<https://www.comworld.co.kr/news/articleView.html?idxno=51285>

<https://www.samsungsds.com/kr/cloud-glossary/paas.html>

<https://www.digitalmarket.kr/web/service/detail.do?s=SAS-4-09-11352>

**PaaS란?**

Platform as a Service(플랫폼 기반 서비스)의 약자로 클라우드 컴퓨팅 서비스의 한 종류로

애플리케이션을 구축, 실행 및 관리하는 데 필요하는 플랫폼을 제공하는 서비스를 의미한다.

개발자가 애플리케이션의 코드 작성에만 집중할 수 있게, 실행 환경, 데이터베이스, 웹서버 등의 애플리케이션 배포에 필요한 기술적인 측면을 제공해준다.

PaaS공급자가 애플리케이션을 실행할 수 있는 플랫폼을 제공하고 관리하기 때문에, 사용자는 서버, 스토리지, 네트워크 등 하드웨어 자원에 대해 걱정할 필요가 없다.

이러한 플랫폼에는 개발에 필요한 라이브러리, 개발 도구, 데이터베이스, 보안 등이 포함될 수 있다. 사용자는 개발 환경에 의존하지 않고 개발을 할 수 있게 됨으로써 더 높은 수준의 추상화를 통한 서비스를 개발하고 배포할 수 있게 되고 이를 통해 개발 생산성을 향상하고 비용을 절감할 수 있게 된다.

비유를 통해 PaaS를 설명하겠다.

공연을 기획하는 사람의 경우, 무대에 필요한 시나리오 구성, 무대 배치 등의 직접적인 공연 요소 구성 뿐만 아니라 무대 장소를 선택하고 무대 장식에 필요한 조명 시설, 음향 시설 등도 고려해야 한다. 이 모든 걸 직접 구비한다면 실질적인 무대 공연에 집중하지 못하게 될 것이다. 이때 기획자가 무대 및 공연에 직접적인 관련이 없는 부대 요소를 제공해주는 서비스를 이용하면, 좀 더 무대 공연에 집중적인 투자를 할 수 있다. 이때 기획자가 원하는 무대까지 요청해서 제공해주는 것을 PaaS에 비유할 수 있다.

**PaaS의 장점**

1. 인프라 및 관리 비용을 절감할 수 있도록 도와준다.

서버, 스토리지, 네트워크 등 하드웨어 인프라를 직접 구매하지 않고, PaaS 공급자가 제공하는 인프라를 이용할 수 있다. 또한, 이러한 인프라 시설의 관리도 자동화하여 관리 인력 비용을 절감할 수 있다.

2. 개발 생산성을 높일 수 있다.

개발자가 애플리케이션 코드 작성에만 집중할 수 있도록 개발에 필요한 미들웨어, 데이터 베이스, 개발 도구 등을 쉽게 구성할 수 있도록 지원해준다. 최신 트렌드에 맞는 기술, 플랫폼, 프레임워크, 데이터베이스, API 등을 통합알 수 있도록 지원하여 사용자의 개발 환경에 필요한 맞춤 서비스를 제공해준다.

3. 애플리케이션 확장을 쉽게 할 수 있도록 도와준다.

개발자가 인프라의 종속적인 개발을 하지 않고, 인프라에 추상적인 개발을 할 수 있게 됨으로써, 인프라의 확장 및 축소가 유연해진다. 수평 및 수직 확장을 지원하며, 필요에 따라 자동으로 자원을 할당하고 해제할 수 있다.

4. 애플리케이션 운영을 자동화하고 안정성과 보안성을 보장한다.

백업, 복구, 보안 등의 작업을 자동으로 처리하여 애플리케이션의 안정성을 유지할 수 있다. 하드웨어 장애나 다른 문제가 발생해도 애플리케이션이 계속해서 작동할 수 있도록 보장한다. 보안에 대한 전문적인 지식과 경험을 바탕으로 안전한 플랫폼을 제공할 수 있다.

**PaaS를 이용할 때 고려해야하는 사항**

PaaS 이용 시 개발자가 코드 중심적인 개발에 집중할 수 있다는 장점이 있지만, 그 외의 것을 PaaS업체에게 맡기기 때문에 코드 이외의 문제에 있어 업체 의존도가 높다는 단점이 존재한다.

1. 제한된 제어 수준을 제공한다.

응용 프로그램에 대해 갖는 제어 수준이 제한될 수 있다. 인프라에 대한 세밀한 정보와 제어를 해야하는 개발자에겐 문제가 될 수 있다.

2. 민감한 데이터를 처리하거나 저장할 경우 보안 문제가 발생할 수 있다.

보안 관련 문제를 PaaS 공급자가 담당하기 때문에 적절한 수준의 보안이 제공되는지 사용자의 지속적인 확인이 필요하다.

3. 제한된 커스터마이징을 제공한다.

PaaS 제공 업체는 제한된 범위의 도구와 서비스를 제공할 수 있으며, 개발자가 특정 요구 사항에 맞게 인프라를 커스터마이징할 수 있는 능력을 제한할 수 있다.

4. 비용을 예측하기 어려울 수 있다.

사용량 및 수요가 갑자기 급증하면 서비스 이용 비용을 예측하기 어려울 수 있다.

기업들의 비니지스가 서비스 중심으로 변화함에 따라, 유연한 인프라 관리와 민첩한 소프트웨어 개발 운영을 지원하는 서비스형 플랫폼의 중요성이 커졌다, 특히 최근에는 인공지능/머신러닝 업무를 실행하는 핵심 기술환경으로 컨테이너와 쿠버네티스가 자리잡으며 이를 담는 클라우드 플랫폼 PaaS 시장이 세계적으로 크게 성장할 것으로 보인다.

클라우드 컴퓨팅의 초기 서비스는 클라우드 인프라 구성을 위한 인프라 중심 서비스(IaaS)였다. IaaS는 클라우드 서비스 제공업체가 서버, 스토리지, 네트워크 등 클라우드 인프라를 대신 관리해주는 것으로, 사용자는 인프라만 대여 받고 그 외 운영체제, 애플리케이션 등을 관리해야 한다.

클라우드 성숙도가 높아지면서 단순 인프라 뿐만 아니라 SW 개발 운영환경도 클라우드로 자리 잡게 되었고, 인프라 위의 미들웨어와 데이터베이스 운영체제 등도 서비스 형태로 제공함으로써 애플리케이션의 환경 구성과 관리까지 지원해주는 PaaS가 등장하였다.



PaaS가 다양한 오픈소스 기반 어플리케이션 개발 및 실행도구, DB관리 시스템, 인프라 모니터링 등의 툴들을 제공하기 때문에 클라우드 위에서 애플리케이션을 신속하게 개발하고 업데이트할 수 있다. 이런 클라우드 네이티브를 구현하기 위해서는 인프라를 컨테이너 기반으로 설계하고, 컨테이너를 관리하는 대표 오픈소스 툴인 쿠버네티스가 필요하다.

**\*클라우드 네이티브**

클라우드 환경을 최적화하여 애플리케이션을 개발하고 운영하는 접근 방식을 의미합니다. 클라우드 네이티브는 주로 클라우드 플랫폼에서 원활하게 동작하도록 설계된 시스템과 애플리케이션을 지칭하는 개념

 **마이크로서비스 아키텍처(Microservices Architecture)**

클라우드 네이티브 애플리케이션은 종종 마이크로서비스 아키텍처를 채택합니다. 이는 큰 애플리케이션을 작은 독립적인 서비스로 나누어 각 서비스가 독립적으로 배포되고 확장될 수 있도록 합니다. 각 서비스는 특정한 비즈니스 기능을 수행하고, 독립적으로 개발 및 유지보수가 가능합니다.

 **컨테이너화(Containerization)**

클라우드 네이티브 애플리케이션은 **컨테이너** 기술을 사용하여 각 서비스나 애플리케이션을 독립된 환경에서 실행하도록 만듭니다. **Docker**나 **Kubernetes** 같은 도구를 사용하여 애플리케이션을 패키징하고, 어떤 환경에서도 일관되게 실행할 수 있게 합니다.

 **자동화(Auto-scaling & Continuous Delivery)**

클라우드 네이티브는 클라우드 환경에서 자동으로 확장(스케일링)하고, 지속적으로 배포(컨티뉴어스 딜리버리)하는 기능을 활용합니다. 이를 통해 애플리케이션은 변화하는 트래픽이나 부하에 맞춰 자원을 자동으로 조절할 수 있습니다.

 **유연성과 확장성**

클라우드 네이티브 애플리케이션은 클라우드에서 제공하는 유연한 리소스를 최대한 활용하여, 트래픽에 맞춰 서버를 확장하거나 축소할 수 있습니다. 이를 통해 효율적인 자원 관리와 비용 절감이 가능해집니다.

 **클라우드 서비스의 활용**

클라우드 네이티브 애플리케이션은 클라우드 환경에서 제공되는 다양한 서비스를 활용하여 개발됩니다. 예를 들어, 데이터베이스, 메시징 큐, 저장소, AI 서비스 등 클라우드 제공업체가 제공하는 다양한 도구를 사용하는 방식입니다.

컨테이너는 SW를 이미지화해 라이브러리 바이너리 파일을 담고 애플리케이션을 구동하는 기술이다. 애플리케이션을 개발할 때 이용한 OS와 WAS 등 전체 요소를 컨테이너로 담고 있어, 개발환경과 운영환경의 차이로 인한 문제를 방지할 수 있다.

하지만 운영하는 컨테이너가 지속적으로 늘어날 경우 이를 관리하는 것은 쉽지 않으며, 이러한 컨테이너를 편리하게 통합하고 관리하는 도구가 쿠버네티스이다. 쿠버네티스는 컨테이너화된 애플리케이션들을 여러 서버나 컴퓨터로 구성하고 이를 하나의 서비스로 이용할 수 있는 클러스터링을 제공하며, 이처럼 구성된 클러스터를 관리하며 애플리케이션을 배치하고 스케일링하는 작업들을 수행한다.



최신의 현대적인 PaaS는 마이크로서비스 아키텍쳐(MSA), 지속적 통합/지속적 배포(CI,CD), AI/ML 등의 모듈 서비스를 포함하는 클라우드 네이티브 플랫폼 기술을 제공한다.

컨테이너 기술로 MSA환경을 구현할 수 있게 되었고, (모듈 별 분리 개발 및 배포 가능) 어플리케이션을 모듈화 해 자유로운 분리와 결합이 가능하고 (고객이 원하는 모듈만 선택적으로 이용 가능) 특정 문제가 발생하면, 전체 서비스는 유지한 상태로 문제가 있는 기능만 따로 수정 보완할 수 있고, 이런 유연하고 민첩한 애플리케이션 서비스의 중심에는 PaaS가 존재한다.

국내 PaaS 시장의 전망은 밝은데, 금융 기업들이 본격적으로 클라우드 기반의 데이터 플랫폼, 차세대 애플리케이션, 신규 고객 대응 애플리케이션 등을 구축 및 계획하면서 PaaS시장이 크게 성장할 것으로 보인다.