Strapi Deployment on AWS ECS Fargate: Project Report

---

1. Introduction

This project aims to deploy a Strapi Content Management System (CMS) application using Amazon Web Services (AWS) Elastic Container Service (ECS) with Fargate launch type. The project follows Infrastructure as Code (IaC) practices using Terraform, allowing automated, scalable, and serverless deployment of the Strapi app. The goal is to containerize the application, push it to a Docker registry, and run it using AWS-managed services.

---

2. Abstract

Strapi is an open-source, headless CMS that enables easy content management and API generation. ECS Fargate eliminates the need to manage servers by allowing container deployment directly. This project involves creating a Docker image of Strapi, pushing it to Docker Hub, defining the infrastructure using Terraform (including ECS Cluster, Task Definition, ALB, Target Groups, and IAM Roles), and deploying it to AWS. A PostgreSQL database is used as the backend. The result is a scalable, fully managed CMS environment.

---

3. Tools Used

- **Strapi** – CMS Application
- **Docker** – Containerization
- **Docker Hub** – Container Registry
- **Terraform** – Infrastructure as Code
- **AWS ECS Fargate** – Serverless Container Deployment
- **AWS ALB** – Load Balancer
- **PostgreSQL** – Database for Strapi
- **GitHub Actions** – CI/CD Automation (optional)

---

4. Steps Involved in Building the Project

1. **Strapi App Setup:** Create a new Strapi app locally using Yarn or NPX.
2. **Dockerization:** Write a `Dockerfile` to containerize the app and push the image to Docker Hub.
3. **Terraform Infrastructure:** Define AWS components:
4. VPC, Subnets, Security Groups
5. ECS Cluster & Fargate Task
6. ALB, Target Group, Listeners
7. IAM Roles & Policies
8. **Database Setup:** Deploy PostgreSQL on RDS or use an external service.
9. **Environment Variables:** Configure sensitive variables like DB credentials via ECS Task Definition.
10. **Terraform Apply:** Use `terraform init`, `plan`, and `apply` to deploy the full stack.
11. **Validation:** Access Strapi via the ALB DNS, create collection types, and verify database persistence.

---

5. Conclusion

Deploying Strapi on AWS ECS Fargate using Terraform allows for a scalable and serverless CMS solution. This project demonstrates key DevOps practices including containerization, IaC, and cloud-native deployment. The modular infrastructure can be reused and extended for similar web applications, reducing deployment time and operational overhead.

End of Report