# TLS Lab Additional Information

## 1 Installing Qiskit:

Qiskit is the python based quantum computing software you will be using for this lab. Depending on what operating system you are using, installing qiskit can be a little tricky. It is very important that you have Python 3 installed first. See the following resources on installing python 3 for mac or for windows. It is important to note that if you are using a mac, you will want to download Xcode and Homebrew as well. These will allow you to download and install packages off of the internet. If you are using a PC, you will need to install Microsoft visual studio installer and specifically check the python development and desktop development with C++ packages. The latter is especially important for installing qiskit as it requires a C++ build tools version of 14.0 or higher in order to properly download. Make sure you have the relevant software add-ons for your OS before you attempt to install qiskit - this will save you a lot of time.

### 1.1 Pip install:

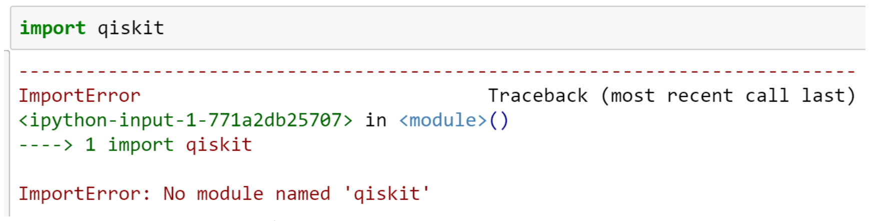
You will do all of the installation process using a terminal, and will use the python "pip install" command to install qiskit. To check if you have the pip package manager installed correctly, open your terminal and type in the command pip –version. If you see an output along the lines: pip 20.2.4 from c:\users\anaconda3 \...\pip (python 3.5), the pip package is correctly installed. If you see an error, you do not have the pip package manager installed on your version of python, even though you have just downloaded python. See installing pip for more information on how to download the package manager. **NOTE:** If the command "python get-pip.py" does not work, you may need to specify your python version. Changing the command to "python3 get-pip.py" on a mac worked.

There are two ways you can go about installing and using qiskit - by following the default installation process for python packages or by creating a virtual environment and installing qiskit there. **NOTE:** If you are using a PC, it is *strongly* recommended to use a virtual environment. The reason for this is there seems to be confusion between the directories your OS installs packages into and where jupyter notebook searches for packages. So even if you successfully install qiskit via your terminal *and* specify the directory you're installing into, the module may not be found when called in a jupyter notebook.

### 1.2 Installing Qiskit using the Default Method:

Open your terminal. This reference gives a good overview of how to successfully install python packages if you are using windows. If you are using a mac, consider looking at the following instructions on installing python modules. If you receive any errors while installing qiskit, qiskit's online documentation may be helpful for debugging. If you do not receive any errors and qiskit has successfully been installed, open a jupyter notebook. You can either do this from your terminal or by launching the 3rd party app if you have Anaconda. To launch a notebook from your terminal, cd into the directory you want to open your notebook in and then simply type "jupyter notebook" into the command prompt. To test if jupyter can find the qiskit modules, type "import qiskit" into the first line of your notebook.

If you receive something like the error on the right, this is likely due to qiskit being installed in a different directory than those jupyter is searching for. This can also happen if you have multiple versions of python installed on your computer. Rather than trying to identify the correct directory, pip install into that specific one, or jump through any other hoops, setting up a virtual environment and pip installing into that instead will save you a *lot* of time.



```
import qiskit

---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-1-771a2db25707> in <module>()
----> 1 import qiskit

ImportError: No module named 'qiskit'
```

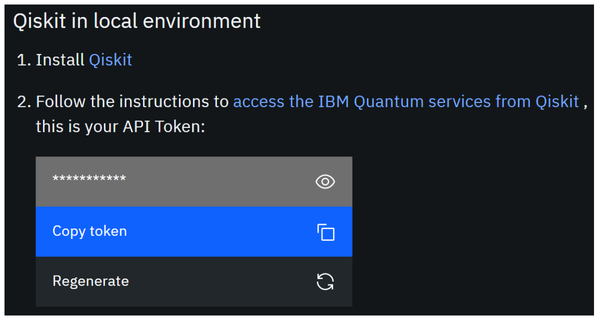
## 1.3 Installing Qiskit Using a Virtual Environment:

Virtual environments allow you to isolate the contents of your environment from the rest of your system's directories. Virtual environments are really useful when you have different python projects with different system prerequisites (versions of python, packages installed, etc.) such that you can pick and choose versions of code and which modules you want to run in your virtual environment without changing the way python works on your main system. Here, they are *especially* useful because of how easily they incorporate jupyter. By creating a virtual environment with its own set of source directories and installing python packages that are all isolated from the rest of your system, your OS and jupyter only have one place to pull from, and thus there is no confusion between the two systems. This is probably why you were receiving an error if you tried to install qiskit onto your main environment with the earlier method.

Open your terminal. Follow the instructions located on the qiskit website here to pip install qiskit. While still in your virtual environment, launch jupyter notebook from your terminal (NOT anaconda \any app on your computer). You can cd into the directory you want the notebook to be in first. You must do *all* of this while in your virtual environment so that all of the correct packages are stored in one place that is accessible by both your operating system and jupyter. In your open jupyter notebook, test to see if qiskit was successfully installed by typing "import qiskit" into your notebook. **NOTE:** matplotlib, numpy, and most of the other relevant python packages you'll need for the lab will already be installed in your virtual environment, so you won't have to pip install those packages separately. You should be good to go just starting a notebook from your terminal while still in your virtual environment.

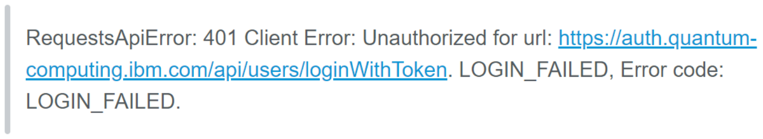
# 2 Setting Up the Qiskit Basics:

After setting up your IBMQE account, go to your profile page and copy your API token. You will need to specify this when you're loading your account into your jupyter notebook. Type the command
"IBMQ.save _ccount('your API token')"
into your notebook. If you receive an error along the lines of "credentials already in use," amend your command to "IBMQ.save _account('your API token', overwrite=True)" This should fix your error.



Qiskit in local environment

1. Install Qiskit
2. Follow the instructions to access the IBM Quantum services from Qiskit , this is your API Token:



RequestsApiError: 401 Client Error: Unauthorized for url: https://auth.quantum-computing.ibm.com/api/users/loginWithToken. LOGIN_FAILED, Error code: LOGIN_FAILED.

If you are getting an error that looks like the one on the left, simply regenerate your API token and re-copy it into your jupyter notebook.

## 3 Creating Quantum States Using Qiskit: