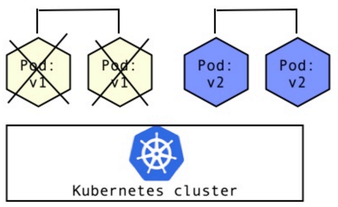
Deployment Strategies: Kubernetes

#1. recreate



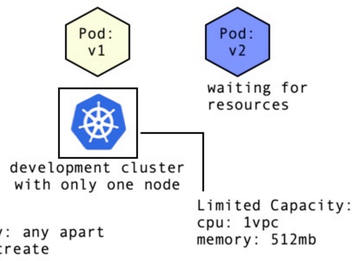
recreate strategy:
1. terminate: first
2. roll-out new version of
the pod/v2 on the cluster

strategy:
The recreate strategy deletes all the existing pods of the version: v1 first. Then it creates the new version: v2 pods on the Cluster.

Note: This strategy incurs down-time of the application and is not recommended for real production rollouts and is usually used in development environments only where downtime doesnt have any significance.

There are few circumstances under which we may have to only use recreate for rolling out the newer version of the pod

1.1

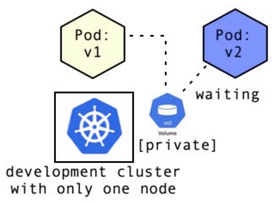


strategy: any apart from recreate

In our development environment we have one node cluster with limited computing capacity using which we can only run one pod on it. In such case if we are rolling out the newer version of the pod v2 with any other deployment strategy other than Recreate will end up in keeping the pod waiting for resources.

The only way in this scenario is to use Recreate strategy so that the existing pod will be terminated first, as the computing resources are freed-up those can be allocated to newer pod v2 and can be brough up.

1.2



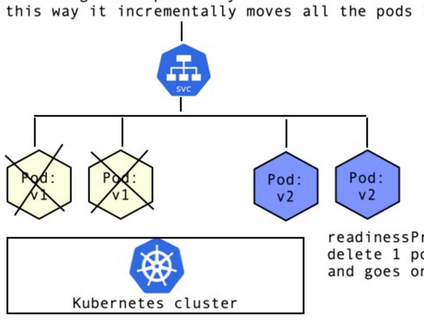
We have an private volume created and mounted on the Pod version: V1. When we are rolling out a newer version of the Pod application V2, The Pod V2 will tries to mount the private volume that is already mounted on V1 pod. So the V2 indefinitely waits for the volume to be available to start which creates an deadlock

if we use deployment strategy as: Recreate, as the Pod V1 gets terminated first, the volume will be detached and available for Pod V2 to mount, so that the new version of the application can be rolled out.

The Recreate strategy is better suitable for rolling out stateful applications.

---

2. Rolling Update (Ramped)
A Rolling update or Ramped deployment updates the pods in a rolling update fashion. It creates an second replicaset with newer version of the pod spec, then it increments the replicas of the new replicaset to 1, once the new pod readinessProbe has been passed, then it deletes the older version of the pod by decrementing the replicas by 1.
In this way it incrementally moves all the pods 1 by 1 from old to newer version of the application.



readinessProbe: Pass
delete 1 pod of older version
and goes on 1 by 1

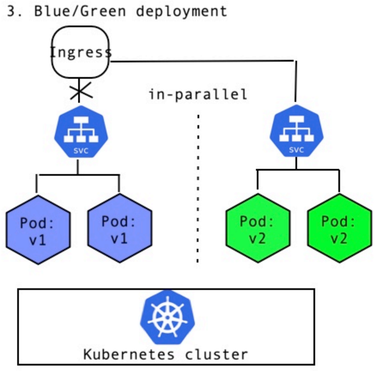
advantages:-
1. no additional infrastructure is required for planning a release
2. no downtime, as we eventually migrate the pods from old version to newer

dis-advantages:
1. rollout and rollback takes huge amount of time
2. as both old and new pods exists in parallel, there is an in-consistency in the behaviour of the application will be observed by the clients while accessing the application
3. there is no control over routing the traffic

Not suitable for stateful applications

---

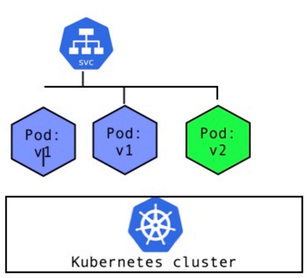
3. Blue/Green deployment



In Blue/Green deployment along with blue version of the pods (older version), the green version of the pods (newer version) also been made available.
The Green version will not be exposed to the enduser and still the user traffic will be routed to the blue version only. The testing team test the new version of the pods in the production environment for an hour or a day or week it takes and once the final tests has been finished, then the traffic will be updated to the newer version of the Service object.

---

4. Canary Deployment



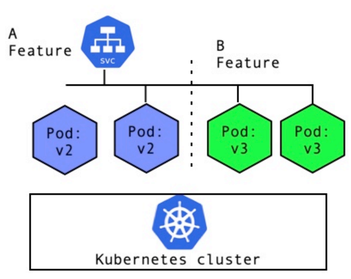
Canary deployment is a technique for rolling out new features or changes to an small subset of users before releasing the update to the entire system.
We create a new ReplicaSet with updated version of the software application while keeping the old ReplicaSet running. A small percentage of the traffic would be routed to the new ReplicaSet, while majority of the traffic continues to be served from the older ReplicaSet itself.

This allows the new version to be tested in the live environment minimizing the risk of issues affecting the entire system. If new system has detected any issues we can quickly rollback to the original ReplicaSet.

Goal: The Goal of canary deployment is to minimize the risk, ensuring high availability and allow controlled testing of the changes before it is been released to large group

---

5. a/b testing



A/B testing deployment strategy we compare 2 different versions (features) A and B of an application to determine which one is performing better in the real world.

Here we release the new version or B Feature to the subset of users in the controlled way and collect the user feedback based on which we decide to rollout the newer version or retain the A version of it.