**STORAGE**

* How do you store application state/dana and exchange it between Pods with Kubernetes? **Volumes (**altough other dana storage options exist**)**
* **Volume** can be used to hold dana and state for Pods and containers
* **Pods live and die so their file system is short-lived**
* Volumes can be used to store dana/state and use it in a Pod
* A pod can have multiple Volumes attached to it
* Containers rely on a **mountPath** to access a Volume
* Kubernetes supports:
  + Volumes
  + PersistentVolumes
  + PersistentVolumeClaims
  + StorageClasses
* **VOLUMES**
  + a volume references a storage location
  + must have a unique name
  + attached to a Pod and may or may not be tied to the Pod's lifetime (depending on the Volume type)
  + **A Volume Mount references a Volume by name a defines a mountPath**
  + **PersistentVolume (PV)** is a cluster-wide storage unit provisioned by an administrator with a lifecycle independent from a Pod. **PersisentVolumeClaims (PVC)** is a request for storage unit (PV).
  + **Types:** 
    - **emptyDir –** empty directory for storing „transient“ dana (shares a Pod's lifetime) useful for sharing files between containers running in a Pod
    - **hostPath –** Pod mounts into the node's filesystem
    - **nfs-** an NFS (Network File System) share mounted into the Pod
    - **configMap/secret –** special types of volumes that provide a Pod with access to Kubernetes resources
    - **persisentVolumeClaims –** provides Pods with a more persistent storage option that is abstracted from the details
    - **Cloud –** cluster-wide storage

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

* **PersistentVolume Workflow**
  + Create network storage resources (NFS, cloud, etc.)
  + Define a Persistent Volume (PV) and send to the Kubernetes API
  + Create a PersistentVolumeClaim (PVC)
  + Kubernetes binds the PVC to the PV
  + Pod Volume references the PVC

**A close up of a clock

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

* **StorageClass (SC)** is a type of storage template that can be used to dynamically provision storage.
* **StorageClass Workflow**
  + Create Storage Class
  + Create PersistentVolumeClaim that references StorageClass
  + Kubernetes uses StorageClass provisioner to provision a PersistentVolume
  + Storage provisioned, PersistentVolume created and bound to PersistentVolumeClaim
  + Pod volume references PersistentVolumeClaim

**A screenshot of text

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**