

* 한동어플 서비스의 시스템 구조입니다. 간략한 플로우는 아래와 같습니다.  
  먼저 사용자는 Flutter로 제작된 한동어플을 이용합니다. 사용자가 발생시키는 이벤트들은 Firebase Analytics에 기록, 분석됩니다.  
  사용자가 어플리케이션의 기능을 이용하게 되면, hguapp.ghoster.cc 주소로 데이터를 요청하게 됩니다. 이때 요청은 443번 포트를 통해 ELB에게 전달됩니다. ELB에서는 EC2 인스턴스의 80번 포트로 요청을 전송합니다. EC2 인스턴스의 80번 포트에서 작동하는 NGINX가 요청을 받으면 url에 따라 각기 다른 프로세스로 요청을 전달합니다.  
  대부분의 요청은 NodeJS 어플리케이션에 전달되며 NodeJS에서는 요청에 맞는 데이터를 DB로부터 불러와 다시 클라이언트로 전송합니다.  
  서비스의 관리자는 관리자페이지를 통하여 데이터를 관리합니다.  
  아래는 사용한 기술 스택에 관한 설명입니다.
* Front-end
  + Flutter: Google에서 개발한 크로스플랫폼 프레임워크 입니다. 기존 한동어플에서 사용하던 Ionic 혹은 ReactNative와 같은 자바스크립트 기반의 크로스플랫폼 보다 성능적인 면에서 우월한 것이 대표적인 장점입니다. 저희는 Flutter를 사용하여 사용자 어플리케이션을 개발하였기에, 기존의 iOS와 Android 사용자 모두에게 동일한 서비스를 제공할 수 있습니다.
  + Angular: 역시 Google 사에서 개발한 웹(SPA) 프레임워크입니다. 기존 서비스에서는 DB 수정을 위해 직접 DB에 접근하여 작업을 했습니다. 그러나 이러한 행위는 보안적인 측면, 그리고 유지보수 측면에서도 문제가 있습니다. 따라서 중간자 역할을 하는 관리자 웹 페이지를 개발하여 위 문제점들을 해결함과 동시에, DB에 대한 지식이 없는 사용자도 서비스 관리를 할 수 있게끔 만들었습니다.
* Back-end
  + Firebase
    - Analytics: Google Firebase의 Analytics를 이용하여 사용자의 이벤트를 분석합니다. Google 사에서 기존에 제공하던 GA와도 유사하지만, Flutter 특성상 Firebase와의 연동이 간단하여 Analytics를 어플리케이션에 등록해놨습니다. 앞으로 사용자의 이벤트를 분석하여 서비스 개선을 위해 활용할 생각입니다.
  + AWS
    - route53, ELB: 각각 Amazon AWS의 도메인 관리 서비스, Load Balancer 입니다. 두 서비스를 같이 이용한 가장 큰 이유는 SSL 인증서 자동갱신을 위함입니다. 일반적으로 SSL 인증서는 1년에 한번 갱신을 시켜줘야하는데, AWS의 서비스를 이용하면 자동갱신이 가능합니다. 또한 ELB를 통해 트래픽을 EC2 인스턴스로 전송함으로써 EC2 인스턴스의 주소를 숨긴 것과 ELB의 80번 포트로 들어오는 요청들을 433번 포트로 재전송 시킴으로써 보안성을 향상하였습니다. 마지막으로 추후 서비스 확장시에도 사용중인 ELB를 이용하여 로드밸런싱이 용이할 것으로 예상됩니다.
    - EC2: NodeJS, MySQL 등이 동작하는 클라우드 컴퓨터로는 EC2의 리눅스 인스턴스를 생성하여 사용하고 있습니다.
  + Nginx: 웹 서버로는 Nginx를 사용하였습니다. 관리자 페이지를 서비스 하기 위함과 동시에 서비스에 사용되는 프로세스의 포트번호를 숨겨 외부 공격을 방지하고자 했습니다.
  + JSP: 기존 어플리케이션 서비스를 위해 사용하던 JSP 서버입니다. API의 NodeJS로 이전을 완료하였기에 새 버전이 배포되고 나면 종료될 예정입니다.
  + NodeJS: API 제작을 위해 NodeJS와 express를 사용하였습니다. 라우트를 통해 API의 버전을 관리하고 있습니다.
  + PM2: NodeJS 어플리케이션의 프로세스 관리를 위해 사용하였습니다. 현재 실행되고 있는 어플리케이션은 실제 서비스용 API 제공 어플리케이션과 테스트용 API 어플리케이션입니다. 실제 서비스용 NodeJS 어플리케이션은 PM2를 사용하여 2개의 프로세스로 작동하도록 클러스터링 하였습니다.
  + MySQL: 각종 데이터를 저장하기 위한 DB입니다. EC2 인스턴스 내에서 작동하고 있습니다.
* 형상관리
  + Gitlab: 그룹 관리와 비공개 저장소에 추가 비용이 발생하지 않는 점으로 인해 Gitlab을 사용하였습니다.
* Task관리
  + Trello: 일정 혹은 일감 관리를 위해 Trello를 사용하였습니다. 아래는 실제 사용한 Trello 보드의 화면입니다.  
    스크린샷이(가) 표시된 사진

    자동 생성된 설명