

How to connect to Mimi & use VSCode

COMP310 – ECSE427 Operating Systems

1. Obtain a CS account

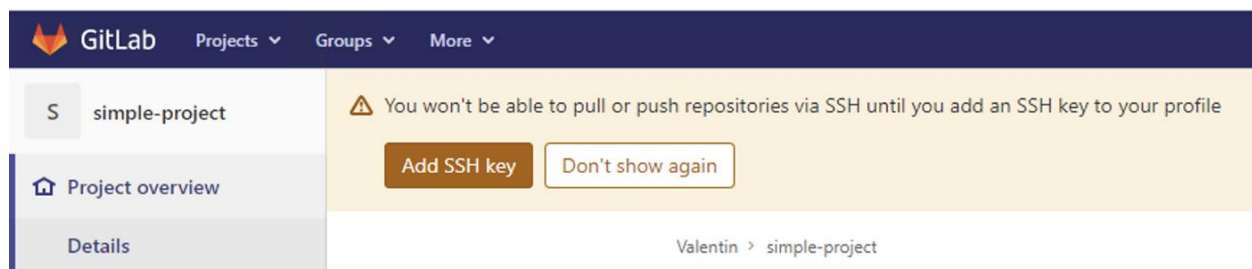
Go to <https://myaccount.cs.mcgill.ca/> to obtain a CS account. More information here: <https://www.cs.mcgill.ca/docs/>

2. Log in to CS GitLab

Use your CS account credentials created in previous step to log in to GitLab at <https://gitlab.cs.mcgill.ca>

3. SSH keys

On first login, GitLab will give you a message like this:



To fix this, you will need to generate an SSH key pair and add your public key to GitLab. First we will ensure we have access to SSH.

SSH for Mac & Linux

You should already have SSH installed on your machine by default. You can skip to “Generating SSH keys”

SSH for Windows

You have multiple options on how to use SSH on Windows.

- **Windows Terminal** (simplest). Follow these instructions to install Windows Terminal: <https://learn.microsoft.com/en-us/windows/terminal/install> . SSH is included with the Windows Terminal installation.
- Git Bash. Install git on Windows through here: <https://git-scm.com/download/win> . It will include a program called Git Bash that you can use as your terminal, it includes SSH.
- Windows Subsystem for Linux. This is overkill, but you can install it using these instructions: <https://learn.microsoft.com/en-us/windows/wsl/install>
- PuTTY. It's an SSH client application. You launch PuTTY and give it your CS credentials and the Mimi hostname. <https://www.putty.org/>
- Cygwin. Not recommended due to complexity. <https://devtidbits.com/2011/07/01/cygwin-walkthrough-and-beginners-guide-is-it-linux-for-windows-or-a-posix-compatible-alternative-to-powershell/>

Generating SSH keys

SSH keys are based on public key cryptography and work as unique and secure identity documents for your computer. They allow you to certify that you are who you claim you are. This enables you, for example, to log in to servers without using a password.

The idea is that you will create 2 files: a **public** and a **private** one. You will copy the public key file on every machine you want access to without using a password.

When you try authenticating without password to a machine that knows your public key; it will authenticate you by asking your machine a question that can only be answered by knowing what the private key is. Thus, it will be your responsibility to ensure the private key remains private.

Obviously, most of this will happen in the background. On your own computer, you can generate the files using the following command on your terminal:

```
ssh-keygen
```

Press enter on every question until you get a diagram. That command creates the files `~/.ssh/id_rsa` which is your **private** key, and `~/.ssh/id_rsa.pub` which is your **public** key and can be copied to any machine you like. These files are generated in such a way that there is no way you could have created one without the other.

Uploading SSH Key to Gitlab

For Mac and Linux, run:

```
cat ~/.ssh/id_rsa.pub
```

For Windows:

```
cd  
type .ssh\id_rsa.pub
```

It will output a string that looks like gibberish, this is your public key. Copy all the output. Now follow these instructions:

1. Sign in to GitLab.
2. On the top bar, in the top right corner, select your avatar.
3. Select **Preferences**.
4. On the left sidebar, select SSH Keys.
5. In the **Key** box, paste the contents of your public key. If you manually copied the key, make sure you copy the entire key, which starts with `ssh-rsa`, `ssh-dss`, `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp384`, `ecdsa-sha2-nistp521`, `ssh-ed25519`, `sk-ecdsa-sha2-nistp256@openssh.com`, or `sk-ssh-ed25519@openssh.com`, and may end with a comment.
6. In the **Title** box, type a description, like Work Laptop or Home Workstation.
7. Optional. Remove the **Expiration date** by clicking “x” next to the date.
8. Select **Add key**.

4. Log in to Mimi

Note: In many of the steps below you may get a warning like this:

```
The authenticity of host 'mimi.cs.mcgill.ca (132.206.51.22)' can't be established.  
ED25519 key fingerprint is SHA256:d5h9yQaHbRsBztzG343Dfiur6gpXTfgAK49FNtnogpA.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

This is fine. You can type `yes` and then enter.

Place your SSH keys into Mimi

In the previous step you created an SSH keypair for your computer and you let GitLab know who you are. If you were to log in to Mimi now and tried to clone, pull, or push code, GitLab won't let you do it, because your Mimi account is not authorized. GitLab doesn't know yet that it's you when you're logged into Mimi. There are 2 ways to fix this:

Option 1: Copying the key files from your computer to Mimi

You can copy the key files that you have created on your computer into our Mimi account. Run the following command:

```
cd  
scp -r .ssh/ <YOUR_CS_USERNAME>@mimi.cs.mcgill.ca:~
```

Option 2: Creating a new SSH keypair for Mimi only

If you're getting an error or you're having trouble with Option 1, follow the next steps of the guide until you have logged into Mimi. Once you're logged in, you will create an SSH key pair specifically for Mimi only.

Once you're logged into Mimi, follow step 3 again (generate the SSH keys and upload to GitLab). You can have as many SSH keys uploaded to GitLab as you want.

Enable passwordless access to Mimi

We will now enable access to Mimi without having to input our password. Run:

```
ssh-copy-id <YOUR_CS_USERNAME>@mimi.cs.mcgill.ca
```

If you are on Windows and get an error saying that `ssh-copy-id` is not found, try this command in **Windows PowerShell** instead:

```
type $env:USERPROFILE\.ssh\id_rsa.pub | ssh  
<YOUR_CS_USERNAME>@mimi.cs.mcgill.ca "cat >> .ssh/authorized_keys"
```

Log in to Mimi

To log in simply use this command:

```
ssh <YOUR_CS_USERNAME>@mimi.cs.mcgill.ca
```

When you log in, sometimes you'll get a message telling you that there are other users, or that there are firmware or software updates. You can ignore all of this. Mimi is a shared server, and those messages are meant for the SOCS administrators.

You can get more details on how to use Mimi and Linux following the SOCS instructions:

<https://www.cs.mcgill.ca/docs/tutorials/linux/>

Configuring your name and email in Git

If it's your first time logging into Mimi, it is possible that Git doesn't know how you are yet, and this can give you some trouble when committing and pushing code. When logged into Mimi, run the following commands:

```
git config --global user.name "<YOUR NAME>"  
git config --global user.email "<YOUR MCGILL EMAIL>"
```

The name that you choose doesn't have to be your official name. You can use your preferred name. But do make sure you use your McGill email.

5. Use VSCode to edit code on Mimi from your local machine

You can use Visual Studio code as the text editor for this course. The setup we recommend is for you to use VSCode on your local machine, while the files being edited are actually stored on Mimi.

Follow these instructions to set up the remote editor for VSCode:

<https://code.visualstudio.com/docs/remote/ssh>

With this setup, you can run VSCode and keep a terminal window open where you compile the latest changes to your code and use the testcases to test your program.