



지구구조대 최종 발표

김영석 안윤지 한승호 한용준 홍서윤 황수영



목차

1. 시스템 아키텍처

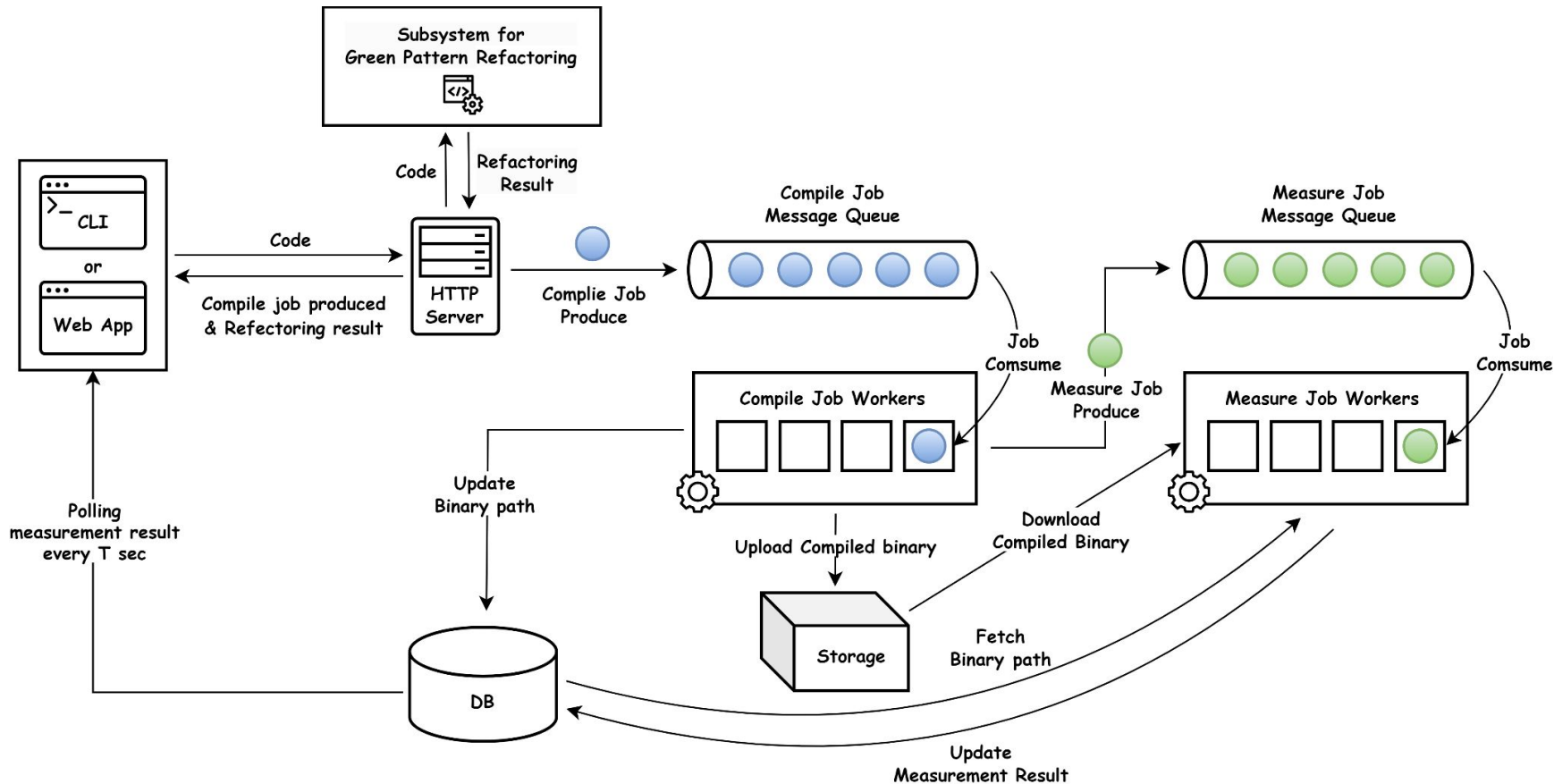
2. 시스템 개발 : 기본 기능

- a. 탄소배출량 측정
- b. 그린패턴 리팩토링

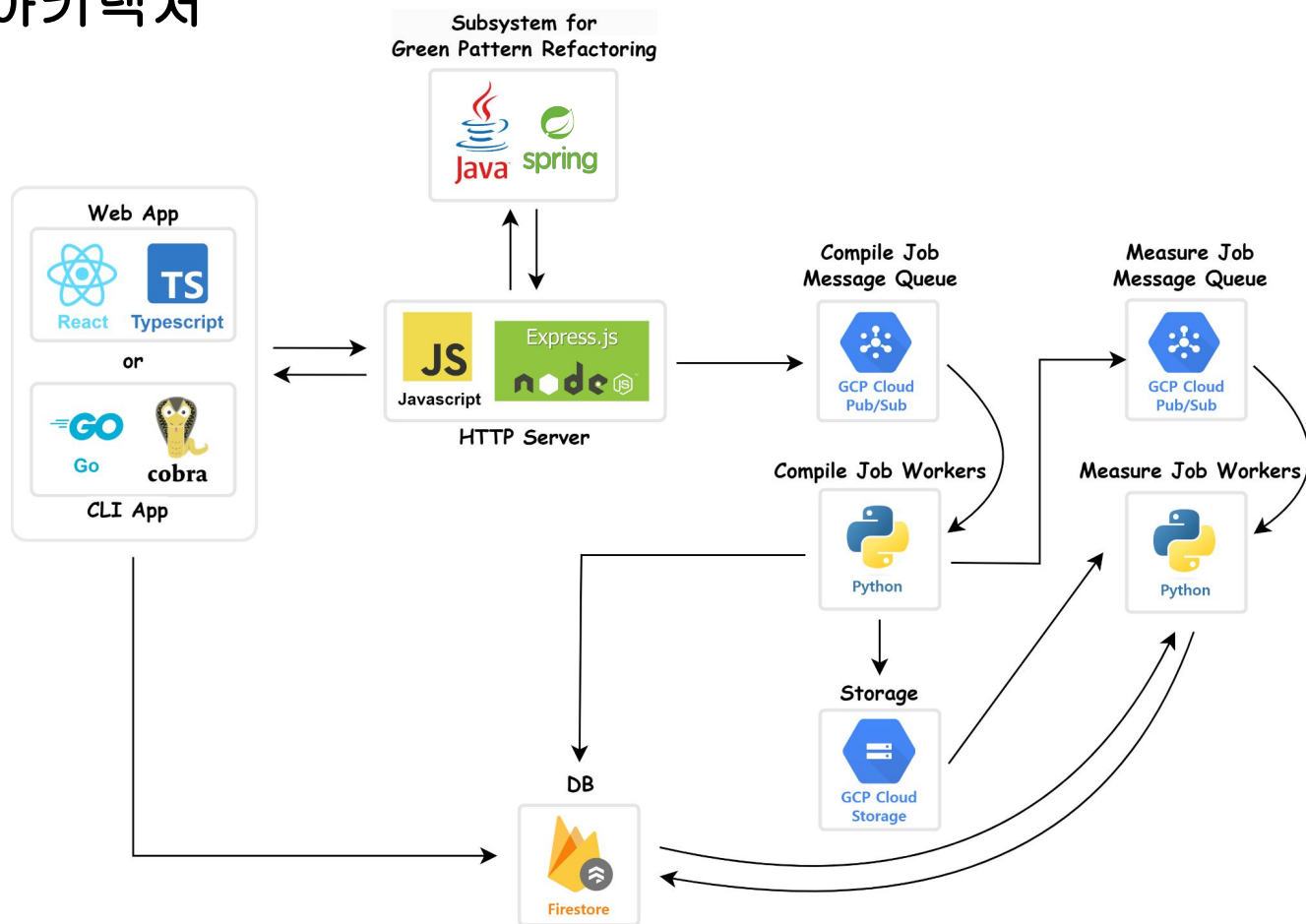
3. 시스템 개발 : 추가 기능

- a. 실시간으로 처리 상태 확인
- b. Producer consumer pattern
- c. CLI app을 통한 Java project 단위 기능 제공
- d. Google app engine을 이용한 배포

시스템 아키텍처



시스템 아키텍처



시스템 개발 - 기본 기능


React App

swe-team9.web.app

Earth Saver


SWE Team9

Earth Saver 지구구조대




탄소 배출량 측정

« »



Green Pattern Refactoring 제안



사용해 보기


JAVA 코드를 입력하고 Submit 버튼을 누르면 코드의 탄소배출량을 측정하고 Green Pattern Refactoring 제안을 받을 수 있습니다.


SUBMIT

완료

```
1 // Write down your code here
2 // Your class name must be Main
3
4 public class Main {
5     public static void main(String[] args) {
6         boolean cond1 = checkCond1();
7         boolean cond2 = checkCond2();
8         boolean cond3 = checkCond3();
9
10        if(cond1) {
11            if(cond2) {
12                if(cond3) {
13                    System.out.println("Hello");
14                }
15            }
16        }
17    }
18 }
```

탄소배출량 (C g) ⓘ 1.548830

 자동차로 0.009330 Km 이동 (연비 16Km/L)

 비행기로 0.000114 Km 이동

시스템 개발 - 기본 기능

(1) 탄소 배출량 측정

- 소스코드 실행 시간 측정 : 최대 10번 or 최대 30초 조건으로 반복 실행 후 평균으로 측정
- cpu, memory 사양에 맞춰 수식 계산

사용해 보기

JAVA 코드를 입력하고 Submit 버튼을 누르면 코드의 탄소배출량을 측정하고 Green Pattern Refactoring 제안을 받을 수 있습니다.

SUBMIT

완료

```
1 // Write down your code here
2 // Your class name must be Main
3
4 public class Main {
5     public static void main(String[] args) {
6         boolean cond1 = checkCond1();
7         boolean cond2 = checkCond2();
8         boolean cond3 = checkCond3();
9
10        if(cond1) {
11            if(cond2) {
12                if(cond3) {
13                    System.out.println("Hello");
14                }
15            }
16        }
17    }
18
19    public static boolean checkCond1() {
20        int sum = 0;
21        for(int i=0; i<10000; i++) {
22            sum += i;
```

탄소배출량 (C g) 0.426137



자동차로 0.002567 Km 이동 (연비 16Km/L)



비행기로 0.000031 Km 이동



나무 0.000003그루가 1년 동안 흡수

시스템 개발 - 기본 기능

(1) 탄소 배출량 측정


React App

swe-team9.web.app

Earth Saver

SWE Team9

Earth Saver 지구구조대



탄소 배출량 측정

사용해 보기

JAVA 코드를 입력하고 Submit 버튼을 누르면 코드의 탄소배출량을 측정하고 Green Refactoring 제안을 받을 수 있습니다.

```
23 }
24 return sum > 50000;
25 }
26
27 public static boolean checkCond2() {
28     int sum = 0;
29     for(int i=0; i<10000; i++) {
30         sum += i;
31     }
32     return sum > 50000;
33 }
```

서버 하드웨어 스펙

모델: Intel Xeon E5-2696 v4 @ 2.20GHz
물리적 CPU 코어 수: 22개
논리적 CPU 코어 수: 1개
코어당 전력 소비율: 150W/22=6.82W
가용 메모리 크기: 0.6GB
메모리 사용량에 대한 전력 소비율: 0.3725W/GB
데이터 센터의 에너지 효율성(PUE): 1.67

탄소배출량 산출식

$$C = E \times CI \quad (1)$$
$$E = t \times (n_c \times P_c \times u_c + n_m \times P_m) \times PUE \times CI \times 0.001 \quad (2)$$

식 (2)에서 각 인자는 C (Carbon Footprint), E (Quantity of Energy), CI (Carbon Intensity), t (Runtime), n_c (Number of Core), P_c (Power of Core), u_c (Core Usage), n_m (Number of memory), P_m (Power of memory), PUE (Power Usage Effectiveness), PSF (Power System Factor)를 의미한다. 프로젝트 특성상 PUE , CI , n_c , P_c , P_m 은 임의의 상수로 계산하며, t , u_c , n_m 만을 측정한다.

닫기

2567 Km 이동 (연비 16Km/L)

시스템 개발 - 기본 기능

(2) Java 소스코드 그린 패턴 리팩토링

Green Pattern Refactoring 결과

탄소배출량을 줄이기 위한 코드 리팩토링 결과를 확인할 수 있습니다.

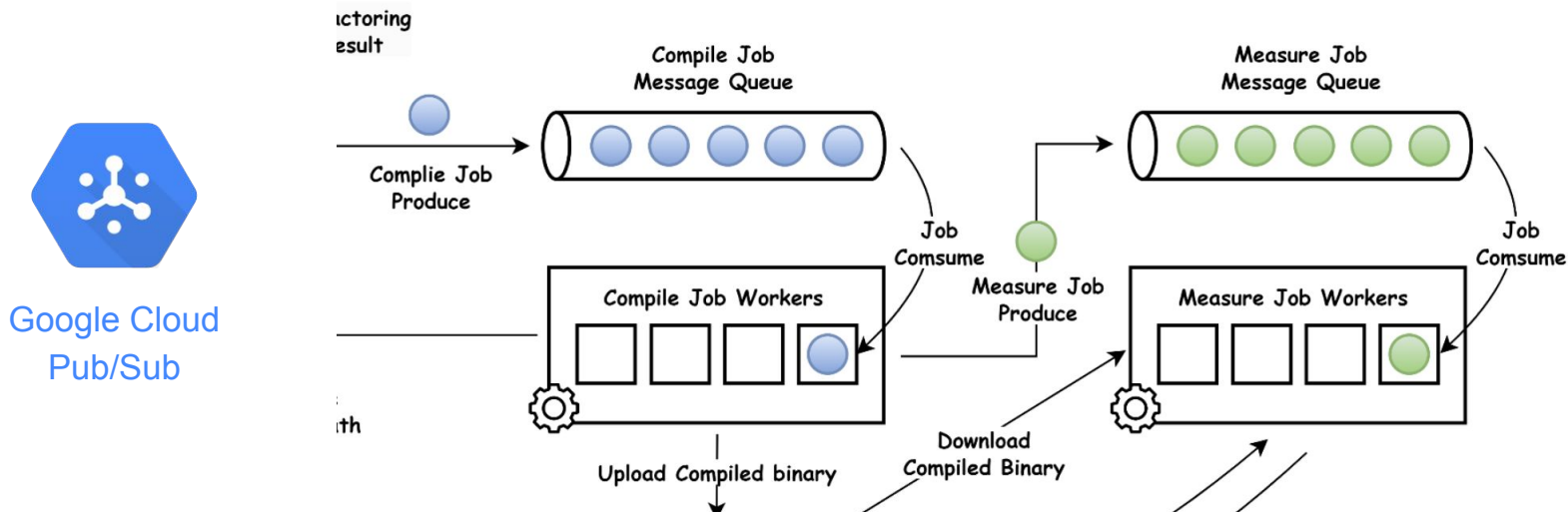
```
1 // Write down your code here
2 // Your class name must be Main
3
4 public class Main {
5     public static void main(String[] args) {
6         boolean cond1 = checkCond1();
7         boolean cond2 = checkCond2();
8         boolean cond3 = checkCond3();
9
10        if(cond1) {
11            if(cond2) {
12                if(cond3) {
13                    System.out.println("Hello");
14                }
15            }
16        }
17    }
18
19    public static boolean checkCond1() {
20        int sum = 0;
21        for(int i=0; i<10000; i++) {
22            sum += i;
23        }
24    }
25}
```

```
1 // Write down your code here
2 // Your class name must be Main
3
4 public class Main {
5     public static void main(String[] args) {
6         boolean cond1 = checkCond1();
7         boolean cond2 = checkCond2();
8         boolean cond3 = checkCond3();
9
10        if((cond1 && cond2) && cond3) {
11            System.out.println("Hello");
12        }
13    }
14
15    public static boolean checkCond1() {
16        int sum = 0;
17        for(int i=0; i<10000; i++) {
18            sum += i;
19        }
20    }
21}
```


시스템 개발 - 추가 기능

(1) 컴파일 & 측정 과정에 **Producer Consumer Pattern** 사용

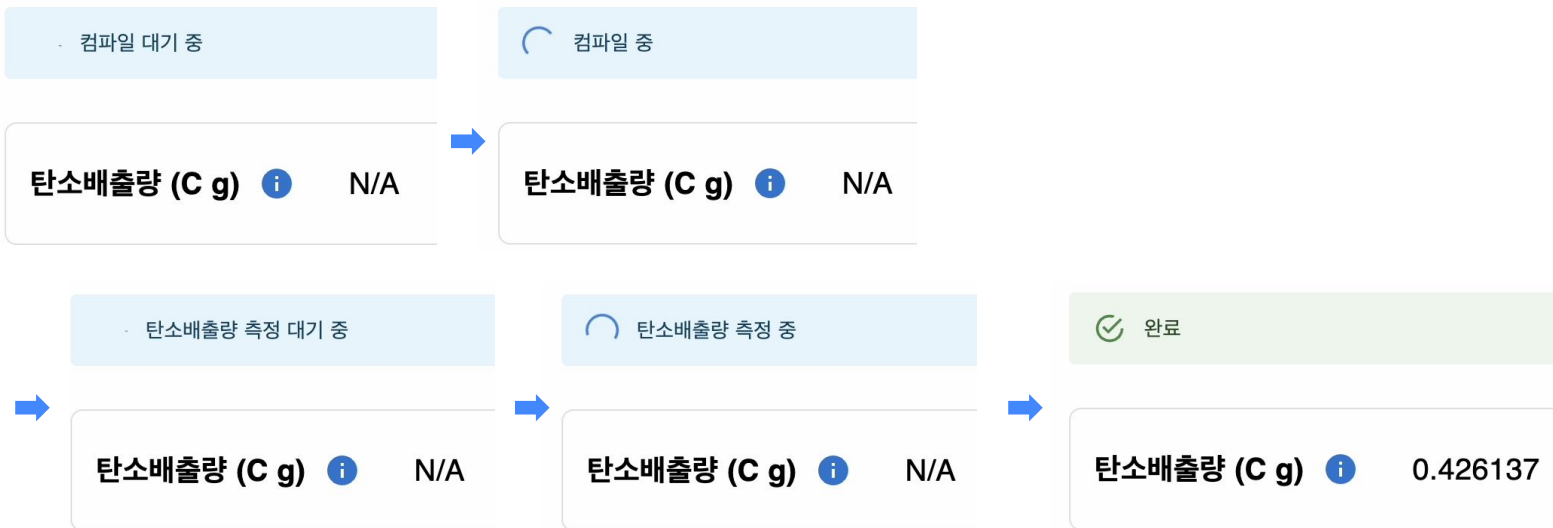
- large scale에서도 그대로 적용될 수 있는 시스템 구조
- worker 개수 늘리는 만큼 동시에 여러 요청 처리 가능
- 각 worker 가 격리된 logical machine에서 동작하므로 변인 최소화하여 측정 가능
- worker failure tolerant



시스템 개발 - 추가 기능

(2) 실시간으로 처리 상태 확인

컴파일 대기 중 → 컴파일 중 → 탄소배출량 측정 대기 중 → 탄소배출량 측정 중 → 완료



시스템 개발 - 추가 기능

(2) 실시간으로 처리 상태 확인

오류 발생 시 오류메시지 출력

Earth Saver

SWE Team9

탄소 배출량 측정

Green Pattern Refactoring 제안


사용해 보기


JAVA 코드를 입력하고 Submit 버튼을 누르면 코드의 탄소배출량을 측정하고 Green Pattern Refactoring 제안을 받을 수 있습니다.

SUBMIT

에러 발생: /tmp/Main.java:4: error: class ee is public, should be declared in a file named ee.java public class ee

탄소배출량 (C g) N/A



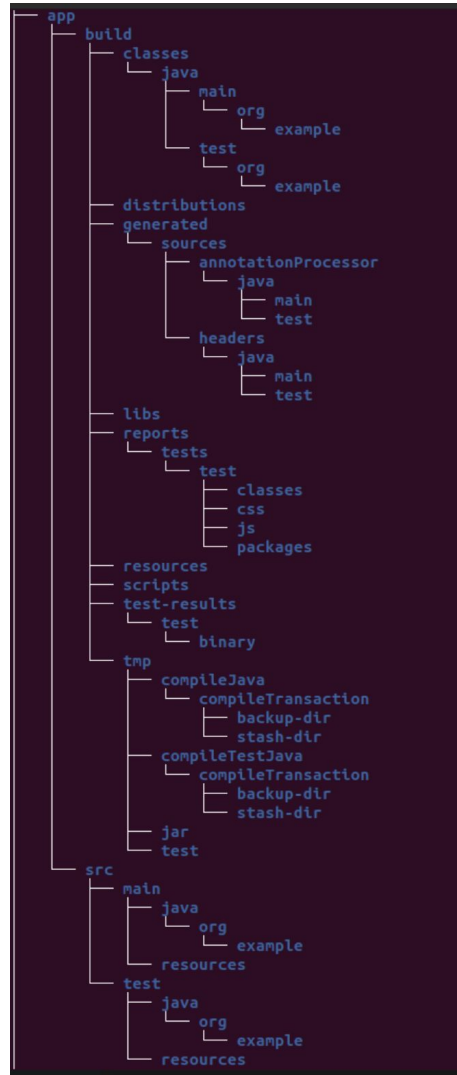


```
1 // Write down your code here
2 // Your class name must be Main
3
4 public class ee {
5     public static void main(String[] args) {
6         boolean cond1 = checkCond1();
7         boolean cond2 = checkCond2();
8         boolean cond3 = checkCond3();
9
10        if(cond1) {
11            if(cond2) {
12                if(cond3) {
13                    System.out.println("Hello");
14                }
15            }
16        }
17    }
18 }
```

시스템 개발 - 추가 기능

(3) CLI app을 통한 Java project 단위 기능 제공

[Java project 구조 예시]



시스템 개발 - 추가 기능

(3) CLI app을 통한 Java project 단위 기능 제공

```
kys2018312565@tibbers:~/gradle_workspace$ sudo earthsaver -m
[sudo] password for kys2018312565:
Sending Project for Measuring...
Wait for measuring amount of Carbon Emission!
COMPILE_ENQUEUED success!
COMPILING success!
MEASURE_ENQUEUED success!
MEASURING success!
..
Measure DONE!!
Your project emits carbon 0.699103 C g, same amount as
Car: 0.004211 km / Plane: 0.000052 km / Tree: 0.000005 그루
kys2018312565@tibbers:~/gradle_workspace$
```

처리 상태 표시

탄소배출량
측정 결과 표시

[CLI app 탄소배출량 측정]

시스템 개발 - 추가 기능

(3) CLI app을 통한 Java project 단위 기능 제공

```
Original Code
0: public class Buggy {
1:     public static void main(String[] args) {
2:         boolean cond1 = checkCond1();
3:         boolean cond2 = checkCond2();
4:         boolean cond3 = checkCond3();
5:
6:         if(cond1) {
7:             if(cond2) {
8:                 if(cond3) {
9:                     System.out.println("Hello");
10:                }
11:            }
12:        }
13:    }
14:
15:    public static boolean checkCond1() {
16:        int sum = 0;
17:        for(int i=0; i<10000; i++) {
18:            sum += i;
19:        }
20:        return sum > 50000;
21:    }
22:
23:    public static boolean checkCond2() {
24:        int sum = 0;
25:        for(int i=0; i<10000; i++) {
26:            sum *= i;
27:        }
28:        return sum > 50000;
29:    }
30:
31:    public static boolean checkCond3() {
32:        int sum = 0;
33:        for(int i=0; i<10000; i++) {
34:            sum *= i;
35:        }
36:        return sum > 50000;
37:    }
38: }
```

```
Refactored Code
0: public class Buggy {
1:     public static void main(String[] args) {
2:         boolean cond1 = checkCond1();
3:         boolean cond2 = checkCond2();
4:         boolean cond3 = checkCond3();
5:
6+        if((cond1 && cond2) && cond3) {
7+            System.out.println("Hello");
8:        }
9:    }
10:
11:    public static boolean checkCond1() {
12:        int sum = 0;
13:        for(int i=0; i<10000; i++) {
14:            sum += i;
15:        }
16:        return sum > 50000;
17:    }
18:
19:    public static boolean checkCond2() {
20:        int sum = 0;
21:        for(int i=0; i<10000; i++) {
22:            sum *= i;
23:        }
24:        return sum > 50000;
25:    }
26:
27:    public static boolean checkCond3() {
28:        int sum = 0;
29:        for(int i=0; i<10000; i++) {
30:            sum *= i;
31:        }
32:        return sum > 50000;
33:    }
34: }
```

[CLI app 리팩토링 결과 확인]

시스템 개발 - 추가 기능

(3) CLI app을 통한 Java project 단위 기능 제공

```
kys2018312565@tibbers:~/buggys$ sudo earthsaver -r
[sudo] password for kys2018312565:
Sending Project for Green Code...
Refactoring Done!!
Getting Diff
Not Saved - ./Buggy2.java
Not Saved - ./Buggy1.java
File Saved - ./Buggy3.java
kys2018312565@tibbers:~/buggys$
```

[CLI app 리팩토링]

시스템 개발 - 추가 기능

(4) Google App Engine (Managed K8s)를 이용한 배포

- Pod auto/manual scale out 손쉽게 가능
- 여러 서비스 각각에 대한 배포 설정을 손쉽게 관리



App Engine

```
measure_worker.yaml
1  runtime: custom
2  env: flex
3  service: measure-worker
4
5  manual_scaling:
6    instances: 1
7
8  env_variables:
9    GOOGLE_APPLICATION_CREDENTIALS: 'pubsub-key.json'
10   FIREBASE_DB_URL: 'https://swe-team9-default-rtdb.firebaseio.com'
11   STORAGE_BUCKET_NAME: 'earth-saver'
12
13  entrypoint: python3 measure_worker.py
14
```