

# Mini-projet

## Déploiement de wordpress

### # Création de fichier secret

vi mysql-password.yml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysql-password
type: Opaque
data:
  mysql_password: cGFzc3dvcmQK
  mysql_root_password: cm9vdAo=
```

### # Création de fichier deployment mysql

vi dp-mysql.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: mysql
  name: dp-mysql
spec:
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      name: mysql
      labels:
        app: mysql
      type: pod
    spec:
      containers:
        - name: mysql
          image: mysql
          env:
            - name: MYSQL_DATABASE
              value: mysql-db
            - name: MYSQL_USER
              value: user
            - name: MYSQL_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-password
                  key: mysql_password
            - name: MYSQL_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-password
                  key: mysql_root_password
          volumeMounts:
            - name: mysql-persistent-storage
              mountPath: /db-data
      volumes:
        - name: mysql-persistent-storage
          persistentVolumeClaim:
            claimName: wp-pvc
```

## # Création de fichier service clusterIP

vi svc-mysql.yml

```
apiVersion: v1
kind: Service
metadata:
  name: svc-mysql
  labels:
    name: mysql
spec:
  type: ClusterIP
  selector:
    app: mysql
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

## # Création de fichier deployment wordpress

vi dp-wordpress.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress
  labels:
    app: wordpress
spec:
  selector:
    matchLabels:
      app: wordpress
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
        - image: wordpress
          name: wordpress
          env:
            - name: WORDPRESS_DB_HOST
              value: wordpress-mysql
            - name: WORDPRESS_DB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mysql-password
                  key: mysql_password
          ports:
            - containerPort: 80
              name: wordpress
          volumeMounts:
            - name: wordpress-persistent-storage
              mountPath: /data
      volumes:
        - name: wordpress-persistent-storage
          persistentVolumeClaim:
            claimName: wp-pvc
```

### # Création de fichier PersistentVolume pv

vi wp-pv.yml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mypv
spec:
  storageClassName: mysql-wp
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /data-mysql-wp
```

### # Création de fichier PersistentVolumeClaim pvc

vi wp-pvc.yml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  storageClassName: mysql-wp
  resources:
    requests:
      storage: 100Mi
  accessModes:
    - ReadWriteOnce
```

### # Création de fichier service nodeport

vi svc-front.yml

```
apiVersion: v1
kind: Service
metadata:
  name: wp-front-service
spec:
  type: NodePort
  ports:
    - targetPort: 80
      port: 80
      nodePort: 30008
  selector:
    app: wordpress
```

### **#Lancer la création du PV et PVC**

```
kubectl apply -f wp-pv.yml
```

```
kubectl apply -f wp-pvc.yml
```

### **#Lancer la création du secret**

```
kubectl apply -f mysql-password.yml
```

### **#Lancer la création mysql et wordpress**

```
kubectl apply -f dp-mysql.yml
```

```
kubectl apply -f dp-wordpress.yml
```

### **#Lancer la création des services**

```
kubectl apply -f svc-mysql.yml
```

```
kubectl apply -f svc-front.yml
```

### **#Expose**

```
kubectl expose svc wp-front-service --name svc-front --port 80
```

### **#Vérification**

```
kubectl get deploy -o wide
```

```
kubectl get pods -o wide
```

```
kubectl get pvc -o wide
```

```
kubectl get pv -o wide
```

```
kubectl get svc -o wide
```

### **Conclusion :**

Avant de commencer, j'ai lu tout le projet pour comprendre.

J'ai commencé avec la création de tous les fichiers, puis je lance un par un les fichiers.

J'ai eu plusieurs problèmes avec les espaces et indentations dans les fichiers.

J'ai passé beaucoup de temps à régler les problèmes, du coup je n'ai pas pu finaliser le projet.

**GitHub : <https://github.com/alainlp/Kubernetes>**