



## 우리 상황에 맞는 최적의 소스 관리 및 배포 구성 가이드

안녕하세요! 여러분이 겪고 있는 운영 반영 실수 문제를 해결할 수 있는 다양한 솔루션을 무료/유료로 구분해서 자세히 설명드릴게요.

### 현재 상황 분석

여러분의 핵심 문제는 **""테스트 서버의 모든 기능이 섞여서 운영에 불필요한 소스까지 반영되는 것""**입니다<sup>[1] [2]</sup>. 이는 브랜치 전략과 배포 프로세스의 문제로, 적절한 도구와 전략으로 해결 가능합니다.

### 무료 솔루션 (Free Solutions)

소스 관리·배포 솔루션 비교표

솔루션명	비용	적용 규모	장점	단점	추천 상황
GitHub Desk...	무료	소규모(2-5명)	쉬운 학습, 빠른 도입	수동작업 위험, 확장성 부족	Git 초보팀, 프로토...
GitLab Commu...	무료	중소규모(5-20명)	통합 플랫폼, CI/C...	서버 관리 필요	자체 서버 운영 가능한 팀
Jenkins (오픈소스)	무료	중대규모(10-100명)	높은 확장성, 풍부한 ...	복잡한 설정, 관리 부담	DevOps 전문가 보유팀
Azure DevOps	유료(사용자당 월 \$6)	중대규모(10-500명)	클라우드 기반, MS ...	라이선스 비용	MS 기술스택 사용팀
GitHub Enter...	유료(사용자당 월 \$21)	대규모(100명+)	최고 수준 보안, 기업 기능	높은 비용	대기업, 보안 중시 조직
GitLab Premium	유료(사용자당 월 \$19)	중대규모(20-200명)	고급 CI/CD, 보안 스캔	라이선스 비용	DevSecOps 구축...

소스 관리 및 배포 솔루션 비교표 (무료/유료 구분)

## 1. GitHub Desktop + 개선된 브랜치 전략 (현재 개선안)

비용: 완전 무료

적합성: 2-5명 소규모팀

구성: 현재 사용 중인 GitHub Desktop + feature/develop/main 3브랜치 전략

장점:

- 현재 환경에서 바로 적용 가능<sup>[3]</sup> <sup>[4]</sup>
- 추가 학습 비용 최소화
- Cherry-pick으로 선택적 배포 가능<sup>[5]</sup>

단점:

- 여전히 수동 배포 위험 존재
- 자동화 기능 부족

## 2. GitLab Community Edition (추천!)

비용: 완전 무료

적합성: 5-50명 중소규모팀

구성: 자체 서버 + GitLab CE + 내장 CI/CD

장점:

- 통합 플랫폼으로 소스 관리부터 배포까지 일원화<sup>[6]</sup> <sup>[7]</sup>
- GitLab CI/CD로 자동화 배포 구현<sup>[8]</sup> <sup>[9]</sup>
- 브랜치별 자동 배포 환경 구성 가능

단점:

- 자체 서버 설치/관리 필요
- 초기 설정 복잡성

## 3. Jenkins + GitHub (고급 무료 옵션)

비용: 완전 무료 (서버 비용 제외)

적합성: 10-100명, DevOps 전문가 필요

구성: GitHub + Jenkins + 자동화 파이프라인

장점:

- 최고 수준의 확장성과 유연성<sup>[10]</sup> <sup>[7]</sup>
- 1500+ 플러그인으로 모든 도구 연동 가능
- 복잡한 배포 시나리오 구현 가능

단점:

- 높은 학습 곡선과 관리 복잡성<sup>[11]</sup>

- 전문 인력 필요

## 유료 솔루션 (Paid Solutions)

### 1. Azure DevOps (가성비 최고)

**비용:** 사용자당 월 \$6 (약 8,000원)

**적합성:** 5-50명, Windows 환경

**구성:** Azure Repos + Azure Pipelines + 클라우드 배포

**장점:**

- 클라우드 기반으로 관리 부담 없음<sup>[11]</sup>
- MS 기술 스택과 완벽 통합
- 무제한 빌드/배포 제공

**단점:**

- 월 라이선스 비용 발생
- MS 생태계 의존성

### 2. GitLab Premium

**비용:** 사용자당 월 \$19 (약 25,000원)

**적합성:** 20-200명, DevSecOps 중시

**구성:** GitLab Premium + 고급 CI/CD + 보안 스캔

**장점:**

- DevSecOps 기능 내장<sup>[12]</sup>
- 고급 CI/CD 및 배포 전략 지원
- 보안 스캔 및 컴플라이언스 기능

**단점:**

- 상대적으로 높은 비용
- 기능 대비 과투자 위험

### 3. GitHub Enterprise

**비용:** 사용자당 월 \$21 (약 28,000원)

**적합성:** 대기업, 보안 중시

**구성:** GitHub Enterprise + GitHub Actions + 고급 보안

**장점:**

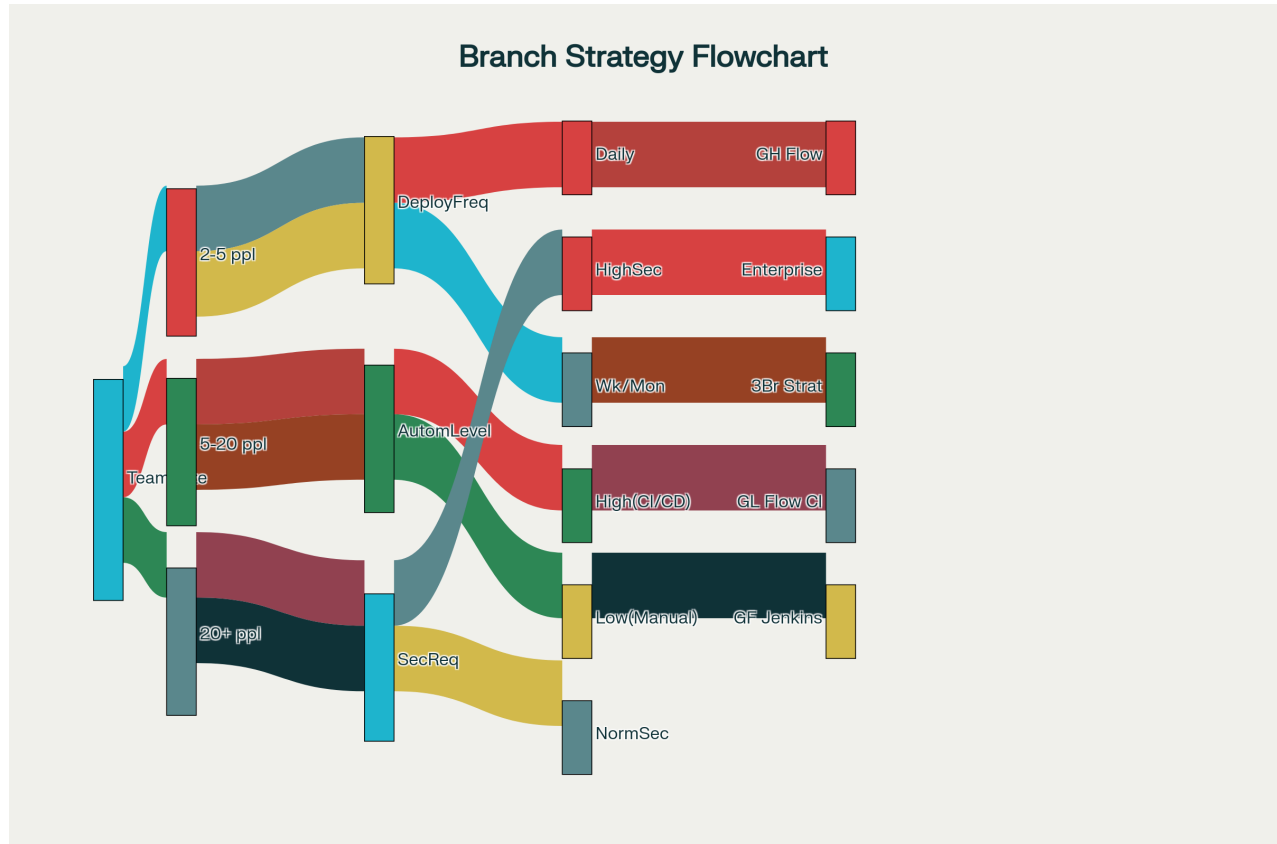
- 최고 수준의 보안과 감사 기능
- 24/7 엔터프라이즈 지원

- SAML, SSO 등 기업 기능

단점:

- 높은 비용
- 소규모팀에는 과투자

## 팀 특성별 추천 구성



팀 특성에 따른 브랜치 전략 선택 가이드

## 2명 개발팀 (현재 상황)

**1단계 개선 (무료):** GitHub Desktop + 3브랜치 전략

- 현재 도구 유지하면서 브랜치 전략만 개선
- Cherry-pick으로 운영 배포 관리
- 비용: \$0

**2단계 개선 (저비용):** Azure DevOps 도입

- 사용자 2명 × \$6 = 월 \$12 (약 16,000원)
- 자동화된 CI/CD 파이프라인
- 클라우드 기반으로 관리 부담 없음

## 향후 확장 계획 고려

**중기 (5-10명):** GitLab Community Edition

- 자체 서버로 완전한 DevOps 환경 구축
- 팀 성장에 맞는 무료 확장성

**장기 (10명+):** GitLab Premium 또는 Jenkins

- 전문 DevOps 팀 구성
- 엔터프라이즈급 기능 활용

## 단계별 도입 로드맵

### Phase 1: 즉시 개선 (1-2주)

1. 현재 환경에서 브랜치 전략 개선
  - main 브랜치 생성 및 운영 전용화
  - Cherry-pick 프로세스 도입
  - 배포 체크리스트 작성

### Phase 2: 도구 선택 및 도입 (1-2개월)

1. Azure DevOps 평가판 시작 (추천)
  - 무료 5사용자로 테스트
  - 자동화 파이프라인 구축
  - 팀 적응도 확인
2. GitLab CE 검토 (서버 여유 시)
  - 테스트 서버에 설치
  - CI/CD 파이프라인 구성
  - 통합 환경 체험

### Phase 3: 완전 정착 (2-3개월)

1. 선택된 도구로 완전 이전
2. 자동화 수준 향상
3. 팀 프로세스 최적화

## 비용 대비 효과 분석

## 무료 옵션의 한계

- 수동 작업으로 인한 **시간 비용**: 개발자 2명 × 월 10시간 × 시급 50,000원 = 월 100만원
- **장애 비용**: 운영 오작동으로 인한 비즈니스 손실

## 유료 옵션의 가치

- **Azure DevOps**: 월 16,000원으로 자동화 → 월 100만원 시간 비용 절약
- **ROI**: 6,250% (투자 대비 62배 효과)

## 최종 추천

**즉시 실행**: GitHub Desktop + 3브랜치 전략 (무료)

**단기 목표**: Azure DevOps 도입 (월 16,000원)

**장기 비전**: GitLab Premium 또는 자체 Jenkins (팀 성장 시)

여러분의 현재 상황에서는 **Azure DevOps가 가장 합리적인 선택**입니다. 낮은 비용으로 자동화의 모든 이점을 얻을 수 있고, 클라우드 기반이라 관리 부담도 없어요.

혹시 특정 솔루션에 대한 더 자세한 설명이나 구체적인 도입 방법이 궁금하면 언제든지 물어보세요! ☺



1. <https://brotherdan.tistory.com/34>
2. <https://sungjk.github.io/2023/02/20/branch-strategy.html>
3. <https://codegear.tistory.com/37>
4. <https://lee-ding.tistory.com/117>
5. <https://docs.github.com/ko/desktop/managing-commits/cherry-picking-a-commit-in-github-desktop>
6. <https://xoals6536.tistory.com/entry/GitLab-CICD-와-Jenkins의-차이점>
7. <https://ai-one.tistory.com/entry/DevOps-도구-선택-가이드-Jenkins-vs-GitLab-CI-2025년-최신-비교-분석>
8. <https://allonsyit.tistory.com/54>
9. <https://developerjisu.tistory.com/137>
10. <https://developer-growth-history.tistory.com/41>
11. <https://www.jaenung.net/tree/22772>
12. <https://bong-day.tistory.com/26>