



멀티클라우드, 글로벌 스케일로 시작하다

클라우드바리스타 커뮤니티 제4차 컨퍼런스

[세션3] CB-Spider / CB-Tumblebug : 멀티클라우드 인프라 서비스

- | 김 병 섭 기술총괄/CB-Spider 프레임워크 리더
- | 손 석 호 CB-Tumblebug 프레임워크 리더

카페모카(Café Mocha) 한잔 어떠세요 ?

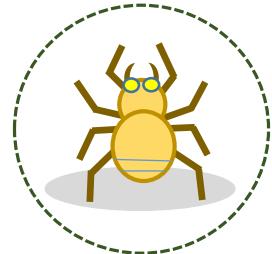


멀티클라우드, “글로벌 스케일로 시작하다”

클라우드바리스타 커뮤니티 제4차 컨퍼런스

CB-Spider :

멀티클라우드 인프라 연동



김 병 섭 / Cloud-Barista 기술총괄, CB-Spider 프레임워크 리더

카페모카(Café Mocha) 한잔 어떠세요 ?



이번 세션은...

응용/도메인/기관 특화 SW



멀티클라우드 서비스 개방형 인터페이스

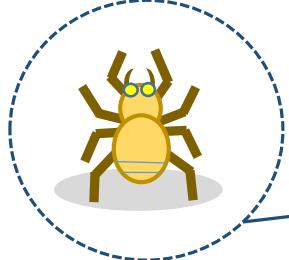
멀티클라우드 애플리케이션
통합관리 프레임워크

멀티클라우드 인프라 서비스
통합 관리 프레임워크

멀티클라우드 인프라 연동
프레임워크

도입 및 확장
비용 절감
구조화된
데이터 관리

CB-Spider



멀티클라우드 서비스 공통 플랫폼

목차

I CB-Spider 기술 개요

II CB-Spider 기능 및 인터페이스

III CB-Spider 개발 로드맵

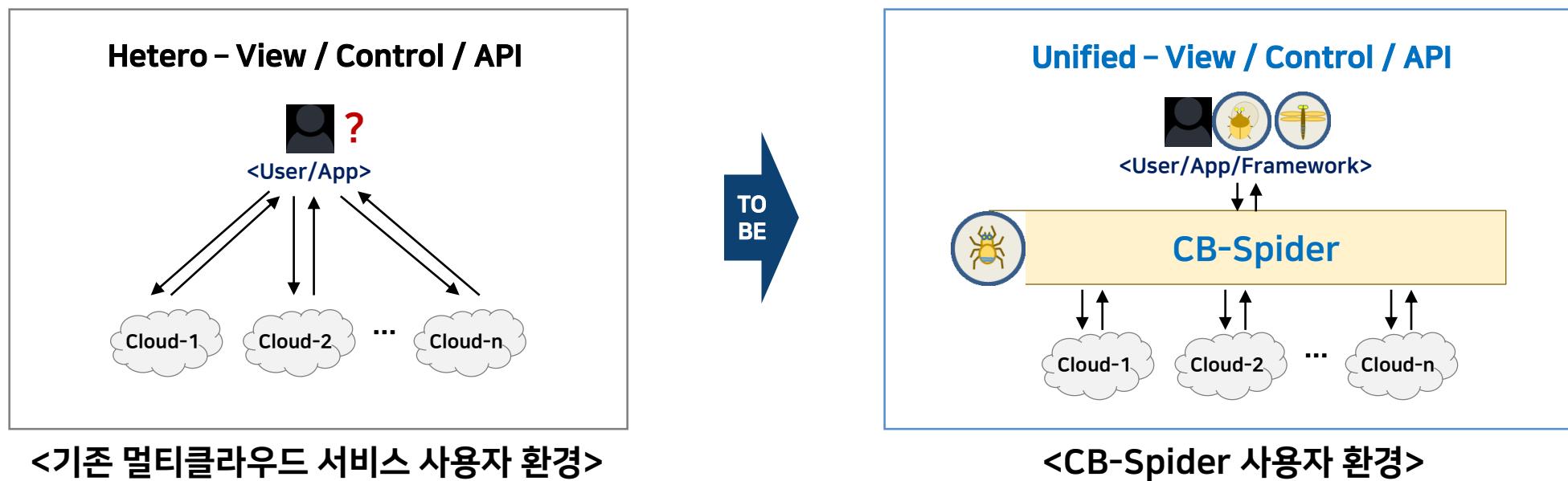
IV CB-Spider 선행 PoC

컴퓨팅 인프라 환경 변화와 CB-Spider 필요성

- 컴퓨팅 인프라 변화 추이 및 현황:

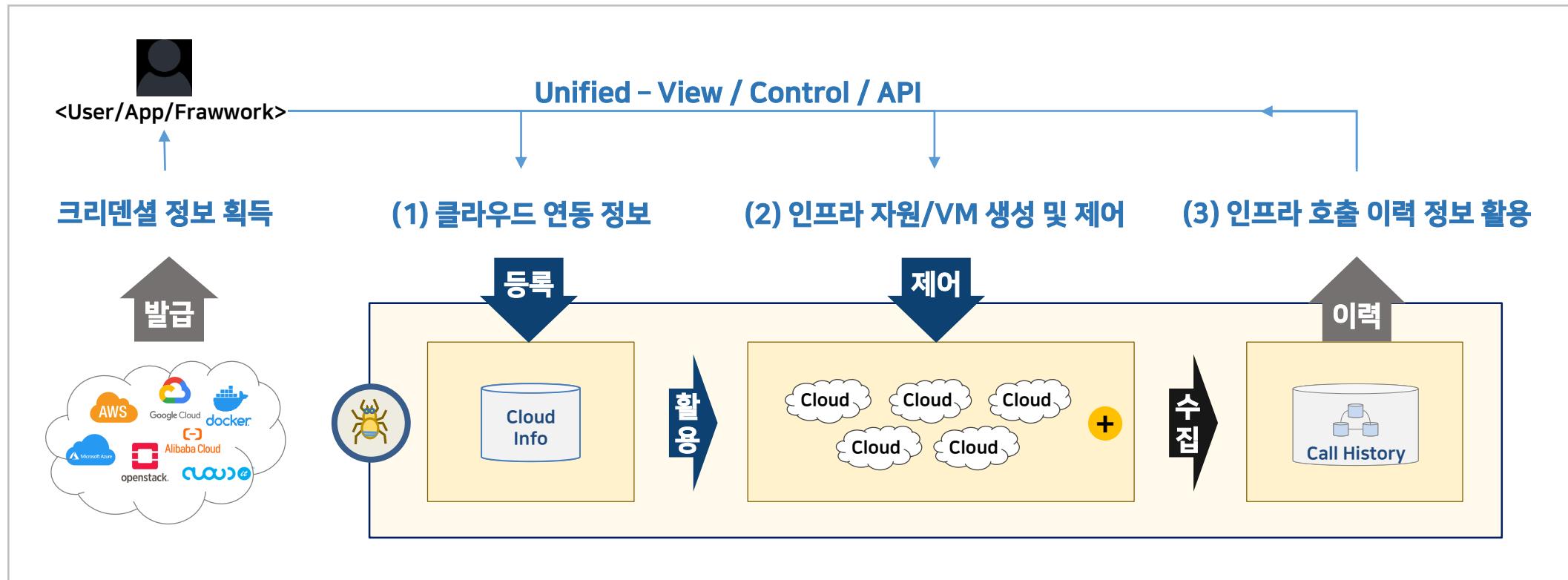


- CB-Spider: [플러그인-드라이버 기반의 클라우드 연동 확장성을 제공하는 멀티클라우드 인프라 연동 프레임워크](#)



CB-Spider 활용 개요

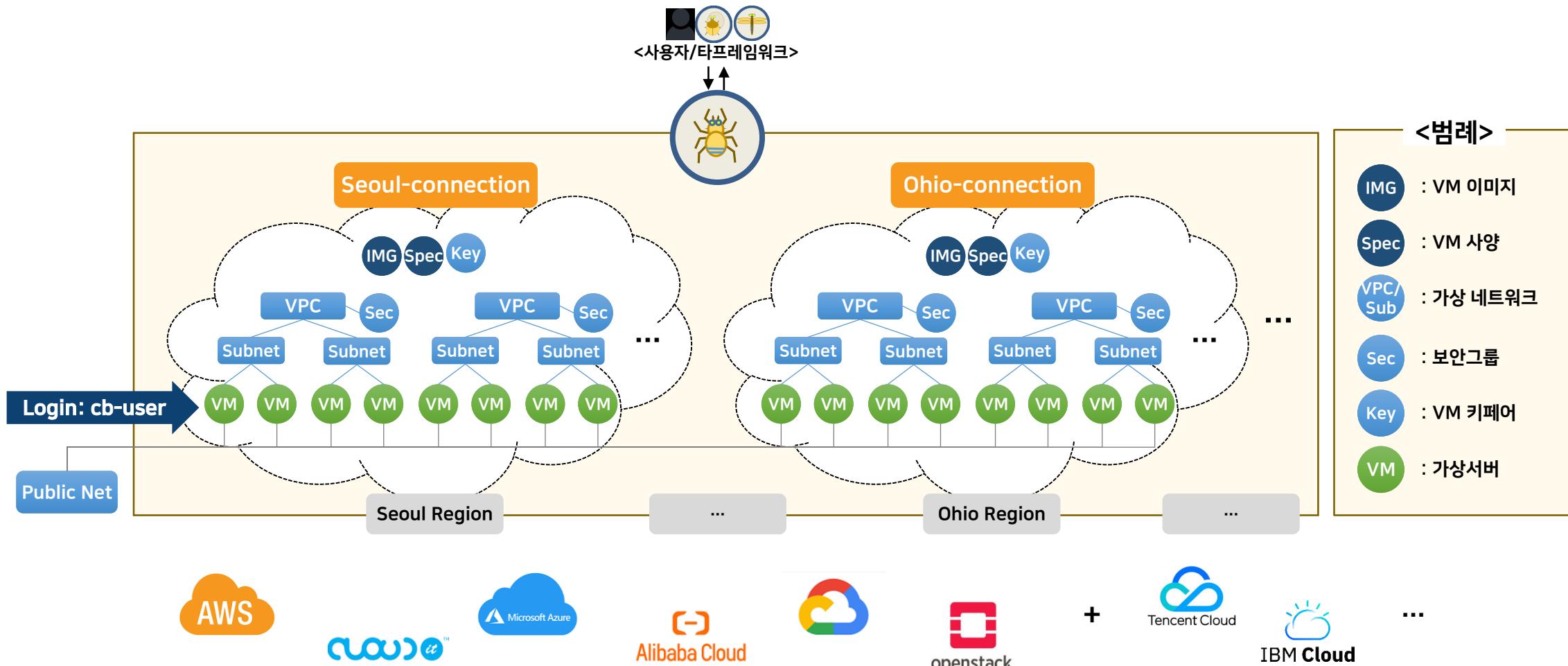
- 흐름: Register(**CID**, cloud info) → Control(**CID**) ※ CID: Connection ID
- CSP별 주요 API Call 이력 정보 제공: 클라우드 및 리전 최적 선택에 활용 ※ CSP: Cloud Service Provider



결국 사용자는 서로 다른 CSP에서 동일한 방법으로 글로벌 스케일의 VM을 활용할 수 있다.

CB-Spider가 제공하는 멀티클라우드 인프라 자원 형상

- 이종 멀티클라우드 자원의 통합 구성 제공 → 동일 CSP 처럼 공통 제어 가능
- 클라우드 '연결 설정' 중심의 자원 독립 구성 제공 (연결 설정: 연동정보=클라우드 드라이버 정보, 크리덴셜 정보, 리전/존 정보)



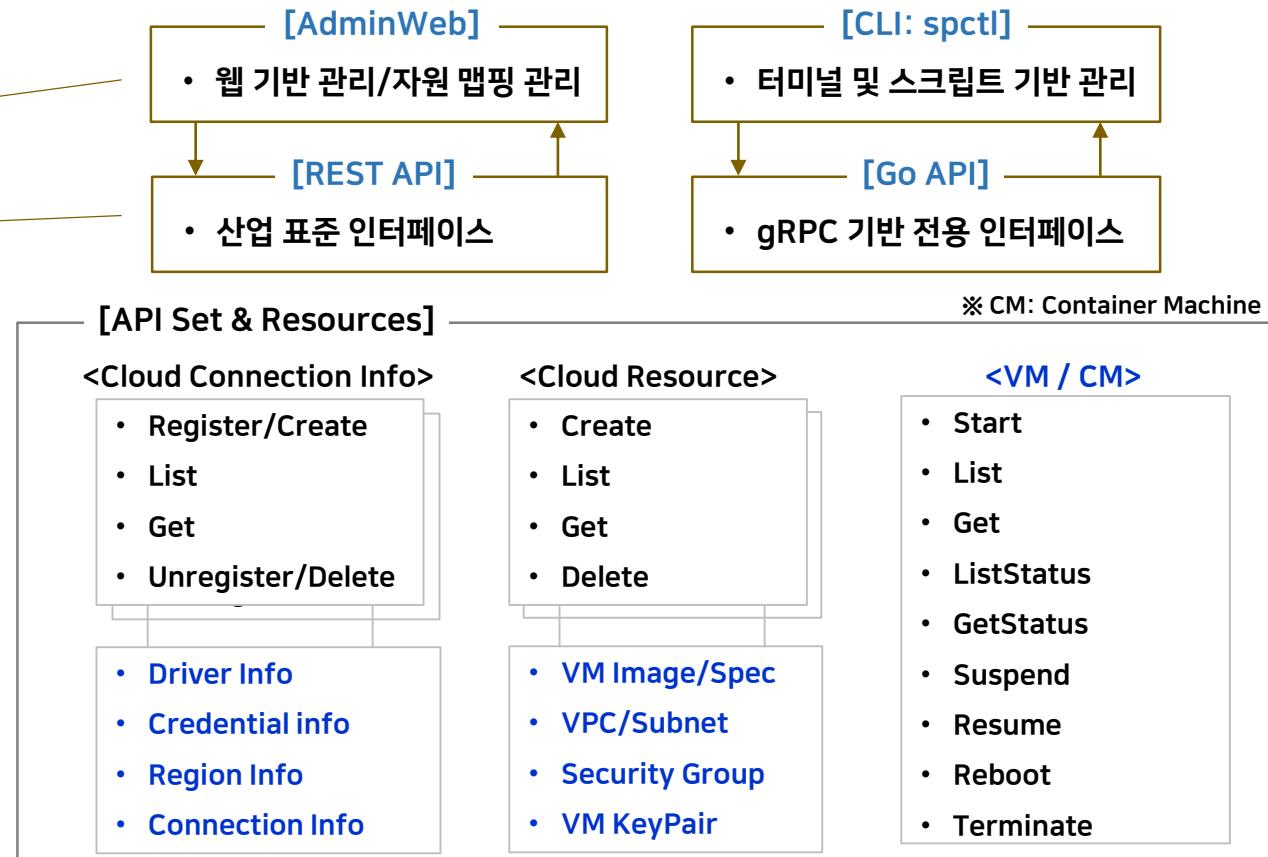
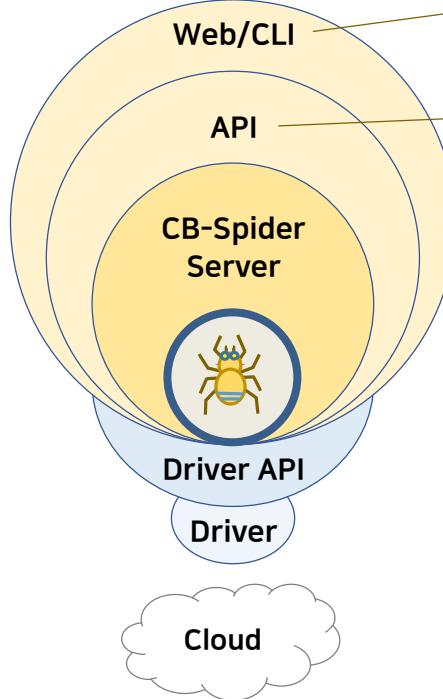
멀티클라우드 연동 확장을 위한 CB-Spider 구조 및 주요 기능

- 프레임워크 구조: Interface – Server – Driver API – Drivers – {Clouds}
- 프레임워크 특징: 드라이버-플러그인 기반 클라우드 연동, 단일 API/통합 제어, CSP API 호출 이력 정보 제공



다양하고 심플한 사용자 인터페이스

- 개발자를 위한 REST 및 Go API, 터미널 및 웹 사용자를 위한 CLI 및 AdminWeb 등 다양한 인터페이스 제공
- 대상별 Create-List-Get-Delete 등 심플한 인터페이스 제공



<CB-Spider 사용자 인터페이스>

CB-Spider를 활용하면...

* 단지 3개 설정 값으로 전세계 수많은 클라우드의 공통 제어 가능한 응용 개발 가능

변수

```
CONN_CONFIG=aws-oregon-config
IMAGE_NAME=ami-090717c950a5c34d3
SPEC_NAME=t2.micro

./vm-start.sh
```



```
CONN_CONFIG=azure-northeu-config
IMAGE_NAME=Canonical:UbuntuServer:18.04-LTS:latest
SPEC_NAME=Standard_B1ls

./vm-start.sh
```



...

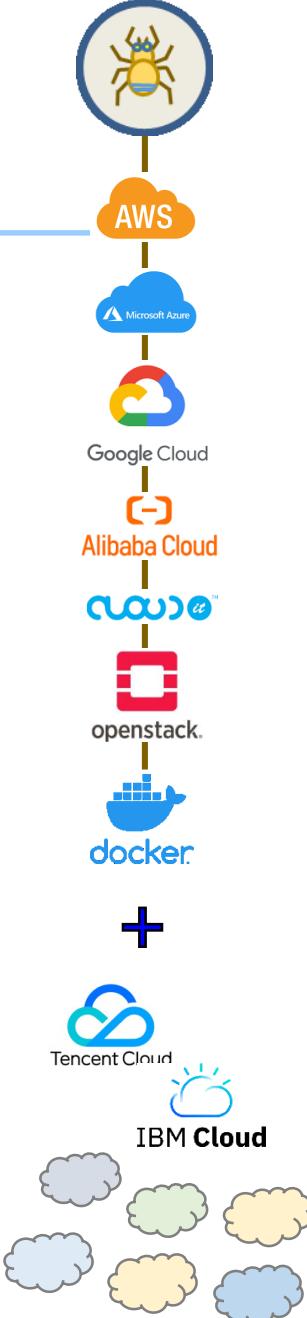
동일 코드 응용

```
#!/bin/bash

time $CLIPATH/spctl vm start -i json -d ¶
{
  "ConnectionName": "${CONN_CONFIG}",
  "ReqInfo": {
    "Name": "${VM_NAME}",
    "ImageName": "${IMAGE_NAME}",
    "VPCName": "${VPC_NAME}",
    "SubnetName": "${SUBNET_NAME}",
    "SecurityGroupNames": [ "${SG_NAME}" ],
    "VMSpecName": "${SPEC_NAME}",
    "KeyPairName": "${KEYPAIR_NAME}"
  }
} 2> /dev/null
```

* 개발 가능 언어: CLI / REST / GO / ...

Free lunch



결국 사용자는 서로 다른 CSP에서 동일한 방법으로 글로벌 스케일의 VM을 활용할 수 있다.



CB-Spider 개발 로드맵

2019년

아메리카노
(19/11)

2020년

카푸치노
(20/6)
에스프레소
(20/11)

2021년

카페모카
(21/6)

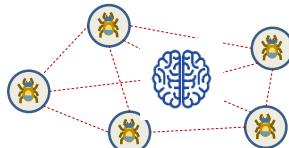
아포가또
(21/11)

2022년

카페라떼
(22/4)
코르타도
(22/9)
핸드드립
(22/11)

엣지
분산
지능

* 장기 로드맵



스파이더 엣지/분산/지능 정의 및 방향 설정 등

PoC 분석 및 기능 검증

연동
이력
관리

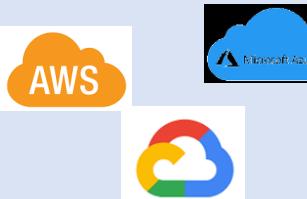


연동 이력 정보
관리 시스템 개발

멀티클라우드 인프라 연동 이력 아카이빙

연동 품질 분석 / 연동 이력 활용 등

연동
및
통합
제어



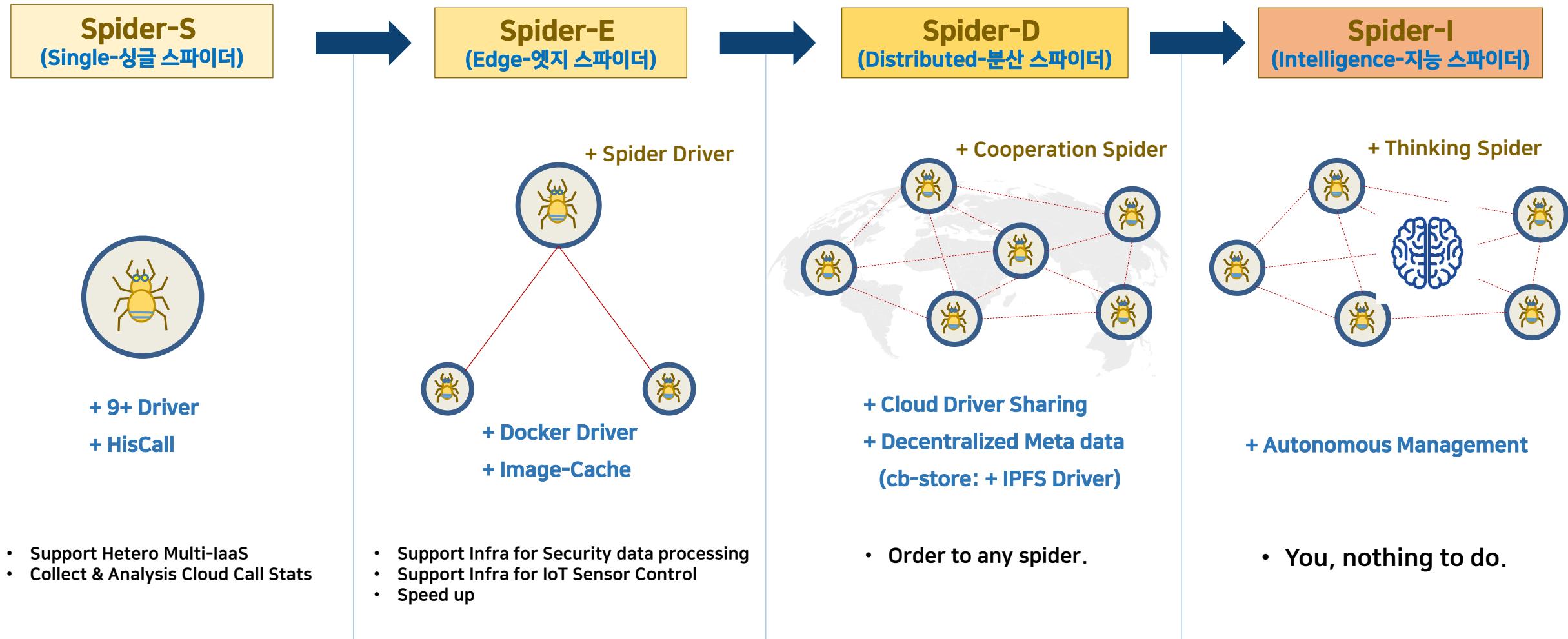
국내외 클라우드 연동

국내외 클라우드 추가 선정 및 연동

스파이더 프레임워크 및 드라이버 최적화/안정화



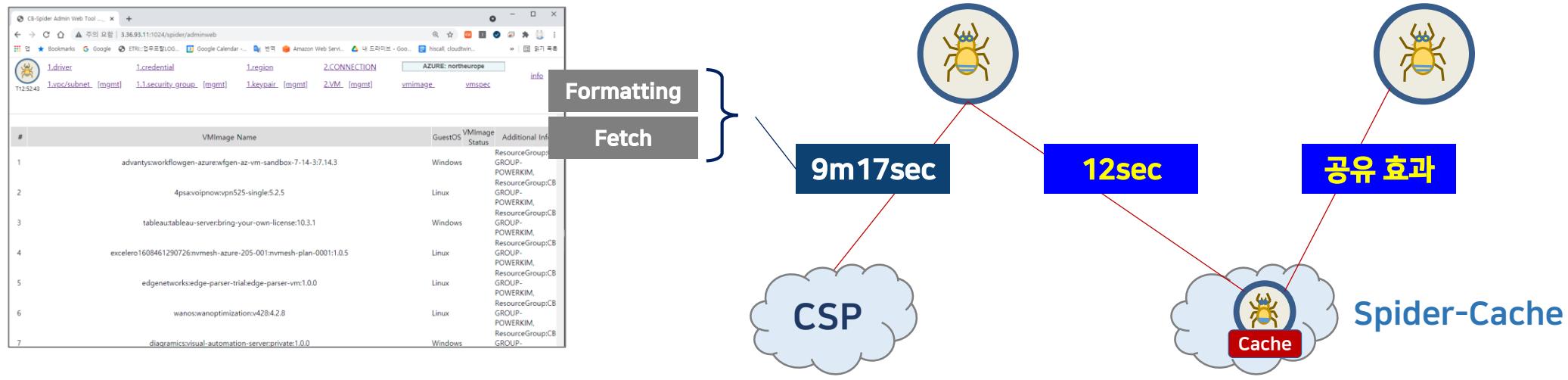
장기 로드맵 - 엣지→분산→지능



※ 전 세계 클라우드를 엮는 그날까지…

엣지 스파이더 PoC 시험: 수직 성능 개선을 위한 Spider-Cache

- 현재, 자원 정보 획득 시 가장 큰 문제 → VM 이미지 정보 제공 성능: 단일 리전-7천개 이상, **9분 이상 소요**
- PoC Test 결과 : **약 46배 개선 (9분17초 → 12초)**



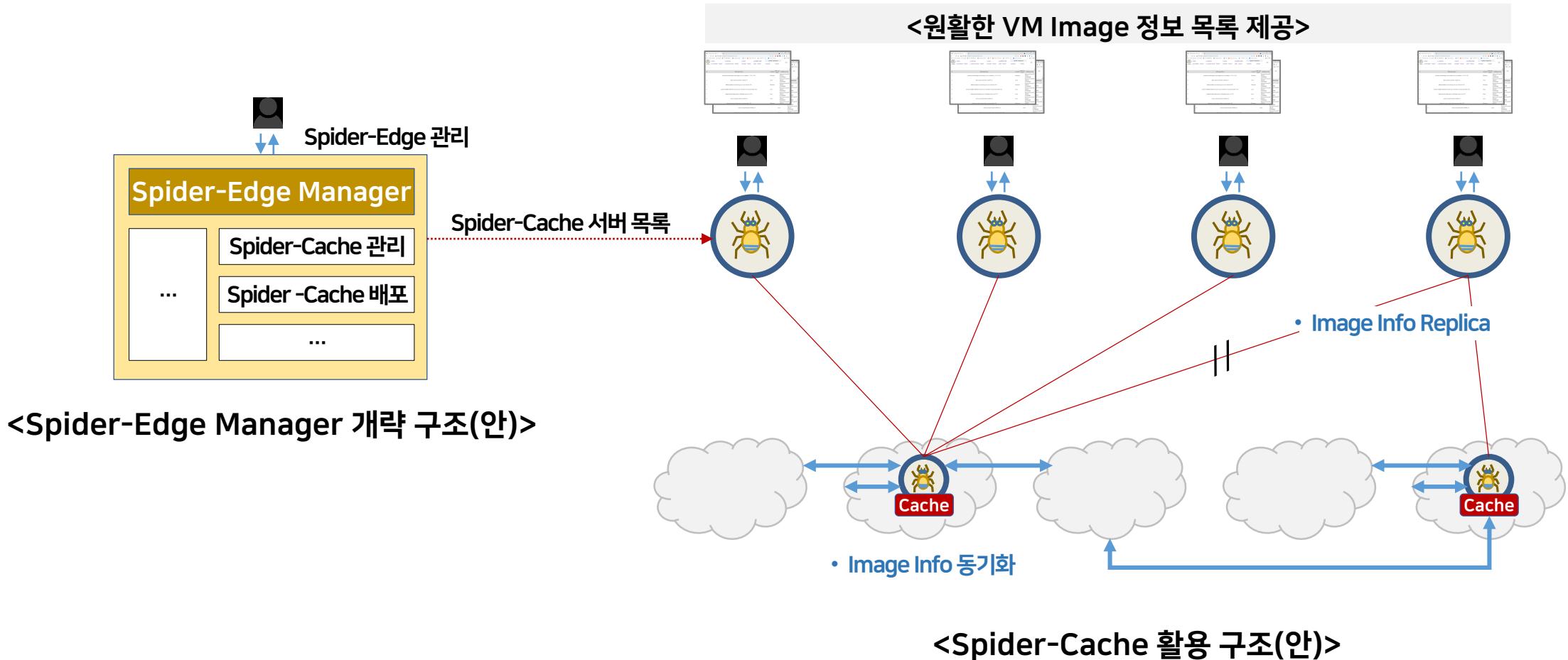
<VM Image 정보 목록 예시: 7,081개/9m17sec>

<AS-IS>

<TO-BE>

엣지 스파이더 PoC 계획

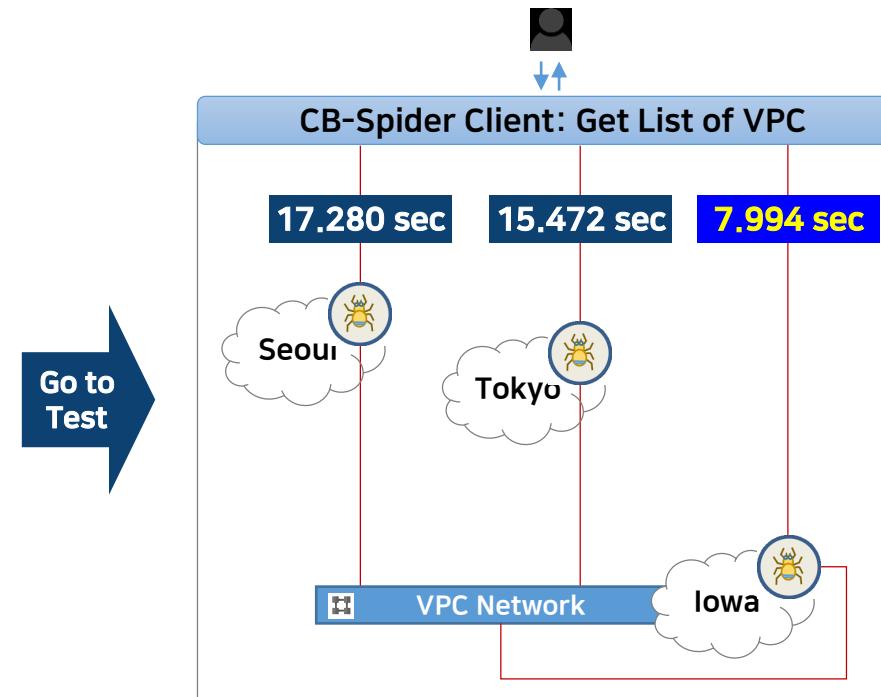
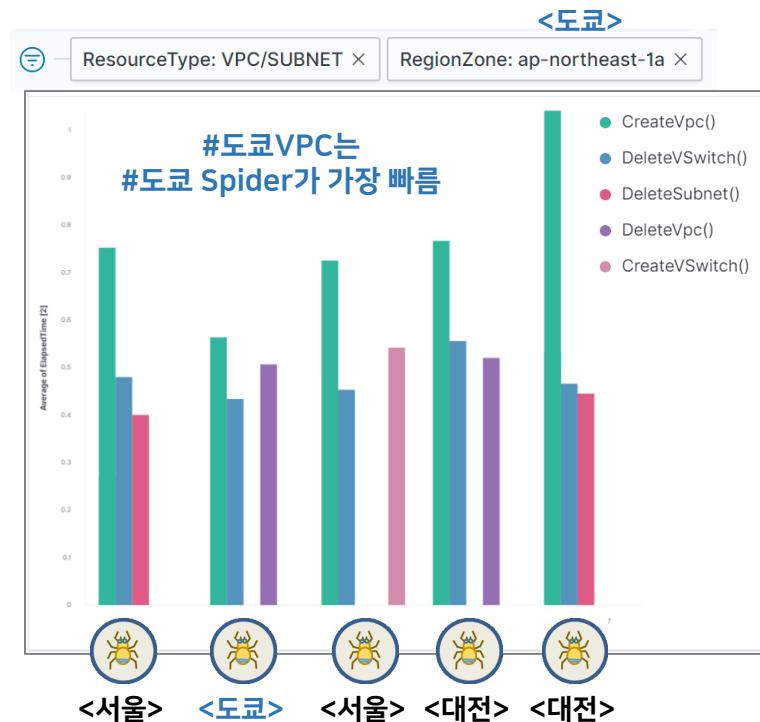
- 개념 구체화 → 세부 구조, 캐싱 메커니즘 등 설계 → 개발 및 활용 시험 → 점증적 적용



분산 스파이더 PoC 시험: 수평 확장을 위한 Dandelion-Spider

- 거리가 멀면 느린게 당연하다. → Spider를 가까이 옮겨 보자 → 숙제: 효과적인가?
- PoC Test 결과 : 약 2배 차이 (17초 vs 8초)

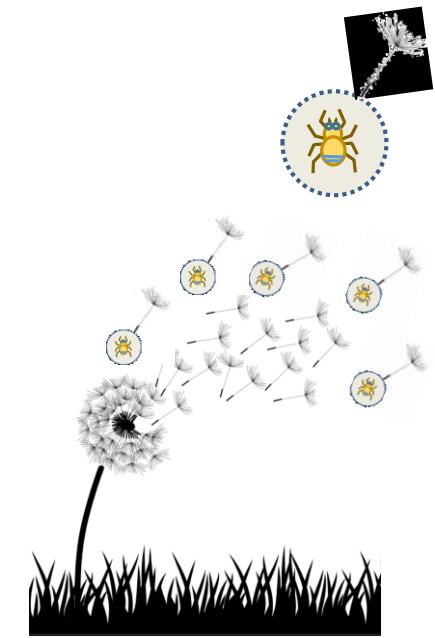
※ Dandelion: 민들레



<HisCall 분석: Spider 위치와 호출 성능>

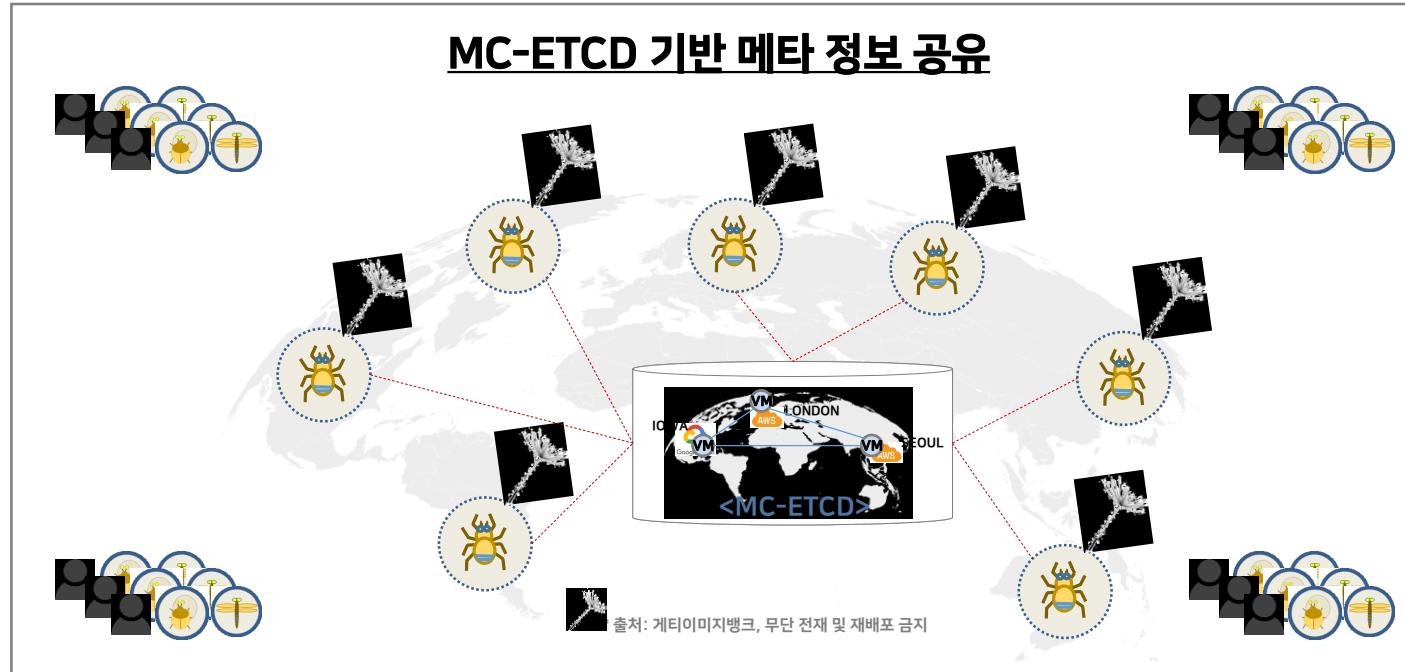
<Spider 배치에 따른 호출 성능 비교 시험>

<Dandelion-Spider>



Dandelion-Spider PoC 시험 환경 및 시나리오

- 7개의 Spider로 구성된 Dandelion-Spider 실행
- 특정 Spider(Seoul)에서 자원 생성/삭제
- 다른 Dandelion-Spider들에서 자원 목록 확인



< Dandelion-Spider 운영 환경 예시 >

Dandelion-Spider PoC 결과: 드라이버 추가/삭제 → 전파 검증

1.AWS:SEOUL

```
bitnami@ip-172-31-11-205: ~$ go/src/github.com/cloud-barista/cb-spider
aws NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36", "status":200, "error": "", "latency":159784059, "latency_human": "159.784059ms", "bytes_in":102, "bytes_out":93}
{"time": "2020-12-24T10:16:52.305616084z", "id": "", "remote_ip": "127.0.0.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/driver", "user_agent": "Go-http-client/1.1", "status":200, "error": "", "latency":770449898, "latency_human": "770.449898ms", "bytes_in":0, "bytes_out":106}
{"time": "2020-12-24T10:16:52.306069502z", "id": "", "remote_ip": "127.0.0.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/cloudos", "user_agent": "Go-http-client/1.1", "status":200, "error": "", "latency":145892, "latency_human": "145.892ms", "bytes_in":0, "bytes_out":106}
```

2.GCP:IOWA

```
ubuntu@ip-172-31-9-201: ~$ gcp
468494618, "latency_human": "468.94618ms", "bytes_in":0, "bytes_out":106}
{"time": "2020-12-24T10:16:50.4359091z", "id": "", "remote_ip": "1.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/adminweb/driver", "user_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36", "status":200, "error": "", "latency":771234351, "latency_human": "771.234351ms", "bytes_in":0, "bytes_out":6100}
```

3.AWS:LONDON

```
ubuntu@ip-172-31-128-219: ~$ aws
1/7.64.0" "status":200, "error": "", "latency":557949840, "latency_human": "557.94984ms", "bytes_in":0, "bytes_out":106}
{"time": "2020-12-24T10:16:56.433501832z", "id": "", "remote_ip": "127.0.0.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/driver", "user_agent": "curl/7.64.0", "status":200, "error": "", "latency":593270880, "latency_human": "593.27088ms", "bytes_in":0, "bytes_out":106}
```

4.AWS:MUMBAI

```
ubuntu@ip-172-31-22-25: ~$ dist$ ./apiserver
<CB-Spider> Multi-cloud Infrastructure Framework
Adminweb: http://13.234.38.147:1024/spider/adminweb
REST API: http://13.234.38.147:1024/spider
Go API: grpc://13.234.38.147:2048
```

<Spider AdminWeb>

#	Provider Name	Driver Library Name	Driver Name
1	AWS	aws-driver-v1.so	aws-driver-01
#	AWS	aws-driver-v1.so	aws-driver-01

```
{"driver": [{"DriverName": "aws-driver-01", "ProviderName": "AWS", "DriverLibFileName": "aws-driver-v1.so"}]}
```

5.AWS:TOKYO

```
ubuntu@ip-172-31-9-201: ~$ dist
{"latency_human": "695.522362ms", "bytes_in":0, "bytes_out":106}
{"time": "2020-12-24T10:16:56.419037121z", "id": "", "remote_ip": "127.0.0.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/driver", "user_agent": "curl/7.68.0", "status":200, "error": "", "latency":192136302, "latency_human": "192.136302ms", "bytes_in":0, "bytes_out":106}
```

```
{"driver": [{"DriverName": "aws-driver-01", "ProviderName": "AWS", "DriverLibFileName": "aws-driver-v1.0.so"}]}
```

6.AWS:SAOPAULO

```
ubuntu@ip-172-31-128-219: ~$ dist
{"latency_human": "446.819517ms", "bytes_in":0, "bytes_out":106}
{"time": "2020-12-24T10:16:56.836504003z", "id": "", "remote_ip": "127.0.0.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/driver", "user_agent": "curl/7.68.0", "status":200, "error": "", "latency":549770639, "latency_human": "549.770639ms", "bytes_in":0, "bytes_out":106}
```

```
{"driver": [{"DriverName": "aws-driver-01", "ProviderName": "AWS", "DriverLibFileName": "aws-driver-v1.0.so"}]}
```

7.AWS:STOCKHOLM

```
ubuntu@ip-172-31-29-85: ~$ dist
localhost:1024", "method": "GET", "uri": "/spider/driver", "user_agent": "curl/7.68.0", "status":200, "error": "", "latency":470869295, "latency_human": "470.869295ms", "bytes_in":0, "bytes_out":106}
{"time": "2020-12-24T10:16:56.533965906z", "id": "", "remote_ip": "127.0.0.1", "host": "localhost:1024", "method": "GET", "uri": "/spider/driver", "user_agent": "curl/7.68.0", "status":200, "error": "", "latency":688427817, "latency_human": "688.427817ms", "bytes_in":0, "bytes_out":106}
```

<Spider Server>

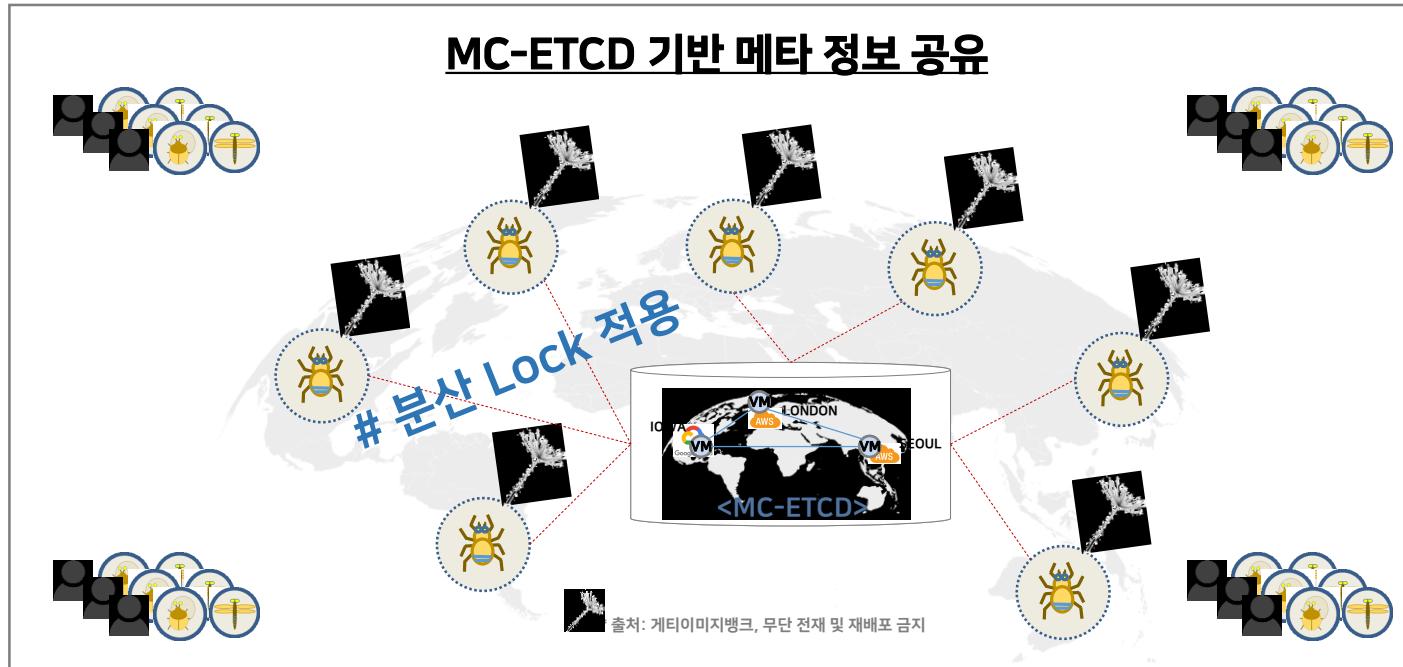
<Spider Client>

<Spider Server>

<Spider Client>

Dandelion-Spider PoC 계획

- 분산 Lock 적용 및 관련 PoC
- Dandelion-Spider는 빠른 적용보다는 다양한 시도에 중점



< Dandelion-Spider 운영 환경 예시 >

Dandelion-Spider는 글로벌 스케일 멀티클라우드의 또 다른 시도입니다.



참고: CB-Spider 활용 가이드

- 설치, 실행 옵션 및 API 규격 등 README

<https://github.com/cloud-barista/cb-spider#readme>

- Quick Start 가이드

<https://github.com/cloud-barista/cb-spider/wiki/Quick-Start-Guide>

- Mock 드라이버 이용한 실습 가이드

<https://github.com/cloud-barista/cb-spider/wiki/How-to-test-CB-Spider-with-Mock-Driver>

- OpenStack on AWS 설치 및 연동 가이드

<https://github.com/cloud-barista/cb-spider/wiki/How-to-install-OpenStack-on-a-VM-for-CB-Spider-Testing>

- AdminWeb 활용 가이드

[https://github.com/cloud-barista/cb-spider/wiki/CB-Spider-AdminWeb-도구-소개-및-활용-가이드\(Korean\)](https://github.com/cloud-barista/cb-spider/wiki/CB-Spider-AdminWeb-도구-소개-및-활용-가이드(Korean))

- CLI 활용 가이드

<https://github.com/cloud-barista/cb-spider/wiki/CLI-Examples>

- REST API 활용 가이드

<https://github.com/cloud-barista/cb-spider/wiki/REST-API-Examples>

- Go API 활용 가이드

<https://github.com/cloud-barista/cb-spider/wiki/Go-API-Examples>



참고: CB-Spider 참여 가이드

- CB-Spider Tutorials

<https://github.com/cloud-barista/cb-spider/wiki/Tutorials>

- CB-Spider 개발 분야

<https://github.com/cloud-barista/cb-spider/wiki/Development-reference-and-Contributor-WANTED!!>

- 클라우드 드라이버 개발자 가이드

<https://github.com/cloud-barista/cb-spider/wiki/Cloud-Driver-Developer-Guide>

- 참여 관련 문의

▷ <https://github.com/cloud-barista/cb-spider/issues>

▷ <https://cloud-barista.slack.com/archives/CLFCLNFTJ>



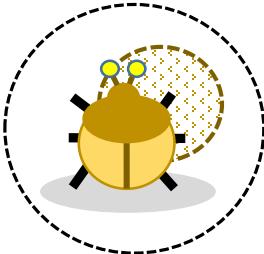


멀티클라우드, “글로벌 스케일로 시작하다”

클라우드바리스타 커뮤니티 제4차 컨퍼런스

CB-Tumblebug :

멀티클라우드 인프라 서비스 통합 운용/관리



손석호 / CB-Tumblebug 프레임워크 리더

카페모카(Café Mocha) 한잔 어떠세요 ?

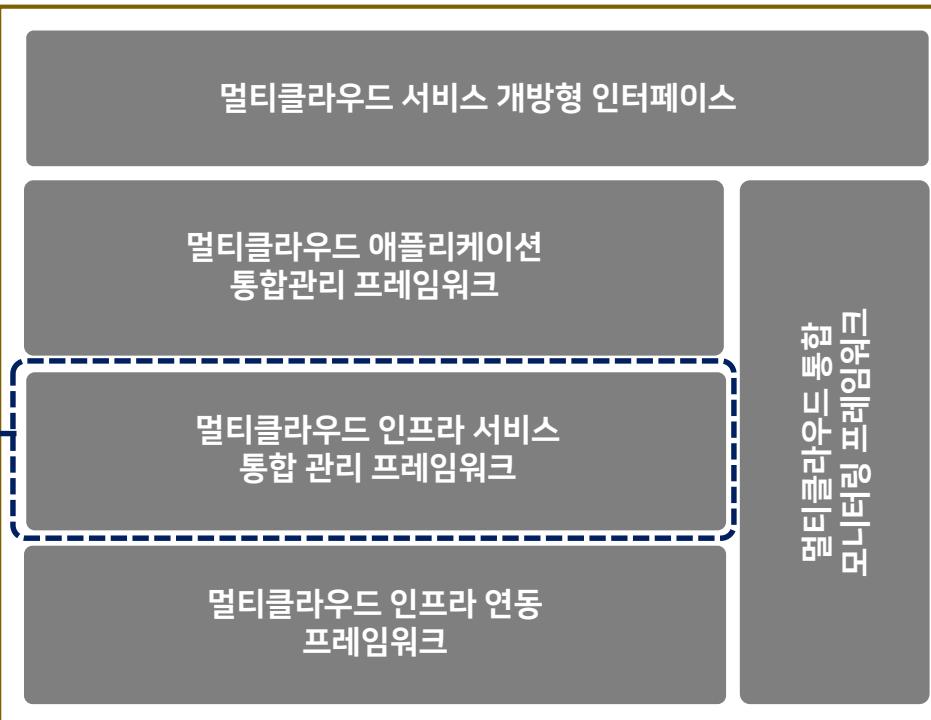
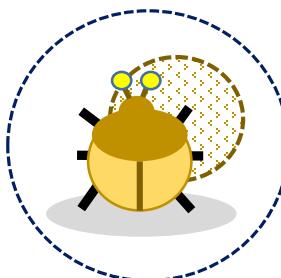


이번 세션은...

응용/도메인/기관 특화 SW



CB-Tumblebug



멀티클라우드 서비스 공동 플랫폼

목차

I

CB-Tumblebug 개요

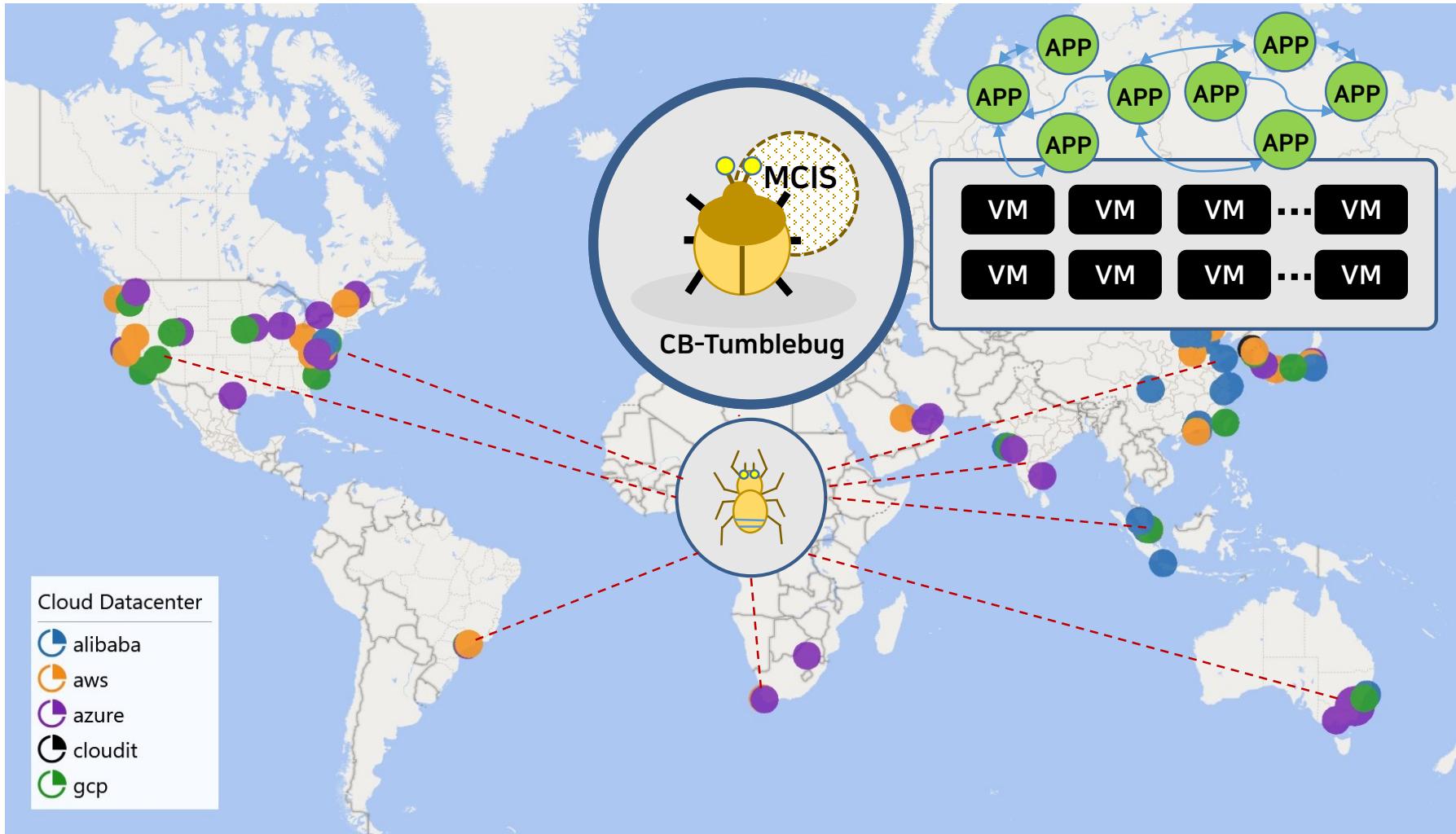
II

CB-Tumblebug 주요 기능 및 특징

III

CB-Tumblebug 주요 개선 포인트 및 로드맵

CB-Tumblebug: 멀티클라우드 인프라 서비스 통합 관리 기술 필요성

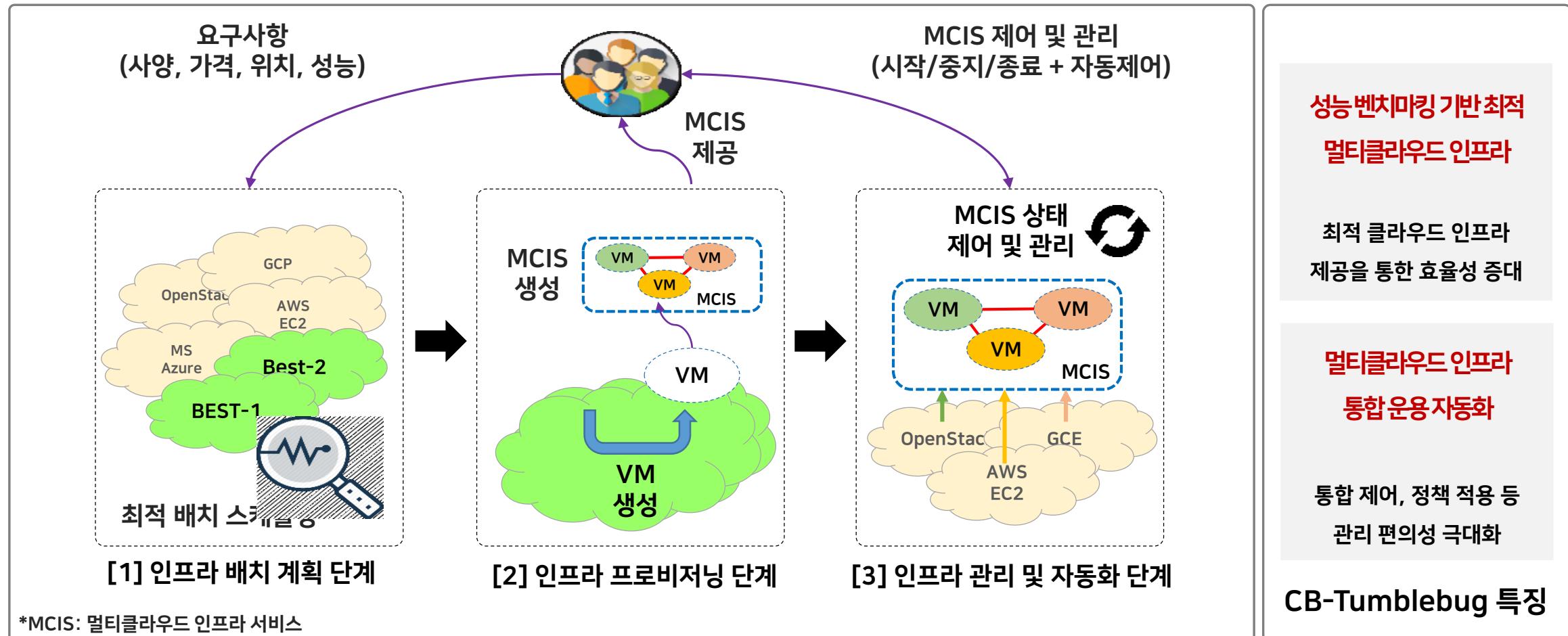


복잡한 멀티클라우드의 효과적인 활용을 위해, 자원을 유기적으로 통합 제공하는 멀티클라우드 인프라 서비스 통합 관리 필요

CB-Tumblebug: 멀티클라우드 인프라 서비스 통합 관리 기술 정의

**기술
정의**

사용자 요구사항에 따라 최적의 멀티클라우드 인프라 서비스를 조합하여 프로비저닝하고,
통합 제어 및 관리를 통해 사용자의 컴퓨팅 인프라 운용을 지원하는 기술

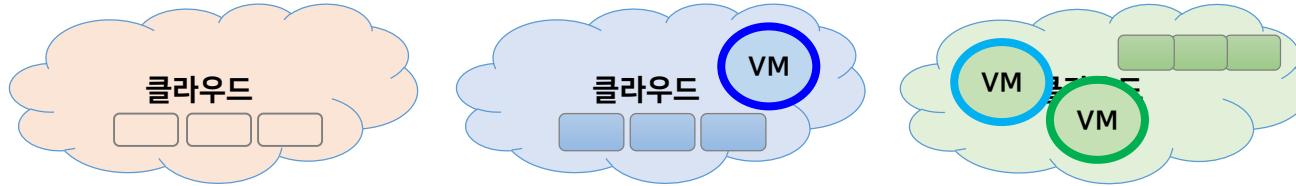
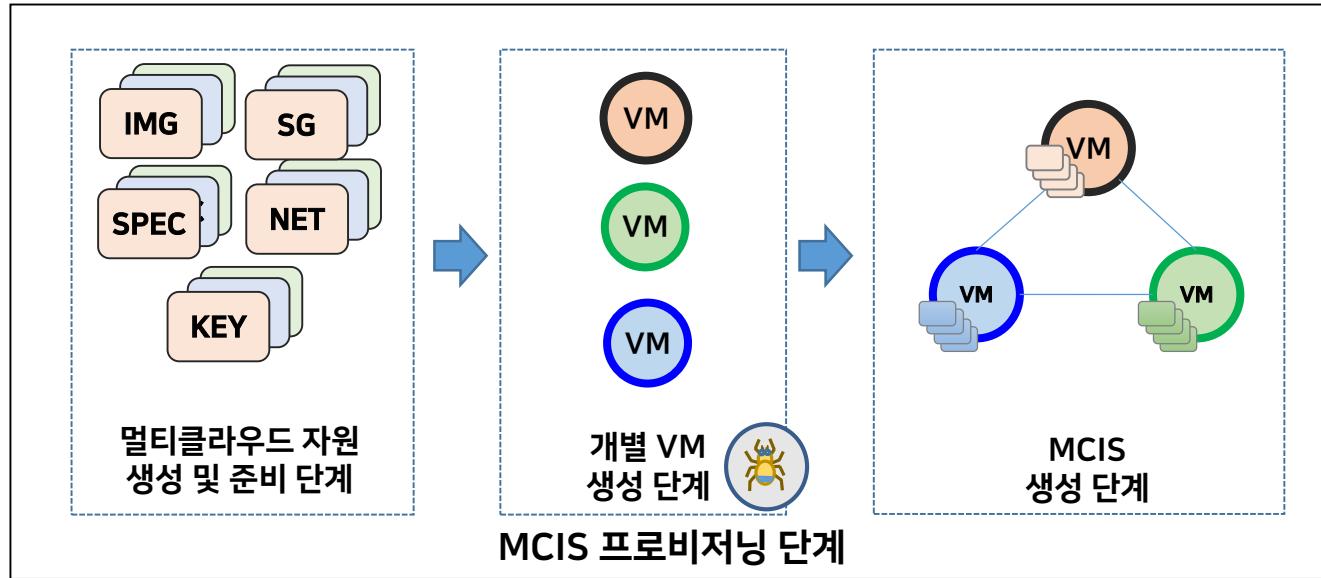


CB-Tumblebug 주요 기능

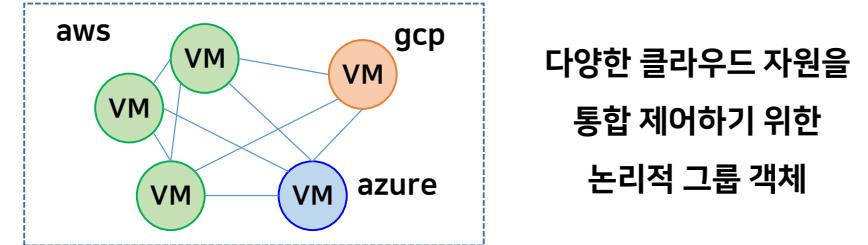
- MCIS 프로비저닝 및 특화 구성
 - 다양한 클라우드 자원을 활용하여 MCIS를 생성하고 특화하는 기능
- MCIS 라이프사이클 관리
 - MCIS 라이프사이클 상태를 종합적으로 관리, 통합 제어하는 기능
- MCIS 최적 구성 및 배치 스케줄링
 - 클라우드 평가를 통해 MCIS를 최적 구성하고 스케줄링 하는 기능
- MCIS 자동 제어
 - MCIS를 진단하고 결과에 따라 자동 제어하는 운영 자동화 기능

MCIS 프로비저닝 및 특화 구성

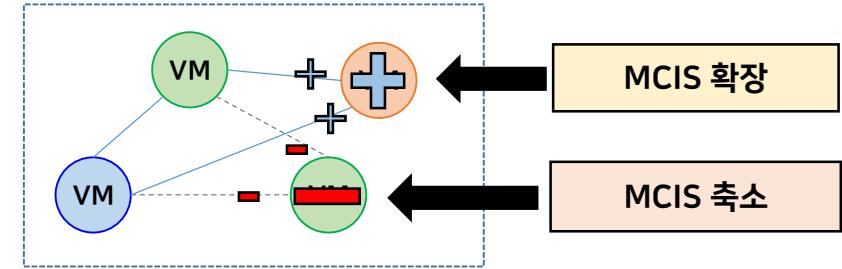
- 멀티클라우드 자원을 활용하여 MCIS를 생성하고 특화하는 기능



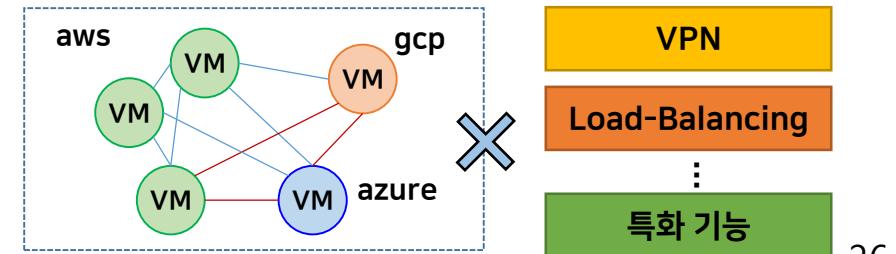
[MCIS 구성 예시]



[MCIS 구성 변경 예시]

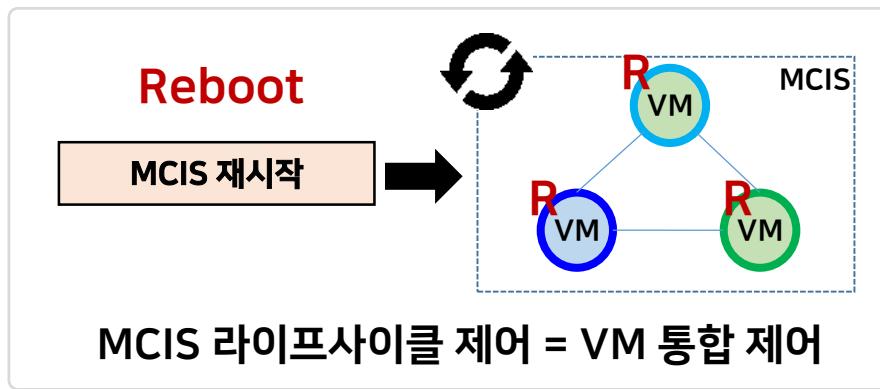


[MCIS 특화 구성 예시]



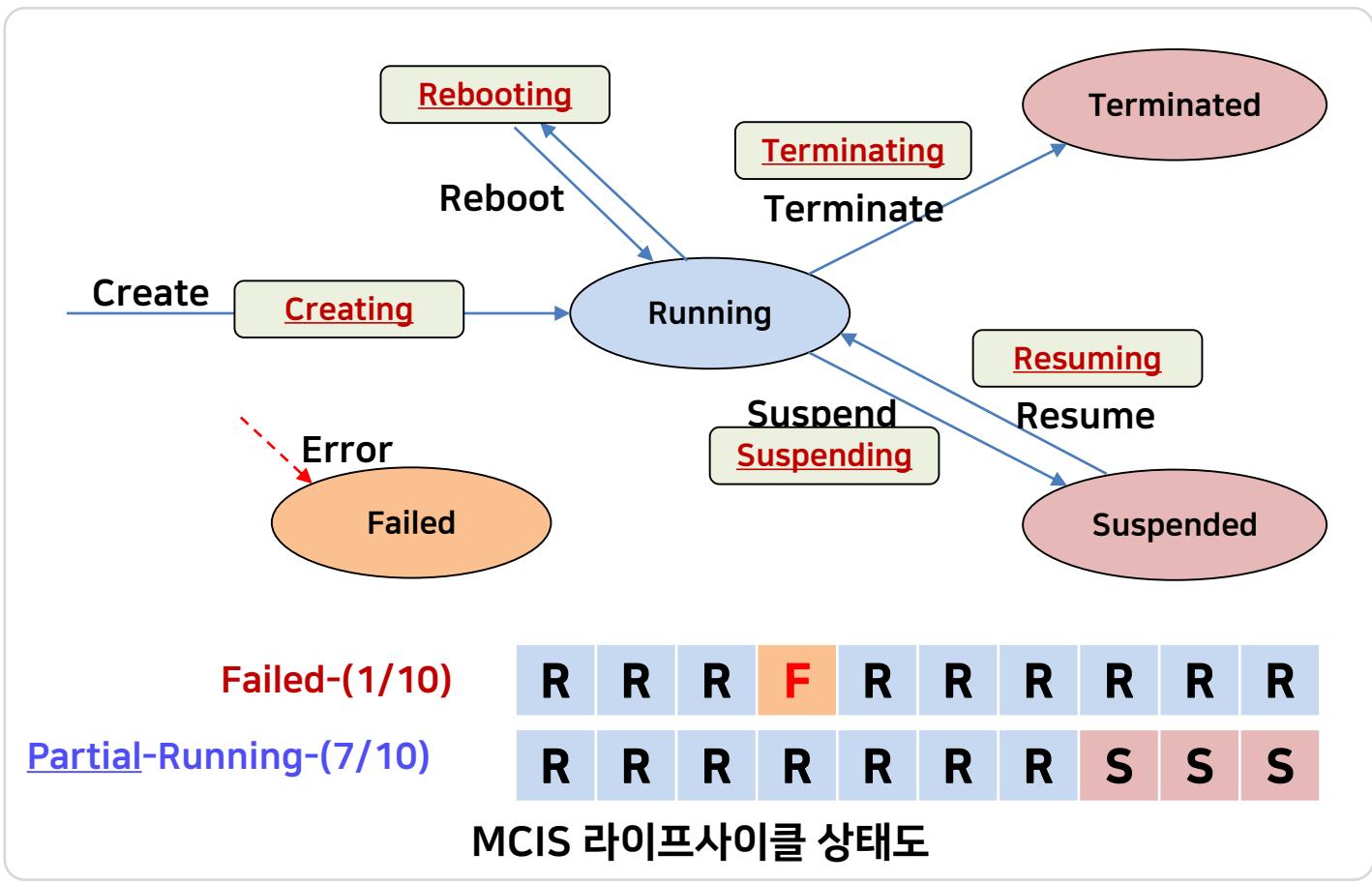
MCIS 라이프사이클 관리

- MCIS의 라이프사이클 통합 제어하고 상태를 쉽게 파악할 수 있는 기능
 - MCIS의 세부 요소들의 대표적인 상태를 한눈에 표시 (ex: Partial State)



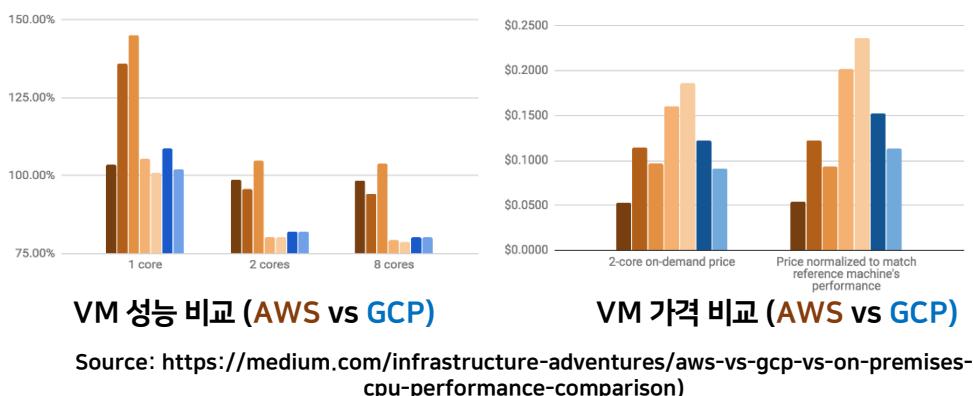
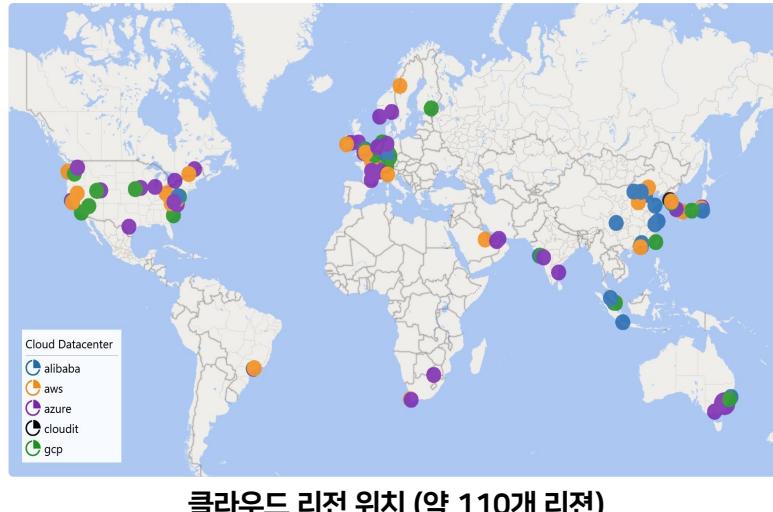
Cloud-Barista	Creating	Running	Suspending	Suspended	Resuming	Rebooting	Terminating	Terminated	Failed
Alibaba	Pending	Running	Stopping	Stopped	Resuming (자체생성상태)	Rebooting (자체생성상태)	Terminating (자체생성상태) (OP: Stop&Delete)	Deleted	
AWS	생성 중 (Creating)	running	stopping	stopped	Resuming (자체생성상태)	Rebooting	shutting-down	terminated	상태 정보를 얻을 수 없을 때 오류로 인한 비정상 상태
GCP	Provisioning staging	running	stopping	terminated	Resuming (자체생성상태)	Rebooting (자체생성상태) (OP: Stop&Start)	Terminating (자체생성상태) (OP: Delete)	(예외)NotExist	상태 정보를 얻을 수 없을 때 오류로 인한 비정상 상태
Azure	starting /-	running /succeeded	stopping /-	stopped /succeeded	Resuming (자체생성상태) → Creating...	Rebooting (자체생성상태) (OP: Stop&Start)	deallocating /-	(예외)NotExist	/Failed
OpenStack	BUILD	ACTIVE	Suspending (자체생성상태)	SHUTOFF	Resuming (자체생성상태)	REBOOT	Terminating (자체생성상태) (OP: Delete)	(예외)NotExist	Error
Cloudit	CREATING	RUNNING	STOPPING	STOPPED	STARTING	REBOOTING	DESTROYING (예외)NotExist		FAILED

MCIS 라이프사이클 상태 검증 및 보정

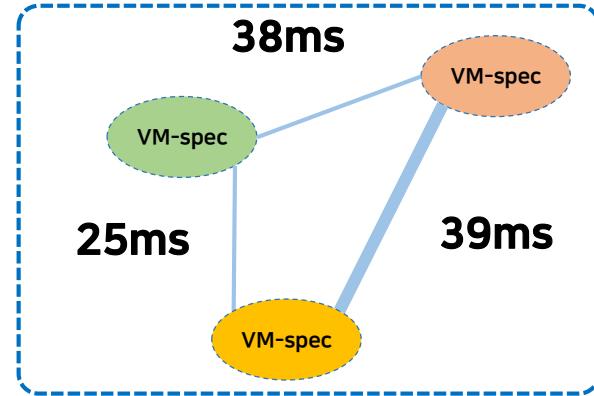


MCIS 최적 구성 및 배치 스케줄링

- 다양한 정보 수집을 통해 MCIS를 최적으로 구성하고 스케줄링
 - 멀티클라우드는 다양한 클라우드를 활용하므로 최적 선정이 매우 중요



- MCIS 통합 최적 배치
 - VM 간 응답속도 기반 배치

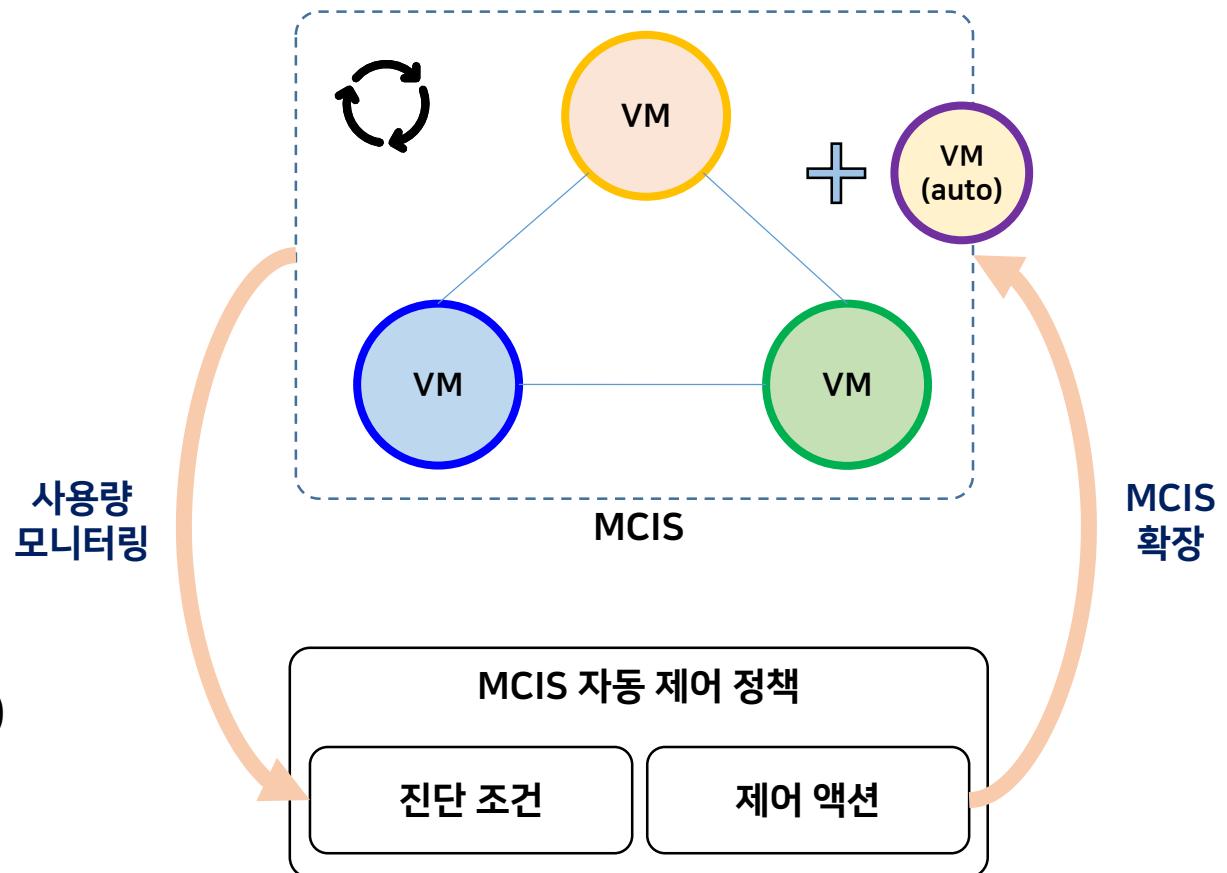


예) VM 간 응답시간 < 40ms

- MCIS 개별 VM의 최적 배치
 - VM 스펙 기반 배치
 - VM 가격 기반 배치
 - VM 위치 기반 배치
 - VM 성능 기반 배치 (벤치마킹)
 - VM 복합 조건 기반 배치

사용자 정책 기반 MCIS 자동 제어

- 사용자 정책을 기반으로 MCIS의 상태를 진단하고, 결과에 따라 MCIS를 자동 제어하는 기능
 - 대규모 자원의 수동 처리 한계 극복
- MCIS 진단 조건
 - MCIS 자원 사용량 진단
 - MCIS 라이프사이클 진단
 - MCIS 성능 진단
 - ...
- MCIS 제어 액션
 - MCIS 규모 제어 (Scale In/Out)
 - MCIS 라이프사이클 제어 (Suspend/Resume)
 - MCIS 자원 교체 (Replace)
 - ...



주요 개선 사항 키워드

- MCIS 형상 및 사용 방법 개선
- 시스템 관리 기능 개선
- 시스템 안정성 개선 및 성능 강화
- 개발자 및 사용자 편의 기능 제공
- 유스케이스 발굴 및 배포 자동화

MCIS VMGroup 개념 지원 (MCIS 형상 및 사용 방법 개선)

- (기존) MCIS 생성시 VM 추가를 위해서는 VM마다 모든 정보(관련 리소스 등)를 추가해야 하는 불편함
- (개선) 대규모 MCIS에 VMGroup 개념을 도입하여, 동일한 정보를 사용하는 VM을 Group 단위로 배포

POST http://\$TumblebugServer/tumblebug/ns/\$NSID/**mcis**

```
{
  "name": "${CONN_CONFIG[$INDEX,$REGION]}",
  "description": "Tumblebug Demo",
  "installMonAgent": "${AgentInstallOn}",
  "vm": [
    {
      "name": "${CONN_CONFIG[$INDEX,$REGION]}",
      "imageId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "vmUserAccount": "cb-user",
      "connectionName": "${CONN_CONFIG[$INDEX,$REGION]}",
      "sshKeyId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "specId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "securityGroupIds": [
        "${CONN_CONFIG[$INDEX,$REGION]}"
      ],
      "vNetId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "subnetId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "description": "description",
      "vmUserPassword": ""
    }
  ],
  "{} X 10"
}
```

기존 방식

POST http://\$TumblebugServer/tumblebug/ns/\$NSID/**mcis**

```
{
  "name": "${CONN_CONFIG[$INDEX,$REGION]}",
  "description": "Tumblebug Demo",
  "installMonAgent": "${AgentInstallOn}",
  "vm": [
    {
      "name": "${CONN_CONFIG[$INDEX,$REGION]}",
      "vmGroupSize": "10",
      "imageId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "vmUserAccount": "cb-user",
      "connectionName": "${CONN_CONFIG[$INDEX,$REGION]}",
      "sshKeyId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "specId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "securityGroupIds": [
        "${CONN_CONFIG[$INDEX,$REGION]}"
      ],
      "vNetId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "subnetId": "${CONN_CONFIG[$INDEX,$REGION]}",
      "description": "description",
      "vmUserPassword": ""
    }
  ]
}
```

개선 방식

데이터 정합성 오류 검토 및 복구 기능 (시스템 관리 기능 개선)

- (기존) 오류에 의해서 리소스 정보가 꼬이는 경우 발생 가능 (**CSP**, **CB-Spider**, **CB-Tumblebug** 정보가 일치하지 않는 오류)
- (개선) CB-Tumblebug의 정보 일치 여부 확인을 위해서 `inspectResources` 기능 제공, 오류 정보 강제 삭제 기능 제공

```

"resourcesOnCsp": [
  {
    "id": "aws-ap-southeast-1-jhseo",
    "cspNativeld": "vpc-00ccdf5d5bcf384ff"
  },
  {
    "id": "",
    "cspNativeld": "vpc-0a3f50515e12e5ea7"
  }
],
"resourcesOnSpider": [
  {
    "id": "aws-ap-southeast-1-jhseo",
    "cspNativeld": "vpc-00ccdf5d5bcf384ff"
  }
],
"resourcesOnTumblebug": [
  {
    "id": "aws-ap-southeast-1-jhseo",
    "cspNativeld": "vpc-00ccdf5d5bcf384ff",
    "nsId": "ns-01",
    "type": "vNet",
    "objectKey": "/ns/ns-01/resources/vNet/aws-ap-southeast-1-jhseo"
  }
]

```

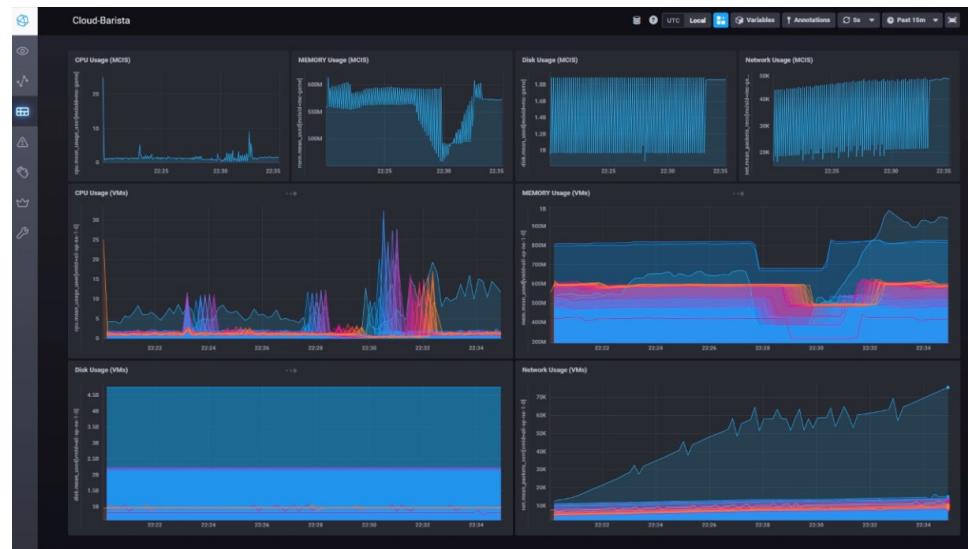
inspect-vNet 결과 예시

[Admin] System management		
GET	/health	Check Tumblebug is alive
DELETE	/object	Delete an object
GET	/object	Get value of an object
DELETE	/objects	Delete child objects along with the given object
GET	/objects	List all objects for a given key

CB-Tumblebug Object 조회 및 강제 삭제 기능

대규모 멀티클라우드 인프라 시험 및 코드 보완 (시스템 안정성 향상)

- (기존) 20개内外의 리전 테스트, 30개内外의 VM으로 구성된 MCIS 테스트
- (개선) 대규모 글로벌 스케일 클라우드 연동 테스트를 통한 시스템 안정성 강화
 - 글로벌 4개 CSP(AWS, Azure, GCP, Alibaba) 약 100개 리전 테스트 진행 및 안정성 향상
 - 100개 이상의 VM으로 구성된 MCIS 테스트를 통한 안정성 및 병렬 처리 속도 향상





Swagger API UI 제공 (개발자 및 사용자 편의성 향상)

The screenshot shows the Swagger UI interface for the CB-Tumblebug API. It displays several sections of endpoints:

- [MCIS] Remote command**:
 - POST /ns/{nsId}/cmd/mcis/{mcisId} Send a command to specified MCIS
 - POST /ns/{nsId}/cmd/mcis/{mcisId}/vm/{vmId} Send a command to specified VM
- [MCIS] Provisioning management**:
 - DELETE /ns/{nsId}/mcis Delete all MCISs
 - GET /ns/{nsId}/mcis List all MCISs
 - POST /ns/{nsId}/mcis Create MCIS
 - DELETE /ns/{nsId}/mcis/{mcisId} Delete MCIS
 - GET /ns/{nsId}/mcis/{mcisId} Get MCIS or Action to MCIS (status, suspend, resume, reboot, terminate, refine)
 - POST /ns/{nsId}/mcis/{mcisId}/vm Create VM in specified MCIS
 - DELETE /ns/{nsId}/mcis/{mcisId}/vm/{vmId} Delete VM in specified MCIS
 - GET /ns/{nsId}/mcis/{mcisId}/vm/{vmId} Get VM in specified MCIS
 - POST /ns/{nsId}/mcis/{mcisId}/vmgroup Create multiple VMs by VM group in specified MCIS
 - POST /ns/{nsId}/mcis/recommend Get MCIS recommendation
 - POST /ns/{nsId}/testRecommendVm Recommend MCIS plan (filter and priority)
- [MCIS] Resource monitor (Developer)**:
 - POST /ns/{nsId}/monitoring/install/mcis/{mcisId} Install monitoring agent (CB-Dragonfly agent) to MCIS
 - GET /ns/{nsId}/monitoring/mcis/{mcisId}/metric/{metric} Get monitoring data of specified MCIS for specified monitoring metric (cpu, memory, disk, network)
- [MCIS] Auto control policy management (WIP)**:
 - DELETE /ns/{nsId}/policy/mcis Delete all MCIS policies

CB-Tumblebug API (Swagger UI)

• Swagger API 지원

- Swagger 기반 자동 API 문서 생성
- CB-Tumblebug 서버의 Swagger UI 웹 제공

This screenshot shows a detailed view of a POST endpoint for creating a MCIS. The endpoint is titled "Create MCIS". The parameters required are "nsId" (Namespace ID) and "mcisReq" (Details for an MCIS object). The "mcisReq" parameter is expanded to show its schema, which includes fields like "connectionType", "imageId", "label", "name", "specType", "subLabel", "subType", "vmGroup", and "vmSpec". Below the schema, there is a code snippet for "Parameter content type" and "Response content type", both set to "application/json".

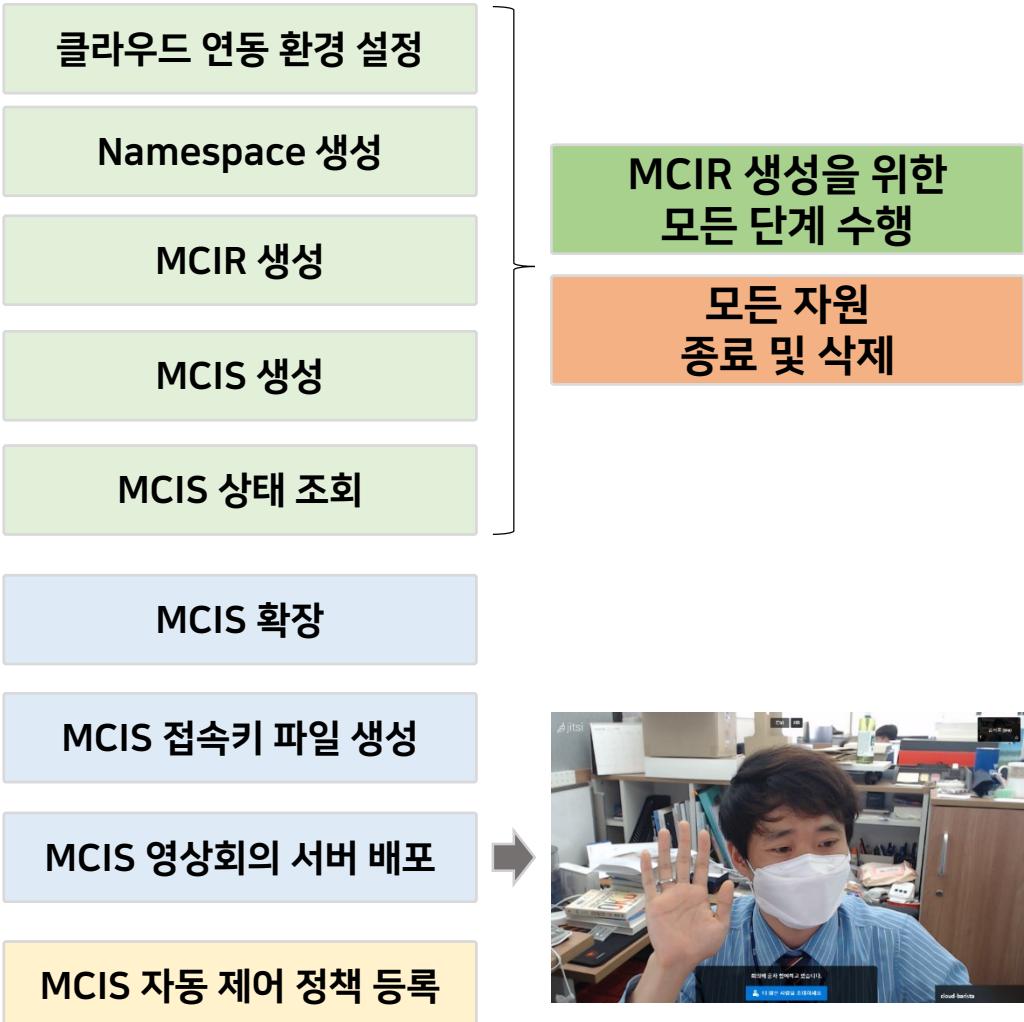
Swagger UI에서 직접 API 호출

```
206 // RestDelMcis godoc
207 // @Summary Delete MCIS
208 // @Description Delete MCIS
209 // @Tags [MCIS] Provisioning management
210 // @Accept json
211 // @Produce json
212 // @Param nsId path string true "Namespace ID"
213 // @Param mcisId path string true "MCIS ID"
214 // @Success 200 {object} common.SimpleMsg
215 // @Failure 404 {object} common.SimpleMsg
216 // @Router /ns/{nsId}/mcis/{mcisId} [delete]
217 func RestDelMcis(c echo.Context) error {
218
219     nsId := c.Param("nsId")
220     mcisId := c.Param("mcisId")
221
222     err := mcis.DelMcis(nsId, mcisId)
223     if err != nil {
224         common.CBLog.Error(err)
225         mapA := map[string]string{"message": err.Error()}
226         return c.JSON(http.StatusInternalServerError, &mapA)
227     }
228
229     mapA := map[string]string{"message": "Deleting the MCIS " + mcisId}
230     return c.JSON(http.StatusOK, &mapA)
231 }
```

프로그램 주석으로 API 문서 자동 생성

시나리오 테스트 도구 개요 (개발자 및 사용자 편의성 향상)

1.configureSpider
2.configureTumblebug
3.vNet
4.securityGroup
5.sshKey
6.image
7.spec
8.mcis
9.monitoring
misc
sequentialFullTest



위치: cb-tumblebug/src/testclient/

[사용 및 개발 편의를 위한 다양한 테스트 스크립트]



다양한 클라우드로 구성된 MCIS 시험 지원 (개발자 및 사용자 편의성 향상)

- CB-Tumblebug 테스트를 위한 다양한 클라우드 리전의 정보를 수집 (conf.env 및 testSet.env)

AWS (Total: 21 Regions)

GCP (Total: 25 Regions)

Alibaba (Total: 23 Regions)

Azure (Total: 40 Regions)

Cloudit (Total: 1 Regions)

OpenStack (Total: 1 Regions)

Mock (Total: 1 Regions)

Cloud-Twin (Total: 1 Regions)

...

- 테스트 대상 클라우드 리전(testSet.env) 설정을 통해, 다양한 클라우드의 VM으로 조합된 MCIS를 쉽게 생성 및 시험 가능

```

## Define sequential test order for cloud types
# Note: you can change order by replacing lines (automatically assign continuous numbers starting from 1)
IX=0
IndexAWS=$((++IX))
IndexAzure=$((++IX))
IndexGCP=$((++IX))
IndexAlibaba=$((++IX))
IndexMock=$((++IX))
IndexOpenstack=$((++IX))
IndexNCP=$((++IX))
IndexCloudTwin=$((++IX))
IndexCloudit=$((++IX))

## Test setting for Regions of Cloud types
# Note: you can change order by replacing lines (automatically assign continuous numbers starting from 1)

# AWS (Total: 21 Regions / Recommend: 20 Regions)
NumRegion[$IndexAWS]=2

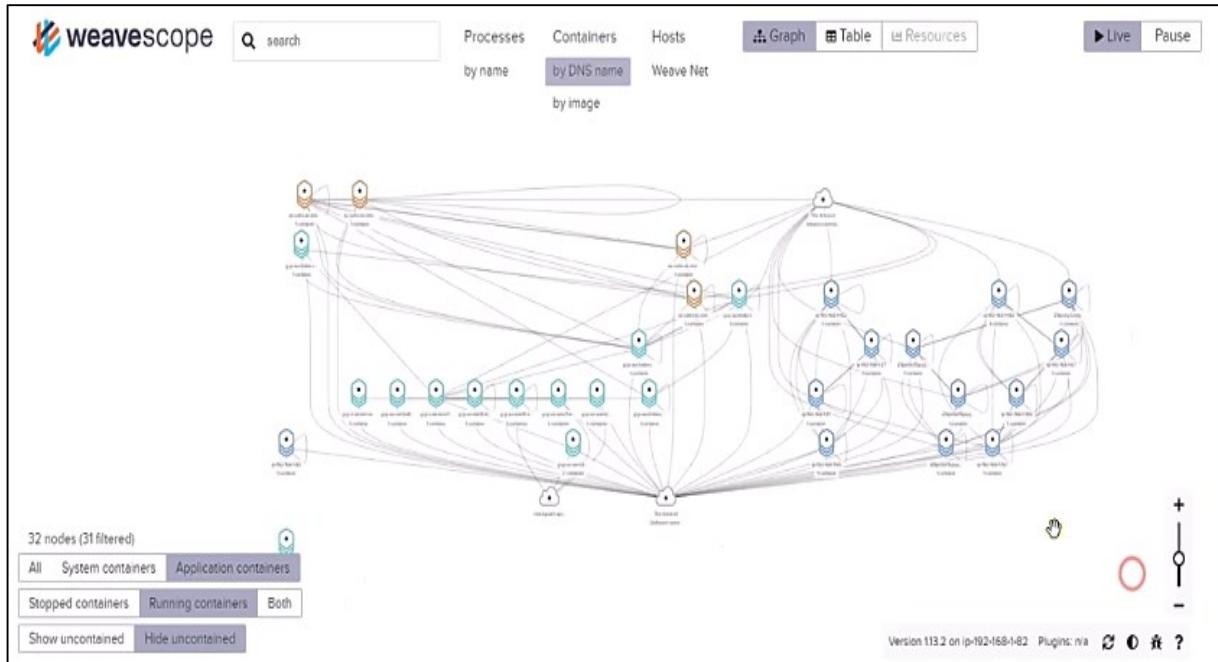
IY=0
AwsApSoutheast1=$((++IY))          # Location: Asia Pacific (Singapore)
AwsCaCentral1=$((++IY))            # Location: Canada (Central)
AwsUsWest1=$((++IY))                # Location: US West (N. California)
AwsUsEast1=$((++IY))                # Location: US East (N. Virginia)
AwsApNortheast1=$((++IY))           # Location: Asia Pacific (Tokyo)
AwsApSouth1=$((++IY))              # Location: Asia Pacific (Mumbai)
AwsApSoutheast2=$((++IY))           # Location: Asia Pacific (Sydney)
AwsEuWest2=$((++IY))                # Location: Europe (London)
AwsUsEast2=$((++IY))                # Location: US East (Ohio)
AwsUsWest2=$((++IY))                # Location: US West (Oregon)
AwsApNortheast3=$((++IY))           # Location: Asia Pacific (Osaka)
AwsEuCentral1=$((++IY))             # Location: Europe (Frankfurt)
AwsEuWest1=$((++IY))                # Location: Europe (Ireland)
AwsEuWest3=$((++IY))                # Location: Europe (Paris)
AwsEuNorth1=$((++IY))               # Location: Europe (Stockholm) - No t2.xxx Specs. t3 c5 m5 r5 ... are available
AwsSaEast1=$((++IY))                # Location: South America (São Paulo)
AwsApNortheast2=$((++IY))           # Location: Asia Pacific (Seoul)
AwsApEast1=$((++IY))                # Location: Asia Pacific (Hong Kong)
AwsMeSouth1=$((++IY))               # Location: Middle East (Bahrain)
AwsAfSouth1=$((++IY))               # Location: Africa (Cape Town) - Opt-In required
AwsEuSouth1=$((++IY))               # Location: Europe (Milan) - Opt-In required

```

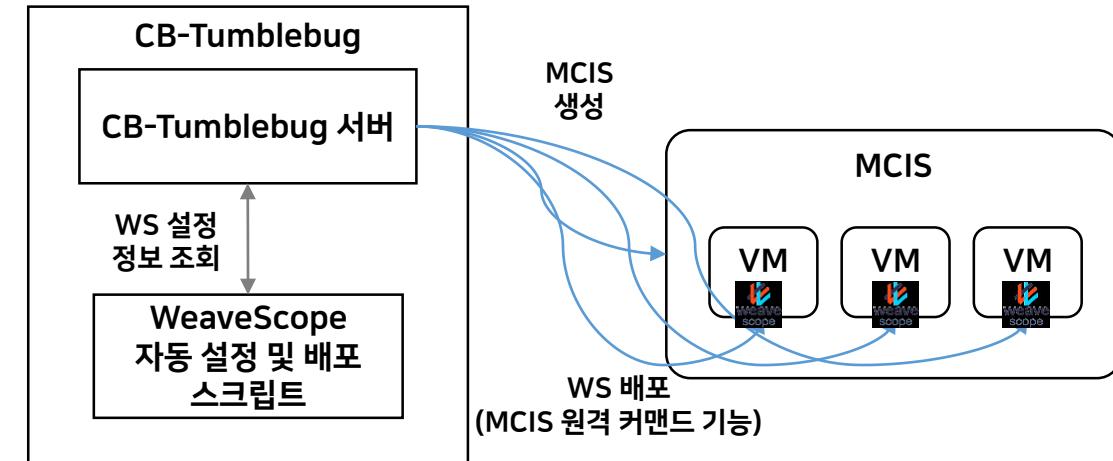
testSet.env

CB-TB 클러스터 모니터링 도구 자동 배치 (유스케이스 발굴 및 배포 자동화)

- CB-Tumblebug의 MCIS에 클러스터 모니터링 도구(WeaveScope) 자동 배치
 - WeaveScope는 컨테이너 클러스터를 모니터링 및 가시화하는 도구
 - CB-Tumblebug의 MCIS에서 WeaveScope 클러스터가 바로 동작하도록 배포 자동화 스크립트 개발
 - MCIS의 형상을 확인하는 데 도움



WeaveScope 동작 형상



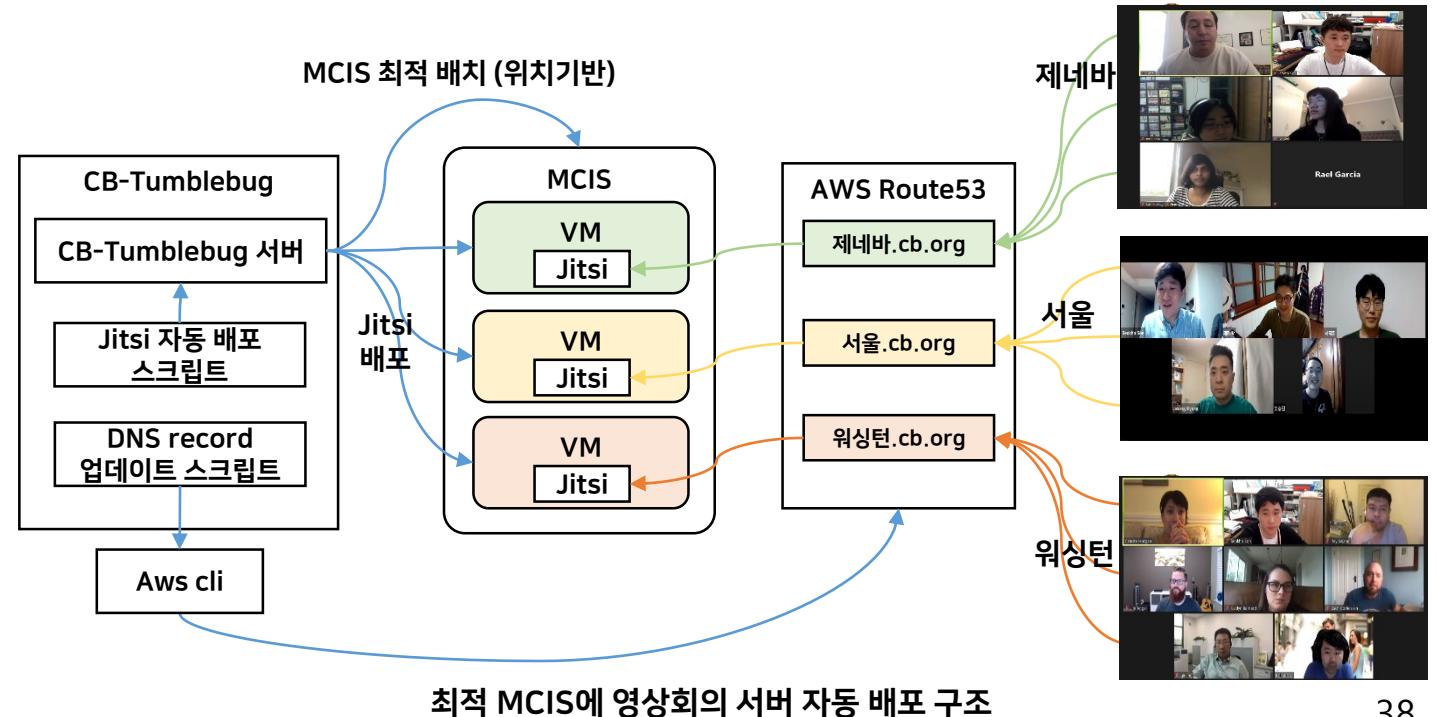
WeaveScope 자동 설정 및 배포 구조

CB-TB 영상 회의 서비스 자동 배포 (유스케이스 발굴 및 배포 자동화)

- CB-Tumblebug 영상 회의 서비스 자동 배포
 - Jitsi는 영상 회의 서버를 제공하는 오픈소스 SW
 - CB-Tumblebug를 통해 MCIS에 Jitsi가 쉽게 배치 될 수 있도록 자동화 스크립트 개발 (단 3개의 명령어로 배포 완료)
- 영상회의는 사용자에게 빠른 응답속도를 제공해야 하므로, 서버의 위치가 중요
 - CB-Tumblebug를 통해서 글로벌 스케일 환경에 최적의 클라우드를 선정하여 영상회의 서비스를 위한 인프라를 제공 가능

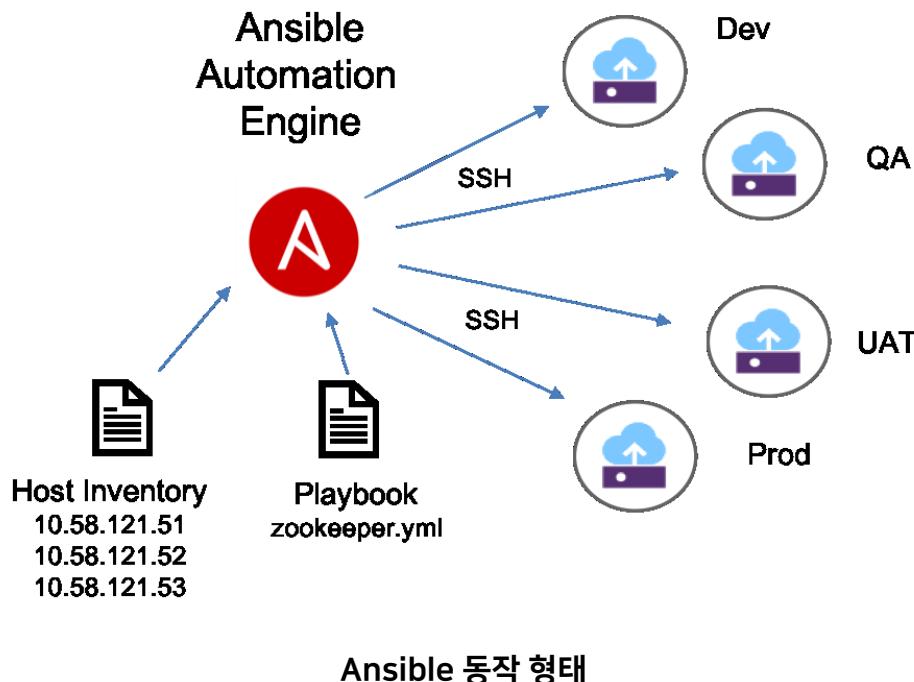


Jitsi 영상회의 접속 화면

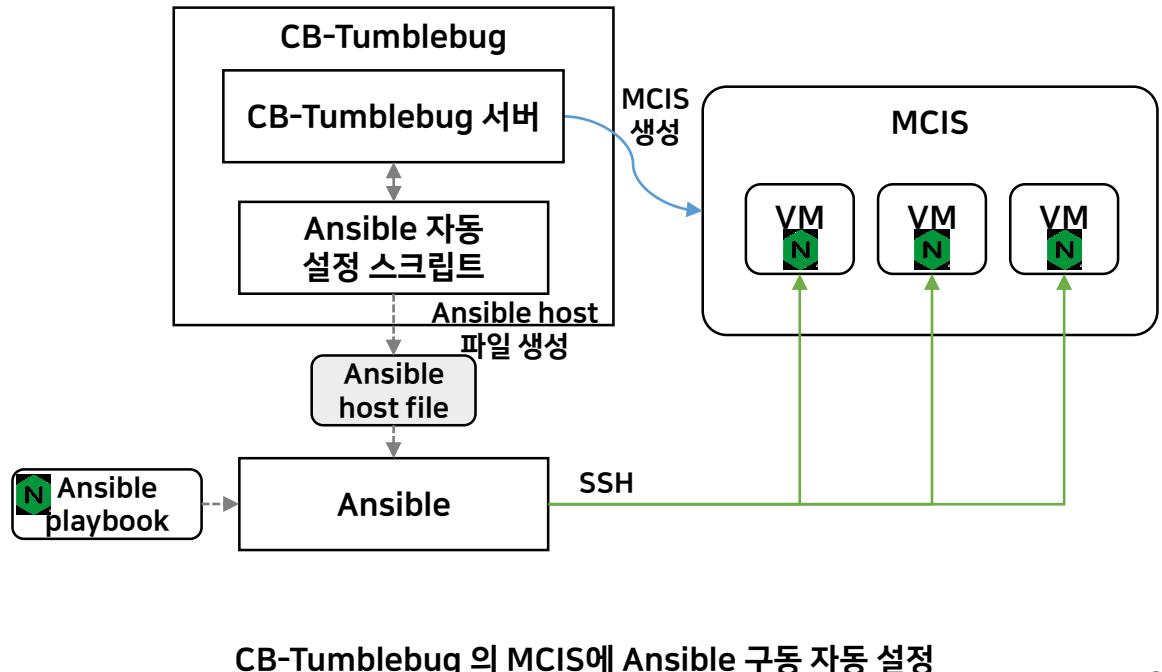


CB-TB의 Ansible 사용자 지원 (유스케이스 발굴 및 배포 자동화)

- CB-Tumblebug의 Ansible 사용자 지원
 - Ansible은 Infrastructure as Code를 지향하는 오픈소스 기반의 자동화 관리 도구
 - 구축/관리 하려는 인프라에 SSH로 원격 명령을 전달하는 방식으로 동작
 - CB-Tumblebug의 MCIS에서 Ansible을 바로 사용할 수 있도록 Ansible 설정 자동화 스크립트 개발

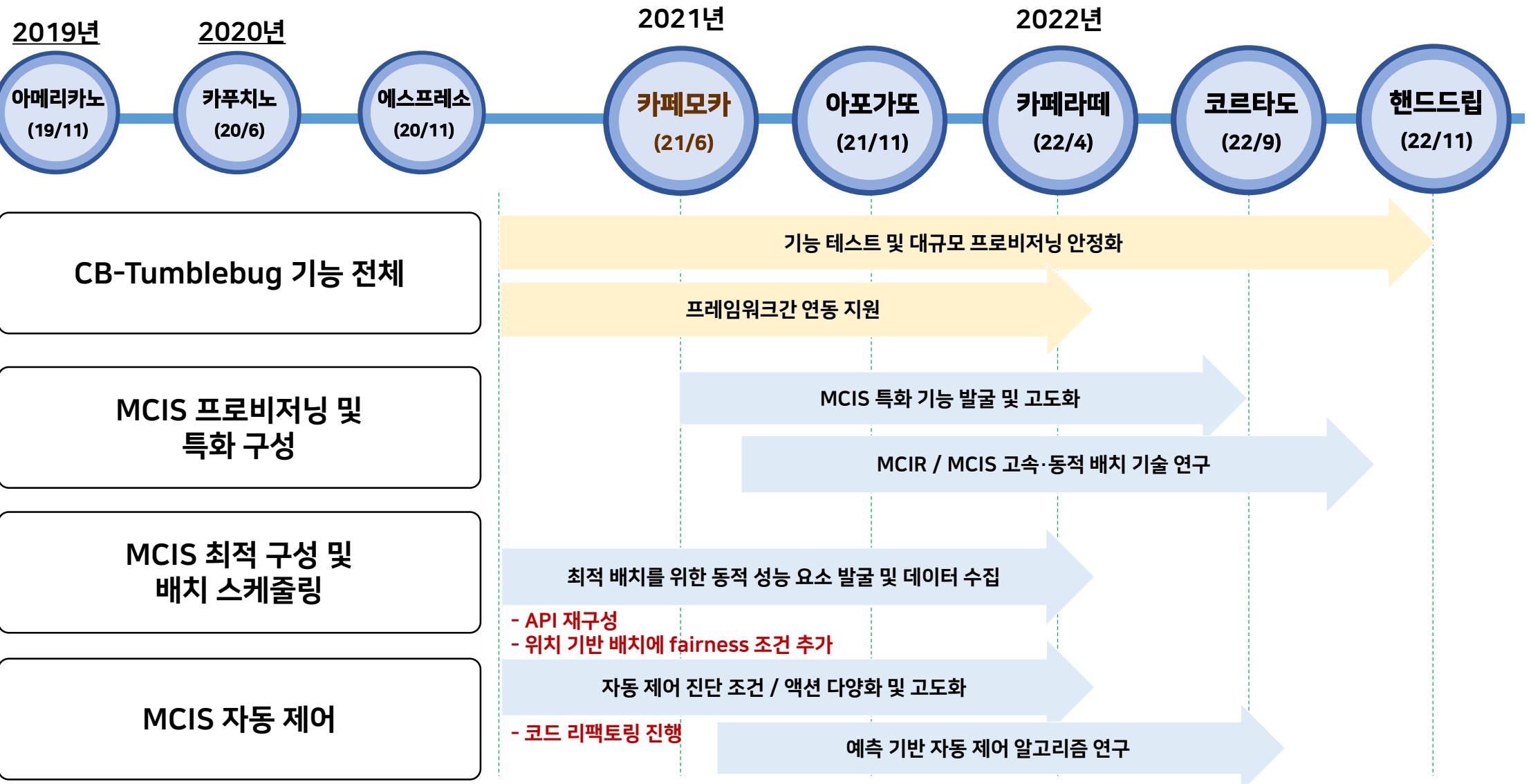


[Source] <https://www.ibm.com/cloud/blog/end-to-end-application-provisioning-with-ansible-and-terraform>



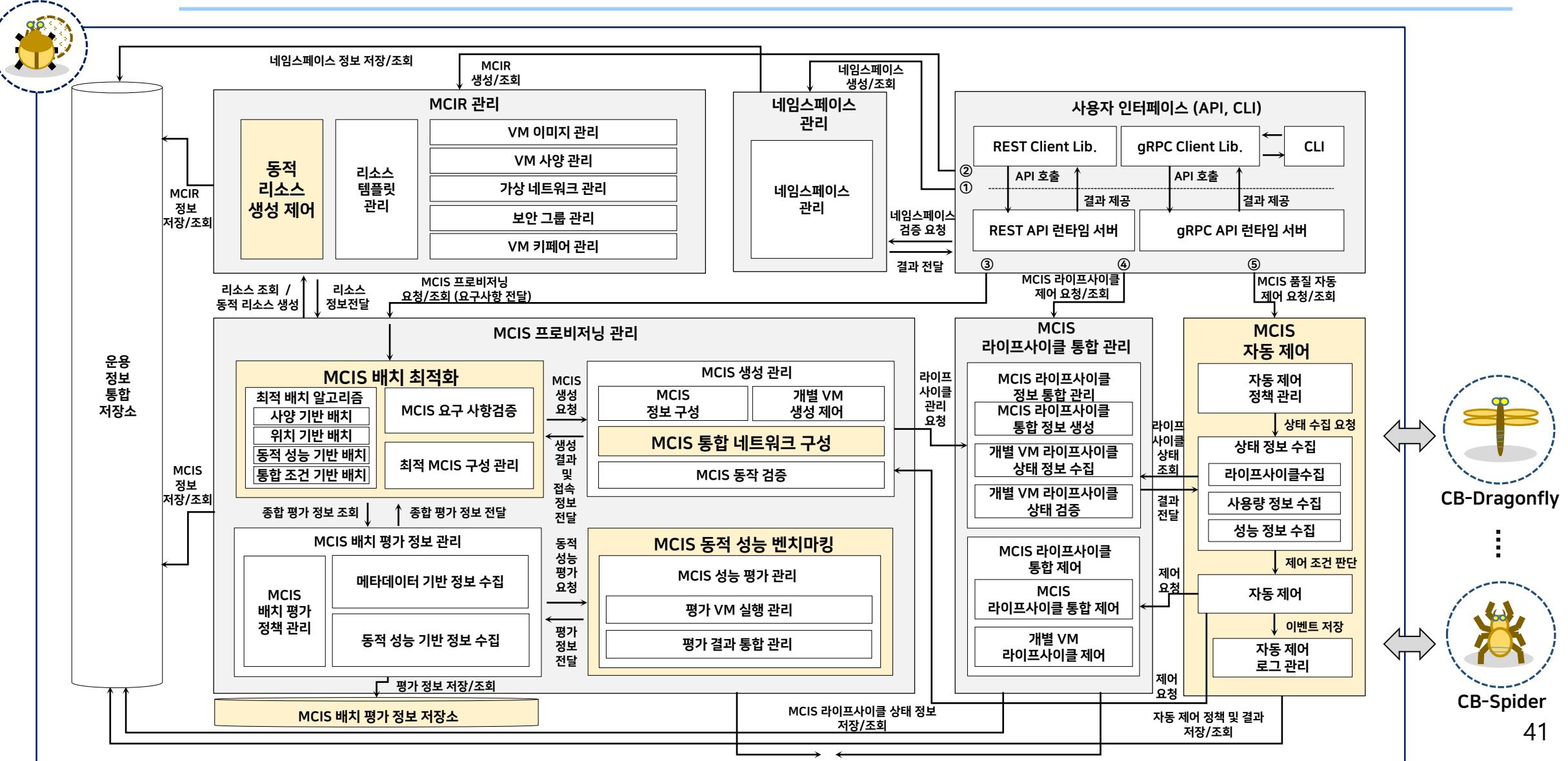
CB-Tumblebug 의 MCIS에 Ansible 구동 자동 설정

CB-Tumblebug 개발 로드맵



[참고] CB-Tumblebug 구조

향후 주요 개선 포인트
(help wanted!)





CB-Tumblebug 참여 방법 (환영합니다..^^)

CB-Tumblebug (Multi-Cloud Infra Service Management)

A sub-system of Cloud-Barista Platform to Deploy and Manage Multi-Cloud Infrastructure.

GitHub stats:

- Code: master, 2 branches, 21 tags
- Issues: 23
- Pull requests: 1
- Discussions: 0
- Actions: 0
- Projects: 1
- Wiki: 0
- Security: 0
- Insights: 0
- Settings: 0

Contributors: 10

Languages: Go 66.2%, Shell 33.6%, Other 0.2%

CB-Tumblebug GitHub
<https://github.com/cloud-barista/cb-tumblebug>

sig-cb-tumblebug

Threads
All DMs
Mentions & reactions
Saved items
More
Starred
sig-cb-tumblebug
Channels
cloud-barista_qna
discussion
etri-discussion
everyone
random
sig-cb-24h
sig-cb-bridge
sig-cb-cicd
sig-cb-cli
sig-cb-docs
sig-cb-domestic_...
sig-cb-dragonfly
sig-cb-ladybug
sig-cb-store
sig-cb-waterstrider
sig-cb-webtool
sig-cb-wip
sig-cloud-twin
sig-grpc
sig-raspberry-pi
sig-release

Message #sig-cb-tumblebug

노영현 3:42 PM WIP 환경에서 MCIS 등록으로 2개의 VM 생성후 MCIS 이를만 바꾸어 똑같은 VM 을 생성했을 때 성공이라고 Message는 날아오지만 실제로는 VM 생성이 되지 않습니다.
해당 MCIS 등을 삭제하려고 했는데 Down된 상태인 것 같습니다.

CB-Tumblebug Slack Channel
<https://cloud-barista.slack.com/archives/CJQ7575PU>

Cloud-Barista SIG-Tumblebug Meeting - 2021/04/15
Seokho Son 59:10

Cloud-Barista SIG-Tumblebug Meeting - 2021/04/01
Seokho Son 1:00:55

Cloud-Barista SIG-Tumblebug Meeting - 2021/03/18
Seokho Son 1:14:19

Cloud-Barista SIG-Tumblebug Meeting - 2021/03/04
Seokho Son 44:33

Cloud-Barista SIG-Tumblebug Meeting - 2021/02/18
Seokho Son 2:02:03

Cloud-Barista SIG-Tumblebug Meeting - 2021/02/04
Seokho Son 1:39:22

Cloud-Barista SIG-Tumblebug Meeting - 2021/01/15
Seokho Son 1:43:34

Cloud-Barista SIG-Tumblebug Meeting - 2021/01/05
Seokho Son 1:35:04

Cloud-Barista SIG-Tumblebug Meeting - 2021/01/04
Seokho Son 1:02:14

CB-Tumblebug 정기회의 (격주)
<https://www.youtube.com/playlist?list=PL9Bd5opncjJAHYr9oPnP8i9kClBacgBw8>

감사합니다.

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

(김 병 섭, 손 석 호 / contact-to-cloud-barista@googlegroups.com)

“멀티클라우드, 글로벌 스케일로 시작하다”

클라우드바리스타들의 네번째 이야기

Cloud-Barista Community the 4th Conference