Arūnas Butkus EKSfmu-16

Practical Task 2

What I did:

Part 1: “Repository creation, adding files into repository”

The 7 steps here are not as clear as having 7 steps would imply, but from what I gather at this point I should have two repos, which are the same, “server_side” is the origin one where the stuff comes from and goes to; and “client_side” – simulating some user getting the original repo. I start of by following YouTube tutorial of setting up git SSH server and client (https://youtu.be/lXSZUuDW4nY)(Fig 1)(What is shown in figure was wrong, missing a line). Sometime later: networking is hard! Even following step by step, something breaks aaaand now I don’t know what to do anymore. Give up for now…

Attempt number two. Trying out git options available on Windows. Checked out “GitHub Desktop” - simple install with a simple new repo creation (Fig. 2). Committing is simple and obvious (Fig. 3). To emulate the harder way of using git with bash – installed Git-2.14.1-64-bit – git for windows command line. Installation is more involved but has plenty of explanations in it (Fig 4-9). Now having CMD access to git I check configuration file and I see a bunch of “GitHub Desktop" configurations with ones I added through CMD (user and email), as I did with bash, at the end (Fig. 10).

So considering only the part description I’ve done it in multiple ways already, but the steps also asks to connect two git users to one repo and commit files from the client one. Then now I will attempt to connect windows user to the repo in VM. Or vice versa.

Hour, or few, later I’m back! And it works. I won’t explain what is already explained in the before mentioned YouTube video as I once again followed that one. The issue was I did not notice that that interface file based static IP setting actually prevented internet access, even though it allowed to see all computers connected to the router. This caused issues getting openssh-server and git client on new VM. Circumvented that issue by setting static IP for the gitServer VM in the router configuration. By the way, now I’m running two VMs: from laboratory work 1 as a client and, downscaled in terms of hardware use, “gitServer” VM. Proof of reaching the end of that tutorial is in Figure 11.

Thus client can now push and pull data from gitServer. Wonder if ”GitHub Desktop” can clone that repo (Fig 12). And the answer is no due to authentication issues that I have no clue how to solve (Fig 12). First idea was, since I added SSH authentication, to use PuTTY to open SSH connection to get to the gitServer. Generated an SSH key with PuTTY key generator (Fig. 13), authorized that key as git user in the gitServer, allowed git users to use bash shell, and voilà (Fig. 14). But did not know what to do with that…

So then I noticed that installing git command line software added git bash to the right click menu (Fig. 15) (already forgetting what I did in Figure 4). Using half-baked knowledge gathered till now, went and made a folder where I’d like to store my gitServer repo. Opened git bash in it, executed `git init --bare` which should’ve given an empty git repo, but instead made a mess of folders and files (Fig. 16). So then proceeded to try pulling remote repo. Don’t quite remember at which point, but eventually it worked; after adding remote origin to git settings (Fig. 17) and when I figured out what parameters `git pull` requires. Did some cleanup with .gitignore file on Windows side repo and eventually got some decent repo on Windows side too (Fig. 18-20).

Just to finally cement the git command line/bash way of data sharing, made a change to the `text.txt` file in Windows (Fig. 21), committed and pushed it to the gitServer (Fig. 22), and pulled it on Ubuntu client (Fig. 23). Committing being the hardest part since I did not specify `-m` parameter with a message which launched VI editor which then required me to go through three more YouTube tutorials where the least Hindu person of those 3 finally told me how to save and close that editor… `ESC -> Shift+zz` (facepalm)

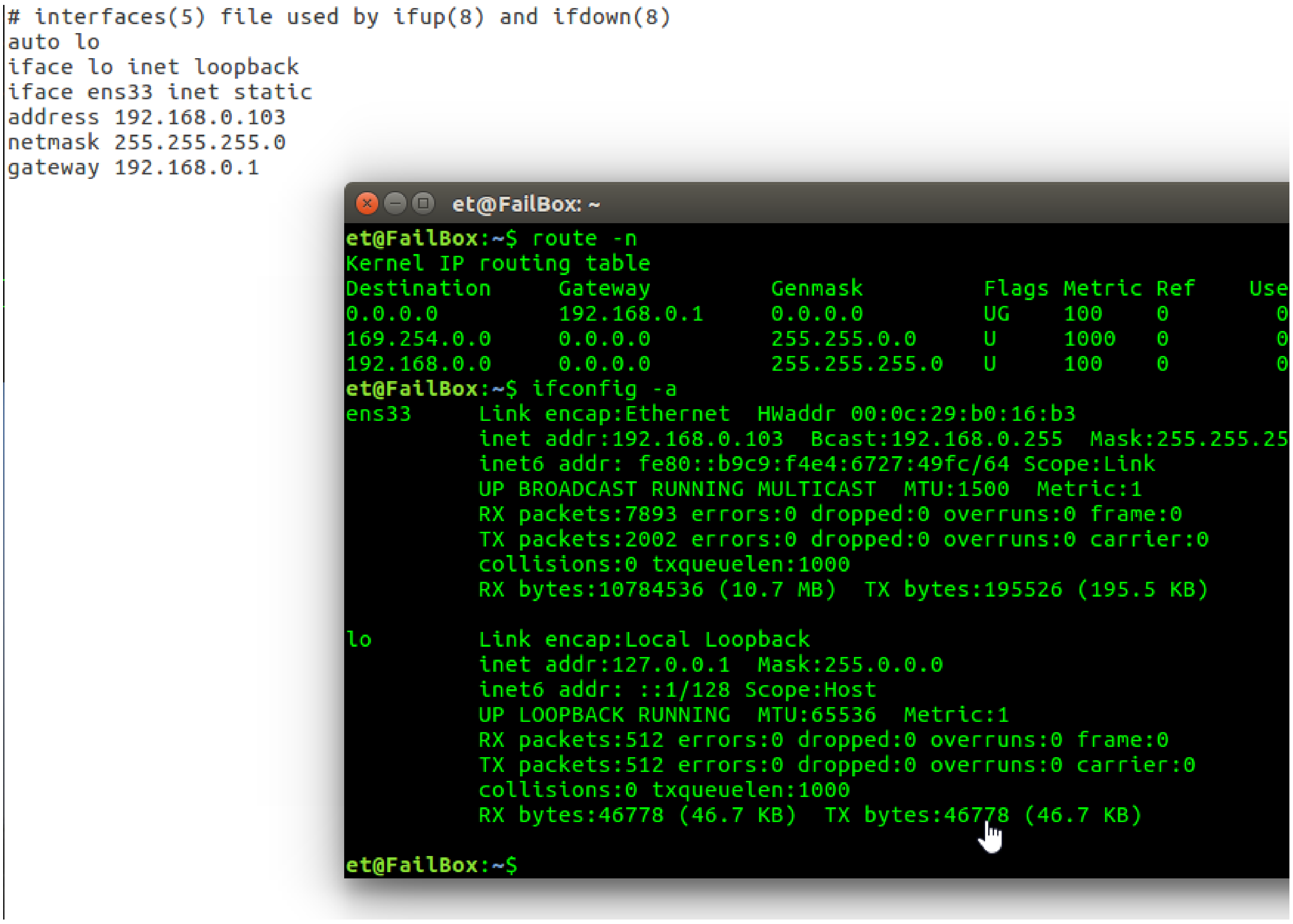
Part 2: “Getting files from repository, making changes”

I did it for the most part already, but I’ll do it again with the provided text then (Fig. 24). Mostly.

Part 3: “Conflict resolving”

First I pulled the change from previous part (Fig. 25). Then I edited and pushed the changes similar to as instructed (Fig. 26). And then I did as instructed on the Windows side (Fig. 27), which ended up with prevented `push` due to files in repo being newer than local `pull`. So conflict now appears when trying git pull. (Fig. 28). Resolved the conflict by removing the auto-generated lines 3, 5 and 7 and re-pushing (Fig. 29).

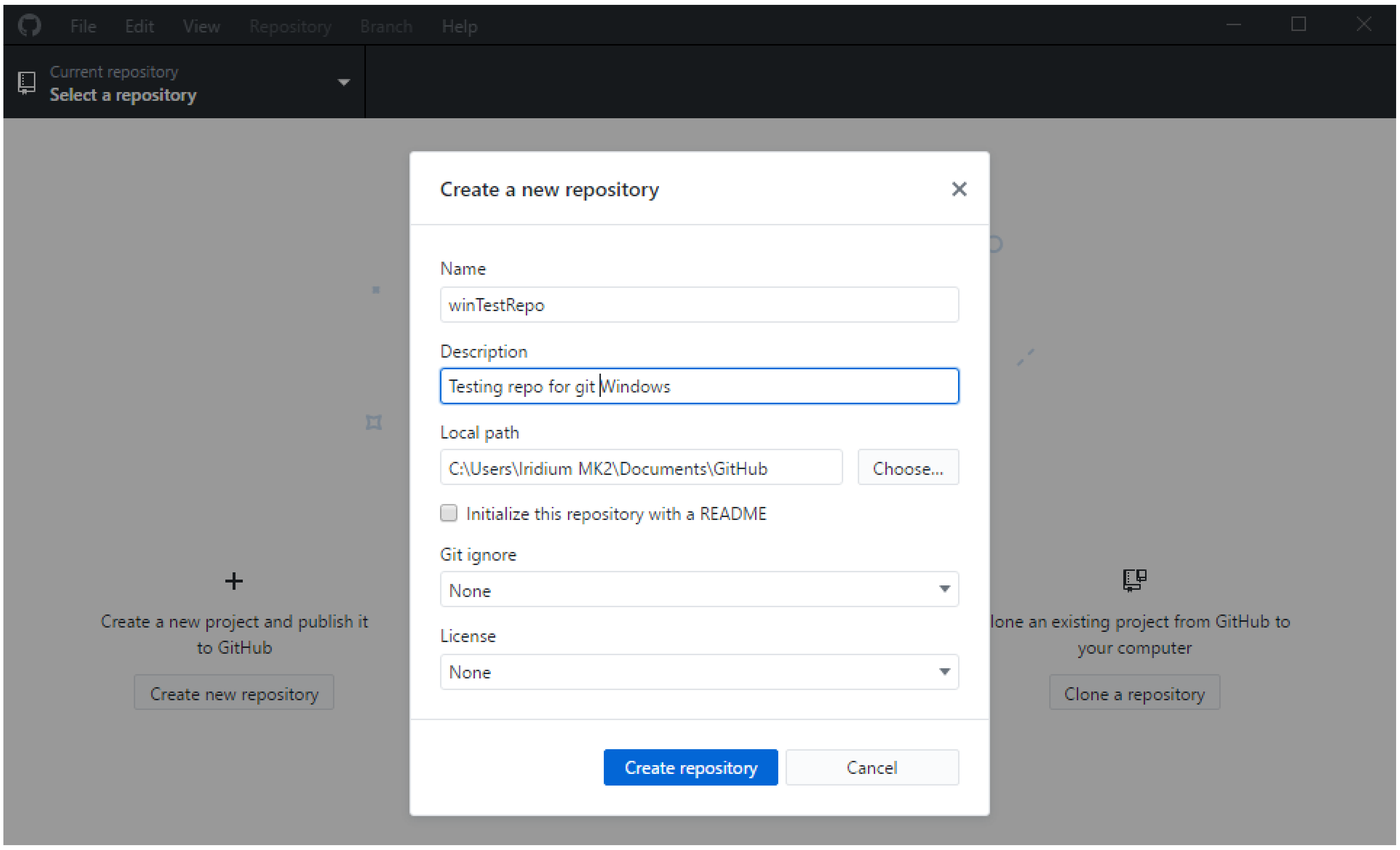
Considering the experiences so far: Turquoise SVN is better; though I didn’t have to configure it and was just using as a client…



```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
iface ens33 inet static
address 192.168.0.103
netmask 255.255.255.0
gateway 192.168.0.1
```

```
et@FailBox:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0
et@FailBox:~$ ifconfig -a
ens33     Link encap:Ethernet  HWaddr 00:0c:29:b0:16:b3
          inet addr:192.168.0.103  Bcast:192.168.0.255  Mask:255.255.25
          inet6 addr: fe80::b9c9:f4e4:6727:49fc/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7893 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2002 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10784536 (10.7 MB)  TX bytes:195526 (195.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:512 errors:0 dropped:0 overruns:0 frame:0
          TX packets:512 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46778 (46.7 KB)  TX bytes:46778 (46.7 KB)

et@FailBox:~$
```

Fig1



Create a new repository

Name: winTestRepo

Description: Testing repo for git Windows

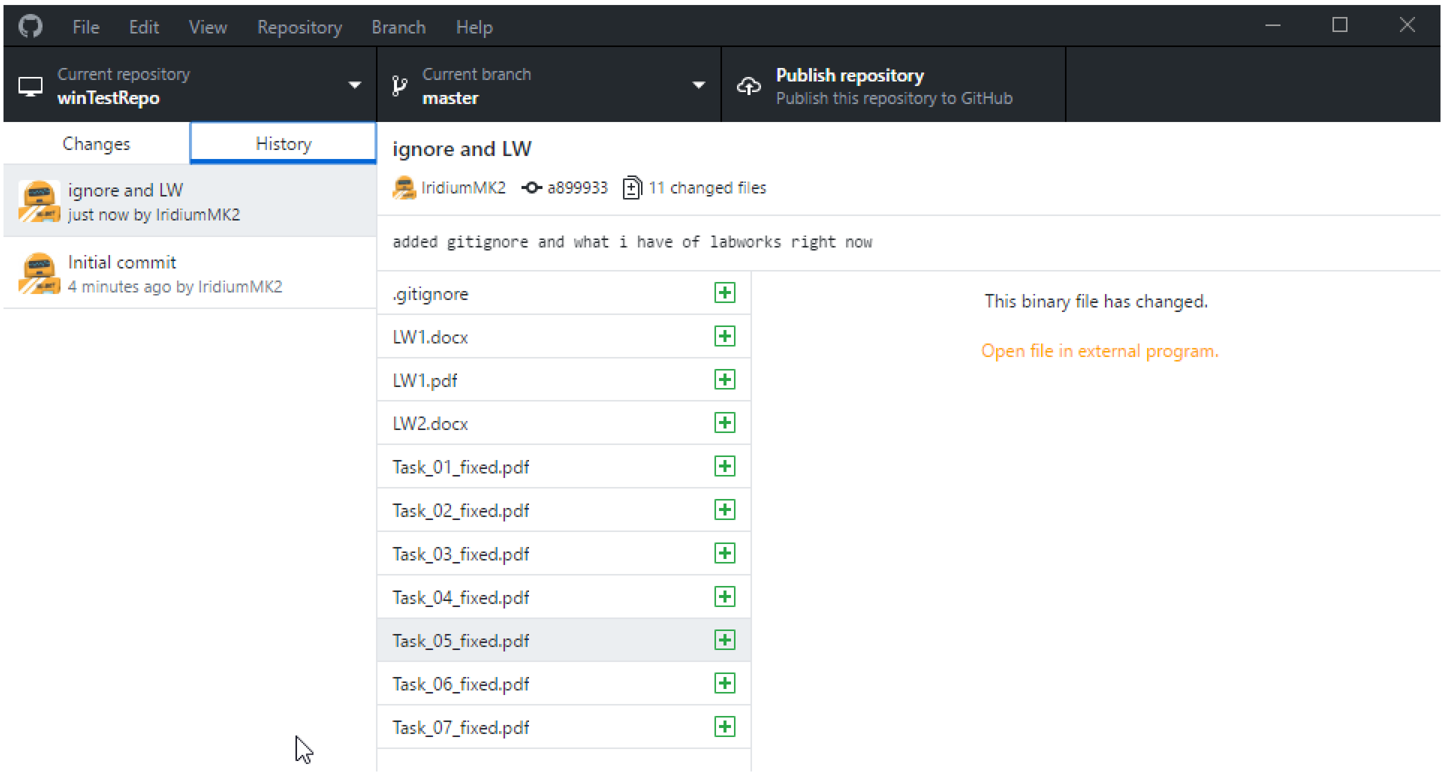
Local path: C:\Users\Iridium MK2\Documents\GitHub

Initialize this repository with a README

Git ignore: None

License: None

Fig2



ignore and LW
just now by IridiumMK2

Initial commit
4 minutes ago by IridiumMK2

ignore and LW

IridiumMK2 a899933 11 changed files

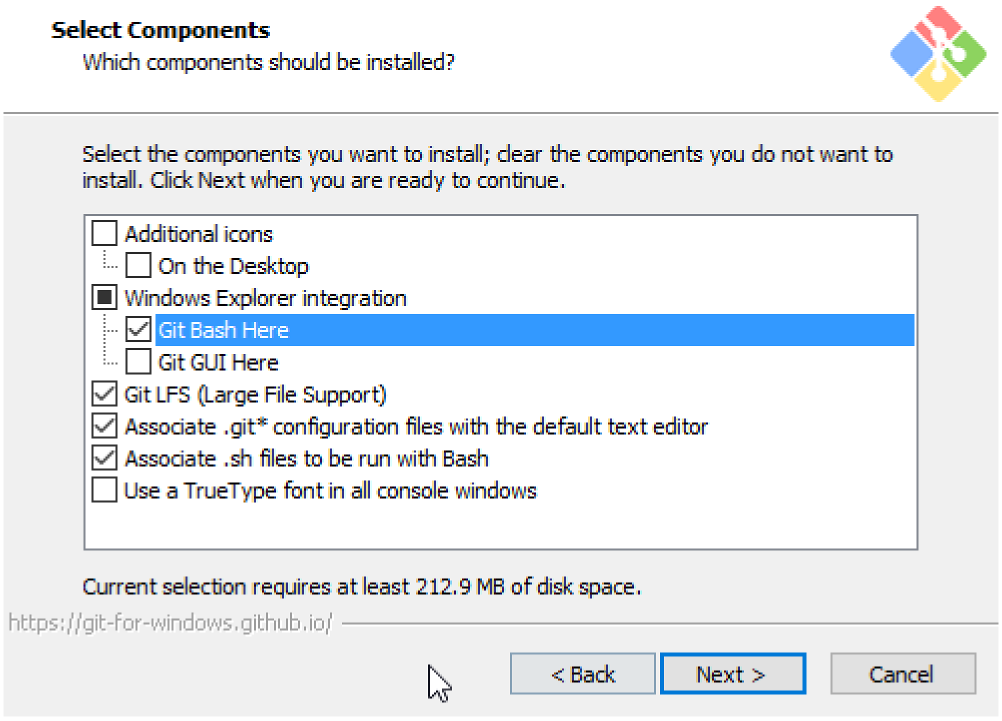
```
added gitignore and what i have of labworks right now
```

.gitignore
LW1.docx
LW1.pdf
LW2.docx
Task_01_fixed.pdf
Task_02_fixed.pdf
Task_03_fixed.pdf
Task_04_fixed.pdf
Task_05_fixed.pdf
Task_06_fixed.pdf
Task_07_fixed.pdf

This binary file has changed.

Open file in external program.

Fig 3



Select Components

Which components should be installed?

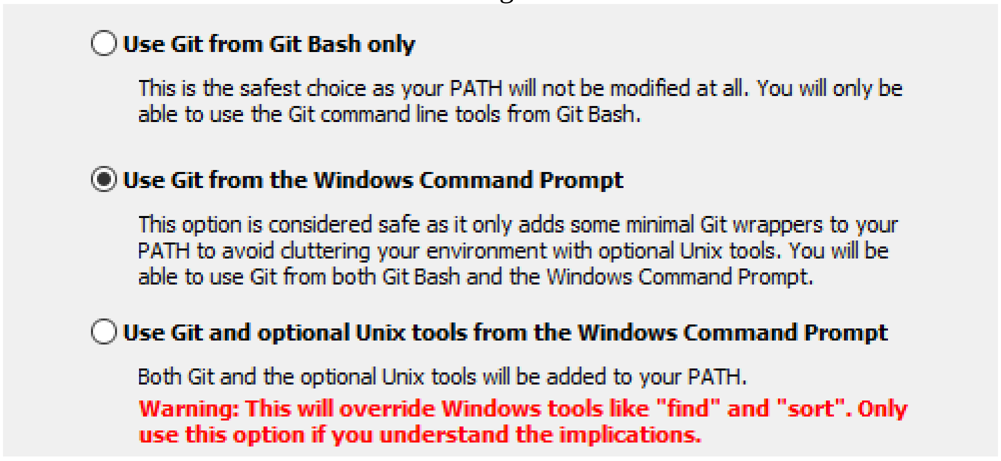
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [ ] Additional icons
  - [ ] On the Desktop
- [■] Windows Explorer integration
  - [✓] Git Bash Here
  - [ ] Git GUI Here
- [✓] Git LFS (Large File Support)
- [✓] Associate .git* configuration files with the default text editor
- [✓] Associate .sh files to be run with Bash
- [ ] Use a TrueType font in all console windows

Current selection requires at least 212.9 MB of disk space.

https://git-for-windows.github.io/

Fig4



○ **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Use Git from the Windows Command Prompt**

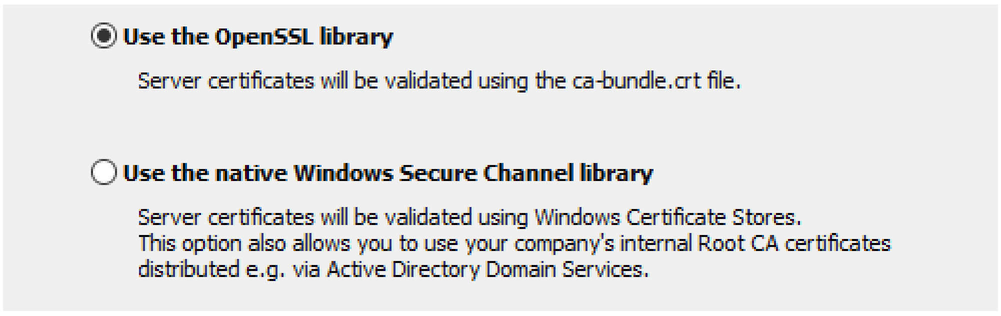
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

○ **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.

Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

Fig5



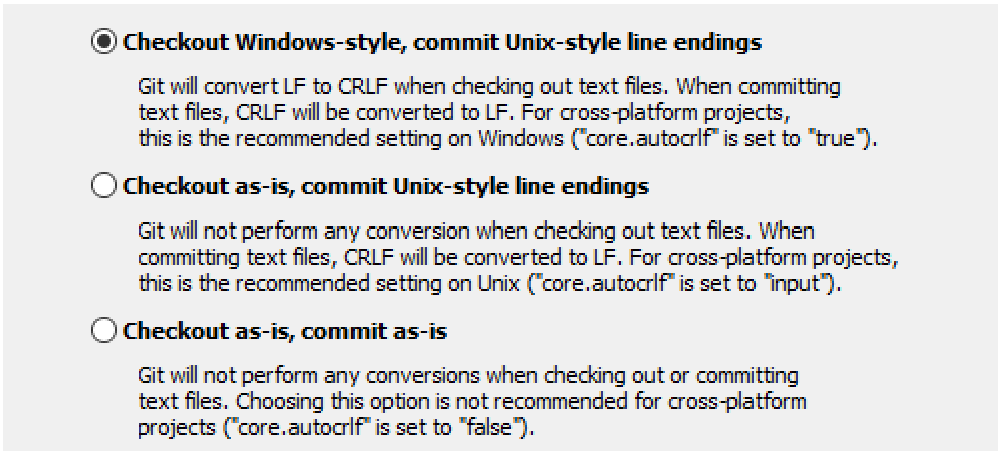
● **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

Fig 6



● **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

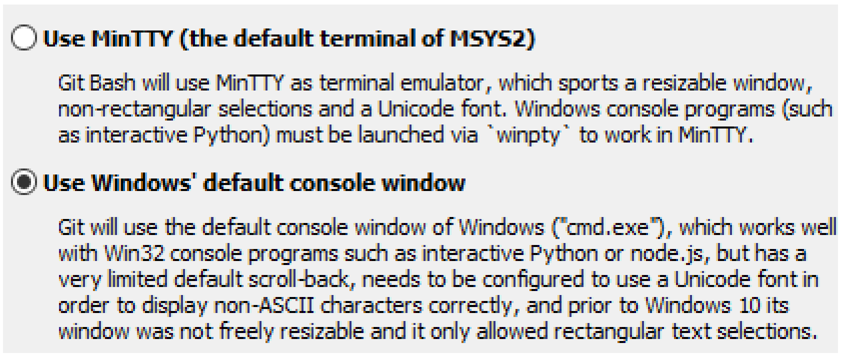
○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

Fig 7



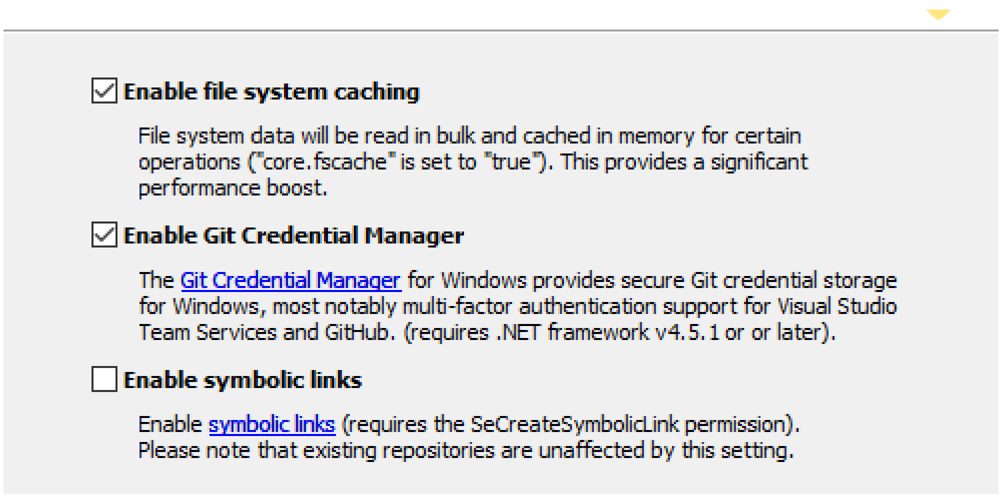
○ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

● **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

Fig 8



✓ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

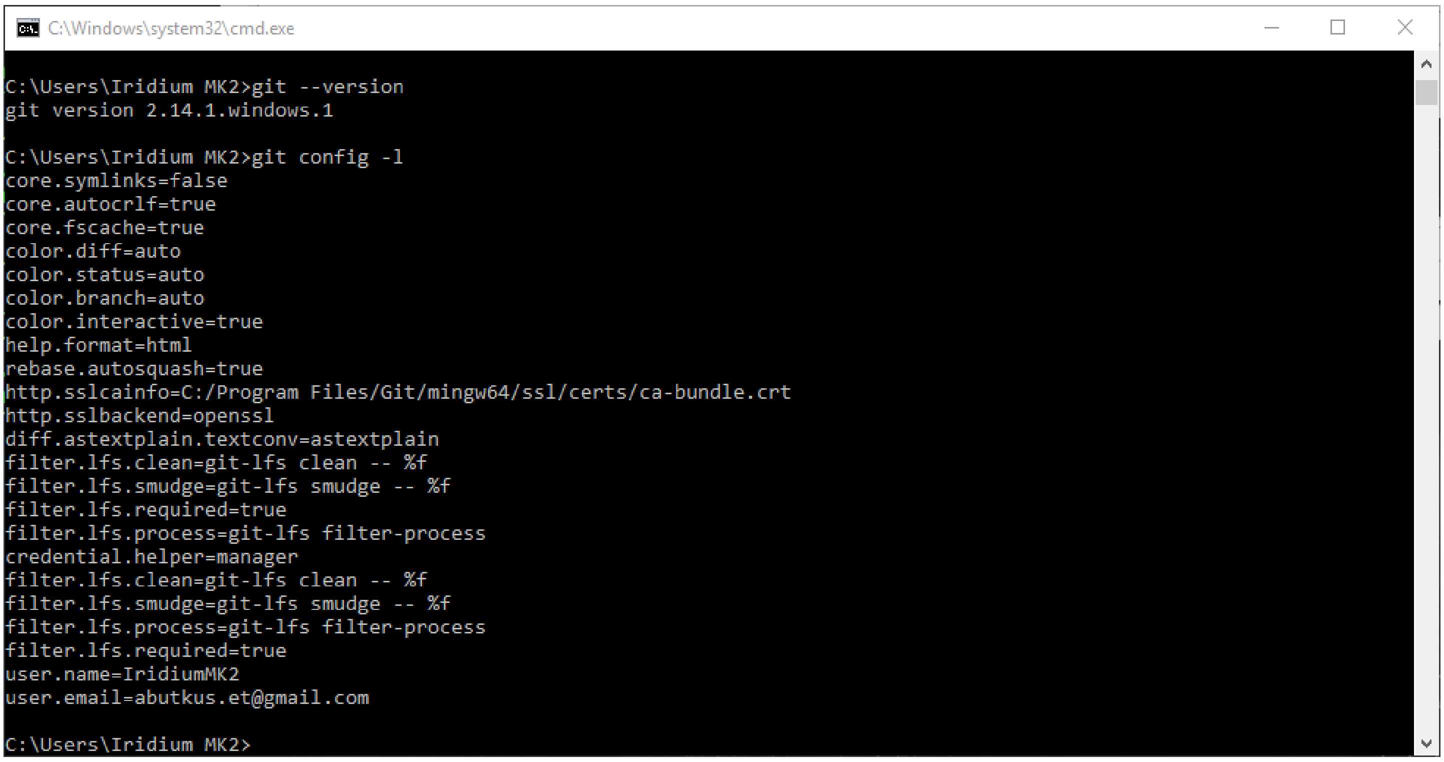
✓ **Enable Git Credential Manager**

The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

☐ **Enable symbolic links**

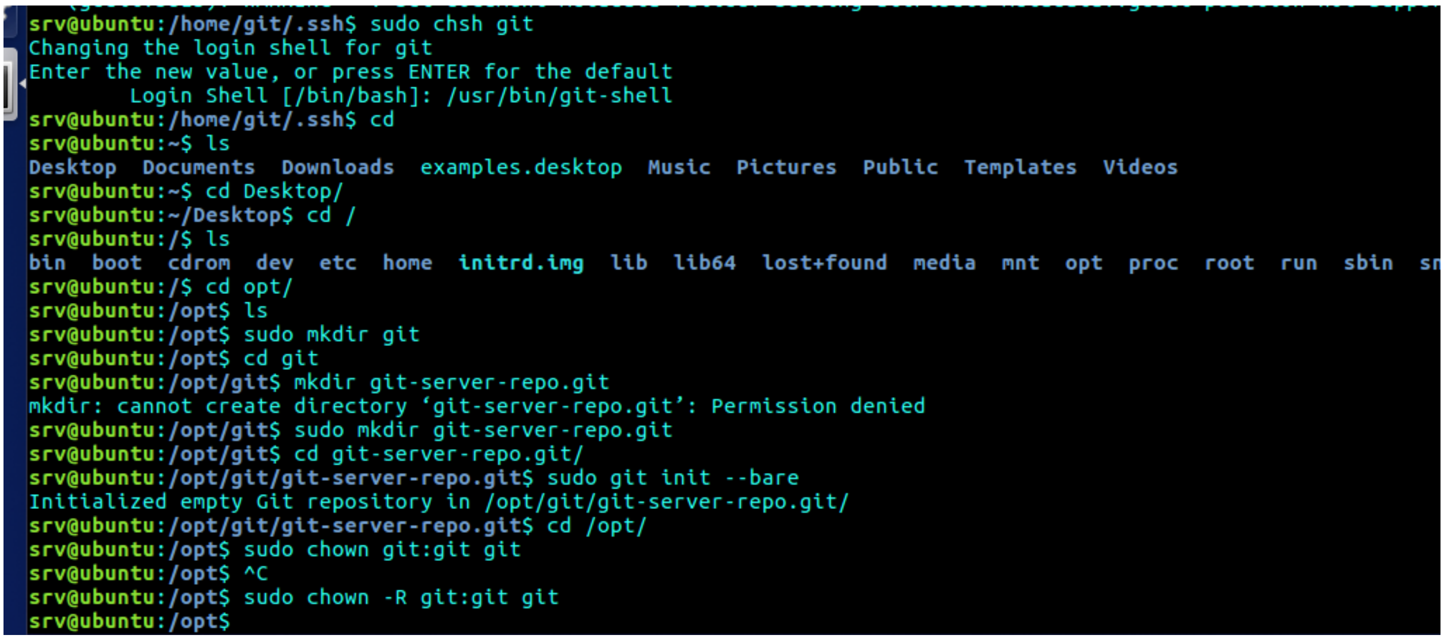
Enable symbolic links (requires the SeCreateSymbolicLink permission).
Please note that existing repositories are unaffected by this setting.

Fig 9



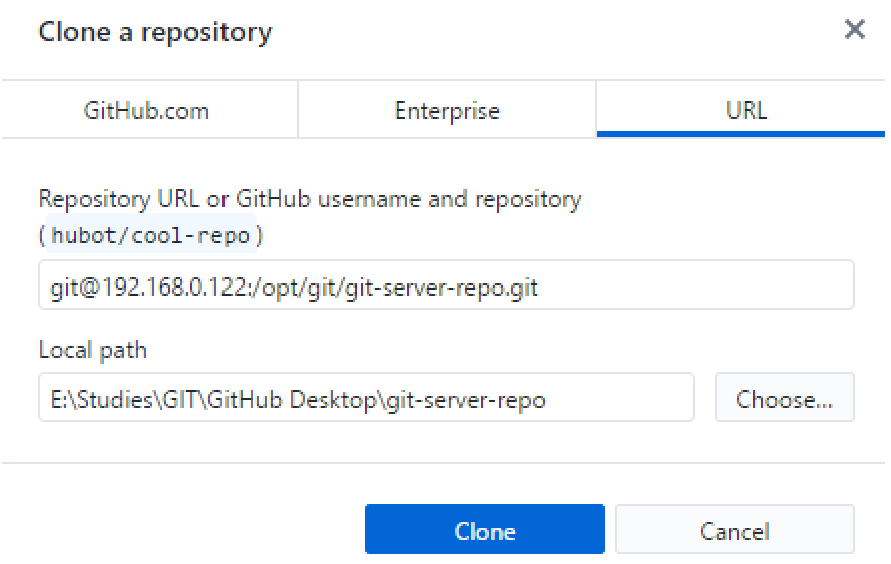
```
C:\Windows\system32\cmd.exe

C:\Users\Iridium MK2>git --version
git version 2.14.1.windows.1

C:\Users\Iridium MK2>git config -l
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.required=true
filter.lfs.process=git-lfs filter-process
credential.helper=manager
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=IridiumMK2
user.email=abutkus.et@gmail.com

C:\Users\Iridium MK2>
```

fig10



```
srv@ubuntu:/home/git/.ssh$ sudo chsh git
Changing the login shell for git
Enter the new value, or press ENTER for the default
        Login Shell [/bin/bash]: /usr/bin/git-shell
srv@ubuntu:/home/git/.ssh$ cd
srv@ubuntu:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  Videos
srv@ubuntu:~$ cd Desktop/
srv@ubuntu:~/Desktop$ cd /
srv@ubuntu:/$ ls
bin  boot  cdrom  dev  etc  home  initrd.img  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  sn
srv@ubuntu:/$ cd opt/
srv@ubuntu:/opt$ ls
srv@ubuntu:/opt$ sudo mkdir git
srv@ubuntu:/opt$ cd git
srv@ubuntu:/opt/git$ mkdir git-server-repo.git
mkdir: cannot create directory ‘git-server-repo.git’: Permission denied
srv@ubuntu:/opt/git$ sudo mkdir git-server-repo.git
srv@ubuntu:/opt/git$ cd git-server-repo.git/
srv@ubuntu:/opt/git/git-server-repo.git$ sudo git init --bare
Initialized empty Git repository in /opt/git/git-server-repo.git/
srv@ubuntu:/opt/git/git-server-repo.git$ cd /opt/
srv@ubuntu:/opt$ sudo chown git:git git
srv@ubuntu:/opt$ ^C
srv@ubuntu:/opt$ sudo chown -R git:git git
srv@ubuntu:/opt$
```

Fig 11



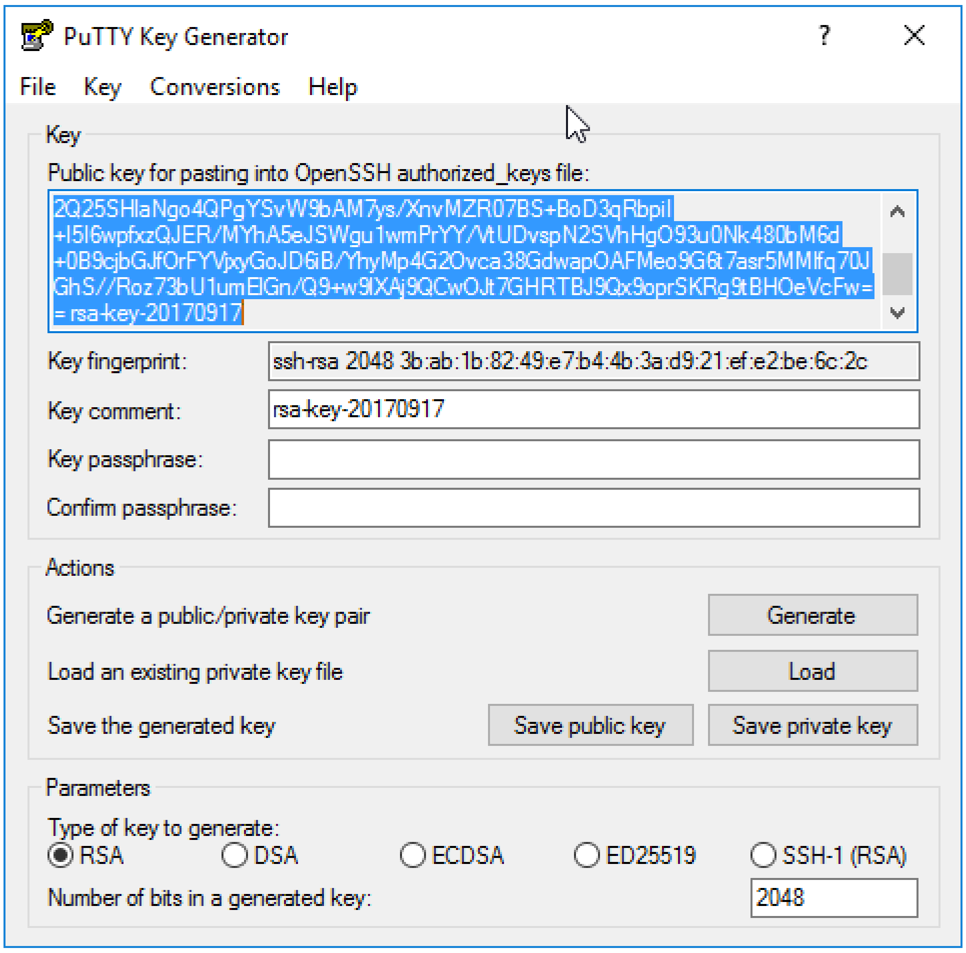
Repository URL or GitHub username and repository (`hubot/cool-repo`)

git@192.168.0.122:/opt/git/git-server-repo.git

Local path

E:\Studies\GIT\GitHub Desktop\git-server-repo

Fig 12



Public key for pasting into OpenSSH authorized_keys file:

```
2Q25SHlaNgo4QPgYSvW9bAM7ys/XnvMZR07BS+BoD3qRbpil
+I5I6wpfxzQJER/MYhA5eJSWgu1wmPrYY/VtUDvspN2SVhHgO93u0Nk480bM6d
+0B9cjbGJfOrFYVpxyGoJD6iB/YhyMp4G2Ovca38GdwapOAFMeo9G6t7asr5MMIfq70J
GhS//Roz73bU1umElGn/Q9+w9IXAj9QCwOJt7GHRTBJ9Qx9oprSKRg9tBHOeVcFw=
= rsa-key-20170917
```

Key fingerprint: ssh-rsa 2048 3b:ab:1b:82:49:e7:b4:4b:3a:d9:21:ef:e2:be:6c:2c

Key comment: rsa-key-20170917

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

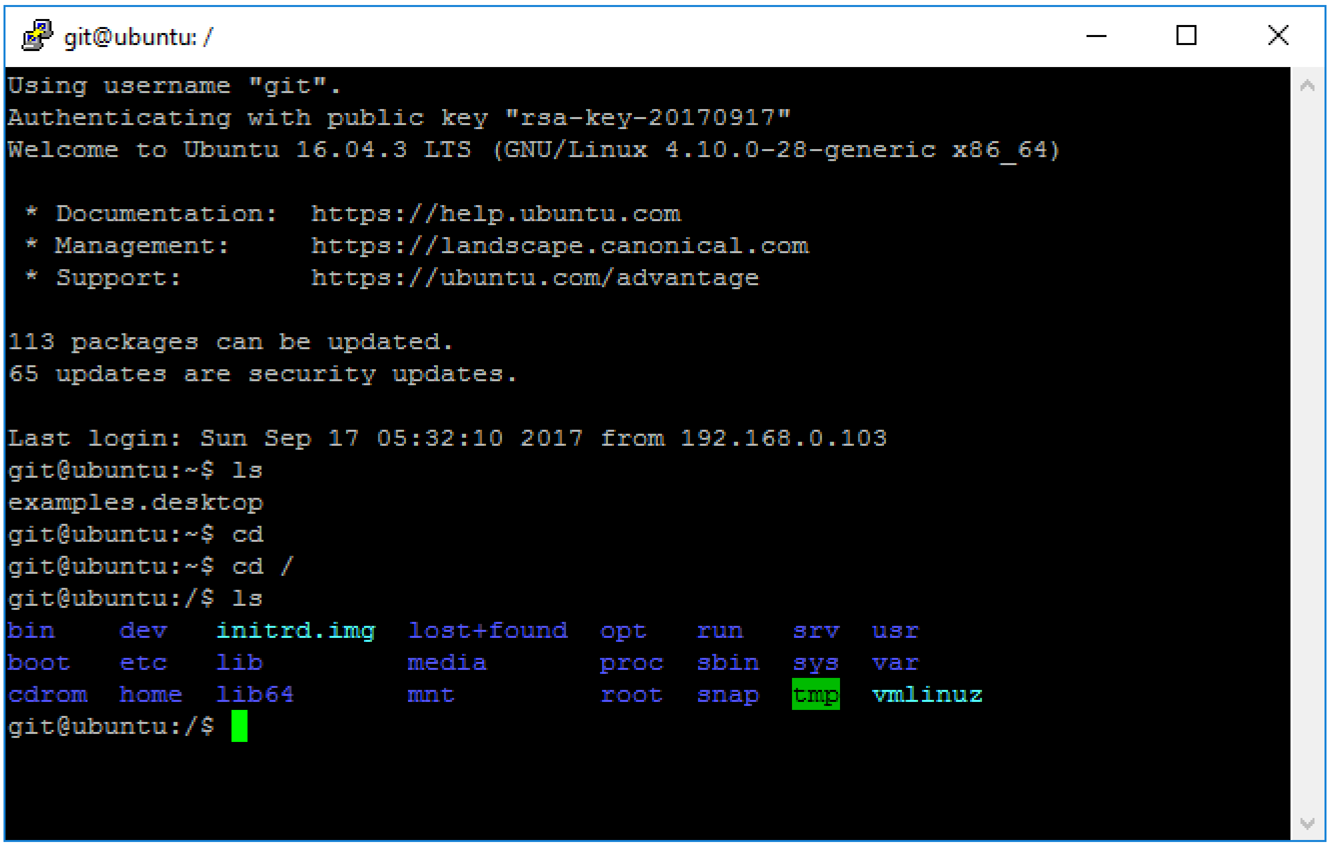
Load an existing private key file

Save the generated key

Type of key to generate:
[✓ RSA] DSA ECDSA ED25519 SSH-1 (RSA)

Number of bits in a generated key: 2048

Fig 13



```
Using username "git".
Authenticating with public key "rsa-key-20170917"
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

113 packages can be updated.
65 updates are security updates.

Last login: Sun Sep 17 05:32:10 2017 from 192.168.0.103
git@ubuntu:~$ ls
examples.desktop
git@ubuntu:~$ cd
git@ubuntu:~$ cd /
git@ubuntu:/$ ls
bin    dev   initrd.img  lost+found  opt   run   srv  usr
boot   etc   lib         media       proc  sbin  sys  var
cdrom  home  lib64       mnt         root  snap  tmp  vmlinuz
git@ubuntu:/$
```

Fig 14

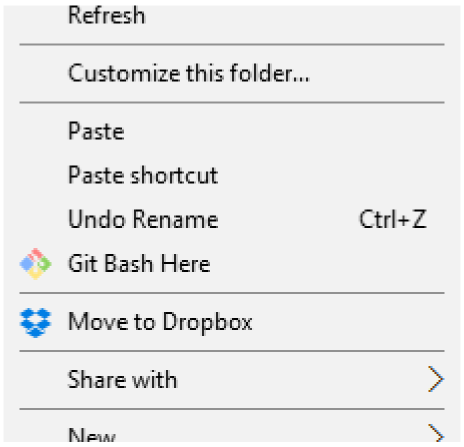
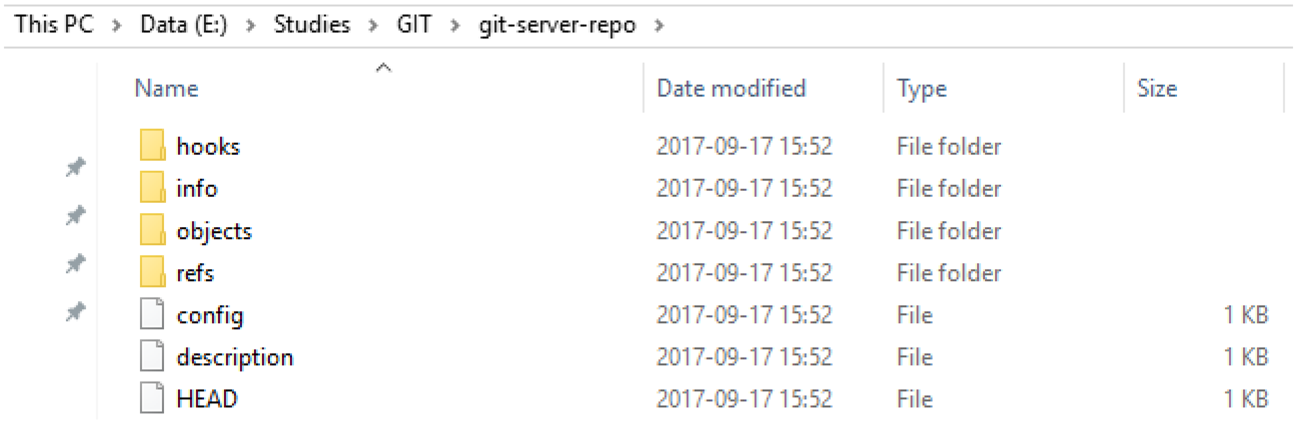
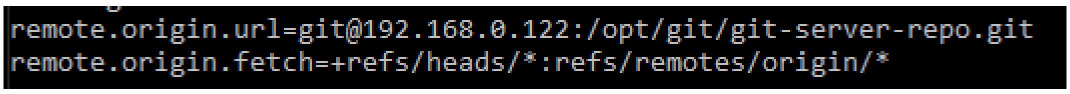


Fig 15



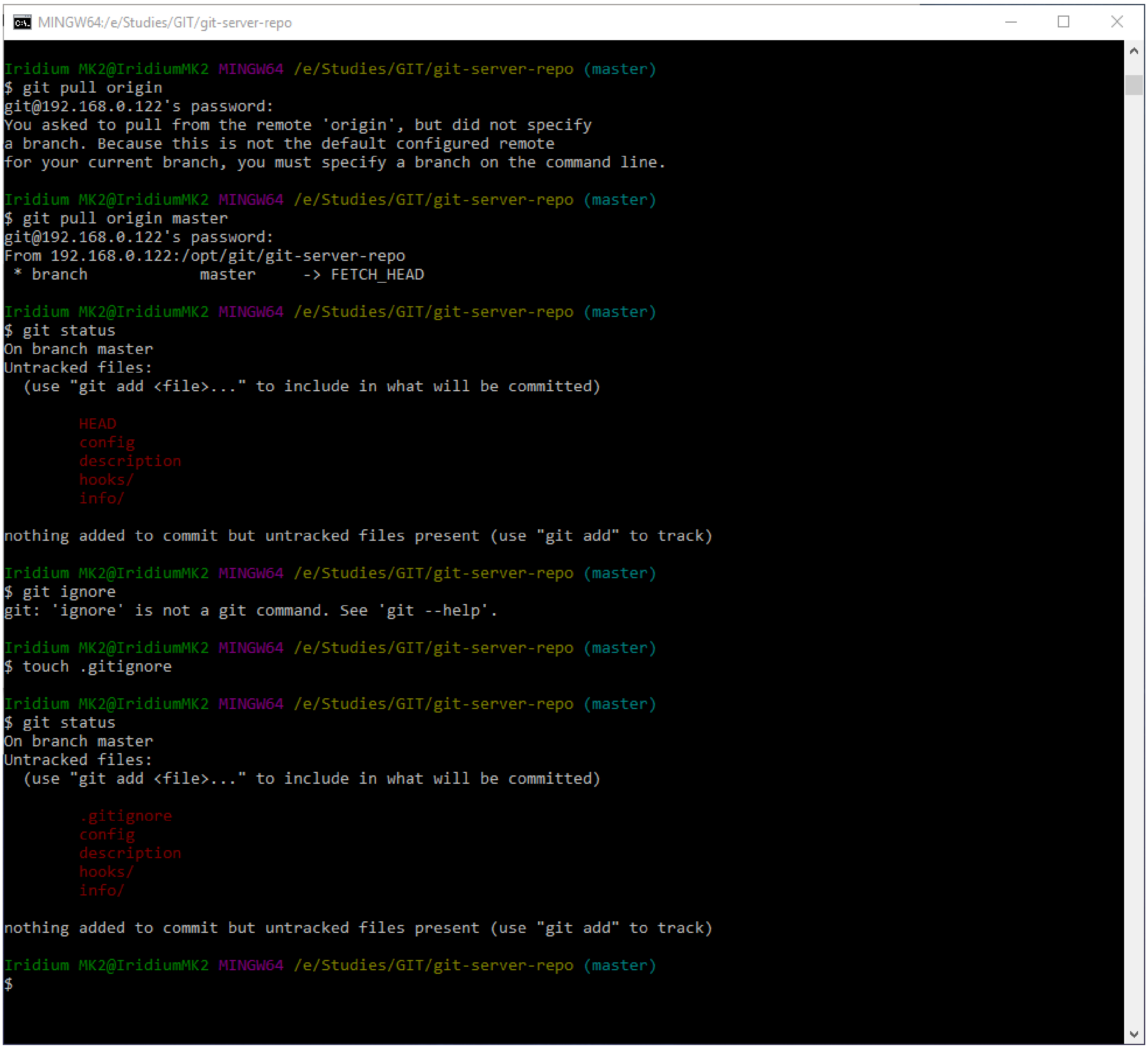
Name	Date modified	Type	Size
hooks	2017-09-17 15:52	File folder	
info	2017-09-17 15:52	File folder	
objects	2017-09-17 15:52	File folder	
refs	2017-09-17 15:52	File folder	
config	2017-09-17 15:52	File	1 KB
description	2017-09-17 15:52	File	1 KB
HEAD	2017-09-17 15:52	File	1 KB

Fig 16



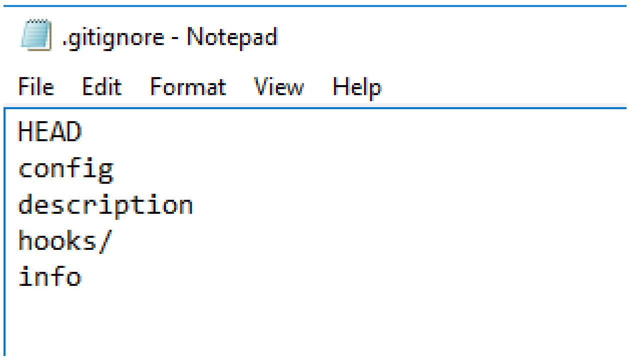
```
remote.origin.url=git@192.168.0.122:/opt/git/git-server-repo.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

Fig 17



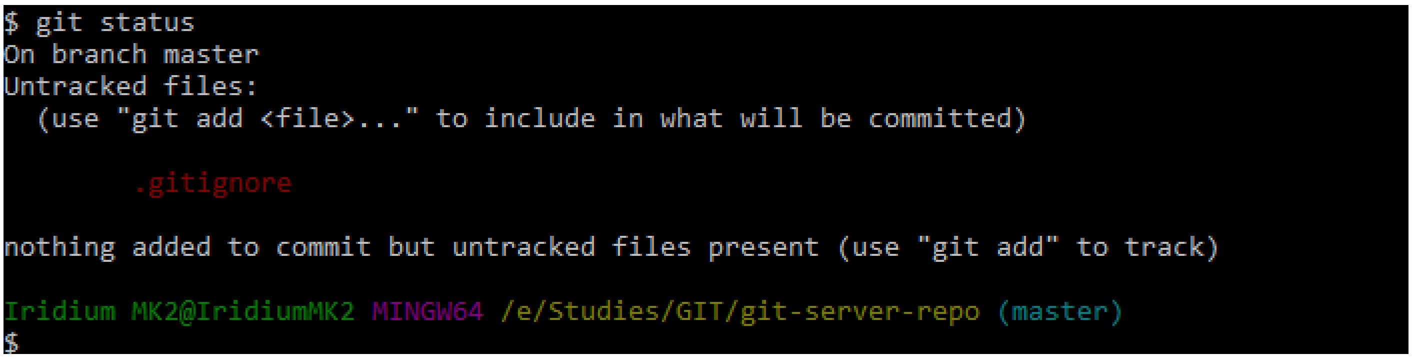
```
Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git pull origin
git@192.168.0.122's password:
You asked to pull from the remote 'origin', but did not specify
a branch. Because this is not the default configured remote
for your current branch, you must specify a branch on the command line.

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git pull origin master
git@192.168.0.122's password:
From 192.168.0.122:/opt/git/git-server-repo
 * branch            master     -> FETCH_HEAD

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        HEAD
        config
        description
        hooks/
        info/

nothing added to commit but untracked files present (use "git add" to track)

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git ignore
git: 'ignore' is not a git command. See 'git --help'.

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ touch .gitignore

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        config
        description
        hooks/
        info/

nothing added to commit but untracked files present (use "git add" to track)

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$
```

Fig 18



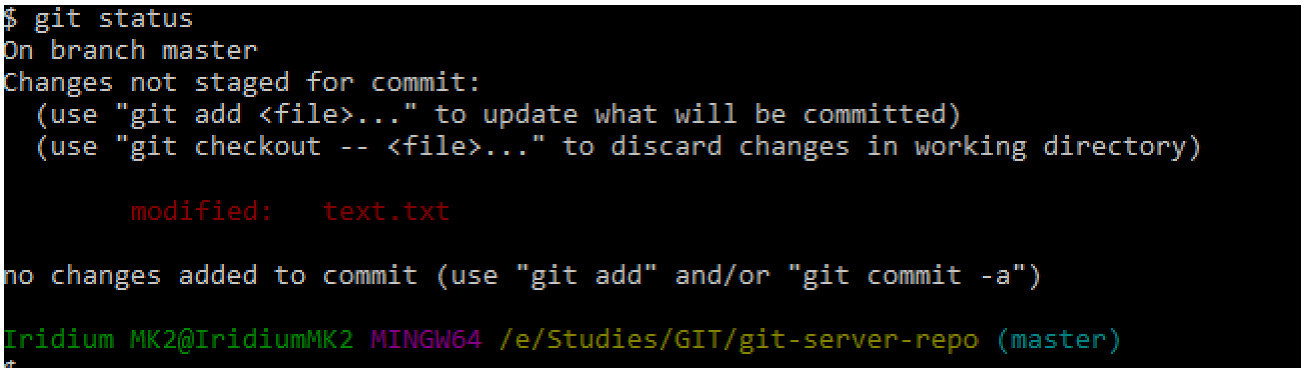
```
HEAD
config
description
hooks/
info
```

Fig 19



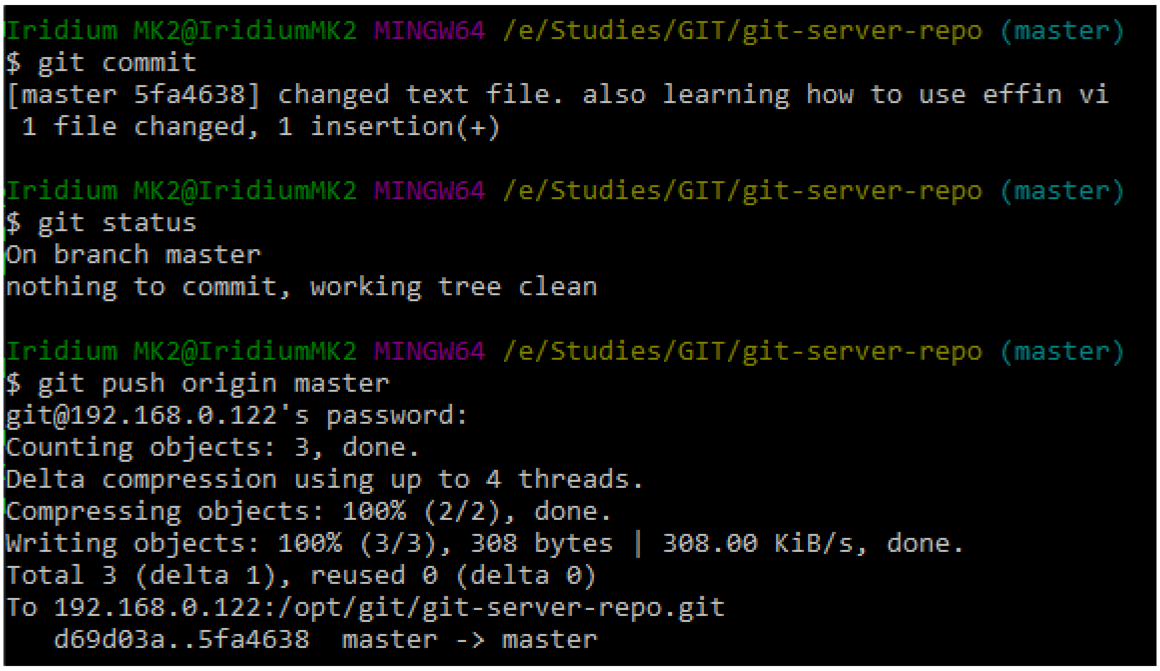
```
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore

nothing added to commit but untracked files present (use "git add" to track)

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$
```

Fig 20



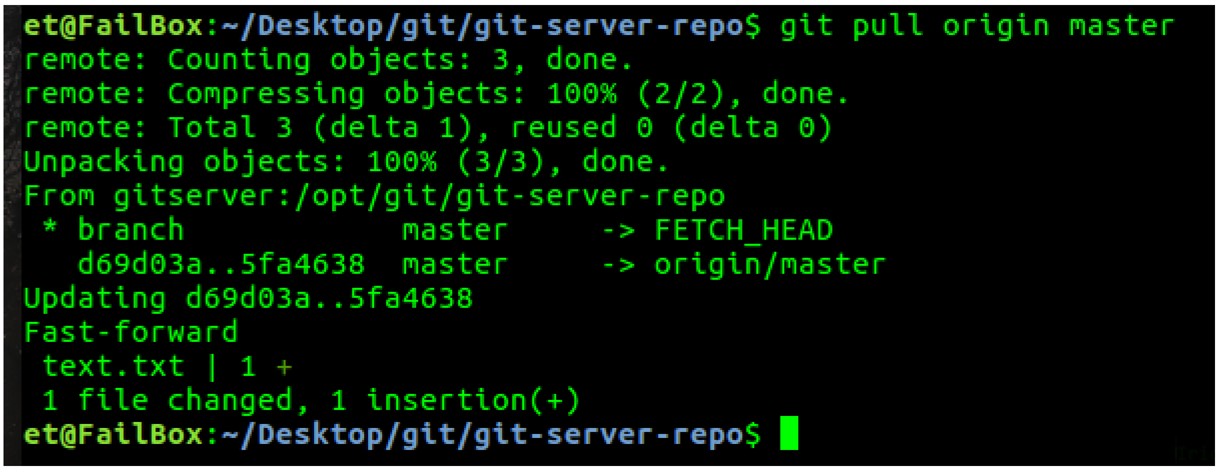
```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   text.txt

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

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
```

Fig 21



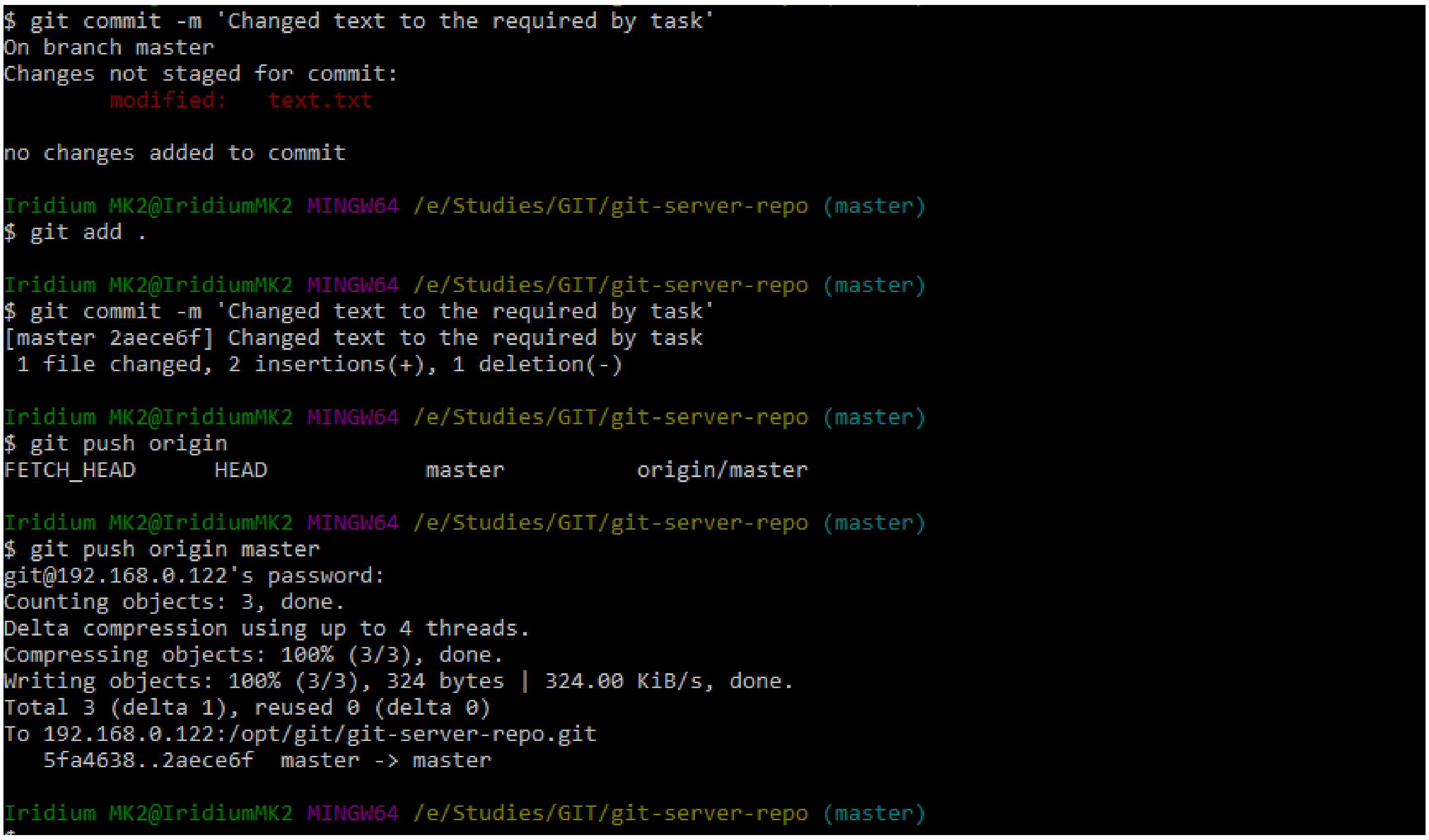
```
Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git commit
[master 5fa4638] changed text file. also learning how to use effin vi
 1 file changed, 1 insertion(+)

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git status
On branch master
nothing to commit, working tree clean

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git push origin master
git@192.168.0.122's password:
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To 192.168.0.122:/opt/git/git-server-repo.git
   d69d03a..5fa4638  master -> master
```

Fig 22



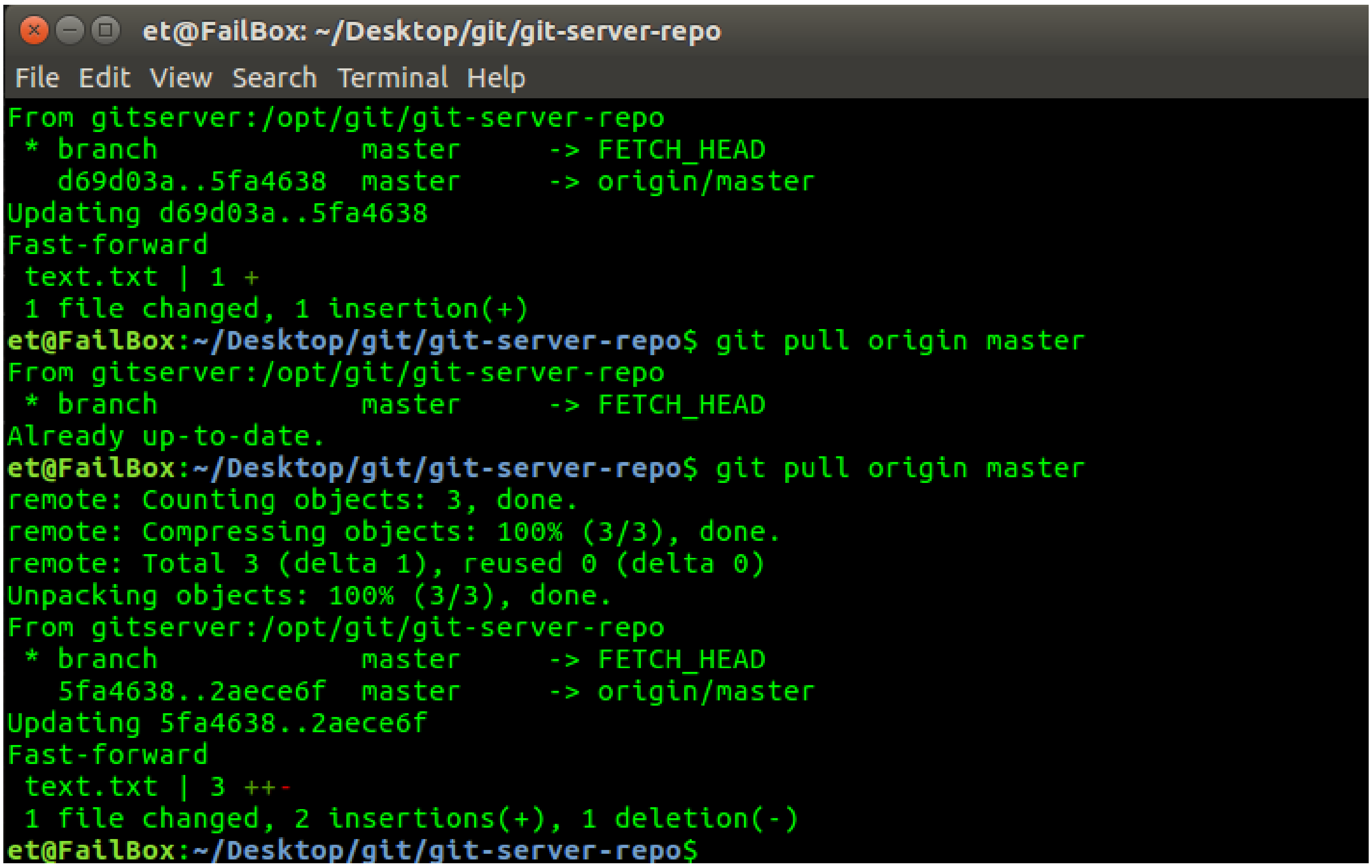
```
et@FailBox:~/Desktop/git/git-server-repo$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From gitserver:/opt/git/git-server-repo
 * branch            master     -> FETCH_HEAD
   d69d03a..5fa4638  master     -> origin/master
Updating d69d03a..5fa4638
Fast-forward
 text.txt | 1 +
 1 file changed, 1 insertion(+)
et@FailBox:~/Desktop/git/git-server-repo$
```

Fig 23



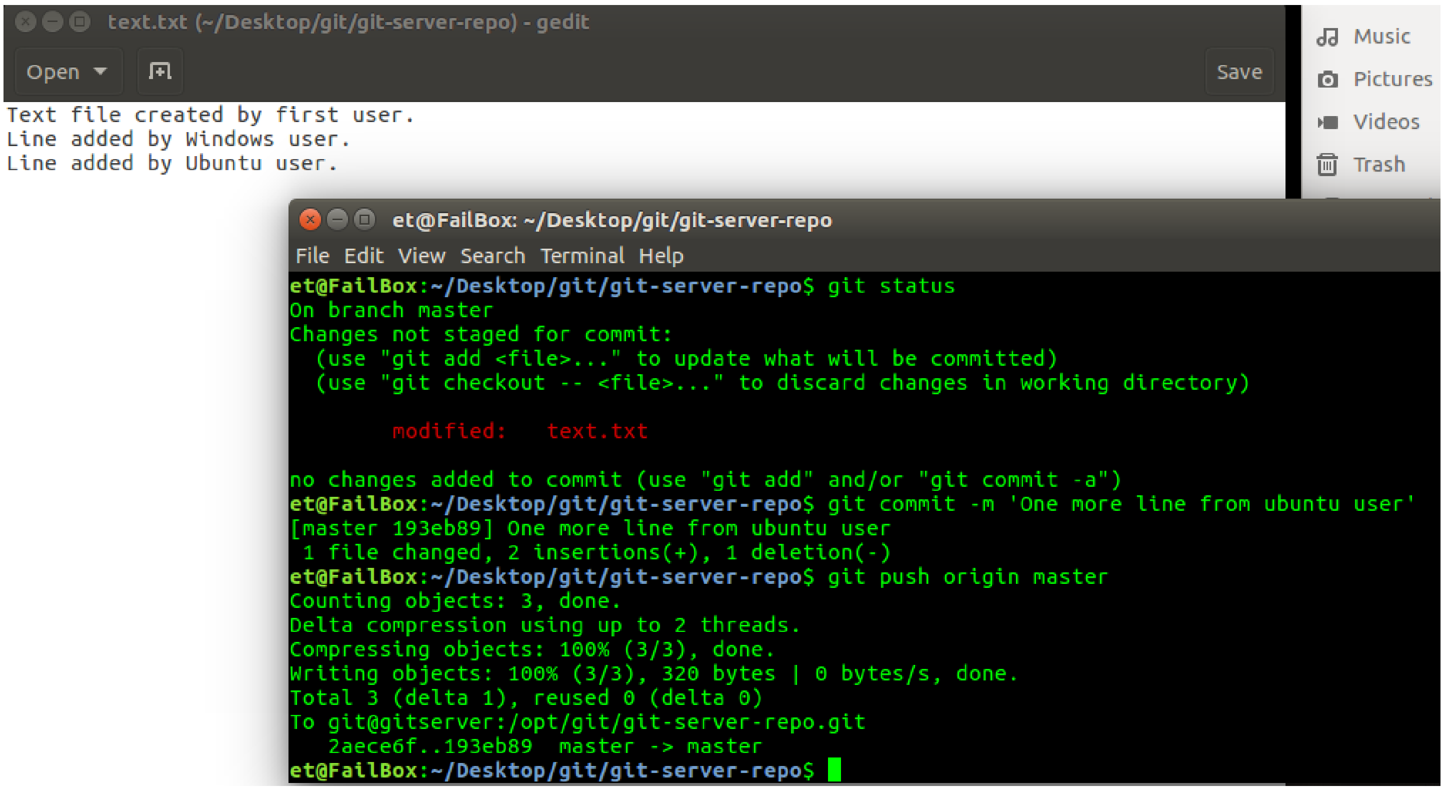
```
$ git commit -m 'Changed text to the required by task'
On branch master
Changes not staged for commit:
        modified:   text.txt

no changes added to commit

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git add .

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git commit -m 'Changed text to the required by task'
[master 2aece6f] Changed text to the required by task
 1 file changed, 2 insertions(+), 1 deletion(-)

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git push origin
FETCH_HEAD      HEAD            master          origin/master

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git push origin master
git@192.168.0.122's password:
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 324 bytes | 324.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To 192.168.0.122:/opt/git/git-server-repo.git
   5fa4638..2aece6f  master -> master

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
```

Fig 24



```
et@FailBox: ~/Desktop/git/git-server-repo
File  Edit  View  Search  Terminal  Help
From gitserver:/opt/git/git-server-repo
 * branch            master     -> FETCH_HEAD
   d69d03a..5fa4638  master     -> origin/master
Updating d69d03a..5fa4638
Fast-forward
 text.txt | 1 +
 1 file changed, 1 insertion(+)
et@FailBox:~/Desktop/git/git-server-repo$ git pull origin master
From gitserver:/opt/git/git-server-repo
 * branch            master     -> FETCH_HEAD
Already up-to-date.
et@FailBox:~/Desktop/git/git-server-repo$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From gitserver:/opt/git/git-server-repo
 * branch            master     -> FETCH_HEAD
   5fa4638..2aece6f  master     -> origin/master
Updating 5fa4638..2aece6f
Fast-forward
 text.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
et@FailBox:~/Desktop/git/git-server-repo$
```

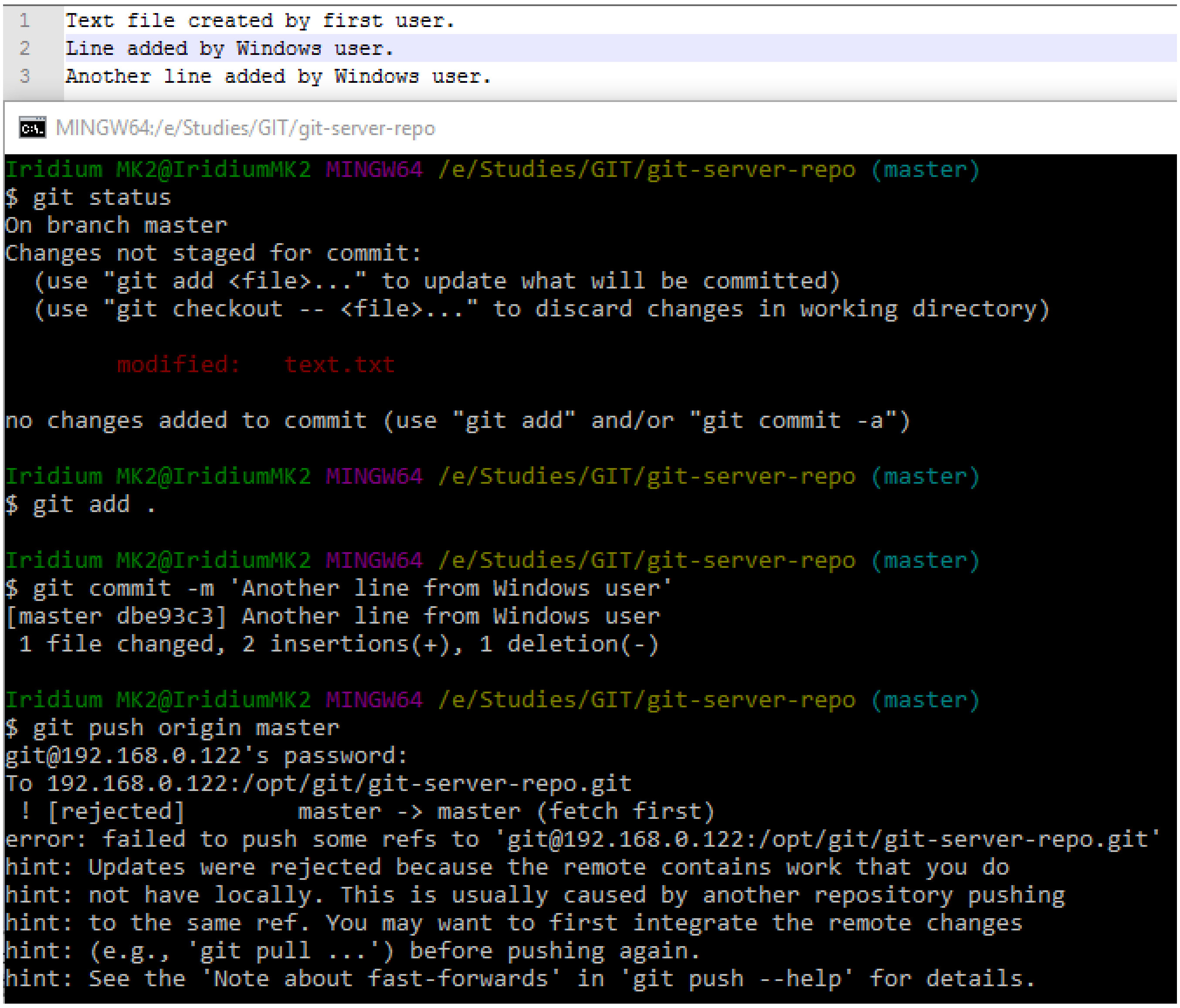
Fig 25


text.txt (~/Desktop/git/git-server-repo) - gedit
Open
Save
Music
Pictures
Videos
Trash
et@FailBox: ~/Desktop/git/git-server-repo
File Edit View Search Terminal Help

```
Text file created by first user.
Line added by Windows user.
Line added by Ubuntu user.
```

```
et@FailBox:~/Desktop/git/git-server-repo$ 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:   text.txt

no changes added to commit (use "git add" and/or "git commit -a")
et@FailBox:~/Desktop/git/git-server-repo$ git commit -m 'One more line from ubuntu user'
[master 193eb89] One more line from ubuntu user
 1 file changed, 2 insertions(+), 1 deletion(-)
et@FailBox:~/Desktop/git/git-server-repo$ git push origin master
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 320 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To git@gitserver:/opt/git/git-server-repo.git
   2aece6f..193eb89  master -> master
et@FailBox:~/Desktop/git/git-server-repo$
```

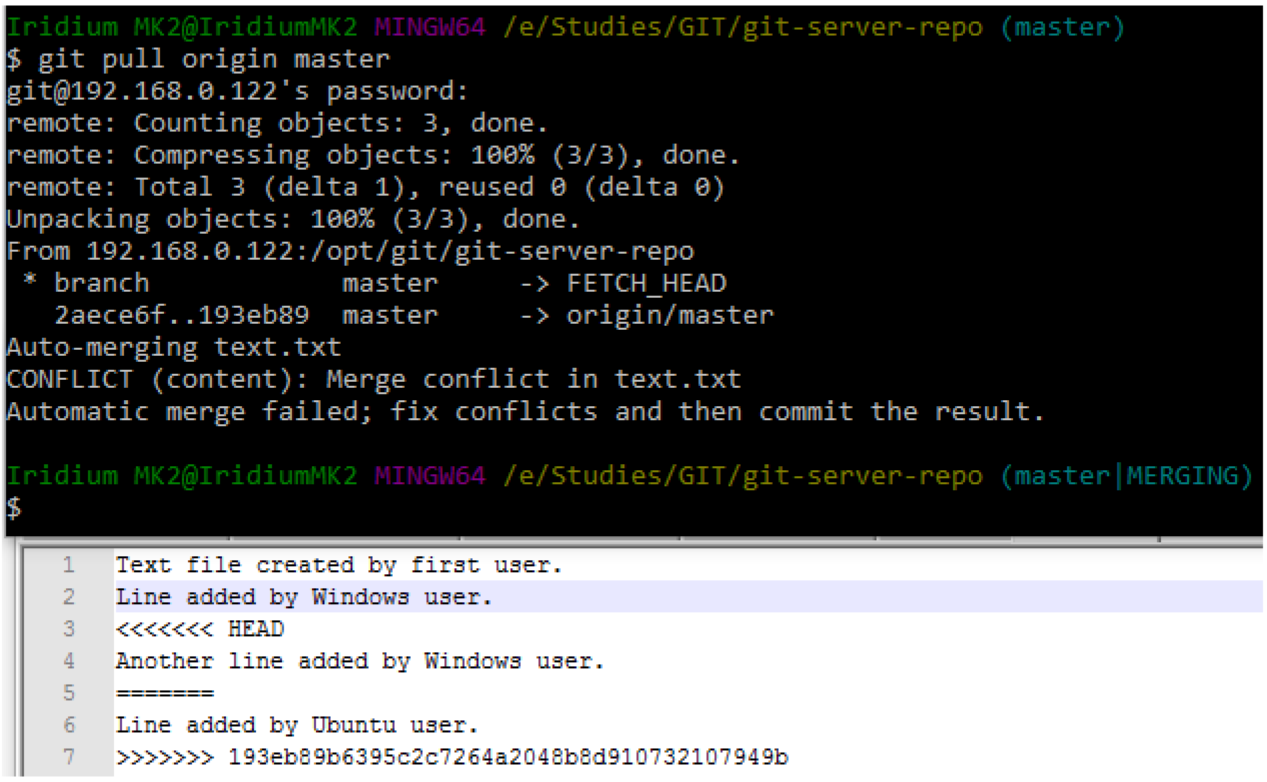
Fig 26



```
Text file created by first user.
Line added by Windows user.
Another line added by Windows user.
```

```
Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (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:   text.txt

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

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git add .

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git commit -m 'Another line from Windows user'
[master dbe93c3] Another line from Windows user
 1 file changed, 2 insertions(+), 1 deletion(-)

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git push origin master
git@192.168.0.122's password:
To 192.168.0.122:/opt/git/git-server-repo.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@192.168.0.122:/opt/git/git-server-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

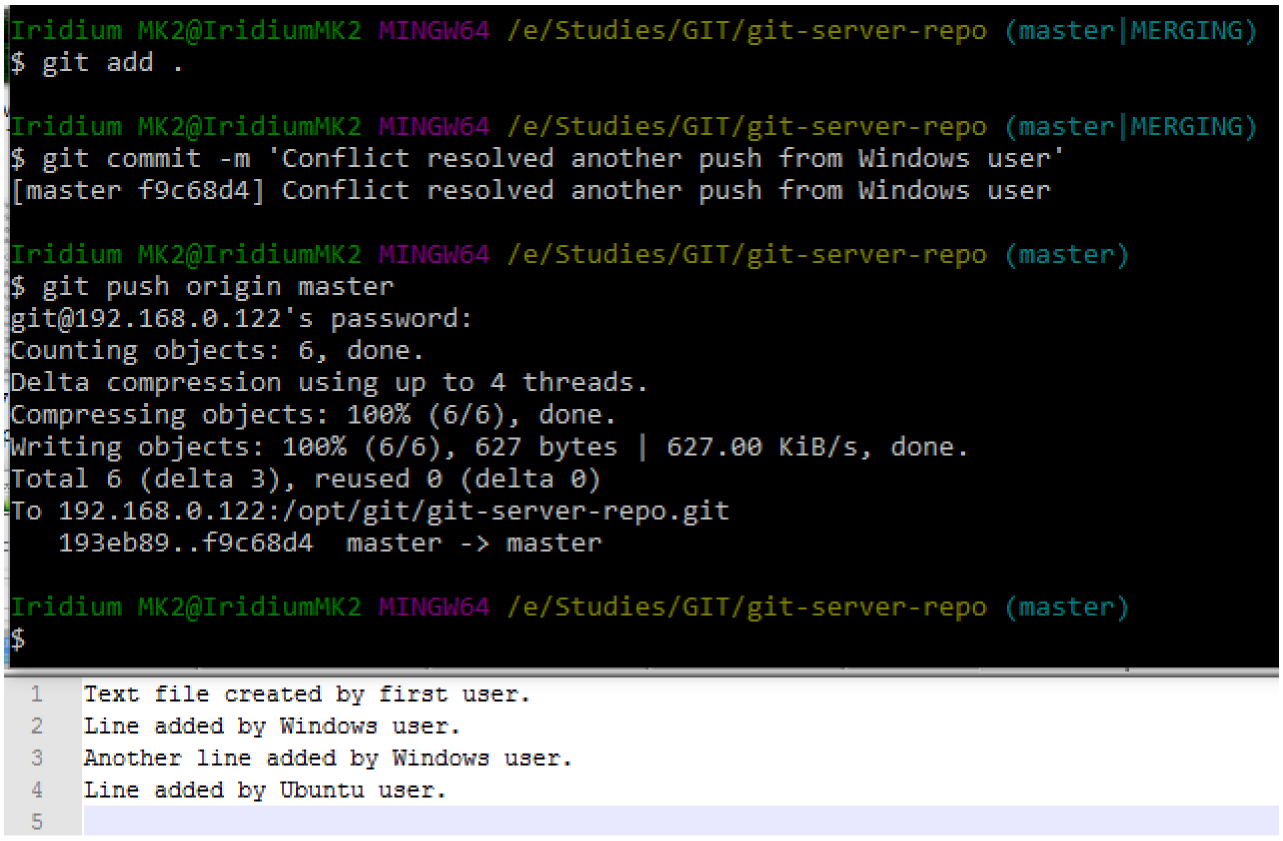
Fig 27



```
Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git pull origin master
git@192.168.0.122's password:
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From 192.168.0.122:/opt/git/git-server-repo
 * branch            master     -> FETCH_HEAD
   2aece6f..193eb89  master     -> origin/master
Auto-merging text.txt
CONFLICT (content): Merge conflict in text.txt
Automatic merge failed; fix conflicts and then commit the result.

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master|MERGING)
$
```

```
Text file created by first user.
Line added by Windows user.
<<<<<<< HEAD
Another line added by Windows user.
=======
Line added by Ubuntu user.
>>>>>>> 193eb89b6395c2c7264a2048b8d910732107949b
```

Fig 28



```
Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master|MERGING)
$ git add .

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master|MERGING)
$ git commit -m 'Conflict resolved another push from Windows user'
[master f9c68d4] Conflict resolved another push from Windows user

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$ git push origin master
git@192.168.0.122's password:
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 627 bytes | 627.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
To 192.168.0.122:/opt/git/git-server-repo.git
   193eb89..f9c68d4  master -> master

Iridium MK2@IridiumMK2 MINGW64 /e/Studies/GIT/git-server-repo (master)
$
```

```
Text file created by first user.
Line added by Windows user.
Another line added by Windows user.
Line added by Ubuntu user.

```

Fig. 29