Boston Office:
1 Harris Street
Unit 7
Newburyport, MA 01950

Little Rock Office:
401 Main Street, Suite 203
North Little Rock, AR 72114

Problem Statement

Darvis is a company which developed a complex application composed of many different microservices working in concert. The microservices are tightly coupled over a message queue framework. Each microservice has its own unit tests, but during the production rollout of the system, Darvis realized this testing was insufficient to guarantee a seamless release cycle. Significant manual testing is required after each deployment. The result is a large amount of overhead for developers and QA during releases. Changes to the individual microservices often lead to breakages of the overall application.

Darvis engaged with Elyxor to design and implement an automated integration testing solution. The goal was to establish a set of 'overall' tests which covered the primary use cases of the application, exercising all the individual microservices together as they would run in production. Furthermore, Darvis developers needed to be able to write additional tests without having to consider the configuration and execution of the test environment.

The automated testing needed to be executed as part of the CI/CD pipeline so that issues or breaking changes would be identified as early in the process as possible. Builds were required to fail if a change to one microservice broke the behavior of another to prevent any deployment of the application to production. The desired outcome was less churn in the development and QA cycle and faster and more frequent releases.

Boston Office:
1 Harris Street
Unit 7
Newburyport, MA 01950

Little Rock Office:
401 Main Street, Suite 203
North Little Rock, AR 72114

Elyxor Solution

Elyxor Test Automation Platform (ETAP)

Elyxor developed **ETAP** as an accelerator to improve quality and speed of implementation of test automation for client projects. **ETAP** brings together test services and utilities that we have found to be reliable and in wide use. The framework itself is versioned and allows for custom adapters and features as required to fill gaps where existing tools do not exist.

Key features include the following:

- Support for test suites written in Java, Python, Cypress, and .NET
- Support for popular test engines such as TestNG, Cucumber, and PyTest
- Integration to test-management and defect-management tools (services use API calls to determine tests to run and for publishing execution results)
- Improved capability for DevOps adoption through integration with CI tools and versioning systems
- Pre-built libraries for test suite configuration and test data
- Improved configuration and isolation of the test environment through Testcontainers, a Java library which executes Docker

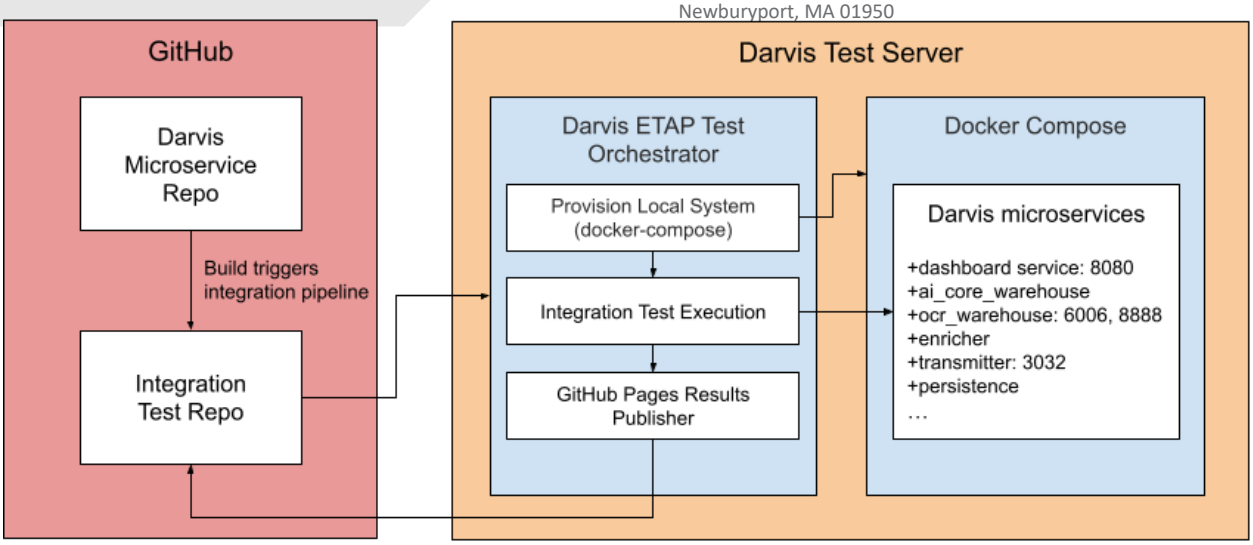
Darvis Framework

Elyxor delivered a test framework for Darvis using **ETAP**:

- Created a Git project to store integration tests and Docker environment configuration
- Developed a new CI/CD pipeline to execute tests on a Darvis test server
- Leveraged existing Darvis Docker Compose file plus a dynamically generated override file to pull in specific versions of published Docker containers
- Configured **ETAP** to launch Darvis application within containers on the test server, run tests, and generate a test report
- Published CI/CD pipeline test results to GitHub Pages
- Tested incoming new code against existing (unchanged) microservices to screen for breaking changes
- Provided early identification of breaking changes, allowing for correction prior to merging into master branch

Boston Office:
1 Harris Street
Unit 7
Newburyport, MA 01950

Little Rock Office:
401 Main Street, Suite 203
North Little Rock, AR 72114



Results

ETAP, along with Elyxor's implementation of Continuous Integration / Continuous Delivery, offered Darvis the following benefits:

- Reduced risk related to code changes/new features
- Elimination of manual activities
- Increased stability of automated deployments
- Reduction of time spent in QA
- Faster time to delivery/market
- Return on investment
    - Increased velocity of new revenue generating features
    - Increased engineering productivity
    - Increased customer satisfaction with improved reliability

Tech Stack

- Elyxor Test Automation Platform
- Java
- Gradle
- GitHub
- GitHub Actions
- GitHub Pages
- Cypress
- Docker
- Docker-Compose