

Table of Contents

Installing Qiskit

Requirements

Qiskit supports Python 3.6 or later. However, both Python and Qiskit are evolving ecosystems, and sometimes when new releases occur in one or the other, there can be problems with compatibility.

If you're having trouble installing or using Qiskit after updating Python, check the [Qiskit Package metadata](#) for **Programming Language** to see which specific versions of Python are supported.

We recommend installing [Anaconda](#), a cross-platform Python distribution for scientific computing. Jupyter, included in Anaconda, is recommended for interacting with Qiskit.

Qiskit is tested and supported on the following 64-bit systems:

- Ubuntu 16.04 or later
- macOS 10.12.6 or later
- Windows 7 or later

Install

We recommend using Python virtual environments to cleanly separate Qiskit from other applications and improve your experience.

The simplest way to use environments is by using the `conda` command, included with Anaconda. A Conda environment allows you to specify a specific version of Python and set of libraries. Open a terminal window in the directory where you want to work.

It is preferred that you use Anaconda prompt installed with the Anaconda. All you have to do is create a virtual environment inside Anaconda and activate the environment. These commands can be run in Anaconda prompt irrespective of Windows or Linux machine.

Create a minimal environment with only Python installed in it.

```
conda create -n name_of_my_env python=3
```

Activate your new environment.

```
conda activate name_of_my_env
```

• NOTE

If you are using conda versions prior to 4.6, use: For Linux or macOS: `source activate name_of_my_env` For Windows: `activate name_of_my_env`

Next, install the Qiskit package, which includes Terra, Aer, Ignis, and Aqua.

```
pip install qiskit
```

• NOTE

Starting with Qiskit 0.13.0 pip 19 or newer is needed to install qiskit-aer from the precompiled binary on Linux. If you do not have pip 19 installed you can run `pip install -U pip` to upgrade it. Without pip 19 or newer this command will attempt to install qiskit-aer from sdist (source distribution) which will try to compile aer locally under the covers.

If the packages were installed correctly, you can run `conda list` to see the active packages in your virtual environment.

If you intend to use visualization functionality or Jupyter notebooks it is recommended to install qiskit with the visualization extra requirements

```
pip install qiskit[visualization]
```

It is worth pointing out that if you're a zsh user (which is the default shell on newer versions of macOS), you'll need to put `qiskit[visualization]` in quotes:

```
pip install 'qiskit[visualization]'
```

• NOTE

```
import qiskit
```

• NOTE

If you want to contribute to the Qiskit community by developing and contributing code with the most recently updated Qiskit code, see [Build Qiskit packages from source](#).

Access IBM Quantum Systems

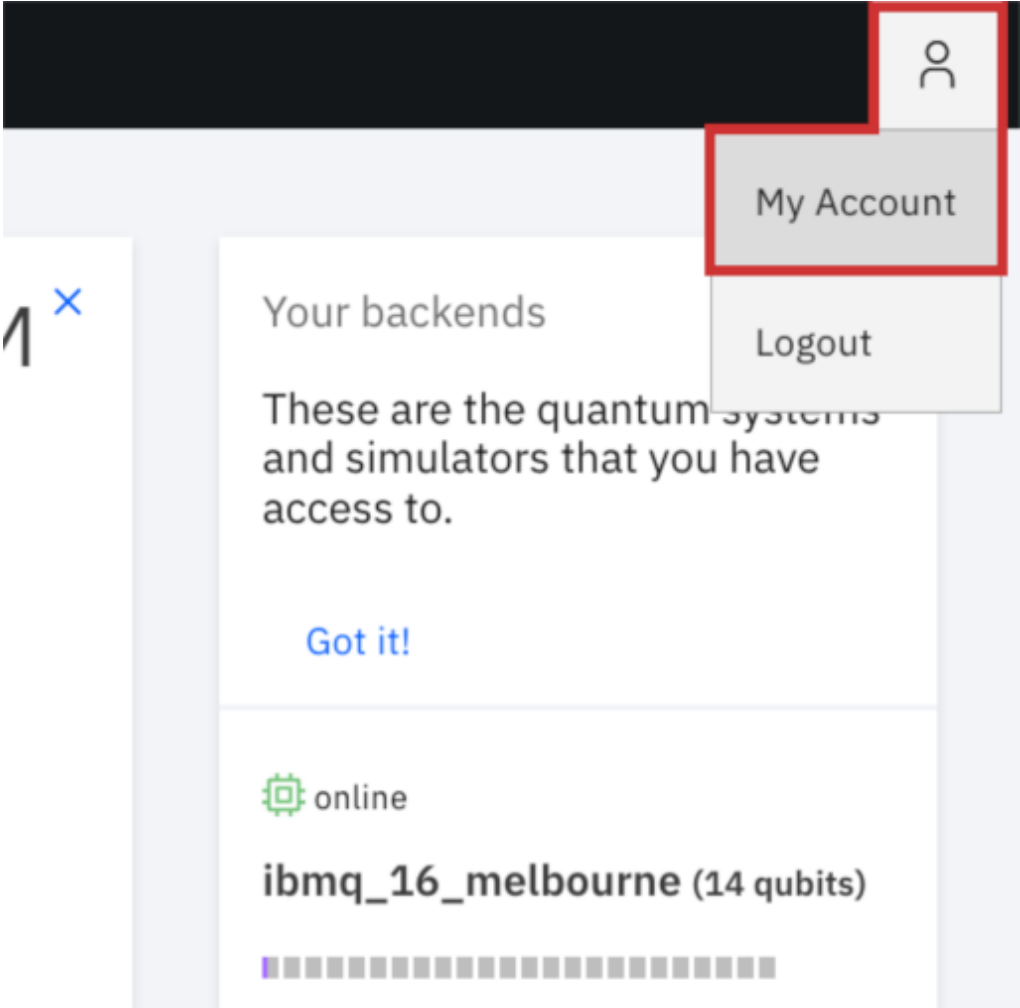
IBM Quantum offers several real quantum computers and high-performance classical computing simulators through its IBM Quantum Experience with Qiskit. Follow these steps to set up your Qiskit environment to send jobs to IBM Quantum systems.

• NOTE

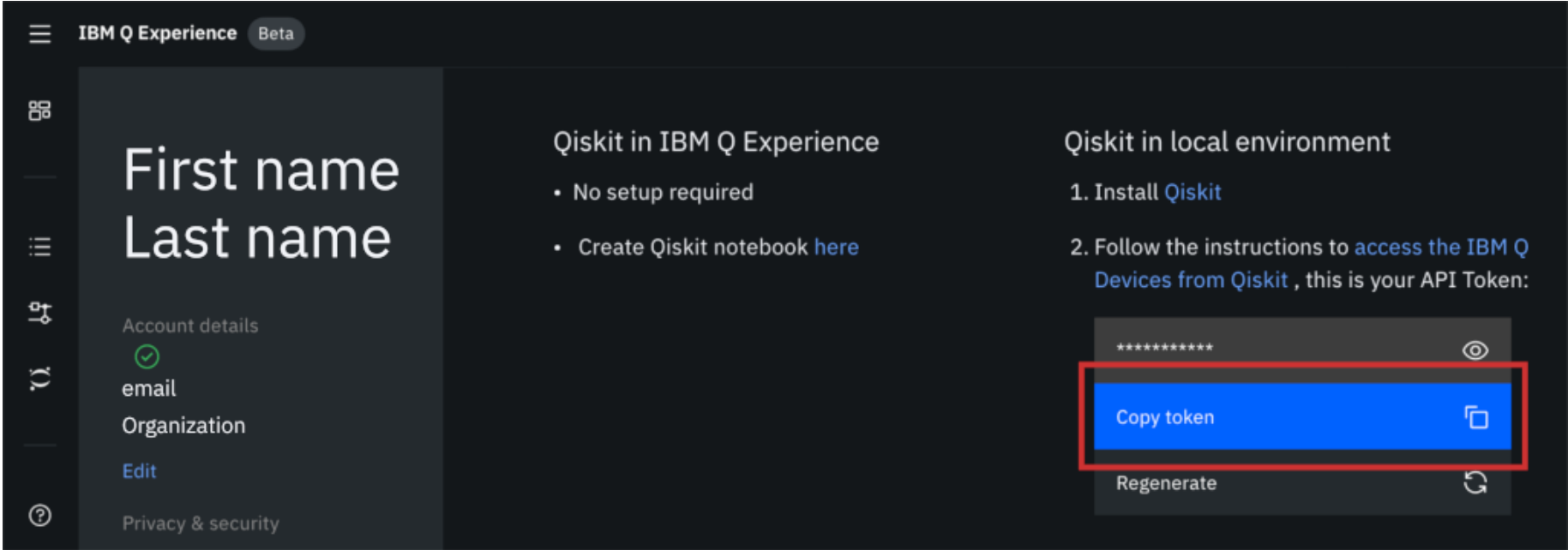
With the release of Qiskit 0.11, if you had previously saved your IBM Quantum Experience credentials locally, you might need to update them so that you can use the new IBM Quantum Experience V2. See [Updating your IBM Quantum Experience Credentials](#).

To configure your account, you will create a local configuration file which includes an API key

1. [Create a free IBM Quantum Experience account](#).
2. Navigate to **My Account** to view your account settings.



3. Click on **Copy token** to copy the token to your clipboard. Temporarily paste this API token into your favorite text editor so you can use it later to create an account configuration file.



4. Run the following commands to store your API token locally for later use in a configuration file called `qiskitrc`. Replace `MY_API_TOKEN` with the API token value that you stored in your text editor.

```
from qiskit import IBMQ
IBMQ.save_account('MY_API_TOKEN')
```

For more details, such as how to manage multiple IBM Quantum account credentials, refer to this tutorial titled [The IBM Quantum Account](#).

Checking Which Version is Installed

Since the Qiskit package includes a constellation of different elements, simply printing the version by running `qiskit.__version__` can be misleading as it returns only the version for the `qiskit-terra` package. This is because the `qiskit` namespace in Python doesn't come from the Qiskit package, but instead is part of the `qiskit-terra` package.

```
import qiskit
qiskit.__version__

'0.16.1'
```

To see the versions of all the Qiskit elements in your environment you can use the `__qiskit_version__` attribute. For example, running the following command will return a dictionary that includes the versions for each of the installed Qiskit packages.

```
qiskit.__qiskit_version__

{'qiskit-terra': '0.16.1',
 'qiskit-aer': '0.7.1',
 'qiskit-ignis': '0.5.1',
 'qiskit-ibmq-provider': '0.11.1',
 'qiskit-aqua': '0.8.1',
 'qiskit': '0.23.1'}
```

• TIP

If you're filing an issue or need to share your installed Qiskit versions for something, use the `__qiskit_version__` attribute.

[◀ Previous](#)

[Next ▶](#)