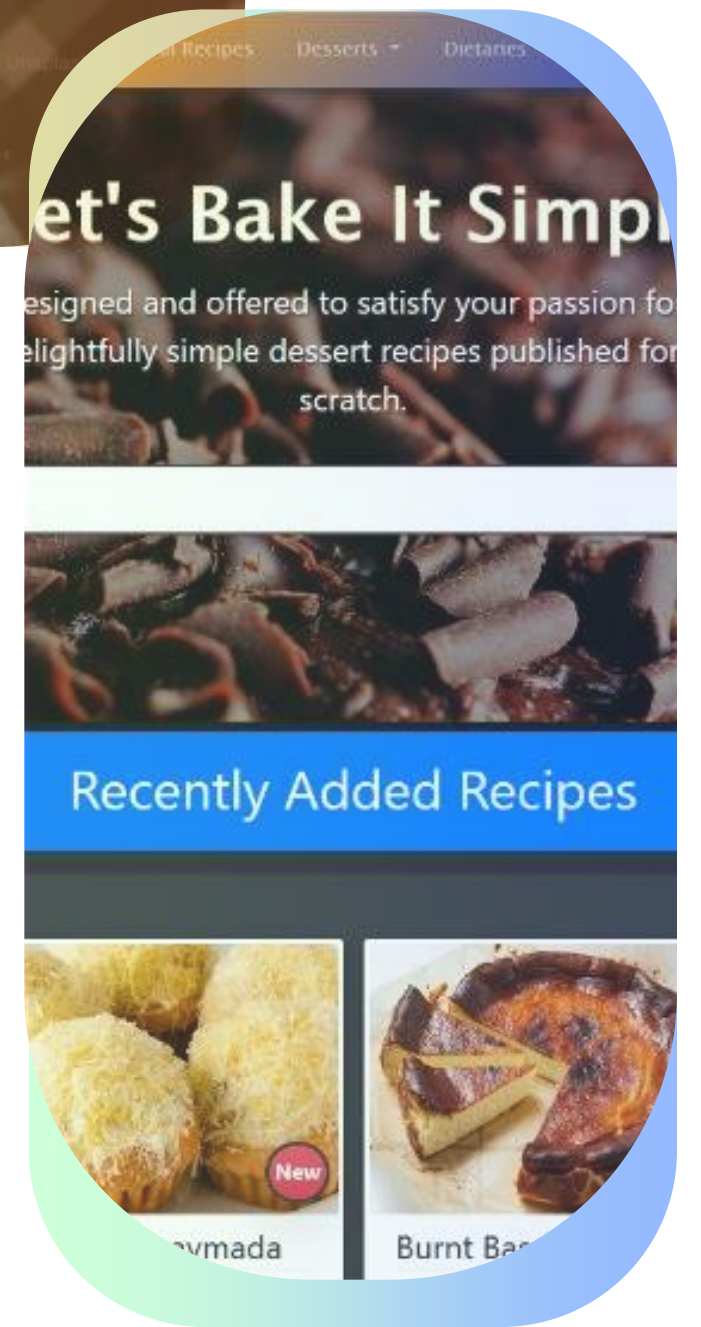
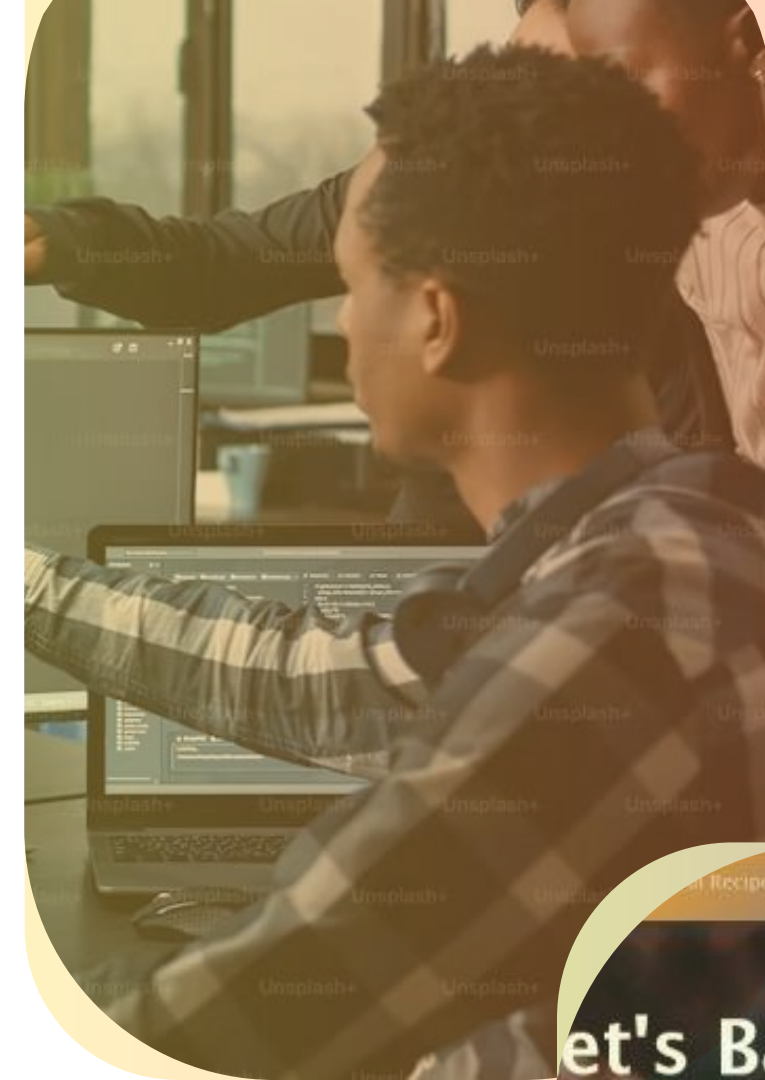


# PROJET DEVOPS

Groupe  
MakeProcess



# Presentation

**Projet** : Déploiement Multi-régional d'une Application Web Recipe App

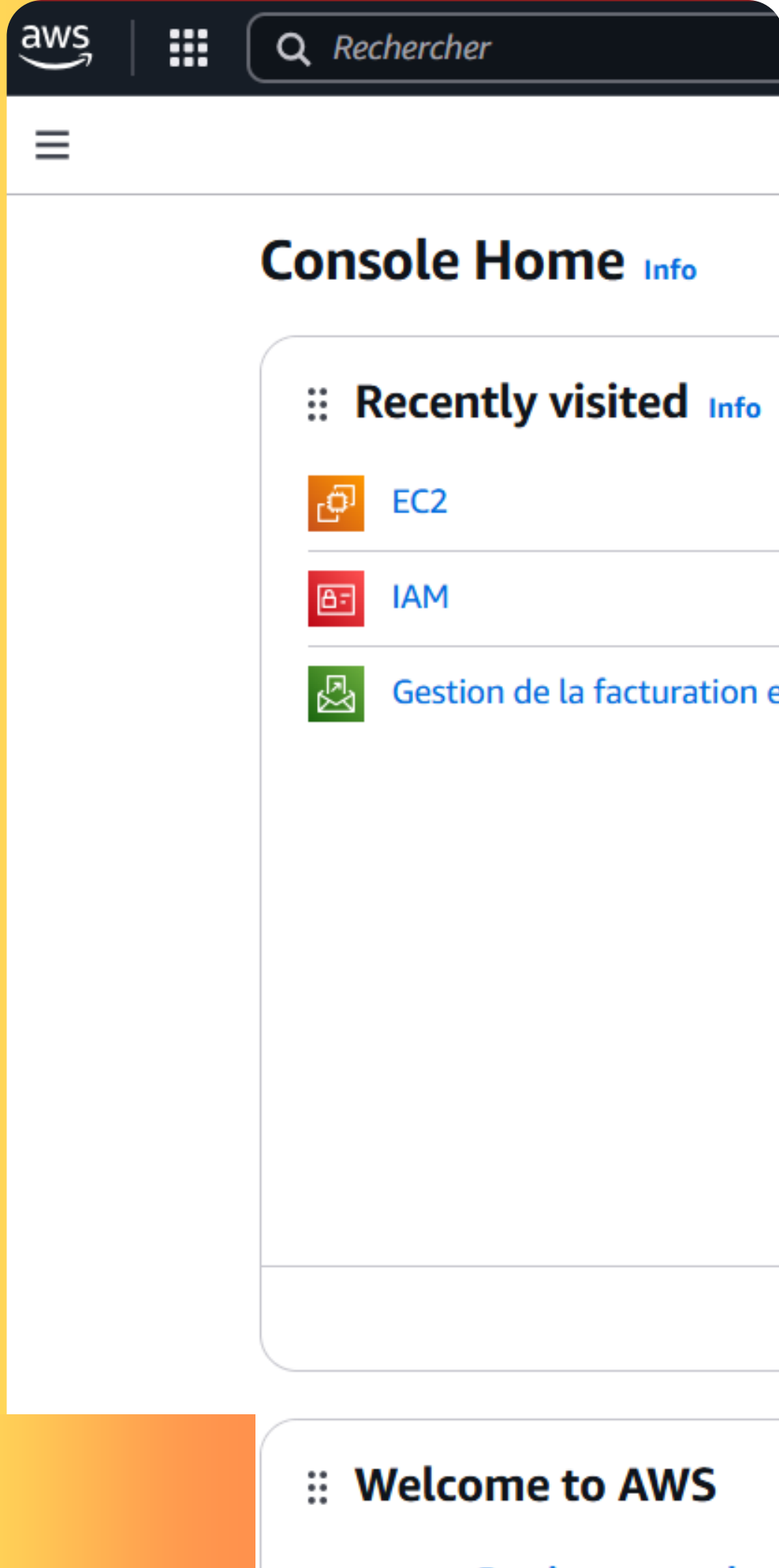
Ce projet a été déployer par 2 étudiants :

- 1.Joel KEMKENG Ingénieur Devops**
- 2.Sarah LUKUNKU Cheffe de projet web**

# Nos objectifs

Travailler en groupes, pour déployer une application web Recipe App dans un contexte multi-régional, tout en respectant les meilleures pratiques DevOps.

- **Infrastructure cloud moderne**
- **Automatisation**
- **Sécurité et observabilité**
- **Documentation**



# Demmarage du projet

💡 **Choix Outils Utilisés**

:

- ✓ **Terraform** → Provisionnement de l'infrastructure AWS.
- ✓ **AWS (EC2, RDS)** → Hébergement des serveurs et de la base de données.
- ✓ **Docker** → Conteneurisation de l'application PHP/MySQL.
- ✓ **Jenkins** (Groovy) → Automatisation du CI/CD.
- ✓ **GitLab** → Gestion du code source et déclenchement du pipeline.
- ✓ **Prometheus & Grafana** → Monitoring des serveurs et des conteneurs.



# Jour 1 : Conception & Architecture

- 🎯 Nous avons défini une architecture cloud AWS multi-régionale (France & Allemagne) et déterminé les composants clés et les interactions.

## ◆ Architecture simplifiée

### 1. Infrastructure AWS :

- **2 Régions AWS** (France, Allemagne).
- **2 Instances EC2** (backend PHP avec Apache).
- **1 Load Balancer** pour gérer la répartition du trafic.
- **1 Base de données MySQL (RDS)** avec réplication multi-régionale.
- **Stockage S3** pour les fichiers statiques.

### 2. Automatisation & CI/CD :

- **GitLab → Jenkins → Docker → AWS EC2.**
- **Terraform** pour provisionner toute l'infrastructure.

### 3. Monitoring & Sécurité :

- **Prometheus & Grafana** pour collecter les métriques.
- **Sécurisation des accès** avec IAM, groupes de sécurité AWS.

# Jour 2 : Infrastructure avec Terraform

 Nous avons déployé l'infrastructure AWS avec Terraform.

## Actions :

### 1.Écrire un script Terraform pour :

- Créer les instances EC2 dans deux régions.
- Déployer un RDS MySQL avec réplication multi-régionale.
- Configurer un load balancer.

### 2.Exécuter Terraform :

- terraform init
- terraform apply

# **Jour 3 : Automatisation avec Docker, Jenkins & GitLab**

 Automatiser le déploiement avec Jenkins + GitLab + Docker.

## **Actions :**

### **1. Dockeriser l'application PHP/MySQL :**

- Créer un Dockerfile pour l'app PHP.
- Créer un docker-compose.yml pour lancer PHP + MySQL.

### **2. Configurer Jenkins (Groovy) pour CI/CD :**

- Build du projet après chaque commit sur GitLab.
- Docker Build + Push vers AWS ECR.
- Déploiement automatique sur EC2.



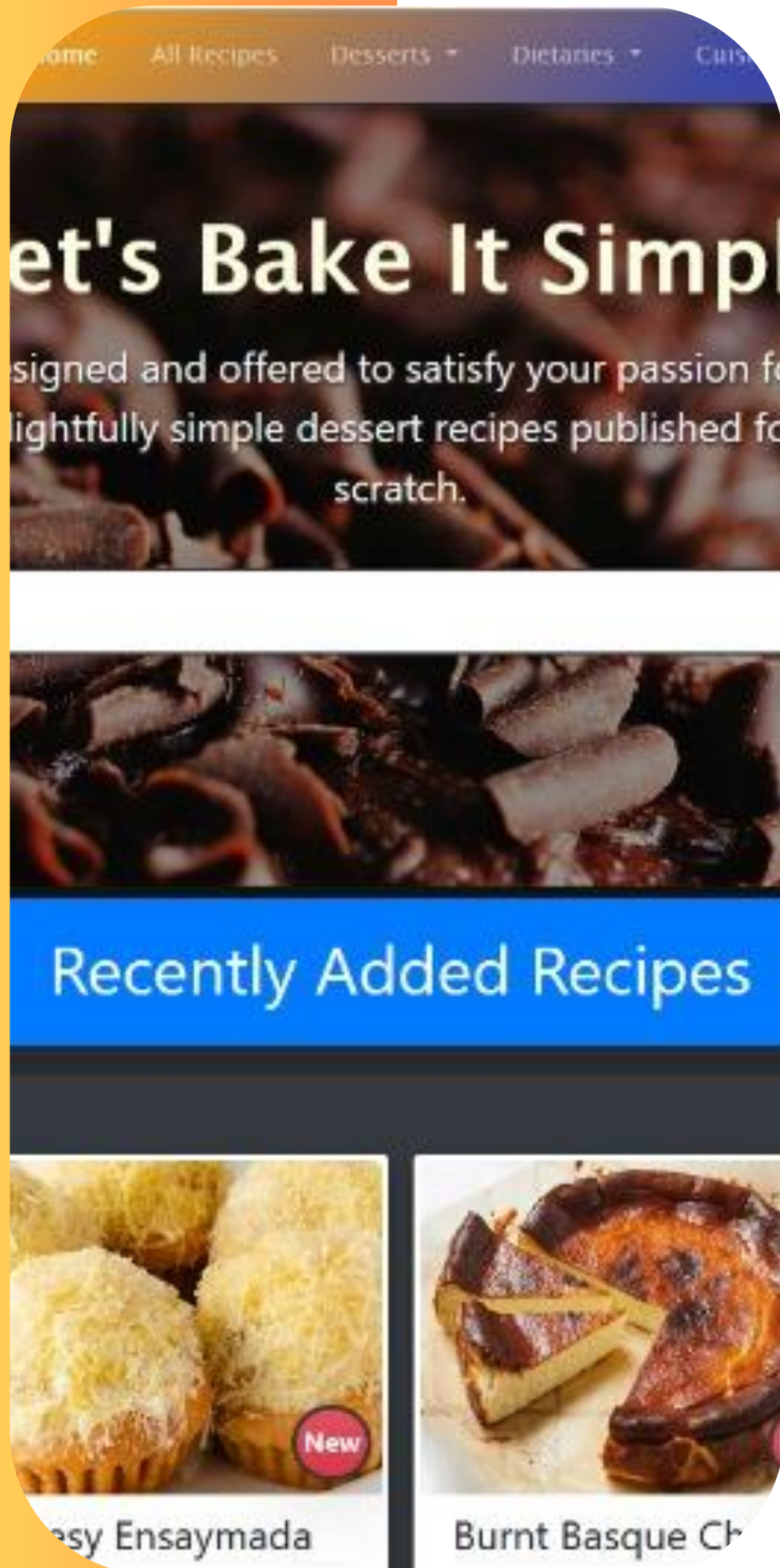
# **Jour 5 : Finalisation & Tests**

 Tester l'ensemble du système et préparer la présentation.

## **Actions :**

- 1. Tester le pipeline CI/CD (Jenkins + Docker + GitLab).**
- 2. Tester la haute disponibilité (failover entre régions AWS).**
- 3. Finaliser la documentation (architecture, configuration Terraform, CI/CD).**
- 4. Démonstration finale avec Grafana et le pipeline en action.**





# Livrables finaux

- ✓ Infrastructure AWS fonctionnelle (EC2, RDS, Load Balancer).
- ✓ Pipeline CI/CD opérationnel avec Jenkins & GitLab.
- ✓ Application conteneurisée avec Docker.
- ✓ Monitoring actif avec Prometheus & Grafana.
- ✓ Documentation et présentation finale.

**Merci !**

Groupe  
**MakeProcess**

