

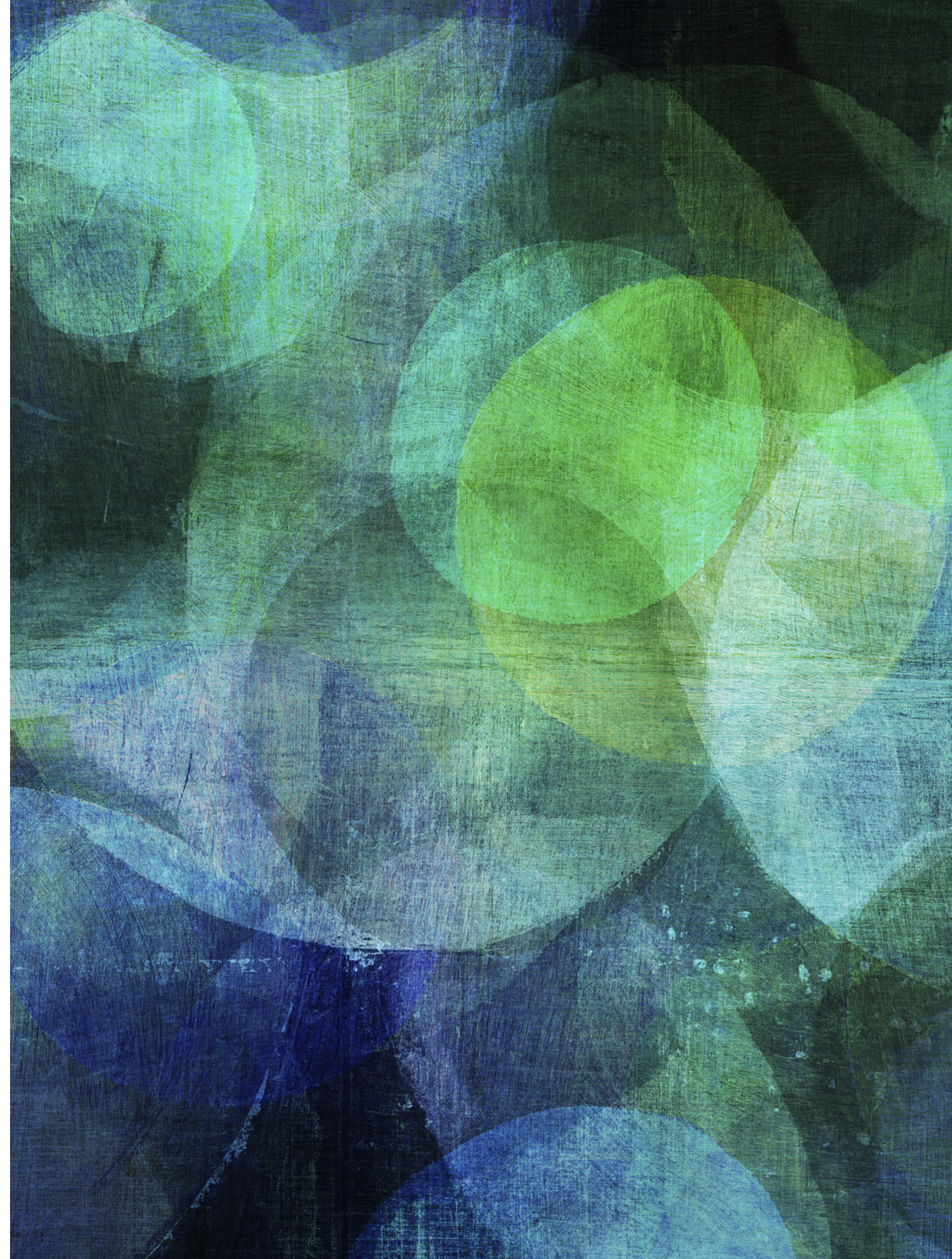
INTRODUCTION TO KUBERNETES



INTRODUCTION TO KUBERNETES

Kubernetes Objects and Components

AGENDA



AGENDA

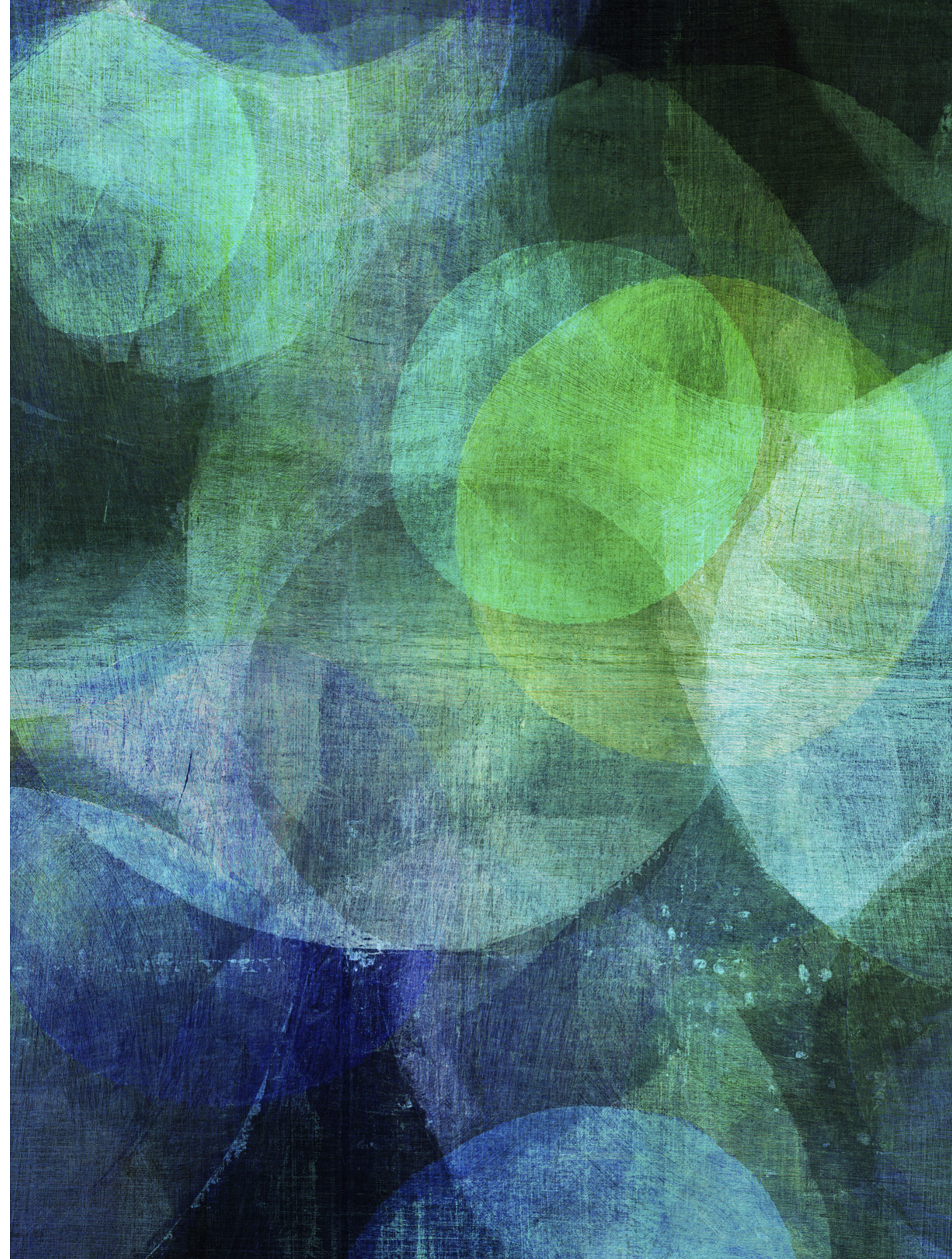
– Node



AGENDA

– Node

– Pod



AGENDA

- Node
- Pod
- Service



AGENDA

- Node
- Pod
- Service
- Ingress



AGENDA

- Node
- Pod
- Service
- Ingress



K8S OBJECTS & COMPONENTS



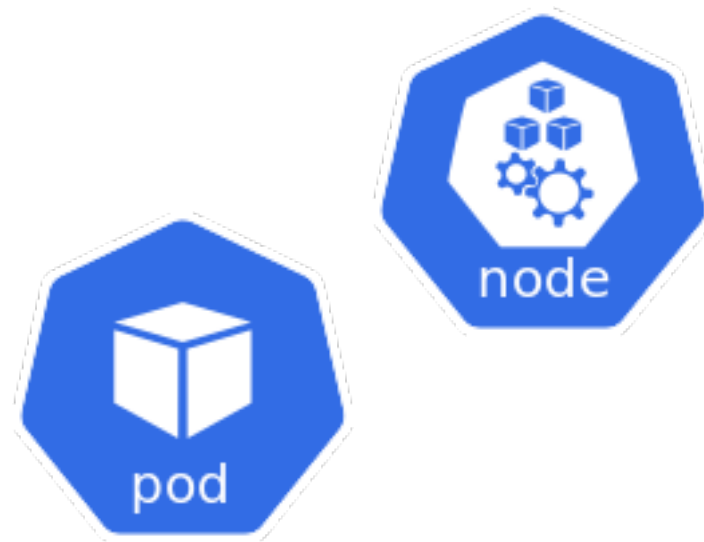
kubernetes

K8S OBJECTS & COMPONENTS



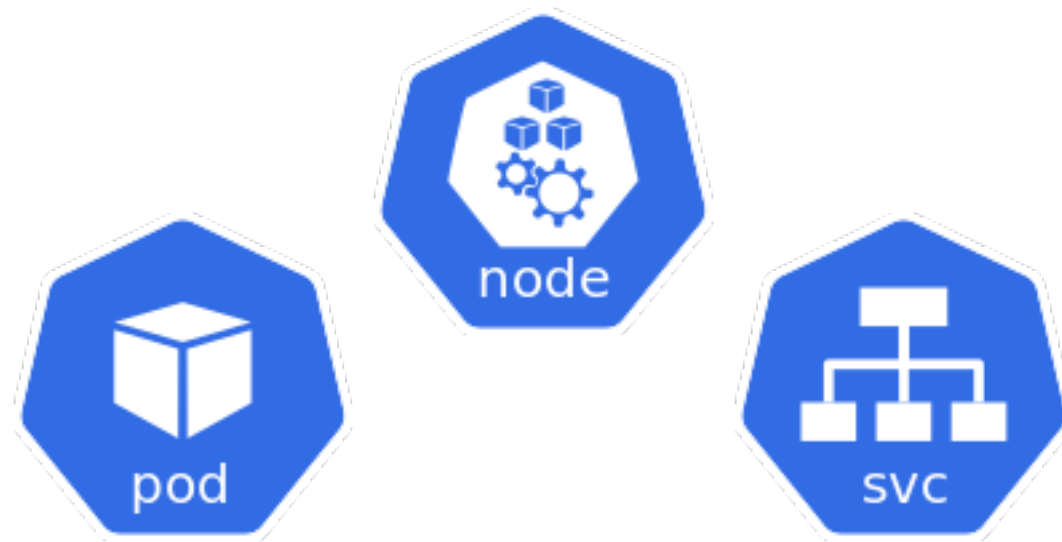
kubernetes

K8S OBJECTS & COMPONENTS



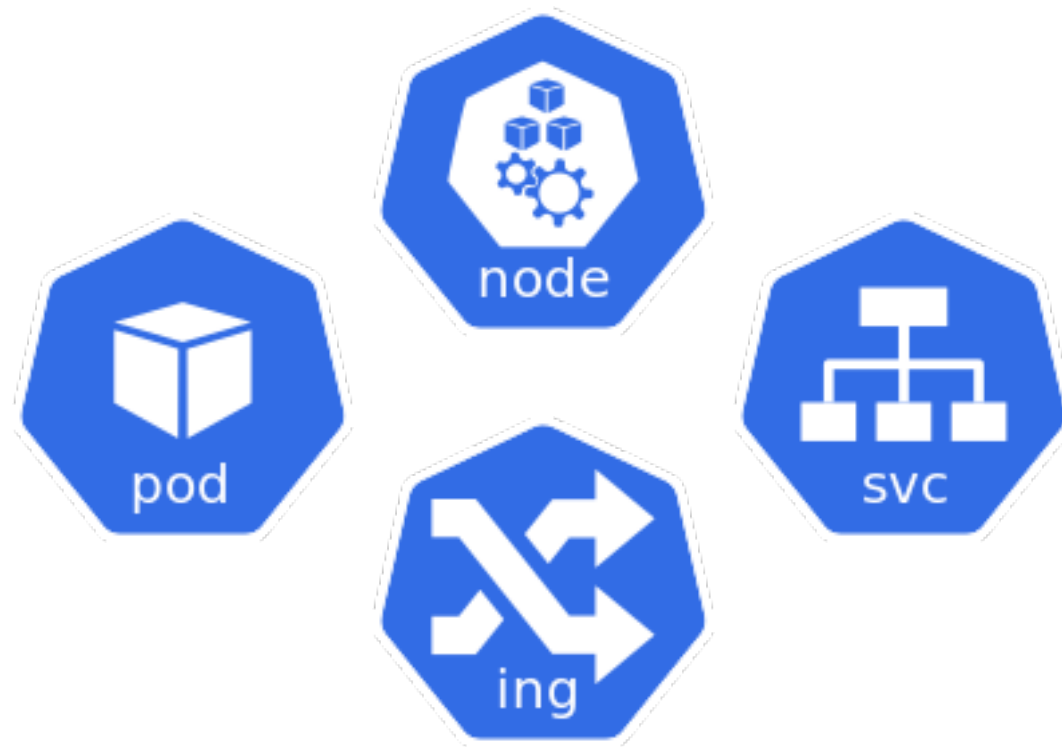
kubernetes

K8S OBJECTS & COMPONENTS



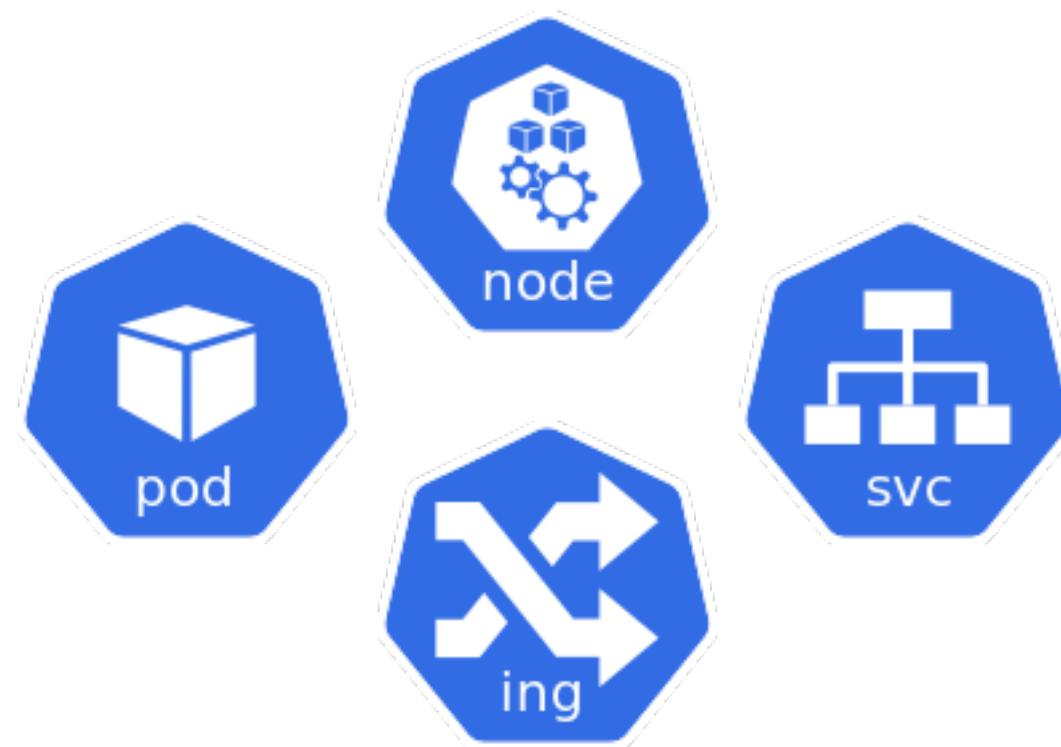
kubernetes

K8S OBJECTS & COMPONENTS



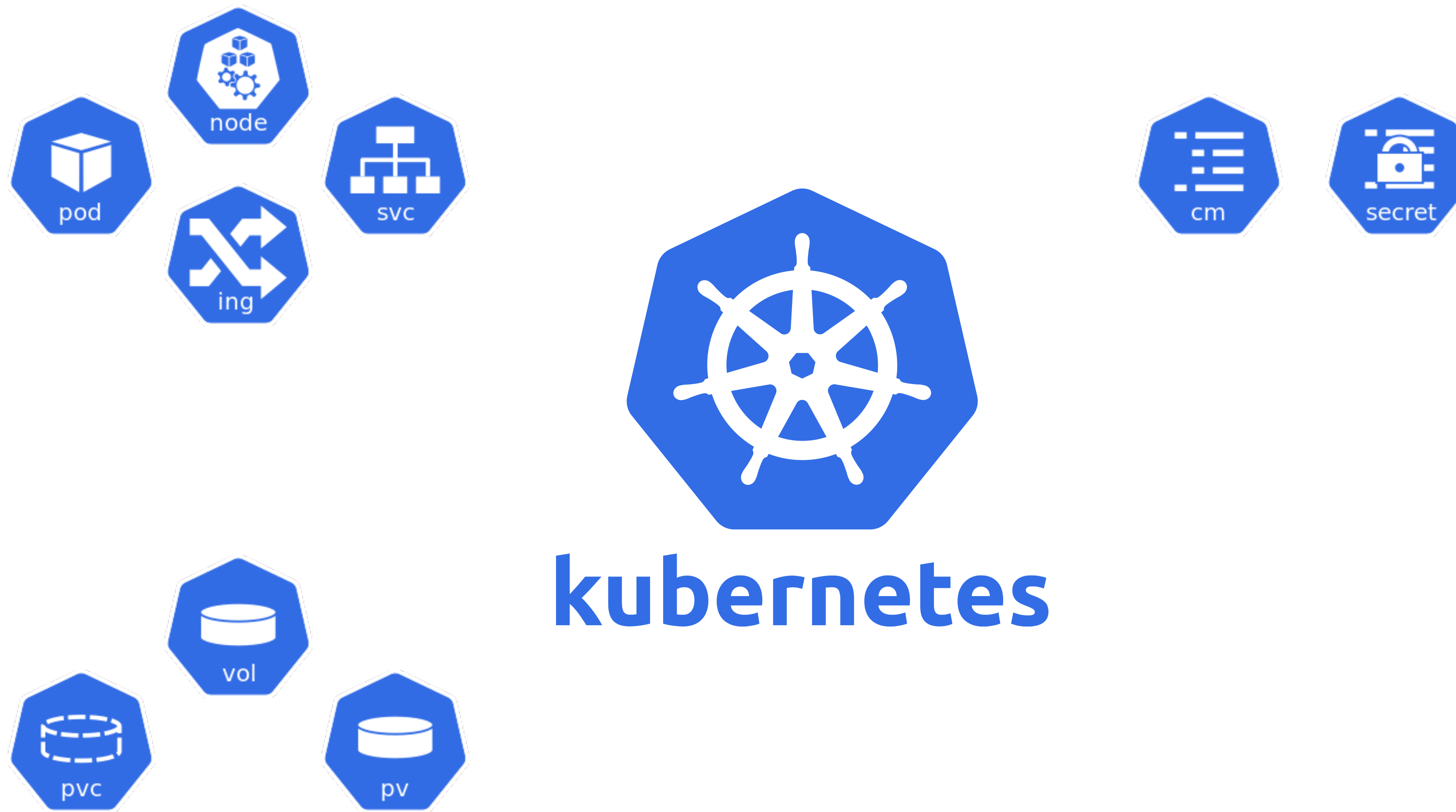
kubernetes

K8S OBJECTS & COMPONENTS

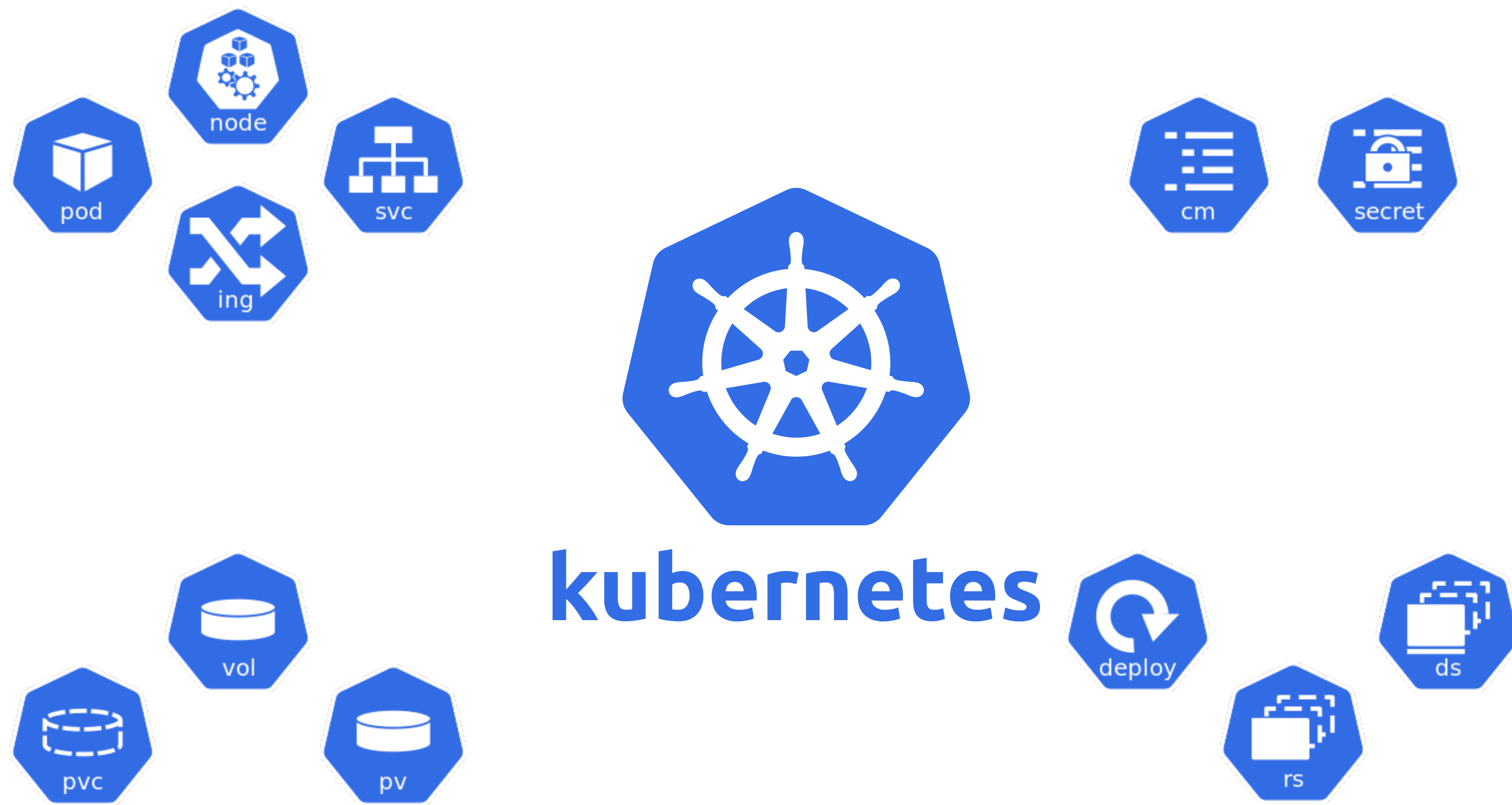


kubernetes

K8S OBJECTS & COMPONENTS

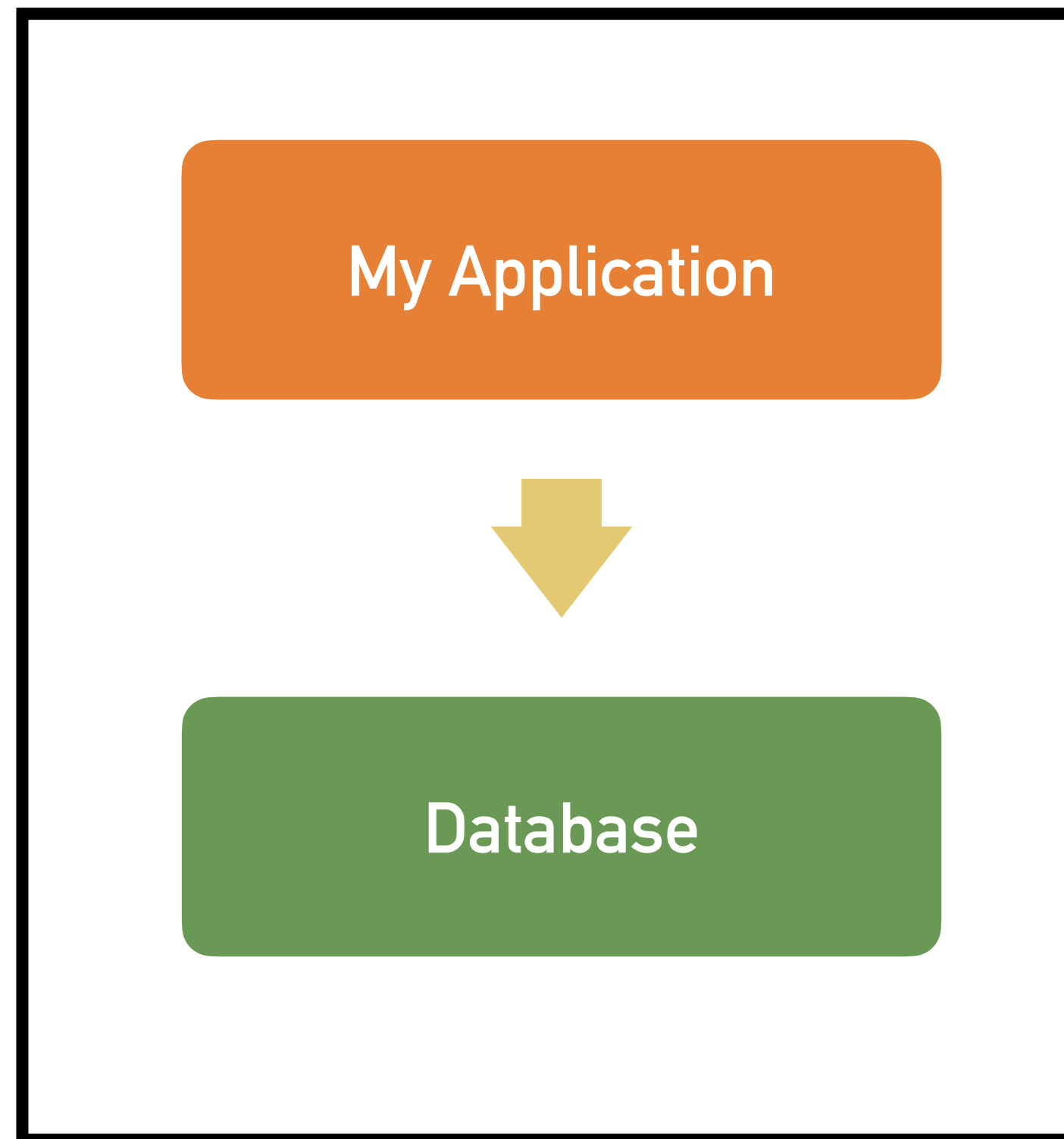


K8S OBJECTS & COMPONENTS

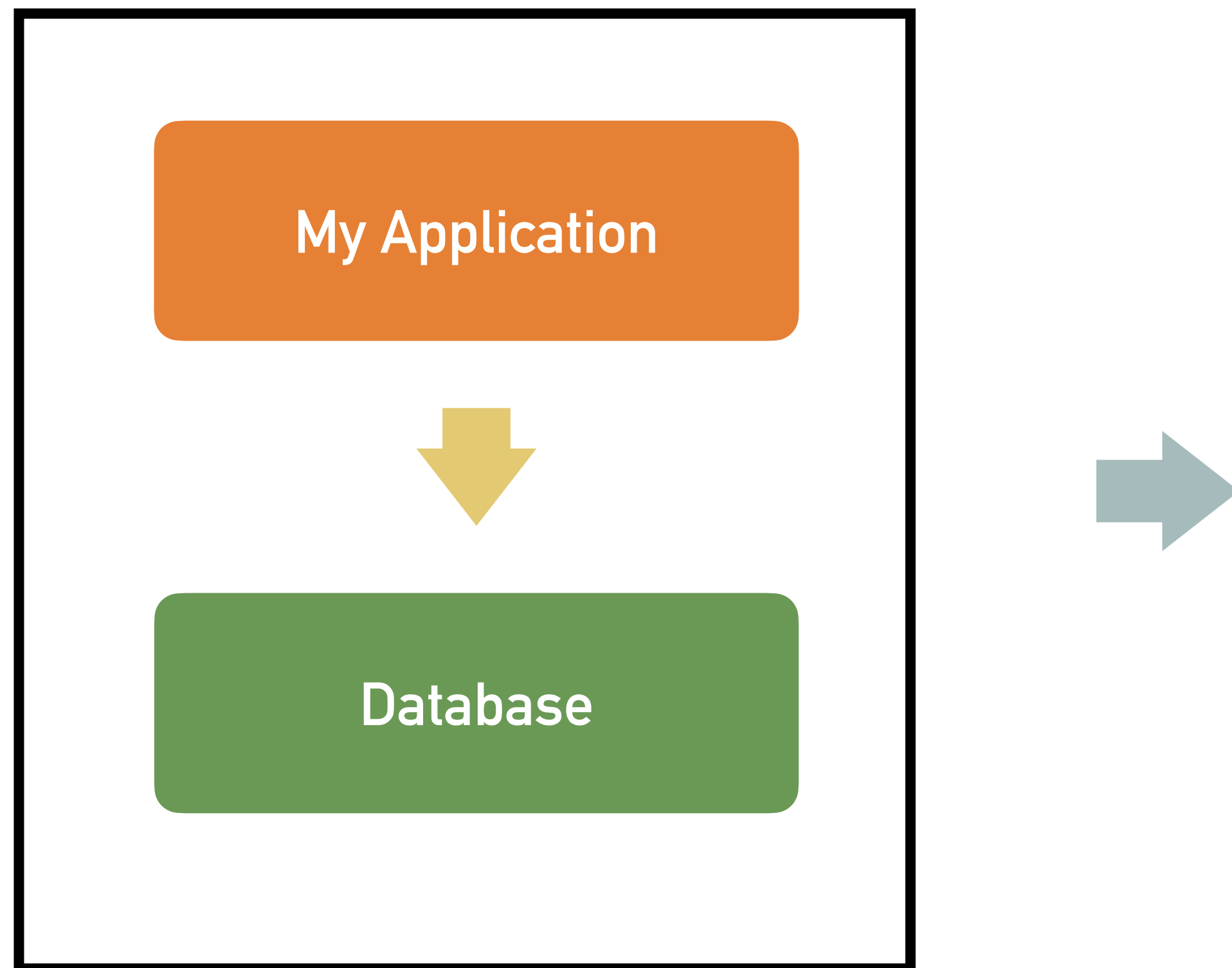


SIMPLE APPLICATION ON K8S

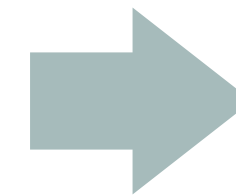
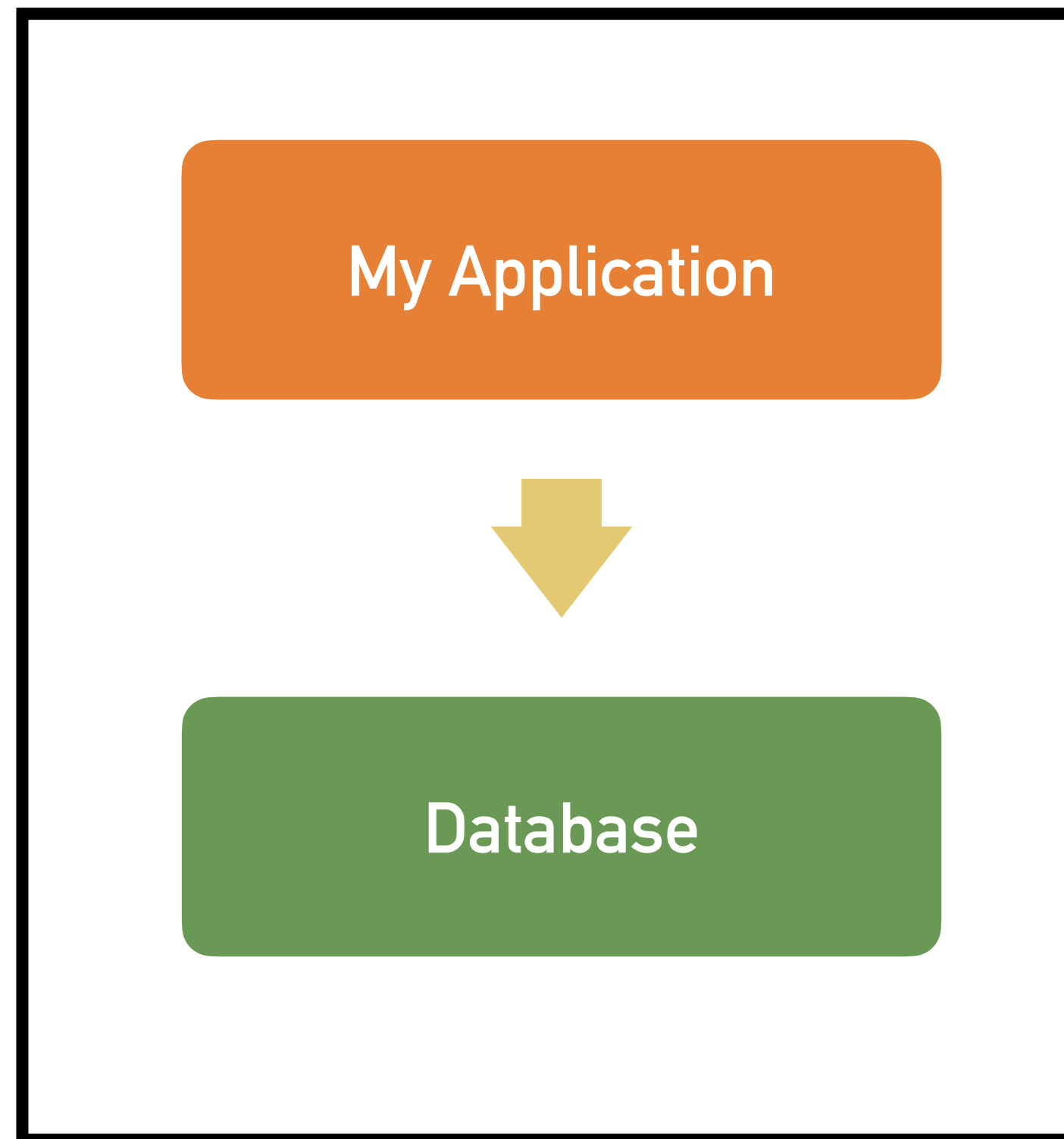
SIMPLE APPLICATION ON K8S



SIMPLE APPLICATION ON K8S



SIMPLE APPLICATION ON K8S



kubernetes

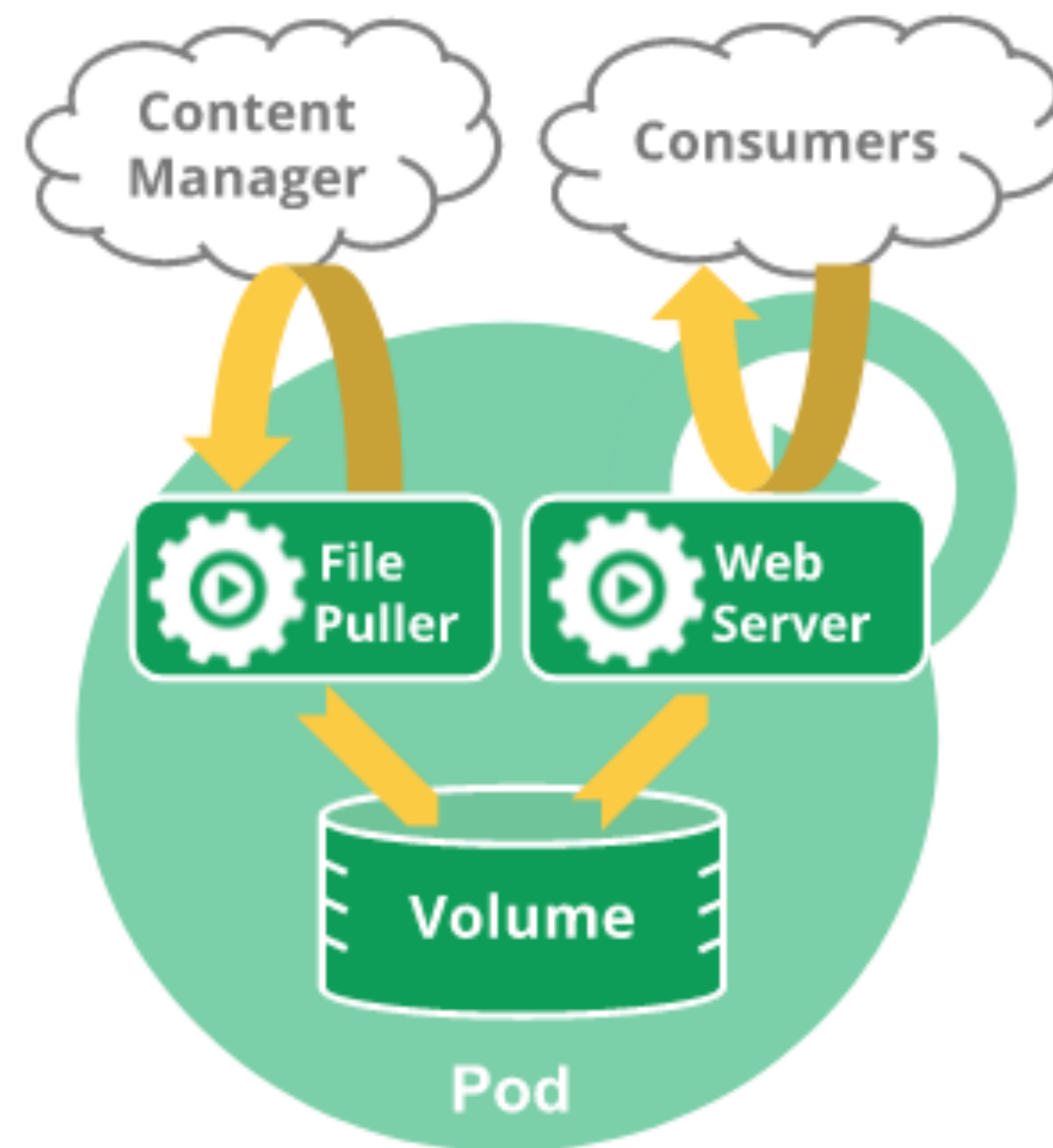
NODE & POD

.....

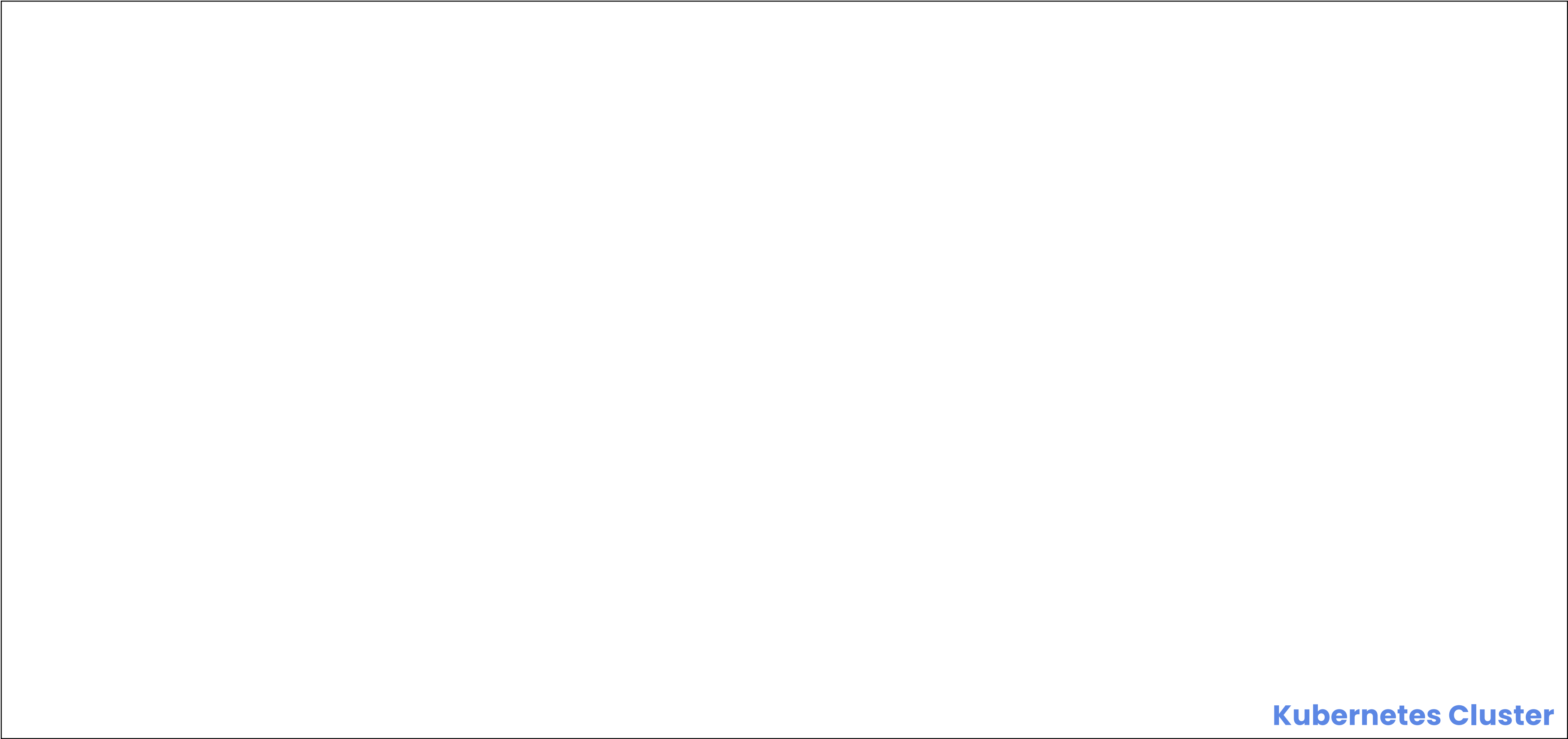
POD

the **smallest deployable units** of computing that you **can create and manage in Kubernetes**.

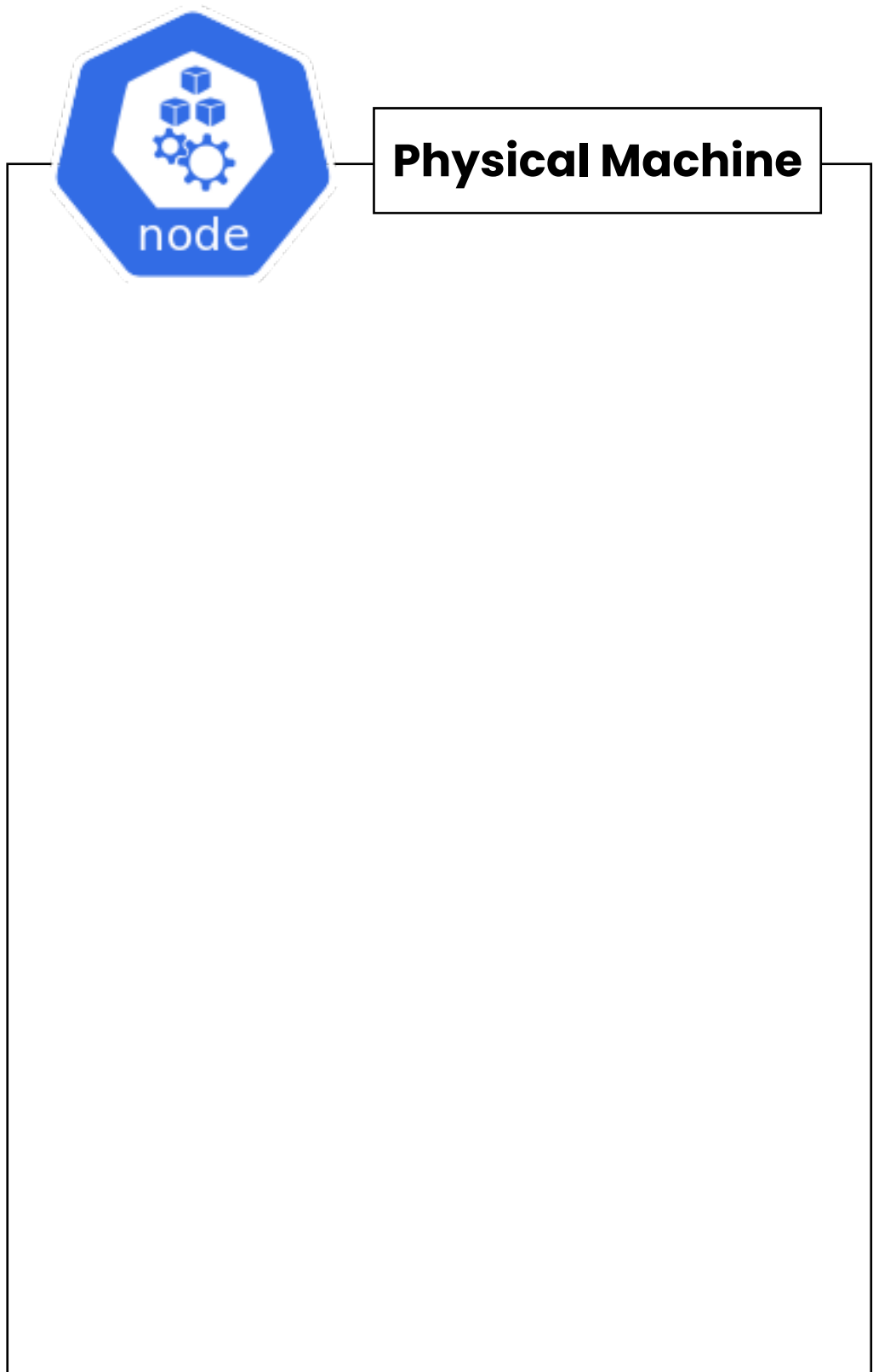
A **Pod** is a group of one or more containers, with **shared storage and network resources**, and a specification for how to run the containers. A Pod's contents are always co-located and co-scheduled, and run in a shared context.



NODE VS POD

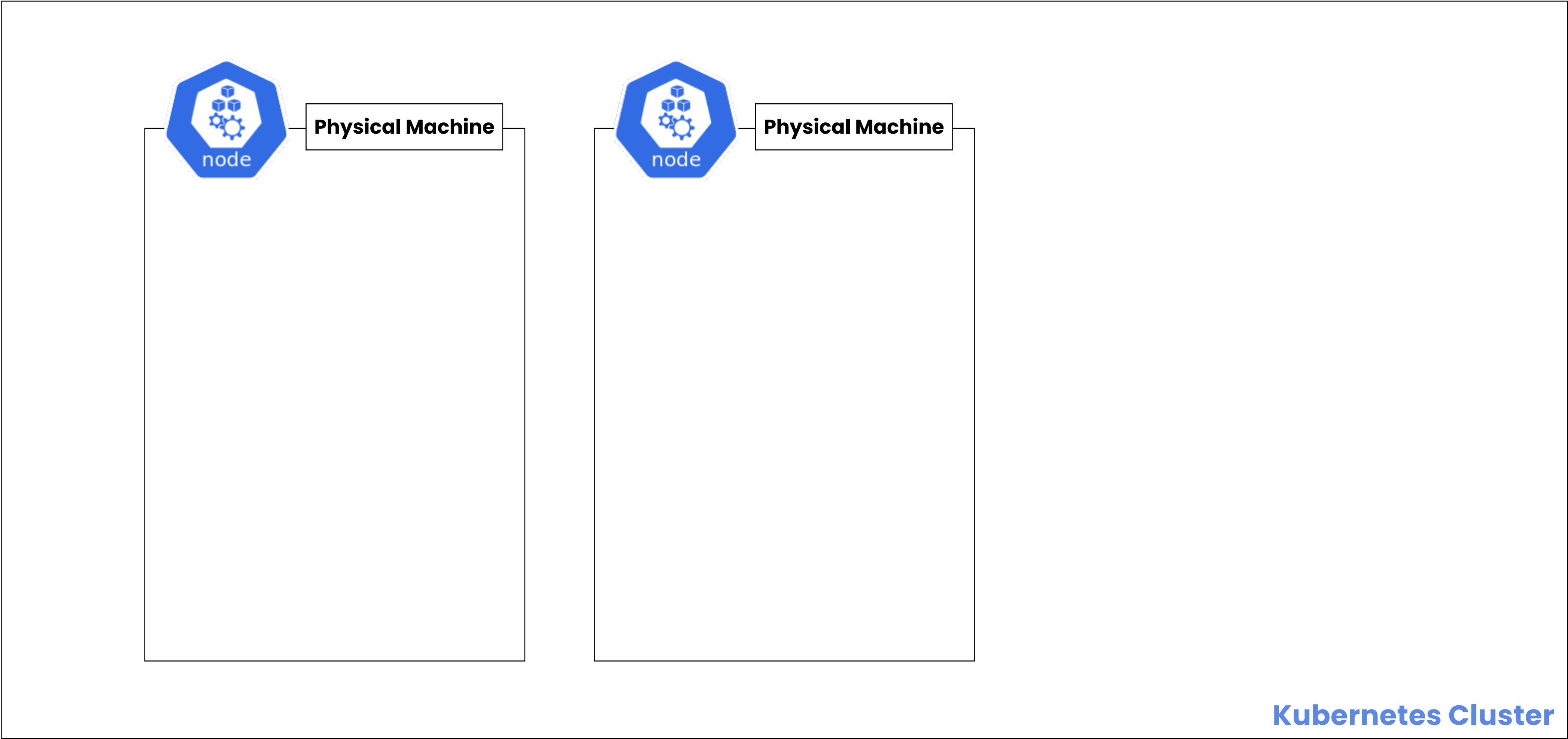


NODE VS POD

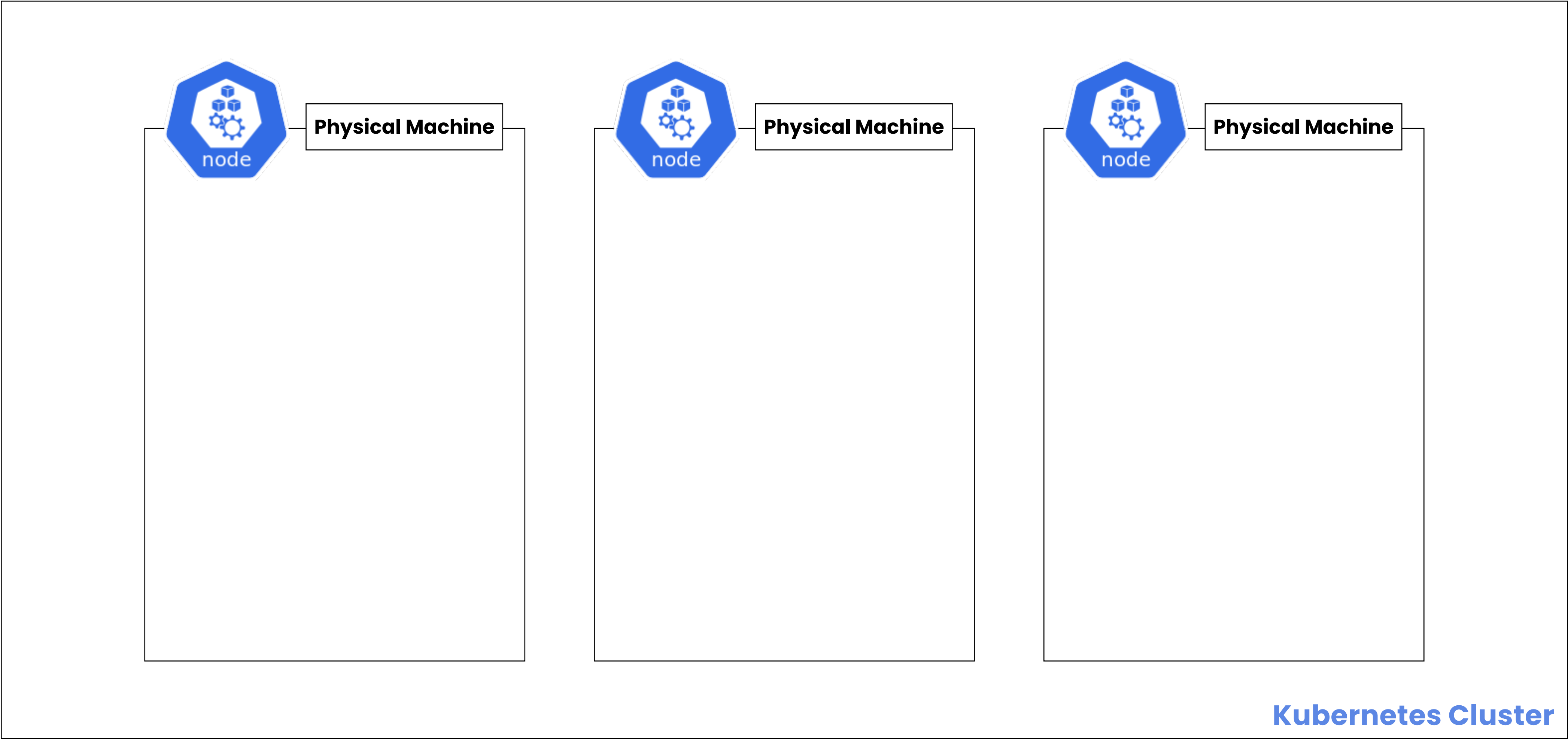


Kubernetes Cluster

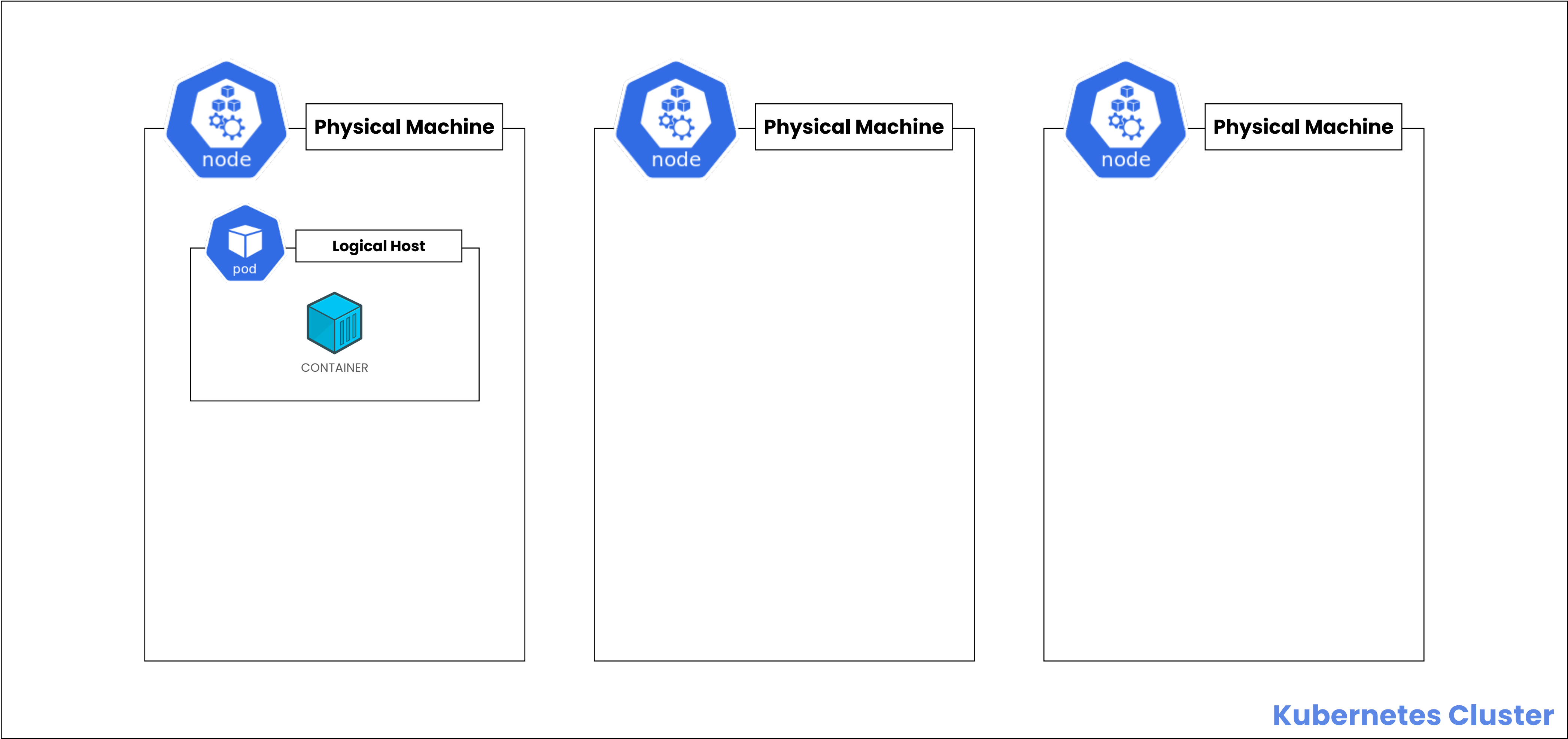
NODE VS POD



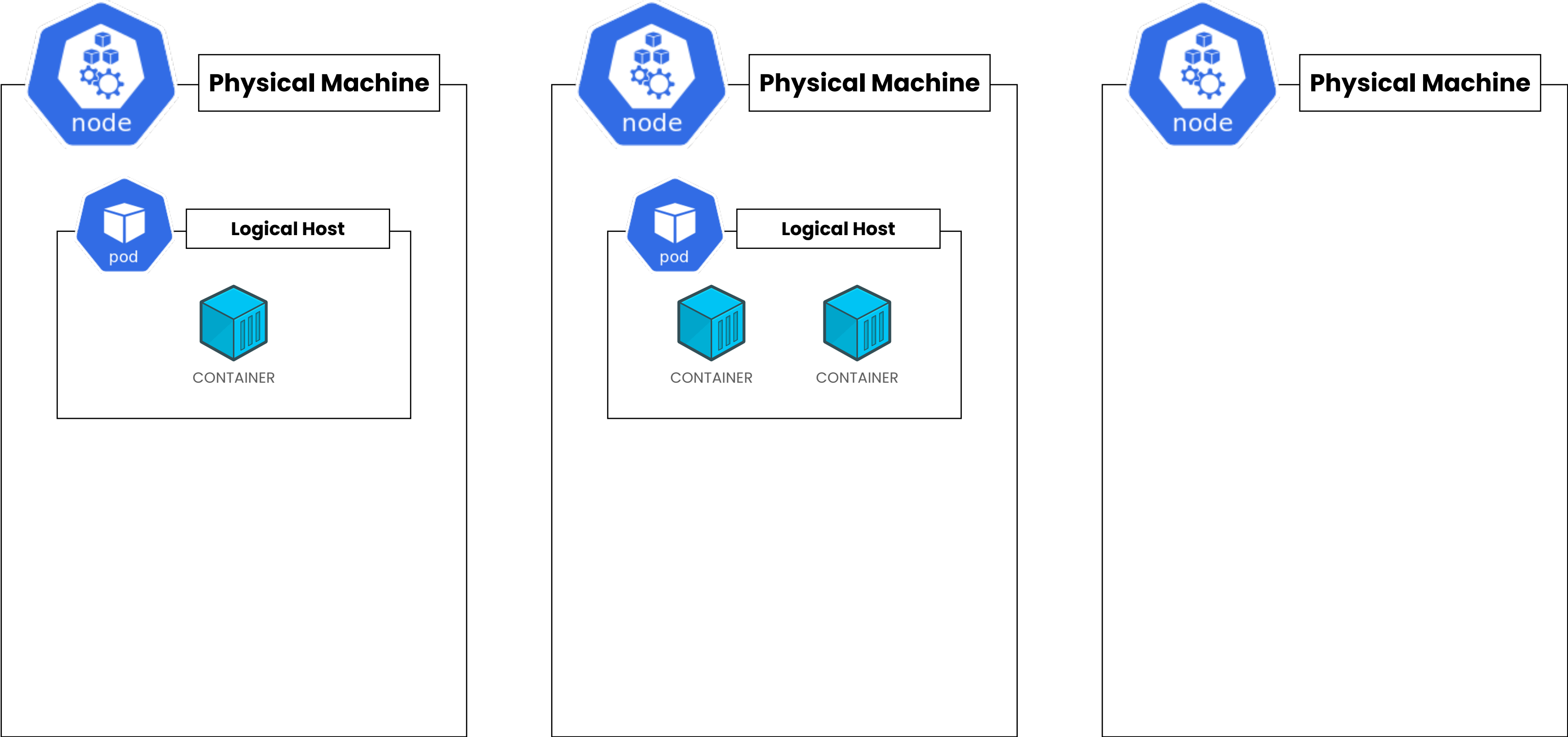
NODE VS POD



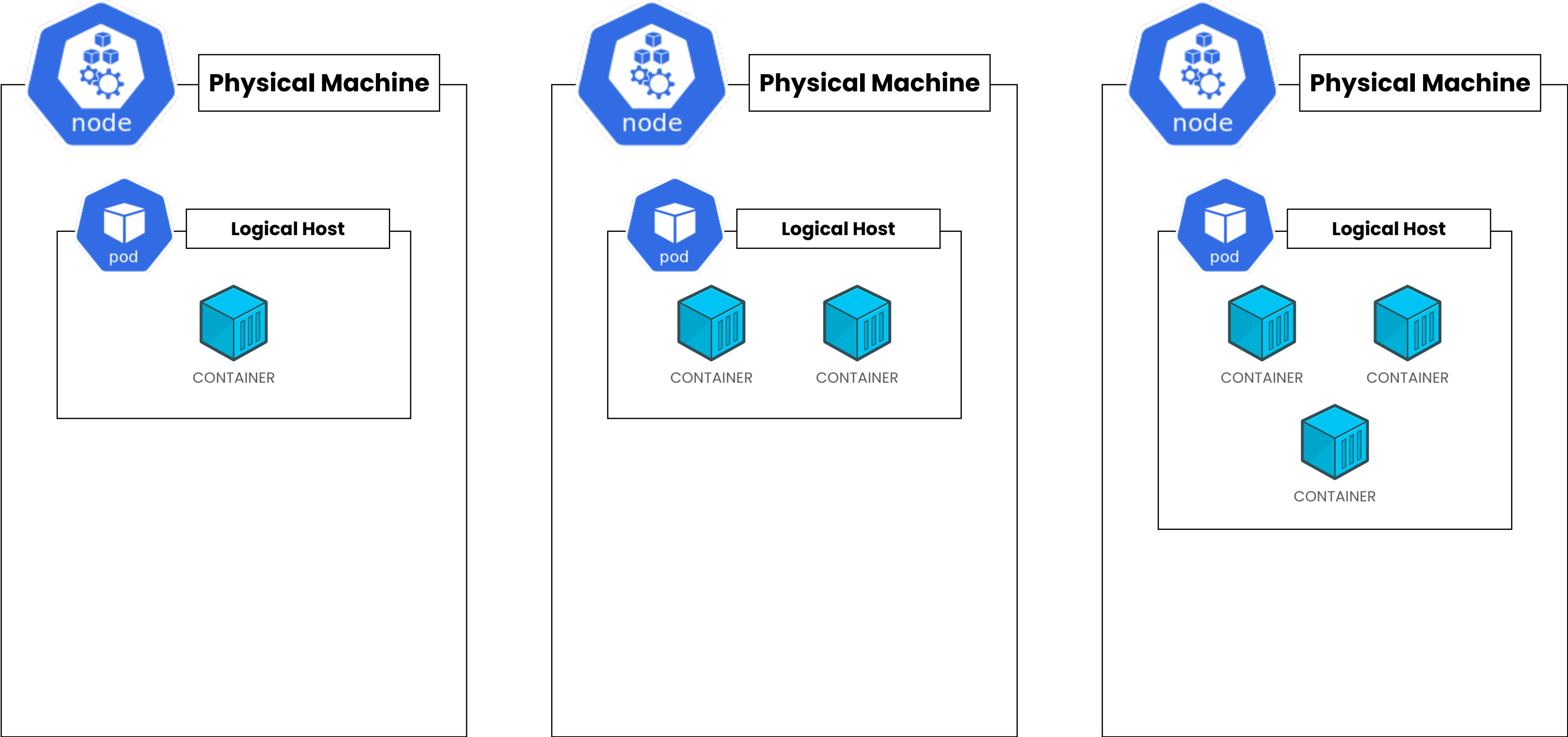
NODE VS POD



NODE VS POD

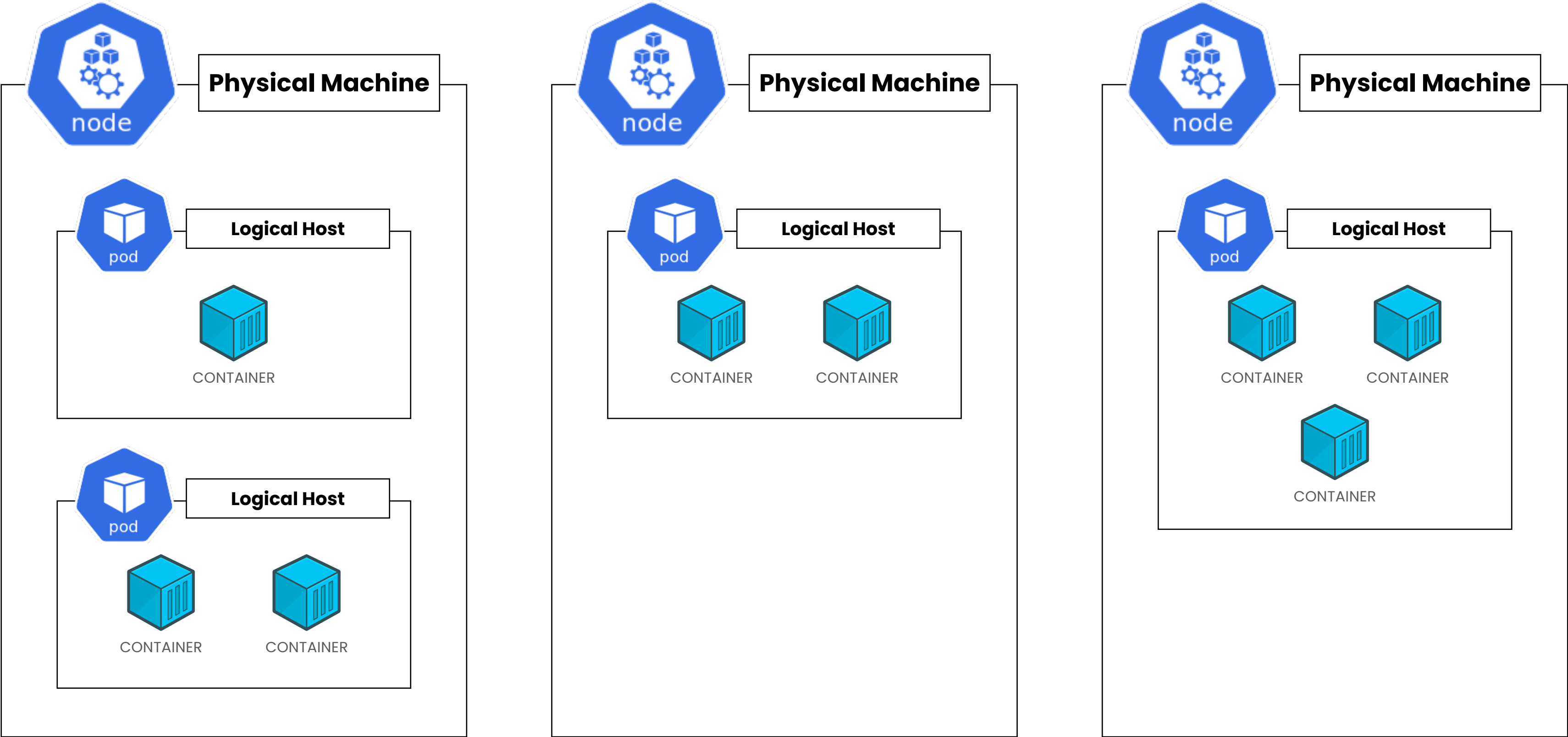


NODE VS POD

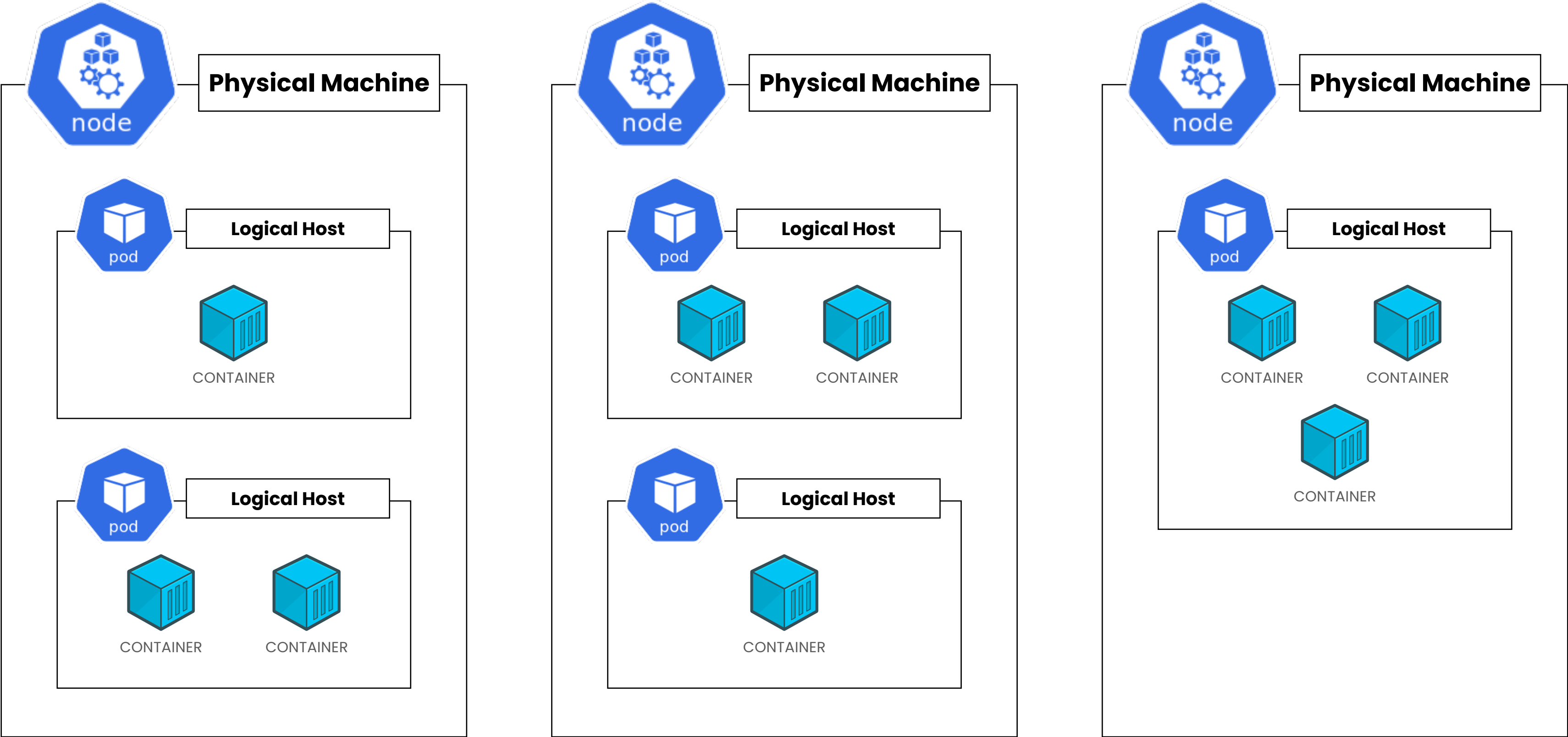


Kubernetes Cluster

NODE VS POD



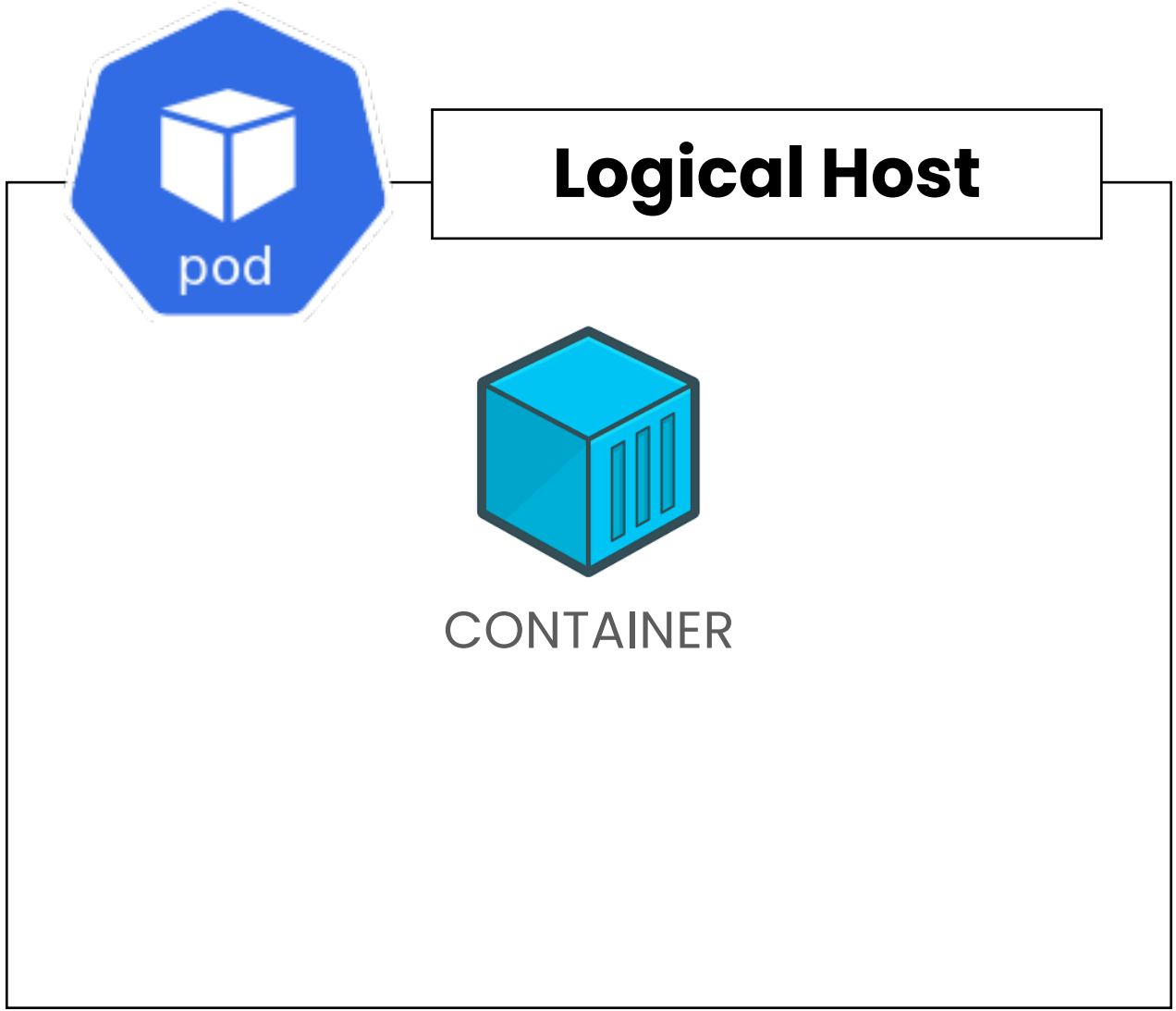
NODE VS POD



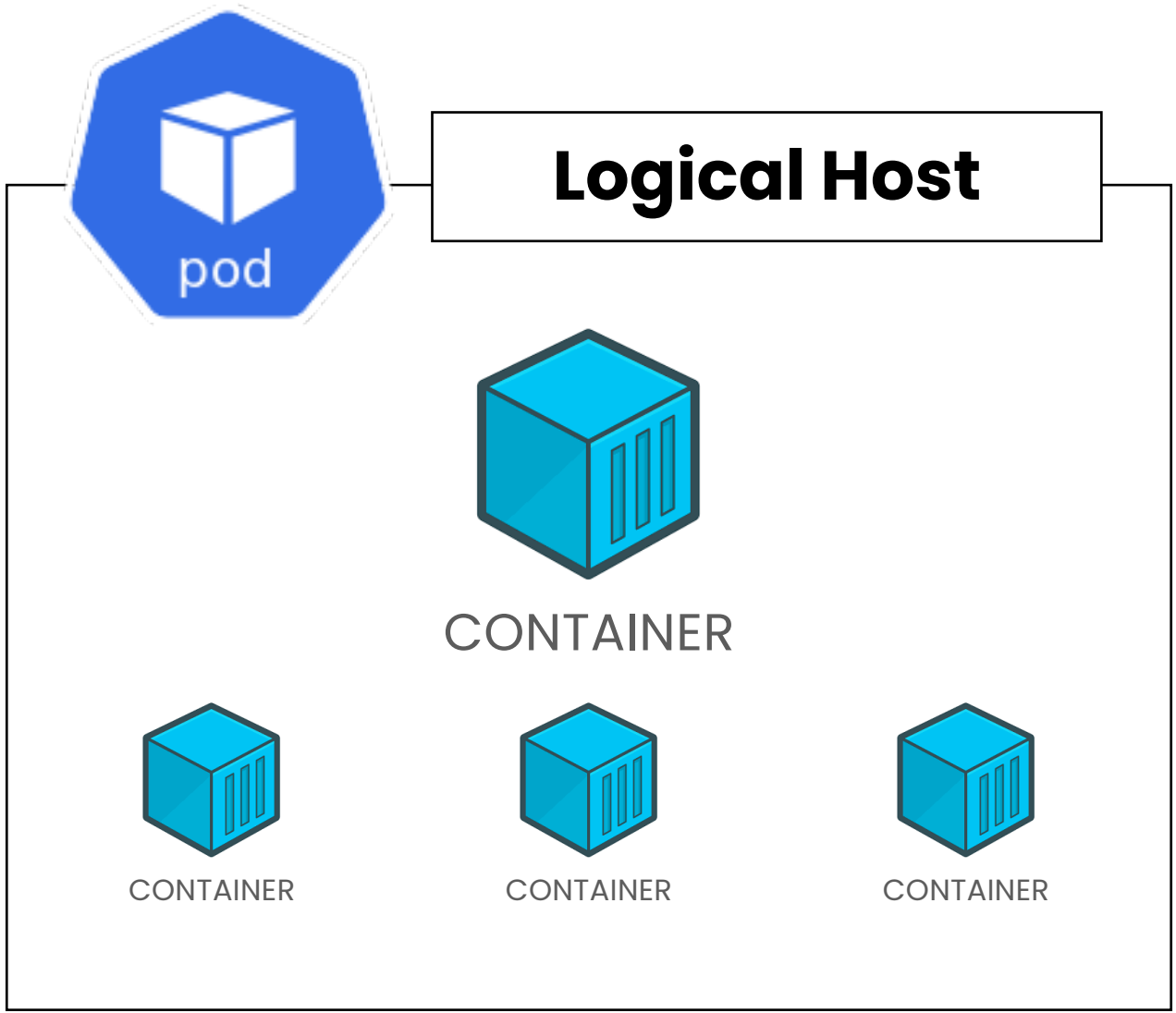
Kubernetes Cluster

POD – STORAGE AND NETWORK

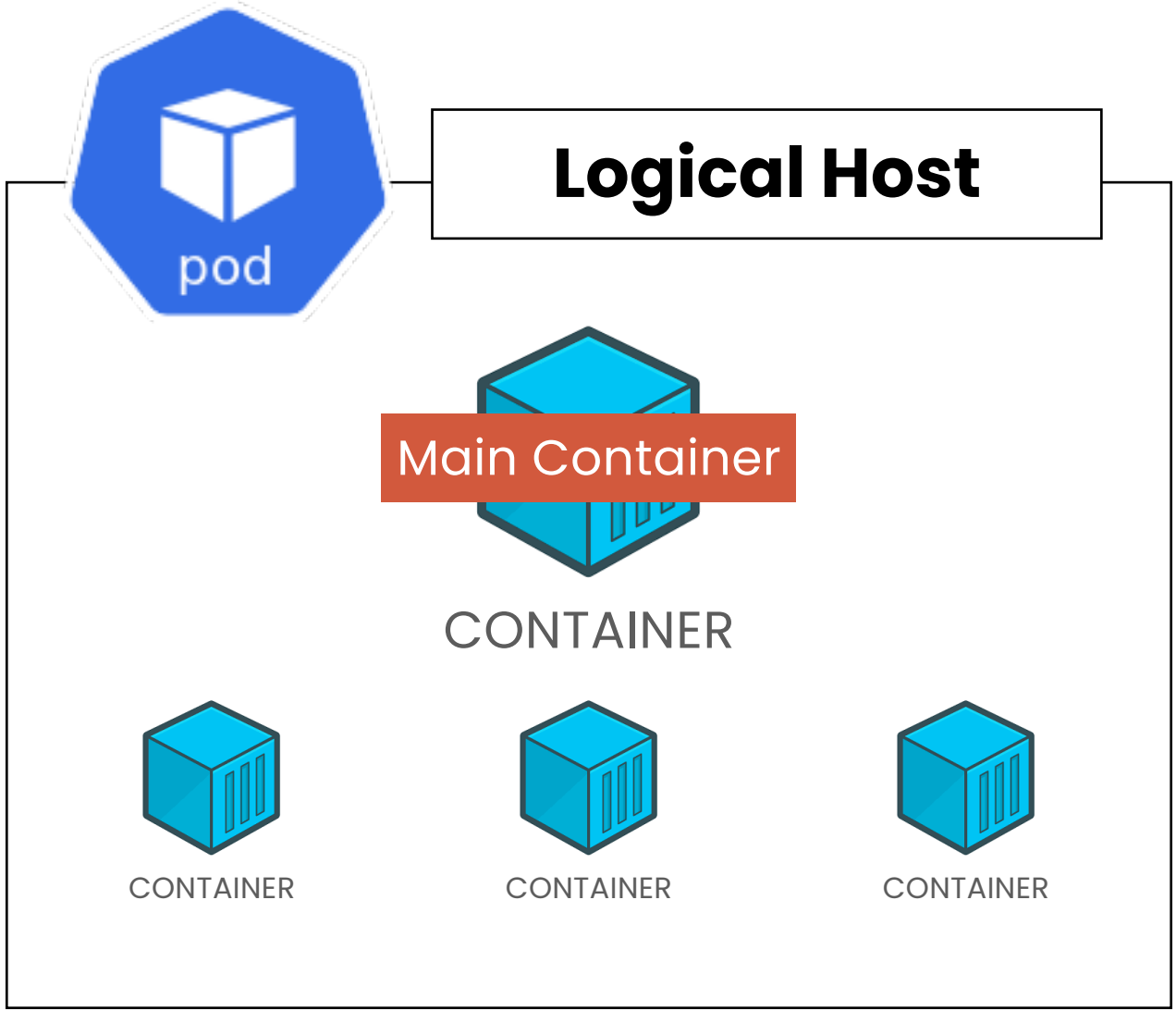
POD – STORAGE AND NETWORK



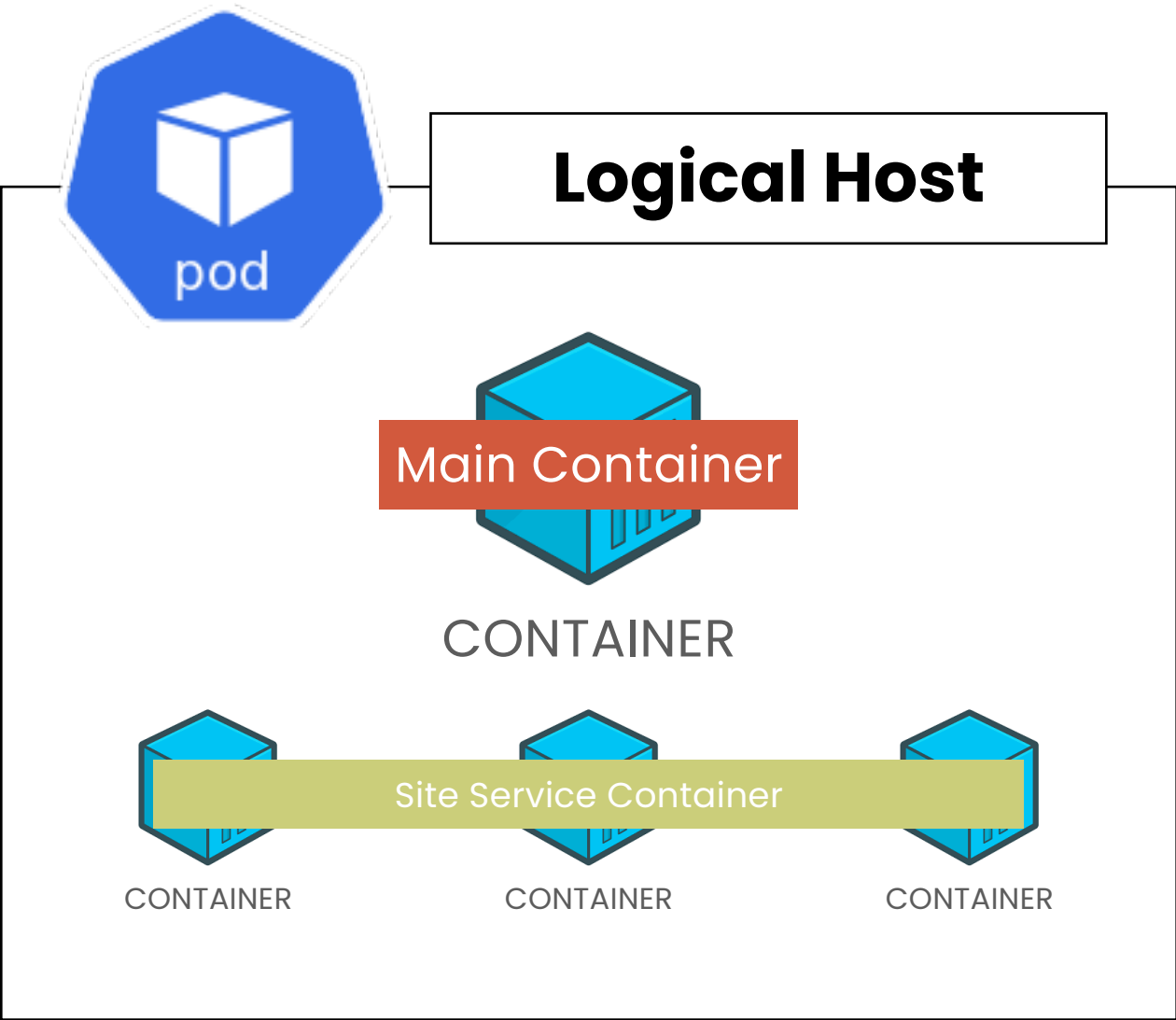
POD – STORAGE AND NETWORK



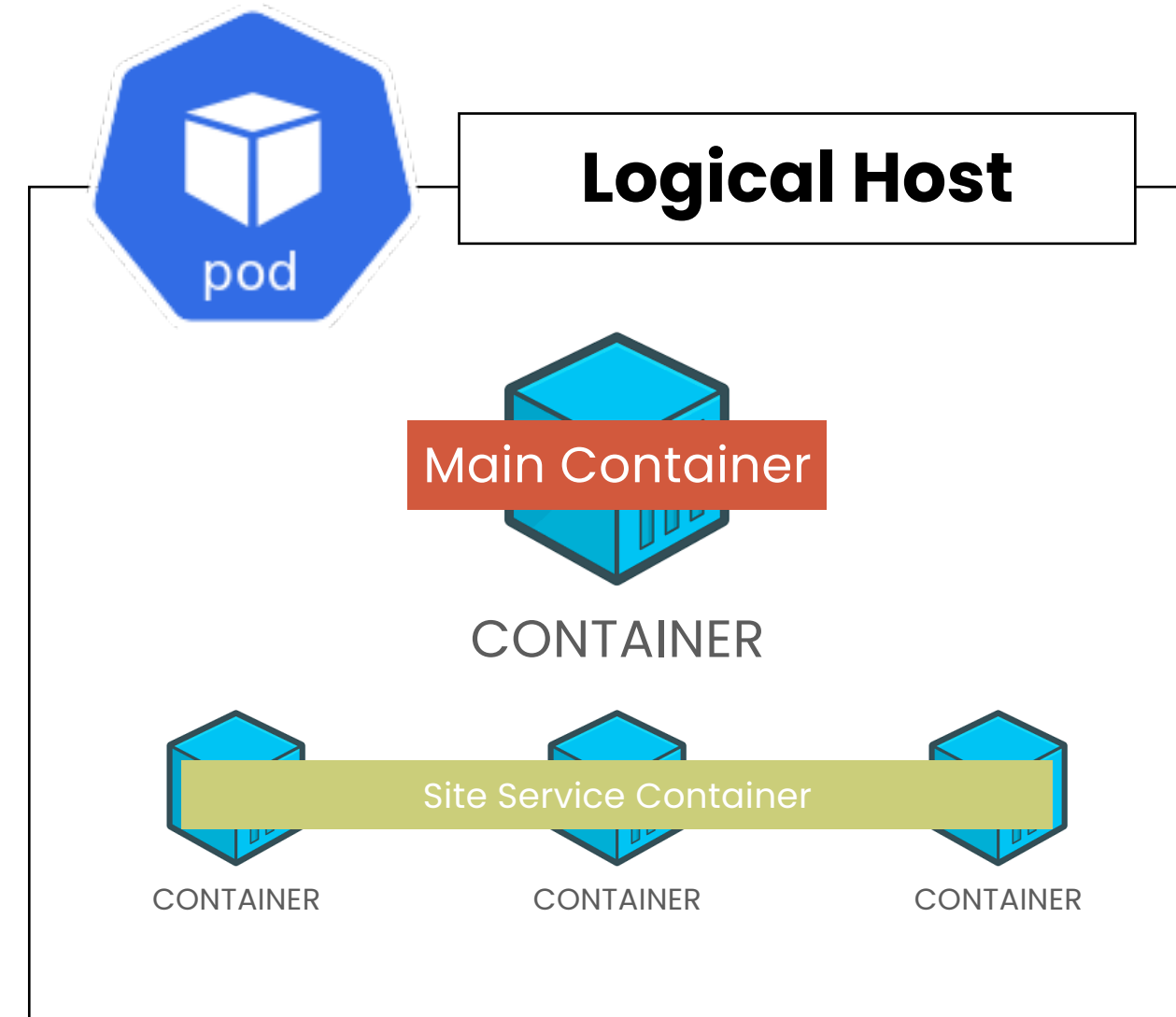
POD – STORAGE AND NETWORK



POD – STORAGE AND NETWORK



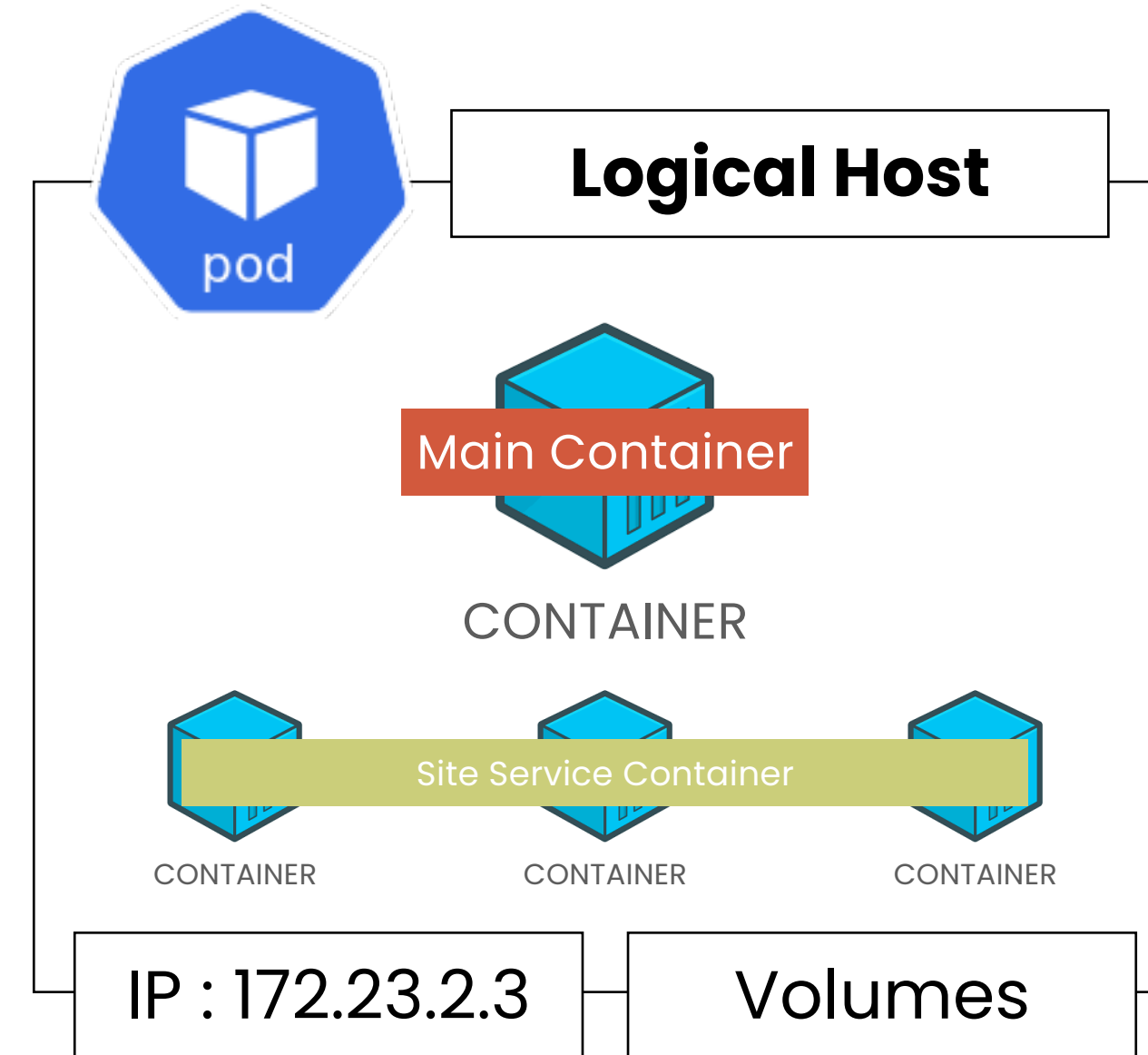
POD – STORAGE AND NETWORK



A Pod can specify **a set of shared storage volumes**. All containers in the Pod can access the shared volumes, allowing those containers to share data.

Each Pod **is assigned a unique IP address** for each address family. Every container in a Pod **shares** the network namespace, including the **IP address and network ports**. Inside a Pod (and **only** then), the containers that belong to the Pod can communicate with one another using **localhost**. When containers in a Pod communicate with entities *outside the Pod*, they must coordinate how they use the shared network resources (such as ports).

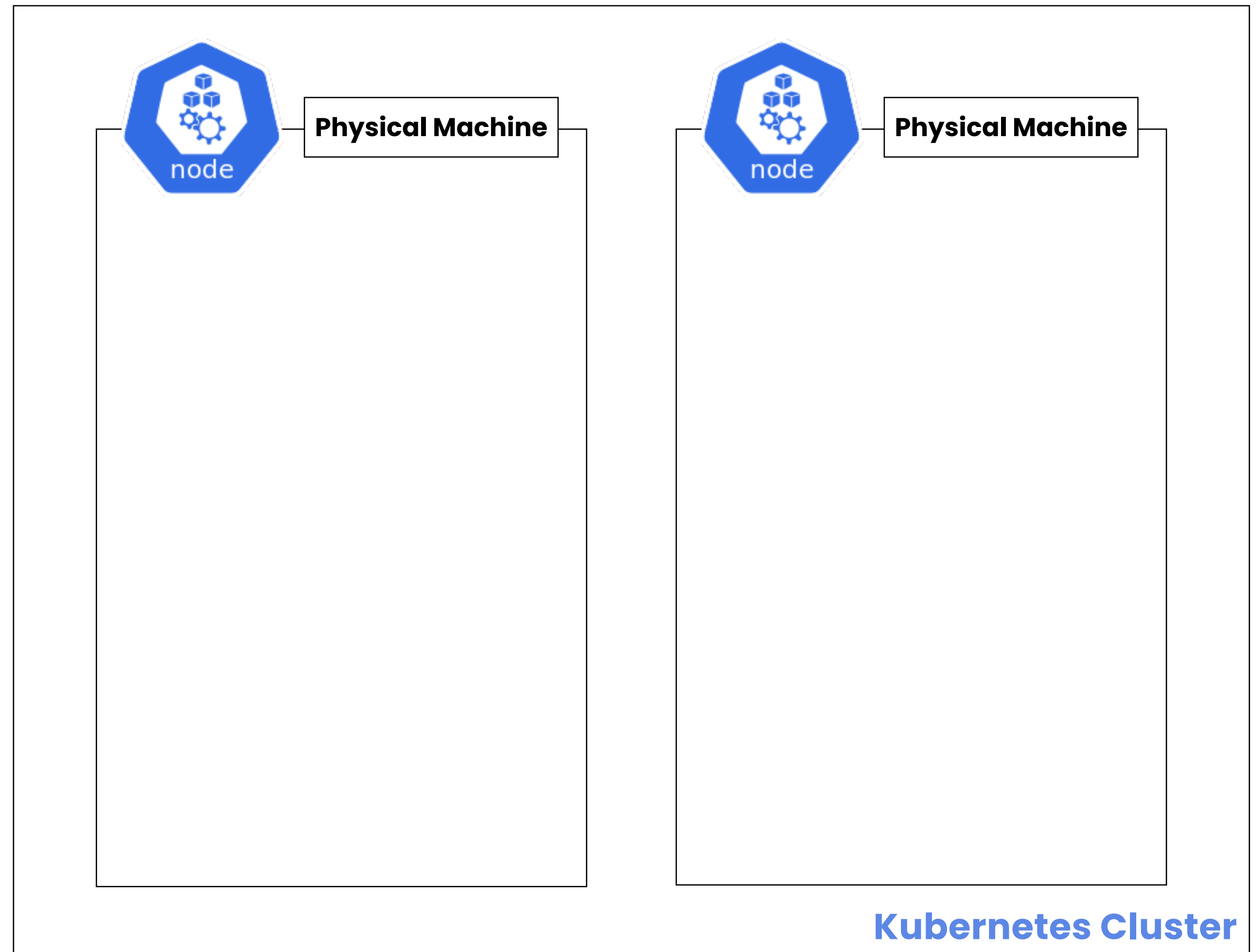
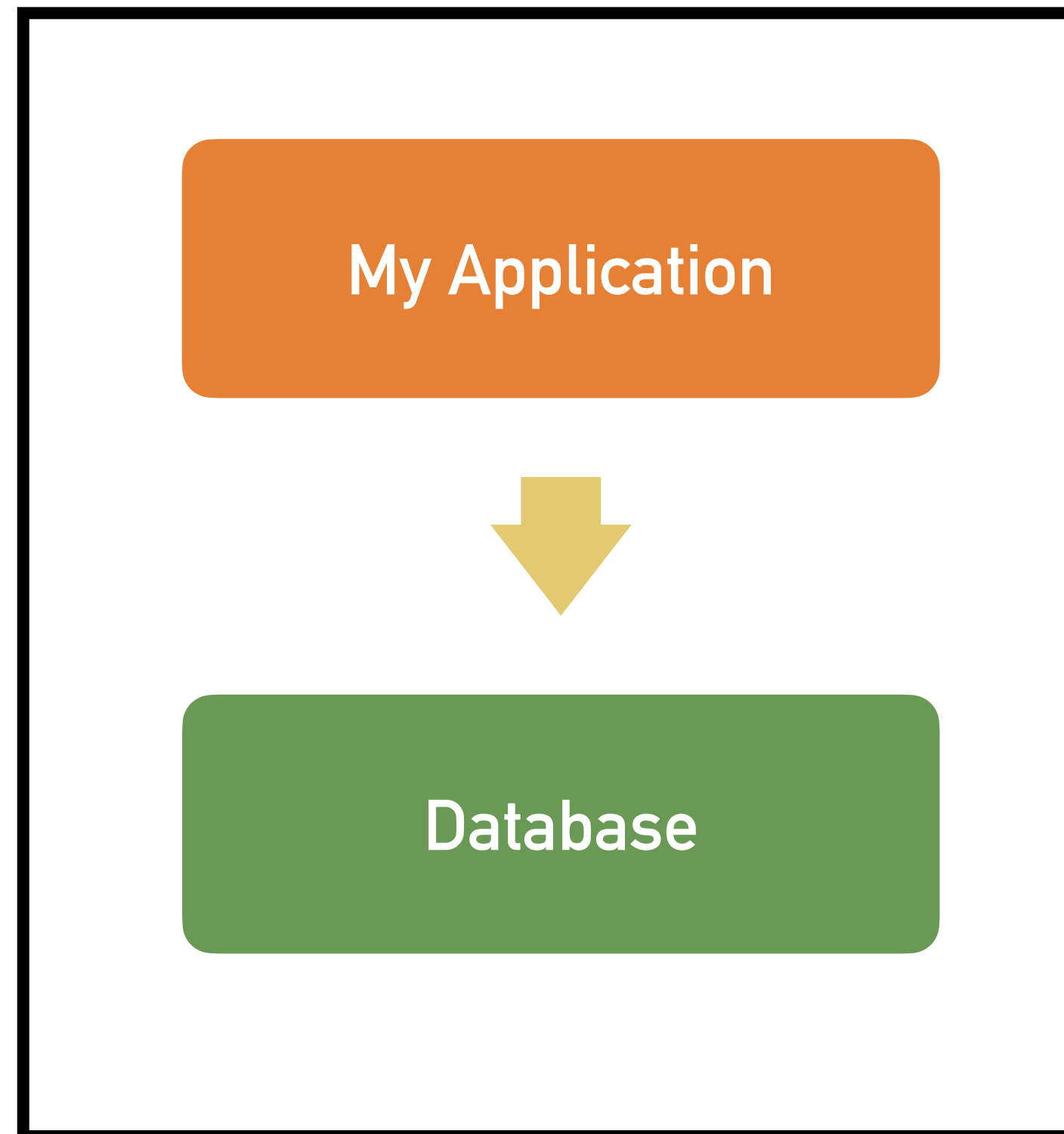
POD – STORAGE AND NETWORK



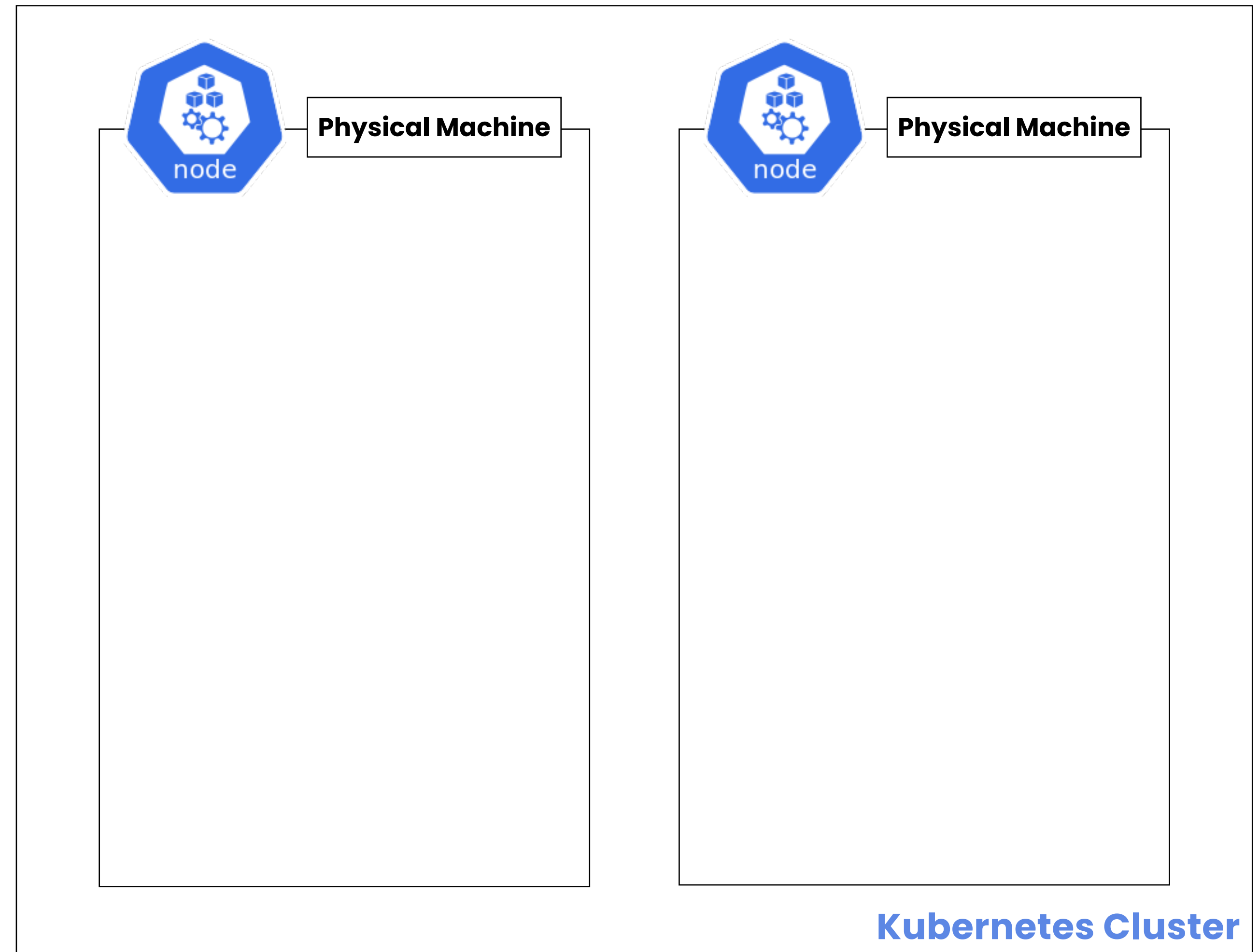
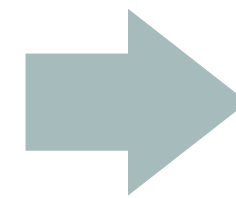
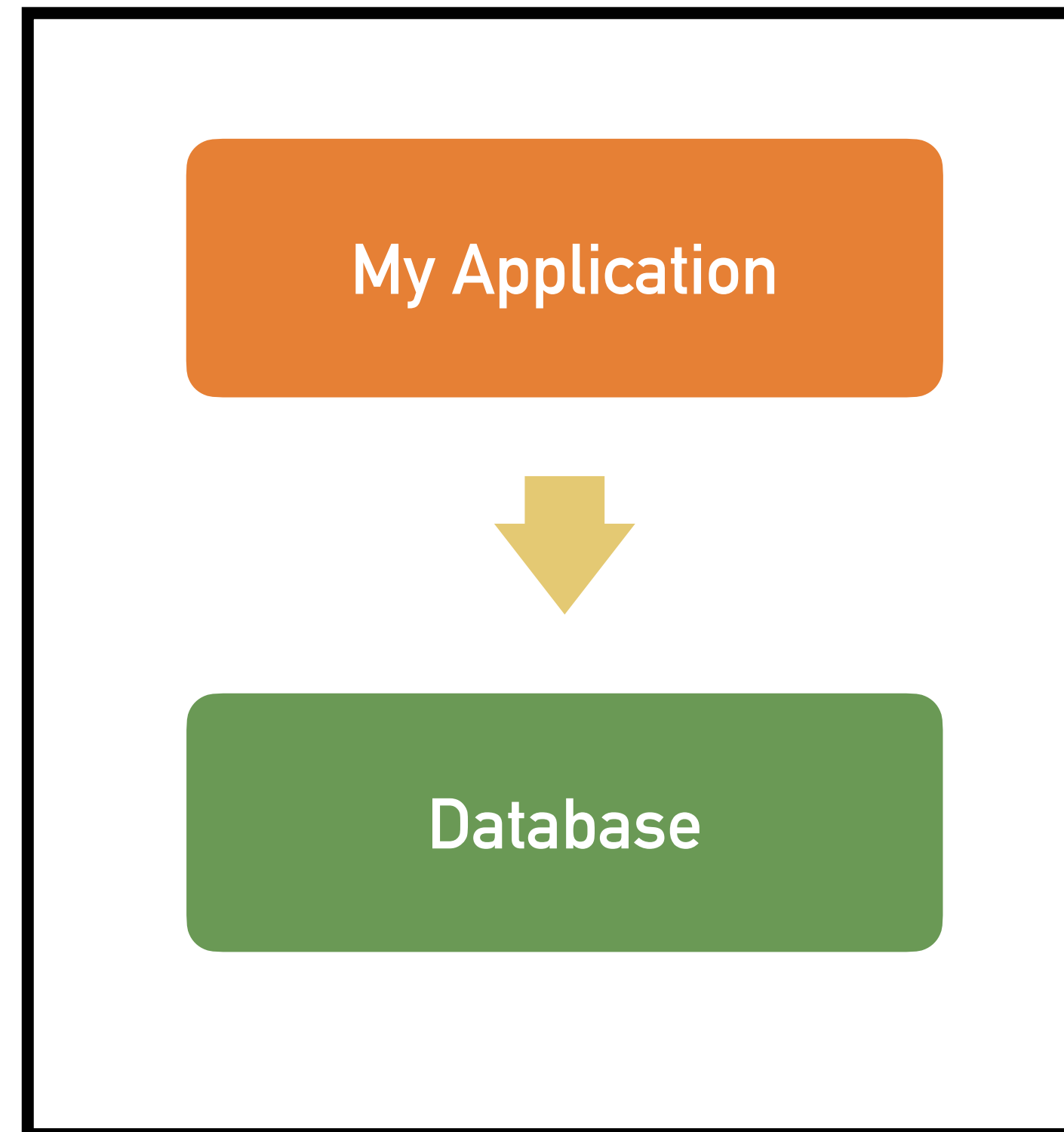
A Pod can specify **a set of shared storage volumes**. All containers in the Pod can access the shared volumes, allowing those containers to share data.

Each Pod **is assigned a unique IP address** for each address family. Every container in a Pod **shares** the network namespace, including the **IP address and network ports**. Inside a Pod (and **only** then), the containers that belong to the Pod can communicate with one another using **localhost**. When containers in a Pod communicate with entities *outside the Pod*, they must coordinate how they use the shared network resources (such as ports).

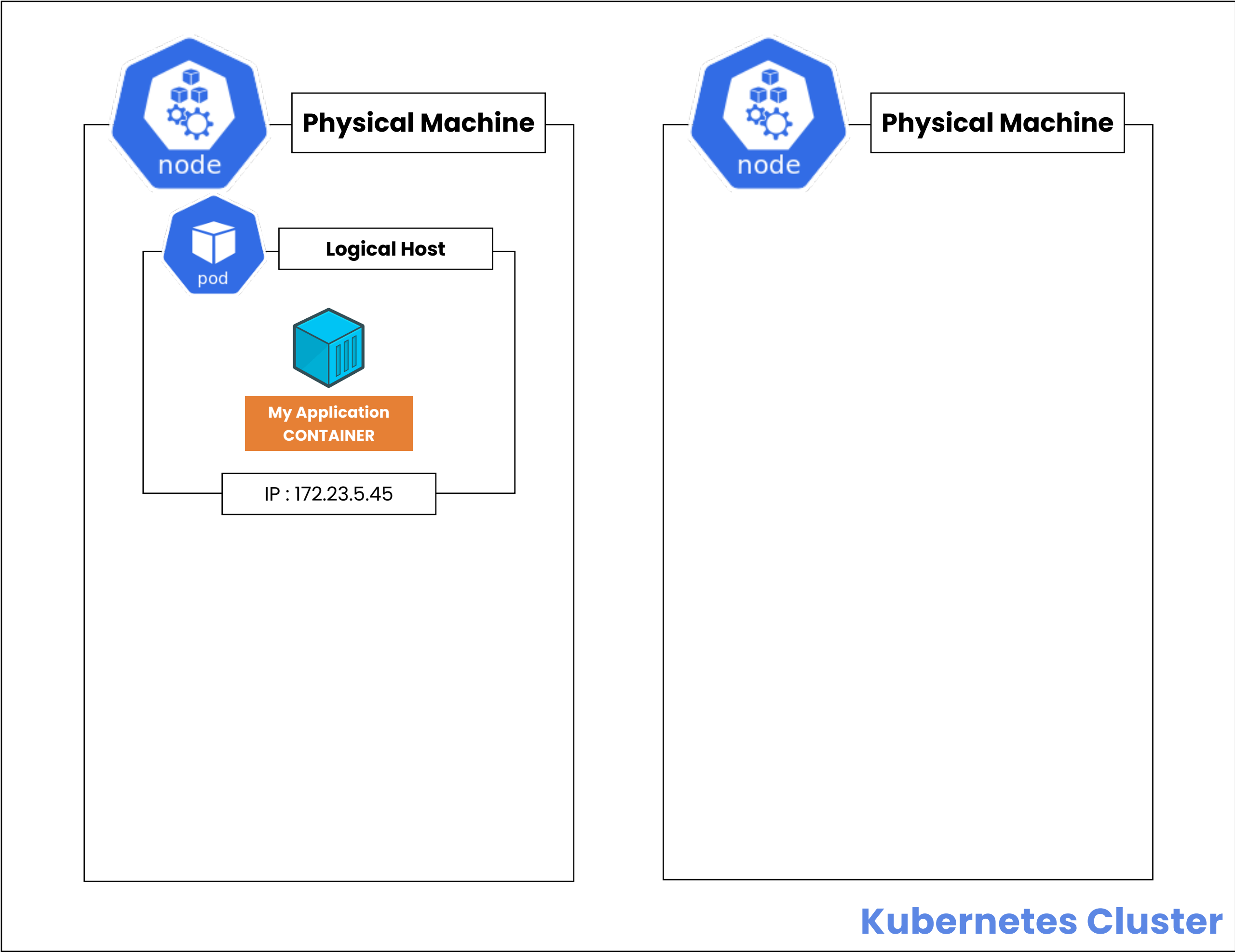
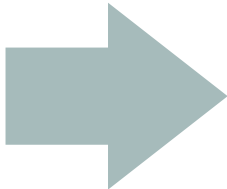
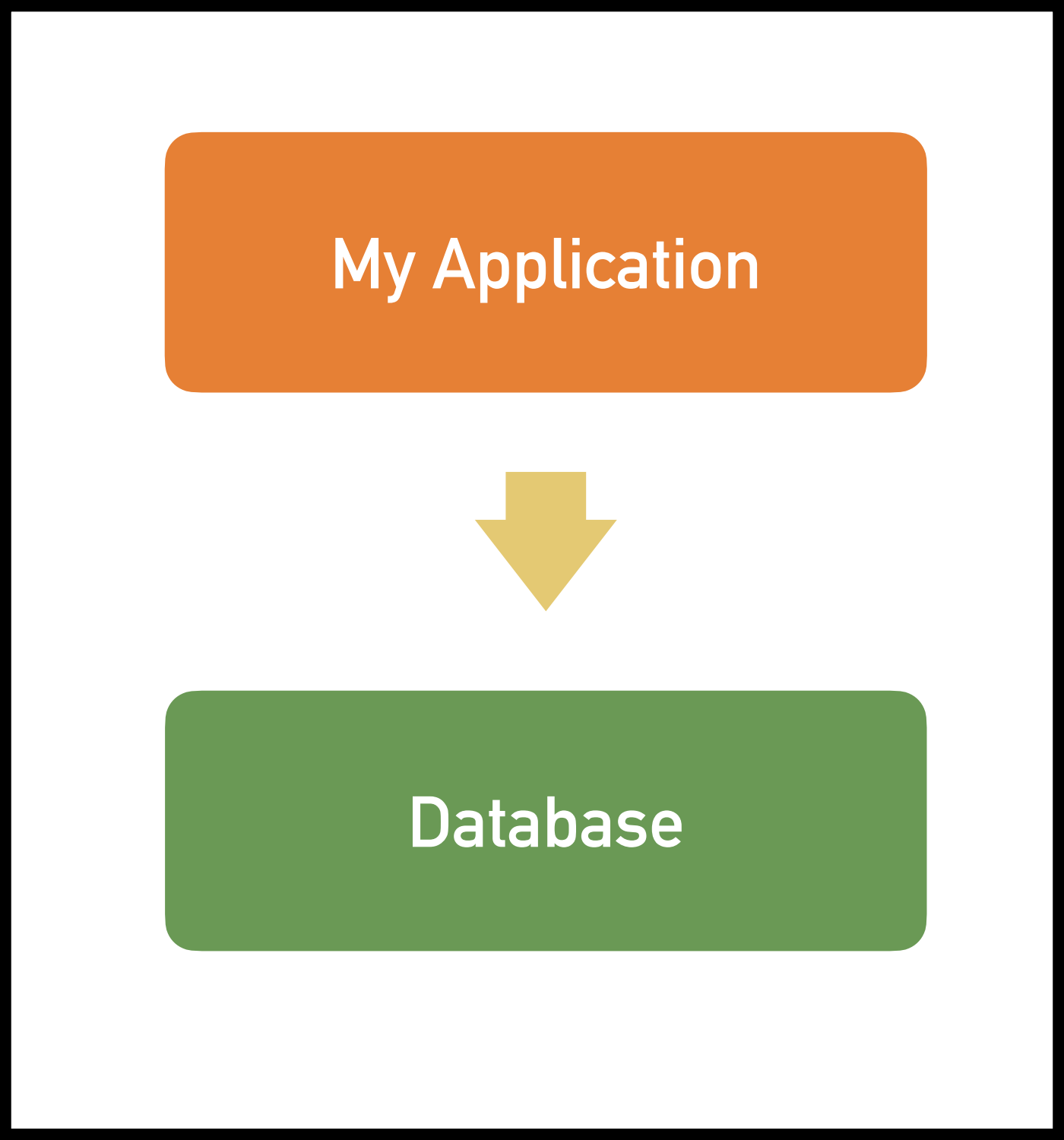
POD FOR SIMPLE APPLICATION



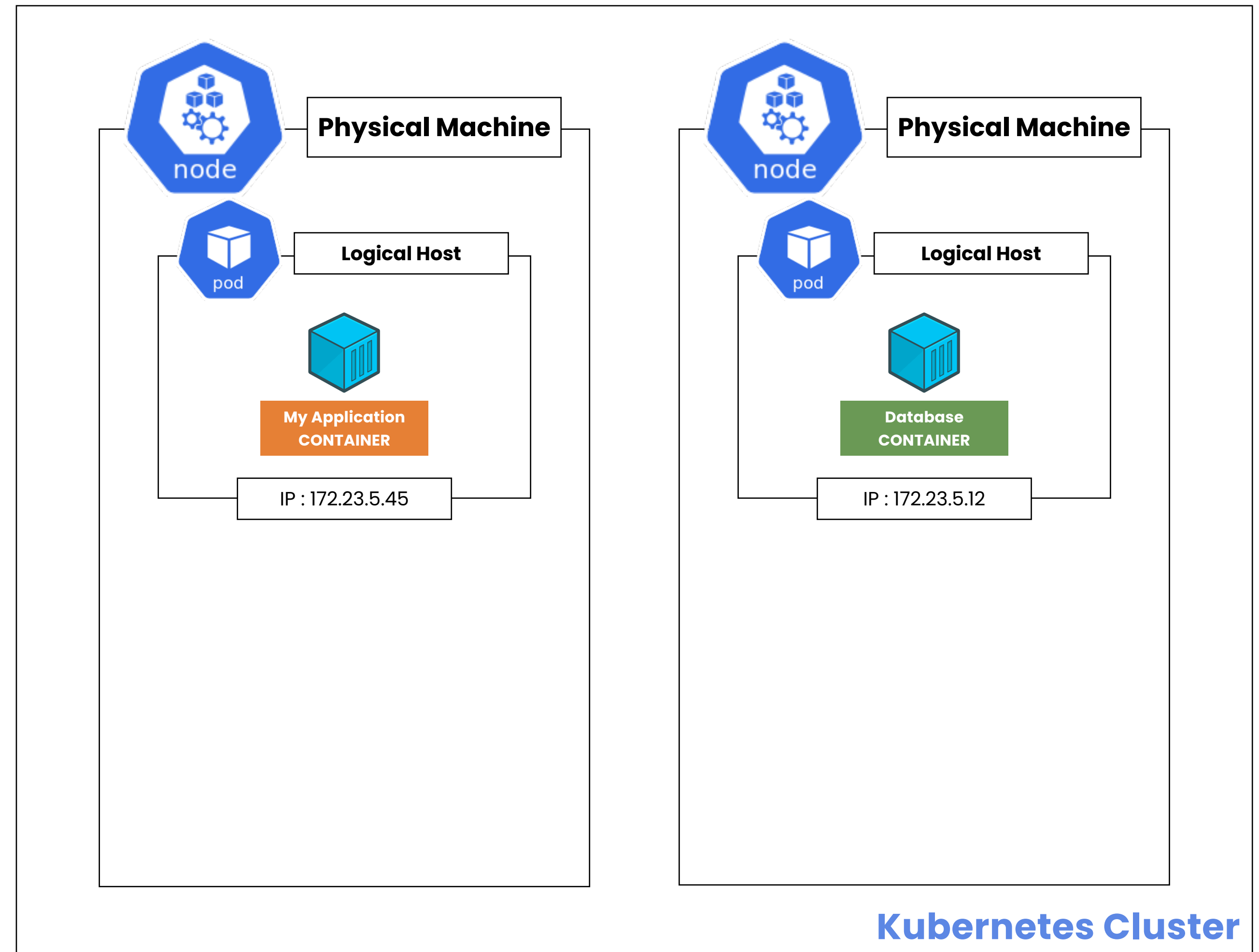
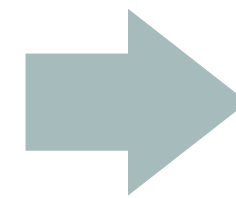
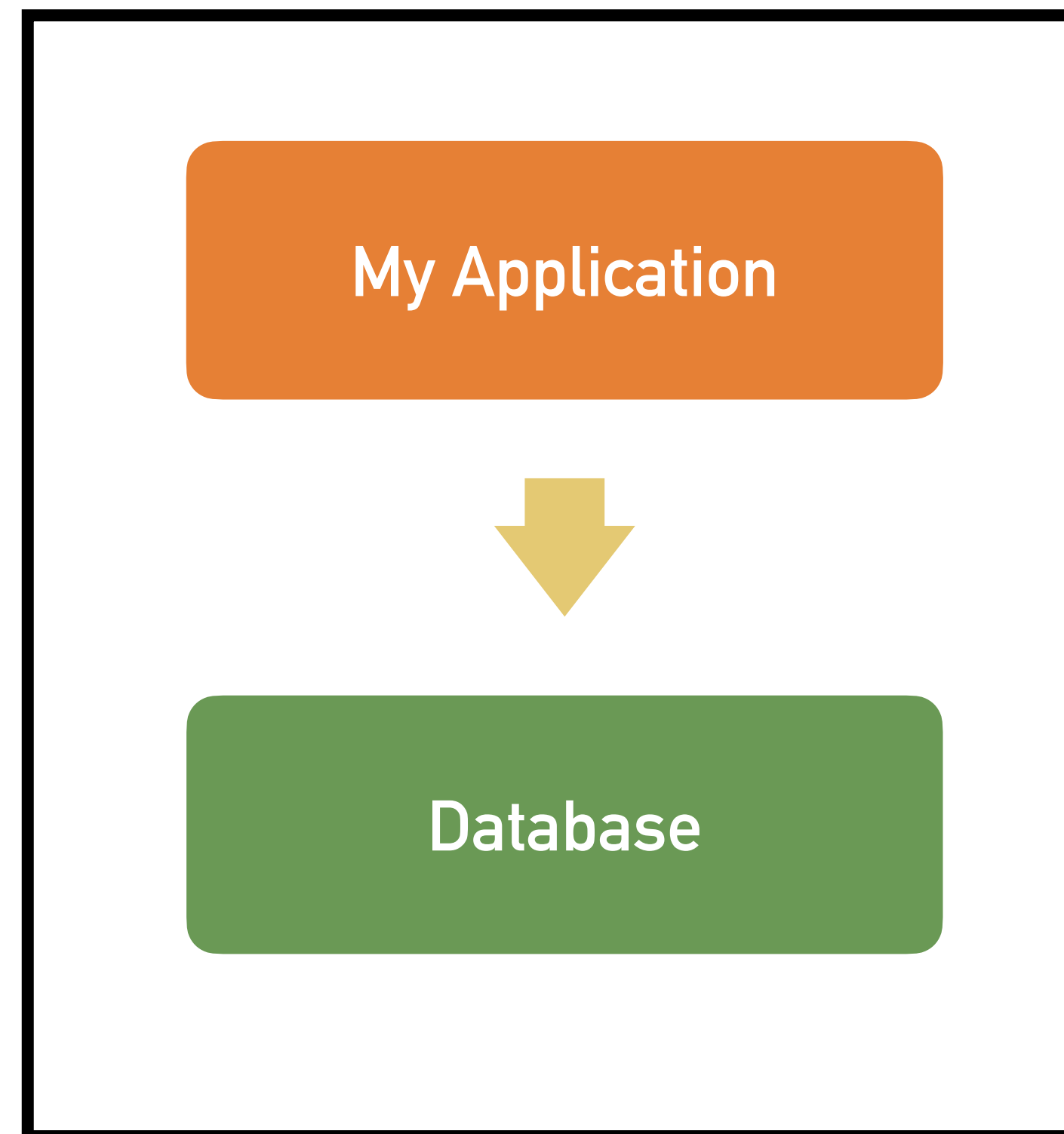
POD FOR SIMPLE APPLICATION



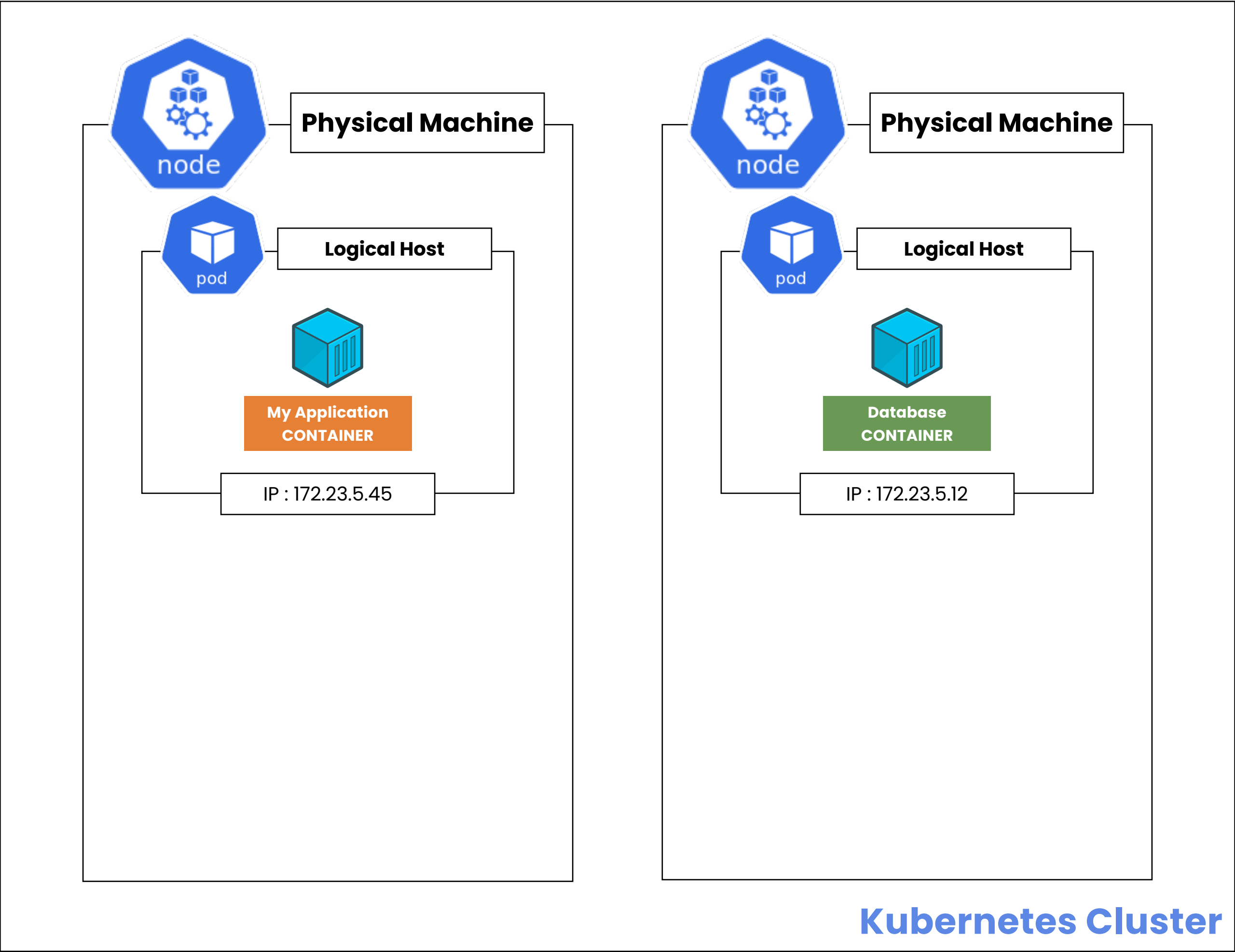
POD FOR SIMPLE APPLICATION



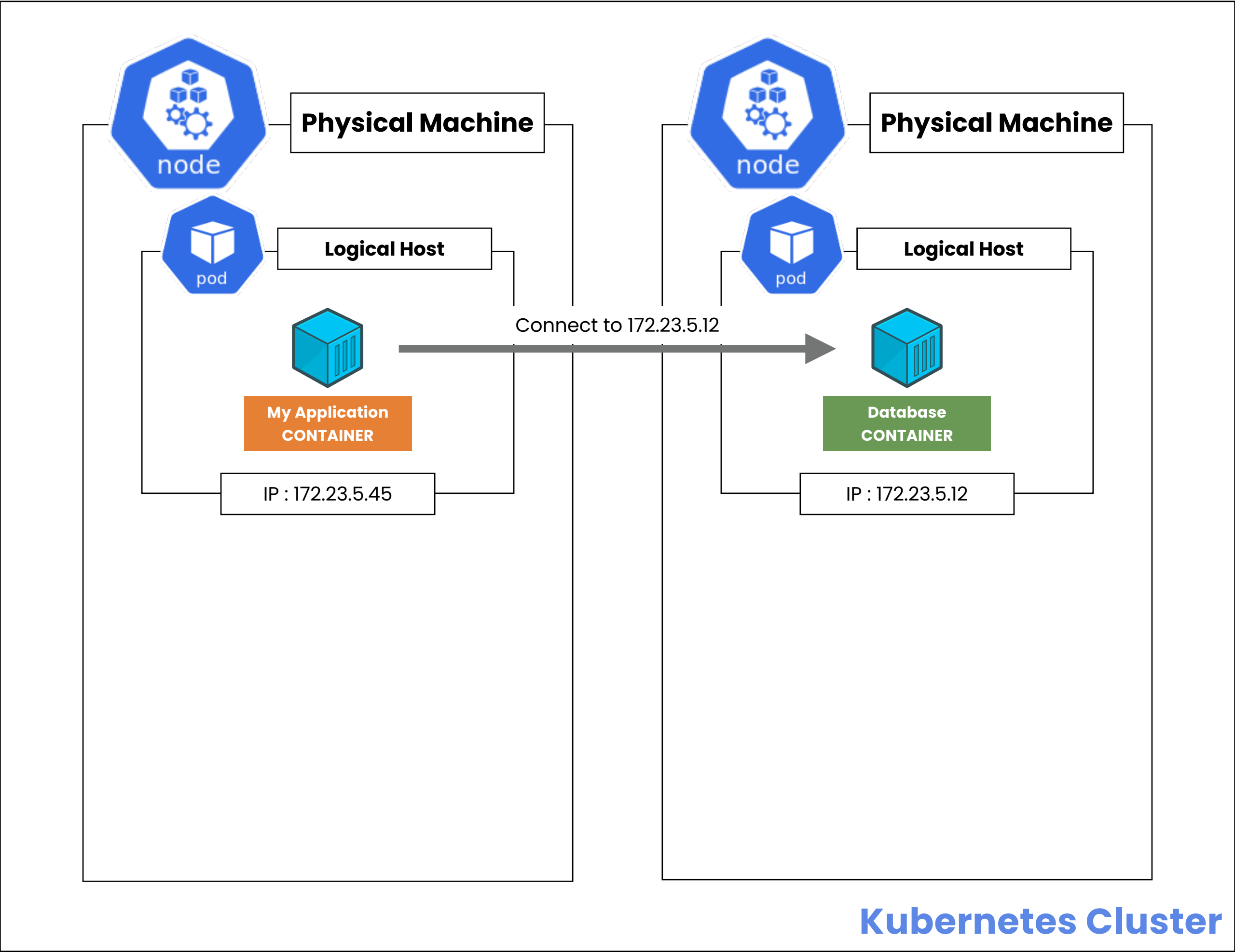
POD FOR SIMPLE APPLICATION



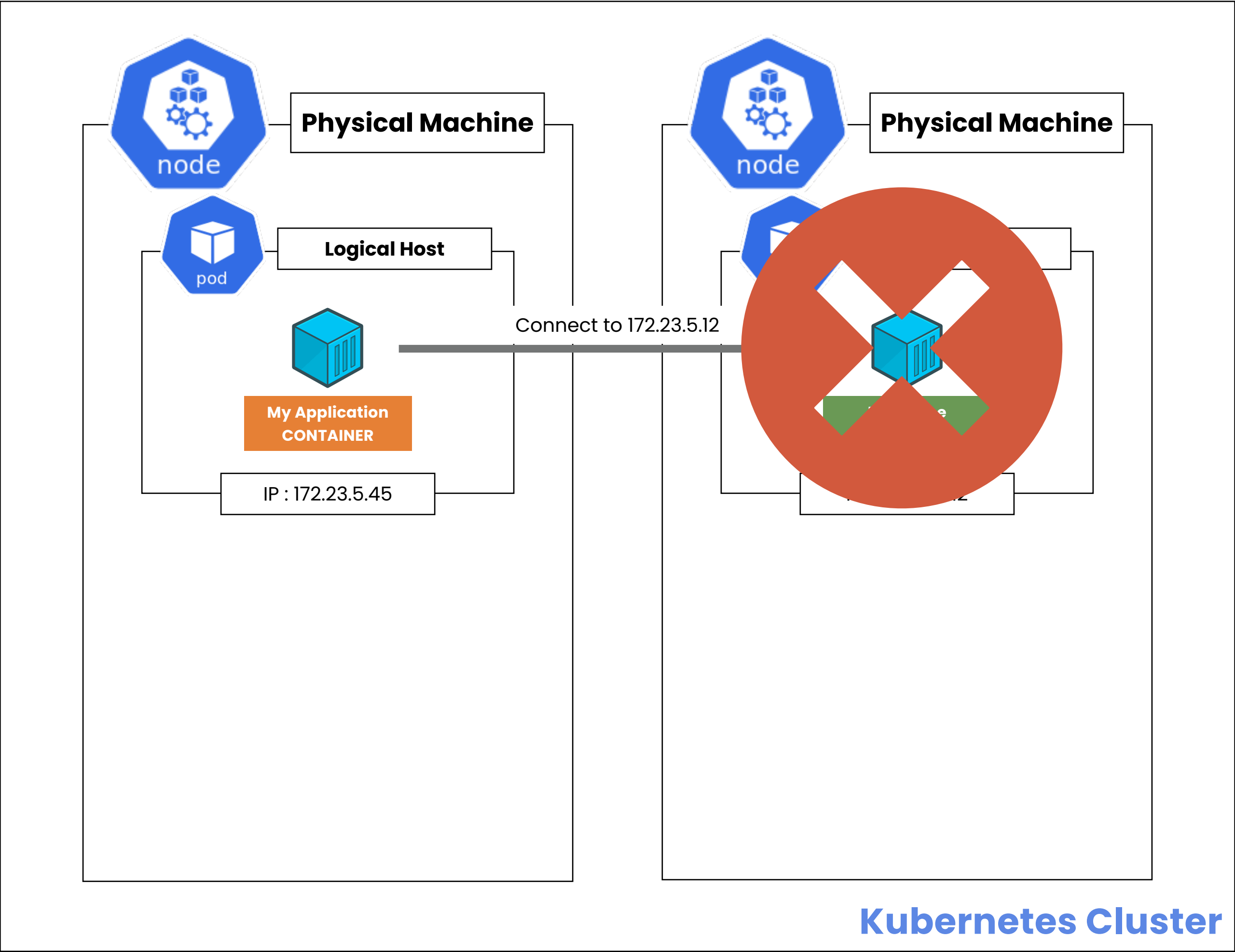
NEW IP ON RE-CREATE



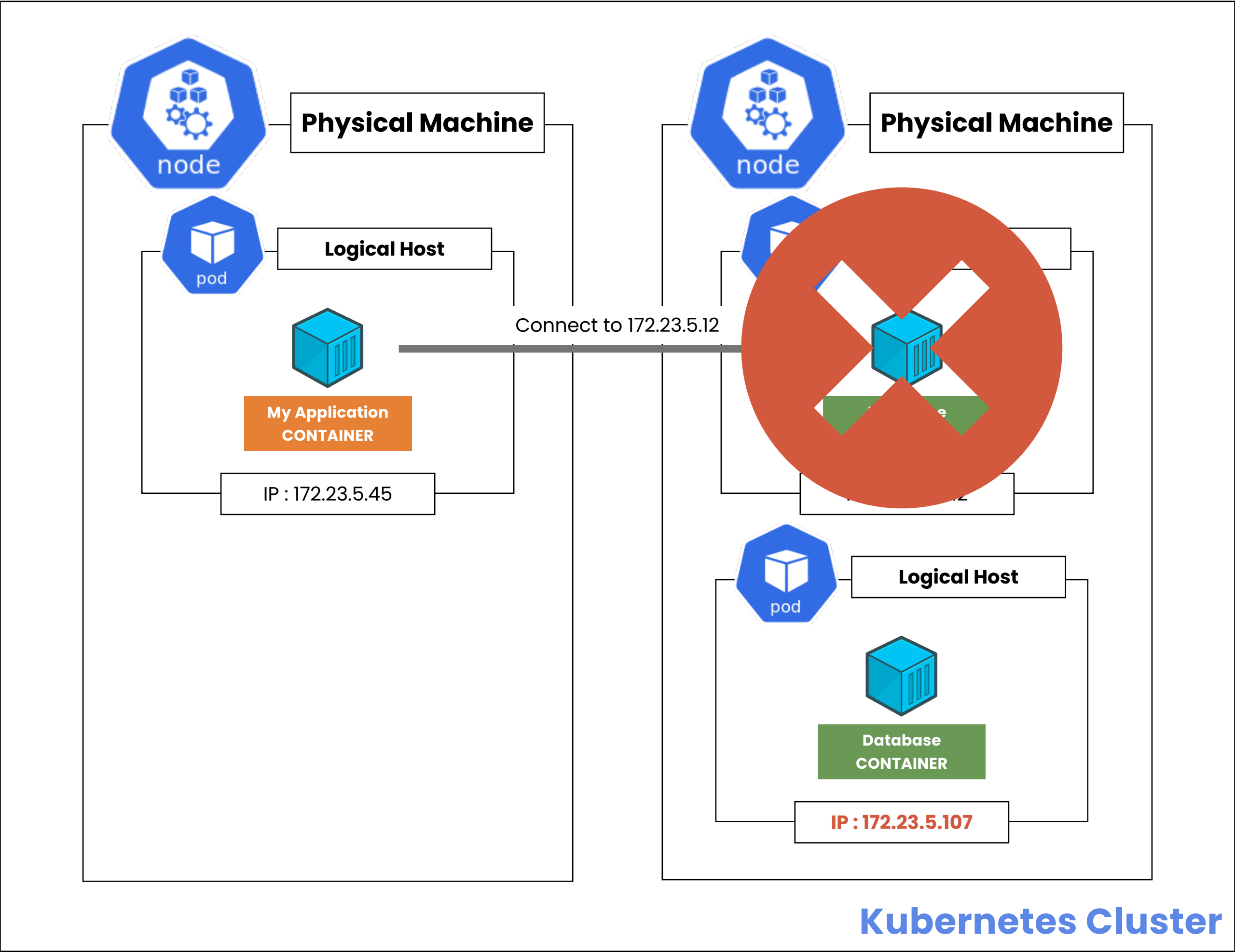
NEW IP ON RE-CREATE



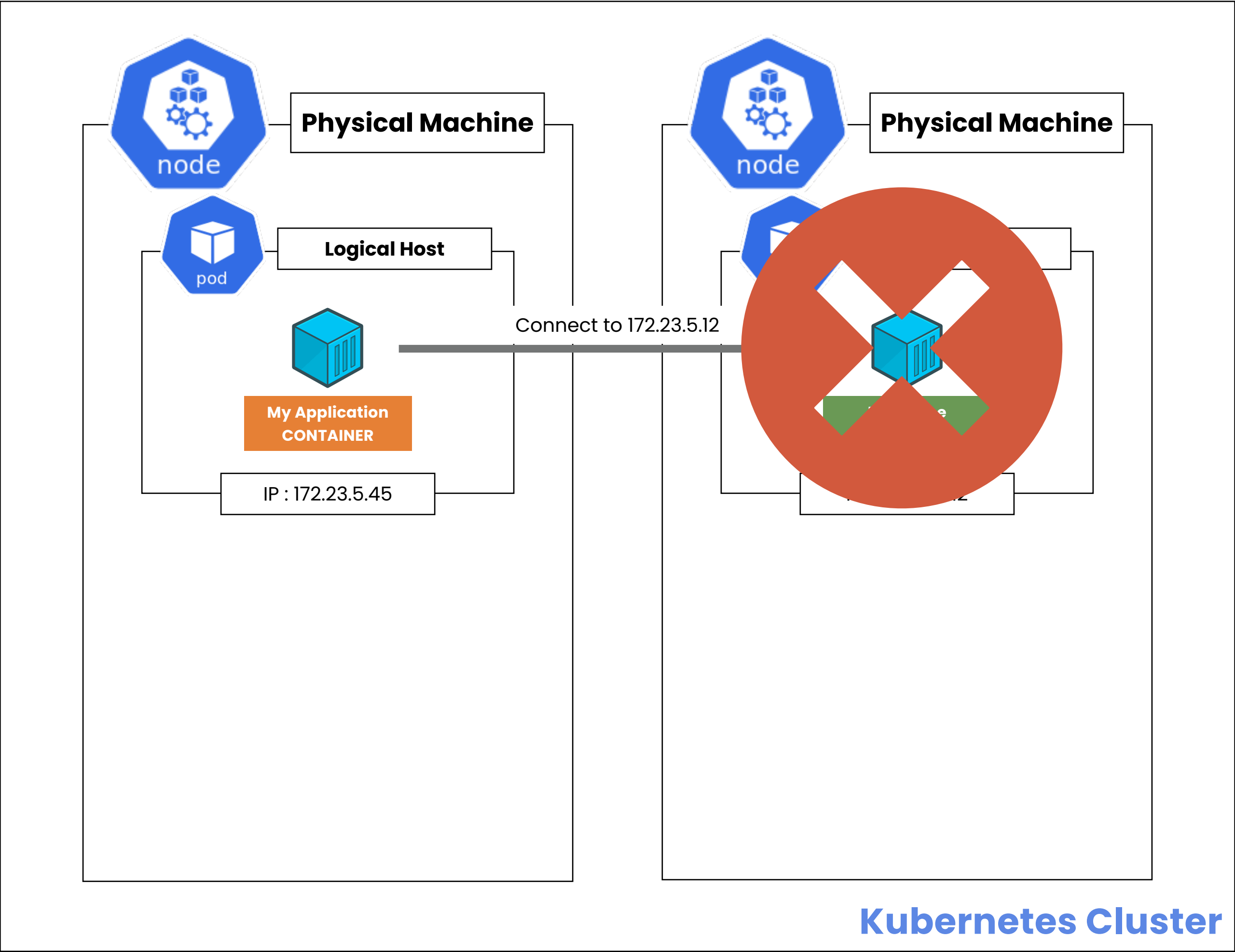
NEW IP ON RE-CREATE



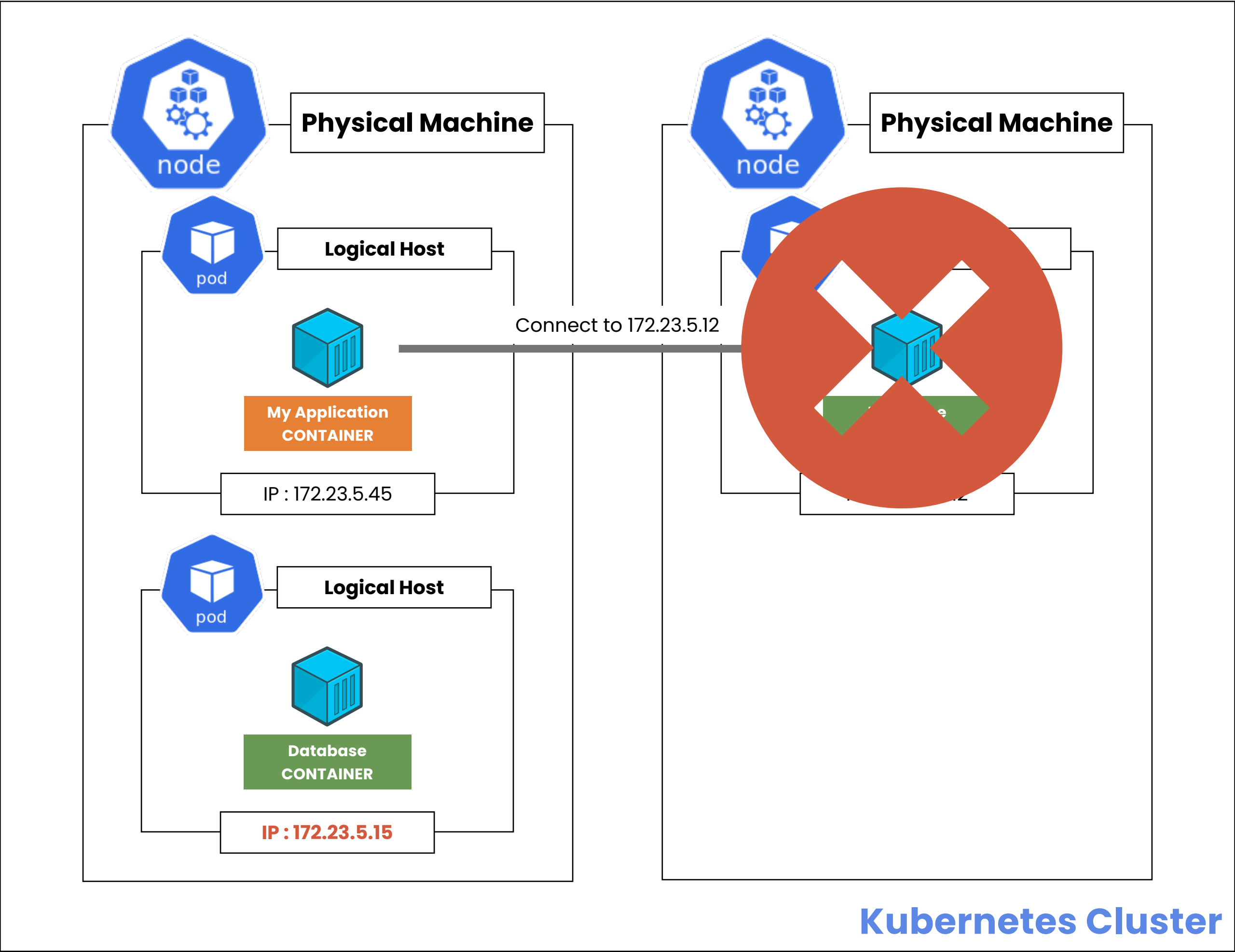
NEW IP ON RE-CREATE



NEW IP ON RE-CREATE



NEW IP ON RE-CREATE



SERVICE & INGRESS

.....

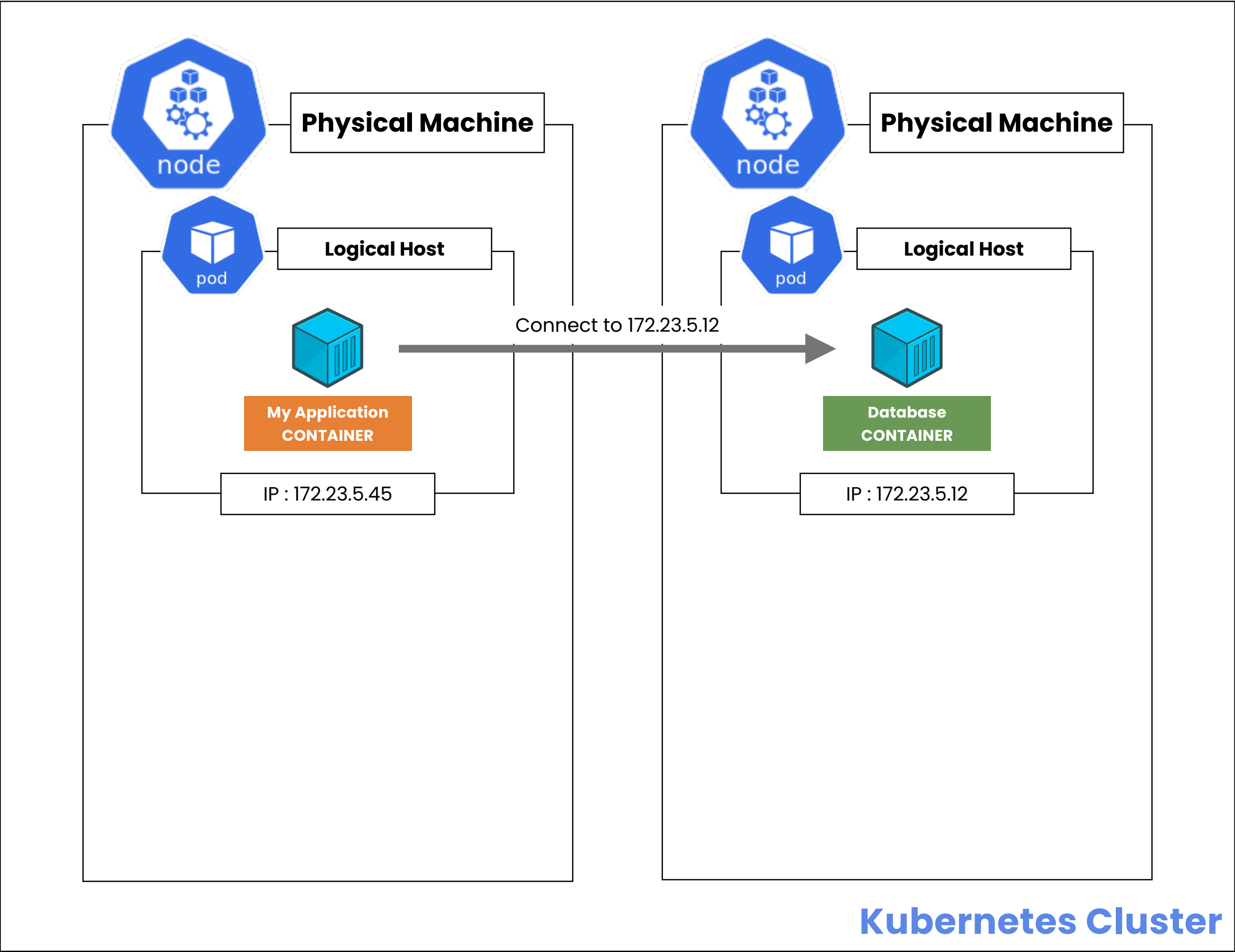
WHAT'S SERVICE

An abstract way to **expose** an application running on a **set of Pods as a network service**.

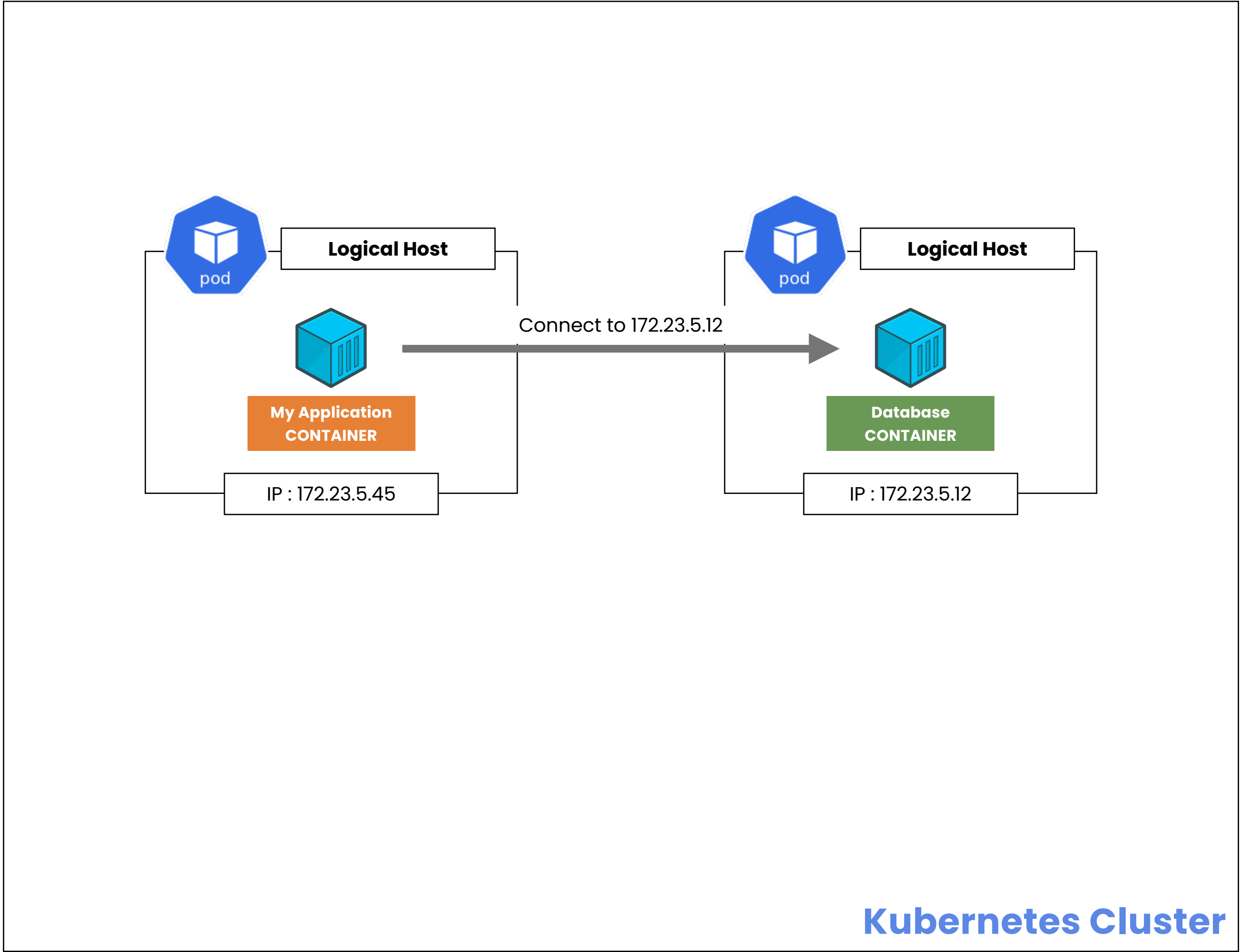
With Kubernetes you don't need to modify your application to use an unfamiliar service discovery mechanism. Kubernetes gives Pods their own IP addresses and a single DNS name for a set of Pods, and can load-balance across them.

a Service is an abstraction which **defines a logical set of Pods** and a policy by which to access them. The set of Pods targeted by a Service is usually determined by a selector. To learn about other ways to define Service endpoints, see [Services without selectors](#).

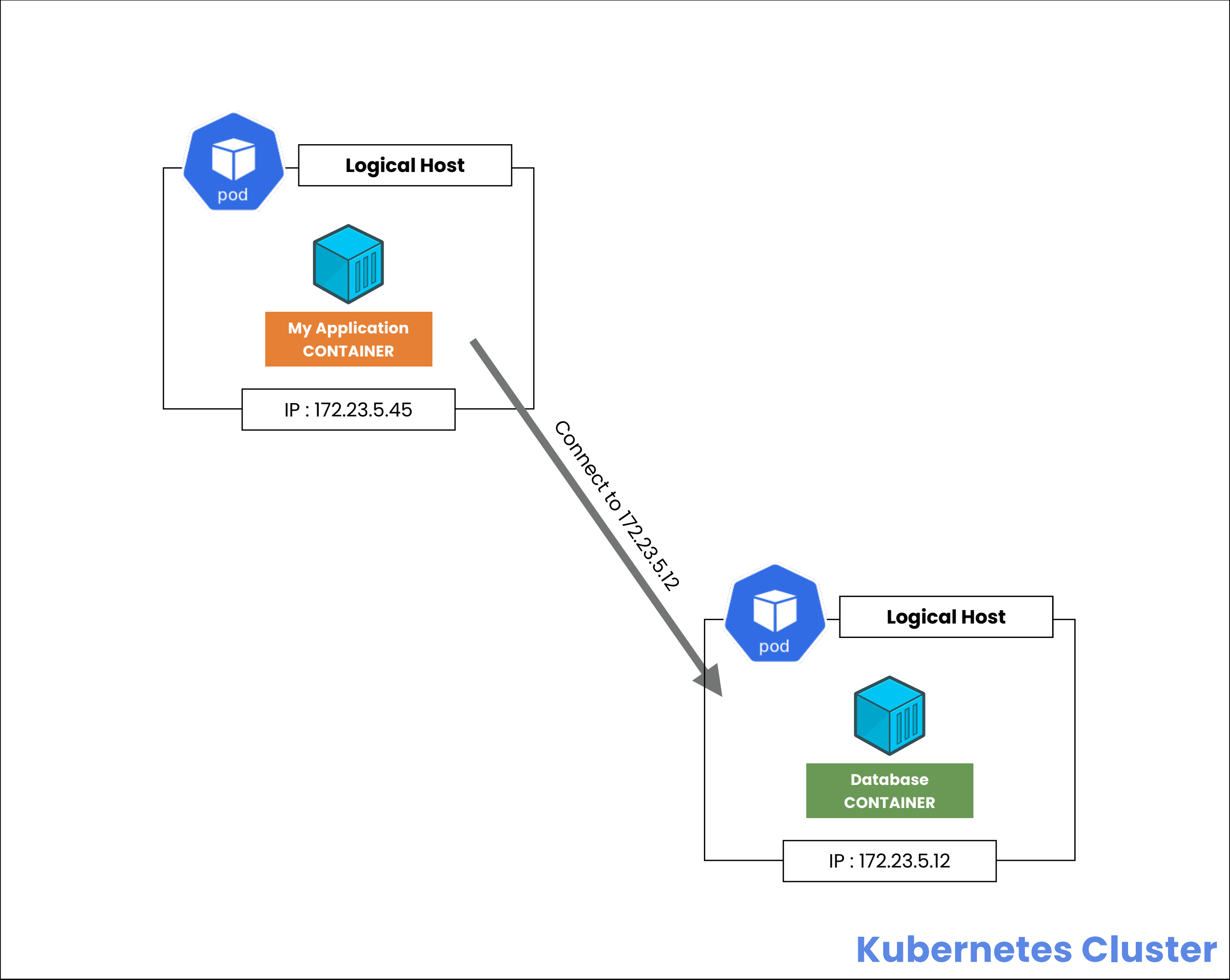
PHYSICAL VS LOGICAL



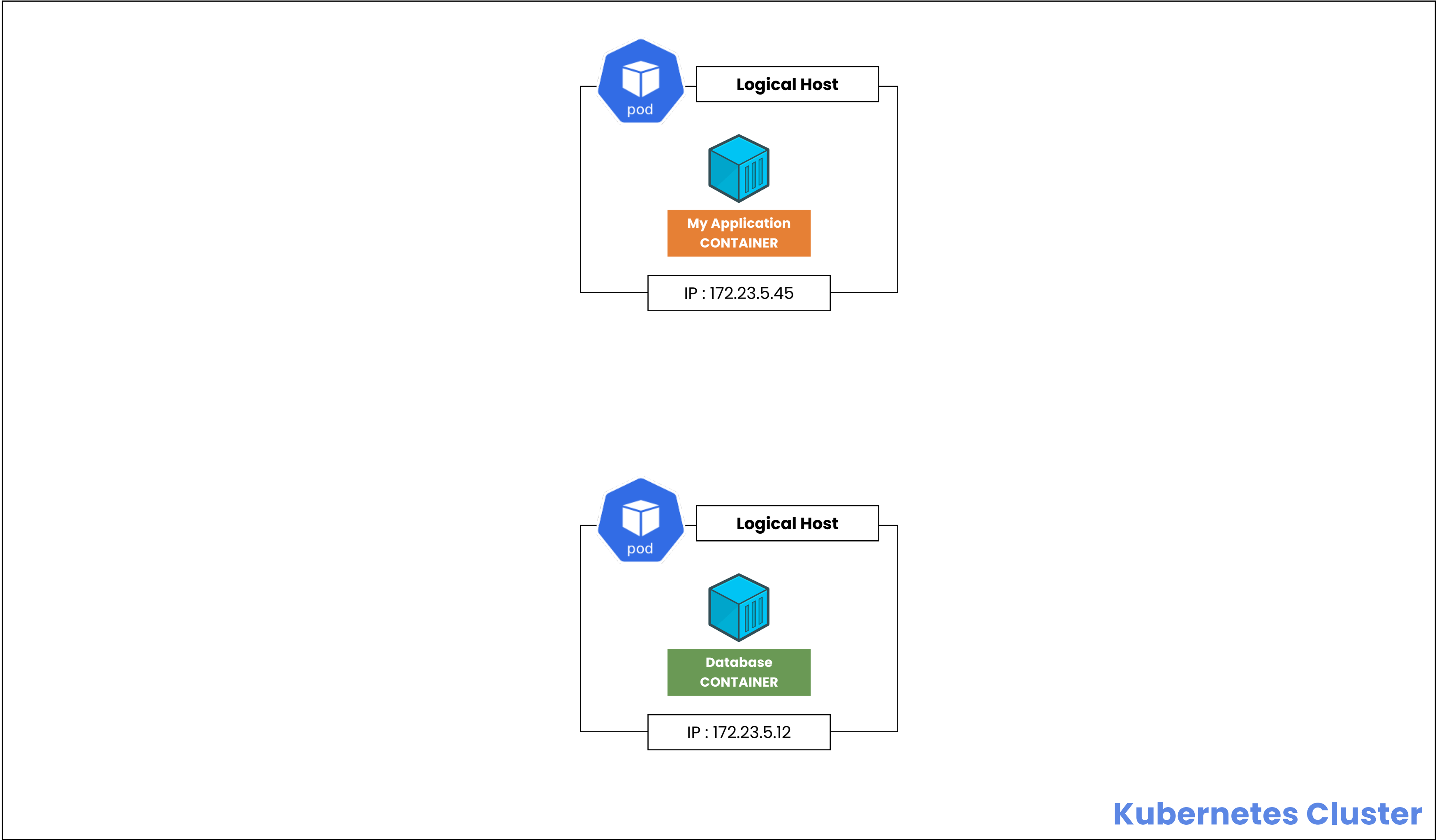
PHYSICAL VS LOGICAL



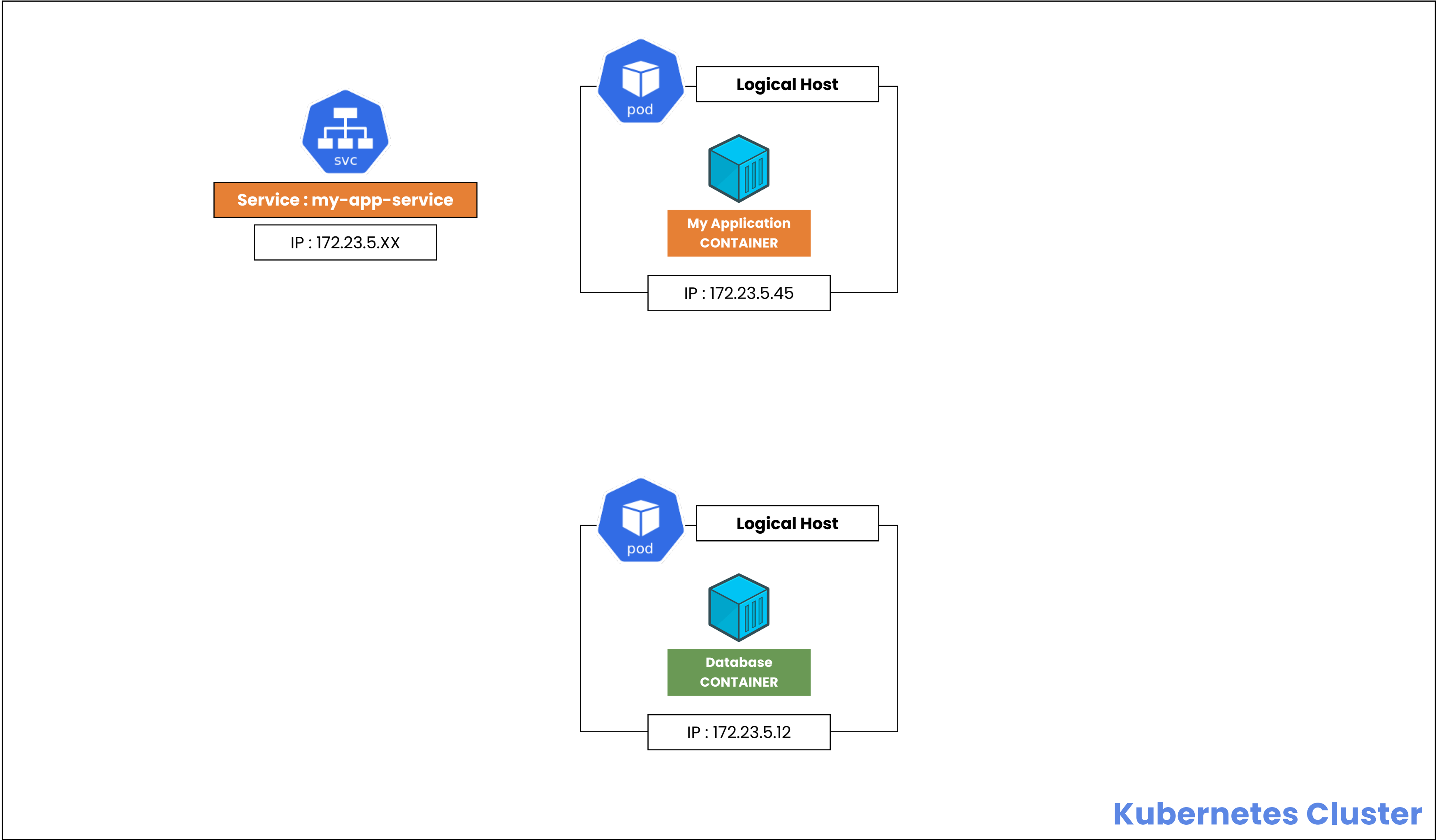
WITHOUT PHYSICAL VIEWS



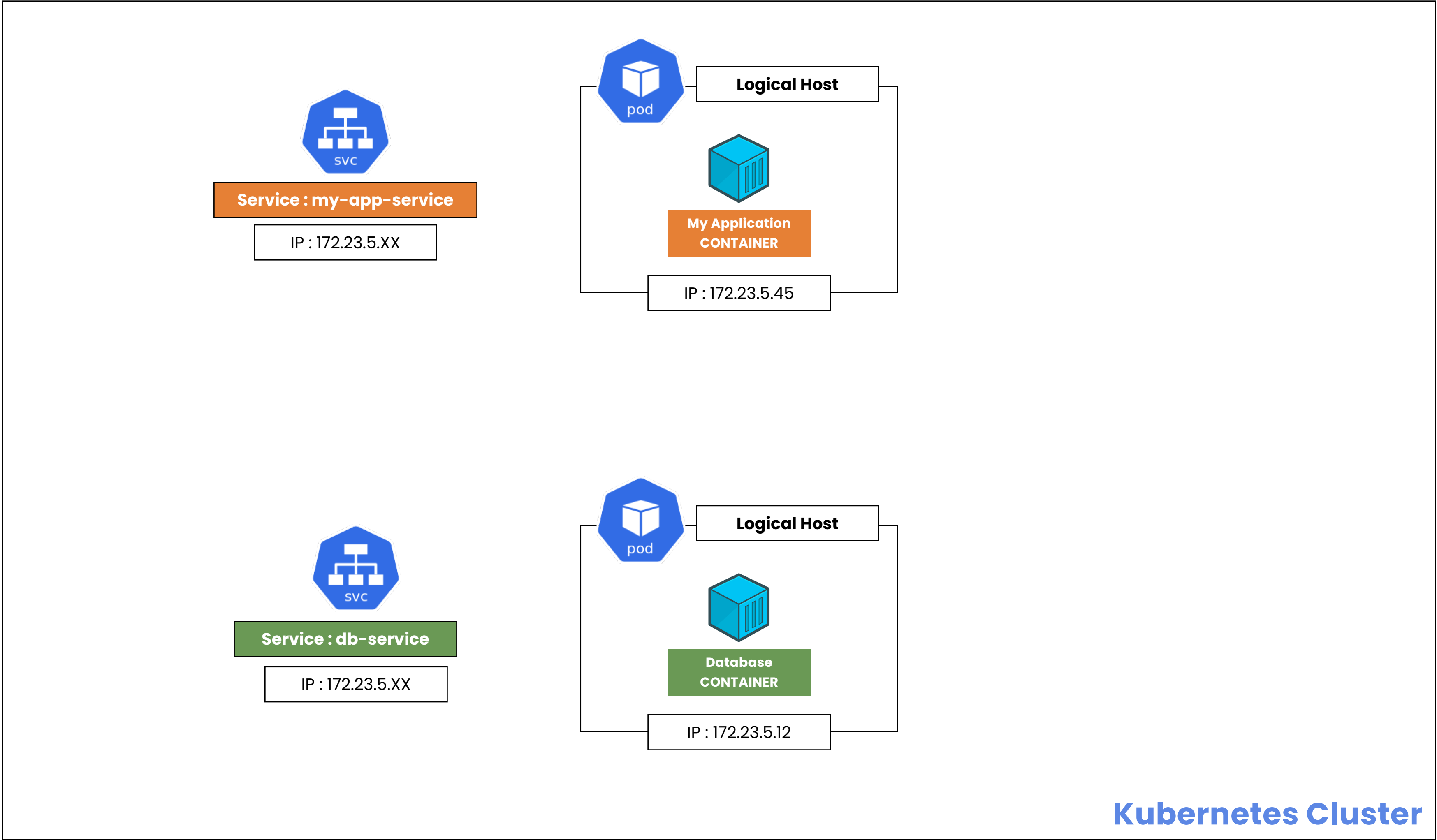
LOGICAL VIEWS WITH NODE AND SERVICE



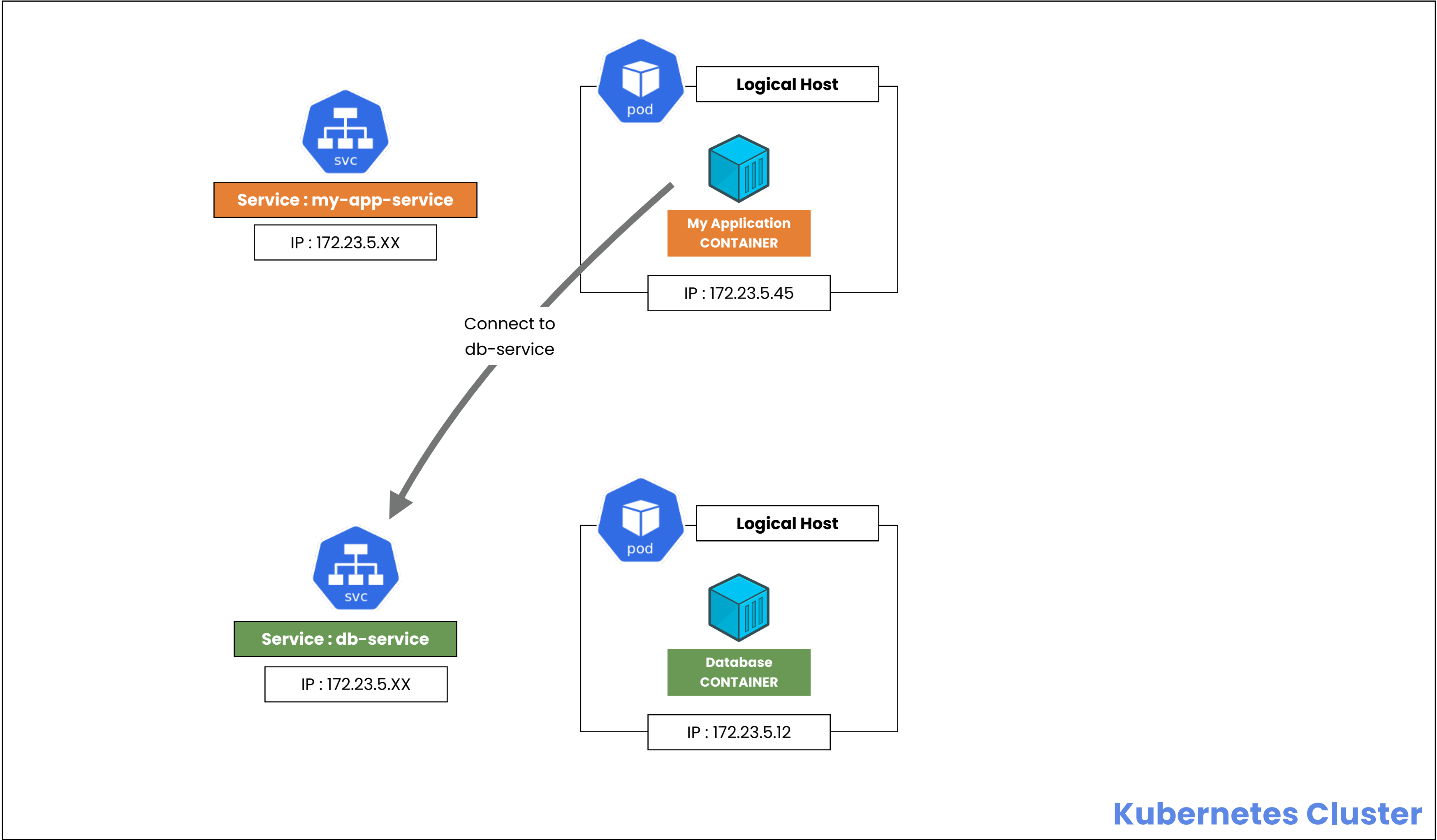
LOGICAL VIEWS WITH NODE AND SERVICE



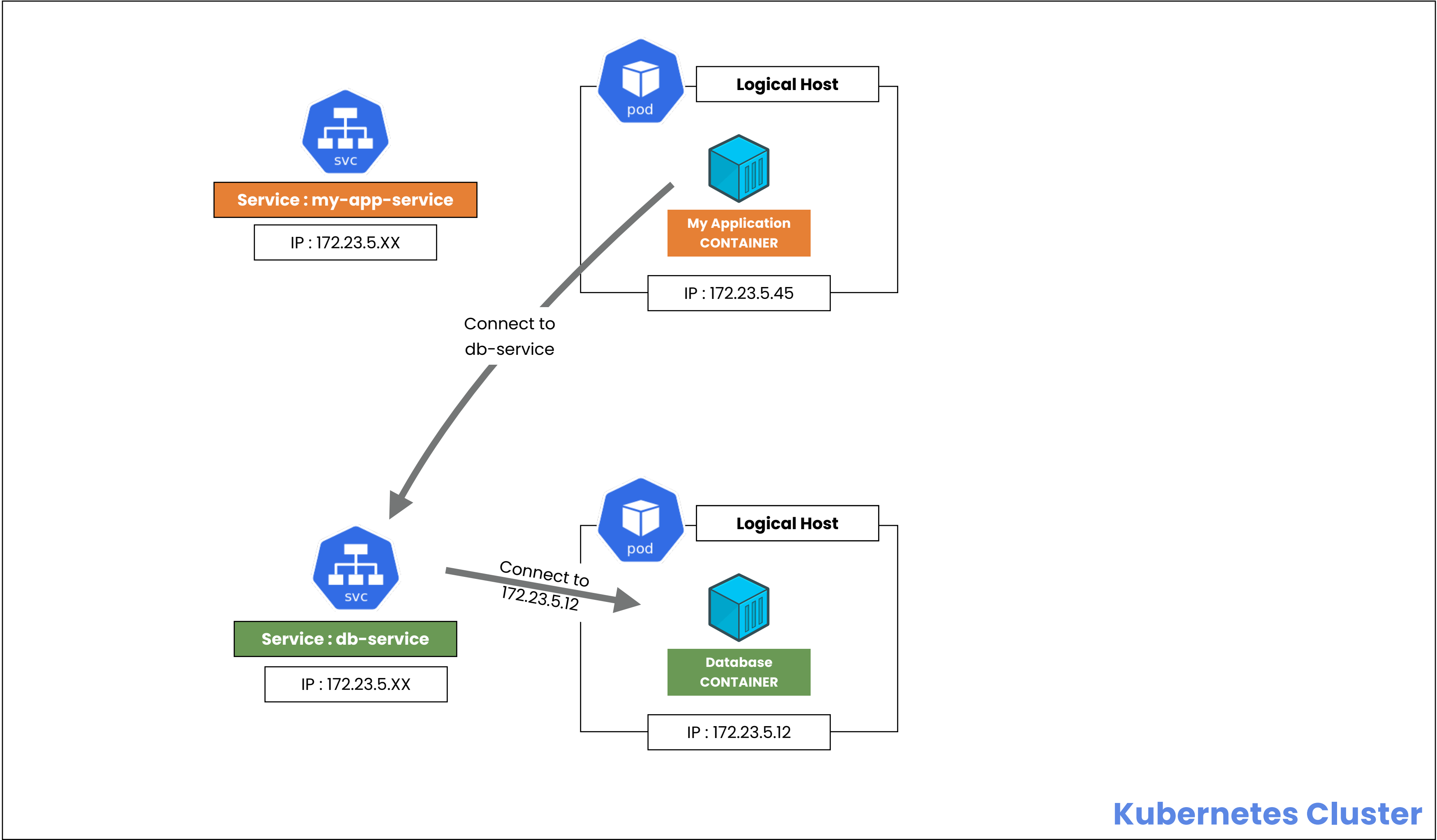
LOGICAL VIEWS WITH NODE AND SERVICE



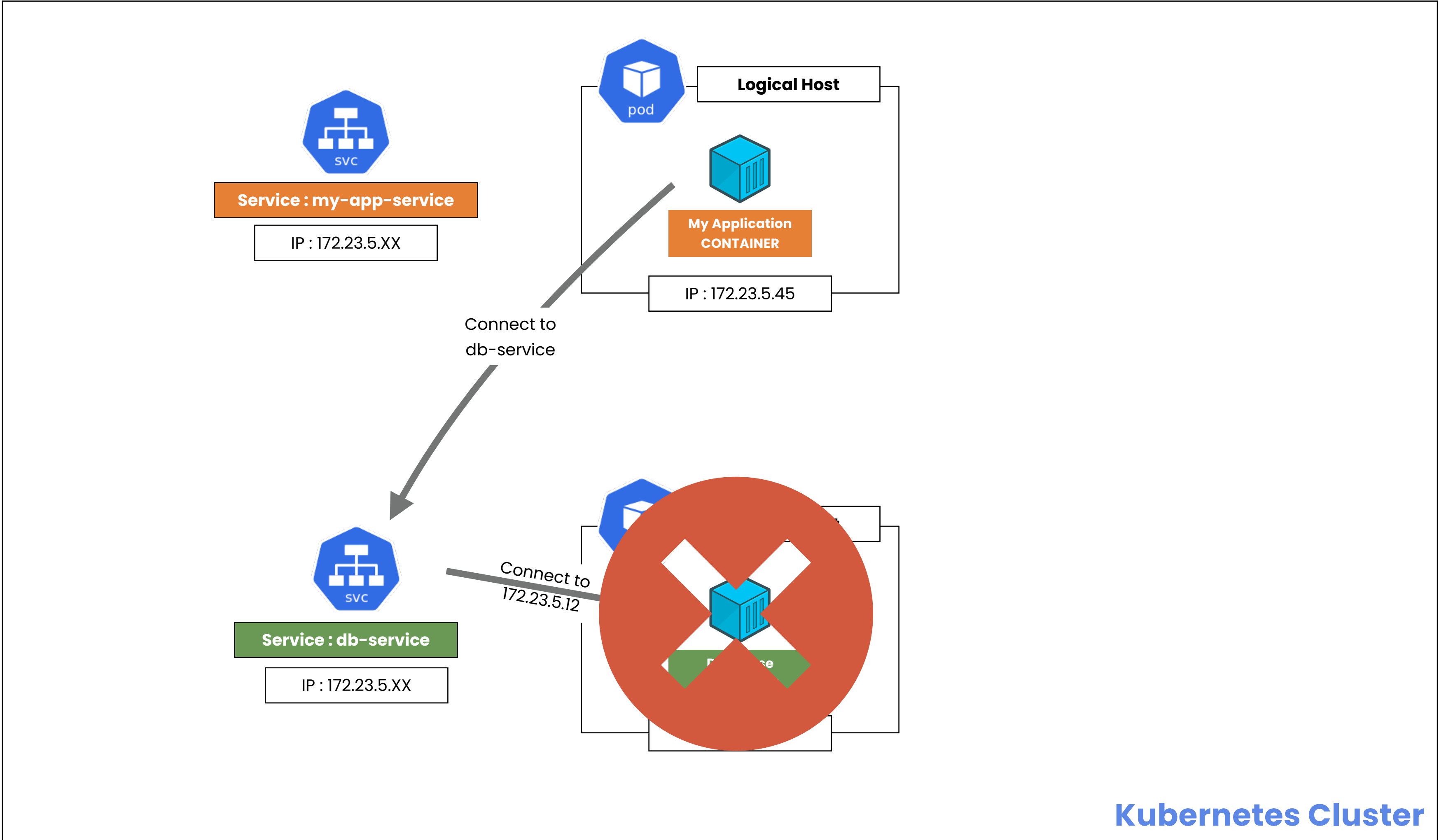
LOGICAL VIEWS WITH NODE AND SERVICE



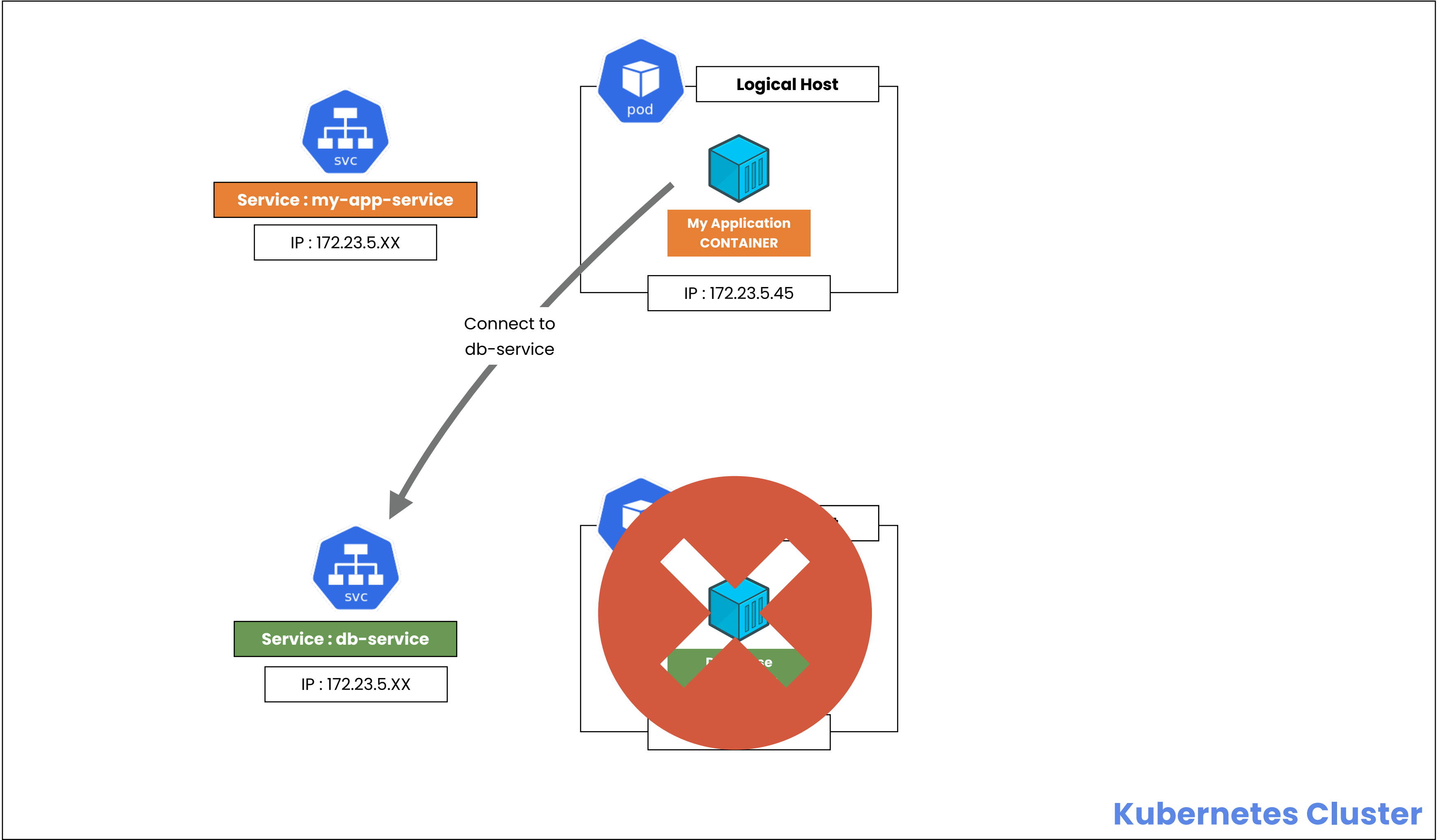
LOGICAL VIEWS WITH NODE AND SERVICE



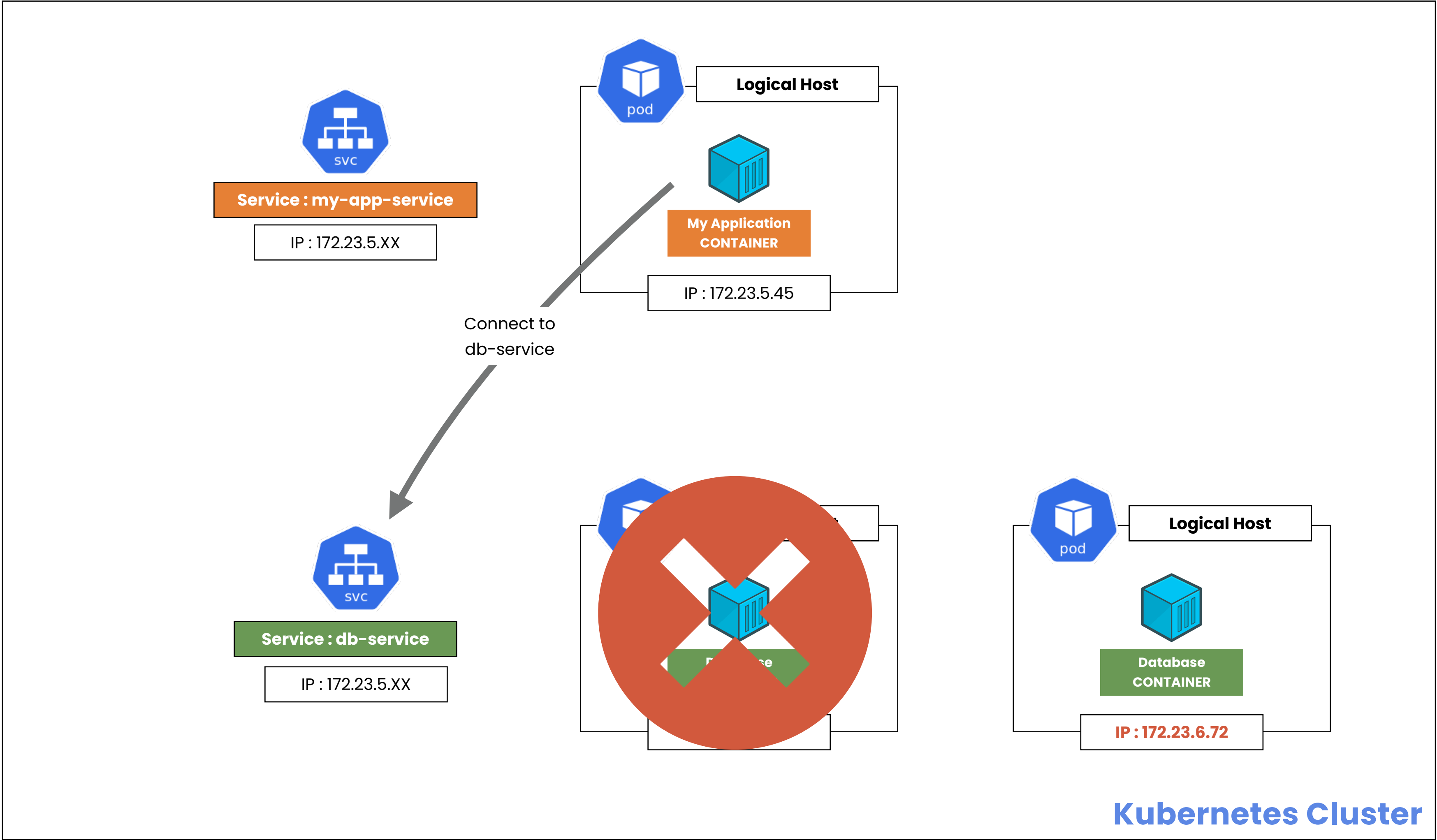
LOGICAL VIEWS WITH NODE AND SERVICE



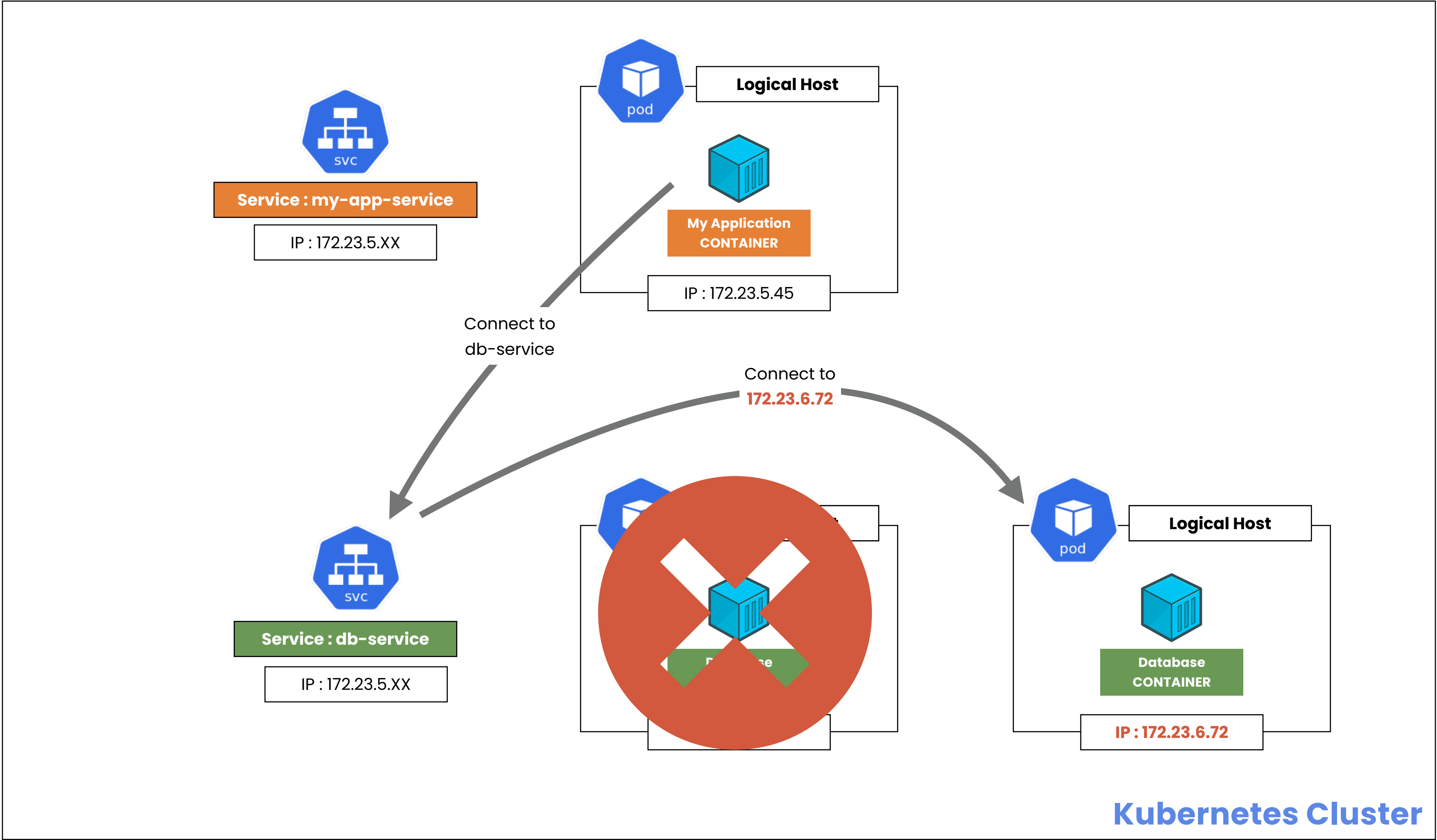
LOGICAL VIEWS WITH NODE AND SERVICE



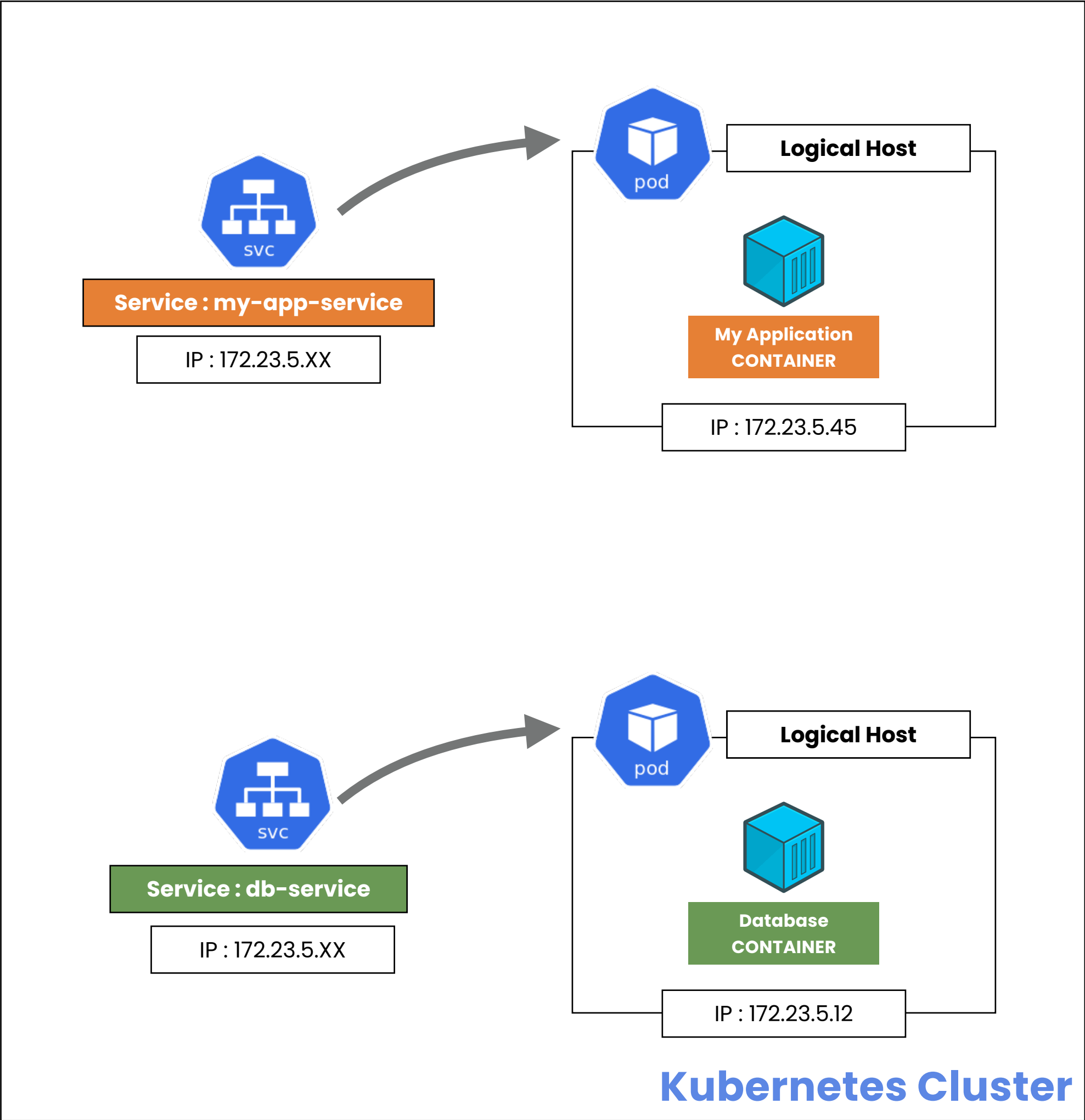
LOGICAL VIEWS WITH NODE AND SERVICE



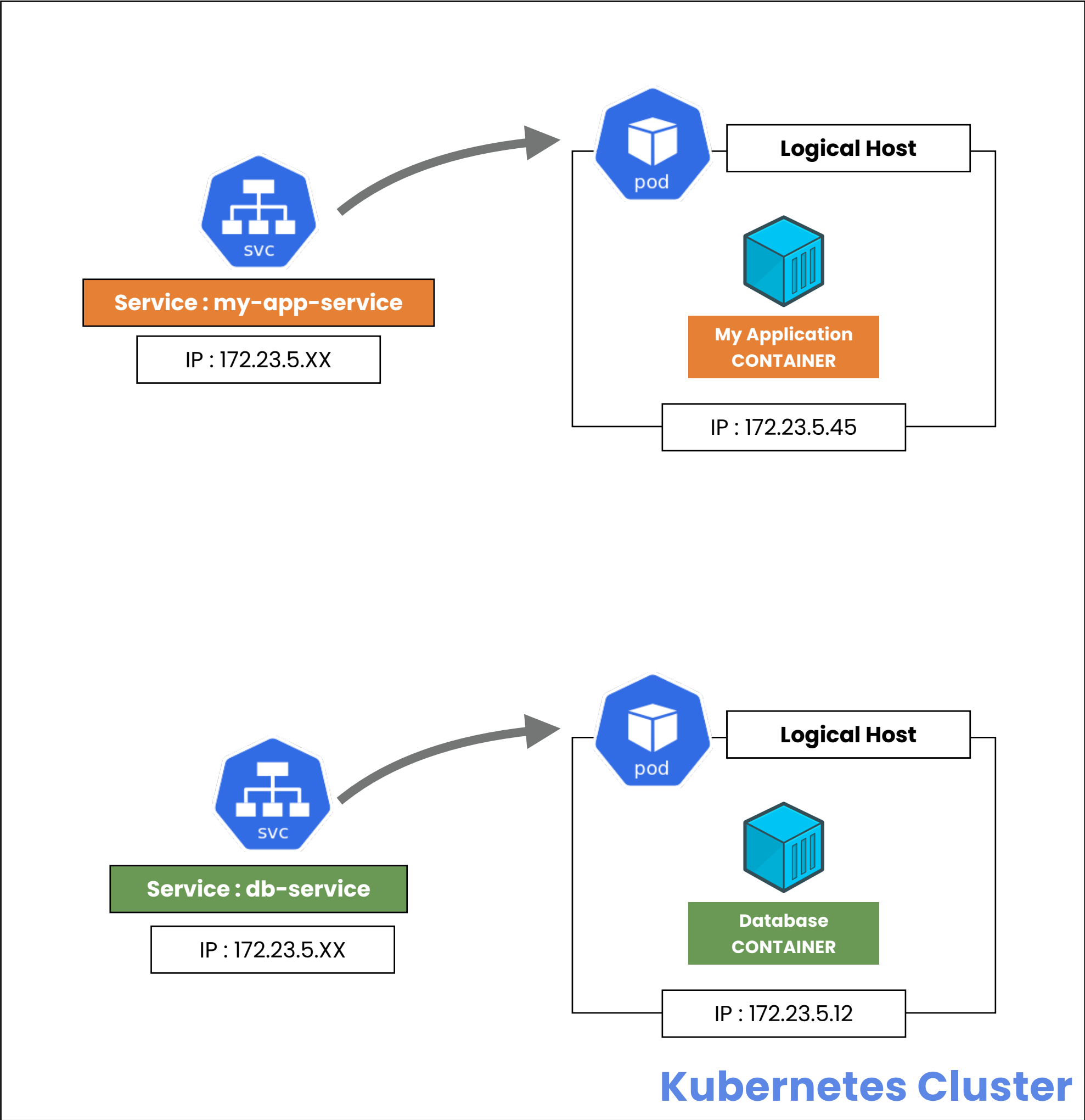
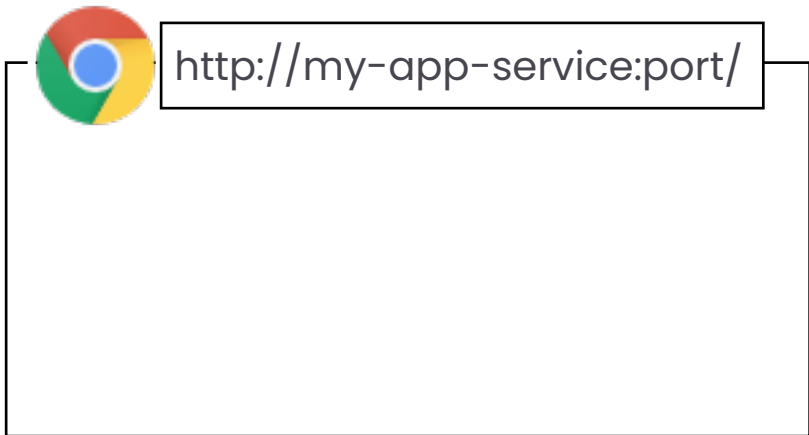
LOGICAL VIEWS WITH NODE AND SERVICE



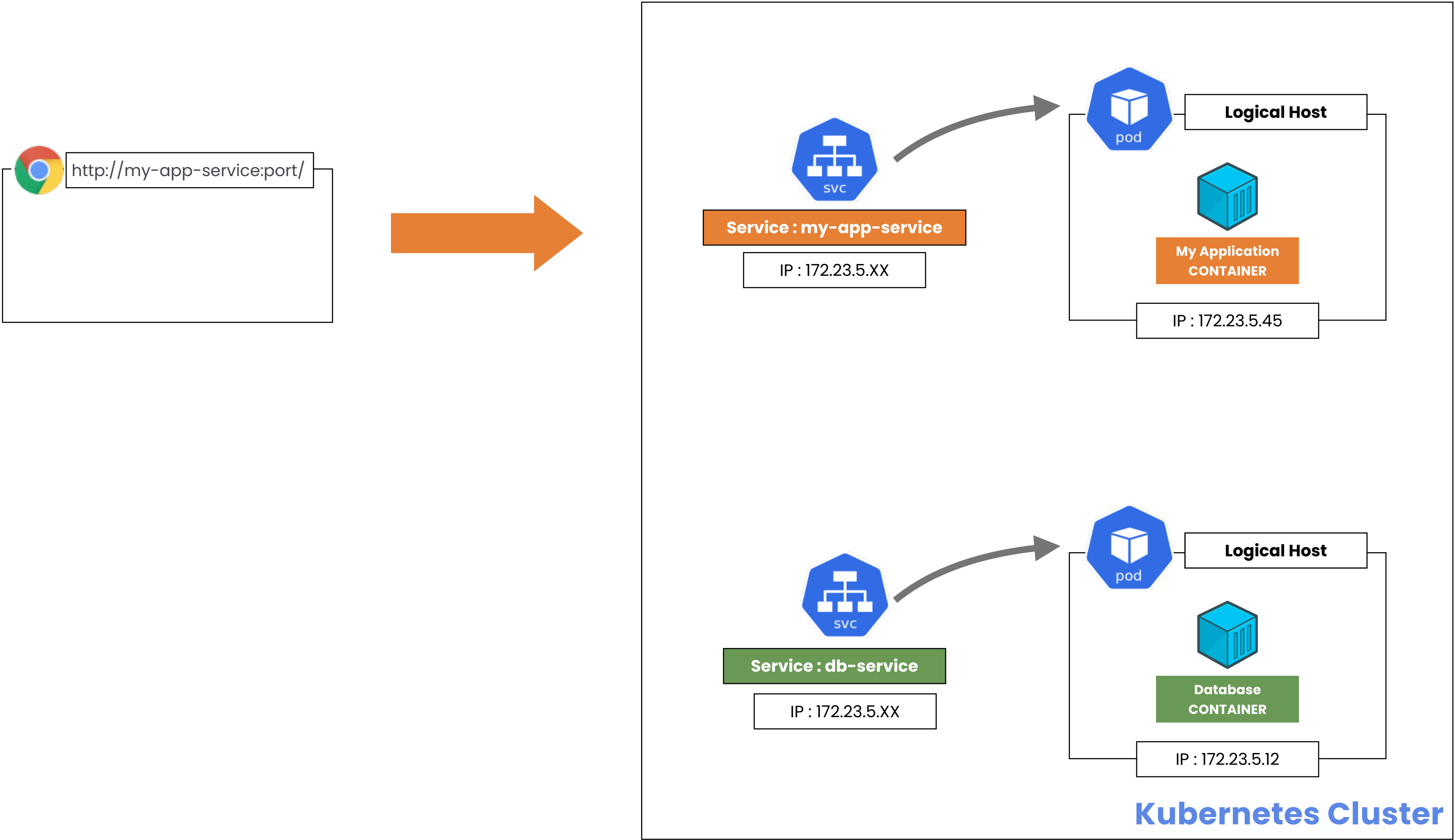
LOGICAL VIEWS WITH NODE AND SERVICE



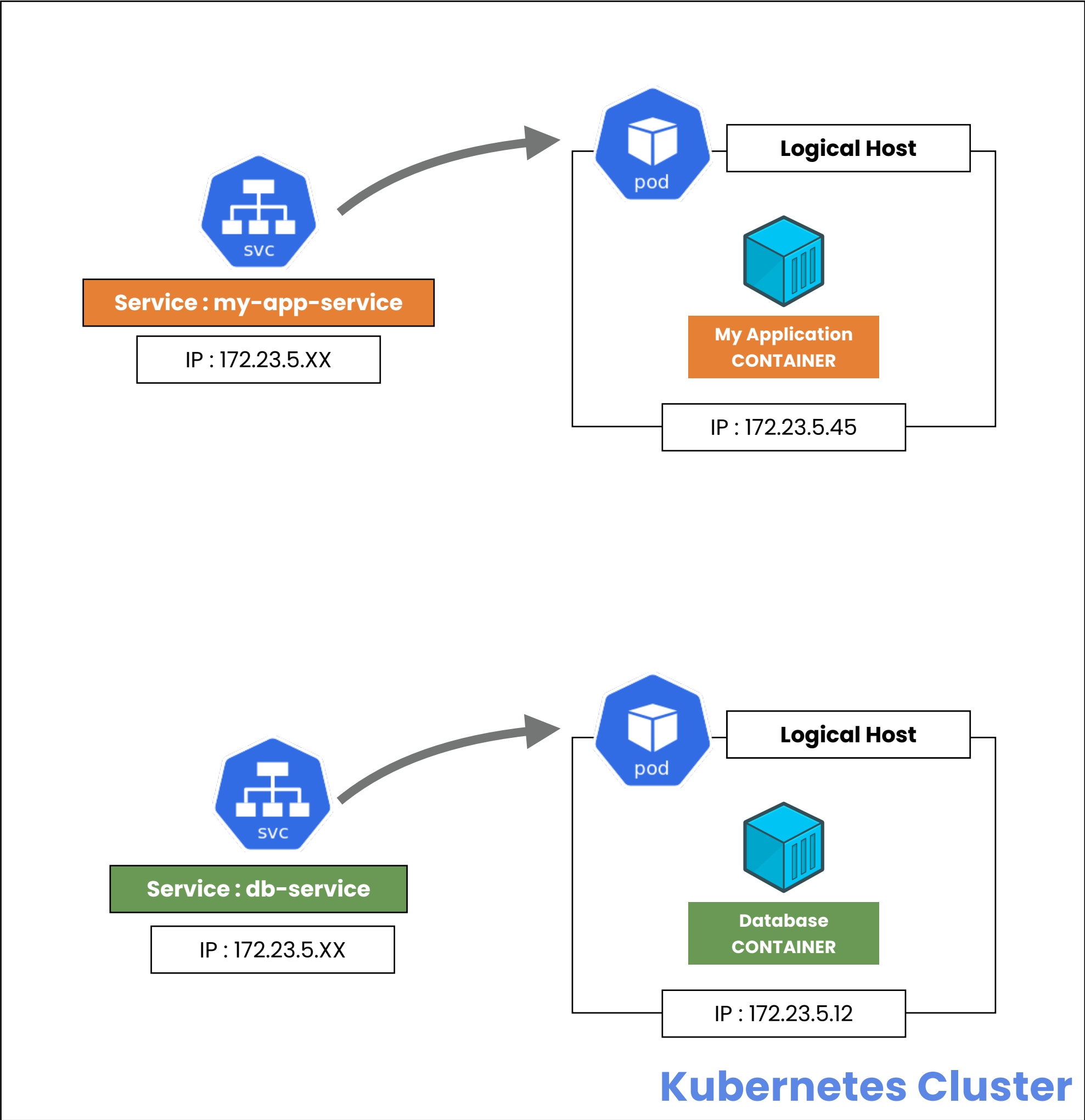
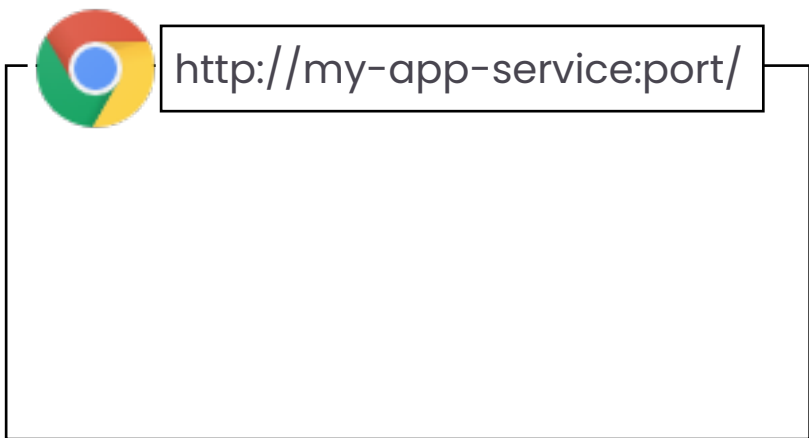
LOGICAL VIEWS WITH NODE AND SERVICE



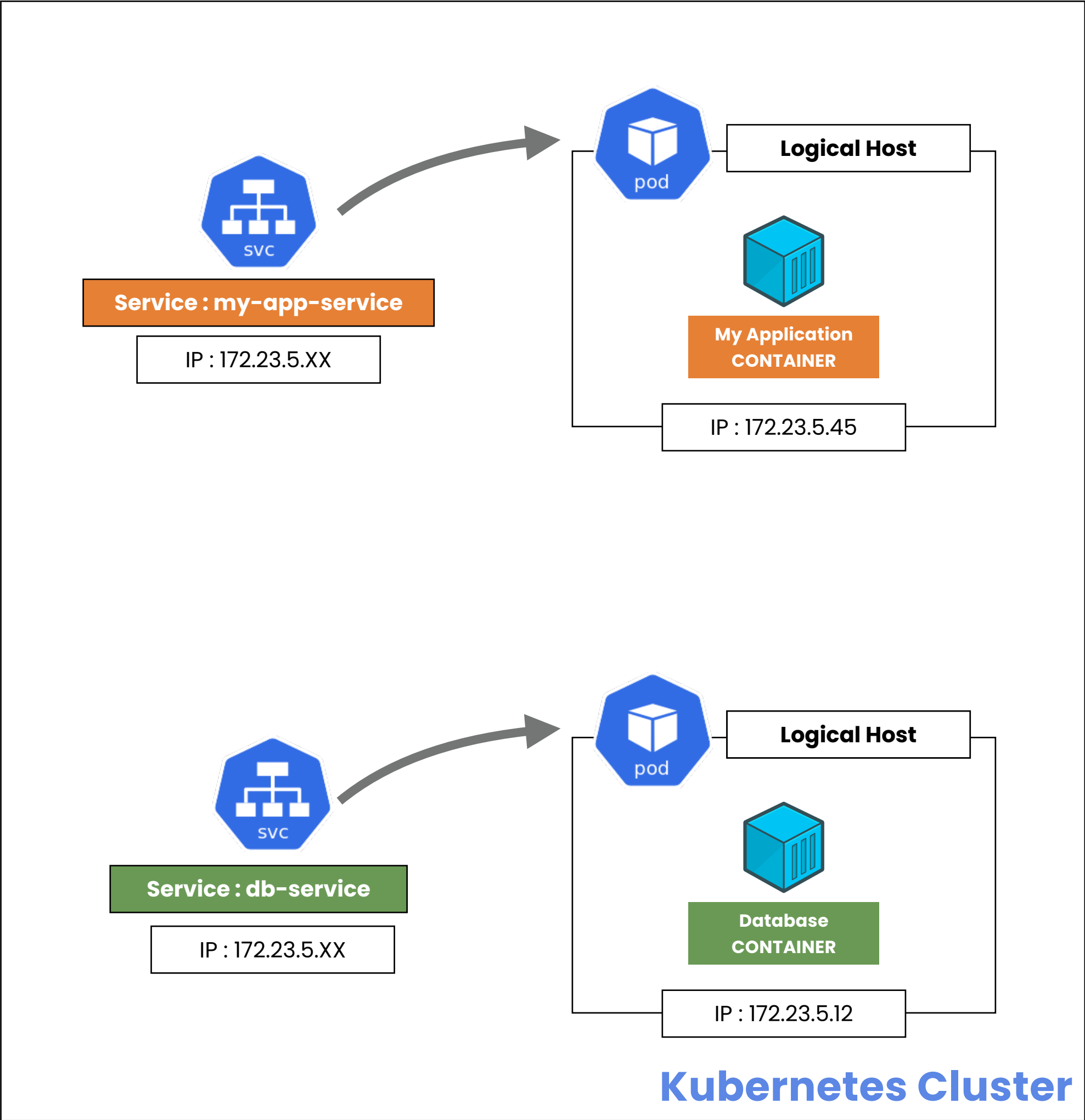
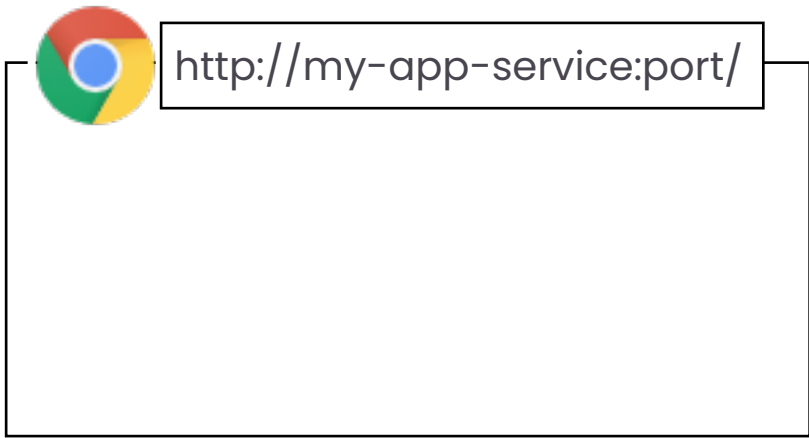
LOGICAL VIEWS WITH NODE AND SERVICE



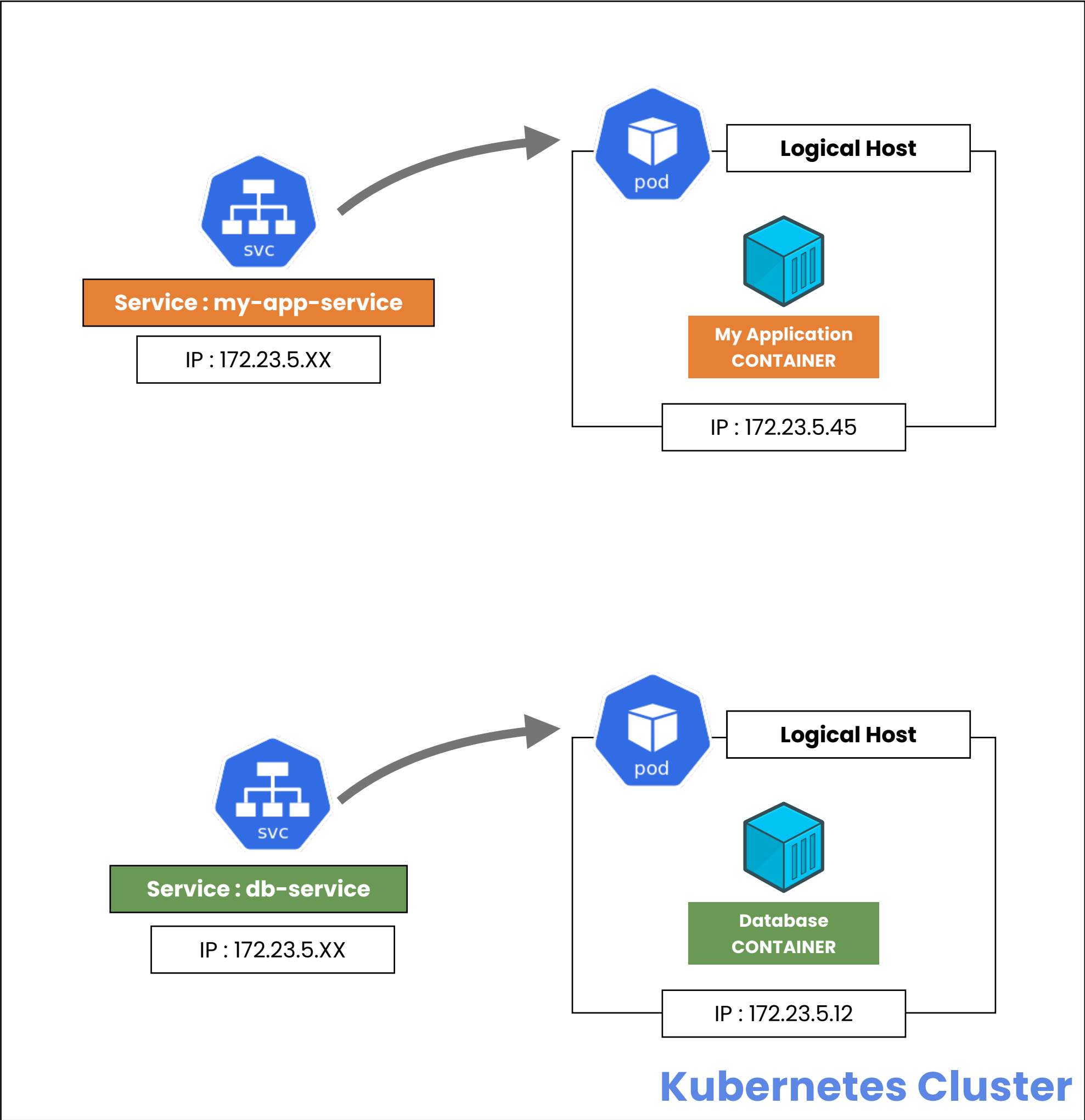
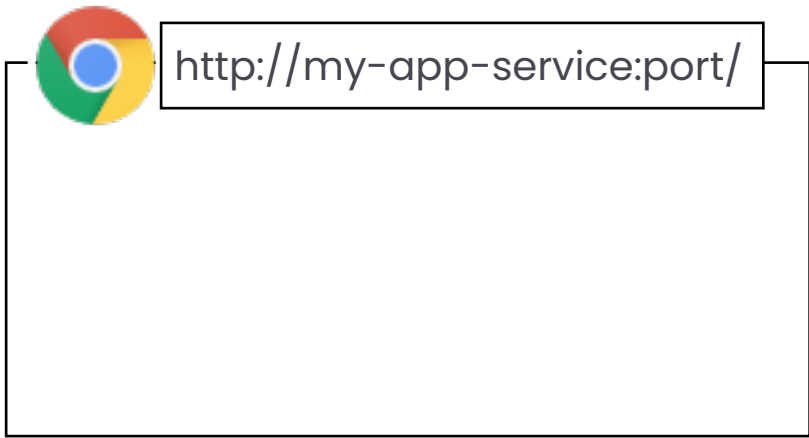
LOGICAL VIEWS WITH NODE AND SERVICE



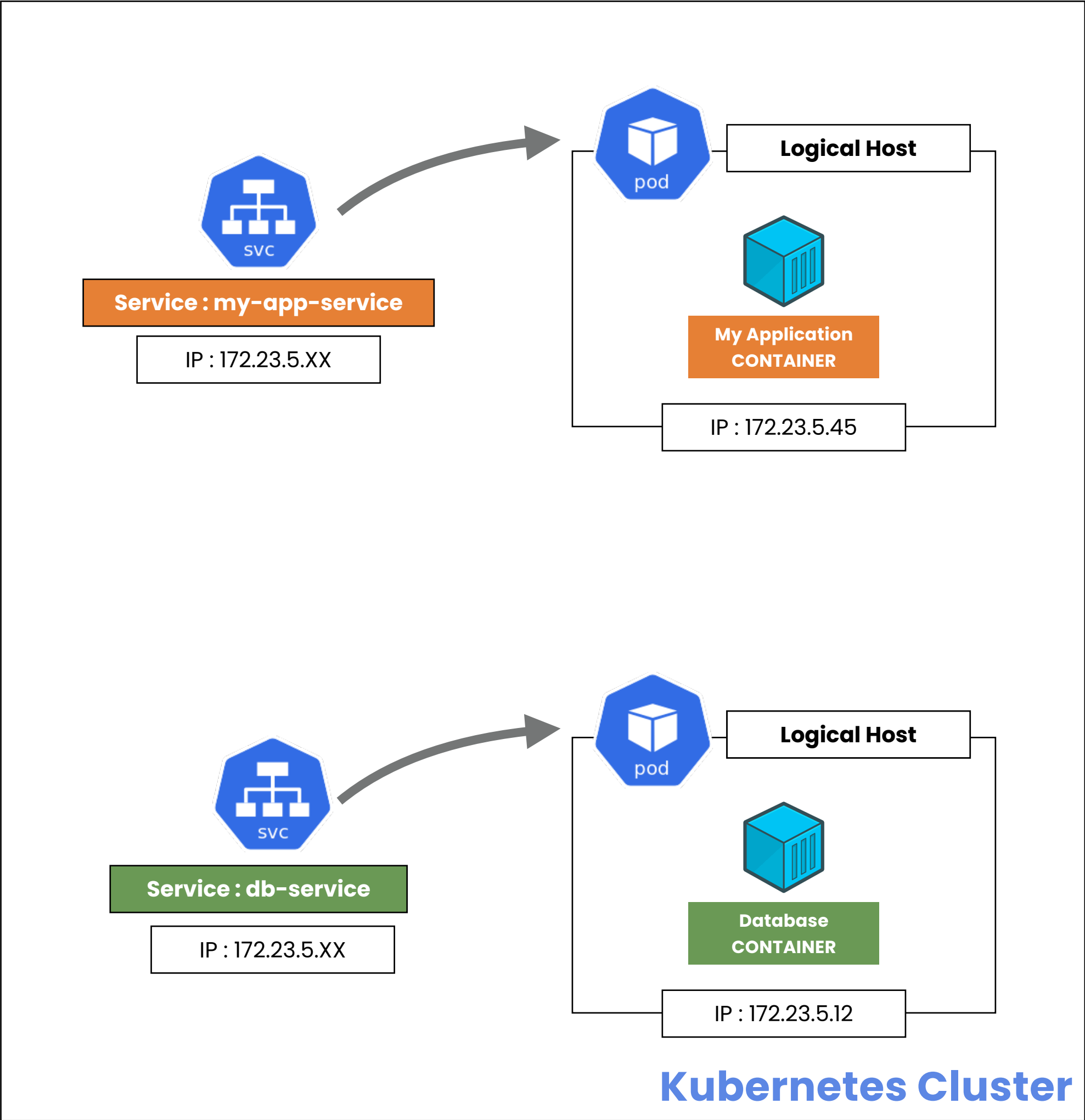
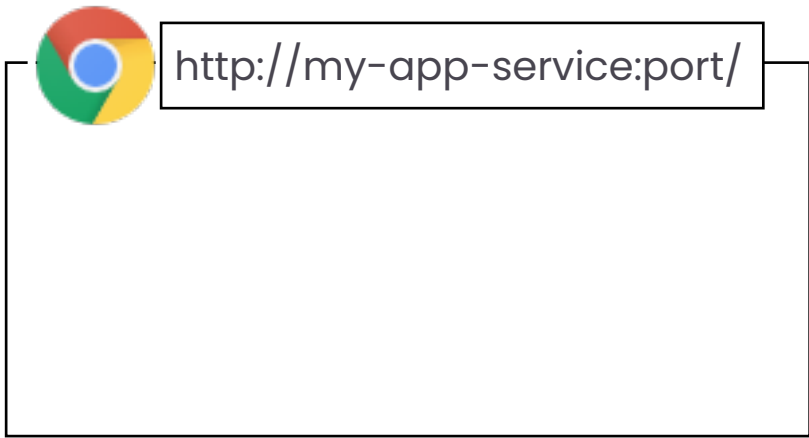
LOGICAL VIEWS WITH NODE AND SERVICE



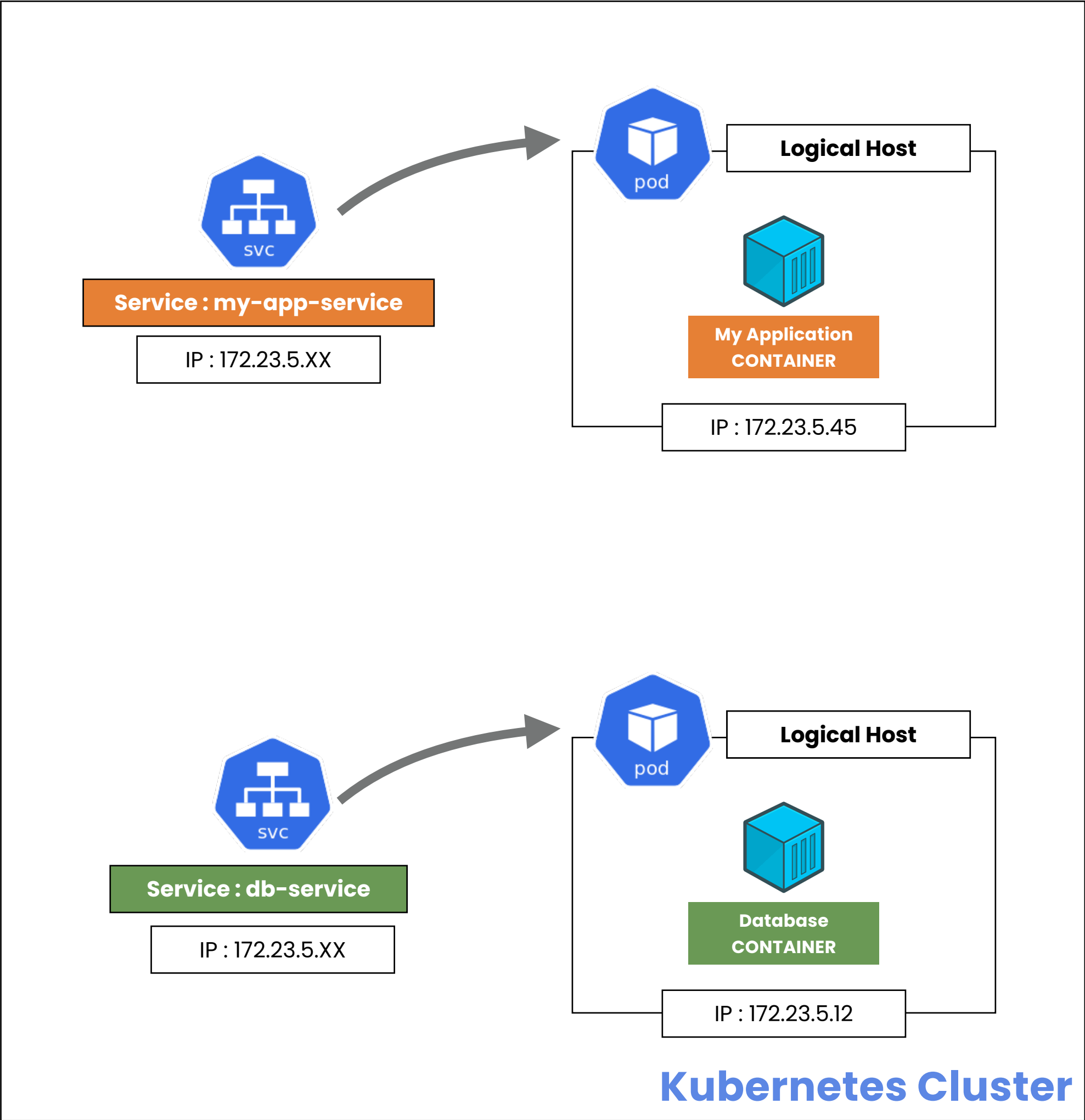
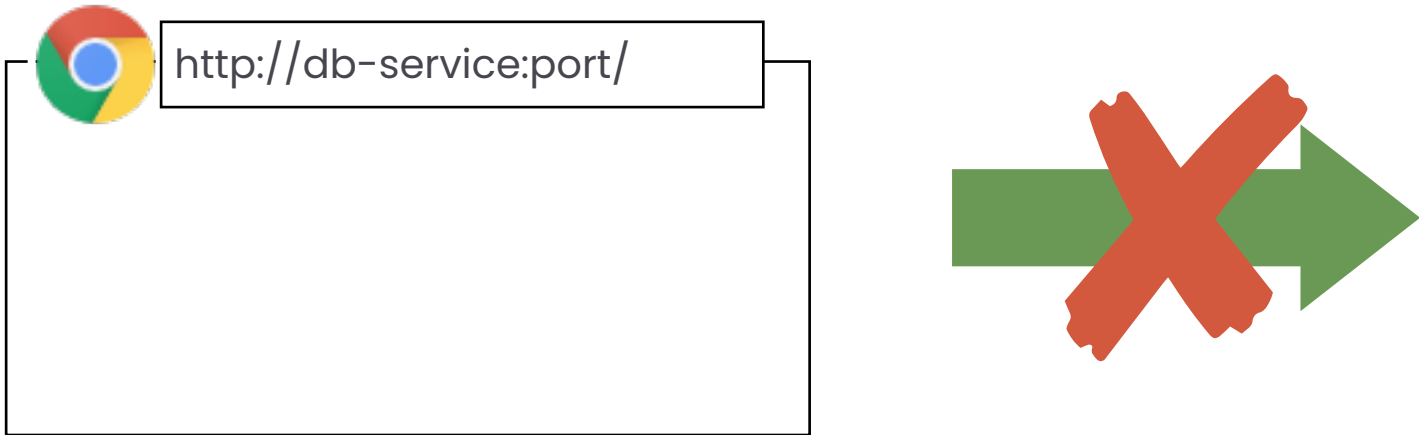
LOGICAL VIEWS WITH NODE AND SERVICE



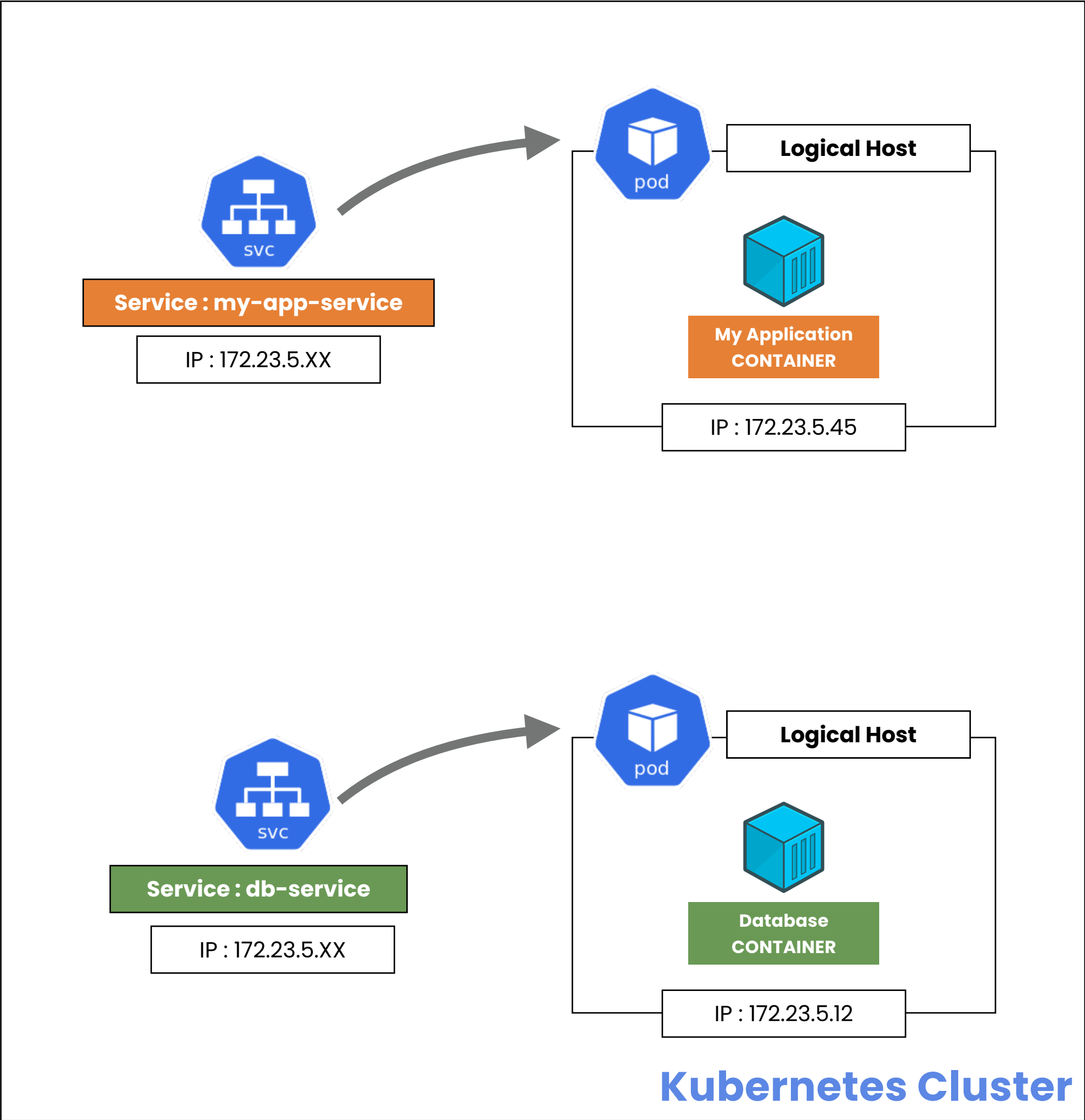
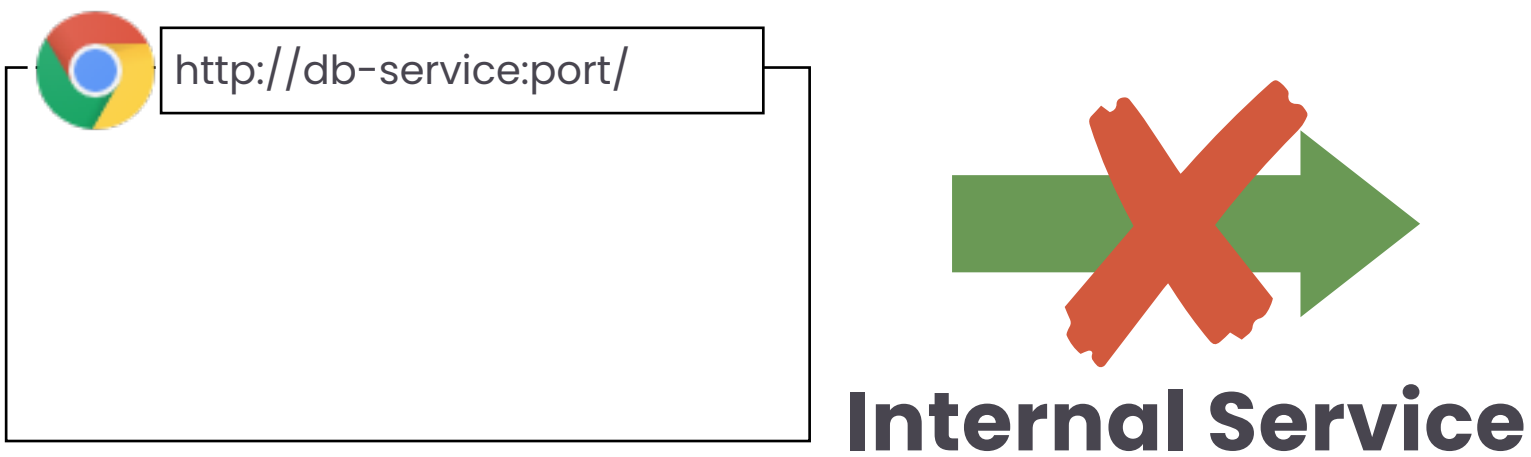
LOGICAL VIEWS WITH NODE AND SERVICE



LOGICAL VIEWS WITH NODE AND SERVICE

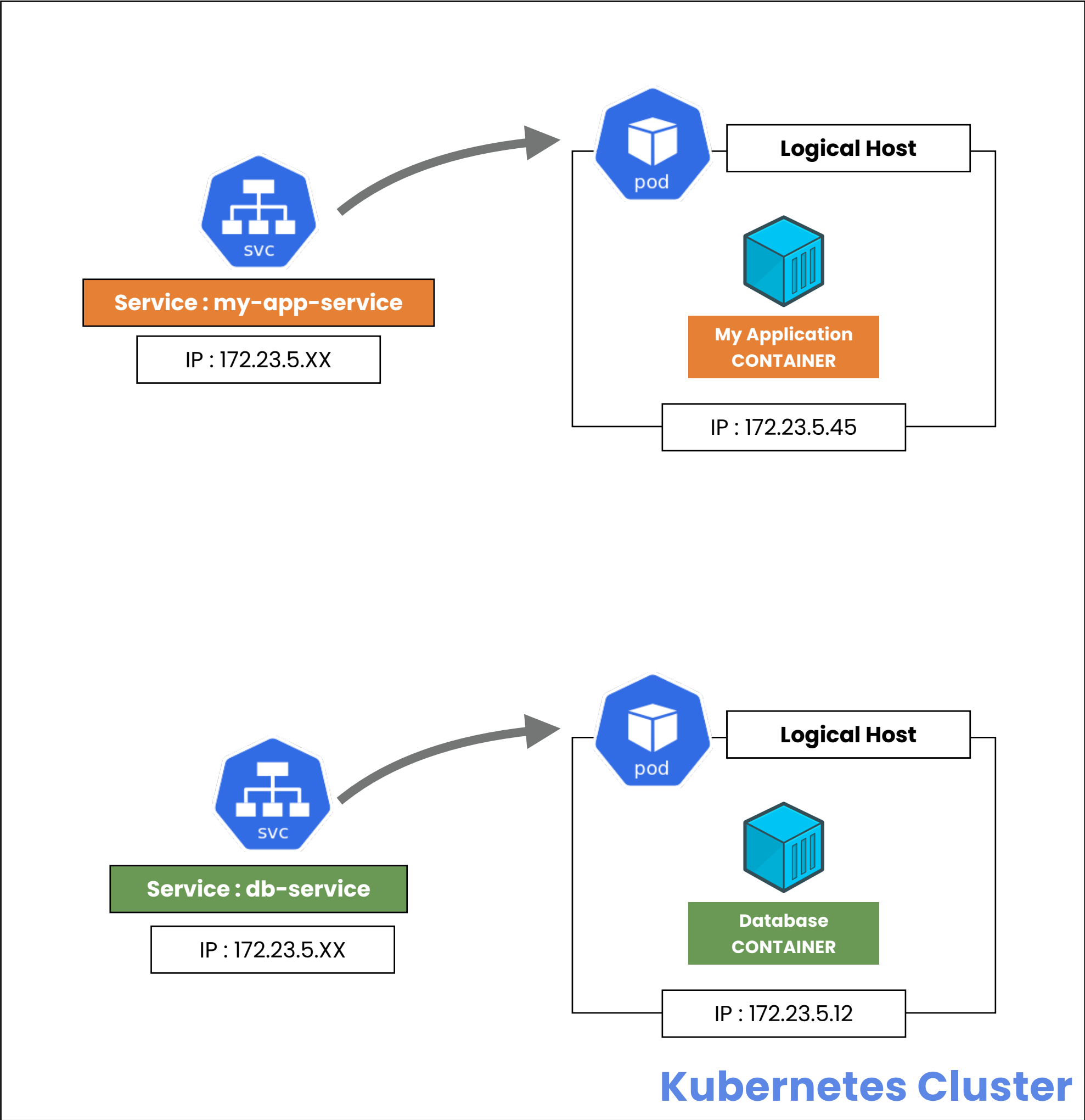


LOGICAL VIEWS WITH NODE AND SERVICE

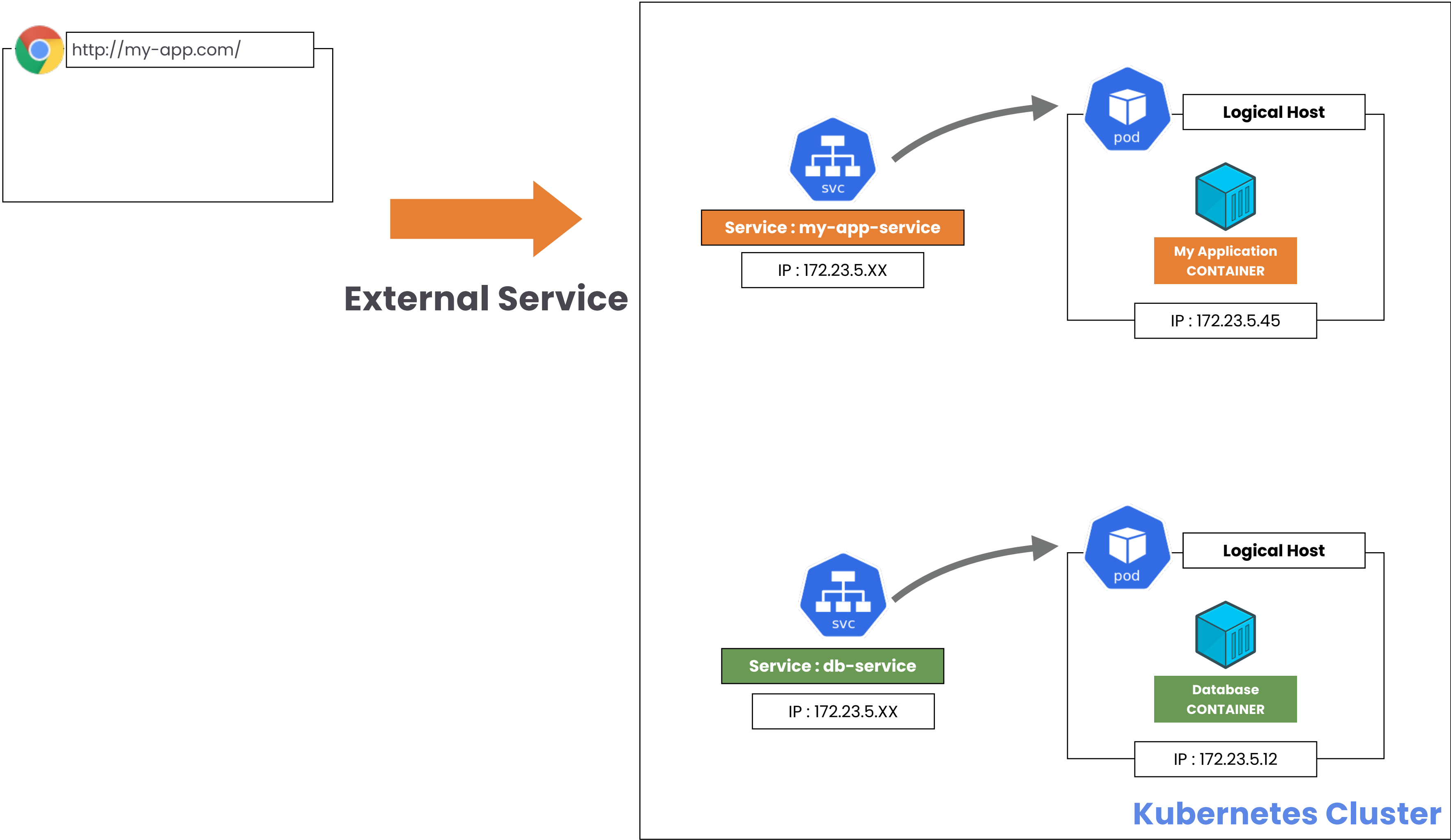


LOGICAL VIEWS WITH NODE AND SERVICE

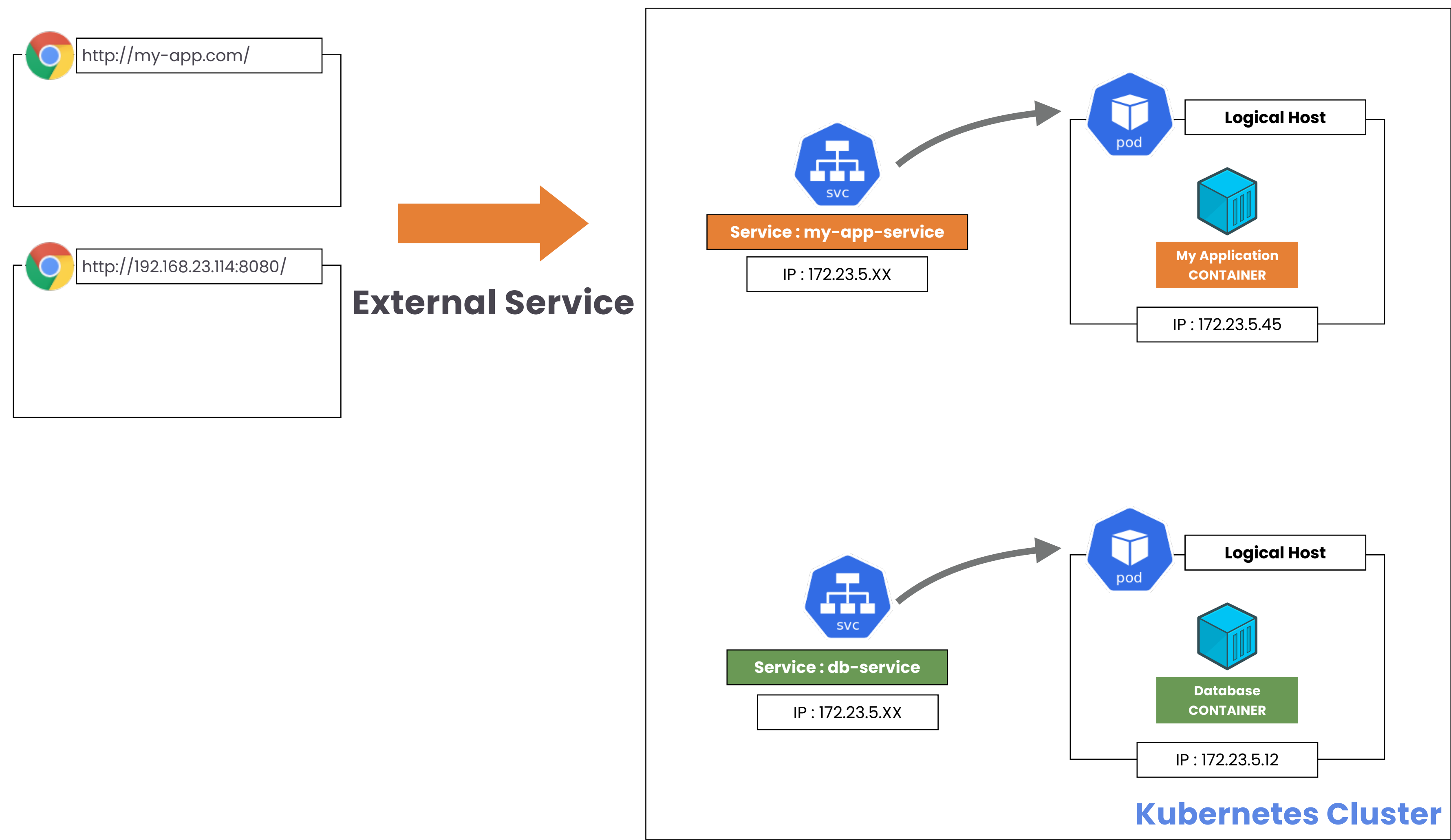

External Service



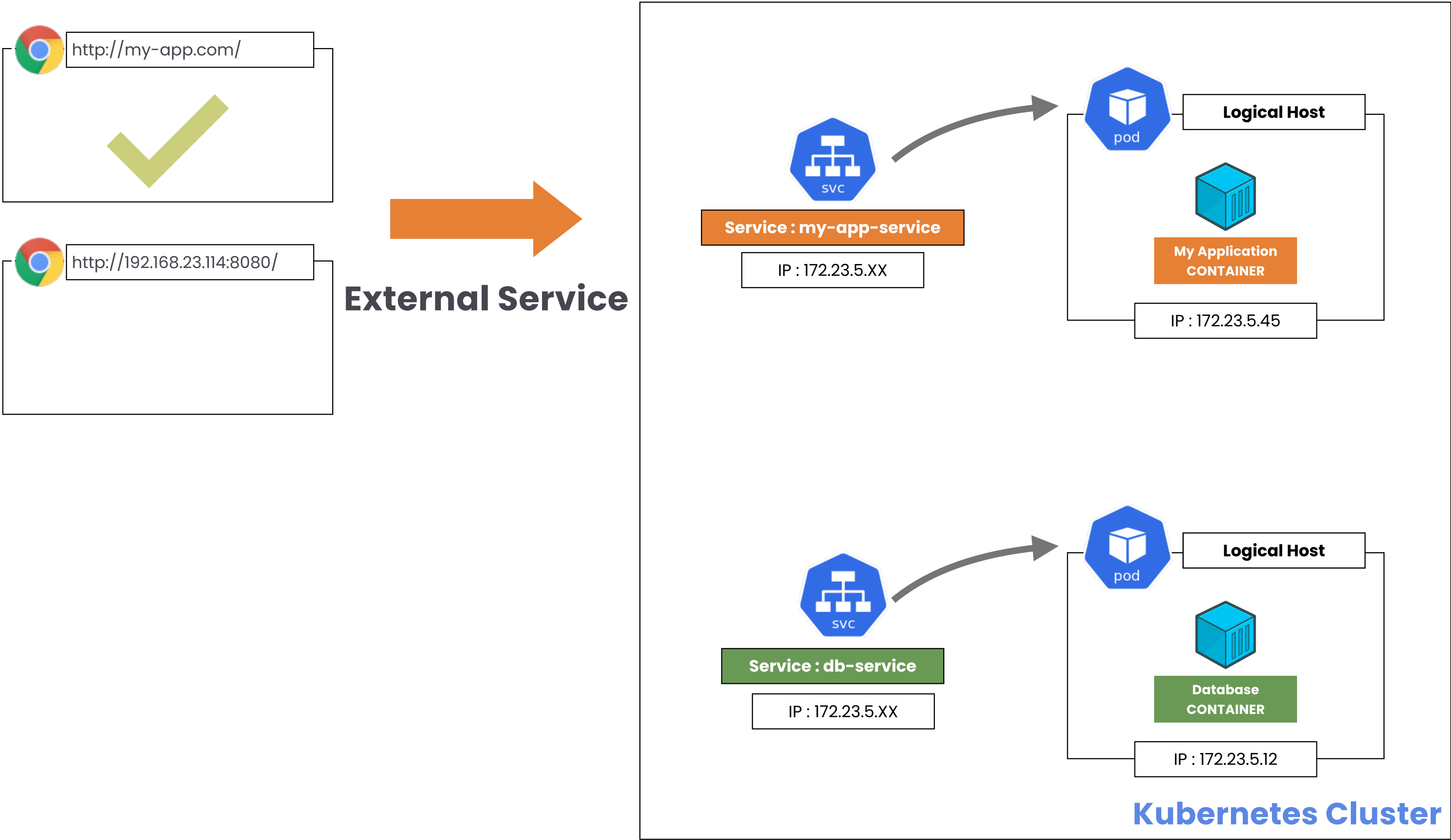
LOGICAL VIEWS WITH NODE AND SERVICE



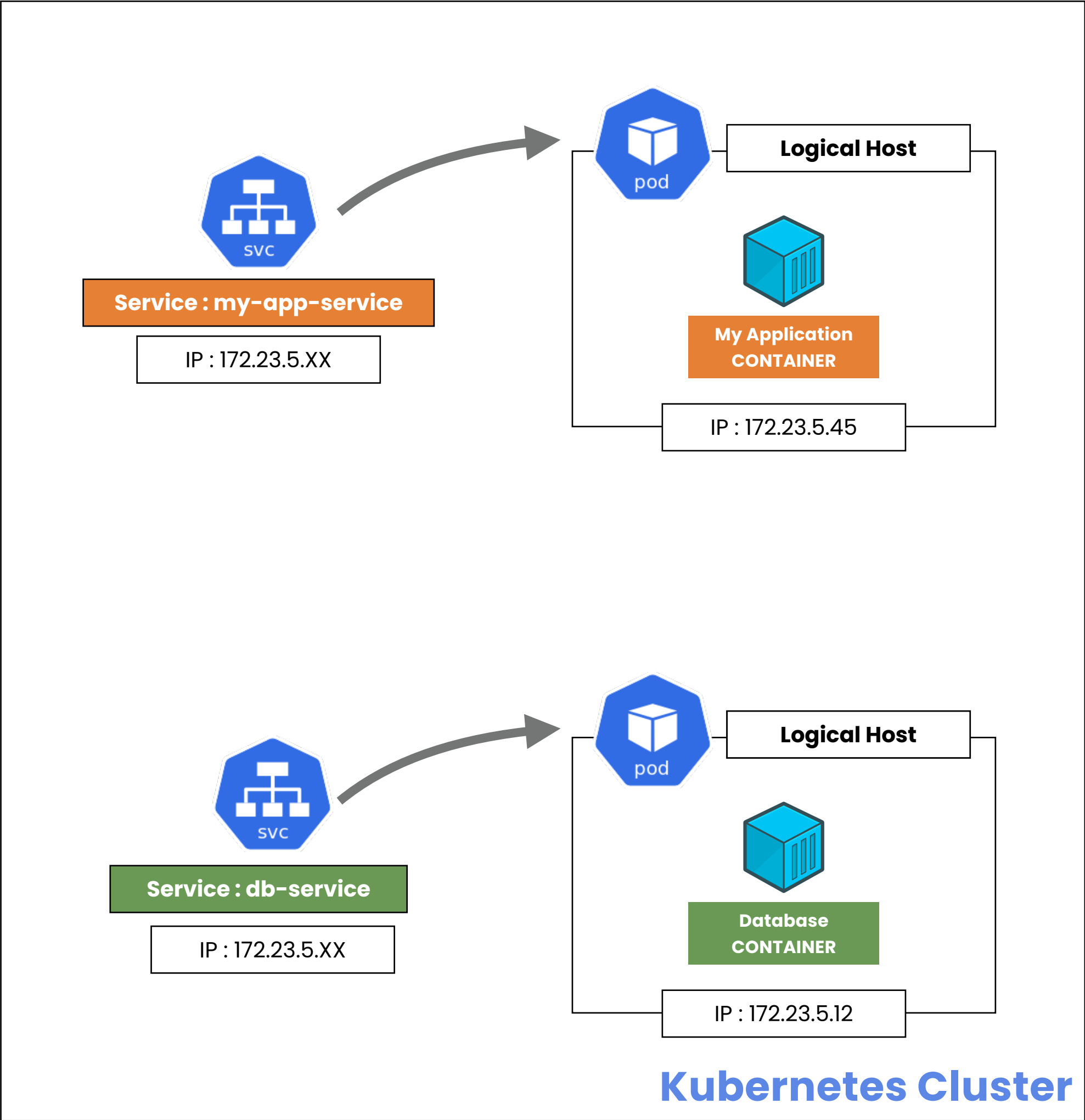
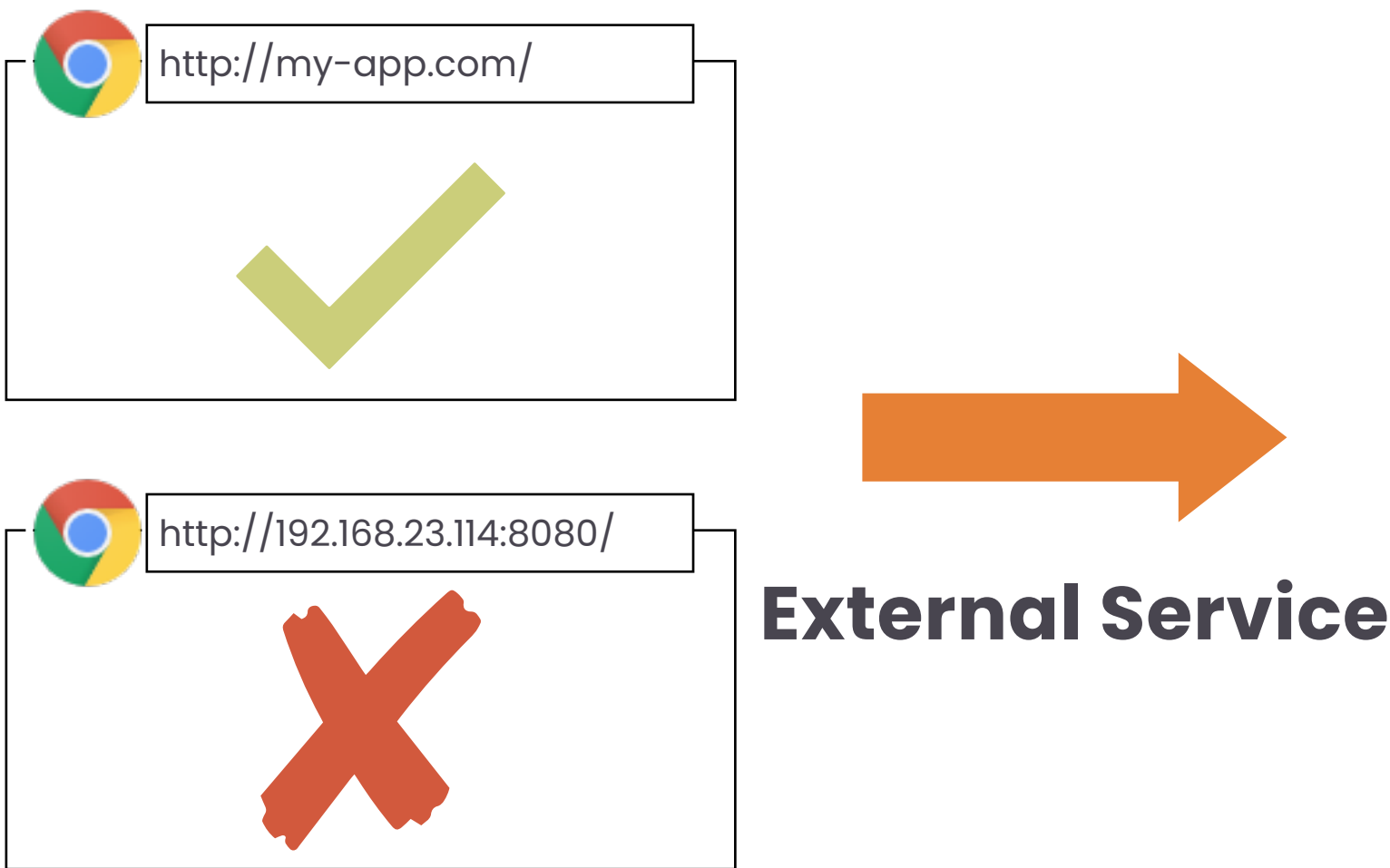
LOGICAL VIEWS WITH NODE AND SERVICE



LOGICAL VIEWS WITH NODE AND SERVICE



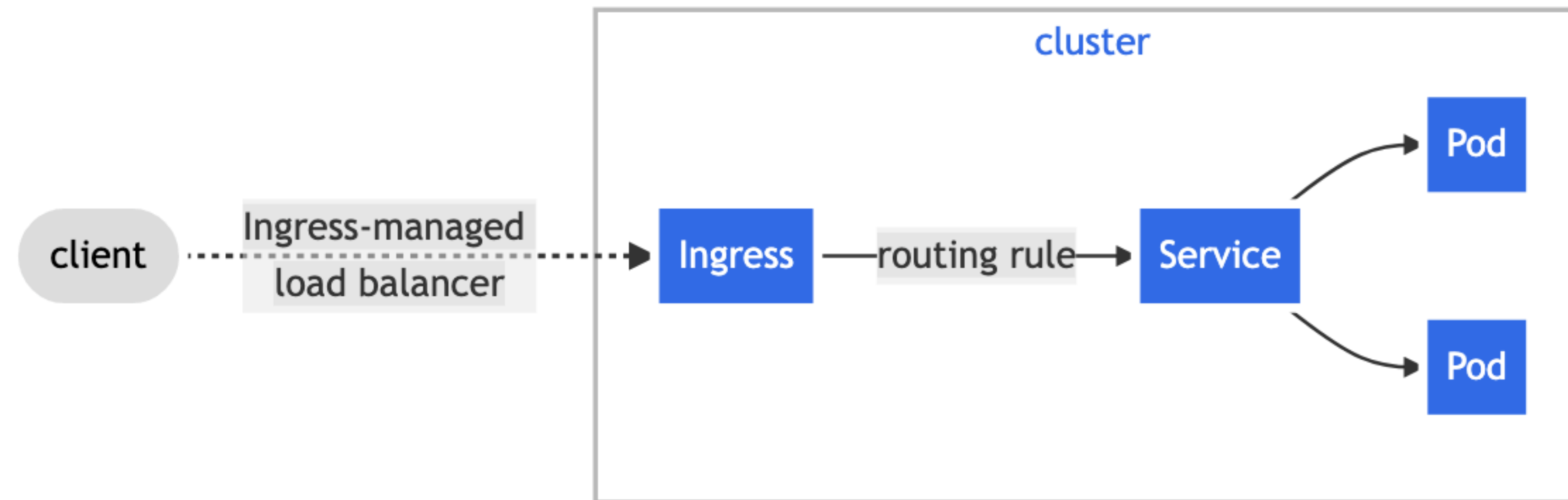
LOGICAL VIEWS WITH NODE AND SERVICE



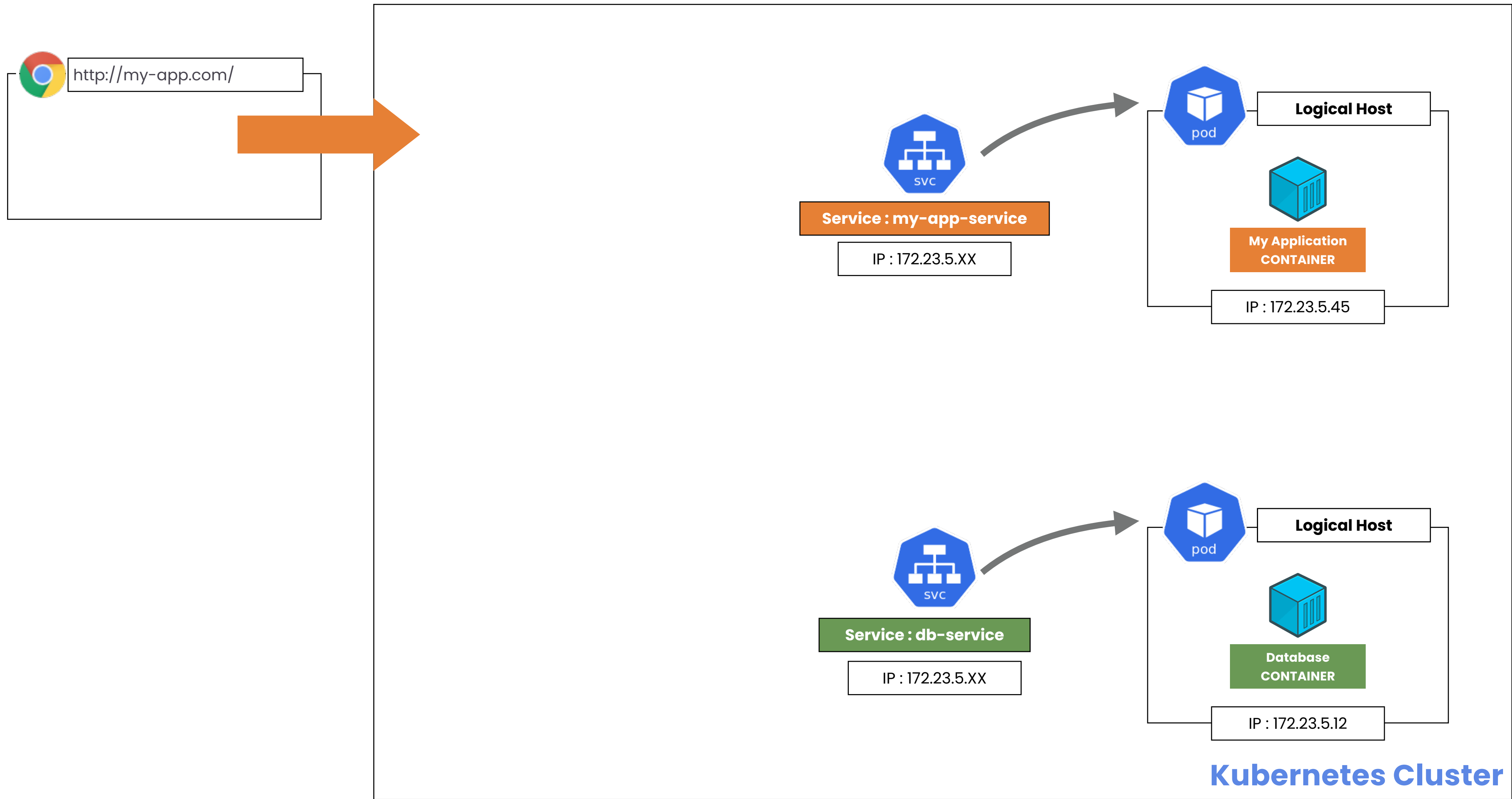
WHAT'S INGRESS

An API object that manages external access to the services in a cluster, typically HTTP.

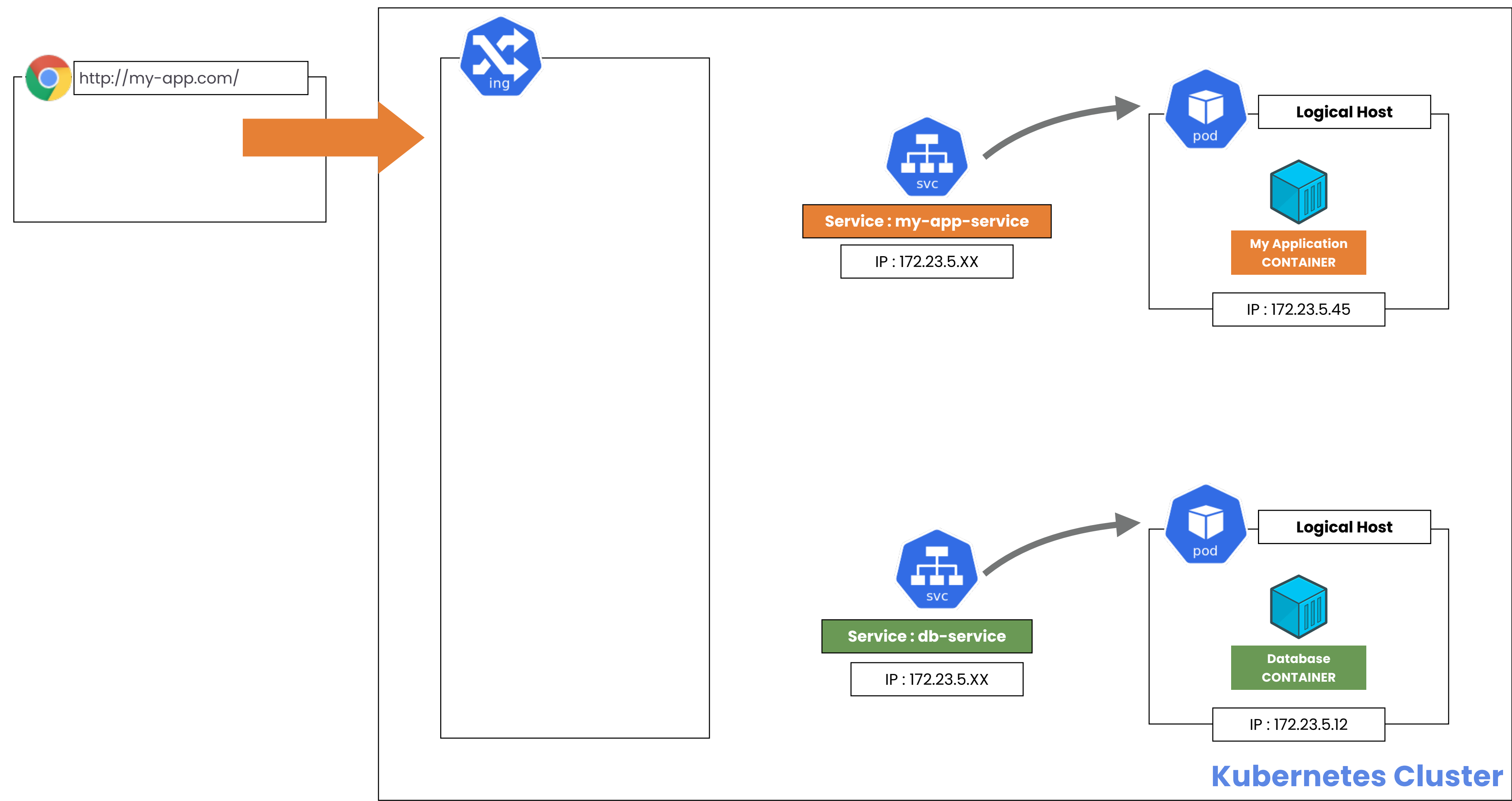
Ingress exposes HTTP and HTTPS routes from outside the cluster to **services** within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.



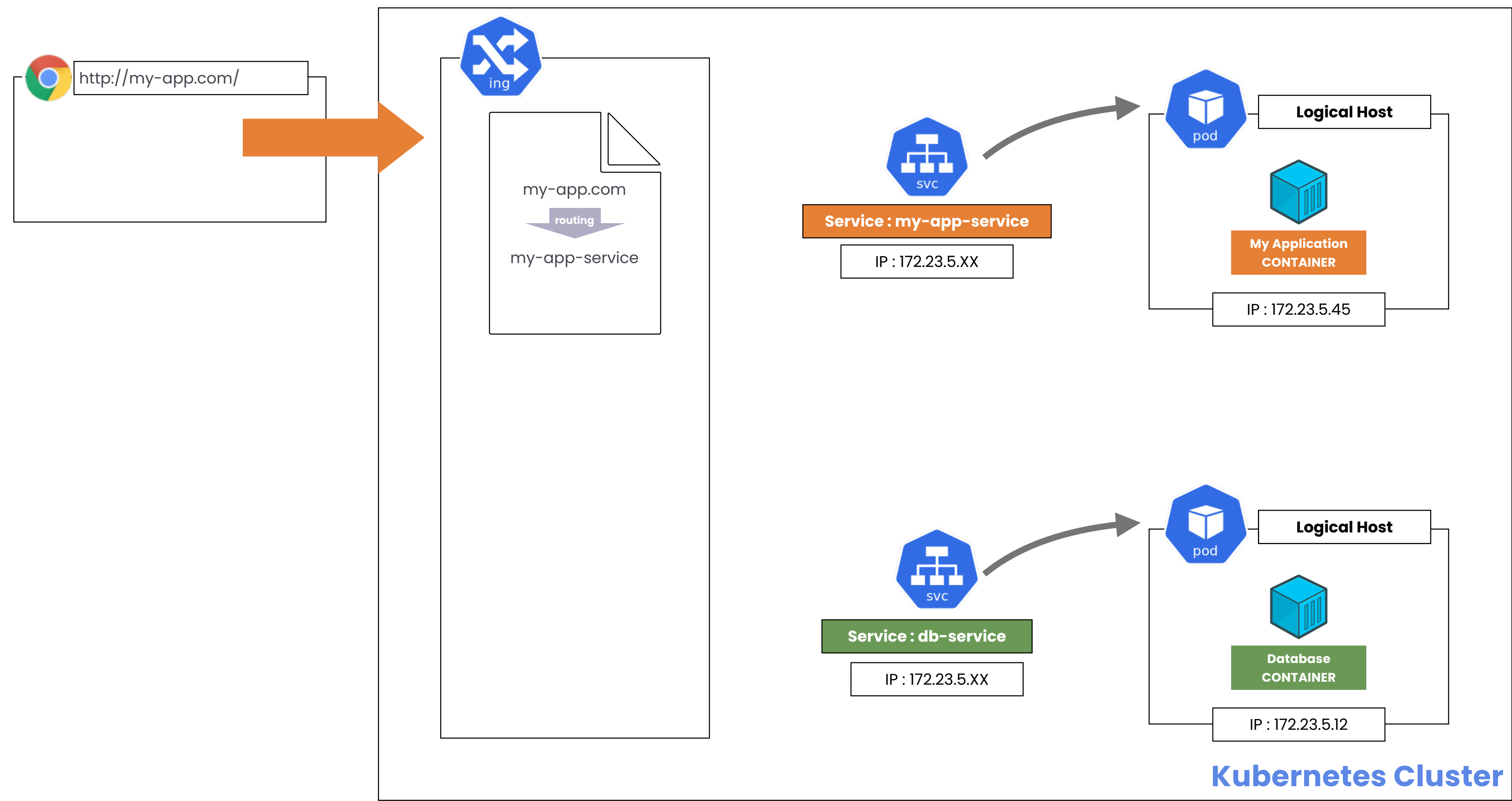
LOGICAL VIEWS WITH NODE AND SERVICE



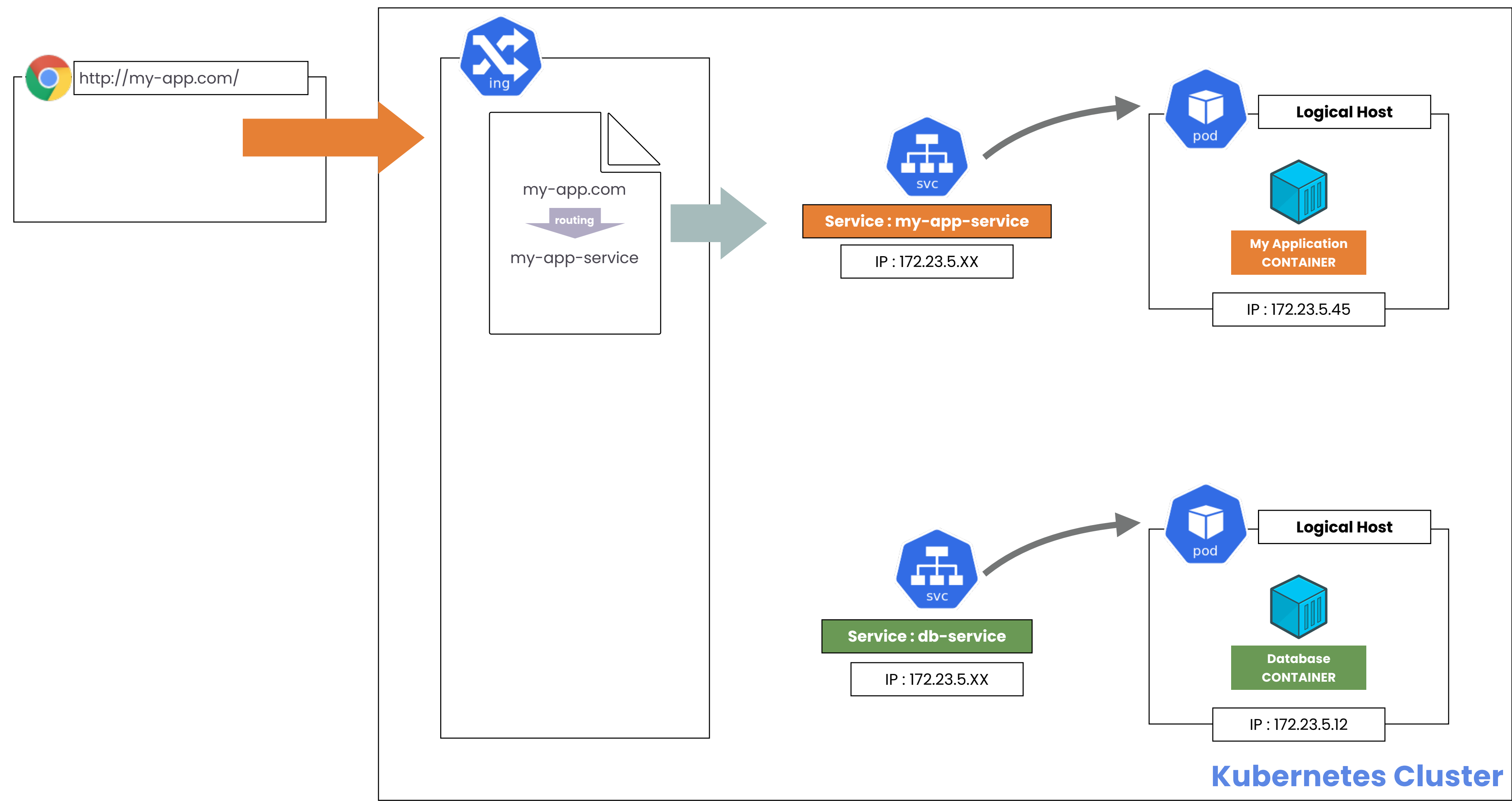
LOGICAL VIEWS WITH NODE AND SERVICE



LOGICAL VIEWS WITH NODE AND SERVICE



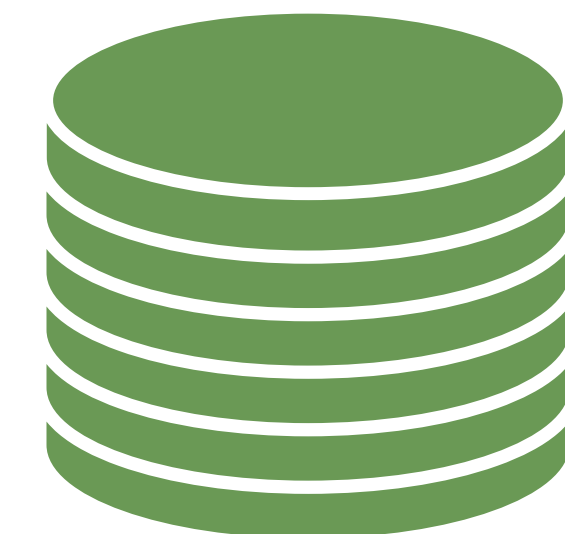
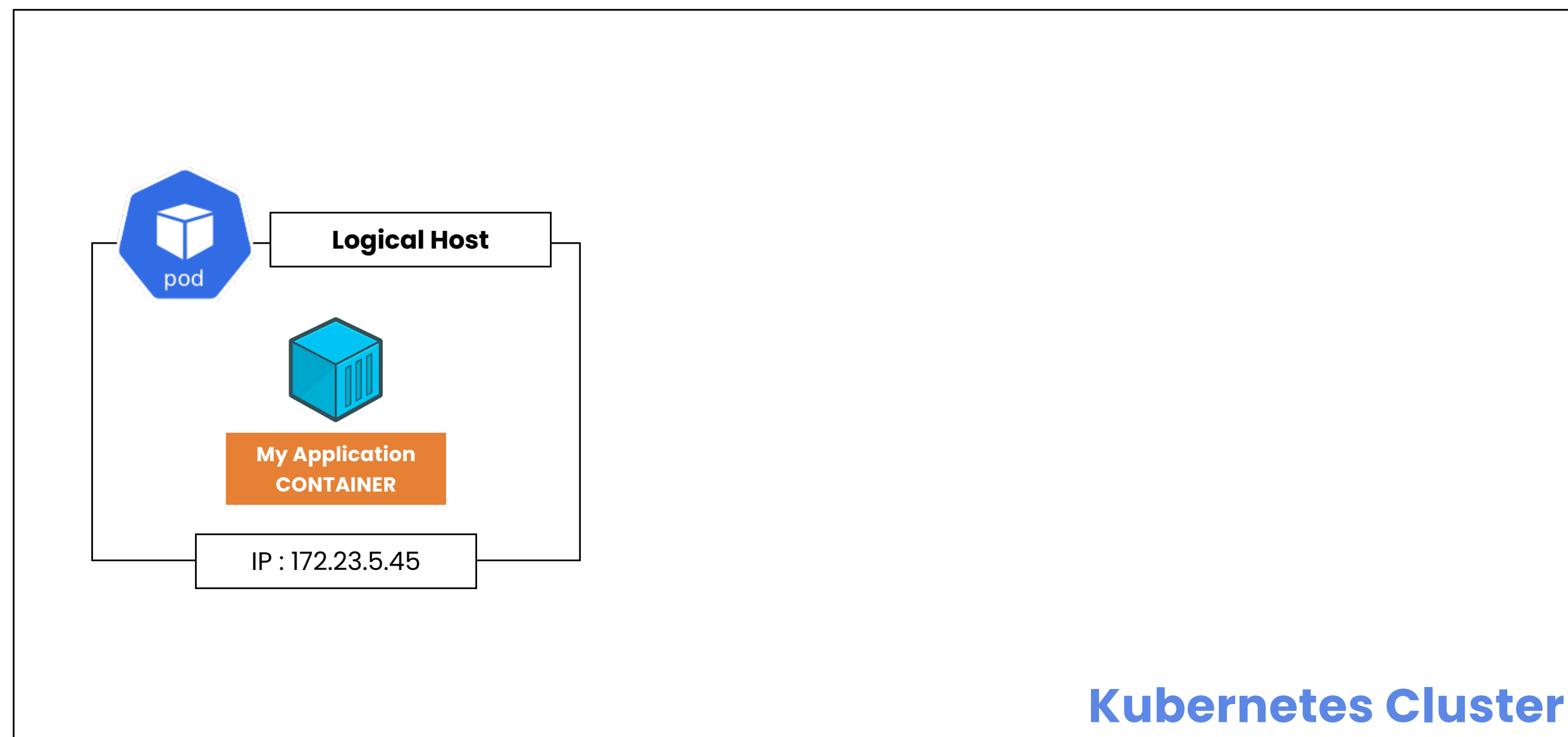
LOGICAL VIEWS WITH NODE AND SERVICE



SERVICE WITHOUT SELECTOR

Services most commonly abstract access to Kubernetes Pods, but they can also abstract other kinds of backends. For example:

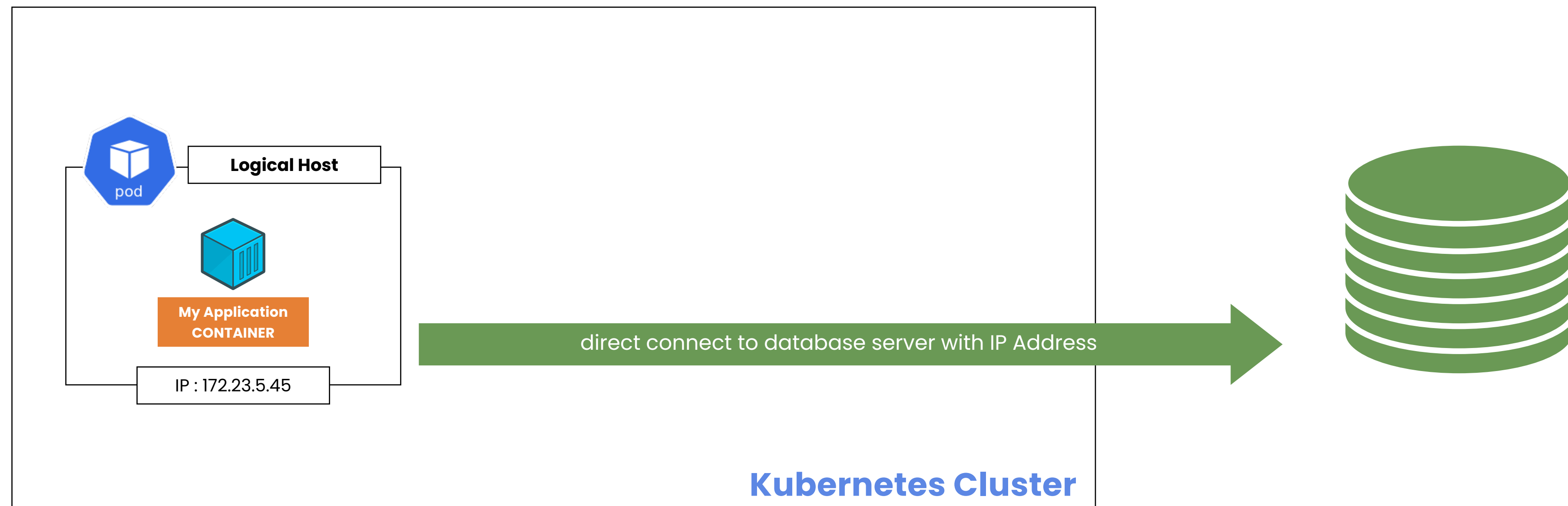
- You want to have an external database cluster in production, but in your test environment you use your own databases.
- You want to point your Service to a Service in a different Namespace or on another cluster.
- You are migrating a workload to Kubernetes. While evaluating the approach, you run only a portion of your backends in Kubernetes.



SERVICE WITHOUT SELECTOR

Services most commonly abstract access to Kubernetes Pods, but they can also abstract other kinds of backends. For example:

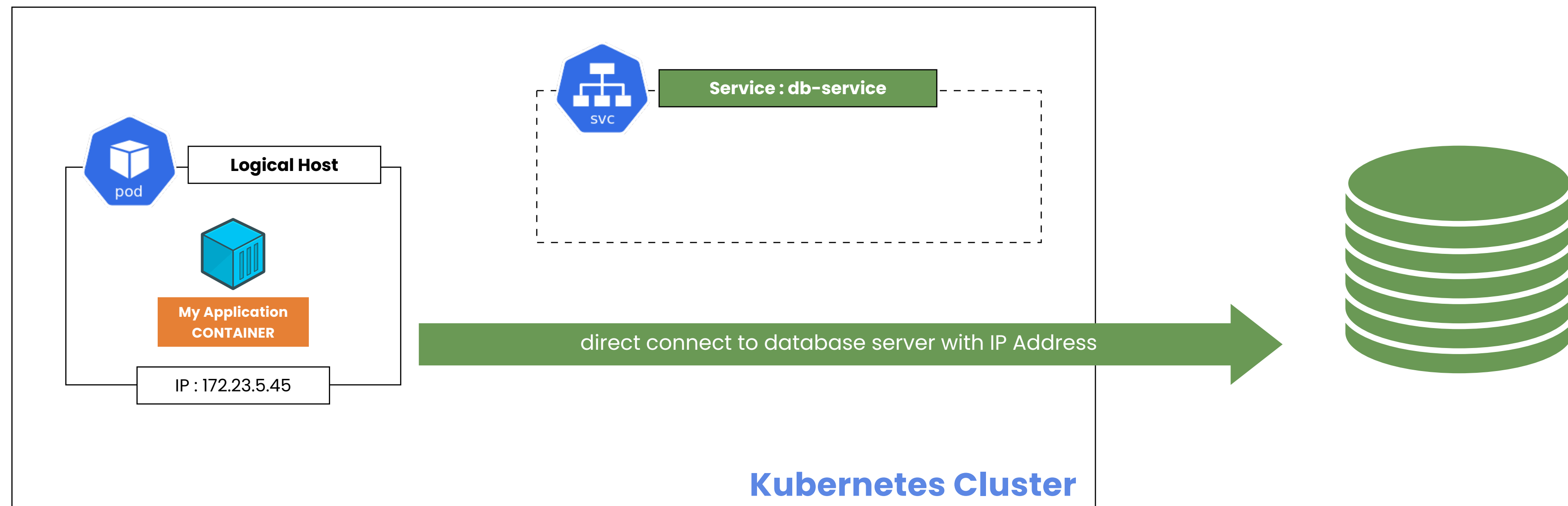
- You want to have an external database cluster in production, but in your test environment you use your own databases.
- You want to point your Service to a Service in a different Namespace or on another cluster.
- You are migrating a workload to Kubernetes. While evaluating the approach, you run only a portion of your backends in Kubernetes.



SERVICE WITHOUT SELECTOR

Services most commonly abstract access to Kubernetes Pods, but they can also abstract other kinds of backends. For example:

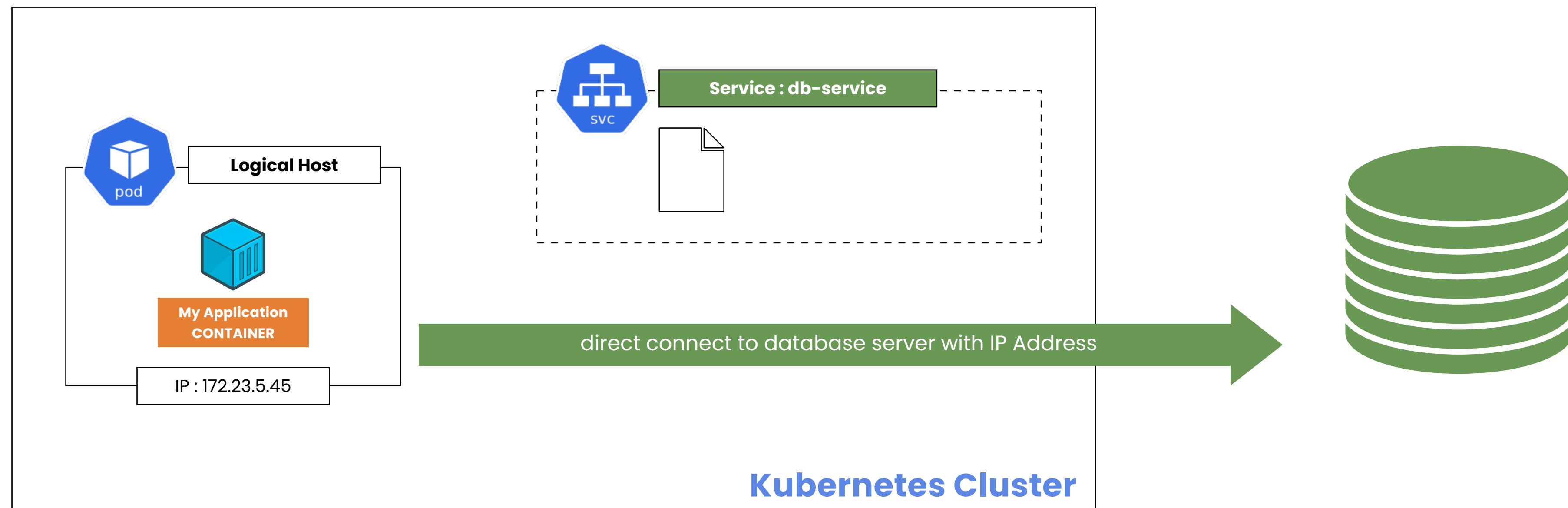
- You want to have an external database cluster in production, but in your test environment you use your own databases.
- You want to point your Service to a Service in a different Namespace or on another cluster.
- You are migrating a workload to Kubernetes. While evaluating the approach, you run only a portion of your backends in Kubernetes.



SERVICE WITHOUT SELECTOR

Services most commonly abstract access to Kubernetes Pods, but they can also abstract other kinds of backends. For example:

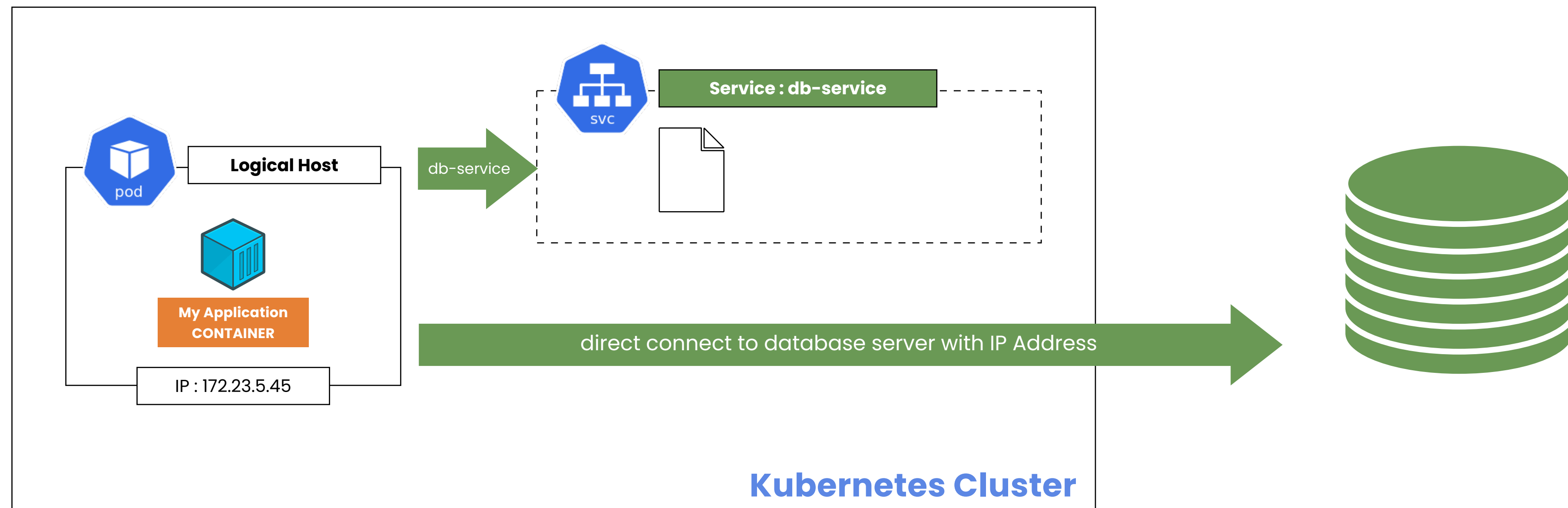
- You want to have an external database cluster in production, but in your test environment you use your own databases.
- You want to point your Service to a Service in a different Namespace or on another cluster.
- You are migrating a workload to Kubernetes. While evaluating the approach, you run only a portion of your backends in Kubernetes.



SERVICE WITHOUT SELECTOR

Services most commonly abstract access to Kubernetes Pods, but they can also abstract other kinds of backends. For example:

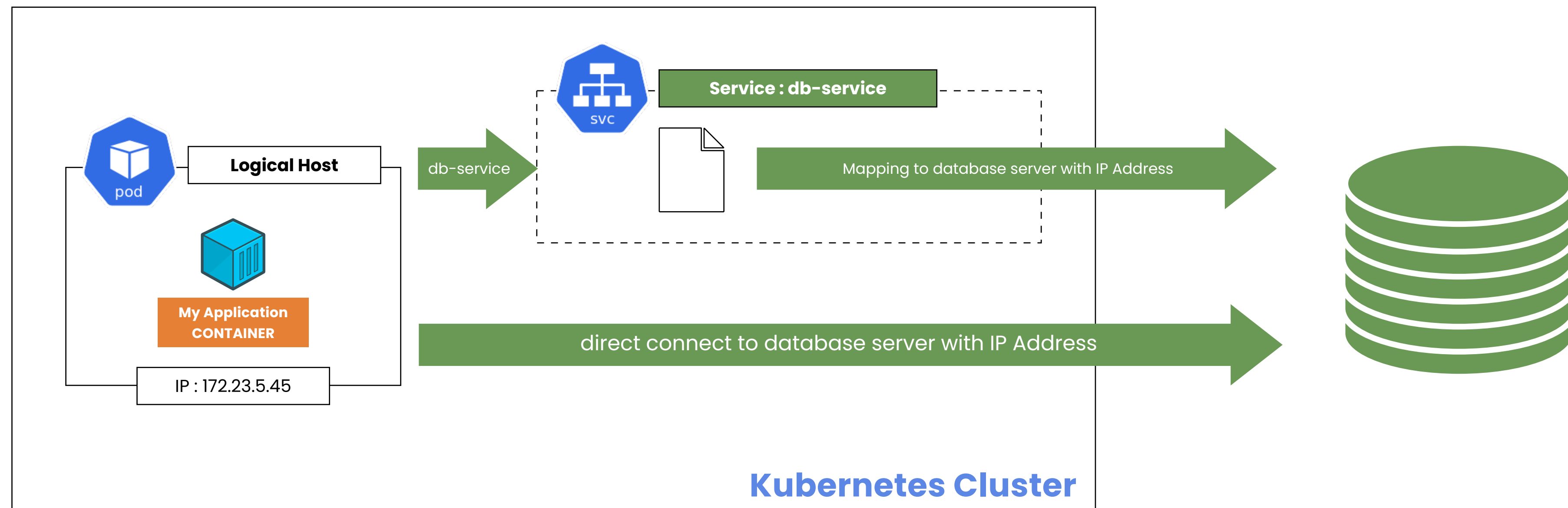
- You want to have an external database cluster in production, but in your test environment you use your own databases.
- You want to point your Service to a Service in a different Namespace or on another cluster.
- You are migrating a workload to Kubernetes. While evaluating the approach, you run only a portion of your backends in Kubernetes.



SERVICE WITHOUT SELECTOR

Services most commonly abstract access to Kubernetes Pods, but they can also abstract other kinds of backends. For example:

- You want to have an external database cluster in production, but in your test environment you use your own databases.
- You want to point your Service to a Service in a different Namespace or on another cluster.
- You are migrating a workload to Kubernetes. While evaluating the approach, you run only a portion of your backends in Kubernetes.



THANK YOU