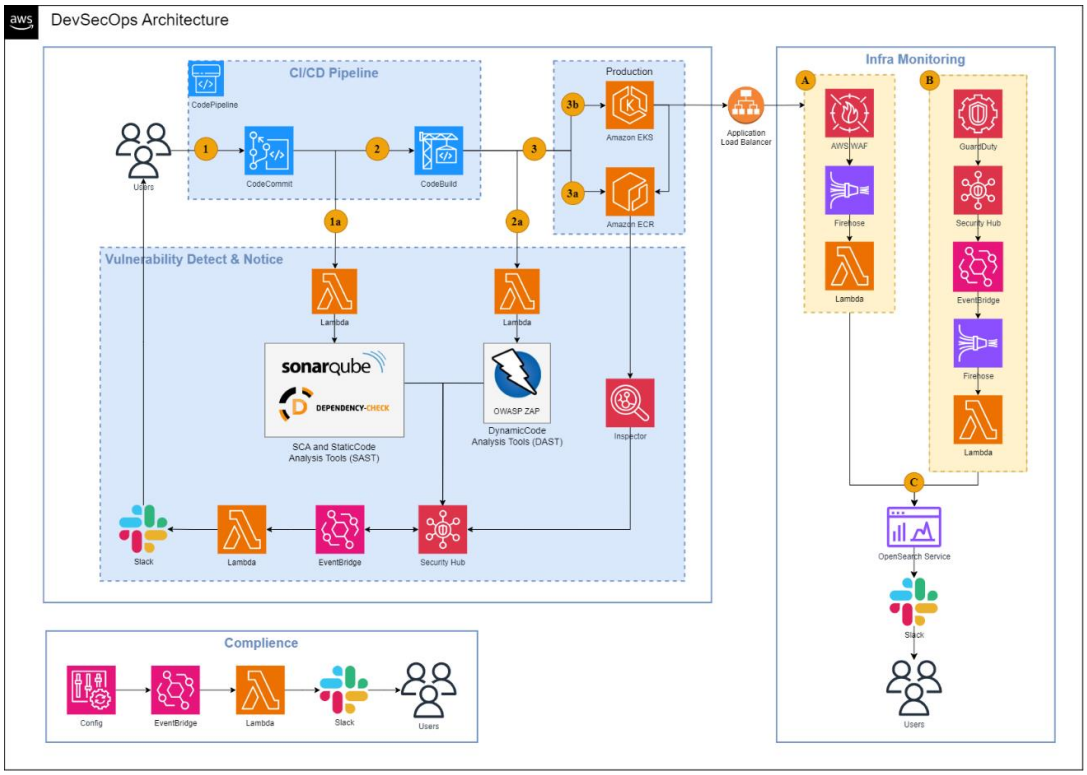
Team. INFINITE

클라우드(AWS) 환경에서 DevSecOps 구축 가이드라인

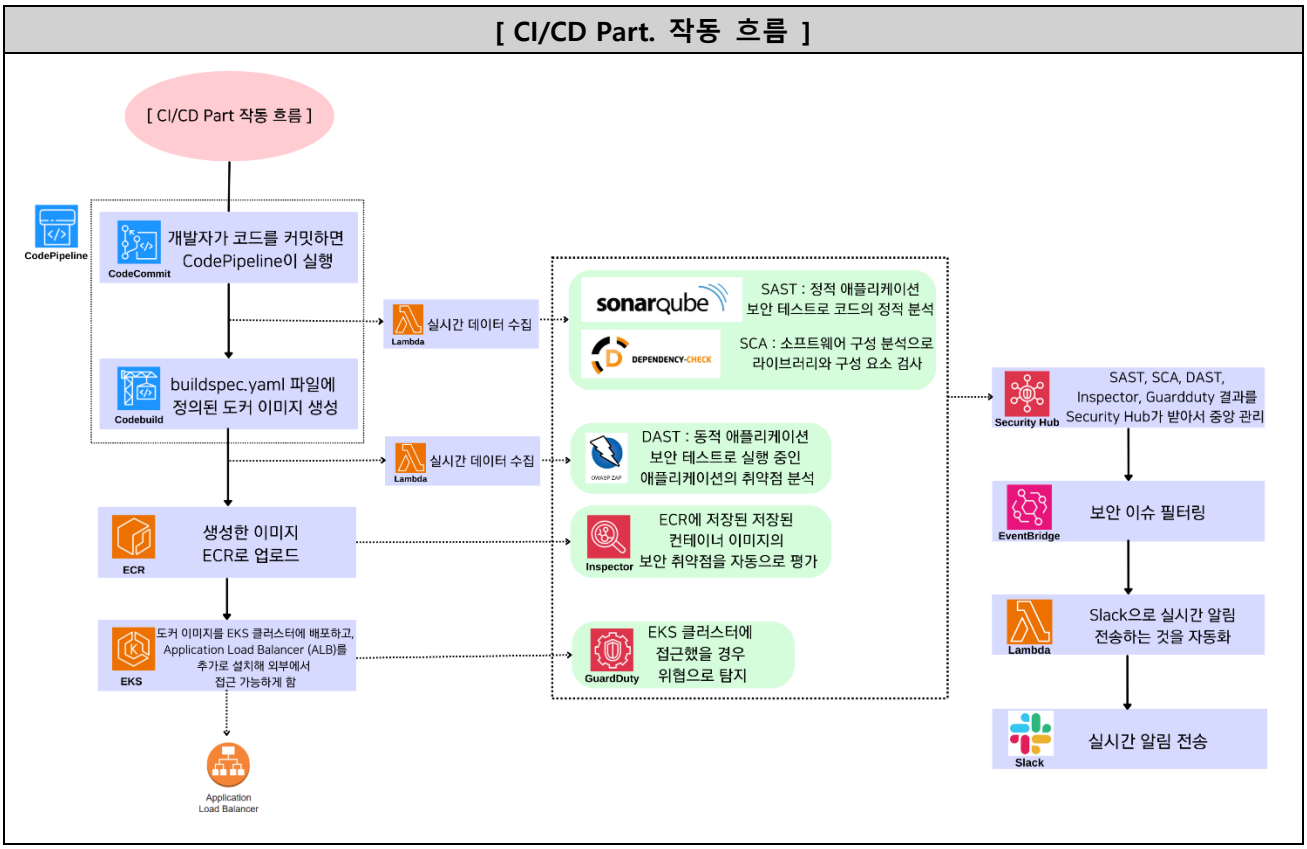
Summary

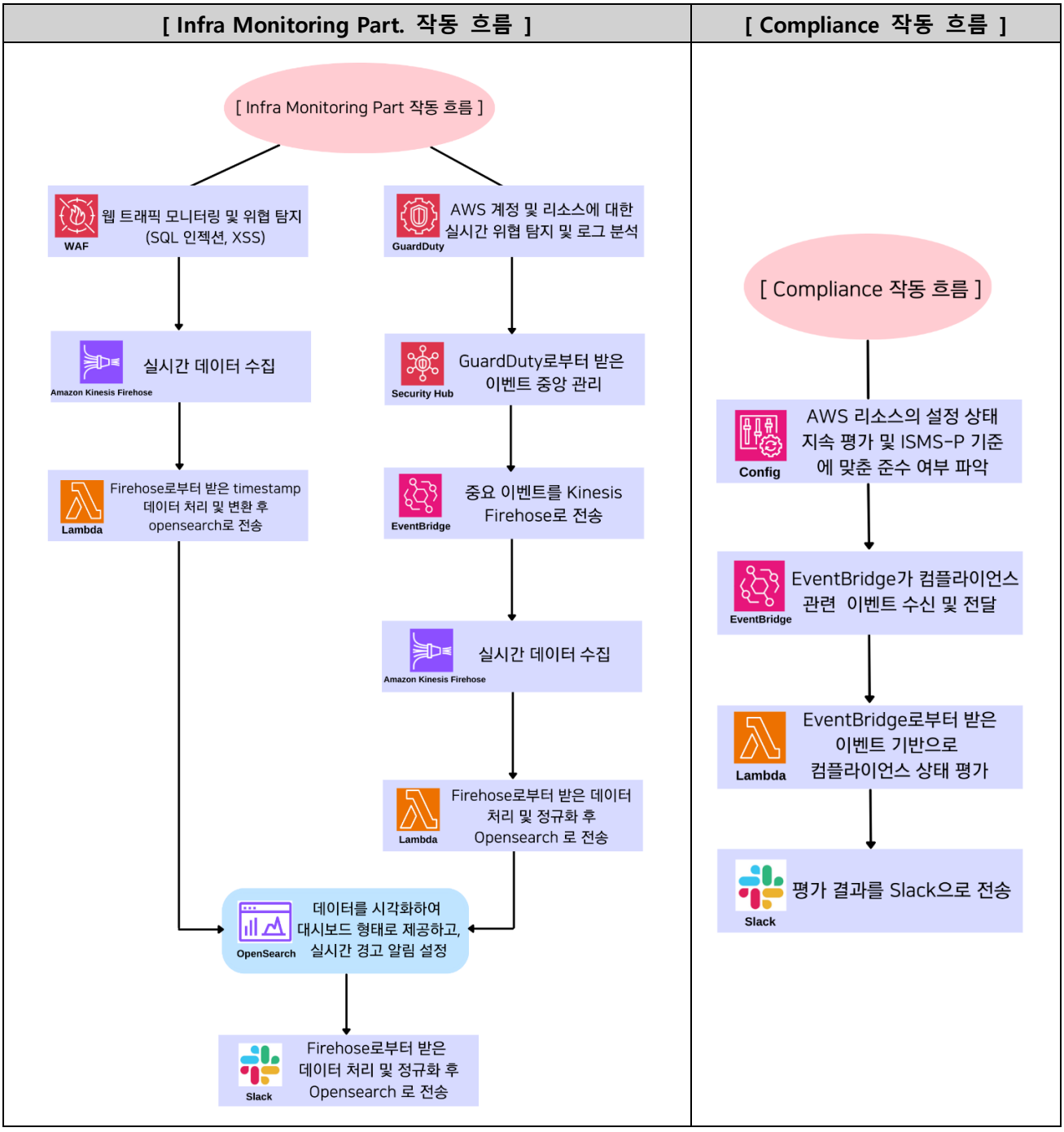
Mentor	권현준
P L	신정식
Team.	김수민(PM), 김수현, 김수민
INFINITE	남지우, 손효림, 천예슬

1. DevSecOps 아키텍처



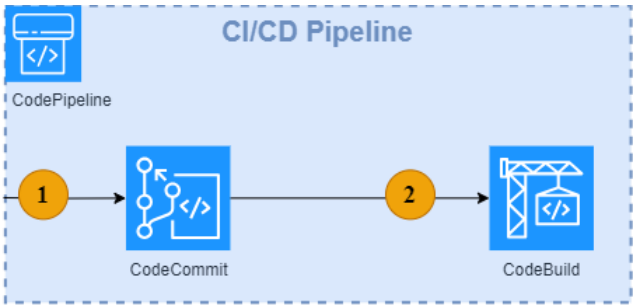
1.1 아키텍처 작동 흐름




[Infra Monitoring Part. 작동 흐름]
[Compliance 작동 흐름]
[Infra Monitoring Part 작동 흐름]
WAF
웹 트래픽 모니터링 및 위협 탐지 (SQL 인젝션, XSS)
GuardDuty
AWS 계정 및 리소스에 대한 실시간 위협 탐지 및 로그 분석
Amazon Kinesis Firehose
실시간 데이터 수집
Security Hub
GuardDuty로부터 받은 이벤트 중앙 관리
Lambda
Firehose로부터 받은 timestamp 데이터 처리 및 변환 후 opensearch로 전송
EventBridge
중요 이벤트를 Kinesis Firehose로 전송
Amazon Kinesis Firehose
실시간 데이터 수집
Lambda
Firehose로부터 받은 데이터 처리 및 정규화 후 Opensearch 로 전송
OpenSearch
데이터를 시각화하여 대시보드 형태로 제공하고, 실시간 경고 알림 설정
Slack
Firehose로부터 받은 데이터 처리 및 정규화 후 Opensearch 로 전송
[Compliance 작동 흐름]
Config
AWS 리소스의 설정 상태 지속 평가 및 ISMS-P 기준에 맞춘 준수 여부 파악
EventBridge
EventBridge가 컴플라이언스 관련 이벤트 수신 및 전달
Lambda
EventBridge로부터 받은 이벤트 기반으로 컴플라이언스 상태 평가
Slack
평가 결과를 Slack으로 전송

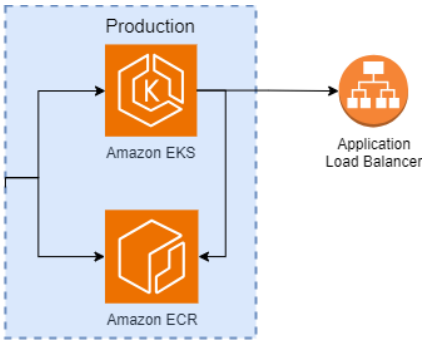
2. DevSecOps 구현 과정_CI/CD Part.

2.1 CI/CD Pipeline



개념 및 필요성	CI/CD(Continuous Integration and Continuous Deployment) 파이프라인은 코드 변경 사항을 자동으로 빌드, 테스트, 배포하는 프로세스
프로세스 흐름	AWS CodeCommit에 코드를 커밋하고, CodeBuild로 Docker 이미지를 생성하여 ECR에 업로드하는 과정을 거쳐 CodePipeline으로 자동 빌드 및 배포를 구현. 코드 커밋 시 자동으로 빌드가 실행되고 Docker 이미지가 ECR에 업로드 됨.
이점	해당 과정을 통해 코드 변경 사항을 신속하게 배포하여 사용자에게 더 빠르게 기능을 제공할 수 있게 하며, 자동화된 테스트를 통해 품질을 유지할 수 있음.

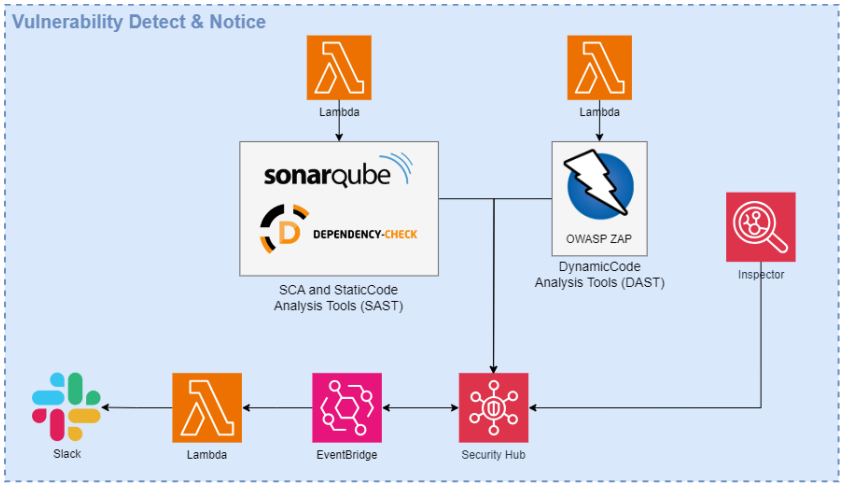
2.2 EKS Deploy



개념 및 필요성	CodePipeline에서 생성한 도커 이미지를 ECR에 업로드 하면 해당 이미지를 EKS에서 참조하는 프로세스
프로세스 흐름	도커 이미지를 ECR에 업로드한 후, eksctl 명령어로 EKS 클러스터를 구축하고, kubectl 명령어를 사용해 WebGoat 애플리케이션을 EKS에 배포하여 전체 배포 작업을 자동화. kubectl 명령어를 사용하면 로드밸런서가 CLB로 생성되는 것이 일반적이지만, 해당 로드밸런서를 WAF와 연결하기 위해서는 로드 밸런서 유형을 ALB로 변경해야 함. 따라서, buildspec.yaml 파일 내에서 AWS Load Balancer Controller를 설치하여 EKS 클러스터에서 ALB를 사용할 수 있도록 코드를 설정. 추가적으로 필요한 정책

	및 파일을 생성함으로써 ALB 배포가 가능해짐.
이점	해당 과정을 통해 컨테이너화된 애플리케이션을 자동으로 배포, 스케일링 및 운영할 수 있음. 또한 ALB 추가 설치를 통해 외부에서의 접근이 가능해짐.

2.3 Vulnerability Detect & Notice



2.3.1 EC2 생성

- EC2 인스턴스 생성: AWS EC2 인스턴스를 생성하고, PuTTY를 통해 SSH로 접근하여 SAST, DAST, SCA 도구를 설치.

2.3.2 [SCA] Dependency-Check

개념 및 필요성	코드베이스를 스캔하여 종속성을 검사하고 공개된 취약성을 식별하는 프로세스
프로세스 흐름	CodePipeline에서 CodeCommit 이후 Lambda 함수가 트리거되며 리포지토리 URL이 파라미터로 전달됨. Lambda에서 System Manager - Run Command로 EC2에 명령을 전송함. EC2에서는 분석할 리포지토리 복제, 설치된 Dependency-Check가 동작하여 분석, 분석 결과를 Security Hub로 전송하는 행위가 이루어짐. Security Hub로 전송된 탐지한 취약점에 대한 정보는 Slack을 통해 사용자에게 전달됨.
이점	해당 프로세스를 통해 조직이 사용하는 오픈 소스 및 타사 코드에 대한 가시성을 제공함. 또한 발생할 수 있는 취약성 등의 문제에 대한 정보를 수집할 수 있음.

2.3.3 [SAST] SonarQube

개념 및 필요성	SAST는 정적 애플리케이션 보안 테스트로, 소스 코드, 바이트 코드 또는 애플리케이션 바이너리를 분석하여 보안 취약점을 식별하는 방법. 정적 분석은 코드가 실행되기 전에 수행되며, 코드의 구조와 논리를 검토하여 잠재적인 보안 결함을 발견.
프로세스 흐름	- 리포지토리 URL을 Lambda에 파라미터로 전달하여 Code Pipeline Lambda SAST 스테이지를 트리거. - Lambda는 Code Pipeline에서 받은 리포지토리 URL을 Run Command로 문서로 전송 후 실행. - sonarqube_scan.py에서 분석 결과를 SonarQube에서 API로 가져온 후 분석 결과를 Json 형식으로 Security Hub로 전송.
이점	해당 과정을 통해 DevSecOps 파이프라인에서 코드 품질과 보안 취약점을 자동으로 분석하여 지속적인 보호를 제공함.

2.3.4 [DAST] OWASP ZAP

개념 및 필요성	동적 애플리케이션 보안 테스트로 실행 중인 애플리케이션을 외부에서 분석하며 모의 공격을 수행하고 취약점을 탐지하는 프로세스
프로세스 흐름	웹 애플리케이션 빌드 및 배포 이후 생성된 로드밸런서 DNS 주소를 대상으로 분석을 수행함. 해당 주소는 buildspec.yaml에서 invoke_lambda.py로 호출한 Lambda 함수를 거쳐 EC2로 전달됨. 또한 Lambda는 System Manager - Run Command로 ZAP가 실행 중인 EC2에서 스크립트를 실행함으로써 Spider로 분석 대상의 정보를 수집한 뒤 Active Scan을 진행함. 분석 결과는 Security Hub로 전송되고 Slack을 통해 사용자에게 전달됨.
이점	실제 공격자가 취약점을 악용하기 전에 취약성을 식별함으로써 위험 요소를 줄임. 또한 자동 스캐닝 툴이므로 일일이 스캔하지 않고도 보안 취약점을 발견하고 조치할 수 있도록 함.

2.3.5 AWS Inspector

개념 및 필요성	Inspector는 ECR에 저장된 저장된 컨테이너 이미지의 보안 취약점을 자동으로 평가하고 개선 권장사항을 제공. 컨테이너 이미지의 보안 상태를 지속적으로 관리하기 위해 필요.
프로세스 흐름	ECR 리포지토리를 생성하고 설정해 이미지 푸시를 하고 AWS Inspector를 실행하면

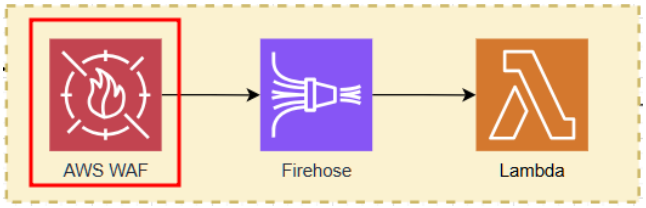
	ECR, EC2 자동 스캔해 보약 취약점 평가. 해당 결과를 Security Hub로 자동전송.
이점	자동화된 스캔을 통해 이미지를 지속적으로 평가하고, 발견된 취약점에 대한 상세 정보를 제공하여 신속한 업데이트와 패치를 가능하게 할 수 있음.

2.3.6 Security Hub to Slack

개념 및 필요성	Security Hub로 들어오는 보안 로그를 원하는 제품과 기준에 따라 필터링하고, 개발자 및 모니터링 담당자에게 알림을 제공.
프로세스 흐름	Security Hub 로그 유입 후 Event Bridge 필터링(제품 ARN, 심각도 등등)에 의해 각 상황에 맞는 lambda 트리거. Lambda json 결과 파싱 후 Slack 알림.
이점	위와 같은 흐름으로 보안 로그의 효과적인 필터링하고, 중요한 정보를 실시간으로 Slack을 통해 전달받을 수 있음.

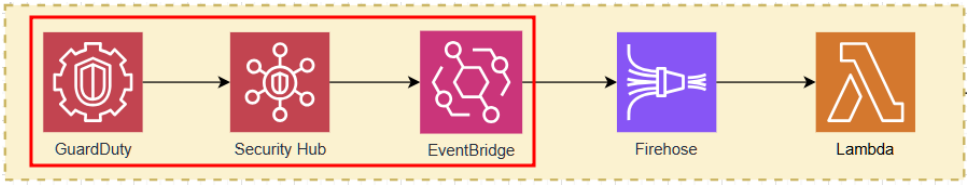
3. DevSecOps 구현과정_Infra Monitoring Part.

3.1 AWS WAF



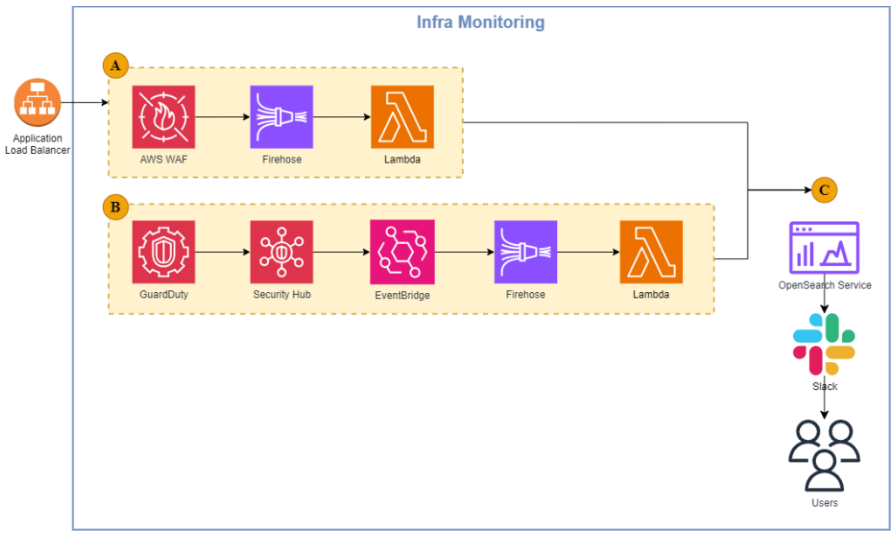
개념 및 필요성	웹 애플리케이션 방화벽(WAF)는 웹에서 발생하는 비정상 트래픽에 대한 탐지와 차단을 제공하는 방화벽 서비스. CI/CD Part에서 EKS로 배포한 웹 서버를 악의적인 외부 요청으로부터 보호하기 위해 WAF를 구현.
프로세스 흐름	웹 서비스에서 요청되는 패킷을 탐지 및 차단하는 Web ACL을 설정하고 EKS에서 추가 설치한 ALB와 연결. SQL Injection 공격과 XSS 공격 등의 탐지 규칙을 설정해 어떤 공격이 들어왔을 때 어떤 행위를 할 것인지에 대한 WAF rule을 구성.
이점	외부 요청으로부터 특정 패턴의 공격을 탐지 및 차단할 수 있음.

3.2 AWS GuardDuty



개념 및 필요성	GuardDuty는 CloudTrail 이벤트 로그 (관리, S3 데이터), DNS 로그, EKS 감사 로그 및 VPC 흐름 로그를 분석하고 처리해 보안 위협을 식별하는 모니터링 서비스. 알려진 위협 뿐만 아니라 알려지지 않은 위협에 대한 지속적인 모니터링으로 악의적인 행위를 탐지.
프로세스 흐름	GuardDuty에서 CloudTrail 위협을 탐지 후 탐지된 위협은 Security Hub로 전송. Event Bridge 설정을 통해 샘플 결과 필터링을 위한 이벤트 패턴을 작성해줄 수 있음. Security Hub를 설정해주고 Event Bridge 설정을 통해 샘플 결과 필터링을 위한 이벤트 패턴을 작성해줄 수 있음.
이점	해당 과정을 통해 CloudTrail 로그 데이터를 자동으로 분석하여 보안 위협을 실시간으로 탐지 가능.

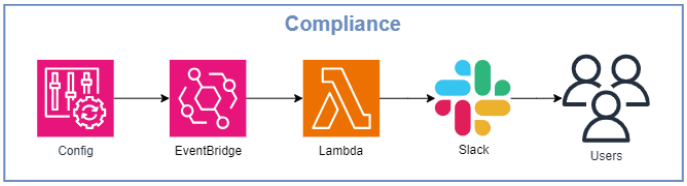
3.3 SIEM



개념 및 필요성	SIEM(Security Information and Event Management)은 보안 정보 관리와 보안 이벤트 관리의 기능을 하나의 보안 관리 시스템으로 통합하여 사이버 보안 위협을 탐지하고 대응하는 솔루션을 의미.
프로세스 흐름	- OpenSearch 도메인을 설정하고, Firehose와 Lambda를 사용하여 WAF 및 GuardDuty 로그 데이터를 변환.

	- 변환된 실시간 스트리밍 데이터를 안정적으로 받은 후 다른 AWS 서비스로 전달하는 기능을 제공하는 Firehose로 OpenSearch에 전송. - 인덱스를 생성하고 Dashboard를 통해 실시간 모니터링을 수행하며, 특정 조건에 맞는 데이터를 Slack으로 알림 전송.
이점	해당 과정을 통해 비정상적인 요청이나 위협이 감지되었을 때, 즉각적으로 알림을 전송. 또한 대시보드로 의심스러운 이벤트에 대해 중앙 집중 형태로 분석 및 사고 대응이 가능.

4. Compliance



4.1 AWS Config

개념 및 필요성	AWS Config는 AWS(Amazon Web Services)에서 제공하는 서비스로 기존 AWS 리소스를 검색, 서드 파티 리소스의 구성 기록, 특정 시점의 리소스 구성 방식 확인 등이 가능 정 준수 감사, 보안 분석, 리소스 변경 추적 및 문제 해결에 사용할 수 있음.
프로세스 흐름	AWS Config를 설정하여 리소스 기록 및 규칙을 추가하고, S3 버킷과 연결하여 데이터 거버넌스를 설정. ISMS-P 규정 준수 팩을 배포하여 운영 모범 사례를 적용. Lambda 함수와 Event Bridge를 설정하여 규정 미준수 알림을 Slack으로 전송.
이점	규정 미준수 시 Slack으로 실시간 알림을 받아 신속하게 대응할 수 있음.