IITB SUMMER INTERNSHIP 2017

Software Requirement Specification

IP-based Hardware Infrastructure Management for the Cloud

Principal Investigator

Prof. D.B.Phatak

Project In-charge

Mr. Nagesh Karmali

Project Mentor

Minali Upreti

Project Members

Harsh Khatore
Rohit Mehta
Mayank Gupta
Tasneem Lightwala

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the project ***IP-based Hardware Infrastructure Management for the Cloud.*** *It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.*

1.2 Document Conventions

In general this document prioritizes in writing the requirements of the system and analyzing in details the tools being provided to its users. Every requirement is having its own priority (non conflicting) . In addition few figures are also being provided to make requirements more clear to the reader.

1.3 Intended Audience and Reading Suggestions

This document is intended for any individual user, developer, tester, project manager or document writer that needs to understand the basic system architecture and its specifications. Here are the potential uses for each one of the reader types:

Developer: The developer of this project are pre-final year B.Tech. students. They are developing this project as their research internship at IIT Bombay.

User: The user of this program can be any person or community or organization who wants to monitor hardware infrastructure for cloud.

Tester: The tester of this project are the students who are developing this project.

1.4 Product Scope

The scope of the project is to build a hardware infrastructure manager using network protocols like IPMI and set up a user interface (on a Docker container) suited to the needs of the cloud management.

Hardware Infrastructure Management of cloud includes monitoring fan rotation speeds, temperatures, voltages, power consumption, power supply performance. This project we are also dealing with failure detection of power supply or fans or disks.

1.5 References

1. *"1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE computer Society, 1998. - IEEE Std 830"*
2. *https://www.taashee.com/nagios*
3. *http://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/SA/v14/i03/a5.htm*
4. *http://docs.opennebula.org/4.12/administration/monitoring/mon.html#the-monitor-metrics*
5. *http://openipmi.sourceforge.net/*
6. *https://www.docker.com/what-docker*
7. *https://www.docker.com/what-container*

2. Overall Description

2.1 Product Perspective

The product is supposed to be an extension of an already open source monitoring application called Nagios. The extension will provide an IPMI interface to the server, for monitoring clients, at the hardware level.

2.2 Product Functions

The major product functions are:

❑ *Monitoring of hardware components of the system below the operating system level.*

❑ *Notifying the server system about any malfunction in the client systems.*

2.3 User Classes and Characteristics

2.3.1 Physical Actors:

A) Administrator:

The Administrator is the one who monitors the client system and receives notification about any failures and acts upon it.

2.3.2 System Actors:

A) Client:

The Client is the individual server in the cloud.

B) Server:

The Server is the system which receives information related to a client system and acts accordingly.

2.4 Operating Environment

This is a web based system and hence will require the operating environment for a client and server GUI. This will be operating only in the Linux/Unix operating environment:

2.5 Design and Implementation Constraints

2.5.1 Software Constraints:

1. Only available for Linux based servers.

2.5.2 Hardware Constraints:

1. The monitoring server must have an IPMI management controller.
2. The server must be connected to all the systems in cloud.

2.6 Assumptions and Dependencies

A) Assumptions:

1. Proper power supply and connectivity is been provided among all the clients, servers.
2. There is IPMI support on all the servers for remote controlling.

B) Dependencies:

1. This project is an extension to Nagios and its IPMI plugin.

3. External Interface Requirements

3.1 User Interfaces

The purpose of the product is to provide a GUI based application which monitors the client system and informs about any failures in any of the servers in cloud.

3.2 Hardware Interfaces

The BMC chip in the system interfaces with the IPMI port, and sends hardware information to the monitoring server, over a dedicated port. Internally, IPMB, which is used by IPMI, communicates with various sensors and control devices.

3.3 Software Interfaces

The system uses Nagios as a monitoring tool along with the IPMI addon plugin.

3.4 Communications Interfaces

The monitoring tool Nagios monitors the server continuously and communicates with the users about problems, and also communicates with hosts and software to take remedial action.

4. System Features

This section concentrates on all of the system features are provided by the system:

4.1 Hardware monitoring

4.1.1 Description and Priority

Continuous monitoring of a client server system in cloud, using IPMI plugin in Nagios and thus, being able to detect hardware malfunctions. This is a high priority task.

4.1.2 Stimulus/Response Sequences

1. Nagios continuously probes the client , and sends system hardware details over to server.
2. Server checks the report and detects if any critical condition has arrived.
3. If anything unusual happens on the client side, it sends an email to the administrator on the server side.
4. Appropriate action can be taken.

4.1.3 Functional Requirements

REQ-1: Powers on/off the server.
REQ-2: Checks the server status and notify whenever a server is down.
REQ-3: Return an error log.
REQ-4: List status of all sensors (Temperature, voltage, and fan sensors).
REQ-5: Detect and notify any disk failure. Also identify which disk has failed.

4.2 Adding a system to be monitored:

4.1.1 Description and Priority

Administrator can add an additional system to be monitored by the server by using the IP address of the remote system to be monitored.

4.1.2 Stimulus/Response Sequences

The server also begins to monitor the additional system.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must be interactive and there should be less delays. The server system should be able to detect and analyse the data received from the remote system and generate notifications and warnings.

5.2 Security Requirements

The main concern regarding security is to have a proper remote access mechanism for the server so that the network connection between the server and client should not be compromised.

5.3 Software Quality Attributes

Availability: All the functionalities of the software are available at all times as long as the administrator server is up. A backup server can also run the software to avoid loss of data during server crashes.

Usability: *The system is easy to handle and navigates in the most expected way with no delays.*

Reliability: *Nagios is most popularly used open source network monitoring solution. It is highly reliable.*

6. Other Requirements

Appendix A: Glossary

None

Appendix B: Analysis Models

None

Appendix C: To Be Determined List

1.) *Exploring the features of Nagios IPMI plugin, and adding more functionalities if needed.*
2.) *Explore the projects and modules that are being built by the Nagios community and contribute to them if possible.*