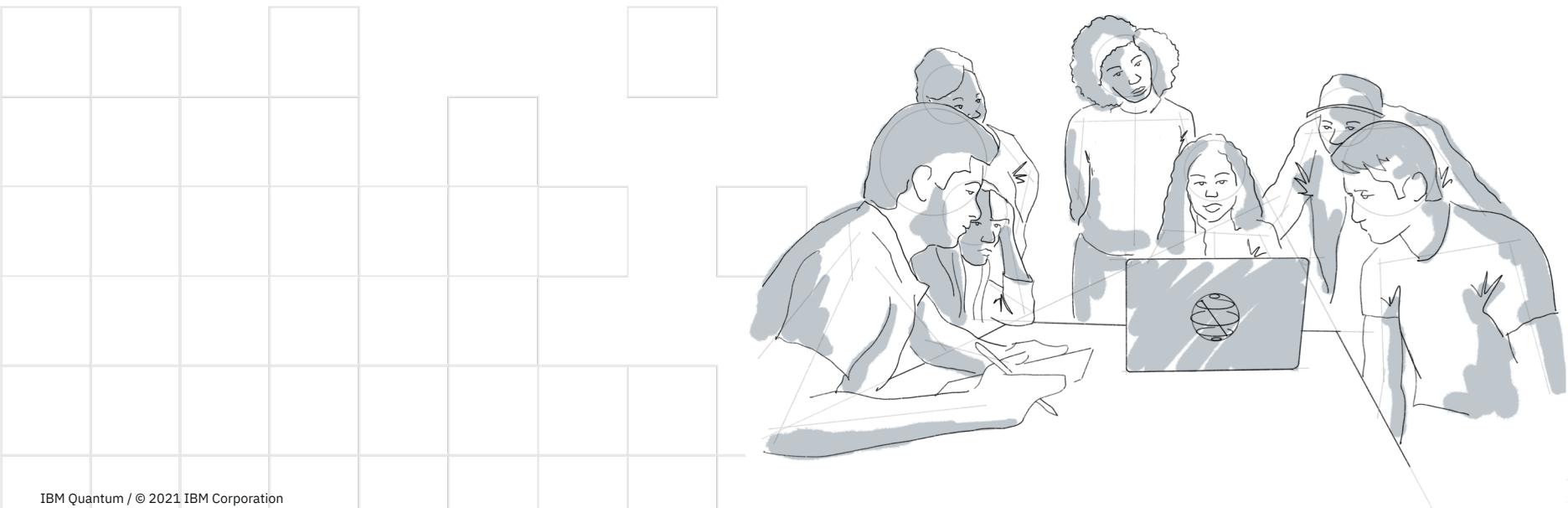


Qiskit Demo Day: Qat Computer



Author : Michaël Rollin

Voice : Neural network cloned voice of Michaël Rollin





Problem ?

- Local Qiskit env and its dependencies complicate to maintained
- Python virtualenv & Conda env, useful but not ideal in the long run
- Making a paper or demo project work for long time is complicated



Solution → Qat Computer !

The **Qat Computer** is first of all a **Qiskit base docker image** designed for any of my quantum platform. Like my Jupyter Lab, or my quantum games.

Secondary it's also a container service designed for batchs or services executions.

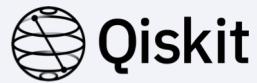
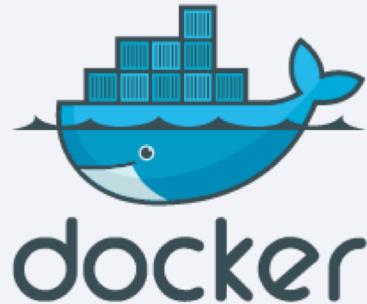
Last (but not least), it contained a Go binary designed to run python project and checking the status of the Qiskit env inside.

Container base

- Allow build and run faster
- Same base for any project
- Automatic update
- Base on last Ubuntu LTS (22.04)

Examples :

- Base for Jupyter quantum lab
- Base for quantum game
- Base for apps/API



How to use it



- Just call the main image in your Dockerfile and add your project around

2 versions available :

- Classical version with Qiskit minimal setup
- Full version with every Qiskit sub-libraries

```
FROM mickahell/qatcomputer-full:latest

# Var for labels
ARG GITHUB_ACTOR
ARG GITHUB_REPOSITORY
ARG GITHUB_REF

ARG DEBIAN_FRONTEND=noninteractive
ENV TZ=Europe/Paris

LABEL org.opencontainers.image.title="Quantum Lab" \
      org.opencontainers.image.authors=${GITHUB_ACTOR} \
      org.opencontainers.image.vendor=${GITHUB_REPOSITORY} \
      org.opencontainers.image.source="https://github.com/mickahell/quantum_lab" \
      org.opencontainers.image.url="https://github.com/mickahell/quantum_lab/tags" \
      org.opencontainers.image.description="Docker image for quantum algorythm" \
      org.opencontainers.image.documentation="https://github.com/mickahell/quantum_lab/blob/main/README.md" \
      org.opencontainers.image.os="Ubuntu Focal" \
      org.opencontainers.image.version=${GITHUB_REF}

# OS requirements
RUN apt-get update -yq \
&& apt-get install -yq \
  python3-tk \
  graphviz \
  git \
&& apt-get dist-upgrade -yq \
&& apt-get clean -yq

# Add script & data
ADD build/* /opt/quantum_lab/build/
ADD data/ /opt/quantum_lab/data/
ADD start_jupyter.sh /opt/quantum_lab/

# General & env requirements
RUN pip install --upgrade pip setuptools
RUN pip install -r /opt/quantum_lab/build/requirements.txt

WORKDIR /opt/quantum_lab/data
VOLUME /opt/quantum_lab/data/share

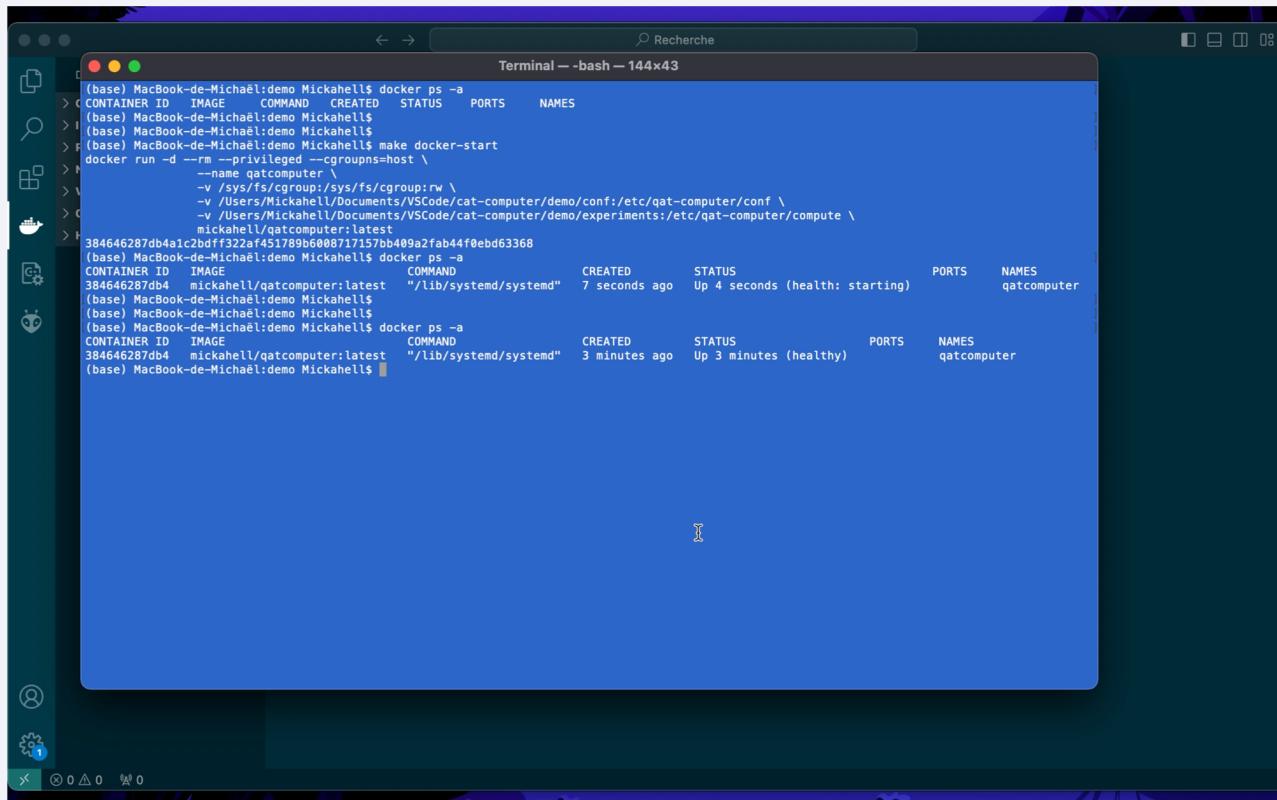
EXPOSE 8888

CMD /bin/bash
```

Initctl emulation



- Container as a service
- Allows usage in GitHub Action
- Allows usage in testing mode with VSCode (or any ide compatible with container)
- Qiskit cluster service (Quantumbernetes from K8S → Q13S ?)



```
(base) MacBook-de-Michaël:demo Mickahell$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
(base) MacBook-de-Michaël:demo Mickahell$ (base) MacBook-de-Michaël:demo Mickahell$ (base) MacBook-de-Michaël:demo Mickahell$ make docker-start
docker run -d --rm --privileged --cgroupsns-host \
           --name qatcomputer \
           -v /sys/fs/cgroup:/sys/fs/cgroup:rw \
           -v /Users/Mickahell/Documents/VSCode/cat-computer/demo/conf:/etc/qat-computer/conf \
           -v /Users/Mickahell/Documents/VSCode/cat-computer/demo/experiments:/etc/qat-computer/compute \
           mickahell/qatcomputer:latest
384646287db4a1c2bdf322af451789b60087f7157bb409a2fab44f0ebd63368
(base) MacBook-de-Michaël:demo Mickahell$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
384646287db4 mickahell/qatcomputer:latest "/lib/systemd/systemd" 7 seconds ago Up 4 seconds (health: starting) qatcomputer
(base) MacBook-de-Michaël:demo Mickahell$ (base) MacBook-de-Michaël:demo Mickahell$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
384646287db4 mickahell/qatcomputer:latest "/lib/systemd/systemd" 3 minutes ago Up 3 minutes (healthy) qatcomputer
(base) MacBook-de-Michaël:demo Mickahell$
```

Go API



- Allow to run your quantum app as a batch/web/service/... container
- Allows usage in GitHub Action
- Give the status of the Qiskit env inside



```
(base) MacBook-de-Michaël:demo Mickahell$  
(base) MacBook-de-Michaël:demo Mickahell$  
(base) MacBook-de-Michaël:demo Mickahell$  
(base) MacBook-de-Michaël:demo Mickahell$ █
```

Example : Quadratic Program



Demo with QUBO project :

- Run QAOA
- Run Grover

The screenshot shows a GitHub Actions pipeline for a project named "qat-computer" under the user "mickahell". The pipeline is titled "Qiskit demo day #17". The "Actions" tab is selected. A summary card shows a successful run triggered 1 minute ago by "mickahell #43" with a total duration of 46 seconds. Below the summary, the "demo.yml" workflow file is expanded, showing two jobs: "qubo" and "qnn", both of which completed successfully. The "qubo" job took 35 seconds and the "qnn" job took 38 seconds.

Example : Quantum Neural Network

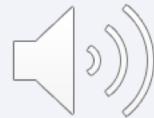


Demo with QNN project :

- Trained your model
- Generated your data

The screenshot shows a GitHub Actions pipeline summary for a repository named 'qat-computer' under user 'mickahell'. The pipeline is titled 'Qiskit demo day #17'. The summary card indicates the run was triggered 3 minutes ago by 'mickahell #43 39-demo', with a status of 'Success' and a total duration of '49s'. Below the card, the 'demo.yml' workflow file is shown, which triggers 'qubo' and 'qnn' jobs on pull requests. Both jobs completed successfully: 'qubo' took 36s and 'qnn' took 40s.

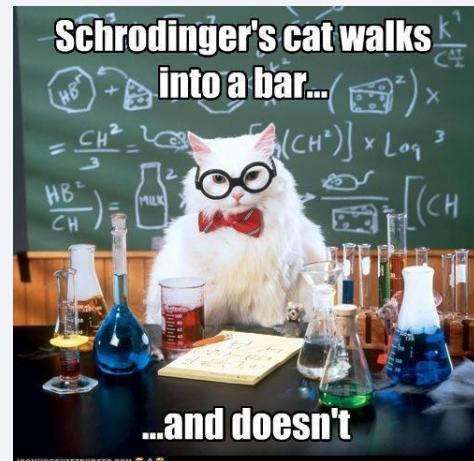
Job	Status	Duration
qubo	Success	36s
qnn	Success	40s



Notable info

- The Quantum Lab base on this image was use during :
- Last Qiskit challenge by ~ 150 participants (a specialized version was builded for)
- Last Qiskit Global Summer School ~ 100 participants

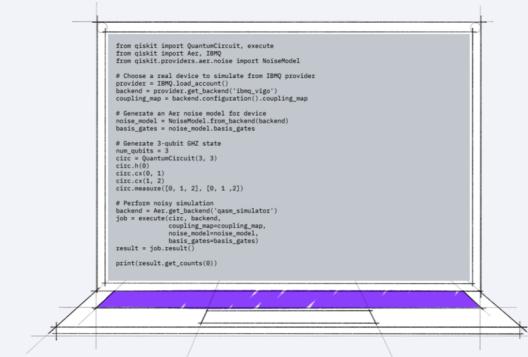
No Qat was dead/alive during the testing of this project.





Next step

- Schedule build synchronized with Qiskit releases
- Unit tests endpoint for the Go API
- Be use as full Qiskit serverless infrastructure in DinD
(link with the official Qiskit Serverless project)
- Cool logo
- Some easter eggs



Thanks for your attention !



Michaël & his clone

All the code is available in the GitHub repository as well as the code of the demo.

<https://github.com/mickahell/qat-computer>

