

# PROJET VIRTUALISATION

**2024/2025**

## **Déploiement d'une Application IoT sur Cloud Open Source et AWS**

## **Ingénierie Informatique et Réseaux**

Réalisé par :

Nom de l'étudiant 1 : ELMOUHTADI Feirouz

Nom de l'étudiant 2 : KELLADI FatimaEzzahra

Nom de l'étudiant 3 : OUHMIDA Soulaimane

Nom de l'étudiant 4 : ETTAQUADOUUMI Hamza

# Déploiement d'une Application IoT sur Cloud Open Source et AWS

## Context

Le projet consiste à concevoir et mettre en œuvre un système de recyclage de l'eau dans les villes et industries, en utilisant des technologies avancées pour améliorer l'efficacité de la gestion de l'eau. L'objectif est de réduire la consommation d'eau potable, d'optimiser les cycles de recyclage, et de garantir la qualité de l'eau recyclée pour des usages non essentiels comme l'irrigation ou le nettoyage industriel.

Le système intégré repose sur l'utilisation de capteurs IoT pour surveiller en temps réel la qualité de l'eau, de technologies de filtration avancées comme l'osmose inverse, et de solutions d'intelligence artificielle pour automatiser et optimiser le processus de recyclage.

## Choix techniques

### Frontend - React.js



- Bibliothèque JavaScript pour créer des interfaces utilisateur dynamiques et responsives.
- Utilisé pour l'affichage des données en temps réel (capteurs IoT) et les recommandations issues du modèle ML.
- Déployé sur **AWS S3** avec **AWS CloudFront** pour la distribution.

### Backend API - Node.js + Express.js



- Environnement JavaScript côté serveur pour créer des API REST performantes.
- Express facilite la gestion des routes, middlewares, et communication avec les services IoT et base de données.
- Déployé sur instances EC2 avec Elastic Load Balancer pour la montée en charge.

## Génération de Données IoT - Python



- Génère des données simulées de qualité de l'eau (pH, turbidité, température, etc.).
- Ces données sont envoyées périodiquement à AWS IoT Core via MQTT.
- Permet de simuler un environnement réel pour le test et l'entraînement des modèles.

## Machine Learning – Python



- Analyse des données collectées (ex : classification eau propre/eau polluée).
- Génère des recommandations : recyclage, filtrage, ou alerte utilisateur.
- Peut être exécuté via AWS Lambda ou EC2 selon le besoin en puissance.

## Base de Données – MongoDB



- Stocke les mesures IoT et les résultats des analyses ML.
- Adaptée pour des données semi-structurées de type JSON (MongoDB).
- Accessible à la fois par le backend Node.js et les fonctions Lambda ML.

## Services AWS Utilisés

Dans le cadre du déploiement de notre application IoT sur le cloud AWS, plusieurs services AWS ont été utilisés afin de garantir scalabilité, haute disponibilité, sécurité et facilité de maintenance.

### AWS IoT Core



- **Rôle :** Plateforme centrale de gestion et de communication des objets connectés.
- **Utilisation :**
  - Connexion des capteurs simulés (via MQTT).
  - Traitement des messages entrants.
  - Déclenchement des fonctions AWS Lambda pour analyse.
- **Avantages :**
  - Gestion sécurisée de millions d'appareils.
  - Intégration directe avec d'autres services AWS (Lambda, DynamoDB, S3...).

### AWS Lambda



- **Rôle :** Exécution de fonctions serverless pour l'analyse des données IoT.
- **Utilisation :**
  - Traitement automatique des données dès réception via IoT Core.
  - Exécution de modèles de Machine Learning (Python).
  - Insertion des résultats dans la base DocumentDB.
- **Avantages :**
  - Aucun serveur à gérer.
  - Paiement à l'exécution (économie de coûts).
  - Haute scalabilité.

## Amazon S3



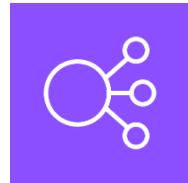
- **Rôle :** Stockage statique du frontend React.
- **Utilisation :**
  - Hébergement du site web (interface utilisateur).
- **Avantages :**
  - Haute durabilité et disponibilité.
  - Coût très faible.
  - Intégration avec CloudFront pour un accès rapide.

## Amazon CloudFront



- **Rôle :** Réseau de distribution de contenu (CDN).
- **Utilisation :**
  - Accélération du chargement du frontend React.
  - Mise en cache globale pour les utilisateurs finaux.
- **Avantages :**
  - Réduction de la latence.
  - Sécurité (via HTTPS).

## Elastic Load Balancer (ELB)



- **Rôle :** Répartition du trafic entre plusieurs instances backend.
- **Utilisation :**
  - Distribution équitable des requêtes HTTP/HTTPS vers les serveurs Node.js et python.
- **Avantages :**
  - Haute disponibilité.
  - Redondance en cas de panne d'une instance EC2.

## Amazon EC2



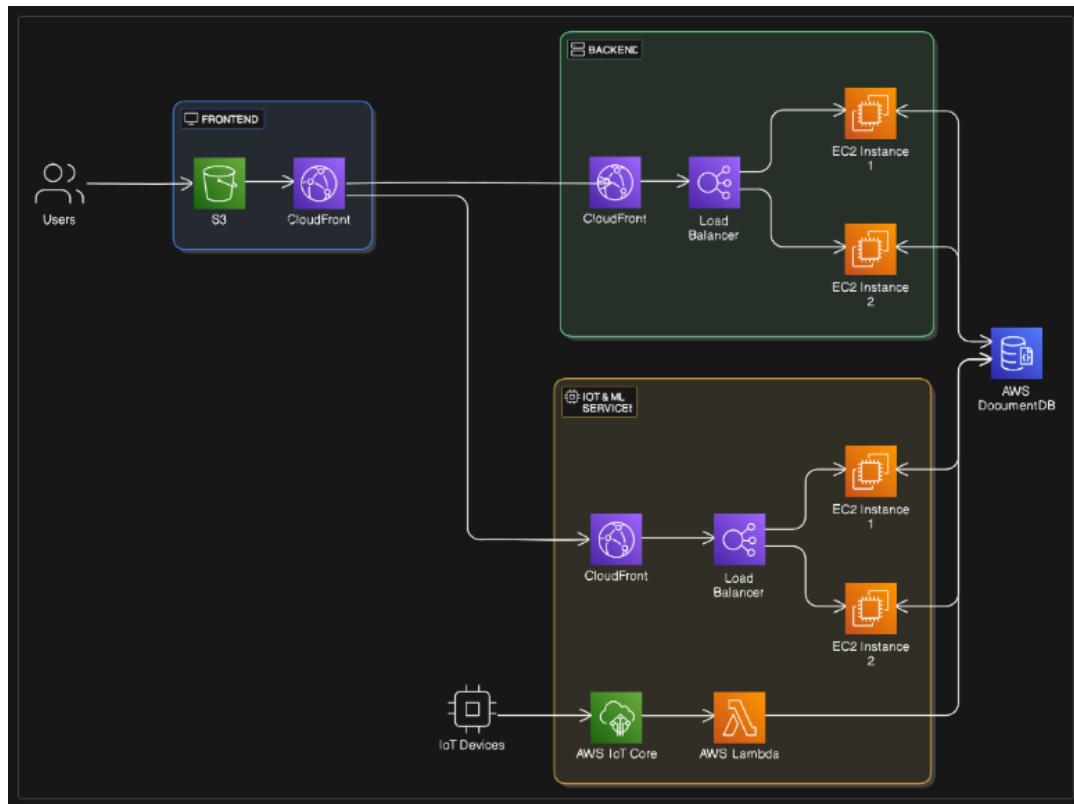
- **Rôle :** Hébergement du backend Node.js (Express) et des services IoT/ML lourds.
- **Utilisation :**
  - API REST pour les appels frontend.
  - Traitement des données utilisateur et IoT.
- **Avantages :**
  - Contrôle total sur l'environnement.
  - Évolutivité (scale up/down automatique avec Auto Scaling).

## Amazon DocumentDB (compatible MongoDB)



- **Rôle :** Base de données NoSQL pour stocker les données des capteurs et les analyses ML.
- **Utilisation :**
  - Stockage des mesures IoT.
  - Historique des analyses et recommandations.
- **Avantages :**
  - Compatible MongoDB (facile à utiliser avec Mongoose ou PyMongo).
  - Haute disponibilité et sauvegarde automatique.

## Diagrammes d'architecture



## Captures d'écran du projet

- Serveurs EC2 (4)
  - 2 Backends (Node js)
  - 2 Machine learning (Python)

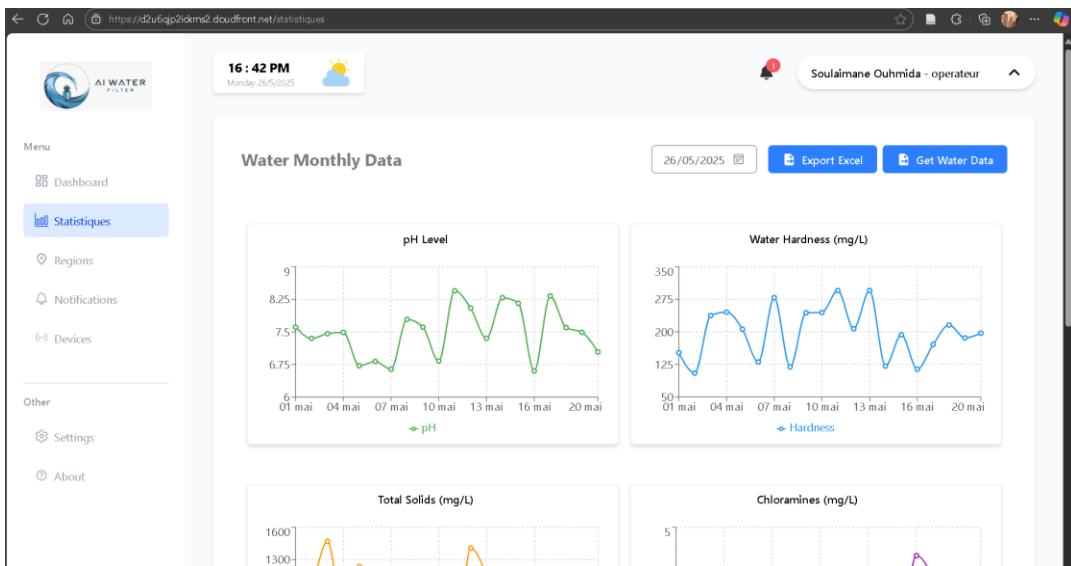
Instances (8) <a href="#">Info</a>								
		Last updated <a href="#">C</a> less than a minute ago	<a href="#">Connect</a>	<a href="#">Instance state</a> ▾	<a href="#">Actions</a> ▾	<a href="#">Launch instances</a>	▼	
<input type="checkbox"/>	Name <a href="#">P</a>	Instance ID	Instance state <a href="#">▼</a>	Instance type <a href="#">▼</a>	Status check	Alarm status	Availability Zone <a href="#">▼</a>	Public IPv
<input type="checkbox"/>	test-server	i-076526a35b3cd0218	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1c	-
<input type="checkbox"/>	iot1-backend	i-0c966162680a4d39e	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1c	-
<input type="checkbox"/>	iot2-backend	i-0d9028dbd42ec953c	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1c	-
<input type="checkbox"/>	mliot1	i-0d3cec3bc9424c576	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1c	-
<input type="checkbox"/>	mliot2	i-0c6fee7df2bf75ea0	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1c	-
<input type="checkbox"/>	onid-1	i-0b8f8ff9771dbc5fe	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1d	-
<input type="checkbox"/>	onid-2	i-05cad5c5c7db85f4d58	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1d	-
<input type="checkbox"/>	devops-backend	i-0b02c0fb753f2df83	<a href="#">Stopped</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	-	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1c	-

- Load balancers (2)
  - 1 pour Backends (Node js)
  - 1 pour Machine learning (Python)

Load balancers (3)							
Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.							
<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date creat
<input type="checkbox"/>	onid	onid-100673152.us-east...	<a href="#">Active</a>	vpc-0852922c64078b...	6 Availability Zones	application	May 11, 2023
<input type="checkbox"/>	iotbackend	iotbackend-947648760.us...	<a href="#">Active</a>	vpc-0852922c64078b...	6 Availability Zones	application	May 24, 2023
<input type="checkbox"/>	ml-iot	ml-iot-114803879.us-east...	<a href="#">Active</a>	vpc-0852922c64078b...	6 Availability Zones	application	May 25, 2023

- CloudFront (3)
  - 1 pour Load balancer (Node js)
  - 1 pour load balancer (Python)
  - 1 pour S3 (Frontend)

Distributions (6) <a href="#">Info</a>								
<a href="#">Create distribution</a>								
<input type="checkbox"/>	ID	Status	Description	Type	Domain name	Alternate ...	Origins	Last mo...
<input type="checkbox"/>	E1RP0OPB0XPR76	<a href="#">Enabled</a>	water-iot-front	Standard	<a href="#">d2u6qjp...</a>	-	water-iot-front.s3	May 25, 2023
<input type="checkbox"/>	E28GMP3MC2M4YR	<a href="#">Enabled</a>	-	Standard	<a href="#">d2x0oh...</a>	-	ml-iot-11480387...	May 25, 2023
<input type="checkbox"/>	E161SQDJ9JXXBN	<a href="#">Enabled</a>	-	Standard	<a href="#">docimd1...</a>	-	iotbackend-9476...	May 25, 2023
<input type="checkbox"/>	EAKQRQKZJHX2A	<a href="#">Enabled</a>	Backend	Standard	<a href="#">d1zd6zg...</a>	-	onid-100673152.j...	May 21, 2023
<input type="checkbox"/>	EBGNPSF3LPEN	<a href="#">Enabled</a>	Frontend	Standard	<a href="#">d1ory24...</a>	-	onid.s3-website-u...	May 21, 2023
<input type="checkbox"/>	E1OQ0068MHL9DG	<a href="#">Disabled</a>	-	Standard	<a href="#">d1n5ryj...</a>	-	www.moviey-front	April 10, 2023



## Liens

<b>EC2 (Node js) – serveur 1</b>	Adresse IP dynamique
<b>EC2 (Node js) – serveur 2</b>	Adresse IP dynamique
<b>EC2 (Python) – serveur 1</b>	Adresse IP dynamique
<b>EC2 (Python) – serveur 2</b>	Adresse IP dynamique
<b>Load balancer – Nodejs</b>	iotbackend-947648760.us-east-1.elb.amazonaws.com
<b>Load balancer – Python</b>	ml-iot-114803879.us-east-1.elb.amazonaws.com
<b>CLoudfront - Nodejs</b>	<a href="https://docimd1ylkn3z.cloudfront.net">https://docimd1ylkn3z.cloudfront.net</a>
<b>CLoudfront - Python</b>	<a href="https://d2x0ohgw8qmfgx.cloudfront.net">https://d2x0ohgw8qmfgx.cloudfront.net</a>
<b>S3 - Frontend</b>	<a href="http://water-iot-front.s3-website-us-east-1.amazonaws.com">http://water-iot-front.s3-website-us-east-1.amazonaws.com</a>
<b>CLoudfront - Frontend</b>	<a href="https://d2u6qjp2ickms2.cloudfront.net">https://d2u6qjp2ickms2.cloudfront.net</a>
<b>Github</b>	<a href="https://github.com/Soulaimane07/PFA-2025.git">https://github.com/Soulaimane07/PFA-2025.git</a>