

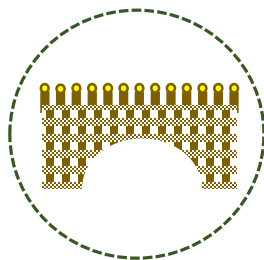


# Journey to the Multi-Cloud

클라우드바리스타 커뮤니티 제3차 컨퍼런스

## CB-Bridge : 원 클릭으로 Cloud-Barista 운용하기

(Cloud-Barista 운용 관리)

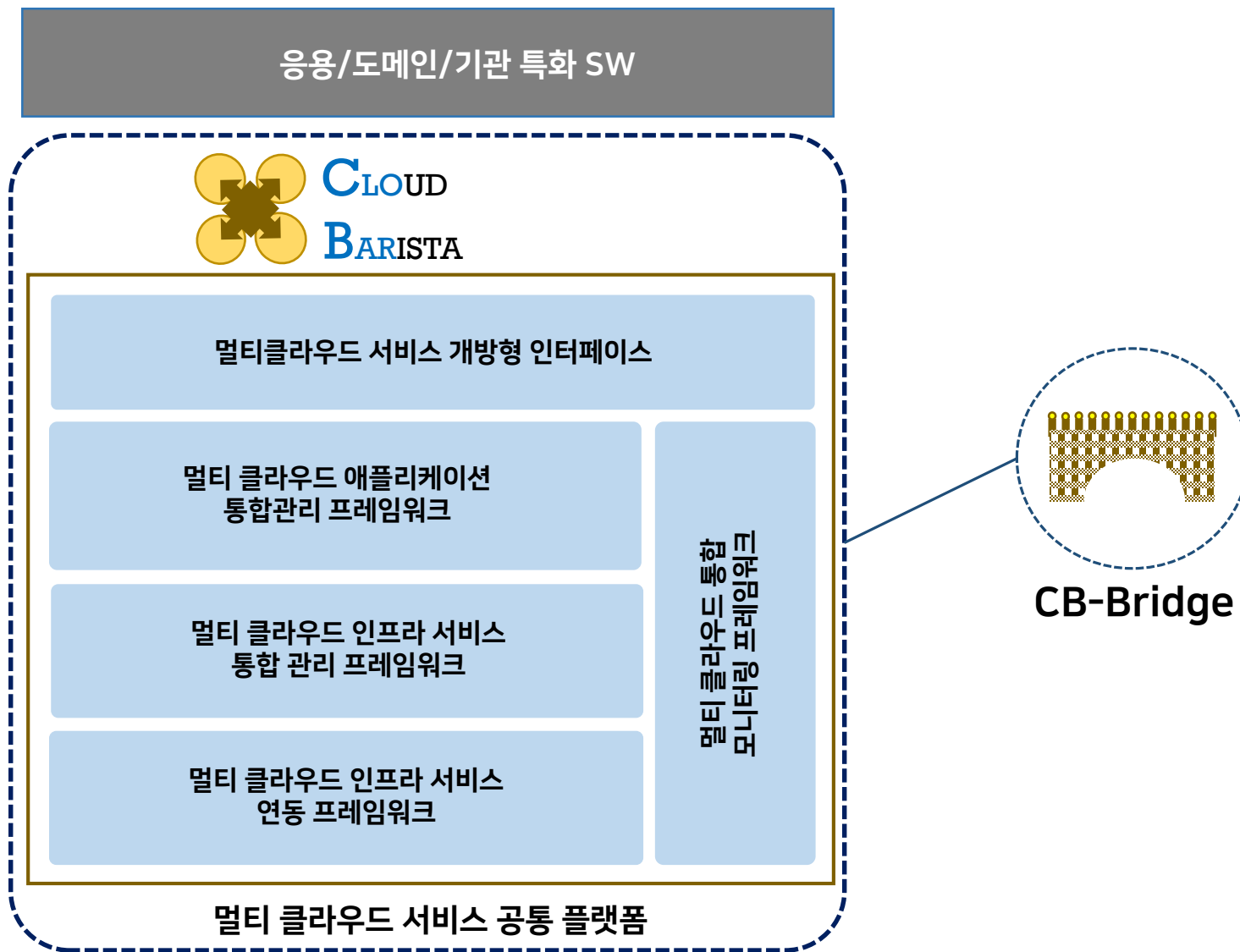


서지훈 / CB-Bridge 프레임워크 리더



에스프레소(Espresso) 한잔 어떠세요 ?

## 이번 세션은...



# 목 차

---

I

CB-Bridge 개요 및 주요 역할

II

Cloud-Barista 설치 및 실행 도구 (cb-operator)

III

Cloud-Barista API Gateway (cb-apigw)

IV

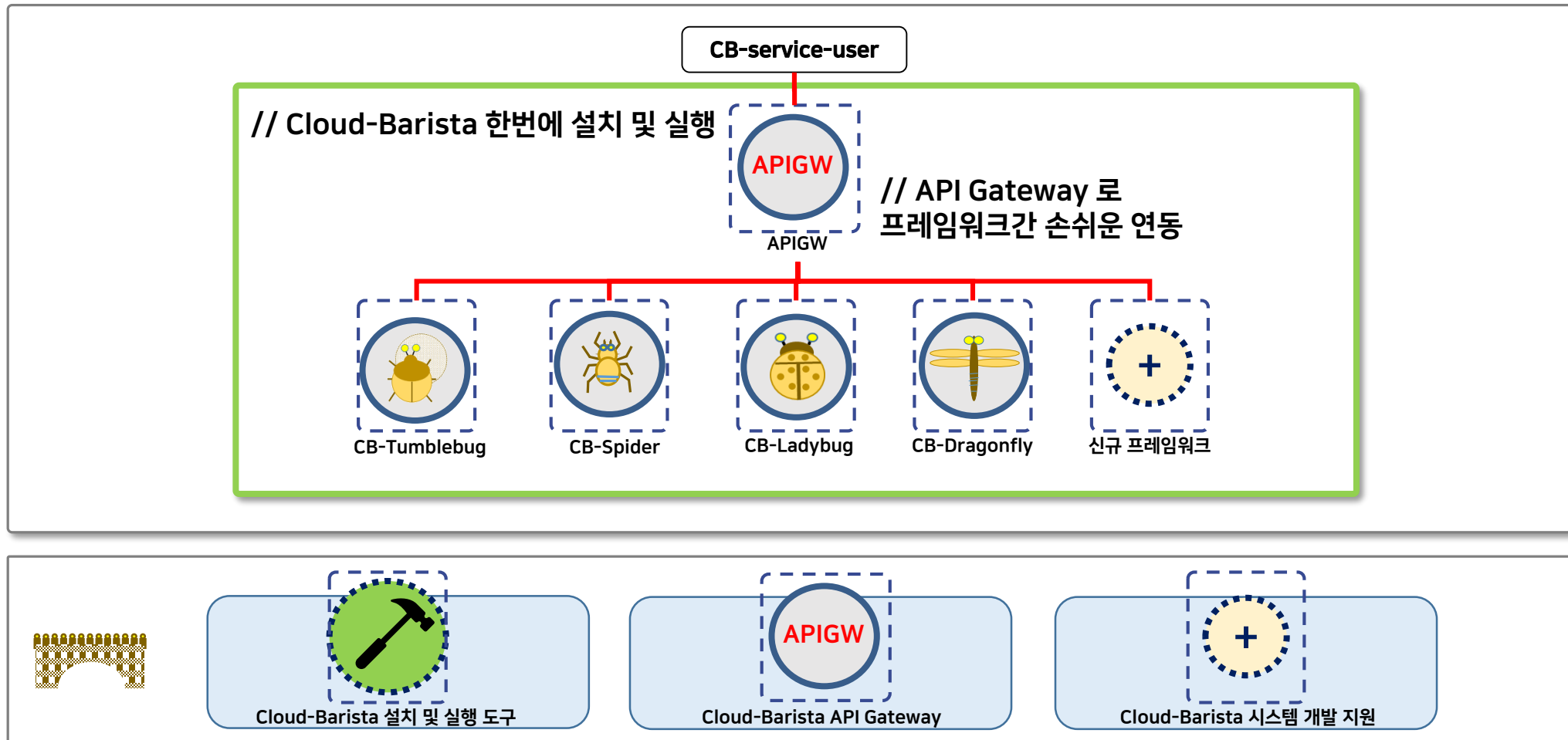
Cloud-Barista 개발에서의 CI (Continuous Integration)

V

CB-Bridge 개발 현황 및 로드맵

# CB-Bridge 개요

- CB-Bridge의 정의 : Cloud-Barista 시스템의 개발, 실행 및 운용 지원 기술 집합





# CB-Bridge 주요 역할

---

- **Cloud-Barista 시스템 설치 및 실행 관리**
  - 정의: 사용자가 손쉽게 Cloud-Barista 시스템을 설치 & 실행할 수 있도록 해 주는 기술
  - 구성
    - Cloud-Barista 설치 및 실행 도구 (cb-operator) (★)
- **Cloud-Barista 시스템 운용 지원**
  - 정의: [Cloud-Barista 설치 및 실행 도구] 를 이용해 실행한 Cloud-Barista 시스템의 운용을 지원하는 기술
  - 구성
    - Cloud-Barista API Gateway (cb-apigw) (★)
    - Cloud-Barista 개별 프레임워크 모니터링 기능
- **Cloud-Barista 시스템 개발 지원**
  - 정의: Cloud-Barista 개발자의 Cloud-Barista 구성요소 개발을 지원하는 기술
  - 구성
    - Cloud-Barista 개발에서의 CI (Continuous Integration) (★)
    - Cloud-Barista 개발 공통 구성요소



# Cloud-Barista 설치 및 실행 도구 (cb-operator)

---

- 개요

- Cloud-Barista 시스템의 실행, 상태정보 제공, 종료 등을 지원하는 관리 도구

- 특징

- Golang 기반의 CLI 도구
- 사용자 명령에 따라 system call을 활용하여 CB 시스템 구동/중지/삭제
- 2가지 모드 (Docker Compose 모드, Kubernetes 모드) 제공

- 기능

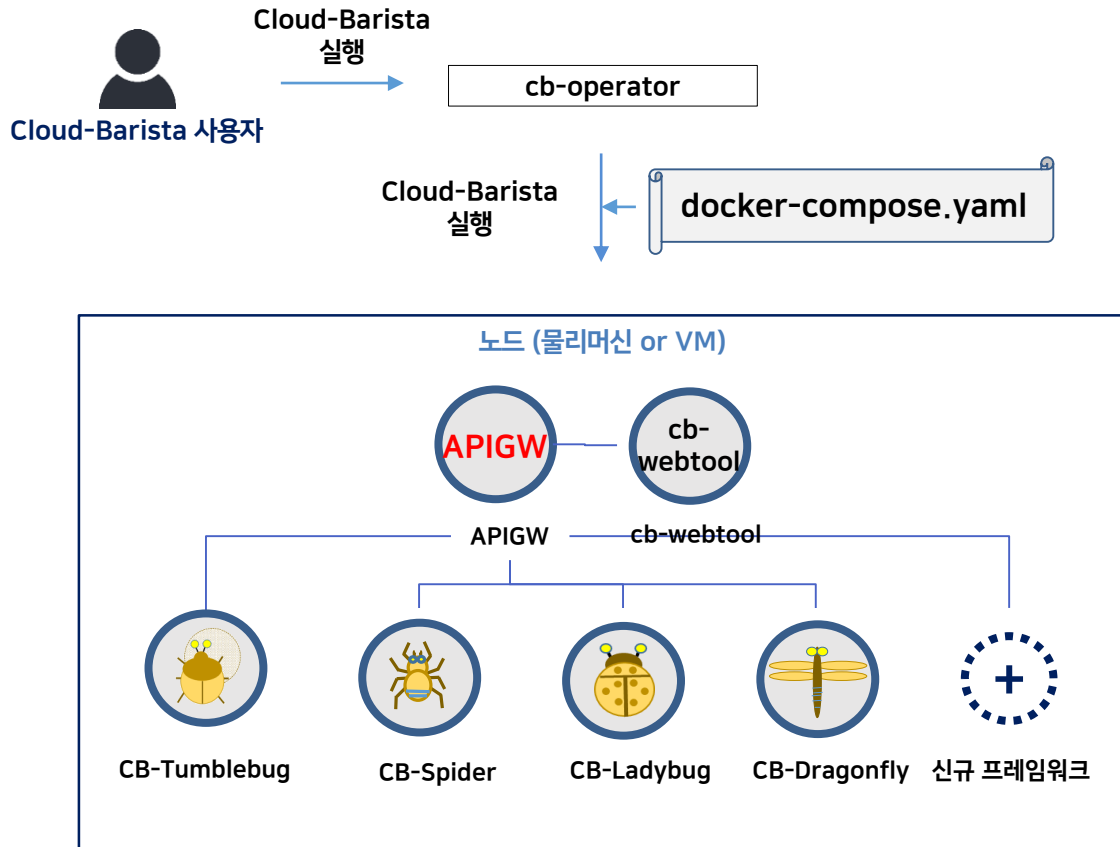
- Cloud-Barista 설치: 공개 이미지 저장소에서 Cloud-Barista 구성요소 컨테이너 이미지를 다운로드하여 저장
- Cloud-Barista 실행: Cloud-Barista 구성요소 컨테이너 이미지를 이용하여 Cloud-Barista 시스템 실행
- Cloud-Barista 상태 정보 보기: 실행 중인 Cloud-Barista 시스템의 상태를 출력
- Cloud-Barista 종료: Cloud-Barista 시스템을 종료하며, 컨테이너 이미지 삭제 옵션도 제공



# Cloud-Barista의 실행 모드

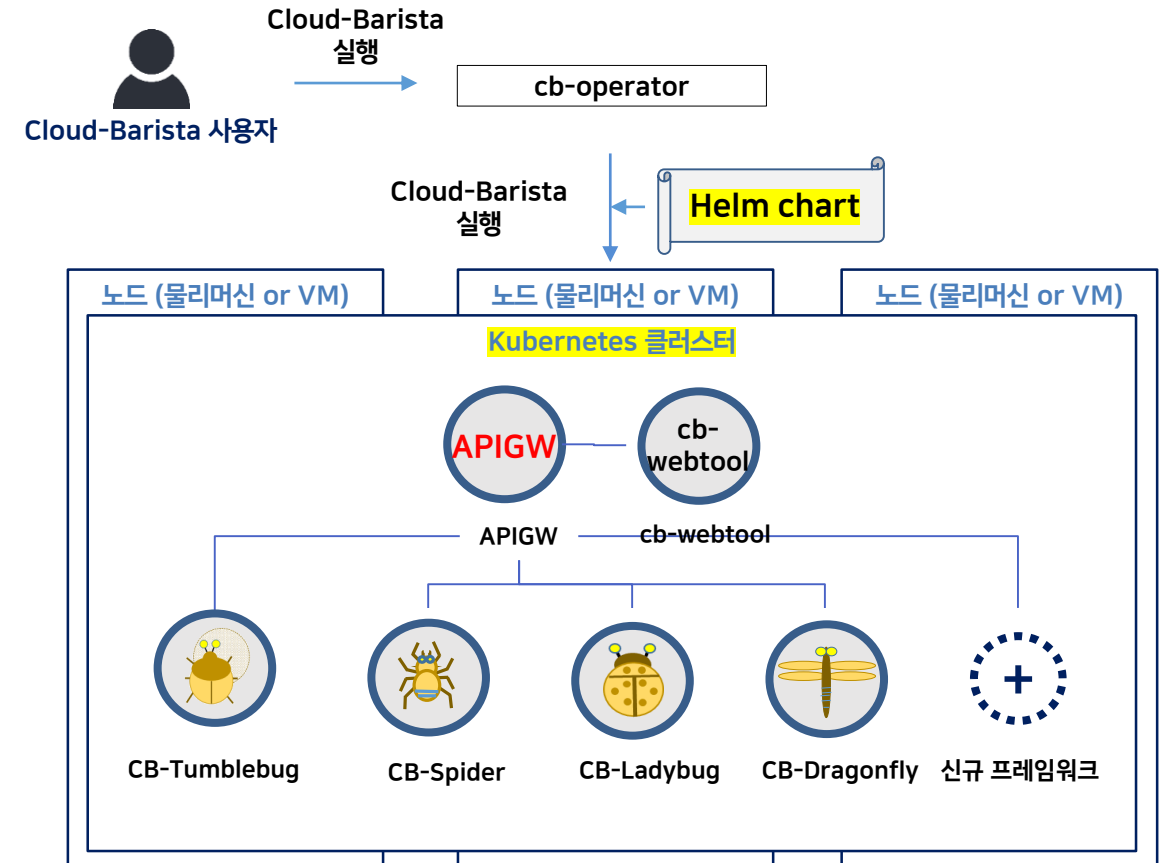
## Cappuccino (2020. 6.)

- Docker Compose 모드
- 단일 노드 환경에 적합



## Espresso (2020. 11.)

- Kubernetes 모드
- 단일/다중 노드 환경에 적합





# Cloud-Barista API Gateway (cb-apigw)

- 정의: Cloud-Barista 시스템에서 내/외부간 API 호출을 중계하는 엔티티

- 필요성

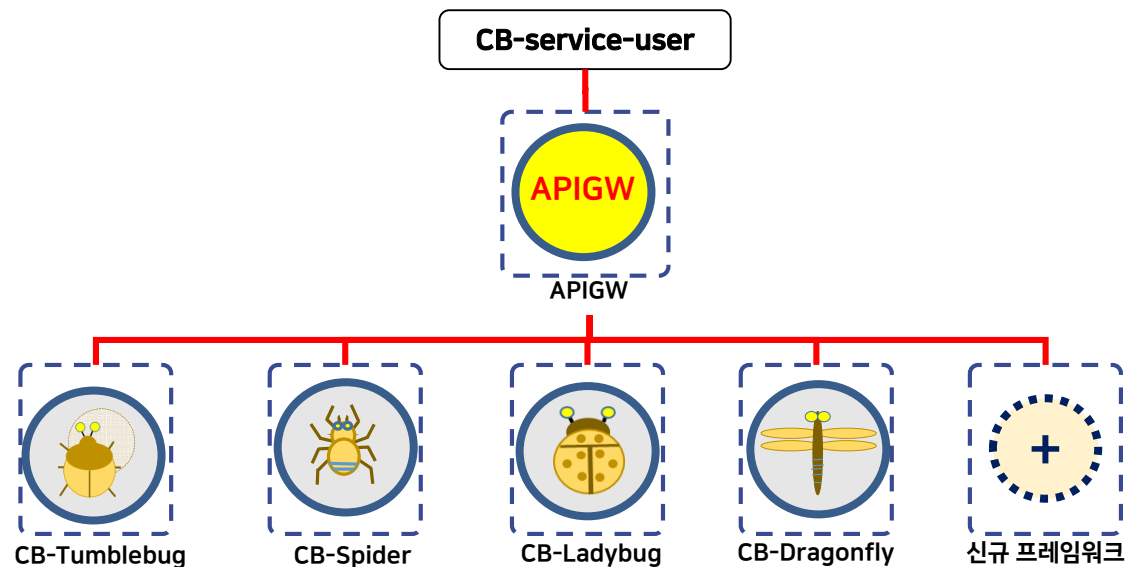
- API 엔드포인트 단일화
- 단일화된 API 통로에서의 로깅, 트레이싱, 모니터링

- 구성

- REST API GW (★)
- gRPC API GW

- 기능

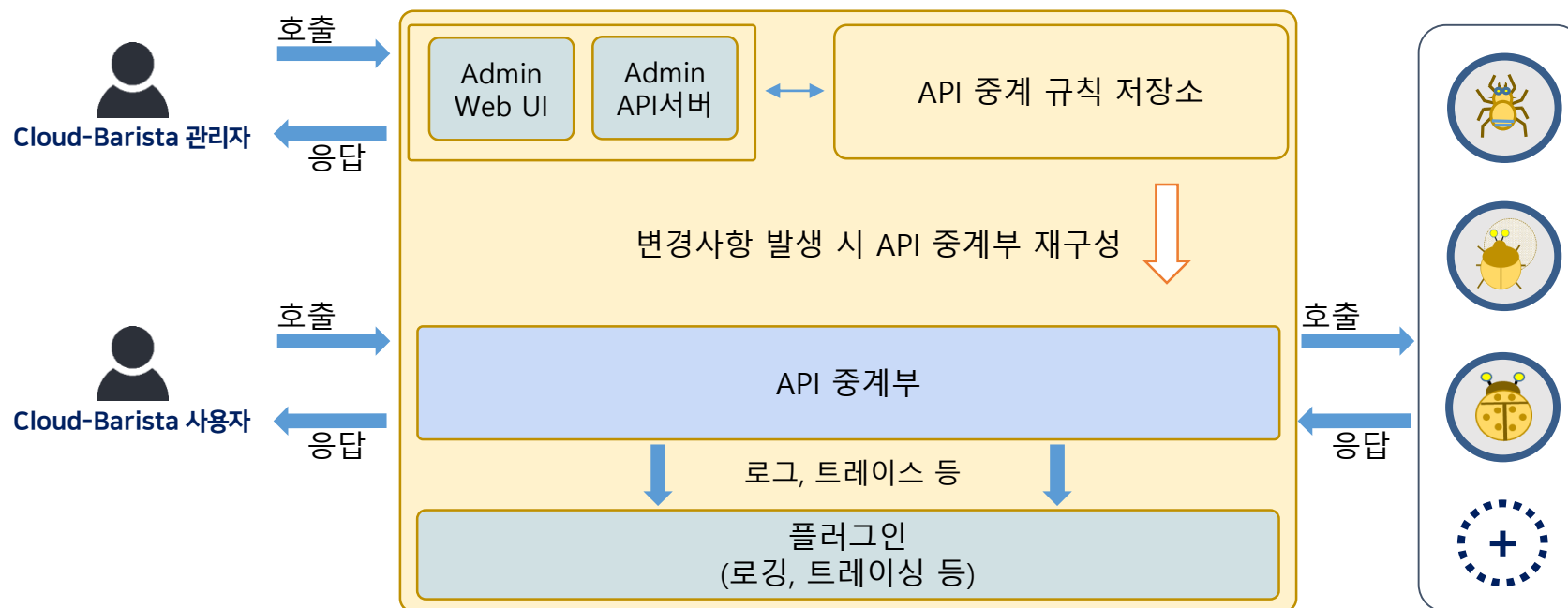
- API 중계
- YAML 파일을 통한 환경설정 기능
- API 중계 로깅 (cb-log 활용)
- API 중계 트레이싱 (Jaeger 활용)
- API 중계 모니터링 (go-metrics + InfluxDB + Grafana 활용)
- CORS, httppcache, httpsecure, Rate Limit 등
- JSON 응답 manipulate (flatten, sequential, merge, ...)





# cb-apigw: 실행 및 설정 / 동작 흐름도

- Cloud-Barista 관리자
  - cb-apigw 실행: cb-operator 등을 통해 cb-apigw를 포함하는 Cloud-Barista 시스템 실행
  - cb-apigw 환경설정: Admin Web UI를 통해 cb-apigw 환경설정을 동적으로 변경 가능
- Cloud-Barista 사용자
  - API 중계: cb-apigw 통해 각 프레임워크와 통신
  - API 모니터링: cb-apigw가 기록한 로깅/트레이싱/모니터링 정보를 확인





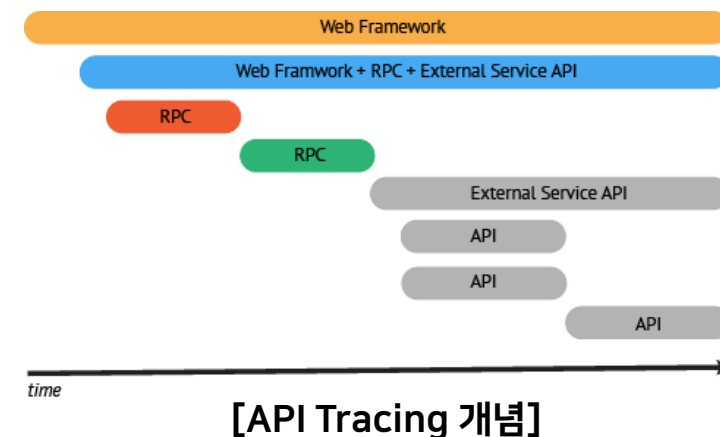
# [향후 개발 기술] Cloud-Barista 내에서의 API Tracing

## • 개념

- MSA (Microservice Architecture) 구조를 갖는 Cloud-Barista 시스템에서
- 한 API 호출과 그로 인해 발생한 파생 API 호출들이 Cloud-Barista 구성요소들 간에 어떻게 발생/처리되는지를 기록

## • 필요성

- Cloud-Barista 개발자 입장:
  - 개발 시 의도치 않은 / 예상치 못한 / 정상적이지 않은 API 호출을 발견할 수 있음
  - API 호출 경로 데이터를 취합하여 개발 문서 생성 등에 활용할 수 있음
- Cloud-Barista 사용자 입장:
  - 각 API 호출들이 어떤 경로로 처리되는지 시각적으로 확인하고 분석할 수 있음



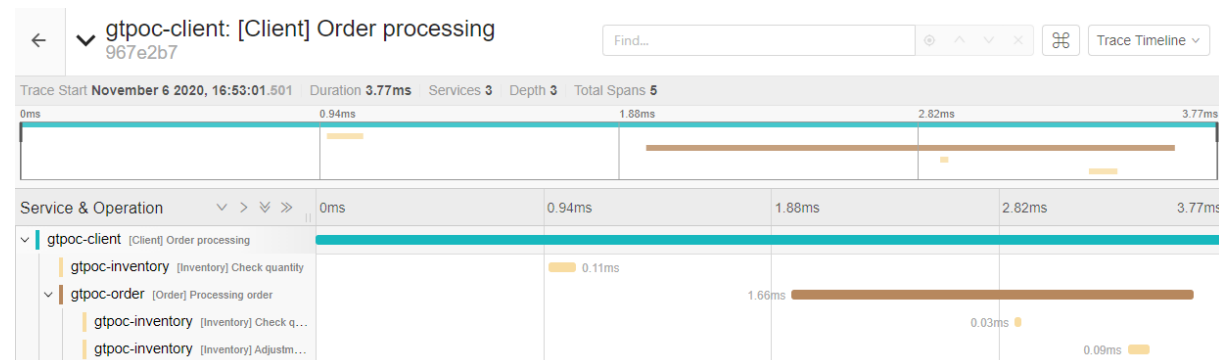
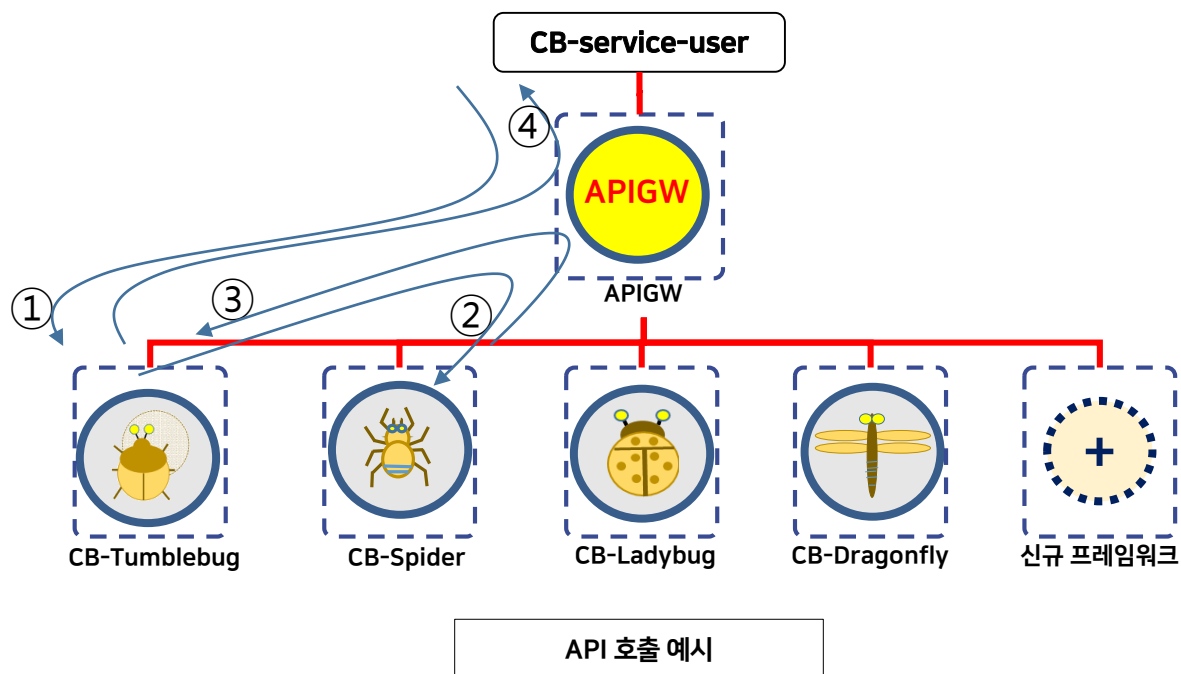
## • 동작

- Flow가 시작되는 곳 (각 프레임워크 또는 API 게이트웨이) 에서 API 호출 시 HTTP 헤더를 붙임
- API 호출이 전달될 때마다 HTTP 헤더를 보고 trace 정보를 기록

## • 추진 일정

- ~ 2020. 12.: PoC 진행
- Cloud-Barista Cafe Mocha 버전 공개 시점: 프로토타입 공개

# Cloud-Barista 내에서의 API Tracing 예시



API 트레이스 타임라인 시각화



API 호출 간 관계도 시각화



# Cloud-Barista 개발에서의 CI (Continuous Integration)

Cappuccino  
(2020. 6.)

Docker Hub을 이용한  
컨테이너 이미지 자동 빌드



Espresso  
(2020. 11.)

GitHub Actions 이용한  
컨테이너 이미지 자동 빌드



GitHub Actions

Cafe Mocha  
(2021.)

- 매 commit/PR/day 마다 빌드 & 등록
- Docker Compose 활용한 연동 테스트
- 멀티 아키텍처용 컨테이너 이미지 빌드
- ...



GitHub Actions  
enhanced

# CB-Bridge 개발 히스토리

|                | Americano<br>(2019. 11.)  | Cappuccino<br>(2020. 06.)   | Espresso<br>(2020. 11.)   |
|----------------|---|---|---|
| 시스템 설치 및 실행 관리 |   | <ul style="list-style-type: none"> <li>• cb-operator 공개</li> <li>• (Docker Compose 모드)</li> </ul>                           | <ul style="list-style-type: none"> <li>• cb-operator 고도화</li> <li>• (K8s 모드 추가)</li> </ul>  |
| 시스템 운용 지원      | <ul style="list-style-type: none"> <li>• cb-restapigw 1차 개발 완료</li> </ul> | <ul style="list-style-type: none"> <li>• (CB 시스템 실행형상에 cb-restapigw 통합)</li> <li>• cb-restapigw 일부 수정 및 기능 추가 구현</li> </ul> | <ul style="list-style-type: none"> <li>• cb-restapigw LB기능 추가</li> <li>• cb-restapigw 구조변경</li> <li>• cb-restapigw 관리API/UI 추가</li> <li>• cb-restapigw 설정동적반영기능 추가</li> </ul> |
| 시스템 개발 지원      | <ul style="list-style-type: none"> <li>• cb-store, cb-log 등 공개</li> </ul> | <ul style="list-style-type: none"> <li>• 이미지 자동 빌드 및 등록 기능 공개</li> </ul>  | <ul style="list-style-type: none"> <li>• GitHub Actions 기반의 CI 활성화</li> <li>• 개발 공통 구성요소 고도화</li> </ul>   |



## 추가/고도화 예정 기술 (차기 버전인 Cafe Mocha 버전)

---

- Cloud-Barista 시스템 설치 및 실행 관리
  - 개별 프레임워크 부하분산/장애복구 기능 추가
  - 개별 프레임워크 모니터링 기능 추가
- Cloud-Barista 시스템 운용 지원
  - cb-restapigw API Tracing 기능 추가
- Cloud-Barista 시스템 개발 지원
  - CI 강화 (유닛 테스트, 연계/연동 테스트 등)

# CB-Bridge 기술 시연

- Helm Chart 기반의 Cloud-Barista 실행 및 운용/제어 -

에스프레소(Espresso) 한잔 어떠세요 ? ^^

# 감사합니다.

<https://github.com/cloud-barista>  
<https://cloud-barista.github.io>

(서 지 훈 / [contact-to-cloud-barista@googlegroups.com](mailto:contact-to-cloud-barista@googlegroups.com))

## “Journey to the Multi-Cloud”

클라우드 바리스타들의 세 번째 이야기

### Cloud-Barista Community the 3<sup>rd</sup> Conference