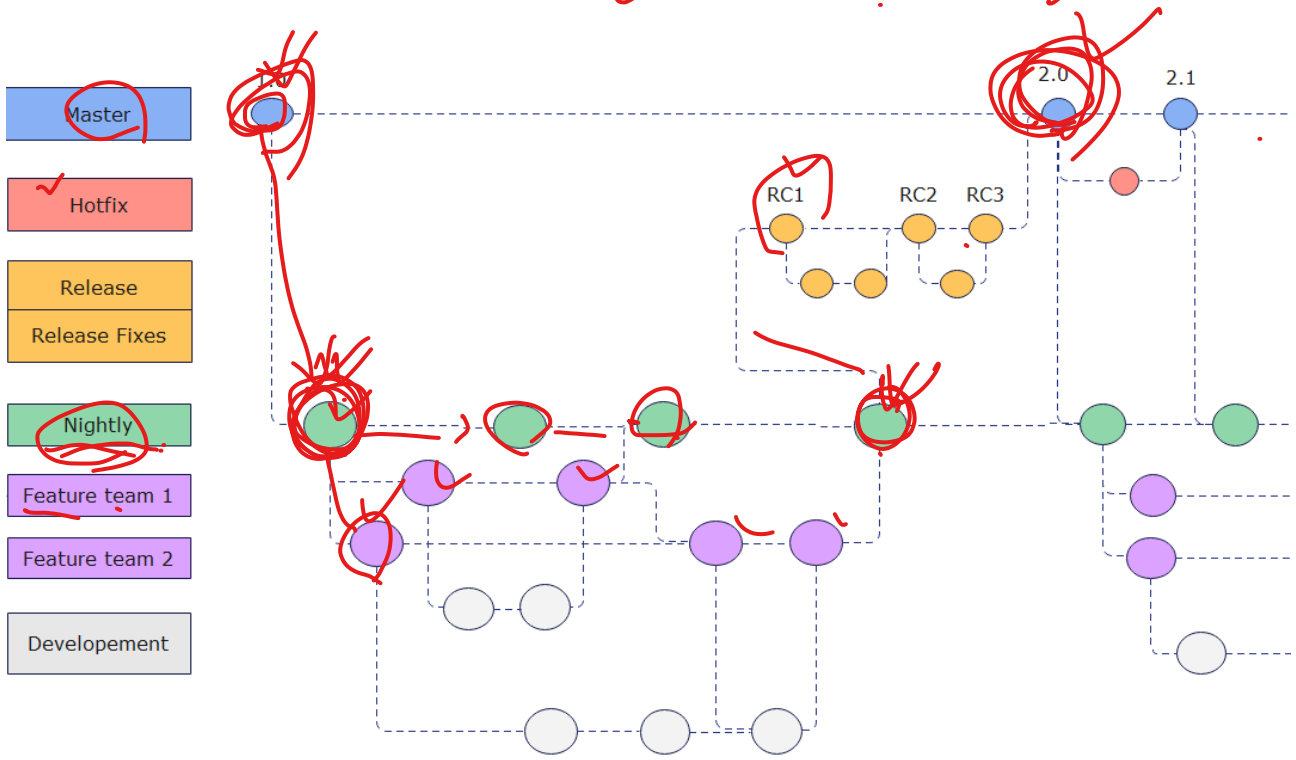
4장 브랜치

김철학

목차

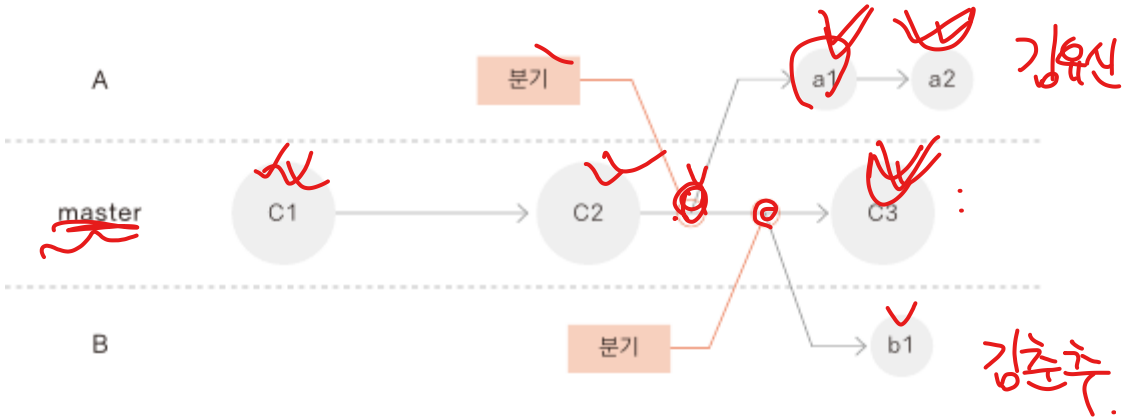
1. 브랜치 개요

- 브랜치Branch 는 버전 관리 시스템에서 나뭇가지 처럼 파생되는 문서 버전을 의미
- Git은 master 또는 main 기본 브랜치를 사용
- 브랜치의 분할과 병합 과정을 통해 문서 버전을 협업하고 관리



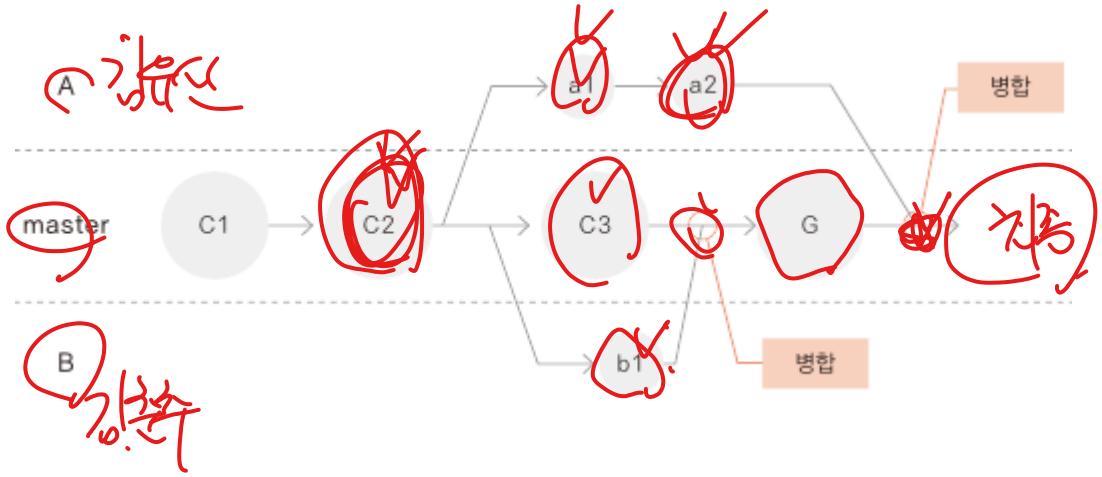
2. 브랜치 분할

- 브랜치 분기는 새로운 브랜치를 생성하여 코드의 독립적인 작업 라인을 만드는 과정
- 브랜치 분기는 새로운 기능을 추가하거나 버그를 수정하는 동안 메인 코드에 영향을 주지 않고 독립적으로 작업을 수행



3. 브랜치 병합

- 브랜치 병합은 분기된 브랜치에서 작업한 내용을 메인 브랜치 또는 다른 브랜치에 통합하는 과정
- 브랜치 병합은 개발 완료된 기능이나 버그 수정 사항을 코드베이스에 반영하여 최종 제품에 포함시키는 단계
- 브랜치 병합 과정에서 발생하는 충돌은 협업 개발에서 자연스러운 부분이며, 충돌을 방지하기 위해 자주 병합하여 예방



4. 브랜치 전략

- 브랜치 전략은 버전 관리 시스템에서 소스 코드의 변경 사항을 체계적으로 관리하기 위해 사용하는 방법론
- 브랜치 전략은 다수의 개발자가 팀 프로젝트 환경에서 개발, 테스트, 배포를 보다 효율적으로 진행하기 위한 규칙을 제공
- 프로젝트 규모, 개발 주기 및 조직 등에 따라 달라질 수 있으며, 프로젝트의 성공적인 개발과 배포를 위해 중요한 역할 수행

종류	설명
Git Flow	• master, hotfix, release, develop, feature 총 5개 branch를 운영하는 방식 • 배포와 개발을 명확히 구분하고, 다양한 브랜치를 활용하여 체계적으로 관리하는 전략
GitHub Flow	• master, feature 총 2개 branch를 운영하는 방식 • 단순하고 직관적인 브랜치 전략으로, 모든 변경 사항을 main 브랜치로 통합 • GitHub의 특성에 맞는 브랜치 전략으로 push와 pull request를 통한 merge 수행
GitLab Flow	• main, pre-prod, prod, feature 브랜치를 운영하는 방식 • GitHub Flow와 유사하지만 환경별 브랜치를 추가한 전략