

Splitting a Repository While Preserving History

Keep in mind that once something is in git is in there forever taking up space and logs even if you remove it, put it on a branch, or whatever.

There is no option to just do reorganization of the existing repo and renaming - it will keep all the old references and versions of junk and keep adding to the size, which is uncouth.

Options

To split there are 2 choices, each *should* keep history properly if done right, but it should be easy to see why #1 is far easier given the piles of stuff in nds-labs, which is 50MB - quite large given what's in there - some due to mistakes and some due to just piles of old junk. By the way apicli is over 2/3 of the entire size, looks like the binary is the culprit.

1. Include what you want by setting up everything under one subdirectory with ONLY that which you wish to keep in master.
 - a. `git filter-branch --prune-empty --subdirectory-filter <path to what you keep>`
2. Exclude everything you don't want individually by running rewrites for every directory you don't want. So for each delete:
 - a. `git filter-branch --force --index-filter 'git rm -r --cached --ignore-unmatch <delete-path>' --prune-empty --tag-name-filter cat -- --all`

Here are the processes to split, I'll give an example after the explanation:

1. Inclusive style - in a clean directory:
 - a. Clone nds-labs to labs.1
 - b. remove the origin from labs.1
 - c. clone labs.1 to labs.2 and remove origin from labs.2
 - d. In labs.2, merge v2
 - e. clone labs.2 to labs.3 and remove its origin
 - f. Arrange the subdirectory that you want to keep - ignore everything outside of the directory - it doesn't matter.
 - i. Use `git mv` to move everything under the path you wish to keep
 - ii. `git rm -f` inside the new path - everything that you don't want but ignore anything not under the path - just make that folder nice.
 - iii. `git commit`
 - iv. Every time you make a major change you can clone again, in case you discover something down the line you can `rm -r` the broken clones and redo
 1. clone the next step: labs3 to labs4 and remove the origin from the new clone
 2. Keep arranging and commit
 - g. When you're done arranging: You should have some clone of clone of .clone of where you are on master and you have one subdirectory where you have the good stuff and no junk.....
 - h. Make one more step-clone
 - i. `git filter-branch --prune-empty --subdirectory-filter <path to what you keep>`
 - j. Make the repo in git, bare (no readme or anything)
 - k. `git remote add origin <github url>`
 - l. `git push origin master`
2. Exclusive style:
 - a. Don't try it, it's rough and error prone
 - b. But if you wish, it's a similar clone-change-commit/clone-change-commit/.... process as in inclusive
 - i. But you need to redact every file and directory that you wish to omit, and it's much easier to setup what you want and split - see Inclusive above.

Moderately complicated example:

The goal - cut the developer-shell out of v2:devtools/Makefiles and v2:devtools/ndsdev into ndslabs-developer-shell

This is slightly complicated by the fact that we want the developer-shell to include the Makefile's under /usr/local/include so that Makefiles can:

```
include Makefile.nds
```

The strategy: The target is devtools/ndsdev, to which the Makefiles in devtools/Makefiles will be added to /usr/local/include and devtools/ndsdev

will be the new root of the new repository github.com/nds-org/ndslabs-developer-shell (this example is hosted in ndslabs-split-example, as ndslabs-developer-shell

is in use.

Here is the log:

```
[raila@raila]$ mkdir split
[raila@raila]$ cd split
[raila@raila split]$ git clone https://github.com/nds-org/nds-labs.git nds.1
```

```

Cloning into 'nds.1'...
remote: Counting objects: 16373, done.
remote: Compressing objects: 100% (101/101), done.
remote: Total 16373 (delta 53), reused 0 (delta 0), pack-reused 16269
Receiving objects: 100% (16373/16373), 45.46 MiB | 3.52 MiB/s, done.
Resolving deltas: 100% (5741/5741), done.
Checking connectivity... done.
[raila@raila split]$ cd nds.1

[raila@raila nds.1]$ git remote rm origin

[raila@raila nds.1]$ git merge -q origin/v2

[raila@raila split]$ git clone nds.1 nds.2
Cloning into 'nds.2'...
done.

[raila@raila split]$ cd nds.2
[raila@raila nds.2]$ ls devtools/

DEVELOPER-NOTES  girder/  go/  keyrings/  Makefiles/  ndsdev/  rest/

[raila@raila nds.2]$ git remote rm origin

##Move all the goodies under devtools/ndsdev

[raila@raila nds.2]$ git mv devtools/Makefiles/{gmsl,__gmsl,,Makefile.nds} devtools/ndsdev/FILES.ndsdev/usr/local/include

[raila@raila nds.2]$ git mv devtools/Makefiles/{gmsl,__gmsl,,Makefile.nds} devtools/ndsdev/FILES.ndsdev/usr/local/include

## Remove anything we don't want

[raila@raila nds.2]$ git rm devtools/ndsdev/ndsdevctl
rm 'devtools/ndsdev/ndsdevctl'

## Checkpoint:

[raila@raila nds.2]$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    deleted:    devtools/Makefiles/Makefile.nds
    deleted:    devtools/Makefiles/README.md
    deleted:    devtools/Makefiles/TODO
    deleted:    devtools/Makefiles/_gmsl
    deleted:    devtools/Makefiles/gmsl

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

    devtools/ndsdev/FILES.ndsdev/usr/local/include/

no changes added to commit (use "git add" and/or "git commit -a")
[raila@raila nds.2]$ git commit -m 'moving files in place for reorg'
On branch master
Changes not staged for commit:
  deleted:    devtools/Makefiles/Makefile.nds
  deleted:    devtools/Makefiles/README.md
  deleted:    devtools/Makefiles/TODO
  deleted:    devtools/Makefiles/_gmsl
  deleted:    devtools/Makefiles/gmsl

Untracked files:
  devtools/ndsdev/FILES.ndsdev/usr/local/include/

no changes added to commit

[raila@raila nds.2]$ git add .
[raila@raila nds.2]$ git commit -m 'moving files in place for reorg'
[master 799343c] moving files in place for reorg
5 files changed, 0 insertions(+), 0 deletions(-)
rename devtools/{ => ndsdev/FILES.ndsdev/usr/local/include}/Makefiles/Makefile.nds (100%)
rename devtools/{ => ndsdev/FILES.ndsdev/usr/local/include}/Makefiles/README.md (100%)
rename devtools/{ => ndsdev/FILES.ndsdev/usr/local/include}/Makefiles/TODO (100%)
rename devtools/{Makefiles => ndsdev/FILES.ndsdev/usr/local/include}/_gmsl (100%)
rename devtools/{Makefiles => ndsdev/FILES.ndsdev/usr/local/include}/gmsl (100%)

```

```
[raila@raila nds.2]$ git commit -m 'moves to ndsdev'
[master 414f0c6] moves to ndsdev
8 files changed, 43 deletions(-)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/kube-up.sh (100%)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/manifests/etcd.json (100%)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/ndslabs-down.sh (100%)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/ndslabs-up.sh (100%)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/ndslabs/apiserver.yaml (100%)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/ndslabs/gui.yaml (100%)
rename devtools/ndsdev/{startup => FILES.ndsdev/usr/local/bin}/toolsrv.sh (100%)
delete mode 100755 devtools/ndsdev/ndsdevctl
```

```
[raila@raila nds.2]$ cd ..
[raila@raila split]$ git clone nds.2 nds.3
Cloning into 'nds.3'...
done.
[raila@raila split]$ cd nds.3
```

```
[raila@raila nds.3]$ git remote rm origin
```

Things in place now.... look over with a fine-tooth comb, make sure all inclusions are included and all exclusions are excluded:

```
[raila@raila nds.3]$ find devtools/ndsdev/
devtools/ndsdev/
devtools/ndsdev/Dockerfile.ndsdev
devtools/ndsdev/FILES.ndsdev
devtools/ndsdev/FILES.ndsdev/usr
devtools/ndsdev/FILES.ndsdev/usr/local
devtools/ndsdev/FILES.ndsdev/usr/local/bin
devtools/ndsdev/FILES.ndsdev/usr/local/bin/kube-up.sh
devtools/ndsdev/FILES.ndsdev/usr/local/bin/manifests
devtools/ndsdev/FILES.ndsdev/usr/local/bin/manifests/etcd.json
devtools/ndsdev/FILES.ndsdev/usr/local/bin/ndslabs-down.sh
devtools/ndsdev/FILES.ndsdev/usr/local/bin/ndslabs-up.sh
devtools/ndsdev/FILES.ndsdev/usr/local/bin/ndslabs
devtools/ndsdev/FILES.ndsdev/usr/local/bin/ndslabs/apiserver.yaml
devtools/ndsdev/FILES.ndsdev/usr/local/bin/ndslabs/gui.yaml
devtools/ndsdev/FILES.ndsdev/usr/local/bin/toolsrv.sh
devtools/ndsdev/FILES.ndsdev/usr/local/bin/usage
devtools/ndsdev/Makefile
devtools/ndsdev/README.md
## looks good, now the rewrite:
```

```
[raila@raila nds.3]$ git status
On branch master
nothing to commit, working directory clean
```

```
[raila@raila nds.3]$ git filter-branch --prune-empty --subdirectory-filter devtools/ndsdev master
Rewrite 414f0c618f01b15414b8ccec3627ac6a9509a5e5 (30/30)
Ref 'refs/heads/master' was rewritten
```

```
[raila@raila nds.3]$ git status
On branch master
nothing to commit, working directory clean
[raila@raila nds.3]$ ls
Dockerfile.ndsdev  FILES.ndsdev/  Makefile  README.md
```

OK, now push to the new repos after creating on github:

```
[raila@raila nds.3]$ git remote add origin https://github.com/nds-org/split-example.git
[raila@raila nds.3]$ git push origin master
Counting objects: 124, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (66/66), done.
Writing objects: 100% (124/124), 17.38 KiB | 0 bytes/s, done.
Total 124 (delta 44), reused 89 (delta 42)
To https://github.com/nds-org/split-example.git
* [new branch] master -> master
```

Check size:

```
[raila@raila nds.3]$ cd ..  
[raila@raila split]$ du -skhc *  
101M   nds.1  
55M    nds.2  
276K   nds.3  
156M   total
```

eliminated 100+M of clone, yay

now clone the repo and take a look:

```
[raila@raila split]$ git clone https://github.com/nds-org/split-example.git  
Cloning into 'split-example'...  
remote: Counting objects: 124, done.  
remote: Compressing objects: 100% (64/64), done.  
remote: Total 124 (delta 44), reused 124 (delta 44), pack-reused 0  
Receiving objects: 100% (124/124), 17.38 KiB | 0 bytes/s, done.  
Resolving deltas: 100% (44/44), done.  
Checking connectivity... done.
```

[raila@