**PHASE 1: COMPREHENSION DU SUJET**

1. **Contexte**

Dans le paysage en constante évolution du développement logiciel, l’émergence des architectures a microservices a radicalement transforme la manière dont les applications sont conçues, déployées et maintenues. Cette transitivité vers la modularité et l’indépendance des services a ouvert la voie a de nouvelles opportunités, mais a également pose des défis complexes, nécessitant une approche holistique pour garantir le succès des applications distribuées.

Notre mémoire se situe au carrefour de deux domaines essentiels du développement moderne : les approches DevOps et la surveillance distribuée des applications base sur les architectures a microservices. Cette convergence vise à relever les défis cruciaux lies a la coordination, a la performance et la sécurité dans un écosystème distribue complexe.

1. **Compréhension des concepts et mots clés**

**DevOps :** c’est une méthodologie favorisant la collaboration entre les équipes de développement et d’exploitation.

C’est en outre une combinaison de développement et d’opérations. Il permet la communication et la collaboration et comble le fossé entre ces deux groupes, ce qui permet un travail d'équipe efficace. De plus, il unifie les méthodes et outils de développement logiciel pour répondre aux interdépendances entre le développement et les opérations. La livraison de logiciels est réalisée grâce à « une rétroaction continue, une réponse rapide aux changements et l'utilisation de pipelines de livraison automatisés entraînant une réduction du temps de cycle » ( Jabbari et al., 2016 ).

C’est encore l’ensemble de pratique conçue pour réduire le temps entre les modifications et le changement du système et le déploiement des mises a jours et des modifications de production, tout en maintenant la qualité du logiciel en termes de code et de livraison.

**Surveillance : c’est** un processus ou un ensemble de processus éventuellement distribués dont la fonction est la collecte dynamique, l'interprétation et l'action sur les informations concernant une application pendant que celle-ci s'exécute. » (**Schroeder**, 1995)

Dans le contexte de DevOps par exemple, les outils de surveillance collectent des informations à différents niveaux, les analysent et fournissent des rapports avec des informations sur le système surveillé ou alertent les équipes DevOps ( Ebert et al., 2016 , Hernantes et al. , 2015 ).

**Microservice :** d’après **Fowler et Lewis**, un microservice peut être vu comme « une approche visant à développer une application unique sous la forme d'une suite de petits services, chacun s'exécutant dans son propre processus et communiquant avec des mécanismes légers, souvent une API de ressources HTTP »

De nombreuses entreprises comme **Amazon, Netflix, LinkedIn, Spotify et SoundCloud** ont modifié le style architectural sous-jacent de leurs applications vers les microservices. Cela leur permet de concevoir, développer, tester et déployer leurs logiciels de manière agile. La livraison continue de leurs applications est rendue possible par une infrastructure automatisée pour la création, les tests et le déploiement. La décentralisation de la gouvernance et de la gestion des données ainsi que le mécanisme de communication léger permettent l'indépendance technologique des différents microservices.

1. **Les défis et challenges**

Dans le contexte de l'intégration des approches DevOps et de la surveillance distribuée dans les architectures à microservices, plusieurs défis et challenges émergent, reflétant la complexité inhérente à ces environnements distribués. Voici quelques-uns des défis majeurs :

* **Coordination des Microservices :**

Le défi majeur est lie à la coordination et le déploiement de microservices indépendants de manière à assurer une intégration fluide. La nature autonome des microservices peut rendre complexe la synchronisation de leurs mises à jour et de leurs déploiements, ce qui peut entraîner des incompatibilités et des interruptions de service.

* **Communication entre Microservices :**

Défi : Assurer une communication efficace et sécurisée entre les microservices. En effet, les microservices doivent échanger des données de manière fiable tout en garantissant la confidentialité et l'intégrité, ce qui peut être complexe dans un environnement distribué.

* **Surveillance Distribuée :**

Défi : Collecter et analyser des données provenant de multiples microservices pour assurer la performance et la disponibilité. La surveillance distribuée nécessite des mécanismes sophistiqués pour suivre l'état de santé de chaque microservice et identifier rapidement les problèmes de performance.

* **Automatisation des Processus DevOps :**

Défi : Mettre en œuvre une automatisation complète des processus DevOps dans un environnement à microservices. L'automatisation est essentielle pour garantir des déploiements rapides et fiables, mais elle doit être adaptée à la diversité des microservices et de leurs dépendances.

* **Gestion de la Complexité :**

Défi : Gérer la complexité croissante due à la multiplication des microservices. Plus le nombre de microservices augmente, plus la gestion des dépendances, des versions et des interactions devient complexe, nécessitant des outils et des stratégies appropriés.

1. **Domaines scientifiques et technique que le thème va aborder**

L'évolution rapide des architectures à microservices, couplée à l'adoption généralisée des pratiques DevOps, a généré un ensemble diversifié de défis scientifiques et techniques. L'état de l'art révèle une convergence de plusieurs domaines pour aborder ces défis de manière holistique(globale).

* **Génie Logiciel et Développement Agile :** Les microservices encouragent la modularité et l'indépendance, nécessitant une adaptation des méthodologies de développement.
* **L’informatique distribue :** Les architectures à microservices sont fondamentalement distribuées, introduisant des défis liés à la coordination, à la communication, et à la cohérence des données.
* **L’automatisation et DevOps :** L'automatisation est au cœur des pratiques DevOps, mais son application efficace dans le contexte des microservices nécessite une approche spécifique.
* **Le Cloud Computing :** Les microservices sont souvent déployés dans des environnements cloud, nécessitant une adaptation des pratiques DevOps à ces infrastructures.
* **La surveillance et l’analyse de données :** La surveillance distribuée est cruciale pour garantir la performance et la disponibilité des microservices.

1. **Problème à résoudre**

Notre recherche vise à résoudre des problèmes sociétaux et scientifiques cruciaux.

**Question 1 :** Comment coordonner harmonieusement le déploiement de microservices indépendants ?

**Question 2 :** Comment garantir la communication entre ces composants distribués ?

**Question 3 :** Quelles sont les principales caractéristiques des outils de surveillance pour les systèmes basés sur des microservices ?

**Question 4 :** Quelles informations sont collectées pour caractériser le comportement du système surveillé ?

Ces questions soulignent la nécessité de trouver des solutions novatrices pour relever les défis pratiques et théoriques de ce contexte émergent.

1. **Intérêt / Objectifs à Atteindre**

L'intérêt majeur de ce travail réside dans sa capacité à adresser des problèmes cruciaux à la fois pour la société et pour la science, en vue d'améliorer la gestion des architectures à microservices à travers les approches DevOps et la surveillance distribuée.

**Dans la Société :** En intégrant les pratiques DevOps et une surveillance distribuée, ce travail vise à renforcer la robustesse des applications utilisées par les entreprises et les consommateurs. Une meilleure gestion du déploiement et une surveillance proactive contribueront à minimiser les interruptions de service et à améliorer l'expérience utilisateur.

**Pour la Science :** Ce travail contribuera à l'évolution des méthodologies de développement logiciel, en particulier dans le contexte des architectures à microservices. En identifiant des solutions innovantes et des bonnes pratiques, il enrichira le corpus de connaissances scientifiques sur la gestion de systèmes distribués.

**Objectifs à Atteindre :** mettre en place des processus DevOps optimisés et adaptés à des architectures à microservices pour assurer un déploiement rapide et fiable,faire un état de l’art sur des solutions pour la coordination efficace et la communication entre microservices autonomes et proposer des mécanismes de surveillance distribuée avancée permettant la collecte, l'analyse et la détection proactives des problèmes de performance dans un environnement distribué.