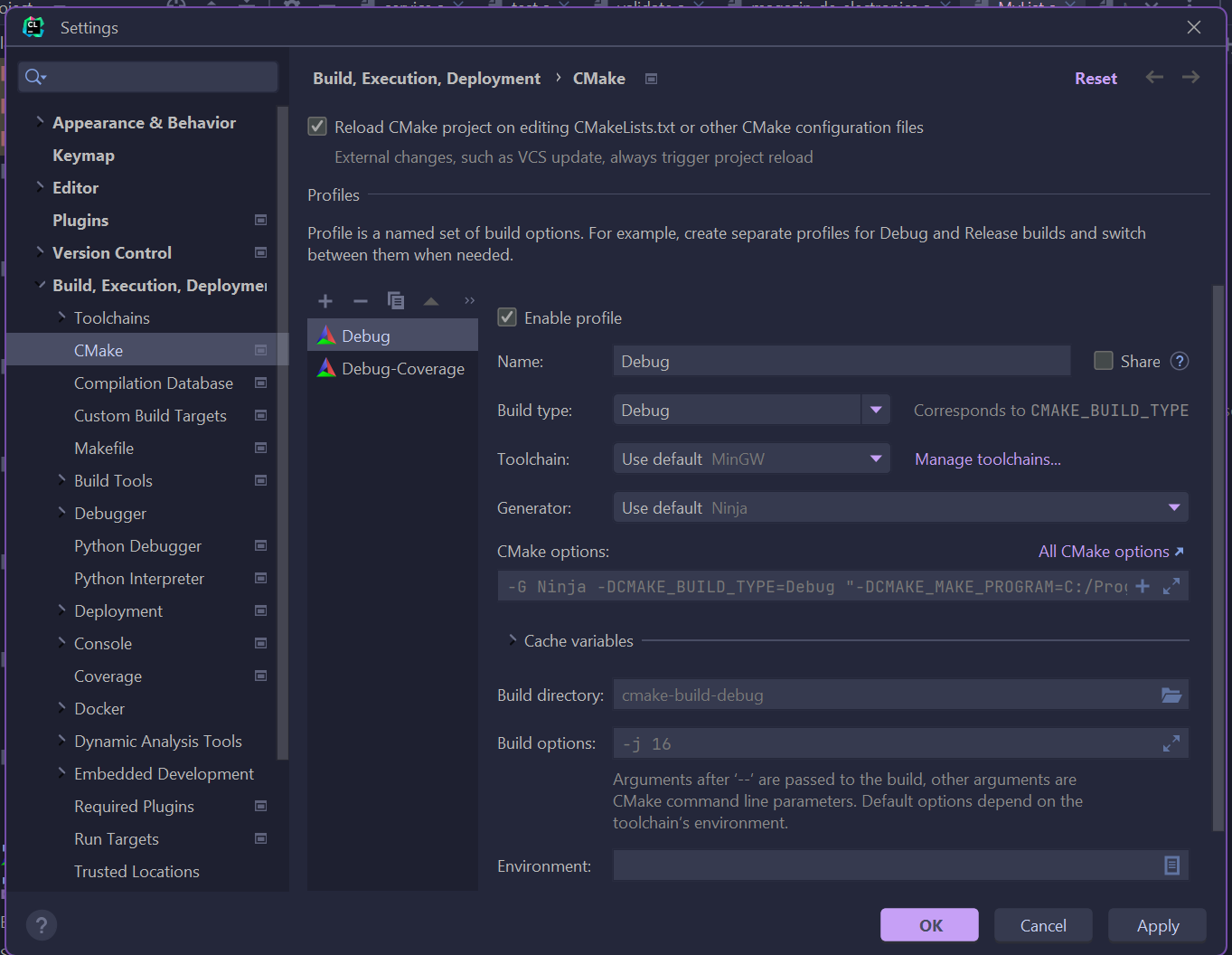
***Configuring Valgrind on Clion***

**After having installed everything, now it is time for the Clion configurations.**

**Step 1: Go to File-> Settings -> Build, Execution, Deployment, Cmake .  
 Or use the key-combination : Ctrl+ Alt + S**



*Now we need to add your CMake configuration! ( YOU HAVE TO DO THIS FOR EVERY PROJECT! It will not save across projects unfortunately…. )*

This is what it should open:

Graphical user interface, text, application

Description automatically generated

Now we need to name it and change some of the fields ( TO THIS EXACT SETTINGS).

I will just name it Valgrind.

This should be all for it. Click apply and then OK to save.

*A screenshot of a computer

Description automatically generated*

Now go back to the same window as before but go to **TOOLCHAINS**.

A screenshot of a computer

Description automatically generated with medium confidence

We need the WSL configuration. So, click plus and select it.

For me it found them all automatically .

A screenshot of a computer

Description automatically generated with medium confidence

Also just to be sure you should check if you have the right location for your Valgrind.

Go to **Dynamic Analysis Tools**, click it and then go to Valgrind.

And see if it found it in your WSL configuration. If it is not detected automatically, it should be in **usr\bin** in your WSL files.

Now you can click OK.

A screenshot of a computer

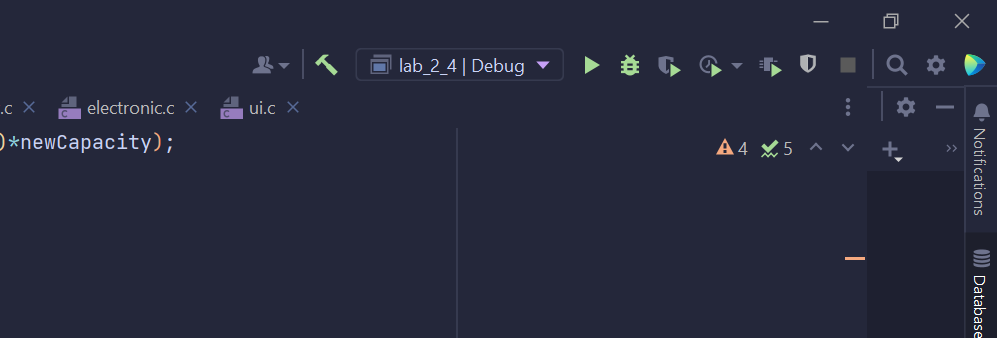
Description automatically generated with medium confidence

YOU ALSO NEED TO MAKE SURE YOU HAVE THE SAME VERSION FOR CMAKE IN CMakeList.txt!!! Please check your cmake version in ubuntu. You can see it when you go to Toolchains but also through the command in the ubuntu terminal “ cmake –version”.

Text

Description automatically generated

Now look in the upper right corner:



Graphical user interface, application

Description automatically generated

Select Valgrind (or whatever name you chose)

Now just try to run it with Valgrind:

A screenshot of a computer

Description automatically generated with medium confidence

I commented something in the program to have a memory leak:  
Text

Description automatically generated  
Now go to Valgrind.  
And done ! It should show all of your memory leaks!

Text

Description automatically generated