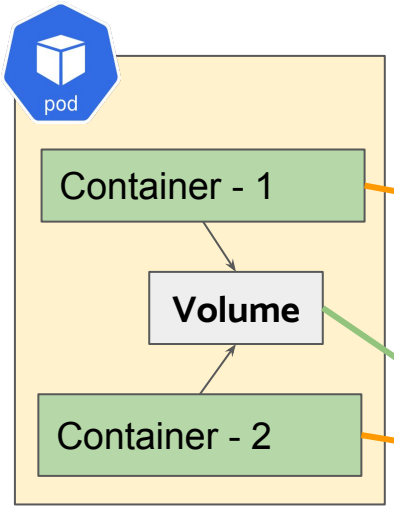
KUBERNETES ACADEMY

Secrets, Configmaps and Env Variable

VOLUMES RECAP

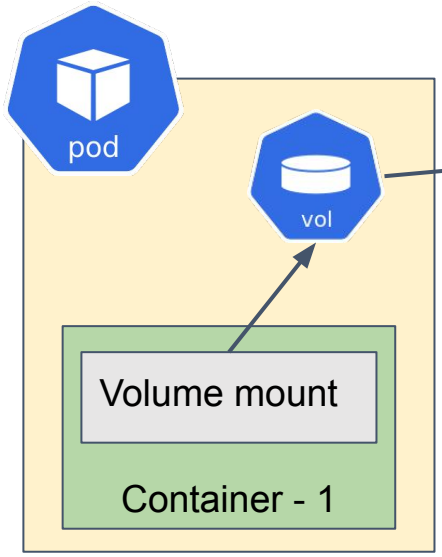
Share files between containers using “emptydir”

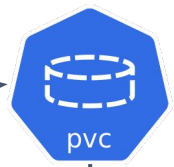



```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-pod
spec:
  containers:
  - image: nginx
    name: container1
    volumeMounts:
    - mountPath: /data/shared-data
      name: shared-volume
  - image: nginx
    name: container2
    volumeMounts:
    - mountPath: /data/shared-data
      name: shared-volume
  volumes:
  - name: shared-volume
    emptyDir: {}
```

- Whatever the “**container1**” saves in the path “**/data/shared-data**”, “**container2**” will be able to see it
- “**MountPath**” doesn’t need to be the same path

Using PV and PVC





BIND


```yaml
volumes:
  - name: task-pv-storage
    persistentVolumeClaim:
      claimName: task-pv-claim
```

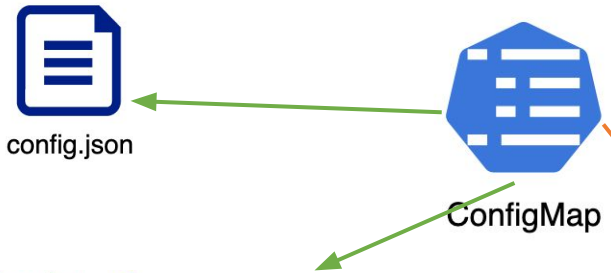
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: task-pv-claim
spec:
  storageClassName: ""
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 3Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"
```

Why Configmap?

- Some applications require files to configure it
- The application was built to use file

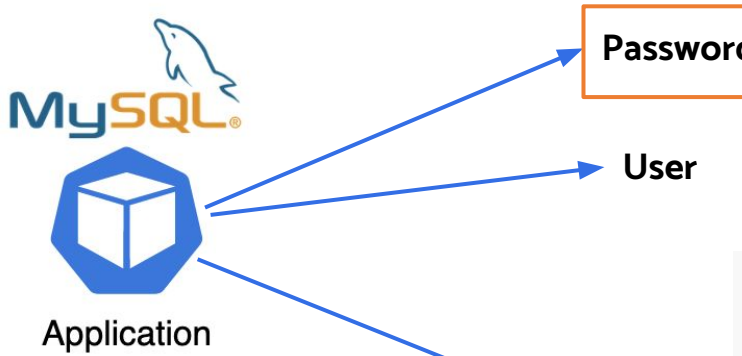
Configmap




```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: myconfigmap
  namespace: default
data:
  config_develop.json: |
    {
        "current_env" : "gcp-development-minikube",
        "Directory_Type" : "folder",
        "Input_Environment_Type" : "gcs",
        "Input_Environment_Location" : "raw-data",
        "Output_Environment_Type" : "bigquery",
        "Output_Environment_Location" : "test_ds",
        "report_location": "reports"
    }
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
      readOnly: true
  volumes:
  - name: foo
    configMap:
      name: myconfigmap
```

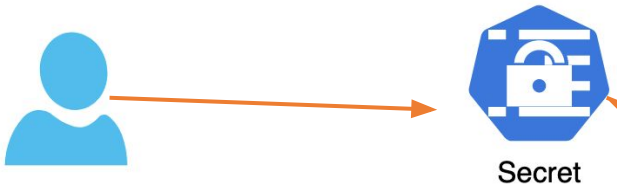
Environment variable




```yaml
spec:
  containers:
  - image: mysql:5.6
    name: mysql
    env:
      # Use secret in real usage
    - name: MYSQL_ROOT_PASSWORD
      value: password
```

- Parameterize the application using env variables

Secrets




```yaml
containers:
- name: mycontainer
  image: redis
  env:
    - name: SECRET_USERNAME
      valueFrom:
        secretKeyRef:
          name: mysecret
          key: username
          optional: false # s
                          # a
    - name: SECRET_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mysecret
          key: password
```

- Passwords should always use secret
- Avoid secret exposing in the deployments

LET’S DO THE TASK