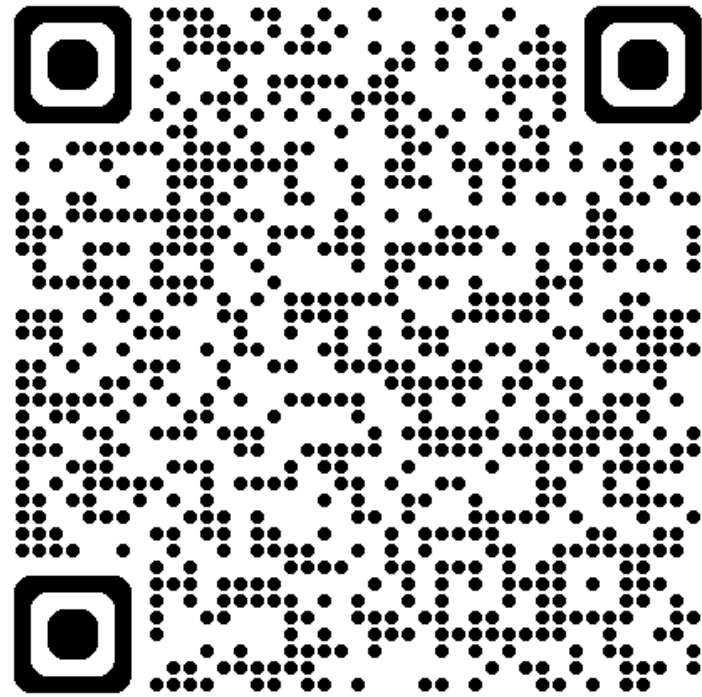


No More SQLite



How to Write Tests with EF Core Using Testcontainers

Daniel Ward



LEAN
TECHNIQUES

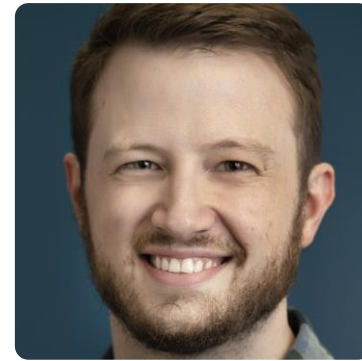





SAN
ANTONIO
NET



AUSTIN
NET

Me



- Software developer, consultant
- Microsoft .NET MVP
- Co-organizer of the San Antonio/Austin .NET User Group
-  daninacan.com
-  [@danielwarddev](https://twitter.com/danielwarddev)
-  [daniel-ward-dev](https://www.linkedin.com/in/daniel-ward-dev)



Works on my machine



No man's land

You're good!



No man's land

Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD
Failed to load resource: net::ERR_BLOCKED_BY_AD

False positives,
false negatives



If you use a different SQL engine, you are testing a different system!

Why does
that matter?

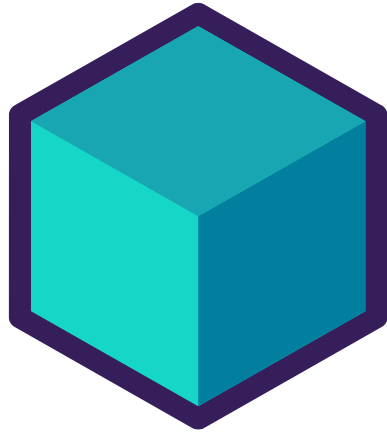


Traditional Options

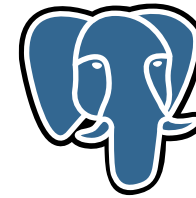
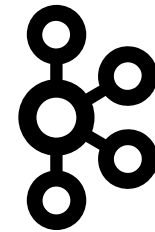
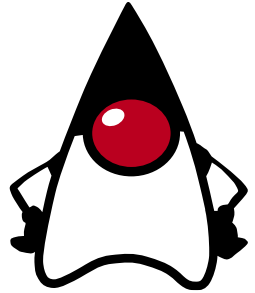
	Pros	Cons
EF in-memory provider	Easy to set up	Very unlike real database
SQLite provider	Easy to set up	Unlike real database
Real database	Same engine as production	Lots more work to maintain
Docker container	Same engine as production	More work to maintain



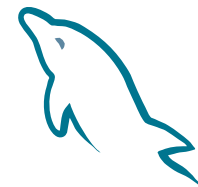
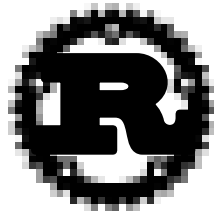
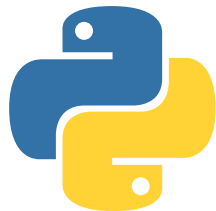
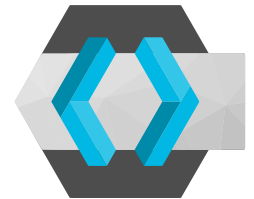
Take the existing Docker workflow, but automate it and add tooling



Testcontainers



Lightweight, throwaway containers





Set Up



Run Tests

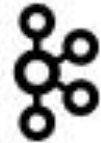


Clean Up



Start Containers

Destroy Containers

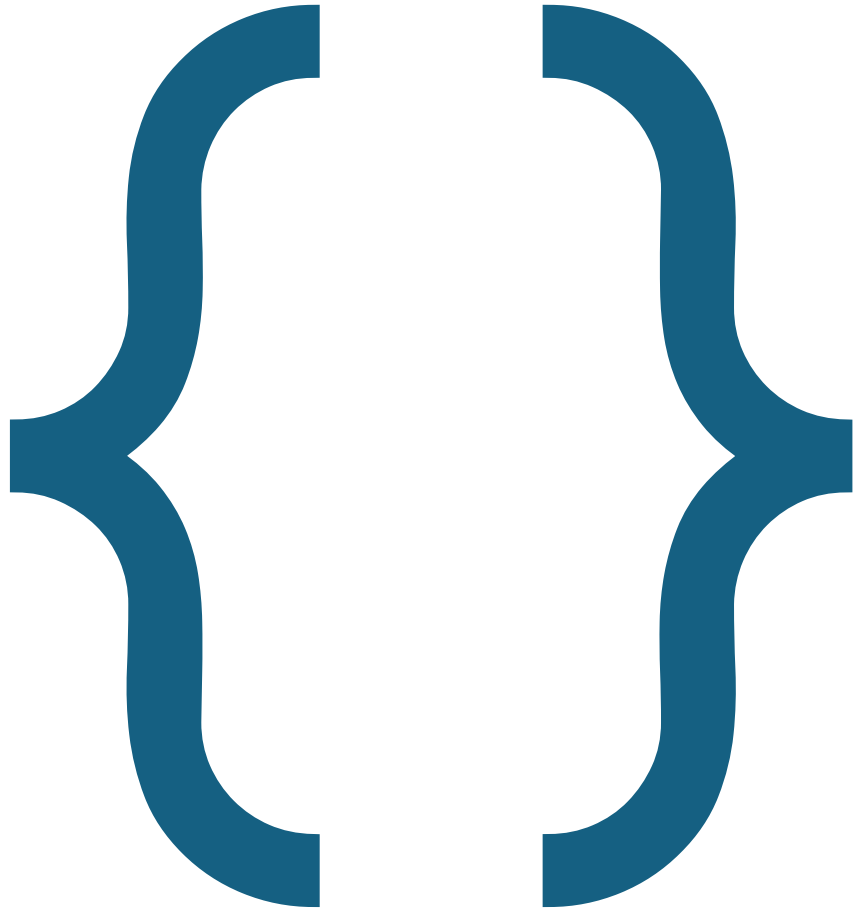




```
private readonly PostgreSQLContainer _container = new PostgreSQLBuilder().Build();
```

```
    await _container.StartAsync();
```

```
    await _container.DisposeAsync();
```



Coding demo

- Testing packages
 - [Autofixture](#)
 - [FluentAssertions](#)
 - [Testcontainers](#)
 - [Respawn](#)



Honorable mentions




- [Resource reuse](#)
- Podman compatibility

Recap



- Traditional options for integration testing
- Testcontainers
- Coding demo
- Respawn

Thank you!

-  daninacan.com
-  [@danielwarddev](https://twitter.com/danielwarddev)
-  [daniel-ward-dev](https://www.linkedin.com/in/daniel-ward-dev)

