- 2024 전기 졸업과제 착수보고서 -

컨테이너 오케스트레이션 기반 서버리스 프로젝트 배포 시스템 구축

부산대학교 정보컴퓨터공학부

지도교수: 탁성우

201924501 신예준
201924515 유승훈
201924435 김선우

[목차]

1. 과제 개요

1.1. 개요

학생들은 텀프로젝트나 비교과 활동을 통해 다양한 프로젝트를 개발하지만 실행 가능한 형태로 결과물을 공유하는 경우는 드뭅니다. 결과물을 배포하지 않는 주된 이유는 상용 클라우드 서비스의 비용 부담이나 기술적 어려움 때문입니다. 그래서 대부분 본인 개발 환경에서 실행한 화면을 녹화하여 보여주는 간접적인 방식을 통해 결과를 공유하고 있습니다.
이에 본 프로젝트에서는 서버리스 프로젝트 배포 시스템을 개발하여 학생들이 비용이나 기술적 장벽 없이 자유롭게 자신의 프로젝트를 배포하고 공유할 수 있도록 지원합니다. 시스템은 프로젝트를 배포 템플릿을 활용해 컨테이너로 쉽게 배포할 수 있도록 하고, Auto Scaling과 Scale to 0를 적용함으로써 평소에는 비활성 상태로 유지하여 자원 사용을 최소화합니다. 이러한 구성을 통해 제한된 자원을 효율적으로 사용해 많은 프로젝트를 배포할 수 있도록 합니다. 이를 통해 학생들은 자신의 프로젝트를 배포하여 실행 가능한 형태로 공유함으로써 쉽게 피드백을 받아볼 수 있고 포트폴리오로도 활용할 수 있습니다.

1.2. 기획 배경

- **프로젝트 배포 및 공유의 부재**
  학생들은 텀프로젝트나 비교과 활동들을 통해 프로젝트를 완성하지만 배포 경험이 부족하거나 비용 문제로 인해 실제로 배포를 진행하여 공유하지 않는 경우가 많습니다. 그렇기에 본 프로젝트를 통해 학생들이 자신의 작업을 쉽게 배포하고 공유할 수 있는 환경을 제공합니다.

- **프로젝트 배포를 위한 비용 문제**
  프로젝트 배포와 운영을 위해서는 다양한 비용이 발생하게 됩니다. 특히 클라우드 서비스의 경우 무료 이용 기간이 종료되거나 잘못된 설정으로 상당한 비용이 발생할 수 있습니다. 그렇기에 본 프로젝트는 학생들이 비용 부담 없이 자신의 프로젝트를 배포하고 운영할 수 있는 시스템을 제공합니다.

- **지속적인 서비스 운영이 어려움**
  프로젝트를 배포 단계까지 진행하더라도 지속적으로 운영되지 못하는 경우가 많습니다. 지난해 진행된 카카오 테크 캠퍼스에서도 프로젝트 개발을 진행하였고 배포까지 완료하여 GitHub에 배포 주소를 노출하고 있습니다. 하지만 현재는 접속을 시도하면 오류가 발생하며 서비스를 확인할 수 없습니다. 이러한 문제는 배포 후 접속자가 없는 경우에도 유지 비용이 지속적으로 비용이 발생하여 배포된 클라우드 인프라를 삭제하기 때문입니다. 따라서 이 프로젝트는 접속자가 없는 경우 자원 사용을 최소화할 수 있는 서버리스로S cale to 0 기능을 통해 효율적으로 활용할 수 있도록 합니다.

- **프로젝트를 실행하려면 과정이 복잡함**
  배포되지 않은 다른 사람의 프로젝트를 실행하면 사용한 프레임워크나 패키지를 모두 설치해야 하기에 실행 과정이 복잡해지고 시간이 많이 소요됩니다. 그렇기에 직접 실행하기보다 동영상 녹화를 통한 간접적인 방식으로 전달하고 있는데 본 시스템에서는 개발자가 간편하게 배포하고 즉시 프로젝트를 실행해볼 수 있는 주소를 공유함으로서 프로젝트를 쉽게 실행할 수 있습니다.

1.3. 기대 효과

- **성과 공유**
  학생들은 배포된 프로젝트를 통해 다른 사용자에게 공유할 수 있습니다. 이를 통해 피드백을 받고 지속적인 개선과 발전을 이룰 수 있습니다.

- **포트폴리오 활용**
  배포한 프로젝트를 포트폴리오로 활용하여 대외 활동이나 구직 활동 시 경쟁력을 강화할 수 있습니다. 실행 가능한 프로젝트를 제출함으로써 본인의 역량을 효과적으로 보여줄 수 있습니다.

- **교수자 편의성 증가**
  교수자는 학생들의 프로젝트를 실행할 때마다 다른 의존성 패키지를 설치해야하는 어려움이 있습니다. 하지만 배포 시스템을 통해 배포된 결과물을 제출받는다면 이러한 번거로움 없이 배포된 프로젝트를 바로 실행할 수 있습니다. 또한 CSV 파일을 통한 LMS 시스템과 정보 교환을 가능하도록 구성하여 평가 과정을 더욱 간편하고 효율적으로 진행할 수 있습니다.

1.4. 시장 현황

클라우드 플랫폼과 서버리스 컴퓨팅은 이미 전 세계적으로 많은 기관과 기업에서 널리 사용되고 있습니다. 그러나 다양한 장점에도 불구하고 학교와 같은 교육 기관에서의 활용에는 몇 가지 제약이 존재합니다.

- **퍼블릭 클라우드: AWS, Azure, GCP 등**
  퍼블릭 클라우드 서비스는 유연성과 확장성을 제공하며 많은 기업과 개발자가 사용하고 있습니다. 대부분 무료 요금제를 제공하지만 실제로 서비스를 이용하기 위해서는 결제 카드의 등록이 필수적이고, 이는 잘못된 설정이나 예상치 못한 트래픽 증가로 인해 예상치 못한 비용이 발생할 수 있는 리스크를 내포하고 있습니다. 반면, 자체적으로 프로젝트 배포 시스템을 구축해서 사용한다면 내부적으로 할당량을 설정하여 교육 목적으로 활용할 수 있어 이러한 비용 발생 리스크 없애고 적극적으로 활용할 수 있습니다.

- **호스팅 서비스**
  호스팅 서비스는 프로젝트를 쉽게 배포할 수 있고 무료 요금제를 제공하고 있습니다. 하지만 제한된 개수의 애플리케이션만 등록할 수 있고 업로드 가능한 프레임워크 또한 제한적이기에 활용할 수 있는 경우가 한정적입니다. 그렇지만 교내에 프로젝트 배포 시스템을 구축한다면 컨테이너와 템플릿을 이용해 모든 유형의 프로젝트를 배포할 수 있도록 지원할 수 있고 이를 통해 다양한 교과와 활동들에 활용할 수 있을 것입니다.

2. 과제 구성

2.1. 과제 목표

- **저비용 고효율 배포 시스템 구축**
  - Auto Scaling: 트래픽에 따라 컨테이너를 자동으로 스케일링하여 원활한 서비스를 유지할 수 있도록 합니다.
  - Scale to 0: 평소에는 컨테이너를 비활성화 상태로 유지하여 리소스 사용량을 최소화합니다. 이후 요청이 들어온다면 특정 기간 컨테이너를 실행하여 요청을 처리합니다.
  - 지연 시간 최소화: 접속 시 컨테이너 시작(콜드부트)으로 인해 발생하는 지연을 최소화하는 방안을 연구하여 사용자 경험을 개선합니다.

- **사용자 친화적 인터페이스 제공**
  - 간단하고 직관적인 사용자 인터페이스를 통해 기술적 지식이 제한된 사용자도 쉽게 자신의 프로젝트를 배포하고 관리할 수 있도록 합니다.
  - ZIP 파일 업로드 및 Git과 연동한 자동 배포(CI/CD) 방식을 구현하여 초보자와 개발자 모두 사용하기 편리하도록 합니다.

2.2. 요구사항 분석

- **기존 자원 활용**: 교내에 있는 기존 서버로 구동이 가능하도록 하여 시스템 도입으로 인한 추가적인 비용이 최대한 발생하지 않도록 합니다.
- **자원 효율적 분배**: 한정된 자원을 효율적으로 분배하여 많은 프로젝트가 지속적으로 운영될 수 있도록 합니다.
- **사용자 친화적 배포 과정**: 저학년 학생이나 비전공자도 쉽게 배포할 수 있도록 배포 과정을 간단하게 구성합니다.
- **다양한 유형의 프로젝트 지원**: 웹 프로젝트 뿐만 아니라 콘솔 프로그램 등 다양한 유형의 배포를 지원합니다. 웹 터미널을 통해 콘솔 프로그램도 운영할 수 있도록 구현합니다.
- **평가 및 접근 제어**: 교과에서 활용될 수 있도록 강의에 대한 기능을 지원하고 강의 내에서 배포된 프로젝트에 대한 권한 설정과 접근 제어 기능을 구현합니다. 또한 학습 관리 시스템(LMS)과의 연동을 지원하기 위해 CSV를 통해 수강생 명단을 가져오고, 프로젝트 제출 내역 및 평가 정보를 내보낼 수 있는 기능을 제공합니다.

- **내부 데이터 관리**: 외부 의존적인 요소가 없도록 시스템 내부에 필요한 데이터베이스 및 파일 시스템 등 필요한 요소들을 모두 구성합니다.
- **안전한 시스템 구축**: 사용자가 업로드하는 프로젝트에는 API 키 및 민감한 정보 포함되어 있음으로 이를 보호하기 위해 강력한 네트워크 보안과 접근 제어가 필요합니다.

2.3. 시스템 구성

- **프론트엔드**
  - ■ 로그인/회원가입
    시스템을 이용하기 위해 계정을 등록하고 인증하는 페이지를 제공한다. 계정을 등록하는 경우 특정 도메인의 메일을 사용하여 소속을 인증하여야 합니다.
  - ■ 사용자 정보 수정
    시스템에 저장된 사용자 정보를 수정할 수 있는 페이지를 제공합니다.
  - ■ 프로젝트 생성/수정
    프로젝트를 생성/수정 가능한 페이지를 제공합니. 원하는 서브 도메인을 지정하고 사용한 프레임워크에 맞는 템플릿을 선택해 프로젝트 파일을 업로드하면 자동으로 프로젝트 배포가 진행됩니다.
  - ■ 그룹 생성
    사용자는 다른 사람과 함께 프로젝트를 공유하기 위해 그룹을 생성하고 다른 사용자를 초대할 수 있으며, 그룹원들의 권한을 관리할 수 있습니다.
  - ■ 그룹
    그룹에 프로젝트를 등록하여 다른 사용자와 공유할 수 있으며 동아리 등에서 활용할 수 있는 기능입니다.
  - ■ 강의
    강의에 학생들을 등록하고 과제를 부여할 수 있는 기능을 제공합니다. 과제로 등록된 프로젝트의 권한은 강의 담당 교수자가 모두 일괄적으로 설정할 수 있습니다. 또한 프로젝트를 평가할 수 있는 기능을 제공하여 시스템에서 바로 점수를 입력하고 CSV로 다운로드 받아 사용할 수 있습니다.

- **백엔드**
  - ■ 학생 API
    - 그룹을 생성하여 동아리 및 스터디에 이용할 수 있으며, 해당 그룹을 개인 프로젝트를 등록하는 저장소로 이용이 가능하도록 한다.
    - 그룹 내 관리자 권한을 가진 학생 사용자는 그룹의 세부 정보를 설정하거나, 해당 그룹에 다른 회원들을 초대할 수 있도록 한다.
    - 그룹 내 프로젝트를 등록할 수 있으며, 등록 시 프로젝트의 규격이 사전에 정의된 프로젝트 템플릿과 호환이 되는지 검사할 수 있도록 한다.
    - 본인이 등록한 프로젝트에 대한 편집/삭제가 가능하도록 한다.

■ 교수자 API

- 수업을 생성하고 과제를 평가 및 관리할 수 있는 API를 제공한다. 과제 평가 시, 평가 중에 입력한 점수를 CSV로 출력해 LMS 서비스와 연동 가능하게 한다.
- 교수자가 생성한 수업에서 과제를 관리하는 경우, 과제 내 프로젝트 등록의 마감 기한 및 공개 여부를 설정하여 학생들의 접근을 제어할 수 있도록 한다.

■ 인증 API

- 회원가입 시 소속 대학 이메일 인증을 통해, 재학 사실을 입증한다.
- 로그인 시, 사용자의 이메일 및 비밀번호를 인증하고 세션을 생성하여 이후 인가 처리에 사용할 수 있도록 한다.

■ 내부 API

- 배포된 프로젝트의 접근 제어를 위해 세션으로 사용자의 프로젝트 접근 인가 여부를 조회하여 제공한다.

■ 관리자 API

- 서버, 스토리지, 네트워크 등 인프라 리소스의 상태를 실시간으로 모니터링이 가능하도록 한다.
- CPU 사용량, 메모리 사용량, 디스크 공간, 네트워크 트래픽 등 다양한 메트릭을 수집하고 시각화한다.
- 서비스 가용성, 응답 시간, 에러율 등의 지표를 수집하고 분석 가능 하도록 한다.
사용자 활동, 보안 이벤트, 시스템 변경 사항 등의 감사 로그를 기록하고 분석할 수 있도록 한다.

■ 메일 발송 시스템

- 자체 메세지 큐와 작업 프로세싱 스레드를 사용하여, 비동기 이메일 발송을 지원한다.

● **데이터베이스**

■ SQL 데이터베이스

SQL 데이터베이스는 관계형 데이터 모델을 기반으로 데이터를 저장하고 검색하는 데 사용됩니다. 이 시스템에서는 사용자 데이터, 프로젝트 설정 및 트랜잭션 로그 등을 관리하는 데 사용됩니다. 데이터의 일관성과 무결성을 보장하기 위해 트랜잭션을 지원한다.

■ 인메모리 데이터베이스

빠른 데이터 접근을 요구하는 작업을 위해 사용되는 데이터베이스입니다. 모든 데이터를 메모리에 저장하여 고속 데이터 처리와 접근 속도를 제공하며, 게이트웨이에서 라운팅 작업 등 실시간 처리가 필요한 작업에 사용됩니다.

● **인프라**

■ 사설 네트워크

사설 네트워크는 시스템의 보안을 강화하기 위해 공용 인터넷에서 분리된 내부 네트입니다. 이를 통해 서버와 서비스들이 외부 위협으로부터 보호받을 수 있고, 내부 통신의 안전성과 속도를 높입니다.

■ 메시지 큐
내부 시스템 간의 비동기 메시지 전송을 처리하기 위해 사용됩니다. 메시지 큐는 시스템 부하가 높을 때 메시지를 임시로 저장하고, 시스템이 처리 준비가 됐을 때 메시지를 전달하여 효율성과 안정성을 높입니다.

■ 스토리지 서버
업로드한 프로젝트 파일과 관련 데이터를 저장하고 관리하는 서버입니다. 스토리지 서버를 분리함으로써 데이터의 백업 및 복업을 쉽게 진행할 수 있고, 여러 시스템에서 공유해야 하는 데이터를 통합하여 관리할 수 있습니다.

■ 프로젝트 빌드 시스템
사용자가 업로드한 템플릿과 프로젝트 소스 코드를 자동으로 컨테이너 이미지로 빌드하는 시스템입니다. 빌드된 컨테이너 레지스트리에 저장됩니다.

■ 컨테이너 레지스트리
빌드된 컨테이너 이미지를 저장하고 관리하는 서비스입니다. 버전 관리와 액세스 제어를 제공합니다.

■ 게이트웨이
사용자가 배포된 주소로 접속하면, 게이트웨이는 호스트 도메인을 확인하고 적절한 내부 컨테이너로 트래픽을 라우팅하여 서비스를 제공합니다.

■ 인증서 발급 시스템
시스템에서 사용하는 도메인과 사용자가 등록한 커스텀 도메인에 대한 인증서 발급 절차를 진행합니다. 이를 통해 배포되는 모든 프로젝트는 별도의 설정 없이 HTTPS가 적용되어 안전한 통신을 할 수 있습니다.

■ 컨테이너 오케스트레이션
여러 컨테이너의 배포, 관리 및 확장을 자동화하는 시스템입니다. 이는 컨테이너화된 애플리케이션의 관리를 간소화하고, 자동 확장, 부하 분산 등의 기능을 제공합니다.

■ 프록시 서버
사용자가 컨테이너에 직접 접근하여 작업할 수 있도록 내부 컨테이너에 프록시를 제공합니다.

2.4. 시스템 구성도

3. 수행 계획

3.1. 역할 분담

이름	역할 분담
신예준	기획 및 인프라 개발 - 기획 및 아키텍처 설계 - 프로젝트 배포 인프라 개발
유승훈	백엔드 개발 - SQL 데이터베이스 및 API 설계 - Spring Boot로 백엔드 개발
김선우	프론트엔드 개발 - Figma를 활용한 UI 설계 - React로 프론트엔드 개발

3.2. 개발 일정

업무	5				6				7				8				9				10			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
기획 및 설계																								
착수보고서 작성																								
모듈 테스트																								
프론트 개발																								
백엔드 개발																								
인프라 개발																								
시스템 연동																								
디버깅																								
중간보고서 작성																								
개선 사항 분석																								
시스템 개선																								
인프라 구축																								
최종 테스트																								
시스템 배포																								
최종보고서 작성																								
발표 준비																								