

클라우드, 클라우드 네이티브, 앱 현대화 한방에 정리하기

맨텍솔루션 | 이진현

manTech
Solution



고가용



자동화



PaaS

(주)맨텍솔루션

(주)맨텍솔루션은 지난 34년 동안 풍부한 경험과 기술력을 바탕으로 시스템 이중화, 재해 복구 시스템 구축 및 운영 자동화 솔루션, 애플리케이션 딜리버리 최적화를 위한 PaaS를 개발하여 공급하는 SW 전문기업입니다. 맨텍솔루션은 '사람과 기술'이라는 창립 이념 아래 사람을 위한 기술의 발전과 혁신을 위해 노력하고 있습니다.

회 사 명	(주)맨텍솔루션
대 표 자	김 형 일 / 김 윤 원
설 립 일	1989년 9월 29일
대표전화	02) 2136-6900
임직원수	130 명 (2023년 8월 기준)
홈페이지	http://www.mantech.co.kr , http://accordions.co.kr

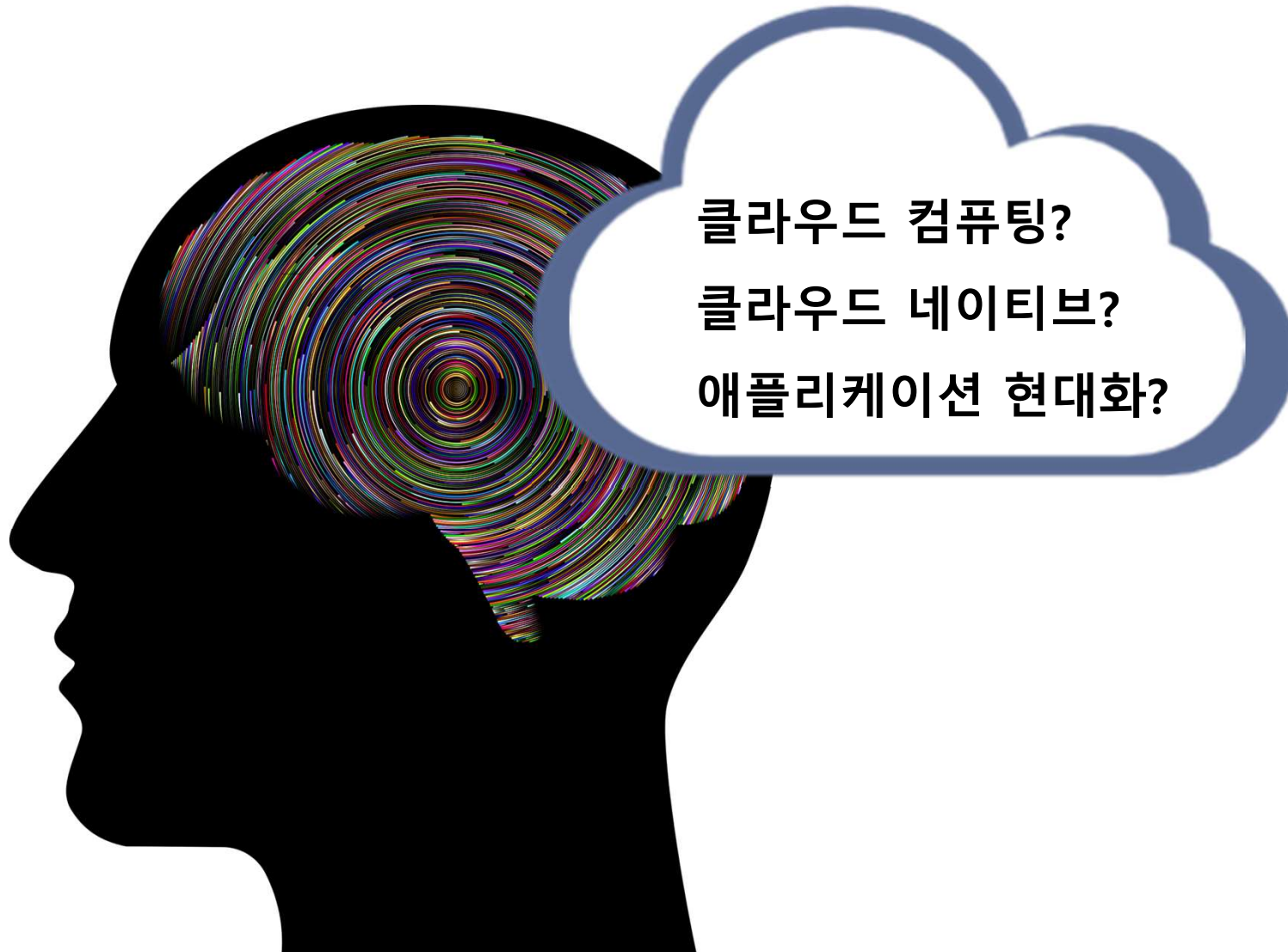
Contents

1. 클라우드 관련 개념 정리
2. 클라우드 네이티브 아키텍처링의 특징
3. 클라우드 네이티브를 통한 문제 해결

1

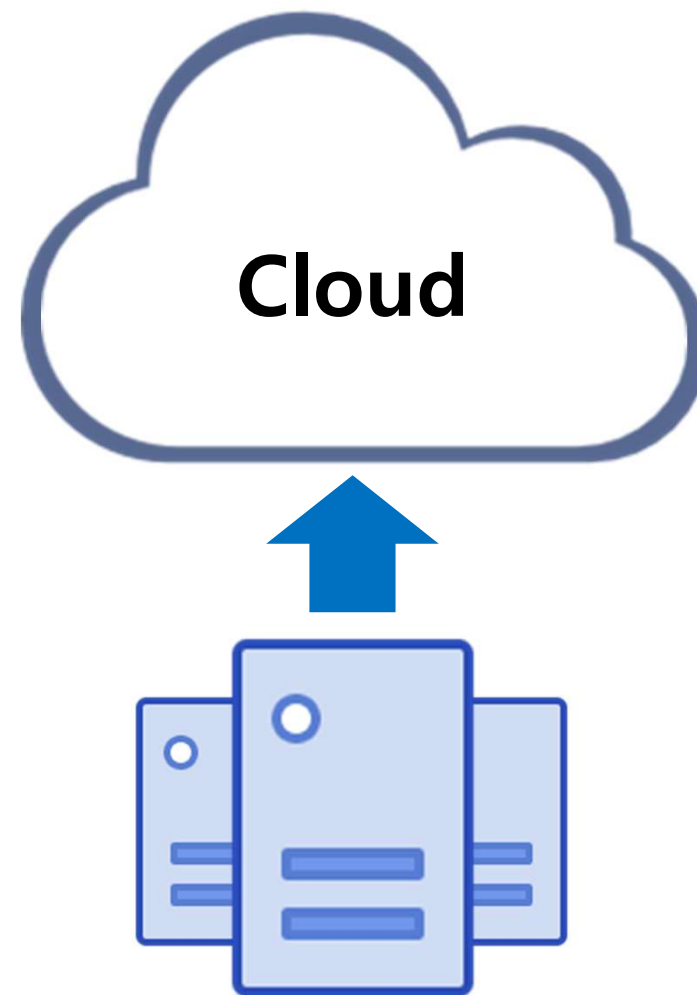
“클라우드 관련 개념정리

복잡 다난한 개념들



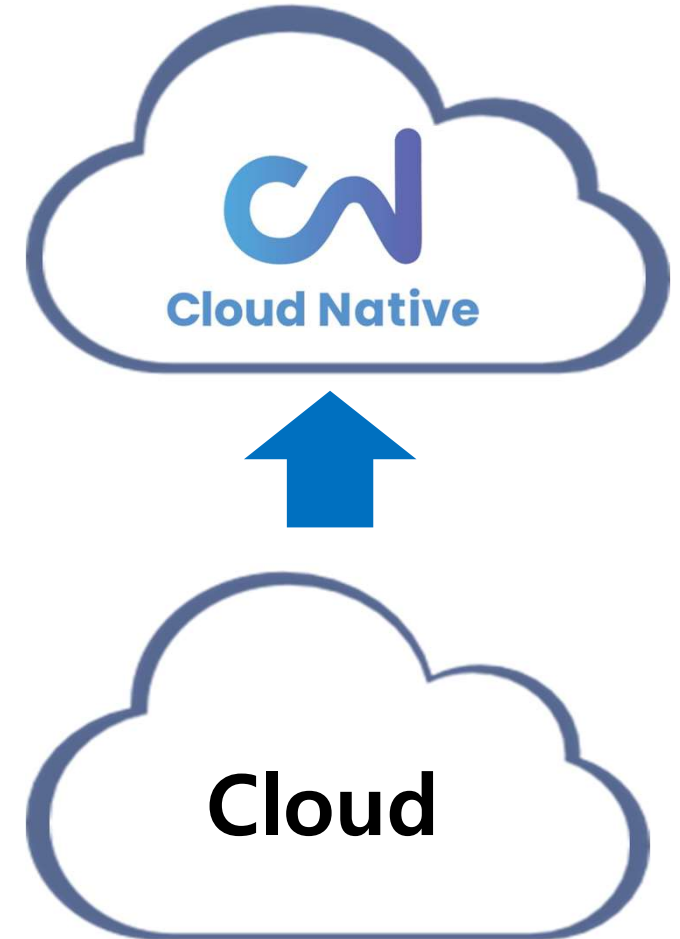
업계에서 정의하는 클라우드 컴퓨팅이란?

- 애플리케이션의 구동에 필요한 **자원**, 즉 서버, 스토리지, 네트워크 등의 **리소스**들을 필요한 만큼 자동으로 할당 받는 컴퓨팅 환경
- 하드웨어 **리소스**들의 가상화와 이에 대한 프로비저닝에 주안점을 두고 있음
- 주로 IaaS 영역에 해당



업계에서 정의하는 클라우드 네이티브란?

- 기존 앱을 추가적인 개발이나 수정없이 클라우드 아키텍처에서 잘 구동될 수 있도록 개선하는 활동
- 앱의 아키텍처, 코드, 데이터, 보안, 개발 및 운영 방식을 개선하는 것을 포함
- 주로 PaaS 영역에 해당



애플리케이션의 현대화란?

- 기존의 애플리케이션을 아키텍처 변화없이 최신의 플랫폼으로 점진적으로 전환시키는 것
- 현재 단계의 애플리케이션 현대화는 클라우드 네이티브로의 전환을 의미



개방성



표준화



민첩성



자동화



인프라의 종속성이 없으며

요구사항에 즉시 배포, 확장

및 자원 회수가 가능한 환경

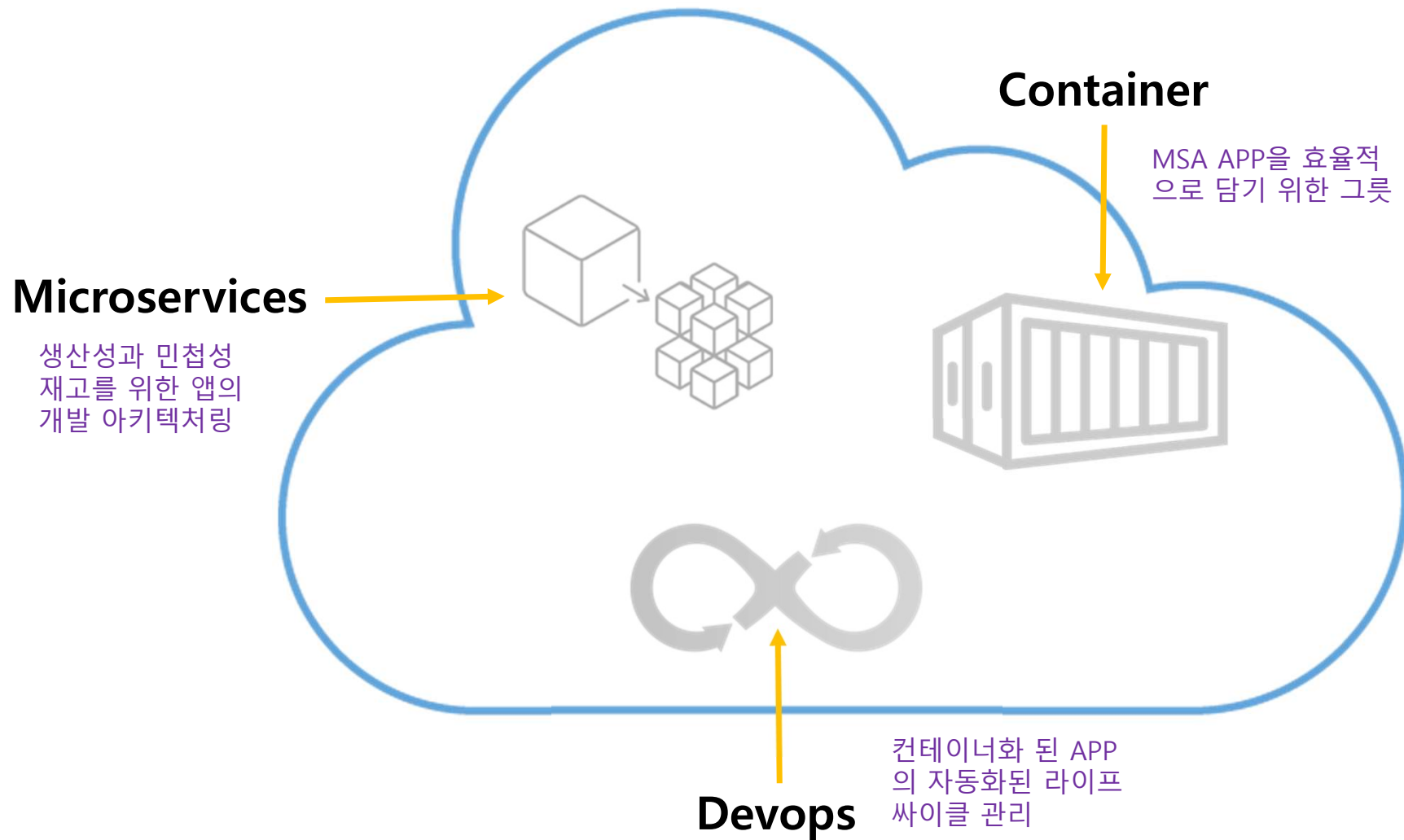
always on.

클라우드 환경으로의 전환을 위한 전략들

구분	유형	애플리케이션 변화			비고	주요기술
		인프라	소스코드	아키텍처		
클라우드 인프라 전환	Re-Host	변경	유지	유지	<ul style="list-style-type: none"> 서버, 네트워크, 스토리지 등의 인프라 요소들만 클라우드로 이전 	<ul style="list-style-type: none"> IaaS
애플리케이션 현대화 (클라우드 네이티브)	Re-Factor	변경	변경	유지	<ul style="list-style-type: none"> 애플리케이션 아키텍처 유지 하되 클라우드 네이티브 플랫폼으로 전환 	<ul style="list-style-type: none"> 컨테이너 DevOps
	Re-Architect	변경	변경	변경	<ul style="list-style-type: none"> 클라우드 네이티브 아키텍처로 애플리케이션 재개발 	<ul style="list-style-type: none"> MSA

2 “클라우드 네이티브 아키텍처의 특징

클라우드 네이티브 아키텍처의 3대 요소



No Vendor Lock In

- Open APP



- Open DBMS



- Open Cloud



- Linux kernel



- x86 Server



표준화

“ 여러 이기종 플랫폼과 클라우드 환경에 호환가능한
표준화된 플랫폼과 운영 환경 체계 마련



민첩성과 가용성



속도

- 비즈니스의 빠른 변화로 인한 애플리케이션의 잦은 배포 / 신속 배포 / 신속 롤백 필요



무중단

- 투명한 모니터링 / 장애 영향의 최소화 / 자동 복구
- 애플리케이션 배포,롤백 시 중단 없는 서비스 요구

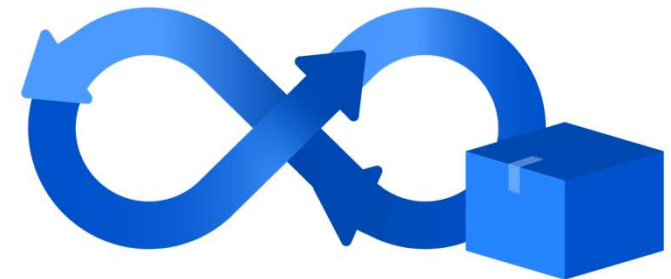


확장성

- 요구사항에 따른 수평적 확장
- 멀티 하이브리드 클라우드에 대한 투명한 확장 요구

자동화

- ✓ DevOps를 통한 형상관리→빌드→배포 과정
- ✓ 빌드된 애플리케이션의 컨테이너화
- ✓ 멀티/하이브리드 클라우드간 배포
- ✓ 애플리케이션의 수평 확장
- ✓ 장애감시와 복구
- ✓ 네트워킹



3

“클라우드 네이티브를 통한 문제 해결

클라우드 네이티브 아키텍처를 통한 고민 사항들의 해결

- ✓ 서로 다른 환경에서 애플리케이션을 안정적으로 실행할 묘안은?
- ✓ 애플리케이션 수정 사항을 다수의 서버에 쉽게 배포할 방법은?
- ✓ 자동화된 서비스 증설 방법은?
- ✓ MSA를 효율적으로 운영관리 하기 위한 인프라의 아키텍처 방안은?
- ✓ 하이브리드 클라우드 환경에서 호환성을 유지할 방안은?
- ✓ 폭증하는 웹 트래픽과 서비스 지연에 효과적으로 대처하기
- ✓ 다수의 컨테이너 앱을 단일의 포인트에서 관리하기

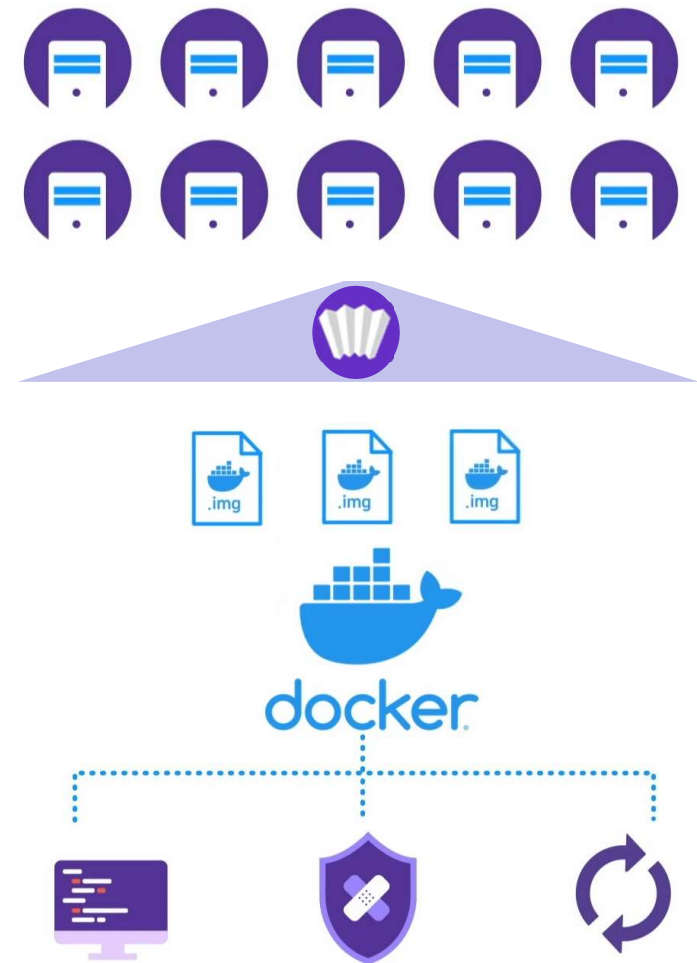
서로 다른 환경에서 애플리케이션을 안정적으로 실행하기

- ✓ 컨테이너에 앱이 실행될 바이너리, 라이브러리, 구성 파일 등을 같이 패키징
- ✓ 호스트 OS의 환경설정과 무관하게 앱이 안정적으로 실행



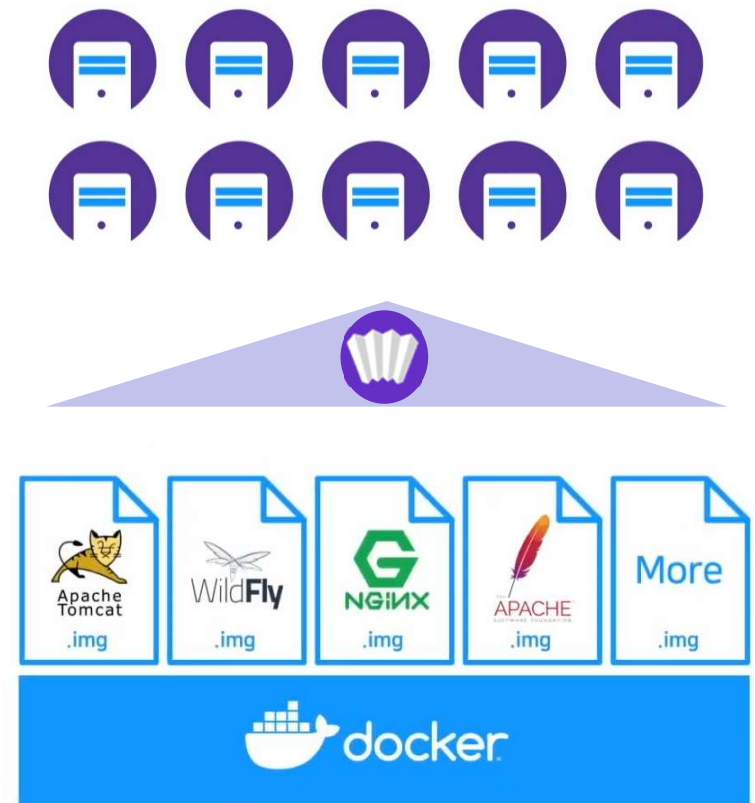
애플리케이션 수정 사항을 다수의 서버에 쉽게 배포하기

- ✓ 앱의 소스 수정 및 보안 패치를 자동으로 컨테이너 이미지화
- ✓ 아코디언을 통해 다수의 서버에 일괄 배포
- ✓ 오류 발생 시 원 클릭 롤백



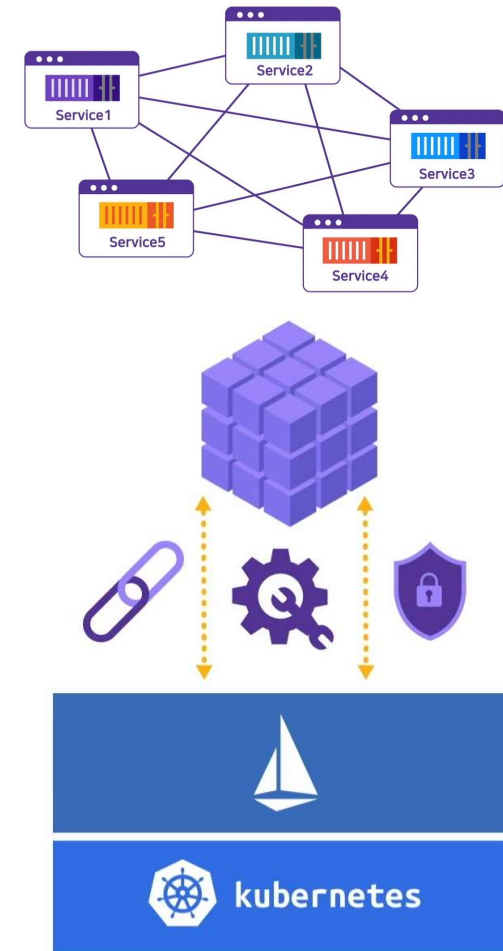
자동화된 서비스 증설

- ✓ 여러 대의 서버에 기존 FTP를 통한 동일한 서버의 서비스 증설 시 환경의 일관성 유지가 어려움
- ✓ 프로비저닝 툴의 경우 스크립트 개발의 어려움 존재
- ✓ 도커 이미지화를 통해 아코디언을 통해 자동 배포와 증설



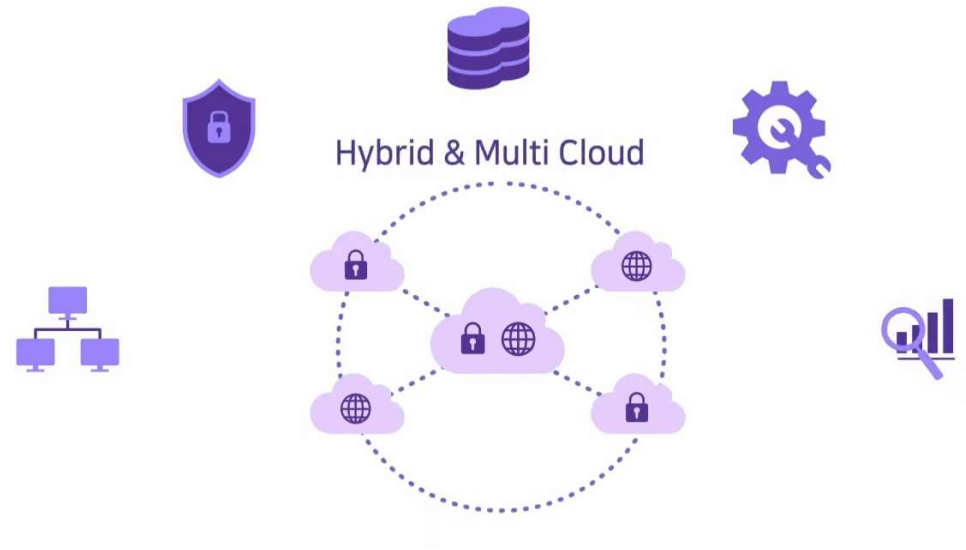
MSA의 효율적인 운영

- ✓ 단일의 앱을 다수의 독립적으로 분리된 서비스 형태로 운영 시 각 서비스 별 유연한 배포, 확장, 장애 처리, 표준화된 인프라가 필요
- ✓ Istio를 통한 MSA간 서비스 연결, 인증, 보안, 모니터링 등의 기능을 제공



하이브리드 환경에서 클라우드 간 호환성 유지하기

- ✓ 이기종 클라우드 간 네트워크, 보안, 스토리지, 환경설정, 모니터링 등을 일관되게 유지하기가 쉽지 않음
- ✓ 아코디언을 통해 표준 컨테이너를 이기종 클라우드 간 배포 및 확장 가능



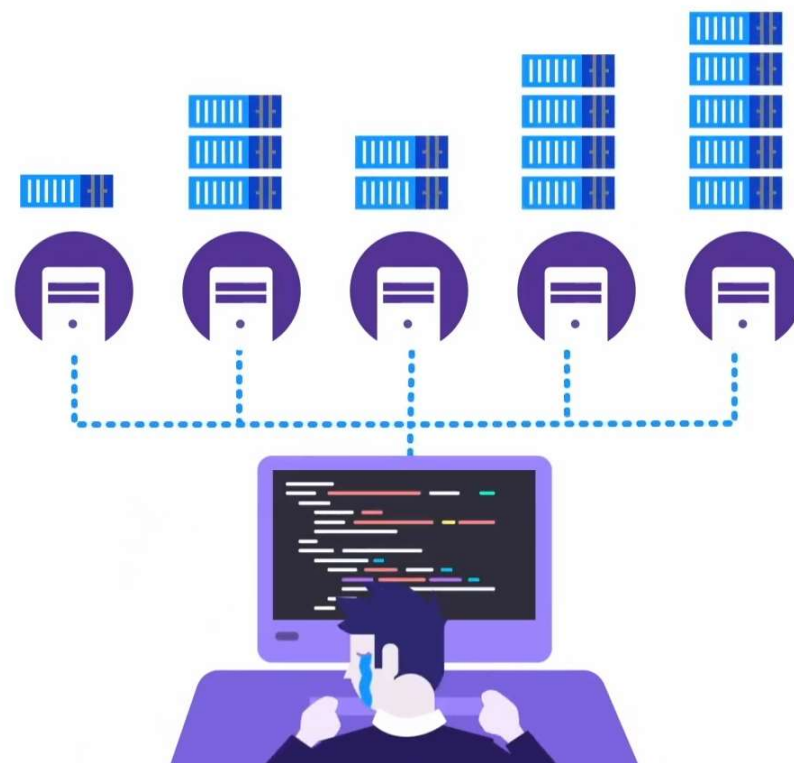
폭증하는 웹 트래픽에 대한 대처

- ✓ 모바일 기기의 증가로 트래픽의 예측이 쉽지 않으며, 외부 서비스는 항상 과부하 및 섯다운의 위험성 내포
- ✓ 아코디언을 통해 클라우드 간 자동 확장 및 축소를 통해 유연한 대처가 가능



수많은 컨테이너 앱을 단일의 포인트에서 관리하기

- ✓ 증가되는 컨테이너의 자동화된 관리를 위한 쿠버네티스는 대세
- ✓ 하지만 추가적으로 보안, 영구 볼륨, 서비스, 환경맵, RBAC, Ingress, 레파지토리 등의 복잡한 환경 구성이 필요
- ✓ 아코디언을 통해 원 클릭으로 자동화된 설정
- ✓ 멀티 클러스터의 단일 관리



정리를 하자면...

- 클라우드 네이티브는 개방성, 표준화, 민첩성, 자동화의 여건 마련이 핵심이다.
- 애플리케이션의 개발, 운영을 생산성 극대화를 위한 현대화 전략을 수립해야 한다.
- 컨테이너 플랫폼을 활용하면 인프라의 종속성에서 해방될 수 있다.
- 모든 시작점은 애플리케이션의 컨테이너화에서 출발한다.



감사합니다.

K-PaaS Summit 2023