

Git 프로그램 설치

git for Window 또는 git Clone 커맨드를 사용할 수 있는 프로그램을 설치해야 된다.

Visual Studio가 설치된 경우

1. Visual Studio Installer 실행
2. 수정 -> 개별 구성요소(상단탭) -> Git for Windows 설정 -> 업데이트

Visual Studio가 설치되지 않은 경우

1. <https://gitforwindows.org/> 공식 홈페이지에서 Download한다.

PC에 Git Bash가 정상적으로 설치되어 있는지 확인 및 실행동작

Visual Studio 확장프로그램에서 [Git Diff Margin] 다운로드 및 사용설정

소스 다운로드

최초 소스 및 프로젝트 정보가 없는 경우 다음 과정에 따라 소스를 다운로드 한다.

Git Bash 실행 (없는 경우 Git프로그램 설치 항목 참조)

Cd [프로젝트 폴더 경로] -> cd C:/Git/CSharp 경로 입력시 \ 대신 /로 사용해야함

경로 이동후 아래에 있는 git clone 커맨드 전체를 복사하여 붙여넣기 후 실행한다.
(붙여넣기 단축키는 Shift Insert)

git clone [다운로드할 프로젝트 URL]

```
git clone http://218.144.146.115:3000/Bonobo.Git.Server/Android.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/APServer.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/AwsServer.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/Core.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/IPAPacking.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/LanguageTool.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/LicenseSolution.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/MakeArduino.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/MakerStore.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/mBizCenter.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/mBizSolution.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/ReportEngine.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/SD.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/ServerCheck.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/SmartBackupProcess.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/SmartPush.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/SmartServer.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/SmartServerUpgrade.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/SpeechAnalyzer.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/TopBuild.git
git clone http://218.144.146.115:3000/Bonobo.Git.Server/Upgrade.git
```

기타 프로그램 설치

컴파일을 진행하기 전 아래 파일을 다운로드하여 PC에 설치한 뒤 컴파일 진행할 수 있도록 합니다.

(미설치 시 컴파일 불가능)

1. [.NET Framework 4.5.2 Developer Pack](#)
2. [Microsoft Speech Platform Runtime](#)
3. [Microsoft Speech Platform SDK](#)

소스 컴파일

```
git clone http://218.144.146.115:3000/Bonobo.Git.Server/DevelopTool.git
```

위 명령어로 DevelopTool Repository Clone한 뒤, DevelopTool/BuilderBatchFile 폴더에 있는 readme.txt 파일을 읽고 지시에 따라 실행해주시면 자동 빌드됩니다.

```
git config --global core.autocrlf true
```

배치파일 실행되지 않을 경우 **git bash**에서 해당 명령어 실행 후 다시 풀받고 실행

수동 컴파일 방법

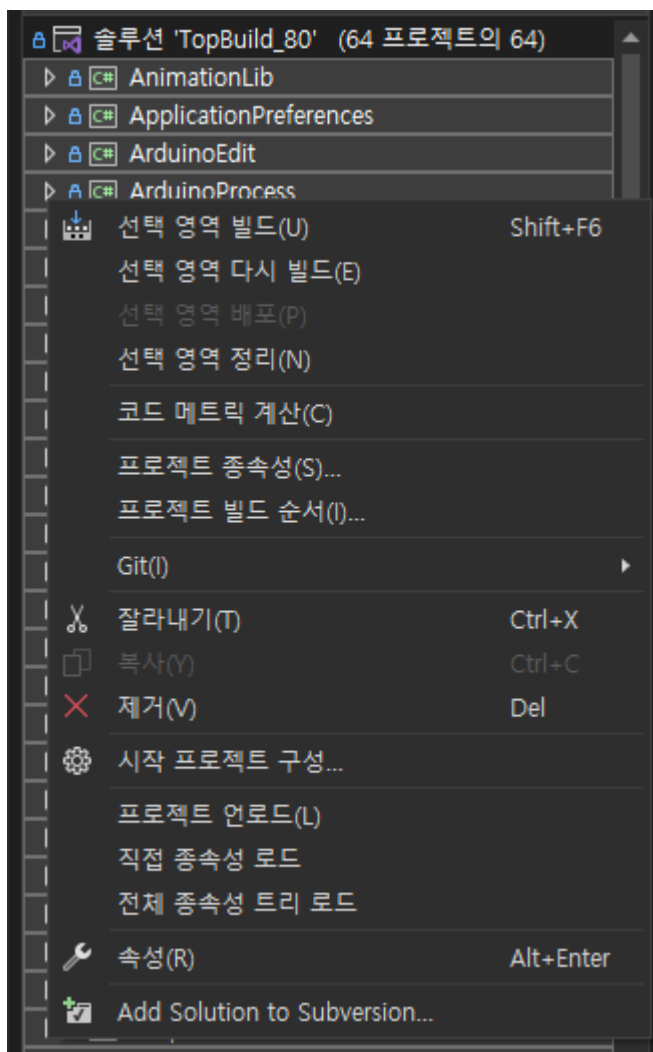
- 모든 컴파일은 **Release**로 진행해야된다.
- 컴파일 성공의 기준은 전체 다시 빌드 시 하단 출력창에 출력되는 성공 개수로 측정함
- **mBizCenter, Android**의 경우 빌드 이벤트를 위해 **Debug, Release** 각각 컴파일 해준다.
- 기본적으로 **Developer** 브랜치에서 컴파일 진행하되, 존재하지 않을 경우 **master**에서 컴파일하도록 합니다.

1. Core (전체 성공)
2. APServer (전체 성공)
3. TopBuild (12개 성공)
4. LicenseSolution (전체 성공)
5. TopBuild (43개 성공)
6. MakeArduino (전체 성공)
7. SpeechAnalyzer (전체 성공)
8. TopBuild (53성공)
9. SD (10개 성공)
10. TopBuild (58성공)
11. SmartBackupProcess (전체 성공)
12. AwsServer (전체 성공)
13. TopBuild (62성공)
14. IPAPacking (1개 성공)
15. Android (전체 성공)
16. SD (전체 성공)

- 17. LanguageTool (전체 성공)
- 18. mBizCenter (전체 성공)
- 19. TopBuild (전체 성공)

전체 다시 빌드 실행법

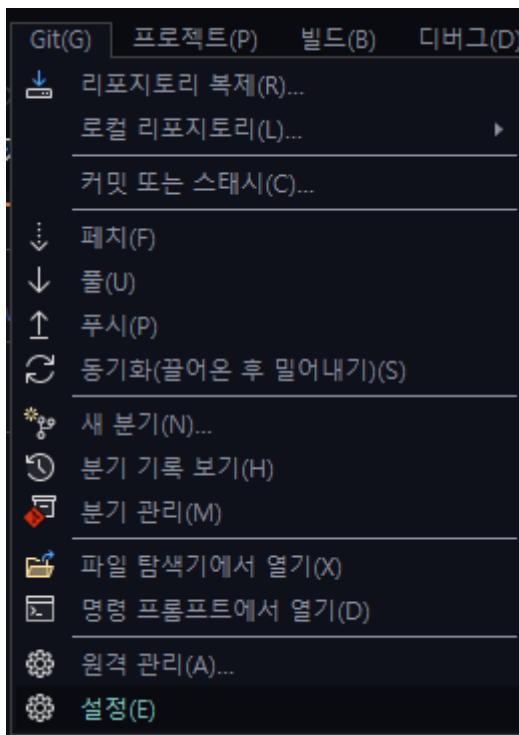
1. 화면 우측의 솔루션 탐색기 클릭 후 **Ctrl + a**로 전체선택
2. 맨 위의 솔루션 파일 클릭해 선택 해제 후 우클릭 메뉴에서 선택 영역 다시 빌드 실행

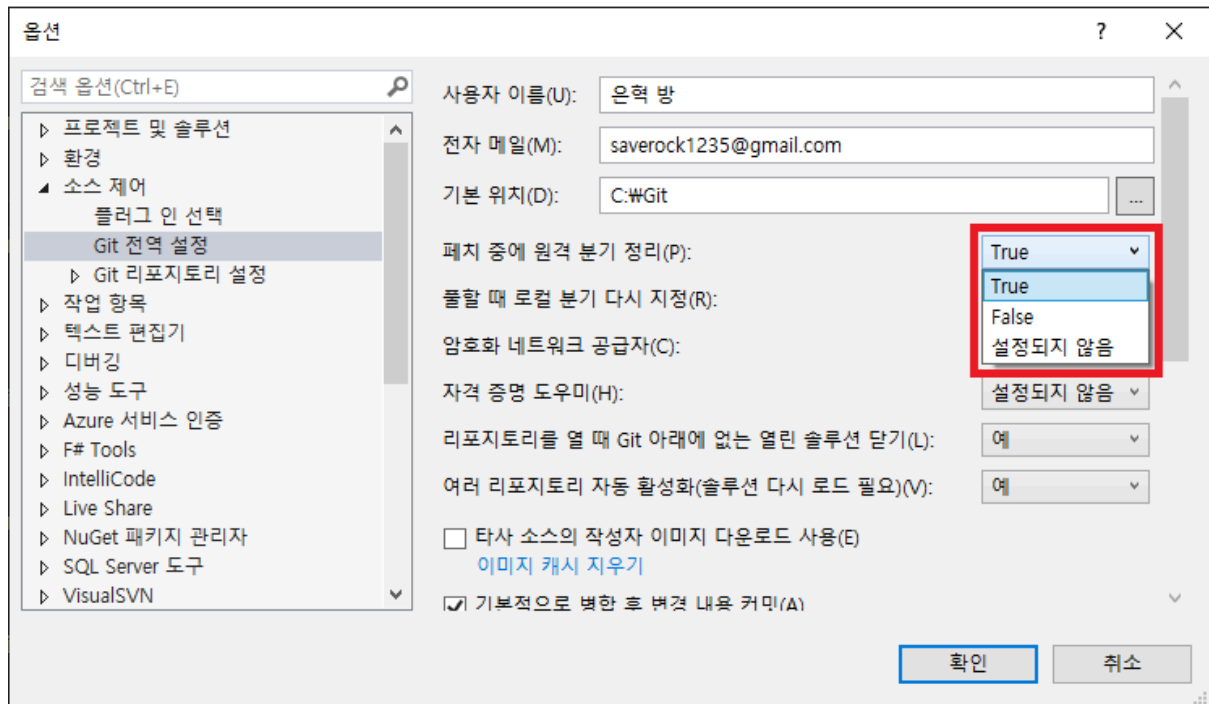


디버깅 세팅

- 아래 목록에 있는 파일 / 폴더를 디버깅할 **exe**경로 위치로 복사한다.
- 예) C:\Git\CSharp\TopBuild\SmartMaker\bin\Debug

1. C:\SmartMaker\SmartBuilder\BOS (폴더 통채로)
2. C:\SmartMaker\SmartBuilder\BOSs2.Key, ProcessQ.dic,
3. C:\Git\Chsarp\Core\ChromeBrowser101 (내용만)





상단메뉴 **Git** -> 설정 -> 페치 중에 원격 분기 정리 옵션을 **True**로 설정합니다.

[https://ghost41m.tistory.com/entry/Visual-Studio-%EB%A5%BC-%ED%95%AD%EC%83%81-%EA%B4%80\[...\]%9C%EC%9C%BC%EB%A1%9C-%EC%8B%A4%ED%96%89%ED%95%98%EA%B8%B0](https://ghost41m.tistory.com/entry/Visual-Studio-%EB%A5%BC-%ED%95%AD%EC%83%81-%EA%B4%80[...]%9C%EC%9C%BC%EB%A1%9C-%EC%8B%A4%ED%96%89%ED%95%98%EA%B8%B0)

위 링크를 따라 관리자권한 이슈 해결할 수 있습니다.

소스 관리

내부 Git서버 관리자 페이지 : <http://218.144.146.115:3000/Bonobo.Git.Server/Repository/Index>

빌더 소스 유지보수는 고객지원 / 신규기능 / 패치 및 셋업 형식으로 구분하여 처리한다.

CSRequest_(고객지원 요청)

매주 패치 이후 고객지원 브랜치를 원격에 생성하고 고객지원 작업을 진행한다.

23년 3월 첫번째 고객지원 브랜치 기준 명칭 : CSRequest_23_03_1

원격에 만들어진 고객지원 브랜치에는 고객지원 엑셀에 올라와 있는 모든 내용을 작업할 수 있다.

고객지원 엑셀에 있는 항목중 신규 기능개발로 분류해야되는경우 신규기능 브랜치에서 작업한다. 고객지원 작업중 항목처리를 위해 로컬 브랜치 생성 및 병합 처리는 작업자 재량에 따라 처리한다.

Future(신규기능)

신규기능 및 기능 개선 작업을 진행하는 경우 **Developer**를 기준으로 하는 신규 원격 브랜치를 생성한다.

진행관리자 신규기능 : **Future_ProcessManager** (Future_기능명)

Developer

프로젝트에서 항상 최종 소스는 이 프로젝트에 유지되어야한다. 고객지원 완료, 신규기능 구현 완료시 모든 소스는 **Developer**에 **Merge**해야하며 테스트 및 패치 완료시 원격에 생성되어 있는 임시 브랜치 (**Future**, **CSRequest**)를 제거한다. **Master**와 병합시 **AssemblyInfo.cs**를 변경하여 **Commit**하고 **Tag**를 추가하여 패치버전 관리를 추가한다.
긴급건으로 간단한 오류 수정하는경우 **Developer**에서 바로 수정 후 **Commit** 가능

Master

Master 브랜치의 경우 최종 패치버전, 셋업버전의 소스가 있어야 하며 패치, 셋업을 생성하는경우 **Tag**를 통해 버전표시를 처리한다. 해당 브랜치에서는 절대로 **Commit**을 진행해서는 안됨

셋업 버전 : **Version_s_6.0.64.15**

업데이트 버전 : **Version_u_6.0.64.15.01**

원격 브랜치 생성이후 일 2회 이상 **Commit**, **Push**가 진행되어야 한다.

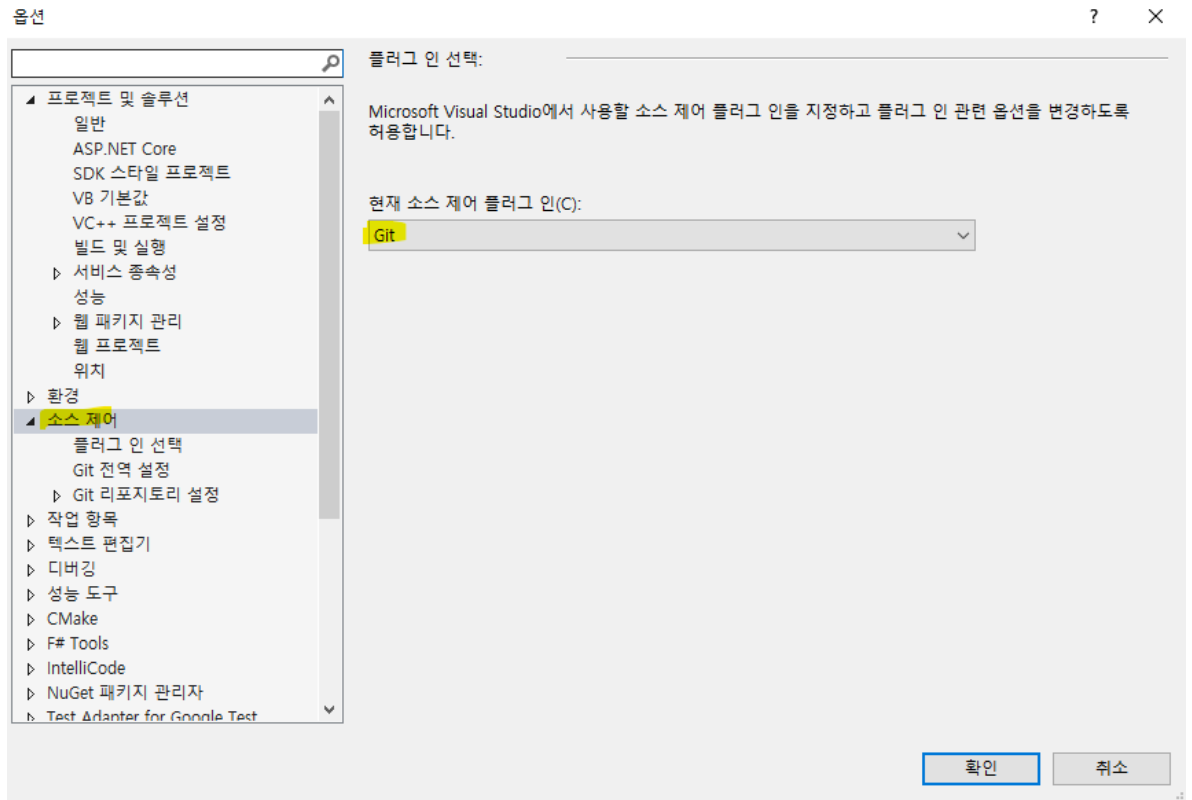
각 브랜치에 작업된 내용의 경우 **Developer**, **Master**를 제외한 나머지 브랜치는 외부로 노출되는 것이 아닌 내부 개발중을 의미하기 때문에 버그가 있는 상태에서도 **Commit**가능
중간 **Commit** 진행시 반드시 구체적인 기능 내용 및 작업 내용을 **Commit** 문구에 포함하여야 한다.

정기 **Commit**주기는 점심시간 이후, 업무보고 이후 총 2번은 의무적으로 진행한다.

Visual Studio Git 사용방법

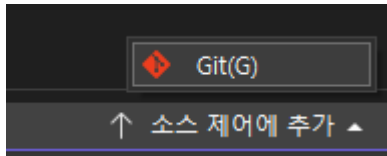
Visual Studio Git 형상관리 설정

- 도구 -> 옵션 -> 소스 제어 -> Git 선택



Visual Studio 프로젝트 Git 리포지토리(Repositories) 연결 방법

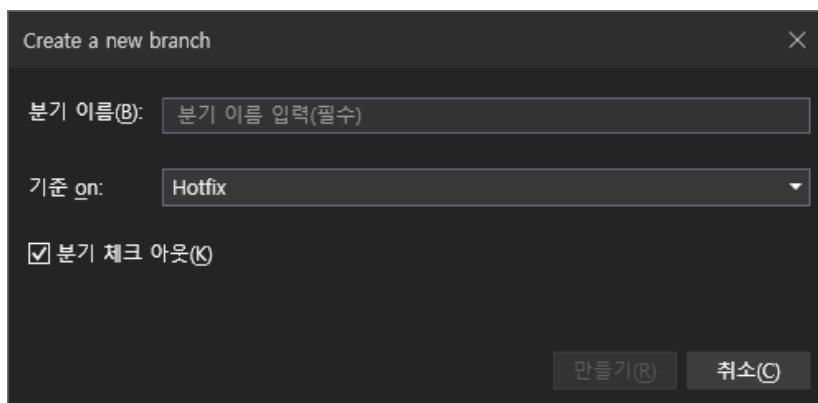
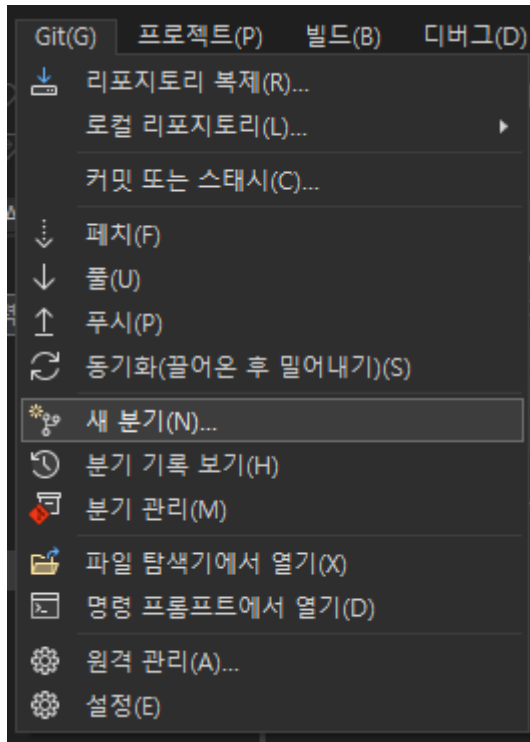
우측하단 소스 제어에 추가 클릭 -> Git -> 기존 원격 -> 기존 원격으로 코드 푸시 설정



브랜치(branch) 생성 방법

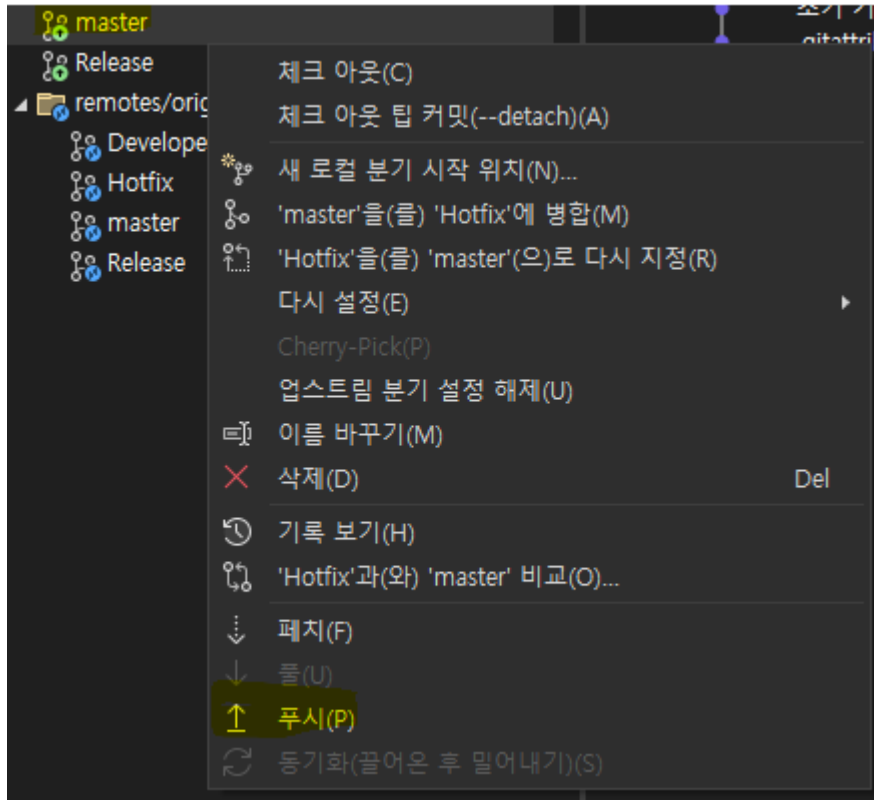
- 로컬에 브랜치 생성

- 솔루션 탐색기에서 프로젝트 하나 선택 (초점만 이동되면됨)-> Git -> 새 분기 -> 복사할 브랜치 선택 -> 만들기



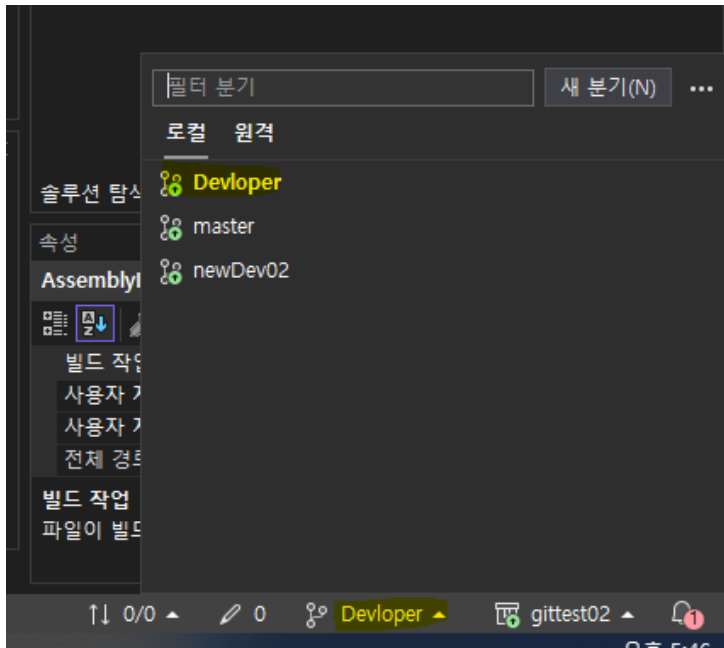
● 서버에 브랜치 생성

- 로컬에 브랜치 생성이후 **Git -> 분기관리 이동**
- 우측 분기문에서 로컬에 있는 브랜치를 **Push**를 통해 서버로 이동
- **remotes/origin** 하위 항목으로 신규 브랜치 추가된것 확인



브랜치(branch) 병합 방법

작업한 newDev02를 Developer에 병합하는 경우 다음과 같은 과정으로 처리한다.
병합할 브랜치인 Developer로 체크아웃한다.



상단메뉴 Git -> 분기 관리

병합할 프로젝트인 newDev02우클릭 -> 'newDev02'를 'Developer'에 병합 클릭 -> 소스 충돌 등 문제 발생 시 문제해결 -> Commit -> Push

