

Table of Contents

1. Skip-able Part	1
2. Backup Git Repository	1
2.1. Backup Git Repository Locally	1
2.1.1. Clone Everything in Git Repository	1
2.1.2. Clone Git Bare	1
2.2. Backup Git Repository on Multiple Platforms	2
2.2.1. Add Git Remote Repository	2
2.2.2. Push Git Bare	2

Git Repository Backup

Robertus Diawan Chris

1. Skip-able Part

Git is a version control system for tracking any changes in any set of files. It's basically a tool that makes tracking changes on your code easier, so you would know which line fix a certain bug and so on.

There're a lot of git remote repository such as github, gitlab, bitbucket, codeberg and so on. The most popular one is github, but like other platform, github could decide whether to let you continue in their platform or banned you in their platform. You can check an article about github banned developer account on <https://techcrunch.com/2019/07/29/github-ban-sanctioned-countries/> and it seems like github in the process of restoring access to all developer in iran, you can check the github CEO tweet (<https://twitter.com/nat-friedman/status/1346517149603438593>).

You might be thinking "well, i'm not on those country so why should i worry about that?". My point is not about the country that got banned, but about what would you do if your account got banned and all access to you code was gone. We can't take anything for granted these days so better start backup your git repository rather than regret it later.

2. Backup Git Repository

First of all, there're two ways you can backup a git repository that i knew. The first one is to backup locally in your machine, and the second one is to backup on multiple platforms or even on your own server.

2.1. Backup Git Repository Locally

There're two ways to backup your git repository locally on your local machine, that is clone everything (basically the normal 'git clone' command) and clone git bare.

2.1.1. Clone Everything in Git Repository

You can backup every file in your git repository by doing the normal 'git clone <git-url.git>' command. By doing that, you download every file in your repository with their actual size and you can do git workflow in it. It's basically the usual way to clone a git repository.

2.1.2. Clone Git Bare

You can backup only the git bare from a git repository. To make it simpler, git bare is a '.git' directory when you initialize a directory as a git repository using 'git init' command.

So if you clone the git bare instead of the full git repository, you can get all commit history and all the branch on your git repository without actually downloading all your file so the git bare size is quite small.

You might ask "if you didn't download all your file then how you called that a backup?". Ok so here's the thing, by cloning a git bare you backup all your commit history and if you decide to upload your git repository to another git services or even your own server, you just need to push those git bare backup and git gonna create all your file with all the commit history on the new git repository. So basically you have your git repository backup without taking too much space on your local storage.

So, how can you backup using git bare clone? git have a backup mechanism but it's not obvious, git has a '--mirror' flag that you can use to backup or push a git bare repository.

First, you need to clone a git bare from git repository by doing

```
git clone --bare <git-repo-url.git>
or
git clone --mirror <git-repo-url.git>
```

Both of the commands are basically the same, you can push using ‘--mirror’ flag or add git remote repository and then push using the usual command. How to push with git bare repository is later in this post.

Here’s an example, if i want to clone my ‘instasaver’ git bare then the command i need is

```
git clone --bare https://github.com/bruhtus/instasaver.git
or git clone --mirror https://github.com/bruhtus/instasaver.git
```

and then it’s gonna make ‘instasaver.git’ directory where it stores all the commit history and branch.

If your git repository contains Git Large File Storage (LFS) objects then you need to download those git LFS objects too. First, you clone git bare with one of the command above (‘--mirror’ flag or ‘--bare’ flag) and then download git LFS object with this command

```
git lfs fetch --all
```

Don’t forget to change directory (‘cd’) into your git bare directory before applying those command.

2.2. Backup Git Repository on Multiple Platforms

There’re two ways you can backup your git repository on multiple platforms. First, you can use ‘git remote add’ command, and second, you can push git bare repository into another git services. Uploading into your own server is beyond the scope of this post, so i’m not gonna explain about that here.

2.2.1. Add Git Remote Repository

With ‘git remote add’ you can add another platform so that you can also push into that platform.

For example, your main git repository service is github so you usually do

```
git remote add origin <github-repo-url.git>
```

in your local git repository. That command is basically assign ‘origin’ as your github repository, so if you want to push into your github repository you need to do

```
git push origin <branch>
```

if you didn’t like ‘origin’ as your github repository alias, you can change it to whatever you want, you can change it into ‘github’ or even ‘anu’ but you also need to change your ‘git push’ command like this

```
git push github <branch>
```

or

```
git push anu <branch>
```

So if you want to add another platform and you have your full git working directory (not git bare version), you just need too assign another git repository service url.

For example, i want to add instasaver to gitlab. I just need to do

```
git remote add gitlab https://gitlab.com/bruhtus/instasaver.git
```

and then if i want to push the changes to master branch, i just need to do

```
git push gitlab master
```

other git workflow such as ‘git add’ and ‘git commit’ is the same. Don’t forget to change directory (‘cd’) into your local git repo directory.

2.2.2. Push Git Bare

If you already backup your git bare repository on your local machine, then all you gotta do is change directory (‘cd’) into your ‘git-bare-repo.git’ directory and then do

```
git push --mirror <new-git-repo-url.git>
```

or

```
git remote add <platform> <new-git-repo-url.git>  
git push <platform> <branch>
```

You can change '`<platform>`' name to whatever you want such as 'anu' or 'nganu', you do you.

The difference between both command is that you can get all the branch if you use '`git push --mirror`' meanwhile you only get main branch if you use the usual '`git push`' command (without '`--mirror`' flag). So if you want to backup all the branch then you should probably use '`git push --mirror`'.

The disadvantage of using git bare is that you cannot use git workflow such as '`git add`' or '`git commit`', you can only use '`git push`' command. If you want to update your git bare repository, you can do

```
git remote update
```

inside your git bare directory.

If you have Git Large File Storage (LFS) objects, you need to push git LFS object separately using this command

```
git lfs push --all <new-git-repo-url.git>
```

after you download all git LFS objects using '`git lfs fetch --all`' command.