Course: CKA   Date: 20/10/2024
Class: 22
Institute: Linux Pathshala Limited
------Md. Nasir Uddin Pavel-----

Topics:
- ConfigMap
- Secrets
- Storage in Kubernetes
- EmptyDir Volume
- hostPath Volume
- nfs Volume

ConfigMap:
* A ConfigMap is an API object used to store non-confidential data in key-value pairs. Pods can consume ConfigMaps as environment variables, command-line arguments, or as configuration files in a volume.
* ConfigMap is namespace specific
* ConfigMap does not provide secrecy or encryption.
* Not Shareable among namespaces

```
#kubectl get configmaps
#kubectl get cm

#kubectl create configmap <name> --from-literal key1=value1
#kubectl create configmap dbconfig1 --from-literal MYSQL_ROOT_PASS=centos321
#kubectl get configmaps
#kubectl describe configmaps  dbconfig1
```

secret:
* A Secret object stores sensitive data such as credentials used by Pods to access services. For example, you might need a Secret to store the username and password needed to access a database.
* secret is namespace specific
* secret provide secrecy / base64 encoded data
* Not Shareable among namespaces

```
#kubectl get secrets
#kubectl create secrets generic <name> --from-literal key1=value1
#kubectl create secrets generic dbconfig1 --from-literal MYSQL_ROOT_PASS=centos321
#kubectl get secrets
#kubectl describe secrets dbconfig
```

:: Volumes In Kubernetes::

On-disk files in a container are ephemeral, which presents some problems for non-trivial applications when running in containers.

One problem occurs when a container crashes or is stopped.
Container state is not saved so all of the files that were created or modified during the lifetime of the container are lost.
During a crash, kubelet restarts the container with a clean state. Another problem occurs when multiple containers are running in a Pod and need to share files.

It can be challenging to setup and access a shared filesystem across all of the containers.

The Kubernetes volume abstraction solves both of these problems. Familiarity with Pods is suggested.

```
 - Volume Types:

  - Ephemeral Volume
   - emptDir

  - PersistentVolume
    -- PersistentVolume Types
    >> hostPath
    >> nf
    >> Local
    >> Iscsi
    >> FC
    >> ceph
    ....
```

LAB:
- emptyDir Volume
- hostPath Volume
- nfs Volume

##

:::emptyDir Volume:::

Think emptyDir as a scratchpad memory for Pods. Unlike persistent volumes that outlive pod lifecycles, emptyDir exists solely for its pod’s lifespan,
making it perfect for temporary data storage and sharing between containers within the same pod

An emptyDir volume is created when the Pod is assigned to a node. As the name says, the emptyDir volume is initially empty.
When a Pod is removed from a node for any reason, the data in the emptyDir is deleted permanently.

```
##############EmptyDir##########################

 volumeMounts:
    - mountPath: /cache
      name: cache-volume
  volumes:
  - name: cache-volume
    emptyDir:
      sizeLimit: 500Mi

#########################################
```

:::hostPath Volume::::

A hostPath volume mounts a file or directory from the host node’s filesystem into your Pod. This is not something that most Pods will need, but it offers a powerful escape hatch for some applications.

```
#############hostPath#########################


 volumeMounts:
    - mountPath: /foo
      name: example-volume
      readOnly: true
  volumes:
  - name: example-volume
    # mount /data/foo, but only if that directory already exists
    hostPath:
      path: /data/foo # directory location on host
      type: Directory # this field is optional
###################################################
```

:::nfs Volumes:::

An nfs volume allows an existing NFS (Network File System) share to be mounted into a Pod. Unlike emptyDir, which is erased when a Pod is removed, the contents of an nfs volume are preserved and the volume is merely unmounted. This means that an NFS volume can be pre-populated with data, and that data can be shared between pods. NFS can be mounted by multiple writers simultaneously.

```
##############nfs#############################
 volumeMounts:
    - mountPath: /my-nfs-data
      name: test-volume
  volumes:
  - name: test-volume
```

```
    nfs:
      server: my-nfs-server.example.com
      path: /my-nfs-volume
      readOnly: true

##############################################
```