

GDSC DevFest KWU - 1st Session

# Cloud Native, 그래서 그게 뭔데?

송승호 (ASC / K-NET)



# Speaker

## 송승호 (Seungho Song | Ricky)

- 소프트웨어학부 19
- CNS LAB (김진우 교수님 연구실) 학부연구생
- K-NET Member (2022~2023)
- Triple G (소프트웨어학부 학술소모임) 회장 (2022)
- AWS Student Club KWU 1st Captain
- AUSG 7th member
- Interested in Cloud, DevOps, and Server

### Contact Me

- E-mail : songseungho9258@gmail.com
- Github : @SeungHo0422
- Linkedin : [www.linkedin.com/in/rickysong0422](https://www.linkedin.com/in/rickysong0422)



# Cloud? Native? It's Cloud Native!

단어를 하나씩 뜯어봅시다.



x



x



x



x

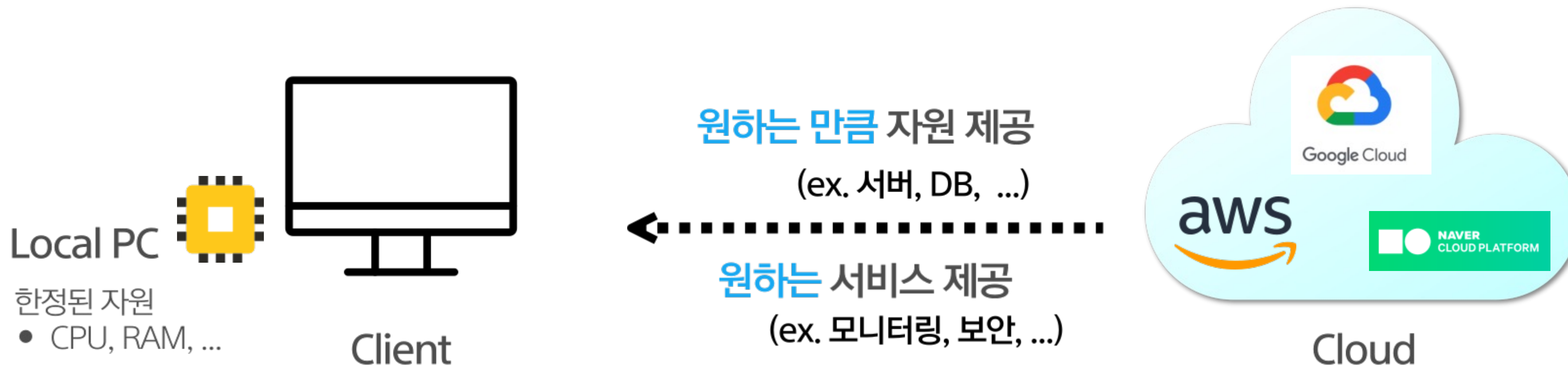




## Cloud Native란?

## Cloud

클라우드(Cloud)란, 광대한 네트워크를 통하여 접근할 수 있는 **가상화된 서버**와 서버에서 작동하는 **프로그램과 데이터베이스**를 제공하는 **IT 환경**을 의미한다.



서비스 간 강한 의존성

- 유지보수 비용 UP

Infra부터 Application까지.

인터넷만 접속할 수 있으면, **컴퓨팅 서비스를 원하는 만큼 대여할 수 있는 기술**이라고 비유할 수 있다.

## References

[1] <https://www.samsungsds.com/kr/cloud-glossary/what-is-cloud.html>

Cloud Native란?

# Native?

native ★★ +

1. 형용사 (사람이) 태어난 곳의 (→native speaker)
2. 형용사 (특정한 곳의[에]) 토박이의[오래 산]
3. 명사 ... 태생[출신]인 사람
4. 명사 토착민, 현지인 (=local)



## 기반 (Based)

## References

[1] <https://en.dict.naver.com/#/search?query=native>

Cloud Native란?

# Cloud Native

클라우드의 규모와 크기 조정 및 성능에 최적화되어 있는  
애플리케이션을 개발하기 위한 접근 방식과 기술

## References

[1] <https://aws.amazon.com/ko/what-is/cloud-native/>

# Cloud Native 101

클라우드 네이티브의 모든 것.



x



x



x



x



## Cloud Native의 특징

# Cloud Native

클라우드의 규모와 크기 조정 및 성능에 최적화되어 있는  
애플리케이션을 개발하기 위한 접근 방식과 기술

### 유연성

다양한 클라우드 환경에서  
수정없이 마이그레이션하거나 배포 가능

01

02

### 개발 생산성

지속적인 통합과 지속적인 배포,  
자동화된 배포 운영 지원

Cloud Native

### 운영 효율성

자동화, 셀프 서비스, 원격 측정, 분석 등  
필요한 기능 확장성 용이함에 따른 운영 효율성 향상

03

04

### 개발 기간 단축

신속한 업데이트 및 기능 추가 개선을 통한  
서비스 신속성 제공 가능

#### References

[1] <https://www.youtube.com/watch?v=jAaOEiCO-xU&t=171s>



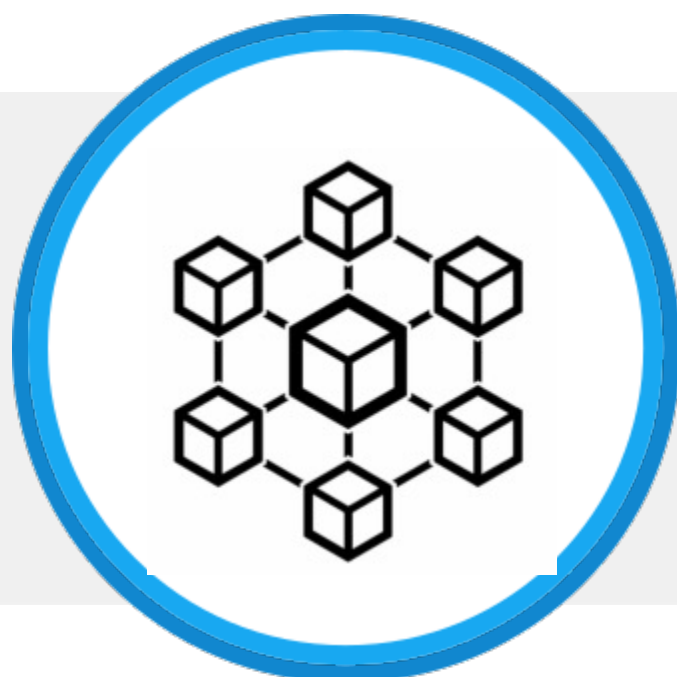
## Cloud Native의 특징

# Cloud Native의 핵심 4요소



### CI / CD

빠르고 안정적인  
Application 배포 관리 자동화



### MicroService

서비스 안정성, 스케일링 용이성 개선,  
독립적 배포 가능



### Container & Orchestration

IT 이식성과 유연성 확보



### DevOps

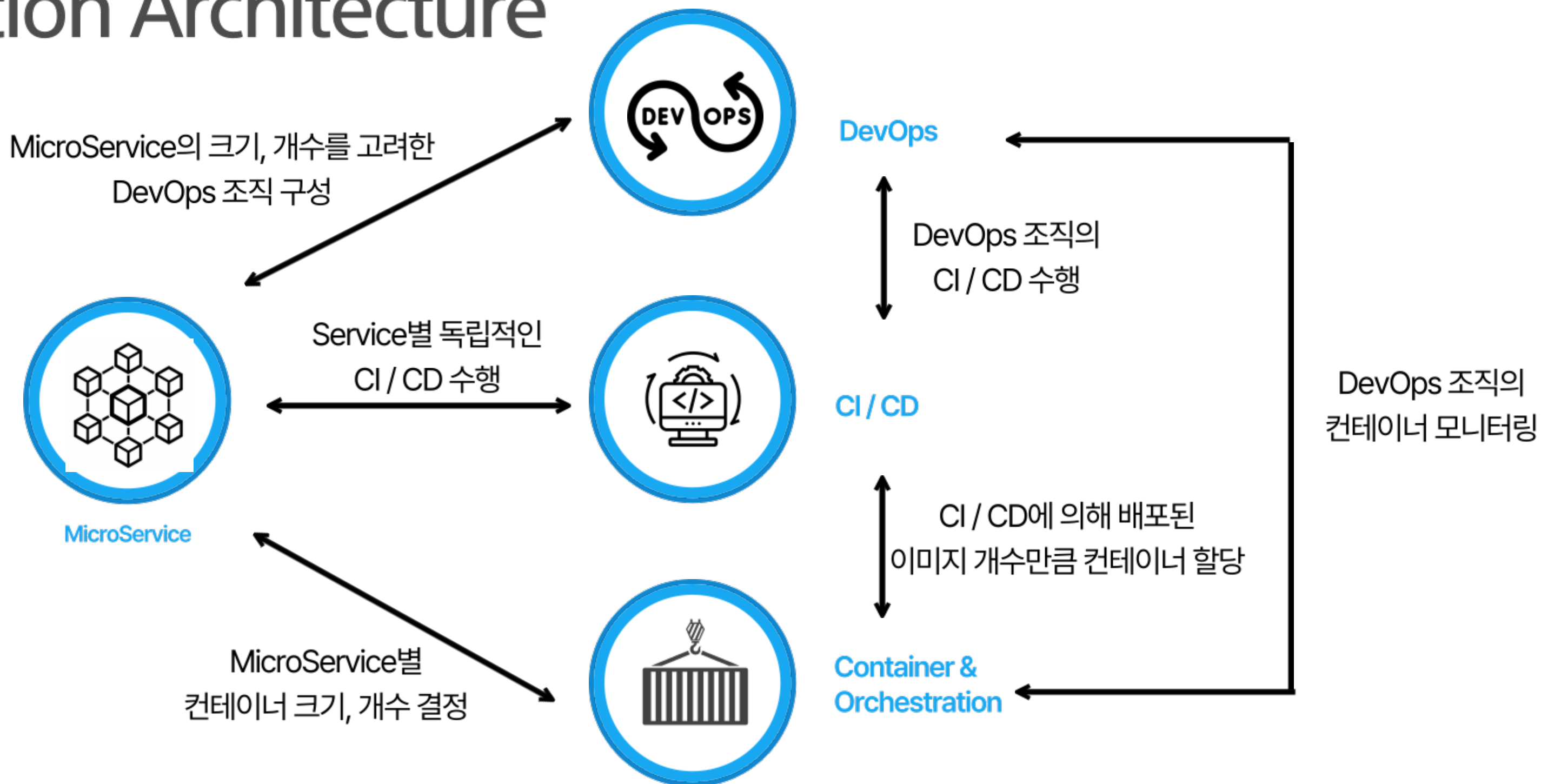
Application 개발, 운영 통합화로  
서비스 개선 속도 증가

#### References

[1] <https://www.youtube.com/watch?v=jAaOEiCO-xU&t=171s>

## Cloud Native의 특징

# Application Architecture



### References

[1] <https://www.youtube.com/watch?v=jAaOEiCO-xU&t=171s>

CNCF Projects

# CNCF 소개

Cloud Native Computing Foundation (CNCF)는 Linux Foundation으로부터 파생된 비영리 재단으로, "클라우드 네이티브 환경이 어디에나 존재하게끔 만드는 것"을 목표로 다양한 활동을 진행하고 있다.



**CLOUD NATIVE**  
**COMPUTING FOUNDATION**

## References

[1] <https://www.cncf.io/about/who-we-are/>

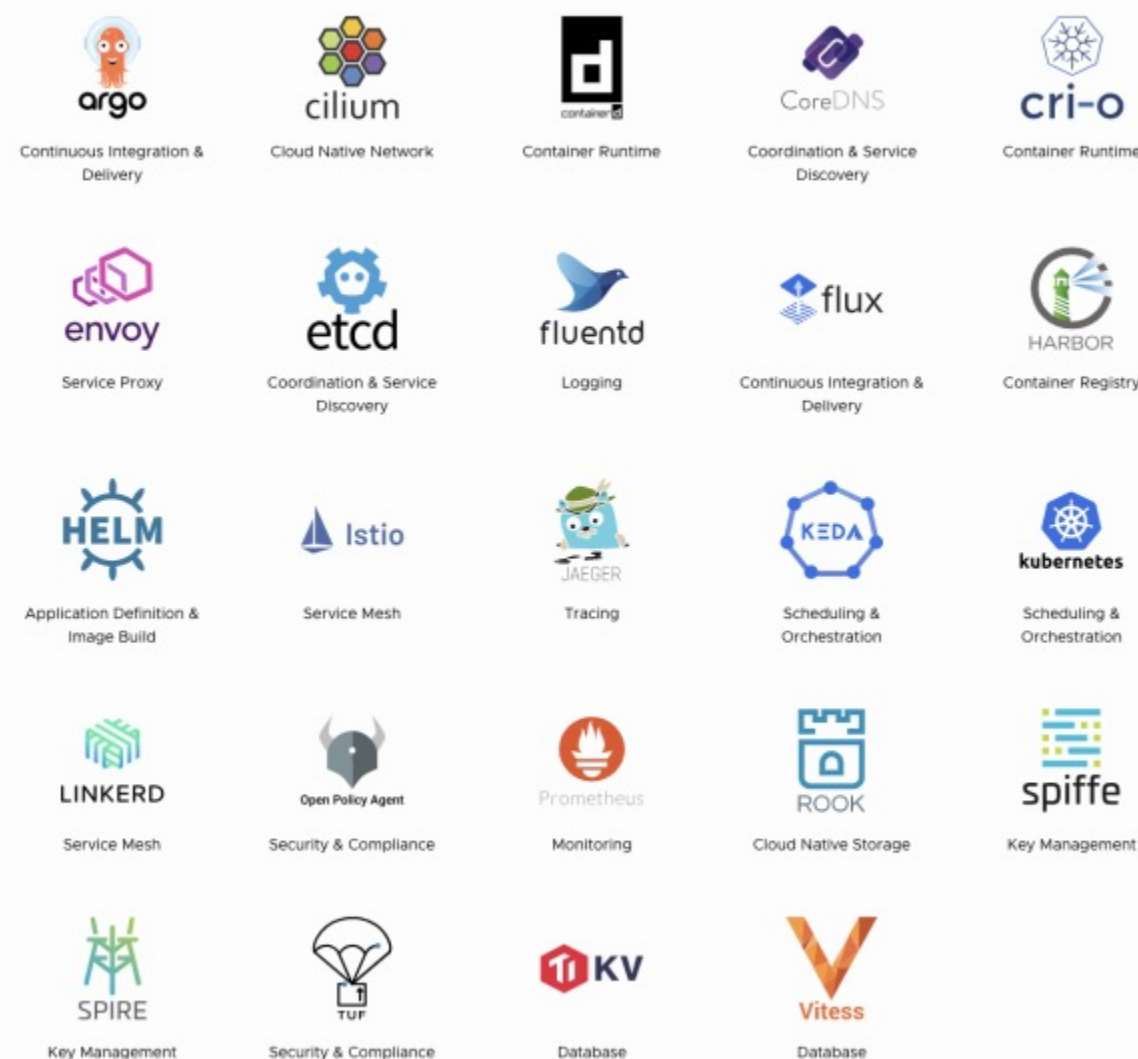
## CNCf Projects

# Cloud Native Solution Projects

CNCf는 클라우드 환경에서 최적의 성능과 비즈니스 모델을 만들 수 있도록 도와주는 오픈소스 솔루션들을 지속적으로 후원하고 있습니다.

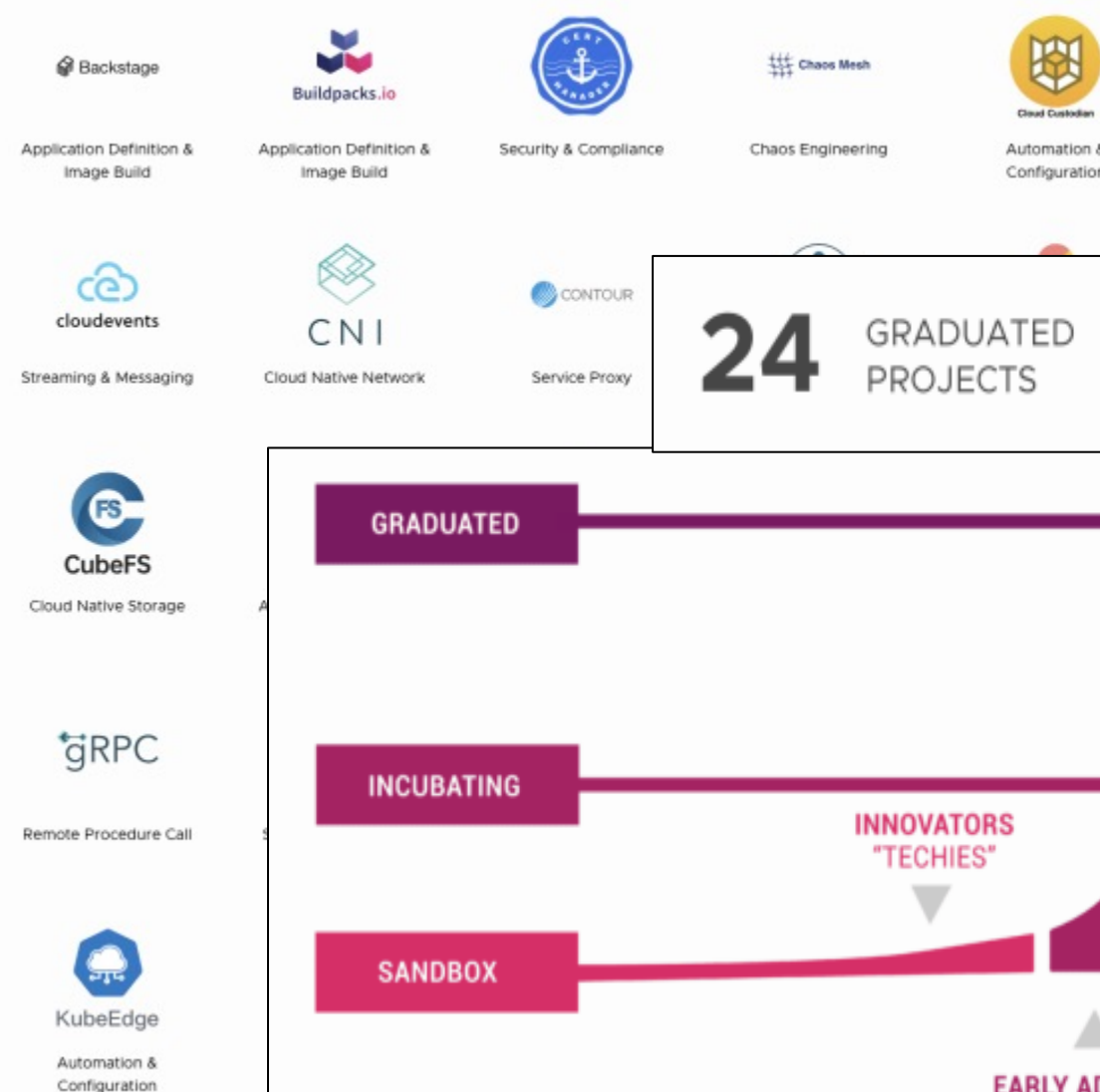
## Our Graduated Projects

VIEW ON CNCf LANDSCAPE



## Our Incubating Projects

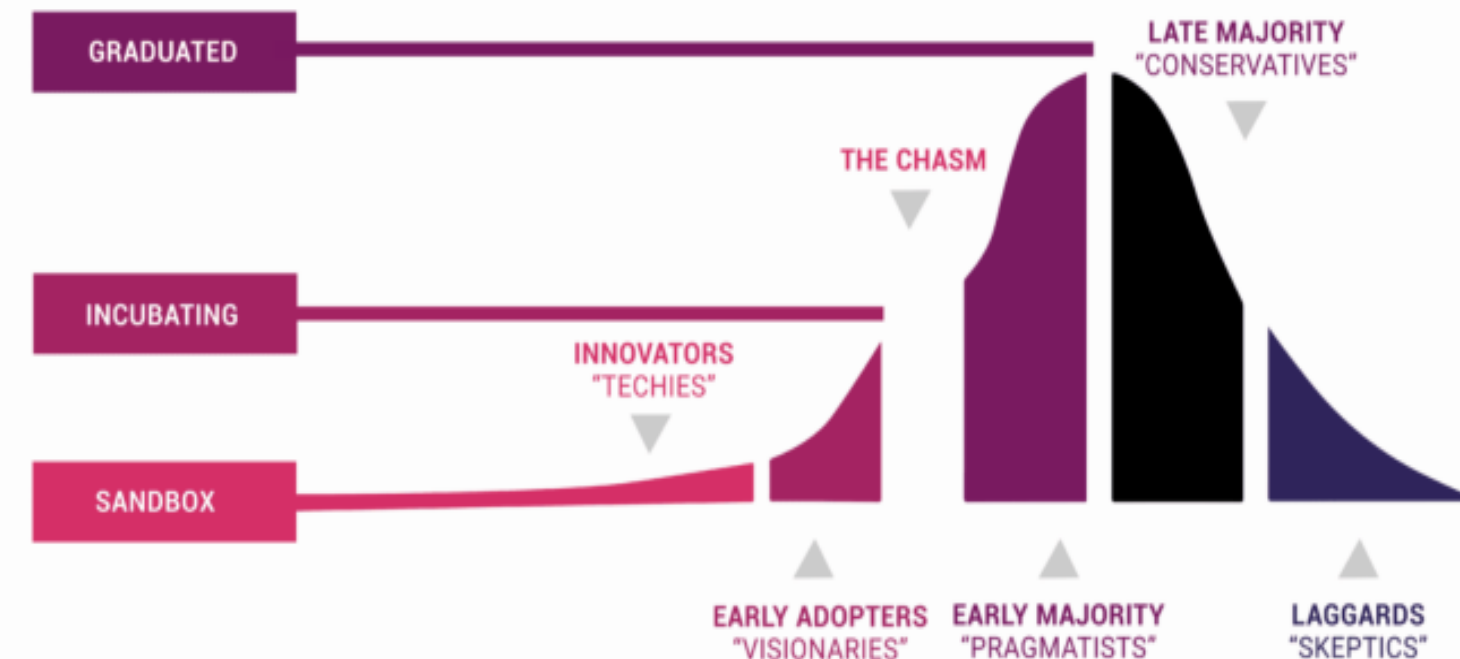
VIEW ON CNCf LANDSCAPE



24 GRADUATED PROJECTS

36 INCUBATING PROJECTS

108 SANDBOX PROJECTS



## References

[1] <https://www.cncf.io>



## CNCF Projects

# Cloud Native Solution Projects

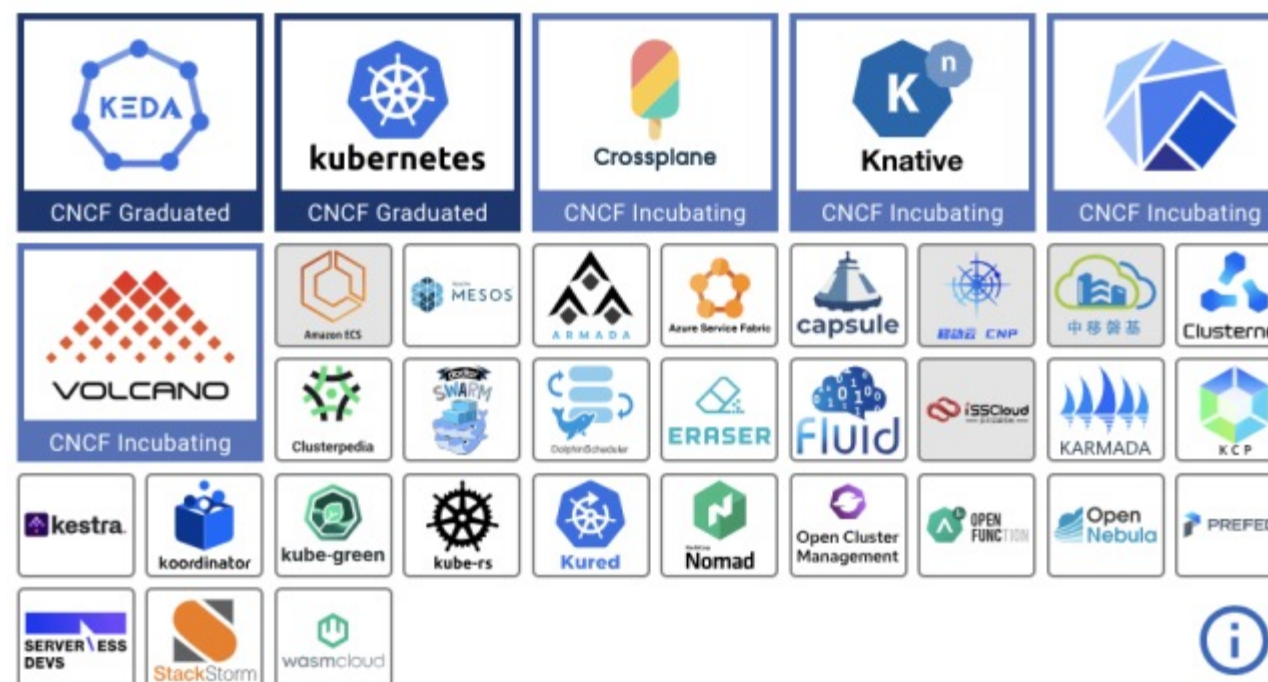
CNCF는 클라우드 환경에서 최적의 성능과 비즈니스 모델을 만들 수 있도록 도와주는 오픈소스 솔루션들을 지속적으로 후원하고 있습니다.



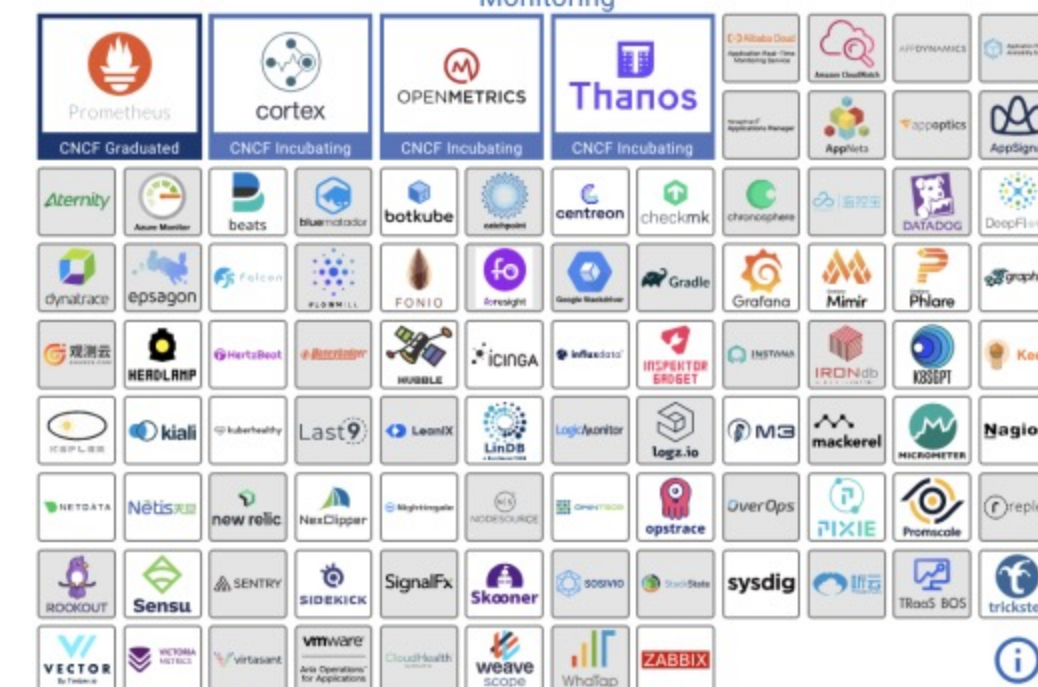
CI / CD



MicroService



Container &  
Orchestration



DevOps

Monitoring

References

[1] <https://www.cncf.io>



# Example.

클라우드 네이티브를 적용한 대표적인 성공 사례.



x



x



x



x



[illegible]

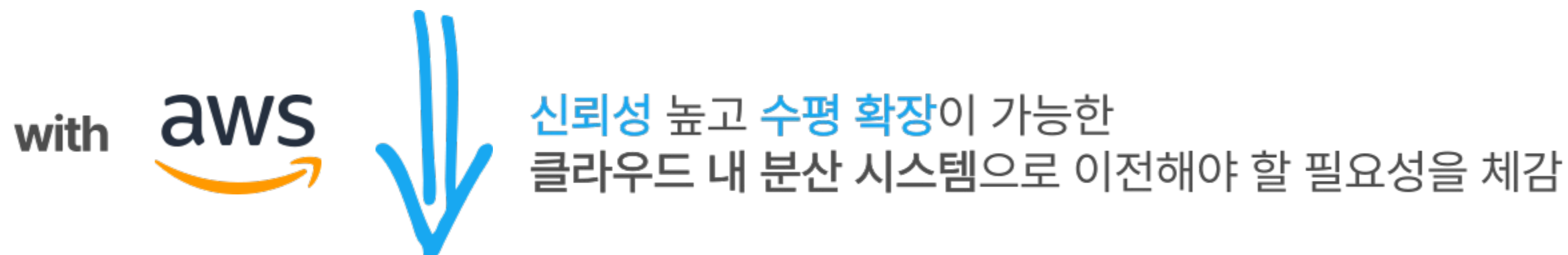
Example of Cloud Native Application

# Netflix의 Cloud-Native 적용 사례

## Before

2008년 : 자체 DataBase 손상 -> 3일간 DVD 배송 지연

- 분산 데이터베이스가 아닌, 수직 확장적 데이터베이스를 사용 -> 단일 실패 지점 발생 (Single Point Of Failure)



## 클라우드 기반 인프라로 Migration

- 2008년, 클라우드로의 Migration 시작 ~ 2016년, 마지막 데이터 센터 종료와 함께 최종 클라우드 Migration 완료
- 기존의 수직 확장적 DB를 수평 확장적 NoSQL DB로 전환 -> 데이터베이스 분산 처리로 SPOF 해결

### References

[1] <https://about.netflix.com/ko/news/completing-the-netflix-cloud-migration>



## Example of Cloud Native Application

# Netflix의 Cloud-Native 적용 사례

## Cloud...생각보다 괜찮은데..?

기존 데이터센터에서의 한계들을 해결해보고자 Cloud-Native 방식 채택

1. 기존의 거대한 앱을 수백 개의 마이크로 서비스로 마이그레이션 (MSA)
2. 중앙화된 릴리스 관리, 서버 프로비저닝을 통한 지속적인 콘텐츠 전달 (CI / CD)
3. DevOps 팀의 지속적인 모니터링, 보안, ...에 대한 빠른 결정 (DevOps)
4. 자체 개발한 Container Management Platform을 통한 컨테이너 관리 (Container & Orchestration)

## Finally, Cloud-Native Application.

- 서비스 가용성
- 자유로운 애플리케이션 도입에 따른 확장성
- 리소스 탄력성
- 보안성과 견고성

물론 전환 과정에서 개발자들은 죽어났습니다..

### Monthly Streaming Hours



폭발적으로 증가하는 User 트래픽을 처리할 수 있는데는  
Cloud의 탄력성이 존재한다.

### References

[1] <https://about.netflix.com/ko/news/completing-the-netflix-cloud-migration>

이론과 현실은 다르다.

# Cloud-Native 개발의 한계

## 1. 솔루션 업체 종속

다른 클라우드 서비스 업체로 이전하는 것이 어려워지고 비용도 많이 들 수 있다.  
컨테이너를 사용하는 CN의 이식성 때문에 쉬워보이지만,  
현실은 여러 이유로 꽤나 복잡하다.

## 2. 기술 격차

클라우드 네이티브 개발이 복잡하고, 기존의 온프레미스 및 퍼블릭 클라우드 개발과는  
다른 기술력과 툴이 필요하다. CN이 익숙치 않은 기업들에게는 오히려 부담.  
엉성하게 만든 클라우드 네이티브 애플리케이션은 그저 복잡하기만 하다.

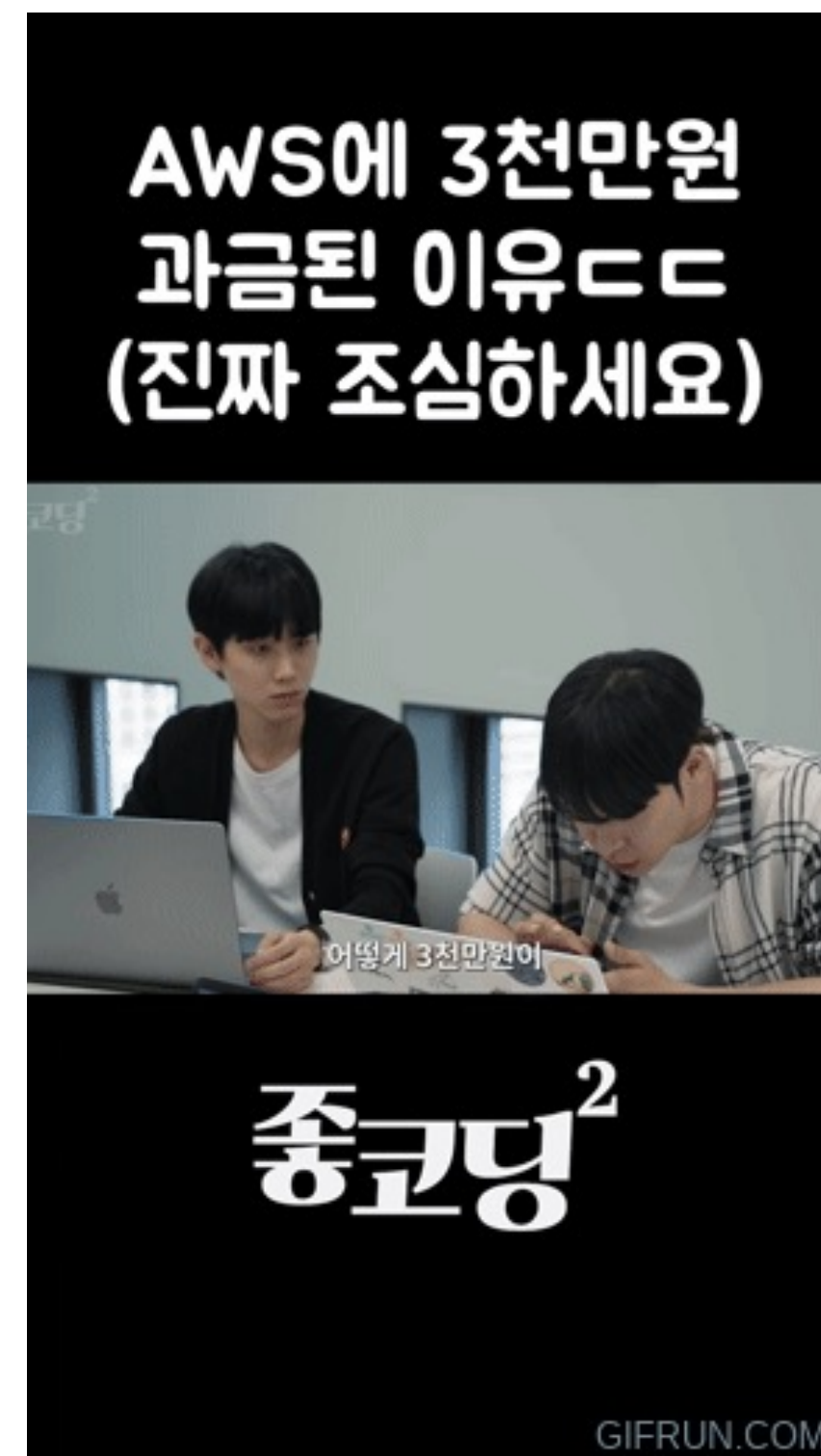
## 3. 과도한 비용

클라우드 네이티브 개발 비용이 예측 불가능하다. (클라우드의 Pay As You Go)  
ex. 트래픽 폭증 -> 리소스 사용량 폭증 -> 비용 폭증  
사용량과 예산을 항상 신중하게 모니터링해야 한다.

### References

[1] <https://www.itworld.co.kr/news/274867>

[2] <https://youtu.be/6jbcpw3YJ4c?si=sLQnYMTAx8-E3ewK>





Cloud Native, 그래서 결론이 뭔데?

# Conclusion

1. Cloud Native는 클라우드의 규모와 크기, 성능에 최적화된 애플리케이션을 개발하기 위한 접근 방식이다.
2. Cloud Native는 유연성, 개발 생산성, 운영 효율성, 개발 기간 단축 등의 이점을 가진다.
3. MSA, CI / CD, DevOps, Container를 통해 Cloud Native Application을 개발할 수 있다.
4. 무조건 좋은 방법은 아니다. 솔루션 업체 종속, 비용 부담, 기술 격차 등의 위험도 고려해야 한다.