Lab 2 : PaaS Environnement

Nom du module : PaaS Environnement **Année Universitaire** : 2021/2022 **Public cible** : 4 ArcTIC	**Enseignants** : ➢ Soumaya Mbarek ➢ Hamdi Gabsi

Objectif :

L'objectif de ce Tp est de maîtriser la mise en place d'une solution d'orchestration (Kubernetes) et de l'administrer.

I. Play-with-k8s.com

1. **Connecter-vous à k8s labs :**

   https://labs.play-with-k8s.com/

2. **Créer un cluster master node.**

3. **Initialiser votre cluster master node.**

   kubeadm init --apiserver-advertise-address $(hostname -i) --pod-network-cidr 10.5.0.0/16

Lab 2 : PaaS Environnement

4. **Initialiser votre cluster master node.**

   kubeadm init --apiserver-advertise-address $(hostname -i) --pod-network-cidr 10.5.0.0/16

5. **Initialiser votre cluster networking.**

   kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml

```
Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.0.33:6443 --token 7xtxlz.4hw192zoobtn8198 \
    --discovery-token-ca-cert-hash sha256:908abf09e6288bc75f7f7b50bc3dd8de7112d29111a257886d5bb72aa6319922
Waiting for api server to startup
Warning: resource daemonsets/kube-proxy is missing the kubectl.kubernetes.io/last-applied-configuration annotatio
n which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by eit
her kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
daemonset.apps/kube-proxy configured
No resources found
[node1 ~]$ kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubead
m-kuberouter.yaml
configmap/kube-router-cfg created
daemonset.apps/kube-router created
serviceaccount/kube-router created
clusterrole.rbac.authorization.k8s.io/kube-router created
clusterrolebinding.rbac.authorization.k8s.io/kube-router created
[node1 ~]$ kubectl get nodes
NAME    STATUS     ROLES                  AGE   VERSION
node1   NotReady   control-plane,master   52s   v1.20.1
```

6. **Déployer a nginx pod nommé 4arctic en se basant sur un fichier de définition yml.**

```
[node1 ~]$ cat  pod-def.yml
apiVersion: v1
kind: Pod
metadata:
  name: 4arctic
  labels:
    name: test
    type: pod

spec:
  containers:
    - name: web
      image: nginx
[node1 ~]$ kubectl create -f pod-def.yml
```

Lab 2 : PaaS Environnement

```
[node1 ~]$ kubectl get pods
NAME       READY    STATUS     RESTARTS    AGE
4arctic    0/1      Pending    0           116s
[node1 ~]$
```

7. **Scaling up pods avec les replica set**

   Créer un replicaset présentant 3 replica du pod 4arctic.

```
[node1 ~]$ cat replicsaet-def.yml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rep4arctic
  labels:
    app: myapp
    type: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      name: test
      type: pod
  template:
      metadata:
        name: 4arctic
        labels:
          name: test
          type: pod

      spec:
        containers:
          - name: web
            image: nginx
```

```
[node1 ~]$ kubectl create -f replicsaet-def.yml
replicaset.apps/rep4arctic created
[node1 ~]$ kubectl get replicasets
NAME         DESIRED   CURRENT   READY   AGE
rep4arctic   3         3         0       35s
[node1 ~]$
```

Lab 2 : PaaS Environnement

8. **Supprimer volontairement un pod et analyser le comportement du replica set.**

```
NAME                READY   STATUS    RESTARTS   AGE
rep4arctic-7d5db    0/1     Pending   0          8m54s
rep4arctic-q6wlh    0/1     Pending   0          8m54s
rep4arctic-xslzk    0/1     Pending   0          8m54s
[node1 ~]$
[node1 ~]$ kubectl delete pod rep4arctic-xslzk
pod "rep4arctic-xslzk" deleted
[node1 ~]$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
rep4arctic-7d5db    0/1     Pending   0          9m37s
rep4arctic-q6wlh    0/1     Pending   0          9m37s
rep4arctic-wt26p    0/1     Pending   0          4s
[node1 ~]$
```

9. **Créer un nouveau pod, et analyser le comportement du replica set.**

```
[node1 ~]$ cat pod-def.yml
apiVersion: v1
kind: Pod
metadata:
  name: 4arctic
  labels:
    name: test
    type: pod

spec:
  containers:
    - name: web
      image: nginx
[node1 ~]$ kubectl create -f pod-def.yml
pod/4arctic created
[node1 ~]$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
rep4arctic-7d5db    0/1     Pending   0          11m
rep4arctic-q6wlh    0/1     Pending   0          11m
rep4arctic-wt26p    0/1     Pending   0          2m22s
```

Lab 2 : PaaS Environnement

10. Afficher une description de différents évènements gérés par le replica set.

```
[node1 ~]$ kubectl describe replicaset
Name:         rep4arctic
Namespace:    default
Selector:     name=test,type=pod
Labels:       app=myapp
              type=frontend
Annotations:  <none>
Replicas:     3 current / 3 desired
Pods Status:  0 Running / 3 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  name=test
           type=pod
  Containers:
   web:
    Image:        nginx
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Events:
  Type    Reason            Age    From                   Message
  ----    ------            ----   ----                   -------
  Normal  SuccessfulCreate  14m    replicaset-controller  Created pod: rep4arctic-q6wlh
```

```
Events:
  Type    Reason            Age    From                   Message
  ----    ------            ----   ----                   -------
  Normal  SuccessfulCreate  14m    replicaset-controller  Created pod: rep4arctic-q6wlh
  Normal  SuccessfulCreate  14m    replicaset-controller  Created pod: rep4arctic-7d5db
  Normal  SuccessfulCreate  14m    replicaset-controller  Created pod: rep4arctic-xslzk
  Normal  SuccessfulCreate  4m53s  replicaset-controller  Created pod: rep4arctic-wt26p
  Normal  SuccessfulDelete  2m35s  replicaset-controller  Deleted pod: 4arctic
[node1 ~]$
```

11. Suite à une augmentation de notre workkoad, on souhaite faire un scale up de nombre pods à 6. Comment faire cela avec deux méthodes différentes.

12. Reprener les questions 7 8 9 10 en considérant un deployment object au lieu d'un replica set.