**Setting Up SSH for Git on Windows**

Why Use SSH Instead of HTTPS?

When working with GitHub, GitLab, or Bitbucket, you can connect to repositories over HTTPS or SSH. While both methods work, SSH is the recommended option because:

* Security: SSH keys use strong cryptographic authentication, eliminating the need to send your username and password for every interaction.
* Convenience: With SSH, you don’t need to enter credentials every time you push, pull, or clone. Once set up, authentication is seamless.
* Compatibility with Personal Access Tokens: Git providers are phasing out password authentication. HTTPS requires you to use personal access tokens (PATs), which can be cumbersome to manage compared to a one-time SSH setup.
* Automation-friendly: SSH keys integrate smoothly into scripts, CI/CD pipelines, and development environments.

Prerequisites:

* A Windows machine
* Git installed ([Download here](https://git-scm.com/downloads))
* A GitHub, GitLab, or Bitbucket account

Step 1: Check for Existing SSH Keys

1. Open Git Bash (comes with Git for Windows).
2. Run:

***ls -al ~/.ssh***

1. If you see files like id\_rsa.pub or id\_ed25519.pub, you already have an SSH key. You can reuse it or generate a new one.

Step 2: Generate a New SSH Key

1. In Git Bash, run:

ssh-keygen -t ed25519 -C „[your\_email@example.com](mailto:your_email@example.com)“

* Use your GitHub/GitLab email address.
* If your system doesn’t support Ed25519, use RSA instead:

ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

1. When prompted:

* Press Enter to accept the default file location.
* Optionally set a passphrase for added security.