



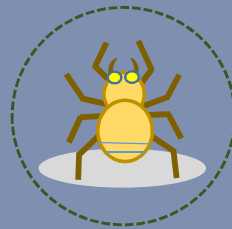
CLOUD

BARISTA

멀티 클라우드 서비스 공통 플랫폼

# CB-Spider : 전세계 클라우드를 엮는 그날까지

(멀티 클라우드 인프라 연동)



김 병 섭 / CB-Spider 프레임워크 리더

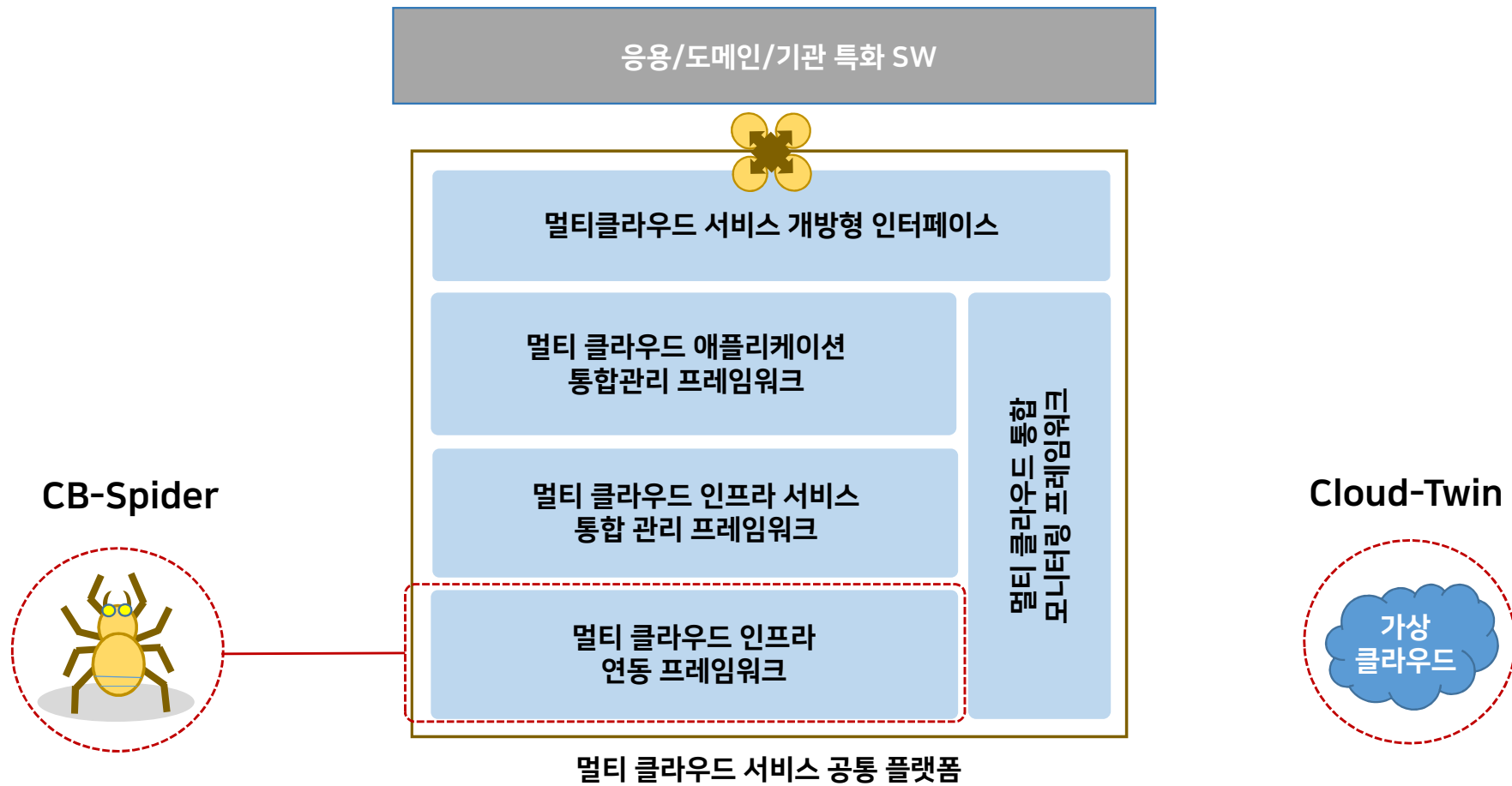
## “Contact to the Multi-Cloud”

클라우드 바리스타들의 두 번째 이야기

### Cloud-Barista Community 2<sup>nd</sup> Open Conference



## 이번 세션은...





## 목 차

---

**I**

CB-Spider 프레임워크 기술 개요

**II**

CB-Spider 프레임워크 주요 기술

**III**

CB-Spider 프레임워크 기술 개발 현황

**IV**

CB-Spider 프레임워크 기술 개발 로드맵

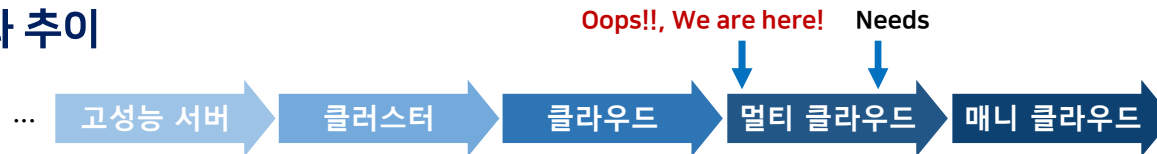
**V**

CB-Spider 프레임워크 활용 및 이종 IaaS 관리 시연

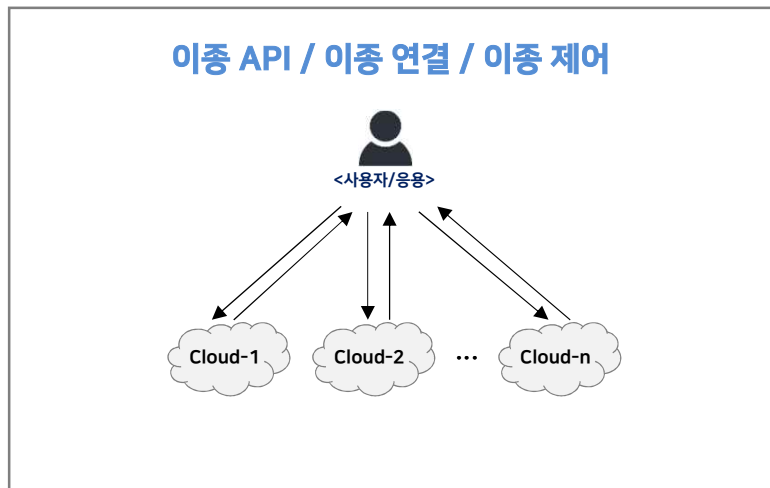


# CB-Spider : 멀티 클라우드 인프라 연동 기술의 개요 및 필요성

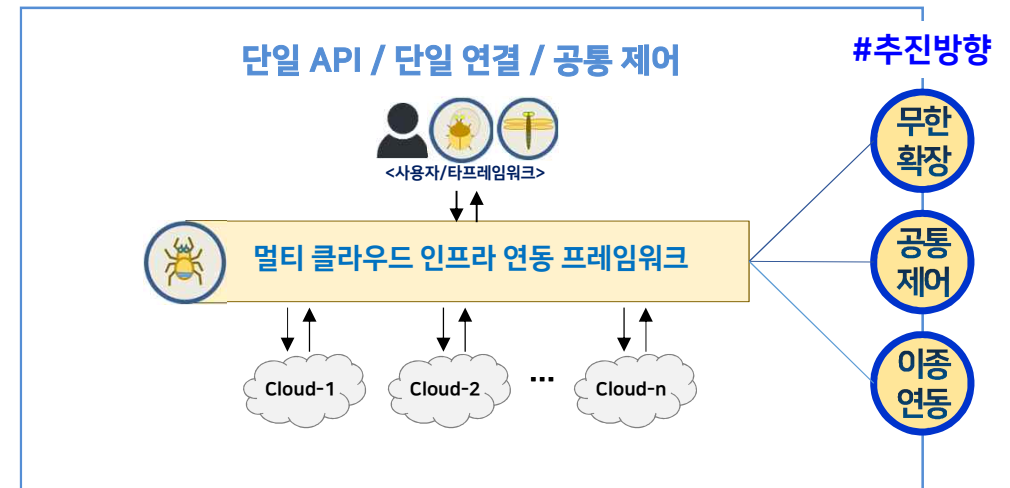
## 컴퓨팅 인프라 변화 추이



- CB-Spider: 플러그인-드라이버 기반의 클라우드 연동 확장성을 제공하는 멀티 클라우드 인프라 연동 프레임워크
- AS-IS: 신규 클라우드 연동 → 클라우드 활용 분석 및 추가 개발 필요
- TO-BE: 신규 클라우드 연동 → 클라우드 등록 및 활용



<기존 멀티 클라우드 서비스 사용자 환경>



<멀티 클라우드 인프라 연동 프레임워크 사용자 환경>



## CB-Spider의 최종 목표는?



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

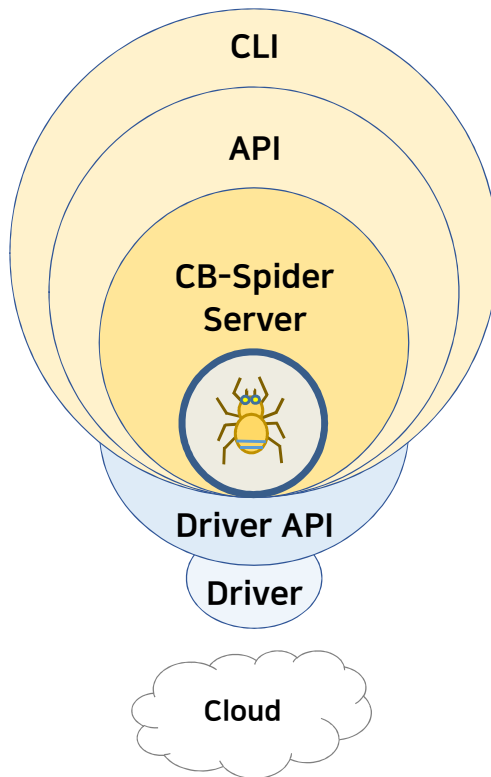
무한  
확장

공통  
제어

이중  
연동



# CB-Spider 구성 및 주요 기능



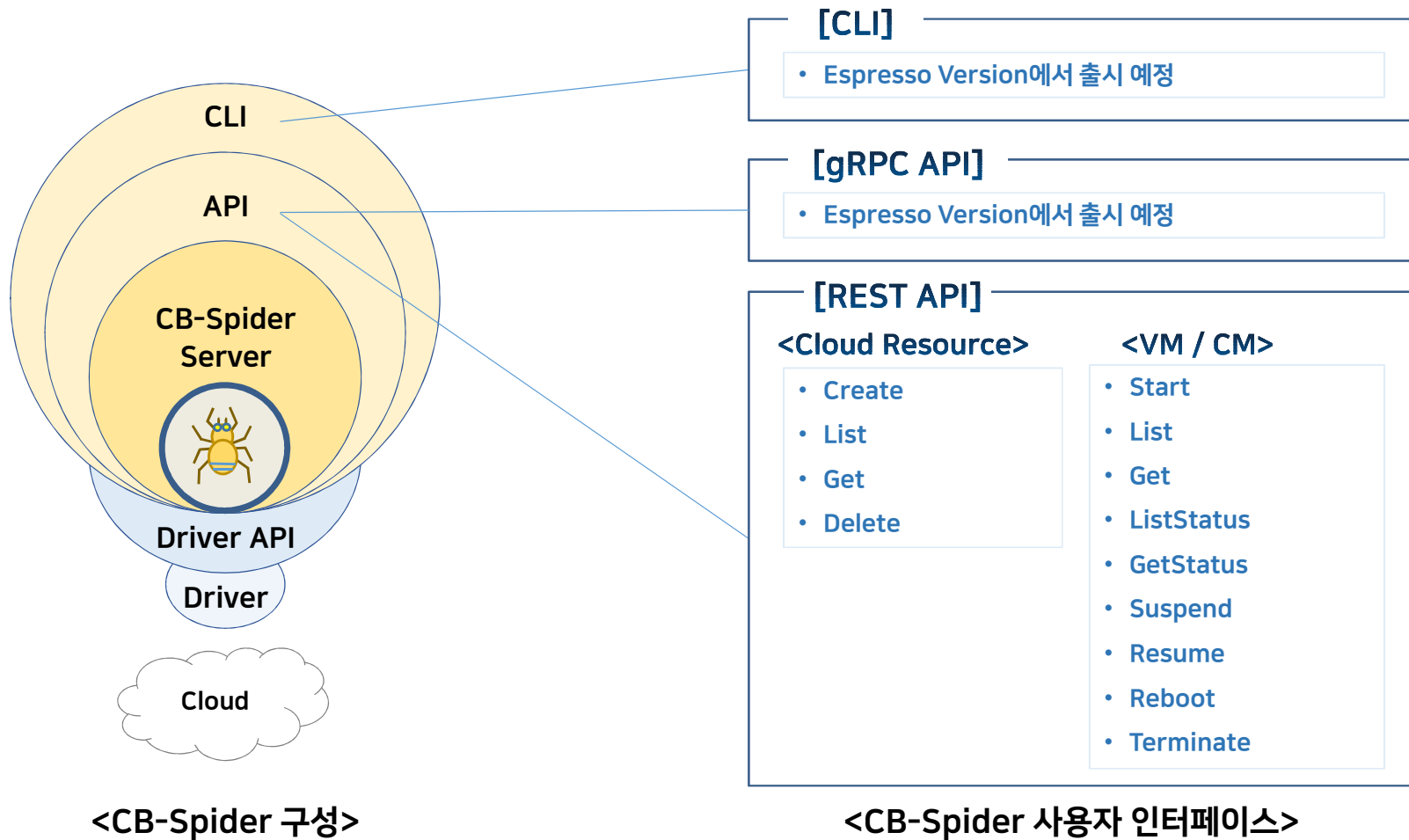
<CB-Spider 구성>

- 터미널 사용자를 위한 Command-Line 인터페이스 제공
- REST 및 gRPC 규격의 프로그래밍 인터페이스 제공
  - 멀티 클라우드 호출 이력 정보 수집/분석
  - 멀티 클라우드 컴퓨팅 인프라 자원/VM/CM 라이프사이클 관리
  - 멀티 클라우드 쉬운 연결 제공
  - 멀티 클라우드 연동 동적 확장
- 클라우드 드라이버 플러그인 방식
- 클라우드 드라이버 공통 인터페이스
- 7종 드라이버 제공(2020.05. 현재 - AWS, GCP, Azure, Alibaba, Cloudit, OpenStack, Docker)
- 대상: 이중 멀티 클라우드 컴퓨팅 인프라 자원

<CB-Spider 주요 기능>

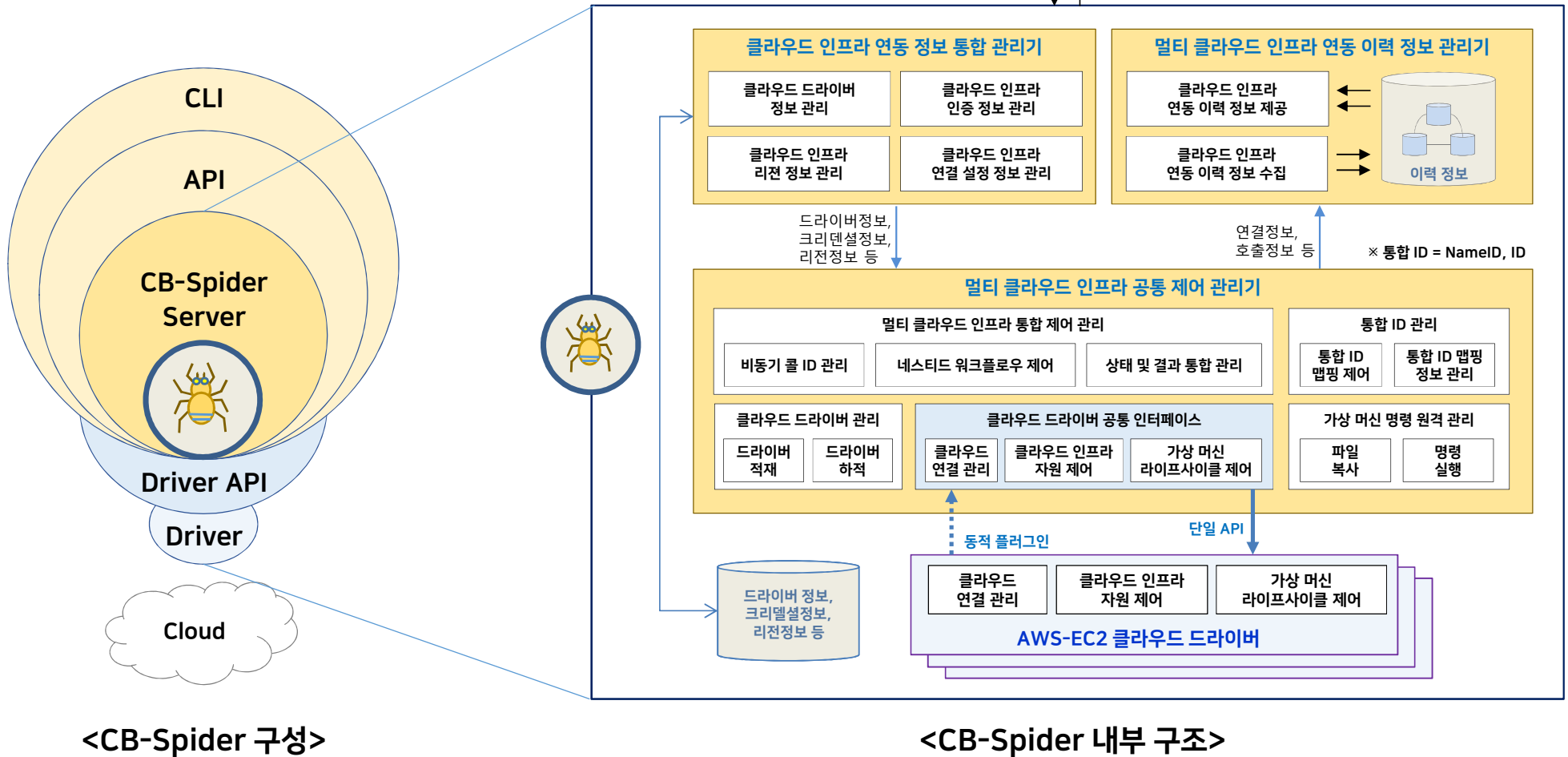


# CB-Spider가 제공하는 사용자 인터페이스





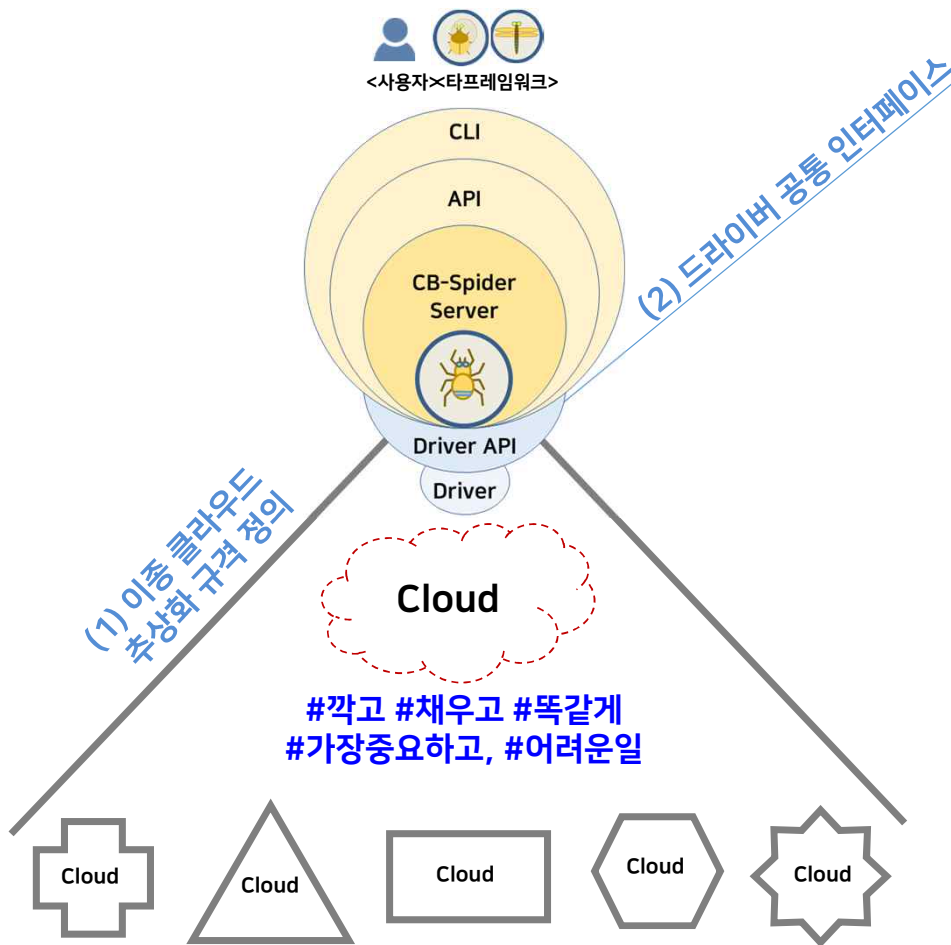
# CB-Spider의 내부 구조







# 멀티 클라우드를 엮는 기술 - 추상화/공통인터페이스



## [발췌: Cloud Driver Common API 형식]

```
type CloudDriver interface {
    GetDriverVersion() string
    GetDriverCapability() DriverCapabilityInfo
    ConnectCloud(connectionInfo ConnectionInfo) (icon.CloudConnection, error)
}
```

```
type CloudConnection interface {
    CreateImageHandler() (irs.ImageHandler, error)
    CreateVPCHandler() (irs.VPCHandler, error)
    CreateSecurityHandler() (irs.SecurityHandler, error)
    CreateKeyPairHandler() (irs.KeyPairHandler, error)
    CreateVMHandler() (irs.VMHandler, error)
    CreateVMSpecHandler() (irs.VMSpecHandler, error)
    IsConnected() (bool, error)
    Close() error
}
```

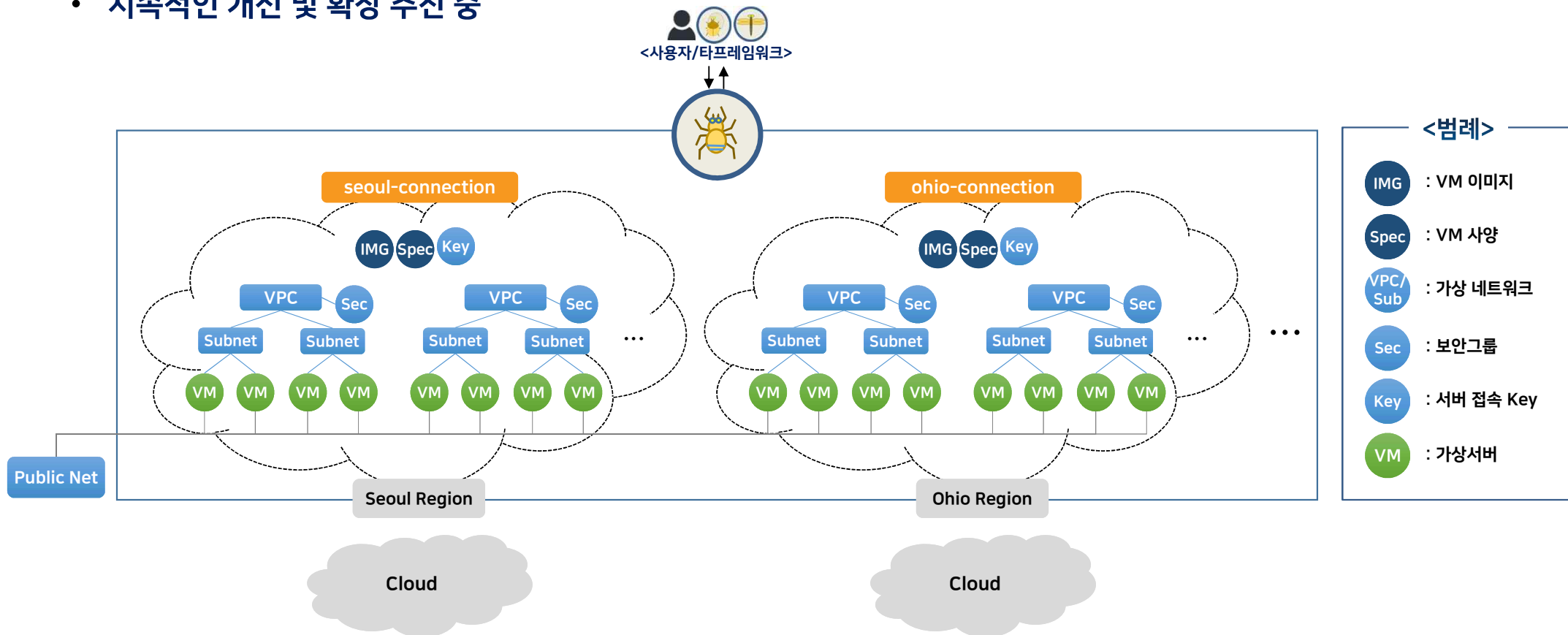
```
type VPCHandler interface {
    CreateVPC(vpcReqInfo VPCReqInfo) (VPCInfo, error)
    ListVPC() ([]*VPCInfo, error)
    GetVPC(vpcIID IID) (VPCInfo, error)
    DeleteVPC(vpcIID IID) (bool, error)
}
```

```
type VMHandler interface {
    StartVM(vmReqInfo VMReqInfo) (VMInfo, error)
    SuspendVM(vmiIID IID) (VMStatus, error)
    ResumeVM(vmiIID IID) (VMStatus, error)
    RebootVM(vmiIID IID) (VMStatus, error)
    TerminateVM(vmiIID IID) (VMStatus, error)
    ListVMStatus() ([]*VMStatusInfo, error)
    GetVMStatus(vmiIID IID) (VMStatus, error)
    ListVM() ([]*VMInfo, error)
    GetVM(vmiIID IID) (VMInfo, error)
}
```



## 멀티 클라우드를 엮는 기술 - 추상화된 통합 자원

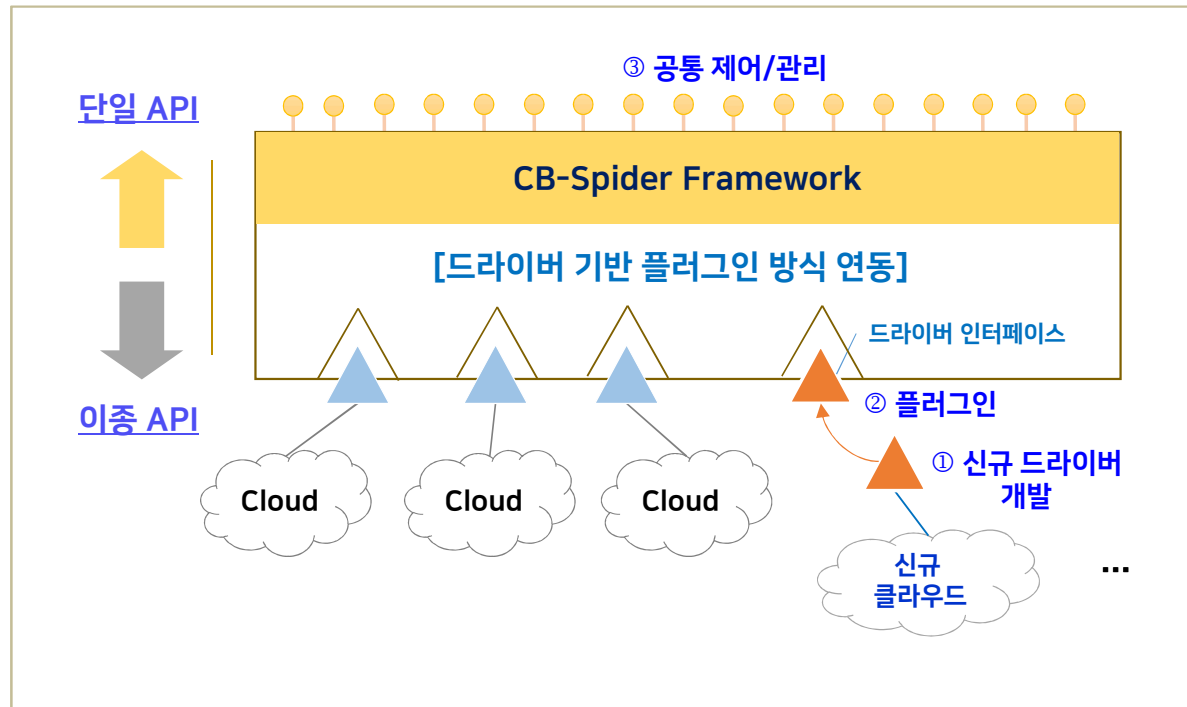
- 멀티 클라우드 자원 통합 구성 가능 형상: 연결 설정을 중심으로 자원 독립 구성
- 지속적인 개선 및 확장 추진 중



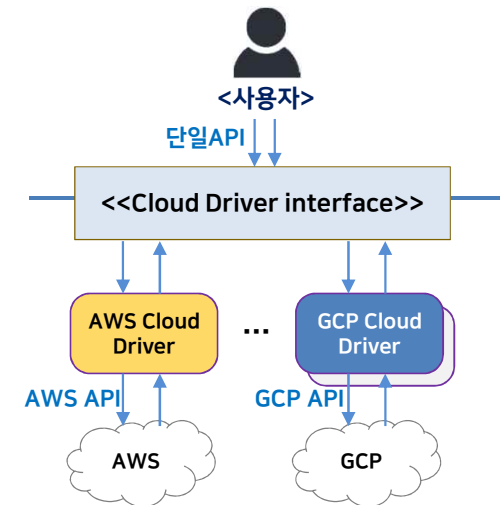


# 멀티 클라우드를 엮는 기술 - 드라이버/플러그인

- 플러그인 기반 멀티 클라우드 인프라 연동 및 공통 제어 방식



<클라우드 드라이버/플러그인 방식 연동 개념>



#Go Interface/Plugin

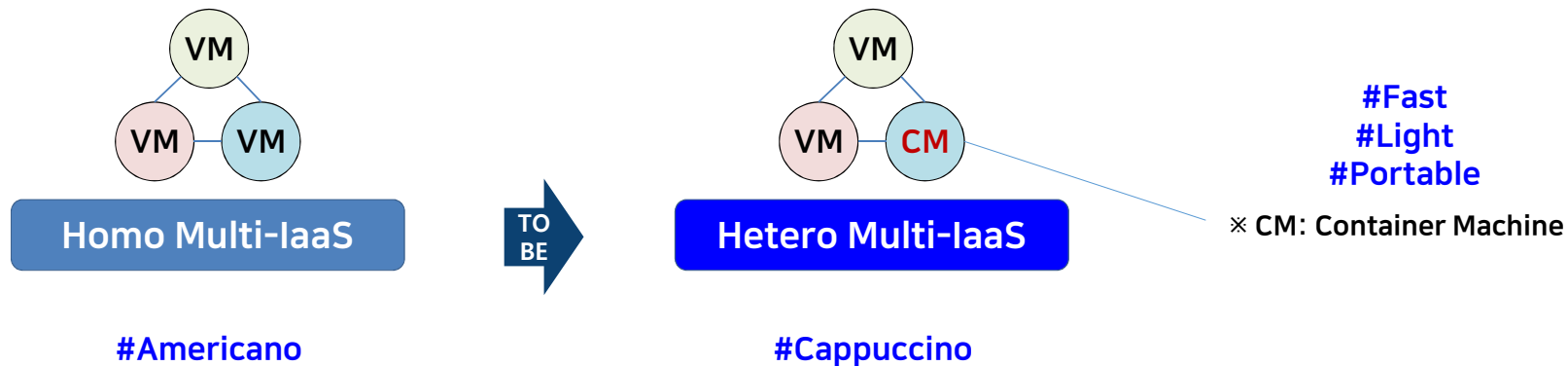
<드라이버/플러그인 개발 형상>



## 다양한 인프라 제공 기술 - Hetero Multi-IaaS

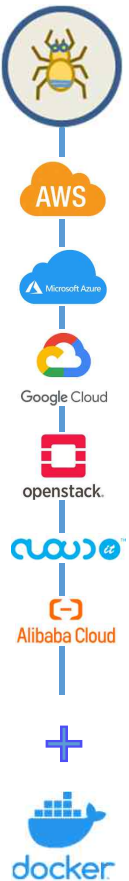
- 이종 멀티 클라우드 환경을 요구하는 멀티 계층 응용 증가  
→ VM과 함께 **시스템 컨테이너** 동시 제공 (Hetero Multi-IaaS)
- 카푸치노 연동 대상 : Docker ( )

※ Container분류  
- App Container  
- System Container



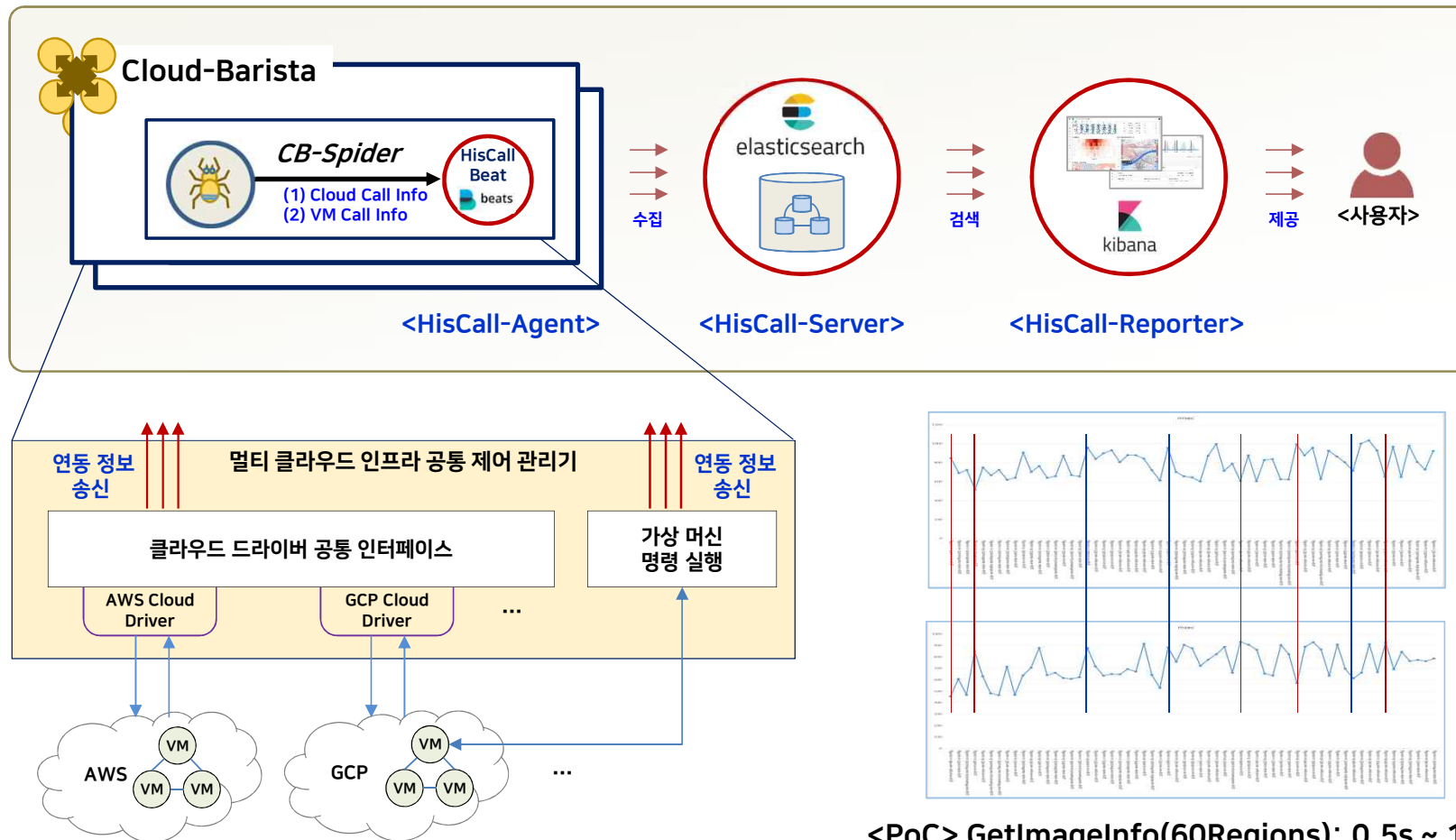
※ 시사점: Docker 설치 가능 환경 → Cloud-Barista Computing 인프라 제공 가능 환경

#물리노드 #가상노드 #라즈베리 #NAS #ETC





# 인프라 연동 이력 정보 제공 기술 - HisCall(History of Call)



## <활용 예시>

- 연동 품질 분석
- 스케줄링 반영 등
- 선수집 후활용

#HisCall  
#HistoryOfCall

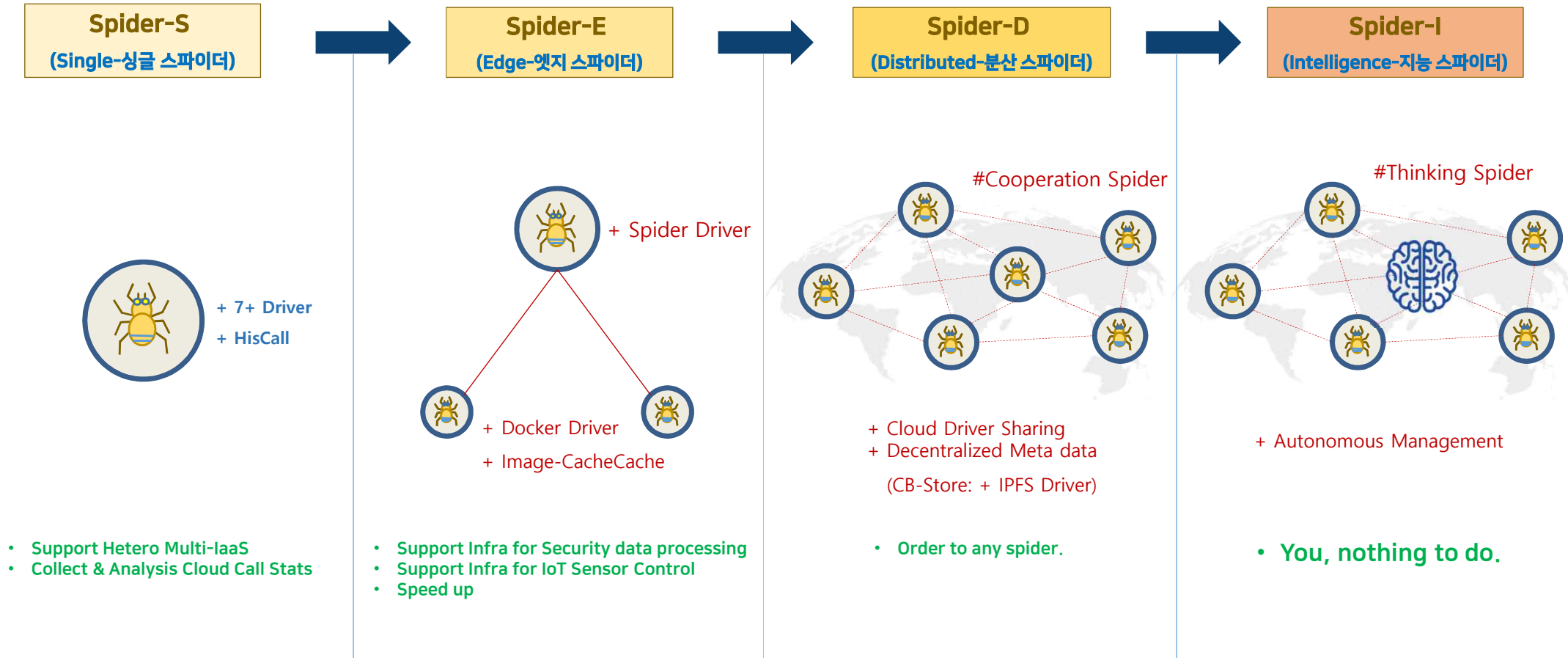
#Design  
#OnlyPoC

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



# 무한 확장 기술 방향 - 분산/지능

※ 커뮤니티 내 협의 중



※ 전 세계 클라우드를 엮는 그날까지...



# CB-Spider History 및 개발 현황

2019.03. ~

**farmoni**

- CB-Spider 전신
- Static 연동(하드코딩)
- 3종 클라우드 연동

farmoni



2019.06. ~

**poc-cb-spider**

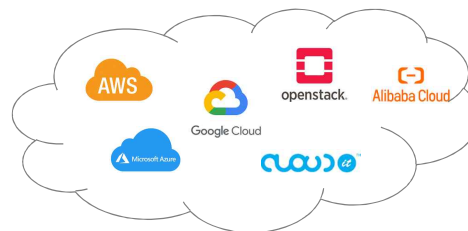
- CB-Spider PoC
- Driver Plugin 도입
- 4종 클라우드 연동



2019.09. ~

**Americano**

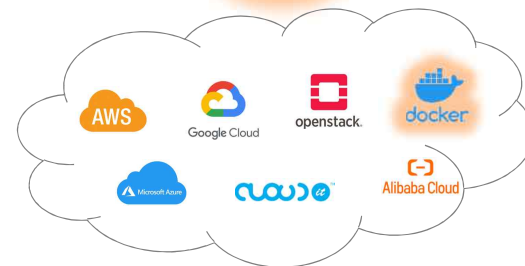
- MC-IaaS 제공
- CB-Store, CB-Log 도입
- 6종 클라우드 연동



2019.12. ~

**Cappuccino**

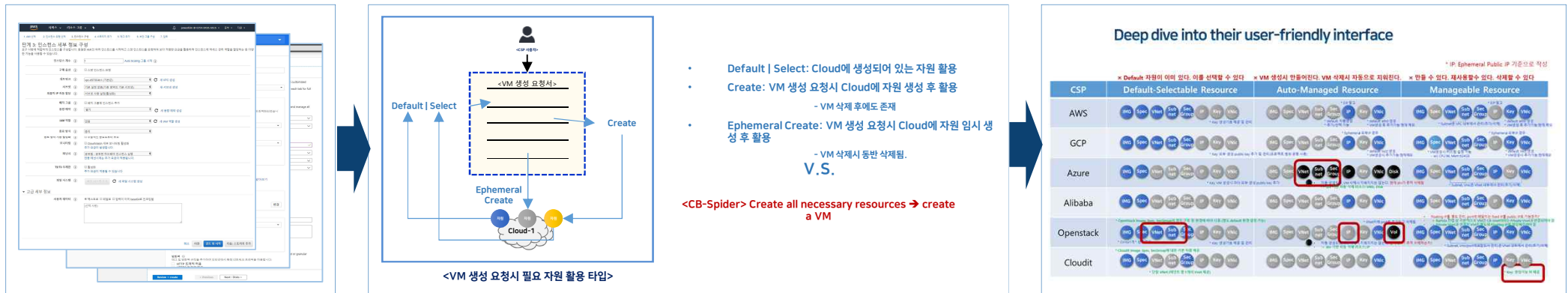
- Americano 리모델링
- Hetero MC-IaaS 제공
- 7종 클라우드 연동





# Americano 리모델링 개발 현황 - 자원 관리 유형 분류 및 개선

- Americano: 모든 자원 사용자 관리 → 기본 VM 생성에도 복잡 → CSP 콘솔 등 재분석 및 관리 유형 재 정의
- Cappuccino: 관리 가능 자원 및 자동 관리 자원 반영 완료



TO-BE

관리 방법에 따른 CB-Spider 자원 분류:

(1) Default Resource, (2) Auto Managed Resource, (3) Manageable Resource

## <기본 제공 자원>

IMG Spec VNet Sub net Sec Group IP Key VNic

※ 이슈: CSP default 제공 자원 활용 여부

## <자동 관리 자원>

IMG Spec VNet Sub net Sec Group IP Key VNic

## <관리 가능 자원>

IMG Spec VNet Sub net Sec Group IP Key VNic

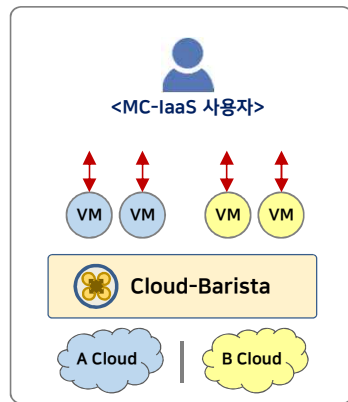
※ Vnic: 향후 추가 고려





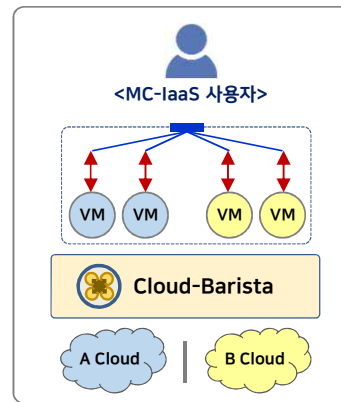
## Americano 리모델링 개발 현황 - 가상 네트워크 추가 개발

- Americano: PublicIP 기반 네트워크 제공 → CSP 고속 네트워크 사용 불가 → 네트워크 통합 재 정의
- Cappuccino: VPC/Subnet 통합 제어 제공



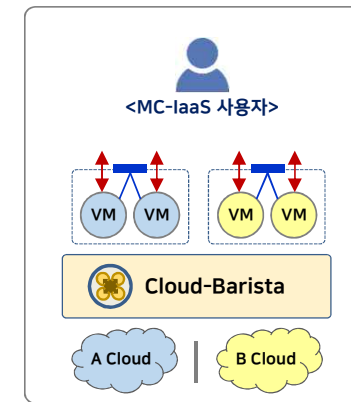
<Independent Network>

- ※ Public-IP만 제공
- ※ Portable VM Replication
- ※ 사용자 근접 서비스
- ※ VM간 네트워크 희박한 실행 환경



<cross dependent Network>

- ※ CB-Subnet 제공
- ※ VM 단위 Composite, BoB 구성
- ※ VM간 네트워크 성능에 덜 민감한 실행 환경



<each dependent Network>

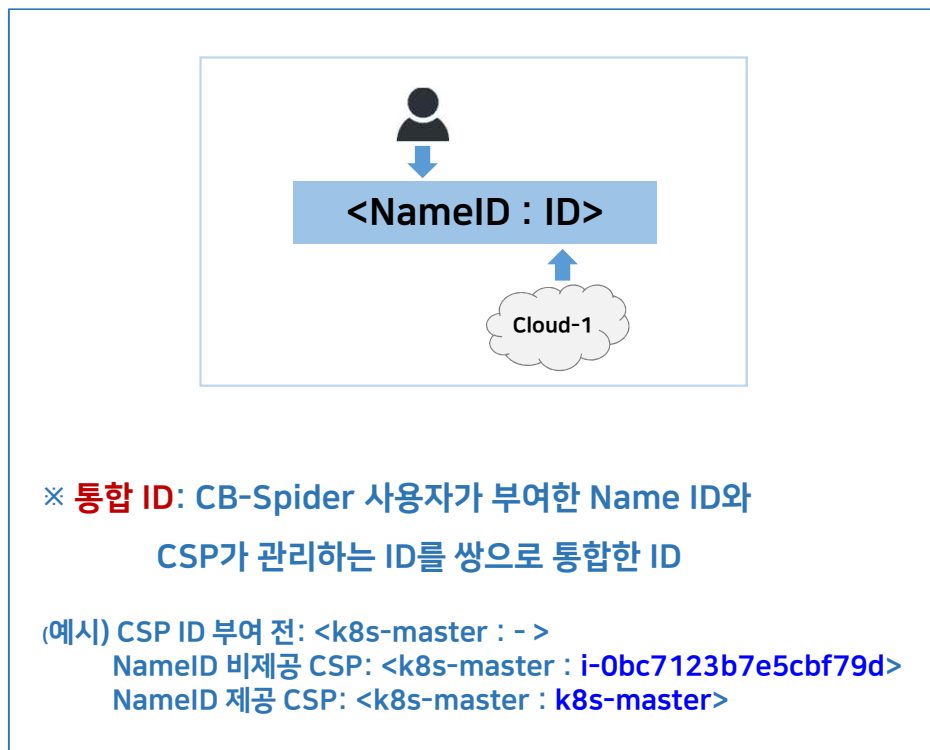
- ※ 개별 CSP들의 전용 고속 네트워크 제공  
- VPC default 자원 사용 가능
- ※ Private-net 단위 Composite
- ※ 사용자의 CSP별 네트워크 자원 관리 필요
- ※ CSP subnet 단위의 분산 처리 서비스 실행 환경

CSP별 VPC 자원 관리 기능 제공 필요 → 관련 드라이버 인터페이스 개선 및 드라이버 패치 완료

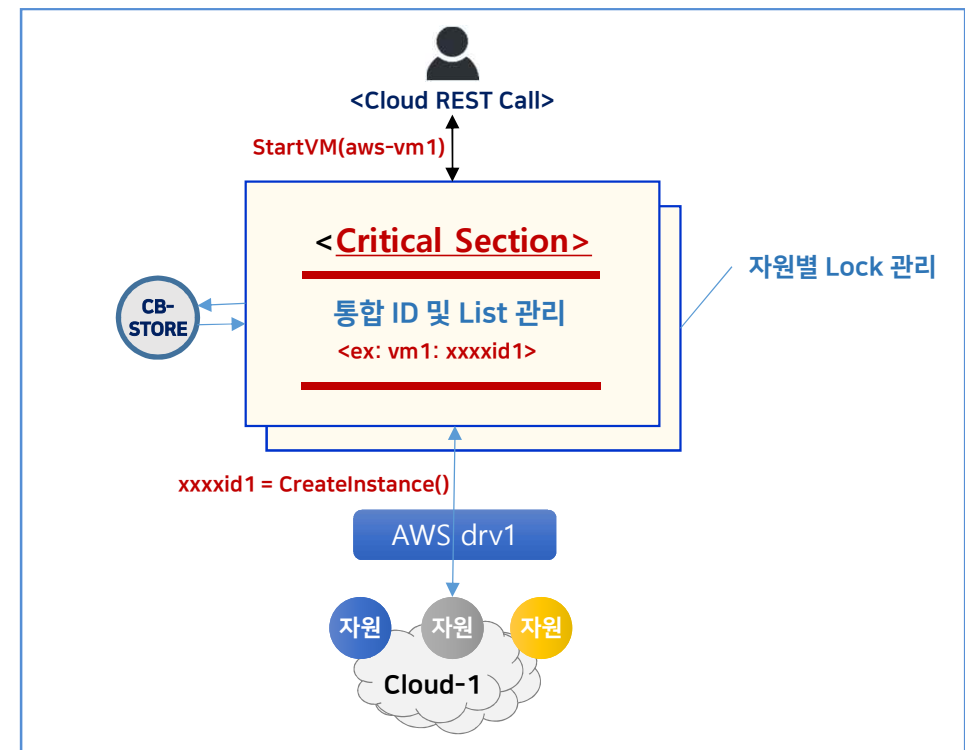


## Americano 리모델링 개발 현황 - 통합ID 관리 체계 도입

- Americano: CSP의 Tagging, Annotation 등을 이용한 드라이버 단에서의 관리 → **드라이버 개발 복잡**
- Cappuccino: 드라이버 상위 수준에서 IID-Manager 추가 개발 → **드라이버 개발 부하 감소**



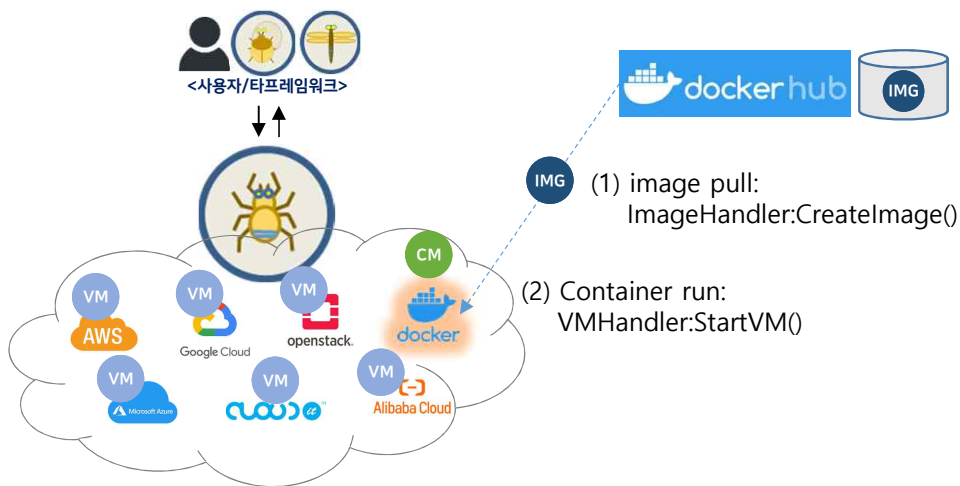
<통합 ID 구조 및 �핑 관계>



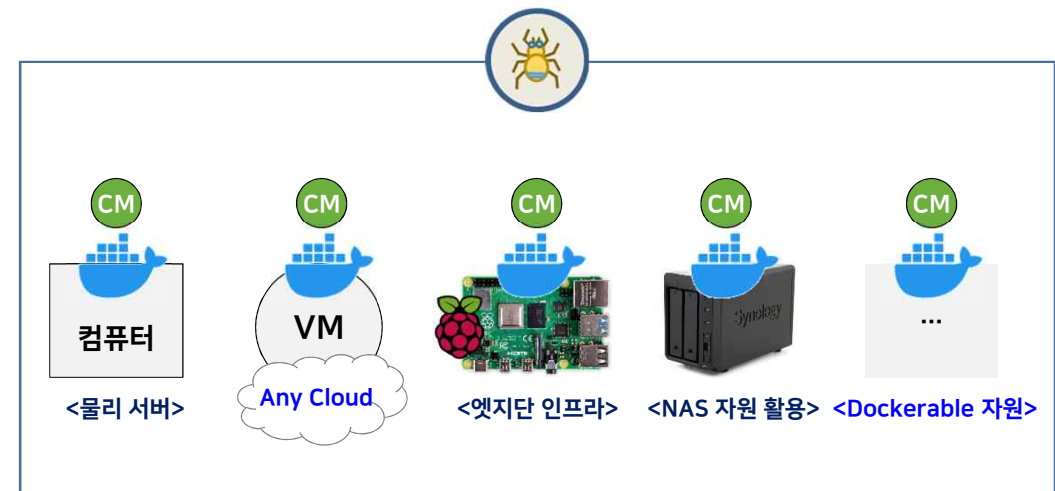
<Cloud 자원 NameID 관리 방안>

# Hetero Multi-iaaS 개발 현황 - Remote Docker 연동

- 연동 대상 선정: 가장 대중적인 Docker Container 우선 연동
- Cappuccino : Image와 Container 제어 기능 개발 및 자체 시험 완료
- Espresso: 타프레임워크 활용, Network, Volume 자원 등 추진 예정
- 향후: Amazon-ECS, Azure-ACS, OpenStack-Zun, Alibaba-Container Service 등 분석 및 연동 고려



## <Docker 연동 형상 및 이미지 제어 방법>



## <Docker 연동으로 가능한 컴퓨팅 자원 유형>



# 클라우드 연동 드라이버 개발 현황

※ 2020.05.23 현재

## <연동 클라우드>

## <공통 제어자원>

	IMG	Spec	VPC/ Sub	Sec	Key	VM
	IMG	Spec	VPC/ Sub	Sec	Key	VM
	IMG	Spec	VPC/ Sub	Sec	Key	VM
	IMG	Spec	VPC/ Sub	Sec	Key	VM
	IMG	Spec	VPC/ Sub	Sec	Key	VM
	IMG	Spec	VPC/ Sub	Sec	Key	VM
	IMG	Spec	VPC/ Sub	Sec	Key	VM

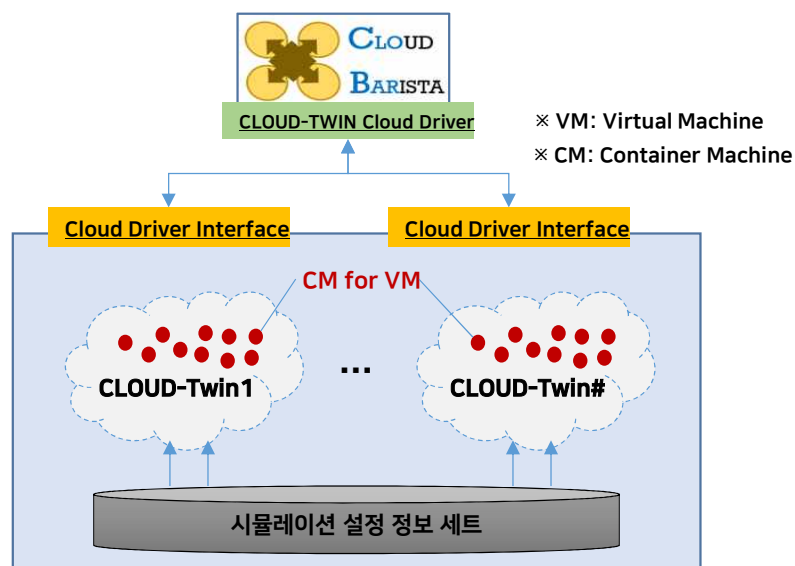
## <범례>

- IMG : VM 이미지
- Spec : VM 사양
- VPC/  
Sub : 가상 네트워크
- Sec : 보안그룹
- Key : 서버 접속 Key
- VM : 가상서버(VM, CM)
- : 현재 미제공

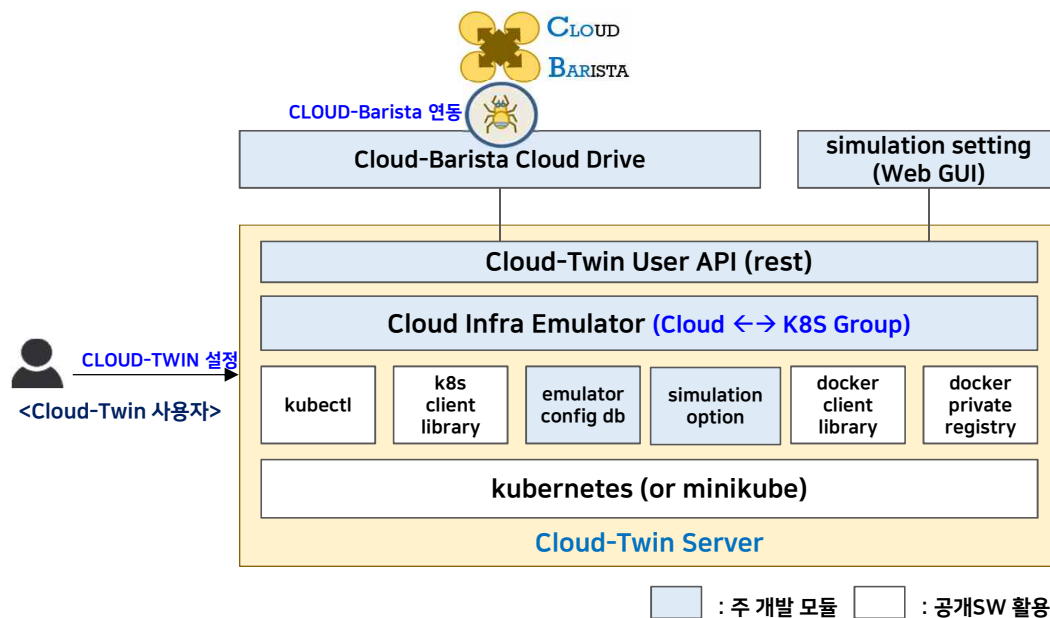


## Cloud-Twin 개발 현황

- Cloud-Twin: 대규모 멀티 클라우드 인프라 가상 시험 환경을 제공하기 위한 에뮬레이터
- Americano: Kubernetes 기반 가상 인프라 환경 주요 기능 개발 및 시험
- Cappuccino : 대규모 모니터링 데이터 시험 환경 제공을 위한 Dragonfly-Agent Container 버전 통합 PoC 완료
- Espresso: Dragonfly 정식 통합 및 대규모 모니터링 데이터 부하 시험을 위한 가상 클라우드 인프라 환경 제공



<Cloud-Twin 개념>



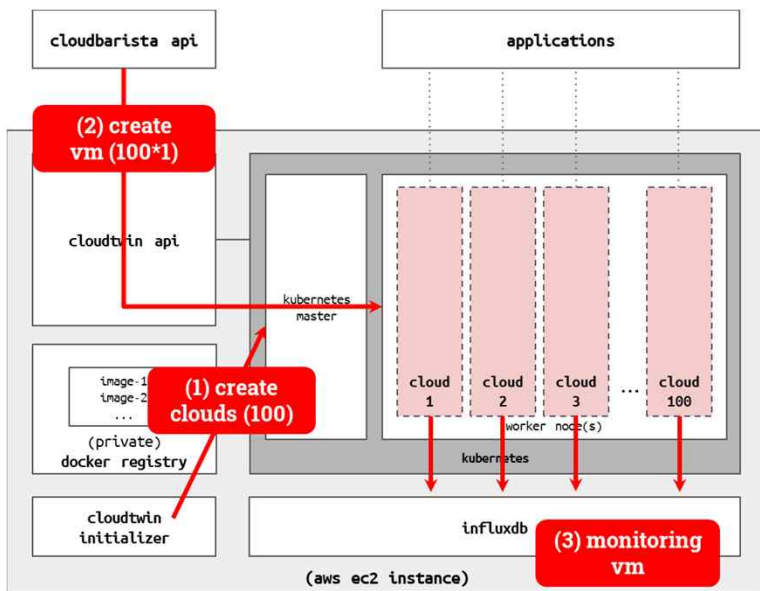
<Cloud-Twin 내부 구조>



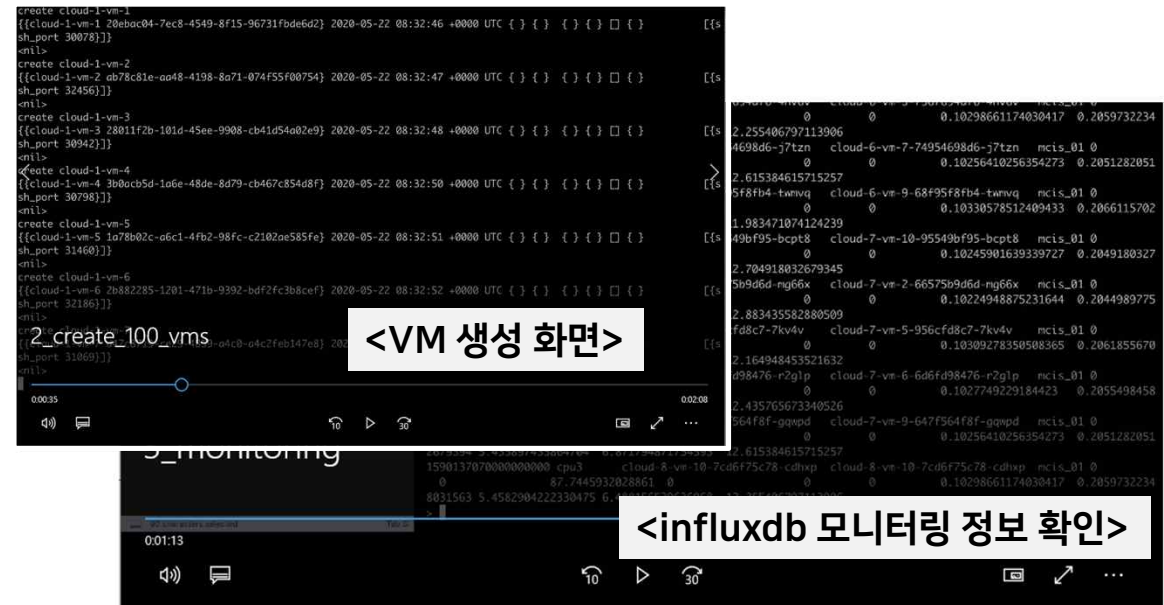
# Cloud-Twin 가상 VM 모니터링 정보 제공 PoC 결과

- (1) 100개의 가상 Cloud 생성 (실제: Kubernetes Namespace)
- (2) Cloud마다 1개의 가상 VM 생성 (실제: Kubernetes PoD-Container)
- (3) 100개의 가상 VM으로 부터 모니터링 데이터 수집 여부 확인 (실제: PoD-Container → influxdb)

※ Influxdb : Dragonfly Server 검증 역할



<PoC 실행 흐름>



<PoC 실행 결과 화면>



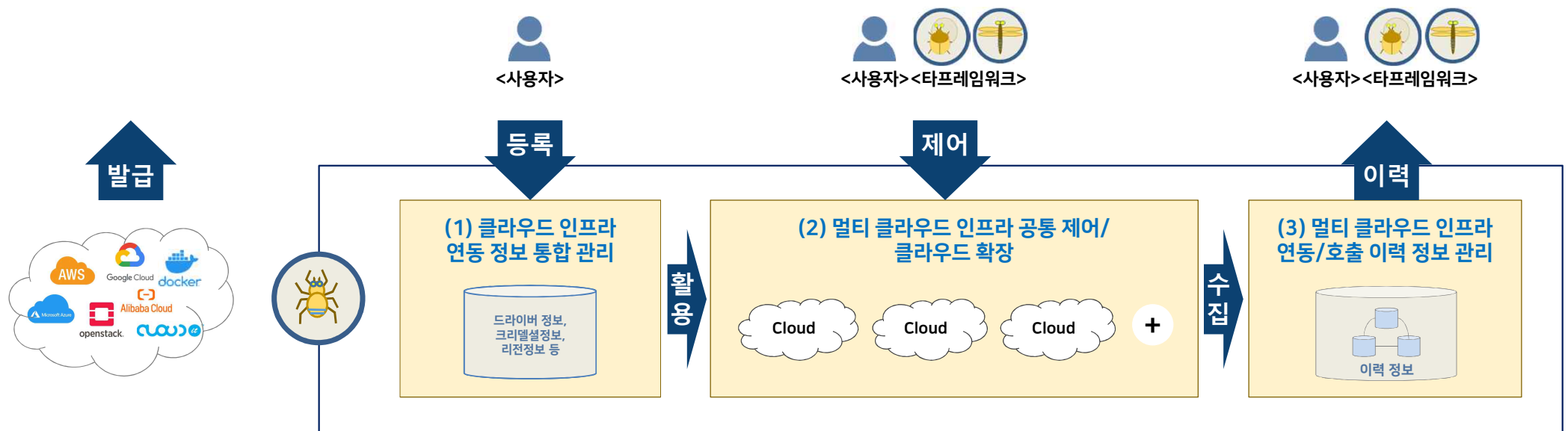
# CB-Spider 서비스 사용 개요

크리덴셜 정보 획득

(1) 클라우드 연동 정보 등록

(2) 인프라 자원/VM 생성 및 제어

(3) 인프라 연결/호출 정보 활용



#공통API #드라이버 #동적플러그인 #클라우드동적추가 #도커연동 #이종멀티클라우드연동 #연동이력제공



# 단일 API/공통 제어 예시

## [이종 클라우드-동일 코드] Image/Spec/VPC/Subnet/SecurityGroup/KeyPair/VM/CM 제어

**AWS:**

```
export CONN_CONFIG=aws-ohio-config
export IMAGE_NAME=ami-f4f4cf91
export SPEC_NAME=t3.micro
```

./full\_test.sh

**Azure:**

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

./full\_test.sh

※ **동일 코드**: 3개의 인자 값 설정으로 멀티 클라우드 **공통 제어** 가능

**full\_test.sh:**

```
echo "#####"
echo "## Full Test Scripts for CB-Spider IID Working Version - 2020.04.22."
echo "## 1. VPC: Create -> List -> Get"
echo "## 2. SecurityGroup: Create -> List -> Get"
echo "## 3. KeyPair: Create -> List -> Get"
echo "## 4. VM: StartVM -> List -> Get -> ListStatus -> GetStatus -> Suspend -> Resume -> Reboot"
echo "## -----"
echo "## 4. VM: Terminate(Delete)"
echo "## 3. KeyPair: Delete"
echo "## 2. SecurityGroup: Delete"
echo "## 1. VPC: Delete"
echo "#####"

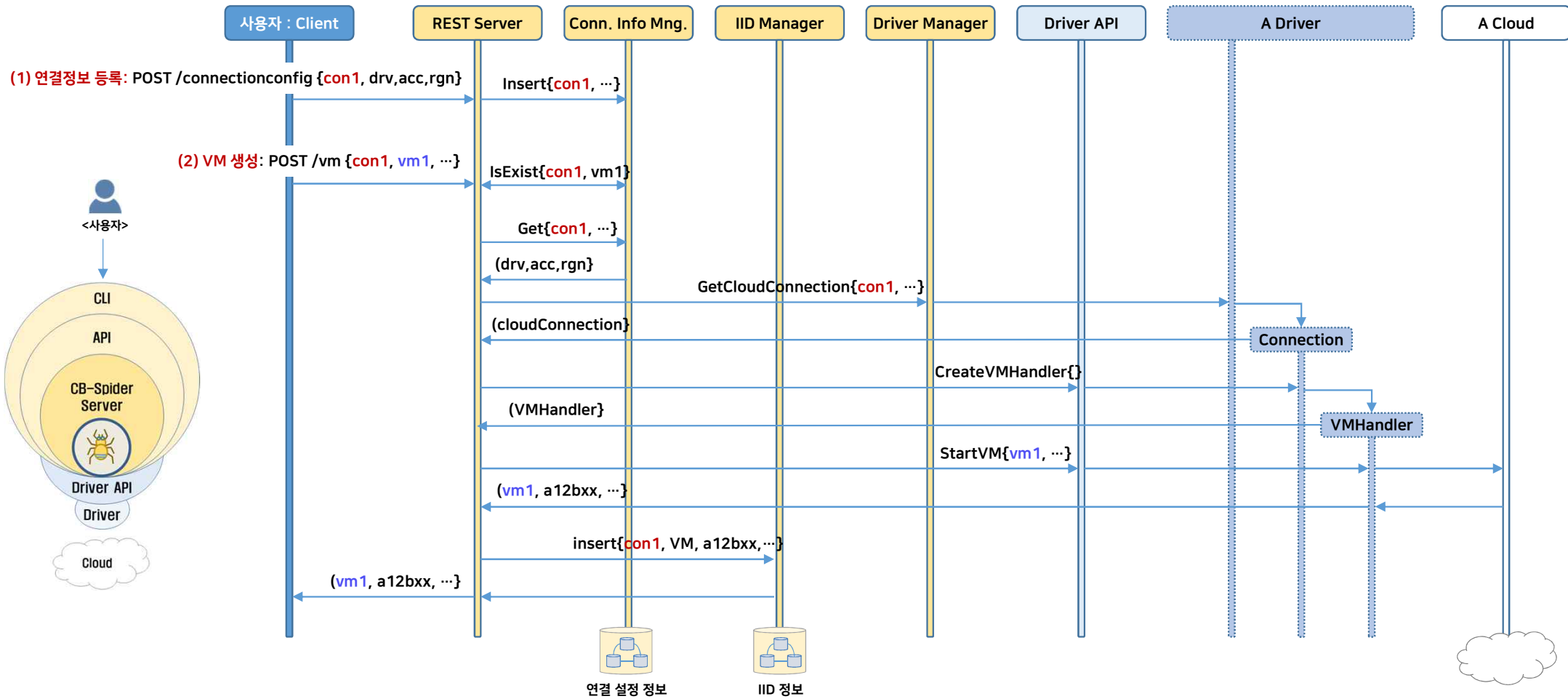
echo "#####"
echo "## 1. VPC: Create -> List -> Get"
echo "#####"
curl -sX POST http://localhost:1024/spider/vpc -H 'Content-Type: application/json'
-d '{ "ConnectionName": "${CONN_CONFIG}",
      "ReqInfo": { "Name": "vpc-01",
                  "IPv4_CIDR": "192.168.0.0/16",
                  "SubnetInfoList": [ { "Name": "subnet-01", "IPv4_CIDR": "192.168.1.0/24" } ] } }'

...
```



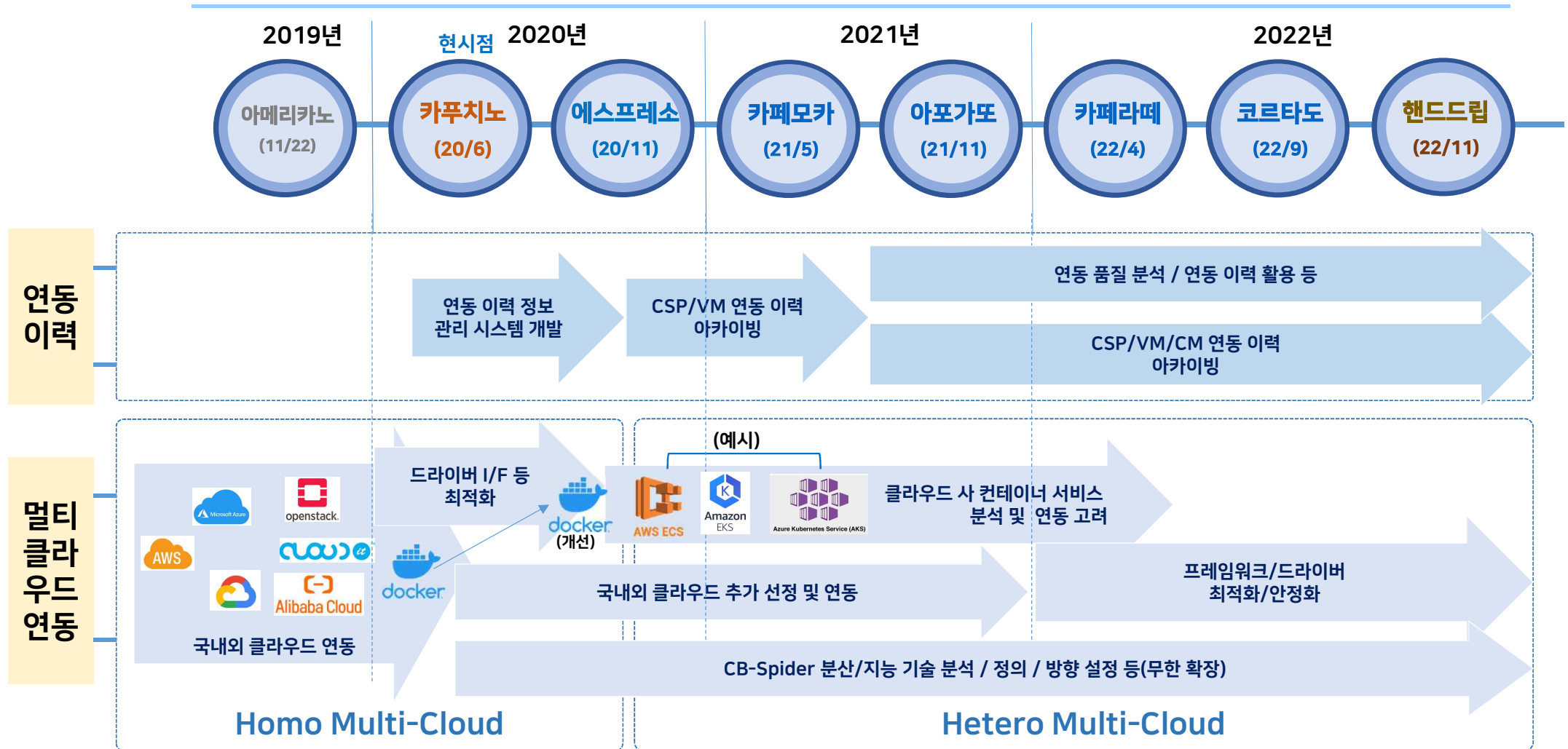


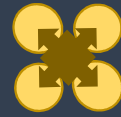
# 멀티 클라우드 제어 주요 흐름 - VM 생성





# CB-Spider 프레임워크 기술 개발 로드맵





CLOUD

BARISTA

멀티 클라우드 서비스 공통 프레임워크

# CB-Spider 활용 및 이종 IaaS 관리 시연

**카푸치노(Cappuccino) 한잔 어떠세요 ? ^^**

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



## 시연 개요

- 전세계 이중 클라우드를 연동(@스마트폰) 후, 개별 클라우드에 VM을 생성하고 제어하기
  - Hetero Multi-iaaS 서비스(6종의 사업자 클라우드 + Container 클라우드)

### <스마트폰 설치 시사점>

- 스마트폰 가동 → 거의 모든 환경에서 운영 가능
- 운영을 위한 추가 자원 불필요
- 멀티 클라우드 기반 다양한 App 개발 가능성 제시



(주의) Cloud-Barista는 스마트폰을 위한 SW는 아닙니다. ^^



# CB-Spider 설치 및 가동

## <1. 사전준비:스마트폰> 시연 생략

- (1) Termux 설치: Android Playstore 참조
  - Android용 터미널 프로그램
- (2) SSH 데몬 설치 및 실행
  - \$ pkg install openssh
  - \$ sshd

## <2. Go 설치:Terminal 접속> 시연 생략

- (1) 스마트폰 접속
  - \$ ssh u0\_a282@10.213.9.104 -p 8022
- (2) Git / Golang 설치
  - \$ pkg install git
  - \$ pkg install golang
- (3) Gold Linker 설치
  - \$ pkg install binutils-gold

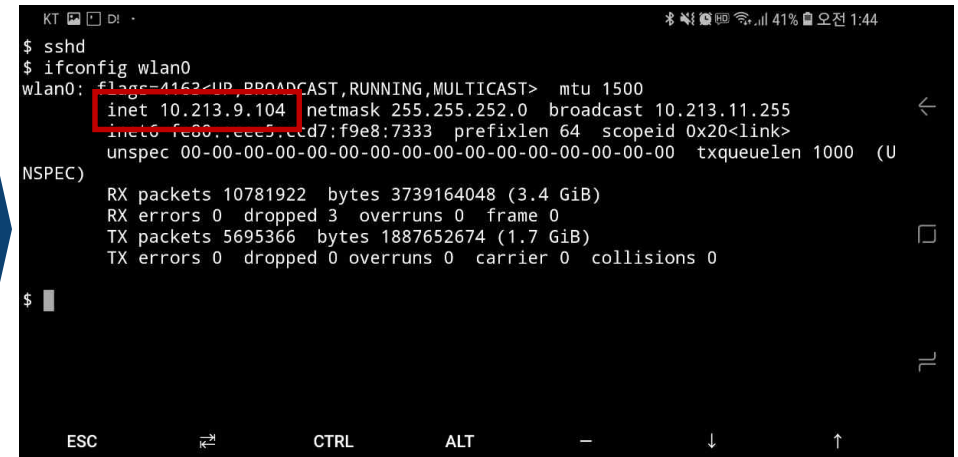
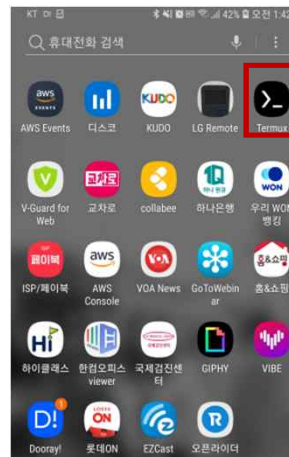
## <3. Spider설치:Terminal 접속>

- (1) 소스 다운로드
  - \$ go get github.com/cloud-barista/cb-spider
  - 저장위치: ~/go/src/github.com/cloud-barista/cb-spider

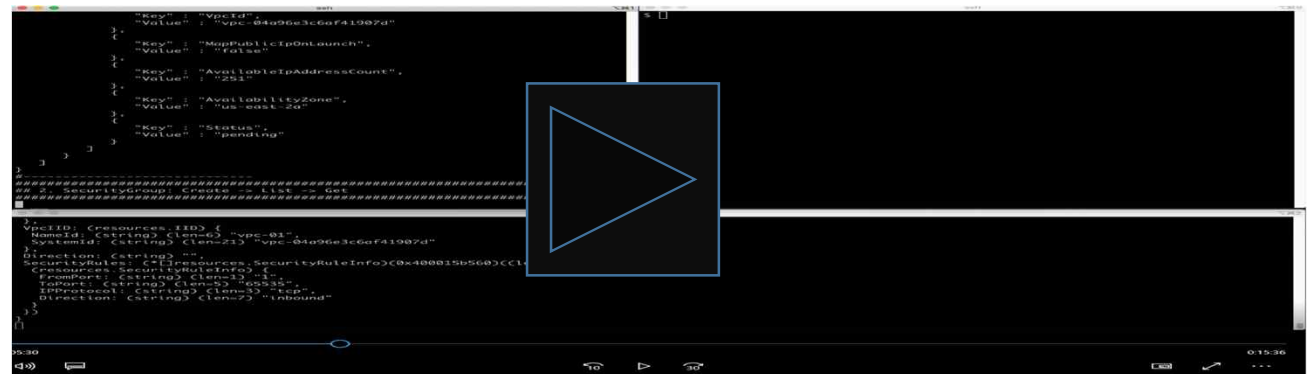
## <4. Spider실행:Terminal 접속>

```
$ cd ~/go/src/github.com/cloud-barista/cb-spider
$ vi setup.env # export PLUGIN_SW=OFF 설정
$ source setup.env
$ cd $CBSPIDER_ROOT/api-runtime/rest-runtime/
$ go run *.go (or $ go build *.go; $ ./CBSpiderRuntime)
```

## <1. 사전준비:스마트폰> Termux 설치 및 sshd 데몬 실행 → 이후 터미널 접속 가능



## <3. Spider설치 이후 영상 참조>





# Hetero Multi-IaaS 통합 제어

## <1. 클라우드 연동 정보 등록>

- (1) Driver 정보 등록
- (2) Credential 정보 등록
- (3) Region 정보 등록

## <2. 인프라자원 및 VM/CM 생성 및 상태 조회>

- (1) VPC/Subnet 생성
- (2) SecurityGroup 생성
- (3) KeyPair 생성
- (4) VM/CM 생성

## <1. 클라우드 연동 정보 등록>

```

{
  "Key": "VpcId",
  "Value": "vpc-04a96e3c6af41907d"
},
{
  "Key": "MapPublicIpOnLaunch",
  "Value": "false"
},
{
  "Key": "AvailableIpAddressCount",
  "Value": "251"
},
{
  "Key": "AvailabilityZone",
  "Value": "us-east-2a"
},
{
  "Key": "Status",
  "Value": "pending"
}
}
}

VpcID: CResources.ID {
  NameId: (string) (len=6) "vpc-01",
  SystemId: (string) (len=21) "vpc-04a96e3c6af41907d"
},
Direction: (string) "inbound",
SecurityRules: (Resources.SecurityRuleInfo)(0x400015b560)(len=1 cap=4) {
  (Resources.SecurityRuleInfo) {
    FrontPort: (string) (len=1) "1",
    ToPort: (string) (len=5) "65535",
    IPProtocol: (string) (len=3) "tcp",
    Direction: (string) (len=7) "inbound"
  }
}
}
}

```

## <2. 인프라자원 및 VM/CM 생성 및 상태 조회>

```








{
  "Key": "VpcId",
  "Value": "vpc-04a96e3c6af41907d"
},
{
  "Key": "MapPublicIpOnLaunch",
  "Value": "false"
},
{
  "Key": "AvailableIpAddressCount",
  "Value": "251"
},
{
  "Key": "AvailabilityZone",
  "Value": "us-east-2a"
},
{
  "Key": "Status",
  "Value": "pending"
}
}
}








VpcID: CResources.ID {
  NameId: (string) (len=6) "vpc-01",
  SystemId: (string) (len=21) "vpc-04a96e3c6af41907d"
},
Direction: (string) "inbound",
SecurityRules: (Resources.SecurityRuleInfo)(0x400015b560)(len=1 cap=4) {
  (Resources.SecurityRuleInfo) {
    FrontPort: (string) (len=1) "1",
    ToPort: (string) (len=5) "65535",
    IPProtocol: (string) (len=3) "tcp",
    Direction: (string) (len=7) "inbound"
  }
}
}
}

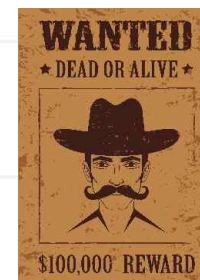
```



## CB-Spider를 만들어 가는 사람들

<input type="checkbox"/>		Jeongwoon
<input type="checkbox"/>		swsok
<input type="checkbox"/>		Seokho Son seokho-son
<input type="checkbox"/>		Yoo Jae-Sung dev4unet
<input type="checkbox"/>		jaehong park iworkist
<input type="checkbox"/>		hellowkorea
<input type="checkbox"/>		ByoungSeob Kim powerkimhub

<input type="checkbox"/>		Jihoon Seo jihoon-seo
<input type="checkbox"/>		kyongminkwon
<input type="checkbox"/>		B.T. Oh innodreamer
<input type="checkbox"/>		hyokyungk
<input type="checkbox"/>		zephy-Jeong zephy-mzc
<input type="checkbox"/>		Jin-Whee-Park Jin-Whee-Park
<input type="checkbox"/>		ChoiNakSoo



# 감사합니다.

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

(김 병 섭 / [contact-to-cloud-barista@googlegroups.com](mailto:contact-to-cloud-barista@googlegroups.com))

## “Contact to the Multi-Cloud”

클라우드 바리스타들의 두 번째 이야기

### Cloud-Barista Community 2<sup>nd</sup> Open Conference

본 발표자료에는 네이버에서 제공한 나눔글꼴이 적용되어 있습니다.