ECE 5/7384 Introduction to Quantum ComputingProgramming -- Lab #1

1.0 Overall Description of the Assignment

This homework assignment is designed to acquaint you with quantum computer programming using the IBM quantum computer simulator (Aer) and an actual IBM quantum computer (QiskitRuntimeService).

- First, you will run a simple quantum program that demonstrates superposition using the IBM simulator and again on an actual IBM quantum computer.
- Second, you will write, execute, basic quantum rotation operators, known as the Pauli matrices.

Also, note that there are questions embedded within the Jupyter notebook. You should answer these questions by entering your text in the cells. Equations can be entered using LaTeX format. You will turn in your homework assignment by populating all cells with source code (where required) and populating the markdown cells with answers to the questions. You will export your notebook to .html and email it to yayum@smu.edu,, and mitch@smu.edu by the due date. Be sure that your saved notebook has all of the output from each code cell present for full credit. If you are unfamiliar with Jupyter notebooks and Python, there are many web-based resources that will quickly get you up to speed. You **DO NOT** have to be an expert Python programmer to do this homework.

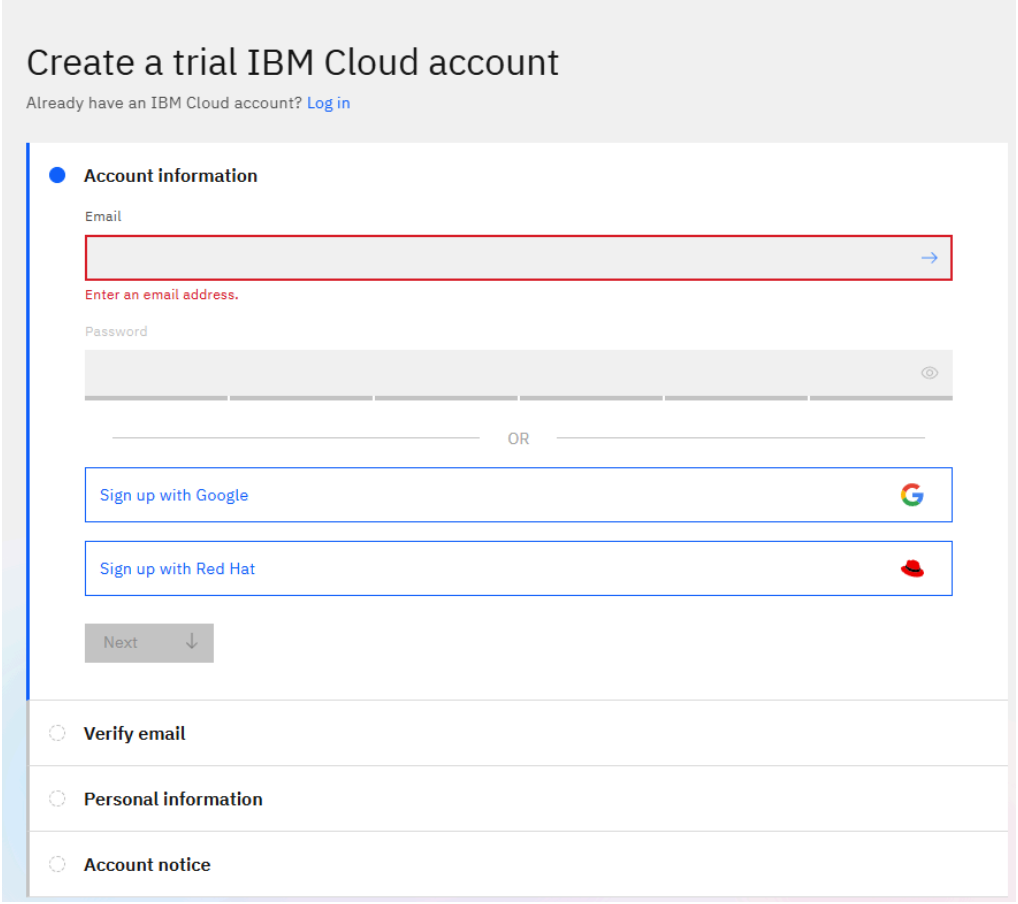
You will need to install or setup the following. Detailed instructions for each of these steps is provided below. Some of you may already have Anaconda/Jupyter installed. IMPORTANT: The IBM Q simulators and backends are only compatible with Python 3.10 (or later) – so you MAY need to install a new conda virtual environment and the Python 3.10 package if you are running a early version. Details are provided below.

1. Create an account on the IBM Quantum Platform website and generate API token and Instance.
2. Install and create an environment in Anaconda.
3. Install the **Python (must be 3.10 or later)**, **qiskit**, **qiskit-aer**, **qiskit-ibm-runtime**, and **matplotlib** libraries.
4. Install **Jupyter lab** (or Jupyter notebook) in Anaconda.

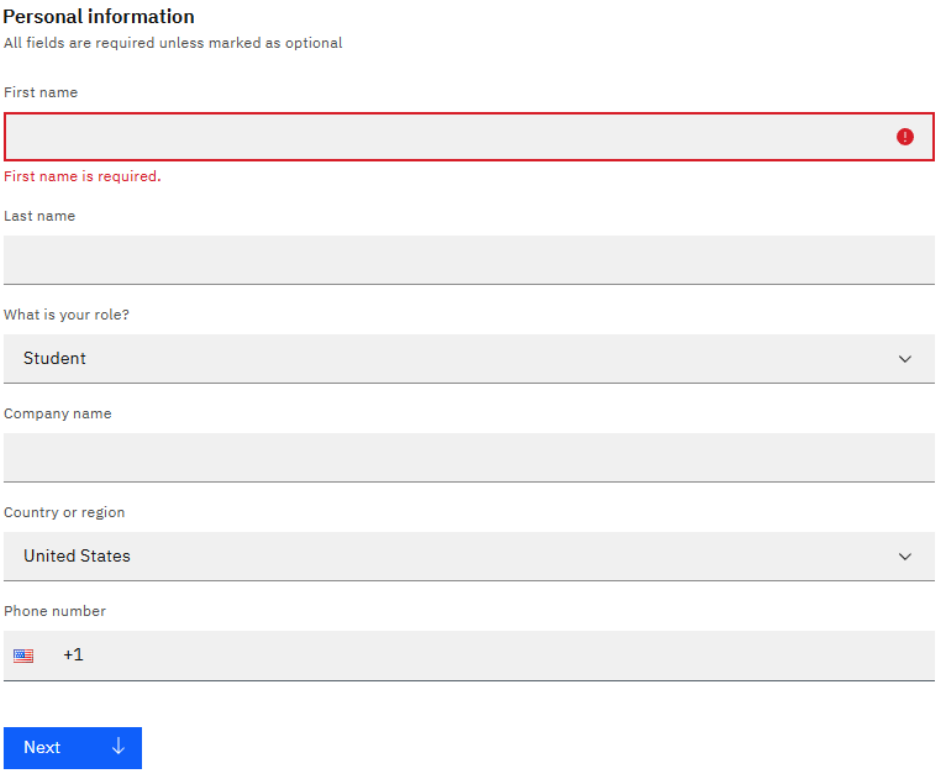
2.0 Creating an IBM Cloud Account

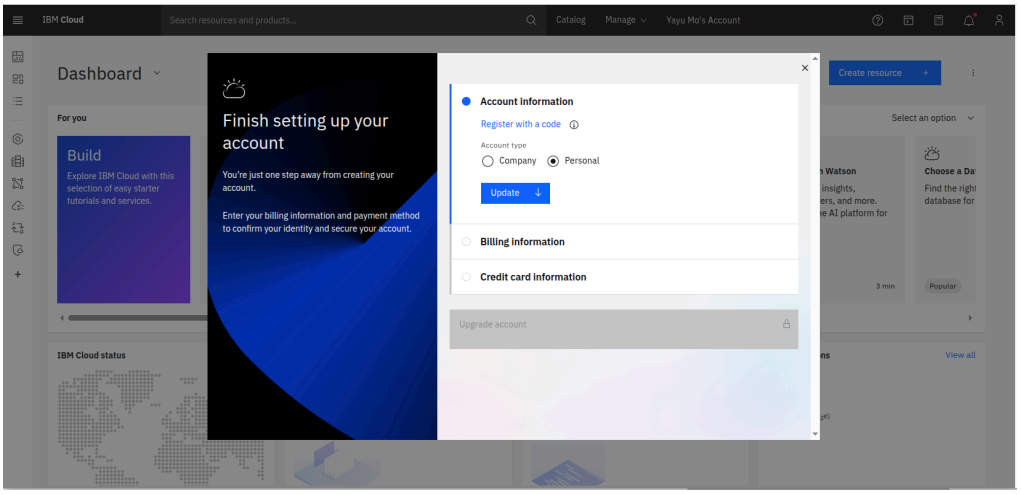
To run your program on actual quantum computing hardware, you will need to create an account that will provide you with an "API token." The API token is required to allow you to run your program on actual hardware, otherwise the simulator (Aer) is used.

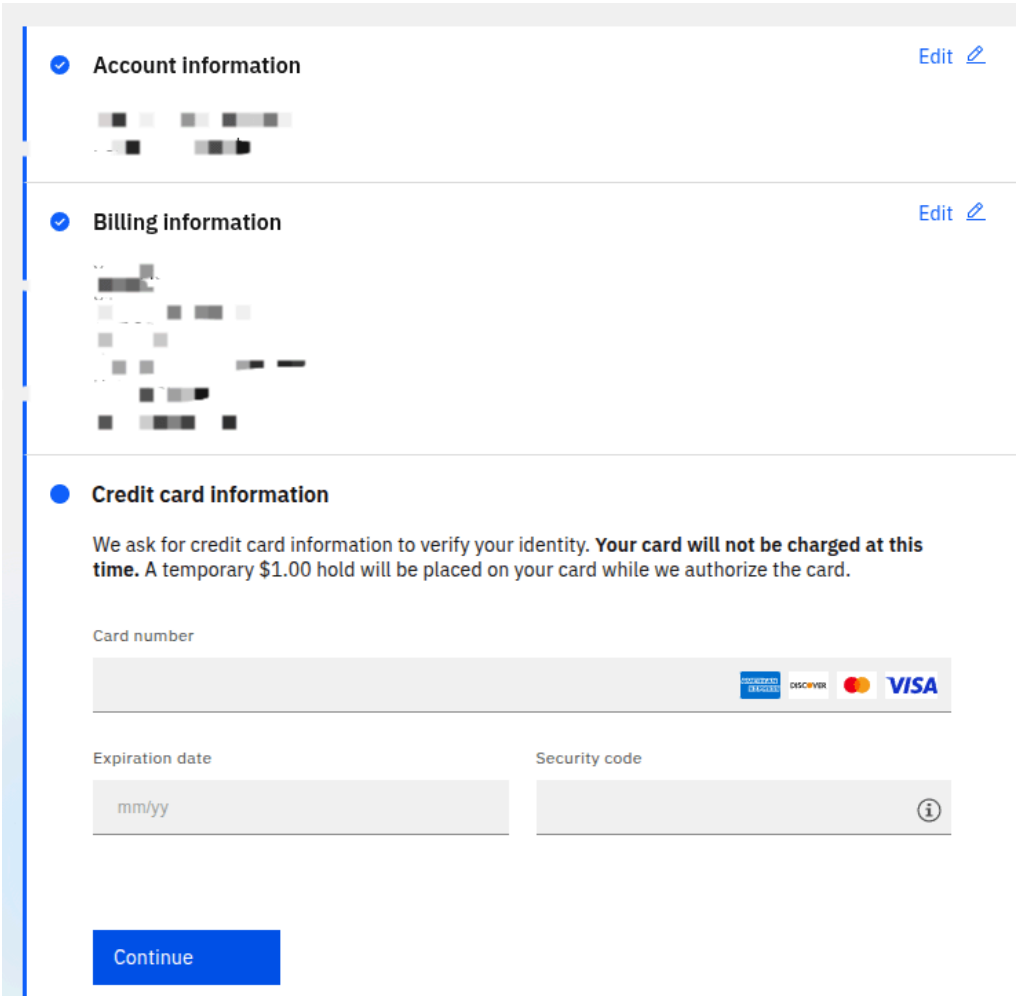
Instructions for creating your IBM account are provided on the **IBM Quantum Platform** official document at: https://quantum.cloud.ibm.com/docs/en/guides/cloud-setup. The document instructs you to do the following:

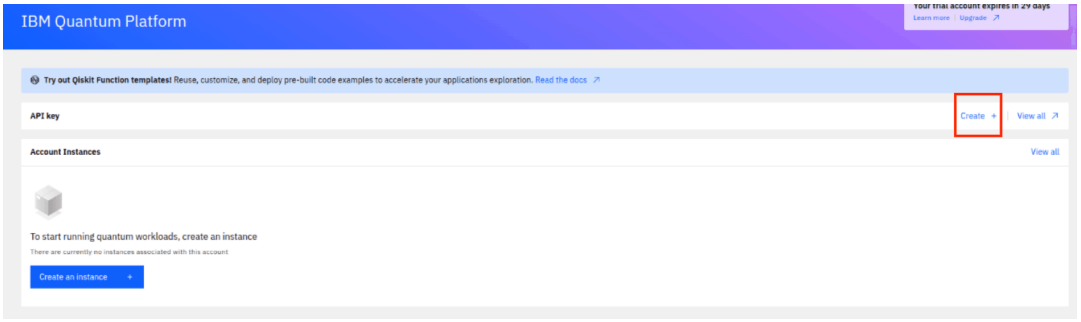
- (1) Create an IBM Quantum account login by visiting: https://cloud.ibm.com/registration?utm_content=quantum-trial&target=https%3A%2F%2Fquantum.cloud.ibm.com&error_uri=On the rightmost side of the webpage, it will guide you to create a 30-day trial IBM Cloud account for student, follow the instructions to create your account. If you already have an IBMid account, you can skip this step and just login using your IBMid credentials. There will be a problem regarding creating a trial account:
    - For a 30-day trial account:
        - You will first need to register using your email, set up a password.
        - 

        - And then follow the instruction of verifying email address, filling up your personal information.

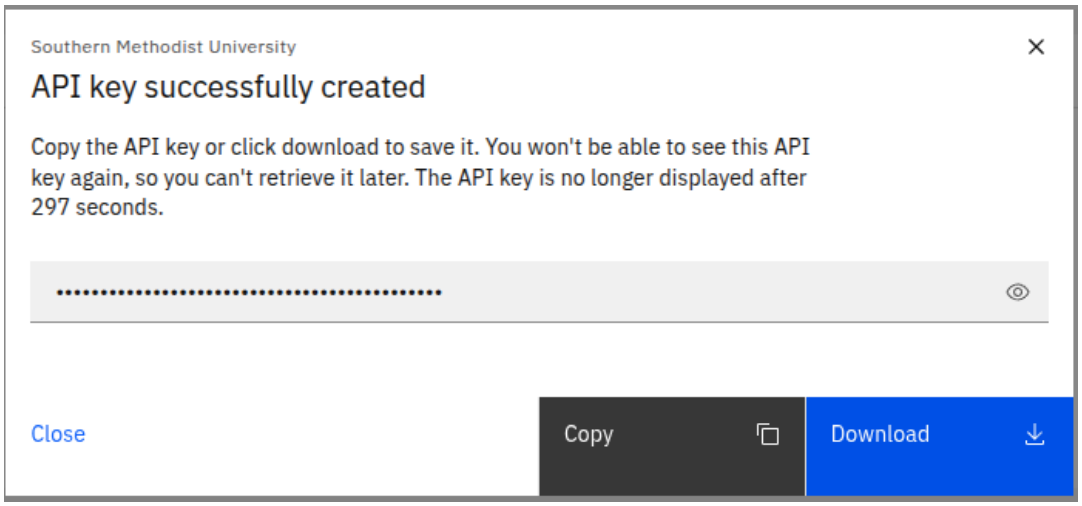


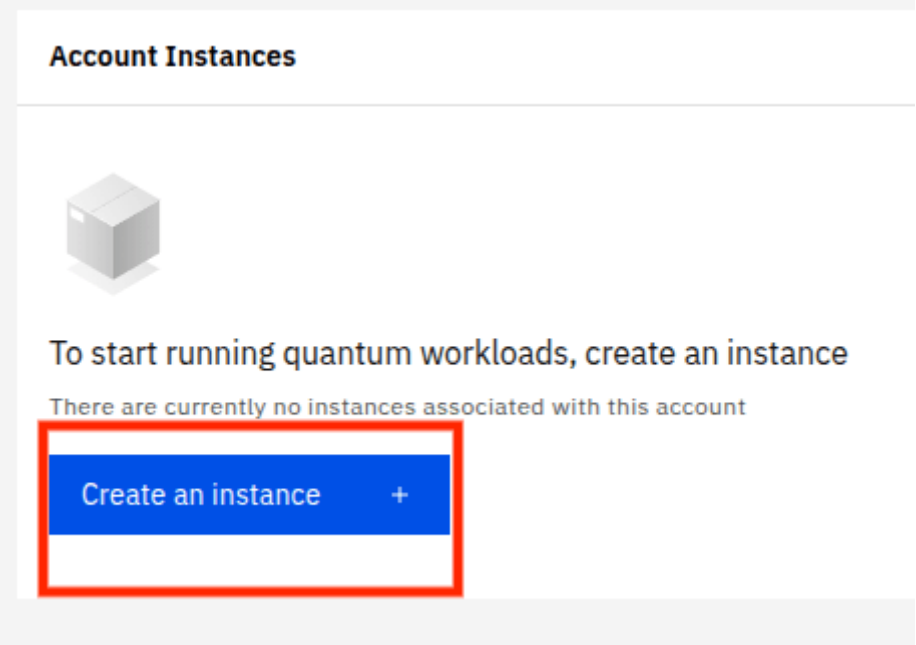
    - Check "I accept the product Terms and Conditions of this registration form". And then click "Create account".
    - Now your trial account has been successfully created. Note that one trial account per email address (Which means if you have multiple email address, you can create bunch of trial account, but IBM won't allow you to create multiple trial account using only one email.) will only be available for 30 days, so here we also provide another way of registering a full account.
- You can try to sign up a full account through https://cloud.ibm.com/registration.
    - After the same process as trial account, the IBM will ask for some additional information and credit card information every time when you are trying to view your dashboard, and you will not have the permission to create an instance.
    - 



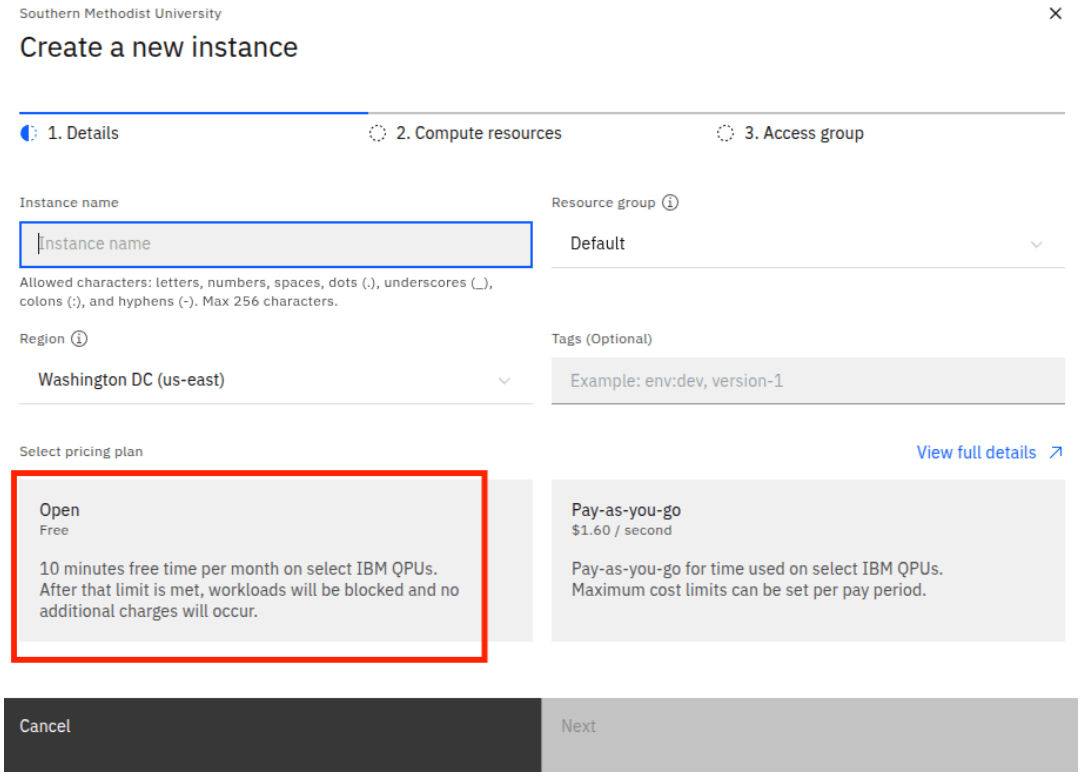
- (2) After creating an account, you will need to login to the **IBM Quantum Platform** dashboard (https://quantum.cloud.ibm.com/signin).
  - Then the first thing you need to do is creating an API key, click "Create+" at the right side of "API key" panel.
  - 

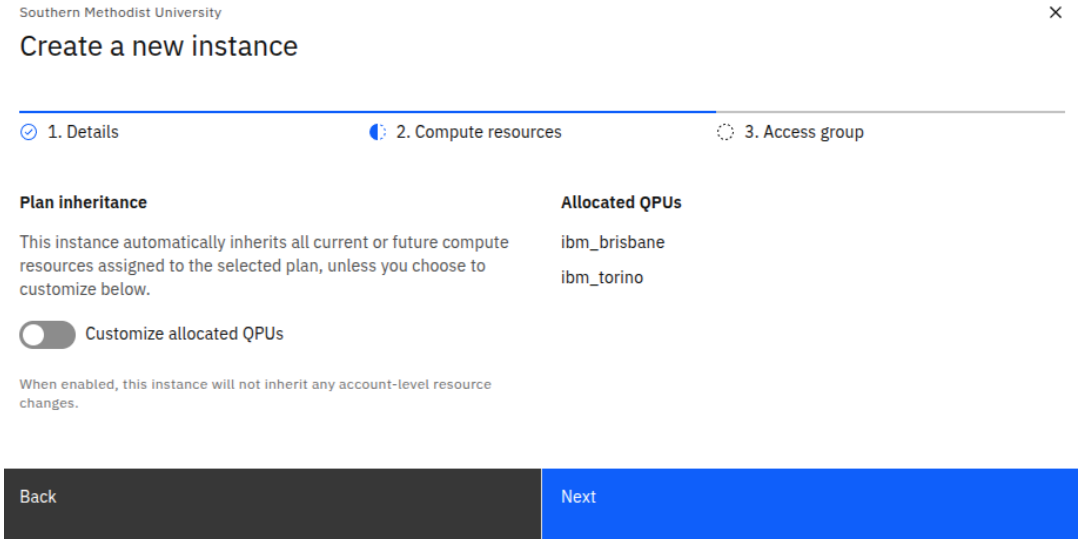
  - You will be asked to enter the name of your API key, enter anything you want.
  - After created, REMEMBER to copy it or download the .json file of your API key, you will not be able to display the API key after closing this window. And the API key will be required when you run your code on the actual IBM quantum computer.

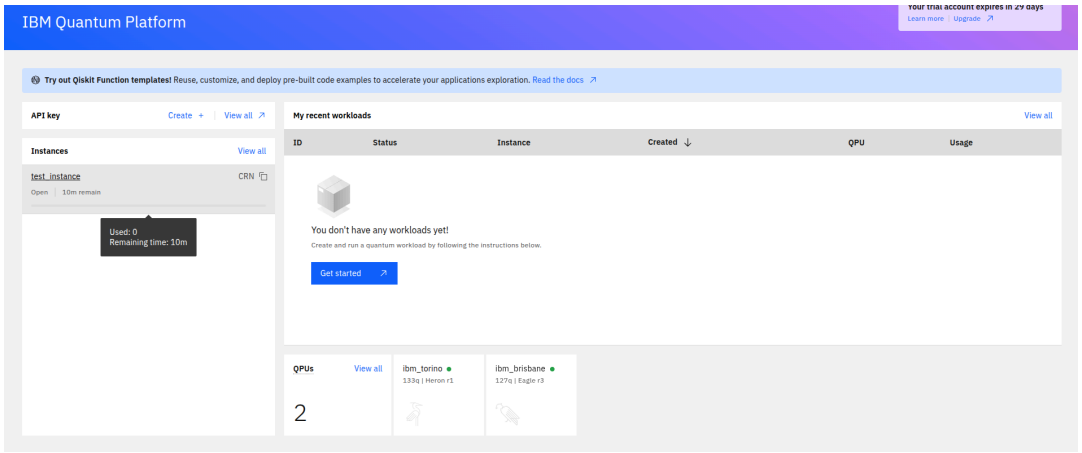
- 

  - Although you will be able to view the API keys you created, the API-Key-ID is **NOT** the actual API key, it's **JUST** an ID!!!
- (3) The last step of this section is to create an instance for managing your quantum cloud jobs.
  - Click "Create an instance".
  - 

  - You will need to enter an instance name and select "Open" pricing plan.

- 

- Click "Next" you will see there are two QPUs available on this plan.
- 

- After creating, you can see there is an instance and your recent workload panel. Now you are good to go!!! Feel free to explore all features in IBM Quantum Cloud!!!
- 

3.0 Installing and Configuring Anaconda with IBM Qiskit

Qiskit is the name of the IBM package that we will use in this homework. It is highly recommended that you create a new environment even if you already have Anaconda installed. Furthermore, as stated above, it is **VERY IMPORTANT** that you install a later version of Python (version 3.10 works well) as Qiskit might not compatible with the early version of Python at the time of this writing. In our testing on Python 3.9, it will raise a deprecating warning while running the actual IBM runtime, a warning will be OK at the moment, but we need to prepare for the situation if Qiskit updates and removes the support for early version.

- (1) Open a terminal on your machine. Enter the command conda info to determine if you have already installed Anaconda. If you do not have Anaconda installed, you can download it from here:
  - https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html
  - Notes: For Windows OS, the anaconda will have a unique terminal called "Anaconda Prompt", therefore the conda command might not work on the machine. As for MacOS and Linux, the conda is embedded automatically into your terminal.
- (2) Using and creating conda environments:
  - a) Open a terminal on your machine.
  - b) Enter the command `conda env list` to see the currently installed environments.
  - c) Choose a name for your new conda environment to use for this homework. Make sure it is not a name that is already being used from the previous step. I used the name qiskitLab. Create the new environment by entering the command:
    - ```
      conda create --name qiskitLab python=3.10
      ```
    - This is the environment in which you will use to run your IBM quantum computer programs.
  - d) After creating the new environment, you need to activate it. To do so, enter:
    - ```
      conda activate qiskitLab
      ```
  - e) When you wish to leave this environment, you can enter:
    - ```
      conda deactivate qiskitLab
      ```
- (3) Downloading Jupyter
  - a) You need to download the *jupyter lab* package if you do not already have it installed (note that *jupyter notebook* can also be used if you prefer). *Jupyter lab* is similar to *jupyter notebook* with runnable cells, but it has additional features. Jupyter can be installed in the base environment since it can access kernels in other environments. Enter the following command:
    - ```
      conda install jupyterlab
      ```
  - b) Install the *jupyter lab* kernel by entering the command (It's optional since usually it's automatically downloaded when installing jupyterlab):

        - `conda install ipykernel`
    - c) Optionally, this command allows you to select the environment from *jupyter lab*.
        - `python -m ipykernel install --user --name qc --display-name "Python (qc)"`
    - We can see the available kernels for jupyter lab with command:
        - `jupyter kernelspec list`
- (4) Downloading additional (required) packages. Although conda install can be used, the pip utility seems to be more reliable and faster. For this reason, I will use pip in the following instructions.
    - a) First, you need to install the required qiskit, qiskit-aer, qiskit-ibm-runtime package.
    - `pip install qiskit, qiskit-aer, qiskit-ibm-runtime`
        - qiskit-aer is the package for Aer simulator, and qiskit-ibm-runtime is for running your code on actual IBM quantum computer.
        - During installing qiskit, it will automatically install some data analysis dependencies such as numpy. Therefore, it will not be necessary to install numpy otherwise it might would cause dependency problem.
        - To support certain graphics operations, it is necessary to install an additional qiskit package. Note that users of later versions of Mac OSX will be using zsh as the default linux shell in the terminal. The command to install the qiskit visualization package for zsh uses single quotes whereas that for Windows users or other linux shells does not. For this reason, I give both versions of the installation command:
        - `pip install qiskit[visualization]  # (for Windows or older versions of Mac OSX that do not use zsh)`
        - `pip install 'qiskit[visualization]'  # (for Mac OSX users running zsh)`
    - b) Next, you need to install a plotting package to display qubit histograms for repeated runs/measurements. Enter:
    - `pip install matplotlib`
    - d) One important dependency package is also needed to be installed to display coupling map for IBM quantum computer.
        - `pip install graphviz`
- (5) Verify that all required packages are installed by entering:
    - ```
      conda list # or pip list
      conda list [package_name] # for verifying specific package
      ```
    - If any packages are missing repeat the appropriate installation steps above.
      ```

4.0 Performing the Homework Exercise

Using the jupyter lab environment. In the terminal window, make sure that you are in the **correct** conda environment (qiskitLab in the example here). If you are not in the correct environment, enter:

```
conda activate qiskitLab
```

Or the environment name you choose, if different.

- To activate the jupyter notebook within your default browser, enter:
    - ```
      jupyter lab
      ```
    - This command should open a web application in your default web browser (note, I used Ubuntu Linux OS, but Firefox is usually the preferred browser). Within the web application, browse to the jupyter notebook file, QuantumLab5.ipynb, and click on it to open it. You can navigate to the notebook by using the leftmost side panel that functions as the file directory tool. Alternatively, if you set up the kernel tab using the "python -m ipykernel install ..." command discussed previously, you can simply select that widget to restart and shutdown the kernel.
- Start at the top of the notebook and follow the directions written in text, and by running the cells with Python/QASM source code, inserting your own code where requested, and answering the questions by typing your answer in the markdown cells. You can select code cells and run them by clicking on the "run/play" button in the top ribbon menu (the small triangle), or, you can select the cell and use **<SHIFT> + <ENTER>** to run the cell.
- The first few cells are fully populated with source code, so you will just execute those as is. Later cells require you to enter your own code to run. Note that the notebook also has homework questions embedded within it. You can answer those questions by adding text to the notebook in a markdown cell that follows each question. Be sure to save your notebook before exiting jupyter lab and please export your notebook to .html before turning it in by email.
- To exit jupyter lab, enter **<CTRL> + C** in the terminal window. For Mac OSX users, note that it is actually the **<CTRL>** key and NOT the **<CMD>** key that is used.

5.0 Submitting the Completed homework

Follow all directions in the notebook (read carefully). When your homework is complete and your notebook has been properly updated, save it to your disk using the "file" drop down menu within the web browser and export it to .html. Make sure that all output cells are properly populated with the output of each code cell. I strongly suggest that you first email the completed notebook to yourself, then open it and examine the content to make sure that you are turning in the updated and completed notebook. Notebooks turned in without the updated source code, output cells, and question answers will be counted as late submissions. When you are sure that you have a properly updated notebook file, please email it to the instructor and the TAs. Send the email to:

mitch@smu.edu
      ```

and

yayum@smu.edu

Have fun programming an actual quantum computer!