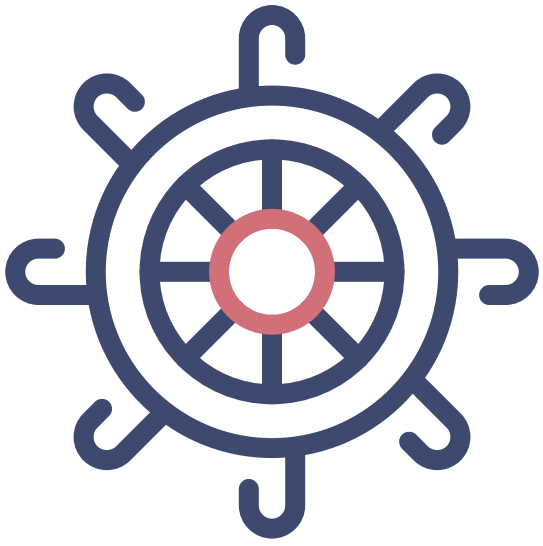
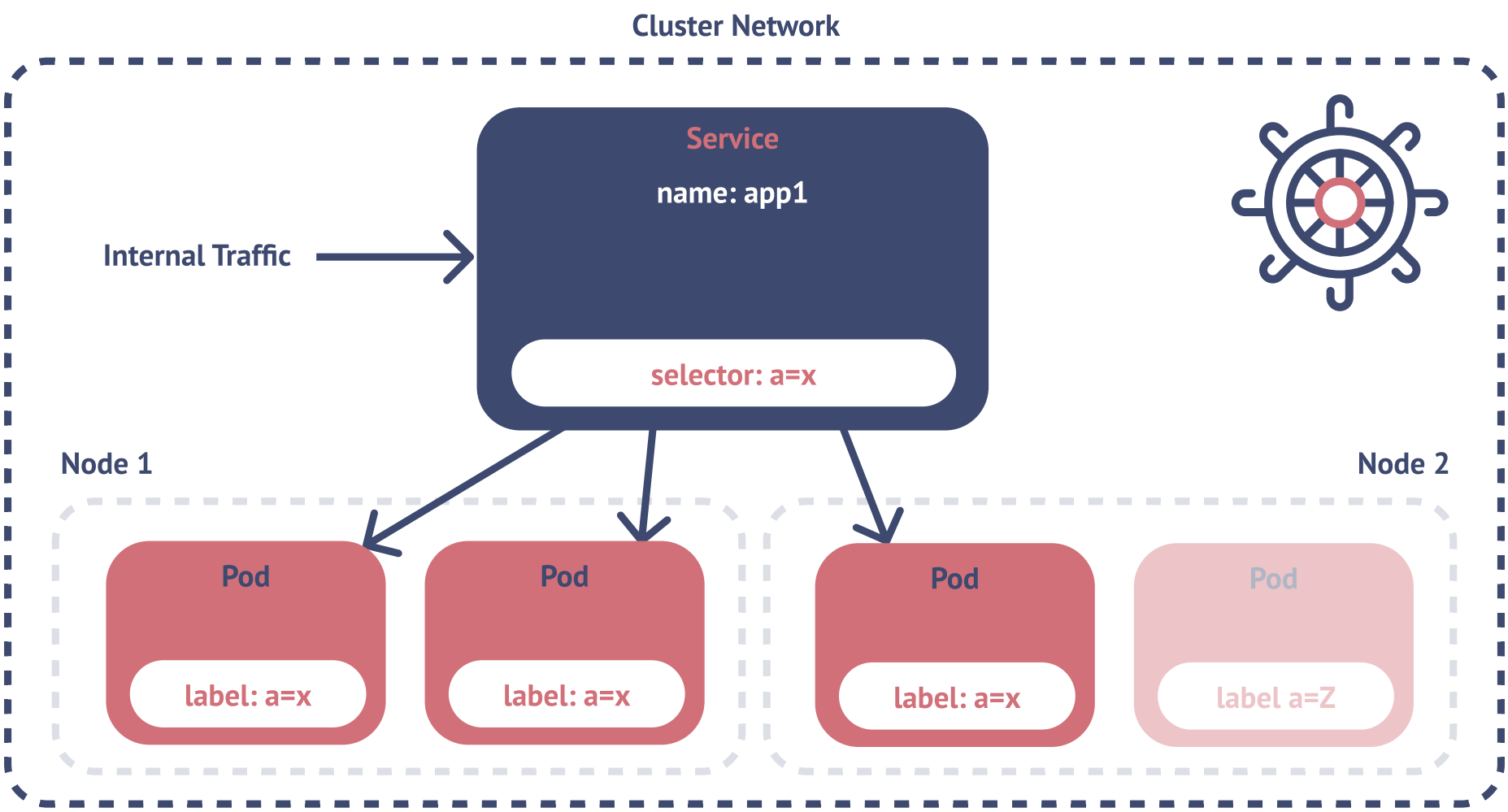


Kubernetes Services *Explained*

Service

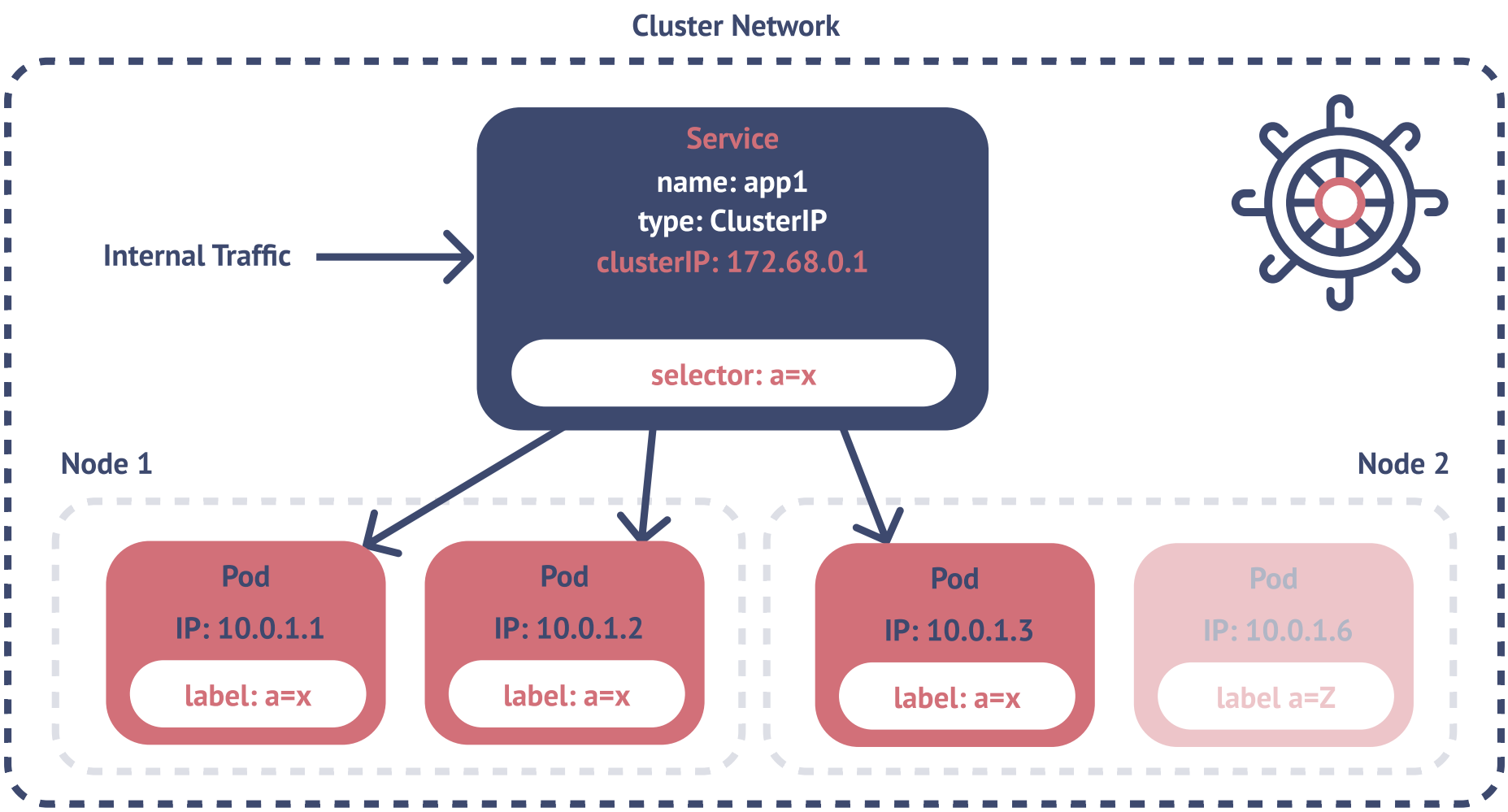


> *Service* object in Kubernetes hides all the complexity of networking behind a simple **metadata matching mechanism**.

Elements of computing (*Pods*) have to have the **same labels** as the selector on the Service.

The iptables rules, endpoints objects, or *Pod* migrations will be handled automatically to ensure that traffic that hits the *Service* will be routed and load-balanced.

ClusterIP

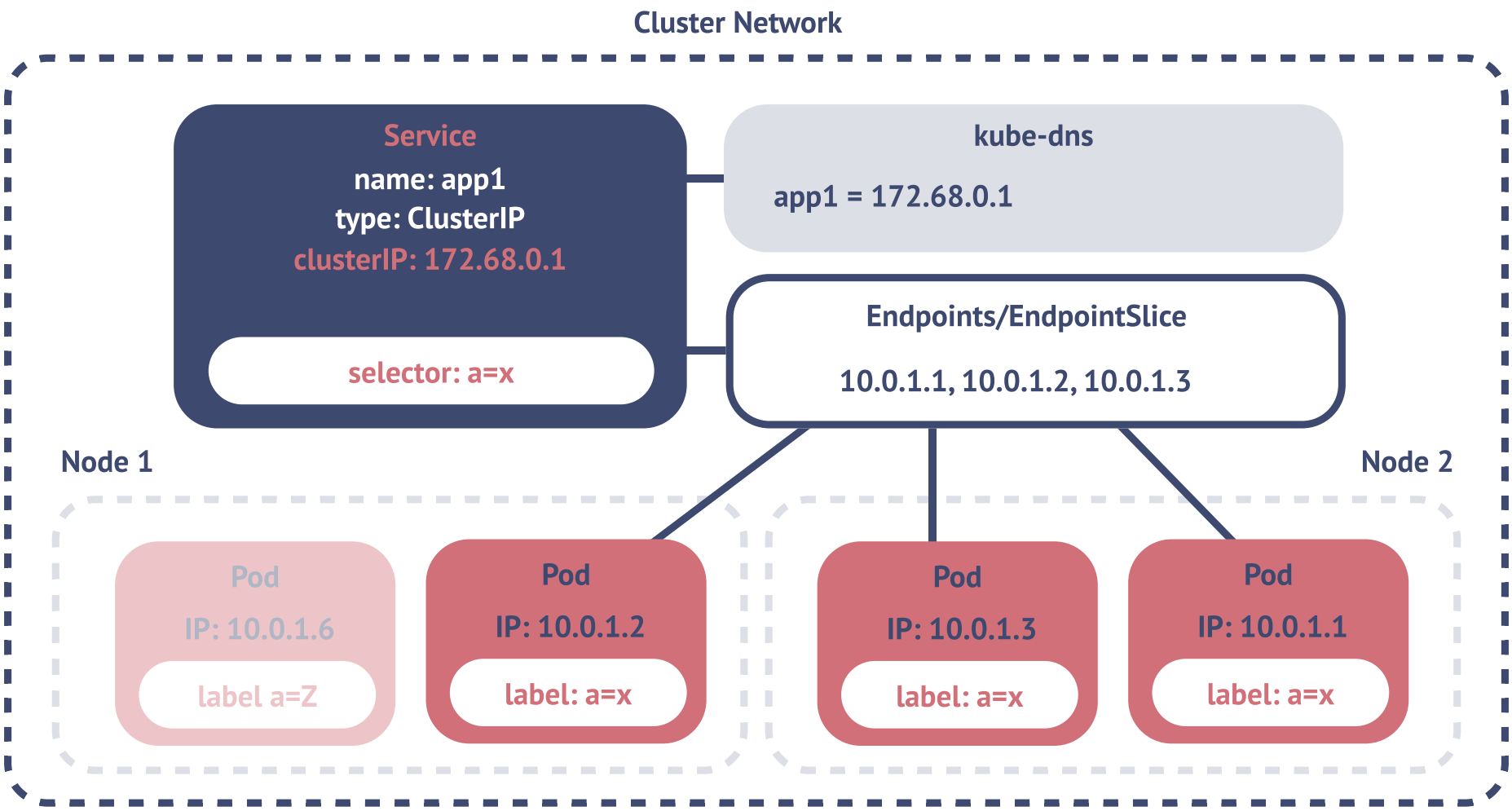


Each *Pod* has its own IP address within the cluster network. But that address is not static. It will most likely change when *Pod* is recreated.

The simplest (and default) *Service*'s type is *ClusterIP*. Each ***Service*** **receives its own IP address** that remains the same as long as the *Service* objects exists.

In order to reach application running in a set of *Pod*s, the client can just call the *Service* using its IP address or DNS name.

ClusterIP

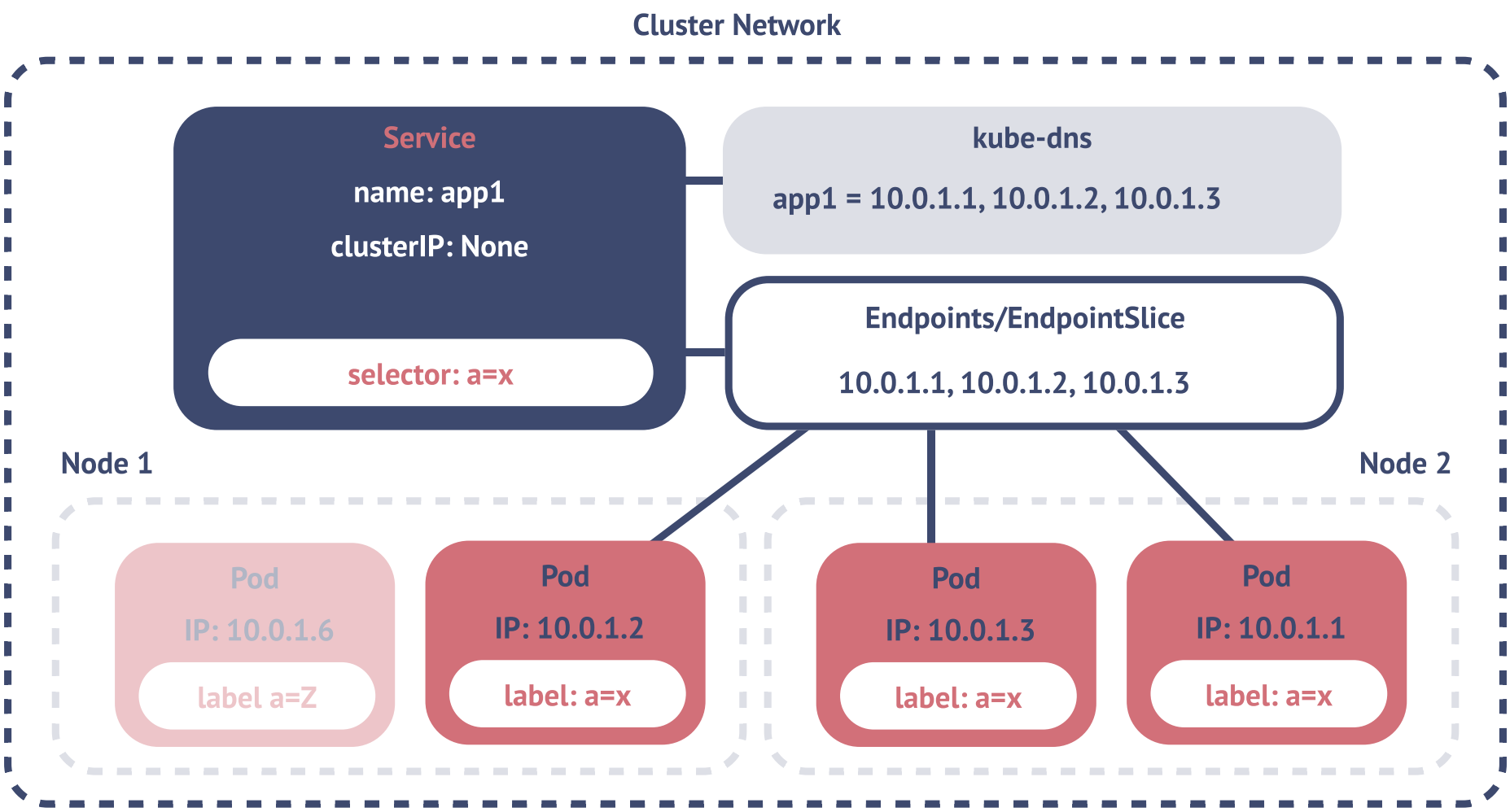


> Kubernetes creates additional *Endpoints* or *EndpointSlice* (depending on the cluster version) objects to support the mapping between *Service* and *Pod*s. Those objects are created automatically whenever the *Pod*s with matching labels are created or deleted.

In addition, Kubernetes binds *Service*'s *IP* address to a DNS name, which matches the name of the *Service e.g.*:

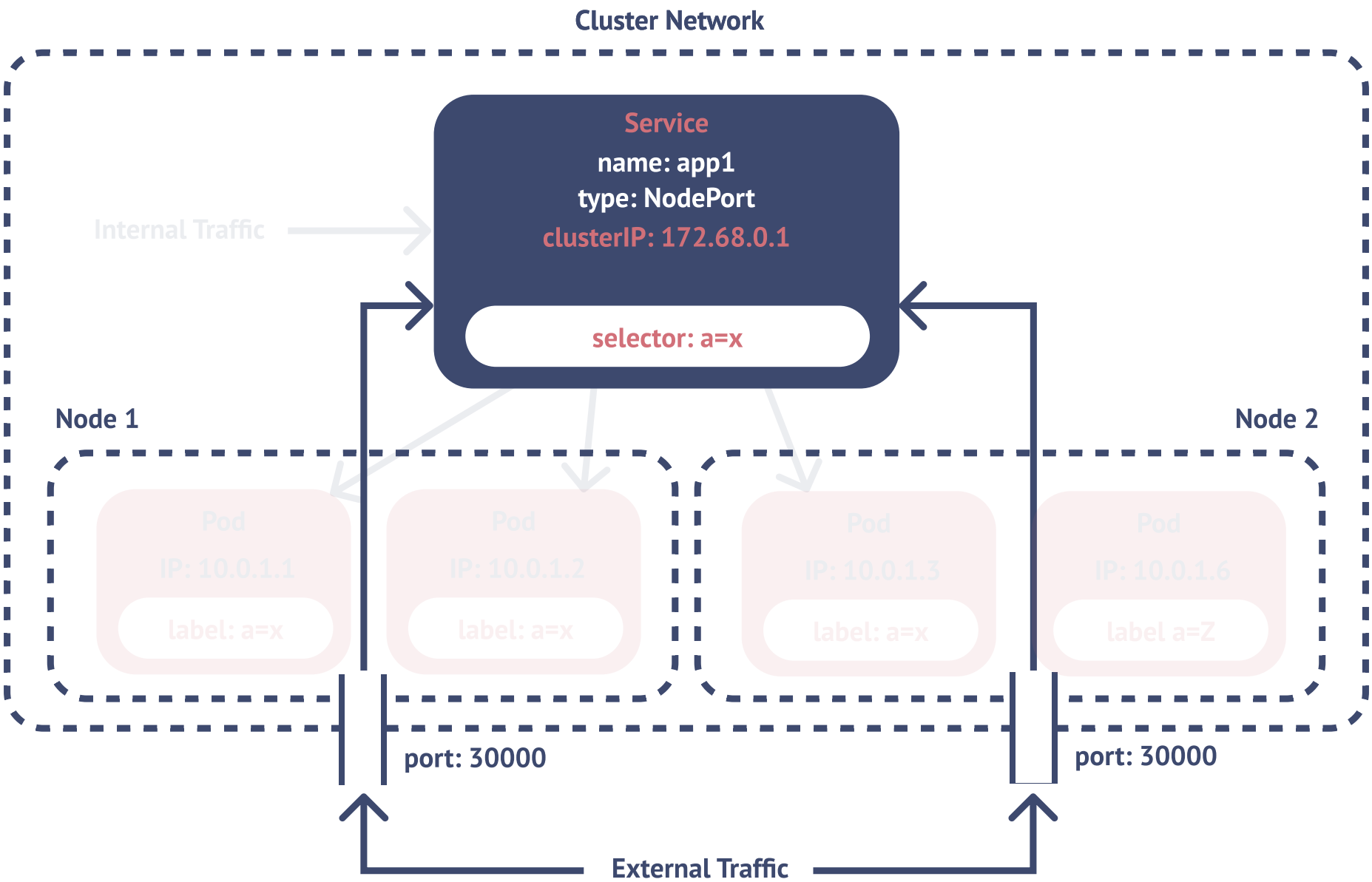
```
app1.namespace.service.cluster.local
```

Headless



> If *clusterIP* field is explicitly set to *None*, then *Service* turns into a so called *Headless Service*. The traffic routing works in the same way through *Endpoint/EndpointSlice* objects, but DNS record is populated with the list of *Pod* IP addresses, which are dynamically updated.
>
> This may be needed for stateful protocols when connection between client and backend has to be preserved.

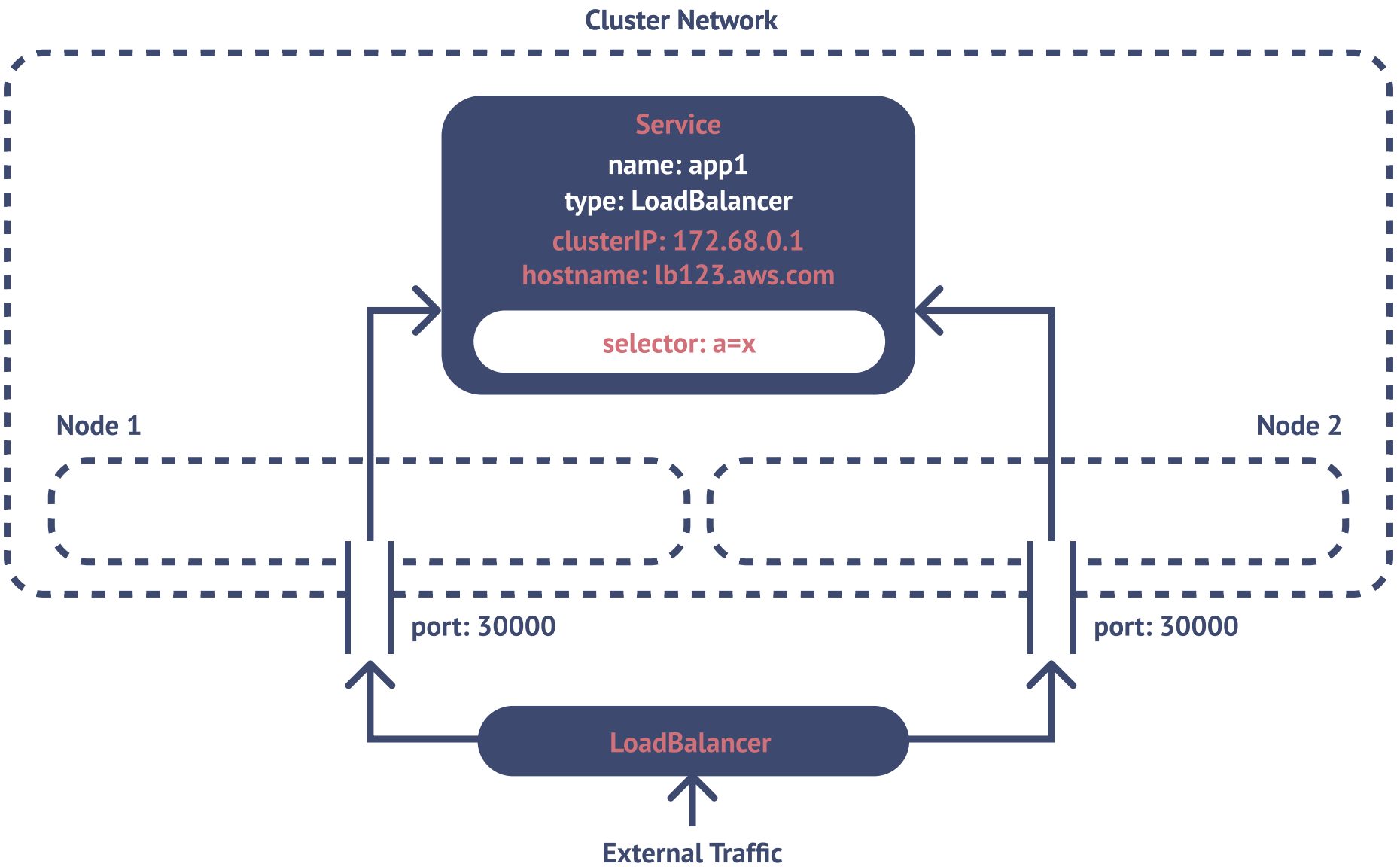
NodePort



NodePort is the service type that has all the attributes of *ClusterIP*, but in addition, Kubernetes opens a random port (in the range of 30000-32767) on every *Node*.

This is needed to get traffic into the cluster from external sources. External client can contact any cluster *Node* and traffic will be routed to the *Pod*s.

LoadBalancer

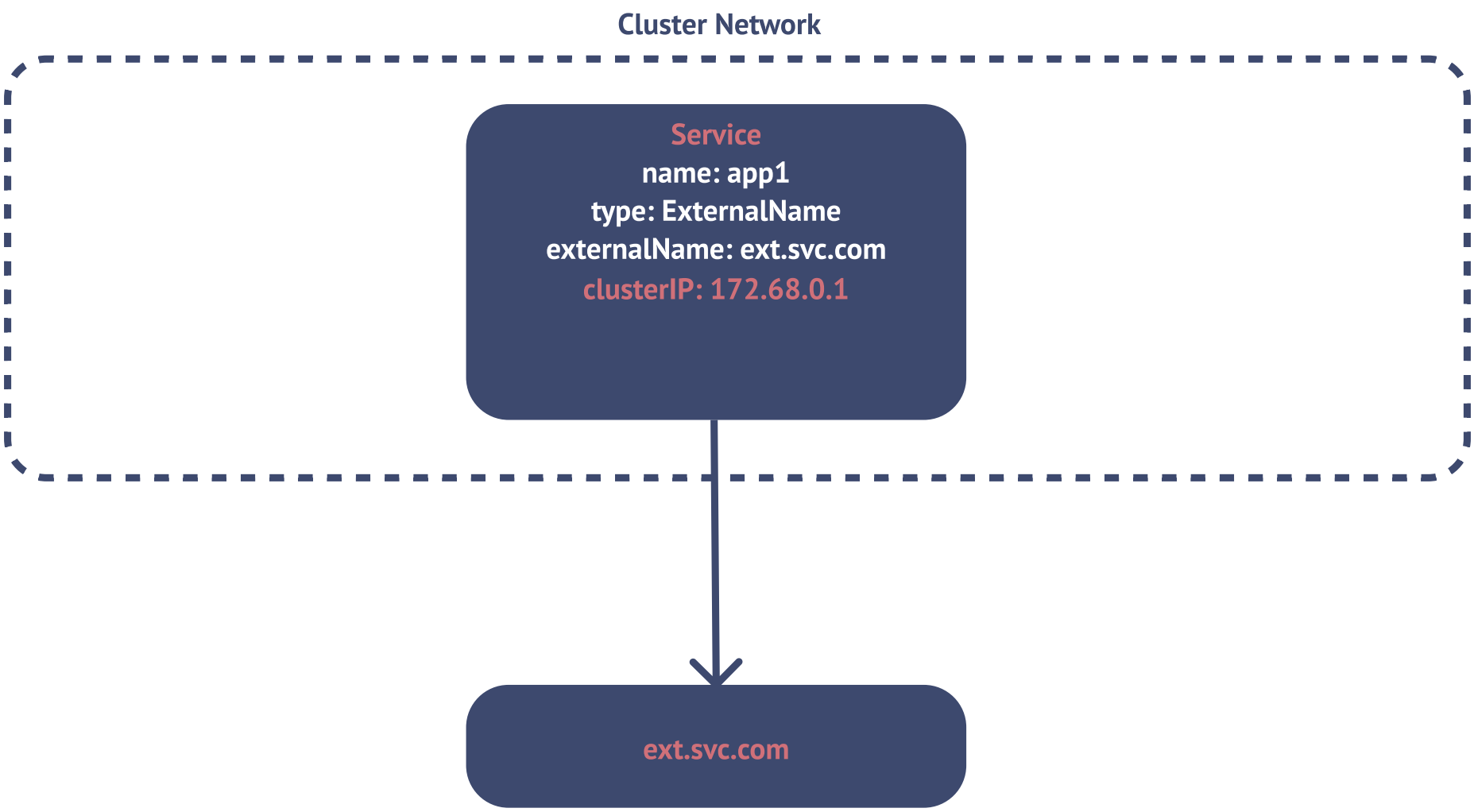


> *LoadBalancer* is the service type that has all the attributes of *NodePort,* but in addition, it provisions external load balancer that automatically balances the traffic between all *Node*s and exposed port.

The type of load balancer depends on the cluster configuration.

For example, if the cluster is running on AWS EKS, then AWS ELB's application load balancer will most likely be provisioned for *Services* of type *LoadBalancer*.

ExternalName



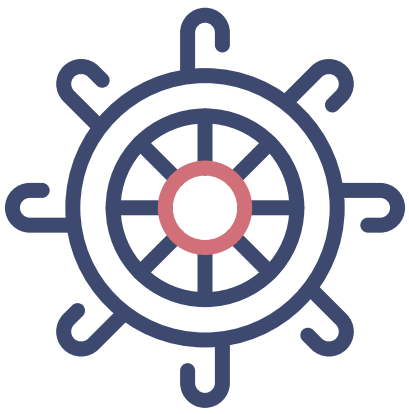
> *Service*s of type *ExternalName* map a *Service* to a DNS name.
>
> This allows to use the same mechanism for discovering internal and external services within the cluster.

Join

Real-life Kubernetes

2-day online workshop

Architecture and Deployment Models



Learn more @

extremeautomation.io