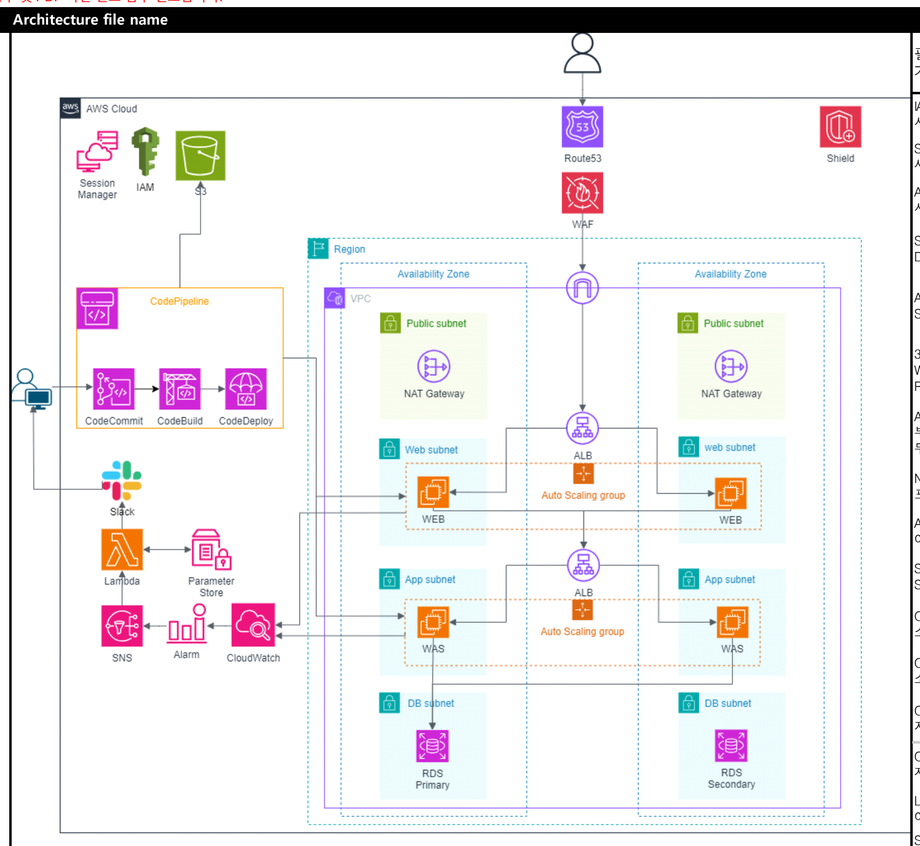
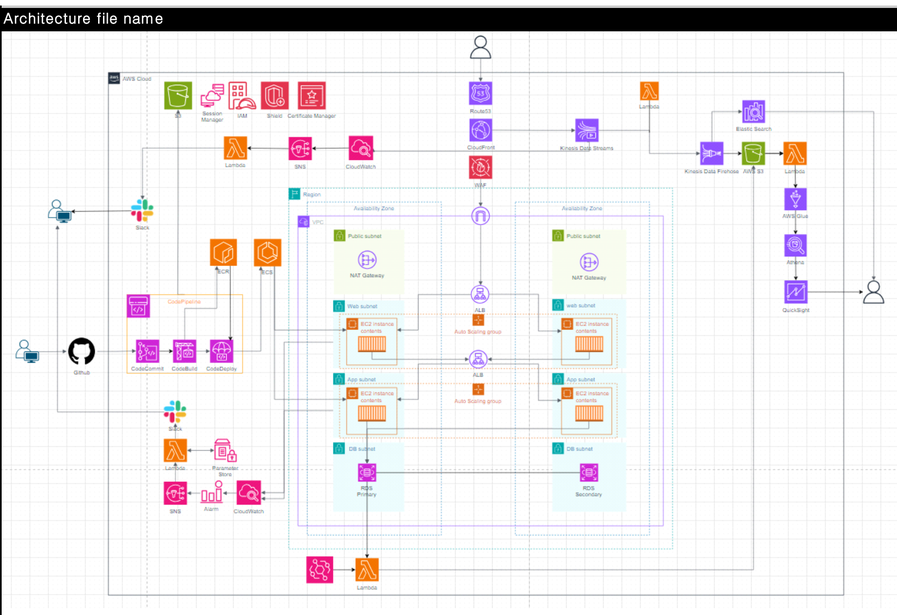
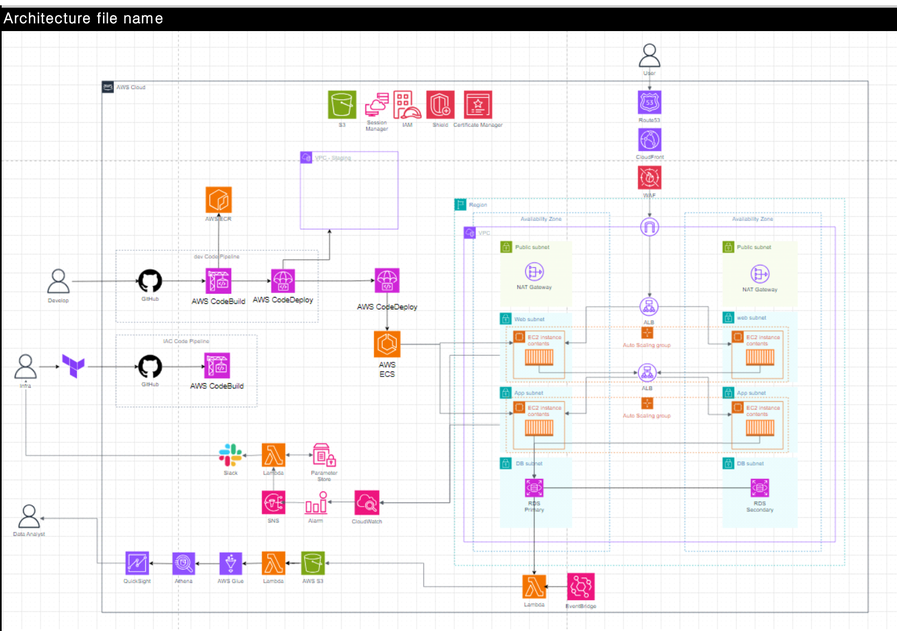
AWS Infra Architecture

아키텍처 이미지 파일 첨부 및 PDF 버전 별도 첨부 필요합니다.

Ver	Architecture file name	Descriptions
1.0		필수 구성 및 보안 기본적인 3tier 구성, 모니터링, 백업, CI/CD, 보안을 구성하였습니다. IAM 사용자와 시스템의 권한 관리와 완전한 접근 제어를 위해 사용 Session Manager 서버 인스턴스에 대한 안전한 액세스를 제공하여 관리 작업을 보다 안전하게 수행 Amazon Route 53 사용자가 도메인 네임으로 ip 주소를 찾도록 함 Shield DDoS 공격으로부터 서비스를 보호 AWS WAF SQL 인젝션 방지, 크로스 사이트 스크립팅 등 규칙을 지정해 악의적인 웹 공격을 방어 3-Tier Web, App, DB 분리로 3-tier의 구조 구성 RDS active-standby를 통해 백업 ALB, Auto Scaling 부하 분산 및 탄력성을 가져감 두 개의 AZ를 통해 고가용성을 보장 NAT Gateway 프라이빗 서브넷의 인스턴스를 인터넷에 연결 Amazon CloudWatch 이벤트 발생시 SNS로 알람을 보내도록 함 SNS SNS의 데이터를 가지고 와 Slack을 통해 사용자가 메시지를 받도록 함 Code Commit 소스 코드 관리를 위한 개인 리포지토리 서비스 사용 Code Build 소스 코드 컴파일, 테스트 실행 및 준비된 소프트웨어 패키지를 생성 Code Deploy 자동화된 소프트웨어 배포 서비스 구성 CodePipeline 지속적인 통합과 지속적인 배포(CI/CD) 워크플로우를 자동화 Lambda 이벤트에 응답하여 코드를 실행 S3 로그 및 데이터 저장

Ver	Architecture file name	Descriptions
1.1		시나리오 1안 : 쇼핑몰 데이터 분석을 위한 인사이트 제공 데이터 분석 파이프라인에 초점을 맞춘 인프라입니다. 쇼핑몰 인프라에 데이터 분석 파이프라인을 붙여 데이터를 분석한다고 가정했을때의 인프라를 구성하였습니다. Certificate Manager SSL/TLS 인증서를 관리하여 웹 트래픽의 보안을 강화 CloudFront 사용자에게 캐시된 콘텐츠를 빠르게 제공하여 응답 속도를 향상시키고 불필요한 트래픽을 차단 실시간 데이터 스트리밍 쇼핑몰 재고 관리를 위한 데이터 분석 파이프라인입니다. 클릭 로그를 실시간으로 데이터를 받아와 일시적으로 저장, 처리합니다. 사용자 선호도, 행동 분석등을 통해 마케팅 전략을 세우거나 예측 알고리즘과 연결하여 인사이트를 제공합니다. Kinesis Data Streams 실시간으로 Data를 받아들여 일시적으로 저장 및 처리 데이터를 실시간으로 수집하고 처리할 필요성 때문에 선택하였습니다. Kinesis Data Firehose Data Streams에서 전송된 데이터를 안전하게 S3로 전송 s3와의 통합을 통해 데이터의 안정적 장기 저장 및 분석 준비, 데이터 레이크를 구성하기 위해 선택하였습니다. Elastic Service 스트리밍 데이터를 저장하고 이를 분석 실시간 데이터에 대한 심층적 분석, 모니터링, 대시보드 생성을 제공하고, 데이터를 신속하게 추출하고 빠르게 대응할 수 있는 서비스를 고려하여 선택하였습니다. 배치 전략 쇼핑몰 종합 데이터를 위한 데이터 분석 파이프라인입니다. 사용자 성별, 나이, 지역과 같은 데이터를 이용합니다. 배치 처리를 통해 s3에 이동시켜 정제 후 데이터 분석을 진행합니다. 심층적인 분석을 통해 리소스 관리, 마케팅 전략, 상품 및 서비스 전략과 같은 인사이트를 제공합니다. Event Bridge 시스템 부하가 낮은 시간대에 Lambda 함수를 실행하도록 스케줄링 Lambda 수집된 데이터를 처리하고 가공, RDS 스냅샷을 생성하고 S3에 저장 S3 로그 데이터와 RDS 데이터를 저장 AWS Glue S3에 저장된 데이터 카탈로그를 생성하여 데이터 관리를 용이하게 함 다양한 데이터 소스 스키마, 사용 가능한 데이터 세트를 중앙에서 관리해줘 데이터 품질 관리를 손쉽게 가져갈 수 있고 데이터 정제 프로세스를 자동화하여 일관성과 정확성을 확보할 수 있는 서비스를 선택하였습니다. Athena AWS Glue에서 생성된 데이터 카탈로그를 기반으로 S3에 저장된 데이터 쿼리 실행 서버리스의 이점, 다양한 데이터 형식 지원, glue와의 통합을 통해 정확하고 신속한 데이터 분석, 데이터 사일로 분석의 이점을 살리기 위해 사용하였습니다. QuickSight Athena에서 분석된 데이터를 시각화 ECS 컨테이너화된 애플리케이션을 배포, 관리 및 스케일링 ECR 컨테이너 이미지를 저장, 관리, 배포를 위한 완전관리형 레지스트리 서비스 사용

Ver	Architecture file name	Descriptions
1.2		시나리오 2안 : 티켓팅 사이트 CI/CD 환경 구축 CI/CD 환경에 초점을 맞춘 인프라입니다. 개발, IAC CI/CD 환경을 구성하였고 배치 전략은 S3까지 데이터를 이동하는 프로세스를 가져간 다음, 추가적으로 데이터 분석을 위한 환경을 구축하였습니다. ECS 컨테이너 환경 실시간 데이터 스트림 처리가 문제 없이 진행될 수 있도록 오케스트레이션 툴이 필요했고 ECS로 구성하였습니다. 서비스 규모를 단정지을 수 없어 우선 마이크로서비스가 아니고, AWS 내에서만 서비스를 한다는 가정으로 EKS가 아닌 ECS를 선택하였습니다. ECR 컨테이너를 쉽게 저장, 배포를 위해 레지스트리를 구성하였습니다. 프라이빗 리포지토리를 사용하여 민감한 데이터 및 코드를 보관합니다. 개발 파이프라인 WEB, WAS 서버 CI/CD입니다. GitHub CodeCommit에 github를 연결하여 구성하였습니다. CI를 위한 형상관리 툴 필요, github action을 통한 Webhook 트리거로 자동화하였습니다. CodeBuild CodeBuild를 통해 빌드, 테스트 단계를 구성하였습니다. 현재는 관리의 편의성, 완전관리형 서비스를 고려해 codebuild를 선택하였습니다. 규모와 필요 플러그인에 따라 jenkins로 교체하는 선택지를 가져갈 예정입니다. CodeDeploy 스테이징, 프로덕션 환경으로의 CD를 위해 사용하였습니다. ECS와의 호환성을 생각하여 선택하였습니다. 구성도에는 안 나왔지만 블루/그린 배포 전략을 취할 예정입니다. 스테이징 환경을 구성하여 실제 안정적인 배포를 위한 검증 단계를 가졌습니다. 크기 때문에 스테이징 환경 구성 표시는 VPC만 표시하였고 프로덕션 vpc구성과 동일합니다. 스테이징 환경에서의 DB 데이터는 현재는 더미 데이터를 생각하고 있습니다. IAC 파이프라인 전체 인프라를 Terraform으로 구성하여 프로비저닝합니다. 모듈화, 상태 유지 및 추적의 이점을 생각하여 Terraform을 선택하였습니다. GitHub 마찬가지로 형상관리, 트리거로 자동화하였습니다. CodeBuild Terraform plan, apply를 buildspec 파일을 통해 정의합니다. 리소스에 변경사항을 적용합니다. 배치 처리 티켓팅 사이트에 쌓이는 데이터를 배치 처리를 통해 보관하고 분석을 추가적으로 가져갔습니다. 과정은 1안과 동일합니다.