

**Kubernetes**

Mastère DevOps

Nicolas BISSIERES

Intervenant : Samir LAKHDARI Année : 2021

Table des matières

[Rappel des consignes 3](#_Toc68434155)

[Producteur 4](#_Toc68434156)

[Création d’un daemonSet producteur basé sur une image alpine 4](#_Toc68434157)

[Création du volume Web 5](#_Toc68434158)

[Monter le volume dans le conteneur 6](#_Toc68434159)

[Ecrire dans le fichier index.html le nom du hostname et la date toutes les 60 secondes 7](#_Toc68434160)

[Consommateur 8](#_Toc68434161)

[Créer un déploiement nommé web (avec 3 réplicas) basé sur l'image httpd 8](#_Toc68434162)

[Monter le volume créé précédemment sur le chemin htdocs du serveur Apache 9](#_Toc68434163)

[Service NodePort 10](#_Toc68434164)

[Créer un service nommé web de type NodePort qui rassemble les pods du deploiement web 10](#_Toc68434165)

[Utilisation 11](#_Toc68434166)

[Depuis le poste de travail, exécuter toutes les minutes la commande "curl node1:30000" 11](#_Toc68434167)

# Rappel des 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 deploiement 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 deploiement web

## Utilisation

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

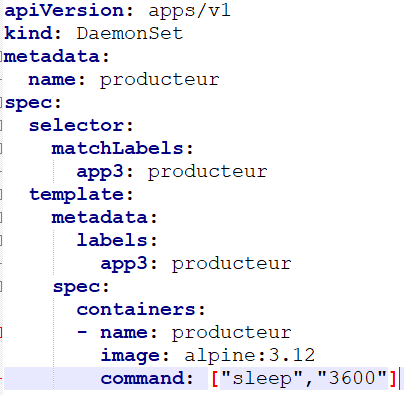
# Producteur

## Création d’un daemonSet producteur basé sur une image alpine

La première étape est de créer le fichier yaml **producteur.yml** dans mon espace de travail **K8work**.

Etant sous Windows 10 il me suffit de faire un clic droit et nouveau fichier.

Celui-ci sera un **daemonset** appelé **producteur** et basé sur une **image alpine 3.12** :



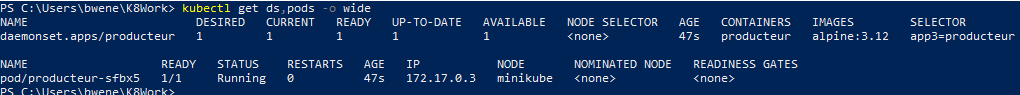
J’applique ensuite le fichier producteur.yml crée précédemment avec la commande :

**kubectl apply -f producteur.yml**

Il est à préciser que je travail sous l’environnement **minikube** installé avec le logiciel **chocolatey** comme montré lors de notre cours **Kubernetes**

Une fois le fichier appliqué, on visualise le résultat grâce à la commande

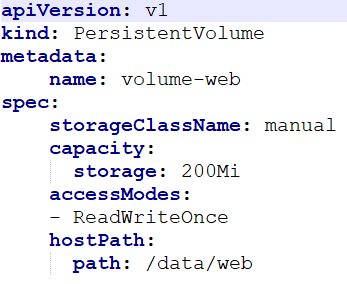
**kubectl get ds,pods -o wide**



## Création du volume Web

Il faut maintenant crée **un volume** qui sera **utilisé par le producteur** que nous nommerons **volume-web**.

Notre volume sera comme ceci :



L’objectif étant de **conserver les données** même si le conteneur se voit supprimé le type du volume sera un **volume persistant** nommé **volume-web** avec une capacité de **200Mo** accessible en **ReadWriteOnce** depuis **le chemin /data/web .**

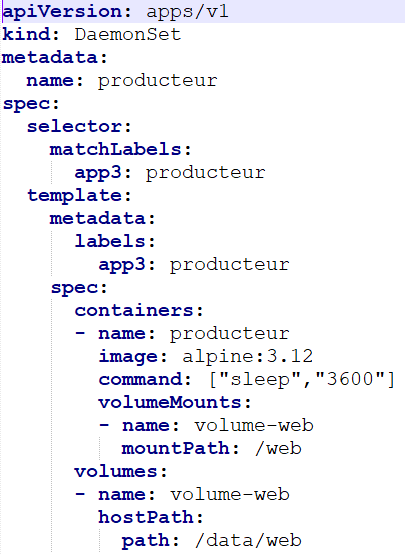
Comme d’habitude on applique notre fichier yml avec la commande **kubectl apply -f volume-web.yml**

Et on vérifie la bonne création du volume avec : **kubectl get pv**



## Monter le volume dans le conteneur

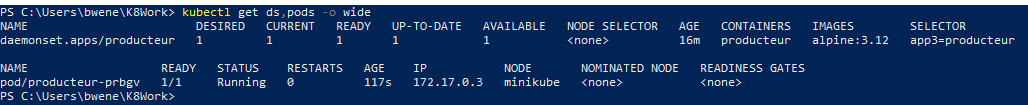
On retourne sur notre producteur.yml afin de rajouter les lignes **volumeMounts** et **volumes** pour pouvoir monter le volume crée précédemment sur notre producteur.



On n’oublie pas **d’appliquer la nouvelle configuration** de notre producteur.



Le producteur a bien redémarré :



Nous allons maintenant nous connecter dessus pour vérifier que le dossier web apparait

**kubectl exec -it producteur-prbgv -- sh**

puis ls



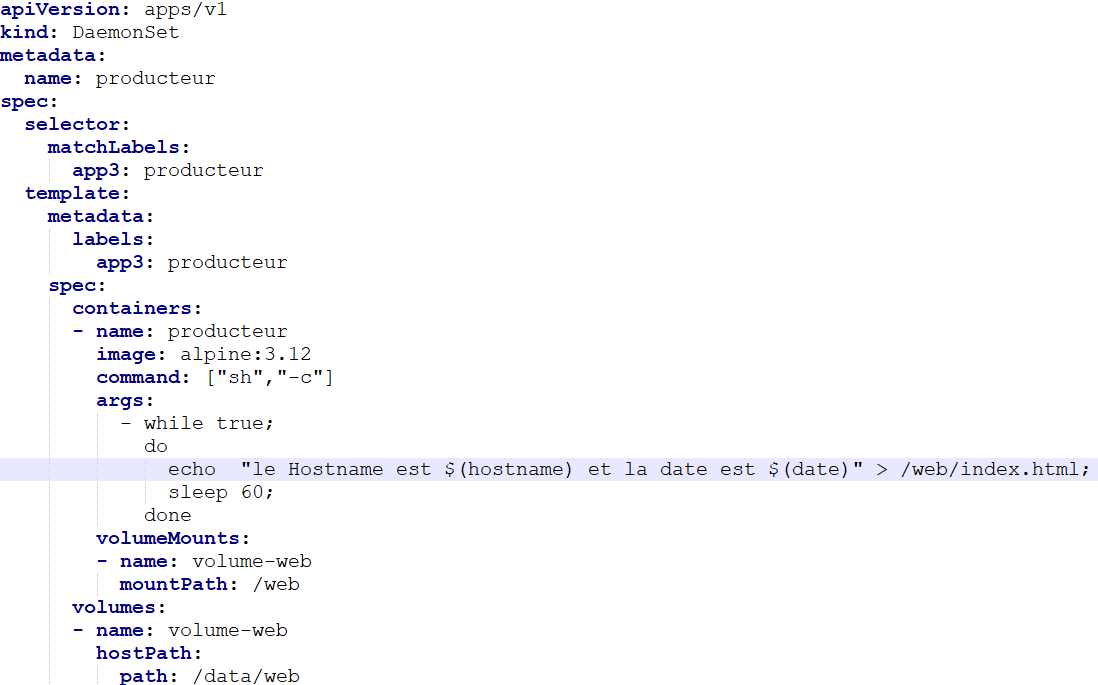
Le dossier apparait bien, pour aller dedans on fait cd web puis touch index.html afin de pouvoir continuer la suite du tp

## Ecrire dans le fichier index.html le nom du hostname et la date toutes les 60 secondes

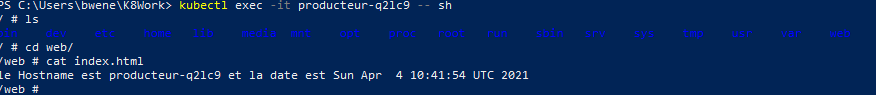
Il faut maintenant **écrire** dans le fichier **index.html** le nom du **hostname** et la **date** toutes les **60 secondes**.

Pour cela **plusieurs méthodes** sont possibles comme la création du **tache cron** ou l’utilisation d’une **boucle while**. C’est **cette dernière solution** que je vais montrer n’ayant pas réussi a utiliser cron comme je le voulais.

Nous allons retourner sur notre fichier producteur.yml et modifier la ligne **command** et ajouter en dessous la ligne **args** ce qui va nous donner ceci :



On **réapplique la configuration**, on se reconnecte dans le volume web (**attention vu qu’on réapplique une nouvelle configuration le nom du producteur change**) et on fait un **cat index.html**



La boucle est bien opérationnelle.

# Consommateur

## Créer un déploiement nommé web (avec 3 réplicas) basé sur l'image httpd

**Il faut maintenant crée le déploiement nommé web basé sur une image httpd (ici une 2.4 alpine) et ses 2 réplicas.**

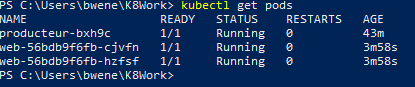
Le fichier deploiement\_web aura le contenu suivant :



On peut voir notre déploiement grâce à la commande : **kubectl get deploy**

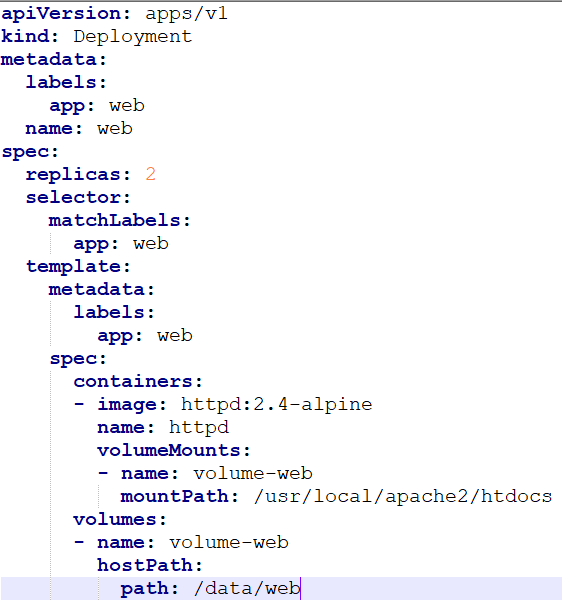


Ainsi que les pods et les replicas avec : **kubectl get pods**



## Monter le volume créé précédemment sur le chemin htdocs du serveur Apache

Il faut maintenant **modifier notre fichier deploiement\_web.yml** comme ceci afin d’ajouter le volume crée précédemment.

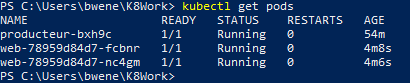


Comme on peut le voir celui-ci sera dans /usr/local/apache2/htdocs.

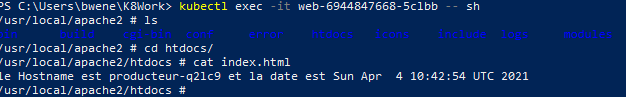
On n’oublie pas d’appliquer la nouvelle configuration :

**kubectl apply -f deploiement\_web.yml**

**De regarder le nom de nos pods :**



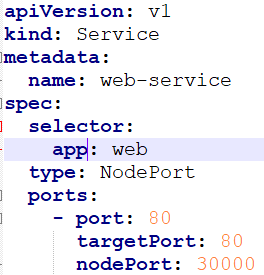
On se connecte sur un de nos replicas (ici web-78959d84d7-fcbnr), on se déplace dans l’arborescence **htdocs** et on affiche le **contenu du index.html du volume web**



# Service NodePort

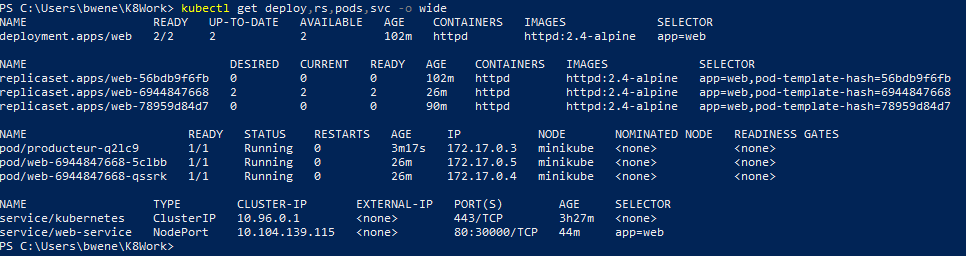
## Créer un service nommé web de type NodePort qui rassemble les pods du deploiement web

Nous allons créer un fichier qui se nommera **service\_nodeport.yml** avec le contenu suivant :

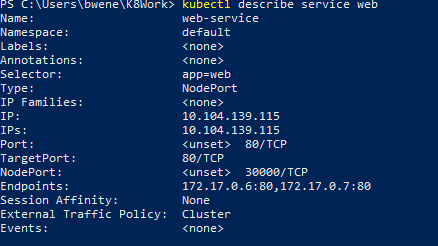


On applique le fichier : **kubectl apply -f service\_nodeport.yml**

On vérifie la bonne création avec : **kubectl get deploy,rs,pods,svc -o wide**



On peut vérifier la bonne liaison de nos 2 replica a notre service **nodeport dans le champ Endpoints** grâce a à la commande : **kubectl describe service web**



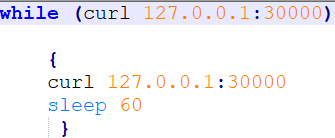
# Utilisation

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

Pour que le NodePort soit accessible depuis mon poste de travail il faut exécuter la commande **kubectl port-forward service/web-service 30000:80** dans un premier terminal



On crée ensuite un script PowerShell appelé **boucle\_curl.ps1** avec le contenu suivant afin de pouvoir faire une requête curl toutes les minutes :



On exécute le script et on constate son bon fonctionnement !

