QuixApps

Build, deploy and host any test data analysis application in one central environment

[WHAT IT IS]

Host your tools in a central workspace your whole team can access

QuixApps make it easy to share the tools you would normally run on your own machine: a dashboard for viewing traces, an automatic data transformation, or a simulation model. Once your data sources are connected to Quix, plugging that data into an app is simple. You can prototype the application code in a web-based IDE, click Deploy, and the app is live within minutes. If coding is not your strength, the Quix IDE includes QuixAI, which helps you write, test and debug your code before you deploy it.

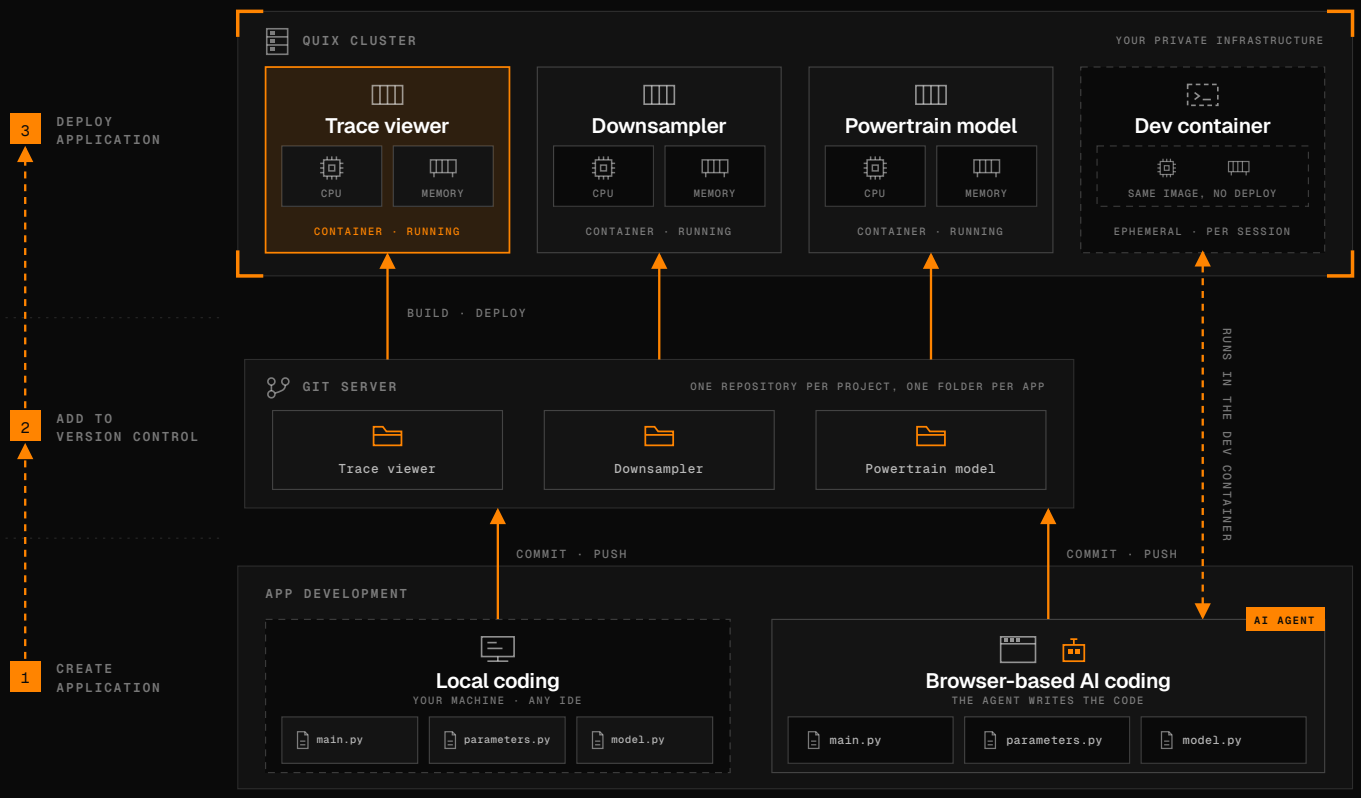
[THE PROBLEM]

Most testing teams have a fragmented set of internal tools running on local PCs

An engineer writes a genuinely useful analysis dashboard. Making it available to the team means finding a server, negotiating with IT, setting up authentication and certificates, and then keeping the whole thing patched.

So the dashboard gets passed around as screenshots instead. Or the team hosts it themselves, quietly taking on an extra DevOps burden when they do not really have capacity for it.

QuixApps solve this problem by allowing a test engineer to take code they would normally run on their PC and move it to a centralized environment, where it runs continuously with its own dedicated CPU and RAM — without the complexity of a platform like AWS or Azure.



[HOW IT WORKS]

From a local codebase to a fully hosted, team-wide service

Take a typical vehicle test scenario: an engineering team needs a live dashboard to monitor dyno sessions, plotting real-time channel telemetry against hardware safety limits alongside the active test configuration. Here is how that application moves onto Quix.

[01] Create an application

Start with a sandboxed environment, either on your local machine or in the Quix cloud IDE. Write your code using the Quix Python SDK to bring in data for testing — live data from a Quix Pipeline, or test traces stored in Quix Lake. With QuixAI inside the Quix IDE you do not need to install the SDK or learn it first: chat with QuixAI and let it guide you through retrieving and analysing your data.

[02] Add the code to version control

QuixApps use Git to store and retrieve code files. If you already run a Git server, commit your application files to a repository and configure Quix to retrieve them from there. If you do not use Git yet, Quix includes a managed Git server to get you started: write your code in the Quix IDE, save it, and Quix automatically commits the file and updates the version history — so you can use Quix without needing to learn Git.

[03] Deploy the application

Once the app is tested, deploy it as a Docker container, which behaves like a lightweight virtual machine. You configure the CPU and RAM per container, so each app gets the resources it needs to do its job efficiently. To host a web UI such as a Grafana dashboard, include a web server that exposes it. Sharing it with the team is a single toggle: the web app is then available to anyone with the link inside your network.

[04] Next steps

SEE IT WITH YOUR OWN DATA

Book a product demonstration and see how easy it is to build and deploy analysis applications.

BOOK A DEMO

READ THE DOCUMENTATION

Applications overview
What an application is and how it lives in Git

Deployments overview
Builds, services and jobs, resources, versions and logs

Code samples
Ready-to-deploy dashboards, APIs and connectors