

# YC Tech

## 웹 백엔드 실무 개발 프로젝트

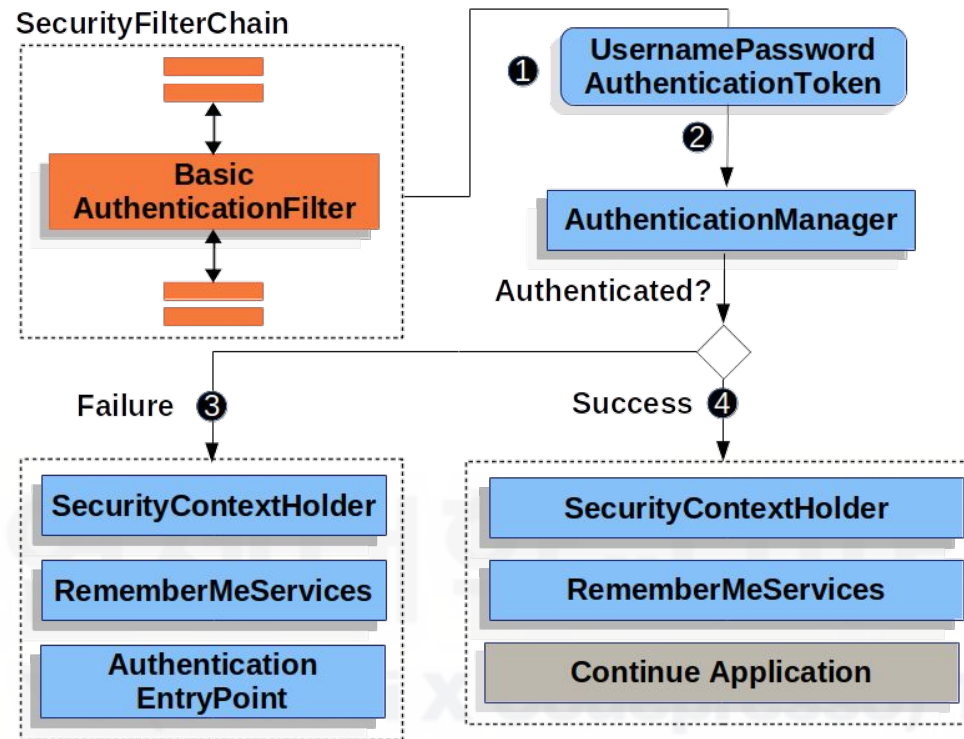


연세대학교 미래교육원  
YC (Yonsei X Codepresso) Tech Academy

# 4주차 recap



# Spring security

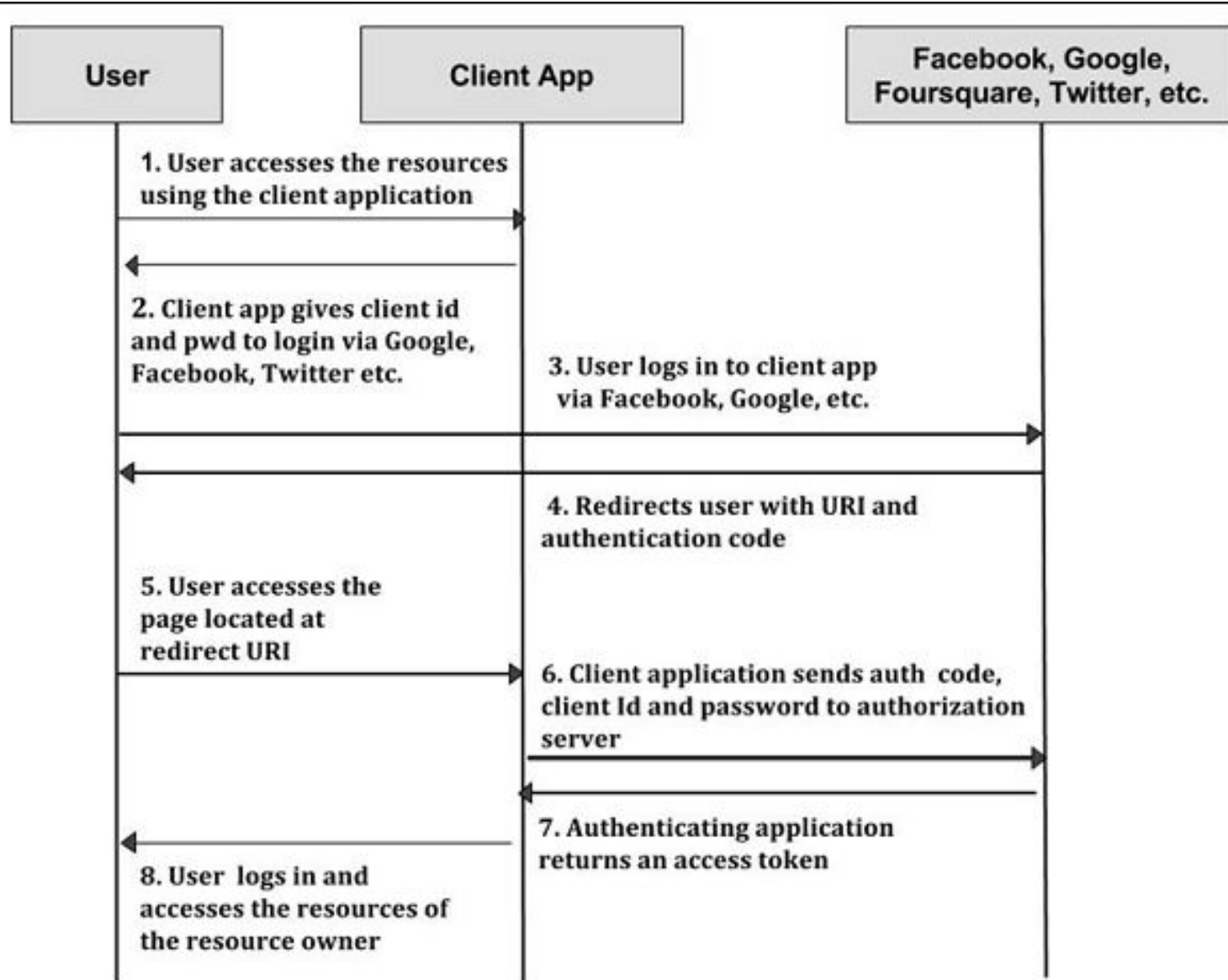


1. username 과 password 를 이용하여 token 생성
2. AuthenticationManager 에서 인증 여부 확인
3. 실패하는 경우 SecurityContextHolder 를 비우고 인증 실패처리
4. 성공하는 경우 인증 filter 통과

# Oauth 2.0



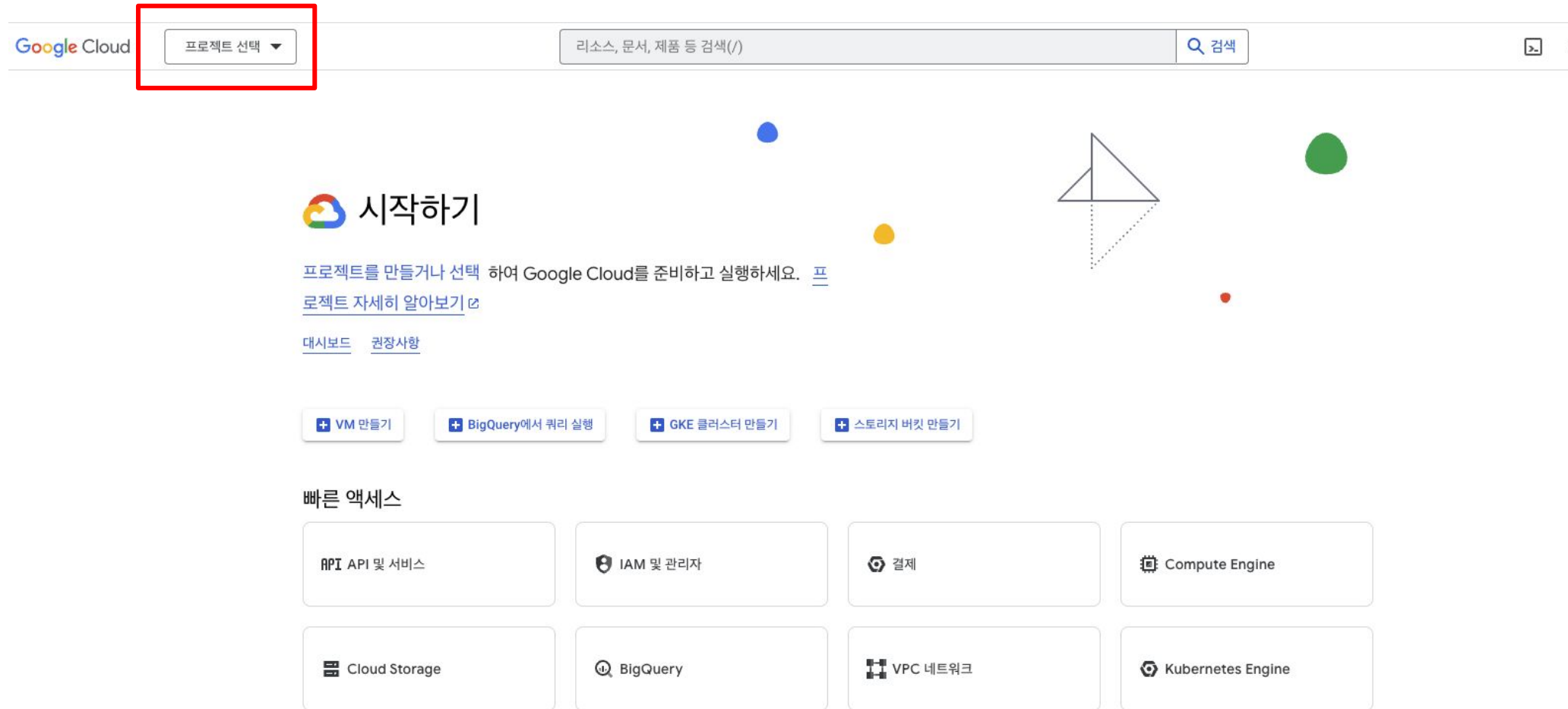
# OAuth2.0 인증 flow



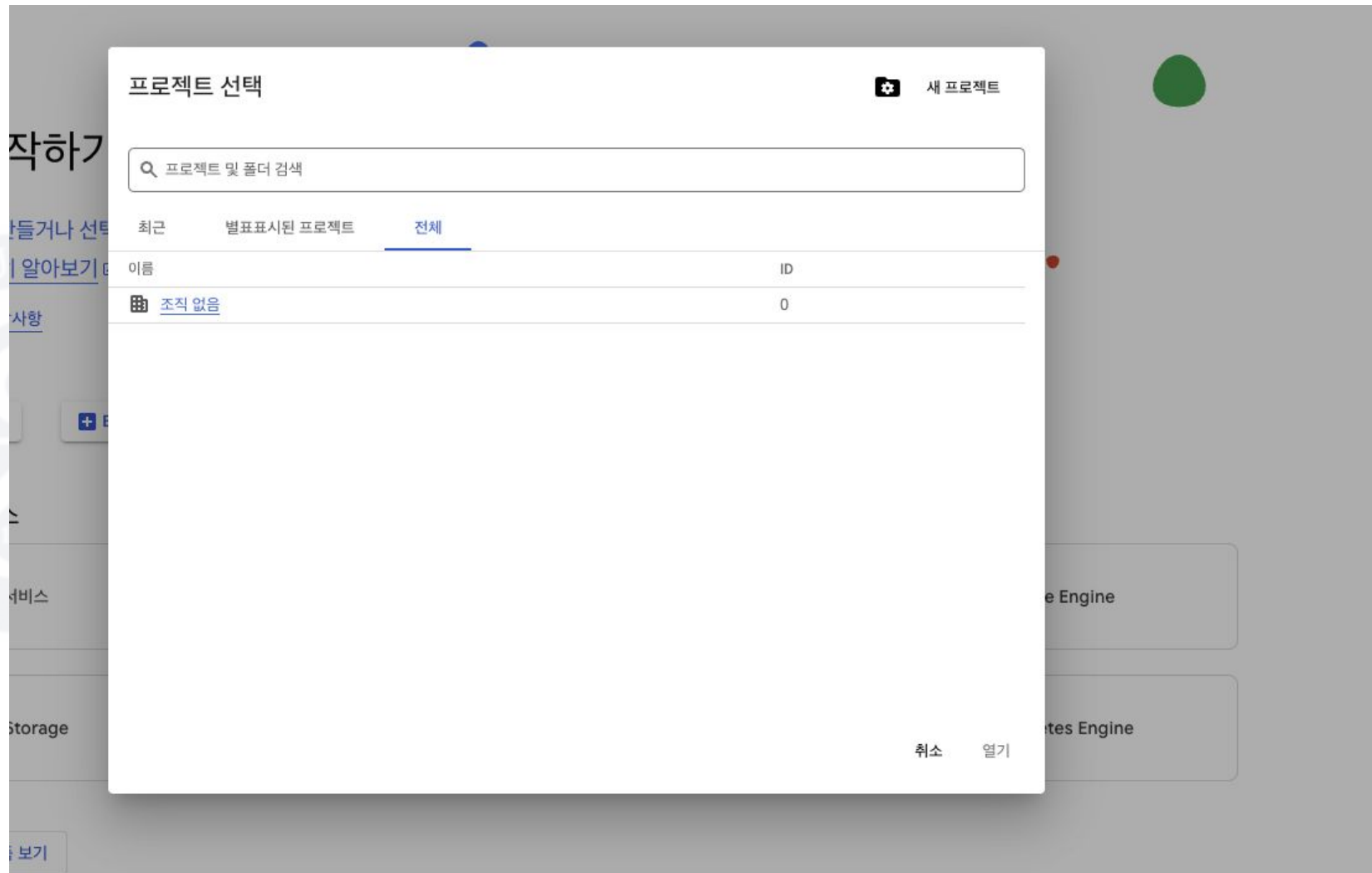
## OAuth2.0

- Third-Party 프로그램에게 리소스 소유자를 대신하여 리소스 서버에서 제공하는 자원에 대한 접근 권한을 위임하는 방식을 제공
- 구글, 페이스북, 카카오, 네이버 등에서 제공하는 간편 로그인 기능도 OAuth2 프로토콜 기반의 사용자 인증 기능을 제공

<https://console.cloud.google.com/>



## ‘새 프로젝트’



## 새 프로젝트



projects 할당량이 12개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)

[MANAGE QUOTAS](#)

프로젝트 이름 \*

TestApp



프로젝트 ID: testapp-387812입니다. 나중에 변경할 수 없습니다. [수정](#)

위치 \*

조직 없음

[찾아보기](#)

상위 조직 또는 폴더

만들기

취소

육원  
academy



빠른 액세스

**API** API 및 서비스



미래교육원  
YC (Yonsei X Codepresso) Tech Academy

Google Cloud

TestApp

리소스, 문서, 제품 등 검색(/)

API API 및 서비스

사용 설정된 API 및 서비스

라이브러리

사용자 인증 정보

OAuth 동의 화면

페이지 사용 동의

사용자 인증 정보

+ 사용자 인증 정보 만들기

삭제

삭제된 사용자 인증 정보 복원

사용 설정한 API에 액세스하려면 사용자 인증 정보를 만드세요. [자세히 알아보기](#)

⚠

애플리케이션에 대한 정보를 포함하여 OAuth 동의 화면을 구성해야 합니다.

API 키

☐

이름

생성일 ↓

표시할 API 키가 없습니다.

OAuth 2.0 클라이언트 ID

☐

이름

생성일 ↓

표시할 OAuth 클라이언트가 없습니다.

서비스 계정

☐

이메일

표시할 서비스 계정이 없습니다.

## ← OAuth 클라이언트 ID 만들기

클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 [OAuth 2.0 설정](#)을 참조하세요. OAuth 클라이언트 유형을 [자세히 알아보세요](#).



OAuth 클라이언트 ID를 만들려면 먼저 동의 화면을 구성해야 합니다.

동의 화면 구성

## OAuth 동의 화면

대상 사용자를 비롯해 앱을 구성하고 등록하려는 방식을 선택하세요. 프로젝트에는 하나의 앱만 연결할 수 있습니다.

### User Type

☐ 내부 ?

조직 내 사용자만 사용할 수 있습니다. 인증을 위해 앱을 제출할 필요는 없습니다. [사용자 유형 자세히 알아보기](#)

☒ 외부 ?

Google 계정이 있는 모든 테스트 사용자가 사용할 수 있습니다. 앱이 테스트 모드로 시작되고 테스트 사용자 목록에 추가된 사용자에게만 제공됩니다. 앱을 프로덕션에 출시할 준비가 되면 앱을 인증해야 할 수도 있습니다. [사용자 유형 자세히 알아보기](#)

만들기

OAuth 경험에 대한 [의견 보내기](#)

## 범위 추가 삭제

×

선택한 범위 업데이트

i

아래에는 사용 설정된 API의 범위만 나와 있습니다. 이 화면에 누락된 범위를 추가하려면 [Google API 라이브러리](#)에서 API를 찾아 사용 설정하거나 아래의 '붙여넣은 범위' 텍스트 상자를 사용하세요. 라이브러리에서 사용 설정한 새 API를 확인하려면 페이지를 새로고침하세요.

≡

필터

속성 이름 또는 값 입력

?

<input type="checkbox"/>	API ↑	범위	사용자에게 표시되는 설명
<input checked="" type="checkbox"/>		.../auth/userinfo.email	기본 Google 계정의 이메일 주소 확인
<input checked="" type="checkbox"/>		.../auth/userinfo.profile	개인정보(공개로 설정한 개인정보 포함) 보기
<input type="checkbox"/>		openid	Google에서 내 개인 정보를 나와 연결
<input type="checkbox"/>	BigQuery API	.../auth/bigquery	View and manage your data in Google BigQuery and see the email address for your Google Account
<input type="checkbox"/>	BigQuery	.../auth/cloud-platform	Google Cloud 데이터 확인, 수정, 구성, 삭제 및 Google 계정의 이

호스트/login/oauth2/code/google 로 설정

## 승인된 리디렉션 URI

웹 서버의 요청에 사용

URI 1 \*

http://localhost:8080/login/oauth2/code/google

+ URI 추가

## 클라이언트 ID, 클라이언트 보안 비밀번호 복사

### Additional information

클라이언트 ID

생성일

2023년 10월 6일 PM 8시 37분 18초 GMT+9

### 클라이언트 보안 비밀번호

클라이언트 보안 비밀번호를 변경하는 중인 경우 다운타임 없이 수동으로 순환할 수 있습니다. [자세히 알아보기](#)

클라이언트 보안 비밀번호

생성일

2023년 10월 6일 PM 8시 37분 18초 GMT+9

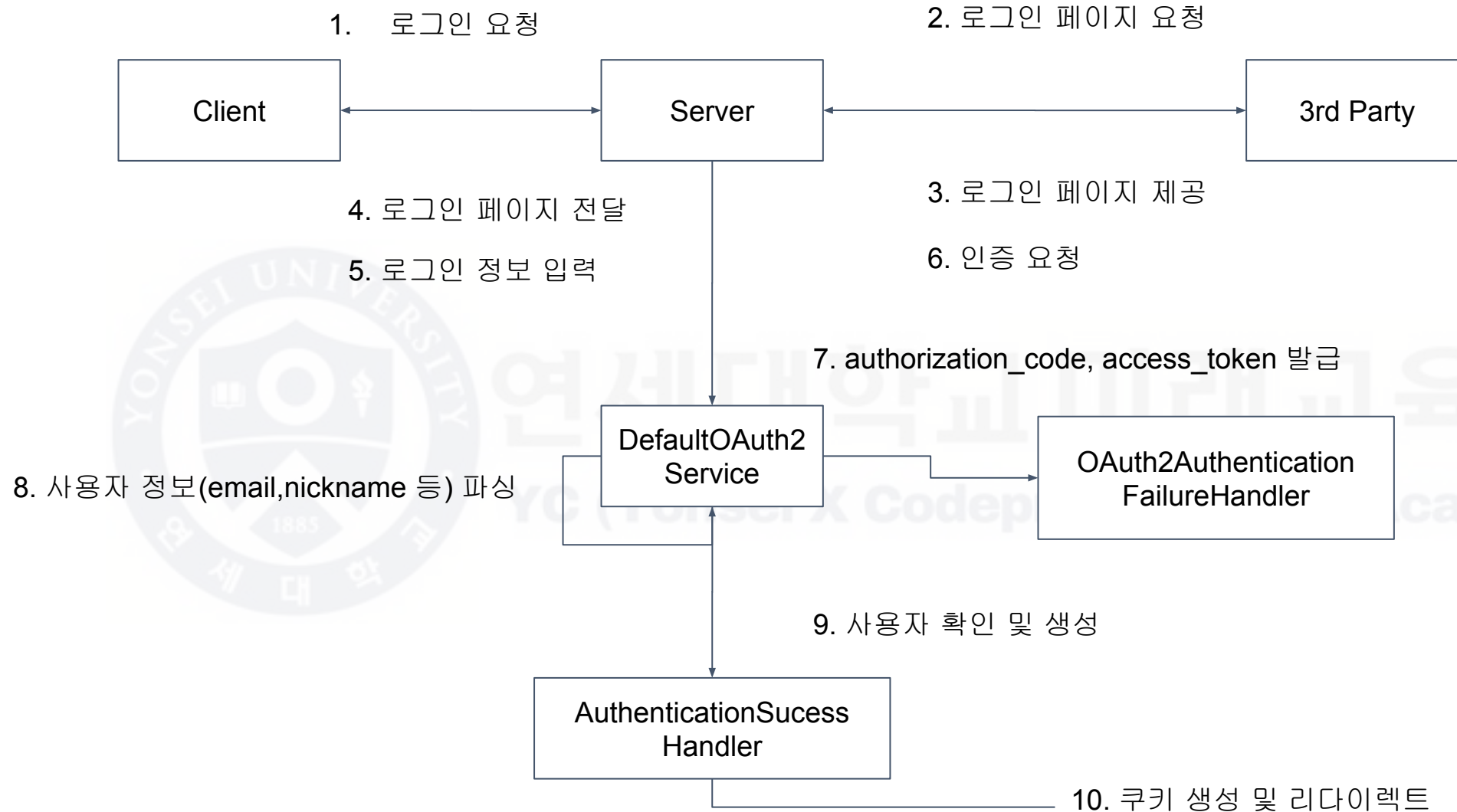
상태

✓ 사용 설정됨

+ ADD SECRET

원  
my

# Spring OAuth2.0 client flow





# Notes

- ❑ Spring security 를 사용하지 않고도 구현 가능
- ❑ 요구사항에 맞춰 security 설정
- ❑ 서비스가 요구하는 보안 수준과 사용자 편의에 맞게 구현

A portrait of Mark Zuckerberg, smiling and looking directly at the camera. He has short brown hair and is wearing a dark-colored t-shirt. The background is a solid dark blue-grey color.

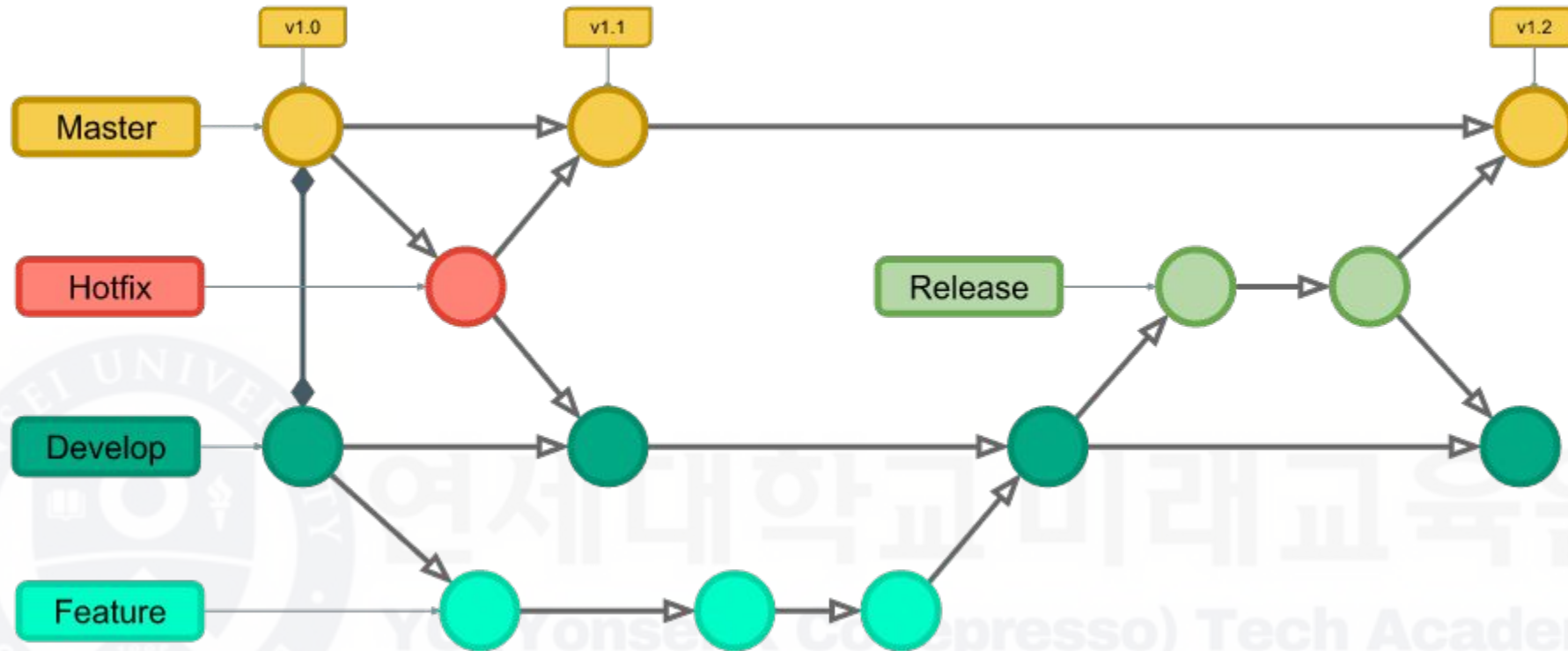
“Done is better  
than perfect.”

- Mark Zuckerberg

# Git/Github

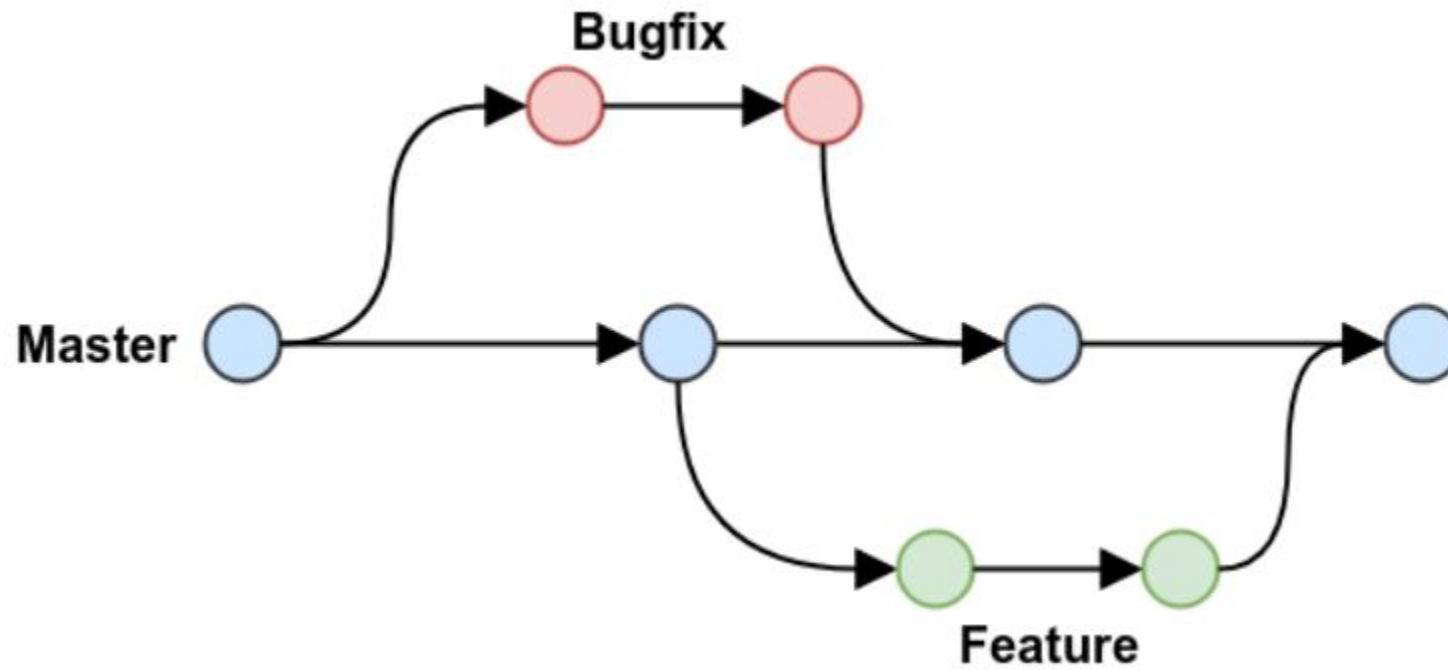


# branch 전략 - git flow



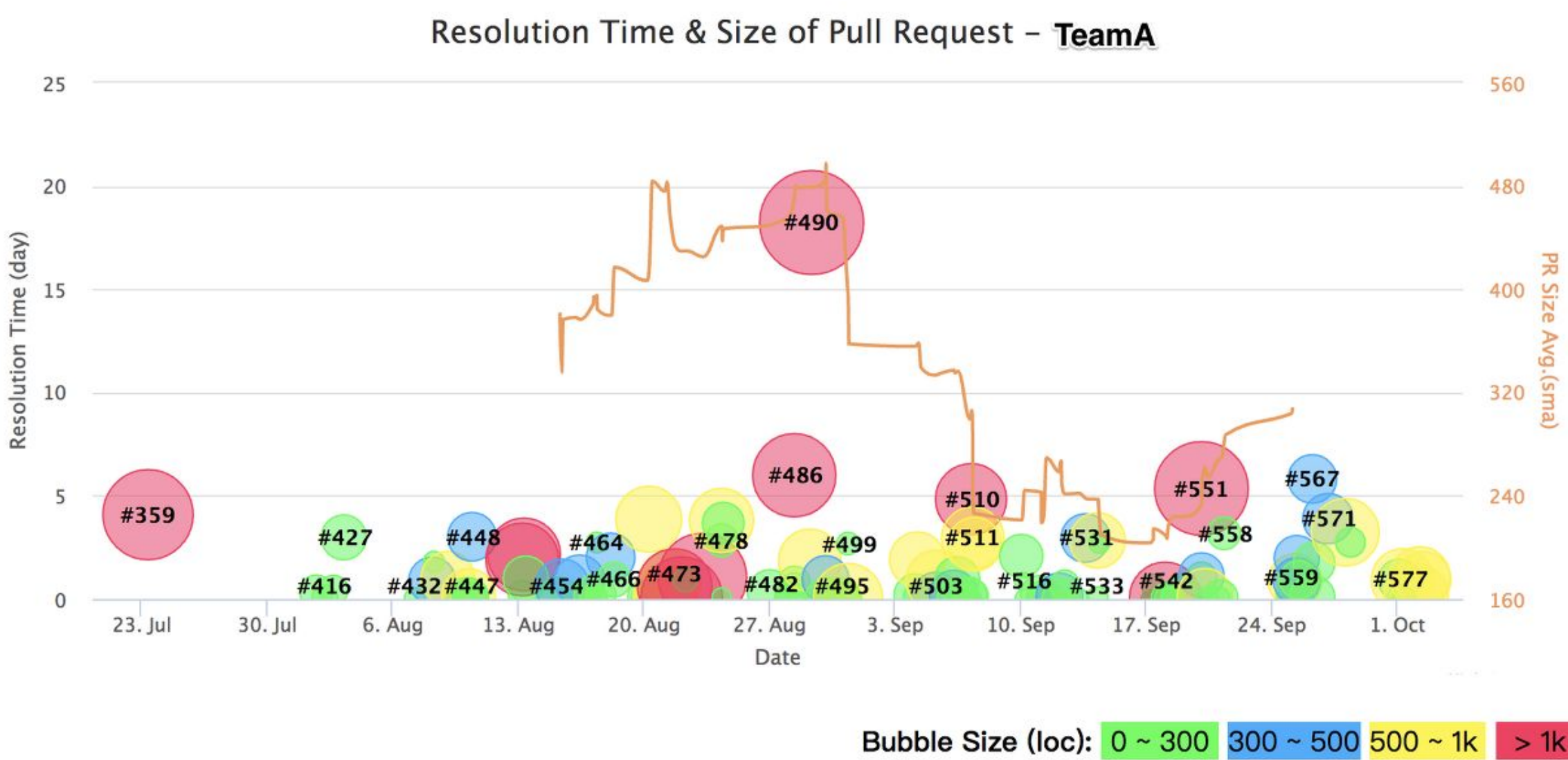
- **master** : 실제로 서비스에 배포되는 브랜치이다.
- **develop** : 새 버전에 대한 개발을 진행하는 브랜치이다.
- **feature** : 개발의 기본 단위인 기능을 추가하기 위한 브랜치이다. (**develop**에서 시작되어 **develop**으로 합쳐진다.)
- **release** : **Develop**에서 새 버전에 대한 개발이 마무리되어 출시하기 위한 브랜치이다. (**develop**에서 시작되어 **master**와 **develop**으로 합쳐진다.)
- **hotfix** : **Master**에서 생긴 버그를 급하게 해결하기 위한 브랜치이다. (**master**에서 시작되어 **master**와 **develop**으로 합쳐진다.)

# branch 전략 - github flow





# 적절한 pull request size



# 어떤 내용을 포함시켜야 하는가

## PTC-2108: Implement sticky header for view page controls

#1031 **MERGED** at fd89951

issue/PTC-2108\_stickyhe...



master

Revert

Approve

1

Overview

Commits

Activity

Author



Blake Riosa

Reviewers



Description

**Problem:** When scrolling down on view page, if you want to edit, share, or watch the page, you have to scroll all the way up to click the buttons. Solution: sticky header that follows you as you scroll.

### Key features:

- Should only appear when scrolling up
- Should disappear when scrolling down
- Should hide inline dialogs
- Should resize when sidebar resizes

1 of 1 passed



PTC-2108

Stop watching

There are no dependencies

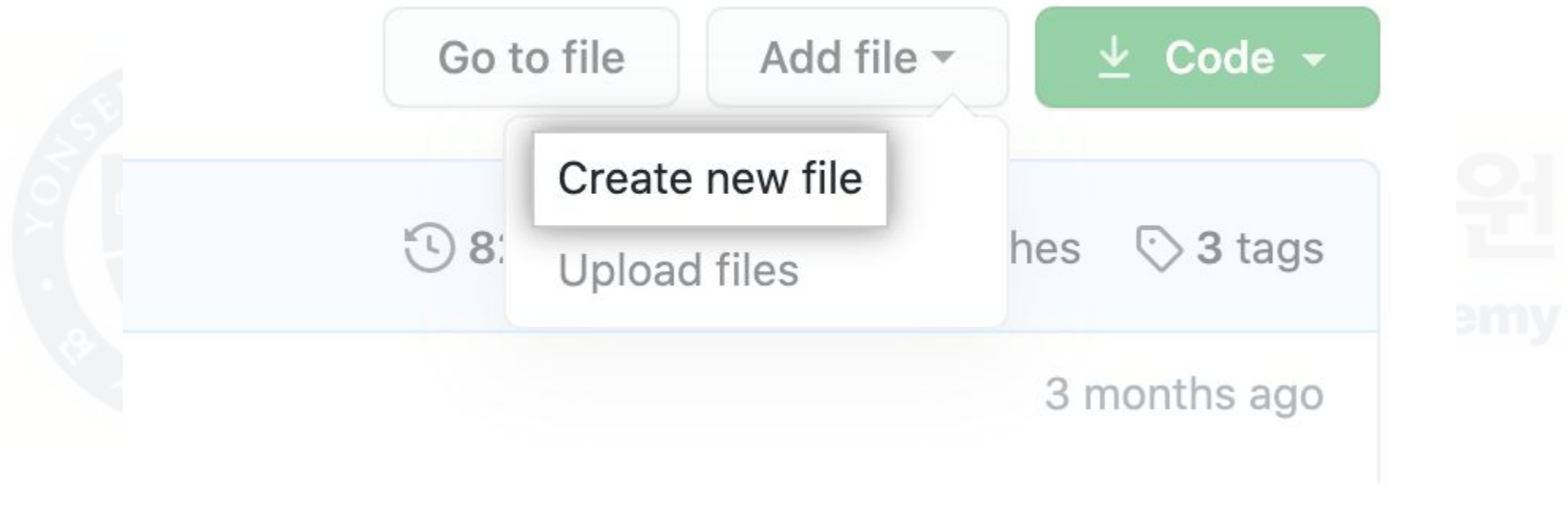
There are no dependents

Pull request guidelines

Deploy to AWS

1. 무슨 이유로 코드를 변경했는지
2. 어떤 위험이나 장애가 발견되었는지
3. 어떤 부분에 리뷰어가 집중하면 좋을지
4. 관련 스크린샷 테스트 계획 또는 완료 사항

# template 설정





# template 설정

/ .github / pull\_request\_template.md or cancel



<https://docs.github.com/ko/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>

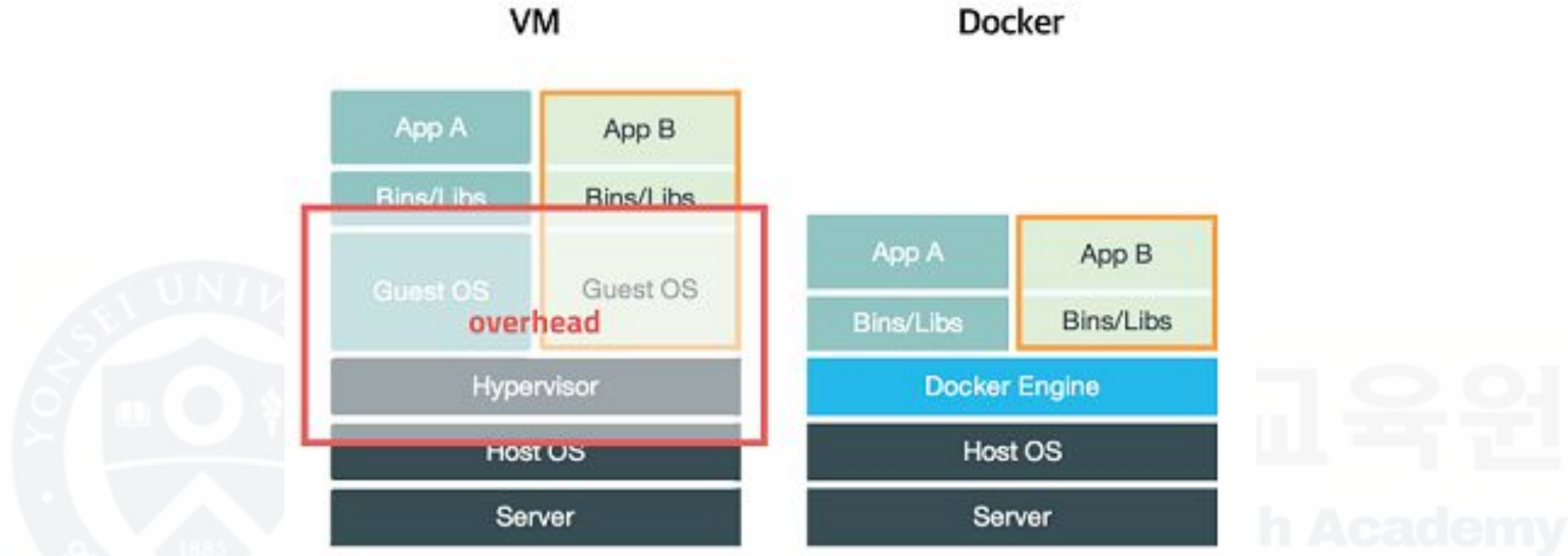
<https://docs.github.com/ko/actions>

- 소프트웨어 **workflow**를 자동화할 수 있도록 도와주는 도구
  - test
  - CI/CD
  - 자동화 스크립트
- **Workflow** : 여러 **Job**으로 구성되고, **Event**에 의해 트리거될 수 있는 자동화된 프로세스 최상위 개념 **Workflow** 파일은 **YAML**으로 작성되고, **Github Repository**의 **.github/workflows** 폴더 아래에 저장됨
- **Event** : **Workflow**를 **Trigger**(실행)하는 특정 활동이나 규칙
- **Job** : 여러 **Step**으로 구성되고, 가상 환경의 인스턴스에서 실행됨 다른 **Job**에 의존 관계를 가질 수 있고, 독립적으로 병렬 실행도 가능함
- **Step** : **Task**들의 집합으로, 커맨드를 날리거나 **action**을 실행할 수 있음
- **Action** : **Workflow**의 가장 작은 블록(**smallest portable building block**) **Job**을 만들기 위해 **Step**들을 연결할 수 있음

# Docker

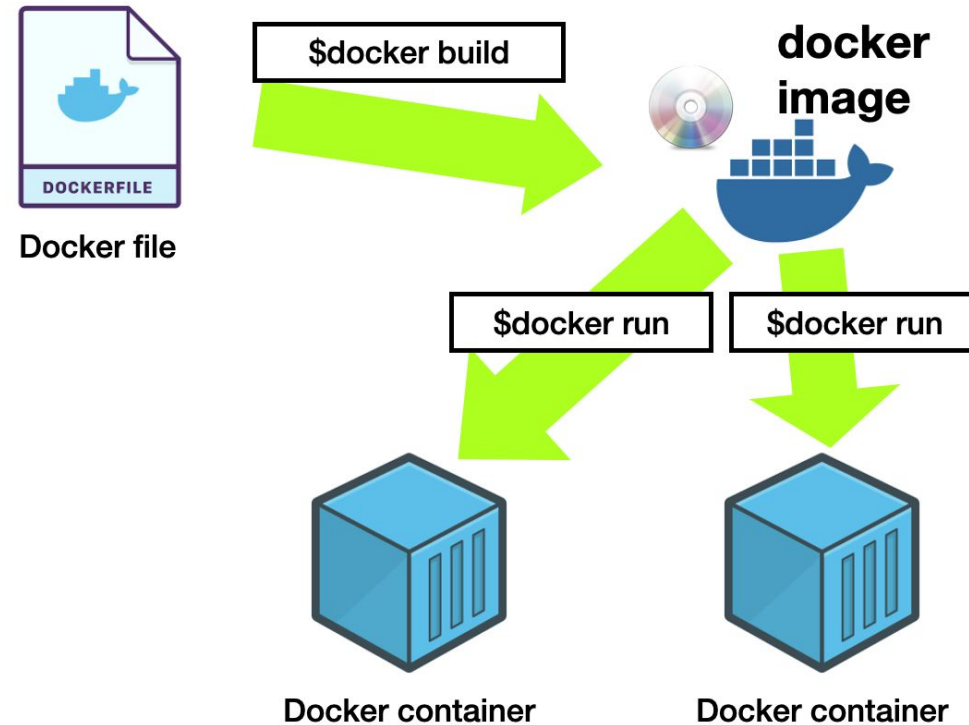


# Container vs Virtual Machine



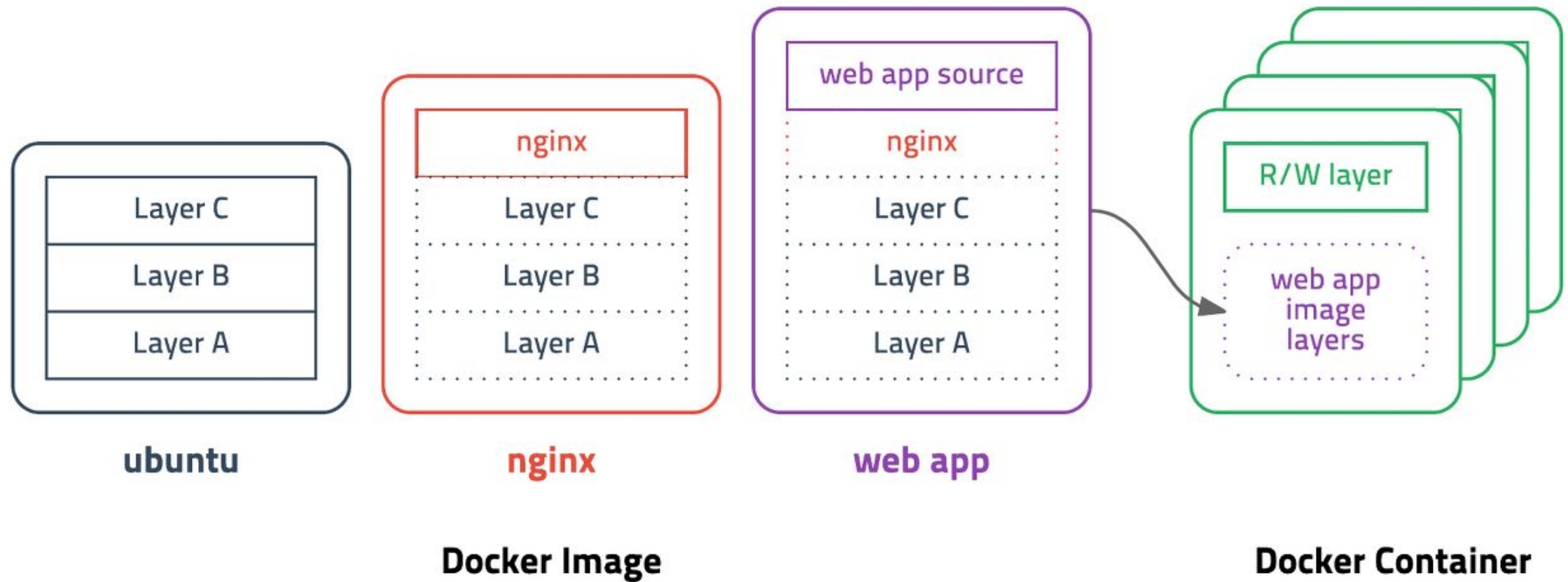
- **VM** : OS 영역도 가상화하여 한 **machine** 내에서도 여러 개의 **OS** 설치 가능
- **Container** : OS 레이어 위에 **container engine** 을 설치하여 OS 를 공유하면서 가상화된 환경 제공

# Docker image, container

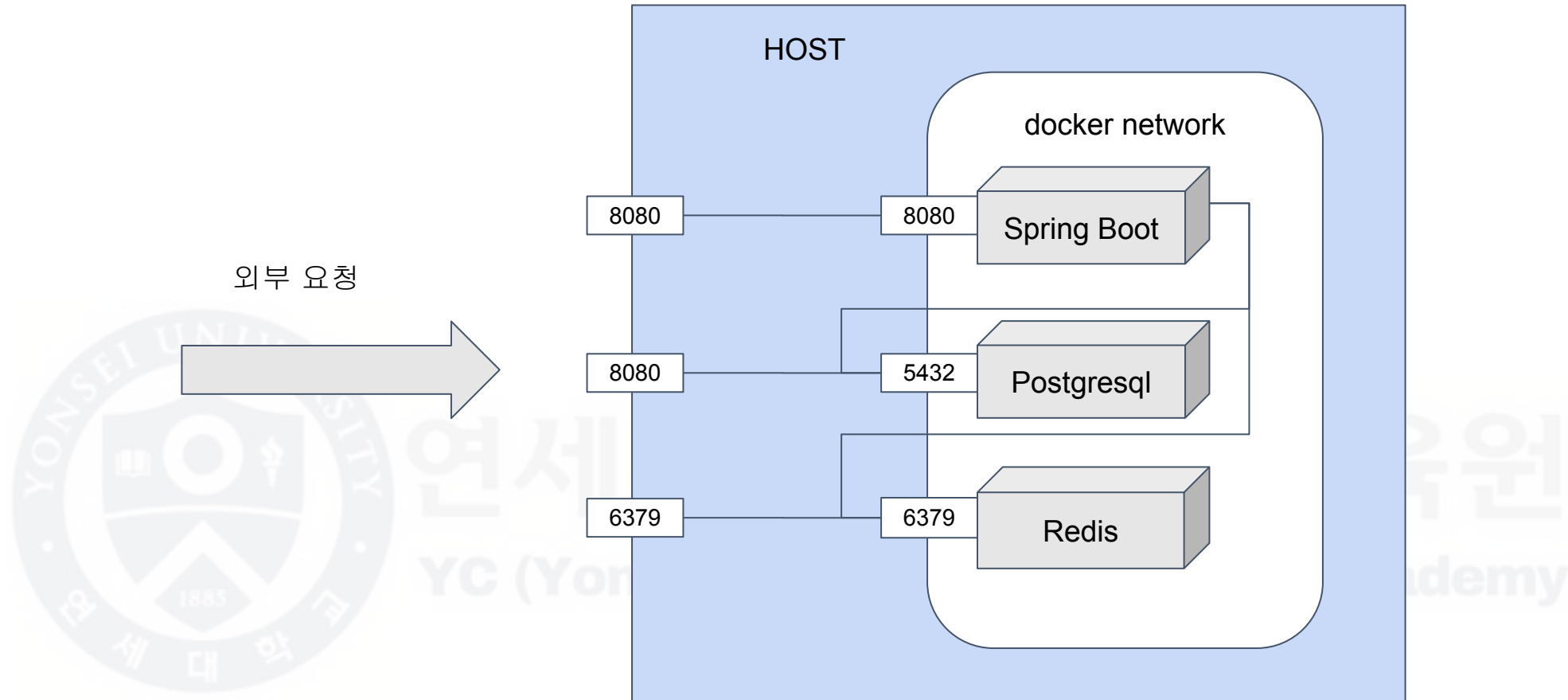


- Dockerfile : docker image 를 생성하기 위한 스크립트
- image : docker : 파일과 설정값 등을 포함한 상태값을 가지고 있는 형상
- container : image 로 부터 instantiate 한 container
  - container 의 내부의 상태들은 container 와 lifecycle 을 함께 함

# Docker image

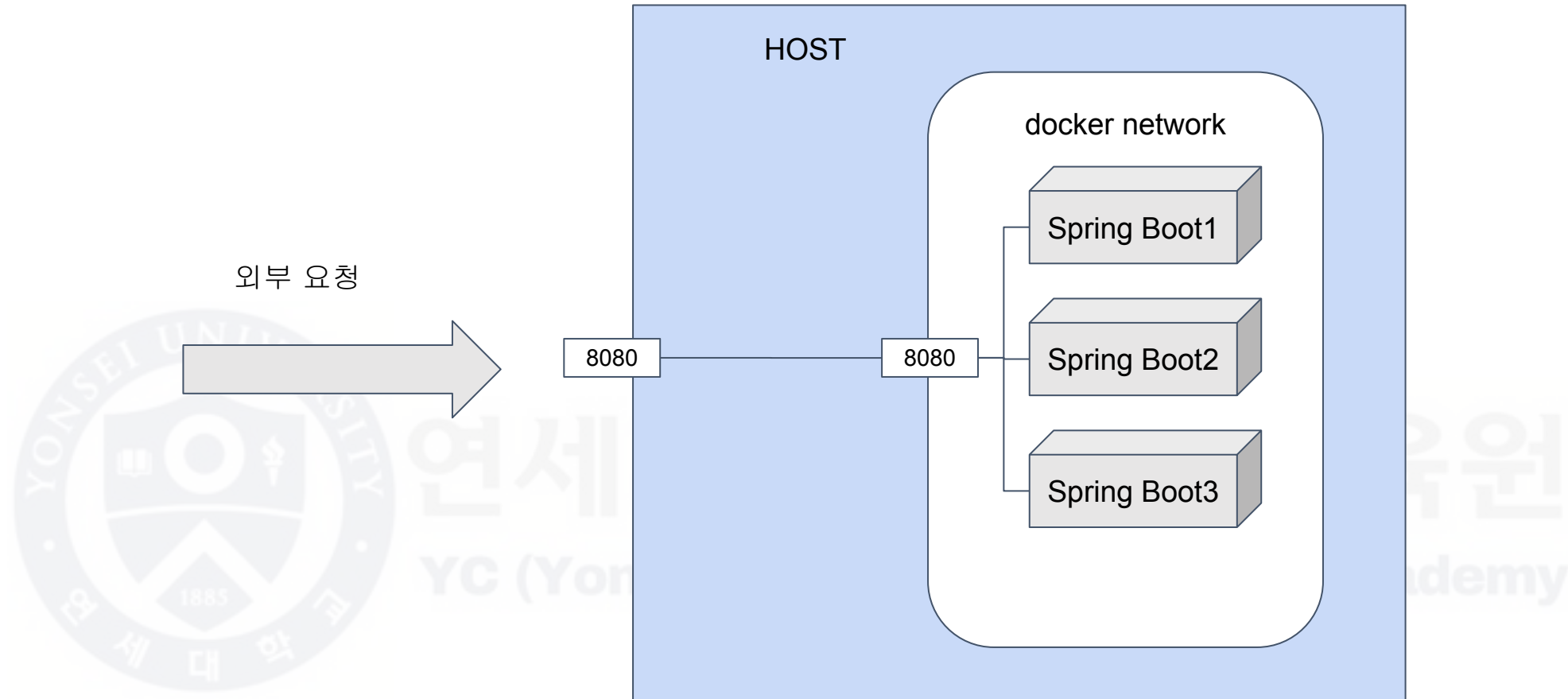


# Docker compose



- multi container 를 한 host 안에 구성
- network 를 생성하여 host와 통신
- 복수 개의 container 를 구동할 수 있음

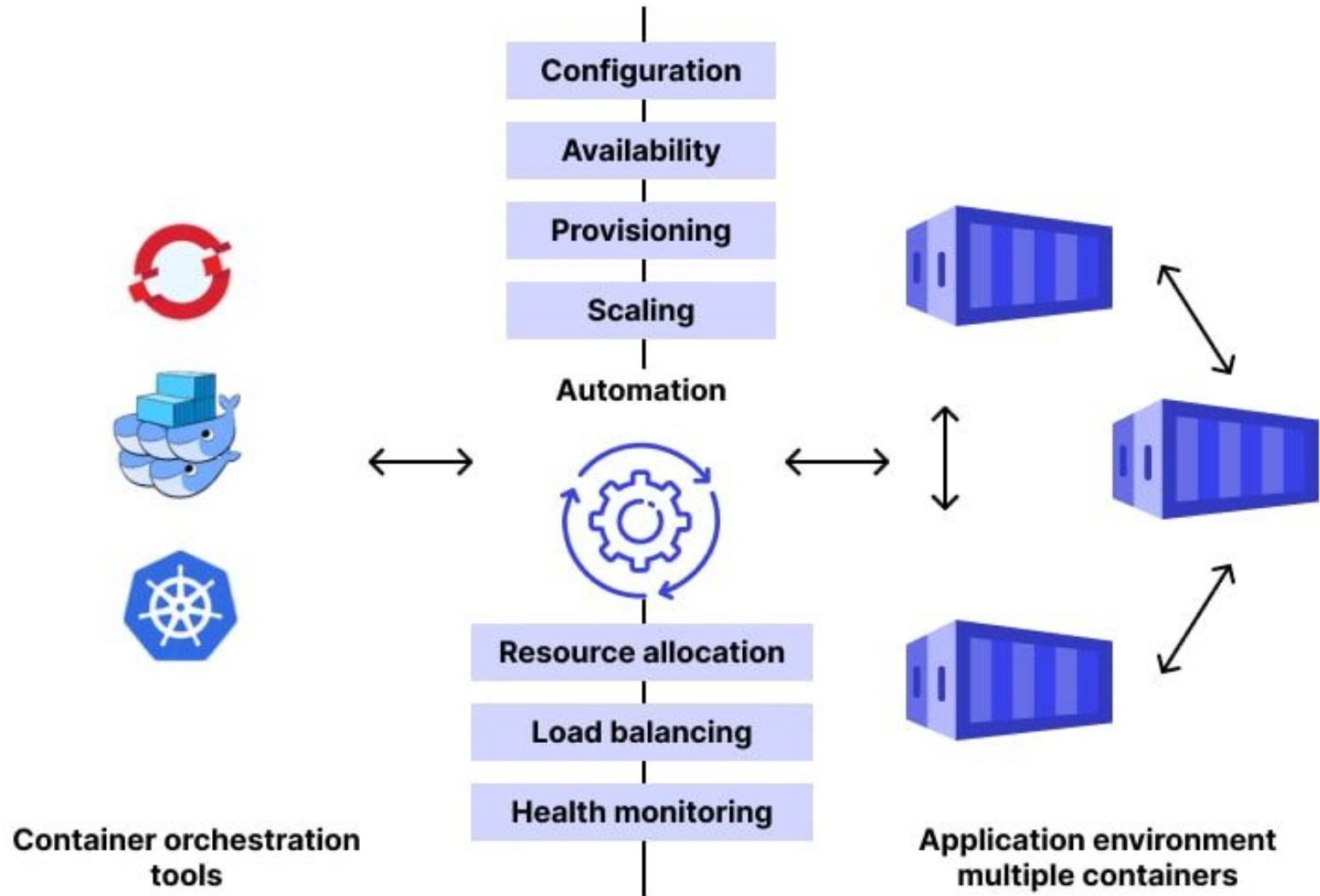
# Docker compose

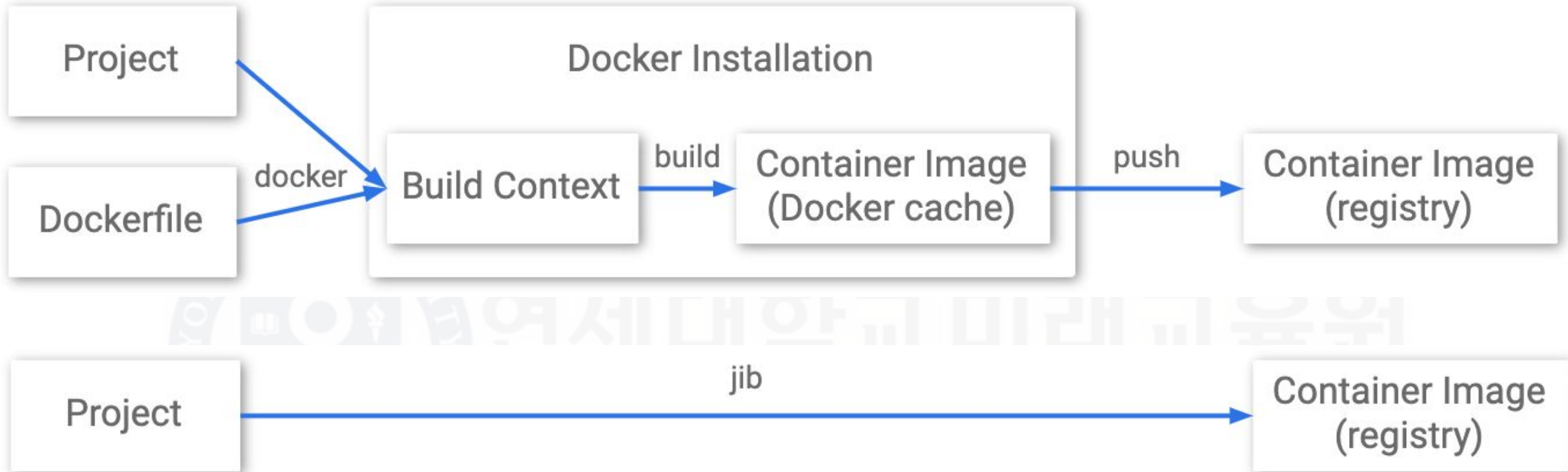


- replication 모드로 한 service 에 복수 개의 container 구동 가능
- round robin 방식으로 호출



# Container Orchestration





Spring project의 향상을 간단하게 docker image 를 만들 수 있는 plugin

- <https://github.com/GoogleContainerTools/jib/tree/master/jib-gradle-plugin>

# Notes

- ❑ Immutable
- ❑ 1 container 1 application
- ❑ keep lightweight
- ❑ no silverbullet

# To DO

- ❑ Spring security OAuth2.0 구현
- ❑ Spring docker image 생성
- ❑ docker compose 로 필요한 컴포넌트 구동

The background is a solid blue color with several abstract geometric elements. In the top-left and bottom-right corners, there are circular halftone patterns of varying densities. Diagonal lines in a lighter shade of blue cross these patterns. On the right side, there are two thin, light-blue circles. The overall design is modern and minimalist.

EOD