Cloud Barista 멀티 클라우드 서비스 공통 플랫폼

CB-Tumblebug : 최적 멀티 클라우드 인프라를 찾아서

(멀티 클라우드 인프라 통합 운용 관리)

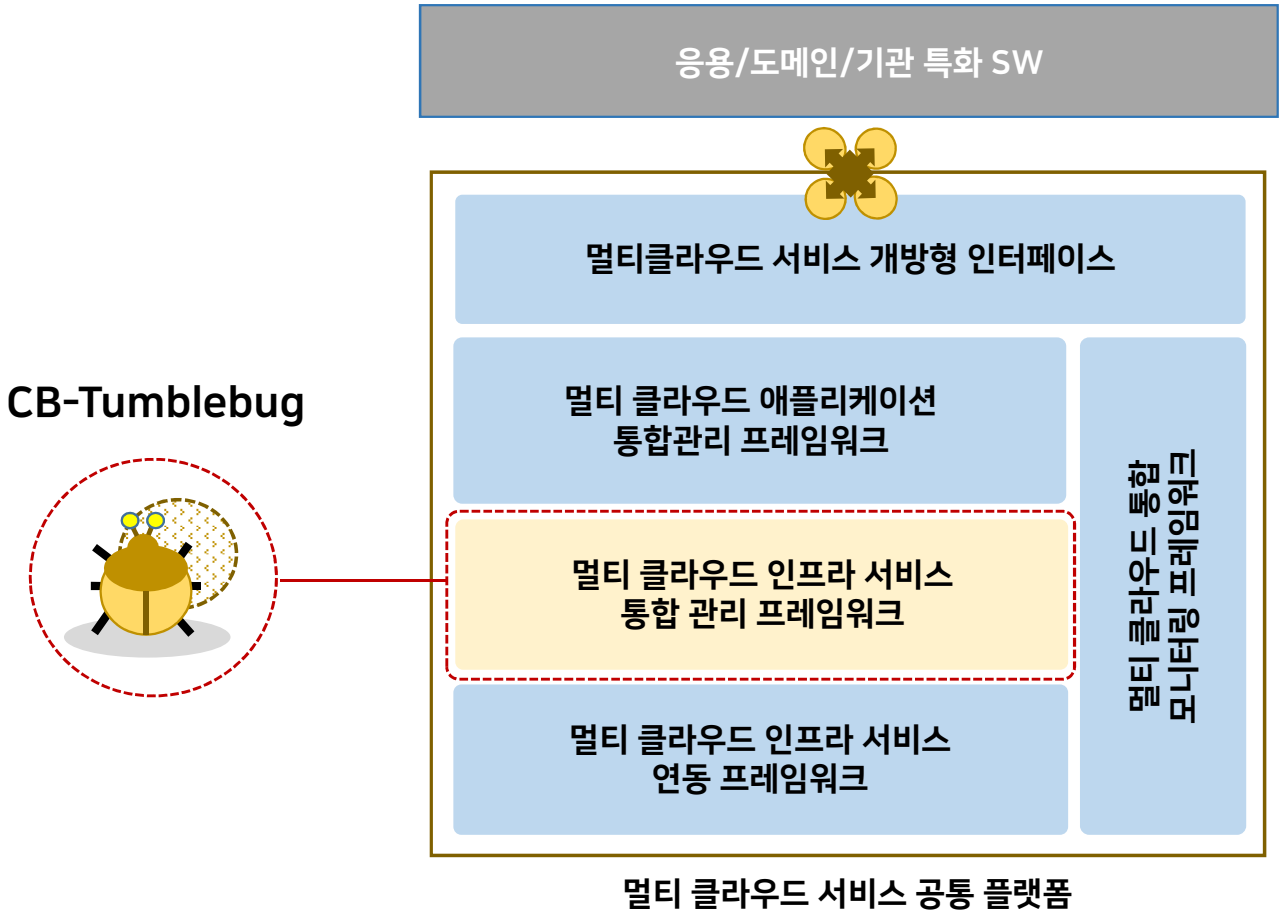
손 석 호 / CB-Tumblebug 프레임워크 리더

“Contact to the Multi-Cloud”

클라우드 바리스타들의 두 번째 이야기

Cloud-Barista Community 2nd Open Conference

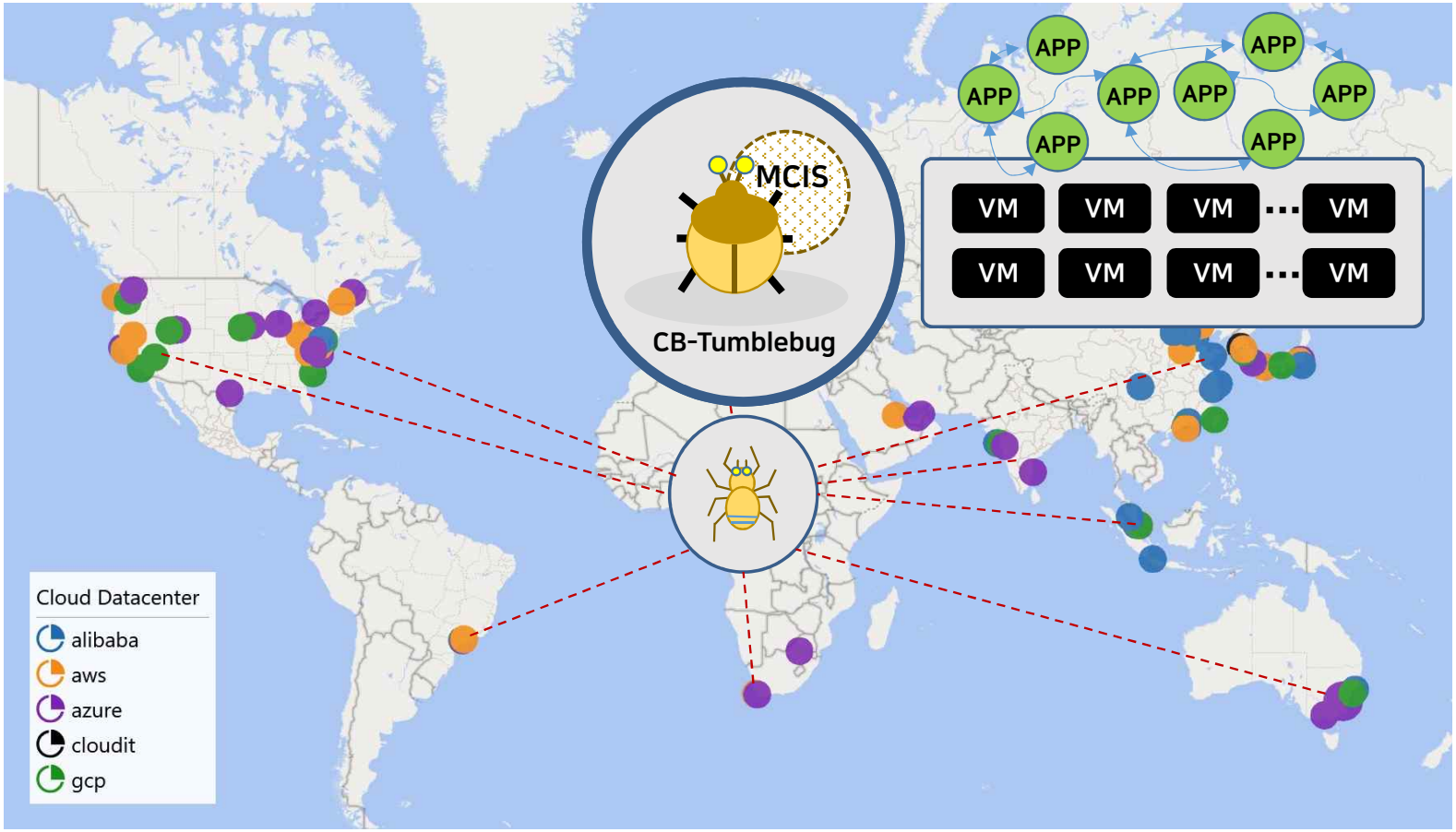
이번 세션은…


응용/도메인/기관 특화 SW
멀티클라우드 서비스 개방형 인터페이스
멀티 클라우드 애플리케이션
통합관리 프레임워크
멀티 클라우드 인프라 서비스
통합 관리 프레임워크
멀티 클라우드 인프라 서비스
연동 프레임워크
멀티클라우드 일관된 운영 프레임워크
CB-Tumblebug
멀티 클라우드 서비스 공통 플랫폼

CLOUD BARISTA

목 차

CB-Tumblebug : 멀티 클라우드 인프라 통합 운용 관리 기술 개요



멀티 클라우드
애플리케이션

멀티 클라우드 자원
통합 컴퓨팅 인프라
멀티 클라우드 인프라 서비스 (MCIS)

컴퓨팅 자원 | 컴퓨팅 자원 | 컴퓨팅 자원

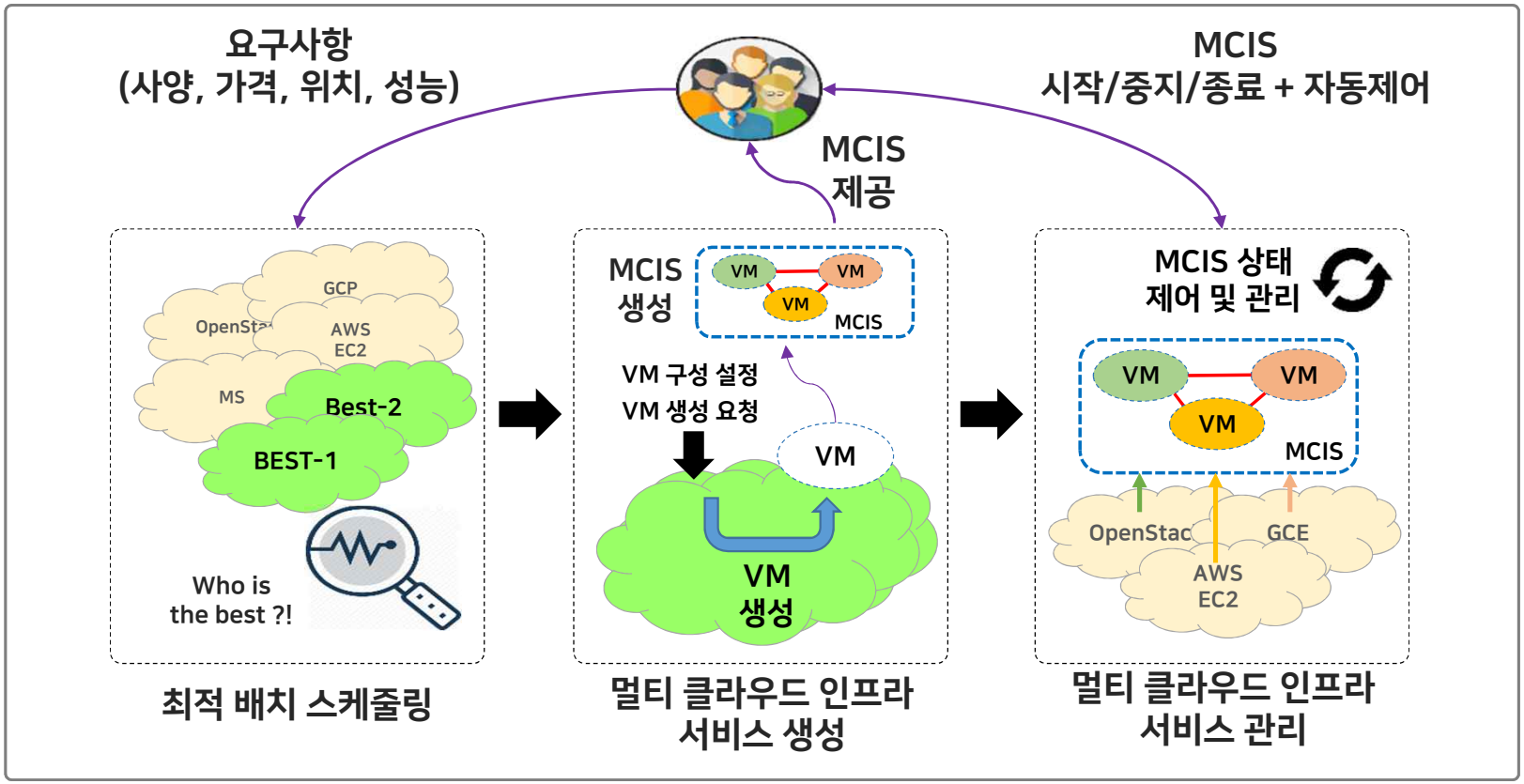
이종 멀티 클라우드 연동

멀티 클라우드 컴퓨팅 자원의 효과적인 활용을 위해, 자원을 유기적으로 통합 제공하는 멀티 클라우드 인프라 서비스 필요

멀티 클라우드 인프라 서비스 통합 운용 관리 프레임워크 기술 정의

기술 정의

사용자 요구사항에 따라, 멀티 클라우드의 인프라 서비스를 조합 및 프로비저닝하고,
멀티 클라우드 인프라 서비스를 통합 관리하여 사용자의 인프라 운영을 지원하는 기술



다양한 성능 벤치마킹

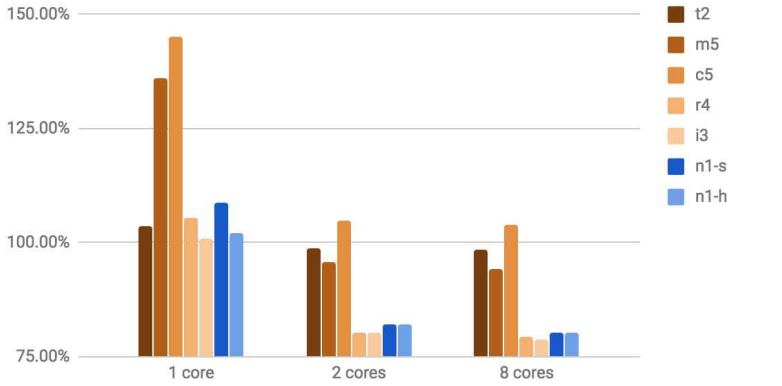
정보를 제공하여
최적 클라우드 인프라 제공

컴퓨팅 서비스들을 인프라
서비스 그룹으로 통합관리

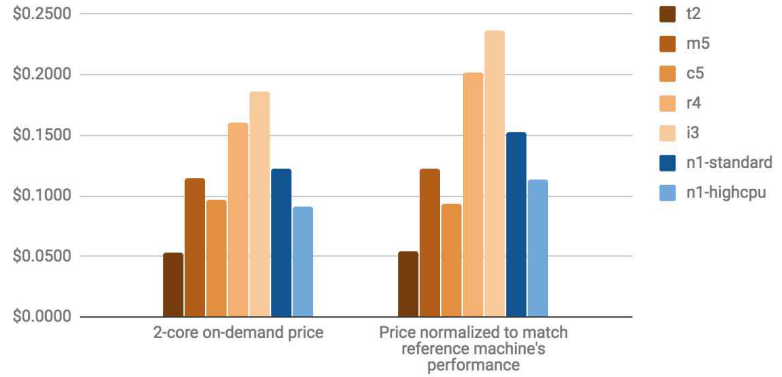
통합 제어, 정책 적용 등
관리 편의성을 극대화

CB-Tumblebug 특징

클라우드 서비스 성능.. 알고 보면 많이 달라요

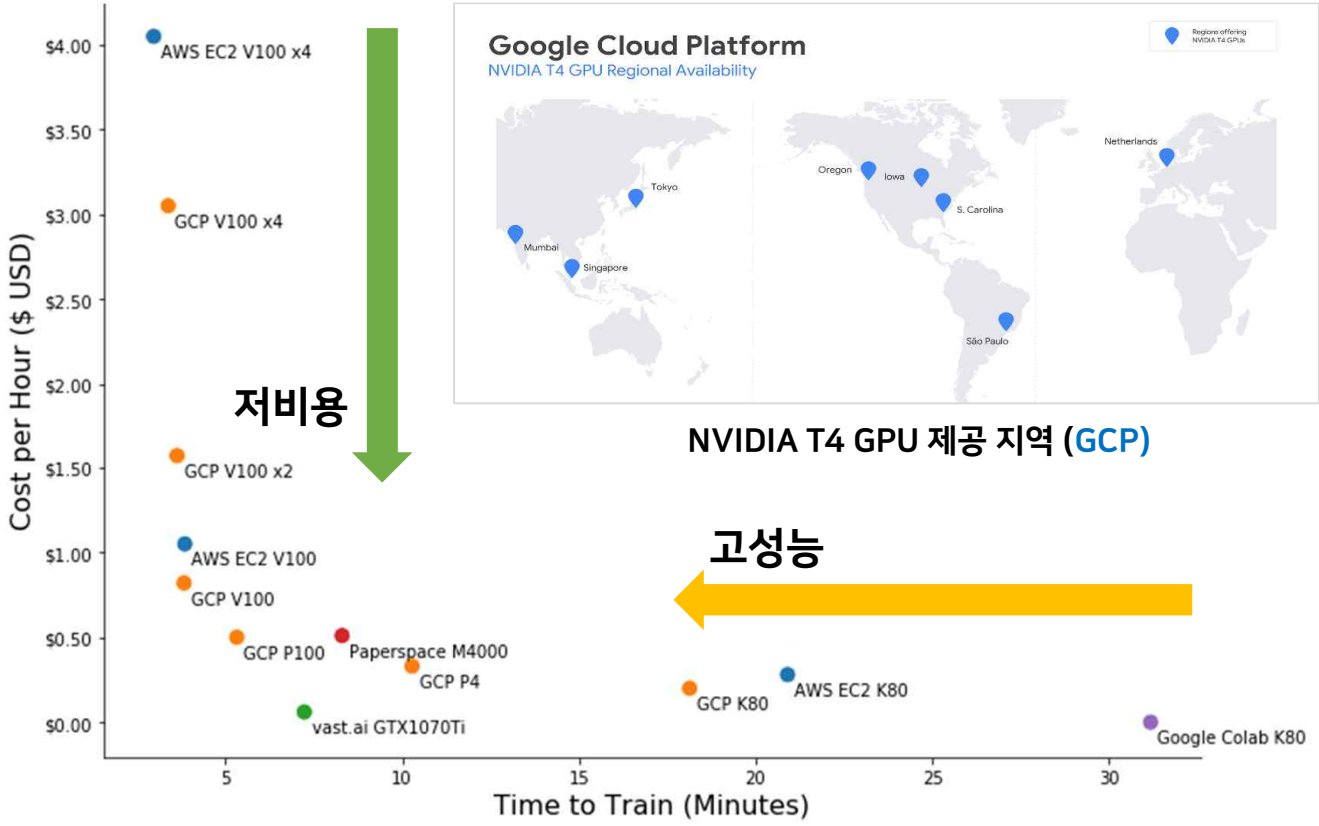


VM 성능 비교 (AWS vs GCP)



VM 가격 비교 (AWS vs GCP)

(https://medium.com/infrastructure-adventures/aws-vs-gcp-vs-on-premises-cpu-



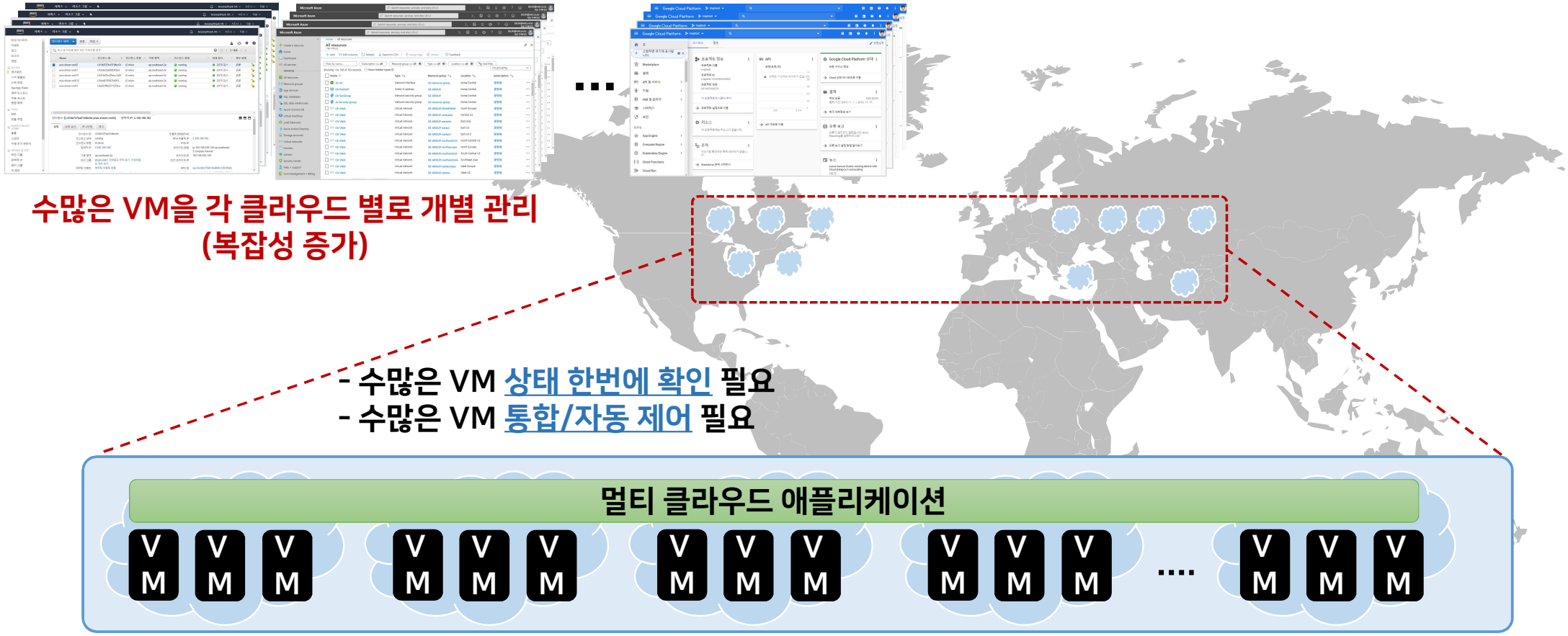
NVIDIA T4 GPU 제공 지역 (GCP)

GPU 성능 및 비용 비교 (AWS vs GCP)

(https://towardsdatascience.com/maximize-your-gpu-dollars-a9133f4e546a by Jeff Hale)

멀티 클라우드에서는 자원들의 성능 및 특성이 매우 다양하므로, 최적의 멀티 클라우드 인프라 서비스 제공 필요

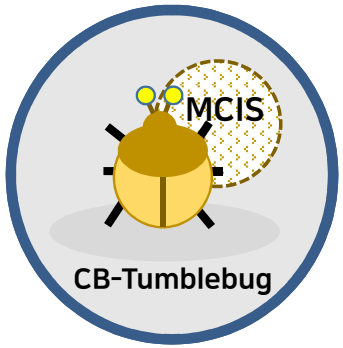
수많은 컴퓨팅 머신을 클라우드마다 개별 관리한다면... OTL..



멀티 클라우드 환경에서는 단일 클라우드에 비해, 관리의 대상이 많고 복잡 => 인프라 통합 관리/자동 제어 기술 필요

CB-Tumblebug 주요 기능

- 멀티 클라우드 인프라 서비스 (MCIS)
  - 지역적으로 격리된 다수의 클라우드 상에서 단일 목적(응용서비스, 애플리케이션 등)을 위해 상호 연계된 하나 이상의 클라우드 인프라 서비스(가상머신(VM) 등) 그룹
- 멀티 클라우드 인프라 자원 (MCIR)
  - 다수의 클라우드 상에서 제공되는 MCIS를 생성 및 운용하기 위한 클라우드 인프라 자원들
    - 클라우드 인프라 자원: 이미지, VM 사양, 네트워크, VM 접속을 위한 자원 등



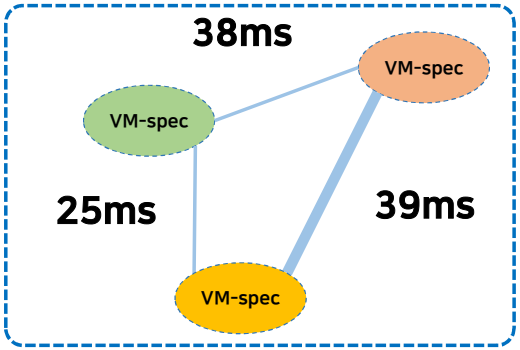
가상 서버 타입 동적 성능 평가	MCIS 최적 구성 및 스케줄링 기능	네임스페이스& MCIR 관리 기능	MCIS 프로비저닝 기능	MCIS 특화 기능	MCIS 라이프사이클 관리 기능	MCIS 자동 제어 기능
정보 수집	배치 계획	자원 준비	배치 수행	인프라 특화	인프라 운용	관리 자동화
[1] 인프라 배치 계획 단계		[2] 인프라 배치 수행 및 특화 단계			[3] 인프라 운용 및 관리 자동화 단계	

CB-Tumblebug는 크게 3 단계로 세부 기능을 분류 가능. [1] 배치 계획, [2] 배치 수행 및 특화, [3] 운용 및 관리 자동화

[1] 인프라 배치 계획

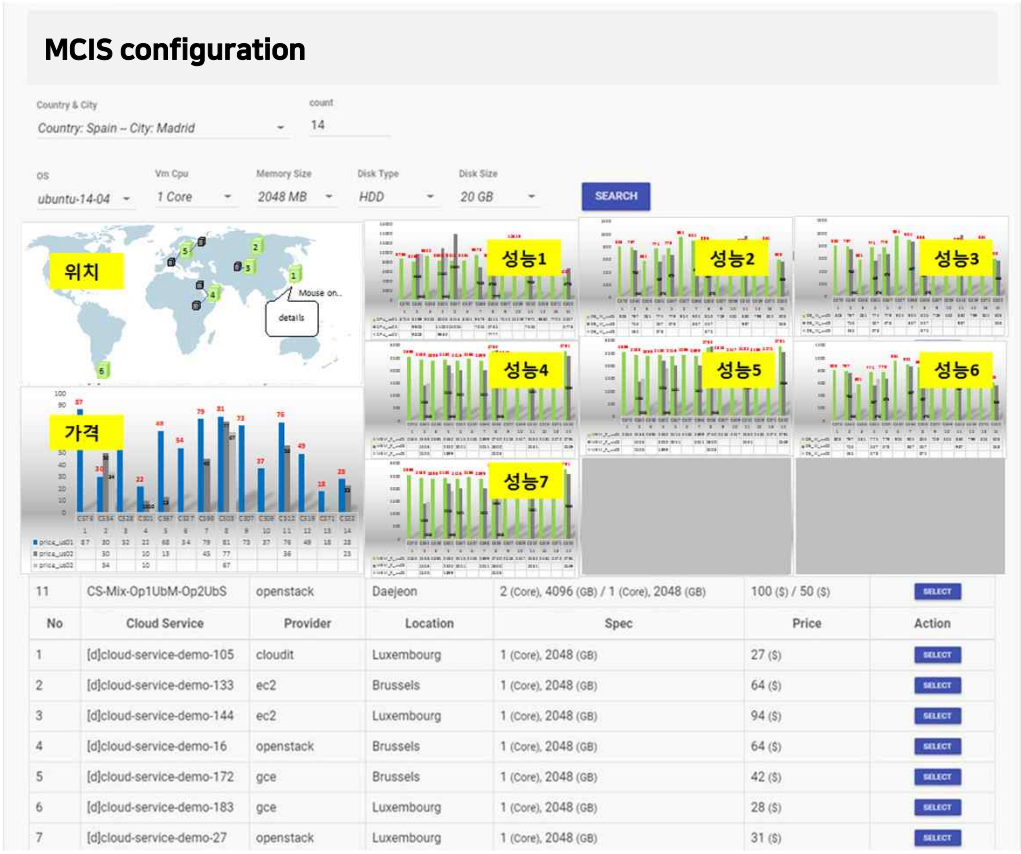
가상 서버 타입 동적 성능 평가	MCIS 최적 구성 및 스케줄링 기능
정보 수집	배치 계획

- **MCIS 통합 최적 배치**
  - VM 간 응답속도 기반 배치



예) VM 간 응답시간 < 40ms

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



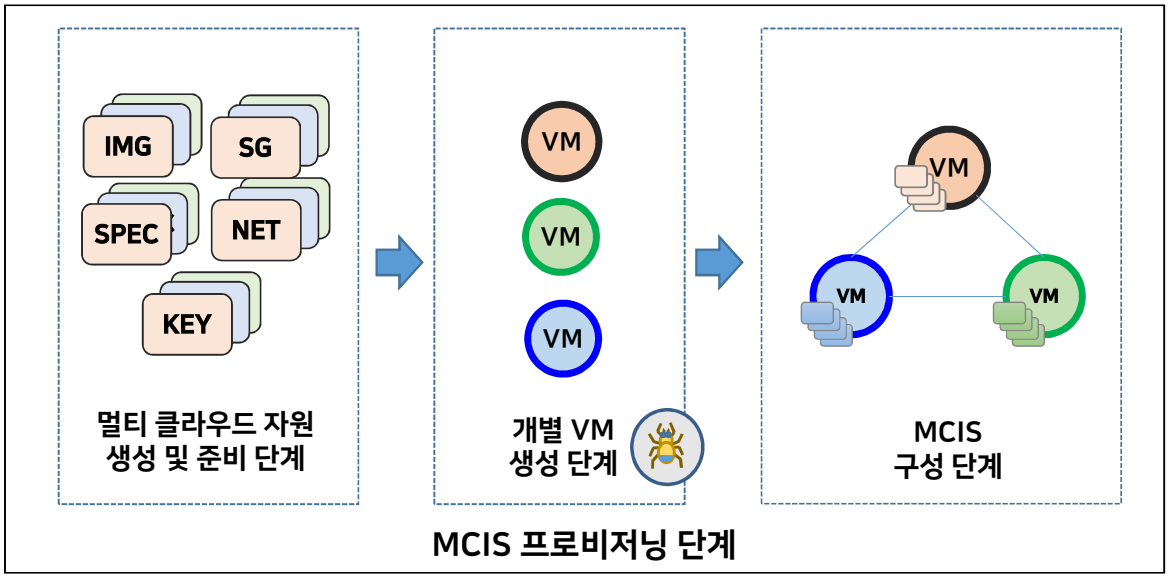
사용자 요구사항

인프라 서비스 평가

배치 우선순위

[2] 인프라 배치 수행 및 특화 단계 (1/2)

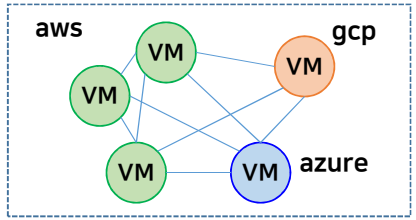
네임스페이스& MCIR 관리 기능	MCIS 프로비저닝 기능	MCIS 특화 기능
자원 준비	배치 수행	인프라 특화



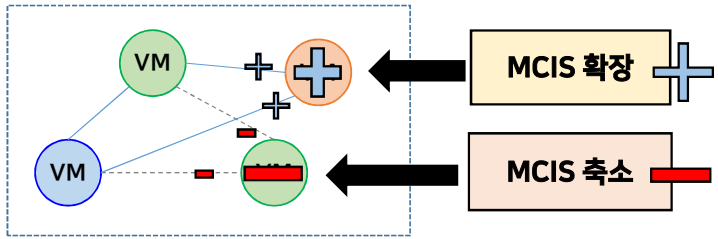
MCIS 프로비저닝 단계

멀티 클라우드 환경

[MCIS 구성 예시]



3종 클라우드의 VM 5개의 MCIS

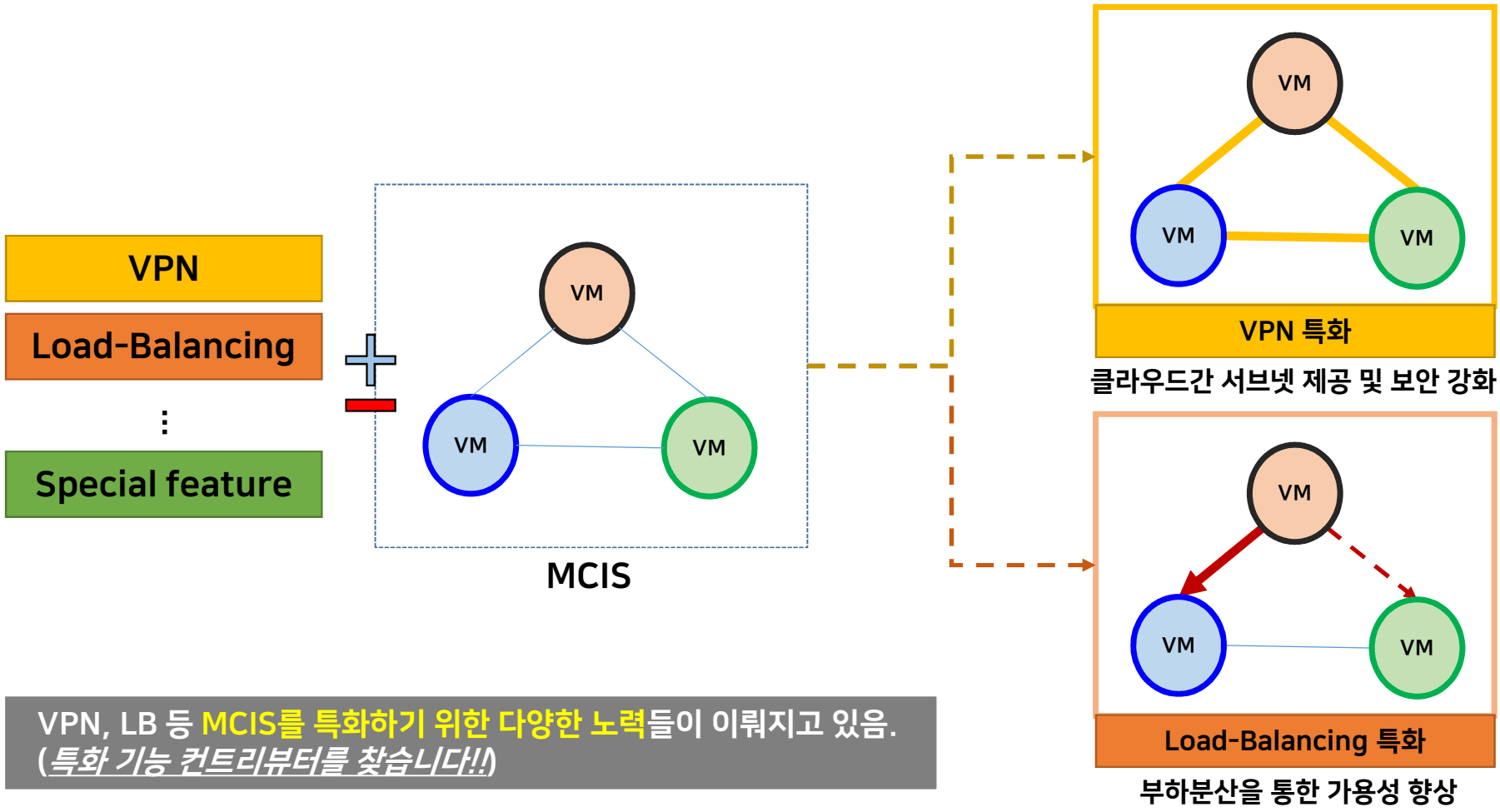


MCIS 확장 및 축소

다중 가상 서버를 담을 수 있는 논리적인 정보 객체 MCIS를 제시

[2] 인프라 배치 수행 및 특화 단계 (2/2)

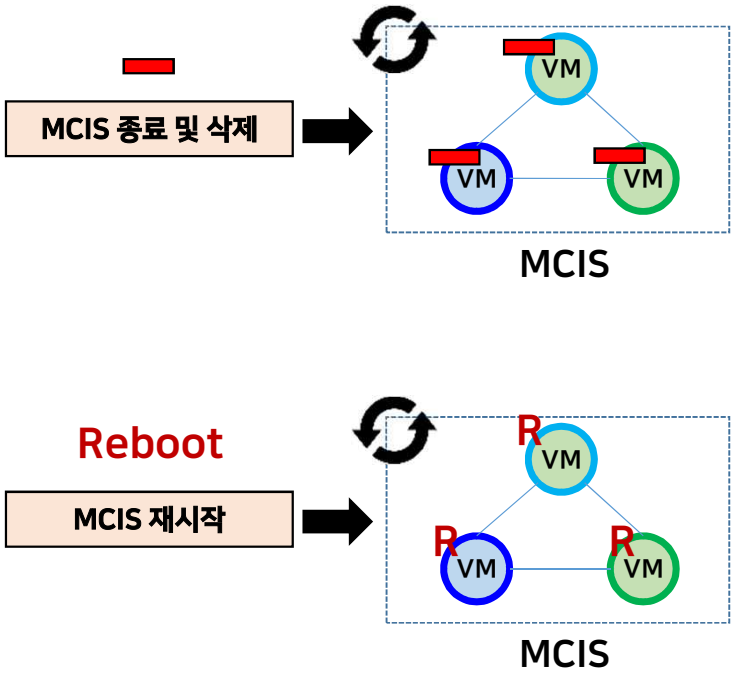
네임스페이스& MCIR 관리 기능	MCIS 프로비저닝 기능	MCIS 특화 기능
자원 준비	배치 수행	인프라 특화



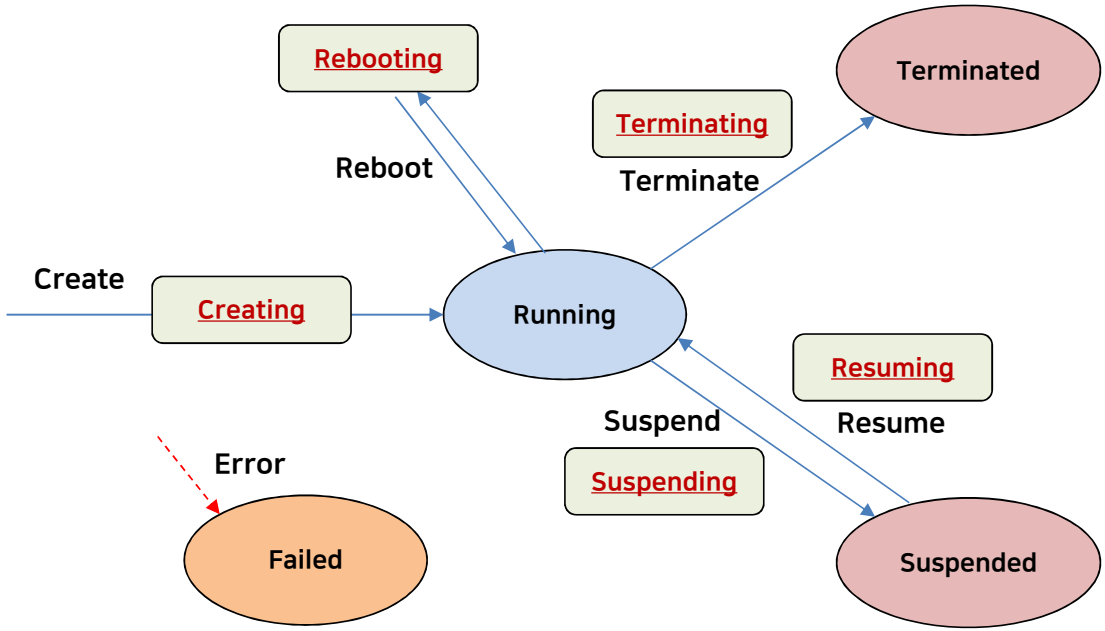
VPN, LB 등 MCIS를 특화하기 위한 다양한 노력들이 이뤄지고 있음.
(*특화 기능 컨트리뷰터를 찾습니다!!*)

[3] 인프라 운용 및 관리 자동화 단계

MCIS 라이프사이클 관리 기능	MCIS 자동 제어 기능
인프라 운용	관리 자동화



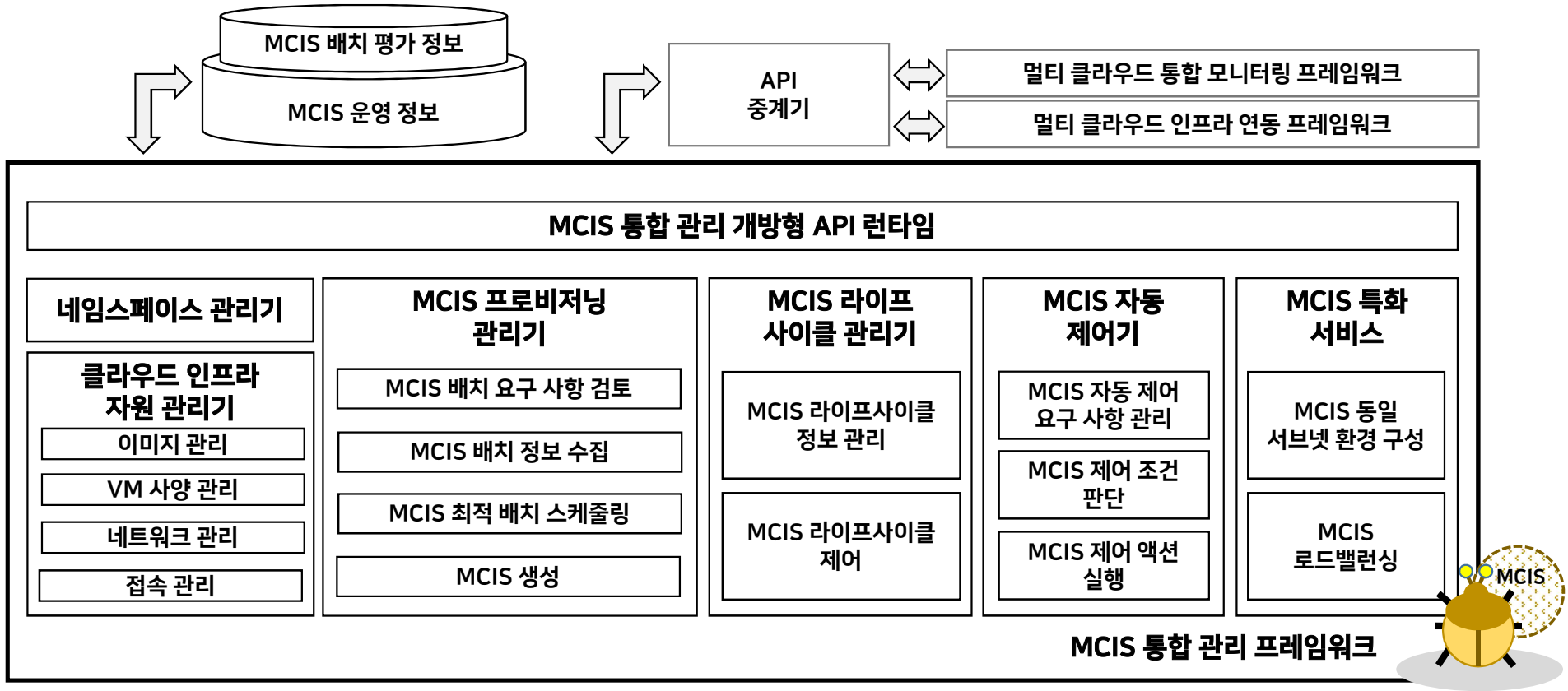
MCIS 단위 제어 = VM 통합 제어



Failed	R	R	R	F	R	R	R	R	R	R
Partial-Running	R	R	R	R	R	R	R	S	S	S

MCIS 라이프사이클 상태도

CB-Tumblebug 기능 구조 (참고)



Cappuccino 릴리스 핵심 포인트

 : 카푸치노 릴리스 핵심포인트

- 최적 MCIS 배치를 위한 동적 성능 벤치마킹 PoC
  - 다양한 클라우드 서비스 사업자가 있으며, 각 사업자는 클라우드의 Region(서울, 대전, ..)을 늘려 나가고 있음
  - 각 지역에서는 사용할 수 있는 가상 서버의 Spec(사양)이 다르며, 성능도 상이
  - 동적 성능 벤치마킹을 통해서 가상 서버 선택의 어려움을 줄이고 효율 향상이 가능
- MCIS 라이프사이클 상태 관리 기능 개선 릴리스
  - MCIS 는 여러 개의 가상 서버를 포함, 종합적인 상태 명시 방법 고안 필요
  - 클라우드 서비스 사업자 마다, 라이프사이클 상태 처리 방법과 표기가 다르므로, 정확한 상태 파악이 어려움
    - 개선: 현재 MCIS 제어 액션을 인지(DB에 저장)하여, 그 정보를 기준으로 오류가 되는 상태 데이터를 보정

가상 서버 타입 동적 성능 평가 | MCIS 최적 구성 및 스케줄링 기능

정보 수집 | 배치 계획

[1] 인프라 배치 계획 단계

네임스페이스& MCIR 관리 기능 | MCIS 프로비저닝 기능 | MCIS 특화 기능

자원 준비 | 배치 수행 | 인프라 특화

[2] 인프라 배치 수행 및 특화 단계

MCIS 라이프사이클 관리 기능 | MCIS 자동 제어 기능

인프라 운용 | 관리 자동화

[3] 인프라 운용 및 관리 자동화 단계

최적 MCIS 배치를 위한 동적 성능 벤치마킹 PoC

릴리스
핵심포인트 [1]

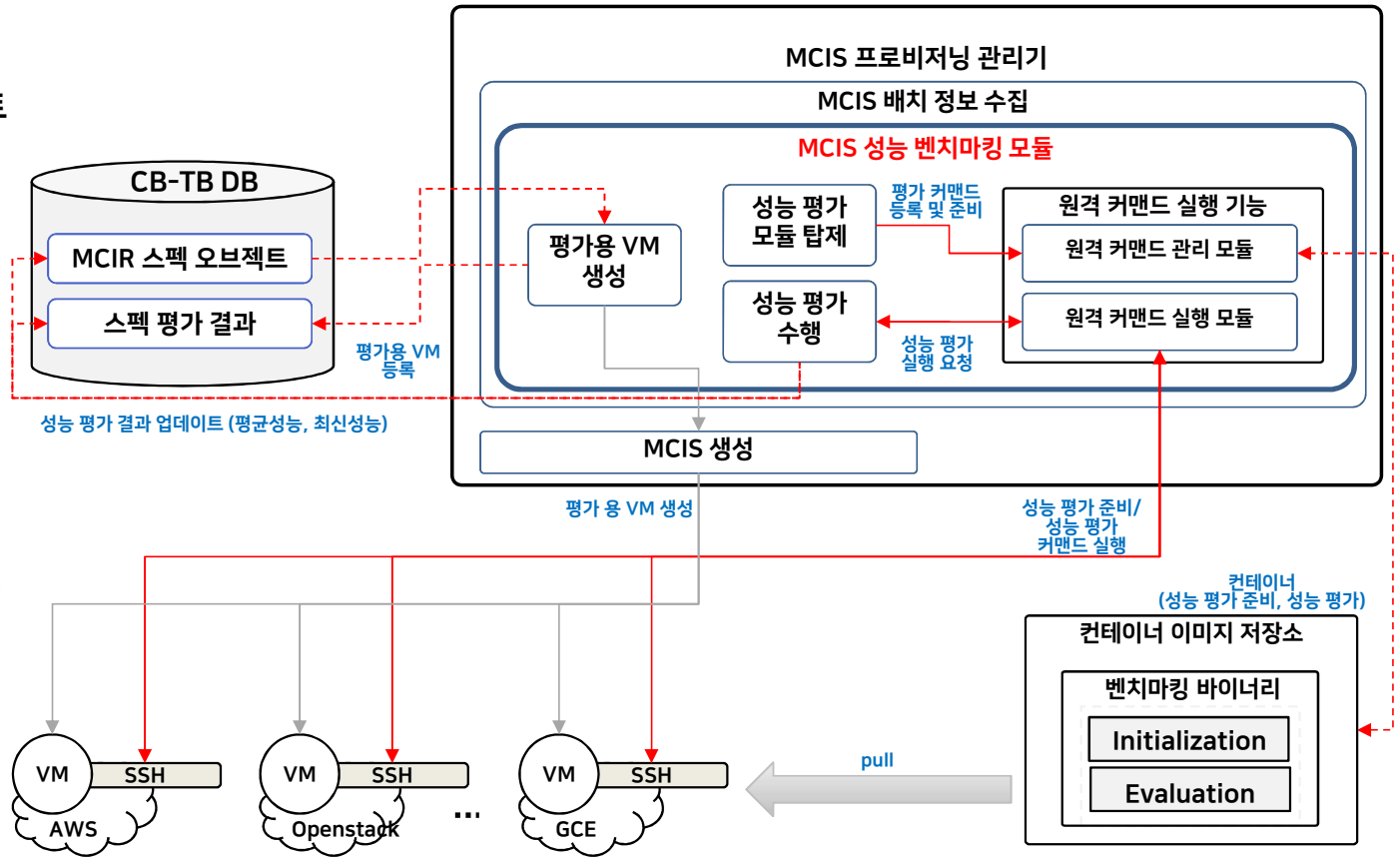
- 에이전트 기반의 동적 성능 벤치마킹
  1. Tumblebug은 각 VM에 벤치마킹 에이전트 (CB-Milkyway) 원격 설치
  2. 주기적으로 Milkyway에 평가 요청
  3. 각 Milkyway는 성능 평가를 수행하고 결과를 Tumblebug에게 통보

- 동적 성능 평가 지표
  - CPU 계산속도 (싱글/멀티 코어)
  - Memory 처리속도 (읽기/쓰기)
  - FileIO 처리속도 평가 (읽기/쓰기)
  - DB 처리속도 평가 (읽기/쓰기)

  Sysbench

  - 목적지까지 네트워크 지연 시간
  - VM간 상호 네트워크 지연 시간

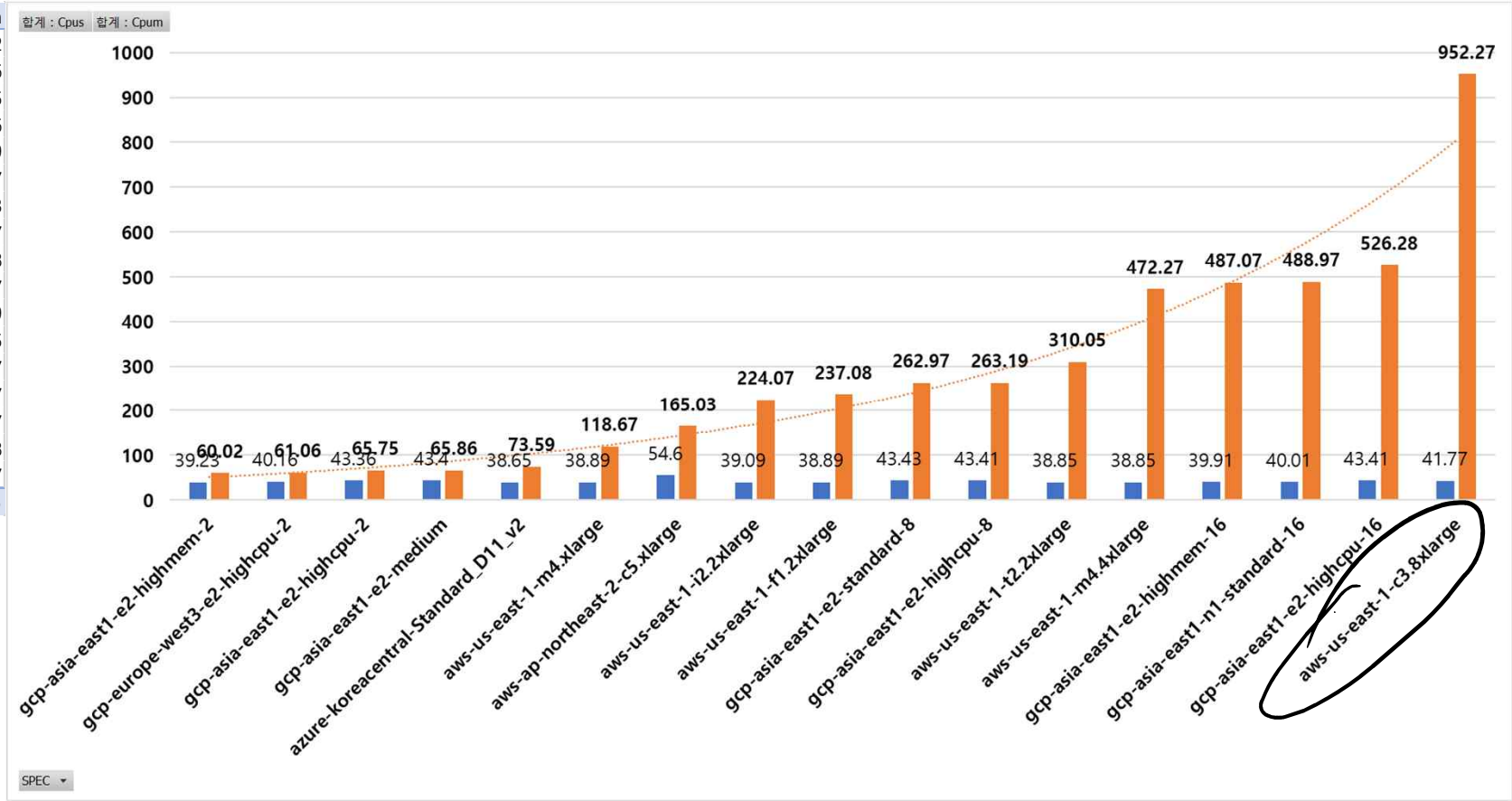
  Ping



에이전트 기반의 동적 멀티 클라우드 컴퓨팅 성능 벤치마킹 구조

동적 성능 평가 결과 (예 : CPU 성능)

행 레이블	평균 : Cpus	평균 : Cpum
gcp-asia-east1-e2-highmem-2	39.23	60.02
gcp-europe-west3-e2-highcpu-2	40.16	61.06
gcp-asia-east1-e2-highcpu-2	43.36	65.75
gcp-asia-east1-e2-medium	43.4	65.86
azure-koreacentral-Standard_D11_v2	38.65	73.59
aws-us-east-1-m4.xlarge	38.89	118.67
aws-ap-northeast-2-c5.xlarge	54.6	165.03
aws-us-east-1-i2.2xlarge	39.09	224.07
aws-us-east-1-f1.2xlarge	38.89	237.08
gcp-asia-east1-e2-standard-8	43.43	262.97
gcp-asia-east1-e2-highcpu-8	43.41	263.19
aws-us-east-1-t2.2xlarge	38.85	310.05
aws-us-east-1-m4.4xlarge	38.85	472.27
gcp-asia-east1-e2-highmem-16	39.91	487.07
gcp-asia-east1-n1-standard-16	40.01	488.97
gcp-asia-east1-e2-highcpu-16	43.41	526.28
aws-us-east-1-c3.8xlarge	41.77	952.27
총합계	41.52	284.36



- Prime Number 계산 Throughput 평가
  - 싱글 코어
  - 멀티 풀 코어

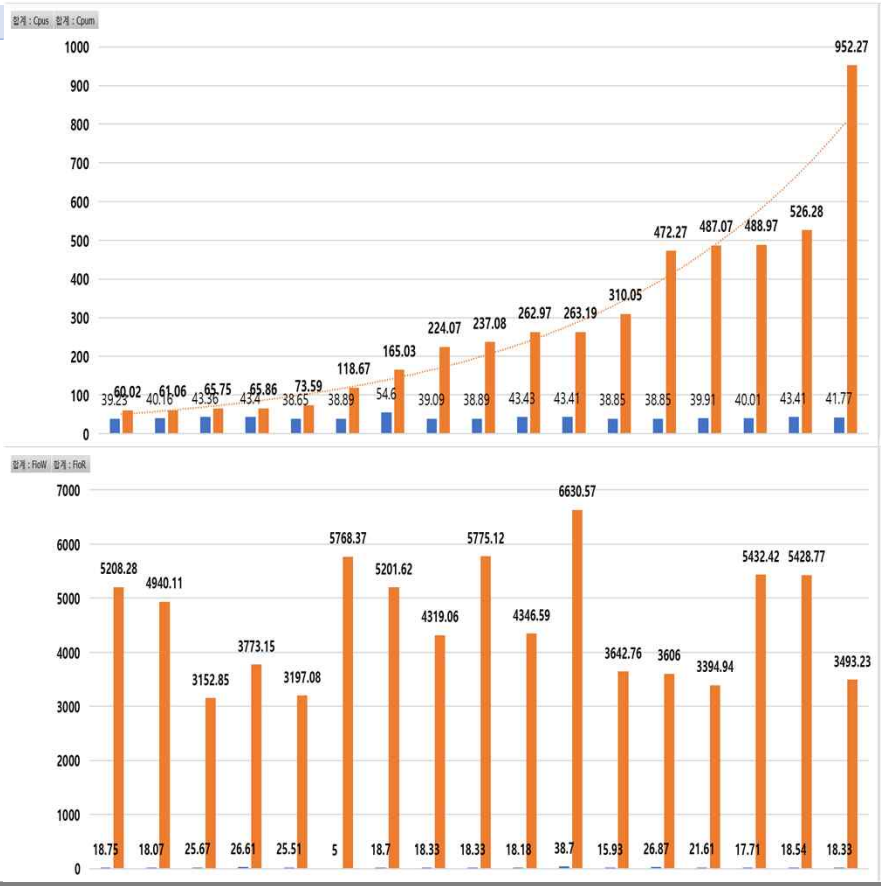
가상 서버의 사양에 따라서 CPU 계산 속도 차이가 발생하는 것을 확인 가능 (vCPU수가 낮아도 높은 성능 내는 경우도 발생)

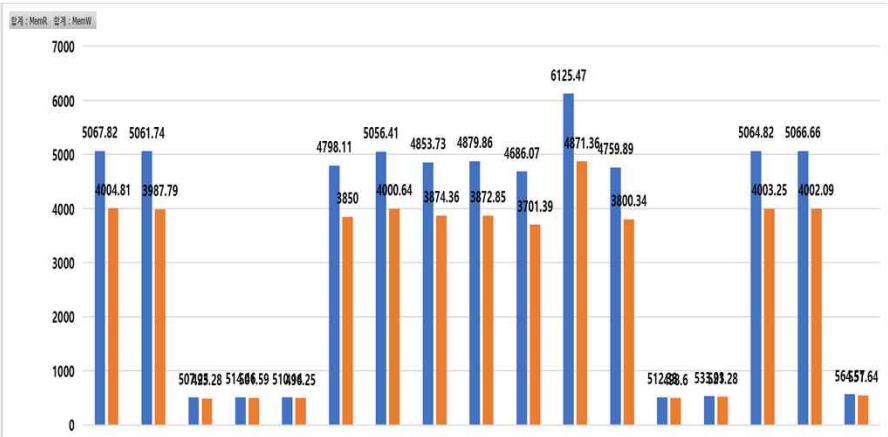
동적 성능 평가 결과 (예: CPU, Memory, FIO, DB 성능)

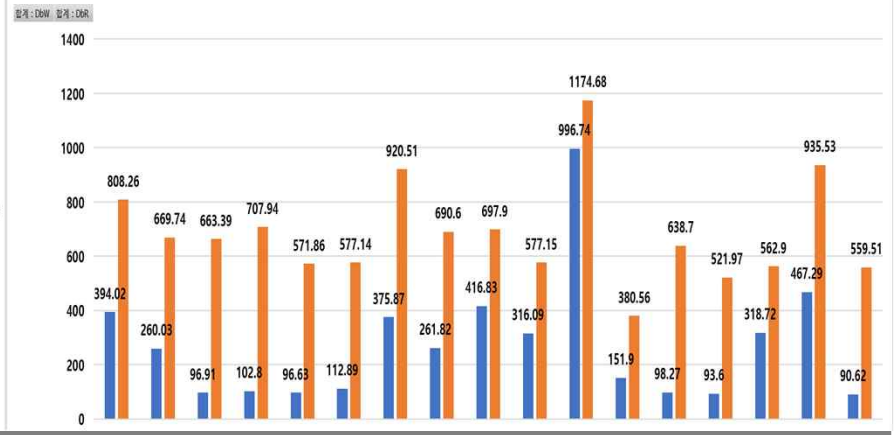
멀티 클라우드 VM 타입 (지역구분)

gcp-asia-east1-e2-standard-8
gcp-asia-east1-e2-highcpu-2
aws-us-east-1-m4.4xlarge
aws-us-east-1-f1.2xlarge
aws-us-east-1-m4.xlarge
azure-koreacentral-Standard_D11_v2
gcp-asia-east1-e2-highcpu-8
gcp-asia-east1-e2-highmem-16
gcp-asia-east1-n1-standard-16
gcp-europe-west3-e2-highcpu-2
aws-ap-northeast-2-c5.xlarge
gcp-asia-east1-e2-highmem-2
aws-us-east-1-t2.2xlarge
aws-us-east-1-i2.2xlarge
gcp-asia-east1-e2-medium
gcp-asia-east1-e2-highcpu-16
aws-us-east-1-c3.8xlarge

- CPU
- Memory
- FIO
- DB



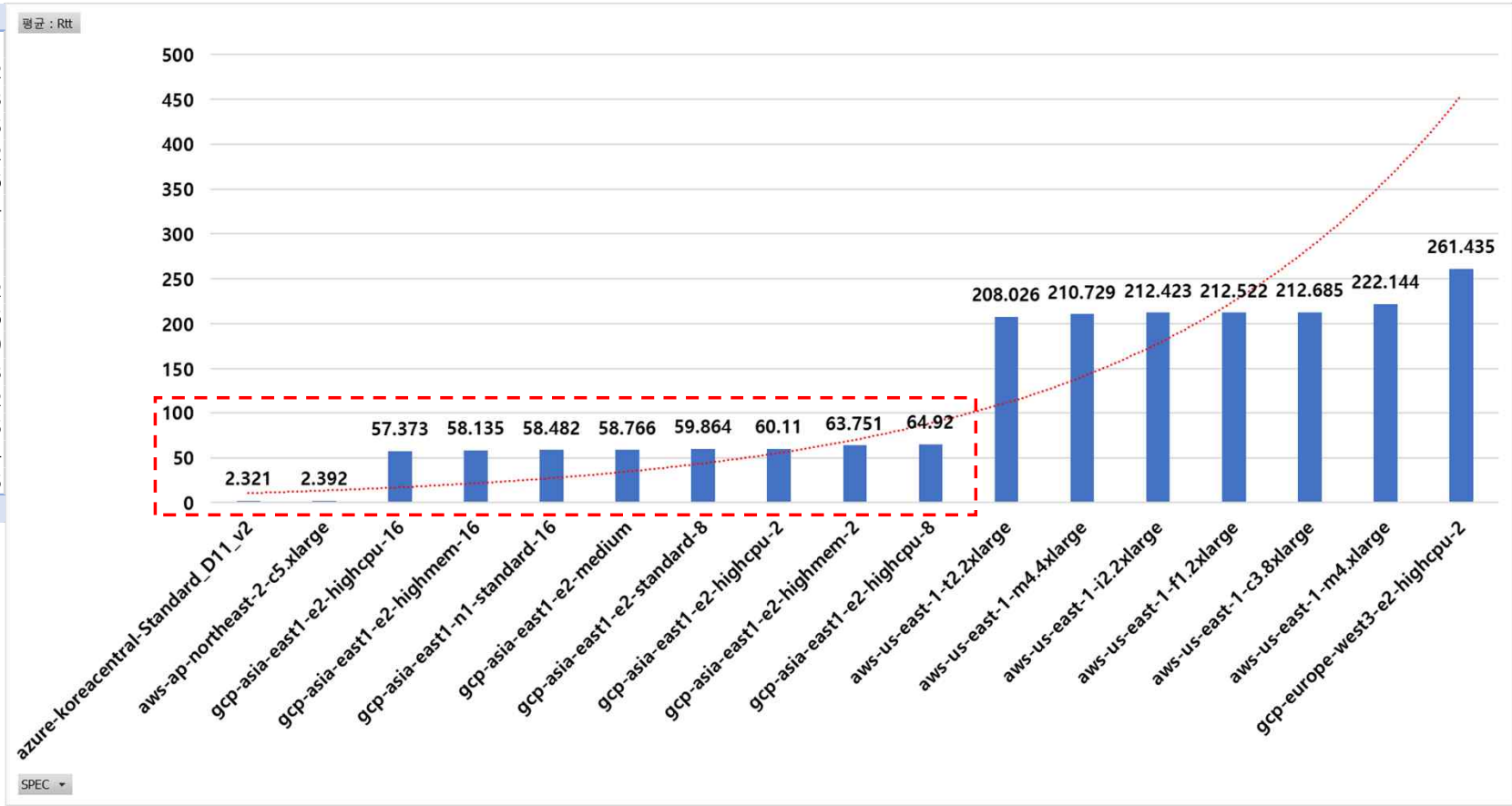




X축을 CPU의 결과를 기준으로 나열하였으나, 높은 CPU 성능이 다른 지표에 대한 높은 성능을 보장하지 않음
(사용자 상황에 맞는 사양 선택 필요 !!)

동적 성능 평가 결과 (예: 한국 지역 기준 응답 속도)

행 레이블	평균 : Rtt
azure-koreacentral-Standard_D11_v2	2.321
aws-ap-northeast-2-c5.xlarge	2.392
gcp-asia-east1-e2-highcpu-16	57.373
gcp-asia-east1-e2-highmem-16	58.135
gcp-asia-east1-n1-standard-16	58.482
gcp-asia-east1-e2-medium	58.766
gcp-asia-east1-e2-standard-8	59.864
gcp-asia-east1-e2-highcpu-2	60.11
gcp-asia-east1-e2-highmem-2	63.751
gcp-asia-east1-e2-highcpu-8	64.92
aws-us-east-1-t2.2xlarge	208.026
aws-us-east-1-m4.4xlarge	210.729
aws-us-east-1-i2.2xlarge	212.423
aws-us-east-1-f1.2xlarge	212.522
aws-us-east-1-c3.8xlarge	212.685
aws-us-east-1-m4.xlarge	222.144
gcp-europe-west3-e2-highcpu-2	261.435
총합계	119.18

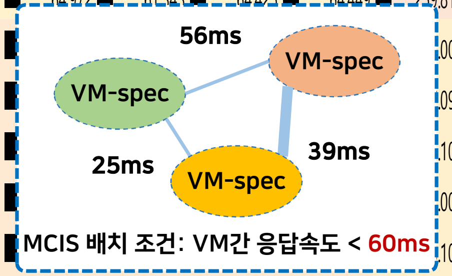


- 특정 IP를 기준으로 응답속도 평가
  - 위도경도를 통한 방식 보다 더 의미 있음

End User 까지의 응답 속도가 중요한 비즈니스의 경우, 응답 속도 평가에 대해 민감하게 MCIS를 구성할 필요

동적 성능 평가 결과 (예: 사양간 응답속도 평가)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
• aws-ap-northeast-2-c5.xlarge	1	0.104	185.13	189.917	185.629	189.697	185.138	186.786	3.14	64.66	61.158	61.367	58.218	64.972	61.343	64.425	64.449	259.812
• aws-us-east-1-c3.8xlarge	2	185.12	0.213	0.221	0.267	0.256	0.243	0.333	191.582	183.796	186.86	187.189	183.568					008
• aws-us-east-1-f1.2xlarge	3	189.926	0.238	0.209	0.203	0.254	0.215	0.335	184.887	186.751	183.789	184.033	187.098					097
• aws-us-east-1-i2.2xlarge	4	185.637	0.222	0.247	0.227	0.258	0.232	0.343	191.917	186.885	183.981	184.078	184.35					107
• aws-us-east-1-m4.4xlarge	5	189.703	0.219	0.221	0.226	0.206	0.212	0.287	190.96	184.077	186.969	183.894	184.27					004
• aws-us-east-1-m4.xlarge	6	185.124	0.23	0.239	0.235	0.257	0.245	0.327	189.72	184.381	184.618	184.463	184.579					104
• aws-us-east-1-t2.2xlarge	7	186.79	0.306	0.361	0.294	0.33	0.356	0.335	185.968	187.126	187.027	184.084	184.217	184.138	184.068	183.789	187.214	87.466
• azure-koreacentral-Standard_D11_v2	8	3.253	191.6	186.853	192.01	191.13	189.836	186.048	2.415	59.129	56.425	58.868	56.965	56.75	58.698	57.581	57.365	243.165
• gcp-asia-east1-e2-highcpu-16	9	61.405	184.408	184.576	187.309	184.403	187.47	184.745	59.274	0.351	0.345	0.699	1.17	1.395	1.303	1.199	0.755	258.283
• gcp-asia-east1-e2-highcpu-2	10	61.197	187.613	187.458	187.467	187.375	187.767	187.699	56.704	0.899	0.302	0.393	1.318	1.336	1.333	1.237	0.464	261.49
• gcp-asia-east1-e2-highcpu-8	11	61.718	184.631	184.604	184.489	184.217	184.625	184.643	59.12	1.117	1.229	0.308	1.162	1.956	1.301	1.42	0.478	261.661
• gcp-asia-east1-e2-highmem-16	12	58.22	184.376	187.759	185.001	184.654	184.879	185.041	56.823	1.313	1.374	1.02	0.447	0.499	1.333	1.282	0.767	264.039
• gcp-asia-east1-e2-highmem-2	13	64.964	185.085	188.072	184.866	187.517	187.616	184.972	55.881	1.743	1.783	1.394	1.786	0.463	2.105	1.638	1.17	262.587
• gcp-asia-east1-e2-medium	14	64.649	184.53	184.424	184.59	184.125	184.759	187.767	58.212	1.417	1.419	1.211	1.57	1.803	0.425	0.967	0.865	259.108
• gcp-asia-east1-e2-standard-8	15	61.401	187.481	187.52	187.576	186.977	187.407	184.266	57.786	0.917	0.909	0.413	0.897	1.183	0.926	0.35	1.001	258.576
• gcp-asia-east1-n1-standard-16	16	64.84	184.984	184.363	187.654	184.483	185.014	185.152	57.671	1.341	1.185	0.478	1.403	1.584	1.204	1.471	0.37	255.574
• gcp-europe-west3-e2-highcpu-2	17	256.501	87.885	87.73	88.161	87.054	88.11	90.092	243.362	259.022	262.704	262.021	264.844	262.443	259.204	258.971	255.515	1.09

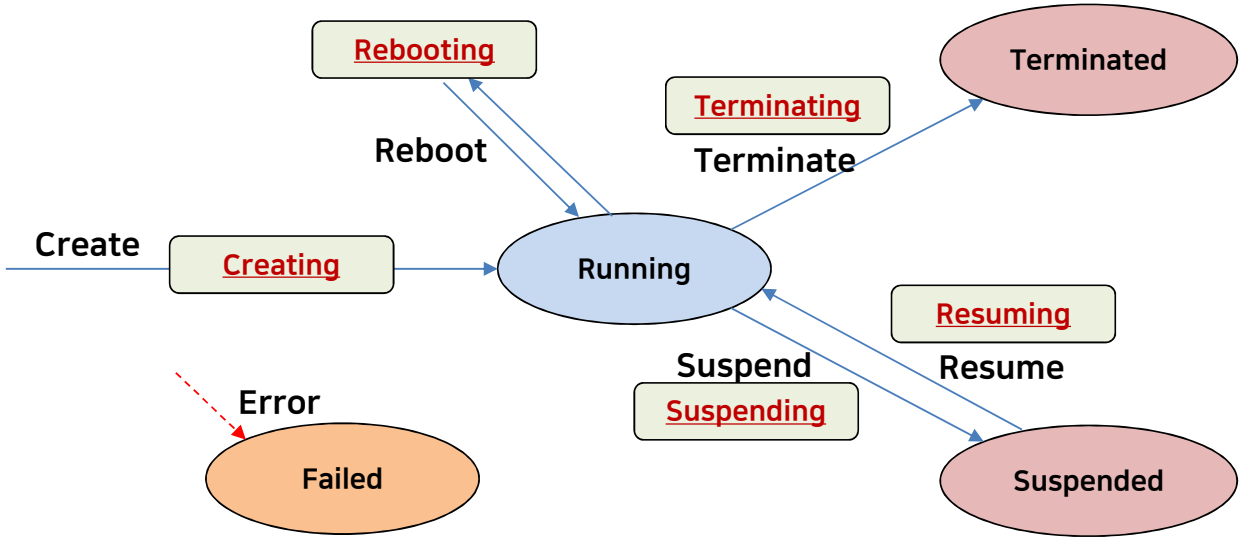


MCIS에서 동작하는 애플리케이션에 의해서 VM간의 통신이 빈번하고 중요한 경우, VM간 응답시간을 낮게 지정하여 MCIS 구성 요청

MCIS 라이프사이클 상태 관리 기능 개선 릴리스

릴리스
핵심포인트 [2]

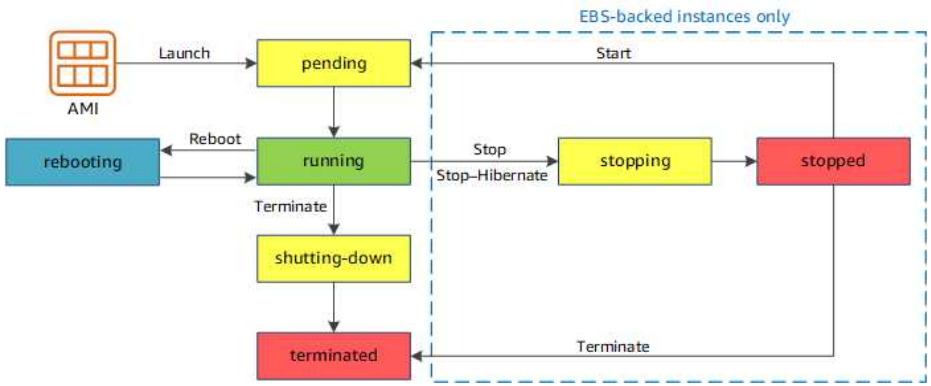
- 멀티클라우드 라이프사이클 이슈
  - CSP 마다, 라이프사이클 상태 처리 방법과 표기가 다름
    - 정확한 상태 파악 어려움
  - MCIS 는 여러 가상 서버를 포함
    - 종합적 상태 명시 방법 고안 필요
- MCIS 라이프사이클 상태 관리 기능 개선 릴리스
  - 표준화 및 오류 보정
    - MCIS의 라이프사이클을 표준화
    - 현재 액션을 인지(DB에 저장)하여, 그 정보를 기준으로 오류를 보정
  - 통합 상태 표시
    - Partial 개념 도입



MCIS 라이프사이클 상태도

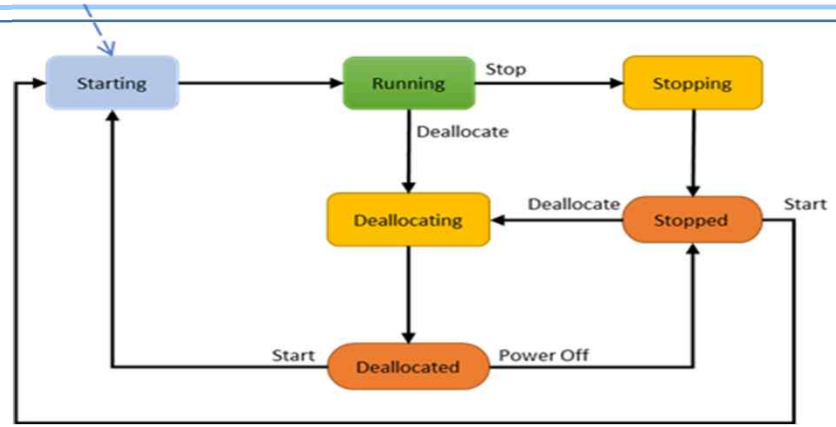
MCIS 라이프사이클 상태 통합 명시 방식 고안 및 안정성 향상

클라우드 별 서로 다른 라이프사이클 처리 방식



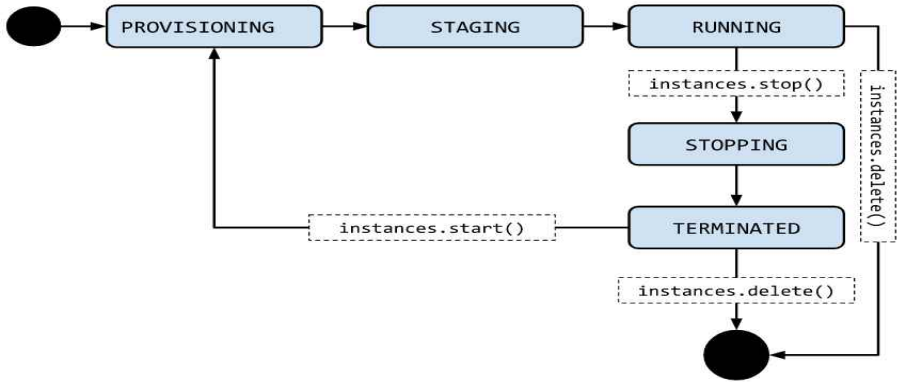
docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html

AWS



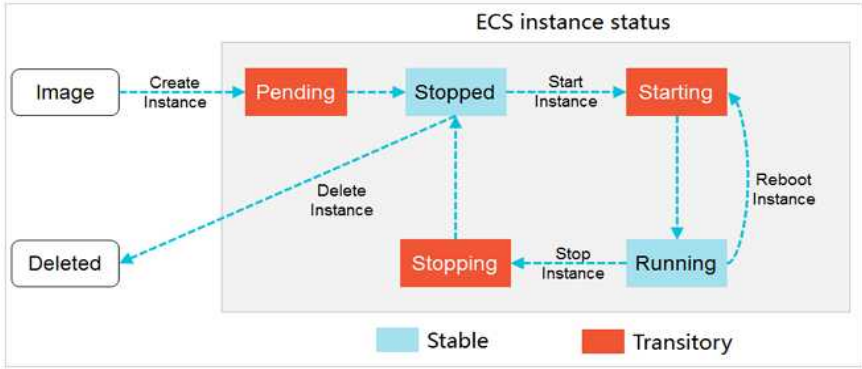
docs.microsoft.com/en-us/azure/virtual-machines/linux/states-lifecycle

Azure



cloud.google.com/compute/docs/instances/instance-life-cycle

GCP



alibabacloud.com/help/doc-detail/25380.htm

AlibabaCloud

클라우드 별 서로 다른 라이프사이클 상태 정보 (현실..)

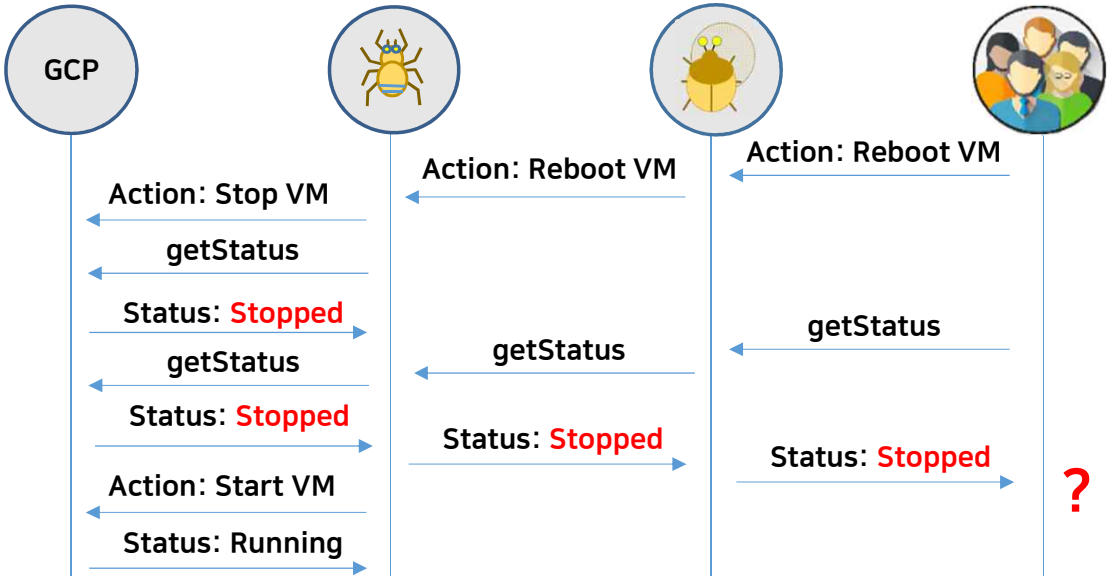
오류발생시점

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) - 실제: suspending	Deleted	상태 정보를 얻을 수 없을 때 오류에 의한 비정상 상태
AWS	맵핑 불필요 (현재: StartVM 내부 Running 상태 확인 후 P-IP 부착 후 return)	running	stopping	stopped	Resuming (자체생성상태)	Rebooting	shutting-down	terminated	상태 정보를 얻을 수 없을 때 오류에 의한 비정상 상태
GCP	Provisioning staging	running	stopping	terminated	Resuming (자체생성상태)	Rebooting (자체생성 (OP: Stop&Start) 실제: suspending~, running	Terminating (자체생성상태) (OP: Delete)	(예외) NotExist	상태 정보를 얻을 수 없을 때 오류에 의한 비정상 상태
Azure	starting /-	runnning /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

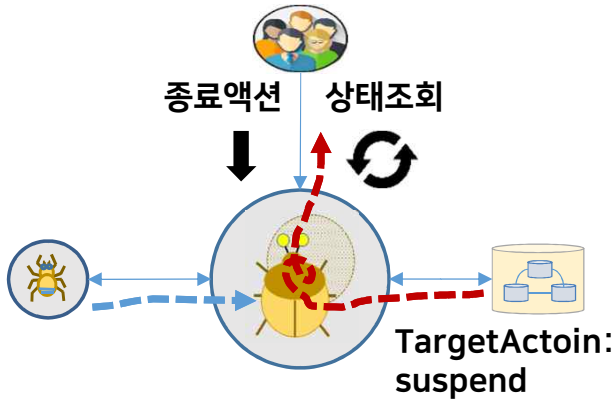
MCIS 라이프사이클 처리 방식 개선

MCIS 라이프사이클 처리 방식 개선을 통한 안정화

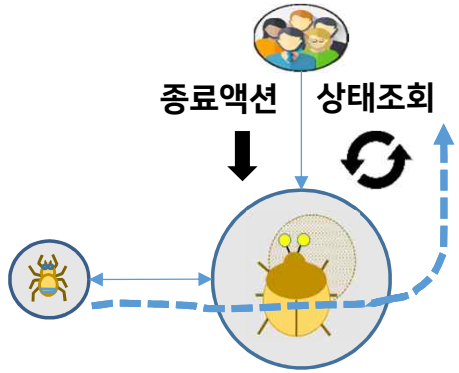
- MCIS 라이프사이클 처리 방식 개선 (Stateless CB-TB로..)
  - 목적 액션, 상태 변경 히스토리 추적
    - 목적 상태와 현재 상태에 대한 validation 가능
    - CB-TB 시스템 중단 및 재실행시 현황 파악 가능



오류 상황 예시



개선 : 정상 출력 (오류 보정)



기존 방식 : 오류 발생

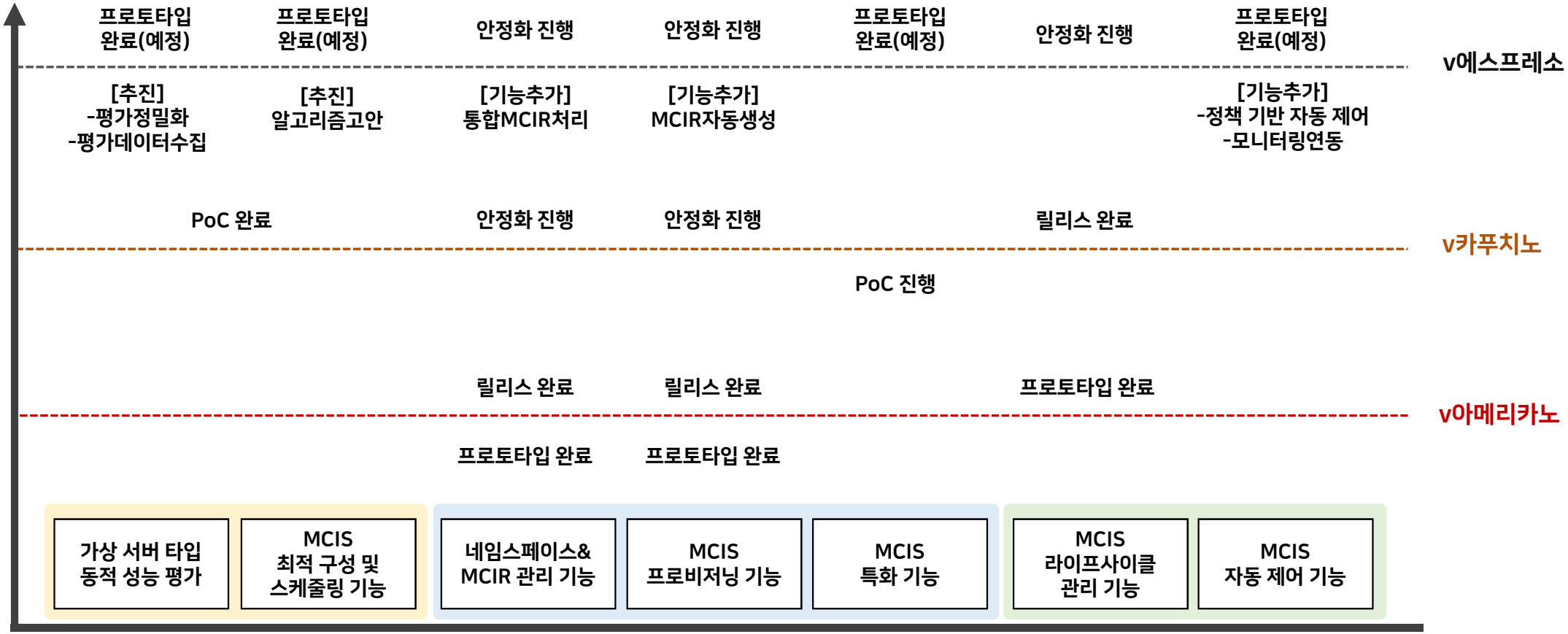
MCIS 라이프사이클 처리 방식 개선 결과

Action: Create MCIS

Action: Create Suspend

CB-Tumblebug 개발 현황 및 로드맵

PoC → 프로토타입 → 릴리스 → 안정화&고도화



MCIS 라이프사이클(릴리스), MCIS 최적 배치 및 멀티클라우드인프라 성능 벤치마킹(PoC)

CB-Tumblebug, 20년도 개발 계획

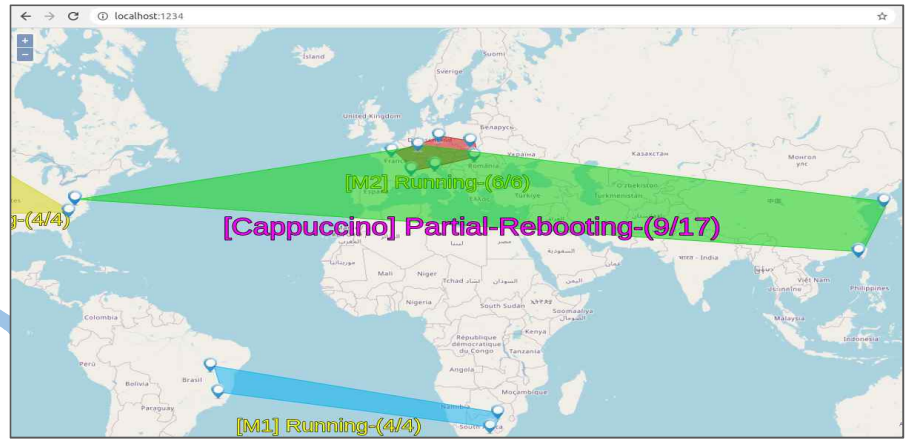
PoC → 프로토타입 → 릴리스 → 안정화&고도화

주요 업무	상세 업무	수행 내용	결과물 공개 수준	대상 버전
CB-Tumblebug 시스템 개선 및 안정화	CB-Tumblebug API 현행화 및 프레임워크 통합	- API 항목 및 파라미터 개선(4월초) - 리모델링 CB-Spider (API 및 정보처리 방식) 통합	대상버전 릴리스	Cappuccino
	CB-Tumblebug 시험 체계 개선	- CB-Tumblebug 시험 체계 구축 - CB-Tumblebug 시험 데이터 생성	대상버전 릴리스	Cappuccino
	멀티 클라우드 인프라 서비스 라이프사이클 개선	- 클라우드별 라이프사이클 제어 상태 처리 방식 분석 - 라이프사이클 제어 트렌젝션 생성 및 처리 기능 개발 - 트렌젝션 단위 라이프사이클 상태 저장 및 조회 기능 개발	대상버전 릴리스	Cappuccino
멀티 클라우드 인프라 서비스 최적 배치 기능 개발	멀티 클라우드 인프라 서비스 배치 메커니즘 개발	- 최적 배치 조건 선정 [3월 말] - 최적 배치 요구사항 템플릿 및 API 체계 개발 [3월 말] - 멀티 클라우드 인프라 서비스 평가 정보 테이블 자동 생성 기능 개발 [4월 말] - 최적 배치 알고리즘 기반 우선순위 리스트 처리 기능 개발 [6월 말]	PoC (일부) 대상버전 릴리스	Cappuccino Espresso
	멀티 클라우드 인프라 서비스 동적 성능 평가 메커니즘 개발	- 멀티 클라우드 인프라 서비스 동적 성능 평가 항목 선정 [4월 중] - 멀티 클라우드 인프라 서비스 동적 성능 평가 수집 스케줄러 개발 [6월 말] - 성능 평가 항목별 측정 기능 개발(ex: 계산 성능, DB처리 성능 등) [5월 ~ 6월 말] - 멀티 클라우드 인프라 서비스 평가 수행 에이전트 연동 [7월 말] - 구동 시험 및 데이터 수집 [8월 중]	PoC (일부) 대상버전 릴리스	Cappuccino Espresso
	멀티 클라우드 인프라 서비스 고속·동적 배치 기술 연구	- 멀티 클라우드 인프라 서비스 고속·동적 배치 기술 분석 및 고안 - 멀티 클라우드 인프라 서비스 고속·동적 배치 메커니즘 PoC 추진	PoC	Espresso
멀티 클라우드 인프라 서비스 통합 품질 제어 자동화 개발	멀티 클라우드 인프라 서비스 통합 품질 제어 자동화 정책 관리 기능 개발	- 멀티 클라우드 인프라 서비스 통합 품질 제어 자동화 정책(조건 및 액션) 요청 템플릿 정의 및 API 체계 개발 - 멀티 클라우드 인프라 서비스 통합 품질 제어 자동화 정책 처리 상태 저장 및 조회 기능 개발	대상버전 릴리스	Espresso
	멀티 클라우드 인프라 서비스 통합 품질 분석 기능 개발	- 멀티 클라우드 인프라 서비스 통합 품질 모니터링 항목 정의 - 멀티 클라우드 인프라 서비스 통합 품질 데이터 수집 모듈 개발 (CB-Dragonfly 연동) - 멀티 클라우드 인프라 서비스 통합 품질 조건 판별 및 트리거 모듈 개발	대상버전 릴리스	Espresso
	멀티 클라우드 인프라 서비스 통합 품질 제어 액션 개발	- 멀티 클라우드 인프라 서비스 통합 품질 제어 액션 선정 - 멀티 클라우드 인프라 서비스 통합 품질 제어 액션 개발	대상버전 릴리스	Espresso
클라우드간 동일 서브넷 기술 개발	동일 서브넷 제공 기술 및 UseCase 분석	- VPN 기술 등 동일 서브넷 구성 위한 관련 기술 분석 (Wireguad, Envoy 기술 등) - 동일 서브넷 구성 서비스 기반 use case 후보 발굴 및 PoC 테스트	PoC, 대상버전릴리스	Cappuccino Espresso
	동일 서브넷 기능 개발 및 시험	- 동일 서브넷 생성, 조회, 삭제 기능 개발 - REST 런타임 서버 개발 및 API 제공	프로토타입	Cappuccino
		- 동일 서브넷 서비스 상태 관리 기능 개발 - REST 런타임 서버 개발 및 API 제공	프로토타입	Cappuccino
클라우드간 서비스 로드밸런싱 기술 개발	로드밸런싱 제공 기술 및 UseCase 분석	- 로드밸런싱 환경 구성 위한 기술 분석 - 클라우드간 로드밸런싱 기반 use case 후보 발굴 - 로드밸런싱 관련 기존 기술 분석 및 PoC 테스트	PoC	Espresso
	로드밸런싱 기능 개발 및 시험	- 로드밸런싱 서비스 생성, 조회 및 상태 관리 기능 개발 - REST 런타임 서버 개발 및 API 제공 - 클라우드간 로드밸런싱 서비스 기반 use case 선정 및 개발	대상버전 릴리스	Espresso

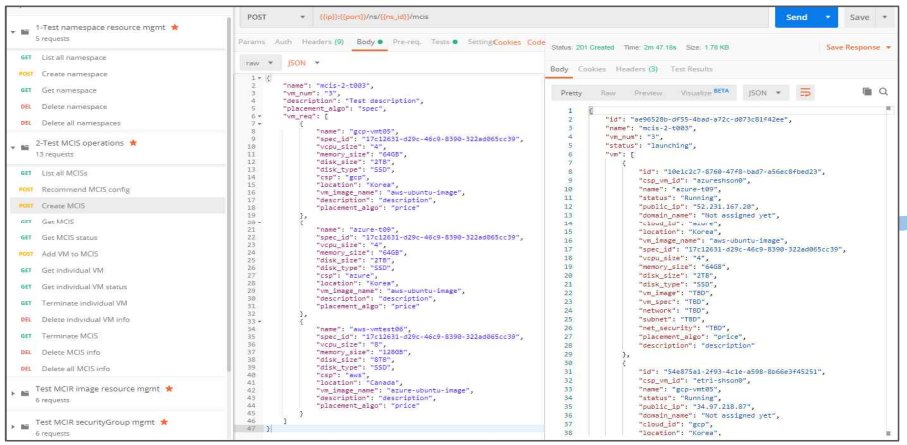
CB-Tumblebug 기능 사용 형태 (API 기반 제어)

- REST 기반으로 명령이 쉽고 간결
- 언어.플랫폼에 독립적인 연동 가능

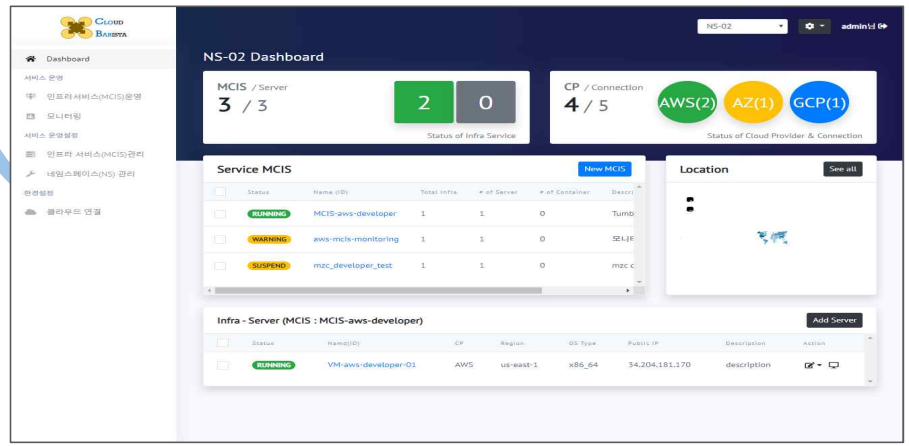
CB-Tumblebug 서버



지도 기반 GUI 클라이언트



REST API 사용을 위한 Postman 클라이언트



Web 기반 GUI 클라이언트 (CB-Waterstrider/WebTool)

멀티 클라우드 서비스 공통 프레임워크

CB-Tumblebug 기술 시연

카푸치노(Cappuccino) 한잔 어떠세요 ? ^^

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

CB-Tumblebug 기술 시연 개요

- 글로벌 지역을 커버하는 MCIS 생성
  - 컴퓨팅 머신 18개 통합 생성
  - 4개의 CSP (AWS, Azure, GCP, Alibaba) : 10개의 지역으로 구성

- MCIS를 기반으로 글로벌 웹서버 분산 실행
  - 여러 지역에 걸친 18개 서버에 Nginx 웹서버를 원클릭으로 배치 및 실행
  - 각 웹서버에 접속하여, 실제 호스트의 위치를 조회

- MCIS 라이프사이클 제어 및 상태 확인
  - 컴퓨팅 머신 18개 통합 관리
  - Suspend, Resume, Terminate MCIS

- 멀티 클라우드 Spec 동적 성능 벤치마킹 수행 및 결과 확인
  - 벤치마킹 에이전트(CB-Milkyway) 자동 배치
  - CPU 계산속도 평가 (싱글 코어/멀티 코어)
  - Memory 처리속도 평가 (읽기/쓰기)
  - FileIO 처리속도 평가 (읽기/쓰기)
  - DB 처리속도 평가 (읽기/쓰기)
  - 목적지까지 네트워크 지연 시간 평가
  - 상호 네트워크 지연 시간 평가

감사합니다.

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

(손 석 호 / contact-to-cloud-barista@googlegroups.com)

“Contact to the Multi-Cloud”

클라우드 바리스타들의 두 번째 이야기

Cloud-Barista Community 2nd Open Conference