# Sequential Breakdown of the Process

In this lesson, we will look into the occurrence of sequential events to create replicas of a Pod.

## Sequential Breakdown of the Process #

The sequence of events that transpired with the `kubectl create -f rs/go-demo-2.yml` command is as follows.

1. Kubernetes client (`kubectl`) sent a request to the API server requesting the creation of a ReplicaSet defined in the `rs/go-demo-2.yml` file.

2. The controller is watching the API server for new events, and it detected that there is a new ReplicaSet object.

3. The controller creates two new pod definitions because we have configured replica value as 2 in `rs/go-demo-2.yml` file.

4. Since the scheduler is watching the API server for new events, it detected that there are two unassigned Pods.

5. The scheduler decided which node to assign the Pod and sent that information to the API server.

6. Kubelet is also watching the API server. It detected that the two Pods were assigned to the node it is running on.

7. Kubelet sent requests to Docker requesting the creation of the containers that form the Pod. In our case, the Pod defines two containers based on the `mongo` and `api` image. So in total four containers are created.

8. Finally, Kubelet sent a request to the API server notifying it that the Pods

were created successfully.
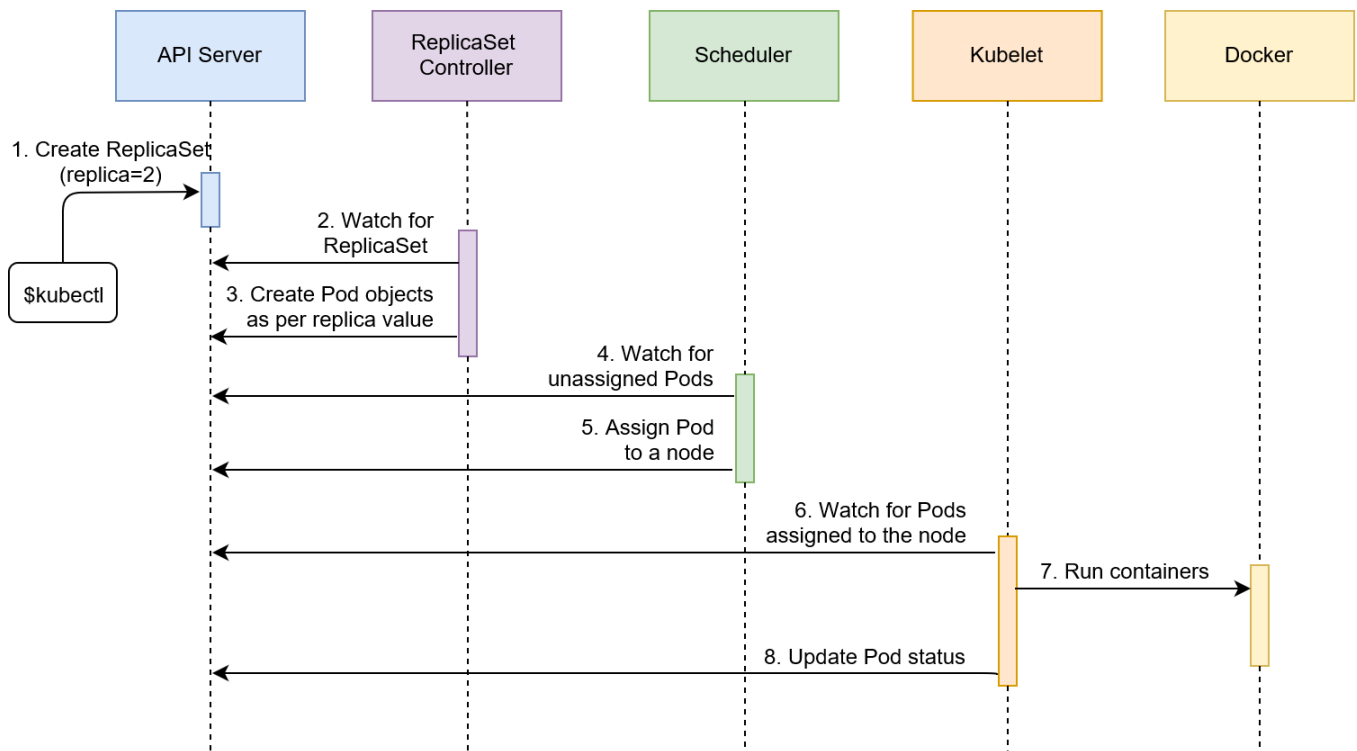


The sequence of events followed by request to create a ReplicaSet

The sequence we described is useful when we want to understand everything that happened in the cluster from the moment we requested the creation of a new ReplicaSet. However, it might be too confusing so we'll try to explain the same process through a diagram that more closely represents the cluster.
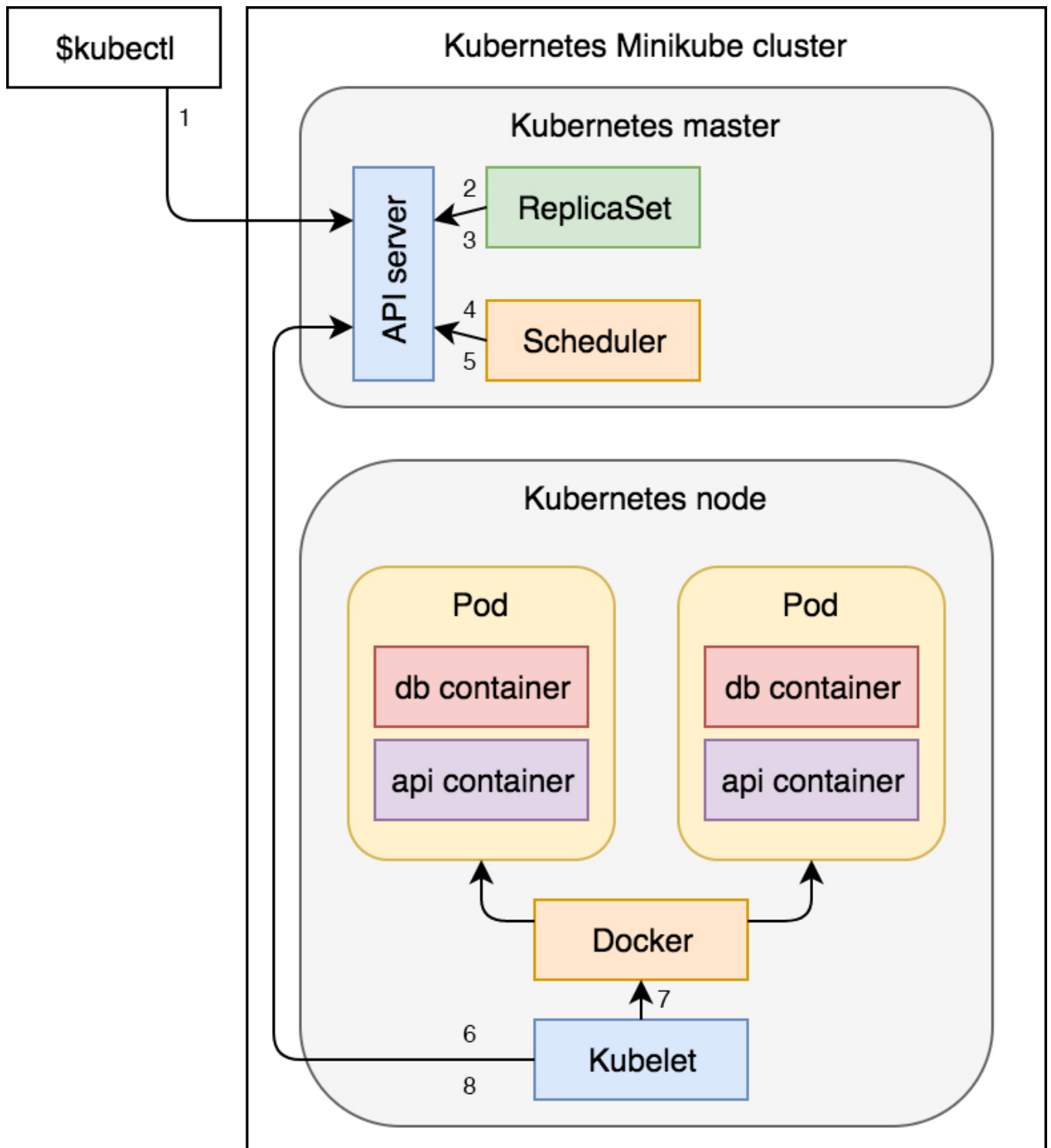
The events followed by request to create a ReplicaSet

📝 Typically, we'd have a multi-node cluster, and the Pods would be distributed across it. For now, while we're using Minikube, there's only one server that acts as both the master and the node. Later on, when we start working on multi-node clusters, the distribution of Pods will become evident. The same can be said for the architecture. We'll explain different Kubernetes components in more detail later on.

In the next lesson, let's see which types of operations we can perform on ReplicaSets.