

EC2 단일 인스턴스 아키텍처의 결합 해제 및 2-Tier 재설계 보고서

1. 문제 상황 요약

현재 시스템은 다음과 같은 구조로 구성되어 있습니다:

- 하나의 EC2 인스턴스에서 Web 서버 + App 서버를 동시에 실행
- 데이터베이스는 프라이빗 서브넷의 Amazon RDS에서 제공

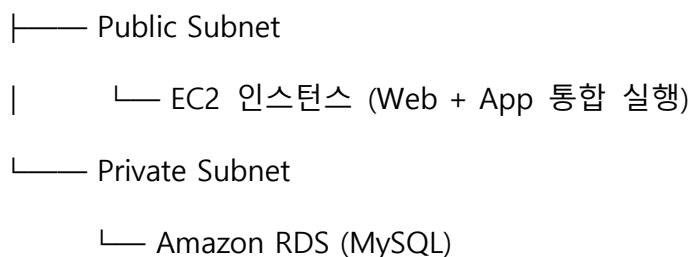
이처럼 모든 웹/앱 기능이 하나의 EC2에 결합되어 있는 구조는 초기에는 단순하지만, 운영/확장/장애 대응 측면에서 취약합니다.

2. 목표

- EC2 단일 인스턴스에서 수행 중인 웹 서버와 애플리케이션 로직을 분리
 - 2-Tier 아키텍처로 재구성하여 결합도는 낮추고, 유연성 및 유지보수성을 향상
 - 마이크로 서비스 기반 운영을 위한 전단 준비 단계 설계
-

3. 기존 아키텍처 구조

[VPC]



기존 구조의 한계점

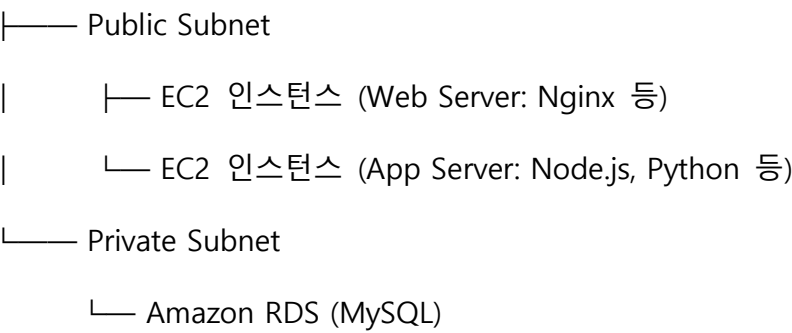
항목	내용
결합도	Web과 App이 동일 인스턴스에 탑재되어 있어 독립 배포 어려움
장애 대응	EC2 장애 시 Web+App 동시에 중단
확장성	로드밸런싱, 오토스케일링 도입이 제한적

항목	내용
운영 관리	로그, 보안 정책, 트래픽 제어 등 분리 관리 어려움
아키텍처 원칙 위반	Well-Architected Framework의 모듈화, 분리 원칙 위반

4. 개선된 2-Tier 아키텍처 설계

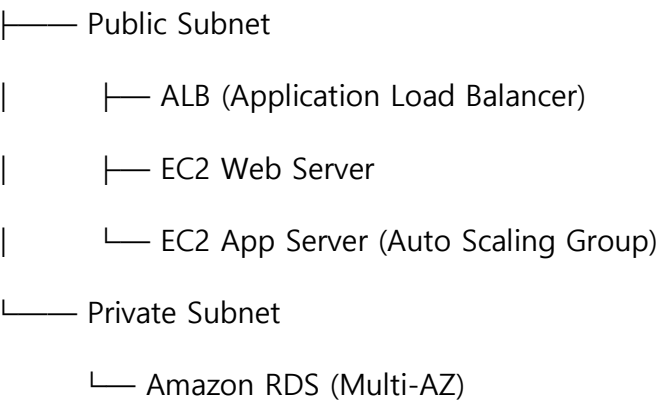
설계 구조도

[VPC]



또는 운영 자동화 적용 시 다음과 같이 구성할 수 있습니다:

[VPC]



5. 기능별 분리 구성 설명

계층	서비스 구성	설명
Web Layer	EC2 (Nginx/Apache)	사용자 요청 수신, 정적 파일 제공

계층	서비스 구성	설명
App Layer	EC2 (Flask, Node.js, Spring 등)	비즈니스 로직 처리, API 응답
DB Layer	Amazon RDS (MySQL)	트랜잭션, 데이터 저장 계층

보안 구성

- Web Layer는 퍼블릭 서브넷에 위치해 외부 접속 허용 (포트 80/443)
- App Layer는 Web Layer로부터만 접근 가능
- RDS는 App Layer에서만 접근 가능 (보안 그룹, NACL로 제한)

6. 결합 해제의 장점

측면	개선 효과
독립 배포/운영	웹/앱 서버를 각각 독립 배포 가능 (무중단 배포 구조 가능)
확장성	EC2 Auto Scaling 적용 시 수평 확장 유리
고가용성	웹 또는 앱 계층 중 하나에 문제가 생겨도 전체 시스템 중단 방지
보안성 향상	각 계층별로 최소 권한 보안 정책 설계 가능
DevOps 호환성	마이크로서비스 또는 컨테이너 기반 배포로 확장 용이
유지보수 용이성	로그 분리, 트래픽 분석, 장애 조치 구분 가능

7. 향후 확장 방안 제안

결합 해제를 기반으로 다음 단계 확장을 고려할 수 있습니다:

확장 항목	설명
Amazon ECS/Fargate	웹/앱 서버를 컨테이너 기반으로 전환
Amazon CloudFront	정적 콘텐츠 글로벌 배포
Amazon Secrets Manager	DB 접근정보 안전하게 관리

확장 항목	설명
CI/CD 도입	CodePipeline, CodeDeploy 통한 자동 배포
Amazon ElastiCache	App Layer 성능 향상을 위한 캐시 레이어 추가

8. 결론

EC2 인스턴스 하나에 모든 기능을 담은 단일 구조는 초기에는 빠른 배포가 가능하지만, 확장성, 유지보수, 장애 대응, 보안 측면에서 명확한 한계를 가집니다.

본 보고서에서 제안한 2-Tier 아키텍처는 기능별 분리를 통해:

- 결합도를 낮추고
- 독립적인 운영이 가능하며
- 향후 마이크로 서비스 구조로의 확장이 용이한 기반을 마련할 수 있습니다.

이는 AWS Well-Architected Framework의 운영 우수성, 보안성, 성능 효율성 측면에서 모두 유리한 설계입니다.