Setting up an IBMQ calculation

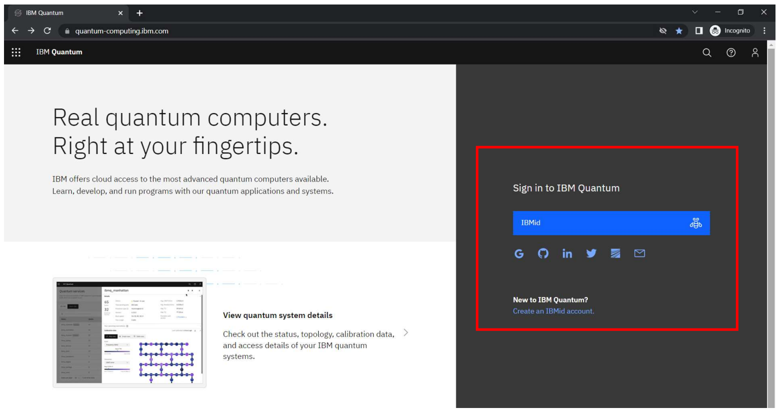
- To use or emulate the IBM quantum computing devices you'll need to provide `pytket.extensions.qiskit` with an **API token** from IBM Quantum.
- *Anyone* can make and use IBM resources, but they'll only have access to a few small devices.
- Website to create account:
  https://quantum-computing.ibm.com/
- Make sure you have both `InQuanto` and `pytket.extensions.qiskit` in your Python environment
  (Use `pip install pytket-qiskit` if needed)

1

Create IBMQ account

- Go to:

  https://quantum-computing.ibm.com/
- Either use an auxiliary login (e.g. Google) or follow instructions for IBMid creation



Real quantum computers.
Right at your fingertips.

IBM offers cloud access to the most advanced quantum computers available.
Learn, develop, and run programs with our quantum applications and systems.

View quantum system details

Check out the status, topology, calibration data, and access details of your IBM quantum systems.

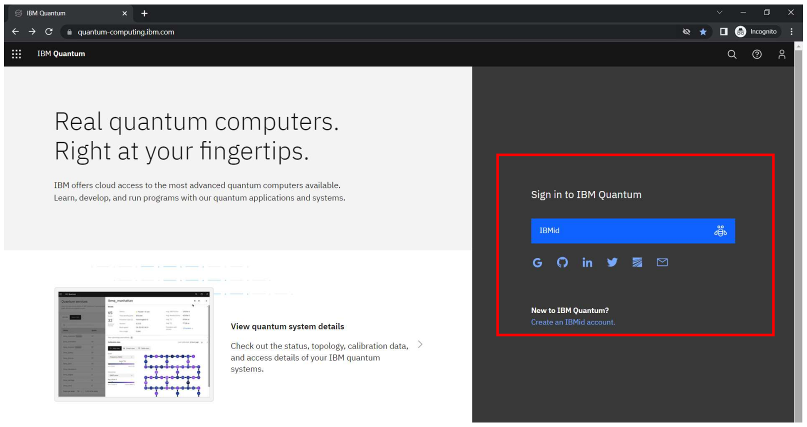
Sign in to IBM Quantum

New to IBM Quantum?
Create an IBMid account.

2

Create IBMQ account

- If using IBMid > input details > complete any 2FA > log in to IBMQ > agree to IBM EULA



IBM Quantum
quantum-computing.ibm.com

Real quantum computers.
Right at your fingertips.

IBM offers cloud access to the most advanced quantum computers available.
Learn, develop, and run programs with our quantum applications and systems.

View quantum system details

Check out the status, topology, calibration data, and access details of your IBM quantum systems.

Sign in to IBM Quantum

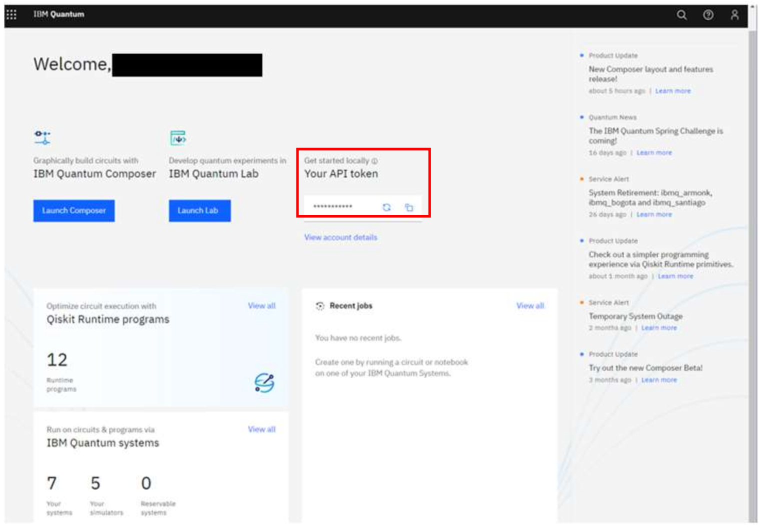
IBMid

New to IBM Quantum?
Create an IBMid account.

3

Get API token

- On logging in to https://quantum-computing.ibm.com/ you should be presented with a home screen:



IBM Quantum

Welcome, [redacted]

Graphically build circuits with
IBM Quantum Composer
Launch Composer

Develop quantum experiments in
IBM Quantum Lab
Launch Lab

Get started locally
Your API token
****
View account details

Optimize circuit execution with
Qiskit Runtime programs
View all
12
Runtime programs

Run on circuits & programs via
IBM Quantum systems
View all
7 Your systems
5 Your simulators
0 Reservable systems

Recent jobs
View all
You have no recent jobs.
Create one by running a circuit or notebook on one of your IBM Quantum Systems.

Product Update
New Composer layout and features release!
about 5 hours ago | Learn more

Quantum News
The IBM Quantum Spring Challenge is coming!
16 days ago | Learn more

Service Alert
System Retirement: ibmq_armonk, ibmq_bogota and ibmq_santiago
26 days ago | Learn more

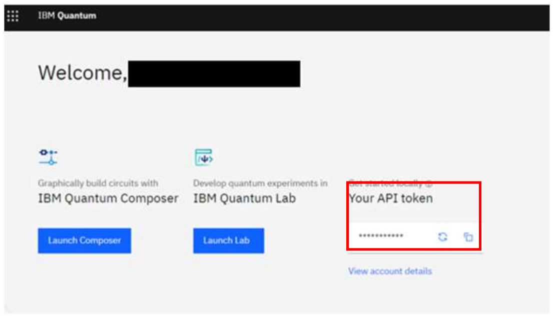
Product Update
Check out a simpler programming experience via Qiskit Runtime primitives.
about 1 month ago | Learn more

Service Alert
Temporary System Outage
2 months ago | Learn more

Product Update
Try out the new Composer Beta!
3 months ago | Learn more

4

Get API token



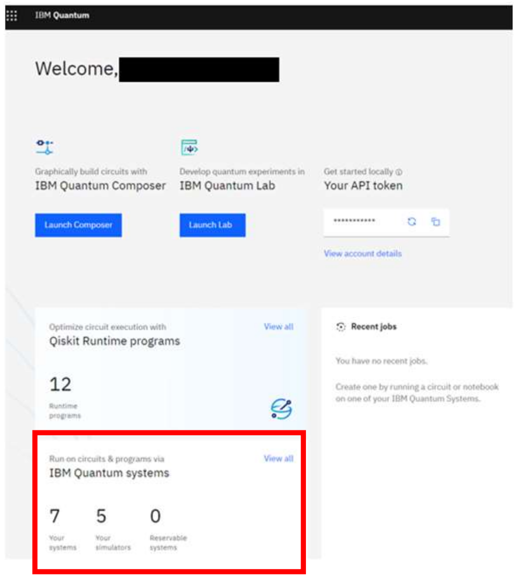
- The API token can be easily copied from this page.
- To *store* the IBMQ API token for use with InQuanto, open a python shell or notebook and use:

```
> from pytket.extensions.qiskit import set_ibmq_config
> set_ibmq_config(ibmq_api_token='XXXX')
```

- **As well as the API token, you will need to specify a machine to emulate** (see following pages)
- Note that generating a new API token (♲) will stop the old token working

5

Choosing a quantum device

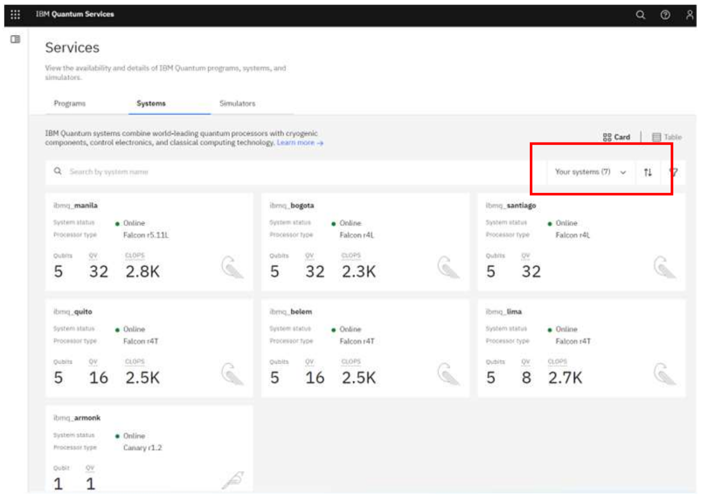


- With the API token set, we can choose a machine to run on/simulate
- The list of available machines can be found by clicking 'View all' on the IBM Quantum systems section of the IBMQ home page

6

Choosing a quantum device

- When on the device page, the list can be filtered to show only devices available to you.



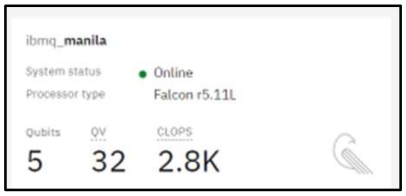
Services

View the availability and details of IBM Quantum programs, systems, and simulators.

IBM Quantum systems combine world-leading quantum processors with cryogenic components, control electronics, and classical computing technology. Learn more →

System	System status	Processor type	Qubits	QV	CLOPS
ibmq_manila	● Online	Falcon r5.11L	5	32	2.8K
ibmq_bogota	● Online	Falcon r4L	5	32	2.3K
ibmq_santiago	● Online	Falcon r4L	5	32	
ibmq_quito	● Online	Falcon r4T	5	16	2.5K
ibmq_belem	● Online	Falcon r4T	5	16	2.5K
ibmq_lima	● Online	Falcon r4T	5	8	2.7K
ibmq_armonk	● Online	Canary r1.2	1	1	

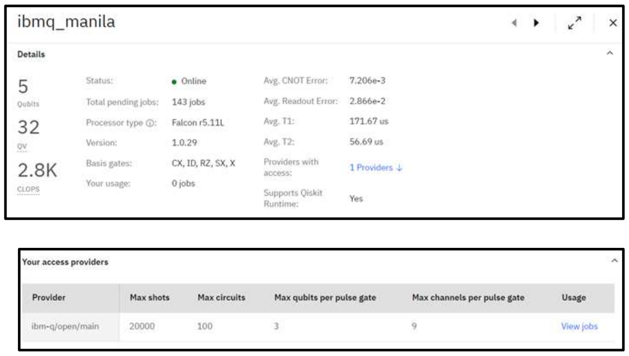
Choosing a quantum device


ibmq_manila

System status ● Online

Processor type Falcon r5.11L

Qubits 5 QV 32 CLOPS 2.8K


ibmq_manila

Details

5 Qubits

32 QV

2.8K CLOPS

Status:	● Online	Avg. CNOT Error:	7.206e-3
Total pending jobs:	143 jobs	Avg. Readout Error:	2.866e-2
Processor type ⓘ:	Falcon r5.11L	Avg. T1:	171.67 us
Version:	1.0.29	Avg. T2:	56.69 us
Basis gates:	CX, ID, RZ, SX, X	Providers with access:	1 Providers ↓
Your usage:	0 jobs	Supports Qiskit Runtime:	Yes

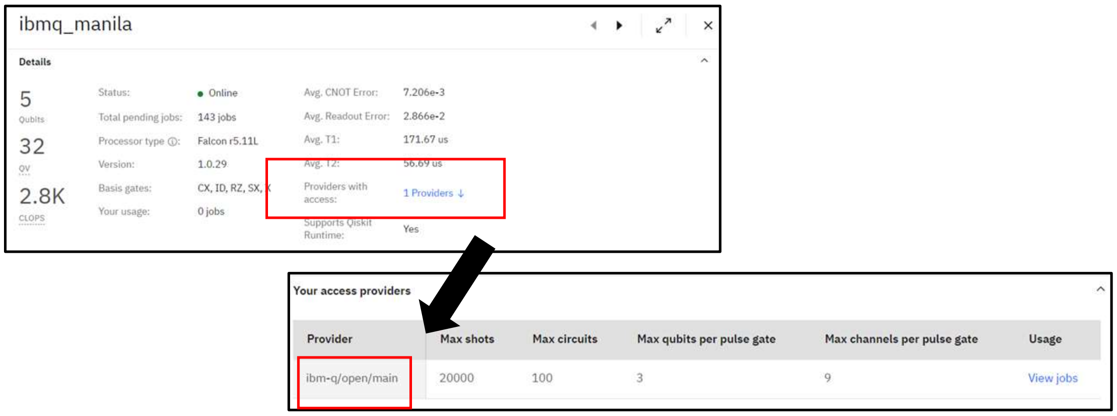
Your access providers

Provider	Max shots	Max circuits	Max qubits per pulse gate	Max channels per pulse gate	Usage
ibm-q/open/main	20000	100	3	9	View jobs

- Clicking on a listed machine will show the user many details about that machine. For example; the gate error, or the number of jobs queueing to run on it.

Setting up the pytket-qiskit backend

- To get the machine details needed for computing, scroll down or click the ‘Providers with access’ link.



ibmq_manila

Details

5 Qubits	Status:	Online	Avg. CNOT Error:	7.206e-3
	Total pending jobs:	143 jobs	Avg. Readout Error:	2.866e-2
32 QV	Processor type:	Falcon r5.11L	Avg. T1:	171.67 us
	Version:	1.0.29	Avg. T2:	56.69 us
2.8K CLOPS	Basis gates:	CX, ID, RZ, SX, X	Providers with access:	1 Providers ↓
	Your usage:	0 jobs	Supports Qiskit Runtime:	Yes

Your access providers

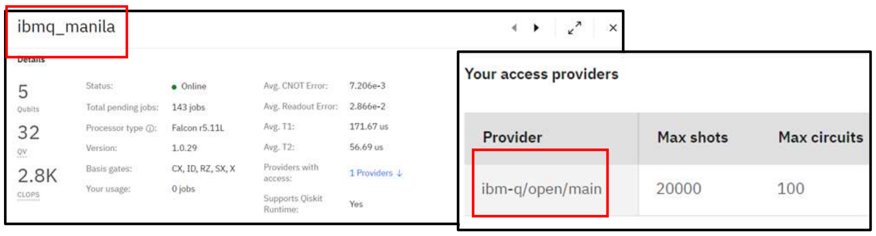
Provider	Max shots	Max circuits	Max qubits per pulse gate	Max channels per pulse gate	Usage
ibm-q/open/main	20000	100	3	9	View jobs

9

Setting up the pytket-qiskit backend

In your python shell or notebook, the machine details can be set, for example using:

```
>from pytket.extensions.qiskit import IBMQEmulatorBackend
>backend = IBMQEmulatorBackend(backend_name="ibmq_manila",
hub="ibm-q", group="open", project="main")
```



ibmq_manila

Details

5 Qubits	Status:	Online	Avg. CNOT Error:	7.206e-3
	Total pending jobs:	143 jobs	Avg. Readout Error:	2.866e-2
32 QV	Processor type:	Falcon r5.11L	Avg. T1:	171.67 us
	Version:	1.0.29	Avg. T2:	56.69 us
2.8K CLOPS	Basis gates:	CX, ID, RZ, SX, X	Providers with access:	1 Providers ↓
	Your usage:	0 jobs	Supports Qiskit Runtime:	Yes

Your access providers

Provider	Max shots	Max circuits
ibm-q/open/main	20000	100

10

Setting up the pytket-qiskit backend

- In the previous example we have set up an emulation of shots on the 5 qubit IBMQ Manila machine using the 'free' queue.
- To run on actual hardware, just change `IBMQEmulatorBackend` to `IBMQBackend`
- Please be considerate when queueing jobs and avoid the free queue if possible.
- When submitting hardware experiments, you will need to keep the python kernel running until results have been returned and processed.

11