# Lesson Review
# Version Control with BitBucket

## Setting-up & Managing BitBucket

Benjamin Kenwright*

**Abstract**
A beginners guide to understanding and getting started with version control using BitBucket. This article gives a brief introduction to what version control is, why we use it, and how to setup and configure a private working repository using the BitBucket and SourceTree tools.

**Keywords**
Version Control, BitBucket, Git, Client Manager, Bug Tracking, Alerts, SourceTree

* *Edinburgh Napier University, School of Computer Science, United Kingdom*: b.kenwright@napier.ac.uk

## Contents



## Introduction

**Getting started** Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. Even though the examples in this article show software source code as the files under version control, in reality any type of file on a computer can be placed under version control (e.g., bmp, doc, and avi).

**Why do we need version control?** If you are a software developer and want to keep a safe and secure record of every files version (which you certainly would), it is very wise to use a Version Control System (VCS). A VCS allows you to: revert files back to a previous state, revert the entire project back to a previous state, review changes made over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more. Using a VCS also means that if you screw things up or lose files, you can generally recover easily. In addition, you get all this for very little overhead.

- Industry requirement (i.e., you're expected to know)
- Secure safe storage (e.g., compared to USB)
- Cross platform (e.g., Windows/Mac)
- Track code modifications
- Alerts (e.g., emails/notifications when changes happen)
- Access anywhere
- Issue/bug tracker (e.g., within bitbucket)
- Branch/split builds (e.g., release/debug/features)
- Multiple users
- ...

**Industry requirement** Version control systems are essential for any form of distributed, collaborative development. Whether it is the history of a wiki page or large software development project, the ability to track each change as it was made, and to reverse changes when necessary can make all the difference between a well managed and controlled process and an uncontrolled 'first come, first served' system. It can also serve as a mechanism for due diligence for software projects.

**Flavours** There is a wide variety of version control systems, for example:

- Bitbucket [https://bitbucket.org]
- Github [https://github.com/]
- SourceForge [http://sourceforge.net/]
- Google Code [http://code.google.com/]
- Git (open source) [http://git-scm.com/]
- CVS (open source) [http://www.nongnu.org/cvs/]
- Arch (open source) [http://www.gnu.org/software/gnu-arch/]
- Subversion (open source) [http://subversion.apache.org/]
- BitKeeper [http://www.bitkeeper.com/]
- ...

**Why bitbucket?** As with the Lilliput and Blefuscu from Gulliver travels - no one version control system is best - each version control system has advantages and disadvantages. However, we focus on bitbucket, as it provide unlimited free private repositories with extra features for universities while

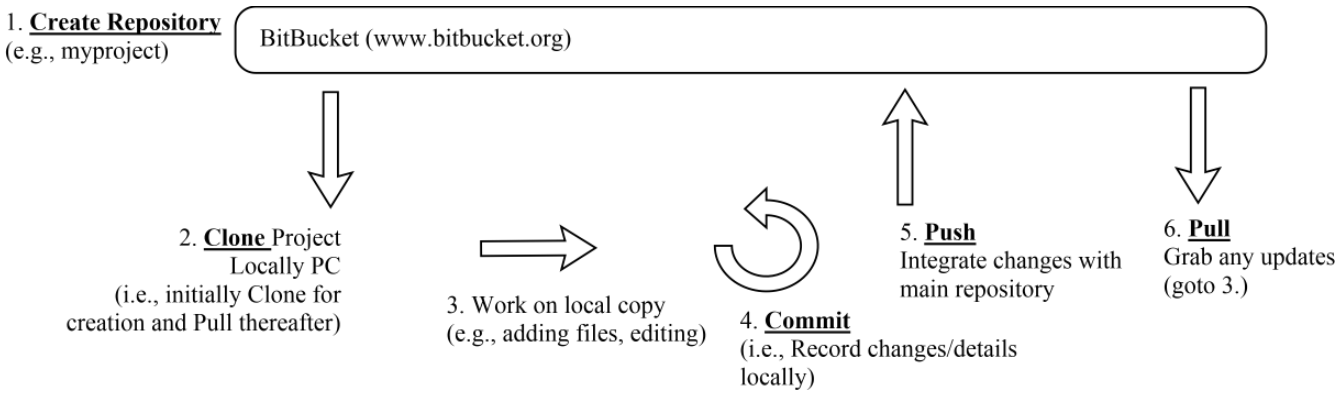


**Figure 1. Cycle Overview** - Key step-by-step flow from repository creation, cloning, editing, adding, committing, pushing, and pulling.

giving a real-world industrial working environment. Finally, bitbucket operates using a distributed version control system, hence, should bitbucket go down, your repository is still accessible. (e.g., if they are experience some serious downtime, you'll need to find a quick alternative web presence for your project - and Bitbucket is a perfect choice).

## 1. Overview

This version control article is divided into specific parts. The practical perspective is intended to give the student a running start with ongoing experience working with a version control environment. In addition, we build the foundation for students who wish to go on and use version control for a variety other collaborative games/applications/project.
The different sections include:

- Motivation for using version control
- Different types of version control
- Setting up a BitBucket account
- Installing and working with a local version control client (i.e., sourcetree)
- Creating and updating repositories
- Practical tips and considerations, such as, adding binaries and comments

## 2. Registering with BitBucket

Registering and creating a bitbucket repository can be accomplished in a few easy steps.

1. login to bitbucket.org
2. select 'sign-up'
3. after signing up you have a valid bitbucket account
4. select 'create' to instance a new repository
5. fill in the fields to give the repository a name and select done (e.g., myproject)
6. you should now have an active bitbucket account and a working repository

## 3. Client

Sourcetree (i.e., http://www.sourcetreeapp.com/) is a free Git & Mercurial client for both Windows and the Mac. The client is installed locally on your machine for pulling and pushing changes to a local working directory.

**Installing and setting up SourceTree** Ensure after installing sourcetree that you go into the settings (i.e., File-¿Setup Wizard) and configure your client so that it can access BitBucket (as shown below in Figure 2).

## 4. Conclusion

**You're ready..** After this short introduction to setting up and using the BitBucket version control system, you should be able to create and manage working repositories. This includes adding files, modifying local and root branch repositories using commit, pull, and push commands.

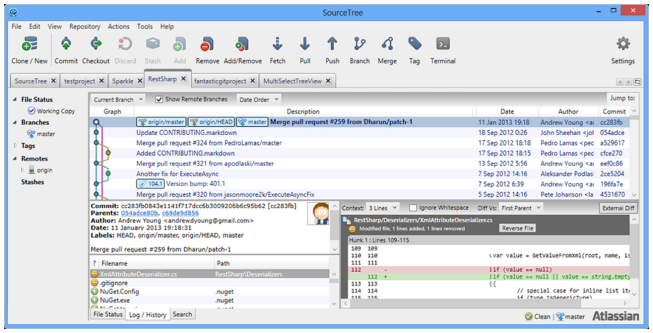

**Figure 2. SourceTree** - After installation you need to configure SourceTree to know about your bitbucket account.

**Remember**

- Setup empty repository
- Add simple file and commit/push it to the repository
- View the repository in the browser (i.e., bitbucket) and verify your files/changes have been saved

- Invite colleagues/lecturers to view (i.e., read only your repository)
- Enable issue tracking in bitbucket for your repository so colleagues/lecturers can give feedback/bugs/tasks

## Acknowledgements

We would like to thank all the readers for taking time out of their busy schedules to provide valuable and constructive feedback to make this article more concise, informative, and correct. However, we would be pleased to hear your views on the following:

- Is the article clear to follow?
- Are the examples and tasks achievable?
- Do you understand the objects?
- Did we missed anything?
- Any surprises?

The article provide a basic introduction to getting started with the BitBucket version control tools. So if you can provide any advice, tips, or hints during from your own exploration of version control development, that you think would be indispensable for a student's learning and understanding, please don't hesitate to contact us so that we can make amendments and incorporate them into future updates.

## Recommended Reading

Version Control with Git: Powerful tools and techniques for collaborative software development, Jon Loeliger and Matthew McCullough, ISBN: 978-1449316389

Git: Version control for everyone, Ravishankar Somasundaram, ISBN: 978-1849517522

Code Complete: A Practical Handbook of Software Construction, Steve McConnell, ISBN: 978-0735619678

Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, ISBN: 978-0132350884

# 1. Appendix

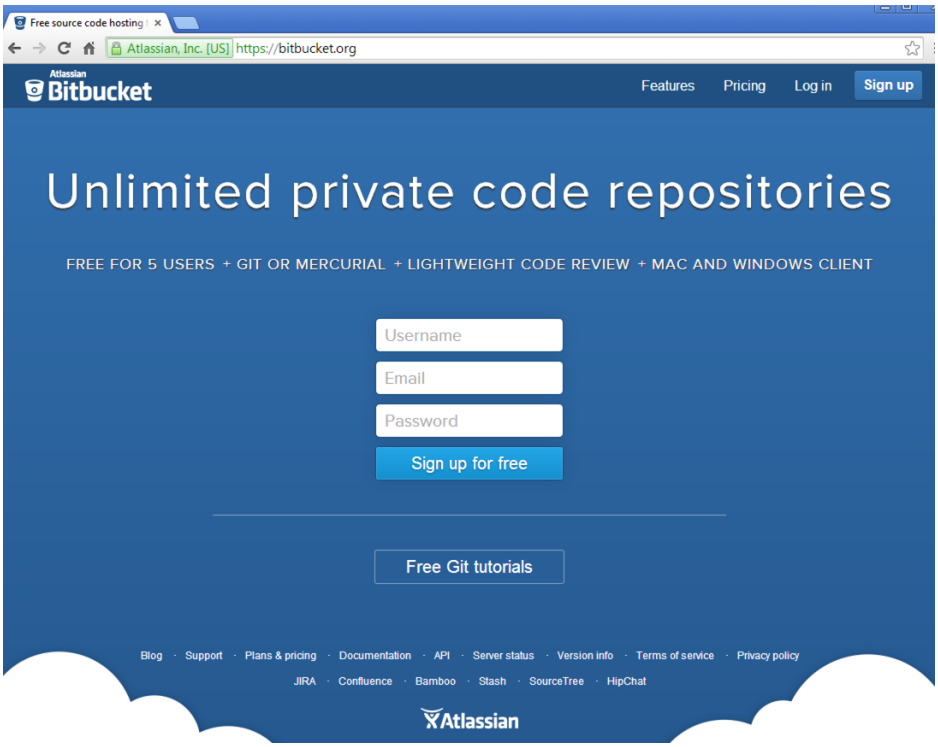


**Figure 3. Getting Started** - www.bitbucket.org

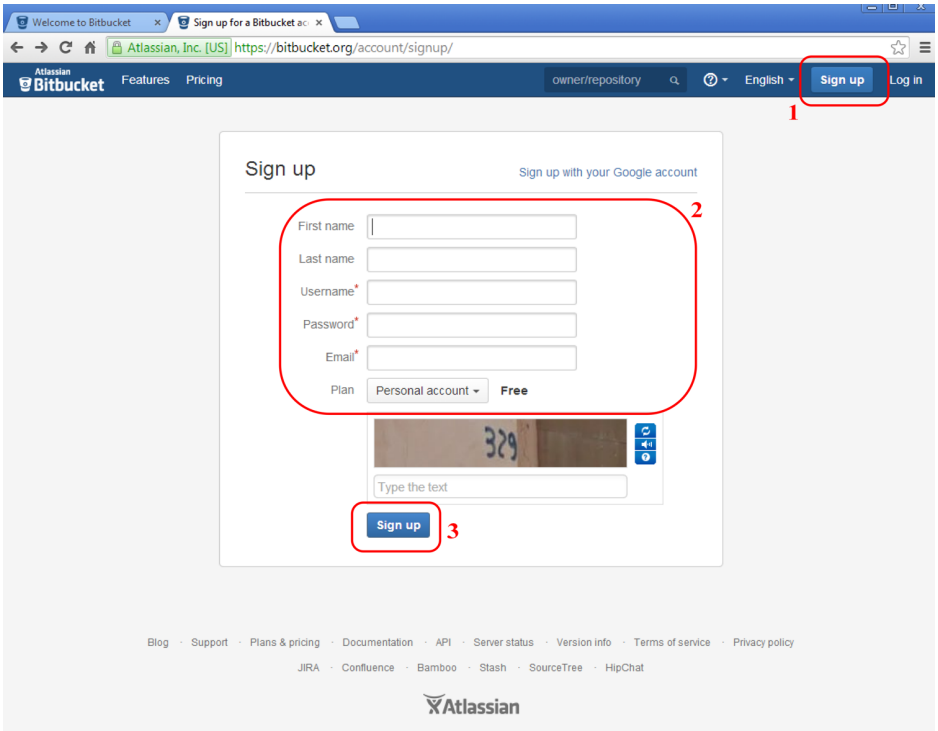


**Figure 4. Registering** - Sign up on the website using your university email address. A university member you are able to create unlimited private repositories.

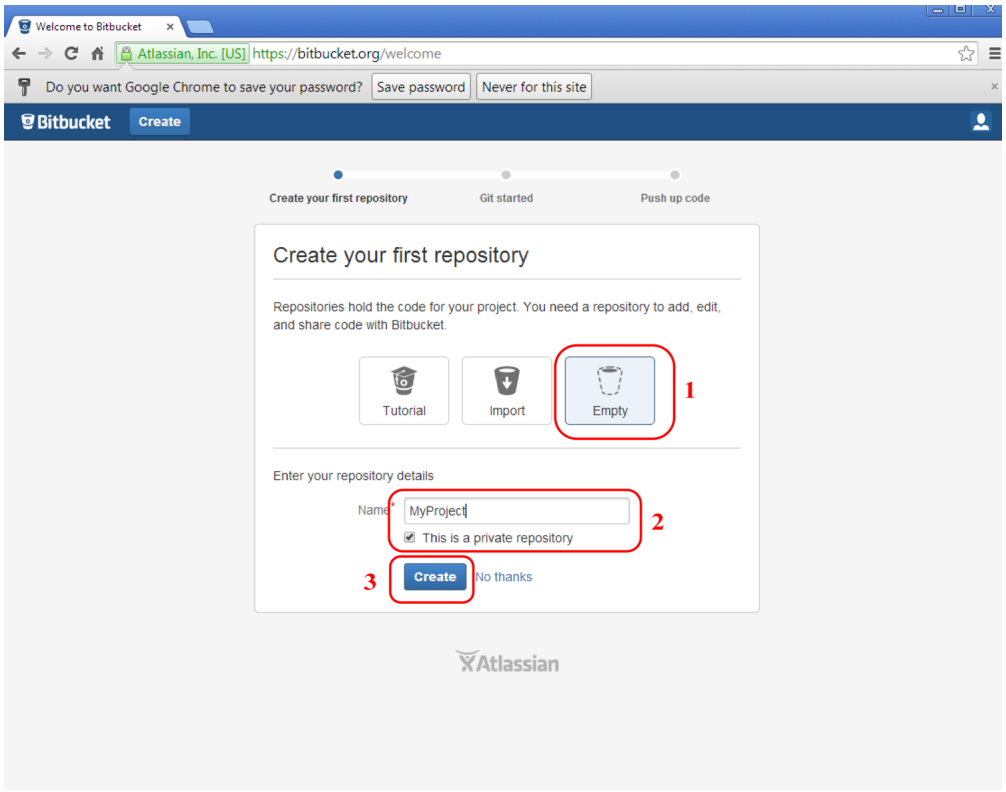


**Figure 5. Creating Repository** - Once you've registered with bitbucket you can create an empty repository for your project.

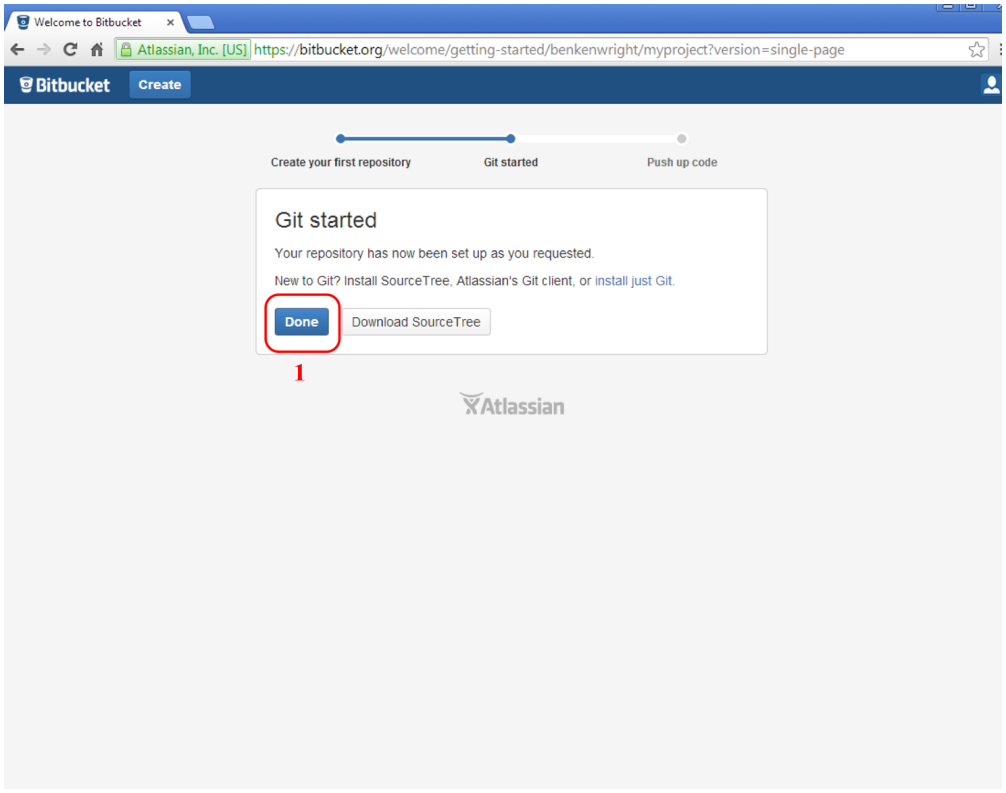


**Figure 6. Complete** - Confirm repository creation.

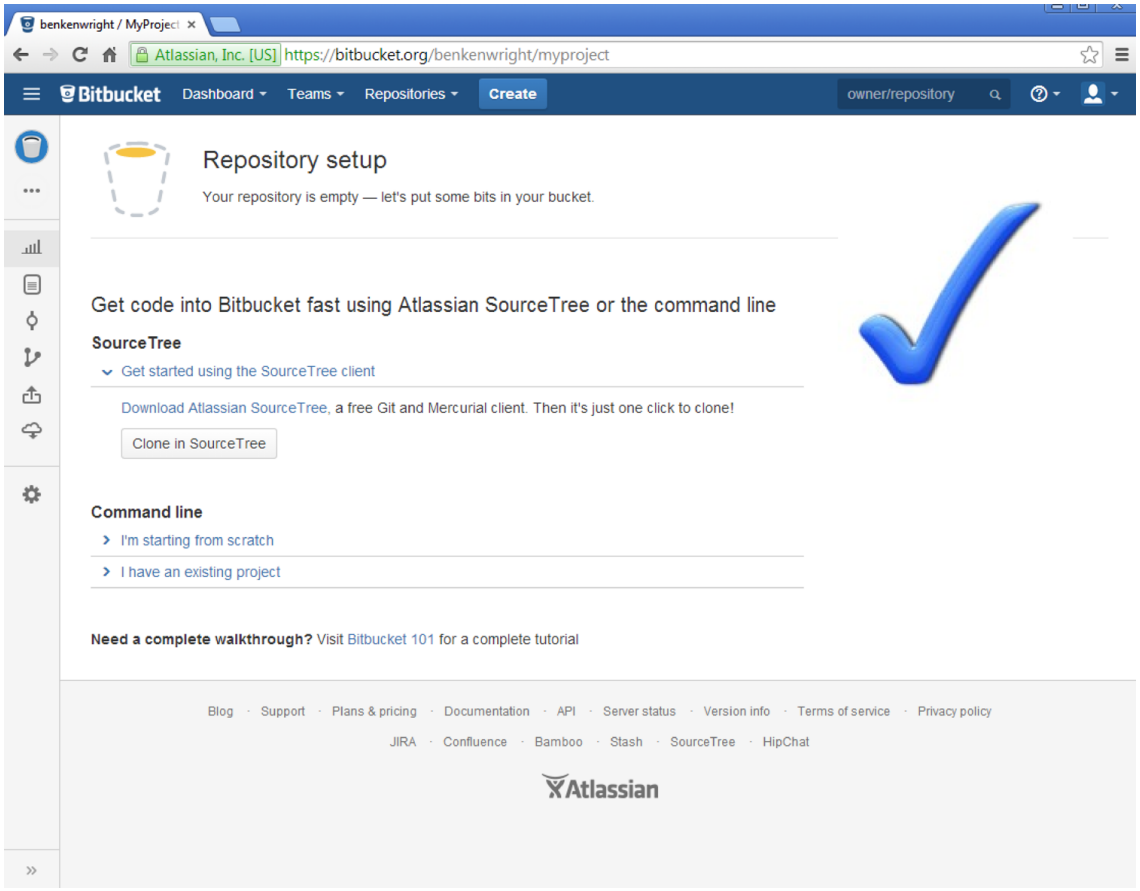


**Figure 7. Bitbucket Setup Complete** - You're registered with bitbucket and you have successfully created an empty repository.

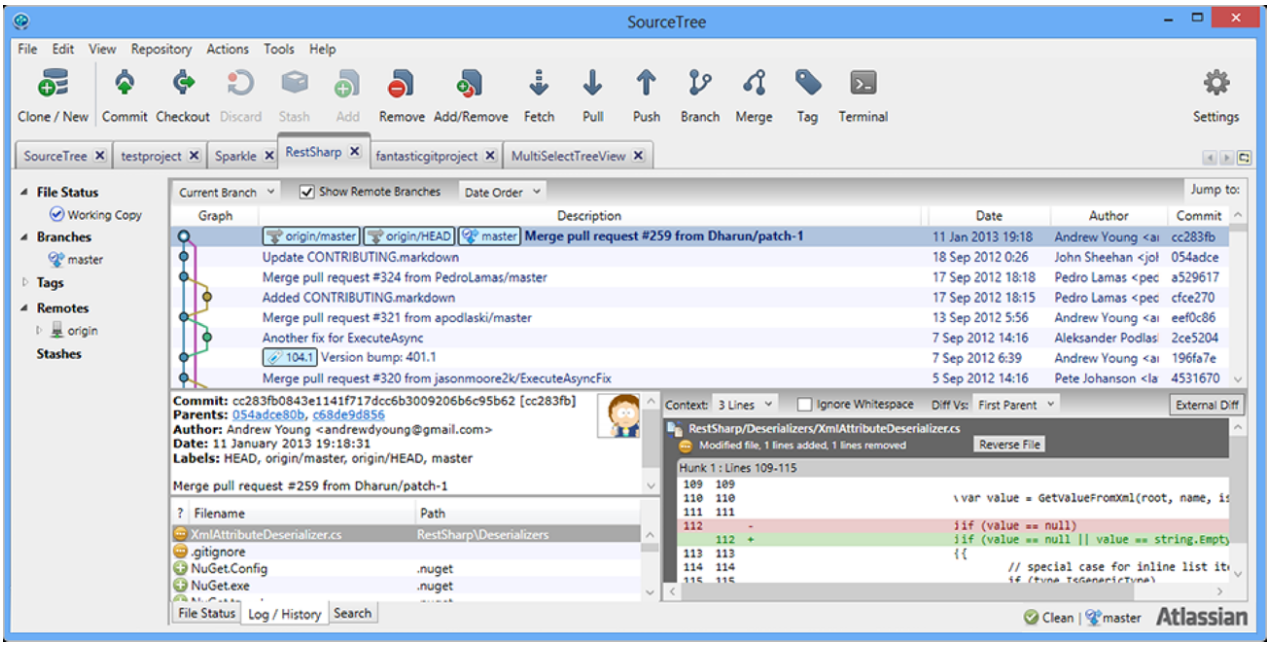


**Figure 8. Installing Local Client (i.e., SourceTree)** - We install a local client on your computer to access the repository on bitbucket.


**Figure 9. Download SourceTree & Install** -

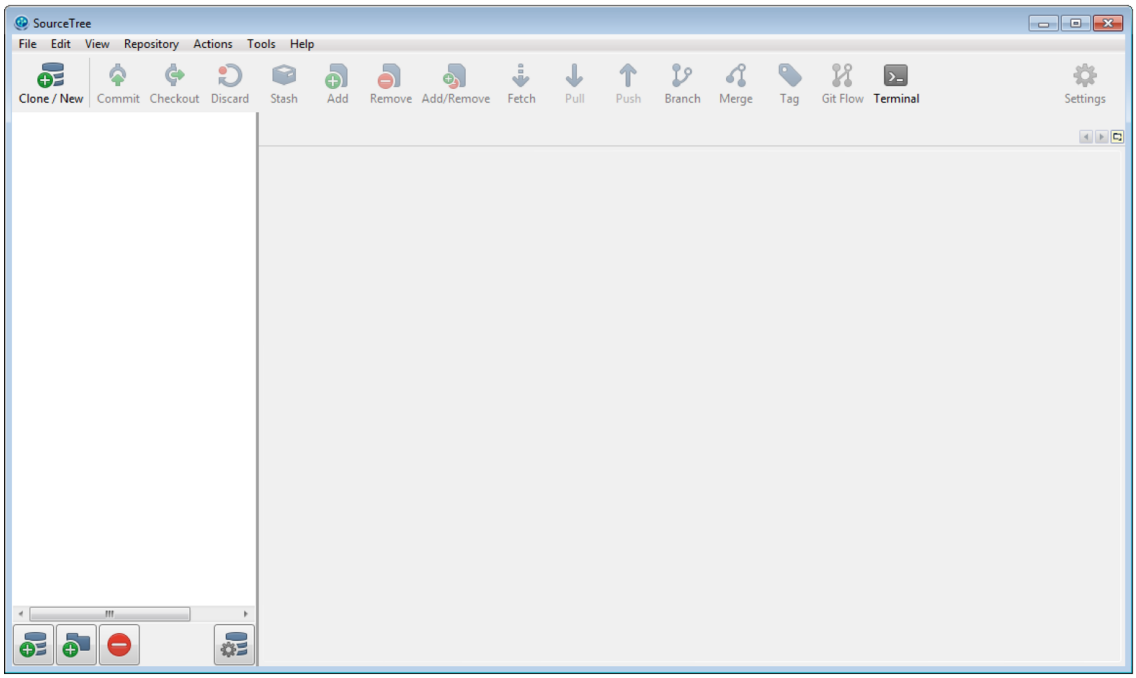


**Figure 10. Running SourceTree** - After installing sourcetree - run it for the first time.


**Figure 11. After Installing - Run File-¿Setup Wizard** - Before you start to access your repository, ensure SourceTree is configured correctly (i.e., to access bitbucket). Select File -¿ Setup Wizard.

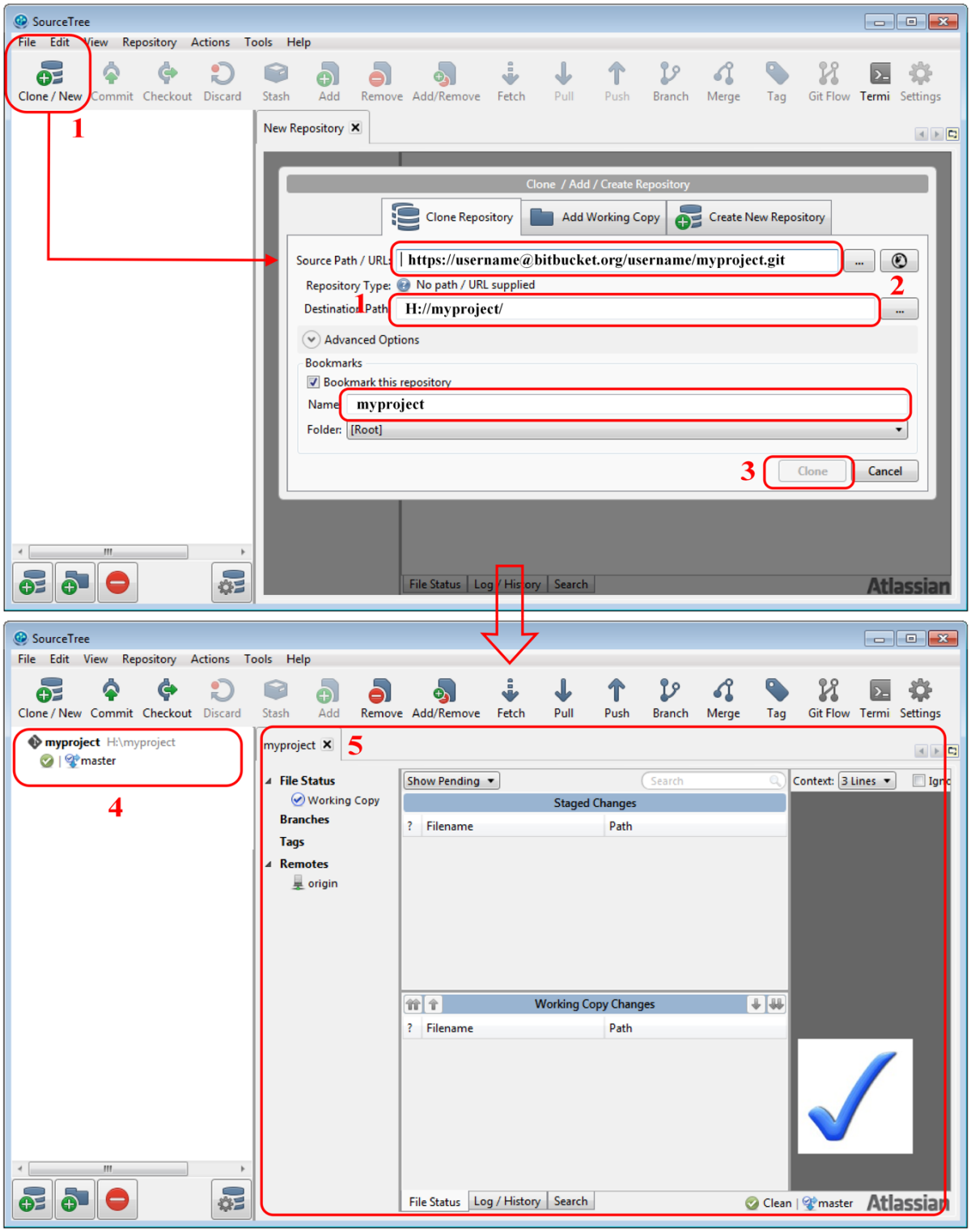


**Figure 12. Cloning Repository Locally** - Setup SourceTree so it connects to your bitbucket repository.
[Source Path / URL]
https://username@bitbucket.org/username/myproject.git
[Destination]
H:/myproject/
[Name]
myproject

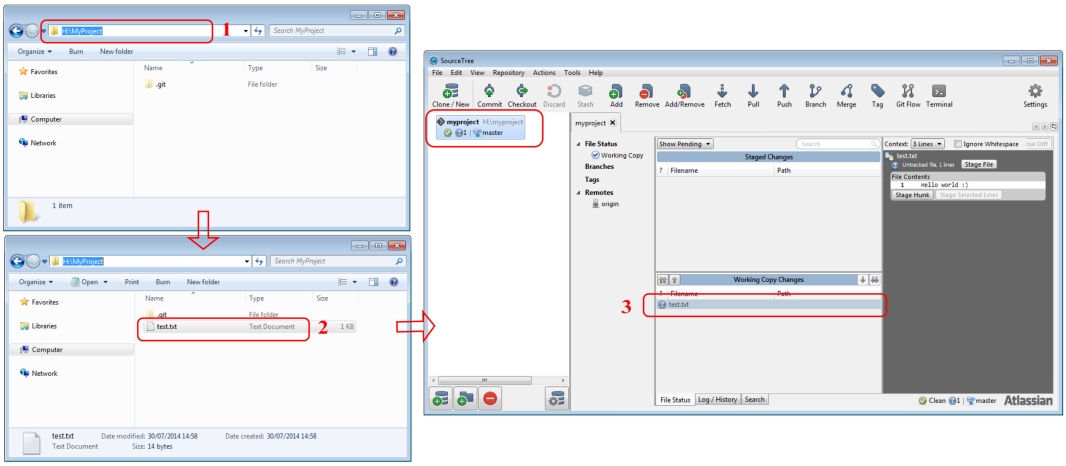


**Figure 13. Start Using Folder** - As you add and modify files within the git folder - SourceTree will automatically detect the changes and report them to you.

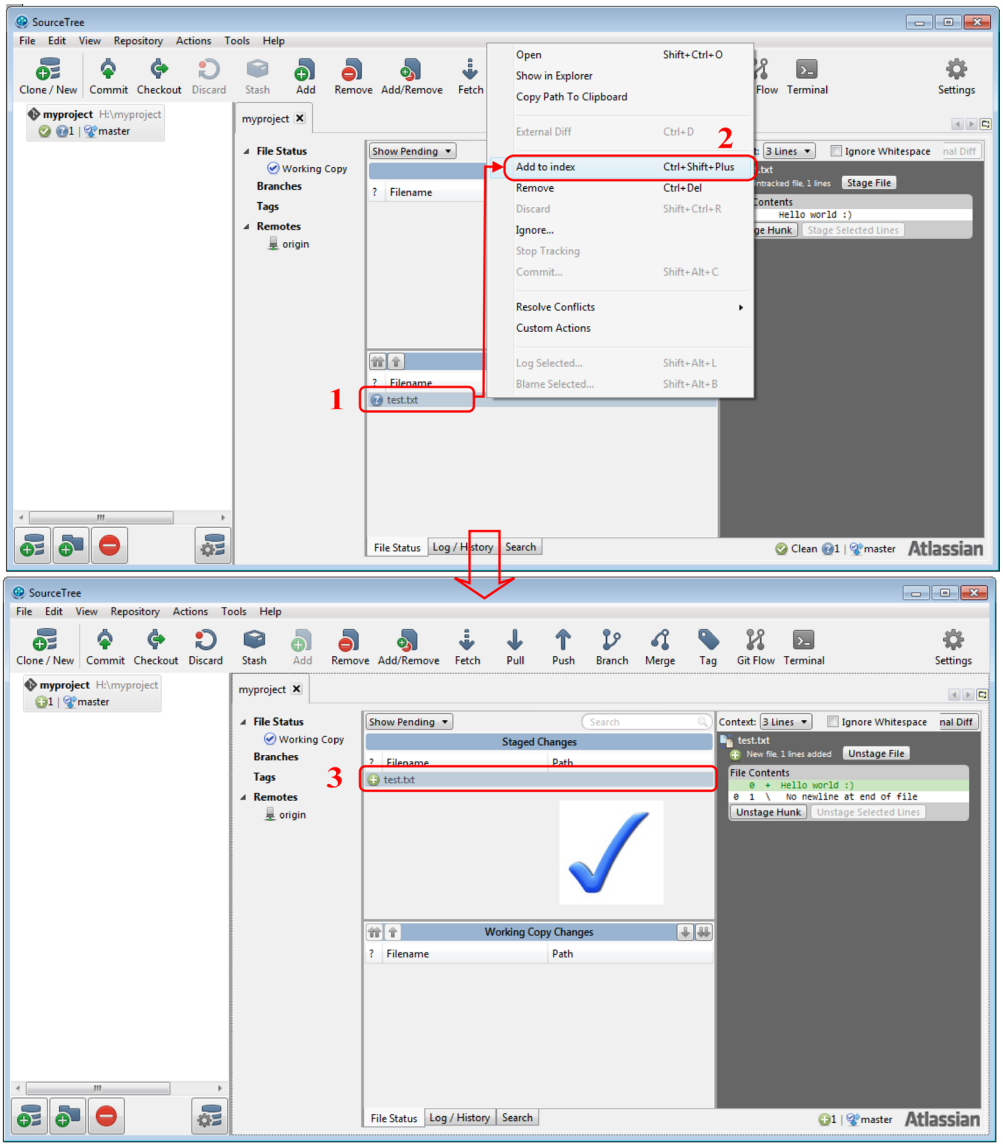


**Figure 14.** -

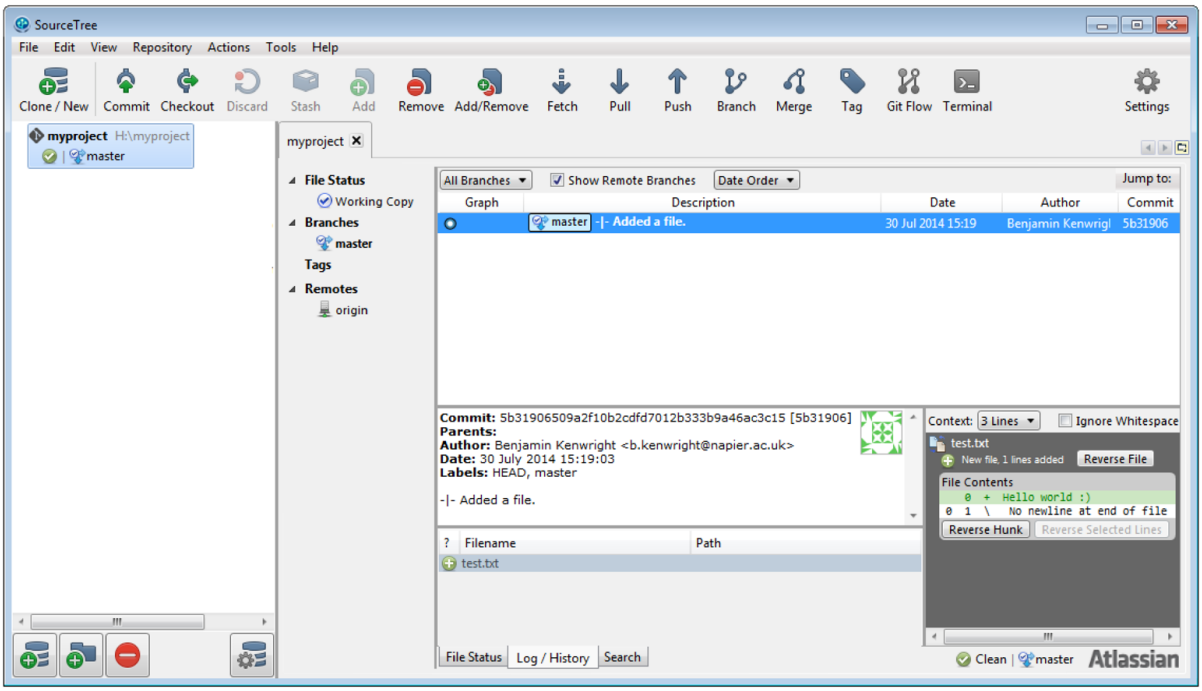


**Figure 15. Adding A File To Repository** - While SourceTree detects changes to the git directory - you have to add files, and commit changes manually - i.e., select which files and add them and push them into the repository.

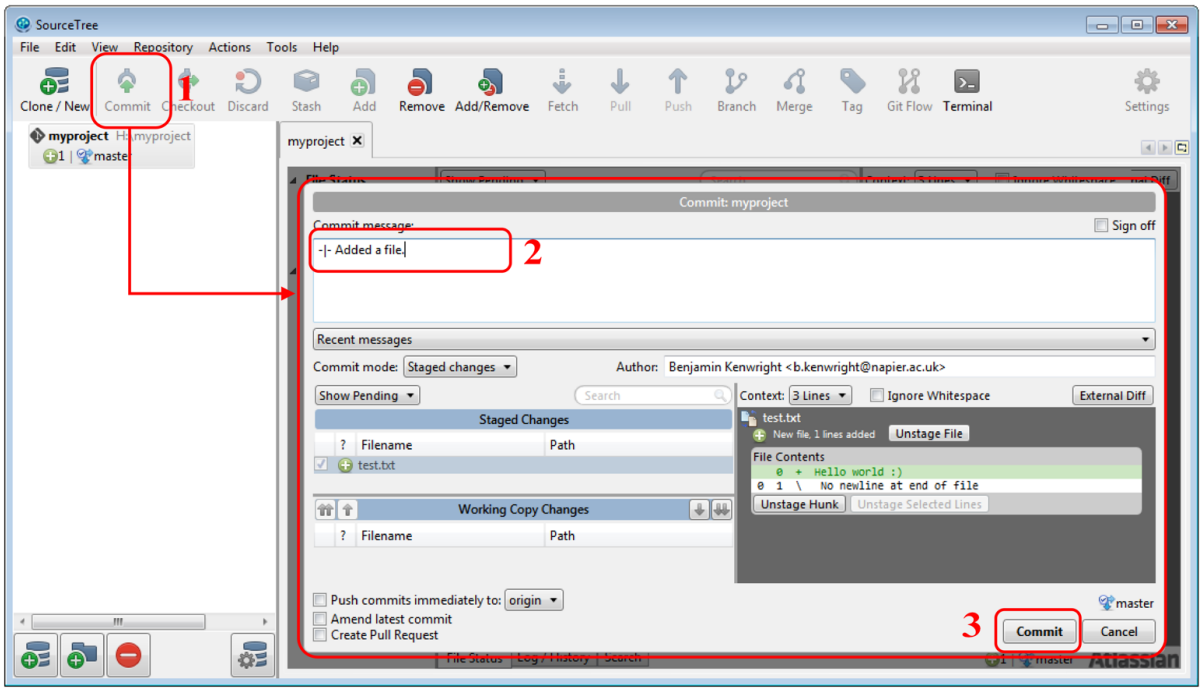


**Figure 16. Commit** - Select 'Commit' and save the repository. Warning - committing only prepares the repository. Pushing the repository transfers the changes to bitbucket.

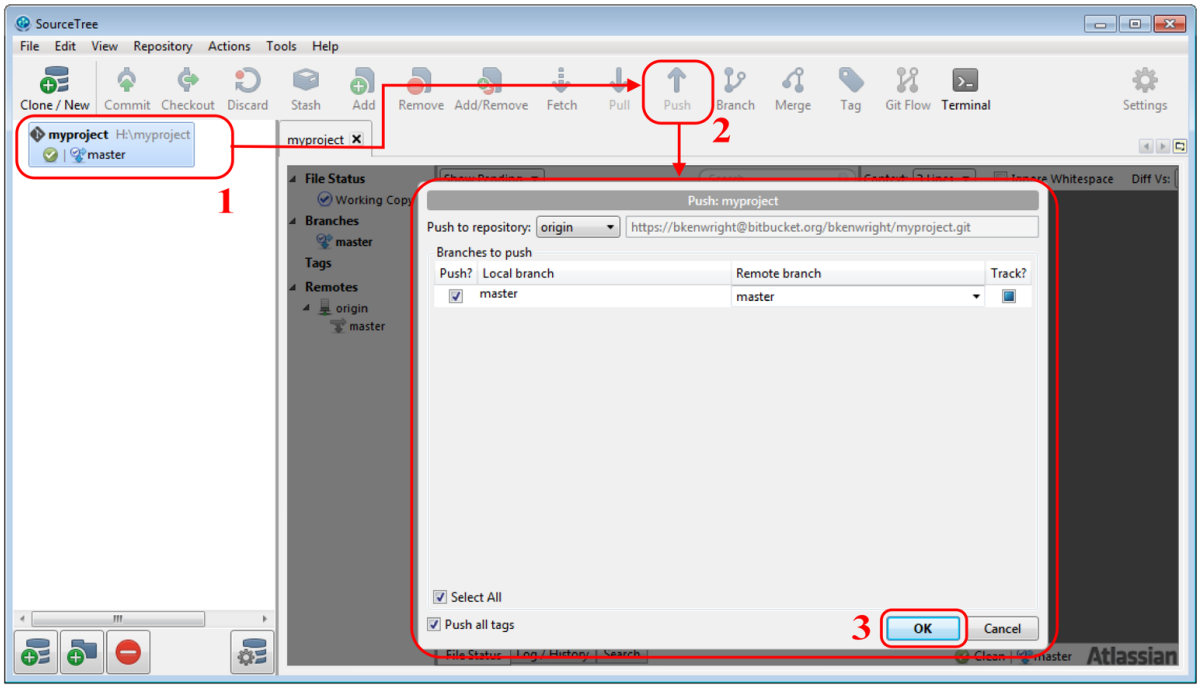


**Figure 17. Finalizing Changes** - Pushing the final changes onto bitbucket.

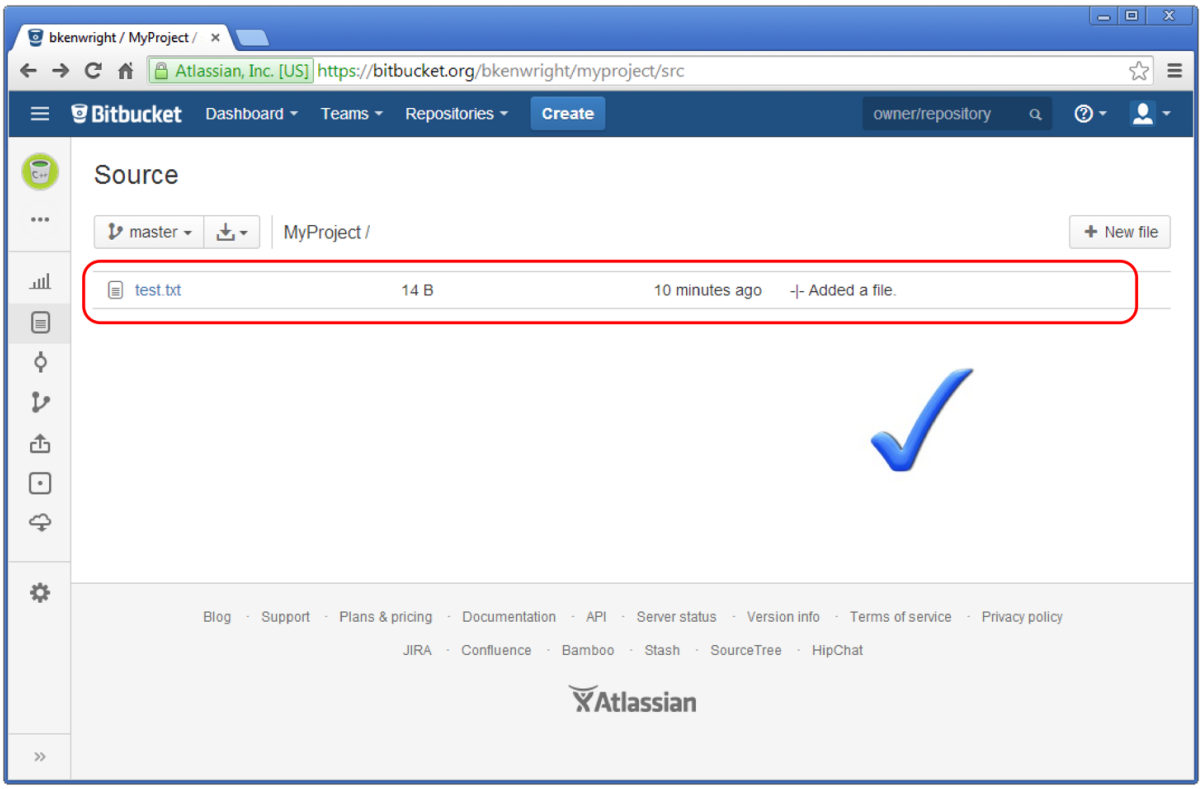


**Figure 18. Did It Work?** - Take a look on the bitbucket website in your repository and you should be able to view the files and the details. Hence, changes have been saved to the main repository.

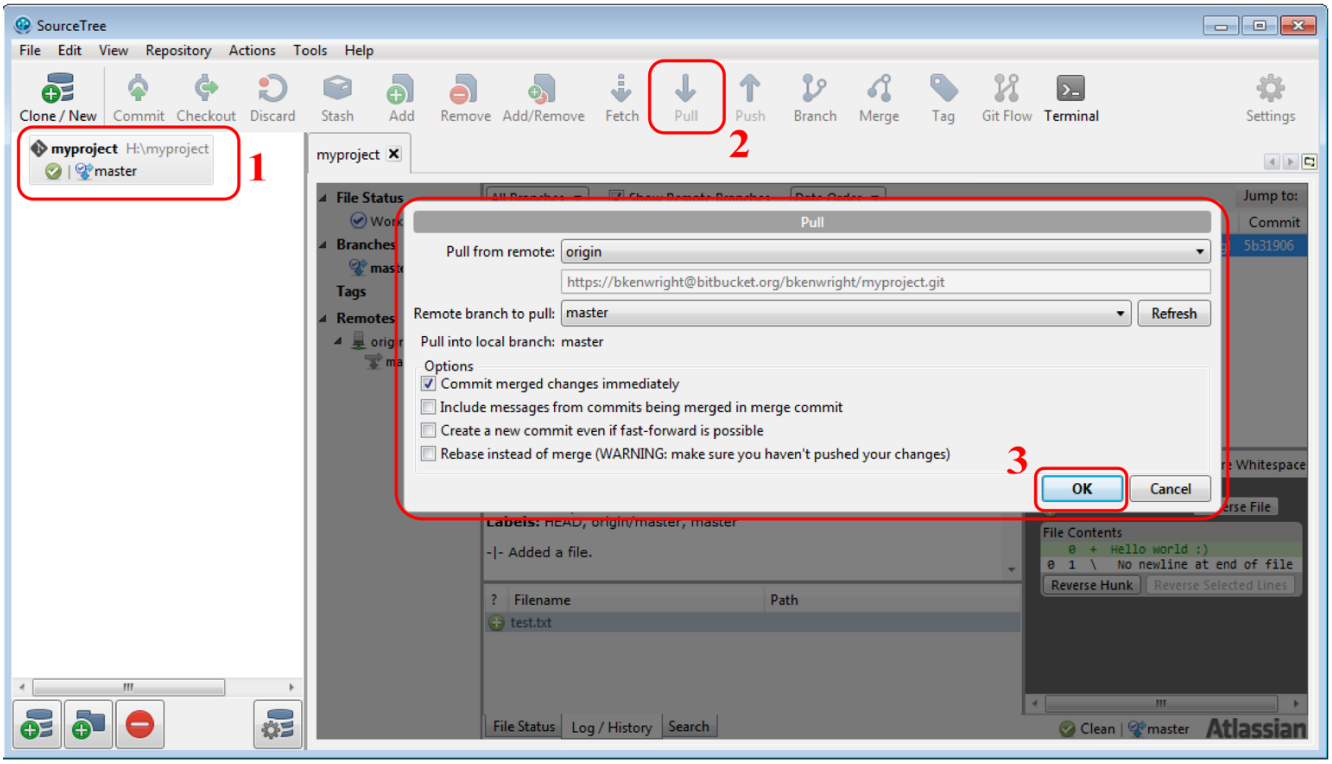


**Figure 19. Pulling** - As you work on different computers and submit your changes - always pull the latest updates to your local repository - do your changes and updates - then commit/push them back into the repository so you never lose your work and are able to track every change.

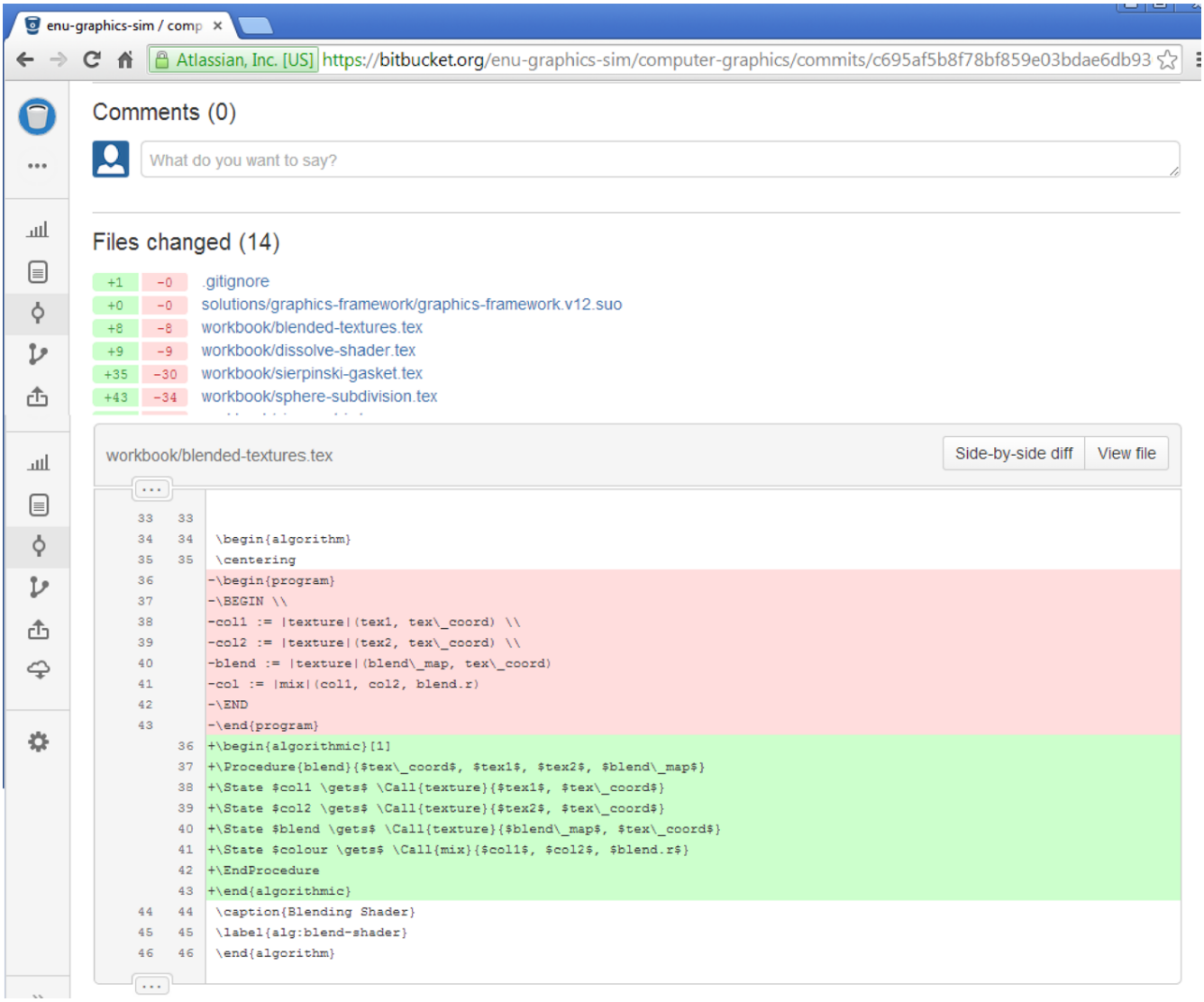


**Figure 20. Tracking Changes** - In summary -
[1]Track changes over time.
[2]Secure & Safe.
[3]Report bugs/progress