

KNOM Conference 2016

Future Trends in Cloud Computing and Docker

2016. 05. 13

이노그리드 클라우드컴퓨팅 연구소

서버 가상화팀 책임연구원 조철용

cheolyong@innogrid.com



ABOUT ME



클라우드 컴퓨팅 연구센터
서버 가상화팀 책임 연구원



그리드 딜리버리 솔루션 개발
클라우드 솔루션 개발



C/C++, JAVA
서버 개발자

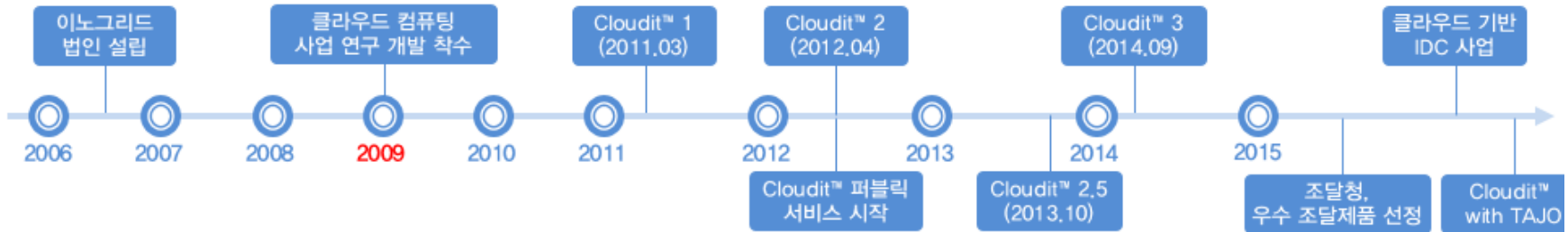


OpenStack
Cloudit,
Ceph, Docker,

ABOUT INNOGRID



이노그리드는 2006년 설립 이후 그리드컴퓨팅, 슈퍼컴퓨팅, CDN 사업 분야를 중심으로 기술 및 사업 기반을 확보하였고, 2009년부터 클라우드 컴퓨팅 연구 개발에 착수하여 꾸준히 성장하고 있습니다.



2006

- 법인 설립

2008

- 슈퍼컴퓨터 4호기 HPC 최적 병렬화 지원 사업 수주 (KISTI)
- 슈퍼컴퓨터 사용자 환경 시스템 구축
- 씨디넷웍스 품질측정 시스템 구축
- 중소기업 선도형 기술혁신 전략과제 지원사업 수주

2010

- 클라우드 컴퓨팅 업계 최초 녹색기술인증 획득
- **법정부 클라우드 서비스 테스트베드 구축 · 운영**
- 소셜 미디어 서비스를 위한 클라우드 플랫폼 및 응용 서비스 기술 개발 (지식경제부)
- INNO-BIZ 기업 선정
- 미국현지법인 설립

2012

- 퍼블릭 클라우드 서비스 상용화
- **Cloudit™ 2 출시**
- 더존 IDC 통합관리시스템 구축 (더존비즈온)
- CSB를 위한 WAN가속 Prototype 개발 (삼성SDS)
- 대검찰청 사설 클라우드 구축 사업 수주
- WMO세계가상정보센터 구축 사업 수주 (가상청)

2014

- **Cloudit™ 3 출시**
- 유럽 CompatibleOne SAS와 CSB 구축 MOU 체결
- **Cloudit™ GS인증 획득**
- 클라우드서비스 안전성 검증 시험사업 참여기업 선정 (미래부)
- 미래부 '신SW 상품대상' 수상 (Cloudit)
- 클라우드 컨퍼런스 개최(IC3)

2007

- 벤처기업확인(기술보증기금)
- **기업부설연구소 설립**
- KISTI Glory 테스트베드 구축
- 지식경제부 우수신기술 지정지원사업 협약

2009

- **Cloudit™ DS 총 처리 용량 200Gbps 돌파**
- 클라우드 컴퓨팅 기반 차세대 방송 플랫폼 사업 공동연구기관 선정 (방송통신위원회)
- 한국 클라우드산업협회 발족(운영위원장)
- 전세계 I&C 통합 CDN 플랫폼 구축

2011

- **Cloudit™ 1 출시**
- 프로테오믹스 · 클라우드 인프라 업무 제휴
- PVFS 활용을 위한 클라우드 기반 대용량 데이터 처리 플랫폼 개발 (정보통신산업진흥원)
- 카자흐스탄 클라우드 교육서비스 P/P 공동수행 사업

2013

- **Cloudit™ 2.5 출시**
- 클라우드 서비스 품질평가 제공사이트 구축 사업
- 클라우드 전용스위치 실증 시험 환경 구축
- 클라우드 기반 중소기업용 플랫폼 구축 ISP 사업
- 공공 클라우드 서비스 도입체계수립(NA)
- 2013년 지역발전포털 고도화사업(KET)

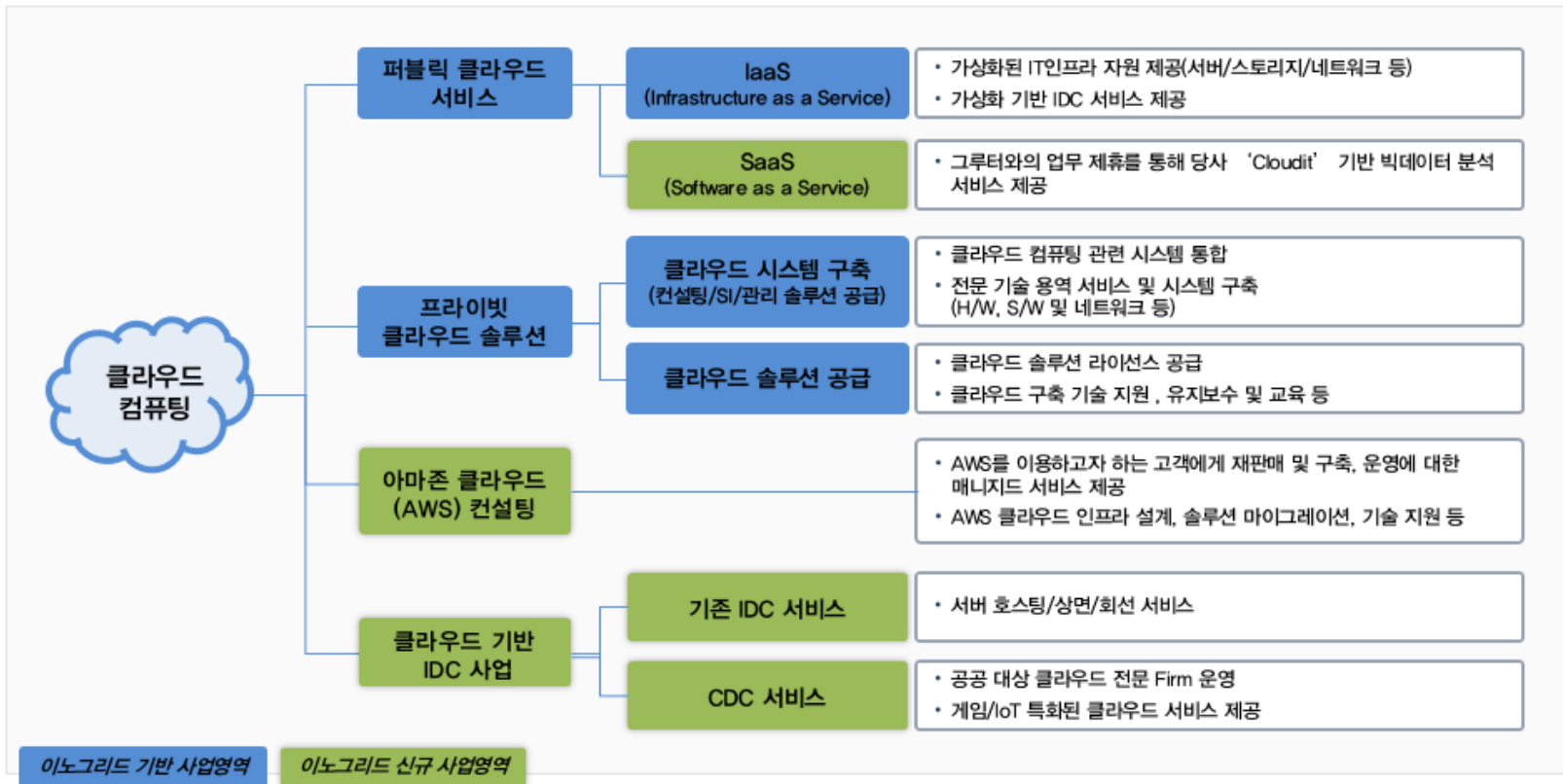
2015

- 클라우드 발전법 시행령 관련 공청회, (중소기업 대표로 패널 참가)
- **Cloudit, 조달청 우수조달 제품으로 선정**
- 빅데이터 전문기업 그루터와 업무 제휴, **클라우드 기반 빅데이터 사업 추진**
- 영국 K글로벌 런던 2015' 전시회 참가
- 티몰러스 출판체결(웹스케이 클라우드)
- **핀터스큐리티시스템(보안)와 MOU 체결**
- **클라우드 기반 IDC 사업 추진**

ABOUT INNOGRID



이노그리드는 기존 클라우드 솔루션 및 서비스 제공 사업에서 CIDC 사업으로 사업 영역을 확장하여 시장 선점 및 영업 이익 증가에 따른 Value-up을 기대하고 있습니다.



- IaaS(Infrastructure-as-a-Service): 서버, 스토리지 등 하드웨어를 서비스하는 클라우드 컴퓨팅 영역
- Paas(Platform-as-a-Service): 데이터베이스 등 소프트웨어 개발 환경을 서비스하는 클라우드 컴퓨팅 영역
- SaaS(Software-as-a-Service): 기업용 소프트웨어, 워드프로세서 등 소프트웨어를 서비스하는 클라우드 컴퓨팅 영역

CONTENTS

1. ABOUT CLOUD COMPUTING

2. FUTURE TRENDS IN CLOUD COMPUTING

3. WHAT IS DOCKER?

4. FUTURE TRENDS IN DOCKER

5. CONCLUSION

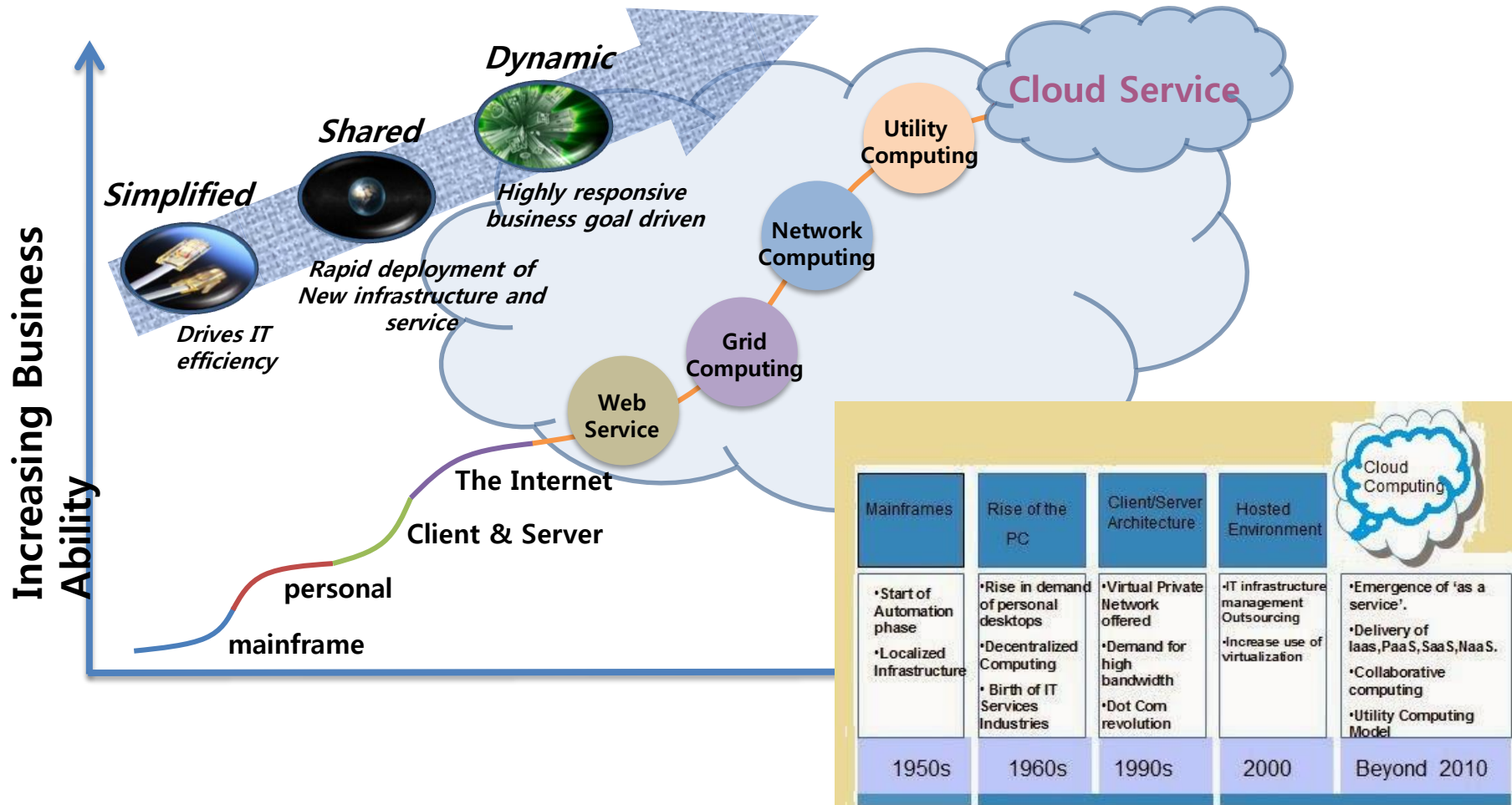




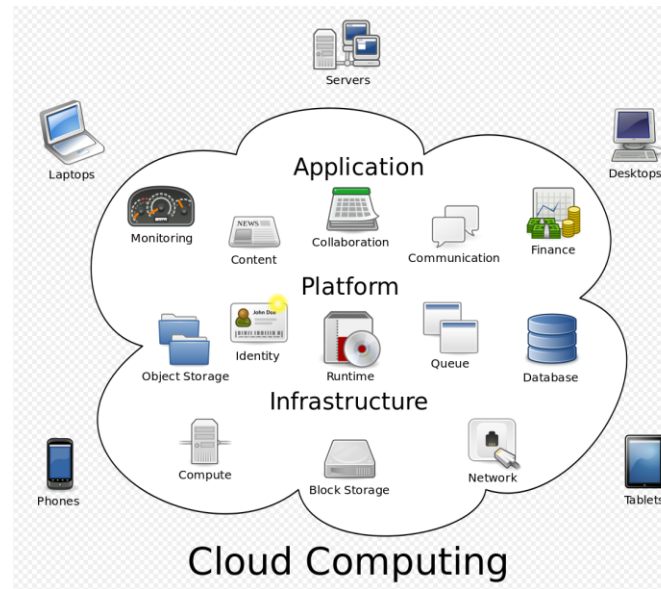
ABOUT CLOUD COMPUTING



EVOLUTION OF COMPUTING

Total growth rate of IT is about 5%, but growth rate of cloud services is about 27%



DEF. OF CLOUD COMPUTING

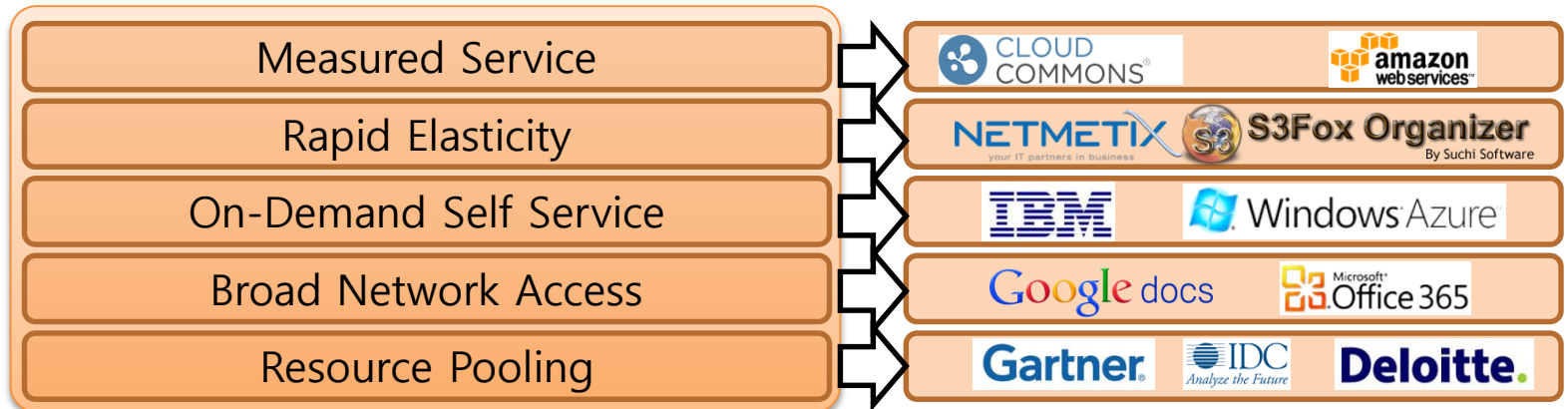


COMPANY 	On-demand IT infrastructure provided as a service to host/run services and user applications
END-USER 	Virtualized infrastructure/platform/service to run client applications and to store data on heterogeneous devices whenever/wherever

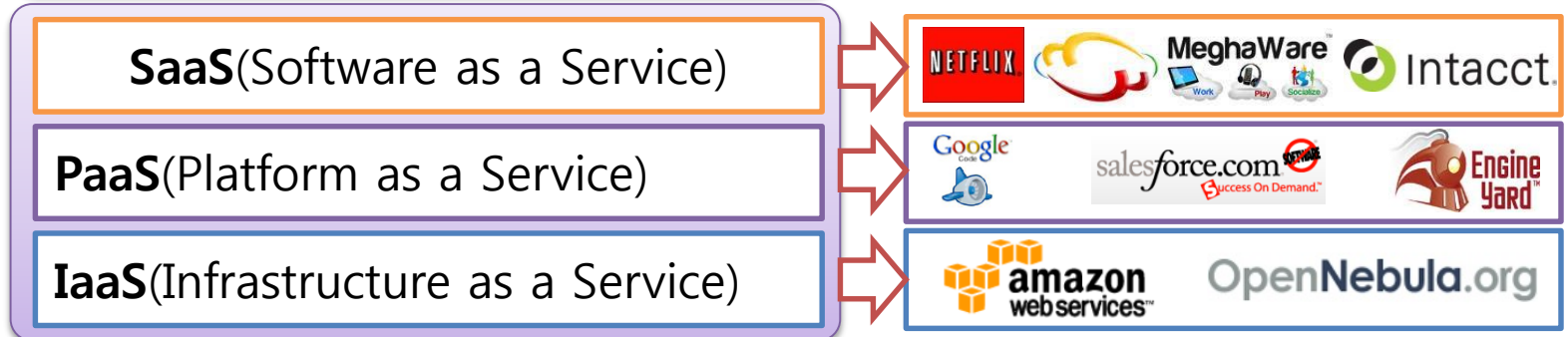
CLOUD COMPUTING ARCHITECTURE



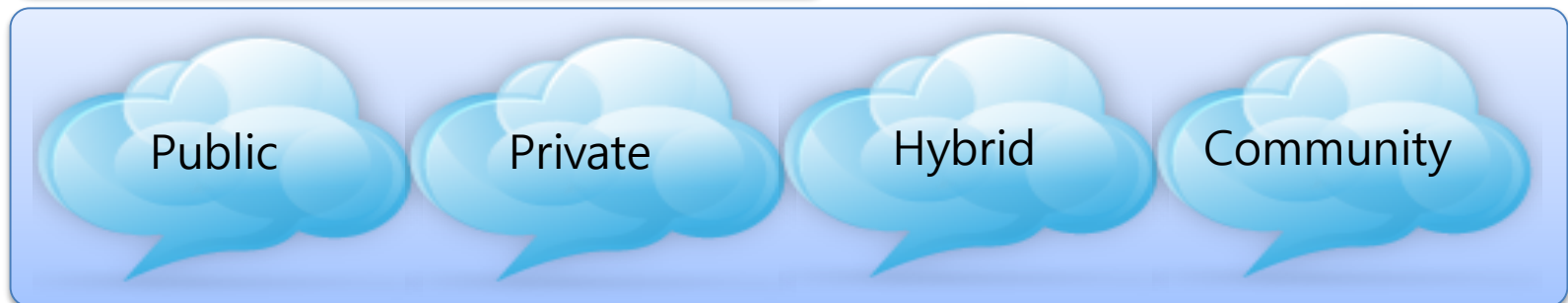
Essential Characteristics



Service Models



Deployment Models



* '09, NIST(National Institute of Standards and Technology's)

WHAT IS THE FIRST COMPANY IN PUBLIC SERVICE?

Compute

Amazon Elastic Compute Cloud (Amazon EC2)



Amazon Elastic MapReduce



Storage

Amazon Simple Storage Service (Amazon S3)



Amazon Elastic Block Storage (Amazon EBS)



AWS Import/Export



AWS Storage Gateway Service



AWS Glacier



Database

Amazon DynamoDB



Amazon Relational Database Service (Amazon RDS)



Amazon ElastiCache



Networking

Amazon Route 53



Amazon Elastic Load Balancing



AWS Direct Connect



Amazon Virtual Private Cloud (VPC)



Content Delivery

Amazon Cloudfront



Elastic Network Instance



Application Services

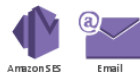
Amazon Simple Queue Service (SQS)



Amazon CloudSearch



Amazon Simple Email Service (SES)



Amazon Simple Workflow (SWF)



Amazon Simple Notification Service (SNS)



Deployment and Management

Amazon Elastic Beanstalk



AWS Identity and Access Management (IAM)



AWS CloudFormation



Monitoring

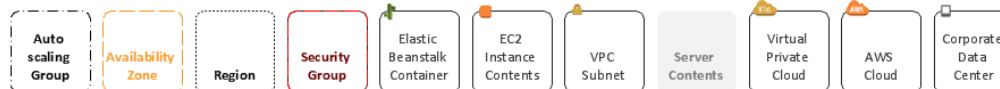
Amazon CloudWatch



Non-Service Specific



Groups

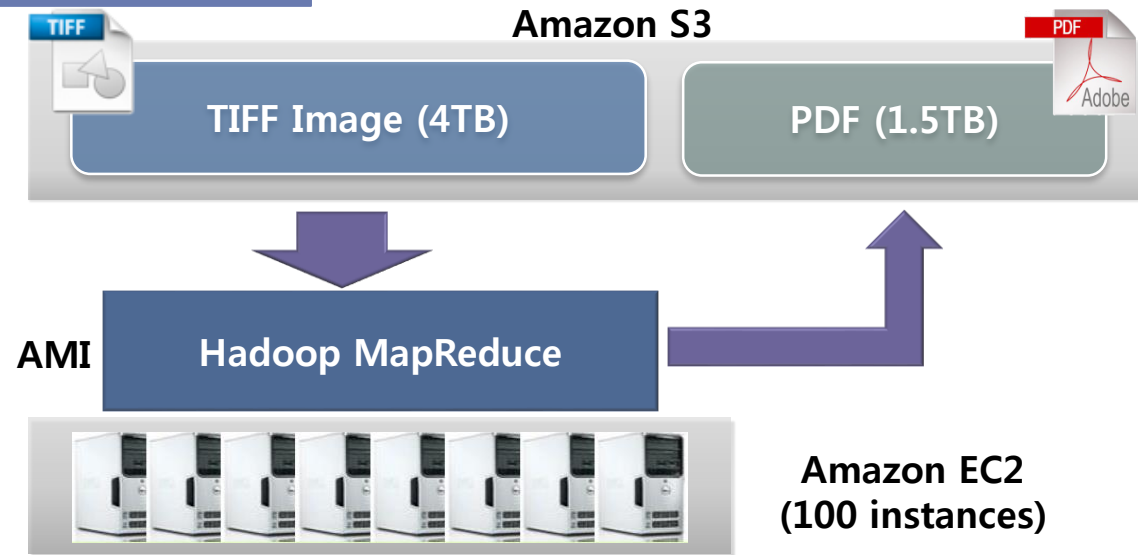


In 2006, the company provided first public cloud service

CASE – NEW YORK TIMES



Architecture



Cost

Only \$ 1,465

S3
Storage: 5.5TB
Data Transfer-in: 4TB

amazon web services™

EC3
Instances:
100 X 24 hours

AWS Simple Monthly Calculator

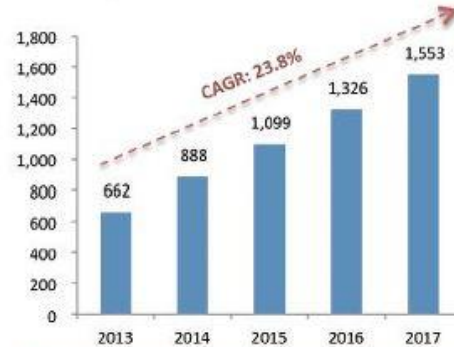
Amazon S3 (US) ☒ Amazon S3(EUR) ☐ Amazon EC2 ☒ Amazon SQS ☐

<http://calculator.s3.amazonaws.com/calc5.html>

CLOUD COMPUTING MARKET

국내 클라우드 시장규모 및 전망

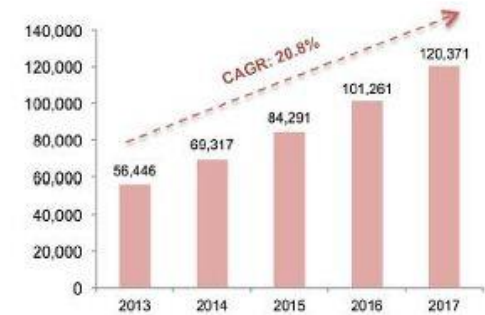
(단위: 백만 달러)



	2013	2014	2015	2016	2017
퍼블릭	487	681	851	1,021	1,172
프라이빗	175	207	248	305	381

글로벌 클라우드 시장규모 및 전망

(단위: 백만 달러)



	2013	2014	2015	2016	2017
퍼블릭	47,396	59,487	73,541	89,411	107,217
프라이빗	9,050	9,830	10,750	11,850	13,154

출처: 부산시 클라우드 산업육성 계획 2014.10 중 IDC 2013.2Q, Infon 2013.11 자료 재구성

국내 클라우드 기업의 클라우드 부문 매출액

(단위 : 백만원)

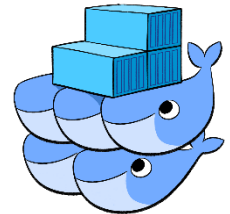
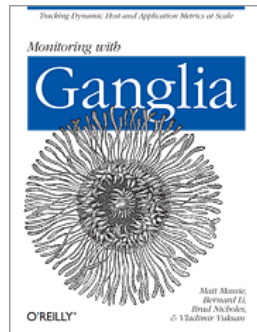
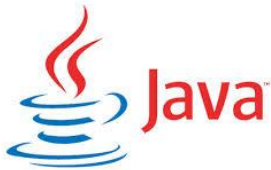
구 분		2011 년	2012 년	2013 년	CAGR
클라우드 서비스	SaaS	138,790	173,848	254,319	35.4%
	PaaS	11,654	18,743	23,428	41.8%
	IaaS	163,848	284,135	331,936	42.3%
클라우드 데이터센터(CDC)		146,721	157,237	167,799	6.9%
클라우드 파트너		35,799	41,734	43,967	10.8%
합계		496,812	675,696	821,450	28.6%

Source : "클라우드 인력 수급
실태조사를 통한 인력양성
체계 마련 (한국클라우드 컴퓨팅 연구
조합, 2014)



FUTURE TRENDS IN CLOUD COMPUTING

CLOUD COMPUTING TECHNOLOGIES



CLOUD PLATFORMS BASED ON OPENS SOURCE FOR IaaS



EUCALYPTUS

<http://www.eucalyptus.com/>



<http://cloudstack.apache.org/>



<http://www.openstack.org/>

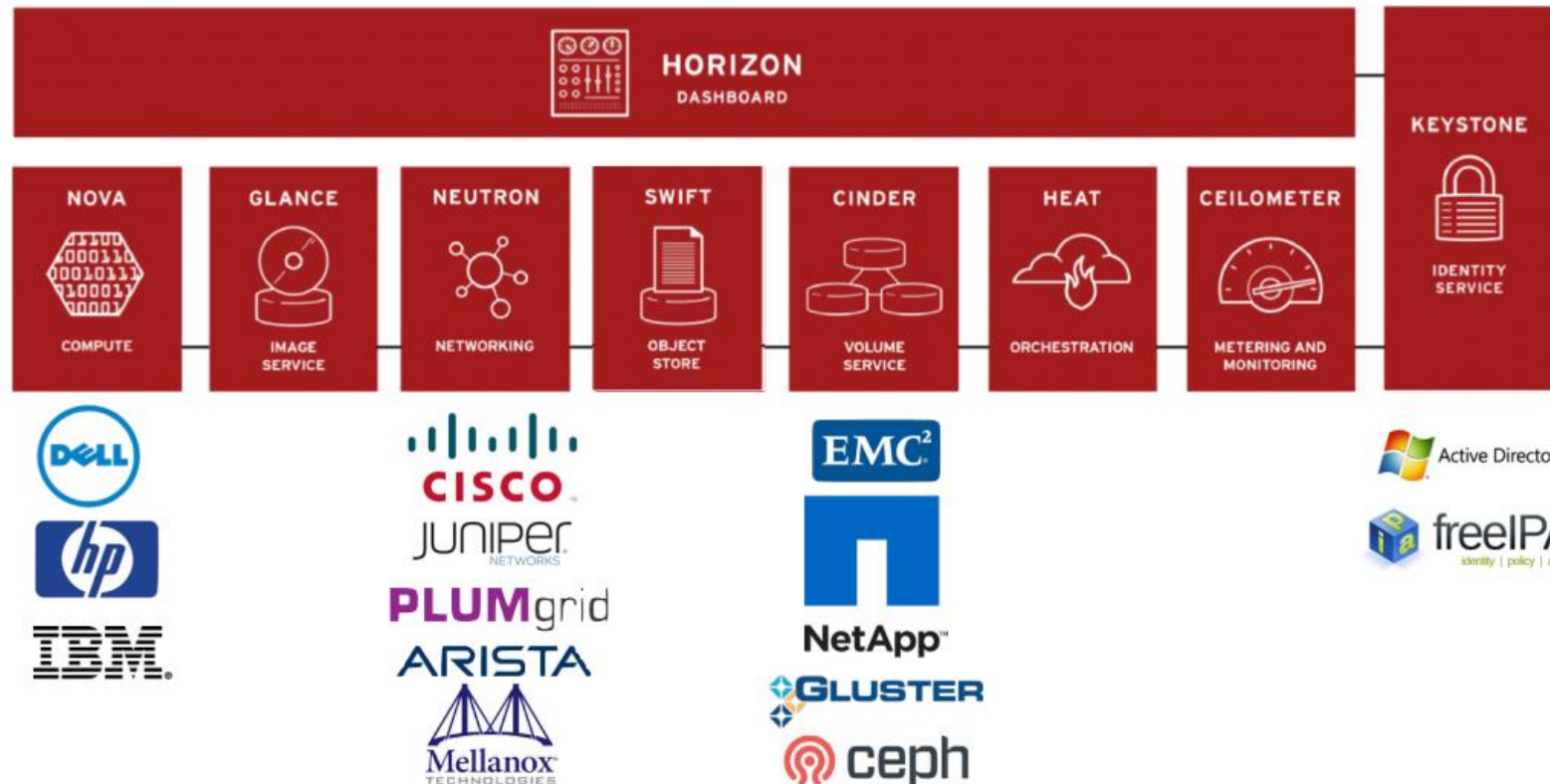


<http://www.opennebula.org/>

WHY OPENSTACK AND WHY NOW?

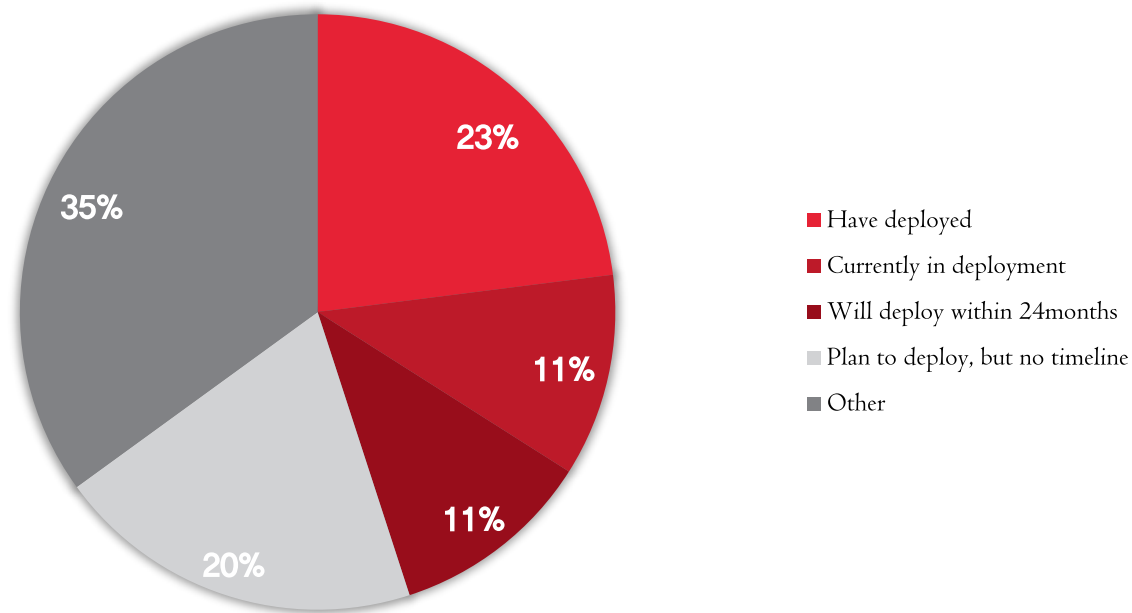


ECOSYSTEM OF HARDWARE AND SOFTWARE





WHO'S USING OPENSTACK?



- 65% of respondents have deployed or are planning to deploy OpenStack
- 62% of respondents plan to use OpenStack for a private cloud infrastructure
- 42% of respondents need OpenStack for massive scalability

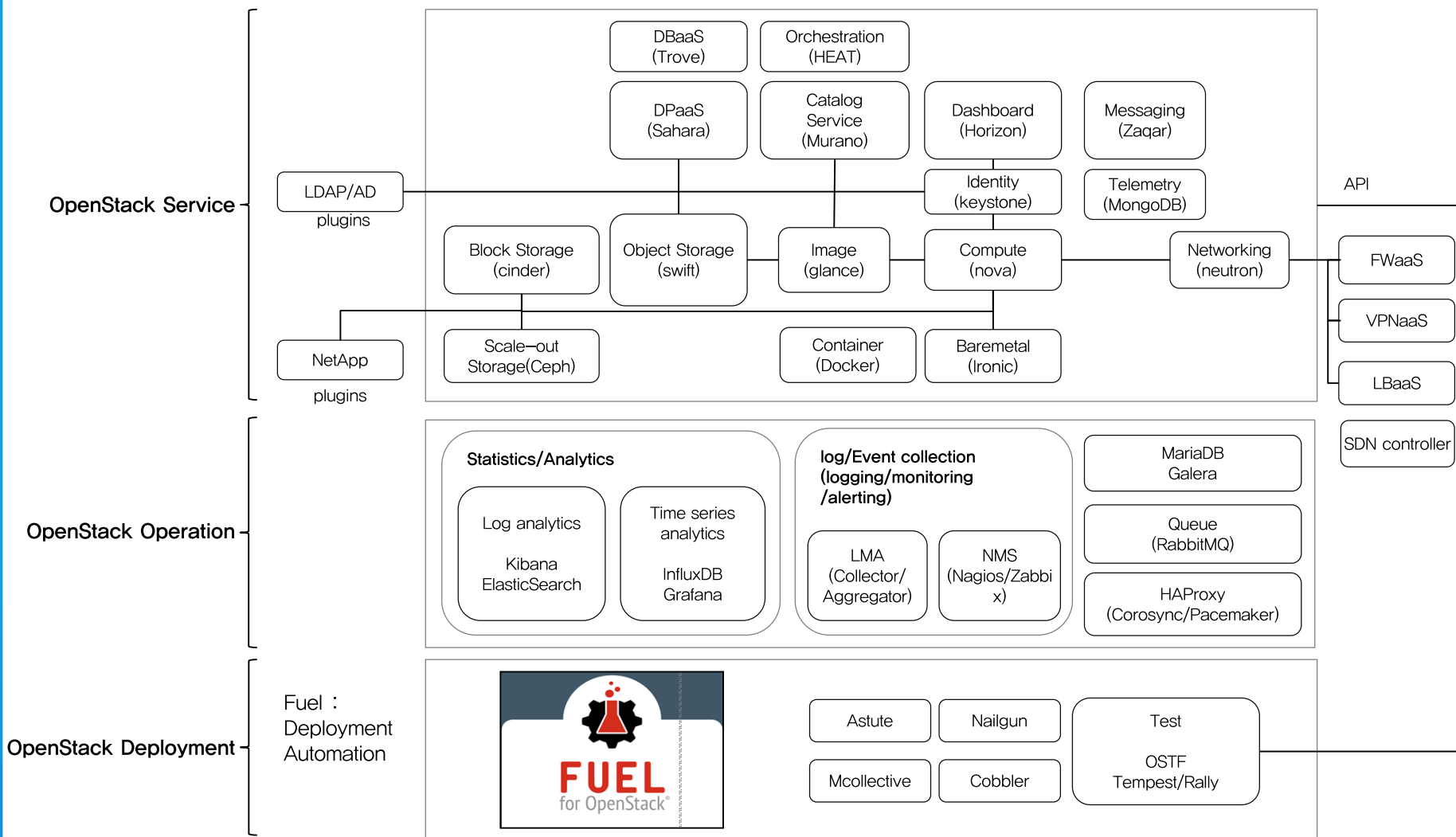


The disadvantage of OpenStack by our experiences at infinition:

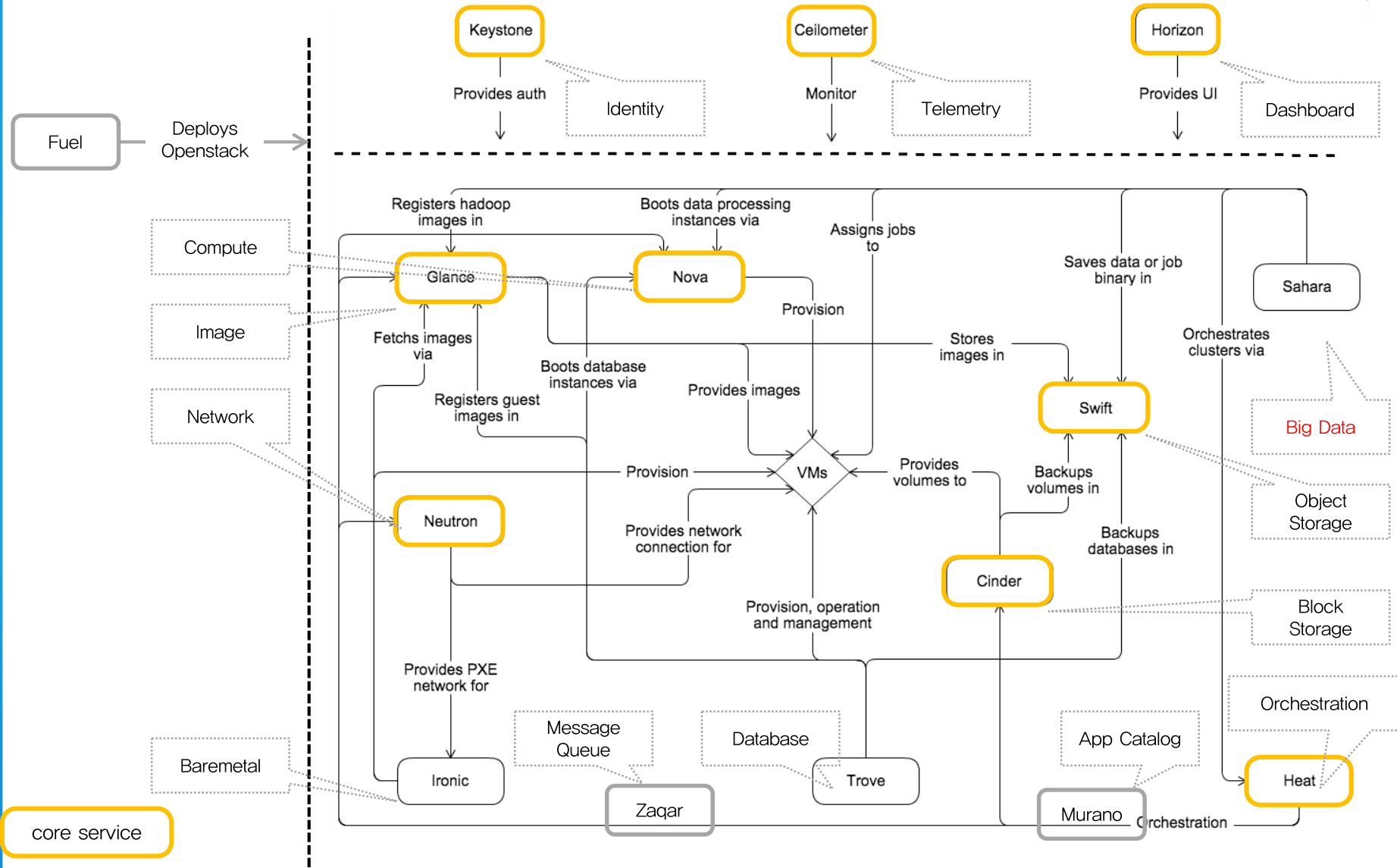
- Very complex to setup and trouble shoot
- Although common code base, might differ from implementations
- Need high numbers of management nodes
- High skills required to run the cluster



OPENSTACK SERVICE STACK



OPENSTACK ARCHITECTURE



3

WHAT IS DOCKER?

VM (VIRTUAL MACHINE)



컴퓨팅 환경을 소프트웨어로 구현하여 여러 OS 를 설치 및 실행

하이퍼바이저(Hypervisor) 를 통한 컴퓨팅 가상화

- KVM, XEN
- VMWare, Virtual PC, VirtualBOX



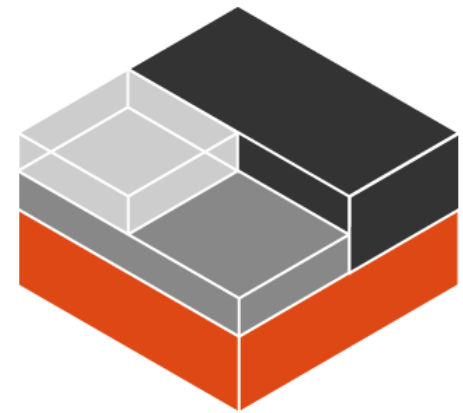
LXC (LINUX CONTAINERS)



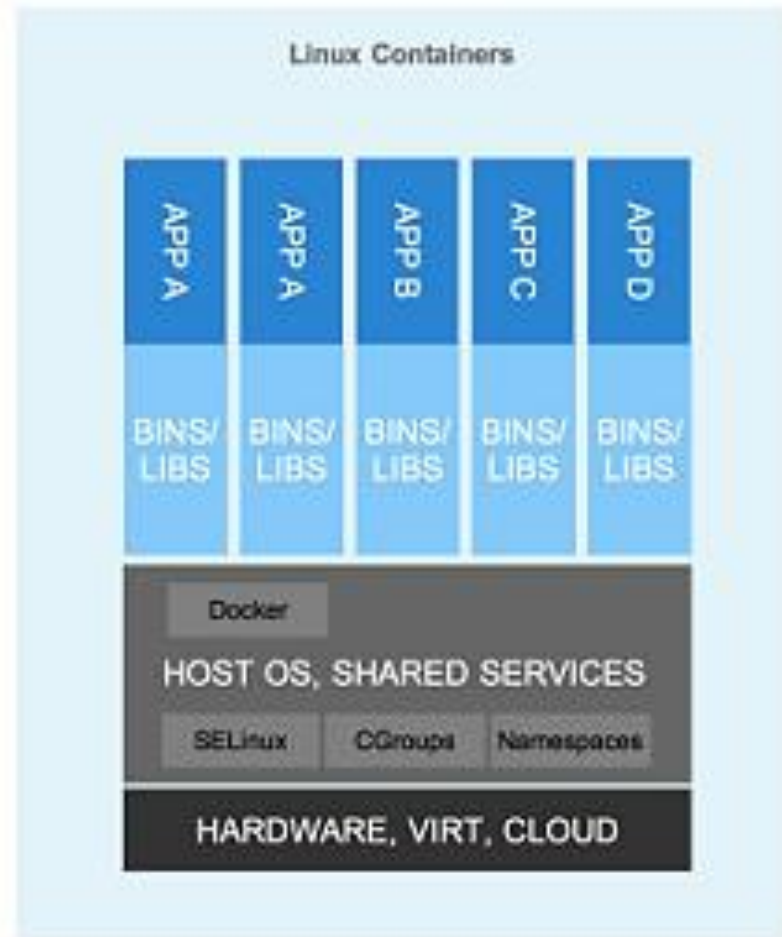
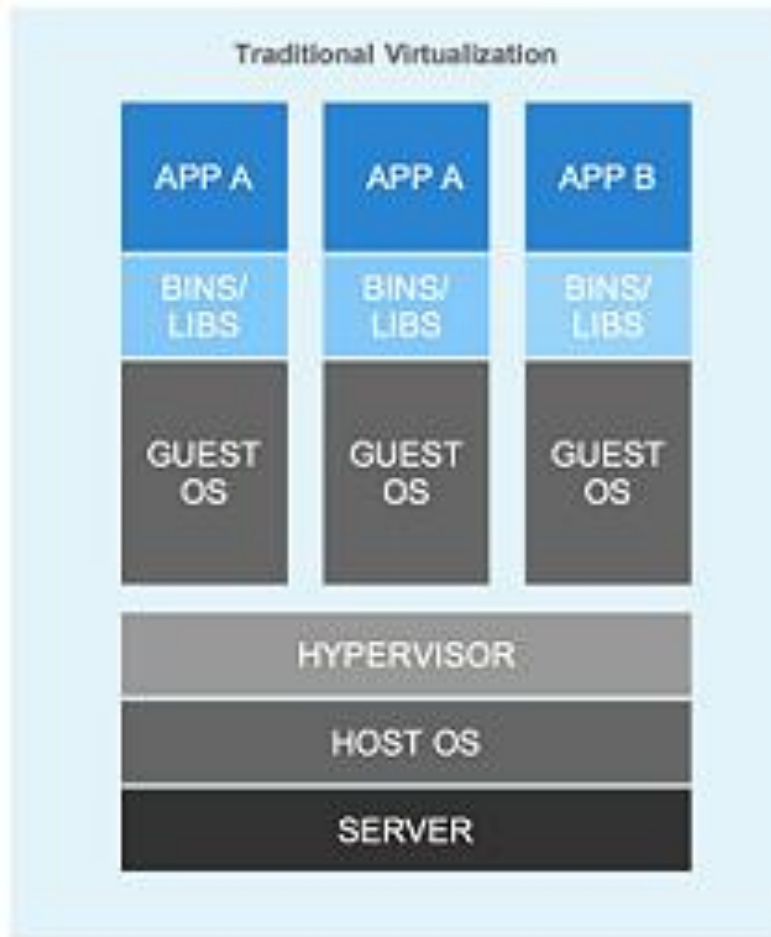
단일 Host OS 내에서 여러 개의 고립된 리눅스 시스템을 실행하기 위한 가상화 방법

Cgroups, namespace 를 이용한 고립된 환경을 제공

가상화가 아닌 자원 격리에 가까움



VM & CONTAINER



CONTAINER VS VM



Container

- 하이퍼바이저(Hypervisor) 필요 없음
- Host OS 커널을 바로 사용
- 자원 격리에 가까움
- Host OS 에 종속적
- 빠르고 효율적으로 app 실행
- 배포 용량 적음
- 자원 효율성 좋음
- Host OS 와 성능 큰 차이 없음

Virtual Machine

- 하이퍼바이저(Hypervisor) 필요
- 하이퍼바이저 위에 Guest OS 가 구동되는 방식
- 커널 자원을 공유하지 않음
- Host OS 에 종속적이지 않음
- 부팅 등 실행시간 오래 걸림
- 배포 용량 큼
- 자원 효율성이 떨어짐
- Host OS 에 비해 성능이 떨어짐



Linux Container 기반 기술

- cgroups, namespace 이용하여 자원 격리

Lightweight

- VM 에 비해 이미지 파일 크기가 작음
- 실행하는 시간이 빠르고 배포 용이

OpenSource

- Python -> GO Language 로 변경
- LXC -> libcontainer 로 구현

Dockerfile

- Docker 이미지를 만들기 위한 설정을 정의하는 파일

Git 과 사용방법 유사

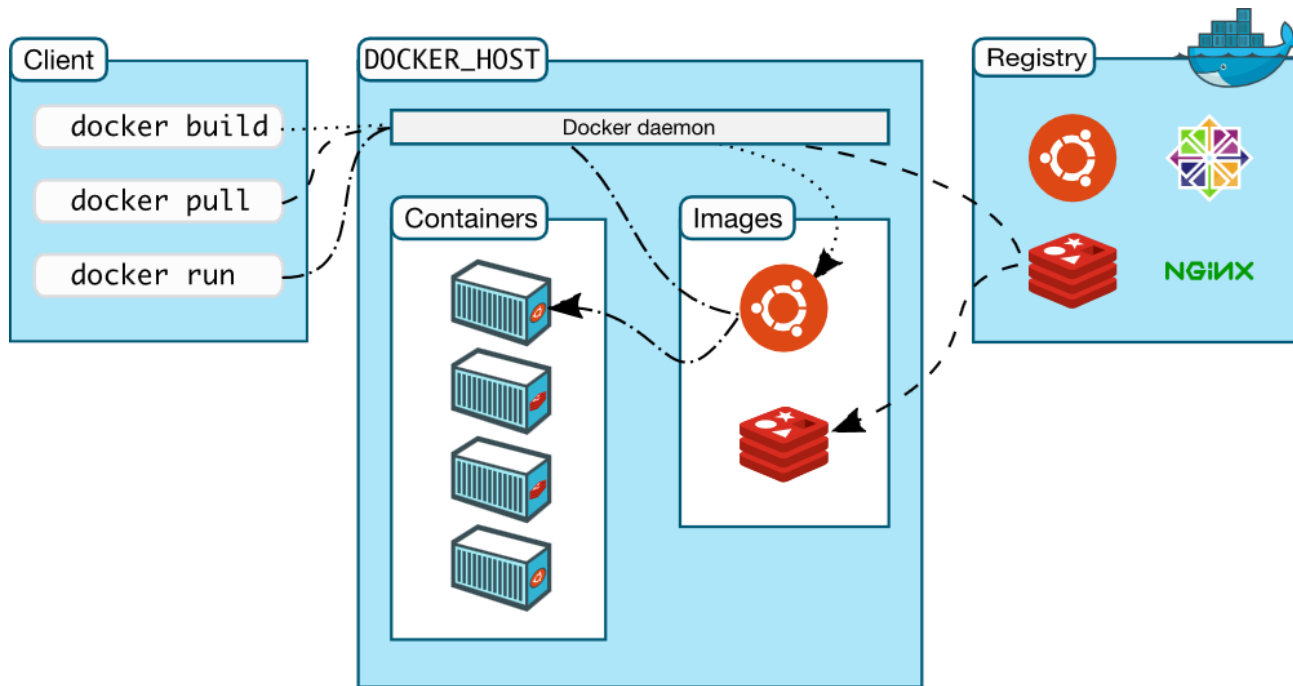
- push / pull 명령을 통해 이미지 전송
- commit 명령을 통해 변경된 사항 업데이트

AWS, Google Cloud Platform, Microsoft Azure 공식 지원

- AWS ECS (EC2 Container Service)
- Google Container Engine
- Azure Container Service



DOCKER 구성요소



- **Docker Host** : Run Container
- **Docker Client** : User Interface
- **Registry** : Image repository (DockerHub)
- **Container** : library, binary, source code
- **Image** : Runtime instance of Image





과거에는 물리 서버에 직접 서비스 운영

- 초기 구축 비용 및 시간이 오래 걸림
- 개발환경 설정이 어려움

클라우드 환경으로 변화

- 가상 서버 자원을 임대하여 서비스
- 동일한 서버 환경 제공 보장
- 구축 비용 및 시간이 단축됨

애플리케이션 다양화, 서버 다양화, 복잡도 증가

- 변화하는 환경에 적응하기 위해 새로운 접근 방법 필요
- Configuration Management Tool 사용 (puppet, chef, ansible, ...)
- Immutable Infrastructure 패러다임 등장

개발자의 역할 변화

- 과거에는 개발자는 개발만, 운영자는 서비스 및 서버 운영만을 독립적으로 수행
- 서비스 스케일링을 고려한 개발 필요성 증대
- 개발에서 배포 단계까지 고려
- DevOps 개념 등장 (Developer + Operation)



CONFIGURATION MANAGEMENT



시스템 인프라 관리 자동화 툴

- Infrastructure 의 코드화

장점

- 서버 운영을 단순화
- 서버 상태를 코드로 관리
- 동일한 설정의 서버를 반복해서 생성 가능 (먹등성)

단점

- 초기 학습 비용이 큼
- 작업 속도가 느림
- 어플리케이션 업데이트에 따른 서버 상태 변경 대응
- 다양한 어플리케이션의 Dependency 관리





VM 사용시 문제점

- 가상머신을 사용할 때마다 소프트웨어 설치
- 설정 자동화가 어려움 (Configuration Management Tool 이용으로 해결)
- 용량이 큼

Immutable Infrastructure 란

- OS 와 서비스 환경을 분리, 서비스 부분만 이미지화 하여 배포
- 서비스가 수정되면 이전 서비스 환경을 새로운 서비스 환경으로 교체
- 클라우드에서 서버를 쓰고 버리듯이 서비스 운영환경을 쓰고 버림

Immutable Infrastructure 특징

- 편리한 관리 : 중앙에서 이미지를 관리 가능
- 확장성 : 하나의 이미지로 다수 서버 생성 가능
- 테스트 : 동일한 환경 구성 가능
- 경량화 : 가볍고 어디든 실행 가능



Immutable Infrastructure 개념을 구현한것

이미지 생성과 배포에 특화된 경량 컨테이너 기술

Docker Image

- 애플리케이션 + 애플리케이션 실행 환경





Minimal OS

- Docker 구동에 최적화된 가볍고 최소화된 OS
- 클러스터링 가능한 Docker 전용 OS
- Chrome OS 기반 개발

No Package Manager

- apt-get, yum 등의 설치 매니저가 없다.

CoreUpdate

- 빠른 OS 업데이트 (2개의 부트 파티션 존재)

CoreOS GUI

- 상용버전 제공





etcd

- 분산 키/값 저장소
- HTTP 프로토콜 지원, JSON 포맷 지원
- 1000s writes/sec per instance
- Raft Consensus Algorithm
- Go Language

systemd

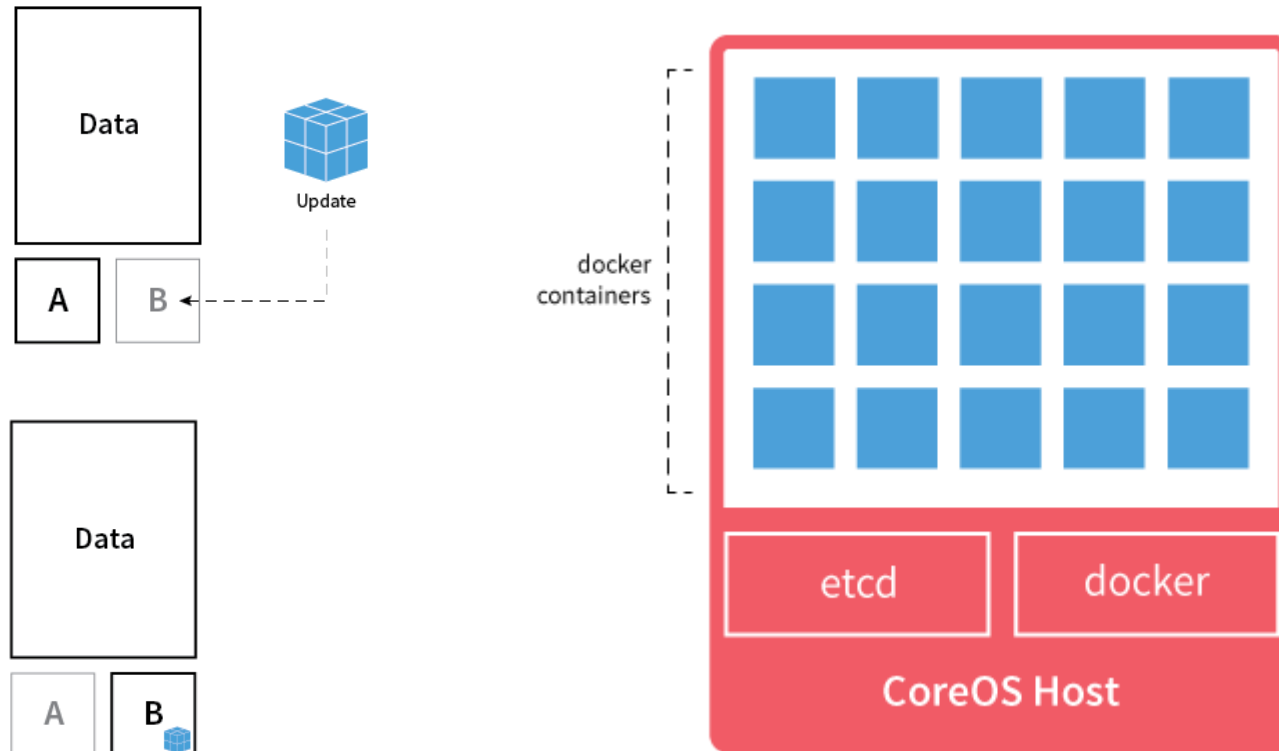
- Linux System & Service Manager

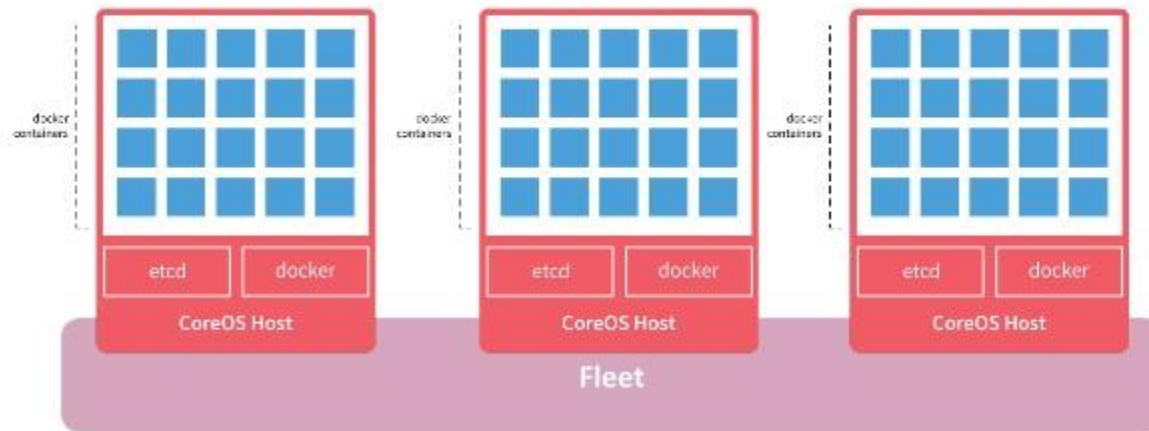
fleet

- etcd 와 systemd 를 이용한 분산 서비스 실행 시스템
- Docker 컨테이너를 임의의 호스트에 배포
- Load Balancing, AutoScaling
- CoreOS Host Discovery



DOCKER & COREOS





fleet

- etcd 와 system 를 이용한 분산 서비스 실행 시스템
- Docker 컨테이너를 임의의 호스트에 배포
- Load Balancing, AutoScaling
- CoreOS Host Discovery

4

FUTURE TRENDS IN DOCKER

DOCKER ECOSYSTEM



Service Providers



Dev Tools



Official Repositories



Operating Systems



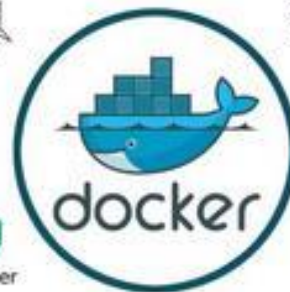
Configuration Management



Big Data



Service Discovery



Orchestration



System Integrators





Docker Orchestration 이란

- 여러 개의 컨테이너로 하나의 서비스를 구성 하는것

Docker Orchestration 필요한 이유

- Composition :
여러 종류의 컨테이너로 구성된 서비스 설정 및 연동
- Replication
Scalability, Fault Tolerance, High Availability
- Write Once Run Anywhere
같은 이미지가 Local/Cluster 에서 모두 동작할 수 있도록



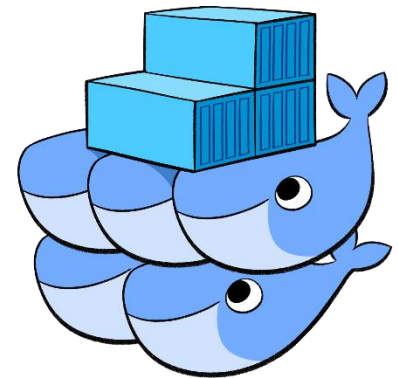
Docker Orchestration 구성요소

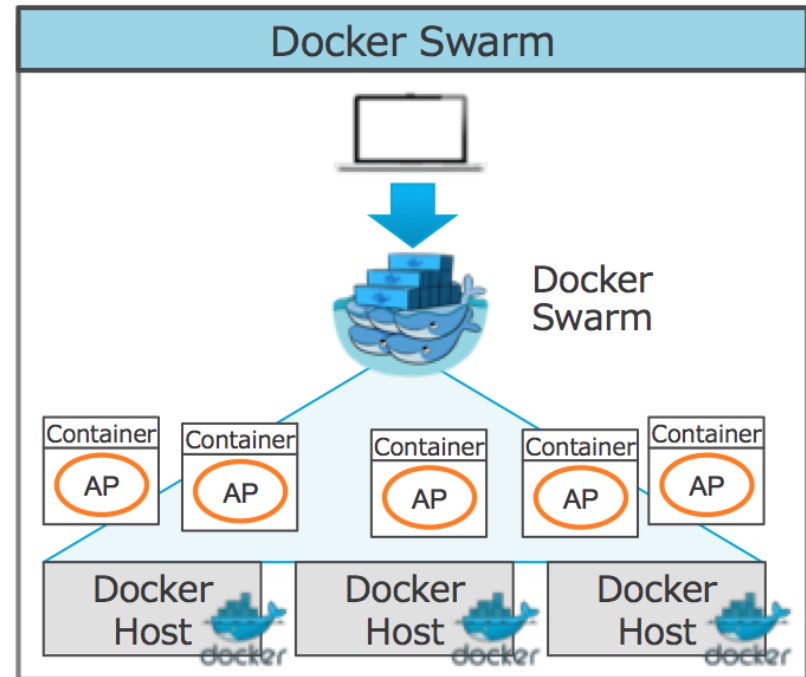
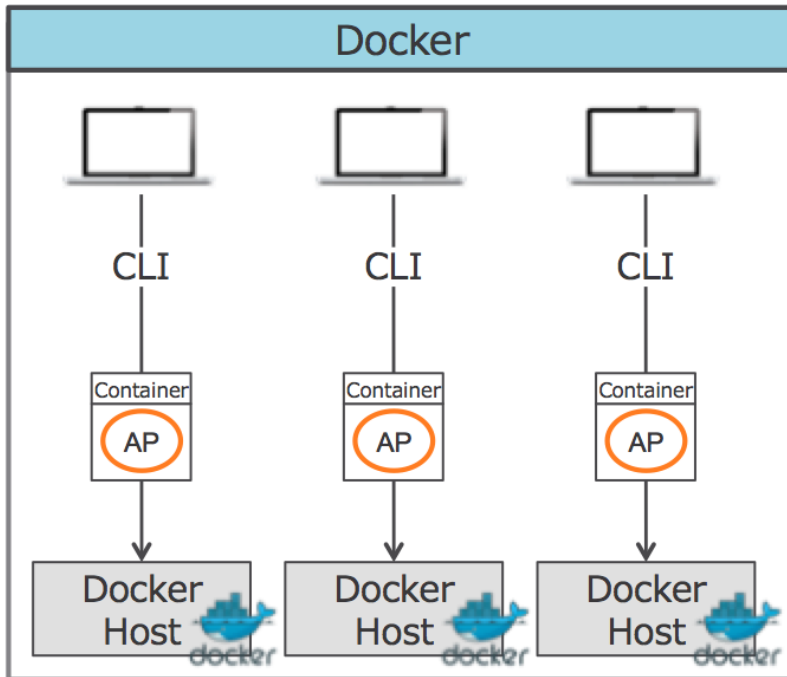
- Scheduling : 클러스터 노드에 컨테이너 실행 및 관리
- Networking : 클러스터 사이의 연결, 외부 네트워크 연결
- Discovery : 클러스터 노드 발견 및 그룹 관리
- Logging
- Monitoring
- Alerting



Docker Swarm

- 도커에서 공식 지원하는 클러스터링 도구
- 표준 Docker API 제공
- Failover & High Availability 지원
- 컨테이너 스케줄링
- 클러스터 노드 Discovery
- 플러그인 지원 (Mesos, Kubernetes, etc..)





Docker Swarm

- 별도의 agent 없이 컨테이너를 통해 동작
- 다른 툴들 (kubernetes, mesos,) 등은 별도의 agent 필요



Linux Container Cluster Manager

- 여러 호스트의 container 들을 클러스터링 하고 관리
- Google 의 컨테이너 관리를 위해 오픈소스로 발표 (2014.05)
- GO Language

특징

- 컨테이너 배치 : 리소스 요구사항 및 제약사항에 따른 컨테이너 배치
- 수평 확장 : CPU 사용량에 따른 스케일 업/다운 가능
- 스토리지 오케스트레이션 : Local, Network, Cloud Storage 지원
- 서비스 및 검색 로드밸런싱 : 컨테이너에 IP, DNS 네임 부여하여 로드 밸런싱 가능



kubernetes



Master

- API Server : 사용자의 request를 받는 http 서버
- Scheduler : 작업관리 및 api 서버 통신
- Controller Manager : pods 의 lifecycle을 관리 AutoScaling

Minion

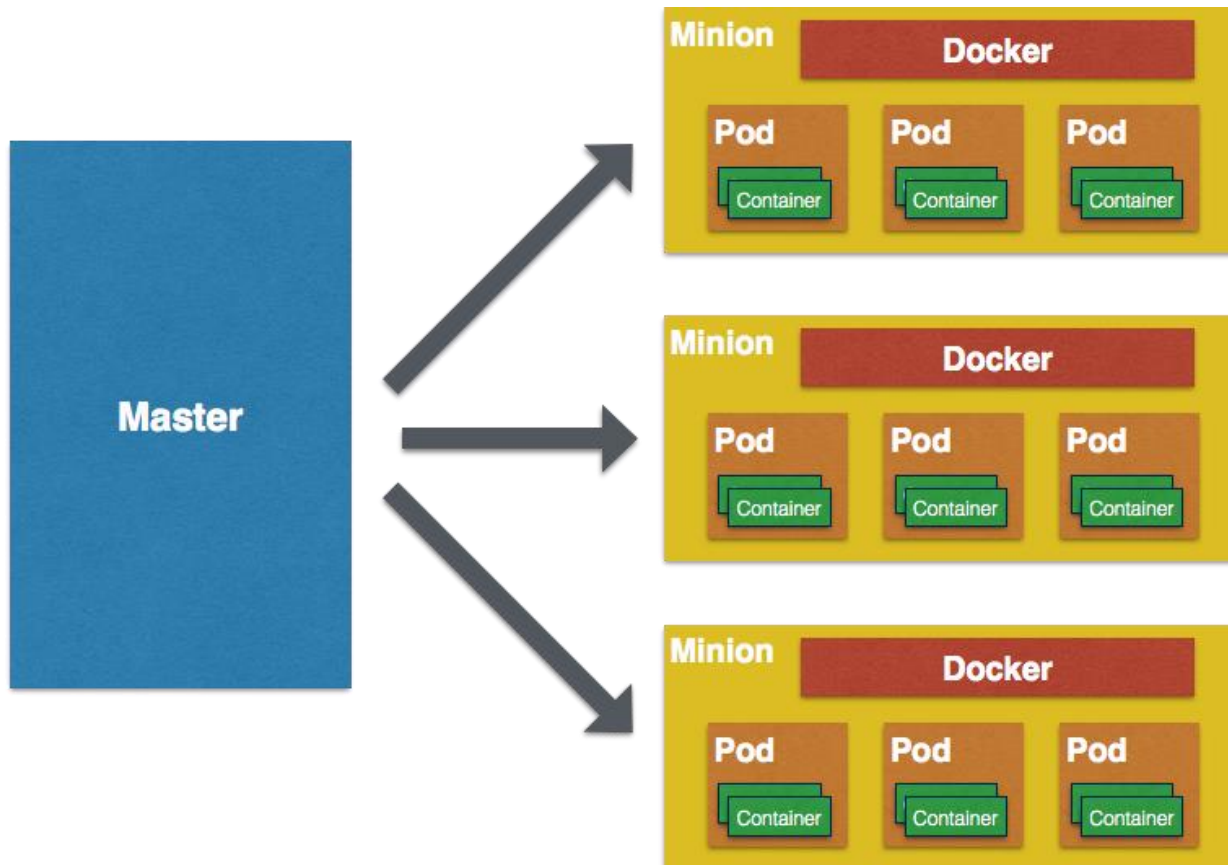
- Docker : Server 또는 Node
- Kubelet : minion 을 제어하는 agent, container 또는 pod 을 생성
- Pod : 여러 container 들의 묶음
- cAdvisor : container 리소스 모니터링
- Label : pod 에 이름을 붙임, 객체 그룹으로 관리

Key/Value Store

- CoreOS 의 etcd 를 사용하여 정보 저장



DOCKER & KUBERNETES





Magnum

- OpenStack API 서비스
- Container as a Service
- Docker, Kubernetes 를 first class resource 로 구성해줌
- OpenStack Heat 를 사용하여 컨테이너 이미지 오케스트레이션

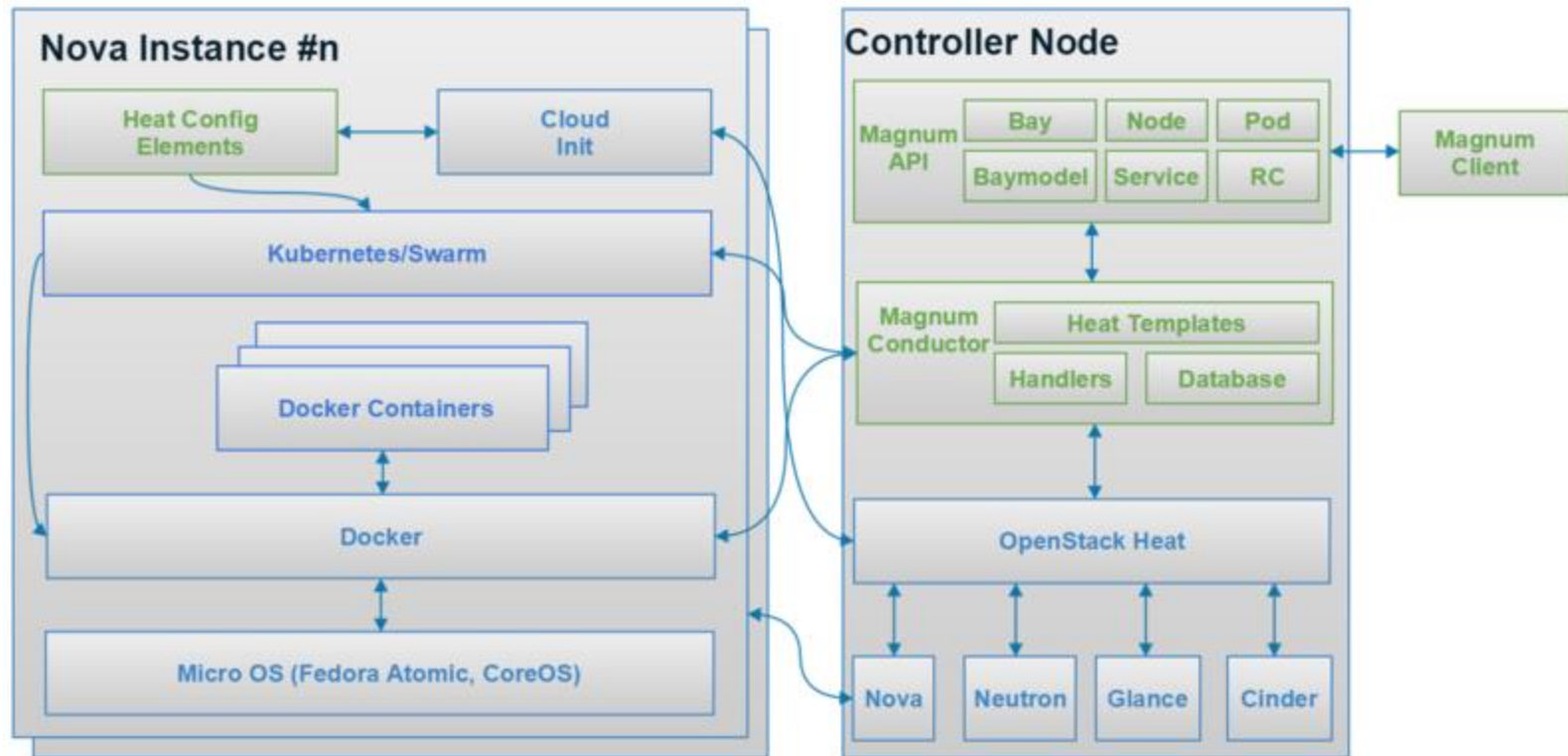
Nova

- OpenStack Compute 프로젝트
- Nova 의 가상 드라이버와 Docker 연동
- Docker 가 새로운 하이퍼바이저로 동작

Heat

- OpenStack Orchestration 프로젝트
- Docker 를 위한 Heat 플러그인 추가
- Docker API 를 이용하여 Heat 과 Docker 직접 통신

DOCKER & OPENSTACK



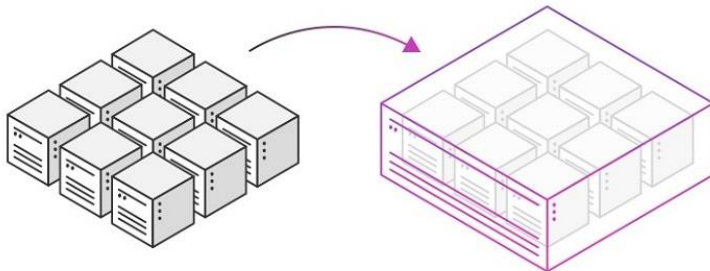


Mesosphere DCOS (Data Center Operating System)

- 컨테이너 오케스트레이션 솔루션
- 2016년 4월 오픈소스로 공개 (보안 부분 제외)

Combine Your Servers Into One Big Computer

Mesosphere combines your datacenter servers and cloud instances into one shared pool from which any app or framework can draw.

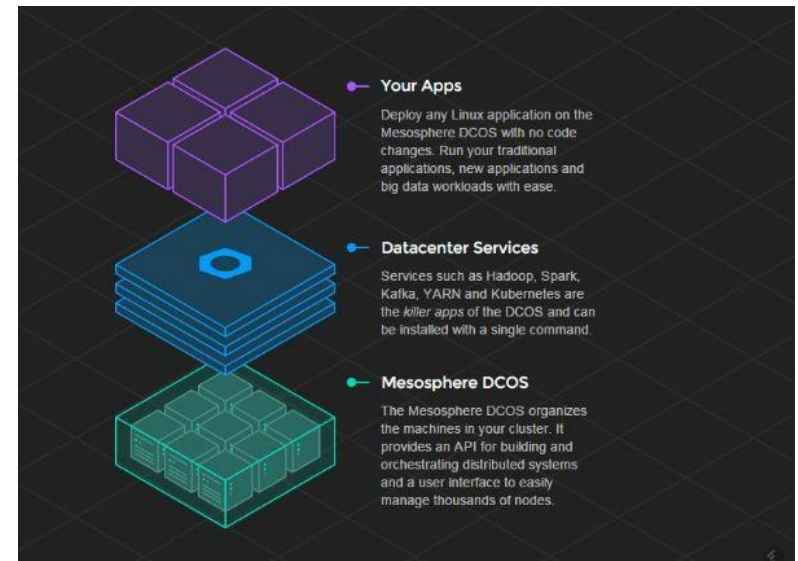


Datacenter or Cloud

Gone are the days where writing and deploying new applications means managing individual machines and static partitions.

With Mesosphere

Pool your datacenter and cloud resources, so all your apps run together on the same machines—reducing complexity and waste.



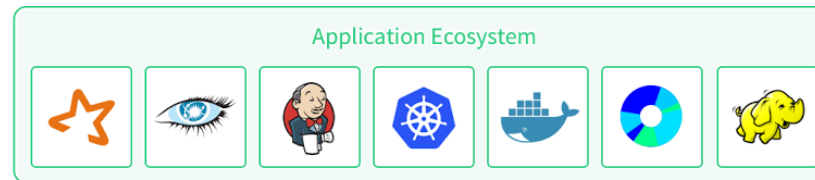


Mesosphere DCOS (Data Center Operating System)

- 컨테이너 오케스트레이션 솔루션

One Click Install

Easily roll out new datacenter services, from our strong and growing ecosystem.



Apache Mesos

Easily run and scale with a production proven distributed systems kernel.

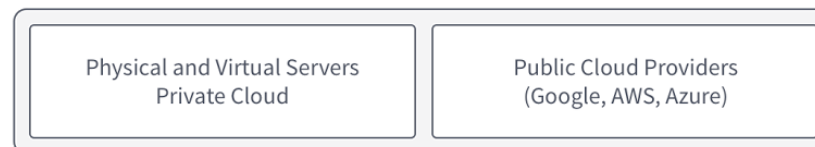


Mesosphere DCOS

Complement Mesos with enterprise capabilities such as security, monitoring, user interface, etc.

Works Everywhere

Run DCOS on any Infrastructure, and standardize your operational and application development experience everywhere.





Mesosphere = Mesos + Marathon + Chronos

- Mesos : 분산 시스템 커널
- Marathon : 컨테이너 관리, GUI 제공, long-running 스케줄러
- Chronos : batch, job, cron, short-running 스케줄러

The screenshot shows the Marathon web interface. At the top, there's a navigation bar with 'MARATHON', 'Apps', 'Deployments', 'About', and 'Docs'. Below this, the breadcrumb 'Apps > /mesosphere/cd-demo-app' is visible. The main section shows the application is 'Deploying'. There are buttons for 'Suspend', 'Scale', 'Restart App', and 'Destroy App'. Below this, there are tabs for 'Tasks' and 'Configuration'. The 'Tasks' tab is active, showing a table of tasks.

ID	Status	Version	Updated	Health
mesosphere_cd-demo-app.28324c96-d8b6-11e4-8df7-56847afe9799 10.84.71.25[31000, 31001]	Staged	a few seconds ago	4/1/2015, 2:29:19 PM	
mesosphere_cd-demo-app.5cf4cc35-d8b3-11e4-8df7-56847afe9799 10.147.80.133[31000, 31001]	Started	20 minutes ago	4/1/2015, 2:09:52 PM	

The screenshot shows the Chronos web interface. At the top, there's a search bar with 'hostings' entered. Below this, there are statistics: '256 TOTAL JOBS' and '17 FAILED JOBS'. There are buttons for 'Dependency Graph' and 'New Job'. The main section is a table of jobs.

NAME	GRAPH	LAST
create_airbed_dump_table_hostings		SUCCESS
create_airbed3_dump_table-hostings_first...		SUCCESS
create_airbed_dump_table_hostings_with...		SUCCESS
create_airbed_dump_table_collection_hos...		SUCCESS
create_omg_table-affiliate_events_hostings		SUCCESS
hostings_summary		SUCCESS
daily_gibson-import_airbed3_hostings		SUCCESS
db_export-airbed_hostings		SUCCESS
hostings_summary_2_quality_score		SUCCESS
hostings_summary_1_pre		SUCCESS
hostings_impressions_normalize		SUCCESS
hostings_impressions_normalize_prepare		FAILURE
daily-update_hostings_summary_history		SUCCESS
hostings_earnings_summary#async		SUCCESS
daily-create_hostings_history		SUCCESS
daily-update_hostings_history		SUCCESS
daily-create_hostings_summary_history		SUCCESS
db_export-airbed_collection_hostings		SUCCESS



2014/11/17 AWS ECS 서비스 발표 일주일 뒤 기사

"All the cloud providers at this time run containers inside a virtual machine," said Alex Polvi, CEO of CoreOS, which produces the slimmed-down CoreOS version of Linux expressly for running containers.

모든 클라우드 제공자가 VM 내부에서 컨테이너를 실행하고 있다.

... 중략 ...

Right now, container isolation as we know it poses too many risks of one workload inadvertently (or otherwise) glimpsing data or communications of another.



So what is the problem? Why don't containers contain?

The biggest problem is everything in Linux is not namespaced. Currently, Docker uses five namespaces to alter processes view of the system: Process, Network, Mount, Hostname, Shared Memory.

리눅스에서 제공하는 5개의 Namespaces로는 Security를 보장 못함

2014/7/22 **Are Docker containers really secure?**

<https://opensource.com/business/14/7/docker-security-selinux>

2015/4/28 **Containers vs. virtual machines: How to tell which is the right choice for your enterprise**

<http://www.itworld.com/article/2915530/virtualization/containers-vs-virtual-machines-how-to-tell-which-is-the-right-choice-for-your-enterprise.html>



Joyent

- Container 를 동작시키기 위한 별도 Container Hypervisor 를 제작
- Bare metal 성능 + VM 확장성
- 완벽히 격리된 멀티 테넌트 환경 컨테이너 제공
- SmartOS
- Public + private cloud service



BARE METAL CLOUD



IBM, Rackspace, Internap

- 성능에 민감한 워크로드를 위한 비 가상화 클라우드 서비스



AWS Dedicated Host & Instance

- 전용 호스팅 물리적 서버 선택가능
- Bare Metal cloud 서비스제공보다 Bare Metal 에 가까운 성능을 내려고 함

5

CONCLUSION



- 기존의 개발 방식과 다른 새로운 형태의 개발, 운영 방식 등장 - DevOps
- 컨테이너 기술을 이용하여 애플리케이션 + 서비스 운영환경을 함께 배포
- Container 와 VM 은 상호 보완적인 관계

CONCLUSION



기존의 개발 방식과 다른 새로운 형태의 개발, 운영 방식 등장

- 클라우드 기반 개발 : 자원을 가상화 하고 효율적으로 활용
- DevOps : 개발, 배포, 확장까지 고려

컨테이너 기술 활용

- 애플리케이션과 애플리케이션 환경을 함께 배포

Container 와 VM 은 상호 보완관계



***Docker** 로 모든 어플리케이션을 컨테이너로 만든다면
서버는 **Docker** 를 돌리기 위해 존재할 뿐*

REFERENCE



<https://docs.docker.com/>

<http://devview.kr/2014/session?seq=20>

<http://www.slideshare.net/rkawxms/docker-container>

<http://www.slideshare.net/devview/221-docker-orchestration>

<http://www.slideshare.net/subicura/coreos-38279596>

<http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=GMW14061KRKO&attachment=GMW14061KRKO.PDF>

<http://www.networkworld.com/article/3011453/public-cloud/would-amazon-ever-offer-a-bare-metal-public-cloud.html>

<http://www.informationweek.com/cloud/infrastructure-as-a-service/amazons-container-strategy-examined/a/d-id/1317515>

감사합니다.

