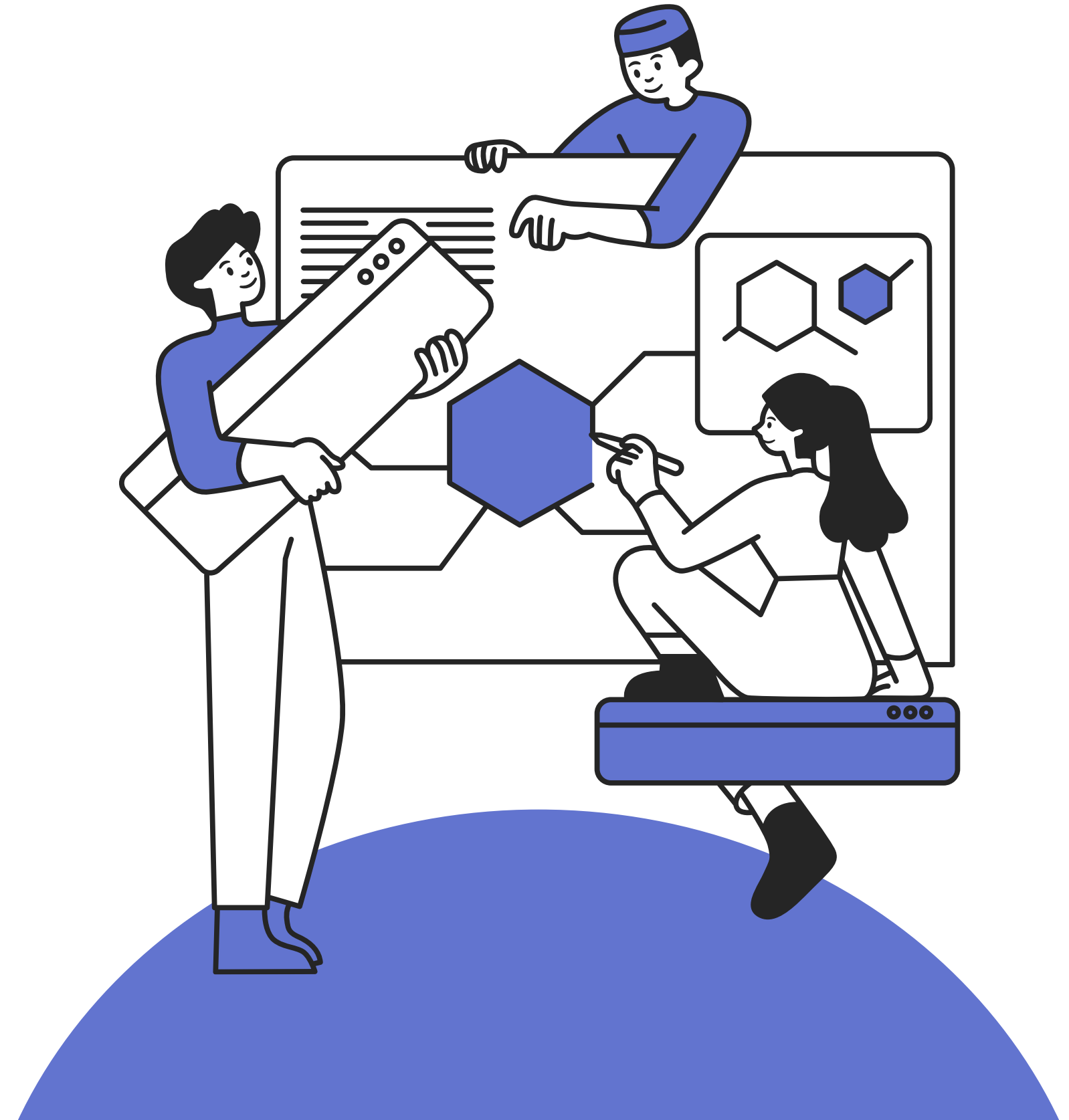


공공데이터 약국 수집 API 및 웹서비스 구축 (Kubernetes 기반)

발표자. 방현민



CONTENTS

- 1 프로젝트 개요
- 2 기술 스택 구성
- 3 전체 시스템 구성도
- 4 주요 함수 설명
- 5 시퀀스 다이어그램
- 6 테스트 결과 요약표
- 7 개발 및 운영 가이드

01

프로젝트 개요

01

목표

공공데이터 포털의 약국 정보를

실시간 수집하여 웹에서 확인

가능한 API 및 UI를

제공하고, Docker + Kubernetes

위에서 실전 배포 환경 구성

02

구성

Flask 백엔드 API (/api/startCollecting)

HTML + JS 프론트 (/web)

Docker 이미지 2개 → Docker Hub 배포

Kubernetes 배포: Deployment, Service, Ingress

MetalLB로 외부 IP 할당

02

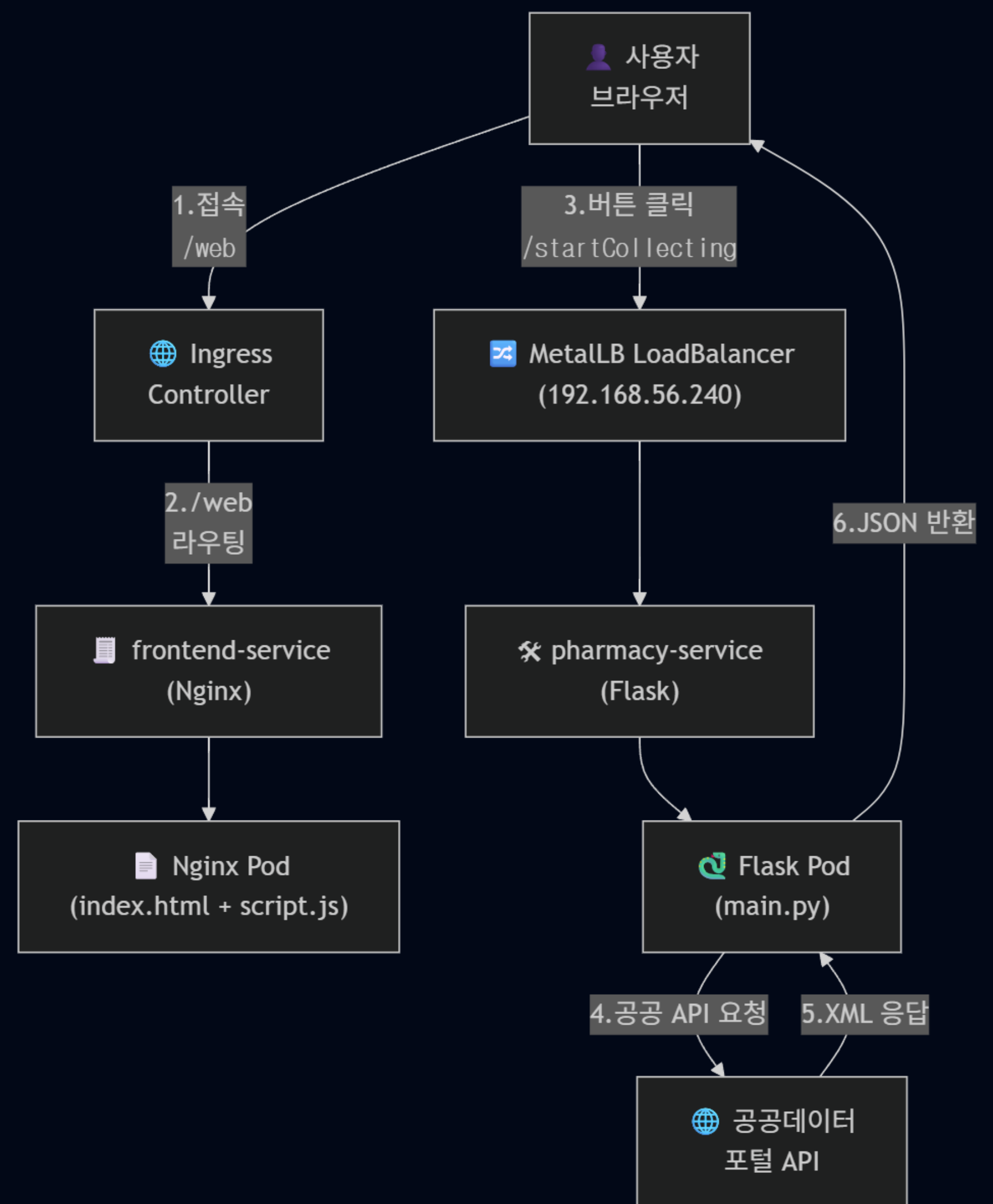
기술 스택 구성

항목	기술
백엔드 API	Flask (Python)
프론트엔드	HTML + jQuery
수집 방식	공공데이터 REST API (XML → JSON 파싱)
컨테이너화	Docker
배포 플랫폼	Kubernetes (Master/Worker 클러스터)
외부 노출 방식	MetalLB (LoadBalancer), Ingress
라우팅 방식	Ingress 경유: /web, LoadBalancer 경유: /startCollecting

03

전체 시스템 구성도

1. 사용자 브라우저는 /web 접속 시 → Ingress → frontend-service → HTML 페이지 반환
2. 버튼 클릭 시 → JavaScript가 MetalLB 외부 IP로 직접 API 요청
3. Flask API는 공공데이터 포털로 요청하여 응답을 가공 후 반환



04

주요 함수 설명

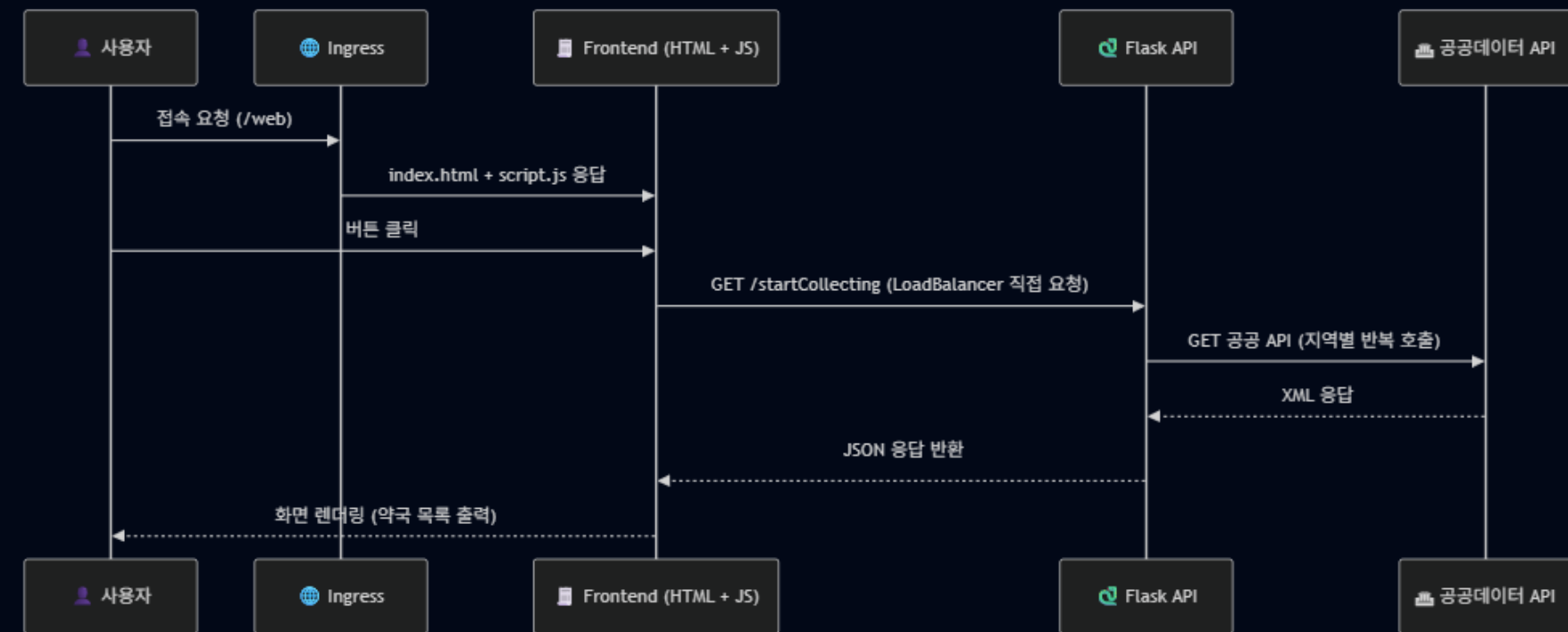
구성 요소 / 함수	역할 설명
collectPharmacyData()	공공데이터 API 요청 → XML 파싱 → 리스트 누적 → DataFrame 반환
/startCollecting	Flask 라우팅 함수 → collectPharmacyData() 호출 후 JSON 응답
script.js	버튼 클릭 시 Ajax로 /startCollecting 호출, JSON 응답을 HTML로 렌더링

05

시퀀스 다이어그램

흐름 요약

1. 사용자가 브라우저에서 /web 접속 → HTML + JS 로드 (Ingress 경유)
2. 사용자가 버튼 클릭 → JavaScript가 `fetch("/startCollecting")` 실행
3. MetalLB 외부 IP를 통해 Flask API에 직접 요청 (LoadBalancer 타입)
4. Flask에서 공공데이터 API에 요청 → 응답 수신 → JSON 가공
5. JSON 응답을 프론트에 반환 → 사용자에게 결과 출력



06

테스트 결과 요약표

테스트 항목	테스트 내용	요청 URL	결과
프론트엔드 접속	브라우저에서 HTML 페이지 접속	http://192.168.56.61/web	☑ 정상
API 직접 호출	수집된 약국 정보를 JSON으로 응답	http://192.168.56.240/startCollecting	☑ 정상
CORS 확인	JS에서 API 호출 시	내부 fetch()	☑ 정상
한글 출력 확인	JSON 응답 내 한글 주소, 이름	API 응답 결과	☑ 정상

06-1

프론트 페이지

캡처 대상 : <http://192.168.56.61/web> 접속
설명 : 버튼 + 결과 출력 영역

서울시 약국 목록

약국 정보 불러오기

[강서구] 21C세계로약국 - 서울특별시 강서구 화곡로 173 , 1층

[강서구] 365그린약국 - 서울특별시 강서구 강서로 43, B104호 (화곡동, 도양라비앙타워)

[강서구] 365샘약국 - 서울특별시 강서구 양천로 424, 가양역 데시앙플렉스 지식산업센터 131호 (등촌동)

[강서구] 365열린약국 - 서울특별시 강서구 공항대로41길 52, 버들빌딩 1층 (등촌동)

[강서구] 365정약국 - 서울특별시 강서구 양천로 570, NH서울축산농협 NH서울타워 1층 115호 (등촌동)

[강서구] 가고싶은 서우약국 - 서울특별시 강서구 양천로 624, 주산빌딩 103호 (등촌동)

[강서구] 가고싶은이화사랑약국 - 서울특별시 강서구 공항대로 지하267, 발산역(5호선) 515-106호 (마곡동)

[강서구] 가까운약국 - 서울특별시 강서구 공항대로 426 (화곡동)

[강서구] 가까운중앙약국 - 서울특별시 강서구 마곡중앙4로 74, 106~107호 (마곡동)

[강서구] 가까운천사약국 - 서울특별시 강서구 공항대로36길 9, 1층 (내발산동)

[강서구] 가양누리약국 - 서울특별시 강서구 허준로 198, 가양프라자 1층 104, 105호 (가양동)

[강서구] 가양메디약국 - 서울특별시 강서구 양천로 490, 1층 101호 (등촌동)

[강서구] 가양메디칼약국 - 서울특별시 강서구 양천로57길 13 (가양동)

[강서구] 가양종로약국 - 서울특별시 강서구 양천로 460, 1층 (등촌동)

[강서구] 가양플러스약국 - 서울특별시 강서구 양천로 431, 홈플러스가양점 B2층 X3,YF호 (가양동)

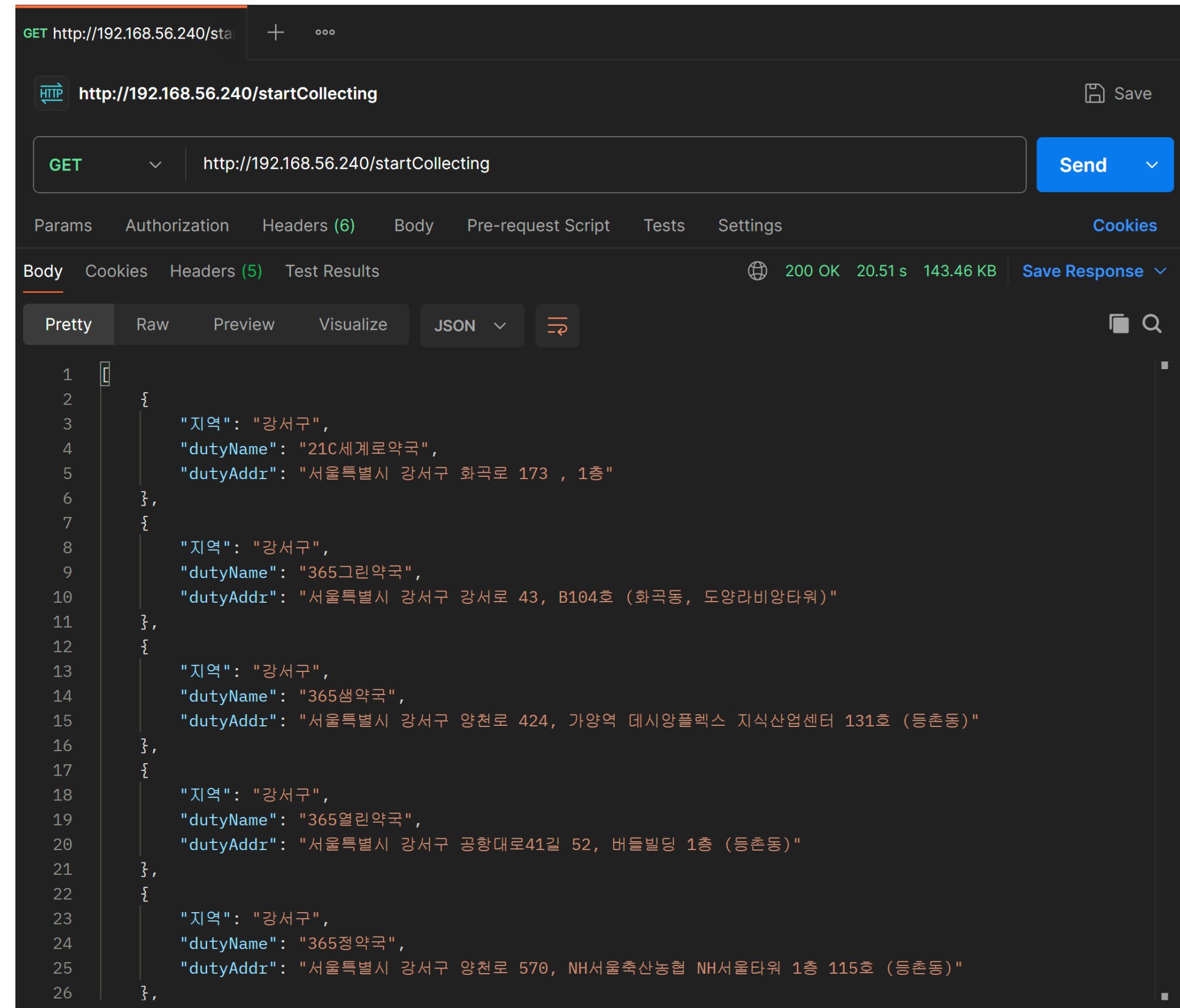
[강서구] 가양하늘약국 - 서울특별시 강서구 화곡로68길 3, 1층 105호 (등촌동)

06-2

API 응답 (Postman)

캡처 대상 : <http://192.168.56.240/startCollecting>

설명 : JSON 응답 성공 화면



06-3

curl 테스트

캡처 대상 : curl

http://192.168.56.240/startCollecting

설명 : 텍스트 JSON 출력

```
[ root@curlpod:/ ]$ curl http://192.168.56.240/startCollecting

[
  {
    "지역": "강서구",
    "dutyName": "21C세계로약국",
    "dutyAddr": "서울특별시 강서구 화곡로 173 , 1층"
  },
  {
    "지역": "강서구",
    "dutyName": "365그린약국",
    "dutyAddr": "서울특별시 강서구 강서로 43, B104호 (화곡동, 도양라비앙타워)"
  },
  {
    "지역": "강서구",
    "dutyName": "365샘약국",
    "dutyAddr": "서울특별시 강서구 양천로 424, 가양역 데시앙플렉스 지식산업센터 131호 (등촌동)"
  },
  {
    "지역": "강서구",
    "dutyName": "365열린약국",
    "dutyAddr": "서울특별시 강서구 공항대로41길 52, 버들빌딩 1층 (등촌동)"
  },
  {
    "지역": "강서구",
    "dutyName": "365정약국",
    "dutyAddr": "서울특별시 강서구 양천로 570, NH서울축산농협 NH서울타워 1층 115호 (등촌동)"
  },
  {
    "지역": "강서구",
    "dutyName": "가고싶은 서우약국",
    "dutyAddr": "서울특별시 강서구 양천로 624, 주산빌딩 103호 (등촌동)"
  },
  {
  }
```

06-4

외부 IP 확인,

라우팅 구조 확인

캡처 대상 : kubectl get svc ,

kubectl get ingress

설명 : 192.168.56.240 할당 확인 ,
/web 경로만 설정 확인

```
kopo@master:~/pharmacy-api-docker-k8s$ kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
frontend-service	ClusterIP	10.97.243.137	<none>	80/TCP	39h
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	26d
pharmacy-service	LoadBalancer	10.110.231.228	192.168.56.240	80:31219/TCP	34h

```
kopo@master:~/pharmacy-api-docker-k8s$ kubectl get ingress
```

NAME	CLASS	HOSTS	ADDRESS	PORTS	AGE
my-nginx	nginx	my-nginx.192.168.56.61.sslip.io	192.168.56.61	80	3d12h
pharmacy-ingress	nginx	*	192.168.56.61	80	34h

07

개발 및 운영 가이드

개발 가이드

항목	내용
언어/프레임워크	Python 3.9 + Flask
외부 연동	공공데이터 포털 API (XML 응답)
주요 기능	약국 이름/주소 수집 및 JSON 응답
보안 처리	.env에 API 키 보관, Flask에서 dotenv로 불러옴
의존성 관리	requirements.txt 사용 (Flask, requests, bs4, pandas)
프론트엔드	HTML + JS(jQuery)로 버튼 클릭 → Ajax 호출 → 결과 표시

07

개발 및 운영 가이드

Docker 운영 가이드

항목	내용
백엔드 이미지	banghyunmin1999/pharmacy-api:latest
프론트엔드 이미지	banghyunmin1999/pharmacy-frontend:latest
Dockerfile 관리	백엔드: backend.Dockerfile, 프론트: frontend.Dockerfile
.env 사용	COPY .env 또는 K8s Secret 활용
빌드 명령어	docker build -t → docker push
재배포 시	이미지 다시 빌드하고, kubectl rollout restart deployment 수행

07

개발 및 운영 가이드

Kubernetes 배포 가이드

항목	설명
구성 요소	Deployment, Service, Ingress, MetalLB
프론트 배포	/web 경로 → Ingress 경유 (frontend-service)
백엔드 배포	LoadBalancer IP → 직접 API 연결 (pharmacy-service)
외부 노출	192.168.56.61/web, 192.168.56.240/startCollecting
ingress.yaml	/web 경로만 설정 (추후 /api 확장 가능)
MetalLB	IP 풀 설정 필요 (192.168.56.240 ~ ...)

07

개발 및 운영 가이드

테스트 및 모니터링

항목	내용
API 테스트	curl, Postman으로 확인
상태 확인	kubectl get pods, kubectl logs, kubectl get svc
에러 분석	Flask 로그 (kubectl logs deploy/pharmacy-api)
리소스 변경	kubectl apply -f ..., kubectl rollout restart

THANK YOU
