

# ***Kubernetes***

Mastère DevOps

Par : Antoine FERRO

## **Consignes :**

### **##Producteur**

1. Créer un daemonSet nommé Producteur basé sur l'image alpine
2. Créer un volume
3. Monter le volume sur le conteneur
4. Ecrire dans le fichier index.html le nom du hostname et la date toutes les 60 secondes

### **##Consommateur**

1. Créer un déploiement nommé web (avec 3 replicas) basé sur l'image httpd
2. Monter le volume créé précédemment sur le chemin htdocs du serveur Apache

### **##Service NodePort**

1. Créer un service nommé web de type NodePort qui rassemble les pods du déploiement web

### **##Utilisation**

1. Depuis le poste de travail, exécuter toutes les minutes la commande "curl node1:30000"

## Description :

Je m'apprête à créer un daemonSet nommé Producteur basé sur l'image alpine.

Pour se faire, j'ai fait un fichier yaml : daemonset\_producteur.yml. Il est important de souligner le fait que je procède étape par étape et les fichiers finaux sont dans le fichier de ce document :

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: producteur
spec:
  selector:
    matchLabels:
      name: daemon
  template:
    metadata:
      labels:
        name: daemon
    spec:
      containers:
      - name: daemon
        image: alpine:3.12
        command: ["sh", "-c"]
```

Je vais maintenant créer un volume.

Pour créer ce volume, j'ai choisi de le nommer volume1. Je vais mettre à jour le fichier yaml avec ces lignes :

```
volumeMounts:
  - name: volume1
    mountPath: /web
volumes:
  - name: volume1
    hostPath:
      path: /web
      type: DirectoryOrCreate
```

J'applique ensuite le fichier que je viens de créer.

```
> kubectl apply -f daemonset_producteur.yml
```

Puis je vérifie que tout a bien fonctionné grâce à un `kubectl get ds,pods -o wide`.

Il faut maintenant écrire dans le fichier index.html le nom du hostname et la date toutes les 60 secondes

Pour commencer, j'initialise le fichier /web/index.html :

```
> kubectl exec -it init -- sh
```

Puis, j'ajoute également les arguments suivants au fichier :

```
args:
  - while true;
    do
      echo "`date`" > "/web/index.html";
      sleep 60;
    done
```

Voici l'intégralité du fichier créé :

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: producteur
spec:
  selector:
    matchLabels:
      name: daemon
  template:
    metadata:
      labels:
        name: daemon
    spec:
      containers:
        - name: daemon
          image: alpine:3.12
          command: ["sh", "-c"]
          args:
            - while true;
              do
                echo "`date`" > "/web/index.html";
                sleep 60;
              done
          volumeMounts:
            - name: web
              mountPath: /web
      volumes:
        - name: web
          hostPath:
            path: /web
            type: DirectoryOrCreate
```

Je dois maintenant créer un déploiement nommé web (avec 3 replicas) basé sur l'image httpd.

Le fichier se nomme deployment\_web.yml :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web
spec:
  replicas: 3
  selector:
    matchLabels:
      name: web
  template:
    metadata:
      labels:
        name: web
    spec:
      containers:
        - name: apache
          image: httpd:2.4-alpine
```

J'effectue les ajouts nécessaires à ce fichier afin que le volume soit monté sur le chemin htdocs du serveur Apache :

```
volumeMounts:
  - name: volume1
    mountPath: /usr/local/apache2/htdocs
volumes:
  - name: volume1
    hostPath:
      path: /web
      type: DirectoryOrCreate
```

J'applique les spécifications :

> *kubectl apply -f .*

Voici l'intégralité de ce fichier :

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web
spec:
  replicas: 3
  selector:
    matchLabels:
      name: web
  template:
    metadata:
      labels:
        name: web
    spec:
      containers:
        - name: apache
          image: httpd:2.4-alpine
          volumeMounts:
            - name: volume1
              mountPath: /usr/local/apache2/htdocs
      volumes:
        - name: volume1
          hostPath:
            path: /web
            type: DirectoryOrCreate
```

A présent, je vais créer un service nommé web de type NodePort qui rassemble les pods du déploiement web

Voici le contenu de mon fichier web.yml :

```
apiVersion: v1
kind: Service
metadata:
  name: web
spec:
  selector:
    app3: web
  type: NodePort
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30000
```

Je valide ensuite à l'aide d'un kubectl apply.

Je vérifie à présent que tout se soit bien passé, encore à l'aide de la commande > *kubectl get deploy,rs,pods,svc -o wide*.

Depuis mon poste de travail, j'exécute la commande "curl node1:30000" à une minute d'intervalle afin de vérifier que tout fonctionne correctement :

```
> curl node1:30000
```

*Sun Mar 21 14:37:43 UTC 2021*

```
> curl node1:30000
```

*Sun Mar 21 14:38:32 UTC 2021*