

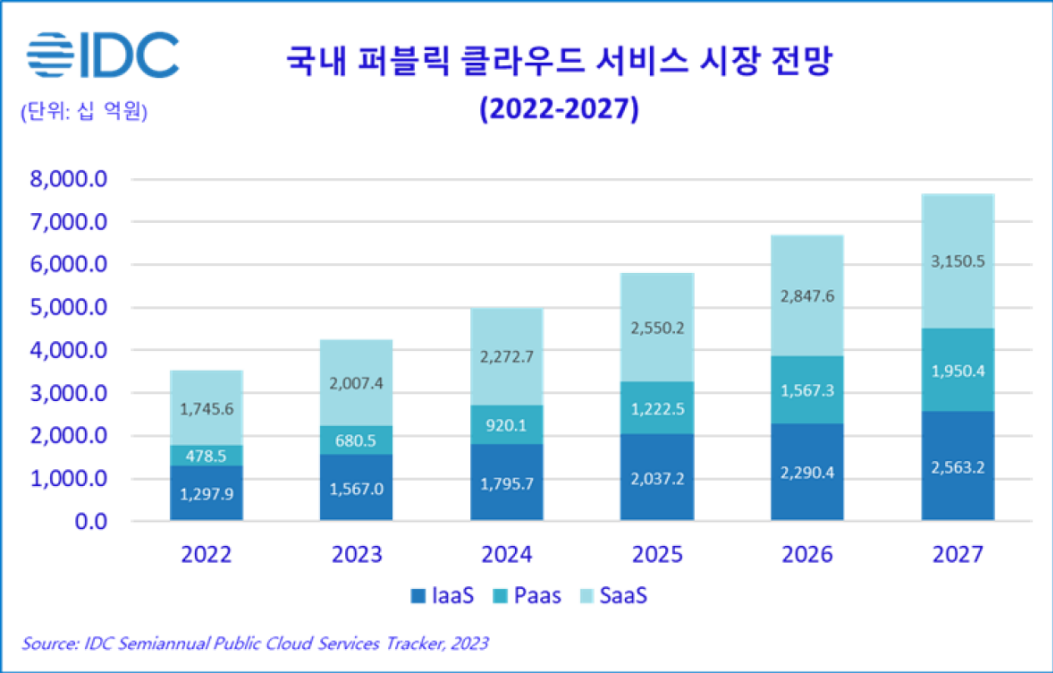
# 클라우드 컴퓨팅 입문

## Cloud Computing Overview 2

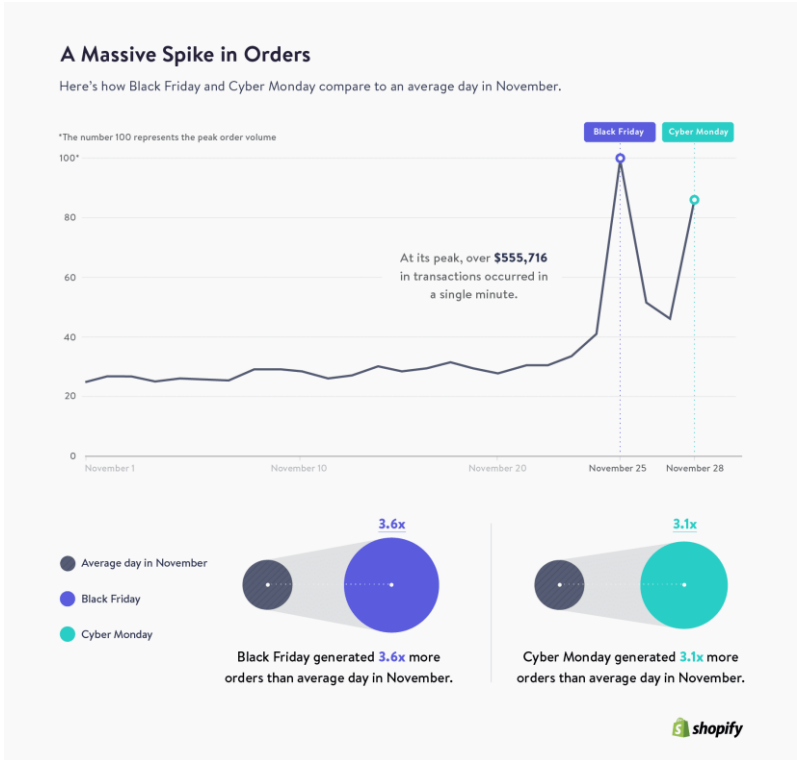
# Why is Cloud Computing Important?

# 소프트웨어 개발자 혹은 데이터 분석가로서 클라우드를 꼭 알아야 하나요?

만약 큰 조직에서 나에게 주어진 업무만 수행하는 경우가 아니라면 이제는 클라우드 컴퓨팅에 대한 지식은 필수적입니다.



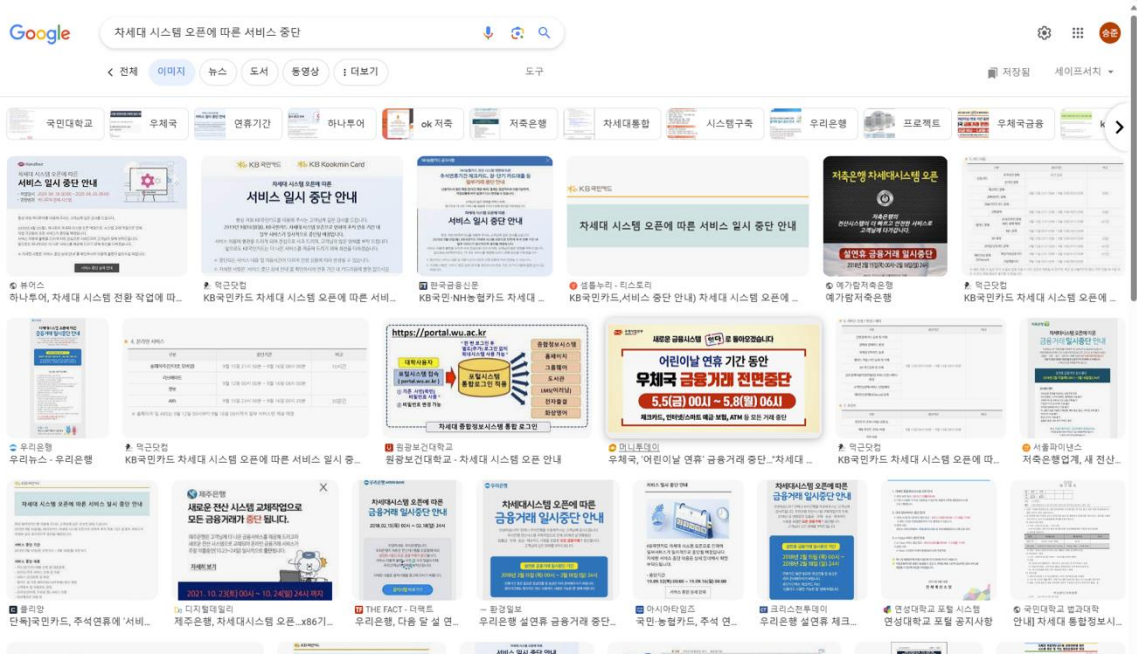
<http://www.itdaily.kr/news/articleView.html?idxno=217094>



<https://www.smartmail.com/blog/guide-to-black-friday-ecommerce-email-marketing/>

# Continuous Integration & Continuous Delivery (지속적인 통합 및 배포)

CI/CD는 현대 소프트웨어 개발 방법론의 핵심 아이디어 중 하나로서 Digital Transformation (디지털 전환)이 핵심 비즈니스 역량이 되는 시대에 기업이 반드시 갖추어야 할 요소입니다.



차세대 시스템 오픈에 따른 서비스 중단 - Google 검색



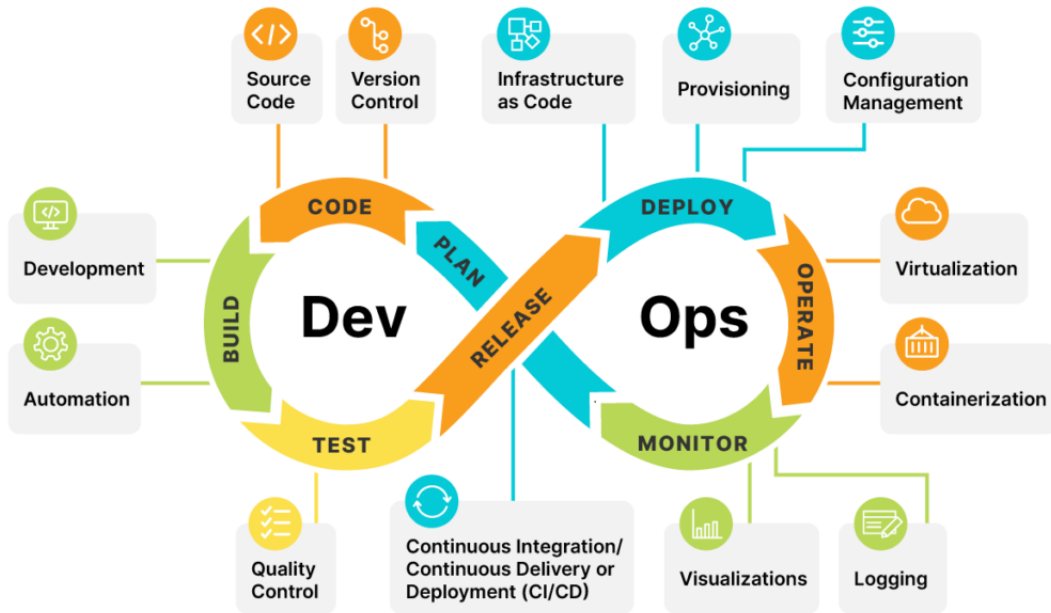
[Continuous Integration and Deployment Best Practices on AWS \(ARC307\) | AWS re:Invent 2013 \(slideshare.net\)](#)

# DevOps (데브옵스)

소프트웨어 개발(development)과 운영(operation)의 약자 Dev와 Ops를 합친 말로, 개발 따로, 운영 따로 하던 관행을 벗어나 보다 더 효율적이고 안정적인 소프트웨어를 제공하기 위한 방법론입니다.

Ask  ChatGPT

데브옵스란?



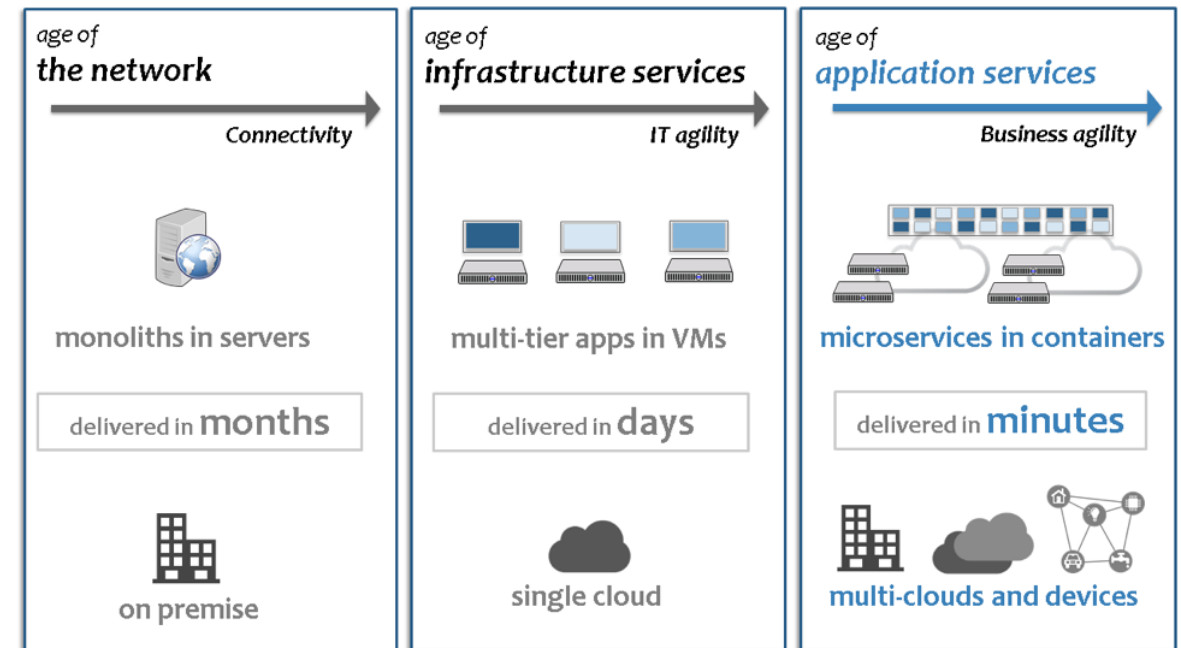
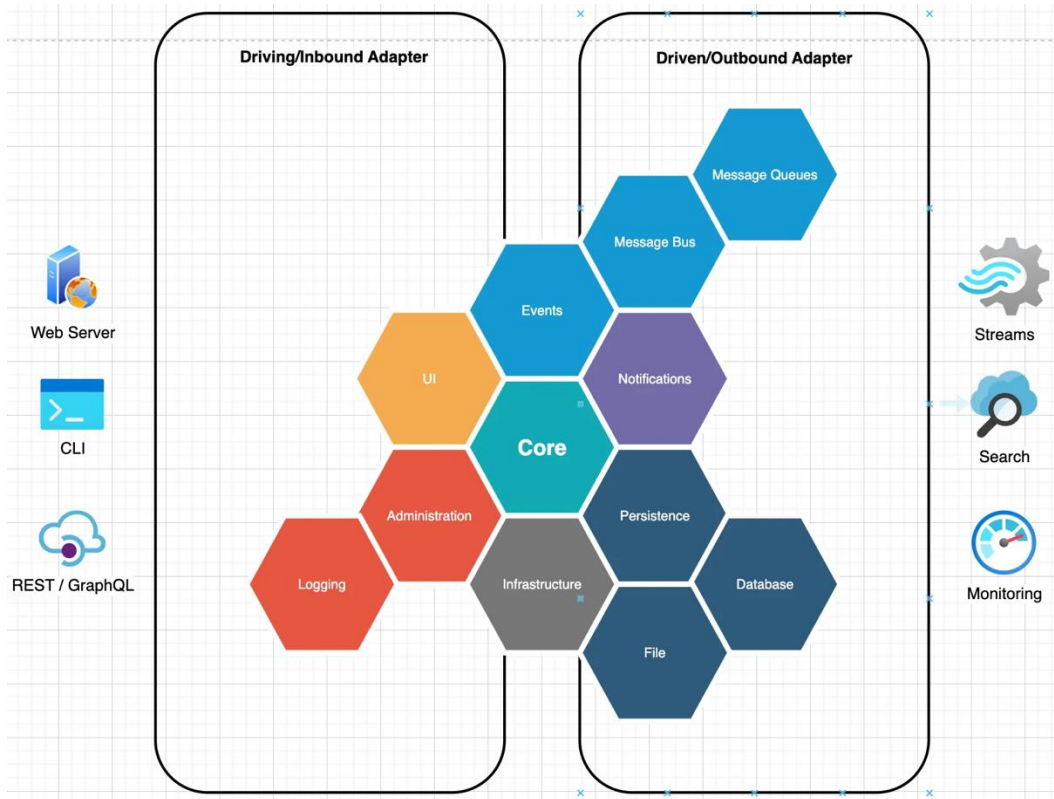
<https://productcoalition.com/12-top-devops-best-practices-for-a-successful-transition-in-2023-b73b54014d0d>

데브옵스(DevOps)는 **개발(Dev)과 운영(Ops) 팀 간의 협업, 커뮤니케이션, 통합을 개선하는 일련의 관행, 방법론 및 문화 철학**입니다. 데브옵스의 주요 목표는 더 효율적이고 안정적인 소프트웨어 제공을 달성하면서 지속적인 개선과 혁신을 장려하는 것입니다. 데브옵스의 주요 특징은 다음과 같습니다:

- 지속적 통합 (CI):
- 지속적 제공 (CD):
- 인프라스트럭처 코드 (IaC):
- 자동화된 배포:
- 모니터링 및 피드백 루프:
- 협업과 커뮤니케이션:
- 공유 책임:

# Cloud Native Application (클라우드 네이티브)

클라우드 네이티브란 클라우드 컴퓨팅 환경을 제대로 활용하여 어플리케이션을 구성하는 방법을 이야기 합니다.



<https://itnext.io/12-factor-methodology-for-cloud-native-microservices-config-and-backing-services-3fd8ef230feb>

<https://thenewstack.io/kubernetes-steering-the-ship-with-cloud-native-management/>



# Full stack developer (풀스택 개발자)

풀스택 개발자란 통상적으로 웹개발에서 프론트엔드(화면 처리)와 백엔드(비즈니스 로직)를 모두 개발할 수 있는 개발자를 지칭하고, 스타트업과 같이 소수 정예로 시작하는 환경에서 더욱 주목을 받는 역할입니다.



**FULL STACK  
DEVELOPMENT  
IN THE CLOUD**



<https://www.linkedin.com/pulse/full-stack-development-cloud-geeks-of-gurukul/>

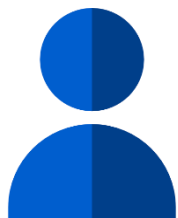
Full stack development in the cloud refers to the practice of creating web applications that use cloud computing resources on top of all software.

Traditionally, full-stack developers are responsible for front-end and back-end development, but with the cloud, **their scope has expanded** to include **infrastructure management and deployment**.

This approach allows developers to use cloud-based services, platforms and tools to streamline **the entire development process**, from coding and testing to deployment and scaling.

# 요약: Why **YOU** need Cloud?

결론적으로, 클라우드 컴퓨팅은 개인으로서나 조직으로서나 꼭 필요한 환경입니다.



## 취업 시장:

- 클라우드 역량은 채용/연봉 협상에서 유리

## 경력 확장:

- 풀스택 개발, 데브옵스, 클라우드 아키텍트로의 성장

## 프로젝트 자유도:

- 인프라 비용 걱정 없이 창의적 프로젝트 실행



## 비용 절감:

- IDC 데이터 기준 2022-2027년 연평균 21% 성장

## 확장성:

- Black Friday처럼 갑작스런 트래픽 증가 자동 대응

## 민첩성:

- 신제품 출시 주기 단축, 빠른 배포/테스트 가능

**Tech professionals could earn a \$10k pay bump if they gained expertise in the following 9 skills:**

- Artificial Intelligence
- Cloud Computing
- C#
- Docker
- Cyber Security
- DevOps
- BASH
- Software as a Service (SaaS)
- VMWare ESXi

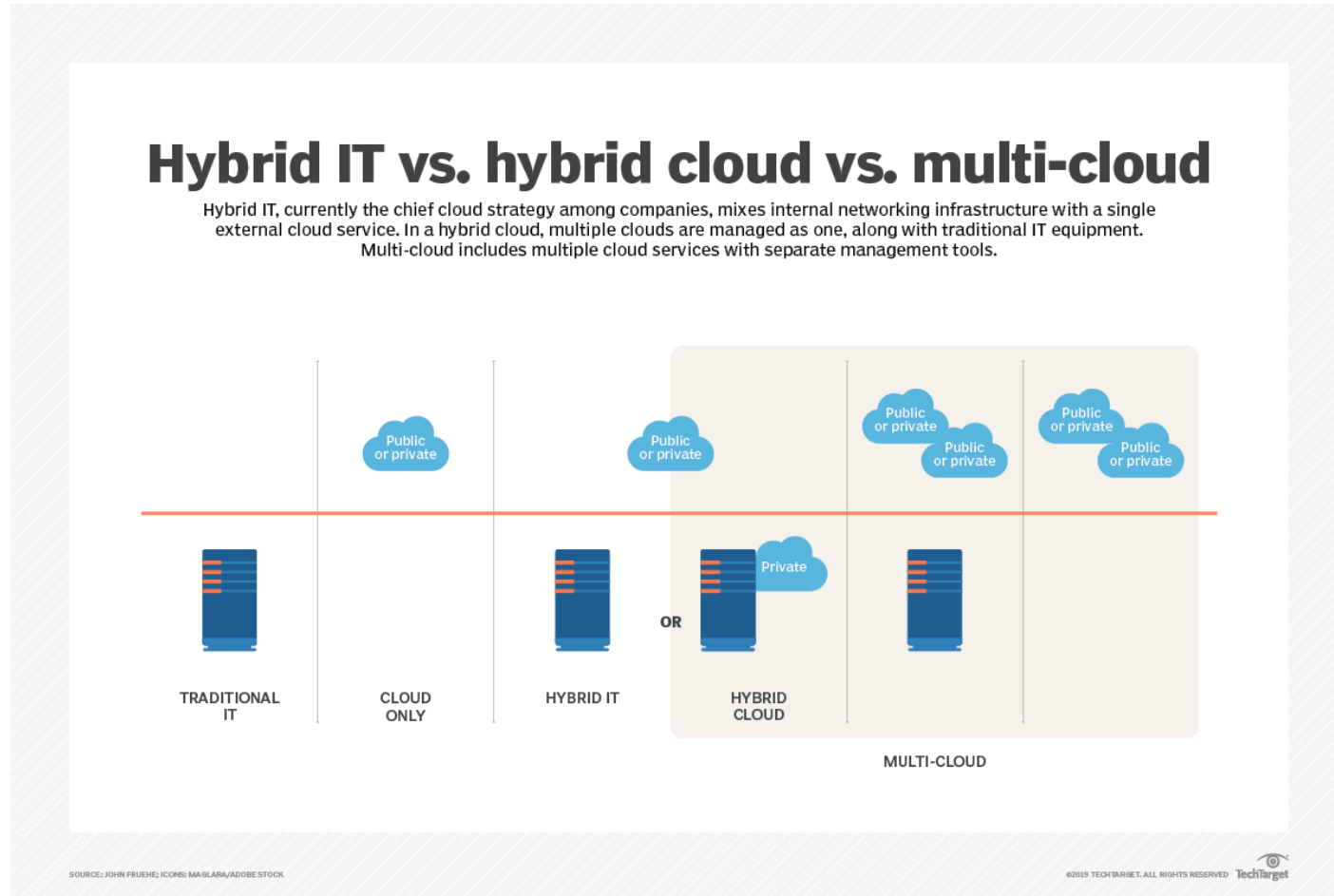
<https://www.dice.com/technologists/ebooks/tech-salary-report/salary-trends.html>



# Cloud Computing 유형

# 클라우드 배포 모델

클라우드 리소스를 배포할 때 Private, Public, Hybrid, Multi Cloud 모델을 적용할 수 있습니다.

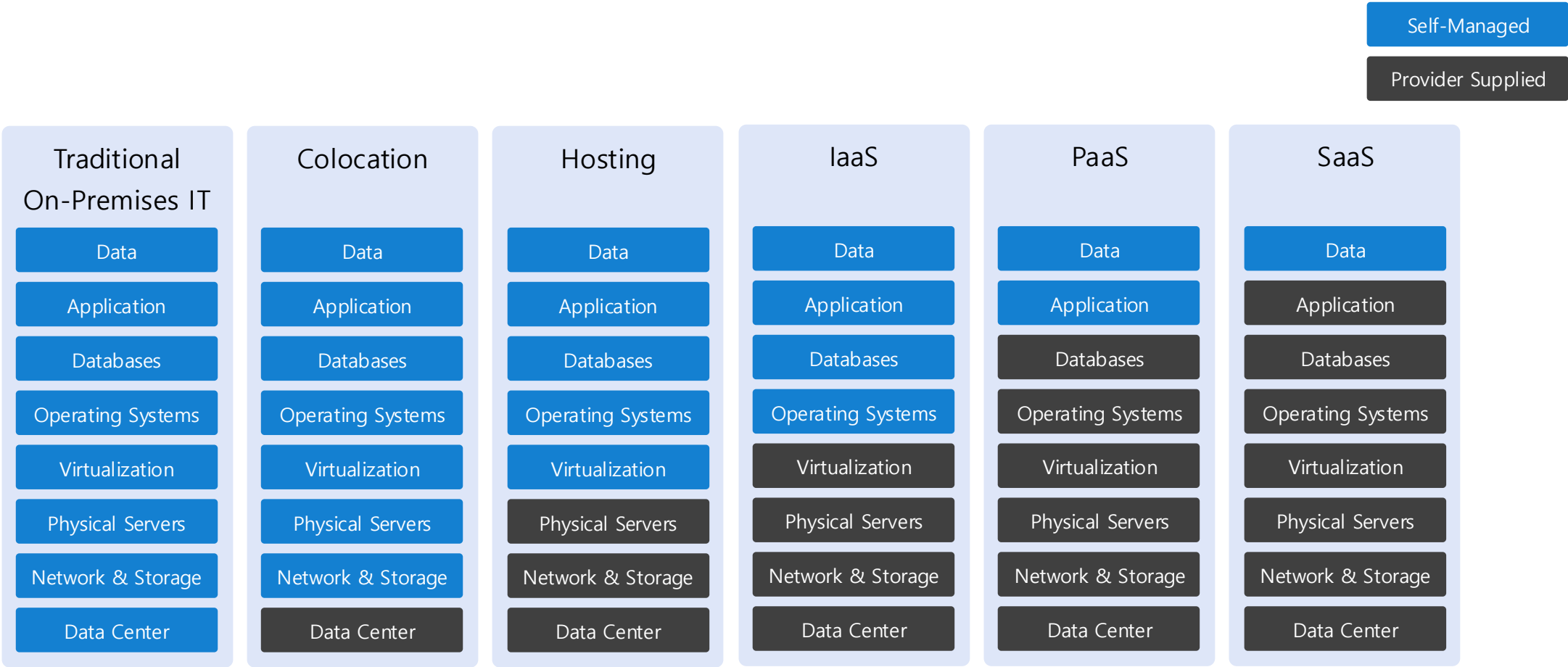


# 클라우드 배포 모델 비교

Feature	Public Cloud	Private Cloud	Hybrid Cloud
소유권	제3자 공급업체 소유	단일 기관 소유	공용 및 사설 클라우드 요소 통합
위치	원격에서 운영 (공급업체 데이터 센터)	온프레미스 또는 외부 호스팅	온프레미스 및 외부 리소스 통합
접근성	인터넷을 통해 접근	사설 내부 네트워크로 접근	배포 모델에 따라 유연
다중 테넌시	다중 테넌트 환경	다른 기관과 리소스 공유 없음	특정 아키텍처에 따라 다름
보안	표준 보안 조치	강화된 보안 제어	요구 사항에 맞게 사용자 정의 가능
제어	제한된 사용자 정의	높은 수준의 제어	배포에 따라 다양
확장성	높은 확장성	사설 인프라 내에서 확장 가능	두 환경 모두 확장 가능
비용	Pay-as-you-go 가격 책정	초기 투자 비용은 높지만 운영 비용은 낮음	사용에 따라 혼합

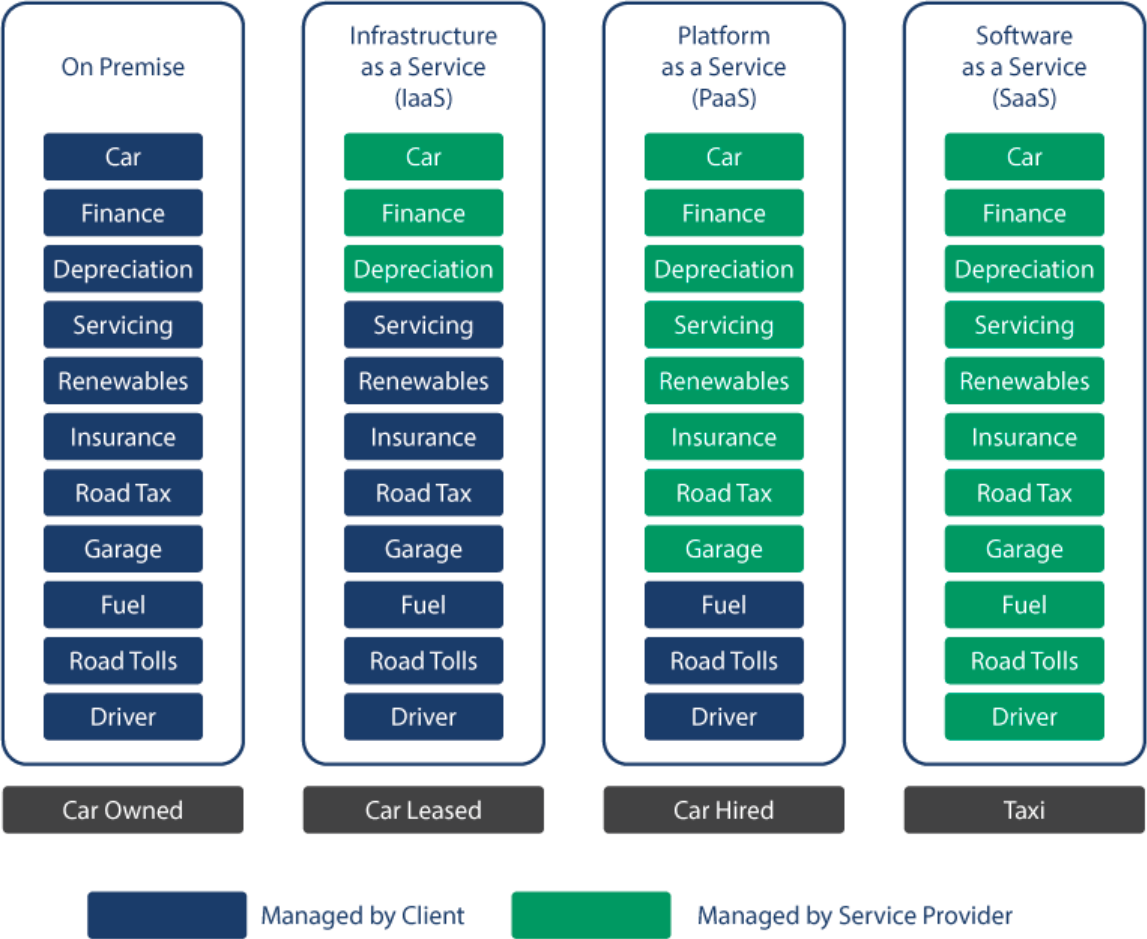
# 클라우드 서비스 유형

클라우드 서비스를 어느 수준까지 사용하는지에 따라 IaaS, PaaS, SaaS로 분류합니다.



Originally created by Gartner 2018

## Car as a Service



# IaaS – Infrastructure as a Service

Ask



ChatGPT

IaaS란?

IaaS는 인터넷을 통해 가상화된 컴퓨팅 리소스를 제공합니다. 이는 **가상 머신, 저장소 및 네트워킹과 같은 기본적인 IT 인프라 구성 요소를 비즈니스에 제공합니다.**

IaaS를 사용하면 조직은 운영 체제, 애플리케이션 및 데이터에 대한 제어권을 가지지만 클라우드 제공업체가 하드웨어 유지보수와 데이터 센터 운영과 같은 기반이 되는 인프라를 관리합니다.

이 모델은 비즈니스가 필요에 따라 리소스를 확장하거나 축소할 수 있도록 하여 유연성과 비용 절감을 제공합니다.

언제 주로 사용하나요? 예를 들면,

- Lift and Shift Migration: 리프트 앤 시프트 이전 (별도 수정없이 그대로 옮기기)
- Setup Test/Development Environment: 테스트 및 개발환경 구성

## 대표적인 IaaS



Compute



Amazon  
EC2



Storage



Amazon S3



Network





# PaaS – Platform as a Service

Ask



ChatGPT

PaaS란?

PaaS는 개발 프레임워크, 데이터베이스, 미들웨어 및 애플리케이션 개발 및 배포에 필요한 기타 도구를 포함합니다. 이 모델은 개발 프로세스를 단순화하여 개발자가 인프라 관련 문제가 아닌 코딩과 애플리케이션 로직에 집중할 수 있도록 합니다. PaaS는 개발 팀 간의 협업을 용이하게 하고 자동 스케일링 및 로드 밸런싱과 같은 기능을 제공합니다.



언제 주로 사용하나요? 예를 들면,

- Setup Development Framework: 개발 프레임워크 구성
- Analytics / Business Intelligence: 분석 / 비즈니스 인텔리전스

<https://www.producthunt.com/products/periodic-table-of-amazon-web-service>  
<https://azureperiodic.data3.com/>

# SaaS – Software as a Service

Ask



ChatGPT

PaaS란?

SaaS는 **구독 기반**으로 인터넷을 통해 소프트웨어 애플리케이션을 제공합니다. SaaS를 사용하면 사용자는 **설치나 로컬 인프라 없이** 웹 브라우저에서 소프트웨어 애플리케이션을 직접 액세스하고 사용할 수 있습니다.

클라우드 제공업체가 인프라, 미들웨어 및 응용 소프트웨어를 포함한 **전체 소프트웨어 스택을 관리**합니다. 사용자는 일반적으로 사용량에 따라 반복적인 요금을 지불합니다.

SaaS는 **업데이트와 유지보수를 제공업체가 처리**하므로 편의성을 제공하며 여러 장치와 위치에서 애플리케이션에 액세스할 수 있습니다.

언제 주로 사용하나요? 예를 들면,

- 메일 및 메시징
- 비즈니스 생산성 애플리케이션
- 재무 및 비용 추적



<https://pro.jumpton365.com/@/hexatown.com/PTO365>

# 클라우드 서비스 모델 비교

Feature	IaaS	PaaS	SaaS
개념	가상화된 하드웨어 지원 제공	애플리케이션 개발 플랫폼 제공	완성된 소프트웨어 제공
관리 범위 (사용자 책임)	OS, 미들웨어, 런타임, 데이터, 앱	데이터, 애플리케이션	없음 (접근 권한 관리 제외)
제어 수준 (Customizing)	높음 (High) 인프라 단위의 세밀한 설정	중간 (Medium) 주어진 프레임워크 내에서 설정	낮음 (Low) 제공된 기능만 사용 가능
기술 난이도	상 (Hard) 네트워크/서버 운영 지식 필요	중 (Medium) 개발 지식만 있으면 충분	하 (Easy) 일반 사용자도 사용 가능
비용 모델	종량제 (Pay-as-you-go) 초/분 단위 사용량 과금	종량제 + 옵션 사용량 또는 티어별 과금	구독형 (Subscription) 월/연 단위 고정 비용 (사용자 당)
주요 활용	레거시 마이그레이션 세밀한 인프라 제어 필요 시	웹/앱 개발 생산성 향상 인프라 관리 없이 배포 시	이메일, 협업도구, CRM 등 비즈니스 도구 즉시 도입 시
대표 예시	AWS EC2, Azure VM, GCP Compute Engine	AWS Lambda, Azure App Service, GCP App Engine	Microsoft 365, Salesforce, Slack

# 공동 책임 모델 (Shared responsibility model)

어떤 경우에도 데이터는  
나의 책임입니다

