

# Cloud Native Application Delivery을 위한 성공적인 전략

이현종 부장

Senior Systems Engineer – Networking

2019년 11월



# Agenda

- 1) Citrix ADC Solution
- 2) 클라우드 네이티브로의 전환 과정
- 3) 마이크로서비스 기반 어플리케이션의 4가지 아키텍처
- 4) 왜 Cloud Native with Citrix?

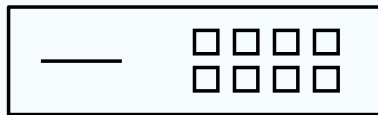


# Citrix ADC Solution

# Citrix Networking Products

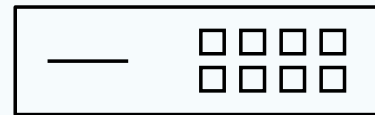
## Application Delivery Management

통합 관리 및 트래픽/사용자 분석 기능 제공



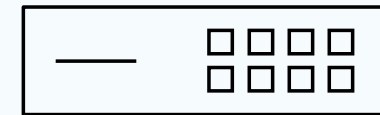
ADC

로드밸런싱을 통한  
서버 가용성 확보



Gateway

언제 어디서든 보안성이  
강화된 사용자 접속 환경 제공



SD-WAN

지점에 안정적이고  
보안성 강화된 서비스 환경 제공

# Citrix ADC Platform – One Code

Physical  
Price-Performance



Virtual  
Run Anywhere



Platform  
Multi-Tenant



Baremetal  
Physical+Virtual

MPX



VPX  
Hypervisor



CPX  
Container



SDX



BLX  
Software ADC





# 클라우드 네이티브로 전환 과정

# 디지털 혁신이 회사의 혁신으로 이어집니다.

## 기대

고객이 쉽게 접근할 수 있는 방법이 필요해!



CEO

네, 현재 우리는 모바일 앱을 만들고 있습니다. 사용자로부터 받은 피드백을 바탕으로, 우리는 앱을 매주 업데이트 할 것입니다.



CPO

기업은 빠르게 전환하기 원합니다 - 새로운 소프트웨어를 생성하고 소프트웨어를 신속하게 구축하기 원함

## 문제점

로드밸런싱 기능이 필요해요!



개발자

대략 2~3주가 필요해요!



네트워킹 운영

조직이 격리화됨 - 앱 개발자 대 네트워킹 팀 - 병목 현상 발생

네트워크가 복잡함 - 변경 요청은 수작업

## 액션

우리는 빠르게 움직여야해!



CIO

자동화? 마이크로서비스?



아키텍트

새로운 플랫폼 팀을 만들어야 해!



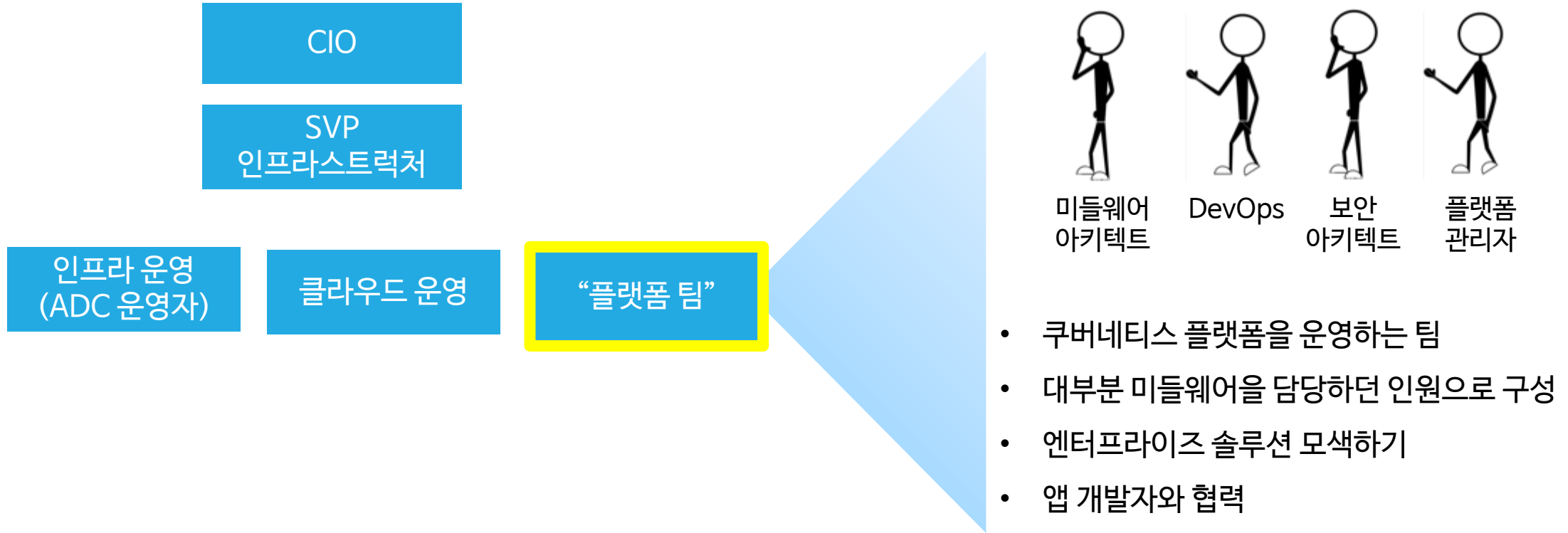
VP

C레벨 경영진이 설계자와 새로운 디지털 책임자가 새로운 접근 방식을 제시할 수 있도록 지원

아키텍트들이 큰 그림의 변화를 주도하고 있음

비즈니스 전반에 걸쳐 업무를 수행할 수 있도록 새로운 플랫폼 팀을 투입하도록 역할 및 책임 조정

# 새로운 “플랫폼 팀”

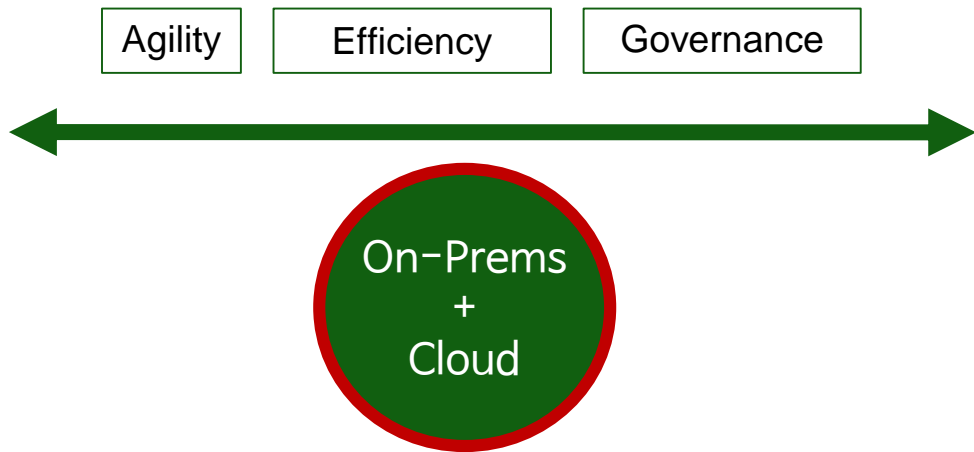


Duke Energy, T-Mobile, Apple, Cisco Webex, Match.com, Electronic Arts (EA), 84.51  
대부분의 회사에서 “플랫폼 팀”을 운영



# 플랫폼 팀 비즈니스 요구사항

플랫폼팀이 원하는 것은,  
온프레미스 및 클라우드에서 일관되게 실행할 수 있는 플랫폼 -  
하이브리드 멀티 클라우드 환경.

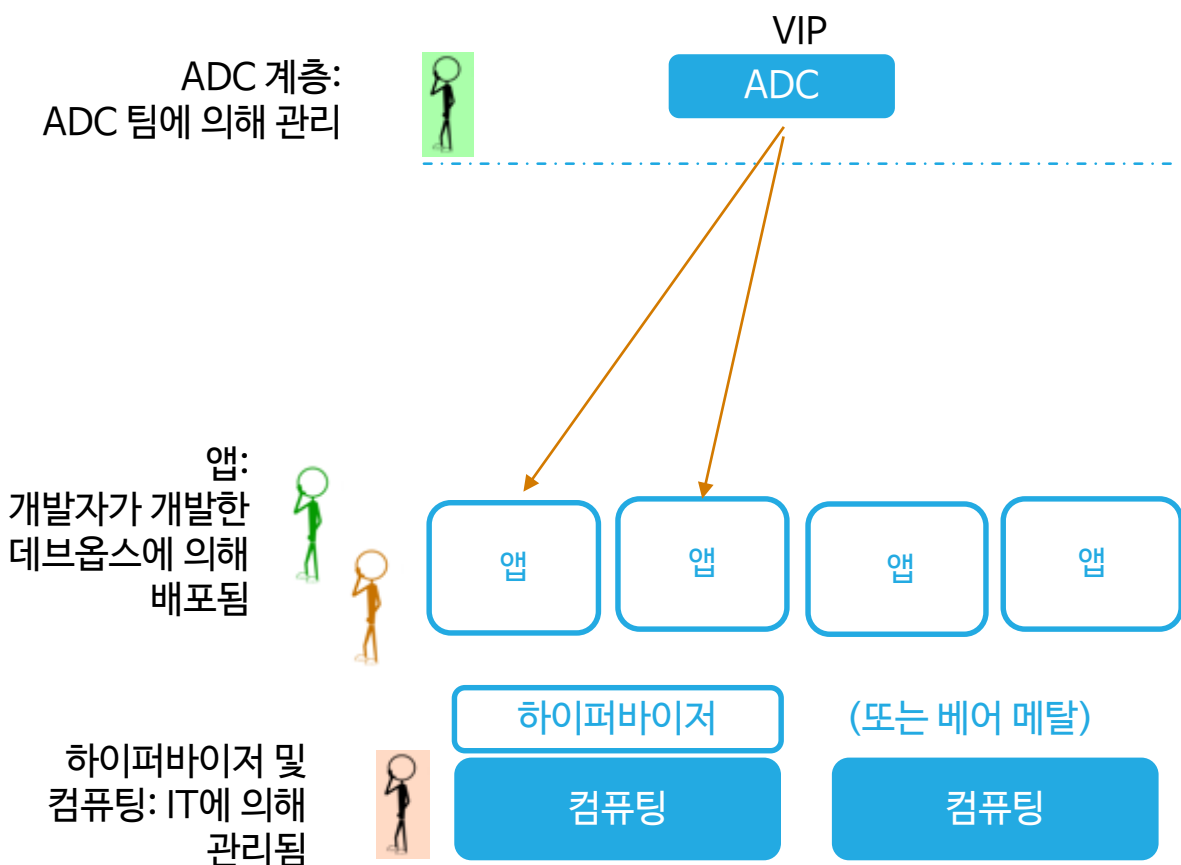


- 균형:
  - 개발자 민첩성: 빠르고 사용하기 쉽게 앱 배포
  - 운영 효율성: 이해 관계자들 간에 공통적으로 사용할 수 있도록 전체 시스템 업데이트, 확장, 관리
  - 플랫폼 관리: 플랫폼의 안전성을 보장하고 보안, 액세스 또는 테넌시(Tenancy) 관련 정책을 충족

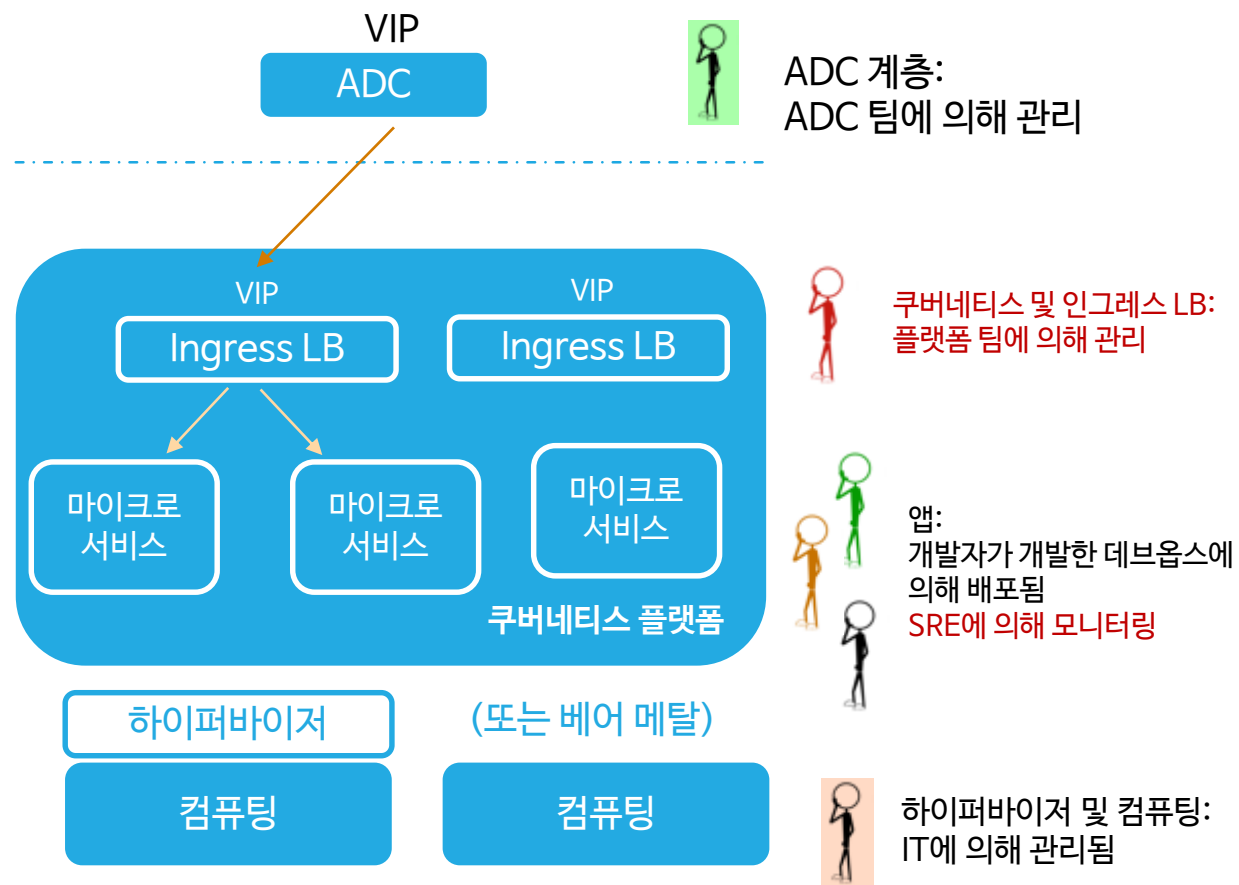
하이브리드 멀티 클라우드 (HMC)

# 3 Tier 에서 Cloud Native : 차이점 및 유사점

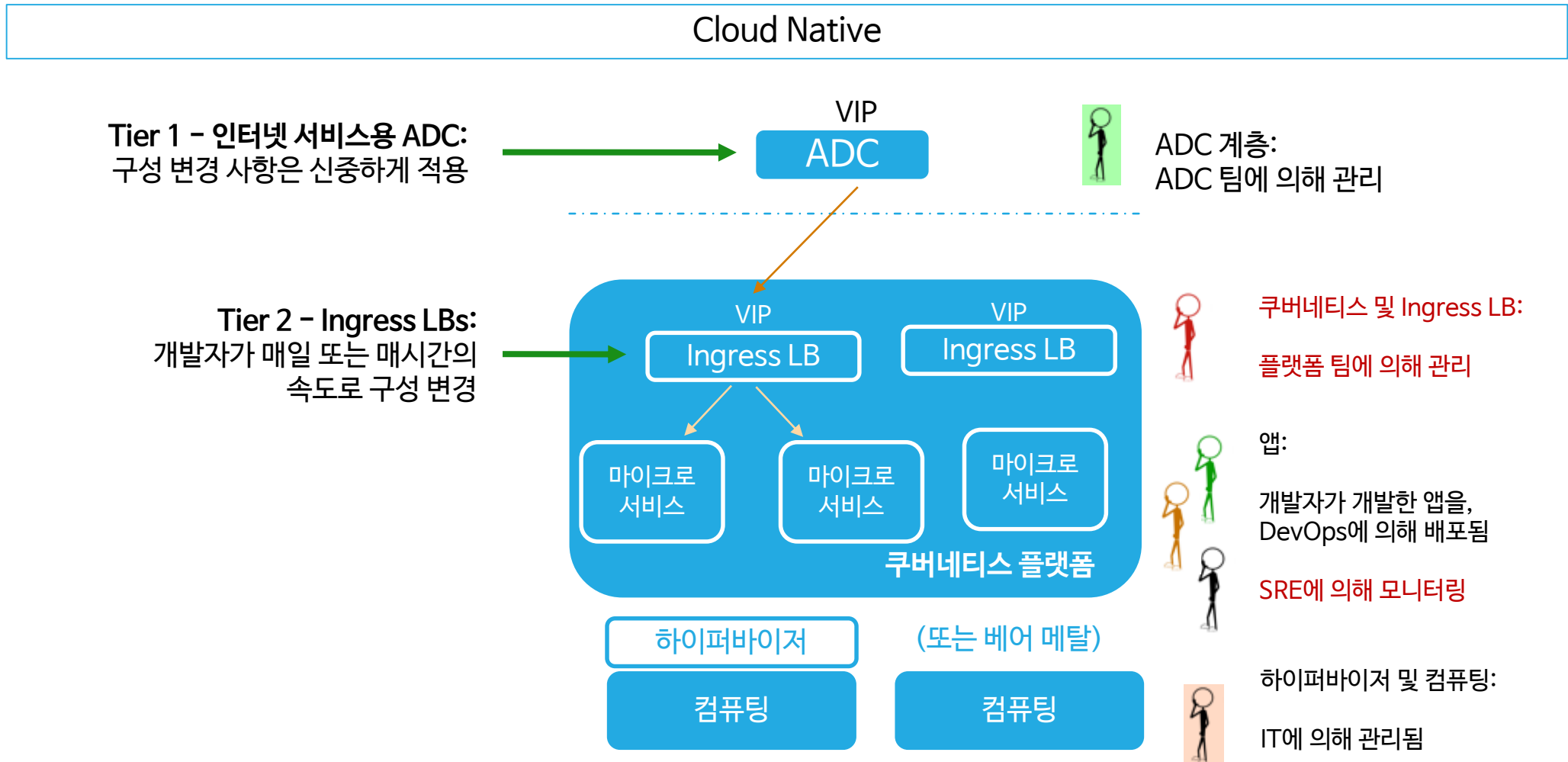
## 3 - Tier



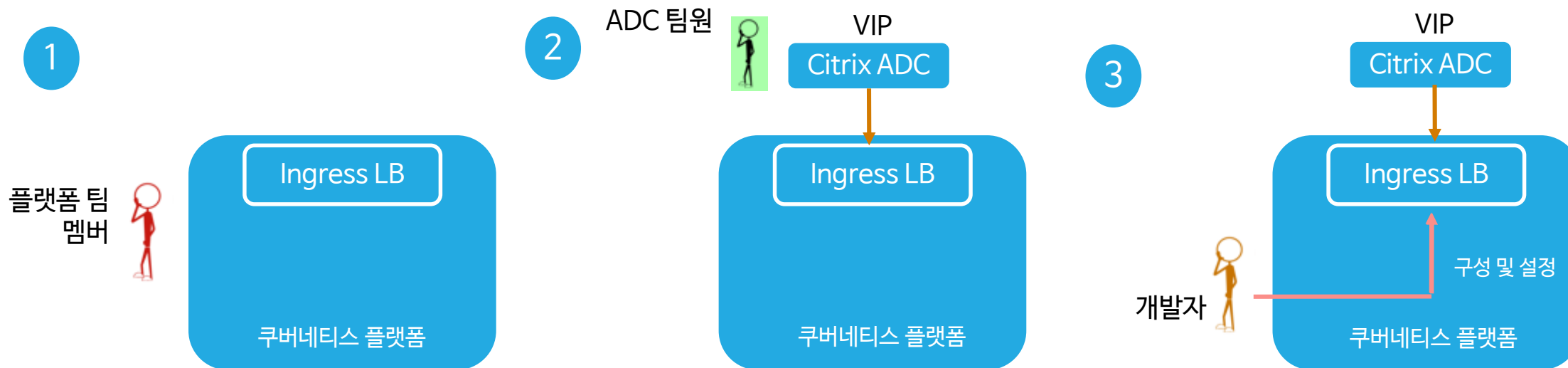
## Cloud Native



# 로드밸런싱 계층을 통한 속도 향상

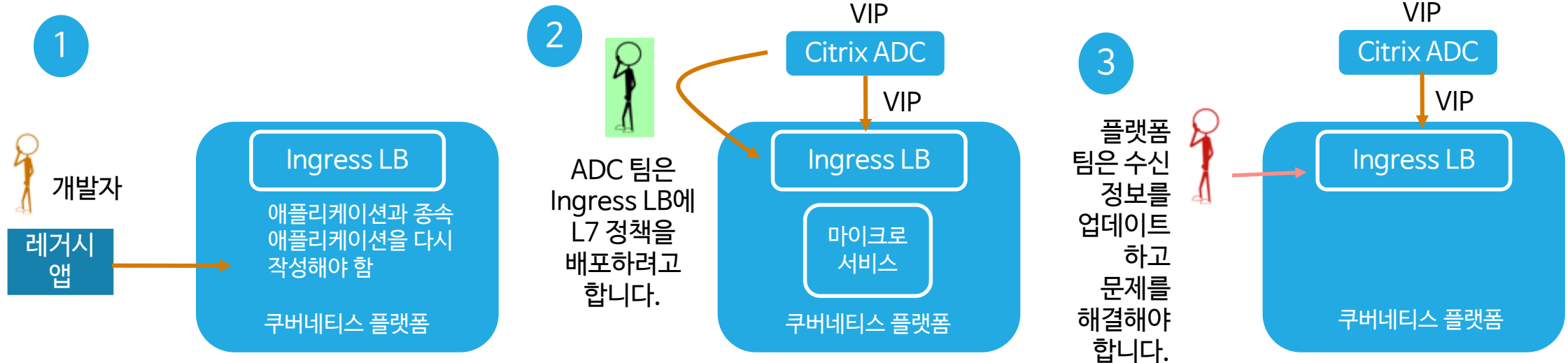


# Customer Journey : 초기 단계



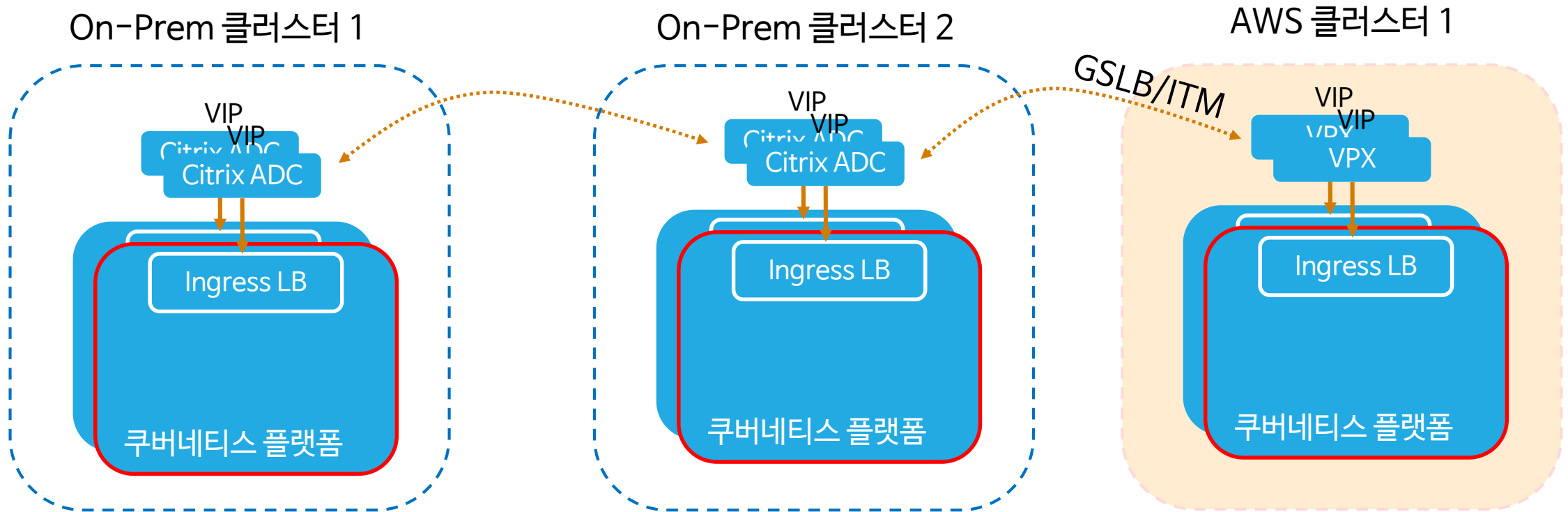
1. 일반적으로 플랫폼 팀은 Kubernetes 플랫폼에 Pre-Built 되어 있는 Ingress LB를 사용
2. 인프라팀(ADC팀)은 플랫폼으로 트래픽을 보내기 위해 VIP를 생성하고, 이 VIP은 Ingress LB의 VIP들을 로드밸런싱 함
3. Citrix ADC는 이제 아키텍처의 일부지만, 아직까지는 수동으로 구성. 이는 Kubernetes 기반 구성의 대상이 아니라는 것을 의미하며, 개발자는 단지 Ingress LB만 구성

# Customer Journey : 컨테이너화



1. 고객은 레거시 앱을 마이그레이션하는 데 문제가 발생 - Ingress LB는 TCP 및 UDP 프로토콜에 대한 지원하는 항목들이 부족하며, 개발자들은 앱과 종속 앱을 다시 작성해야 함
2. 인프라팀(ADC팀)은 애플리케이션을 보호하고자 하며 Citrix ADC의 Rewrite/Responder 정책에 익숙하며 오픈 소스 LB보다 유연성이 뛰어나다고 판단
3. 고객은 Ingress LB를 운영하는 방법을 고려하기 시작하고 정기적으로 업데이트되는 지원되는 제품이 더 안전한 방법이라는 것을 이해

# Customer Journey : 하이브리드 멀티 클라우드

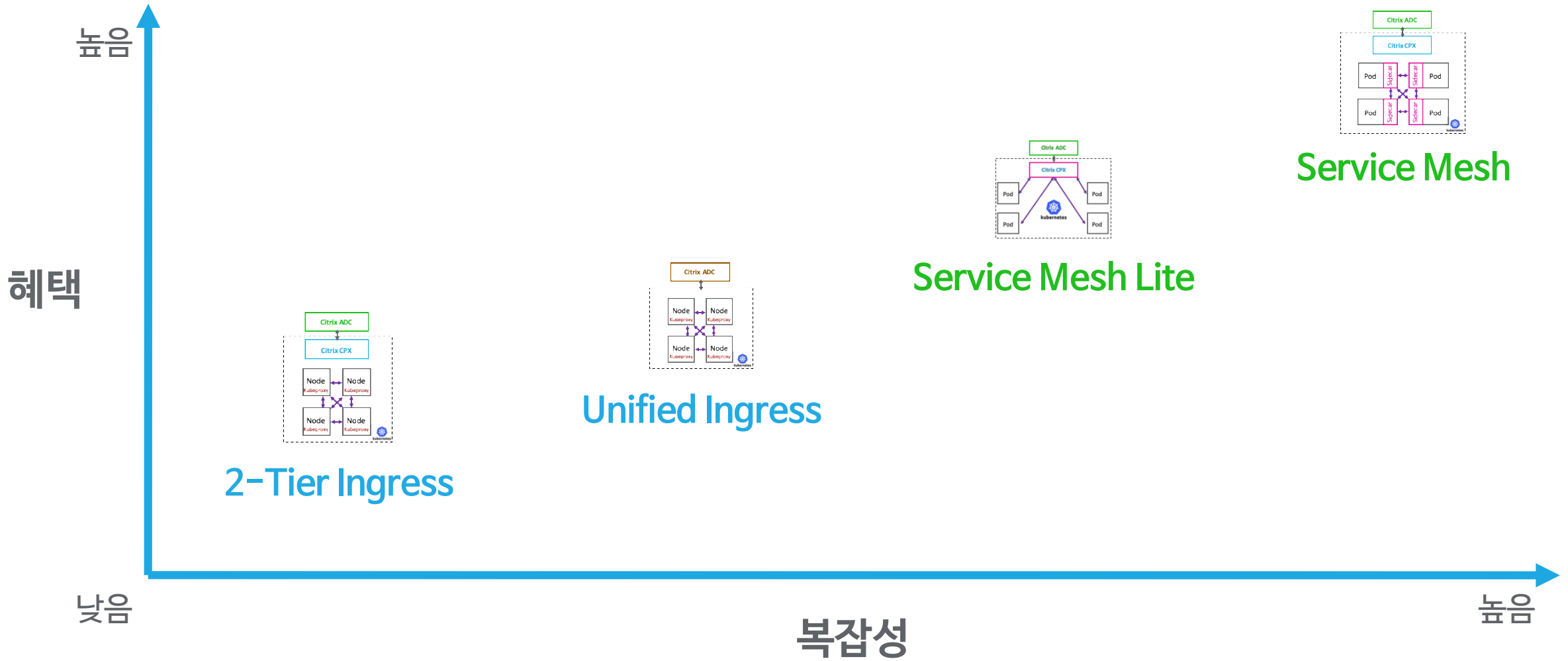


- 고객이 퍼블릭 클라우드에 하나 이상의 Kubernetes 클러스터와 하나 이상의 클러스터를 연결하고자 합니다.
- 그들은 이 VIP를 노출하기를 원함



# 마이크로서비스 기반 어플리케이션의 4가지 아키텍처

# 마이크로서비스 기반 애플리케이션을 위한 아키텍처 선택



혜택

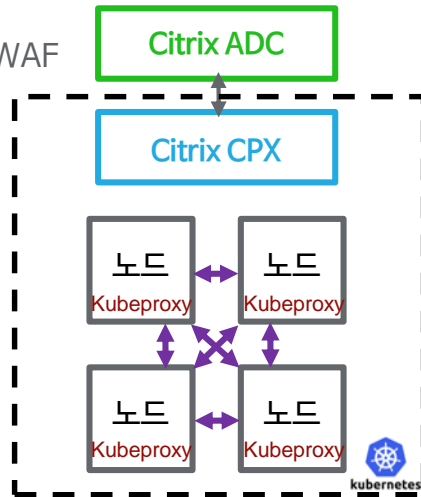


# 2-Tier Ingress : Production에서 가장 단순하고 빠르게 사용

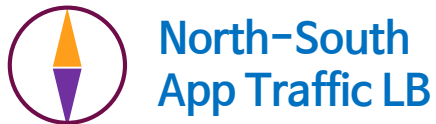
클라우드 네이티브 초보자와 전문가 모두에게 적합

## 2-Tier Ingress

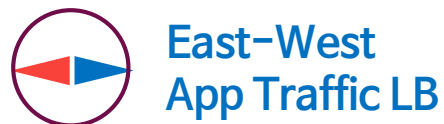
- 네트워킹 팀에 의해 관리
- N-S L4 로드 밸런싱, SSL, WAF
- Citrix ADC



- 플랫폼 팀에 의해 관리
- N-S L7 로드 밸런싱
- Citrix CPX



클라우드 네이티브를 위한 L4 LB용 **녹색 ADC**, 모노리스 애플용 L4-7 LB  
L7 LB용 **파란 ADC** 및 빠른 속도 변화



Kubeproxy에 의한 기본 레이어 4 로드 밸런싱 (라운드 로빈)

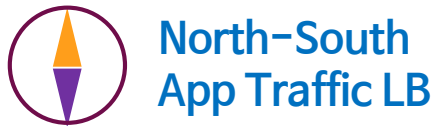
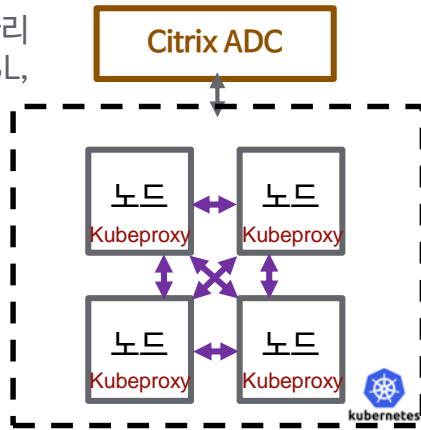
<b>앱 보안</b>	N-S: 녹색 ADC에 의한 탁월한 보호 기능 E-W: 없음. 네트워크 정책/세그먼트 필요 예: 프로젝트 캘리코
<b>관측 가능성</b>	N-S: 우수, 녹색 및 파란색 ADC에서 모든 트래픽 확인 E-W: 매우 제한된 원격 측정
<b>연속 배포</b>	N-S: 우수; ADC에 의한 고급 트래픽 제어 E-W: Kubeproxy 제한으로 인한 부족
<b>확장 성능</b>	N-S: 스케일 아웃에 적합함 E-W: IPVS 모드 사용, iptables 모드는 확장성이 없음
<b>오픈 소스 소프트웨어 도구 지원</b>	N-S: 우수; 예: 프로메테우스, 스피나커, EFK E-W: Kubeproxy 제한으로 인해 제한됨
<b>Istio: 통합 제어 평면</b>	N-S: Istio 지원 ADC를 통한 지원 E-W: Kubeproxy가 Istio가 활성화되지 않았습니다.
<b>IT 기술 필요사항</b>	플랫폼 및 네트워킹 팀을 위한 최소한의 교육 두 팀 모두 자신에 맞는 속도로 옮겨갈 수 있습니다.

# Unified Ingress : 네트워크 기반 플랫폼 팀에게 단순함

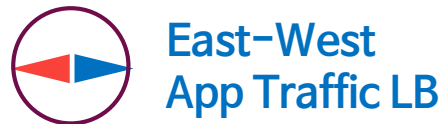
WAF / SSL 및 외부 앱을 나중에 추가할 수 있는 내부 앱에 적합

- 네트워크에 정통한 플랫폼/인프라 팀에 의해 관리
- N-S L4-7 로드 밸런싱, SSL, WAF

## Unified Ingress



클라우드 네이티브 및 모노리스 앱을 위한 L4-7 로드 밸런싱용 브라운 ADC



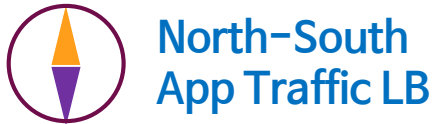
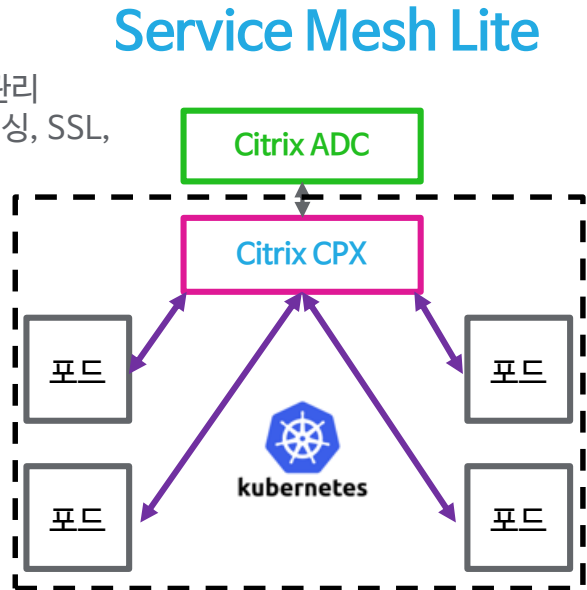
Kubeproxy에 의한 기본 레이어 4 로드 밸런싱 (라운드 로빈)

<b>앱 보안</b>	N-S: 갈색 ADC에 의한 탁월한 보호 E-W: 없음. 네트워크 정책/세그먼트 필요 예: 프로젝트 캘리코
<b>관측 가능성</b>	N-S: 우수; 갈색 ADC가 모든 트래픽을 확인. E-W: 매우 제한된 원격 측정
<b>연속 배포</b>	N-S: 우수; ADC에 의한 고급 트래픽 제어 E-W: Kube-proxy 제한으로 인한 부족
<b>확장 성능</b>	N-S: 스케일 아웃에 적합함 E-W: IPVS 모드 사용, Iptables 모드는 확장성이 없음
<b>오픈 소스 소프트웨어 도구 지원</b>	N-S: 우수; 예: 프로메테우스, 스피나커, EFK E-W: Kube-proxy 제한으로 인해 제한됨
<b>Istio: 통합 제어 평면</b>	N-S: Istio 지원 ADC를 통한 지원 E-W: Kube-proxy가 Istio가 활성화되지 않았습니다.
<b>IT 기술 필요사항</b>	<i>플랫폼/인프라 팀이 네트워크에 능숙해야 함</i>

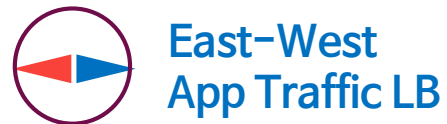
# Service Mesh Lite : 서비스 메시와 같은 이점 및 단순성

마이크로서비스 간의 트래픽 보호, 앱별 암호화 옵션, 세분화된 트래픽 관리, 관측 가능성

- 네트워킹 팀에 의해 관리
- N-S L4-7 로드 밸런싱, SSL, WAF
- Citrix ADC



클라우드 네이티브 및 모노리스 앱을 위한 L4-7 LB 및 보안을 위한 **녹색 ADC**



E-W 고급 로드 밸런싱을 위한 **보라색 ADC**

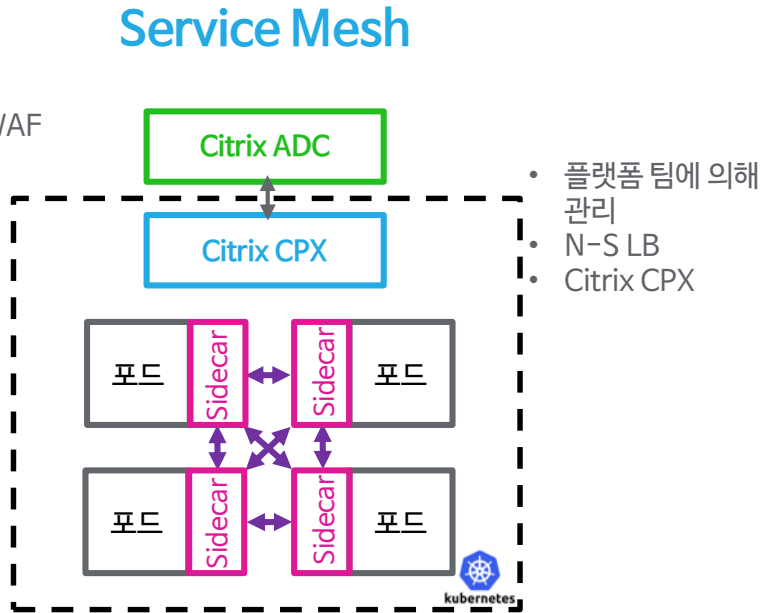
<b>앱 보안</b>	N-S: 녹색 ADC에 의한 탁월한 보호 기능 E-W: 보라색 ADC, <i>mTLS (옵션)</i> 으로 탁월한 보호 기능 제공
<b>관측 가능성</b>	N-S: 우수; 녹색 ADC이 모든 트래픽이 확인 E-W: 우수; 보라색 ADC가 모든 트래픽을 확인.
<b>연속 배포</b>	N-S: 우수; ADC에 의한 고급 트래픽 제어 E-W: 우수; 보라색 ADC에 의한 고급 트래픽 제어
<b>확장 성능</b>	N-S: 스케일 아웃에 적합함 E-W: 뛰어난 확장성, 1-홉 지연 시간 추가
<b>오픈 소스 소프트웨어 도구 지원</b>	N-S: 우수; 예: 프로메테우스, 스피나커, EFK E-W: 우수; 예: 프로메테우스, 스피나커, EFK
<b>Istio: 통합 제어 평면</b>	N-S: Istio-지원 ADC를 통한 지원 E-W: Istio API를 통해 지원, Istio 믹서 병목 현상.
<b>IT 기술 필요사항</b>	<i>플랫폼 및 네트워킹 팀을 위한 최소한의 교육</i> <i>2-Tier Ingress 아키텍처에서 간편한 전환</i>

# Service Mesh : 복잡하지만, 최고의 관찰성, 보안성 제공

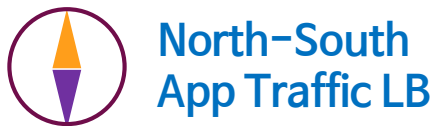
마이크로 서비스 간 매우 안전한 트래픽, 세분화된 트래픽 관리, 일부 앱 기능을 사이드카로 오프로드.

- 네트워킹 팀에 의해 관리
- N-S L4 로드 밸런싱, SSL, WAF
- Citrix ADC

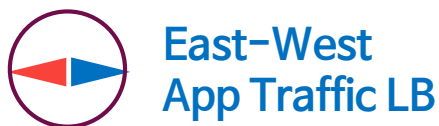
- 플랫폼 팀이 관리하는 사이드카
- E-W 로드 밸런싱 등
- 사이드카로 시트릭스 CPX



- 플랫폼 팀에 의해 관리
- N-S LB
- Citrix CPX



클라우드 네이티브를 위한 L4 LB용 **녹색 ADC**, 모노리스 앱용 L4-7 LB; L7 LB용 **파란 ADC** 및 빠른 속도 변화



E-W 고급 로드 밸런싱을 위한 **사이드카**. 포드는 사이드카를 통해 통신합니다.

<b>앱 보안</b>	N-S: 녹색 ADC에 의한 탁월한 보호 기능 E-W: 사이드카, 정책, 속도 제어, 인증, MTL, API 및 레이어 7 공격 보호로 탁월한 보호
<b>관측 가능성</b>	N-S: 우수; 녹색 및 파란색 ADC가 모든 트래픽을 확인. E-W: 우수; 사이드카는 모든 트래픽을 확인
<b>연속 배포</b>	N-S: 우수; ADC에 의한 고급 트래픽 제어 E-W: 우수; 사이드카로 고급 트래픽 제어
<b>확장 성능</b>	N-S: 스케일 아웃에 적합함 E-W: 분산형 아키텍처 확장성, 사이드카 품질에 따라 2홉 지연 시간 추가, CPU/메모리 증가
<b>오픈 소스 소프트웨어 도구 지원</b>	N-S: 우수; 예: 프로메테우스, 스피나커, EFK E-W: 우수; 예: 프로메테우스, 스피나커, EFK
<b>Istio: 통합 제어 평면</b>	N-S: Istio-지원 ADC를 통한 지원 E-W: Istio API를 통해 지원, Istio 믹서 병목 현상.
<b>IT 기술 필요사항</b>	<i>플랫폼 및 네트워킹 팀을 위한 빠른 학습 곡선</i>



# Why Cloud Native with Citrix?

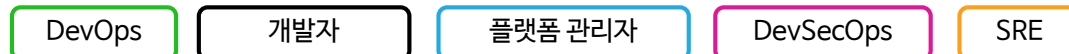
# Cloud Native Stack without Citrix



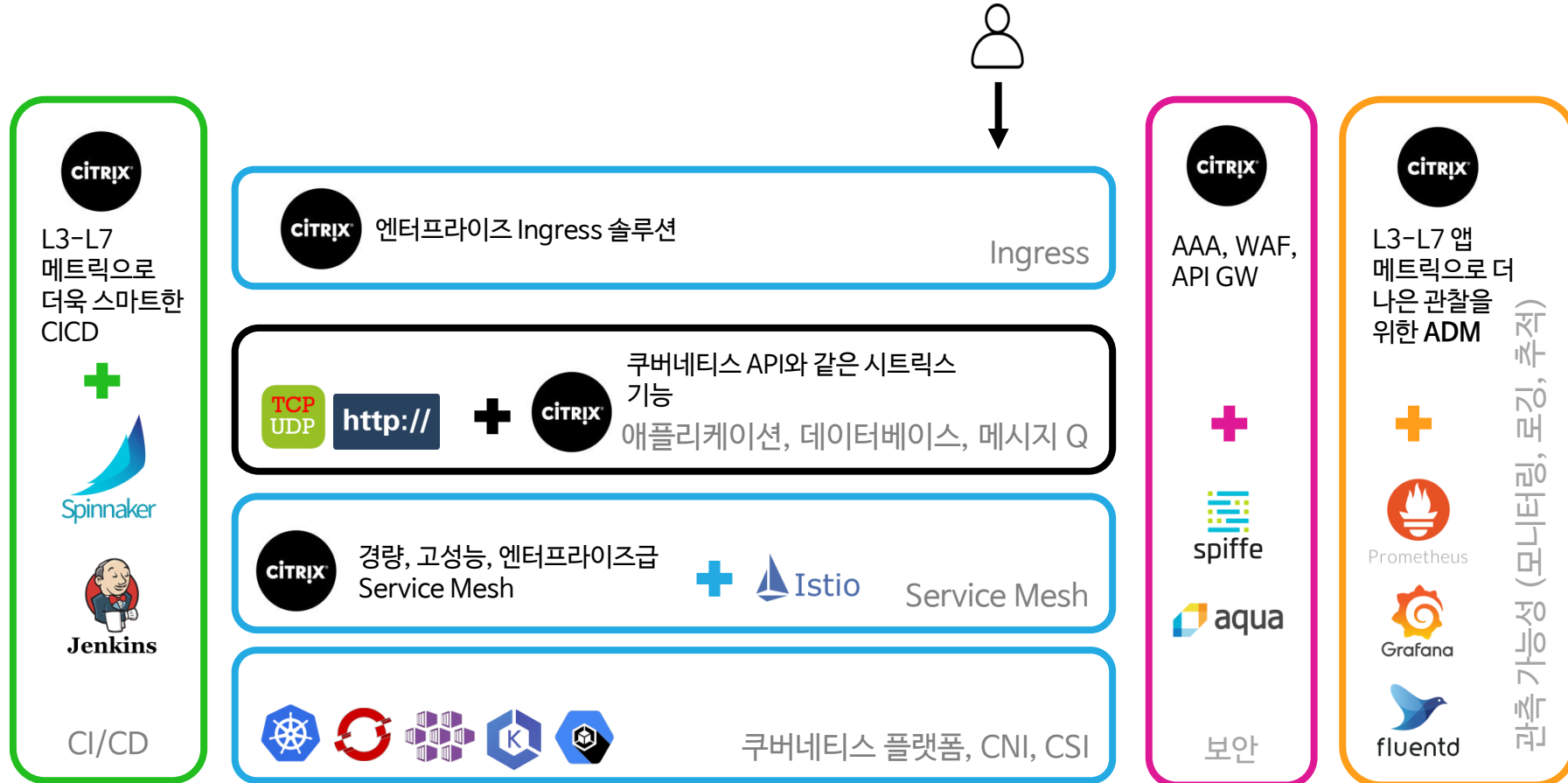
## • 고객의 Challenges,

- 레거시 앱, 데이터베이스 및 보안 앱에 대한 제한된 Ingress 지원
- OSS 프록시를 업데이트, 패치 및 지원하는 개발자
- 일부 OSS 프록시는 재구성 중에 재시작이 필요하므로 다운타임 최소화
- SSL 관련 문제 해결
- OSS 툴을 확장하여 애플리케이션 성능의 가시성을 위한 L3-L7 메트릭스 확보

이해 관계자:

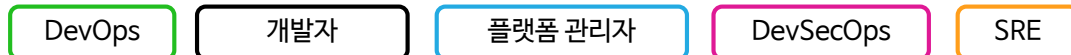


# Cloud Native Stack with Citrix



- Citrix ADC 및 ADM의 이점,
  - 레거시 앱을 다시 작성할 필요 없이 이동
  - 개발자는 Kubernetes API를 사용하여 Citrix ADC 정책을 사용하여 앱을 보호할 수 있습니다.
  - North-South 및 서비스 메쉬를 위한 고성능 마이크로 서비스 배포
  - 모든 마이크로 서비스에 하나의 애플리케이션 서비스 그래프 사용
  - TCP, UDP, HTTP/S, SSL을 통해 마이크로서비스 문제를 보다 빠르게 해결
  - API를 보호하고 쿠버네티스 API를 사용하여 구성
  - 카나리아 배포를 위한 CICD 프로세스 자동화

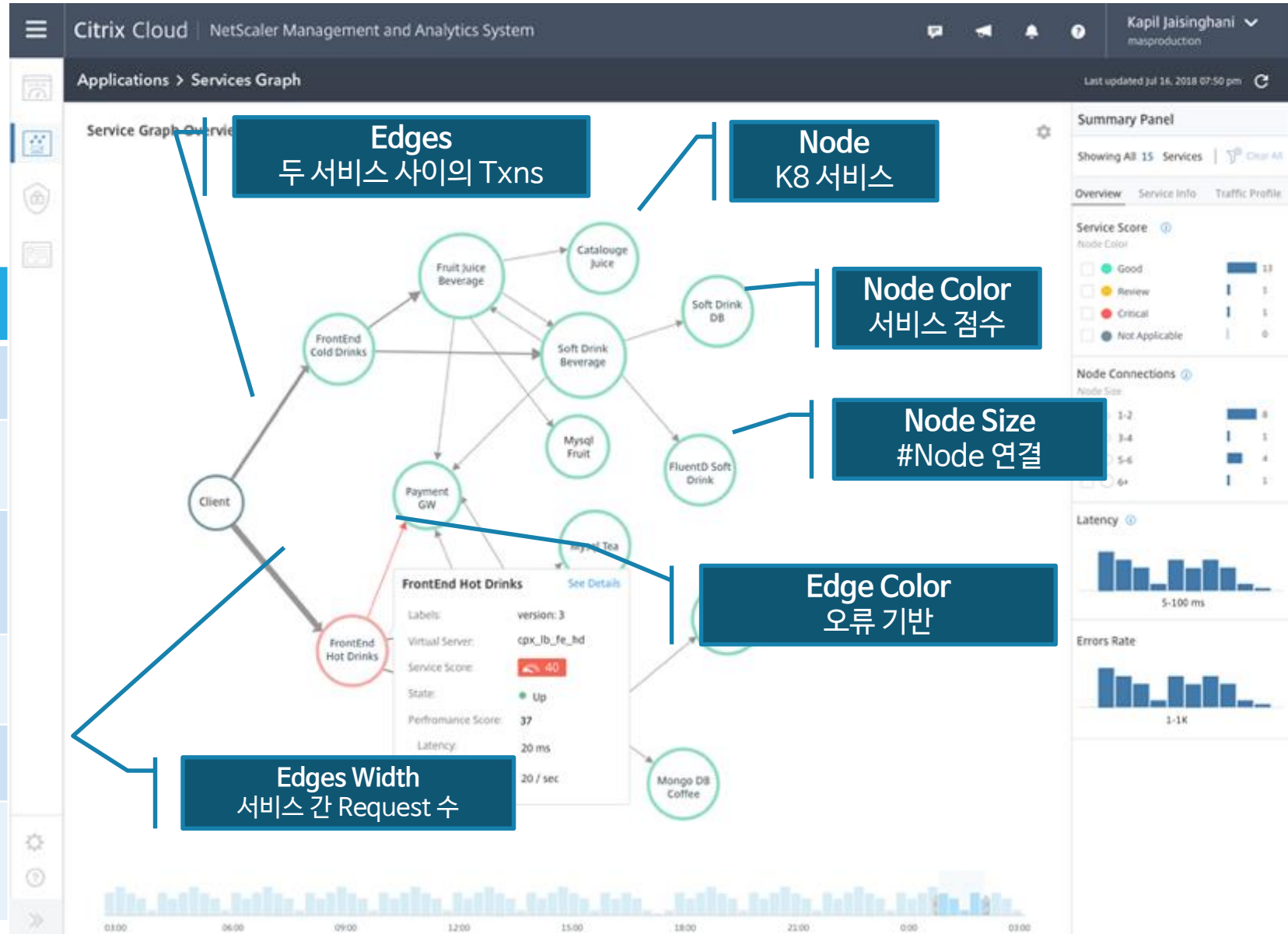
이해 관계자:



Citix ADC는 Envoy에 비해 지연 시간 (1ms) 이 거의 없음 (최대 6ms)

# 마이크로서비스 관리

기능	매핑
Node	K8 서비스
Edges	서비스 기반 — 서비스 Txn
Node Color	서비스 점수 (성능 요소 기준) <ul style="list-style-type: none"> <li>지연 시간</li> <li>오류 비율</li> </ul>
Node Size	노드 연결 수 (엣지)
Edges Width	서비스 간 요청 수
Edge Color	강조 표시는 오류를 바탕으로 합니다.



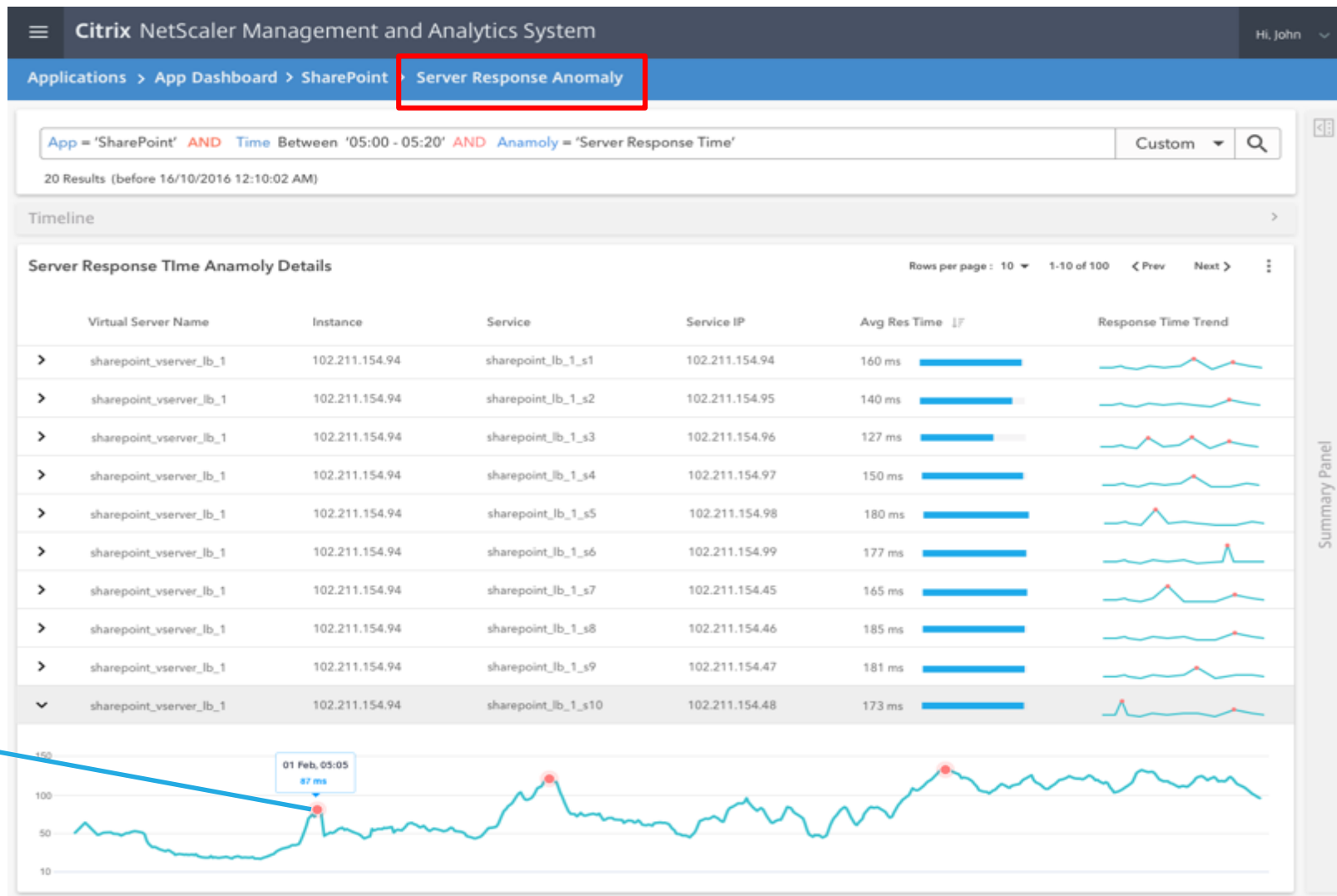


# 성능 문제 해결을 위한 이상 징후 감지

서버 응답 시간  
이상에 기여하는  
서비스는 무엇입니까?

이 서비스의 이상  
징후 추세는  
어떻습니까?

이상 징후가 확인된  
지점은?



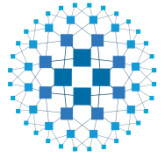
# 왜 오픈 소스 프록시상에서 Citrix를 사용하는가? (1/2)



NGINX

## 1. 오픈 소스는 실제로 프로덕션에 무료입니까?

- 기업에서는 프로덕션계에 솔루션을 배포하려면 엔터프라이즈등급 지원이 필요합니다. 모든 사람이 무료 오픈 소스 NGINX, HAProxy, Specialty를 관리/패치/업그레이드하기 위한 전용 엔지니어링 대역폭을 가지고 있는 것은 아님
- 종종 오픈 소스 프록시는 기능이 제한되어 있으므로 고객은 프리미엄 버전에 가입해야 합니다.



HAPROXY

## 2. Citrix ADC가 동적 워크로드를 더 잘 처리 — 구성 파일이 변경될 경우 재로드/재부팅 불필요

- HAProxy 구성을 변경하려면 다시 로드해야 합니다. 활성 연결이 있으면 HAProxy 는 기존 프로세스를 종료하지 않습니다. 따라서 많은 수의 재로드로 동시에 실행되는 HAProxy 프로세스의 수가 많을 가능성이 있습니다. 메모리 부족 상태가 발생합니다. [오픈시프트 참조](#).



envoy

## 3. 통합 ADC — Citrix ADC는 동일한 코드 기반, 다양한 폼 팩터 (MPX, SDX, VPX, CPX, BLX) 에서 사용자 환경을 사용하는 유일한 ADC/프록시입니다. 플랫폼 설계자는 부하 분산 장치의 무분별한 증가를 방지하는 동종 인프라 구성 요소를 좋아합니다. 동서남북에 대해 동일한 솔루션

## 4. 고성능 프록시 — Citrix ADC 는 Envoy에 비해 지연 시간 (1ms) 이 거의 없음 (최대 6ms)

## 5. 다중 프로토콜 지원 — TCP/UDP 응용 프로그램 (예: 데이터베이스, 메시지 버스, 레거시 응용 프로그램), HTTP2 최신 응용 프로그램, HTTP3 지원 예정



## 왜 오픈 소스 프록시상에서 Citrix를 사용하는가? (2/2)

NGINX



HAPROXY



envoy



træfik

6. **개발자 친화적인 CRD** — CRD를 사용하면 NetScaler 전문가가 되지 않고도 Citrix ADC 기능을 쉽게 사용할 수 있습니다. 다양한 팀 (개발자, 데브옵스, SRE, 플랫폼 관리자) 이 Citrix ADC를 신속하게 사용할 수 있도록 도와줍니다.
  - NGINX, HAProxy는 주석 및 ConfigMaps를 사용합니다.
7. **ADM을 통한 관찰 가능성** — 직관적인 서비스 그래프를 통해 클라우드 네이티브 애플리케이션 문제 해결
8. **보안** — WAF, IP 화이트리스트/블랙리스트, 속도 제한, TLS 인증서/키 관리
9. **풍부한 통계, 모니터 카운터** — 다양한 SSL, TCP, HTTP/S 메트릭/카운터 세트
10. **다중 클러스터 솔루션** — 여러 지역에서 호스팅되는 Kubernetes 클러스터에 ITM, GSLB 사용

**CITRIX®**