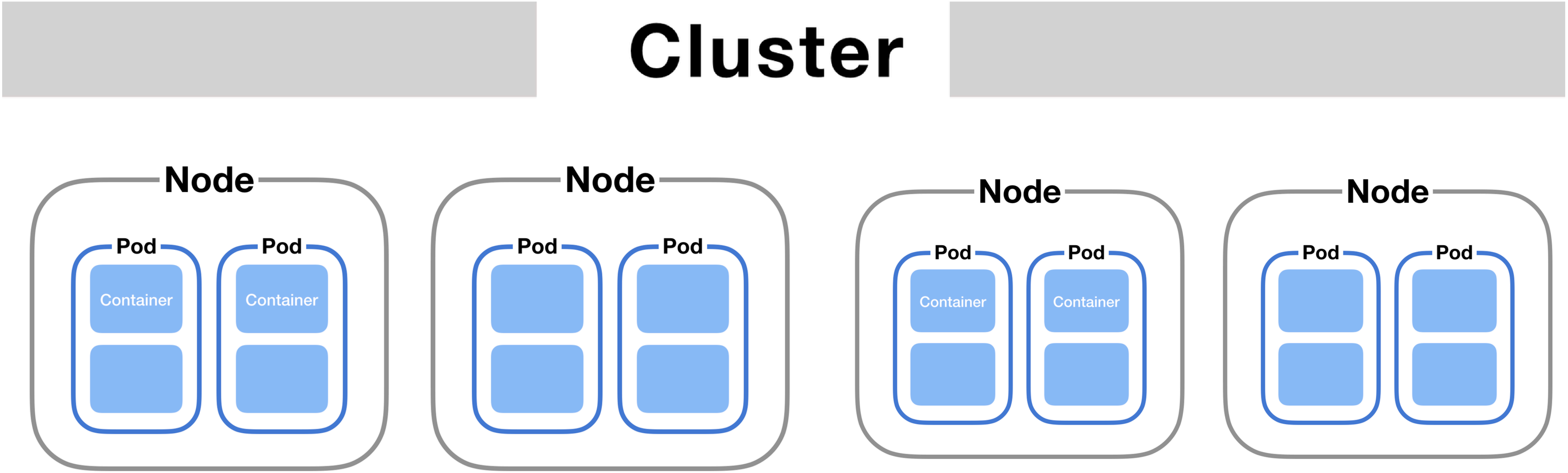
STATELESS APPLICATION

Last night i had a call with mobile customer care, and i explained my issue with her and requested to resolve the issue. She asked to stay on call for some time. Meanwhile the call was dropped due to network issues

Again i called to same number, but this time another person pick the call. He don't know what the conversation we had with the first person. Again i had to repeat the details of my problem to the second person, though it is an inconvenience but it still works out

Kubernetes Deployment suits perfectly here. Let’s assume you deployed your stateless application as a Deployment with 10 Pod replicas running in multiple worker nodes.



If one of those Pods running in a particular worker node got terminated due to some issue

The ReplicaSet Controller takes care of replacing the bad Pod with a new healthy Pod either on the same node or on a different node.

Here replicaSet's will only take care of those 10 pods are running in the cluster or not.
It doesn't take care of in which worker node they are running!

If pod 1 got terminated from node 1, then new pod will gets replaced on either node 2 or node 1 or anywhere in the cluster.

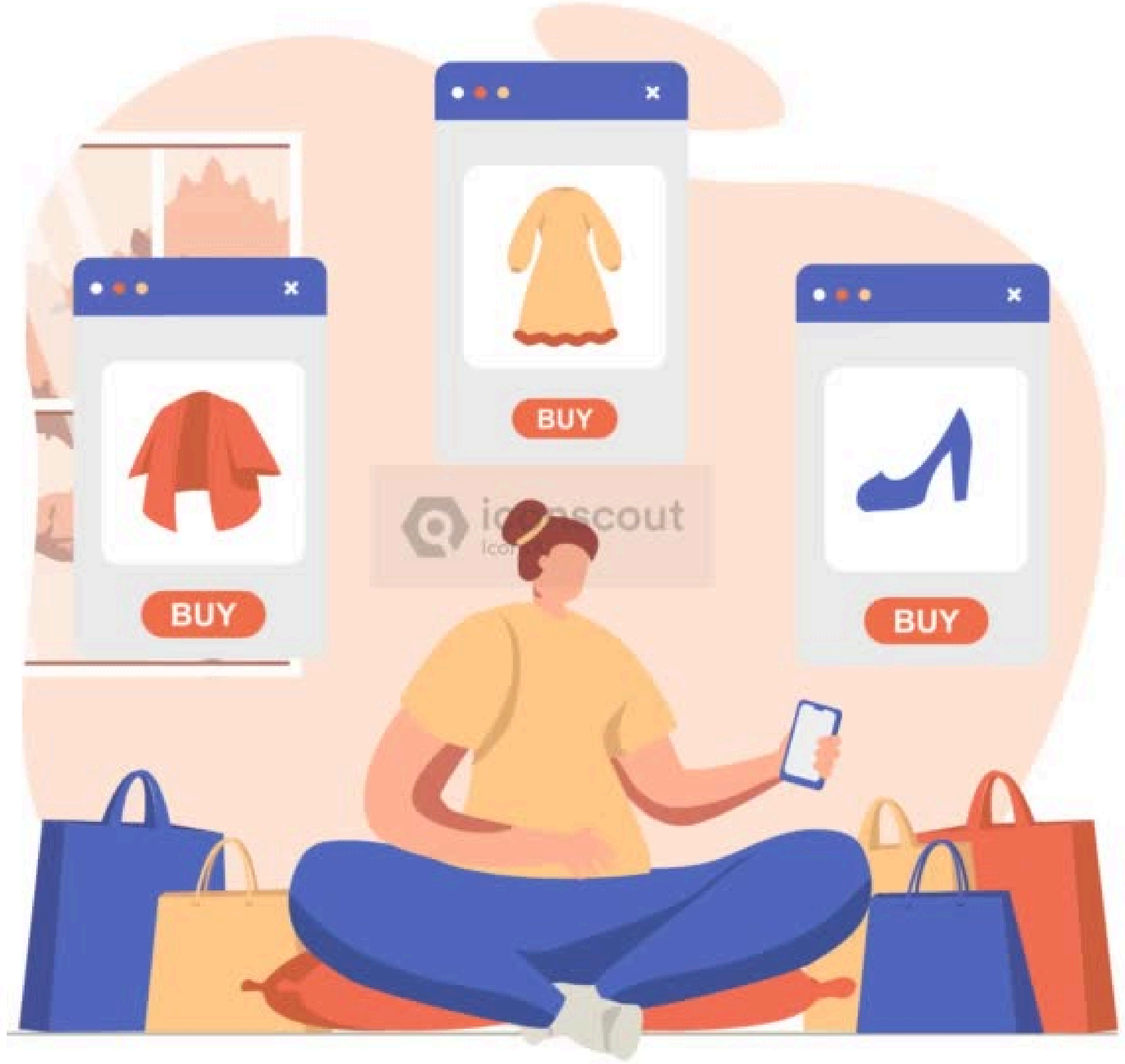
This is possible because, the stateless application Pod 1 hasn’t stored any data on worker node 1, hence can be easily replaced by new pod running on a different worker node 2

If the pod is not storing the data then we called it as **STATELESS**

STATELESS APPLICATION:

- It can't store the data permanently.
- The word STATELESS means no past data.
- It depends on non-persistent data means data is removed when Pod, Node or Cluster is stopped.
- Non-persistent mainly used for logging info (ex: system log, container log etc..)
- In order to avoid this problem, we are using stateful application.
- A stateless application can be deployed as a set of identical replicas, and each replica can handle incoming requests independently without the need to coordinate with other replicas.

Let’s say you want to order some some products in Amazon, you added some products to your shopping cart. After you added the book(s) to your shopping cart, If forgot to switch off the stove in kitchen and you closed the web browser hurry.



Well, you can anytime re-login to amazon portal and resume from where you left last time, as the amazon portal will ensure all items added to the Shopping cart are persisted in database.

Technically, if Amazon is using a ShoppingCart that runs in a Pod,
When i added some products on my cart that will stores on one pod and after some time when i resumed back to the amazon portal that will shows all the products from the different pod.

The interesting thing to understand here is, the StatefulSet Controller managed to bind the exact same Persistent Volume to two Shopping Cart Pods associated to that customer at two different point of time.

STATEFUL APPLICATION:

- Stateful applications are applications that store data and keep tracking it.

Example of stateful applications:

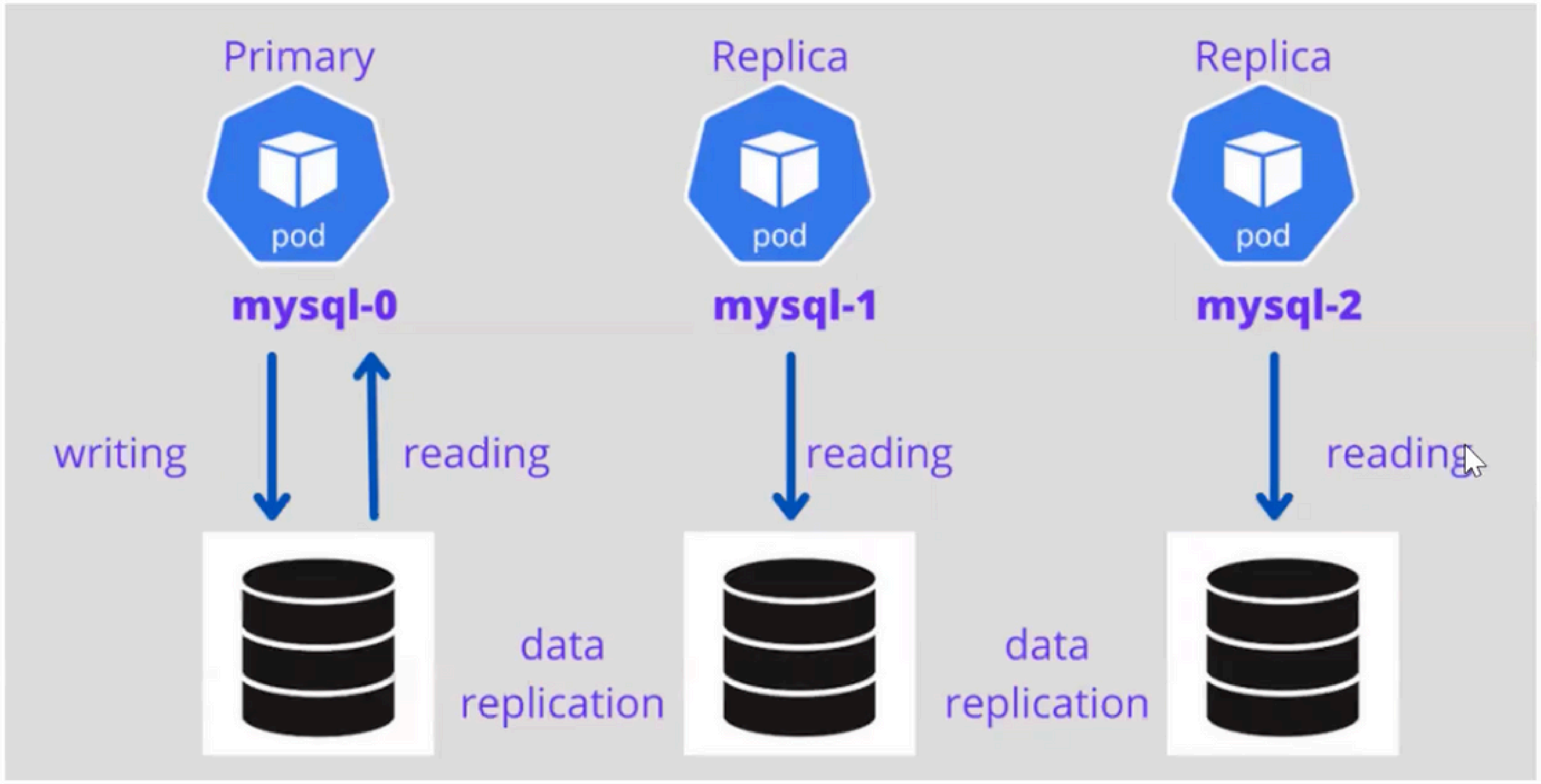
- All RDS databases (MySQL, SQL)
- Elastic search, Kafka , Mongo DB, Redis etc...
- Any applicaiton that stores data

To get the code for stateful application use this link:
https://github.com/devops0014/k8s-stateful-set-application.git

STATEFUL SET

- A StatefulSet is the Kubernetes controller used to run the stateful application as containers (Pods) in the Kubernetes cluster.
- StatefulSets assign a sticky identity-an orginal number starting from zero-to each Pod instead of assigning random IDs for each replica Pod.
- A new Pod is created by cloning the previous Pod's data. If the previous Pod is in the pending state, then the new Pod will not be created.
- If you delete a Pod, it will delete the Pod in reverse order, not in random order.

StatefulSet



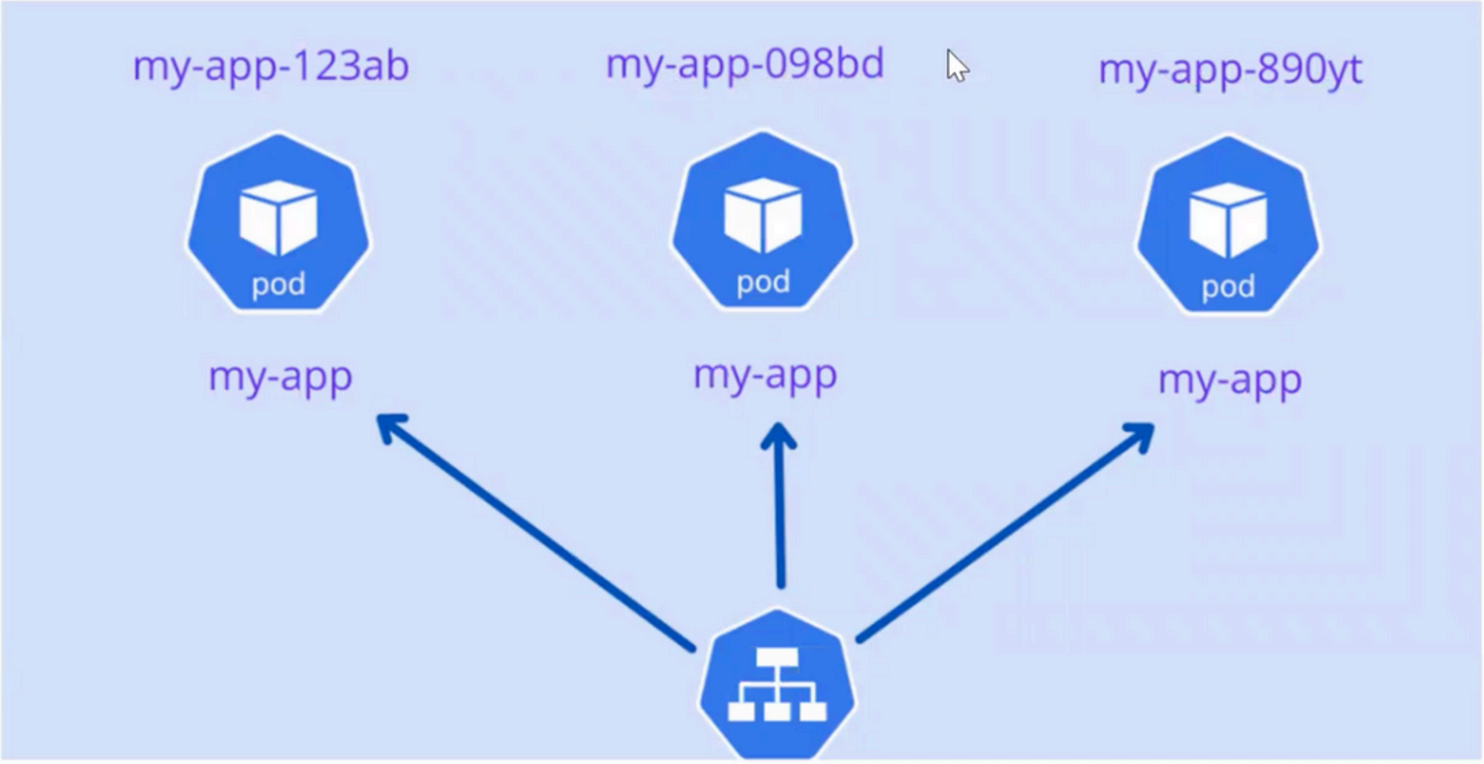
- Lets assume we deployed 3 replicas in a node,
- 1st pod is primary pod, remaining pods are secondary pods.
- Primary pod is having both read and write access
- But secondary pod is having only read access
- If we insert some data in primary pod, we can access the data from any pod i.e.. pod will share the data each other

DEPLOYMENT VS STATEFUL SET

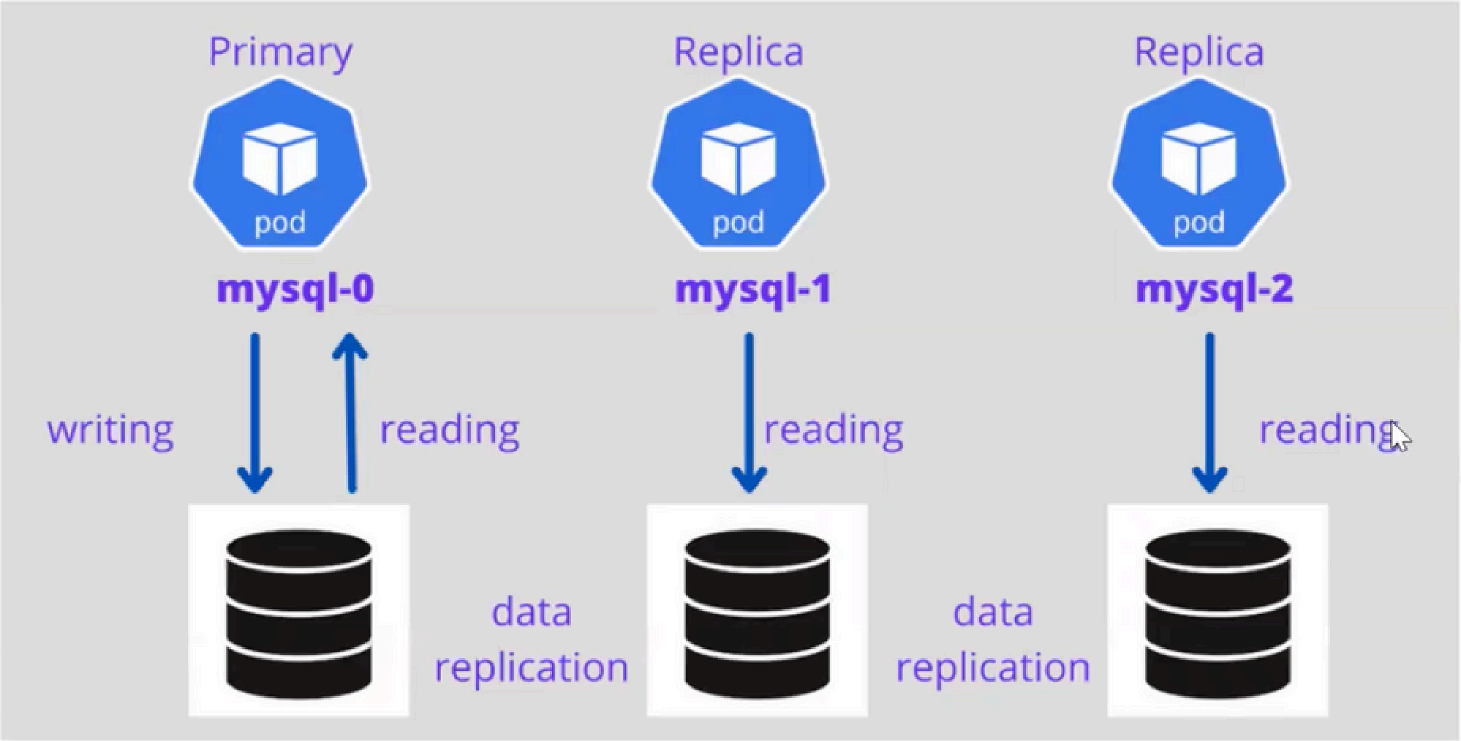
deployment cannot give the numbers or index for resources
but the stateful set will give the sticky identities called index numbers

when we delete pods in stateful set last pod will be deleted first
and primary pod deletes last

Deployment



StatefulSet



DEPLOYMENT	STATEFUL SET
• It will create POD's with random ID's	• It will create POD's with sticky ID's
• Scale down the POD's in random ID's	• Scale down the POD's in reverse order
• POD's are stateless POD's	• POD's are stateful POD's
• We use this for application deployment	• We use this for database deployment