



CLOUD
BARISTA

멀티클라우드, 컴퓨팅 인프라에 제약없는 서비스 생태계

클라우드바리스타 커뮤니티 제7차 컨퍼런스

[CB-Tumblebug]

멀티클라우드 인프라 서비스 통합 관리

손석호

CB-Tumblebug 프레임워크 리더

코르타도(Cortado) 한잔 어떠세요 ?

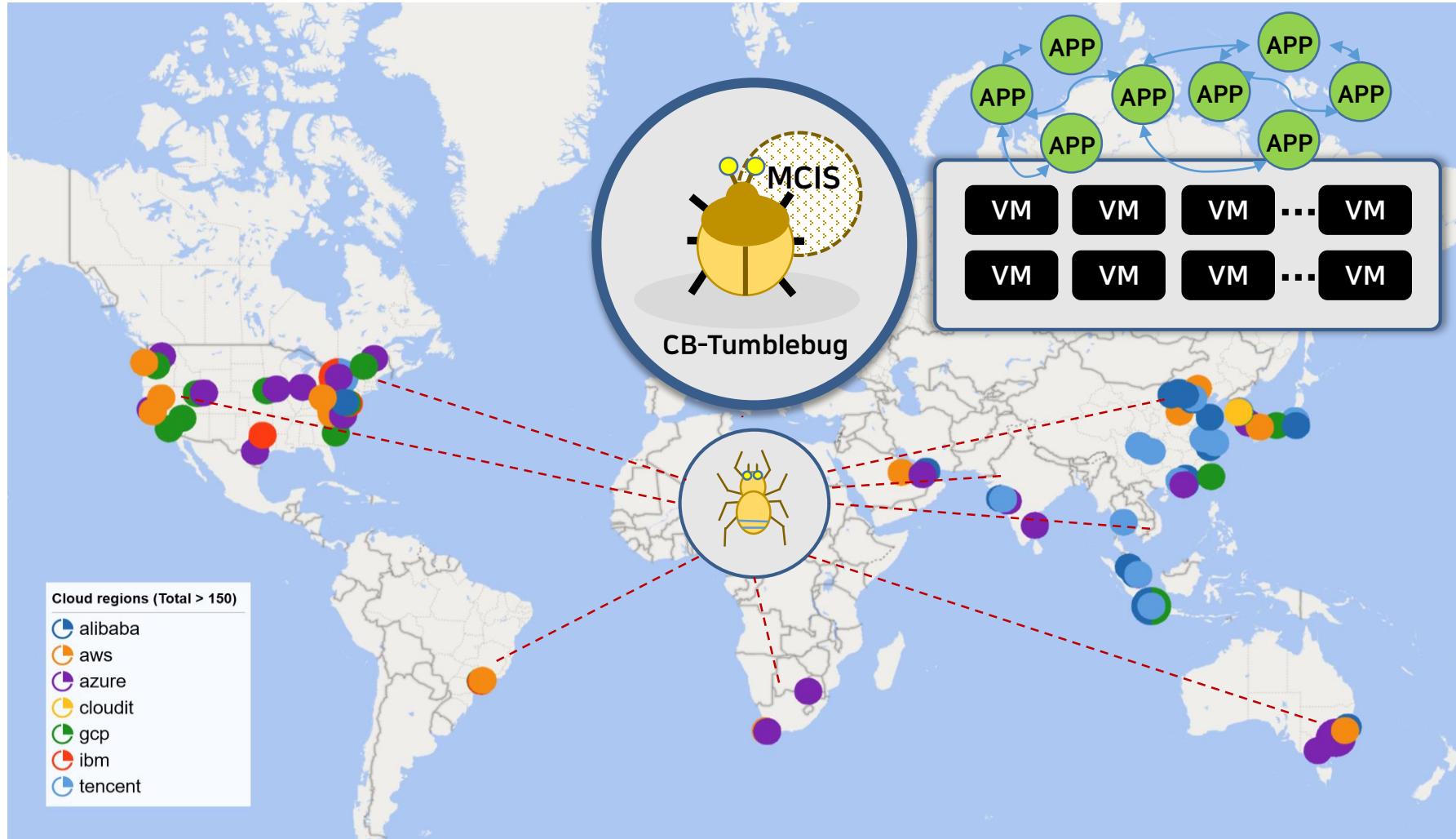
목 차

I CB-Tumblebug 개요

II CB-Tumblebug 기능 및 구조 요약

III CB-Tumblebug 릴리스 주요 개선 사항

CB-Tumblebug: 멀티클라우드 인프라 서비스 통합 관리 필요성

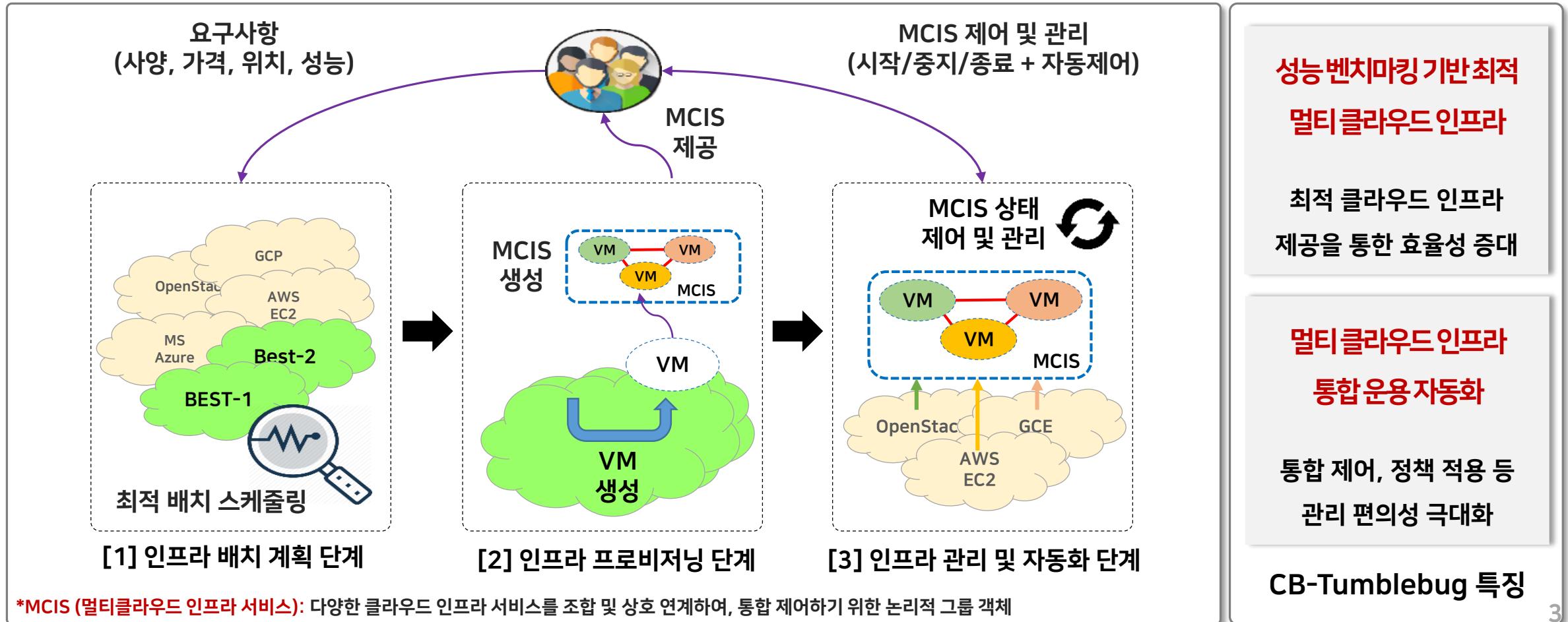


복잡한 멀티클라우드의 효과적인 활용을 위해, 자원을 유기적으로 통합 제공하는 멀티클라우드 인프라 서비스 통합 관리 필요

CB-Tumblebug: 멀티클라우드 인프라 서비스 통합 관리

**기술
정의**

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

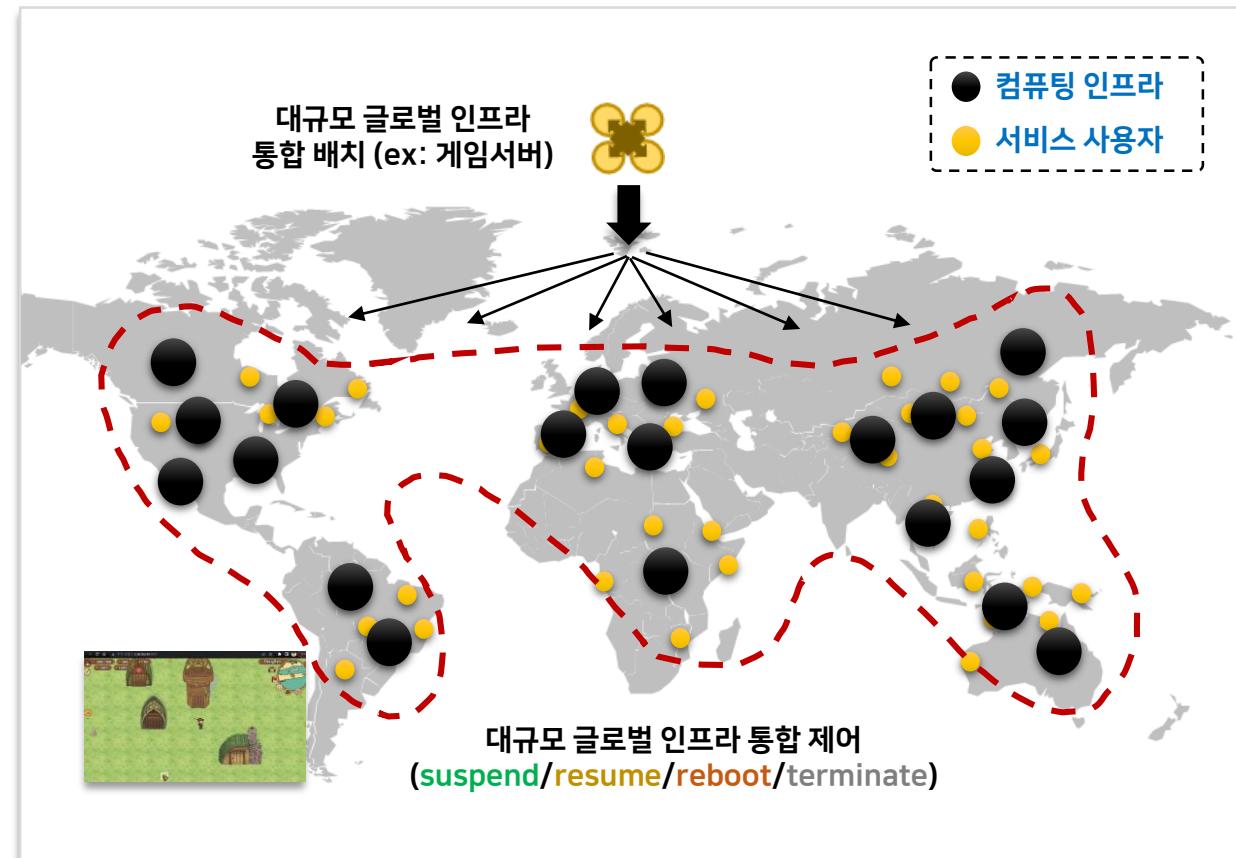


CB-Tumblebug 활용 사례

<최적배치 기반의 멀티클라우드 기반 영상 회의 서비스>



<대규모 글로벌 스케일 멀티클라우드 인프라 서비스>





CB-Tumblebug 활용 사례 (글로벌 3D FPS 게임 서비스)

localhost:1324

[CB-MapUI for Cloud-Barista Tumblebug]

System config
TB IP: localhost TB Port: 1323 Username: default Password: Total CSP regions: Display interval: 10

User setting
NS ID: ns01 MCIS ID: mcis-australia Status Overview resources Hide/Show Register resources

MCIS control
 Refine Terminate Delete Release default res

MCIS provisioning
 Provisioning Policy Location-based OS Image Type Ubuntu 18.04 CPU 1 ~ 8 RAM 0.5 ~ 8

Select Application FPSGame(Xonotic) Deploy application Start App Stop App Status App

[Complete] Number of Registered Cloud Regions: 150
[0] ALIBABA(ap-northeast-1) Location: 139.9917|34.6895 (Japan (Tokyo))
[1] ALIBABA(ap-northeast-2) Location: 126.7800|37.3600 (South Korea (Seoul))
[2] ALIBABA(ap-south-1) Location: 72.1774|19.1761 (India (Mumbai))

Display recommendation table:

[mcis-north-america] Running:20 (R:20/20)
[mcis-europe] Running:18 (R:18/18)
[mcis-northeast-asia] Running:14 (R:14/14)
[monitoring]

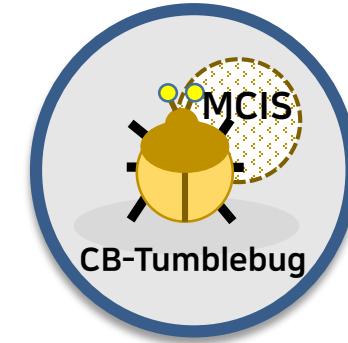
글로벌 MCIS 생성 (VM: 23)

CB-Tumblebug 주요 기능 요약

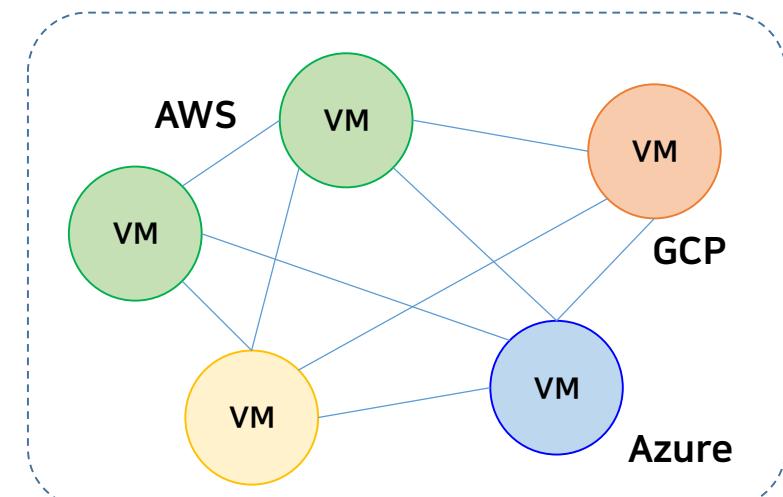
멀티클라우드 인프라를 ..

최적으로 [구성]→[생성]→[설정]→[관리]→[자동제어]

- MCIS 최적 배치 스케줄링
 - 클라우드 자원의 평가 및 정보 수집을 통해 최적 MCIS를 구성하는 기능
- MCIS 프로비저닝 및 특화
 - 다양한 클라우드 자원을 활용하여 MCIS를 생성하고 *특화하는 기능
- MCIS 라이프사이클 제어 및 관리
 - MCIS 라이프사이클 상태를 종합적으로 관리, 통합 제어하는 기능
- MCIS 정책 기반 자동 제어
 - MCIS를 진단하고 결과에 따라 자동 제어하는 운영 자동화 기능



CB-TB v0.6.16 API: 126개



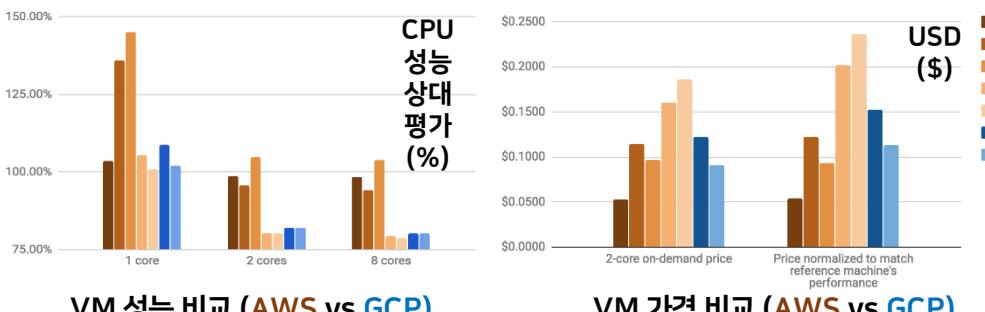
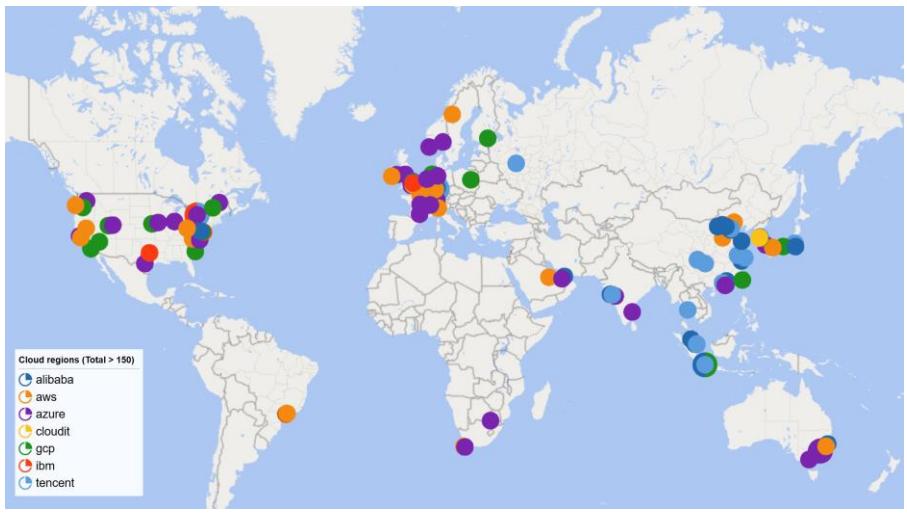
멀티 클라우드 인프라 서비스 (MCIS)

*MCIS 특화: MCIS에 다양한 설정 및 특성을 가미할 수 있는 구조적 특징 (Ex: VPN 설정, LB 설정 등)

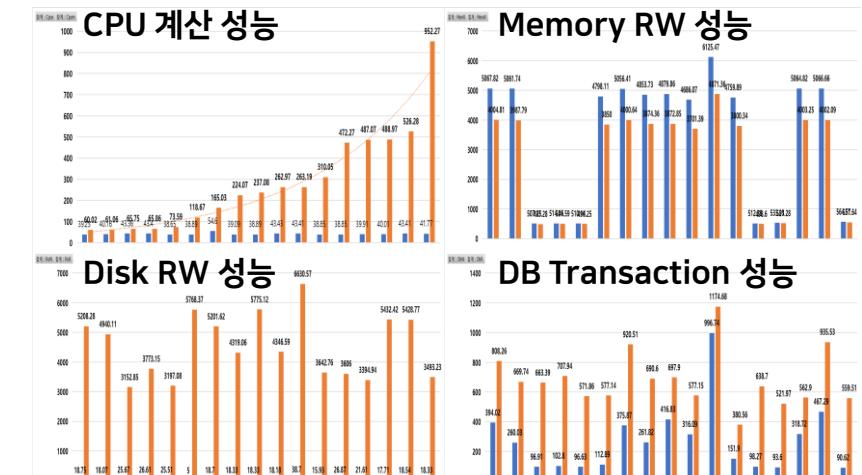
MCIS 최적 배치 스케줄링

[구성]→[생성]→[설정]→[관리]→[자동제어]

- 다양한 정보를 수집하여 사용자 요구사항에 적합한 자원을 선정하여, MCIS를 최적으로 구성(배치/스케줄링)하는 기능



- * **MCIS 최적 배치 방법 (Filtering & Prioritizing)**
 - 스펙 기반 선정
 - 가격 기반 선정
 - 위치 기반 선정
 - 성능 기반 선정 (benchmark)
 - CSP 기반 선정
 - 네트워크 응답속도 기반 선정
 - 복합 조건 기반 선정



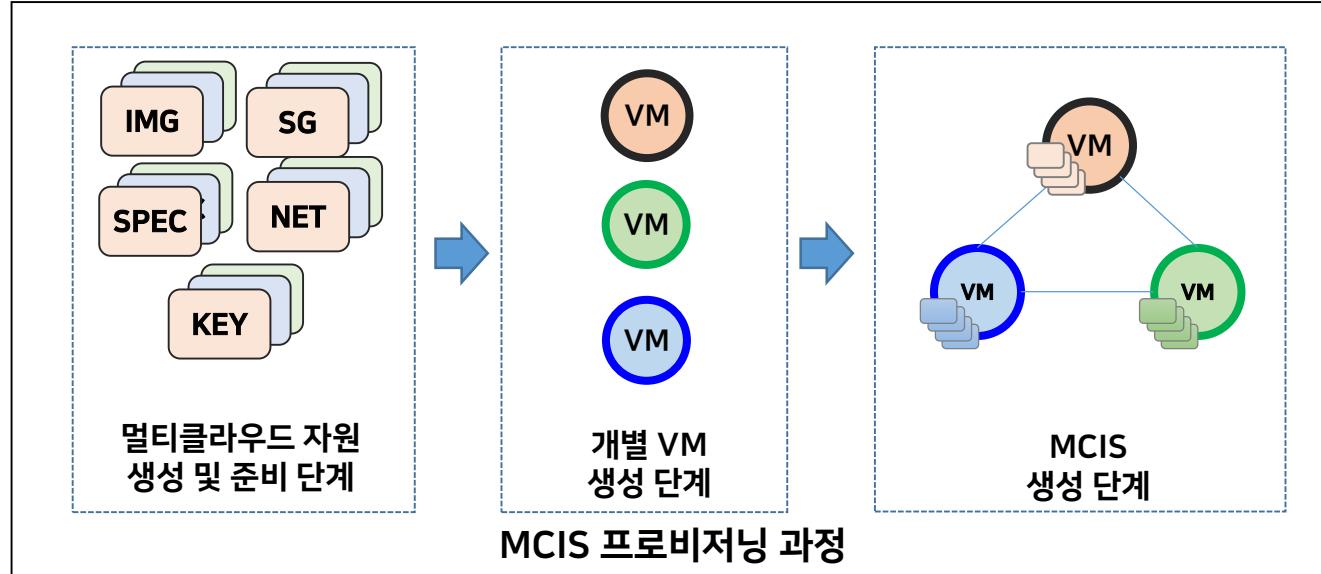
* MCIS 최적 배치: 최적으로 구성(배치/스케줄링)된 MCIS의 프로비저닝을 의미

MCIS 프로비저닝 및 특화 (1/2)

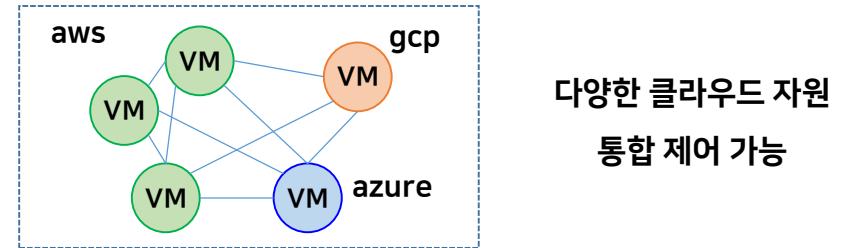
[구성]→[생성]→[설정]→[관리]→[자동제어]

- 이종 클라우드의 컴퓨팅 자원을 통합 생성 및 상호 연계하여 *MCIS로 구성하고 설정을 통해 특화하는 기능

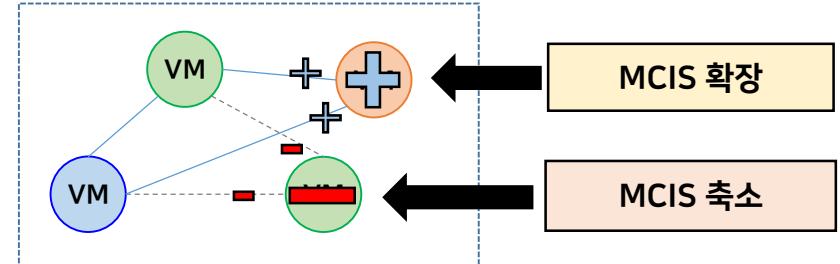
*MCIS: 다양한 클라우드 인프라 서비스를 조합 및 상호 연계하여, 통합 제어하기 위한 논리적 객체



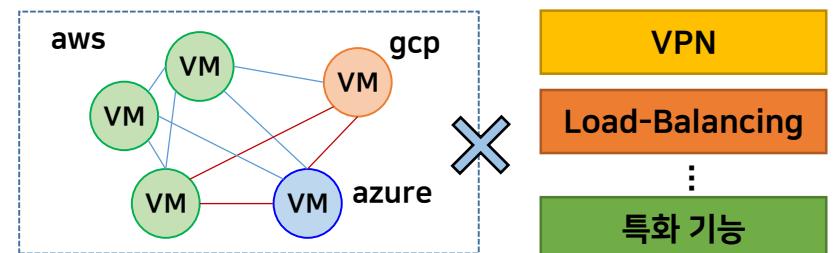
[MCIS 구성 예시]



[MCIS 구성 변경 예시]



[MCIS 특화 구성 예시]



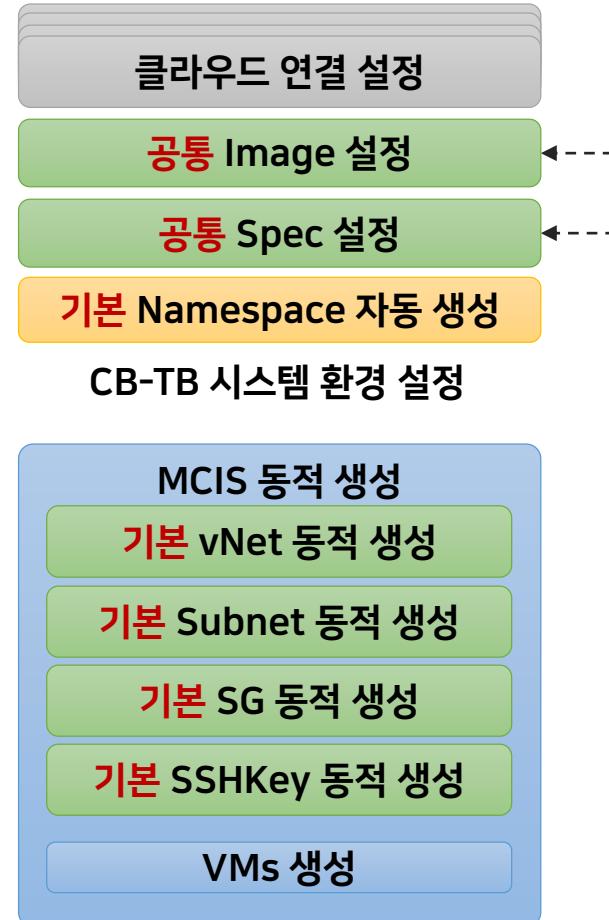
MCIS 프로비저닝 및 특화 (2/2)

[구성]→[생성]→[설정]→[관리]→[자동제어]

12개 API 호출



1개 API 호출



[시스템 환경 설정 스크립트]
`./scripts/initMultiCloudEnv.sh`

[공통 이미지] <https://github.com/cloud-barista/cb-tumblebug/blob/main/assets/cloudimage.csv>

Provider Name	Connection Name	Csp Image Id	Os Type
AWS	aws-ap-east-1	ami-036915aa0cb1d91a1	Ubuntu 18.04
AWS	aws-me-south-1	ami-026dde872642a6ffe	Ubuntu 18.04
AWS	aws-af-south-1	ami-0e4b4778694305983	Ubuntu 18.04
AWS	aws-eu-south-1	ami-0f274cb646af3475	Ubuntu 18.04
AZURE	azure-westus	Canonical:UbuntuServer:18.04-LTS:1	Ubuntu 18.04
AZURE	azure-eastus	Canonical:UbuntuServer:18.04-LTS:1	Ubuntu 18.04
AZURE	azure-northeurope	Canonical:UbuntuServer:18.04-LTS:1	Ubuntu 18.04
AZURE	azure-westeurope	Canonical:UbuntuServer:18.04-LTS:1	Ubuntu 18.04
AZURE	azure-eastasia	Canonical:UbuntuServer:18.04-LTS:1	Ubuntu 18.04

[공통 스펙] <https://github.com/cloud-barista/cb-tumblebug/blob/main/assets/cloudspec.csv>

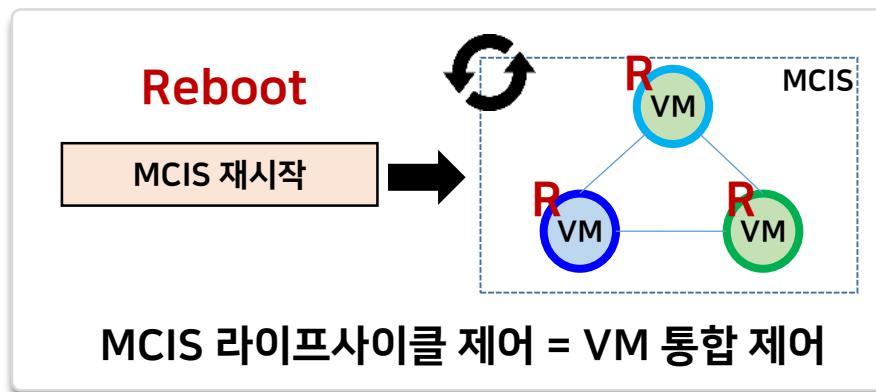
Provider Name	Region Name	Connection Name	Csp Spec Name	Cost Per Hour	Evaluation Score
azure	azure-westus	azure-westus	Standard_A1	0.04	34.29
azure	azure-westus	azure-westus	Standard_B1s	0.49	34.4
azure	azure-westus	azure-westus	Standard_A1	0.04	33.96
gcp	gcp-asia-east	gcp-asia-east	g1-small	0.03	33.85
gcp	gcp-asia-east	gcp-asia-east	e2-standard-2	0.08	54.43
gcp	gcp-asia-east	gcp-asia-east	g1-small	0.03	34.4

MCIR 생성 단계 간소화 및 자동화 ❤

MCIS 라이프사이클 제어 및 관리

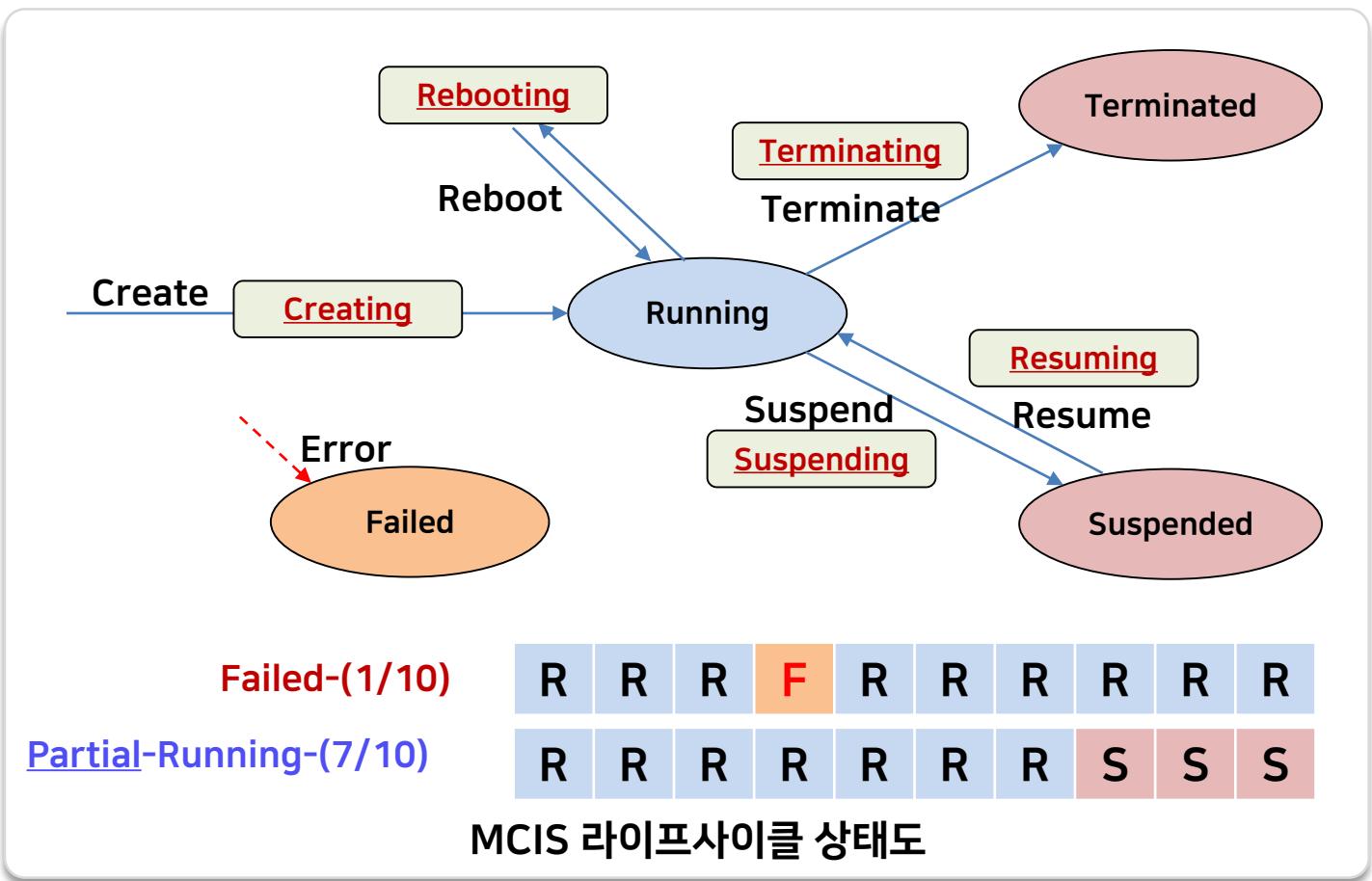
[구성]→[생성]→[설정]→[관리]→[자동제어]

- 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	Pending (AWS: Start/Stop 대상) or Pending (AWS: Create)	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)	Terminating (자체생성상태) (OP: Delete)	(예외)NotExist	상태 정보를 얻을 수 없을 때 오류에 의한 비정상 상태
Azure	starting /- running /succeeded	running /- stopping /- stopped /succeeded	stopping /- stopped /succeeded	terminated	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

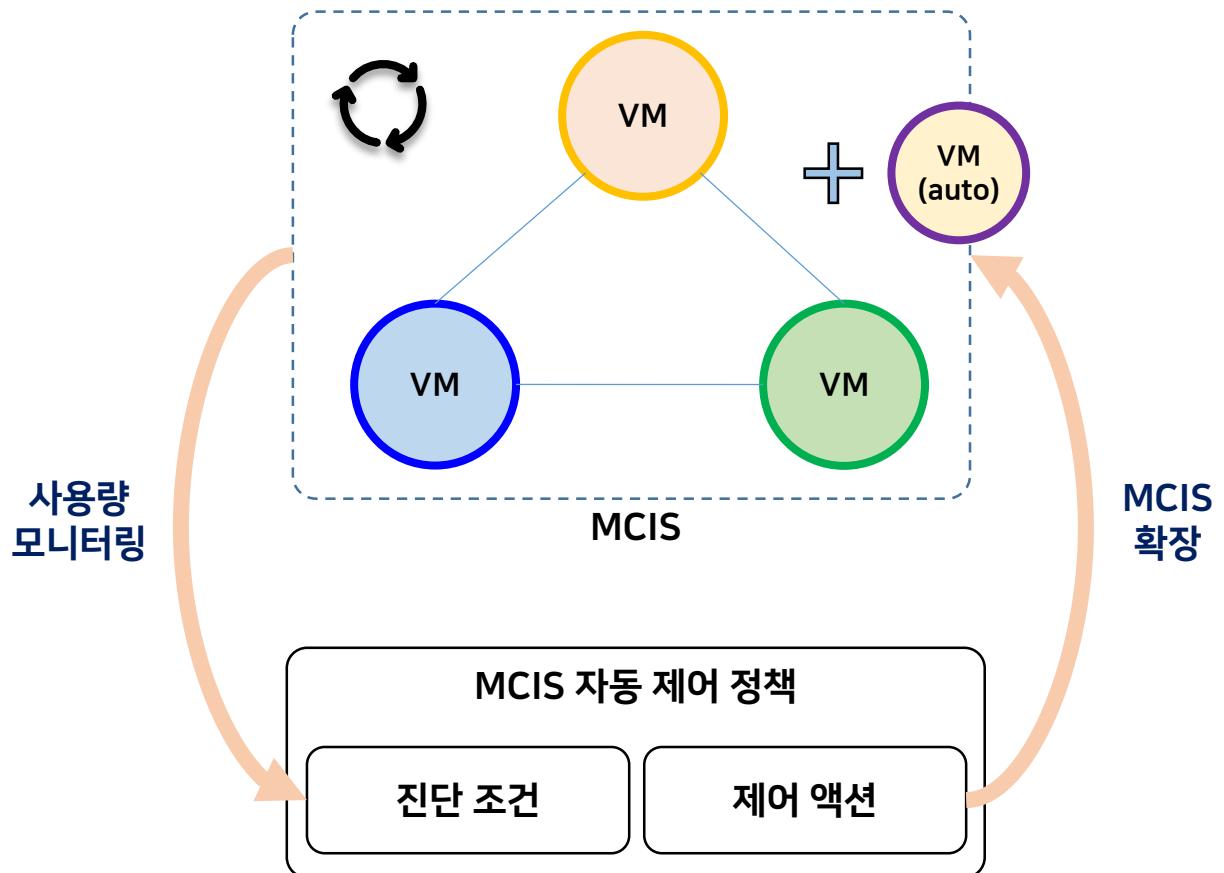
MCIS 라이프사이클 상태 검증 및 보정



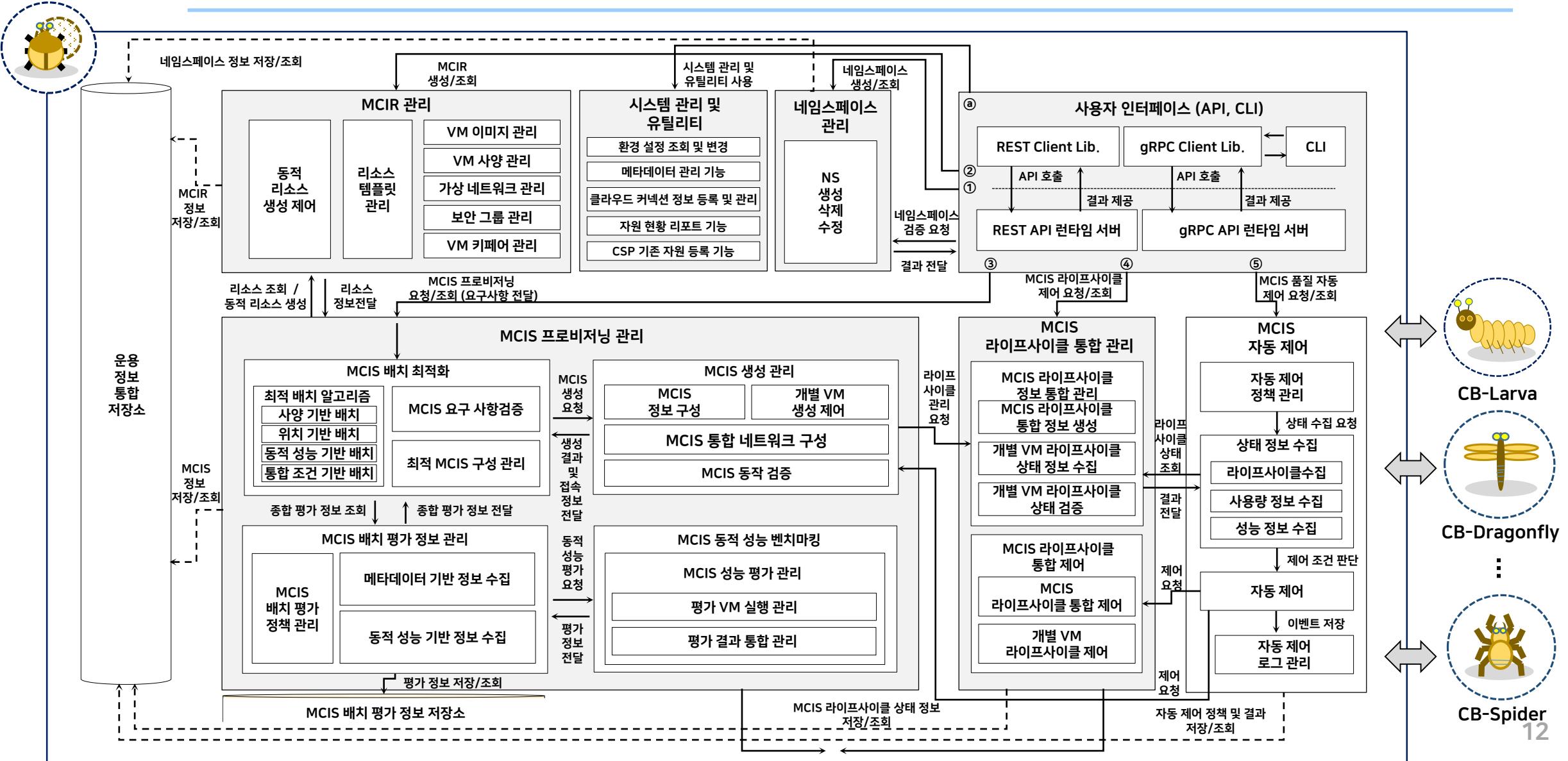
사용자 정책 기반 MCIS 자동 제어

[구성]→[생성]→[설정]→[관리]→[자동제어]

- 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)
 - ...



(참고) CB-Tumblebug 시스템 구조 [구성]→[생성]→[설정]→[관리]→[자동제어]

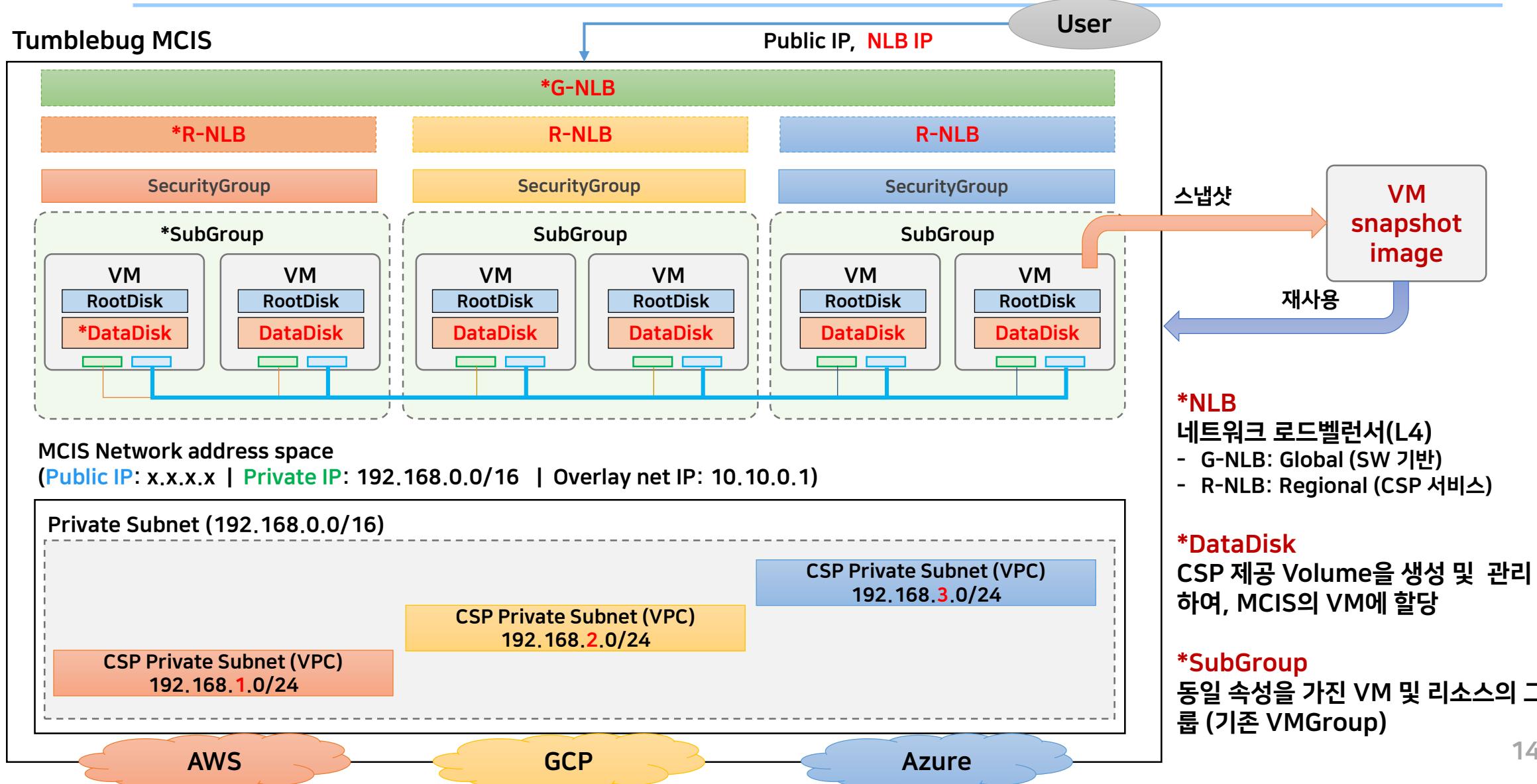




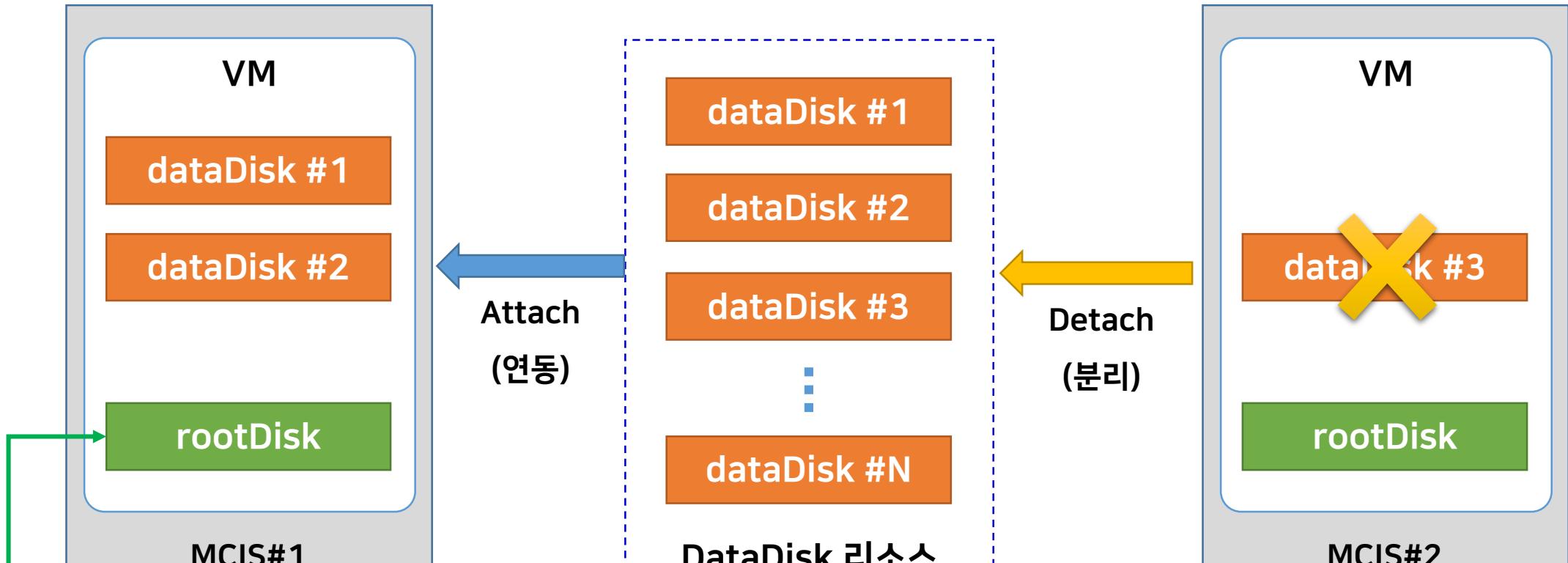
CB-TB 코르타도 (v0.7.0) 주요 개선 포인트

- 1) MCIS 인프라 형상 개선 (+DataDisk, +Snapshot, +NLB)
- 2) MCIS 자동 제어 안정화 및 개선 (+MC-AutoScale)

1) MCIS 인프라 형상 개선 (+DataDisk, +Snapshot, +NLB)



1-1) DataDisk (volume) 관리 기능



MCIS 생성시 기본 지정

- RootDiskType
- RootDiskSize

DataDisk 관리

(생성/수정/연동/삭제)

- DiskType
- DiskSize



1-1) DataDisk (volume) 관리 기능

[Infra resource] MCIR Data Disk management		
GET	/ns/{nsId}/mcis/{mcisId}/vm/{vmId}/dataDisk	Get available dataDisks for a VM
PUT	/ns/{nsId}/mcis/{mcisId}/vm/{vmId}/dataDisk	Attach/Detach available dataDisk
GET	/ns/{nsId}/resources/dataDisk	List all Data Disks or Data Disks' ID
POST	/ns/{nsId}/resources/dataDisk	Create Data Disk
DELETE	/ns/{nsId}/resources/dataDisk	Delete all Data Disks
GET	/ns/{nsId}/resources/dataDisk/{dataDiskId}	Get Data Disk
PUT	/ns/{nsId}/resources/dataDisk/{dataDiskId}	Upsize Data Disk
DELETE	/ns/{nsId}/resources/dataDisk/{dataDiskId}	Delete Data Disk

DataDisk 관리 API (+8개)

[VM 기준, 가용 DataDisk 리스트 조회](#)

[VM 기준, DataDisk VM 연동/분리](#)

DataDisk 리스트 조회

DataDisk 생성

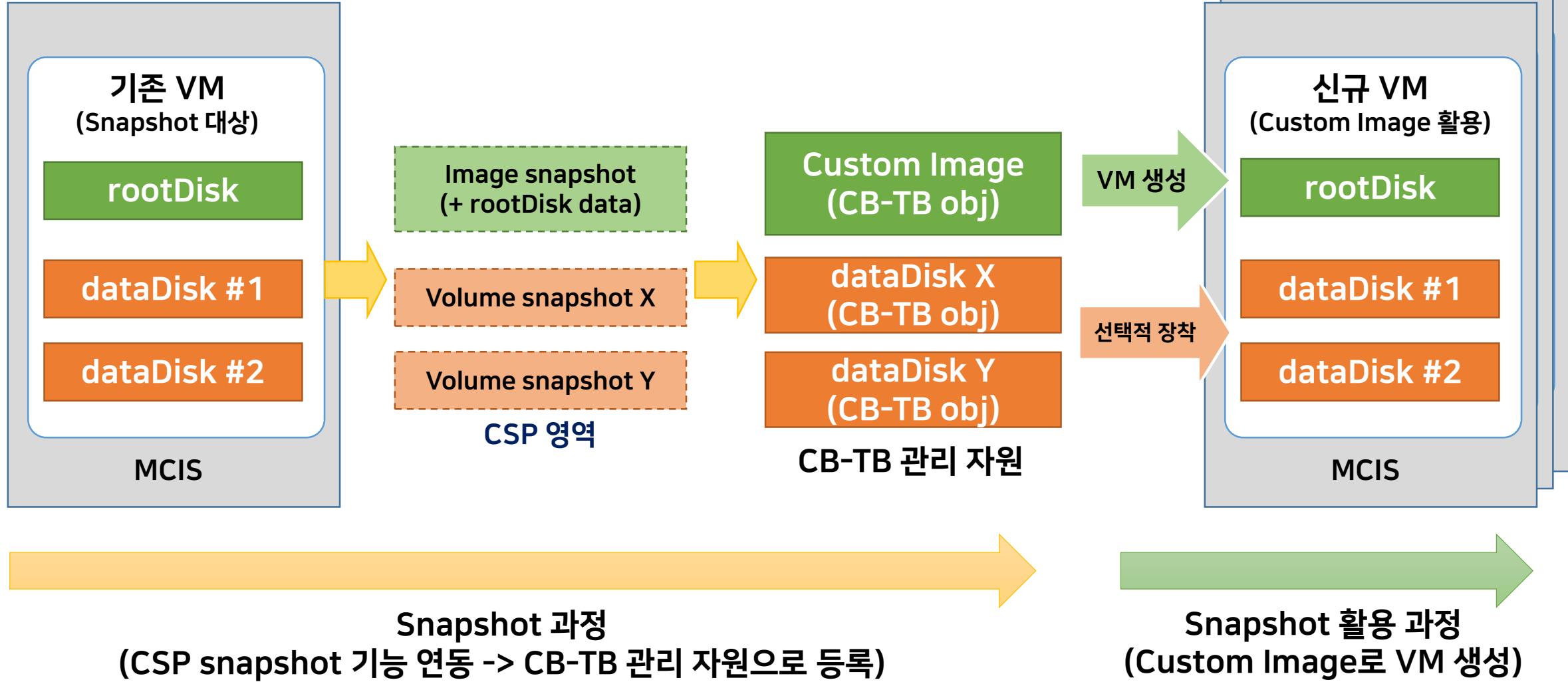
DataDisk 전체 삭제

DataDisk 조회

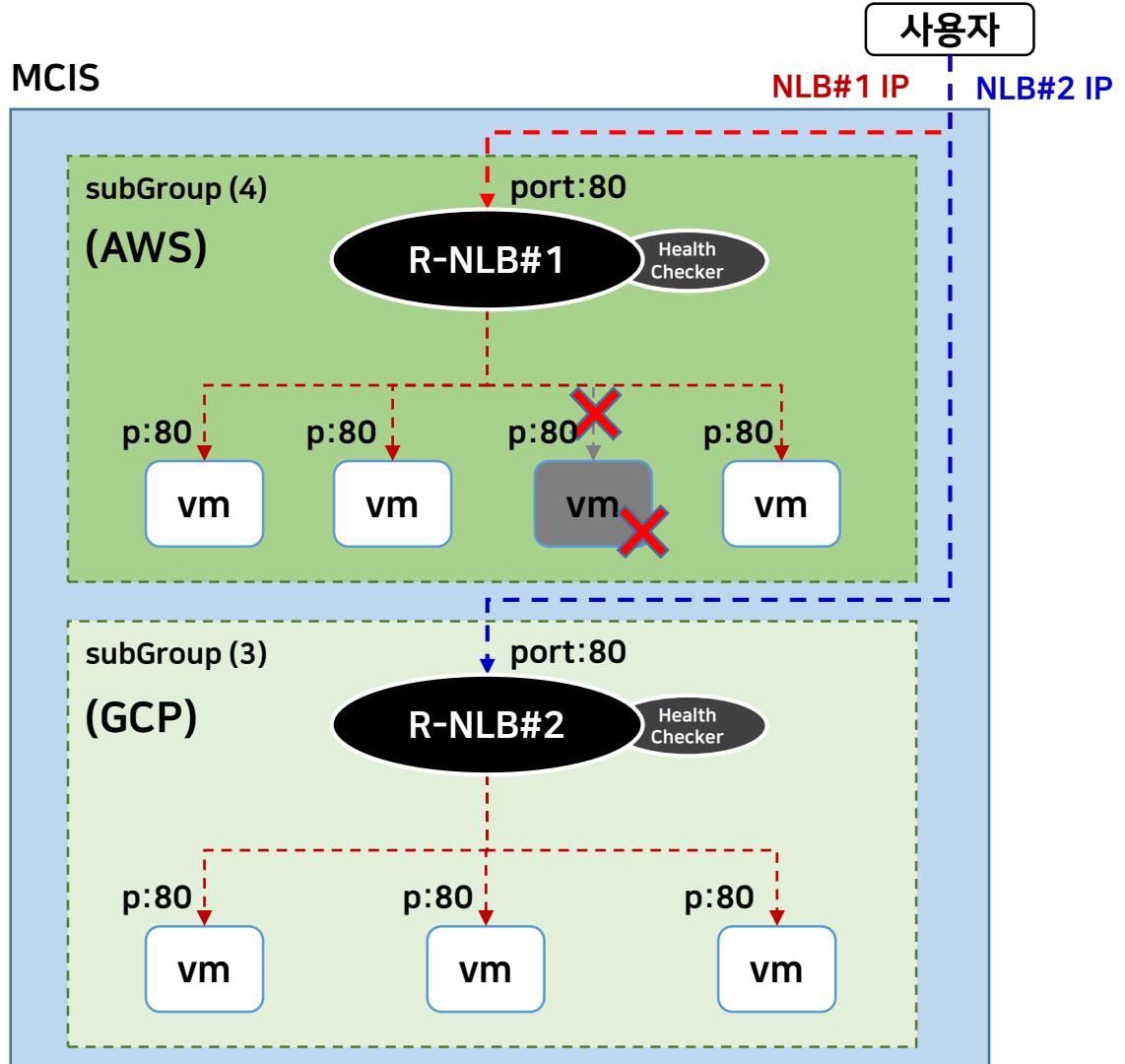
DataDisk 크기 변경

DataDisk 삭제

1-2) VM Snapshot 기능



1-3) [Global, Regional] NLB



Regional NLB (CSP의 NLB 서비스 연동)

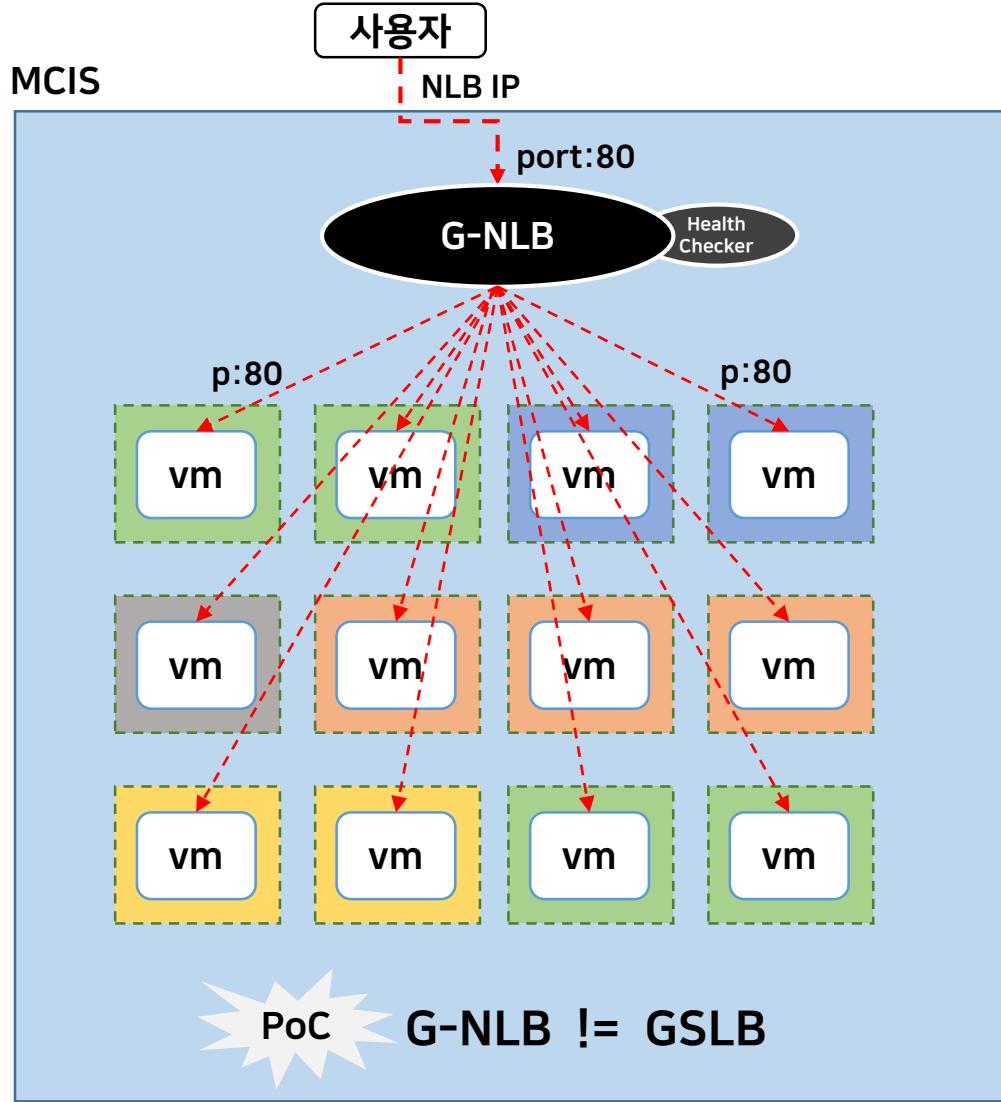
Regional NLB

- CSP의 NLB 서비스 연동 (*L4 NLB)
 - CSP가 HW를 기반으로 직접 제공하는 서비스이므로 고성능
 - 이용료 과금
- MCIS 내부의 SubGroup 단위로 연동
 - CSP에 종속된 NLB 서비스는 일반적으로 동일 CSP 및 리전(VPC) 내의 자원들만 연동 가능하므로, 편의성을 고려

* Layer 4 NLB

Layer 4 (TCP, UDP)에서 IP와 Port를 활용하여 네트워크 부하를 연결된 서버들로 분산하는 것을 의미

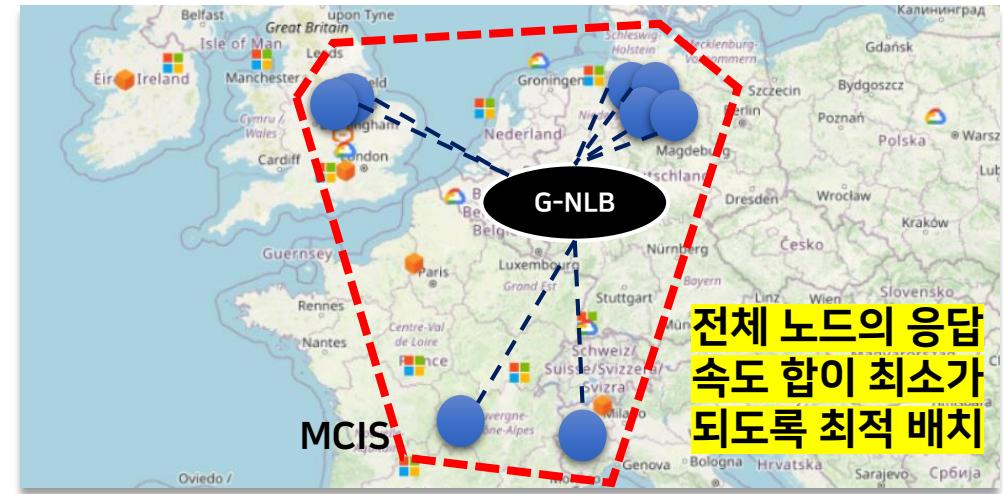
1-3) [Global, Regional] NLB



Global NLB (VM에 SW로 배포된 NLB 서비스)

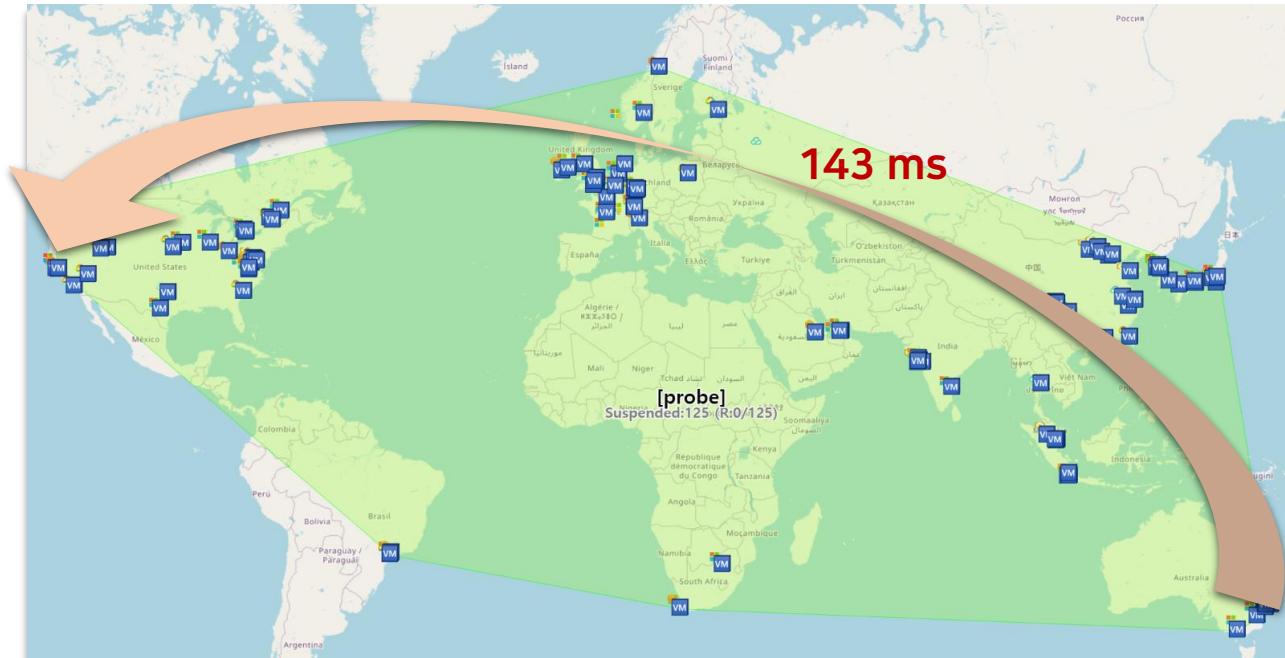
Global NLB

- MCIS 내의 모든 VM을 Public IP를 기반으로 연동
- SW 기반 NLB를 배포하여 서비스 형태로 제공 (L4)
 - Script 기반으로 HAProxy 제공
 - Script 변경을 통해 사용자 선호 SW로 변경 가능
- PoC 단계 (Global DNS인 GSLB와 다름)
 - 최적 배치 기능을 통해, 최적의 NLB Host 배포



특징: 최적 배치 기능을 통한, 최적의 NLB Host 배포

1-3) [Global, Regional] NLB: 최적 배치 (1/2)

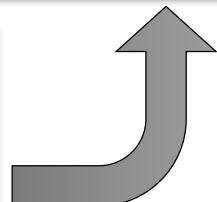


POST /systemMcis Create System MCIS Dynamically for Special Purpose in NS:system-purpose-common-ns

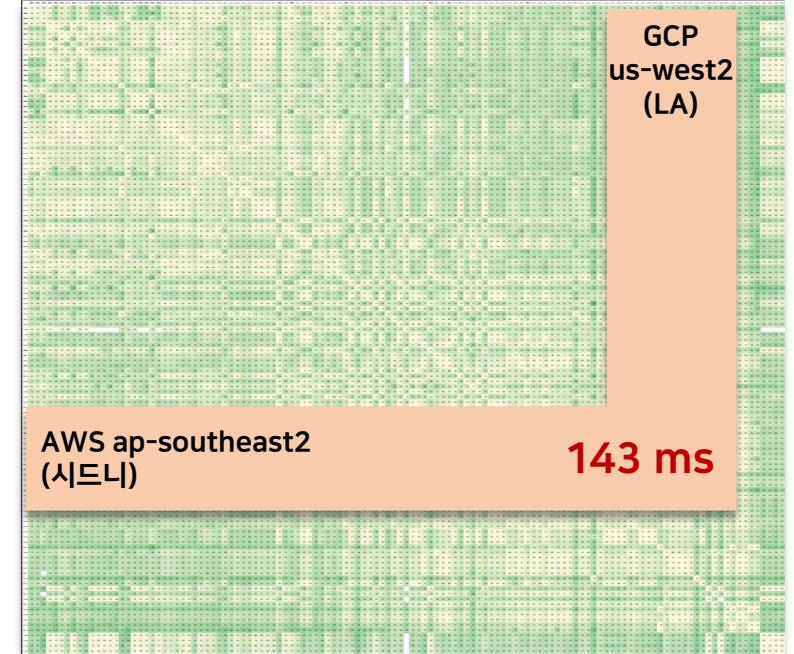
Create System MCIS Dynamically for Special Purpose

Parameters

Name	Description
option string {query}	Option for the purpose of system MCIS
probe	probe



모든 클라우드 리전에
Network Probe 배치

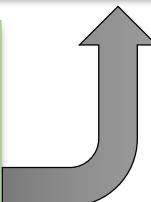


GET /ns/{nsId}/benchmarkLatency/mcis/{mcisId} Run MCIS benchmark

Run MCIS benchmark for network latency

Parameters

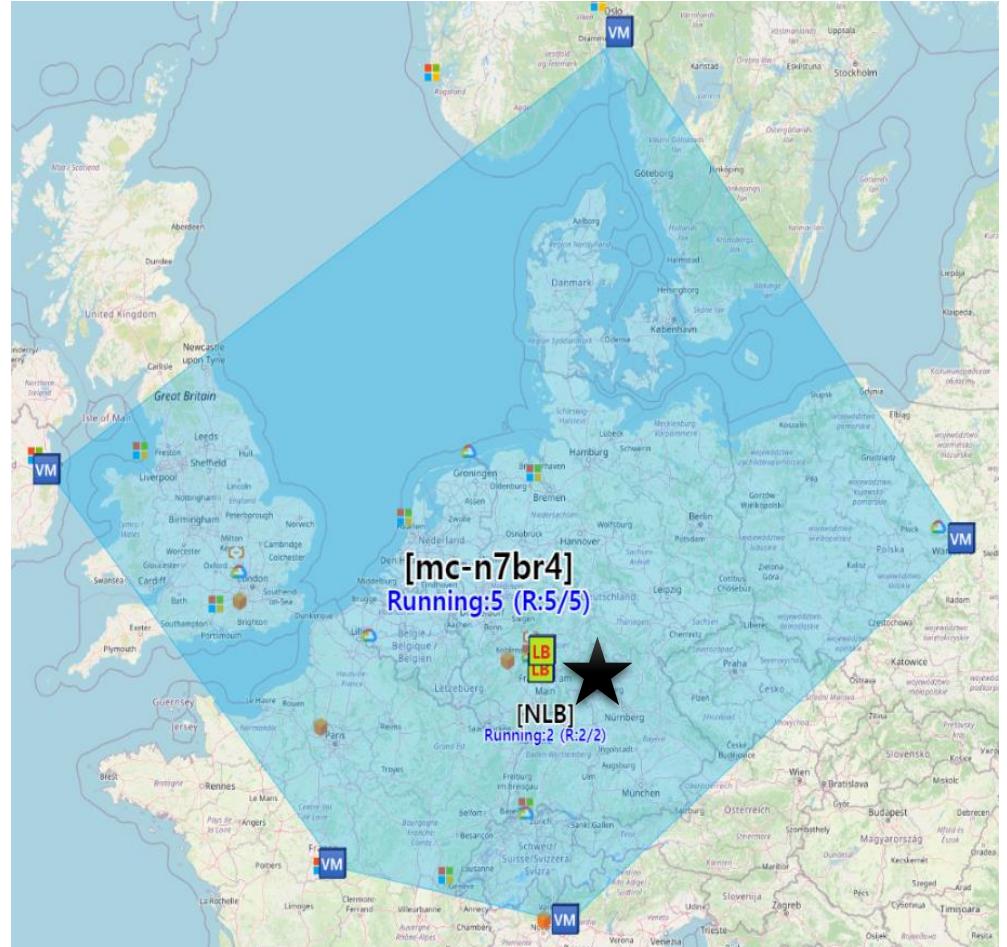
Name	Description
nsId * required string {path}	Namespace ID
system-purpose-common-ns	
mcisId * required string {path}	MCIS ID
probe	



모든 리전간
Network Latency 측정

1-3) [Global, Regional] NLB: 최적 배치 (2/2)

	VM#1 (리전 a)	VM#2 (리전 s)	VM#3 (리전 d)	VM#4 (리전 f)	VM#5 (리전 g)	VM#6 (리전 h)	응답시간 총합 (ms)	평가 순 위
리전 z	20	34	234	345	123	45	801	4
리전 x	123	456	54	34	45	78	790	3
리전 c	34	56	23	34	56	67	270	1 
리전 v	456	234	223	125	34	67	1139	6
리전 b	34	54	34	188	78	34	422	2
리전 n	213	56	36	567	23	89	984	5



NLB 최적 CSP 및 리전 선정
(NLB와 전체 VM 사이의 응답속도 합을 최소화하는 선정)

[CB-MapUI for Cloud-Barista Tumblebug]

Configuration

TB IP:
localhost

MCIS Setting

NS ID:
ns01

MCIS Control

Status

EU MCIS 생성 [R:7 // VM: 24]

MCIS Provisioning

Location-based Create MCIS Clear Selection OS Image Type Ubuntu 18.04 Disk Size default CPU range (N): 1 ~ 8 Memory range (GB): 0.5 ~ 8

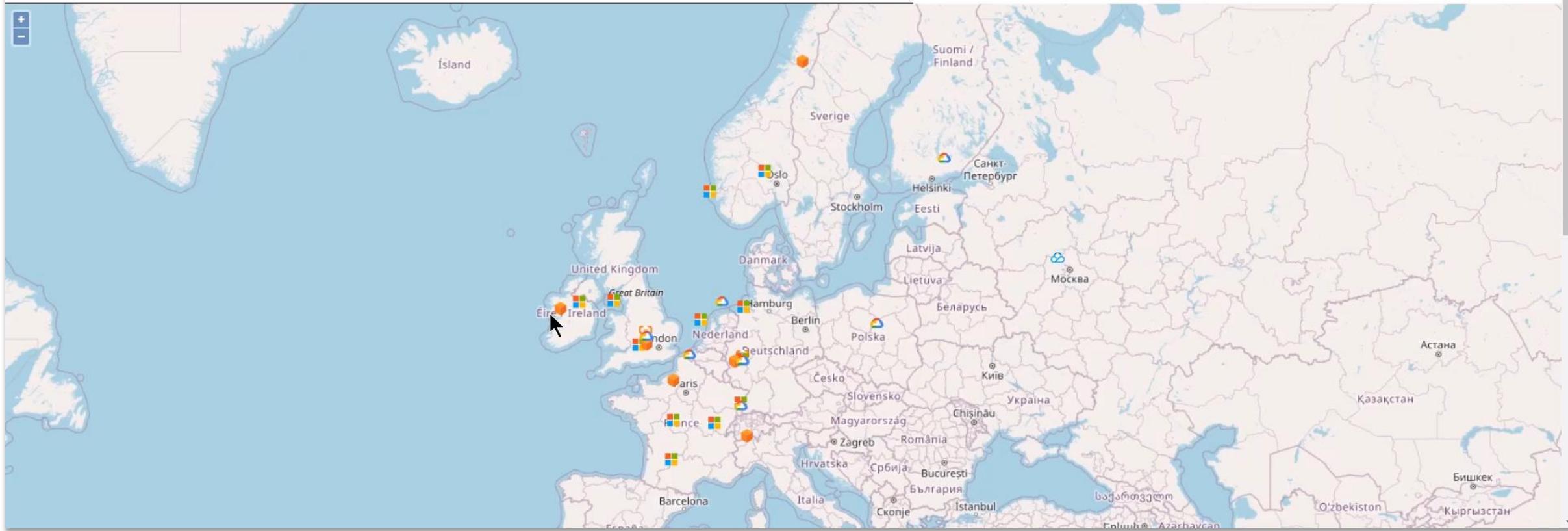
MCIS Application

FPSGame(Xonotic) Start App Stop App Status App Remote Command MCIS Access: Get Access Info Save SSH Key

[Started MCIS configuration]

- [VM-1] Location: 53.4168, -10.1385 | Best Spec:

Display recommendation table:

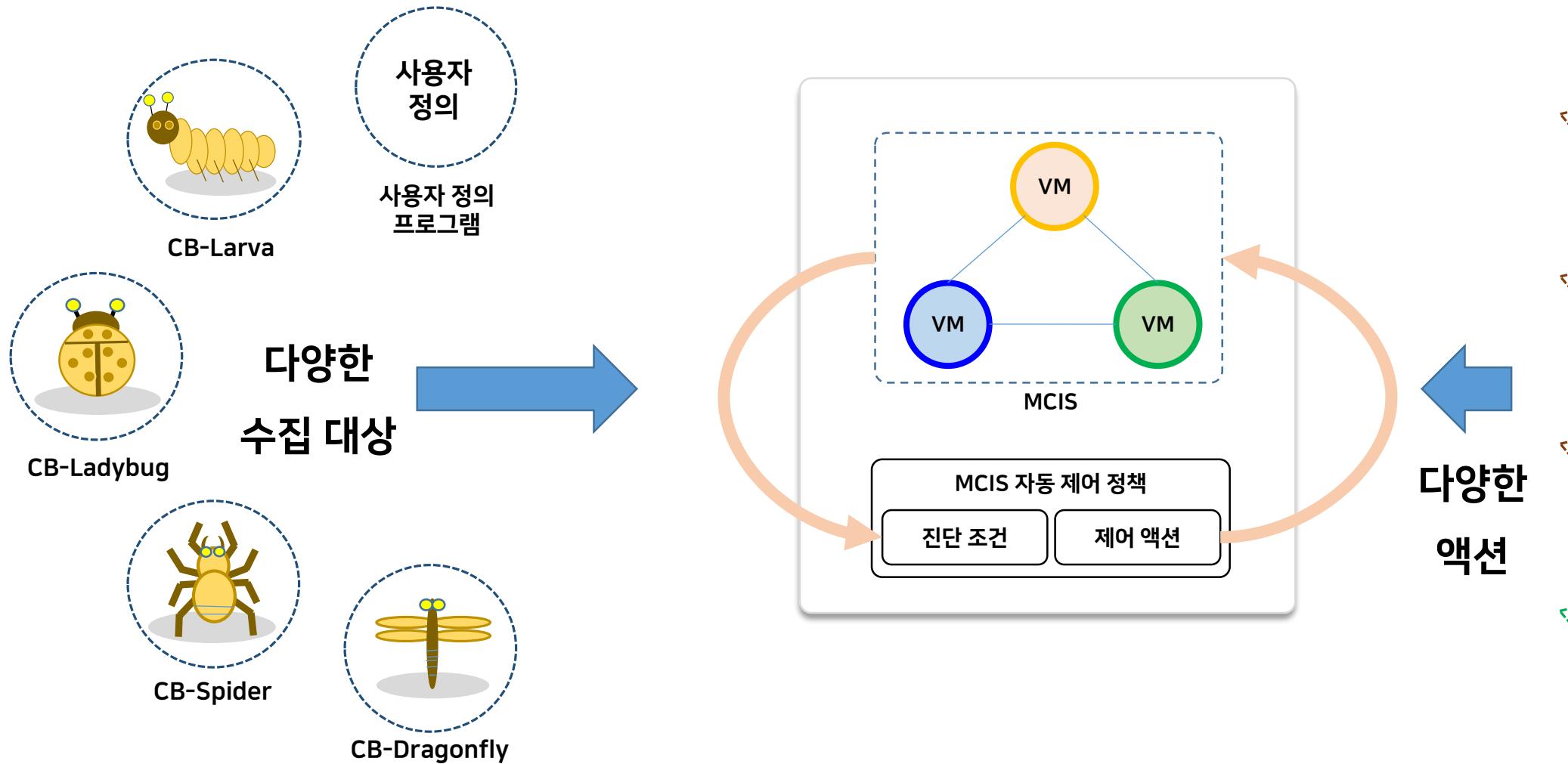




CB-TB 코르타도 (v0.7.0) 주요 개선 포인트

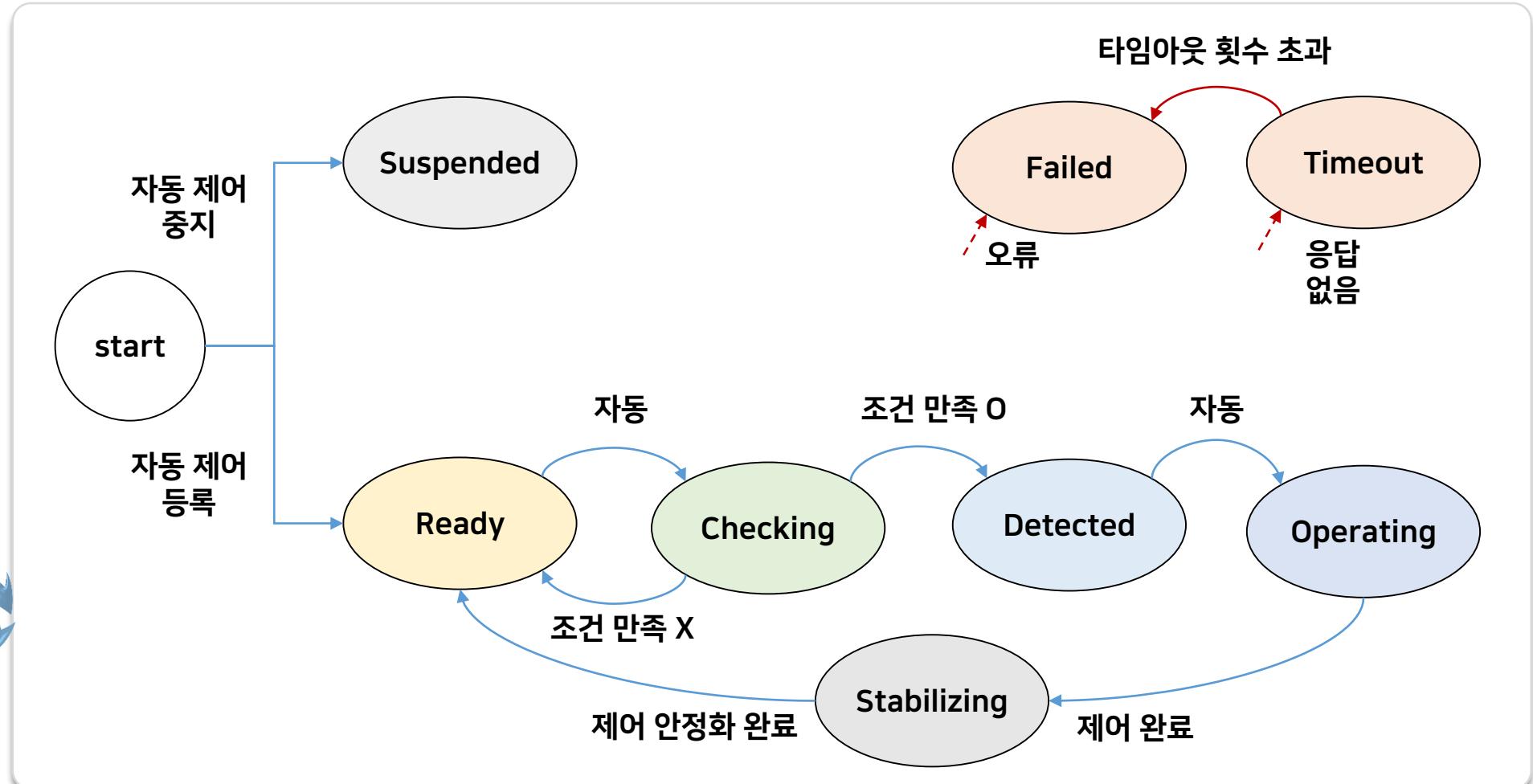
- 1) MCIS 인프라 형상 개선 (+DataDisk, +Snapshot, +NLB)
- 2) MCIS 자동 제어 안정화 및 개선 (+MC-AutoScale)

2) MCIS 자동 제어 안정화 및 개선



2) MCIS 자동 제어 메커니즘 안정화: State-Machine

- State-Machine 메커니즘
 - 자동 제어 컨트롤러는 정책 오브젝트 상태를 기준으로 동작
 - 시스템 중단 및 재실행 시에도 정상 동작 가능

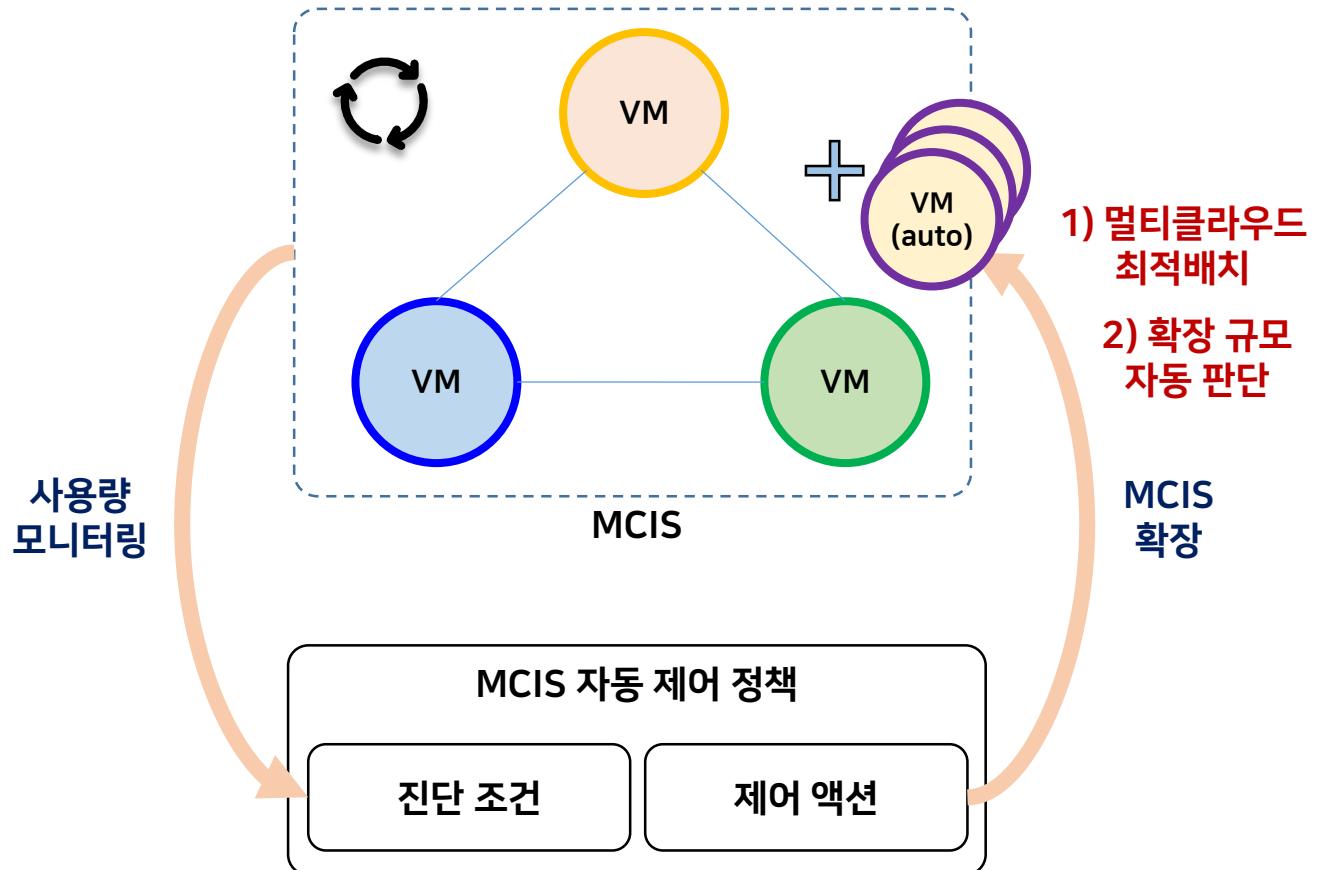


- 주기적으로 정책 오브젝트 상태 확인
- 상태에 맞는 동작 수행

2) MCIS 자동 제어 메커니즘: MC-AutoScale

MC-AutoScale

- 자동제어 메커니즘을 기반으로 멀티클라우드 단위로 ScaleOut을 자동으로 수행하는 기능
 - ScaleOut 멀티클라우드 최적 배치
 - ScaleOut할 노드를 워크로드 상황 및 요구사항에 따라 최적 배치
 - ScaleOut 확장 규모 자동 판단
 - 워크로드를 분석하여, ScaleOut 추가 노드 수 자동 판단
 - ScaleOut 기존 서비스 재구성
 - PostAction을 활용하여, 기존 서비스 구성 재구성



Cloud-Barista Tumblebug Map × Swagger UI × Weave Scope ×

localhost:1324

[CB-MapUI for Cloud-Barista Tumblebug]

Configuration TB IP: localhost TB Port: 1323 Username: default Password: Total CSP regions: MCIS Display: Show CSP Hide/Show Clear map Display Interval: 10

MCIS Setting NS ID: ns01 MCIS ID: mcis-auto SubGroup ID: g1 VM ID: g1-1 Multi-Cloud NLB: Add MC-NLB Add NLB Del NLB Register resource Overview resource

MCIS Control Status Suspend Resume Reboot Refine Terminate Delete Release resource

MCIS Provisioning Location-based Create MCIS Clear Selection OS Image Type Ubuntu 18.04 Disk Size default CPU range (N): 1 ~ 8 Memory range (GB): 0.5 ~ 8

MCIS Application Monitor(WeaveScop) Start App Stop App Status App Remote Command MCIS Access: Get Access Info Save SSH Key

[Complete: Deployed App]
{
resultArray: [
{
mcisId: mcis-auto
vmId: g3-1
}

Display recommendation table:

[mcis-auto] Running:14 (R:14/14)

[df] Running:1 (R:1/1)



멀티클라우드 서비스 공통 플랫폼

부 록

코르타도(Cortado) 한잔 어떠세요 ?



[참고] CB-Tumblebug 참여 방법 (환영합니다..^^)

Search or jump to... Pull requests Issues Marketplace Explore

cloud-barista / cb-tumblebug

Code Issues 23 Pull requests 1 Discussions Actions Projects Wiki Security Insights Settings

master 2 branches 21 tags Go to file Add file + Code +

seokho-son Merge pull request #580 from jihoon-seo/210603_Enhance_FilterSpecsByRange

.github Bump actions/cache from 2.1.5 to 2.1.6 8 days ago

assets Add cloud regions last month

conf Make a section for not touching 8 days ago

src Enhance FilterSpecsByRange function yesterday

.all-contributorsrc docs: update all-contributorsrc [skip ci] 23 days ago

.ignore Enhance FilterSpecsByRange function yesterday

CHANGELOG.md Update changelog for espresso 6 months ago

CODEOWNERS Update CODEOWNERS 5 months ago

CODE_OF_CONDUCT.md Create CODE_OF_CONDUCT.md 4 months ago

CONTRIBUTING.md Update CONTRIBUTING.md 25 days ago

README.md

CB-Tumblebug (Multi-Cloud Infra Service Management)

go report A+ build passing go 66.2% go.mod v1.16 repo size 28.7 MB go reference API Doc Swagger

release v0.3.0-espresso release(dev) v0.3.7 license Apache-2.0 Slack SIG-TB

all contributors 12

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

CB-Tumblebug GitHub

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

Contributors

Seokho Son	Jihoon Seo	Yunkon (Alvin) Kim	jmleefree	ByoungSeob Kim	Sooyoung Kim	KANG DONG JAE
Youngwoo-Jung	Sean Oh	MZC-CSC	Eunsang	hyokyung	pjini	sunmi
sglim	jangh-lee	이도훈	Park Beomsu	Hassan Alsamahi	Taegeon An	INHYO
Modney	Seongbin Bernie Cho	Gibaek Nam	Abidin Durdu	soyeon Park	Jayita Pramanik	Mukul Kolpe
EmmanuelMarianMat	Carlos Felix	Stuart Gilbert	Ketan Deshmukh	Triona Barrow	BamButz	dogfootman

[참고] CB-Tumblebug 사용 방법

모든 정보는 여기에:

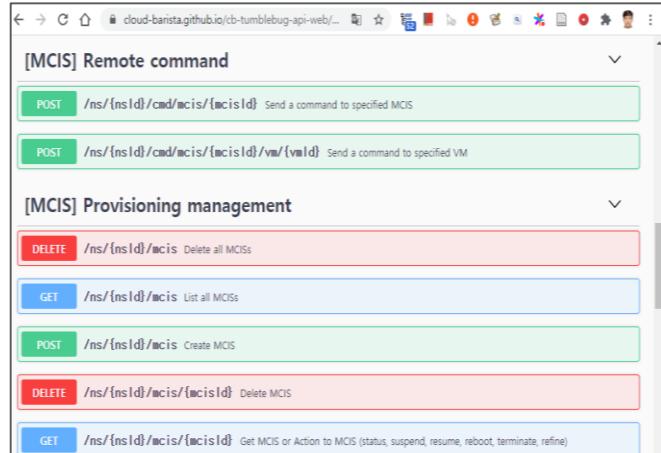
<https://github.com/cloud-barista/cb-tumblebug>



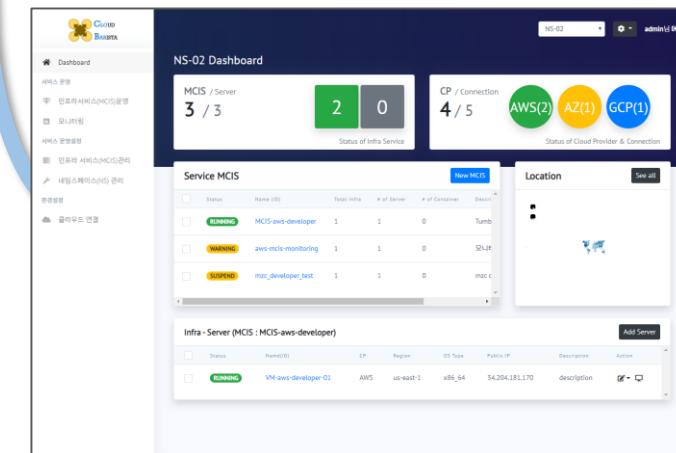
CB-Tumblebug 서버



지도 기반 GUI 클라이언트



REST API Swagger dashboard



Web 기반 GUI 클라이언트

- CB-TB 서버 실행
- CB-TB 제어
 - REST API (MapUI, Dashboard,...)
 - 스크립트
- CB-TB로 생성한 MCIS 맛보기
 - MCIS VM에 SSH 접속
 - MCIS에 영상회의 서버 자동 배포
 - MCIS에 클러스터 모니터링 도구 배포
 - MCIS에 Ansible 환경 자동 구성
 - MCIS에 Nginx 자동 배포
 - MCIS 토이 게임 서버 배치
 - ...

[참고] CB-TB 서버 실행

(1) CB-Tumblebug 빌드 환경 구성

- 필요 패키지 또는 도구 설치 (git, make, go, …)
- CB-Tumblebug 소스 다운로드
 - `git clone https://github.com/cloud-barista/cb-tumblebug.git`
`$HOME/go/src/github.com/cloud-barista/cb-tumblebug`
- CB-Tumblebug 실행에 필요한 환경변수 설정
 - `source ./conf/setup.env`
- (2) CB-Tumblebug 빌드
 - 빌드 명령 실행
 - `make`
- (3) CB-Tumblebug 실행
 - CB-Spider 실행 (다른 탭에서)
 - 실행 명령 실행
 - `make run`





[참고] CB-TB 제어하기 (REST API)

The screenshot shows the CB-Tumblebug API Swagger UI interface. It includes sections for [MCIS] Remote command, [MCIS] Provisioning management, [MCIS] Resource monitor (Developer), and [MCIS] Auto control policy management (WIP). Each section lists various REST API endpoints with their methods, URLs, and descriptions.

- [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 specific endpoint from the Swagger UI: POST /ns/{nsId}/mcis. The interface displays the 'Create MCIS' form with fields for 'Name' (nsId) and 'mcisReq' (mcisId). Below the form, there is a detailed JSON schema for the mcisReq object, which includes properties like 'connectionType', 'imageId', 'label', 'memorySize', 'osType', 'specType', 'storageType', and 'subnetId'. The 'Responses' section indicates a successful response with status code 200 and type application/json.

Swagger UI에서 직접 API 호출

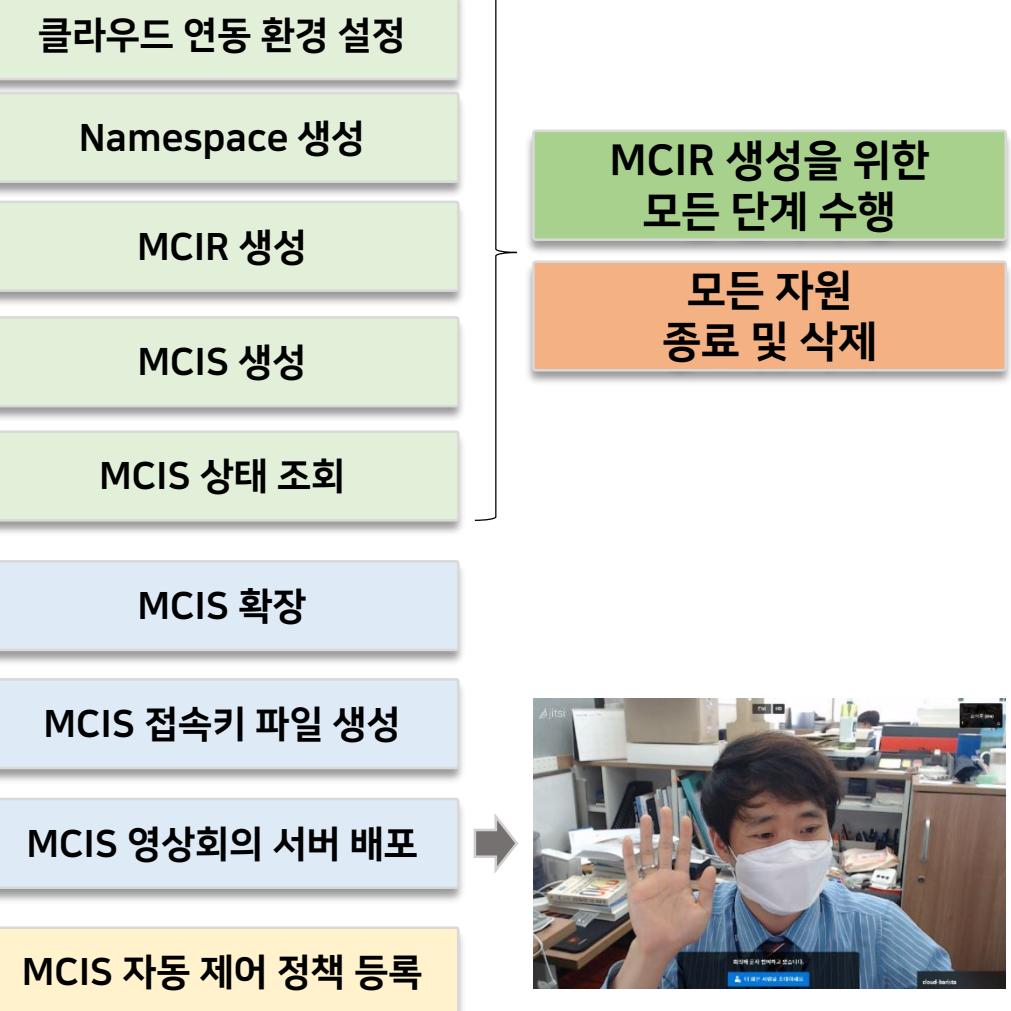
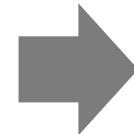
```
206 // RestDelMcis godoc
207 // @Summary Delete MCIS
208 // @Description Delete MCIS
209 // @Tags [MCIS] Provisioning management
210 // @Accept json
211 // @Produce json
212 // @Param nsId path string true "Namespace ID"
213 // @Param mcisId path string true "MCIS ID"
214 // @Success 200 {object} 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

[참고] CB-TB 제어하기 (스크립트)

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

- credentials.conf**
- CSP 사용자 인증 크레덴셜
- conf.env**
- 클라우드 리전, 이미지 ID 등
- testSet.env**
- 테스트 대상 클라우드 리전 지정



위치: cb-tumblebug/src/testclient/

[사용 및 개발 편의를 위한 다양한 테스트 스크립트]

[참고] 다양한 클라우드로 구성된 MCIS 시험 지원 (스크립트)

- 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) -
AwsEuSouth1=$((++IY)) # Location: Europe (Milan) - Opt-In required
```

testSet.env

[참고] 다양한 클라우드로 구성된 MCIS 시험 지원 (스크립트)

[Configuration in `../testSet01.env` & `../conf.env` files]

1) System Endpoints

- Tumblebug Server : `localhost:1323`
- Spider Server : `localhost:1024`

2) Enabled Clouds and Regions

- [1] Cloud : aws (enabled regions : 2)
 - [1,1] Region : aws-ap-southeast-1 (Asia Pacific (Singapore))
 - VM SPEC : m4.4xlarge
 - VM IMAGE : Ubuntu 18.04 (ami-061eb2b23f9f8839c)
 - [1,2] Region : aws-ca-central-1 (Canada (Central))
 - VM SPEC : t2.micro
 - VM IMAGE : Ubuntu 18.04 (ami-0d0eaed20348a3389)
- [2] Cloud : gcp (enabled regions : 2)
 - [2,1] Region : gcp-asia-east1 (Changhua County Taiwan [zone:a])
 - VM SPEC : e2-small
 - VM IMAGE : Ubuntu 18.04 (<https://www.googleapis.com/compute/v1/projects/ubuntu-os-cloud/global/images/ubuntu-minimal-1804-bionic-v20191024>)
 - [2,2] Region : gcp-europe-west3 (Frankfurt Germany [zone:a])
 - VM SPEC : e2-small
 - VM IMAGE : Ubuntu 18.04 (<https://www.googleapis.com/compute/v1/projects/ubuntu-os-cloud/global/images/ubuntu-minimal-1804-bionic-v20191024>)
- [3] Cloud : alibaba (enabled regions : 2)
 - [3,1] Region : alibaba-ap-northeast-1 (Japan (Tokyo))
 - VM SPEC : ecs.t5-lc1m2.large
 - VM IMAGE : Ubuntu 18.04 (ubuntu_18_04_x64_20G_alibase_20210824.vhd)
 - [3,2] Region : alibaba-ap-south-1 (Mumbai Zone A)
 - VM SPEC : ecs.t5-lc1m2.large
 - VM IMAGE : Ubuntu 18.04 (ubuntu_18_04_x64_20G_alibase_20210824.vhd)

3) MCIS Configuration

- NameSpace ID : `tb01`
- MCIS ID : `cb-etri02`
- Number of Total VMs : `60`
 - [1] VMs(20) = aws(2) * Replica(10)
 - [2] VMs(20) = gcp(2) * Replica(10)
 - [3] VMs(20) = alibaba(2) * Replica(10)

Confirm the above configuration. Do you want to proceed ? (y/n) :

AWS 리전*2, GCP 리전*2, Alibaba리전*2
VM 복제본을 각 10개씩 배치 요청 (단일 커멘드)

Table: All VMs in the MCIS : cb-etri02

VM-ID	Status	PublicIP	PrivateIP	CloudType	CloudRegion
ali-ap-ne-1-0	Running	8.211.158.0	192.168.4.232	alibaba	ap-northeast-1
ali-ap-ne-1-1	Running	8.211.132.187	192.168.4.230	alibaba	ap-northeast-1
ali-ap-ne-1-2	Running	47.74.39.26	192.168.4.228	alibaba	ap-northeast-1
ali-ap-ne-1-3	Running	8.211.155.198	192.168.4.233	alibaba	ap-northeast-1
ali-ap-ne-1-4	Running	8.209.224.238	192.168.4.226	alibaba	ap-northeast-1
ali-ap-ne-1-5	Running	47.74.27.111	192.168.4.229	alibaba	ap-northeast-1
ali-ap-ne-1-6	Running	8.209.226.75	192.168.4.231	alibaba	ap-northeast-1
ali-ap-ne-1-7	Running			alibaba	ap-northeast-1
ali-ap-ne-1-8	Running			alibaba	ap-northeast-1
ali-ap-ne-1-9	Running			alibaba	ap-northeast-1
ali-ap-south-1-0	Running	147.139.84.121	192.168.8.137	alibaba	ap-south-1
ali-ap-south-1-1	Running	147.139.84.43	192.168.8.135	alibaba	ap-south-1
ali-ap-south-1-2	Running	147.139.40.35	192.168.8.140	alibaba	ap-south-1
ali-ap-south-1-3	Running	147.139.75.135	192.168.8.138	alibaba	ap-south-1
ali-ap-south-1-4	Running	147.139.30.0	192.168.8.132	alibaba	ap-south-1
ali-ap-south-1-5	Running	147.139.85.28	192.168.8.133	alibaba	ap-south-1
ali-ap-south-1-6	Running	147.139.46.151	192.168.8.134	alibaba	ap-south-1
ali-ap-south-1-7	Running			alibaba	ap-south-1
ali-ap-south-1-8	Running			alibaba	ap-south-1
ali-ap-south-1-9	Running			alibaba	ap-south-1
aws-ap-se-1-0	Running	18.139.227.185	192.168.2.199	aws	ap-southeast-1
aws-ap-se-1-1	Running	54.179.156.130	192.168.2.118	aws	ap-southeast-1
aws-ap-se-1-2	Running	52.221.191.243	192.168.2.34	aws	ap-southeast-1
aws-ap-se-1-3	Running	54.251.184.127	192.168.2.185	aws	ap-southeast-1
aws-ap-se-1-4	Running	13.229.130.178	192.168.2.41	aws	ap-southeast-1
aws-ap-se-1-5	Running	52.221.216.250	192.168.2.203	aws	ap-southeast-1
aws-ap-se-1-6	Running	18.136.201.111	192.168.2.223	aws	ap-southeast-1
aws-ap-se-1-7	Running			aws	ap-southeast-1
aws-ap-se-1-8	Running			aws	ap-southeast-1
aws-ap-se-1-9	Running			aws	ap-southeast-1
aws-ca-ct-1-0	Running	35.183.111.146	192.168.4.7	aws	ca-central-1
aws-ca-ct-1-1	Running	15.222.60.92	192.168.4.223	aws	ca-central-1

리전 당 10개의 VM

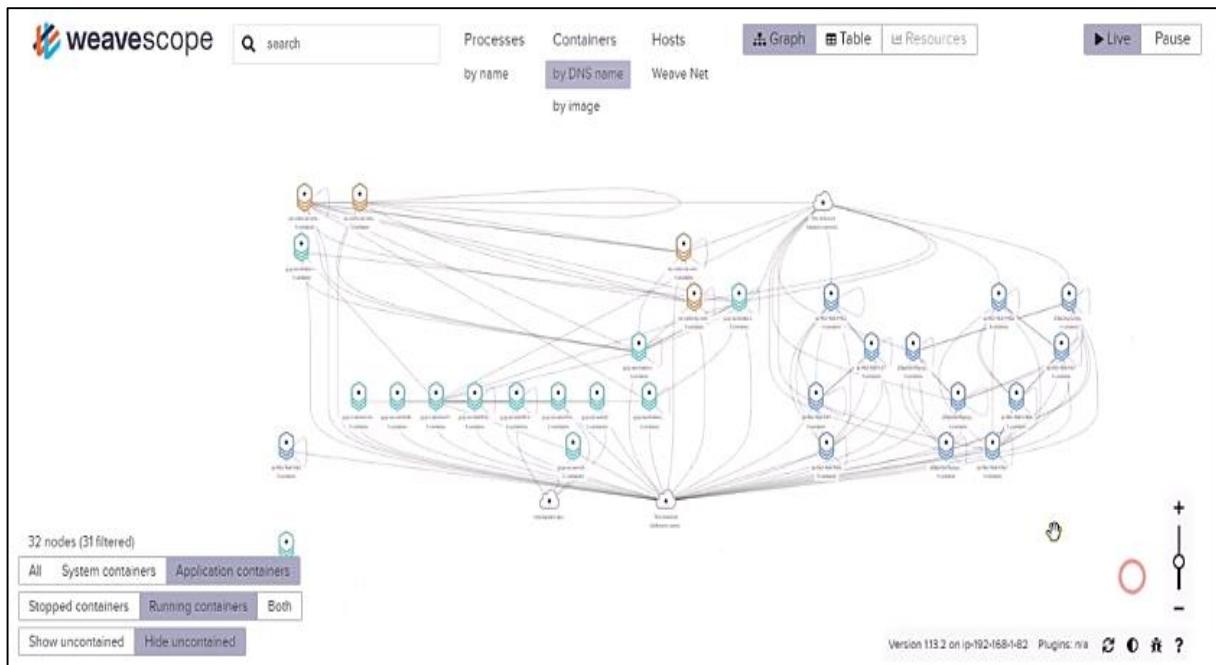
리전 당 10개의 VM

리전 당 10개의 VM

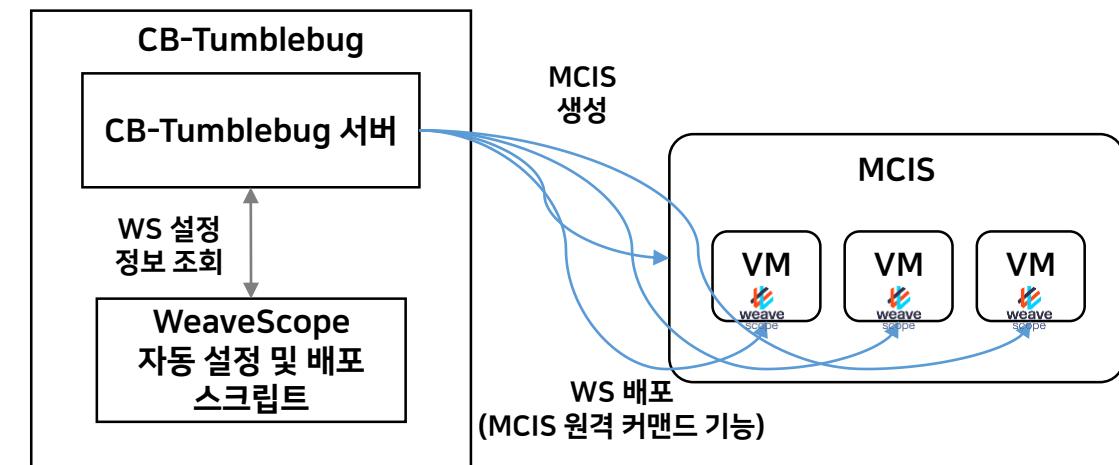
배치 결과

[참고] CB-TB 클러스터 모니터링 도구 (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



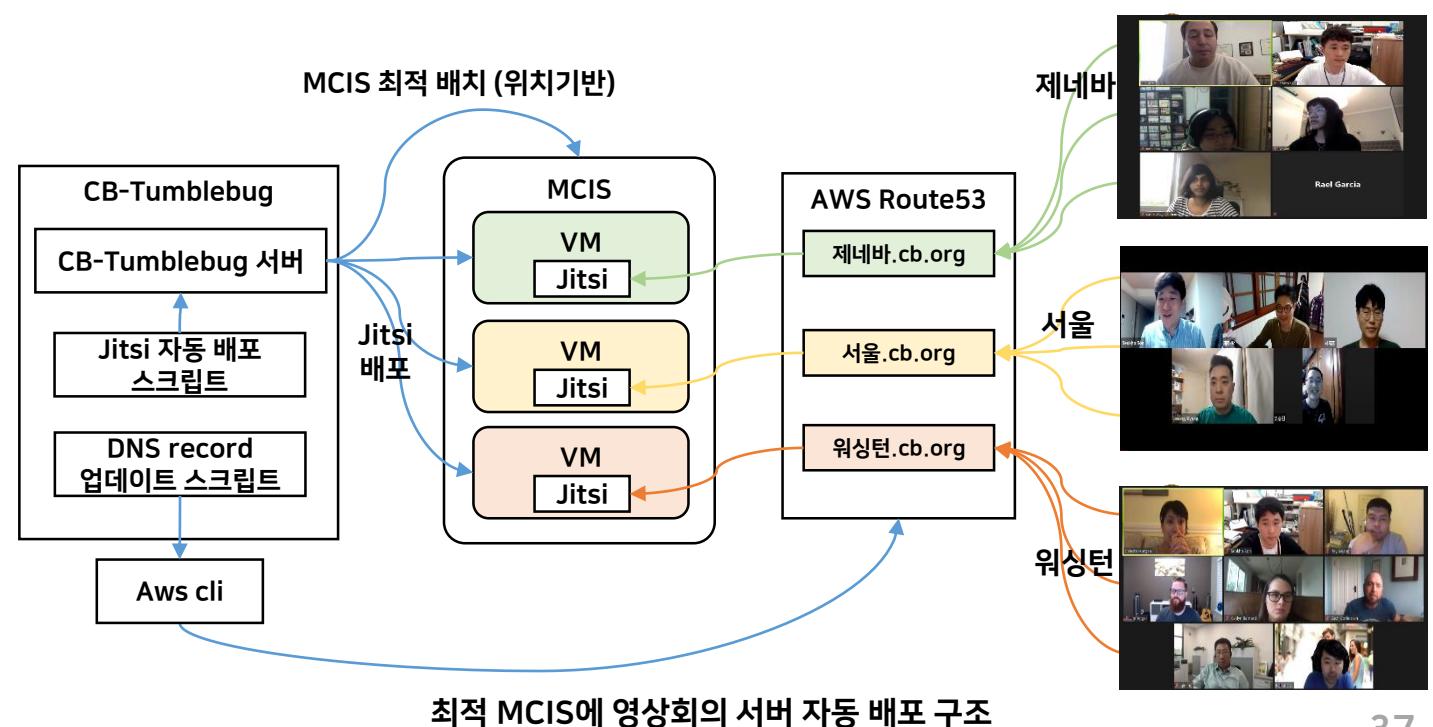
WeaveScope 자동 설정 및 배포 구조

[참고] 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

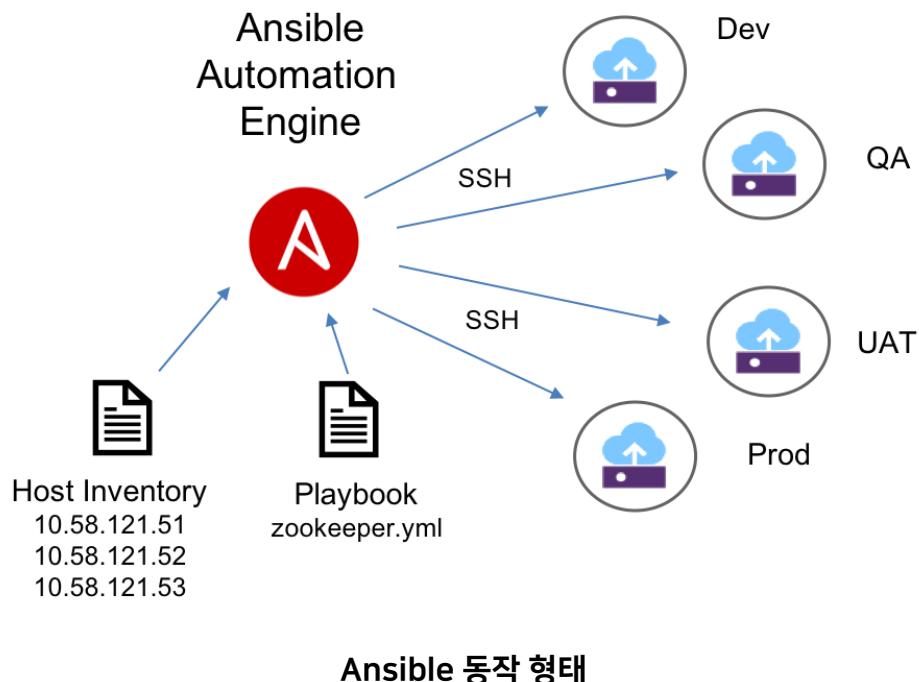


Jitsi 영상회의 접속 화면

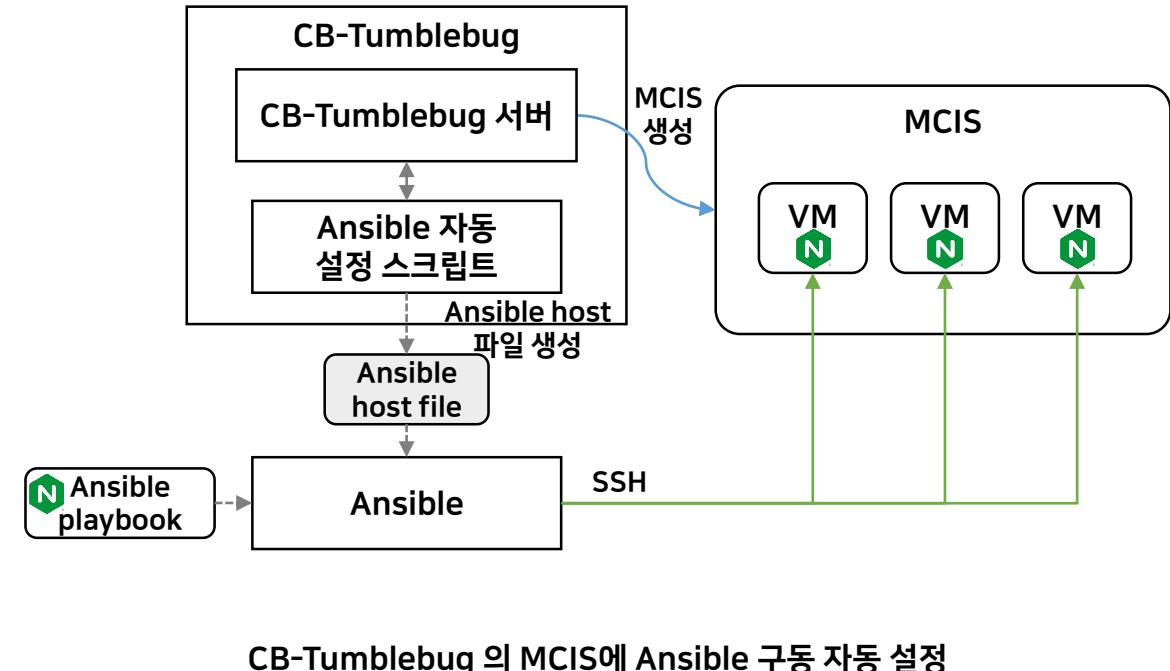


[참고] CB-Tumblebug의 Ansible 사용자 지원 (유스케이스 발굴 및 배포 자동화)

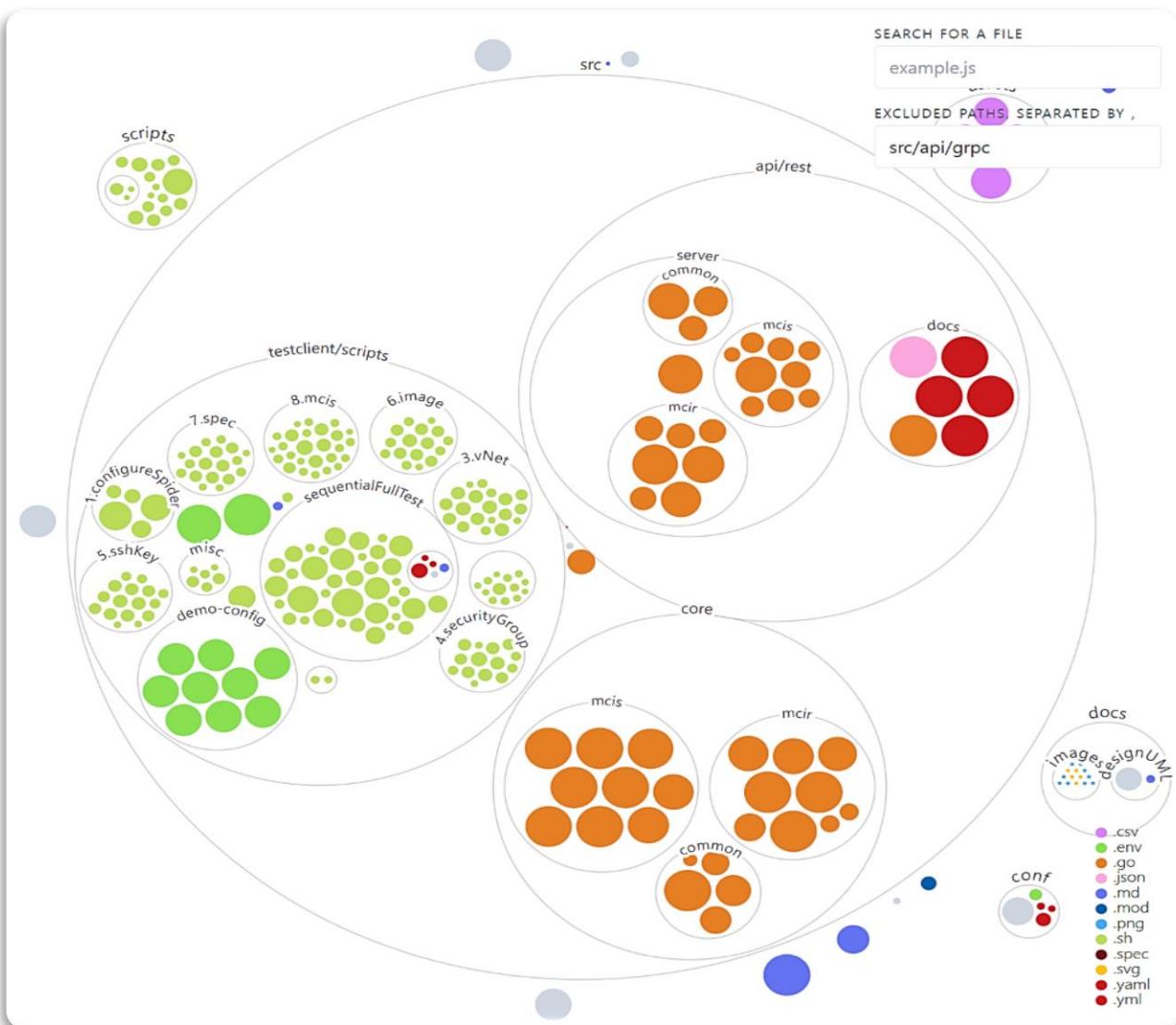
- CB-Tumblebug의 Ansible 사용자 지원
 - Ansible은 Infrastructure as Code를 지향하는 오픈소스 기반의 자동화 관리 도구
 - 구축/관리 하려는 인프라에 SSH로 원격 명령을 전달하는 방식으로 동작
 - CB-Tumblebug의 MCIS에서 Ansible을 바로 사용할 수 있도록 Ansible 설정 자동화 스크립트 개발



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



[참고] CB-Tumblebug 소스코드 구조



- 소스코드

- 저장소: <https://github.com/cloud-barista/cb-tumblebug>
- 개발 언어: Golang v1.19
- goDoc
 - <https://pkg.go.dev/github.com/cloud-barista/cb-tumblebug@main>

클라우드바리스타들의 일곱번째 이야기

멀티클라우드, 컴퓨팅 인프라에 제약없는 서비스 생태계

Cloud-Barista Community the 7th Conference

감사합니다.

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

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

손석호 / contact-to-cloud-barista@googlegroups.com