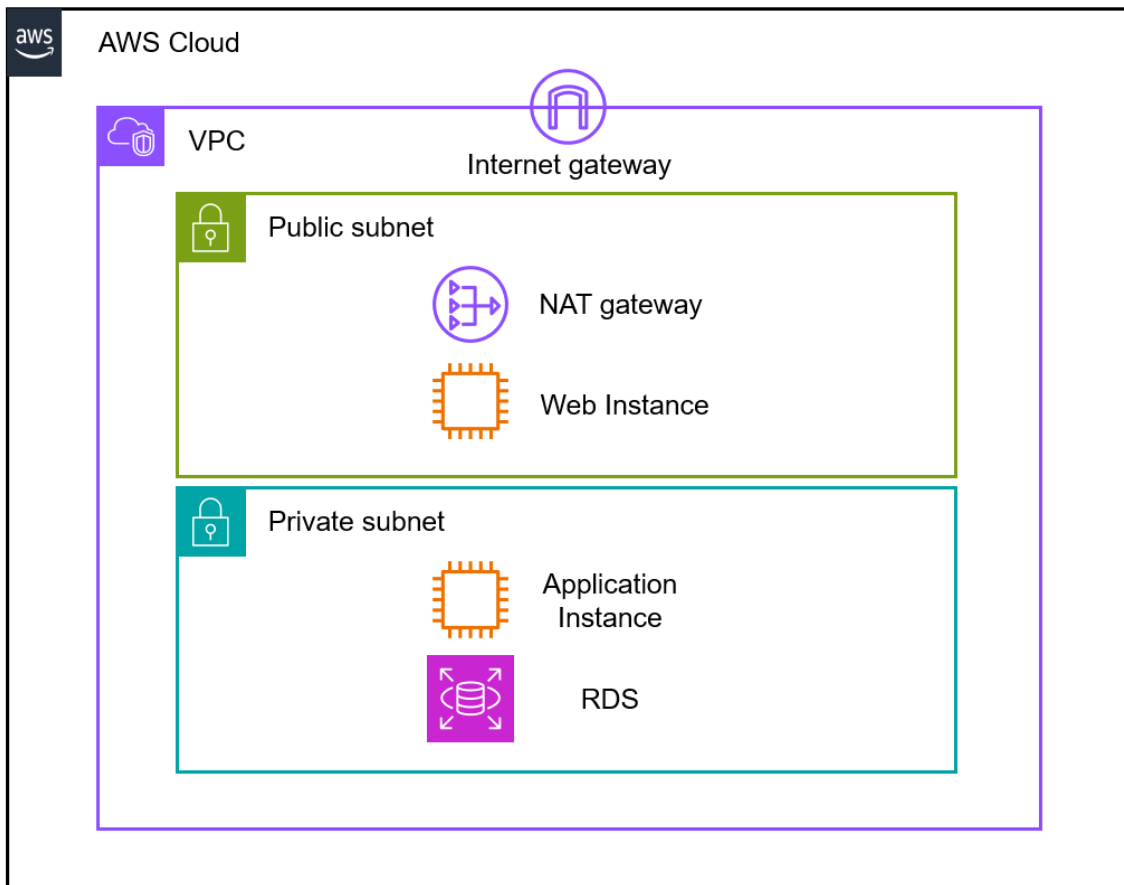


# 단일 EC2 환경의 서비스 분리와 아키텍처 재설계

웹·애플리케이션 결합 구조를 분석하고 마이크로서비스 아키텍처로 전환하기

하나의 EC2 인스턴스에 웹 및 애플리케이션을 서비스 중입니다. EC2 하나에 종속된 웹과 애플리케이션을 분리하기 위한 보고서입니다. 본 보고서는 마이크로 서비스 서비스 아키텍처의 운영 방식을 채택합니다.



## 1. 서비스 독립성 강화

- 각 서비스는 자신의 역할에만 집중하며 다른 계층과의 의존도를 낮춘 구조입니다.
- 변경이나 장애 발생 시 영향 범위가 제한적이므로 특정 서비스만 수정 및 배포할 수 있습니다.
- 배포와 운영을 서비스 단위로 분리할 수 있어 CI/CD 적용에 유리합니다.

## 2. 유지보수 및 운영 효율 향상

- Web, App, DB 계층이 분리되어 있으므로 문제 식별과 해결이 빠르게 이루어집니다.
- 로그, 모니터링, 백업 등을 서비스별로 독립적으로 설정 및 관리할 수 있습니다.
- 팀 간 업무 분담이 명확해지고 전문화된 운영이 가능해집니다.

### 3. 확장성과 유연성 확보

- 특정 서비스에만 부하가 집중될 경우 해당 서비스만 확장하면 되므로 자원 효율성이 높아집니다.
- 예를 들어, Web 서비스만 추가하거나 DB 서비스 성능만 개선하는 방식으로 유연하게 대응할 수 있습니다.
- 향후 컨테이너(ECS/EKS)나 서버리스(Lambda)로의 전환도 용이한 구조입니다.

### 4. 보안 경계 구분이 명확함

- Private Subnet에 위치한 APP과 DB는 외부에서 직접 접근이 불가능합니다.
- Public Subnet을 통한 유입 트래픽은 NAT와 보안 그룹 설정을 통해 통제되므로 취약점 노출이 줄어듭니다.

### 5. 장애 격리 및 복구 용이

- 하나의 서비스에 장애가 발생해도 나머지 서비스는 정상적으로 운영될 수 있습니다.
- 복구 작업은 전체 시스템이 아닌 해당 컴포넌트에만 국한되므로 MTTR를 단축할 수 있습니다.