배포 프로세스 및 특이사항

Canary 배포 프로세스 및 특이사항

1. Canary 배포 개요

Canary 배포는 새 버전을 점진적으로 사용자에게 노출시키는, 위험을 최소화하는 배포 전략입니다. 전체 사용자 중 일부에게만 새 버전을 제공하여 문제를 조기에 감지하고 필요시 빠르게 롤백할 수 있습니다.

2. 시스템 구성

- **프론트엔드 서비스**: Next.js (Stable: 3001 포트, Canary: 3002 포트)
- **백엔드 서비스**: Spring Boot (Stable: 8081 포트, Canary: 8082 포트)
- **Python 서비스**: FastAPI (Stable: 8010 포트, Canary: 8020 포트)
- **라우팅 서비스**: NGINX (80/443 포트, HTTP/HTTPS)
- **데이터베이스**: MySQL (3307 포트)
- **캐시 서버**: Redis (6379 포트)

3. Canary 배포 프로세스

3.1 초기 설정 확인

1. **배포 유형 결정**:
   - `stable`: 안정 버전 100% 배포
   - `canary-manual`: 수동으로 설정한 비율로 Canary 배포
   - `canary-auto`: 자동으로 단계적 비율 증가 (10% → 40% → 70%)
   - `promote-canary`: Canary 버전을 100%로 승격
   - `rollback`: 안정 버전으로 롤백
2. **NGINX 설정 관리**:
   - `upstream` 블록에서 각 서비스별 트래픽 가중치 조정
   - 서비스별 가중치를 조정하여 트래픽 분배 (weight 파라미터 사용)

3.2 배포 흐름

1. **GitLab 트리거 확인**:
   - develop 브랜치에 Push나 Merge 이벤트 감지 시 자동으로 `canary-auto` 유형으로 배포
2. **코드 체크아웃**:
   - 원격 서버와 Jenkins 서버 모두에서 최신 코드 체크아웃
   - 원격 서버에 변경사항이 있는 경우 자동 백업 및 스태시 처리
3. **NGINX 설정 준비**:
   - Canary 설정 파일 검증 및 필요시 기본 설정 생성
4. **환경 변수 설정**:
   - 필요한 모든 환경 변수를 .env 파일로 생성하여 원격 서버에 복사
5. **배포 실행**:
   - Canary 환경용 Docker Compose 파일 실행
   - 성공적인 애플리케이션 시작 확인을 위한 헬스체크 수행
6. **트래픽 전환**:
   - 유형에 따라 NGINX 가중치 업데이트:
     - `canary-manual`: 지정된 가중치로 트래픽 분배
     - `canary-auto`: 10% → 40% → 70%로 점진적 증가
     - `promote-canary`: Canary로 100% 전환
     - `rollback`: Stable로 100% 전환
7. **모니터링 및 검증**:
   - 각 단계에서 애플리케이션 상태 모니터링
   - 자동 배포의 각 단계 간 30초 대기 및 건전성 확인
8. **배포 검증**:
   - 트래픽 가중치가 의도한 대로 설정되었는지 확인

4. 특이사항 및 주요 기능

4.1 자동화된 Canary 배포

- **점진적 트래픽 이동**: 10% → 40% → 70%의 사전 정의된 단계별 진행
- **단계별 모니터링**: 각 단계마다 30초간 서비스 안정성 확인
- **자동 롤백**: 헬스체크 실패 시 자동으로 Stable 환경으로 롤백
- **환경 변수 관리**: Jenkins Credentials를 통한 보안 정보 관리

4.2 서비스 건전성 확인 메커니즘

- **다중 헬스체크 전략**:
  - API 엔드포인트 확인을 통한 직접 헬스체크
  - 실패 시 Docker 로그 확인을 통한 간접 확인
  - 최대 3회 재시도 로직으로 일시적 지연 대응

4.3 NGINX 설정 관리

- **가중치 기반 라우팅**:
  - 세 가지 서비스(백엔드, 프론트엔드, Python)에 대한 개별 가중치 설정
  - 최소 1의 가중치 유지로 0% 트래픽 상황에서도 모니터링 가능
  - 설정 자동 백업 및 변경사항 검증

4.4 배포 알림 시스템

- **Mattermost 통합**:
  - 배포 성공/실패 시 자동 알림
  - 배포 유형, 커밋 정보, 작성자, 트래픽 비율 등 상세 정보 포함
  - 실패 시 자동 롤백 정보 및 로그 링크 제공

5. 주의사항 및 제한점

1. **최소 트래픽 유지**:
   - 완전한 0% 또는 100% 설정 대신 1:999 비율 사용으로 모니터링 가능성 유지
2. **AWS S3 및 외부 API 통합**:
   - 환경 변수에 AWS 자격 증명 및 Claude API 키 포함
   - 다중 환경에서 동일한 외부 서비스 접근 필요
3. **타임존 설정**:

- 모든 컨테이너에 Asia/Seoul 타임존 설정으로 로그 시간 일관성 유지

4. **로컬 변경사항 관리**:
   - 원격 서버의 로컬 변경사항은 배포 전 자동 백업 및 스태시
5. **가중치 검증 로직**:
   - 설정된 가중치와 실제 가중치의 차이가 2 이상일 경우 오류로 처리
   - 설정 실패 시 재시도 로직 포함

6. 추가 개선 가능 사항

1. **A/B 테스트 통합**: 사용자 세그먼트 기반 라우팅 추가로 특정 사용자 그룹에 기능 출시
2. **자동화된 성능 모니터링**: 응답 시간, 오류율 등 메트릭 기반 자동 판단 로직 추가
3. **점진적 롤백**: 문제 발생 시 점진적으로 트래픽을 안정 버전으로 되돌리는 메커니즘
4. **DB 마이그레이션 관리**: 스키마 변경이 필요한 배포를 위한 관리 메커니즘

7. 사용 예시

1. **자동 Canary 배포**:
   - develop 브랜치에 코드 푸시
   - Jenkins가 자동으로 Canary 환경에 배포
   - 트래픽이 10% → 40% → 70%로 점진적 증가
   - 모든 단계에서 문제 없으면 70%에서 대기
   - 수동으로 100% 승격 여부 결정
2. **수동 롤백**:
   - 문제 발견 시 `rollback` 유형으로 파이프라인 수동 실행
   - 트래픽 즉시 Stable 환경으로 100% 전환
3. **Canary 승격**:
   - 충분한 테스트 후 `promote-canary` 유형으로 파이프라인 수동 실행
   - Canary 버전으로 트래픽 100% 전환