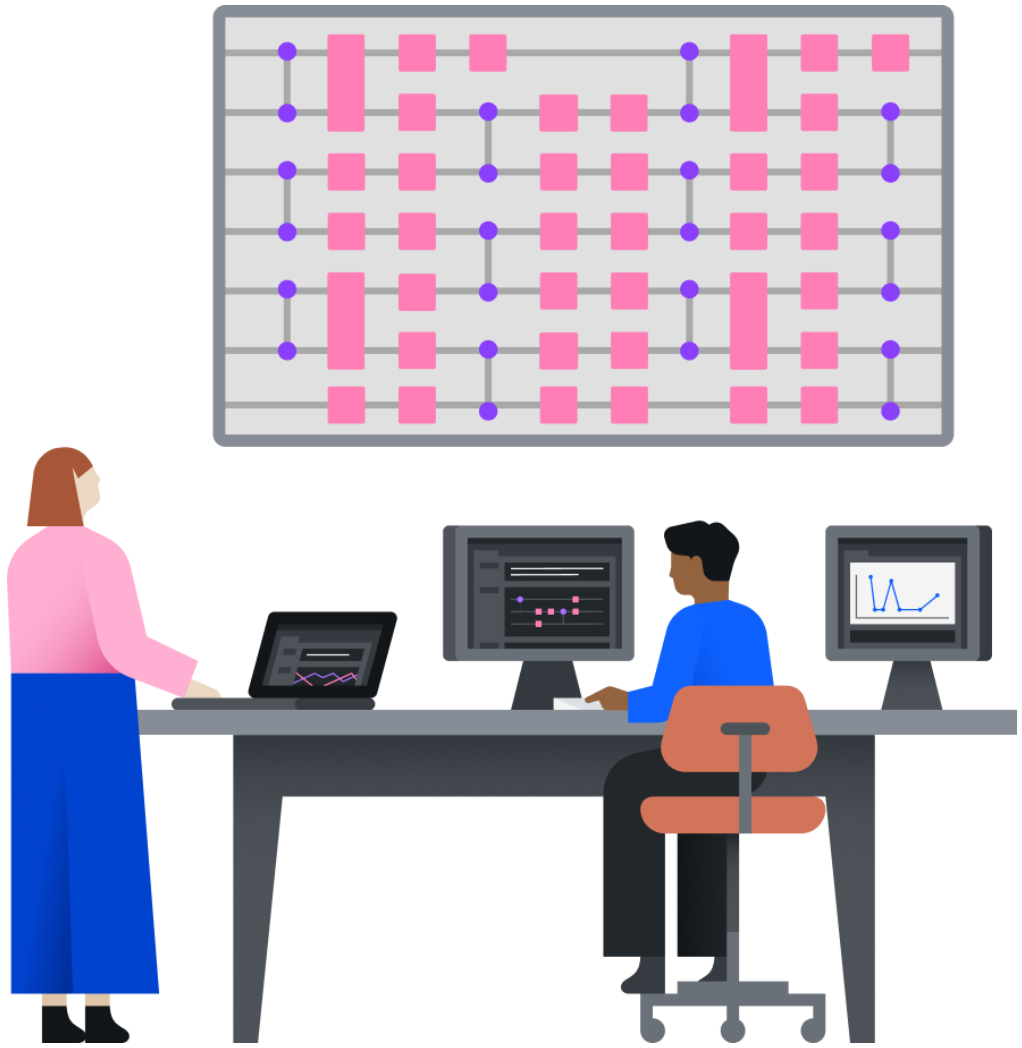
Fast track your development journey through the

IBM **Quantum** Network

Performance & optimization lab



IBM Quantum Network members can **submit their eligible workload for expert review** through the new IBM Quantum Platform, hosted on IBM Cloud®

IBM **Quantum** Network

Performance & optimization lab

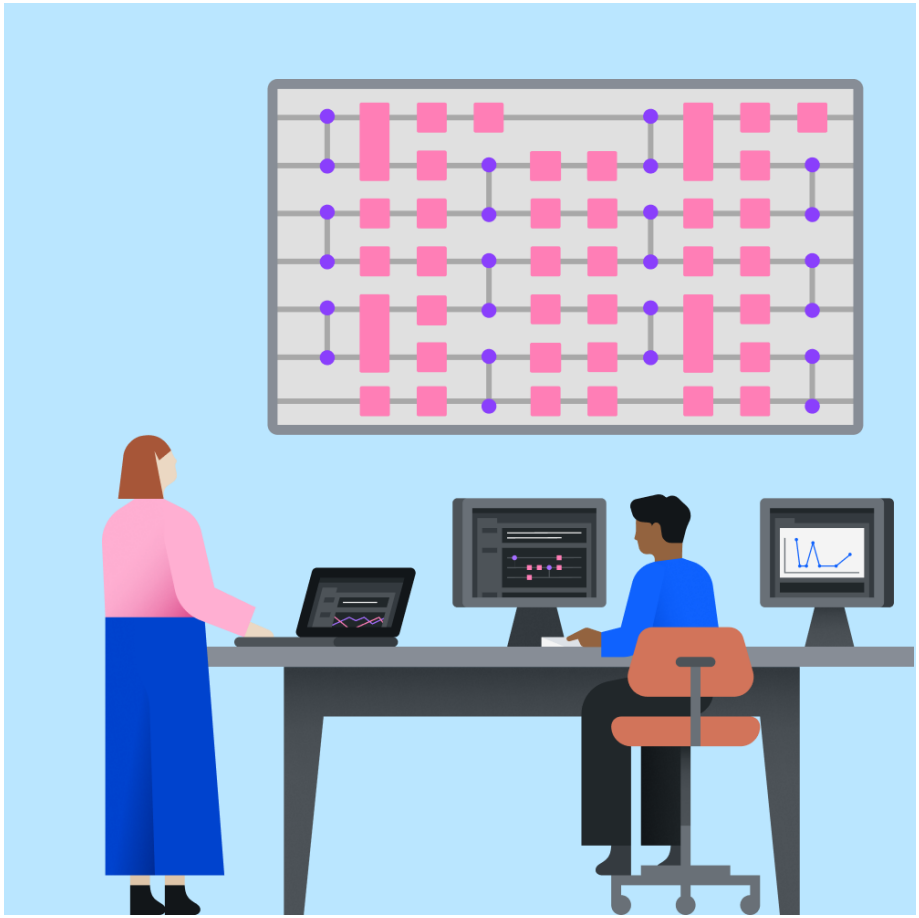
IBM Quantum Network members get exclusive access to fast track their development journey for eligible workloads

---

IBM Quantum Network members can submit their workload for expert review through the new IBM Quantum Platform, hosted on IBM Cloud®

Coordinated through their IBM Quantum engagement manager, all submissions will be evaluated by the IBM Quantum technical team.

Insights and recommendations are tailored to help improve efficiency, scalability, and performance and shared back directly or in a review session with partners.



Preparing your workload for submission

Ensure Code Eligibility

- ❑ Testable with the latest Qiskit* and Qiskit Runtime version
- ❑ Code does not include any deprecated modules/ features
- ❑ Code does not require setup or installation of third-party libraries or licenses
  *We are **not** able to support Qiskit community, non-quantum-related, or third party quantum-related libraries.*
- ❑ Includes quantum circuits, operators, and/or runtime operations
- ❑ Does not contain any confidential or proprietary information
- ❑ Must be of reasonable length (max of ~1000 lines of code)

*Qiskit refers to the Qiskit SDK, Qiskit Runtime, Qiskit Functions Catalog, IBM Circuit Function, Qiskit addons, Qiskit Transpiler Service and any other IBM Quantum-owned & managed software feature

Prepare your code to speed up troubleshooting by enabling environment reproduction, and easy code exchange.

- ❑ Provide a `requirements.txt` file with a list of the python/pypi packages required to run the notebook, including specific version numbers to enable environment reproduction
- ❑ Omit API token by using a function to store your unique API token. This makes it easy to pass the code back and forth.
- ❑ Organize your code so that you can run many trials in a loop, collecting statistics as you go.
- ❑ Make a Minimum Reproducible Example (MRE) by removing *any* extraneous details and unused code, so that the team can focus on relevant code blocks.
- ❑ If it's possible, pinpoint the line(s) of code that not giving an expected result. The following is an idea example:
  - *If the algorithm is working correctly, the plot in cell 21 should show a correct coloring, with no neighboring nodes of the same color. In fact, there are pairs of neighboring red nodes, indicating failure.*
- ❑ All materials for review and supporting documentation can be packaged as a .zip and submitted via the IBM Cloud Support.

Recommendations before your workload submission

Review IBM Quantum Learning and Tutorials to learn how to apply Qiskit to common quantum computing use cases.
These materials include tutorials to get started, building workflows toward advantage, and advanced Qiskit capabilities. We recommend starting here for tips on choosing initial parameters.

Remove old code, unused calculations, and variables. This makes it easier to reason about the code, and removes potential sources of error.

Organize your code so that you can run many trials in a loop, collecting statistics as you go. Tune the parameters (there may be others) on the Aer simulator till the statistics are nearly optimal.

- Then run on a fake backend and address problems that arise.
- Then run on real hardware. By the time you run on real hardware you'll know the best you can possibly do statistically.
  - First, choose parameters where the results on simulators are nearly perfect. You'll need all the help you can get when tuning error mitigation parameters.
  - When you are able to make this work, increase the size of your problem.

Additional Questions to ask as you work through your problem:

- Did you run the Qiskit code on a simulator, such as a FakeBackend? What were the results?
- What results were you expecting and why do you define the outputs “poor”?
- Does the algorithm ever find a correct solution? How often?
- What are the expected outcomes? What is likely contributing to the deviation?

All Performance & Optimization lab questions will be handled through the **IBM Cloud Support Center**

IBM Quantum Support & FAQs:
https://quantum.cloud.ibm.com/docs/en/support

Get help with IBM Quantum Platform and Qiskit

Premium Plan, On-Prem Plan, Direct Access, Flex Plan, Pay-As-You-Go Plan, special programs, and Open Plan (excludes Trial accounts):

All support inquiries must be submitted by creating a case in the IBM Cloud® Support Center ↗ and selecting 'Qiskit Runtime' in the Topic field. Ensure you have selected the appropriate IBM Cloud account (your institutional cloud account or any non-trial account) in the account switcher.

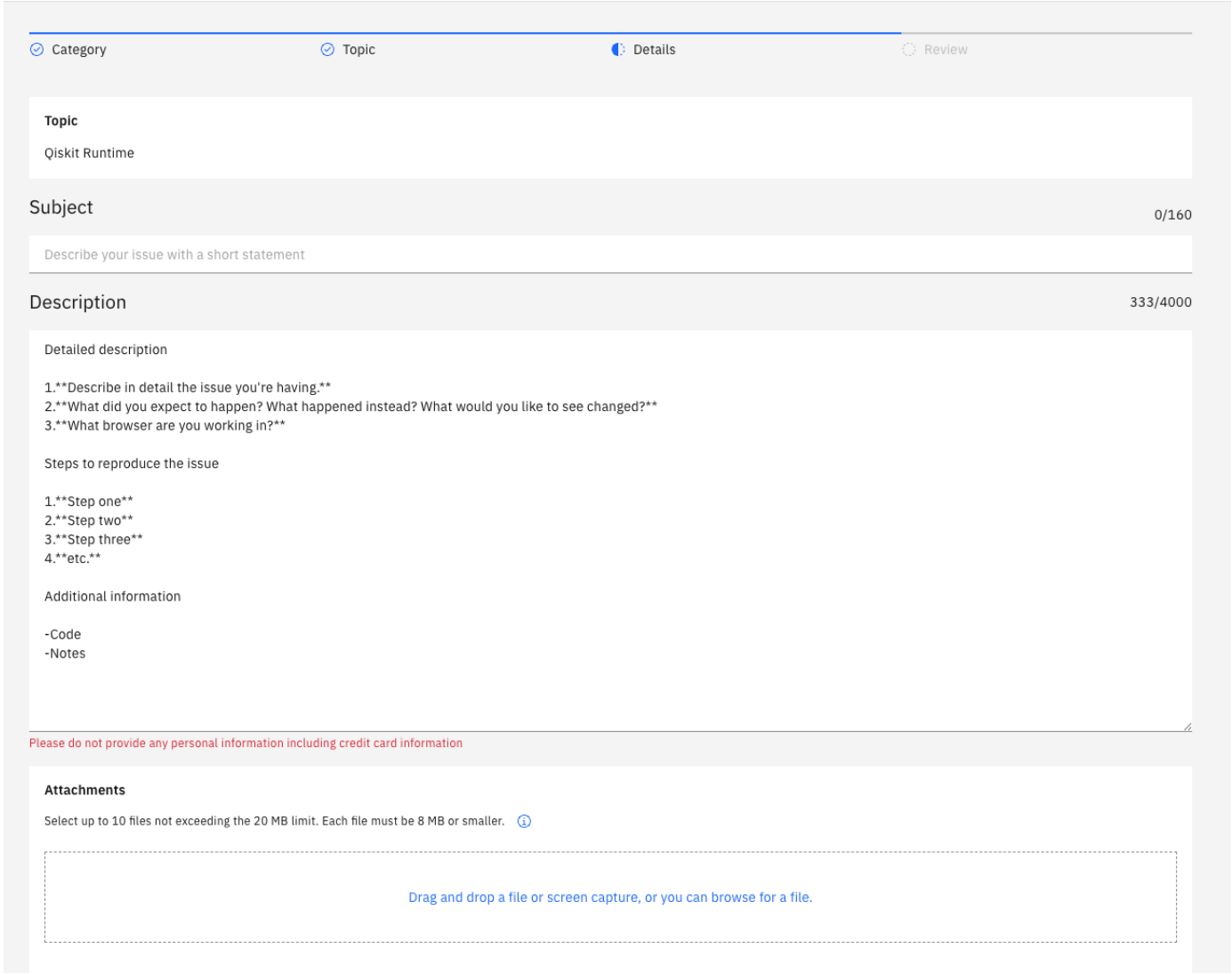
Please reach out for any technical matter, including platform issues, job errors, Qiskit guidance, or broader quantum computing questions. Our team includes technical experts ready to assist across this full range of topics.

Community support:

Have a Trial account or looking for other support options? Join the Qiskit Slack community ↗ to get help from our large community of users and experts. In Slack you will find topical channels such as #qiskit-runtime, #ibm-quantum-platform, #qiskit-transpiler, #qiskit-ecosystem, and many more.

You can also use Stack Exchange ↗ for support from the wider community about quantum computing and programming-related questions.

Creating a case in the **IBM Cloud Support Center**



IBM Cloud Support Center:

https://cloud.ibm.com/unifiedsupport/cases/form

1. Ensure you have selected the appropriate IBM Cloud account (your institutional cloud account or any non-trial account) in the account switcher.
2. Select ***Qiskit Runtime*** as Topic
3. Include *Performance & Optimization Lab* in the subject, along with a detailed description of your problem
4. In your description, include a description of the supporting materials, along with where you are getting stuck in the code and what you are expecting in your results
5. Attach code and supporting documentation as a .zip file, include a requirements text and code as an MRE