

Play Date 클라우드 2기

AWS 기반의고가용성 아키텍처 구축 (커뮤니티 게시판)

미니 프로젝트 2nd



PROCESS

목차

프로젝트 개요

요구사항

DB

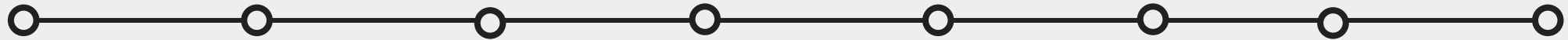
기대효과

팀원 소개

아키텍처

스토리보드

보안, 가용성 및 안정성



프로젝트 개요



팀원 소개

강윤호



DB ERD 설계

윤창운



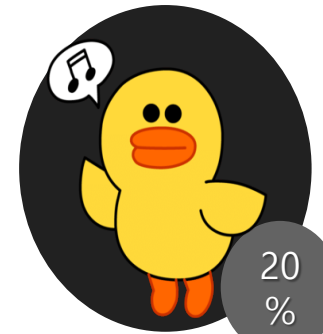
게시판 기능 설계 및 구현
디자인

이정빈



게시판 기능 설계 및 구현
디자인

조원균



시스템 아키텍처
설계 및 구축

한지영



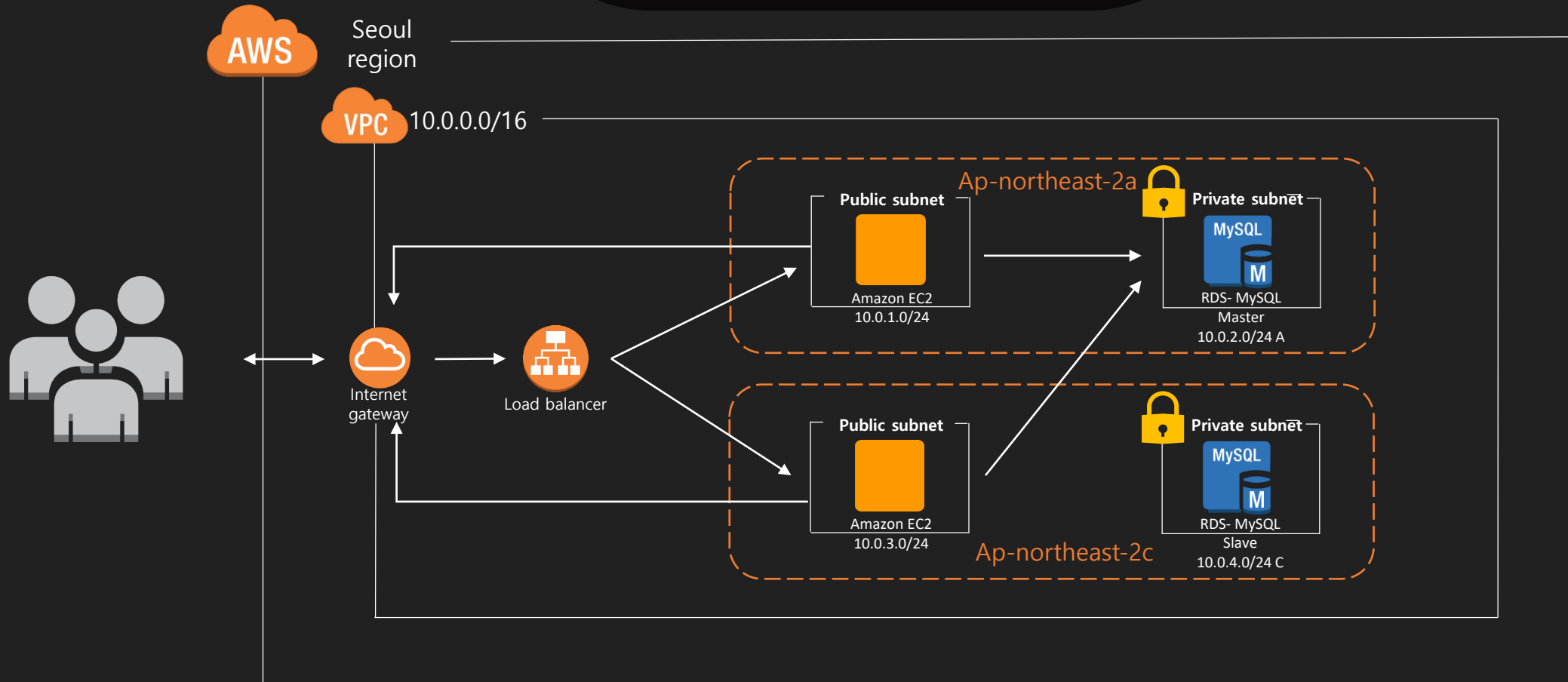
시스템 아키텍처
설계 및 구축

100%

커뮤니티 게시판 요구사항

1. 기본적인 게시판 형태를 구성 한다.
2. 기능은 게시판 목록 조회, 수정, 작성, 삭제 기능 요구.
3. 게시판 기능 수준의 관계형 데이터 베이스가 필요
4. 안정적인 서비스 제공을 위해 장애 예방 차원에서 이중구조로 사용
5. 보안 기능 필요
6. 천만원 미만의 예산으로 게시판 구축

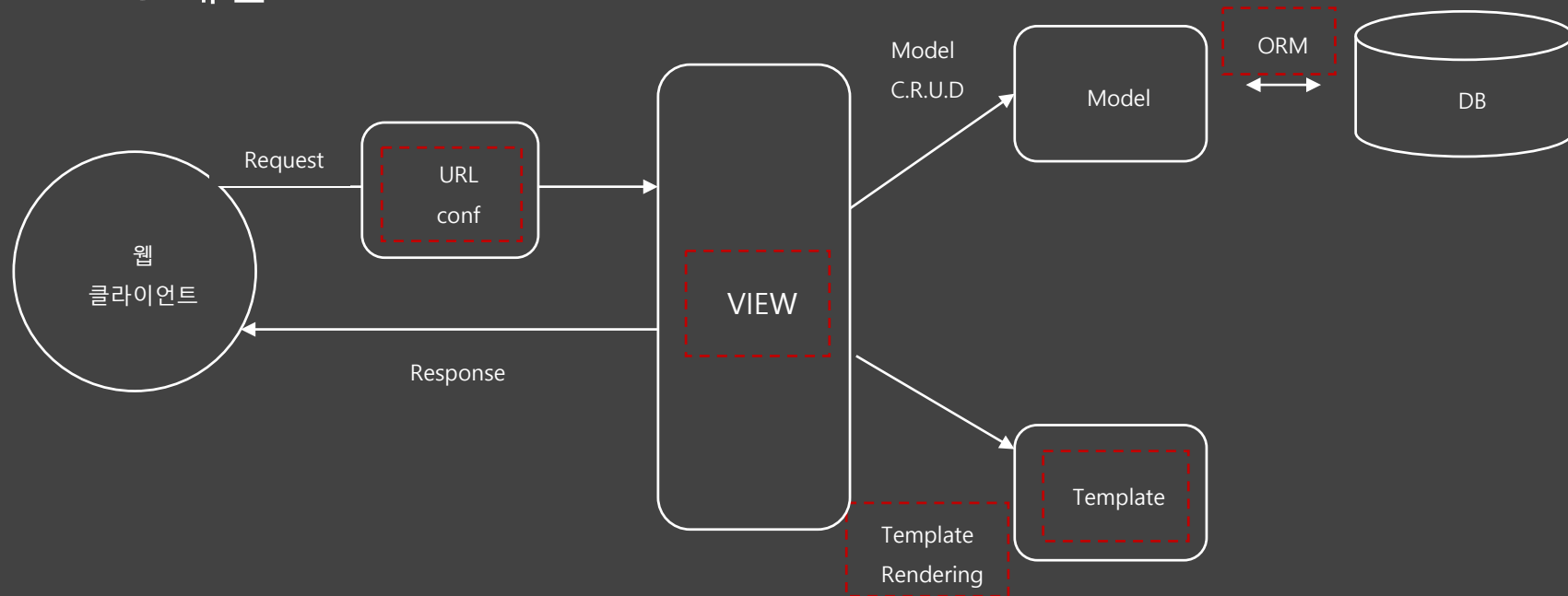
시스템 아키텍처



- 하나의 가용 영역이 장애가 발생 하더라도 서비스에는 이상이 없음
- 부하분산 기능을 활용하여 웹 서버의 부하를 분산 시킴

소프트웨어 아키텍처

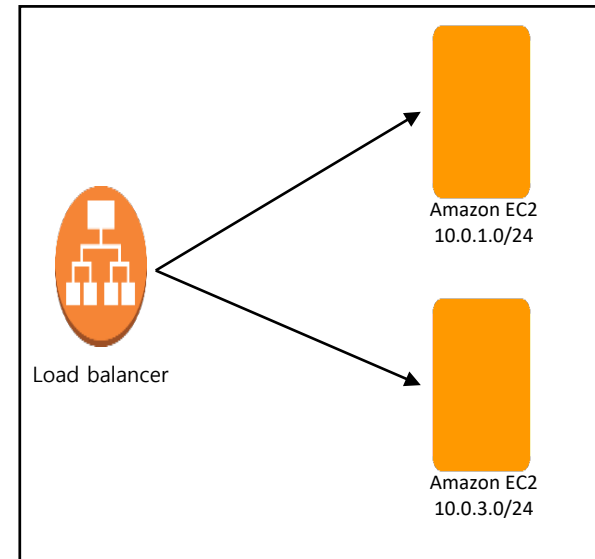
MVC 패턴



ELB

Elastic Load Balancing

- 수신되는 애플리케이션 트래픽을 여러 Amazon EC2 인스턴스 분산하여 인스턴스의 부하방지
- HTTP, HTTPS, TCP, UDP 및 SSL 프로토콜을 사용합니다.
- 비정상 인스턴스를 인식하고 이에 대응합니다.
- 4계층 또는 7계층의 로드 밸런서가 존재하며 이번 프로젝트에선 7계층 로드 밸런서를 사용합니다.



DB ERD

사용자		users	
아이디	id	Key	VARCHAR(225)
비밀번호	pw	Password	VARCHAR(225)
이름	name	Name	VARCHAR(225)
닉네임	nickname	Name	VARCHAR(225)

게시글		board	
게시글 번호	b_number	Key	INT
제목	title	Name	VARCHAR(225)
게시글 내용	b_context	Context	VARCHAR(10000)
게시글 등록일시	b_datetime	Datetime	DATETIME
작성자	id	Key	VARCHAR(225)

- 게시판 글 작성시 사용자 테이블 (users)의 id 컬럼을 외래키로 참조하여 작성됨
- 게시판 글 작성시 아이디가 아닌 작성자 이름 으로 변경 되어 출력됨

DB 테이블

Users Table : 유저 정보에 대한 테이블

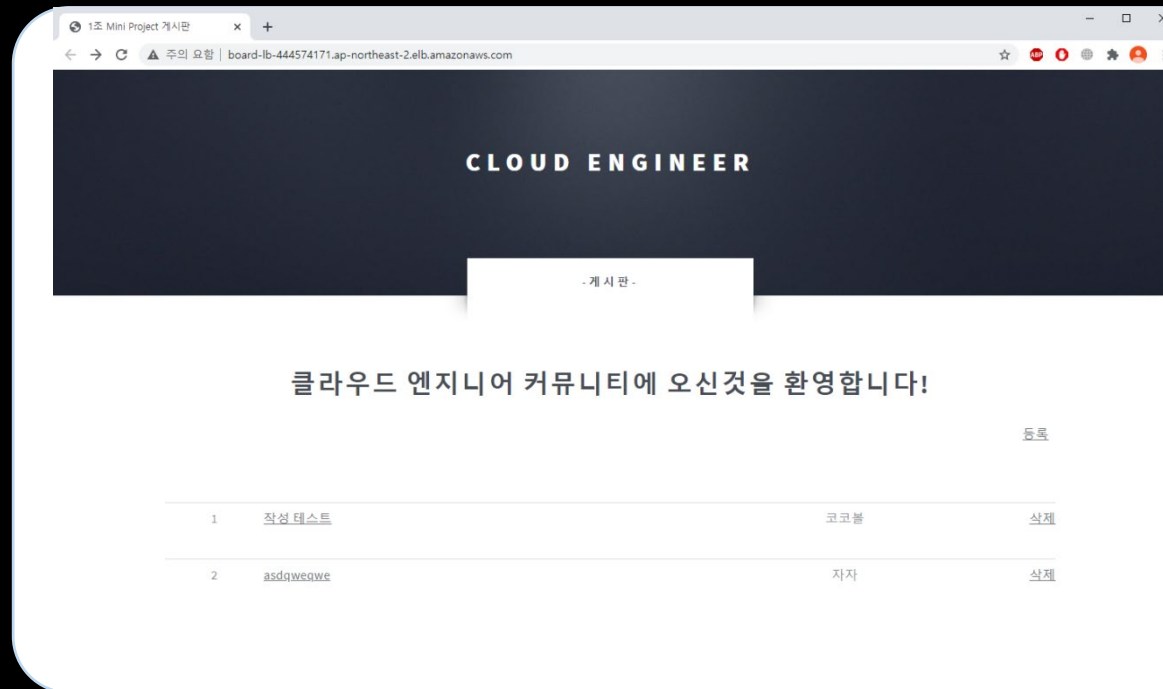
테이블명	users								
COMMENT	사용자 목록								
행 번호	키	논리	물리	도메인	타입	Null 허용	기본값	코멘트	
1	PK	아이디	id	Key	VARCHAR(225)	N		사용자 아이디	
2		비밀번호	pw	Password	VARCHAR(225)	N		사용자 비밀번호	
3		이름	name	Name	VARCHAR(225)	N		사용자 이름	
4		닉네임	nickname	Name	VARCHAR(225)	N		사용자 닉네임	

DB 테이블

board Table : 게시판 데이터 테이블

테이블명	board							
COMMENT	사용자 목록							
행 번호	키	논리	물리	도메인	타입	Null 허용	기본값	코멘트
1	PK	게시글 번호	b_number	Key	INT	N		게시글의 고유번호
2	FK	작성자	id	Key	VARCHAR(225)	N		users의 id를 참조
3		제목	Title	Name	VARCHAR(225)	Y		게시글의 제목
4		게시글 내용	b_context	Context	VARCHAR(10000)	Y		
5		게시글 등록일시	b_datetime	Datetime	DATETIME	Y		

스토리보드



- 접속 URL : <http://board-lb-444574171.ap-northeast-2.elb.amazonaws.com/>

스토리보드

1조 Mini Project 게시판 x +

← → ↻ 주의 사항 board-lb-444574171.ap-northeast-2.elb.amazonaws.com/board_write ☆

CLOUD ENGINEER

- 게시판 -

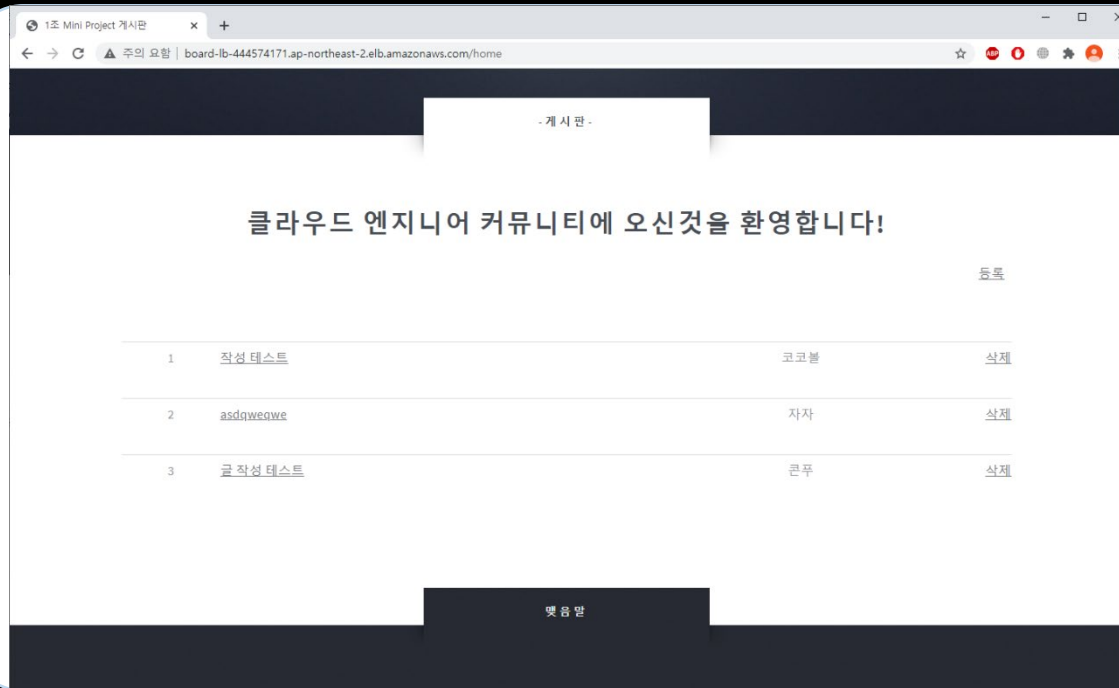
게시판 등록

제목	글 작성 테스트
작성자	admin
내용	게시판 기능 작동 확인

저장

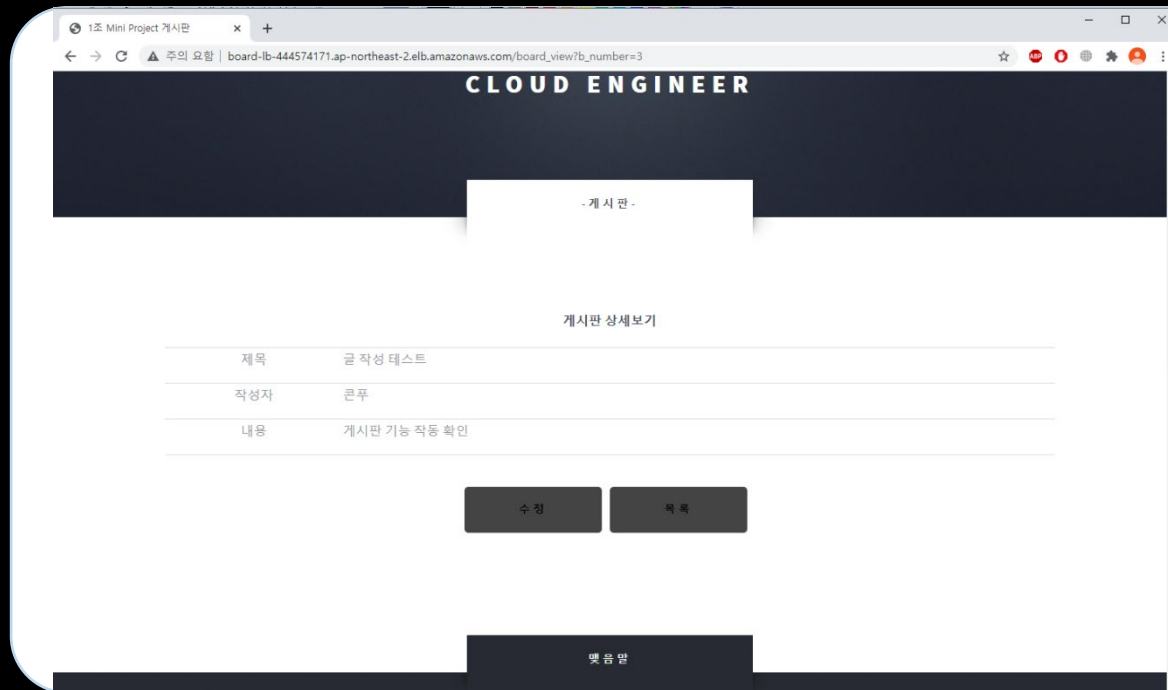
- 제목 입력
- 작성자 : 사용자 정보의 id 와 외래키로 연결 되어 있습니다.
- 내용 작성 후 저장

스토리보드



- 등록 : 게시판 작성 기능
- 삭제 : 게시글 삭제 기능

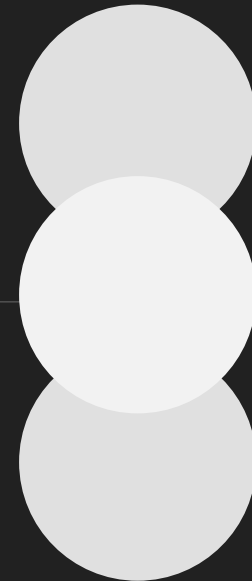
스토리보드



- 게시글 클릭 후 내용 확인

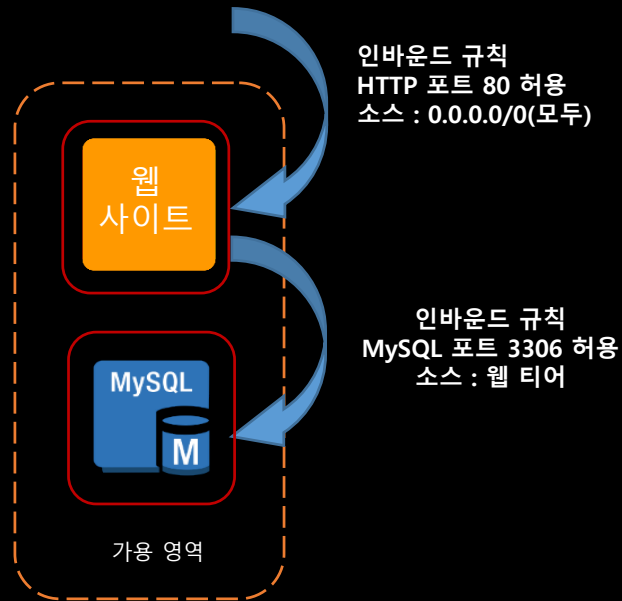
기대 효과

- 요구사항을 충족 시킬 목적으로 구성된 아키텍처 구조로 관리가 용이하다.
 - 문제 발생시 어떤 문제가 발생했는지 파악하기 쉽다.
 - Amazon RDS를 사용하면 작업을 쉽게 처리할 수 있다.
- 백업, 복원, 소프트웨어 자동 패치, 오류감지 기능으로 안정성 및 가용성을 확보한다.
 - AWS 실제로 사용한 자원에 대해서만 비용을 지불 하기 때문에 비용이 저렴하다.
- 인프라를 프로비저닝 하고 DB SW 설치 및 유지 관리가 편리하다.
 - 게시판을 이용해 원하는 정보를 빠르게 얻을 수 있다.
 - 누구나 자기의 글을 쉽게 공유 할 수 있다.



보안

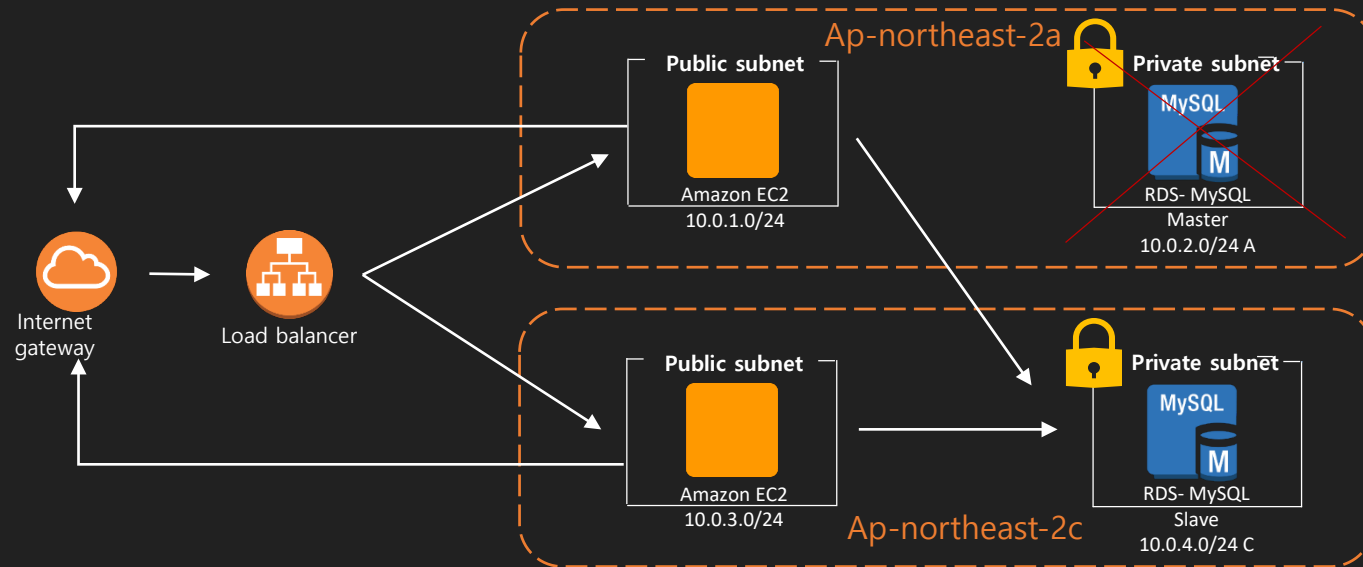
보안그룹 체인링크



보안

- VPC는 클라우드 내의 가용영역에 따로 분리되어 운용되어 외부 접근 1차 차단.
- 클라이언트가 접속 요청을 통해 인터넷 게이트웨이로 접근, 라우터를 통해 접속요청.
- 서브넷의 보안 그룹으로 EC2 인스턴스로 접속 요청 허가.
- EC2 내부의 보안그룹으로 외부트래픽 검증후 DB 서버 접근 권한 획득. 최종 수락.
- 다음과 같은 과정을 걸쳐 보안 수준 격상.

가용성 및 안전성



가용성

- 복수의 가용 영역 + **로드 밸런서**를 이용, 부하분산을 통해 서버에 가해지는 부담 경감 + 단일 장애로 인해 전체 시스템 중단되는 일을 방지하여 안정성과 가용성 높임.
- 자동으로 데이터베이스 소프트웨어를 패치

안정성

- 데이터베이스를 백업
- 사용자가 정의한 보존 기간 동안 백업을 저장하여 특정 시점으로 복구를 가능하게 함
- 시중에 제공되고있는 오픈소스 데이터 베이스 엔진의 모든 기능을 제공, 아마존 RDS 에서 제공



Q & A



감사합니다.