**[How to generate a new SSH key and adding it to the ssh-agent](https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/)**

**Generating a new SSH key**

1. Open Git Bash.
2. Paste the text below, substituting in your GitHub email address.
3. ssh-keygen -t rsa -b 4096 -C "*your\_email@example.com*"

This creates a new ssh key, using the provided email as a label.

Generating public/private rsa key pair.

1. When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.
2. Enter a file in which to save the key (/c/Users/*you*/.ssh/id\_rsa):*[Press enter]*
3. At the prompt, type a secure passphrase. For more information, see ["Working with SSH key passphrases"](https://help.github.com/articles/working-with-ssh-key-passphrases).
4. Enter passphrase (empty for no passphrase): *[Type a passphrase]*
5. Enter same passphrase again: *[Type passphrase again]*

**Adding your SSH key to the ssh-agent**

Before adding a new SSH key to the ssh-agent to manage your keys, you should have [checked for existing SSH keys](https://help.github.com/articles/checking-for-existing-ssh-keys) and [generated a new SSH key](https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key).

If you have [GitHub Desktop](https://desktop.github.com/) installed, you can use it to clone repositories and not deal with SSH keys. It also comes with the Git Bash tool, which is the preferred way of running git commands on Windows.

1. Ensure the ssh-agent is running:
   * If you are using the Git Shell that's installed with GitHub Desktop, the ssh-agent should be running.
   * If you are using another terminal prompt, such as Git for Windows, you can use the "Auto-launching the ssh-agent" instructions in "[Working with SSH key passphrases](https://help.github.com/articles/working-with-ssh-key-passphrases)", or start it manually:
   * # start the ssh-agent in the background
   * eval $(ssh-agent -s)
   * Agent pid 59566
2. Add your SSH private key to the ssh-agent. If you created your key with a different name, or if you are adding an existing key that has a different name, replace *id\_rsa* in the command with the name of your private key file.
3. ssh-add ~/.ssh/id\_rsa
4. [Add the SSH key to your GitHub account](https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account).

Basic commands

git clone git://git.kernel.org/pub/scm/git/git.git (update the version)

git add <filename> (adding to staging. -can use \* wildcard)

git add . (all)

git commit -m “Commit msg” (commit to changes)

git push origin master (send changes to the master branch)

git init (start a new repository)

git config –global user.name ‘name’

git config –global user.email ‘me@email.com’

git rm –cached index.html (or use wild card \*.html which will delete any file with .html extension)

git commit (commit changes. Press “I” to insert message and type “:wq” to exit)

git commit -m “msg”

git branch login (create another branch – doing this so that master does not change)

git checkout login (switch to another branch)

git merge (merging another branch to the master)

git remote add origin <https://github.blabla> (connect local to remote repo)

Creating a new repo

echo "# Development" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin git@github.com:arellanodaniel12/Development.git

git push -u origin master

OR if you already have a folder

git add .

git commit -m “msg”

git remote add origin git@github...

git push -u origin master

**Day to Day command**

$ git status (see if there is any changes to the app or data)

$ git add . (moving all to the staging area)

$ git commit -m “My message” (commiting to all changes)

$ git push (send changes to the remote repo)

$ git pull (merge changes from remote to local directory)