

SK네트웍스 Family AI과정 3기

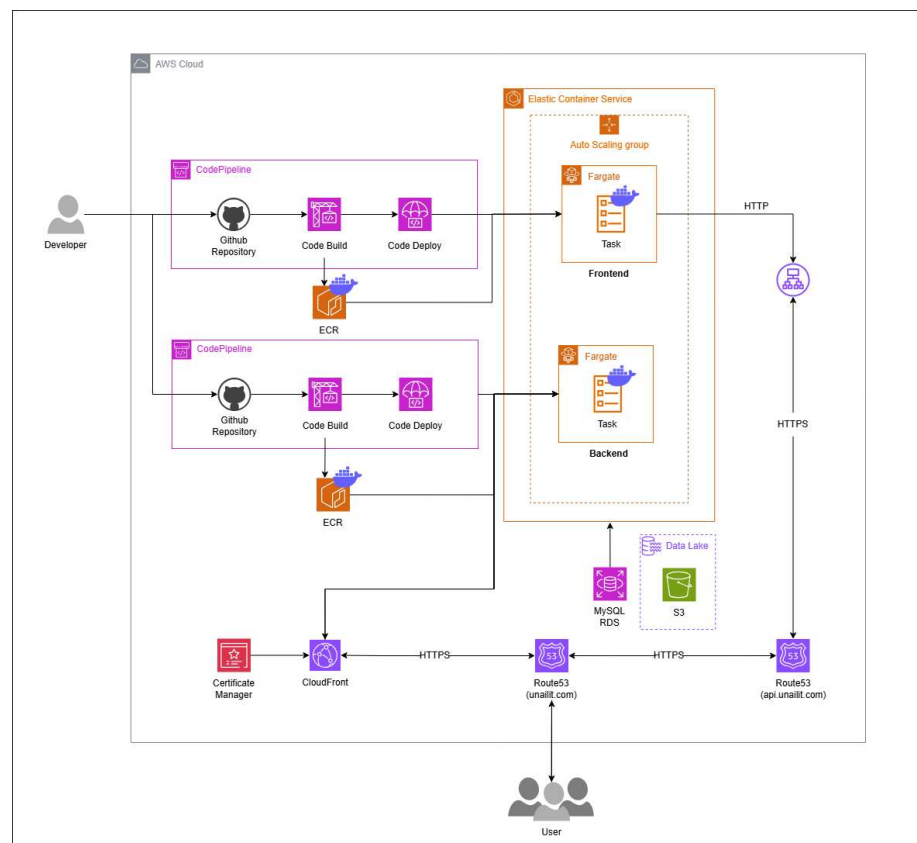
모델배포 시스템 구성도

□ 개요

- 산출물 단계 : 모델배포
- 평가 산출물 : 시스템 구성도
- 제출 일자 : 12/28
- 깃허브 경로 : <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN03-FINAL-5Team>
- 작성 팀원 : 김중식, 김원철, 유혜린

개요

본 프로젝트에서는 AWS를 활용하여 CI/CD를 자동화하고, 서버리스 컨테이너 환경에서 효율적으로 애플리케이션을 배포 및 실행하는 것을 목표로 한다. 트래픽 부하에 따라 Auto Scaling을 통해 자원을 동적으로 할당하며, RDS와 S3를 통해 데이터를 안전하게 관리한다. 또한 CloudFront와 ACM을 통해 글로벌 콘텐츠 전송과 HTTPS 보안을 제공한다.



구성 요소

컴퓨팅 및 배포

- GitHub: 애플리케이션 소스 코드 저장 및 관리.
- CodePipeline: GitHub에서 변경 사항을 감지하고 자동 빌드 및 배포를 수행.
- CodeBuild: 코드를 컴파일하고, 의존성을 설치하여 실행 가능한 빌드 파일 생성.
- ECR (Elastic Container Registry): 빌드된 Docker 이미지를 저장.
- Fargate: ECS에서 서버리스 방식으로 컨테이너 실행.
- ECS Tasks: 컨테이너 기반의 애플리케이션 실행 단위.
- ALB (Application Load Balancer):
 - 트래픽을 백엔드 및 프론트엔드 ECS Tasks로 분산.
 - HTTPS 요청을 처리하며, 인증서를 통해 보안을 강화.
 - Route 53과 연결되어 사용자 요청을 적절한 서비스로 라우팅.
- Auto Scaling: 트래픽 부하에 따라 ECS 인스턴스를 동적으로 조정.

데이터 관리

- RDS (Relational Database Service): 관계형 데이터베이스로 애플리케이션 데이터를 저장.
- S3 (Simple Storage Service): 대용량 데이터 및 백업을 저장.
- Data Lake: S3 기반으로 데이터를 통합 관리 및 분석.

네트워크 및 보안

- Route 53: 도메인 네임 관리 및 HTTPS 요청 라우팅.
- CloudFront: 글로벌 콘텐츠 전송을 위한 CDN 서비스.
- ACM (AWS Certificate Manager): HTTPS 연결 보장을 위한 SSL/TLS 인증서 관리.

데이터 흐름

데이터 흐름

- 코드 푸시 및 CI/CD 프로세스
 - 개발자가 GitHub에 코드를 푸시하면 CodePipeline이 변경 사항을 감지.
 - CodeBuild가 실행되어 코드를 컴파일하고 빌드.
 - ECR을 통해 도커이미지를 저장한 후 ECS를 통해 컨테이너를 생성하고 실행
- 애플리케이션 배포 및 실행
 - ECS 컨테이너는 Task를 생성하고, ALB의 규칙에 따라 배포
 - Fargate는 ECS Tasks를 통해 컨테이너를 서버리스로 실행.
 - 트래픽 부하에 따라 Auto Scaling이 인스턴스를 동적으로 확장/축소.
- 데이터 저장 및 관리
 - 생성된 데이터는 RDS를 통해 관계형 데이터베이스로 저장.
 - 파일 업로드 및 정적 데이터는 S3에 저장.
 - MySQL(RDS)와 S3를 통해 필요한 데이터를 효율적으로 관리.
- 사용자 요청 처리
 - 사용자는 Route 53을 통해 애플리케이션에 접근.
 - CloudFront가 콘텐츠를 빠르게 전송하고 ACM이 HTTPS 보안을 제공.
 - 사용자 요청은 Route 53을 통해 적절한 도메인으로 연결되고, ALB가 이를 ECS Tasks로 분배