SK 주식회사 AX

AI DevOps on Azure with GitHub

GitHub Enterprise Cloud

2026.04

SK 주식회사 AX

Contents

현대 소프트웨어 개발 환경의 도전 과제

" 복잡해진 DevOps 환경과 개발자 생산성 문제 "

- 마이크로서비스, 멀티 클라우드, 오픈소스 의존성 증가로 개발 및 운영 환경이 더욱 복잡해짐
- 개발자들은 여러 도구와 잦은 컨텍스트 스위칭으로 인해 핵심 개발 업무에 집중하기 어려워 개발자 생산성 저하
- 개발 초기부터 보안은 필수적이지만, 전문 인력 부족과 도구 파편화로 인한 어려움으로 DevSecOps 정착에 어려움 존재
- 복잡한 환경과 내부 프로세스 이해에 수개월이 걸려 신규 개발자 교육으로 인한 신규 개발자 온보딩 지연 이슈

Agentic AI가 가져오는 개발 방식의 변화

" Agentic AI를 통한 자동화와 개발자 생산성 향상 "

- Agentic AI는 개발자의 의도를 이해하고 다양한 개발 작업을 보조하여 반복 업무 자동화와 개발 효율 향상을 지원
- 반복 작업을 AI가 처리함으로써 개발자는 문제 해결과 설계 등 고부가 가치 업무 집중
- 코드 작성 단계에서 취약점 탐지와 개선 가이드를 제공하여 개발 단계의 보안 내재화를 지원
- 팀의 표준과 정책을 반영한 작업을 AI 에이전트의 지원을 통해 조직 표준 준수와 개발 일관성 확보에 기여

GitHub Enterprise Cloud 특/장점

" Enterprise을 위한 AI, Cloud기반 통합 개발 플랫폼 "

- 인프라 유지보수 없이 최신 기능과 보안 패치를 제공하는 Azure Cloud 기반 확장성 보장
- SAML 기반 SSO, 고급 권한 관리, 접근 제어 및 감사 로그로 엄격한 보안과 규제 준수를 위한 Enterprise 보안/Governance 지원
- 수천 명의 개발자가 글로벌 환경에서 동시에 협업하고 복잡한 조직 구조를 유연하게 관리할 수 있는 대규모 협업 지원, AI 도구 도입과 신속한 실험을 가능하게 하여 빠른 변화 대응이 가능한 디지털 혁신과 민첩성
- Copilot과 함께하는 DevSecOps 플랫폼 제공
  - 코드 작성부터 보안, 협업, 자동화까지 일관된 경험을 제공
  - 이슈 작성, 코드 리뷰, 테스트 생성 등 다양한 작업에 Copilot과 통합
  - 개발 단계에 자연스럽게 통합되어 DevSecOps 성숙도 빠르게 향상
  - 단일 플랫폼으로 학습 곡선을 완화하고 도구 간 연계 문제를 최소화

비즈니스 성과와 고객 가치

"Agentic AI, SDLC 전 과정 생산성·품질·보안 동시 향상 "

- **개발자 생산성 향상 :** 약 22%
  . GitHub 기반 통합 DevSecOps는 전체 개발 주기를 단축
- **신규 개발자 온보딩 단축 :** 최대 80%
  . 표준화된 워크플로우와 협업 기능, 자동화 활용
- **도구 관리 효율화 :** 약 75% 관리시간 감소
  . GitHub 통합으로 분산된 개발 및 보안 도구 관리 비용 절감
- **보안 리스크** 감소
  . 자동화된 보안 대응과 조기 취약점 발견으로 보안 사고 위험과 준수 비용 절감

고객의 장애 요인을 구체화 후, ACM(Adoption & Change Management)와 Pilot으로 GitHub + Copilot 확산을 조기 지원

"왜 GitHub + Copilot 도입이 어려운가 ? "

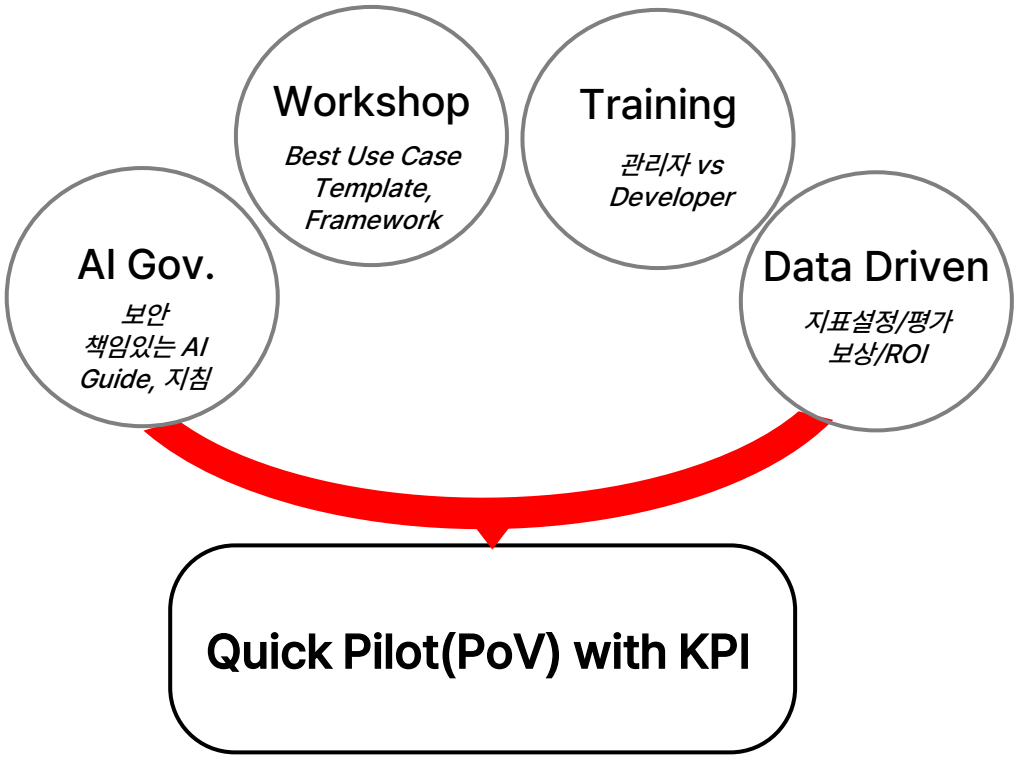
기업의 전사적 App. DevOps 과정에 AI 도입 장애 사유

- 개발자의 심리적 거부감 여전히 존재
  - "내가 Coding은 잘해", "AI가 나를 대체하나" 같은 불안, 직업 정체성 이슈
  - 초기 부족한 Coding Agent의 성능 경험(장시간 디버깅)으로 AI에 대한 신뢰
  - "나는 잘 모르는데..", AI를 언제, 어디까지 사용하는지에 대한 가이드 부족
  - AI 적용 후 발생하는 실수와 책임에 대한 우려 존재

- Enterprise는 제한적, 소극적 AI 도입 현실
  - AI 프로젝트 실패 사례("AI 거품론"), ROI 불명확(License, Training 등)
  - 많은 협력사, 조직간 이해관계가 복잡하여 일괄 적용은 시간, 노력에 부담감
  - Data 유출(Source Code유출), AI 오류 및 책임 소재가 불명확
  - 경영층의 적극적 후원, 지원이 없으면 AI 도입은 추가 업무로 인식하는 한계

" Enterprise에 AI 적용을 위한 Approach "

Adoption & Change Management



고객의 분산된 Azure 기반 App. 개발환경을 GitHub Enterprise를 이용하여 전사적 통합관리, AI 활용, 전환 Tool 및 Data, 보안을 고려하여 전환 계획을 수립/지원

1단계) 소스만 GitHub 전환 Hybrid 안정 전환

Azure DevOps 유지 + GitHub
(Source Code 이전/관리)

- 대상 : Azure DevOps 또는 3rd Party DevOps이용, 초기 전환 Risk 최소화
- Hybrid 구성
  - Azure DevOps : Planning, CI/CD 등
  - GitHub : PR/코드리뷰/Repo
- 장점
  - 소스 호스팅으로 에이전틱 기반 확보
  - CI/CD 무변경으로 Risk/중단 최소화
  - GHEC에 Azure DevOps 기본 License 포함
  - Data Residence 대응
- AI/Agentic
  - Copilot Chat/IDE 일부
  - Repo 전용 기능

2단계) Copilot, GitHub Model로 AI 생산성 가속화

GitHub 기반 Repo + AI
(AI 기반 개발 생산성 확대)

- 대상 : AI 생산성 효과 빠르게 활용 App.현대화 기술 부재 해소
- Hybrid 구성
  - Azure Boards 또는 혼합 운영
  - GitHub : Agent Mode/코딩 Agent
- 장점
  - 계획, 코딩, Test 전 과정 AI 활용
  - GHAS, Copilot Autofix 확대
  - 개발 표준 AI 학습, 자동 수정 지원
  - Data Residence 대응
- AI/Agentic
  - Copilot/코딩Agent 기반 자동화
  - GitHub 비용/성능 검증 후 Azure AI Foundry로 손쉬운 확장

3단계) GitHub 중심으로 Agentic DevOps 극대화

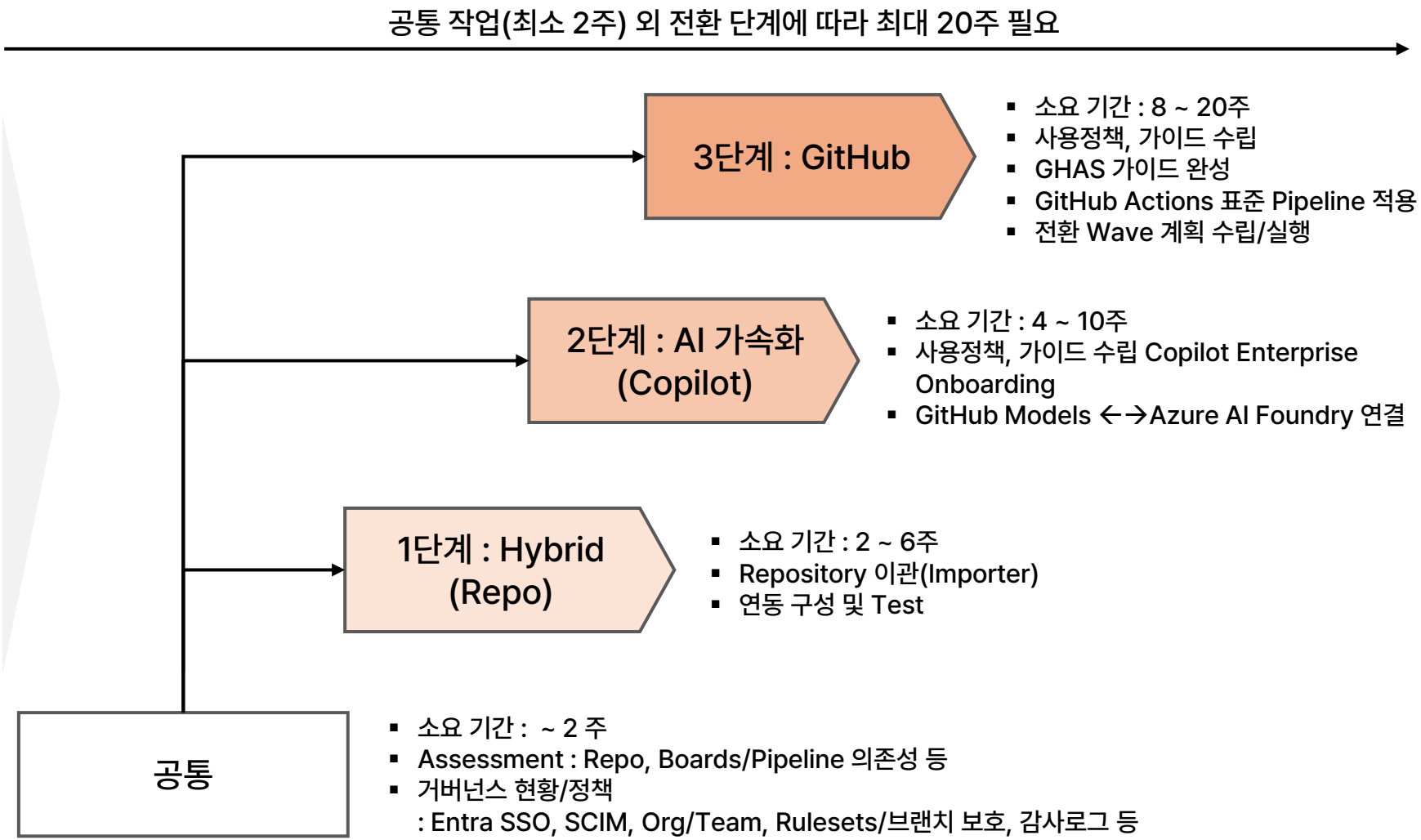
GitHub 기반 단일 환경
(End-to-End 환경)

- 대상 : DevSecOps/AI를 엔드투엔드 표준화하려는 조직
- GitHub 기반 통합 구성
  - Azure 단계적 종료
  - GitHub Enterprise 전면 도입
- 장점
  - GitHub Actions, Agentic 최대 활용
  - 보안 신기능 최우선 적용(GHAS) (Autofix, Secret 보호, 의존성 검사 등)
  - GHAS, Copilot Autofix 확대
  - CI/CD 표준화, Template 활용
- 고려사항
  - 조직 변화 관리/거버넌스 조직 필수
  - Migration Plan/실행

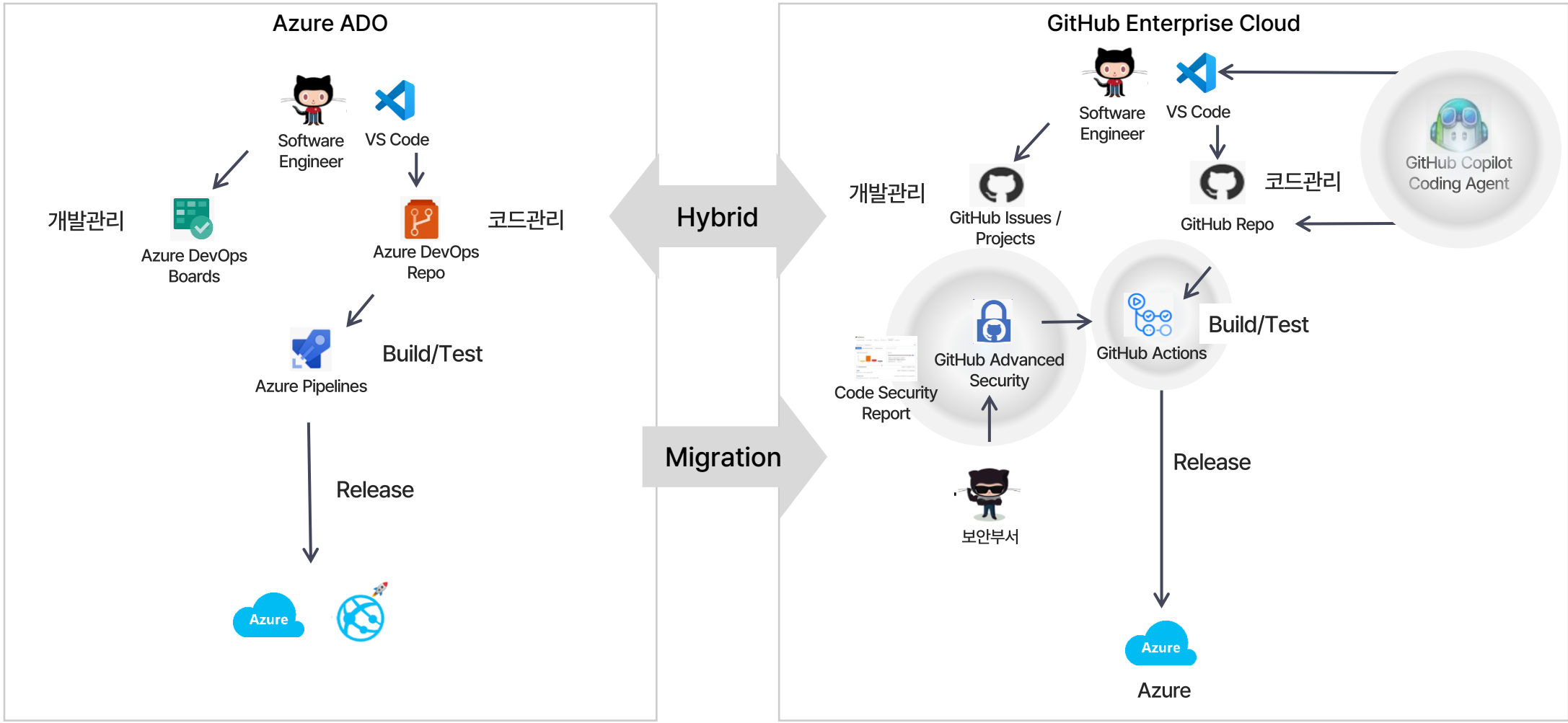
Enterprise 고객의 목표 확인 후 사전 Assessment를 통해 Migration Plan 수립 & 실행

사전 고려 /의사결정 사항

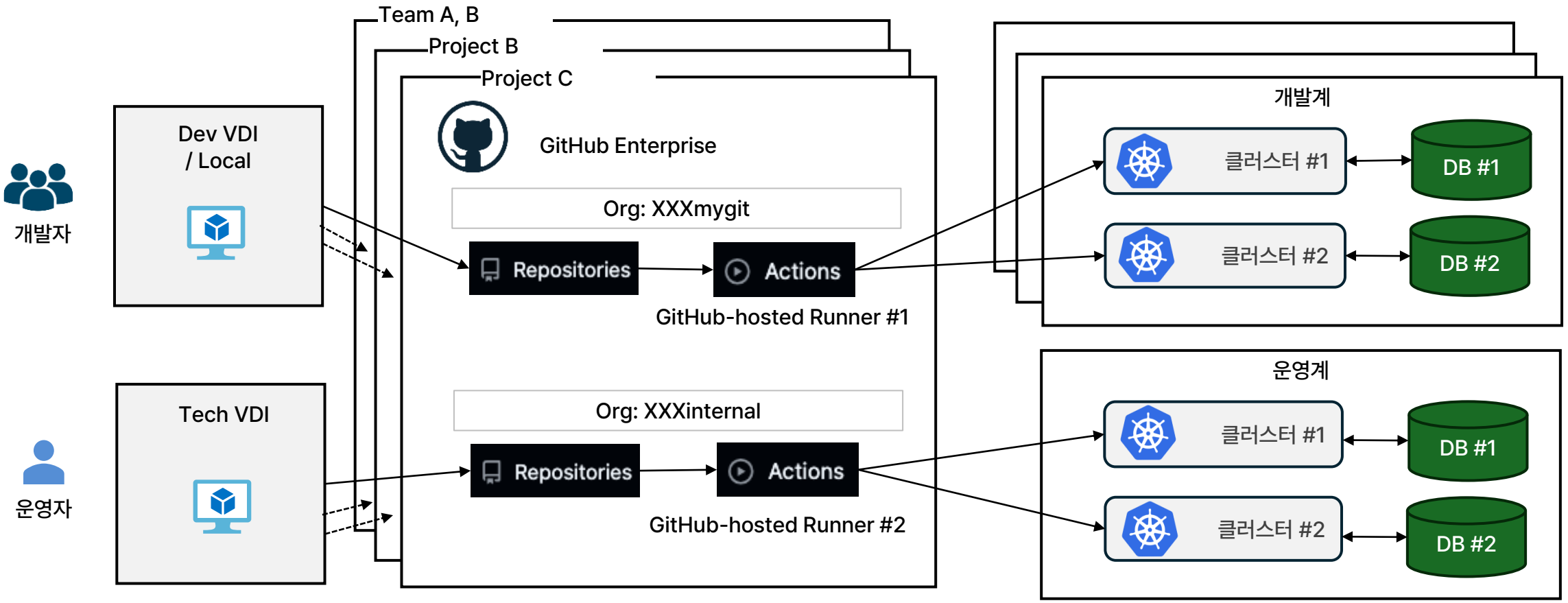
- 기업의 AI/DevSecOps 목표 설정
  - 단기 리스크 최소화 : 1단계
  - AI 생산성 빠른 적용 : 2단계
  - End-To-End 적용 : 3단계
- 주요 고려사항
  - 이관 복잡도 : 조직, 히스토리
    . 자체 개발, 3rd Party(협력사)
  - 보안/Compliance 기준/통제 수준
  - 변화관리 : 보유 역량 /기술 수준
  - 목표 수준/Guide 명확화
    . 명확한 KPI 설정
    . 생산성 지표, 취약점 처리, 자동화율



Azure DevOps를 사용하는 고객은 쉽고 간단하게 GitHub을 통해 AI(Copilot) 기반 개발 생산성을 높이고 표준화된 DevSecOps 환경으로 전환

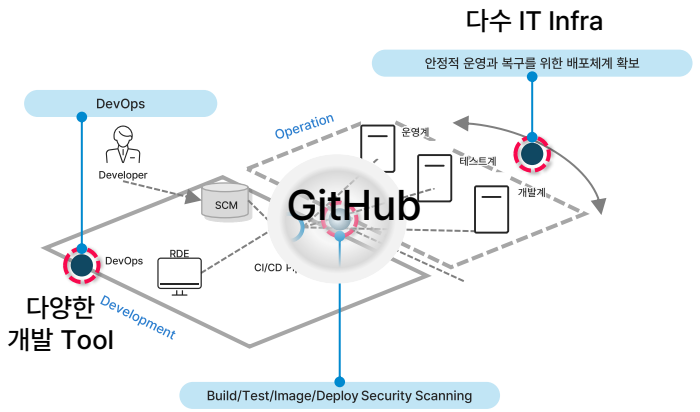


Enterprise의 전사 Repo는 기본 단위를 개발(xxxmygit)과 내부용(xxxInternal)으로 분리 및 GitHub Actions을 통한 개발/운영계 서비스를 개별 배포관리



다수 Project의 다양한 개발 Tool과 여러 Infra 구성 환경을 대응하기 위해 GitHub 기반으로 App. 개발 체계를 표준화해 App.의 개발기간 단축, 보안&품질 향상

GitHub 중심 전사 Level 개발/운영 표준화



A사 고객 Benefits

개발 환경 표준화	• 개발 소스 저장소 표준화 • GitHub Actions 기반 배포 파이프라인 표준화 • 일관된 개발·배포 프로세스 정립
개발 지식 공유	• Wiki 기능을 통해 quick하게 지식공유 • 다양한 언어(java, typescript, python) 및 산출물의 협업 체계 강화 • 프로젝트 협업 도구
전사 정책 및 가이드 라인	• 보안정책 수립 및 일관된 적용 • Actions Runner, 빌드 환경구성 등 일관된 구성 가이드 제공 • Inbound/Outbound 접근제어를 통해 보안강화

[참고] GitHub Service 기능 비교

Enterprise Capability	GitHub Enterprise Cloud	GitHub.com	GitHub Enterprise Server
Data residency	✓		Self-managed
Core productivity & collaboration features	✓	✓	✓
Enterprise managed identities	✓	✓	✓
Private unique namespace	✓		Self-managed
Always up-to-date (code updates to GitHub)	✓	✓	
Platform security features	✓	✓	✓
Azure Availability Zones & Multi-region BC/DR	✓		Self-managed
Copilot Business: code completion + chat	Add-on	Add-on	Add-on
Copilot Enterprise features	Add-on	Add-on	
GitHub Actionss	Self-hosted + GitHub hosted *	Self-hosted + GitHub hosted	Self-hosted only
GitHub Packages	✓	✓	✓
GitHub Advanced Security	Add-on	Add-on	Add-on
GitHub Connect	✓	✓	✓
GitHub Codespaces	Future	✓	
Advanced/regional privacy certifications	Future		
SOC1, SOC2, Type II reports annually	✓	✓	N/A

** Excludes GitHub-hosted Mac runners*

SK 주식회사 AX

THANK YOU