DATABASE SYSTEMS AND CLOUD COMPUTING

Project Proposal #1

Due on: January 3, 2024

Project Title: Using MongoDB on Kubernetes to Build and Launch a Containerized Python Application with Flask

Objective: The purpose of this assignment is to develop a Python application that is scalable and containerized, capable of interacting with a MongoDB database, Flask, and to install it on a Kubernetes cluster.

Students will gain important experience working with database interactions, Kubernetes orchestration, and containerization in a real-world setting with this project. It goes over the basic ideas behind setting up and running containerized apps in a distributed setting.

Tasks:

Task-1: Construct MongoDB:

- Use a Kubernetes StatefulSet to deploy a MongoDB instance.
- To expose the MongoDB instance, create a Kubernetes service.
- Use BOOKSTORE database from our MongoDb Classes

Task-2: Develop Python Application: Create a Python script that carries out CRUD (Create, Read, Update, Delete) actions by connecting to the MongoDB instance.

- Employ a Python driver for MongoDB (like PyMongo) to communicate with the database.
- Use Flask tools to ensure a RESTful application
- Use Docker to containerize the Python program.

Task-3: Kubernetes Deployment:

- Establish a Python application Kubernetes deployment.
- For configuration, use Kubernetes ConfigMaps or environment variables.

Task-4: Service Discovery: Using Kubernetes Services, implement service discovery between the Python application and MongoDB.

• Verify that the MongoDB instance can be dynamically found and connected to by the Python program.

Task-5: Documentation:

• Clearly explain how to install and use the complete system in your documentation.

• Provide guidance on how to launch the Python application on Kubernetes, build and push the Docker image, and launch the MongoDB instance.

Submission:

The following must be turned in by students:

• Entire source code for the Python program.

• YAML files pertaining to StatefulSet, Deployment, and other Kubernetes resource types.

• Explanations and detailed instructions included in the documentation.

Evaluation Standards:

• Proper implementation of CRUD operations in the Python application.

• The Python + Flask application was successfully deployed on Kubernetes.

• Proper MongoDB setup and configuration on Kubernetes.

• Clear and well-organized documentation.

IMPORTANT

- Academic dishonesty, including but not limited to cheating, plagiarism, and collaboration, is unacceptable and subject to disciplinary action. Any student found guilty will have a grade of F. Assignments are due in class on the due date. Late assignments will generally not be accepted. Any exception must be approved. Approved late assignments are subject to a grade penalty.