TP – Les Pods

1.Manipuler notre premier pod

Création en mode impératif

```
ludo@kubernetes:~$ kubectl run nginx-ludo –image=nginx:latest

ludo@kubernetes:~$ kubectl get all
```

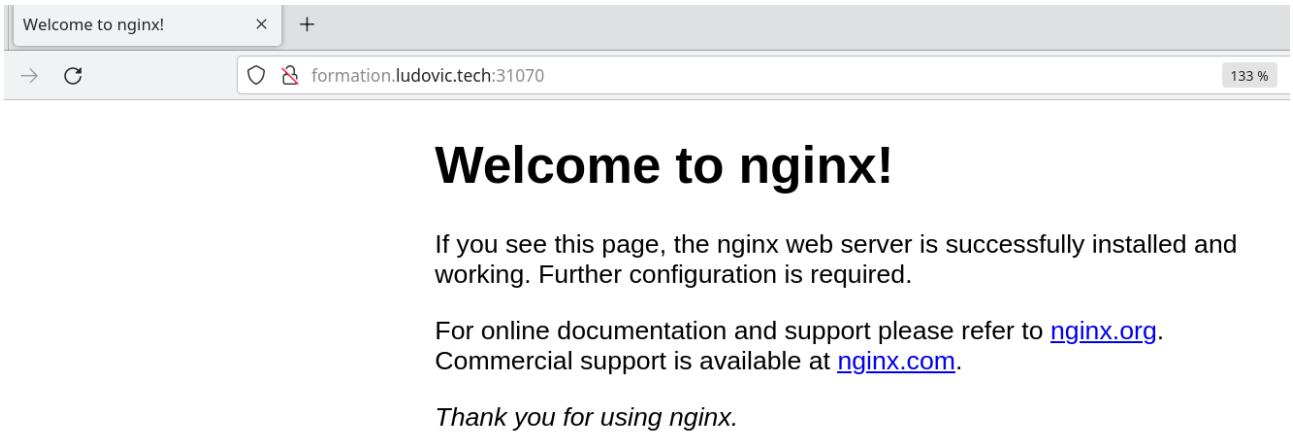
Exposer le pod sur extérieur avec un service NodePort

```
ludo@kubernetes:~$ kubectl expose pod nginx-ludo --type NodePort --port 80

ludo@kubernetes:~$ kubectl get service
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
nginx    NodePort    10.99.252.217   <none>                  80:31070/TCP
```

afficher les endpoints du service

```
ludo@kubernetes:~$ kubectl get pod nginx-ludo -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP               NODE
nginx   1/1     Running   0          68s   192.168.196.92   worker04

ludo@kubernetes:~$ kubectl get endpoints nginx
NAME    ENDPOINTS           AGE
nginx   192.168.196.92:80   5m6s
```

On « requête» le pod



On rentre dans le pod et on modifie le ficher html

```
ludo@kubernetes:~$ kubectl exec -it nginx-ludo -- bash
root@nginx-ludo:/# echo "<h1>Bienvenue sur le serveur de Ludo</h1>"  \
   /usr/share/nginx/html/index.html
```

On supprime le pod et le service

```
ludo@kubernetes:~$ kubectl delete pod nginx
pod "nginx" deleted

ludo@kubernetes:~$ kubectl delete service nginx
service "nginx" deleted
```

2.Gérer les pods avec des manifestes

Se rendre dans le répertoires $HOME/formation/pod , ouvrir le fichier nginx-pod.yaml et on change le nom du pod.

```
ludo@kubernetes:/pod$ vim nginx-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-ludo
  labels:
    app: web
spec:
  containers:
  - image: nginx
    name: nginx
```

On applique le manifeste à l'api server

```
ludo@kubernetes:/pod$ kubectl appply -f nginx-pod.yaml
```

Exposer un pod avec un manifeste

```
ludo@kubernetes:/pod$ vim service-nginx.yaml
apiVersion: v1
kind: Service
metadata:
  name: nodeport
spec:
  type: NodePort
  selector:
    app: web
  ports:
    - port: 80
      targetPort: 80
```

On applique

```
ludo@kubernetes:/pod$ kubectl apply -f service-nginx.yaml
```

TP - La gestion des sondes Kubernetes

La LivenessProbe

Ouvrir le fichier kuard-pod-liveness.yaml

```
ludo@kubernetes:/pod$ vim kuard-pod-liveness.yaml
apiVersion: v1
kind: Pod
metadata:
  name: kuard-ludo
  labels:
    app: liveness-ludo
spec:
  containers:
  - image: gcr.io/kuar-demo/kuard-amd64:1
    name: kuard
    livenessProbe:
      httpGet:
        path: /healthy
        port: 8080
      initialDelaySeconds: 5
      timeoutSeconds: 1
      periodSeconds: 10
      failureThreshold: 2
    ports:
    - containerPort: 8080
      name: http
```

On change le nom du pod, les labels et on applique

```
ludo@kubernetes:/pod$ kubectl apply -f kuard-pod-liveness.yaml
pod/kuard-ludo created
```

On expose l'application

```
ludo@kubernetes:/pod$ vim service-kuard.yaml
apiVersion: v1
kind: Service
metadata:
  name: kuard-ludo
spec:
  type: NodePort
  selector:
    app: demo-ludo
  ports:
    - port: 8080
      targetPort: 8080
```

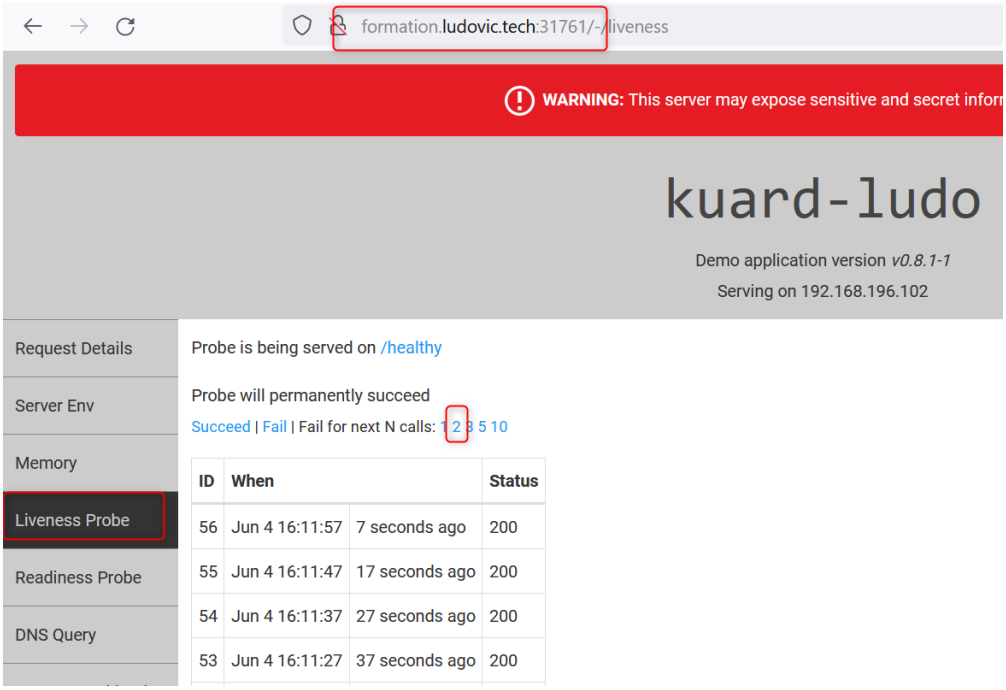
On applique le manifeste

```
ludo@kubernetes:/pod$ kubectl apply -f service-kuard.yaml
service/kuard-ludo created
```

On liste les ressources

```
ludo@kubernetes:/pod$ kubectl get all
NAME             READY   STATUS    RESTARTS   AGE
pod/kuard-ludo   1/1     Running   0          2m10s

NAME                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
service/kuard-ludo   NodePort    10.111.216.187   <none>        8080:30101/TCP
10s
```

On teste la sonde




```
ludo@kubernetes:/pod$ kubectl get pod kuard-ludo

Every 2.0s: kubectl get pod
kubernetes.ludovic.tech: Sun Jun  4 18:14:13 2023

NAME         READY   STATUS    RESTARTS   AGE
kuard-ludo   1/1     Running   0          11m

NAME         READY   STATUS    RESTARTS      AGE
kuard-ludo   1/1     Running   1 (9s ago)    12m
```

On supprime tout

```
ludo@kubernetes:/pod$ kubectl delete all --selector app=demo-ludo
```

La sonde ReadinessProbe

Ouvrir le fichier kuard-pod-readiness.yaml

```
ludo@kubernetes:/pod$ vim kuard-pod- readiness.yaml
apiVersion: v1
kind: Pod
metadata:
  name: kuard
  labels:
    app: demo
spec:
  containers:
  - image: gcr.io/kuar-demo/kuard-amd64:1
    name: kuard
    ports:
    - containerPort: 8080
      name: http
      protocol: TCP
   readinessProbe:
      httpGet:
        path: /ready
        port: 8080
      initialDelaySeconds: 30
      timeoutSeconds: 1
      periodSeconds: 10
      failureThreshold: 1
```

On change le nom du pod, les labels et on applique

```
ludo@kubernetes:/pod$ kubectl apply -f kuard-pod- readiness.yaml
pod/kuard-ludo created

ludo@kubernetes:/pod$  kubectl get all --show-labels
NAME        READY   STATUS    RESTARTS   AGE   LABELS
pod/kuard   0/1     Running   0          7s    app=demo-ludo


ludo@kubernetes:/pod$  kubectl get all
NAME        READY   STATUS    RESTARTS   AGE
pod/kuard   1/1     Running   0          47s
```

On expose l'application

```
ludo@kubernetes:/pod$ vim service-kuard.yaml
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: kuard-ludo
spec:
  type: NodePort
  selector:
    app: demo-ludo
  ports:
    - port: 8080
      targetPort: 8080
```

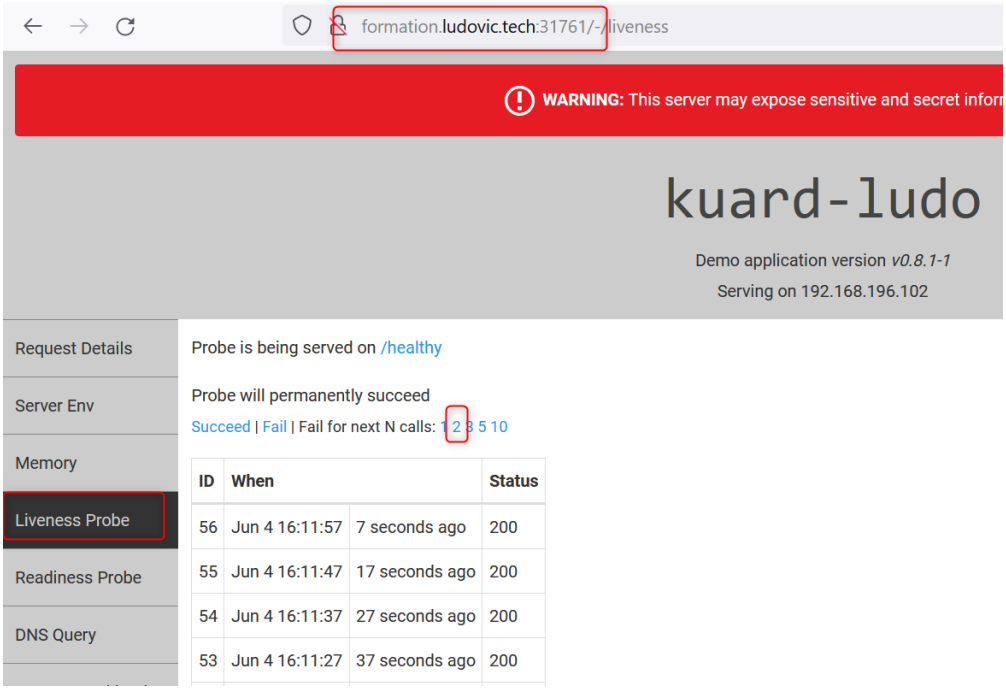
On applique le manifeste

```
ludo@kubernetes:/pod$ kubectl apply -f service-kuard.yaml
service/nodeport created
```

On liste les ressources

```
ludo@kubernetes:/pod$ kubectl get all
NAME             READY   STATUS    RESTARTS   AGE
pod/kuard-ludo   1/1     Running   0          2m10s

NAME                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
AGE
service/kuard-ludo   NodePort    10.111.216.187   <none>        8080:30101/TCP
10s
```

On teste la sonde



```
ludo@kubernetes:/pod$  watch kubectl get pod kuard-ludo

Every 2.0s: kubectl get pod
kubernetes.ludovic.tech: Sun Jun  4 18:14:13 2023

NAME         READY   STATUS    RESTARTS   AGE
kuard-ludo   1/1     Running   0          11m

NAME         READY   STATUS    RESTARTS     AGE
kuard-ludo   1/1     Running   1 (9s ago)   12m
```

La gestion des ressources.

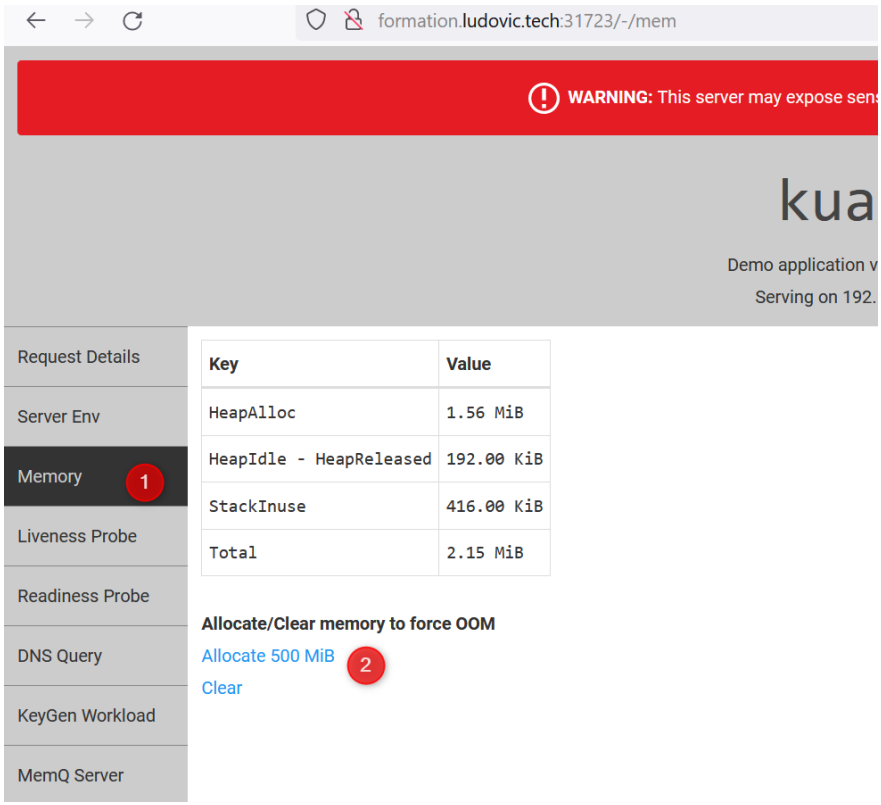
On ouvre le fichier kuard-pod-full.yaml

```
ludo@kubernetes:/pod$ vim kuard-pod-full.yaml
apiVersion: v1
kind: Pod
metadata:
  name: kuard #nom du pod
  labels:
   app: demo
  namespace: forma-ludo # nom de votre namescpace
spec:
  containers:
    - image: gcr.io/kuar-demo/kuard-amd64:1
      name: kuard
      ports:
        - containerPort: 8080
          name: http
          protocol: TCP
      resources:
        requests:
          cpu: "500m"
          memory: "128Mi"
        limits:
          cpu: "1000m"
          memory: "256Mi"
 . . . .
```

On applique après modification

```
ludo@kubernetes:/pod$ kubectl apply -f kuard-pod-full.yaml
pod/kuard configured
```

On test



Le pod redémarre

```
ludo@kubernetes:/pod$ watch kubectl get pod
Every 2.0s: kubectl get pod                 kubernetes.ludovic.tech: Mon Jun
5 18:11:33 2023

NAME    READY   STATUS    RESTARTS      AGE
kuard   1/1     Running   1 (74s ago)   113s
```