

알유컴공?

알아두면 유용한 컴퓨터 공부

WAGI

Git 사용법1



01 Git과 Github

Git과 Github의 차이점

02 Git 다운로드

Git 다운로드하는 방법

01

Git과 Github

- **Github** : Github라 불리는 회사에서 서비스하고 있는 서버에 올라간 Git
개인 계정 등록 필요
인터넷 연결 필요
- **Git** : 개인 컴퓨터에서 돌아가는 Verion Control System(버전 관리 시스템)
개인 계정 등록 필요X
인터넷 연결 필요X



Github는 Git 소프트웨어를 지원하는 일종의 클라우드 서비스

02

Git 다운로드

링크 : <https://git-scm.com/download/>

git --local-branching-on-the-cheap

Search entire site...

About

Documentation

Downloads

GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

macOS **Windows** Linux/Unix

Latest source Release
2.40.1
[Release Notes \(2023-04-17\)](#)
Download for Windows

Older releases are available and the [Git source repository](#) is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

Git via Git

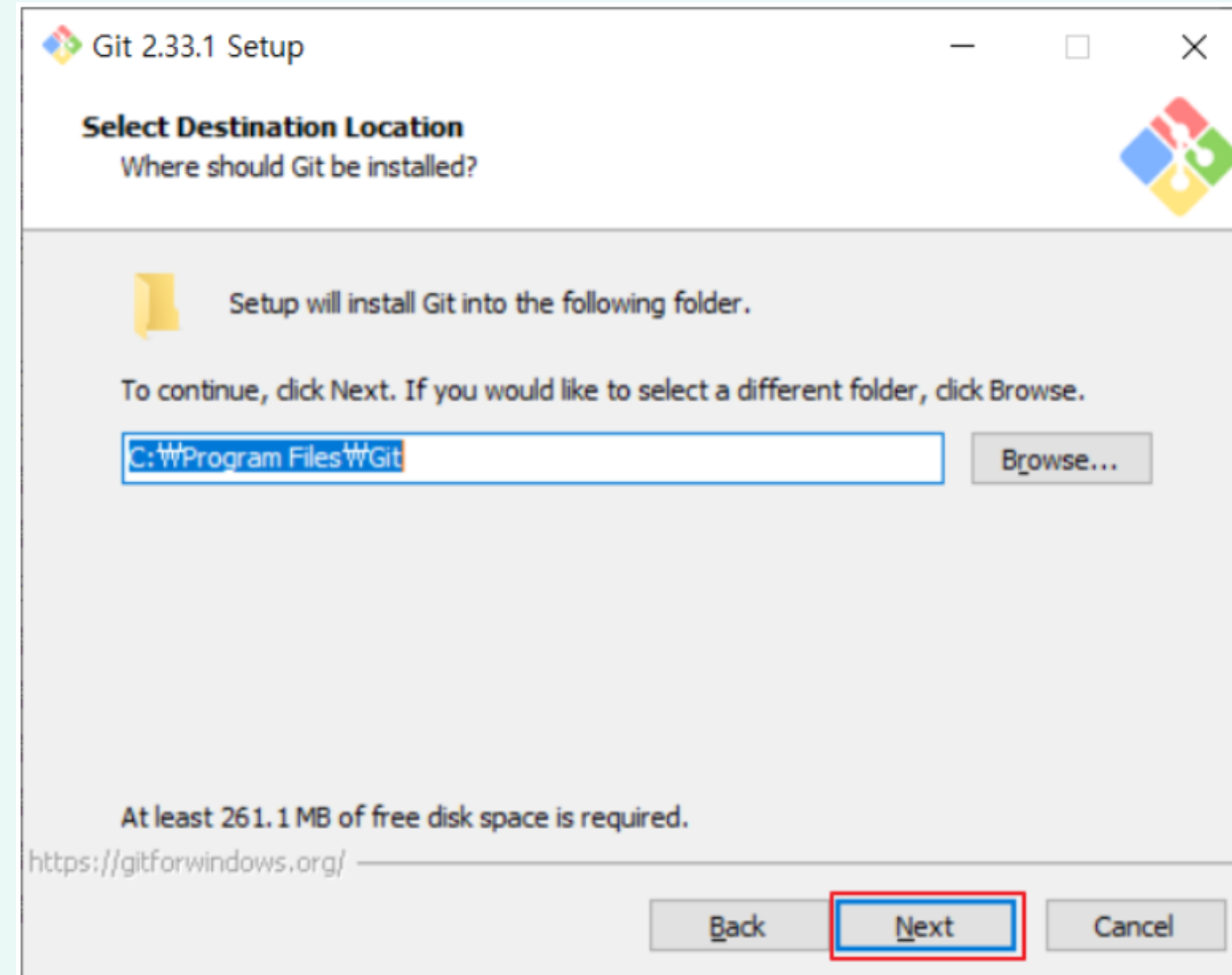
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

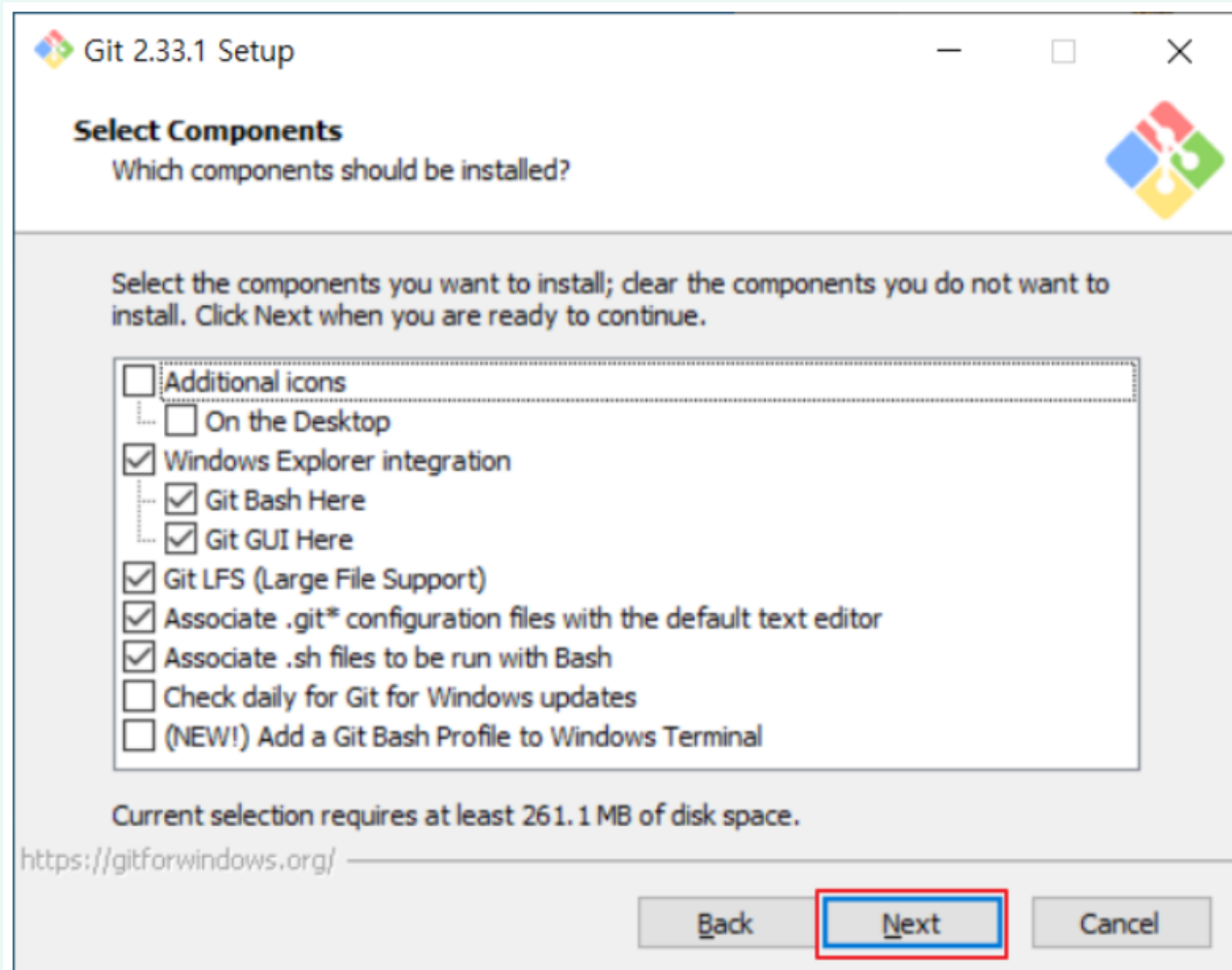
02

Git 다운로드



02

Git 다운로드



◆ Additional icons

- On the Desktop : 바탕화면에 바로가기 생성

◆ ☒ Windows Explorer integration

- ☒ Git Bash Here : 폴더 오른쪽 클릭 메뉴에 Git Bash Here(연결 기능) 추가
- ☒ Git GUI Here : 폴더 오른쪽 클릭 메뉴에 Git GUI Here(연결 기능) 추가

◆ ☒ Git LFS (Large File Support)

- 대용량 파일 지원

◆ ☒ Associate .git* configuration files with the default text editor

- 기본 텍스트 에디터에 git 구성(.git 확장자) 연결

◆ ☒ Associate .sh files to be run with Bash

- Bash에 .sh 확장자 파일 연결

◆ Check daily for git for Windows updates

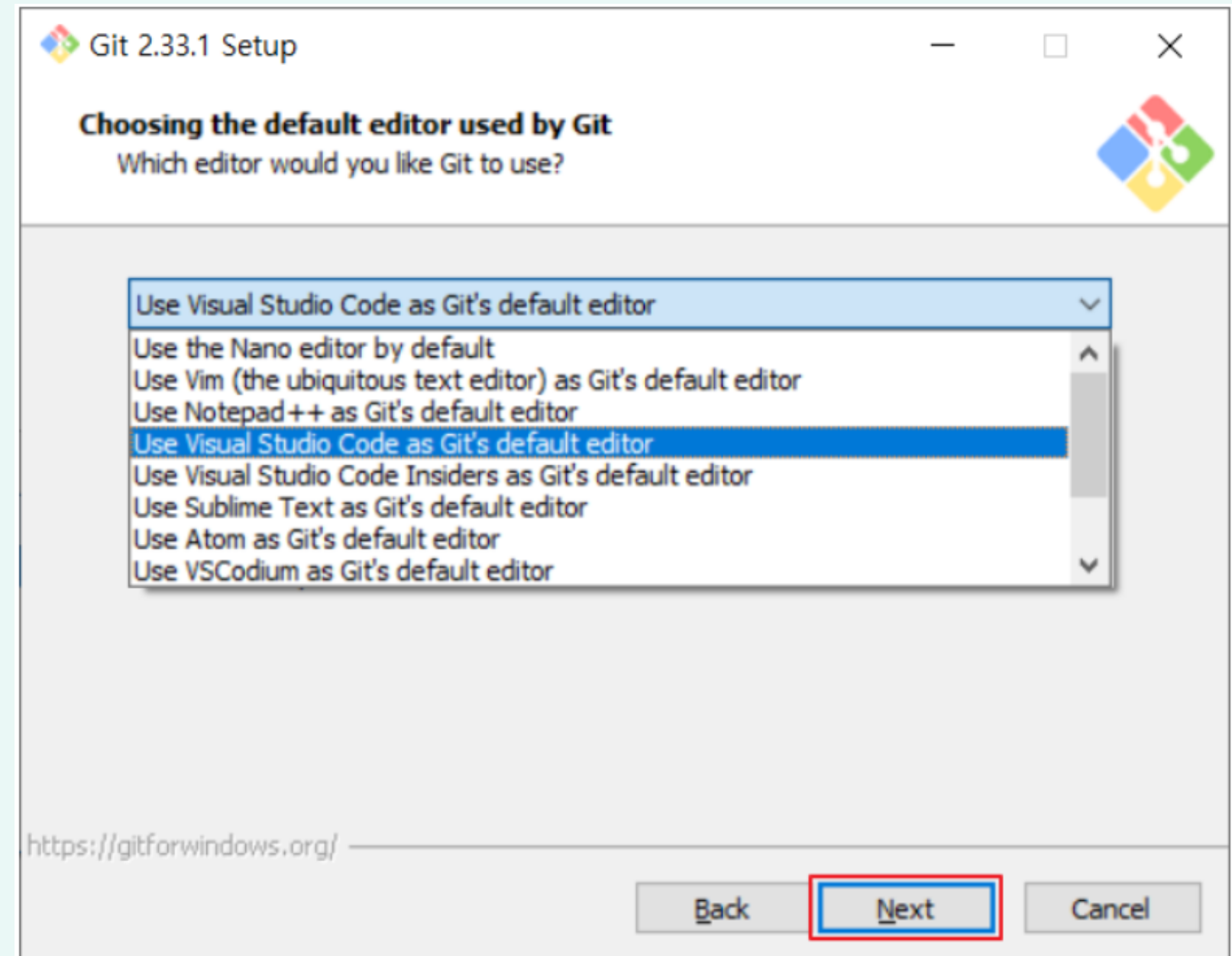
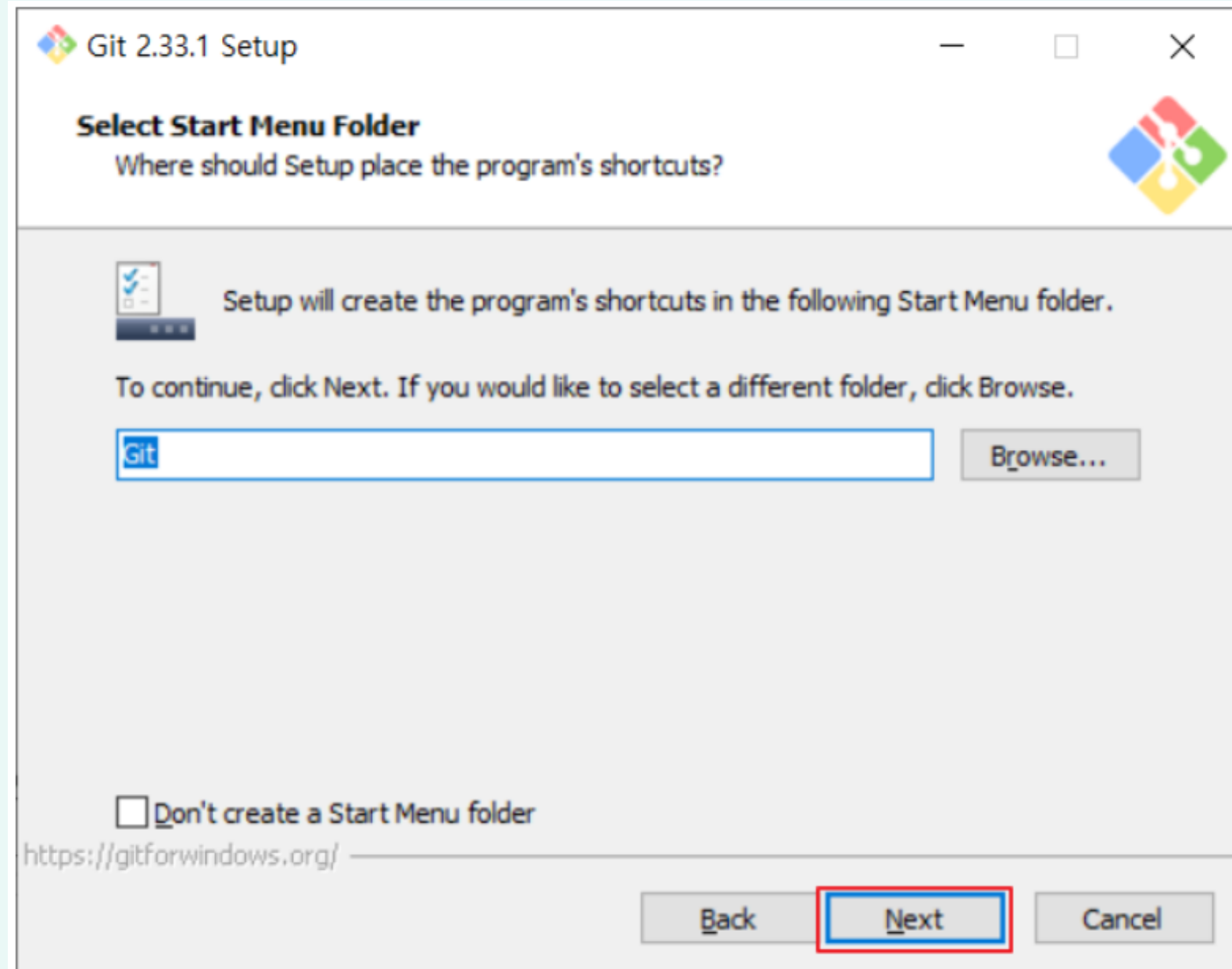
- 매일 새로운 업데이트 확인

◆ (NEW!) Add a Git Bash Profile to Windows Terminal

- 윈도우 기본 터미널에 Git Bash 프로파일 추가

02

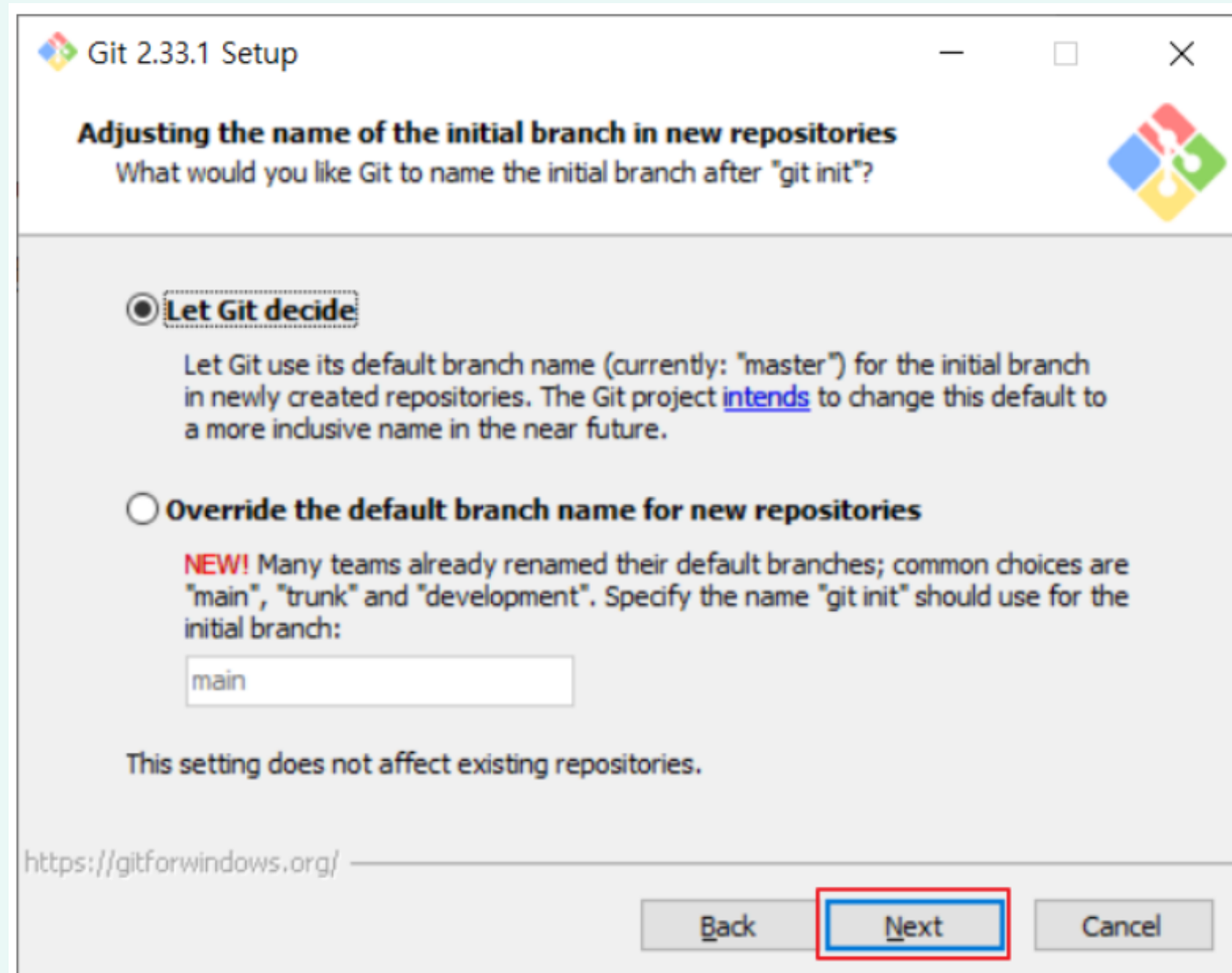
Git 다운로드



✓ Git의 기본 편집기 설정

02

Git 다운로드

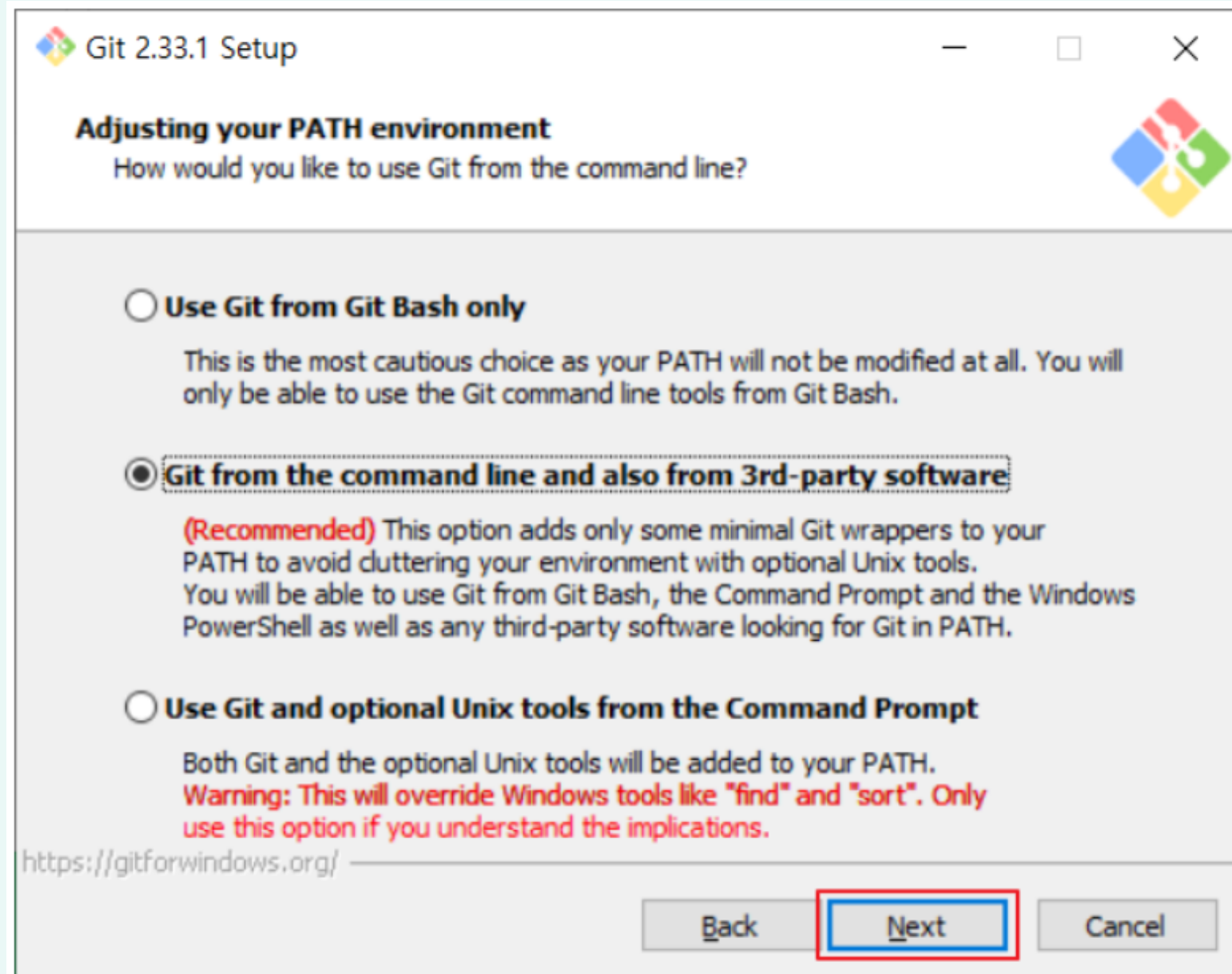


- ✦ ☒ **Let Git decide** : 새로운 Repository의 initial branch에 기본 분기 이름("master")을 사용합니다.
- ✦ **Override the default branch name for new repositories** : 새로운 Repository의 initial branch에 사용자 지정 분기 이름을 사용합니다. 이미 팀에서 관용적으로 쓰는 명칭이 있을 때 사용하면 됩니다.

✓ 새 Repository를 생성할 때, initial branch 이름을 지정하는 방법을 선택

02

Git 다운로드

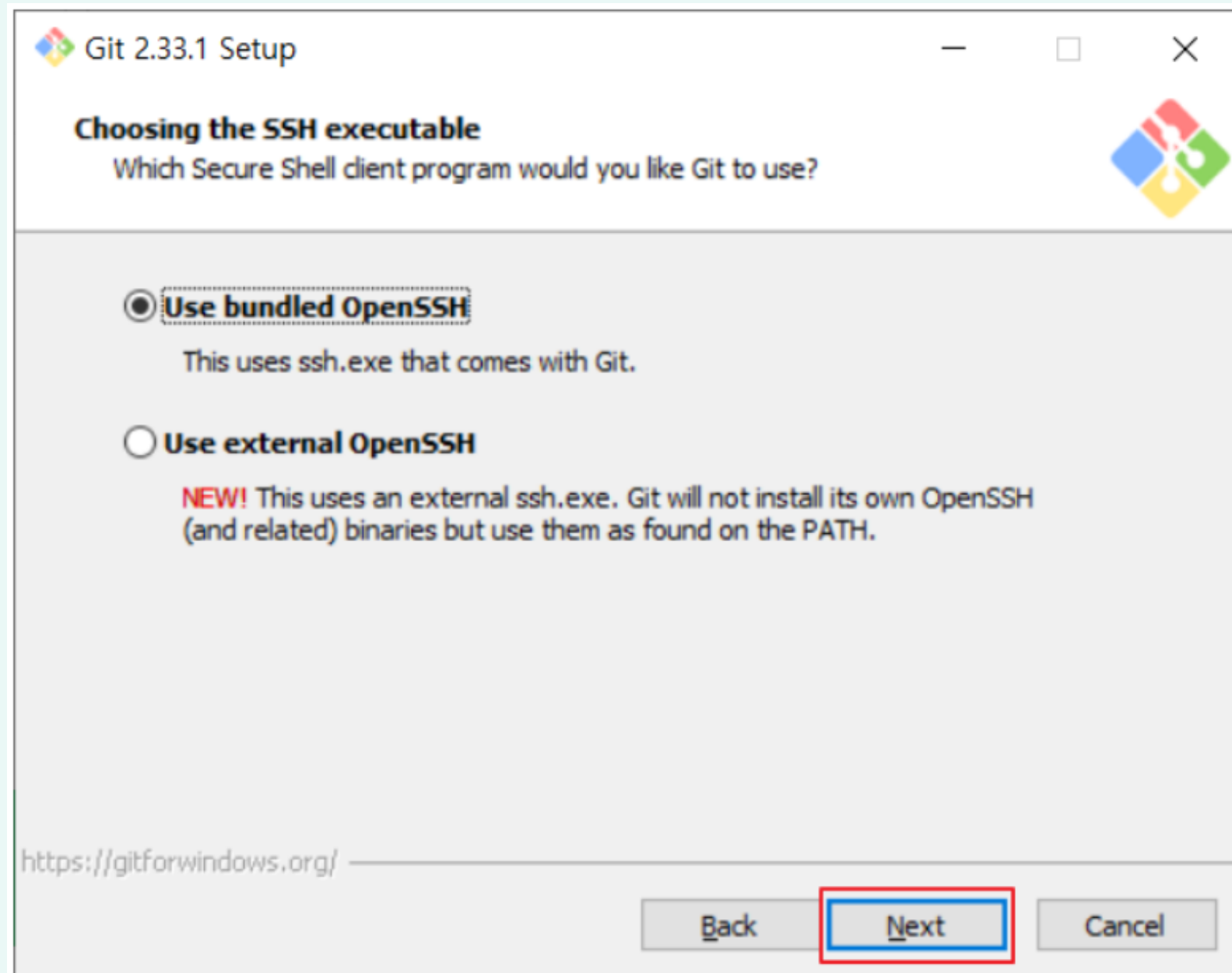


- ✦ Use Git from Git Bash only : Git Bash에서만 Git 명령어를 수행할 수 있습니다.
- ✦ ☒ Git from the command line and also from 3rd-party software : Git을 환경변수 (PATH)에 추가하여 윈도우 기본 명령 프롬프트(CMD) 등에서도 Git 명령어를 수행할 수 있습니다.
- ✦ Use Git and optional Unix tools from the Command Prompt : Git과 Unix 도구 모두 환경변수(PATH)에 추가합니다. 이 경우 몇 가지 Windows 기본 도구가 새롭게 재정의 됩니다. 이러한 위험을 충분히 숙지하고 있는 경우에만 이 옵션을 사용하기를 권장합니다.

✓ Git커맨드를 사용하기 위한 환경변수를 설정

02

Git 다운로드



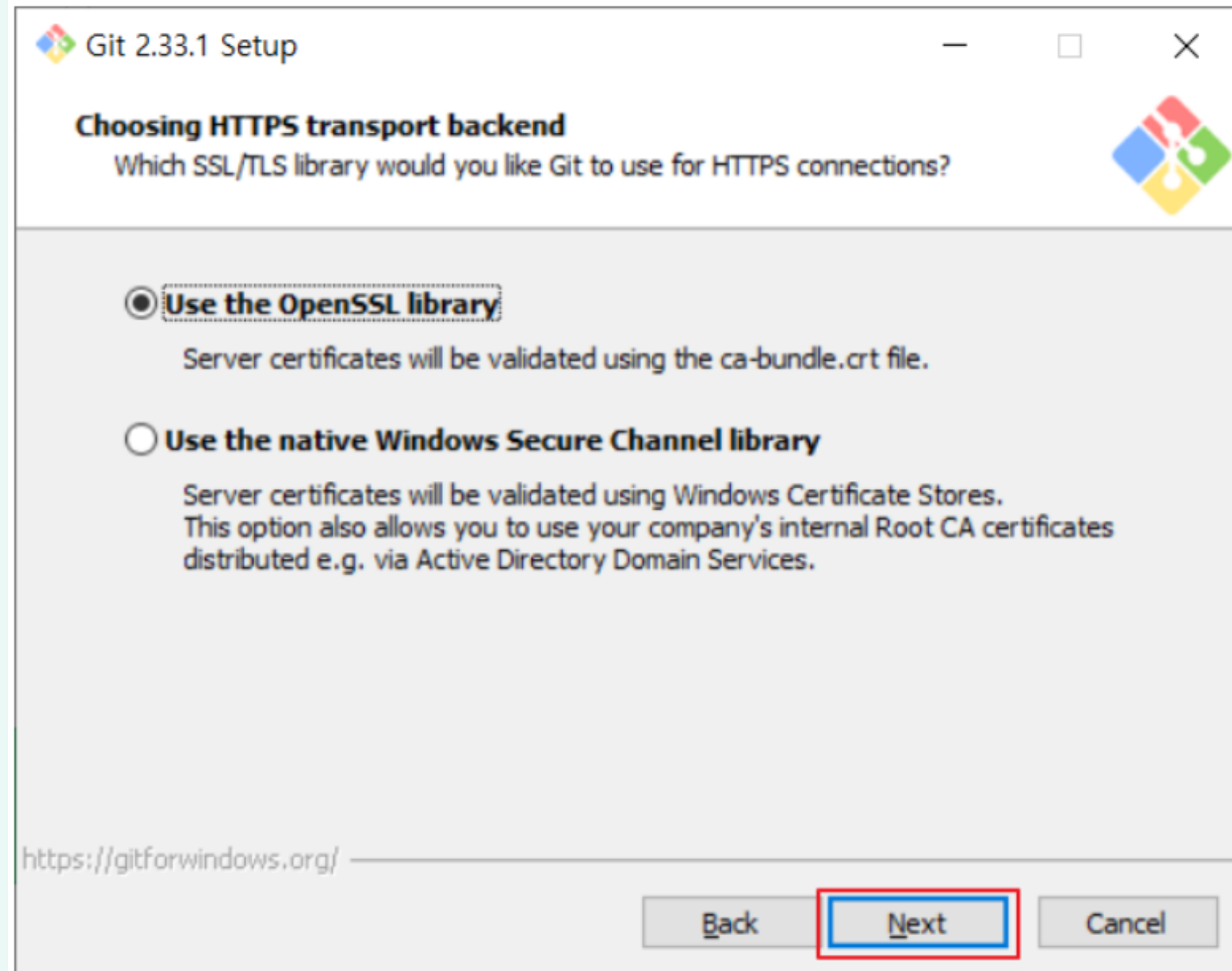
- ✦ ☒ **Use bundled OpenSSH** : Git에서 기본으로 제공되는 OpenSSH를 사용합니다.
- ✦ **Use external OpenSSH** : 외부 OpenSSH를 사용합니다. 이 경우 Git에서 기본 제공하는 OpenSSH는 따로 설치되지 않으며 PATH에 명시된 OpenSSH를 사용합니다.

* SSH : Secure Shell의 줄임말로, 원격 호스트에 접속하기 위해 사용되는 보안 프로토콜

✓ SSH 실행 도구를 선택

02

Git 다운로드

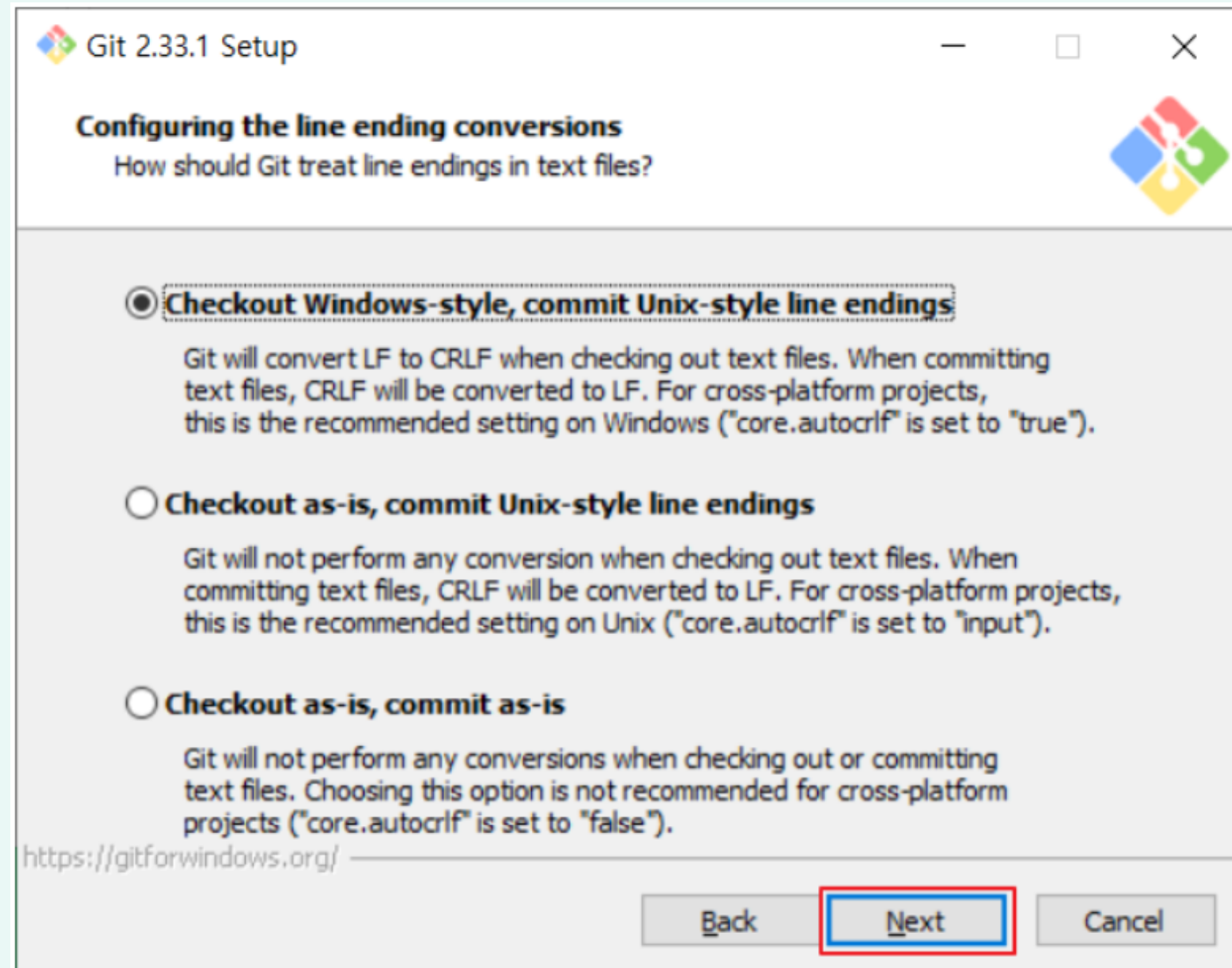


- ✦ ☒ **Use the OpenSSL library** : OpenSSL 라이브러리를 사용합니다. ca-bundle.crt 파일로 검증합니다.
- ✦ **Use the native Windows Secure Channel library** : Windows 인증서 저장소를 사용하여 검증합니다. 이 옵션은 Active Directory 도메인 서비스를 통한 회사의 내부 Root CA 인증서를 사용할 수 있습니다.

✓ HTTP 연결 옵션을 선택

02

Git 다운로드

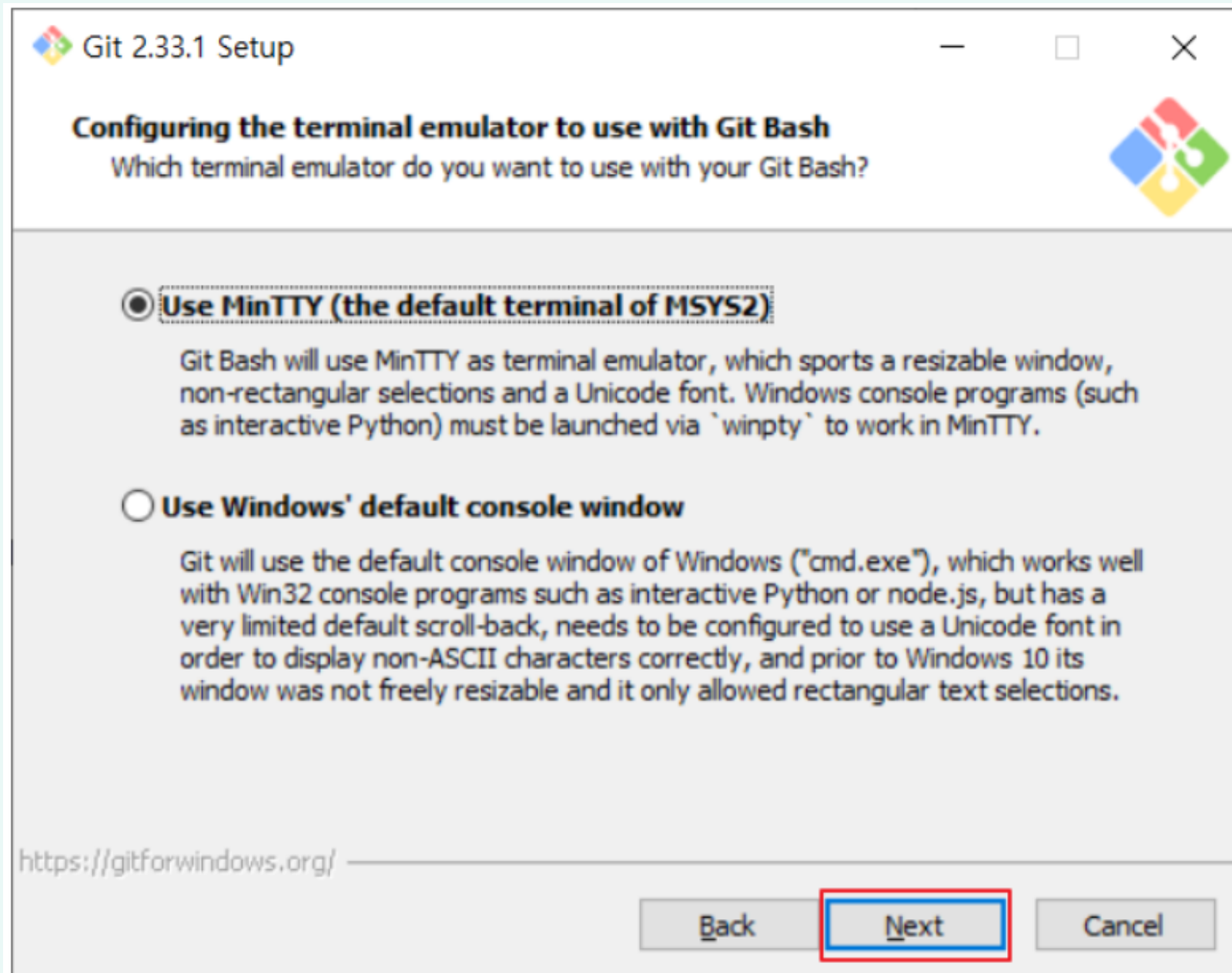


- ✦ ☒ **Checkout Windows-style, commit Unix-style line endings** : 체크아웃은 윈도우 스타일, 커밋은 유닉스 스타일로 자동 변경되도록 설정합니다.
- ✦ **Checkout as-is, commit Unix-style line endings** : 체크아웃은 변경 없이, 커밋은 유닉스 스타일로 설정합니다.
- ✦ **Checkout as-is, commit as-is** : 체크아웃, 커밋 모두 스타일 변경 없이 진행합니다.

✓ Git 저장소에 체크인/아웃할 때의 줄 바꿈 방법을 선택

02

Git 다운로드



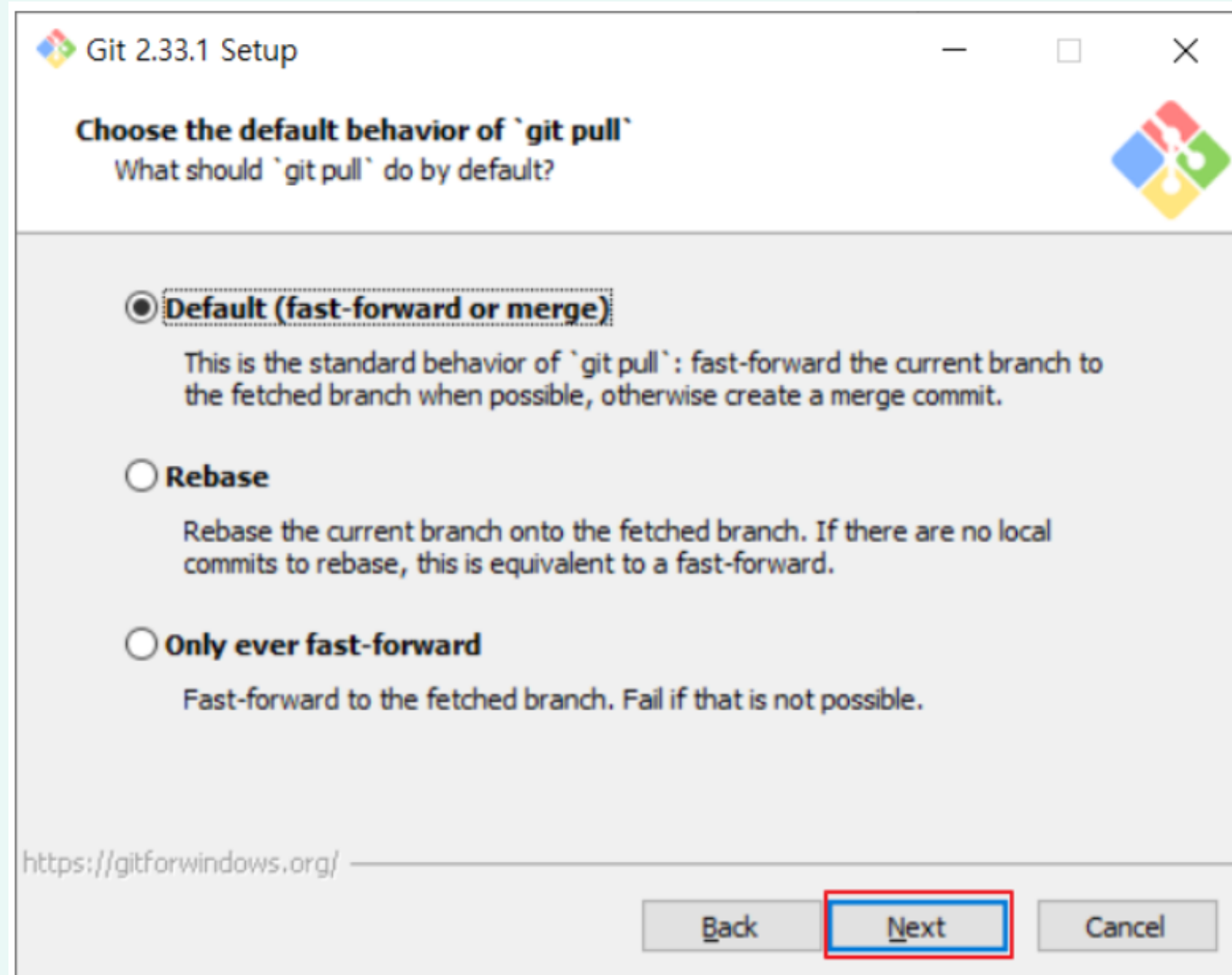
✦ ☒ Use MinTTY (the default terminal of MSYS2) : Git Bash 기본 터미널 에뮬레이터 (MinTTY)를 사용합니다.

✦ Use Windows' default console window : 윈도우 기본 콘솔(cmd)을 사용합니다.

✓ Git Bash 터미널 에뮬레이터를 선택

02

Git 다운로드

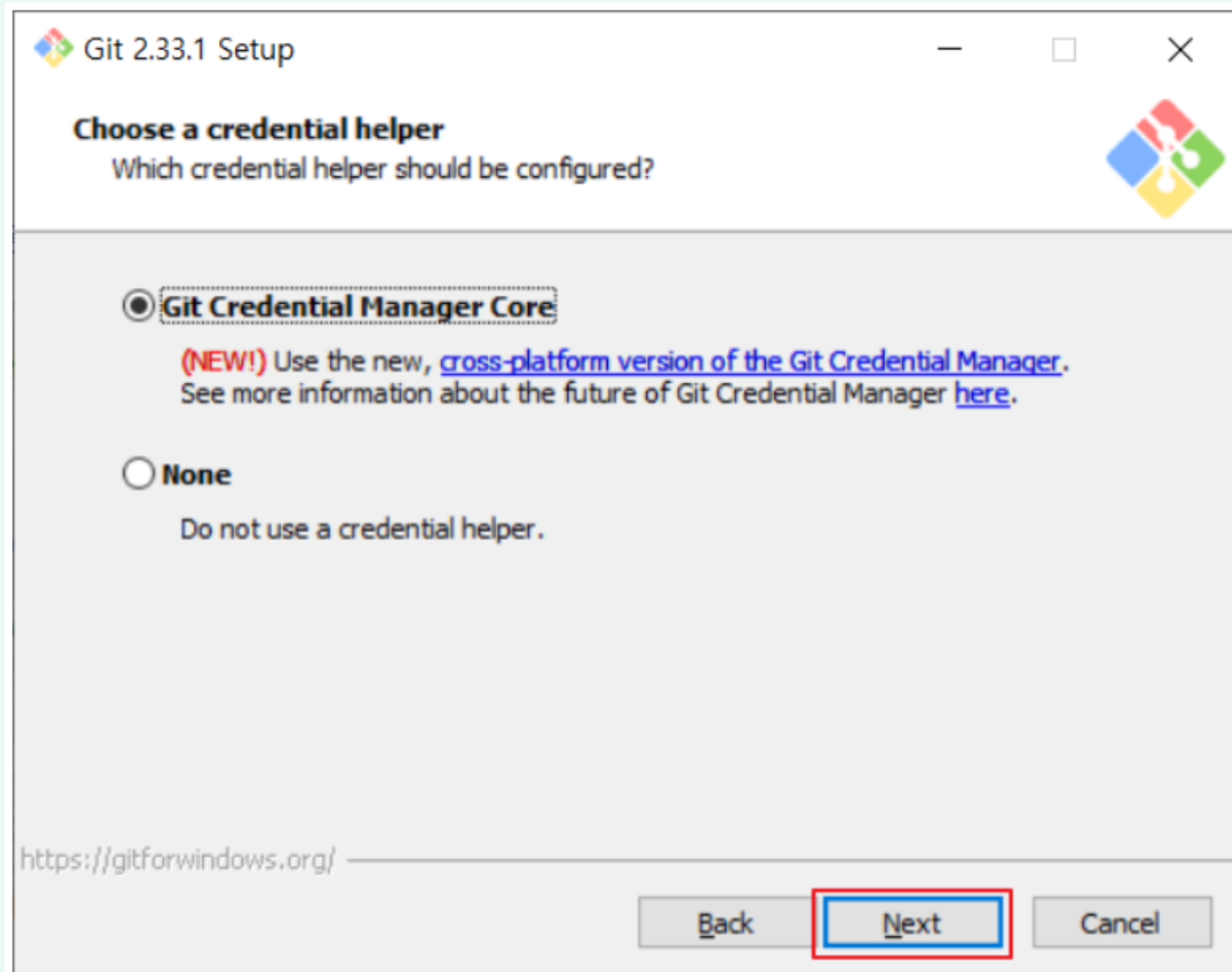


- ✦ ☒ **Default (fast-forward or merge)** : 'git pull'의 수행 동작을 기본으로 설정합니다.
- ✦ **Rebase** : 'git pull'의 수행 동작으로 현재 분기를 불러온 분기에 재배치합니다.
- ✦ **Only ever fast-forward** : 'git pull'의 수행 동작으로 불러온 분기로 빠르게 넘어갑니다. 명령어 수행에 실패할 가능성이 있습니다.

✓ 'git pull' 명령어에 수행될 작업을 선택

02

Git 다운로드

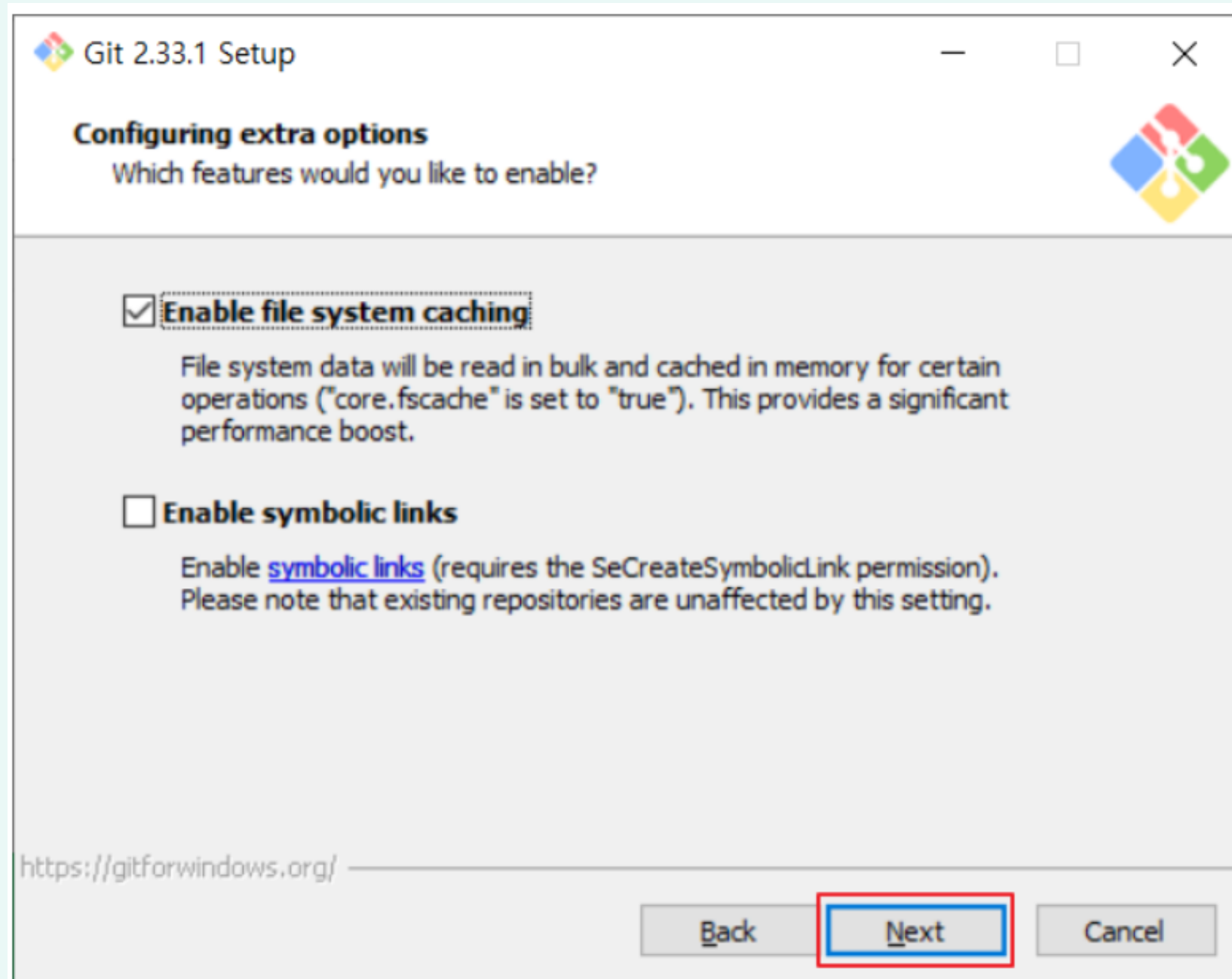


- 📌 ☒ **Git Credential Manager Core** : Git의 자격 증명 도우미를 사용합니다.
- 📌 ☐ **None** : 자격 증명 도우미를 사용하지 않습니다.

✓ 자격 증명 도우미를 선택

02

Git 다운로드



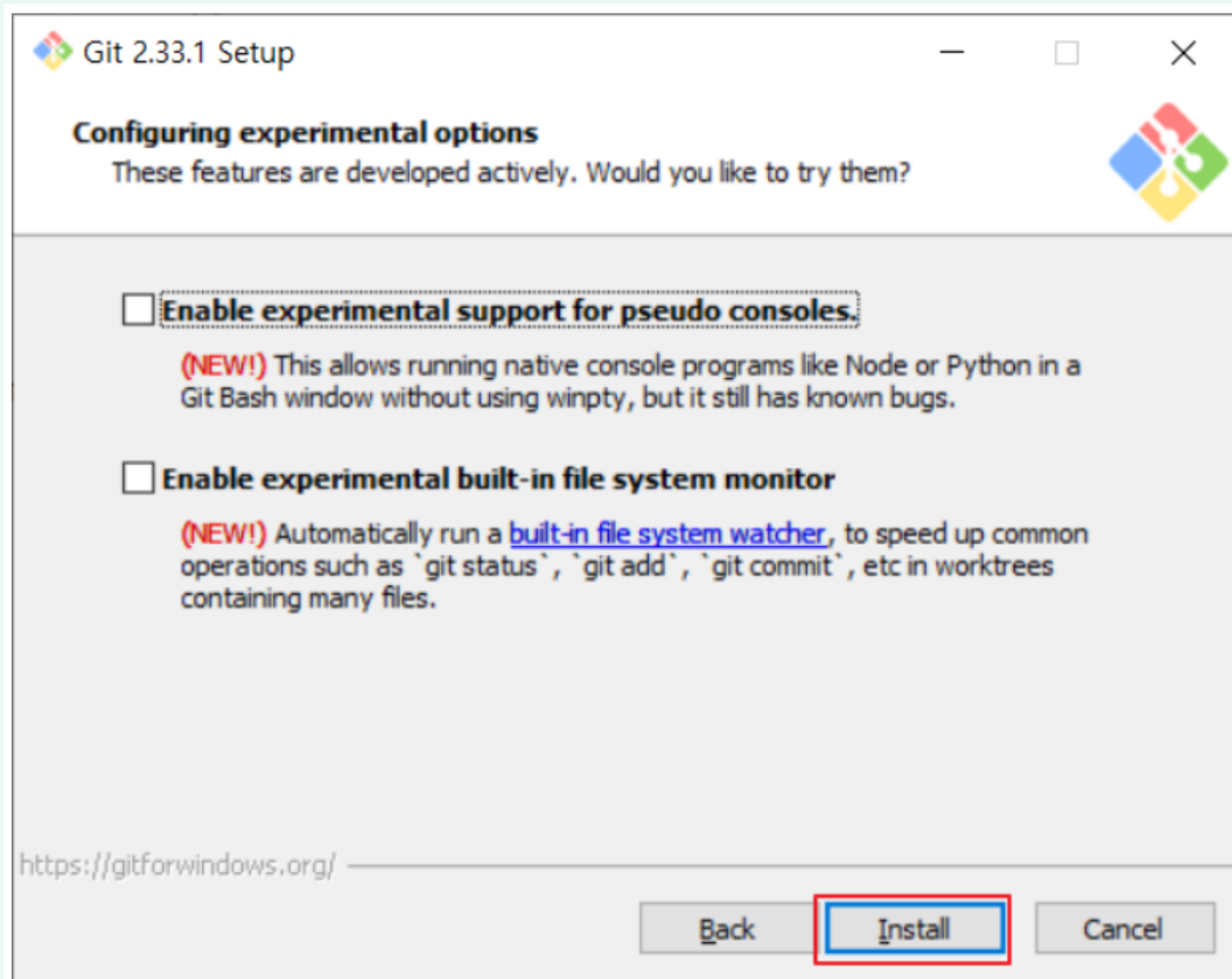
✦ ☒ **Enable file system caching** : 파일 시스템 캐싱을 활성화하여 상당한 성능 향상을 제공합니다.

✦ ☐ **Enable symbolic links** : 심볼릭 링크를 활성화합니다. (SeCreateSymbolicLink 권한이 필요합니다.) 기존 리포지토리에는 이 설정이 적용되지 않습니다.

✓ 기타 옵션을 선택

02

Git 다운로드



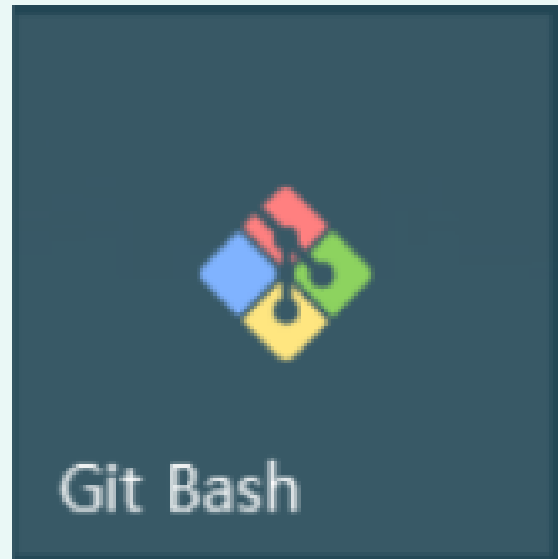
✓ 실험적 기능 사용 여부를 선택

- ✦ **Enable experimental support for pseudo consoles** : winpty를 사용하지 않고 Git Bash에서 Node나 Python과 같은 콘솔을 실행할 수 있게 합니다. 그러나 몇 가지 버그가 있습니다.
- ✦ **Enable experimental built-in file system monitor** : 'git status', 'git commit' 등과 같은 명령어 실행 속도를 높이기 위해 built-in file system monitor를 자동으로 실행합니다.

* 실험적 기능, 개발중인 기능들임. Git에 익숙하지 않다면 선택하지 않는 것을 권장

02

Git 다운로드



1. Git Bash를 실행

```
git config --global user.name "사용자이름"  
git config --global user.email "이메일@abc.com"
```

2. 사용자 이름/이메일 등록을 진행

```
git config --list
```

3. 등록된 사용자를 확인

THANK
YOU

Next : 관련 명령어, 사이트 소개