

Using Containers to Isolate Remote Code Execution for an Online Development Environment

Joseph Fazzino
24026478

Supervisor: Dr. Hong Wei

29th April 2019

Contents

0.1	Abstract	3
0.2	Acknowledgements	4
0.3	Glossary of Terms and Abbreviations	5
1	Introduction	6
2	Problem Articulation & Technical Specification	7
2.1	Context	7
2.2	Problem Statement	7
2.3	Technical Specification	8
2.3.1	Writing and Executing Code	8
2.3.2	Personal Environments	9
2.3.3	Local Tooling Replacement	10
2.3.4	Encourage Exploration into Development	10
2.4	Stakeholders	11
2.5	Constraints	12
2.6	Assumptions	13
3	Literature Review	14
4	The Solution Approach	15
5	Implementation	16

6	Testing: Verification and Validation	17
7	Discussion: Contribution and Reflection	18
8	Social, Legal, Health and Safety and Ethical Issues	19
9	Conclusion and Future Improvements	20

0.1 Abstract

250 - 300 words

outline aims, methods, implementation, achievements, and conclusions

0.2 Acknowledgements

I'd like to acknowledge Dr. Hong Wei for being my project supervisor. Dan Justin and Dr. Martine Magnan for their continued support through the early stages of my career. Suhail Parmar for contributing to my caffeine levels and providing help with Docker and UNIX. Dan Davis and Max Denning for helping me brainstorm the idea for the project and enduring me talking about frontend for the past year. And finally, my parents, Paul and Joanna Fazzino for being unwaveringly supportive and great role models.

0.3 Glossary of Terms and Abbreviations

API - Application Platform Interface

Chapter 1

Introduction

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory which states that this has already happened.



Figure 1.1: The Universe

Chapter 2

Problem Articulation & Technical Specification

2.1 Context

As computers have become more pervasive, coding has become a skill that has graduated past being something that only people who work in laboratories need to concern themselves with, to a skill that has become highly desirable commercially and is starting to be taught in the regular curriculum to children studying at a primary level education[1]. This new demand for beginner friendly coding tools lends itself nicely to the promise of an online based environment where people can get started with basic coding concepts without having to trawl through documentation and technical detail about how to get running with one of the popular languages/tools available. This has led to an explosion of popularity for web applications such as `codecademy.com` which offer pre-made, executable exercises for a number of languages. A similar platform `repl.it` offers a more open and free-form experience and attempts to recreate the environment a developer may have on their machine through the web browser along with online compilation.

2.2 Problem Statement

A common pattern with the current platforms that exist is that they provide a strict sandbox within the confines of a predetermined configuration that the user selects, for example, in `codecademy` and `repl.it` you're stuck in the environment you pick when you start desired tool. An argument can be made that this makes a new developers life easier as they don't have to consider the more nuanced parts of the file system or learn any sort of terminal commands. However, it seems as though there would be value in a system that can provide both the ease of use that current existing solutions offer and also the freedom to explore a full environment with an array of tools preconfigured that encourage exploration without compromising the security and integrity of the underlying system.

2.3 Technical Specification

Based on the problem statement the potential scope for the project is very broad, there are companies and teams of developers that have the sole goal of making sure their online environments are providing users with as smooth an experience as they would expect if they had installed the tools locally.

This project will focus on the essential functionality required to behave as an online development environment while supporting a good variety of languages and offering a space which encourages exploration into different coding concepts.

With the above in mind the enumerated objectives of this project are:

1. Create a platform where users can write/execute code
2. Give every user their own personal environment
3. Eliminate the need for locally installed tooling
4. Provide a system that encourages exploration into the world of development

2.3.1 Writing and Executing Code

As an essential requirement for the development experience, the ability to edit and execute code is crucial to satisfy the overarching objective of creating an online environment. The execution of code presents a significant technical challenge however as the only code execution that can be done remotely is on a web browser which must be able to execute HTML, CSS and JavaScript. Mobile applications developed for iOS and Android are not capable of executing code.

Functional Requirements

- Code will be able to be typed using the platform
- Code will be able to be saved

- Code will be able to be read from the platform
- Code will be able to be executed

Non-Functional Requirements

- A good variety of languages will be supported
- The basic features of a code editor will be available (i.e. syntax highlighting)
- Code that is executing will not stall the platform

2.3.2 Personal Environments

The need for the space that the user occupies to feel personal is a vital element to local development environment and therefore must be well implemented for an online equivalent.

Functional Requirements

- A personal environment will be allocated to every user

Non-Functional Requirements

- The personal environments will be isolated from the rest of the system
- The personal environments will be isolated from each other
- The personal environments will perform well and be responsive to user input
- If a personal environment fails then it will be restarted

2.3.3 Local Tooling Replacement

Tooling has been through some big changes both in web browsers and locally. Web browsers have got to the point where they are so powerful that some of the most popular desktop software is being powered by them[2]. It is important to provide tools that will help those new to development, while also offering experience in tools that are of a high quality.

Functional Requirements

- High quality tools will be available to the user
- Industry standard tools will be available to the user
- The system will eliminate the need for local tooling

Non-Functional Requirements

- Popular tools will be researched and considered before being added to the system
- Tools will be standardised across the system
- Tools will be customisable to the users needs
- Tools will behave in a responsive manner

2.3.4 Encourage Exploration into Development

Lowering the barrier to entry through the requirements stated above will inherently make it easier to explore development but more steps can be taken in order to engage users with the system such as allowing them to create short coding exercises that can be shared with friends or on social media.

Functional Requirements

- Implement exercises for users to do

- Allow creation of exercises by users

Non-Functional Requirements

- Allow any exercise to be shared
- Assign difficulty level to exercises
- Provide an open area for the user to explore their personal environment

2.4 Stakeholders

This project has a number of relevant stakeholders with various degrees of interest in the outcomes. All of them will be considered during the construction of the system.

The Developer - Joseph Fazzino

The developer of the system is responsible for making 100% of the the technical decisions and is responsible for delivering a fully functioning system adhering to the technical specification found in Section 2.3 of this report.

Project Supervisor - Dr. Hong Wei

The supervisor of this project is overseeing the development and design process that is being undertaken.

They provide guidance when it comes to essential functionality and ways that technical requirements can be implemented.

User - Beginner Level Developer

Those new to development will not have experience with the terminology and syntax that exists in programming and wider computer science. They

may have an understanding of basic coding concepts taught to them during formal education.

The beginner user should be able to use the system in order to become more familiar with generic programming concepts. The exercises available through the system will likely be the area they spend the most time.

User - Intermediate Level Developer

A user more familiar with the general work flow of a developer will be able to understand certain levels of nuance of how a system might be implemented and consider how they may solve certain problems.

This kind of user would benefit more from the ability to have a playground to explore the system in so they can understand the functionality that it provides and maybe try to explore the extent to which it works.

User - Experienced Level Developer

This user will have successfully developed systems with a high level of complexity and will most likely have specialised knowledge in a certain domain/environment.

This type of developer will be difficult to convince the benefits of an online working environment when they undoubtedly have a solution that works well for them locally.

2.5 Constraints

Some constraints on the development of the project exist.

- Permanent deployment - as the system is likely to be complex, deploying it will be costly and time consuming. Test deployment will be done to experiment with configuration settings in the system but a permanent live deployment will not be.
- Computer resource availability - the system will be constrained performance wise by the resources available during development meaning that any stress tests are not representative of a deployed system

- Significant testing base - as the system will not be deployed it will be difficult to adequately test the system in the manner which it would be used by end user. A different method of testing will have to be explored.

2.6 Assumptions

A number of assumptions must be made to reasonably meet the technical requirements.

- The users will have a reliable internet connection
- The users will have the necessary software/hardware configuration in order to access the system (e.g. a modern web browser)

Chapter 3

Literature Review

Chapter 4

The Solution Approach

Chapter 5

Implementation

Chapter 6

Testing: Verification and Validation

Chapter 7

Discussion: Contribution and Reflection

Chapter 8

Social, Legal, Health and Safety and Ethical Issues

Chapter 9

Conclusion and Future Improvements

Bibliography

- [1] S. Chalmers, “Why schools in england are teaching 5 year olds how to code,” Oct. 2014. [Online]. Available: <https://www.bloomberg.com/news/2014-10-15/why-schools-in-england-are-teaching-5-year-olds-how-to-code.html> (visited on 04/01/2019).
- [2] T. Claburn, “Carlo has a head for apps and a body (tag) for rendering: Google takes on electron with js desktop app toolset,” Nov. 2018. [Online]. Available: https://www.theregister.co.uk/2018/11/02/carlo_chromium/ (visited on 04/02/2019).