Stateful Application

Deploy a WordPress site and a MySQL database using Minikube

🎯 Objective:

This lab will deploy a WordPress site and a MySQL database using Minikube. Both applications use PersistentVolumes and PersistentVolumeClaims to store data.

- Create PersistentVolumeClaims and PersistentVolumes
- Use a kustomization.yaml with
  - a Secret generator
  - MySQL resource configs
  - WordPress resource configs
- Apply the kustomization directory by kubectl apply -k ./
- Clean up

Apply and Verify

The kustomization.yaml contains all the resources for deploying a WordPress site and a MySQL database.

```
secretGenerator:
  - name: mysql-pass
    literals:
      - password=mySecureP@ssw0rd
resources:
  - mysql-deployment.yaml
  - wordpress-deployment.yaml
```

You can apply the Kustomization file by:

```
kubectl apply -k ./
```

🛠️ Steps

1. **Verify that the Secret exists by running the following command:**

```
kubectl get secrets
```

The response should be like this:

```
NAME                  TYPE                                  DATA   AGE
mysql-pass-c57bb4t7mf   Opaque                                1      9s
```

2. **Verify that a PersistentVolume got dynamically provisioned.**

```
kubectl get pvc
```

`Note:`

It can take up to a few minutes for the PVs to be provisioned and bound.

The response should be like this:

```
NAME             STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS       AGE
mysql-pv-claim   Bound     pvc-8cbd7b2e-4044-11e9-b2bb-42010a800002   20Gi       RWO            standard           77s
wp-pv-claim      Bound     pvc-8cd0df54-4044-11e9-b2bb-42010a800002   20Gi       RWO            standard           77s
```

3. **Verify that the Pod is running by running the following command:**

```
kubectl get pods
```

Note:
It can take up to a few minutes for the Pod's Status to be RUNNING.
The response should be like this:

```
NAME                                READY     STATUS    RESTARTS   AGE
wordpress-mysql-1894417608-x5dzt    1/1       Running   0          40s
```

4. **Verify that the Service is running by running the following command:**

```
kubectl get services wordpress
```

The response should be like this:

```
NAME        TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
wordpress   LoadBalancer   10.0.0.89    <pending>     80:32406/TCP   4m
```

Note:
Minikube can only expose Services through NodePort. The EXTERNAL-IP is always pending.

5. **Run the following command to get the IP Address for the WordPress Service:**

```
minikube service wordpress --url
```

The response should be like this:

```
http://1.2.3.4:32406
```

6. **Copy the IP address from Step 5, and load the page in your browser to view your site.**

You should see the WordPress set up page. If desired select English and continue the setup process.

NOTE: This lab is one of the Kubernetes Tutorials from the Kubernetes Documentation site:

```
https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/
```