

TP KUBERNETES

Q1. Scheduling

Quand on fait un kubectl apply, le manifeste passe par l'API Server, puis est stocké dans etcd.

Les controllers détectent ce nouvel objet et créent les ressources.

Le scheduler choisit sur quel nœud lancer les Pods, et le kubelet du nœud télécharge et démarre les conteneurs.

Q2. Service / Endpoints

Le Service ClusterIP agit comme point d'entrée stable.

Il sélectionne les Pods grâce aux labels et maintient la liste de leurs IP via les EndpointSlice.

Quand un Pod arrive ou disparaît, le service se met à jour tout seul.

Q3. Probes

Readiness → indique quand le Pod est prêt à recevoir du trafic.

Liveness → vérifie si le conteneur tourne encore.

StartupProbe → évite les redémarrages prématuress.

Exemple :

```
readinessProbe:  
  httpGet: { path: "/", port: 80 }  
  periodSeconds: 10  
  timeoutSeconds: 2  
  failureThreshold: 3
```

```
livenessProbe:  
httpGet: { path: "/", port: 80 }  
periodSeconds: 10  
timeoutSeconds: 2  
failureThreshold: 5
```

Q4. DNS interne

`http://api:80/` marche car Kubernetes gère un DNS interne.

Chaque service a un nom résolu automatiquement.

Nom complet :

`http://api.demo.svc.cluster.local`

Q5. NodePort

Pratique en local pour tester facilement via `localhost:<port>`.

Mais en prod, c'est limité et peu sécurisé → Ingress est préféré pour gérer le routage HTTP/HTTPS et les domaines.

PARTIE B :

```
startupProbe:  
httpGet:  
path: /  
port: 80  
initialDelaySeconds: 3  
periodSeconds: 5  
failureThreshold: 10
```

La startupProbe permet à Kubernetes d'attendre que le conteneur ait réellement démarré avant de lancer les probes de **readiness** et **liveness**.

Sans elle, un Pod qui met quelques secondes à s'initialiser pourrait être considéré à tort comme “mort” et redémarré inutilement.

Cela rend le déploiement plus stable et évite des redémarrages prématuress lors du démarrage du service API.

```
PS C:\Users\mickael.cayroche\Documents\TP1_Guide_Stack_K8s> kubectl get deploy,rs,pods,svc -n demo -o wide
NAME          READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES           SELECTOR
deployment.apps/api   3/3     3          3          46m   echo        ealen/echo-server:latest   app=api
deployment.apps/front  1/1     1          1          46m   nginx       nginx:1.27-alpine    app=front

NAME          DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES           SELECTOR
replicaset.apps/api-54bc9c68f8  3         3         3         23m   echo        ealen/echo-server:latest   app=api,pod-template-hash=54bc9c68f8
replicaset.apps/api-8f84d84ff  0         0         0         46m   echo        ealen/echo-server:latest   app=api,pod-template-hash=8f84d84ff
replicaset.apps/front-68c4c9f5c9  1         1         1         46m   nginx       nginx:1.27-alpine    app=front,pod-template-hash=68c4c9f5c9

NAME          READY   STATUS   RESTARTS   AGE   IP           NODE   NOMINATED-NODE   READINESS   GATES
pod/api-54bc9c68f8-77z8c  1/1   Running   0          23m  10.42.0.9  k3d-s3-agent-0  <none>      <none>
pod/api-54bc9c68f8-h9bb8  1/1   Running   0          23m  10.42.1.8  k3d-s3-server-0  <none>      <none>
pod/api-54bc9c68f8-s58kl  1/1   Running   0          23m  10.42.0.8  k3d-s3-agent-0  <none>      <none>
pod/front-68c4c9f5c9-cfpb9 1/1   Running   0          46m  10.42.1.7  k3d-s3-server-0  <none>      <none>

NAME          TYPE    CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE   SELECTOR
service/api   ClusterIP  10.43.41.162  <none>        80/TCP   46m   app=api
service/front NodePort   10.43.22.226  <none>        80:30080/TCP 46m   app=front
```

```
PS C:\Users\mickael.cayroche\Documents\TP1_Guide_Stack_K8s> kubectl -n demo exec deploy/front -c wget -qO http://api:80/
{"host":{"hostname":"api","ip":::ffff:10.42.1.7,"ips":[]}, "http":{"method":"GET","baseUri":"","originalUrl":"/","protocol":"http"}, "request":{"params": {"@": "/"}, "query": {}, "cookies": {}, "body": {}}, "headers": {"host": "api:80", "user-agent": "wget", "accept": "/", "connection": "close"}, "environment": {"PATH": "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin", "HOSTNAME": "api-54bc9c68f8-77z8c", "NODE_VERSION": "20.11.0", "YARN_VERSION": "1.22.19", "API_PORT_80_TCP_ADDR": "10.43.41.162", "FRONT_SERVICE_HOST": "10.43.22.226", "KUBERNETES_PORT_": "tcp://10.43.0.1:443", "KUBERNETES_PORT_443_TCP_ADDR": "10.43.0.1", "API_PORT_80_TCP_PROTO": "tcp", "API_PORT_80_TCP_PORT": "80", "API_SERVICE_PORT_HTTP": "80", "API_PORT_80_TCP": "tcp://10.43.41.162:80", "FRONT_PORT_80_TCP_PROTO": "tcp", "FRONT_PORT_80_TCP_PORT": "80", "KUBERNETES_SERVICE_HOST": "10.43.0.1", "KUBERNETES_PORT_443_TCP": "tcp://10.43.0.1:443", "KUBERNETES_PORT_443_TCP_PROTO": "tcp", "FRONT_PORT_80_TCP_PROTO": "tcp", "FRONT_PORT_80_TCP_PORT": "80", "KUBERNETES_SERVICE_PORT": "443", "FRONT_PORT_80_TCP_PORT": "80", "FRONT_SERVICE_PORT_HTTP": "80", "FRONT_PORT_80_TCP_PROTO": "tcp", "FRONT_PORT_80_TCP_PORT": "80", "KUBERNETES_SERVICE_PORT_HTTPS": "443", "API_SERVICE_HOST": "10.43.41.162", "API_SERVICE_PORT": "80", "HOME": "/root"}}}
```

