IT.3503 - Architecture Virtualisée

TP 2: An Introduction to Container Management and Orchestration

GUO Xiaofan

YIN Chenghao

CONTENT

1. Environment Setup

1. Install k3s

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ curl -sfL https://get.k3s.io | sh -
[sudo] password for gxf:
[INFO]  Finding release for channel stable
[INFO]  Using v1.30.6+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.30.6+k3s1/sha256sum-amd64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.30.6+k3s1/k3s
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Skipping /usr/local/bin/ctr symlink to k3s, command exists in PATH at /usr/bin/ctr
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s.service
[INFO]  systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.
[INFO]  systemd: Starting k3s
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ mkdir -p config
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ sudo cp /etc/rancher/k3s/k3s.yaml ~/Desktop/ArchitectureVietualisee/k3s-cluster/config/config.yaml
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ sudo chown $(whoami):$(whoami) ~/Desktop/ArchitectureVietualisee/k3s-cluster/config/config.yaml
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ export KUBECONFIG=~/Desktop/ArchitectureVietualisee/k3s-cluster/config/config.yaml
```

1) How many nodes your cluster contains ?

The cluster contains 1 node, named “gxf”, which is in Ready status and serves as the control plane/master node.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ kubectl get nodes
NAME   STATUS   ROLES                  AGE    VERSION
gxf    Ready    control-plane,master   5m52s  v1.30.6+k3s1
```

2) Which container runtime is used ?

The container runtime used is containerd://1.7.22-k3s1

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ kubectl get nodes -o wide
NAME   STATUS   ROLES                  AGE    VERSION        INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION     CONTAINER-RUNTIME
gxf    Ready    control-plane,master   6m5s   v1.30.6+k3s1   10.0.2.15     <none>        Ubuntu 24.04.1 LTS   6.8.0-47-generic   containerd://1.7.22-k3s1
```

3) What are the Kubernetes namespace resources defined in your cluster ?

The Kubernetes namespace resources defined in the cluster are:

Default, kube-node-lease, kube-public, kube-system

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   7m42s
kube-node-lease   Active   7m42s
kube-public       Active   7m42s
kube-system       Active   7m42s
```

4) What are the pods running on your cluster ?

The cluster is running the following pods in the kube-system namespace:

coredns-7b98449c4-q477m, helm-install-traefik-crd-5zr8g, helm-install-traefik-phzhn, local-path-provisioner-595dffc56f-wb648, metrics-server-ccdc87586-ntnmr, svclb-traefik-0af462d7-j7xvn, and **traefik-d7c9c5778-zzq99**

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ kubectl get pods --all-namespaces
NAMESPACE     NAME                                      READY   STATUS      RESTARTS   AGE
kube-system   coredns-7b98449c4-q477m                   1/1     Running     0          10m
kube-system   helm-install-traefik-crd-5zr8g            0/1     Completed   0          10m
kube-system   helm-install-traefik-phzhm                0/1     Completed   1          10m
kube-system   local-path-provisioner-595dcfc56f-wb648   1/1     Running     0          10m
kube-system   metrics-server-cdcc87586-ntmnr            1/1     Running     0          10m
kube-system   svclb-traefik-0af462d7-j7xvm              2/2     Running     0          9m21s
kube-system   traefik-d7c9c5778-zzq99                   1/1     Running     0          9m21s
```

5) What are the lists of replica sets and deployments that you have on you cluster ?

The cluster has the following replica sets: **c**oredns-7b98449c4, local-path-provisioner-595dffc56f, metrics-server-ccdc87586, and traefik-d7c9c5778

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ kubectl get replicasets --all-namespace
s
NAMESPACE     NAME                                DESIRED   CURRENT   READY   AGE
kube-system   coredns-7b98449c4                   1         1         1       11m
kube-system   local-path-provisioner-595dcfc56f   1         1         1       11m
kube-system   metrics-server-cdcc87586             1         1         1       11m
kube-system   traefik-d7c9c5778                   1         1         1       10m
```

The cluster has the following deployments: **coredns, local-path-provisioner, metrics-server,** and **traefik**

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/k3s-cluster$ kubectl get deployments --all-namespace
s
NAMESPACE     NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
kube-system   coredns                  1/1     1            1           12m
kube-system   local-path-provisioner   1/1     1            1           12m
kube-system   metrics-server           1/1     1            1           12m
kube-system   traefik                  1/1     1            1           11m
```

2. Deploy an Application

```
gxf@gxf: ~/Desktop/ArchitectureVietualisee/TP3
Every 1.0s: kubectl get pods,rs,deploy,svc -A ...  gxf: Wed Nov 27 10:18:12 2024

NAMESPACE     NAME                                         READY   STATUS
RESTARTS   AGE   IP          NODE   NOMINATED NODE   READINESS GATES
kube-system   pod/coredns-7b98449c4-q477m                  1/1     Running
0          38m   10.42.0.2   gxf    <none>           <none>
kube-system   pod/helm-install-traefik-crd-5zr8g           0/1     Completed
0          38m   10.42.0.3   gxf    <none>           <none>
kube-system   pod/helm-install-traefik-phzhm               0/1     Completed
1          38m   10.42.0.6   gxf    <none>           <none>
kube-system   pod/local-path-provisioner-595dcfc56f-wb648  1/1     Running
0          38m   10.42.0.4   gxf    <none>           <none>
kube-system   pod/metrics-server-cdcc87586-ntmnr           1/1     Running
0          38m   10.42.0.5   gxf    <none>           <none>
kube-system   pod/svclb-traefik-0af462d7-j7xvm             2/2     Running
0          37m   10.42.0.8   gxf    <none>           <none>
kube-system   pod/traefik-d7c9c5778-zzq99                  1/1     Running
0          37m   10.42.0.7   gxf    <none>           <none>

NAMESPACE     NAME                                                  DESIRED   CURR
ENT   READY   AGE   CONTAINERS              IMAGES
        SELECTOR
kube-system   replicaset.apps/coredns-7b98449c4                     1         1
      1       38m   coredns                 rancher/mirrored-coredns-coredns:1.
11.3    k8s-app=kube-dns,pod-template-hash=7b98449c4
kube-system   replicaset.apps/local-path-provisioner-595dcfc56f     1         1
```

1. Creating a Pod

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl run httpd --image=httpd:alpine
pod/httpd created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
httpd   1/1     Running   0          60s
```

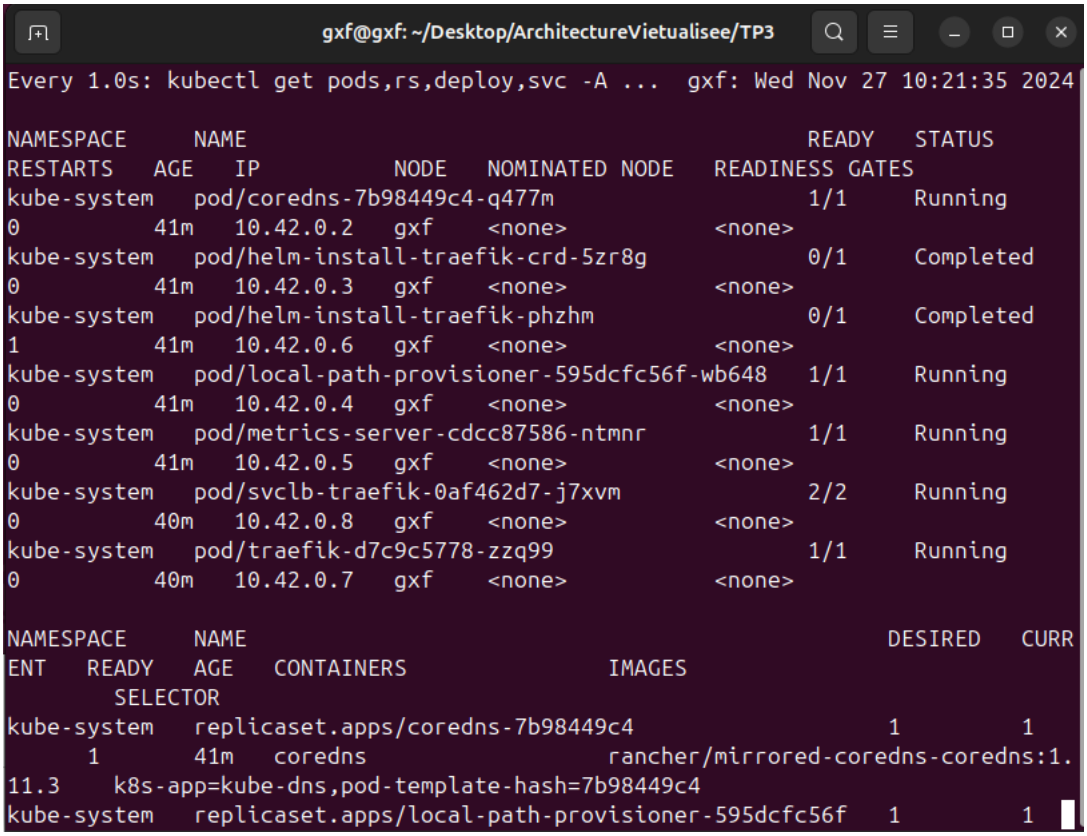
```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl describe pod httpd
Name:             httpd
Namespace:        default
Priority:         0
Service Account:  default
Node:             gxf/10.0.2.15
Start Time:       Wed, 27 Nov 2024 10:08:21 +0100
Labels:           run=httpd
Annotations:      <none>
Status:           Running
IP:               10.42.0.9
IPs:
  IP:  10.42.0.9
Containers:
  httpd:
    Container ID:   containerd://093f4a0d5e471d4ae9a27c10b2ffe1c7d6fe825f06bc6de99cbe8f888ca536c5
    Image:          httpd:alpine
    Image ID:       docker.io/library/httpd@sha256:b64b5734fbc0fbb8fb995d5cc29a2ff2d86ed4c83dfd4f4d82d183f2a66d
aed4
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 27 Nov 2024 10:08:33 +0100
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-qsx92 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-qsx92:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
```

```
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  76s   default-scheduler  Successfully assigned default/httpd to gxf
  Normal  Pulling    75s   kubelet            Pulling image "httpd:alpine"
  Normal  Pulled     65s   kubelet            Successfully pulled image "httpd:alpine" in 10.701s (10.701s incl
uding waiting). Image size: 22038396 bytes.
  Normal  Created    64s   kubelet            Created container httpd
  Normal  Started    64s   kubelet            Started container httpd
```

1) In which Kubernetes namespace your httpd pod is deployed ?

The httpd pod is deployed in the default namespace.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl delete pod httpd
pod "httpd" deleted
```



2. Using Manifest files

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ sudo apt install yamllint -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pathspec
The following NEW packages will be installed:
  python3-pathspec yamllint
0 upgraded, 2 newly installed, 0 to remove and 136 not upgraded.
Need to get 71.6 kB of archives.
After this operation, 321 kB of additional disk space will be used.
Get:1 http://fr.archive.ubuntu.com/ubuntu noble/universe amd64 python3-pathspec all 0.12.1-1 [24.5 kB]
Get:2 http://fr.archive.ubuntu.com/ubuntu noble/universe amd64 yamllint all 1.33.0-1 [47.2 kB]
Fetched 71.6 kB in 0s (178 kB/s)
Selecting previously unselected package python3-pathspec.
(Reading database ... 149385 files and directories currently installed.)
Preparing to unpack .../python3-pathspec_0.12.1-1_all.deb ...
Unpacking python3-pathspec (0.12.1-1) ...
Selecting previously unselected package yamllint.
Preparing to unpack .../yamllint_1.33.0-1_all.deb ...
Unpacking yamllint (1.33.0-1) ...
Setting up python3-pathspec (0.12.1-1) ...
Setting up yamllint (1.33.0-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3/manifests$ nano namespace.yaml
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3/manifests$ yamllint namespace.yaml
```

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/namespace.yaml
namespace/my-httpd-namespace created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/pod.yaml
pod/httpd created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME    READY   STATUS    RESTARTS   AGE
httpd   1/1     Running   0          31s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$
```

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl delete namespace my-httpd-namespace
namespace "my-httpd-namespace" deleted
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   72m
kube-node-lease   Active   72m
kube-public       Active   72m
kube-system       Active   72m
```

1) What happens when you delete a namespace ?

When delete a namespace, all resources within it, including Pods, ReplicaSets, and Deployments, are removed from the cluster.

3. Using Controllers

1) What is the role of the ReplicaSet controller ?

The ReplicaSet controller ensures that the specified number of pod replicas are running at all times. It continuously monitors the state of the Pods it manages and automatically creates new Pods if the number falls below the desired count or deletes excess Pods if there are too many. It also provides self-healing by replacing failed or terminated Pods to maintain the desired state. Additionally, it works with Deployments to facilitate updates and ensure high availability and fault tolerance for applications.

2) What is the role of a Deployment controller ?

The Deployment controller manages application updates, scaling, and self-

healing by ensuring the desired state of Pods and ReplicaSets, performing rolling updates and rollbacks, and maintaining high availability and reliability for applications.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/namespace.yaml
namespace/my-httpd-namespace created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/replicaset.yaml
replicaset.apps/my-httpd-replicaset created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get replicasets --namespace=my-httpd-namespace
NAME                DESIRED   CURRENT   READY   AGE
my-httpd-replicaset   3         3         3       25s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                        READY   STATUS    RESTARTS   AGE
my-httpd-replicaset-88gkj   1/1     Running   0          47s
my-httpd-replicaset-bfdbg   1/1     Running   0          47s
my-httpd-replicaset-jfltz   1/1     Running   0          47s
```

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                        READY   STATUS    RESTARTS   AGE
my-httpd-replicaset-88gkj   1/1     Running   0          2m53s
my-httpd-replicaset-bfdbg   1/1     Running   0          2m53s
my-httpd-replicaset-jfltz   1/1     Running   0          2m53s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl delete pod my-httpd-replicaset-88gkj --namespace=my-http
d-namespace
pod "my-httpd-replicaset-88gkj" deleted
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                        READY   STATUS    RESTARTS   AGE
my-httpd-replicaset-bfdbg   1/1     Running   0          4m10s
my-httpd-replicaset-jfltz   1/1     Running   0          4m10s
my-httpd-replicaset-xnxj9   1/1     Running   0          18s
```

```
kube-system          pod/traefik-d7c9c5778-zzq99                 1/1     Runni
ng     0          79m     10.42.0.7    gxf     <none>          <none>
my-httpd-namespace   pod/my-httpd-replicaset-bfdbg               1/1     Runni
ng     0          5m8s    10.42.0.13   gxf     <none>          <none>
my-httpd-namespace   pod/my-httpd-replicaset-jfltz               1/1     Runni
ng     0          5m8s    10.42.0.12   gxf     <none>          <none>
my-httpd-namespace   pod/my-httpd-replicaset-xnxj9               1/1     Runni
ng     0          76s     10.42.0.15   gxf     <none>          <none>

NAMESPACE            NAME                                              DESIRED
```

3) What do you notice ?

A new namespace my-httpd-namespace was created, and within it, the ReplicaSet my-httpd-replicaset successfully launched 3 Pods, all of which are running and ready as expected.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                        READY   STATUS    RESTARTS   AGE
my-httpd-replicaset-bfdbg   1/1     Running   0          4m10s
my-httpd-replicaset-jfltz   1/1     Running   0          4m10s
my-httpd-replicaset-xnxj9   1/1     Running   0          18s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl scale replicaset.apps/my-httpd-replicaset --replicas=5 -
-namespace=my-httpd-namespace
replicaset.apps/my-httpd-replicaset scaled
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                        READY   STATUS    RESTARTS   AGE
my-httpd-replicaset-bfdbg   1/1     Running   0          6m26s
my-httpd-replicaset-jfltz   1/1     Running   0          6m26s
my-httpd-replicaset-k7qns   1/1     Running   0          36s
my-httpd-replicaset-lwhpc   1/1     Running   0          36s
my-httpd-replicaset-xnxj9   1/1     Running   0          2m34s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl scale replicaset.apps/my-httpd-replicaset --replicas=2 -
-namespace=my-httpd-namespace
replicaset.apps/my-httpd-replicaset scaled
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                        READY   STATUS    RESTARTS   AGE
my-httpd-replicaset-bfdbg   1/1     Running   0          6m52s
my-httpd-replicaset-jfltz   1/1     Running   0          6m52s
```

4) What is the role of the ReplicaSet controller ?

The role of the ReplicaSet controller is to ensure the desired number of pod replicas are running. When the ReplicaSet was scaled to 5, it created additional Pods to match the desired count. Similarly, when scaled down to 2, it terminated the excess Pods to maintain the updated desired state.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/deployment.yaml
deployment.apps/my-httpd-deployment created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get deployments --namespace=my-httpd-namespace
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
my-httpd-deployment   0/3     3            0           36s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                                  READY   STATUS             RESTARTS   AGE
my-httpd-deployment-7878b4545f-55stn   0/1     ErrImagePull       0          48s
my-httpd-deployment-7878b4545f-z58h2   0/1     ErrImagePull       0          48s
my-httpd-deployment-7878b4545f-zfjzg   0/1     ImagePullBackOff   0          48s
my-httpd-replicaset-bfdbg              1/1     Running            0          9m25s
my-httpd-replicaset-jfltz              1/1     Running            0          9m25s
```

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl edit deployment my-httpd-deployment --namespace=my-httpd
-namespace
deployment.apps/my-httpd-deployment edited
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                                 READY   STATUS        RESTARTS   AGE
my-httpd-deployment-7487856846-8cvfs   1/1     Running       0          4s
my-httpd-deployment-7487856846-p4rkd   1/1     Running       0          8s
my-httpd-deployment-7487856846-prm7f   1/1     Running       0          20s
my-httpd-deployment-7878b4545f-z58h2   1/1     Terminating   0          13m
my-httpd-replicaset-bfdbg              1/1     Running       0          22m
my-httpd-replicaset-jfltz              1/1     Running       0          22m
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl edit deployment my-httpd-deployment --namespace=my-httpd
-namespace
deployment.apps/my-httpd-deployment edited
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get pods --namespace=my-httpd-namespace
NAME                                 READY   STATUS         RESTARTS   AGE
my-httpd-deployment-655db675cd-wsljx   0/1     ErrImagePull   0          5s
my-httpd-deployment-7487856846-8cvfs   1/1     Running        0          43s
my-httpd-deployment-7487856846-p4rkd   1/1     Running        0          47s
my-httpd-deployment-7487856846-prm7f   1/1     Running        0          59s
my-httpd-replicaset-bfdbg              1/1     Running        0          23m
my-httpd-replicaset-jfltz              1/1     Running        0          23m
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl rollout undo deployment my-httpd-deployment --namespace=
my-httpd-namespace
deployment.apps/my-httpd-deployment rolled back
```

Image: httpd:2.4.46-alpine & Image: httpd:2.4.150-alpine

5) What is the role of the Deployment controller ?

The Deployment controller manages application updates by ensuring the desired state of Pods. Shown as the result, it updated the image from **httpd:2.4.46**-alpine to **httpd:2.4.150-alpine** using a rolling update. When an error occurred (e.g., ErrImagePull), it allowed rolling back to the previous stable version (httpd:2.4.46-alpine).

3. Expose an Application

1) What are the different possible ways to publish a service in Kubernetes ?

In Kubernetes, services can be published using ClusterIP (internal access within the cluster), NodePort (external access via a specific port on each node), LoadBalancer (external access via a cloud provider's load balancer), or Ingress

(HTTP/HTTPS routing for external access).

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/service.yaml
service/httpd-service created
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get services --namespace=my-httpd-namespace
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
httpd-service   ClusterIP   10.43.101.101   <none>        80/TCP     14s
```

2) How can a service "knows" which deployment to expose ?

A service "knows" which deployment to expose by using labels and selectors. The service defines a selector that matches the labels assigned to the Pods created by the deployment. This allows the service to automatically route traffic to the appropriate Pods.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get services --namespace=my-httpd-namespace
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
httpd-service   ClusterIP   10.43.101.101   <none>        80/TCP     2m52s
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ curl 10.43.101.101:80
<html><body><h1>It works!</h1></body></html>
```

3) What are your httpd service endpoints ?

The httpd service endpoint is: **ClusterIP**: `10.43.101.101`

This endpoint is accessible only within the Kubernetes cluster.

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/service.yaml
service/httpd-service configured
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get services --namespace=my-httpd-namespace
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
httpd-service   NodePort    10.43.101.101   <none>        80:30080/TCP   9m14s
```

```
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl apply -f manifests/service.yaml
service/httpd-service configured
gxf@gxf:~/Desktop/ArchitectureVietualisee/TP3$ kubectl get services --namespace=my-httpd-namespace
NAME            TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
httpd-service   LoadBalancer   10.43.101.101   <pending>     80:30080/TCP   11m
```