

# Présentation du projet

- Application WEB
  - Frontend
  - Backend
  - Microservice (exécution du code)
- Application mobile (Android)

# Fonctionnalités réalisées (Principale)

- ✓ Authentification
- ✓ Relation Followers / Followings
- ✓ Gestion des groupes
- ✓ Partage de contenu
- ✓ Likes
- ✓ Commentaires
- ✓ Gestion des versions
- ✓ Exécution de code sécurisé isolé
- ✓ Scalabilité verticale
- ✓ 4 langages
- ✓ Pipelines

# Fonctionnalités réalisées (Bonus)

- ✓ Commentaire sur sous-partie de code
- ✓ Modification à plusieurs en temps réel
- ✓ Gestion de visibilité des programmes (privé / follower / public)
- ✓ Groupe public / privé

# Technologies

*Frontend*

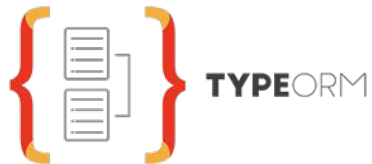


# Technologies

*backend*



**NestJS**



PostgreSQL

# Technologies

*code runner service*

# django



C E L E R Y



RabbitMQ



docker

# Proposition de Déploiement

- 3 instance de AWS EC2 :
  - frontend
  - backend
  - code runner service
- AWS Bucket S3 : sauvegarde des fichiers
- AWS RDS : base de données PostgreSQL backend et code runner service
- AWS AmazonMQ : Messaging broker
- AWS elastic API
- NameCheap : DNS / SSL / HTTPS
- certbot

# Architecture

- Architecture Monolithique modulaire pour le frontend et l'API principale du projet
  - Facilité de déploiement
  - Modulaire et facile à maintenir
  - Adaptée pour les petits/moyens projets
- L'utilisation d'un micro-service pour l'exécution de code
  - permettre d'isoler l'exécution du code de l'API principale de l'application
  - Renforce la sécurité
  - Facilite la scalabilité verticale et horizontale
- L'utilisation de Docker pour l'environnement d'exécution de code
  - Renforce encore la sécurité
  - Permettre de lutter contre les programmes malveillants
- Celery app et Messaging broker pour gérer l'exécution de plusieurs tâches en même temps



