We're now writing and committing code like a developer. The following assumes that you have .NET 6 installed on a machine acting as your "developer workstation". I'm using .NET 6 and PowerShell Core on Windows, but since these are both cross-platform, you can use whichever OS you like.

First, clone the empty project.

```
PS C:\Tools> git clone http://172.23.157.233/rasta/mycalculator.git

Cloning into 'mycalculator'...
warning: You appear to have cloned an empty repository.
```

Move into the new directory and switch to the main branch.

```
PS C:\Tools> cd .\mycalculator\

PS C:\Tools\mycalculator> git switch -c main
Switched to a new branch 'main'
```

Create a new dotnet solution.

```
PS C:\Tools> dotnet new sln
The template "Solution File" was created successfully.

PS C:\Tools\mycalculator> ls

Directory: C:\Tools\mycalculator

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a---          14/11/2022    14:35            441 mycalculator.sln
```

Now we can commit that change and push it back to GitLab.

```
PS C:\Tools\mycalculator> git add .
PS C:\Tools\mycalculator> git commit -m "create solution"
[main (root-commit) da41cea] create solution
1 file changed, 14 insertions(+)
create mode 100644 mycalculator.sln

PS C:\Tools\mycalculator> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 444 bytes | 444.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://172.23.157.233/rasta/mycalculator.git
* [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```