Task9.1P

Jiakun Li *sit323*

223313972

Instructions：

Define a StorageClass (createStorageClass.yaml): This is used to define the storage type and manage policies for persistent volumes.

文本

描述已自动生成

Create a PersistentVolume (createPersistentVolume.yaml), specifying the storage capacity, access mode, and physical storage path.

文本

描述已自动生成

Next, create a PersistentVolumeClaim (createPersistentVolumeClaim.yaml) that requests storage resources matching the PersistentVolume.

文本

描述已自动生成

The deployment file (createDeployment.yaml) sets up a single replica for MongoDB, binds the previously created PVC, and sets environment variables for initializing the database user.

文本

描述已自动生成

Finally, define a Service (createService.yaml) to expose MongoDB's port within the cluster, allowing other applications to connect to the database.

文本

描述已自动生成

Verification and Testing:

Ensure all the YAML files are correctly deployed.

Use kubectl get pods to check if the MongoDB Pod's status is Running.

Use kubectl get svc to check if the service correctly exposes port 27017.

Use a MongoDB client to test the connection to the MongoDB instance.

Configuration and Deployment of MongoDB:

Step 1: Create StorageClass

First, create the StorageClass:

Step 2: Create PersistentVolume

Create the PersistentVolume:

Step 3: Create PersistentVolumeClaim

Create the PersistentVolumeClaim:

Step 4: Deploy MongoDB

Apply the deployment file:

Step 5: Create Service

Create the Service to expose MongoDB

文本

描述已自动生成

Step 6: Verify the Deployment:

a). Verify the MongoDB Pod is running

图片包含 文本

描述已自动生成

b). Verify the Service exposes the port

文本

描述已自动生成

And we can see the mongo-svc has been deployed.

Step 7: Configure the Application to Connect to MongoDB

图形用户界面, 应用程序

描述已自动生成

Step 8: Testing CRUD

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成