



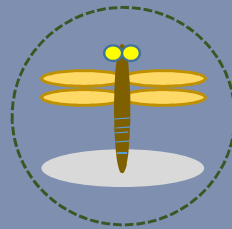
CLOUD

BARISTA

멀티 클라우드 서비스 공통 플랫폼

CB-Dragonfly : 전세계 클라우드가 다보여

(멀티 클라우드 통합 모니터링)



권 경 민 / CB-Dragonfly 프레임워크 리더

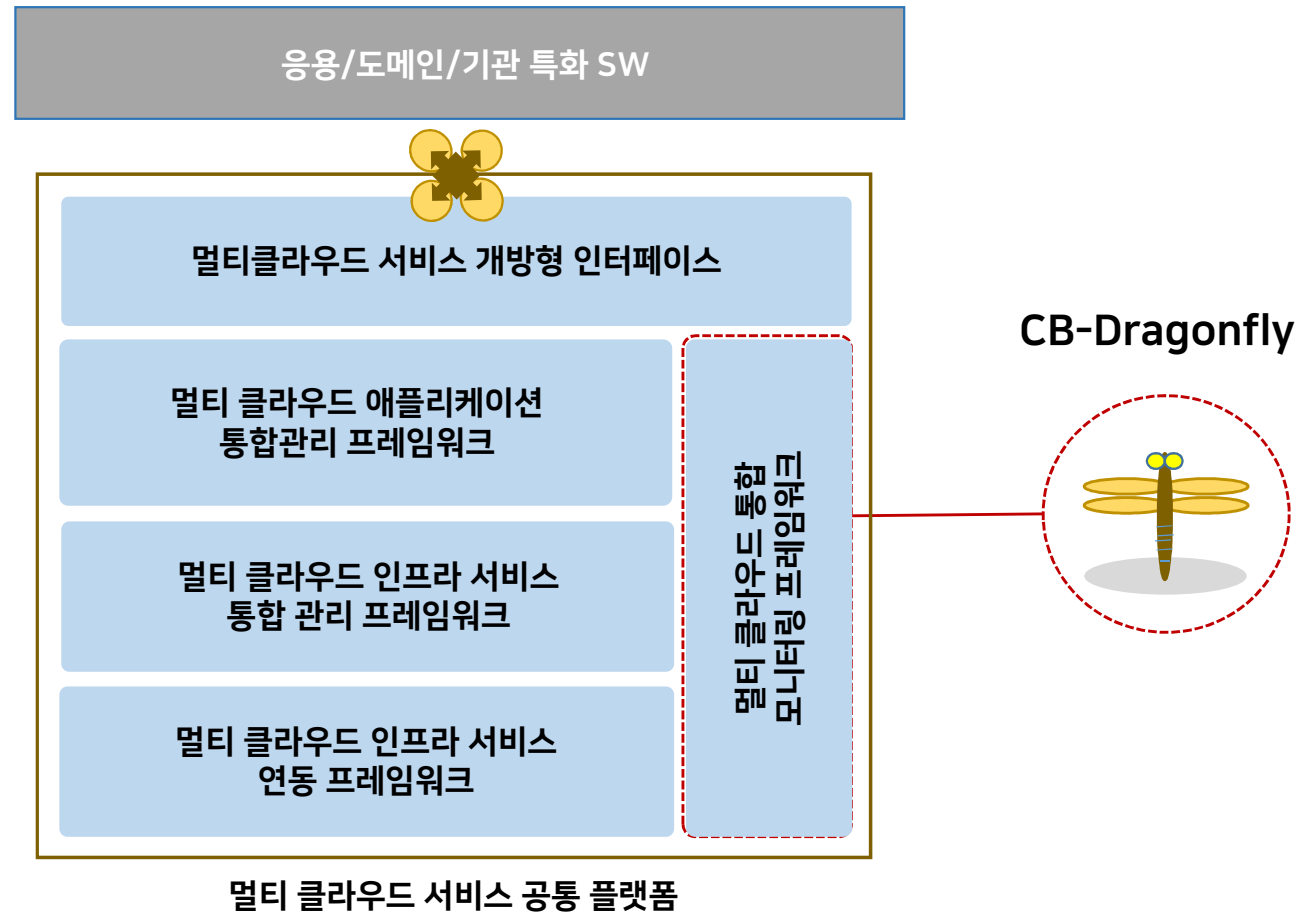
“Contact to the Multi-Cloud”

클라우드 바리스타들의 두 번째 이야기

Cloud-Barista Community 2nd Open Conference



이번 세션은...





목 차

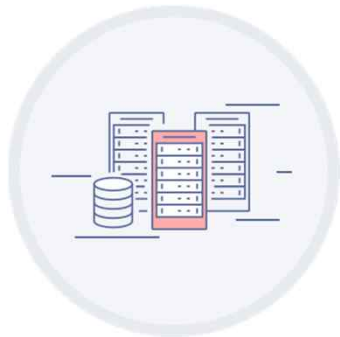
- I** CB-Dragonfly 프레임워크 기술 개요
- II** CB-Dragonfly 프레임워크 2019(Americano) 기능 요약
- III** CB-Dragonfly 프레임워크 2020(Cappuccino) 개발 현황
- IV** CB-Dragonfly 프레임워크 2020(Espresso) 개발 계획
- V** Load-Balancer 기반 대규모 모니터링 안정성 검증 시연



CB-Dragonfly : 멀티 클라우드 통합 모니터링 기술 개요

- 멀티 클라우드 **인프라 서비스 및 애플리케이션 통합 모니터링 기능을 제공하며 대규모 멀티 클라우드 모니터링을 지원하는 멀티 클라우드 통합 모니터링 프레임워크**

멀티 클라우드 통합 모니터링 특징점



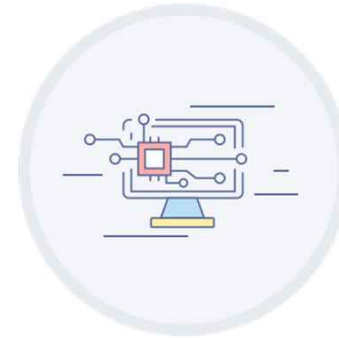
대규모 모니터링

모니터링 콜렉터 확장/축소 기능을 통하여 대용량 모니터링 데이터 수집/처리 가능



안정적인 모니터링

에이전트 기반으로 모니터링 데이터가 수집되며 동일한 메트릭 및 보존정책을 적용하여 안정적 모니터링 제공



특화 모니터링

멀티 클라우드 인프라 서비스 기반 특화 모니터링 발굴 및 수집/제공



CB-Dragonfly 프레임워크 아키텍처

Service &
Platform

웹 도구

통합 모니터링 프레임워크 API

3rd-Party



chronograf

CB-
Dragonfly

모니터링 콜렉터 매니저

모니터링 정책/알람 매니저

모니터링 에이전트

모니터링 콜렉터

모니터링 정책 설정/관리

모니터링 알람 설정/관리

모니터링 콜렉터 오토스케일

MCIS 모니터링

MC-APP 모니터링

대규모 모니터링

가상머신 모니터링

컨테이너 모니터링

온디멘드 모니터링

Hetero
MC-IaaS

VM Environment

Container Environment

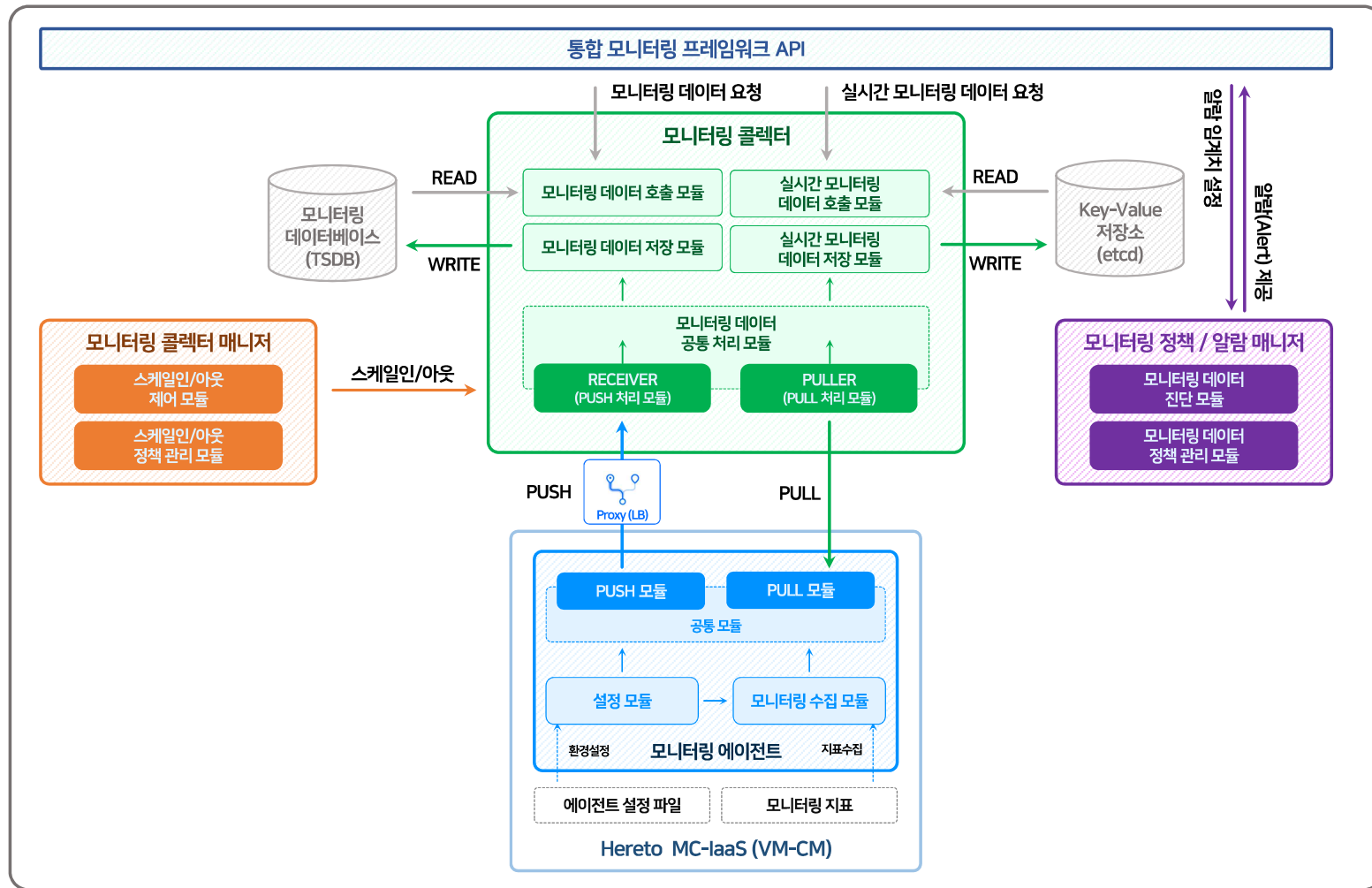


Multi-Cloud





CB-Dragonfly 모듈 구조





CB-Dragonfly 프레임워크 2019(Americano) 기능 요약

멀티 클라우드 인프라 서비스 통합 모니터링 프레임워크



통합 모니터링 메트릭 제공

멀티 클라우드 환경에서
에이전트 기반의 통합 모니터링
메트릭 제공



모니터링 콜렉터 확장 / 축소

Scalable & Stable 수집
서버 아키텍처를 통한 대용량
모니터링 제공



서버 성능 모니터링 및 실시간 모니터링

오픈소스 기반 가상머신의
서버 성능 모니터링 및
실시간 모니터링



모니터링 임계치 관리 및 알람

멀티 클라우드 인프라 서비스
임계치 관리 및 알람 기능



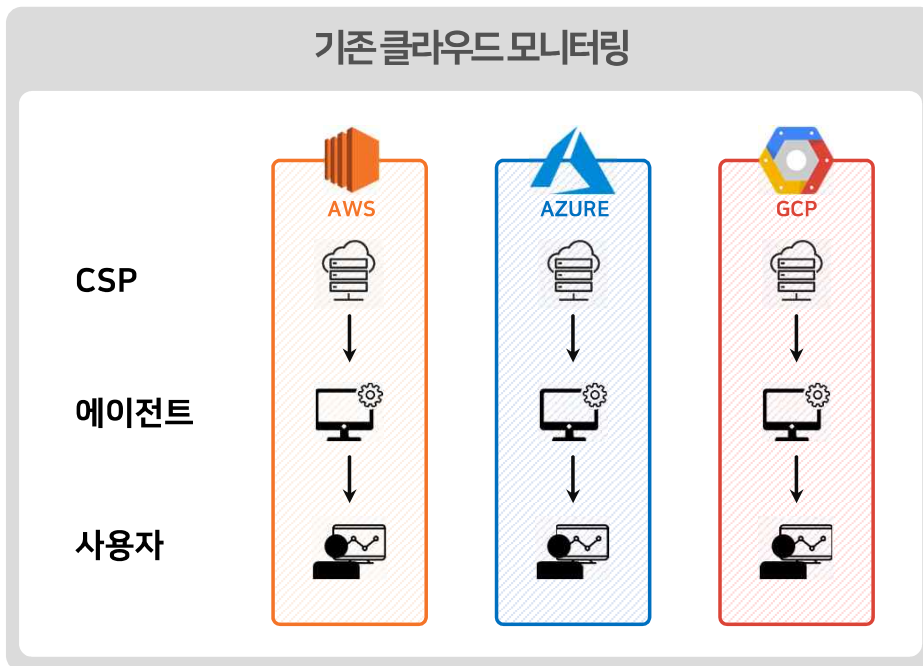
개방형 API 제공

멀티 클라우드 모니터링
개방형 API 제공

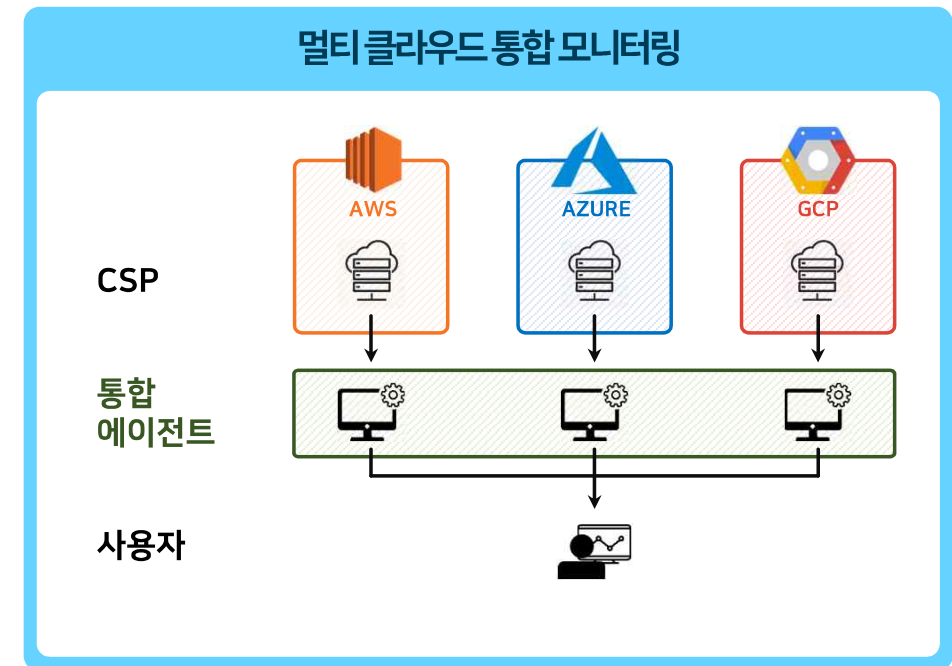


통합 모니터링

- 멀티 클라우드 통합 모니터링
 - 에이전트 기반의 멀티 클라우드 통합 모니터링 메트릭 제공



- 각 클라우드 CSP의 VM 모니터링 제공
- 클라우드 CSP 별도의 모니터링 메트릭 보관 정책 적용

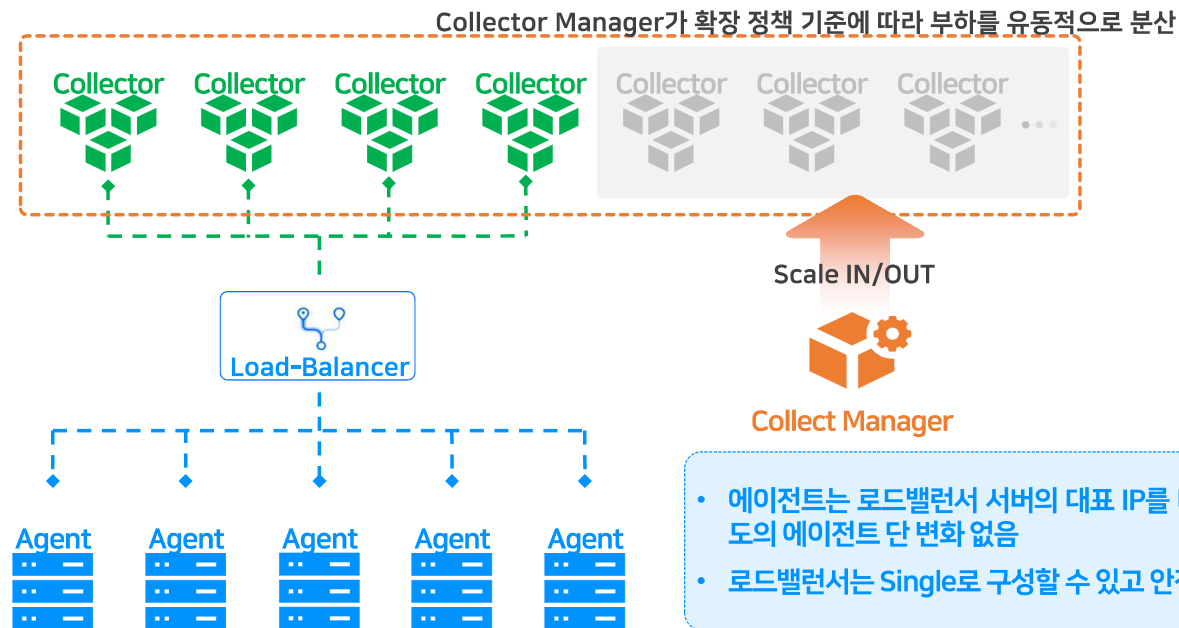


- 에이전트 기반의 멀티 클라우드 통합 모니터링 제공
- 통합 모니터링 메트릭 보관 정책 적용



확장성

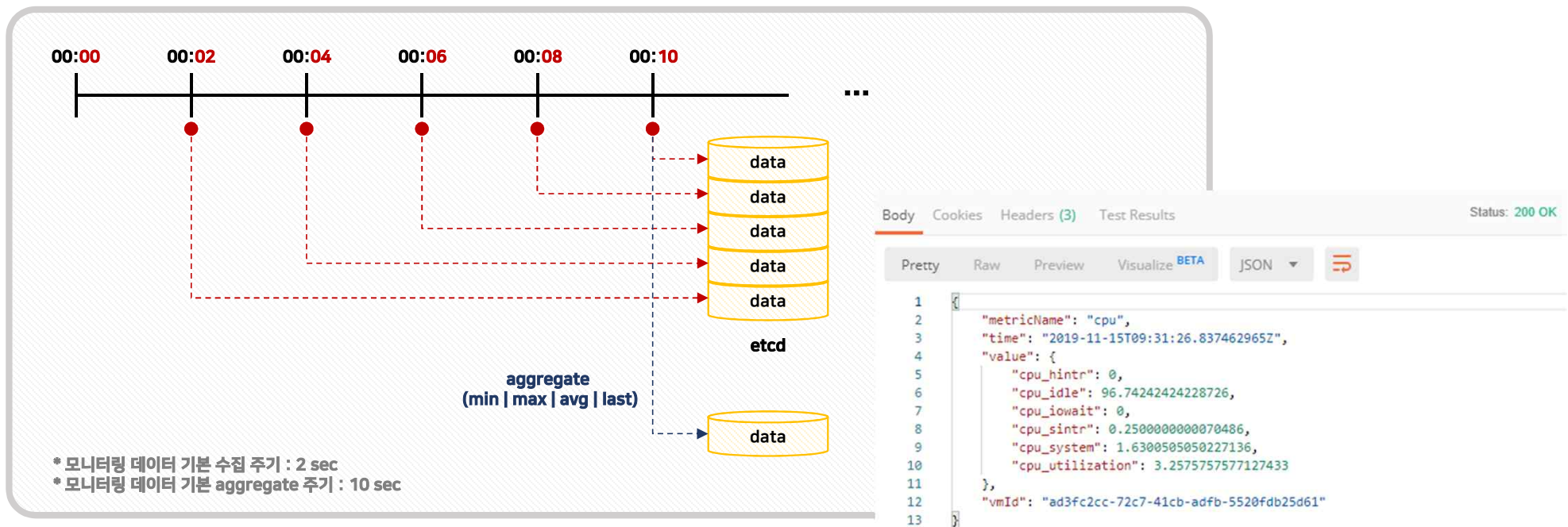
- 모니터링 콜렉터 확장/축소
 - 수많은 에이전트로부터의 대규모 모니터링 데이터를 Load-Balancer를 통해 부하 분산 처리
 - 확장 정책 기준에 따라 모니터링 콜렉터의 스케일을 조정
 - 가상머신 개수, CSP별 기준, CSP 지역
 - 대규모 멀티 클라우드 서비스 환경에서 모니터링을 안정적으로 제공





최신 데이터

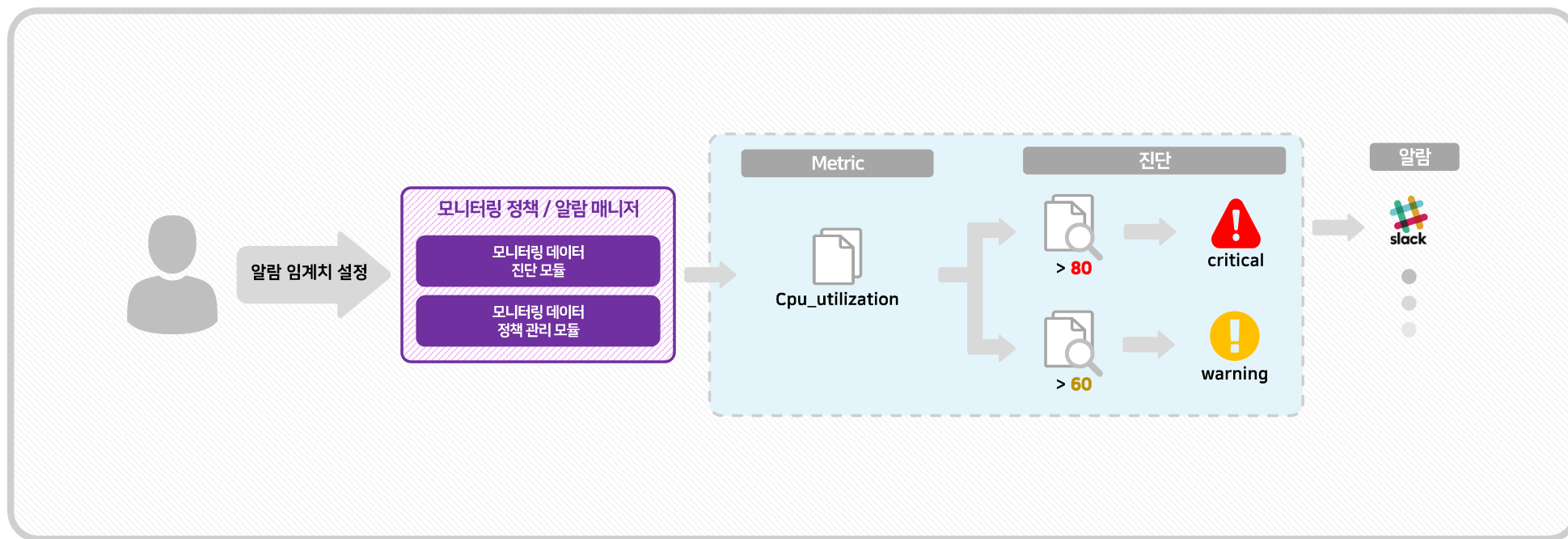
- 클라우드 인프라 서비스 성능 모니터링 및 최신 데이터 모니터링
 - 에이전트에서 기본 2초 단위로 가상 머신 모니터링 데이터 수집
 - 2초 주기의 데이터 기반 statistics criteria(min, max, avg, last) 최신 데이터 모니터링 데이터 조회
 - etcd 저장소에 저장된 최신 데이터 aggregation 및 조회





정책 설정 및 알람

- 멀티 클라우드 인프라 서비스 임계치 관리 및 알람 기능
 - 모니터링 알람 정책 설정 및 알람 임계치 진단
 - 연동된 이벤트 핸들러 기준 email, slack 등으로 알림 지원





개방형 API

• 개방형 API 제공

- CB-Dragonfly의 주요 기능들을 개방형 API를 통해 제공

- API 예시:

`http://{{api-server}}:9090/mon/mcis/:mcisID/vm/:vmID/metric/:metricName/info?periodType=m&statisticsCriteria=avg&duration=5m`

- API 상세 정보:

개방형 API 파라미터	설명
mcisID	MCIS (Multi Cloud Infra Service) ID
vmID	가상머신 ID
metricName	모니터링 메트릭 (cpu, memory, disk, network)
periodType	모니터링 단위 ("m" "h" "d")
statisticsCriteria	모니터링 통계 기준 ("min" "max" "avg" "last")
duration	모니터링 조회 범위 (now() - duration), 모니터링 단위 ("m" "h" "d")

```

Body  Cookies  Headers (3)  Test Results  Status: 200 OK
Pretty  Raw  Preview  Visualize BETA  JSON  ↩
1  {
2    "name": "cpu",
3    "tags": {
4      "hostId": "99345fbc-7b20-4d5b-88bd-174691e62ef7"
5    },
6    "columns": [
7      "time",
8      "cpu_utilization",
9      "cpu_system",
10     "cpu_idle",
11     "cpu_iowait",
12     "cpu_hintr",
13     "cpu_sintr"
14   ],
15   "values": [
16     [
17       "2019-11-15T07:30:00Z",
18       2.2481936756893695,
19       1.0113040988651163,
20       97.75180632431064,
21       0.14204307939700625,
22       0,
23       0.05007306020703197
24     ],
25     [
26       "2019-11-15T07:31:00Z",
27       2.2507083865598125,
28       1.0266403046302546,
29       97.74929161344018,
30       0.15768089297013793,
31       0,
32       0.03547147489211024
33     ]
34   ]
35 }

```



CB-Dragonfly 프레임워크 2020(Cappuccino) 개발 현황

- 모니터링 에이전트 확장
 - 리눅스 계열 운영체제 확장 (Americano 3종, Cappuccino 6종)
 - 리눅스 계열 에이전트 수집 메트릭 추가 (Americano 24개, Cappuccino 52개)
 - 윈도우 계열 에이전트 지원
- 대규모 모니터링 안정성 검증
 - 대규모 모니터링 안정성 시험 방안
 - 대규모 모니터링 안정성 시험 결과
 - Load-balancer 기반 대규모 모니터링 안정성 시험 결과
- 모니터링 신규 기능 PoC 테스트
 - 라즈베리파이 환경 구동 테스트
 - InfluxDB-Relay 오픈소스 설치 및 구동 테스트
 - 온디멘드 모니터링 구동 테스트



모니터링 에이전트 확장

- 리눅스 계열 지원 운영체제 확장 → 총 6종 지원
 - 2019(Americano) 3종 지원: CentOS, FreeBSD, Debian
 - 2020(Cappuccino) 3종 추가: Ubuntu, Fedora, Slackware

Americano 버전 모니터링 에이전트 지원 운영체제



리눅스 운영체제 3종 지원

Cappuccino 버전 모니터링 에이전트 지원 운영체제



리눅스 운영체제 6종 지원



모니터링 에이전트 확장

• 리눅스 계열 에이전트 추가 메트릭 개발

[2019(Americano) 메트릭 목록]

제공 릴리즈	분류	모니터링 항목	개수
Americano	CPU	cpu_utilization, cpu_system, cpu_idle, cpu_iowait, cpu_hintr, cpu_sintr	6개
	Memory	mem_utilization, mem_total, mem_used, mem_free, mem_shared ...	6개
	Disk	disk_utilization, disk_total, disk_used, ops_read, kb_read/written ...	8개
	Network	bytes_in, bytes_out, pkts_in, pkts_out	4개

총 24개 메트릭 제공

[2020(Cappuccino) 버전 추가 메트릭 목록]

제공 릴리즈	분류	모니터링 항목	개수
Cappuccino	CPU	cpu_usr, cpu_num, uptime, proc_run, proc_num, load_avg_1/5/10m ...	13개
	Memory	swap_utilization, swap_used, swap_free, swap_in/out, page_in/out ...	8개
	Disk	read_time, write_time ...	2개
	Network	errors_in, errors_out, drops_in, drops_out ...	4개
	CpuFreq	cpu_speed(frequency)	1개

총 52개 메트릭 제공

메트릭 추가 정의

- ✓ Cpu
- ✓ Memory
- ✓ Disk
- ✓ Network
- ✓ Cpufreq



모니터링 에이전트 확장

- 윈도우 계열 에이전트 지원
 - 모니터링 에이전트는 리눅스/윈도우 운영체제와 무관하게 동일한 메트릭 항목 수집
 - CPU, Mem, Disk, Network 동일한 메트릭 항목 수집
 - 모니터링 메트릭 수집 시 MCIS_ID, HOST_ID, OS_TYPE (linux,windows) 정보 태깅
- 윈도우 환경 모니터링 에이전트 구동 화면

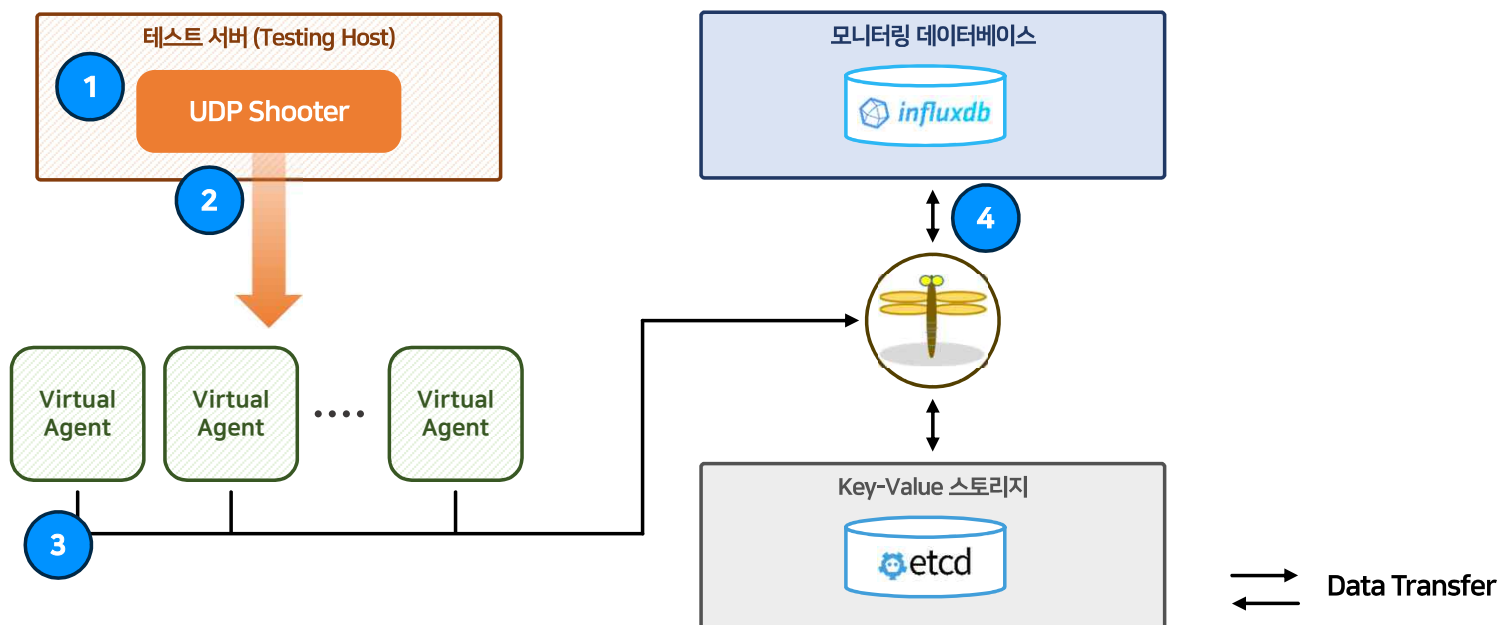
```
Windows PowerShell
PS C:\Users\Wadmin\Downloads\wcbmon\Wagent>
PS C:\Users\Wadmin\Downloads\wcbmon\Wagent> .\Wagent.exe --config .\Wagent.conf
2020-05-21T21:14:45Z !! Starting Telegraf 1.13.4
2020-05-21T21:14:45Z !! Loaded inputs: disk diskio mem swap win_perf_counters cpu
2020-05-21T21:14:45Z !! Loaded aggregators:
2020-05-21T21:14:45Z !! Loaded processors:
2020-05-21T21:14:45Z !! Loaded outputs: influxdb
2020-05-21T21:14:45Z !! Tags enabled: host=DESKTOP-3NG2Q22
2020-05-21T21:14:45Z !! [agent] Config: Interval:10s, Quiet:false, Hostname:"DESKTOP-3NG2Q22", Flush Interval:10s
```




대규모 모니터링 성능 안정성 검증

• 대규모 모니터링 안정성 시험 방안

1. 대규모 환경 검증을 위해 UDP-Shooter 모듈 별도 개발
2. UDP-Shooter를 통해 고루틴 기반 가상 에이전트 구현 (고루틴 1개 = 가상머신 모니터링 에이전트 1개)
3. 가상 에이전트를 통해 모니터링 데이터 전송
4. CB-Dragonfly 모듈은 가상 에이전트를 통해 전송된 모니터링 메트릭 데이터 처리 및 저장





대규모 모니터링 성능 안정성 검증

• 대규모 모니터링 안정성 시험 결과

성능 지표 / 에이전트 수	50	100	200	300	400	600	800	1000
cpu_utilization	0.6	0.8	1.56	2.75	3.19	3.8	4.3	5.7
cpu_load_avg_1	0.01	0.01	0.02	0.54	1.19	1.52	1.63	1.78
cpu_load_avg_5	0.04	0.05	0.30	0.29	0.55	1.15	1.30	1.40
cpu_load_avg_15	0.08	0.12	0.29	0.19	0.27	0.73	1.09	1.12
mem_utilization	105M /30.6G	121M /30.6G	230M /30.6G	440M /30.6G	483M /30.6G	579M /30.6G	1.37G /30.6G	1.72G /30.6G
read_bytes	0M	0M	0M	0M	0M	0M	0M	0M
wrtie_bytes	20.2M	66.7M	83.3M	120M	202M	292M	379M	475M
aggregate_time	54.69ms	92.70ms	173.44ms	212.90ms	255.02ms	310.53ms	330.01ms	410.20ms



대규모 모니터링 성능 안정성 검증

• Load-balancer 기반 대규모 모니터링 안정성 시험 결과

- Ubuntu 18.04, 8 Core VCPU 환경에서 UDP-Shooter, CB-Dragonfly 모듈 설치 후 테스트 완료
- Load-balancer 적용 시 **23.7%** aggregate 시간 감소
- 추후 콜렉터를 고루틴이 아닌 컨테이너 또는 VM 단위의 확장성을 고려하여 Load-balancer 기반 CB-dragonfly 개발

성능 지표 / 에이전트 수	1	10	20	30	40	60	80	100
cpu_utilization	0.1	0.1	0.1	0.2	0.4	0.4	0.6	0.7
cpu_load_avg_1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
cpu_load_avg_5	0.01	0.01	0.01	0.02	0.04	0.04	0.05	0.05
cpu_load_avg_15	0.01	0.01	0.02	0.05	0.07	0.09	0.10	0.11
mem_utilization	1.7M	20M	37M	58M	87M	109M	116M	121M
read_bytes	0M	0M	0M	0M	0M	0M	0M	0M
wrtie_bytes	4.2M	8.22M	12.45M	28.2M	36.6M	53.3M	62.9M	95.2M
collector_count	1	1	2	3	4	6	8	10
aggregate_time	1.23ms	6.23ms	11.59ms	23.32ms	32.95ms	49.23ms	61.42ms	70.73ms



모니터링 신규 기능 PoC 테스트

- 라즈베리 파이 OS(Raspbian) 환경 구동 테스트
 - Raspbian OS Kernel version : 4.19 (Release data : 2020-02-12)
 - GitHub repository에 등록된 CB-Dragonfly 모듈 Check-out 및 구동하여 정상 동작 확인
- 라즈베리 파이 OS에서 정상 동작 화면

```

pi@raspberrypi: ~
[GitHub - cloud-barist...

File Edit Tabs Help

root@raspberrypi:/home/pi/cb-dragonfly# go run pkg/manager/main/main.go

v4.1.10
High performance, minimalist Go web framework
https://echo.labstack.com

0/
0\

= http server started on [::]:9090
  
```

< CB-Dragonfly 구동 >

```

root@5c0da31b75c9: /
[GitHub - cloud-barist...

File Edit Tabs Help

root@raspberrypi:/home/pi/cb-dragonfly# docker run -it --name cb-dragonfly -p 8094:8094 -p 9090:9090 tesolor

v4.1.10
High performance, minimalist Go web framework
https://echo.labstack.com

0/
0\

= http server started on [::]:9090

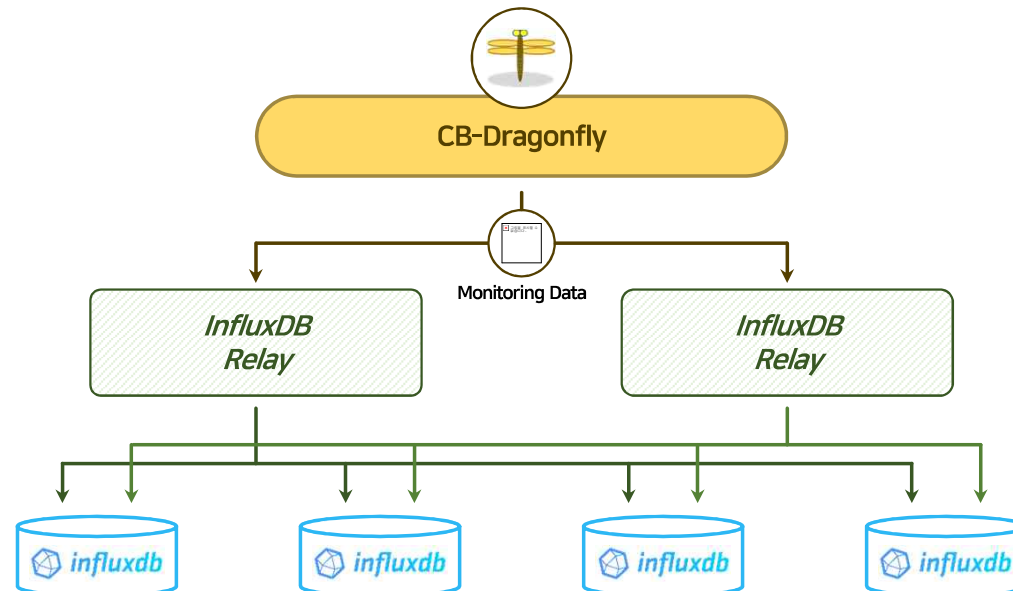
root@raspberrypi:/home/pi/cb-dragonfly# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS
888f8f950fc3   tesolor   "./runMyapp"            3 seconds ago Up 2 seconds
0.0.0.0:8094->8094/tcp, 8094/udp, 0.0.0.0:9090->9090/tcp   cb-dragonfly
  
```

< CB-Dragonfly Dockerizing 및 Docker 구동 >



모니터링 신규 기능 PoC 테스트

- InfluxDB-Relay 오픈소스 설치 및 구동 테스트
 - InfluxDB HA(High Availability) 구성을 위해 InfluxDB-Relay 설치 및 구동 PoC 테스트
 - InfluxDB Official 기반의 Open Source(Strike-team git repository) 활용
 - InfluxDB-Relay를 적용한 CB-Dragonfly 아키텍처





모니터링 신규 기능 PoC 테스트

- 온디멘드 모니터링 구동 테스트
 - 사용자가 모니터링 데이터를 요청한 시점에 실시간 모니터링 데이터 수집
 - API 호출 시 실시간 모니터링 데이터 수집 및 JSON 형태로 데이터 정보 전달
 - 온디멘드 모니터링 API 호출 화면

```
{
  "cpu": {
    "usage_guest": 0,
    "usage_guest_nice": 0,
    "usage_idle": 83.58585858585644,
    "usage_iowait": 1.767676767676938,
    "usage_irq": 0,
    "usage_nice": 0,
    "usage_softirq": 0.2525252525252448,
    "usage_steal": 0,
    "usage_system": 4.5454545454546755,
    "usage_user": 9.848484848484414
  },
  "disk:sda3": {
    "free": 530554880,
    "inodes_free": 0,
    "inodes_total": 0,
    "inodes_used": 0,
    "total": 536854528,
    "used": 6299648,
    "used_percent": 1.173436689352092
  },
  "disk:sda5": {
    "free": 166255407104,
    "inodes_free": 13518900,
    "inodes_total": 14704640,
    "inodes_used": 1185740,
    "total": 236766593024,
    "used": 60358234112,
    "used_percent": 26.634863544895204
  },
  "diskio:loop0": {
    "io_time": 0,
    "iops_in_progress": 0,
    "merged_reads": 0,
    "merged_writes": 0,
    "read_bytes": 123904,
    "read_time": 60,
    "reads": 46,
    "weighted_io_time": 0,
    "write_bytes": 0,
    "write_time": 0,
    "writes": 0
  },
  "diskio:loop1": {
    "io_time": 12,
    "iops_in_progress": 0,
    "merged_reads": 0,
    "merged_writes": 0,
    "read_bytes": 1148928,
    "read_time": 84,
    "reads": 110,
    "weighted_io_time": 48,
    "write_bytes": 0,
    "write_time": 0,
    "writes": 0
  },
  "diskio:loop10": {
    "io_time": 0,
    "iops_in_progress": 0,
    "merged_reads": 0,
    "merged_writes": 0,
    "read_bytes": 47104,
    "read_time": 96,
    "reads": 19,
    "weighted_io_time": 0,
    "write_bytes": 0,
    "write_time": 0,
    "writes": 0
  },
  "diskio:loop11": {
    "io_time": 0,
    "iops_in_progress": 0,
    "merged_reads": 0,
    "merged_writes": 0,
    "read_bytes": 116736,
    "read_time": 112,
    "reads": 39,
    "weighted_io_time": 0,
    "write_bytes": 0,
    "write_time": 0,
    "writes": 0
  },
  "diskio:loop12": {
    "io_time": 0,
    "iops_in_progress": 0,
    "merged_reads": 0,
    "merged_writes": 0,
    "read_bytes": 0,
    "read_time": 0,
    "reads": 0,
    "weighted_io_time": 0,
    "write_bytes": 0,
    "write_time": 0,
    "writes": 0
  }
}
```



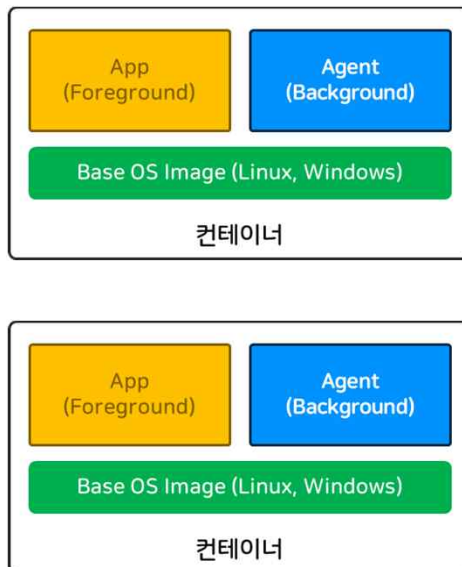
CB-Dragonfly 프레임워크 2020(Espresso) 개발 계획

- 컨테이너 모니터링 기능 개발
 - 모니터링 수집 에이전트 Dockerize
 - 컨테이너 기반 에이전트 배포 기능
 - 컨테이너 환경 모니터링 정보 수집 기능
- CLI 기반 모니터링 모듈 관리 도구 제공
 - cbmon 명령어를 통해 CB-Dragonfly 모듈의 CLI 기반 사용자 인터페이스 제공
- MCIS 모니터링 기술 개발
- MC-APP 모니터링 기술 개발
- 모니터링 알람 모듈 개발
- 모니터링 확장 기능 개발 (CSP 별 모니터링 콜렉터 확장 정책 개발)



컨테이너 모니터링

- 컨테이너 환경의 모니터링
- 사이드카 패턴을 활용해 컨테이너 내부의 모니터링 메트릭 수집
- Hetero MC-IaaS 환경에서 동일한 모니터링 항목을 수집



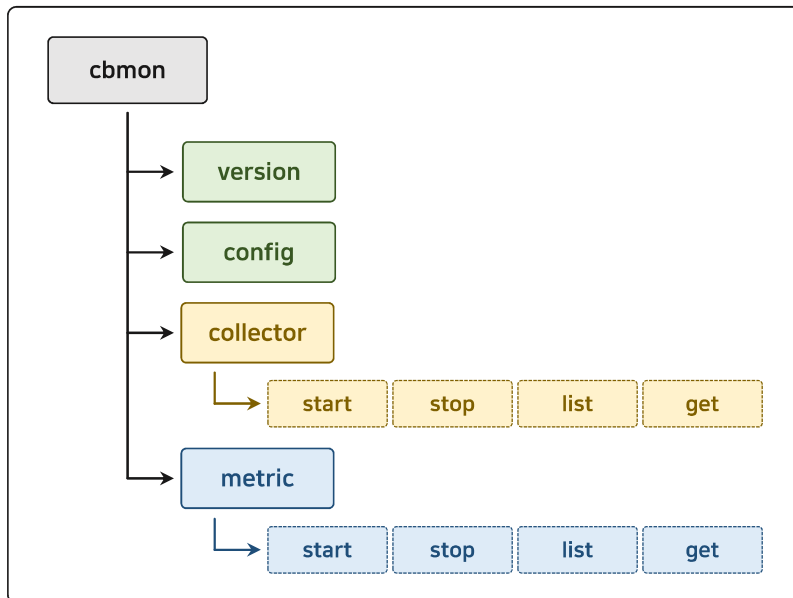
모니터링 항목	수집 데이터	컨테이너 OS 이미지
CPU	Speed, Utilization, Num ...	Linux(Ubuntu, Centos) Windows
Memory	Used_Percent, Total, Free ...	
Disk	Used_Percent, Total, Free ...	
Network	Packet Errors[In/Out], Packet Drops[In/Out] ...	



사용자 모니터링 CLI(cb-mon)

- CLI 환경 기반 모니터링 모듈 관리 도구 제공
 - cbmon을 통해 모니터링 모듈 가동/중지, 메트릭 정보 조회 등의 기능 제공

\$ cbmon



기능 목록	루트 명령어	명령어	설명
시스템 관리	-	• version	cbmon 버전 정보 출력
		• config	cbmon 환경 정보 설정 및 출력
콜렉터 관리	collector	• start • stop • list • get	콜렉터 시작/중지 및 콜렉터 정보 출력
메트릭 조회	metric	• list • get	모니터링 메트릭 정보 출력



사용자 모니터링 CLI(cb-mon)

- CLI 환경 기반 모니터링 모듈 관리 도구 제공

```
CLOUD-BARISTA
CB-MON

CB-Dragonfly CLI Tools for Cloud-Barista

Usage:
  cbmon [command]

Available Commands:
  help      Help about any command
  version

Flags:
  -h, --help      help for cbmon
  --verbose       Enable verbose logging

Additional help topics:
  cbmon collector
  cbmon config
  cbmon metric

Use "cbmon [command] --help" for more information about a command.
```



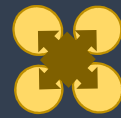
CB-Dragonfly 2020년도 개발 계획

서브 시스템명	주요 업무	상세 업무	수행 내용
멀티 클라우드 통합 모니터링 프레임워크	기 구현 기능의 개선 및 확장 개발	2019(Americano) 결과물의 검증 및 이슈 정리	- 라즈베리파이 상에서의 동작 여부 확인 및 이슈 해결 방안 정리 - 컨테이너 기반 CB-Dragonfly 이미지 빌딩 및 시험, 이슈 정리 - 공개SW의 활용 현황 검토 및 향후 활용 정책 수립(w/ETRI)
		에이전트 기능 개선 및 추가 기능 개발	- CB-Spider API를 활용한 에이전트 설치 기능 개발(Secure Shell 사용) - 리눅스 계열 에이전트 호환성 시험 및 개발 - 윈도우 계열 에이전트 개발/지원
		대규모 모니터링 성능 안정성 검증	- 대규모 모니터링 안정성 시험 방안 수립 (@Cloud-Twin 또는 실환경) - 대규모 모니터링 안정성 시험(스트레스->CloudTwin->실환경 기반)
		최신 데이터 모니터링 기능 개발	- 실시간 모니터링 관련 개념 정의 - 실시간 모니터링 기능 개발
		모니터링 정책 관리 모듈 개발	- 모니터링 수집, 알람, 스케줄러 주기 관리 기능 개발
		모니터링 알람 모듈 개발	- VM/Container 모니터링 알람 매트릭 개발 - 모니터링 알람 임계치 등록, 수정, 삭제 기능 개발 - 모니터링 설정 주기 내 알람 횟수에 따른 이벤트 발생 기능 개발 - 알람 생성, 수정, 삭제 API 제공 - 알람 이벤트 핸들러(slack, email 등) 등록, 수정, 삭제 API 제공
		모니터링 콜렉터 정책 확장 개발 (CSP별)	- 모니터링 매니저의 콜렉터 확장 정책 도출 및 개발
	신규 모니터링 방식	PULL 방식 구조 및 기능 개발	- PULL 방식 모니터링 설계 - PULL-Caller 모듈 개발 - PULL-Listener 모듈 개발 - 모니터링 콜렉터 및 Agent와의 통합 - PUSH / PULL의 운용 정책 정의 - PUSH / PULL의 선택적 운용 모듈 개발
		온디맨드 모니터링 개발 (w/CB-Tumblebug)	- 온디맨드 모니터링의 구조 및 기능 설계 (Tumblebug 주관) - 업무 배분에 따른 담당 부분의 개발
	멀티 클라우드 인프라 특화 모니터링	MCIS 모니터링 기술 개발	- MCIS 기반 모니터링 매트릭 (그룹 모니터링 매트릭) 발굴 및 정의 - MCIS 모니터링 매트릭 수집 기능 개발
		VM/Container 혼용 인프라 모니터링 기술 개발	- 컨테이너 인프라의 모니터링 구조 설계 - 컨테이너 인프라 모니터링 기능 개발
	멀티 클라우드 애플리케이션 서비스 모니터링	애플리케이션 모니터링 매트릭 정의	- 애플리케이션 모니터링 매트릭 도출 - 애플리케이션 모니터링 매트릭의 수집 PoC 검증
		애플리케이션 모니터링 구조 설계 및 기능 개발	- 애플리케이션 모니터링을 위한 구조 및 기능 설계 - 애플리케이션 모니터링 기능 개발 및 검증
	모니터링 개방형 API 도구	사용자 모니터링 CLI(cbmon) 도구 개발	- cbmon 제공 기능 범위 및 기능 설계 - cbmon 기능 개발



CB-Dragonfly 프레임워크 기술 로드맵





CLOUD

BARISTA

멀티 클라우드 서비스 공통 플랫폼

Load-Balancer 기반 대규모 모니터링 안정성 검증 시연

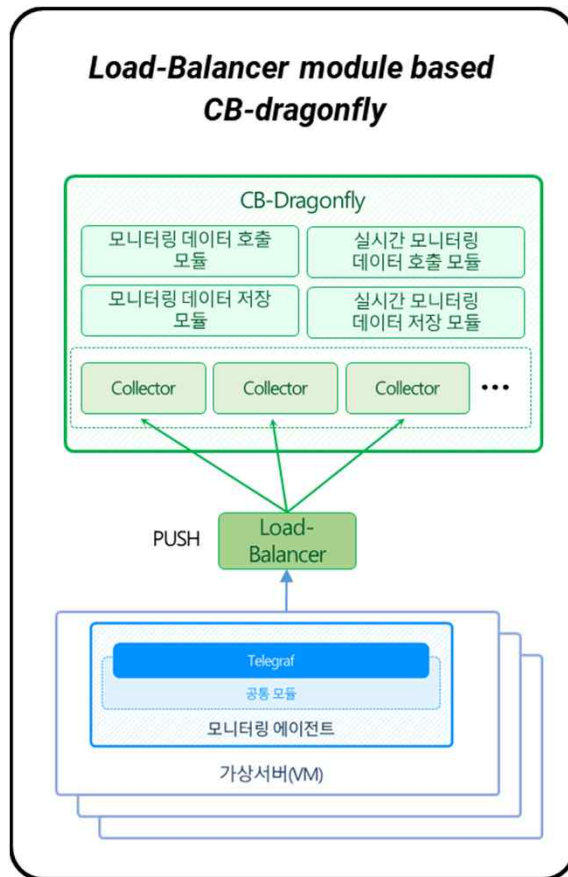
카푸치노(Cappuccino) 한잔 어떠세요 ? ^^

카푸치노(Cappuccino) : Cloud-Barista의 2nd 소스코드 버전명칭



Load-Balancer 기반 대규모 모니터링 안정성 검증 시연

• Load-Balancer 모듈을 적용한 CB-Dragonfly 데모 시연



기능	구분	추가 및 변경점
Load Balancing	추가	<ul style="list-style-type: none"> Load-Balancer가 Tagging할 VM을 Collector에게 분배
VM Tagging	변경	<ul style="list-style-type: none"> Collector에서 Tagging 되어있지 않은 VM들을 선점 (ConCurrency Dead-Lock issue) => 기능 제거 <u>Aggregating</u> 시점마다 Collector에 Tagging된 <u>VM 리스트</u> Untagging => Load-Balancer에 연결된 VM의 TTL을 설정하여 <u>Untagging</u> 시점 제어
Auto Scaling	변경	<ul style="list-style-type: none"> Scaling 검사 시점 때 <u>현재 Collector 내 Tagging 된 VM의 총 개수</u>를 기준으로 Scaling 여부 결정 => <u>Load-Balancer에 연결된 VM 총 개수</u>를 기준으로 Scaling 여부 결정



Load-Balancer 기반 대규모 모니터링 안정성 검증 시연

- Demo 개요

- Load-Balancer 기반 CB-Dragonfly 모듈 2020(Cappuccino) 형상 기준 Demo
- 고루틴 기반 콜렉터
- VM 수 기준 콜렉터 Scale-In/Out 스케줄링

- Demo 시나리오

- UDP-Shooter 실행을 통해 에이전트와 유사한 UDP Packet들을 CB-Dragonfly로 전송
- UDP-Shooter 의 데이터 전송 주기는 2초로 설정, 에이전트 수를 증가시켜가며 Demo 진행
- UDP Stress(에이전트 수) 에 따라 콜렉터 Scale-in/out 확인
- Load-Balancer의 VM ID TTL(Time To Live) 기능 동작 확인



시연

- Load-Balancer 모듈을 적용한 CB-Dragonfly 데모 시연
 - [동영상 재생](#)

감사합니다.

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

(권 경 민 / contact-to-cloud-barista@googlegroups.com)

“Contact to the Multi-Cloud”

클라우드 바리스타들의 두 번째 이야기

Cloud-Barista Community 2nd Open Conference