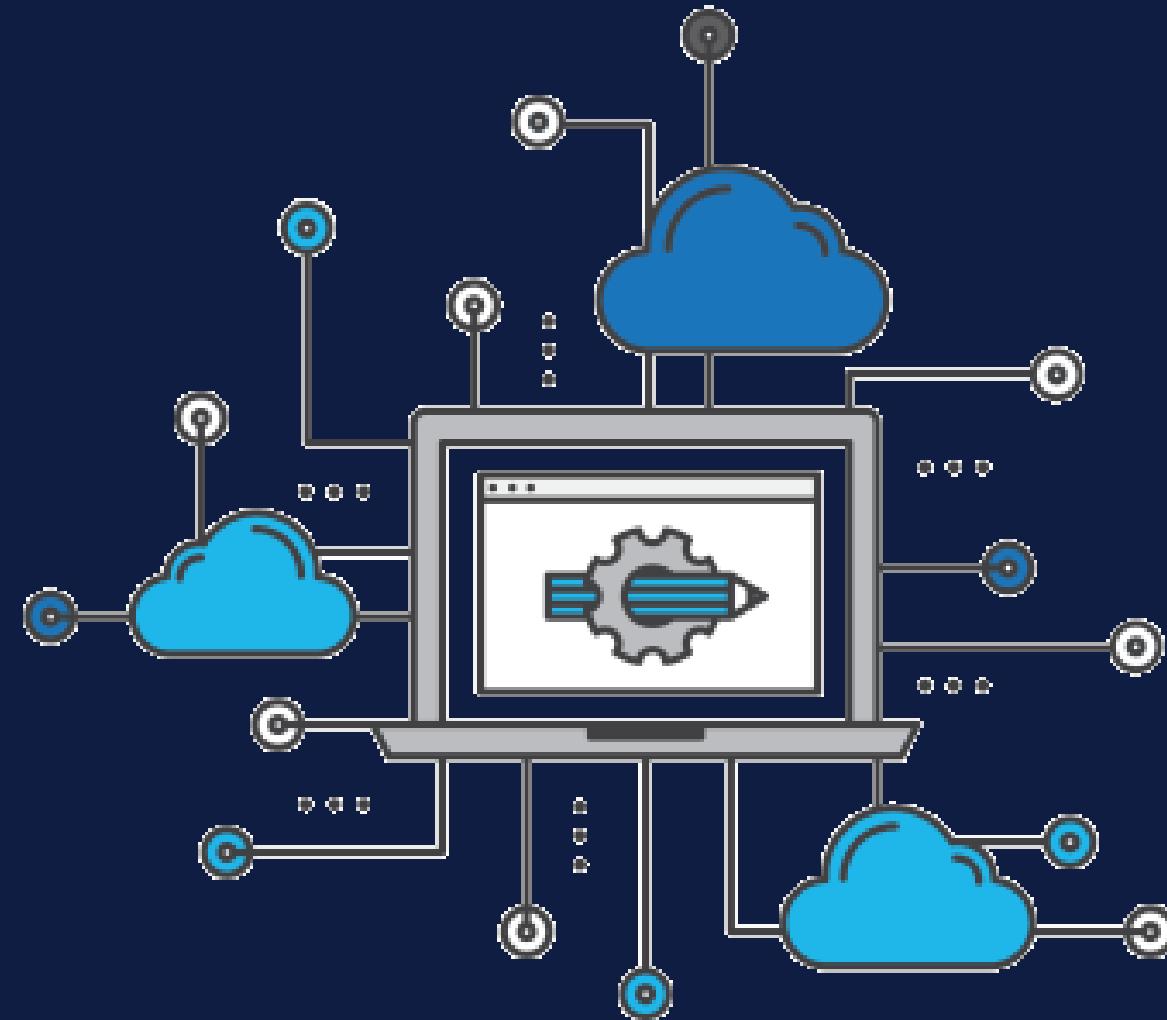


Optimisation d'une Plateforme d'Automatisation de l'Infrastructure avec Ansible & Terraform

Présentée par : Hadil Labidi

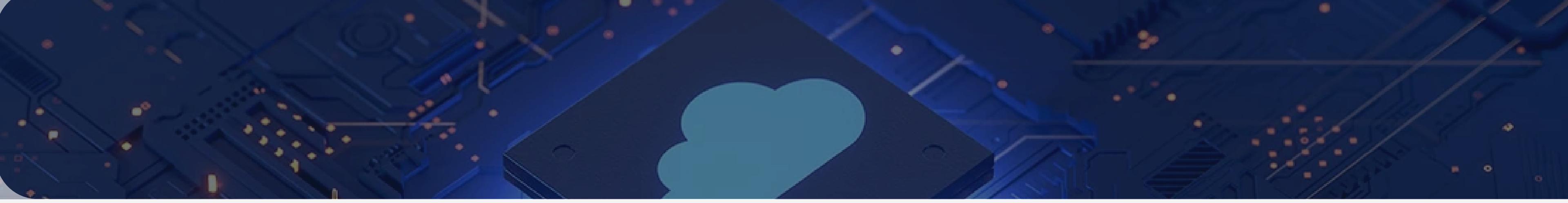


SOMMAIRE

- 01 Problématique**
- 02 Étude Critique de l'Existant**
- 03 Solution Proposée**
- 04 Les Besoins**
- 05 Technologies Utilisées**
- 06 Méthodologie de travail**
- 07 Avancement et Perspectives**

1 - Problématique





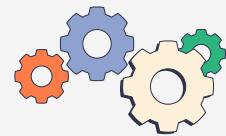
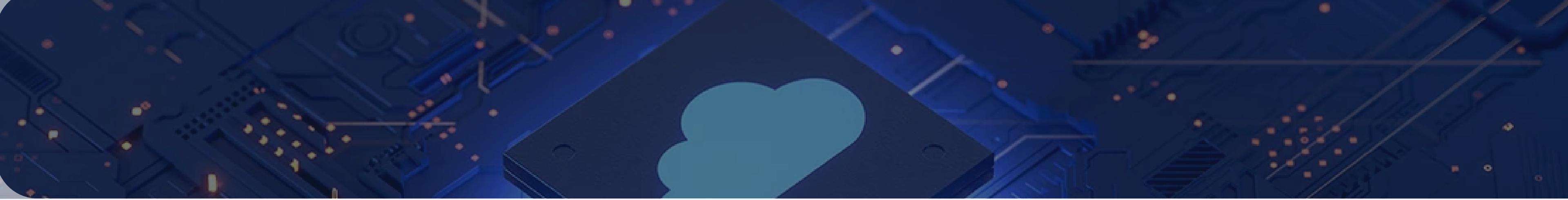
**Comment améliorer l'automatisation d'une infrastructure
cloud en intégrant un provisioning déclaratif , une
configuration avancée , et une supervision centralisée , tout en
renforçant la sécurité ?**



2- Étude Critique de l’Existant



ConfigAutomation



Fonctionnalités actuelles :

**Configuration automatisée avec Ansible,
organisée simplement en rôles et
playbooks.**



Forces :

reproductibilité et simplicité d'utilisation.



Limites :

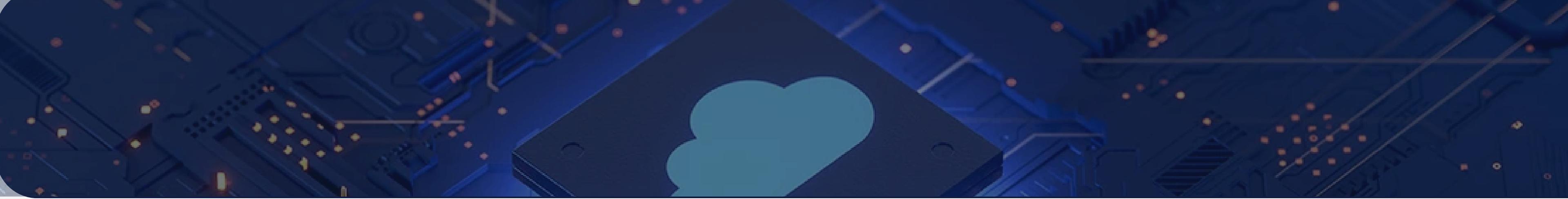
**provisioning manuel ou partiel, intégration cloud li
absente , supervision centralisée absente, faible
gestion des erreurs.**

3 - Solution Proposée





InfraPilot



Optimisation de l'existant Ansible :

**Refactoring des playbooks (modularité,
gestion d'erreurs).**



Intégration Terraform :

**Provisionnement complet de l'infrastructure
(VM, réseau, stockage) avec une
standardisation assurée par l'Infrastructure
as Code (IaC).**



Supervision & sécurité :

**Supervision centralisée avec la stack ELK, Grafana
et Prometheus, complétée par l'intégration de
Suricata pour les fonctions IDS/IPS.**



CI/CD :

**Automatisation des livraisons avec des pipelines
Jenkins .**

4- Besoins de L'entreprise



Les Besoins de Medianet

Automatisation et Optimisation de l'Infrastructure

- Réduire les interventions manuelles dans le déploiement des environnements.
- Accélérer la mise à disposition des ressources cloud (VM).
- Optimiser les coûts cloud en allouant dynamiquement les ressources selon la charge

Les Besoins de Medianet

Faciliter l'Expérience Utilisateur et la Gestion

- Proposer une interface web d'administration (Angular + Spring Boot) pour lancer des déploiements, visualiser l'état de l'infrastructure ...

Les Besoins de Medianet

Standardisation et Maintenabilité

- Standardiser les workflows DevOps grâce à des scripts Ansible et Terraform réutilisables et bien documentés.
- Rendre la plateforme évolutive avec des modules modulaires et une documentation claire.

Les Besoins de Medianet

Visibilité, Supervision et Observabilité

- Superviser en temps réel l'état de l'infrastructure avec Prometheus et Grafana.
- Centraliser les logs et événements avec Elastic Stack (Filebeat, Logstash, Kibana).
- Déetecter et prévenir les intrusions avec une solution IDS/IPS (Suricata + Elastic Stack).

5 - Technologies Utilisées



Automatisation



Terraform est utilisé pour l'automatisation de l'infrastructure en tant que code (IaC)



Ansible gère la configuration et l'orchestration des déploiements.



Jenkins

Jenkins pour automatiser l'exécution de pipelines de test de code, permettant ainsi de valider automatiquement la qualité des développements avant le déploiement.

Cloud



Google Cloud Platform

Google Cloud Platform (GCP) est utilisé pour l'hébergement de l'infrastructure.



AWS est une plateforme cloud qui fournit des services à la demande pour le calcul, le stockage et la gestion d'applications.

Supervision



Prometheus



Prometheus et Grafana sont utilisés dans mon projet pour assurer la collecte, la surveillance et la visualisation en temps réel des métriques de l'infrastructure de la plateforme.

Sécurité



Suricata est déployé comme solution IDS/IPS sur les machines virtuelles créées, avec une redirection des logs vers la stack ELK pour centraliser l'analyse et la supervision.

Développement

Front-end :



AngularJS

Back-end :

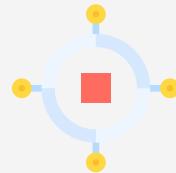


Spring Boot est utilisé pour développer le back-end de l'application, tandis qu'Angular assure la création d'une interface utilisateur moderne et réactive.

6 - Méthodologie de travail



Méthodologie Scrum



Pourquoi SCRUM ?

- **Méthode Agile, adaptée aux projets évolutifs.**
- **Approche itérative et collaborative.**
- **Permet une meilleure productivité et une livraison continue.**



Organisation en Sprints :

Chaque Sprint suit ce cycle :

1. Planification
2. Développement
3. Revue (Review)
4. Rétrospective



Rôles SCRUM dans le projet :

Mr Zied tlili a joué le rôle de Product Owner, chargé de définir les besoins et priorités.

Le Scrum Master, Zied Tlili, assurait le respect de la méthode SCRUM.

En tant que membre de la Scrum Team, j'ai contribué au développement, à la configuration et au déploiement.



Les artefacts SCRUM :

- **Le Product Backlog contient l'ensemble des fonctionnalités à développer.**
- **Le Sprint Backlog regroupe les tâches à réaliser pendant un Sprint.**
- **L'Increment Product correspond aux fonctionnalités finalisées et validées à la fin de chaque Sprint.**

5 - Avancement et Perspective

Avancement

Les Taches déjà faits

- Configuration Ansible : mise en place d'un environnement automatisé avec des playbooks pour installer et configurer les services.
- Historique des actions utilisateur : les actions des utilisateurs sont enregistrées sous forme de logs.
- Cloud avec Terraform (GCP) : infrastructure déployée sur Google Cloud Platform à l'aide de Terraform.

Avancement

Les Taches a faire

- Intégration de l'ELK Stack : prévue pour analyser les logs générés.
- Intégration de prometheus et grafana pour la supervision
- Les playbooks Suricata (IDS/IPS) et Filebeat (collecte des logs) sont déjà prêts.
- Pipelines Jenkins : automatisation des déploiements et intégration continue à mettre en place.

Prospective

Perspectives et Améliorations

- Extension multi-cloud avec d'autres fournisseurs cloud pour plus de flexibilité.
- Optimisation des playbooks Ansible
- Renforcement de la sécurité
- Automatisation avancée des pipelines CI/CD
- Ajouter des tests automatiques, scans de sécurité, et déploiements conditionnels pour améliorer la fiabilité.

Conclusion