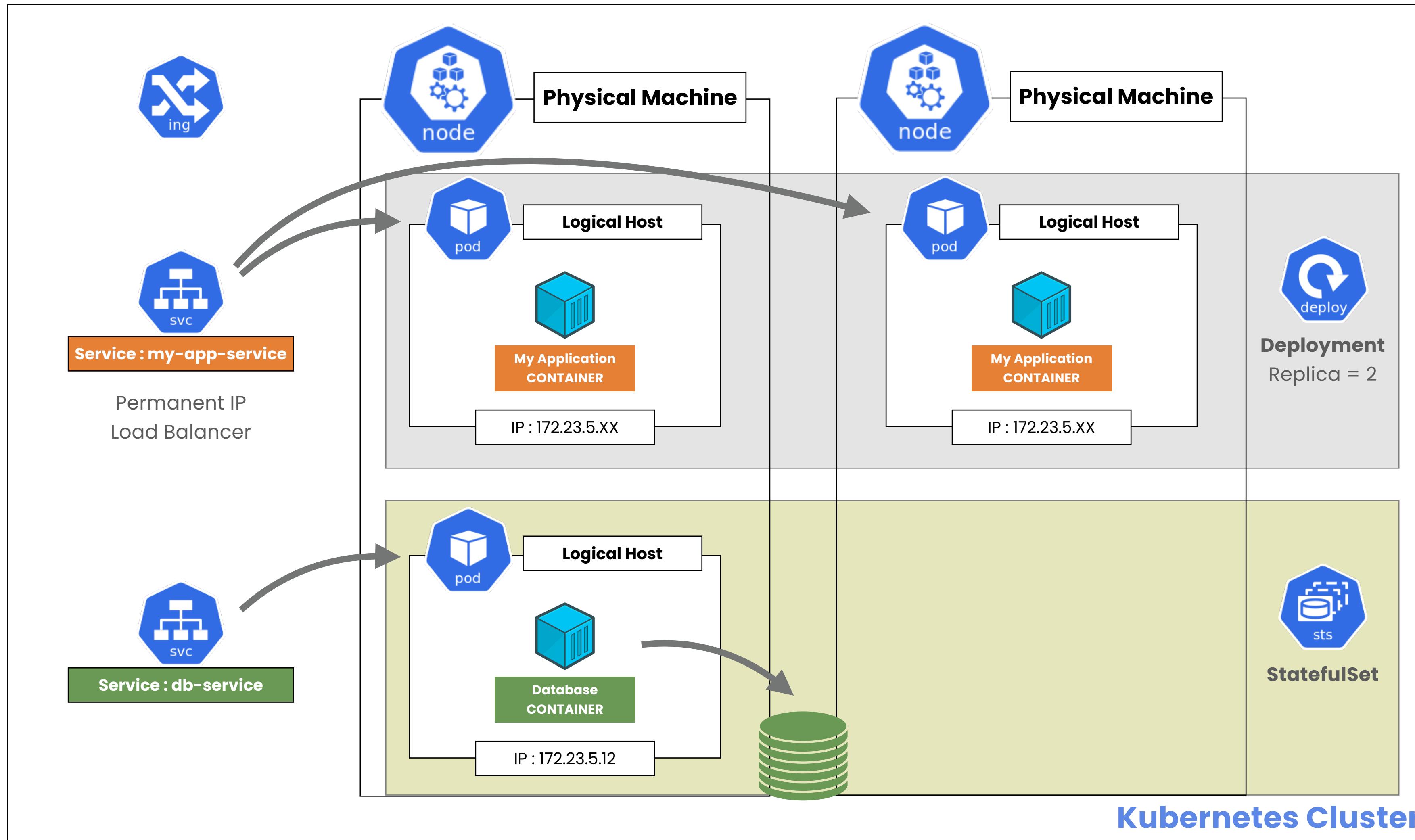


MySQL™

mongoDB®

elasticsearch

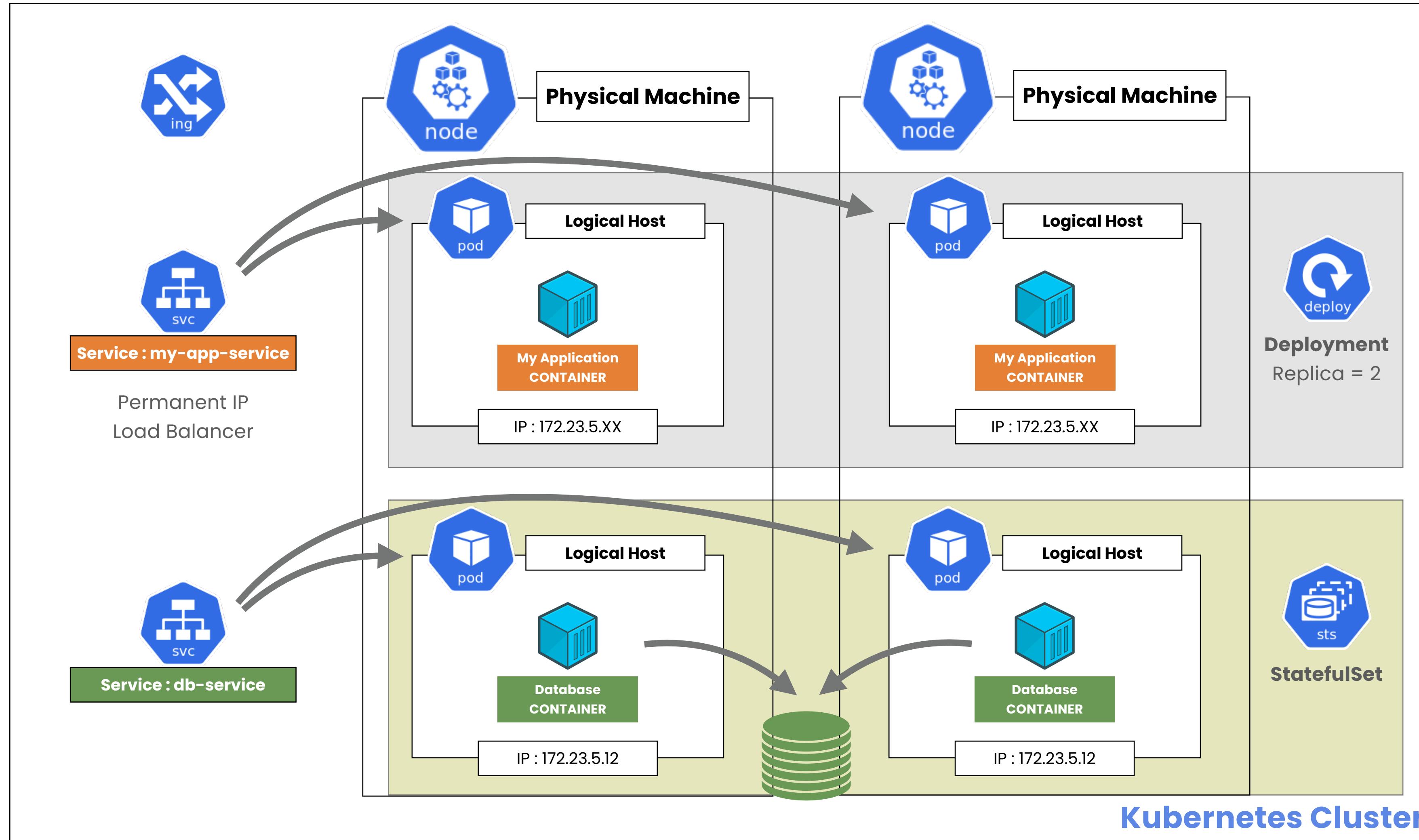
STATEFUL-SET



Kubernetes Cluster

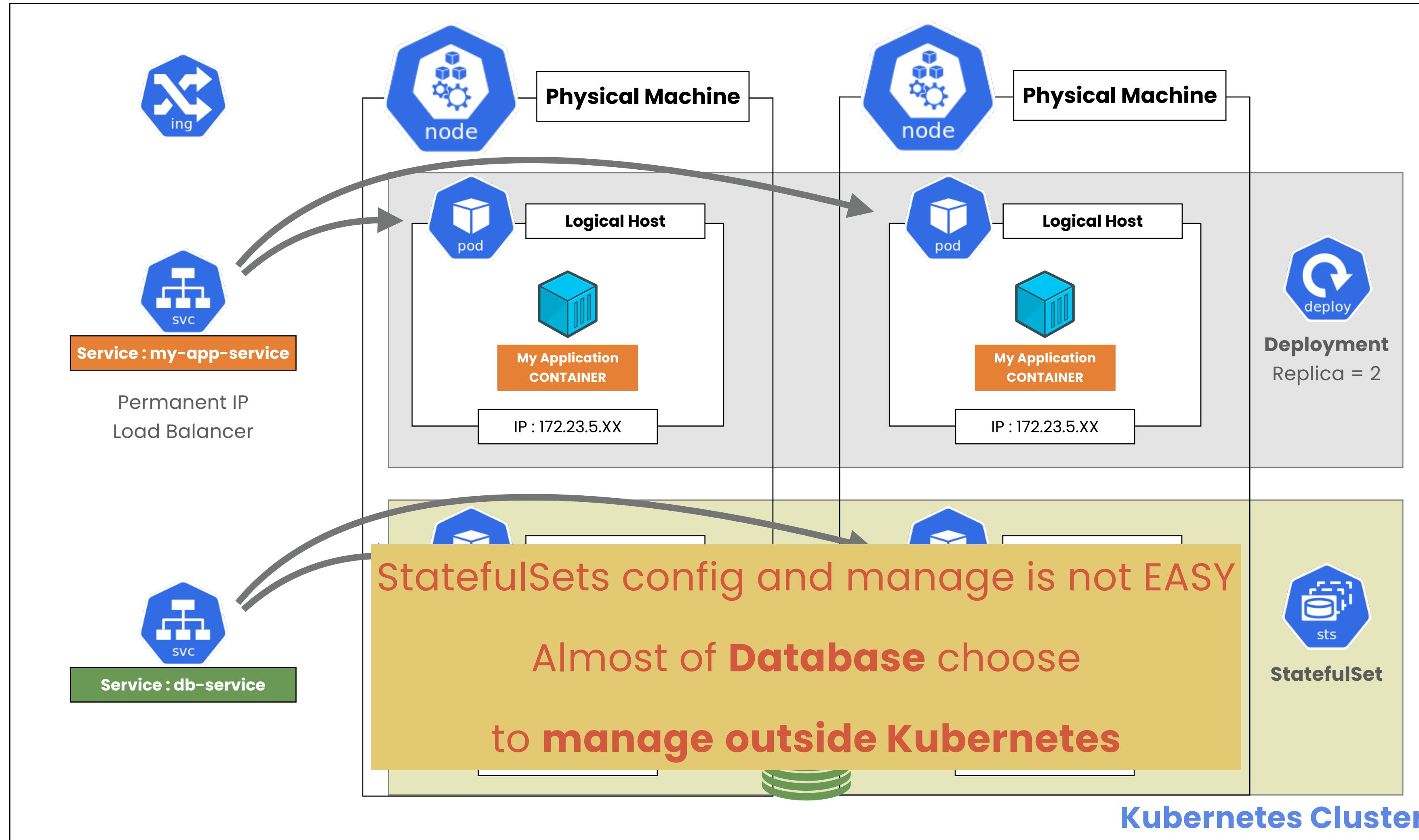


STATEFUL-SET

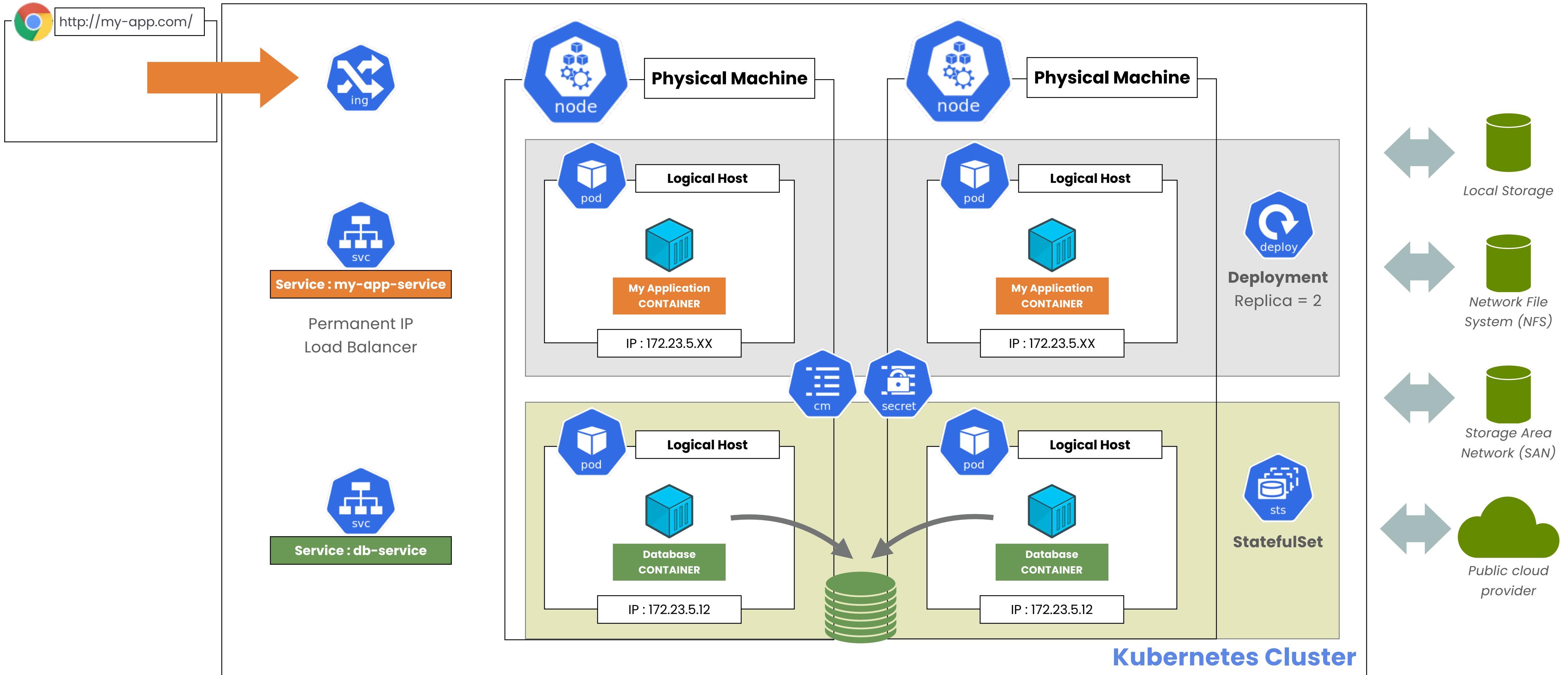


MySQL
mongoDB®
elasticsearch

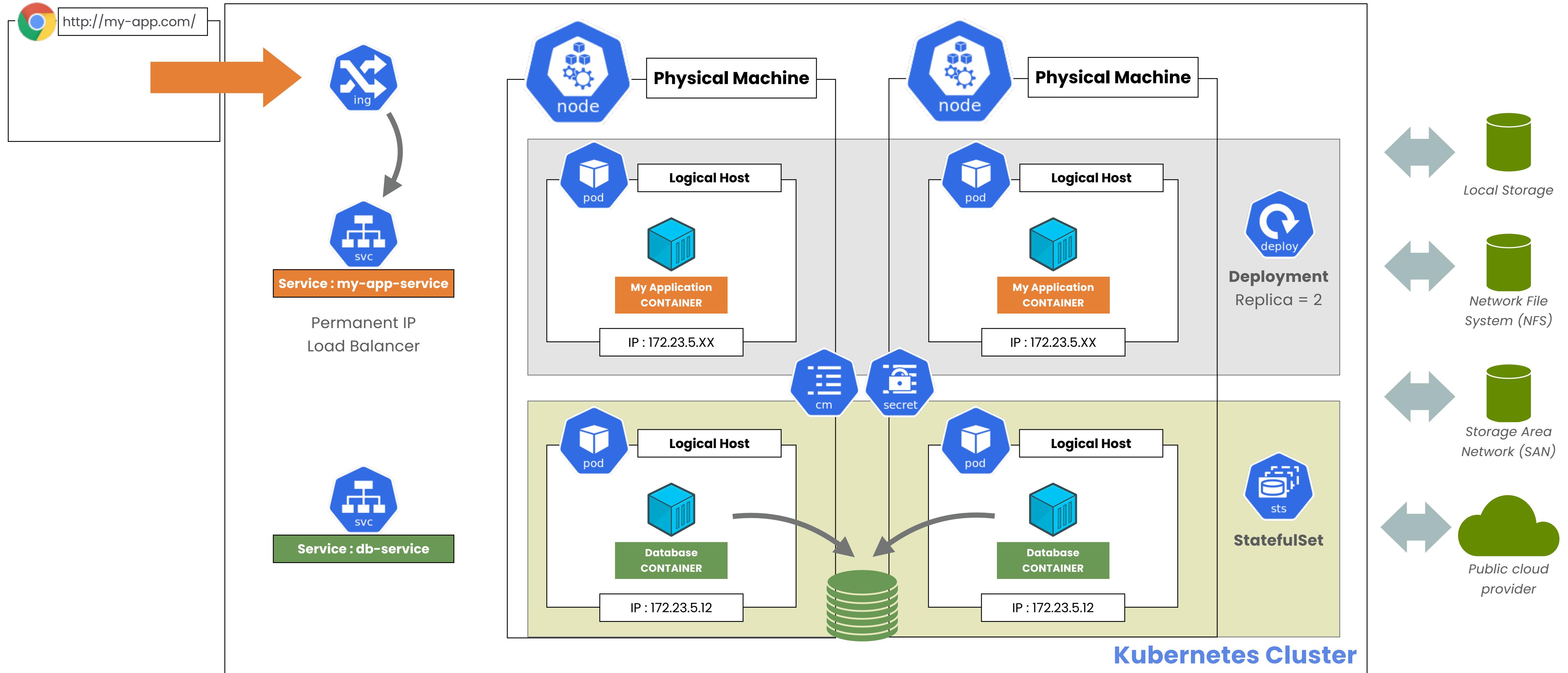
STATEFUL-SET



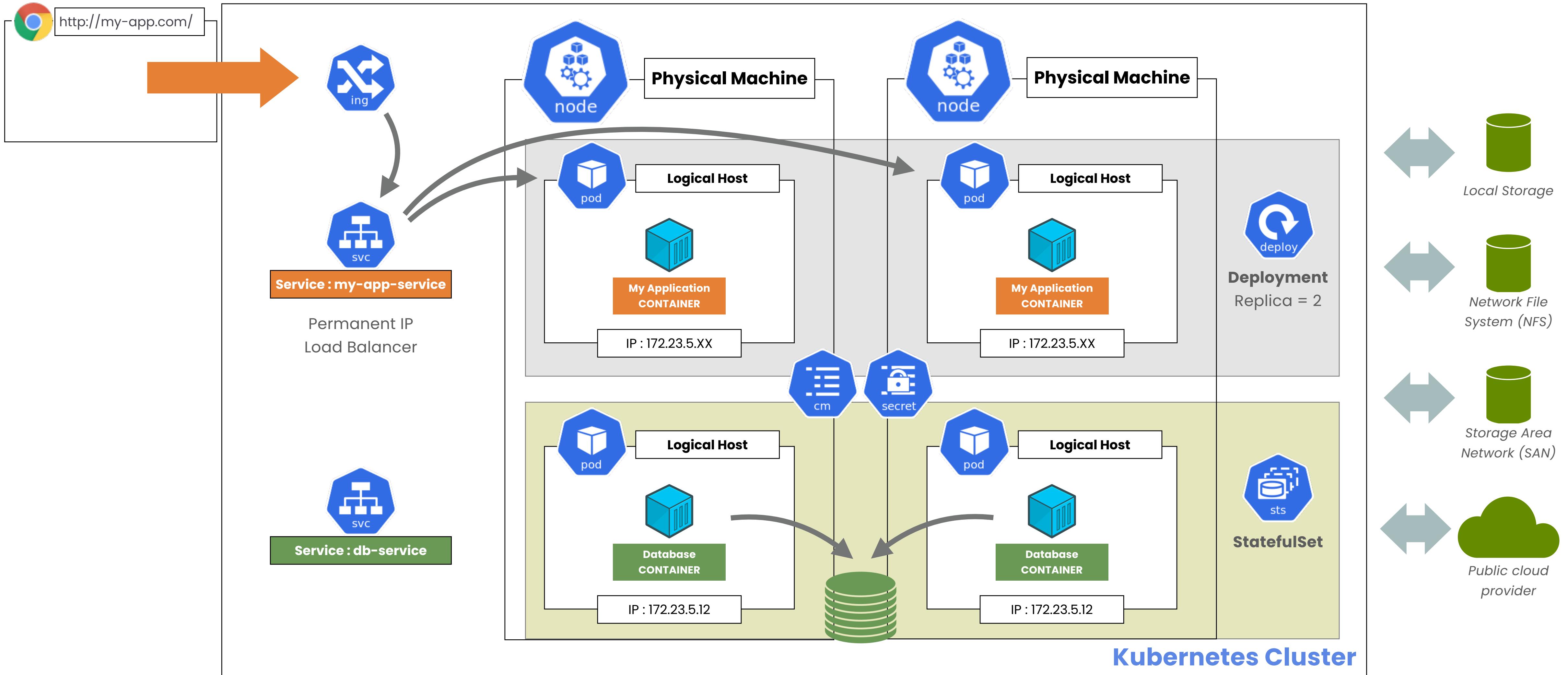
HIGH AVAILABLE ARCHITECTURE



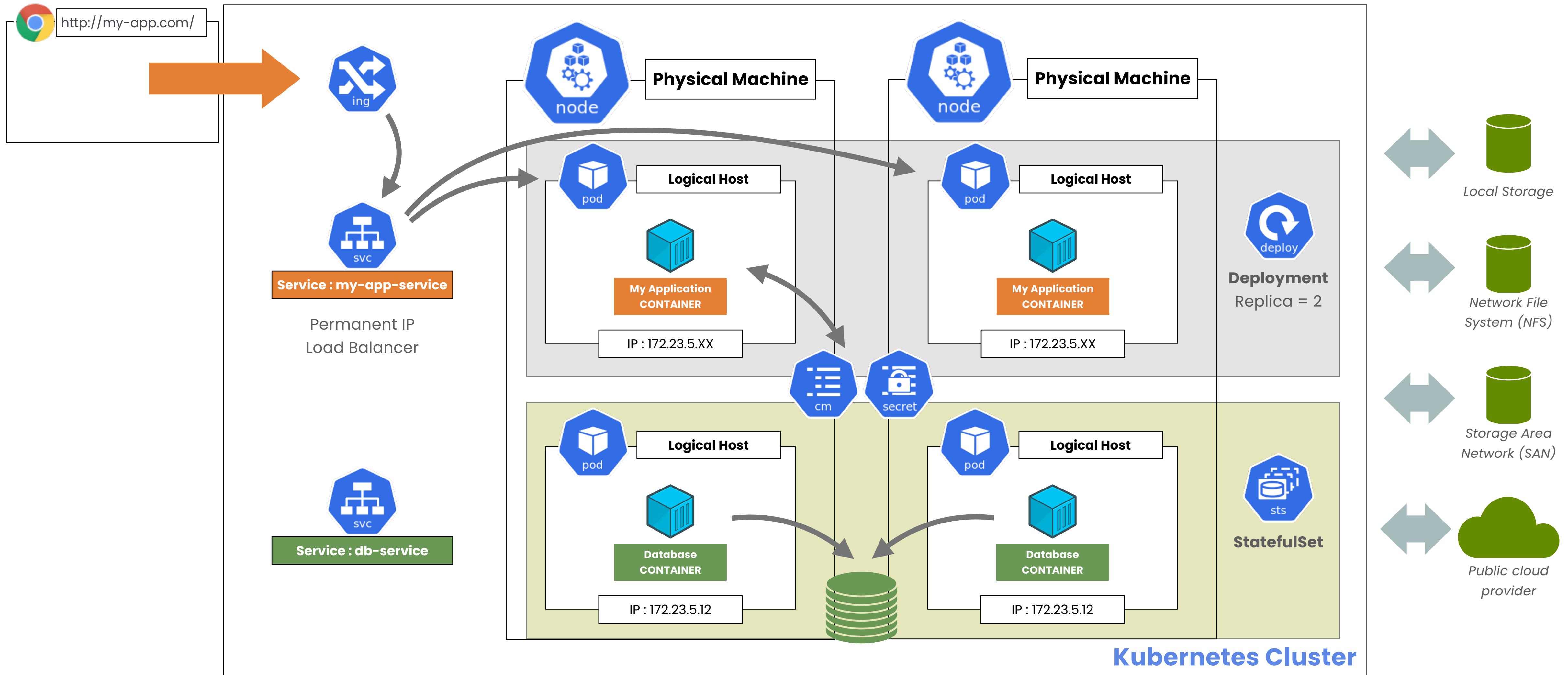
HIGH AVAILABLE ARCHITECTURE



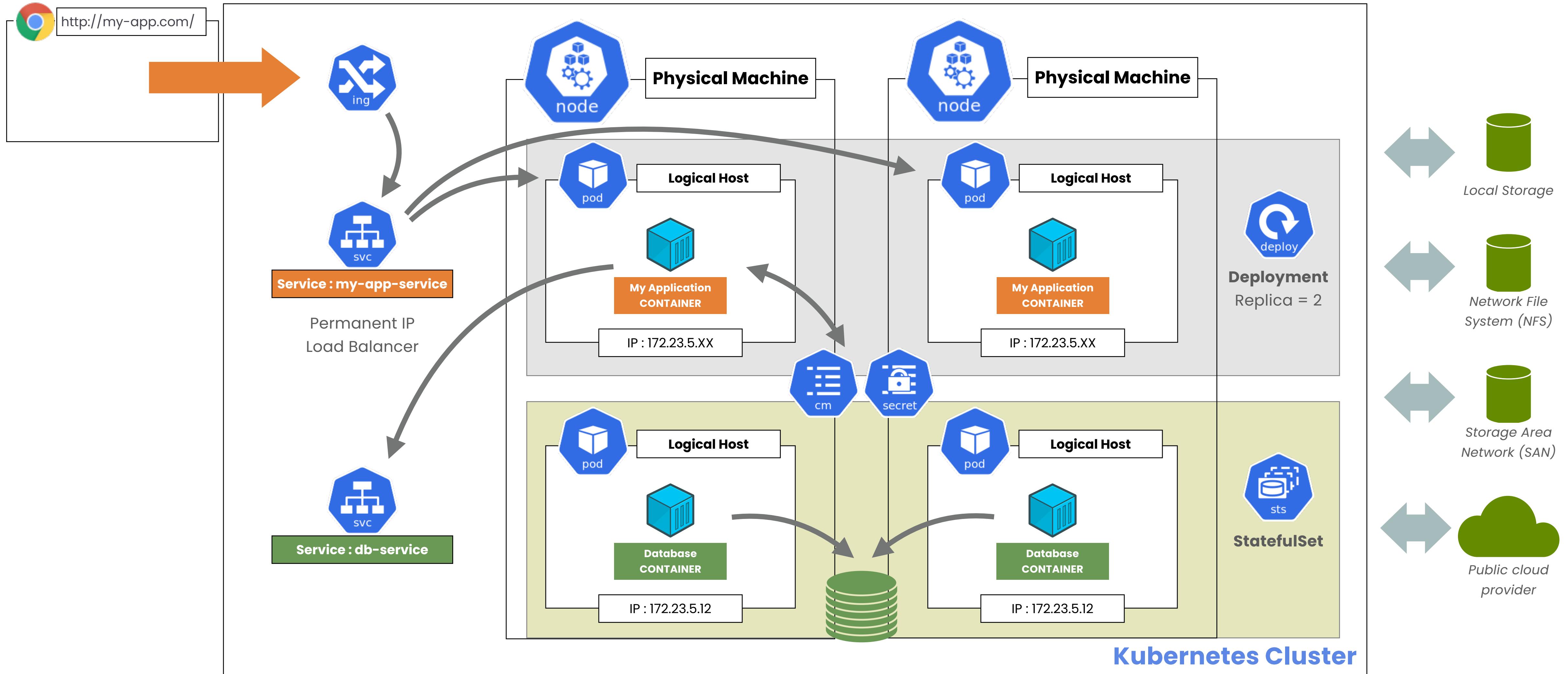
HIGH AVAILABLE ARCHITECTURE



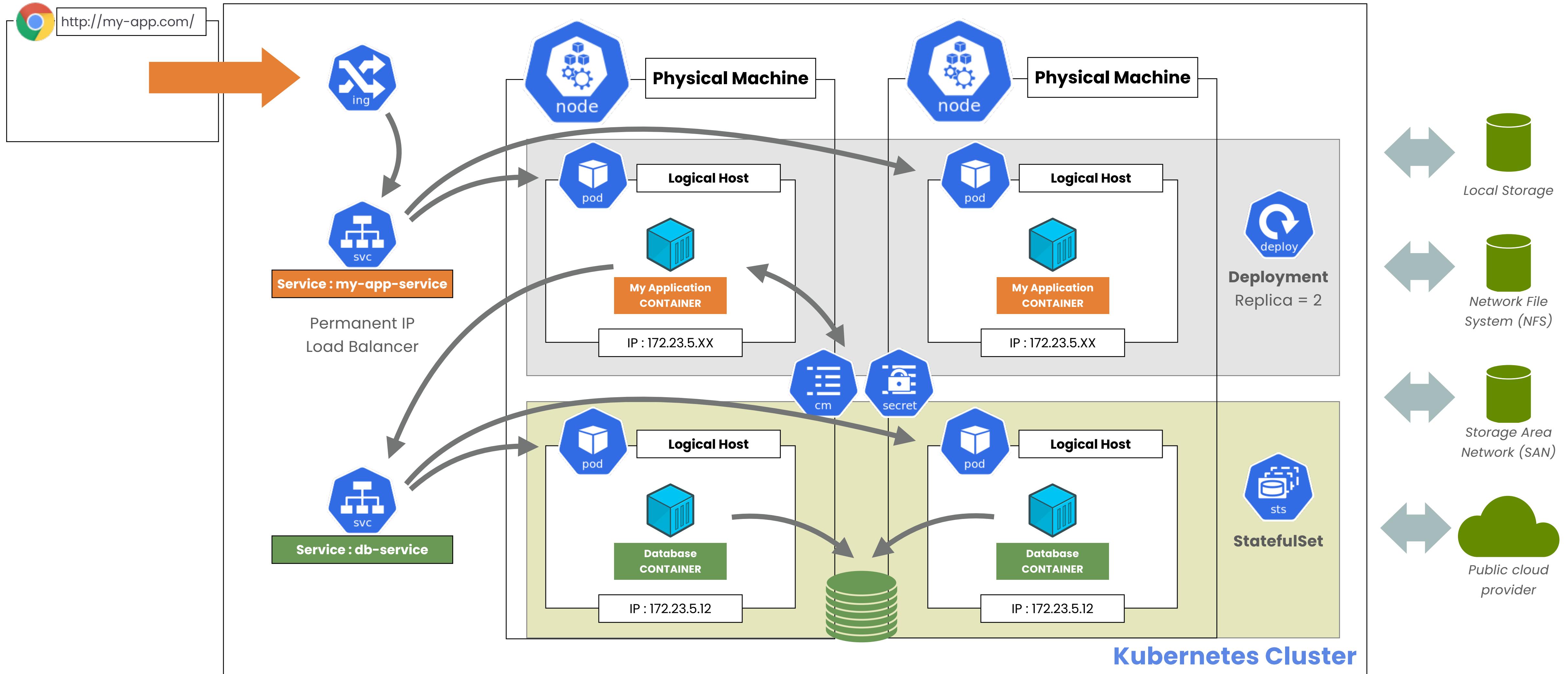
HIGH AVAILABLE ARCHITECTURE



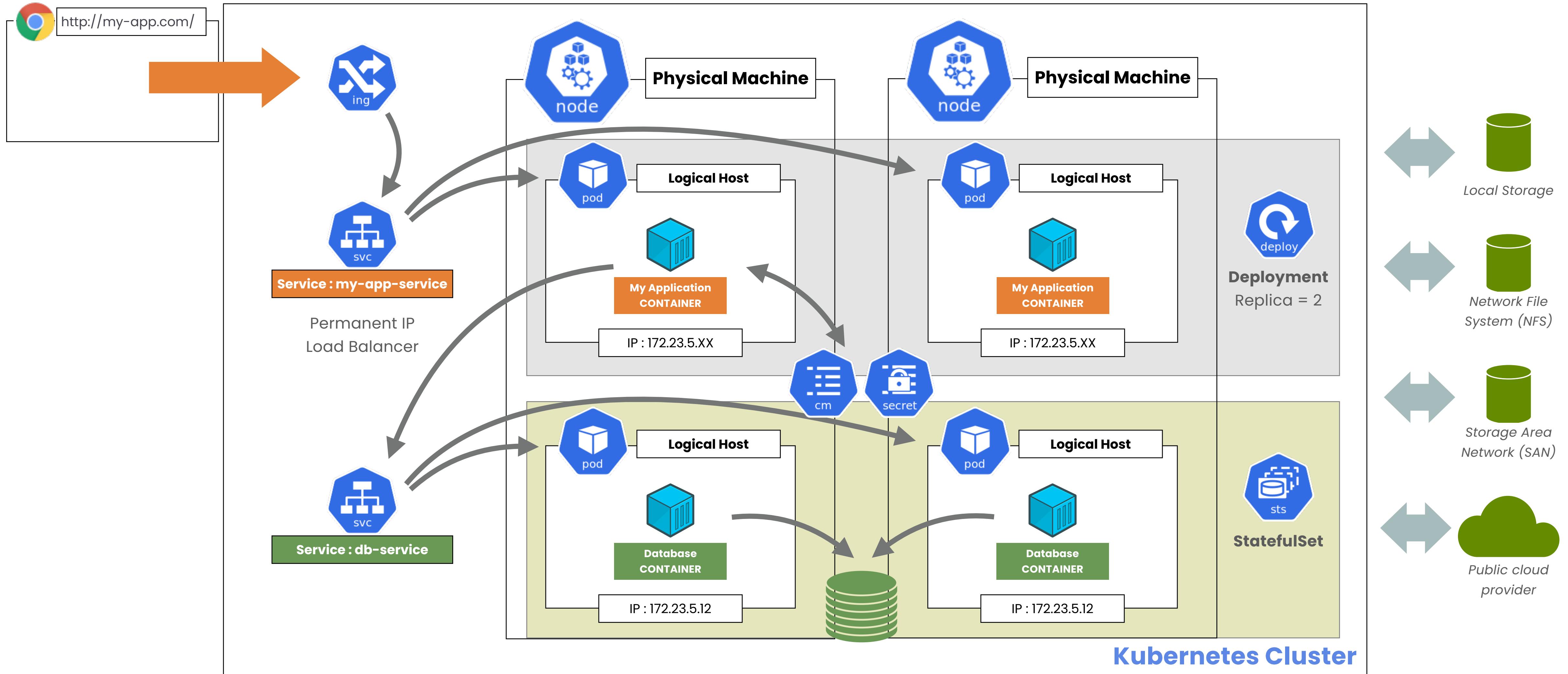
HIGH AVAILABLE ARCHITECTURE



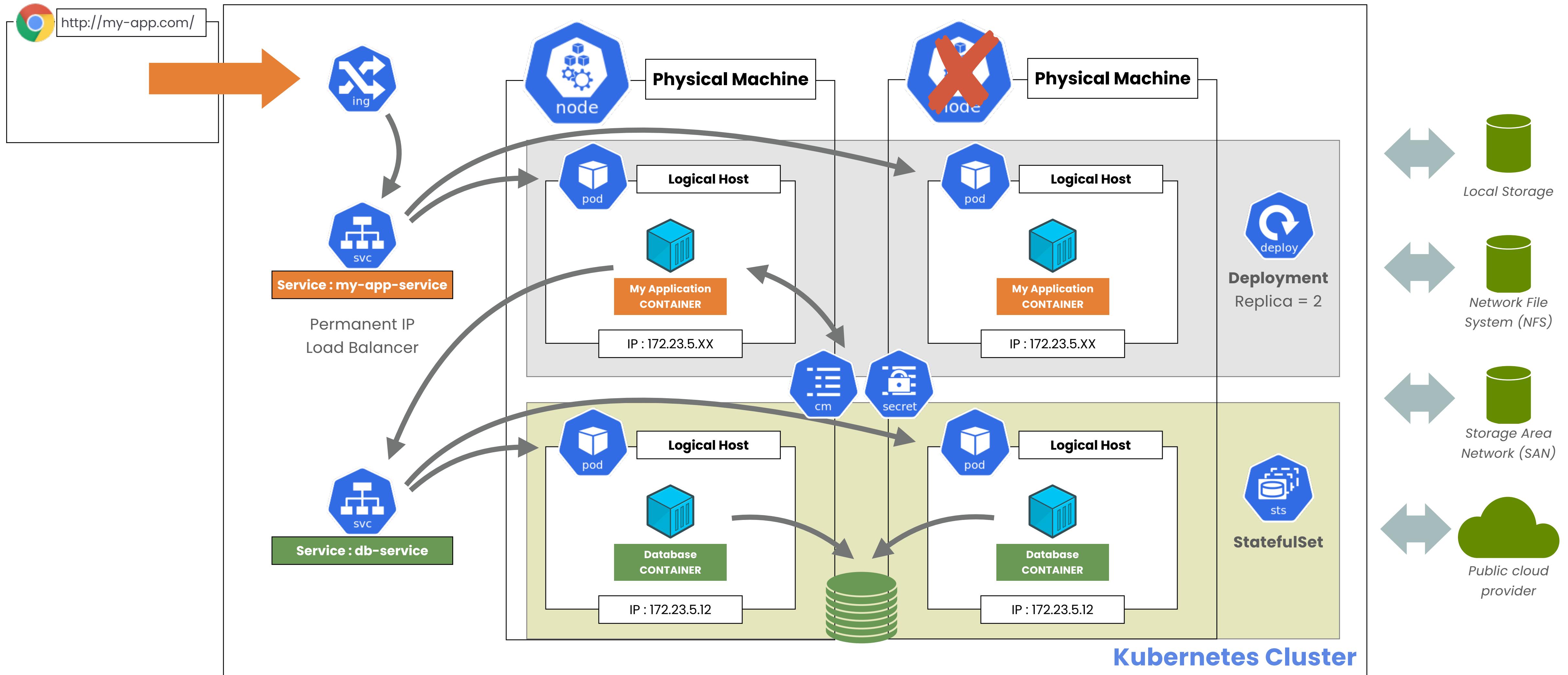
HIGH AVAILABLE ARCHITECTURE



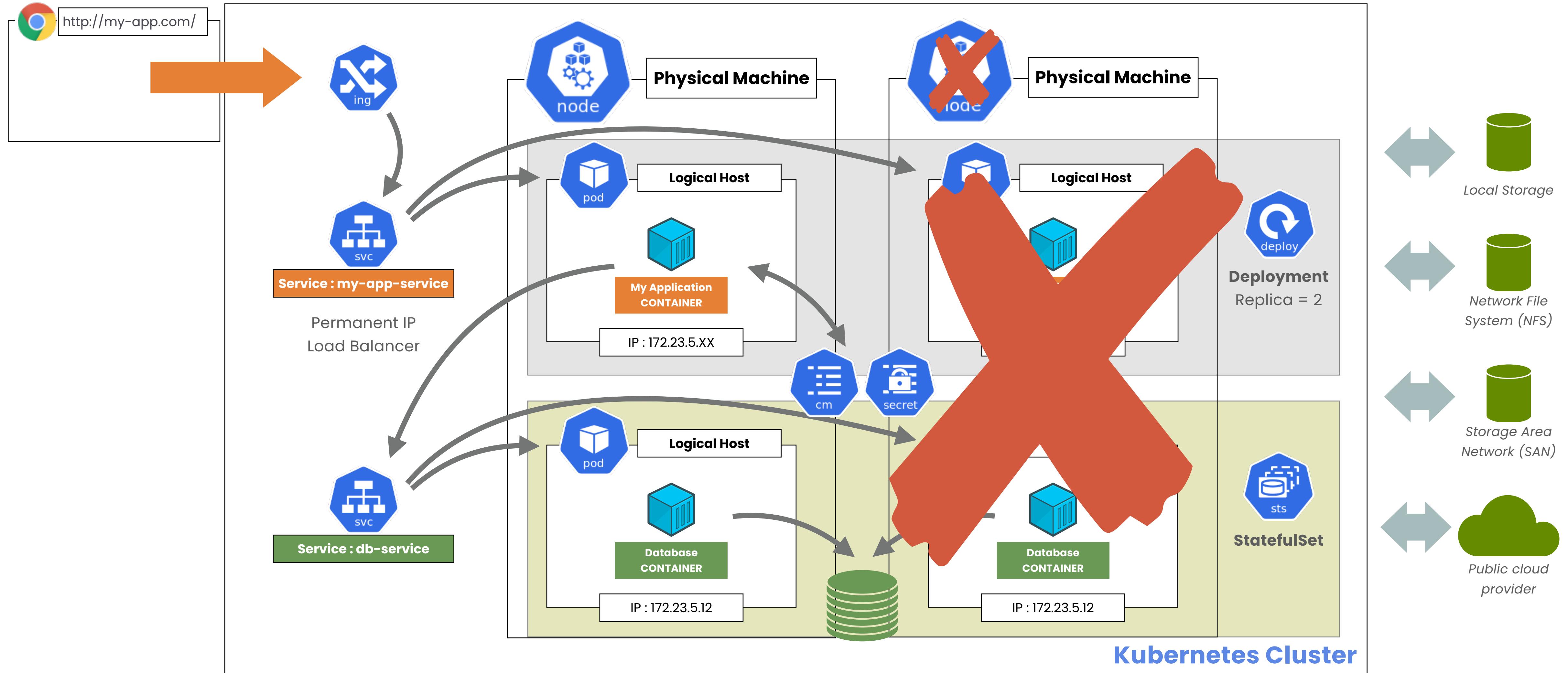
HIGH AVAILABLE ARCHITECTURE



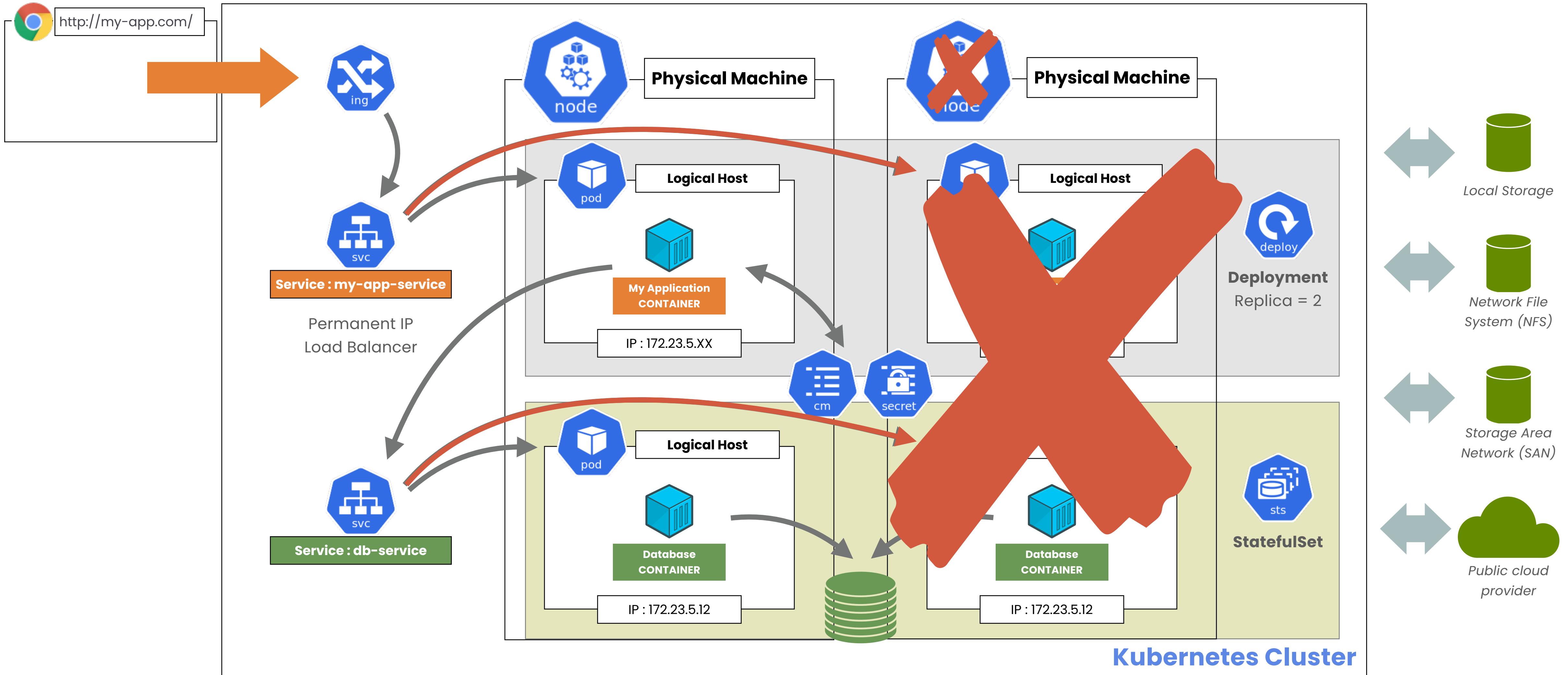
HIGH AVAILABLE ARCHITECTURE



HIGH AVAILABLE ARCHITECTURE



HIGH AVAILABLE ARCHITECTURE



THANK YOU