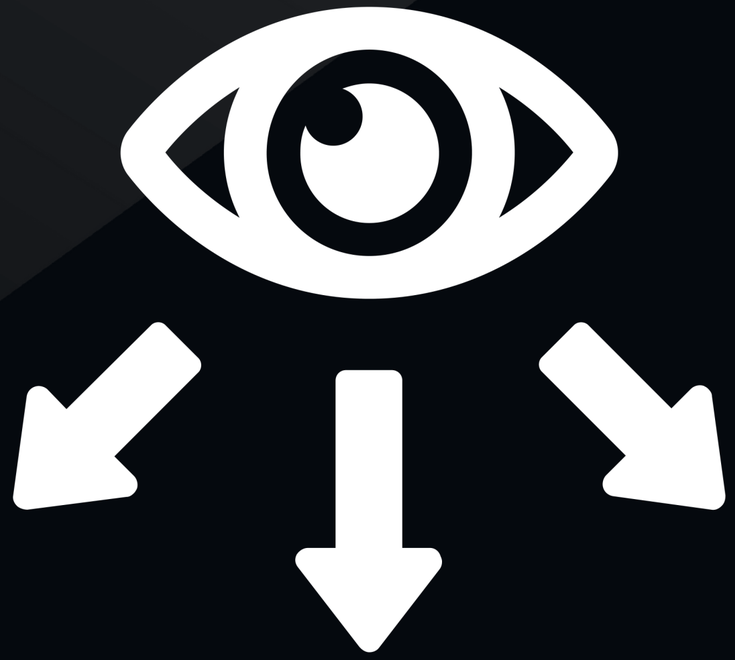
CONTROLLER

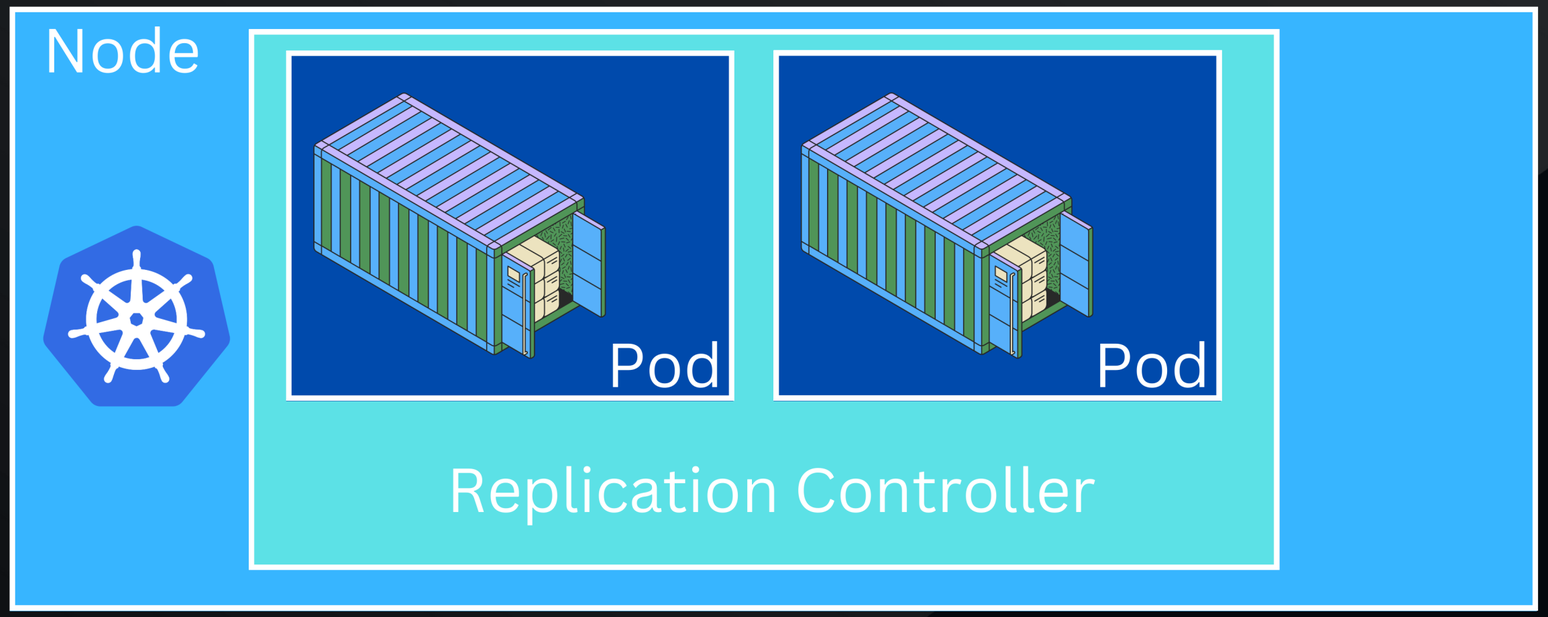
- A controller is a control loop that **watches the state of the system and makes changes to move the system towards the desired state**. Controllers in Kubernetes are responsible for maintaining the desired state of the system, such as ensuring that the **correct number of replicas of a pod** are running or that a **service has the correct endpoints**.



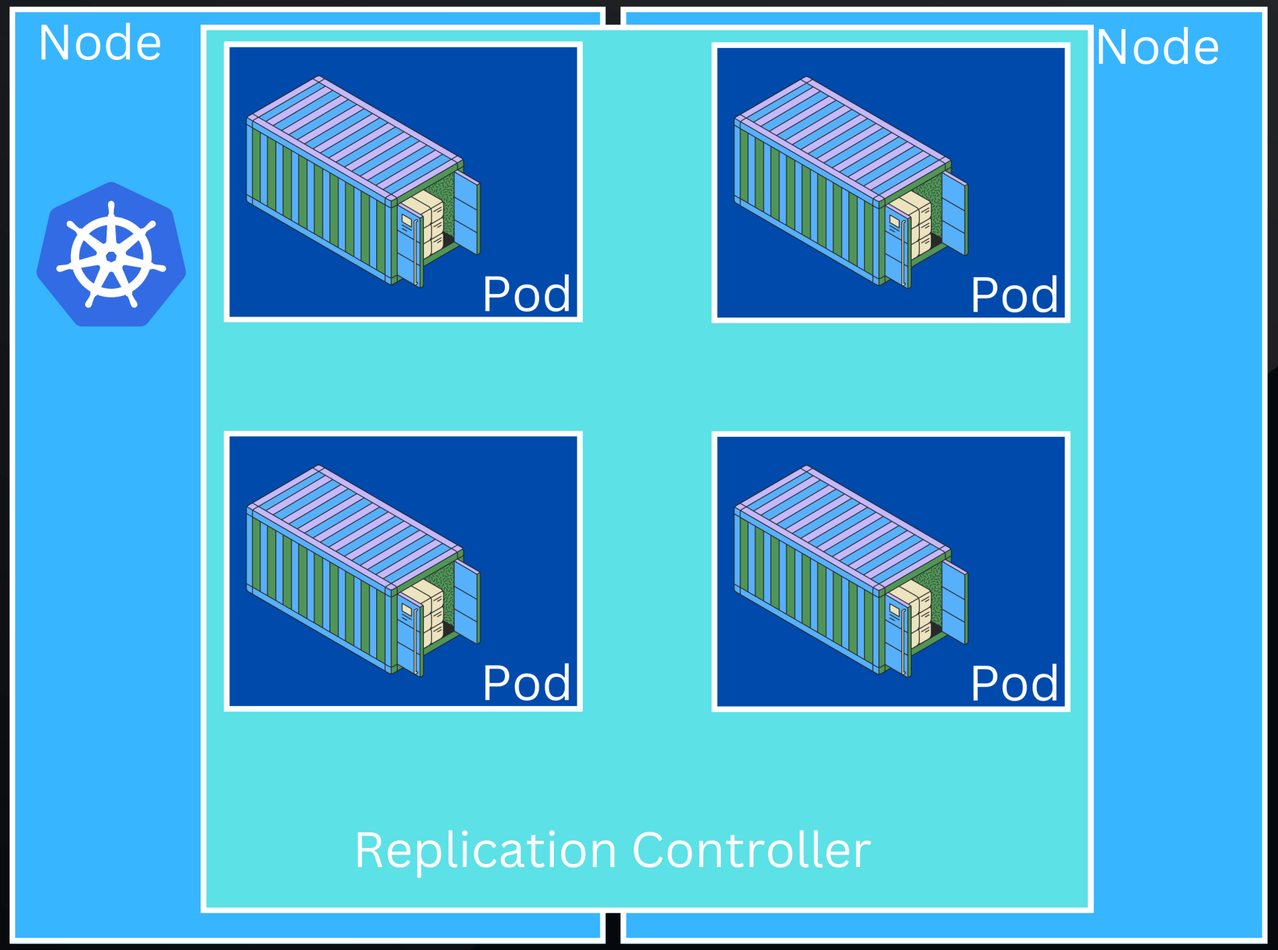
CONTROLLER

- **Replication Controller**: Ensures that a specified **number of replicas of a pod** are running at all times. If a pod fails, the replication controller creates a new one to replace it.
- **Deployment Controller**: A higher-level abstraction that builds on top of replication controllers and provides additional features such as rolling updates and rollbacks.
- **StatefulSet Controller**: Used to manage stateful applications, such as databases, where the pods must be deployed and scaled in a specific order.
- **DaemonSet Controller**: Ensures that a specified pod is running on every node in the cluster.
- **Job Controller**: Used to run batch jobs to completion.

REPLICATION CONTROLLER



REPLICATION CONTROLLER



REPLICATION CONTROLLER VS. REPLICA SET

- **Namespace**: RC - core API vs. RS - part of extension
- **Selector**: both manage pod. but RS support more powerful selector using multiple label selectors and logical operations
- **Updating Strategy**: RC only support simple create new one and delete old one. However, RS support rolling updates and automatic rollbacks
- **Pod template**: RC does not support updating the pod template without changing the selector, but RS supports updating the pod template without changing the selector. This means that **you can make changes to the pod template, such as updating the image version or adding environment variables, without affecting the selector or the replicas that are already running**. The Replica Set automatically updates the existing pods to match the new pod template.
- **Deprecation**: RC is deprecated, and new user should use RS

RC YML

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: nginx
spec:
  replicas: 3
  selector:
    app: nginx
  template:
    metadata:
      name: nginx
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx
        ports:
        - containerPort: 80
```

```
 ~/git/learn-k8s/ k get rc
NAME    DESIRED   CURRENT   READY   AGE
nginx   3         3         0       5s
```

```
 ~/git/learn-k8s/ k get pod
NAME                                READY   STATUS    RESTARTS   AGE
hello-minikube-77b6f68484-76hjs     1/1     Running   0          85s
nginx-g8r7l                         1/1     Running   0          114s
nginx-k777t                         1/1     Running   0          114s
nginx-vpwp7                         1/1     Running   0          114s
```

RS YML

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: php-redis
        image: gcr.io/google_samples/gb-frontend:v3

```

```
 ~/git/learn-k8s/ k get rs
NAME                          DESIRED   CURRENT   READY   AGE
frontend                      3         3         0       4s
hello-minikube-77b6f68484     1         1         1       2d14h
```

```
 ~/git/learn-k8s/ k get pod
NAME                                READY   STATUS    RESTARTS   AGE
frontend-8pbkn                      1/1     Running   0          20s
frontend-snp8p                      1/1     Running   0          20s
frontend-zncpw                      1/1     Running   0          20s
hello-minikube-77b6f68484-76hjs     1/1     Running   0          10m
nginx-g8r7l                         1/1     Running   0          10m
nginx-k777t                         1/1     Running   0          10m
nginx-vpwp7                         1/1     Running   0          10m
```

RS SCALE

- update the `spec.replicas` then apply

```
1   kubectl replace -f replicaset.yml
2
3   kubectl scale --replicas=3 -f replicaset.yml
4
5   kubectl scale --replicas=3 rs/frontend
```