Exercise 6 - Group Tests Into an Orchestration

Exercises · https://docs.functionize.com/hc/en-us/articles/43767015732887-Exercise-6-Group-Tests-Into-an-Orchestration

Lab handout — Lab 6
A printable worksheet for this exercise: the tasks, the exact prompts, a checkpoint and stretch challenges. Hand it round when you are onboarding a team.
Download the PDF (84 KB)

One test tells you about one flow. A suite tells you whether the application still works. In this exercise you will group your tests into an orchestration and run them as a single job. About ten minutes.

Watch: Functionize Studio 101: Build Your First Orchestration
Watch it end to end first if you would rather see the flow before doing it. The steps below are the same journey, against the demo store, in about ten minutes.

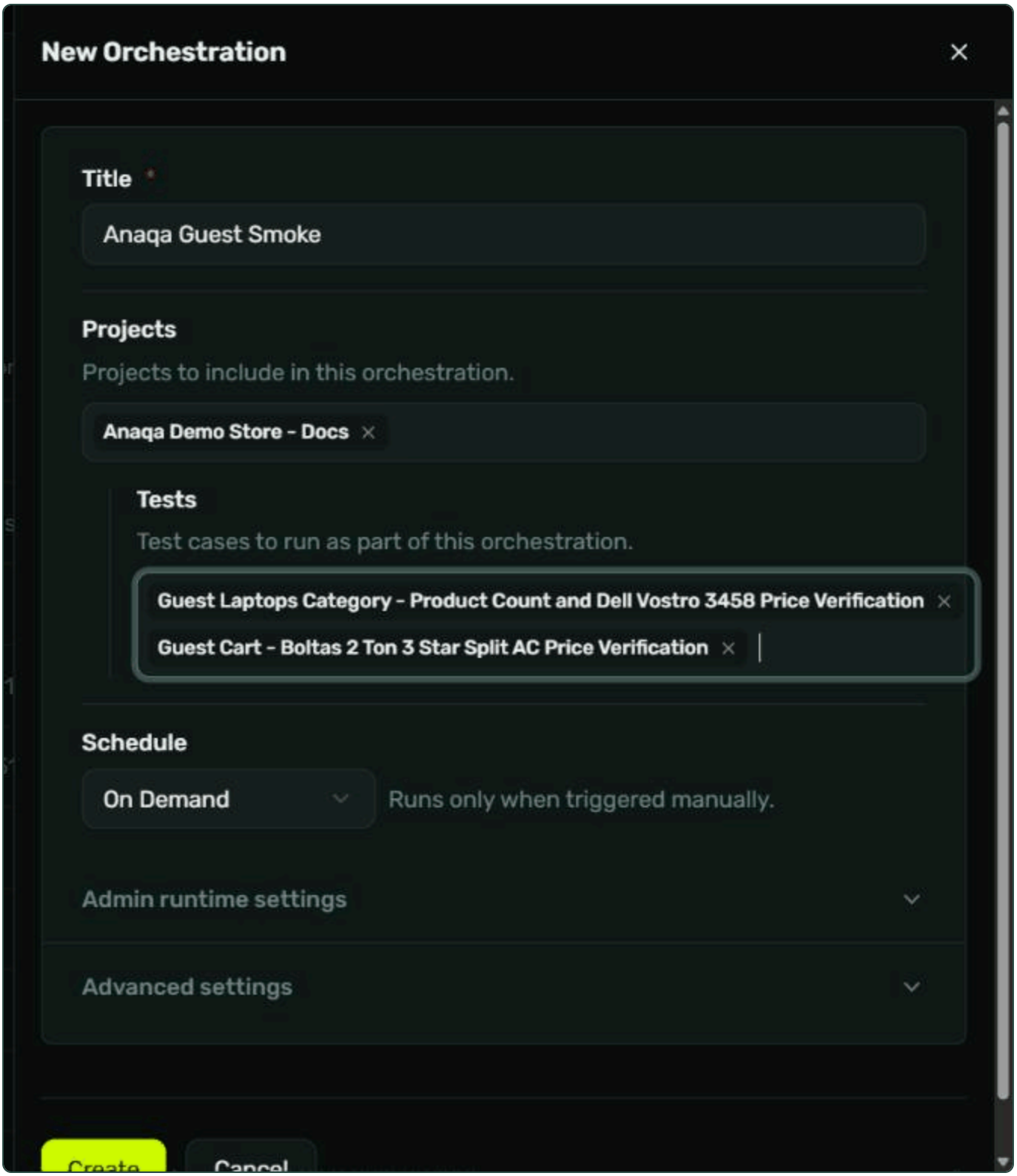


Watch: youtube.com/watch?v=TfMgmZLQqQE

You need at least two tests in your project.

Step 1 — Create the orchestration

Open **Orchestrations** in the left sidebar and choose **New Orchestration**.



Fill in four things:

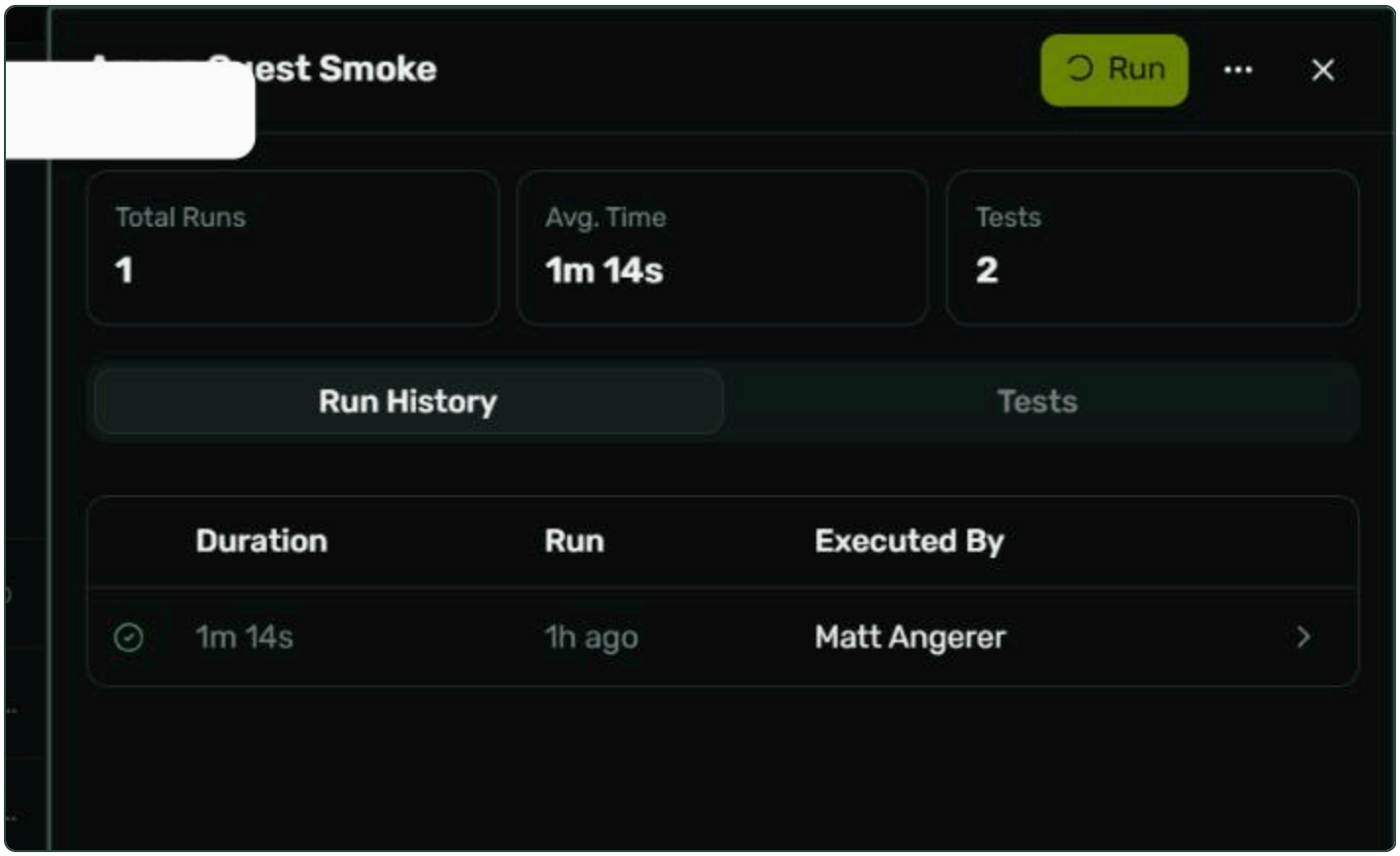
- **Title.** Name it after what it proves, not when it runs. *Guest smoke* ages better than *Nightly 2am*, which becomes a lie the first time someone changes the schedule.
- **Projects.** Pick your project first. The field takes more than one, and until at least one project is in it the **Tests** field has nothing to offer — it takes focus and then stays empty, which reads like a bug and is not one.
- **Tests.** Add your tests. They run as one job.

- **Schedule.** Leave it on **On Demand** for now.

Click **Create**.

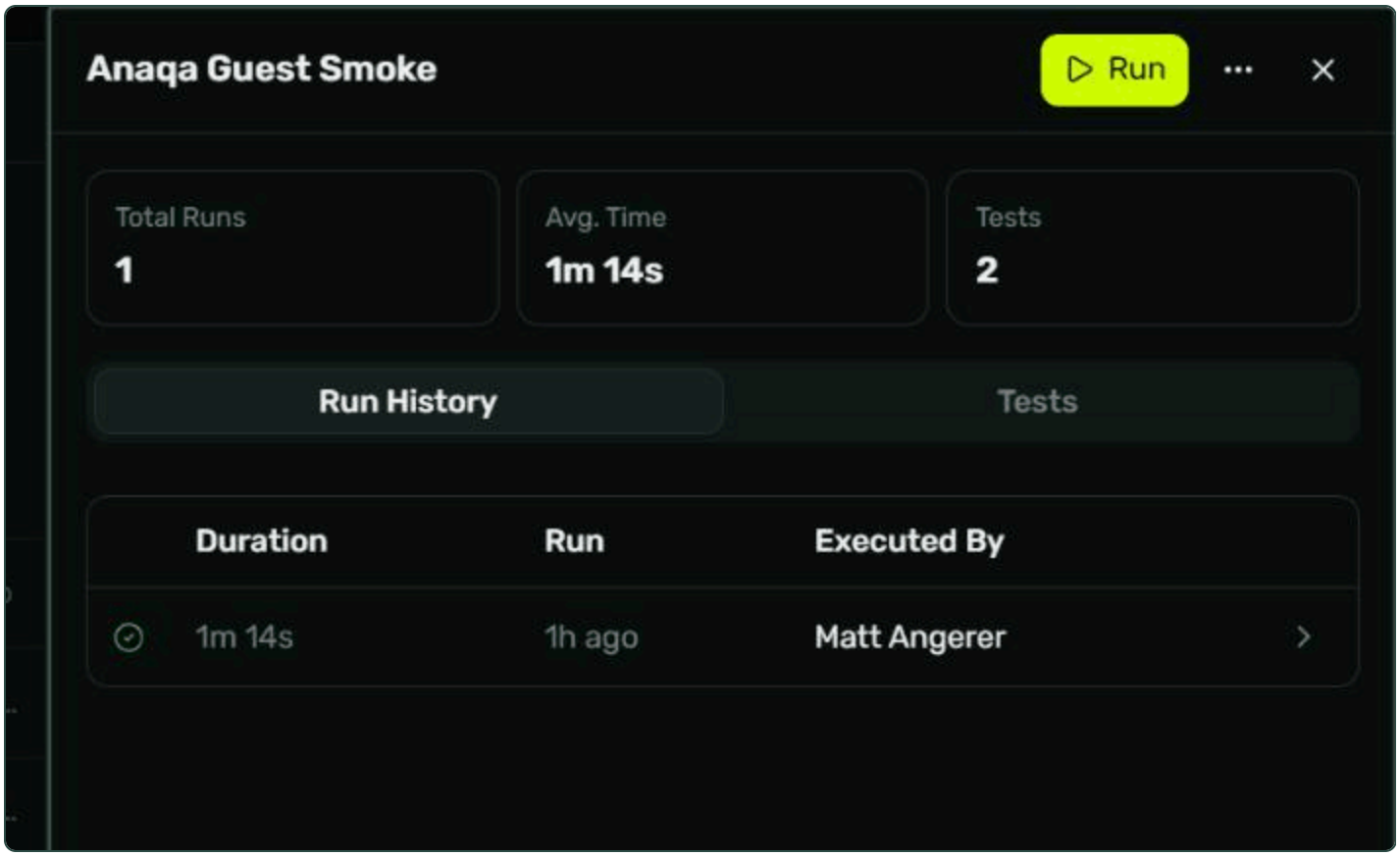
Step 2 — Run it

The orchestration opens on its own summary — total runs, average time, how many tests it contains, and a **Run History** tab. Click **Run**; the button becomes **Stop** while the job is in flight.



Step 3 — Read the result

When it finishes, the summary fills in and the run appears in the history with its result, duration and who triggered it:



Two tests ran as one job in 1 minute 14 seconds. The history entry covers the whole job, and you can still open any individual test to see its steps, screenshots and data.

If a run you have just triggered does not appear straight away, reload the page.

Step 4 — Think about the schedule before you set one

The schedule options are **On Demand**, **Hourly**, **Every 4 Hours**, **Every 12 Hours**, **Daily** and **Monthly**.

Two things worth saying plainly, because they are the mistakes people actually make.

Frequent schedules cost runtime. An hourly suite of fifty tests is a real commitment. Start daily and tighten it only if the feedback is genuinely too slow.

A scheduled suite nobody reads is worse than no suite, because it manufactures confidence. Decide who looks at the result, and where they will see it, before you set the schedule — not after.

For this exercise, leave it on On Demand.

How to organize suites

Smoke	Five to ten tests. The flows that mean the product is up at all. Run often.
Regression	The broad suite. Run daily, or before a release.

Release candidate	Everything, including slow edge cases. Run on demand, when it matters.

Keep the smoke suite genuinely small. Its value is that a failure means something, and that depends on people trusting it enough to look.

Checkpoint

You are done when your orchestration shows one completed run in its history, with a duration, and you can open one of its tests and see that run's steps.

Try this

Break one of the tests using the method from Exercise 5, then run the orchestration again. Watch how a suite reports a partial failure, and notice you can still get from the suite result down to the exact red step.

What you learned

An orchestration turns individual tests into a signal about the application. The work is in choosing which tests belong together and who will act on the result.

Next: Exercise 7 — Drive Studio from your own AI tool.

Where to go next

- Exercise 7 - Drive Studio From Your Own AI Tool — drive all of this from your editor
- Orchestrations: Running a Suite as One Job — the feature, in full
- Keeping a Suite Green: Flake, Tiers and Who Owns Red — making a scheduled suite worth reading