Requirements specification document
Design Project

Christopher Arredondo-Fallas
2016094916

Tecnológico de Costa Rica

Contents

1 Introduction

1.1 System purpose

The purpose of this project is to test the idea of cloud edge computing using Wind River products and Kubernetes, and how this can facilitate the implementation of this technology. The idea is to use three layers of cloud edge computing where each layer is in charge of specific tasks depending on the latency importance of the task.

1.2 System scope

The scope of this project is to fully implement a cloud edge computing prototype using Wind River Linux and Kubernetes, the first layer will be implemented on an embedded device of choice, the second layer will be implemented on a server located inside of the company and finally the third layer will be located on a cloud server of choice. To test this product, a simple app for image processing will be implemented. The general scope of this project is to demonstrate that it is feasible to use Wind River products in cloud edge computing.

1.3 System overview

The complete project consists of a system that implements the cloud edge computing paradigm with three levels of computing power, the first level is a local processing embedded device which is the entry to the system. This level is in charge of low processing tasks and the idea of this level is to have it replicated several times locally to reduce latency. The second layer is in charge of mid processing tasks and the idea of this layer is to have several branches so it can have availability if necessary. Finally there's the last layer, this layer is in charge of high computing tasks and storage of data. The idea of this system is for it to act as one single service where internally it will redirect the traffic to higher levels depending on the task. This will solve latency issues in single server systems where all the services are done in the same device.

1.3.1 System context

The problem this system solves is the idea that tasks that need low latency, usually get affected by cloud server performance because these systems usually are located really far away from the device in need of the service. A cloud edge computing system solves these issues by having levels of devices in which the lower the level the closer it is to the device in need of the service. In this case, out first level is a low power embedded device which is located as close as possible to the device.

1.3.2 System functions

The practical function of this project is to measure the latency advantage of a cloud edge computing system versus a single server system like AWS. In general the function of this project is to lower latency's in services that need fast and reliable connections and cannot afford to wait for a result from a server on the other side of the planet. This is important for application like autonomous vehicles in which a good solution to latency when connecting to a server is an implementation of cloud edge computing with an embedded device near by being the entrance of the system that could be even inside the vehicle.

1.3.3 User characteristics

The users that will test this system consists of coworkers that are inside of the company, the reason for this is that the products used in the implementation of this service are owned by Wind River Systems and it is not allowed to share the final product with external people. In general the users of this product are engineers, familiar with the technologies implemented in this product.

1.4 Definitions

1.4.1 Embedded Device

An embedded device is an object that contains a special-purpose computing system. The system, which is completely enclosed by the object, may or may not be able to connect to the Internet. Embedded systems have extensive applications in consumer, commercial, automotive, industrial and healthcare markets.

1.4.2 Wind River Linux

Wind River® Linux is the industry's most advanced embedded Linux development platform, with a comprehensive suite of products, tools, and life cycle services to build and support intelligent edge devices.

1.4.3 Cloud

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers.

1.4.4 Latency

Latency = delay. It's the amount of delay (or time) it takes to send information from one point to the next. Latency is usually measured in milliseconds or ms. It's also referred to (during speed tests) as a ping rate.

2 References

- Wind River Linux. (2020). Retrieved 24 February 2020, from https://www.windriver.com/products/linux/
- Production-Grade Container Orchestration. (2020). Retrieved 24 February 2020, from https://kubernetes.io/

3 System requirements

3.1 Functional requirements

3.1.1 [FR-001]

The system must not fail

3.1.2 [FR-001]

The system must use Wind River Linux as its embedded operating system.

3.2 Usability requirements

3.2.1 [UR-001]

The user should be able to send and view images using an app.

3.3 Performance requirements

3.3.1 [PR-001]

The latency of the first layer must be less than the second and third, and the latency of the second layer must be less than the third layer.

3.3.2 [PR-002]

The Linux image must be as small as possible giving its a embedded system.

3.4 System interface

3.4.1 [SIR-001]

The app should have a way to send images to the implemented system.

3.4.2 [SIR-002]

The app should have a way to obtain the sent image.

3.5 System operations

3.5.1 [SOR-003]

The system must be able to run 24/7.

3.6 System modes and states

3.6.1 [SMS-001]

The app should have a way to send images to the implemented system.

3.7 Physical characteristics

3.7.1 [PCR-001]

The first layer must be implemented in a embedded system of choice.

3.7.2 [PCR-002]

The second layer must be implemented in a local server.

3.7.3 [PCR-003]

The third layer must be implemented in a cloud server.

3.8 Environmental conditions

3.8.1 [ECR-001]

The embedded system must be a low power device to reduce environmental impact.

3.9 System security

3.9.1 [SSR-001]

The transmission of data can be implemented with clear text.

3.10 Information management

3.10.1 [IMR-001]

The data in the server does not need to be encrypted.

3.11 Policies and regulations

3.11.1 [PRR-001]

The code must not be shared with anybody external from Wind River Systems.

4 Verification

4.0.1 [VR-003]

The system must have a way to verify latency's.

5 Appendices

5.1 Diagrams

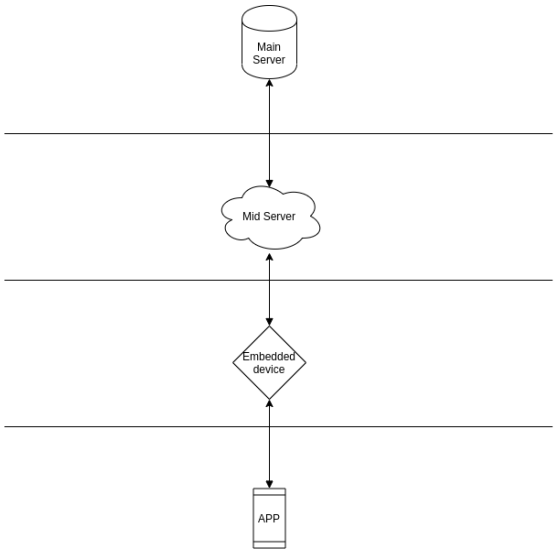

Figure 1: General diagram of the system to be implemented.

5.2 Acronyms and abbreviations

5.2.1 WRL

Wind River Linux

5.2.2 K8s

Kubernetes

5.2.3 AWS

Amazon web services