CONTROLLERS

kubernetes

What is controller

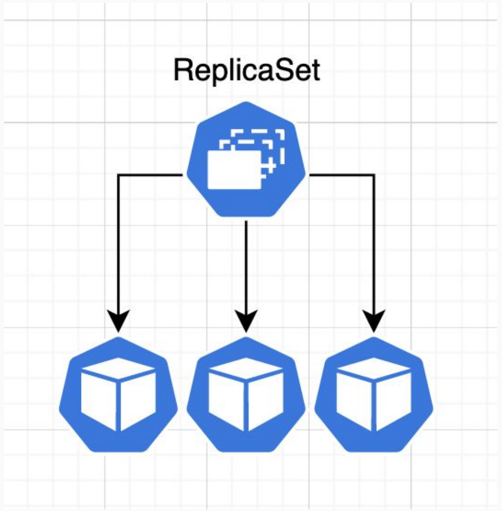
- Componenta that manage the pods behavior

Controller available

- **ReplicasSets**
- **Deployment controller**
- **StatefulSet**
- **Daemonset**
- **Jobs**
- **CronJob**

REPLICASET

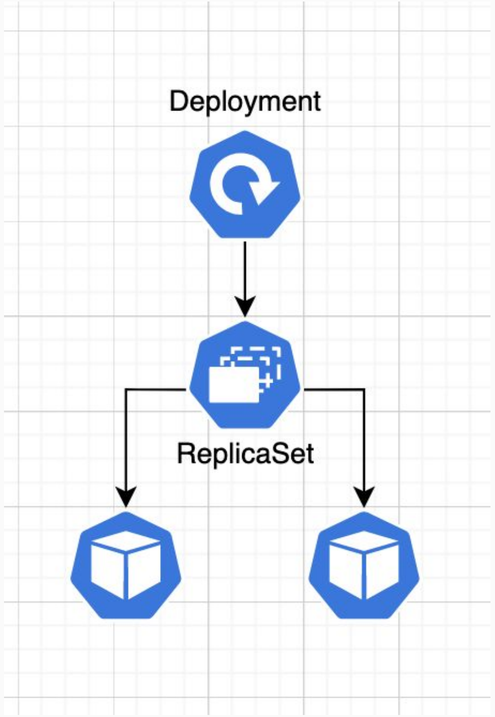
- The replica number indicate how many Pods it should be maintaining by the resource
- Increase the availability



```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
```

DEPLOYMENT CONTROLLER

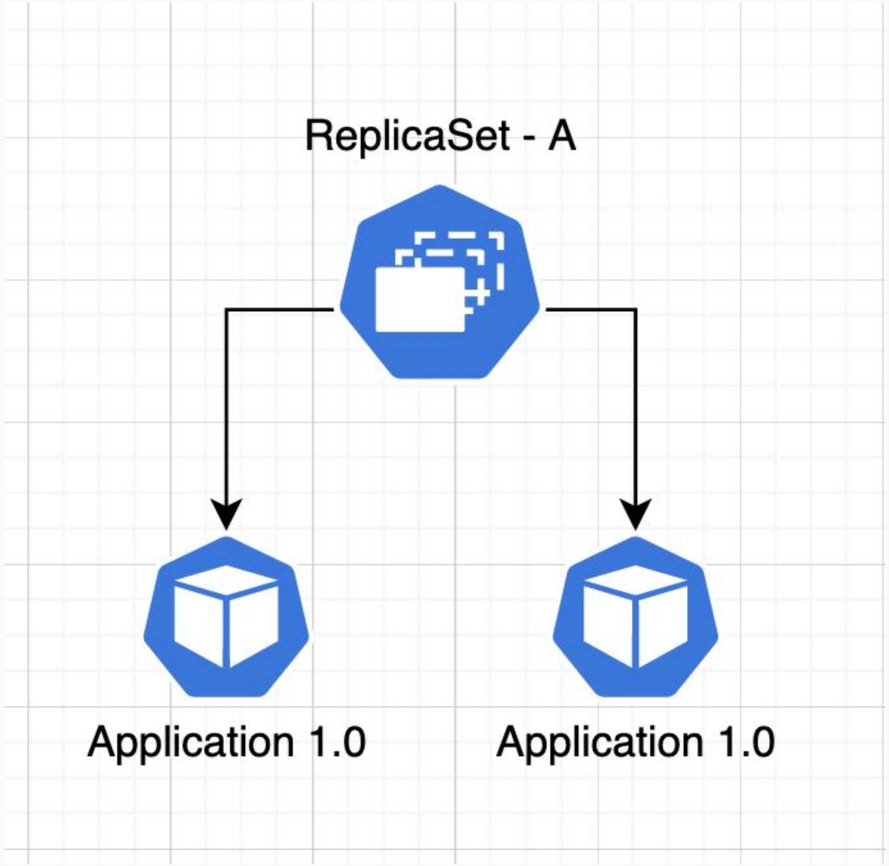
- Deployment controller, control replicaSets



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

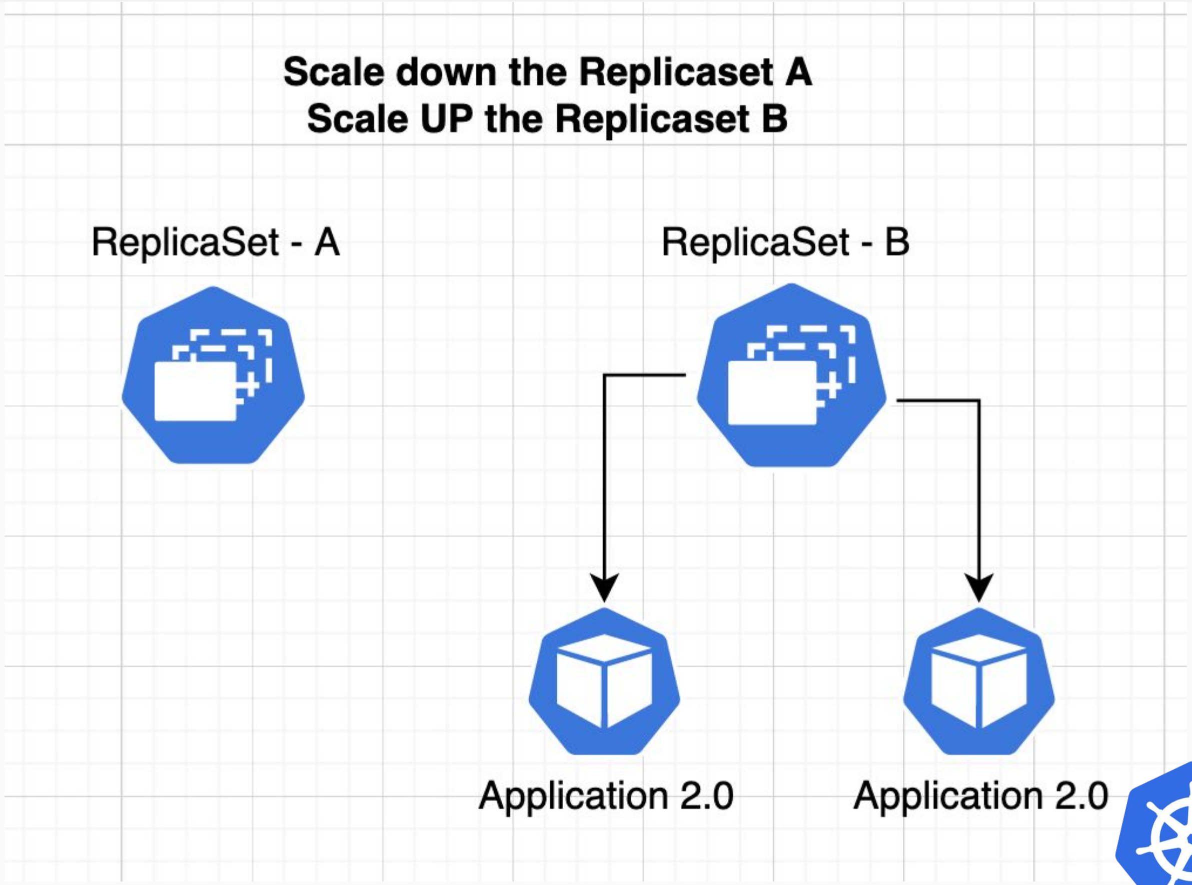
Updating application using ReplicaSet

- Application 1.0 is running
- I need to update the application to 2.0



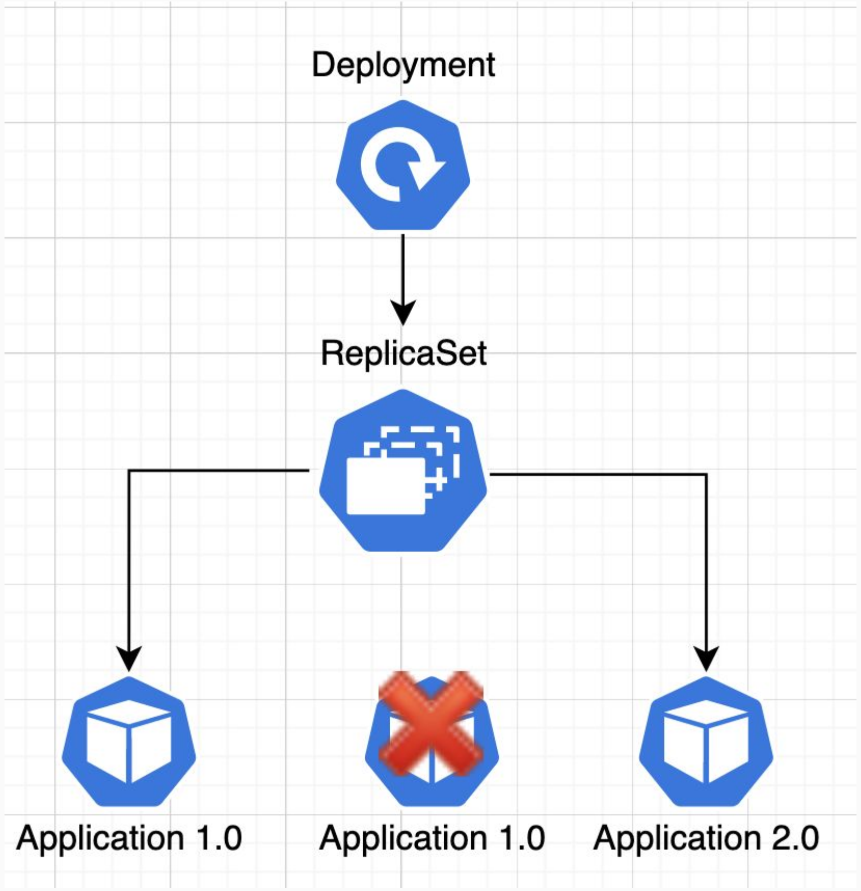
Updating application using ReplicaSet

- Create another replicaset



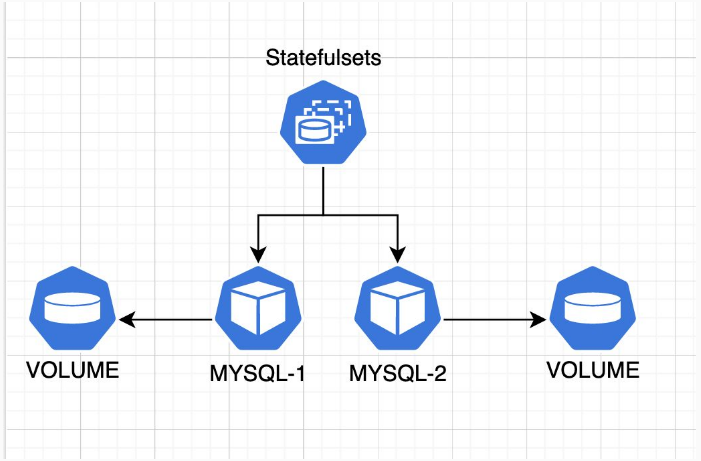
Updating application using Deployment controller

- The rolling update process is automatic
- Default behavior is schedule a new pod version, and kill the old one



Statefulsets controller

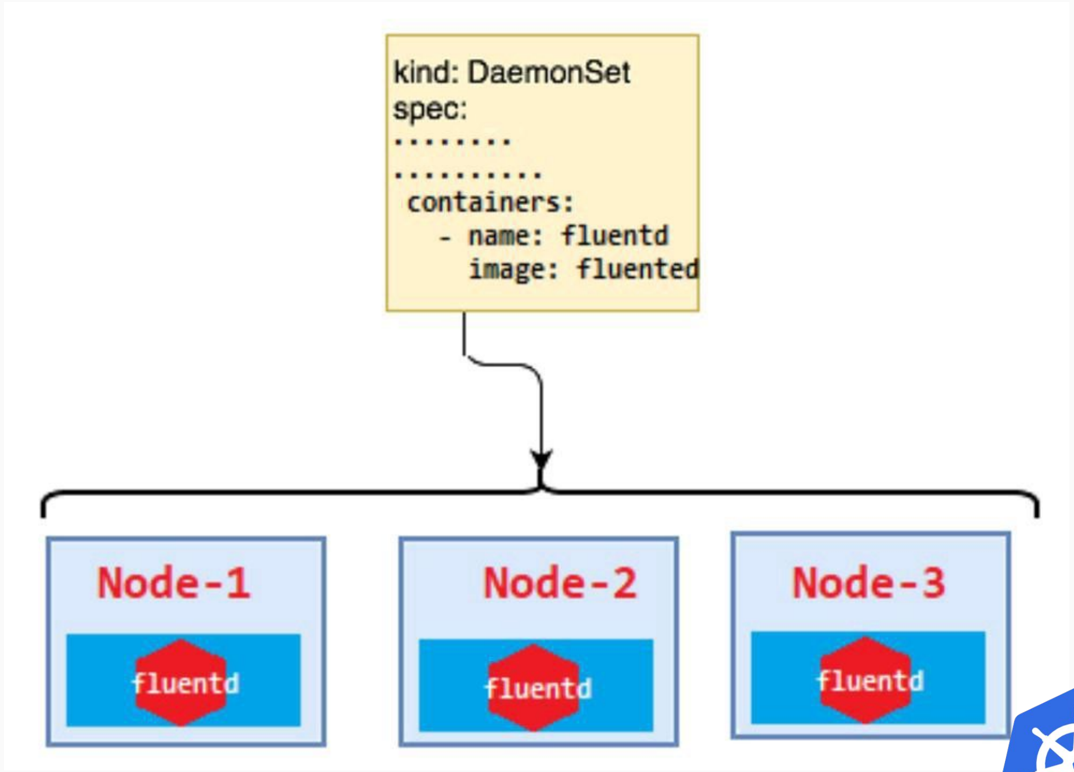
- Used for stateful application
- Application that keeps track its state
- Store the interactions
- Used on application needs to persist data, like MySQL
- Each volume has the same data
- You need to configure the sync between volumes
- Complex to manage



```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  minReadySeconds: 10 # by default is 0
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
```

Daemonset controller

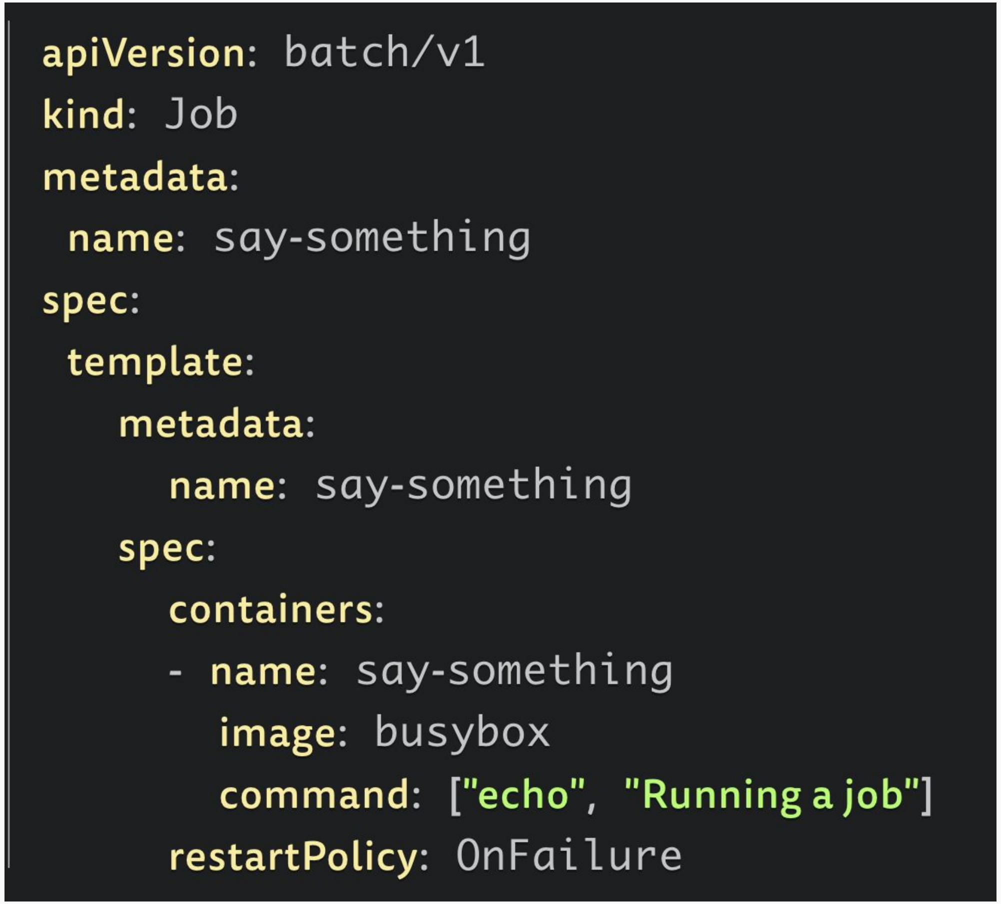
- Insure that each Node has one POD instance
- **Example**: fluentd that collects logs from "/var/logs/containers" folder in each node



```
kind: DaemonSet
spec:
........
..........
 containers:
   - name: fluentd
     image: fluented
```

Jobs

- It runs a job and then it dies



```yaml
apiVersion: batch/v1
kind: Job
metadata:
 name: say-something
spec:
 template:
   metadata:
     name: say-something
   spec:
     containers:
     - name: say-something
       image: busybox
       command: ["echo", "Running a job"]
     restartPolicy: OnFailure
```

CronJobs

- It schedules jobs to run every X time
- Use https://crontab.guru to help with the schedule time

```yaml
apiVersion: batch/v1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "* * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello
            image: busybox:1.28
            imagePullPolicy: IfNotPresent
            command:
            - /bin/sh
            - -c
            - date; echo Hello from the Kubernetes cluster
          restartPolicy: OnFailure
```

LET’S DO THE TASK