📌 Deploying WordPress on Kubernetes with MySQL

This guide walks you through deploying a **WordPress** application with **MySQL** in a **Kubernetes cluster**. It includes **persistent storage** for the database and static content, ensuring data is not lost when containers restart. 🚀

🏰 Architecture

The deployment consists of:

✅ **MySQL** as the database (with Persistent Storage).
✅ **WordPress** as the frontend & backend (with Persistent Storage).
✅ **Persistent Volume Claims (PVCs)** for **database storage** and **static content**.
✅ **Secrets** for secure database credentials.
✅ **Kubernetes Services** for internal and external connectivity.

📌 Steps to Deploy

1️⃣ Create a Namespace (Optional but Recommended)

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: wordpress-app
```

Apply the namespace:

```
kubectl apply -f namespace.yaml
```

2️⃣ Create a Secret for MySQL Credentials

Store MySQL credentials securely as a **Kubernetes Secret**.

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-secret
  namespace: wordpress-app
type: Opaque
data:
  mysql-root-password: U2VjdXJlUGFzczEyMyE=    # SecurePass123! (base64 encoded)
  mysql-database: ZGI=                          # db (base64 encoded)
  mysql-username: YWRtaW4=                    # admin (base64 encoded)
  mysql-password: U2VjdXJlUGFzczEyMyE=        # SecurePass123! (base64 encoded)
```

🔐 **Note:** Use `echo -n 'your-secret' | base64` to encode secrets.

Apply the Secret:

```
kubectl apply -f mysql-secret.yaml
```

3️⃣ Create Persistent Volume Claims (PVCs)

PVC for MySQL

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
  namespace: wordpress-app
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
```

PVC for WordPress

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: wordpress-pvc
  namespace: wordpress-app
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

Apply the PVCs:

```
kubectl apply -f mysql-pvc.yaml
kubectl apply -f wordpress-pvc.yaml
```

4️⃣ Deploy MySQL Database

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql-deployment
  namespace: wordpress-app
spec:
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        image: mysql:5.7
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-root-password
        - name: MYSQL_DATABASE
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-database
        - name: MYSQL_USER
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-username
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-password
        ports:
        - containerPort: 3306
        volumeMounts:
        - name: mysql-storage
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-storage
        persistentVolumeClaim:
          claimName: mysql-pvc
```

Apply the MySQL Deployment:

```
kubectl apply -f mysql-deployment.yaml
```

MySQL Service (ClusterIP)

```
apiVersion: v1
kind: Service
metadata:
  name: mysql-service
  namespace: wordpress-app
spec:
  selector:
    app: mysql
  ports:
    - port: 3306
      targetPort: 3306
```

Apply the MySQL Service:

```
kubectl apply -f mysql-service.yaml
```

---

5 Deploy WordPress

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
  namespace: wordpress-app
spec:
  selector:
    matchLabels:
      app: wordpress
  template:
    metadata:
      labels:
        app: wordpress
    spec:
      containers:
      - name: wordpress
        image: wordpress:latest
        ports:
        - containerPort: 80
        env:
```

```
        - name: WORDPRESS_DB_HOST
          value: mysql-service
        - name: WORDPRESS_DB_USER
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-username
        - name: WORDPRESS_DB_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-password
        - name: WORDPRESS_DB_NAME
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: mysql-database
        volumeMounts:
        - name: wordpress-storage
          mountPath: /var/www/html
      volumes:
      - name: wordpress-storage
        persistentVolumeClaim:
          claimName: wordpress-pvc
```

Apply the WordPress Deployment:

```
kubectl apply -f wordpress-deployment.yaml
```

WordPress Service (NodePort)

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

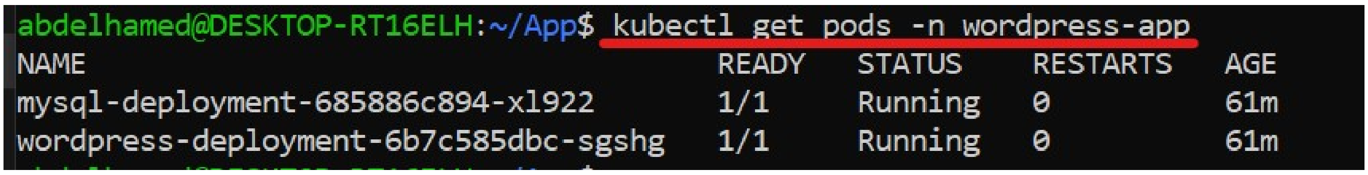
Apply the WordPress Service:

```
kubectl apply -f wordpress-service.yaml
```

🔄 Verify Deployment & Data Persistence

✅ Check Running Pods

```
kubectl get pods -n wordpress-app
```

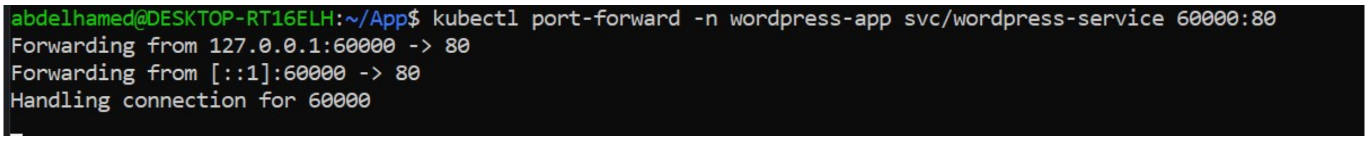


```
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl get pods -n wordpress-app
NAME                                   READY   STATUS    RESTARTS   AGE
mysql-deployment-685886c894-xl922      1/1     Running   0          61m
wordpress-deployment-6b7c585dbc-sgshg  1/1     Running   0          61m
```

✅ Access WordPress Locally

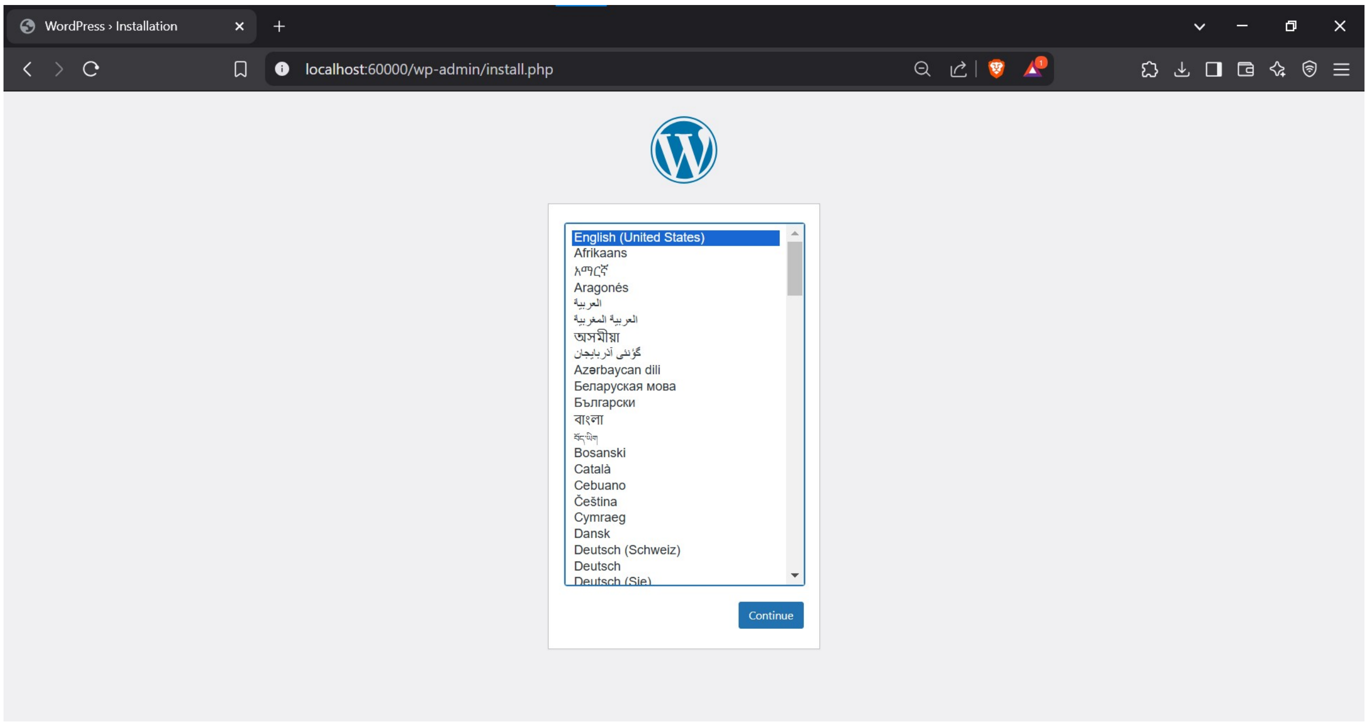
Forward the port to your local machine:

```
kubectl port-forward -n wordpress-app svc/wordpress-service 60000:80
```



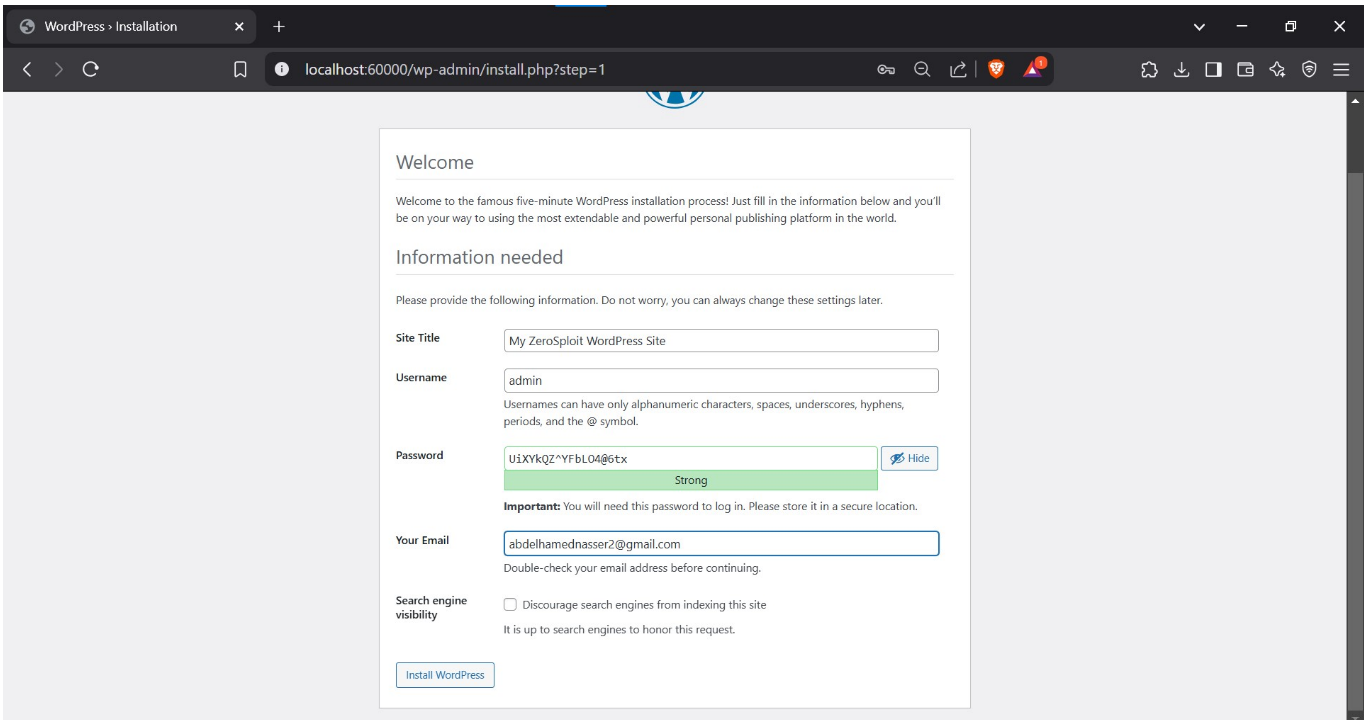
```
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl port-forward -n wordpress-app svc/wordpress-service 60000:80
Forwarding from 127.0.0.1:60000 -> 80
Forwarding from [::1]:60000 -> 80
Handling connection for 60000
```

Then open:
📌 **http://localhost:60000** in your browser to access WordPress.


WordPress › Installation

localhost:60000/wp-admin/install.php

English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch
Deutsch (Sie)


WordPress › Installation

localhost:60000/wp-admin/install.php?step=1

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

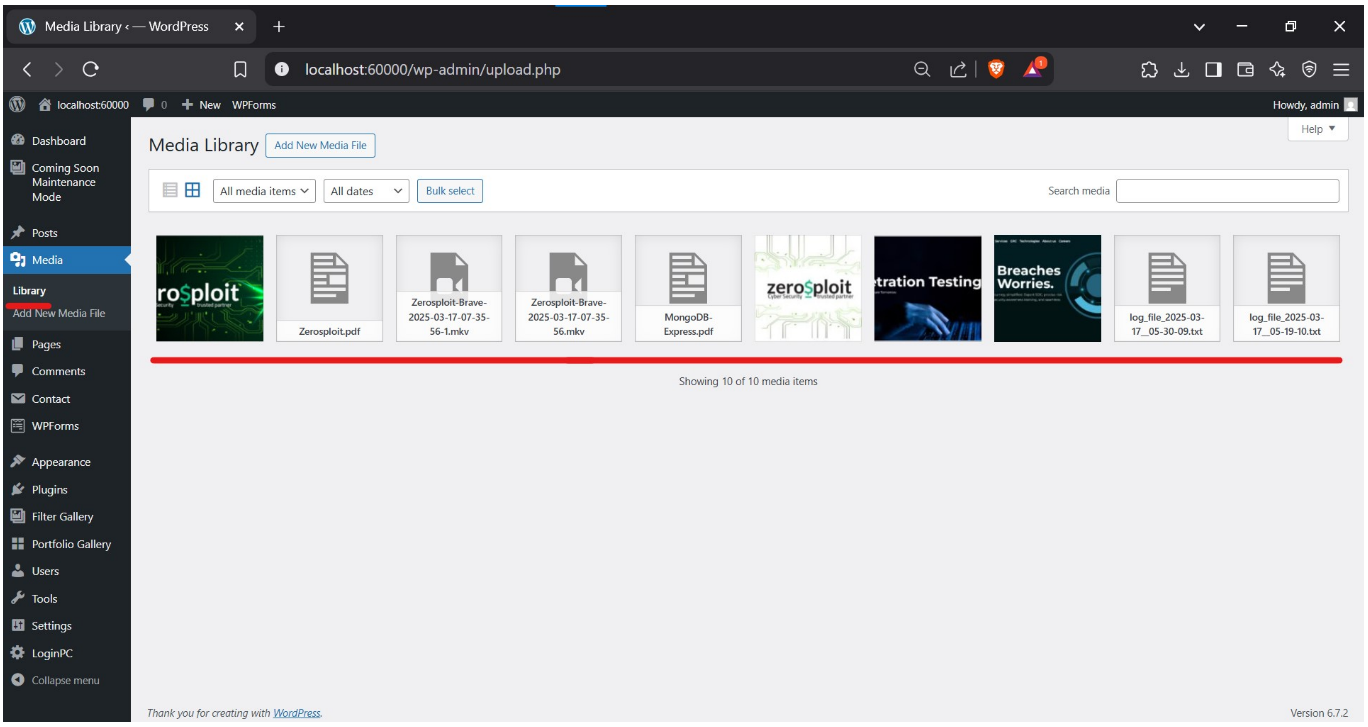
Information needed

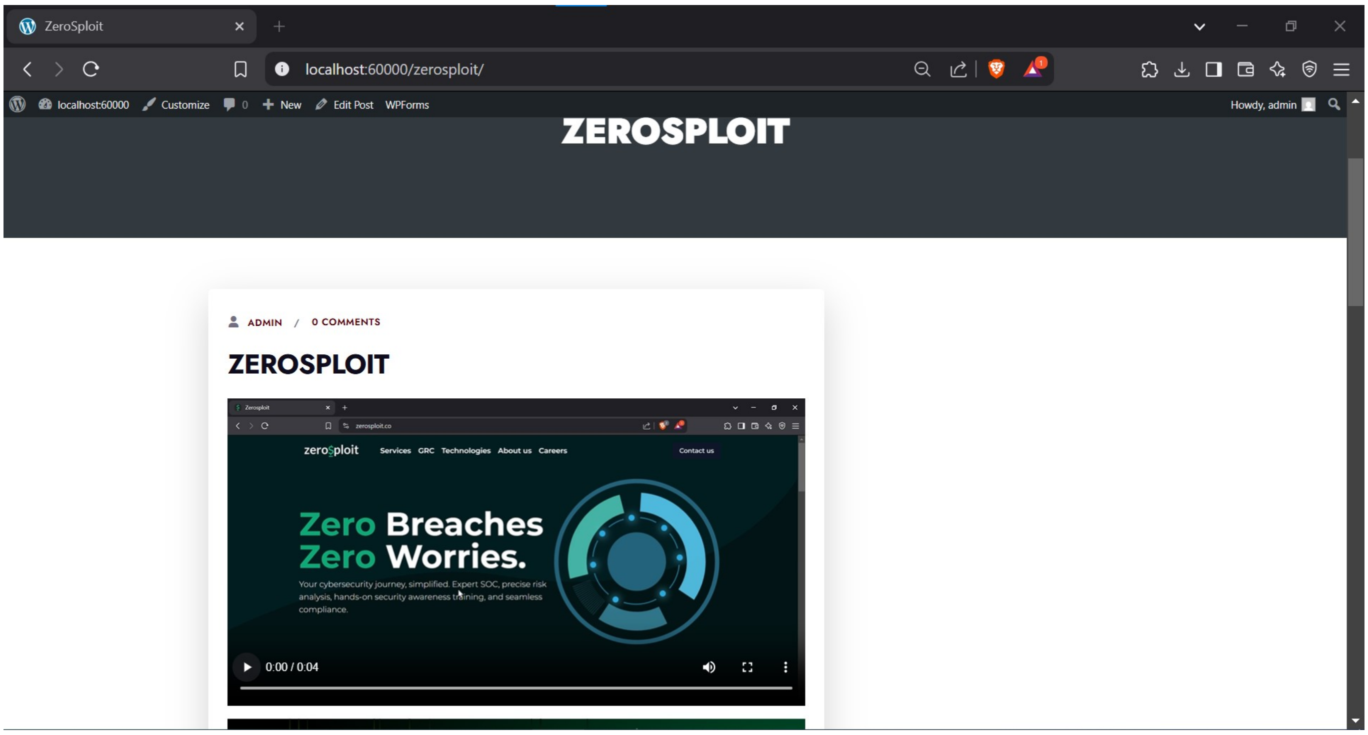
Please provide the following information. Do not worry, you can always change these settings later.

Site Title	My ZeroSploit WordPress Site
Username	admin Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	UiXYkQZ^YFbLO4@6tx Strong **Important:** You will need this password to log in. Please store it in a secure location.
Your Email	abdelhamednasser2@gmail.com Double-check your email address before continuing.
Search engine visibility	☐ Discourage search engines from indexing this site It is up to search engines to honor this request.

✅ Test Data Persistence

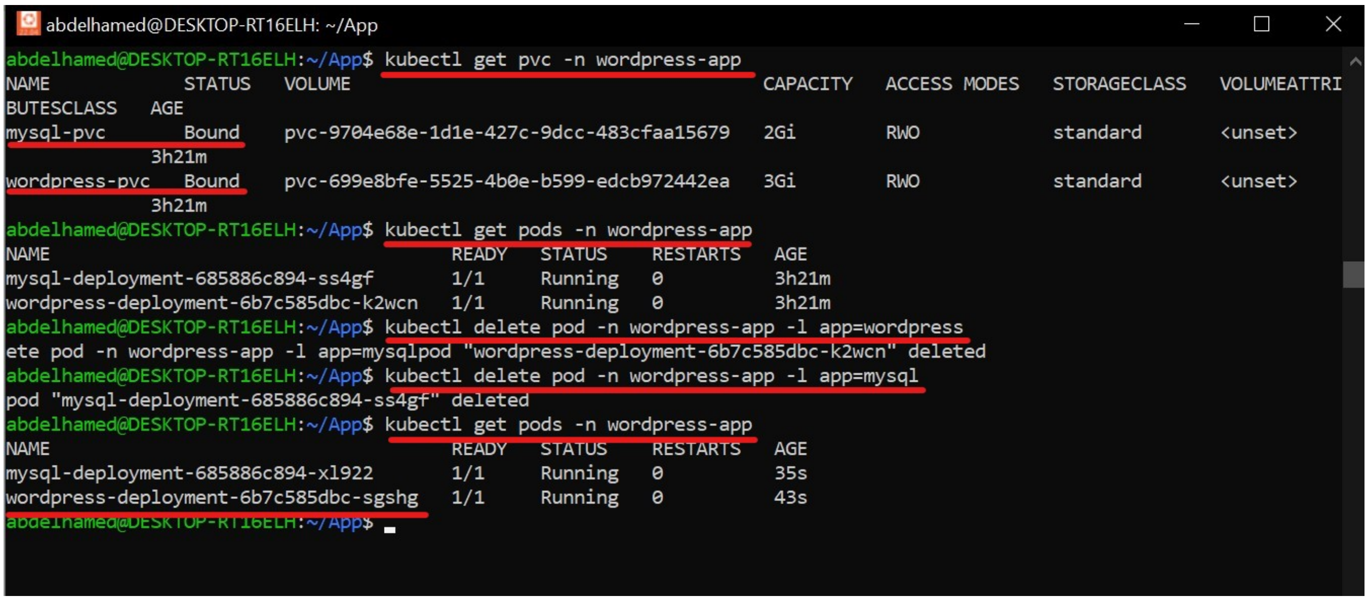
1️⃣ Upload images, videos and docs & create posts in WordPress.


Media Library ‹ — WordPress
localhost:60000/wp-admin/upload.php
Dashboard
Coming Soon Maintenance Mode
Posts
Media
Library
Add New Media File
Pages
Comments
Contact
WPForms
Appearance
Plugins
Filter Gallery
Portfolio Gallery
Users
Tools
Settings
LoginPC
Collapse menu
Media Library
Add New Media File
All media items
All dates
Bulk select
Search media
Zerosploit.pdf
Zerosploit-Brave-2025-03-17-07-35-56-1.mkv
Zerosploit-Brave-2025-03-17-07-35-56.mkv
MongoDB-Express.pdf
log_file_2025-03-17_05-30-09.txt
log_file_2025-03-17_05-19-10.txt
Showing 10 of 10 media items
Thank you for creating with WordPress.
Version 6.7.2



2 Delete the WordPress pod:

```
kubectl delete pod -n wordpress-app -l app=wordpress
kubectl delete pod -n wordpress-app -l app=mysql
```



```
abdelhamed@DESKTOP-RT16ELH: ~/App
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl get pvc -n wordpress-app
NAME            STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRI
BUTESCLASS   AGE
mysql-pvc       Bound    pvc-9704e68e-1d1e-427c-9dcc-483cfaa15679   2Gi        RWO            standard       <unset>
             3h21m
wordpress-pvc   Bound    pvc-699e8bfe-5525-4b0e-b599-edcb972442ea   3Gi        RWO            standard       <unset>
             3h21m
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl get pods -n wordpress-app
NAME                                   READY   STATUS    RESTARTS   AGE
mysql-deployment-685886c894-ss4gf      1/1     Running   0          3h21m
wordpress-deployment-6b7c585dbc-k2wcn  1/1     Running   0          3h21m
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl delete pod -n wordpress-app -l app=wordpress
ete pod -n wordpress-app -l app=mysqlpod "wordpress-deployment-6b7c585dbc-k2wcn" deleted
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl delete pod -n wordpress-app -l app=mysql
pod "mysql-deployment-685886c894-ss4gf" deleted
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl get pods -n wordpress-app
NAME                                   READY   STATUS    RESTARTS   AGE
mysql-deployment-685886c894-x1922      1/1     Running   0          35s
wordpress-deployment-6b7c585dbc-sgshg  1/1     Running   0          43s
abdelhamed@DESKTOP-RT16ELH:~/App$ _
```

3 Verify that the data is still available after the pods restart.

open the localhost again and you will find all the data:
📌 **http://localhost:60000**

✅ Test MySQL Connectivity and ensure that your WordPress application can communicate with the MySQL database

Run the following command to start a MySQL client pod and connect to the MySQL database:

```
kubectl run mysql-client --rm -it --image=mysql:5.7 --namespace=wordpress-app --
bash
```

Inside the MySQL client pod, run the following command to connect to the MySQL database:

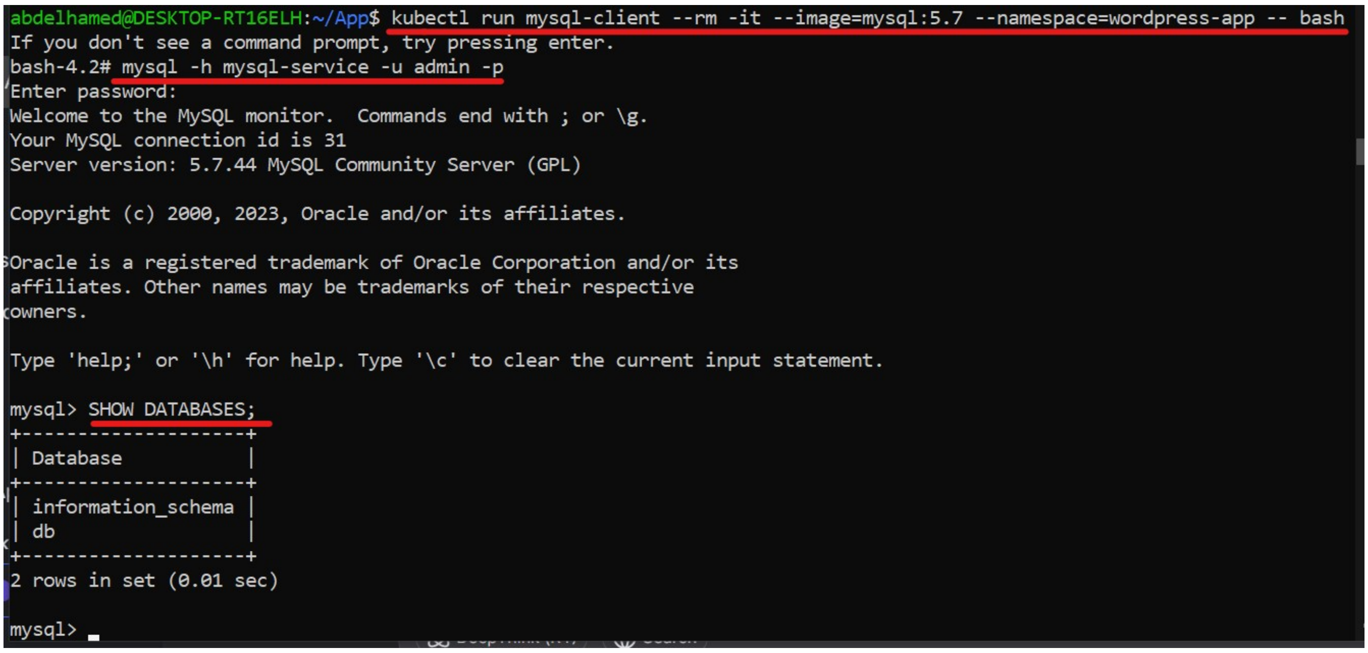
```
mysql -h mysql-service -u root -p
```

When prompted for the password, enter the decoded value of mysql-root-password:

```
SecurePass123!
```

Once connected, you can run SQL commands to verify the connection. For example:

```
SHOW DATABASES;
```



```
abdelhamed@DESKTOP-RT16ELH:~/App$ kubectl run mysql-client --rm -it --image=mysql:5.7 --namespace=wordpress-app -- bash
If you don't see a command prompt, try pressing enter.
bash-4.2# mysql -h mysql-service -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db                 |
+--------------------+
2 rows in set (0.01 sec)

mysql>
```

🎉 Success! You Have Deployed WordPress on Kubernetes 🚀

Your WordPress site is now live in Kubernetes with a persistent **MySQL database** and **stored media files** 🏆 🎊