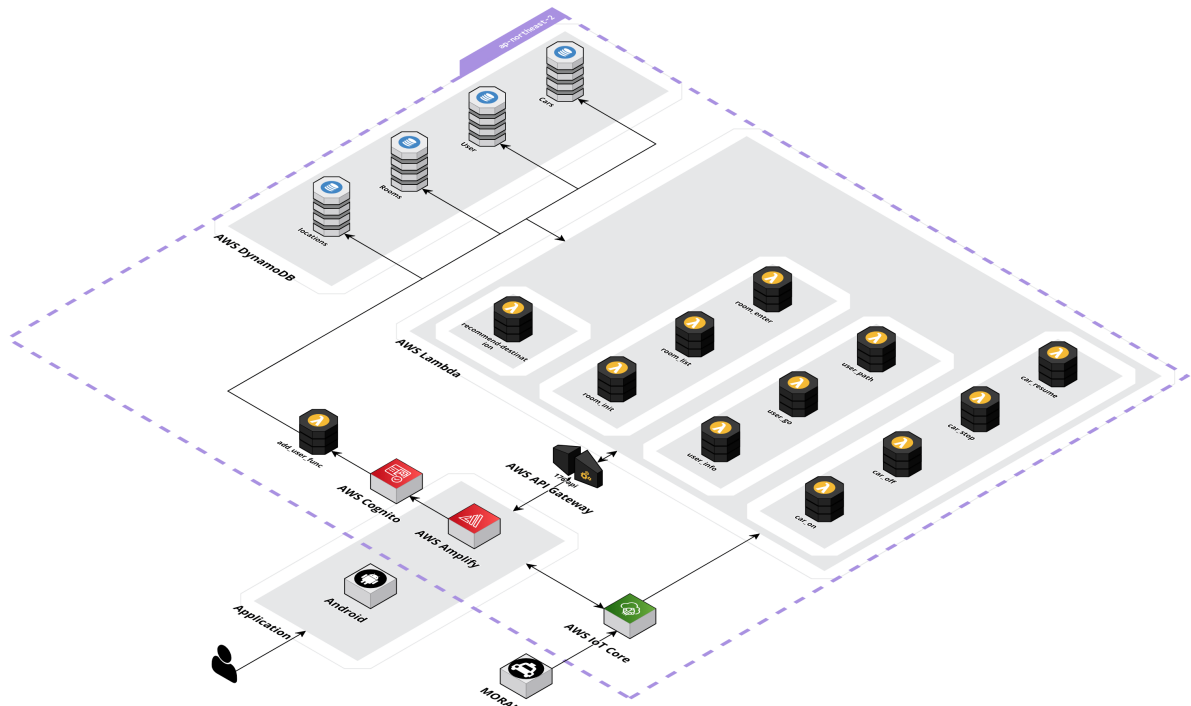


시스템 아키텍처/구성도



Amazon Cognito는 웹 및 모바일에 대한 인증, 권한 부여 및 사용자 관리를 제공

Amazon Cognito의 두 가지 주요 구성 요소는 '사용자 풀'과 '자격 증명 풀'입니다. 사용자 풀은 앱 사용자의 가입 및 로그인 옵션을 제공하는 사용자 디렉터리입니다. 자격 증명 풀을 통해 사용자에게 기타 AWS 서비스에 액세스할 수 있는 권한을 부여할 수 있습니다.

AWS Amplify는 개발자가 웹 또는 모바일 플랫폼에서 클라우드 서비스로 애플리케이션을 구축할 수 있도록 하는 AWS에서 제공하는 JavaScript 라이브러리입니다.

AWS는 웹 및 모바일 개발자가 AWS에서 호스팅하는 확장 가능한 전체 스택 애플리케이션을 개발할 수 있도록 설계된 전체 스택 플랫폼입니다. AWS Amplify를 이용하면 사용자가 백엔드를 쉽게 구성하고, 앱을 연결하고, 정적 웹 앱을 즉시 배포하고, AWS 콘솔 외부에서 콘텐츠를 관리할 수 있게 해주는 수많은 도구와 서비스가 포함되어 있습니다.

AWS Cognito의 장점

- 애플리케이션에 대한 사용자를 인증하고 권한을 부여하는 안전한 방법을 제공하는 완전 관리형 사용자 ID 및 액세스 관리 서비스
- 강력한 보안
 - AWS의 보안 인프라를 활용하여 사용자 데이터 및 인증 정보를 안전하게 보호

- 개발 시간 절약
 - 사용자 인증 및 권한 부여와 관련된 복잡한 작업을 간소화 가능

AWS Amplify의 장점

- 다른 AWS 서비스의 추가, 확장이 간편
- 사용자 인증 간편함
- 프론트엔드와 백엔드 개발환경을 분리해서 관리가 가능하다.
- 간단하게 배포 자동화 시스템만을 구축하는 서비스가 아닌 다양한 기능이 있다.
- 애플리케이션에서 사용자 인증 및 권한 부여를 구현하는 데 사용할 수 있는 사전 빌드된 인증 모듈이 포함되어 있음
- AWS Amplify는 개발자가 애플리케이션에 사용자 인증 및 권한 부여를 쉽게 추가할 수 있도록 AWS Cognito와의 원활한 통합을 제공합니다.

AWS Cognito를 사용해 Amplify Auth의 인증 기능 구축

AWS Cognito를 이용해 User Pool을 생성 및 관리

회원 가입 시 유저를 User Pool에 추가, 사후 확인 Lambda 트리거를 이용해 유저 정보를 AWS DynamoDB에 저장

서버리스(Serverless) 장점

- 생산성
 - 작성해야 하는 코드가 줄어 빠르게 애플리케이션 완성 가능
- 확장성
 - 서버리스 아키텍처에서는 공급자가 사용량에 따라 서버 및 데이터베이스를 필요한 만큼 확장해준다.
 - 또한 FaaS를 사용하기 때문에, 함수를 실행한 만큼에 대한 비용만 지불하면 됨
- 안정성
 - 전문 엔지니어들이 최적화해놓은 서버리스 아키텍처를 사용하기 때문에, 직접 구축하는 것보다 안정적으로 애플리케이션 서비스 가능
 - 유저 접속이 갑자기 폭등하더라도 안정적인 서비스 유지 가능

AWS API Gateway

- 경로와 엔드포인트로 구성되어 정의된 HTTP 서버
- 경로마다 리소스와 연결되어 FaaS의 function으로 가는 관문 역할
 - 즉, 요청과 일치하는 라우팅 구성을 찾아서 관련되는 FaaS의 기능을 호출
 - 기능, function → AWS Lambda
- AWS API Gateway에서는 API 메서드를 통해 Lambda 함수를 제공

AWS Lambda

- 이벤트에 대한 응답으로 코드를 실행하고 자동으로 기본 컴퓨팅 리소스를 관리하는 서버리스 컴퓨팅 서비스
- 장점
 - 확장성 및 탄력성
 - AWS Lambda는 자동으로 확장하므로, 트래픽이 증가하더라도 안정적으로 서비스 제공 가능
 - 비용 절감
 - 사용하지 않으면 비용도 부과되지 않으며, 실행 시간에 대한 비용만 지불하면 됨
 - 다양한 통합 기능
 - AWS Lambda는 다양한 AWS 서비스와 통합되어 있어, 다른 AWS 서비스를 사용하여 Lambda 함수를 확장 가능

AWS DynamoDB

- NoSQL 데이터베이스
- HTTP로 통신
- 서버리스
 - AWS Lambda와 같은 다른 서버리스 기반 서비스와 시너지
- Key-Value 데이터베이스
 - Key를 제외한 테이블의 속성을 미리 정의해둘 필요 없음

애플리케이션에서 AWS API Gateway로 요청을 보내면, 그 요청에 맞는 Lambda 함수 호출

Lambda 함수에서는 DynamoDB의 값을 읽거나 쓰면서 유저의 요청을 처리

AWS IoT Core

- IoT 디바이스 간의 연결 및 AWS 클라우드 서비스와의 연동기능 제공

Application과 Application 그리고 Application과 MORAI(차량)은 MQTT 프로토콜을 이용해 통신

AWS IoT Core는 브로커 역할을 수행하며 게시(publish)된 메시지에 따라 AWS Lambda를 이용해 데이터를 처리하기도 함