Arthur Delgard – Aurélien Ah-sane

Rapport Project ST2SCL

Cloud\_Calculatrice

Table of Contents

[Rapport Project ST2SCL 1](#_Toc191649579)

[1. Présentation du projet 3](#_Toc191649580)

[2. Architecture du projet 3](#_Toc191649581)

[3. Prérequis 3](#_Toc191649582)

[4. Implémentation 4](#_Toc191649583)

[5. Problèmes rencontrés 5](#_Toc191649584)

[6. Résultats attendus 5](#_Toc191649585)

[7. Google labs – screenshots 6](#_Toc191649586)

## Présentation du projet

Pour ce projet, nous avons développé une application de calculatrice déployée sur un cluster Kubernites. L’objectif de ce projet est de mettre en place une application scalables et sécurisé en utilisant différentes technologies.

L’application permet d’effectuer plusieurs opérations mathématiques tels que : l’addition, la soustraction, la multiplication, la division, la racine carrée, et la puissance de nombres Réels via une interface web.

## Architecture du projet

L’architecture de ce projet repose sur un front et un back déployé dans des pods Kubernetes :

* **Frontend** : Développé en Vue.js.
* **Backend**: Développé avec Spring Boot.
* **Kubernetes**: Orchestration des services, création des pods et gestion du déploiement.
* **Istio**: Sécuriser la communication entre les services via mTLS

## Prérequis

Afin de déployer et exécuter l’application, il faut s’assurer d’avoir :

**Technologies nécessaires**

* **Git**
* **Docker** (<https://www.docker.com/get-started>)
* **Kubernetes Minikube** (<https://minikube.sigs.k8s.io/docs/start/>)
* **Istio** (<https://istio.io/latest/docs/setup/getting-started/>)
* **Spring Boot** (Java 17+) : Déploiement en local (<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>)
* **Node.js**: Déploiement en local (<https://nodejs.org/fr>)
* **Importer le projet Git :**

**git clone** [**https://github.com/Arthurdlg/Cloud\_calculatrice\_project.git**](https://github.com/Arthurdlg/Cloud_calculatrice_project.git)

**git checkout master**

## Implémentation

Pour effectuer ce projet, nous avons travaillé étapes par étapes :

1. **Développement du Backend en Spring Boot**
   * Création d’un contrôleur REST (CalculatriceController.java) avec plusieurs Endpoint pour chaque opération.
   * Mise en place de la gestion des requêtes et des paramètres (@RequestParam).
   * Intégration d’une base de données avec MySQL.
   * Création de l’image Docker du backend.
2. **Développement du Frontend en Vue.js**
   * Création de l’interface web avec Vue.js et intégration des boutons d’opérations.
   * Connexion au backend via fetch API pour envoyer les requêtes.
   * Création de l’image Docker du frontend.
3. **Déploiement sur kubernete**
   * Création des fichiers YAML
   * Déploiement du backend avec un service en ClusterIP
   * Déploiement du frontend avec un service en LoadBalancer
   * Déploiement de MySQL
4. **Configuration d’Istio**

* Installation d’Istio sur le cluster Kubernetes
* Création d’une Gateway Istio
* Mise en place d’un VirtualService pour router les requêtes
* Activation de mTLS via un PeerAuthentication pour sécuriser la communication entre les microservices.

## Résultats obtenus

* **Front :**

On récupère l’url avec**: minikube service calculatrice-front –url** afin d’accéder au front.

A screenshot of a computer

AI-generated content may be incorrect.

* **Back :**

On expose le back avec **kubectl port-forward service/calculatrice-service 8080:80** afin de pouvoir requêter directement dans le back. (http://localhost:8080/calculatrice/addition?a=4&b=9)

A screenshot of a computer

AI-generated content may be incorrect.

## Problèmes rencontrés

* **Communication entre le back et le front :**

En local, le front arrive bien a communiqué avec le front. Cependant, dans le cluster, il n’y a pas de communication (erreur 404). Lorsque l’on vérifie que les pods du back et du front puisse bien communiquer entre eux, on constate que l’on obtient bien une réponse du back (200).

A computer screen shot of a program

AI-generated content may be incorrect.

* **Sécurisation du cluster :**

Lorsque l’on essaie de sécuriser notre cluster en appliquant les yaml (**kubernetes/peer-authentication.yml** et **kubernetes/authorization-policy.yml**). Et que l’on utilise un tunnel minikube afin de pouvoir accéder au front (minikube tunnel), le front ne s’affiche pas.

## Google labs – screenshots

Arthur DELGARD:

A screenshot of a computer

AI-generated content may be incorrect.

Aurélien AH-SANE: