---

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

---

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: nginx-rs

spec:

replicas: 3

selector:

matchLabels:

env: prod

matchExpressions:

- { key: tier, operator: In, values: [frontend] }

template:

metadata:

name: nginx

labels:

env: prod

tier: frontend

spec:

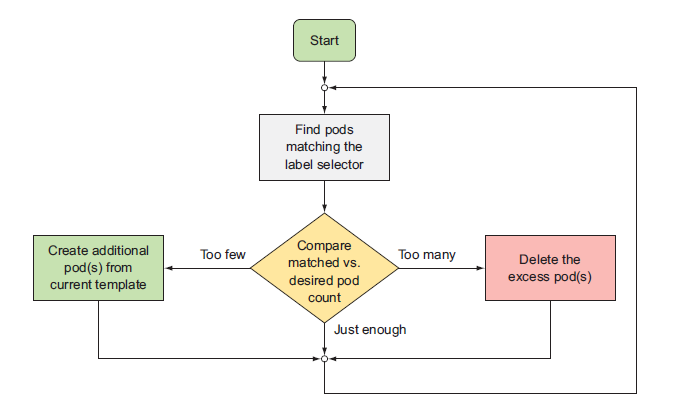
containers:

- name: nginx-container

image: nginx

ports:

- containerPort: 80



Creating a POD from KUBECTL

ku run --generator=run-pod/v1 testpod --image=index.docker.io/sreeharshav/rollingupdate:v3

1. If we change the label of the existing standalone pod similar to label of a RC, then the POD will fall under the RC scope.

2. The extra pods will be removed based on the timestamp and new PODs will be evicted/deleted.

https://medium.com/@zwhitchcox/matchlabels-labels-and-selectors-explained-in-detail-for-beginners-d421bdd05362

Only Job, Deployment, Replica Set, and Daemon Set support matchLabels.

kubectl rolling-update superstar superstar2 --image=index.docker.io/sreeharshav/testcontainer:v1

1. It will be a manul update from one image to other image.

2. New RC will be created and old RC will be deleted.

3. Roll back needs to change again to the old image.

4. Overall manual process and RC rolling-update is deprecated.

kubectl rolling-update superstar2 superstar3 --image=index.docker.io/sreeharshav/rollingupdate:v1

kubectl rolling-update superstar2 superstar3 --rollback