(13) PERSISTANT VOLUMES AND CLAIMS

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

→ static provisioning

cluster admin has to create persistent volume manually

dynamic provisioning

Create storage class
EBS
Compute Disk
Azure File

if (pvc) → mapped to (pv) ← ✗ no other (pvc) can claim it

LIFECYCLE → Retain [default] - even after pod deleted data will still be there until cluster admin deletes the contents

→ Recycle - (Deprecated)

→ Delete - Deletes pv and daletes data

ACCESS MODE

- RWO	RWM	RO
Read write once ↓ only one node will dble to R/w. All other nodes will be only ~~op~~ able to read	Read write many ↓ all nodes will be able to write + read	Read only

example configuration

```
spec:
 StoragclassName : manual
 capacity
   Storage   1Gi
 accessModes
 - ReadWrite Once
 hostPath
 - path : /kube
```

Host path is juste 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 -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 enviroment variable

OR

mount as a volume

↳ one file per ~~co~~ key pair value

→ ~~upda~~

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

– You can create configmap from ~~you~~ your ~~fil~~ own config file.

5.1

IMMUTABLE CONFIGMAPS / SECRETS

↳ can only delete → cant edit

just add

↳ immutable : true in yaml

— Have to ~~can~~ enable Immutable Ephemeral Volumes feature gate

↳ enabled by default in latest kubernetes

— * Even if you remove the immutable tag later, you wont 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 enviroment, 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 ~~c~~ can be created

- applied to namespaces. not on entire cluster

Resource limits

- CPU, Memory

- For entire namespace

Limit Range

- applies to all containers

Requesb. 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 ~~all~~ 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

--ignore daem

Drain server → `kubectl drain <worker name>`

↳ it will stop all the running pods

↳ NO new pods will be scheduled

(18) RANCHER

- cluster management tool

↳ can manage multiple cluster's

↳ runs inside ↓ docker container
single

(18.1) Ranchor v2

↳ Nice prometheus + graphana monitoring

(18.2) ↳ Fluent d + Elastic stack

(18.3) Slack alerts