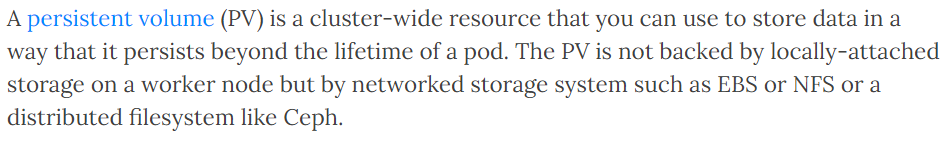
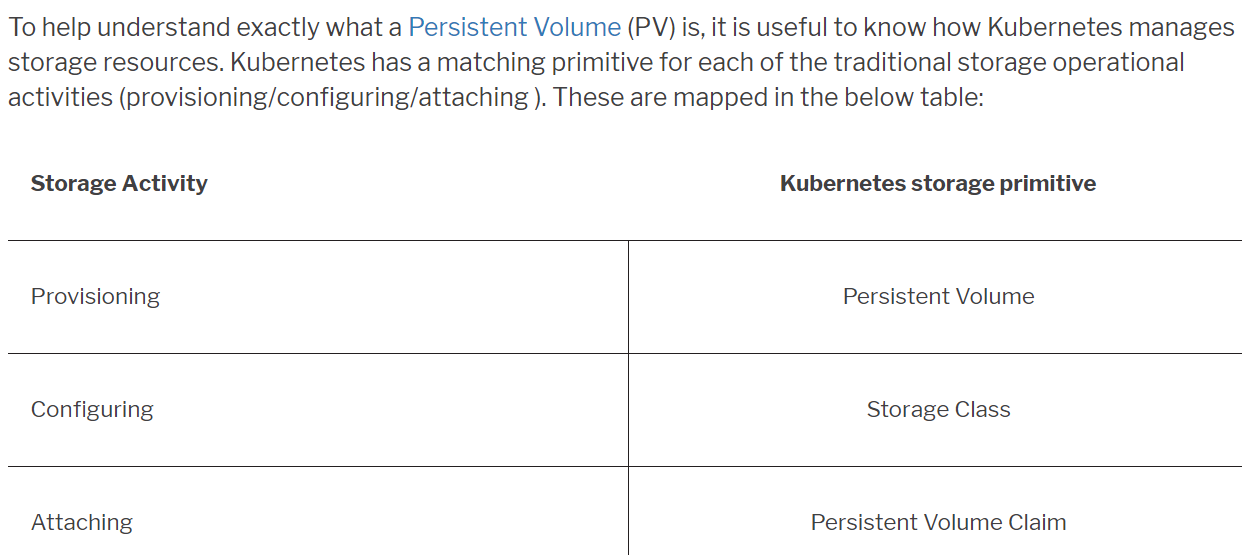
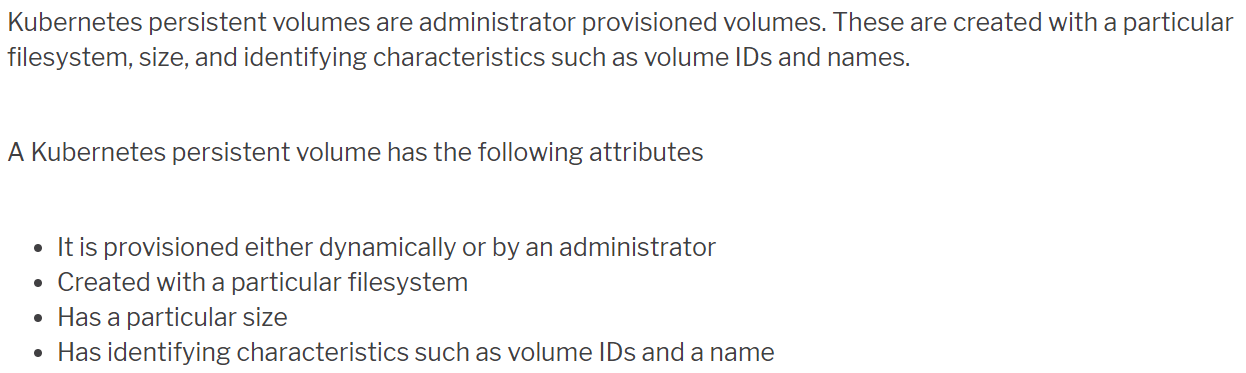
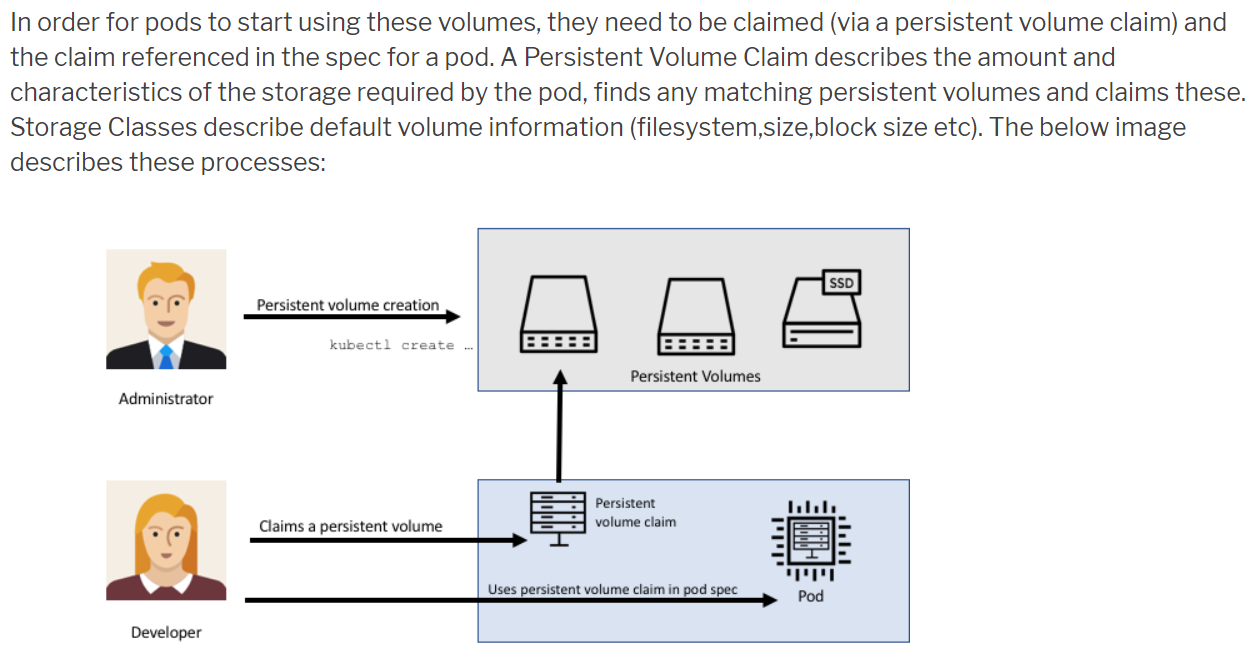
**Persistent Volumes:**

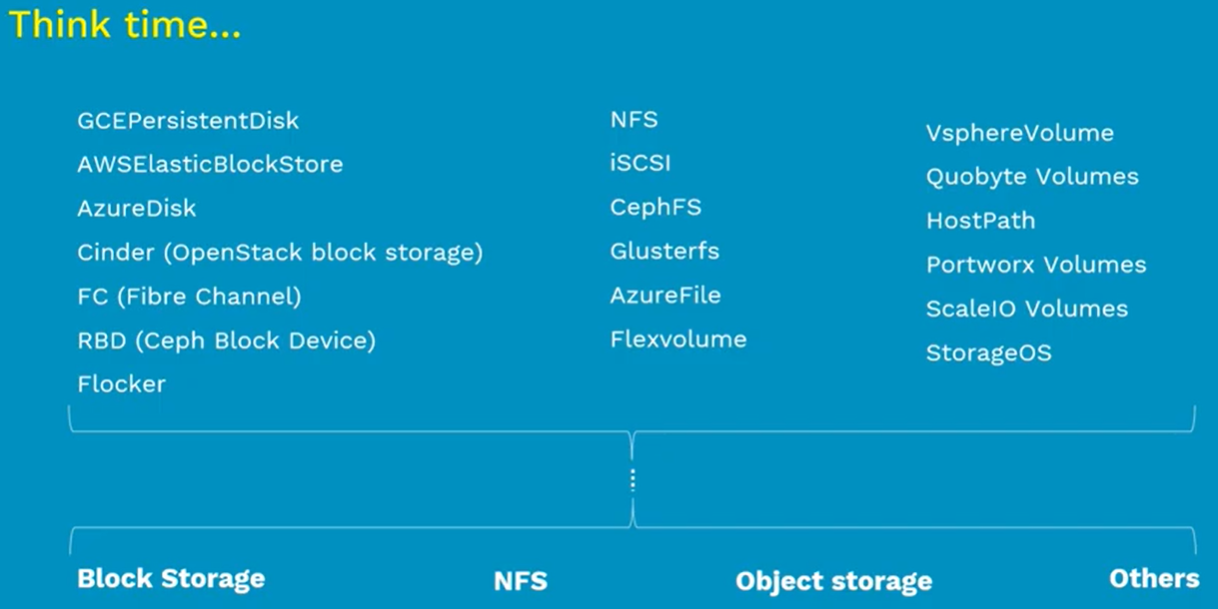


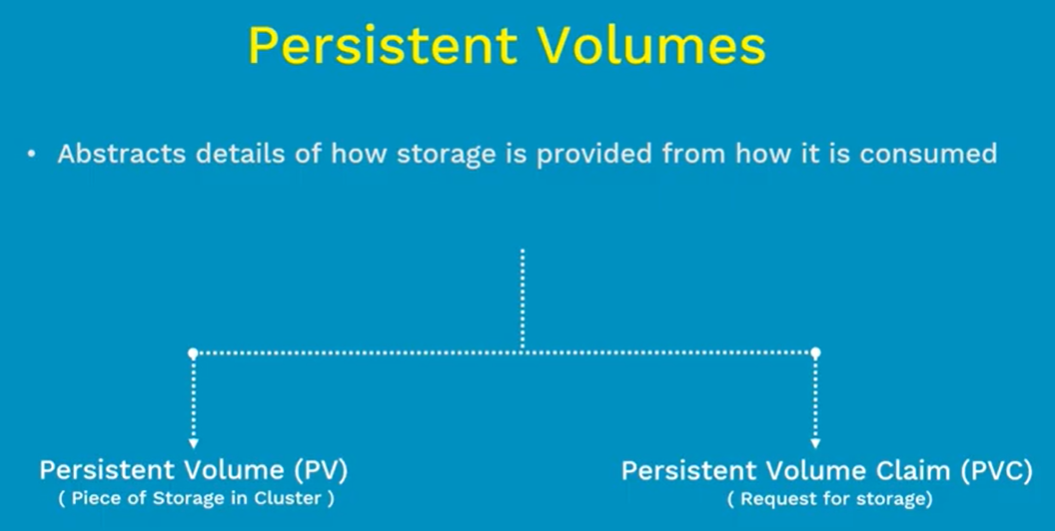






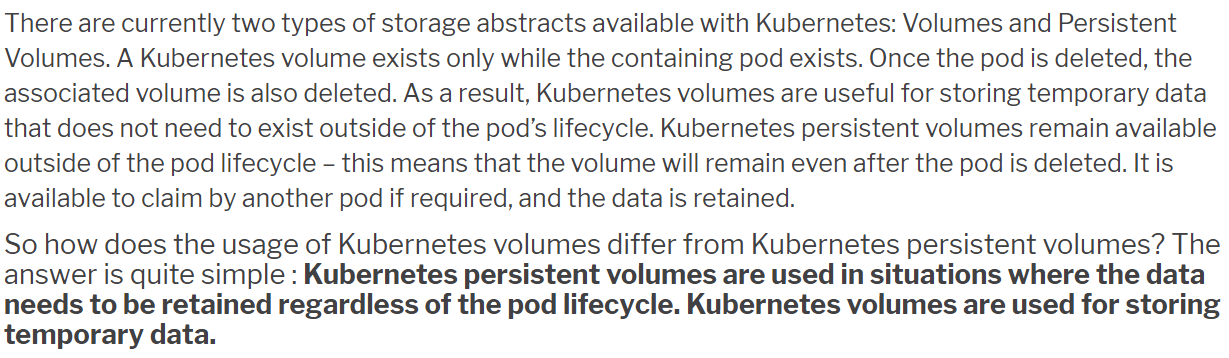




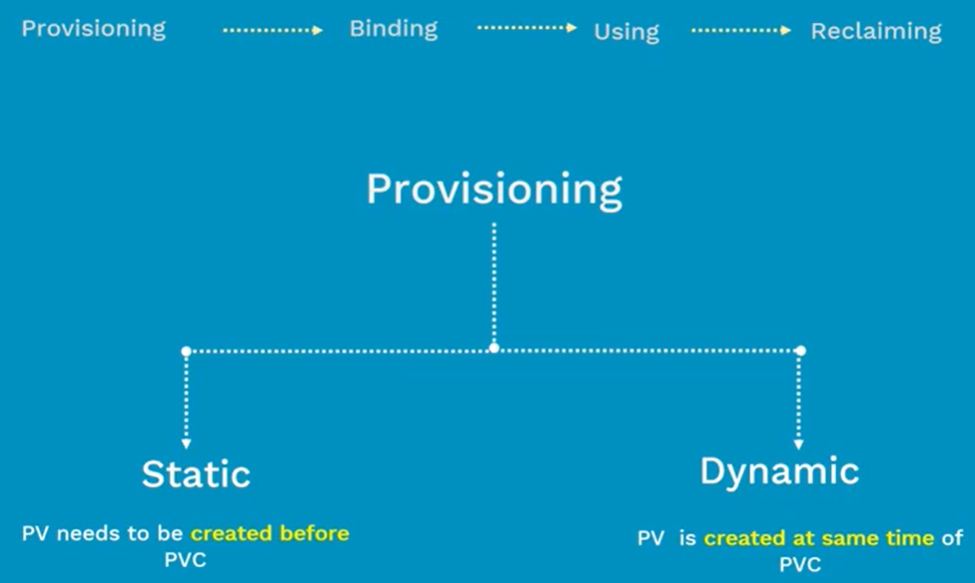


* Persistent volume is a piece of provisioned storage inside a cluster.
* PVC is a request of storage by user with the permissions like read or write.

**Kubernetes volumes vs persistent volumes:**

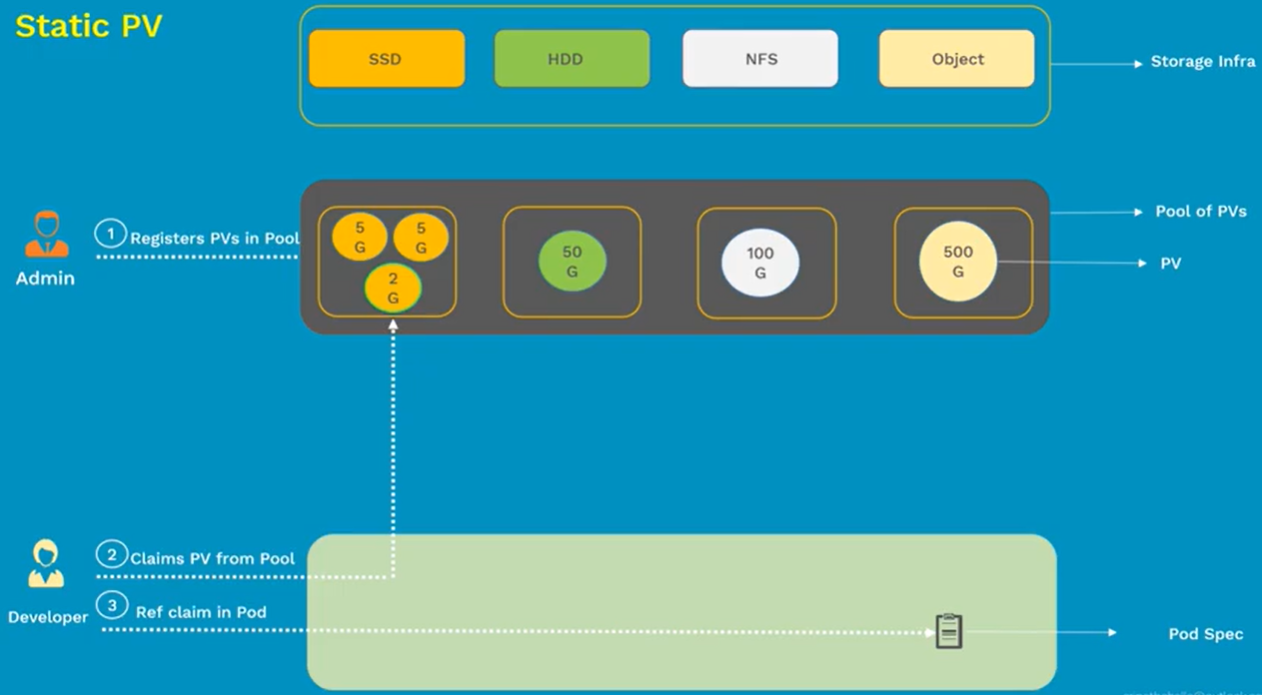


**Lifecycle of PV:**



* These PVs can be provisioned statistically or dynamically.
* In provisioning stage. We need to create storage like block storage or nfs or object storage etc.
* In binding stage, we bind the storage request to the persistent volume which was created earlier. This is called PVC
* If we do persistent volume claim. Kubernetes will watch for the match of the storage to binding.
* If we claim for 100 GB but the we have only 50 GB pool, then Kubernetes can’t bind the claim. It will wait for the 100 GB storage.
* Once this binding is done. We can use this claim as volume inside the POD

**Static PV:**



* Let’s say we have created a PV with SSD and requested for 2 GB PVC. Then this claim gets bounded with PV. And the same can be used in POD.
* If the PVC doesn’t match with one of the chunks in pool. Then we need to wait until that chunk is created in pool.
* To overcome this issue. We have dynamic PV

**Dynamic PV:**

* We don’t create PVs here. We create storage classes. We also have to configure default storage class. This will be helpful when we don’t mention storage class in PV config. It can take the default one.
* So, here we no need to worry about the correct storage chunk is available or not in the storage pool.
* We just need to check if the required storage class is there or not.
* Once that is confirmed. We can create PVC which intern creates storage respective PV and it gets bounded and gets mounted on POD.