 **Protocole d'expérimentation FDI-LEEX**

# Période d'expérimentation :

Date de début :2023-02-21 Date de fin : 2023-03-24

# Membres du groupe :

* **login / poste d’expérimentation**

# Contexte d'expérimentation :

**Que voulez-vous comparer ?**

Google cloud platform ,Amazon web service et Azure

# Objet du protocole :

**Décrivez en une phrase l'objet de ce protocole expérimental**

Nous souhaitons comparer deux des plus grands fournisseurs de service cloud, dans leurs performances, coûts, facilité d’usage, en exécutant des charges de travail.

# Objectifs détaillés :

**Décrivez de façon détaillée l'objectif de la comparaison. Pourquoi faites-vous cette comparaison et que souhaitez-vous démontrer, par quel biais?**

La comparaison de Google Cloud Platform (Gcp) et Amazon Web Services (AWS) est souvent réalisée pour aider les utilisateurs et les entreprises à choisir la plateforme de cloud computing qui convient le mieux à leurs besoins. Les deux offrent des solutions de stockage, de traitement et de calcul en nuage, mais leurs caractéristiques, leurs coûts et leurs performances peuvent varier.

L'objectif de la comparaison est de mettre en évidence les forces et les faiblesses de chaque plateforme pour aider les entreprises à prendre des décisions informées sur laquelle utiliser pour leur infrastructure de cloud computing. Il peut également aider les développeurs à comprendre les différences entre les deux plateformes pour faciliter la création d'applications ou de services qui s'intègrent facilement avec l'une ou l'autre.

Les comparaisons peuvent inclure une analyse des coûts, des performances, de la facilité d'utilisation, des fonctionnalités, de la compatibilité avec les autres outils de développement, de la disponibilité des services, de la sécurité et de l'assistance technique.

Le but est de démontrer les avantages et les inconvénients de chaque plateforme de manière impartiale, en fournissant des informations détaillées sur les services offerts, les coûts et la qualité de l'expérience utilisateur. Cela permettra aux utilisateurs de prendre des décisions éclairées en fonction de leurs besoins spécifiques.

Les comparaisons peuvent être menées à travers divers biais, y compris des tests de performance en temps réel, des études de cas, des sondages et des analyses de données. Les biais doivent être évités autant que possible pour fournir des résultats précis et impartiaux qui aident les utilisateurs à prendre des décisions informées.

# Environnement de test :

* **Quelles technologies allez-vous utiliser? linux, windows, browser, vscode**

avec des outil de benchmarking adapté à notre cas d’utilisation on pourra utiliser Apache JMeter pour tester les performances de notre application Web, ou on pourra utiliser Sysbench pour tester les performances de notre base de données.

* **Quelles solutions ? Fournisseurs ? utiliser des scripts de tests, calcule de performance**

# Protocole d'expérimentation :

**Lister ici tous les tests que vous allez effectuer pour pouvoir comparer les deux technologies, paradigmes ou méthode de travail que vous avez choisis. Vous pouvez également décrire les scénarios que vous avez mis en place**.

* rapidité de hello world sur chaque service (doc bien faite?; bug ?; gratuité?).
* rapidité upload data
* temps de réponse requet
* temps de déploiement app complète
* latence du réseau
* mesure de disponibilité du service avec des outils tiers ( prometheus ?)

# Documentation :

**Lister ici les documentations que vous avez utilisées et les recherches opérées.**

* <https://cloud.google.com/docs?hl=fr>
* <https://docs.aws.amazon.com/>

# Extension possible du sujet, perspectives d'évolutions :

**Jusqu’où irait votre projet si vous pouviez aller plus loin**

* Oracle cloud, IBM cloud, alibaba cloud.

# Notes de synthèse / Observations :

Listez ici les points positifs et les points de vigilances que vous avez pu observer durant vos expérimentations.

* + Risques
  + Evolutions futures
  + Impacts
  + Taille du groupe
  + La communauté
  + La documentation
  + Facilité de prise en mains