***Replicaset***

A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time. As such, it is often used to guarantee the availability of a specified number of identical Pods.

**How ReplicaSet works**

A ReplicaSet is defined with fields, including a selector that specifies how to identify Pods it can acquire, a number of replicas indicating how many Pods it should be maintaining, and a pod template specifying the data of new Pods it should create to meet the number of replicas criteria. A ReplicaSet then fulfills its purpose by creating and deleting Pods as needed to reach the desired number. When a ReplicaSet needs to create new Pods, it uses its Pod template

**Writing ReplicaSet Specs**

To understand how ReplicaSets works with the selectors lets launch a pod in the new namespace with existing specs.

1. cd k8s-code/pods
2. kubectl apply -f vote-pod.yaml
4. kubectl get pods
5. cd ../projects/instavote/dev/

Lets now write the spec for the Rplica Set. This is going to mainly contain,

* replicas
* selector
* template (pod spec )
* minReadySeconds

*file: vote-rs.yaml // After creation run this one*

1. apiVersion: apps/v1
2. kind: ReplicaSet
3. metadata:
4. name: vote
5. spec:
6. replicas: 5
7. minReadySeconds: 20
8. selector:
9. matchLabels:
10. role: vote
11. matchExpressions:
12. - {key: version, operator: In, values: [v1, v2, v3]}
13. template:

**Lets now add the metadata and spec from pod spec defined in vote-pod.yaml. And with that, the Replica Set Spec changes to**

*file: vote-rs.yaml*

1. apiVersion: apps/v1
2. kind: ReplicaSet
3. metadata:
4. name: vote
5. spec:
6. replicas: 5
7. minReadySeconds: 20
8. selector:
9. matchLabels:
10. role: vote
11. matchExpressions:
12. - {key: version, operator: In, values: [v1, v2, v3]}
13. template:
14. metadata:
15. name: vote
16. labels:
17. app: python
18. role: vote
19. version: v1
20. spec:
21. containers:
22. - name: app
23. image: schoolofdevops/vote:v1
24. ports:
25. - containerPort: 80
26. protocol: TCP