

# Volume, ConfigMap / Secret



# Objectifs

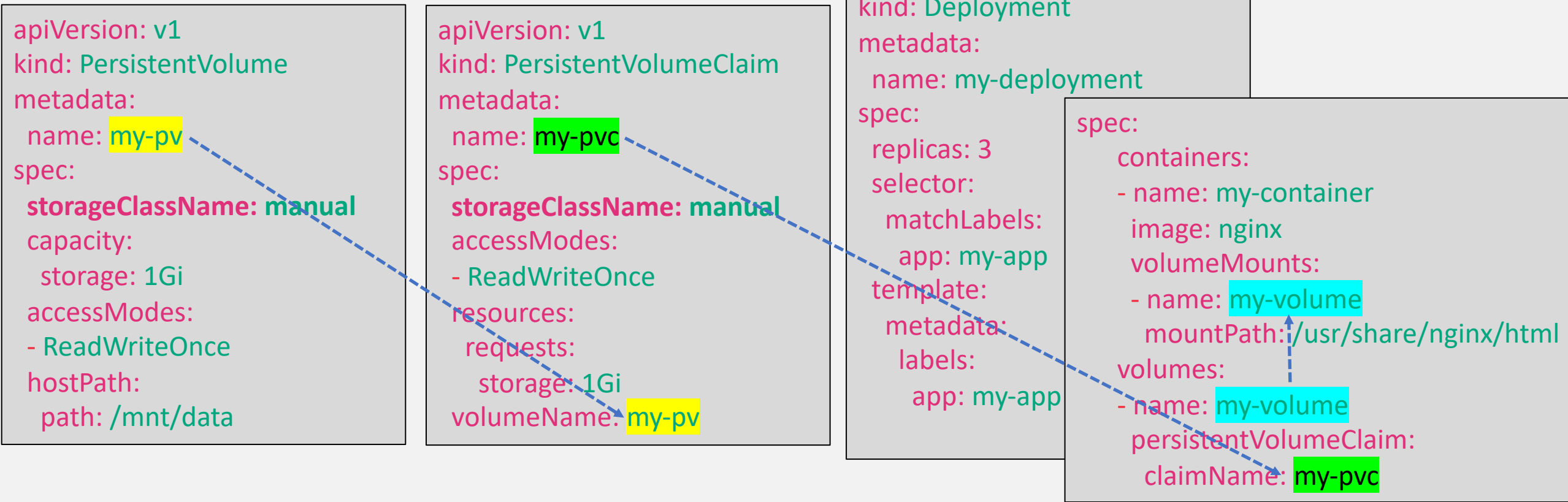
- Volume
- Configmap/secret

# Persistent volume

- Les volumes persistants sont des espaces de stockage utilisés pour persister des données
- Un volume peut être attaché à un Pod
- Ils restent disponibles même si le Pod est redémarré ou mort
- Il existe plusieurs type de volume : locaux, distants (NFS, SMB, iSCSI), réseau (Gluster, Red Hat Ceph) et Cloud
- Pour créer un volume persistant, il faut au préalable créer :
  - Un « persistentVolume » (PV) : espace de stockage dédié et durable. Il est normalement créé par les administrateurs du cluster
  - Un « persistentVolumeClaim » (PVC) : demande d'utilisation d'un PV, donc de l'espace

# Persistent volume - Cas local

- Dans notre cas, nous allons utiliser un stockage sur l'hôte pour nos tests en local (hostPath : ne pas utiliser en production !) et dans le cloud pour les tests dans Azure





# Persistent volume - Cas local

```
~/tmp/k8s/pvdemo/data k get pv,pvc,deployment
```

NAME	REASON	AGE	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS
persistentvolume/my-pv		8m14s	1Gi	RWO	Retain	Bound	default/my-pvc	manual

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
persistentvolumeclaim/my-pvc	Bound	my-pv	1Gi	RWO	manual	8m9s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/my-deployment	3/3	3	3	8m5s

```
~/tmp/k8s/pvdemo/data k describe pv/my-pv
```

Name: my-pv  
Labels: <none>  
Annotations: pv.kubernetes.io/bound-by-controller: yes  
Finalizers: [kubernetes.io/pv-protection]  
StorageClass: manual  
Status: Bound  
Claim: default/my-pvc  
Reclaim Policy: Retain  
Access Modes: RWO  
VolumeMode: Filesystem  
Capacity: 1Gi  
Node Affinity: <none>  
Message:  
Source:  
  Type: HostPath (bare host directory volume)  
  Path: /Users/pfleon/tmp/k8s/pvdemo/data  
  HostPathType:  
Events: <none>

```
~/tmp/k8s/pvdemo/data k describe pvc/my-pvc
```

Name: my-pvc  
Namespace: default  
StorageClass: manual  
Status: Bound  
Volume: my-pv  
Labels: <none>  
Annotations: pv.kubernetes.io/bind-completed: yes  
Finalizers: [kubernetes.io/pvc-protection]  
Capacity: 1Gi  
Access Modes: RWO  
VolumeMode: Filesystem  
Used By: my-deployment-5676d67558-9tz82  
         my-deployment-5676d67558-dw2kt  
         my-deployment-5676d67558-srwfb  
Events: <none>

# ConfigMap et Secret

- Un objet ConfigMap/Secret contient des données sous forme clef/valeur
- Ces objets peuvent être utilisés comme variables d'environnement, arguments de ligne de commande et comme volume
- Les valeurs des secrets doivent être encodées en base64
- Utilisation :
  - Rendre l'environnement configurable au travers de variable d'environnement
  - Créer des fichiers de configuration

# ConfigMap et Secret – Exemple

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: config-map-cours-demo
data:
  ma_variable: "Une valeur"

monfichier.conf: |
  Une ligne
  Une autre...
  C'est un beau fichier de c
```

```
apiVersion: v1
kind: Secret
metadata:
  name: secret-cours-demo
data:
  connectionString_Default:
c2VydmdVpPXNxbDdEYXRhYmFzZT1ub3Rlc0V0dWRpYW50cztQZXJzaXN
0IFNlY3VyaXR5IEluZm89VHJ1ZTtVc2VyIElEPW5vdGVzV2ViVXNlcj
tQYXNzd29yZD1QYXNzd3ByZDtdNdWx0aXBsZUFjdGJlZ2VJlc3VsdFNld
HM9dHJ1ZQ==
  monfichierSecret.conf:
UXVvaSBxdSdlbiBkaXNlIG1lcYBnZXN0ZXMsIGplIG5lIHBlcXgKdG9
sw6lyZXIgc2d1dGlsaXNlciBsZSB0ZXJtZXMgcGFpbiBhdSBjaG9jb2
xhdAplbiBsaWV1IGV0IHBSYWNlIGRlIGNob2NvbGF0aW5lICE=
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo-config-map
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demo-config-map-app
  template:
    metadata:
      labels:
        app: demo-config-map-app
    spec:
      containers:
        - name: my-container
          image: nginx
          volumeMounts:
            - name: fichier-config
              mountPath: /var/demo
            - name: fichier-secret-c
              mountPath: /var/demoSe
```

```
env:
  - name: MA_VAR
    valueFrom:
      configMapKeyRef:
        name: config-map-cours-demo
        key: ma_variable
  - name: ConnectionString_Default
    valueFrom:
      secretKeyRef:
        name: secret-cours-demo
        key: connectionString_Default

volumes:
  - name: fichier-config
    configMap:
      name: config-map-cours-demo
      items:
        - key: monfichier.conf
          path: monfichier.conf
  - name: fichier-secret-config
    secret:
      secretName: secret-cours-demo
      items:
        - key: monfichierSecret.conf
          path: monfichierSecret.conf
```

# ConfigMap et Secret – Exemple

```
~/tmp/k8s/pvdemo k exec pod/demo-config-map-7d4fc87dbb-zs7nq -it -- /bin/bash
root@demo-config-map-7d4fc87dbb-zs7nq:/# cat /var/demo/monfichier.conf
Une ligne
Une autre...
C'est un beau fichier de configuration !
root@demo-config-map-7d4fc87dbb-zs7nq:/# cat /var/demoSecret/monfichierSecret.conf
Quoi qu'en dise mes gestes, je ne peux
tolérer d'utiliser le termes pain au chocolat
en lieu et place de chocolatine !root@demo-config-map-7d4fc87dbb-zs7nq:/# printenv
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_SERVICE_PORT=443
MY_DEPLOYMENT_PORT_80_TCP_PORT=80
MY_DEPLOYMENT_PORT_80_TCP_ADDR=10.102.77.175
HOSTNAME=demo-config-map-7d4fc87dbb-zs7nq
connectionString_Default=server=sql;Database=notesEtudiants;Persist Security Info=True;User ID=notesWe
bUser;Password=Passw0rd;MultipleActiveResultSets=true
PWD=/
MY_DEPLOYMENT_PORT_80_TCP=tcp://10.102.77.175:80
PKG_RELEASE=1~bullseye
HOME=/root
MA_VAR=Une valeur
KUBERNETES_PORT_443_TCP=tcp://10.06.0.1:443
```



# Références

- <https://kubernetes.io/docs/concepts/storage/persistent-volumes/>
- <https://kubernetes.io/docs/concepts/configuration/configmap/>
- <https://kubernetes.io/docs/concepts/configuration/secret/>