

# 클라우드의 이해

2018.11

**SMART**  
**2018 TOGETHER**

# Index

- 01 Trend
- 02 Cloud Model
- 03 PaaS-TA
- 04 Application on Cloud
- 05 Practice

**SMART**  
**2018 TOGETHER**

# 01 Trend

글로벌 클라우드 시장과 국내 동향

**SMART**  
**2018 TOGETHER**

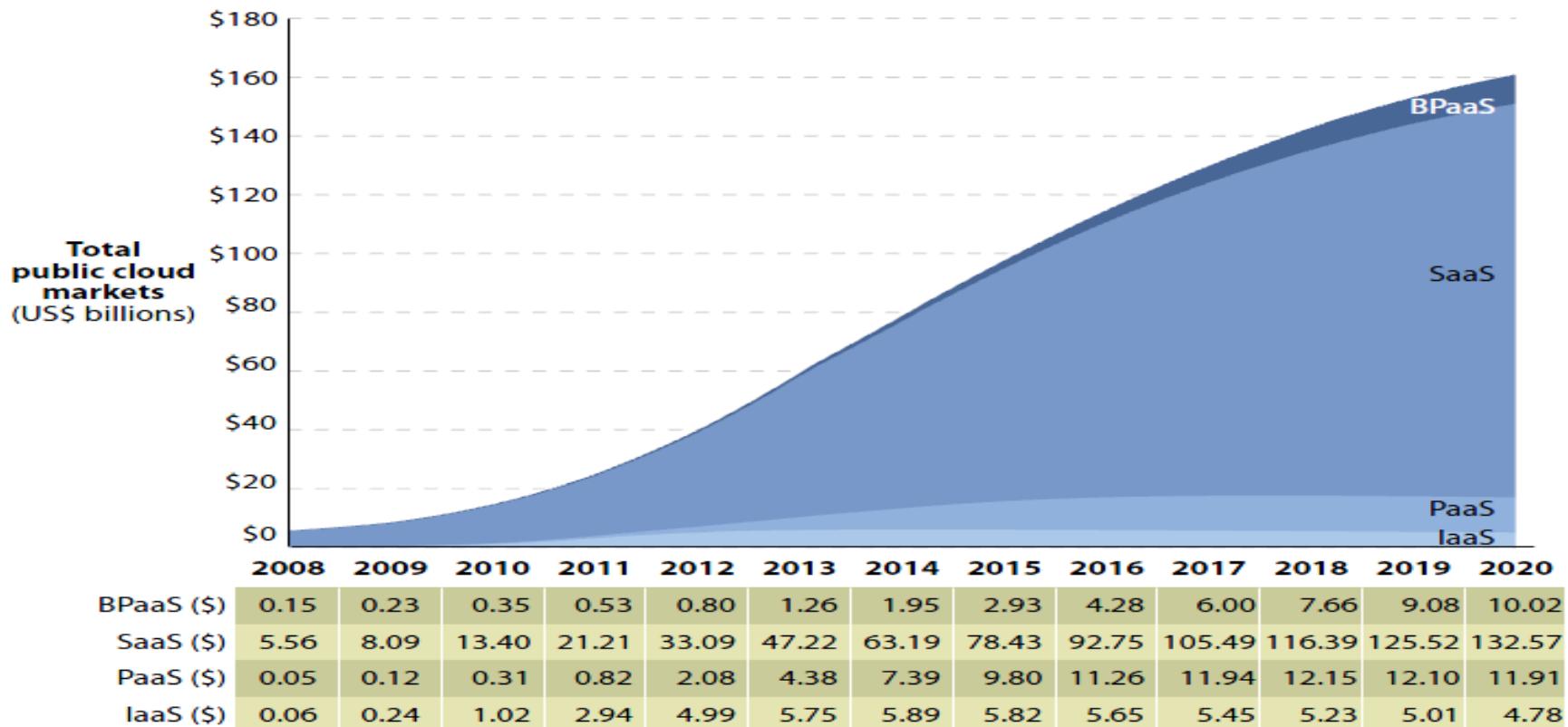


# 01 Trend – 클라우드 시장 규모

- Forrester Research에 따르면 2020년경 Public 클라우드 시장 규모는 1600억 달러를 돌파할 것으로 예상됨
- 그 중 대부분은 SaaS가 차지할 것으로 보이며, 시장과 사용자의 다채로운 요구사항을 즉각적으로 서비스화 하기 위해서는 클라우드 컨테이너의 자동화 관리와 조절이 필수

**Figure 3 Forecast: Global Public Cloud Market Size, 2011 To 2020**

The spreadsheet detailing this forecast is available online.



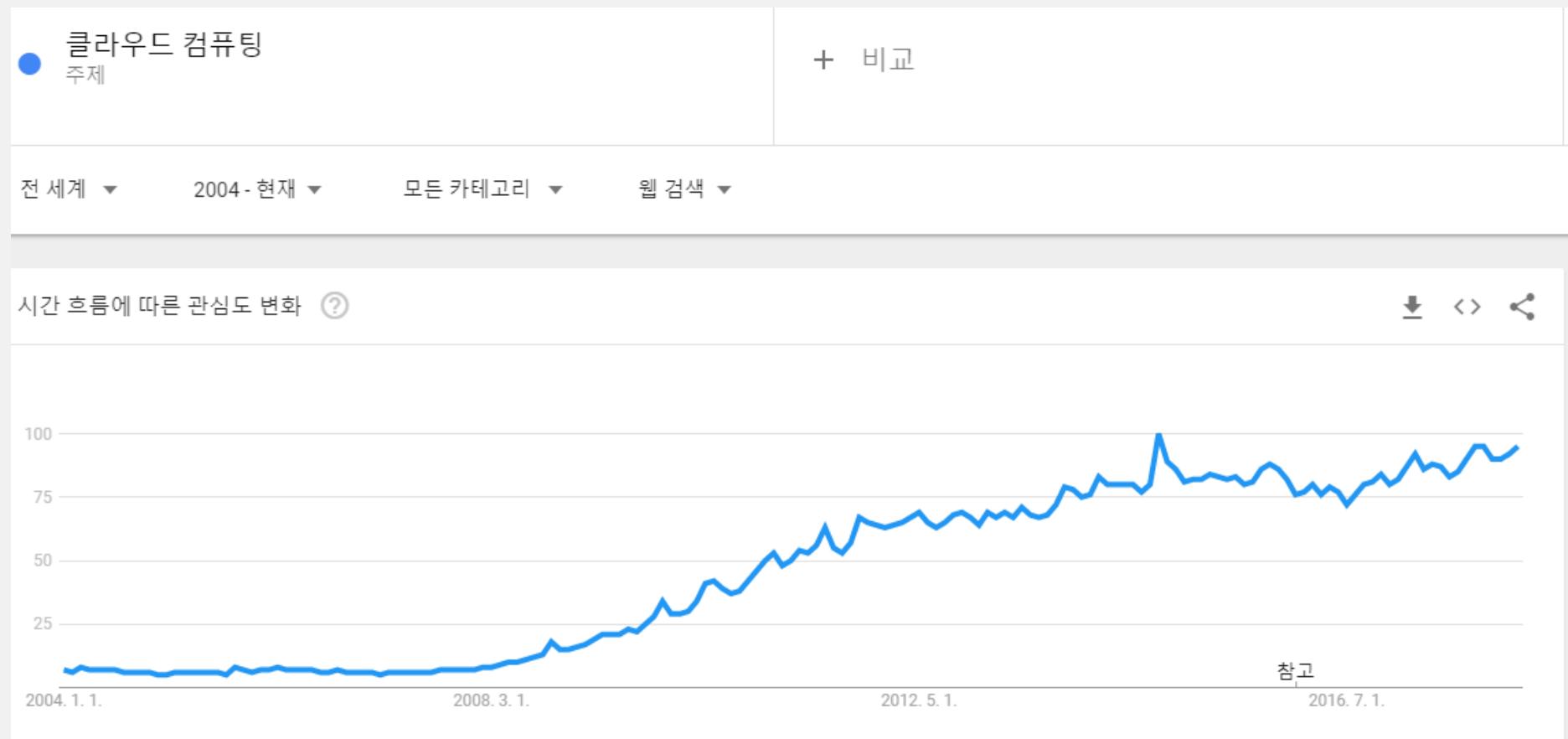
58161

Source: Forrester Research, Inc.



# 01 Trend – 전세계적 트랜드

이미 2008년 이전부터 시작되어 전 세계적 주요기술로 자리매김.





# 01 Trend- 이미 실생활에 만연한 클라우드

이미 우리는 사용하고 있었다!!

주로 이미지, 문서 백업용으로 활용 → 클라우드 프로바이더로 성장

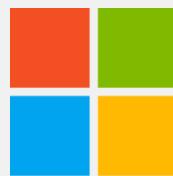
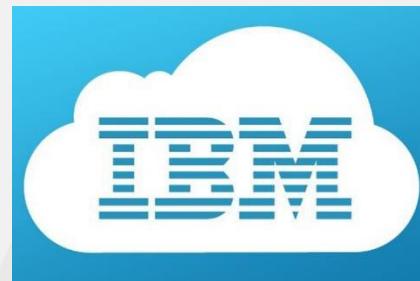


출처 ZDNet Korea



# 01 Trend - 시장을 이끄는 Contributors

대형 IT 선도 업체의 후원으로 지속적으로 발전되고 있음.



Microsoft

Google

AMD

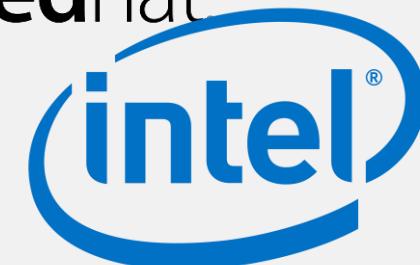
docker

ORACLE®

verizon

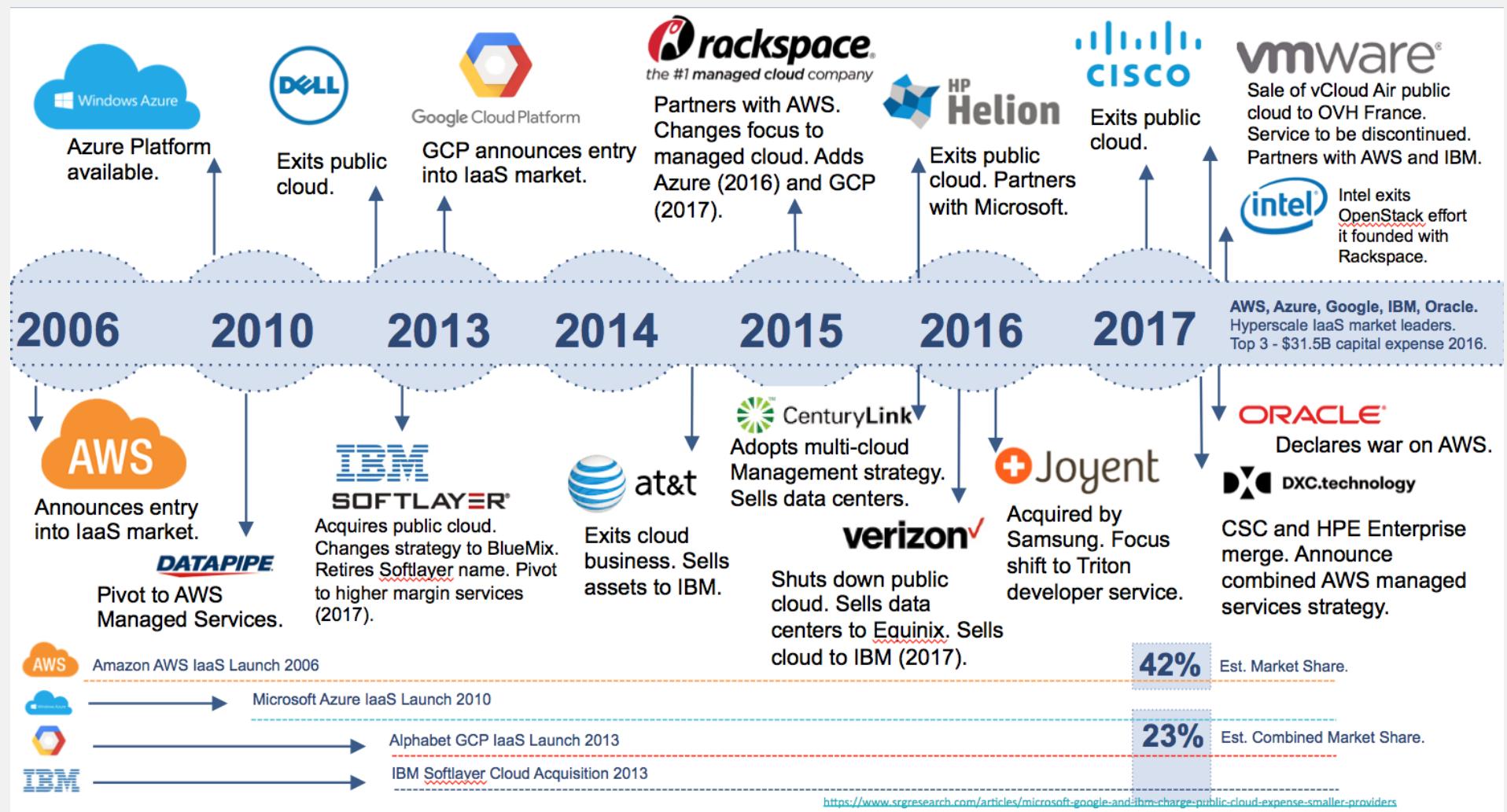


redhat





# 01 Trend - 클라우드의 역사

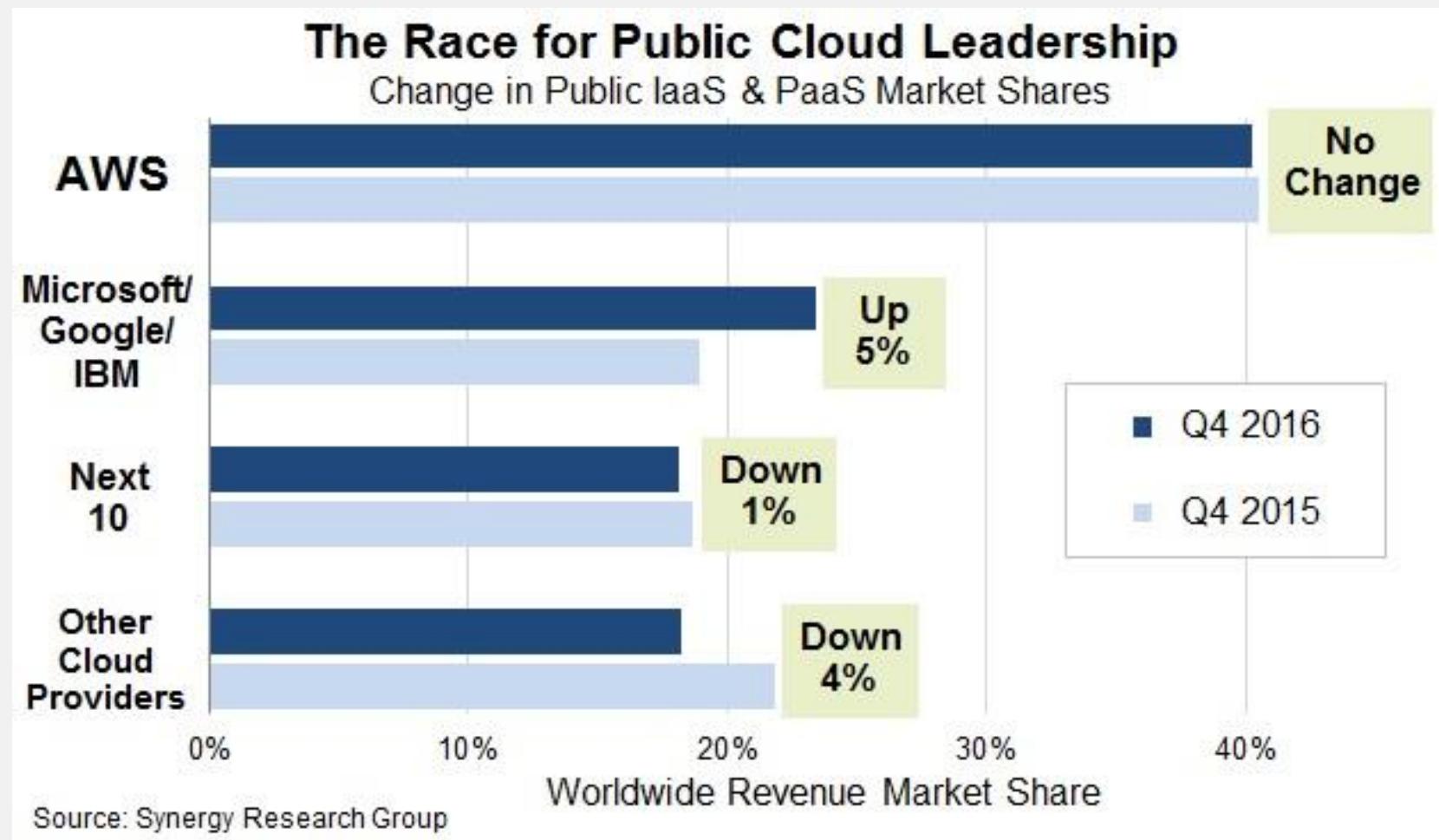


Source : <https://medium.com/@dougpollei/there-can-be-only-one-37667ff80647>



# 01 Trend - 시장을 이끄는 주요 자본들

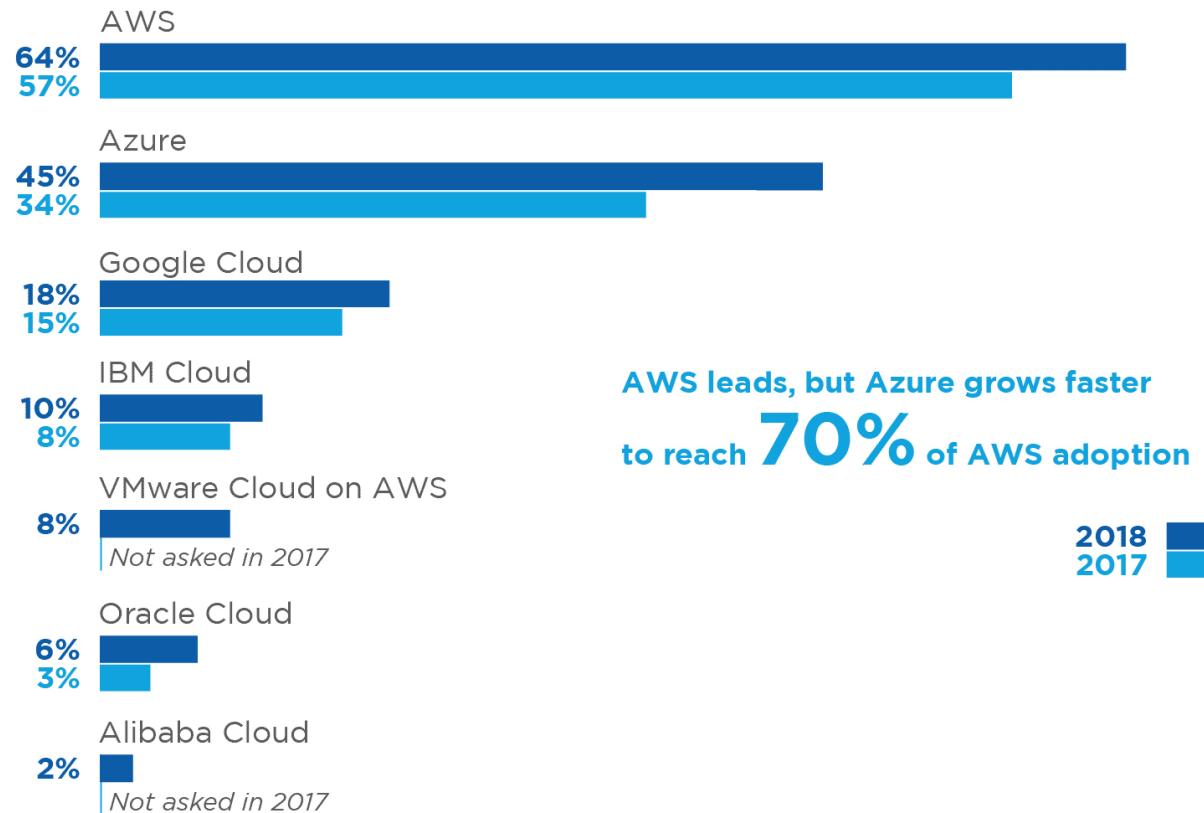
Public cloud를 이끄는 주요 4강





# 01 Trend - 시장을 이끄는 주요 자본들

Respondents Running Apps 2017 vs. 2018



Source: RightScale 2018 State of the Cloud Report

# 02 Cloud Model

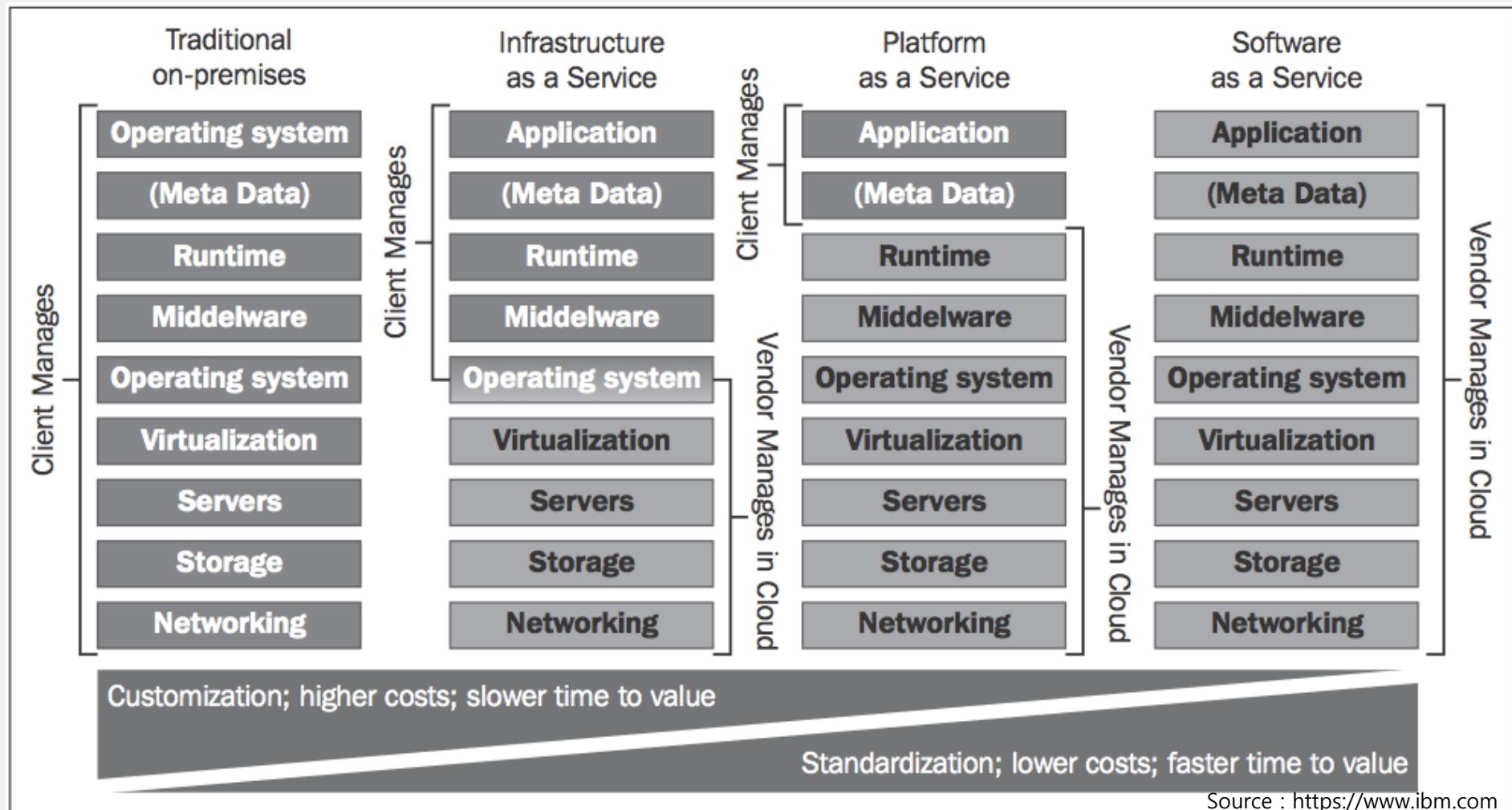
IaaS, PaaS, SaaS

**SMART**  
2018 TOGETHER



## 02 As a Service – 모델별 관리 범위

서비스로 제공한다 = 사용에 따른 요금을 부과한다

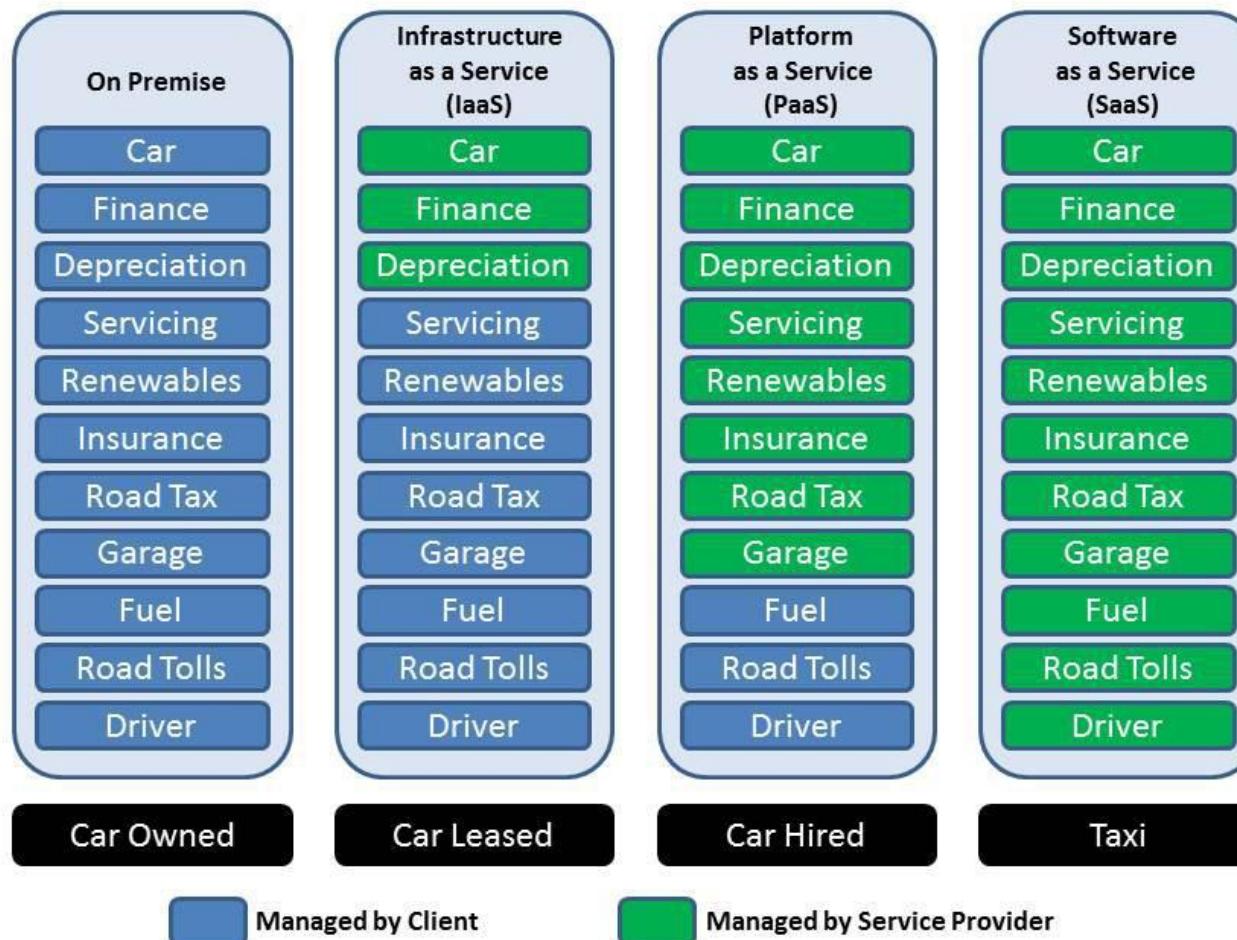




## 02 As a Service – 클라우드 모델 예시

단순한 예로 보는 as a Service

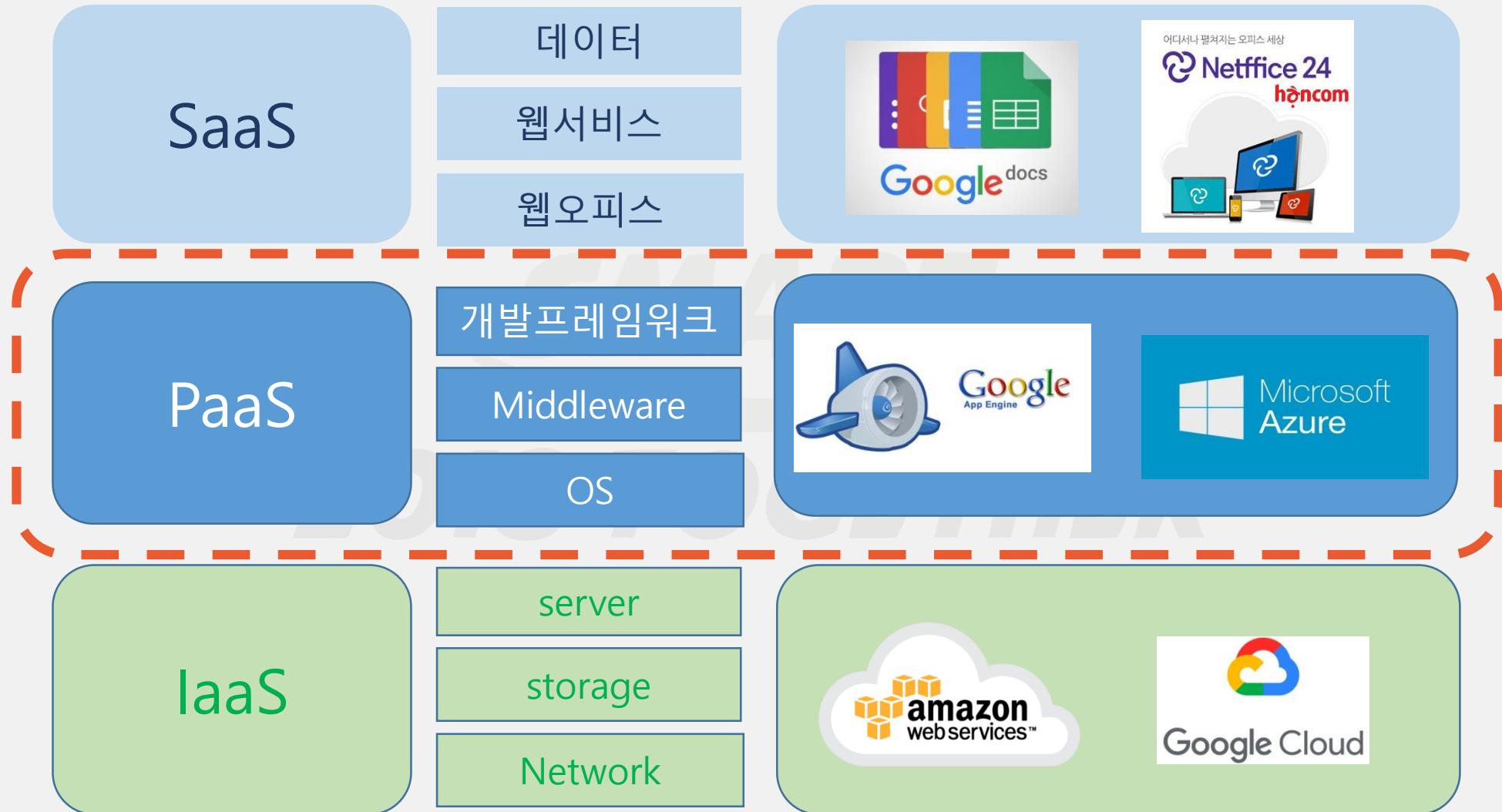
### Car as a Service(CaaS)



Source : <http://blog.reforceinternational.com/post/143438992023/caas-car-as-a-service>



# 02 As a Service



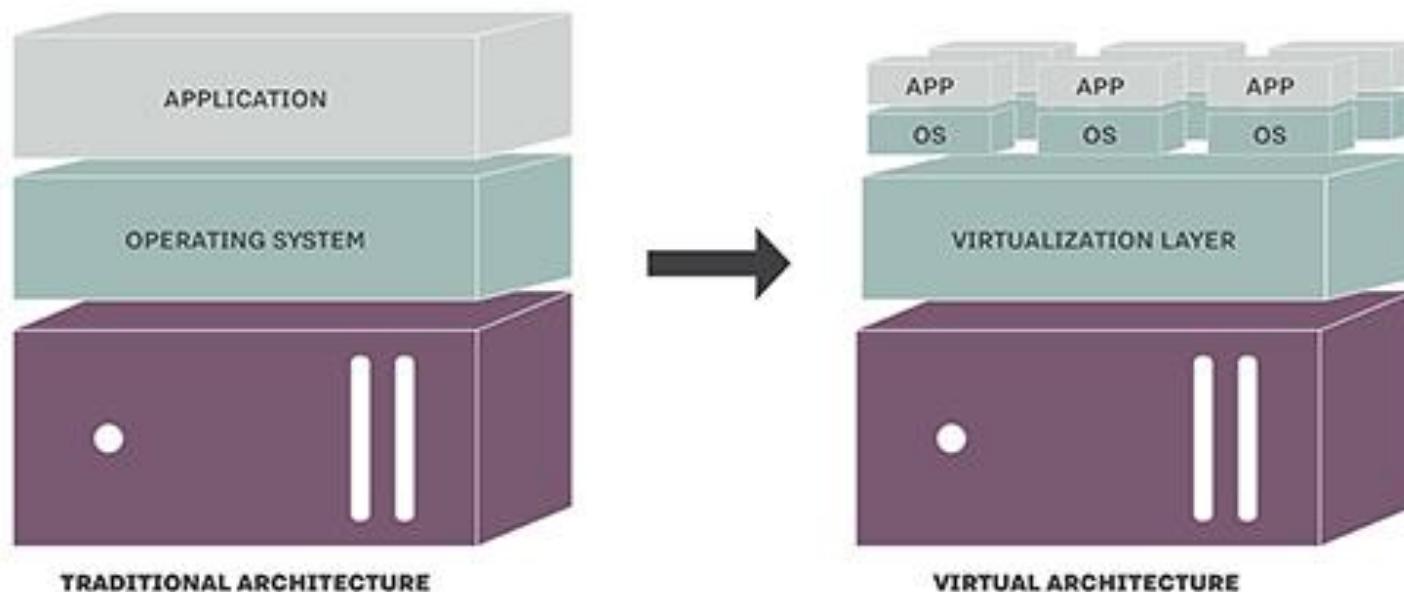


## 02 As a Service - IaaS

### Virtual Architecture

물리적 리소스를 가상화하여 유연한 Infrastructure를 제공

### TRADITIONAL AND VIRTUAL ARCHITECTURE



Source : <http://whatis.techtarget.com/definition/virtualization-architecture>

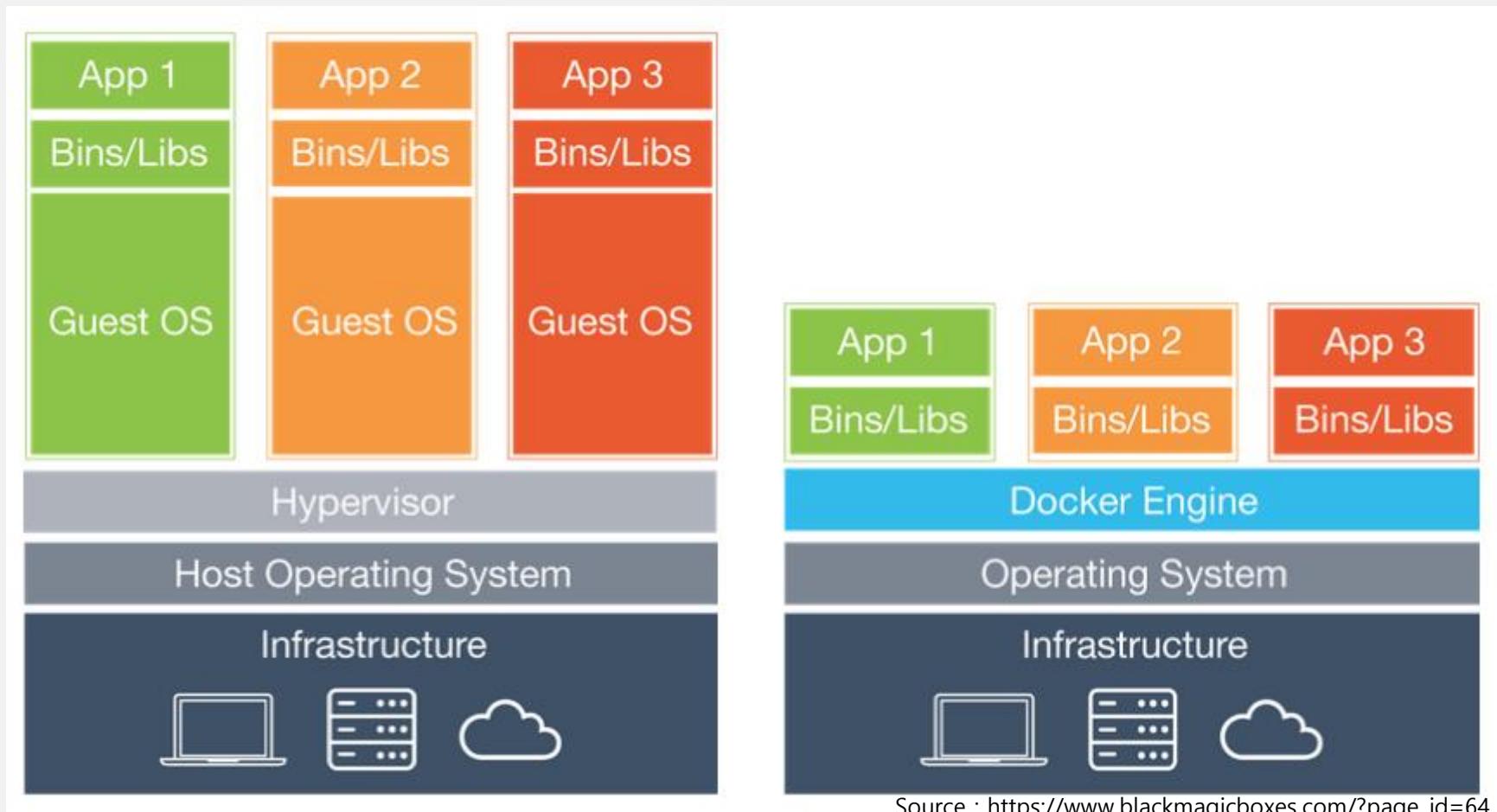




# 02 As a Service - IaaS

## Virtualization

### hypervisor vs container

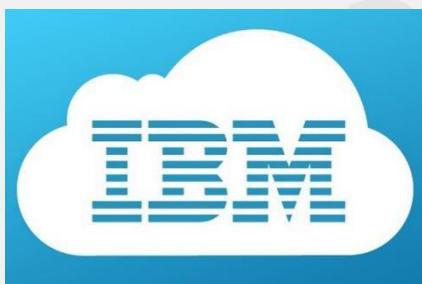




# 02 As a Service - IaaS

Infrastructure as a Service

Public IaaS provider & Project



[OpenNebula.org](http://OpenNebula.org)

[apachecloudstack™](http://ApacheCloudstack.org)

 [openstack](http://OpenStack.org)®

 [EUCALYPTUS](http://Eucalyptus.com)



# 02 As a Service - IaaS

## OpenStack

인스턴스 - OpenStack Dashboard

주의 요함 | 45.248.75.138/horizon/project/instances/

ubuntu® Default ● penne admin

프로젝트 Compute 개요 인스턴스 블룸 이미지 키 페어 API 액세스 네트워크 관리자 인증

프로젝트 / Compute / 인스턴스

### 인스턴스

Displaying 20 items | Next »

인스턴스 이름	이미지 이름	IP 주소	Flavor	키 페어	Status	가용 구역	작업	전원 상태	생성된 이후 시간	Actions
delivery-pipeline-binary-storage-api/e236a87e-4544-425b-98e4-b6ffac8df309	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.78	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
ci_server/b4dd2da4-45bc-4cda-ac00-c8c64c07e4d9	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.72	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
binary_storage/4a488951-dd10-425f-8a55-8fa14d930944	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.83	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
delivery-pipeline-scheduler/49072f05-ca72-4284-b15f-88290557a987	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.82	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
mariadb/c790bb4e-3cf4-478d-beb5-84aeb70f152d	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.73	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
inspection/5f20d60c-e63c-4bdf-b287-73a88a7e44b1	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.75	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
delivery-pipeline-inspection-api/6cfcd27ab-c664-4684-9dc7-82ad6ccf2129	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.77	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
ci_server/5a5e6a3f-0fee-4491-b76a-0632aa4ec517	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.71	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
delivery-pipeline-ui/df6e3f4e-60a6-4203-80d3-b39b5f14e05c	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.81	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>
delivery-pipeline-api/e0a9f0d1-9db6-4705-aa21-1fa5fff96605	bosh-openstack-kvm-ubuntu-trusty-go_agent/3445.2	10.2.40.79	m2.small	Inception-key	Active	nova	None	Running	2주, 6일	<button>스냅샷 생성</button>



# 02 As a Service - IaaS

## nhn- Toast

TOAST CONSOLE

https://console.toast.com/project/a7buR0nG/compute/instance#management

TOAST Console 조작-인프라닉스(주) playpark 서비스 선택 한국 리전 | 한국어 | In\*\*\*@cloud.or.kr

playpark Instance 관리 URL & Appkey 사용자 가이드

관리 키 페어 플로팅 IP 보안 그룹 Block Storage Image

인스턴스 이름 이름을 입력해 주세요. 검색 API 엔드포인트 설정

+ 인스턴스 생성 인스턴스 타입 변경 이미지 생성 플로팅 IP 관리 보안 그룹 변경 인스턴스 재부팅 ...

총 50개 < 이전 다음 >

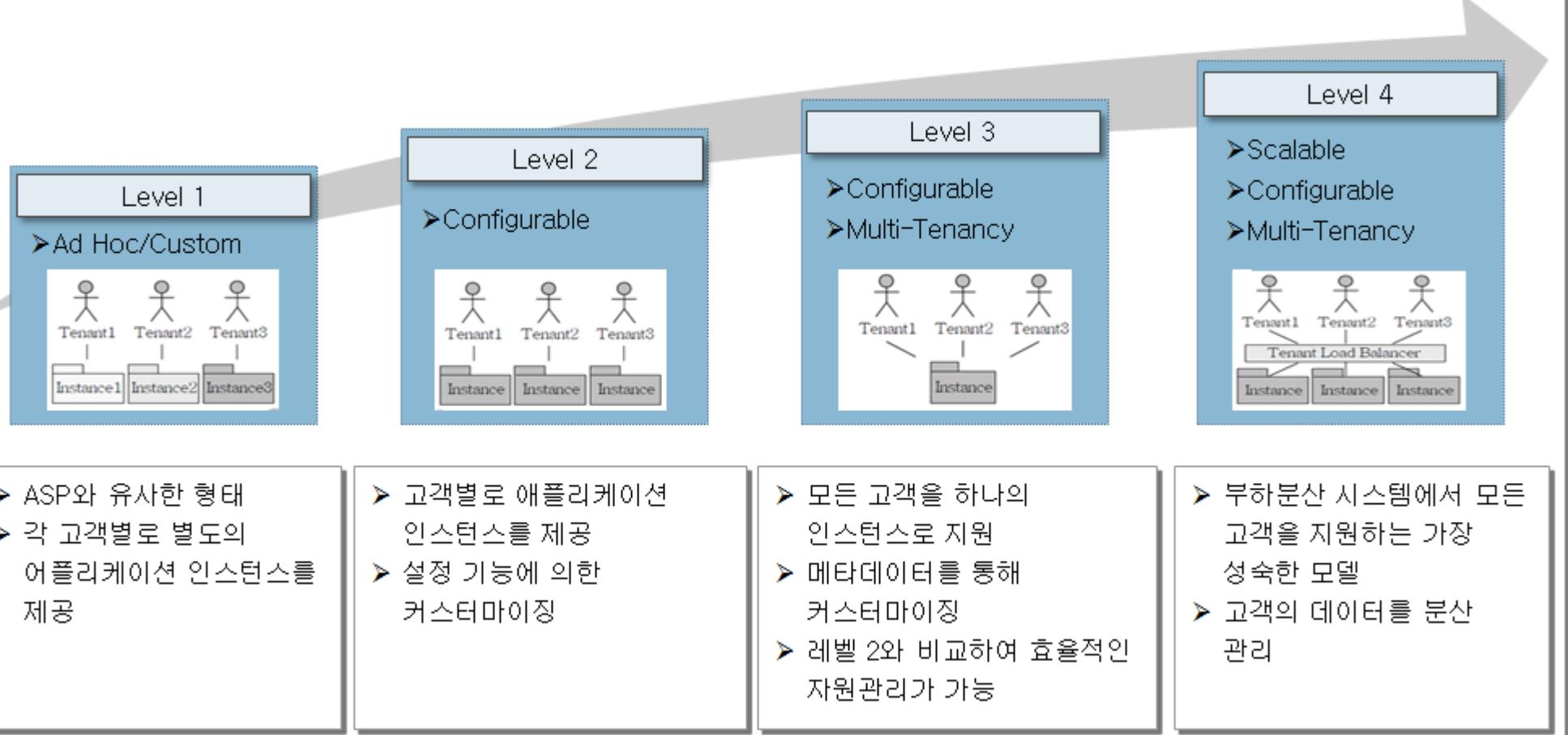
이름 ↑	OS	IP 주소	인스턴스 타입	키 페어 ↑	가용성 영역 ↑	상태 ↑
vm-7fc9cfc6-d83a-44c2-b011-85b52...	ubuntu	10.10.3.131, 133.186.152.187	m2.small (1Core, 2GB)	playpark_key	kr-pub-a	●
vm-899c6baf-1f93-4c80-9914-29885...	ubuntu	10.10.3.127, 133.186.134.105	m2.small (1Core, 2GB)	playpark_key	kr-pub-a	●
vm-d77b1b0d-c3df-4a73-827e-4cbd...	ubuntu	10.10.3.112, 133.186.146.81	m2.small (1Core, 2GB)	playpark_key	kr-pub-a	●
vm-5cd2e0fc-9c6f-4dc8-b03a-e27f2...	ubuntu	10.10.3.111, 133.186.240.196	m2.small (1Core, 2GB)	playpark_key	kr-pub-a	●
vm-c92f1345-45f6-4cb0-85a7-5214f...	ubuntu	10.10.3.113, 133.186.135.185	m2.small (1Core, 2GB)	playpark_key	kr-pub-a	●
vm-7ce419f6-f9db-4333-8012-de04a...	ubuntu	10.10.3.138, 133.186.159.30	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-af75429a-ac20-496f-99cd-bb313...	ubuntu	10.10.3.124, 133.186.153.136	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-cdb1da91-c1ea-456a-969c-32b6...	ubuntu	10.10.3.136	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-962e0c6e-8121-4087-ac80-8fcfa...	ubuntu	10.10.3.134	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-ed4be659-2ca3-425a-bbdd-ca8...	ubuntu	10.10.3.135	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-423d8af7-dbf6-4e98-beeb-208d...	ubuntu	10.10.3.137	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-867b3d38-a4a1-4fc6-91aa-0aeb...	ubuntu	10.10.3.133	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●
vm-1efd7ac9-26dc-4b77-91b2-67a5...	ubuntu	10.10.3.132	m2.medium (2Core, 4GB)	playpark_key	kr-pub-a	●

이용 약관 개인 정보 처리 방침 © NHN Entertainment Corp. All rights reserved.



# 02 As a Service - SaaS

## Software as a Service



성숙된 SaaS는 Multi Tenant, Configuration, Scalability 충족



# 02 As a Service - SaaS

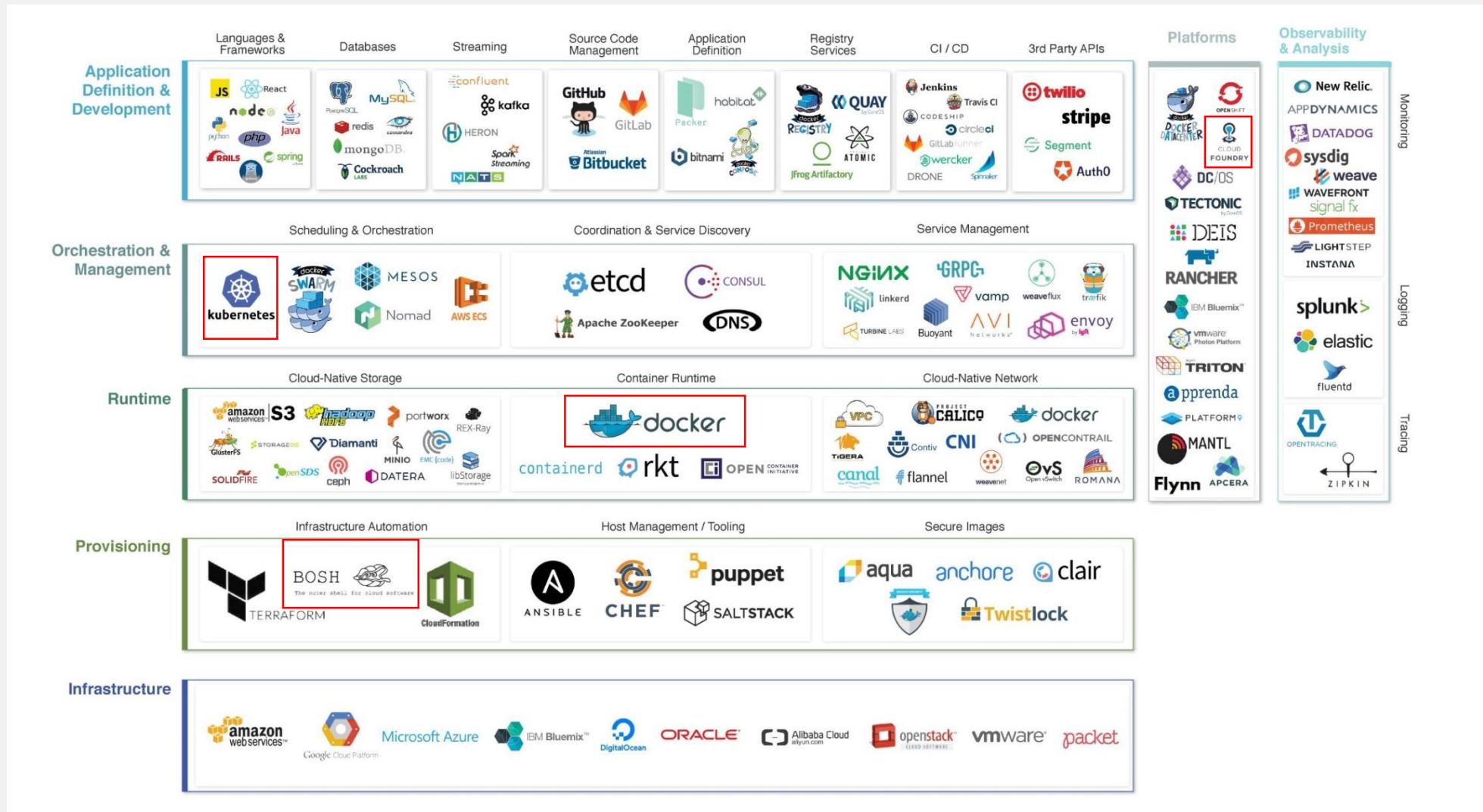
우리 주변의 Software as a Services





# 02 As a Service - CaaS

## Container native landscape





# 02 As a Service - CaaS

## Orchestration Service 추이

### Kubernetes의 급성장

- 대표적인 Container Orchestration 솔루션은 Docker Swarm, 아파치 재단의 Mesos, 그리고 과거 구글의 소유였다가 현재는 리눅스 파운데이션 재단으로 기증된 Kubernetes, Pivotal 의 Bosh 등이 있음
- Container 기술은 **Kubernetes** Orchestration 플랫폼의 등장으로 발전을 촉발 시킴
- 현재 Kubernetes는 대부분의 클라우드 서비스 제공자들이 서비스를 하고 있으며, Docker와 Pivotal Cloud Foundry 또한 Kubernetes 지원하기 시작하였고 자체적인 Orchestration을 제공하던 AWS 또한 Kubernetes 서비스를 발표





# 02 As a Service - CaaS

## Kubernetes란?

- Container Orchestration
- Container화 된 애플리케이션의 배포, 확장 및 관리를 위한 오픈 소스 플랫폼
- Docker Datacenter, Mesosphere DC/OS 등과 유사
- "쿠버네티스", "쿠베르네테스", "K8s", "쿠베", "쿠버", "큐브"
- 구글 설계 후, 현재 리눅스 재단에서 관리
- Go 언어로 작성
- Apache License 2.0
- 패키징, 배포, 서비스 검색, 확장 및 롤링 업그레이드를 위한 개방적이며 효율적인 모델 제공
- 자동 배치, 자동 재시작, 자동 복제, 자동 확장으로 애플리케이션 상태 확인 및 셀프 복구 수행
- 컨테이너화 된 애플리케이션과 해당 리소스의 즉시 확장 기능 지원
- 스토리지 장착 및 추가로 스테이트풀 애플리케이션 실행 가능(e.g. MySQL, Redis)
- 컨테이너를 Pod으로 분류, 컨테이너 급증 등의 문제 해결
- 효과적인 하드웨어 활용으로 리소스 극대화, 엔터프라이즈 애플리케이션 실행에 용이
- Atomic Registry, Open vSwitch, heapster, OAuth, SELinux 등 OSS를 이용, 컨테이너 인프라 모든 부분의 Orchestration 가능





# 02 As a Service - CaaS

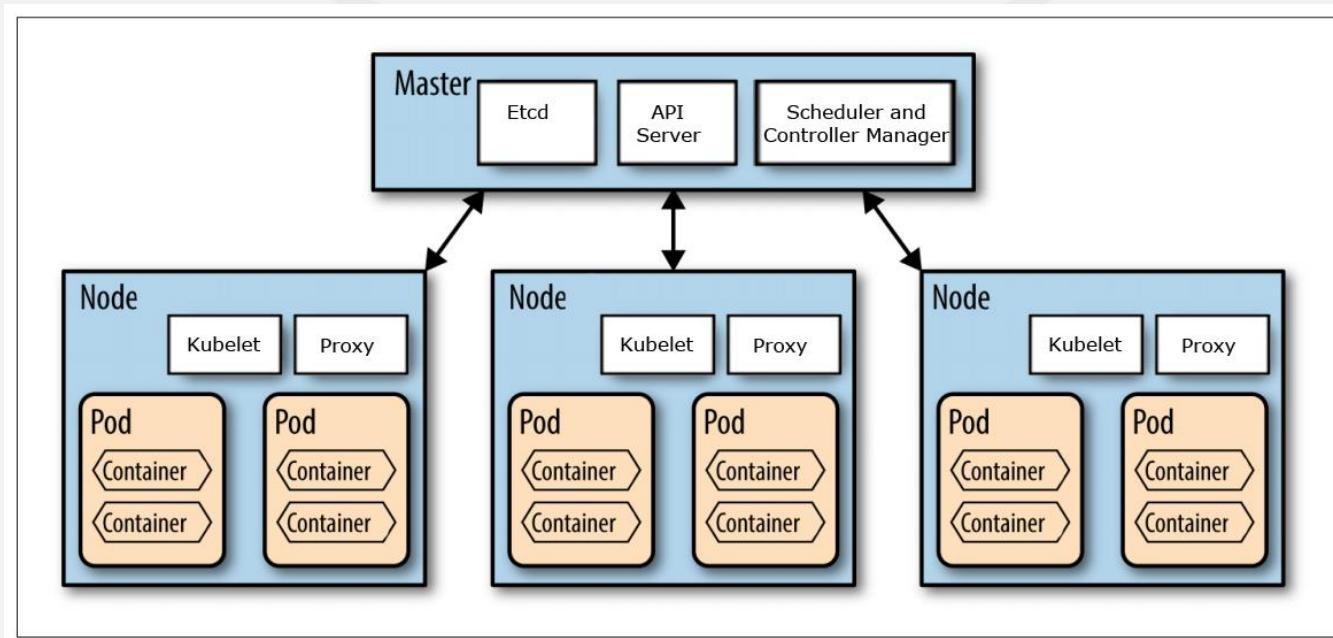
최소 1개의 Master와 n개의 Node(Worker)로 구성

## 마스터

- 전체 클러스터 관리

## 노드

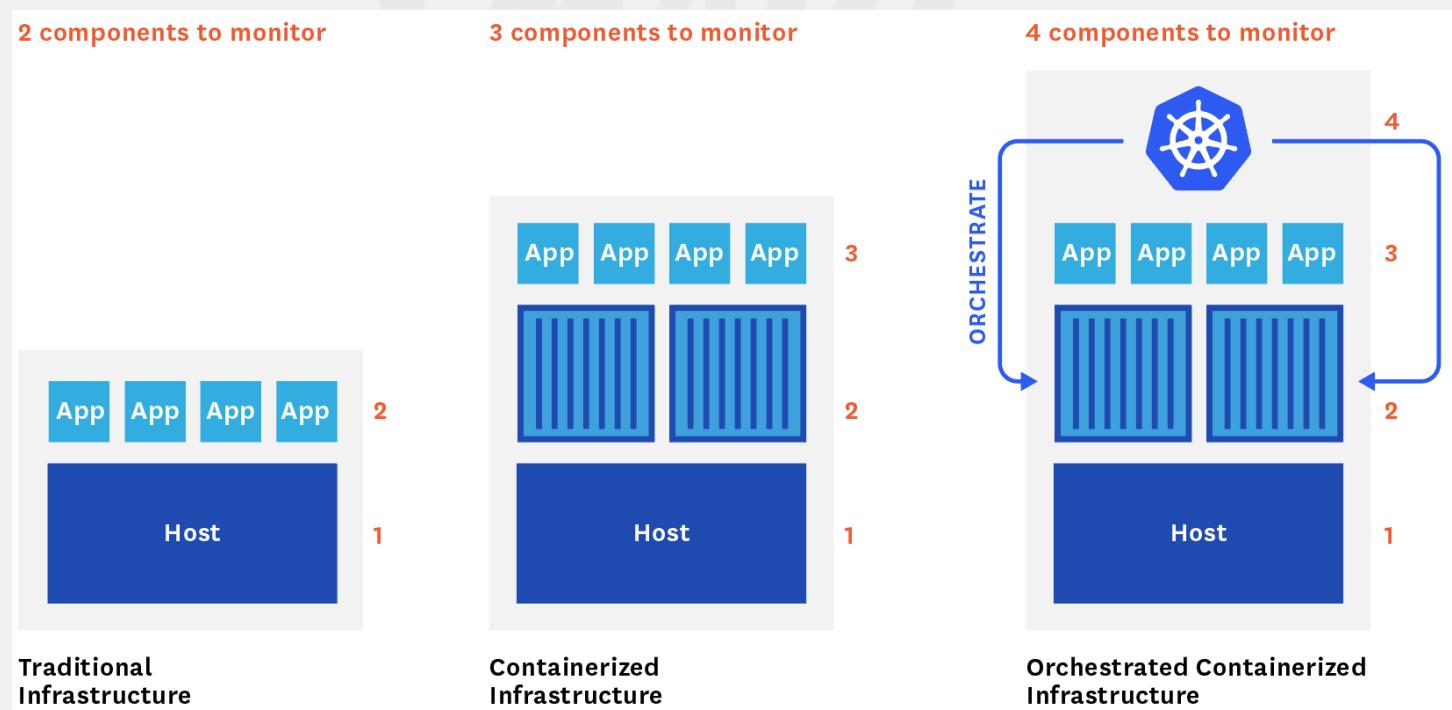
- 마스터와 통신하는 에이전트
- Docker, rkt 등의 컨테이너 런타임 실행
- 컴퓨팅, 네트워킹 및 스토리지 리소스를 애플리케이션에 제공
- 로깅, 모니터링, 서비스 검색 및 선택적 Add-on을 위한 추가 구성 요소 실행
- 클라우드에서 실행되는 가상 머신(VM) 또는 데이터 센터에서 실행 중인 베어 메탈 서버





# 02 As a Service - CaaS

- Automatic binpacking  
리소스 요청 및 제약 조건에 따라 가용성을 유지하며 컨테이너 자동 배치
- Self-healing  
컨테이너 또는 노드 오류 등으로 health check 실패 시, 컨테이너 자동 복구
- Horizontal scaling  
Pod의 CPU 사용량 또는 애플리케이션 metric에 따라 ReplicaSet의 스케일링 지원
- Service discovery and load balancing  
Service discovery 메커니즘을 위한 애플리케이션 수정 불필요  
컨테이너에 고유 IP 및 단일 DNS 제공으로 로드 벨런싱 가능





## 02 As a Service - CaaS

Automated rollouts and rollbacks

애플리케이션 및 구성 변경 시, 전체 인스턴스의 중단 없이 점진적 롤아웃 가능

Rolling update 문제 발생 시, 배포 리비전에 따라 변경 사항 자동 롤백

Secret and configuration management

이미지 재구성 없이 Secret과 애플리케이션 구성 배포 또는 업데이트 가능

스택 구성을 외부 노출 없이 관리 또는 사용 가능

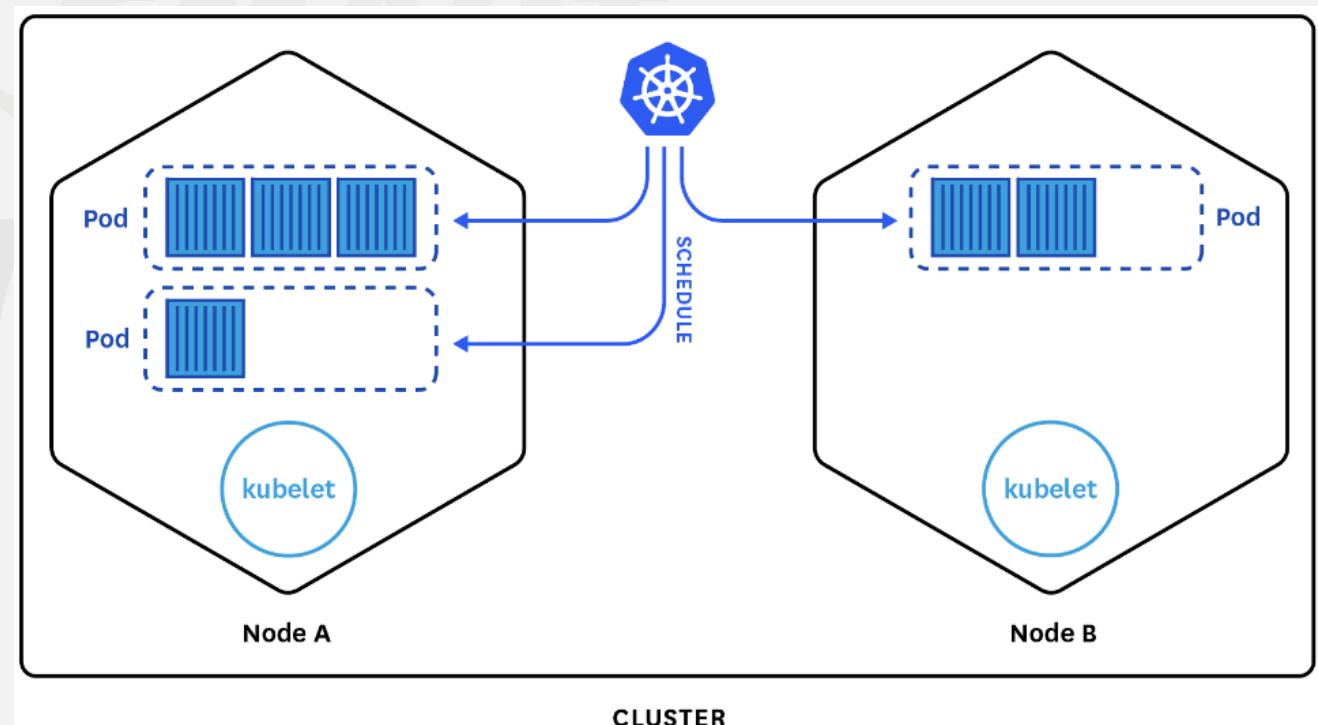
Storage orchestration

로컬 저장소 및 Public Cloud Provider(AWS, GCP), 네트워크 저장소 등에 자동 마운트 지원

Batch execution

배치 및 CI 작업 관리 지원

작업 실패 시, 컨테이너 대체





# 02 As a Service - CaaS

## weakness

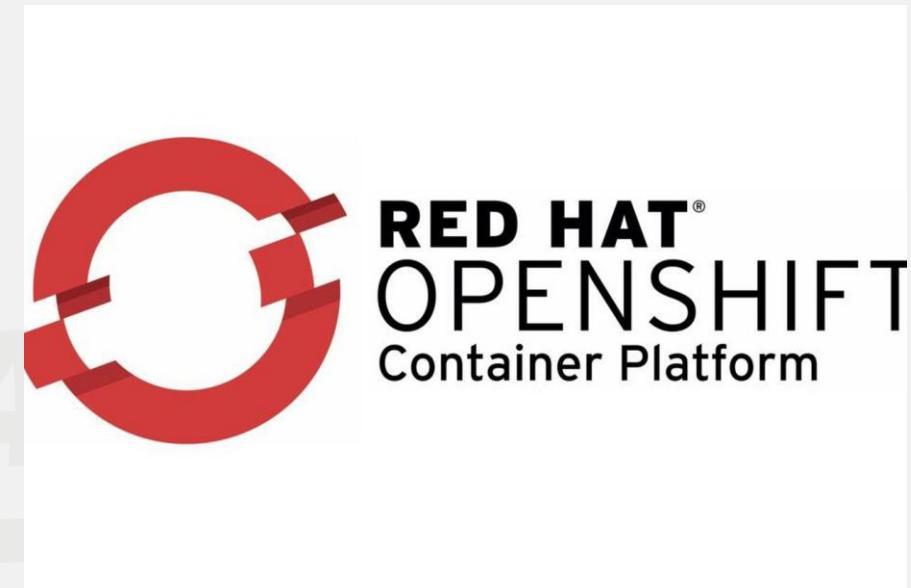
- Master 및 Node(Worker), etcd에 대한 장애 복구 미지원
- Master와 etcd VM 확장 불가능
- 클러스터는 Node(Worker)만 체크
- 구동 중인 클러스터 업그레이드 불가능
- Node(Worker)가 5000개까지 확장 가능, 그 이상 사용하려면 여러 개의 클러스터 필요
- 스테이트풀 서비스 관리 기능 부족
- 기본 인프라 필요
- 빠른 버전업으로 버그 등 안정성 불안
- 국내 GO언어 개발자는 극소수
- GO언어 디버깅 불가능(현재 완전한 Go debugger는 없음)





## 02 As a Service - CaaS

Kubernetes를 활용한 Container Platforms





## 02 As a Service - CaaS

Application Runtime 기반인 파스-타에서  
Container Runtime 을 활용한 시너지 효과 제시



### Code 중심의 개발 플랫폼

- 개발 언어/Framework에 관계없이 원하는 Cloud에 Code를 실행
- Application Lifecycle 관리
- Application 자동 실행
- Open Service Broker API 를 통한 손쉬운 SaaS 서비스와 연동



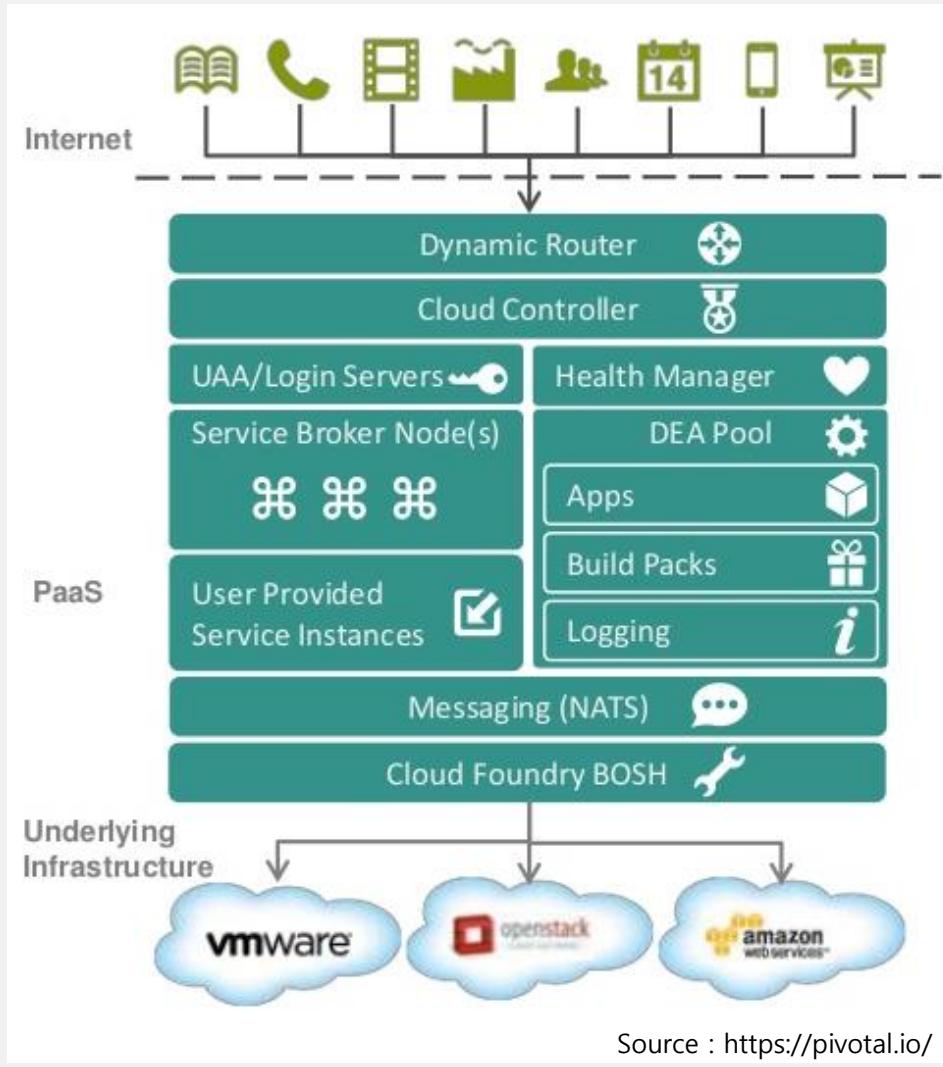
### Container 관리 플랫폼

- Kubernetes 인증된 Container의 쉬운 관리, 유지보수 가능
- Cloud Foundry BOSH를 통한 Lifecycle 관리
- 기 개발된 Application의 Container화 및 Cloud 상에서 운영 및 재 사용이 용이



# 02 As a Service - PaaS

## Platform as a Service

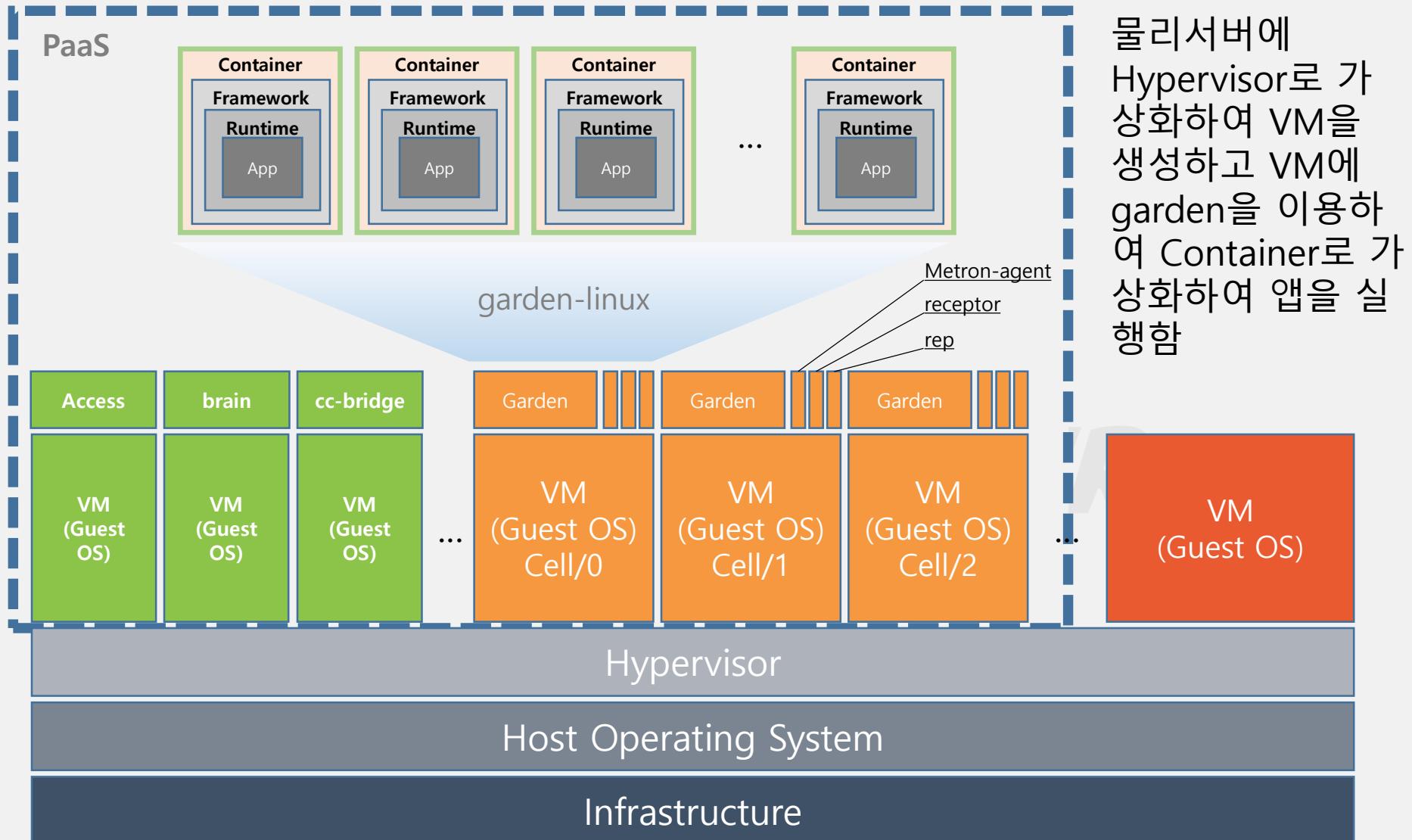


- 애플리케이션 개발, 실행, 관리 할 수 있게 하는 플랫폼을 제공
- SaaS의 개념을 개발 플랫폼에 확장한 방식
- 개발을 위한 플랫폼 구축 필요 없이 웹에서 쉽게 빌려 쓸 수 있는 모델
- 개발자는 개발에만 집중, 애플리케이션이 동작하는 주변 환경은 자동화, marketplace에서 가져다 쓰는 구조
- 개발자는 개발에만 집중, 애플리케이션이 동작하는 주변 환경은 가져다 쓰는 구조
- 개발에서 배포까지 라이프사이클이 짧아 DevOps 문화를 적용하기 용이함



# 02 As a Service - PaaS

IaaS자원 위에 여러 인스턴스들의 유기적인 조합으로 PaaS 동작





# 02 As a Service - PaaS

## Diego VMS

```
Acting as user 'admin' on deployment 'cf-diego' on 'microbosh'
```

```
Director task 2090
```

```
Task 2090 done
```

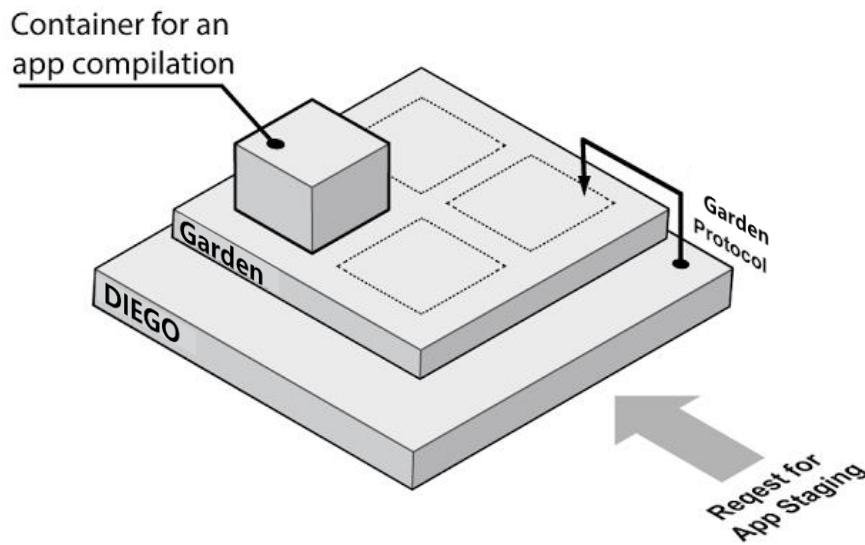
VM	State	AZ	VM Type	IPs
access_z1/0 (e512b3ea-3dc5-45f5-8b1b-6206abe3a3da)	running	n/a	access_z1	10.2.0.67
brain_z1/0 (0262f24d-1151-4b4e-996f-2d6a6cbf405d)	running	n/a	brain_z1	10.2.0.55
cc_bridge_z1/0 (30ac5913-ca79-476f-b92d-bb034a51d3dd)	running	n/a	cc_bridge_z1	10.2.0.61
cell_z1/0 (ce3158f0-6e7a-4510-8a58-e2ca044ee47b)	running	n/a	cell_z1	10.2.0.58
cell_z1/1 (63eee668-e45b-4a0e-93b3-664c5eecbf12)	running	n/a	cell_z1	10.2.0.59
cell_z1/2 (06fbaf28-1f5d-47ea-94b4-7d394d29c420)	running	n/a	cell_z1	10.2.0.60
database_z1/0 (1b243b98-a796-4b5e-8a11-2a6f8e694234)	running	n/a	database_z1	10.2.0.52
route_emitter_z1/0 (66e59fe6-8ed1-454a-92c5-88194114a3bf)	running	n/a	route_emitter_z1	10.2.0.64



## 02 As a Service - PaaS

Application이 실제 배포시 PaaS의 구성요소중 Diego cell에서의 모습

### Application Compilation



배포 명령이 실행되면 Cloud Controller에 의해 Diego는 컨테이너를 생성하고 그 안에 Application 코드를 배치한다.

이때 Buildpack은 코드를 분석하여 적합한 Buildpack을 찾는다.

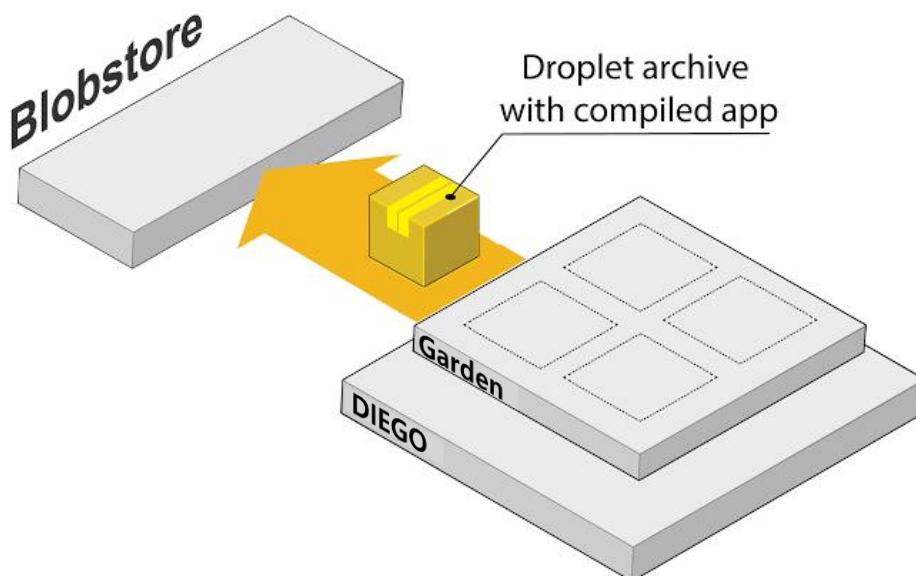
그 후 컴파일을 진행한다.



## 02 As a Service - PaaS

Application이 실제 배포시 PaaS의 구성요소중 Diego cell에서의 모습

Staging is Complete



컴파일이 완료되고 Application이 동작할 수 있는 부가 요소를 모두 갖춘 형태를 Droplet Archive라고 한다.

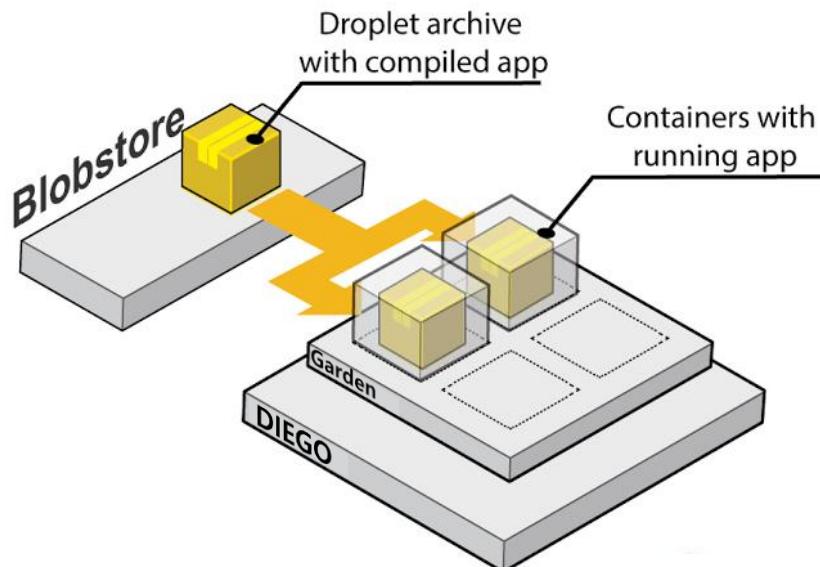
이를 Blobstore라는 공간에 저장한다.



## 02 As a Service - PaaS

Application이 실제 배포시 PaaS의 구성요소중 Diego cell에서의 모습

### Running an Application



Blobstore에 업로드가 완료되면 Diego는 garden에 Application Instance가 실행될 새로운 컨테이너생성을 요청한다.

그 후에 release스크립트가 실행되어 Application이 동작한다.



## 02 As a Service - PaaS

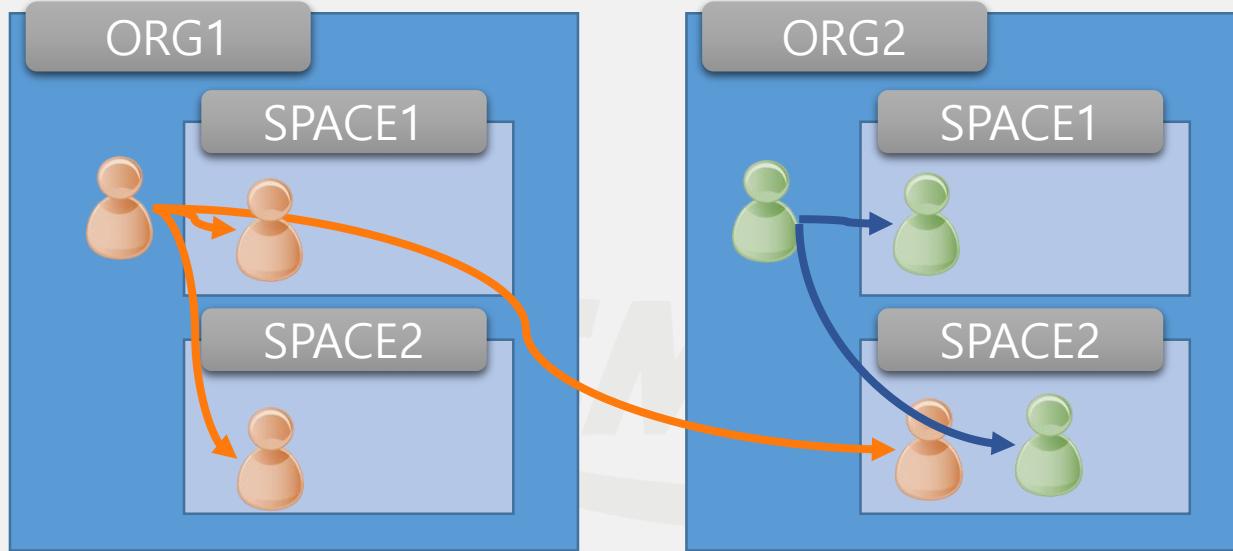
BuildPack : 언어별 프로그램이 동작할 수 있도록 언어 프레임워크등의 기술 지원

Name	Supported Languages, Frameworks, and Technologies
Binary	n/a
Go	Go
Java	Grails, Play, Spring, or any other JVM-based language or framework
.NET Core	.NET Core
Node.js	Node or JavaScript
PHP	Cake, Symfony, Zend, Nginx, or HTTPD
Python	Django or Flask
Ruby	Ruby, JRuby, Rack, Rails, or Sinatra
Staticfile	HTML, CSS, JavaScript, or Nginx
Jeus	Jeus was for tibero DB



## 02 As a Service - PaaS

Org & Space : 효과적으로 업무를 구분할 수 있는 논리적인 그룹기능



ORG : 개인 또는 여러 공동 작업자가 소유하고 사용할 수 있는 개발 계정  
각 계정은 ORG에 룰을 부여받을 수 있음  
**Org Manager, Org Auditor, Org Billing Manager, Org User**

SPACE : 모든 애플리케이션 및 서비스의 범위가 공간으로 지정  
**Space Manager, Space Developer, Space Auditor**

Space Developer만이 애플리케이션 실행 및 서비스 연동 할 수 있음



# 02 As a Service - PaaS

## Application & Services

- Application : 사용자가 제작한 결과물
- Service : Application동작에 필요한 부가 서비스 (MySQL, Redis, RabbitMQ 등)

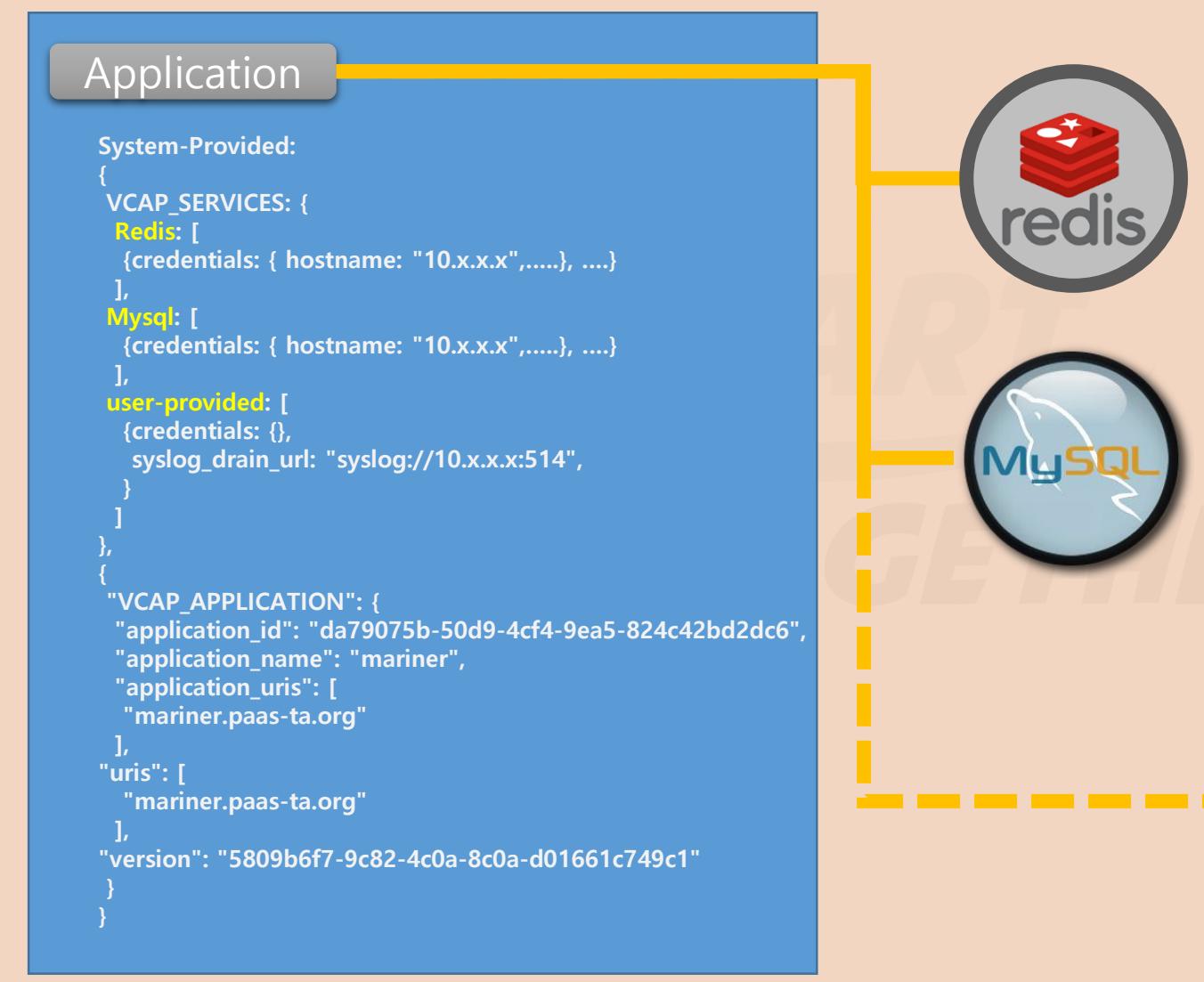
MarketPlace에서 준비되어있는 서비스를 골라서 사용  
(외부 서비스도 사용가능)





# 02 As a Service - PaaS

Application & Services : 연결된 서비스는 해당 Application 환경변수로 확인 가능





# 02 As a Service - PaaS

## Service Packs : 제공가능한 서비스들

no	Service	plan	description
1	glusterfs	glusterfs-5Mb, glusterfs-100Mb, glusterfs-1000Mb	데이터 저장소
2	rabbitMQ	standard	메시지 큐
3	ARCUS	arcus-100mb-plan	NoSQL 데이터베이스
4	redis	shared-vm, dedicated-vm	NoSQL 데이터베이스
5	Mongo-DB	default-plan	NoSQL 데이터베이스
6	AltibaseDB	altibase-plan-dedicated-vm	관계형 데이터베이스
7	CubridDB	utf8, euckr	관계형 데이터베이스
8	Mysql-DB	Mysql-Plan1-10con, Mysql-Plan2-100con	관계형 데이터베이스
9	Tibero	Basic	관계형 데이터베이스
10	pinpoint	Pinpoint_standard	애플리케이션 모니터링(APM)



# 02 As a Service - PaaS

## PaaS 의 종류



- \* Public : 내부, 외부망을 사용 할 수 있고, 과금이 가능함
- \* Private : On-premise 형태로 자사 클라우드로 구축하여 사용하는 형태(내부망)



# 02 As a Service - PaaS

PaaS를 이루는 VM들을 설치 및 관리하는 도구



Cloud Foundry Foundation active projects

**cf-deployment**  
The canonical open source deployment manifest for Cloud Foundry

**relint-ci-pools**

**garden-runc-release**

Stemcells

BOSH produces official [stemcells](#) for popular open source operating systems and cloud infrastructures that support it, [light stemcells](#) are also available for environments that we have pre-uploaded and shared within the community.

Version	Download
3586.25	Full Stemcell
3586.25	Full Stemcell
3586.25	Full Stemcell
3586.25	Light Stemcell

Releases

A release is a versioned collection of configuration properties, configuration templates, start up scripts, source code, binary artifacts, and anything else required to build and deploy software in a reproducible way ([learn more](#)).

### Community Releases

- [DataDog/datadog-agent-boshrelease](#)
- [DataDog/datadog-firehose-nozzle-release](#)
- [Dynatrace/bosh-oneagent-release](#)
- [SAP/bosh-kubernetes-cpi-release](#)
- [SAP/ipsec-release](#)
- [bosh-packages\(cf-cli-release\)](#)
- [cf-platform-eng/couchdb-boshrelease](#)
- [cf-platform-eng/docker-boshrelease](#)
- [cf-platform-eng/hadoop-boshrelease](#)
- [cf-platform-eng/mesos-boshrelease](#)



## 02 As a Service - PaaS

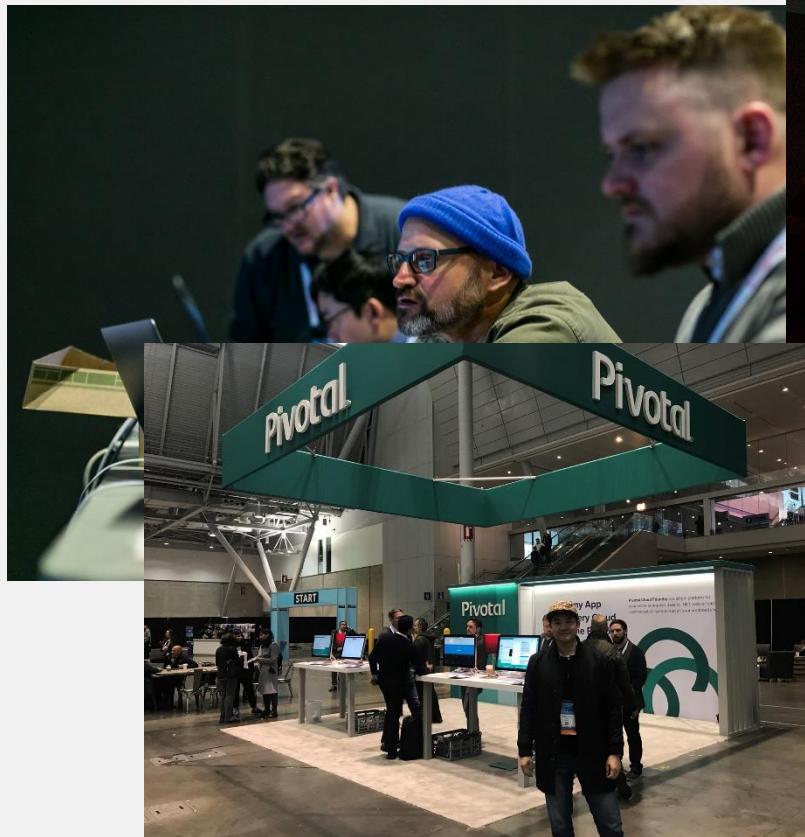
기반이 되는 opensource community의 활발한 활동





## 02 As a Service - PaaS

최신기술과 접목시킬수 있는 자료가 풍부



# 03 PaaS-TA

정부 주도 Platform as a Service

**SMART**  
**2018 TOGETHER**



# 03 PaaS-TA

Cloud foundry 오픈소스로 만들어진 PaaS  
NIA에서 만들었으며, KT, Koscom 등에서 사용, 3.0까지 발표됨.





# 03 PaaS-TA - releases



PaaS-TA  
OpenPaaS

- 개발 지원서비스 포함  
(cubrid, mysql, mongoDB, redis, rabbitMQ, swift, api플랫폼)
- 전자정부 프레임워크 2.5지원
- 플랫폼 설치 자동화 지원



LINGUIINE  
Open PaaS PaaS-TA 2.0

- 운영 및 관리도구 추가
- 사용자/운영자 포털, 모니터링, 미터링, 설치 자동화 고도화
- 전자정부 프레임워크 3.5지원
- WEB-IDE추가(제한적)



PENNE  
Open PaaS PaaS-TA 3.0

- 운영 및 관리도구, 서비스 모니터링 고도화
- 형상관리 서비스 도구 추가
- 배포 파이프 라인 서비스 도구 추가
- IaaS 관리 대시보드 추가



Rotelle  
Open PaaS PaaS-TA 4.0

- 운영 및 관리도구 고도화
- Azure IaaS 수용
- CaaS(Container As A Service)



# 03 PaaS-TA – 단계별 개발 진행

단계	핵심과제	내용	진행년도
1단계 개발	개발/운영/관리환경 고도화	<ul style="list-style-type: none"> <li>· IaaS, PaaS, SaaS 통합 모니터링 기능 개발</li> <li>· 모니터링 기반 오토스케일링 기능 개발</li> <li>· 포탈 통합 계정 인증/권한 관리 기능 개발</li> <li>· 응용의 개발/테스트/배포 개발도구 개발</li> <li>· 신규 클라우드 IaaS 지원 검증 및 설치 자동화 개발</li> <li>· 이종 클라우드 기반의 개방형 PaaS 플랫폼 지원 검증</li> </ul>	2017년
2단계 개발	이종 클라우드 지원 및 관리기술 개발	<ul style="list-style-type: none"> <li>· IaaS, PaaS, SaaS 통합 모니터링 기능 확장 개발</li> <li>· Application Gateway 기능 개발</li> <li>· 신규 클라우드 IaaS 지원 검증 및 설치 자동화 확장 개발</li> <li>· 이종 클라우드 기반의 개방형 PaaS 플랫폼 지원 개발</li> <li>· 플랫폼 운영자 및 개발자 포탈 기능 확장 개발</li> </ul>	2018년
3단계 개발	응용 마켓플레이스 구현 및 개발자 지원환경 개발	<ul style="list-style-type: none"> <li>· IaaS, PaaS, SaaS 통합 모니터링 기능 확장 개발</li> <li>· 응용의 전주기 라이프 사이클 관리 기능 개발</li> <li>· PaaS상에 동작하는 응용을 위한 마켓플레이스 기능 개발</li> <li>· 응용의 사용량 측정 기능 고도화</li> <li>· 신규 클라우드 IaaS 지원 검증 및 설치 자동화 확장 개발</li> <li>· On-Demand 서비스 설치 개발</li> <li>· 이종 클라우드 기반의 개방형 PaaS 플랫폼 지원 기능 확장 개발</li> </ul>	2019년

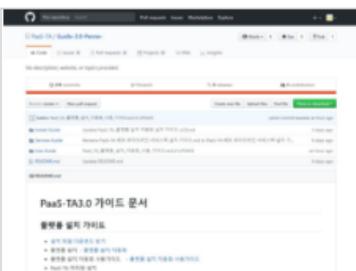


# 03 PaaS-TA - 활동



## 공동연구 협력기업 워크샵

- 1차년도 개발 추진 내용 발표 및 피드백
- 각 협력기업 별 공동연구 진행현황 공유



## 소스 코드 공개

- CF, Apache 재단과 같이 모든 소스코드를 오픈소스로 공개 및 관리화 예정
- CF코어를 제외한 R&D 자체 개발 파트만 공개



## PUB 공개 세미나

- 파스-타 고도화 사업 1차년도 개발내용 발표
- 주요 기능 시연



## 정식 버전 3.0 릴리즈

- Github 및 파스-타 포탈에 소스코드 및 패키징 파일 등록
- 정식 버전 3.0 공개('17년 12월 14일)



# 03 PaaS-TA – 파스-타 포탈

포탈을 통한 한글 가이드 및 교육, 기술 지원

← → C | 안전함 | <https://paas-ta.kr/notice/qna>

로그인 | 회원가입 | 사이트맵 | 참조링크

PaaS-TA | 파스-타 인사이드 | 알림마당 | 다운로드 | 체험하기 | 협력지원

공지사항 | 자주 묻는 질문 | 묻고 답하기

## 묻고 답하기

PaaS-TA 에 궁금하신 부분을 질문해 주시면 답변해 드립니다.

전체 검색어를 입력해주세요. 검색 등록하기

전체 | 개발환경 | 체험신청 | 기술지원 | 다운로드 | 회원가입

총 167건

번호	구분	제목	작성자	진행상태	등록일	조회수
167	개발환경	eclipse에서 open paas 계정 연결	이광열	완료	09:31:01	6
166	개발환경	openpaas_dev_env.jar 파일 받을 수 있는곳.	이광열	완료	2018-07-10	20

# 04 Application on Cloud

Cloud Native Architecture

**SMART**  
**2018 TOGETHER**



# 04 Application on Cloud – 개발 방법론

## Cloud Native Model

Cloud Native

- Microservices 구조
- API 기반 설계

Cloud Resilient

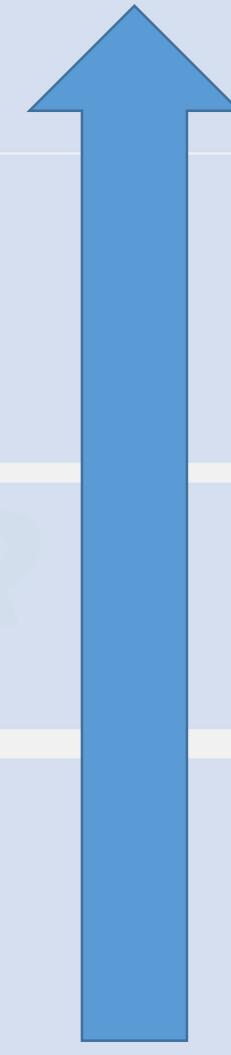
- Fault-tolerant 와 resilient design
- Cloud-agnostic runtime implementation
- Metrics과 monitoring 포함
- Proactive failure testing

Cloud Friendly

- 12 Factor App 준수
- 수평적 확장(Horizontally scalable)
- 고 가용성을 위해서 플랫폼 활용

Cloud Ready

- No permanent disk access
- Self-contained application
- 플랫폼에서 네트워킹 및 포트 관리
- 플랫폼 서비스 사용





# 04 Application on Cloud – 개발 방법론

## Cloud Native key point

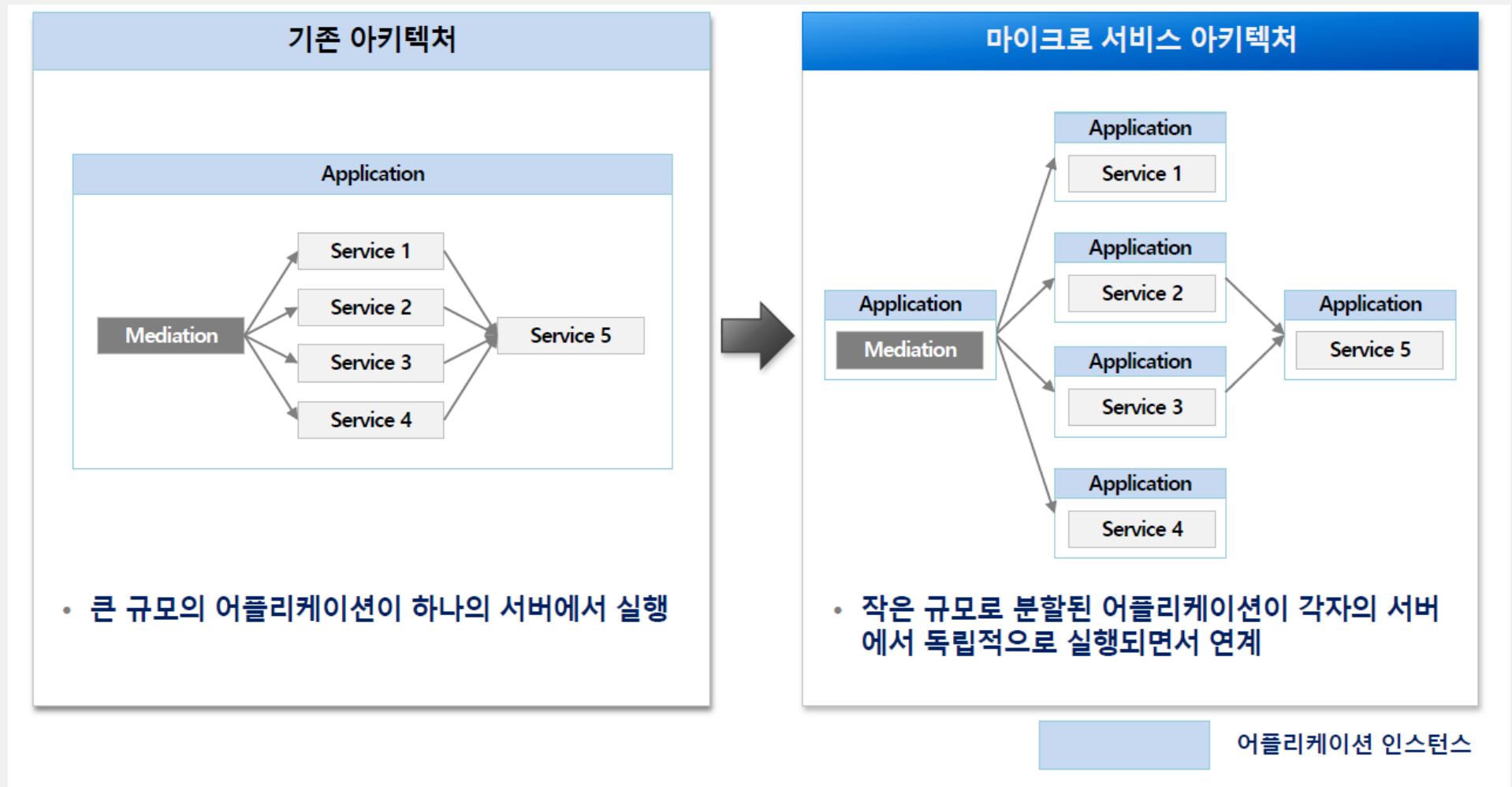
1. 필요에 따라 유연하게 프로비저닝 할 수 있도록 IaaS를 사용해라.  
→ PaaS는 IaaS 위에서 동작.
2. 배포를 위한 수작업을 스크립트 혹은 코드로 대체하라.  
→ PaaS는 CLI로 스크립트를 실행하여 배포
3. 컨테이너화 하여 이동, 배포가 쉽게 하라  
→ PaaS는 컨테이너에 Application을 구동
4. 마이크로 서비스 아키텍쳐를 사용해라.  
→ 개발자의 몫

“PaaS는 *Cloud Native*한 개발 환경을 제공  
개발자는 *microservice*를 고려해서 개발”



# 04 Application on Cloud

## Microservice architecture

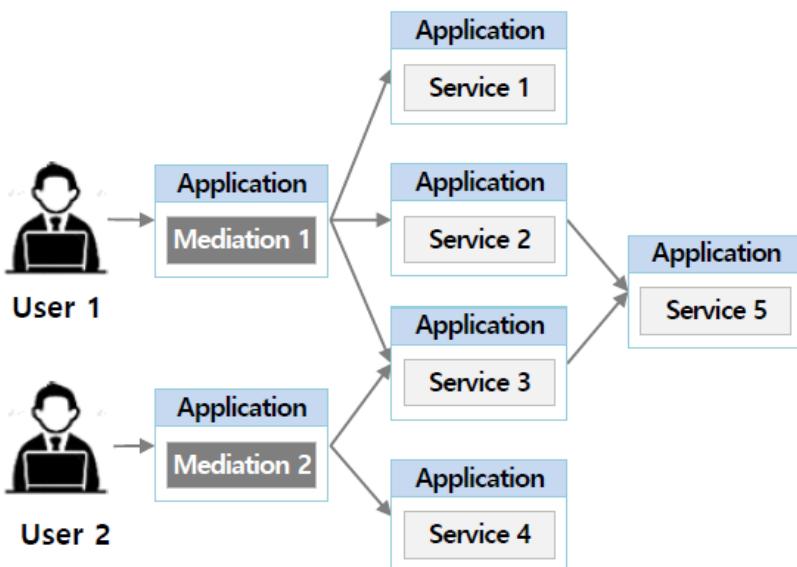




# 04 Application on Cloud

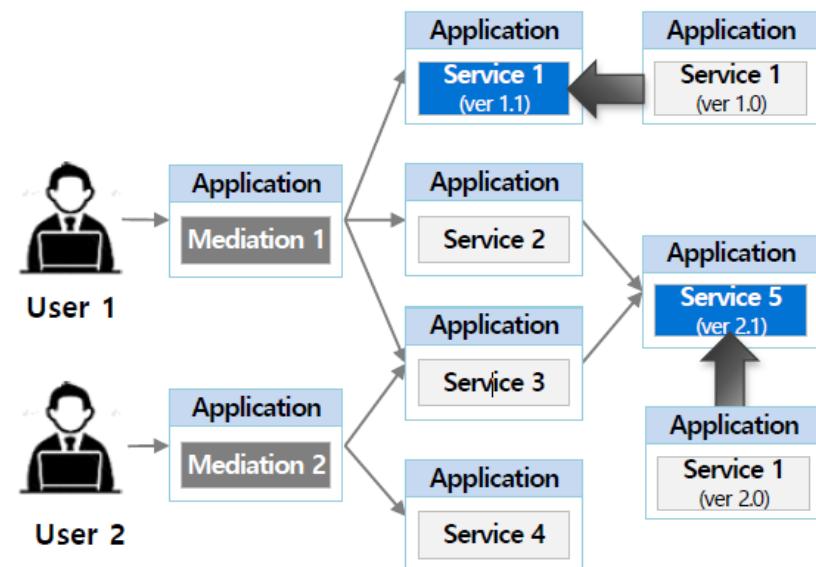
## Microservice architecture

### ① 고객 맞춤형 서비스 제공



- 고객 요구사항에 맞는 최적의 서비스 조합
- 서비스의 재사용성 향상

### ② 신속한 서비스 개선



- 작은 서비스 단위의 업그레이드가 가능하기 때문에 고객의 요구사항을 민첩하게 반영할 수 있음

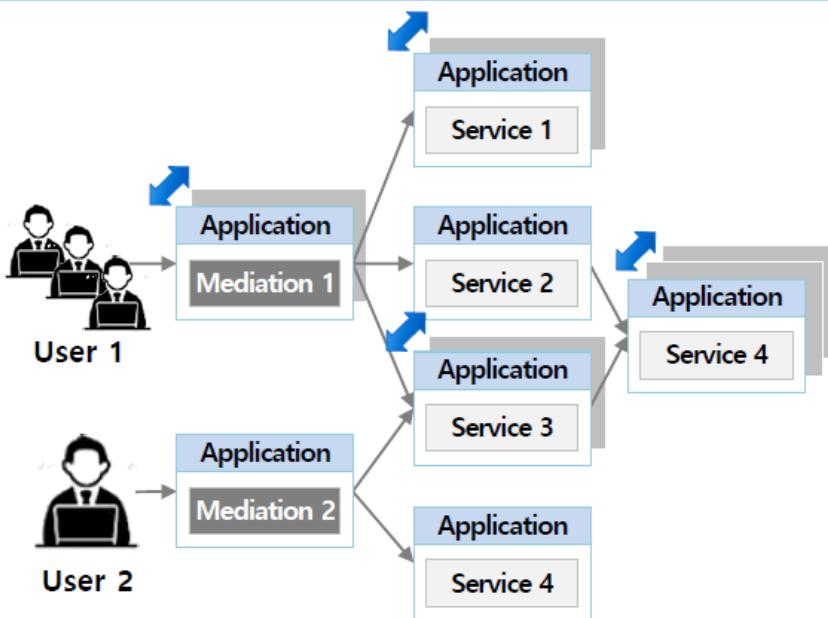
어플리케이션 인스턴스



# 04 Application on Cloud

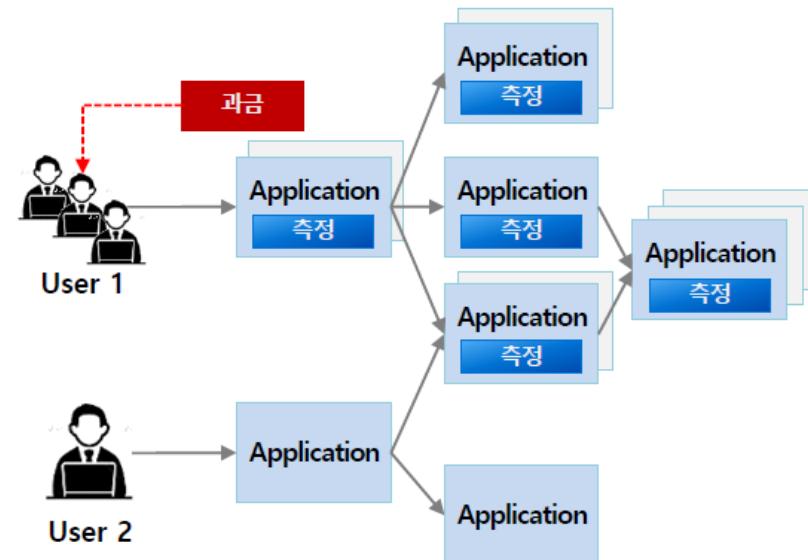
## Microservice architecture

### ③ 유연한 서비스 인스턴스 확대·축소



- 작은 서비스 단위의 유연한 용량 확대·축소  
▶ IT자원의 효율적인 활용

### ④ 정확한 측정·과금



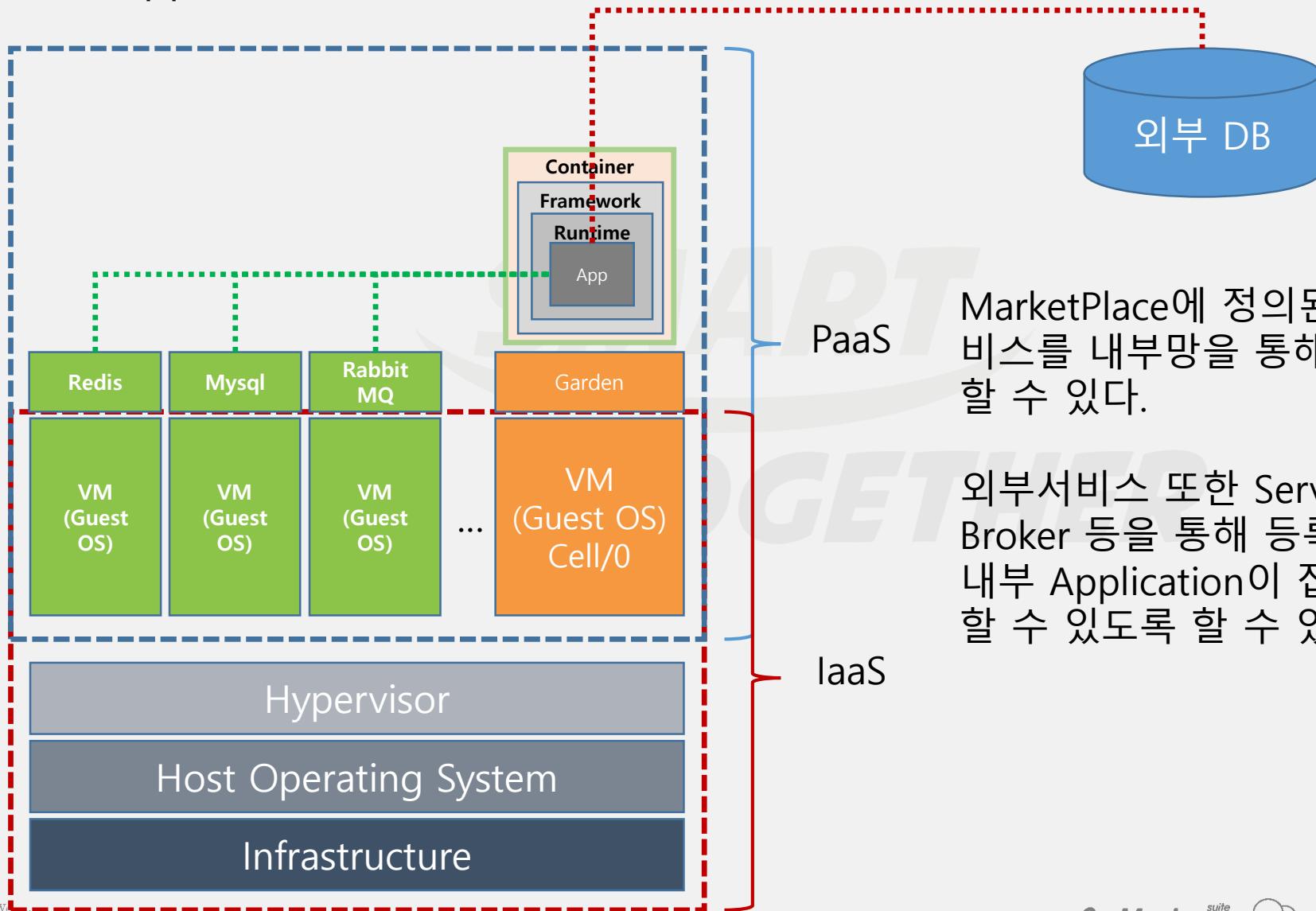
- 사용한 만큼 정확한 측정·과금  
▶ 투명하고 합리적인 가격 정책 수립

어플리케이션 인스턴스



# 04 Application on Cloud

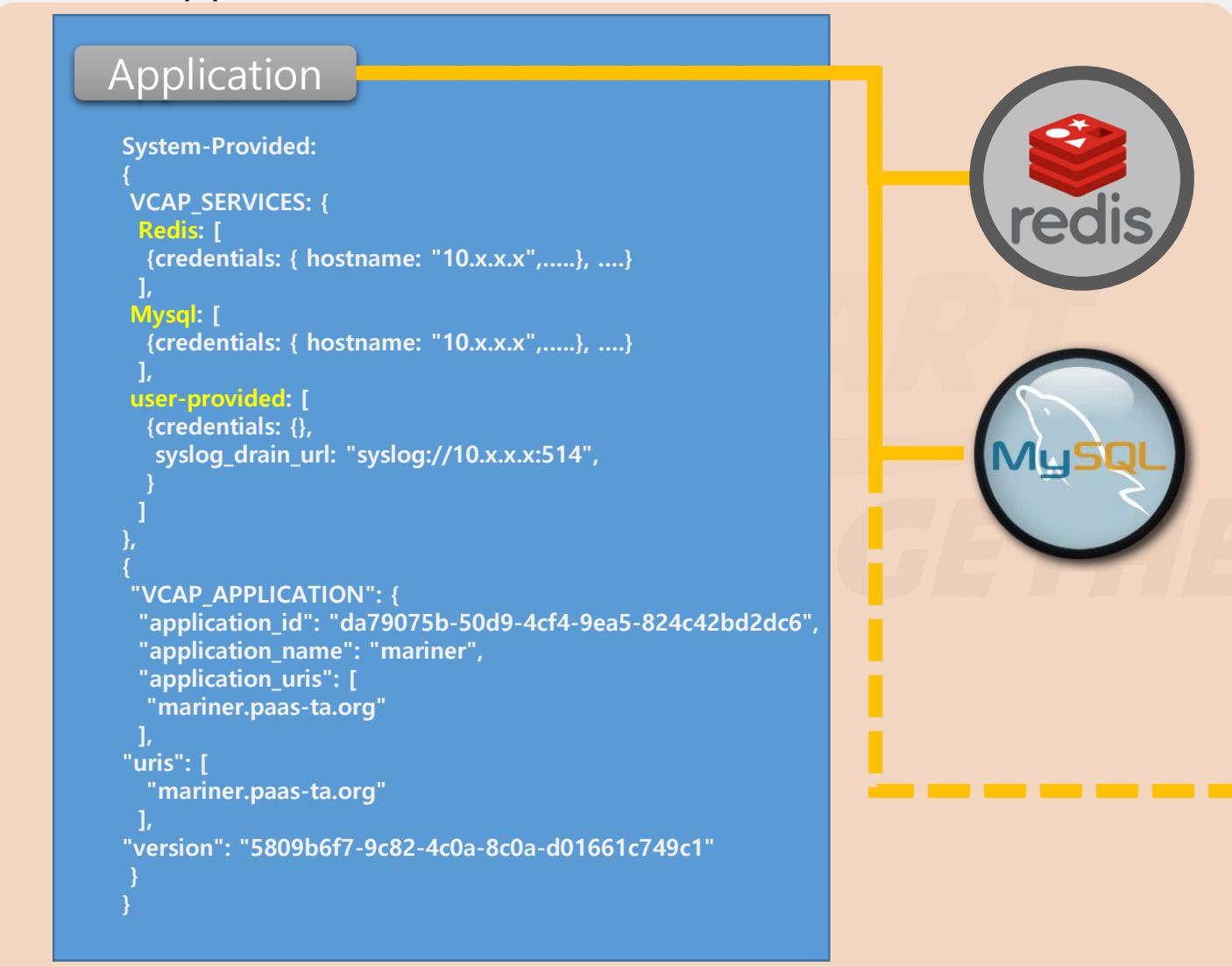
## Cloud Application & Service





# 04 Application on Cloud

## Cloud Application & Service





# 04 Application on Cloud

## cloud properties

Cloud환경에서 적용될 Propertie로 Application이 사용 할 서비스명을 설정.

Redis와 같이 이미 널리 쓰는 서비스에 대한 라이브러리 정보를 제공하고 있으나  
커스터마이징 서비스 등은 제공하지 않음.

Application의 VCAP SERVICE 환경정보를 활용하여 서비스 사용.

resource/**application-cloud.properties**

**redis.service.name:** test\_redis #서비스 생성시 이름



# 04 Application on Cloud

## Sample source code

```
@Profile("cloud") // Cloud환경 인식하는 Annotation(앱 배포시 환경 변수 SPRING_PROFILES_ACTIVE: cloud 입력)
@Configuration // Spring Config
@ServiceScan // Cloud Config Service Scan Annotation
public class CloudDataConfig extends AbstractCloudConfig {

    private String marinerIp;
    private String marinerPort;

    private String semonIP;
    private int semonPort;

    @Value("${redis.service.name}")
    private String redisServiceName; //application-cloud.properties의 redis 서비스 인스턴스명 읽어온다

    @Bean(name = "connOtherDB")
    public DataSource marinerDataSource() {
        try {
            String vcap_services = System.getenv("VCAP_SERVICES");
            JSONObject jsonObj = JSONObject.fromObject(vcap_services);
            JSONArray userPro = jsonObj.getJSONArray("OtherDB");
            jsonObj = JSONObject.fromObject(userPro.get(0));
            jsonObj = jsonObj.getJSONObject("credentials");
            marinerIp = (String)jsonCredential.get("hostname");
            marinerPort = (String)jsonCredential.get(" port ");
            CommandSearchRequest command = new CommandSearchRequest(marinerIp, Integer.parseInt(marinerPort));
            CommandSearchRequest.setProps(marinerIp, Integer.parseInt(marinerPort), 30000, 50, 100);
            return CommandSearchRequest;
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}
```



# 04 Application on Cloud

## Sample source code

Application에서 사용하는 Maven Dependency

```
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.2.3.RELEASE</version>
    <relativePath />
</parent>

..
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-config</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-configuration-processor</artifactId>
    <optional>true</optional>
</dependency>
```



# 04 Application on Cloud

## Sample source code

```
<!-- Spring Cloud Connector Start -->
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-spring-service-connector</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-core</artifactId>
</dependency>

<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-cloudfoundry-connector</artifactId>
</dependency>
<!-- Spring Cloud Connector End -->

<!-- Redis Start -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-redis</artifactId>
</dependency>

<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-pool2</artifactId>
</dependency>
<!-- Redis End -->
```

# 05 Practice

배포, 서비스 연동

**SMART**  
**2018 TOGETHER**



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 - 대시보드

The screenshot shows the PaaS-TA web portal dashboard. At the top, there is a navigation bar with links for '대시보드' (Dashboard), '카탈로그' (Catalog), '도움말' (Help), '문서' (Documentation), '공지' (Announcements), and '커뮤니티' (Community). A banner message says '▶ 알림사항 파스타 포털에 오신걸 환영합니다.' (Welcome to the Pasta Portal!). On the left, a sidebar shows user information: 'null' and 'pilot1213▼'. It also lists '조직생성' (Organization Creation) under 'nia' and '공간생성' (Space Creation) under 'dev'. The main content area has four sections: '조직' (Organization) showing 'nia' with 1GB used of 20GB; '공간(1)' (Space(1)) showing one 'dev' space with 2 apps and 9 services; '도메인(1)' (Domain(1)) showing 'paas-ta.org' with 'SHARED'; and '사용자(2)' (User(2)) with a '사용자 초대하기' (Invite User) button.

PaaS-TA  
OpenPaaS

대시보드 카탈로그 도움말 문서 공지 커뮤니티

▶ 알림사항 파스타 포털에 오신걸 환영합니다.

조직  
nia 이름변경 삭제 WEB IDE 신청 취소  
1GB 사용중 / 20GB

공간(1) 공간생성  
도메인(1) 도메인 추가  
paas-ta.org SHARED

사용자(2) 사용자 초대하기

null  
pilot1213▼

조직생성  
nia  
▶ nia

공간생성  
dev



# 05 Practice – web portal

Portal을 통한 Application Instance 관리 – 공간 개요

The screenshot shows the PaaS-TA web portal interface. At the top, there is a navigation bar with tabs: 대시보드 (Dashboard), 카탈로그 (Catalog), 도움말 (Help), 문서 (Documents), 공지 (Announcements), and 커뮤니티 (Community). On the left, a sidebar shows user information (null, pilot1213) and organization/service definitions (조직생성: nia, 공간생성: dev).

The main area displays two sections: '애플리케이션(2)' (Applications) and '서비스(9)' (Services).

**애플리케이션(2)**

- welcome-cf (Cloud Foundry): welcome-cf12.paas-ta.org, 실행 중 (Running), 인스턴스: 1, 메모리: 512 MB. Actions: 정지 (Stop), 설정 (Configure).
- mariner (Cloud Foundry): mariner.paas-ta.org, 실행 중 (Running), 인스턴스: 1, 메모리: 512 MB. Actions: 정지 (Stop), 설정 (Configure).

**서비스(9)**

- mariner (Mariner4, non-proxy): 연결 앱 수: 1. Action: 설정 (Configure).
- semon (Semon, non-proxy): 연결 앱 수: 0. Action: 설정 (Configure).
- syslog (user-provided): 연결 앱 수: 1. Action: 설정 (Configure).
- mysql (Mysql-DB, Mysql-Plan1-10con): 연결 앱 수: 1. Action: 설정 (Configure).



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 – Application 대쉬보드

The screenshot shows the PaaS-TA web portal interface for managing application instances. The top navigation bar includes links for Dashboard, Catalog, Help, Document, Knowledge, and Community. A sidebar on the left lists organization and space names: null, pilot1213, nia, and dev.

The main content area displays the details of an application instance named "welcome-cf". It shows the app's name ("welcome-cf"), URL ("welcome-cf12.paas-ta.org"), creation date ("2017-10-26 06:33:52"), buildpack ("java\_buildpack.."), and current state ("실행 중"). The instance configuration section allows changing the number of instances (1), memory (512 MB), and disk (1024 MB). Resource usage metrics are shown as bars: CPU (0.1%), Memory (44.7%), and Disk (12.8%).

The bottom section, titled "이벤트" (Events), lists recent events. The table headers are: 이벤트, 상태, 서비스(3), 환경변수, 라우트(1), and 로그. The events listed are:

이벤트	상태	서비스(3)	환경변수	라우트(1)	로그
2017-11-17 04:57:48	audit.app.ssh-authorized	pilot1213			
2017-11-06 11:52:12	audit.app.restage	pilot1213			app:RESTAGE
2017-11-06 11:39:52	app crash	welcome_cf			app:CRASHED



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 - 카탈로그

PadS-TA OpenStack

대시보드 카탈로그 도움말 문서 공지 커뮤니티 MENU

▶ 알림사항 파스타 포털에 오신걸 환영합니다.

검색어를 입력하세요.

최근작업

paasta-user paasta.user@gmail.com\*

전체보기 앱 템플릿 기본유형 앱 개발환경 시스템 앱 개발환경 사용자 앱 개발환경

Redis 서비스 자바 기본유형 앱 템플릿 Java 애플리케이션 APM 서비스

애플리케이션 프레임워크 및 개발환경을 제공합니다.

시스템 앱 개발환경 파스타에서 기본적으로 제공하는 앱 개발 환경으로 지금 앱을 시작합니다.

서비스 데이터 저장소 메시징 NoSQL 데이터베이스 관계형 데이터베이스 아플리케이션 모니터링

PHP 앱 개발환경 Python 앱 개발환경 GO 앱 개발환경 Node.js 앱 개발환경 Java8 온라인 앱 개발환경 전자정부 프레임워크 앱 개발환경

Java8

RAILS

Infranics We Believe INFRANICS



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 – 앱개발환경 생성

The screenshot shows the PaaS-TA web portal interface. On the left, there's a sidebar with navigation links like '전체보기', '앱 템플릿', '기본유형', '앱 개발환경' (selected), and '서비스'. The main area displays a 'Java8 온라인 앱 개발환경' template with a Java8 logo. To the right, a '앱 생성' (App Create) dialog is open, containing fields for '조직' (Organization: OCP), '공간' (Space: dev), '앱 이름' (App Name: app-test), '앱 URL' (App URL: app-test.115.68.46.29.xip.io), '메모리(MB)' (Memory: 512), '디스크(MB)' (Disk: 1024), and '도메인' (Domain: 115.68.46.29.xip.io). A red arrow points to the 'ON' button under '앱 시작여부' (App Start). At the bottom right of the dialog are '생성' (Create) and '취소' (Cancel) buttons.



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 – 서비스

The screenshot shows the PaaS-TA web portal interface. At the top, there's a yellow header bar with the PaaS-TA logo, a search bar, and navigation links: 대시보드 (Dashboard), 카탈로그 (Catalog), 도움말 (Help), 문서 (Documents), 공지 (Announcements), 커뮤니티 (Community), and MENU.

In the top right corner, a message says "알림사항 파스타 포털에 오신걸 환영합니다." (Welcome to the PaaS-TA portal!).

The left sidebar has a user profile icon and the text "paasta-user paasta.user@gmail.com". It includes links for 전체보기 (Overview), 앱 템플릿 (App Templates), 기본유형 (Basic Types), 앱 개발환경 (App Development Environment), 시스템 앱 개발환경 (System App Development Environment), and 사용자 앱 개발환경 (User App Development Environment). A red box highlights the "서비스" (Services) section.

The main content area has a search bar and a "최근작업" (Recent Work) section featuring icons for Redis, Java 기본유형 앱 템플릿, and PINPOINT.

Below that is a "서비스" (Services) section with a sub-section for "데이터 저장소" (Data Storage). It shows an icon for "OBJECT STORAGE" and "Swift 오브젝트 스토리지 서비스". Another red box highlights this "OBJECT STORAGE" section.

At the bottom, there's a copyright notice: "Copyright © 2016 PaaS-TA. All rights reserved."



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 – 서비스 생성

The screenshot shows the PaaS-TA web portal interface for creating a MySQL DBMS service. The top navigation bar includes links for Dashboard, Catalog, Help, Document, Public, Community, and MENU. The left sidebar shows the user profile (관리자 admin) and navigation links for Overall View, App Template, Basic Type, App Development Environment, System App Development Environment, User App Development Environment, Services, Data Store, Message, NoSQL Database, Relational Database, and Application Monitoring.

The main content area displays the MySQL DBMS service details. It features a MySQL logo and a brief description: "MySQL은 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템(RDBMS)이고 다중 스레드, 다중 사용자 형식의 구조질의여 형식의 데이터베이스 관리 시스템이다."

The right panel is titled "서비스 생성" (Service Creation) and contains the following fields:

- 조직: dev (highlighted with a red arrow)
- 서비스 이름: app-test (highlighted with a red arrow)
- 서비스 파라미터: {"test": "mysql"} (highlighted with a red arrow)
- 앱: 연결 없이 시작 (highlighted with a red arrow)
- 서비스 이용사양 선택: Mysql-Plan1-10con (highlighted with a red arrow)

At the bottom right of the creation panel are "생성" (Create) and "취소" (Cancel) buttons.

Copyright © 2016 PaaS-TA. All rights reserved.



# 05 Practice – web portal

## Portal을 통한 Application Instance 관리 – 서비스 관리

The screenshot shows the PaaS-TA web portal interface. At the top, there is a header bar with the PaaS-TA logo, navigation links for Dashboard, Catalog, Help, Document, Public, Community, and a search bar. On the left, there is a sidebar with sections for User Profile (null, pilot1213), Organization Creation (nia), and Space Creation (dev). The main content area displays an application instance summary: App URL (welcome-cf12.paas-ta.org), Last Pushed (2017-10-26 06:33:52), Development Environment (java\_buildpack..), and Status (Running, with a red 'Stop' button and an orange 'Restart' button). To the right of this is a resource monitoring section showing CPU (0.2%), Memory (44.7%), and Disk (12.8%) usage. Below these are tabs for Events, Status, Services (3 selected), Bindings, Routes, and Logs. The Services tab shows a list of services: mariner, mysql, semon, syslog, cf\_log\_drain, test\_log, abc, and log\_test3. A dropdown menu for '서비스명' is open over 'mariner'. To the right of the service list is a table for connecting services. The first row connects 'mariner' to 'Mysql-DB' with a ' 연결' (Connect) button highlighted with a red box. The second and third rows connect 'mariner' to 'user-provided' services with '연결해제' (Disconnect) buttons highlighted with red boxes. A green '+ 서비스 연결' (Add Service Connection) button is also present.



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 - CLI설치

- <https://cloudfoundry.org>

The screenshot shows a web browser window with the URL <https://cloudfoundry.org/docs-download/> in the address bar. The page itself is the 'Docs' section of the Cloud Foundry website. At the top of the page, there are several navigation links: 'Choose Certified Platform', 'Become a Corporate Member', and 'Blog'. Below these are the Cloud Foundry logo and a main navigation bar with links for 'Platform', 'Get Started', 'The Foundry', 'Community', 'Case Studies', 'Events', 'Newsroom', 'About Us', and a search icon. A secondary navigation bar below it includes 'Start Here', 'Certified Platforms', 'Docs & Download' (which is highlighted with an orange box), 'Developer Training', 'Developer Certification', and 'Free Community Courses'. The main content area features a large 'Docs' heading and a sub-headline: 'Our community maintains docs related to Application Runtime and BOSH – the two [platforms](#) that comprise Cloud Foundry.' Below this are two buttons: 'APPLICATION RUNTIME DOCS →' (highlighted with an orange box) and 'BOSH DOCS →'. Further down the page, there are sections for 'Download' and 'Installation'.

Docs

Our community maintains docs related to Application Runtime and BOSH – the two [platforms](#) that comprise Cloud Foundry.

APPLICATION RUNTIME DOCS →

BOSH DOCS →

Download

The best way to experience Cloud Foundry is by using a [Certified Provider](#) offering. If you choose to download Cloud Foundry and run it yourself, here's how.

Installation



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 – CLI 설치

The screenshot shows a web browser displaying the Cloud Foundry Documentation at [docs.cloudfoundry.org](https://docs.cloudfoundry.org). The page is titled "Cloud Foundry CLI". A box highlights the "Installing the cf CLI" section. To the right, there are links for Java, Node.js, and Ruby buildpacks.

**Cloud Foundry CLI**  
Direct a CF deployment from your local command line.

**Installing the cf CLI**

- [Getting Started with the cf CLI](#)
- [Using the cf CLI with an HTTP Proxy Server](#)
- [Using the cf CLI with a Self-Signed Certificate](#)
- [Using cf CLI Plugins](#)
- [Developing cf CLI Plugins](#)
- [About Starting Applications](#)
- [cf CLI Reference Guide](#)

---

**Develop and Manage Applications**  
Develop, run, and track applications in the cloud.

- [Considerations for Designing and Running Apps in the Cloud](#)
- [Deploy an Application](#)
- [Deploying a Large Application](#)
- [Deploy an App with Docker](#)
- [Application Container Lifecycle](#)
- [Using Application Health Checks](#)
- [Deploying with Application Manifests](#)

**Java Buildpack**  
[Tips for Java Developers](#)  
[Grails](#)  
[Ratpack](#)  
[Spring](#)  
[Configure Service Connections for Grails](#)  
[Configure Service Connections for Play](#)  
[Configure Service Connections for Spring](#)  
[Cloud Foundry Eclipse Plugin](#)  
[Cloud Foundry Java Client Library](#)

**Node.js**  
[Node.js Buildpack](#)  
[Tips for Node.js Applications](#)  
[Environment Variables Defined by the Node Buildpack](#)  
[Configure Service Connections for Node.js](#)

**Ruby**  
[Ruby Buildpack](#)  
[Getting Started Deploying Ruby Apps](#)  
[Getting Started Deploying Ruby on Rails Apps](#)  
[Deploy a Sample Ruby on Rails App](#)  
[Configure Rake Tasks for Deployed Apps](#)  
[Tips for Ruby Developers](#)



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 – CLI 설치

The screenshot shows a web browser displaying the Cloud Foundry Documentation page for 'Installing the cf CLI'. The left sidebar lists various topics under 'CLI' and 'Information for Operators'. The main content area is titled 'Use an Installer' and contains instructions for Windows and Mac OS X installations. The 'Windows Installation' section is highlighted with an orange border.

**Use an Installer**

Follow the instructions for your operating system below.

### Windows Installation

To use the cf CLI installer for Windows, perform the following steps:

1. Download [the Windows installer](#).
2. Unpack the zip file.
3. Double click the `cf CLI` executable.
4. When prompted, click **Install**, then **Close**.
5. To verify your installation, open a terminal window and type `cf`. If your installation was successful, the cf CLI help listing appears.

### Mac OS X Installation

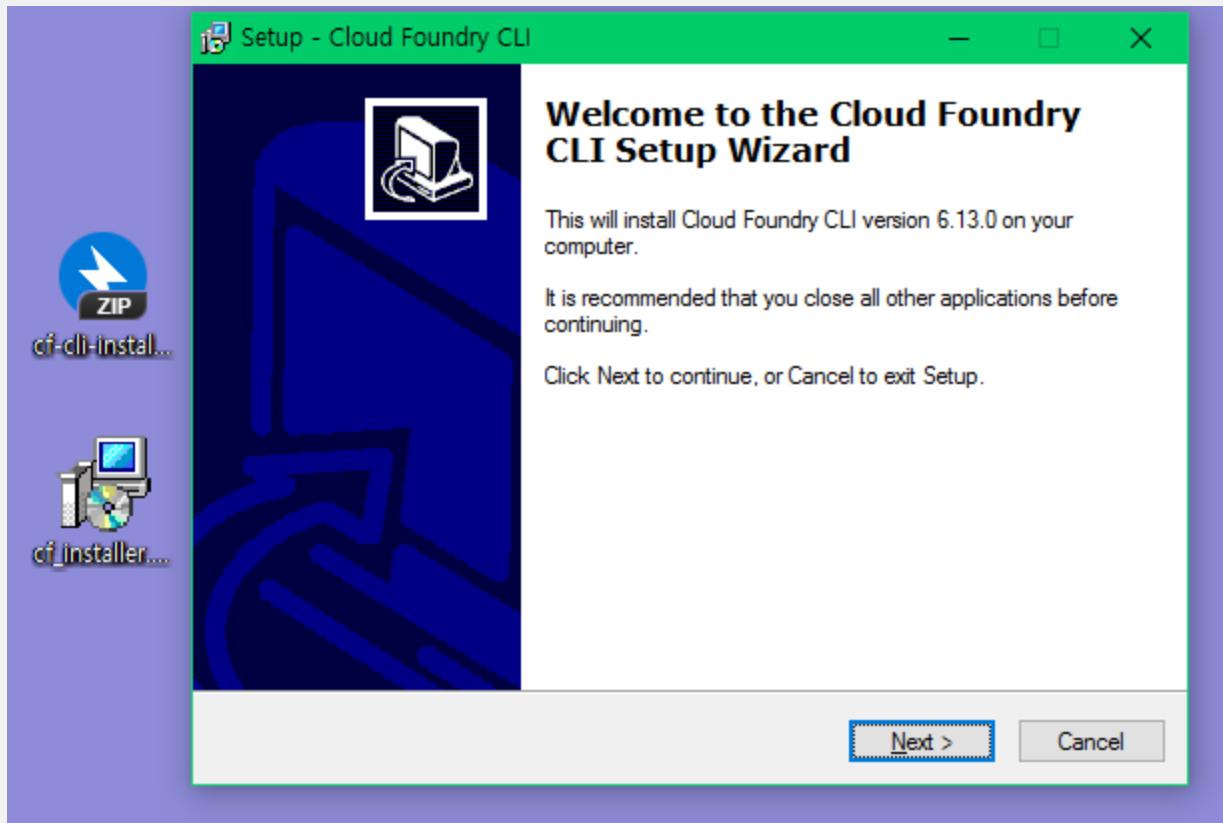
To use the cf CLI installer for Mac OS X, perform the following steps:

1. Download [the OS X installer](#).
2. Open the `.pkg` file.



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 – CLI 설치





# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 - login

```
관리자: 명령 프롬프트
C:\WINDOWS\system32>cf login -a https://api.paas-ta.org --skip-ssl-validation
API endpoint: https://api.paas-ta.org

Email> admin

Password>
Authenticating...
OK

Select an org (or press enter to skip):
There are too many options to display, please type in the name.

Org> infranics
Targeted org infranics

Select a space (or press enter to skip):
1. playpark
2. playpark-metering

Space> 1
Targeted space playpark


API endpoint: https://api.paas-ta.org (API version: 2.65.0)
User:      admin
Org:       infranics
Space:    playpark

C:\WINDOWS\system32>
```



# 05 Practice - CLI

CLI를 통한 Application Instance 관리 – Application 배포

CF명령어로 배포를 하기 위해 manifest.yml 를 작성한다.

```
---
applications:
- name: test01
  host: test01
  memory: 1024M
  instances: 1
  random-route: false
  path: ./test01.war
```

Name : 배포 된 앱의 이름

Host : Floating domain 앞에 들어갈 구분자 도메인

Memory : 앱의 최대 사용가능 메모리

Instances : 앱이 사용 할 instance 개수

Random-route : true값을 주면 URL을 랜덤으로 생성여부

Path: 배포하고자 하는 파일의 위치를 명시.

배포를 위한 cf 명령어는 "cf push"

이때 manifest.yml 파일이 있는 위치에 서 실행



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 – Application 배포

```
C:\Users\spring\Desktop\J\test0>cf push
Using manifest file C:\Users\spring\Desktop\J\test0\manifest.yml

Creating app test01 in org cloudpak / space devpack04 as devpack04...
OK

Using route test01.devpack.co.kr
Binding test01.devpack.co.kr to test01...
OK

Uploading test01...
Uploading app files from: C:\Users\spring\Desktop\J\test01\test01.war
Uploading 1.2K, 4 files
Done uploading
OK
```

배포를 위한 명령어는  
**\$ cf push**

```
requested state: started
instances: 1/1
usage: 1G x 1 instances
urls: test01.devpack.co.kr
last uploaded: Sat May 14 08:10:50 UTC 2016
stack: cflinuxfs2
buildpack: java-buildpack=v3.6-https://github.com/cloudfoundry/java-buildpack.git#5194155 open-jdk-like-jre=1.8.0_91-unlimited-crypto open-jdk-like-memory-calculator=2.0.2_RELEASE tomcat-access-logging-support=2.5.0_RELEASE tomcat-instance=8.0.33 tomcat-lifecycle...
```

state	since	cpu	memory	disk	de
tails					
#0	running	2016-05-14 05:11:41 PM	13.8%	141.5M of 1G	128.8M of 1G

배포 결과 running이  
확인되면 배포 완료!



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리

```
C:\Users\spring cf start test01
Starting app test01 in org cloudpak / space devpack04 as devpack04...
0 of 1 instances running, 1 starting
1 of 1 instances running

App started

OK

App test01 was started using this command 'CALCULATED_MEMORY=$($PWD/.java-buildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-2.0.2_RELEASE -memorySizes=metaspace:64m.. -memoryWeights=heap:75,metaspace:10,native:10,stack:5 -memoryInitials=heap:100%,metaspace:100% -totMemory=$MEMORY_LIMIT) && JAVA_HOME=$PWD/.java-buildpack/open_jdk_jre JAVA_OPTS="-Djava.io.tmpdir=$TMPDIR -XX:OutOfMemoryError=$PWD/.java-buildpack/open_jdk_jre/bin/killjava.sh $CALCULATED_MEMORY -Daccess.logging.enabled=false -Dhttp.port=$PORT" exec $PWD/.java-buildpack/tomcat/bin/catalina.sh run'

Showing health and status for app test01 in org cloudpak / space devpack04 as devpack04...
OK

requested state: started
instances: 1/1
usage: 512M x 1 instances
urls: test01.devpack.co.kr
last uploaded: Sat May 14 05:32:06 UTC 2016
stack: cflinuxfs2
buildpack: java-buildpack=v3.6-https://github.com/cloudfoundry/java-buildpack.git#5194155 maria-db-jdbc=1.4.4 open-jdk-like-jre=1.8.0_91-unlimited-crypto open-jdk-like-memory-calculator=2.0.2_RELEASE tomcat-access-logging-support=2.5.0_RELEASE tomcat-instance=8.0...

     state      since          cpu    memory       disk
details
#0  running   2016-05-14 02:48:36 PM  0.0%  142.8M of 512M  129.1M of 1G
```

CF 명령어를 사용하여 시작/중지/재시작을 수 행한다.

CF에 로그인이 되어있다는 전제 하에 이 과정을 시행한다.

시작을 위한 명령어는  
**\$ cf start 앱이름**

중지할 위한 명령어는  
**\$ cf stop 앱이름**

재시작을 위한 명령어는  
**\$ cf restart 앱이름**

App상태를 확인하기 위한 명령어  
**\$ cf app 앱이름**  
**\$ cf apps**

앱을 삭제하기 위한 명령어는  
**\$ cf delete 앱이름**



# 05 Practice - CLI

CLI를 통한 Application Instance 관리 – Service 관리

Service 를 사용할 수 있는 리스트 확인 **\$ cf marketplace**

```
관리자: 명령 프롬프트
C:\>cf marketplace
Getting services from marketplace in org cloudpak / space systeer-dev as systeer-dev...
OK

service      plans          description
mysql        5gb*, 1gb*, 10gb* MySQL databases on demand
rabbitmq     standard*      RabbitMQ is a robust and scalable high-performance multi-protocol
messaging broker.
redis-100mb   shared-vm*    Redis service to provide a key-value store
redis-250mb   shared-vm*    Redis service to provide a key-value store
redis-30mb    shared-vm*    Redis service to provide a key-value store

* These service plans have an associated cost. Creating a service instance will incur this cost.

TIP: Use 'cf marketplace -s SERVICE' to view descriptions of individual plans of a given service.

C:\>
```



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 – Service 연결

Service 생성 및 확인 (bound apps가 비어있음-연결되지 않았으므로)

\$ cf create-service 서비스명 플랜명 인스턴스명

\$ cf services

```
C:\>cf create-service mysql 1gb test_mysql_db
Creating service instance test_mysql_db in org cloudpak / space systeer-dev as systeer-dev...
OK

Attention: The plan `1gb` of service `mysql` is not free. The instance `test_mysql_db` will incur a
cost. Contact your administrator if you think this is in error.

C:\>cf services
Getting services in org cloudpak / space systeer-dev as systeer-dev...
OK

name      service  plan   bound apps  last operation
test_mysql_db  mysql    1gb        create succeeded

C:\>
```



# 05 Practice - CLI

## CLI를 통한 Application Instance 관리 – 서비스 확인

생성한 Service를 Application에 연결 및 연결 확인 (bound apps 애플리케이션 명 확인)

\$ cf bind-service [애플리케이션 명] [서비스 인스턴스 명]

\$ cf services

```
C:\>cf bind-service welcome_cf test_mysql_db
Binding service test_mysql_db to app welcome_cf in org cloudpak / space systeer-dev as systeer-dev.
.
OK
TIP: Use 'cf restage welcome_cf' to ensure your env variable changes take effect

C:\>cf services
Getting services in org cloudpak / space systeer-dev as systeer-dev...
OK

name      service    plan   bound apps   last operation
test_mysql_db  mysql     1gb    welcome_cf  create succeeded

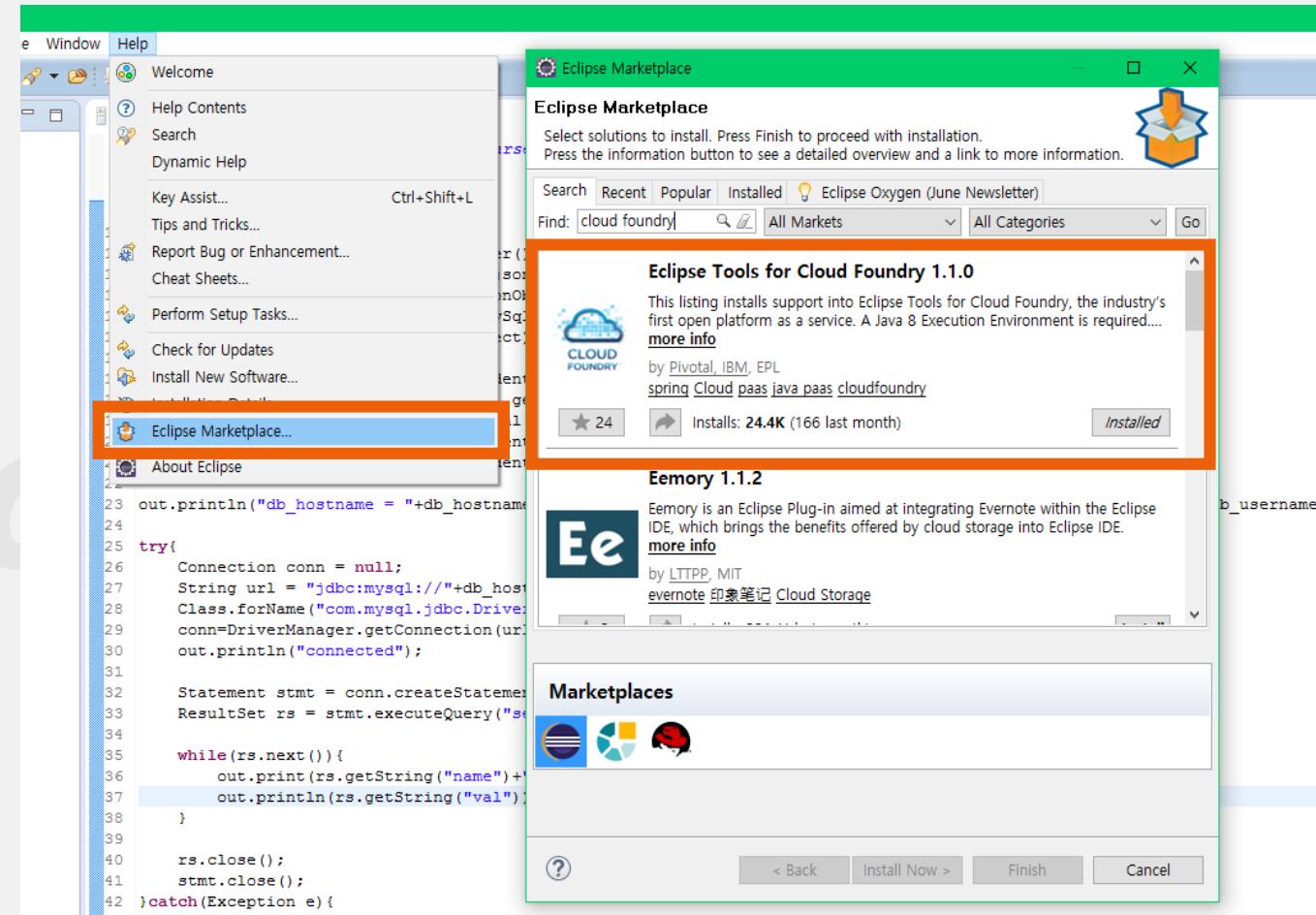
C:\>
```

# 05 Practice – IDE tool(eclipse)

IDE Plugin를 통한 Application Instance 관리

플러그인은 Eclipse v3.8 및 v4.3 (Java EE 버전 권장) 및 STS 3.0.0 이상을 지원합니다. – 최신버전 권장(기본 설치 되어있음 oxygen)

Eclipse > Help >  
Eclipse  
Marketplace 선택  
후 Cloud Foundry  
를 검색 후 Install

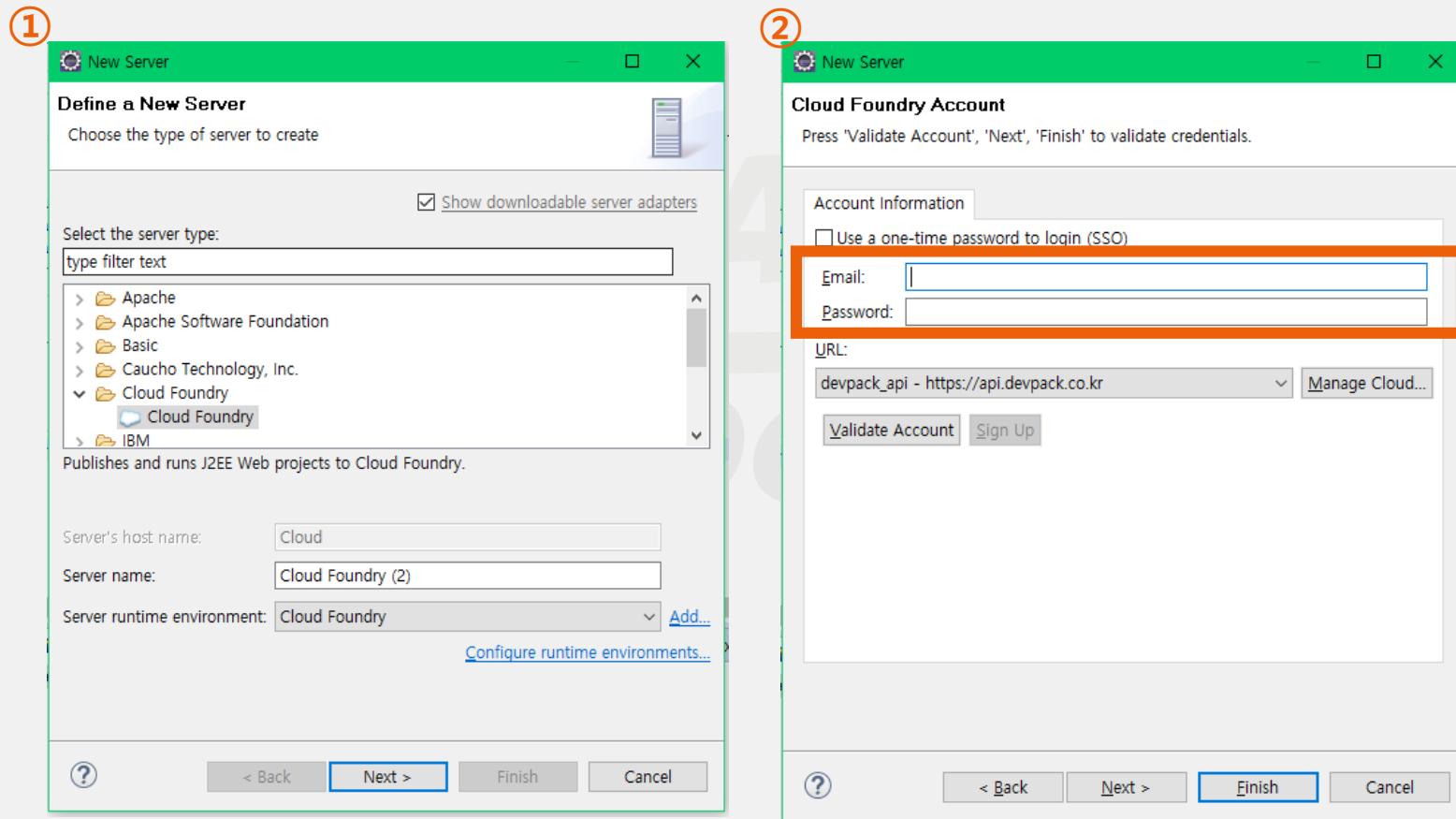


# 05 Practice – IDE tool(eclipse)

## IDE Plugin을 통한 Application Instance 관리

Server > new

*Cloud Foundry선택*



# 05 Practice – IDE tool(eclipse)

IDE Plugin를 통한 Application Instance 관리

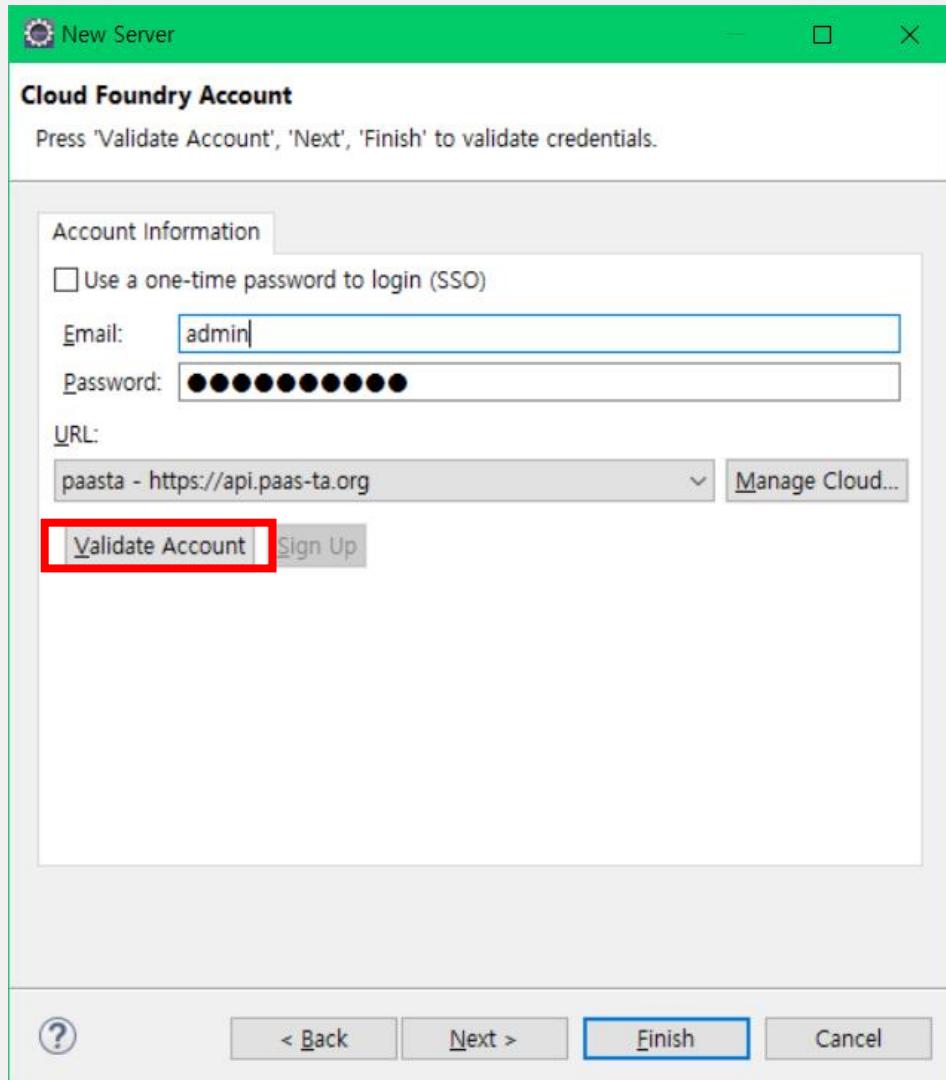
PaaS-TA API endpoint를 지정

The screenshot shows two windows from the Eclipse IDE:

- New Server (Left Window):** A "Cloud Foundry Account" configuration screen. It includes fields for "Email" and "Password", a "URL" dropdown containing "paasta - https://api.paas-ta.org", and a "Manage Cloud..." button. The "Manage Cloud..." button is highlighted with a red box.
- Manage Cloud URLs (Right Window):** A list of URLs for different server types. It shows one entry for "paasta" with URL "https://api.paas-ta.org". There are "Add...", "Edit...", and "Remove" buttons. The "Add..." button is highlighted with a red box.
- Add and validate a Cloud URL (Bottom Window):** A sub-dialog for adding a new cloud URL. It has a "Name" field set to "Hackathon", a "URL" field containing "https://api.paas-ta.org", and a checked "Trust self-signed certificate" checkbox. Both the "URL" field and the "Trust self-signed certificate" checkbox are highlighted with red boxes.

# 05 Practice – IDE tool(eclipse)

## IDE Plugin를 통한 Application Instance 관리

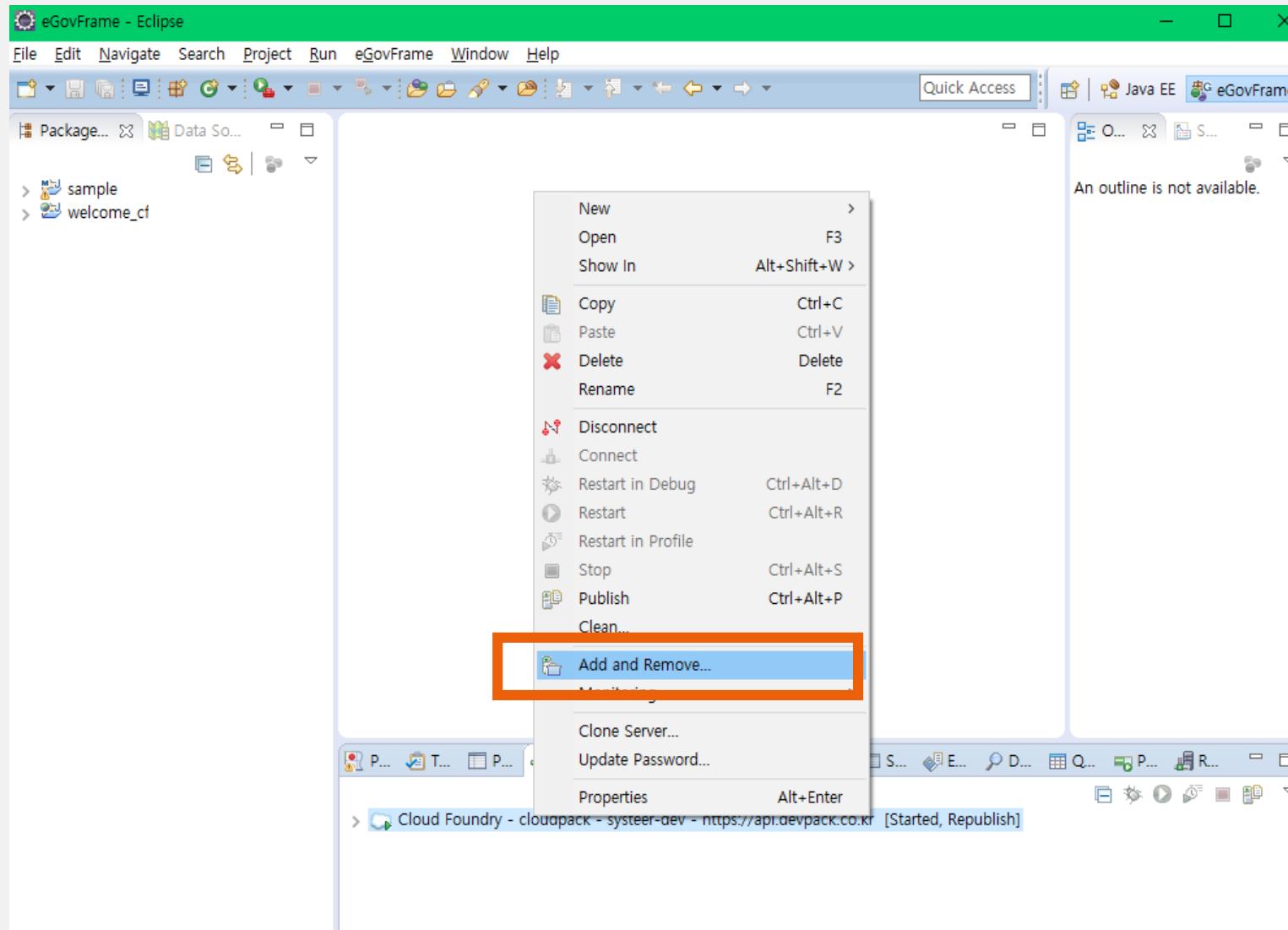


validate account버튼을 눌러 계정 인증 후 정상 인증완료가 되면 Finish버튼을 눌러 server 생성을 완료한다.

이제 생성된 server에 개발 프로젝트를 배포할 수 있다.

# 05 Practice – IDE tool(eclipse)

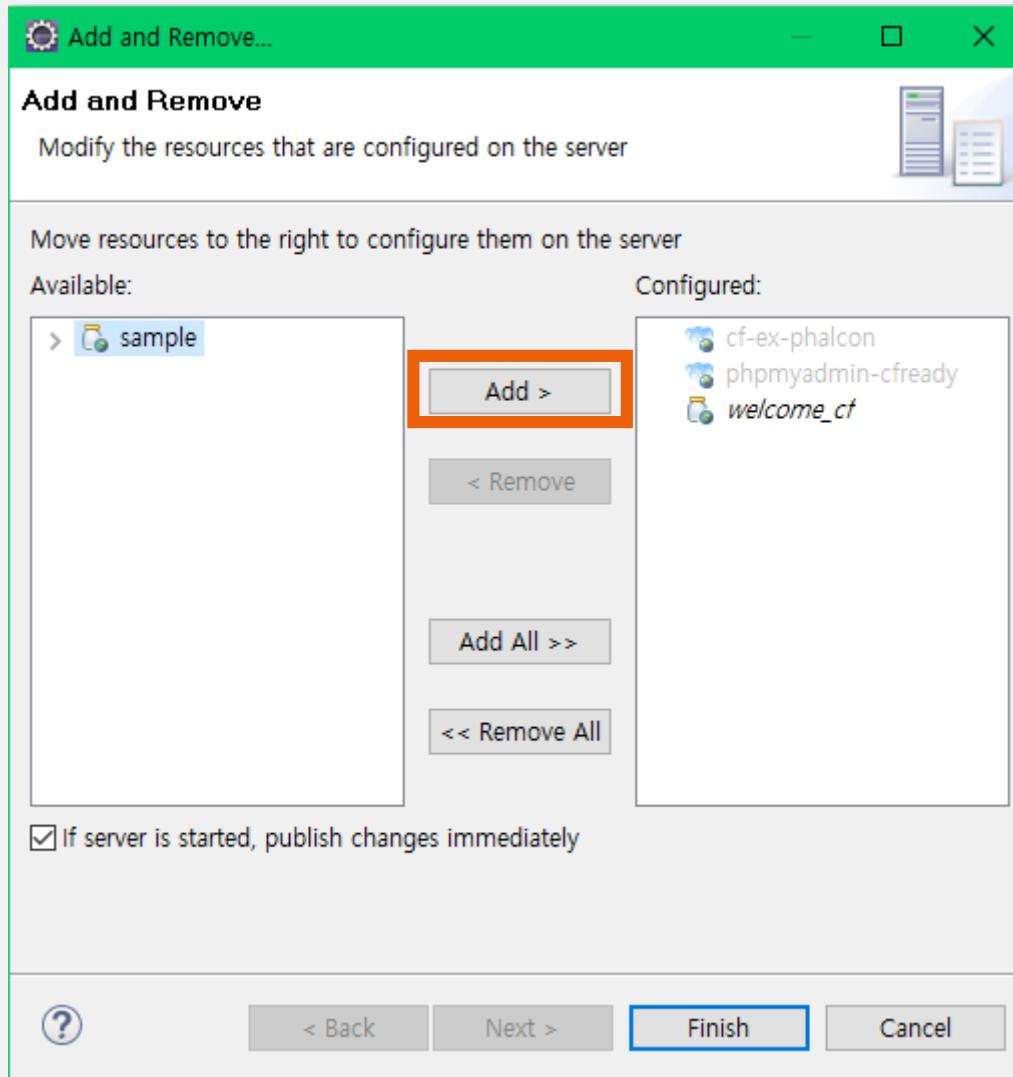
## IDE Plugin를 통한 Application Instance 관리



생성한 Cloud Foundry 서버목록에 우 클릭을 하여 Add and Remove를 선택 한다.

# 05 Practice – IDE tool(eclipse)

## IDE Plugin를 통한 Application Instance 관리



배포하고자 하는 개발 프로젝트를 선택하고 “Add>”버튼을 클릭하여 서버에 배포 목록으로 추가 한다.

# 05 Practice – IDE tool(eclipse)

## IDE Plugin를 통한 Application Instance 관리

The figure consists of three screenshots of the Eclipse IDE Application plugin interface:

- Application details:** Shows the 'Name' field set to 'mariiner' (highlighted with a red box).
- Launch deployment:** Shows the 'Subdomain' field set to 'mariiner' (highlighted with a red box), 'Domain' set to 'paas-ta.org', 'Memory Limit (MB)' set to '512' (highlighted with a red box), and the 'Start application on deployment' checkbox checked.
- Services selection:** Shows a list of services to bind to the application. The 'marin' service is selected (highlighted with a red box). The table data is as follows:

Name	Service	Plan	Version
abc			
cf_log_drain			
log_test2			
log_test3	Mariner4	non...	
marin	Mysql-DB	Mys...	
mysql	Semon	non...	
semon			

The 'Finish' button at the bottom right of the Services selection window is highlighted with a red box.

1. Space에서 구분자를 Application명 설정
2. Subdomain명 설정
3. 연결할 Service설정

# 05 Practice – IDE tool(eclipse)

## IDE Plugin를 통한 Application Instance 관리

The screenshot shows the Eclipse IDE interface with the Cloud Foundry plugin installed. The main window is titled 'Cloud Foundry' and contains several panes:

- Applications**: A list of deployed applications. One application, 'welcome\_cf - Deployed [Started]', is highlighted with a blue selection bar. A red circle with the number 3 is positioned to the left of this list.
- Services**: A table showing a single service named 'test-mysql' with a MySQL plan. A red circle with the number 2 is positioned to the left of this table.
- Routes**: A section with a 'Remove...' button.
- Instances**: A table showing one instance of the 'welcome\_cf' application with details like Host (192.168.1.31), Port (60388), CPU (0.001...), Memory (212M (512...)), Disk (157M (102...)), and Uptime (24h:37m:...).
- General**: Application details for 'welcome\_cf'. It includes fields for Name (welcome\_cf [Started]), Mapped URLs (welcome-cf.devpack.co.kr), Instances (1), and Manifest (Save). A red circle with the number 4 is positioned above the 'Application Operations' section.
- General (Application Restart Required)**: Configuration for memory limit (512 MB) and environment variables (Edit...).
- Application Operations**: Buttons for Restart, Stop, Update and Restart, Push, and Debug. This section is also highlighted with a red box.
- Application Services**: A table showing the bound service 'test-mysql'.

At the bottom, there are tabs for Overview, Applications and Services, and a toolbar with various developer tools.

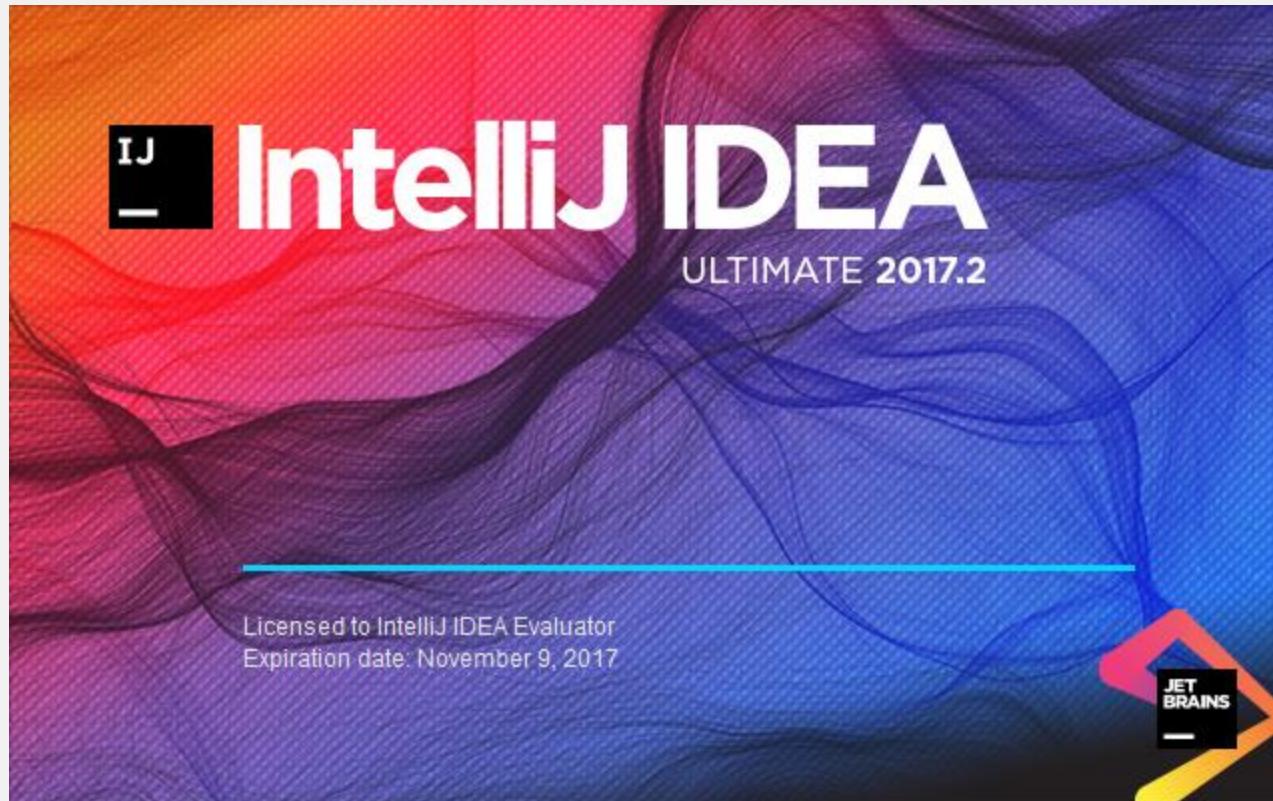
③ 애플리케이션 리스트 : 클라우드상에 올라가 있는 앱을 확인. (삭제도 가능)

④ 앱 컨트롤 : 앱을 배포 업데이트 시작/중지/재시작

\*기타 연결 서비스 상태 메모리 등을 확인 관리 가능

# 05 Practice – IDE tool(IntelliJ)

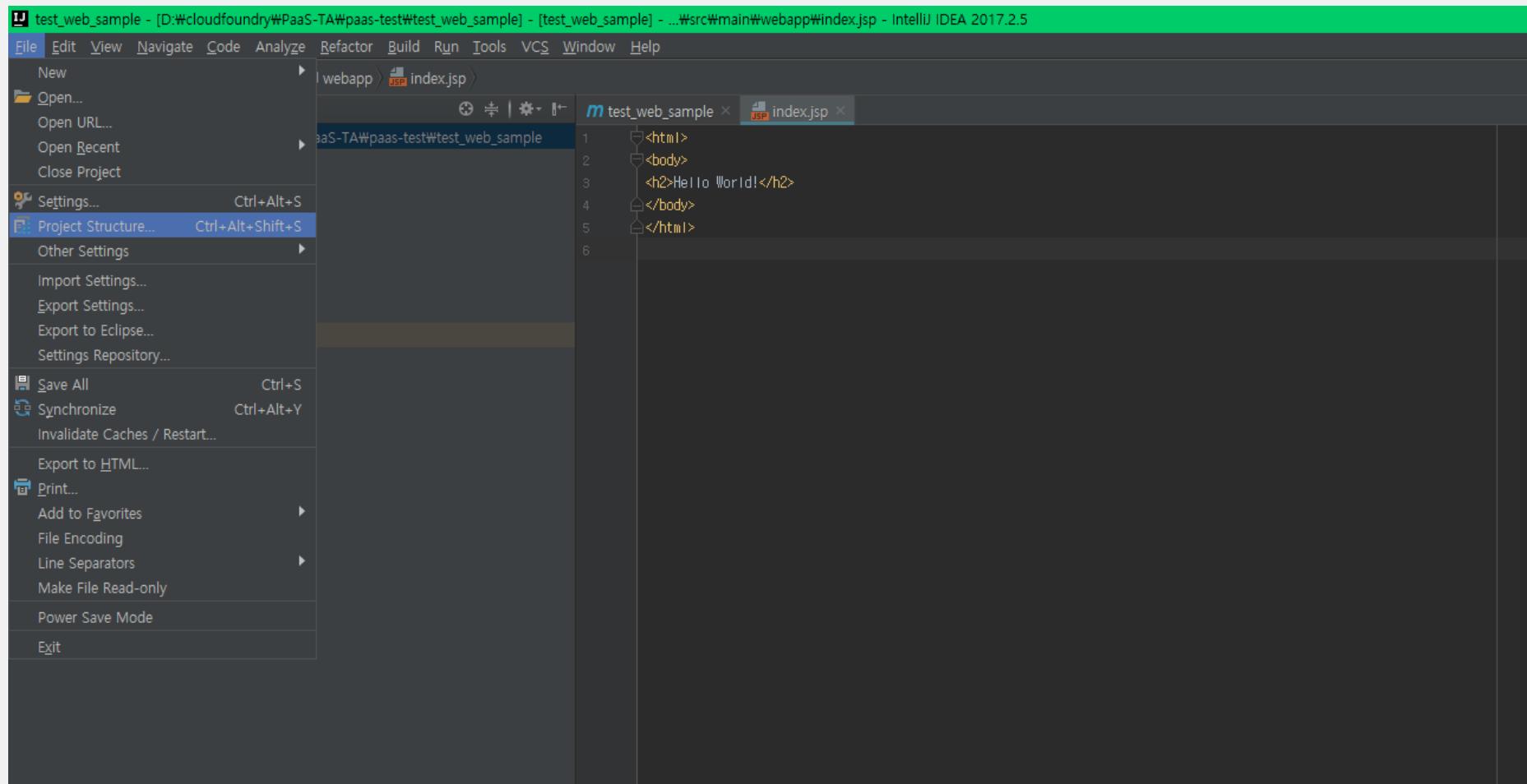
IDE Plugin를 통한 Application Instance 관리



# 05 Practice – IDE tool(IntelliJ)

IDE Plugin를 통한 Application Instance 관리

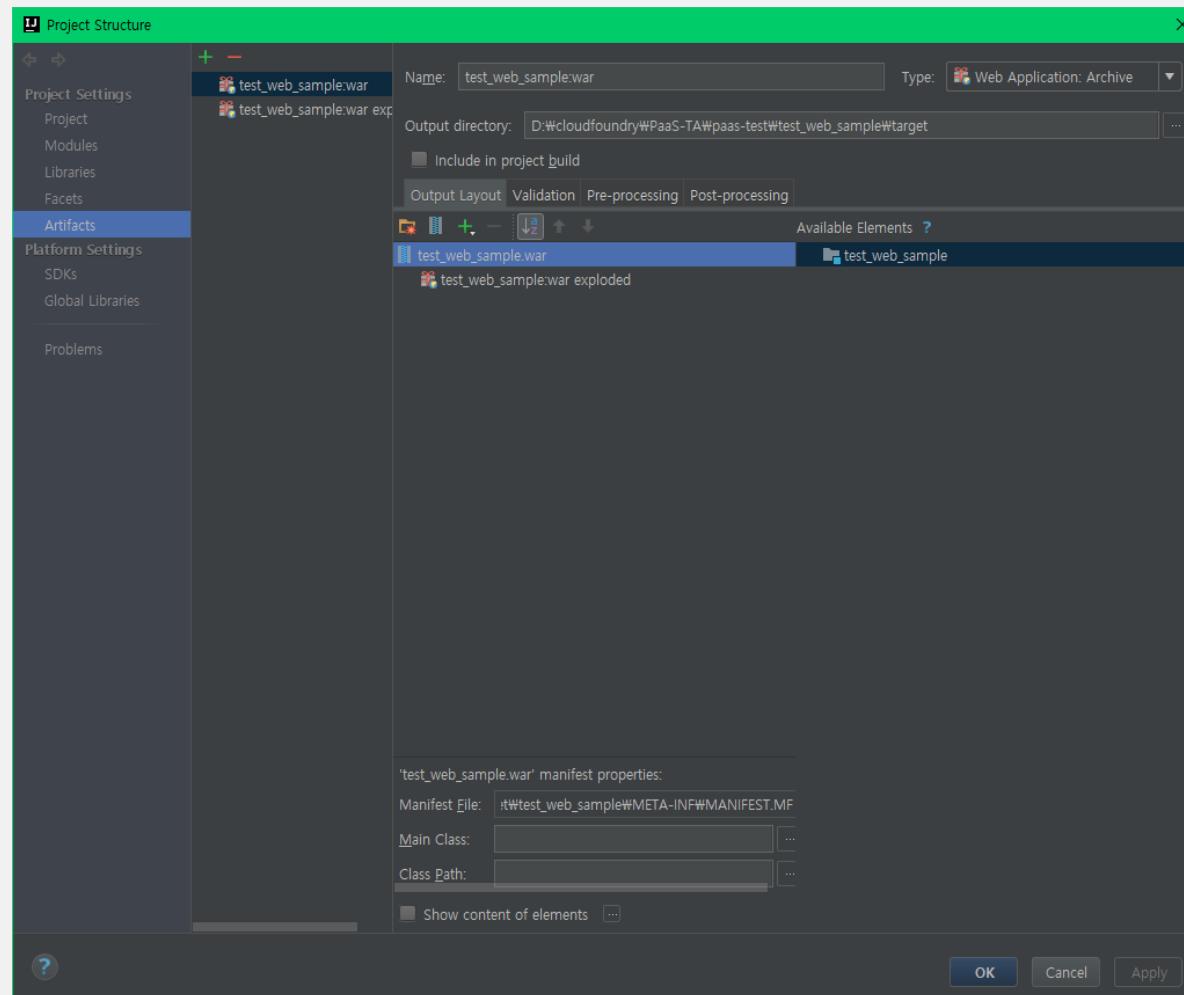
Project Structure에서 Artifact를 설정해야 Cloud Foundry를 사용할 수 있다.



# 05 Practice – IDE tool(IntelliJ)

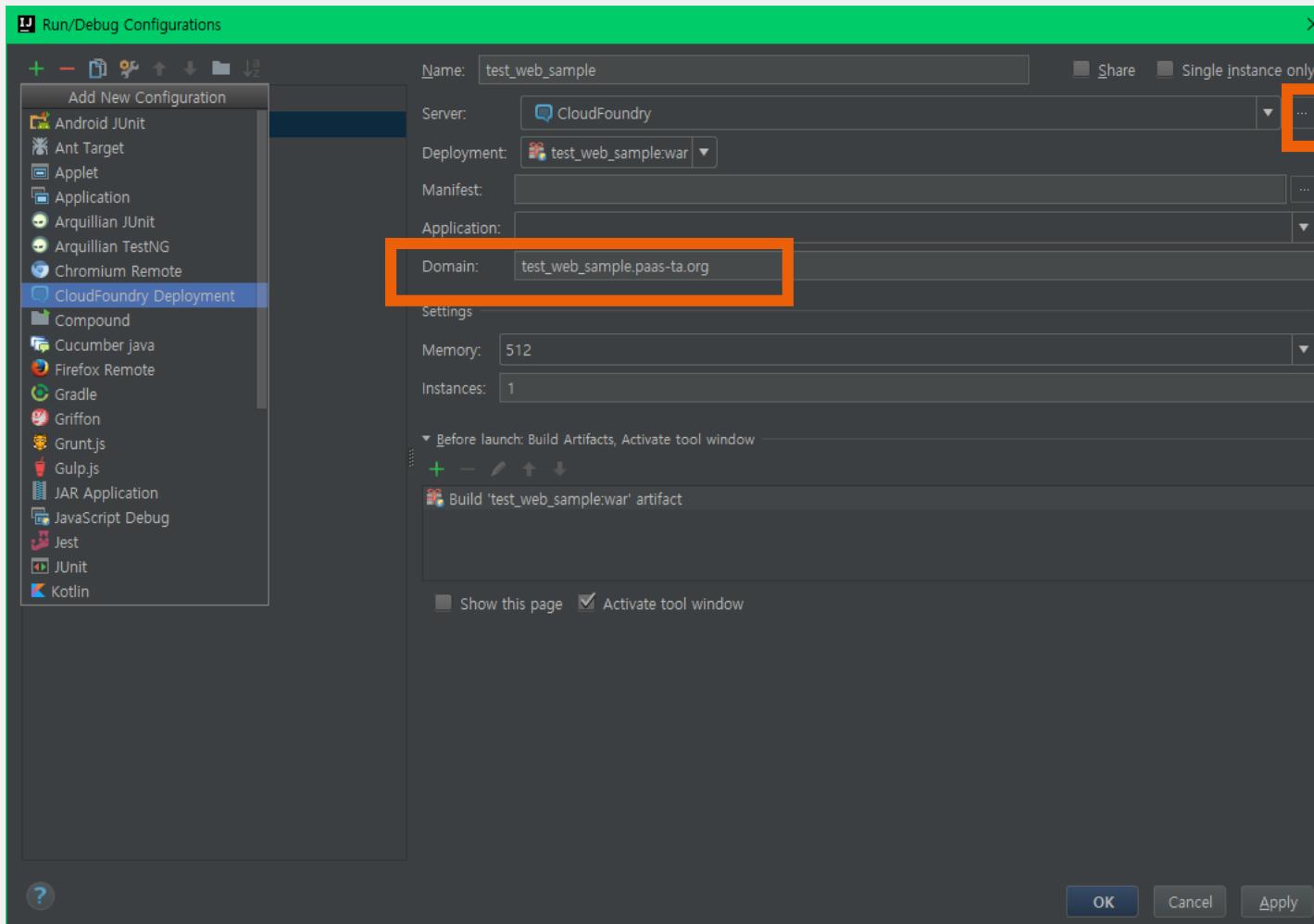
IDE Plugin를 통한 Application Instance 관리

Project Structure에서 Artifact를 war로 설정한 예제



# 05 Practice – IDE tool(IntelliJ)

## IDE Plugin를 통한 Application Instance 관리



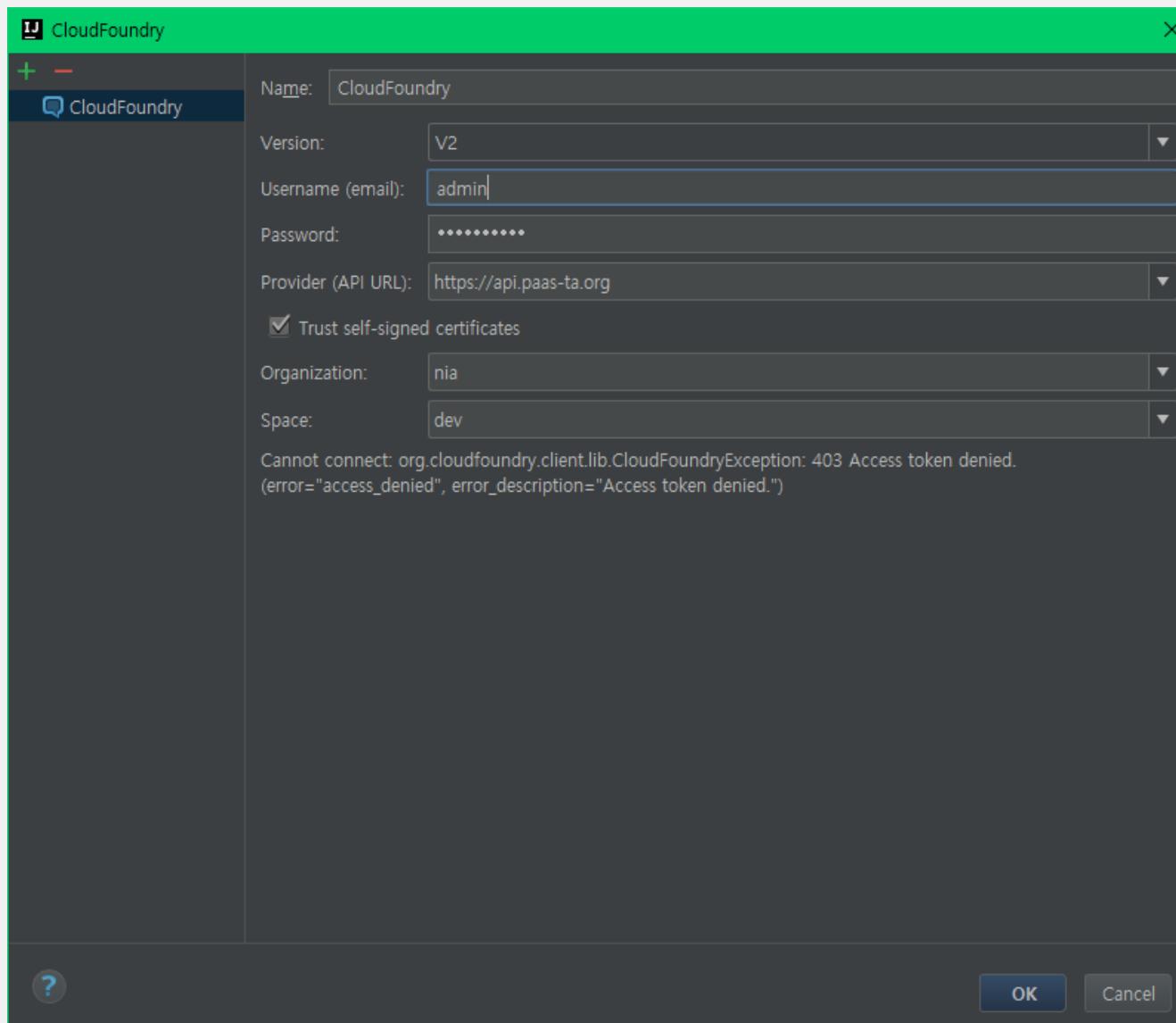
Run메뉴에서  
Cloudfoundry를 선  
택

1. 서버 선택

2. Domain은 full  
Domain을 입력

# 05 Practice – IDE tool(IntelliJ)

## IDE Plugin를 통한 Application Instance 관리

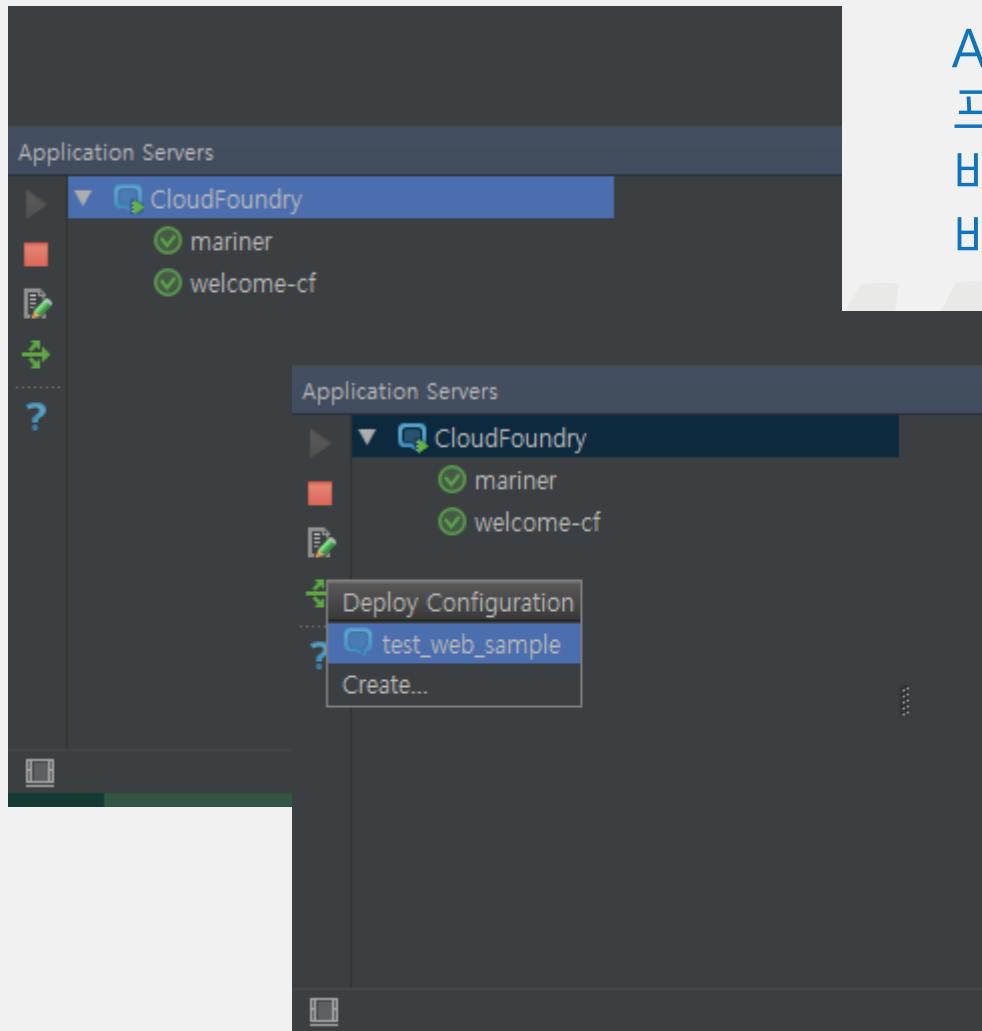


PaaS-TA Endpoint API  
주소 및 계정정보 입력

IER

# 05 Practice – IDE tool(IntelliJ)

## IDE Plugin를 통한 Application Instance 관리



Application Server 리스트에서  
프로젝트 배포 및  
배포된 Application 정보 확인 가능  
배포 및 삭제만 됨.

# THANK YOU