SVN To Git Migration

Portfolio & DevOps Toolchain - Internal

Exported on 12/21/2022

Table of Contents

Before we start the Git repository migration, make sure the following pre-requisite is in place.

- The powerful system, If you are dealing with a number of SVN repositories at the same time.
- Good and continue network bandwidth.
- Git and TortoiseSVN client installs on your system.

Below is the very high-level step for SVN to Git repository migration.

1. Get the SVN repository users and map them with Git repository users
2. Clone SVN repository on the local system and convert to git repository
3. Convert SVN tag to Git tag
4. Convert SVN branch to Git branch
5. Create an empty git repository, this will be the target repository.
6. Push local converted repository to target empty repository.

Now let's start the repository preparation.

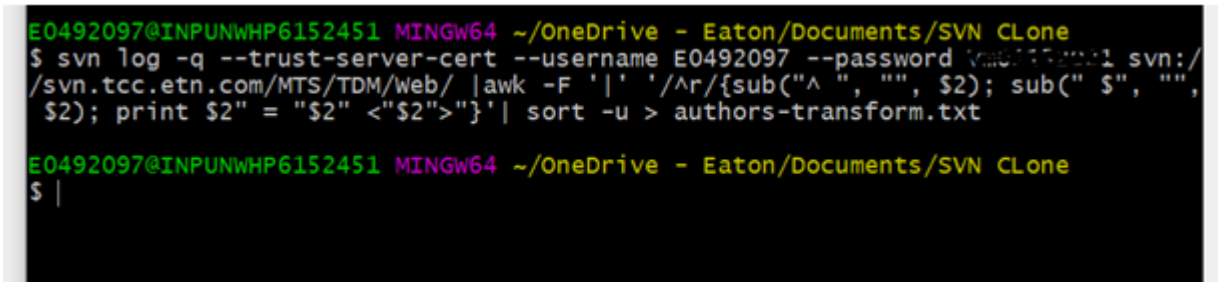
> **Note**
>
> In this example, we have used the SVN "web" repository and Vishalmande, HitendraChauhan as mapped git users.

1 Step 1: Get the SVN repository users and map them with Git repository users

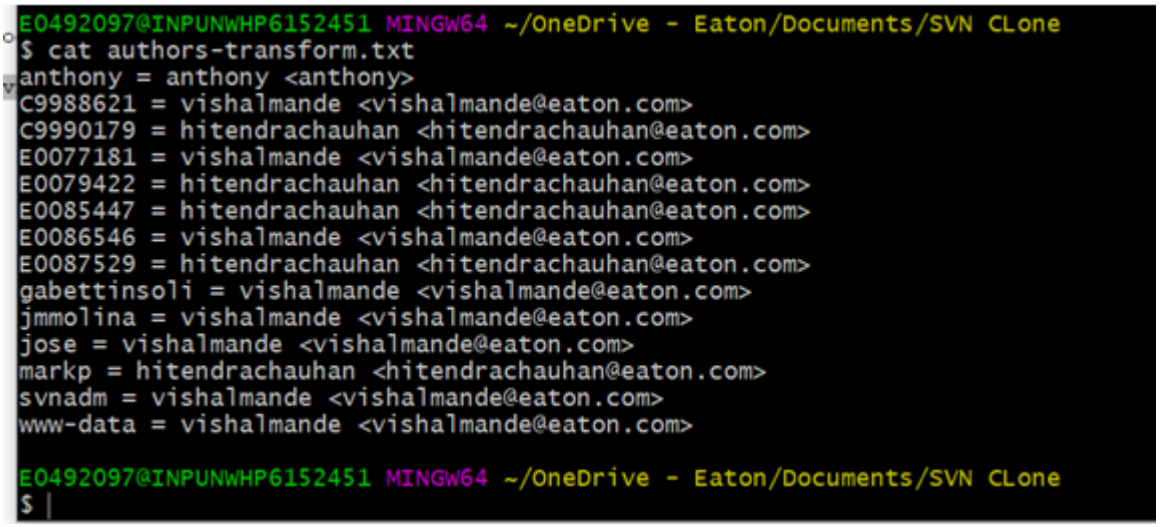
The AUthors/Users from the SVN repository along with historical commits can be extracted using the below command.

svn log -q --trust-server-cert --username E0492097 --password <passwords >svn://svn.tcc.etn.com/MTS/TDM/Web/ ***|awk -F '|' '/^r/{sub("^ ", "", $2); sub(" $", "", $2); print $2" = "$2" <"$2">"}'| sort -u > authors-transform.txt***

Here user name and password will be your SVN repository credentials.



Mapp SVN users to Git users, seem below screenshot.



2 Step 2: Clone the SVN repository on the local system and convert to the git repository

Clone SVN repository by using following command

git svn clone svn://svn.tcc.etn.com/MTS/TDM/Web/ ***--authors-file=./authors-transform.txt -T trunk -b branches -t tag***

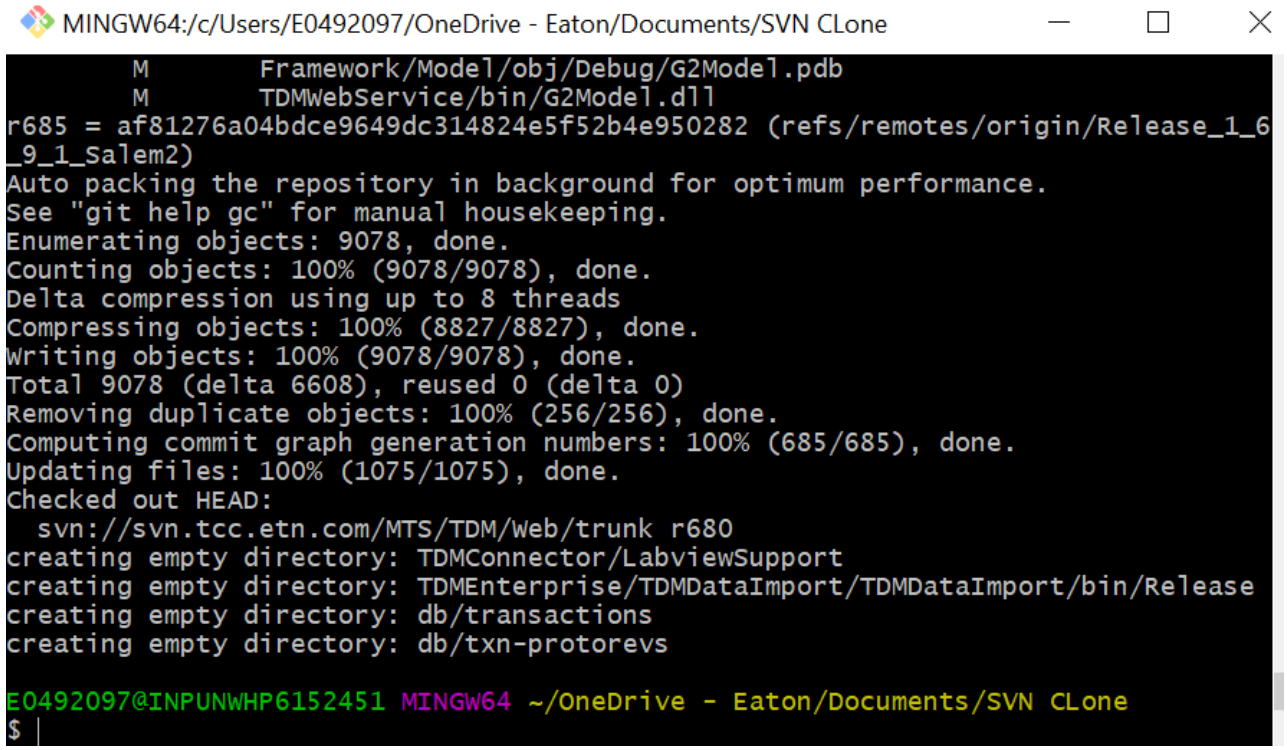
Here, -T Trank à Trunk Directory.

-b branches à branches directory.

-t tag à tags directory.

This will trace your first SVN revision and clone all the commits starting from the first commit to the latest one. The ***git svn clone*** will drill down all the files from the svn repository. This clone will take some time based on your repository size, the number of revisions in your SVN repository and network bandwidth. So go ahead and take your coffee 🙂

Once you are done with Clone, the result will look like the one below. Also, you will find the SVN repository folder along with .git metadata.



3 Step 3: Convert SVN tag to Git tag

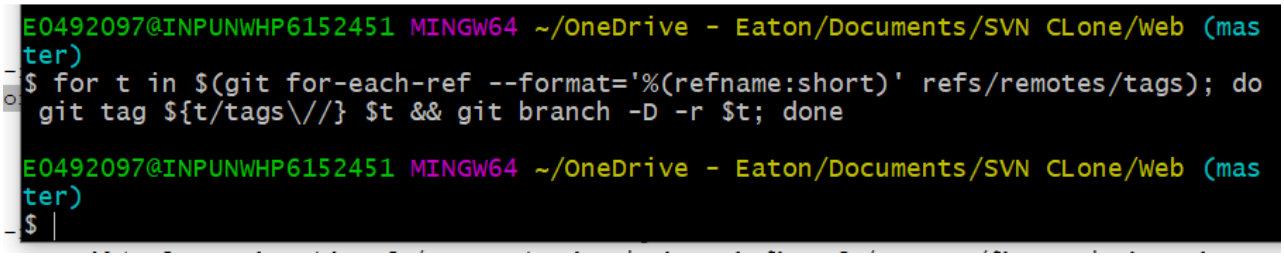
Convert the SVN tag into Git tag by using the below command.

If you have multiple tags then:

for t in $(git for-each-ref --format='%(refname:short)' refs/remotes/tags); do git tag ${t/tags\//} $t && git branch -D -r $t; done

If you have few tags then:

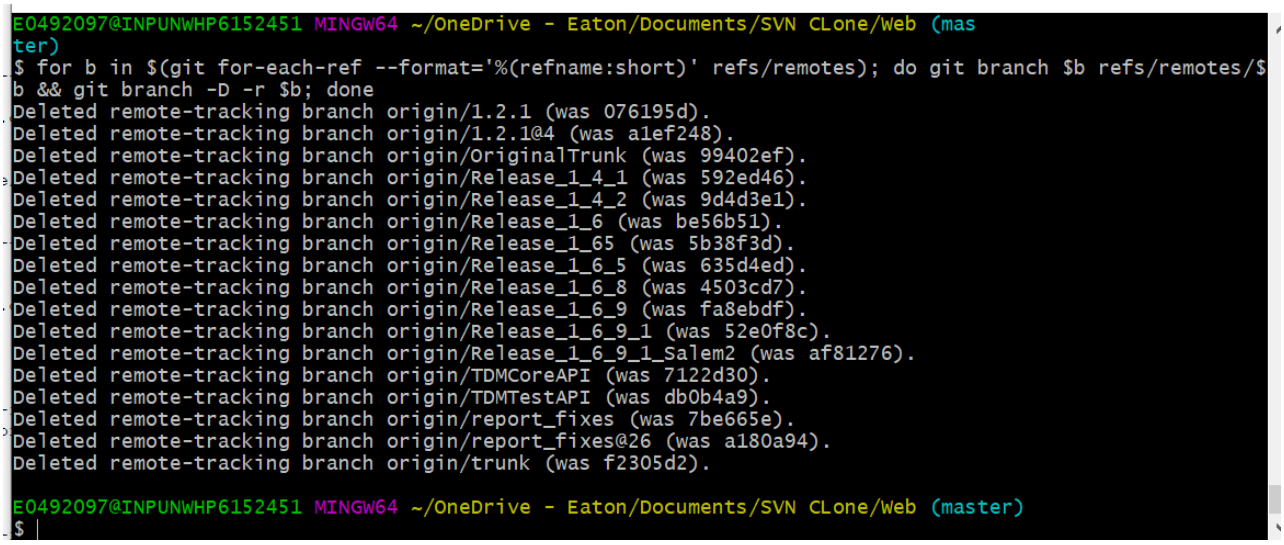
for t in $(git for-each-ref --format='%(refname:short)' refs/remotes/tags); do git tag -a "$t" -m"say farewell to SVN" "/tags/$t"; done

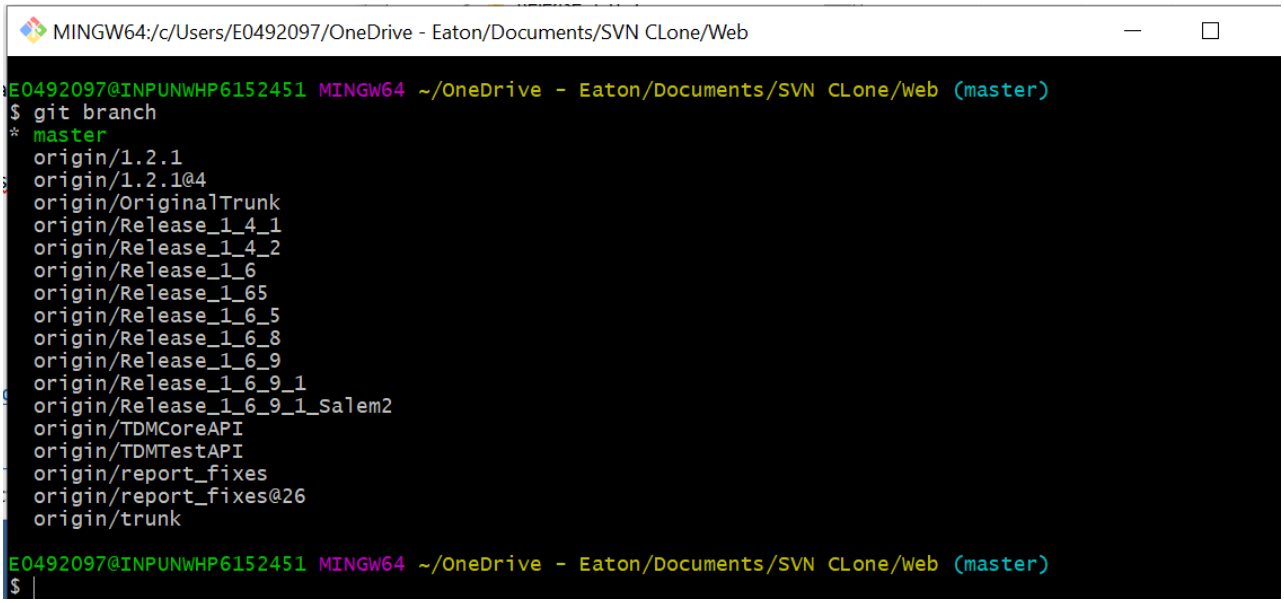


4 Step 4: Convert the SVN branch to the Git branch

Convert the SVN branches to Git branches by using the following command.

for b in $(git for-each-ref --format='%(refname:short)' refs/remotes); do git branch $b refs/remotes/$b && git branch -D -r $b; done



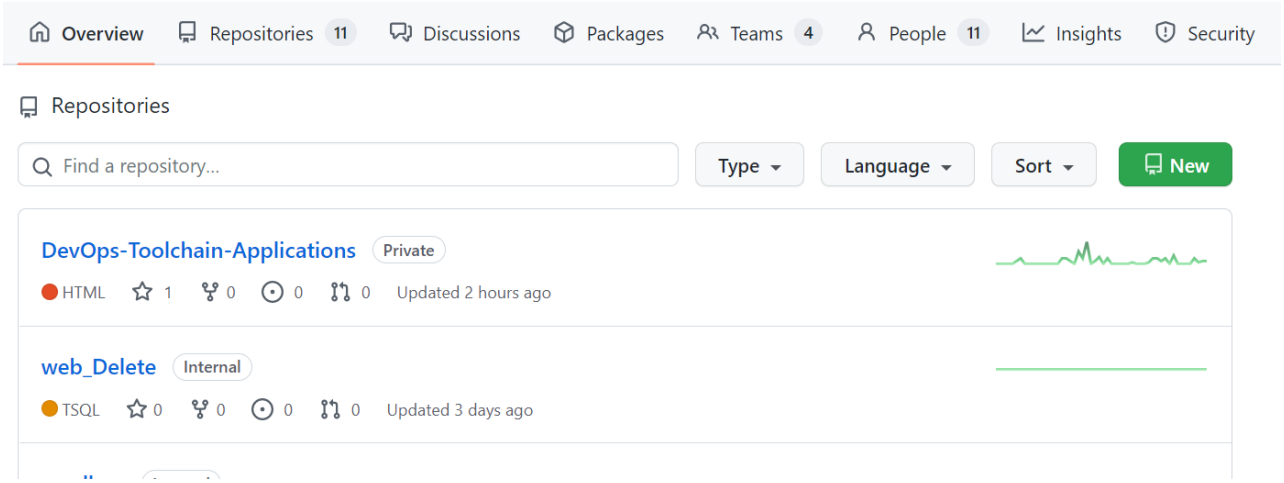


Now at this point, your SVN repository is converted to the git repository and ready to push to the remote repo.

5 Step 5: Create an empty git repository, this will be the target repository.

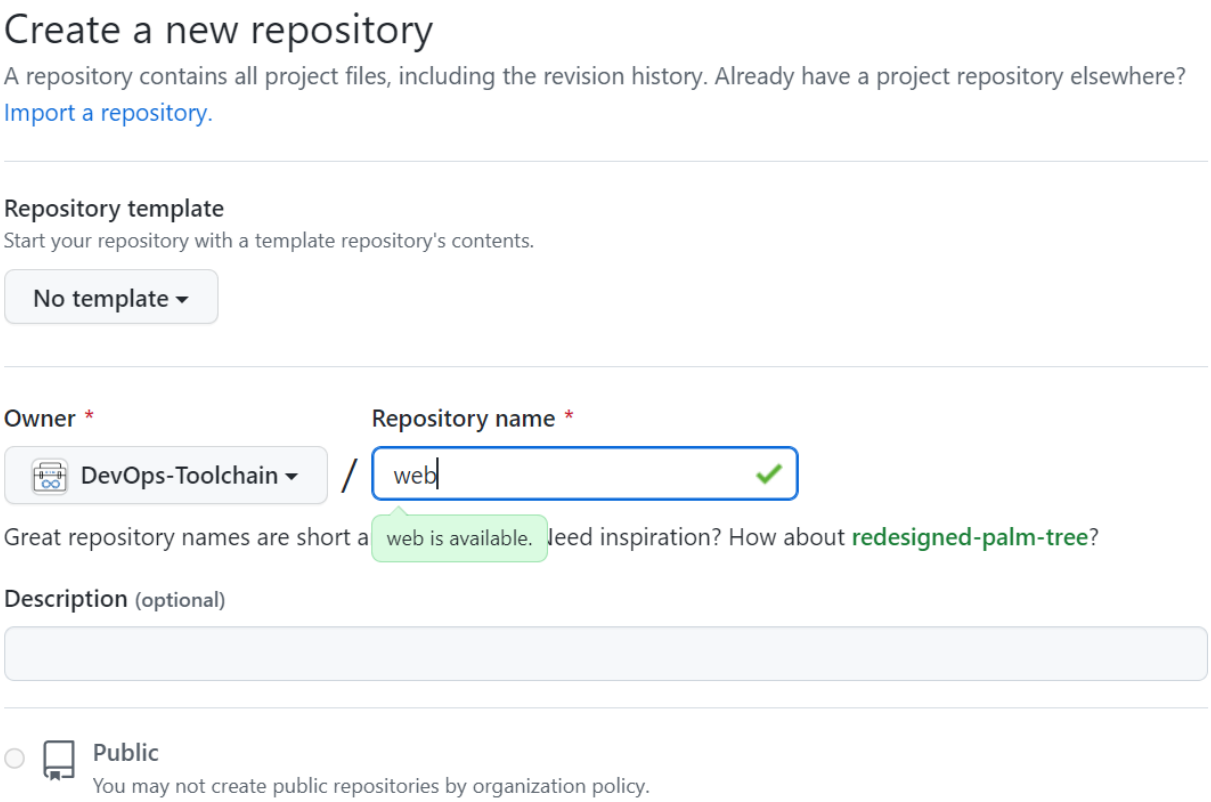
> **Note**
>
> Here we are using GitHub as a git repository, you might have other git-based apps like Bitbucket.

Log in to your GitHub[1] account and create a new empty repository by clicking on the "New" button.

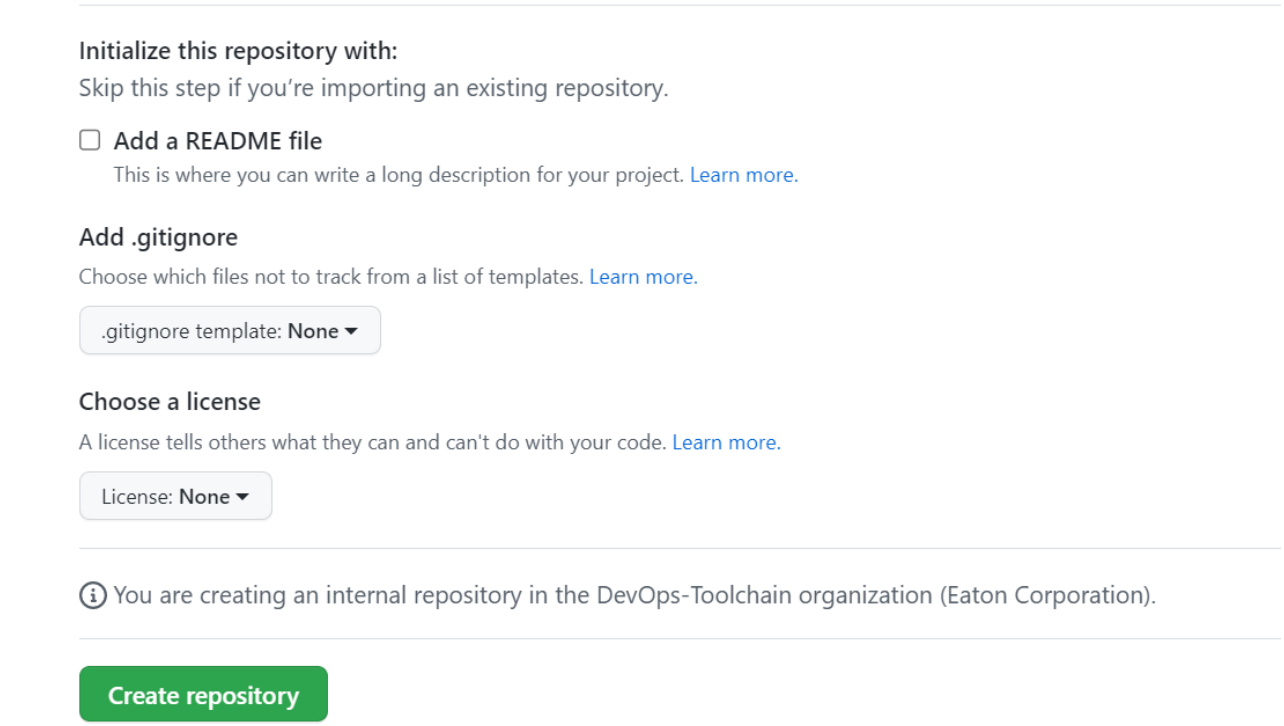


Fill in the repository details, the same repo name which is cloned on the local system (recommended)

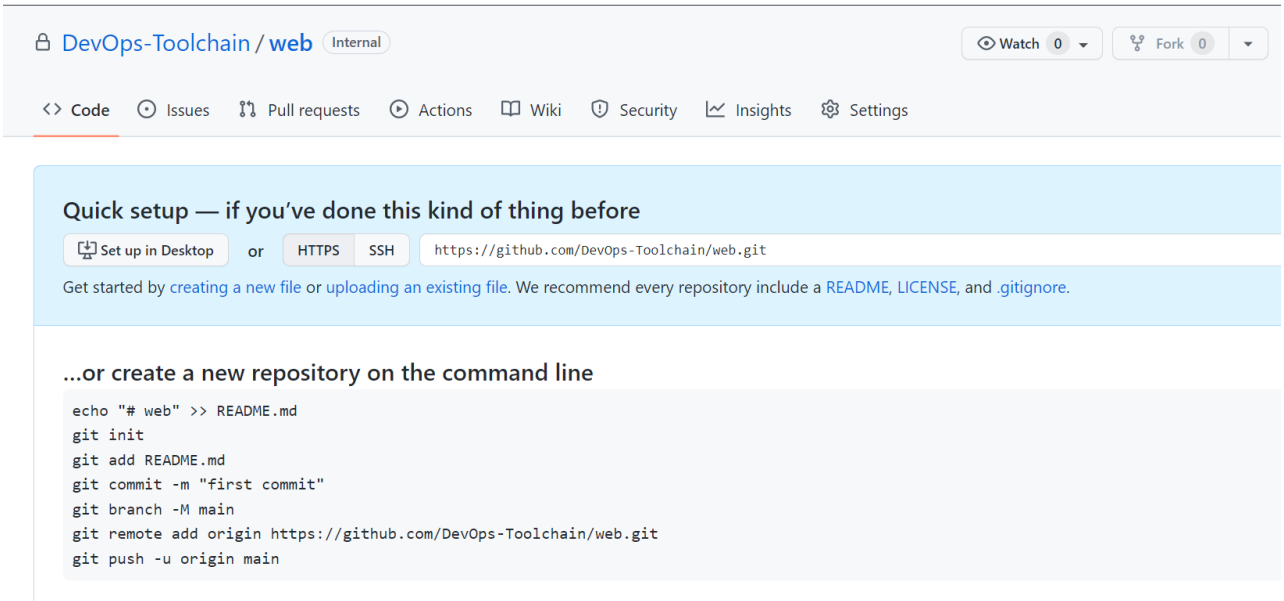
1 https://github.com/



Click on the "Create" button.



The empty git repository will look like the below:



6 Step 6: Push the local converted repository to target the empty repository:

Now commit the git repository, and add all commits in the stage.

```
E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Web (mas
$ git commit -m "svn to git"
On branch master
nothing to commit, working tree clean

E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Web (mas
ter)
$ git add --all

E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Web (mas
```

Update the origin in the local git repository to the remote empty repo.

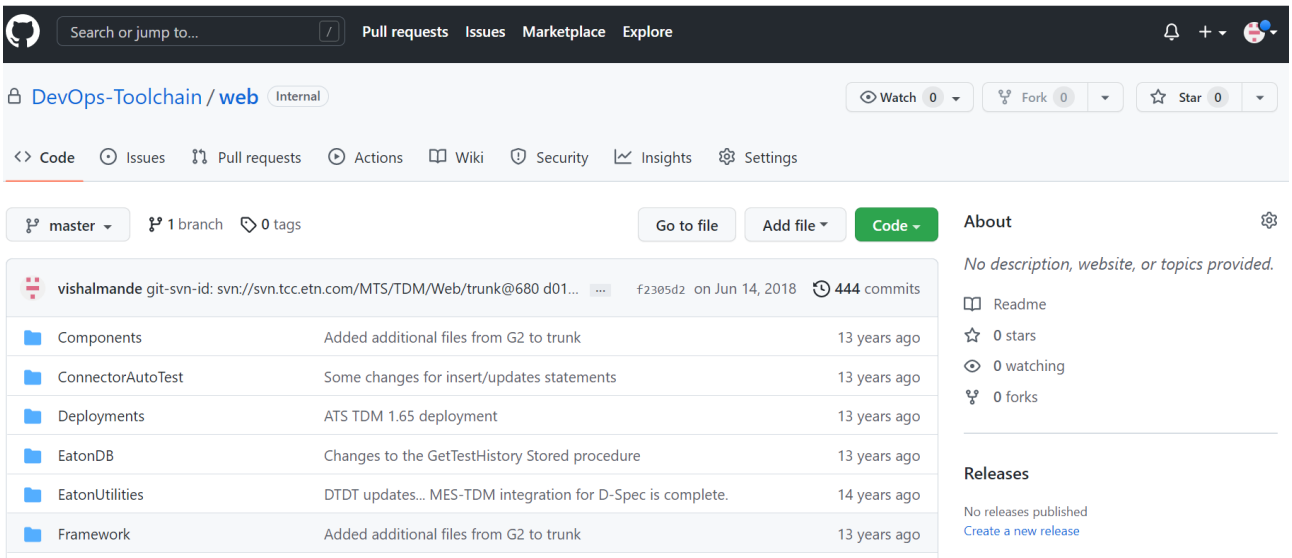
```
E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Web (mas
ter)
$ git remote add origin https://vishalmande:ghp_t2hOSnNX5lGto36POGPqSBgRY7jEqq4U
HlWj@github.com/DevOps-Toolchain/web.git

E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Web (mas
```

Push the master branch to the remote repository.

```
E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Web (mas
ter)
$ git push -u origin master
Enumerating objects: 5815, done.
Counting objects: 100% (5815/5815), done.
Delta compression using up to 8 threads
Compressing objects: 100% (1654/1654), done.
Writing objects: 100% (5815/5815), 89.55 MiB | 1.24 MiB/s, done.
Total 5815 (delta 4479), reused 5258 (delta 4059)
remote: Resolving deltas: 100% (4479/4479), done.
To https://github.com/DevOps-Toolchain/web.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

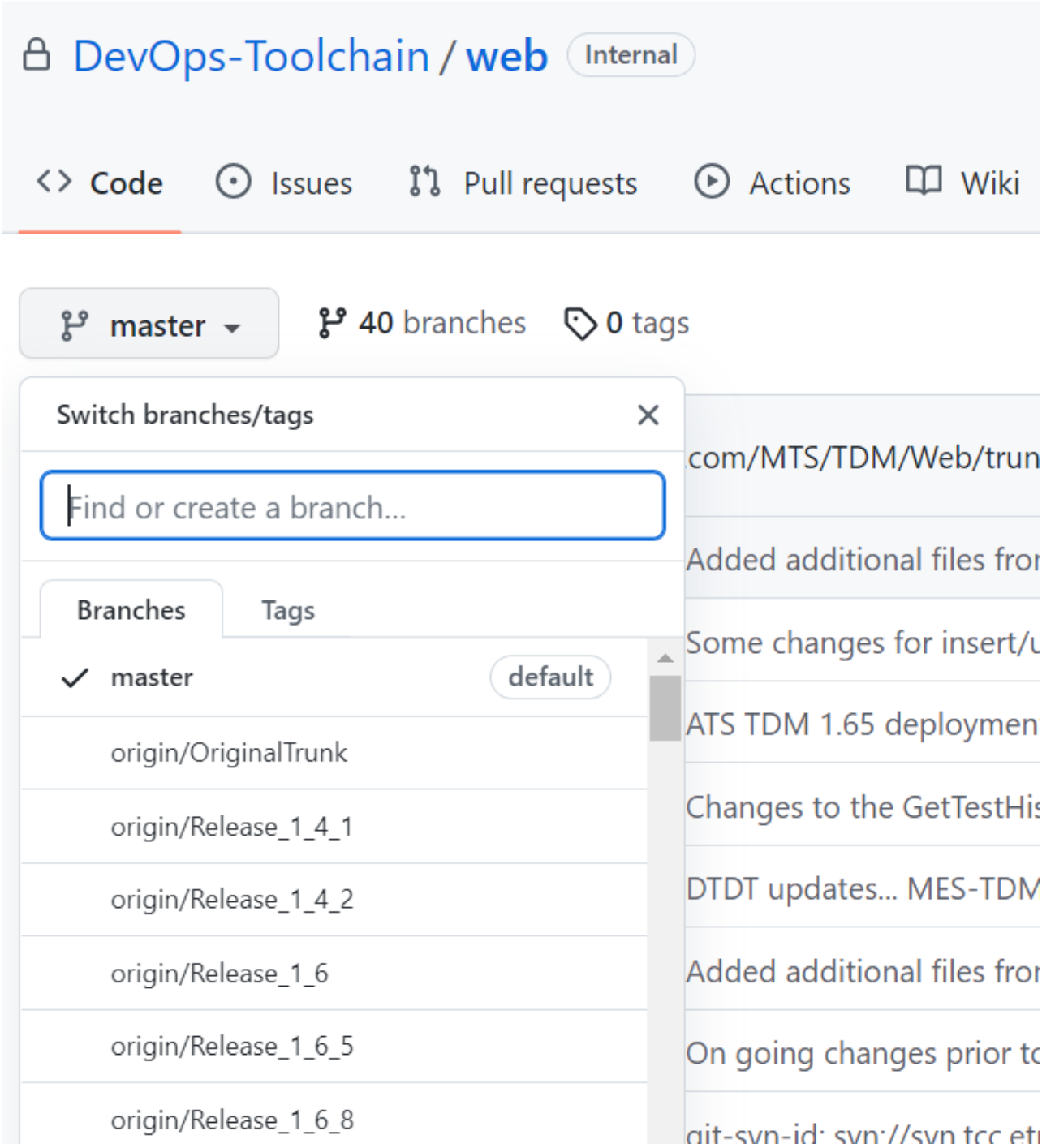
Check your repository data in the remote repository along with the history



Push all other tags/branches to the remote repository.

```
E0492097@INPUNWHP6152451 MINGW64 ~/OneDrive - Eaton/Documents/SVN CLone/Test/Web
 (master)
$ git push -u origin --all
Enumerating objects: 5141, done.
Counting objects: 100% (4201/4201), done.
Delta compression using up to 8 threads
Compressing objects: 100% (1575/1575), done.
Writing objects: 100% (3602/3602), 123.35 MiB | 203.08 MiB/s, done.
Total 3602 (delta 2302), reused 3107 (delta 1860)
remote: Resolving deltas: 100% (2302/2302), completed with 259 local objects.
To https://github.com/DevOps-Toolchain/web.git
 * [new branch]      origin/1.2.1 -> origin/1.2.1
 * [new branch]      origin/1.2.1@4 -> origin/1.2.1@4
 * [new branch]      origin/OriginalTrunk -> origin/OriginalTrunk
 * [new branch]      origin/Release_1_4_1 -> origin/Release_1_4_1
 * [new branch]      origin/Release_1_4_2 -> origin/Release_1_4_2
 * [new branch]      origin/Release_1_6 -> origin/Release_1_6
 * [new branch]      origin/Release_1_65 -> origin/Release_1_65
 * [new branch]      origin/Release_1_6_5 -> origin/Release_1_6_5
 * [new branch]      origin/Release_1_6_8 -> origin/Release_1_6_8
 * [new branch]      origin/Release_1_6_9 -> origin/Release_1_6_9
 * [new branch]      origin/Release_1_6_9_1 -> origin/Release_1_6_9_1
 * [new branch]      origin/Release_1_6_9_1_Salem2 -> origin/Release_1_6_9_1_Sal
em2
 * [new branch]      origin/TDMCoreAPI -> origin/TDMCoreAPI
 * [new branch]      origin/TDMTestAPI -> origin/TDMTestAPI
 * [new branch]      origin/report_fixes -> origin/report_fixes
 * [new branch]      origin/report_fixes@26 -> origin/report_fixes@26
 * [new branch]      origin/tags/1-2-b -> origin/tags/1-2-b
 * [new branch]      origin/tags/1-2-b@27 -> origin/tags/1-2-b@27
 * [new branch]      origin/tags/1-2-c -> origin/tags/1-2-c
 * [new branch]      origin/tags/1-2-c@34 -> origin/tags/1-2-c@34
 * [new branch]      origin/tags/1-2-d -> origin/tags/1-2-d
```

Check all tags and branches data in the remote repository along with historical data and commits.



You have done a good job! Now from here, you can make the svn repository read-only and eventually stop using it and start using git. You have my best wishes for your git journey from here. 🙂