

TP Service Cluster IP

Dans cet exercice, vous allez créer un Pod et l'exposer à l'intérieur du cluster en utilisant un Service de type *ClusterIP*.

1. Création d'un Pod

Créez un fichier `www_pod.yaml` définissant un Pod ayant les propriétés suivantes:

- nom: `www`
- label associé au Pod: `app: www` (ce label est à spécifier dans les metadatas du Pod)
- nom du container: `nginx`
- image du container: `nginx`

Créez ensuite le Pod spécifié dans `www_pod.yaml`.

2. Définition d'un service de type ClusterIP

Créez un fichier `www_service_clusterIP.yaml` définissant un service ayant les caractéristiques suivantes:

- nom: `www`
- type: `ClusterIP`
- un selector permettant le groupement des Pods ayant le label `app: www`.
- exposition du port `80` dans le cluster
- forward des requêtes vers le port `80` des Pods sous-jacents

Créez ensuite le Service spécifié dans `www_service_clusterIP.yaml`.

3. Accès au Service depuis le cluster

- Lancez le Pod dont la spécification est la suivante:

```
apiVersion: v1
kind: Pod
metadata:
  name: debug
spec:
  containers:
  - name: debug
    image: debian
    command:
```

- "sleep"
- "10000"

Nous allons utiliser ce Pod pour accéder au Service *www* depuis l'intérieur du cluster. Ce Pod contient un seul container, basé sur *alpine* et qui est lancé avec la commande *sleep 10000*. Ce container sera donc en attente pendant 10000 secondes. Nous pourrions alors lancer un shell interactif à l'intérieur de celui-ci et tester la communication avec le Service *www*.

- Lancez le Pod avec *kubectl*.
- Lancez un shell interactif *sh* dans le container *debug* du Pod.
- Utilisez *wget* pour envoyer une requête HTTP Get sur le port *80* du service *www*. Vous devriez obtenir le contenu, sous forme textuel, de la page *index.html* servie par défaut par *nginx*.

4. Visualisation de la ressource

Visualisez la spécification du service *www*.

5. Détails du service

Listez les détails du service *www*