

# 온라인으로 컴파일 가능한 Web IDE 서비스

202155603 정윤서  
202155619 최범주  
202155508 권내현  
201924486 배명진

# 프로젝트 소개

- 로컬의 별도 IDE 없이, 웹 브라우저를 통해 코드를 작성하고 실행 결과를 확인할 수 있는 Web IDE
- 다양한 프로그래밍 언어(C++, Python, Rust, GO) 지원
- PC가 아닌, 모바일 / 태블릿 환경에서도 실행 가능
- 사용자별로 컨테이너화된 환경으로, 안정성 및 보안 향상

# 프로젝트의 필요성

## 높은 접근성, 편의성

- 수 많은 프로그래밍 언어마다 독립적인 컴파일러 및 IDE 설치 곤란
- 해당 서비스를 사용하면, 별도의 구축 과정이 없으므로 편리하게 개발에 몰두 가능
- 모바일 / 태블릿 환경에서도 문제 없이 프로그래밍 가능

## 통일된 개발 환경 제공

- 교육 환경에서 복잡한 설치 과정, 버전의 차이 등으로 인해 차질을 겪는 경우가 많음
- 해당 서비스를 사용하면, 교육자와 학습자가 모두 동일한 환경에서 프로그래밍 학습 가능

# 기존 기술

## 구름 IDE

- 클라우드 통합 개발 환경(CDE)
- Python, JavaScript, Java, C/C++ 지원
- 실시간 협업 기능 제공
- pay-as-you-go 방식 채용 부분 유료화 서비스
- ‘컨테이너’ 에서 클라우드의 기본적인 성격 확인 가능
- ‘팀’ 기능 이용하여 각 팀원에게 특정 권한 설정 가능

## Codingground

- 웹 컴파일러
- Python, Java, C++ 등 다양한 언어 지원
- 회원가입 및 로그인을 필요로 하지 않으나, 이미 작성한 코드를 웹 상으로 불러올 수 없고 새로고침 시 초기화 됨

# 기존 기술

## 구름 IDE

- 클라우드 통합 개발 환경(CDE)
- Python, JavaScript, Java, C/C++ 지원
- 실시간 협업 기능 제공
- pay-as-you-go 방식 채용 부분 유료화 서비스
- '컨테이너'에서 클라우드의 기본적인 성격 확인 가능
- '팀' 기능 이용하여 각 팀원에게 특정 권한 설정 가능

WebIDE의 기능을  
사용자별로 컨테이너화하여 제공  
➡ 안정성 및 보안 ↑  
: 구름 IDE에서 착안

# 조원 별 담당 파트

## 정윤서

- 프론트엔드와 백엔드 연동
- 사용자가 입력한 데이터를 MySQL DB 테이블과 RabbitMQ의 Queue에 추가
- 실행 결과가 저장된 DB 테이블에서 request\_id가 일치하는 데이터를 받아와 프론트엔드로 전달
- 실행 결과가 DB 테이블에 저장될 때까지 주기적으로 요청하여 새로고침 없이 실행 결과를 화면에 출력

## 권내현

- 사용자가 입력한 코드를 RabbitMQ를 통해 전송받은 후 소스코드로 저장
- 저장한 코드를 선택한 언어에 맞게, 웹컴파일러 기능을 통해 실행
- 소스코드 실행결과를 전용 데이터베이스 테이블에 저장하여 사용자가 실행 결과를 확인할 수 있도록 함

## 최범주

- cloudflare와 서버 연결
- nginx docker image 생성 및 docker compose 구성
- 샌드박스 환경 생성 및 리소스 제한
- azure vm 생성
- linux namespace를 통한 가상 파일 시스템 구축
- Github Action을 통한 CI/CD

## 배명진

- 프론트엔드(React)와 백엔드(Django)를 연동
- 사용자가 원하는 프로그래밍 언어, 테마, 코드를 입력하면, 해당 내용을 백엔드(Django) 서버로 전달
- 적절한 요청에 대한 응답을 받을 때까지 백엔드에게 요청을 반복적(2초 간격)으로 보냄.
- 백엔드로부터 받은 응답을 '컴파일 결과' 화면에 출력

# 실행 결과

## PC 환경

### 코드 컴파일러

언어를 선택하세요

C++

테마를 선택하세요

Light version

코드를 입력하세요

```
1 #include <iostream>
2
3 int main(){
4     std::cout << "Hello, Wrold! This is team 3." << std::endl;
5 }
```

컴파일

초기화

컴파일 결과

Hello, Wrold! This is team 3.

## 모바일 환경

2:03 PM Thu Jun 6



AA

fc.fiene.dev



54%

### 코드 컴파일러

언어를 선택하세요

C++

테마를 선택하세요

Dark version

코드를 입력하세요

```
1 #include<iostream>
2
3 int main() {
4     std::cout << "Hello, World! This is team 3." << std::end
5 }
```

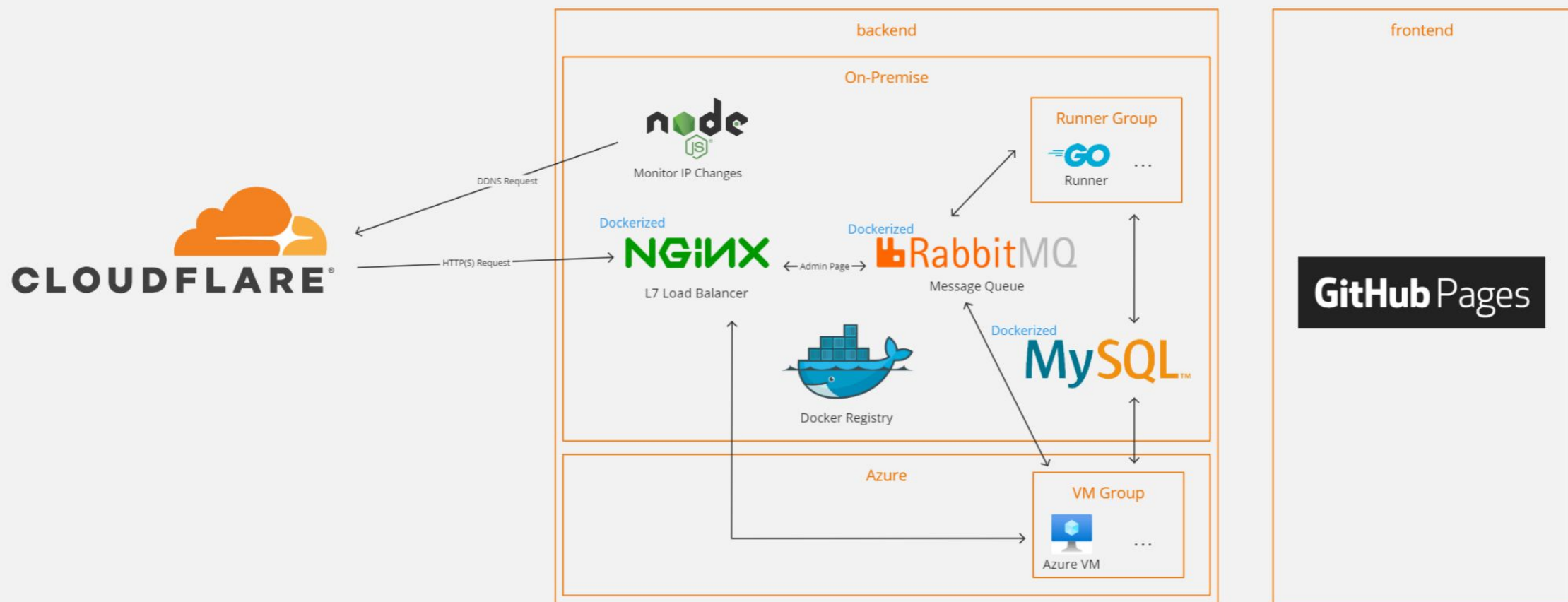
컴파일

초기화

컴파일 결과

Hello, World! This is team 3.

# 매커니즘





# 사용 방법

1. <https://fc.fiene.dev/> 에 접속합니다.

### 코드 컴파일러

**언어를 선택하세요**  
(언어 선택) ▾

**테마를 선택하세요**  
Light version ▾

**코드를 입력하세요**  
1 // 코드를 입력하세요 //

컴파일

초기화

**컴파일 결과**

# 사용 방법

2. 작성하고자 하는 언어 형식을 드롭-다운 방식을 통해 선택한 후, 언어 형식에 맞게 코드를 작성합니다.

## 언어를 선택하세요

C++

(언어 선택)

C++

Python

Rust

Go

## 코드를 입력하세요

```
1  #include<iostream>
2
3  int main() {
4      std::cout << "Hello World! This is Team 3!" << std::endl;
5  }
```

# 사용 방법

3. 코드를 작성한 후에는 컴파일 버튼을 눌러 원격 서버로 코드를 전송합니다.  
버튼을 누르면 코드가 자동으로 전송됩니다.

언어를 선택하세요

C++

테마를 선택하세요

Light version

코드를 입력하세요

```
1  #include<iostream>
2
3  int main() {
4      std::cout << "Hello World! This is Team 3!" << std::endl;
5  }
```

컴파일 결과

Compiling...

# 사용 방법

4. '컴파일 결과' 화면에서, 컴파일 결과를 확인합니다.
  - 컴파일 및 실행 후 에러가 발생하지 않았다면, 정상적으로 실행 결과가 창에 출력됩니다.

## 컴파일 결과

```
Hello World! This is Team 3!
```

# 사용 방법

4. '컴파일 결과' 화면에서, 컴파일 결과를 확인합니다.
- 컴파일 및 실행 후 에러가 발생했다면, 발생한 에러가 '컴파일 결과' 화면에 출력됩니다.

## 컴파일 결과

[illegible]

## 코드 컴파일러

언어를 선택하세요

C++

테마를 선택하세요

Light version

코드를 입력하세요

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6
7 }
```

컴파일

초기화

컴파일 결과

# 활용 방안

- 언제 어디서나 할 수 있는 프로그래밍
  - PC가 없어도, 모바일 및 태블릿 기기를 통한 간단한 프로그래밍
- 교육 환경
  - 각종 프로그래밍 수업에서 통일된 교육 환경 보장 가능
  - 개발 환경 구축으로 인한 시간적, 경제적 비용 감소

**감사합니다**