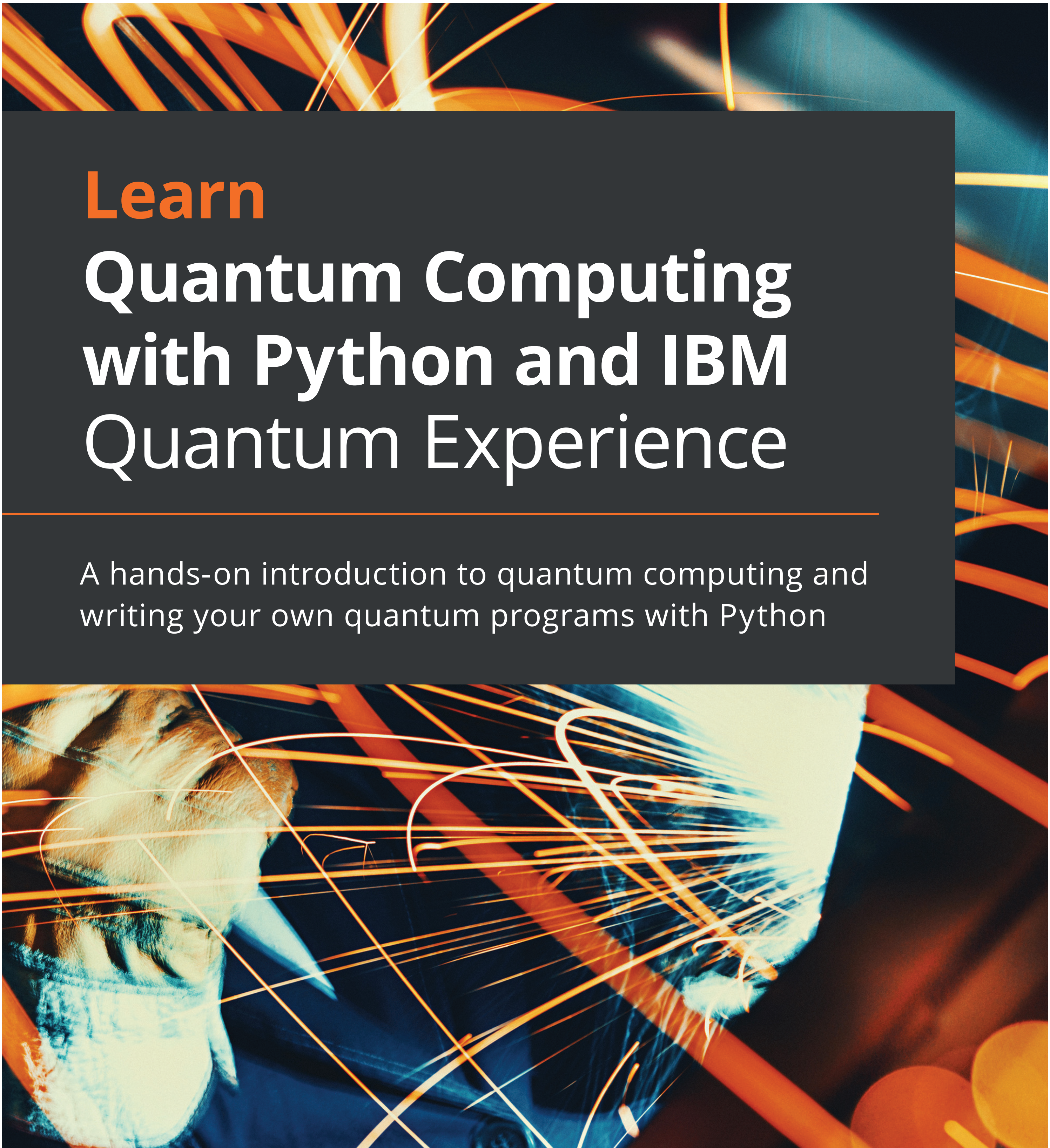

Learn Quantum Computing with Python and IBM Quantum Experience

A hands-on introduction to quantum computing and writing your own quantum programs with Python

Robert Loredo

Learn Quantum Computing with Python and IBM Quantum Experience

A hands-on introduction to quantum computing and writing your own quantum programs with Python

Robert Loredo

Packt
BIRMINGHAM—MUMBAI

Learn Quantum Computing with Python and IBM Quantum Experience

Commissioning Editor: Richa Tripathi
Acquisition Editor: Alok Dhuri
Senior Editor: Rohit Singh
Content Development Editor: Rosal Colaco
Technical Editor: Gaurav Gala
Copy Editor: Safis Editing
Project Coordinator: Deeksha Thakkar
Proofreader: Safis Editing
Indexer: Priyanka Dhadke
Production Designer: Aparna Bhagat

First published: September 2020

Production reference: 1250920

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-83898-100-6

www.packt.com

To my father, Israel, my mother, Sadie, and brother, Pierre; without their support throughout my life I would not have gotten to where I am today.

Contributors

About the author

Robert Loredo is the IBM Quantum Global Technical Ambassador lead with over 20 years' experience in software architecture and engineering. He is also a Qiskit Advocate and Master Inventor who holds over 160 patents and has presented various workshops, lectures, and articles covering quantum computing, artificial intelligence, and bioinformatics world-wide. As an adjunct professor, he has taught cloud computing and software engineering at the Florida International University School of Computer Science. He holds both a bachelor's and a master's degree in Computer and Electrical Engineering from the University of Miami and is currently pursuing his PhD in Computer Science, specializing in Machine Learning and Neuroscience, at Florida International University.

I want to thank Arvind Krishna, Dario Gil, Jay Gambetta, Bob Sutor, Scott Crowder, Anthony Annunziata, Denise Ruffner, Mehdi Bozzo-Rey, Tammy Cornell, Abe Asfaw, Andrew Wack, Bill Minor, Brian Eccles, Brian Ingmanson, Bryce Fuller, Chris Nay, Chris Schnabel, Dan Maynard, Enrique Vargas, Frederik Flöther, Gavin Jones, Hanhee Paik, Hassi Norlen, Heather Cortes, Heather Higgins, Heike Riel, Ingolf Wittman, Ismael Faro, James Weaver, James Wootton, Jeanette (Jamie) Garcia, Jerry Chow, Jody Burks, Jules Murphy, Katie Pizzolato, Ken Wood, Liz Durst, Leron Gil, Luuk Ament, M. Lewis Temares, Mark Ritter, Mei-Ling Shyu, Melissa Turesky, Michele Grossi, Miguel Paredes Quiñones, Miroslav Kubat, Mohammed Abdel-Mottaleb, Muir Kumph, Nick Bronn, Noam Zakay, Paco Martin, Paul Kassebaum, Paul Nation, Pete Martinez, Rafael Nepomechie, Rafael Sotelo, Reuven Lask, Reza Sanatinia, Robert Eades, Rudy Wojtecki, Sarah Sheldon, Stefan Woerner, Suzie Kirschner, Talia Gershon, Walter Riess, and Zaira Nazario for their continued support.

About the reviewer

Michele Grossi is a Technical Architect and Quantum Technical Ambassador who has worked for IBM Italy since 2015 with a focus on problem solving, innovation, and technology.

Born in 1989, he received his MSc degree in Physics from the University of Pavia, where he received a scholarship from the **Istituto Nazionale di Fisica Nucleare** (**INFN**). He's an industrial PhD student at the University of Pavia specializing in high energy physics and quantum computing. He is the co-author of several scientific publications, a conference speaker, and a lecturer at various universities. In his current role, Michele collaborates with customers to evaluate the latest technology utilization, ranging from machine learning to quantum computing, and with research institutions, among them IBM Research and CERN.

In 2019, Forbes selected Michele as the top "30 under 30s" brightest young Italian leaders.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

3
Creating Quantum Circuits using Quantum Lab Notebooks

Section 2: Basics of Quantum Computing

4
Understanding Basic Quantum Computing Principles

5
Understanding the Quantum Bit (Qubit)

6

Understanding Quantum Logic Gates

Section 3: Algorithms, Noise, and Other Strange Things in Quantum World

7

Introducing Qiskit and its Elements

8
Programming with Qiskit Terra

9
Monitoring and Optimizing Quantum Circuits

10
Executing Circuits Using Qiskit Aer

11
Mitigating Quantum Errors Using Ignis

12
Learning about Qiskit Aqua

13
Understanding Quantum Algorithms

14
Applying Quantum Algorithms

Appendix A
Resources

Assessments

Other Books You May Enjoy

Index

Preface

IBM Quantum Experience is a platform that enables developers to learn the basics of quantum computing by allowing them to run experiments on a quantum computing simulator and a real device. This book will explain the basic principles of quantum computing, along with one principle of quantum mechanics, entanglement, and the implementation of quantum algorithms and experiments on IBM's quantum processors.

This book provides you with a step-by-step introduction to quantum computing using the **IBM Quantum Experience** platform. You will learn how to build quantum programs on your own, discover early use cases in your business, and help to get your company equipped with quantum computing skills.

You will start working with simple programs that illustrate quantum computing principles and slowly work your way up to more complex programs and algorithms that leverage advanced quantum computing algorithms. As you build on your knowledge, you'll understand the functionality of the IBM Quantum Experience and the various resources it offers.

We'll explore quantum computing principles such as superposition, entanglement, and interference, then we'll become familiar with the contents and layout of the IBM Quantum Experience dashboard.

Then, we'll understand quantum gates and how they operate on qubits and discover the **Quantum Information Science Kit** (**Qiskit**) and its elements such as Terra and Aer.

We'll then get to grips with quantum algorithms such as Deutsch-Jozsa, Simon, Grover, and Shor's algorithms, and then visualize how to create a quantum circuit and run the algorithms on any of the available quantum computers hosted on the IBM Quantum Experience.

Furthermore, you'll learn the differences between the various quantum computers and the different types of simulators available. Later, you'll explore the basics of quantum hardware, pulse scheduling, quantum volume, and how to analyze and optimize your quantum circuits, all while using the resources available on the IBM Quantum Experience.

By the end of this book, you'll have learned how to build quantum programs on your own and will have gained practical quantum computing skills that you can apply to your research or industry.

Who this book is for

This book is for Python developers who are interested in learning about quantum computing and expanding their abilities to solve classically intractable problems with the help of the IBM Quantum Experience and Qiskit. Some background in computer science, physics, and some linear algebra is required.

What this book covers

Chapter 1, *Exploring the IBM Quantum Experience*, will be your guide to the IBM Q Experience dashboard. This chapter will describe the layout and what each section in the dashboard means. The dashboard might alter over time, but the basic information should still be available to you.

Chapter 2, *Circuit Composer – Creating a Quantum Circuit*, will help you learn about Circuit Composer. This chapter will outline the user interface that will assist you in learning about quantum circuits, the qubits, and their gates that are used to perform operations on each qubit.

Chapter 3, *Creating Quantum Circuits Using Quantum Lab Notebooks*, will help you learn how to create circuits using the Notebook with the latest version of Qiskit already installed on the IBM Quantum Experience. You will learn how to save, import, and leverage existing circuits without having to install anything on your local machine.

Chapter 4, *Understanding Basic Quantum Computing Principles*, will help you learn about the basic quantum computing principles used by the IBM Quantum systems, particularly, superposition, entanglement, and interference. These three properties, often used together, serve as the base differentiators that separate quantum systems from classical systems.

Chapter 5, *Understanding the Quantum Bit (Qubit)*, will help you learn about the basic fundamental component of a quantum system, the quantum bit or qubit, as it is often called. After reading this chapter, you will understand the basis states of a qubit, how they are measured, and how they can be visualized both mathematically and graphically.

Chapter 6, *Understanding Quantum Logic Gates*, will help you learn how to perform operations on a qubit. These operations are often referred to as quantum gates. This chapter will enable you, via the IBM Quantum Experience, to get to grips with the operations that each of these quantum gates performs on a qubit and the results of each of those operations. Examples of the quantum principles such as reversibility, which is a core principle for all quantum gates, will be included.

Chapter 7, *Introducing Qiskit and its Elements*, will help you learn about Qiskit and all of its libraries that can help you develop and implement various quantum computing solutions. Qiskit is composed of four elements, each of which has a specific functionality and role that can be leveraged based on the areas you wish to experiment in. The elements are Terra (Earth), Aer (Air), Ignis (Fire), and Aqua (Water). This chapter will also discuss how to contribute to each of the elements and how to install it locally on your machine.

Chapter 8, *Programming with Qiskit Terra*, will help you learn about the basic foundational element, Terra. Terra is the base library upon which all the other elements of Qiskit are built. Terra allows a developer to code the base of an algorithm to the specific operator on a qubit. This is analogous to assembly language with just a slightly easier set of library functions. It will also include a section on the Pulse library, which allows you to create pulse schedules to manipulate the quantum qubits via the hardware.

Chapter 9, *Monitoring and Optimizing Quantum Circuits*, will help you learn how to monitor the job requests sent to either the simulator or the quantum computers on the IBM Quantum Experience. Optimization features will also be covered here to allow you to leverage many of the existing optimization features included in the Qiskit libraries or to create your own custom optimizers.

Chapter 10, *Executing Circuits Using Qiskit Aer*, will help you learn about Qiskit Aer, a high-performance framework that you will use to simulate your circuits on various optimized simulator backends. You will learn what the differences are between the four various simulators of Qasm, State vector, Unitary, and Pulse, and what functionality each one exhibits. Aer also contains tools you can use to construct noise models, should you need to perform some research to reproduce errors due to noise.

Chapter 11, *Mitigating Quantum Errors Using Ignis*, will help you learn about the various errors that currently affect experiments on read devices, such as relaxation and decoherence, so you can design quantum error correction codes. You will also learn about readout error mitigation, which is a way to mitigate the readout errors returned from a quantum computer.

Chapter 12, *Learning about Qiskit Aqua*, will, in essence, pull everything together so that end users such as researchers and developers from the various domains of chemistry, machine learning, finance, optimization, and more can run their computations on a quantum computer system without having to know all the inner workings. Aqua is the tool connected to quantum algorithms that has been created to do just that. You will learn how to extend your classical application to include running a quantum algorithm.

Chapter 13, *Understanding Quantum Algorithms*, will dig into some basic algorithms using the IBM Quantum Experience Composer. This chapter will start with some simple algorithms that illustrate the advantages of superposition and entanglement, such as Bell's state theorem, and extends into some more common algorithms to solve some problems that illustrate uses of superposition and entanglement such as Deutsch-Josza and a few others, each of which provides some variance to the different algorithm types.

Chapter 14, *Applying Quantum Algorithms*, describes the various quantum computing properties and algorithms used to create some of the more well-known algorithms such as Quantum Amplitude Estimation, Variational Quantum Eigensolvers, and Shor's algorithm.

Appendix A, *Resources*, will help you get familiar with all the available resources in the IBM Quantum Experience and Qiskit community. These resources that have been contributed either by the Qiskit open source community, or the IBM Quantum research teams themselves. The information is laid out so anyone with basic to expert-level knowledge can jump in and start learning. There is a full quantum course, textbook, and Slack community that you can connect to in order to extend your learning and collaborate with others.

Assessments contains the answers to the questions asked in the chapters.

To get the most out of this book

You will need to have internet access to connect to the IBM Quantum Experience. Since the IBM Quantum Experience is hosted on the IBM Cloud, you will not need anything more other than a supported browser and to register with the IBM Quantum Experience. Everything else is taken care of on the IBM Quantum Experience.

Software/hardware covered in the book	OS requirements
Latest browser (Firefox, Chrome, Safari)	Windows, Mac OS X, and Linux (any)

If you are using the digital version of this book, we advise you to type the code yourself or access the code via the GitHub repository (link available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from your account at `www.packt.com`. If you purchased this book elsewhere, you can visit `www.packtpub.com/support` and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at `www.packt.com`.
2. Select the **Support** tab.
3. Click on **Code Downloads**.
4. Enter the name of the book in the **Search** box and follow the onscreen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR/7-Zip for Windows
- Zipeg/iZip/UnRarX for Mac
- 7-Zip/PeaZip for Linux

The code bundle for the book is also hosted on GitHub at `https://github.com/PacktPublishing/Learn-Quantum-Computing-with-Python-and-IBM-Quantum-Experience`. In case there's an update to the code, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Code in Action

Code in Action videos for this book can be viewed at `https://bit.ly/35o5M80`.

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `https://static.packt-cdn.com/downloads/9781838981006_ColorImages.pdf`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "This will initialize our `t1`, `a`, and `b` parameters, which we will use to generate `T1Fitter`."

A block of code is set as follows:

```
# Initialize the parameters for the T1Fitter, A, T1, and B
param_t1 = t1*1.2
param_a = 1.0
param_b = 0.0
```

Any command-line input or output is written as follows:

```
[[1. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 1. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "As shown in the following screenshot, **ibmq_qasm_simulator** can run wider circuits than most local machines and has a larger variety of basis gates."

> **Tips or important notes**
> Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at `customercare@packtpub.com`.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit `packt.com`.

Section 1: Tour of the IBM Quantum Experience (QX)

In this section, we will tour all the features and resources available to you on the IBM Quantum Experience. These will include some educational materials for all levels, information on the many simulators and real devices available to you, and tools that you can use to perform experiments from the many tutorials as you learn, or to simply create experiments on your own.

This section comprises the following chapters:

1
Exploring the IBM Quantum Experience

Quantum computing has been growing in popularity over the past few years, most recently since IBM released the **IBM Quantum Experience** (**IQX**) back in May 2016. This release was the first of its kind, hosted on the cloud and providing the world with the opportunity to experiment with a quantum computer for free. The IQX includes a user interface that allows anyone to run experiments on both a simulator and on a real quantum computer.

The goal of this chapter is to first introduce you to the IBM Quantum Experience site, specifically the **dashboard**, which contains everything you need in order to run experiments. It also allows you to experiment with existing experiments contributed by other developers from around the world, the benefits of which can help you to understand how others are experimenting, and you can perhaps collaborate with them if the experiments correlate with your own ideas.

This chapter will help you understand what actions and information are available in each view. This includes creating an experiment, running experiments on a simulator or real quantum device, information about your profile, available backends, or pending results to experiments. So, let's get started!

The following topics will be covered in this chapter:

- Navigating the IBM Quantum Experience
- Getting started with IBM Quantum Experience

Technical requirements

Throughout this book, it is expected that you will have some experience in developing with Python and, although it isn't necessary, some basic knowledge of classical and quantum mechanics would help.

Most of the information will be provided with each chapter, so if you do not have knowledge of classical or quantum mechanics, we will cover what you need to know here.

For those of you that do have knowledge, the information here will serve as a refresher. The Python editor used throughout this book is **Jupyter Notebook**. You can, of course, use any Python editor of your choice. This may include **Watson Studio**, **PyCharm**, **Spyder**, **Visual Studio Code**, and so on. Here is the link for the CiA videos: `https://bit.ly/35o5M80`

Here is the source code used throughout this book: `https://github.com/PacktPublishing/Learn-Quantum-Computing-with-Python-and-IBM-Quantum-Experience`.

Navigating the IBM Quantum Experience

As mentioned earlier, the dashboard is your high-level view of what you will normally see once you log in to IQX. It aggregates multiple views that you can see, and this helps you to get an idea as to what machines you have access to and what experiments you have pending, running, or completed.

In this section, we will go through the steps to get registered on IQX. Let's do that in the next section.

Registering to the IBM Quantum Experience

In this section, we will get registered and explain what happens in the background once you sign up to IQX for the first time. This will help you understand what features and configurations are prepared and available to you upon registration.

To register to the IBM Quantum Experience, follow these steps:

1. The first step is to head over to the IBM Quantum Experience site at the following link: `https://quantum-computing.ibm.com/`
2. Sign-in to your account from the login screen, as shown in *Figure 1.1*. Your individual situation will determine how to proceed from there.

   If you already have an account or are already signed in, you can skip this section and move on to the next one.

   If you have not registered, then you can select the login method of your choice from the sign-in screen. As you can see, you can register using various methods, such as with your **IBM ID**, **Google**, **GitHub**, **Twitter**, **LinkedIn**, or by email.

   If you do not have any of the account types listed, then you can simply register for an **IBMid** account and use that to sign in:

Figure 1.1 – The IBM Quantum Experience sign-in page

3. Once you select the login method of your choice, you will see the login screen for that method. Simply fill out the information, if it's not already there, and select login.

4. Once signed in, you will land on the **Home** page. This is the first page you will see each time you log in to the IBM Quantum Experience site:

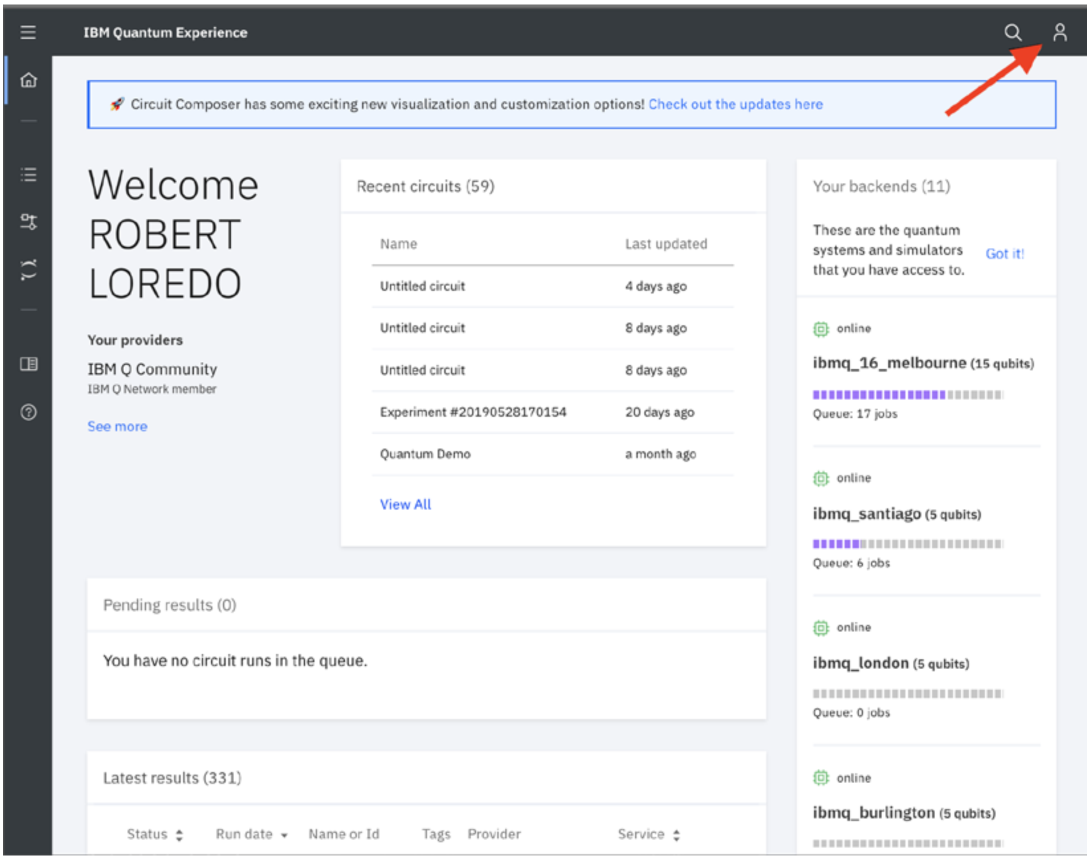

Figure 1.2 – The IBM Quantum Experience home page

Now that you have registered to the IBM Quantum Experience, let's take a quick tour and delve into some features that make up the IQX home page. Let's start by reviewing the home page, specifically the **Personal** profile tab. You can access your personal profile via your avatar, located at the top right of the page (as pointed out in *Figure 1.2*).

Understanding the Personal profile tab

This section explains the profile of the logged-in user. This is helpful if you have multiple accounts and you wish to keep track of them. The provider limits the number of jobs that can be executed or queued on a given device at any one time to a maximum, as specified in the documentation. There are many ways to access all the various quantum devices; those listed in the open group will see all freely available quantum devices, as illustrated along the right side of *Figure 1.2*. For those who are members of the **IBM Q Network**, you will have access to the open devices, as well as premium quantum devices such as the 65 qubit quantum computer.

Now that you have completed the sign-up process and successfully logged in, we can start off by taking a tour of the IBM Quantum Experience application. This will be where most of the work within this book will take place, so it will benefit you in understanding where everything is so that you can easily make your way around it while developing your quantum programs.

Getting started with IBM Quantum Experience

This section provides a quick way to launch either **Circuit Composer** or the notebooks located in the Quantum Lab views, herein simply referred to as **Qiskit notebooks**, each of which we will cover in detail in *Chapter 2*, *Circuit Composer – Creating a Quantum Circuit*, and *Chapter 3*, *Creating Quantum Circuits Using Qiskit Notebooks*, respectively, so hang in there. But as with other views, know that you can kick-start either from the main dashboard view or from the left panel. Each button easily provides a quick launch for either of the two circuit generators.

Learning about your backends

This section lists the available backend quantum systems that are provisioned for your use (as shown in *Figure 1.3*). It not only provides a list of the available backends but also provides details for each, such as the *status* of each backend. The status includes whether the device is online or in maintenance mode, how many **qubits** (**quantum bits**) each device contains, and how many experiments are in the queue to be run on the device. It also contains a color bar graph to indicate queue wait times, as illustrated between **ibmq_16_melbourne** and **ibmq_rome** in the following screenshot. Be aware that the quantum devices listed for you may be different from those listed here:

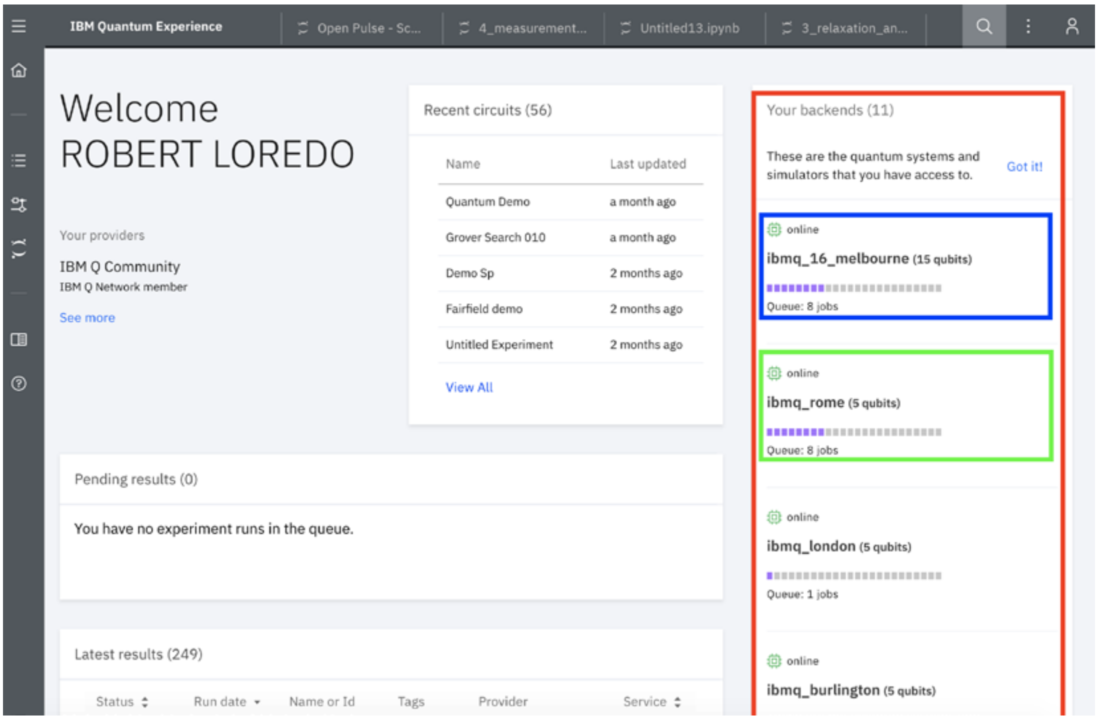

Figure 1.3 – Provisioned backend simulators and devices

From the preceding screenshot, you can see that another great feature that IQX has with respect to the backend service is the ability to see the hardware details of each real quantum device. If you hover your mouse over each device listed, you will see an expansion icon appear at the top right of the device information block. If you select a device (for example, **ibmq_16_melbourne**), you will see the device details view appear, as shown in the following screenshot:

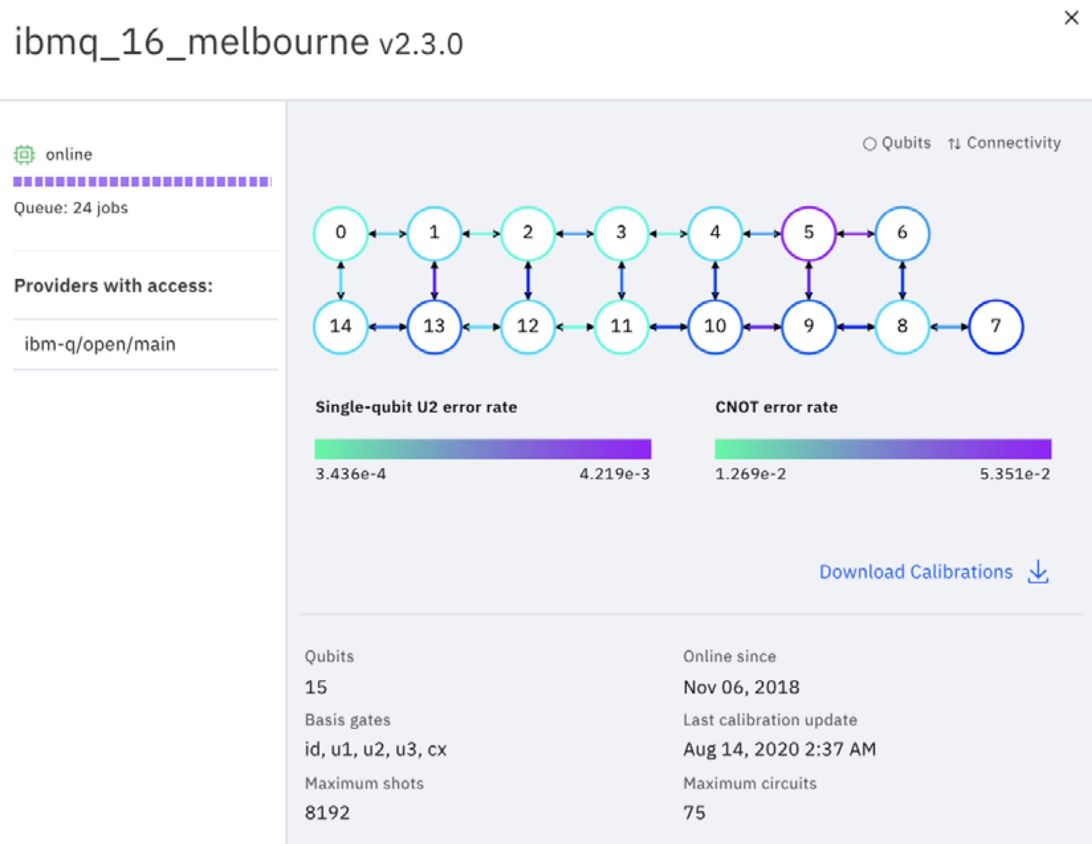

Figure 1.4 – Device details view: The status (left) and configuration and error rates (right)

From the previous screenshot, you can see that the device details view contains some very relevant information, particularly if you are working on any experiments that have intricate connectivity between qubits or analyzing error mitigation techniques. On the left of the screenshot (*Figure 1.4*), you can see the basic status information of the device. This is similar to what you see before expanding the device information. In the square on the right, we get into a little more detail with respect to the devices' configuration, connectivity, and error rates.

As described in the shaded bar area, where the error rate range is illustrated by **Single-qubit U3 error rate**, and **CNOT error rate** (single qubit and multi-qubit, respectively), qubits are identified as the circles where the number specifies the qubit number in the device. The arrows in between identify how each qubit is connected to the other qubits. The connections are specific to how the multi-qubit operations are specified.

For example, in the **15** qubit configuration in *Figure 1.4* (on the right), you can see that qubit number **4** is the source for target qubits **3** and **10** (we will get into what source and target mean later, but for now just assume that actions to the target qubit are triggered by the source qubit). You can also see that qubit **4** is the target qubit of qubit **5**. This visual representation is based on information provided by the device configuration, which you can also access programmatically using **Qiskit**.

Another piece of information you can get here is the error rates. The devices are calibrated at least once a day or so, and each time they are calibrated, they calculate the average error rates for a single gate (**u3**) and multi-gates (**CNOT**). The error rates vary per qubit, or qubits for multi-gates, and therefore, the diagram uses a color heat map to identify where the qubit sits on the error rate scale. Each qubit has a different color associated with it. This color makes it possible to visually identify where on the error rate scale that qubit falls. If you are running an experiment on a qubit that requires low error rates, then you can see from this diagram which of these qubits has the lowest error rate when last calibrated.

Below the qubit configuration, you will see a link that also allows you to download the entire configuration information in a spreadsheet. The details there are very specific to each qubit and they provide more information that isn't visible on the qubit configuration diagram.

Finally, at the bottom of the view are the specifics of the device itself, which includes the number of qubits, the date the device went online, and the basis gates available on the device.

You can now close the device configuration diagram to return to the dashboard, where we will next learn about the quantum programs and how to monitor them.

Learning about pending and latest results

The table shown in *Figure 1.5* contains the experiments that are pending completion on the backend devices. You can use this view to quickly see whether your experiments have run, and if not, where in the queue your experiment is set to run next.

Under your pending results table is the table where all your latest results are stored. These are the last few experiment results that were run on either the simulator or real devices on the backend. Each device is initially sorted by creation date but can be sorted by either backend or status, if need be.

Important Note

Details regarding job objects will be covered in *Chapter 9*, *Executing Circuits Using Qiskit Aer*.

As well as this, the job ID is listed so that you can call back the details from that job at a later time, as seen in the following screenshot:

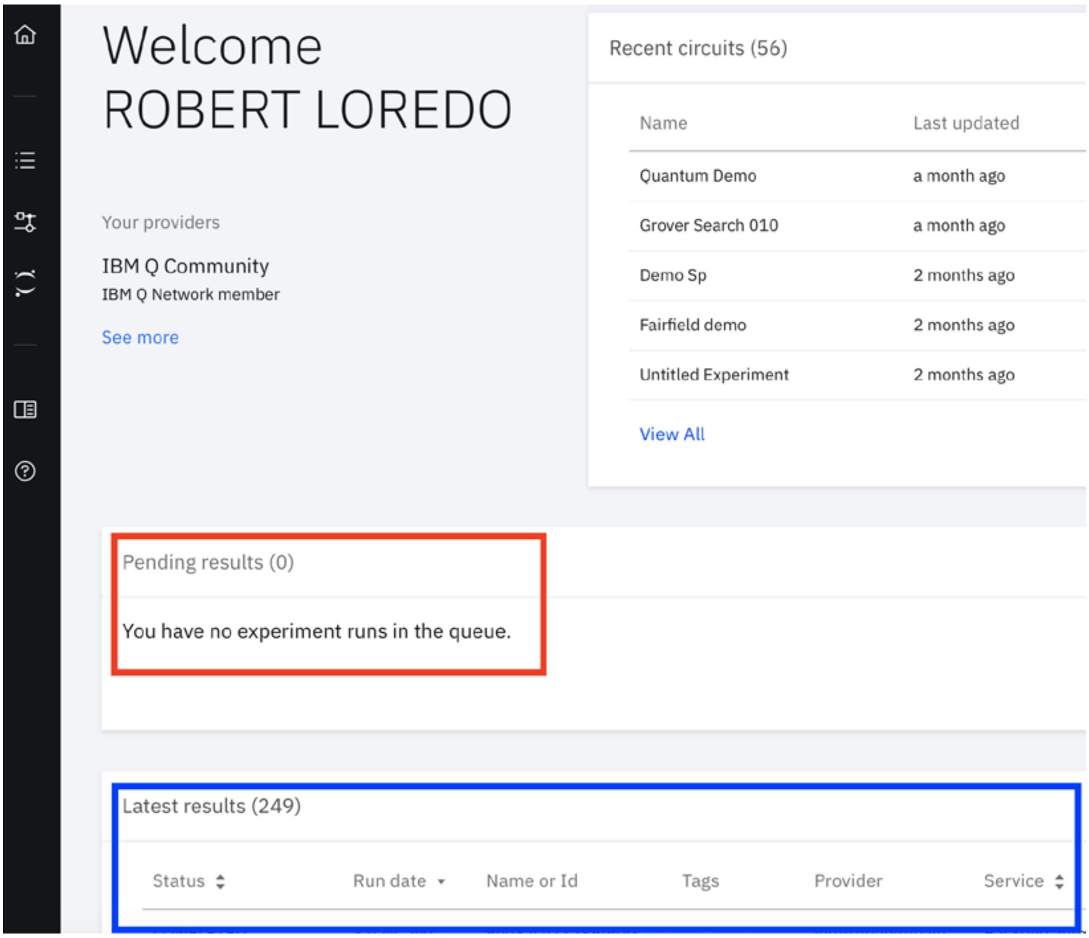

Figure 1.5 – Pending results and latest results

In this section, you have learned where to find information about your experiments, hardware details about the simulators and the real quantum devices. Next, we will explore your account profile.

Exploring My Account

In this section, you will explore your account details view, where you will find information about your account and what services are available to you. This includes services such as the ability to view the list of backend systems available to you, notification settings, and resetting your password.

To open the account view, follow these steps:

1. Click on your avatar at the top right of the dashboard (as highlighted in the following screenshot) and select **My Account**:

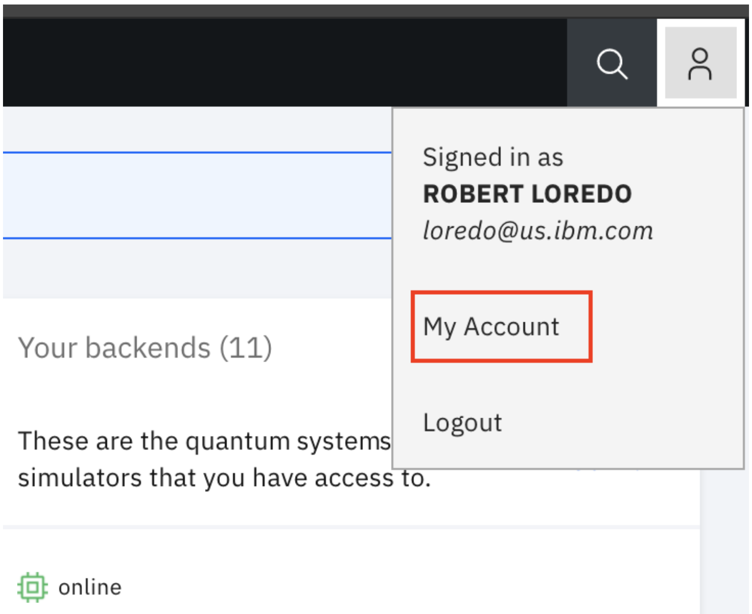

Figure 1.6 – The My Account option on the dashboard

2. Once the **My Account** view is loaded, you will see a page similar to this:

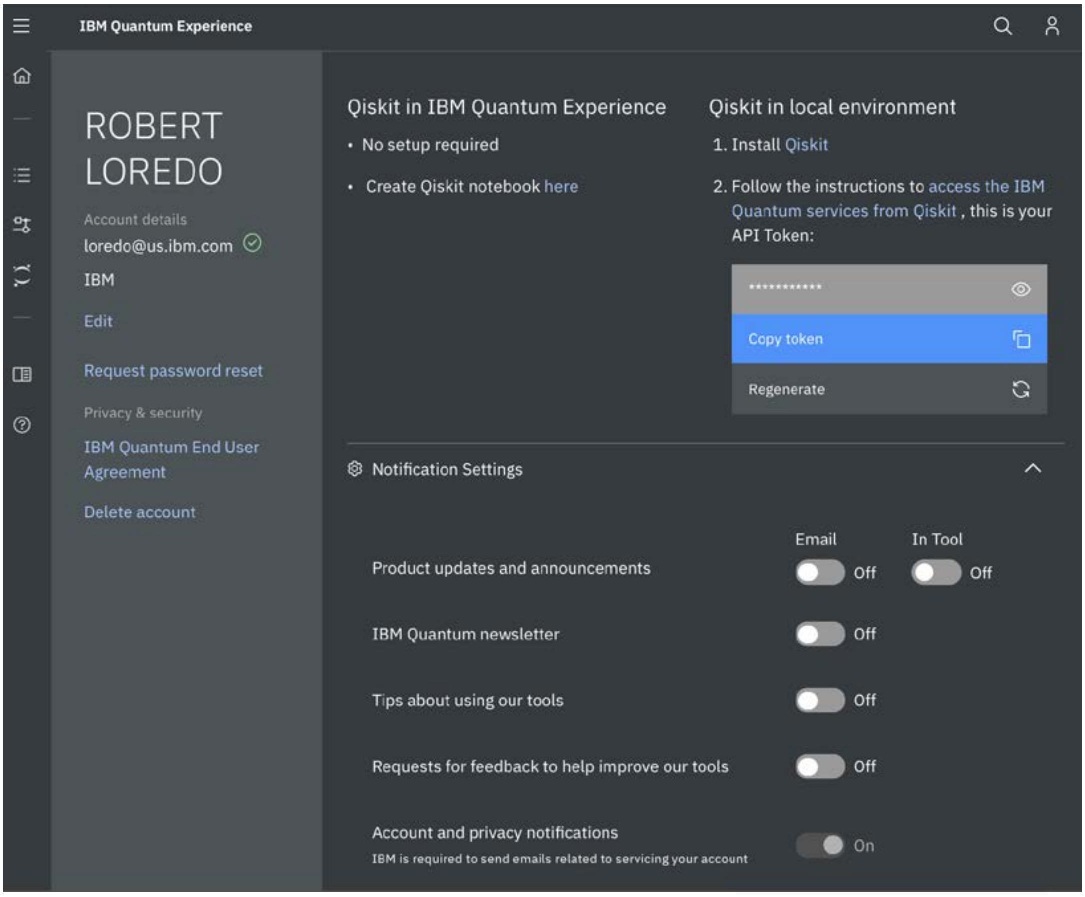

Figure 1.7 – The My Account view

From the preceding screenshot, you can see that on your account page, you will see the following information sections:

- **Account details**: This section has your account and contact information that you used to register. It also includes options such as resetting your password, privacy and security information, and the option to delete your account.
- **Qiskit in IBM Quantum Experience**: This includes a quick link to launch a Qiskit notebook to run your experiments. We will review the Qiskit notebook later in this book, but for now, just know that you can launch a Qiskit notebook from here as well.

- **Qiskit in local environment**: This section allows you to install Qiskit and run experiments from your local machine without the need to connect to IQX via the cloud. This is exceptionally helpful when you wish to run experiments but do not have access to a network. By running experiments from your local machine, this allows you to run simulators that are installed as part of the Qiskit installation. However, keep in mind that in order to run the experiments on a real quantum device, you will need network connectivity to those real devices.

  If you want to run the experiments on a real device from your local machine, then you will need to copy the token (highlighted in *Figure 1.7*) that was generated for you in the background. You should then assign it to the **Qiskit IBMQ provider** class. Details of the IBMQ provider class will be discussed in *Chapter 9*, *Executing Circuits Using Qiskit Aer*, but for now, this is where you can copy the **Application Programming Interface** (**API**) token.

  Also, note that there is an option to regenerate the API token. If you choose to regenerate the token, you will need to delete your old token and save the regenerated one in your local IBMQ provider class. The save account method of the IBMQ provider class will persist the value in your local machine, so you will only have to save it once and then load the account each time you wish to use a real quantum device for your experiment.

  Since this book is written primarily for use on the IBM Quantum Experience site, we will cover running and setting up on your local machine. Just in case you happen to not have network connectivity, you can still run simulated experiments locally.

- **Notification Settings**: This section simply allows you to set your notifications and how you prefer to receive information, such as when experiments have completed or other information or surveys that you wish to contribute.

- **Your accounts**: This last section toward the bottom of the **My Account** view is an overview of the accounts that you have and a list of the provisioned systems you have access to. These provisions are selected and assigned as part of the sign-up process. This includes information such as when you first signed up, the project that you are associated with (**main** is usually the default project), provider information, and the allocated backend systems that you have access to. These allocated backends that you can see are either real devices, such as **ibmq_16_melbourne**, or simulators, such as **ibmq_qasm_simulator**, which are running on the IQX cloud. We will discuss the details of the simulators and devices in later chapters.

Now that we are done with our tour of the IBM Quantum Experience layout, we're ready to get to work. In the following chapters, we will delve into each section and progress to writing quantum programs.

Summary

In this chapter, we reviewed the dashboard, which provides plenty of information to help you get a good lay of the land. You now know where to find information regarding your profile, details for each of the devices you have available, the status of each device, as well as the status and results of your experiments.

Knowing where to find this information will help you monitor your experiments and enable you to understand the state of your experiments by reviewing your backend services, monitoring queue times, and viewing your results queues.

You also have the skills to create an experiment using either Circuit Composer or the Qiskit notebooks. In the next chapter, we will learn about Circuit Composer in detail.

Questions

1. Which view contains your API token?
2. Which device in your list has the fewest qubits?
3. How many connections are there in the device with the fewest qubits?
4. What are the two tools called that are used to generate quantum circuits?
5. Which view would provide you with the list of basis gates for a selected device?