Working Remotely with GIT

Find a repository

1. Find a repository online
2. Use https://github.com/github/VisualStudio.git
3. Use your created repository

Clone a Remote Repository

git clone https://github.com/github/VisualStudio.git

```
Prabhav.Agrawal@PT-PTL-001 MINGW64 ~/Desktop/Amadeus (master)
$ git clone git@gitlab.prabhav.com:root/Amadeus.git
Cloning into 'Amadeus'...
The authenticity of host 'gitlab.prabhav.com (172.30.11.237)' can't be established.
ECDSA key fingerprint is SHA256:BadczzXYJsqdVNmgnOQi2DG+rAM+PVnXp67qQ3Htw2Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gitlab.prabhav.com' (ECDSA) to the list of known hosts.
Enter passphrase for key '/c/Users/prabhav.agrawal/.ssh/id_rsa':
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.

Prabhav.Agrawal@PT-PTL-001 MINGW64 ~/Desktop/Amadeus
$ |
```

Cloning a Remote Repository
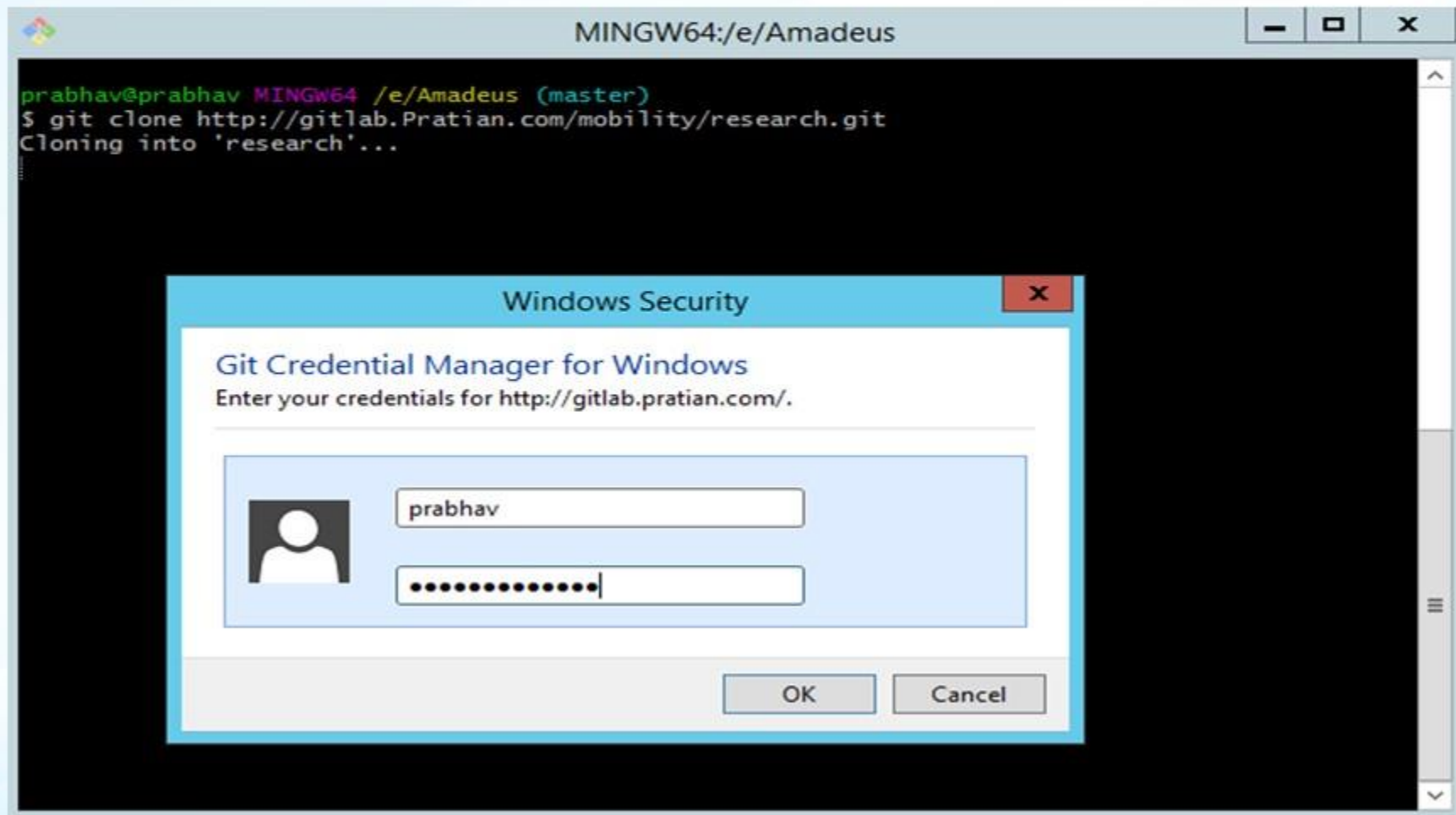
Cloning a Remote Repository


```
MINGW64:/e/Amadeus                                    _  □  X

prabhav@prabhav MINGW64 /e/Amadeus (master)
$ git clone http://gitlab.pratian.com/mobility/research.git
Cloning into 'research'...
remote: Counting objects: 1837, done.
remote: Compressing objects: 100% (987/987), done.
remote: Total 1837 (delta 734), reused 1765 (delta 671)
Receiving objects: 100% (1837/1837), 35.76 MiB | 9.47 MiB/s, done.
Resolving deltas: 100% (734/734), done.
Checking out files: 100% (834/834), done.

prabhav@prabhav MINGW64 /e/Amadeus (master)
$
```
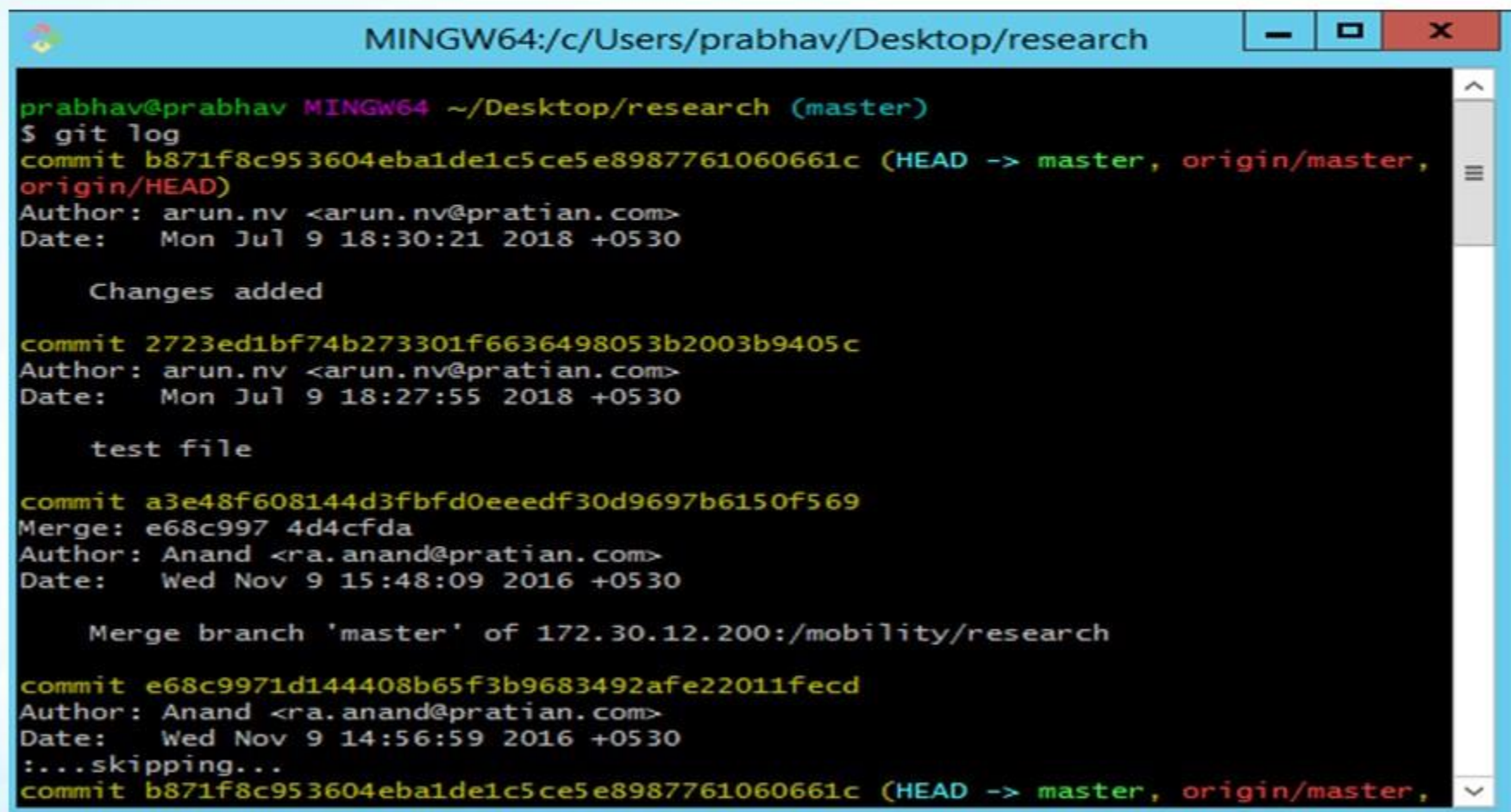
See Repository Statistics

- Let's see the logs
  - git log
- See the commits in online
  - git log --oneline
- See total number of commits
  - git log –oneline | wc –l
- See different branches and merges logs
  - git log –oneline --graph
- Get logs in short format
  - Git log –format=short
  - Git shortlog
  - Both of these will give the same output, i.e. the list of authors who has made the commit with commit messages and number of commits
- Let's see summary of number of commits by each author
- Git shortlog -sne

See Repository Statistics

- -n
- --numbered
- Sort output according to the number of commits per author instead of author alphabetic order.

- -s
- --summary
- Suppress commit description and provide a commit count summary only.

- -e
- --email
- Show the email address of each author.

See Repository Statistics

See Repository Statistics

See Repository Statistics

See Repository Statistics

See Repository Statistics

GitHub Graph

- GitHub provides number of statistics for us.
- Go to the repository
- Click on Graphs
- You will be able to see a lot of graphs based on contributors, code frequency, commit activity, etc.
- Click on Contributors
- Here you will be able to see Contributors graph.

Viewing Commits

- Let's see what was the last commit made
  - git show HEAD
  - git log
- See any specific commit
  - Git show <number>
- See the remote fetch and push URL
  - Git remote –v
    - V stands for verbose
- See the URLs, Do they look similar?
- These can actually be different. (e.g.)
  - Suppose you want to fetch from an https URL
  - But you want to push changes to ssh URL

Git Protocols

- **HTTP(S)**
  - https://github.com/jquery/jquery.git
  - 80/443
  - Allows read/write or both access
  - May or may not need password for auth
  - These URLs are usually firewall friendly

- **git**
  - git://github.com/jquery/jquery.git
  - 9418
  - Read only URL
  - Allows only anonymous access
  - These URLs are not firewall friendly

Git Protocols

- **ssh**
  - git@github.com:jquery/jquery.git
  - 22
  - Allows read/write or both access
  - Uses ssh key for auth

- **File**
  - /Prabhav/code/jquery
  - No protocols
  - Allows read/write or both access

Git Config

- **Go to your Git Project directory.**

- ls –a
  - You will see .git directory.

- Inside .git directory you will be able to see config file

- cat .git/config

- Here you will be able to see a variety of configuration like URL, branch, etc.

Viewing Branches and Tags

- **See a local branch**
  - git branch

- **See remote branches**
  - Git branch –r

- **See tags**
  - Git tag

Viewing Branches and Tags

Fetching from a Remote

Now fork a repository in your GitHub Account.

Go to the folder which we used in "working locally with GIT session"

- **Add a remote destination to a local repository**
  - git remote add origin <Repository URL>
  - Add the http URL. Later we will see the ssh

- **Pull down any changes from remote repository**
  - Git fetch

- **See logs**
  - git log
  - You will see there is no changes incorporated

- **See remote logs**
  - git log origin/master
  - You will see that logs are there which are not merged with you working copy

- **Merge from origin master into my current branch**
  - Git merge origin/master

Fetching from a Remote


```
MINGW64:/e/Git                                              _ □ x

prabhav@prabhav MINGW64 /e/Git (master)
$ git remote -v

prabhav@prabhav MINGW64 /e/Git (master)
$ git remote add origin git@gitlab.pratian.com:mobility/research.git

prabhav@prabhav MINGW64 /e/Git (master)
$ git remote -v
origin   git@gitlab.pratian.com:mobility/research.git (fetch)
origin   git@gitlab.pratian.com:mobility/research.git (push)

prabhav@prabhav MINGW64 /e/Git (master)
$ git fetch
The authenticity of host 'gitlab.pratian.com (172.30.11.225)' can't be establish
ed.
ECDSA key fingerprint is SHA256:QccGryoyLl5f+pJnpdptMOgj5Gl/Dd8EzO40Jl54X4Q.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'gitlab.pratian.com,172.30.11.225' (ECDSA) to the lis
t of known hosts.
remote: Counting objects: 1837, done.
remote: Compressing objects: 100% (987/987), done.
remote: Total 1837 (delta 734), reused 1765 (delta 671)
Receiving objects: 100% (1837/1837), 35.76 MiB | 9.46 MiB/s, done.
Resolving deltas: 100% (734/734), done.
From gitlab.pratian.com:mobility/research
 * [new branch]      master      -> origin/master

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

Fetching from a Remote


```
MINGW64:/e/Git

prabhav@prabhav MINGW64 /e/Git (master)
$ git fetch origin

prabhav@prabhav MINGW64 /e/Git (master)
$ git log origin/master
commit b871f8c953604eba1de1c5ce5e8987761060661c (origin/master)
Author: arun.nv <arun.nv@pratian.com>
Date:   Mon Jul 9 18:30:21 2018 +0530

    Changes added

commit 2723ed1bf74b273301f663649805b2003b9405c
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
Merge: b99964d f78b165
Author: Anand <ra.anand@pratian.com>
Date:   Fri Oct 14 16:44:33 2016 +0530
```
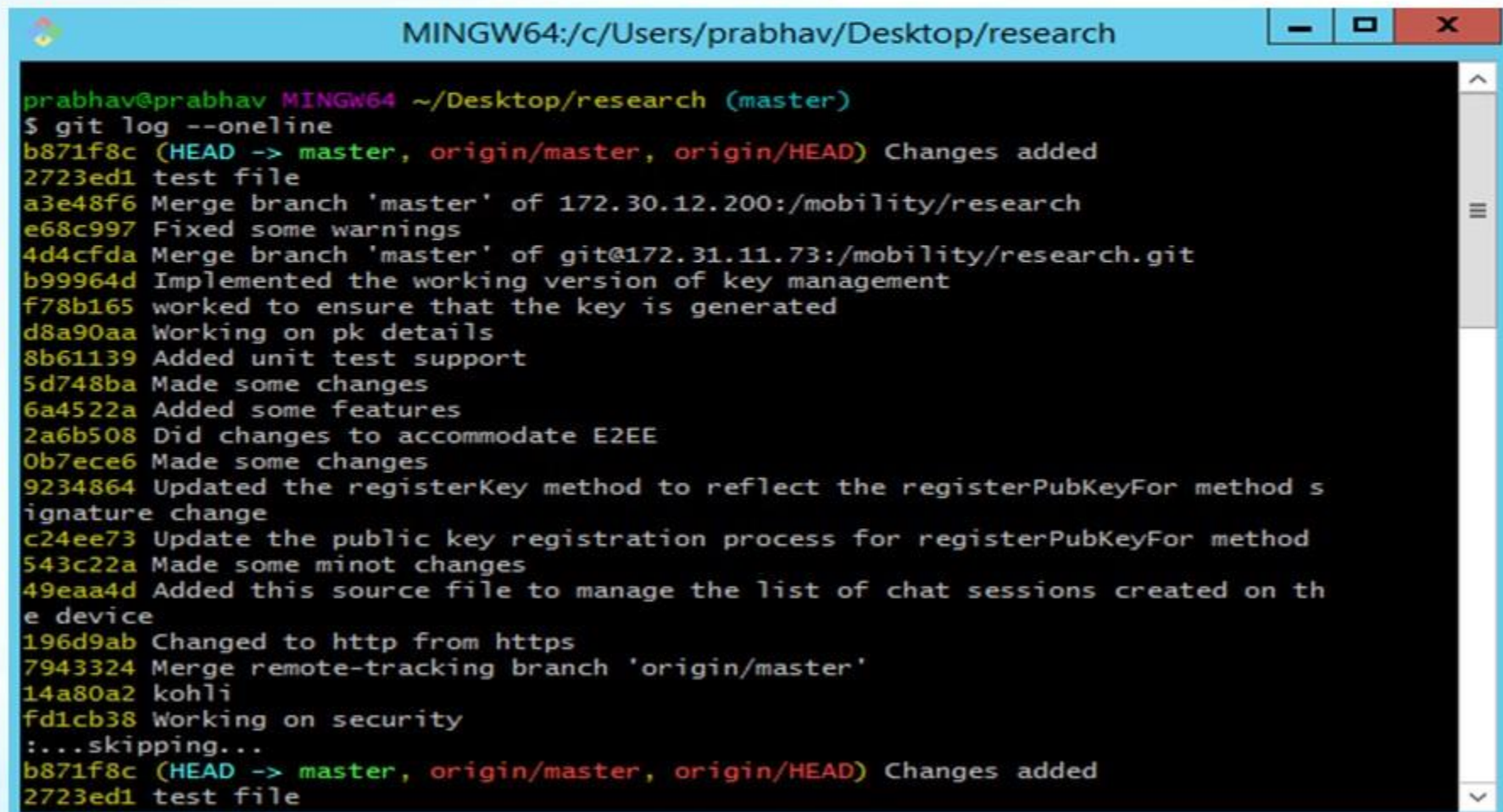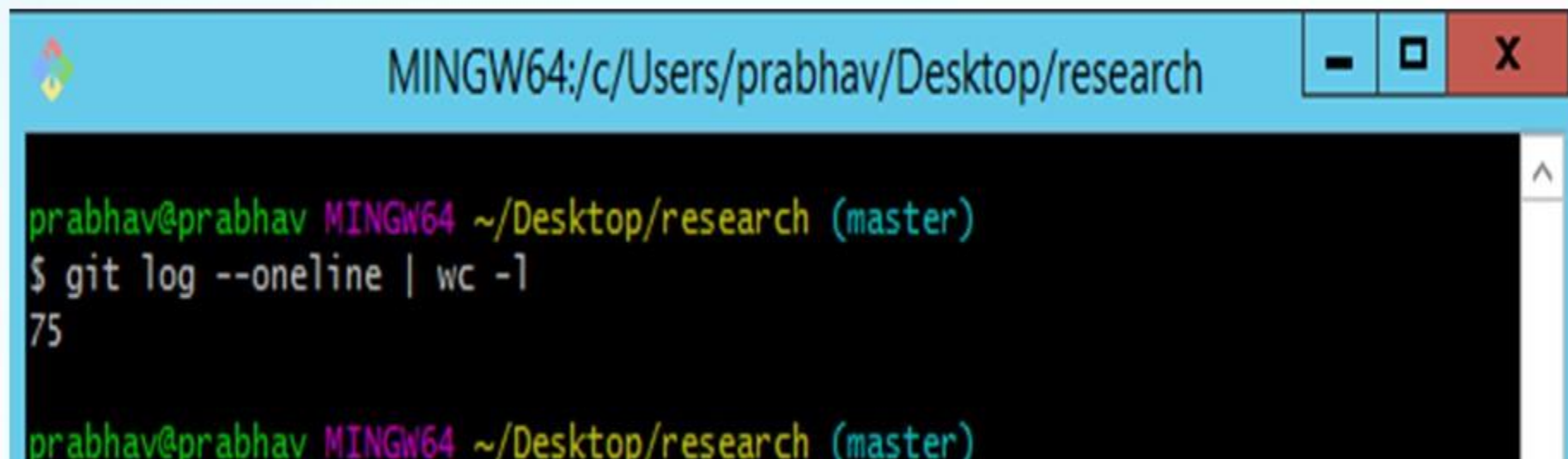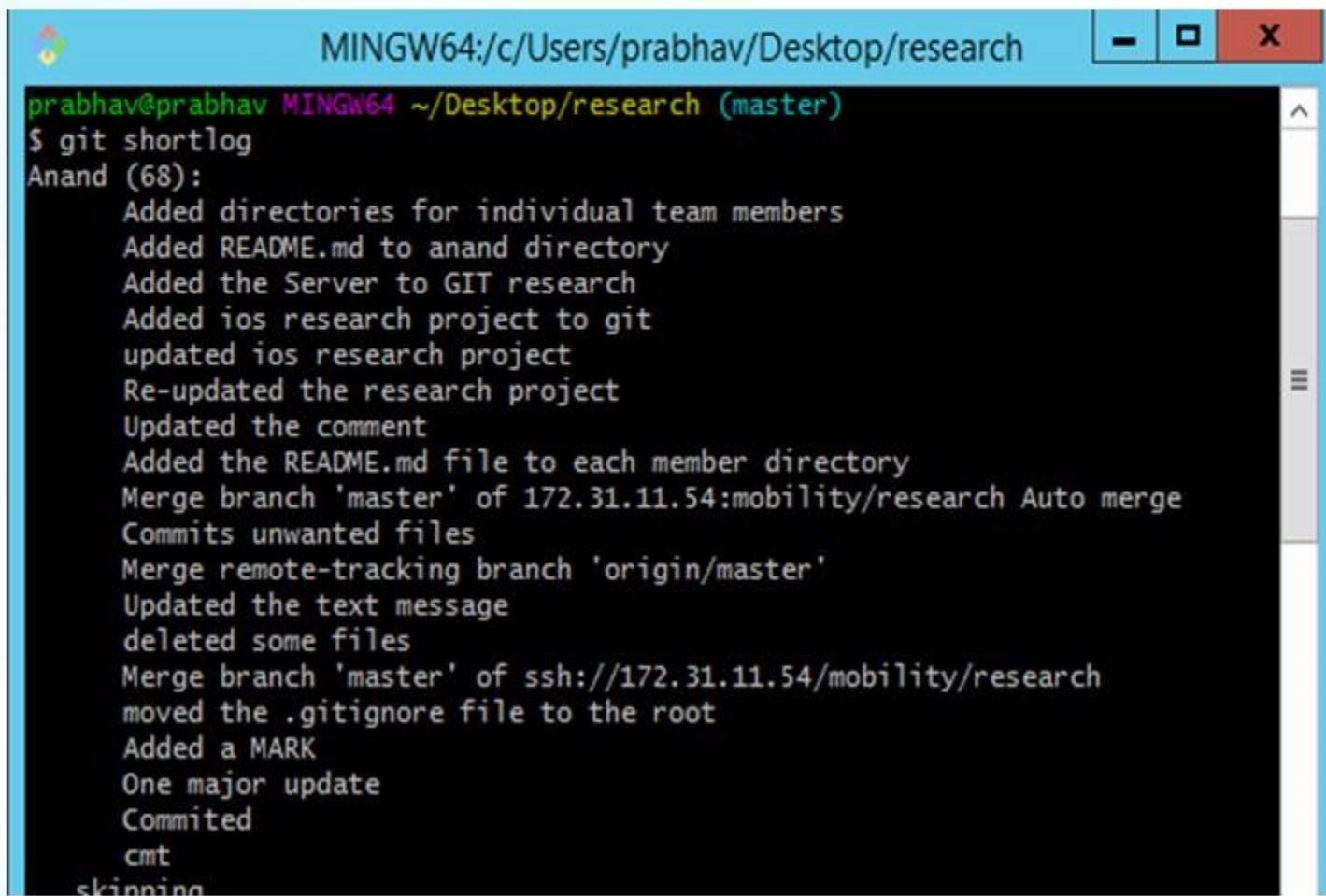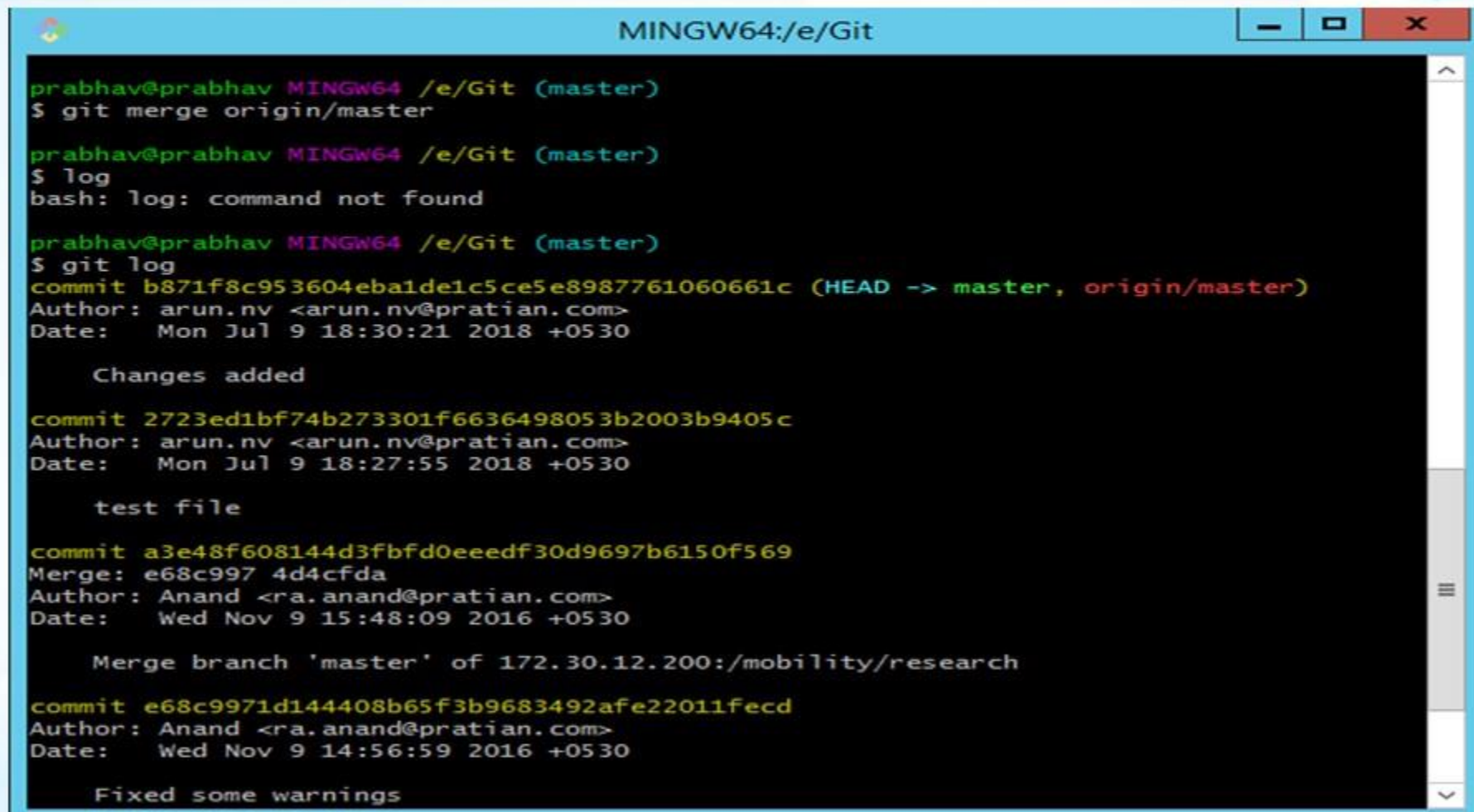
Fetching from a Remote

Pulling from a Remote
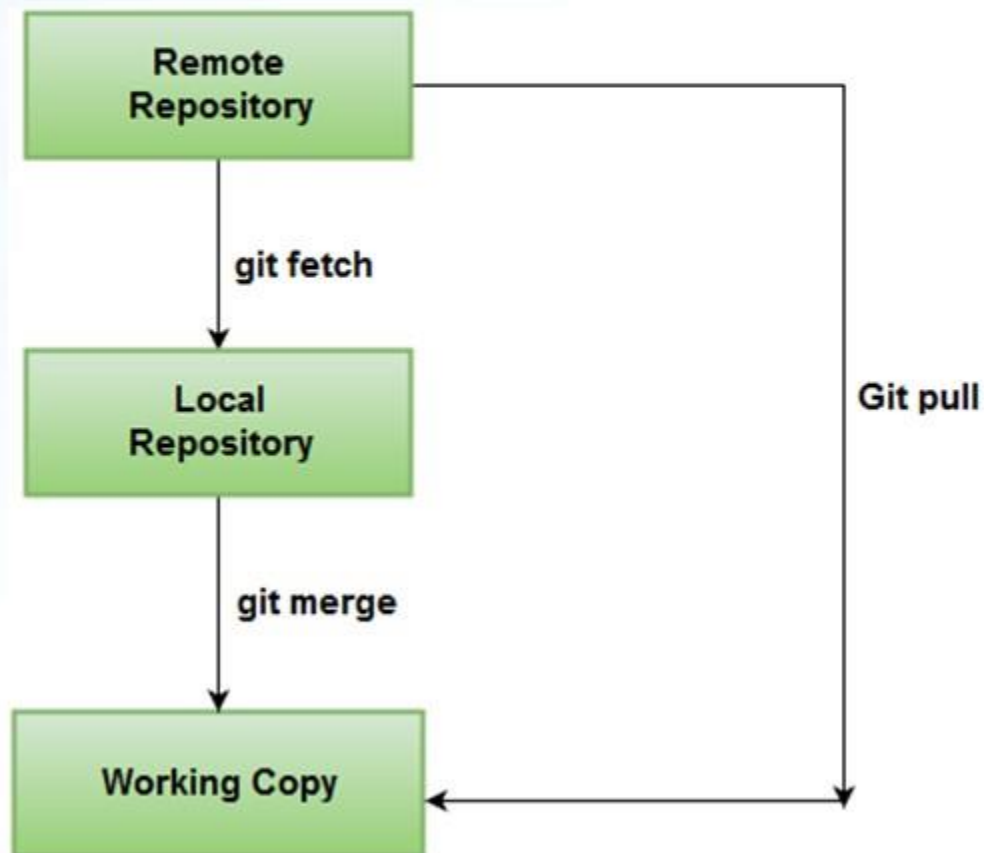
Pulling from a Remote
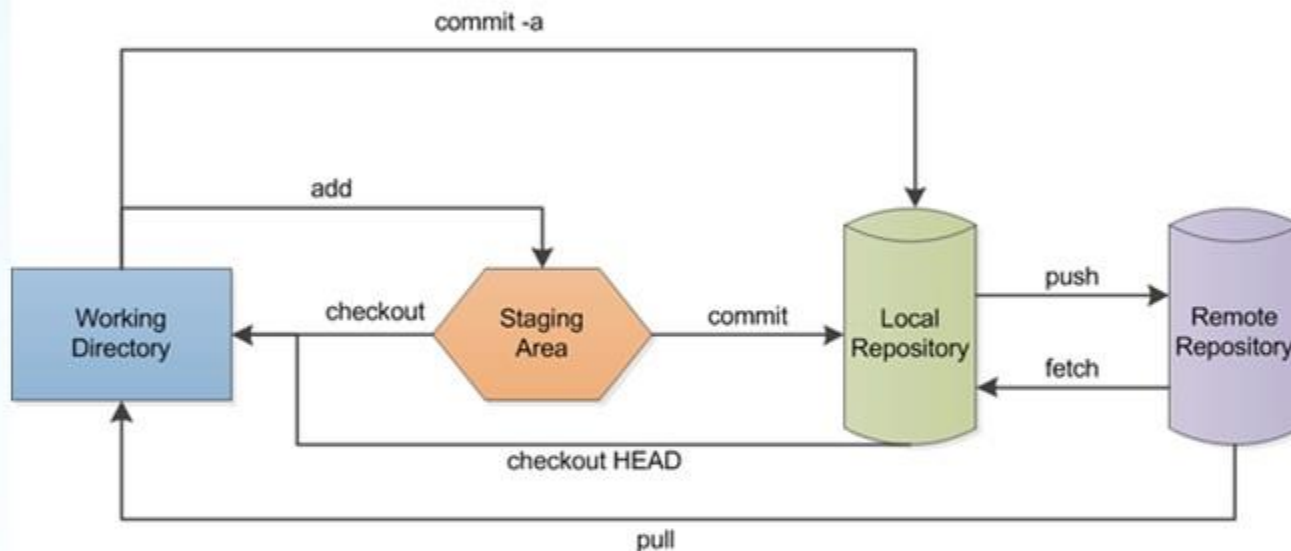
commit -a: Directly commit modified and deleted files into the local repository (*no new files!*)
add: Add a file to the staging area.
checkout: Get a file from the staging area.
checkout HEAD: Get a file from the local repository
commit: Commit files from the staging area to the local repository
push: Send files to the remote repository
fetch: Get files from the remote repository
pull: Get files from the remote repository and put a copy in the working directory
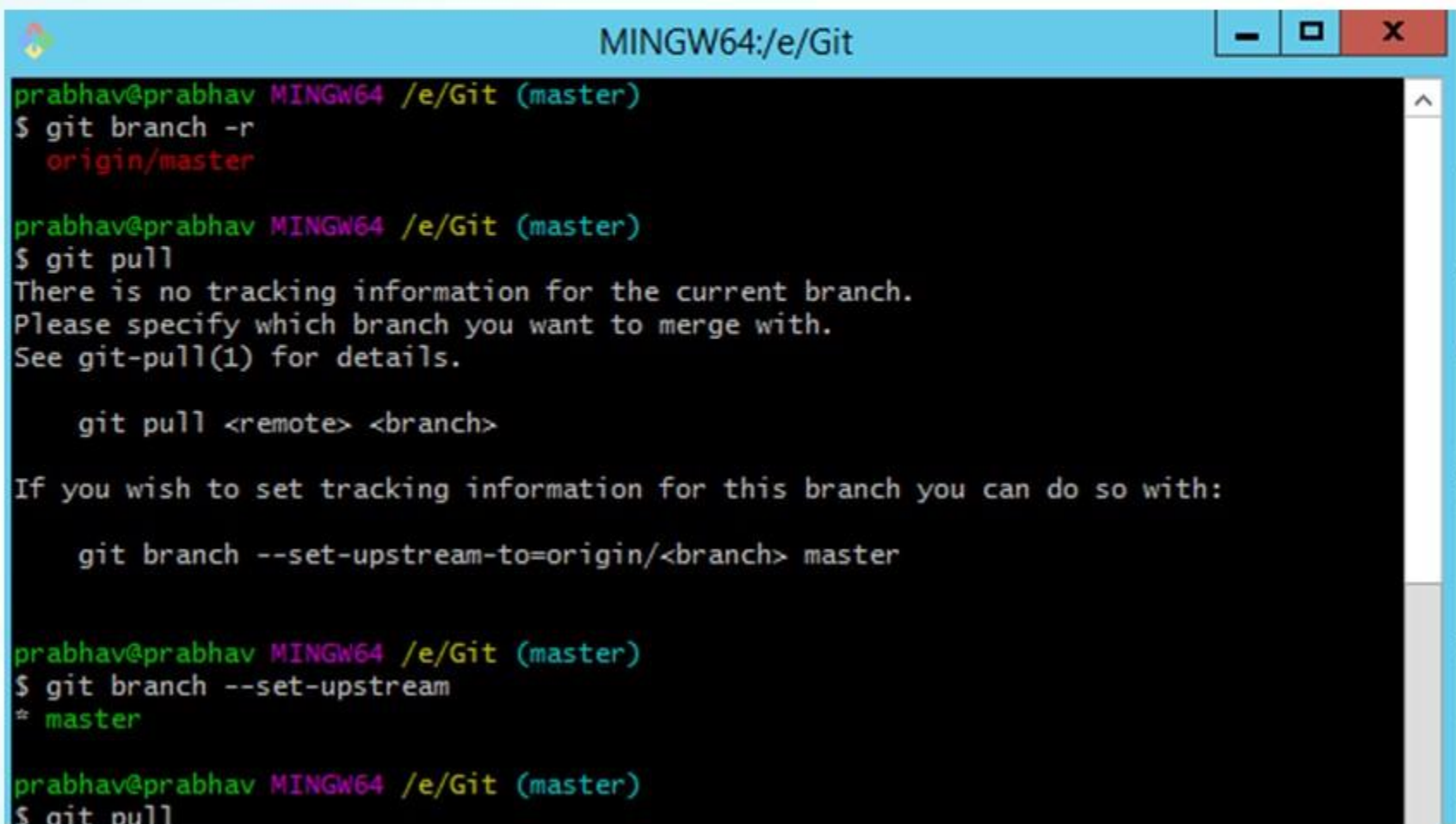
Pulling from Remote

- **See the remote branch**
  - git branch -r

- **Pull down changes from remote repository**
  - git fetch; Git merge origin/master; **or** Git pull
  - You will find that there is no correspondence setup between my master branch and the origin master(remote)

- **Set upstream**
  - git branch --set-upstream master origin/master

- **Do a Pull again**
  - git pull

Pulling from a Remote

```
MINGW64:/e/Git                                    _  □  ✕

prabhav@prabhav MINGW64 /e/Git (master)
$ git branch -r
  origin/master

prabhav@prabhav MINGW64 /e/Git (master)
$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master


prabhav@prabhav MINGW64 /e/Git (master)
$ git branch --set-upstream
* master

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

Pushing to a Remote

- **Edit your README.txt file**
  - Vim README.txt

- **Check Git status**
  - Git status

- **Add and commit the change**
  - Git –am "Adding"

- **Check Git status**
  - Git status
  - Git will say a message that your branch is ahead of "origin/master" by 1 commit.
  - That means there are pending changes to be pushed to remote.

- **Push changes to remote repository**
  - Git push
  - It will prompt for username and password

Pushing to a Remote

- **Push changes to remote repository. Without Username and Password**

- **Remove the remote**
  - Git remote rm origin

- **Check if you have remote repository**
  - Git remote -v

- **Now add the repository and add the ssh version**
  - Git remote add origin <ssh GIT URL>

- **Make some changes**

- **Push changes to remote repository**
  - Git push
  - It will not prompt for username and password because its using the ssh

Pushing to a Remote


```
MINGW64:/e/Git

prabhav@prabhav MINGW64 /e/Git (master)
$ ls
anand/  arun/  preeti/  README.md  REAMDE.txt  ritesh/  saikat/  sanjay/

prabhav@prabhav MINGW64 /e/Git (master)
$ vi README.md

prabhav@prabhav MINGW64 /e/Git (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master


prabhav@prabhav MINGW64 /e/Git (master)
$ git push --set-upstream origin master
Everything up-to-date
Branch 'master' set up to track remote branch 'master' from 'origin'.

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

Creating and Verifying Tags

- **Tag your repository. Tag the current master branch as v1.0**
  - git tag v1.0

- **See your tag**
  - Git tag

- **Add annotations with your tag**
  - git tag –a v1.0_withMessage
  - You will need to put the message.
  - You can also provide a –m to provide message directly

- **See your tag**
  - Git tag

Creating and Verifying Tags

- **Provide a signed tag**
  - git tag –s v1.0_signed
  - You will need to put the message.
  - You will need to provide passphrase
  - If signing failed
    - gpg --list-key
    - gpg --gen-key
    - git config --global user.signingkey 86F87C475A6E68D050E3C772F3EBAF4A79F6A0BB

- **See your tag**
  - Git tag

- **Verify a tag**
  - git tag –v v1.0
  - git tag –v v1.0_withMessage
  - git tag –v v1.0_signed

Pushing Tags to remote

- **Go to GitHub and see your created Tags**

- **Do a git push**
  - git push
  - You will notice that there are no tags when you browse through the UI

- **You will again notice that there are no tags**
  - By default git will not push tags.

- **Push git tags**
  - Git push –tags
  - It will push the tags to remote repository.
  - You will notice now tags are available when you browse through the UI

- **Tagging gives you stable points in your source code.**
  - E.g. You may want to tag individual release which was successful.

Thank You