AWS BRAKET QUANTUM COMPUTING: HOW TO GET STARTED



Amazon now offers a quantum computing simulator with actual quantum computing access. In this article, we:

- Explain AWS Braket
- Look at pricing and hardware options
- Give directions for running the local simulator
- Run some sample code

What's AWS Braket?

In 2019, AWS released Braket, a fully managed service that offers quantum computing. Just this month, though, the more exciting thing happened: Braket became available to the general public! Braket is aimed at developers and researchers who want to build, experiment, and test quantum algorithms on actual quantum circuit simulators and hardware.

Quantum computing is a big deal (though how much of a big deal is up for debate). That's because it has the potential to solve computational problems that are, to this day, unable to solve with classic computers. Quantum computing might be able to revolutionize machine learning, drug discovery, financial markets, and dozens of other fields.

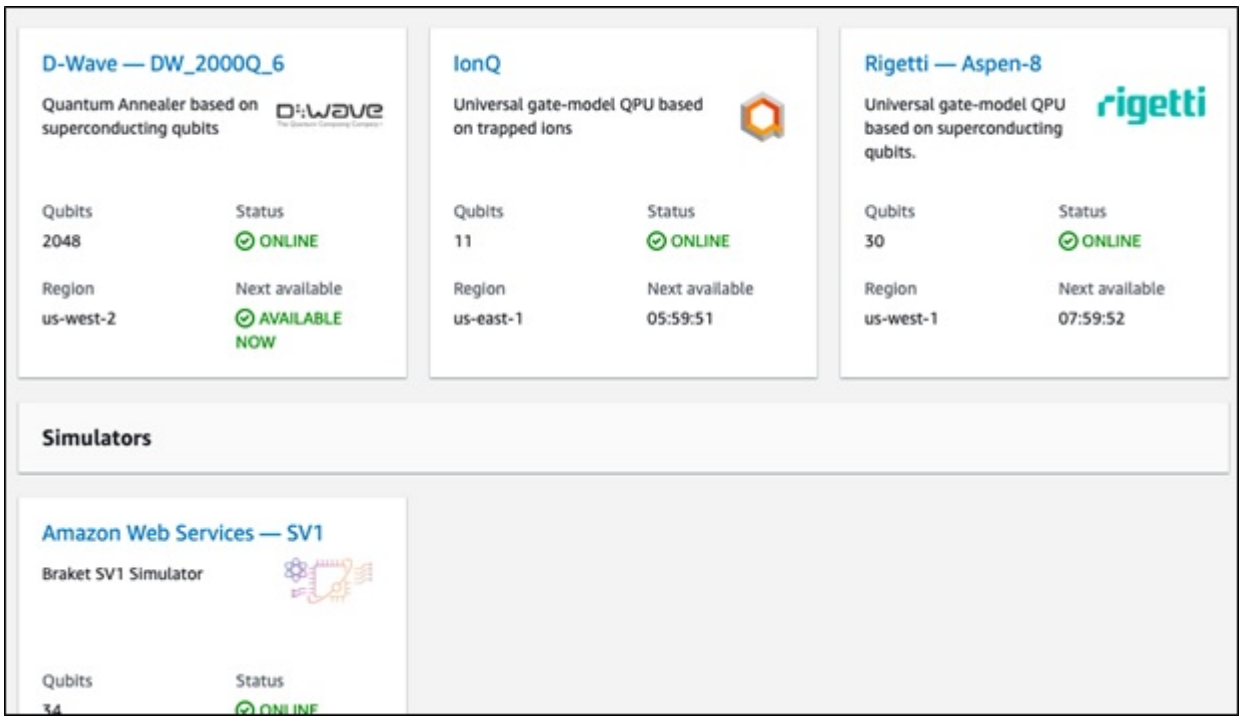
But traditional quantum computing, which dates to the early 1980s, is expensive and resource heavy. With AWS Braket, Amazon offers an affordable way for developers, researchers, scientists—anyone—to experiment with quantum computing. Braket allows you to:

- Design and build quantum algorithms (from scratch or pre-built)

- Choose simulators to test and run algorithms
- Run algorithms on different quantum computers
- Learn quantum computing and evaluate its potential for your organization

Qubit access: Paid quantum computing

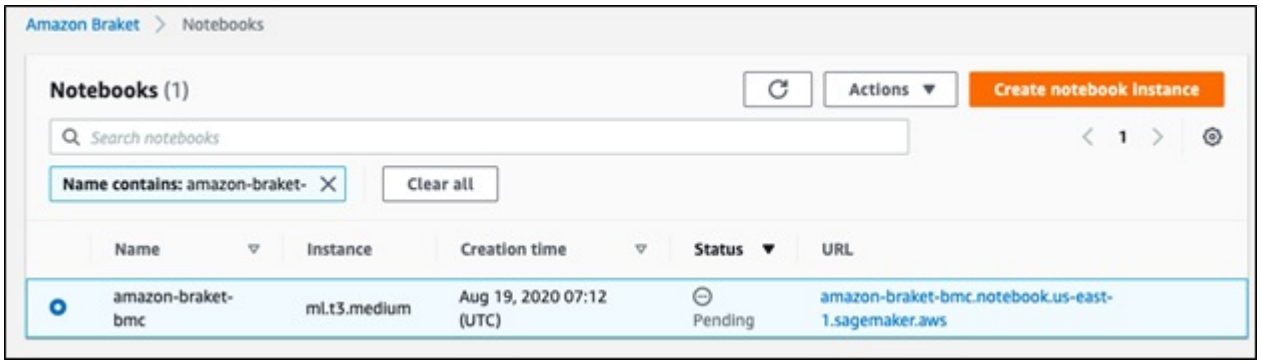
Braket offers free and paid versions. If you opt for the paid version, this is what the instances look like:



Amazon Braket with Jupyter Notebooks

You can also run a Jupyter Notebook to access the environment, which means you can write Python code to use it.

In Braket, Glue, and other AWS services, Amazon will offer you a Jupyter notebook. Install it on your own device to save money.



AWS Braket pricing

For this tutorial, I used a Jupyter notebook, size ml.t3.medium, which cost $0.0582/per hour. The Amazon Braket simulator is billed at a rate of $4.50 per hour. Consult the Amazon website for prices on actual physical Qubits. (A qubit, short for quantum bit, is the most basic unit of quantum

information.)

Note: Set up billing alerts first, so you don't get blindsided by a high bill as you experiment with this.

Hardware options for Braket

You have these options for running Braket, either simulation or the actual Quantum computing hardware:

- Local emulator (Free)
- Managed simulator
- Quantum annealer on D-Wave 2000Q quantum computer
- Gate-based QPU on the Rigetti Aspen-8 quantum computer

Free SDK for Braket

Now, we'll walk you through how to use option 1, the local simulator, free on your own device.

First, install AWS CLI and configure credentials.

```
sudo apt install awscli
aws configure
```

Create an S3 bucket to contain Braket working files. Then, put that name into the Python code:

```
aws s3 mb s3://amazon-braket-output-(some name)
```

Upgrade to Python 7 if you do not have it.

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt upgrade
sudo apt install python3.7
```

Install virtualenv and activate Python 3.7 environment.

```
sudo apt-get virtualenv
virtualenv py372 --python=python3.7
source py372/bin/activate
```

At this time, in August 2020, you can't get Braket from pip. So, instead, install it like this:

```
git clone https://github.com/aws/amazon-braket-sdk-python.git
cd amazon-braket-sdk-python
pip install .
```

Install Amazon boto3:

```
pip install boto3
```

You can use vi and python from the command line, but you should use Jupyter notebooks. (Installed on your machine for lower cost, as mentioned above.)

Now, launch the Jupyter notebook on an IP address that you can reach from the internet, assuming you are using a cloud server. In this case that is a 172. address.

Open firewall port 8888. If that port is busy Jupyter will pick 8889, 8890, etc., until it finds an open one. Then open Jupyter in the browser. Set a password using Jupyter notebook password but when you start Jupyter, as shown below, it will authenticate with a token.

```
pip install boto3
pip install jupyter
jupyter notebook password.
jupyter notebook --ip=paris2x

http://(public IP
address):8888/?token=7a423a9f391caa863cf9e1497ee8bf01f069f8a74fffbaf0
```

Next, write this code. It will grab the credentials you configured above and then run a local (free) emulator.

```
import boto3
from braket.aws import AwsDevice
from braket.circuits import Circuit

account_id = boto3.resource('iam').CurrentUser().arn.split(':')


from braket.devices import LocalSimulator

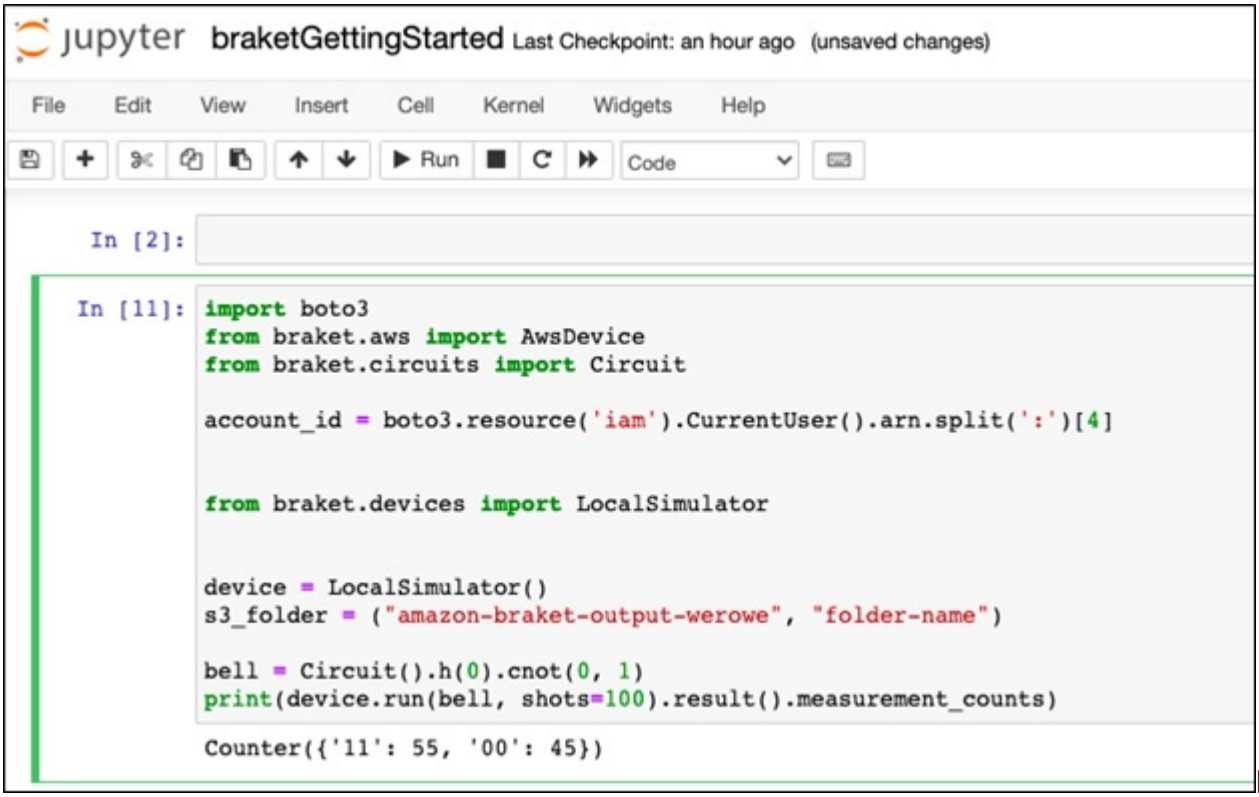

device = LocalSimulator()
s3_folder = ("amazon-braket-output-werowe", "folder-name")

bell = Circuit().h(0).cnot(0, 1)
print(device.run(bell, shots=100).result().measurement_counts)
```

Produces this output:

```
Counter({'11': 55, '00': 45})
```

It looks like this when run in Jupyter.



Happy quantum experimenting!

Additional resources

For more on AWS, machine learning, and related technologies, check out these resources:

- BMC Machine Learning & Big Data Blog
- AWS Guide, with 15+ articles and tutorials
- AWS Sagemaker vs Amazon Machine Learning
- Machine Learning with TensorFlow and Keras, with 10+ tutorials
- Python vs Java: Why Python is Becoming More Popular Than Java