KUBERNETES - REPLICATIONS

Github repo..

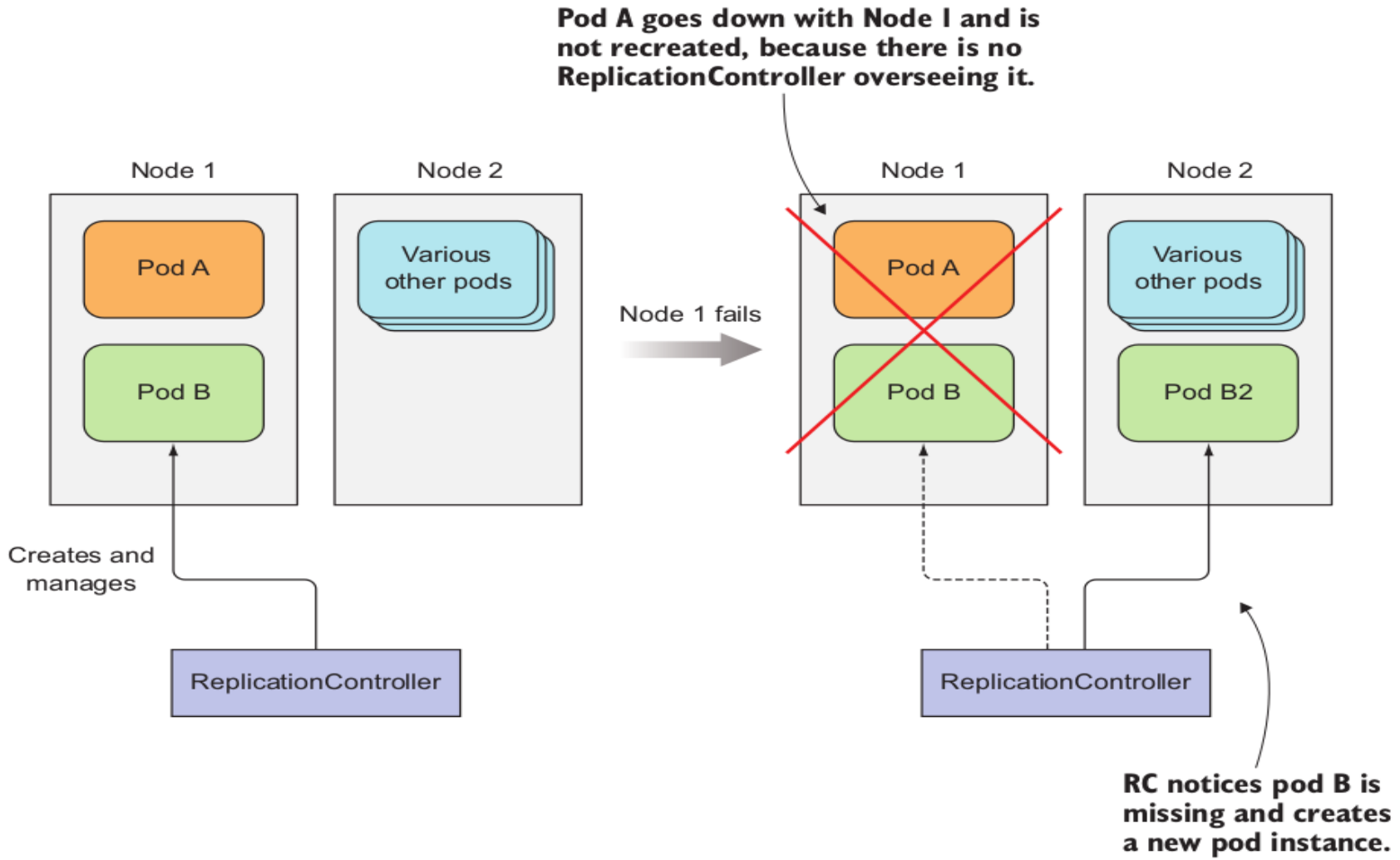
https://github.com/vsaini44/KubernetesRepo.git

What is Replication Controller ?

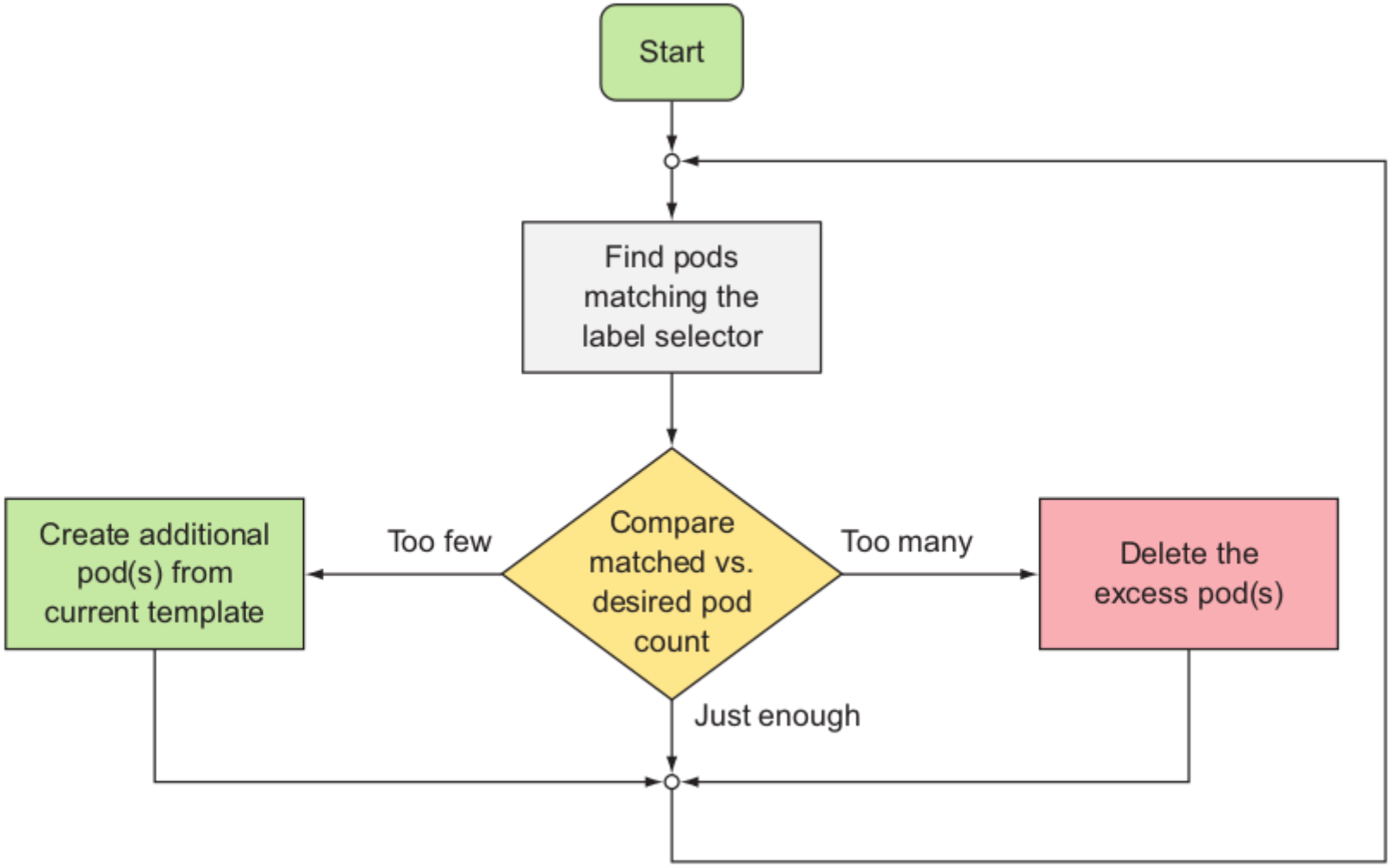
A ReplicationController is a Kubernetes resource that ensures its pods are always kept running.

If the pod disappears for any reason, such as in the event of a node disappearing from the cluster or because the pod was evicted from the node, the ReplicationController notices the missing pod and creates a replacement pod.

RC ?


Pod A goes down with Node I and is not recreated, because there is no ReplicationController overseeing it.
Node 1
Node 2
Pod A
Pod B
Various other pods
Node 1 fails
Node 1
Node 2
Pod A
Pod B
Various other pods
Pod B2
Creates and manages
ReplicationController
ReplicationController
RC notices pod B is missing and creates a new pod instance.

RC LifeCycle



RC Anatomy

A ReplicationController has three essential parts:

- A label selector, which determines what pods are in the ReplicationController's scope
- A replica count, which specifies the desired number of pods that should be running
- A pod template, which is used when creating new pod replicas

Manual Scaling of Pod

Your ReplicationController has been keeping three instances of your pod running.

You’re going to scale that number up to 10 now.

$ kubectl scale rc kubia --replicas=10

Replica Set

A ReplicaSet behaves exactly like a ReplicationController, but it has more expressive pod selectors.

Whereas a ReplicationController's label selector only allows matching pods that include a certain label, a ReplicaSet's selector also allows matching pods that lack a certain label or pods that include a certain label key, regardless of its value.

Replica Set Label Selector

- **In— Label’s value must match one of the specified values .**
- **NotIn— Label’s value must not match any of the specified values .**
- **Exists— Pod must include a label with the specified key (the value isn’t important). When using this operator, you shouldn’t specify the values field.**
- **DoesNotExist— Pod must not incl**

DaemonSets

To run a pod on all cluster nodes, you create a DaemonSet object, which is much like a ReplicationController or a ReplicaSet, except that pods created by a Daemon- Set already have a target node specified and skip the Kubernetes Scheduler.

They aren’t scattered around the cluster randomly.

A DaemonSet makes sure it creates as many pods as there are nodes and deploys each one on its own node

Summary

- **Understanding Need for Replication Controller**
- **RC Working**
- **Replica’s Set**
- **Daemon Set**