

AURORA READ REPLICA PRE-WARMING

자동 확장되는 READ REPLICA의 성능 개선을 위한 방안

강준우

Email : jwk9490@gmail.com

Github : <https://github.com/ballenabox/AuroraPreWarming.git>

CONTENTS

프로젝트 개요

03

주제 선정 과정

구성

04

구성 / Architecture

기능

07

각 구성 요소의 역할

실행 과정

08

작동 과정

결과

09

실행 결과

프로젝트 개요

작업 정보

기간 : 25.05.26-25.06.06

기여도 : 100%



amazon AURORA

문제 상황

AURORA의 AUTO SCALING으로 생성된 READ REPLICA는 기존 인스턴스와 달리 캐시 데이터가 적재되어 있지 않아 성능 차이 발생

해결 방안

새로 생성된 READ REPLICA 인스턴스를 애플리케이션에 연결하기 전에 별도의 WARMING 과정을 진행

비교 대상

WARMING을 거치지 않은(COLD) 상태의 인스턴스와 응답 시간 비교

구성

제공 기능

- 생성된 READ REPLICA의 상태 확인
- 대상 인스턴스에 TOP100 쿼리 기반 WARMING 작업 실행
- WARMING이 정상 종료된 후 CUSTOM ENDPOINT에 편입

사용 서비스/기술

- AWS
 - AURORA POSTGRESQL
 - EVENTBRIDGE
 - STEP FUNCTION
 - LAMBDA
 - S3
- PYTHON

구성



Aurora PostgreSQL



Lambda(Python)

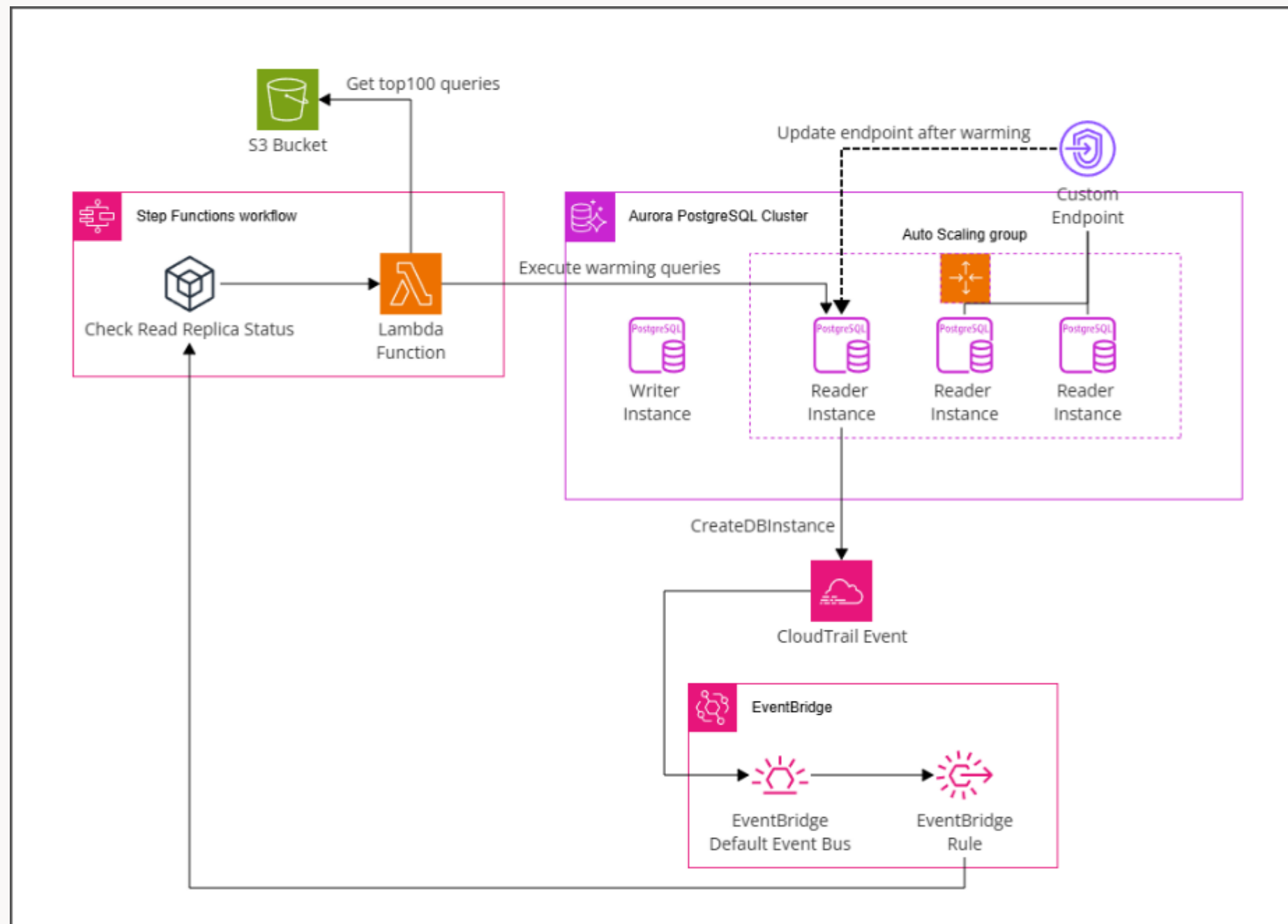


Step Function



EventBridge

ARCHITECTURE



1. AUTO SCALING으로 READ REPLICA 생성

→ CLOUDTRAIL에 이벤트 발생

→ EVENTBRIDGE가 STEP FUNCTION 트리거

2. STEP FUNCTION WORKFLOW 진행

→ 생성된 READ REPLICA의 STATUS 확인

→ S3에 있는 TOP100 쿼리 파일에서 WARMING에 사용할 쿼리 추출

→ READ REPLICA 대상으로 WARMING QUERY 실행

3. WARMING 종료 후 CUSTOM ENDPOINT 업데이트

기능

각 서비스 요소의 역할



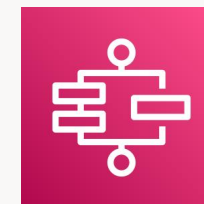
Aurora
PostgreSQL

본 프로젝트의 대상이 되는 데이터베이스



EventBridge

CloudTrail에서 이벤트를 감지하여 Step Function 작동



Step Function

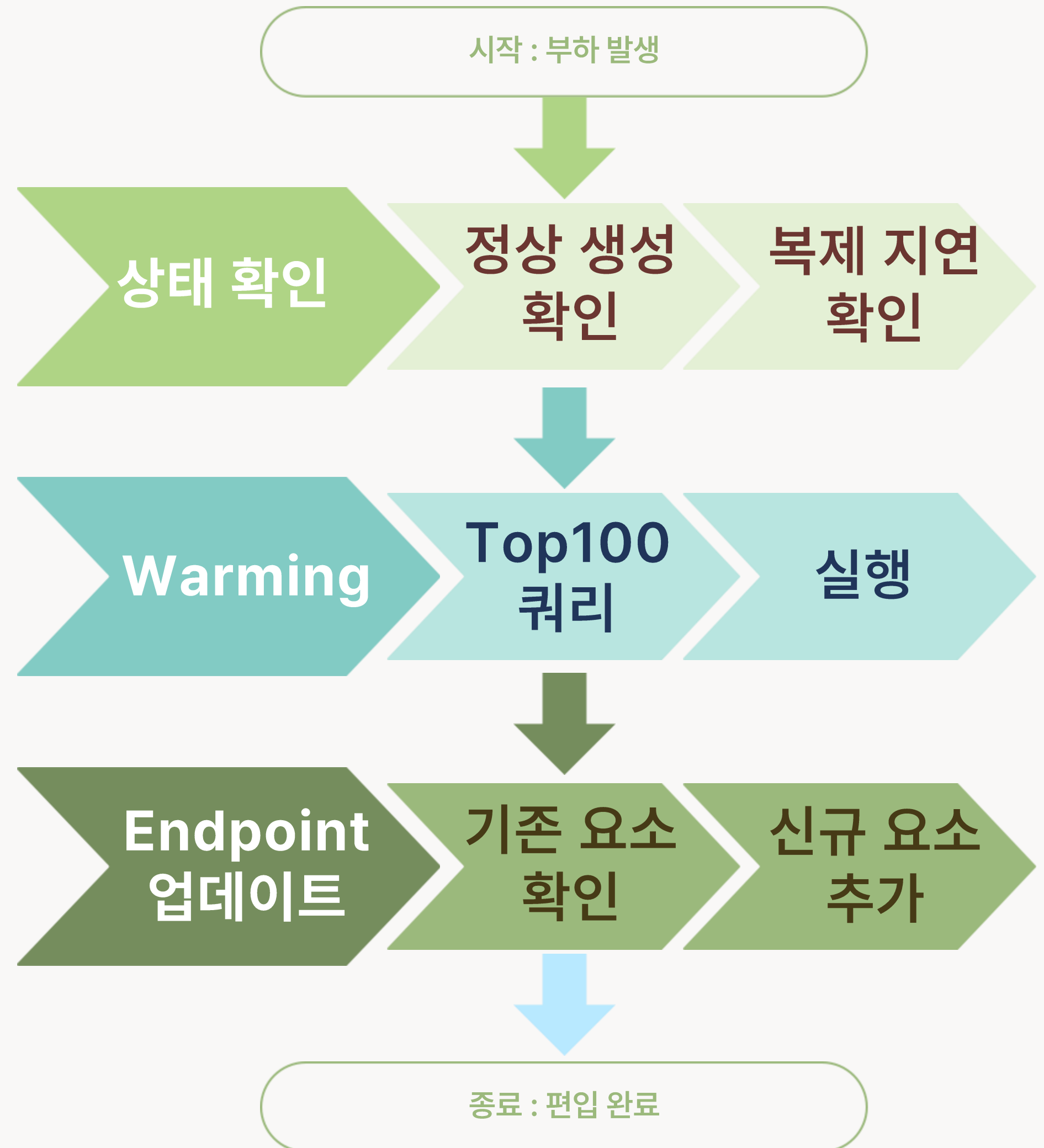
- Auto Scaling으로 생성된 Read Replica의 상태 확인
- Read Replica 대상으로 Warming 과정 진행
- Warming이 완료된 인스턴스를 Custom Endpoint에 추가



Lambda

- S3에서 Warming Query를 받아와 Read Replica 대상으로 실행
- Custom Endpoint의 구성 인스턴스 목록 업데이트

실행 과정



결과

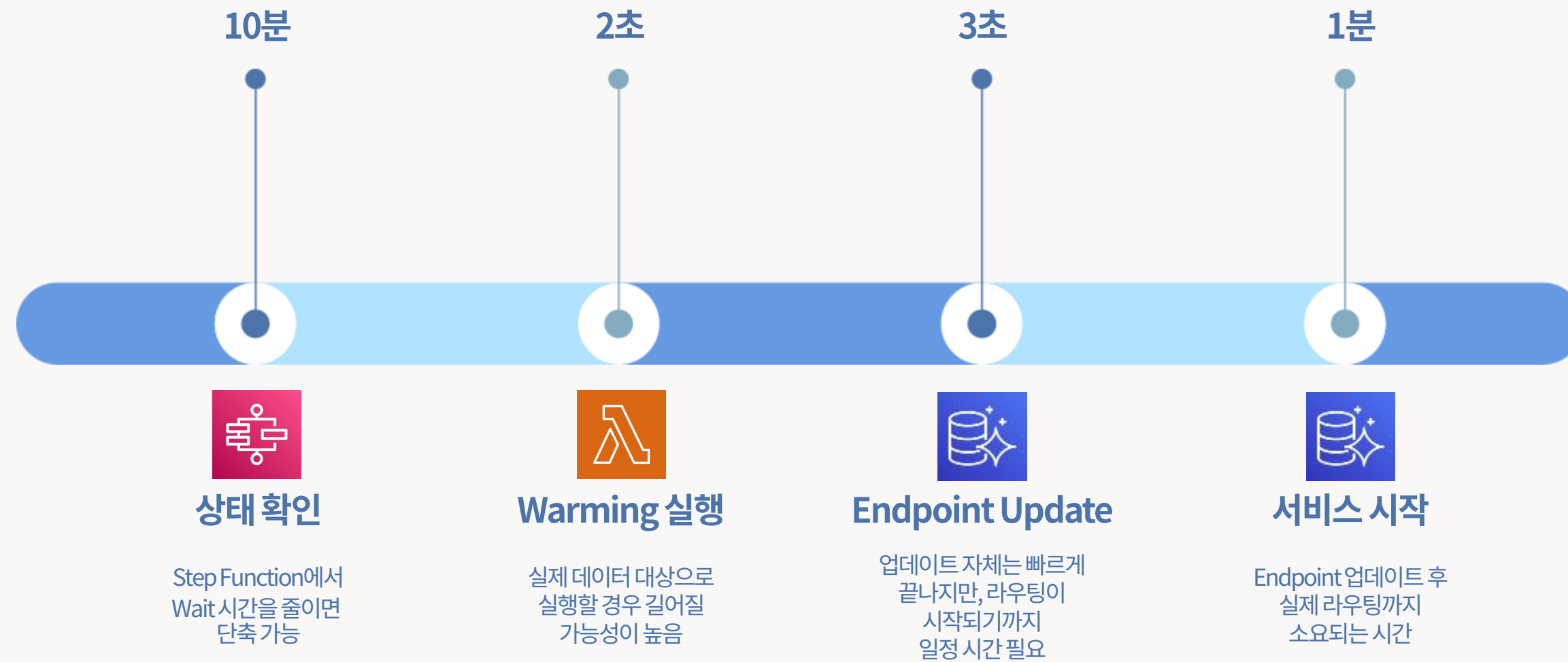
작업 소요 시간

READ REPLICA 생성 시작부터
ENDPOINT 업데이트까지 걸린 시간은?

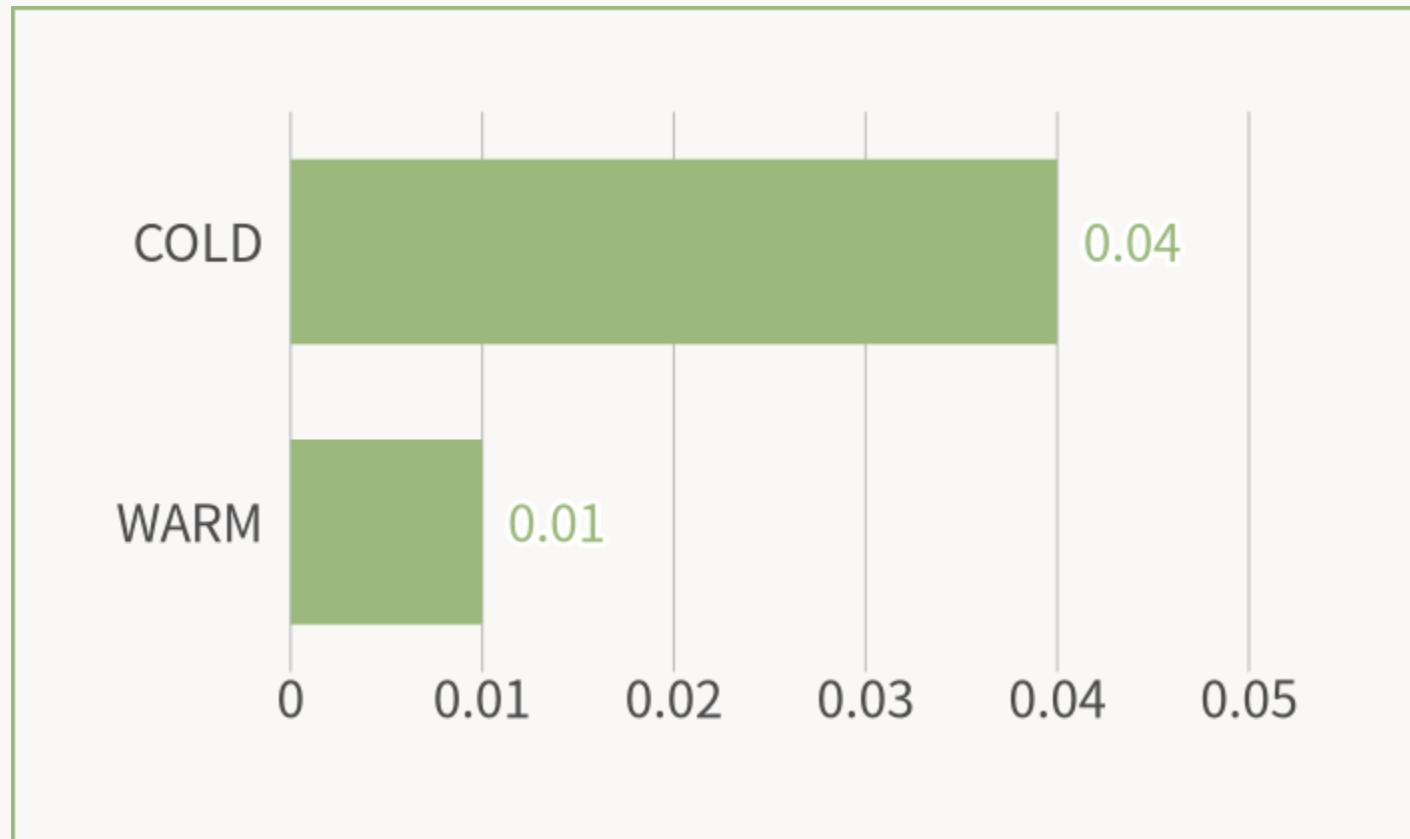
성능 개선 확인

COLD 인스턴스와 WARM 인스턴스의 응답 시간 차이는?

작업 소요 시간



성능 개선 확인



쿼리 응답 시간 개선

COLD 상태 대비 쿼리 응답 속도 약 75% 개선

※ 테스트 데이터로 진행되었기에 실제 운영 환경에서는 다를 수 있습니다.

감사합니다