
Cloud Barista
전세계 클라우드를 내 손안에, 멀티 클라우드
클라우드바리스타 커뮤니티 제11차 컨퍼런스
One Service for Multi-Cloud
이종 클라우드 서비스를 하나의 서비스로 관리하기
메인테이너@클라우드바리스타
손 석 호
얼그레이 (Earl Grey) 한잔 어떠세요 ?

What is your Github handle?

@seokho-son

@seokho-son

@seokho son (CNCF / K8s Slack)

Ph.D. in Computer Science,

Senior Researcher & **Special Fellow**, ETRI

Ambassador, CLOUD NATIVE COMPUTING FOUNDATION

Maintainer, Cloud-Barista CB-Tumblebug

Maintainer, M-CMP Platform

Maintainer, CNCF Cloud Native Glossary

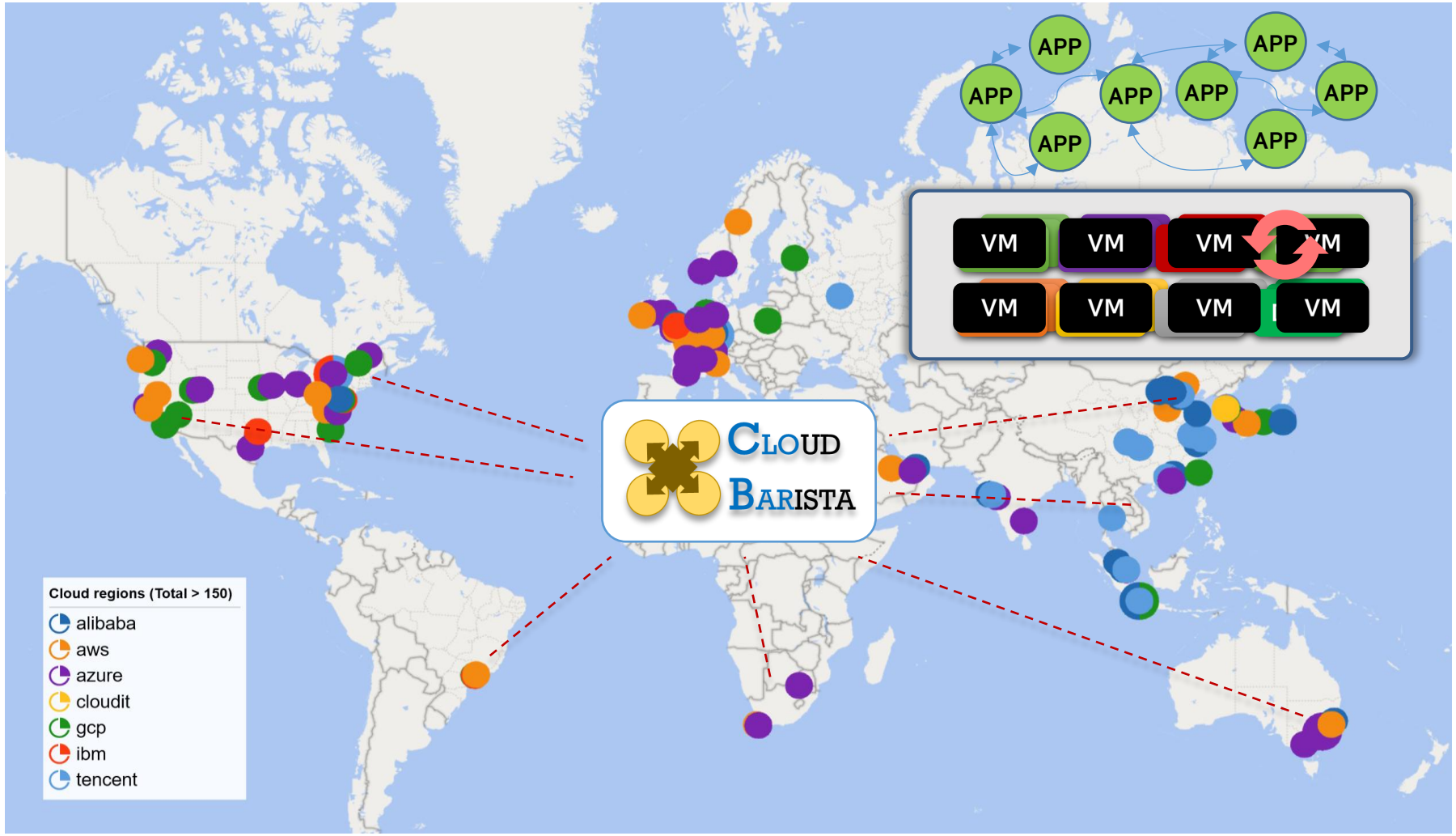
Maintainer, Kubernetes Website

Lead, Kubernetes SIG-Docs Subproject

Lead, Kubernetes SIG-Docs Korean l10n Team

목 차

Manages Multi-Cloud Infrastructures !



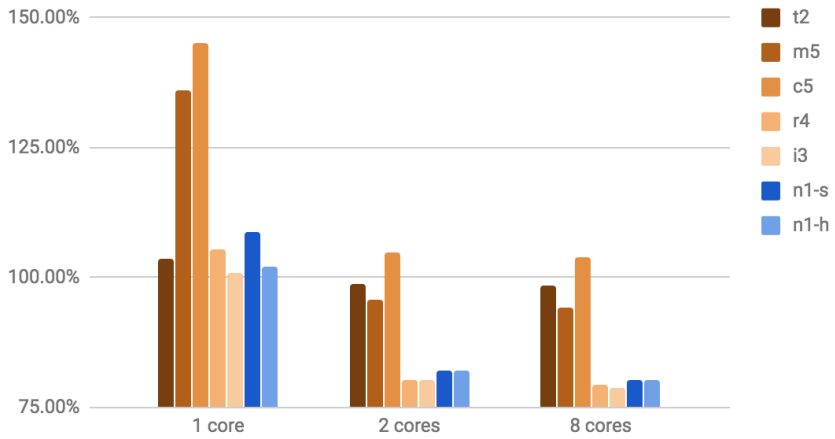
Multi-Cloud Application

Multi-Cloud Computing Infrastructure

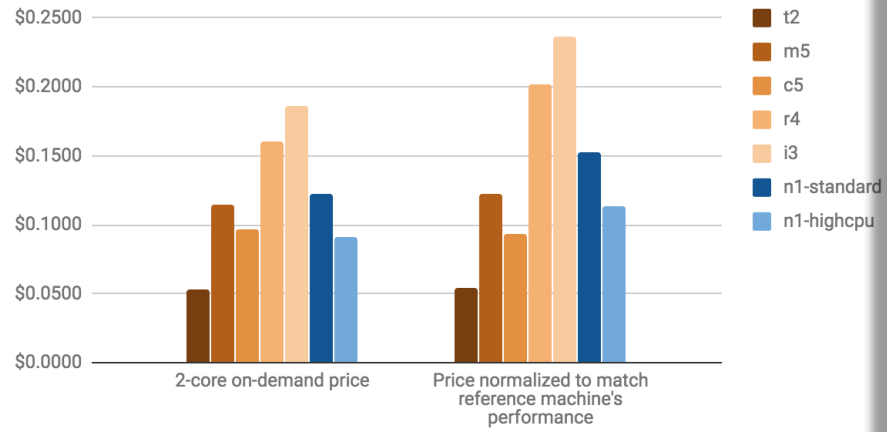
computing resource | computing resource | computing resource

Heterogeneous Clouds abstraction

Cloud performance varies more than you think

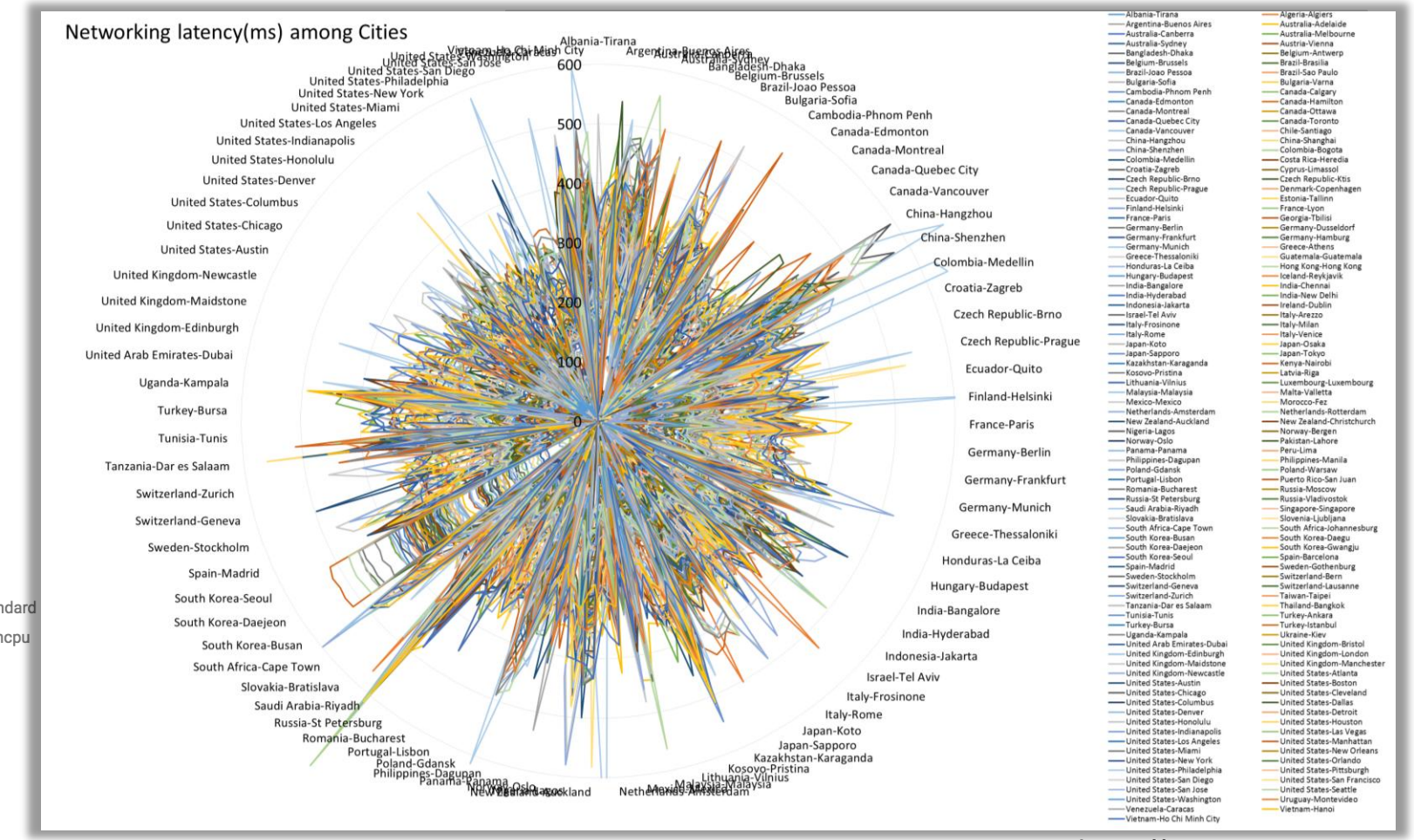


VM performance (AWS vs GCP)



VM price (AWS vs GCP)

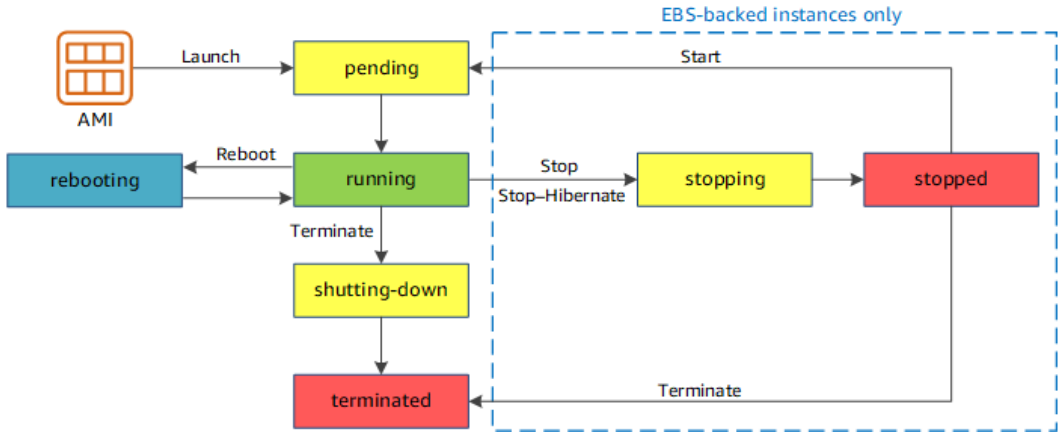
(https://medium.com/infrastructure-adventures/aws-vs-gcp-vs-on-premises-cpu-performance-comparison)



GPU performance and cost comparison (AWS vs GCP)

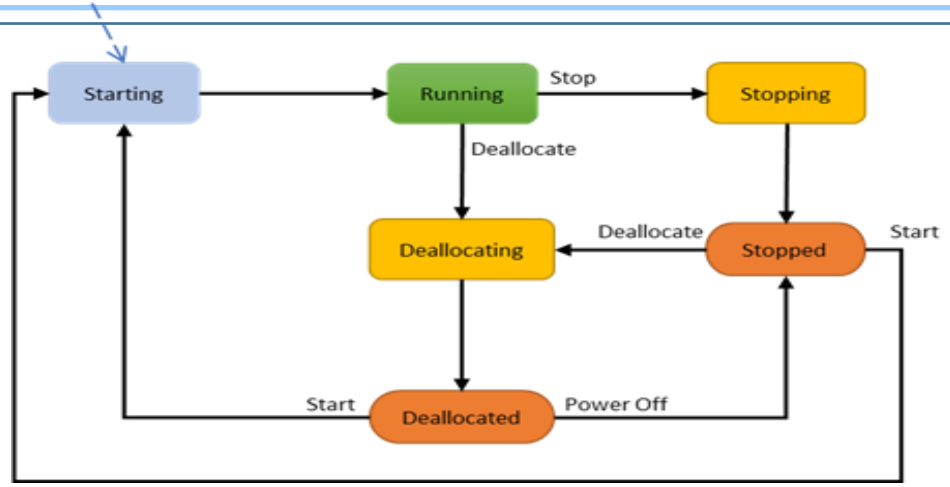
(https://towardsdatascience.com/maximize-your-gpu-dollars-a9133f4e546a)

Different Terms, APIs, and Methods Across Clouds



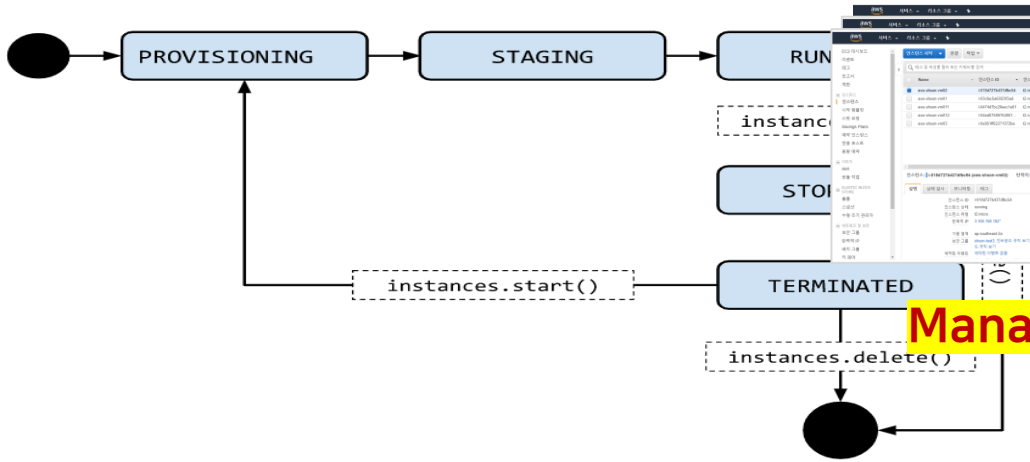
docs.aws.amazon.com/ko_kr/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html

AWS



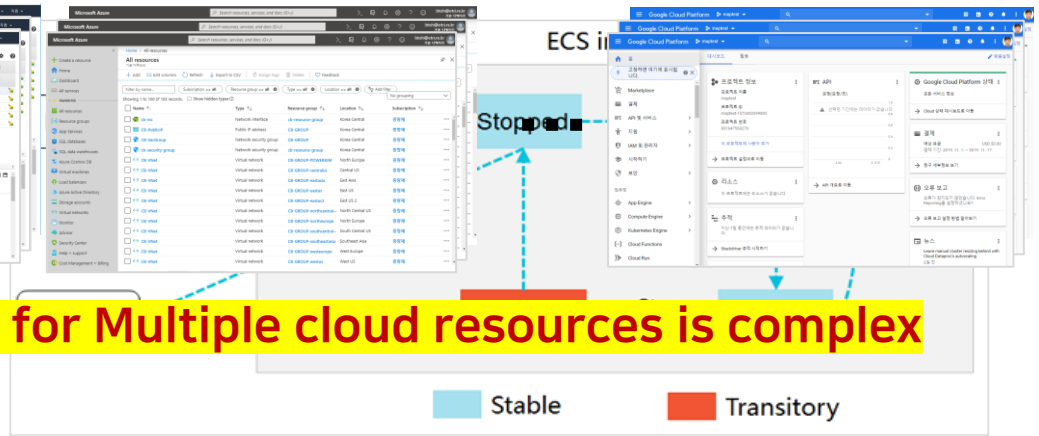
docs.microsoft.com/en-us/azure/virtual-machines/linux/states-lifecycle

Azure



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

GCP

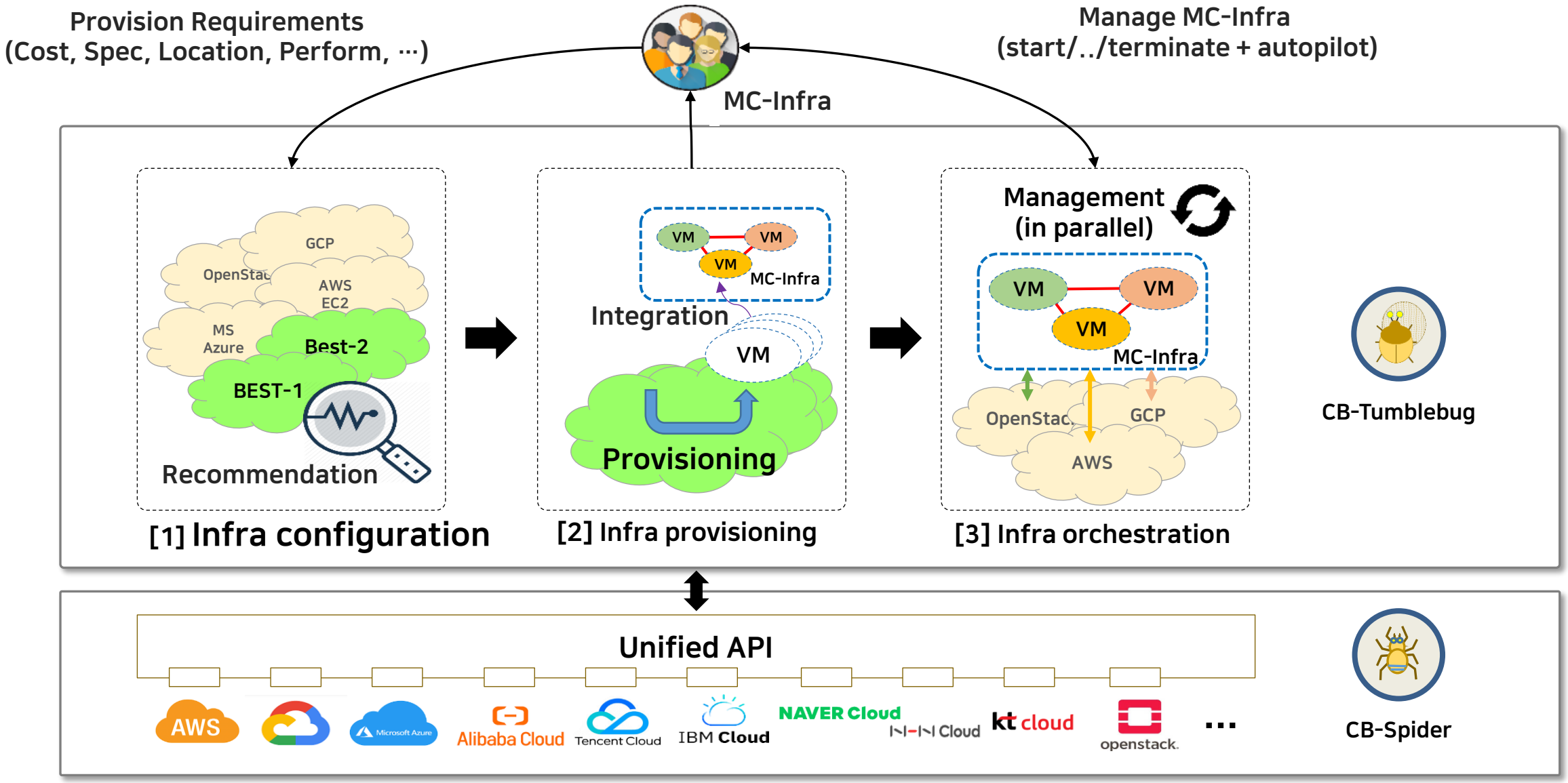


Management for Multiple cloud resources is complex

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

AlibabaCloud

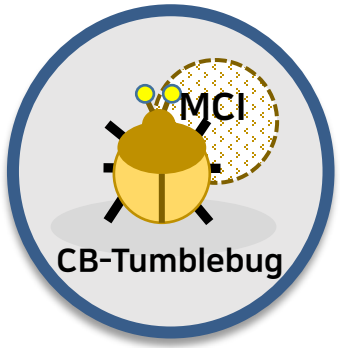
Supported Main Features



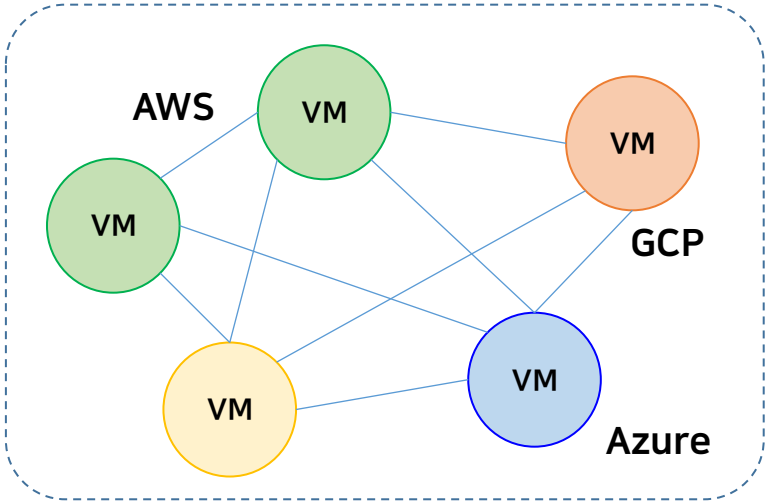
CB-Tumblebug 기본 기능 요약

멀티클라우드 인프라를 ..

최적으로 [구성]→[생성]→[설정]→[관리]→[자동제어]



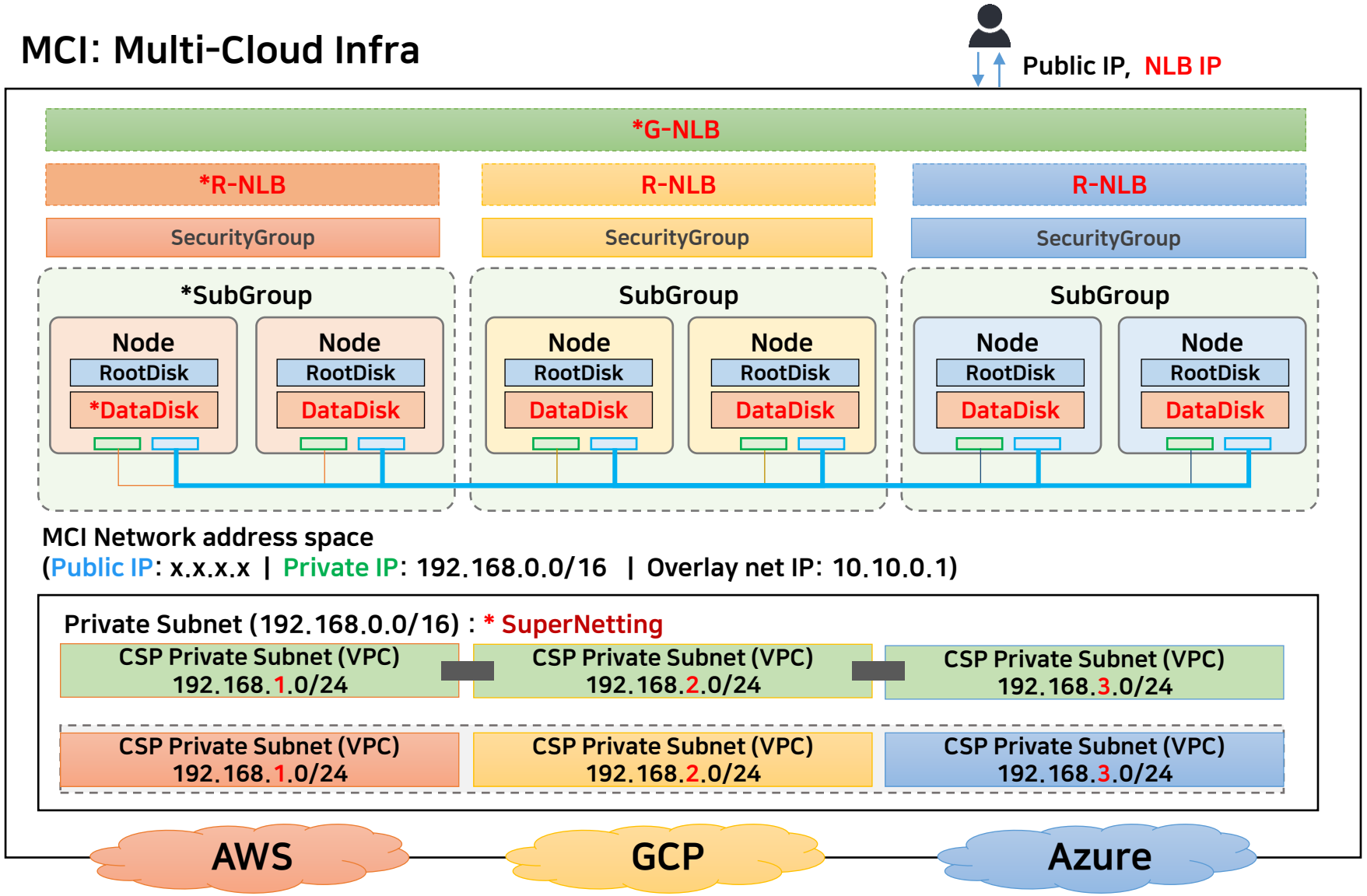
- MC-Infra 최적 배치
  - 클라우드 자원의 평가 및 정보 수집을 통해 최적 MCI를 구성
- MC-Infra 프로비저닝 및 특화
  - 다양한 클라우드 자원을 활용하여 MCI를 생성하고 특화
- MC-Infra 라이프사이클 제어 및 관리
  - MCI 라이프사이클 상태를 종합적으로 관리, 통합 제어
- MC-Infra 정책 기반 자동 제어
  - MCI를 진단하고 결과에 따라 자동 제어하는 운용 자동화



멀티 클라우드 인프라 (MCI)

Structure of the Multi-Cloud Infra

MCI: Multi-Cloud Infra



*** NLB**
Network Load-Balancer (L4)
- G-NLB: Global (SW-based)
- R-NLB: Regional (CSP service)

*** SubGroup (Node Group)**
Node and resource groups with identical attributes

*** DataDisk**
Disk Volume attached to a node

*** Site-to-Site VPN**
Secure tunneling through CSP VPN services

*** SuperNetting**
For greater use of IP address space, routing table simplification, and tunneling.

Supported CSPs and Resource Types

Supported Resource Types

Resource Type	AWS	Azure	GCP	Alibaba	Tencent	IBM	NCP	NHN	KT	Open Stack
VM (CPU)	YES	YES	YES	YES	YES	YES	YES	Partial	Partial	YES
VM (GPU)	YES	YES	YES	YES	WIP	TBD	TBD	TBD	TBD	TBD
Virtual Network	YES	YES	YES	YES	YES	YES	YES	Partial	Partial	YES
Security Group	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
SSH key	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Volume (Disk)	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
NLB (L4 LB)	YES	YES	YES	YES	YES	YES	YES	WIP	Partial	YES
Kubernetes	YES	YES	YES	YES	YES	TBD	TBD	YES	NA	NA
AWS-site VPN	YES	YES	YES	YES	YES	YES	TBD	TBD	TBD	TBD

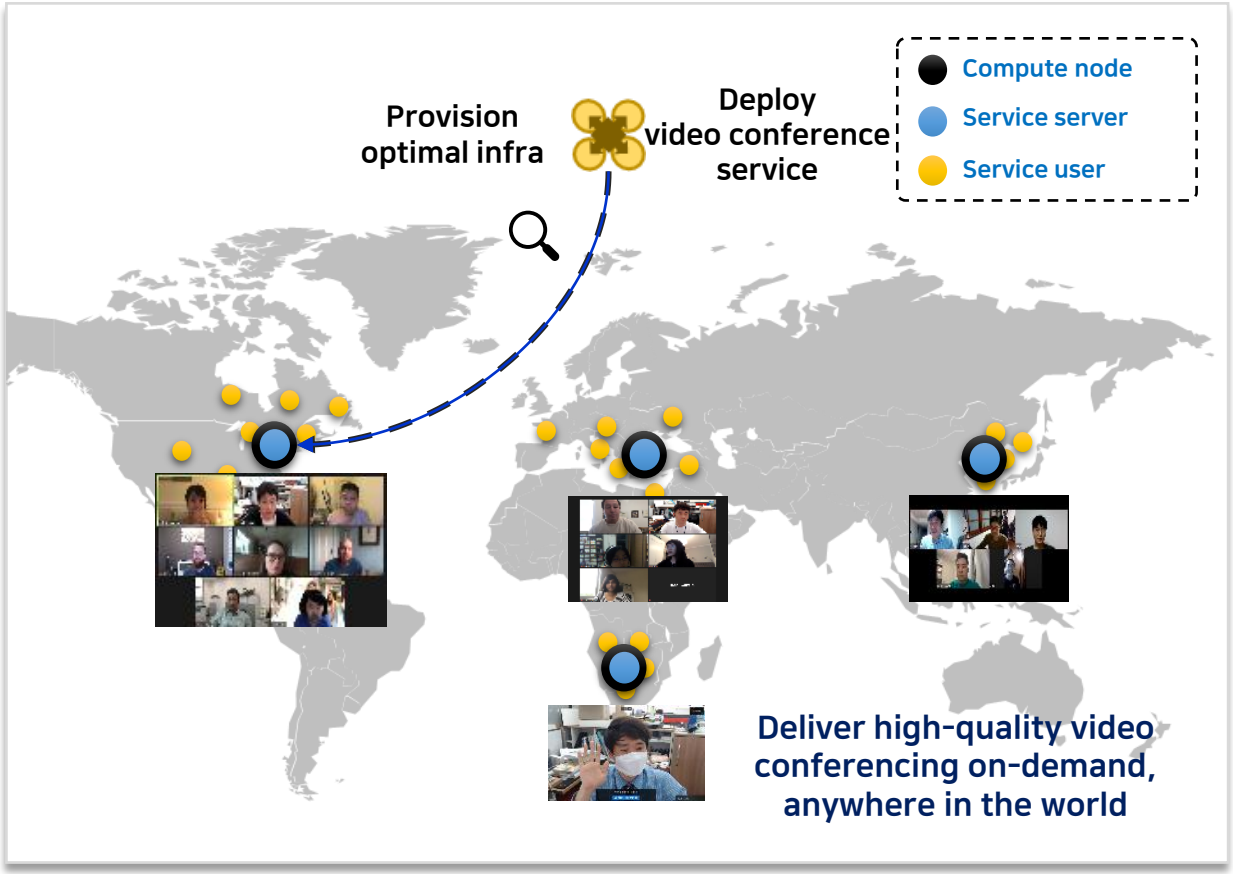
(Note: reference only, and we do not guarantee its functionality. It is regularly updated.)

[Live doc: CB-TB Supported Resource Types]

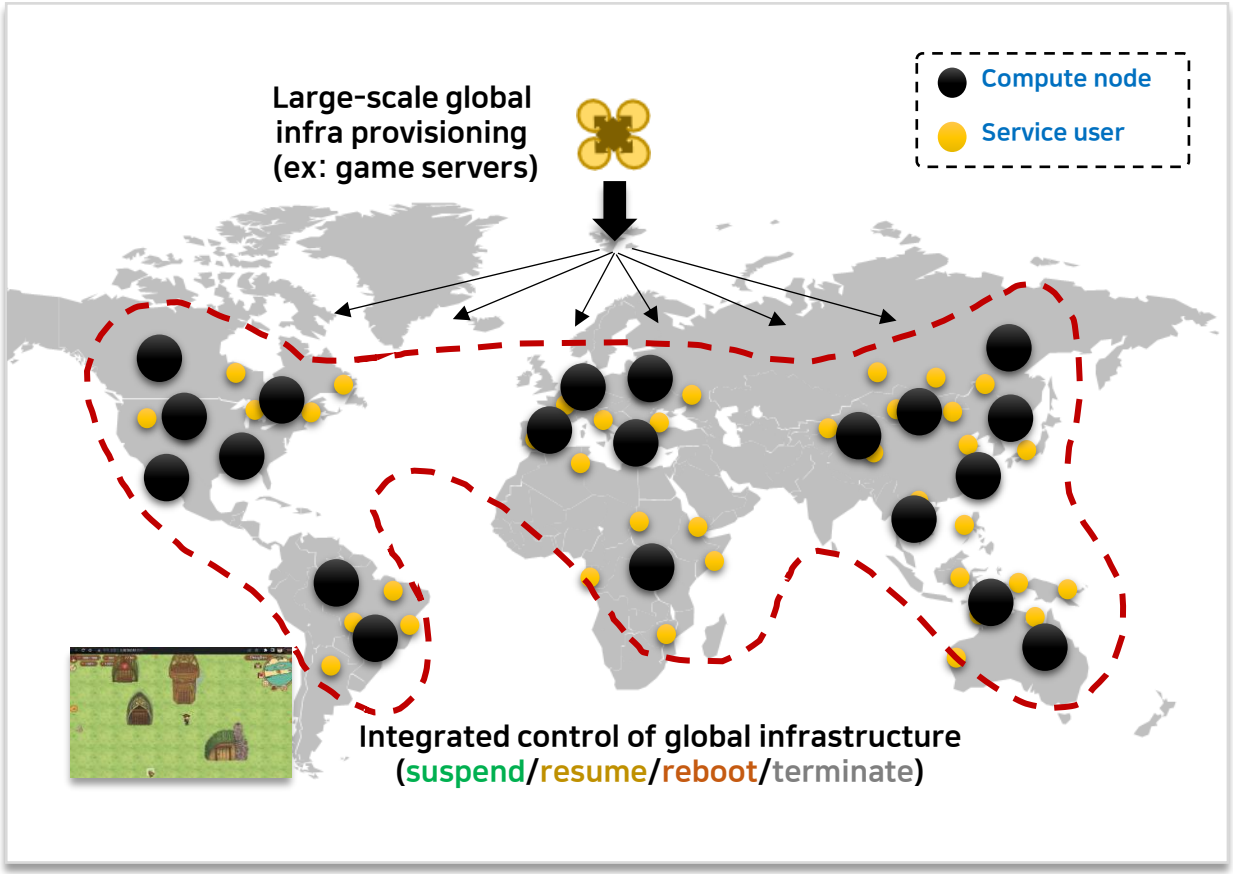
REST API: 200 - 4
(POST:66 GET:75 PUT:14 DEL:42)

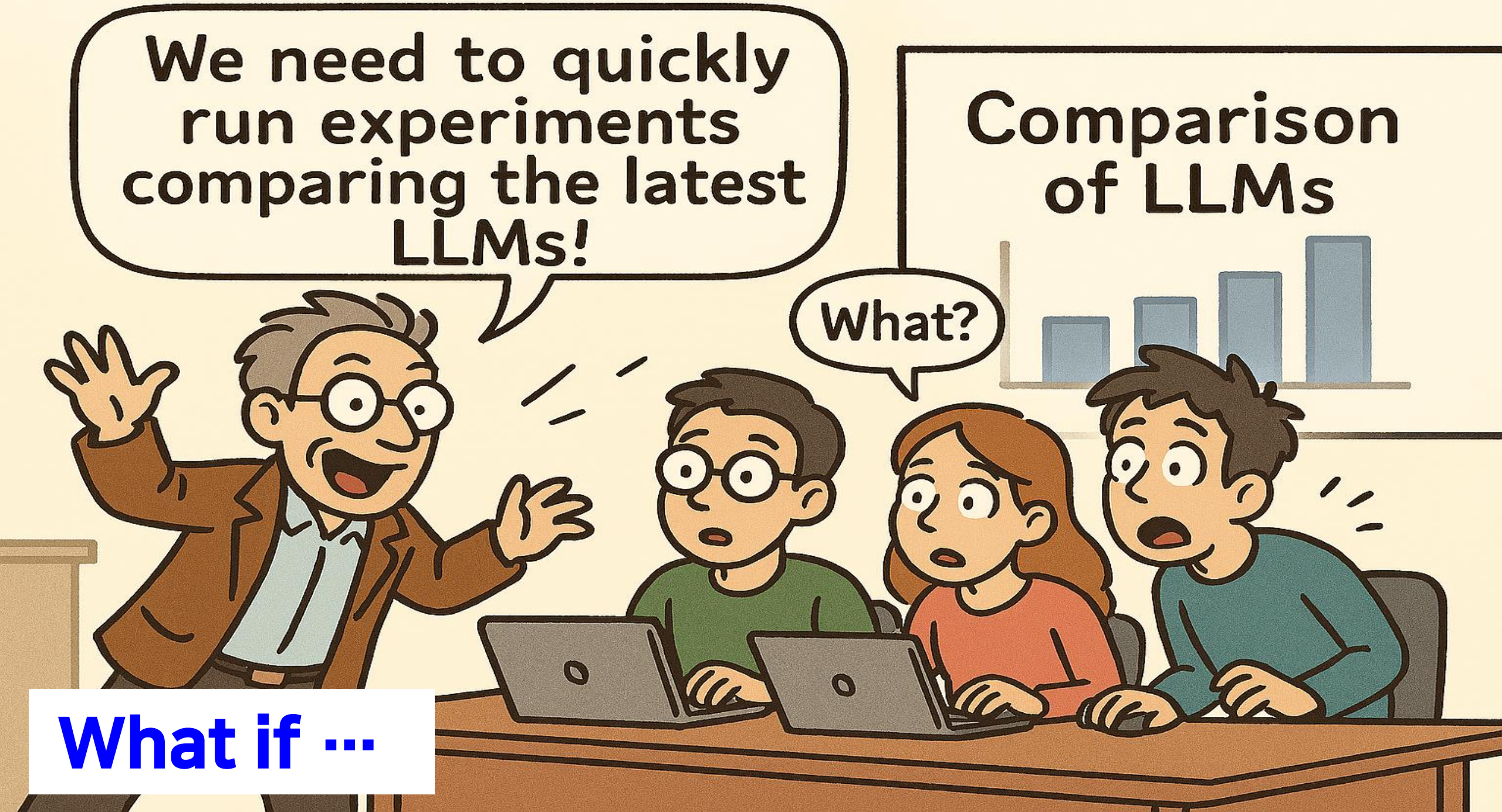
CB-TB Use cases

<멀티 클라우드 기반 최적 비디오 컨퍼런스 서비스>



< 멀티 클라우드 기반 대규모 글로벌 게임 서비스>




We need to quickly run experiments comparing the latest LLMs!
Comparison of LLMs
What?

What if ...

shson@DESKTOP-KTDT15E: ~,

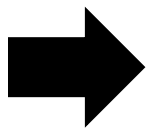
```
shson@DESKTOP-KTDT15E:~/go/src/github.com/cloud-barista/cb-tumblebug$
shson@DESKTOP-KTDT15E:~/go/src/github.com/cloud-barista/cb-tumblebug$
shson@DESKTOP-KTDT15E:~/go/src/github.com/cloud-barista/cb-tumblebug$
shson@DESKTOP-KTDT15E:~/go/src/github.com/cloud-barista/cb-tumblebug$
```

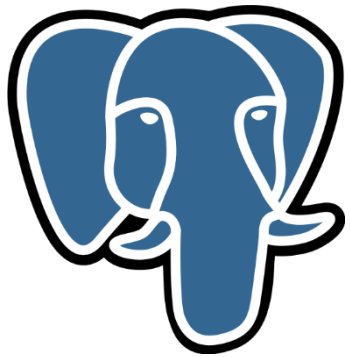
목 차

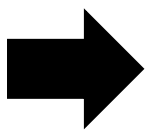
What's Changed

- Add APIs to manage SQL Database by **@yunkon-kim** in #1919
- CHORE: update cloudspec.csv and k8sclusterinfo.yaml by **@sykim-etri** in #1920
- Update CODEOWNERS by **@yunkon-kim** in #1921
- Support SQL DB APIs for AWS, Azure, GCP, and NCP by **@yunkon-kim** in #1924
- CHORE: [k8scluster] update cloudimage.csv for nhncloud by **@sykim-etri** in #1926
- Bump github.com/golang-jwt/jwt/v4 from 4.5.0 to 4.5.1 by **@dependabot** in #1917
- Add dist-lock for safe in-parallel processing of MCI dynamic creation by **@yunkon-kim** in #1927
- Staging v0.10.1 by **@yunkon-kim** in #1929
- Update performance test case by **@seokho-son** in #1934
- Refactor label removal to use DeleteLabelObject function by **@yunkon-kim** in #1931
- Support Object Storage APIs for AWS and Azure by **@yunkon-kim** in #1935
- FEAT: [k8scluster] support k8snodegroupDynamic by **@sykim-etri** in #1930
- [Workflow] Update Swagger REST API doc by **@cb-github-robot** in #1936
- FIX: [k8scluster] API path by **@sykim-etri** in #1937
- [Workflow] Update Swagger REST API doc by **@cb-github-robot** in #1939
- FIX: [k8scluster] update k8scluster infor from CB-SP for ListK8sCluster() by **@sykim-etri** in #1938
- [Workflow] Update Swagger REST API doc by **@cb-github-robot** in #1940
- Enable MC-Infra dynamic provisioning with custom image by **@seokho-son** in #1942
- Bugfix: update distributed-locking mechanism in provisioning by **@yunkon-kim** in #1943
- Fix 1941 and validate naming rule by **@sykim-etri** in #1944
- [Workflow] Update Swagger REST API doc by **@cb-github-robot** in #1945
- Staging v0.10.2 by **@yunkon-kim** in #1946
- Improve shared resource handling mechanism on MCI dynamic creation by **@yunkon-kim** in #1947
- Remove the CSP support for KT Cloud classic by **@seokho-son** in #1948
- Update dependencies by **@yunkon-kim** in #1949
- Update assets/cloudimage.csv and assets/k8sclusterinfo.yaml by **@sykim-etri** in #1952
- Feat support k8scluster container remote command by **@sykim-etri** in #1950
- [Workflow] Update Swagger REST API doc by **@cb-github-robot** in #1953
- CHORE: set tencent's ap-hongkong as service unavailable by **@sykim-etri** in #1954
- CHORE: [k8scluster] update some regions of tencent as unavailable by **@sykim-etri** in #1955
- Add individual VM access info API by **@seokho-son** in #1956
- Staging v0.10.4 by **@seokho-son** in #1957
- CHORE: [k8scluster] prevent incorrect k8scluster version parsing by **@sykim-etri** in #1959
- Fix incorrect screenshots in the README.md by **@hanizang77** in #1960

v0.10.0 → v0.11.0
(6 months)





XORM  GORM

Get Objects
By LabelSelector

Object Labels

Merge with
CSP Tags

Remote Command
By LabelSelector

Are you sure you want to create this MCI?

mc-dfttq (1 node(s))

Usage Period	Estimated Cost
Hourly	$0.0056
Daily	$0.1344
Monthly	$4.1664

(Do not rely on this estimated cost. It is just an estimation using spec price.)

[#1] SubGroup Name	g1 (1 node(s))
Estimated Price(USD/1H)	$0.0056 ($0.0056 * 1)
Spec	aws+ap-south-1+t4g.micro
vCPU	2
Mem(GiB)	1
Accelerator	none
RootDisk(GB)	500 (type:)
Selected Image	ami-0ebae7cd3dacd2216

☐ Hold VM provisioning of the MCI

☐ Deploy a monitoring agent

☑ Add post-deployment commands

Confirm Cancel

Managed Kubernetes

(prototype) Dynamic Provisioning

(prototype) Node Recommendation

Thanks to 💚

@sykim-etri

@hanizang77

Spec ++

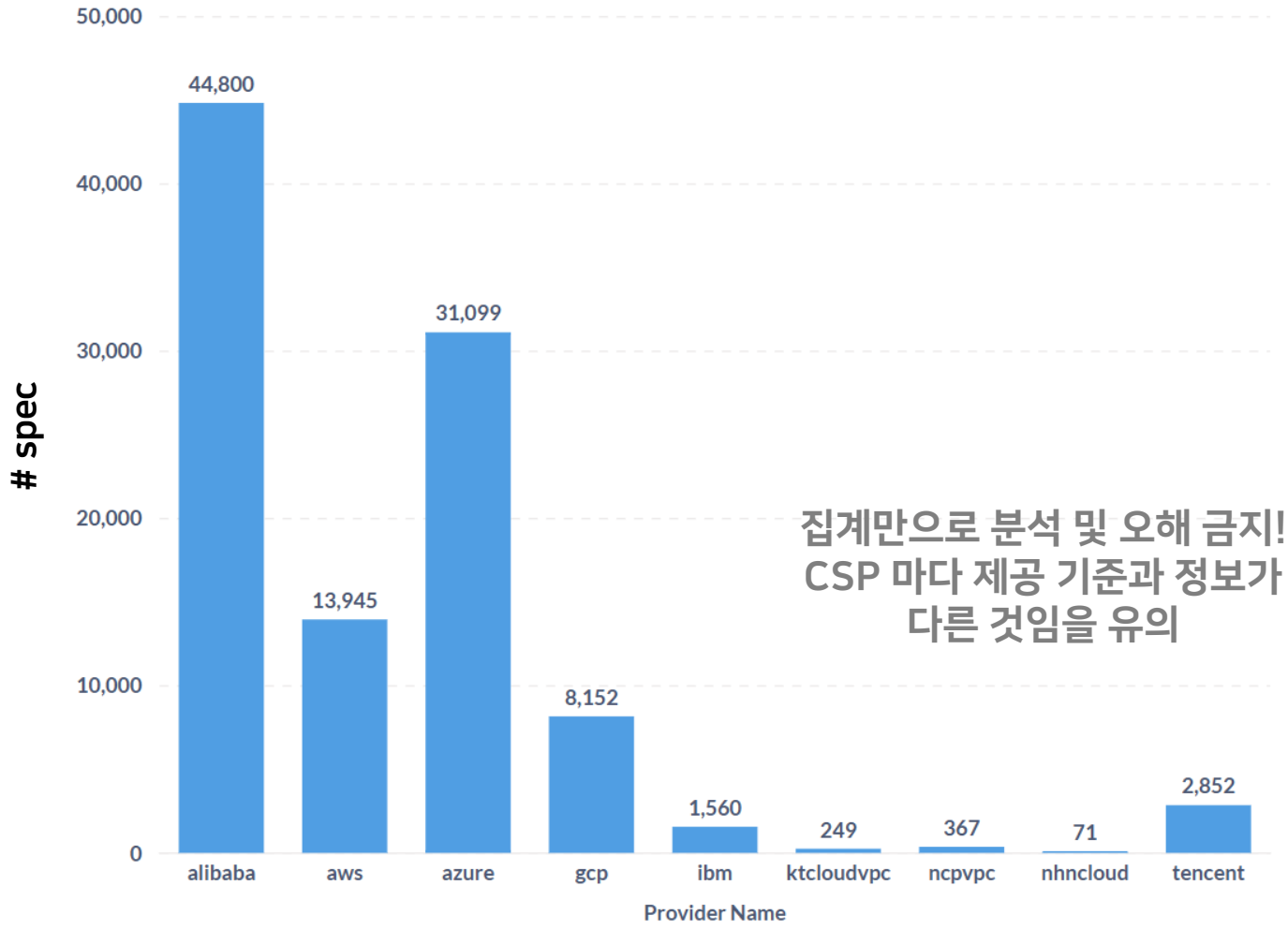
Image ++

인프라 추천 기능 ++

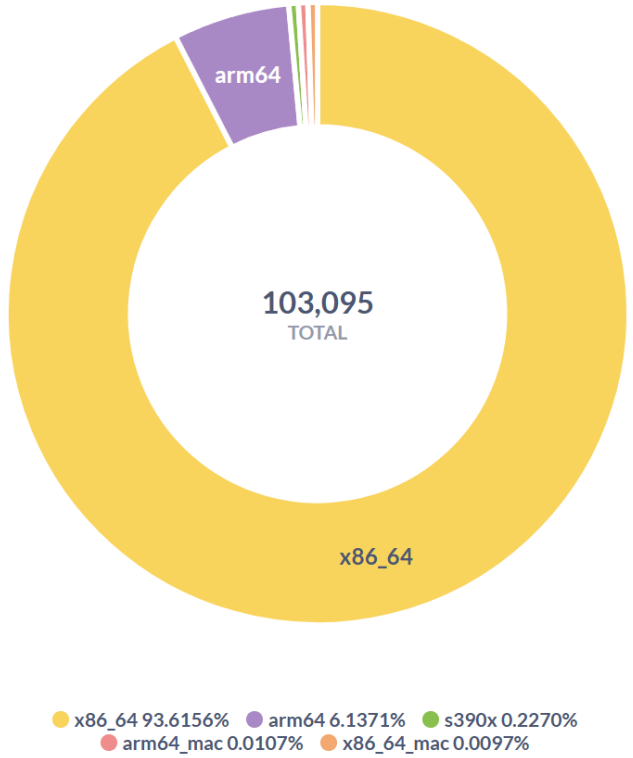
데이터 소스 다각화
- Asset files
- Web docs
- CSP CLI
- CSP API (CB-SP★)

합당한 시간 내에
(10시간 → 10분)
필요한 정보만
(40% 다이어트)
꽉꽉 눌러 담았습니다.
(정제/분류)

인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선



집계만으로 분석 및 오해 금지!
CSP 마다 제공 기준과 정보가
다른 것임을 유의



인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선

Provider Name
aws
Region Name
ap-northeast-2

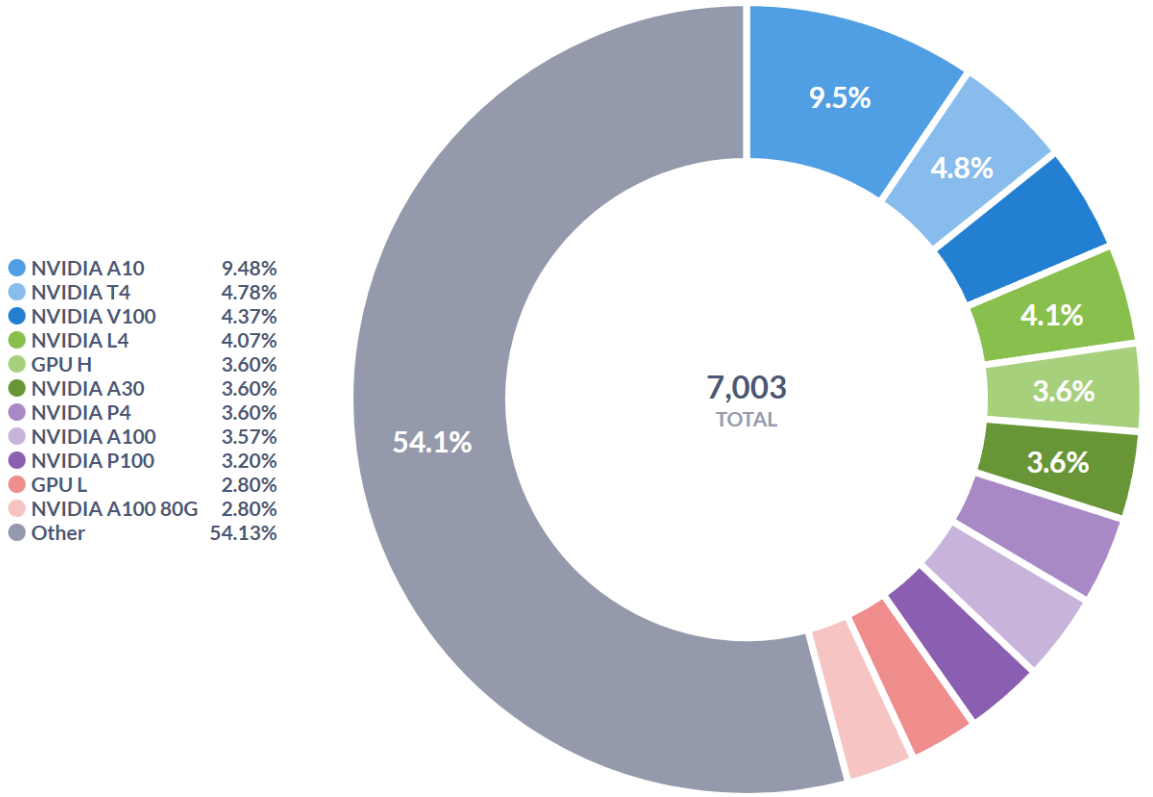
Csp Spec Name
u-24tb1.112xlarge
Cost Per Hour ($)
$263.47 (36만원)

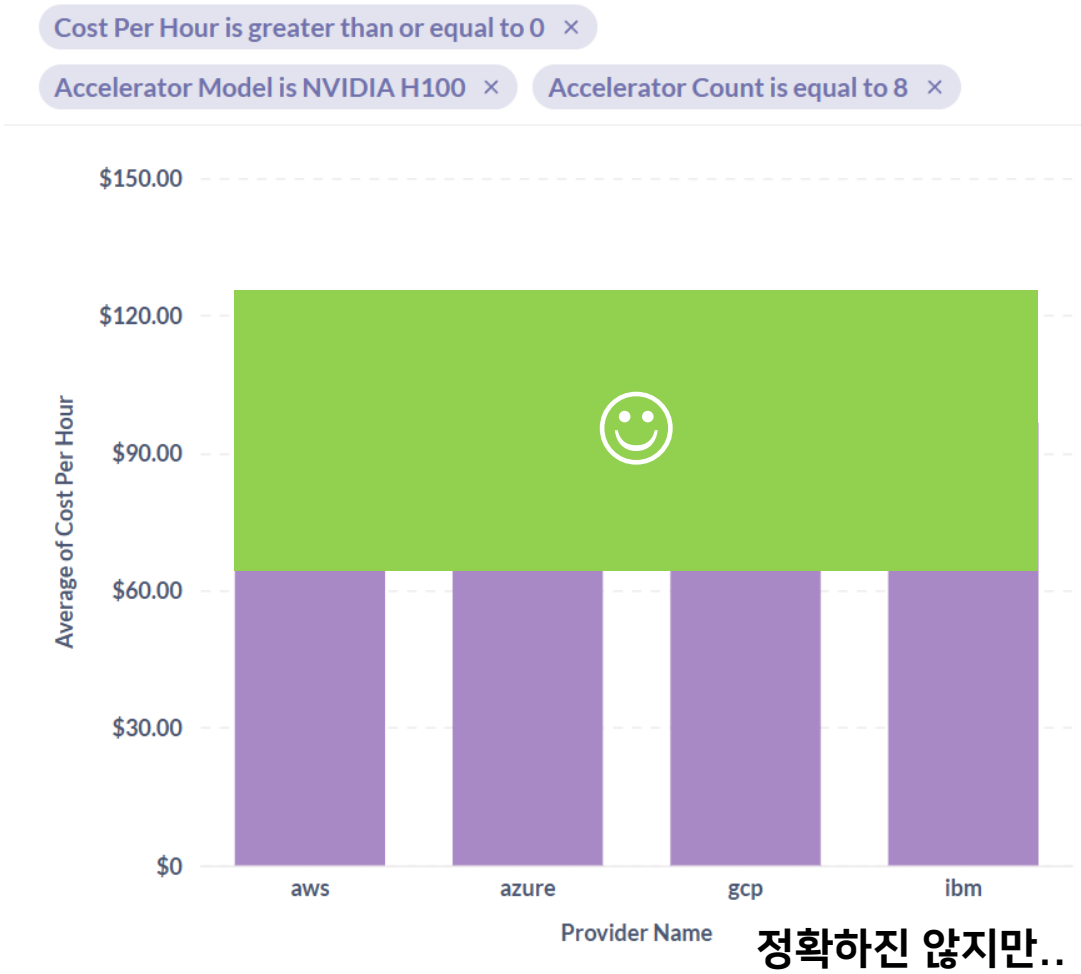
Vcpu 448
Memory 24,576

Architecture
x86_64

No GPU

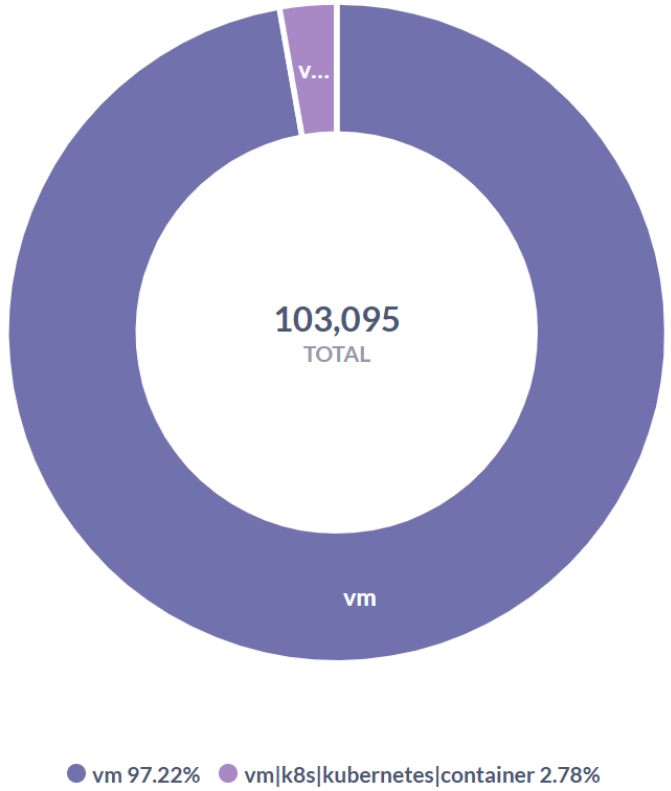
인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선



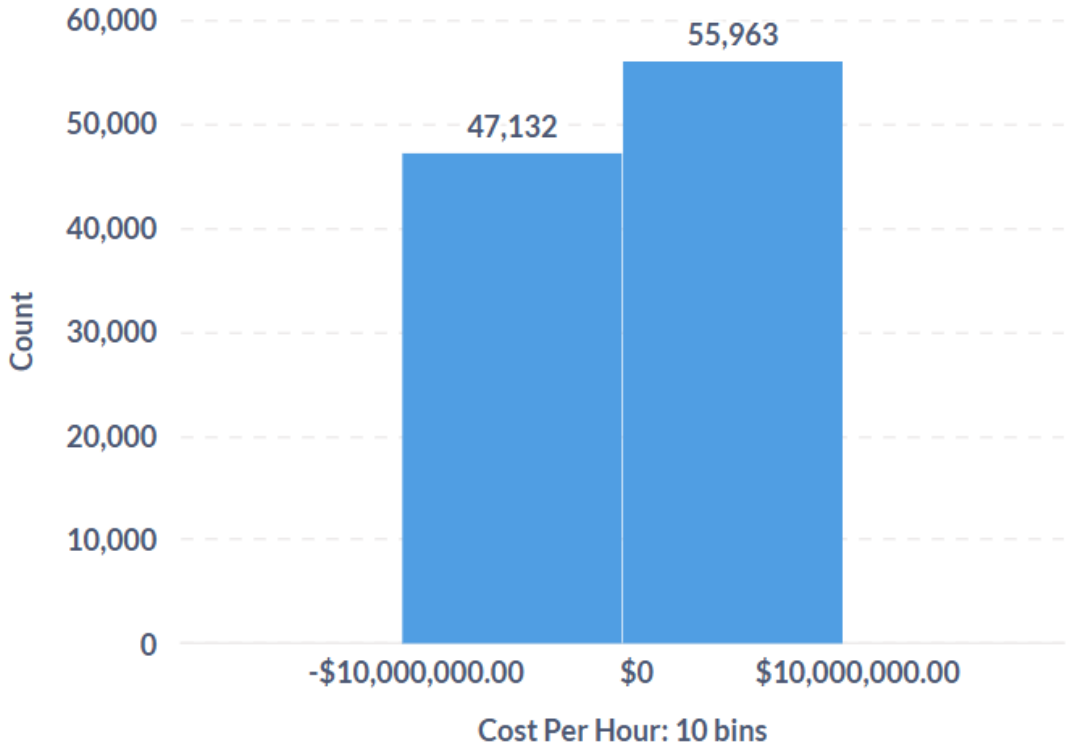


정확하진 않지만..

인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선



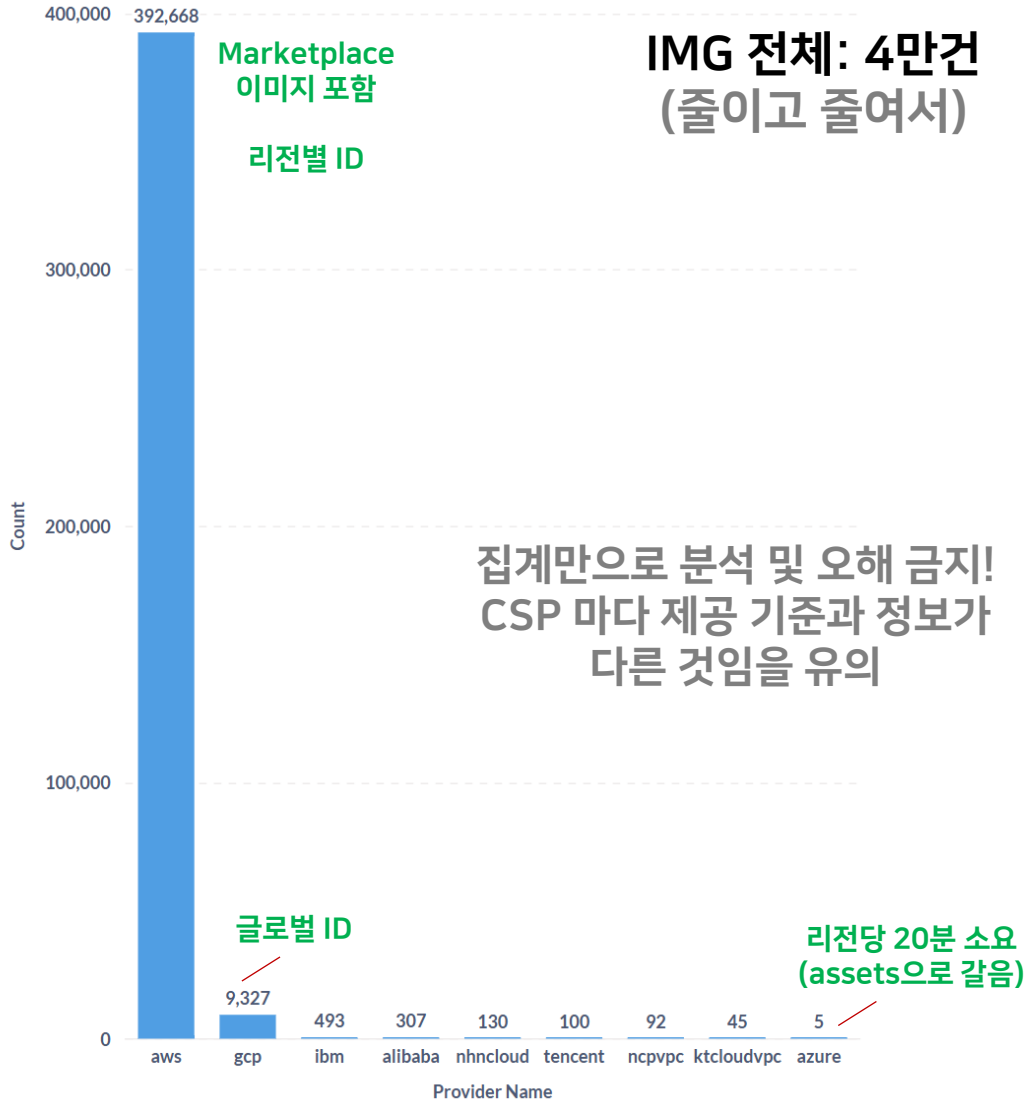
Kubernetes Node 용 Spec 구분 미완



Cost 정보 없는 Spec

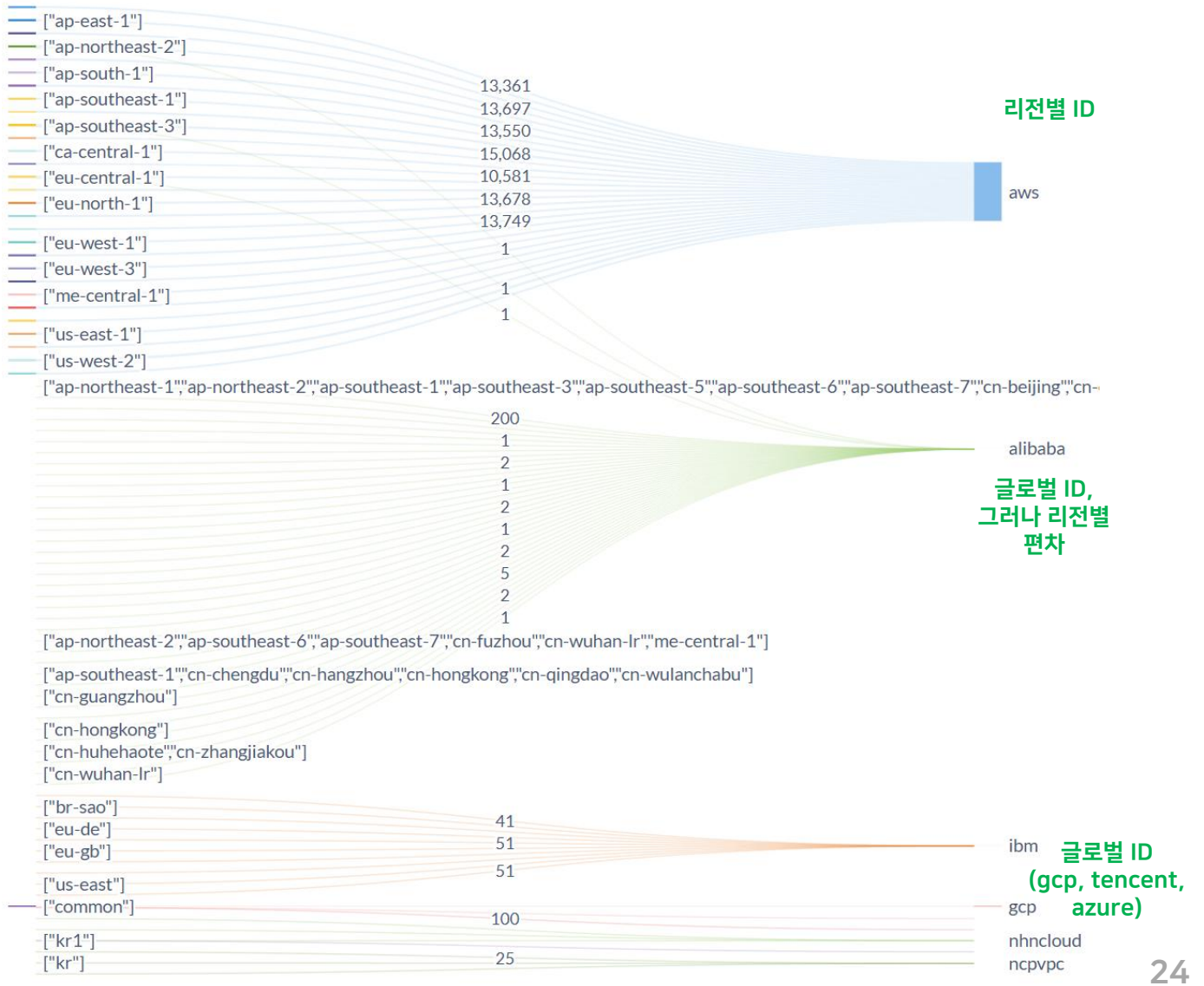
아직 정보 부족..

인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선

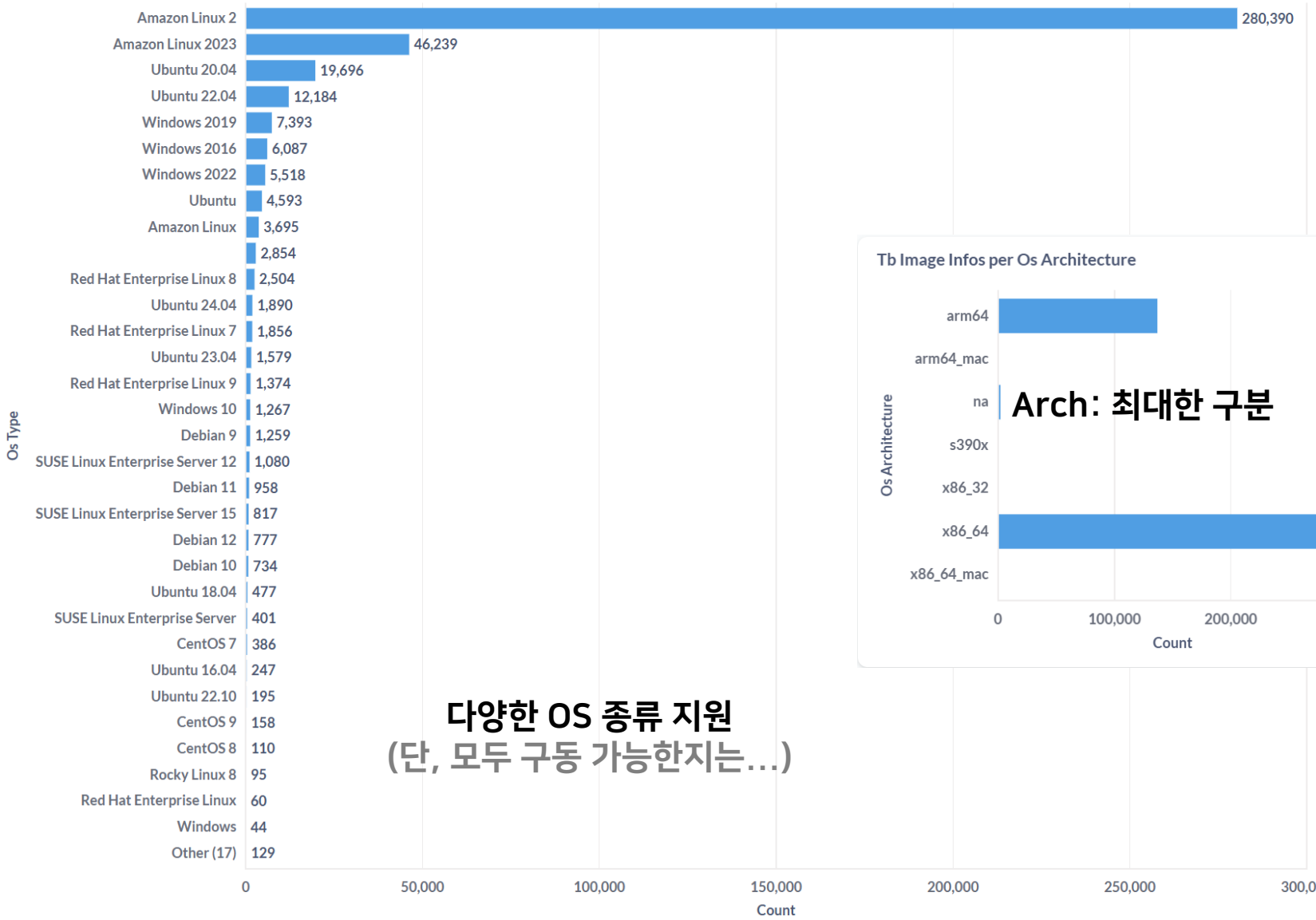


IMG 전체: 4만건
(줄이고 줄여서)

집계만으로 분석 및 오해 금지!
CSP 마다 제공 기준과 정보가
다른 것임을 유의



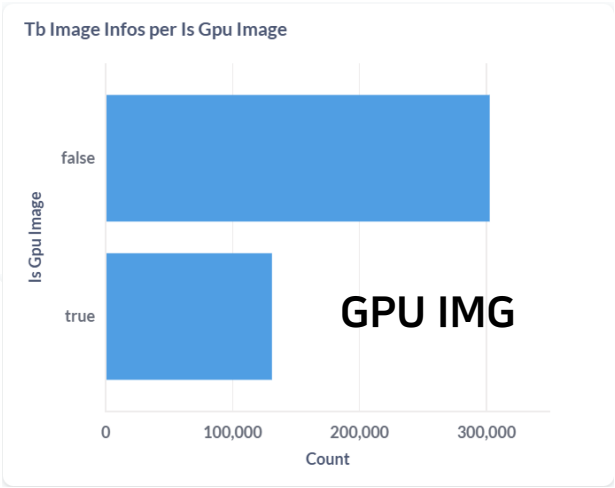
인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선



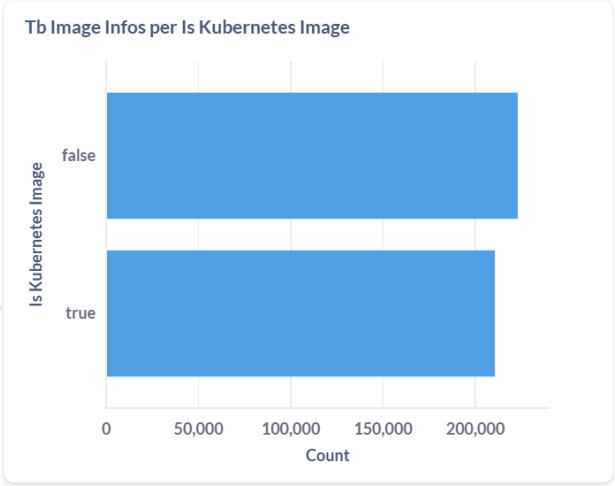
다양한 OS 종류 지원

(단, 모두 구동 가능한지는...)

Arch: 최대한 구분



GPU IMG



K8s IMG

인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선

How to use ?

All you need is ~~Love~~ init.sh

```
shson@DESKTOP-KTDT15E:~/go/src/github.com/cloud-barista/cb-tumblebug$ ./init/init.sh
Detected Python version: 3.12.3
Python version is sufficient.

Checking for uv...

Running the application...
Using CPython 3.10.17
Creating virtual environment at: .venv
Installed 12 packages in 31ms
Current Configuration
Please set the corresponding environment variables to make changes.
 - TUMBLEBUG_SERVER: localhost:1323
```

```
Loading common Specs and Images... (Estimated: 180s)

|                                                  | 100% [1800/1800]

Loading completed (elapsed: 151.35731840133667s)

The system is ready to use.

Initiating price fetching information from all CSPs...
Price for Specs will be updated (it may take around 10 mins).

You can run this procedure in the background using by ctrl+c or ctrl+z.
Price fetching initiated: Fetched 65397 prices (from 172 connConfigs)

Cleaning up the venv and uv.lock files...

Environment cleanup complete.
```

인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선

POST /ns/{nsId}/resources/fetchImagesAsync Fetch images asynchronously

Fetch images in the background without waiting for completion

Parameters Cancel

Name Description

nsId * required Namespace ID

string (path) system

Execute Clear

Responses Response content type application/json

Curl

```
curl -X 'POST' \
  'http://localhost:1323/tumblebug/ns/system/resources/fetchImagesAsync' \
  -H 'accept: application/json' \
  -H 'Authorization: Basic ZGVmYXVsdDpkZWZhdWx0' \
  -d ''
```

Request URL

```
http://localhost:1323/tumblebug/ns/system/resources/fetchImagesAsync
```

Server response

Code Details

202

Response body

```
{
  "message": "Started fetching images in the background. Check server logs for progress."
}
```

Download

Request URL

```
http://localhost:1323/tumblebug/ns/system/resources/fetchImagesResult
```

Server response

Code Details

200

Response body


```
{
  "namespaceId": "system",
  "totalImages": 105350,
  "successCount": 62,
  "failCount": 0,
  "startTime": "2025-04-22T14:41:44.614846318Z",
  "endTime": "0001-01-01T00:00:00Z",
  "totalTime": "",
  "connResults": [
    {
      "connName": "ncpvpc-jpn",
      "provider": "ncpvpc",
      "region": "JPN",
      "imageCount": 20,
      "startTime": "2025-04-22T14:41:44.620781794Z",
      "endTime": "2025-04-22T14:41:45.405642053Z",
      "elapsedTime": "784.860257ms",
      "success": true
    },
    {
      "connName": "ncpvpc-kr",
      "provider": "ncpvpc",
      "region": "KR",
      "imageCount": 25,
      "startTime": "2025-04-22T14:41:45.405643494Z",
      "endTime": "2025-04-22T14:41:45.571221342",
      "elapsedTime": "165.577845ms",
```


Download

POST /ns/{nsId}/resources/searchImage Search image

condition * required

object (body)

condition

Example Value | Model

```
{
  "detailSearchKeys": [
    "sql",
    "2022"
  ],
  "isGPUImage": false,
  "isKubernetesImage": false,
  "osType": "ubuntu 22.04",
  "providerName": "aws",
  "regionName": "us-east-1"
}
```

Deep(?)Search

Is gpu

Is k8s

Architecture

OS (simple)

CSP and Region

Response body


```
{
  "count": 4,
  "image": [
    {
      "namespace": "system",
      "providerName": "aws",
      "cspImageName": "ami-085925f297f89fce1",
      "regionList": [
        "us-east-1"
      ],
      "id": "aws+us-east-1+ami-085925f297f89fce1",
      "name": "aws+us-east-1+ami-085925f297f89fce1",
      "connectionName": "aws-us-east-1",
      "infraType": "vm",
      "fetchedTime": "2025.04.21 10:10:54 Mon",
      "creationDate": "2020-04-09T16:44:13.000Z",
      "osType": "Ubuntu18.04"
```

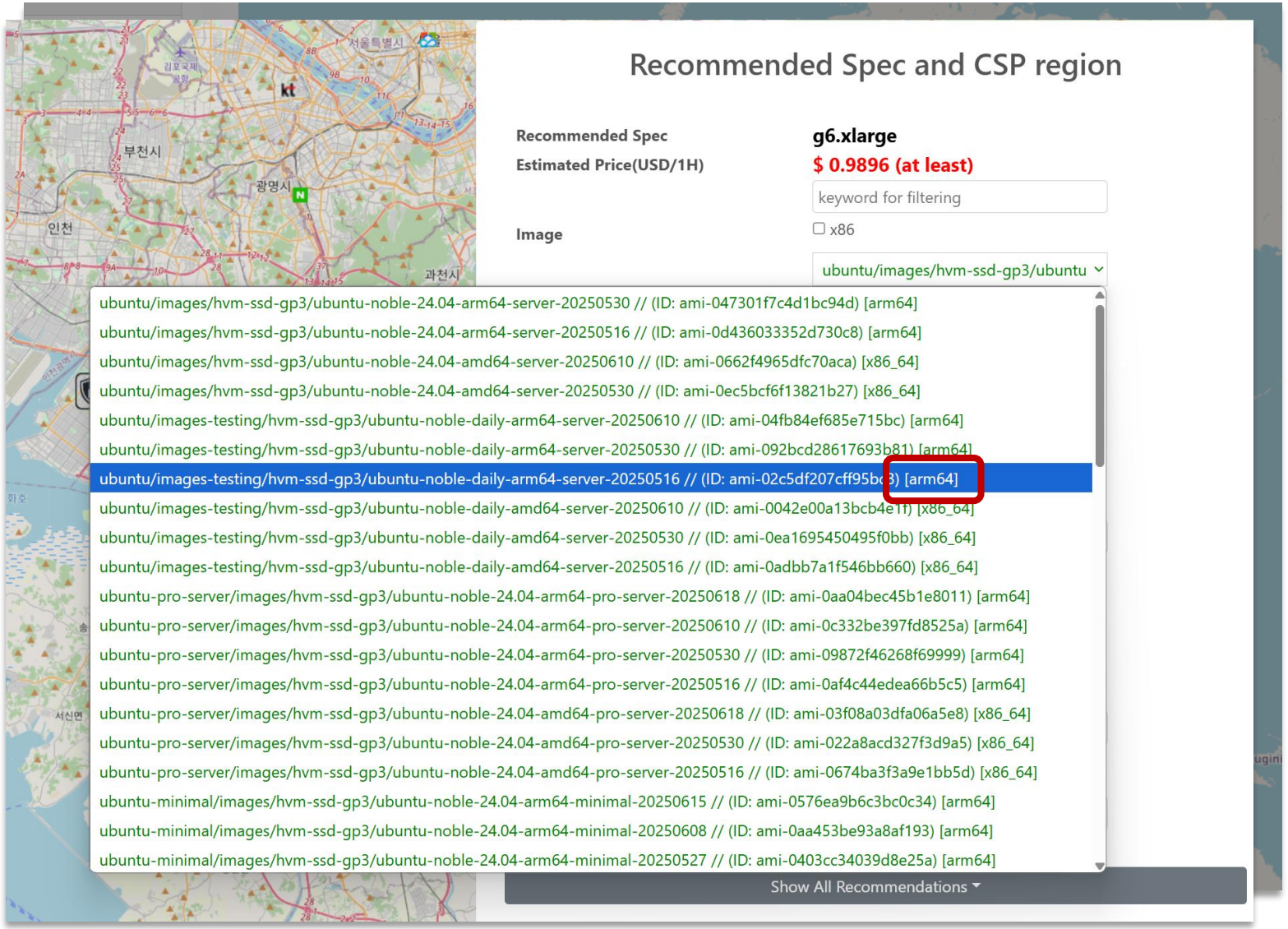
인프라 추천 개선: Spec/Image 처리 방식 및 정보 개선

So what?

Your freedom

but

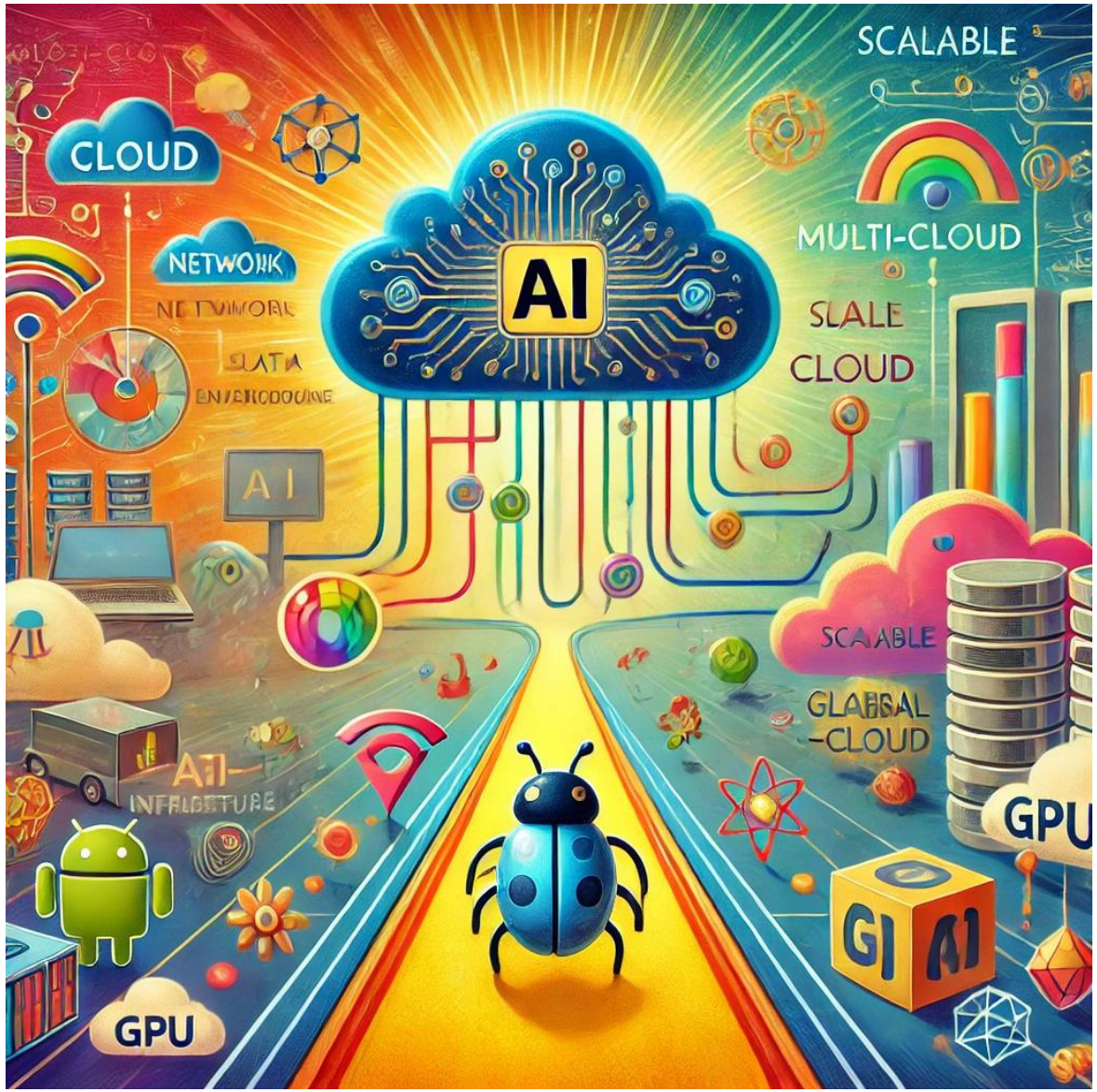
WITH GREAT POWER
COMES GREAT
RESPONSIBILITY



목 차

재미있는 AI 탐구생활

멀티 클라우드 기반 자유로운
AI 인프라를 위한 CB-TB의 여정



GPU 인프라 프로비저닝 지원 확대 (유스케이스: Ray cluster)

CLOUD BARISTA

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

Ollama
LLM Server

n8n
workflow

RAG
Agentic
MCP (Model Context)

Open WebUI
chat UI

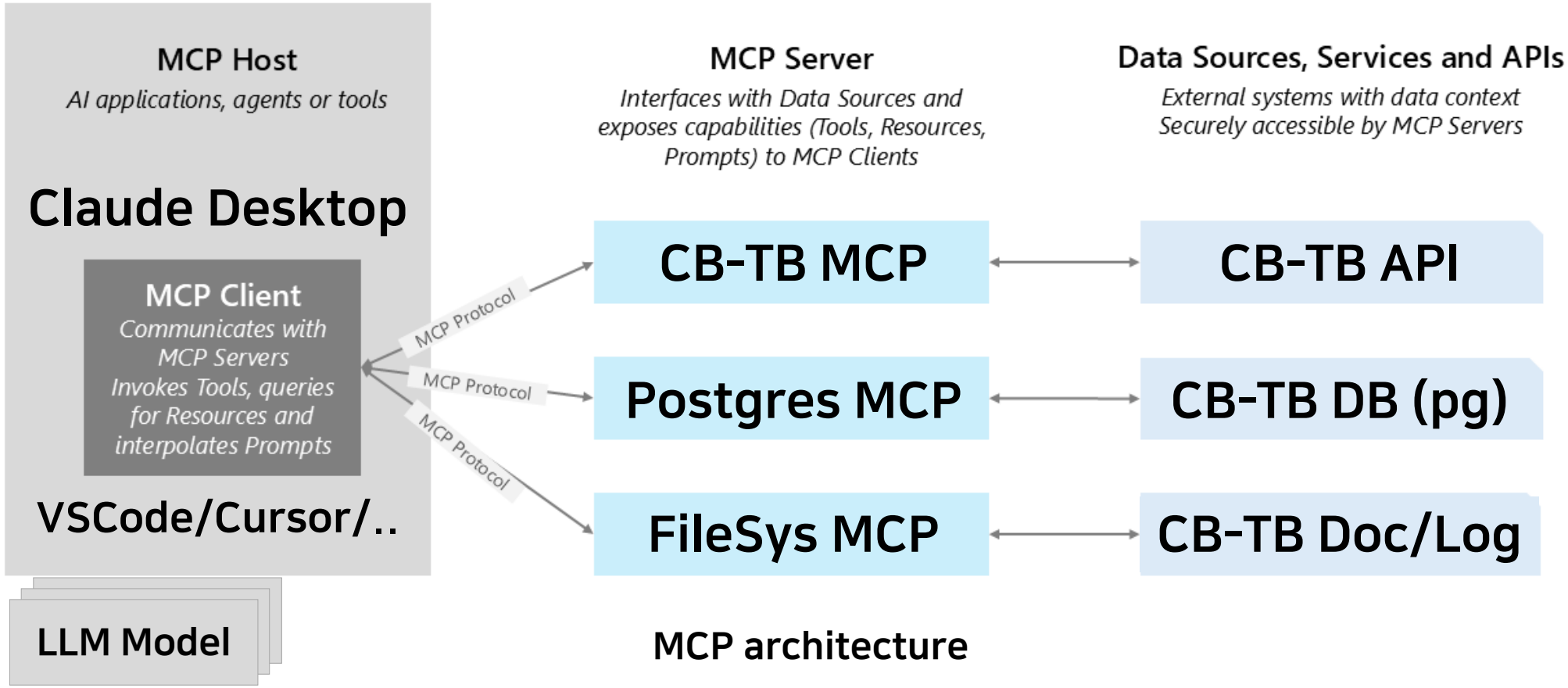
Nomic
embed-text

Ray
machine learning cluster

Llama3.1
LLM+Tools

Qdrant
vector Database

MCP (Model Context Protocol) PoC



MCP architecture

Ref: https://learn.microsoft.com/en-us/azure/api-management/export-rest-mcp-server#test-and-use-the-mcp-server

Claude — Control+Alt+Space

Cloud-Barista

오늘 밤 어떤 생각이 드시나요?

오늘 어떤 도움을 드릴까요?

연구

Claude Sonnet 4

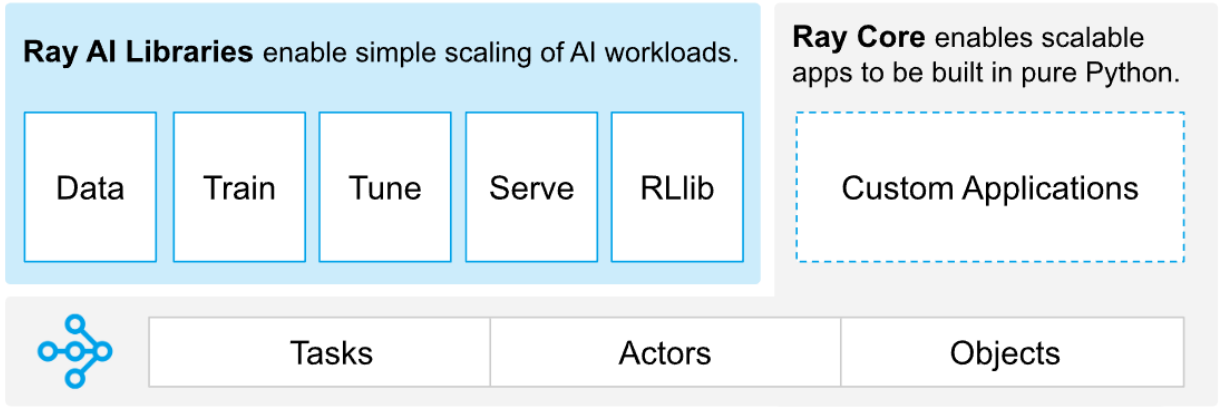
코드 | 작성하기 | 커리어 상담 | Claude의 선택

앱 연결 신규

GPU 인프라 프로비저닝 유스케이스: Ray cluster

RAY

Ray is a unified framework for scaling AI and Python applications. Ray consists of a core distributed runtime and a set of AI libraries for simplifying ML compute:



https://github.com/ray-project/ray

Ray AI Libraries:

- Data: Scalable Datasets for ML
- Train: Distributed Training
- Tune: Scalable Hyperparameter Tuning
- RLlib: Scalable Reinforcement Learning
- Serve: Scalable and Programmable Serving

Ray Core and its key abstractions:

- Tasks: Stateless functions executed in the cluster.
- Actors: Stateful worker processes created in the cluster.
- Objects: Immutable values accessible across the cluster.

Dashboard Configuration

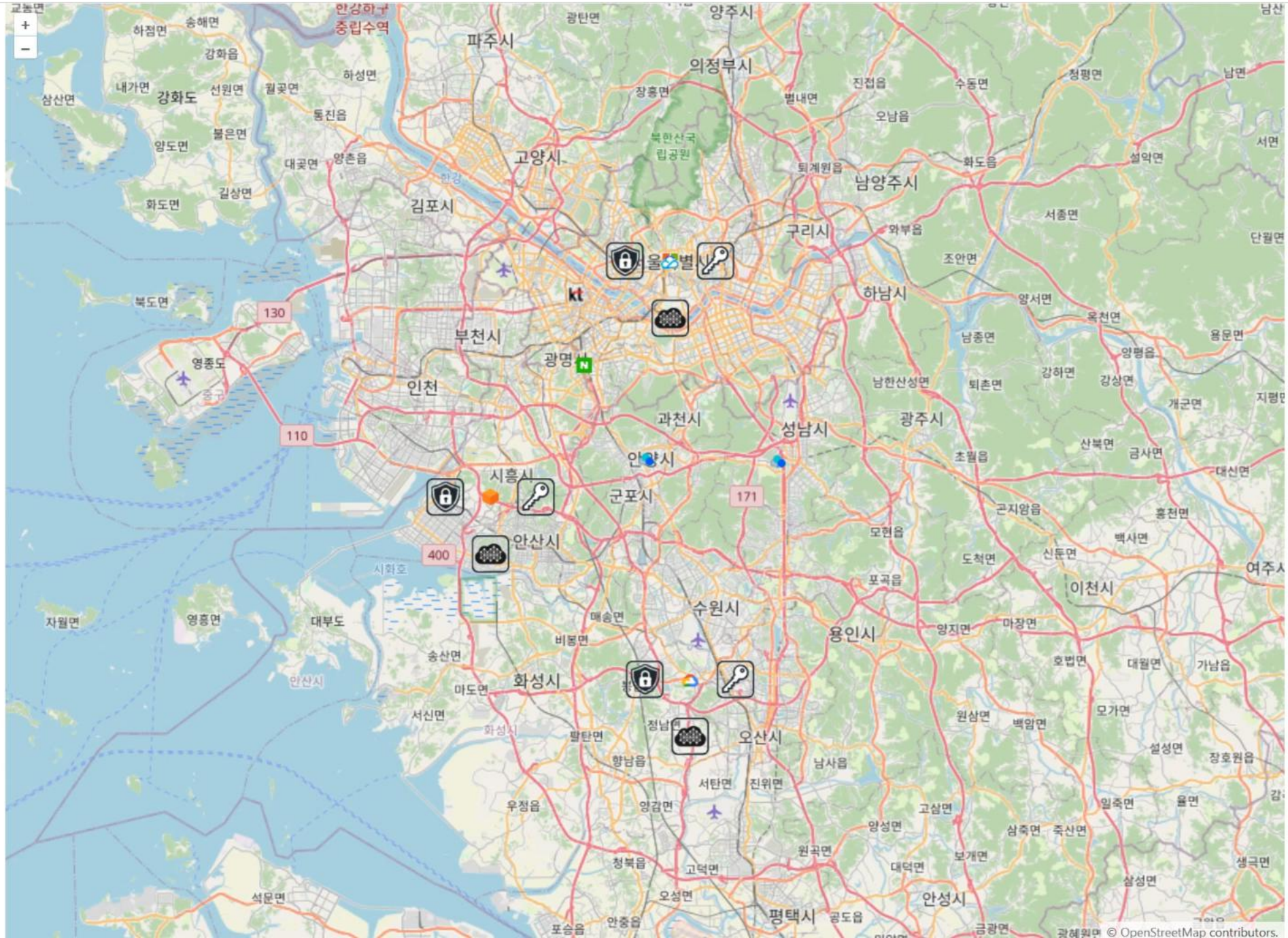
IP	localhost
Port	1323
Username	default
Password	•••••••
Refresh(s)	5
Req Log	Select ID

Map Control

Toggle Map/API

MC-Infra Provisioning

Policy	Location-based
Provider	ALL
vCPU	1
Mem(Gi)	0.5
SpecID	Ex: 2xlarge
Image(OS)	Ubuntu 22.04



Take Home Message, and Your Action

CB-Tumblebug (Multi-Cloud Infra Management) 👋

접속: github.com/cloud-barista/cb-tumblebug

★ 함께 만들어가는 오픈소스★

- 사랑과 관심: 구독(Fork)과 좋아요(Star)
- 그리고 컨트리뷰션이 필요합니다! ☺

- Golang
- Apache 2.0

멀티 클라우드에 진심인 사람들의 이야기

전세계 클라우드를 내 손안에, 멀티 클라우드

Cloud-Barista Community 11th Conference

감사합니다.

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

손 석 호 / shsonkorea 골뱅이 etri.re.kr