Hi again. The objective of this video is to learn about Reclaim policies with respect to Kubernetes storage.

In the earlier videos, we have learned about creating Persistent Volumes in Static as well as Dynamic provisioning methods. Persistent volumes provide an abstraction to the actual storage object. These Persistent Volumes are then claimed by Persistent Volume Claims, which are eventually mounted inside the containers.

Status of a Persistent Volume, with respect to Claims, is referred to as its phase. A volume will be in one of the following phases -

Available – a free resource that is not yet bound to a claim

Bound – the volume is bound to a claim

Released – the claim has been deleted, but the resource is not yet reclaimed by the cluster

Failed – the volume failed to create, which could happen because of incorrect storage details, permission issues, etc. Basically the PV couldn’t be created successfully.

We earlier saw the example of Static Provisioning. Let’s revisit it quickly. I have created the EBS volume. Now I have created the PV resource. If I check its status, we see that it is Available as it is not claimed yet. I now create the PVC, and the status of PV changes to Bound.

Similarly, when we used Dynamic Provisioning, we configured the storage class for Immediate Binding mode. Which means that as soon as PVC request is made, a corresponding PV is immediately created and bound to the PVC. So, if I create the PVC again, we get PV created immediately in Bound state.

Reclaim Policy -

In the output of the previous command, you can notice that both these PVs have a Reclaim Policy defined. The static provisioning pv has reclaim policy as Retain whereas Dynamically created PV has reclaim policy as Delete. Let’s understand what it means.

When a user is done with their volume, they can delete the PVC objects from the API that allows reclamation of the resource. The reclaim policy for a PersistentVolume tells the cluster what to do with that Persistent Volume after it has been released of its claim. Volumes can either be Retained, Recycled, or Deleted.

Retain - When the PersistentVolumeClaim is deleted, the PersistentVolume still exists and the volume is considered “released”. But it is not yet available for another claim because the previous claimant’s data remains on the volume.

We had earlier released the static PV. Let’s try to create the PVC again which bounds to the same PV. And we check its status; we see that it is in Pending state. We can describe it to get more details and we see here that it cannot bind to this PV which is in released state but is still not available.

An administrator can manually reclaim the volume by either deleting the volume itself, or by cleaning up the data inside the volume or by cleaning the source itself. What we will do here is that we will clear up the reference of the earlier claim. I can edit the PV and remove the reference of the earlier claim. When I save it, it first gets into Available state and then quickly it is claimed by the pending PVC. We can check the status of the PVC. It moves from Pending to Bound.

Let’s delete the pvc, PV and the EBS volume before going forward. I will just confirm the volume name before deleting.

Recycle - When the PersistentVolumeClaim is deleted, the PersistentVolume gets recycled back into the pool of unbound persistent volumes.

the Recycle reclaim policy performs a basic scrub (rm -rf /thevolume/\*) on the volume and makes it available again for a new claim.

The Recycle reclaim policy is deprecated. Instead, the recommended approach is to use dynamic provisioning.

Delete - When the PersistentVolumeClaim is deleted, the persistent volume and the associated storage in external infrastructure, is also attempted to be deleted.

For dynamically provisioned PersistentVolumes, the default reclaim policy is "Delete". This means that a dynamically provisioned volume is automatically deleted when a user deletes the corresponding PersistentVolumeClaim.

If we check the earlier created Storage Class, we can see that the Reclaim Policy is set as Delete. And thus the Persistent Volume created as part of dynamic provisioning also has reclaim policy set as delete. Right now in the EBS volumes we have 1 volume used by this PV. If i delete the PVC now, it should delete PV as well as the EBS volume as per definition. Let’s check. I have just deleted the PVC. If I check the PV, I see that it is gone. If I refresh the EBS volumes page, I see that the volume is also gone.

This automatic behavior might be inappropriate if the volume contains precious data. In that case, it is more appropriate to use the "Retain" policy. With the "Retain" policy, if a user deletes a PersistentVolumeClaim, the corresponding PersistentVolume will not be deleted. Instead, it is moved to the Released phase, where all of its data can be manually recovered.

I will now delete the existing Storage Class. Just to check and it is gone. In this yaml file, I have made the addition about explicitly specifying the reclaim policy as Retain. Let’s create the storage class; it is created successfully. We can see that the policy is configured as Retain. Let’s create the PVC.A corresponding PV is created with Reclaim policy as Retain. And also the EBS volume is created. Now, I will delete the PVC again. The PVC is deleted, and the PV goes in the Released state instead of getting deleted. The EBS volume also remains. In this case, all the data will be preserved from automatic deletion.

So, that’s all about this video. We learnt about the phases a Persistent Volume can be in. We also learnt with examples that in the newer versions of Kubernetes, PersistentVolumes can have reclaim policies as "Retain" and "Delete".

Thank You!!