

Branch model & pipeline example A

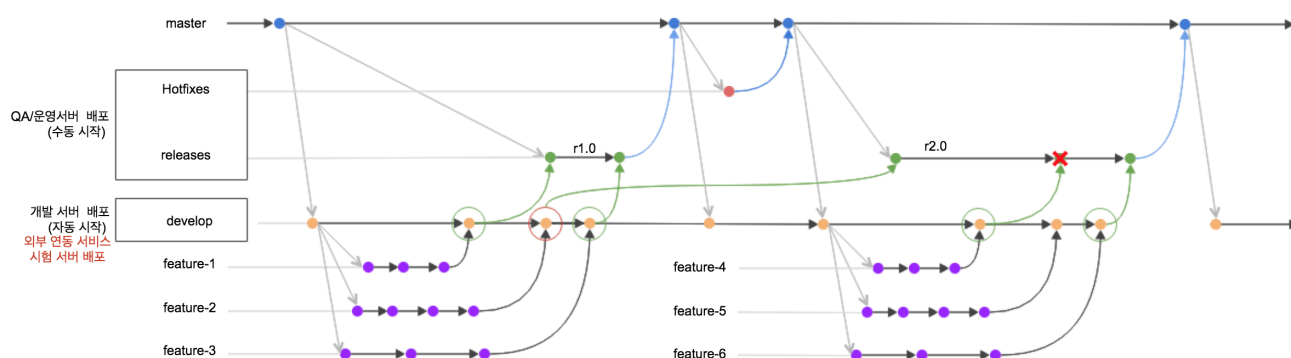
이 페이지는 Bitbucket & Bamboo를 활용한 브랜치 모델의 예로 작성되었다.

- Branch model
 - 고객 요구사항
- 제안)
 - 업무 순서
 - 브랜치 역할
 - master
 - Hotfixes
 - Releases
 - Develop
 - Features
- Branch and Pipeline Mapping

Branch model

고객 요구사항

- 구현된 기능의 일부를 선택하여 배포할 수 있어야 한다
- 긴급히 수정하여 배포할 수 있어야 한다
- (optional) Rollback 기능 고려



제안)

업무 순서

Step 0) 배포된 코드가 master 브랜치에 반영됨

Step 1) Develop 브랜치 동기화

- master의 내용을 develop 브랜치에 반영 (merge)

Step 2) 기능별 구현

- feature 브랜치를 생성하여 기능 구현
- 로컬 커밋 → 원격 (서버) push

Step 3) Develop 브랜치에 feature 개발 내용 반영

- 개발 프랜치에 merge (feature-* >> Develop)
- 빌드 성공시 개발 테스트 서버로 배포됨

Step 4) release (배포) 브랜치 구성

- 배포할 feature에 해당하는 commit을 develop 브랜치에 merge (cherry-pick >> push >> merge)
- 조건에 따라 merge된 commit 제거
- merge 확인을 위해 build 검증 (자동 시작)
- 운영 서버에 배포

Step 5) master 브랜치 갱신

- 배포된 코드를 master 브랜치에 반영 (release >> master 브랜치 merge)

Step 7) master를 기반으로하는 브랜치 갱신

Step 8) Hotfix 배포 (선택)

- 필요에 의해 hotfix 배포를 위해 master로부터 hotfix 브랜치 생성
- 수정 내용 반영
- QA/운영서버에 반영

브랜치 역할

master

역할:

- 운영되고 있는 서비스와 일치하는 소스코드 유지
- 즉시 배포가 가능한 소스코드 유지
- 배포된 소스코드가 merge되면 release version tagging 권장
- 변경이 반영되면 master를 base로 생성된 branch들이 갱신되어야 함

배포 대상:

- 브랜치로부터 직접 배포하지 않음 → 긴급한 수정 반영이 필요시 Hotfix 브랜치 생성하여 배포

생명 주기:

- 영구적으로 유지

Hotfixes

역할:

- 긴급한 수정 후 배포를 위한 역할

배포 대상:

- QA (Staging) 및 운영서버

생명주기:

- 유지하지 않음
- 배포 후 master에 merge하고 삭제

Releases

역할:

- 주기적인 배포 소스코드 관리
- 최신 개발 소스 (develop)로부터 배포할 commit을 선택하여 구성

배포 대상:

- QA (Staging) 및 운영서버

생명주기:

- 정기 배포 후 삭제 또는 지속적으로 유지

Develop

역할:

- 최신 개발 코드 유지
- 개발 테스트 서버 배포하여 기능 검증

배포 대상:

- 개발 테스트 서버
- 외부 서비스와의 연동 검증용 서버

생명주기:

- 배포 주기의 3배 이상 운영하지 안길 권장

Features

역할:

- 단위 기능 개발용 브랜치

배포 대상:

- 없음

생명주기:

- 기능 개발이 완료되고 develop 브랜치에 merge 후 삭제

Branch and Pipeline Mapping

Branch	Pipeline	Trigger	배포 대상	역할
feature-N	Commit build	Commit Merge	x	빌드 검증 코드 품질 점검 테스트 자동화 수행 (옵션)
develop	Commit build	Commit Merge	개발 서버	빌드 검증 (증분) 코드 품질 점검 테스트 자동화 수행 (옵션) 개발 서버에 자동 배포
	Daily build	Scheduled	개발 서버	빌드 검증 (클린) 코드 품질 점검 테스트 자동화 수행 (옵션) 개발 서버에 자동 배포
releases	Commit build	Commit Merge	x	빌드 검증 (증분) 코드 품질 점검 테스트 자동화 수행 (옵션)
	QA/Production	빌드: • Commit • Merge 배포: • 수동	QA 서버 Production 서버	빌드 검증 (클린) 코드 품질 점검 테스트 자동화 수행 (옵션) 자동 배포 1) QA서버 배포 2) Production 서버 배포 (수동)
hotfixes	Commit build	Commit Merge	x	빌드 검증 (증분) 코드 품질 점검 테스트 자동화 수행 (옵션)
	QA/Production	빌드: • Commit • Merge 배포: • 수동	QA 서버 Production 서버	빌드 검증 (클린) 코드 품질 점검 테스트 자동화 수행 (옵션) 자동 배포 1) QA서버 배포 2) Production 서버 배포 (수동)
master	Commit build	Commit Merge	x	빌드 검증 (증분) 코드 품질 점검 테스트 자동화 수행 (옵션)
	Daily	Scheduled	x	빌드 검증 (클린) 자동 문서 작성 코드 품질 점검 테스트 자동화 수행 (옵션)

