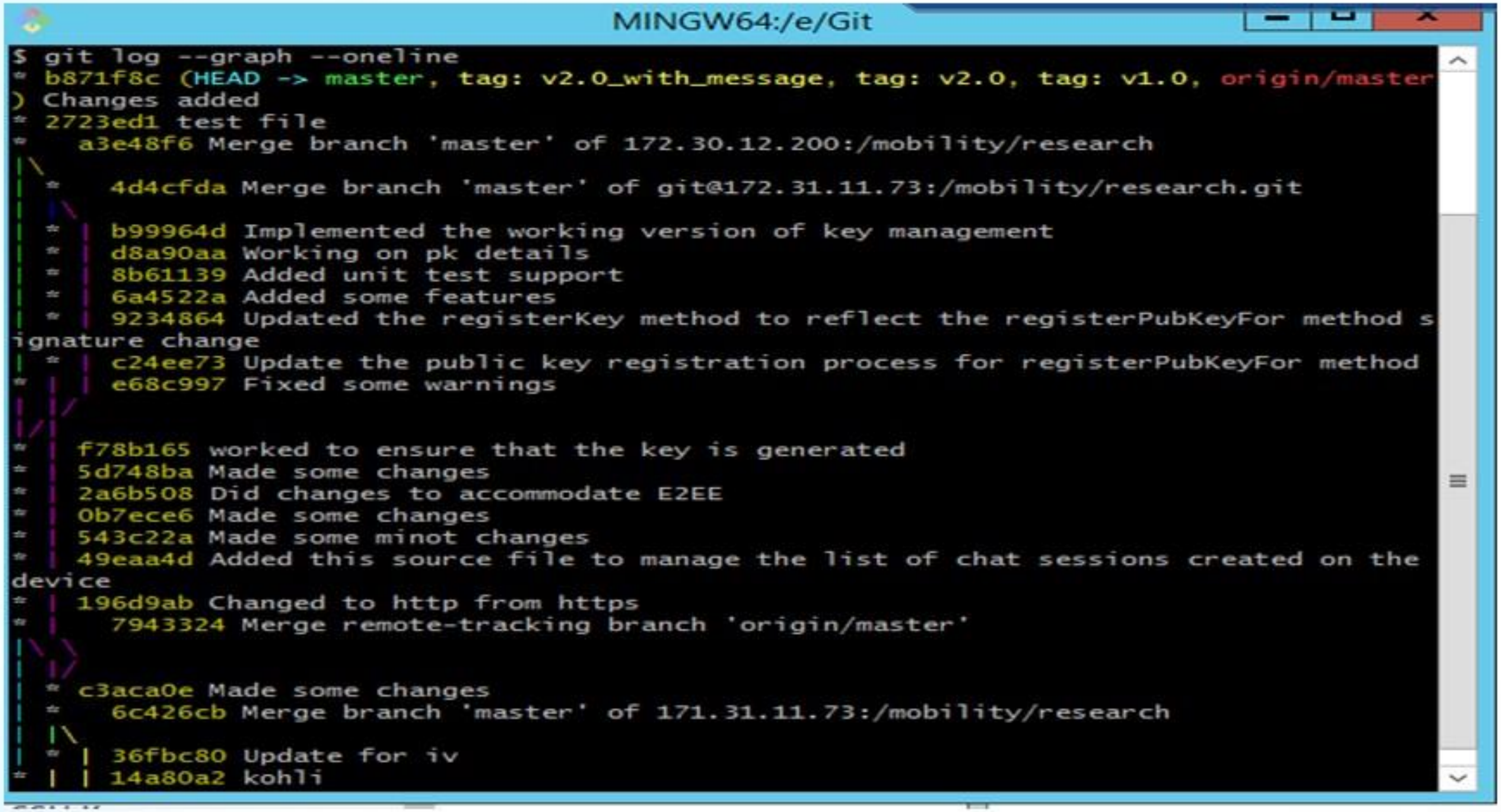
Branching, Merging with GIT

Visualizing Branches

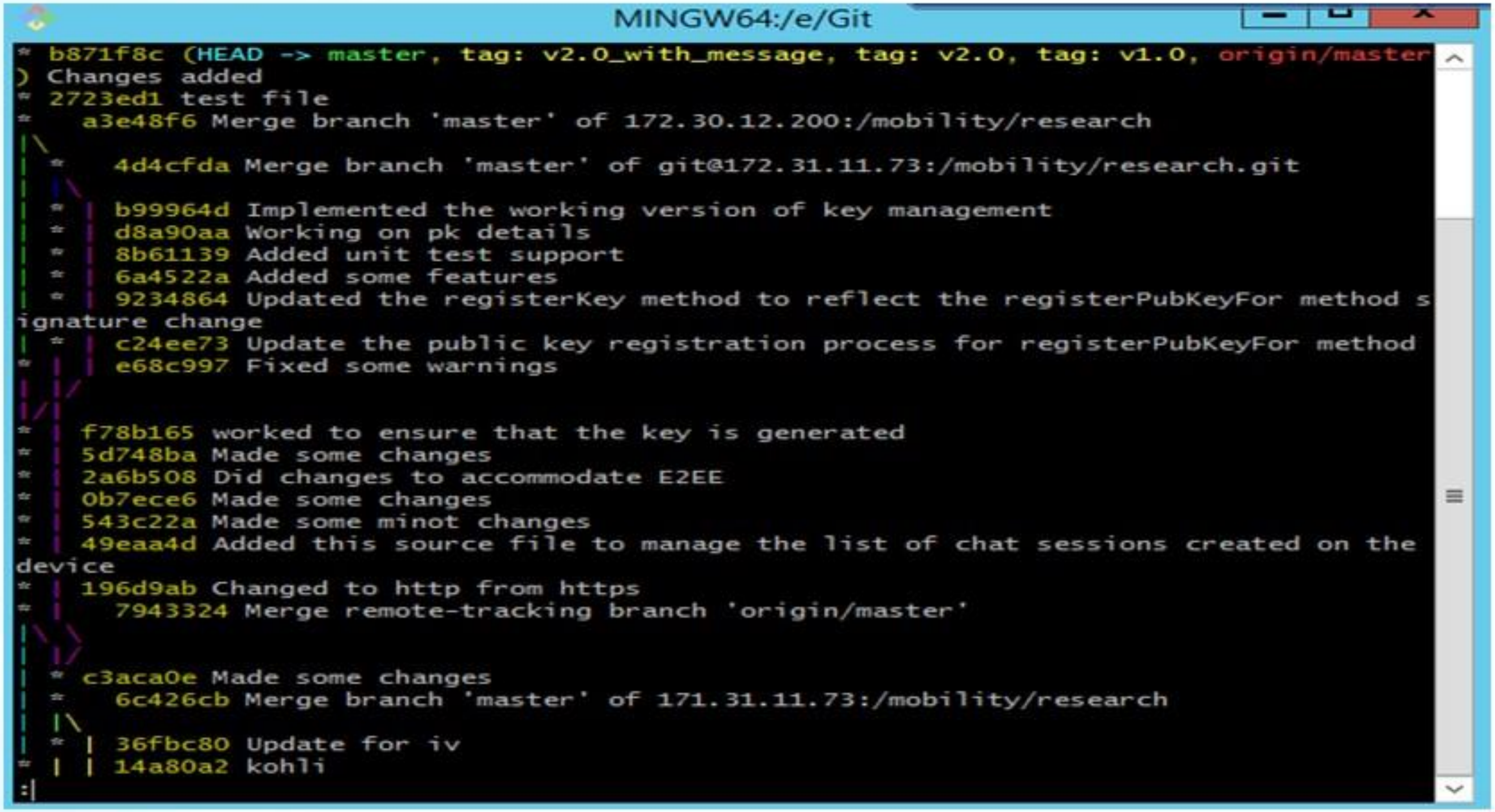
- **List commits on the current branch with graphs**
  - git log --graph --oneline
- **List commits on all the branches with graphs**
  - git log --graph --oneline --all
- **See commits along with labels**
  - git log --graph --oneline --all --decorate

Visualizing Branches



```
$ git log --graph --oneline
* b871f8c (HEAD -> master, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
) Changes added
* 2723ed1 test file
*   a3e48f6 Merge branch 'master' of 172.30.12.200:/mobility/research
|\
| *   4d4cfda Merge branch 'master' of git@172.31.11.73:/mobility/research.git
| |\
| * | b99964d Implemented the working version of key management
| * | d8a90aa Working on pk details
| * | 8b61139 Added unit test support
| * | 6a4522a Added some features
| * | 9234864 Updated the registerKey method to reflect the registerPubKeyFor method s
ignature change
| * | c24ee73 Update the public key registration process for registerPubKeyFor method
* | | e68c997 Fixed some warnings
| |/
|/|
* | f78b165 worked to ensure that the key is generated
* | 5d748ba Made some changes
* | 2a6b508 Did changes to accommodate E2EE
* | 0b7ece6 Made some changes
* | 543c22a Made some minot changes
* | 49eaa4d Added this source file to manage the list of chat sessions created on the
device
* | 196d9ab Changed to http from https
* |   7943324 Merge remote-tracking branch 'origin/master'
|\ \
| |/
| * c3aca0e Made some changes
| *   6c426cb Merge branch 'master' of 171.31.11.73:/mobility/research
| |\
| * | 36fbc80 Update for iv
* | | 14a80a2 kohli
```

Visualizing Branches



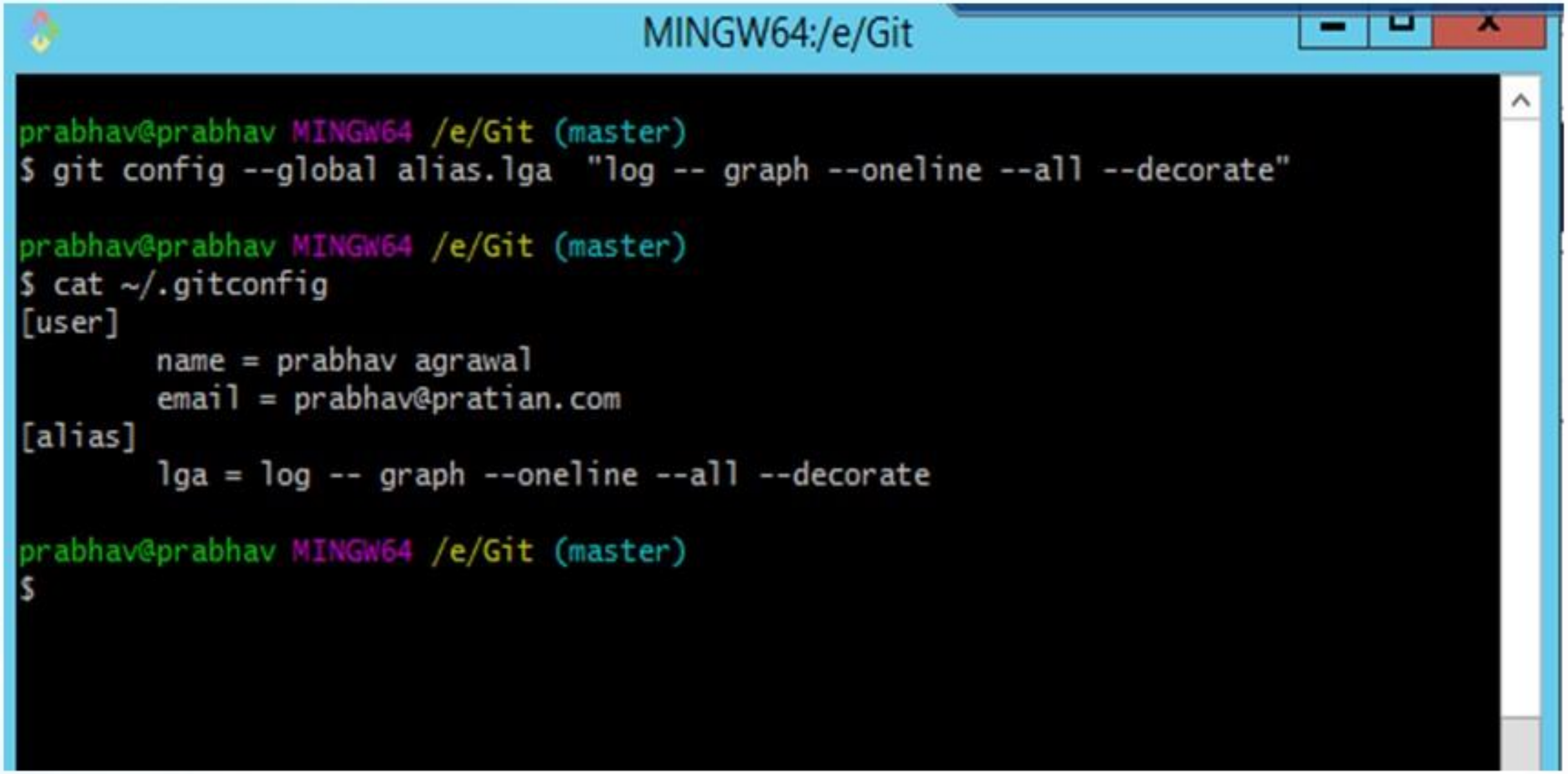
MINGW64:/e/Git

```
* b871f8c (HEAD -> master, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
) Changes added
* 2723ed1 test file
*   a3e48f6 Merge branch 'master' of 172.30.12.200:/mobility/research
|\
| *   4d4cfda Merge branch 'master' of git@172.31.11.73:/mobility/research.git
| |\
| * | b99964d Implemented the working version of key management
| * | d8a90aa Working on pk details
| * | 8b61139 Added unit test support
| * | 6a4522a Added some features
| * | 9234864 Updated the registerKey method to reflect the registerPubKeyFor method s
ignature change
| * | c24ee73 Update the public key registration process for registerPubKeyFor method
* | | e68c997 Fixed some warnings
| |/
|/|
* | f78b165 worked to ensure that the key is generated
* | 5d748ba Made some changes
* | 2a6b508 Did changes to accommodate E2EE
* | 0b7ece6 Made some changes
* | 543c22a Made some minot changes
* | 49eaa4d Added this source file to manage the list of chat sessions created on the
device
* | 196d9ab Changed to http from https
* |   7943324 Merge remote-tracking branch 'origin/master'
|\ \
| |/
| * c3aca0e Made some changes
| *   6c426cb Merge branch 'master' of 171.31.11.73:/mobility/research
| |\
| * | 36fbc80 Update for iv
* | | 14a80a2 kohli
:
```

Adding Alias

- **Add an alias for logs**
  - git config --global alias.lga “log --graph --oneline --all –decorate”
  - Here lga is the name of alias
- **Use alias**
  - git lga
  - You will notice that it does exactly the same thing
- **Let’s see how this is configured in gitconfig**
  - Cat ~/.gitconfig
  - You will see at the bottom that there is new alias

Adding Alias

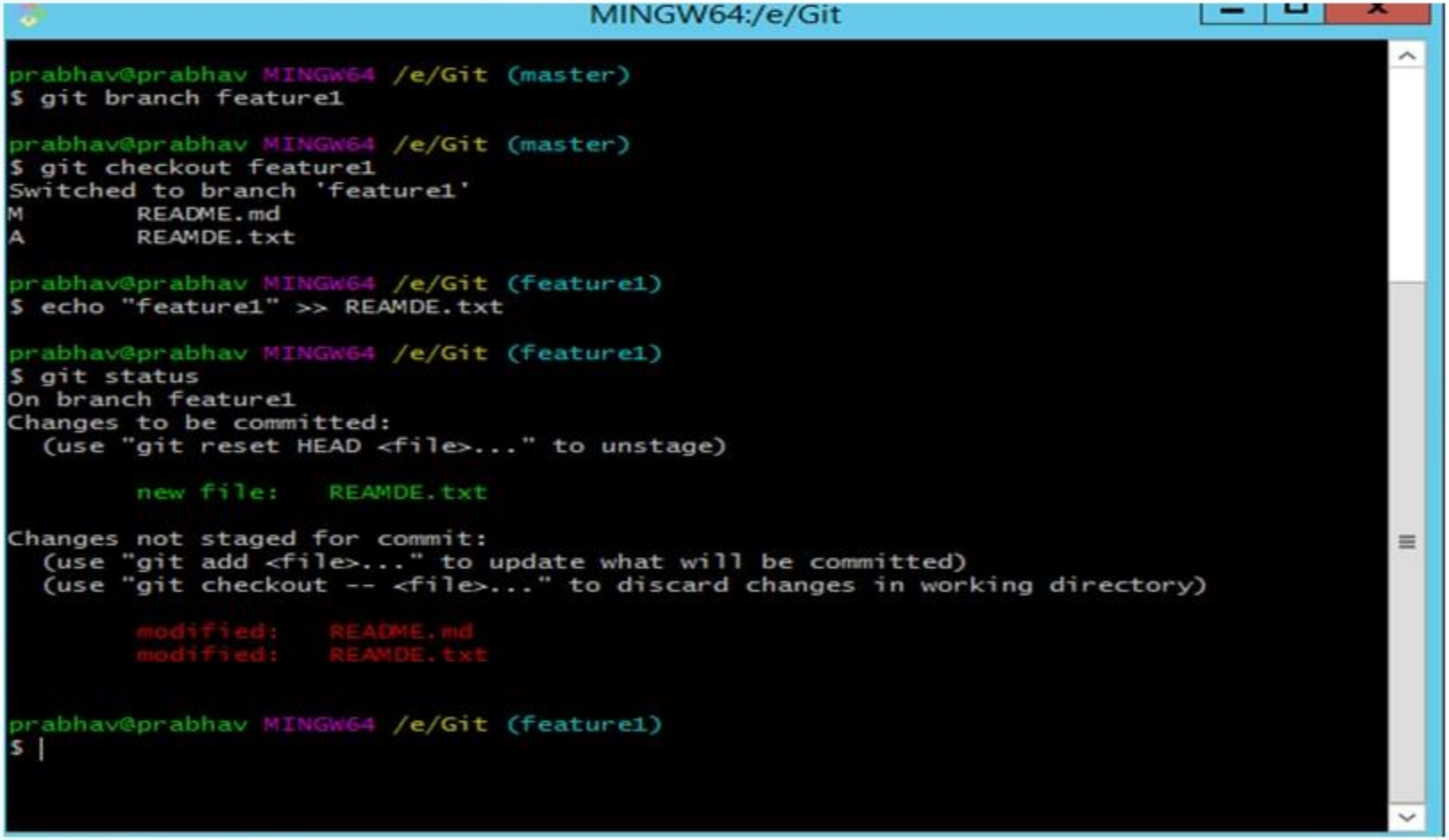


```
prabhav@prabhav MINGW64 /e/Git (master)
$ git config --global alias.lga  "log -- graph --oneline --all --decorate"

prabhav@prabhav MINGW64 /e/Git (master)
$ cat ~/.gitconfig
[user]
        name = prabhav agrawal
        email = prabhav@pratian.com
[alias]
        lga = log -- graph --oneline --all --decorate

prabhav@prabhav MINGW64 /e/Git (master)
$
```

Creating Local Branches

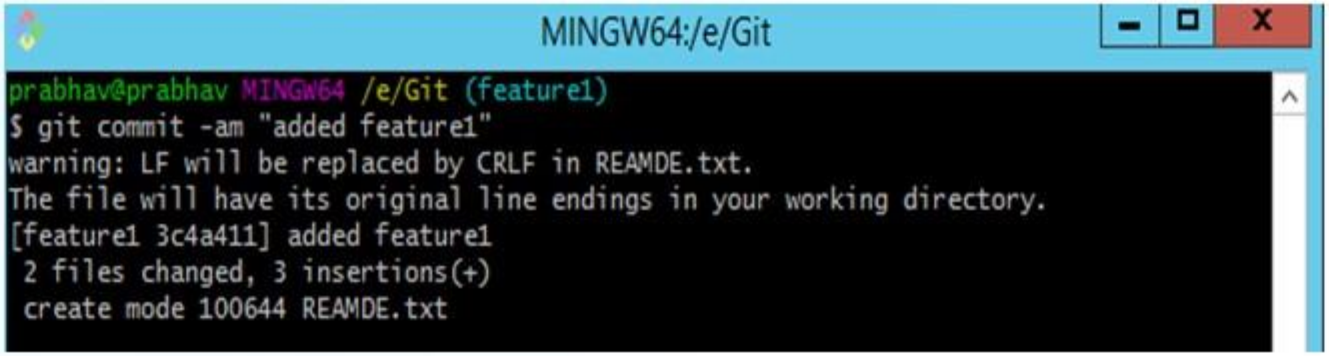
- **Create a local branch**
  - Git branch <branch name>
- **Checkout the newly created branch**
  - git checkout <branch name>
- **Check the logs**
  - Git lga
  - You will notice that there are two branches. Both points to the same commit.
- **Make some modifications.**
  - Echo “abcd” >> file1.txt
- **Check the status**
  - git status
- **Commit the change**
  - git commit –am “message”
- **Check the logs**
  - Git lga
  - You will notice that the new branch is now pointing to a different commit

Creating Local Branches

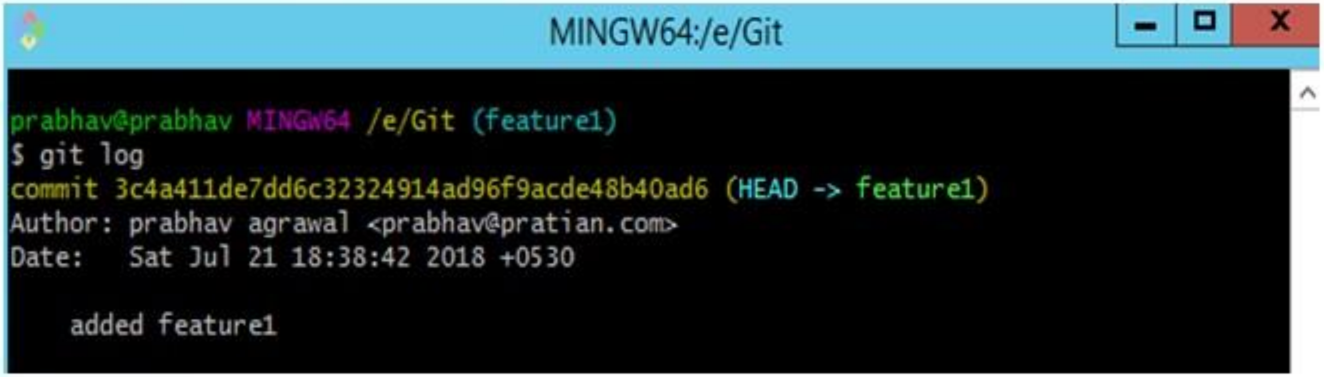


```
prabhav@prabhav MINGW64 /e/Git (master)
$ git branch feature1

prabhav@prabhav MINGW64 /e/Git (master)
$ git checkout feature1
Switched to branch 'feature1'
M       README.md
A       REAMDE.txt

prabhav@prabhav MINGW64 /e/Git (feature1)
$ echo "feature1" >> REAMDE.txt

prabhav@prabhav MINGW64 /e/Git (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   REAMDE.txt

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
        modified:   REAMDE.txt


prabhav@prabhav MINGW64 /e/Git (feature1)
$ |
```

Creating Local Branches



```
prabhav@prabhav MINGW64 /e/Git (feature1)
$ git commit -am "added feature1"
warning: LF will be replaced by CRLF in REAMDE.txt.
The file will have its original line endings in your working directory.
[feature1 3c4a411] added feature1
 2 files changed, 3 insertions(+)
 create mode 100644 REAMDE.txt
```



```
prabhav@prabhav MINGW64 /e/Git (feature1)
$ git log
commit 3c4a411de7dd6c32324914ad96f9acde48b40ad6 (HEAD -> feature1)
Author: prabhav agrawal <prabhav@pratian.com>
Date:   Sat Jul 21 18:38:42 2018 +0530

    added feature1
```

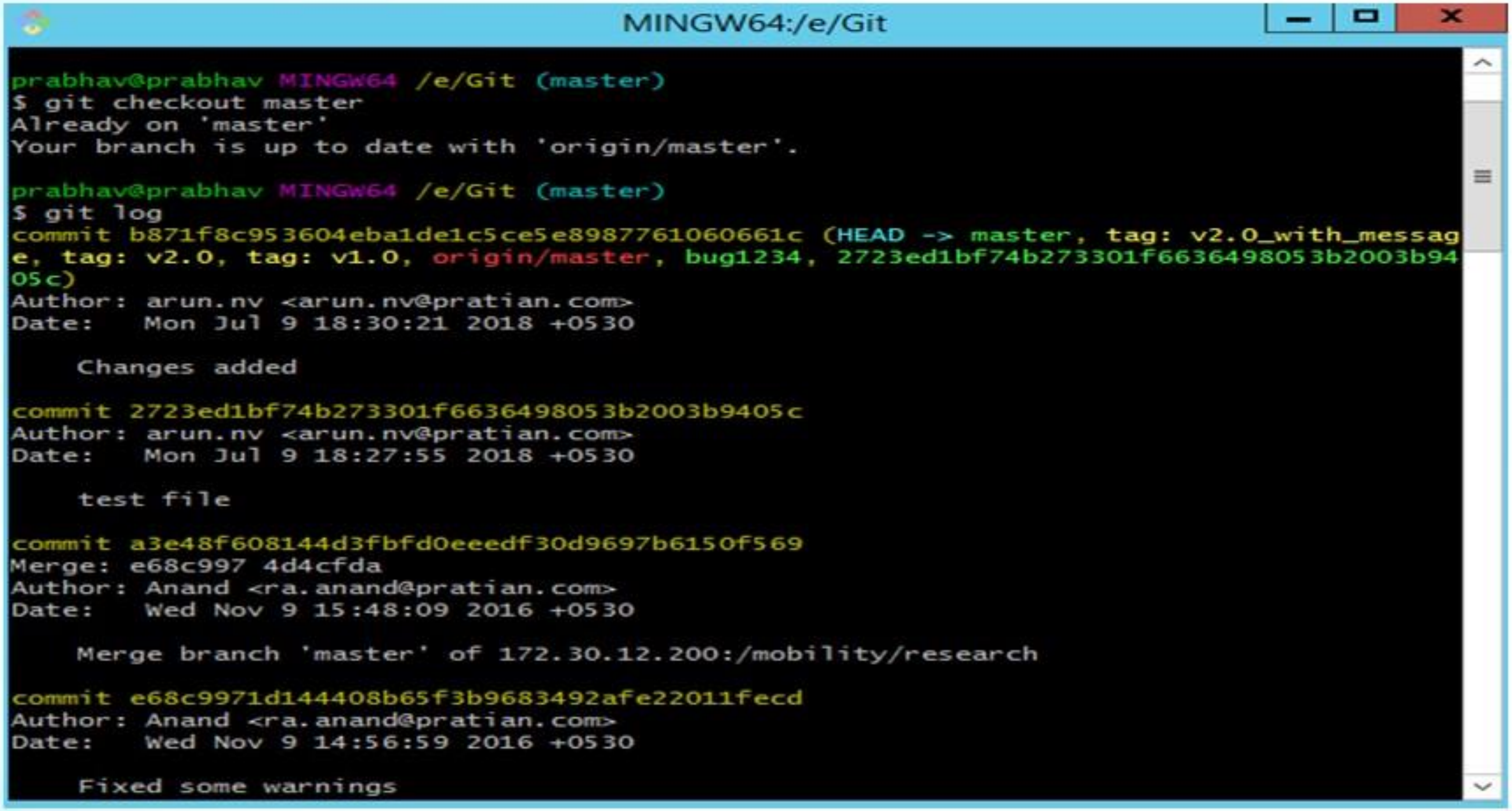
Difference between branches and tags

- **Branches will follow the commits as you make additional commits to the branch. Branch will move along**
- **Checkout the master**
  - git checkout master
- **Check the logs**
  - git lga
  - You will see HEAD is pointing to a previous commit
- **Let's create a branch from a commit**
  - Git branch fix1 <commit number>
- **Checkout the bug fix branch**
  - git checkout fix1
- **Make some modification**
  - Echo "fixing bugs" >> file1.txt
- **Commit the changes**
  - git commit –am "fixed bug123"
- **Check the logs**
  - git lga
  - You will see a different branching structure

Renaming and Deleting branch

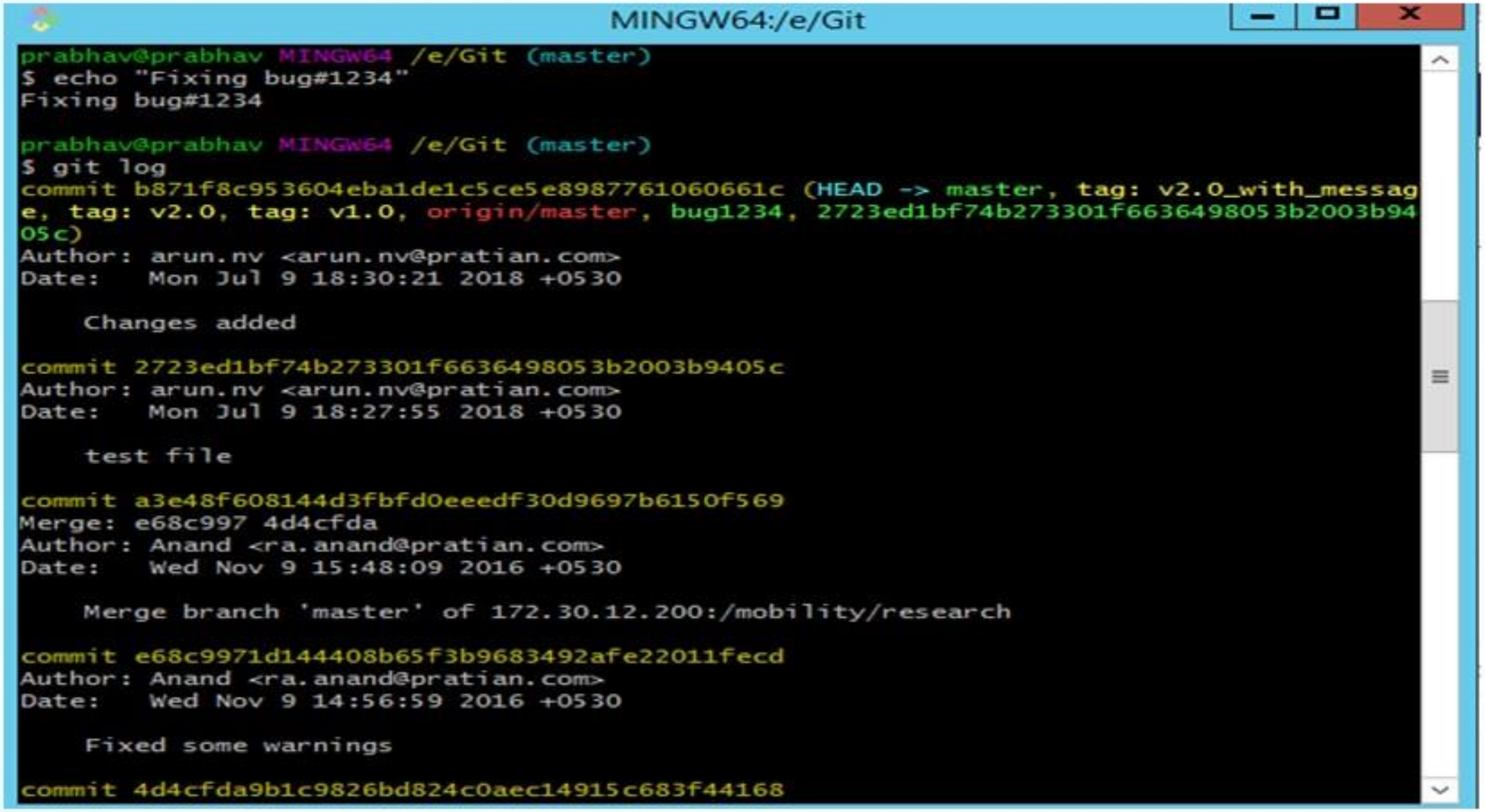
- **Checkout the master branch**
  - Git checkout master
- **Now suppose I want to rename the branch “fix1” to “bug#1234”**
  - Git branch –m fix1 bug1234
- **Check the logs**
  - git lga
  - You will see that we have easily renamed the branch
- **Suppose if you want to delete branch bug1234**
  - Git branch –d bug1234
  - You will see that git prevents you from doing this and it will say the branch isn’t merged.
  - It will say so because we are actually going to loose that commit.
- **Suppose if you want to force delete**
  - Git branch –D bug1234
- **Check the logs**
  - git lga

Renaming and Deleting branch



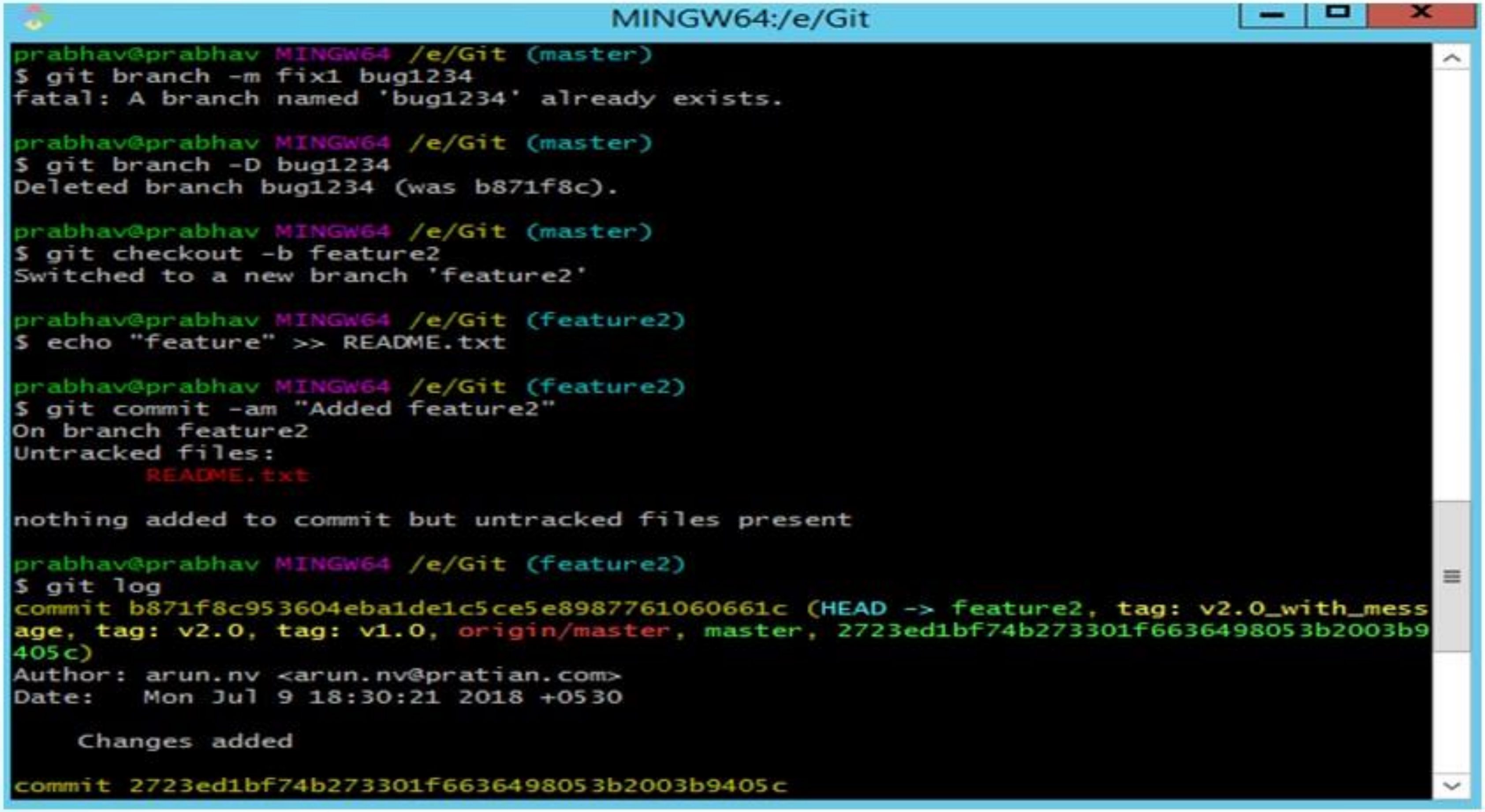
```
prabhav@prabhav MINGW64 /e/Git (master)
$ git checkout master
Already on 'master'
Your branch is up to date with 'origin/master'.

prabhav@prabhav MINGW64 /e/Git (master)
$ git log
commit b871f8c953604eba1de1c5ce5e8987761060661c (HEAD -> master, tag: v2.0_with_messag
e, tag: v2.0, tag: v1.0, origin/master, bug1234, 2723ed1bf74b273301f663649805 3b2003b94
05c)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530

    Changes added

commit 2723ed1bf74b273301f6636498053b2003b9405c
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:27:55 2018 +0530

    test file

commit a3e48f608144d3fbfd0eeedf30d9697b6150f569
Merge: e68c997 4d4cfda
Author: Anand <ra.anand@pratian.com>
Date:   Wed Nov 9 15:48:09 2016 +0530

    Merge branch 'master' of 172.30.12.200:/mobility/research

commit e68c9971d144408b65f3b9683492afe22011fecd
Author: Anand <ra.anand@pratian.com>
Date:   Wed Nov 9 14:56:59 2016 +0530

    Fixed some warnings
```

Renaming and Deleting branch



```
prabhav@prabhav MINGW64 /e/Git (master)
$ echo "Fixing bug#1234"
Fixing bug#1234

prabhav@prabhav MINGW64 /e/Git (master)
$ git log
commit b871f8c953604eba1de1c5ce5e8987761060661c (HEAD -> master, tag: v2.0_with_messag
e, tag: v2.0, tag: v1.0, origin/master, bug1234, 2723ed1bf74b273301f663649805 3b2003b94
05c)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530

    Changes added

commit 2723ed1bf74b273301f6636498053b2003b9405c
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:27:55 2018 +0530

    test file

commit a3e48f608144d3fbfd0eeedf30d9697b6150f569
Merge: e68c997 4d4cfda
Author: Anand <ra.anand@pratian.com>
Date:   Wed Nov 9 15:48:09 2016 +0530

    Merge branch 'master' of 172.30.12.200:/mobility/research

commit e68c9971d144408b65f3b9683492afe22011fecd
Author: Anand <ra.anand@pratian.com>
Date:   Wed Nov 9 14:56:59 2016 +0530

    Fixed some warnings

commit 4d4cfda9b1c9826bd824c0aec14915c683f44168
```

Renaming and Deleting branch



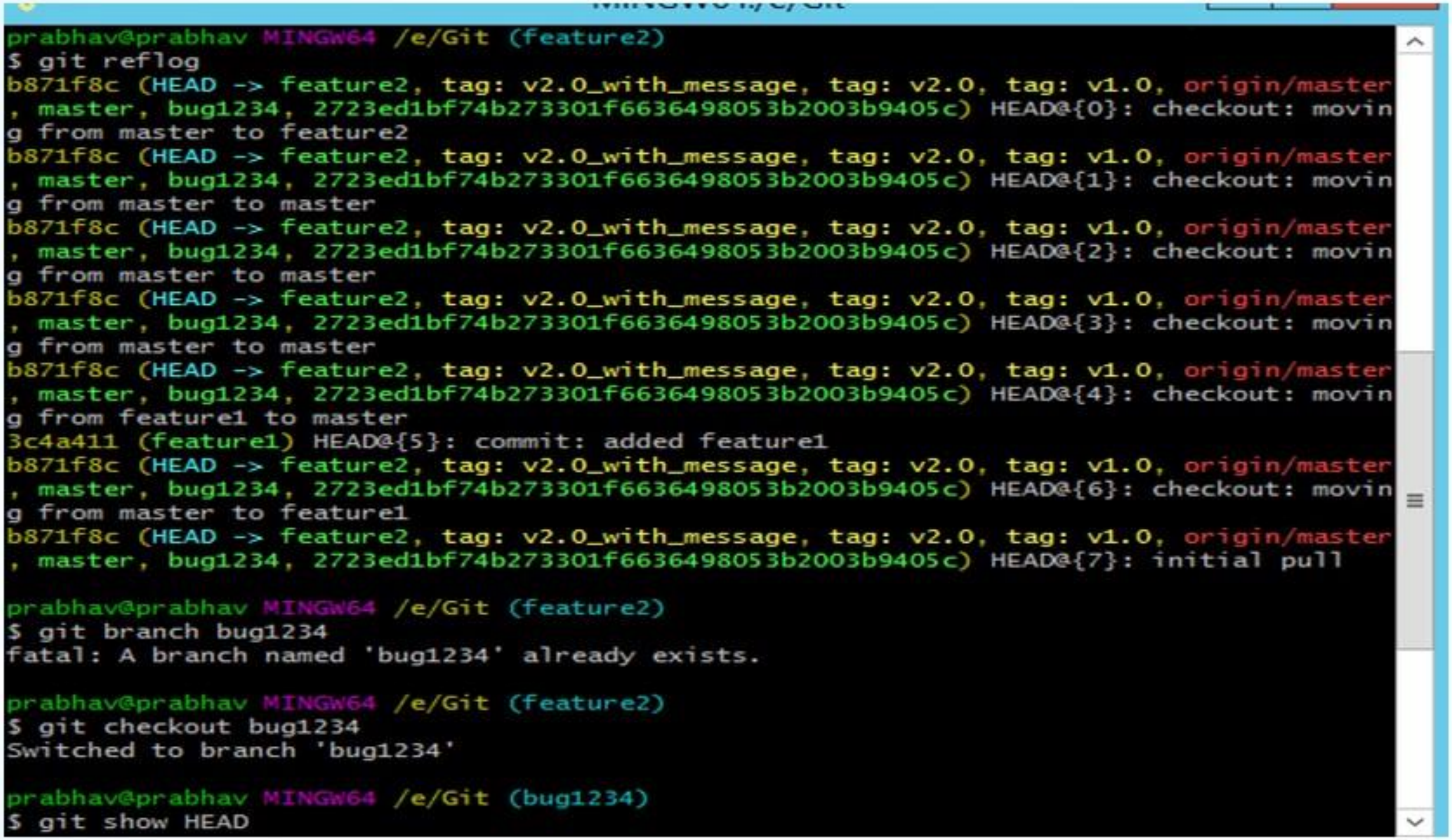
```
prabhav@prabhav MINGW64 /e/Git (master)
$ git branch -m fix1 bug1234
fatal: A branch named 'bug1234' already exists.

prabhav@prabhav MINGW64 /e/Git (master)
$ git branch -D bug1234
Deleted branch bug1234 (was b871f8c).

prabhav@prabhav MINGW64 /e/Git (master)
$ git checkout -b feature2
Switched to a new branch 'feature2'

prabhav@prabhav MINGW64 /e/Git (feature2)
$ echo "feature" >> README.txt

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git commit -am "Added feature2"
On branch feature2
Untracked files:
        README.txt

nothing added to commit but untracked files present

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git log
commit b871f8c953604eba1de1c5ce5e8987761060661c (HEAD -> feature2, tag: v2.0_with_mess
age, tag: v2.0, tag: v1.0, origin/master, master, 2723ed1bf74b273301f663649805 3b2003b9
405c)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530

    Changes added

commit 2723ed1bf74b273301f663649805 3b2003b9405c
```

Recovering deleted commits

- **Suppose if you want to recover the deleted branch bug#1234**
- **Check log of all references where HEAD has pointed**
  - git reflog
- **Find the commit which was done to fix "bug#1234"**
  - Git branch bug1234 <commit number>
  - Commit number should be the one where HEAD changed for bug1234 branch.
- **Check the logs**
  - git lga
  - You will see that branch bug1234 is back

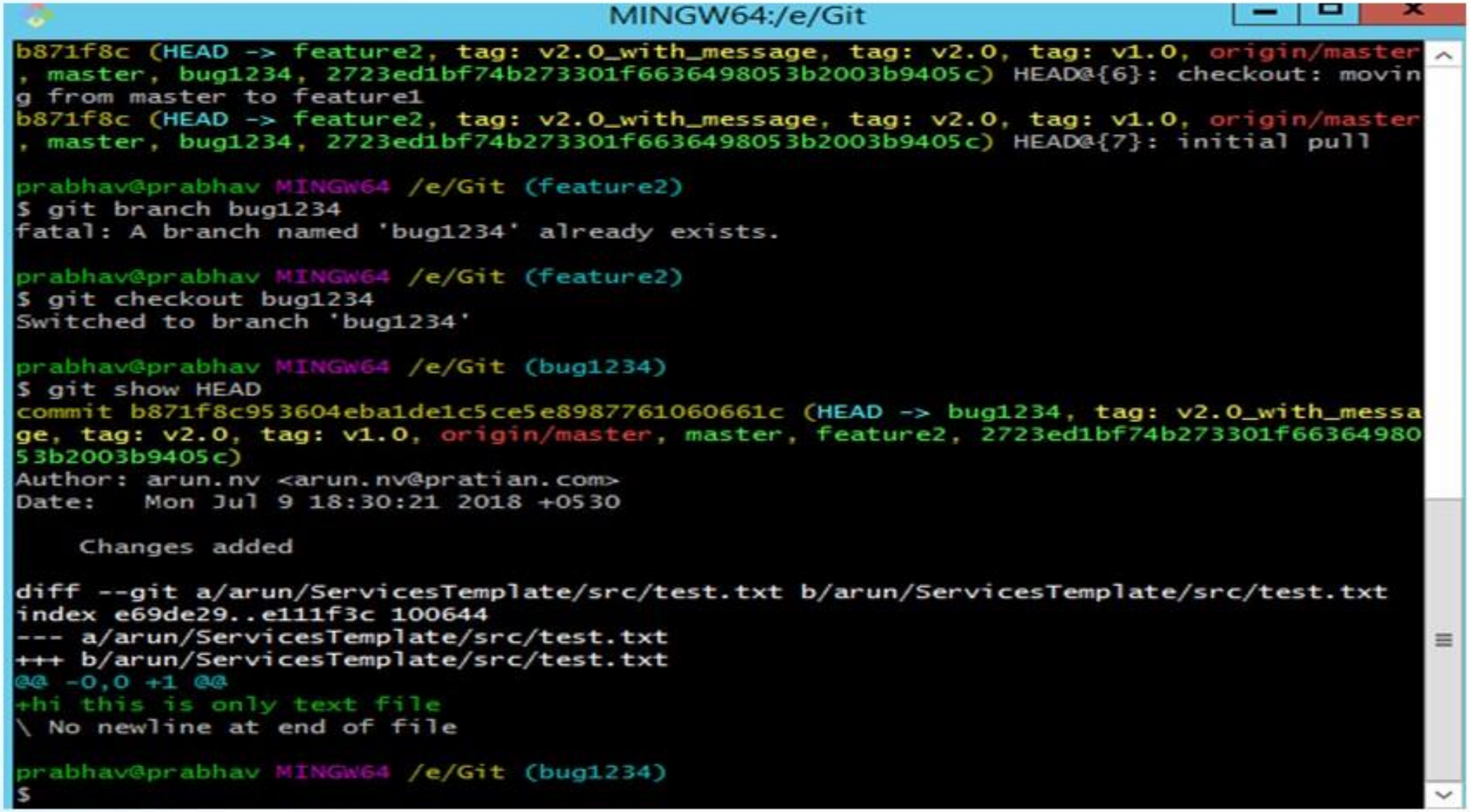
SkillAssure

Recovering Deleted Commits



```
prabhav@prabhav MINGW64 /e/Git (feature2)
$ git reflog
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{0}: checkout: movin
g from master to feature2
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{1}: checkout: movin
g from master to master
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{2}: checkout: movin
g from master to master
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{3}: checkout: movin
g from master to master
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{4}: checkout: movin
g from feature1 to master
3c4a411 (feature1) HEAD@{5}: commit: added feature1
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{6}: checkout: movin
g from master to feature1
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f6636498053b2003b9405c) HEAD@{7}: initial pull

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git branch bug1234
fatal: A branch named 'bug1234' already exists.

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git checkout bug1234
Switched to branch 'bug1234'

prabhav@prabhav MINGW64 /e/Git (bug1234)
$ git show HEAD
```

Recovering Deleted Commits



```
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f663649805 3b2003b9405c) HEAD@{6}: checkout: movin
g from master to feature1
b871f8c (HEAD -> feature2, tag: v2.0_with_message, tag: v2.0, tag: v1.0, origin/master
, master, bug1234, 2723ed1bf74b273301f663649805 3b2003b9405c) HEAD@{7}: initial pull

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git branch bug1234
fatal: A branch named 'bug1234' already exists.

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git checkout bug1234
Switched to branch 'bug1234'

prabhav@prabhav MINGW64 /e/Git (bug1234)
$ git show HEAD
commit b871f8c953604eba1de1c5ce5e8987761060661c (HEAD -> bug1234, tag: v2.0_with_messa
ge, tag: v2.0, tag: v1.0, origin/master, master, feature2, 2723ed1bf74b273301f66364980
53b2003b9405c)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530

    Changes added

diff --git a/arun/ServicesTemplate/src/test.txt b/arun/ServicesTemplate/src/test.txt
index e69de29..e111f3c 100644
--- a/arun/ServicesTemplate/src/test.txt
+++ b/arun/ServicesTemplate/src/test.txt
@@ -0,0 +1 @@
+hi this is only text file
\ No newline at end of file

prabhav@prabhav MINGW64 /e/Git (bug1234)
$
```

SkillAssure

Stashing changes

- **Let's say I am working on a branch feature2. I will checkout it first.**
  - git checkout feature2
- **I made some changes in feature2 branch**
  - Echo "modifying feature2" >> README.txt
- **Check the status**
  - git status
- **Someone come along with a bug report of production and say I want it fixed immediately.**
- **The changes which I made is not ready to be checked in to feature 2. I also do not want to loose my work.**
- **I can save this using git stash**
  - git stash
  - You will see that it saved the working copy
- **Check the status**
  - git status
  - You will see that there is nothing to commit

Stashing changes

- **See the list of git stash.**
  - git stash list

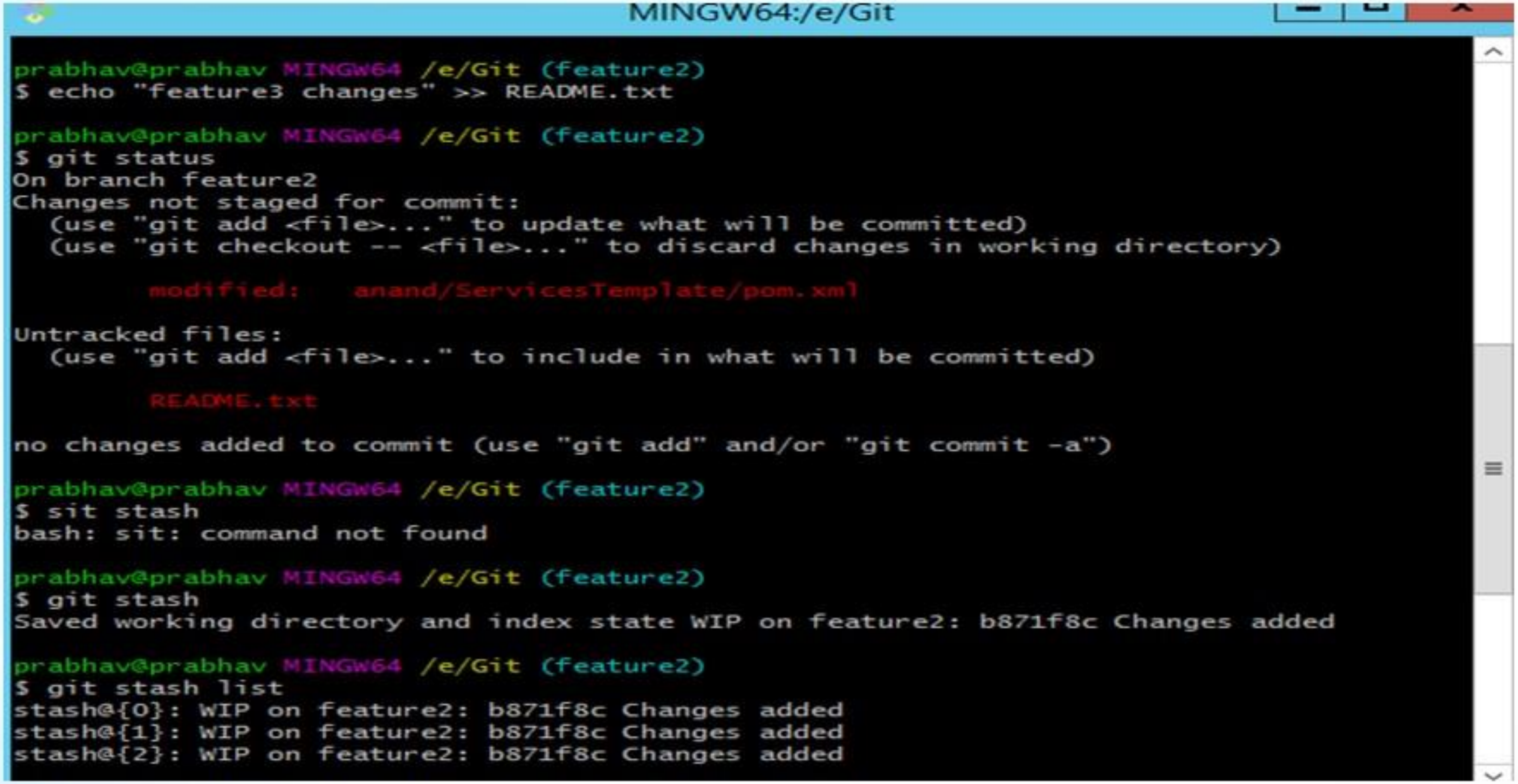
- **Fix the production Bug.**
  - git checkout bug1234
  - Echo “Production fix on 1234” >> README.txt
  - git commit –am “Fixing bug1234”

- **Checkout Feature2 branch**
  - git checkout feature2

Stashing changes

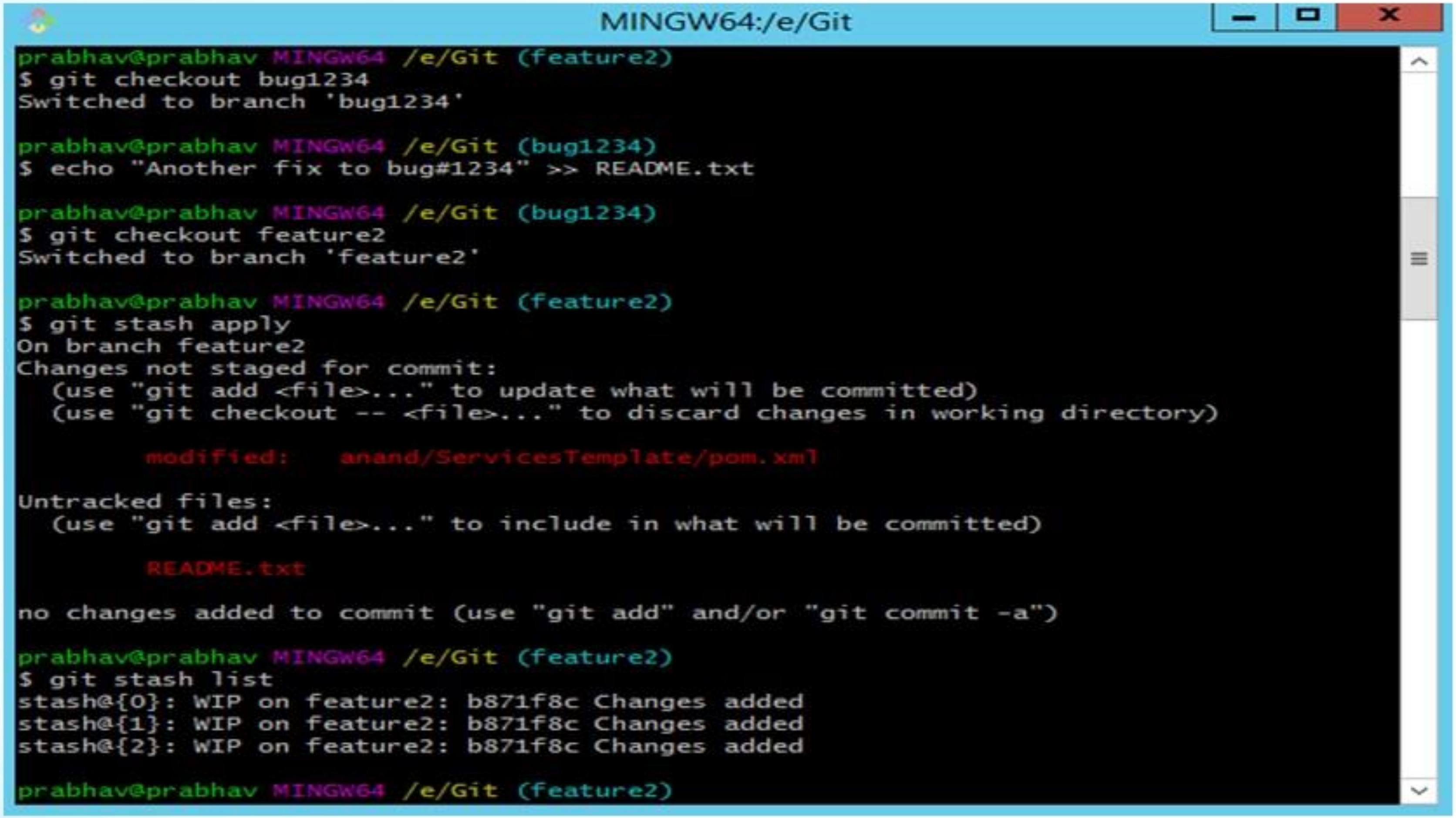
- **Reapply the stash changes**
  - git stash apply
- **Check README.txt**
  - cat README.txt
  - You will see that the changes you have kept it on stash is back again.
- **See the list of git stash.**
  - git stash list
- **Now if you think that you will need to create another branch from the stash.**
  - Git stash branch <branch name>
- **Reapply the stash changes and remove it from the stash list**
  - git stash pop
- **Commit the changes**
  - git commit –am “added additional feature branch”

Stashing Changes



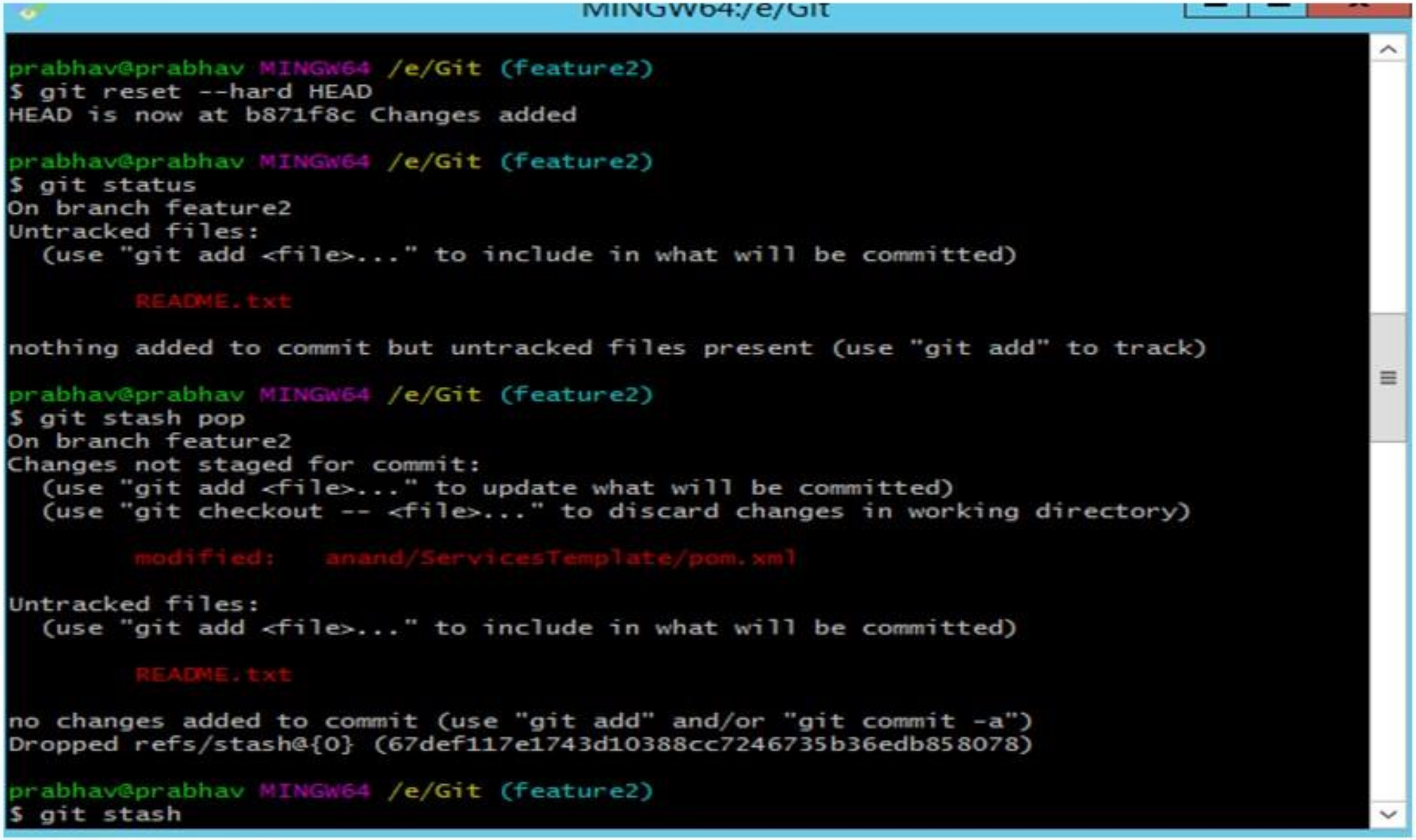
```
prabhav@prabhav MINGW64 /e/Git (feature2)
$ echo "feature3 changes" >> README.txt

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git status
On branch feature2
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:   anand/ServicesTemplate/pom.xml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.txt

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

prabhav@prabhav MINGW64 /e/Git (feature2)
$ sit stash
bash: sit: command not found

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash
Saved working directory and index state WIP on feature2: b871f8c Changes added

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash list
stash@{0}: WIP on feature2: b871f8c Changes added
stash@{1}: WIP on feature2: b871f8c Changes added
stash@{2}: WIP on feature2: b871f8c Changes added
```

Stashing Changes



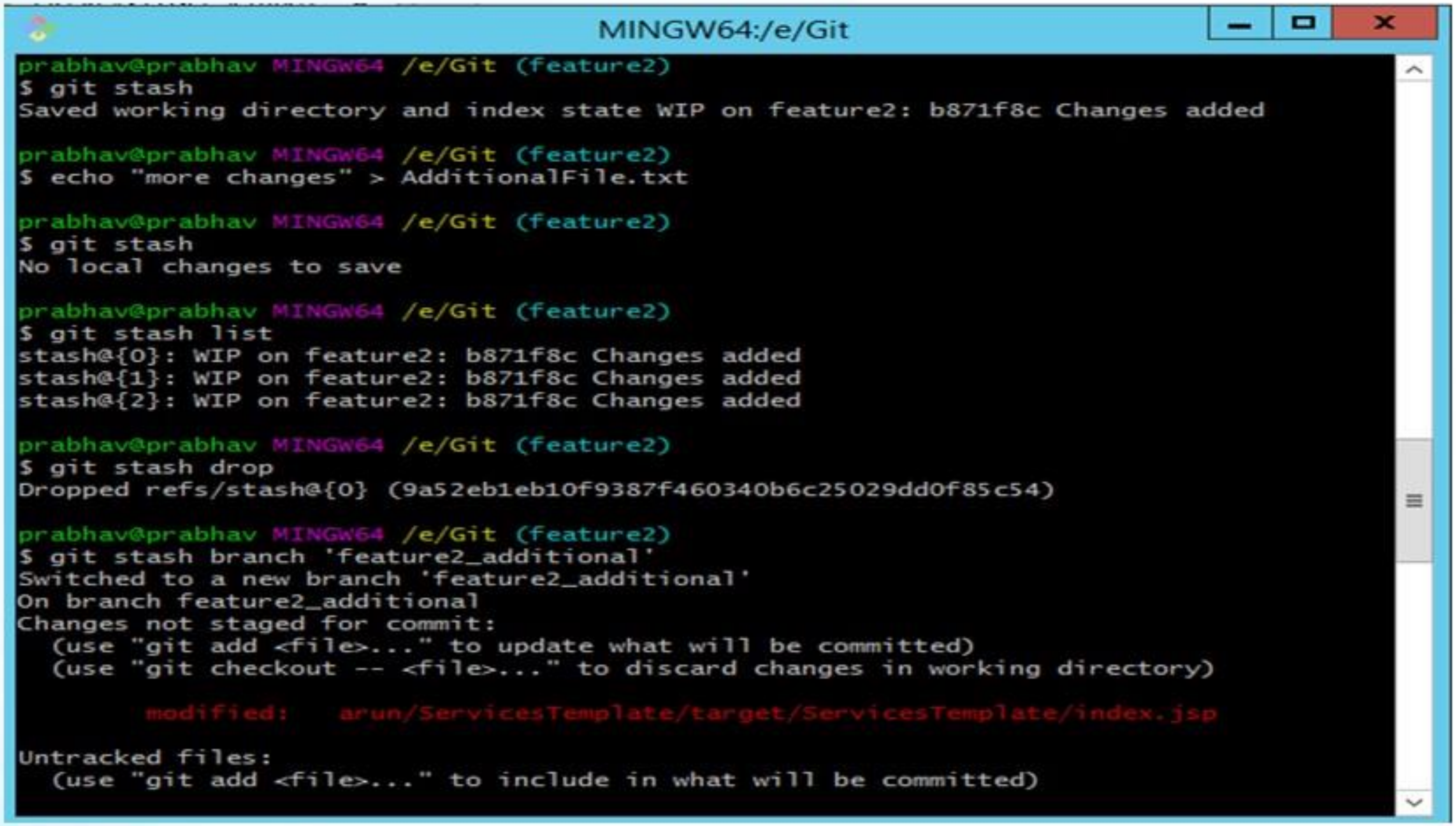
```
prabhav@prabhav MINGW64 /e/Git (feature2)
$ git checkout bug1234
Switched to branch 'bug1234'

prabhav@prabhav MINGW64 /e/Git (bug1234)
$ echo "Another fix to bug#1234" >> README.txt

prabhav@prabhav MINGW64 /e/Git (bug1234)
$ git checkout feature2
Switched to branch 'feature2'

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash apply
On branch feature2
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:   anand/ServicesTemplate/pom.xml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.txt

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

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash list
stash@{0}: WIP on feature2: b871f8c Changes added
stash@{1}: WIP on feature2: b871f8c Changes added
stash@{2}: WIP on feature2: b871f8c Changes added

prabhav@prabhav MINGW64 /e/Git (feature2)
```

Stashing Changes



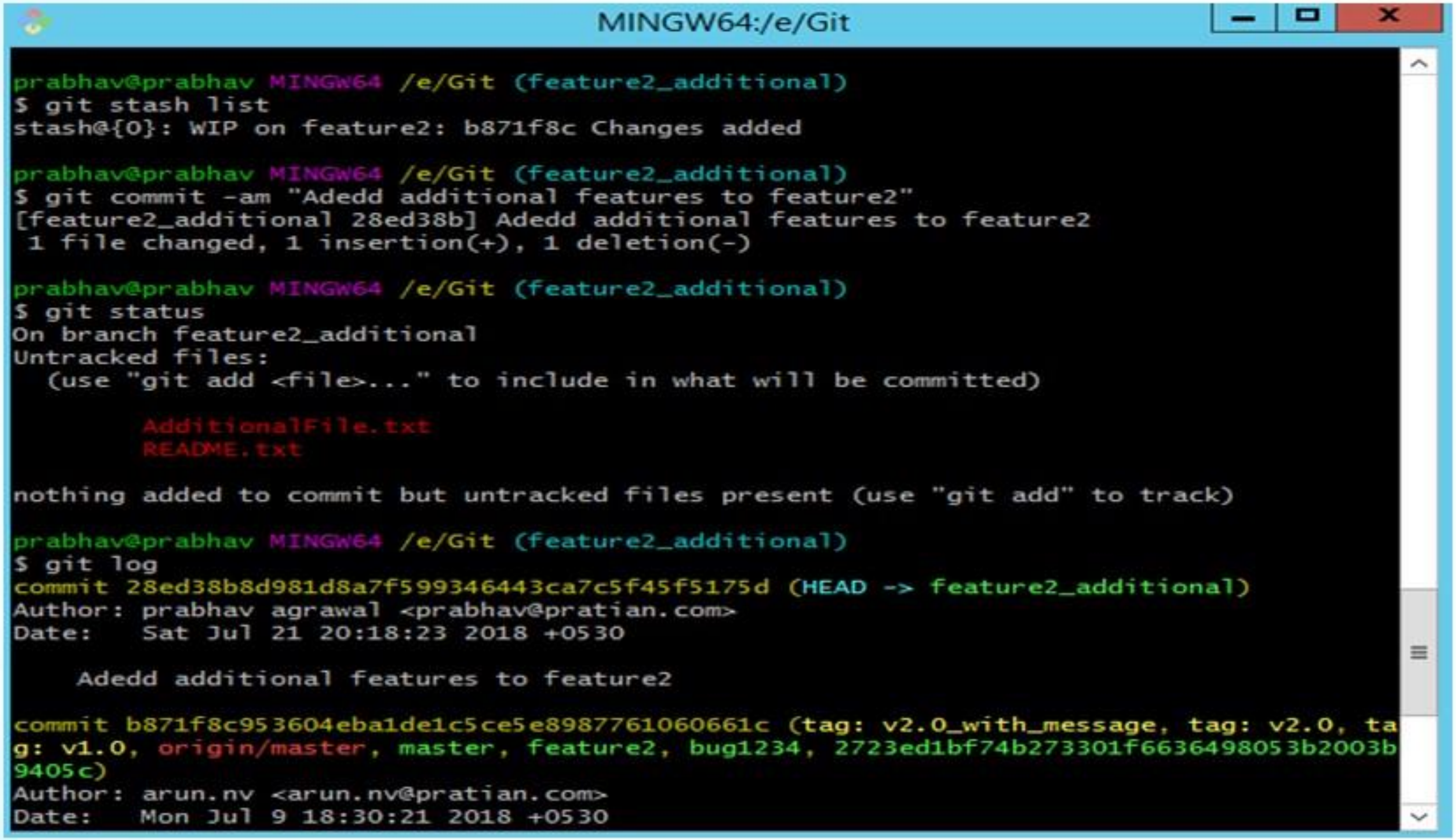
```
prabhav@prabhav MINGW64 /e/Git (feature2)
$ git reset --hard HEAD
HEAD is now at b871f8c Changes added

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git status
On branch feature2
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.txt

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

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash pop
On branch feature2
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:   anand/ServicesTemplate/pom.xml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        README.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (67def117e1743d10388cc7246735b36edb858078)

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash
```

Stashing Changes



```
prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash
Saved working directory and index state WIP on feature2: b871f8c Changes added

prabhav@prabhav MINGW64 /e/Git (feature2)
$ echo "more changes" > AdditionalFile.txt

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash
No local changes to save

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash list
stash@{0}: WIP on feature2: b871f8c Changes added
stash@{1}: WIP on feature2: b871f8c Changes added
stash@{2}: WIP on feature2: b871f8c Changes added

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash drop
Dropped refs/stash@{0} (9a52eb1eb10f9387f460340b6c25029dd0f85c54)

prabhav@prabhav MINGW64 /e/Git (feature2)
$ git stash branch 'feature2_additional'
Switched to a new branch 'feature2_additional'
On branch feature2_additional
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:   arun/ServicesTemplate/target/ServicesTemplate/index.jsp

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

Stashing Changes



```
prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git stash list
stash@{0}: WIP on feature2: b871f8c Changes added

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git commit -am "Adedd additional features to feature2"
[feature2_additional 28ed38b] Adedd additional features to feature2
 1 file changed, 1 insertion(+), 1 deletion(-)

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git status
On branch feature2_additional
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        AdditionalFile.txt
        README.txt

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

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git log
commit 28ed38b8d981d8a7f599346443ca7c5f45f5175d (HEAD -> feature2_additional)
Author: prabhav agrawal <prabhav@pratian.com>
Date:   Sat Jul 21 20:18:23 2018 +0530

    Adedd additional features to feature2

commit b871f8c953604eba1de1c5ce5e8987761060661c (tag: v2.0_with_message, tag: v2.0, ta
g: v1.0, origin/master, master, feature2, bug1234, 2723ed1bf74b273301f663649805 3b2003b
9405c)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530
```

Merging Branches

- **Let's go back to master and we will see merging.**
  - git checkout master
- **Check the logs**
  - git lga
- **Merge master with any other branch**
  - Git merge <branch name>
- **Check the logs**
  - git lga
  - You will see that the branches are now merged.
- **Now if you want to delete the branch which you have already merged with master.**
  - git branch –d <branch name>
- **Check the logs**
  - git lga
  - You will see that the branch is deleted

Rebasing Branches

- **Let’s say we are working on a new feature “feature3” which is based on the tab v1.0.**
  - git branch feature3 v1.0
- **Checkout “feature3”**
  - git checkout feature3
- **Make some changes**
  - Echo “Adding more code” >> file1.txt
- **Commit that change**
  - Git commit –am “Added Feature3”
- **Check the logs**
  - git lga
  - You will see that feature3 is coming from master branch tag v1.0
- **If I want to move “feature3” commit and look like it was always off of master branch.**
  - git rebase master
- **Check the logs**
  - git lga
  - You will see that the feature is just added on top of master.

Rebasing Branches

- Rebasing does not always go as planned

Cherry Picking Changes

- **Create a branch**
  - git branch bug_fixes1.0
- **Checkout the branch**
  - git checkout bug_fixes1.0
- **Make some changes**
  - Echo “Fix1” >> file1.txt
- **Commit that change**
  - Git commit –am “Added Fix1”
- **Make some changes to another file**
  - Echo “Fix2” >> file2.txt
- **Commit that change**
  - Git commit –am “Added Fix2”
- **Check the logs**
  - Git lga

Cherry Picking Changes

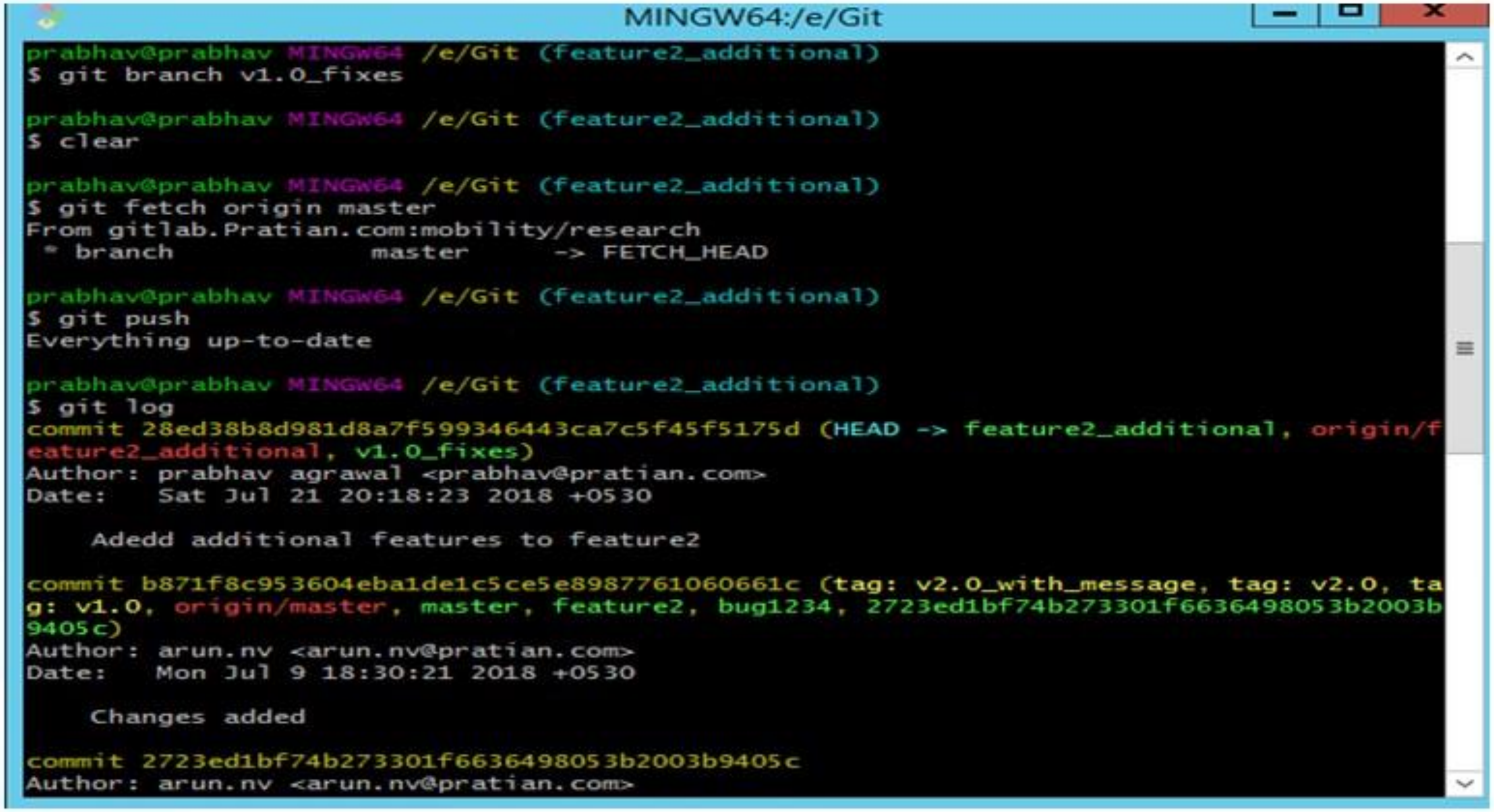
- **Looking at the logs you will see that there are two changes fix 1 and fix2. Now if you want one of these fixes on the master branch**
- **Checkout the master branch**
  - git checkout master
- **Check the logs**
  - Git lga
- **Cherry Pick the changes**
  - Git cherry-pick <commit number>
- **Check the logs**
  - Git lga
  - You will see that only that one change is now present in the master
- **Later if I decide to merge all fixes (i.e. both 1 & 2)**
  - git merge bug_fixes1.0
- **Check the logs**
  - Git lga
  - You will see that earlier commit was already added. It won't apply it again.

Creating a remote branch

- **Push changes to remote branch**
  - git push
- **Check the logs**
  - Git lga
  - You will see that the remote branch “origin/master” is fast forwarded to “master”
- **Check back in the github as well**
  - You will see that changes are updated in remote master branch.
- **Push changes from a local branch to a new remote branch**
  - git push origin <branch name>
  - E.g. git push origin bug_fixes1.0
- **Check back in the github as well**
  - You will see that a new branch is created.
- **Check the Remote branch**
  - git branch -r
  - You will see the branches
- **Push changes from a local branch to remote branch with new name**
  - git push origin <local branch name>:<new remote branch name>
- **Check back in the github as well**
  - You will see that a new branch is created with different name.

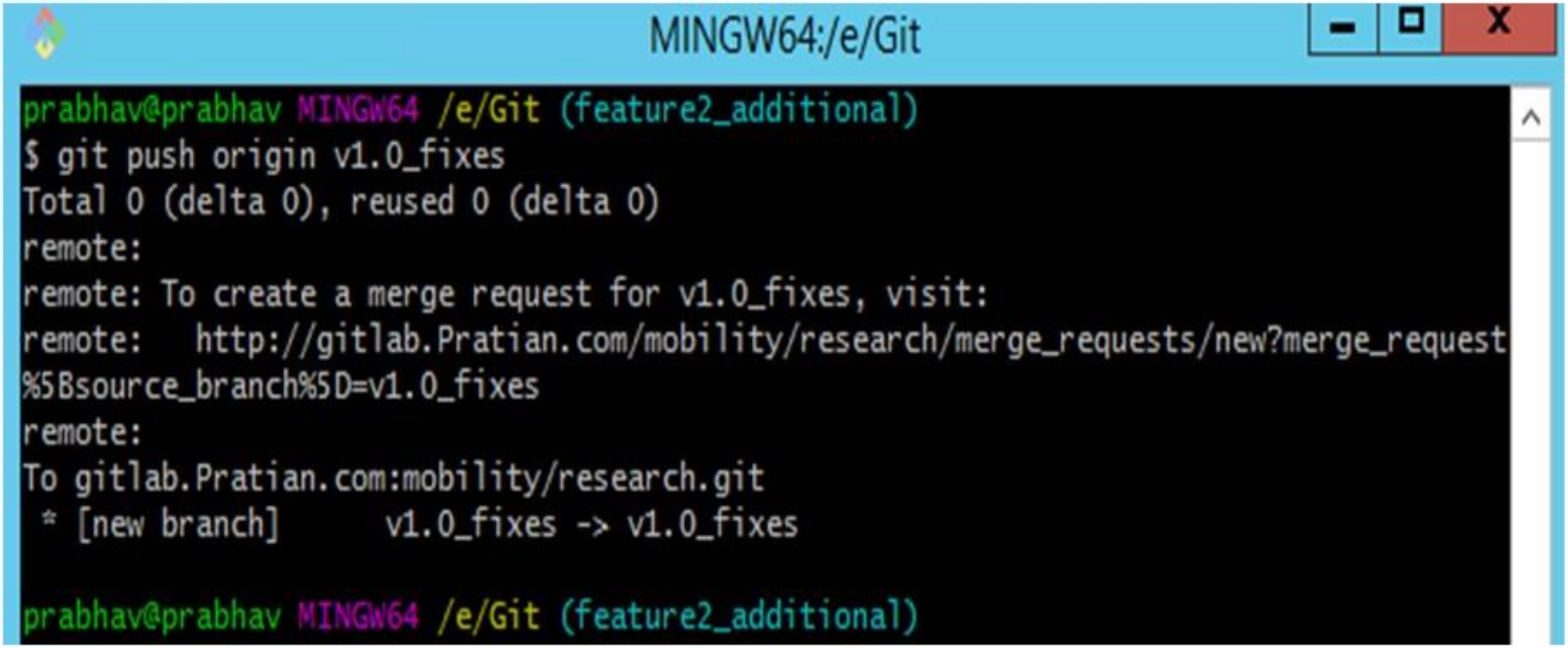
SkillAssure

Creating a Remote Branch



```
prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git branch v1.0_fixes

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ clear

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git fetch origin master
From gitlab.Pratian.com:mobility/research
 * branch            master     -> FETCH_HEAD

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git push
Everything up-to-date

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git log
commit 28ed38b8d981d8a7f599346443ca7c5f45f5175d (HEAD -> feature2_additional, origin/f
eature2_additional, v1.0_fixes)
Author: prabhav agrawal <prabhav@pratian.com>
Date:   Sat Jul 21 20:18:23 2018 +0530

    Adedd additional features to feature2

commit b871f8c953604eba1de1c5ce5e8987761060661c (tag: v2.0_with_message, tag: v2.0, ta
g: v1.0, origin/master, master, feature2, bug1234, 2723ed1bf74b273301f663649805 3b2003b
9405c)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530

    Changes added

commit 2723ed1bf74b273301f663649805 3b2003b9405c
Author: arun.nv <arun.nv@pratian.com>
```

Creating a Remote Branch



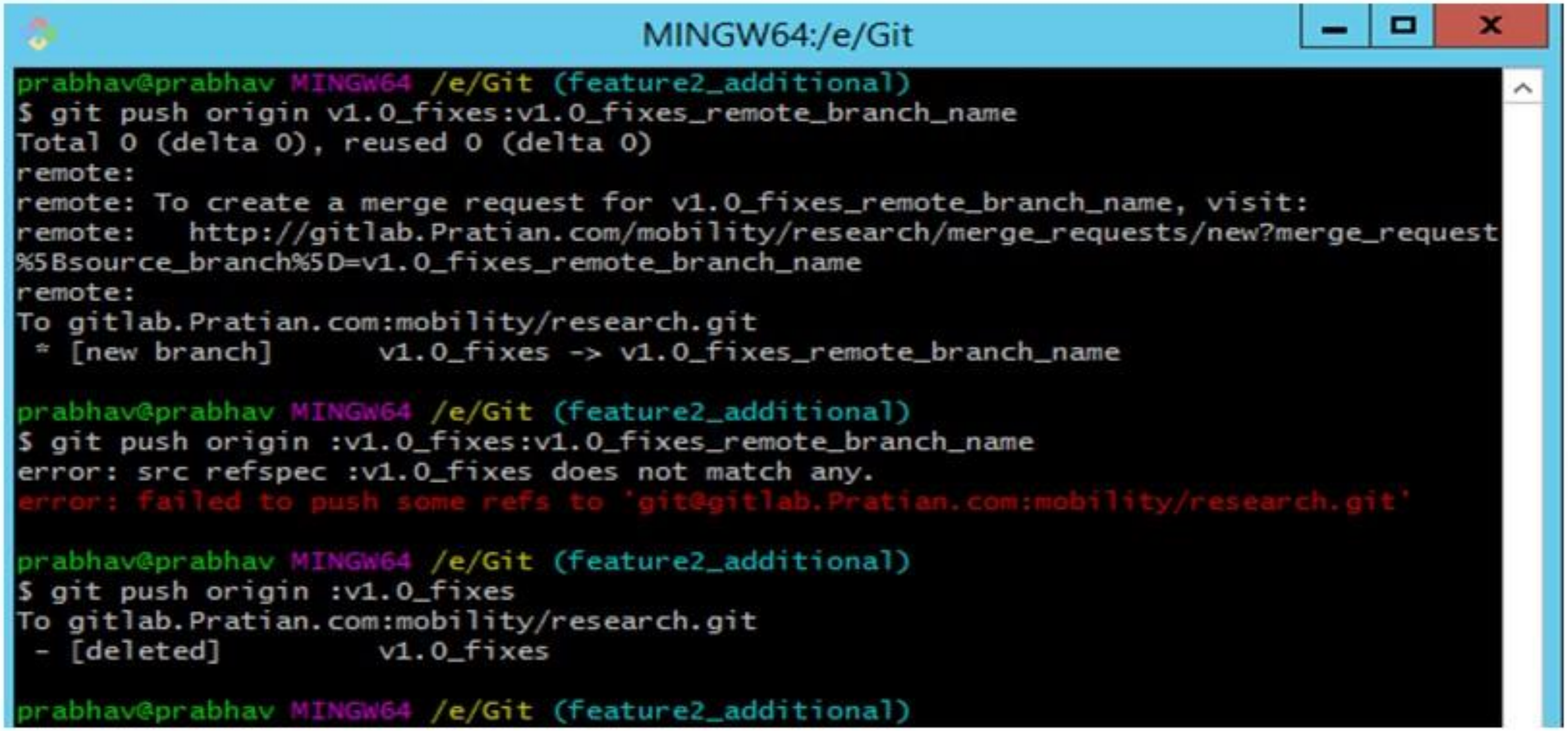
MINGW64:/e/Git

```
prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git push origin v1.0_fixes
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for v1.0_fixes, visit:
remote:   http://gitlab.Pratian.com/mobility/research/merge_requests/new?merge_request
%5Bsource_branch%5D=v1.0_fixes
remote:
To gitlab.Pratian.com:mobility/research.git
 * [new branch]      v1.0_fixes -> v1.0_fixes

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
```

Deleting a remote branch

- **This is something which you have to do with caution.**
- **Let us try deleting remote branch “bug_fixes1.0”**
- **Delete a remote branch.**
  - git push origin **:<remote branch name>**
  - **git push origin :bug_fixes1.0**
- **Check the Remote branch**
  - git branch -r
  - You will see the branches are deleted
- **Check back in the github as well**

Deleting a Remote Branch



```
prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git push origin v1.0_fixes:v1.0_fixes_remote_branch_name
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: To create a merge request for v1.0_fixes_remote_branch_name, visit:
remote:   http://gitlab.Pratian.com/mobility/research/merge_requests/new?merge_request
%5Bsource_branch%5D=v1.0_fixes_remote_branch_name
remote:
To gitlab.Pratian.com:mobility/research.git
 * [new branch]      v1.0_fixes -> v1.0_fixes_remote_branch_name

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git push origin :v1.0_fixes:v1.0_fixes_remote_branch_name
error: src refspec :v1.0_fixes does not match any.
error: failed to push some refs to 'git@gitlab.Pratian.com:mobility/research.git'

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
$ git push origin :v1.0_fixes
To gitlab.Pratian.com:mobility/research.git
 - [deleted]         v1.0_fixes

prabhav@prabhav MINGW64 /e/Git (feature2_additional)
```

Thank You