SIT323/SIT737- Cloud Native Application Development

Name: Sizhe Wang

Student ID: 223314413

9.1P: Adding a database to your application

GitHub Link: <https://github.com/AndyWanng/sit323-737-2024-t1-prac9p.git>

**Deployment Guide for Kubernetes Cluster: Task Calculator Application**

**Introduction**

This Kubernetes configuration encompasses the deployment of a multi-component application which includes a frontend server, an authentication server, a calculator server, and a MongoDB database. This setup ensures that all components are deployed in a secure and scalable fashion within a Kubernetes environment.

**Components Overview**

1. **Secrets:**
   * **jwt-secret:** Manages the JWT secret key for authentication services.
   * **mongo-secret:** Stores MongoDB credentials for database access.
2. **Deployments:**
   * **auth-server-deployment:** Handles user authentication and JWT management.
   * **calculator-server-deployment:** Provides APIs for calculator functionalities.
   * **frontend-server-deployment:** Serves the user interface.
   * **mongo:** MongoDB database deployment for data storage.
3. **Services:**
   * Expose application components within the Kubernetes cluster.
4. **Persistent Volumes:**
   * Ensure data persistence for MongoDB with a PersistentVolumeClaim.
5. **Ingress:**
   * Routes external traffic to the services based on configured paths.

**Deployment Guide**

**Prerequisites:**

* A Kubernetes cluster is up and running.
* kubectl is configured to interact with your cluster.
* Docker images for the application components are available in a registry.

**Deployment Steps:**

1. **Prepare Your Configuration Files:**
   * Ensure all your Kubernetes YAML configurations are stored in one directory. This typically includes your deployments, services, secrets, persistent volume claims, and ingress configurations.
2. **Deploy the Entire Configuration:**
   * Navigate to the directory where your Kubernetes configuration files are located.
   * Use the following command to apply all configurations at once:

*kubectl apply -f .*

1. **Check Deployment Status**:
   * Verify that all pods are running correctly:

*kubectl get pods*

* + Check the status of your services to ensure they are properly set up:

*kubectl get services*

* + Inspect the stateful sets, particularly for MongoDB:

*kubectl get statefulsets*

1. **Monitor Resource Usage and Logs:**
   * Monitor the resource usage:

*kubectl top pod*

* + Tail the logs of a specific pod if needed:

*kubectl logs -f <pod-name>*

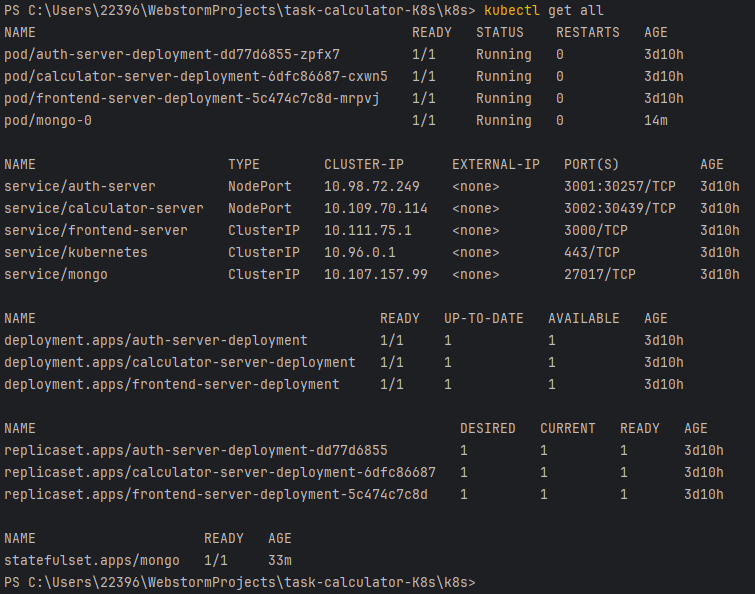
1. **Access the Application:**
   * If using an ingress controller, access your application via the URLs configured in the ingress rules.
   * Otherwise, you might need to use port-forwarding or external IPs based on your service configurations to access your application.

Screenshots:

K8s files:

文本

描述已自动生成

Pods, services, deployments and statefulsets monitoring:  


Interacting using mongosh  
文本

描述已自动生成

Using the data from mongo to login:  
图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

Web pages:

图形用户界面, 文本, 网站

描述已自动生成