TP – Le stockage persistent

Création d'un volume de cache

On ouvre le manifeste redis-emptydir.yml et on personnalise son pod, on constate le volume

```
ludo@kubernetes:$ cd k8s/storage

ludo@kubernetes:$ vim redis-emptydir.yaml
apiVersion: v1
kind: Pod
metadata:
  name: redis
spec:
  containers:
  - name: redis
    image: redis
    volumeMounts:
    - name: redis-storage
      mountPath: /data/redis
  volumes:
  - name: redis-storage
    emptyDir: {}
```

Utiliser des Classe de stockage

```
ludo@kubernetes:$ kubectl get storageclasses.storage.k8s.io
NAME                   PROVISIONER        RECLAIMPOLICY   VOLUMEBINDINGMODE
nfs-storage (default)   ludovic.tech/nfs   Delete          Immediate
```

on ouvre le manifeste storage-pvc.yaml

```
ludo@kubernetes:$ vim storage-pvc.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: web-nfs
  namespace: web
spec:
  storageClassName: nfs-storage
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

Les volumes dans les pods

On ouvre le manifeste web-volumes.yaml , on le personnalise.

```
ludo@kubernetes:$ vim web-volumes.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: web
  name: nginx-deploy
  namespace: web
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      volumes:
      - name: www
        persistentVolumeClaim:
          claimName: web-nfs
      containers:
      - image: nginx:1.19
        name: nginx
      initContainers:
      - name: init-web
        image: busybox:1.28
        command: ['sh', '-c', "echo init par initContainer >
      /usr/share/nginx/html/index.html"]
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
```