



“오픈소스”로 만들어가는 “멀티클라우드” 생태계

클라우드바리스타 커뮤니티 제6차 컨퍼런스

[세션] CB-Ladybug

멀티클라우드 애플리케이션 실행환경 통합 관리

김수영

CB-Ladybug 프레임워크 리더

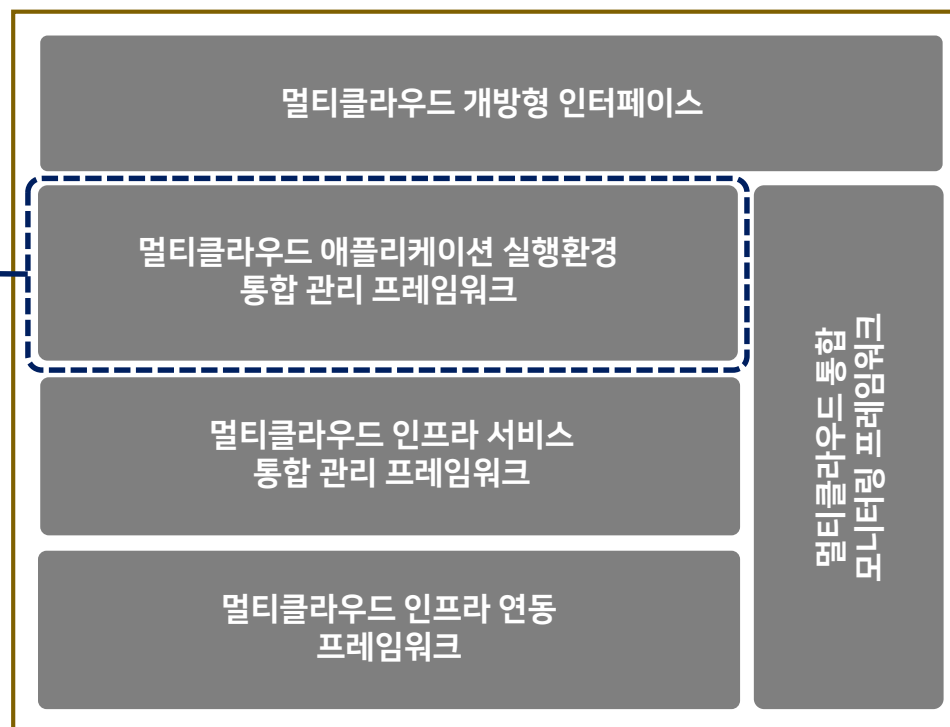
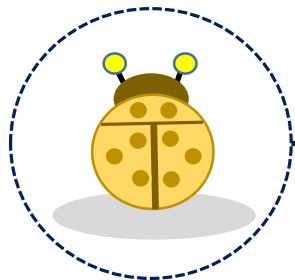
카페라떼(Cafe Latte) 한잔 어떠세요 ?

이번 세션은...

응용/도메인/기관 특화 SW



CB-Ladybug



멀티클라우드 서비스 공통 플랫폼

I CB-Ladybug 개요

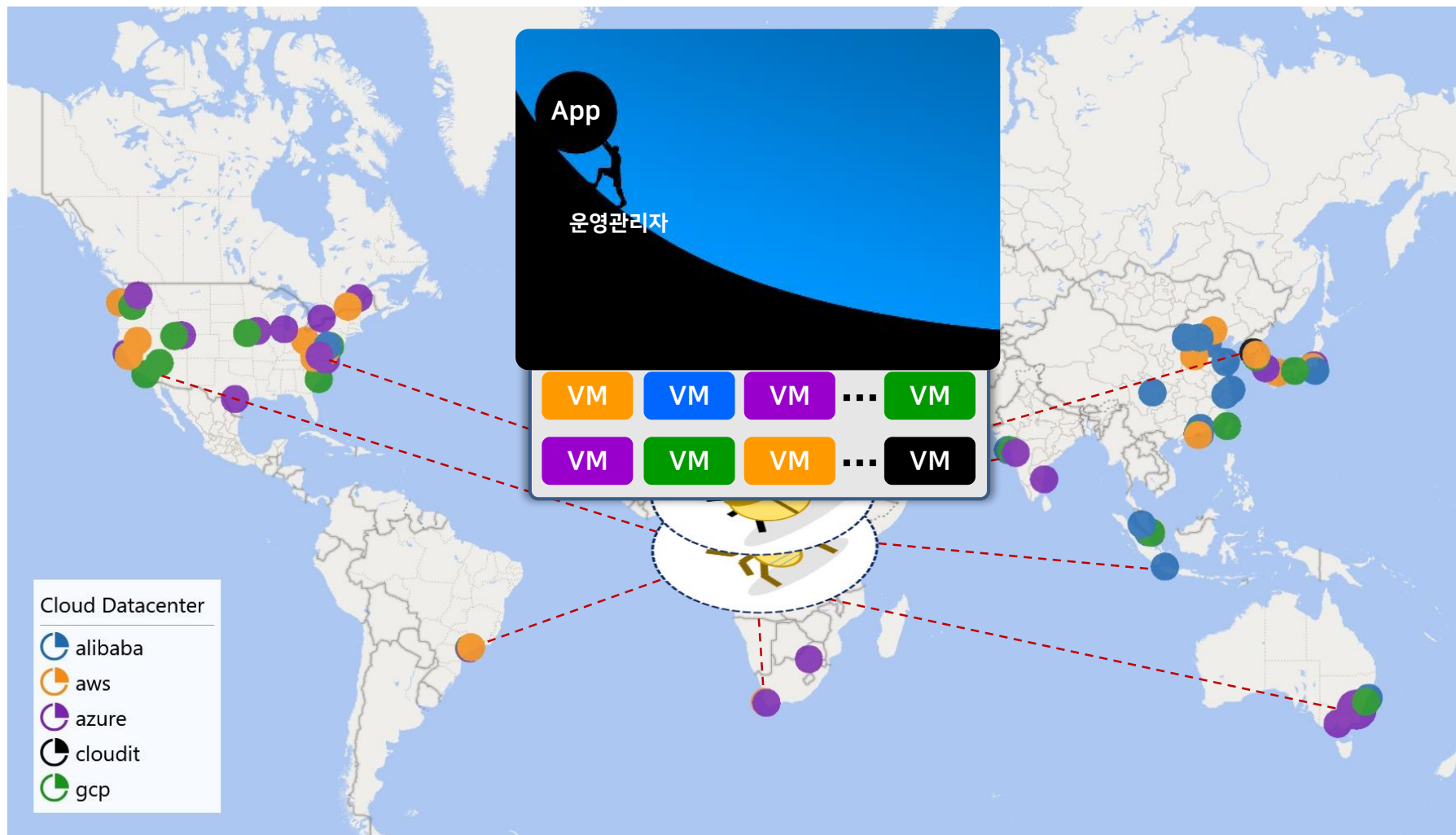
II CB-Ladybug 개발 방향 및 제공 서비스

III CB-Ladybug 특징 및 기대 효과

IV CB-Ladybug 구조 및 기술 현황

V CB-Ladybug 로드맵

멀티클라우드 애플리케이션의 배포와 관리는?



?????

멀티클라우드 자원
통합 컴퓨팅 인프라

멀티클라우드 인프라 서비스 (MCIS)

컴퓨팅
자원

컴퓨팅
자원

컴퓨팅
자원

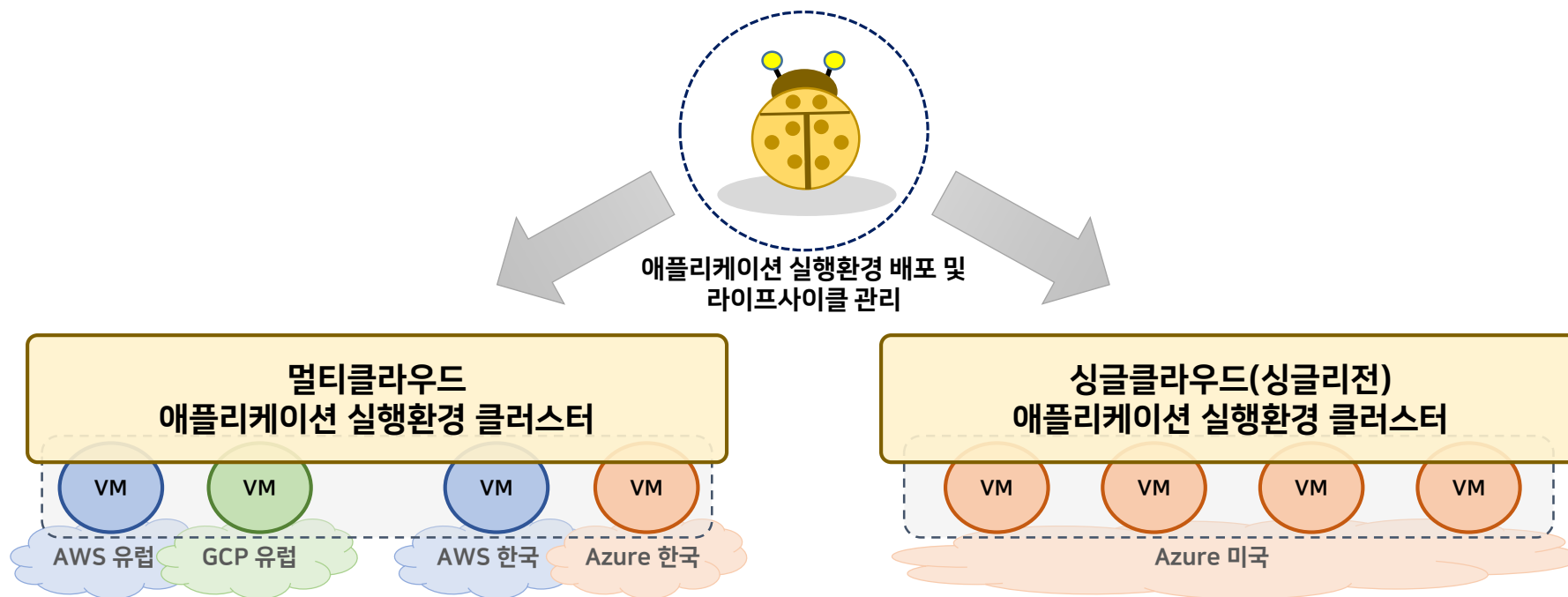
이중 멀티클라우드 연동



멀티클라우드 인프라를 기반으로, 글로벌 서비스를 위한 애플리케이션을 보다 쉽고 빠르게 배포하고 운영/관리할 방법이 필요

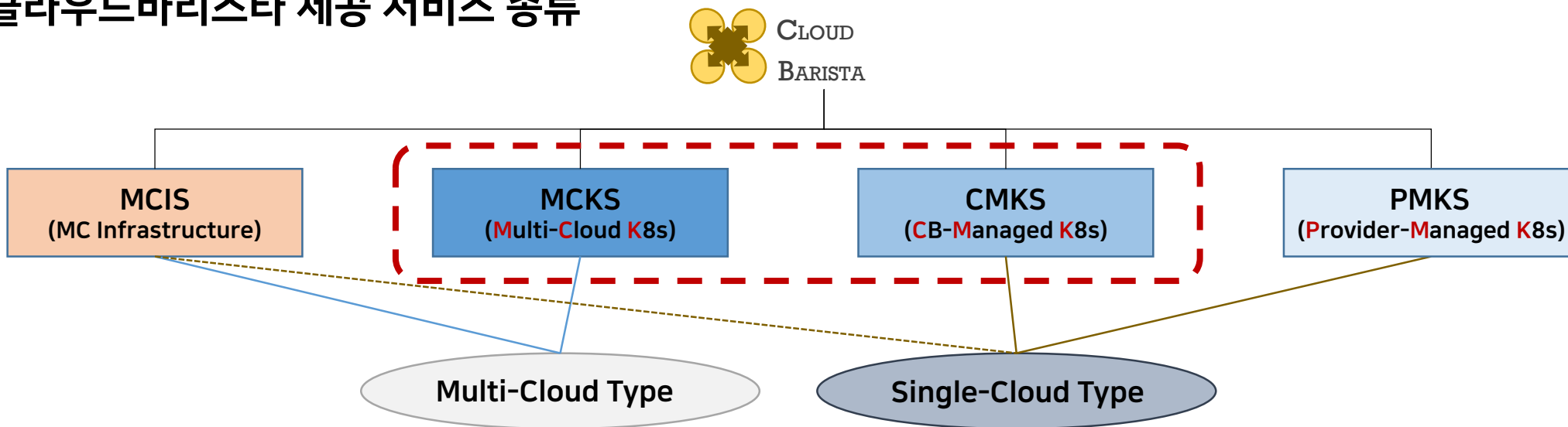
CB-Ladybug 개요

- 멀티클라우드 애플리케이션 실행환경 통합 관리 프레임워크(CB-Ladybug)
 - 멀티클라우드 인프라 상에서 실행되는 멀티클라우드 애플리케이션(MC-App)의 실행/제어를 위한 애플리케이션 실행환경(쿠버네티스) 클러스터의 배포 및 라이프사이클 관리 기능 제공 프레임워크
 - 주요 활용 분야: 사용자/데이터 근접 처리 응용 환경, 가용성 극대화 응용 환경, 초광역 커버리지 필요 응용 환경, 분산 병렬 처리 응용 환경 등



CB-Ladybug 개발 방향 (1/2)

- 클라우드바리스타 제공 서비스 종류



- 클라우드 애플리케이션 실행환경의 산업표준이 된 쿠버네티스(K8s)
- 멀티클라우드 애플리케이션(MC-App) 통합 운영관리보다 다양한 실행환경에 대한 요구
- 커뮤니티 차원에서 다양한 KaaS(K8s as a Service) 타입 제공을 위한 로드맵 변경
 - CB-Ladybug: MCKS, CMKS 제공

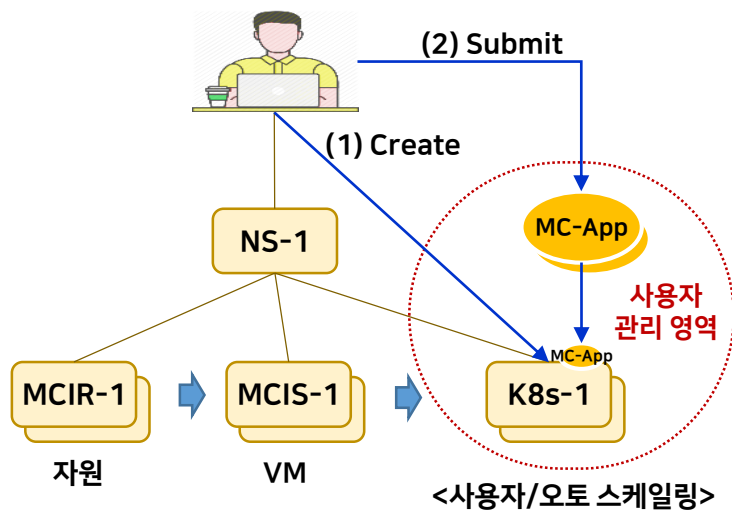
CB-Ladybug 개발 방향 (2/2)

- 멀티클라우드 환경에 적합한 애플리케이션 실행환경 제공 기술 개발 (MCKS)
 - 목표: 미래 수요를 대비한 Extreme-Scale 확장성 및 고가용성, 데이터 주권 등 제공
 - 멀티클라우드 인프라 상에서 애플리케이션 실행환경의 운용 가능성/적합성 확인 및 검증
 - 클라우드를 인지하는 멀티 클러스터 간/클러스터 내 MC-App 최적 배치 알고리즘 개발
- 클라우드 자원을 온전히 활용할 수 있는 애플리케이션 실행환경 제공 기술 개발 (CMKS)
 - 목표: 현실적 수요 충족하고 안정적인 서비스 제공
 - 클라우드 내 자원(로드밸런서, 스토리지, 네트워크 등) 연계 지원
 - 싱글클라우드(싱글리전) 인프라 환경에서도 동일 인터페이스로 애플리케이션 실행환경 설치/운용 지원

CB-Ladybug 제공 서비스

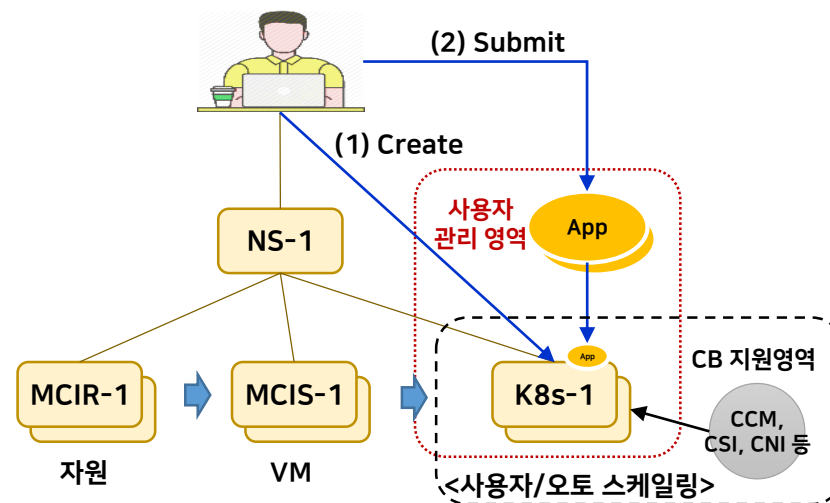
멀티클라우드 쿠버네티스 서비스 (MCKS)

- 멀티클라우드 대상 쿠버네티스 클러스터 제공
- 클러스터 간/내 MC-App 최적 배치 기능 제공



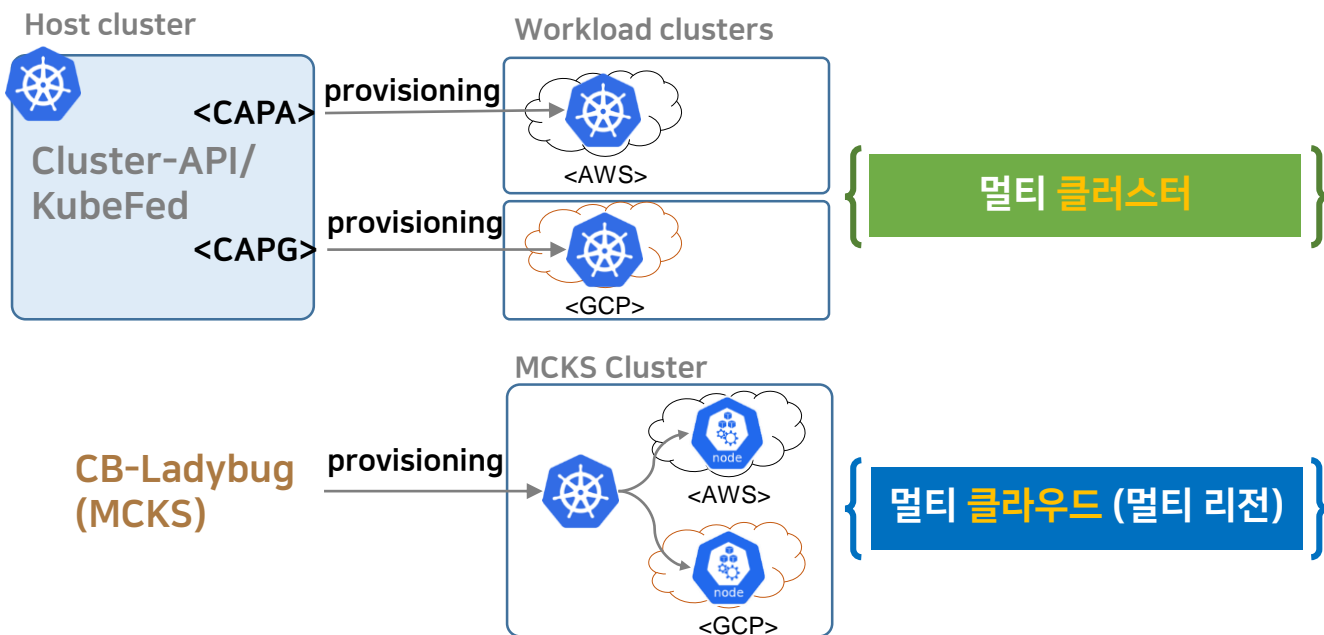
클라우드바리스타 관리형 쿠버네티스 서비스 (CMKS)

- 싱글클라우드 대상 쿠버네티스 클러스터 제공
- 클라우드 자원 활용 플러그인(CCM, CSI, CNI 등) 제공

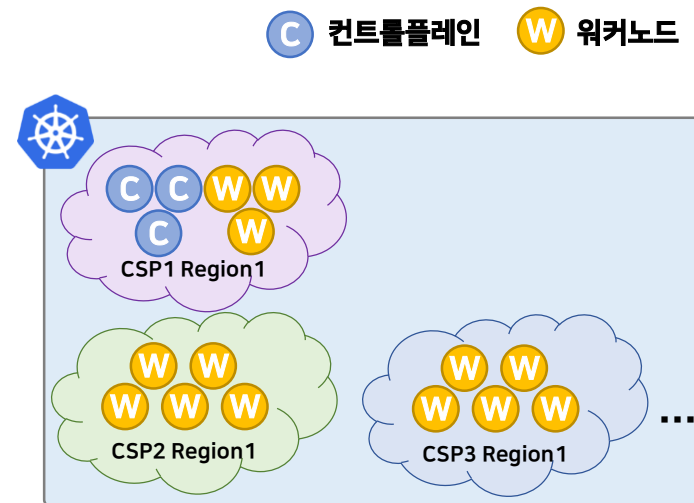


CB-Ladybug 특징

- 멀티 클라우드 리전별로 노드를 분산시키고, 이들을 묶어 하나의 쿠버네티스 클러스터로 구성
 - 멀티 클러스터 통합 관리 방식(Cluster-API/KubeFed)과는 구별
 - 분산된 멀티 클라우드 환경에서도 단일 쿠버네티스 클러스터와 동일한 방식으로 쿠버네티스 활용 및 관리 가능
 - 분산된 지역을 대상으로 소규모 클러스터 운영부터 대규모 서비스 제공 가능



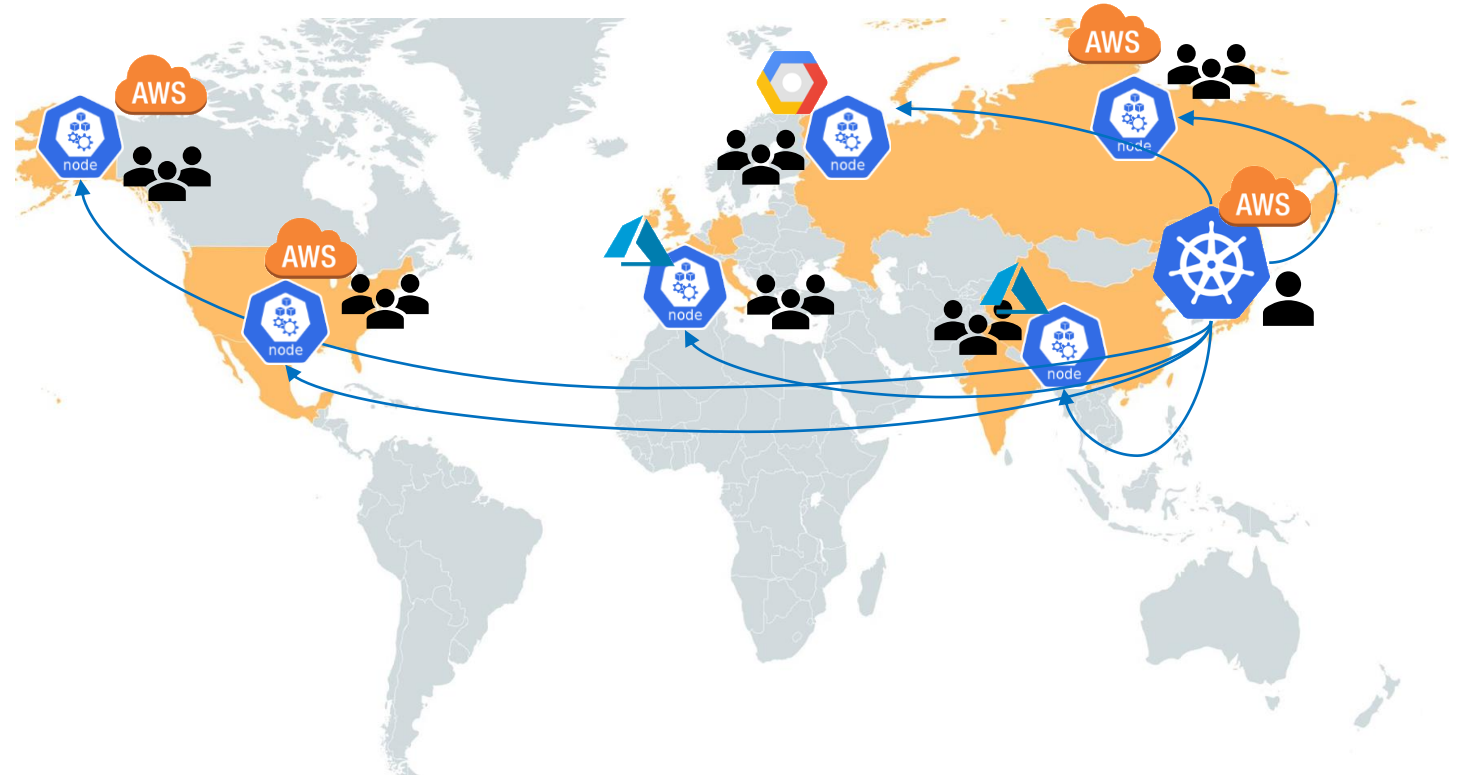
<멀티 클러스터 vs 멀티 클라우드 구성 방식 비교>



<MCKS 멀티 클라우드 노드 구성 예시>

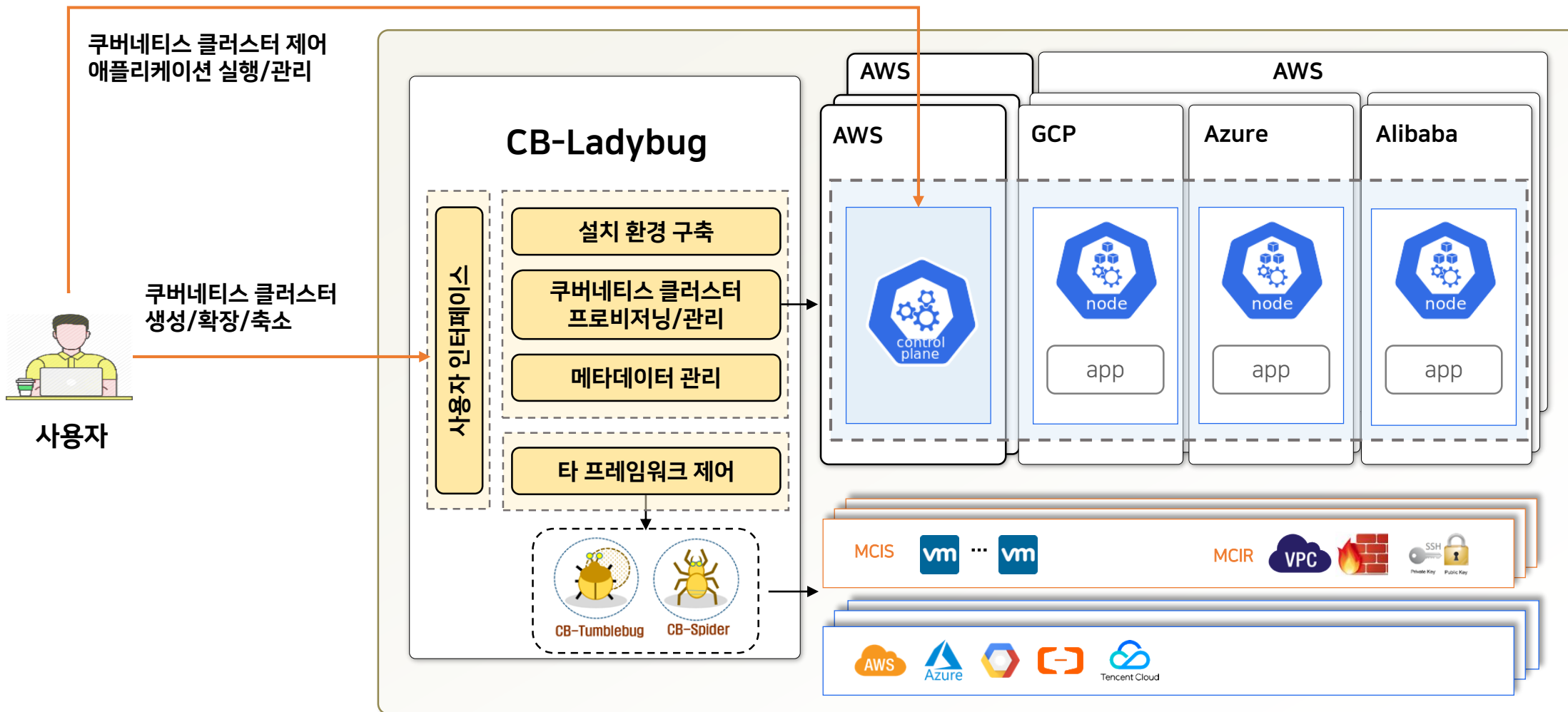
CB-Ladybug 기대효과

- 글로벌 스케일 서비스 디스커버리 및 고가용성 제공
- 글로벌 스케일 사용자 서비스 배치 및 운용, 확장성 제공
- 클라우드 벤더 락인 탈피
- 멀티클라우드 자원 통합 운용/관리
- 개인정보 보호 및 데이터 주권



<멀티클라우드 쿠버네티스 서비스(MCKS) 형상 예시>

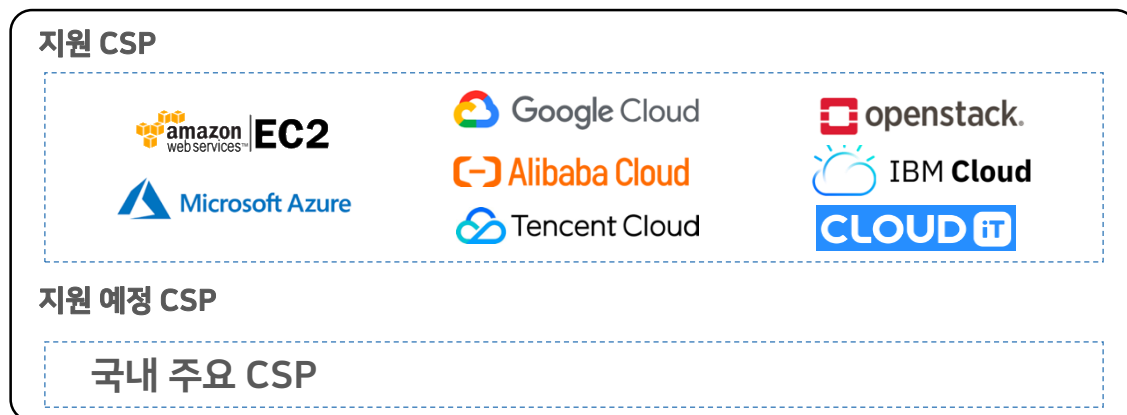
CB-Ladybug 구조



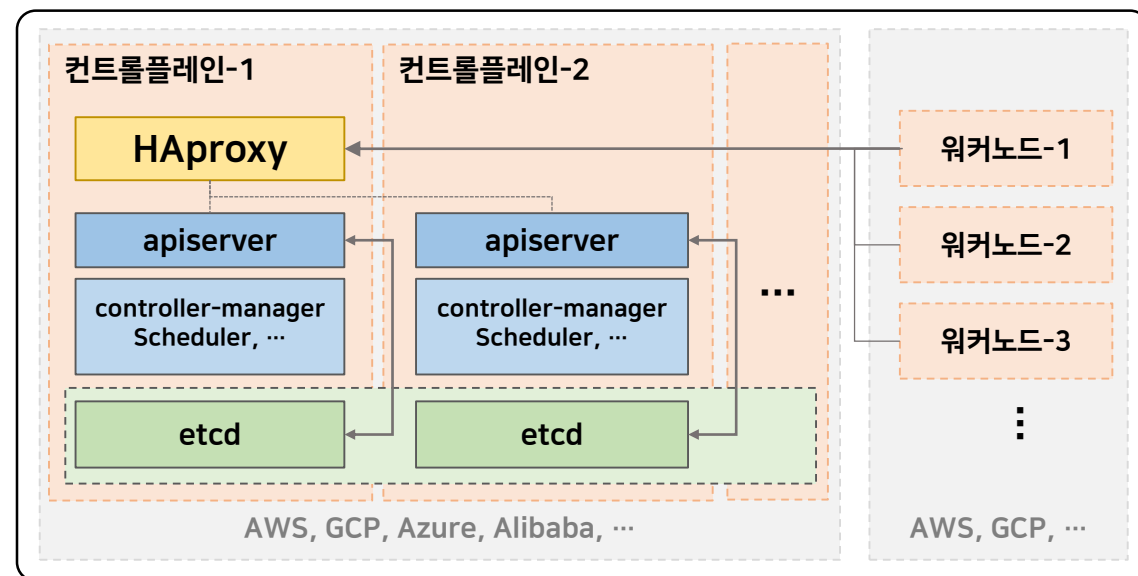
< CB-Ladybug 구조 및 동작 개요 >

MCKS 기술 현황

- 멀티클라우드 대상
쿠버네티스 클러스터 프로비저닝 지원

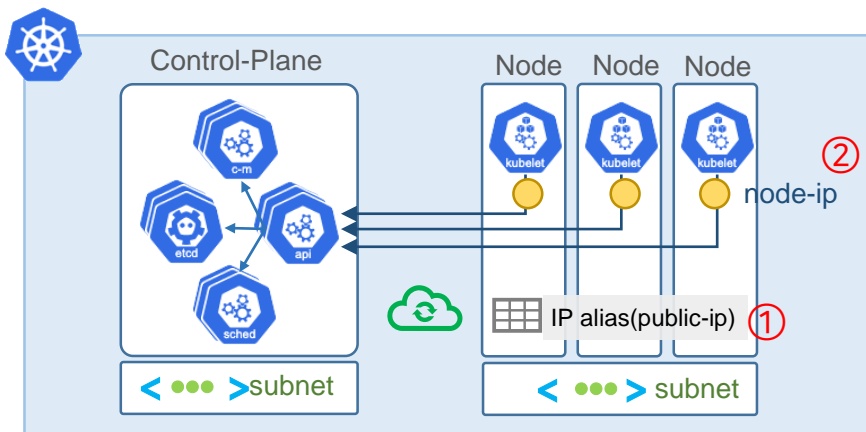


- 쿠버네티스 컨트롤플레인의 HA 구성 지원



- 지원 쿠버네티스 버전: 1.18.x, 1.23.y (.x, .y 버전 사용자 선택 가능)
- 멀티클라우드 운영 가능 쿠버네티스 네트워크 플러그인
2종 (Canal, Kilo) 적용 가능
 - Canal (Flannel+Calico): 다양한 네트워크 정책 적용
 - Kilo: VPN(Wireguard) 활용 서로 다른 클라우드의 노드 간 통신 보안 제공

MCKS 프로비저닝 주요 이슈



<서로 다른 서브넷을 가진 MCKS 클러스터>

①

```
$ sudo ifconfig eth0:1 ${PUBLIC_IP} netmask 255.255.255.255 broadcast 0.0.0.0 up
```

```
eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
      inet 3.112.226.180 netmask 255.255.255.255 broadcast 0.0.0.0
      ether 06:4d:23:c6:b4:7b txqueuelen 1000 (Ethernet)
```

②

```
$ sudo echo "KUBELET_EXTRA_ARGS=-\\"-node-ip=${PUBLIC_IP}\\\"" > /etc/default/kubelet
```

<노드 IP alias 부여 및 kubelet node-ip 옵션 지정>

IP alias(public-ip)

```
~$ kubectl get node -o wide
```

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXT
c-1-2vkt2	Ready	master	108s	v1.18.9	3.112.226.180	<nor
w-1-3f5l8	Ready	<none>	85s	v1.18.9	34.146.174.214	<nor
w-2-q03z1	Ready	<none>	76s	v1.18.9	3.112.127.148	<nor

<프로비저닝된 MCKS 노드들의 Internal-IP>

- VM의 reboot 등 Public-IP의 변경 이슈
 - system daemon 활용 kubelet의 node-ip 옵션을 동적으로 변경
- 컨트롤 플레인-노드 간 통신 보안 이슈
 - 노드에서 컨트롤 플레인까지 통신은 신뢰할 수 있는 구간 (TLS)
 - 컨트롤 플레인에서 노드까지 통신은 신뢰할 수 없는 구간이므로 터널링 등을 통해 신뢰 구간 확보 (예정)

MCKS 맛보기

• 사용자 맛보기 절차

<MCKS, CB-Spider, CB-Tumblebug 실행>

```
$ docker run -d -p 1024 ...cloudbaristaorg/cb-spider:latest
$ docker run -d -p 1323 ...cloudbaristaorg/cb-tumblebug:latest
$ docker run -d -p 1470 ...cloudbaristaorg/cb-mcks:latest
```

<클러스터 생성>

```
$ cbadm create cluster \
  --namespace "cloud-barista"\
  --name "cb-cluster"\
  --control-plane-connection="config-aws-tokyo"\
  --control-plane-count="1"\
  --control-plane-spec="t2.medium"\
  --worker-connection="config-gcp-tokyo"\
  --worker-count="2"\
  --worker-spec="e2-highcpu-4"
```

<워커노드 확장>

```
$ cbadm create node \
  --namespace "cloud-barista"\
  --cluster "cb-cluster"\
  --worker-connection="config-aws-tokyo"\
  --worker-count="1"\
  --worker-spec="t2.medium"
```

• 개발자 맛보기 절차

<소스 및 dependency 다운로드>

```
$ git clone https://github.com/cloud-barista/cb-mcks.git
$ cd cb-mcks
$ go get -v all
```

<CB-Spider, CB-Tumblebug 실행>

```
$ docker run -d -p 1024 ...cloudbaristaorg/cb-spider:latest
$ docker run -d -p 1323 ...cloudbaristaorg/cb-tumblebug:latest
```

<환경 설정 및 MCKS 실행>

```
$ export CBLOG_ROOT="$(pwd)"
$ export CBSTORE_ROOT="$(pwd)"
$ export APP_ROOT="$(pwd)"

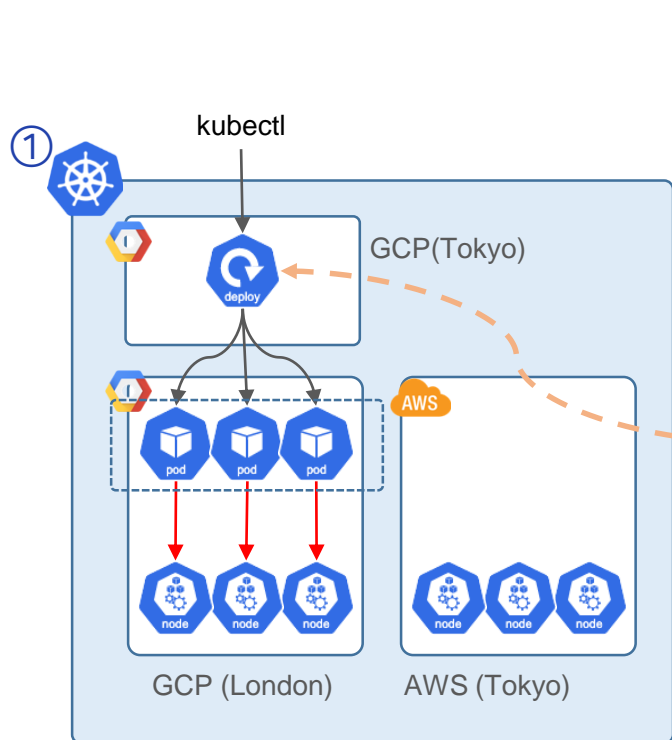
$ go run src/main.go
```

<테스트>

```
$ curl -s http://localhost:1470/mcks/healthy -o /dev/null -w
"code:%{http_code}"
```

MCKS 운용 예시

- GCP Tokyo, GCP London, AWS Tokyo 리전으로 구성된 MCKS 클러스터 구성 환경
 - nodeAffinity 활용 GCP London 리전(europe-west2) 노드에 busybox 배포 요청 및 정상 배포 확인



topology.cloud-barista.github.io/csp
topology.kubernetes.io/region

<예시 클러스터 및 워크로드 배포 형상>

③

```
> spec:
>   replicas: 3
>   selector:
>     matchLabels:
>       app: busybox
>   template:
>     metadata:
>       labels:
>         app: busybox
>     spec:
>       affinity:
>         nodeAffinity:
>           requiredDuringSchedulingIgnoredDuringExecution:
>             nodeSelectorTerms:
>               - matchExpressions:
>                 - key: topology.cloud-barista.github.io/csp
>                   operator: In
>                   values:
>                     - gcp
>                 - key: topology.kubernetes.io/region
>                   operator: In
>                   values:
>                     - europe-west2
>       containers:
>         - name: busybox
>           image: busybox
>           imagePullPolicy: IfNotPresent
>           command: ['sh', '-c', 'echo Running ; sleep 3600']
> EOF
```

<GCP 런던 리전 배포 요구 사항>

②

```
~$ kubectl get nodes -o custom-columns='NAME:.metadata.name,KERNEL:.status.nodeInfo.kernelVersion'
```

NAME	INTERNAL-IP	CSP	REGION
c-1-296ts	34.146.106.159	gcp	asia-northeast1
c-2-abf0q	34.146.220.157	gcp	asia-northeast1
c-3-sliw2	34.146.105.57	gcp	asia-northeast1
w-1-abj4c	13.115.139.255	aws	ap-northeast-1
w-2-n5iwj	54.249.37.67	aws	ap-northeast-1
w-3-1fp3m	18.182.27.247	aws	ap-northeast-1
w-4-cgp00	34.89.19.147	gcp	europe-west2
w-5-7f7dx	35.230.154.106	gcp	europe-west2
w-6-z6gpz	34.89.9.11	gcp	europe-west2

<CSP 리전별 노드 조회>

④

```
~$ kubectl get po -o custom-columns='NAME:.metadata.name,IP:.status.podIP,NODE:.spec.nodeName'
```

NAME	IP	NODE
busybox-646f587557-89b25	10.244.8.3	w-5-7f7dx
busybox-646f587557-p2725	10.244.3.3	w-6-z6gpz
busybox-646f587557-sqxt9	10.244.7.2	w-4-cgp00

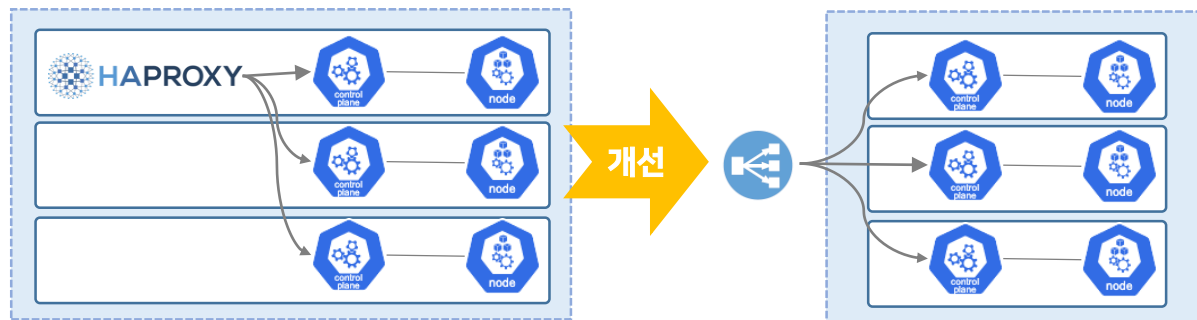
<배포된 Pod 정보 확인>

MCKS 활용 예시

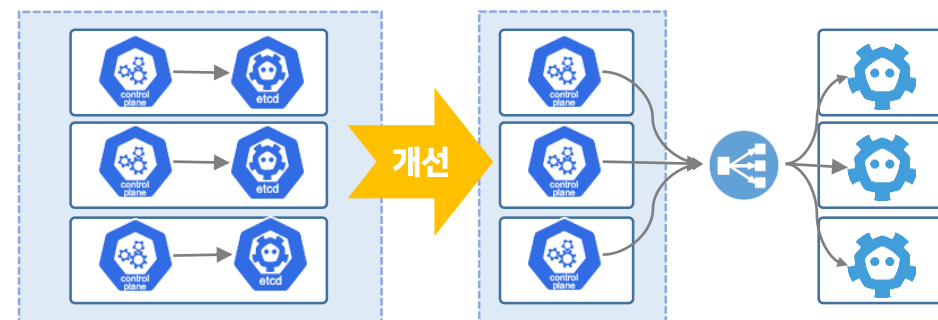


MCKS 향후 계획

- 국내 CSP 추가 지원
 - NCP, NHN Cloud, KT Cloud 등 수요 파악 후 선정
- 스토리지 클래스 지원
 - storageclass 옵션 추가 및 옵션에 따른 설치 및 storageclass 생성 지원
- CB-Spider NLB(Network Load Balancer) 적용을 통한 구조 개선



<HAProxy를 NLB로 대체하여 Control-Plane HA 개선>



<Stacked etcd 에서 external etcd cluster로 구조 개선>

- 프로비저닝 성능 개선

CMKS 기술 현황 및 향후 계획

- 싱글클라우드 대상
쿠버네티스 클러스터 프로비저닝 및 CCM 기반 로드밸런서 연계 지원 (PoC)

지원 CSP



지원 예정 CSP



Google Cloud



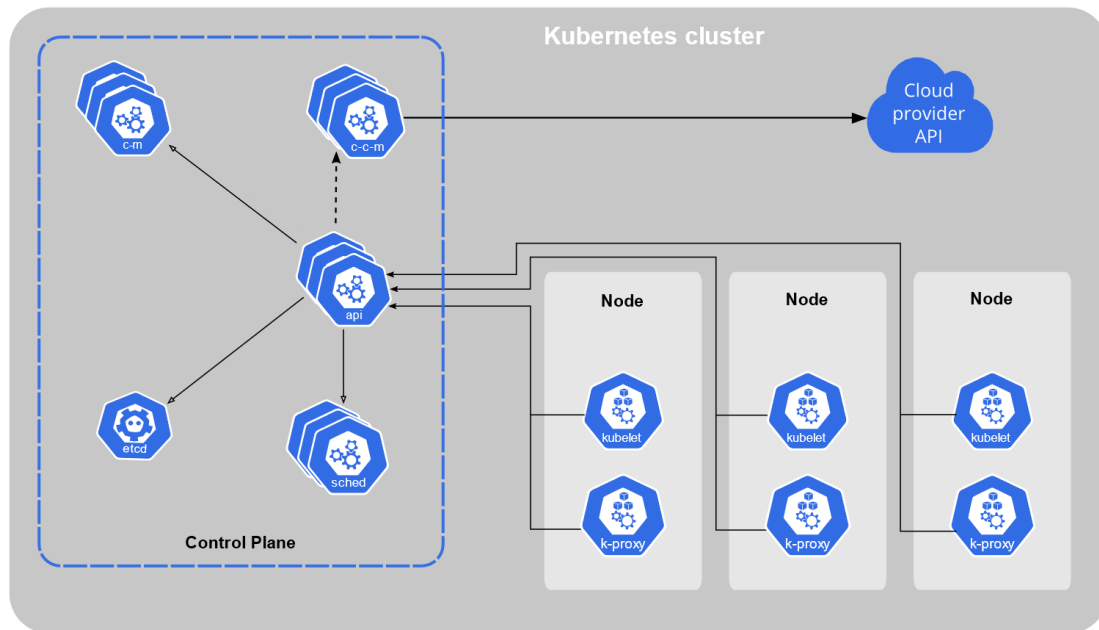
Microsoft Azure

국내 주요 CSP

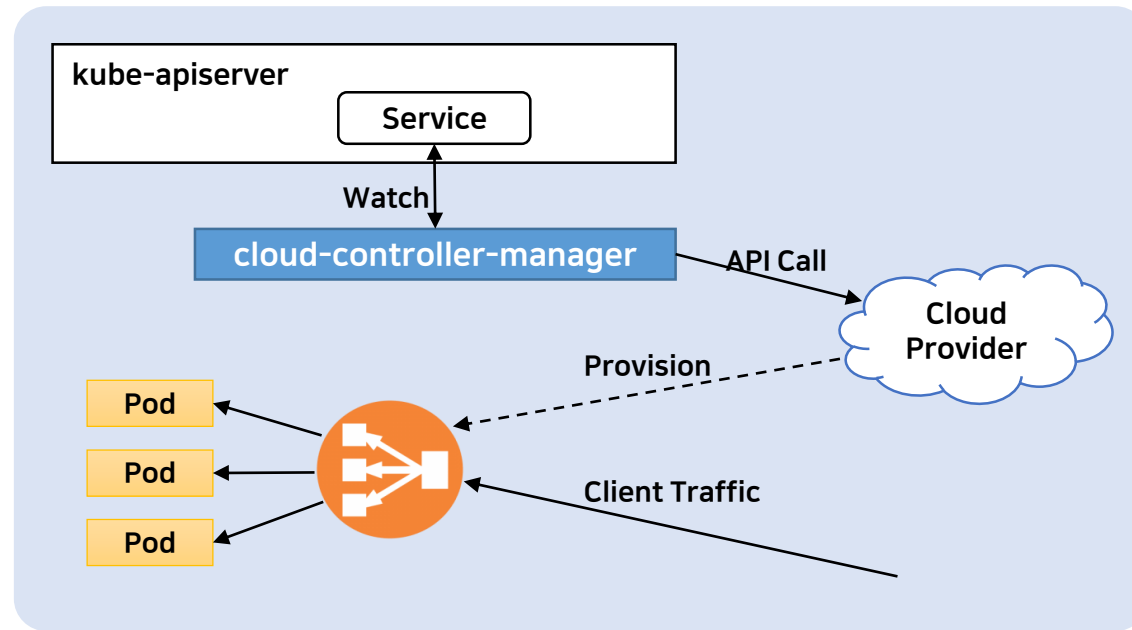
- 지원 쿠버네티스 버전: 1.23.y
- 향후 계획
 - 클라우드 특화 CSI 드라이버 설정 지원
 - 클라우드 특화 CNI 드라이버 설정 지원

(참고) 쿠버네티스 CCM(Cloud Controller Manager) 개요

- 쿠버네티스 클라우드 컨트롤러 매니저
 - 클라우드별 컨트롤 로직이 포함된 쿠버네티스 컨트롤 플레인 컴포넌트
 - 쿠버네티스와 구분된 클라우드 컨트롤러 매니저를 통해 클러스터를 클라우드 공급자의 API에 연결



<쿠버네티스 클러스터 구성>



<CCM 기반 로드밸런서 연계 흐름도>

CB-Ladybug 개발 로드맵

~ 에스프레소 '2020.11.

- 공인IP 기반 쿠버네티스 프로비저닝, 데이터 통신 / 성능, 가용 CNI 검증
- MCKS 규격 정의
- 단일 클라우드 대상 MCKS 프로토타입 개발
- AWS, GCP 지원(총 2종)

카페모카 '2021.6.

- MCKS 멀티클라우드 모델 확장: 단일 쿠버네티스 on 3 CSPs
- MCKS 컨트롤플레인 HA 구성 지원
- MCKS 멀티클라우드 네트워크 플러그인 지원
- MCAS 설계
- Azure 추가 지원(총 3종)

아포가토 '2021.11.

- MCKS 자원 롤백 등 기능 안정화
- 쿠버네티스 Addon 관리 지원
- MCAS 싱글 클러스터 모델 기반 MC-App 프로비저닝/제어 (개념검증)
- Alibaba, Tencent 추가 지원 (총 5종)

카페라떼 '2022.6.

- 지원 쿠버네티스 버전 확대 (1.23)
- CLI 기능 및 구조 개선
- 클라우드바리스타 관리형 쿠버네티스 서비스 (개념검증)
- Openstack, IBM, CLOUDiT (총 8종)

중장기 로드맵

- 컨트롤플레인-노드간 **통신 보안성** 강화
- 클러스터 단위/노드 단위 **동적 최적 배치**
- **부하 기반** 클러스터 단위/노드 단위 **오토스케일링**
- 국내외 CSP & 리전 지원 확대

클라우드 바리스타들의 여섯번째 이야기

“오픈소스”로 만들어가는 “멀티클라우드” 생태계

Cloud-Barista Community the 6th Conference

감사합니다.

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

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

(김 수 영 / contact-to-cloud-barista@googlegroups.com)