

GitHub App과 AWS Lambda를 활용한 ChatOps 자동화 시스템

배포, 모니터링, 비용 관리까지 하나의 자동화 흐름으로




Morgan Bot Project

기획 변경

원래 목표

인프라 프로비저닝까지 챗봇에서 직접 처리

 IAM, VPC, EC2 자동 구성


한계점

입력값이 많고 검증 로직이 복잡함

 보안 고려 요소가 다수 필요

전환

"실행형 자동화" 중심으로 기획 수정

 배포, 모니터링, 비용 관리



Provisioning

인프라 구축 중심



Automation

운영 자동화 중심

구축형 → 운영형 자동화로 전환

IAM, VPC, EC2 프로비저닝보다 배포 자동화, 상태 모니터링, 비용 추적에 집중

빠른 피드백과 운영 효율성 향상을 위한 기획 방향 전환

ChatOps란?

Chat + Operations = 채팅 플랫폼에서 개발·운영을 제어하는 방식

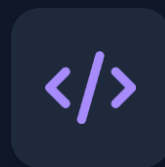
명령어 기반으로 코드 배포, 모니터링, 알림 등을 수행하는 DevOps 확장 모델



Chat



Command



API



Result

장점

명령과 결과가 팀 채널에 공유되어 투명한 협업

개발자 친화적 인터페이스로 효율성 향상

즉각적인 피드백과 자동화된 실행 가능

ChatOps 사례



GitHub + Slack

배포 트리거

코드 리뷰

PR, Issue 이벤트가 Slack에 자동 알림되고 배포 작업을 Slack에서 직접 트리거



Teams + Azure

빌드 관리

릴리즈 관리

Teams 채널에서 Azure DevOps 파이프라인 관리 및 실시간 빌드/배포 상태 확인



Discord + Custom Bot

배포 자동화

상태 모니터링

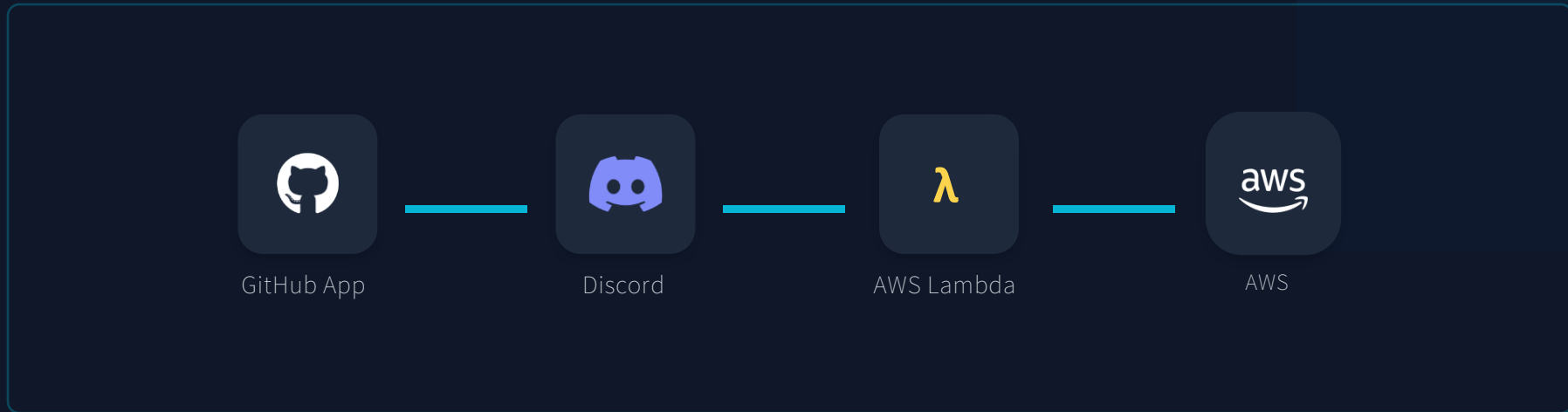
비용 관리

본 프로젝트: 슬래시 커맨드로 DevOps 라이프사이클 자동화

플랫폼별 ChatOps 기능 비교

플랫폼	GitHub + Slack	Teams + Azure	Discord + Bot
코드 알림	✓	✓	✓
배포 트리거	✓	✓	✓
인프라 모니터링	—	✓	✓
비용 관리	—	—	✓

시스템 개요



하루의 개발 운영을 자동화하는 ChatOps 시스템

사용자 명령 → 자동화 처리 → 즉각적 피드백

서버리스 아키텍처로 유지보수 최소화

GitHub App

GitHub API와 Actions를 안전하게 제어하는 공식 통합 앱



JWT 인증

안전한 API 접근 인증



**Repository
Webhook**

이벤트 기반 알림 트리거



**Workflow
Dispatch**

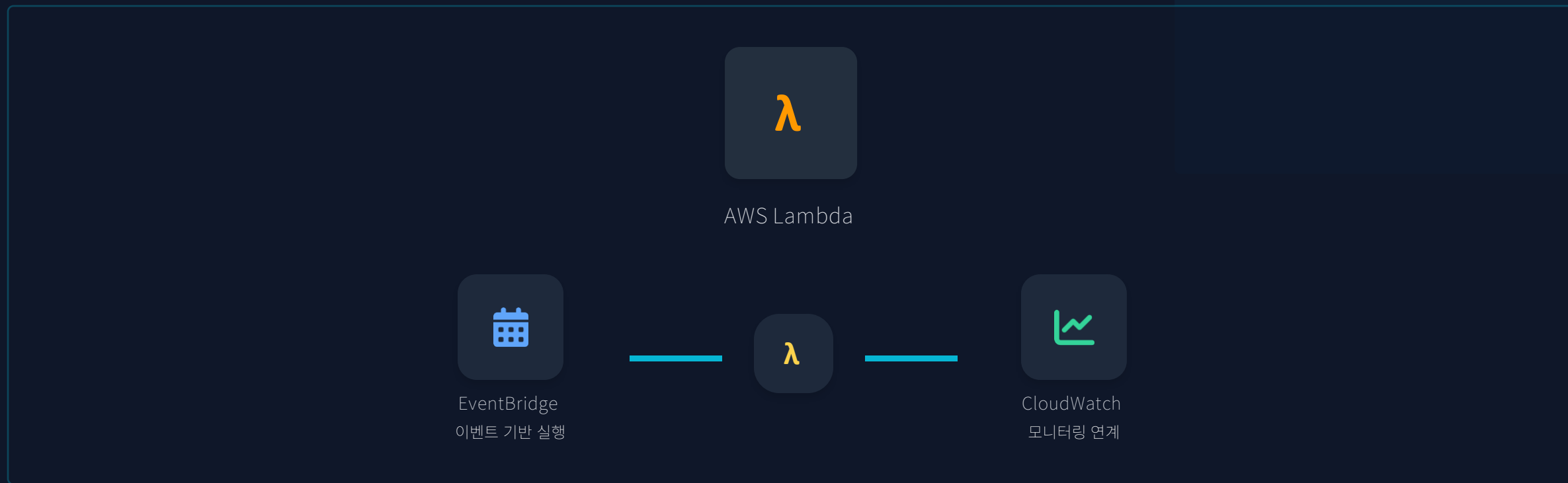
원격 액션 실행 트리거

GitHub API를 안전하게 호출할 수 있는 인증/권한 관리 시스템

외부 시스템에서 GitHub Actions 실행 및 저장소 이벤트 수신 가능

AWS Lambda

서버 없이 코드를 실행하는 서버리스 컴퓨팅 서비스



이벤트 기반 실행

API 호출, 파일 변경, 스케줄 등 이벤트에 반응



AWS 서비스 연계

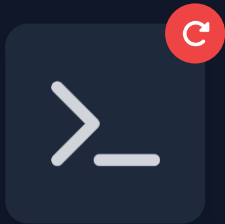
CloudWatch, EventBridge와 원활한 통합



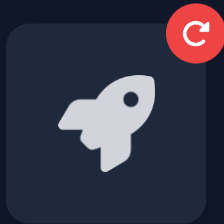
관리 부담 최소화

서버 프로비저닝 없이 자동 확장/관리

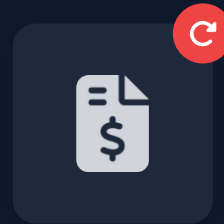
문제 인식



콘솔 반복 접속
다수 서비스 관리 불편



수동 배포
많은 단계 & 에러 가능성



비용 확인
예상치 못한 비용 발생

목표: 명령 한 줄로 운영 자동화

채팅 인터페이스로 운영 작업을 간소화하여 개발에 집중

운영 업무 효율화 및 휴먼 에러 감소

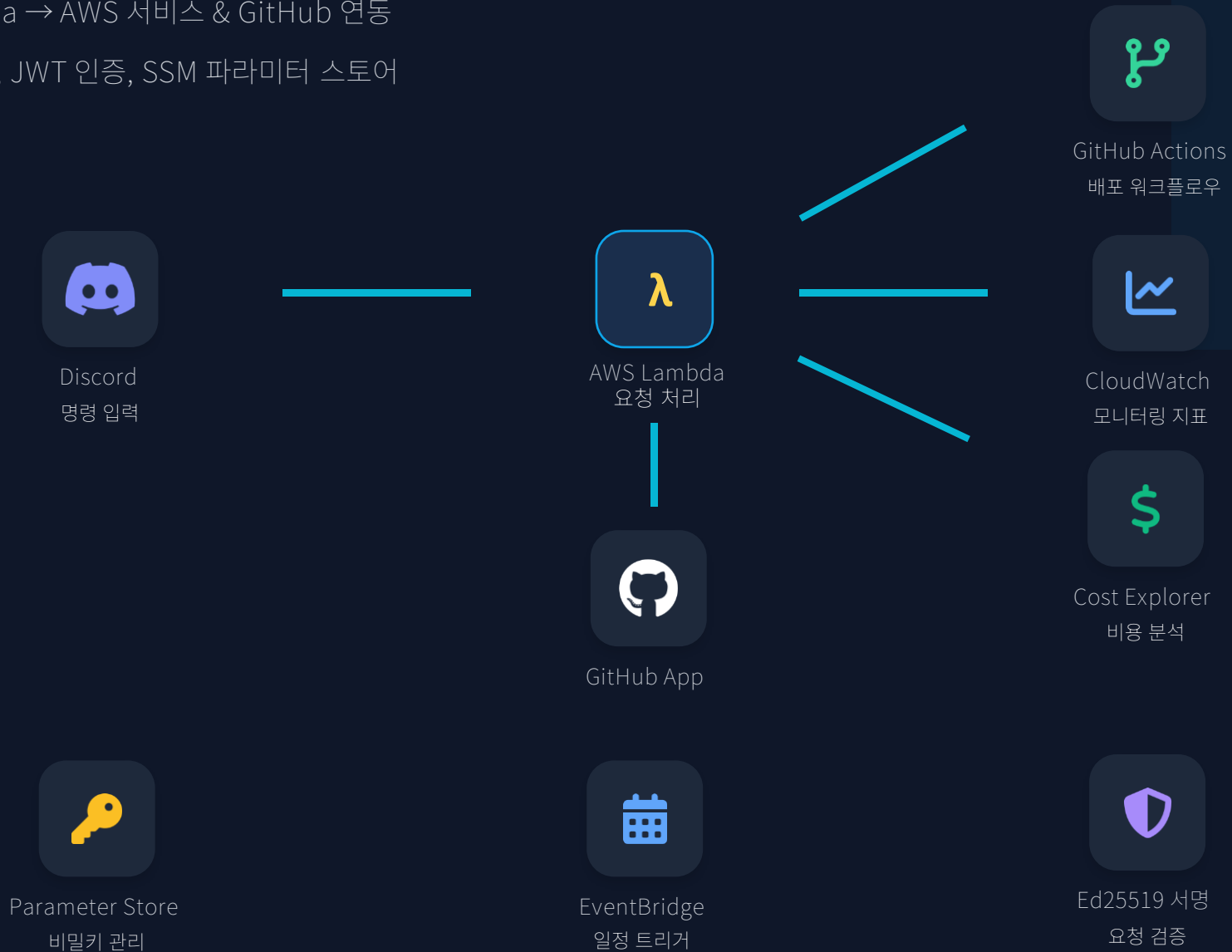
팀 전체가 동일한 인터페이스로 작업 수행

Discord 내에서 모든 알림 및 작업 통합

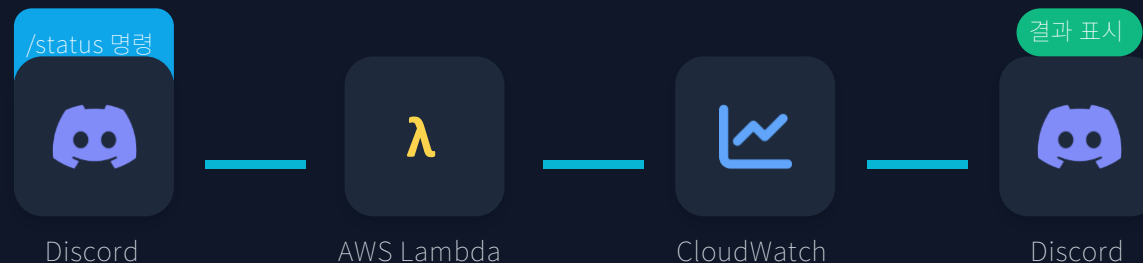
전체 아키텍처

Discord 명령 → Lambda → AWS 서비스 & GitHub 연동

보안: Discord 서명 검증, JWT 인증, SSM 파라미터 스토어



기능 1 - /status



EC2/ECS/EKS의 CPU 및 메모리 지표를 Discord에서 명령어 한 줄로 확인

콘솔 접속 필요 없이 실시간 모니터링

AWS 콘솔에 로그인하지 않고도 Discord에서 즉시 시스템 상태 확인

한 번의 명령으로 여러 리소스 상태 확인

EC2, ECS, EKS 등 다양한 리소스의 CPU 및 메모리 지표를 한 번에 조회

팀 공유 및 협업 용이성

공유 채널에서 실시간 상태 공유로 팀 전체가 동일한 정보 확인

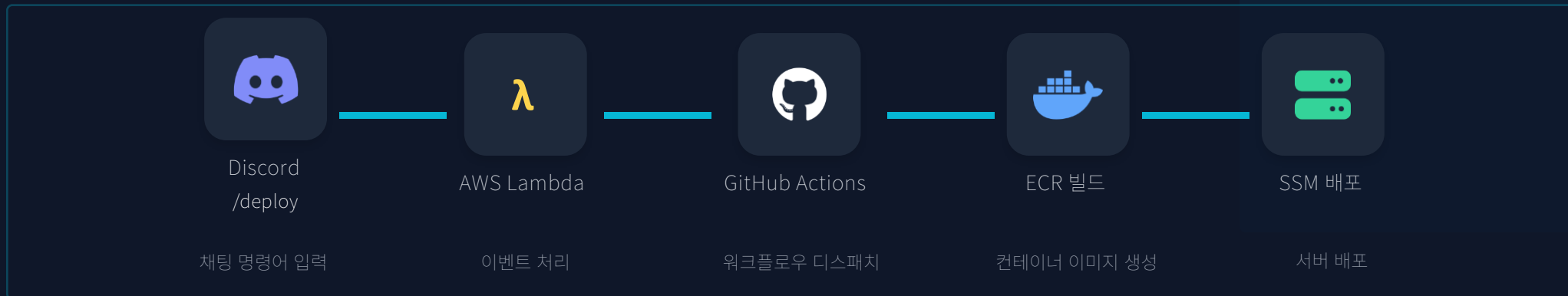


/status 명령 응답 예시

기능 2 - /deploy

간편한 배포 자동화

단 한 줄로 서비스 배포 완료

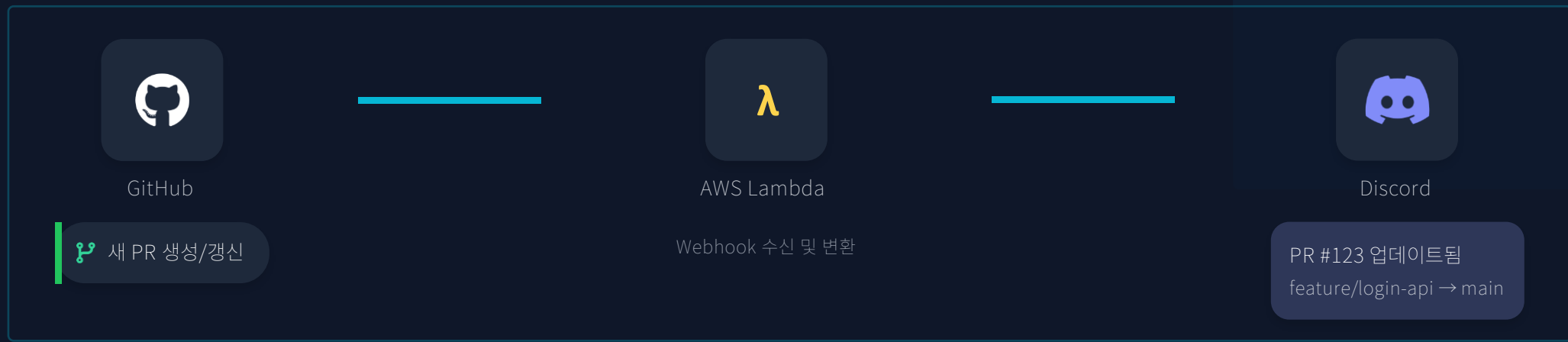


GitHub Actions를 통한 지속적 통합 (CI) 자동화
배포 과정 실시간 알림 및 결과 피드백
콘솔 로그인 없이 채팅에서 직접 배포



기능 3 - PR 알림

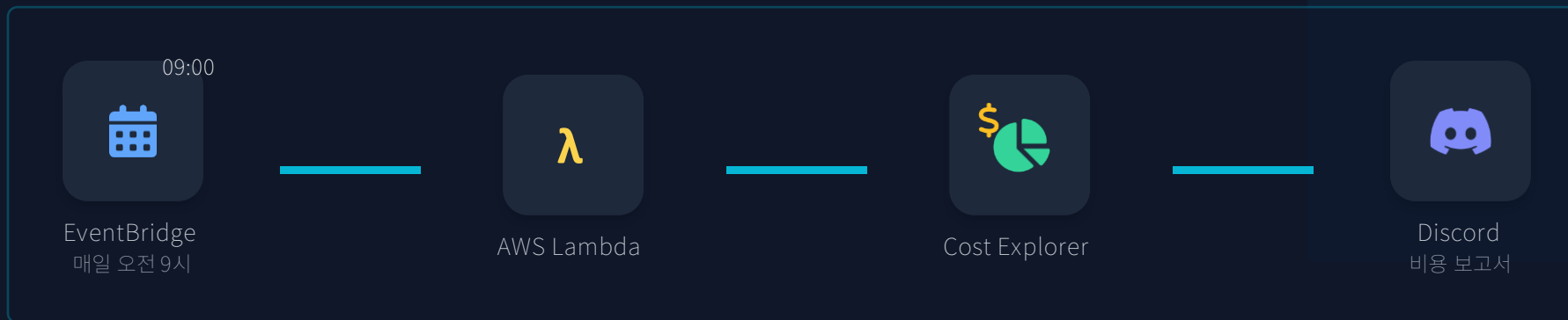
GitHub → Lambda → Discord 단순 흐름으로 코드 리뷰 효율성 향상



팀 협업 속도 향상, 코드 리뷰 누락 방지
클릭 시, 바로 해당 PR 페이지로 이동



기능 4 - 비용 알림



매일 아침 Discord 채널에 누적 비용 자동 알림

📅 실시간 비용 추적 및 예산 관리

🔔 비정상적인 비용 발생 시 조기 경고

🕒 일간/주간/월간 비용 트렌드 가시화

👥 팀 전체가 비용 인식 공유

 알림봇  오전 10:26

 **AWS 월 누계 비용 (MTD) 2025-10**

기간(MTD): 2025-10-01 ~ 2025-10-17

Top Services (USD):

- EC2 - Other: \$49.55
- Tax: \$5.82
- Amazon Route 53: \$5.79
- AWS Glue: \$2.80
- AmazonCloudWatch: \$0.00

 **누계 합계(MTD): \$63.97 USD**

오늘 오전 10:26

보안 및 관리 포인트



SSM Parameter Store

안전한 API 키 및 비밀정보 관리

암호화된 저장소



Discord Ed25519

커맨드 요청 무결성 검증

서명 기반 인증 시스템



GitHub JWT 인증

GitHub App 통합 인증

시간 제한 토큰 방식

모든 인증 키는 AWS 관리형 저장소에 안전하게 보관

다중 레이어 보안으로 시스템 무결성 확보

성과 및 기대 효과



Discord 통합

모든 운영 알림이 Discord 한 곳으로 통합되어 관리
편의성 향상



서버리스 아키텍처

별도 서버 관리 없이 AWS 서비스만으로 완전한 시스템
구성



유지보수 최소화

관리형 서비스만 활용하여 운영 부담 대폭 감소



빠른 피드백

모니터링과 배포 결과를 즉시 확인하여 대응 시간 단
축



효율적 협업

PR 알림과 배포 자동화로 팀 내 커뮤니케이션 효율
화



비용 투명성

일일 비용 알림으로 자원 사용과 예산 관리 개선

향후 확장

새로운 명령어



/assign

PR/이슈 자동 할당 및 알림

협업 효율 향상



/rollback

빠른 롤백으로 이전 버전 복구

안정성 향상



/scale

컨테이너/인스턴스 자동 스케일링

리소스 최적화



AI 기반 비용 예측

머신러닝을 활용한 사용량 예측 및 경고

FinOps 고도화

n8n 통합 워크플로우



n8n



Slack
알림 연동



Notion
문서 자동화



Jira
이슈 연계



배포 완료



n8n 트리거



후속 자동화

배포 후 외부 도구 자동 연계



IaC 도구 연계

Terraform

Ansible

CloudFormation

“ 채팅 기반 운영에서 완전 자동화된 End-to-End DevOps 경험으로 ”

채팅으로 관리하는 개발·배포의 흐름

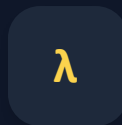
자동화로 운영을 단순하게



GitHub



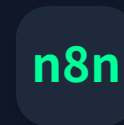
Discord



Lambda



AWS



n8n