

SYSTEMS ENGINEERING

Enterprise Architectures

Laboratory 5

Luis Daniel Benavides Navarro

Author:
Angie Daniela Ruiz Alfonso

Contents

1	Introduction	2
2	Architecture	2
3	Results	3
4	References	4

1 Introduction

In this lab create a small web application using the Spark java micro-framework, also create a container for docker (to be used with the aforementioned application), where you deploy and configure both locally. Then in a repository that I created in Docker-Hub I uploaded the image so that finally in a virtual machine that I created in AWS (where we previously installed Docker) I could deploy my container.

2 Architecture

The architecture used is the Round Robin and we can see the physical design in the figure 1, this is a simplest architecture of an embedded system. The main method involves a loop that runs again and again, checking each I/O device in each round to see if they need maintenance, this no fancy interrupts, no worry about sharing data and use just a simple thread of execution can execute again and again.[1]

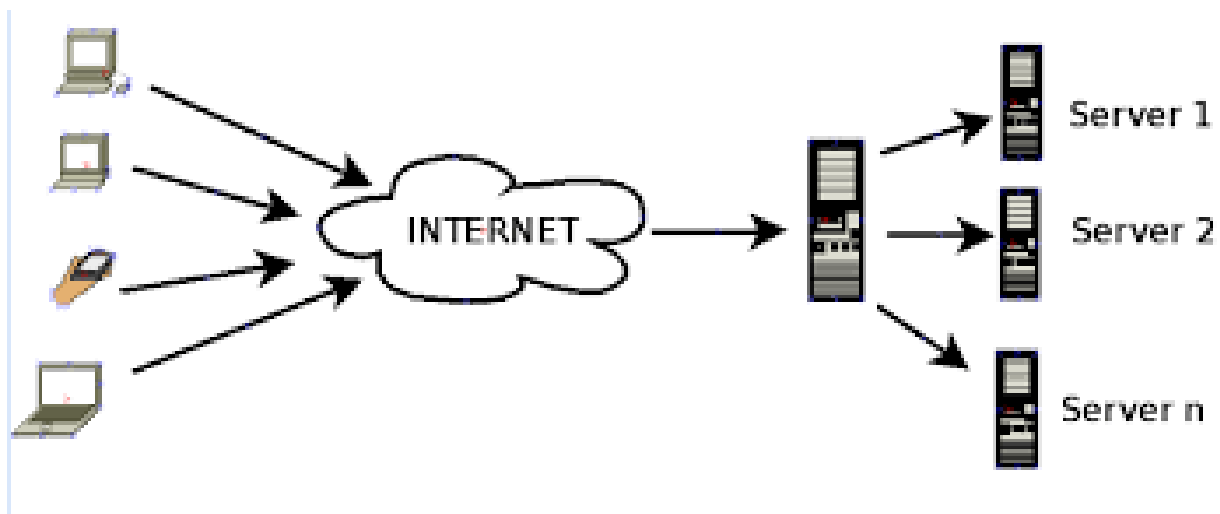


Figure 1: The architecture used

Pros:

- Simplest of all the architectures
- No interrupts
- No shared data
- No latency concerns
- No tight response requirements

Cons:

- A sensor connected to the Arduino that urgently needs service must wait its turn.
- Fragile. Only as strong as the weakest link. If a sensor breaks or something else breaks, everything breaks.
- Response time has low stability in the event of changes to the code.

3 Results

The results that were obtained in this laboratory are in the figure 2,3, 4

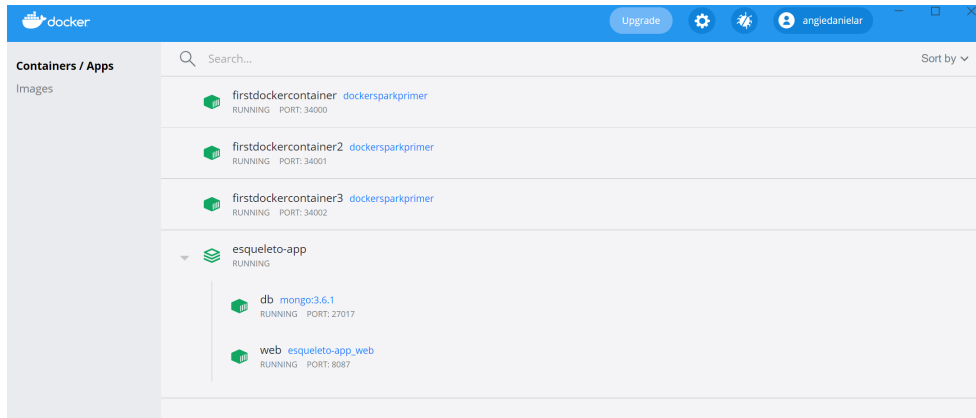


Figure 2: The results obtained

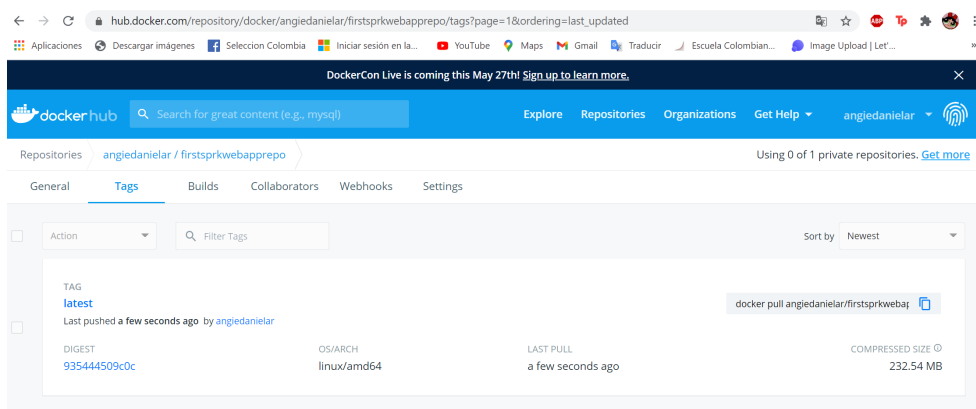
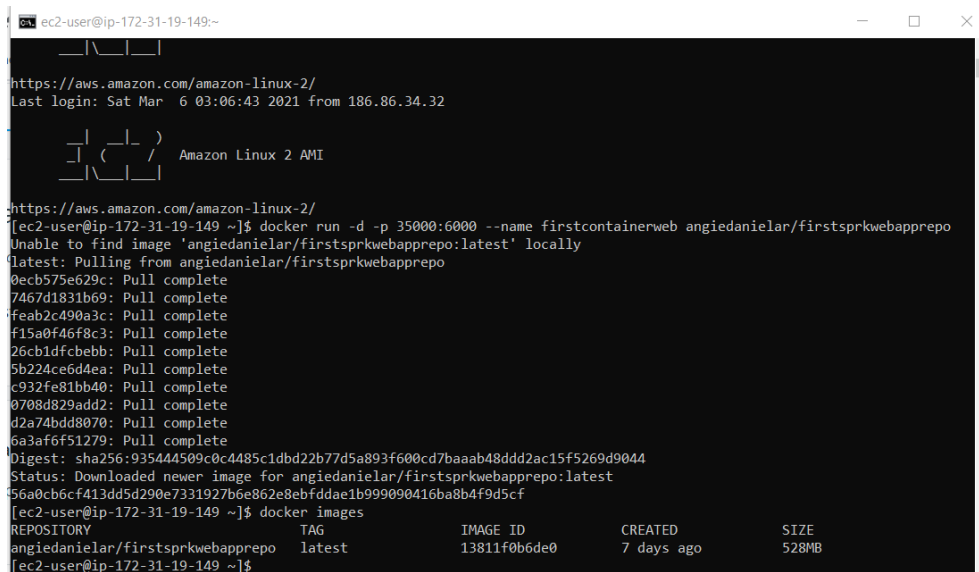


Figure 3: The results obtained



```

ec2-user@ip-172-31-19-149:~$
https://aws.amazon.com/amazon-linux-2/
Last login: Sat Mar  6 03:06:43 2021 from 186.86.34.32

 _ _ _ _ _
| |   | |   |
| |   | |   |   Amazon Linux 2 AMI
|_|   |_|   |

ec2-user@ip-172-31-19-149:~$ docker run -d -p 35000:6000 --name firstcontainerweb angiedanielar/firstsprkwebapprepo
Unable to find image 'angiedanielar/firstsprkwebapprepo:latest' locally
latest: Pulling from angiedanielar/firstsprkwebapprepo
0ecb575e629c: Pull complete
7467d1831b69: Pull complete
feab2c490a3c: Pull complete
f15a0f46f8c3: Pull complete
26cb1dfcbebb: Pull complete
5b224ce6d4ea: Pull complete
c932fe81bb40: Pull complete
0708d829add2: Pull complete
d2a74bdd8070: Pull complete
6a3af6f51279: Pull complete
Digest: sha256:935444509c0c4485c1dhd22b77d5a893f600cd7baaab48ddd2ac15f5269d9044
Status: Downloaded newer image for angiedanielar/firstsprkwebapprepo:latest
56a0cb6cf413dd5d290e7331927b6e862e8ebfddae1b999090416ba8b4f9d5cf
ec2-user@ip-172-31-19-149:~$ docker images
REPOSITORY              TAG               IMAGE ID           CREATED            SIZE
angiedanielar/firstsprkwebapprepo   latest           13811f0b6de0      7 days ago        528MB
ec2-user@ip-172-31-19-149:~$

```

Figure 4: The results obtained

4 References

- [1] A. Addison., *Round-robin vs function-queue-scheduling | embedded software architecture*, <https://automaticaddison.com/round-robin-vs-function-queue-scheduling-embedded-software-architecture/#:~:text=The%20Round%20Robin%20architecture%20is,see%20if%20they%20need%20service>, Accessed on 2021-03-15.