

Cloud Infrastructure Operations

Day2 Module1 - App Service 및 서버리스

윤혜식 | 2miles CEO
david@2miles.co.kr





App Service



클라우드 환경에서 앱 개발시 주로 하는 질문들

처음부터 새로운 앱을 만들
여유가 없습니다. 어떻게
해야 하나요?



클라우드를 사용하려면
새로운 언어와 새로운
툴을 배워야합니까?



기존 코드를 재사용
할 수는 없습니까?

애플리케이션 현대화 장점

민첩성

지속적인 전달(CD)
컨테이너



제품 출시 시기 단축

고가용성 및 확장성
분석 & 인사이트



TCO 절감

인프라 비용 절감
유지 보수 절감



IT 구조 단순화

표준화
단순화



클라우드 전환 모델

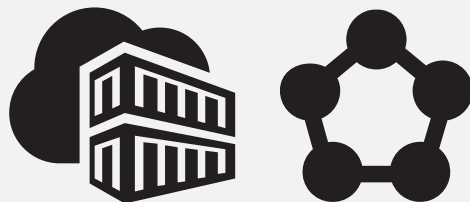
Lift and Shift



- Azure VMs (IaaS)
- Azure SQL Database Managed Instance

코드 수정 없음

DevOps 현대화 (Containers, DevOps)



- Docker Containers
- Service Fabric
- Azure SQL Database Managed Instance
- CI/CD

최소한의 코드 수정

클라우드 네이티브 (PaaS, Serverless, Microservices)

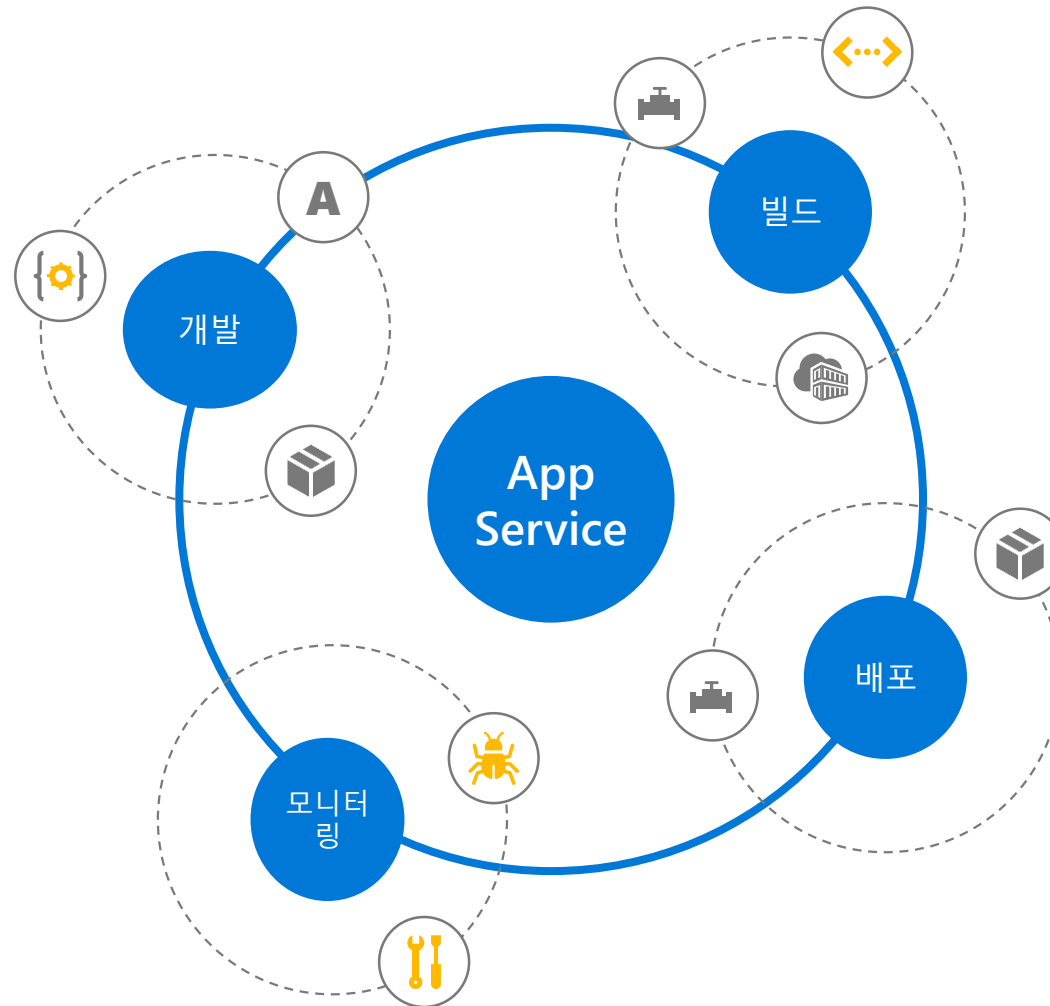


- App Service
- SQL Database
- Functions
- Service Fabric
- CI/CD

새롭게 코드 작성

App Service 란?

개발, 빌드, 배포 및 모니터링이 완전히 관리되는 환경 제공



Azure App Service 특징

인프라에 대한 걱정없이 강력한 클라우드 애플리케이션을 신속하게 구축, 배포 및 확장



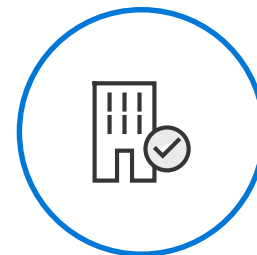
높은 생산성

기존 개발 언어, 친숙한 프레임워크, 내장 CI / CD 및 지능형 진단 기능을 사용하여 개발 가속화



완전한 관리

응용 프로그램 로직에 집중하고 Azure에서 용량 프로비저닝, OS 패치, 확장, 부하 분산 조정 및 도메인 관리와 같은 작업 처리

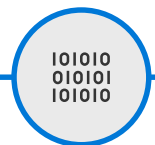


엔터프라이즈 지원

다양한 호스팅 옵션, 강력한 규정 지원 및 MSI 통합을 통해 엄격한 성능, 보안 및 규정 준수 요구 사항 충족

Azure App Service 도입 효과

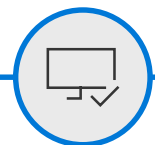
높은 생산성



.NET, Node, Java, Docker,
PHP, Ruby, Python



스테이징/개발
배포 지원

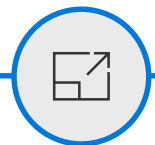


상용환경에서
테스트



앱 갤러리
마켓플레이스

완전한 관리



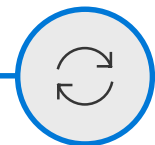
오토스케일링 및
부하 분산



고가용성 및
자동 패치

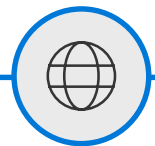


운영 비용
절감

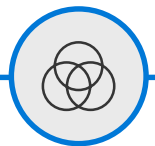


백업 및 복구

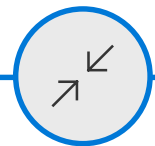
엔터프라이즈 지원



글로벌 데이터
센터를 통한 확장



하이브리드
지원

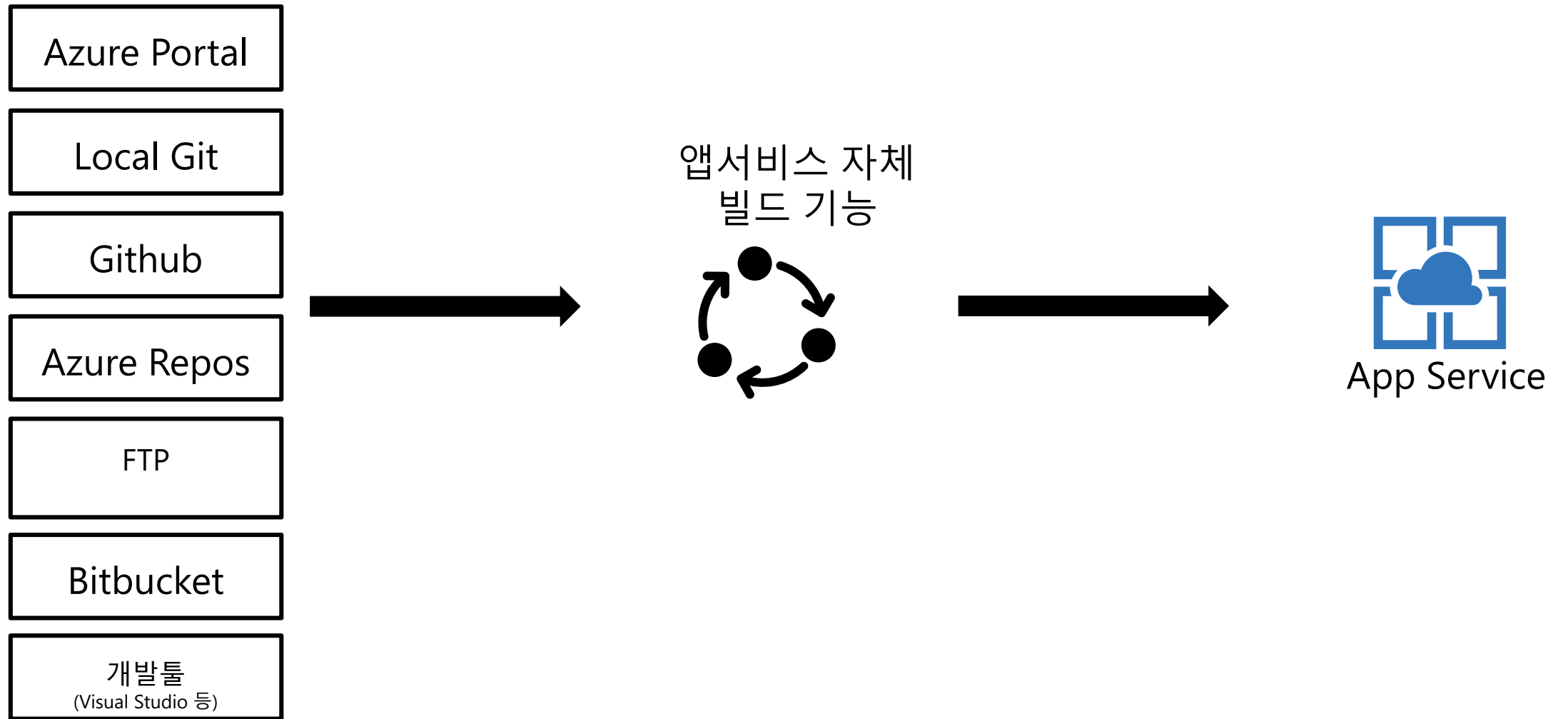


AAD
통합



보안 및
컴플라이언스

Azure App Service: 빌드 및 배포 방법

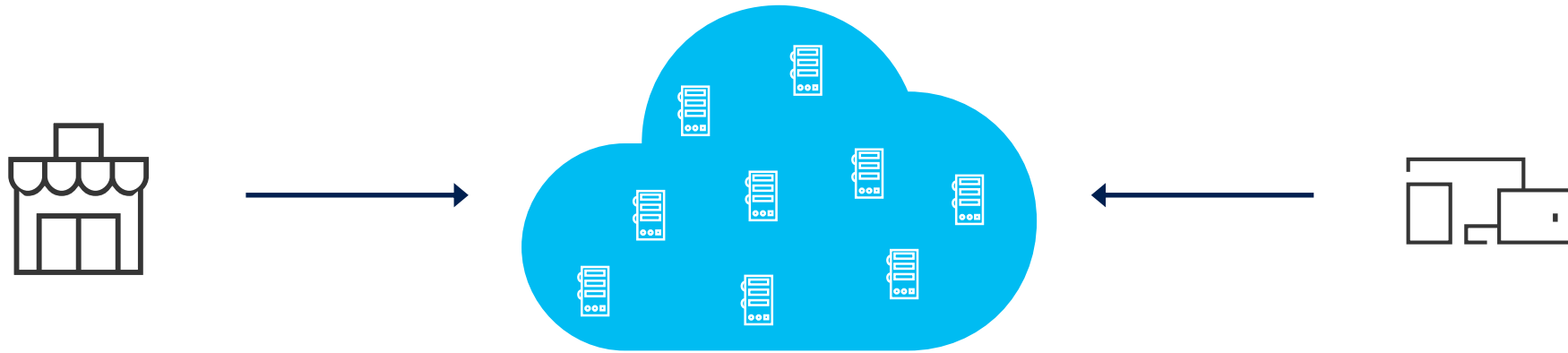




Serverless



클라우드 네이티브 앱 개발을 위한 서버리스



On-Premises

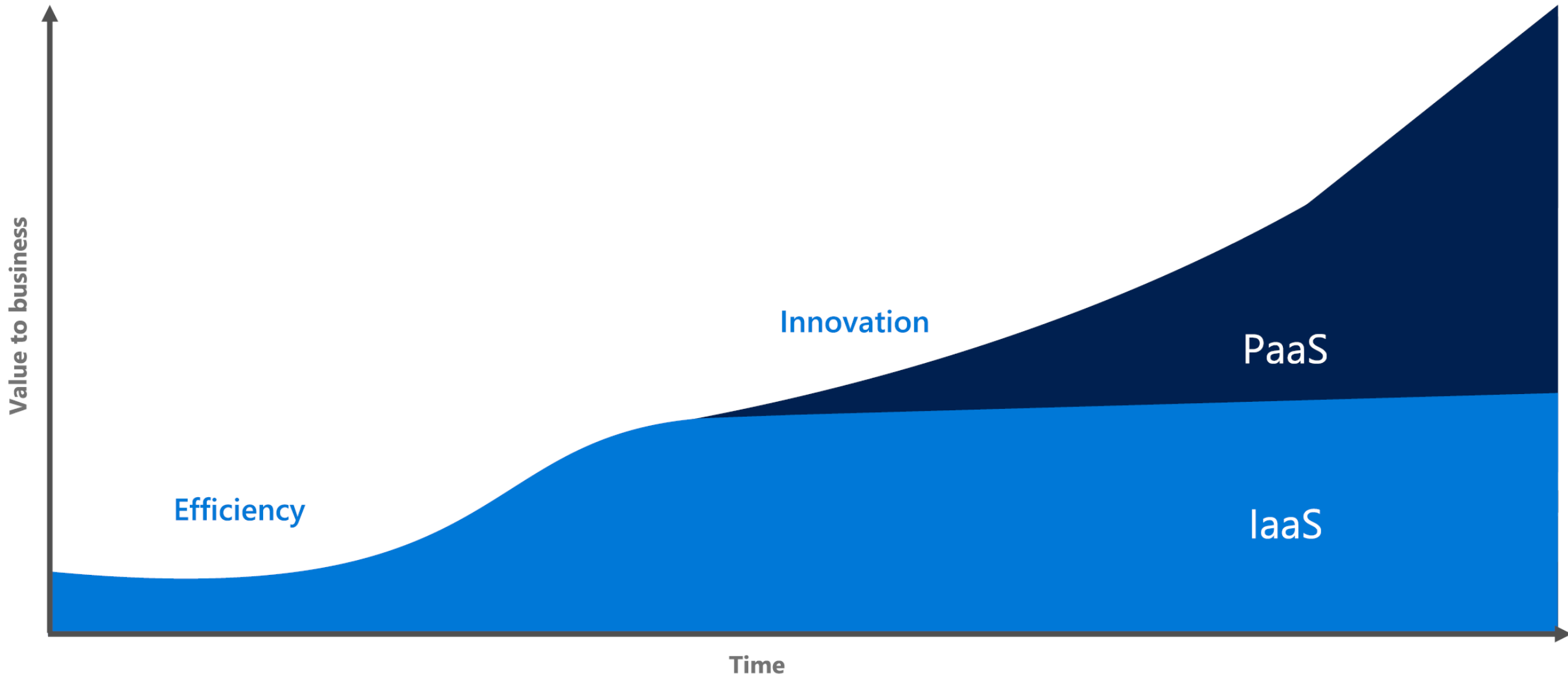
IaaS

PaaS

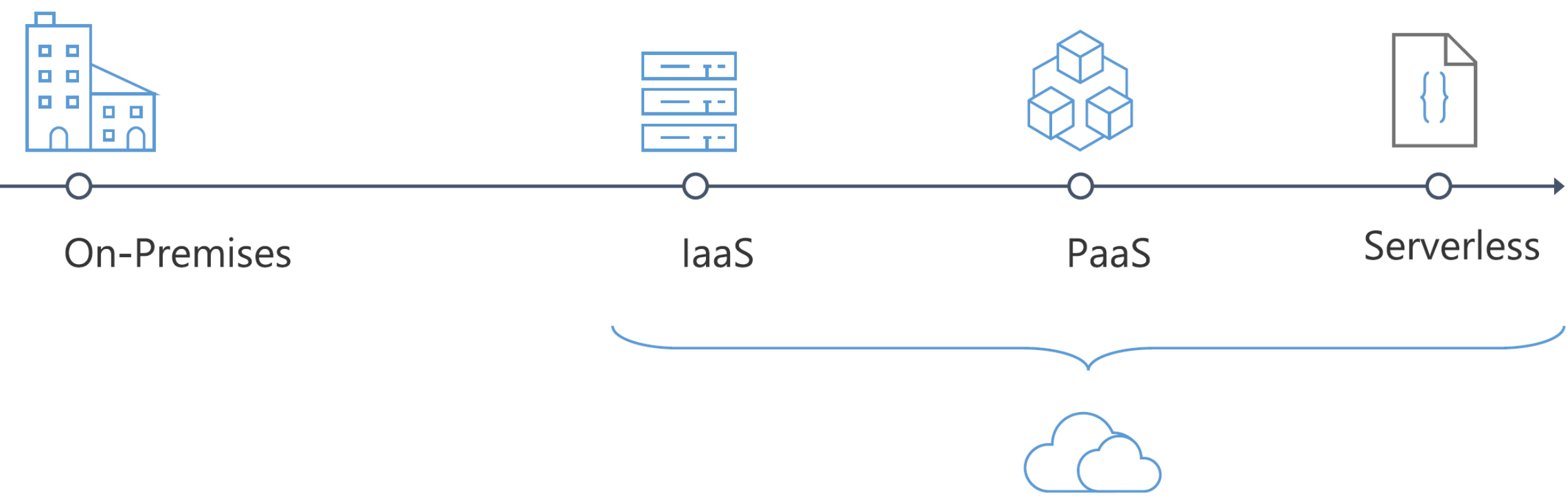
Serverless

애플리케이션 플랫폼의 진화

성숙해진 클라우드 PaaS 서비스



응용프로그램 플랫폼의 진화

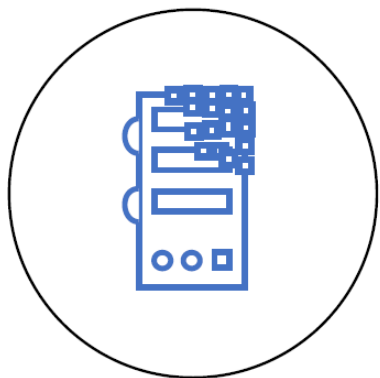


Serverless, 차세대 응용프로그램을 위한 플랫폼

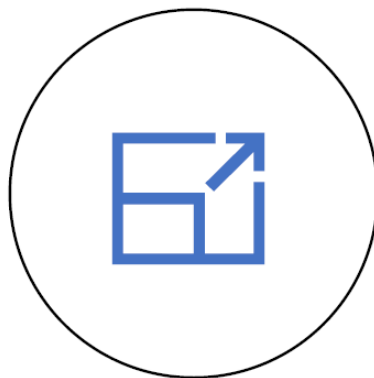
앱을 서버리스가 되도록 하려면 어떻게 설계해야 할까?



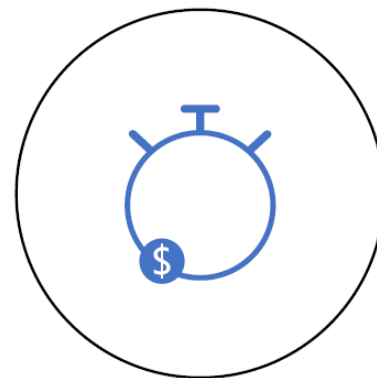
Serverless란 무엇일까?



서버의
추상화



이벤트 기반/
인스턴스 스케일



마이크로
과금

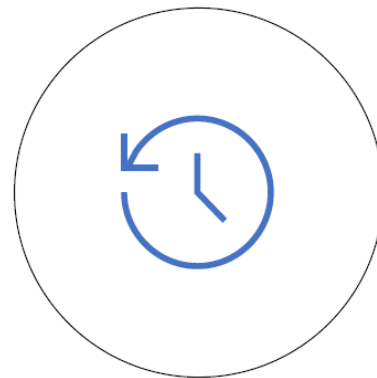
Serverless의 혜택



서버 관리는 무시,
앱만을 관리

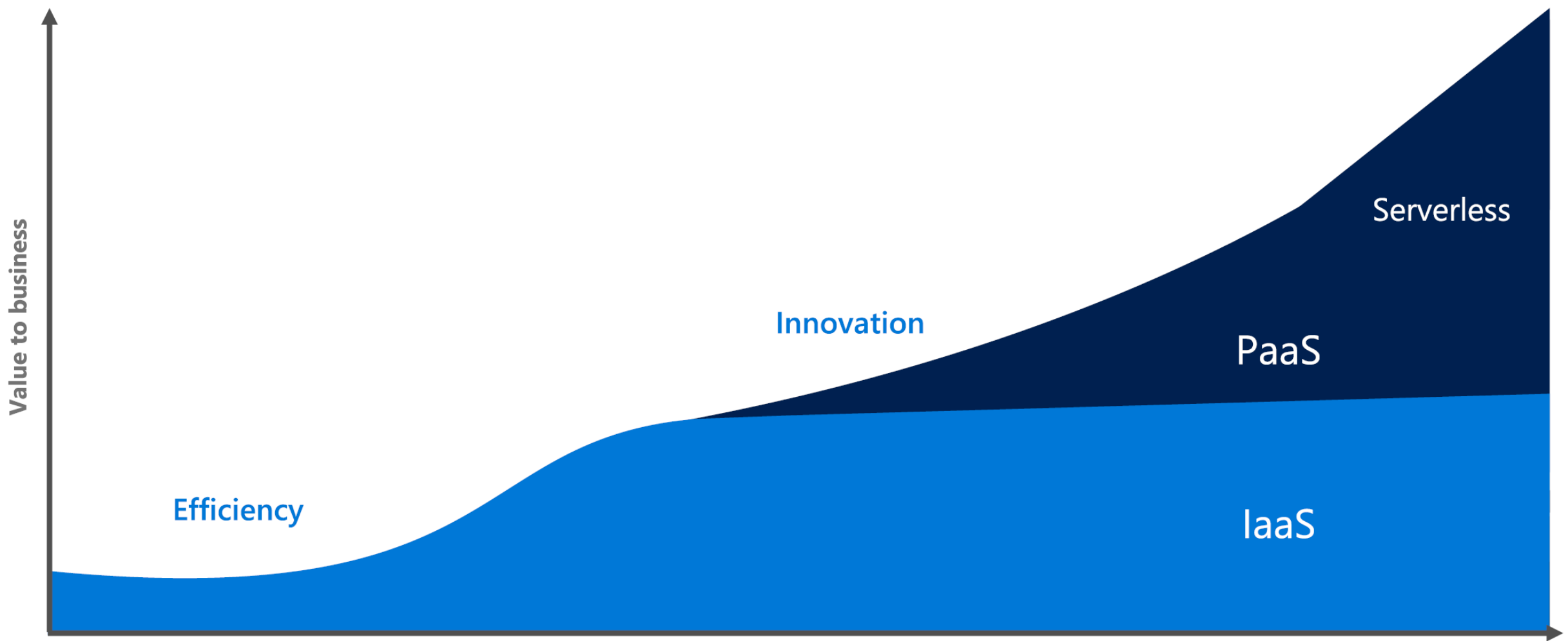


빨라지는
DevOps

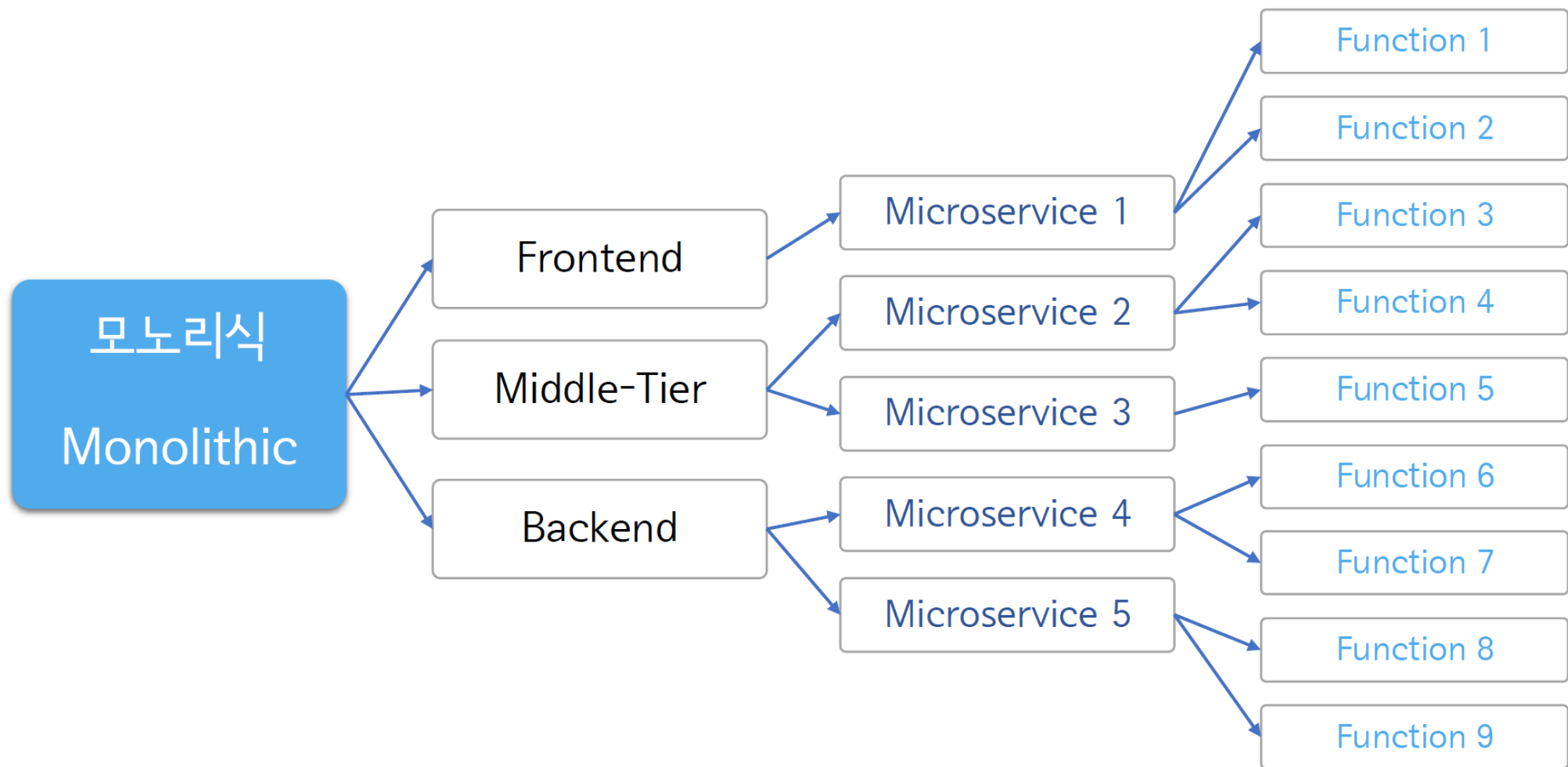


더 빠른
제품 출시

Serverless로 더욱 빠르게 앱을 개발



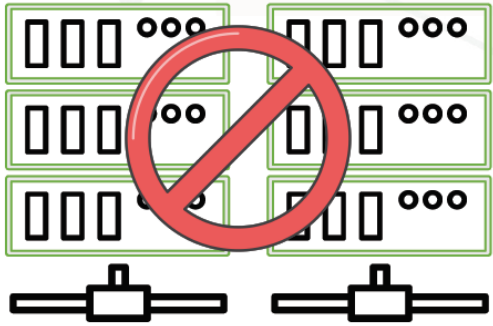
응용프로그램 플랫폼의 진화



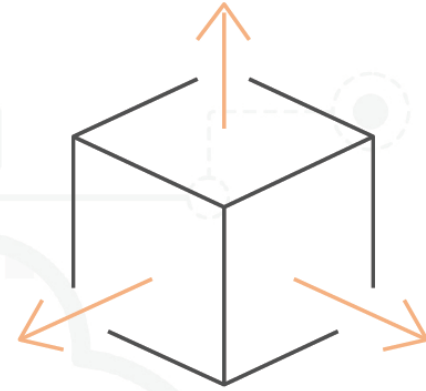
서버리스(Serverless)의 의미

- 패턴이기 보다는 신뢰할 수 있는 시스템
 - 서버리스 시스템에 부합하는 패턴들이 별도로 존재함
- 서버리스 응용프로그램
 - 개발자가 서버를 구축하거나 관리할 필요가 없는 응용프로그램
- 훌륭한 PaaS 혹은 Micro-PaaS
- FaaS(Function as a Service)는 서버리스의 일부

서버리스의 이점



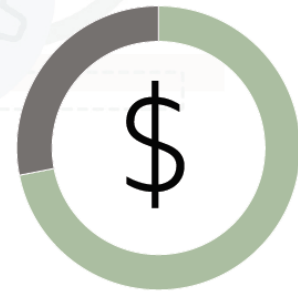
서버 관리 불필요



유연한 확장성



고가용성



유휴 용량 없음

Azure Functions 서비스란?

Code



Azure Functions

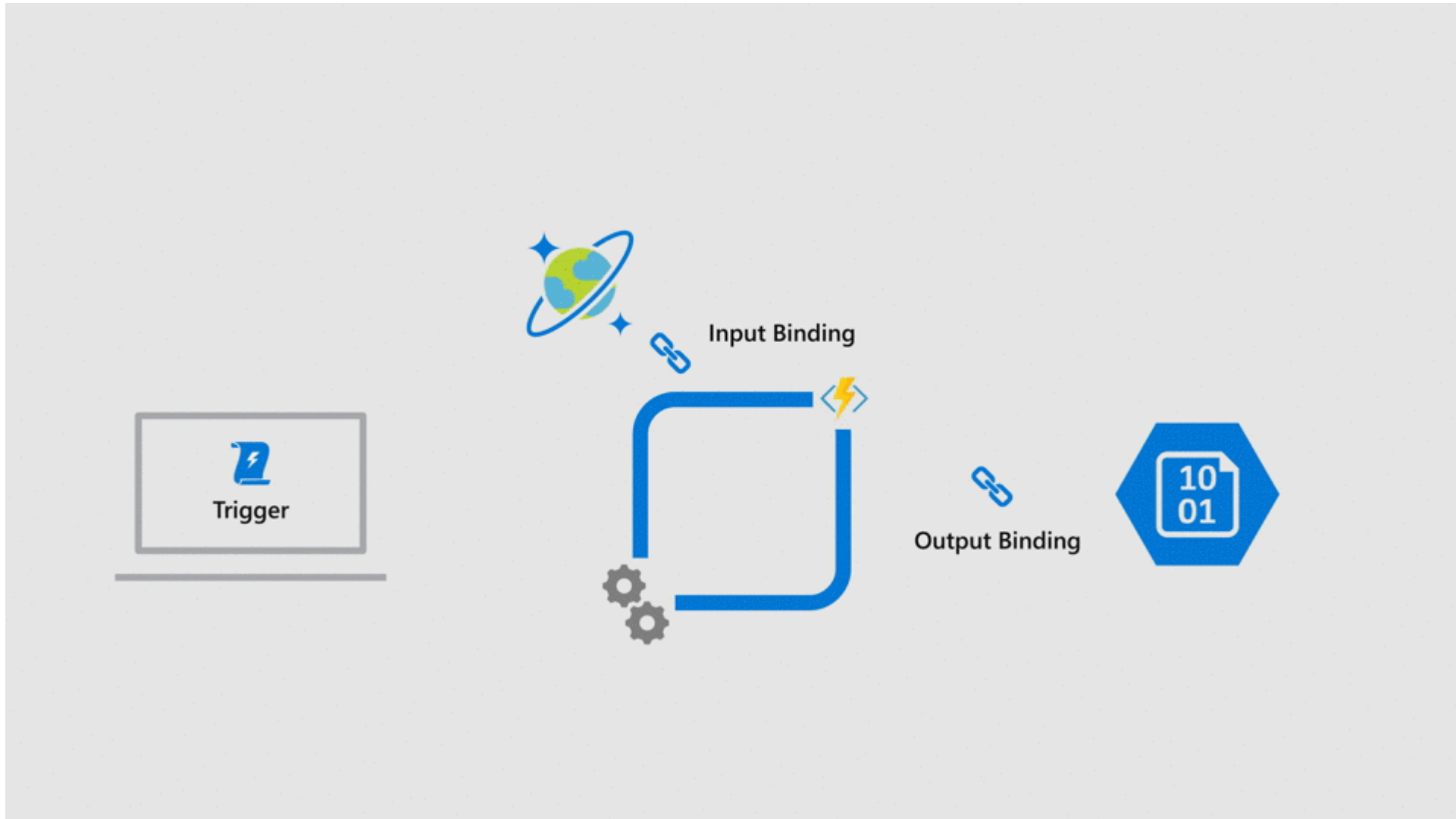


Events

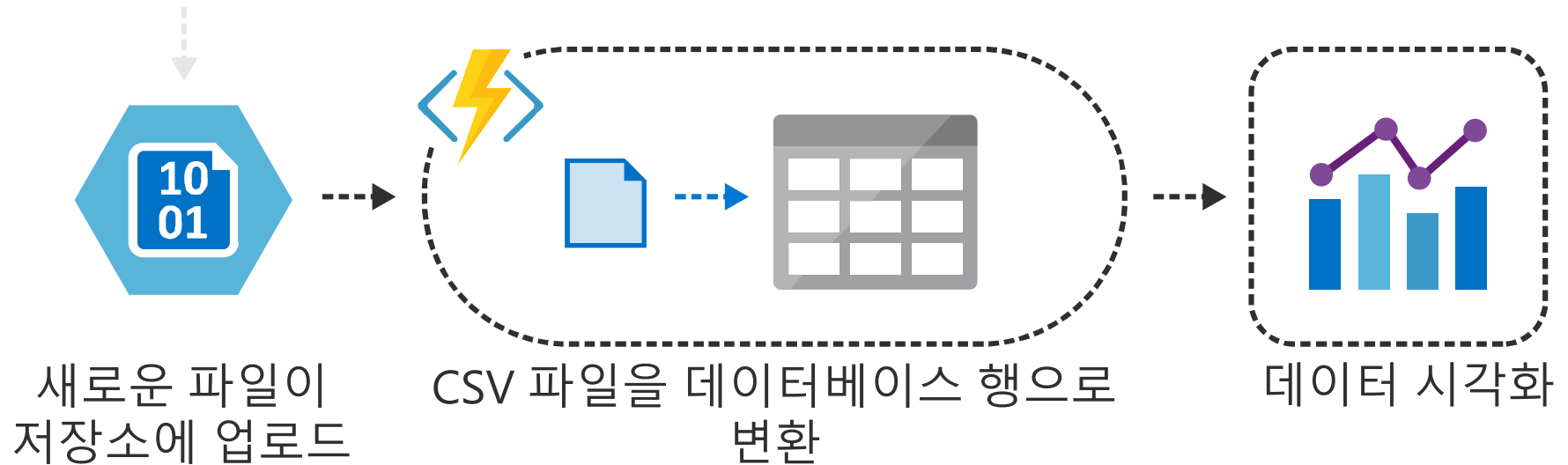


비즈니스 로직에만 집중

하드 코딩 통합 없이 다른 서비스를 연결하여 더 빠르게 솔루션 개발

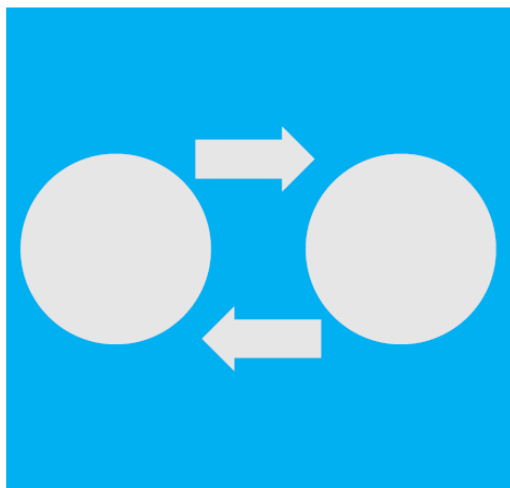


Azure Functions 활용 예제

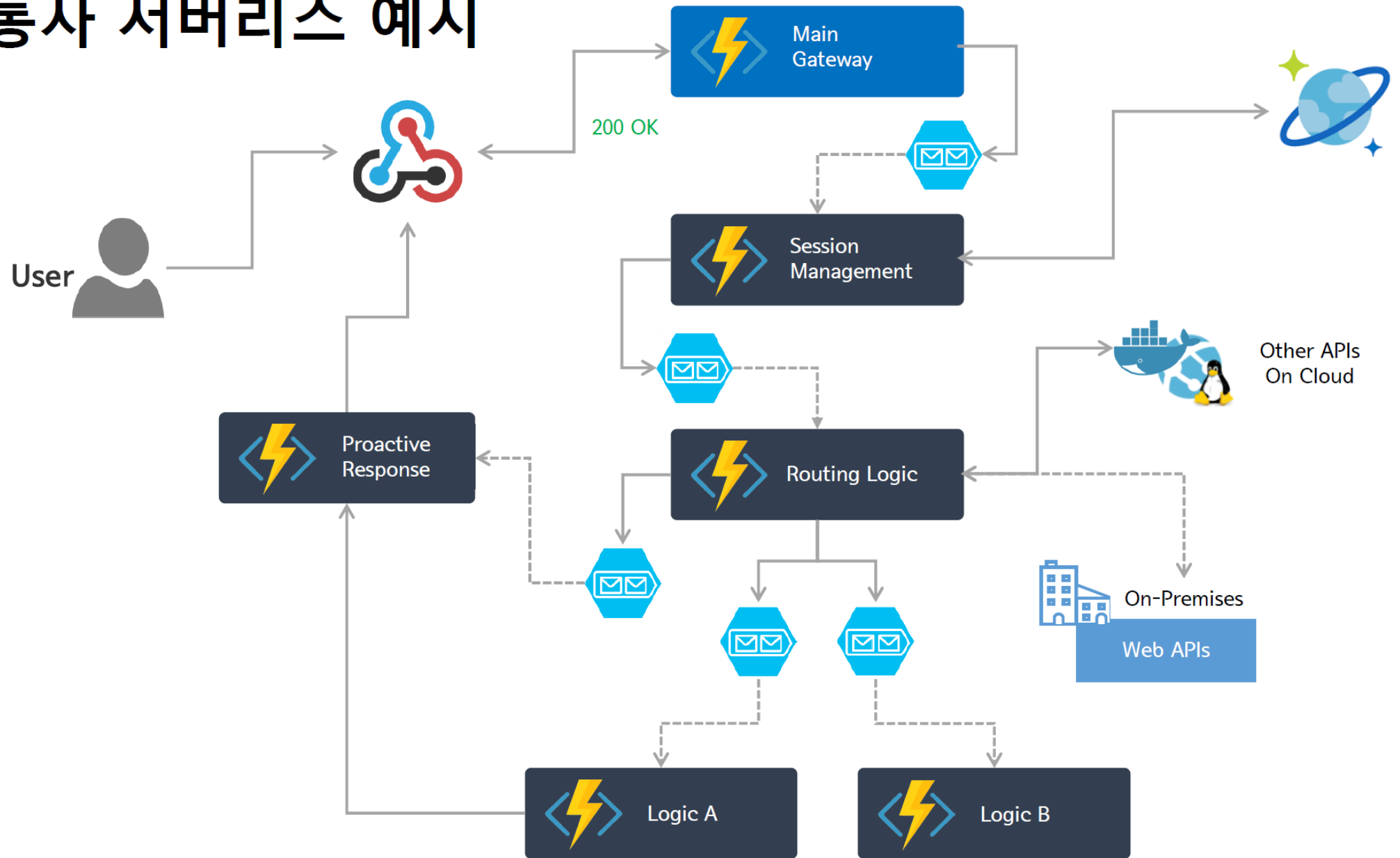


“Functions” 프로그래밍 모델, 기본 규칙

- Functions은 반드시 “Do One Thing”을 해야 한다
- Functions은 반드시 멱등적(idempotent)이어야 한다
- Functions은 반드시 가능한 한 빨리 끝나야 한다

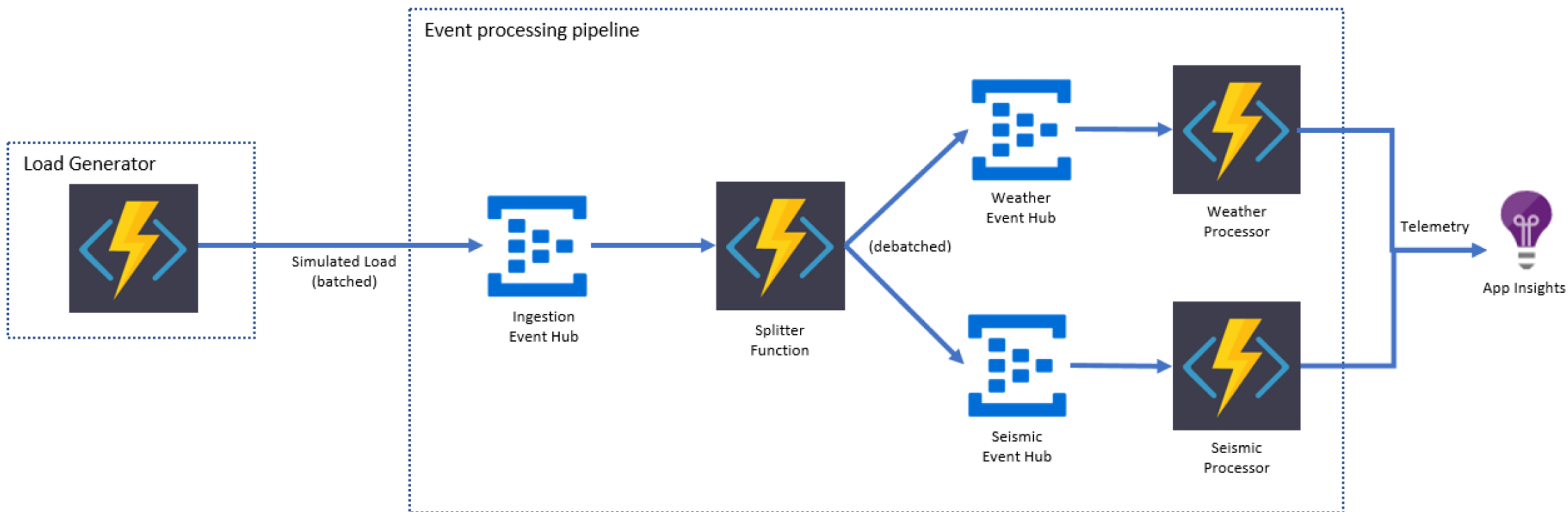


예시 : A 유통사 서버리스 예시

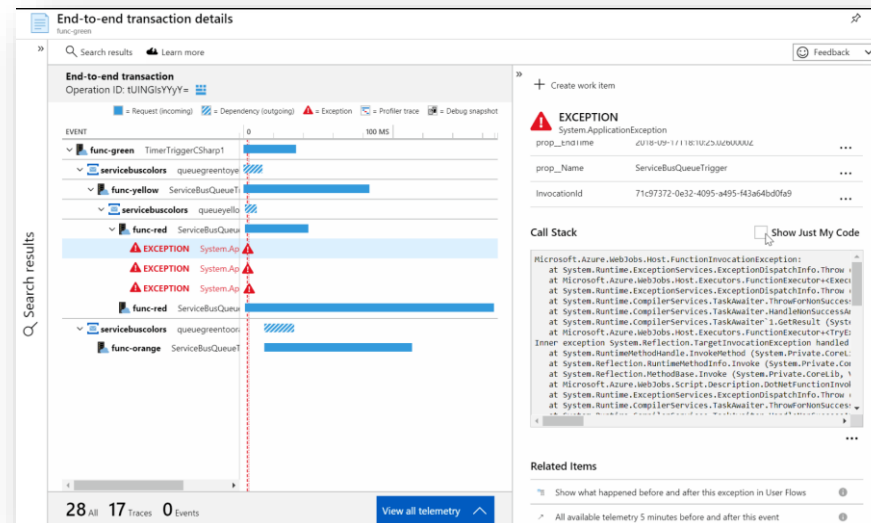
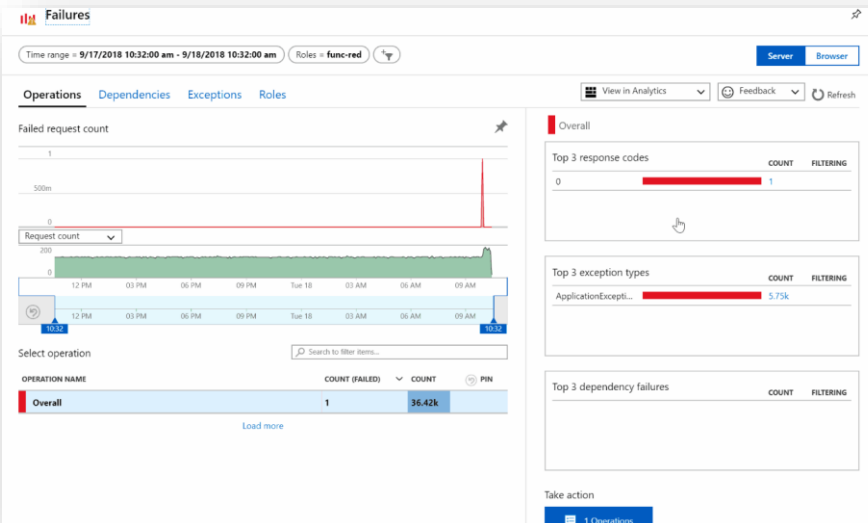
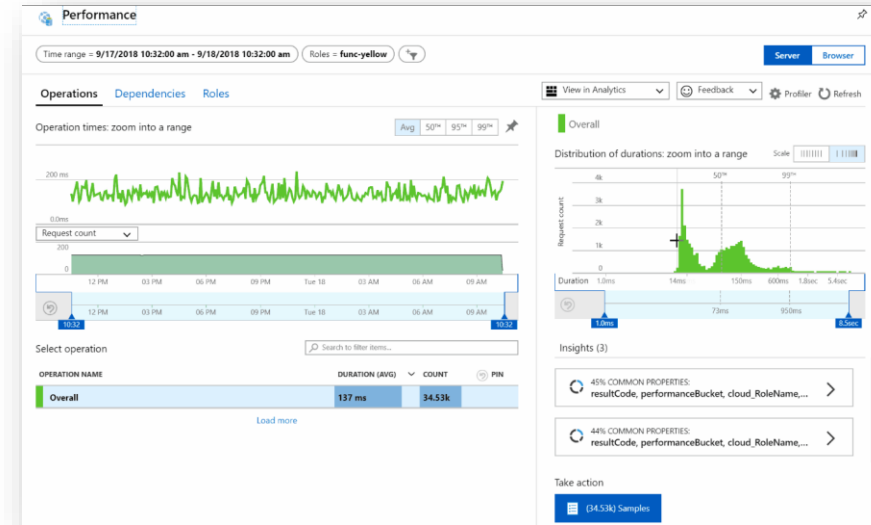
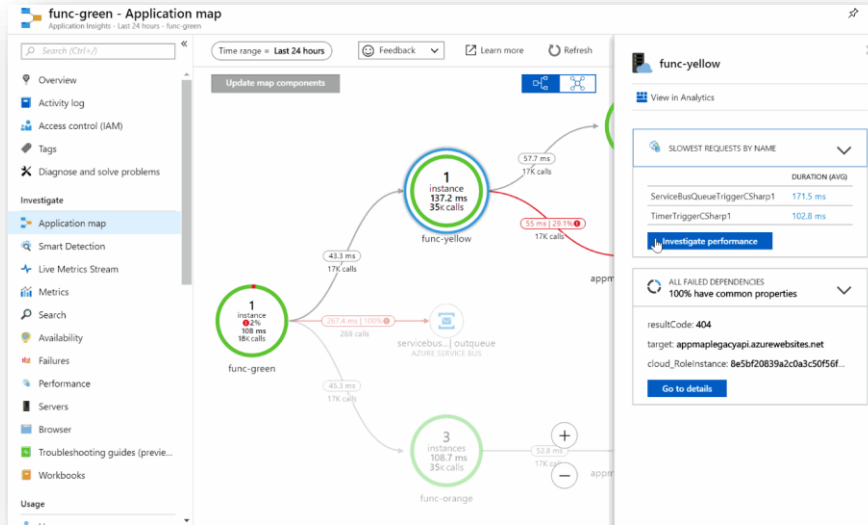


예시 : 초당 10 만건의 이벤트 처리 예시

Processing 100,000 Events Per Second on Azure Functions



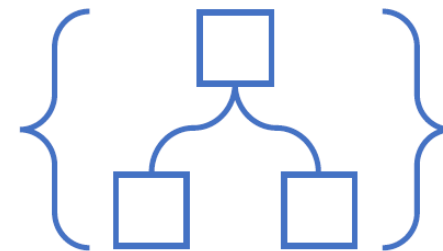
기본 제공 보안 및 모니터링으로 안정적인 서버리스 애플리케이션 만들기



Azure Logic Apps

Azure에서 제공되는 강력한 워크플로우 및 통합 엔진

- 비주얼 디자인어를 사용하여 트리거와 액션을 갖는 워크플로우를 빠르게 생성 및 통합
- 다양한 응용프로그램, 데이터, 서비스와 연결
 - Azure Functions와 연결 및 오케스트레이션



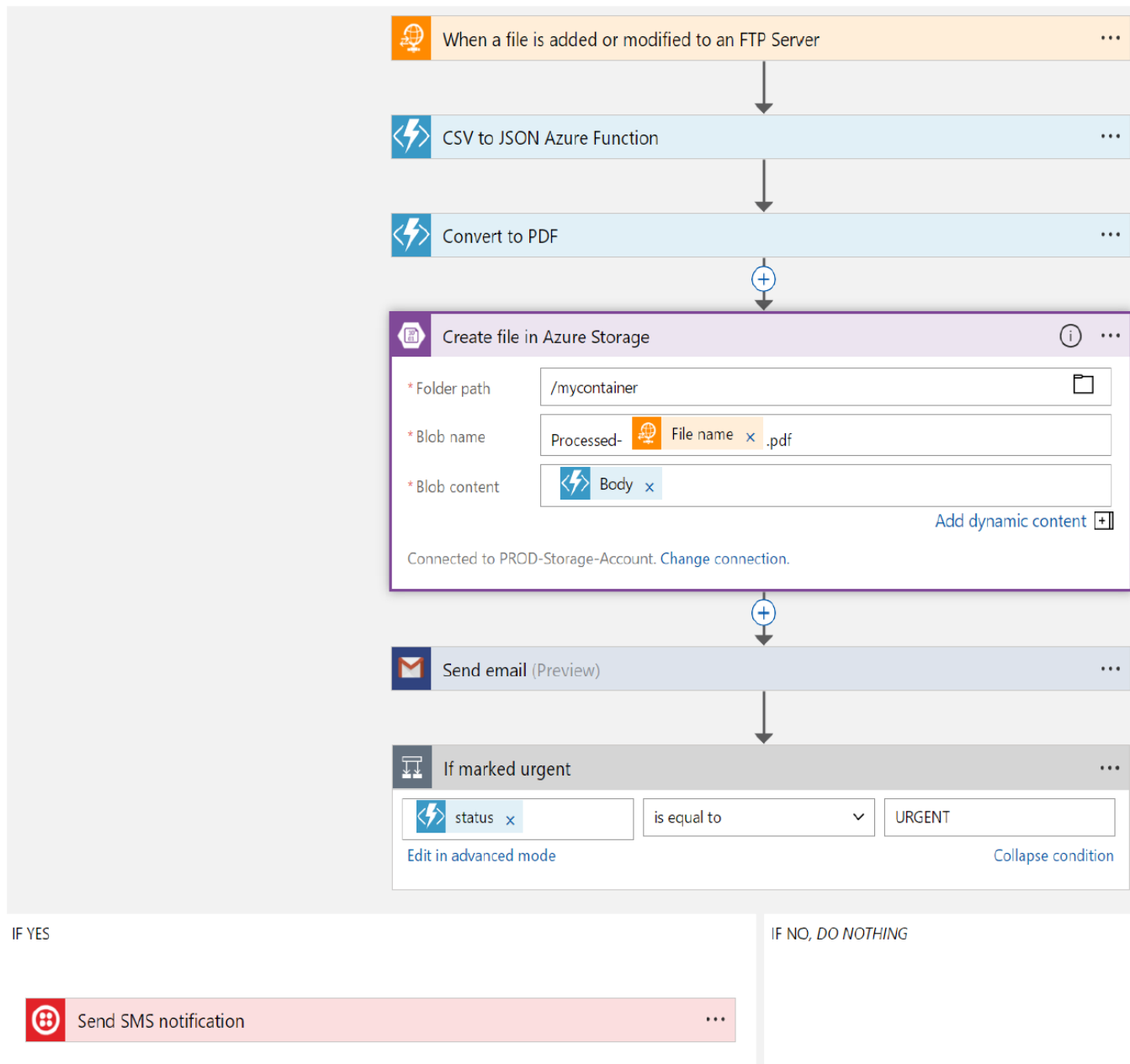
Logic Apps 워크플로우 디자이너

클라우드 기반 워크플로우

강력한 제어 흐름 제공

서로 다른 역할의
Functions과 API들을 연결

선언적으로 정의 가능. 소스
제어 체크인하거나 배포 시에
활용 가능



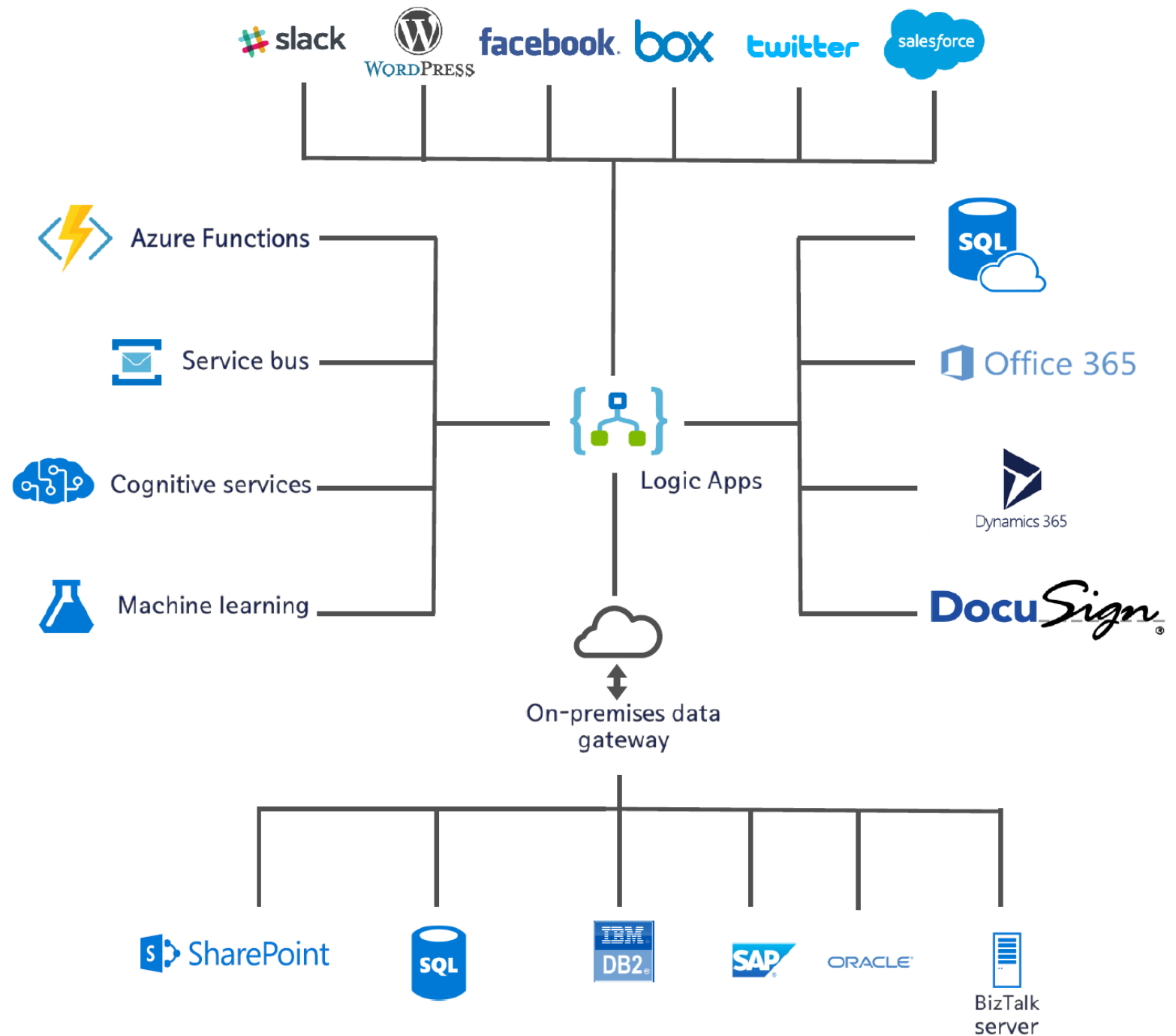
Logic Apps Connector

170개 이상 제공,
계속 늘어나는 중



.... and more!

Logic Apps connects everything



감사합니다

