

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

---

### 1. 문제 개요

현재 시스템은 하나의 EC2 인스턴스에 Web 서비스와 Application 로직이 통합된 상태로 배포되어 있으며,  
데이터베이스는 프라이빗 서브넷에 위치한 Amazon RDS에서 운영되고 있습니다.

이 구조는 간단하지만 다음과 같은 단점이 존재합니다:

- 웹과 앱 계층이 단일 인스턴스에 공존함으로써 운영상 결합도가 높고,
- 기능별 장애 복구/확장성/보안 설계가 어렵습니다.

이에 관리자는 Well-Architected Framework의 원칙을 반영하여 결합 해제 및 2-Tier 아키텍처로의 분리를 추진하고자 합니다.

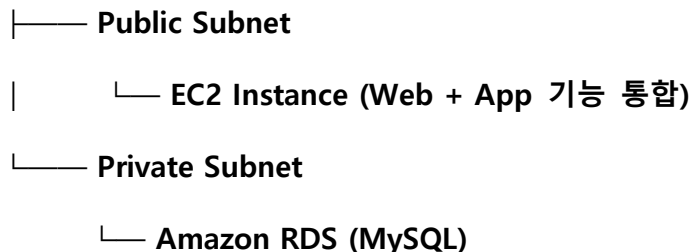
---

### 2. 목표

- EC2 단일 인스턴스에 묶여 있는 Web과 App 기능을 분리하여 구조화
  - 기능별 중복성 제거 및 독립 운영이 가능한 아키텍처 설계
  - 향후 마이크로 서비스 기반 구조로 전환이 가능한 기반 마련
- 

### 3. 기존 아키텍처 구조 (문제 상태)

[VPC]



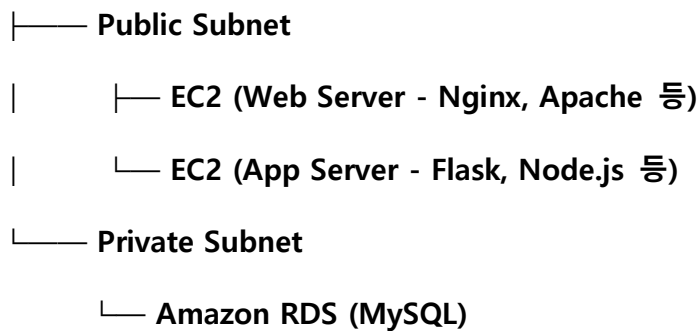
구성 요소	역할
-------	----

EC2 단일 인스턴스	Web 서비스 + 애플리케이션 로직 + API 처리
-------------	------------------------------

구성 요소	역할
RDS	데이터베이스 전용, 프라이빗 접근 제한

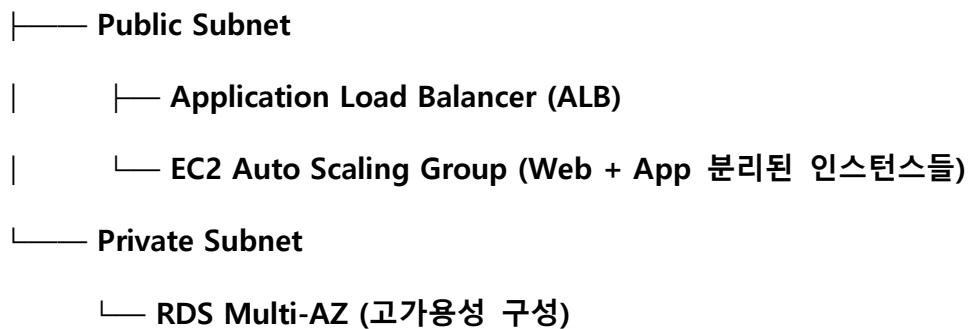
#### 4. 개선된 아키텍처 설계 (2-Tier, 기능 분리)

[VPC]



또는 자동화 구성 시:

[VPC]



#### 5. 기능별 분리 구성 설명

계층	역할	상세 설명
----	----	-------

Web Layer EC2 Web 서버 정적 파일 제공, HTTPS 처리, 요청 수신

App Layer EC2 App 서버 비즈니스 로직, REST API, DB 연동

DB Layer Amazon RDS 트랜잭션 처리, 백업, 복제 관리

- 보안 분리: Web은 인터넷 접근 허용, App은 Web만 접근 가능, RDS는 App만 접근 가능

- 성능 분리: 각 계층별로 성능 튜닝과 모니터링이 가능

---

## 6. 결합 해제 아키텍처의 장점 설명

항목	설명
모듈화	Web/App/DB 기능 독립 → 각 계층 단위로 배포 및 운영 가능
확장성	App 서버만 Auto Scaling 가능 → 비용 효율적인 구조
장애 격리	App 서버 장애 발생 시 Web은 유지 → 전체 장애 방지
DevOps 친화	CI/CD 구축 시 계층별 독립 배포 가능
보안성 향상	최소 권한 기반 접근 제어, IAM/보안 그룹 분리 적용
향후 확장	컨테이너화, 마이크로서비스, Lambda 전환 등 유연성 확보

---

## 7. 향후 마이크로 서비스 확장 방안

이 구조는 단순한 계층 분리뿐만 아니라, 향후 마이크로서비스 구조로 확장하기 위한 기초 설계가 됩니다.

개선 요소	적용 가능 서비스
컨테이너 오케스트레이션	Amazon ECS, EKS, Fargate
서버리스 전환	AWS Lambda, API Gateway
메시지 큐 분리	Amazon SQS, SNS
보안 고도화	WAF, Shield, Secrets Manager
운영 자동화	CloudWatch Logs, CloudTrail, CodeDeploy 등

---

## 8. 결론

이번 보고서는 단일 EC2 인스턴스에 집중된 Web + App 서비스를 기능별로 분리하고, 2-Tier 구조로 아키텍처를 재설계함으로써 모듈화된, 확장 가능한, 운영 친화적 시스템

환경을 도입하고자 합니다.

이러한 결합 해제 설계는 다음과 같은 AWS Well-Architected 원칙을 충족합니다:

- 운영 우수성: 계층별 운영 가능
- 보안성: 최소 권한 접근
- 신뢰성: 장애 격리
- 성능 효율성: 계층별 자원 최적화
- 비용 최적화: 불필요한 리소스 축소