SIT323/SIT737- Cloud Native Application Development 9.1P: Adding a database to your application

Overview

When developing an application, it's common to work with data, files, or both. Microservices are no exception to this. In order to store dynamic data that's generated and updated by the microservice, we need to have a database. Additionally, we need to have a storage location for assets that are either served by the application or uploaded to it. For this task, your objective is to integrate a database into your existing containerized microservice application.

The required tools for doing this task are as follows:

- Git (https://github.com)
- Visual Studio (https://code.visualstudio.com/)
- Node.js (https://nodejs.org/en/download/)
- Docker
- Kubernetes // a computing platform to host your microservice
- Kubectl // the command-line tool for interacting with Kubernetes cluster
- MongoDB
- Docker Compose

Instructions

- Install MongoDB into the Kubernetes cluster, either as a standalone instance or a replica set, depending on your requirements.
- Create a MongoDB user with appropriate permissions for your application.
- Configure persistent storage for the MongoDB database by creating a Persistent Volume and Persistent Volume Claim.
- Create a Kubernetes Secret for the MongoDB user credentials and add them to the deployment manifest.
- Modify the Kubernetes deployment manifest for your application to include the newly added MongoDB database. Ensure that the configuration includes information such as the database type, credentials, and other necessary parameters.
- Configure the application to connect to the MongoDB database using the MongoDB client driver library and the connection string in the deployment manifest.
- Test the deployment to ensure that the application can connect to the MongoDB database and perform basic CRUD (Create, Read, Update, Delete) operations.
- Set up database backups and disaster recovery options as necessary.
- Monitor the MongoDB database and application performance to ensure that the database is running smoothly and efficiently.

Submission Details

- You can share your deliverables for this task through a GitHub repository, which can include the Dockerfile, Kubernetes deployment and service configuration files, and any other necessary files (https://github.com/username/ sit323/737-2024-t1-prac7p). You can also provide clear instructions on how to access and interact with the deployed application, as well as any screenshots or videos demonstrating the successful deployment and interaction.

- Ensure that detailed documentation is included with your code, offering step-by-step instructions that explain the process undertaken for this task. Failure to provide this documentation will result in an incomplete mark for the task.