

Création d'un Pod

Exercice

Dans cet exercice, vous allez créer une spécification pour votre premier Pod et le lancer.

1. Création de la spécification

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

- nom du Pod: *www*
- image du container: *nginx:1.14-alpine*
- nom du container: *nginx*

2. Lancement du Pod

Lancez le Pod à l'aide de *kubectl*

3. Vérification

Listez les Pods lancés et assurez vous que le Pod *www* apparait bien dans cette liste.

4. Détails du Pod

Observez les détails du Pod à l'aide de *kubectl* et retrouvez l'information de l'image utilisée par le container *nginx*.

5. Lancement d'un shell dans le container nginx

Lancez un shell dans le container *nginx* et vérifiez la version du binaire *nginx*.

Note: utilisez la commande `nginx -v`.

6. Suppression du Pod

Toujours à l'aide de *kubectl*, supprimez le Pod.

Correction

1. Création de la spécification

La spécification, définie dans le fichier *www_pod.yaml*, est la suivante:

```
apiVersion: v1
kind: Pod
metadata:
  name: www
spec:
  containers:
  - name: nginx
    image: nginx:1.14-alpine
```

2. Lancement du Pod

Le Pod peut être créé avec la commande suivante:

```
$ kubectl create -f www_pod.yaml
pod "www" created
```

3. Vérification

La commande suivante permet de lister les Pods présent:

```
$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
www       1/1     Running   0           14s
```

Note: il est aussi possible de préciser *pod* (au singulier) ou simplement *po*

```
$ kubectl get pod
NAME      READY   STATUS    RESTARTS   AGE
www       1/1     Running   0           16s

$ kubectl get po
NAME      READY   STATUS    RESTARTS   AGE
www       1/1     Running   0           22s
```

4. Details du Pod

Les details d'un Pod peuvent être obtenus avec la commande suivante:

```
$ kubectl describe pod www
Name:          www
Namespace:     default
Node:          minikube/192.168.99.100
Start Time:    Sun, 01 Apr 2018 13:08:07 +0200
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:            172.17.0.4
Containers:
  nginx:
    Container ID:
docker://3cfa8cbd0fd34121ff46fca592f327057ea9ac727f443c15d77b620fb886ac64
    Image:          nginx:1.12.2
    Image ID:       docker-
pullable://nginx@sha256:547ea435d7d719b1a18b33e1a859b3ba0c81348d2f86d1d99ca1ba9c142

    Port:          <none>
    State:         Running
      Started:     Sun, 01 Apr 2018 13:08:08 +0200
    Ready:         True
    Restart Count: 0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-brp4l
(ro)
Conditions:
  Type            Status
  Initialized     True
  Ready           True
  PodScheduled    True
Volumes:
  default-token-brp4l:
    Type:          Secret (a volume populated by a Secret)
    SecretName:    default-token-brp4l
    Optional:      false
QoS Class:        BestEffort
Node-Selectors:   <none>
Tolerations:      <none>
Events:
  Type    Reason              Age    From              Message
  ----    -
  Normal  Scheduled           4m     default-scheduler Successfully assigned
www to minikube
  Normal  SuccessfulMountVolume 4m     kubelet, minikube MountVolume.SetUp
succeeded for volume "default-token-brp4l"
  Normal  Pulled              4m     kubelet, minikube Container image
```

```
"nginx:1.12.2" already present on machine
```

Normal	Created	4m	kubelet, minikube	Created container
Normal	Started	4m	kubelet, minikube	Started container

Note: les commandes suivantes peuvent également être utilisées:

- `kubectl describe pods www`
- `kubectl describe po www`
- `kubectl describe pods/www`
- `kubectl describe pod/www`
- `kubectl describe po/www`

Dans cette sortie, on peut voir la liste des containers du Pods et l'image utilisée pour le container *nginx*.

Il est également possible d'obtenir la spécification du Pod avec la commande suivante dans laquelle on spécifie via *-o yaml* le format de sortie.

```
$ kubectl get po/www -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: 2018-04-01T11:08:07Z
  name: www
  namespace: default
  resourceVersion: "1210998"
  selfLink: /api/v1/namespaces/default/pods/www
  uid: f5280b3b-359c-11e8-80f1-080027f0e385
spec:
  containers:
  - image: nginx:1.12.2
    imagePullPolicy: IfNotPresent
    name: nginx
    resources: {}
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    volumeMounts:
    - mountPath: /var/run/secrets/kubernetes.io/serviceaccount
      name: default-token-brp4l
      readOnly: true
  dnsPolicy: ClusterFirst
  nodeName: minikube
  restartPolicy: Always
  schedulerName: default-scheduler
  securityContext: {}
  serviceAccount: default
  serviceAccountName: default
```

```
terminationGracePeriodSeconds: 30
volumes:
- name: default-token-brp4l
  secret:
    defaultMode: 420
    secretName: default-token-brp4l
status:
  conditions:
  - lastProbeTime: null
    lastTransitionTime: 2018-04-01T11:08:07Z
    status: "True"
    type: Initialized
  - lastProbeTime: null
    lastTransitionTime: 2018-04-01T11:08:08Z
    status: "True"
    type: Ready
  - lastProbeTime: null
    lastTransitionTime: 2018-04-01T11:08:07Z
    status: "True"
    type: PodScheduled
  containerStatuses:
  - containerID:
    docker://3cfa8cbd0fd34121ff46fca592f327057ea9ac727f443c15d77b620fb886ac64
    image: nginx:1.12.2
    imageID: docker-
    pullable://nginx@sha256:547ea435d7d719b1a18b33e1a859b3ba0c81348d2f86d1d99ca1ba9c142

    lastState: {}
    name: nginx
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: 2018-04-01T11:08:08Z
  hostIP: 192.168.99.100
  phase: Running
  podIP: 172.17.0.4
  qosClass: BestEffort
  startTime: 2018-04-01T11:08:07Z
```

5. Lancement d'une commande dans le container nginx

Pour lancer la commande `nginx -v` shell dans le container *nginx*, on utilise la commande suivante:

```
$ kubectl exec -ti www -- nginx -v
```

6. Suppression du Pod

Le Pod peut être supprimé avec la commande suivante:

```
$ kubectl delete po/www
```