



# 멀티클라우드, 글로벌 스케일로 시작하다

## 클라우드바리스타 커뮤니티 제4차 컨퍼런스

### [세션6] CB-Dragonfly : 멀티클라우드 대규모 통합 모니터링

김 효 경

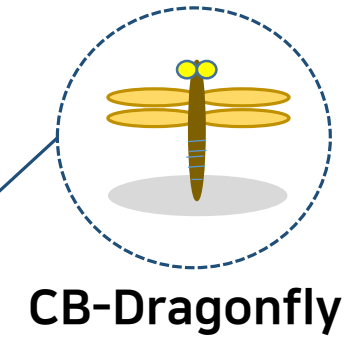
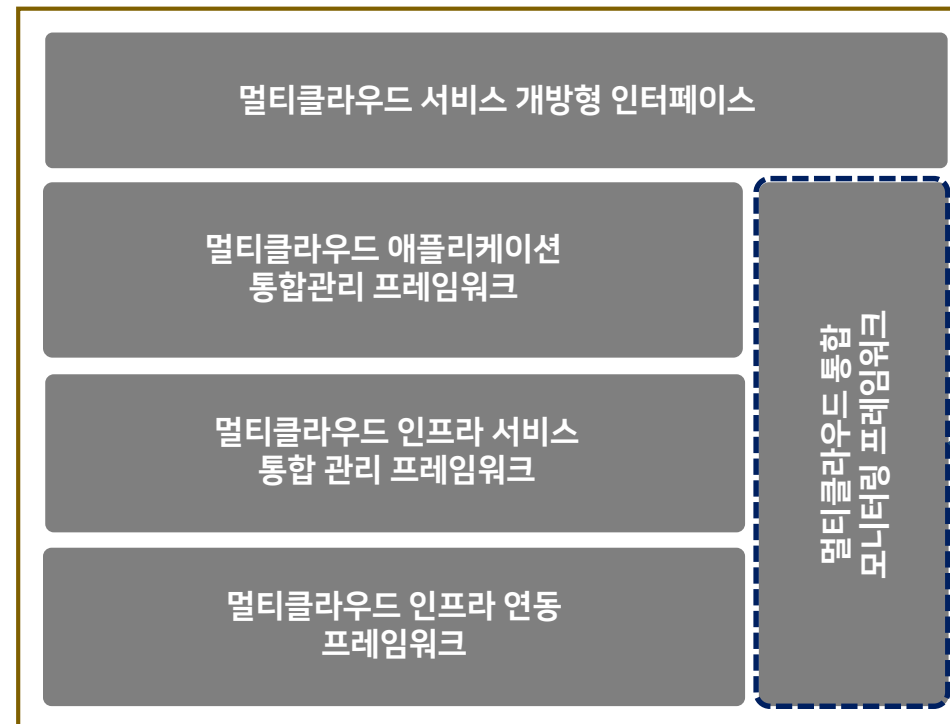
CB-Dragonfly 프레임워크 리더



카페모카(Café Mocha) 한잔 어떠세요 ?

# 이번 세션은...

응용/도메인/기관 특화 SW



멀티클라우드 서비스 공통 플랫폼

# 목 차

---

I

CB-Dragonfly 프레임워크 기술 개요

II

CB-Dragonfly 프레임워크 주요 기능

III

CB-Dragonfly 프레임워크 개발 현황 (Café Mocha)

IV

CB-Dragonfly 프레임워크 로드맵

V

CB-Dragonfly PULL 방식 모니터링 시연

# CB-Dragonfly 프레임워크 기술 개요

- 이종 클라우드 연동 환경에서의 대규모 인프라 및 애플리케이션 통합 모니터링 기능을 제공하는 멀티클라우드 통합 모니터링 프레임워크

## 멀티클라우드 통합 모니터링 특징



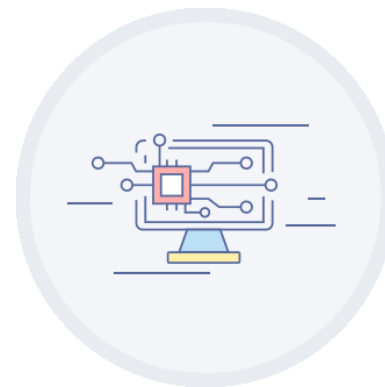
### 대규모 멀티클라우드 서비스 모니터링

대규모 멀티클라우드 모니터링을 위한  
메시지 큐 기반 모니터링 트래픽 부하분산 및  
모니터링 데이터 처리 기능 제공



### 멀티클라우드 특화 모니터링

멀티클라우드 인프라 서비스 기반  
MCIS, MCKS, MC-App 모니터링 등  
특화 모니터링 발굴 및 수집/제공

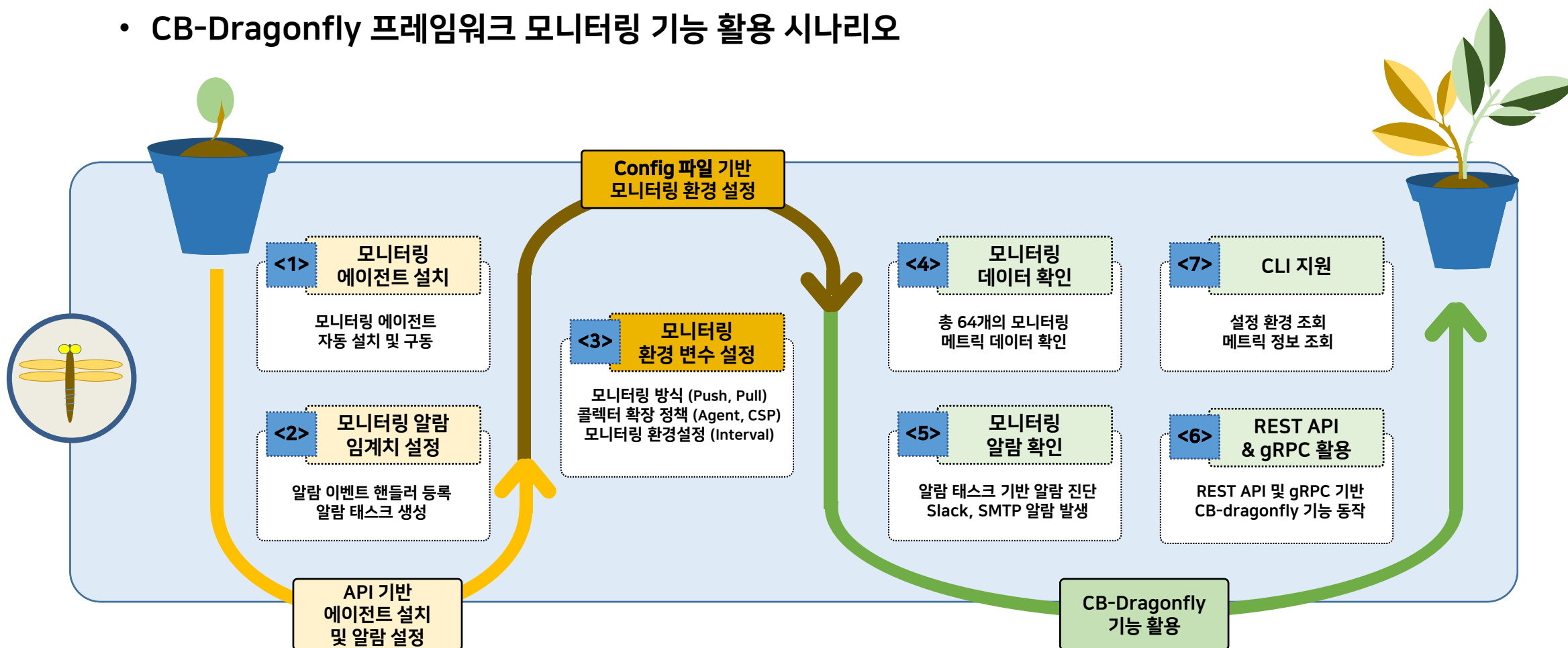


### 다양한 이종 환경의 유연한 모니터링

에이전트 기반 VM 환경, 컨테이너 환경 등  
다양한 환경에서 모니터링 기능 제공

# CB-Dragonfly 프레임워크 사용자 관점 활용 시나리오

## • CB-Dragonfly 프레임워크 모니터링 기능 활용 시나리오



# CB-Dragonfly 프레임워크 모니터링 수집 방식 및 대상

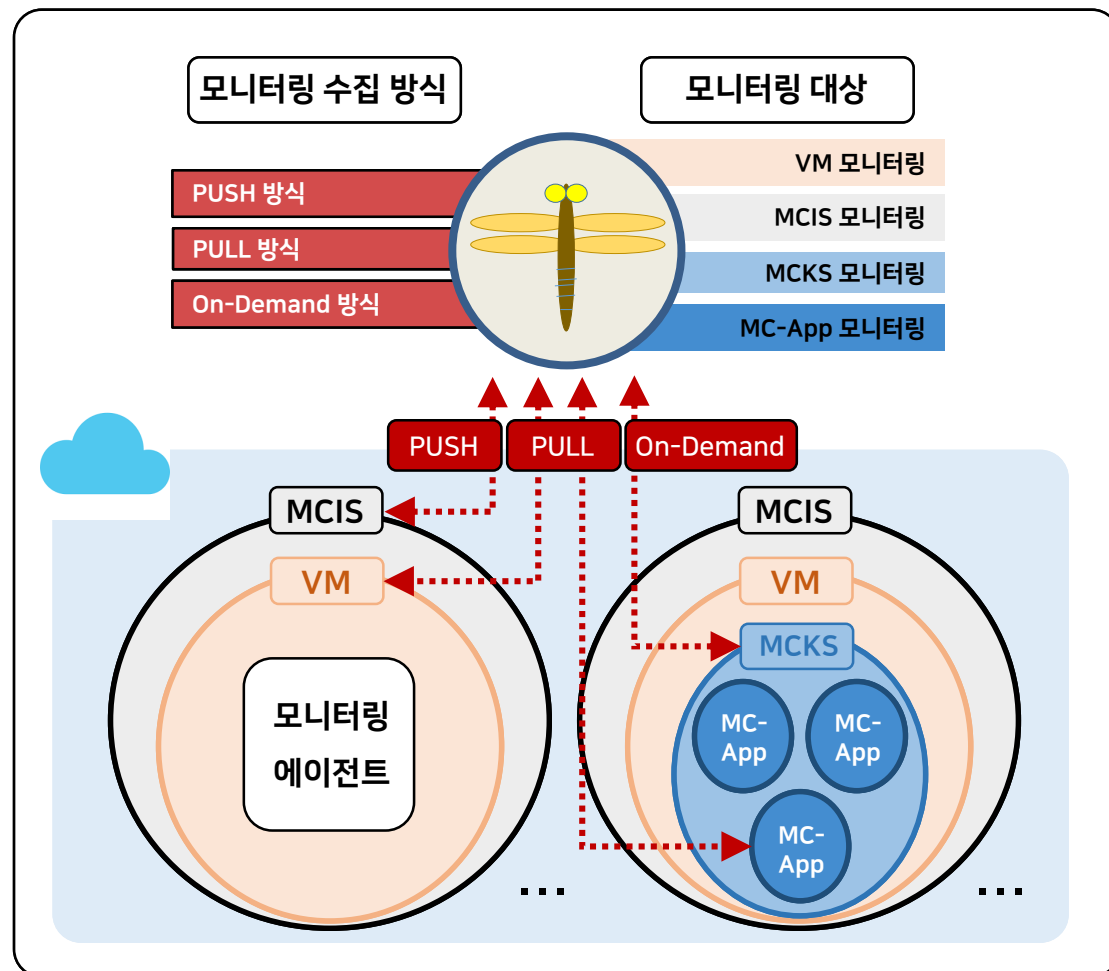
## • CB-Dragonfly 프레임워크 모니터링 수집 방식 및 대상

### 모니터링 수집 방식

- 에이전트 기반 PUSH 방식
- CB-Dragonfly 서버 기반 PULL 방식
- 사용자 요청 시점 기준 On-Demand 방식

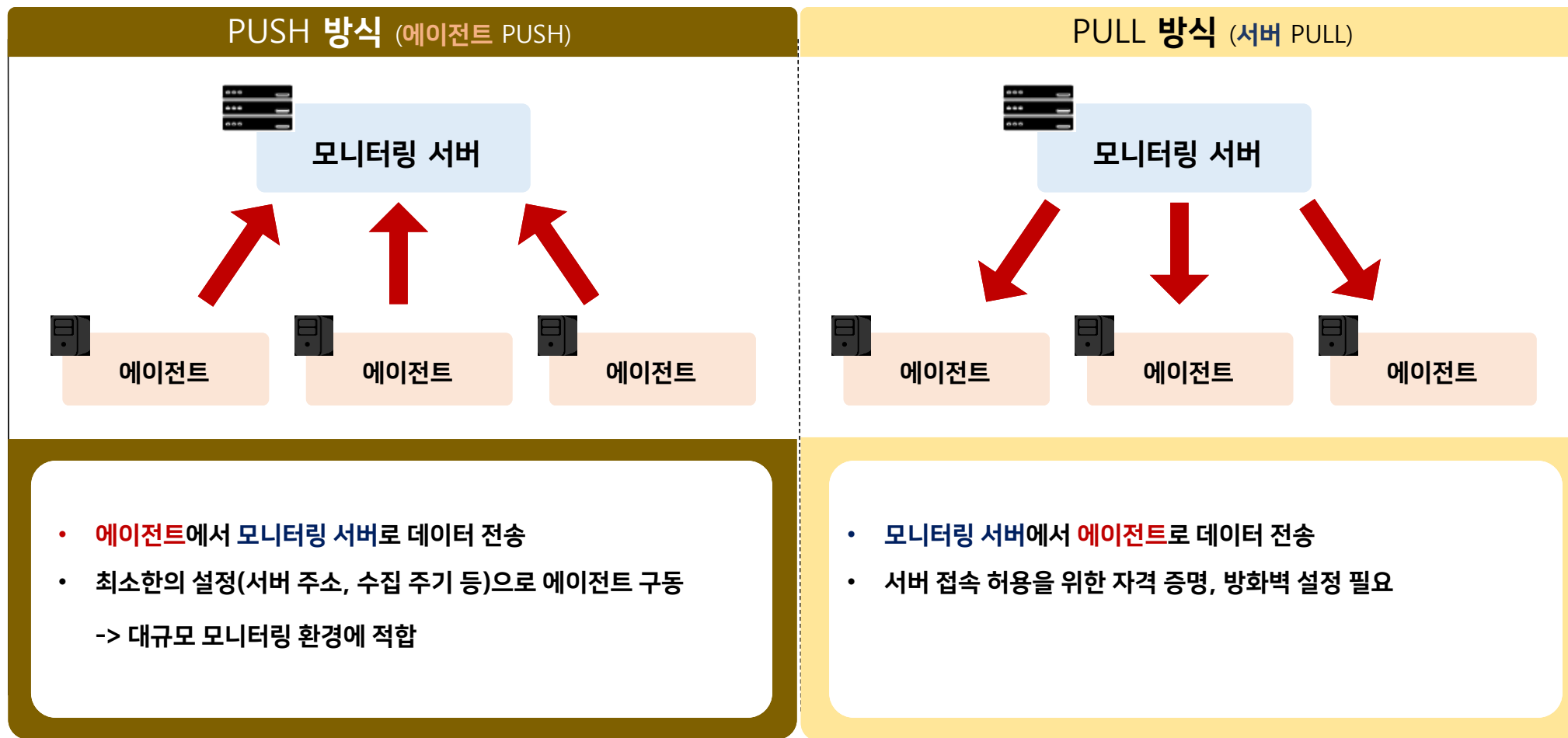
### 모니터링 대상

- VM 모니터링
  - 에이전트 기반 모니터링 수행
  - CPU, Memory, Disk 등 인프라 모니터링 메트릭 제공
- MCIS 모니터링
  - 에이전트 기반 모니터링 수행
  - CPU read/write, File I/o 등 성능 모니터링 메트릭 제공
- MCKS 모니터링
- MC-App 모니터링

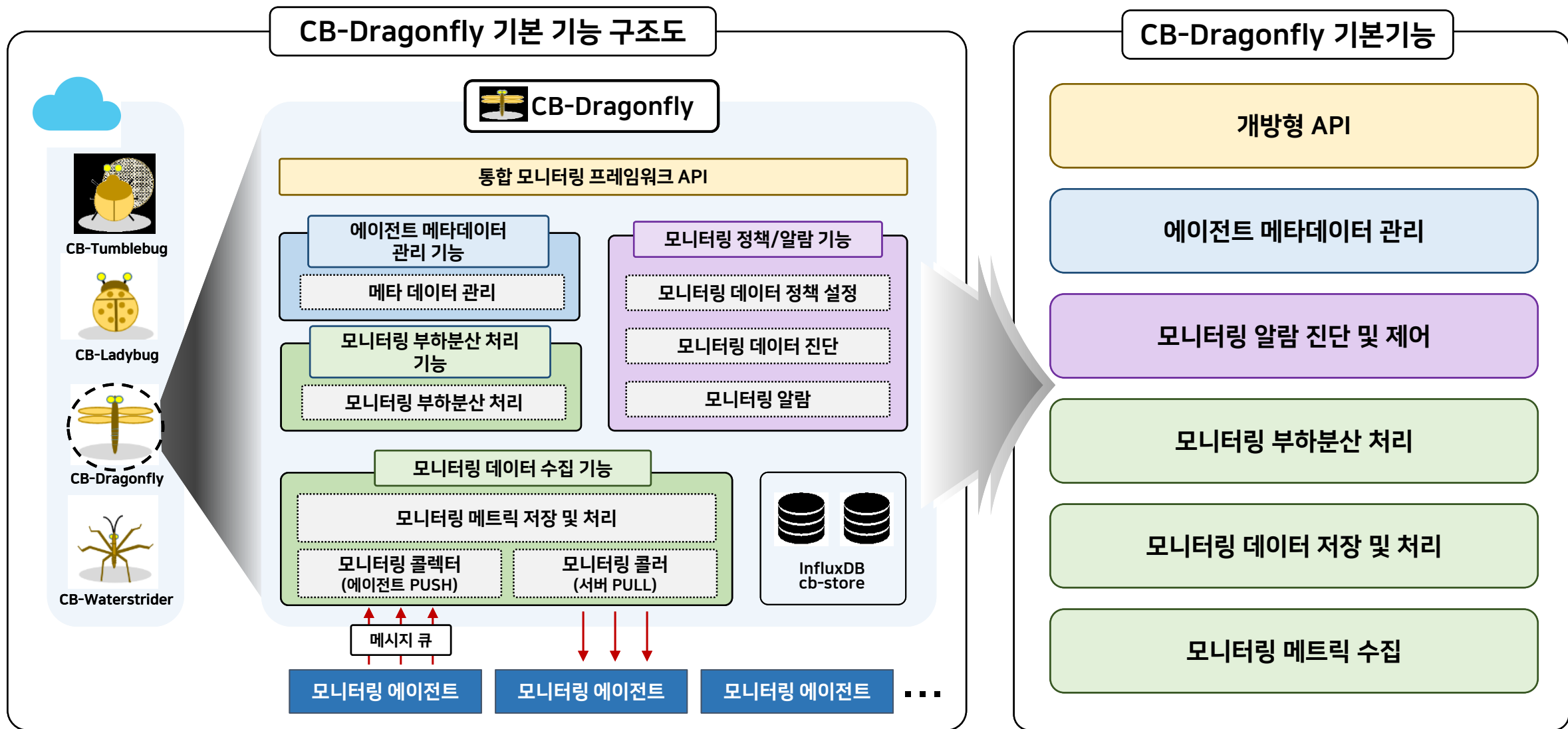


# 모니터링 PULL 방식과 PUSH 방식 소개

## • PUSH 방식, PULL 방식 모니터링 비교

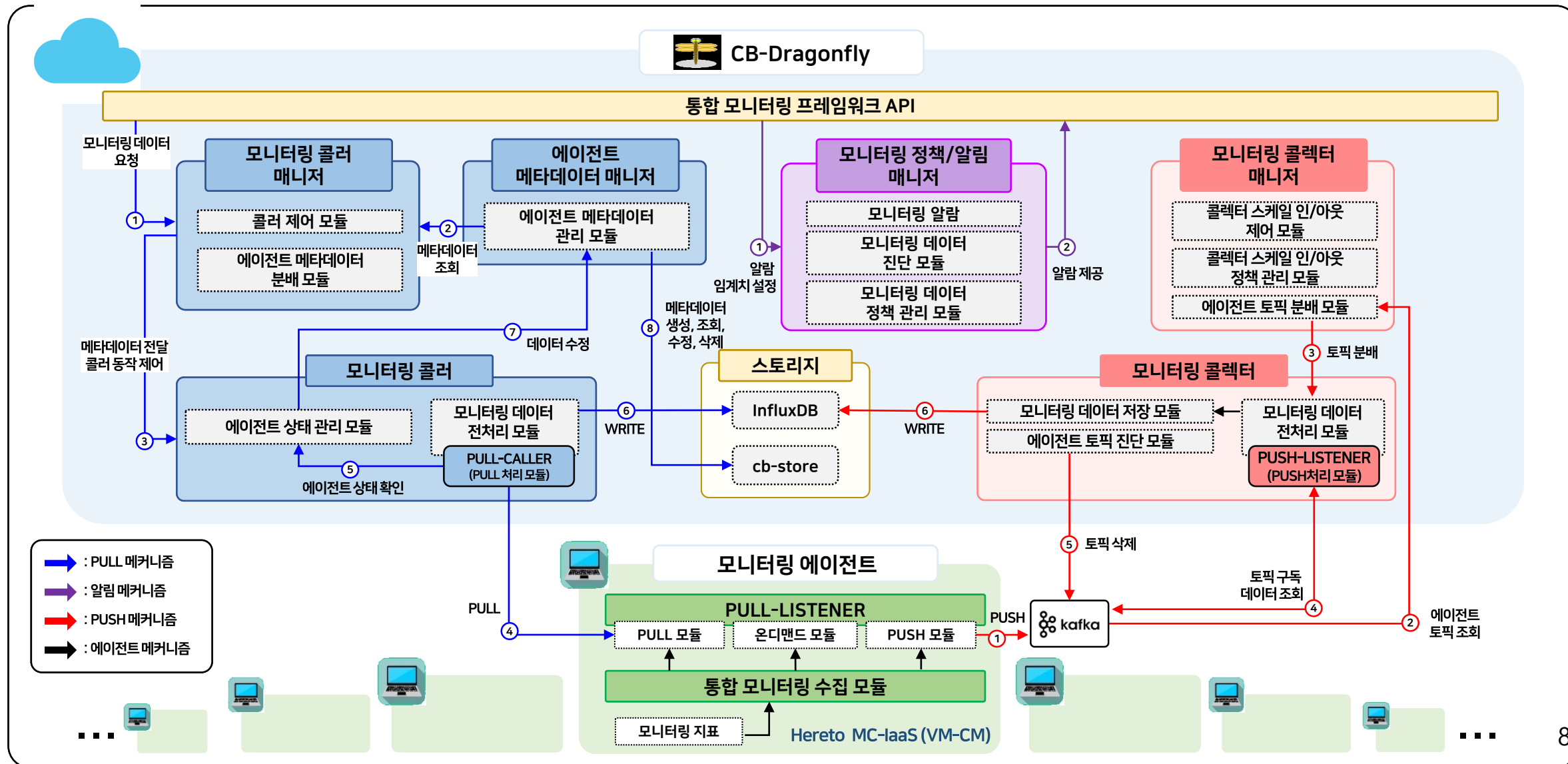


# CB-Dragonfly 프레임워크 기본 기능 구조도

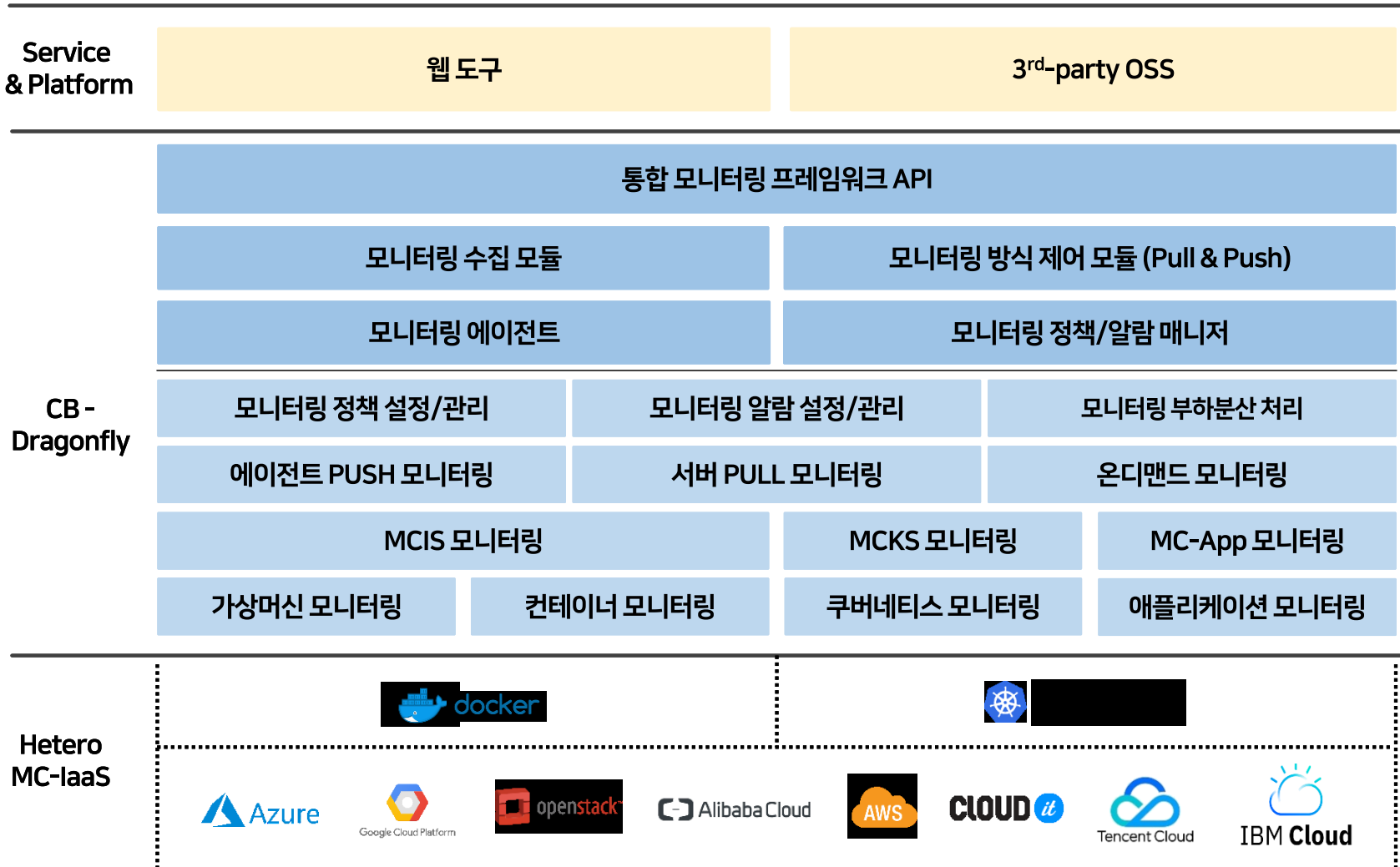




# CB-Dragonfly 프레임워크 기본 기능 세부 구조도



# CB-Dragonfly 프레임워크 아키텍처



# CB-Dragonfly 주요 기능

---

- 1 • 에이전트 기반 모니터링 메트릭 수집 기능
- 2 • 메트릭 저장 및 처리 기능
- 3 • 모니터링 부하분산 처리 기능
- 4 • 모니터링 에이전트 메타데이터 관리 기능
- 5 • 모니터링 알람 진단 및 제어 기능
- 6 • 개방형 API

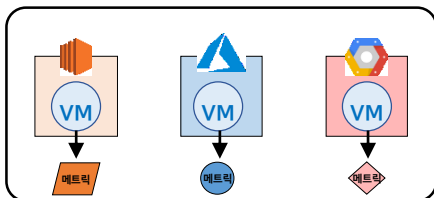
# 에이전트 기반 모니터링 메트릭 수집 기능 (1/3)

## • 모니터링 에이전트

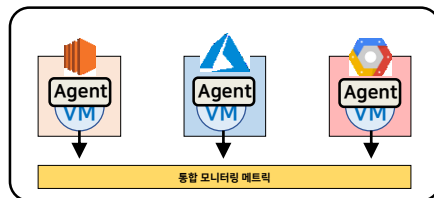
- 가상머신에 설치된 에이전트를 기반으로 모니터링 메트릭 수집
- CSP에 종속되지 않고 **통합 모니터링 메트릭** 제공
- 에이전트 기반의 **VM 모니터링**, **MCIS 모니터링** 제공

### 모니터링 특징

멀티클라우드 환경에서 **통합 모니터링 메트릭** 수집



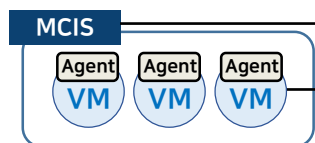
<AS-IS>  
CSP 별 모니터링 메트릭 제공



<TO-BE>  
에이전트 기반 통합 모니터링 메트릭 제공

### 모니터링 대상

모니터링 수집 대상에 따라 **VM 모니터링**, **MCIS 모니터링** 제공



※ MCIS: Multi-Cloud Infra Service  
※ VM: Virtual Machine

MCIS 모니터링

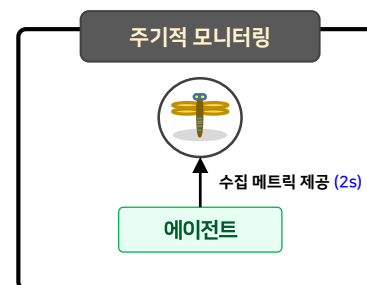
- MCIS 성능 모니터링

VM 모니터링

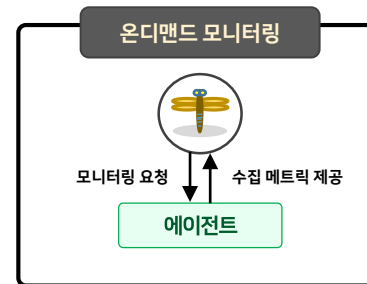
- VM 리소스 모니터링

### 모니터링 수집 방식

모니터링 수집 방식에 따라 **주기적 모니터링**, **온디맨드 모니터링** 제공



- 에이전트 기본 동작 방식
- 에이전트에 설정된 주기로 모니터링 정보 수집
- 수집된 모니터링 데이터를 CB-Dragonfly에 전달



- 사용자가 요청한 시점의 모니터링 메트릭 제공
- CB-Dragonfly에서 에이전트에 모니터링 요청
- 요청 시점에 에이전트는 모니터링 메트릭 수집
- 수집된 메트릭 데이터를 CB-Dragonfly에 전달

## 에이전트 기반 모니터링 메트릭 수집 기능 (2/3)

- 모니터링 메트릭
  - Café Mocha 버전 기준 총 64개의 모니터링 메트릭 제공
  - 메트릭 종류에 따라 VM 모니터링 메트릭, MCIS 모니터링 메트릭 제공

### 모니터링 메트릭

VM 모니터링 메트릭

분류	모니터링 항목	개수
CPU	cpu_usr, num, ...	19개
CpuFreq	cpu_speed(freq)	1개
Memory	swap_utilization, ...	14개
Disk	read_time, w_time ...	10개
Network	errors_in, err_out, ...	8개

CPU, Memory, Network 등 VM 모니터링 메트릭 52개 제공

MCIS 모니터링 메트릭

분류	모니터링 항목	개수
CPU	single,, multiple	2개
Memory	read, write	2개
File I/O	read, write	2개
Network	rtt/mrtt	2개
DB	create_time, ...	4개

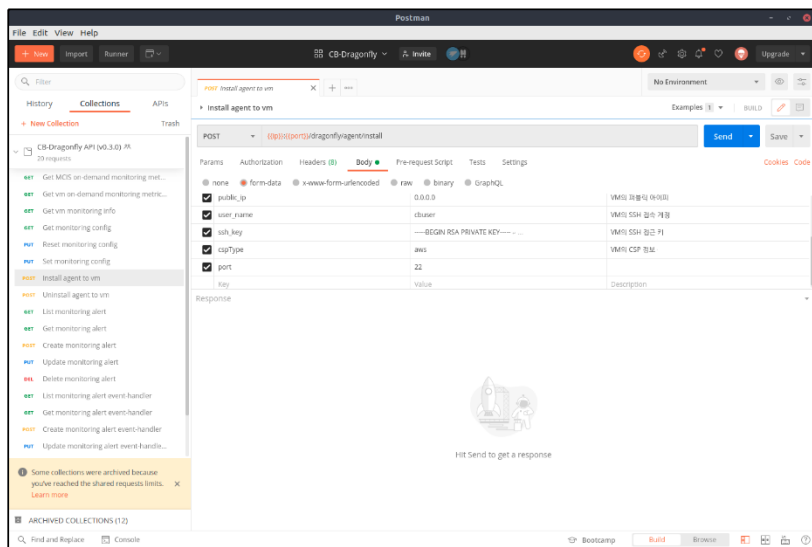
CPU, Memory, File I/O 등 MCIS의 성능 측정 메트릭 12개 제공

# 에이전트 기반 모니터링 메트릭 수집 기능 (3/3)

- 모니터링 에이전트 설치 자동화
  - (Linux) 에이전트 설치 API 제공
  - (Windows) 에이전트 설치 도구 제공

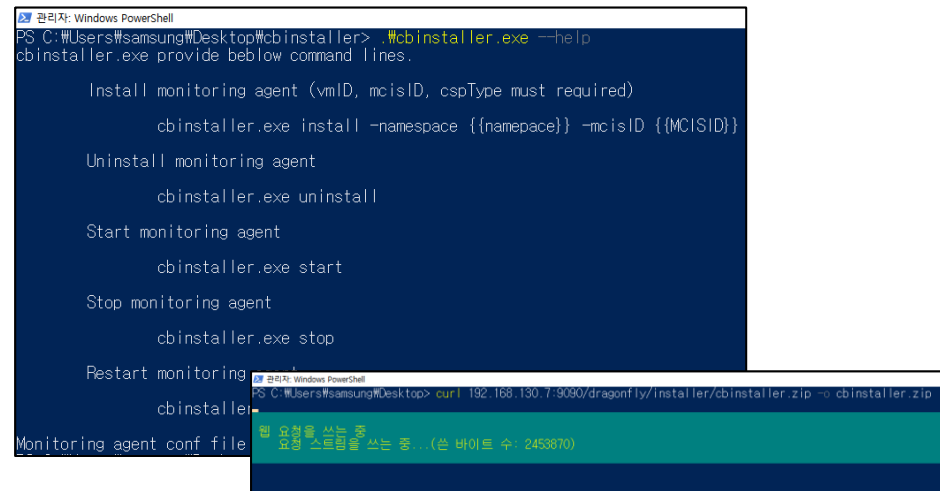
## 에이전트 설치 API

- MCIS 내의 VM 생성 시 API 기반 에이전트 자동 설치



## 에이전트 설치 도구

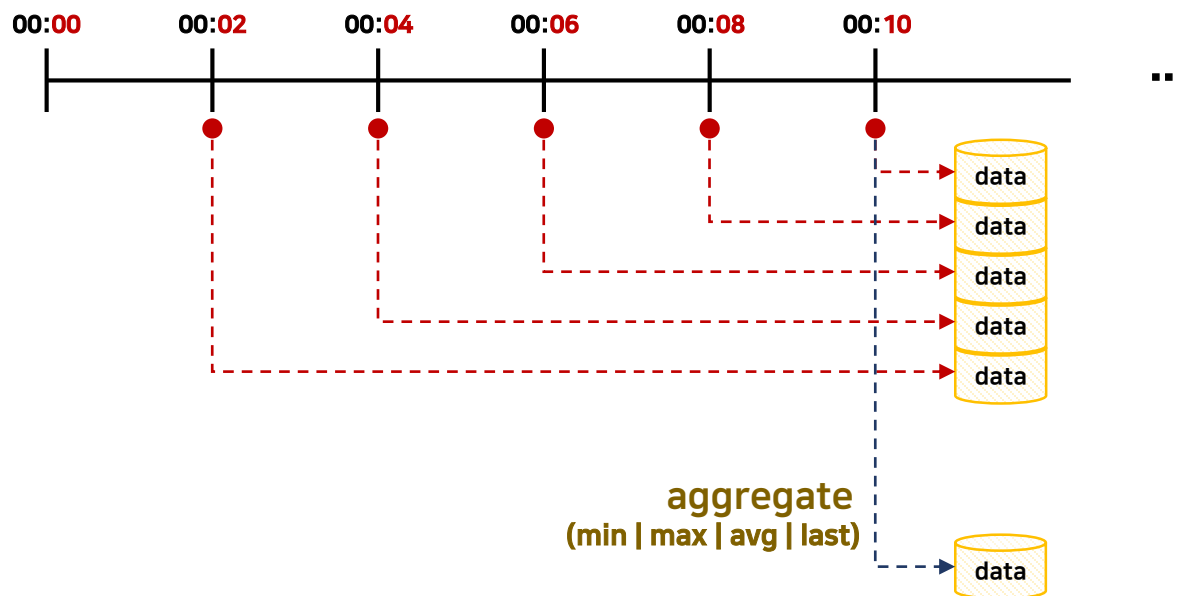
- Windows 환경에서 에이전트 서비스 등록 도구 제공



# 모니터링 메트릭 저장 및 처리 기능

- 모니터링 Aggregator 모듈
  - Aggregate 주기에 따라 캐싱된 모니터링 데이터를 처리 후 TSDB(InfluxDB)에 저장
  - Aggregate 함수: **MIN, MAX, AVG, LAST**

모니터링 Aggregator 모듈 동작 방식

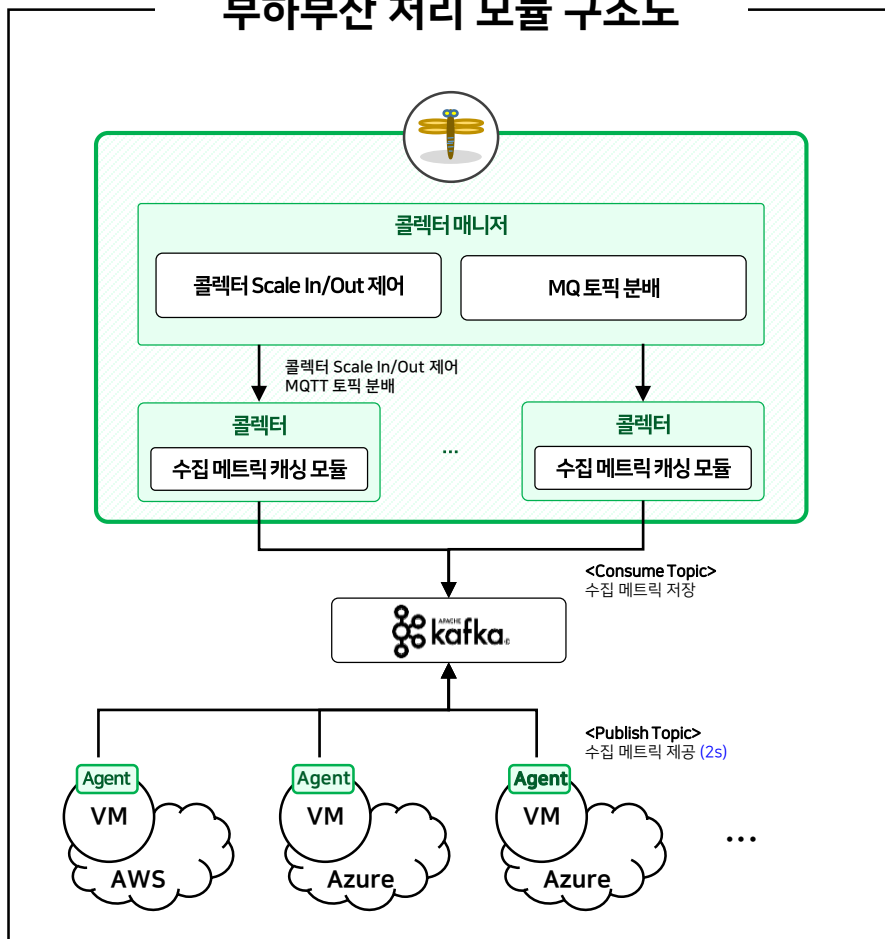


\* 모니터링 데이터 기본 수집 주기 : 2 Sec  
\* 모니터링 데이터 기본 aggregate 주기 : 10 Sec

# 모니터링 부하분산 처리 기능 (1/2)

- 대규모 환경의 모니터링 수집 데이터 트래픽을 부하분산 처리하는 기능

부하분산 처리 모듈 구조도

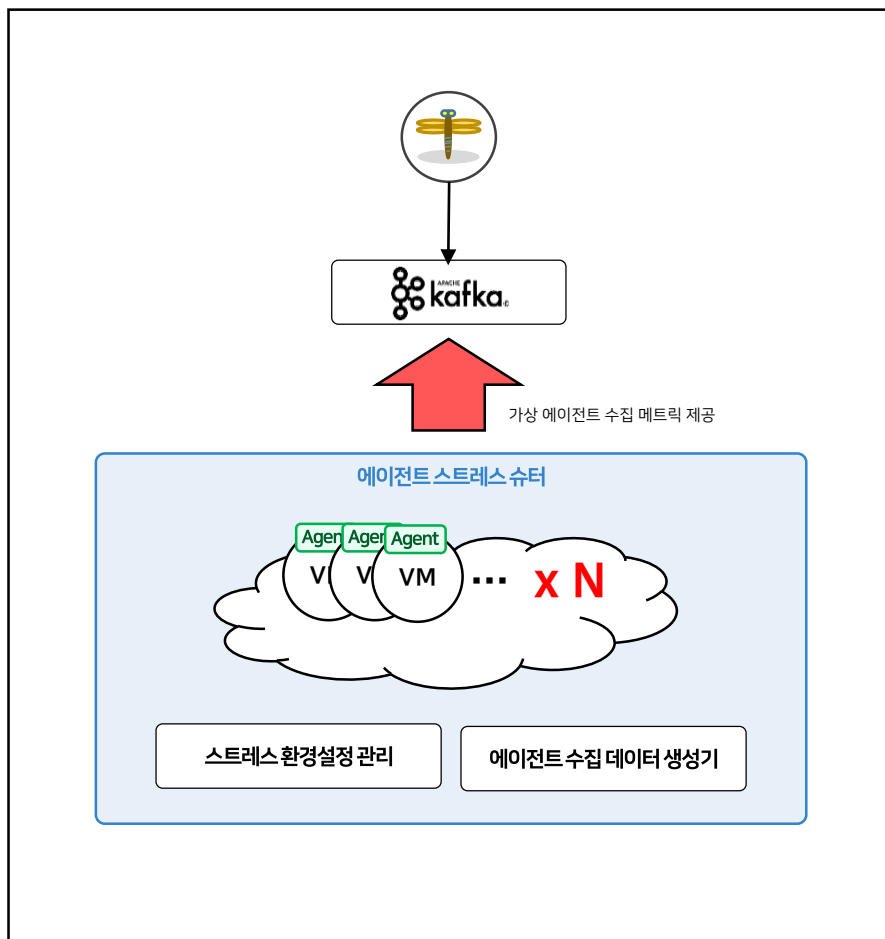


모듈	기능
콜렉터 매니저	<ul style="list-style-type: none"> <li>대규모 환경에서 모니터링 콜렉터의 스케일을 관리하는 모듈</li> <li>설정된 부하분산 처리 정책에 따라서 콜렉터 스케일 인/아웃</li> <li>카페모카 버전까지 부하분산 정책 2종 제공               <ol style="list-style-type: none"> <li>에이전트 수 기준</li> <li>CSP 종류 기준</li> </ol> </li> <li>Kafka 메시지 큐 토픽 목록 관리 및 토픽 분배</li> </ul>
콜렉터	<ul style="list-style-type: none"> <li>모니터링 데이터 캐싱 모듈</li> <li>Kafka 토픽 데이터 Consume 및 모니터링 데이터 캐싱</li> </ul>
메시지 큐 (Kafka)	<ul style="list-style-type: none"> <li>고가용성 및 확장성을 지원하는 MQ Broker</li> </ul>
에이전트	<ul style="list-style-type: none"> <li>모니터링 데이터 수집 모듈</li> <li>수집된 모니터링 메트릭을 Kafka 메시지 큐에 Publish</li> </ul>



## 모니터링 부하분산 처리 기능 (2/2)

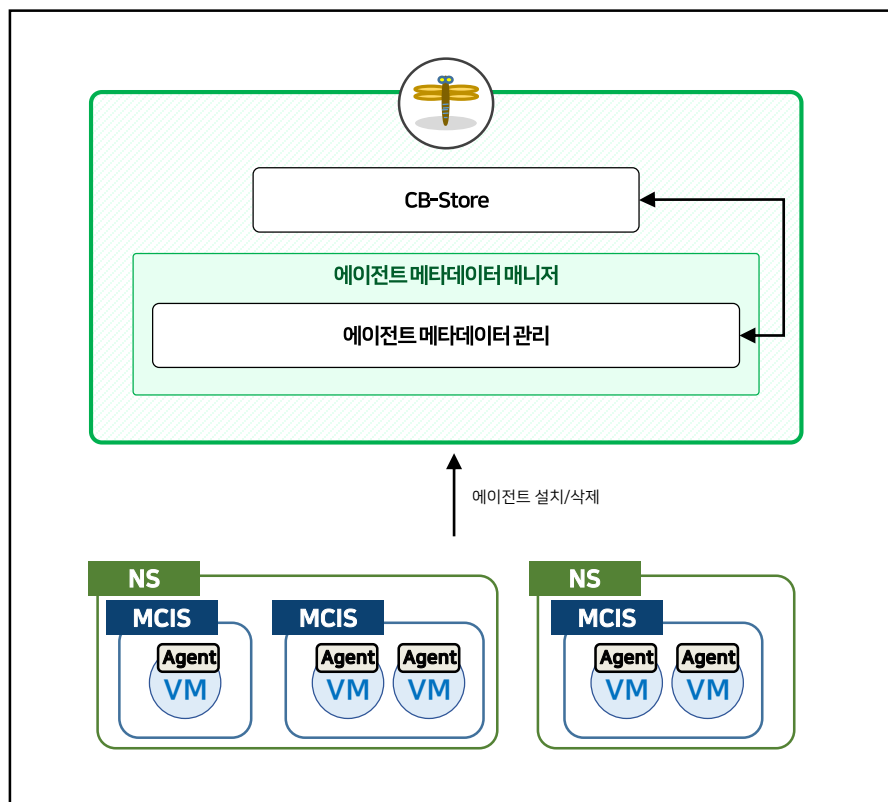
- 대규모 환경 부하분산 테스트 도구



- 대규모 환경 모니터링 테스트를 위해 스트레스 테스트 도구 개발
- 가상의 에이전트를 생성해 모니터링 수집 메트릭 데이터 생성
- 생성된 메트릭 데이터를 Kafka MQ에 전송
- 설정된 에이전트 수(N)에 따라서 모니터링 데이터 스트레스 조정

# 모니터링 에이전트 메타데이터 관리 기능

- 모니터링 에이전트 메타데이터 관리 기능
  - CB-Dragonfly FW에서 수집 중인 모니터링 에이전트 목록 관리
  - 모니터링 에이전트 태그 정보, CSP 유형, 수집 메커니즘, 헬스체크 상태 저장



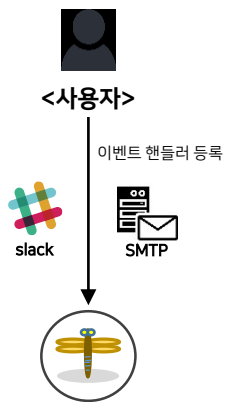
에이전트	키	항목	값
Agent1	{NS_ID}/ {MCIS_ID}/ {VM_ID}/ {CSP_TYPE}	NS ID	ns-01
		MCIS ID	mzc-mcis-01
		VM ID	mzc01-c-1-1yn8c
		CSP Type	AWS
		Agent Mechanism	push
		Public IP	210.207.100.101
		HealthCheck	healthy

# 모니터링 알람 진단 및 제어 기능

## • 모니터링 알람 진단 및 제어 기능

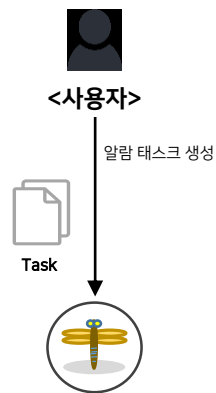
### 1. 이벤트 핸들러 등록

- 알람 이벤트 핸들러 등록
- **Slack, SMTP** 이벤트 핸들러 지원



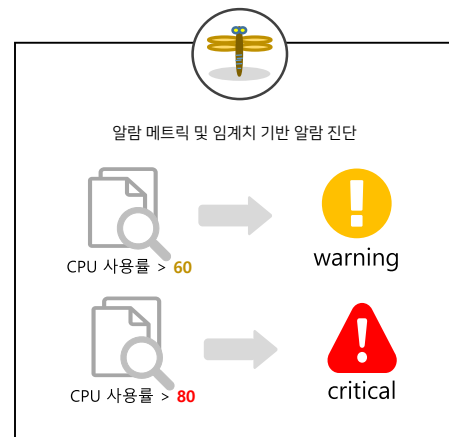
### 2. 태스크 생성

- 알람 태스크 생성
- 메트릭 및 알람 임계치 설정
- 메트릭 **3종** 지원
- **warning, critical** 알람 레벨 지원



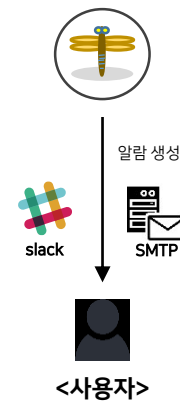
### 3. 알람 진단

- 생성된 태스크 기반 알람 진단
- 알람 발생 시 알람 발생 로그 생성



### 4. 알람 생성

- 등록된 이벤트 핸들러 기반 알람 메시지 전달



# 개방형 API

- CB-Dragonfly 프레임워크를 활용할 수 있는 다양한 API 도구 제공
  - REST API, Go API, CLI 유형으로 CB-Dragonfly 기능 활용

## REST API

- REST API 기반 CB-Dragonfly API 기능 제공
- 공식 문서:  
<https://documenter.getpostman.com/view/7454078/TzJu8wwi>

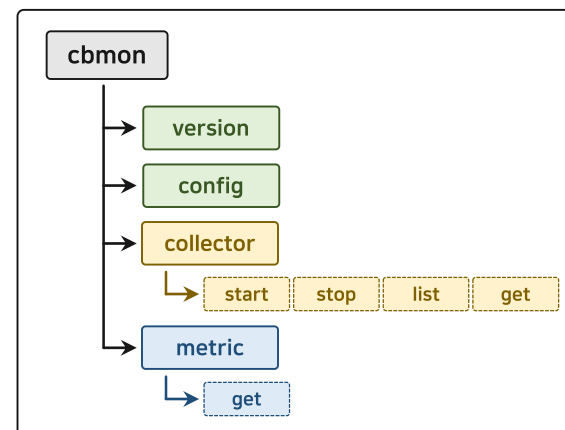
## Go API

- gRPC 기반의 CB-Dragonfly GO API 제공
- gogoproto 라이브러리 기반 gRPC 제공

## CLI 도구

- CLI 도구를 통해 CB-Dragonfly 기능 제공
- 환경설정 조회 기능 제공
- 콜렉터 제어 기능 제공
- 메트릭 정보 조회 기능 제공

**\$ cbmon**



# CB-Dragonfly 프레임워크 버전 별 개발 현황

- CB-Dragonfly 프레임워크 버전 별 개발 현황

## Americano

### 멀티클라우드 모니터링 기본 메커니즘 개발

- 콜렉터, 콜렉터 매니저 개발
- 모니터링 수집 및 처리 모듈 개발
- VM 모니터링 개발
- 최신 데이터 모니터링 개발
- etcd 기반 모니터링 캐싱

## Cappuccino

### 신규 모니터링 메트릭 발굴 및 개발

- VM 모니터링 메트릭 추가
- MCIS 모니터링 메트릭 설계
- 온디맨드 모니터링 설계
- Windows 환경 에이전트 설치 도구 개발

## Espresso

### 대규모 모니터링 기능 고도화

- Kafka 기반 부하분산 모듈 고도화
- MCIS 성능 모니터링 메트릭 개발
- 온디맨드 모니터링 개발
- CB-Store 기반 모니터링 캐싱
- 알람 모듈 개발
- gRPC 통신 및 CLI 도구 지원
- Cloud-Twin 환경 에이전트 구동

## Café Mocha

### 신규 모니터링 메커니즘 개발

- PULL 방식 모니터링 개발
- PUSH/PULL 메커니즘 기반 CB-DF 구동 모듈 고도화
- PUSH/PULL 메커니즘 기반 에이전트 모듈 고도화
- MCIS 특화 메트릭 도출
- MCKS, MCAS 모니터링 PoC

# CB-Dragonfly, 카페모카(Café Mocha) 개발 현황

---

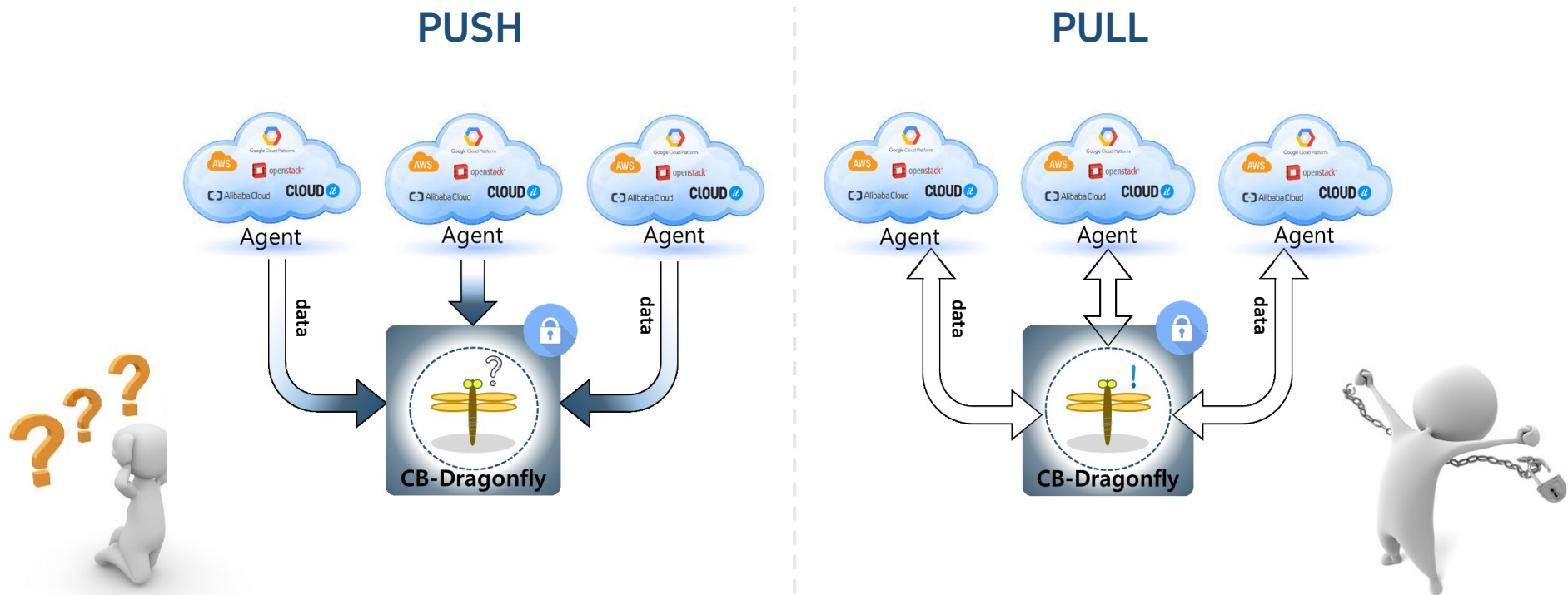
PULL 방식 모니터링  
메커니즘 개발

MCIS 특화 메트릭  
도출

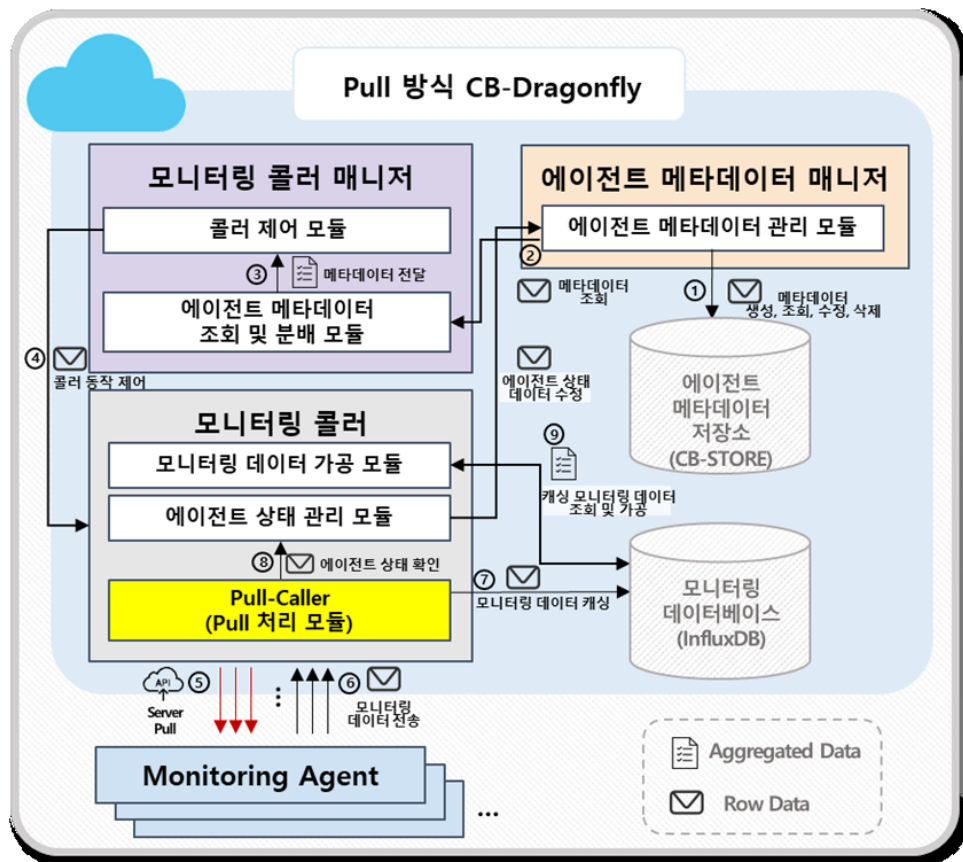
MCKS, MCAS  
모니터링 PoC

# CB-Dragonfly PULL 방식 모니터링 개요

- 퍼블릭 네트워크 환경의 CB-Dragonfly는 Agent-PUSH 모니터링
- 프라이빗 네트워크 환경의 CB-Dragonfly 이슈
  - 에이전트의 PUSH 모니터링 데이터 수신 불가능
  - Server-PULL 방식 모니터링 메커니즘 필요성



# CB-Dragonfly PULL 모니터링 구현 모듈



CB-Dragonfly

## 에이전트 메타데이터 매니저

(1) 에이전트 메타데이터 관리

## 모니터링 콜러 매니저

(2, 3) 에이전트 메타데이터 조회 및 분배

(4) 모니터링 콜러 동작 제어

## 모니터링 콜러

(5,6,7) 에이전트 PULL API 호출 및 모니터링 데이터 캐싱

(8) 에이전트 상태 전달

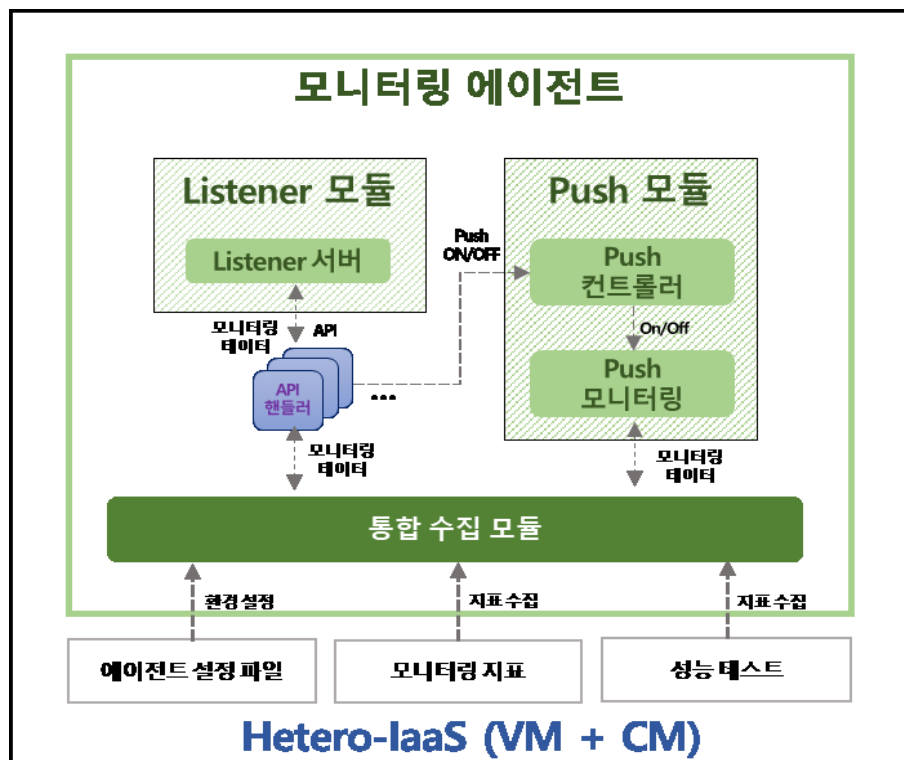
(9) 캐싱 모니터링 데이터 가공

PULL 실패 에이전트 상태 추적 및 메타데이터 수정 요청



# 에이전트 Listener 모듈 구현

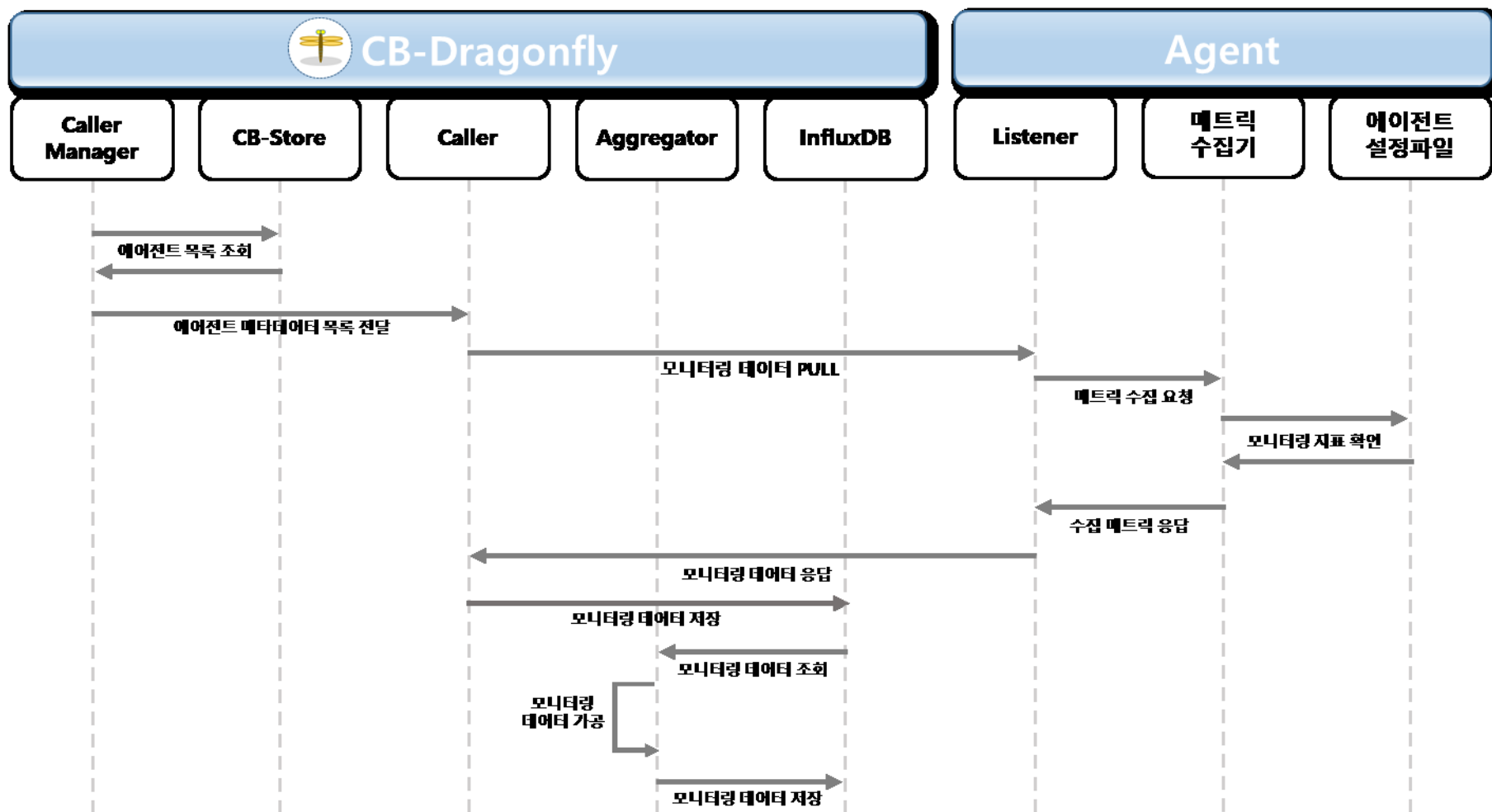
- 에이전트 모니터링 메커니즘 확장
  - PUSH  $\Rightarrow$  PUSH + **PULL**
  - REST API 기반 **Listener**를 통해 PUSH 모니터링 선택 및 변경 기능 지원
- Hetero-iaaS 환경의 통합 모니터링 메트릭 수집 기능 지원
  - MCIS 모니터링 메트릭, MCIS 성능 메트릭 등



# CB-Dragonfly 모니터링 방식 비교

	Agent - PUSH 방식 모니터링	Server - PULL 방식 모니터링
활용 목적	퍼블릭 네트워크 환경의 서버 기반 에이전트 모니터링	프라이빗 네트워크 환경의 서버 기반 에이전트 모니터링
활용 모듈	Collector Manager + Collector + Message Queue	Caller Manager + Caller
통신 프로토콜	MQTT	HTTP (REST API)
수집 모듈 스케일링	지원	미지원
에이전트 메인 모듈	PUSH 모듈	Listener 모듈
에이전트 상태 추적	필요	불필요

# CB-Dragonfly PULL 모니터링 상세 흐름도



# MCIS 그룹 모니터링 특화 메트릭 발굴

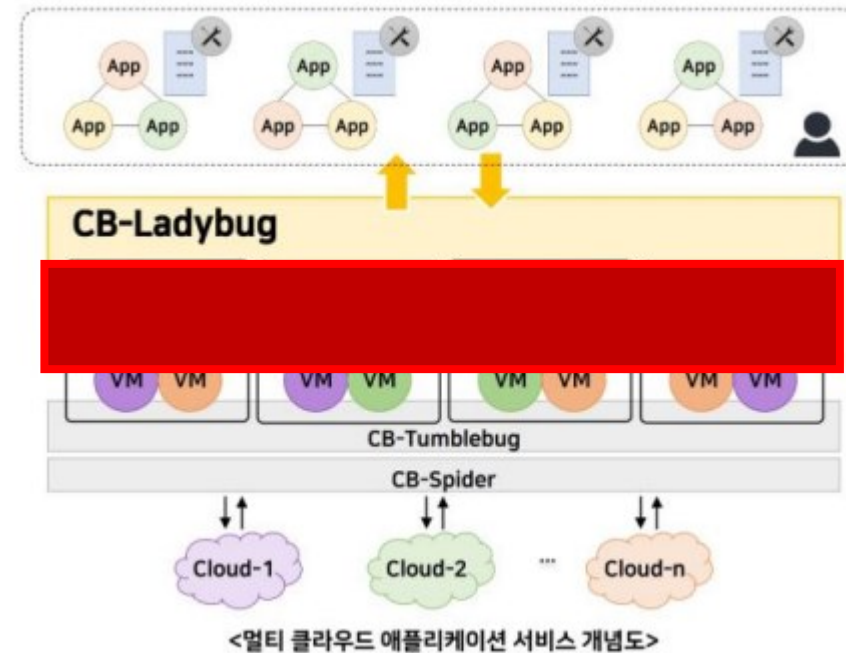
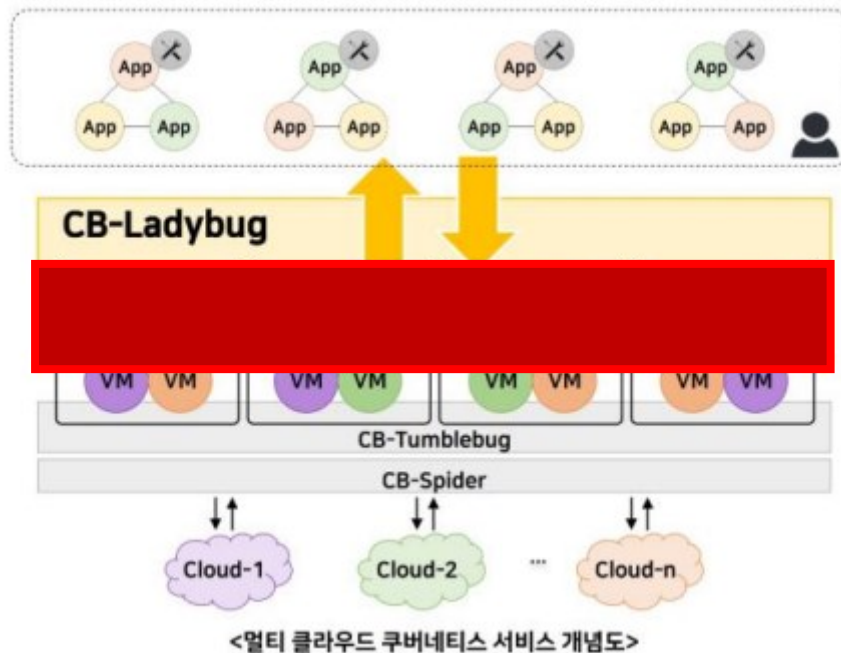
- 신규 MCIS 그룹 모니터링 특화 메트릭
  - MCIS 기반 그룹 모니터링 특화 메트릭 발굴 및 정의

항목	분류	모니터링 항목	비고
MCIS	Aggregate	그룹 단위의 리소스 사용량 총합	그룹 단위는 기본적으로 NS, MCIS 존재, 또는 추가적 단위로는 CSP 별, OS 별 존재 디스크 사용량 (byte), 네트워크 사용량 (packet) 등의 합계를 제공 해당 메트릭을 기준으로 어플리케이션 배포 등 리소스 사용량에 따라 MCIS 활용 가능
		NS 또는 MCIS 모니터링 사용률 오더링 (order by CPU, order by MEM)	최대 사용률의 NS 또는 MCIS 최소 사용률의 NS 또는 MCIS
		VM 컴퓨팅 스펙 지표 총합	NS 단위 또는 MCIS 단위 컴퓨팅 스펙 평균 또는 합산치 제공
	Network	TPS (Transaction per second)	MCIS 내의 VM 합산 또는 평균치의 TPS(agent 웹서버 기준) 값을 제공
		MCIS의 CB-Dragonfly 프레임워크 통신 네트워크 지표	모니터링 데이터 push 시 평균치의 네트워크 레이턴시 값 제공
		MRTT 추가 메트릭	MCIS 내의 VM Public IP 정보를 기반으로 서로 간의 RTT 평균치 계산
	Compute	VM 컴퓨팅 스펙 지표	NS 단위 또는 MCIS 단위 컴퓨팅 스펙 평균 또는 합산치 제공
	Process	MCIS 내 각 VM에서 구동중인 프로세스 수의 합	MCIS 내 VM의 프로세스 수의 총 합

# MCKS/MCAS 모니터링 개요

## • MCKS/MCAS 모니터링 대상

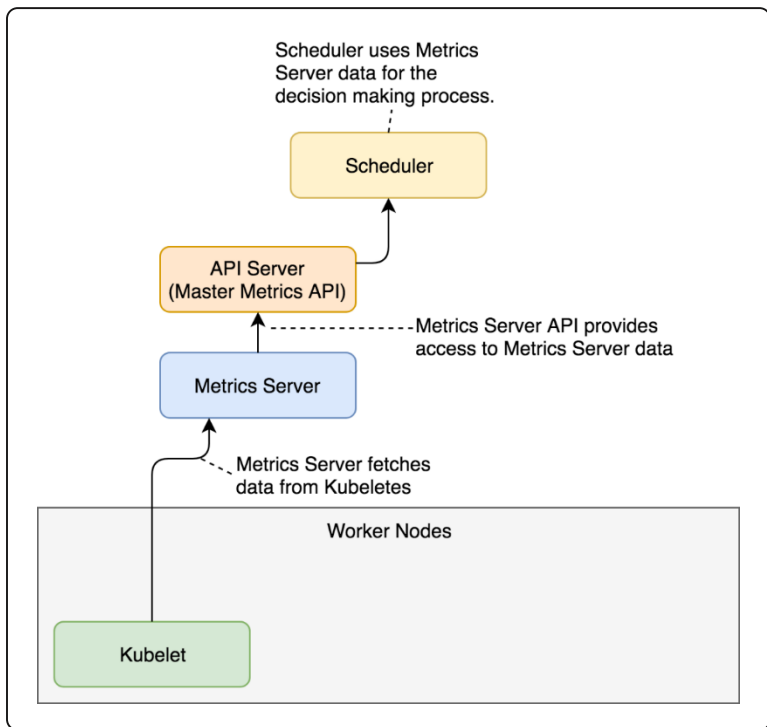
- CB-Ladybug FW는 Kubernetes 환경 기반의 MCKS 서비스 및 MCAS 서비스 제공
- CB-Dragonfly FW는 MCKS, MCAS 모니터링 서비스 제공 예정
- MCKS, MCAS 모니터링 PoC 진행



# MCKS/MCAS 모니터링 PoC 수행

- **Kubernetes 모니터링 – Metrics-Server**

- 노드 별 CPU, Memory 사용량 등
- Metrics-Server는 Deployment 리소스 형태로 Kubernetes 클러스터에 배포
- Metrics-Server가 설치되면, Kubernetes API를 통해서 모니터링 정보 조회 가능
- Node, Pod의 CPU, Memory 사용량을 캐싱
  - Node는 최근 1분 단위의 사용량, Pod는 최근 5분 단위의 사용량 조회



1. Kubernetes 노드 내의 kubelet 에서 pod 및 node 상태를 모니터링
2. 메트릭 서버에 해당 데이터를 주기적으로 전송
3. Kubernetes는 [metrics.k8s.io/v1beta1/nodes](https://kubernetes.io/docs/reference/generated/kube-api-docs/swagger-docs/operations/operation_1000_metrics_k8s_io_v1beta1_nodes_get), [metrics.k8s.io/v1beta1/pods](https://kubernetes.io/docs/reference/generated/kube-api-docs/swagger-docs/operations/operation_1000_metrics_k8s_io_v1beta1_pods_get) API 제공
4. Kubernetes 스케줄러는 Metrics Server의 CPU, Memory 사용률을 기준으로 오토스케일링

# CB-Dragonfly 프레임워크 로드맵





멀티클라우드 서비스 공통 플랫폼

# CB-Dragonfly기술 시연

- PULL 방식 모니터링 시연 -

카페모카(Café Mocha) 한잔 어떠세요 ?



# 감사합니다.

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

(김 효 경 / [contact-to-cloud-barista@googlegroups.com](mailto:contact-to-cloud-barista@googlegroups.com))

## “멀티클라우드, 글로벌 스케일로 시작하다”

클라우드바리스타들의 네번째 이야기

### Cloud-Barista Community the 4<sup>th</sup> Conference