Computer Setup Programming

Robin Hellmers

November 3, 2021

Contents

1 Git

1.1 Multiple repositories

1.1.1 Repository across 2 computers

Relevant StackOverflow **answer**

One might want to develop code on two machines, without using a server such as Github or similar.

Lets say you have created a repository and developed on **Computer 1**. That could be only locally or still through some server which you might not want to access through the other computer.

So you have a repository and the working directory on **Computer 1**, could also be on **Computer 2**, does not matter:

`~/<pathToRepo>/<repoName>`

which contains a `.git` with all the git related revision history.

Now you want to be able to push to **Computer 2**. In order to do that you need

Computer 1:

- Normal repository (non-bare) with the working directory

Computer 2:

- Normal repository (non-bare) with the working directory.
- Bare repository

The usage will be:

1. From **Computer 1** working directory: Push to the bare repository on **Computer 2**
2. From **Computer 2** working directory: Pull from bare repository on **Computer 2**.

or the other way around

1. From **Computer 2** working directory: Push to the bare repository on **Computer 2**.
2. From **Computer 1** working directory: Pull from the bare repository on **Computer 2**

Instructions

Lets say you have the original repository with its working directory on **Computer 1** with its `.git` directory in `~/<pathToRepo>/<repoName>/`

1. **Computer 1**: Make a bare clone of the repository with the name extension `.git`

   `git clone --bare ~/<pathToRepo>/<repoName> ~/<pathToRepo>/<repoName>.git`

2. **Computer 1**: Copy it over to **Computer 2**, using SSH.

   `scp -r ~/<pathToRepo>/<repoName>.git <username>@<ipAddress>:<relativePathBare>`

   or

```
scp -r ~/<pathToRepo>/<repoName>.git ssh://<username>@<ipAddress>/<fullPathBare>
```

3. **Computer 2**: Clone the bare repository into a non-bare repository.

```
git clone ~/<relativePathBare>/<repoName>.git ~/<pathToStoreRepo>/.
```