

[DRAFT] Towards scalable CI/CD tools on low-end devices.

Pieter De Clercq

2019-02-28

1 Introduction

Building, Testing and Deployment are three important steps of the software development life cycle. A typical development process iterates through all of these steps a lot of times. Various tools already exist as of today to simplify these tedious, yet repetitive tasks, collectively referred to as “Continuous Integration/Continuous Delivery”, or CI/CD for short. These tools already vastly reduce the amount of time it takes for a developer to ensure working, good quality code, but as with all things, there exist some drawbacks.

As of today, the most popular options for running a buildserver are probably Jenkins and Travis-CI. A commonly used, open-source, tool for analysing code quality is Sonarqube. While these tools serve their purpose, they all share two major issues. The first, and most important one, is scalability. As of today, it is not possible to quickly scale-up Jenkins/Sonarqube instances to load-balance projects upon experiencing high loads. Another important issue is the fact that these CI/CD-tools are not designed to run on low-end, cheap, devices, such as Raspberry Pi's. Summarised, the only feasible solution to this problem is vertical scaling: increase the capacity of 1 single server.

2 Goal

The goal of this research consists of multiple objectives. First of all, study the already existing open-source CI/CD projects. Jenkins/Travis-CI were mentioned, but more exist, eg. Gitlab CI. Afterwards, try to extend one or more of these projects with custom-made plugins to allow horizontal scalability to a given extent. This plugins should use an algorithm (perhaps based on Machine Learning/Artificial Intelligence?) to decide whether it's required to spin up an extra server to run the build on, or use an already existing server in the pool.

Moving further away from the already existing solutions, study the advantages and disadvantages of multiple container platforms, such as Docker, CRI-O, rkt. Eventually the final objective will be to bundle the previously gathered experiences into creating a plug and play network of multiple low-end devices that can be used to build, test, analyse and deploy projects. The builds are ran in containerized environments to ensure compatibility.