

컨테이너라이제이션 (Containerization) 어디까지 왔나?

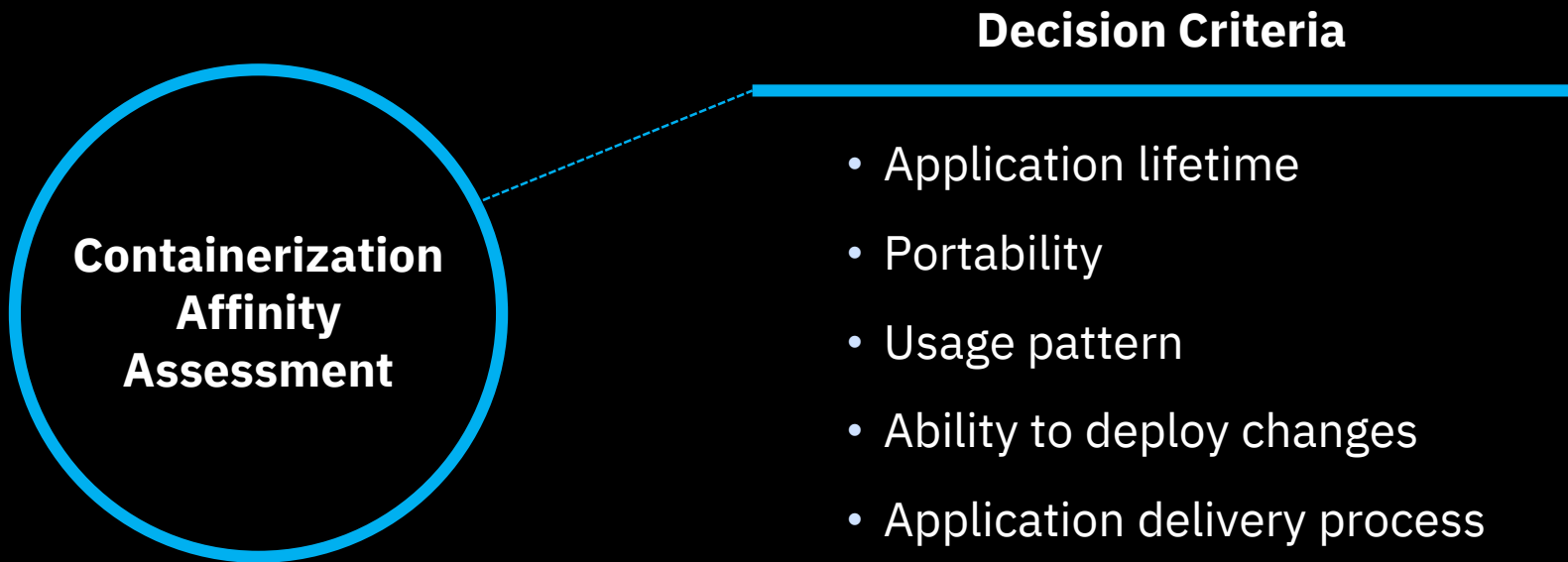
# 컨테이너라이제이션

## 집중 분석

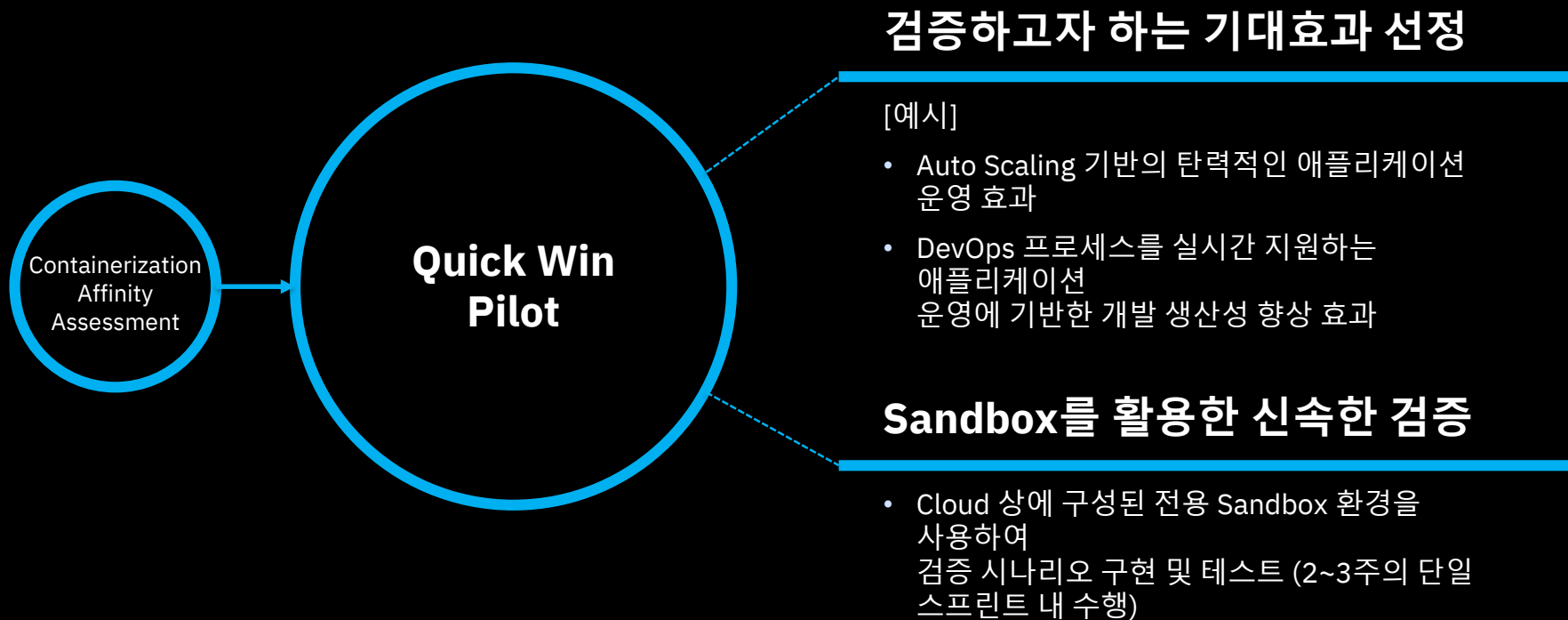
컨테이너라이제이션의 핵심 키워드와  
기술적 조연



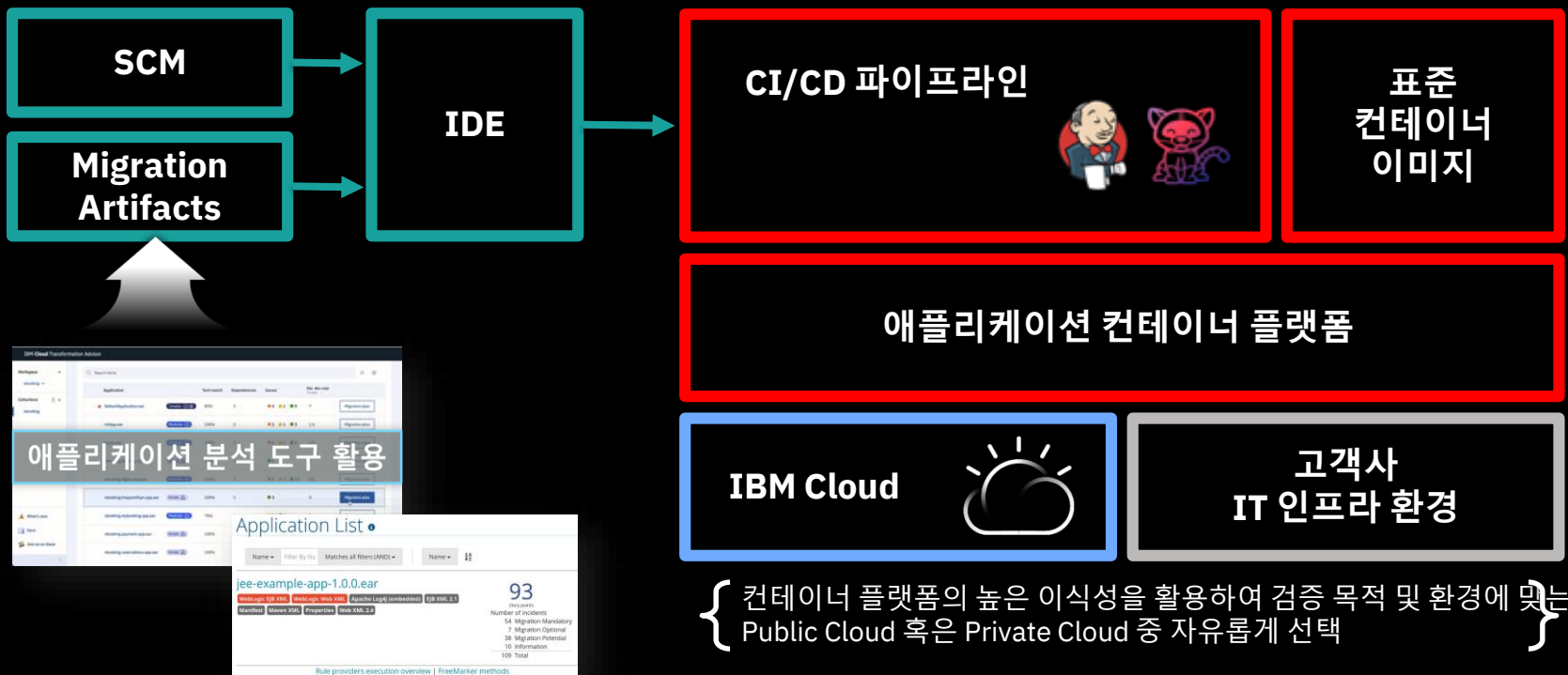
# Containerization 적용의 효율적인 시작을 위해...



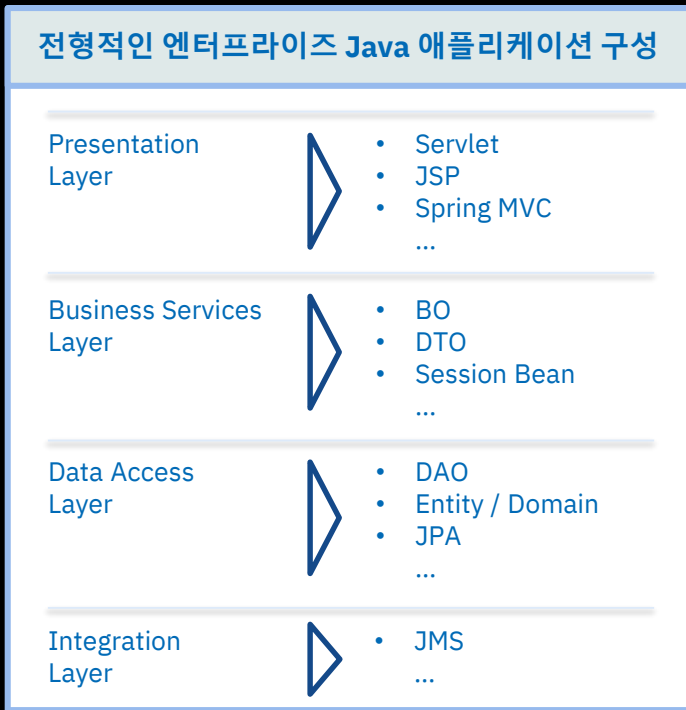
# Containerization 기대효과를 구체화하는 과정



# Pilot Sandbox는 어떻게 효과적으로 구성하는가?

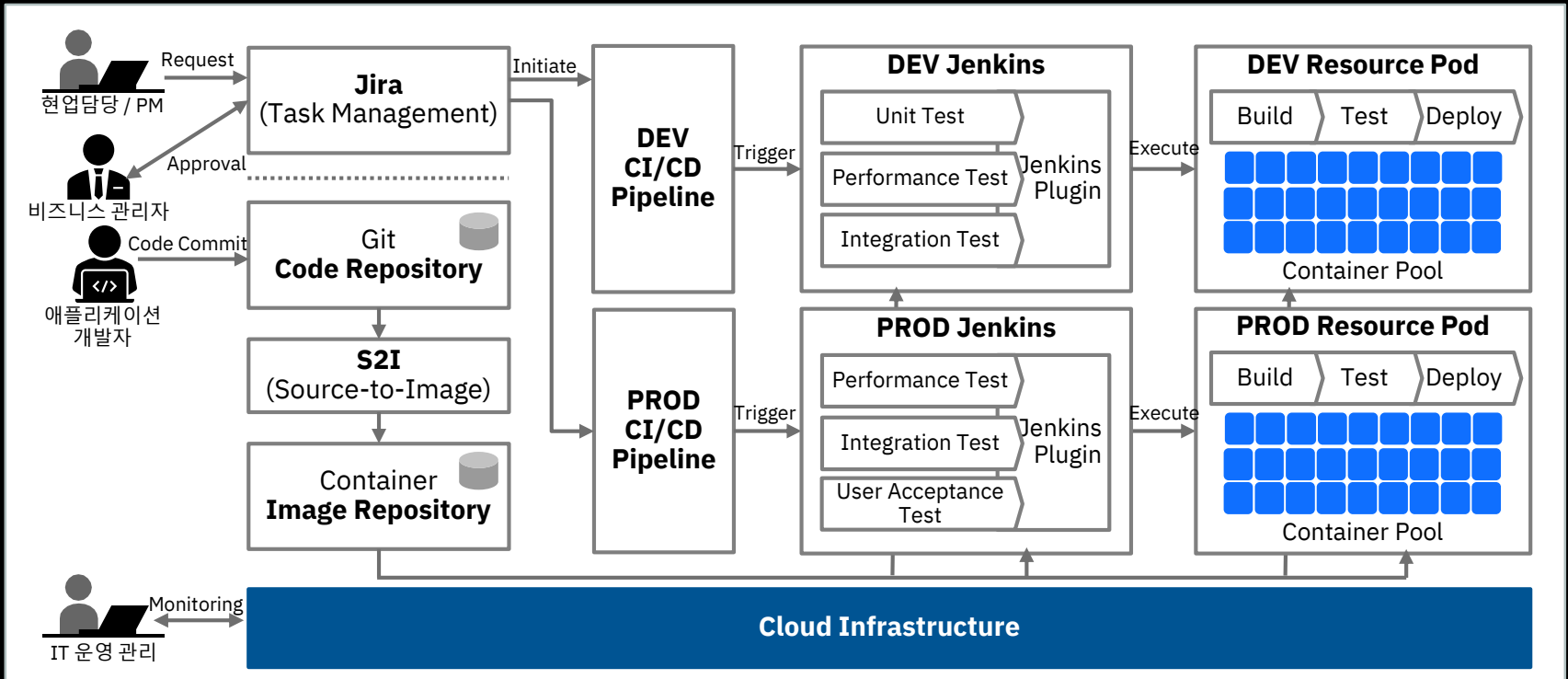


# Quick Win Pilot을 통해 확인한 Lessons Learned

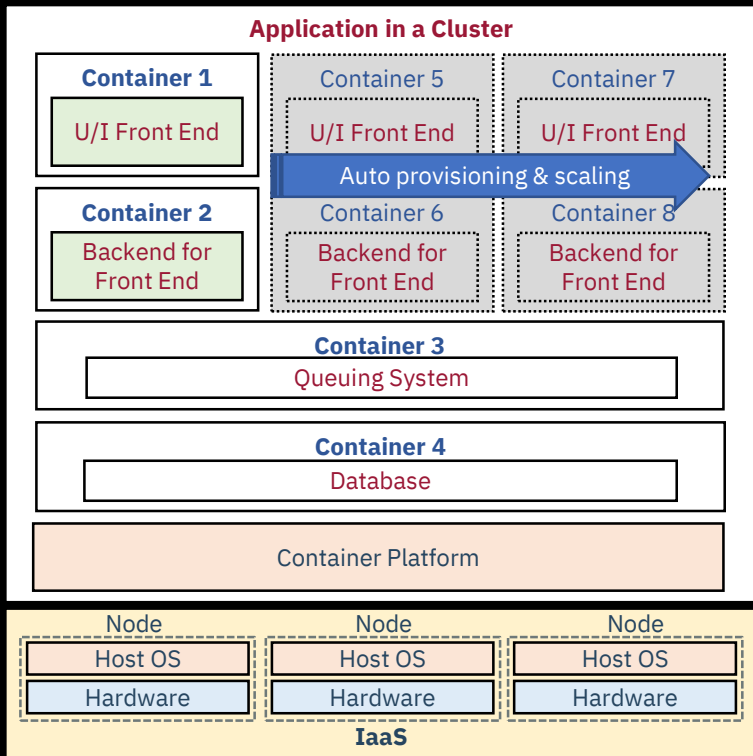


- 기존 엔터프라이즈 Java 개발 패키지 구조는 기술적으로 상호의존성이 없어도 전체 단일 패키지 형태로 개발/테스트/배포해오고 있음
- 상호의존성 기준으로 **단일 패키지를 작은 규모의 복수 개 패키지로 분화**하여 Container로 배포 가능
- **DevOps Pipeline과 Container 기반 실행 환경을 결합**함으로써 비즈니스의 요구에 따른 빠른 애플리케이션 변경, 개발 생산성 향상 기대

# Containerization 기반의 개발 생산성 강화



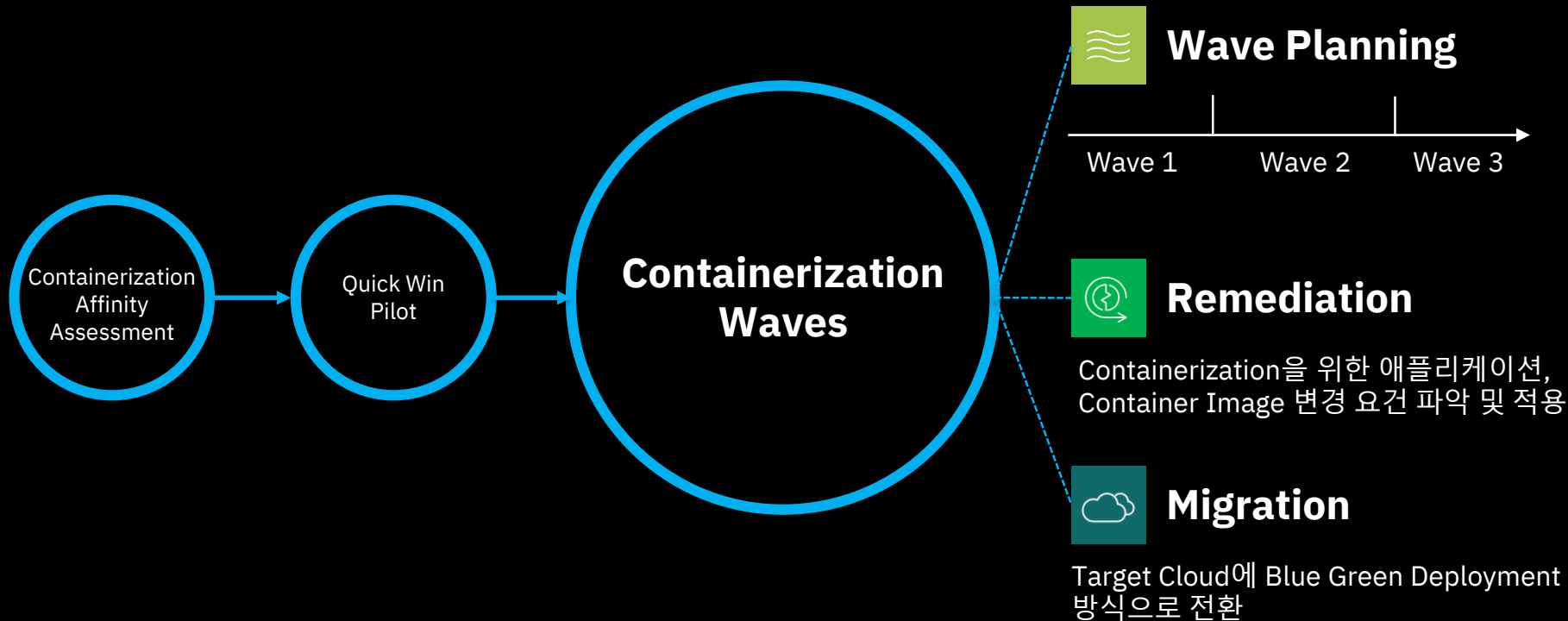
# Quick Win Pilot을 통해 확인한 Lessons Learned



## 주요 기대효과

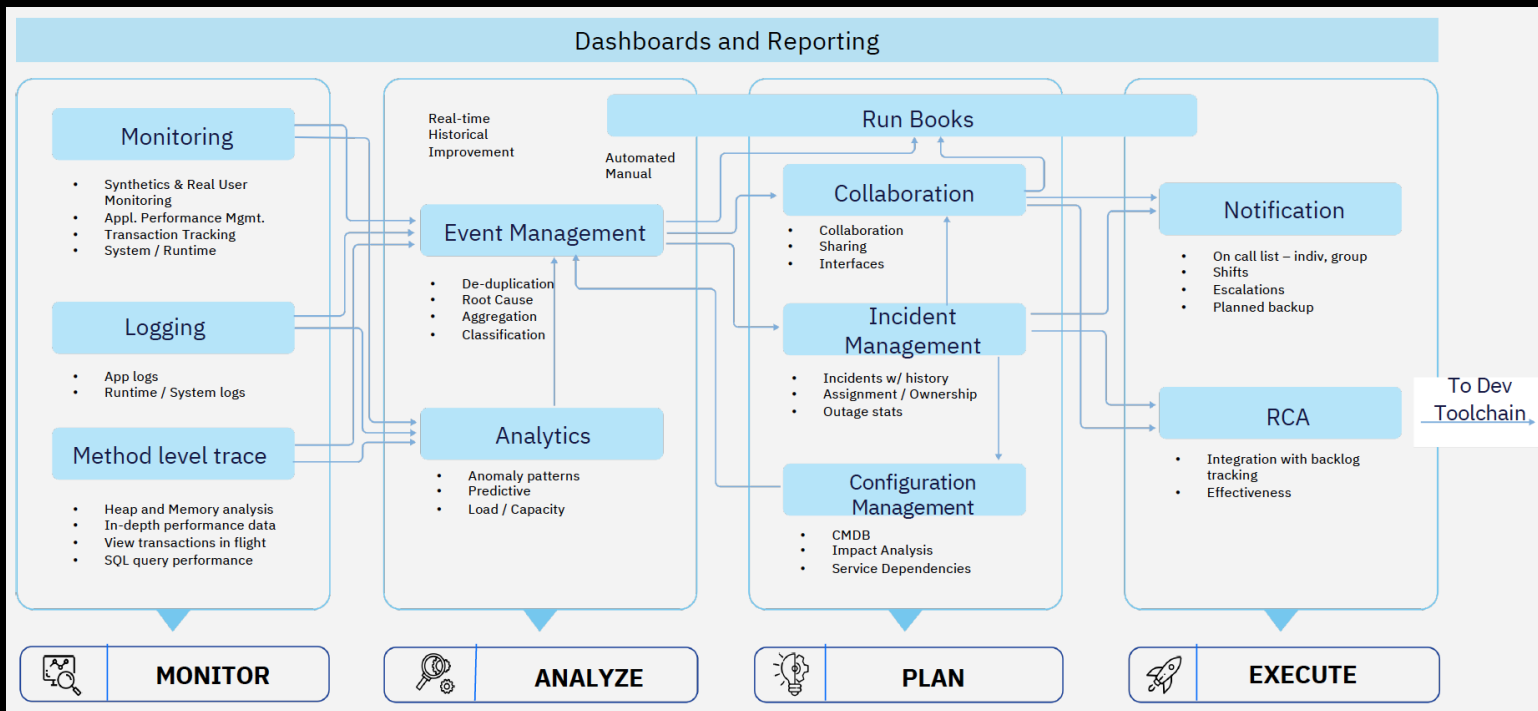
- 애플리케이션 배포/운영에 대한 **Automation** 적용으로 확장성 높은 구조로 전환
- 애플리케이션이 실행 환경에 귀속되지 않는 **이동성(Portability)** 측면의 개선 효과
- Container의 빠른 기동 시간의 장점을 살려 **신속한 애플리케이션 배포 및 장애로부터의 복구** 가능
- Tax와 같았던 상용 가상화 솔루션의 **라이선스 비용 부담 절감**
- **Infra 자원 투자/사용 비용 절감** 효과

# Containerization 기대효과 검증 이후 점진적 확산 적용

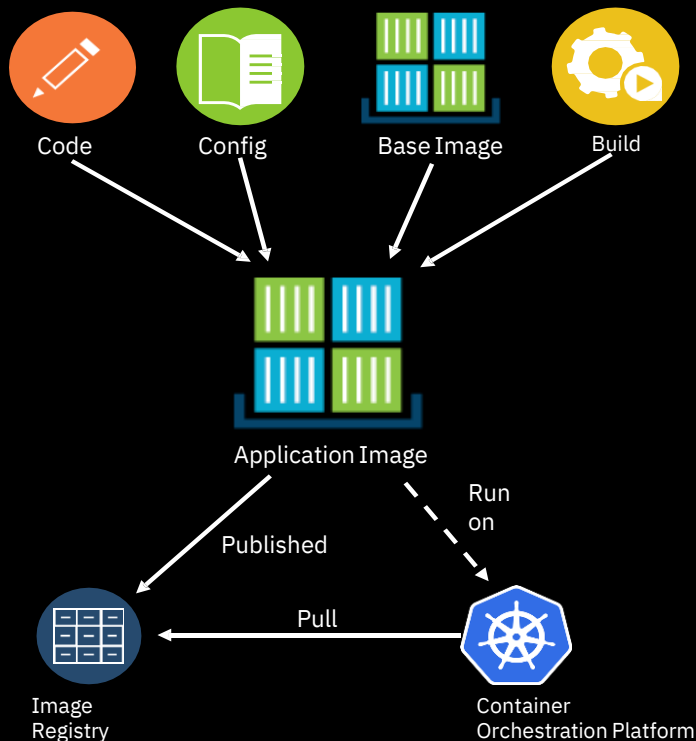




# Containerization 확대 적용에 따른 통합 운영 관리 모델 수립 필수적



# Container Platform 관점의 보안 체크 포인트



- 신뢰할 수 있는 Base Image를 사용하고 있는가?
- Container Image Registry 보안 관리 정책을 명확하게 정의하고 있는가?
- Import하는 Module 혹은 Library에 보안 취약점이 있는지 체크가 가능한가?
- 접근 권한 통제에 대한 관리 체계가 갖추어져 있고 실시간 모니터링이 이루어지고 있는가?
- 외부와의 연계를 위한 접속 포트 제한 및 관리가 이루어지고 있는가?
- ...

# Containerization 확산을 위한 보안 거버넌스 고려사항

## Image & Container Scan

---

Container Image의 보안 취약점 탐지 뿐만 아니라  
실행중인 Container를 대상으로 정책기반 보안 Scan 구현 필수적



## Health Check

---

Container 자체 및 Container Orchestration  
영역에 대한 CIS Benchmark에서 제시하는  
기준 준수 여부 체크 필요

# Thank you

진승의 상무  
Cloud Competency Center  
한국IBM

—

Seungeui.Jin1@ibm.com  
+82-10-4995-6124

