|  |
| --- |
| 클라우드 컴퓨팅 보고서 |



|  |  |  |  |
| --- | --- | --- | --- |
| 클라우드 컴퓨팅 프로젝트 보고서 | | | |
| 학과 | 컴퓨터소프트웨어공학과 | 과목 | 클라우드 컴퓨팅 |
| 학번 | 20204026 | 교수님 | 김수현 교수님 |
| 이름 | 정명훈 | 작성일 | 2024.12.01 ~ 2024.12.05 |

|  |
| --- |
| 목차 |
|  |

[1. 프로젝트 개요 4](#_Toc184398171)

[1-1. 프로젝트 주제 4](#_Toc184398172)

[1-2. 프로젝트 명 4](#_Toc184398173)

[1-3. 프로젝트 주요 기술 및 도구 4](#_Toc184398174)

[1-4. 프로젝트 목적 4](#_Toc184398175)

[1-5. 프로젝트 기능 5](#_Toc184398176)

[1-6. 프로젝트 주요 목표 5](#_Toc184398177)

[2. 시스템 설계 6](#_Toc184398178)

[2-1. 요구 사항 분석 6](#_Toc184398179)

[2-2. 전체 시스템 아키텍처 6](#_Toc184398180)

[2-3. 데이터베이스 설계 7](#_Toc184398181)

[2-4. 서비스 흐름도 7](#_Toc184398182)

[2-5. 아키텍처 인스턴스 설명 8](#_Toc184398183)

[2-6. 주요 상호작용 8](#_Toc184398184)

[3. 개발 과정 9](#_Toc184398185)

[3-1. 클라우드 설정 및 구성 9](#_Toc184398186)

[3-2. 기술 스택 9](#_Toc184398187)

[3-3. 개발 단계 9](#_Toc184398188)

[4. 프로젝트 기능 구현과 설명 12](#_Toc184398189)

[4-1. Compute Engine VM 생성 및 Apache, PHP 설치 12](#_Toc184398190)

[4-2. Cloud SQL 인스턴스 생성 및 MySQL 설정 13](#_Toc184398191)

[4-3. PHP와 Cloud SQL 연결 설정 15](#_Toc184398192)

[4-4. PHP파일 생성 후 var-www-html 파일 ls 15](#_Toc184398193)

[4-5. 고정된 IP 주소 생성 16](#_Toc184398194)

[4-6. IAM을 이용한 권한 부여 및 관리 16](#_Toc184398195)

[4-7. API 활성화 17](#_Toc184398196)

[4-8. GCP 서비스 계정 키 파일 18](#_Toc184398197)

[4-9. GCS(Google Cloud Storage) 버킷 18](#_Toc184398198)

[5. 문제 발생 및 해결 23](#_Toc184398199)

[5-1. PHP (php -v 커맨드 입력 시 버전 출력이 안 되는 현상) 23](#_Toc184398200)

[5-2. PHP와 MySQL 연동 과정에서 에러 발생 24](#_Toc184398201)

[5-3. PHP, mysqli 모듈도 다운받았지만 PHP 해석을 못 하는 문제 발생 24](#_Toc184398202)

[5-4. Composer 설치 문제 27](#_Toc184398203)

[6. 결과 및 성과 30](#_Toc184398204)

[6-1. 34.70.47.182/main.php 30](#_Toc184398205)

[6-2. 34.70.47.182/sign\_up.php 30](#_Toc184398206)

[6-3. 34.70.47.182/home.php 31](#_Toc184398207)

[6-4. 34.70.47.182/profile.php 32](#_Toc184398208)

[6-5. 웹 SNS 서비스를 클라우드 기술을 사용함으로써 장점 33](#_Toc184398209)

[6-6. 온프레미스 형식과 클라우드 방식 사용 시의 비용 차이 비교 34](#_Toc184398210)

[7. 프로젝트를 마치며 36](#_Toc184398211)

[7-1. 소감 36](#_Toc184398212)

|  |
| --- |
| 1. 프로젝트 개요 |

## 1-1. 프로젝트 주제

* 클라우드 컴퓨팅 기술을 사용하여 웹 서비스 구현하기

## 1-2. 프로젝트 명

* Soonstagram <SCH Instagram> with Cloud (\*졸업작품과는 별개의 작품입니다.)

## 1-3. 프로젝트 주요 기술 및 도구

* Google Cloud Platform (GCP): 클라우드 환경에서 서비스를 운영
* MySQL: 게시물 정보와 사용자 데이터를 저장하기 위한 데이터베이스
* PHP: 웹 애플리케이션의 백엔드 로직을 구현하기 위한 서버 측 스크립트 언어
* PDO (PHP Data Objects): PHP에서 MySQL과 안전하게 연결하고 쿼리를 실행하기 위한 데이터베이스 접근 방법

## 1-4. 프로젝트 목적

* 클라우드 기술을 이용한 효율적인 웹 애플리케이션 개발과 이를 통해 서비스 제공
* 같은 순천향 대학생들끼리 만의 소셜 네트워크 사이트를 만들어서 서로 팔로우하고 소통할 수 있는 인터페이스 웹을 개발
* 사용자들이 자신의 게시물을 업로드하고, 다른 사람들과 소통하며, 콘텐츠를 공유할 수 있는 소셜 미디어 플랫폼 제공
* 같은 순천향 학생들을 넘어 인스타그램과 같은 전세계 SNS 서비스로 발전
* 실질적인 클라우드 기술 사용 경험을 쌓고, 이를 활용한 서비스 개발 과정을 익히는 것을 목표
* 클라우드 기반 서비스의 효율성과 비용 절감 효과 간접적으로 체험
* 개발 중 발생한 다양한 문제를 해결하며 실질적인 개발 및 배포 과정에서의 문제 해결 능력을 향상

## 1-5. 프로젝트 기능

* 로그인/회원가입 기능 – 본인 프로필 관리 가능
* 다른 유저 검색/팔로워 기능
* 게시판 기능 – 글 작성, 수정, 삭제, 열람 기능
* 게시판 댓글/좋아요 기능 – 댓글 작성, 수정, 삭제, 좋아요 누르기/취소 기능

## 1-6. 프로젝트 주요 목표

* GCP를 활용하여 안정적이고 확장 가능한 서비스를 구현
* Cloud SQL을 사용하여 데이터 저장을 효율화
* MySQL 및 PHP 기반의 데이터베이스 관리와 서버 로직 구현

|  |
| --- |
| 2. 시스템 설계 |

## 2-1. 요구 사항 분석

Soonstagram에 회원으로 가입하려면 이름, 성별, 나이, 학번, 아이디(닉네임), 비밀번호를 입력해야 한다.

가입한 회원에게는 본인의 프로필이 주어진다.

각각의 유저들은 가입하고 고유 id값을 부여받고 그 id로 식별한다.

유저는 여러 개의 게시물을 올릴 수 있고, 게시글 하나는 한 명의 유저만 작성, 수정, 삭제할 수 있으며, 게시물에 대한 작성자, 내용, 사진 정보를 유지해야 한다.

유저는 여러 개의 댓글을 달 수 있고, 댓글에 대한 작성자, 내용 정보를 유지해야 한다.

유저는 게시물에 좋아요를 누르거나 취소할 수 있다.

유저는 다른 상대방 유저를 검색할 수 있고, 그 사람의 프로필을 볼 수 있으며, 팔로우를 할 수 있다.

게시글은 post\_id로 식별한다.

하나의 게시물에 여러 댓글이 달릴 수 있다.

하나의 게시물에 여러 유저에게 좋아요를 받을 수 있다.

한 유저는 여러 유저와 팔로우 관계를 맺을 수 있다.

## 2-2. 전체 시스템 아키텍처

* 사용자 (User): 웹 브라우저를 통해 웹 애플리케이션에 접속하고 서비스에 액세스하는 클라이언트
* 사용자 인터페이스(UI): 사용자는 웹 애플리케이션에 접속하여 게시물 작성 및 업로드를 진행
* 서버: 사용자의 요청을 처리하고 응답을 반환. GCP의 Compute Engine을 사용하여 PHP 기반의 웹 애플리케이션을 실행 -> Compute Engine에서 호스팅 된 Apache와 PHP로 구현
* 데이터베이스: Cloud SQL (MySQL) 인스턴스를 사용하여 사용자와 게시물 데이터를 저장하고 관리
* 애플리케이션 로직: PHP로 작성된 서버 측 코드가 사용자의 요청을 처리하고 데이터베이스와 연동하여 필요한 데이터 반환
* 인터넷 (Google Cloud Network): 사용자와 GCP 리소스 간의 안전하고 빠른 연결을 제공

## 2-3. 데이터베이스 설계

|  |  |
| --- | --- |
| soonstagramusers | 이름, 성별, 나이, 학번, 아이디, 비밀번호 |
| posts | 유저 아이디, 작성자, 내용, 사진 |
| likes | 유저 아이디, 게시물 아이디, 누른 자 |
| comments | 유저 아이디, 게시물 아이디, 작성자, 내용 |
| follows | 팔로워 한 사람 아이디, 팔로우 받은 사람 아이디 |

## 2-4. 서비스 흐름도

* 사용자 로그인: 사용자는 로그인 화면을 통해 인증을 받고, 로그인한 상태에서 게시물을 업로드
* 게시물 업로드: 사용자가 게시물을 작성
* 게시물 표시: 업로드 된 게시물들을 확인할 수 있으며, 각 게시물에 댓글, 좋아요 기능 사용
* 다른 사용자 검색 및 팔로워
* 사용자 로그아웃

## 2-5. 아키텍처 인스턴스 설명

* 클라이언트 (사용자) -> 웹 브라우저

웹 애플리케이션에 요청

* 웹 서버 (GCP Compute Engine)

GCP Compute Engine 인스턴스에서 호스팅

PHP와 Apache를 실행하여 사용자 요청 처리

Cloud SQL을 통해 데이터 저장 및 조회

* Cloud SQL (MySQL 데이터베이스)

데이터베이스에 데이터 저장

사용자의 요청에 따라 데이터 제공

* GCS(Google Cloud Storage) 버킷

대규모 정적 데이터 저장에 더 저렴하기에 이미지 데이터 관리

## 2-6. 주요 상호작용

* 사용자 <-> 서버 상호작용

사용자가 게시물을 작성하면, 웹 서버에서 이를 처리하고 데이터베이스에 저장

* 서버 <-> Cloud SQL 상호작용

데이터베이스 쿼리를 통해 사용자와 게시물 정보를 저장하거나 조회

|  |
| --- |
| 3. 개발 과정 |

## 3-1. 클라우드 설정 및 구성

* GCP 인스턴스: GCP의 Compute Engine을 사용하여 웹 서버를 구성
* MySQL 데이터베이스: GCP의 Cloud SQL을 사용하여 MySQL 데이터베이스 설정
* GCP IAM: 권한 관리 및 액세스 제어를 통해 GCP 리소스를 안전하고 효율적으로 관리

## 3-2. 기술 스택

* 클라우드 서비스: Google Cloud Platform

Cloud SQL, Compute Engine, IAM, API 및 서비스, GCS 버킷, 서비스 계정 키 파일

* 백엔드

PHP: 서버 로직 작성 및 데이터베이스 연동

PDO: MySQL 데이터베이스와 통신

* 프론트엔드

HTML, CSS를 활용한 UI 구축

* 데이터베이스

Cloud SQL (MySQL 기반)을 사용하여 데이터 관리

## 3-3. 개발 단계

3-3-1. 요구사항 분석 및 설계

* 요구사항 도출: 사용자 등록/로그인, 게시물 작성, 이미지 업로드, 댓글 및 좋아요
* 시스템 설계

데이터베이스 설계: 테이블 구조 정의

아키텍처 설계: 클라이언트 요청 -> Compute Engine (PHP 서버) -> Cloud SQL

3-3-2. 환경 설정

* GCP 설정

Compute Engine VM 생성 및 Apache, PHP 설치

Cloud SQL 인스턴스 생성 및 MySQL 설정

IAM을 통해 권한 관리 및 액세스 제어 설정

* GCS 버킷 관리하기 위해 API 활성화

GCS 버킷 -> 대규모 정적 데이터 저장에 더 저렴하기에 이미지 데이터 관리

feisty-well-443311-a6-ec9ad4e95f11.json (Google Cloud 서비스 계정 키 파일)

-> GCP의 리소스에 안전하게 액세스하기 위한 인증 정보를 포함

(API를 사용하는 서비스가 GCP 리소스에 접근할 때 사용)

-> GCS 버킷 사용할 때 서비스 계정 인증 역할

* 개발 환경 구축

VM 인스턴스와 Cloud SQL 연결 설정

3-3-3. 기능 구현

* 사용자 인증: 사용자 등록, 로그인, 세션 관리 구현
* 게시물 작성: 게시물 내용과 이미지를 업로드하는 폼 생성
* 이미지 업로드
* 게시물 표시: 게시물의 데이터와 이미지 데이터를 가져와 사용자 프로필 및 메인 피드에 표시
* 상호작용 기능: 좋아요 및 댓글 기능 구현

3-3-4. 사용한 클라우드 기술

* GCP Cloud SQL

PDO를 사용하여 Cloud SQL과 안전하게 연결

쿼리를 통해 게시물, 사용자, 댓글 등의 데이터를 관리

* GCP Compute Engine

웹 서버를 구성

* GCP IAM

권한 관리 및 액세스 제어 설정

* GCP API

GCS 버킷 관리하기 위해 API 활성화

* GCS(Google Cloud Storage) 버킷

대규모 정적 데이터 저장에 더 저렴하기에 이미지 데이터 관리

* GCP 서비스 계정 키 파일

GCP의 리소스에 안전하게 액세스하기 위한 인증 정보를 포함

|  |
| --- |
| 4. 프로젝트 기능 구현과 설명 |

## 4-1. Compute Engine VM 생성 및 Apache, PHP 설치

soonstagram VM 인스턴스 생성

텍스트, 번호, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

PHP와 Apache 설치 후 연동

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

## 4-2. Cloud SQL 인스턴스 생성 및 MySQL 설정

mysql-soonstagram 인스턴스에 연결하여 필요한 데이터베이스 구축

soonstagram 데이터베이스를 생성하고 필요한 테이블 생성

텍스트, 소프트웨어, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

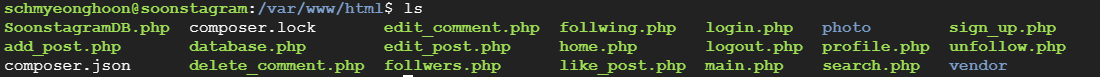
## 4-3. PHP와 Cloud SQL 연결 설정

mysql-soonstagram 인스턴스의 고유 IP주소를 이용하여 접근

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

## 4-4. PHP파일 생성 후 var-www-html 파일 ls



## 4-5. 고정된 IP 주소 생성

고정 IP를 사용해 항상 동일한 주소로 접근 가능하게 설정

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

## 4-6. IAM을 이용한 권한 부여 및 관리

IAM은 클라우드 리소스에 대한 사용자 및 서비스 계정의 접근 권한을 관리하는 도구

서비스 계정의 적절한 역할 부여와 관리를 통해 프로젝트의 안정성과 보안성을 확보

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

IAM을 통해 GCS 버킷과 Cloud SQL 데이터베이스에 접근하는 사용자 및 서비스 계정을 설정

서비스 계정은 GCS를 활용한 파일 업로드 작업 및 Cloud SQL에 데이터를 저장하는 작업을 수행하며, 각각의 역할에 필요한 최소 권한을 부여하여 보안을 강화

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

## 4-7. API 활성화

GCP API는 Google Cloud의 서비스와 리소스(GCS, Cloud SQL, Compute Engine 등)에 대한 프로그래밍 인터페이스를 제공

애플리케이션이 클라우드 리소스와 통신하고, 리소스를 생성, 관리, 모니터링하는 기능을 제공

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

API를 통해 클라우드 리소스를 프로그래밍 방식으로 생성, 수정, 삭제, 모니터링

API 호출과 관련된 트래픽, 응답 시간(지연 시간), 오류율을 실시간으로 확인하여 서비스 상태를 분석하고 문제를 해결

## 4-8. GCP 서비스 계정 키 파일

JSON 키 파일 경로를 환경 변수로 설정

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

## 4-9. GCS(Google Cloud Storage) 버킷

텍스트, 폰트, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

composer.json 파일에 google/cloud-storage 패키지가 포함되어 있는지 확인

스크린샷, 텍스트, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

composer가 vendor/ 디렉토리 안에 필요한 라이브러리 파일을 다운로드

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

버킷에 접근하기 위한 액세스 권한 부여

- 테이블 posts 수정

URL 저장하기 위함

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명

- add\_post.php 수정 -> 기존의 Cloud SQL 인스턴스만 사용하던 코드에 버킷 사용 위해

GCS 클라이언트 설정: google/cloud-storage SDK를 사용하여 GCS 클라이언트를 초기화하고, 인증 정보를 설정

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

add\_post에서 GCS에 파일 업로드 후 URL 저장

텍스트, 스크린샷이(가) 표시된 사진

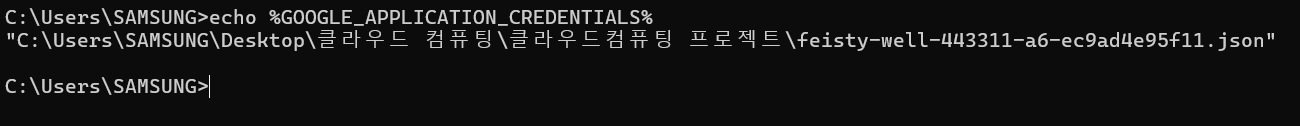
자동 생성된 설명

JSON 키 파일 경로를 환경 변수로 설정하였었지만 접근 문제로 인해 방식 교체

GOOGLE\_APPLICATION\_CREDENTIALS 환경 변수를 설정 -> 시스템 환경 변수를 이용

텍스트, 폰트, 라인, 번호이(가) 표시된 사진

자동 생성된 설명



|  |
| --- |
| 5. 문제 발생 및 해결 |

## 5-1. PHP (php -v 커맨드 입력 시 버전 출력이 안 되는 현상)

* PHP를 설치하였지만 계속해서 php -v를 입력했을 때 php 버전이 출력되지 않고, 웹에서 php 화면이 출력되지 않는 문제가 지속적으로 발생

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

위의 4-1. Compute Engine VM 생성 및 Apache, PHP 설치에서 첨부한 사진을 보면

sudo apt-get update

sudo apt-get install php-cli

이 명령어를 진행한 후에 php -v를 입력하면 정상적으로 php 버전이 출력되고 있음

이 문제를 해결하면서 구글링을 며칠동안 했었는데 그 결과 PHP CLI(Command Line Interface) 컴포넌트가 설치되지 않았던 것이 문제점

Command Line Interface의 약자로 명령을 통해서 컴퓨터를 제어하는 방법 또는 응용 프로그램을 의미(콘솔)

따라서 php-cli 패키지를 설치하면서 PHP 실행 파일과 PHP CLI가 시스템에 추가되면서 php 명령이 인식되고 PHP 버전도 정상적으로 출력될 수 있게 됨.

## 5-2. PHP와 MySQL 연동 과정에서 에러 발생

* PHP와 MySQL 간의 데이터베이스 작업(연결, 쿼리 실행 등)이 불가능하여 Fatal error 오류가 발생

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

이 문제는 mysqli이 제대로 설치가 되지 않아서 발생하였는데

sudo apt update

sudo apt install php-mysqli

이 명령어를 진행한 후에 php -m | grep mysqli을 입력하면 정상적으로 mysqli이 출력되는 것을 볼 수 있음

그 결과 PHP와 MySQL 간의 연동을 가능하게 하는 MySQLi 모듈이 추가되었고, 이제 PHP에서 MySQL 데이터베이스를 활용할 수 있게 됨.

## 5-3. PHP, mysqli 모듈도 다운받았지만 PHP 해석을 못 하는 문제 발생

* 위의 5-1, 5-2 문제를 해결함으로서 PHP와 mysqli 모듈을 정상적으로 모두 다운 받았지만 여전히 PHP를 해석하지 못 하는 상황이 발생

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

PHP와 mysqli을 제대로 설치하였지만 계속해서 PHP를 해석하지 못 하는 상황이 발생

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이 문제는

sudo apt update

sudo apt install libapache2-mod-php

sudo systemctl restart apache2

이 명렁어를 실행한 후에 정상적으로 진행

PHP와 웹 서버(Apache)와 제대로 연동이 안 되어 있어서 이러한 문제가 발생

따라서 Apache와 PHP의 연동 모듈인 libapache2-mod-php을 설치하여 주고 Apache를 재시작하여서 문제를 해결

## 5-4. Composer 설치 문제

* /var/www/html 디렉터리 권한 문제로 인해 설치가 되지 않음
* GCS 버킷 연결하기 위함

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

현재 사용자를 whoami를 통해 확인한 결과 schmyeonghoon으로 되어있었고, /var/www/html 디렉토리의 소유자가 www-data임을 확인하였으며

sudo chown -R schmyeonghoon:schmyeonghoon /var/www/html 명령어를 통해 소유자를 변경

하지만 여전히 composer 명령이 작동하지 않았으며 검색해보니 composer가 시스템에 설치되지 않았거나 PATH에 포함되지 않았다는 내용을 보게 됨

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

php composer-setup.php

php -r "unlink('composer-setup.php');"

이 명령어들을 실행해서 composer.phar 파일을 현재 디렉토리에 다운로드 하였고 하지만 여전히 composer –version 명령을 실행했을 때 나오지 않고 있음을 확인

그래서 sudo mv composer.phar /usr/local/bin/composer 이 명령어를 사용함으로써 composer를 전역적으로 사용할 수 있도록 설정해주었고 이후에 다시 composer –version 명령을 실행했을 때 정상적으로 출력되는 것을 확인

이후에 GCS 버킷 연결하기 위해 composer require google/cloud-storage 명령어를 이용하여 Google Cloud Storage (GCS) SDK 설치를 하는 와중에 오류가 발생

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

The zip extension and unzip/7z commands are both missing 오류와 git was not found in your PATH, skipping source download 두 가지 오류를 확인.

The zip extension and unzip/7z commands are both missing 오류는 PHP가 zip 확장을 지원하지 않거나, 시스템에서 unzip 또는 7z 명령어가 없어서 발생하는 문제이고 PHP의 zip 확장을 설치하고 unzip 도구를 설치해야 한다고 해서

sudo apt-get install php-zip

sudo service apache2 restart

sudo apt-get install unzip

이 명령어를 통해서 Ubuntu에서 PHP zip 확장과 unzip 도구 설치

git was not found in your PATH, skipping source download 오류는 git 명령어가 시스템에 설치되지 않았거나, PATH에 추가되지 않았다는 오류이며 sudo apt-get install git 명령어를 통해 git을 설치

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이제 정상적으로 composer require google/cloud-storage 명령어를 이용하여 Google Cloud Storage (GCS) SDK 설치

|  |
| --- |
| 6. 결과 및 성과 |

## 6-1. 34.70.47.182/main.php

사용자는 로그인을 통해 접속

텍스트, 소프트웨어, 스크린샷, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

## 6-2. 34.70.47.182/sign\_up.php

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

## 6-3. 34.70.47.182/home.php

자신이 팔로우하고 있는 사람들의 게시물을 확인, 댓글 및 좋아요 기능

본인이 작성한 댓글, 게시물만 수정 및 삭제 가능

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

## 6-4. 34.70.47.182/profile.php

home 화면의 오른쪽 상단의 프로필 버튼을 이용해 자신의 프로필 화면으로 입장하여 프로필 정보 수정, 게시물 추가, 다른 유저 검색 기능 그리고 자신이 업로드한 게시물을 확인 가능

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

## 6-5. 웹 SNS 서비스를 클라우드 기술을 사용함으로써 장점

SNS 서비스는 트래픽 변동이 심하고 데이터의 실시간 처리가 중요한 만큼 클라우드 컴퓨팅이 제공하는 확장성, 비용 효율성, 안정성이 필수적

GCP를 활용한 이번 구현에서는 서비스의 안정성과 보안, 빠른 개발 주기가 SNS의 특성에 잘 맞아떨어진다고 생각 함

이러한 장점들은 SNS 서비스의 품질을 높이고 사용자 만족도를 극대화하는 데 크게 기여한다고 생각 함

- 대규모 데이터 관리와 저장에 용이

주로 이미지와 동영상 콘텐츠를 다루며, 이러한 미디어 파일은 큰 저장 공간과 효율적인 데이터 관리가 필요

클라우드 스토리지의 확장성으로 급격히 증가하는 데이터를 처리하기 위해 필요한 용량을 즉시 확장 가능

전 세계 사용자에게 이미지를 빠르게 제공하기 위해 클라우드의 CDN 서비스(AWS CloudFront, GCP CDN)를 활용 가능

- 높은 트래픽 처리 능력

사용자 수가 많고 트래픽이 급증하는 경우가 자주 발생

사용자 트래픽 증가 시, 자동으로 서버 리소스를 추가해 서비스 성능을 유지 가능

여러 서버에 트래픽을 분산시켜 과부하 방지 및 빠른 응답 시간 유지 가능

초기에는 소규모 리소스를 사용하고, 사용자 수가 증가함에 따라 리소스를 유연하게 확장할 수 있어 비용을 효율적으로 관리

- 물리적 서버 구매와 관리에 드는 막대한 초기 비용을 절감

트래픽이 적은 시간대에는 리소스를 줄여 비용을 절감하고, 트래픽이 많은 시간대에는 자동으로 확장하여 자원을 효율적으로 사용

물리적 서버 관리가 필요 없어 인프라 유지보수에 드는 시간과 노력을 절약

## 6-6. 온프레미스 형식과 클라우드 방식 사용 시의 비용 차이 비교

온프레미스 형식과 클라우드 방식 사용 시의 비용 차이를 비교하기 위해 가상의 SNS 웹 서비스(인스타그램과 같은 전세계적인 이미지/동영상 중심 플랫폼)로 설정하고 주요 비용 항목을 비교

* 가상 설정

트래픽: 하루 평균 사용자 10만 명, 피크 시간 동시 사용자 5천 명

데이터 저장: 하루 평균 이미지/동영상 업로드 1TB

네트워크 대역폭: 하루 2TB 데이터 전송

- 온프레미스 방식 (온프레미스 서버 구축)

예상 초기 투자 비용 약 2,300만원

서버 구매: 4대의 고성능 물리 서버 400만원 x 4대 = 1,600만원

스토리지 구매: 스토리지 서버(10TB) 50만원/1TB x 10TB = 500만원

네트워크 장비: 장비 및 방화벽 구성 = 200만원

운영비용: 예상 월간 비용 약 380만원

IT 인력: 월급 약 300만원/월

전기 비용: 서버 4대 x 24시간 x 30일 = 약 50만원

유지보수 및 소모품 비용: 약 30만원

- 클라우드 방식 (GCP 사용)

예상 월간 비용 약 1,067,689원

Compute Engine (VM): ₩194,637/ month

Cloud Storage: ₩259,375/ month

Cloud SQL: ₩161,092/ month

네트워크: ₩452,585/ month

텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| 7. 프로젝트를 마치며 |

## 7-1. 소감

* 이번 프로젝트를 진행하면서 PHP 기반 웹 서비스와 GCP의 통합하는 과정에서 다양한 클라우드 기능을 접해보고 활용해보는 경험을 해보았습니다. 클라우드 기능을 수업시간에 직접 다루어 보긴 했었지만, 클라우드 기능을 직접 다루는 것이 이번이 처음이다 보니까 이번 프로젝트를 진행하면서도 여러 에러가 발생하였고 문제를 해결해 나가는 과정에서 많은 어려움을 겪었습니다.

하지만 이를 하나 하나씩 해결해 나가면서 클라우드의 작동 원리에 대해 좀 더 이해하게 된 계기가 된 것 같습니다. 내년에 AWS 자격증 취득을 목표로 하고 있기에 먼저 클라우드 컴퓨팅에 대해 접해보자는 마음으로 클라우드 컴퓨팅 수업을 수강 신청하였고 이번 프로젝트로 AWS로 구현하려 했으나 구현하는 도중에 역시 많은 어려움을 겪고 프리티어 무료 기간이 종료되는 바람에 GCP로 구현하게 되었습니다.

그래도 AWS뿐만 아니라 GCP를 사용해서 구현함으로써 다양한 클라우드 플랫폼을 경험해봤다는 것에 만족하고 다음 번에는 AWS를 이용해서 경험을 쌓고 클라우드 환경에 대한 좀 더 깊은 이해와 능력을 키워보고 싶습니다.