

Git 브랜치 전략

Git Branch Strategy

류창선



Git Flow

Github Flow



Gitlab Flow



TBD

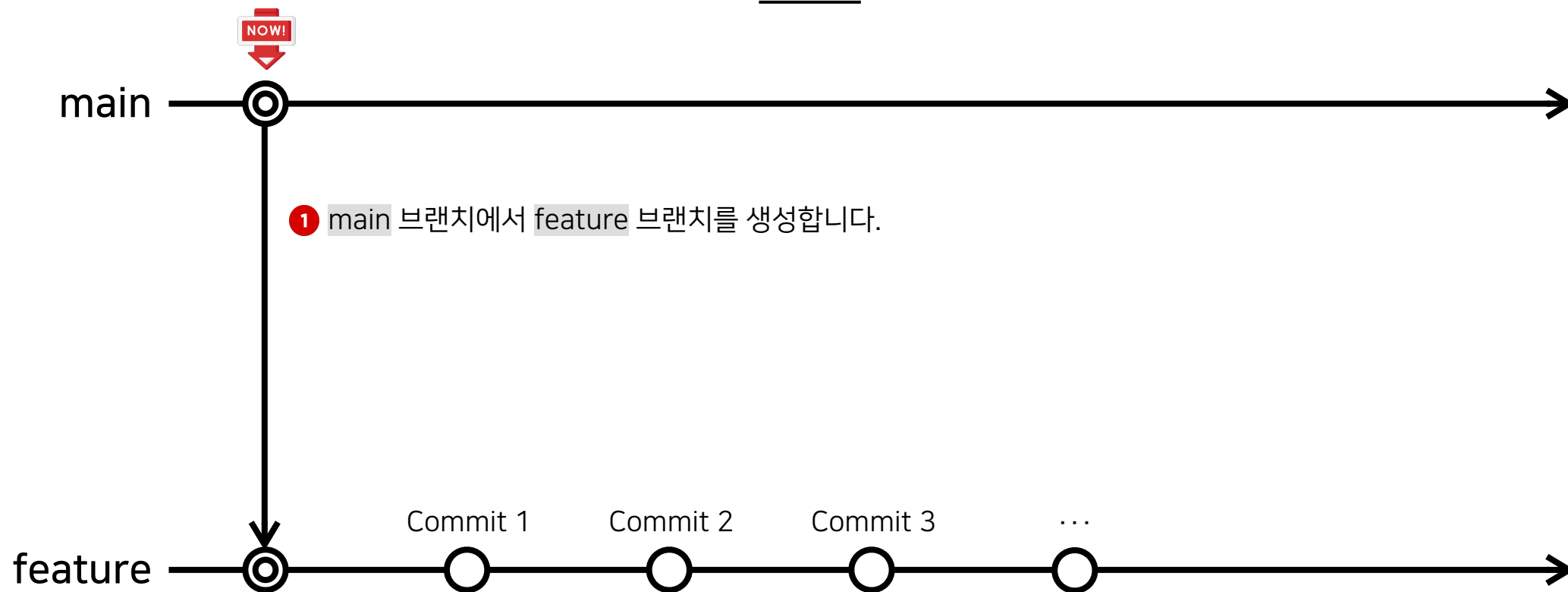


Git 브랜치 전략

현재 기준



dev



1 main 브랜치에서 feature 브랜치를 생성합니다.

2 feature 브랜치에서 개발 및 자체 테스트를 진행하면서 Commit을 쌓습니다.



dev

main



- 3 feature 브랜치에서의 개발이 완료되면 main 브랜치로 MR을 날립니다.
- 4 MR 완료 후 개발계 배포하여 개발계 테스트를 진행합니다.
- 5 테스트 완료 시 uat / prod 환경의 소스 코드로 이동합니다.

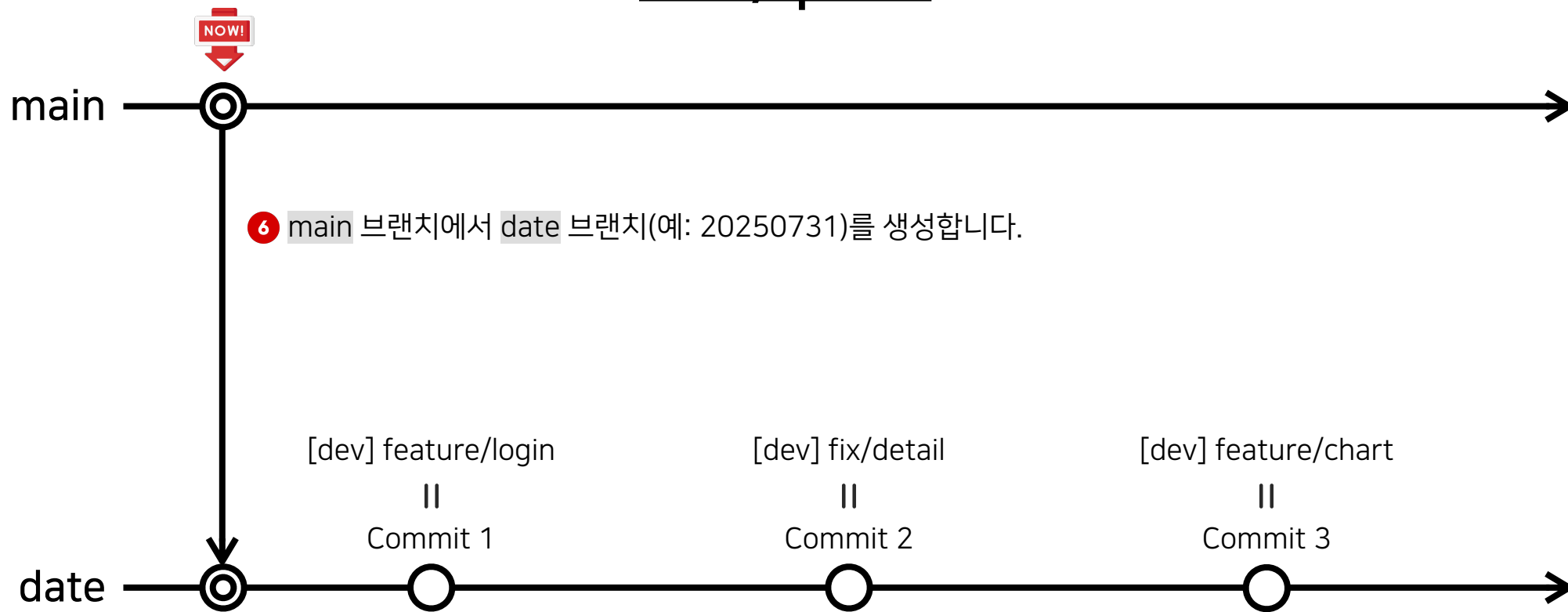
feature



Last Commit



uat / prod

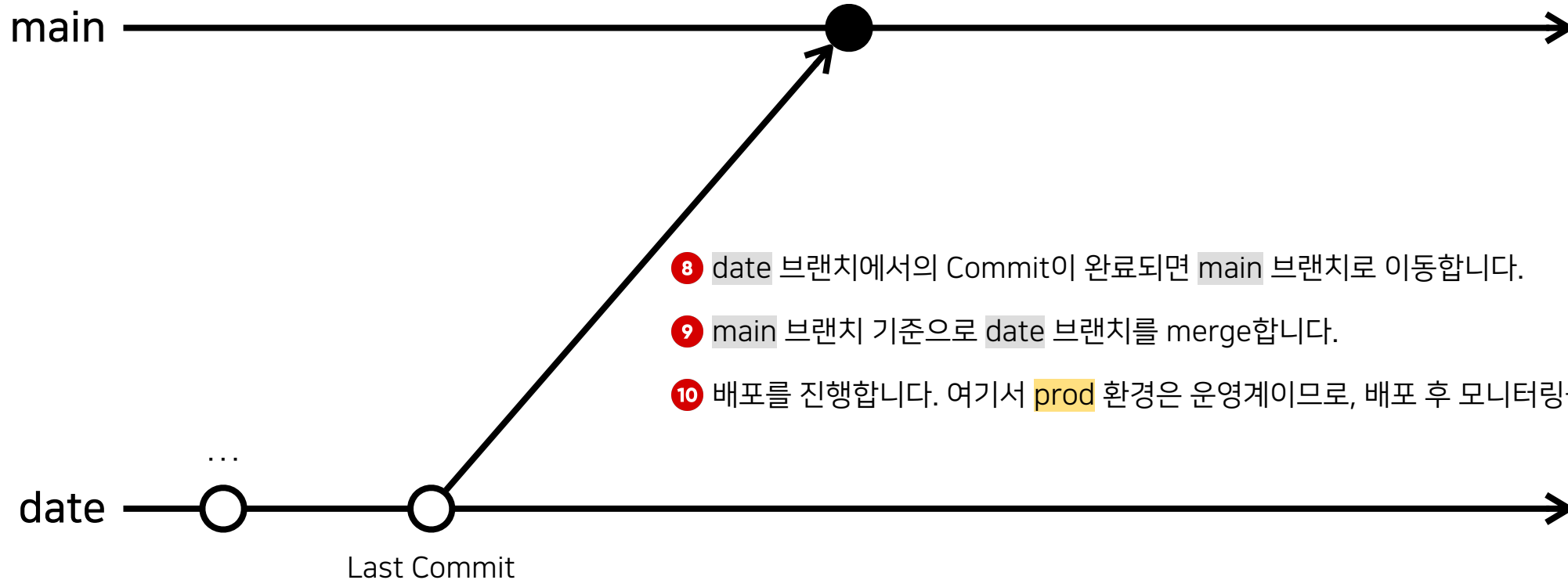


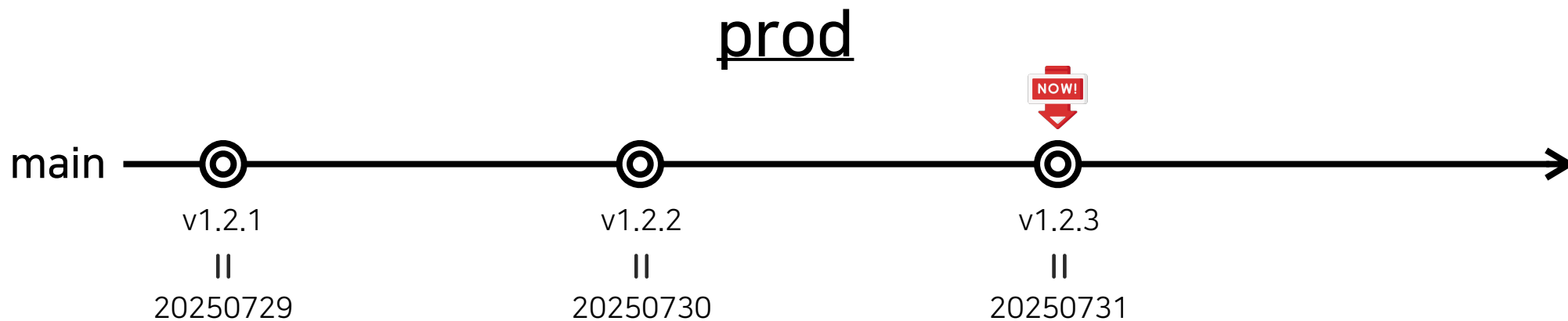
6 main 브랜치에서 date 브랜치(예: 20250731)를 생성합니다.

7 date 브랜치에 dev 환경에서 작업한 feature 브랜치의 최종 작업 내역을 하나씩 Commit합니다.



uat / prod





- 11 방금 배포한 서비스 버전을 v1.2.3이라고 가정합니다.
- 12 운영 배포 후 장애가 발생하여 Rollback을 진행해야 하는 상태입니다.
- 13 운영 배포 후 안정화 모니터링 단계를 거쳤던 v1.2.2를 Rollback 시점으로 삼습니다.
- 14 v1.2.2의 hash를 확인합니다.
- 15 git revert hash 명령어를 사용해 Rollback한 후 재배포하여 장애 발생 여부를 확인합니다.

Gitlab 통합 이후 기준



Git 브랜치 전략

Gitlab CI/CD

