**Replica Sets in Action**

If you are not running a monitoring screen, start it in a new terminal with the following command.

1. watch -n 1 kubectl get pod,deploy,rs,svc
2. kubectl delete pod vote
3. kubectl apply -f vote-rs.yaml --dry-run
5. kubectl apply -f vote-rs.yaml
7. kubectl get rs
9. kubectl describe rs vote
11. kubectl get pods

**Exercise** :

* Switch to monitoring screen, observe how many replicas were created and why
* Compare selectors and labels of the pods created with and without replica sets

1. kubectl get pods
3. kubectl get pods --show-labels

Exercise: Deploying new version of the application

1. kubectl edit rs/vote

Update the version of the image from **schoolofdevops/vote:v1** to **schoolofdevops/vote:v2**

Save the file. Observe if application got updated. Note what do you observe. Do you see the new version deployed ??

**Exercise: Self Healing Replica Sets**

List the pods and kill some of those, see what replica set does.

1. kubectl get pods
2. kubectl delete pods vote-xxxx vote-yyyy

where replace xxxx and yyyy with actual pod ids.

Questions:

* Did replica set replaced the pods ?
* Which version of the application is running now ?

Lets now delete the pod created independent of replica set.

1. kubectl get pods
2. kubectl delete pods vote

**Observe what happens.**

**\* Does replica set take any action after deleting the pod created outside of its spec ? Why?**