Git 이란?

파일의 변경 사항 추적 및 여러 사용자들이 같은 파일에서 작업하는 것을 조율하는 ‘분산 버전 관리 시스템’ 이다.

매우 빠른 수행 속도, 데이터 무결성, 분산 처리, 비선형 워크플로우가 특징

로컬에서 중앙 서버와 독립적으로 이력 및 버전 관리가 가능하다.

GNU License를 가진 자유 소프트웨어이다.

기본적으로 Linux & CLI 환경에서 동작하나 여러 OS 및 GUI 툴이 존재

Git 기초

Git은 모든 데이터를 체크섬을 구해 관리한다.(SHA-1 알고리즘 이용)

Git의 로컬 상태는 Committed, Modified, Staged 3가지다.(git add <file> 이후)

Committed : 로컬 데이터베이스에 안전하게 저장 완료.

Modified : 수정되었으나 아직 로컬 데이터베이스에 commit 되지 않음.

Staged : 수정된 파일을 곧 commit 할 것이라 표시.

git add <file>을 하여 등록된 이후부터 관리가 시작되는데, .gitignore 파일을 이용하여 디렉토리 단위로 저장할 때 필요 없는 파일을 저장하지 않을 수 있다.(log, preview, obj 등)

로컬에서 commit 된 파일들을 원격 저장소에 push하기 위한 방법

(파일 add) -> (파일 commit) -> (파일 push)

원격 저장소 기반 프로토콜

1. File system : 매우 느린 편(NFS나 로컬 시스템)
2. SSH(사용 예정) : 암호화 전송, 빠른 편
3. HTTPS : SSH와 유사하나 외부 접근 용이, 구축 어려움.
4. Git protocol : 매우 빠르나, 보안 x, 외부 접근 불가

기본적으로 SSH는 공개키를 만들고, 접근할 유저를 등록하여 쓰기 권한을 주는 방식이나,

회사 서버의 외부 접근은 완전히 차단되어 있어, git repository의 소유 계정인 git의 password를 공개하여 회사 내부에서 누구나 접근 가능하도록 할 것.

Branch?

Git의 브렌치는 버전 관리를 수월하게 진행하도록 도와주는 git만의 특징이다.

Commit 된 파일들을 자유로이 오갈 수 있는 ‘포인터’이다.

기본적으로 master 라는 이름의 브렌치를 사용하고, 차후 다른 이름의 브렌치로 다른 버전의 commit 파일을 접근 할 수 있다. (현재 접근 중인 branch를 HEAD라고 한다)

안정화 버전 개발 직후, 각 브렌치를 merge할 수 있다. (merge는 다른 더 강력한 툴을 사용)

Git 호스팅 서비스

내부에 git server와 repository를 구축하고 저장하는 과정이 복잡할 수 있다.

이를 위해 gitweb, gitlab, github등의 외부 호스팅 서비스 들이 존재한다.

이는 회사의 소스코드나 문서, 파일 들이 외부 서버에 저장되는 것이고, 유료이다.

변경 이력 확인

Word나 소스 코드 같은 텍스트 기반 파일은 어느 부분이 변경되었는 지 알 수 있으나,

이미지, CAD와 같은 이진 데이터는 알 수 없다.(+ hash 함수로 암호화 되어 내용도 알기 힘들다.)

Sadasdasd

sada