



CLOUD

BARISTA

멀티 클라우드 서비스 공통 프레임워크

멀티 클라우드 인프라 연동 프레임워크



[CB-Spider]

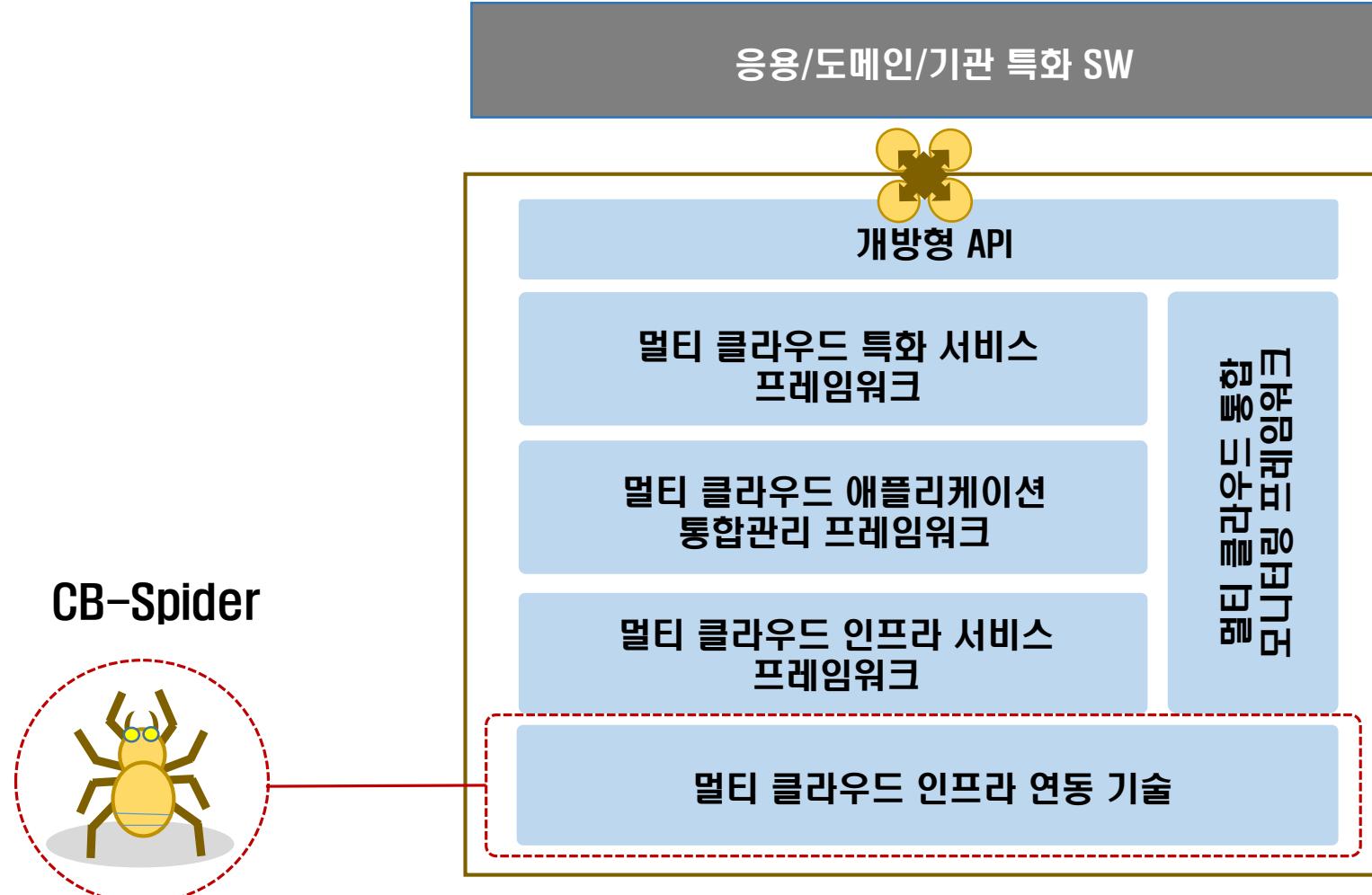
404

This is not the
Single-Cloud you
are looking for.



Cloud-Barista Community, 김병섭 T.L. & F.L.

이번 세션은...



목 차

I CB-Spider 시스템 및 서비스 개요

II CB-Spider 개발 현황

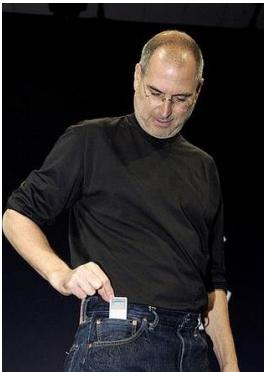
III CB-Spider 개발 로드맵

IV CB-Spider 시연



I. CB-Spider 시스템 및 서비스 개요

<2001.10.23>



iPod.
1,000 songs in your pocket.

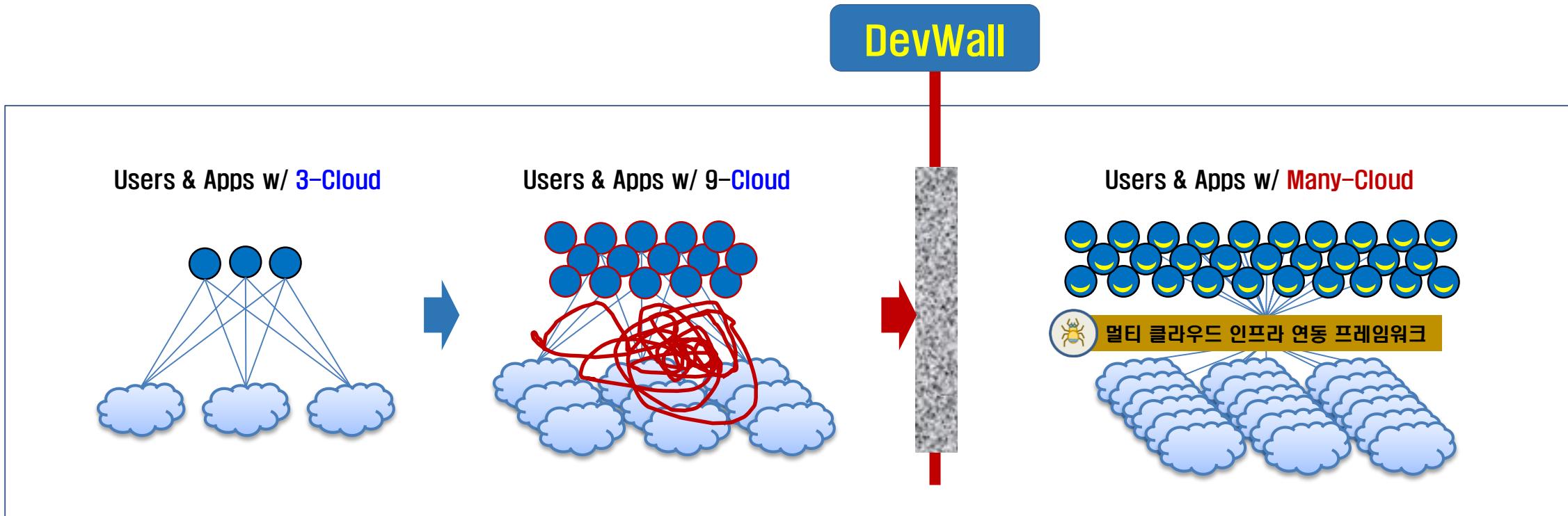
<Today>



CB-Spider.
100+ Regions in your Berry.

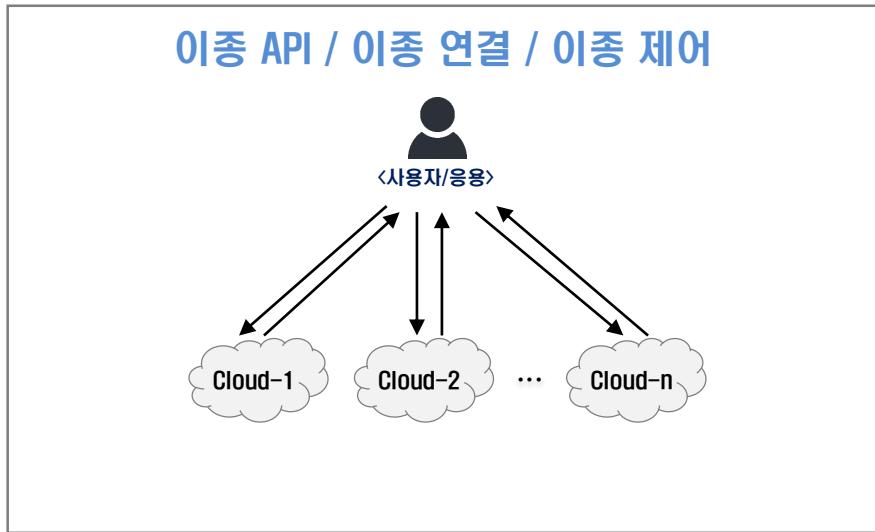
CB-Spider 왜 필요하지?

. *2019 Cloud State Report(RightScale)*: 94% of enterprises are using cloud, 84% of organizations have a multi-cloud strategy.

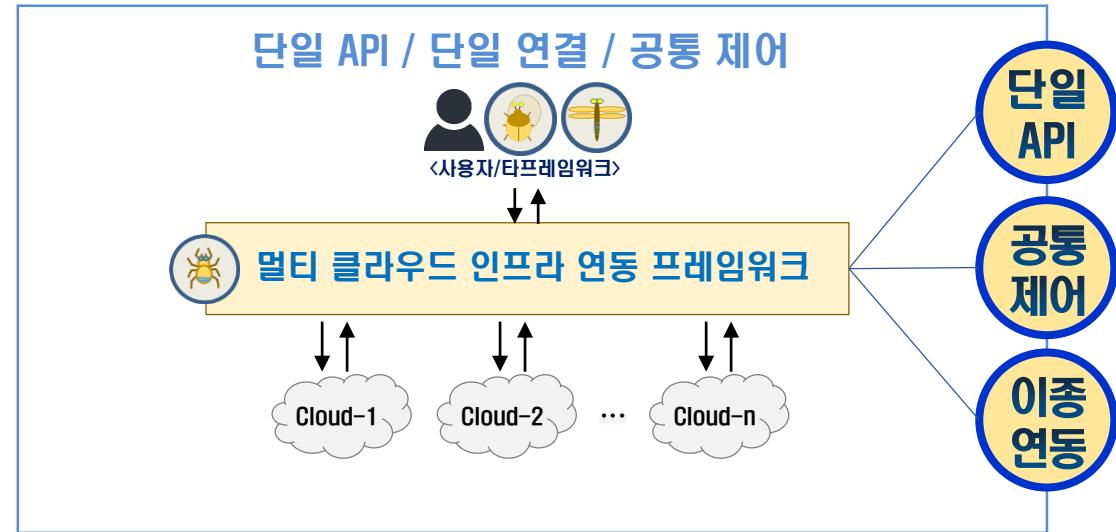


#DeveloperWall

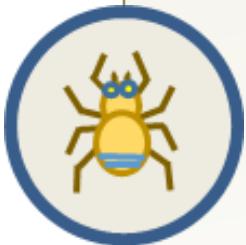
CB-Spider 왜 필요하지?



TO BE



CB-Spider 최종 목표는?



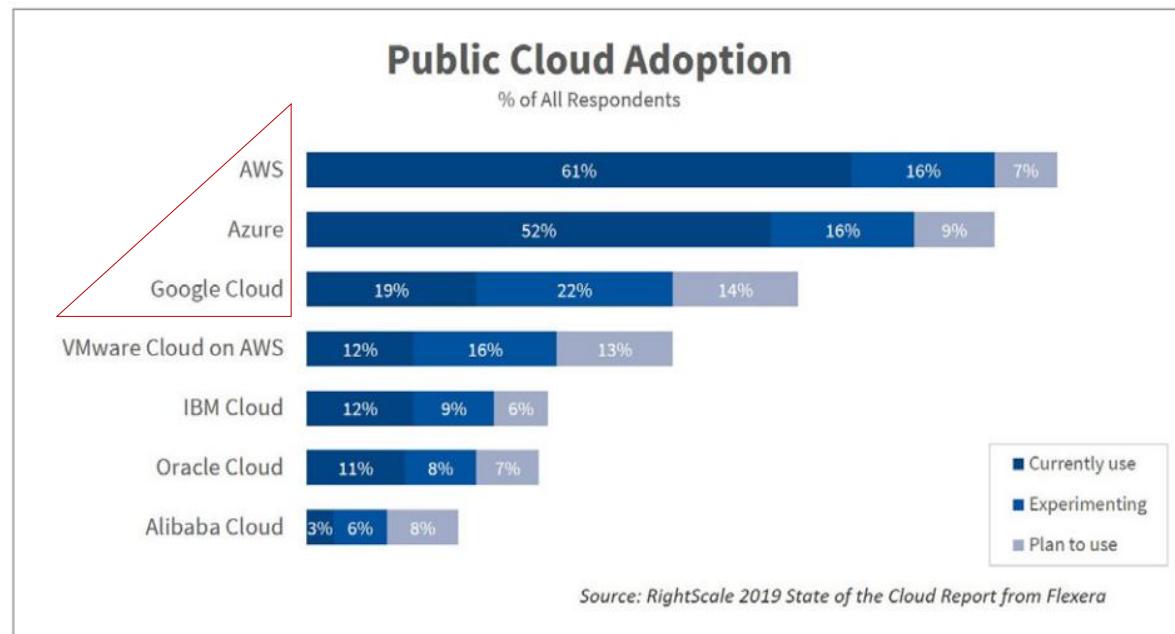
- 전세계 이종 클라우드를 엮을 수 있게 한다.
- 전세계 클라우드를 하나처럼 쓸 수 있게 한다.



Americano는 어떤 클라우드를 엮지?

주로 사용하는 클라우드는?

AWS, Azure, GCP, ...



유명 오픈 소스는?

OpenStack, ...

국내 클라우드는?

Cloudit, ...

급성장하는 ?

Alibaba, ...

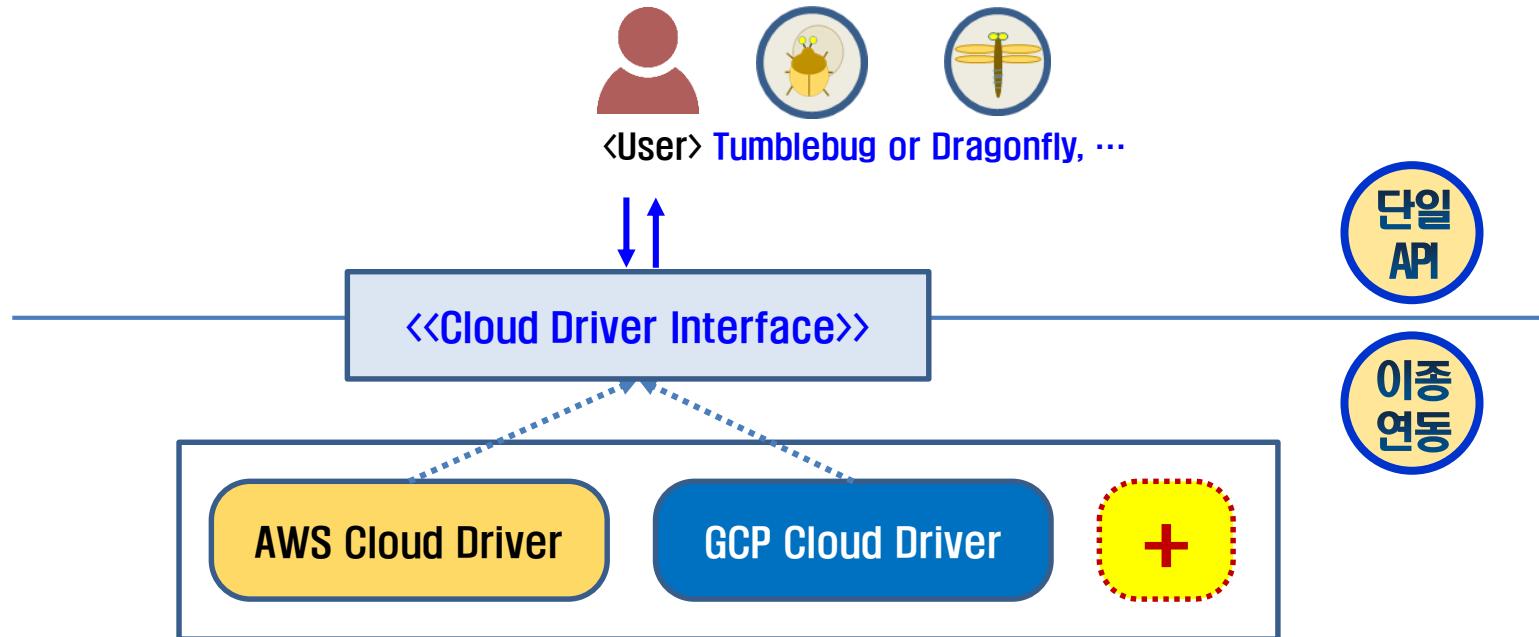


CB-Spider가 클라우드를 엮는 방법 - 플러그인



* **플러그인 기반 멀티 클라우드 인프라 연동 구조 및 공통 제어**

CB-Spider가 클라우드를 엮는 방법 - 드라이버 인터페이스

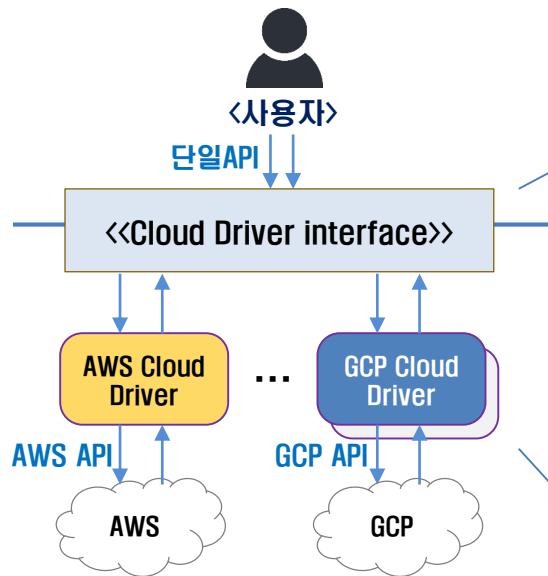


#Go #Interface #Driver → #plugin

CB-Spider가 클라우드를 엮는 방법 - internal

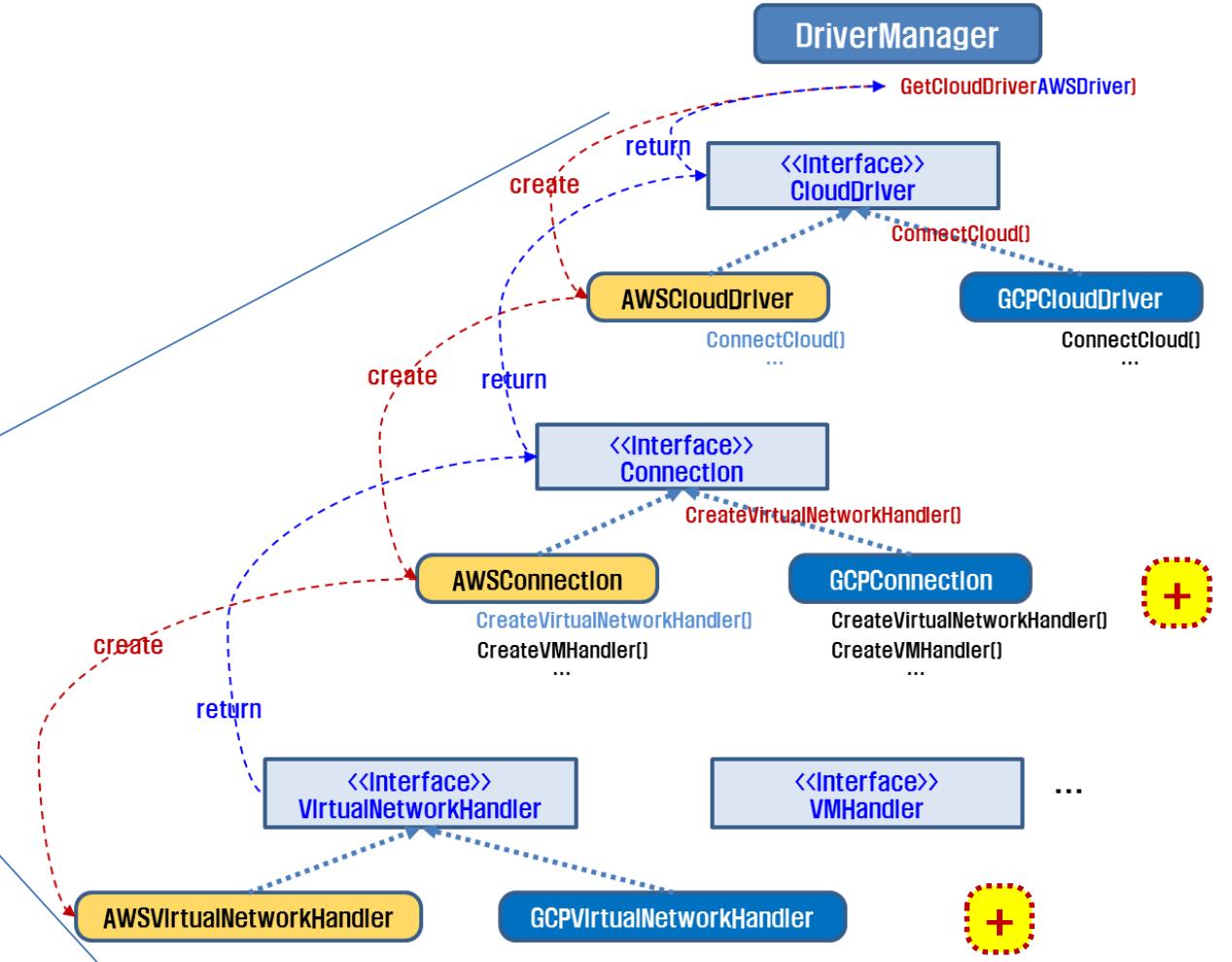
- 드라이버 공통 인터페이스
 - 드라이버 플러그인 방식
 - 공통 인터페이스 구현
 - CSP 전용 API 사용한 구현
 - CSP 당 다수 드라이버 지원
- ※ CSP: Cloud Service Provider

#단일API #드라이버 #플러그인

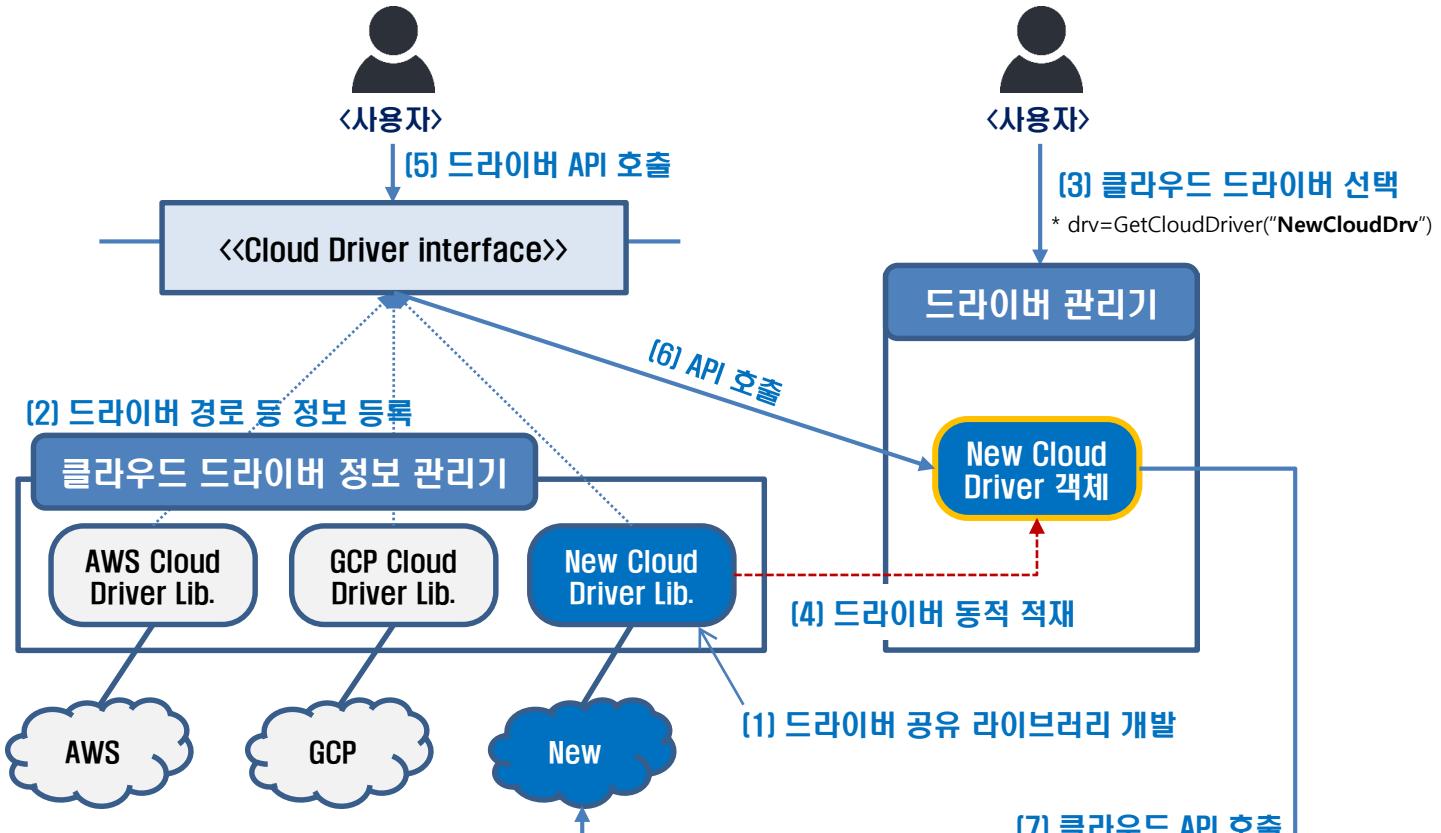


〈드라이버 방식 플러그인 구조〉

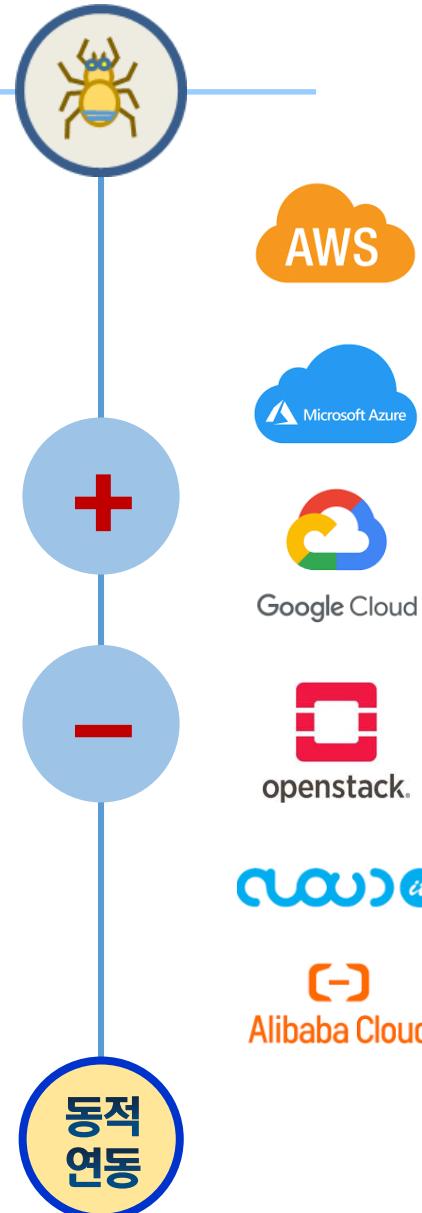
〈인터페이스와 드라이버 구현의 상관 관계〉



CB-Spider가 클라우드를 엮는 방법 - 동적연동

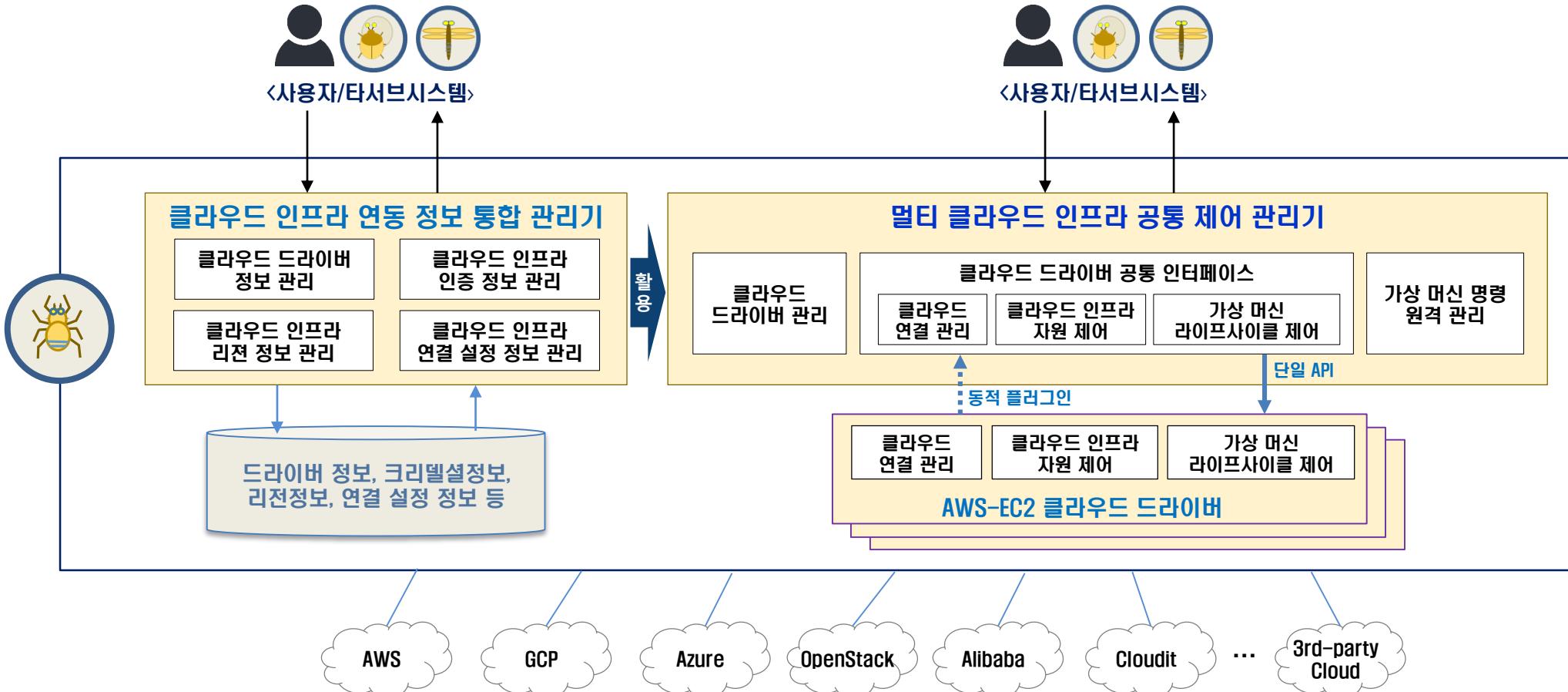


#공유LIB
#동적플러그인
#클라우드동적추가



CB-Spider가 클라우드를 공통 제어하는 방법

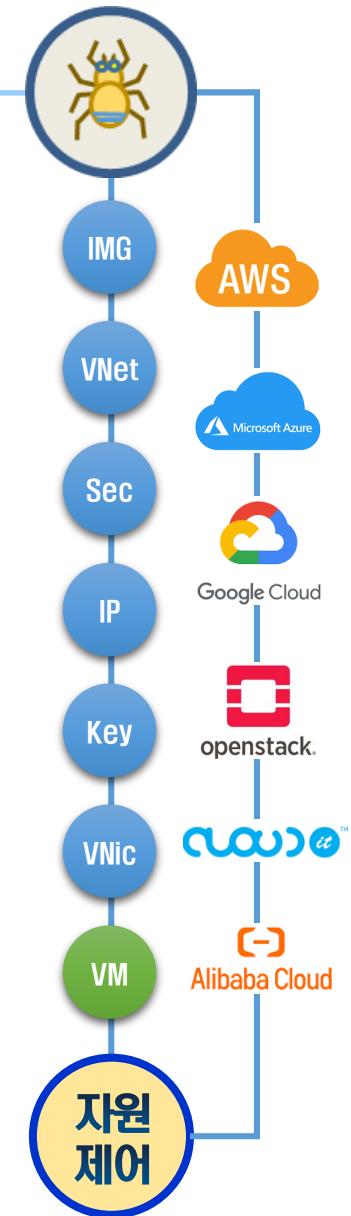
(1) 클라우드 연동 정보 설정



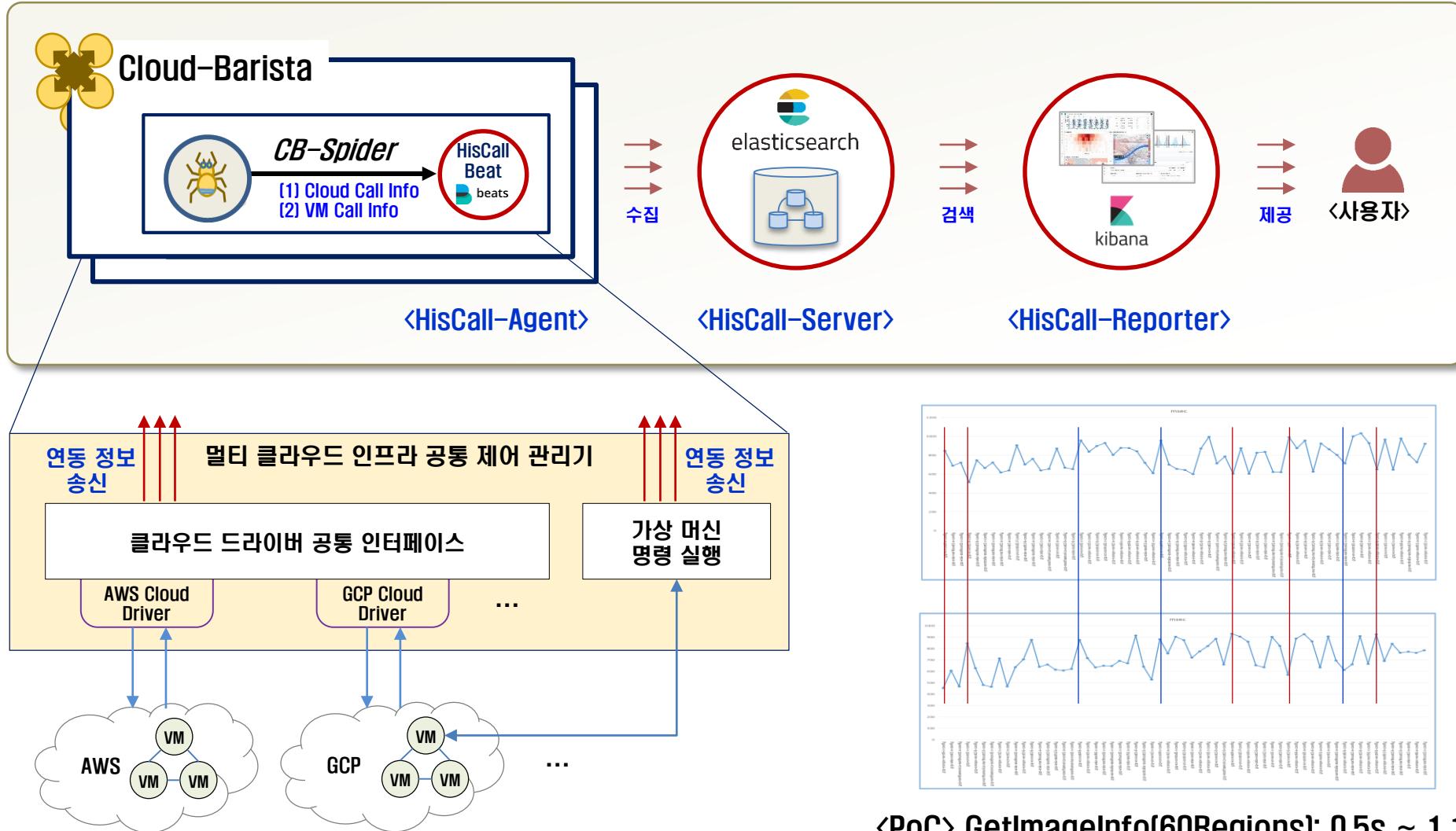
#연결설정 → #드라이버API → #드라이버 → #CSP



CB-Spider가 공통 제어하는 클라우드 자원



CB-Spider가 제공하는 다른 기능은?



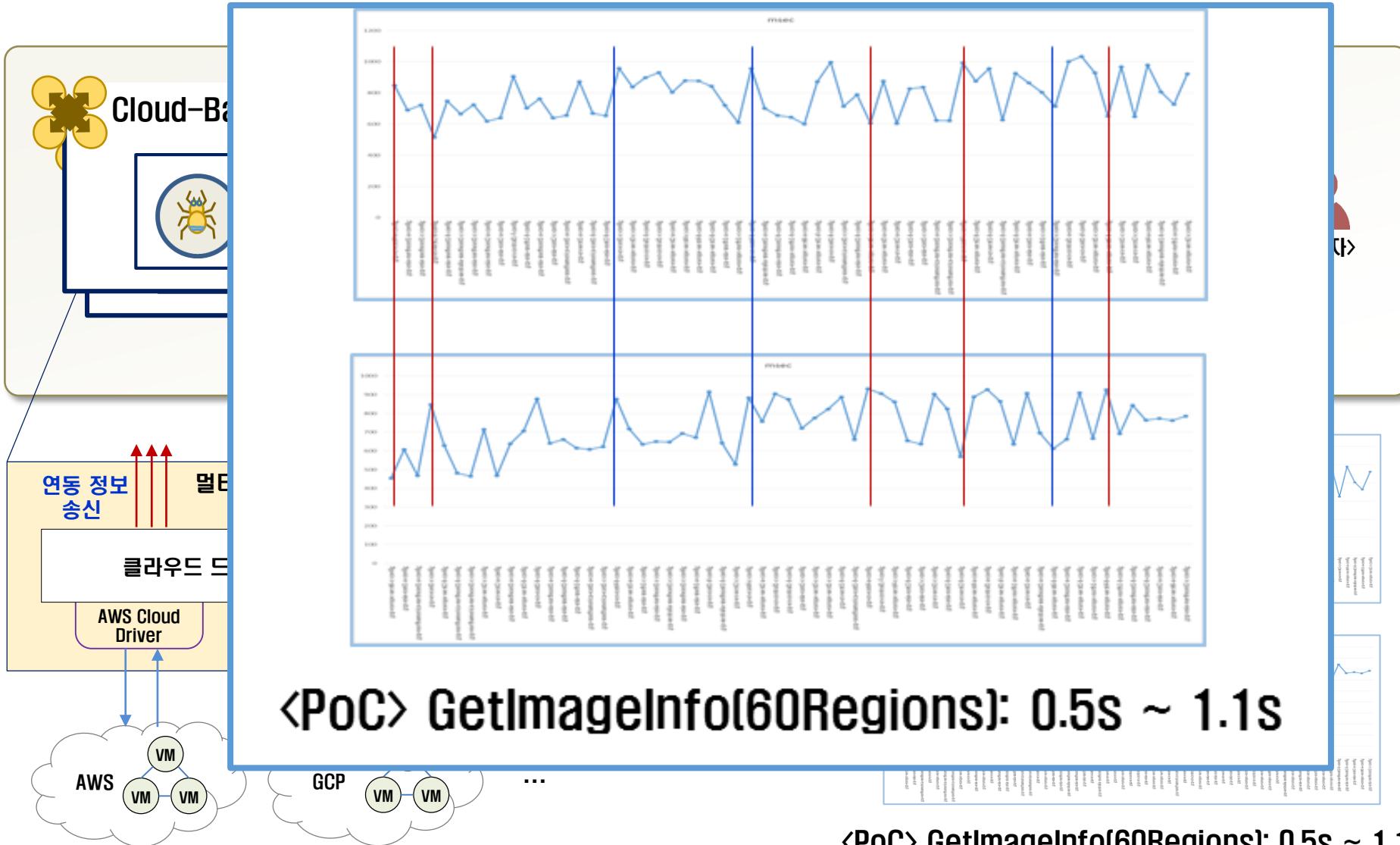
<활용 예시>

- 연동 품질 분석
- 스케줄링 반영 등
- 선수집 후활용

#HisCall
#HistoryOfCall
#Design
#OnlyPoC

<PoC> GetImageInfo(60Regions): 0.5s ~ 1.1s

CB-Spider가 제공하는 다른 기능은?

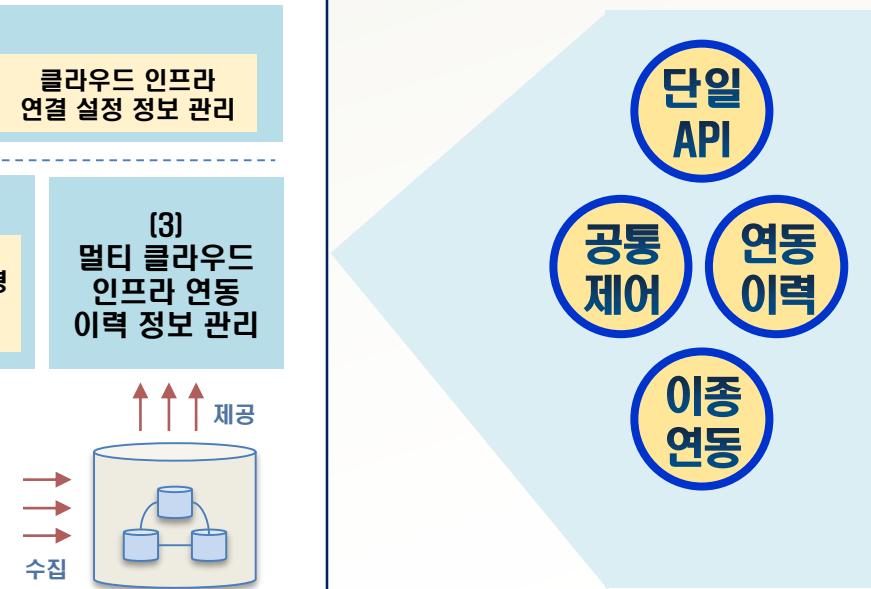
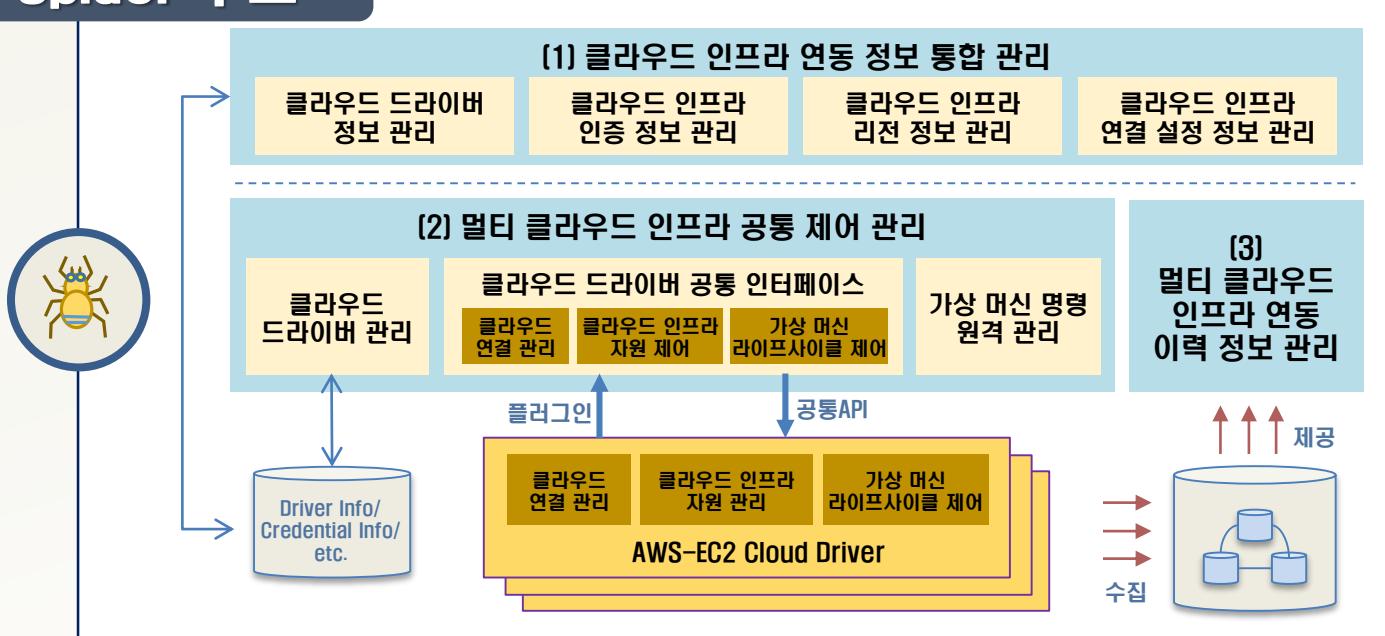


〈요약〉 CB-Spider 시스템 기능 및 구조

주요 기능

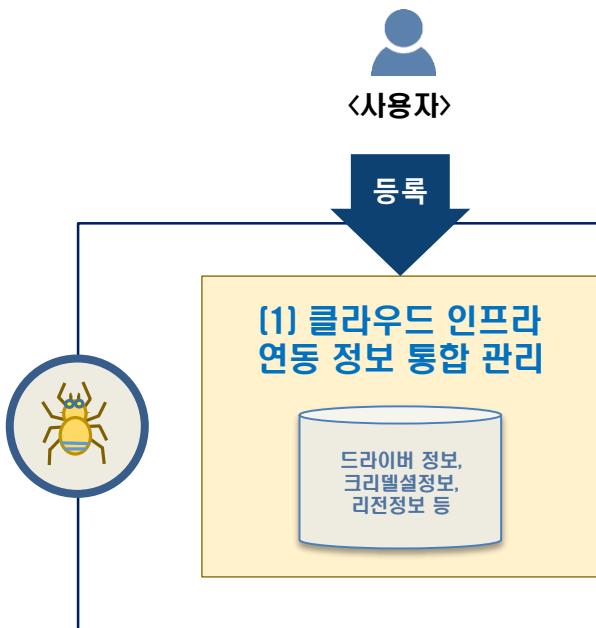
- 클라우드 인프라 연동 정보(드라이버/인증/리전 등) 추가/삭제/정보 제공
 - 클라우드 드라이버 라이브러리 동적 적재/제공 등 클라우드 드라이버 관리
 - 클라우드 인프라 자원 추가/삭제/정보제공 등 인프라 자원 관리: 이미지, VNet, 방화벽, 키페어 등
 - 가상 머신 실행/중지/재개/종료 등 실행 제어
 - 가상 머신 내부 명령 실행 및 파일 복사 등 가상 머신 명령 원격 실행
 - 클라우드 연결 및 기능 호출 등 이력 정보 로깅/정보 제공 등 클라우드 연결 이력 정보 관리 등

CB-Spider 구조



CB-Spider 서비스 개요

[1] 인프라 연동 정보 등록



[2] 인프라 자원/VM 생성 및 제어



[3] 인프라 연결/호출 정보 활용



#공통API #드라이버 #동적플러그인 #클라우드동적추가 #K8S연동 #이종멀티클라우드연동 #연동이력제공

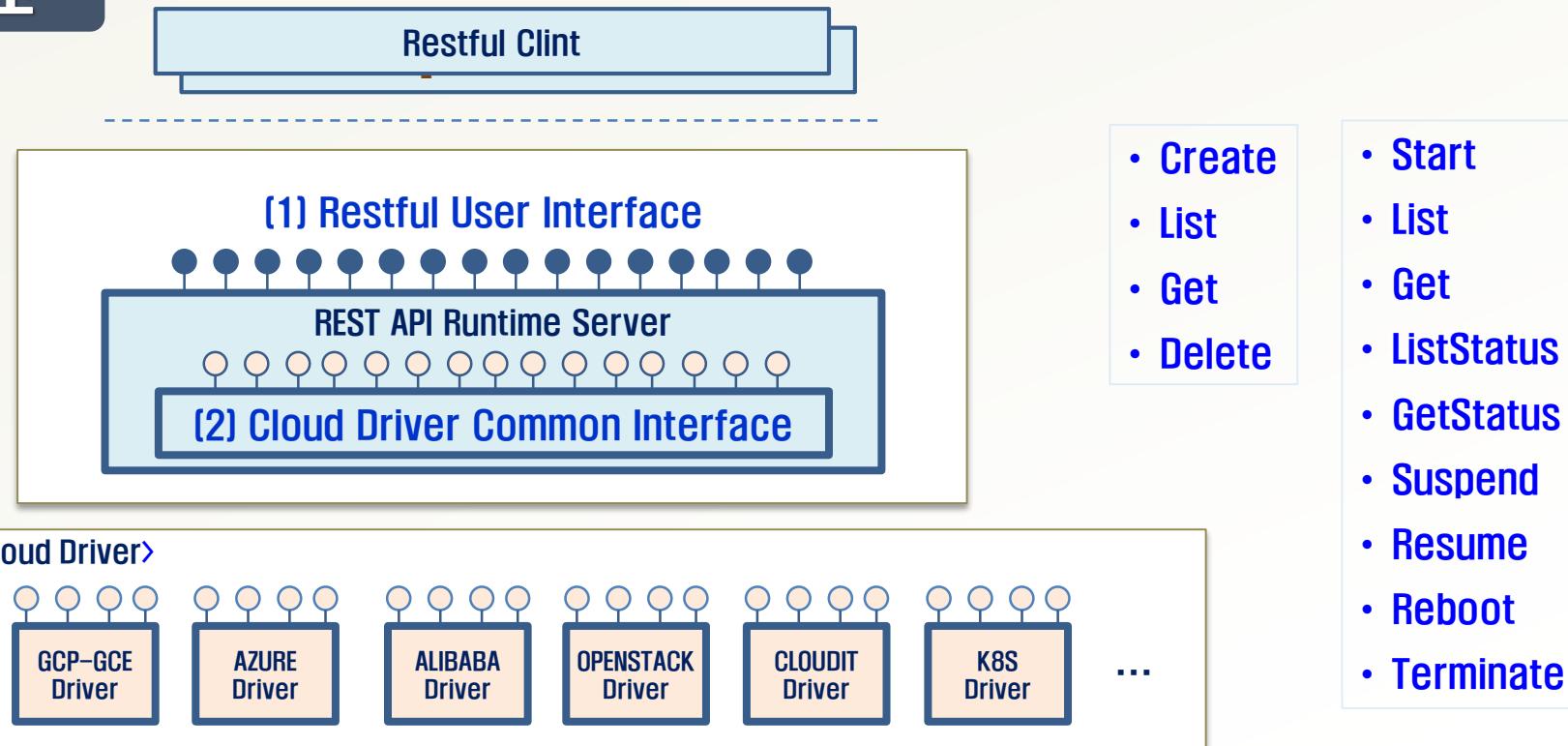
CB-Spider 서비스 활용 인터페이스



CB-Spider API

- [1] Framework User Interface: REST 기반 표준 인터페이스 제공
- [2] Cloud Driver Common Interface: Go 기반 드라이버 개발자 인터페이스 제공

인터페이스 구조





III. CB-Spider 개발 현황

CB-Spider 커뮤니티팀 개발 방법

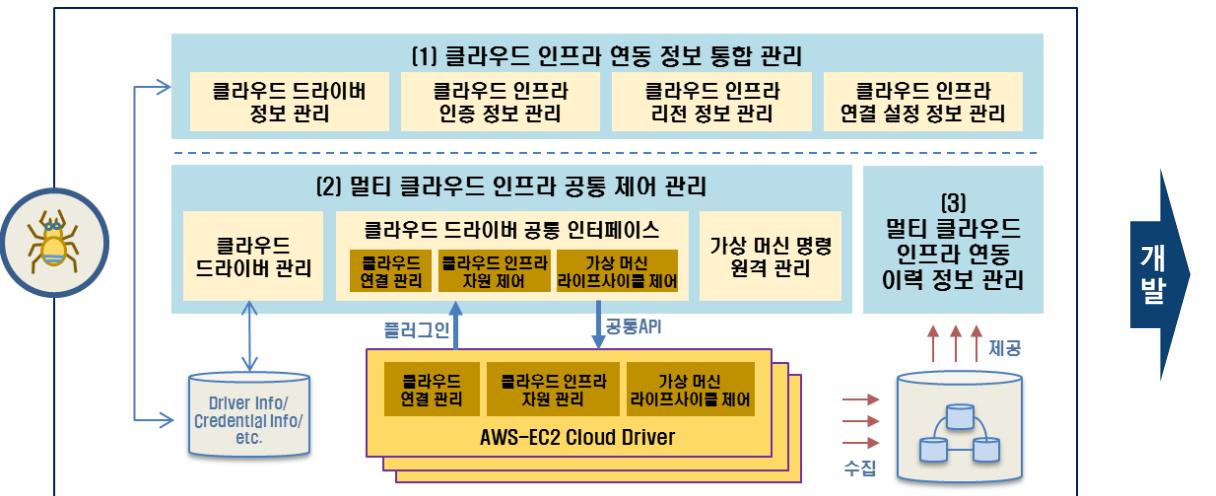


〈참조〉

- (1) 수정요청 <https://github.com/cloud-barista/poc-cb-spider/issues>
- (2) 반영요청: <https://github.com/cloud-barista/poc-cb-spider/pulls>
- (3) 협의요청: <https://cloud-barista.slack.com/messages/CLFCLNFTJ>

〈Thin&Fast 중심의 단계별 깃헙/슬랙 기반 공개 개발〉

CB-Spider 개발 현황 및 계획



[1] 클라우드 인프라 연동 정보 통합 관리

- 현황: API/Runtime 개발 및 통합 완료
- 계획: 동시성 등 개선 및 보완

[2] 멀티 클라우드 인프라 공통 제어 관리

- 현황: Driver REST API/Runtime 개발 통합 완료
- 현황: 가상 머신 명령 원격 관리 개발 통합 완료
- 현황: Driver Common Interface 개정 및 공유 완료
- 현황: 6 + 1종 드라이버 개발 추진
 - AWS/Azure/GCP: 드라이버 개발 및 통합 완료
 - OS/Cloudit: 드라이버 개발 완료, 통합 및 디버깅 중
 - Alibaba: 드라이버 개발 마무리 중
 - Cloud-Twin(K8S): PoC 개발 및 통합 완료
- 계획: Driver I/F 재검증 및 자원 통합 제어 방법 개선 등
 - 상반기: CSP 확장 전 I/F 및 통합 제어 방법 강화
 - 하반기: CSP 추가 연동
- 계획: K8S 드라이버 개발 및 연동

[3] 멀티 클라우드 인프라 연동 이력 정보 관리

- 현황: 설계 및 PoC 완료
- 계획: 2차년도 상반기 개발 추진 예정
- 계획: 연동 이력 정보 수집 및 아카이빙

#뼈대 #국외4종 국내1종 #공개1종 → #선개선후확장
 → #연동이력 #K8S



이종 CSP 단일API 정의 및 자원 통합 제어 개발 실전은...

The diagram illustrates the classification of ID & NameID & NameID sharing methods, and the management approach for CB-Spider ID.

Classification:

- Type-I:** Identity 제공 (Identity Provider)
- Type-II:** CSP (Cloud Service Provider) 제공 (NameID 연락처 제공)
- Type-III:** 사용자 제공 (NameID 연락처 제공)

Management Approach:

- CB-Spider : NameID 제공 추진** (Promote CB-Spider : NameID provision)
 - × 입력: NameID
 - × 제공: NameID, CSP 생성 ID(부가 정보로 제공)
 - × 명령: CSP의 Tag, Name, Description 등 부기 정보 활용

VM 상태 정보 및 CSP별 맵핑 관계 현황									* 2019.11.11. 현재	
Cloud	Barista	Starting	Running	Suspending	Suspended	Resuming	Rebooting	Terminating	Terminated	Failed
Alibaba	Pending	Running	Stopping	Stopped	Resuming [자체생성상태]	Rebooting [자체생성상태]	Terminating [자체생성상태] [OP: Stop&Delete]	Deleted	상태 정보를 얻을 수 있을 때 오늘에 의해 비정상 상태	
AWS	웹팅 필요요 [연결: Stopped 부팅 후 p-IP 무작 정리]	running	stopping	stopped	Resuming [자체생성상태]	Rebooting	shutting-down	terminated	상태 정보를 얻을 수 있을 때 오늘에 의해 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 있을 때 오늘에 의해 비정상 상태	
Azure [PowerState /Provisioning State]	starting /-	/running /succeeded	stopping /-	/stopped /succeeded	Resuming [자체생성상태]	Rebooting [자체생성상태] [OP: Stop&Start]	deallocating /-	[예외]NotExist	-	
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	


**CLOUD
BARISTA**

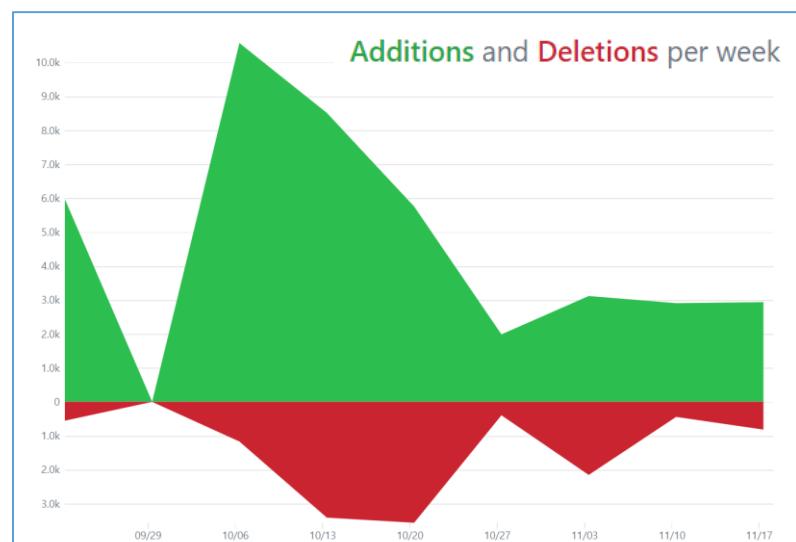
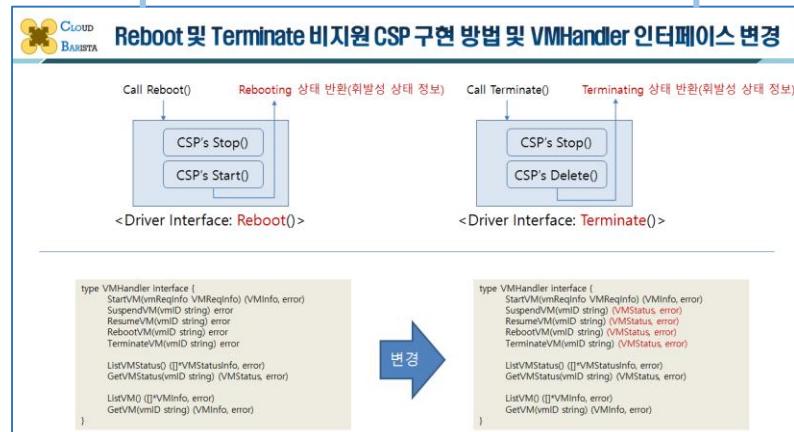
CSP별 Vnic 관리 현황 (Go SDK 기준)

* 2019.11.08. 현재

Cloud-Barista	<ul style="list-style-type: none"> VM launch 시 Vnic 자동 생성 	비고
Alibaba	<ul style="list-style-type: none"> VM launch 시 설정이 없으면 Vnic 자동 생성됨 Vnic 생성 제공: Secondary Vnic 추가 가능 	
AWS	<ul style="list-style-type: none"> VM launch 시 Vnic 자동 생성 Vnic 생성 제공: Secondary Vnic 추가 가능 	
GCP	<ul style="list-style-type: none"> VM launch 시 자동 생성 Vnic 생성 미제공 	
Azure	<ul style="list-style-type: none"> Vnic 생성 필수 Vnic 생성 후 VM launch 가능 → 드라이브: VM launch 시 Vnic 자동 생성, Terminate 시 자동 삭제 	
OpenStack	<ul style="list-style-type: none"> VM launch 시 자동 생성 Vnic 생성 제공: Secondary Vnic(Port) 추가 가능 	
Cloudit	<ul style="list-style-type: none"> VM launch 시 자동 생성 Vnic 생성 제공: Secondary Vnic(Port) 추가 가능 Vnic 생성 시 VMID 양도 필요 → Vnic 생성 인터페이스 수정 또는 다른 방안 검토 필요 	

ETRI | INNOVATION MEASURES CLOUD KACI

자원별 NameID 제공 방안 – 1/2				* 2019.11.13. 현재																									
	VNetwork	KeyPair	PublicIP																										
Alibaba	Create(NameID) NameID Get(NameID) NameID <ul style="list-style-type: none"> 구현형: Name 구현형: Name / ID 기능부수: 기본 반영방법: VPC ID 및 Subnet ID로 대체 NameID 	Create(NameID) NameID Get(NameID) NameID <ul style="list-style-type: none"> 구현형: Name 특이사항: Name-ID 유통 - Tag 사용 	Create(NameID) NameID Get(NameID) NameID <ul style="list-style-type: none"> 구현형: Name / ID 기능부수: 기본(보통) 반영방법: Name-ID 유통 - Tag 사용 																										
AWS	자원별 NameID 제공 방안 – 2/2																												
GCP																													
Azure	Security Group <table border="1"> <thead> <tr> <th></th><th>Create(NameID) NameID, ID</th><th>Get(NameID) NameID, ID</th></tr> </thead> <tbody> <tr> <td>Alibaba</td><td> <ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 </td><td> <ul style="list-style-type: none"> 구현형: ID 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 </td></tr> <tr> <td>AWS</td><td> <ul style="list-style-type: none"> 구현형: Name 특이사항: - GroupNameID Name-ID로 표기 - GroupID Tag로 표기 </td><td> <ul style="list-style-type: none"> 기본형: ID 기능부수: 기본 반영방법: 둘다 기반 GroupName 조합 </td></tr> <tr> <td>GCP</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 </td><td> <ul style="list-style-type: none"> 구현형: NameID </td><td></td></tr> <tr> <td>Azure</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 Azure SourcePortRange.ToPortRange로 구현 SourcePortRange.FromPortRange로 구현 </td><td> <ul style="list-style-type: none"> 구현형: NameID </td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 </td></tr> <tr> <td>OpenStack</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td><td></td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 </td></tr> <tr> <td>Cloudit</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td><td></td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td></tr> </tbody> </table>					Create(NameID) NameID, ID	Get(NameID) NameID, ID	Alibaba	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 	<ul style="list-style-type: none"> 구현형: ID 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 	AWS	<ul style="list-style-type: none"> 구현형: Name 특이사항: - GroupNameID Name-ID로 표기 - GroupID Tag로 표기 	<ul style="list-style-type: none"> 기본형: ID 기능부수: 기본 반영방법: 둘다 기반 GroupName 조합 	GCP	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 	<ul style="list-style-type: none"> 구현형: NameID 		Azure	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 Azure SourcePortRange.ToPortRange로 구현 SourcePortRange.FromPortRange로 구현 	<ul style="list-style-type: none"> 구현형: NameID 	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 	OpenStack	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 	Cloudit	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소
	Create(NameID) NameID, ID	Get(NameID) NameID, ID																											
Alibaba	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 	<ul style="list-style-type: none"> 구현형: ID 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 																											
AWS	<ul style="list-style-type: none"> 구현형: Name 특이사항: - GroupNameID Name-ID로 표기 - GroupID Tag로 표기 	<ul style="list-style-type: none"> 기본형: ID 기능부수: 기본 반영방법: 둘다 기반 GroupName 조합 																											
GCP	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 	<ul style="list-style-type: none"> 구현형: NameID 																											
Azure	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 Azure SourcePortRange.ToPortRange로 구현 SourcePortRange.FromPortRange로 구현 	<ul style="list-style-type: none"> 구현형: NameID 	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 																										
OpenStack	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 																										
Cloudit	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 																										
	VM <table border="1"> <thead> <tr> <th></th><th>VM(NameID) NameID, ID</th><th>Get(NameID) NameID, ID</th></tr> </thead> <tbody> <tr> <td>Alibaba</td><td> <ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 </td><td> <ul style="list-style-type: none"> 구현형: Name / ID 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 </td></tr> <tr> <td>AWS</td><td> <ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 반영방법: Tag로 Name 정보 설정 </td><td> <ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 </td></tr> <tr> <td>GCP</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 </td><td> <ul style="list-style-type: none"> 구현형: NameID </td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 </td></tr> <tr> <td>Azure</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td><td> <ul style="list-style-type: none"> 구현형: NameID </td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 조합 후 Name-주소 </td></tr> <tr> <td>OpenStack</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td><td></td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 조합 후 Name-주소 </td></tr> <tr> <td>Cloudit</td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td><td></td><td> <ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 </td></tr> </tbody> </table>					VM(NameID) NameID, ID	Get(NameID) NameID, ID	Alibaba	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 	<ul style="list-style-type: none"> 구현형: Name / ID 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 	AWS	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 반영방법: Tag로 Name 정보 설정 	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 	GCP	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 	<ul style="list-style-type: none"> 구현형: NameID 	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 	Azure	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 	<ul style="list-style-type: none"> 구현형: NameID 	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 조합 후 Name-주소 	OpenStack	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 조합 후 Name-주소 	Cloudit	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소
	VM(NameID) NameID, ID	Get(NameID) NameID, ID																											
Alibaba	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 	<ul style="list-style-type: none"> 구현형: Name / ID 기능부수: 기본 Name-ID로 상호식별 - TSV 지원 																											
AWS	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 반영방법: Tag로 Name 정보 설정 	<ul style="list-style-type: none"> 구현형: Name 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 																											
GCP	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 	<ul style="list-style-type: none"> 구현형: NameID 	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 기본 Tag로 표기 																										
Azure	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 	<ul style="list-style-type: none"> 구현형: NameID 	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 조합 후 Name-주소 																										
OpenStack	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 둘다 조합 후 Name-주소 																										
Cloudit	<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 		<ul style="list-style-type: none"> 구현형: NameID 기능부수: 기본 반영방법: 조합 후 Name-주소 																										



#어려워 #공조필수 #Thin&Fast적합 #개선중



이종 CSP 단일API 정의 및 자원 통합 제어 개발 실전은...

CLOUD BARISTA

ID & Name & NameID

```

graph TD
    User -- "Name, NameID" --> CSP
    CSP -- "Name, NameID" --> Type
    subgraph Type
        direction TB
        AWS[AWS  
OpenStack  
Cloud]
        GCP[GCP]
    end

```

CLOUD BARISTA

VM 상태 정보 및 CSP별 맵핑 관계 현황

(1) Americano 버전 확정용: 필요시 향후 변경
(2) Stop&Start 등 자체생성상태란?: Reboot() 함수 호출 반환 값으로만 제공(취발성 상태), 이후 상태 제공 불가, 다음 페이지 참조

※ 2019.11.11. 현재

Cloud-Barista	Starting	Running	Suspending	Suspended	Resuming	Rebooting	Terminating	Terminated	Failed
Alibaba	Pending	Running	Stopping	Stopped	Resuming [자체생성상태]	Rebooting [자체생성상태]	Terminating [자체생성상태] (OP: Stop&Delete)	Deleted	상태 정보를 얻을 수 없을 때 오류에 의한 비정상 상태
AWS	맵핑 불필요 [현재: StartVM 내부 Running 상태 확인 후 P-IP 부탁 주 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 [자체생성상태] (OP: Stop&Start)	Terminating [자체생성상태] (OP: Delete)	[예외]NotExist	상태 정보를 얻을 수 없을 때 오류에 의한 비정상 상태
Azure (PowerState / Provisioning State)	starting /-	running /succeeded	stopping /-	stopped /succeeded	Resuming [자체생성상태]	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

* 2019.11.08. 현재

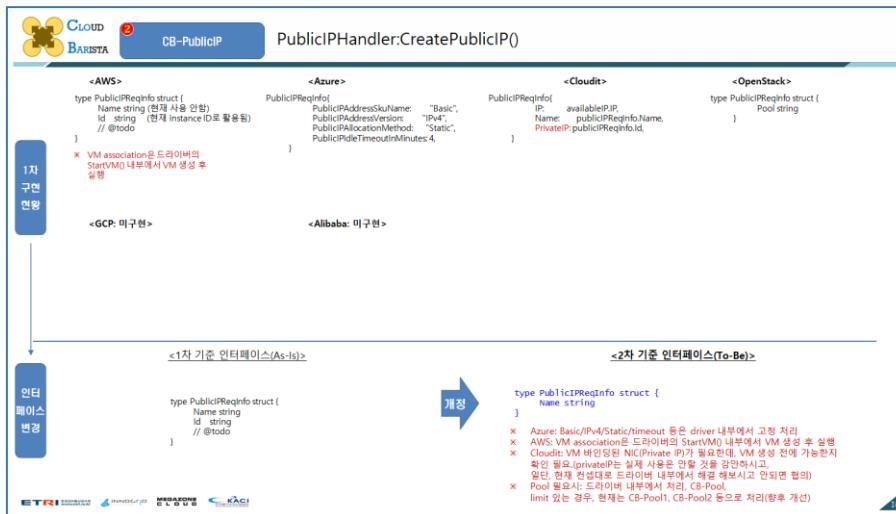
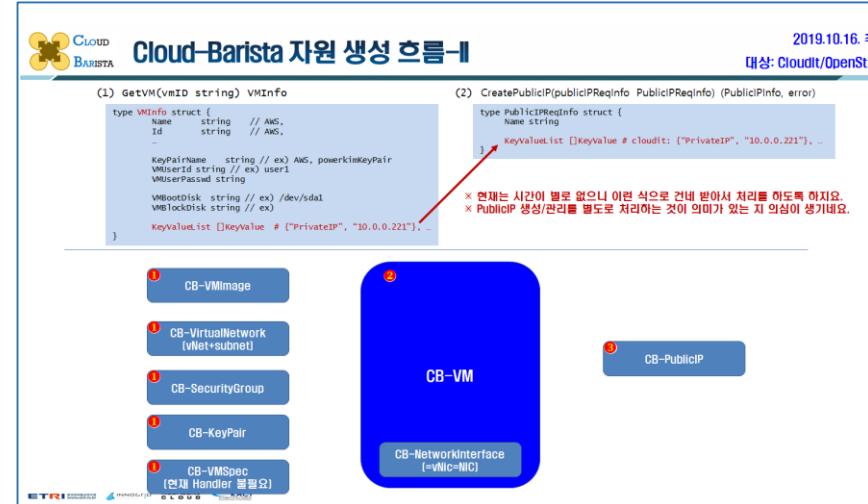
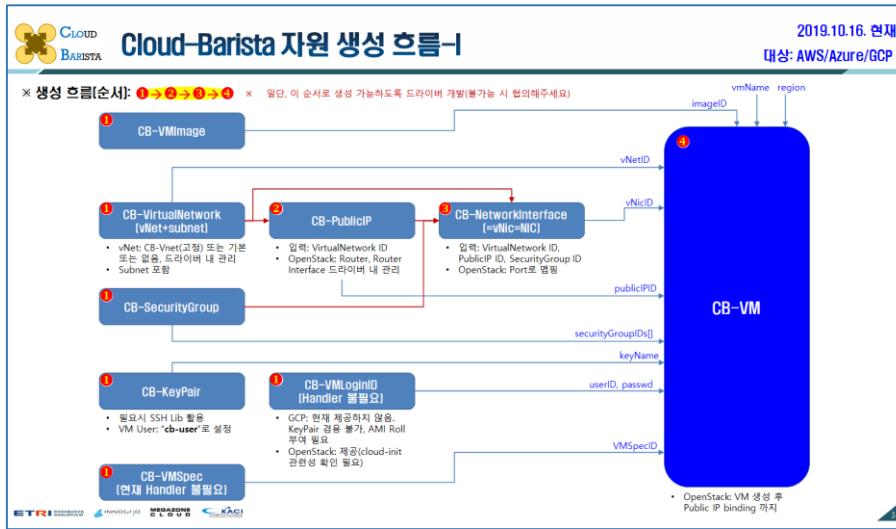
상	비고
면 VNIC 자동 생성됨 y Vnic 추가 가능	
상 y Vnic 추가 가능	
상 y Vnic 추가 가능	
상 y Vnic 자동 생성. Terminate 시 자동 삭제	
y VnicPort 추가 가능	
y VnicPort 추가 가능 정 또는 다른 방안 검토 필요	

s and Deletions per week

#어려워 #공조필수 #Thin&Fast적합 #개선중



이종 CSP 단일API 정의 및 자원 통합 제어 개발 실전은...



#어려워 #공조필수 #Thin&Fast적합 #개선중

CB-Spider 클라우드 드라이버 개발 및 통합 현황

*관리자원 : Image, VNet, Key, P-IP, Sec, Vnic, VM

* 1차: VM 내부관리

Americano 행사

드라이버 단위시험 (INNO/MEGA) → 드라이버 REST Test (INNO/MEGA) → Multi Region REST Test (ETRI) → Multi Resource REST Test (향후)

AWS



Azure



GCP



Cloudit



OpenStack



Alibaba

* 개발/시험 마무리 중
 * Publicip 관리는 가능
 * 제공정보: IP, ID만 제공, * publicIP 내부관리
 Tagging 불가

Twin(K8S)

* PoC for Cappuccino



<범례>

시험완료

- VM 생성/접속
- 향후 점검 필요

시험대기

보완필요

Private
Image

CSP
미지원
방안 필요

방안 필요

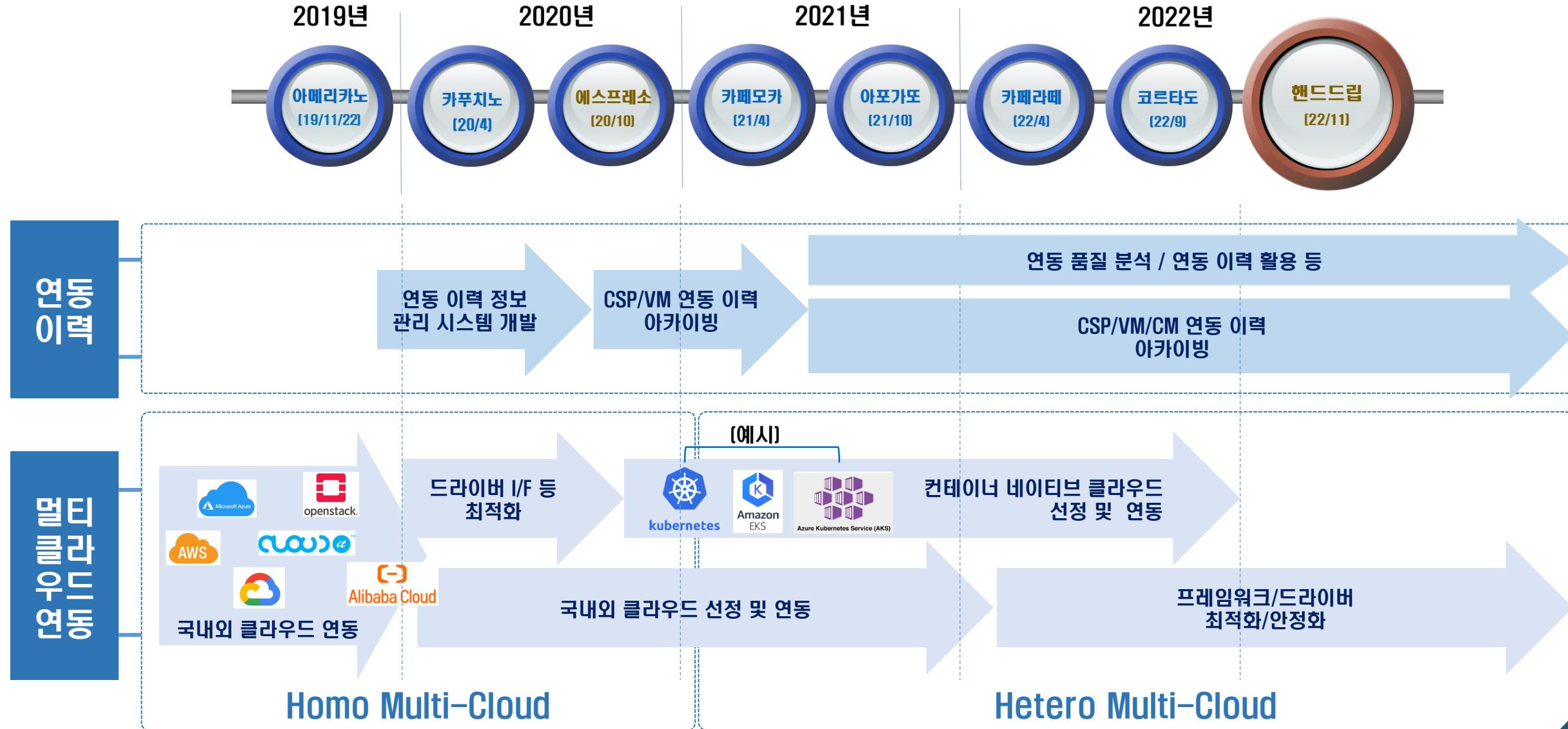
<이슈>

- * 자원 생성 흐름 차이 고려
- * Vnic 추가 지원 고려
- * VM User ID 관리 방법 개선 필요
- * Vnet 입력 Name
 - Case-I: 사용하지 않음.
 - Case-II: subnetwork 이름으로 활용
- ...

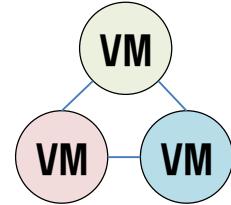


IV. CB-Spider 개발 로드맵

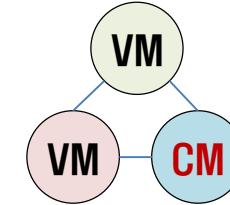
CB-Spider 개발 로드맵



Cappuccino에선 어떤 클라우드를 엮지?



Homo Multi-IaaS



Hetero Multi-IaaS

* 이종 멀티 클라우드 환경을 요구하는 멀티 계층 App 출현 → Hetero-Multi IaaS 실행 환경 제공 시급 ➡



...



kubernetes

CB-Spider 상세 규격 및 설치 가이드

규격 및 설계 등

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

사용자 인터페이스 (REST)

- 인프라 연동 정보 관리: <https://documenter.getpostman.com/view/9027676/SVzz4fb4>
- 인프라 공통 제어 관리: <https://documenter.getpostman.com/view/9027676/SVtSXpZE>

드라이버 인터페이스 (Go)

- <https://github.com/cloud-barista/cb-spider/tree/master/cloud-control-manager/cloud-driver/interfaces>

소스 및 설치 가이드

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



V. CB-Spider 시연



시연 목적 : 최종 목표 가능성 제시 / 전세계 이종 클라우드 간보기

CB-Spider 최종 목표



- 전세계 이종 클라우드를 엑스포할 수 있게 한다.
- 전세계 클라우드를 하나처럼 쓸 수 있게 한다.



<Today>

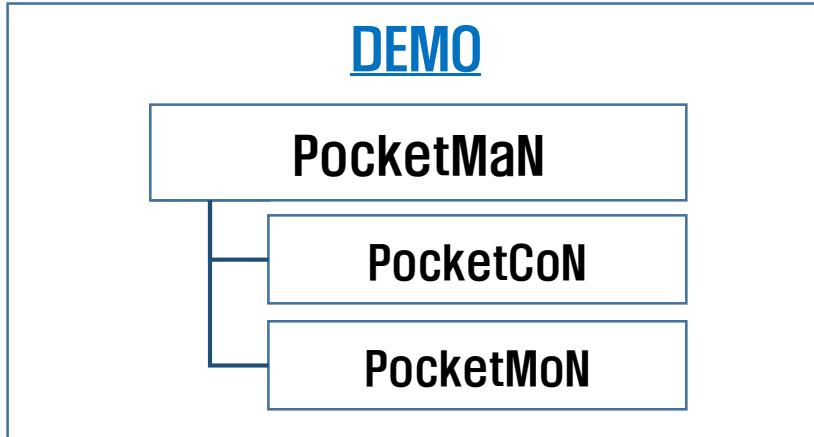


CB-Spider.
100+ Regions in your Berry.

연동 대상: 전세계 이종 클라우드 / 가상 클라우드



시연 환경 및 구성



----- REST API -----



#CB-Spider 간단 응용 #Pocket Manager for Multi-Cloud

#멀티클라우드 자원 통합 제어 데모

#멀티클라우드 자원 상태 확인 데모

#단일API #공통제어 #이종연동

#PM #VM #PI ...



#AnyClouds



시연 화면 구성 및 시나리오

```
root@node12: ~/go/src/github.com/cloud-barista/cb-spider/api-runtime/rest-runtime/log
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:06:41 CMRest.go:92, main.getImage() - aws-us-central-1-config : elapsed 2123
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-us-west-2-config : elapsed 198
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 236
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-eu-west-2-config : elapsed 321
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-sa-east-1-config : elapsed 340
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-eu-west-1-config : elapsed 364
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-sa-east-1-config : elapsed 371
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 393
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 238
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-eu-west-3-config : elapsed 299
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 332
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 344
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 412
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 414
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 237
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 301
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 334
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 349
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 367
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 379
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:18 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 232
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:18 CMRest.go:92, main.getImage() - aws-eu-west-3-config : elapsed 106
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:38 CMRest.go:92, main.getImage() - aws-eu-central-1-config : elapsed 317
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:38 CMRest.go:92, main.getImage() - aws-sa-east-1-config : elapsed 484
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:38 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 723
[CLOUD-BARISTA] [INFO]: 2019-11-19 20:04:26 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 825
[CLOUD-BARISTA] [INFO]: 2019-11-19 20:04:26 CMRest.go:92, main.getImage() - aws-eu-west-2-config : elapsed 867
[CLOUD-BARISTA] [INFO]: 2019-11-19 20:04:26 CMRest.go:92, main.getImage() - aws-us-west-2-config : elapsed 867
```

PocketCON

#connect → #create → #control

```
root@node12: ~/go/src/github.com/cloud-barista/cb-spider/api-runtime/rest-runtime/log
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:06:41 CMRest.go:92, main.getImage() - aws-us-central-1-config : elapsed 2123
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-us-west-2-config : elapsed 198
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 236
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-eu-west-3-config : elapsed 321
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-sa-east-1-config : elapsed 340
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-eu-west-1-config : elapsed 364
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-sa-east-1-config : elapsed 371
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:11 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 393
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 238
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-eu-west-1-config : elapsed 299
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 332
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 344
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 412
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 414
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 237
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 301
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 334
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 349
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 367
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:17 CMRest.go:92, main.getImage() - aws-us-west-1-config : elapsed 379
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:18 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 232
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:18 CMRest.go:92, main.getImage() - aws-eu-west-3-config : elapsed 306
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:38 CMRest.go:92, main.getImage() - aws-eu-central-1-config : elapsed 337
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:38 CMRest.go:92, main.getImage() - aws-sa-east-1-config : elapsed 444
[CLOUD-BARISTA] [INFO]: 2019-11-19 23:07:38 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 723
[CLOUD-BARISTA] [INFO]: 2019-11-19 20:04:26 CMRest.go:92, main.getImage() - aws-us-east-2-config : elapsed 825
[CLOUD-BARISTA] [INFO]: 2019-11-19 20:04:26 CMRest.go:92, main.getImage() - aws-eu-west-2-config : elapsed 867
```

PocketMON

#list #get #status

```
root@node12: ~/go/src/github.com/cloud-barista/cb-spider/api-runtime/rest-runtime
[CLOUD-BARISTA] [INFO]: 2019-11-20 00:23:35 VMHandler.go:474, github.com/cloud-barista/cb-spider/cloud-control-manager/cloud-driver/drivers/aws/resources.(*AwsVMHandler).GetVm() - 향고 웨일리 VM 정보 확인 : [1]
[CLOUD-BARISTA] [INFO]: 2019-11-20 00:23:35 VMHandler.go:502, github.com/cloud-barista/cb-spider/cloud-control-manager/cloud-driver/drivers/aws/resources.ExtractDescribeInstances() - EC2 정보 : [shutting-down]
[CLOUD-BARISTA] [INFO]: 2019-11-20 00:23:35 VMHandler.go:535, github.com/cloud-barista/cb-spider/cloud-control-manager/cloud-driver/drivers/aws/resources.ExtractDescribeInstances() - ===> BlockDeviceMapper - trueof : [
    DeviceName: "/dev/sda1",
    Ebs: {
        AttachTime: 2019-11-19 15:20:09 +0000
        DeleteOnTermination: true,
        Status: "attached",
        VolumeId: "vol-058865ec56b193e4e"
    }
]
[CLOUD-BARISTA] [INFO]: 2019-11-20 00:23:35 VMHandler.go:488, github.com/cloud-barista/cb-spider/cloud-control-manager/cloud-driver/drivers/aws/resources.(*AwsVMHandler).GetVm() - vmInfo{vm-powerkim01 i-06110d3050501c93d 0001-01-01 00:00:00 +0000 UTC { us-west-2d} ami-0347ed3d05fc5b06 t3.micro vpc-0e98deadb99fec3d3 [sg-06d08244bed0d3349] eni-attach-059ffcc5c97dabcf4 44.228.160.31 192.168.210.208 ip-192-168-210-208.us-west-2.compute.internal mcb-keypair-powerkim /dev/sdal /dev/sda1 [State shutting-down] {Alef4ccdf25b} {KeyName mcb-keypair-powerkim}]
{"time":"2019-11-20T00:23:35.090977454+09-00","user_agent":"curl/7.35.0"} {"time": "2019-11-20T00:23:35.090977454+09-00", "user_agent": "curl/7.35.0", "connection_name": "aws-us-west-2-config", "bytes_in": 0, "bytes_out": 826}
```

CB-Spider 엔진

#단일API #공통제어 #이종연동

CSP 컨솔 : 120 Regions/Locations/Zones

100 VCloud



CB-Spider를 만들어 가는 사람들

<input type="checkbox"/>		ChoiNakSoo
<input type="checkbox"/>		Yoo Jae-Sung dev4unet
<input type="checkbox"/>		helloworldkorea
<input type="checkbox"/>		hyokyungk
<input type="checkbox"/>		B.T. Oh innodreamer
<input type="checkbox"/>		jaehong park iworkist
<input type="checkbox"/>		Jeongwoon

<input type="checkbox"/>		Jihoon Seo jihoon-seo
<input type="checkbox"/>		kyongminkwon
<input type="checkbox"/>		ByoungSeob Kim powerkimhub Maintainer
<input type="checkbox"/>		Seokho Son seokho-son Maintainer
<input type="checkbox"/>		sksok
<input type="checkbox"/>		zephy-Jeong zephy-mzc

감사합니다.

www.github.com/cloud-barista/cb-spider

(김병섭 / powerkim@etri.re.kr)



Cloud
BARISTA

Multi-cloud service common framework software
To maximize capability and utilization of Multi-cloud resources

What is Multi-Cloud

Cloud-Barista Research Team

Accessibility

Contact

Cloud-Barista drives you to Multi-Cloud

Collaborate to create Multi-Cloud world, together.