Introduction to Git for Research and Software Project

By Eufrat Tsaqib

eufat.github.io/about

Download this presentation at:

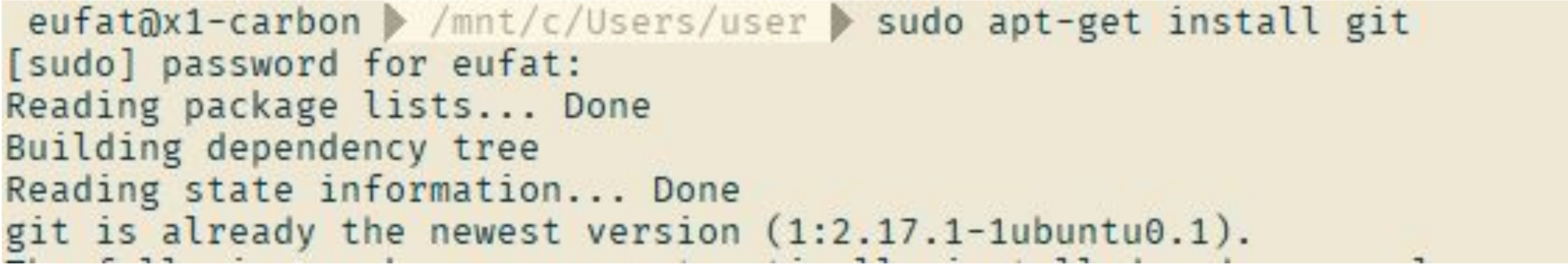
eufat.github.io/docs/introgit.pdf

Do this before we start ...

1. Let’s install Git

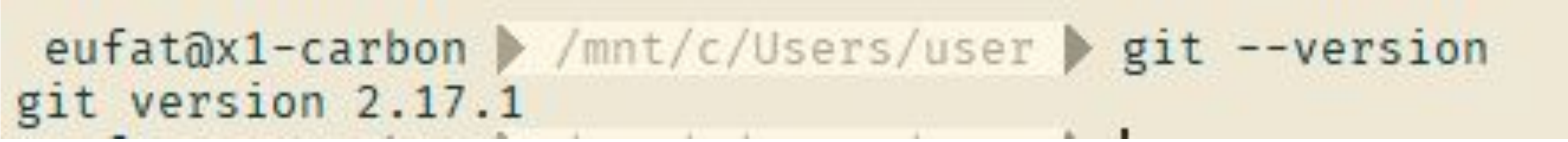
On Windows or Mac: https://git-scm.com/downloads

On Linux Ubuntu:



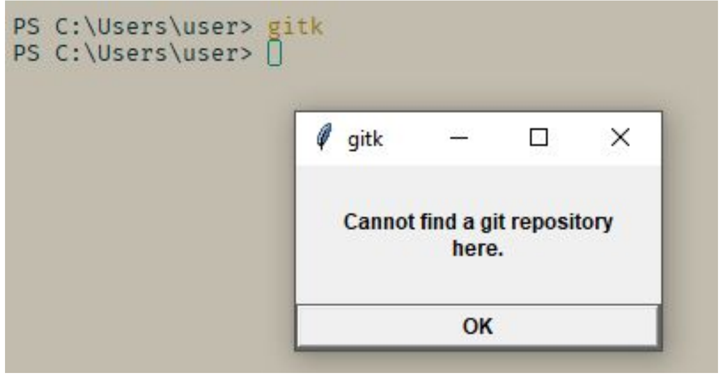
```
 eufat@x1-carbon ▶ /mnt/c/Users/user ▶ sudo apt-get install git
[sudo] password for eufat:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.1).
```

2. Check newly installed git



```
 eufat@x1-carbon ▶ /mnt/c/Users/user ▶ git --version
git version 2.17.1
```

3. (Optional) Check git repository browser “gitk”



```
PS C:\Users\user> gitk
PS C:\Users\user>
```

Cannot find a git repository here.

4. (Optional) Create github account

Presentation Outline

Chapter 0: Background

- Basic Versioning
- Basic Collaboration

Chapter 1: Git as VCS

- Overview
- Initialization and Configuration
- File Lifecycle (stage, commit)
- Log and Gitk
- Diff
- Head (with caret and tilde)
- Reset (soft, mixed and hard)
- Reflog

Chapter 2: Git as Collaboration Tools

- Overview
- Branching
- Merge vs Rebase
- Blaming
- Remote Repository

Chapter 3: Git Supporting Tools

Chapter N: Your Next Chapter

Chapter 0:
Background

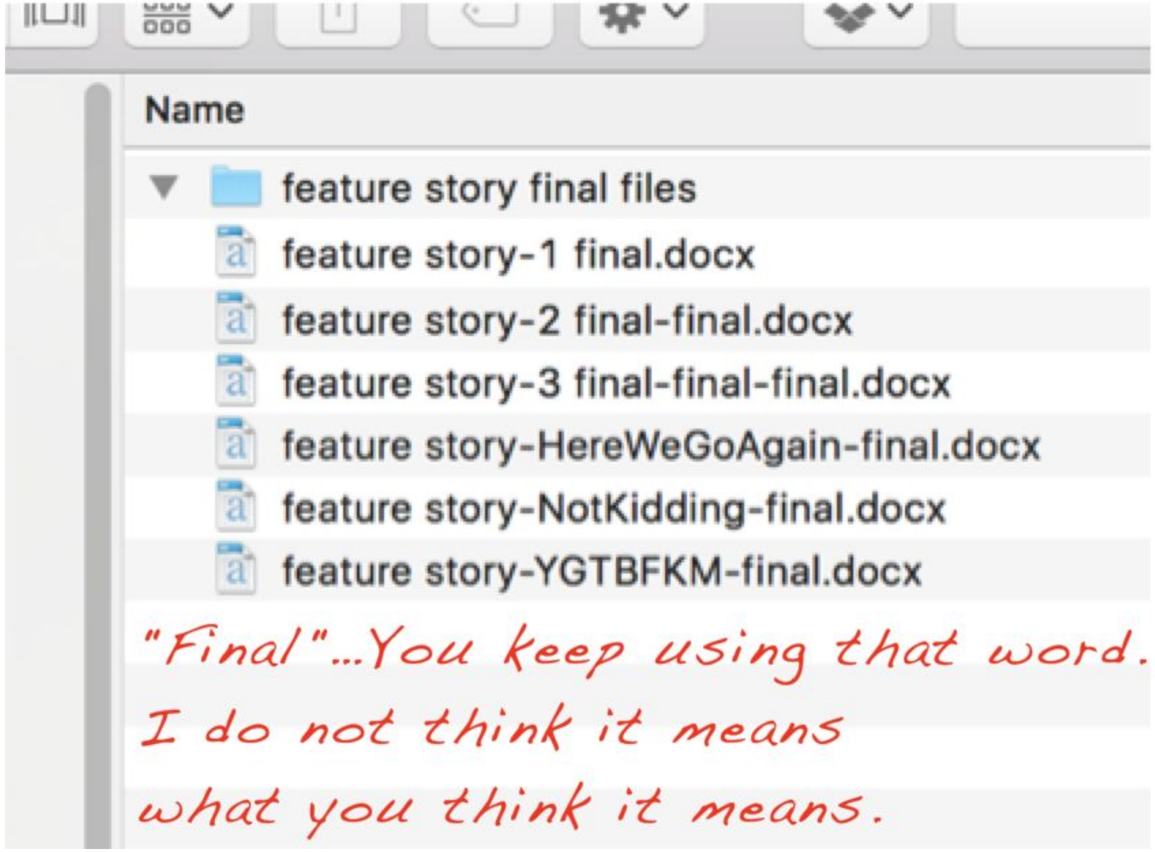
Basic Versioning

- Write document/program, save.
- Add numbering: (v1.2, 1-5, 2.3 dst).
- Add an “I feels like I am done” words (A never ending of “final”, “complete”, “done”.)

Untrackable

Bad Semantic

Unstructured



feature story final files

feature story-1 final.docx

feature story-2 final-final.docx

feature story-3 final-final-final.docx

feature story-HereWeGoAgain-final.docx

feature story-NotKidding-final.docx

feature story-YGTBFKM-final.docx

"Final"...You keep using that word. I do not think it means what you think it means.

Basic Collaboration

- Use email to send code or document: Use gmail, outlook etc.
- Use chat messaging app to collaborate: Copy-paste code to whatsapp, line etc.
- Use cloud storage to collaborate codes: Use dropbox, google drive to backup folders and share to the team.

Solution: Git

Git designed for text-based data.
ex: codes, books, papers, article etc.

Can be used with binary data (not to useful): ex: music, image, executable etc.

Git is NOT GitHub!

GitHub ***is a web-based hosting service***
for version control using Git.

Chapter 1:
Git as VCS

Git as VCS: Overview

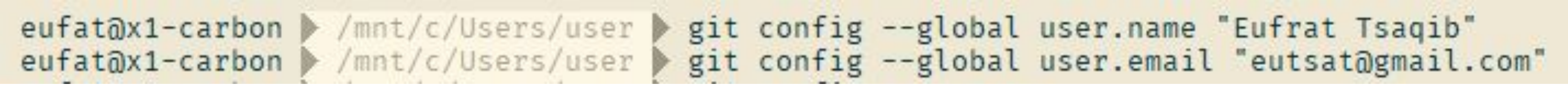
- VCS: Version Control System

  records changes to a file or set of files over time so that you can recall specific versions later.

- Distributed: Git could distributed to users that has the repo.
- Included in (most) Linux: Git preinstalled on Linux
- Widely used: Almost all tech companies use Git, even CERN use Git.
- Tiny and Fast: Git has small footprint and lightweight.
- Easy to learn: Git is easy to learn (but hard to master).
- A must known skill when applying to almost any software company on earth

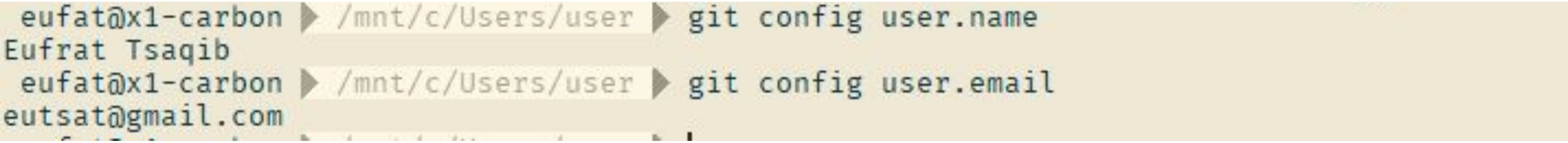
Git as VCS: Configuration

Let’s set our username and email to git



```
 eufat@x1-carbon ▶ /mnt/c/Users/user ▶ git config --global user.name "Eufrat Tsaqib"
 eufat@x1-carbon ▶ /mnt/c/Users/user ▶ git config --global user.email "eutsat@gmail.com"
```

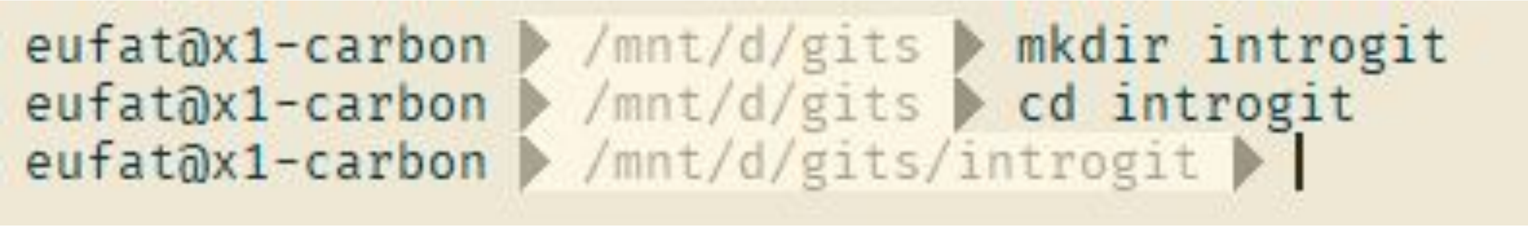
After that we can view our configs with



```
 eufat@x1-carbon ▶ /mnt/c/Users/user ▶ git config user.name
Eufrat Tsaqib
 eufat@x1-carbon ▶ /mnt/c/Users/user ▶ git config user.email
eutsat@gmail.com
```

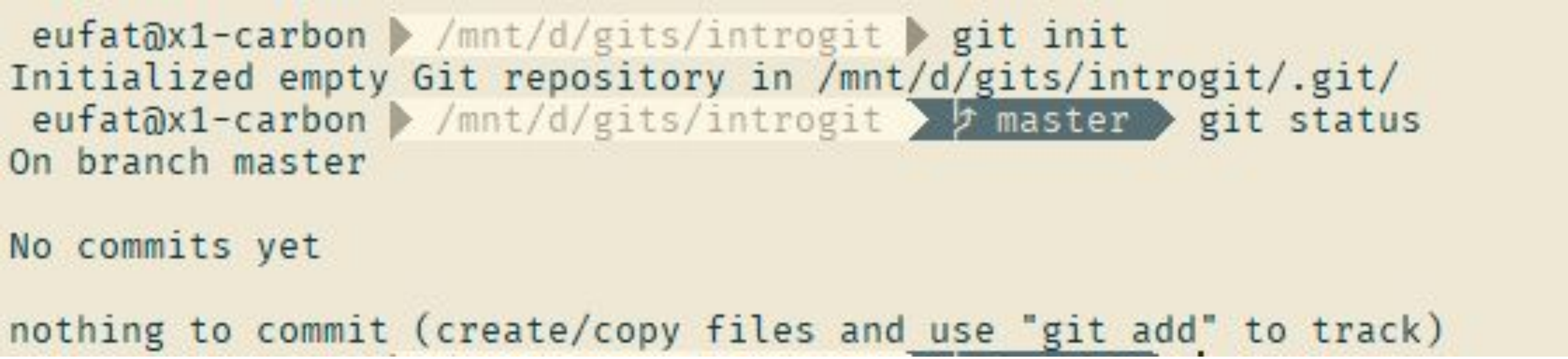
Git as VCS: Initialization

This gonna be easy, create a new folder. Then, change the directory



```
eufat@x1-carbon ▶ /mnt/d/gits ▶ mkdir introgit
eufat@x1-carbon ▶ /mnt/d/gits ▶ cd introgit
eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ |
```

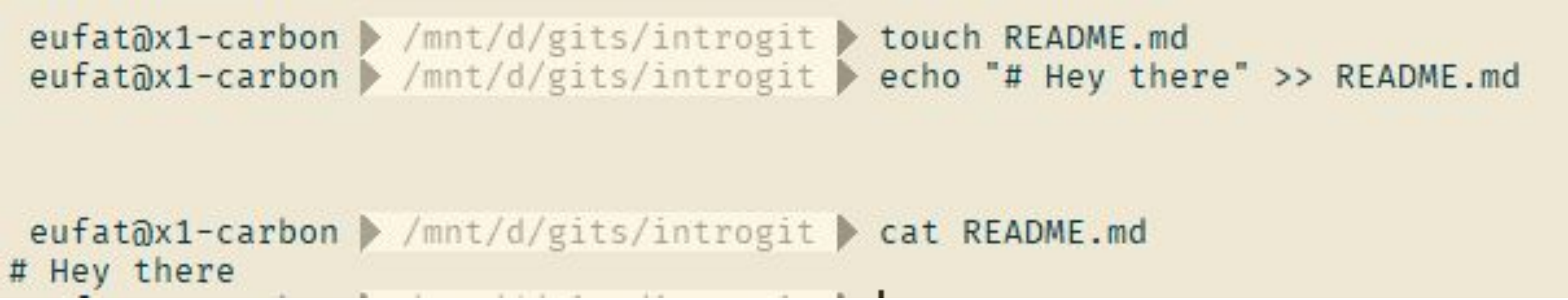
So we can set a *repository* to the folder, by doing this



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ git init
Initialized empty Git repository in /mnt/d/gits/introgit/.git/
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ ⎇ master ▶ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

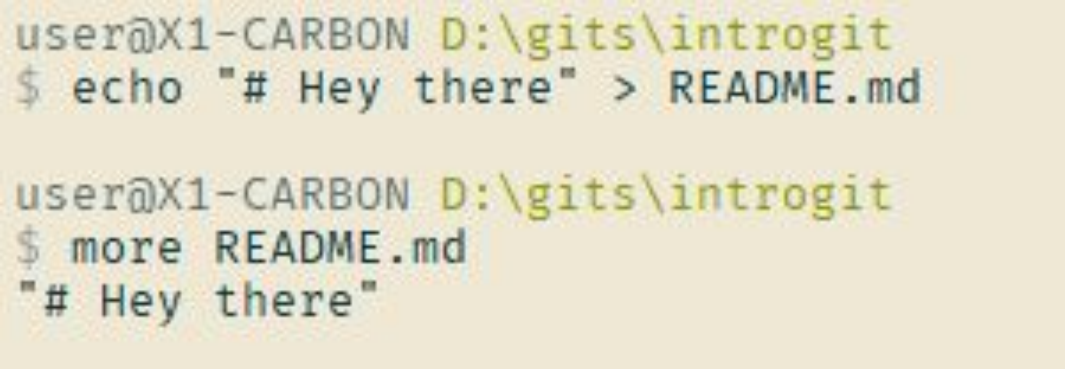
Git as VCS: Create a file

Let’s create a file named README.md and add some content to it



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ touch README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ echo "# Hey there" >> README.md



 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ cat README.md
# Hey there
```

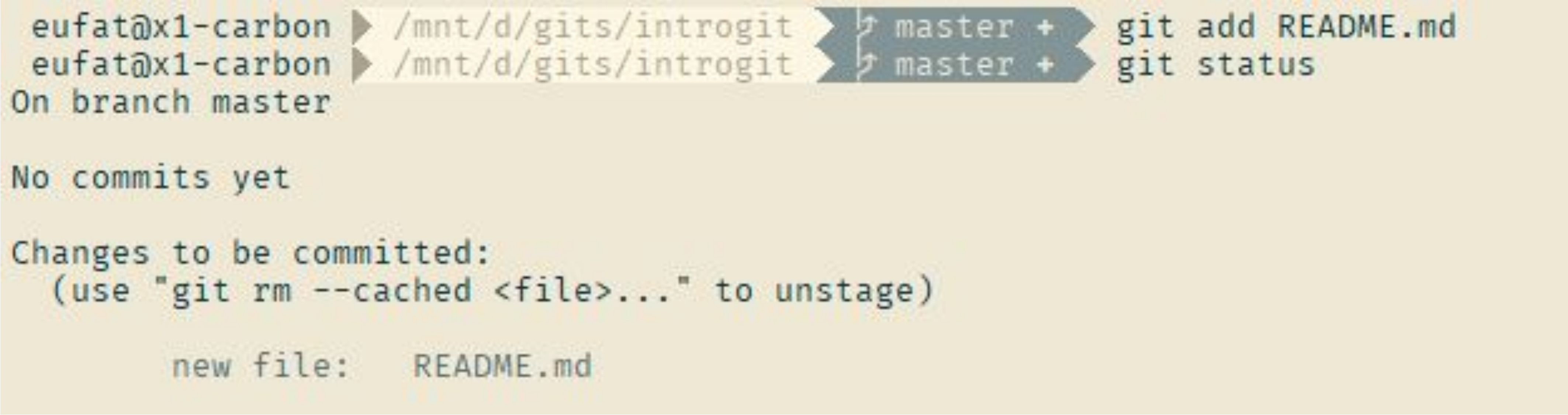
In Windows use this command, it should works the same



```
user@X1-CARBON D:\gits\introgit
$ echo "# Hey there" > README.md

user@X1-CARBON D:\gits\introgit
$ more README.md
"# Hey there"
```

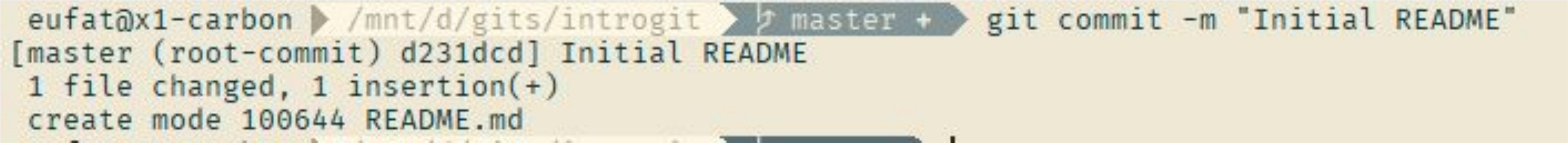
Git as VCS: Add and Commit

Now we have README.md, add that to “staging area”



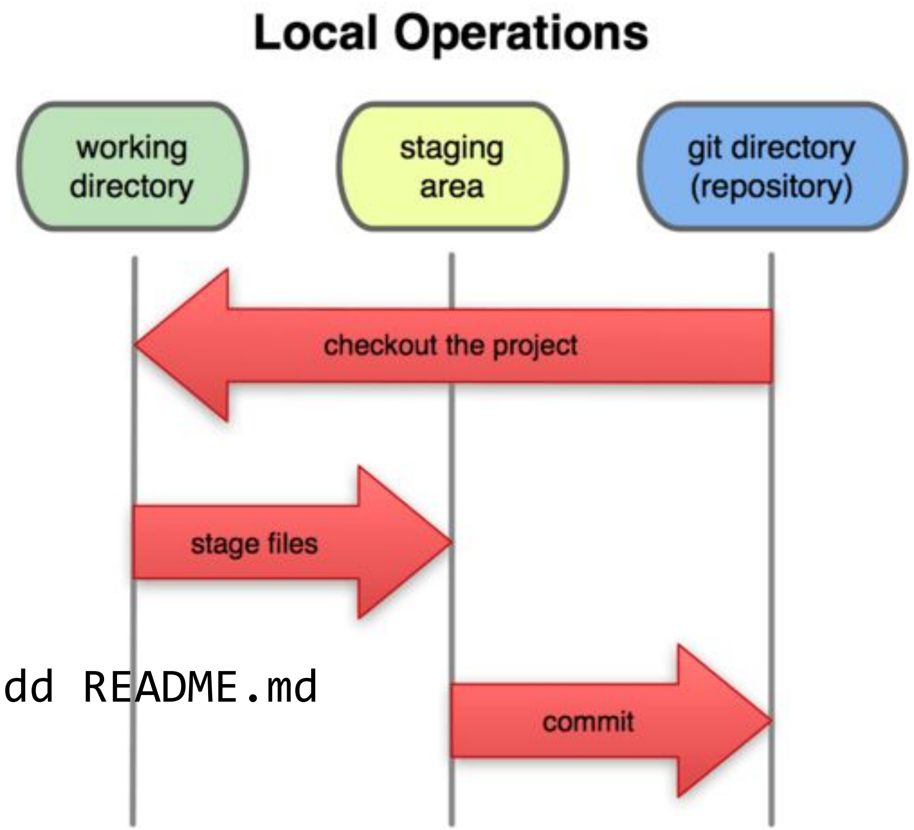
```
 eufat@x1-carbon  /mnt/d/gits/introgit   master +  git add README.md
 eufat@x1-carbon  /mnt/d/gits/introgit   master +  git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
```

“Commit” the file to repository



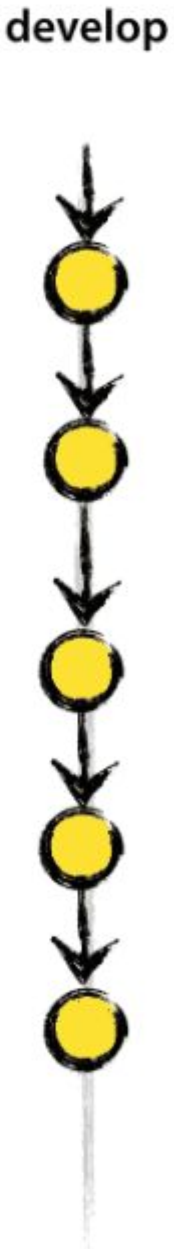
```
 eufat@x1-carbon  /mnt/d/gits/introgit   master +  git commit -m "Initial README"
[master (root-commit) d231dcd] Initial README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Git as VCS: File Lifecycle



```
$ git add README.md
```

```
$ git commit –m “Initial README.md”
```

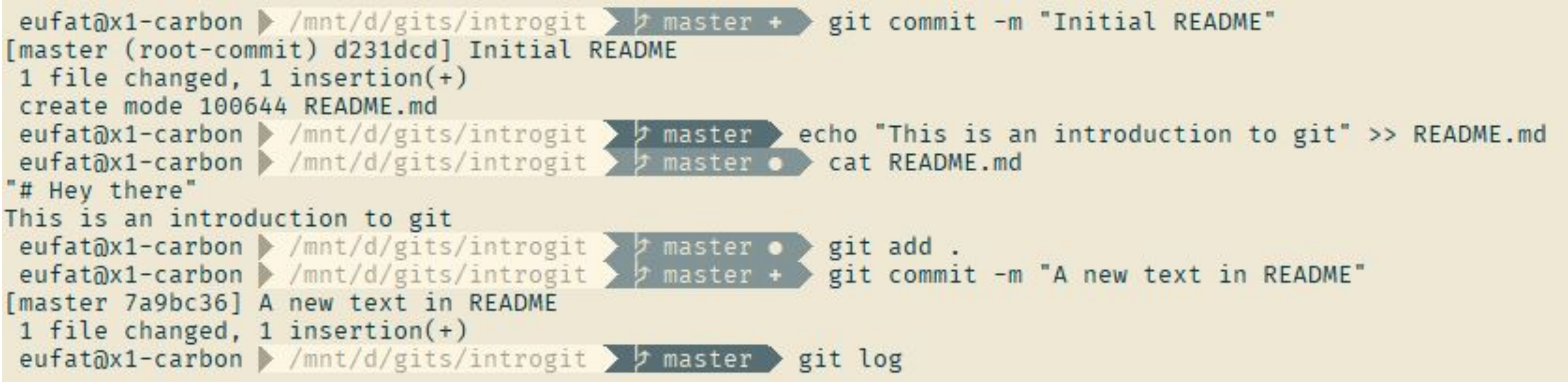


Commit ID

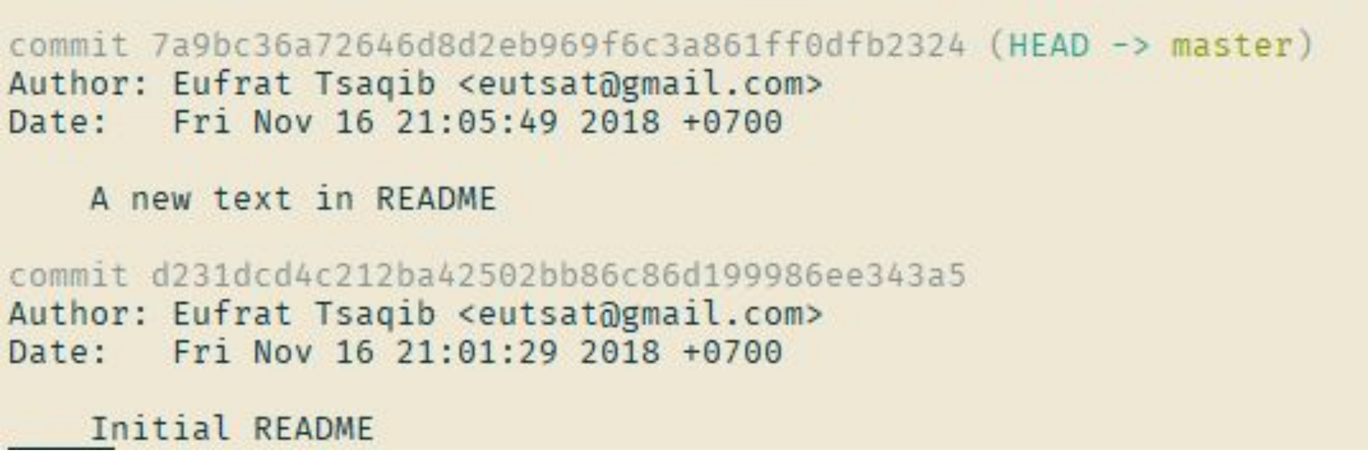
```
Commit abc123: Add main.c

Commit def456: Update main.c
With cool function.

Commit hij789: Update main.c
With infinite loop.

Commit lmn101: Fix main.c
From undefined variable.

Commit opq112: Add recursive
Function to replace
infinite loop.
```

Git as VCS: Log

Edit the README.md file, add to staging, commit and view the log



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master + ▶ git commit -m "Initial README"
[master (root-commit) d231dcd] Initial README
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ▶ echo "This is an introduction to git" >> README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ cat README.md
"# Hey there"
This is an introduction to git
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ git add .
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master + ▶ git commit -m "A new text in README"
[master 7a9bc36] A new text in README
 1 file changed, 1 insertion(+)
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ▶ git log
```

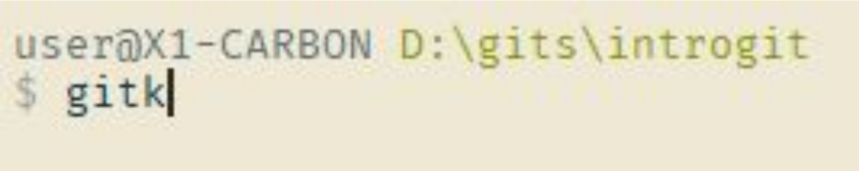


```
commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324 (HEAD -> master)
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:05:49 2018 +0700

    A new text in README

commit d231dcd4c212ba42502bb86c86d199986ee343a5
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:01:29 2018 +0700

    Initial README
```

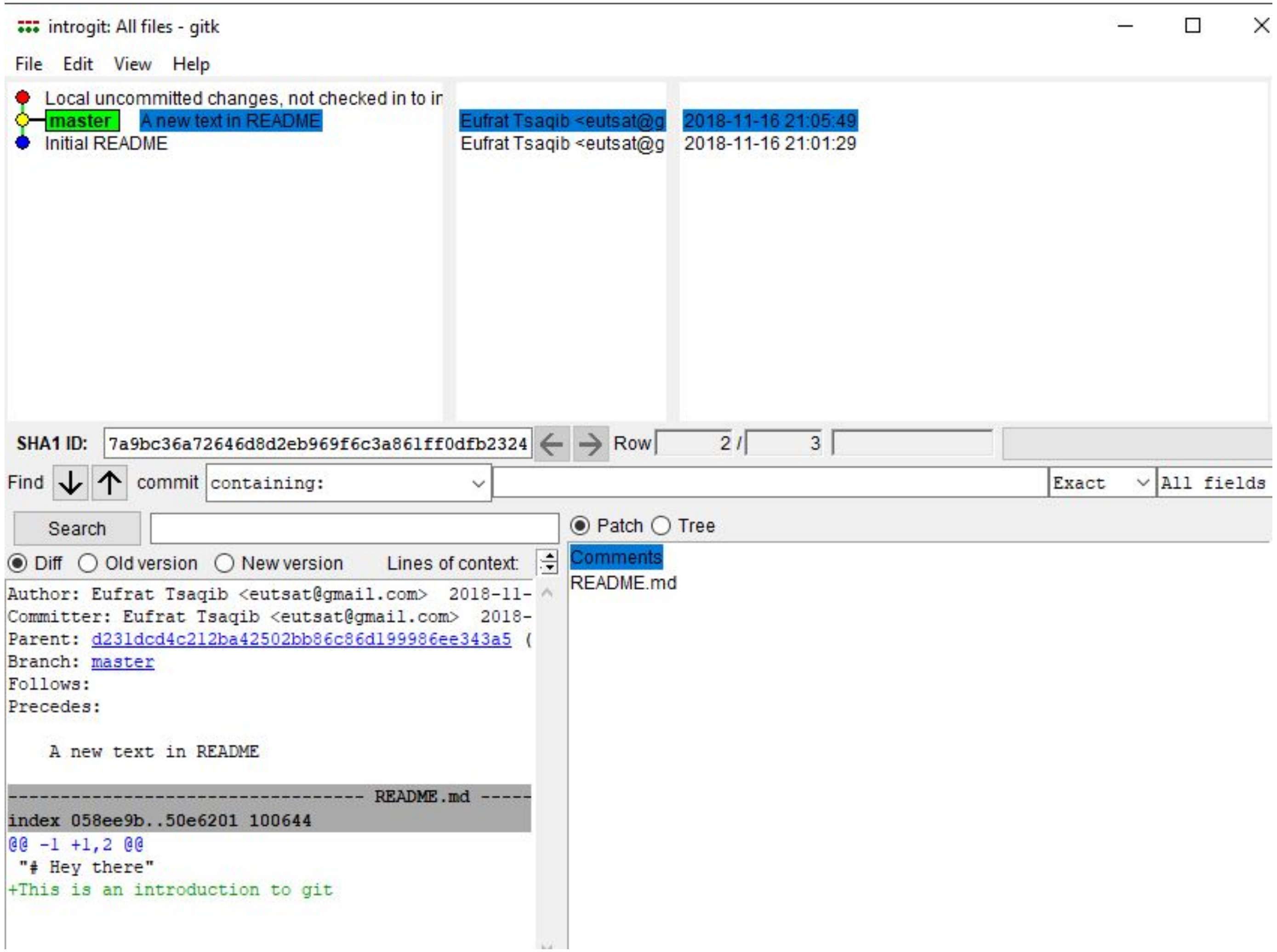
PS: Press "q" to quit log view.

Git as VCS: Gitk

Lets view our history using gitk



```
user@X1-CARBON D:\gits\introgit
$ gitk
```



```
Local uncommitted changes, not checked in to ir
master   A new text in README          Eufrat Tsaqib <eutsat@g   2018-11-16 21:05:49
Initial README                         Eufrat Tsaqib <eutsat@g   2018-11-16 21:01:29
```

SHA1 ID: 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324 Row 2 / 3

Find commit containing: Exact All fields

● Patch ○ Tree

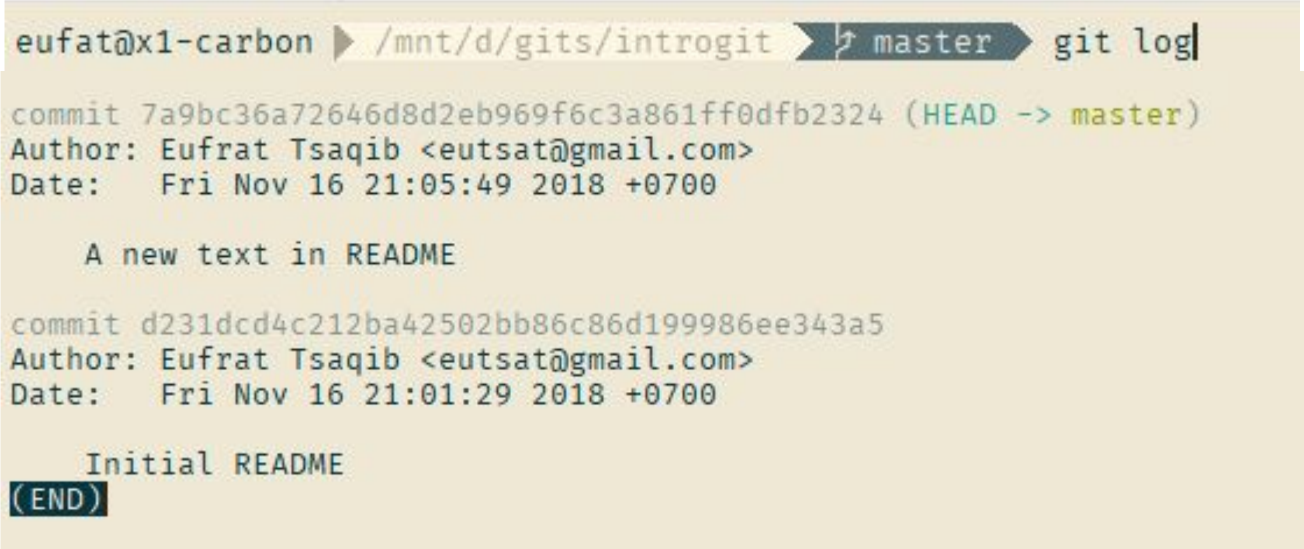
● Diff ○ Old version ○ New version Lines of context:

```
Author: Eufrat Tsaqib <eutsat@gmail.com>  2018-11-
Committer: Eufrat Tsaqib <eutsat@gmail.com>  2018-
Parent: d231dcd4c212ba42502bb86c86d199986ee343a5 (
Branch: master
Follows:
Precedes:

    A new text in README

--------------------------------- README.md -----
index 058ee9b..50e6201 100644
@@ -1 +1,2 @@
 "# Hey there"
+This is an introduction to git
```

Comments
README.md

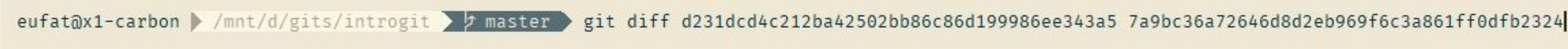
Git as VCS: Diff

Diff takes two parameter either it’s a commit ids, files, or branches

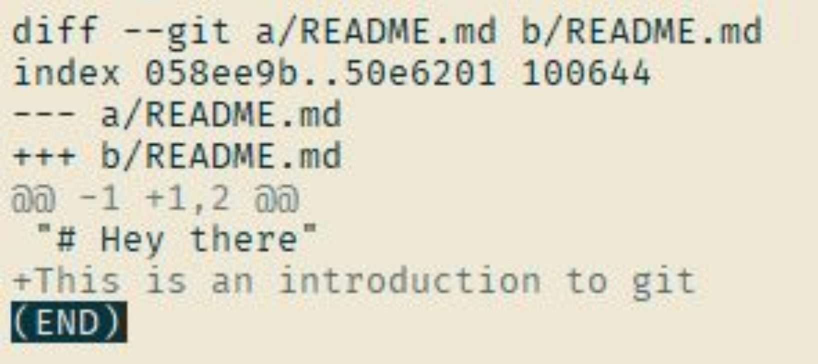


```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git log

commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324 (HEAD -> master)
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:05:49 2018 +0700

    A new text in README

commit d231dcd4c212ba42502bb86c86d199986ee343a5
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:01:29 2018 +0700

    Initial README
(END)
```

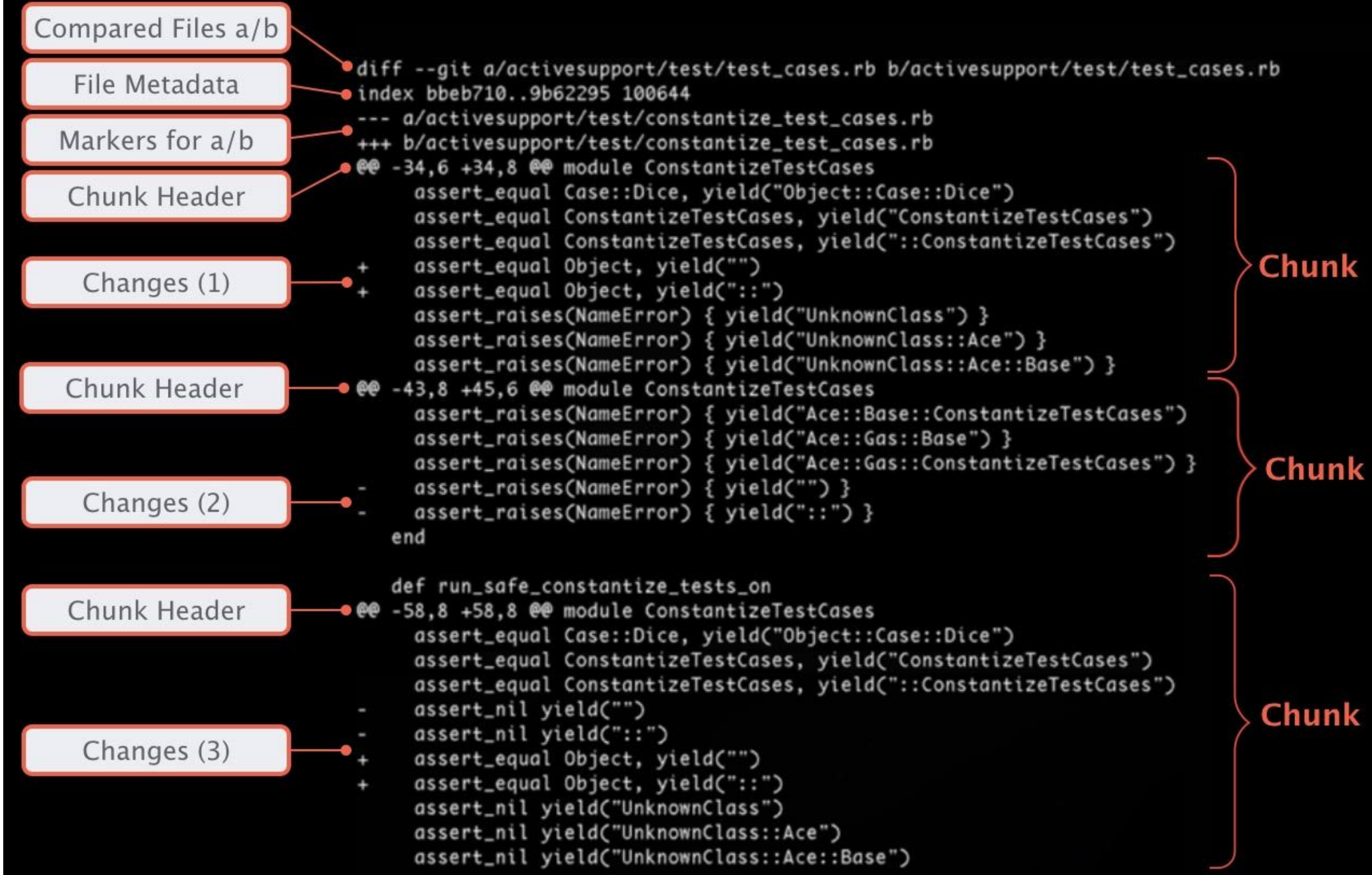
Let’s use those two commit ids and check the diff between the old and new one



```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git diff d231dcd4c212ba42502bb86c86d199986ee343a5 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
```



```
diff --git a/README.md b/README.md
index 058ee9b..50e6201 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,2 @@
 "# Hey there"
+This is an introduction to git
(END)
```

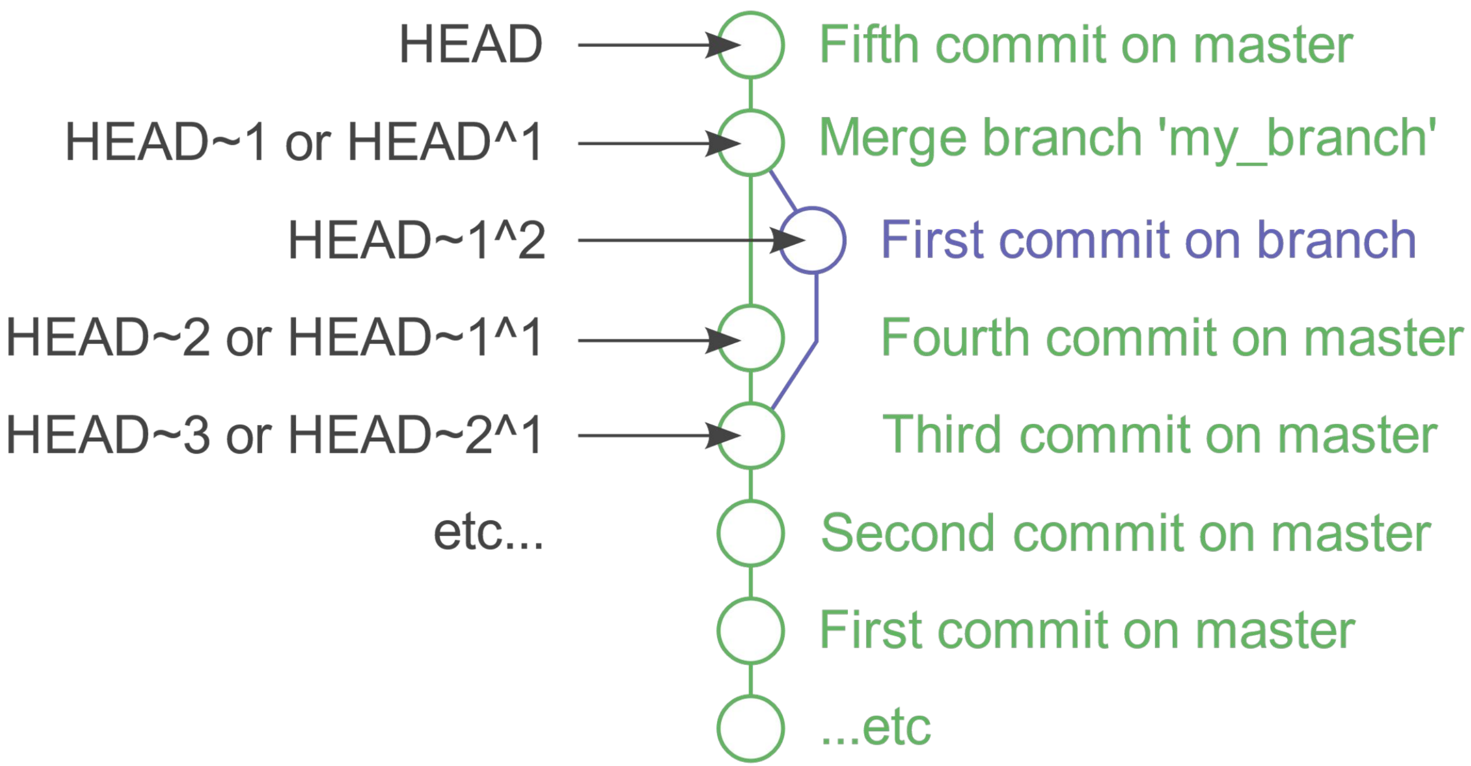


```
diff --git a/activesupport/test/test_cases.rb b/activesupport/test/test_cases.rb
index bbeb710..9b62295 100644
--- a/activesupport/test/constantize_test_cases.rb
+++ b/activesupport/test/constantize_test_cases.rb
@@ -34,6 +34,8 @@ module ConstantizeTestCases
     assert_equal Case::Dice, yield("Object::Case::Dice")
     assert_equal ConstantizeTestCases, yield("ConstantizeTestCases")
     assert_equal ConstantizeTestCases, yield("::ConstantizeTestCases")
+    assert_equal Object, yield("")
+    assert_equal Object, yield("::")
     assert_raises(NameError) { yield("UnknownClass") }
     assert_raises(NameError) { yield("UnknownClass::Ace") }
     assert_raises(NameError) { yield("UnknownClass::Ace::Base") }
@@ -43,8 +45,6 @@ module ConstantizeTestCases
     assert_raises(NameError) { yield("Ace::Base::ConstantizeTestCases") }
     assert_raises(NameError) { yield("Ace::Gas::Base") }
     assert_raises(NameError) { yield("Ace::Gas::ConstantizeTestCases") }
-    assert_raises(NameError) { yield("") }
-    assert_raises(NameError) { yield("::") }
   end

   def run_safe_constantize_tests_on
@@ -58,8 +58,8 @@ module ConstantizeTestCases
     assert_equal Case::Dice, yield("Object::Case::Dice")
     assert_equal ConstantizeTestCases, yield("ConstantizeTestCases")
     assert_equal ConstantizeTestCases, yield("::ConstantizeTestCases")
-    assert_nil yield("")
-    assert_nil yield("::")
+    assert_equal Object, yield("")
+    assert_equal Object, yield("::")
     assert_nil yield("UnknownClass")
     assert_nil yield("UnknownClass::Ace")
     assert_nil yield("UnknownClass::Ace::Base")
```

Git as VCS: Head

HEAD is a reference to the last commit in the current branch.

Referencing commits from HEAD using ~ and ^



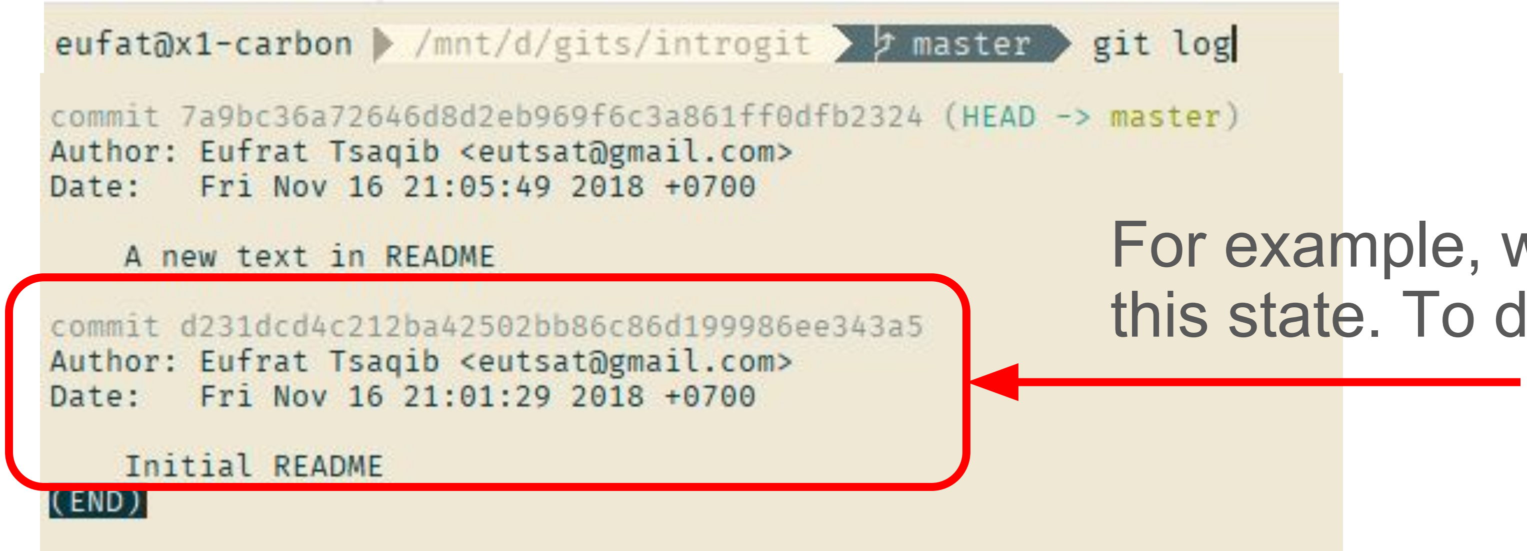
Note about caret (^) and tilde (~)

^N is N-th parent commit.

~N is previous N commit.

Git as VCS: Reset

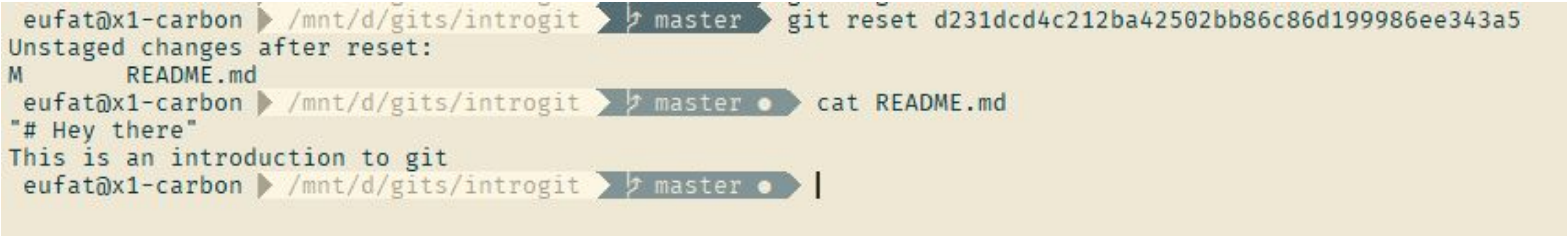
Reset provides you a way to reset to specified state. First, view our last commits



```
eufat@x1-carbon ▶ /mnt/d/gits/introgit  master  git log

commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324 (HEAD -> master)
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:05:49 2018 +0700

    A new text in README

commit d231dcd4c212ba42502bb86c86d199986ee343a5
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:01:29 2018 +0700

    Initial README
(END)
```

For example, we need to go back to this state. To do so, copy the commit id

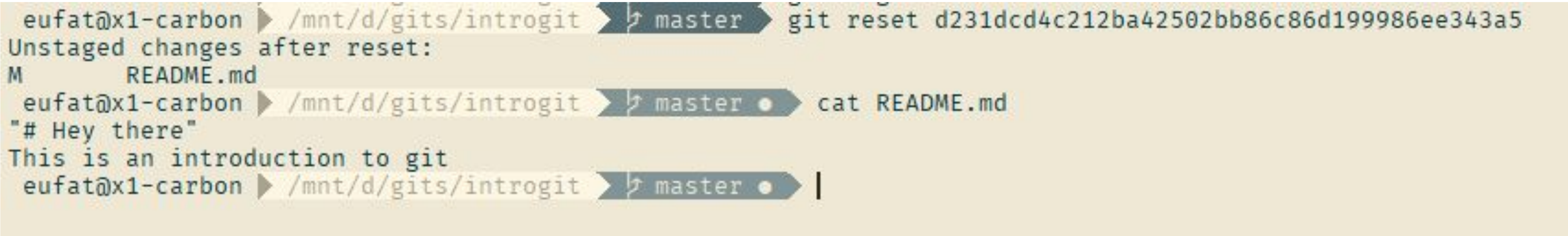
Now our current directory will go back to



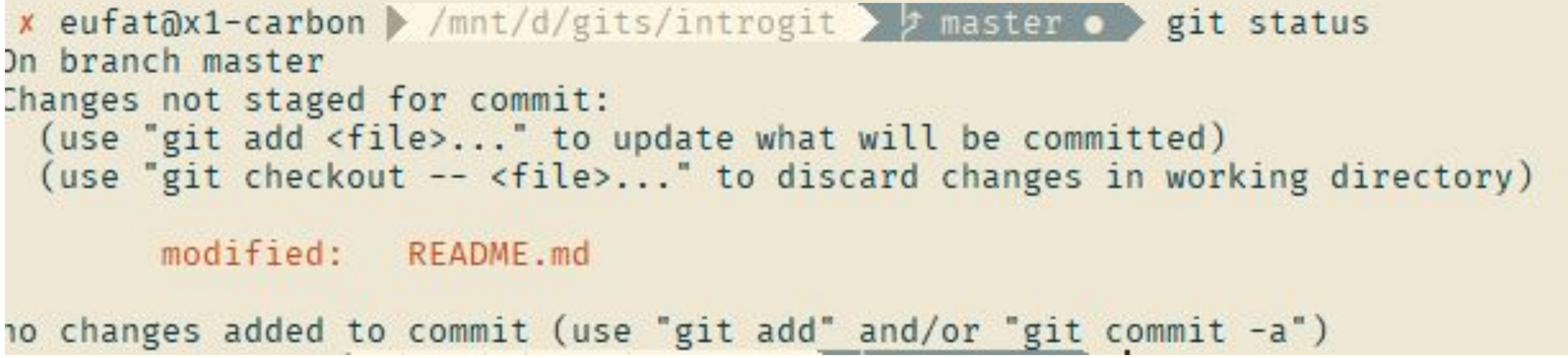
```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  master  git reset d231dcd4c212ba42502bb86c86d199986ee343a5
Unstaged changes after reset:
M       README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  master ●  cat README.md
"# Hey there"
This is an introduction to git
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  master ●  
```

Git as VCS: Reset

Now our current repo will go back to “Initial commit” state with changes on “A new text in README” added to working directory.



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ▶ git reset d231dcd4c212ba42502bb86c86d199986ee343a5
Unstaged changes after reset:
M       README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ cat README.md
"# Hey there"
This is an introduction to git
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ |
```



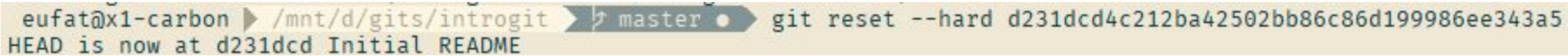
```
x eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

This is the default mode, which called --mixed mode (move changes to working directory).

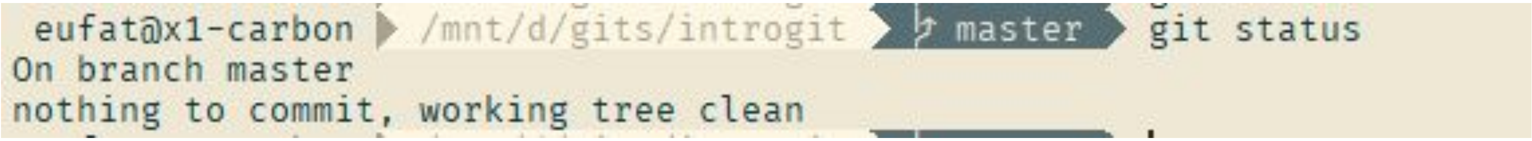
Git as VCS: Reset

If we want changes to be moved on staging area, use --soft mode.

If we want a clean reset without something added to our working directory use --hard mode.

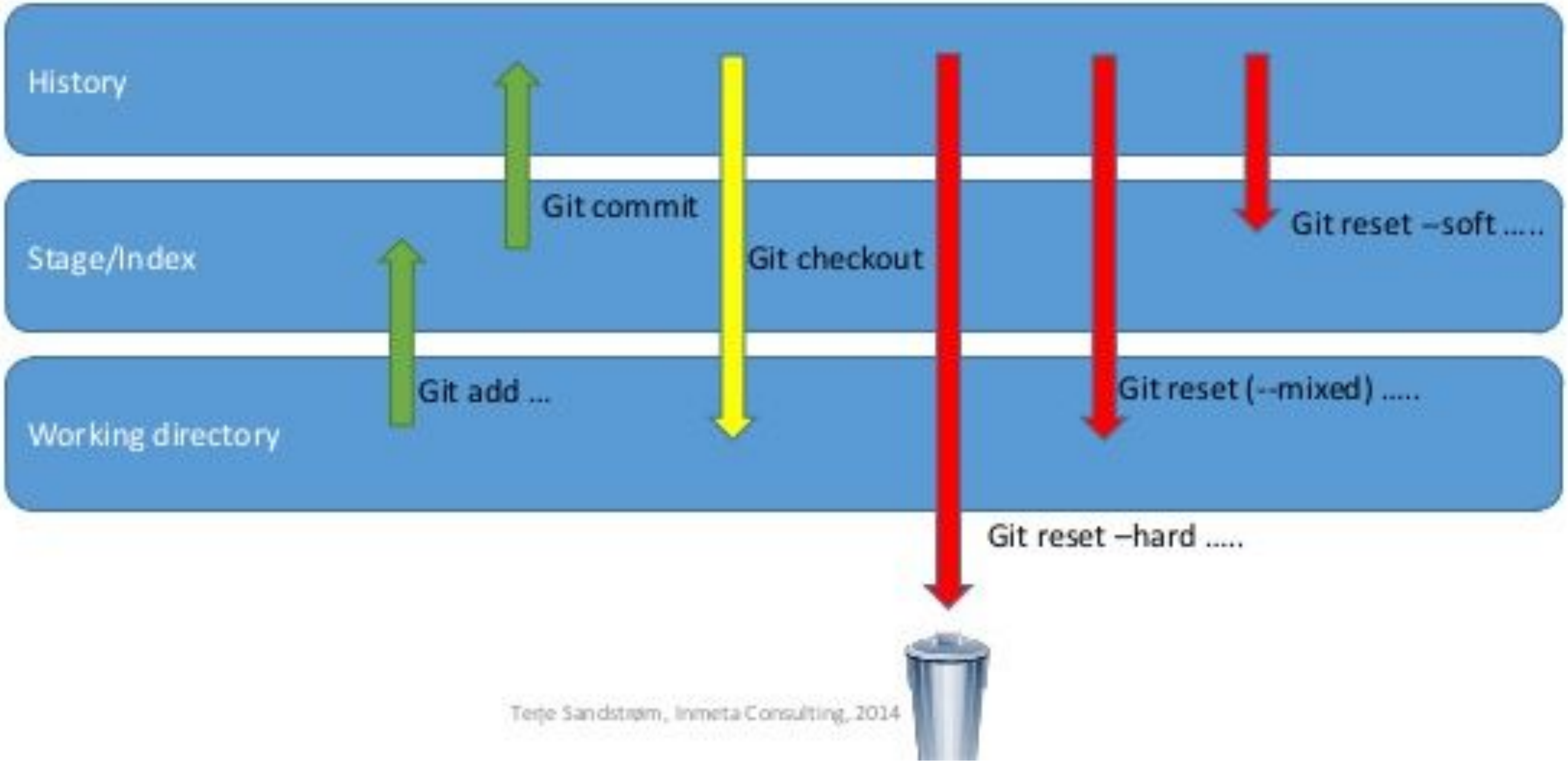


```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ git reset --hard d231dcd4c212ba42502bb86c86d199986ee343a5
HEAD is now at d231dcd Initial README
```



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ▶ git status
On branch master
nothing to commit, working tree clean
```

Git tree movements visualized



History

Stage/Index

Working directory

Git commit

Git add ...

Git checkout

Git reset –soft .....

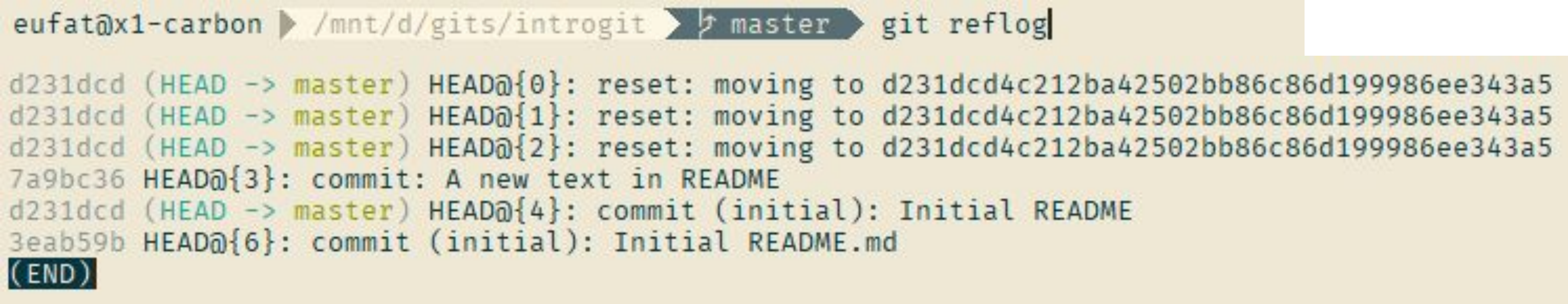
Git reset (--mixed) .....

Git reset –hard .....

Terje Sandstrøm, Inmeta Consulting, 2014

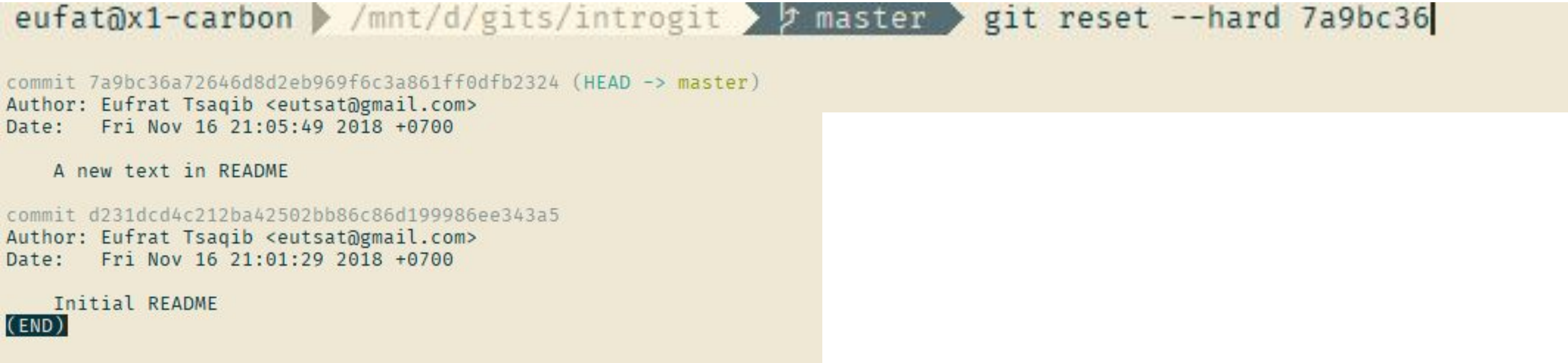
Git as VCS: Reflog

Reflog provides history what action we’ve execute. You can also reset to certain action you’ve made.



```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git reflog

d231dcd (HEAD -> master) HEAD@{0}: reset: moving to d231dcd4c212ba42502bb86c86d199986ee343a5
d231dcd (HEAD -> master) HEAD@{1}: reset: moving to d231dcd4c212ba42502bb86c86d199986ee343a5
d231dcd (HEAD -> master) HEAD@{2}: reset: moving to d231dcd4c212ba42502bb86c86d199986ee343a5
7a9bc36 HEAD@{3}: commit: A new text in README
d231dcd (HEAD -> master) HEAD@{4}: commit (initial): Initial README
3eab59b HEAD@{6}: commit (initial): Initial README.md
(END)
```

Let’s resurrect the “A new text in README” commit.



```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git reset --hard 7a9bc36

commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324 (HEAD -> master)
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:05:49 2018 +0700

    A new text in README

commit d231dcd4c212ba42502bb86c86d199986ee343a5
Author: Eufrat Tsaqib <eutsat@gmail.com>
Date:   Fri Nov 16 21:01:29 2018 +0700

    Initial README
(END)
```

Git as VCS: Summary

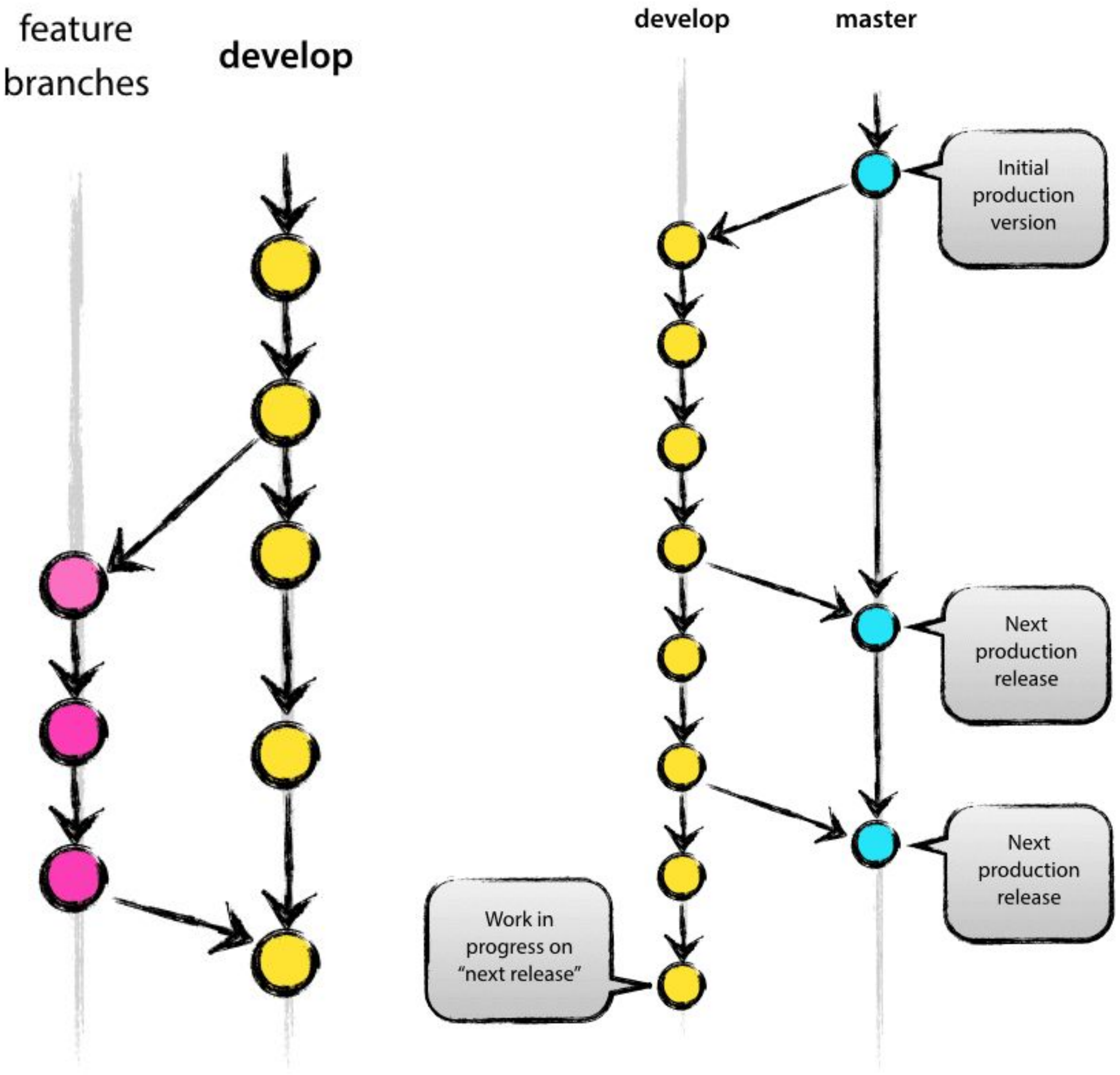
- We use *git config user.name “first last”* and *git config user.email “name@domain.com”* to set username and email
- Create a git repository using *git init*
- Add a new file named “README.md”
- Move the new file to “staging area” using *git add filename*
- Commit to repository using *git commit -m “commit message”*
- View the status of current git state using *git status*
- View the log of current branch using *git log*
- Browse the git repository using *gitk*
- See differences between two files using *git diff one two*
- HEAD is a reference of last commit on current branch
- Use *git reset --mode commitid* to reset to certain state
- View our actions using *git reflog*

Chapter 2:
Git as Collaboration Tool

Git as Collaboration Tool: Overview

- Collaboration features: Branching, Blaming, Merging, Rebasing etc.
- Remote repository options: GitHub, GitLab, BitBucket and DIY remote repo
- Share Projects: open source software, research, books etc.

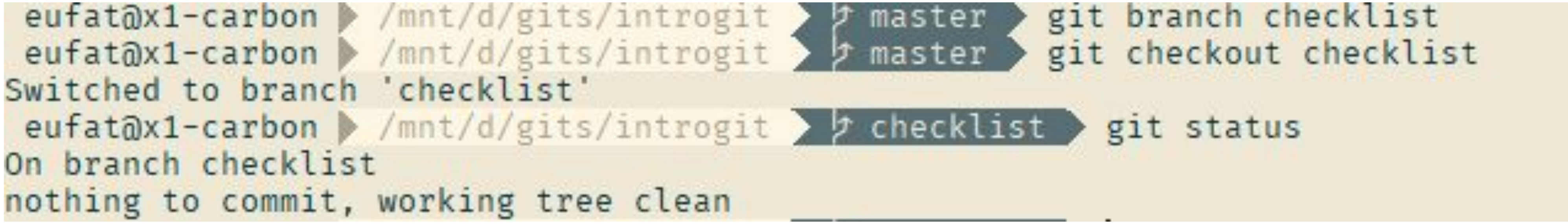
Git as Collaboration Tool: Branching



- Usually there is a main line of development of a project.
- Sometimes we want to add a feature to the project.
- Branching provide us a way to not mess around with the main line while we working the feature.

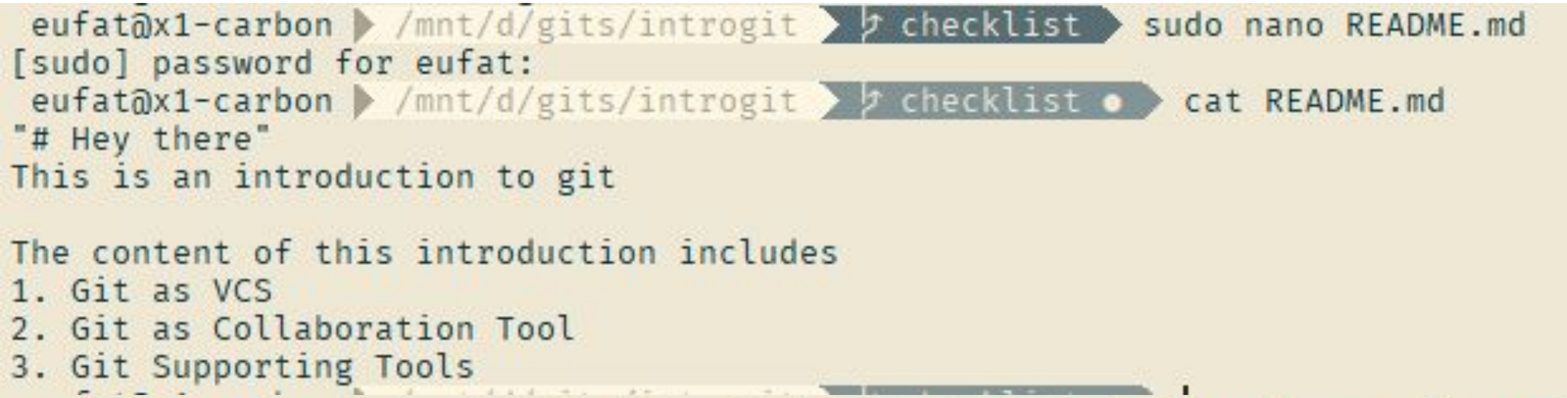
Git as Collaboration Tool: Checkout

For example, we create a new branch called “checklist”. Use branch to create new branch, use checkout to move to certain branch.

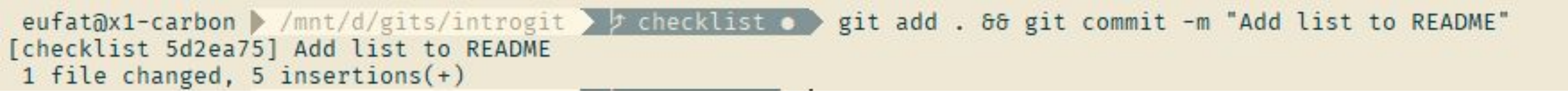


```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  master ▶ git branch checklist
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  master ▶ git checkout checklist
Switched to branch 'checklist'
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  checklist ▶ git status
On branch checklist
nothing to commit, working tree clean
```

Let’s fill that file on checklist branch with a list.



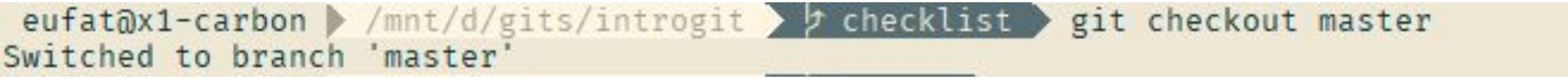
```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  checklist ▶ sudo nano README.md
[sudo] password for eufat:
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  checklist ● ▶ cat README.md
"# Hey there"
This is an introduction to git

The content of this introduction includes
1. Git as VCS
2. Git as Collaboration Tool
3. Git Supporting Tools
```



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit  checklist ● ▶ git add . && git commit -m "Add list to README"
[checklist 5d2ea75] Add list to README
 1 file changed, 5 insertions(+)
```

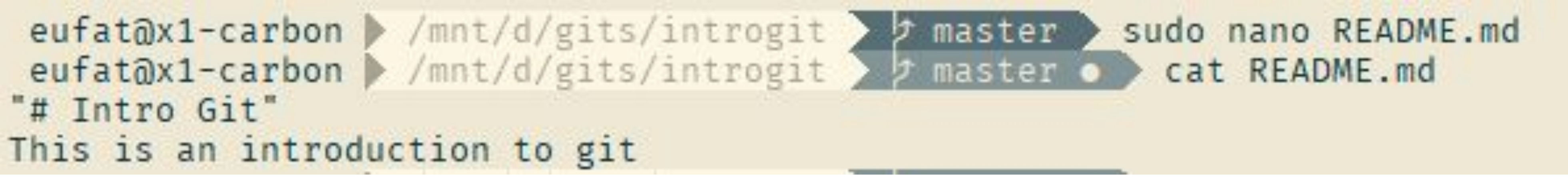
Git as Collaboration Tool: Checkout

Okay, so for the tutorial purpose let’s move back to master branch.



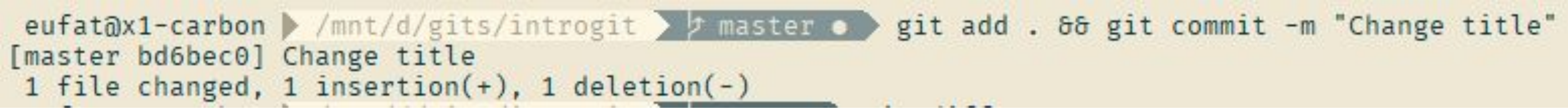
```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ checklist ▶ git checkout master
Switched to branch 'master'
```

Change the title from “ # Hey there” to “# Intro Git”.



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ▶ sudo nano README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ cat README.md
"# Intro Git"
This is an introduction to git
```

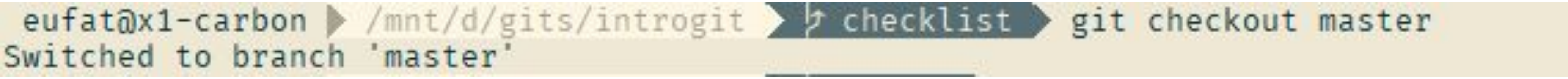
Add all file to staging and commit it.



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ git add . && git commit -m "Change title"
[master bd6bec0] Change title
 1 file changed, 1 insertion(+), 1 deletion(-)
```

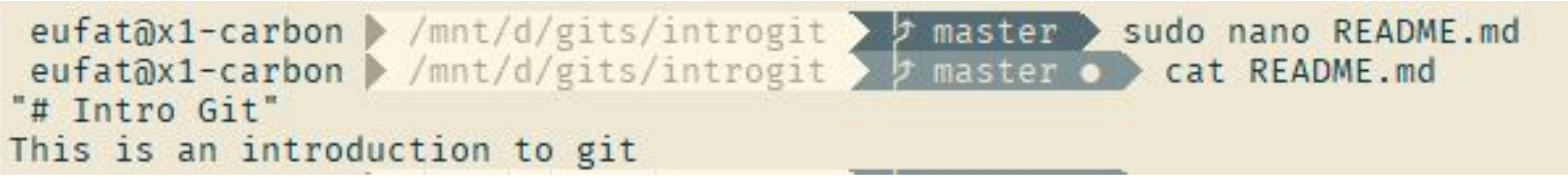
Git as Collaboration Tool: Checkout

Okay, so for the tutorial purpose let’s move back to master branch.



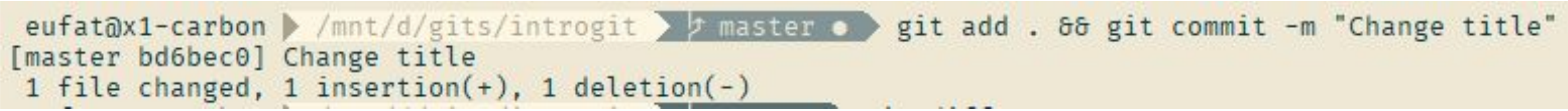
```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ checklist ▶ git checkout master
Switched to branch 'master'
```

Change the title from “ # Hey there” to “# Intro Git”.



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ▶ sudo nano README.md
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ cat README.md
"# Intro Git"
This is an introduction to git
```

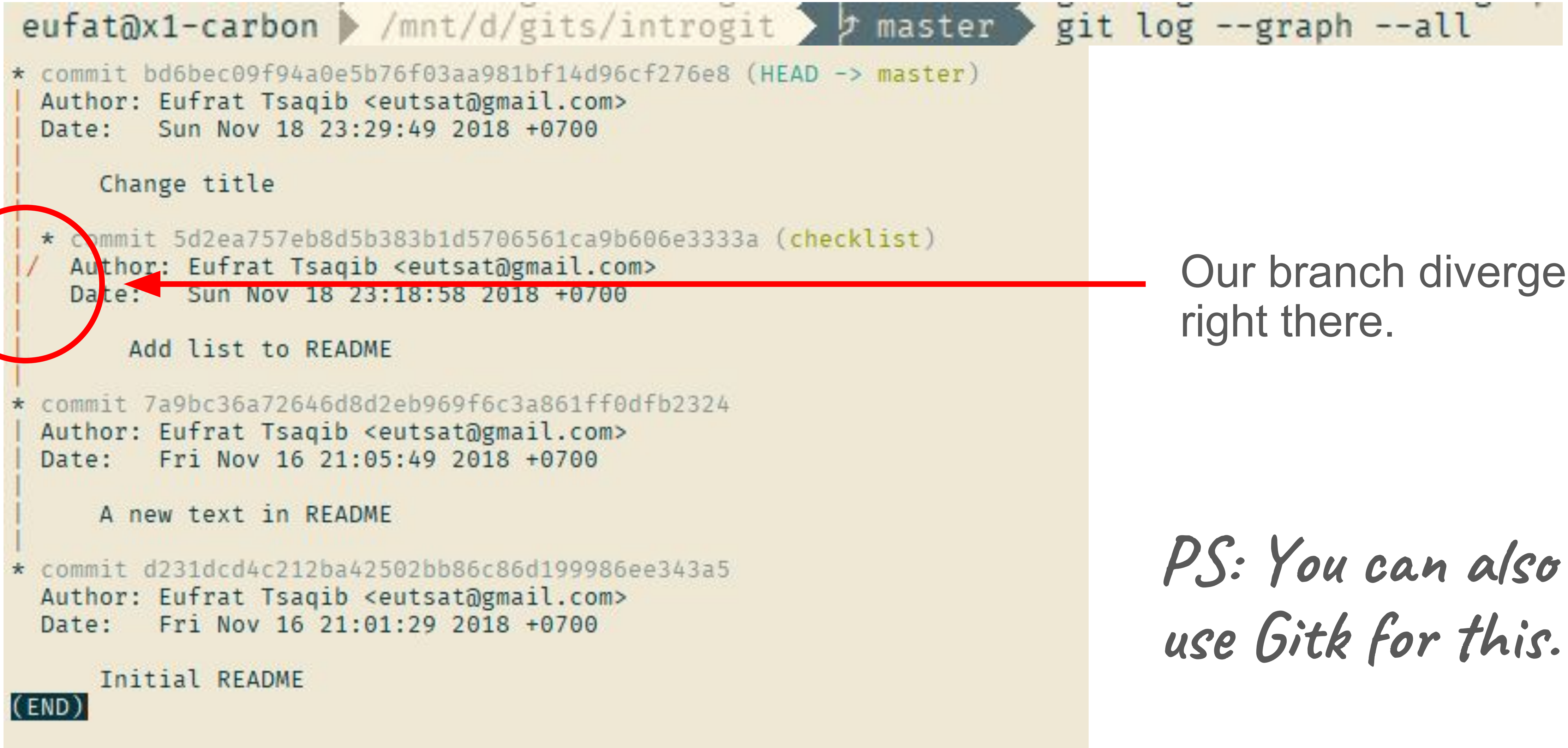
Add all file to staging and commit it.



```
 eufat@x1-carbon ▶ /mnt/d/gits/introgit ▶ master ● ▶ git add . && git commit -m "Change title"
[master bd6bec0] Change title
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Git as Collaboration Tool: Graph Log

To see our log with fancy branch line graph, just use --graph mode.



```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git log --graph --all
* commit bd6bec09f94a0e5b76f03aa981bf14d96cf276e8 (HEAD -> master)
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:29:49 2018 +0700
|
|     Change title
|
| * commit 5d2ea757eb8d5b383b1d5706561ca9b606e3333a (checklist)
|/  Author: Eufrat Tsaqib <eutsat@gmail.com>
|   Date:   Sun Nov 18 23:18:58 2018 +0700
|
|       Add list to README
|
* commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Fri Nov 16 21:05:49 2018 +0700
|
|     A new text in README
|
* commit d231dcd4c212ba42502bb86c86d199986ee343a5
  Author: Eufrat Tsaqib <eutsat@gmail.com>
  Date:   Fri Nov 16 21:01:29 2018 +0700

      Initial README
(END)
```

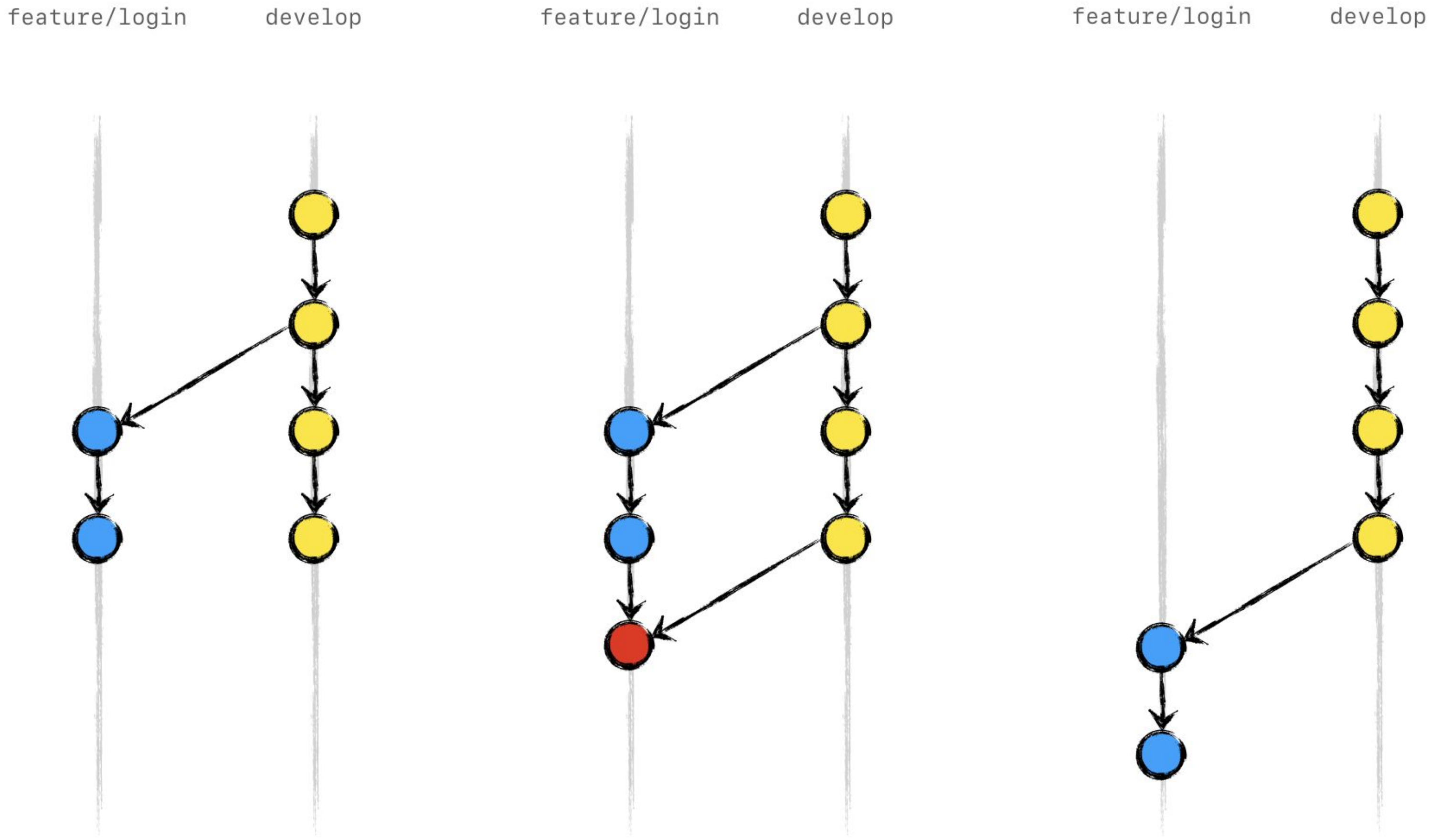
Our branch diverged right there.

PS: You can also use Gitk for this.

Git as Collaboration Tool: Merge vs Rebase

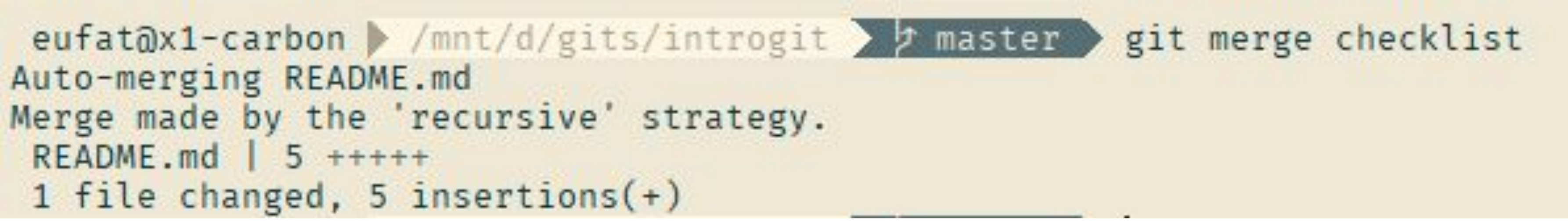
```
$ git merge develop
```

```
$ git rebase develop
```



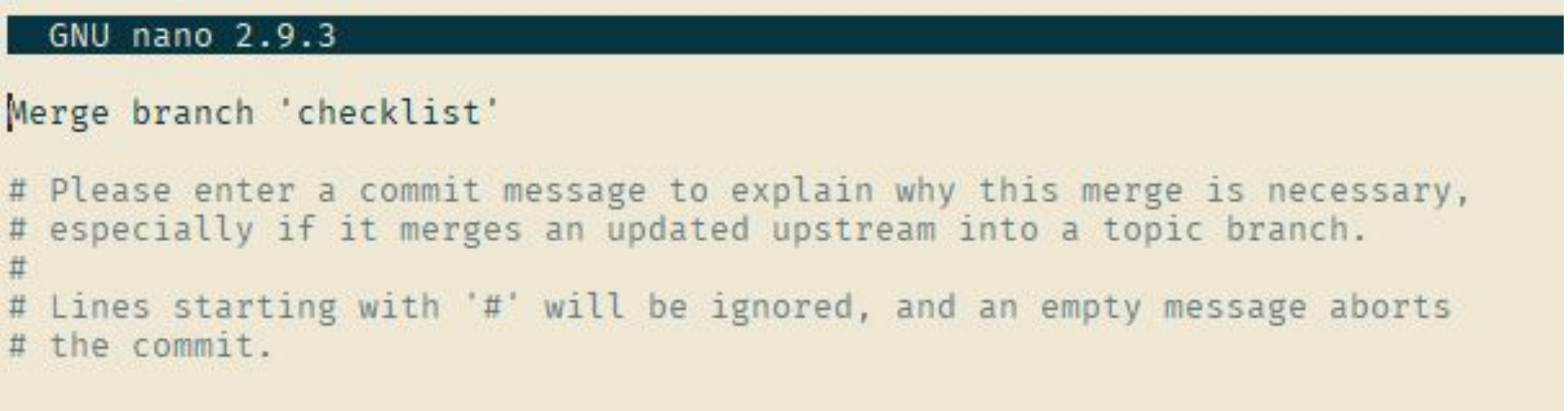
Git as Collaboration Tool: Merge

In our case, we can merge checklist branch to master using



```
 eufat@x1-carbon  /mnt/d/gits/introgit  master  git merge checklist
Auto-merging README.md
Merge made by the 'recursive' strategy.
 README.md | 5 +++++
 1 file changed, 5 insertions(+)
```

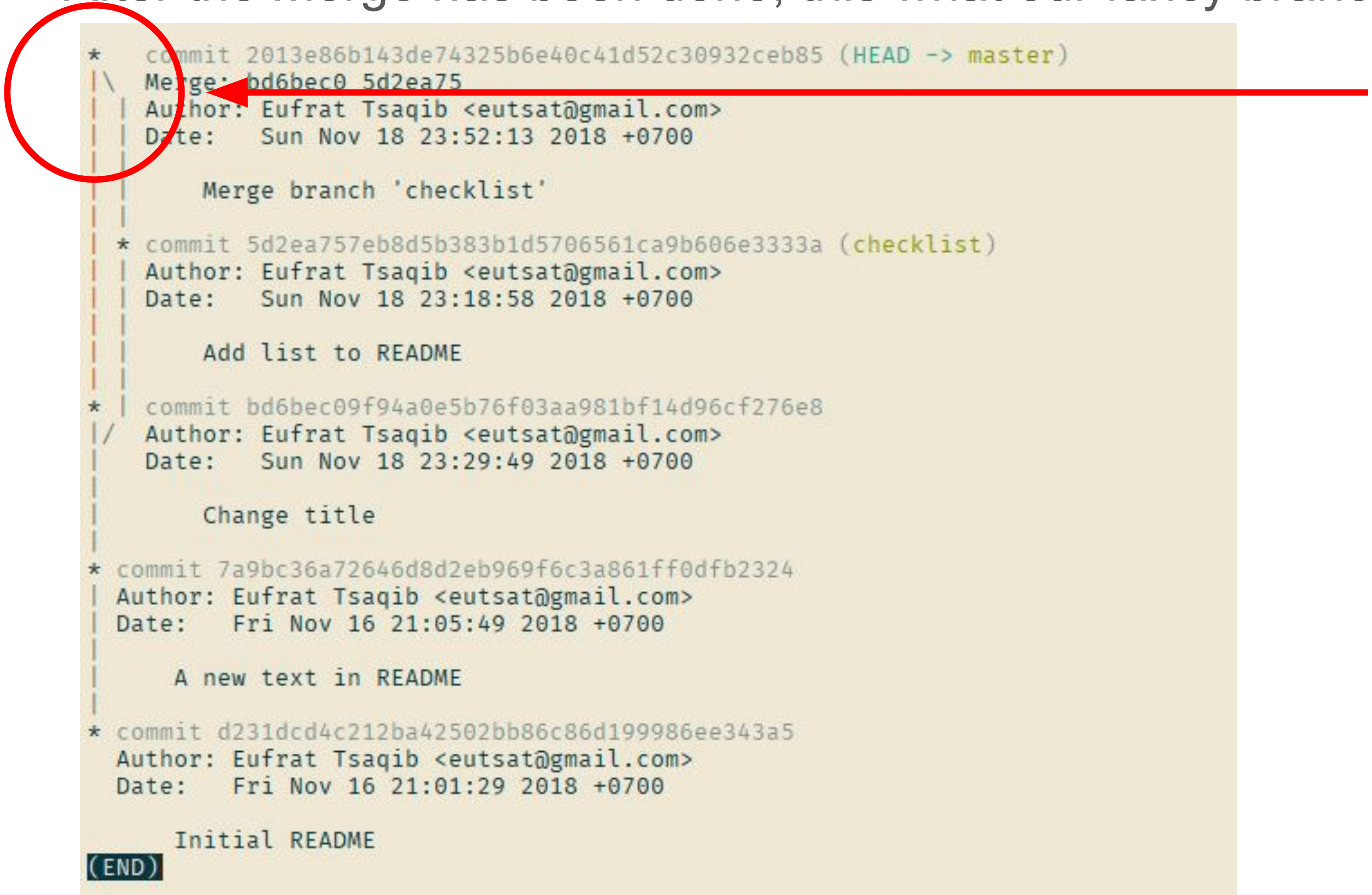
Git will ask to enter a commit message regarding the merging.



```
  GNU nano 2.9.3

Merge branch 'checklist'

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

Git as Collaboration Tool: Merge

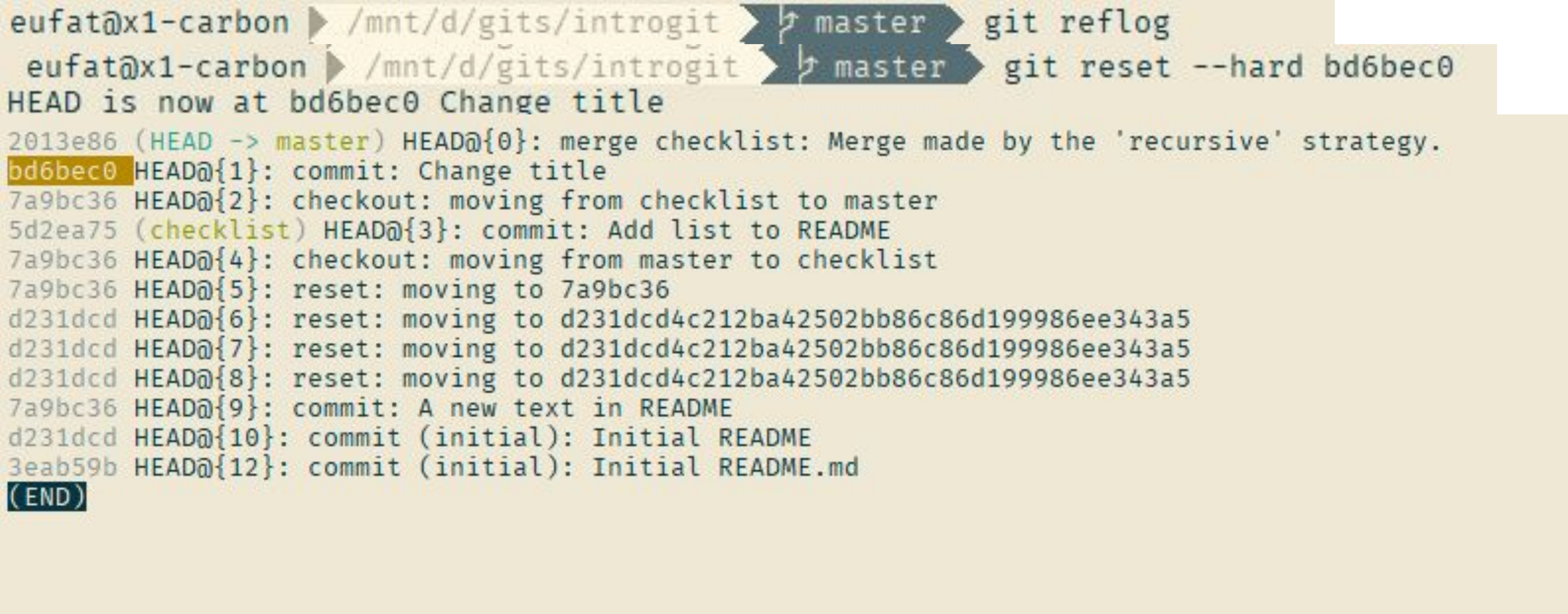
After the merge has been done, this what our fancy branch graph shows us



```
*   commit 2013e86b143de74325b6e40c41d52c30932ceb85 (HEAD -> master)
|\  Merge: bd6bec0 5d2ea75
| | Author: Eufrat Tsaqib <eutsat@gmail.com>
| | Date:   Sun Nov 18 23:52:13 2018 +0700
| |
| |     Merge branch 'checklist'
| |
| * commit 5d2ea757eb8d5b383b1d5706561ca9b606e3333a (checklist)
| | Author: Eufrat Tsaqib <eutsat@gmail.com>
| | Date:   Sun Nov 18 23:18:58 2018 +0700
| |
| |     Add list to README
| |
* | commit bd6bec09f94a0e5b76f03aa981bf14d96cf276e8
|/  Author: Eufrat Tsaqib <eutsat@gmail.com>
|   Date:   Sun Nov 18 23:29:49 2018 +0700
|
|       Change title
|
* commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Fri Nov 16 21:05:49 2018 +0700
|
|     A new text in README
|
* commit d231dcd4c212ba42502bb86c86d199986ee343a5
  Author: Eufrat Tsaqib <eutsat@gmail.com>
  Date:   Fri Nov 16 21:01:29 2018 +0700

      Initial README
(END)
```

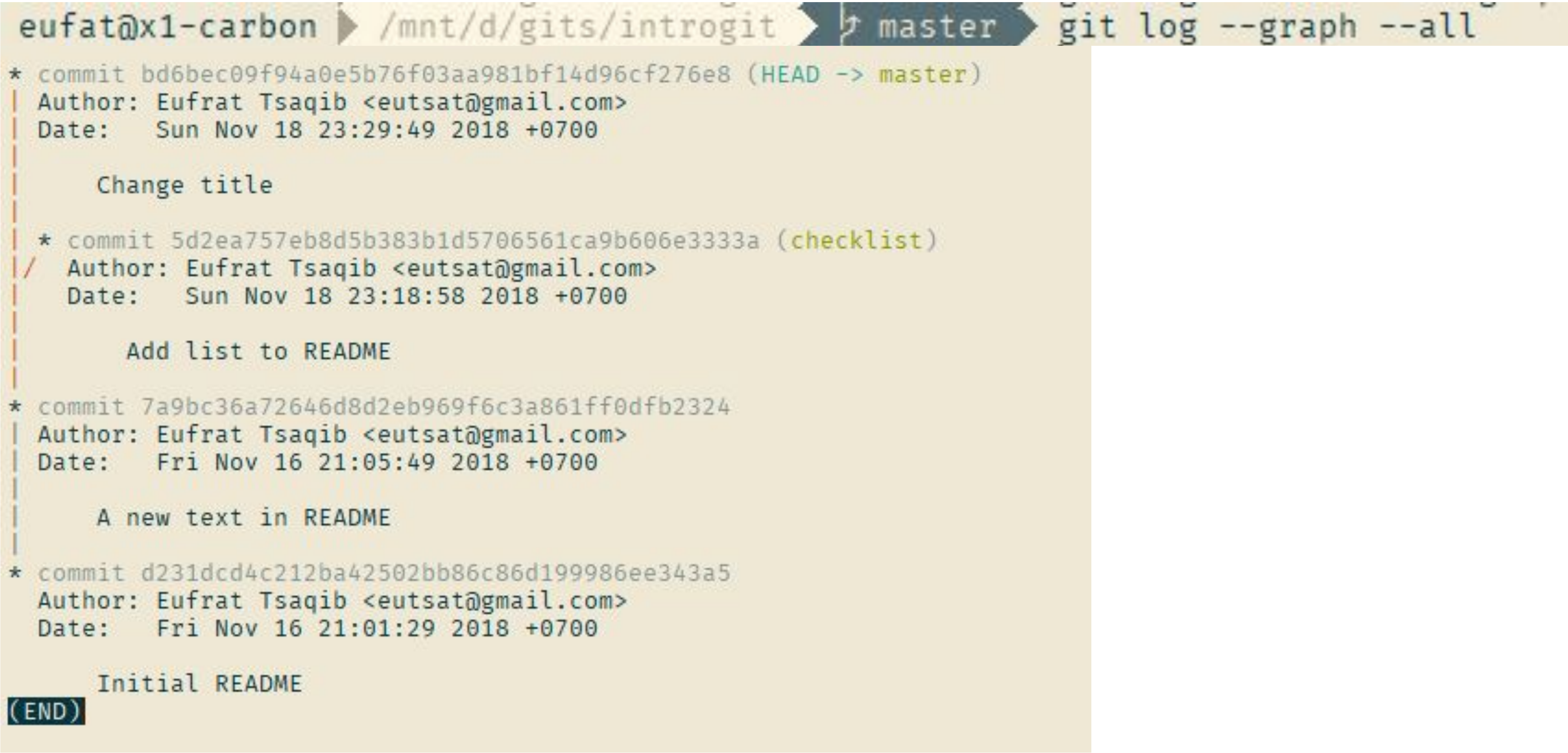
Our branch merged right there.

For tutorial purpose, please rewind our last action using *git reflog* and *git reset --hard actionid*



```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git reflog
 eufat@x1-carbon  /mnt/d/gits/introgit  master  git reset --hard bd6bec0
HEAD is now at bd6bec0 Change title
2013e86 (HEAD -> master) HEAD@{0}: merge checklist: Merge made by the 'recursive' strategy.
bd6bec0 HEAD@{1}: commit: Change title
7a9bc36 HEAD@{2}: checkout: moving from checklist to master
5d2ea75 (checklist) HEAD@{3}: commit: Add list to README
7a9bc36 HEAD@{4}: checkout: moving from master to checklist
7a9bc36 HEAD@{5}: reset: moving to 7a9bc36
d231dcd HEAD@{6}: reset: moving to d231dcd4c212ba42502bb86c86d199986ee343a5
d231dcd HEAD@{7}: reset: moving to d231dcd4c212ba42502bb86c86d199986ee343a5
d231dcd HEAD@{8}: reset: moving to d231dcd4c212ba42502bb86c86d199986ee343a5
7a9bc36 HEAD@{9}: commit: A new text in README
d231dcd HEAD@{10}: commit (initial): Initial README
3eab59b HEAD@{12}: commit (initial): Initial README.md
(END)
```

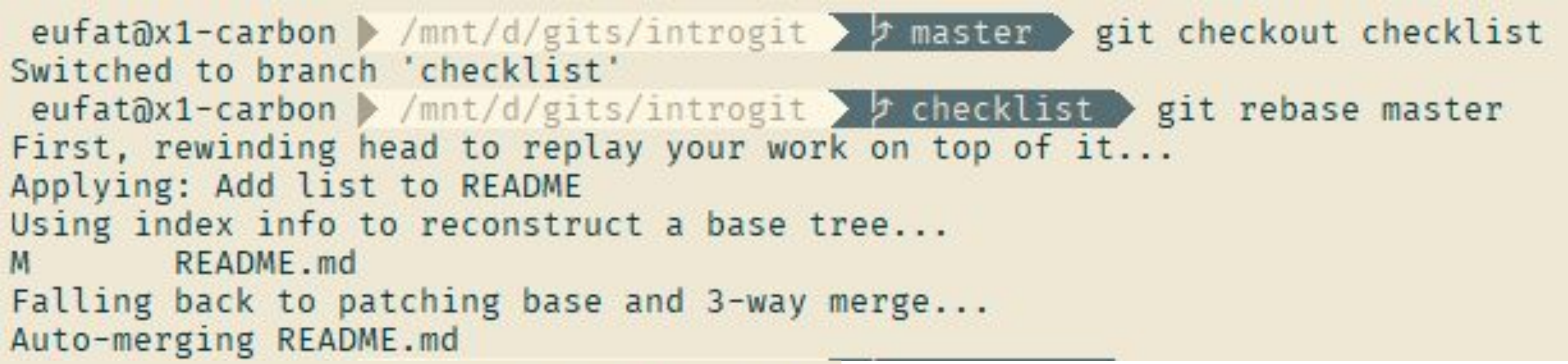
The should be reverted and look like this now.



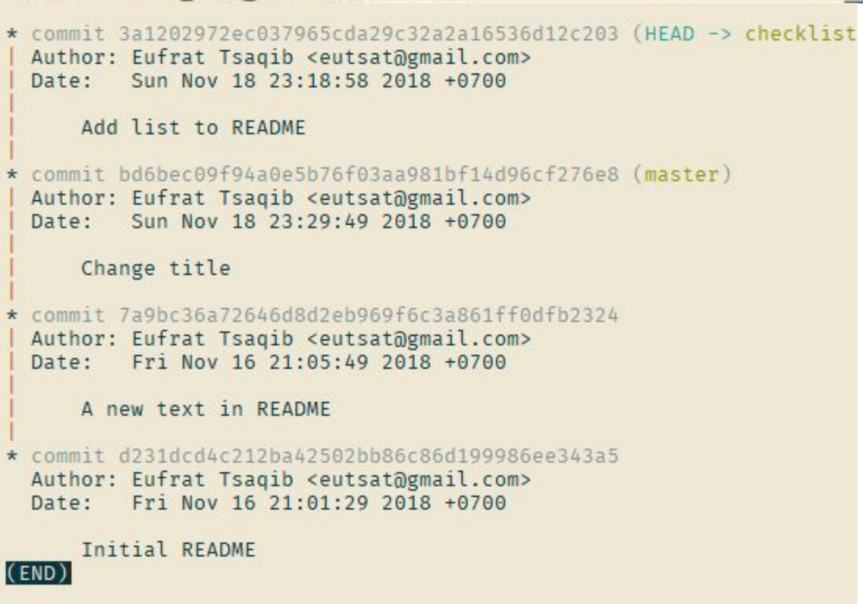
```
eufat@x1-carbon  /mnt/d/gits/introgit  master  git log --graph --all
* commit bd6bec09f94a0e5b76f03aa981bf14d96cf276e8 (HEAD -> master)
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:29:49 2018 +0700
|
|     Change title
|
| * commit 5d2ea757eb8d5b383b1d5706561ca9b606e3333a (checklist)
|/  Author: Eufrat Tsaqib <eutsat@gmail.com>
|   Date:   Sun Nov 18 23:18:58 2018 +0700
|
|       Add list to README
|
* commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Fri Nov 16 21:05:49 2018 +0700
|
|     A new text in README
|
* commit d231dcd4c212ba42502bb86c86d199986ee343a5
  Author: Eufrat Tsaqib <eutsat@gmail.com>
  Date:   Fri Nov 16 21:01:29 2018 +0700

      Initial README
(END)
```

Git as Collaboration Tool: Rebase

With rebase, we place the new commit in targeted branch on the tail of current branch.



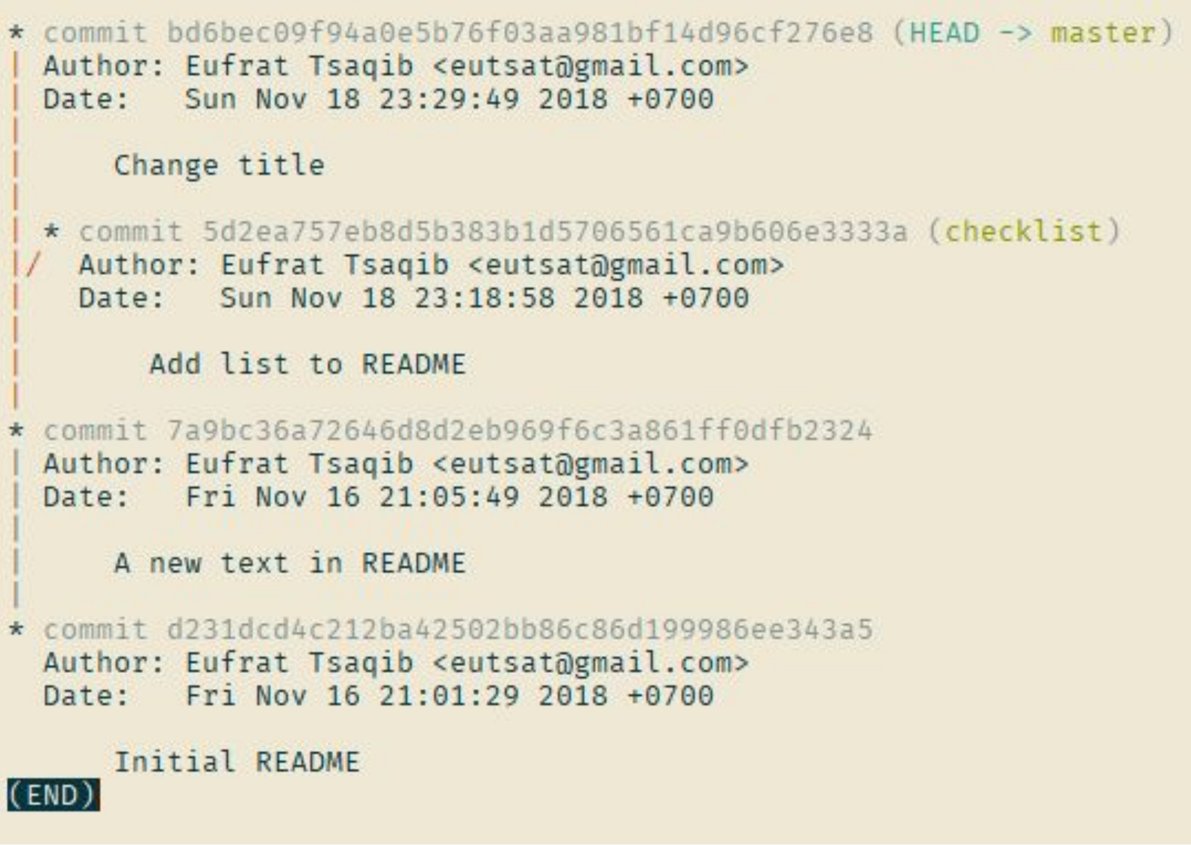
```
 eufat@x1-carbon  /mnt/d/gits/introgit  master  git checkout checklist
Switched to branch 'checklist'
 eufat@x1-carbon  /mnt/d/gits/introgit  checklist  git rebase master
First, rewinding head to replay your work on top of it...
Applying: Add list to README
Using index info to reconstruct a base tree...
M       README.md
Falling back to patching base and 3-way merge...
Auto-merging README.md
```



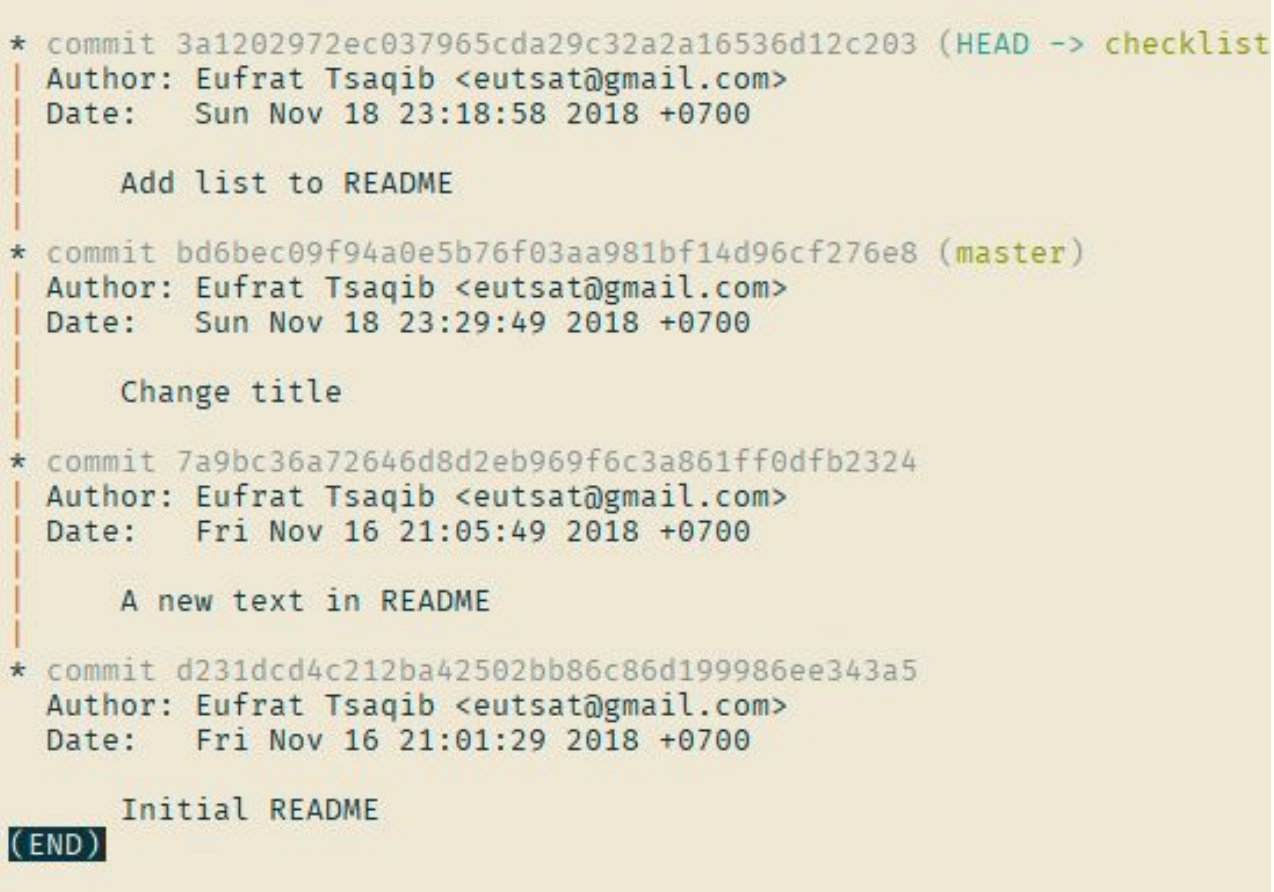
```
* commit 3a1202972ec037965cda29c32a2a16536d12c203 (HEAD -> checklist
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:18:58 2018 +0700
|
|     Add list to README
|
* commit bd6bec09f94a0e5b76f03aa981bf14d96cf276e8 (master)
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:29:49 2018 +0700
|
|     Change title
|
* commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Fri Nov 16 21:05:49 2018 +0700
|
|     A new text in README
|
* commit d231dcd4c212ba42502bb86c86d199986ee343a5
  Author: Eufrat Tsaqib <eutsat@gmail.com>
  Date:   Fri Nov 16 21:01:29 2018 +0700

      Initial README
(END)
```

Git as Collaboration Tool: Rebase

Using our handy graph log, the new checklist branch transform from left graph to the right graph.



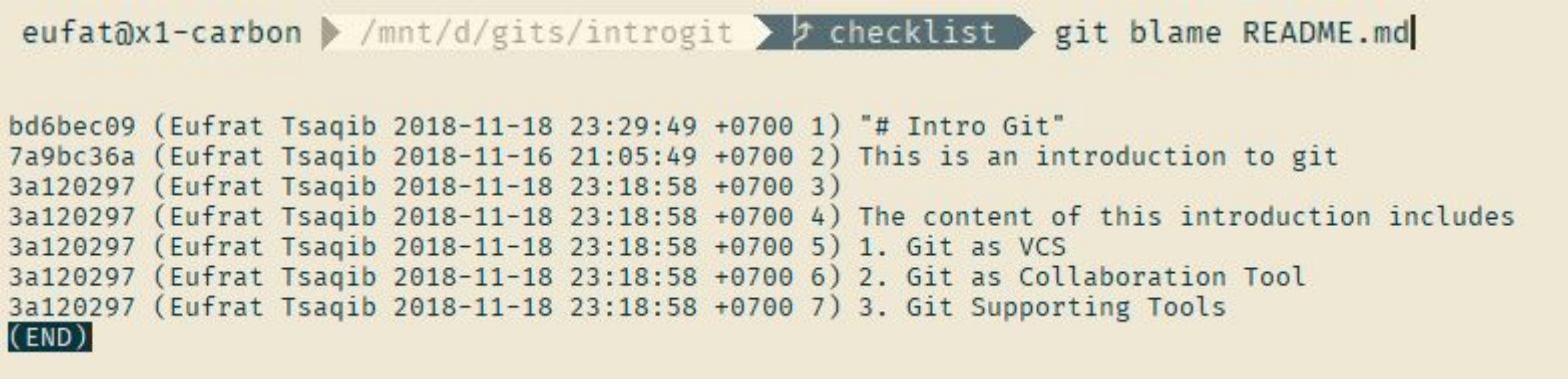
```
* commit bd6bec09f94a0e5b76f03aa981bf14d96cf276e8 (HEAD -> master)
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:29:49 2018 +0700
|
|     Change title
|
| * commit 5d2ea757eb8d5b383b1d5706561ca9b606e3333a (checklist)
|/  Author: Eufrat Tsaqib <eutsat@gmail.com>
|   Date:   Sun Nov 18 23:18:58 2018 +0700
|
|       Add list to README
|
* commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Fri Nov 16 21:05:49 2018 +0700
|
|     A new text in README
|
* commit d231dcd4c212ba42502bb86c86d199986ee343a5
  Author: Eufrat Tsaqib <eutsat@gmail.com>
  Date:   Fri Nov 16 21:01:29 2018 +0700

      Initial README
(END)
```



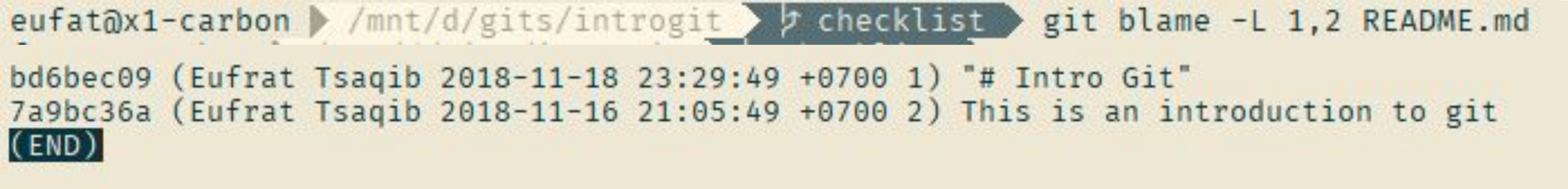
```
* commit 3a1202972ec037965cda29c32a2a16536d12c203 (HEAD -> checklist
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:18:58 2018 +0700
|
|     Add list to README
|
* commit bd6bec09f94a0e5b76f03aa981bf14d96cf276e8 (master)
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Sun Nov 18 23:29:49 2018 +0700
|
|     Change title
|
* commit 7a9bc36a72646d8d2eb969f6c3a861ff0dfb2324
| Author: Eufrat Tsaqib <eutsat@gmail.com>
| Date:   Fri Nov 16 21:05:49 2018 +0700
|
|     A new text in README
|
* commit d231dcd4c212ba42502bb86c86d199986ee343a5
  Author: Eufrat Tsaqib <eutsat@gmail.com>
  Date:   Fri Nov 16 21:01:29 2018 +0700

      Initial README
(END)
```

Git as Collaboration Tool: Blaming

Blame help you determine who made changes to a file or in a certain line.

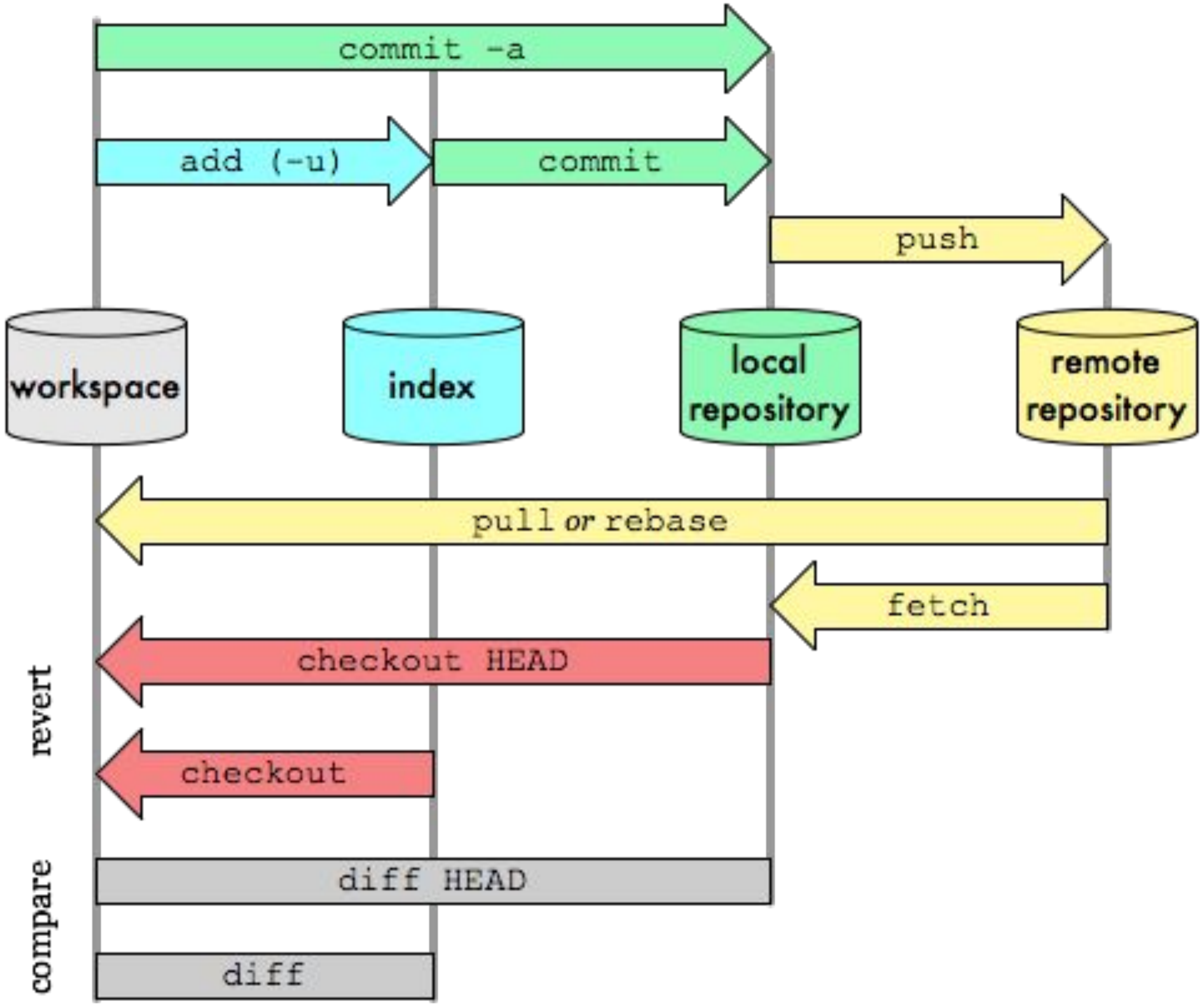


```
eufat@x1-carbon  /mnt/d/gits/introgit  checklist  git blame README.md

bd6bec09 (Eufrat Tsaqib 2018-11-18 23:29:49 +0700 1) "# Intro Git"
7a9bc36a (Eufrat Tsaqib 2018-11-16 21:05:49 +0700 2) This is an introduction to git
3a120297 (Eufrat Tsaqib 2018-11-18 23:18:58 +0700 3)
3a120297 (Eufrat Tsaqib 2018-11-18 23:18:58 +0700 4) The content of this introduction includes
3a120297 (Eufrat Tsaqib 2018-11-18 23:18:58 +0700 5) 1. Git as VCS
3a120297 (Eufrat Tsaqib 2018-11-18 23:18:58 +0700 6) 2. Git as Collaboration Tool
3a120297 (Eufrat Tsaqib 2018-11-18 23:18:58 +0700 7) 3. Git Supporting Tools
(END)
```



```
eufat@x1-carbon  /mnt/d/gits/introgit  checklist  git blame -L 1,2 README.md
bd6bec09 (Eufrat Tsaqib 2018-11-18 23:29:49 +0700 1) "# Intro Git"
7a9bc36a (Eufrat Tsaqib 2018-11-16 21:05:49 +0700 2) This is an introduction to git
(END)
```

Git as Collaboration Tool: Remote Repository



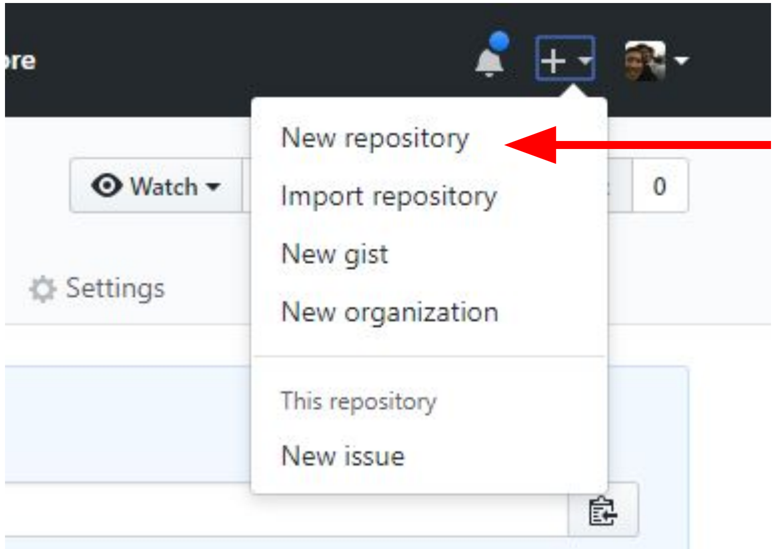
Remote Repository:

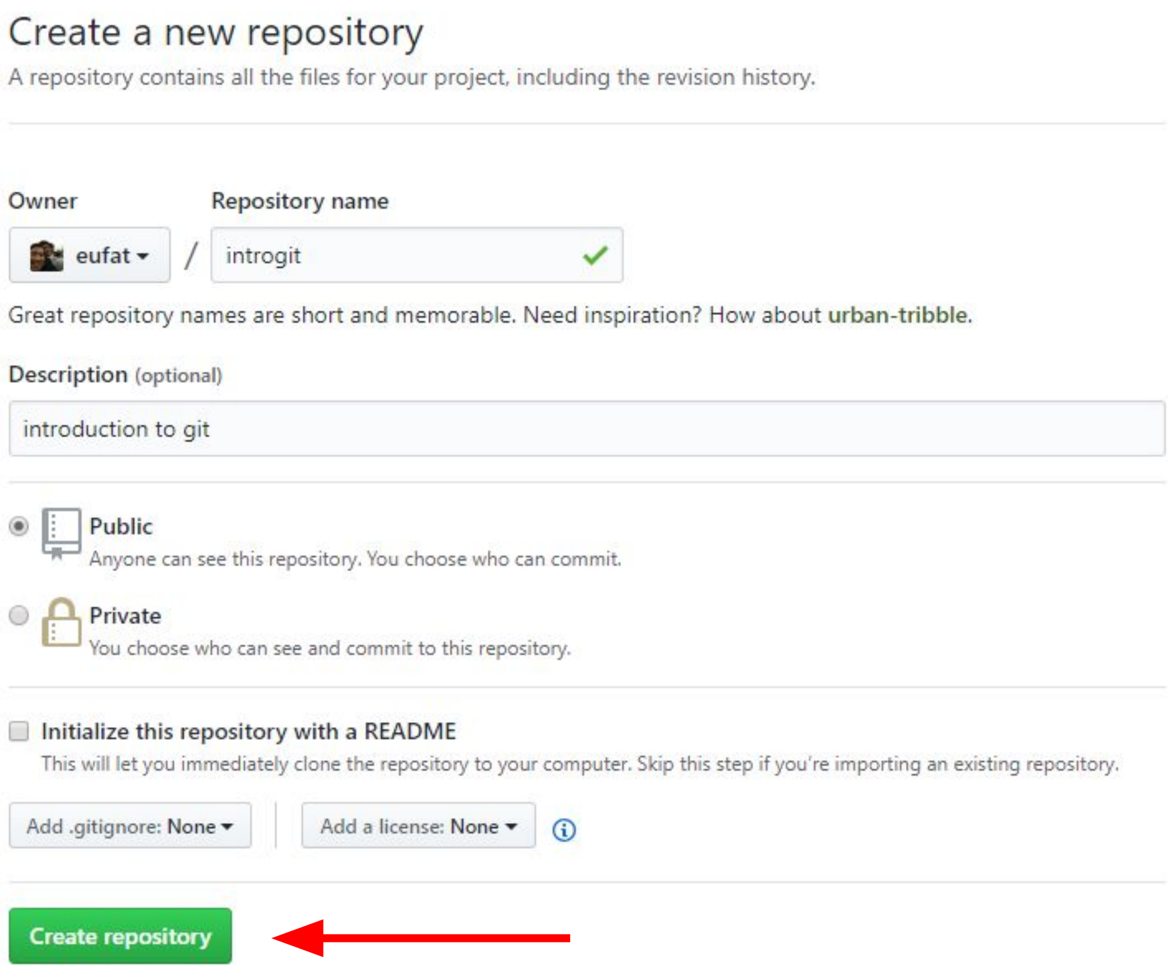
- GitHub: Great for open source, paid for private repo.
- GitLab: GitHub alternative, DIY Gitlab.
- BitBucket: Atlassian made, free private repo.

https://education.github.com/students

Git as Collaboration Tool: Remote Repository

Sign in or sign up to GitHub, create new repo.





Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: eufat / Repository name: introgit ✓

Great repository names are short and memorable. Need inspiration? How about **urban-tribble**.

Description (optional): introduction to git

● Public
Anyone can see this repository. You choose who can commit.

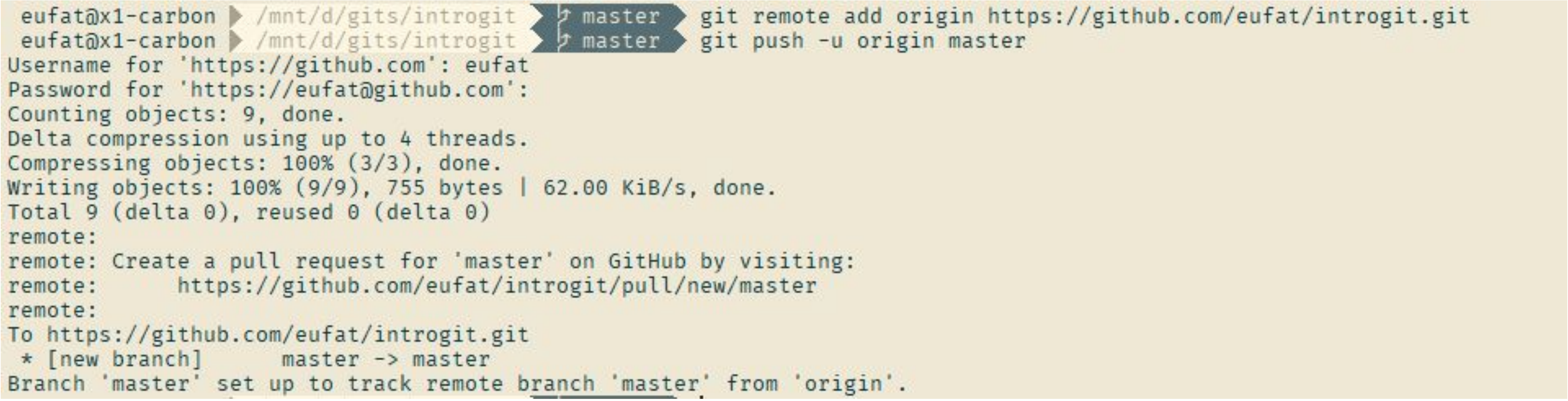
○ Private
You choose who can see and commit to this repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾ | Add a license: None ▾

Git as Collaboration Tool: Remote Repository

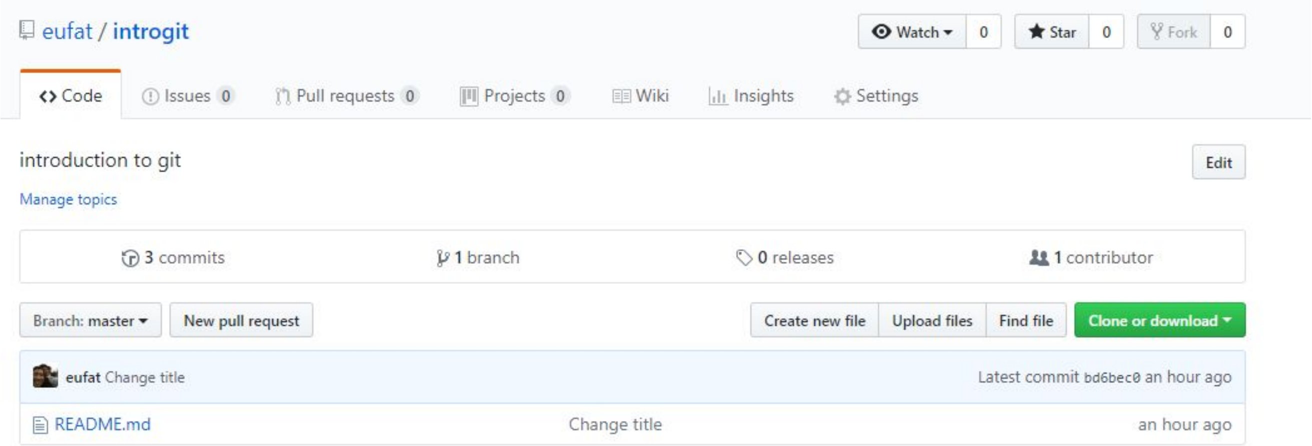
Let’s add remote origin and push our local repo to remote repo.



```
 eufat@x1-carbon  /mnt/d/gits/introgit   master  git remote add origin https://github.com/eufat/introgit.git
 eufat@x1-carbon  /mnt/d/gits/introgit   master  git push -u origin master
Username for 'https://github.com': eufat
Password for 'https://eufat@github.com':
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (9/9), 755 bytes | 62.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/eufat/introgit/pull/new/master
remote:
To https://github.com/eufat/introgit.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Git as Collaboration Tool: Remote Repository

Voilà, we just created our remote repo.

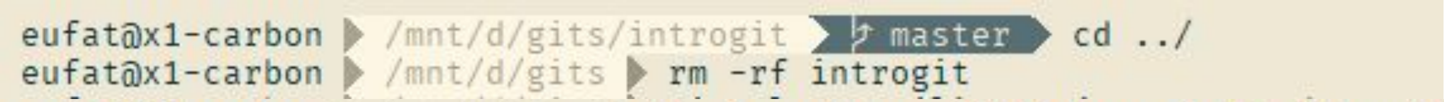


introduction to git

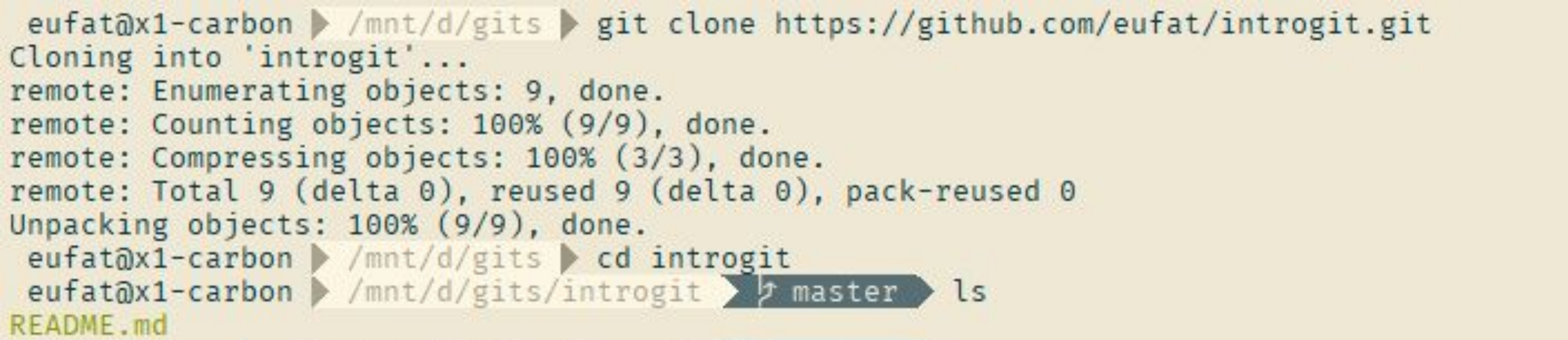
eufat Change title — Latest commit bd6bec0 an hour ago

README.md — Change title — an hour ago

Now, we can delete our local repo and clone our remote repo safely.



```
eufat@x1-carbon  /mnt/d/gits/introgit   master  cd ../
eufat@x1-carbon  /mnt/d/gits  rm -rf introgit
```



```
 eufat@x1-carbon  /mnt/d/gits  git clone https://github.com/eufat/introgit.git
Cloning into 'introgit'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
 eufat@x1-carbon  /mnt/d/gits  cd introgit
 eufat@x1-carbon  /mnt/d/gits/introgit   master  ls
README.md
```

Chapter 3:
Git Supporting Tools

Git Supporting Tools

Interfacing and other tools

- GITK: Included git repository browser (https://git-scm.com/docs/gitk)
- Sublime Merge: All in one Git tools with user interface (https://www.sublimemerge.com/)
- ZSH: shell terminal with git plugin (https://ohmyz.sh/)
- Visual Studio Code: Great editor with out-of-the-box git support (https://code.visualstudio.com/)
- GitLens: Great git plugin for Visual Studio Code (https://gitlens.amod.io/)

Chapter N:
Your Next Chapter

Your Next Chapter

- Use git in your thesis or other project
- To mastering git, read every bit of the documentation try each individually and finally implement to your own project (https://git-scm.com/doc)
- Use supporting tools to help you cope with git complexities
- Contribute to open source projects on github, gitlab etc
- Teach others

This tutorial repo is on:

github.com/eufat/introgit