# Configuring GIT on Windows

## Introduction

I had to do a full reinstall of Windows after my computer was infected with a pretty nasty virus. I ran into a problem after reinstalling GIT: when I tried to clone my archives from GitLab I got the following error:

D:\tmp\GitTests>git clone git@github.com:gbukauskas/BlazorCodeDemo.git

Cloning into 'BlazorCodeDemo'...

git@github.com: **Permission denied (publickey).**

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists

The reason for the error is clear: the standard installation of GIT uses a certificate authentication when the archive is cloned using SSH. After reinstalling Windows, this certificate was lost, so GIT won't let me clone my code. The same problem occurs when you try to access your archive from another computer.

There are two solutions:

1. Switch on HTTPS protocol,
2. Configure SSH registering new public key on Github (GitLab):

## SSH

Windows 11 now has native support for **OpenSSH**, meaning you can use the built-in SSH client instead of a third-party client like Putty. This is a significant change for Windows users, eliminating the need to download and install a standalone SSH client. [Geekrewind.com](https://geekrewind.com/how-to-install-openssh-client-in-windows-11/) provides instructions on how to activate OpenSSH. Follow this site. Verify an installation typing command **ssh** in the terminal. You would get this answer:

>**ssh**

usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind\_interface] [-b bind\_address]

[-c cipher\_spec] [-D [bind\_address:]port] [-E log\_file]

[-e escape\_char] [-F configfile] [-I pkcs11] [-i identity\_file]

[-J destination] [-L address] [-l login\_name] [-m mac\_spec]

[-O ctl\_cmd] [-o option] [-P tag] [-p port] [-Q query\_option]

[-R address] [-S ctl\_path] [-W host:port] [-w local\_tun[:remote\_tun]]

destination [command [argument ...]]

Open services window, start OpenSSH Authentication Agent and change Startup Type to Automatic or Manual. The second startup type preserves RAM but will require to start the Agent before connecting to GitHub or GitLab.

Generate a pair of public and private keys emitting this command:

ssh-keygen -t ed25519

The instruction for generating a key is provided on [Microsoft](https://learn.microsoft.com/en-us/windows-server/administration/openssh/openssh_keymanagement) site. The article offers **ecdsa** algorithm. Ssh-keygen supports 6 different algorithms:

1. dsa,
2. ecdsa,
3. ecdsa-sk,
4. ed25519,
5. ed25519-sk,
6. rsa.

I don’t know which is the best. Consult the [goteleport](https://goteleport.com/blog/comparing-ssh-keys/) site for selecting the best one.

**ssh-keygen** allows you to specify the directory where both keys will be saved. Make a note of this value and press Enter without changing anything. Placing the keys into a different directory will require you to adjust the global GIT settings. It's easy, you just need to type this command:

ssh-add ~/yourKeyFolder/yourPrivateKey

You will also need to specify a password. Don't generate a very long and 'secure' password, as you will have to type it yourself every time you access GitLab or GitHub. Change your SSH keys and password periodically instead.

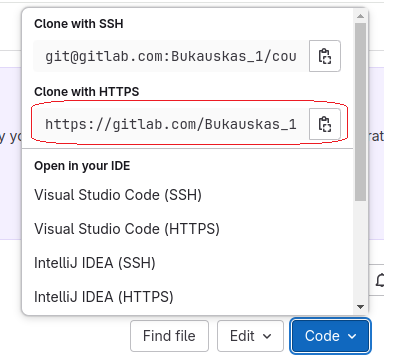
You will find the private and public keys in the selected directory. Public key has the **.pub** extension. Open it in any text editor, connect to your repository on GitHub or GitLab and follow the instructions provided:

1. Click on avatar; **GitLab** displays it on left side of top row, **GitHub** - on right side,
2. Select "**Edit Profile/SSH keys**" on GitLab, "**Settings/SSH and GPG keys**" on GitHub,
3. Click on "**Add new key**" (GitLab), "**New SSH key**" (GitHub),
4. Copy and paste public key from your editor into the new key dialog.

## SSL (HTTPS)

You can download an archive from GitHub or GitLab using SSL (HTTPS). To do this, simply specify the appropriate link in the **git clone** command:

git clone {your\_https\_address}



The **GIT clone** command works exactly the same as the SSH version:

