

AWS Well-Architected 기반 관리형 서비스 재설계 보고서

1. 문제 상황 개요

기존 아키텍처는 다음과 같은 방식으로 구성되어 있습니다:

- **NAT 인스턴스(NAT Instance):** 퍼블릭 서브넷에 위치, 프라이빗 서브넷 인스턴스의 인터넷 접근을 중계
- **DB 서버(MySQL):** 프라이빗 서브넷에 위치, EC2 인스턴스를 통해 직접 관리

이 구성은 모든 기능을 EC2 인스턴스에 의존하는 비관리형(Non-managed) 환경으로, 관리자는 보안 패치, 인스턴스 상태 모니터링, 장애 대응 등 직접 관리해야 하는 부담이 큼니다.

2. 목표

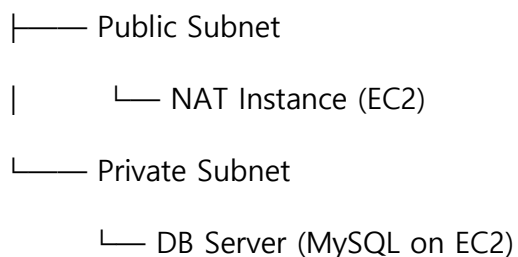
Well-Architected Framework의 원칙(운영 우수성, 보안성, 신뢰성, 성능 효율성, 비용 최적화)을 기반으로,

비관리형 EC2 인스턴스를 관리형 서비스로 전환하여 관리 부담을 줄이는 방향의 아키텍처 재설계를 수행합니다.

3. 기존 아키텍처 vs 개선 아키텍처

기존 아키텍처 (비관리형 기반)

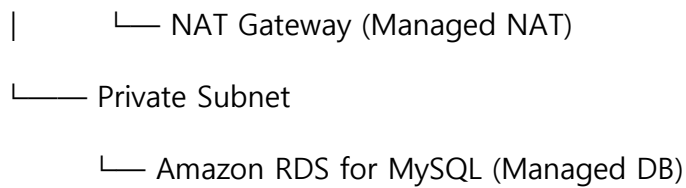
[VPC]



개선 아키텍처 (관리형 서비스 기반)

[VPC]





4. 개선된 아키텍처 설명

구성 요소	비관리형 구성 (기존)	관리형 구성 (개선)
NAT	EC2 기반 NAT 인스턴스	NAT Gateway (완전 관리형)
DB	EC2 위에 MySQL 설치	Amazon RDS for MySQL
관리	필요 수동 업데이트, 모니터링 직접 구성	AWS가 운영, 자동 백업,고가용성
보안성	보안 설정 직접 구성	IAM, 자동 패치, 암호화 설정 포함
장애 대응	인스턴스 상태 수동 감시	RDS Multi-AZ, NAT GW 이중화 지원

5. 관리형 서비스 vs 비관리형 서비스 비교

항목	비관리형 서비스 (EC2)	관리형 서비스 (NAT GW / RDS)
설정 복잡도	직접 설치 및 설정 필요	클릭 몇 번으로 구성 가능
유지보수	OS/패치/네트워크 수동 유지	AWS에서 자동 관리
고가용성	직접 이중화 구성 필요	기본 제공 (Multi-AZ, 탄력성)
비용	낮을 수 있으나 관리 시간/비용 발생	비용 다소 증가, 운영 편의성 높음
모니터링	CloudWatch 설정 필요	내장형 지표 및 경보 제공
확장성	수동으로 스케일링	자동 스케일링 (RDS, GW)
백업	스크립트 또는 에이전트 설정	자동 백업, Snapshot 기능 제공

6. 관리형 서비스 제안 이유

- **운영 효율성 향상:** 패치, 백업, 모니터링 자동화
 - **보안 강화:** IAM, KMS, 보안 그룹 연동 강화
 - **비용 절감:** 장기적 인건비 및 관리비용 절감
 - **확장성:** 수요 변화에 따른 유연한 리소스 조절
 - **신뢰성 향상:** 장애 복구, 다중 가용성 영역(AZ) 지원
-

7. 결론

기존의 비관리형 EC2 기반 아키텍처는 모든 유지보수 책임이 사용자에게 있어 운영 부담이 크며, 장애 대응 및 보안 측면에서도 취약점이 존재합니다.

따라서 AWS의 관리형 서비스를 도입하여 **Well-Architected Framework의 5가지 원칙** (운영, 보안, 신뢰성, 성능, 비용)에 부합하는 구조로 전환하는 것이 이상적입니다.

- **NAT Gateway**는 NAT Instance를 완벽히 대체하며 더 높은 가용성과 관리 편의성을 제공합니다.
- **Amazon RDS for MySQL**은 DB 관리의 대부분을 AWS에 위임함으로써 신뢰성과 안정성을 향상시킵니다.

결과적으로, 관리자는 관리 부담을 줄이고 비즈니스 로직에 집중할 수 있으며, 전반적인 시스템 안정성과 보안이 강화됩니다.