# Ubuntu 22.04

## Copied from [liboqs](https://github.com/open-quantum-safe/liboqs)

Install dependencies

sudo apt install astyle cmake gcc ninja-build libssl-dev python3-pytest python3-pytest-xdist unzip xsltproc doxygen graphviz python3-yaml

Clone the repository and make for a shared library

git clone -b main https://github.com/open-quantum-safe/liboqs.git  
cd liboqs  
mkdir build && cd build  
cmake -GNinja .. -DBUILD\_SHARED\_LIBS=ON  
ninja

After compiling, we need to copy the lib folder and include folder. The default installation is in the /usr/local/ directory. so we’ll copy there.

sudo mv lib/\* /usr/local/lib/  
sudo mv include/oqs /usr/local/include/

## Working with liboqs-java

Normally, we would either have to implement the algorithms in java, or use the JNI in order to create an interface into the library. Luckily, this has already been done by the open quantum safe project. so all we have to do is build their project. They’ve gone with maven, so we just have to go through a couple extra steps to ensure that this will all compile and work correctly.

git clone -b master https://github.com/open-quantum-safe/liboqs-java.git  
cd liboqs-java  
java -version  
javac -version  
echo $JAVA\_HOME

You should have some output on your screen. For me mine looked like this:

$ java -version  
openjdk version "17.0.2" 2022-01-18  
OpenJDK Runtime Environment (build 17.0.2+8-Ubuntu-1)  
OpenJDK 64-Bit Server VM (build 17.0.2+8-Ubuntu-1, mixed mode, sharing)  
$ javac -version  
javac 17.0.2  
$ echo $JAVA\_HOME  
/usr/lib/jvm/java-17-openjdk-amd64/

I already did the setup for all of this. I used java 17 (openjdk) and my java\_home is at /usr/lib/jvm/java-17-openjdk-amd64/

If JAVA\_HOME is not set, or if java is not installed, please refer to your version of Java installer in order to install this correctly. After doing this, it should be a simple compile, package, and install, maven will take care of all of the hard work.

mvn compile  
mvn package  
mvn install

## QuantumJava

Finally, we can pull down this repository from github.

git clone https://github.com/RappelBerryPi/QuantumJava.git  
cd QuantumJava/QuantumJava  
mvn compile  
mvn package  
mvn install

After we install this, we will need to modify our jvm configuration to include the new class that provides the quantum cryptographic functions.

Edit $JAVA\_HOME/conf/security/java.security

sudo vim $JAVA\_HOME/conf/security/java.security

Add in the following line after your list of security providers. Please make sure that you replace ## with the next number in the sequence. Providers can also be dynamically added if required.

security.provider.##=com.thatsdarlingmama.security.QuantumProvider

We should be able to use these providers for assymetric key generation, Key Encapsulation, and Signatures at this point.