



Home

Services

About Us

Contact

Articles

Search

Careers

Sign in

Undo in Git using SourceTree

17 JANUARY 2018 on [Tips & Tricks](#), [Development](#)

If you are not a command-line person, working with Git could be a painful exercise. You can use [SourceTree](#) to make your Git workflow a lot simpler. Recently, I encountered a few folks who were not happy with the [easy & traditional undo](#) options in Git. This article will try to address various cases where you would want to undo and rewrite history. You can simply follow

through, but it would be best if you try it out yourself to get a proper hang of it. Let's get started.

Step 1: Setup a repository for test

Go to [Bitbucket](#) and click on the  icon to create a new repository.

Create a new repository

Import repository

Owner:  itzrahulsoni

Repository name*:

Access level: This is a private repository

Include a README?: No

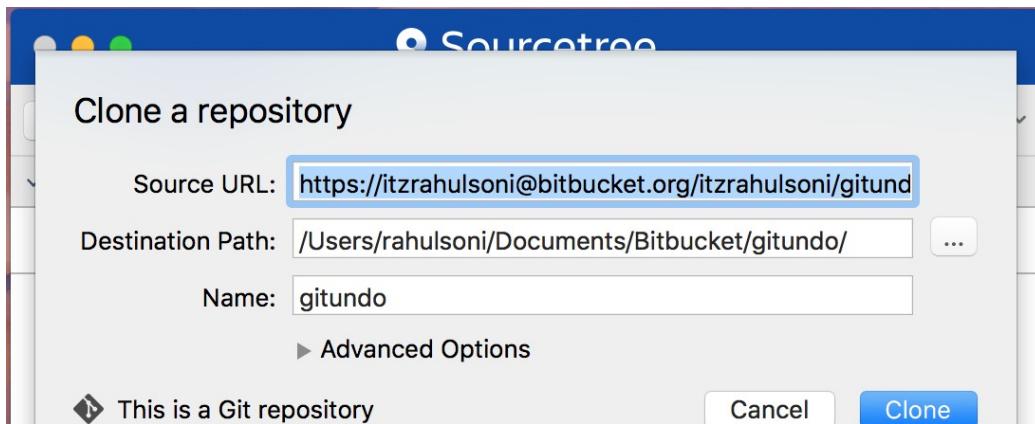
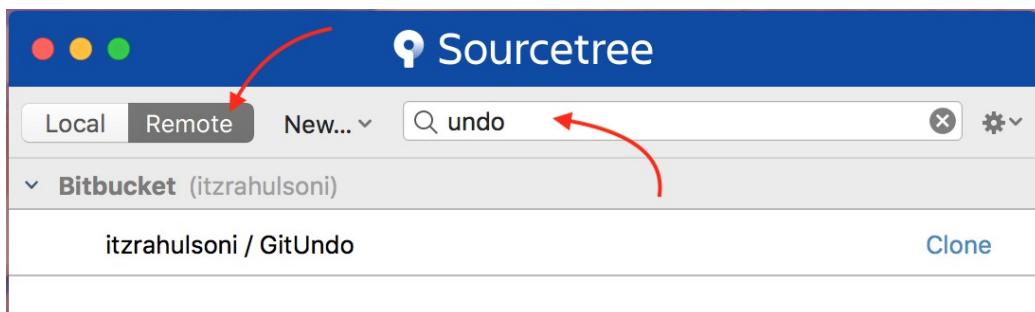
Version control system: Git Mercurial

[Advanced settings](#)

[Create repository](#) [Cancel](#)

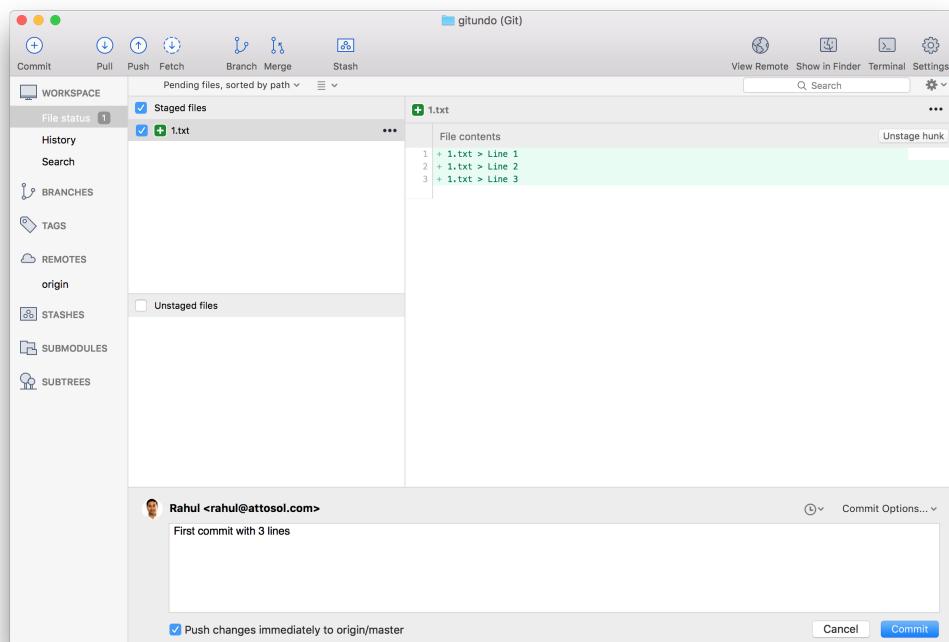
Step 2: Clone it locally

Open SourceTree and switch to [Remote](#). You can filter using the search bar to find your repository. Click on the [Clone](#) option.

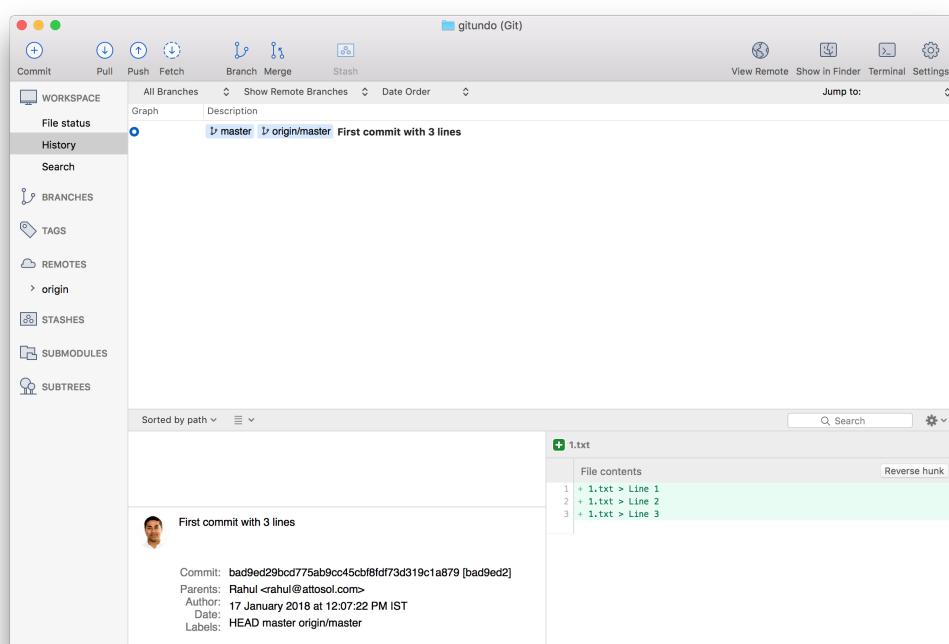


Step 3: Commit a file

By default, with SourceTree you will commit a file and **push** the changes immediately. Notice the check box just below the commit message.

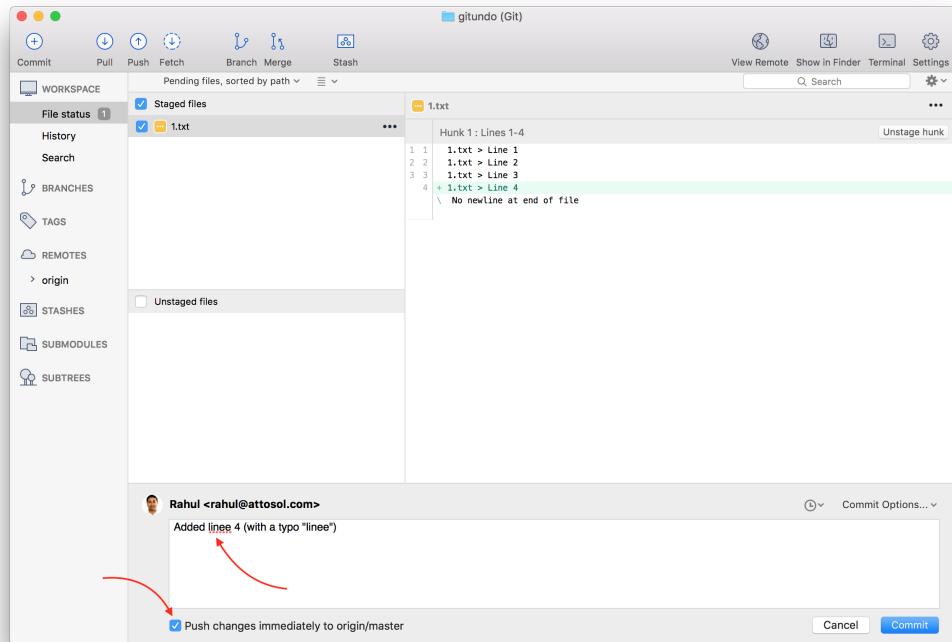


You can view the history and if everything is good, you will see the messages along with all the commits.

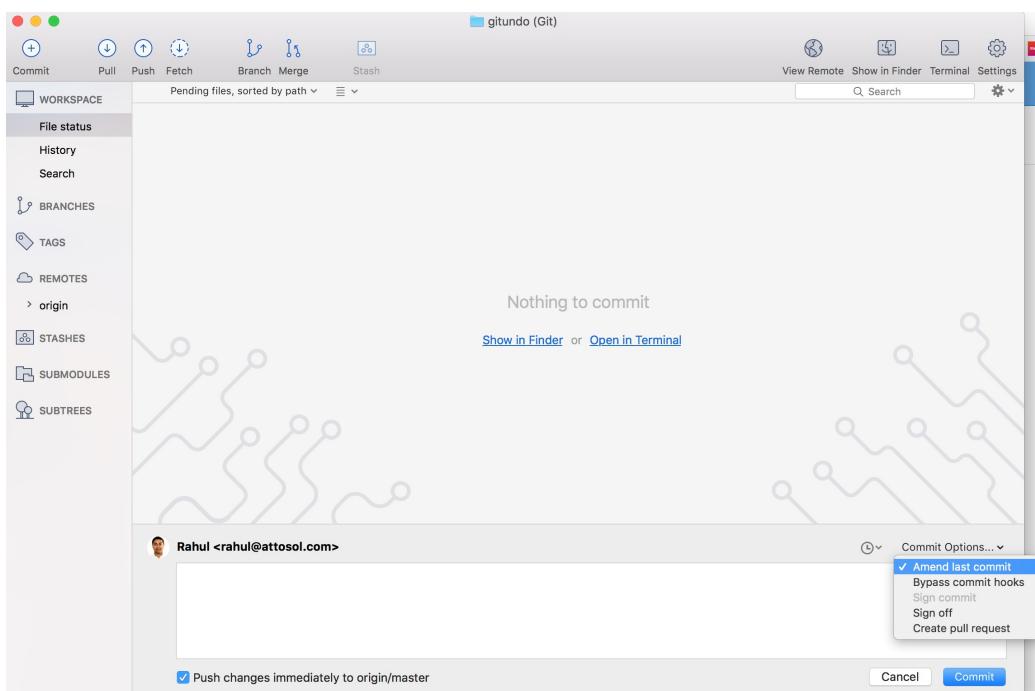


Case 1: How to fix the last commit message?

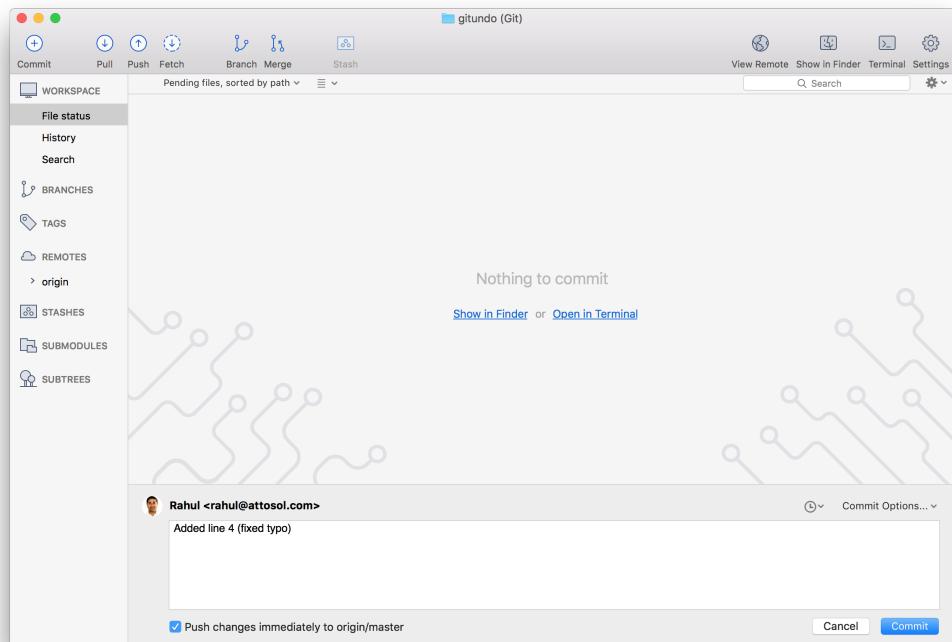
To demonstrate this case, I have modified the `1.txt` file and intentionally introduced a typo in the commit message.



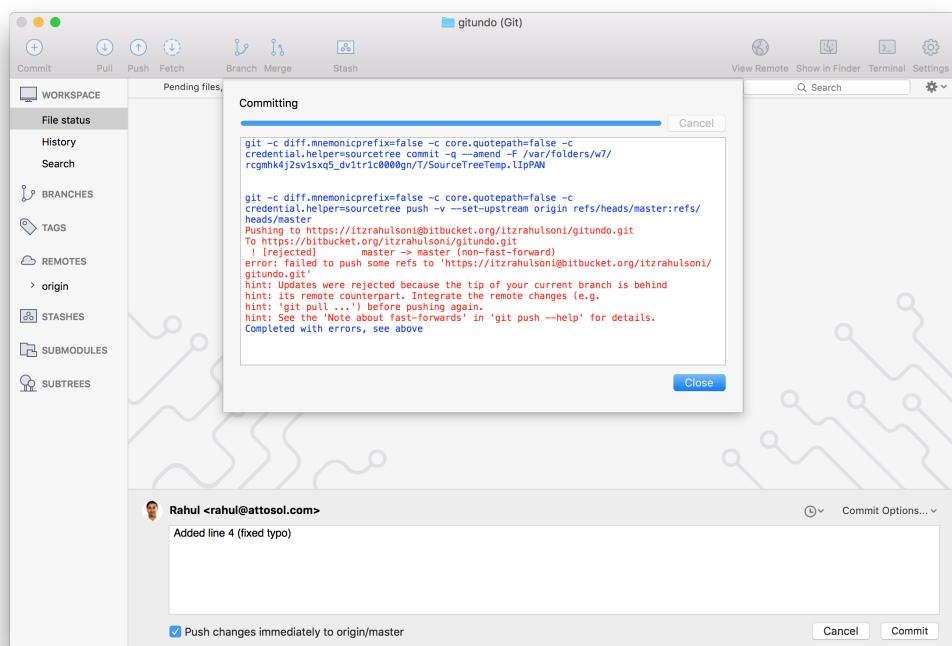
You can fix this error message, clicking on the `Commit` button followed by `Commit Options > Amend Last Commit` option.



Fix the comment message and commit.

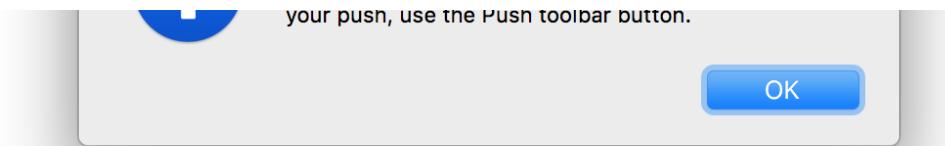


Wait, what? An error message?? Why did that happen???

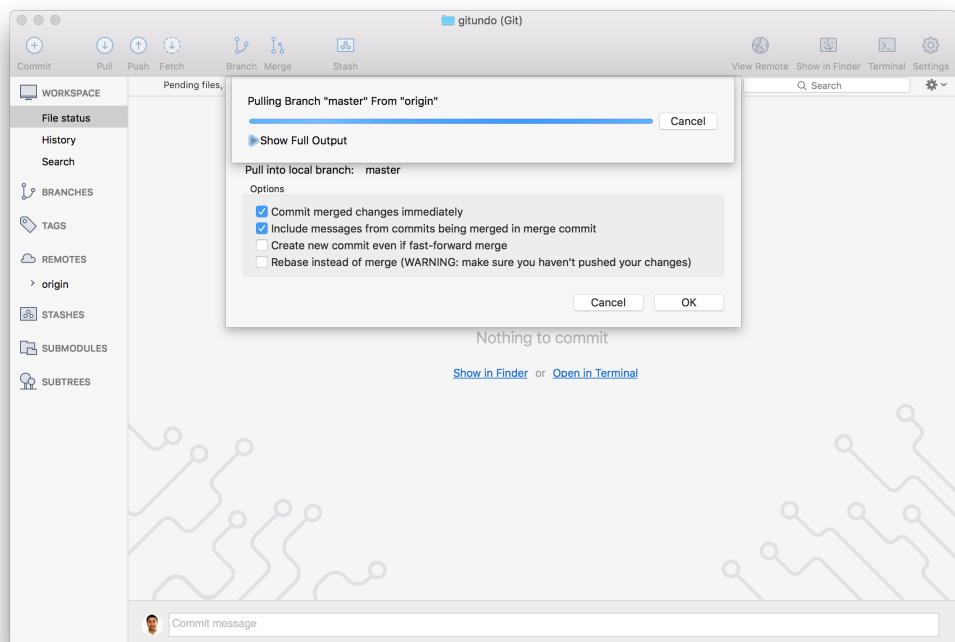


Push Failed

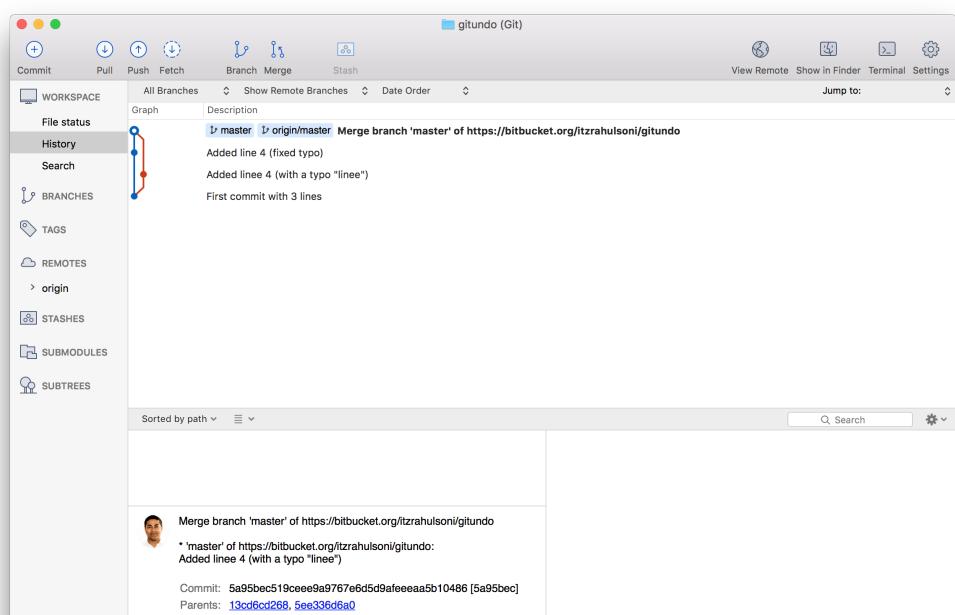
Your commit succeeded but your push failed. To retry



If you read it carefully, you will find that it gives you a hint about pulling from remote and pushing again. Let's do that! Pull it down.



AND, push! The history clarifies what went on behind the scenes.

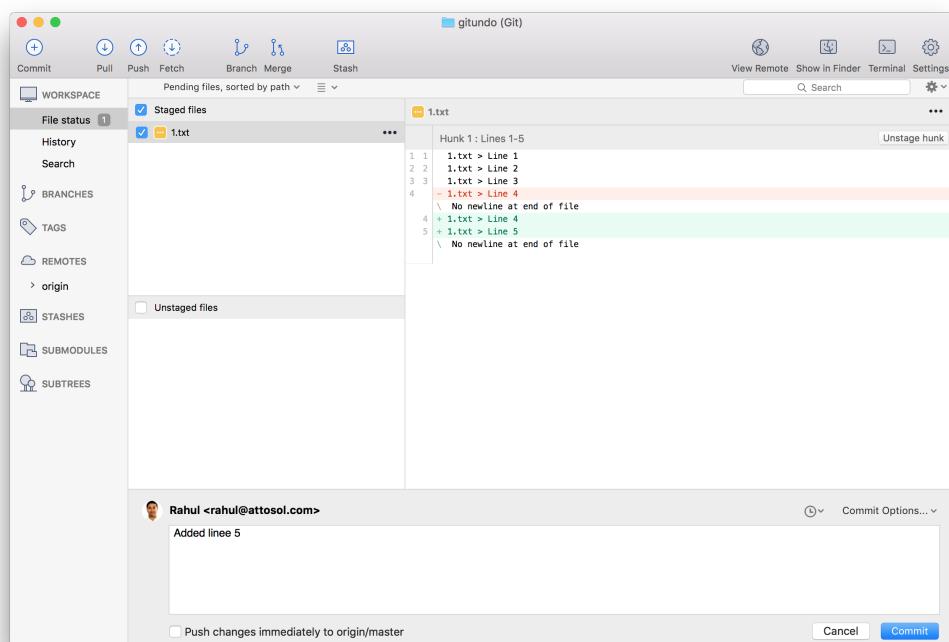


Author: Rahul <rahul@attosol.com>
Date: 17 January 2018 at 12:16:03 PM IST

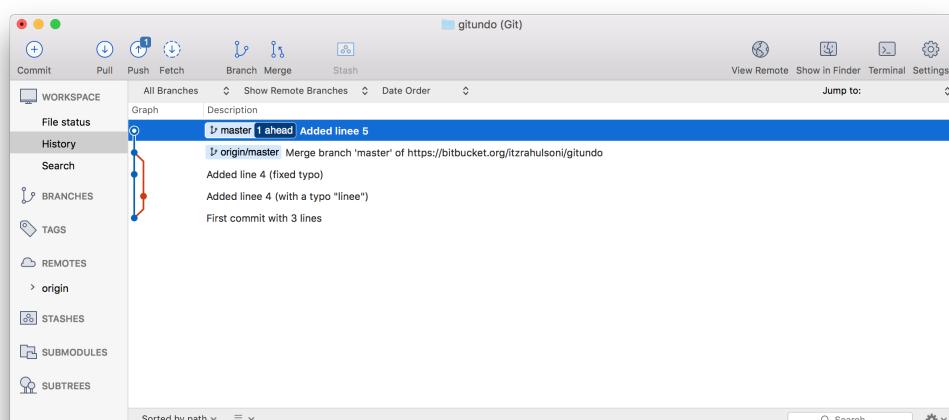
How to fix the ugliness though? It would have been great if there was no tree and merge here, right? So, basically this happened due to the fact that we are pushing the changes instantly.

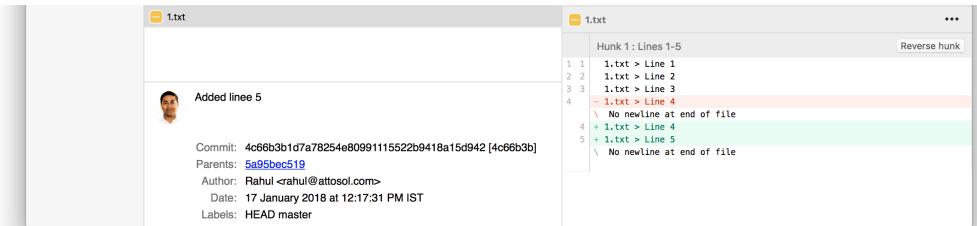
Fix: Don't push instantly to remote!

Make some more changes and commit. Notice that the check box **Push changes immediately to origin/master** is NOT checked.

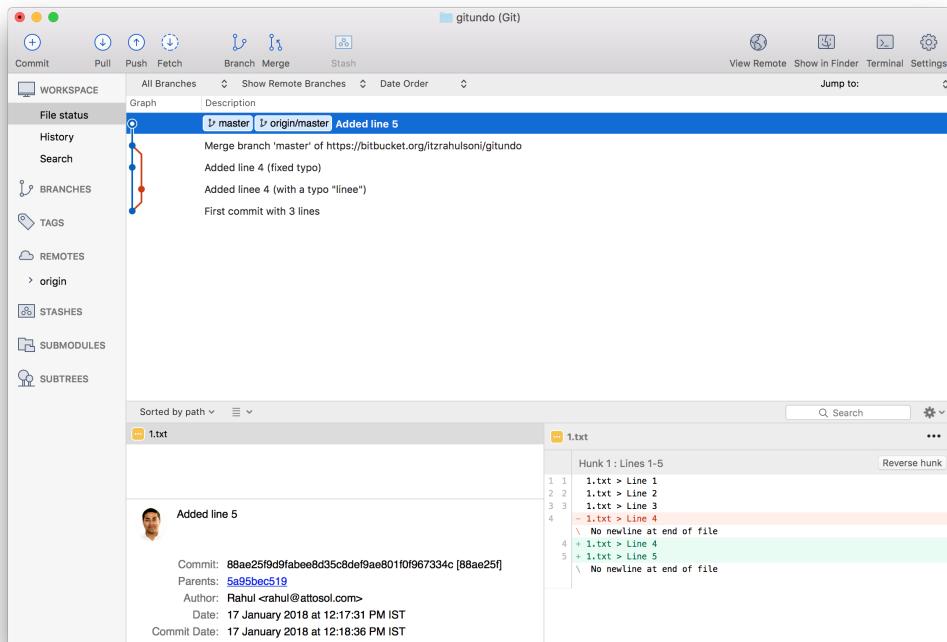


Now, check the history. You will find the **local master** 1 ahead. This is good. You get the goodness of git locally, without changing the remote yet. Also notice the typo in the commit. It reads **linee** instead of **line**.

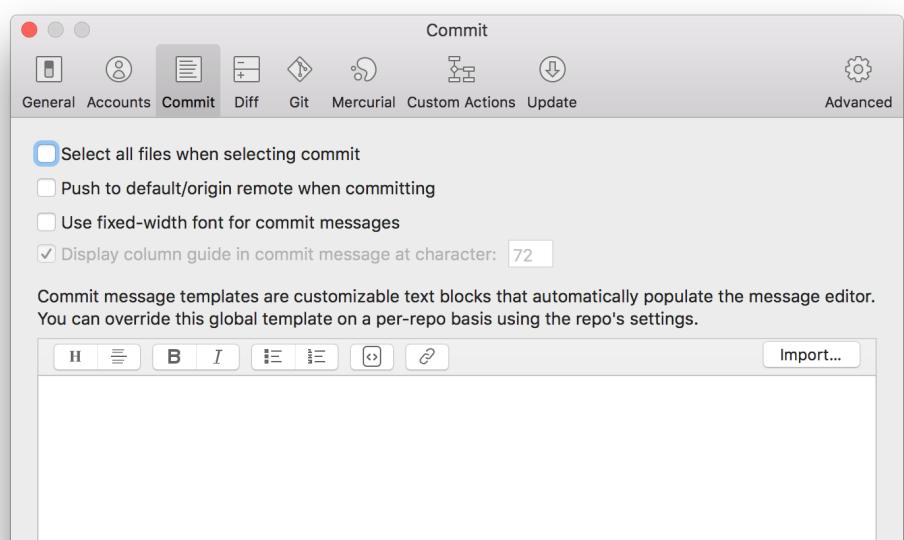




Try [Amend last commit](#) again. This time, you will find that the message appropriately changes without creating a split in the tree structure!

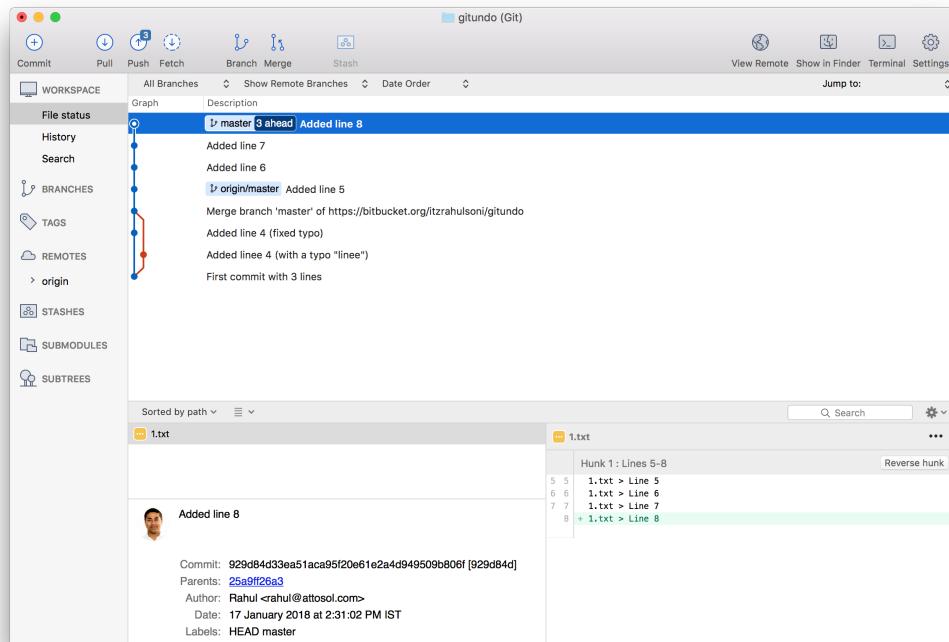


You can set up your preferences in SourceTree so that it doesn't push automatically.

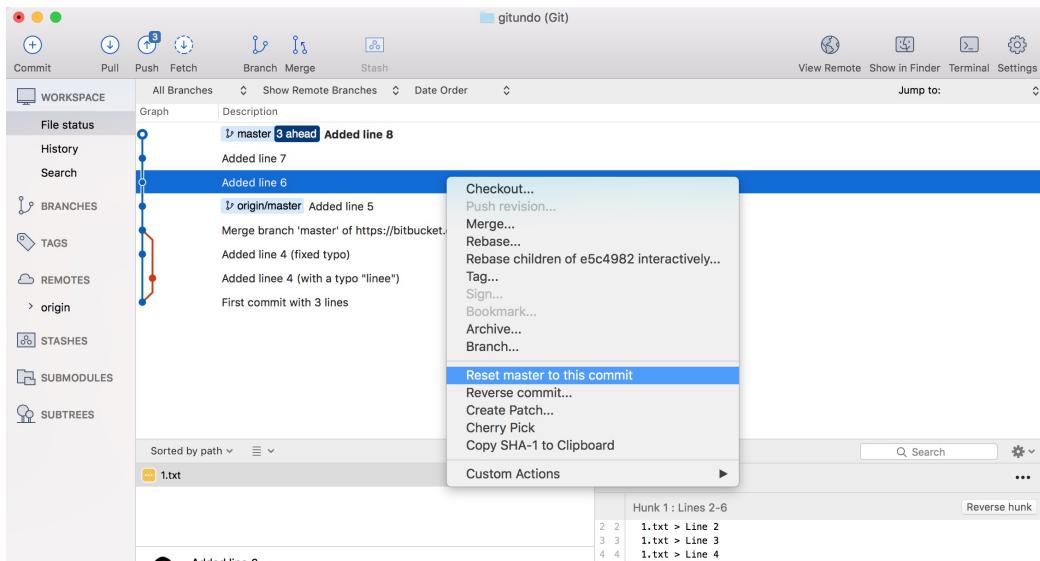


Case 2: Undo changes by Soft Reset

So, let's add a line to the `1.txt` followed by a commit. Repeat this exercise three times, so that you have a history similar to the following:



The requirement here is to reset to the commit where we `Added line 6` and NOT lose the changes after that (like Line 7 & 8). In this case, you should do a `Soft Reset`. Right click and choose `Reset master to this commit`.



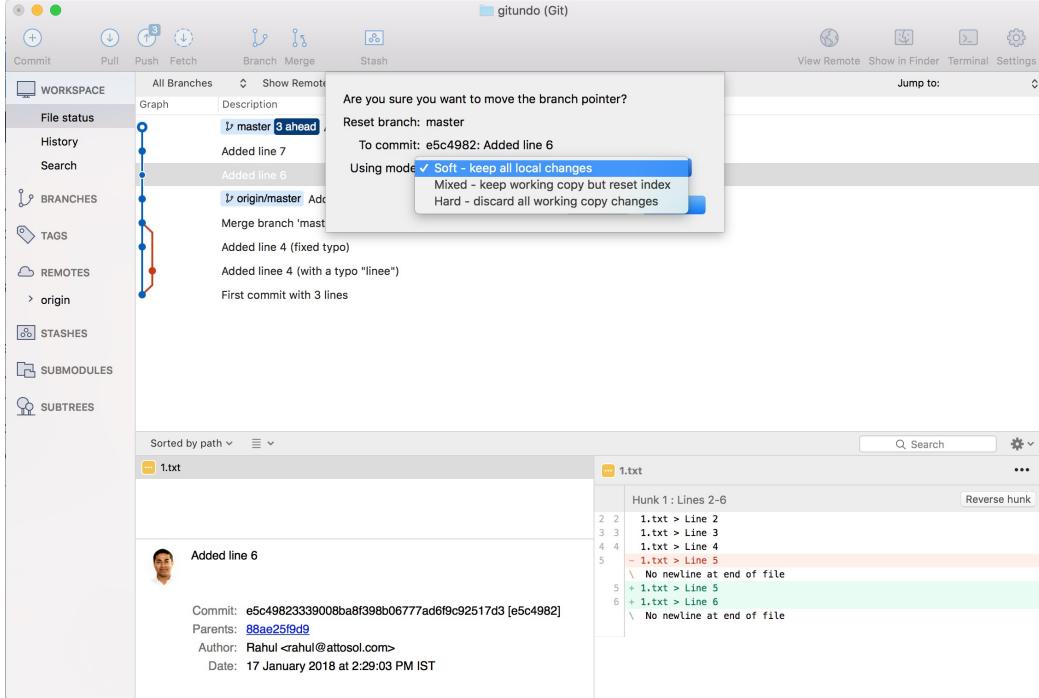
```

Commit: e5c49823339008ba8f398b06777ad6f9c92517d3 [e5c4982]
Parents: 88ae25f9d9
Author: Rahul <rahul@attosol.com>
Date: 17 January 2018 at 2:29:03 PM IST

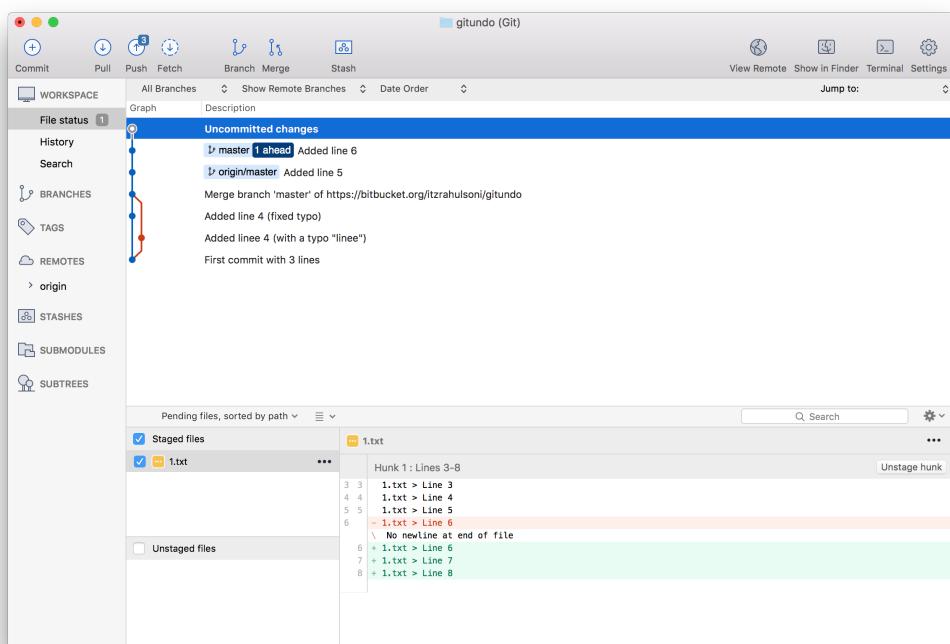
5 - 1.txt > Line 5
\ No newline at end of file
5 + 1.txt > Line 5
6 + 1.txt > Line 6
\ No newline at end of file

```

In the option that appears, select **Soft - keep all local changes**

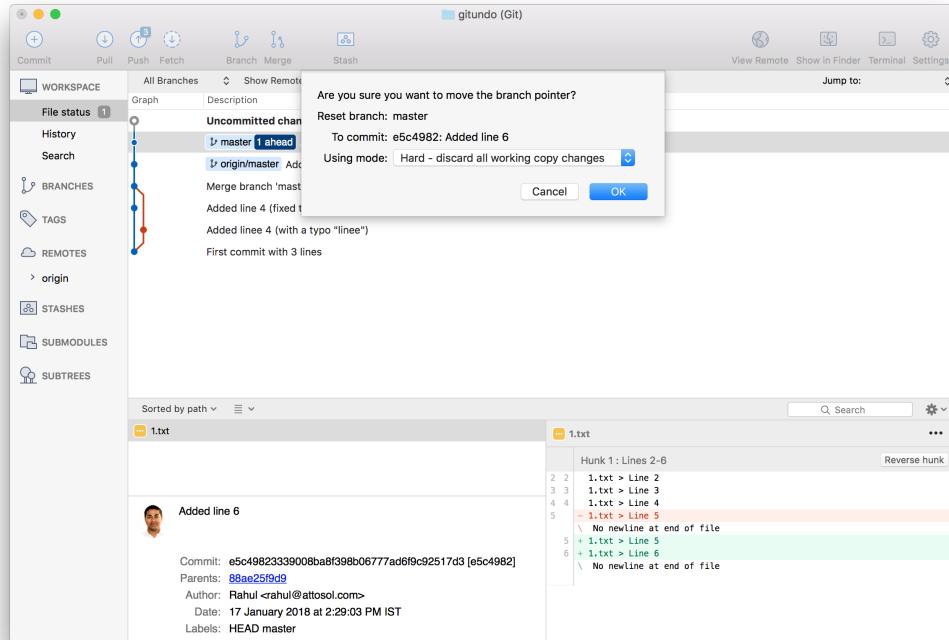


Once done, you can check the result. Notice the **Uncommitted changes**. So, basically all lines are still intact and you are good to go. Also note, that **1.txt** is still staged.



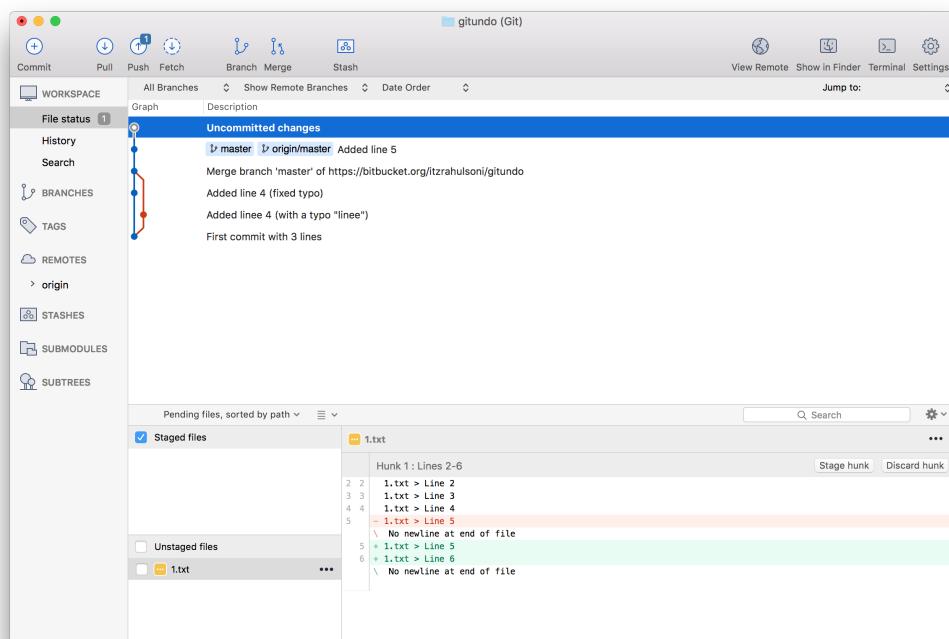
Case 3: Undo changes by Hard Reset

It is quite similar to Reset when it comes to the way it handles history. However, it discards all the local changes. Hence you should use it with caution.



Case 4: Undo changes by Mixed Reset

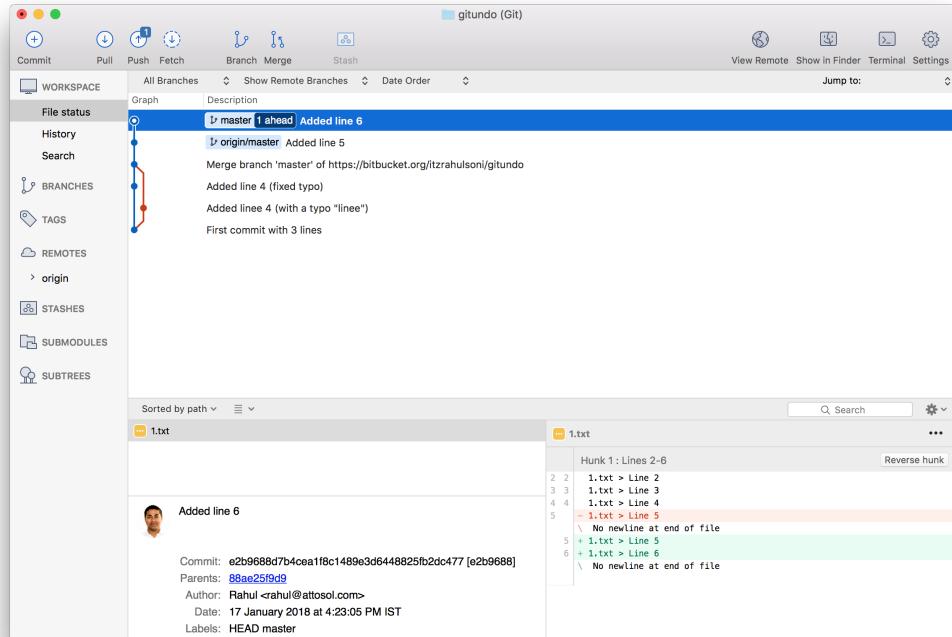
This is again similar to Reset when it comes to the way it handles history. However, it unstages the files too while keep the changes. Note that `1.txt` is now unstaged.



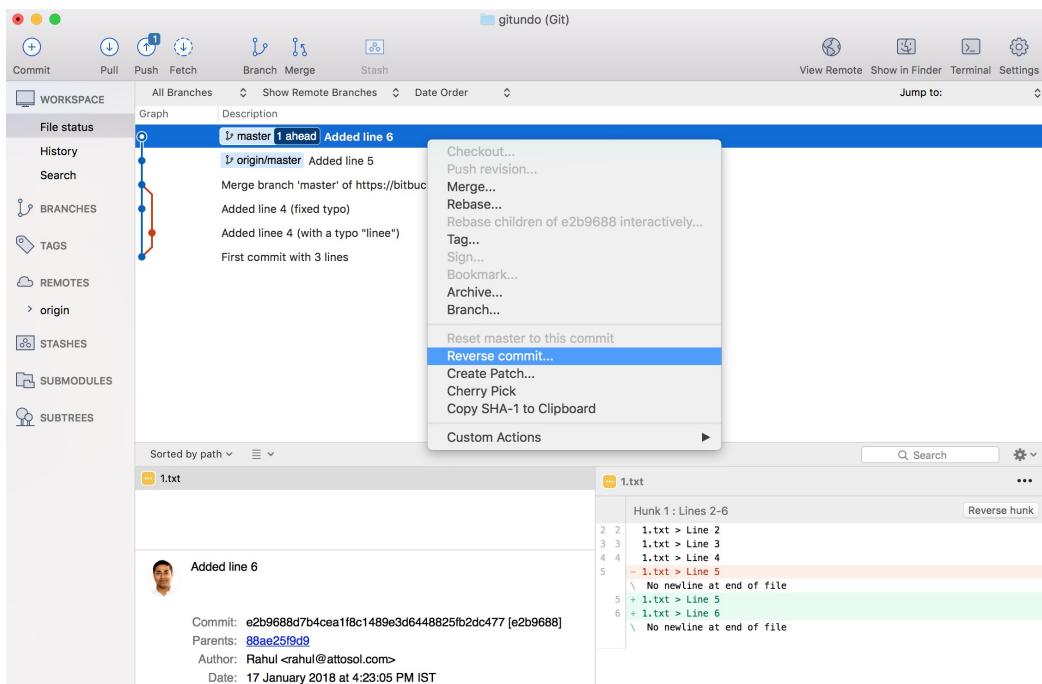
Case 5: Undo using Revert

When you use Revert, basically you don't rewrite the history. You actually, revert the changes using a new commit. Check the example below.

Original history

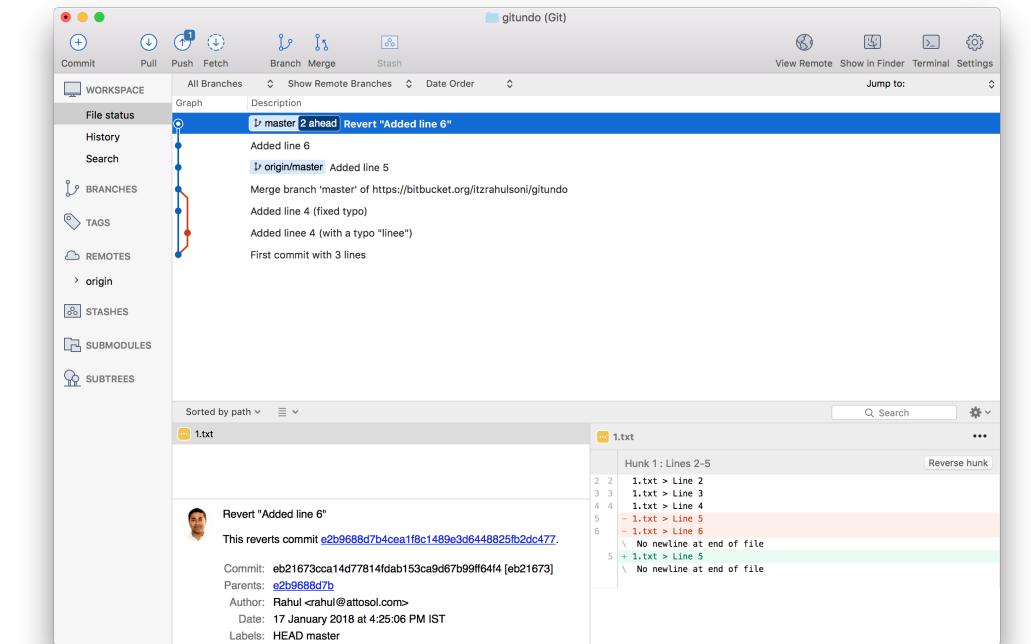


Reverting it...





And now you have...



Case 6: Reset vs Rebase

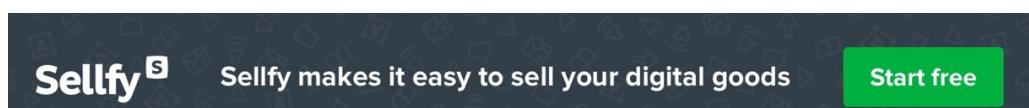
Jarrod Spillers has explained it [beautifully here](#). **TL;DR** > Don't rebase branches you have shared with another developer.

Hope this helps! Happy Coding :-)

What next?

Well, stay tuned for upcoming articles. Say *hi*, share this article, leave a comment or [Subscribe now](#) to stay updated through our newsletter. Also, check out [our services](#) or contact us at contact@attosol.com for your software and consultancy requirements.

Ads:



Rahul Soni

⌘⌘ Entrepreneur. Author. Geek. ⌘⌘

📍 Kolkata, India

Share this post



Subscribe to Attosol Technologies

Get the latest posts delivered right to your inbox.

Your email address

Subscribe

or subscribe [via RSS](#) with Feedly!

1 Comment

Attosol Technologies

🔒 [Disqus' Privacy Policy](#)

Login

Recommend 2

Tweet

Share

Sort by Best



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Dheeraj • a year ago

This was an excellent tutorial. Thank you so much for the step-by-step instructions with screenshots and a explanatory narrative!

• Reply • Share >

READ THIS NEXT

Learn why Identity is a common denominator and SAVES \$\$

Learn why identity management is a common denominator, how you can

YOU MIGHT ENJOY

Benefits of Azure Active Directory

Microsoft Azure is a vast offering which pivots around Azure Active Directory. Learn about its core benefits and the value it provides for your enterprise....

SAVE COST \$\$ and secure your data
effectively with automation and great
user experience....

COMPANY INFO

[About Us](#)

[Careers](#)

[Contact](#)

IMPORTANT LINKS

[Home](#)

[Services](#)

[Attotags](#)

LEARN & COLLABORATE

[Nginx - From Beginner to Pro](#)

[Write For Us](#)

[Subscribe](#)



Attosol Technologies © 2015