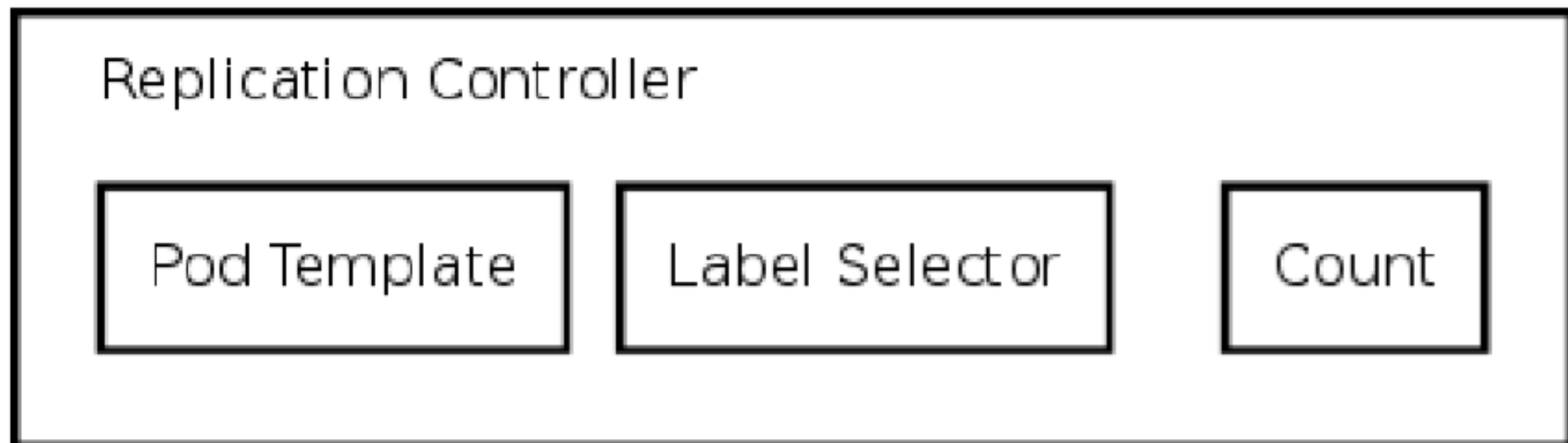


# Replication Controller

- Consists of
  - Pod template
  - Count
  - Label Selector
- Kube will try to keep \$count copies of pods matching the label selector running
- If too few copies are running the replication controller will start a new pod somewhere in the cluster

# Replication Controller



# Replication Controller

- Ensures that a Pod or homogeneous set of Pods are always up and available
- Always maintains desired number of Pods
  - If there are excess Pods, they get killed
  - New pods are launched when they fail, get deleted, or terminated
- Creating a replication controller with a count of 1 ensures that a Pod is always available
- Replication Controller and Pods are associated through Labels

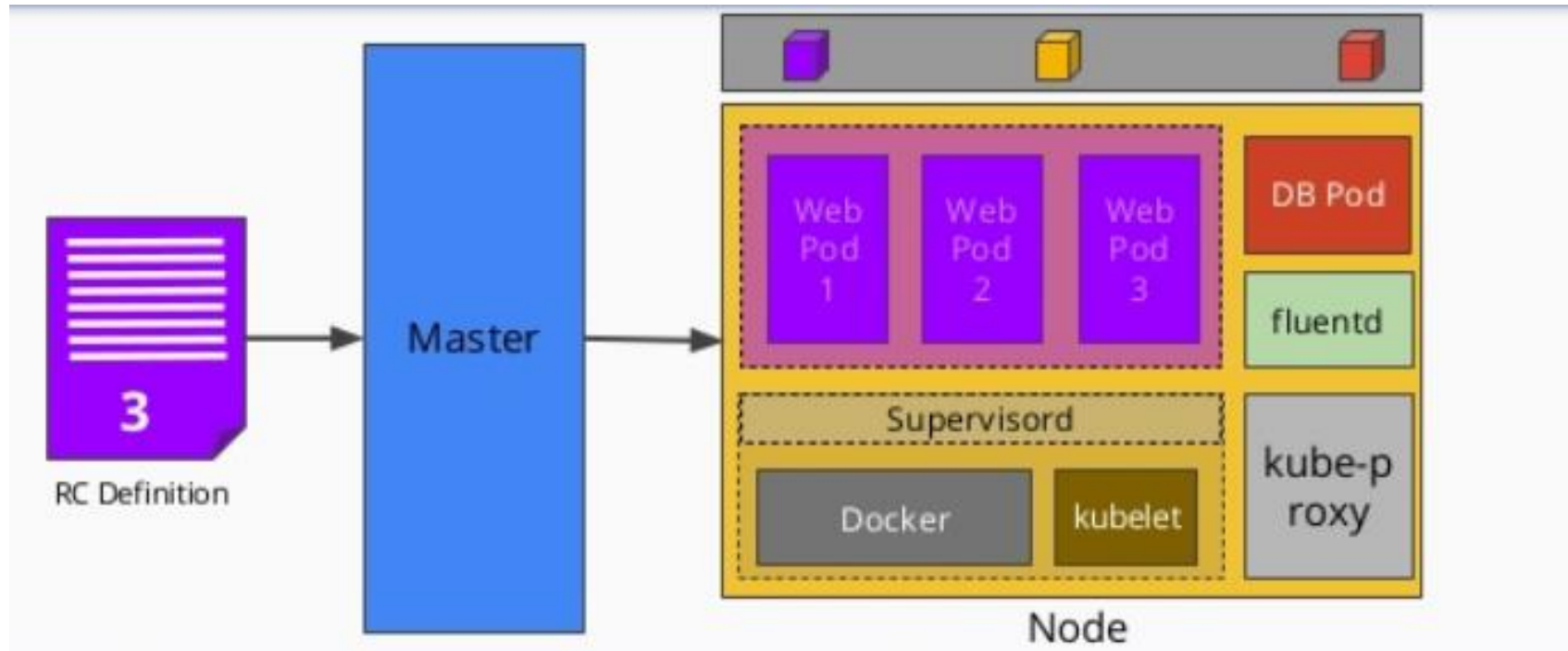
# Replica Set

- **Replica set** is the next-gen Replication Controller
- It supports a new selector that can do the selection based on **filtering** according to a **set of values**
  - Example: Environment "Dev" or "QA"
  - not only based on equality, like replication controller
    - e.g "Environment" == "Dev"
- This **Replica Set**, rather than the replication controller, is used by the Deployment Object

# Replica Set

- Replica Sets are the next generation Replication Controllers
- Ensures specified number of pods are always running
- Pods are replaced by Replica Sets when a failure occurs
  - New pods are automatically scheduled
- Labels and Selectors are used for associating Pods with Replica Sets
- Usually combined with Pods when defining the deployment

# Scaling Pods with Replication Controller



# Demo

- Scaling Pods with Replication Controller