

K-DA 제품을 통한 컨테이너 어플리케이션 운영전략

장 민

나무기술 / 기술3본부장

Table of Contents

- Why Cloud
- 인프라의 변경 그 후는
- 멀티 클라우드의 고민
- 컨테이너 – 멀티클라우드를 구축하기 위한 도구
- 각테일 - 컨테이너 매니지먼트 플랫폼
- Q&A

Why Cloud?

왜 레거시(Legacy) 시스템을 떠나야 할까요?

빠르게 변화하는 기술에 대응이 어렵고,
여러 가지 한계에 봉착합니다.

변화 대응에 어려움

높은 복잡도

높은 초기 투자 비용

시스템 통합 한계

지속적으로 상승하는 유지보수 비용

비즈니스 가치 창출의 걸림돌

클라우드 도입 효과



비용 절감

하드웨어, 소프트웨어 등을 구매하지 않고 필요한 만큼만 빌려쓰으로써 구매·운영에 소요되는 비용을 절감할 수 있습니다.

생산성 향상

오토스케일링, 클러스터링 등을 통해 보다 효과적인 데이터 처리가 가능합니다.



편리성 증대

개발서버, 운영서버 등 추가시 웹화면을 통해 간편하게 추가·삭제, 모니터링 등을 할 수 있어 운영이 편리해 집니다.

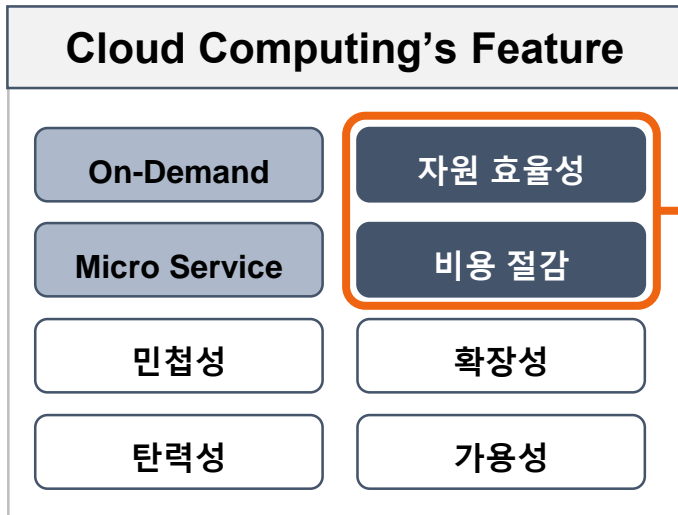
융합서비스의 발전

사물인터넷, 빅데이터, 모바일 서비스를 효과적으로 운용할 수 있을 뿐 아니라, 다양한 서비스로 발전할 수 있는 기반이 됩니다.



클라우드 도입 효과

클라우드 도입시 주요 고려 사항



IT 인프라 자원의 효율적인 활용으로 비용 절감에 초점
→ VM 중심 IaaS의 도입
→ IaaS의 효율적인 이용에 만족?

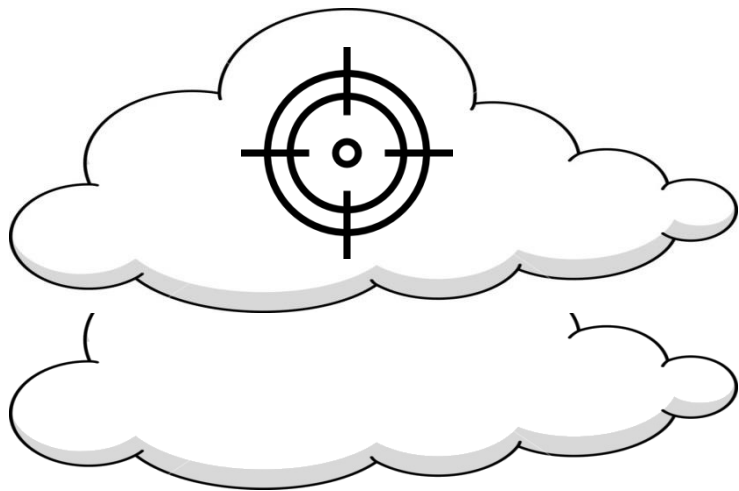
인프라의 변경 그 후는?

지금까지의 클라우드 도입은...



어플리케이션의 변화없이
IT 인프라 자원의 가상화/클라우드 화에 초점

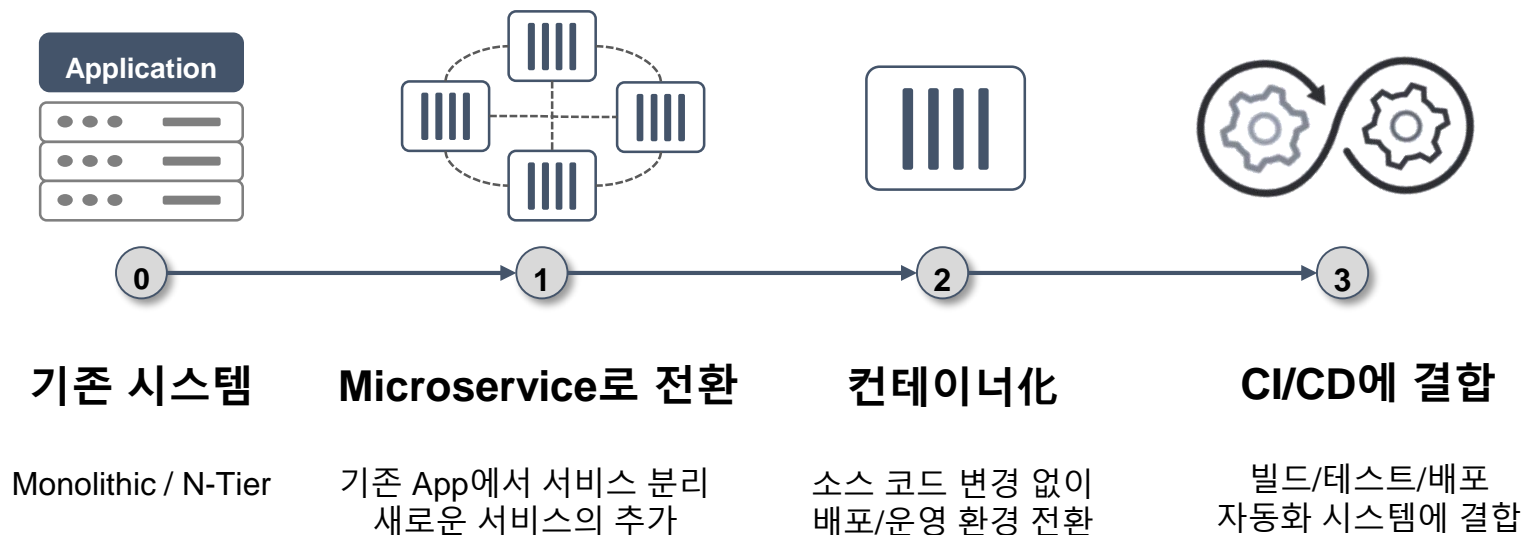
어플리케이션의 클라우드化에 초점을 맞추어야 한다.



Application

Infrastructure

어떻게 도입 할 것인가?



마이크로 서비스로의 단계적 전환.

멀티클라우드의 고민

Enterprise(Multi) Cloud

비즈니스와 IT 전략을 구현한 기업 클라우드
기술, 인프라 보다는 어플리케이션 서비스 중심
기업 고유의 클라우드로 맞춤화, 최적화가 중요



Build Your Own Cloud

Enterprise Cloud 도전 과제

- **어플리케이션 서비스의 가용성과 확장성**
 - 가용성과 확장성은 클라우드 인프라의 장점이며, 다양한 기술과 공급자를 통해 확보 가능
 - 단, 어플리케이션 서비스의 경우 인프라 만으로 가용성과 확장성을 보장 하기 어려움
 - 어플리케이션 특성에 맞는 클라우드 구축과 운영 최적화가 중요
- **멀티/하이브리드 클라우드 관리(Multi/Hybrid Cloud)**
 - 단계적 도입, 보안, 기술, 비용을 고려 단일 보다는 멀티/하이브리드 클라우드 구축이 일반적
 - 어플리케이션 서비스의 요구에 따라 클라우드 인프라 및 기술의 선택이 유연하여야 함
 - 이를 위해 복잡성과 이질성을 극복하기 위한 멀티/하이브리드 클라우드 통합 관리가 필요

Enterprise Cloud 구성

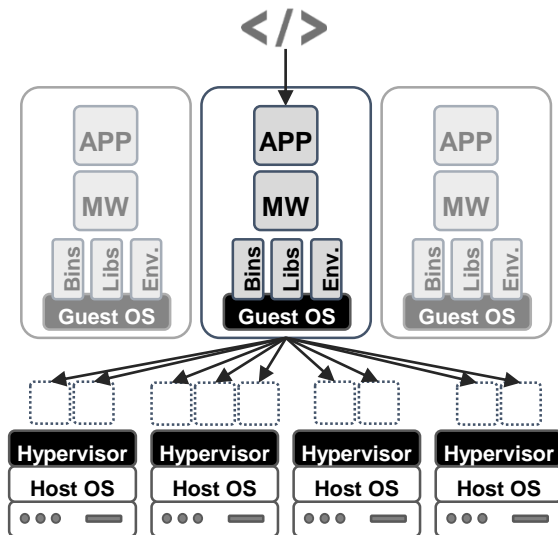
- 엔터프라이즈 클라우드는 어플리케이션 서비스에 적합한 클라우드 도입과 가용성/확장성을 위한 최적화, 효율적인 개발과 운영 체계로 구성되며, 기업 비즈니스 전략 구현의 클라우드 플랫폼을 제공



컨테이너

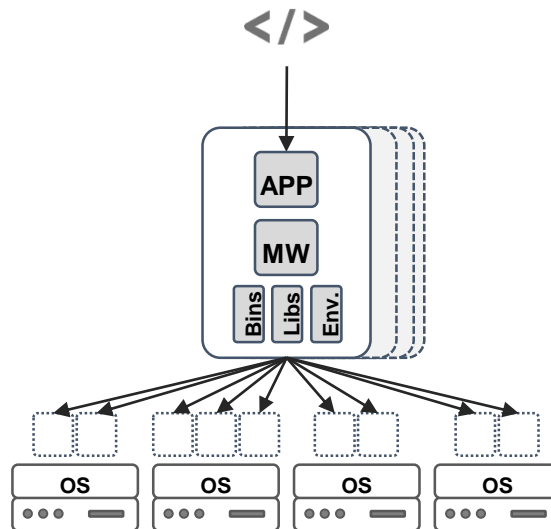
멀티클라우드를 구축하기 위한 도구

Journey to Portability



Virtual Machine

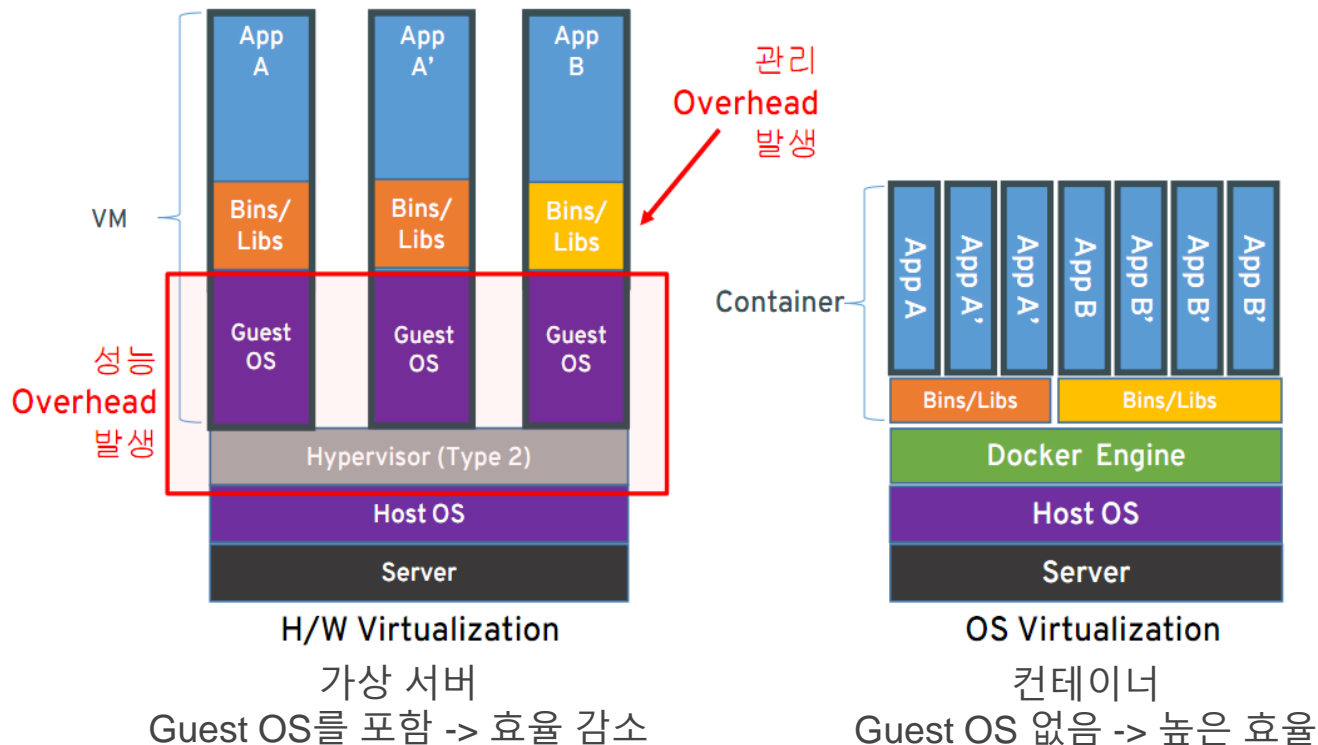
Emulation for Isolation
VM based Portability
Heavyweight, Performance Loss
Inefficient for App Change
Proprietary



Container

OS Level Virtualization
Isolation, Not Emulation
Share OS Kernel
No Guest OS, No Hypervisor
Self-Contained (App, Bins, Libs, Env.)

컨테이너와 가상서버(VM) 비교



컨테이너 기술의 도입이 기업을 바꾸고 있습니다.

70%
Increase in
operational
efficiency



Up To
90%
Decrease

Deployment
Time

Up To
90%
Decrease

MTTR For Deploying
Security Patches

Up To
90%
Decrease

Time
to Scale

80%

Of IT Budgets are spent on
maintaining existing legacy
applications

- Apps are deployed across 1500 servers
- Takes 6 weeks to deploy any changes
- And 72 hours to scale

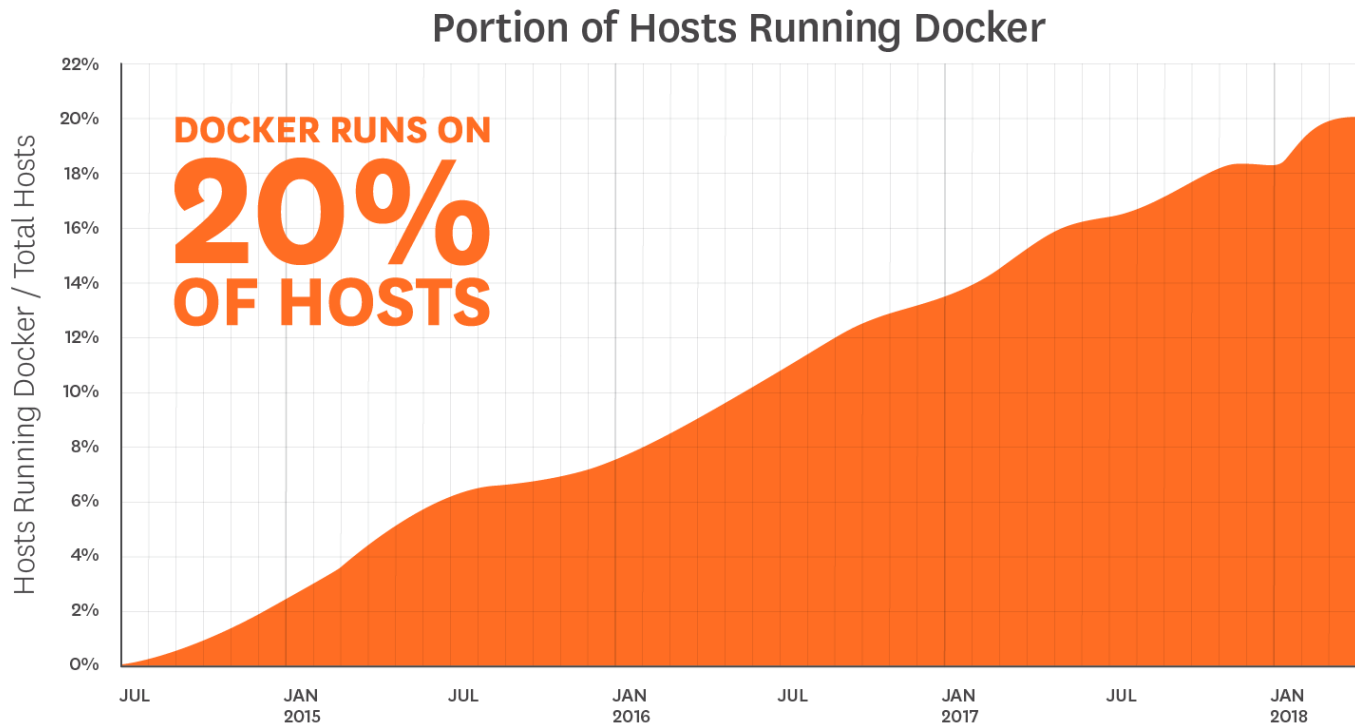
80% of \$30M budget

- Apps are condensed to **450 servers** instead of 1500
- Changes get deployed **multiple times an hour**
- Scale in just **minutes**

30% of \$30M budget

Source : ROI Infographic from [docker.com](https://www.docker.com)

컨테이너 기술은 트렌드입니다.



Source: Datadog

컨테이너 시장규모

2021년 까지 34억\$ 규모의 시장 될 전망
연평균 성장률(CAGR) 35.4% 예상

451 Research의 CET (Cloud Enabling Technologies)

Market Monitor & Forecast 서비스에 따르면

응용 프로그램 컨테이너 시장은 향후 5 년 동안

폭발 적으로 성장할 것 예상

APPLICATION CONTAINERS

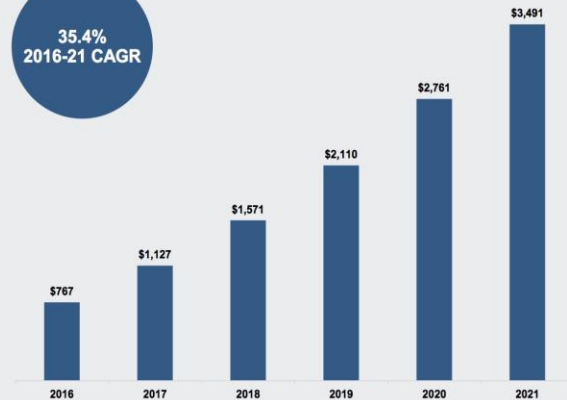
Total Market Revenue (\$m)

The modern application container market and ecosystem are growing fast, with new open source software projects adding to an already crowded field characterized by a fairly even mix of new and established vendors. 'We see the same enterprise customer growth that these vendors do in our latest Voice of the Enterprise (VoTE) research, indicating increased interest and use overall with growing production use as well.

We expect an increased enterprise use, particularly in production, to drive more vendors and investment toward plugging the enterprise gaps. The early involvement of the largest vendors in the industry is further evidence that the application container market is real and presents a significant change in the way enterprise organizations develop and deploy software and manage their IT infrastructure. The participation of startups and newer vendors serves to further disrupt not only the larger enterprise software market, but also the process, planning and spending of enterprise customers. We also expect this will fuel consolidation, mergers and acquisitions among vendors across the market segments we cover in this report.

The ecosystem and market around modern application containers is still early and rapidly growing given the early nature of enterprise use. However, the container trend has attracted a plethora of startups and early involvement from the largest vendors in the enterprise software industry.

35.4%
2016-21 CAGR

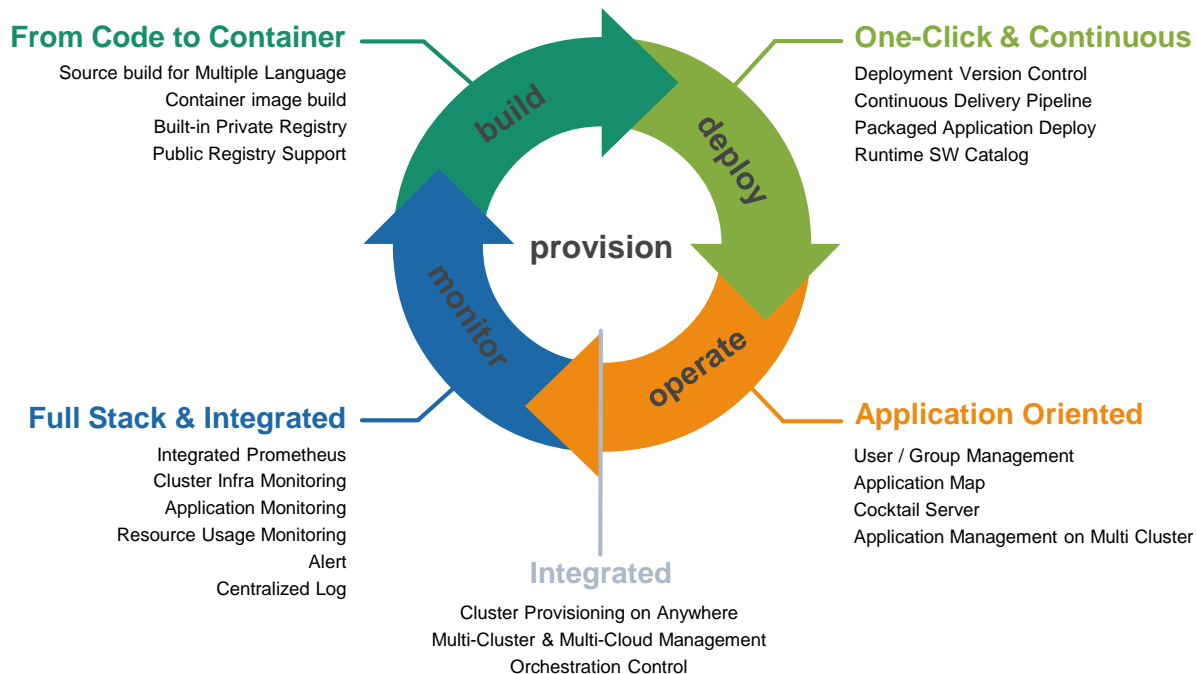


각테일

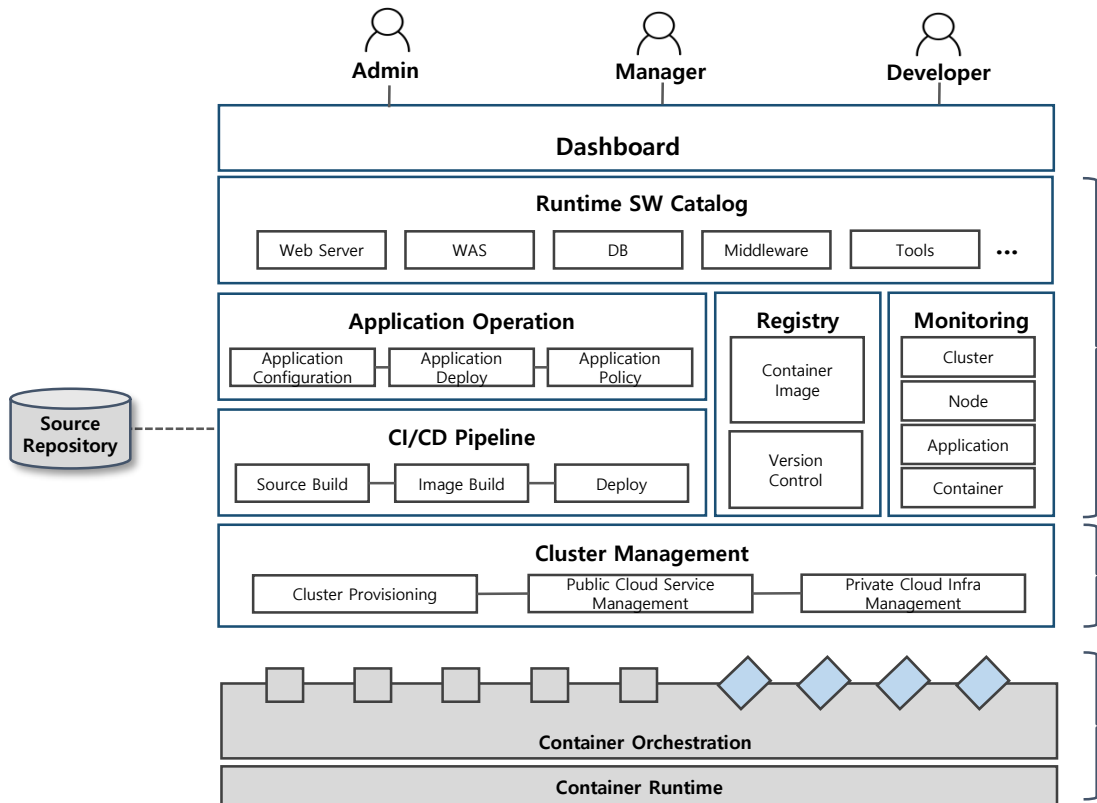
컨테이너 매니지먼트 플랫폼

칵테일 클라우드는 무엇?

All-in-one Container Management Platform



Overview



대쉬보드

- DevOps, Admin, Operator 역할별 클러스터 자원 할당관리, 마이크로 서비스 모니터링 대쉬 보드 화면

DevOps 프로세스 자동화 관리

- 개발 언어별 빌드/배포 자동화
- 컨테이너 이미지의 저장과 보안 관리
- 배포 어플리케이션 운영 관리
- 모니터링
- 컴포넌트 SW의 카탈로그 관리

클러스터의 선택적 사용과 통합 관리

- 클러스터 설치와 구성 관리
- 퍼블릭 클라우드 서비스 관리
- 프라이빗 클라우드 서비스 관리

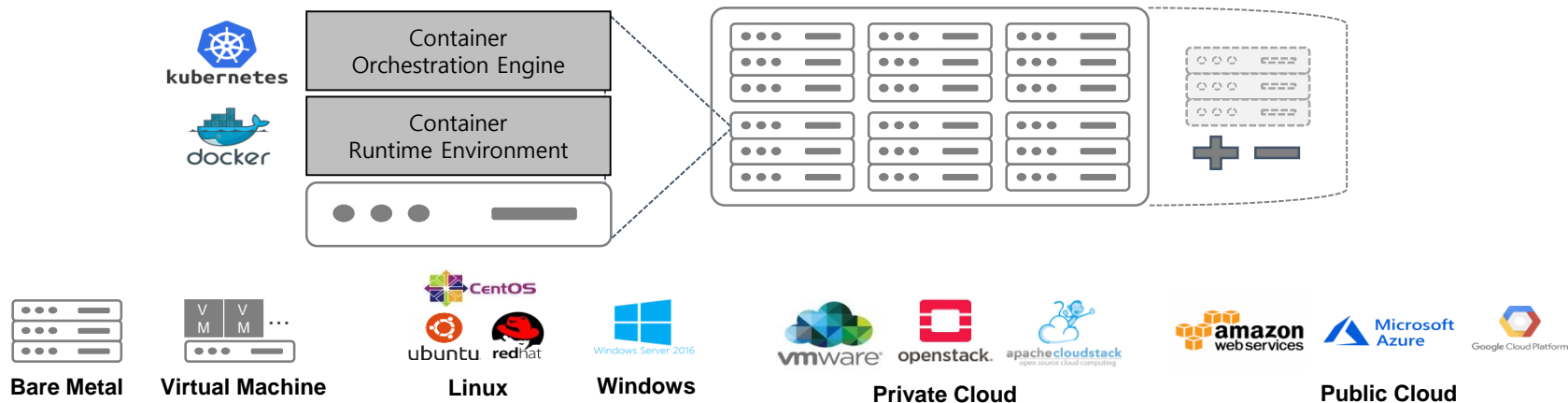
컨테이너 운영 환경

- Custom Controller (Replication)
- Cluster Provisioning
- Storage Volume Provisioning

Functional Scope

Integrated Cluster Management

Kubernetes Cluster Provisioning

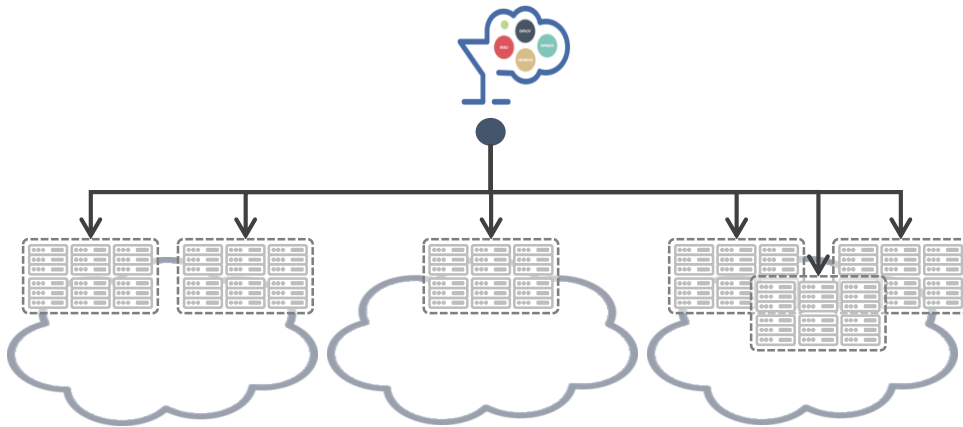


- Kubernetes의 설치와 Cluster Provisioning
 - Infra에 무관한 Cluster 구성
 - 서비스 중단없이 Node를 추가/삭제

Functional Scope

Integrated Cluster Management

Multi Cluster on Multi Cloud



- 각테일 하나로 멀티 클러스터 관리를 한 번에
- 복수의 클라우드 인프라 환경에서 복수의 클러스터 통합 관리
 - 어플리케이션의 인프라 간 이전 자동화 환경

Functional Scope

Integrated Cluster Management

Multi Cluster on Multi Cloud

Cocktail Cloud Demo

장민

서비스

빌드

카탈로그

클러스터

구성원

클러스터

클러스터 현황

2019-06-05 23:06:47

정철

이름

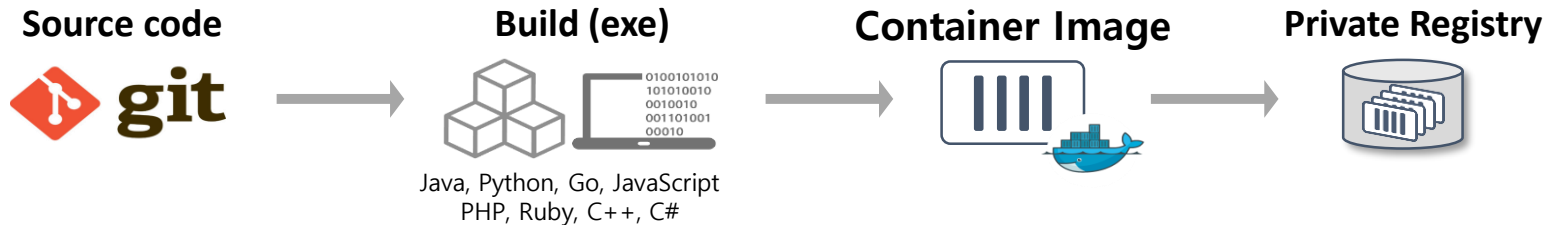
오름차순

클러스터	상태	서비스	Core	Gbyte	Gbyte	Ea
acorn-demo-cluster	Running	16	27.2	33.6	214	70
On-Premise, Seoul	5					
gke-acornsoft-demo-cluster	Running	1	6.5	6.3	15	9
Google Cloud Platform, Tokyo	0	gke-acornsoft-demo-cluster				

Functional Scope

From Code to Container **Build**

Build (Source, Container) & Registry



- Source Build for Multiple Language
 - Container Image Build
 - Built-in Private Registry
 - Public Registry Support

Functional Scope

From Code to Container **Build**

Build (Source, Container) & Registry

acloud

Cocktail Cloud Demo

장민

서비스

빌드

카탈로그

클러스터

구성원

빌드

빌드 관리

Filter

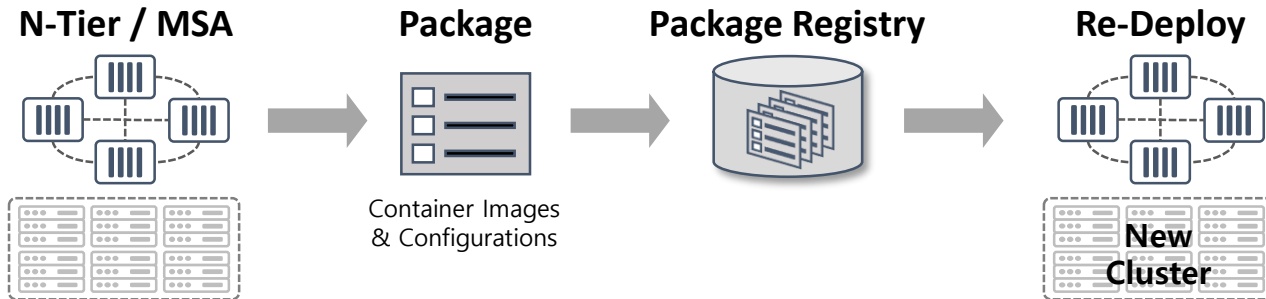
빌드 생성

빌드명	이미지명	작업 수	상태	마지막 빌드 일시	이미지 크기 (Mb)
demo-was-ant	regi.accloud.run/cocktail-demo/was:1.0.B000005	4	DONE	2019-06-05 14:25:44	503.59
build-step	cocktail-demo/build-step:0.0.1	6	ERROR 4/6	2019-04-22 12:09:16	
demo-jupyterlab	regi.accloud.run/cocktail-demo/jupyterlab:1.0.B000001	2	DONE	2019-03-14 10:40:28	1305.81
nginx	regi.accloud.run/cocktail-demo/nginx:stable.B000001	2	DONE	2019-03-13 13:48:08	104.07
rstudio	regi.accloud.run/cocktail-demo/rstudio:3.5.2.B000001	2	DONE	2019-03-13 13:33:48	1063.43

Functional Scope

One-Click & Continuous Deploy

Packaged Application Deploy



- 복수의 컨테이너로 구성된 Application을 Package로 저장
 - Application의 Configuration을 버전 관리
- Application을 임의의 클러스터에 Package단위로 한 번에 재배포
 - Stage별, Cloud Infra별, Region별 Application 배포 관리

Functional Scope

One-Click & Continuous Deploy

Packaged Application Deploy

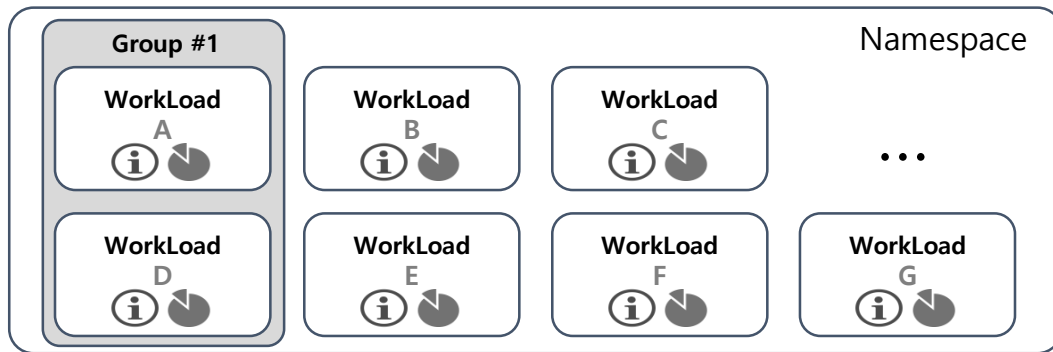
The screenshot displays the 'acloud' Cocktail Cloud Demo interface. The top navigation bar includes the 'acloud' logo, a 'Cocktail Cloud Demo' dropdown, and a user profile '장민'. The left sidebar contains navigation icons for '서비스' (Service), '빌드' (Build), '카탈로그' (Catalog), '클러스터' (Cluster), and '구성원' (Members). The main content area is titled '카탈로그' (Catalog) and '마이 카탈로그' (My Catalog). It features a grid of application cards, each with a title, description, and a '6 템플릿' (6 Templates) button. The cards are:

- DB 데모 애플리케이션**: MySQL과 PostgreSQL로 구성된 DB 데모 애플리케이션입니다.
- Elasticsearch 6.4.2**: Elasticsearch Cluster
- jupyter-spark-demo**: jupyter-spark-demo
- electro-shop-demo**: electro-shop-demo
- benchmark-sock-shop**: sock-shop for benchmarking
- [Demo]tensorflow-jupyterlab**: JupyterLab을 이용한 웹 기반 TensorFlow 개발 환경
- notebook**: 빅데이터 분석용 R IDE환경 및 패키지 관리, 시각화 기능 제공
- spark**: POC spark

Functional Scope

Application Oriented **Operate**

Service Map

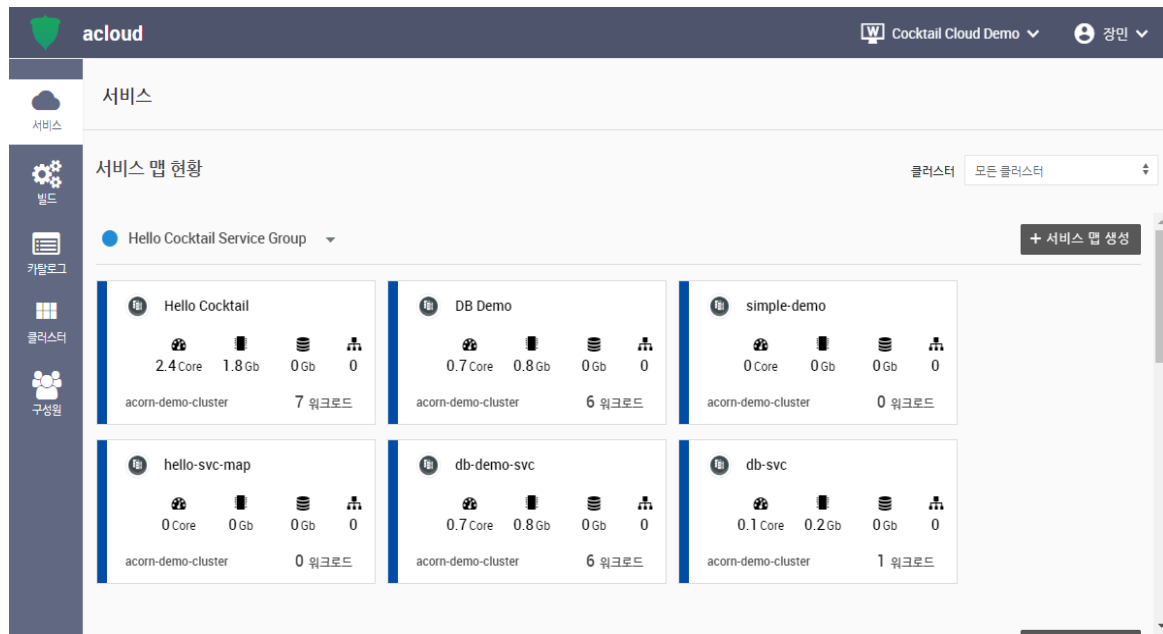


- 동일한 Namespace를 갖는 WorkLoad들을 배포하는 공간
- Application Map에 속하는 WorkLoad의 정보와 상태를 한번에 감시/관리
 - 복수의 WorkLoad를 논리적으로 Grouping 가능
- 동일 Application을 Stage별, Cloud Infra별, Region별로 구성 가능
 - 동일 Cluster에 복수의 Application Map 생성 가능

Functional Scope

Application Oriented **Operate**

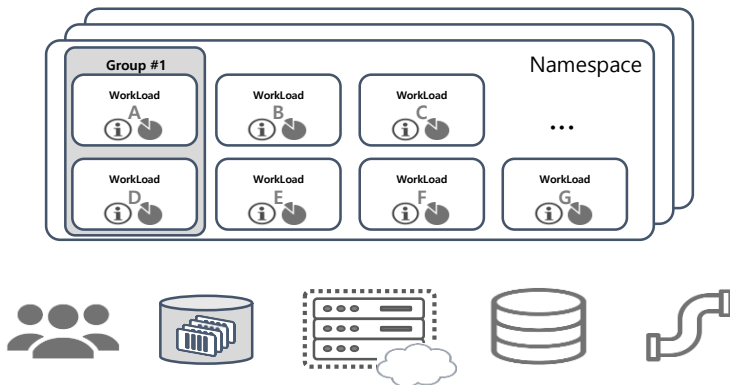
Service Map



Functional Scope

Application Oriented **Operate**

Work Space



- 프로젝트 별 개발 운영을 위한 독립적인 작업 공간
 - Workspace별 User의 등록/삭제
- Workspace별 사용할 수 있는 Registry, Cluster, Volume의 할당
 - Workspace에서 사용할 Build 관리
- Application Map마다 CI/CD Pipeline 생성과 관리

Functional Scope

Application Oriented **Operate**

Work Space


시스템


클러스터


워크스페이스

시스템 관리


워크스페이스 관리

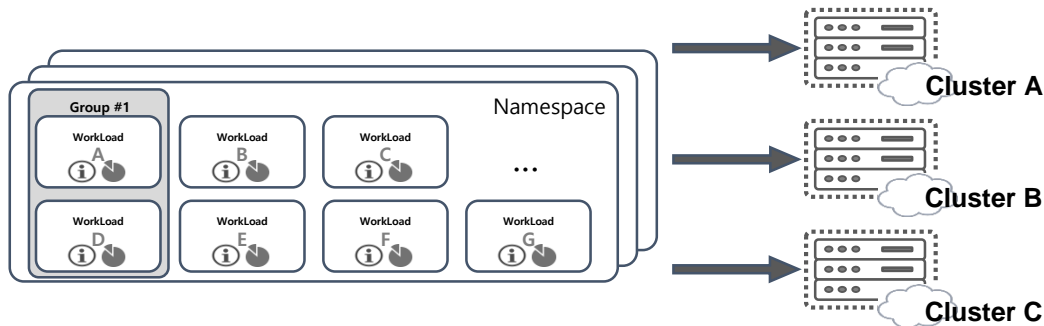
워크스페이스 생성

워크스페이스 명	클러스터 수	구성원 수	설명
Cloud Team	8	6	...
Sports Team	3	9	...
MIS Team	6	8	...

Functional Scope

Application Oriented **Operate**

Applications on Multi Cluster

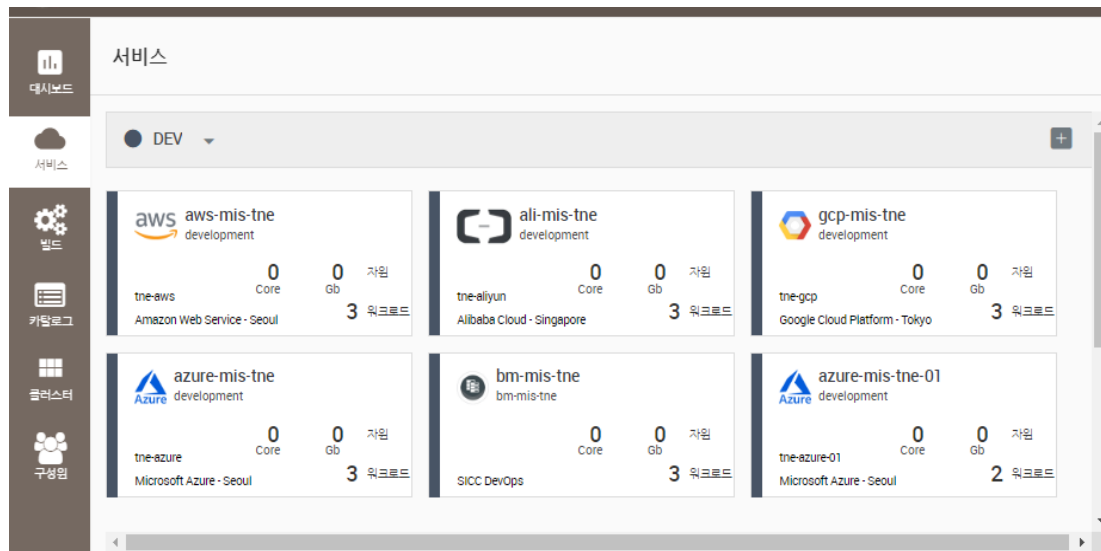


- Workspace에 할당된 Cluster를 선택하여 Application 배포
- 목적 (Stage, Region, Infra)에 따라 Application 배포 운영
 - 사용하는 Cluster의 정보와 상태를 확인

Functional Scope

Application Oriented **Operate**

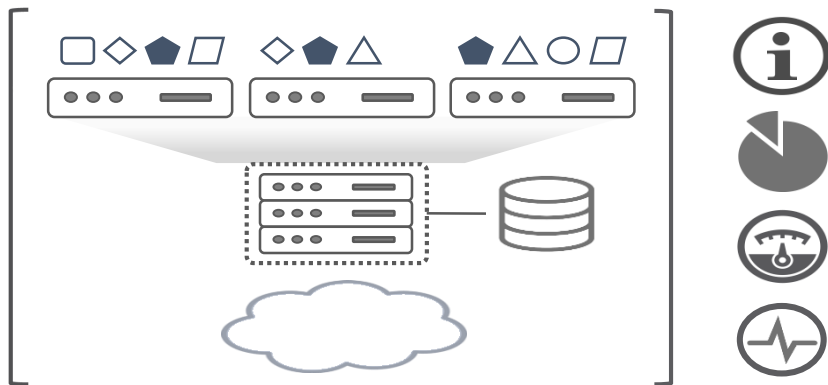
Applications on Multi Cluster



Functional Scope

Full Stack and Integrated Monitor

Cluster Infra Monitoring

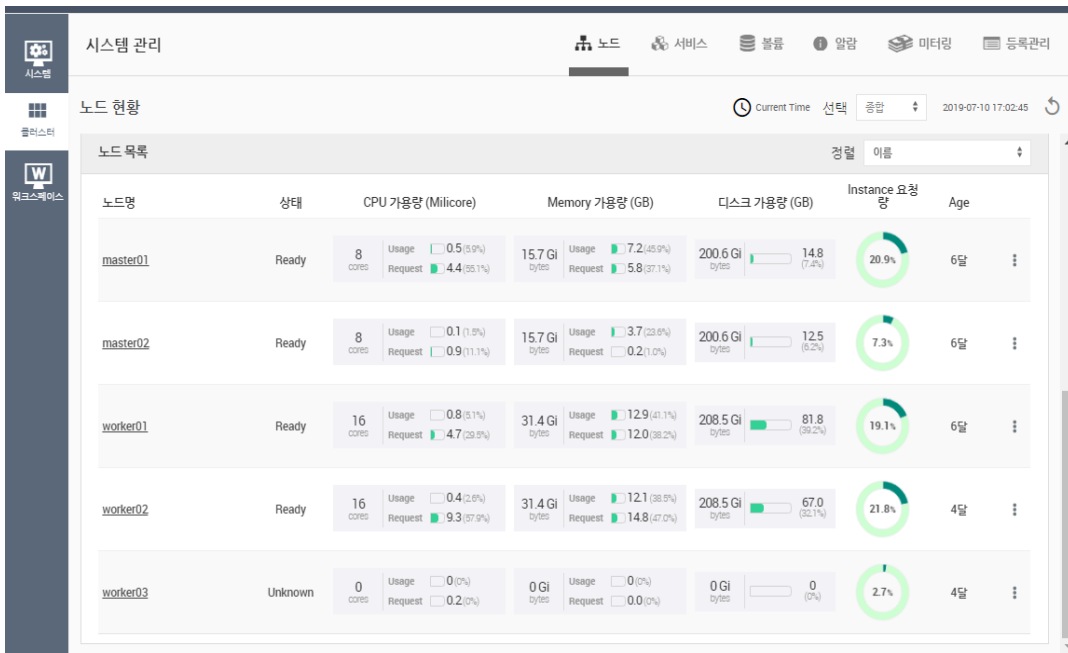


- Cluster별 Provider, Node, Volume의 기본 정보와 상태 정보
 - Node별 Application의 배치 상태 확인
 - Public Cloud Service는 Metering 정보까지 제공

Functional Scope

Full Stack and Integrated Monitor

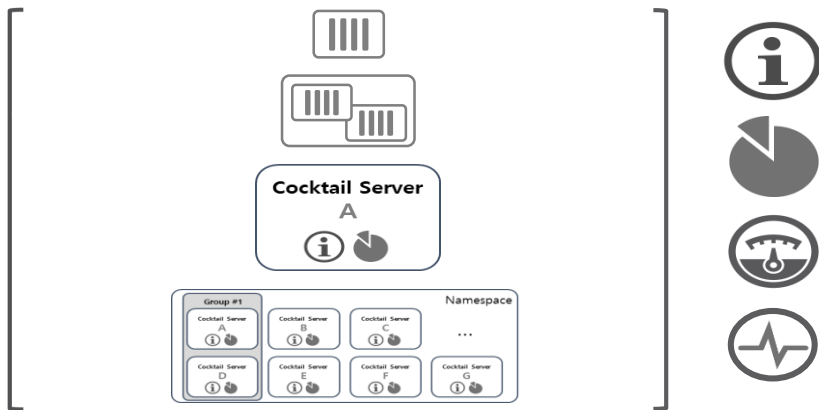
Cluster Infra Monitoring



Functional Scope

Full Stack and Integrated Monitor

Application Monitoring

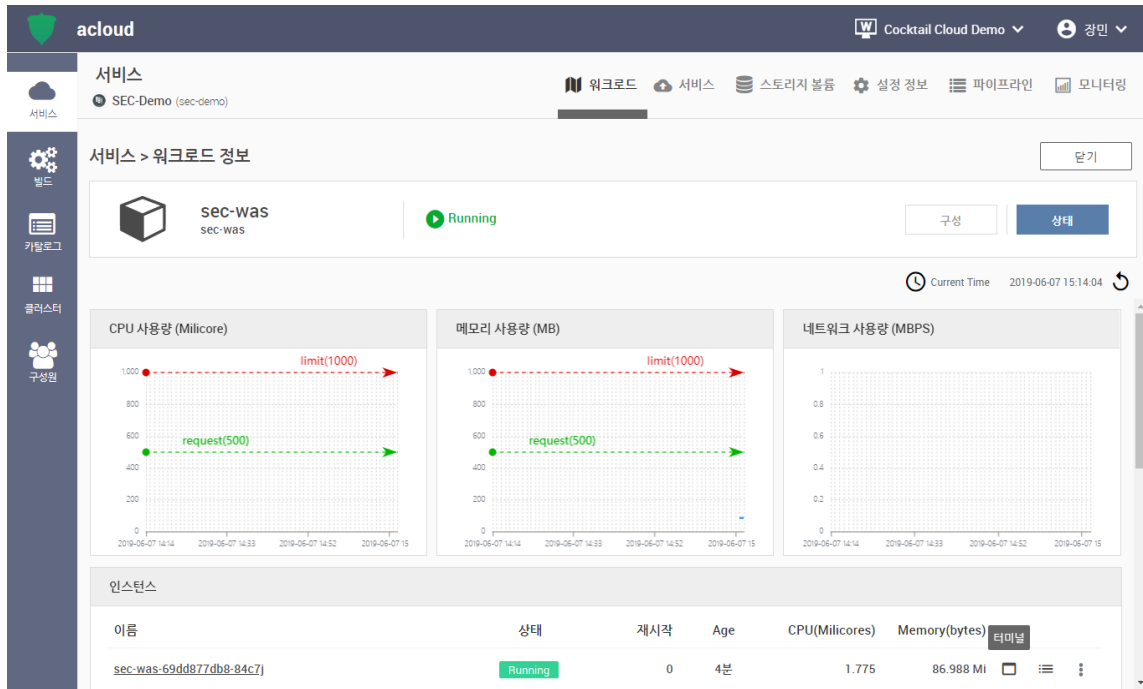


- Application Map별 Monitoring
- WorkLoad/Pod/Container별 Monitoring

Functional Scope

Full Stack and Integrated Monitor

Application Monitoring



Q&A





감사합니다.



namu[•]

장민 min.jang@namutech.co.kr