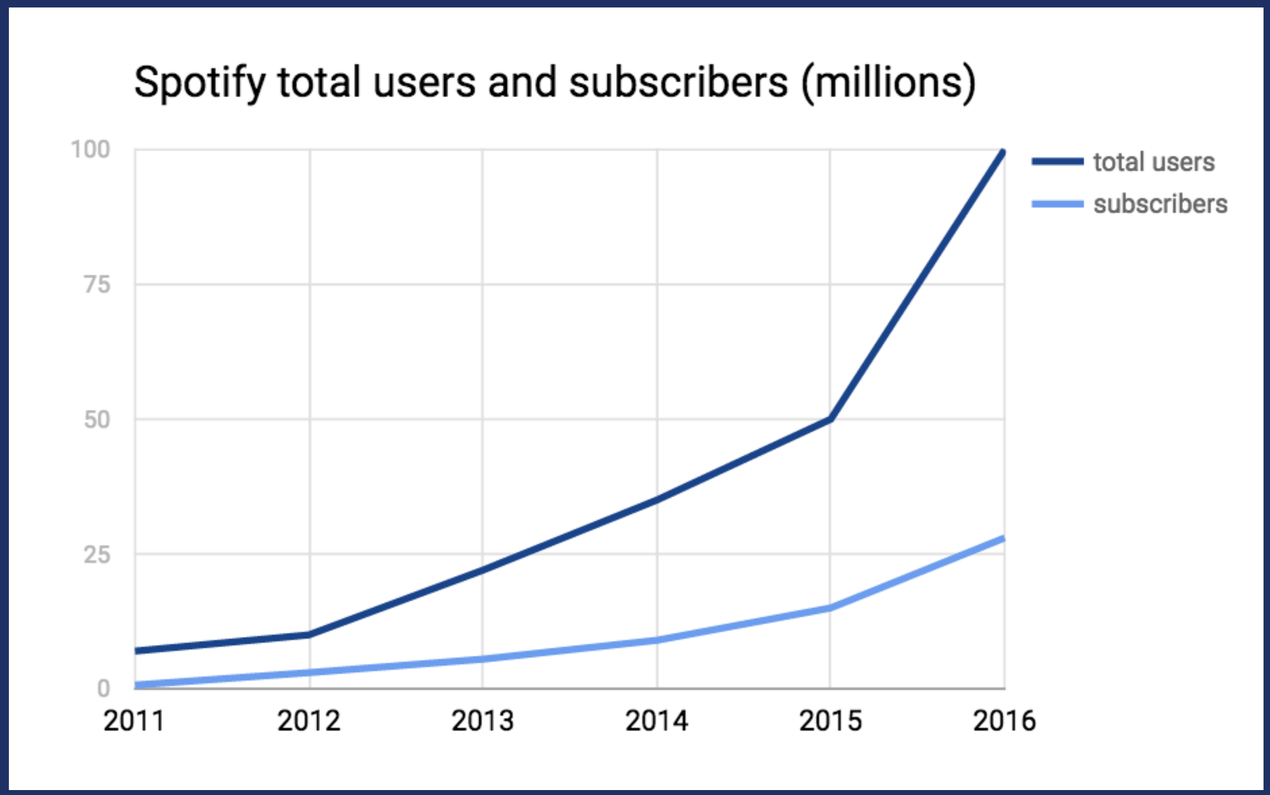
Reach Production Faster with Containers in Testing

David Xia

Spotify®

Spotify's Scale



- 150+ people in infrastructure
- Thousands of hosts,
  2000+ running containers
- 1500+ deploys in past month,
  majority were containers

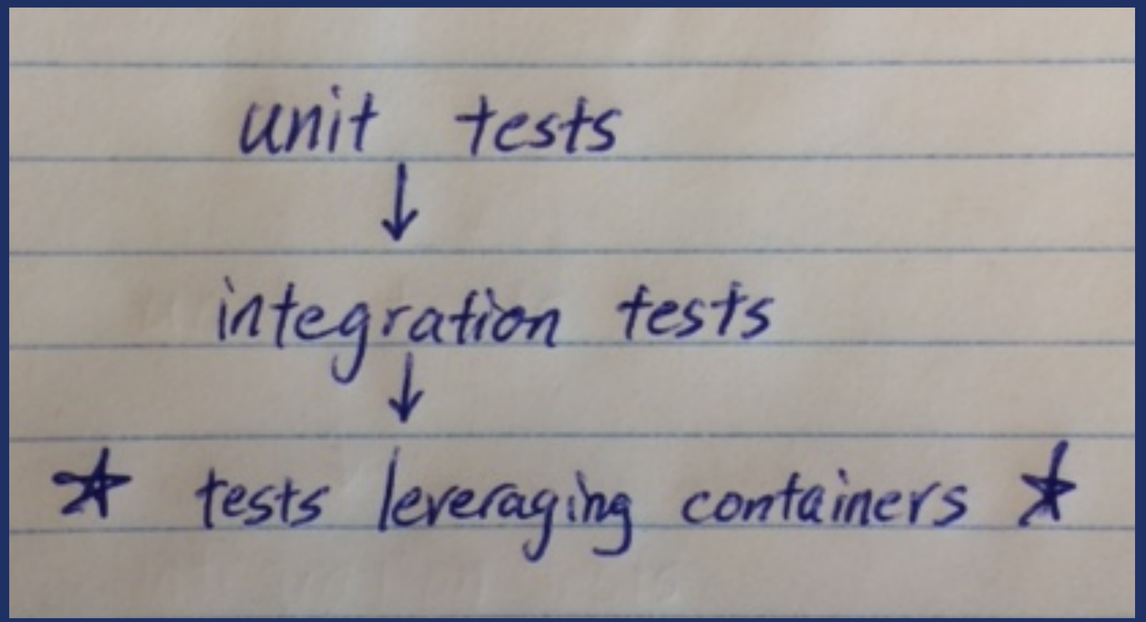
About David Xia

- Work on deployment infrastructure

- Work on open-source Docker orchestration tool Helios

Prerequisites for This Talk

- You're familiar with containers
- You like tests
- That's it!



Prerequisites for This Talk

You don’t need to use containers in production for this talk to be useful!

Three Problems, Three Ways to Solve with Containers

How can I enable developers to:

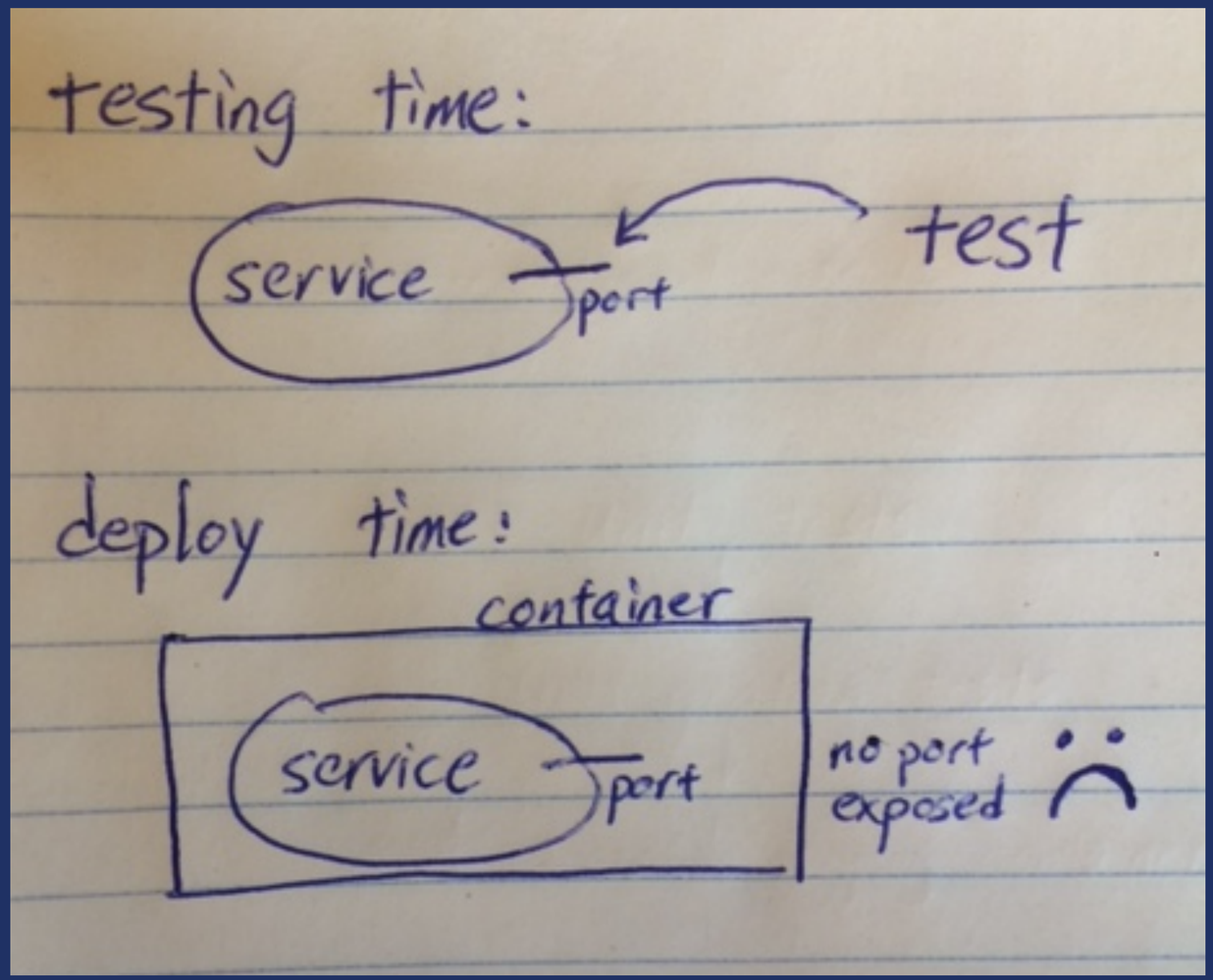
- catch container misconfiguration in tests?
- easily install and start non-trivial test dependencies?
- make their tests isolated and reproducible?

Problem 1: Container Misconfiguration

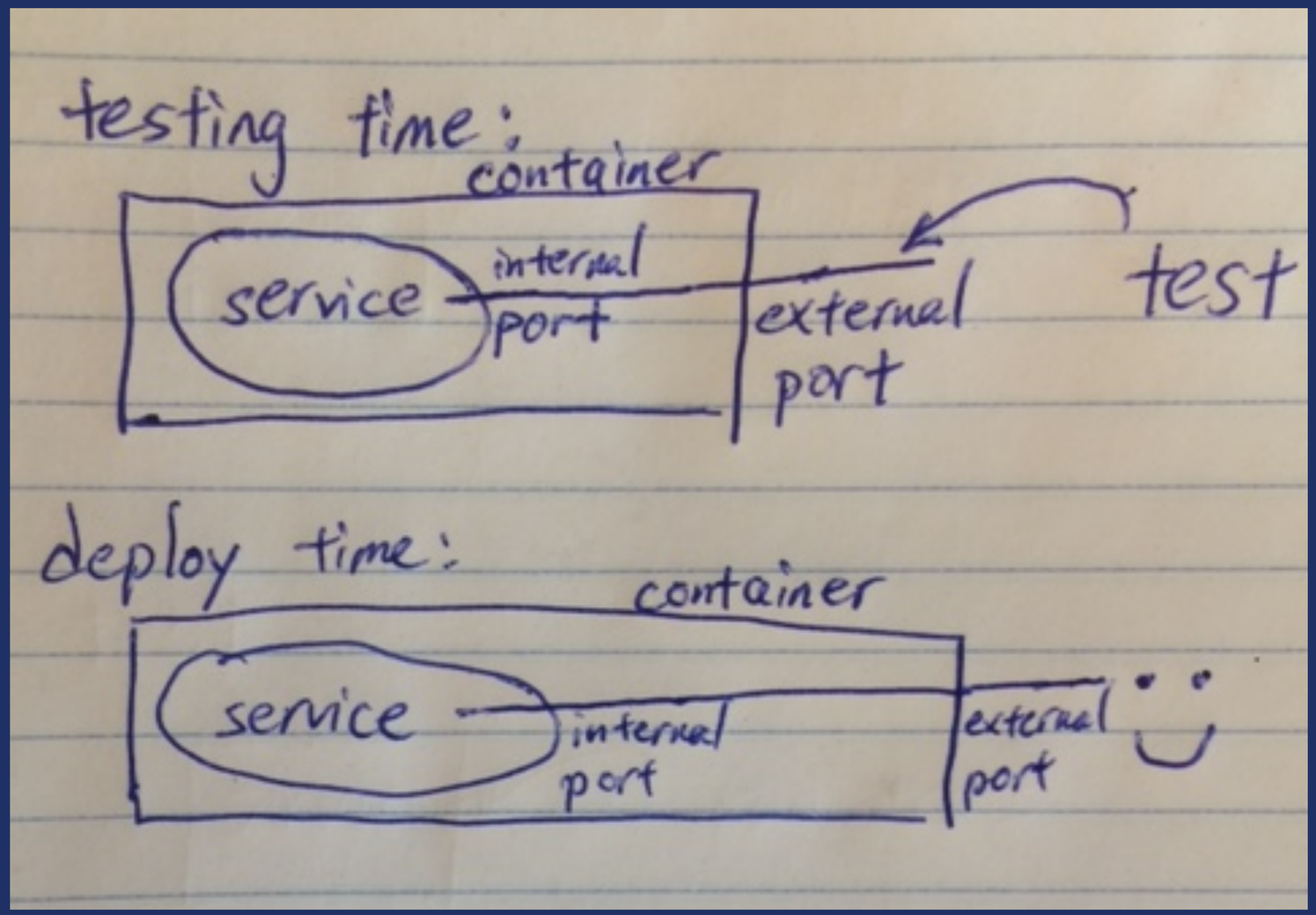
“Why did my service pass integration tests but fail when I deployed it as a container?”

- Sad Developer 1

Problem 1: Container Misconfiguration



Solution 1: Container Misconfiguration

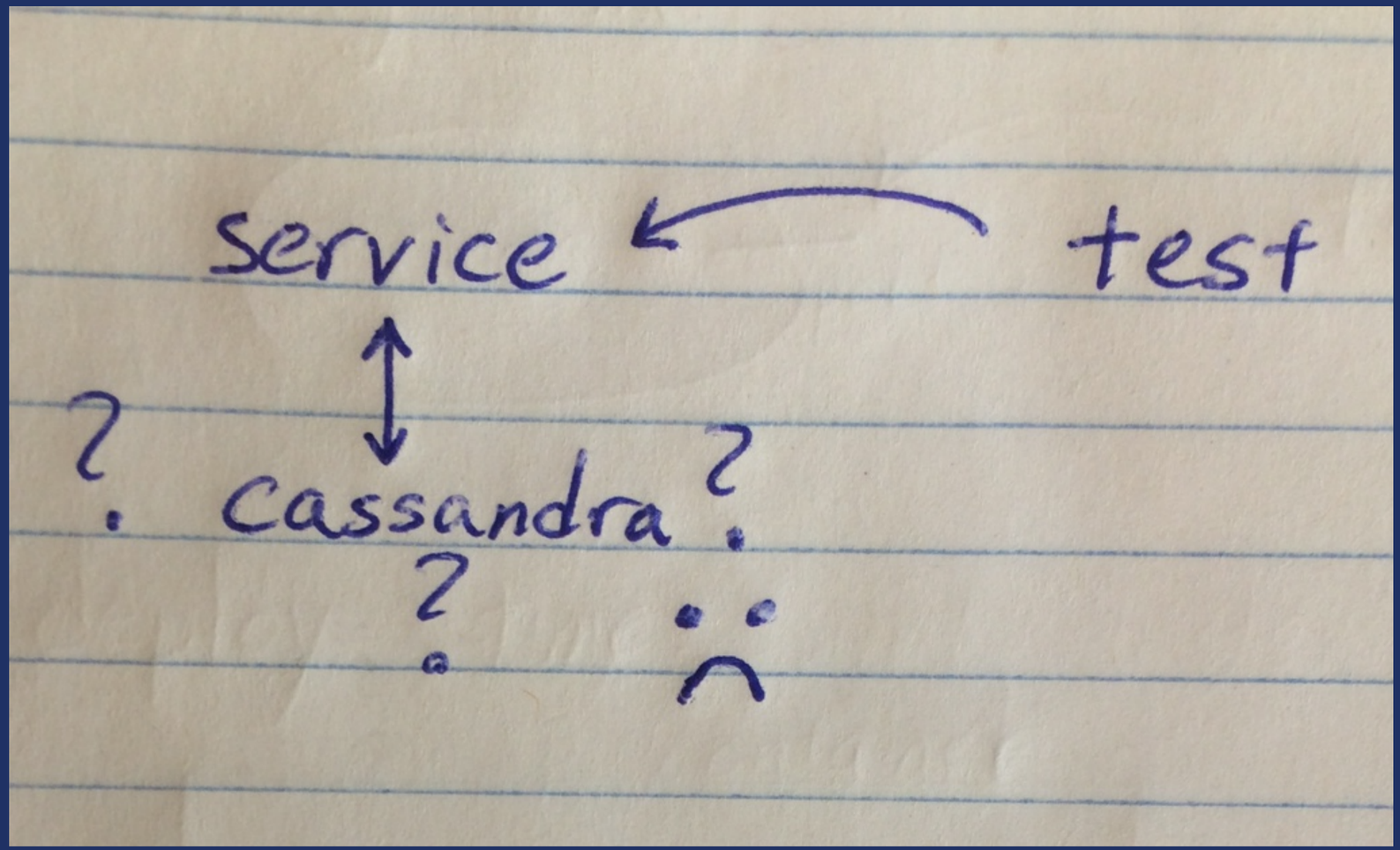


Problem 2: Non-trivial dependencies

“I want to run my project's integration tests locally. The tests need a local Cassandra/other DB. How do I set everything up?”

- Sad Developer 2

Problem 2: Non-trivial dependencies

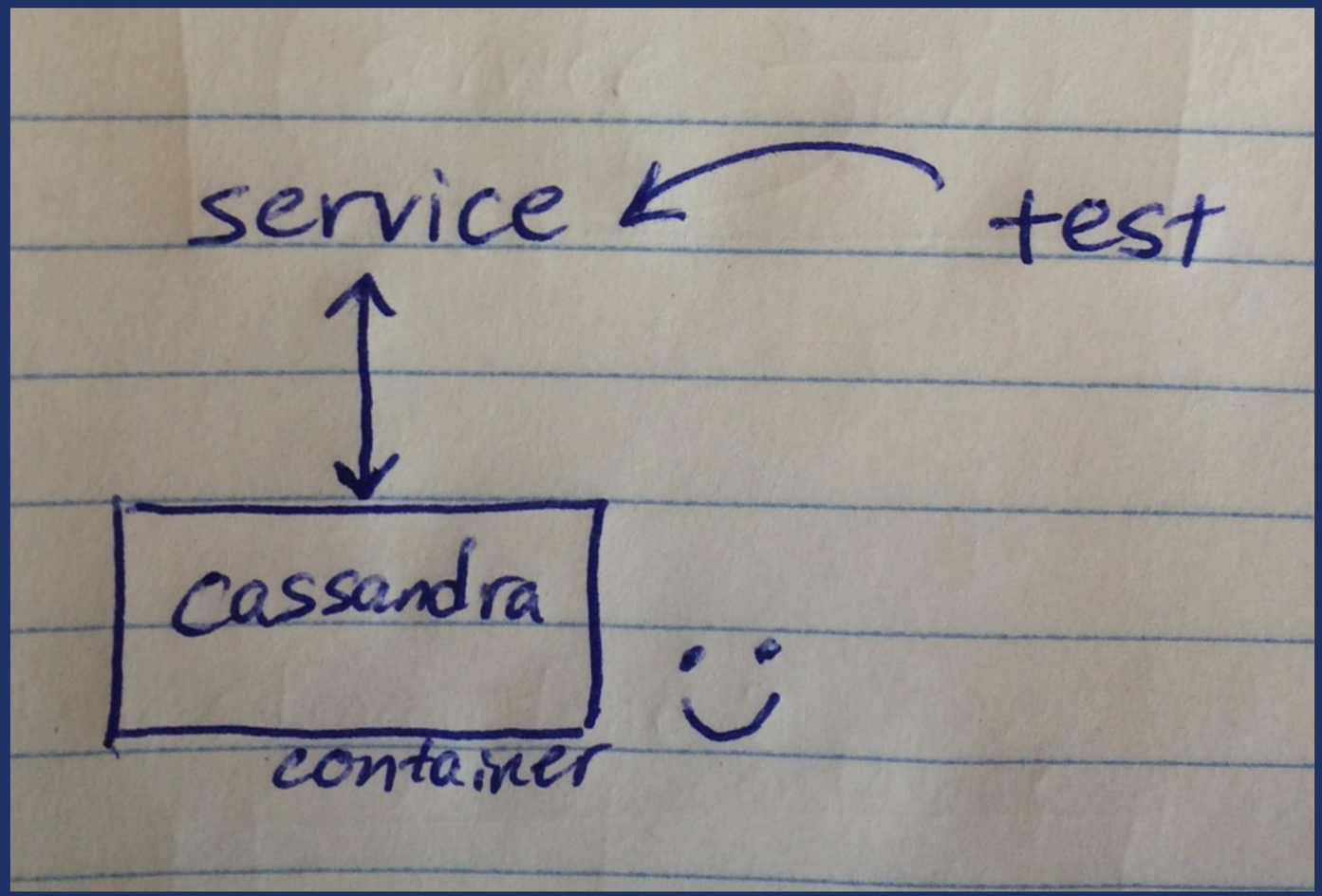


Solution 2: Non-trivial dependencies

```
docker run --name foo -d cassandra
```

- Happy Developer 2

Solution 2: Non-trivial dependencies



Problem 3: Reproducible Tests

“How can I easily restore my test dependencies to a clean state?”

- Sad Developer 3

Solution 3: Reproducible Tests

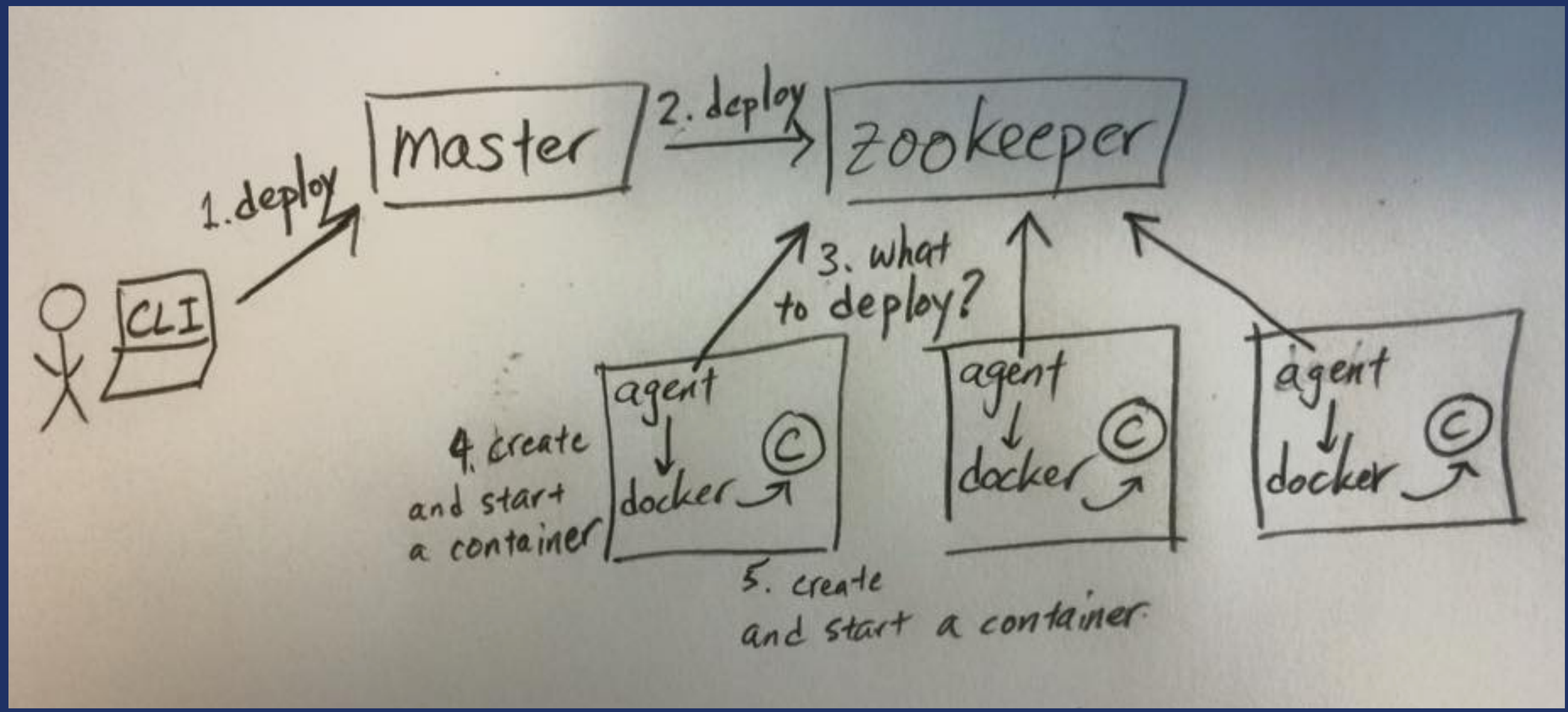
```
docker stop <container ID>
docker run --name foo -d cassandra
```

- Happy Developer 3

Key Takeaways - Using Containers in Tests Can Help You:

- Test more of the stack in an env resembling production
- Easily start real dependencies
- Ensure tests are reproducible and isolated

Helios in a Nutshell



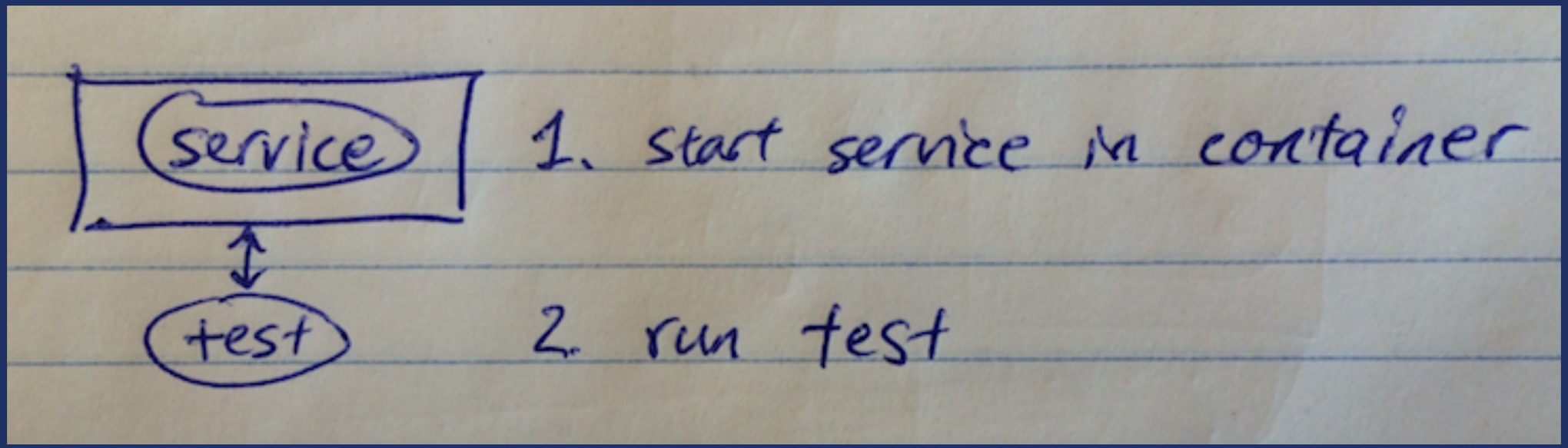
How helios-testing was born

- multitenancy
- docker
- container orchestration (helios)
- container testing framework (helios-testing)

Spotify®

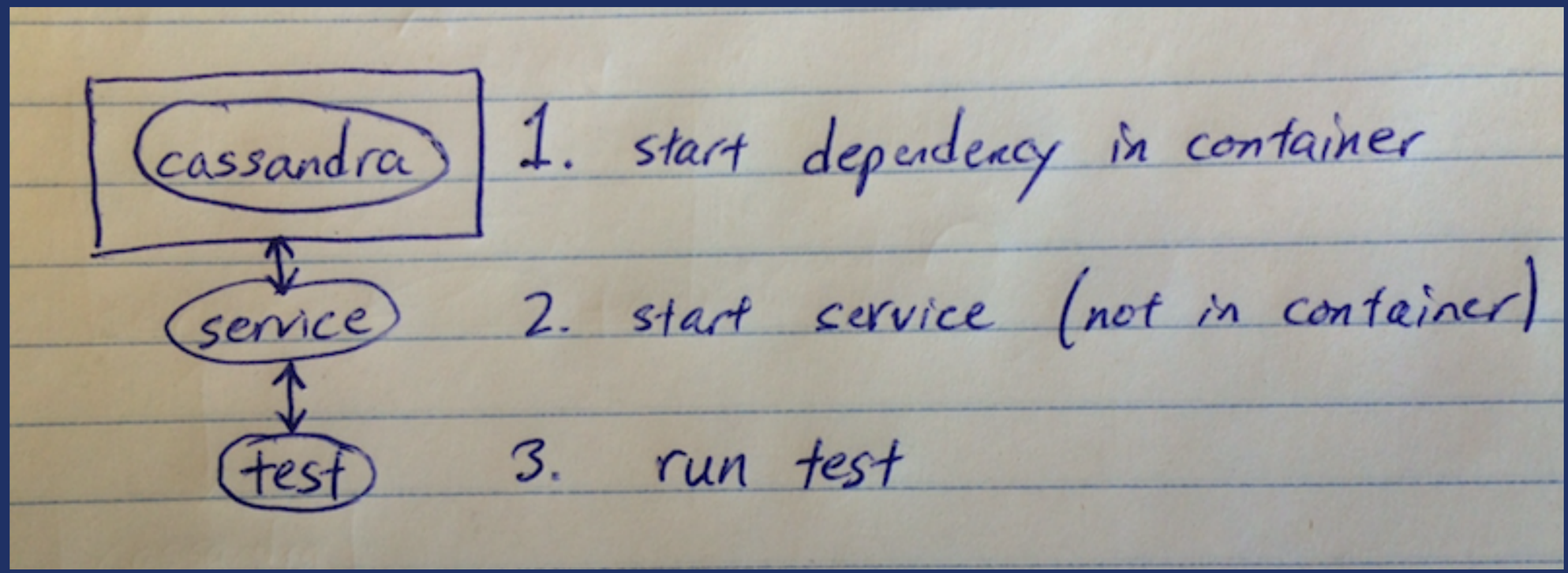
What does helios-testing do?

Let's you write code to start and stop containers.



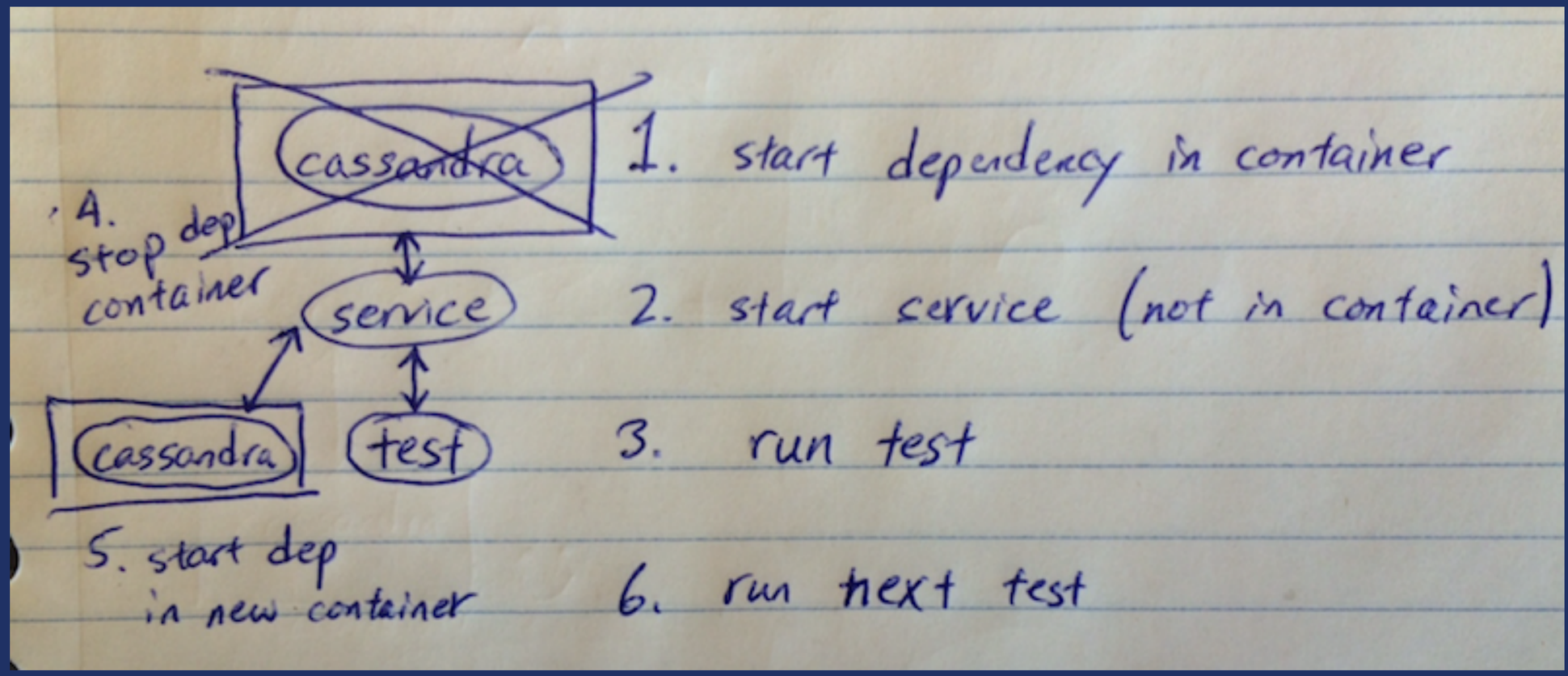
What does helios-testing do?

Let’s you write code to start and stop containers.

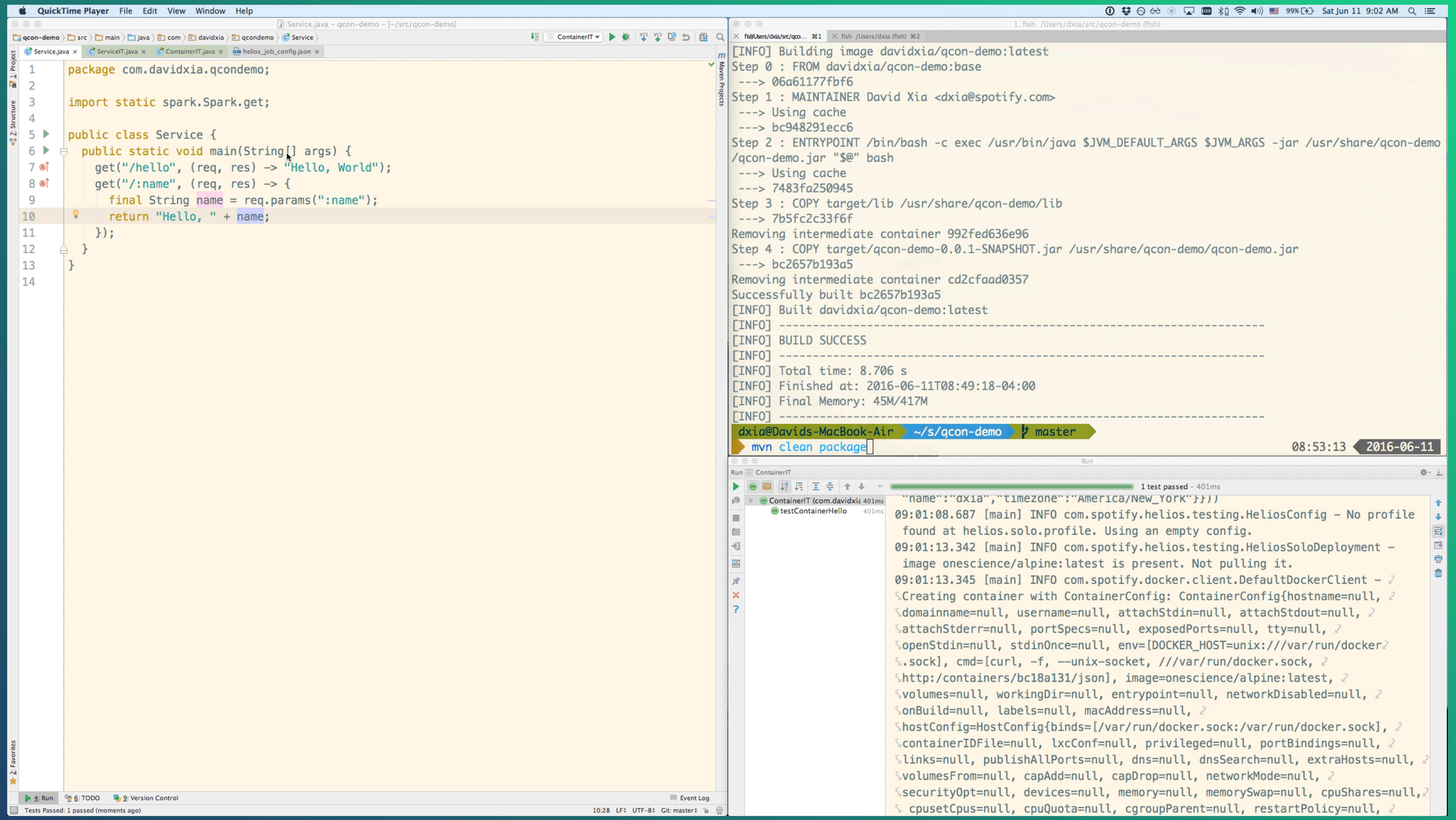


What does helios-testing do?

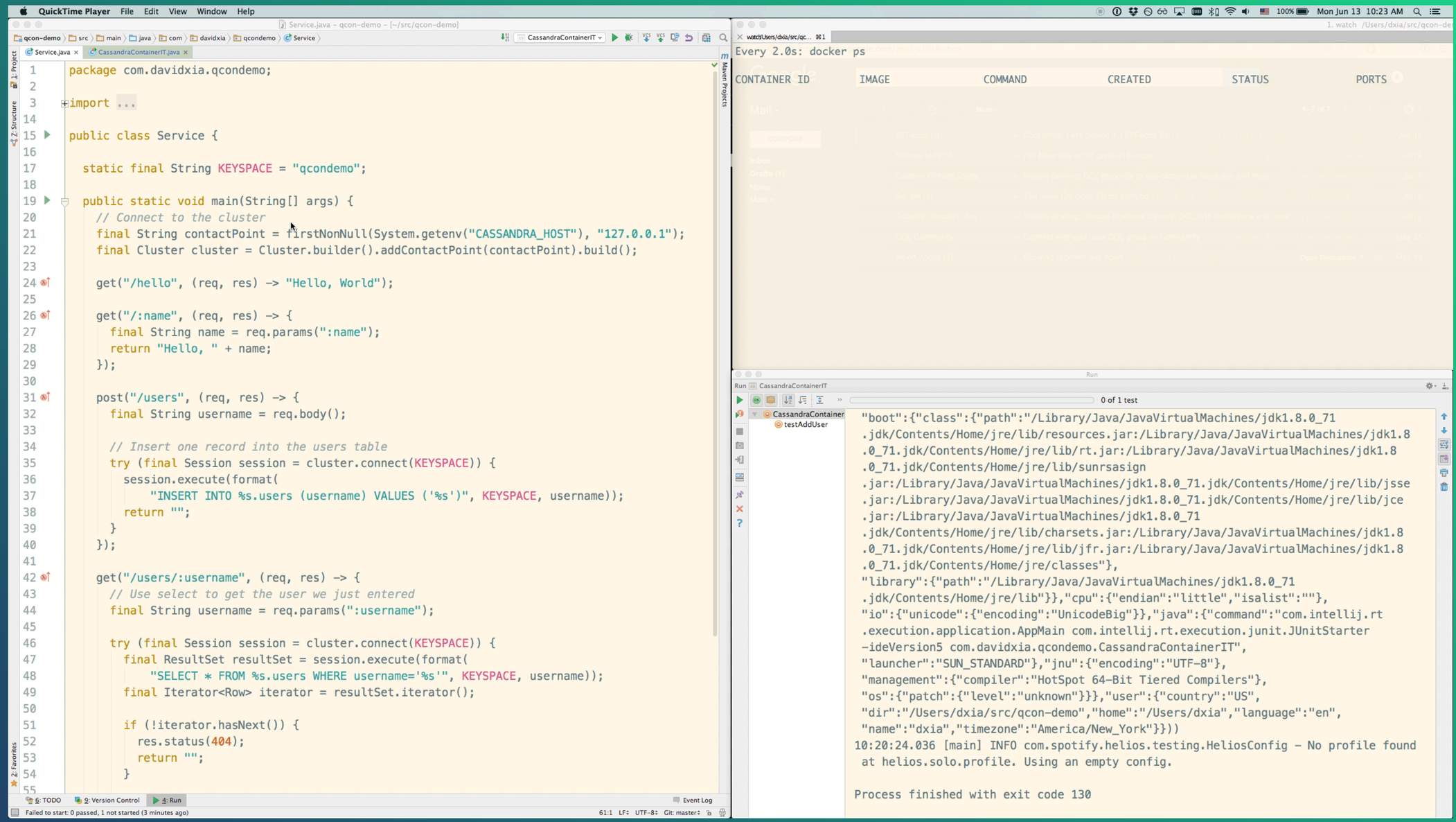
Let's you write code to start and stop containers.



Demo! Solution 1: Container Configuration



Demo! Problem 2



Successes

“Testing with a Cassandra container is the closest I can get to testing against Cassandra in reality.”

Successes

“I especially like the fact that I can test my image in its final state and be confident that it will work in production.”

Successes

“Using helios-testing to run datastores in containers has made the tests portable and setup free (by setup I mean no manual installation of the datastore on the test machine or locally).”

Lessons Learned

Make sure your testing framework and infrastructure are fast and reliable.

Lessons Learned

Make framework's interface and implementation as simple as possible.

Lessons Learned

Provide great test examples.

Key Takeaways - Using Containers in Tests Can Help You:

- Test more of the stack in an env resembling production
- Easily start real dependencies
- Ensure tests are reproducible and isolated

When Not to Use Containers in Tests

- Don’t test functionality unrelated to containers that you can easily test separately
- When your container-based tests overlaps a lot with regular integration tests

Acknowledgements

Rohan Singh

Matt Brown

Staffan Gimåker

Mats Linander

Nic Cope

Spotify®

Q&A

@davidxia_

github.com/davidxia

github.com/spotify/helios

github.com/davidxia/qcon-demo

Demo videos: example 1 and 2

helios-testing