📚 Projet de Recherche : CI/CD & Terraform

🎯 Objectif du projet

L'objectif de ce projet est d'explorer et d'analyser les **outils CI/CD et Terraform** afin de produire une **documentation détaillée et critique**.

Vous devez adopter une approche **réflexive et analytique** pour comprendre **les enjeux du déploiement et de l'infrastructure en tant que code (IaC)**.

⚠ Aucune mise en pratique n'est demandée : ce projet est uniquement basé sur la recherche et l'analyse.

📌 **Livrable attendu** :

- Un **manuel complet** structuré sous forme de documentation technique.
- Un **dépôt GitLab** contenant un site **Docusaurus** hébergé sur **GitLab Pages** pour une lecture facile.

📝 Organisation du travail

Le projet est divisé en **deux grandes parties** :

- 🔍 **Partie 1 : Étude et comparaison des outils CI/CD**
- 🏗 **Partie 2 : Recherche sur Terraform et Infrastructure as Code (IaC)**

Chaque groupe travaillera sur une section et devra produire un **document structuré, clair et argumenté**.

🔍 Partie 1 : Recherche sur le CI/CD

📌 1️⃣ Comparaison des outils CI/CD

🛠️ **Outils à comparer** :

- **GitLab CI/CD**
- **GitHub Actions**
- **Jenkins**
- **Travis CI**

📌 **Travail à réaliser** :

- Comparer ces outils selon les critères suivants :
    1. **Facilité d'utilisation** (prise en main, documentation, UX)
    2. **Coût et modèle économique** (open-source, SaaS, limitations des versions gratuites)
    3. **Flexibilité** (personnalisation, compatibilité avec d'autres outils)
    4. **Scalabilité** (capacité à gérer plusieurs projets complexes)
    5. **Cas d'usage idéal** (startup, grande entreprise, projet open-source…)

🎯 **Livrable** :

- Un **tableau comparatif clair et synthétique**
- Une **analyse argumentée** sur les forces et faiblesses de chaque outil

---

📌 2️⃣ Recherche sur les stratégies de déploiement

🔄 **Stratégies à analyser** :

- **Blue/Green Deployment**
- **Canary Deployment**
- **Rolling Updates**

📌 **Travail à réaliser** :

- Expliquer chaque stratégie avec un **exemple concret**.
- Identifier les **avantages et inconvénients** de chaque approche.
- Discuter **quel type de projet nécessite quelle stratégie**.

🎯 **Livrable** :

- Un **document analytique** avec des comparaisons et des schémas explicatifs.
- Une **recommandation pour les projets Symfony (ou autres)**.

📌 3️⃣ Étude des runners GitLab CI/CD

🚀 **Travail de réflexion sur les runners GitLab CI/CD** :

- **Différences entre un runner partagé et un runner privé**.
- **Pourquoi et quand choisir un runner privé ?**.
- **Impact sur la sécurité et la performance des pipelines CI/CD**.

🎯 **Livrable** :

- Un **article détaillé** mettant en avant les enjeux des runners.
- Une **analyse critique sur les implications en entreprise**.

🏗️ Partie 2 : Recherche sur Terraform et l'Infrastructure as Code (IaC)

📌 1️⃣ Étude des bases de Terraform

🔷 **Concepts clés à expliquer** :

- Infrastructure-as-Code (IaC) : définition et enjeux.
- Fonctionnement de Terraform (fichiers `.tf`, HCL, notion de **state**).
- Pourquoi stocker le state sur **S3/DynamoDB** ou un **backend distant** ?

📌 **Travail à réaliser** :

- Une **explication pédagogique** de Terraform, avec des analogies et des mises en situation.
- Une **comparaison avec des approches classiques** (scripts Bash, Ansible).

🎯 **Livrable** :

- Un **document expliquant les fondamentaux de Terraform**, lisible par un débutant.

📌 2️⃣ Terraform vs Autres Outils d'IaC

📌 **Outils à comparer** :

- **Terraform**
- **Ansible**
- **CloudFormation (AWS)**
- **Pulumi**

📌 **Axes de réflexion** :

1. Quelle approche ces outils adoptent-ils ? (**Déclaratif vs Impératif**)
2. Facilité d'apprentissage et d'utilisation.
3. Cas d'usage typique de chaque outil.
4. Limitations et points faibles de chaque solution.

🎯 **Livrable** :

- Un **tableau comparatif** + analyse critique des situations où chaque outil est le plus adapté.

📌 3️⃣ Terraform et l'Infrastructure Managée (AWS, Vercel, Coolify, etc.)

📌 **Problématique** : Pourquoi utiliser Terraform au lieu de services managés comme **AWS Elastic Beanstalk, Vercel, Coolify** ?

📌 **Travail à réaliser** :

- Comparer **infrastructure managée vs Terraform**.
- Expliquer les avantages (**flexibilité, indépendance**) et inconvénients (**complexité, maintenance**) de Terraform.
- Réflexion sur **les coûts et la scalabilité**.

🎯 **Livrable** :

- Une **analyse critique des choix d'architecture** selon le type d'entreprise.

🏆 Évaluation et Critères de Notation

Critères	Points
Qualité de la recherche et des explications (clarté, profondeur, structuration)	25
Comparaison détaillée des outils CI/CD (tableaux, analyses, recommandations)	15
Étude des stratégies de déploiement et des runners GitLab	15
Explication et comparaison des outils Terraform/IaC	20
Réflexion sur Terraform vs infrastructure managée	15
Présentation et structuration dans Docusaurus (qualité de la doc)	10

Total : 100 points 🎯

🚀 Modalités de rendu

✅ **Chaque groupe devra :**

1. **Créer un dépôt GitLab** contenant :
    - Les documents de recherche en format Markdown.
    - Une documentation **Docusaurus** pour rendre la lecture plus interactive.
    - Le tableau comparatif et les analyses sous forme d'articles.
2. **Héberger leur documentation sur GitLab Pages**.
3. **Présenter leur travail sous forme d'un mini-rapport final.**

📅 Date de rendu

🛑 **Deadline : 18 mars 2025**

Tout travail remis en retard sera pénalisé.

Bon courage à tous ! 🚀 🔥