DATABASE SYSTEMS AND CLOUD COMPUTING

Project Proposal #2

Due on: January 3, 2024

Project Title: Building and Deploying a Containerized Python Application with Flask and MongoDB on Azure Kubernetes Service (AKS)

Objective: The goal of this assignment is to create a scalable and containerized Python and Flask application that interacts with a MongoDB database and deploy the application on Azure Kubernetes Service (AKS).

Through this project, students will gain hands-on experience using Azure Kubernetes Service and other Azure services to deploy cloud-native apps. It goes over the basic ideas of database interactions in a cloud environment, container orchestration, and cloud computing.

Tasks:

Task-1: Setup MongoDB on Azure:

- Deploy a MongoDB instance on Azure using Azure Cosmos DB with the MongoDB API.
- Configure necessary network settings and authentication
- Use BOOKSTORE database from our MongoDb Classes.

Task-2: Python Application:

- Write a Python script that connects to the Azure Cosmos DB MongoDB instance and performs CRUD operations.
- Use the PyMongo driver for Python to interact with the database.
- Use Flask to ensure a RESTful application
- Containerize the Python application using Docker.

Task-3: Azure Kubernetes Service (AKS):

- Create an AKS cluster on Microsoft Azure.
- Deploy MongoDB on AKS.
- Configure a Kubernetes Service to expose the MongoDB instance.

Task-4: Python Application Deployment:

- Create a Kubernetes Deployment for the Python application on AKS.
- Use Flask to ensure a RESTful application
- Utilize Azure Container Registry (ACR) to store the Docker image.
- Configure the Deployment to use multiple replicas for scalability.
- Use Azure Kubernetes ConfigMaps for configuration.

Task-5: Service Discovery and Networking:

- Implement service discovery between the Python application and MongoDB on AKS.
- Utilize Azure Virtual Network and Network Policies for secure communication.

Task-6: Documentation:

- Provide clear documentation on how to set up and run the entire system on Azure.
- Include instructions for deploying MongoDB on Azure Cosmos DB, building and pushing the
- Docker image to ACR and deploying the Python application on AKS.

Submission: Students are required to submit the following:

- Entire source code for the Python application.
- YAML files for Kubernetes resources (Deployment, Service, StatefulSet, etc.).
- Documentation with step-by-step instructions and explanations for deploying on Azure.

Grading Criteria:

- Proper implementation of CRUD operations in the Python application.
- Correct deployment and configuration of MongoDB on Azure Cosmos DB.
- The Python + Flask application was successfully deployed on Kubernetes.
- Effective use of Azure services for container registry and logging.
- 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.