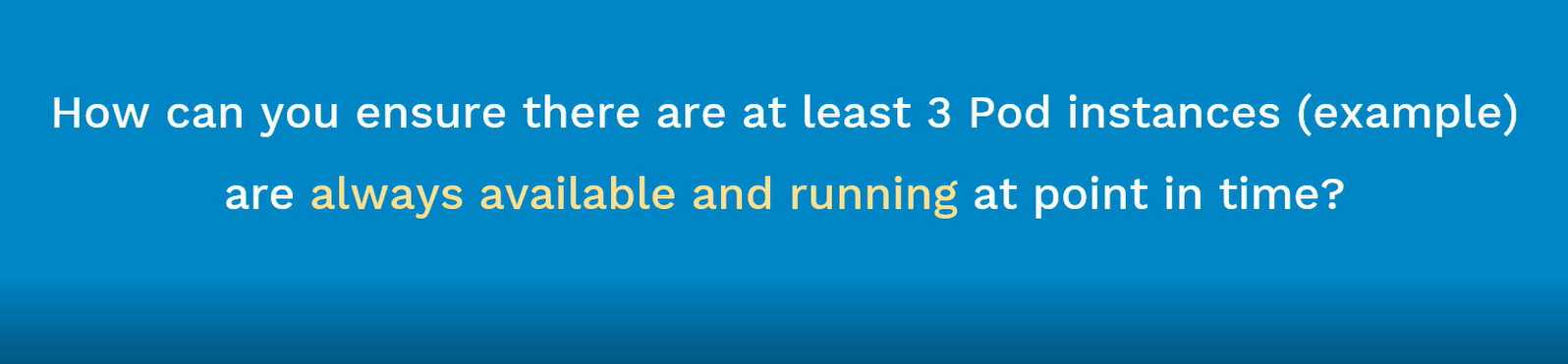
**Replication Controller**

Let’s imagine that we want to deploy containerized app inside kubernetes cluster.

Ex:

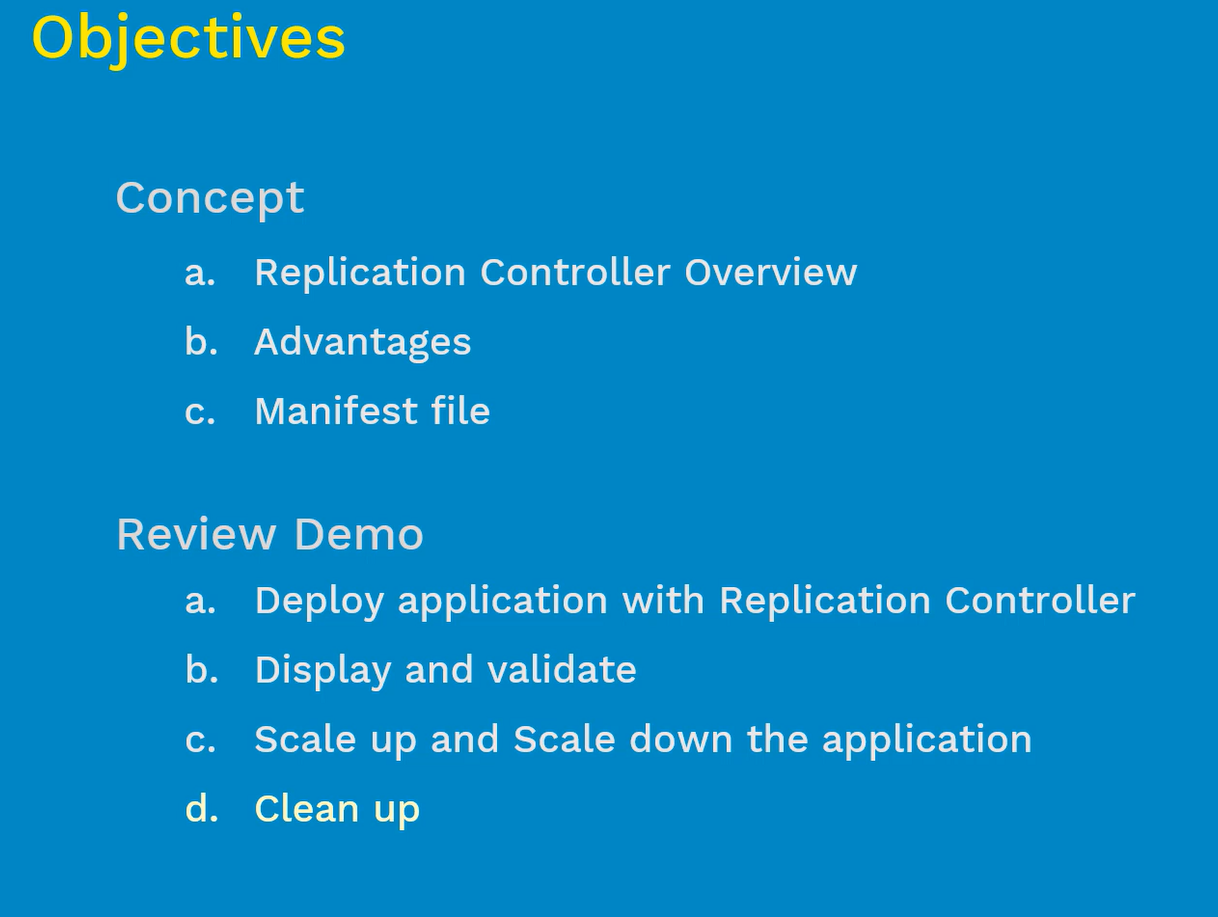
**How can we ensure there are at least 3 pod Instances are always available and running at point in time?**

**Replication Controller**



What is Replication Controller? What are its advantages? (Before learning this one, we need to have an idea of kubernetes, pods and kubectl)

Objectives:



# Replication Controller Overview:

* It ensures that specified number of pods are running at any time(always) – { Eg: If we define 5 pods in replication controller, then this controller make sure 5 pods will running always}

1. If there are excess pods, they get killed and vice-versa – (In case, there are too many pods are running, then what was defined in the manifest file? Then replication controller will terminate the extra pods)
2. New Pods are launched when they get fail, get deleted or terminated – ( If pods were less than the defined in the manifest file, the replication controller will start more pods, sometimes pods do fail, deleted or die for various reasons- when they do, if they were back by the replication controller then these pods will get automatically re-created }
3. The best practice and good idea for the replication controller make the value as 1.

One very important point note here is,

**How this replication controller does aware of the pods that need to manage?**

**What is the relation between the replication controller and pods need to manage? Answer is ‘Labels’**

* Replication Controllers and Pods are associated with “Labels” – { Labels are nothing but “tags” that given to the PODS, Using Labels that we will relate the replication controller to pods that its need manage, here we will mention the exact labels inside the pod and the replication controller }
* Creating a “rc” with count of 1 ensure that a pod is always available – { It is often abbreviated as “rc” or “rcs” }

