

13

PERSISTENT VOLUMES AND CLAIMS

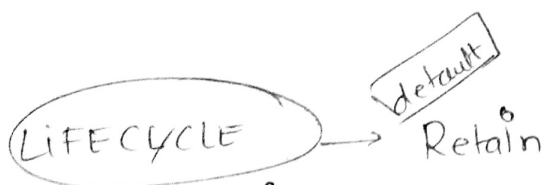
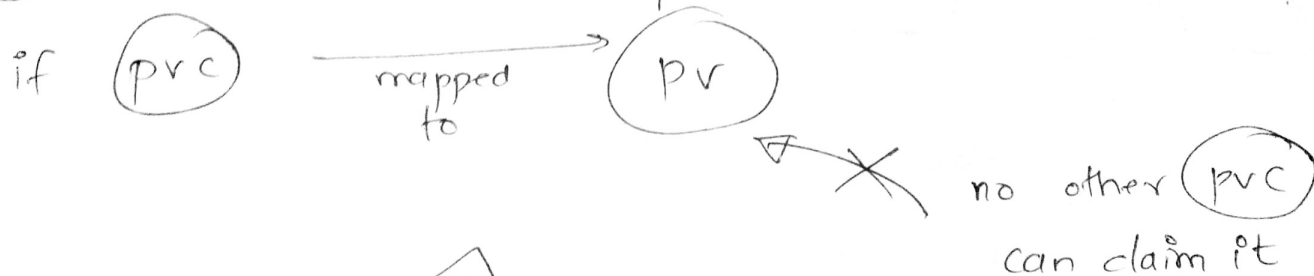
- Once persistent volume is created users have to use it using persistent volume claim

static provisioning

cluster admin has to create persistent volume manually

dynamic provisioning

create storage class o
eBS
Compute Disk
Azure File



- even after pod deleted data will still be there until cluster admin deletes the contents

Recycle

- [Deprecated]

Delete

- Deletes pv and deletes data

Access mode

RWO

Read write once



only one node will
be able to R/W. All
other nodes will
be only able to
read

RWM

Read write many



all nodes will
be able to
write + read

RO

Read only

example configuration

spec:

StorageClassName: manual

capacity

Storage 1Gi

Access Modes

- ReadWriteOnce

hostPath

- path: /kube

Hostpath is just a directory on server.

↳ if multinode

↳ it will work only if pod
gets scheduled on node
with this directory

↳ He showed a demo
using node selector

→ hostpath only works with /tmp

14

SECRETS

Example :- mysql database password

5 ways to use Secret

① as a environment variable
↳ can be used by container

② mount as volume

↳ mounted as individual file
↳ one file for each key value pair
↳ depends on no of files

→ Secrets have to be base-64

eg ~~echo~~ echo -n 'secret-value' | base64

15

Config Maps

→ useful in updating configuration dynamically

— Can create using yaml or command

— in pod yaml

↳ use config map variable

↳ as environment variable

OR

mount as a volume

↳ one file per ~~key~~ key pair value

→ ~~update~~

value is automatically updated in
running pod whenever you update
config map

— You can create configmap from ~~your~~ your
own config file

IMMUTABLE CONFIGMAPS / SECRETS

↳ can only delete → can't edit

just add

↳ `immutable: true` in yaml

Have to ~~can~~ enable `ImmutableEphemeralVolumes` feature gate

↳ enabled by default in latest kubernetes

* Even if you remove the immutable tag later, you won't be able to edit it.

↳ only way to edit would be to delete and recreate it

→ Applies to both configmap & secret.

(18)

RESOURCE QUOTAS & LIMITS

- In production environment, if you have multiple teams running in single cluster, you can have this limits so that no single team can utilize all the resources.

→ Resource quotas

- limit on the resources

eg only this many pods, services etc ~~can~~ can be created

- applied to namespaces, not on entire cluster

Resource limits

- CPU, Memory
- for entire namespace

Limit Range

- applies to all containers

Request limit

- Resource can only request memory within that limit.

17) RENAMING KUBERNETES NODES

- changed hostname of one of nodes in cluster
 - ✓ updated /etc/hosts in host and master.
- Checked kubelet status on worker.

good practice :-

- drain the node.
- stop all pods
- then ~~stop~~ remove the node from cluster & readd

In the error case

- ↳ Had to delete the node
- ↳ join the node again the cluster.

For maintainance

Drain server → kubectl drain ← worker name → - --ignore-daemon

- ↳ it will stop all the running pods
- ↳ NO new pods will be scheduled

18

RANCHER

* cluster management tool

↳ can manage multiple clusters

↳ runs inside docker container
↓
single

18.1

Rancher v2

↳ Nice prometheus + grafana monitoring

18.2

Fluentd + Elastic Stack

18.3

* Slack alerts