Round Robin service with Docker and AWS

David Alejandro Vasquez Carreño
Ingeniería de sistemas
Escuela Colombiana de Ingeniería Julio Garavito
Bogotá, Colombia
david.vasquez@mail.escuelaing.edu.co

***Abstract*—Using the powerful tool of docker and java, we have made an easy implementation of a log service exposed to the web using AWS, where 5 docker containers are stored and running locally, and listening in ports mapped to the real ports of an EC2 instance of AWS.**

This little implementation allow to show how easy and extensible docker is, and how it can be deployed into a virtual machine that has only few ports listening to the internet.

***Index Terms*—Http, javascript, Server, Docker, AWS, EC2**

I. Introduction

Virtualization is a mechanism that allow us to create flexible infrastructures into a single machine, without the need of re buying multiple physical machines that can make cost go higher and does not exploit all the potential of existing hardware.

Horizontal escalation is a common thing for create bigger and bigger systems that need multiple services running into a internal network, like accounting, web services, databases or custom services that need an special port into the machine for easier configuration. This kind of escalation is not always the best option in small companies that have not exploited all their current hardware, like servers or dedicated machines.

We'll see how virtualization allow us to run multiple instances of the exact same application, with the same port configuration, but with docker in the middle it is not a problem, because docker guarantee us to isolate all containers one from another.

II. The need of virtualization

With so many machines into a internal architecture, it is smart to exploit every part of the hardware, but with so many king of applications running into this machine, it can be tricky and harder to handle so many connections, configurations, and the mis configuratiioon of the machine is every time bigger.

The image below1 shows the same machine can store multiple instances of operative system, like linux of windows, all those running with their own configuration and listening

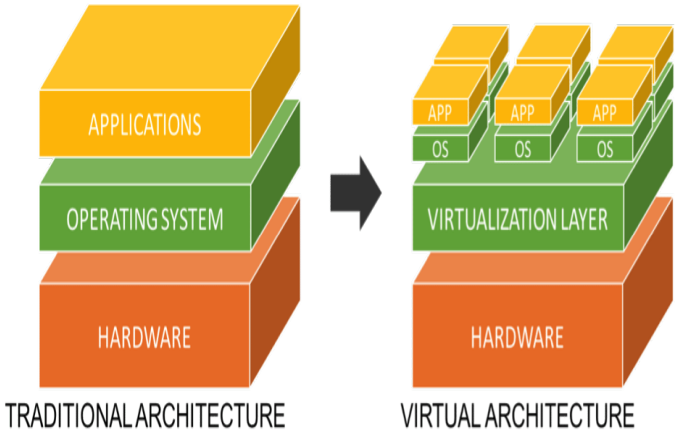

Fig. 1. Web servers statistics

ports, that are isolated from the real machine, which allow us to have the concept of multiple real machines into a network, but is instead a single machine [1].

Unfortunately, running multiple Os into the same machine can lead to performance issues, because it needs so many resources for those machines, like memory, CPU, and disks partitions for each one, and all this is just not sustainable.

III. What is a lightweight container?

Lightweight containers like docker are little instances of virtual machines running just with the minimum requirements for being functional, like offering custom ports and isolation form the host machine, but without the hard penalty if running multiple instances in the same hardware [2].

Docker2 is a easy and flexible container tool for running multiple little virtual environments into a single machine, and all this based in images that the developer create for later use into development, like images for pre built database configuration running into a isolated port.

IV. Solution

The goal of this exercise is to show the power of this kind of tool running is a single machine. We are going to create 5

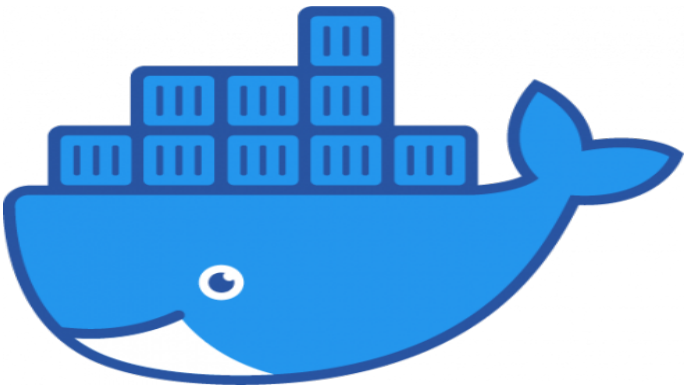

Fig. 2. Web server communication

containers, the first is going to be a mongodb instance, another a web service exposed to internet that uses another 3 container that are instances of the same image, that use spark to generate sets of data into the mongodb instance.

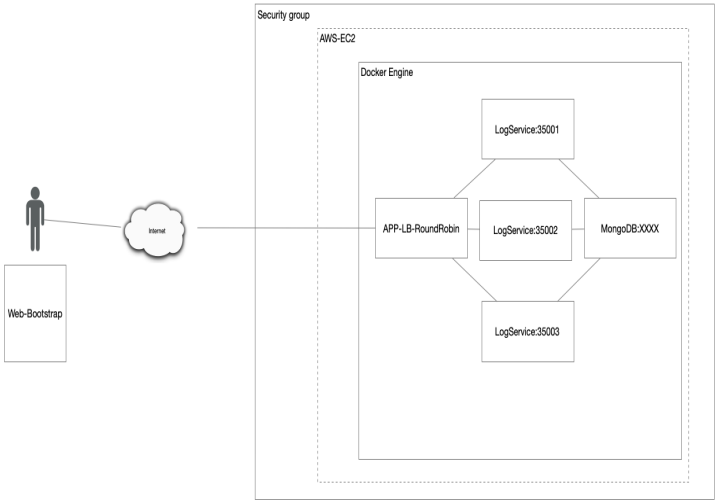

Fig. 3. arq

V. Evaluation

The solution that we made was a simple an easy method for building multiple application that are just equal, no matter what the hardware is, just by looking the images and making instances.

VI. Conclusions

Docker is a powerful tool for making different kind of applications that can be reusable and maintainable into a architecture. Docker allow to have great performance and escalability.

References

[1] W. Techs, "11 points to consider when virtualizing security." https://resources.infosecinstitute.com/11-points-consider-virtualizing-security/, JANUARY 2018. Accessed on 2020-09-22.

[2] jgarzas, "¿qué es docker? ¿para qué se utiliza? explicado de forma sencilla." https://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html, JULY 2015. Accessed on 2020-09-22.