# **PrepIQ Deployment Report**

## **1. Overview**

This document outlines the deployment architecture and process used for the EduRAG platform, consisting of a FastAPI backend and a React frontend, hosted on AWS using ECS (for backend) and EC2 (for frontend). The system is secured, scalable, and integrated with Route53 for custom domain routing.

## **2. Architecture Diagram**

+----------------------+ +-----------------------+

| Route53 Domain | | AWS ECR (Images) |

| prepiq.online | | Backend + Frontend |

+----------+-----------+ +-----------------------+

|

v

+----------------------+ HTTPS/HTTP +--------------------------+

| Application Load | <----------------> | ECS Fargate (Backend) |

| Balancer (ALB) | | FastAPI + S3 Access |

+----------+-----------+ +------------+-------------+

| |

| v

+------------------+ +--------------------------+

| EC2 (Frontend) | <--------------------- | S3 Bucket (cmpe-295...) |

| React + Nginx | API Calls +--------------------------+

+------------------+

## **3. Backend (ECS Fargate)**

### **Components:**

* **FastAPI app**
* Hosted on **ECS Fargate** behind an **Application Load Balancer (ALB)**
* Integrated with:  
  + **S3** (for FAISS index storage)
  + **RabbitMQ** (for task queue)

### **Key CloudFormation Resources:**

* AWS::ECS::Cluster
* AWS::ECS::Service
* AWS::ElasticLoadBalancingV2::TargetGroup
* AWS::IAM::Role with S3 full access

### **Health Check:**

* ALB health check at /api/

## **4. Frontend (React + EC2)**

### **Components:**

* React app built with REACT\_APP\_API\_URL pointing to backend ALB URL
* Dockerized and pushed to **ECR**
* Deployed on an **EC2 instance** using a CloudFormation stack
* Nginx used to serve the frontend on port 80

### **CloudFormation Setup:**

* EC2 instance with Docker
* UserData script pulls from ECR and runs the image
* Route53 A record points to EC2 public IP

### **Notes:**

* The EC2 instance is in the **default VPC with public subnet**
* Security group allows HTTP (port 80)

## **5. Route53 + Domain Setup**

* Domain: prepiq.online
* Registered and hosted via Route53
* A record created for root domain pointing to **EC2 public IP** (frontend)
* Optionally, subdomains can be pointed to ALB or other services

## **6. CI/CD via GitHub Actions**

### **Workflow Highlights:**

* Backend and frontend are built and pushed to **ECR**
* Backend: triggers **CloudFormation deploy** to update ECS Service
* Frontend: builds using backend ALB URL, then triggers EC2 stack

## **7. Observability & Debugging**

* Logs from ECS tasks can be monitored in **CloudWatch Logs**
* EC2 logs available via user-data.log
* ALB health status shows service health per target (backend/EC2)

## **8. Security**

* IAM Roles restrict access per service
* S3 access granted only to backend ECS tasks
* Public internet access only exposed via ALB (backend) and Nginx (frontend)

## **9. Future Improvements**

* Add HTTPS using ACM and ALB
* Replace EC2 frontend with ECS Fargate or S3 Static Hosting
* Add auto-scaling for backend
* Use Route53 alias record to point to ALB instead of EC2 IP

## **10. Summary**

The deployment is cloud-native, modular, and cost-efficient. It ensures isolated environments for backend processing and frontend rendering while integrating seamlessly through environment configurations and scalable AWS infrastructure.