

# ABOUT CLOUD



# CONTENT



## 01.CLOUD 개념 소개

CLOUD 기본 개념과 특징

## 02. CLOUD 기반 개발

CLOUD NATIVE APPLICATION 이란

source from

1. aws practitioner lec (udemy)
2. spring cloud msa lec (inflearn)



# 01.CLOUD 개념 소개



# CLOUD 기본 특성

## 개념

인터넷을 통해 액세스할 수 있는 서버와 이러한 서버에서 작동하는 소프트웨어와 데이터베이스

01

On Demand Self Service

사용자가 서비스 관리자의 개입 없이 원하는 시점에 서비스를 바로 사용할 수 있어야 한다

02

Broad Network Access

네트워크 기반의 서비스 접속을 통하여 다양한 클라이언트에 의해 접속 가능해야 한다.

03

Resource Pooling

자원은 풀(Pool)로 관리되며, 사용자의 요청에 의해 할당되거나 다시 풀로 반환되어야 한다.  
자원의 물리적인 위치, 크기 등에 대해서는 모르고 자원을 추상화시켜 제공한다.

04

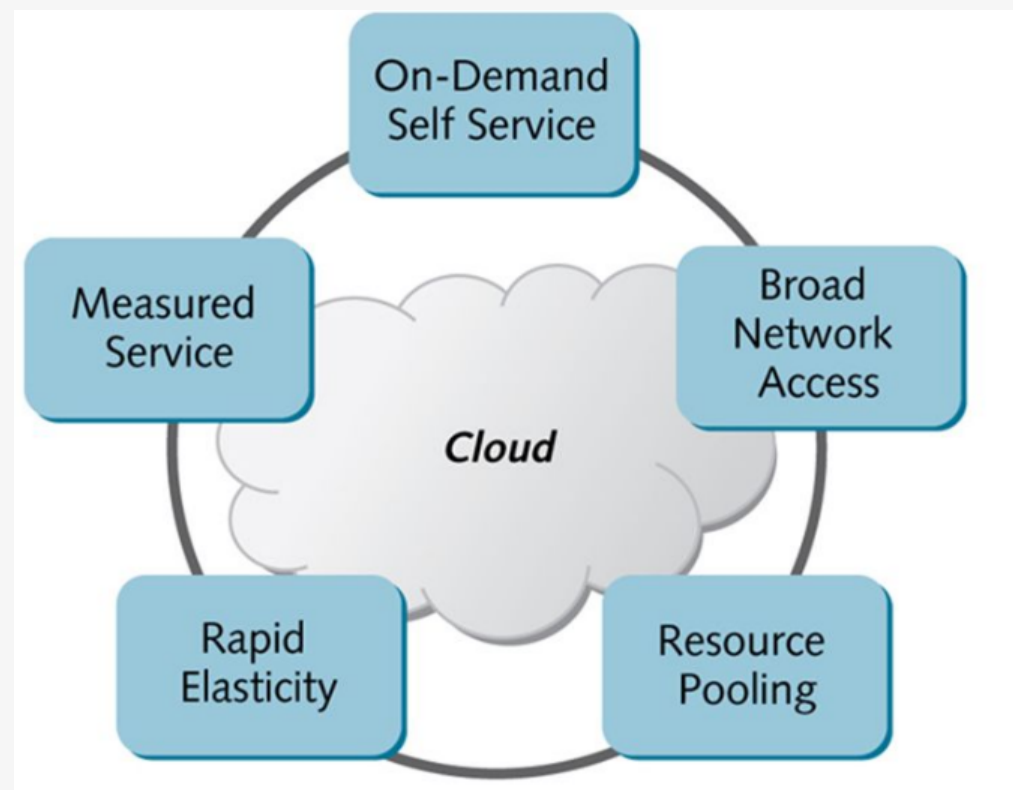
Rapid Elasticity

자원을 무한대로 확장할 수 있거나 필요한 만큼의 수준으로 마음대로 줄일 수 있어야 한다.  
이러한 작업은 수분 이내로 작업을 할 수 있어야 한다.

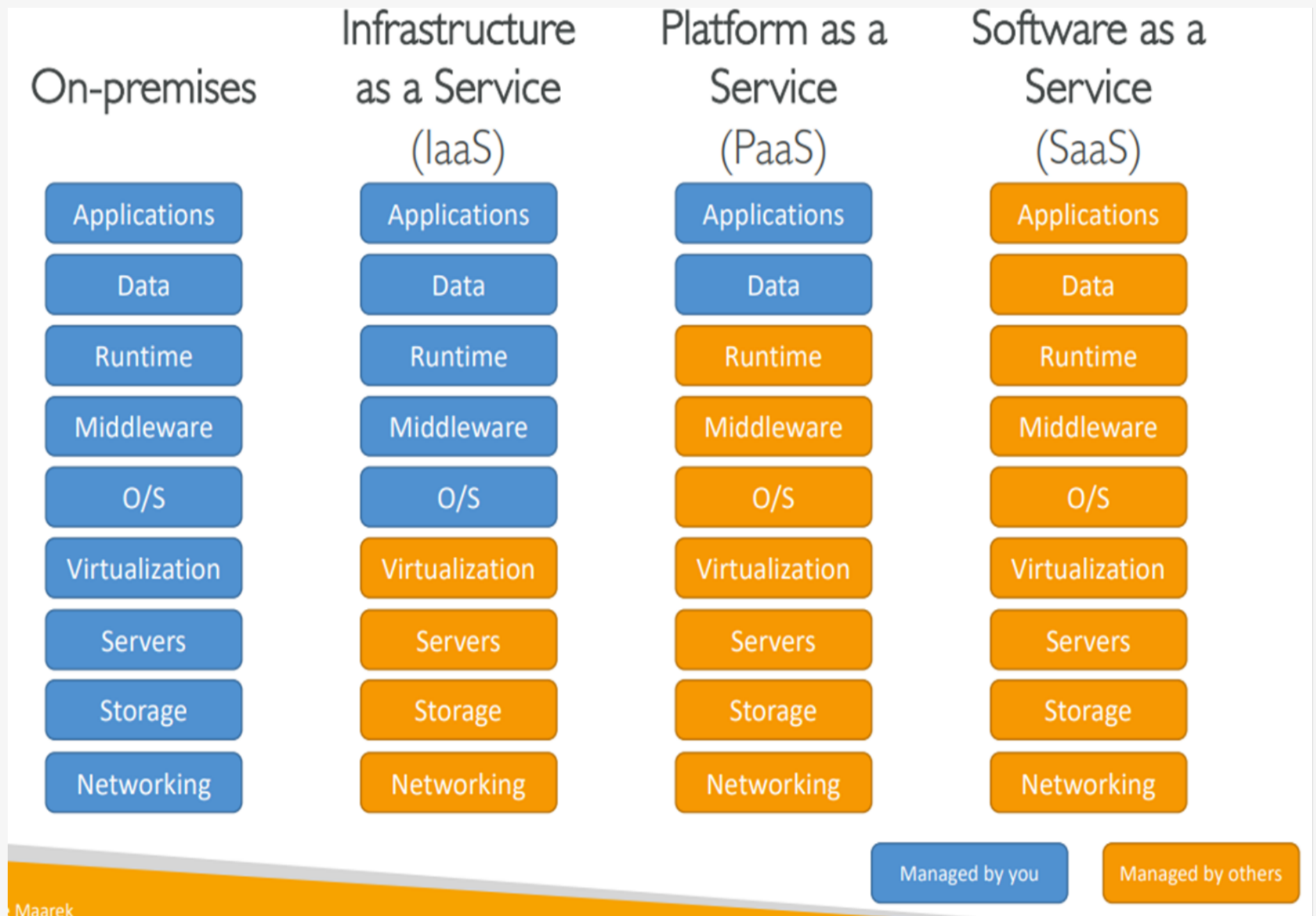
05

Measured Service

자원의 사용량이 실시간으로 수집되고 모니터링 되어 사용량에 따른 과금을 할 수 있어야 한다.



# CLOUD 종류

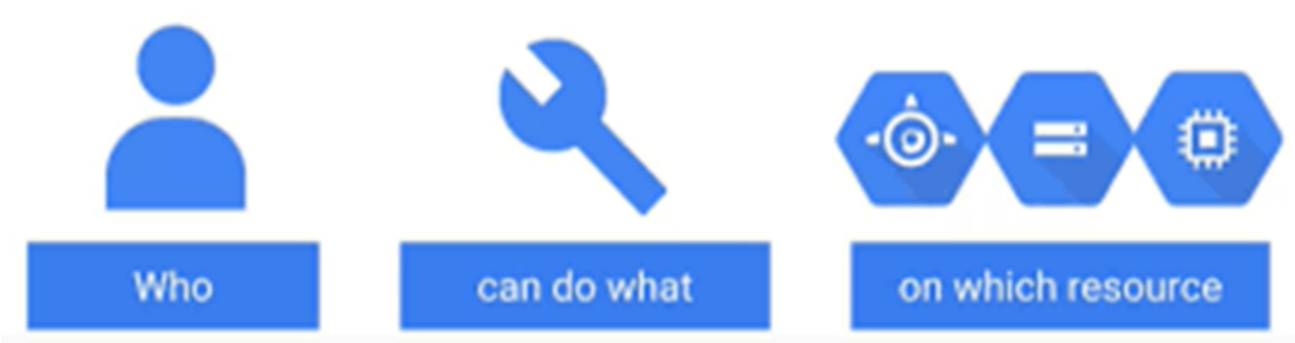


# CLOUD 개발 이점

## 01.Instance 단위 개발

General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
<div>A1</div> <div>ARM based core and custom silicon</div>	<div>C4</div> <div>Compute - CPU intensive apps and DBs</div>	<div>R4</div> <div>RAM - Memory intensive apps and DBs</div>	<div>P2</div> <div>Processing optimised - Machine Learning</div>	<div>H1</div> <div>High Disk Throughput - Big data clusters</div>
<div>T2</div> <div>Tiny - Web servers and small DBs</div>		<div>X1</div> <div>Xtreme RAM - For SAP/Spark</div>	<div>G3</div> <div>Graphics Intensive - Video and streaming</div>	<div>I3</div> <div>IOPS - NoSQL DBs</div>
<div>M4</div> <div>Main - App servers and general purpose</div>		<div>z1d</div> <div>High Compute and High Memory - Gaming</div>	<div>F1</div> <div>Field Programmable - Hardware acceleration</div>	<div>D2</div> <div>Dense Storage - Data Warehousing</div>

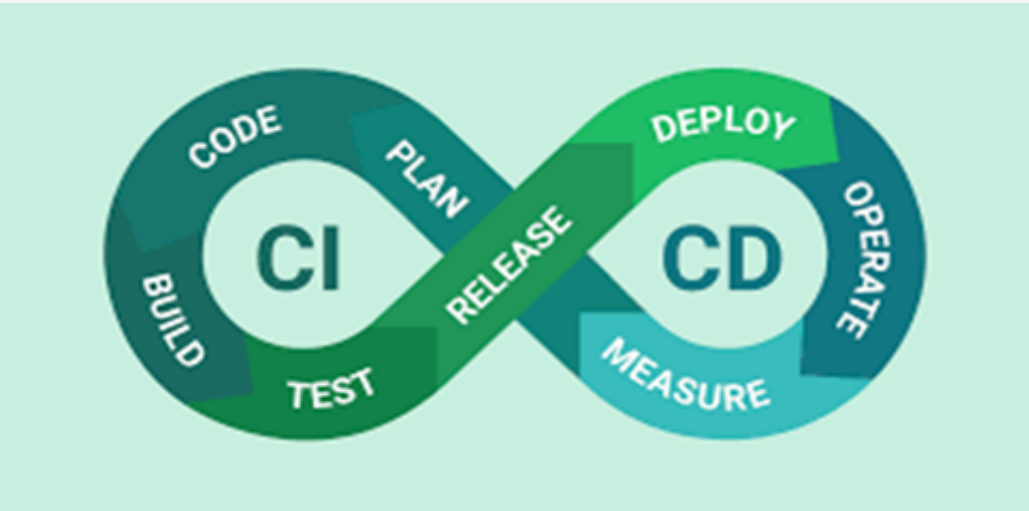
## 02. 명확한 권한과 책임



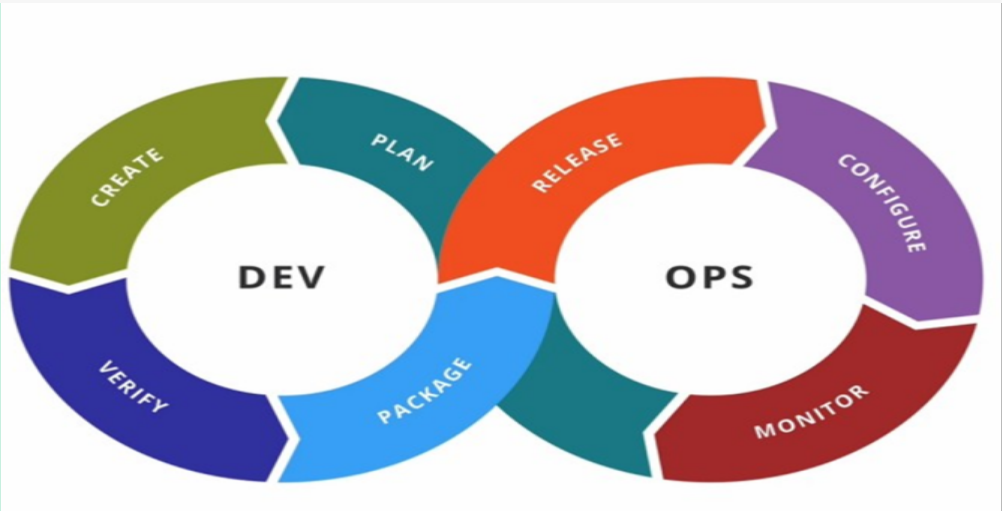
## 02. CLOUD 기반 개발



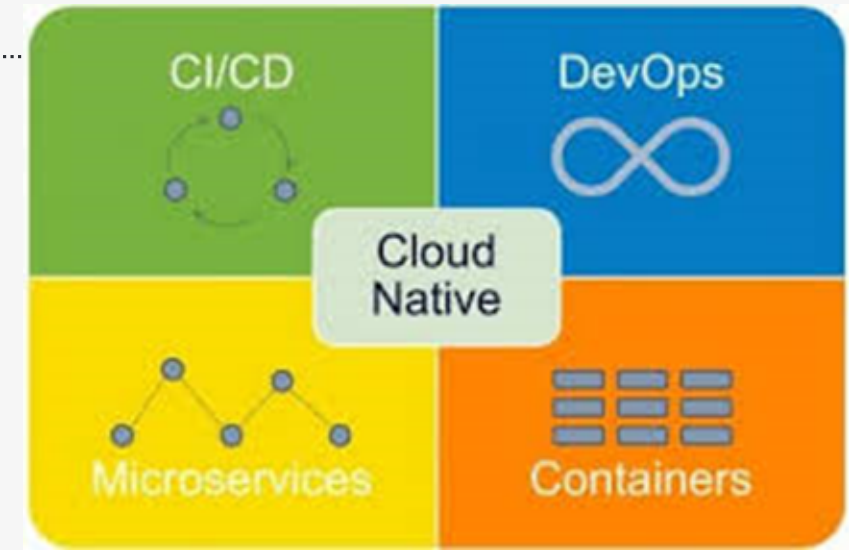
# CLOUD NATIVE APPLICATION



CICD

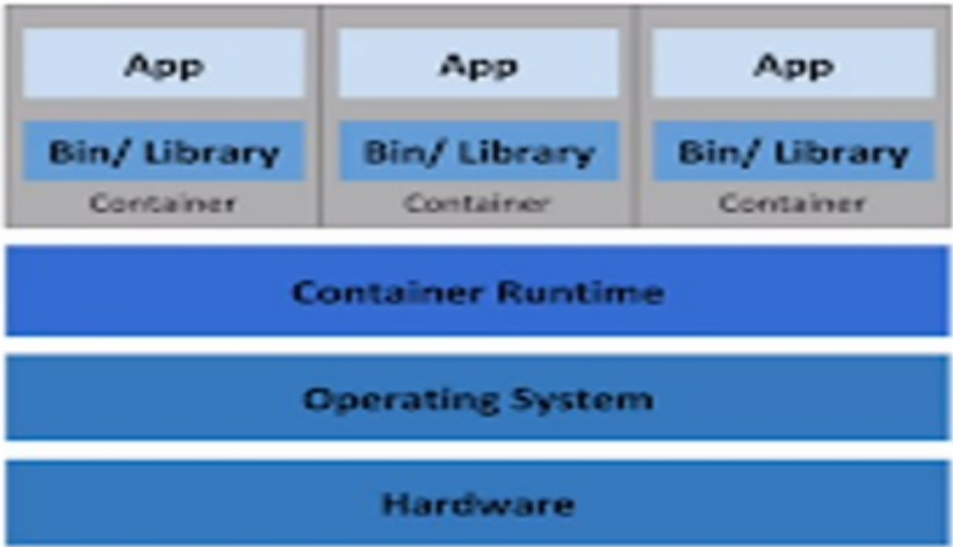
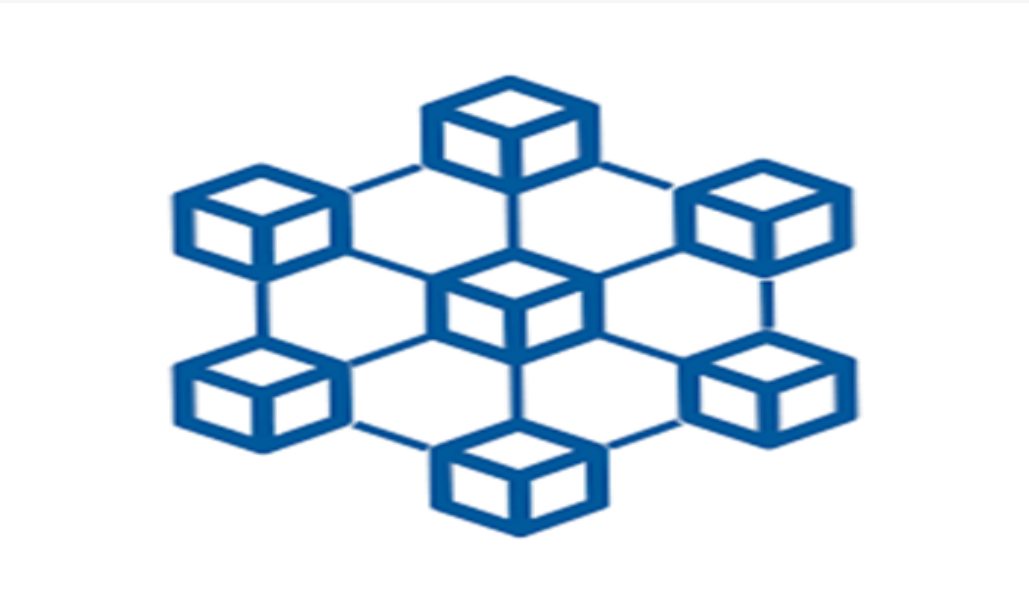


DevOps



MSA

Containers





# 12 FACTORS

---



## 12 Factor App Principles

### Codebase

One codebase tracked in revision control, many deploys

### Dependencies

Explicitly declare and isolate the dependencies

### Config

Store configurations in an environment

### Backing Services

Treat backing resources as attached resources

### Build, release, and, Run

Strictly separate build and run stages

### Processes

Execute the app as one or more stateless processes

### Port Binding

Export services via port binding

### Concurrency

Scale-out via the process model

### Disposability

Maximize the robustness with fast startup and graceful shutdown

### Dev/prod parity

Keep development, staging, and production as similar as possible

### Logs

Treat logs as event streams

### Admin processes

Run admin/management tasks as one-off processes