Ministry of education and science of the Russian Federation
Federal state-founded educational institute of high professional education

«NATIONAL RESEARCH TOMSK POLYTECHNIC UNIVERSITY»

Institute Cybernetics
Educational programme Computer Science and Engineering
Department Software Engineering

MASTER THESIS

Research title
Web application to manage and monitor many Linux servers

UDC 004.383.2-047.36 :004.451

Student

Group	Name	Signature	Date
8ВМ5И	V. Pardosi		

Supervisor

Position	Name	Academic degree	Signature	Date
Associate professor	S.V. Axyonov	PhD		

CONSULTANTS:

On «Financial management, resource efficiency and resource saving» chapter

Position	Name	Academic degree	Signature	Date
Associate professor	N.O. Chistyakova	PhD		

On «Social responsibility» chapter

Position	Name	Academic degree	Signature	Date
Associate professor	Y.V. Anischenko	PhD		

PERMIT TO DEFENCE:

Head of department	Name	Academic degree	Signature	Date
Head of department	M.A. Ivanov	PhD		

Tomsk – 2017

Ministry of education and science of the Russian Federation
Federal state-founded educational institute of high professional education

«NATIONAL RESEARCH TOMSK POLYTECHNIC UNIVERSITY»

Institute Cybernetics
Educational program Computer Science and Engineering
Department Software Engineering

APPROVED BY:
Head of department
_____ _____ M.A.Ivanov
(Signature) (Date) (Name)

TASK
for the final qualifying research

Form:

Master thesis

To student:

Group	Name
8ВМ5И	Victor Pardosi

Title:

Web application to manage and monitor many Linux servers	
Approved by the rector's order (date, ID)	

Date of research completion:	02.06.2017

TECHNICAL TASK:

Initial data *(Product requirements, User services, Description of solution, Market analysis, Impact on the Environment)*	The purpose of this thesis to create an application which provides the easy way to manage hundreds Linux servers with the ability to send a command to multiple Linux servers, ability to upload a file to multiple Linux servers, provide automation system to check all services and repair the services, also monitor resources of servers.
List of tasks must be presented in the thesis *(Review. Related research, Task description, Research procedure, Development and design procedures, Results obtained, Additional chapters, Appendix, Conclusion).*	Overview technologies and programming tools, Methods of systems to monitoring and handling multiple servers, Designing and programming the system, System testing, Financial management, resource efficiency and resource saving, Social responsibility, Conclusion

List of graphical data:	Presentation

Consultants

Part	Consultants
Financial management, resource efficiency and resource saving	N.O. Chistyakova, Associate professor, PhD
Social responsibility	Y.V. Anischenko, Associate professor, PhD

Date of task obtaining	22.03.2017

The task was given by:

Position	Name	Academic degree	Signature	Date
Associate professor	S.V. Axyonov	PhD		

The student gets the task:

Group	Name	Signature	Date
8ВМ5И	V.Pardosi		

Планируемые результаты обучения по ООП

Код Результата	Результат обучения (выпускник должен быть готов)
Профессиональные компетенции	
Р1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области информатики и вычислительной техники.
Р2	Применять глубокие специальные знания в области информатики и вычислительной техники для решения междисциплинарных инженерных задач.
Р3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием аппаратных и программных средств информационных и автоматизированных систем, с использованием аналитических методов и сложных моделей.
Р4	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
Р5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
Р6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
Универсальные компетенции	
Р7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности.
Р8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
Р9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
Р10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.

Код Результата	Результат обучения (выпускник должен быть готов)
Профессиональные компетенции	
Р11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной деятельности, способность к педагогической деятельности.

Ministry of education and science of the Russian Federation
Federal state-founded educational institute of high professional education

«NATIONAL RESEARCH TOMSK POLYTECHNIC UNIVERSITY»

Institute Cybernetics
Educational program Computer Science and Engineering
Educational level Master
Department Software Engineering
Research period Summer term 2016-2017

Form:

Master thesis

CALENDAR RATING PLAN
of the final qualifying research

Date of research completion:	02.06.2017

Checkpoint date	Research section	Max score
22.03.2017	*Overview technologies and programming tools*	*10*
29.04.2017	*Designing and programming the software*	*20*
15.05.2017	*Software testing*	*20*
20.05.2017	*Financial management, resource efficiency and resource saving*	*20*
25.05.2017	*Social responsibility*	*20*
30.05.2017	*Presentation*	*10*

The task was given by:

Position	Name	Academic degree	Signature	Date
Associate professor	S.V. Axyonov	PhD		

ARGEED BY:

Head of department	Name	Academic degree	Signature	Date
Head of department	M.A. Ivanov	PhD		

TASK FOR
«FINANCIAL MANAGEMENT, RESOURSE EFFICIENCY AND RESOURSE SAVING» PART

To student:

Group	Name
8ВМ5И	Victor Pardosi

Institute	**Cybernetics**	**Department**	**Software Engineering**
Educational level	**Master**	**Educational program**	**Computer Science and Engineering**

Initial data to «Financial management, resource efficiency and resource saving » chapter:	
1. *Costs of research, including technical, financial, energy, information and human costs* 2. *Norms of expenditure of resources* 3. *The taxation system used, the rates of taxes, discounting and lending*	*Work with related research presented in articles, journals, bulletins, and official documents*
List of tasks:	
1. *Evaluation of commercial and innovative potential* 2. *Development of the charter of the technical project* 3. *Planning of management process: structure and schedule, budget, and risks* 4. *Estimation of resource, financial and economical efficiency*	*Analysis of potential consumers. Assessment of the quality and prospective of the project. Research planning.*
List of graphical data:	
1. *«Portrait»of consumer* 2. *Market segmentation* 3. *Assessment of the completeness of solution* 4. *FAST diagram* 5. *SWOT matrix* 6. *Calendar and budget of the research* 7. *Assessment of resource, financial and economical efficiency* 8. *Potential risks*	

Date of task obtaining	

The task was given by the consultants:

Position	Name	Academic degree	Signature	Date
Associate professor	N.O. Chistyakova	PhD		

The task was accepted by the student:

Group	Name	Signature	Date
8ВМ5И	V. Pardosi		

TASK FOR
«SOCIAL RESPONSIBILITY» PART

To student:

Group	Name
8ВМ5И	Victor Pardosi

Institute	**Cybernetics**	**Department**	**Software Engineering**
Educational level	**Master**	**Educational program**	**Computer Science and Engineering**

Initial data to «Social responsibility» chapter:	
1. Description of work place: – *Harmful factors in the industrial environment (meteorological conditions, harmful substances lighting, noise, vibrations, electromagnetic fields, ionizing radiation)* – *dangerous industrial factors (mechanical, thermal, electrical, etc.)* – *negative impact on the environment (atmosphere, hydrosphere, lithosphere) emergency situation (industrial, natural, ecological types)*	*Work place located in the office 421 in the Institute of Cybernetics Building*
2. Legislative and normative documents on the topic	*State standards, GOST, SNiP, NPB, SanPiN, federal laws*
List of tasks:	
1. Analysis of the identified harmful factors of the industrial environment in the following sequence: *– the physical and chemical nature of harmfulness, its relation to the topic being developed;* *– the effect of the factor on the human body;* *– reduction of permissible norms with the required dimensionality (with reference to the relevant normative and technical document);* *– proposed remedies*	Identification of all the harmful factors when researching, including physical, chemical and biological
2.Analysis of identified hazards of the industrial environment in the following sequence *– mechanical hazards (sources, means of protection;* *– thermal hazards (sources, means of protection);* *– electrical safety (including static electricity, lightning protection - sources, protective equipment);* *– fire and explosion safety (causes, preventive measures, primary means of fire extinguishing)*	Identification of the all possible hazards when researching
3.Protection of the environment: *– protection of the residential area* *– analysis of the impact of the facility on the atmosphere (emissions);*	Identification of the all possible kinds of waste when researching

– *analysis of the impact of the object on the hydrosphere (discharges);* – *analysis of the impact of the object on the lithosphere (waste);* – *develop solutions to ensure environmental safety with references to environmental standards.*	
4.Protection in emergency situations: *List of possible emergencies on the site;* *Choice of the most typical emergency situation;* *Development of preventive measures to prevent emergencies;* *Development of measures to improve the stability of the facility to this emergency situation;* *The development of actions as a result of the emergencies and measures to eliminate its consequences*	Identification of the all possible emergencies when researching
List of graphical data:	
Graphical plans	

Date of task obtaining	

The task was given by the consultants:

Position	Name	Academic degree	Signature	Date
Associate professor	Y.V. Anischenko	PhD		

The task was accepted by the student:

Group	Name	Signature	Date
8ВМ5И	V. Pardosi		

Table of Contents

Abstract

Whether a business is small or big, today's businesses prefer to run their own server to ensure that all services are delivered well, either their lease or host it in the data center. Typically, servers are managed by a system administrator who holds the responsibility to ensure all services in the servers are running smoothly as well are the configuration and the maintenance of the servers.

SSH protocol allows a systems administrator to handle various commands with securely. It is easy to remote and executes command in the server through SSH, but if they have hundreds of servers, they might experience some problems. An example to check how much CPU and Disk usage, a system administrator needs to remote the server through SSH and execute the command and then repeat this steps to all servers.

The purpose of this thesis to create an application which provides the easy way to manage hundreds Linux servers with the ability to send a command to multiple Linux servers, ability to upload a file to multiple Linux servers, provide automation system to check all services and repair the services, also to monitor resources of servers.

Keywords: Web Application; Server; Monitoring; Networking; System Administrator

Introduction

Whether a business is small or big, today's businesses prefer to run their own server to ensure that all services are delivered well, either their lease or host it in the data center. Typically, servers are managed by a systems administrator who holds the responsibility to ensure all services in the servers are running smoothly as well are the configuration and the maintenance of the servers.

In most cases, organization possesses more than one server with examples such as web server, database server, backup server, file server and load balancer server for balancing high traffic website. Therefore, a server using Linux operating system should lean towards using SSH Protocol in order to manage many instructions such as configuration, updating, installation changing files and monitoring services. Further, the use of SSH Protocol allows a system administrator to handle various commands with securely. It is not hard to handle all those things if they have only little servers, but if they have hundreds of servers, they might experience some problems. An example to check how much CPU and Disk usage, a system administrator needs to remote the server through SSH and execute the commands and then repeat this steps to all servers.

Those issues also make high cost because if they run hundreds of servers, then they need more system administrator to handle the servers to do all tasks. Moreover, also they need to monitor all services, including server performance such as how much ram utilize, how much resource like CPU and Disk usage, ensuring all those resources are not over usage and this might spend much time depending on how many servers they have. Those issues are facing by a system administrator, and they need to repeat all steps to monitor all servers.

Thus, the purpose of this thesis to write Web-Based Application to solve all those issues with ability to send command to multiple Linux servers, ability to modify same file to multiple Linux servers, provide automation system to check all services and start or restart service if it is down or reach the peak, provide an alert to email or social media of system administrator if there are abused or over usage in Network, CPU, RAM which will interference server performance, the system will send an alert notification if the automation system cannot handle it, because the research further seeks to provide an automated system, so the organizations would not need many system administrators to control their servers and impact to reduce the cost of managing servers.

Chapter 1 Overview technologies and programming tools

1.1 Methods and Technologies for building an application

A cryptographic network protocol for operating network services securely over an unsecured network access known as Secure Shell (SSH), through SSH protocol users can securely remote other computers and allow it to authenticate the user and execute commands [1]. Therefore, to ensure communication is secure, this application used SSH to communicate between User Interface and server.

Figure 1.1 displays there are 3 parts to make this system works; a user interface, the forwarder server, and the target server. The user interface provides a web-based interface for sending and received a request to or from the forwarder server. The forwarder server receives a request from user then forward it to the target server and forward back the results to the user interface.

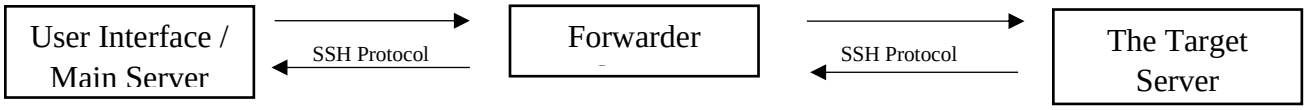

Figure 1.1 Diagram for communication systems

The forwarder server is doing a big part in this system because it handles all communication between the user interface and the target server. If the target server location is around the world then using multi-forwarder server could turn system more efficiently, select the forwarder server from the nearest country or at least same continent with the target server, this will provide faster communication and reliable connectivity because in some countries they have only faster connection to local and slow connection to international.

Moreover, to monitoring all resources, each server is frequently doing several commands to check resource then send all data to the main server then main server catch the data and save all information to the database. There are so many commands in Linux operating systems which can be useful to make the system works, the methods will be explained in chapter 2.

Figure 1.2 displays flowchart for monitoring services, monitoring services are using a different method than monitoring resources, usually, services using ports which open for public, example: Apache (80), SSH (22), FTP (21), etc.

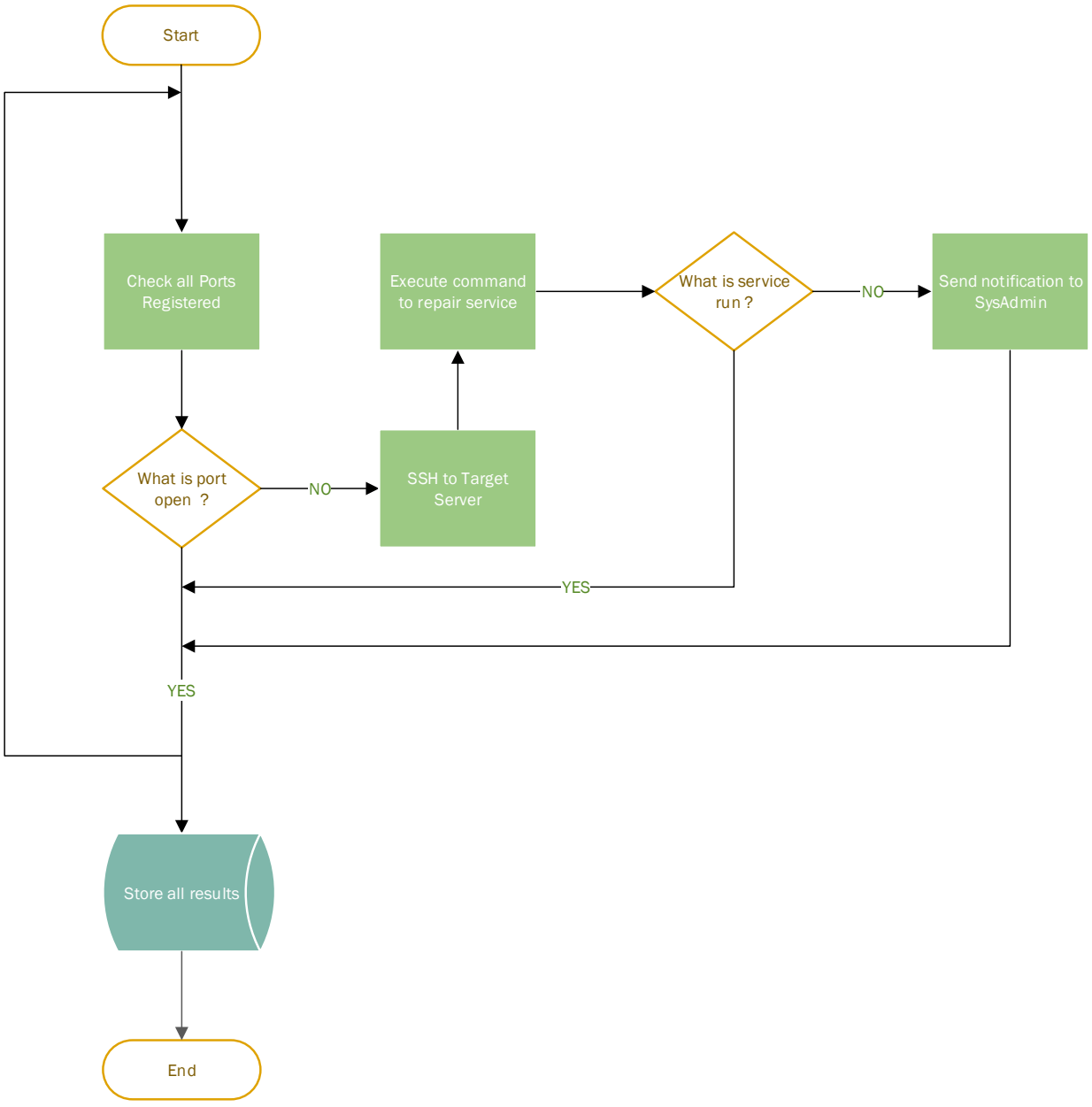

Figure 1.2 Flowchart for monitoring services

Then the system can take advantages of checking the ports, whether the ports is open or closed. If the port is open, then that means service is run, but if the port is closed, then the service is down. Using this method is faster to monitoring services rather than remote server through SSH.

If services are down, then the systems send commands to the forwarder server to repair this issue. The forwarder server remote the target server through SSH and execute a command to repair services. An example: the system has set to schedule to monitor port 22 (Apache) then suddenly port 22 is closed, then the system automatically sends requests to the forwarder server, using all data provided by System then the forwarder server connect to the target server through SSH and execute command service apache restart (CentOS). This command would Stop and Start service, in most cases this command is enough to repair service, but in case the results are not changed then the system will send an email notification to the system administrator and whatever the results then store it into the database.

There is no additional software for the target server, but system administrator needs to insert username and password of target server in the user interface. In some cases, the target server is overloaded caused entirely system down and the forwarder cannot remote through SSH to the destination server, the other way to fix this issue is to using API provided by the target server provider to send a command to boot and reboot.

The last part is using programming languages to build an application also gateway to communicate between User and Server through a secure protocol, System on the application would need to loop through all servers to do all tasks.

1.2 Related technologies

1.2.1 Web Server

1.2.1.1 Apache

The creation of Apache HTTP Server was by Robert McCool in 1995 and was developed under the supervision of Apache Software Foundation. The most popular server on the internet since 1996 is the Apache Web Server resulting in the presence of excellent documentation and integrated support from various software projects. The preference of Apache is due to its flexibility, widespread support and power as well its ability to extend through a dynamically loadable module system.

Apache servers can handle static content using its conventional file-based methods. The performance of these operations is mainly a function of the MPM methods.

Apache can also process dynamic content by embedding a processor of the language in question into each of its worker instances. This allows it to execute dynamic content within the web

server itself without having to rely on external components. These dynamic processors can be enabled through the use of dynamically loadable modules.

Apache's ability to handle dynamic content internally means that configuration of dynamic processing tends to be simpler. Communication does not need to be coordinated with an additional piece of software and modules can easily be swapped out if the content requirements change. [2]

1.2.1.2 Nginx

Nginx invention was to answer the C10k problem which was a challenge in terms of handling ten thousand concurrent connections as required for the modern web. Its launch was in 2004 to meet the goal of event driven architecture. The increased popularity is due to its light weight resource utilization and its ability to scale easily on hardware at a minimum. Nginx excels in static content and is designed to pass dynamic requests. Its choice by systems administrators is its ability to use its resources efficiently. Its weakness is its inability to process dynamic content natively. This means that communication needs to be configured with it and a processor over one of the protocols familiar to Nginx.

For administrators, this means that communication must be configured between Nginx and the processor over one of the protocols Nginx knows how to speak (HTTP, FastCGI, SCGI, uWSGI, Memcache). This can complicate things slightly, especially when trying to anticipate the number of connections to allow, as an additional connection will be used for each call to the processor.

However, this method has some advantages as well. Since the dynamic interpreter is not embedded in the worker process, its overhead will only be present for dynamic content. Static content can be served in a straight-forward manner, and the interpreter will only be contacted when needed. Apache can also function in this manner, but doing so removes the benefits in the previous section. [2]

1.2.2 Programming Language

PHP (Hypertext Preprocessor) is a server scripting language designed by Rasmus Lerdorf, a powerful tool to create dynamic and interactive websites. It is fast, flexible, widely-used scripting language for everything from a simple blog to the most popular and dynamic websites in the world.

Python is a widely-used high-level (but it also used in a wide range of non-scripting language) design for programmers to express concepts with fewer lines of code. It was conceived in the late 1980s and was implemented by Guido van Rossum.

Python code resembles the pseudo-code just like all the scripting languages. The elegant design and syntax rules of this programming language make it quite readable even among the multi-programmer development teams. It supports multiple ways of building the structure and elements of computer programs, including object-oriented and functional programming.[3]

1.2.3 Database

MySQL is the most popular one of all the large-scale database servers. It is feature-rich, open-source product that powers a lot of websites and applications online. Getting started with MySQL is relatively easy and developers have access to a massive array of information regarding the database on the internet. MySQL is sponsored by the Swedish company MySQL AB, which is owned by Oracle Corp. However, the MySQL source code is freely available because it was originally developed as freeware. MySQL is written in C and C++ and is compatible with all major operating systems.

MySQL was a free-software database engine originally developed and first released in 1995. MySQL is named after My, the daughter Michael Widenius, of one of the product's originators. It was originally produced under the GNU General Public License, in which source code is made freely available.

MySQL is very popular for Web-hosting applications because of its plethora of Web-optimized features like HTML data types, and because it's available for free. It is part of the Linux, Apache, MySQL, PHP (LAMP) architecture, a combination of platforms that is frequently used to deliver and support advanced Web applications. MySQL runs the back-end databases of some famous websites, including Wikipedia, Google and Facebook- a testament to its stability and robustness despite its decentralized, free-for-all philosophy.

MySQL was originally owned by Sun Microsystems; when the company was purchased by Oracle Corp. in 2010, MySQL was part of the package. Although MySQL is technically considered a competitor of Oracle DB, Oracle DB is mainly used by large enterprises, while MySQL is used by smaller, more Web-oriented databases. In addition, MySQL differs from Oracle's product because it's in the public domain.

MySQL supports multiple storage engines each with its own specifications while other systems like SQL Server only support a single storage engine. Moreover, MySQL has high performance compared to other relation database systems. This is due to its simplicity in design and support for multiple storage engines. Also, MySQL works on many platforms which means it can be deployed on most machines. Other systems such as MS SQL Server only run on the Windows platform. [4]

1.2.4 Secure Shell (SSH)

SSH, also known as Secure Socket Shell, is a network protocol that provides administrators with a secure way to access a remote computer. SSH is designed to deliver strong authenticity and secure encrypted data communication between computers connecting over an insecure network. Network

administrator uses SSH to manage systems and applications remotely thus providing users with login rights, execution of commands and moving files from one computer to another.

1.2.5 Linux Operating System

Linux is an open source operating system for computers, mainframes, mobile devices, embedded devices and servers. It is one of the most widely supported operating system due to the fact that it is supported on almost every major computer platform.

Linux is a Unix-like computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on September 17, 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux to describe the operating system, which has led to some controversy.

Linux was originally developed for personal computers based on the Intel x86 architecture, but has since been ported to more platforms than any other operating system. Because of the dominance of Android on smartphones, Linux has the largest installed base of all general-purpose operating systems. Linux is also the leading operating system on servers and other big iron systems such as mainframe computers, and is used on 99.6% of the TOP500 supercomputers. It is used by around 2.3% of desktop computers. The Chromebook, which runs on Chrome OS, dominates the US K–12 education market and represents nearly 20% of the sub-$300 notebook sales in the US. Linux also runs on embedded systems – devices whose operating system is typically built into the firmware and is highly tailored to the system. This includes TiVo and similar DVR devices, network routers, facility automation controls, televisions, video game consoles and smartwatches. Many smartphones and tablet computers run Android and other Linux derivatives.

The development of Linux is one of the most prominent examples of free and open-source software collaboration. The underlying source code may be used, modified and distributed—commercially or non-commercially—by anyone under the terms of its respective licenses, such as the GNU General Public License. Typically, Linux is packaged in a form known as a Linux distribution (or distro for short) for both desktop and server use. Some of the most popular mainstream Linux distributions are Arch Linux, CentOS, Debian, Fedora, Gentoo Linux, Linux Mint, Mageia, openSUSE and Ubuntu, together with commercial distributions such as Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Distributions include the Linux kernel, supporting utilities and libraries, many of which are provided by the GNU Project, and usually a large amount of application software to fulfill the distribution's intended use.

Desktop Linux distributions include a windowing system, such as X11, Mir or a Wayland implementation, and an accompanying desktop environment such as GNOME or the KDE Software Compilation; some distributions may also include a less resource-intensive desktop, such as LXDE or Xfce. Distributions intended to run on servers may omit all graphical environments from the standard install, and instead include other software to set up and operate a solution stack such as LAMP. Because Linux is freely redistributable, anyone may create a distribution for any intended use.[5]

1.3 Summary

Apache is the best option for the web server on this System because Apache was built to handling many connections and also can execute dynamic content within the web server itself without having to rely on external components. For programming language both Python and PHP are powerful to build this System, and also both of them are open source but finding hosting for Python is a bit hard and expensive rather than hosting for PHP. Therefore, this System will be built using PHP language. After deciding the web server and programming language, choosing database is easy because they are related one to all, so to make a good combination between Apache and PHP. Therefore MySQL is a good choice here. Then the last but not the least is Linux as an operating system because Linux has it own repository for all technologies mentioned before, so updating and installing all programs are easier and reliable.

Chapter 2 Methods of systems to monitoring and handling multiple servers

2.1 Monitoring Resources

2.1.1 Introduction

Monitoring resources of servers are necessary to ensure uptime and stability of servers when running all services. If one of the resources is not meet a minimum of requirements, then it might cause the service crash or forcibly stop.

Crontab is a Linux tool which schedules a command or script on the server to run automatically. This tool will be needed to run script frequently and send the result to the main server.

2.1.2 Methods for collecting and storing information

Each server will be installed with Bash script which gathers all information about resources then combine all data and send it to the main server where database hosted. These are some steps from collecting data until sending it to the database server.

Step 1, Collecting information of the server. These are some commands in Linux server to gathering information of resources:

1. Determine IP Address of server
   These are commands to get IP Address of server:

```
ip addr | grep 'state UP' -A2 | tail -n1 | awk '{print $2}' | cut -f1
  -d'/'

ip addr | grep 'inet' -A2 | tail -n1 | awk '{print $2}' | cut -f1  -
  d'/'
```

   Because some servers using different network adapter and have different information on their network adapter, so using those both commands will provide more accurate information.

2. Collecting information of Uptime
   This is command to get information about Uptime of server:

```
awk
  '{printf("%d:%02d:%02d:%02d\n",($1/60/60/24),($1/60/60%24),($1/60%6
  0),($1%60))}' /proc/uptime
```

3. Collecting information about CPU usage

   This is a command to get information about CPU usage of the server:

```
echo $[100-$(vmstat 1 2|tail -1|awk '{print $15}')]
```

4. Collecting information of RAM and RAM utilized
   This is command to get information about free RAM and RAM utilized:
   ```
   free -m | awk 'NR==2{printf "%s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2 }'
   ```

5. Collecting information of Disk and Disk utilized
   This is a command to get information about Disk:
   ```
   df -P -T -B 1k | grep '^/'
   ```

   This is command to get information about Disk usage:
   ```
   df -h | awk '$NF=="/"{printf "%s",$5}'
   ```

6. Set date and time

   Because some servers using different time zone then date and time will be set according to time on the main server

Step 2, Filtering and combining all information. When all information has been acquired then filter all character and remove unnecessary character then combine all information and put | as separator each information of the resource to make it easy when sending data.

Step 3, Sending to the main server. After the data combined then use Curl command in Linux to send the data to the main server, then the main server will catch the data and store it in the database.

2.1.3 Methods of automation

When all data of each server has stored in the database of the main server, then the system use Crontab every 5 minutes to check whether any resources are over the limit, if there is resource over the limit then send an alert email to the system administrator. Crontab can be set lower or higher depending on the specification of the main server.

2.2 Monitoring Services

2.1.1 Introduction

Usually, Linux services using ports which open for public, example: Apache (80), SSH (22), FTP (21), etc. The idea is to use those ports as advantages to ensure services, whether the ports is open or closed. If the port is open, then that means service is run, but if the port is closed, then the service is down. Using this method is faster to monitoring services rather than remote each server through SSH, but some services are not using the port for public, example: MySQL (3306). Mostly MySQL

port is used for the private or local connection, therefore, checking services using port cannot be used here, so an alternative method is remote through SSH to the target server and executes a command to check whether service is run or stop.

2.2.2 Methods to monitoring services

There are 2 methods used to monitoring services, monitoring services through SSH and monitoring services using ports.

1. Monitoring services through SSH

Figure 1.3 displays flowchart for monitoring services through SSH, System begin with trying to remote through SSH to the target server if System unable to remote to SSH then System will send an alert via email to Systems Administrators whether username or password is wrong or because the target server is completely down. If remote to the target server is successfully then System continue with executing command to check status or using grep command to ensure those services are listed and if services are down then System will try to repair the service with execute command to start service, but in case the services are not running then System will send an alert email to Systems Administrators.

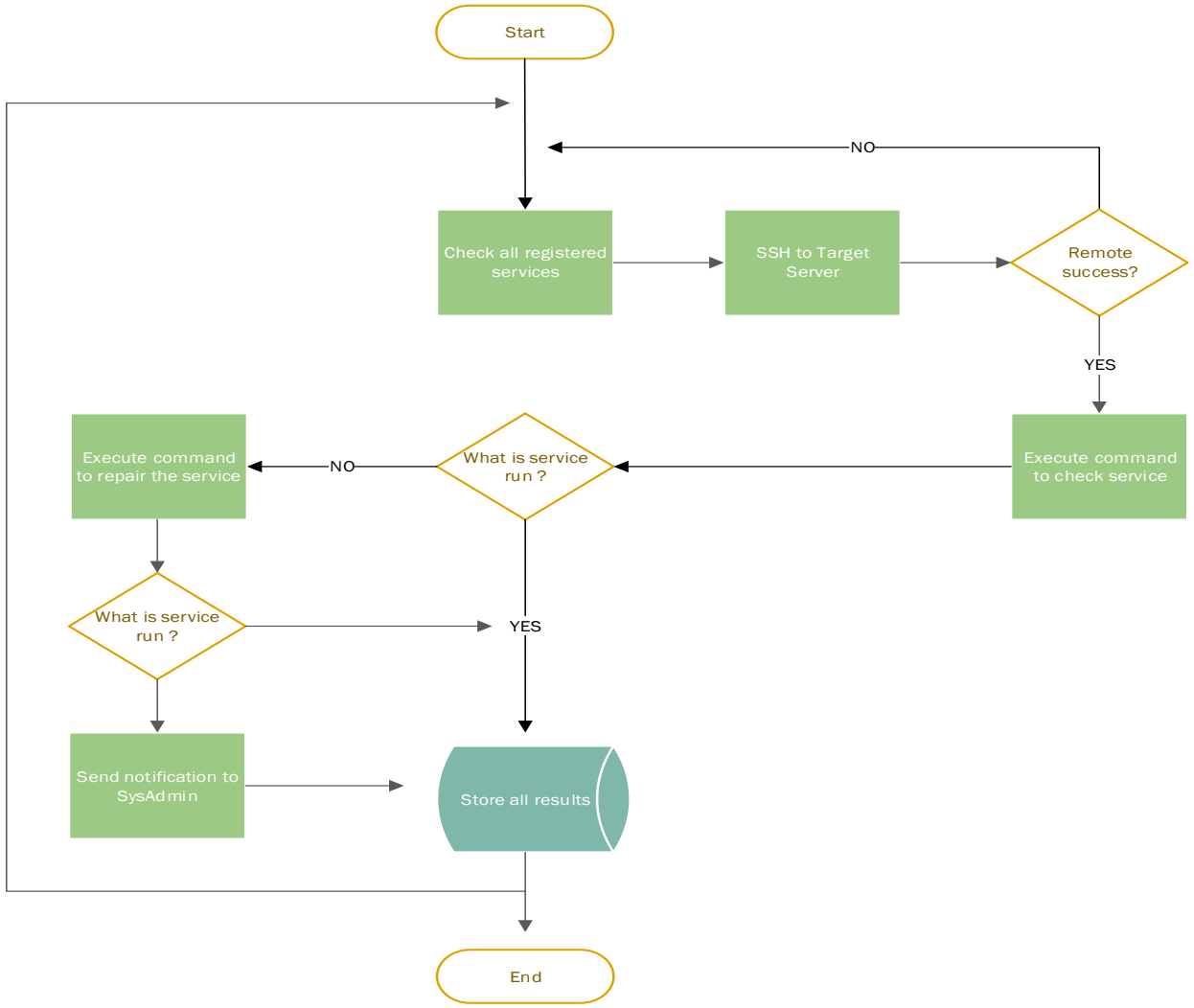

Figure 1.3 Flowchart for monitoring services through SSH

in the end, whatever the results the system will store everything into the database to make easy the system administrator to debugging and handling this issue.

2. Monitoring services using ports.

Figure 1.4 displaying flowchart for monitoring services using ports, System will start with checking all ports registered in the database, then check each port for each server whether the ports is open or closed. If the port is open, then that means service is run, but if the port is closed, then the service is down.

If services are down, then System will send commands to Forwarder server to repair this issue. Forwarder server remote the target server through SSH and execute a command to repair services.

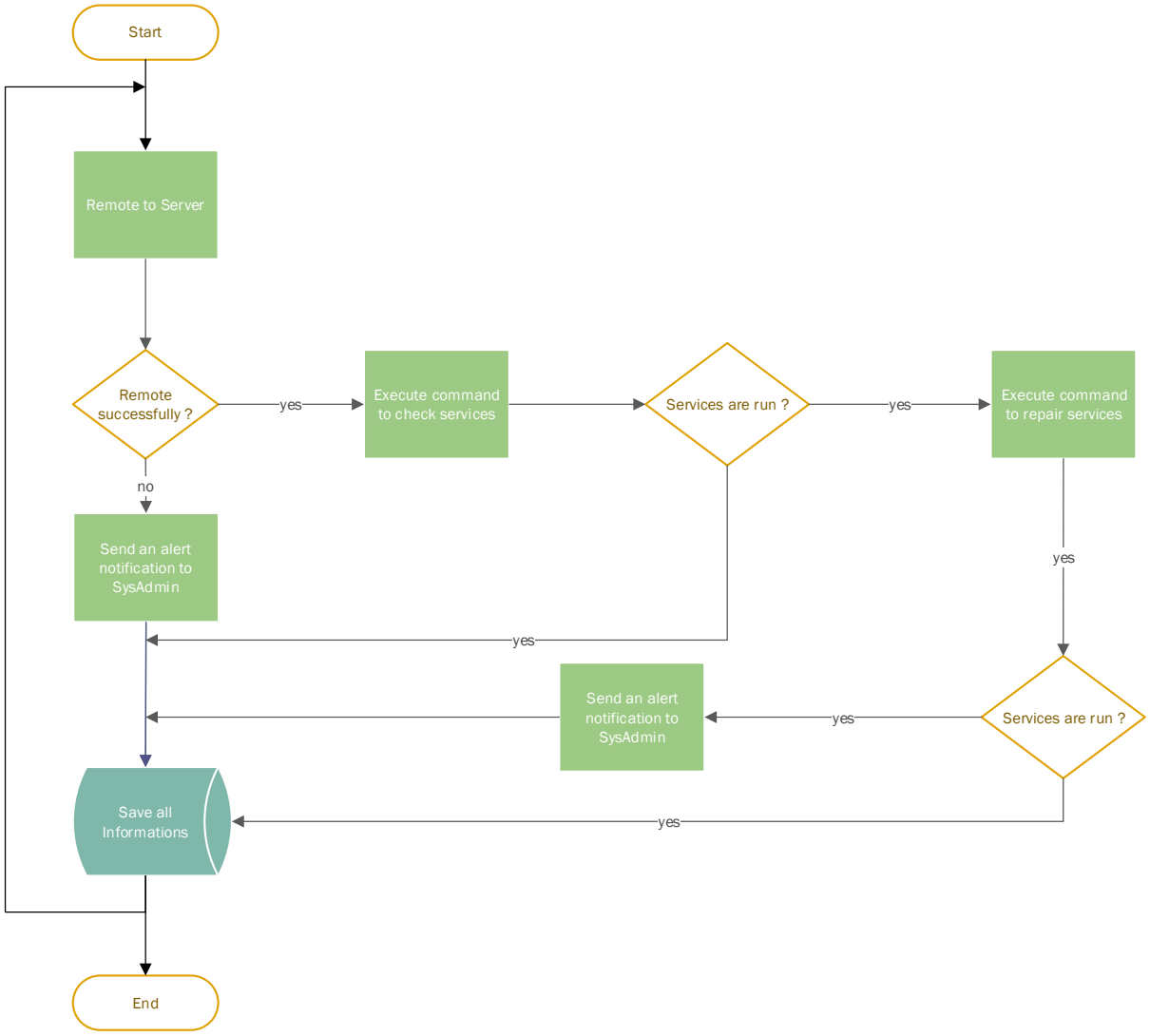

Figure 1.4 Flowchart for monitoring services using ports

An example: the system has set to schedule to monitor port 22 (Apache) then suddenly port 22 is closed, then system automatically send requests to the forwarder server, using all data provided by the system then the forwarder server connect to the target server through SSH and execute command service httpd restart (CentOS). This command would stop and start service, in most cases this command is enough to repair service, but in case the results are not changed then the system will send an email notification to the system administrator and the last store all results into a database.

2.2.3 Methods of automation

Using utilities in Linux named Crontab, the system will run based on the schedule (example: every X seconds) and performance checking and repair all services and stored all results into the database.

2.3 Send commands to multiple servers

Figure 1.5 displays how does this method work, the system start with getting command inputted, and all servers selected by users then send all this data to the forwarder server then

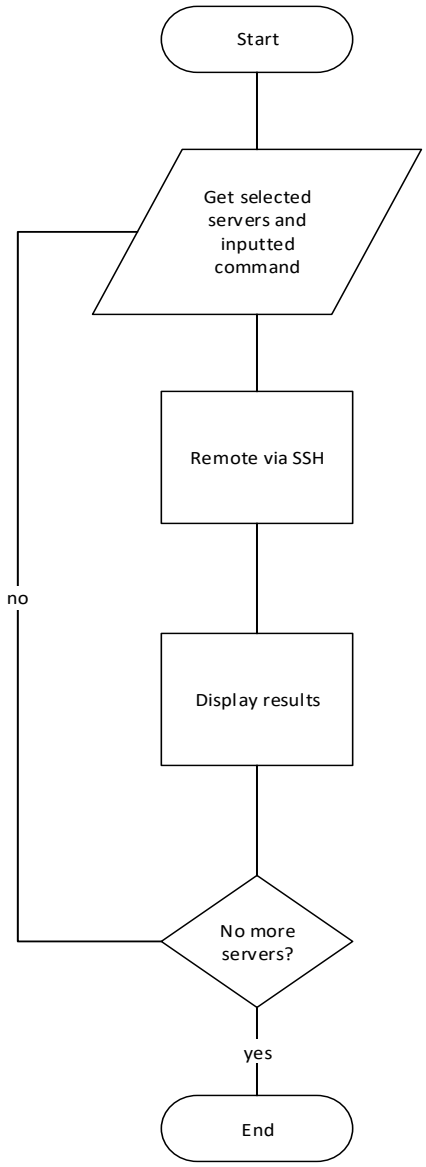

Figure 1.5 Flowchart for sending command to multiple servers

The forwarder server will remote via SSH to target server then return with results then the forwarder server forwards the result to the main server and display results from the server then looping through all servers.

2.4 Upload file to multiple servers

2.4.1 Introduction

This System communicate through SSH therefore to upload file SCP is the suitable protocol, SCP is most useful with a pre-established SSH connection. SCP or Secure Copy is an encrypted method of transferring files over an already established SSH connection,

2.4.2 Methods to uploading file to multiple servers

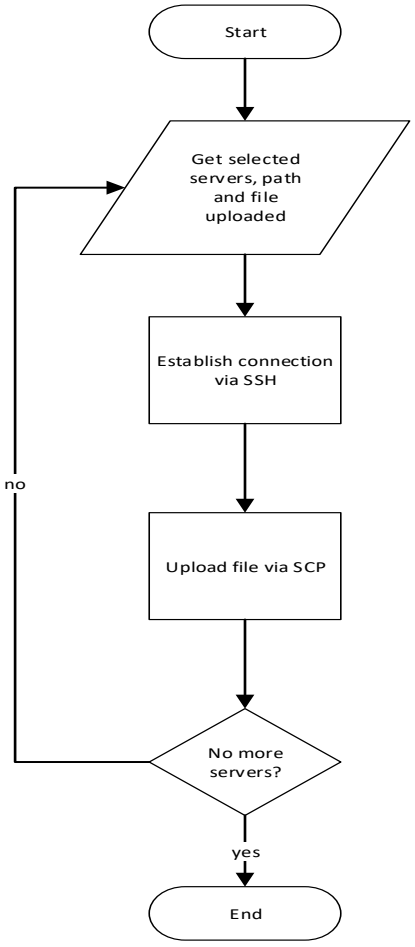

Figure 1.5 Flowchart for uploading file to multiple servers

Figure 1.5 displays the idea of this approach, System start with getting all data by user: path, file, and servers then establish a connection through SSH to the server then upload the file to the path mentioned before then looping this step through all servers.

2.5 Automation Systems when installing services in new server

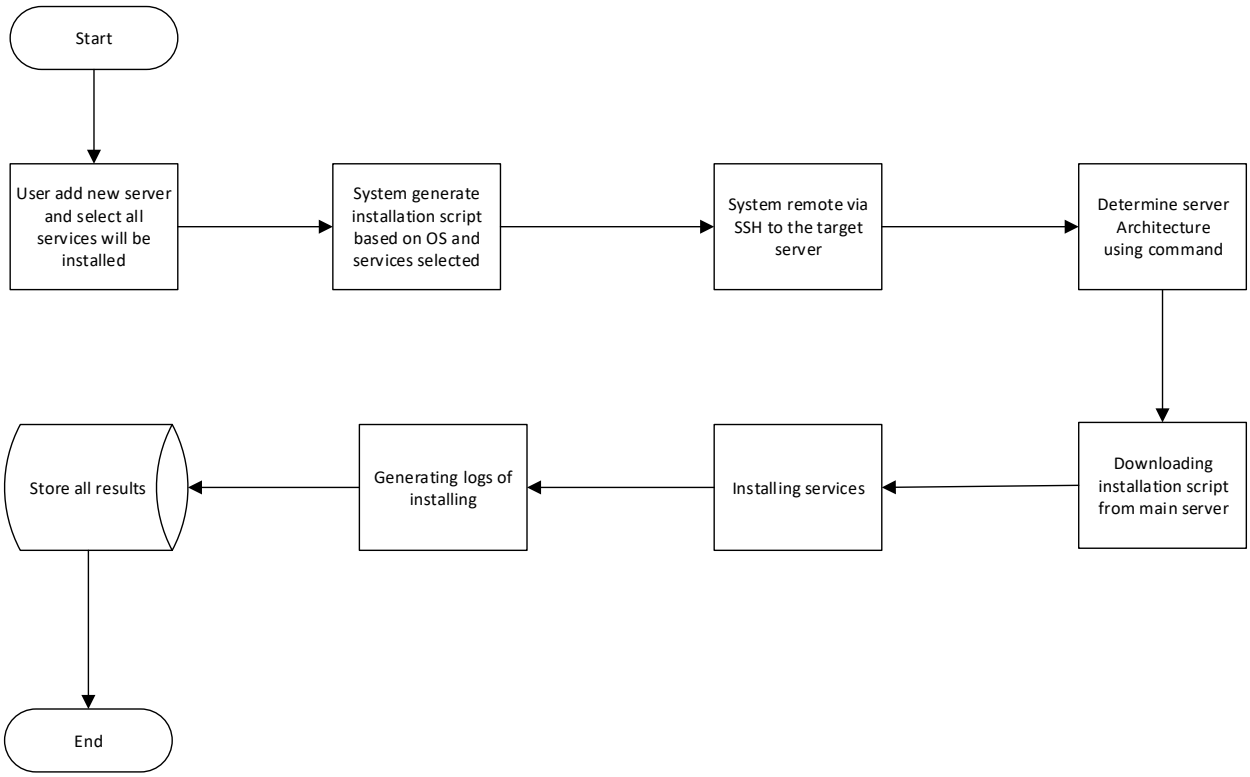

Figure 1.6 Flowchart for Automation systems when installing services in new server

Figure 1.6 displays flowchart for automatic installing all services in new server, after user added new server and defined which services will be activated then System automatically generate new temporary file which contains automatic installation based on OS and services selected by user then remote that new server through SSH protocol then using Wget command in Linux, the system will download temporary file which were generated then performance installation services then store all screen logs into temporary file then using curl commands in Linux, send that file to main server as results.

These are some methods used to determine and getting information to build automation:

a. Determine operating system in Linux
   To determine which operating system, System get information from a built-in file which located in /etc/lsb-release, example result on Centos and Ubuntu:

```
root@centos:~# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.2 LTS"
```

```
[root@ubuntu ~]# cat /etc/lsb-release
LSB_VERSION=base-4.0-amd64:base-4.0-noarch:core-4.0-amd64:core-4.0-
noarch:graphics-4.0-amd64:graphics-4.0-noarch:printing-4.0-
amd64:printing-4.0-noarch
```

in some cases, there are some versions of Centos which doesn't have that file, so System also will take information from another file located in /etc/redhat-release, this is an example results when grabbing content inside that file:

```
[root@centos ~]# cat /etc/redhat-release
CentOS release 6.8 (Final)
```

b. Determine architecture

Determining architecture of the operating system can use the built-in command in Linux "uname -i", that command will display architecture of OS, this is an example of the result:

```
[root@centos ~]# uname -i
x86_64
```

System need to understand which architecture used in server to pointing correctly to which RPM in Centos and install needed files on Ubuntu based on architecture

2.6 Summary

After knowing the methods to monitoring services and resources, then the key to making it automatic is using Crontab utilities and set schedule every X seconds to turn the system work automatically doing the task from checking until repair. Further, when uploading the file to multiple servers, the system depends on SSH protocol then iterate through all servers selected by users then execute a command to each server. Since all communication between server using SSH then SCP is the suitable protocol to provide uploading file to the server and to handle multiple servers, the system will do looping through all servers and upload to the path. Using all of these methods, building the system to provide solutions for those issues are highly possible.

Chapter 3 Designing and programming the system

3.1 Requirements

There is some software shall be installed on the main server to running all of this system, these are software and operating system required:

a. Apache
   Apache as web server to running web application because the system will be built based on the web
b. MySQL
   Database is needed to store all data, the system uses MySQL
c. Crontabs
   Crontabs will schedule scripts or commands to run periodically
d. Centos
   The main server uses Centos as the operating system.
e. PHP
   PHP should be installed on the main server to run the system and handle server scripting.

When all requirements are fulfilled, then the next part is designing database.

3.2 Databases

Databases and tables are designed based on system requirements. the following tables are used on the system:

Table report_installation

This table store information about logs of installation new services

Table 3.1 table for reports of installation

Column Names	Data Types	Description
installation_id	Int(AI)	Auto increment
server_host	Varchar(30)	Hostname of server
logs	Text	Logs of installation
date	date	Date of installation

Table report_resources

This table store information about reports of resources used by all servers.

Table 3.2 table for reports of resources

Column Names	Data Types	Description
report_resource_id	Int(AI)	Auto increment
server_ip	Varchar(19)	IP Address of server
uptime	Varchar(35)	Uptime of server
cpu	Int(3)	Cpu usage
ram	Varchar(35)	Total of RAM
ram_utilised	Int(3)	Ram usage
disk	Text	Total of disk
disk_utilised	Int(3)	Disk usage
date	Date	Date of check
time	Time	Time of check

Table report_services

This table store information about reports of services.

Table 3.3 table for reports of services

Column Names	Data Types	Description
report_service_id	Int(AI)	Auto increment
server_ip	Varchar(19)	IP Address of server
service_name	Varchar(35)	Name of service
date	Date	Date of report
time	Time	Time of report
status	enum('ONLINE', 'OFFLINE')	Status of server
description	text	Full logs of reports

Table servers

This table store information about servers

Table 3.4 table for servers

Column Names	Data Types	Description
server_id	Int(AI)	Auto Increment
server_name	Varchar(30)	Name of server
server_host	Varchar(30)	Hostname of server
server_ip	Varchar(19)	IP Address of server
server_port	Int(6)	SSH Port of server
server_os	Enum('centos', 'ubuntu')	Server operating system
username	Varchar(210)	SSH username
password	Varchar(255)	SSH password
platform	Varchar(25)	Platform of server
api_key	Varchar(255)	Api for platform
api_pass	Varchar(255)	Api for platform
api_host	Varchar(60)	Api for platform
status	Enum('0', '1')	Status of server

Table server_service

This table store information about services which installed on servers

Table 3.5 table for server services

Column Names	Data Types	Description
server_ip	Varchar(19)	IP Address of server
service_name	Varchar(35)	Name of service
service_port	Varchar(60)	Port of service

Table services

This table store information about all available services

Table 3.6 table for services

Column Names	Data Types	Description
service_name	Varchar(35)	Name of service

process_name	Varchar(35)	Process name of service
cmd_restart	Varchar(255)	Command to restart
cmd_start	Varchar(255)	Command to start
cmd_stop	Varchar(255)	Command to stop
cmd_check	Varchar(255)	Command to check
check_method	Enum('1', '2', '3')	Check method
server_os	Enum('centos', 'ubuntu')	Operating system
status	Enum('0', '1')	Status of service

Table users

This table store information about users

Table 3.7 table for users

Column Names	Data Types	Description
username	Varchar(16)	Username
password	Text	Password
salt	Text	Salt for password
name	Varchar(30)	Name of User
email	Varchar(35)	Email address
session	Text	Session code
last_login	Date	Last login date
level	Enum('centos', 'ubuntu')	Level of users

3.3 Scripting

3.3.1 Monitoring Resources

3.3.1.1 Child server or the target server

Because of child server using Linux hence script for checking resources created using bash programming, so the system doesn't require any additional software to install.

```
#!/bin/bash
server_ip="http://MAIN-SERVER-URL/get.php"
if [ -f /etc/lsb-release ]; then
       os_name=$(lsb_release -s -d)
       df -h | awk '$NF=="/"{printf "%s",$5}'
else
```

```
	os_name=`cat /etc/redhat-release`
fi

ip=$(ip addr | grep 'state UP' -A2 | tail -n1 | awk '{print $2}' | cut -f1  -d'/')
if [[ $ip =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]]; then
	IP_ADDR=$ip
else
	ip2=$(ip addr | grep 'inet' -A2 | tail -n1 | awk '{print $2}' | cut -f1  -d'/')
	if [[ $ip2 =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]]; then
		IP_ADDR=$ip2
	else
		IP_ADDR="IP_ADDRESS_NOT_FOUND"
	fi
fi

ram_total=$(free -m | awk 'NR==2{printf "%s/%sMB (%.2f%%)\n", $3,$2,$3*100/$2 }')
uptime=$(awk
'{printf("%d:%02d:%02d:%02d\n",($1/60/60/24),($1/60/60%24),($1/60%60),($1%60))}'
/proc/uptime)
disk_total=$(df -P -T -B 1k | grep '^/')
disk_usage=$(df -h | awk '$NF=="/"{printf "%s",$5}')
cpu_usage=$(echo $[100-$(vmstat 1 2|tail -1|awk '{print $15}')])

DATA="ip_addr=$IP_ADDR|server_os=$os_name|ram=$ram_total|uptime=$uptime|disk=$disk_total
|disk_utilised=$disk_usage|cpu=$cpu_usage"
echo "data=$DATA" | curl -m 50 -k -s -d @- $server_ip
```

3.3.1.2 Main server

The main server where database hosted will catch all data which send by child server and store all data to the database. Because the main server using apache then the script written in PHP

```
$data=$_post['data'];
	$explode=explode('|',$data);
	for($i=0;$i<count($explode);$i++){
		$explodeSeparator=explode('=',$explode[$i]);
```

```
            $dataResources[$explodeSeparator[0]]=$explodeSeparator[1];
      }
$ip_address=$dataResources['ip_addr'];
$uptime=$dataResources['uptime'];
$disk=$dataResources['disk'];
$disk_utilised=str_replace('%','',$dataResources['disk_utilised']);
$cpu=$dataResources['cpu'];
preg_match('@ ((.*?)%)@',  $dataResources['ram'],$info);
preg_match('@(.*?) @',  $dataResources['ram'],$info2);
$ram_utilised=number_format(str_replace('(','',$info[2]),0);
$ram=trim($info2[1]);
if (!filter_var($ip_address, FILTER_VALIDATE_IP) === false) {
      $saveReport= mysqli_query($conn,"insert into `report_resource`
values('','$ip_address','$uptime','$cpu','$ram','$ram_utilised','$disk','$disk_utilised'
,'".date('Y-m-d')."','".date('H:i:s')."') ");
}
```

3.3.2 Monitoring Services

3.3.2.1 Monitoring services by port

Monitoring services by port written in PHP language, the main idea is the system check each port registered in database and looping to all servers.

```
function checkPort($server,$port){
   $socket = @fsockopen($server, $port, $errorNo, $errorStr, 3);
   if(!$socket){
            return false;
      }else{
            return true;
     }
}

if($services['check_method']=='1'){
if(strstr($server_service['service_port'],',')){
      $service_port_single=trim(explode(',',$server_service['service_port'])[0]);
```

```
}else{
	$service_port_single=$server_service['service_port'];
}
	$check_port=checkPort($servers['server_ip'],$service_port_single);
	if($check_port){
		//save to database with status online
	}else{
		//save to database with status offline
	}
}
```

3.3.2.2 Monitoring services through SSH

Monitoring services through SSH also written in PHP language, the system will do checking servers every 5 seconds and store whatever the results to the database.

```
if($services['check_method']=='2'){
$dataServer=mysqli_fetch_array(mysqli_query($conn,"select * from `servers` where server_ip='".$servers['server_ip']."' "));
$execute_command=str_replace('[service-name]',$services['process_name'],$services['cmd_check']);
$VCurl= New VCurl;
$VCurl->setCurl();
$VCurl->enablePost(true);
$VCurl->dataPost("execute_command=".$execute_command."&server_os=".$dataServer['server_os']."&server_user=".$dataServer['username']."&server_host=".$dataServer['server_ip']."&server_port=".$dataServer['server_port']."&server_pass=".decPass($dataServer['password']) );
$output= $VCurl->goCurl("http://cp-sg2.serverip.co/vba/panel/cmd.php");
if(strstr($output[0],$services['process_name'])){
	//success
}else{
	//failed
	//execute command to repair the service
$VCurl->enablePost(true);
$VCurl->dataPost("execute_command=".$services['cmd_restart']."&server_os=".$dataServer['server_
```

```
os']."&server_user=".$dataServer['username']."&server_host=".$dataServer['server_ip']."&
server_port=".$dataServer['server_port']."&server_pass=".decPass($dataServer['password']
) );
$output= $VCurl->goCurl("http://cp-sg2.serverip.co/vba/panel/cmd.php");
            if(strstr($output[0],$services['process_name'])){
                  //success
            }else{
                  //failed
            }
}
}
```

That is part of scripts to check service through SSH protocol, if the service is down the system will try to repair the services and if the services still down even after repair then the system will save the status as offline to the database and also store the logs of the target servers, this will help systems administrators to troubleshoot the servers

3.3.3 Send command to multiple servers

One input text and checkbox to each server, those are input fields needed by the systems after that to make the system lighter and faster, the system will use Jquery to handle all requests and send all requests to PHP script as server side programming.

This is Jquery script used to looping through all servers and send the data to PHP script

```
(function($) {
  var ajaxQueue = $({});
  $.ajaxQueue = function(ajaxOpts) {
       var oldComplete = ajaxOpts.complete;
       ajaxQueue.queue(function(next) {
         ajaxOpts.complete = function() {
              if (oldComplete) oldComplete.apply(this, arguments);
              next();
         };
         $.ajax(ajaxOpts);
       });
```

```
  };
})(jQuery);


var m = 1;
$('input[name^="server_host"]:checked').each(function() {
$.ajaxQueue({
        url: '?page=send_commands',
        data: { server_host: $(this).val(),cmd:$("#cmd").val(),action:"cmd"},
        type: "POST",
        success: function(data){
              $('.report').append(data);
        }
});
m+=1;
});
```

This is PHP script which receives the data from Jquery script and forwards the data to the forwarder server then display the results.

```
if(isset($_POST['cmd'])){
$cmd=($_POST['cmd']);
$server_host=$_POST['server_host'];
$dataServer=mysqli_fetch_array(mysqli_query($conn,"select * from `servers` where
server_host='$server_host' "));
$VCurl= New VCurl;
$VCurl->setCurl();
$VCurl->enablePost(true);
$VCurl-
>dataPost("server_user=".$dataServer['username']."&server_host=".$dataServer['server_hos
t']."&server_port=".$dataServer['server_port']."&server_pass=".decPass($dataServer['pass
word'])."&cmd=".$cmd);
$output= $VCurl->goCurl("http://LINK-TO-FORWARDER-SERVER./script.php");
echo $output[0];
}
```

VCurl above is a class made by using Curl function and using POST method to send data to the forwarder server. Further, the forwarder server receives the data then remote to the target server and execute command then return with the results, below is the script to do it.

```
if(!$con = ssh2_connect($server_host, $server_port)){
	echo "Cannot connect";
}else{
	if(!ssh2_auth_password($con, $server_user, $server_pass)) {
		echo 'EL';
	} else {
		$sendCommands= cmd($con,$_POST['cmd']);
		echo $sendCommands;
	}
}
}

function cmd($con,$cmd){
$stream = ssh2_exec($con, $cmd );
$errorStream = ssh2_fetch_stream($stream, SSH2_STREAM_STDERR);
stream_set_blocking($errorStream, true);
stream_set_blocking($stream, true);

  $output = "";
  while( $buf = fread($stream,128) ){
	$output .= $buf;
  }

  $error = "";
  while( $buff = fread($errorStream,128) ){
	$error .= $buff;
  }

fclose($errorStream);
fclose($stream);
```

```
	if($output){
		return $output;
	}else{
		return $error;
	}
}
```

The last part of the script above to check the output if the output is empty then return with a $error message to help the system administrator to troubleshoot the server.

3.3.4 Upload file to multiple servers

The system uses Jquery to handle client side, receive all requests and forward it to PHP scripting then looping through all servers. In detail, jQuery getting the value of path (where file will be saved), server detail and file then send those data to the PHP script then PHP script will use SSH protocol to connect to the target server and use SCP protocol to upload file and the last looping through all servers using jQuery to make the system lighter and faster.

This is the PHP script used to connect through SSH protocol and upload file using SCP protocol:

```
if(!$con = ssh2_connect($server_host, $server_port)){
	$output = "Cannot connect to server";
}else{
	if(!ssh2_auth_password($con, $server_user, $server_pass)) {
		$output = 'EL';
	} else {
		$upload = ssh2_scp_send($con, $localfile, $path, 0644);
		if($upload){
			$output = "File has been uploaded in '$path' ";
		}else{
			$output = "cannot upload files !";
		}
	}}
```

Before upload to the target server, the script move the file to temporary folder then upload that local file to the server.

3.4 User Interface

3.4.1 Servers

Figure 3.1 displays form input for server, and on this form, the system will collect information about what kind of services will be running on the server.

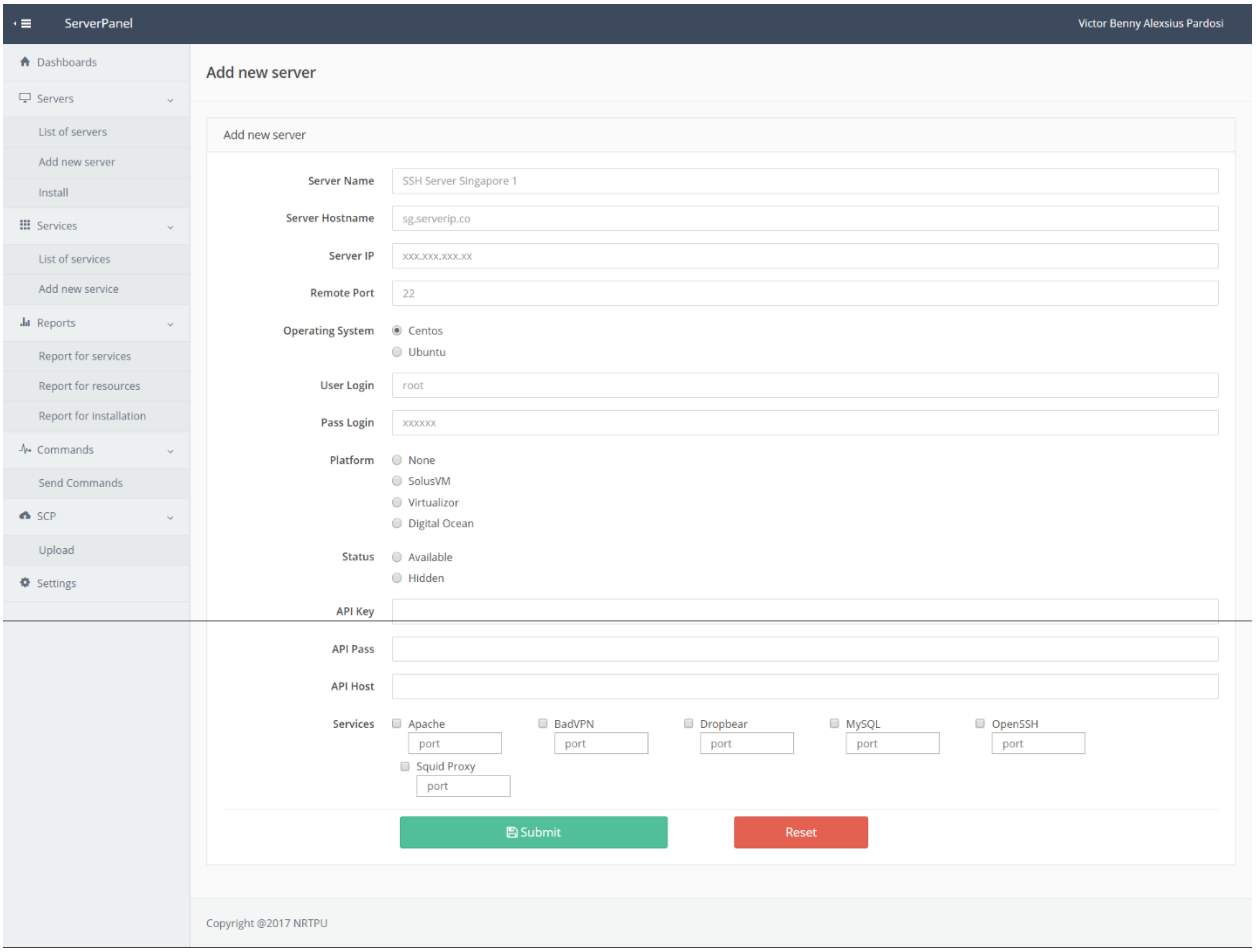

Figure 3.1 Form servers

Even if the server still does not have any services which are selected, the users should fill in every service which will be installed on the server.

3.4.2 Services

Form for add new service designed as Figure 3.1 to get information for both operating systems Centos and Ubuntu, and the users need to specify process name of each operating system.

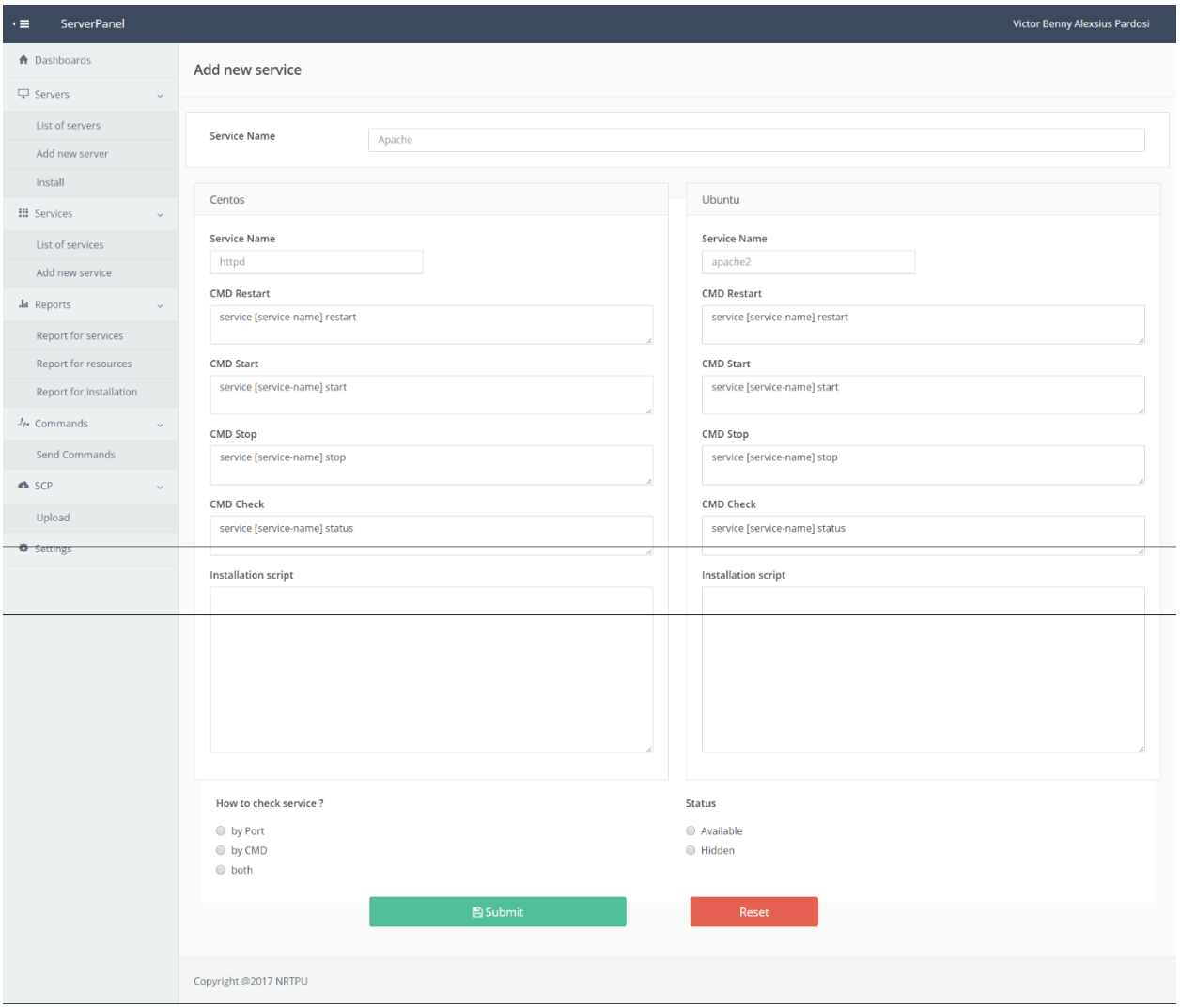

Figure 3.2 Form services

Because for both operating systems the process name is different, further the users need to fill in all commands which will work to start, restart and check the services, so when the system wants to check the services through SSH protocol, it has the knowledge to execute that command.

Also on this form the users need to specify script for automatic installation, this is example script to automatic installation for Apache on Centos:

```
##INSTALL APACHE
echo "Installing Apache" >> $LOG 2>&1
```

```
yum install httpd >> $LOG 2>&1

echo "Installing PHP and Extensions" >> $LOG 2>&1

yum install php php-mysql -y >> $LOG 2>&1

yum install php-mysql php-gd php-imap php-ldap php-mbstring php-odbc php-
pear php-xml php-xmlrpc -y >> $LOG 2>&1

echo "Set Auto Start" >> $LOG 2>&1

chkconfig httpd on

echo "Restart Apache" >> $LOG 2>&1

sudo service httpd restart   >> $LOG 2>&1

echo "Disable default port" >> $LOG 2>&1

sed -i '/Listen 80/s/^/#/' /etc/httpd/conf/httpd.conf;

echo "Open correct port" >> $LOG 2>&1

echo "Listen 80" >>  /etc/httpd/conf/httpd.conf;

echo "installation of Apache is complete" >> $LOG 2>&1
```

that script will automatically install Apache on the server and open the correct port that depends on which port defined by the users when adding a server. If there are many services will be installed then the systems will pack similar like the above script into a single file then save it to a temporary file.

3.4.4 Commands

Form commands designed as simple as possible to easily send commands to multiple servers as displays on figure 3.3, all requests handled by Ajax and jQuery making this form lighter and faster.

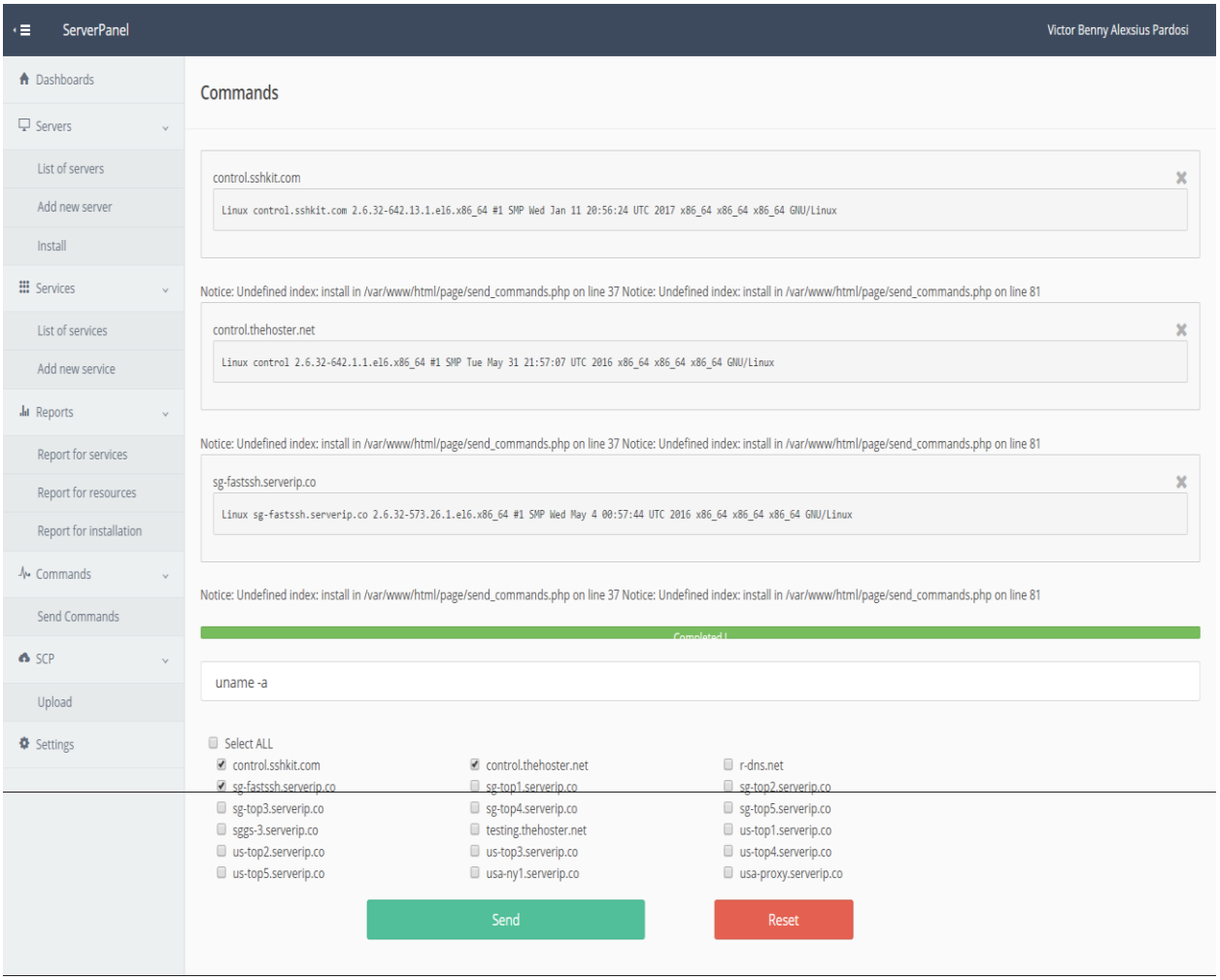

Figure 3.3 Send Multiple commands

Even though there are hundreds of servers there, the system can process it pretty fast, although the speed of processing also depends on the forwarder server.

3.4.5 Uploads

The most important on form uploads is to specify the path for the location of the file, because if the path is not defined the systems cannot upload the file.

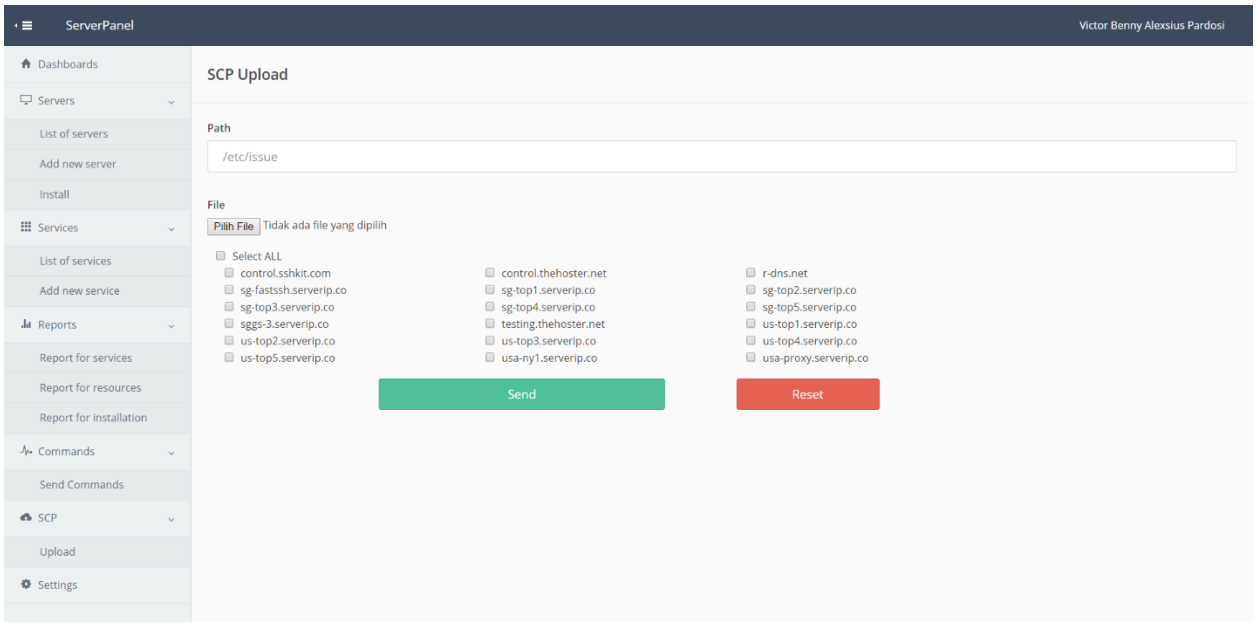

Figure 3.4 Send Multiple commands

On Figure 3.4 displays there are many checkboxes, those checkboxes are examples to shows the system able to handle upload to multiple servers.

3.5 Summary

All methods in the chapter to 2 applied on this chapter to make script and design interface of systems. User interface built with HTML and CSS, client-side scripting handled with Ajax jQuery, on the server side processing with PHP language, monitoring, and automation on the Linux server created using bash programming and the last, the database using MySQL.

Chapter 4 System testing

4.1 Automatically install services

Before trying to install services, the systems should have data about the target server, so the users need to input the server details.

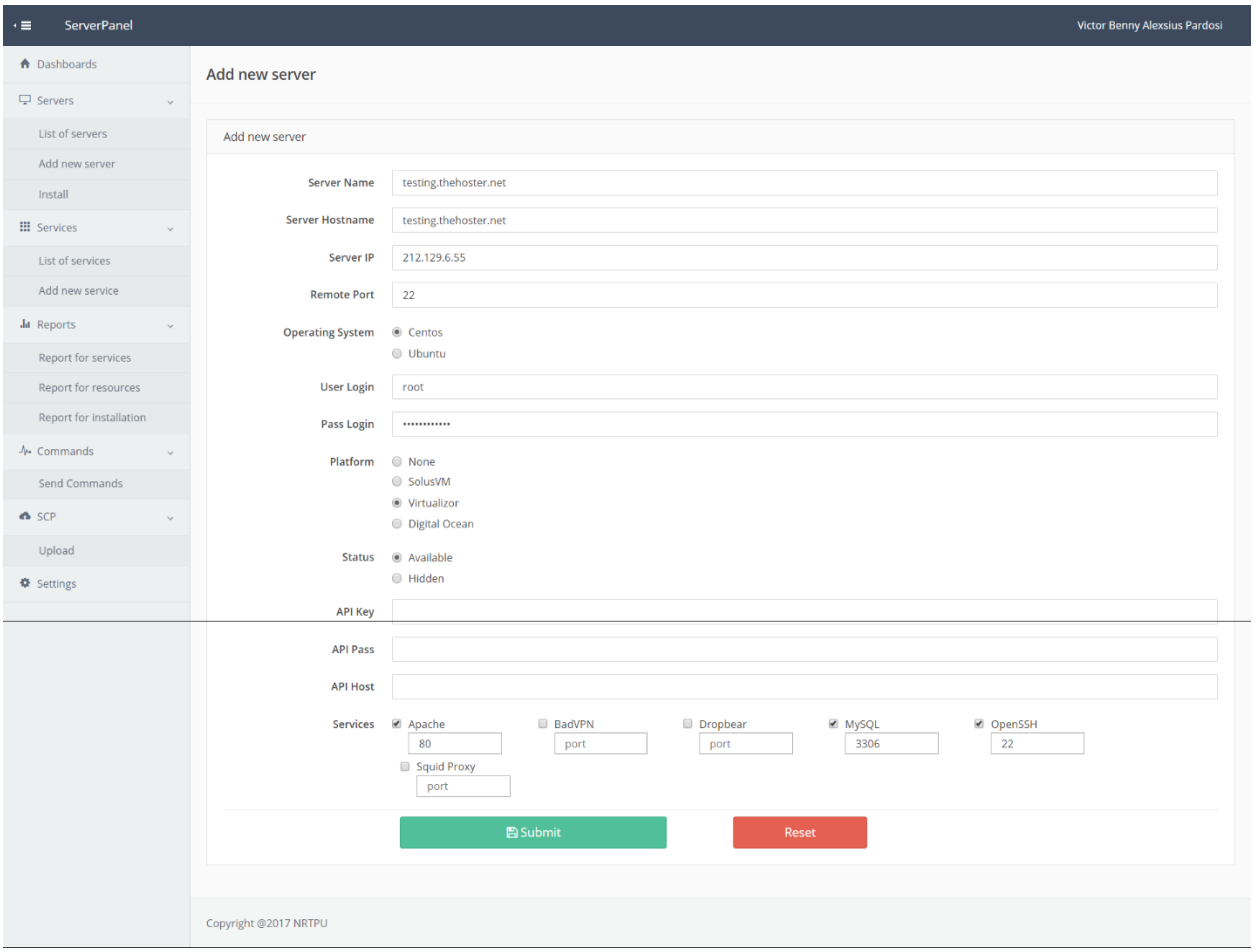

Figure 4.1 Add new server and select services

Figure 4.1 displays all fields to store new server and the user also need to select which server will be activated, if those services need port then specify which port it is. on figure 4.1 the user selected Apache and MySQL as services and defined each port.

The next step is going to install menu and select which server will be installed.

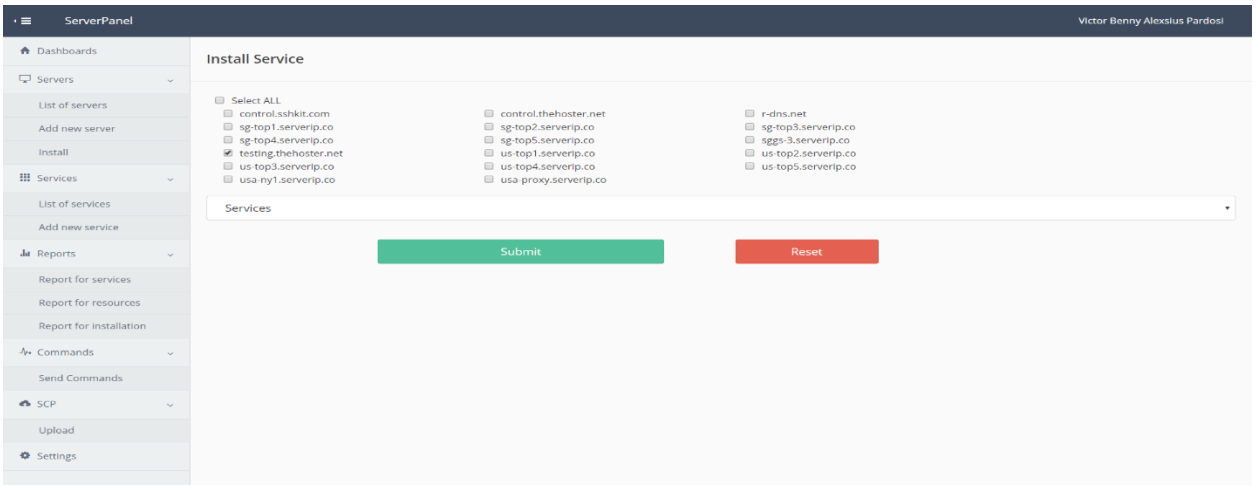

Figure 4.2 Install service

After selecting the server name then change the select box to services as displayed on Figure 4.2 then click submit to start the installation process.

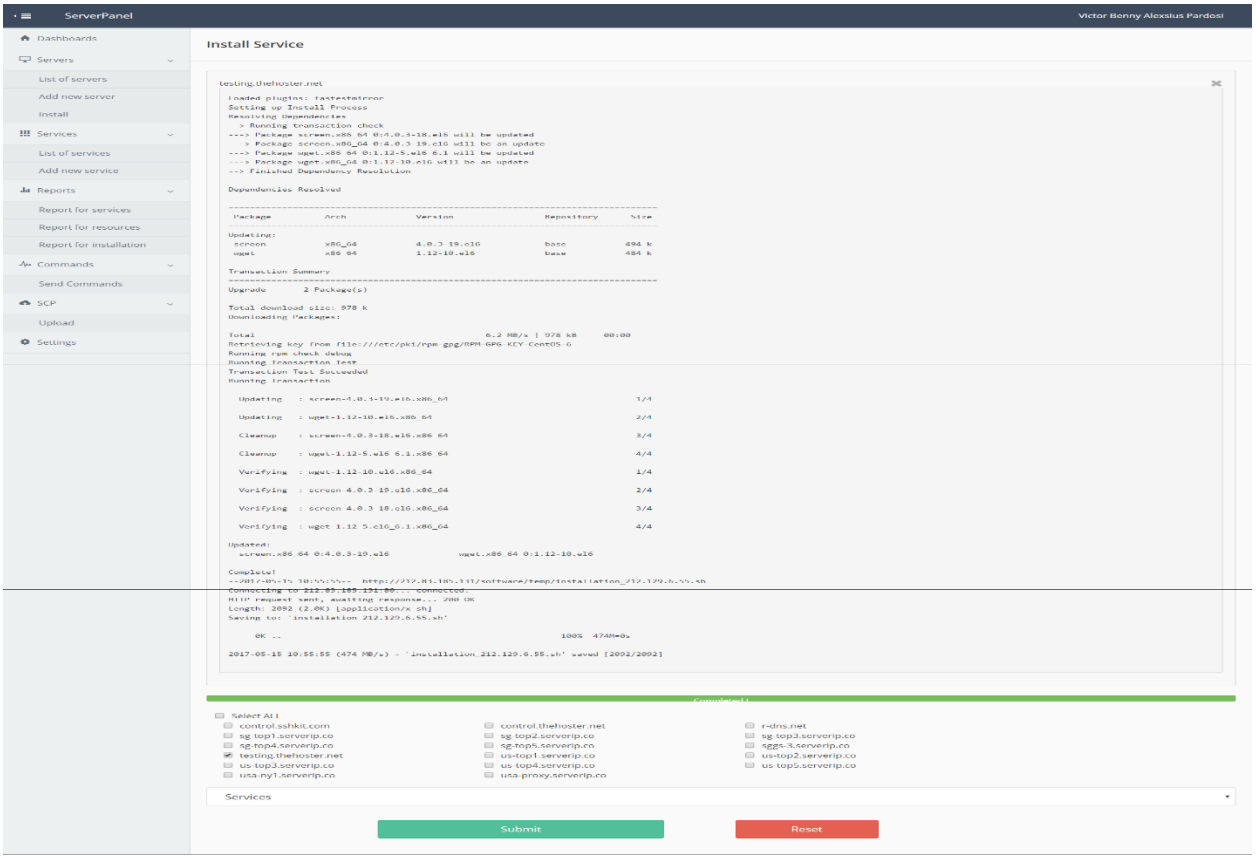

Figure 4.3 Process of installation services

The output on figure 4.3 displays results from the server target, it means the system initiate the process of installation. The system creates a temporary file based on services selected then the system remotes

the destination server through SSH protocol and downloads that file from the main server then run that script. It will create 2 logs file, one log with all details installation and the other are summary of installation which will send to the main server as results.

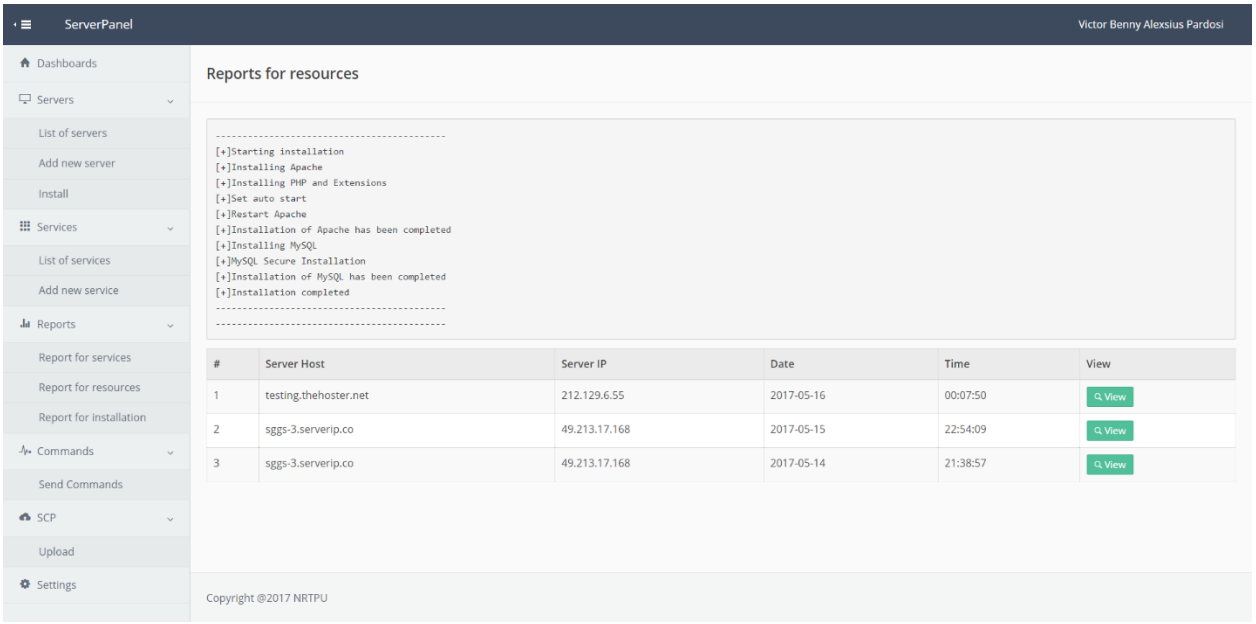

Figure 4.4 Results of installation service

Next step is to check the results of installation, because of some factors such as server network speeds and server input/output speeds then the result might show after 2 minutes or more. The results will show on Report of installation menu as displays on figure 4.4

To ensure the system working perfectly, the users can remote through SSH or check a report of services to verify the services. Figure 4.5 displays verify through SSH and execute command “netstat -tulpn” to check services.

```
Using username "root".
Server refused our key
[root@testing ~]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State       PID/Program name
tcp        0      0 0.0.0.0:22                  0.0.0.0:*                   LISTEN      572/sshd
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN      624/sendmail
tcp        0      0 0.0.0.0:3306                0.0.0.0:*                   LISTEN      1371/mysqld
tcp        0      0 0.0.0.0:80                  0.0.0.0:*                   LISTEN      1559/httpd
tcp        0      0 :::22                       :::*                        LISTEN      572/sshd
[root@testing ~]#
```

Figure 4.5 Verify installation of services

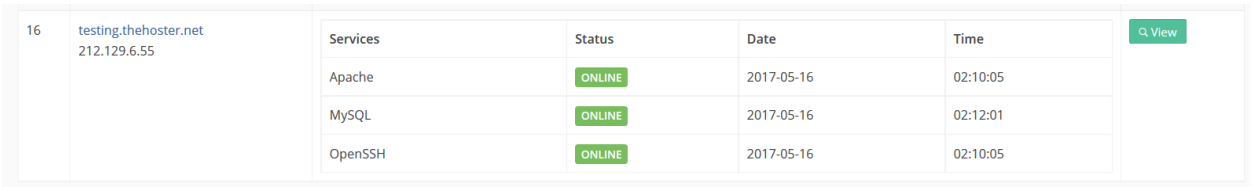

Figure 4.6 Services status

All services which already selected to installed is appear in figure 4.5 and in figure 4.6 displayed all services are online, that mean automatic installation is working perfectly.

4.2 Automatically install monitoring

The steps almost like automatically install services, the difference is when select to install choose to install monitoring on install menu as displayed in figure 4.6

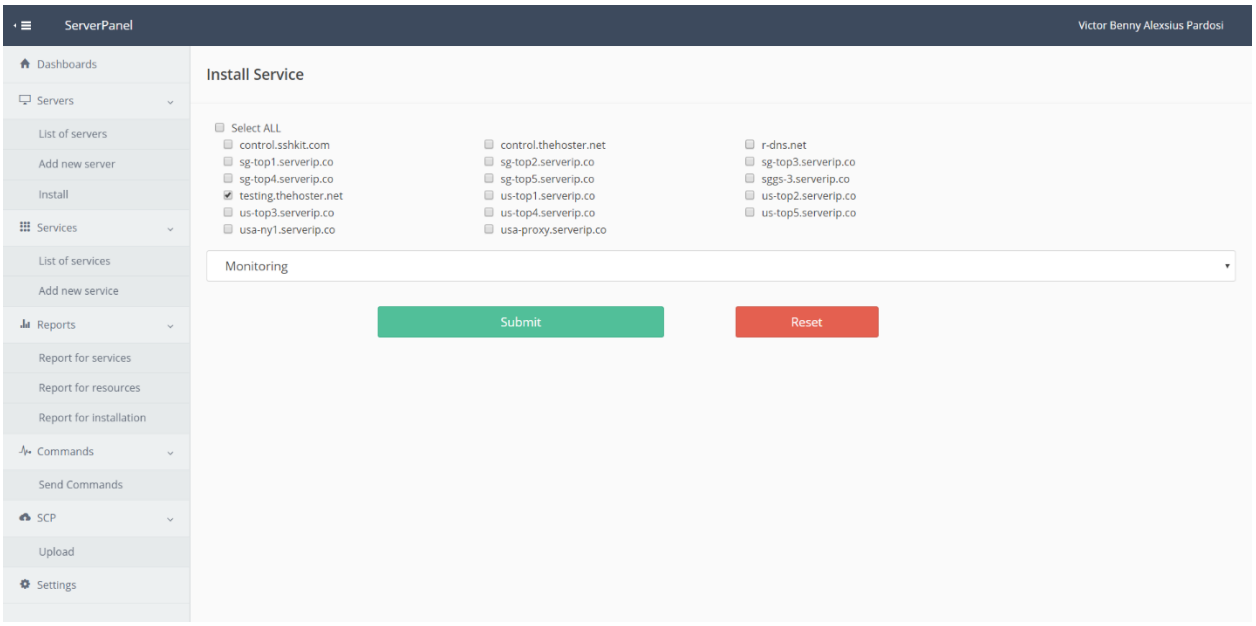

Figure 4.6 User interface for install monitoring system

Installing the monitoring system making the system able to collect information about resource of servers. After selecting which server will be installed then click submit to start installation progress.

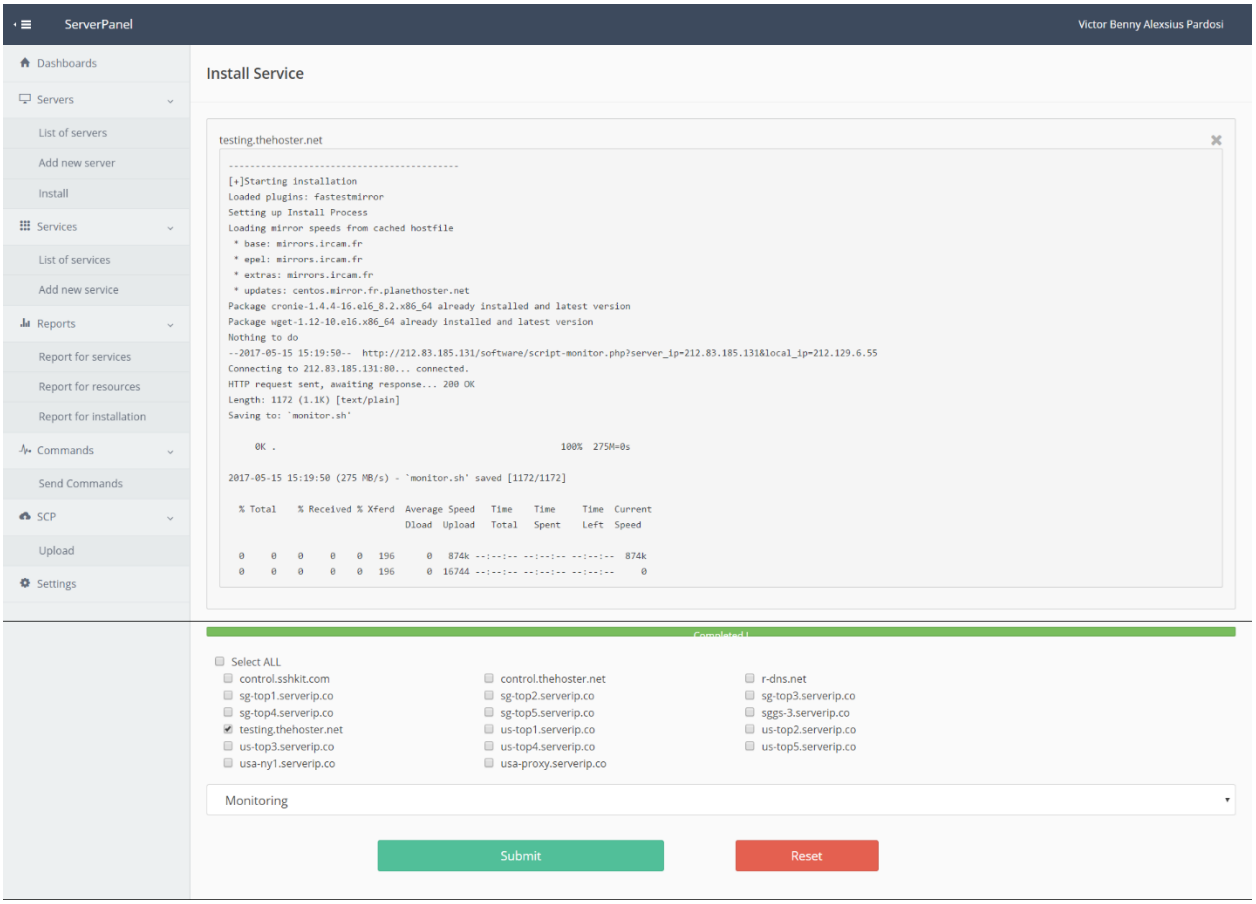

Figure 4.7 Response of server when installing monitoring system

Figure 4.7 displayed response of server which telling downloading script from the main server has successfully and it is automatically set crontab every 5 minutes to execute the script monitoring, every time the script executed the server will send the results of resources to the main servers.

14	control.thehoster.net	62.210.73.66	342:04:21:43	3%	6734/7858MB	2%	2017-05-16	02:23:45
15	sggs-3.serverip.co	49.213.17.168	1:15:38:17	0%	61/1024MB	3%	2017-05-15	21:35:07
16	testing.thehoster.net	212.129.6.55	0:02:13:14	0%	289/1024MB	2%	2017-05-16	02:20:02

Figure 4.8 Report of resources

Users can view resources information on Report of resources menu as displayed in figure 4.8

4.3 Send commands to multiple servers

To test this system which sending commands to all servers selected, the user needs to define the commands, in this example the writer using command “uname -a” which print name and version Linux operating systems.

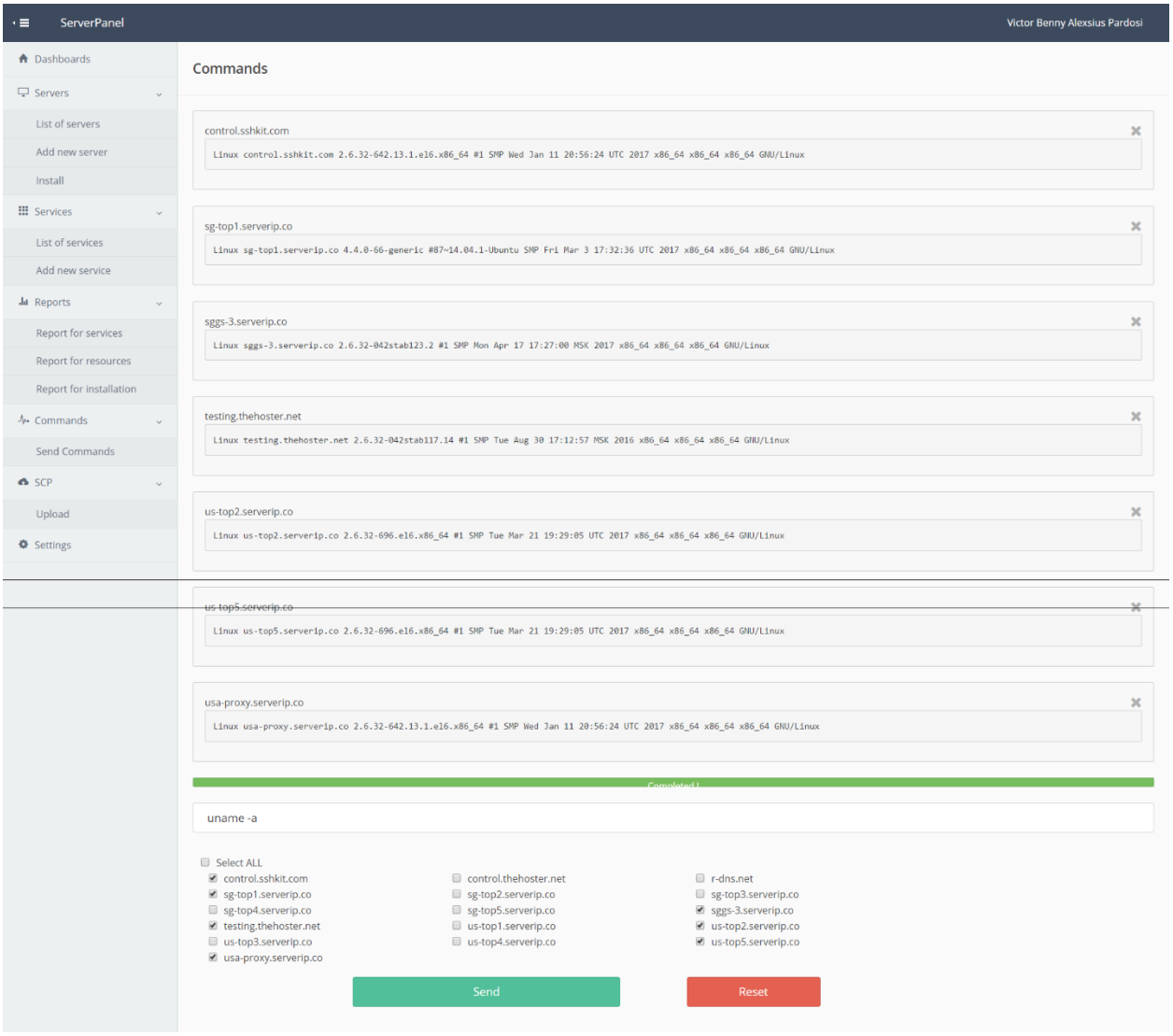

Figure 4.9 Sending commands to multiple servers

Figure 4.9 displays responses of servers after command “uname -a” has been executed. The outputs are showing Linux version and architecture of the machine; it is evident the system working perfectly. Using this system making the systems administrators tasks are easier, example when flushing all iptables or execute any commands which need to send to many servers.

4.4 Upload file to multiple servers

The purpose of this system to help systems administrators to upload the file to multiple servers without remote the server one by one. When testing this system, the writer creates a sample file which will be uploaded to the server and define a path for the location where the file will be stored and select the servers as displayed in figure 4.10

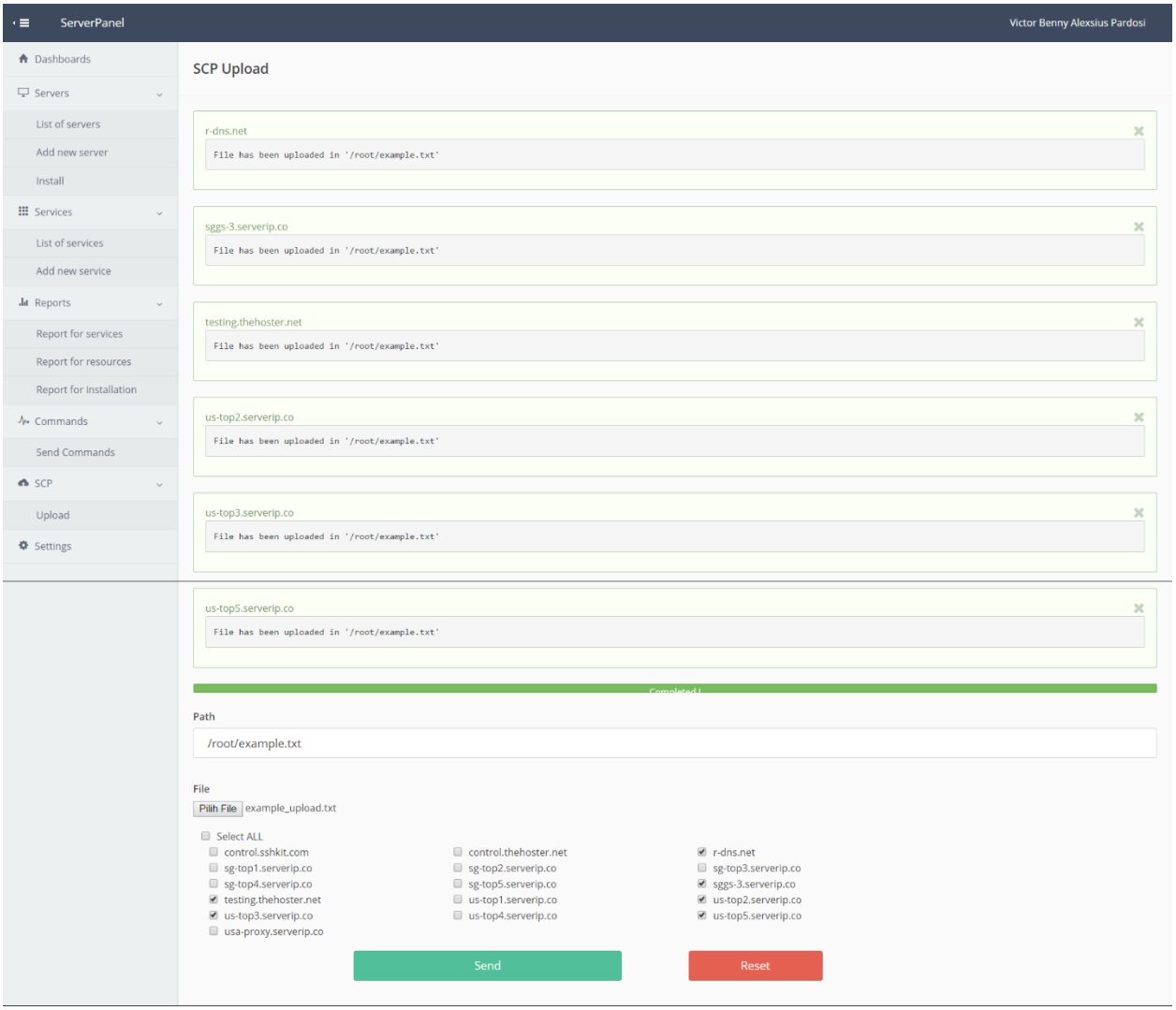

Figure 4.10 Uploading file to multiple servers

Further, in figure 4.10 also displays responses of each server which indicated the file had been uploaded successfully.

To ensure this system working perfectly, the writer using the send commands menu to check the file, whether it exists or not as displayed in the figure 4.11

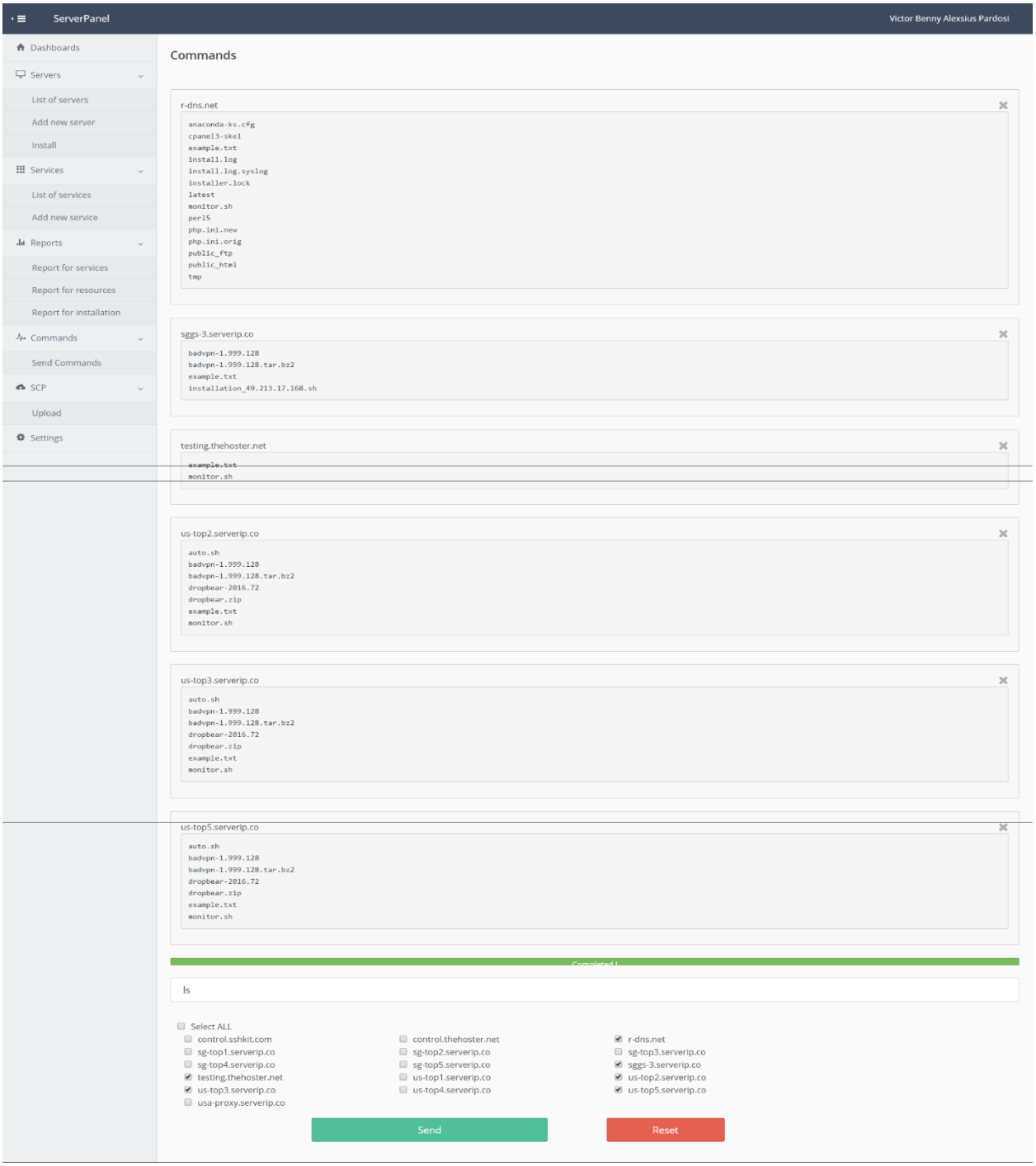

Figure 4.11 Checking the file which already uploaded

In figure 4.11 displays each server has the file ‘example.txt’, that mean the systems successfully uploaded the file and it is working perfectly.

4.5 Reports for resources and services

On this part, there is no input because the system monitoring for resources and services are automatically, so only the reports are shown here.

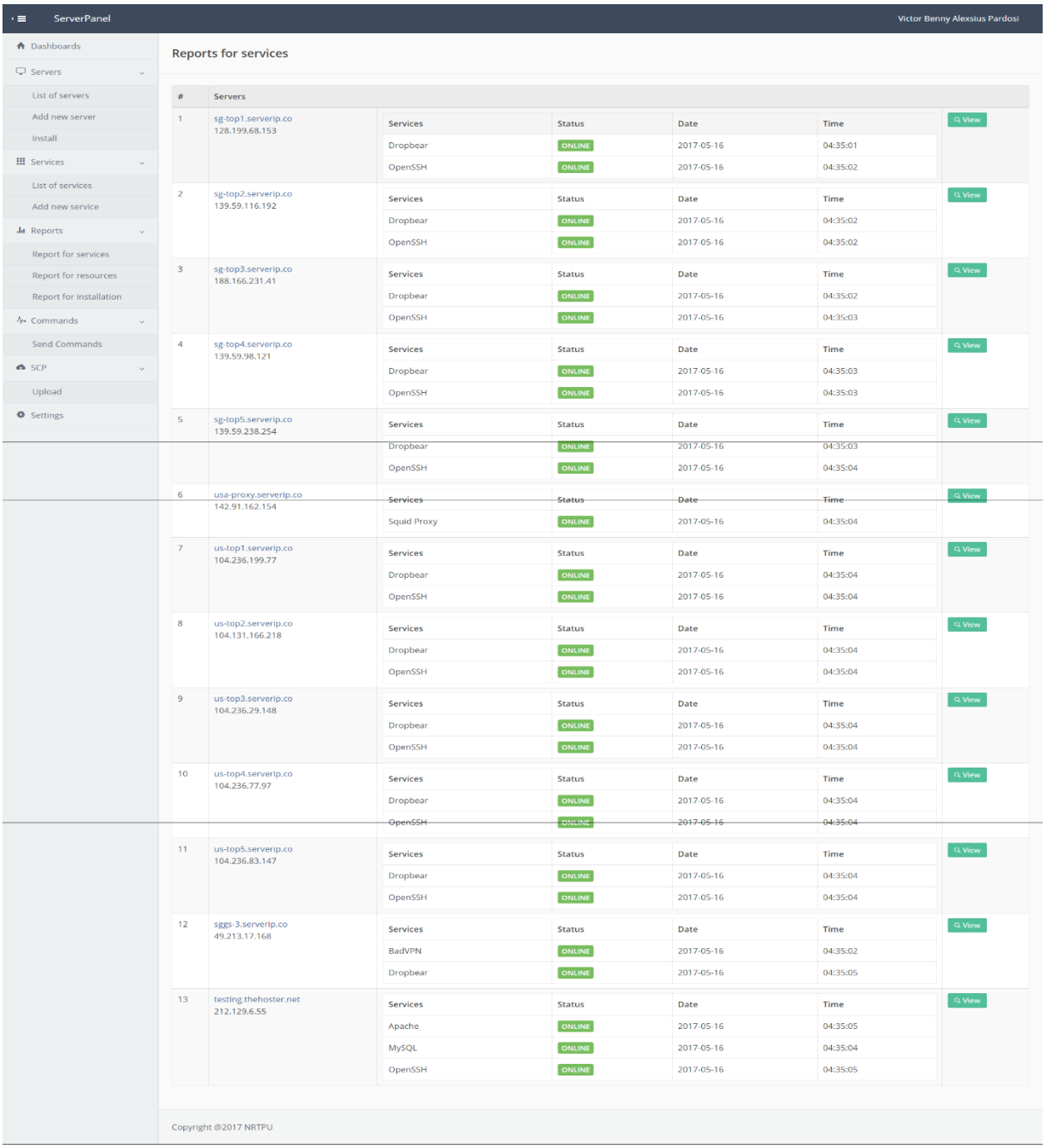

Figure 4.12 Reports for services

Reports of services are useful to check which services are offline as displayed in figure 4.12

Next, reports for resources which displaying resources usage by servers, this report also important to ensure there are no resources over usage.

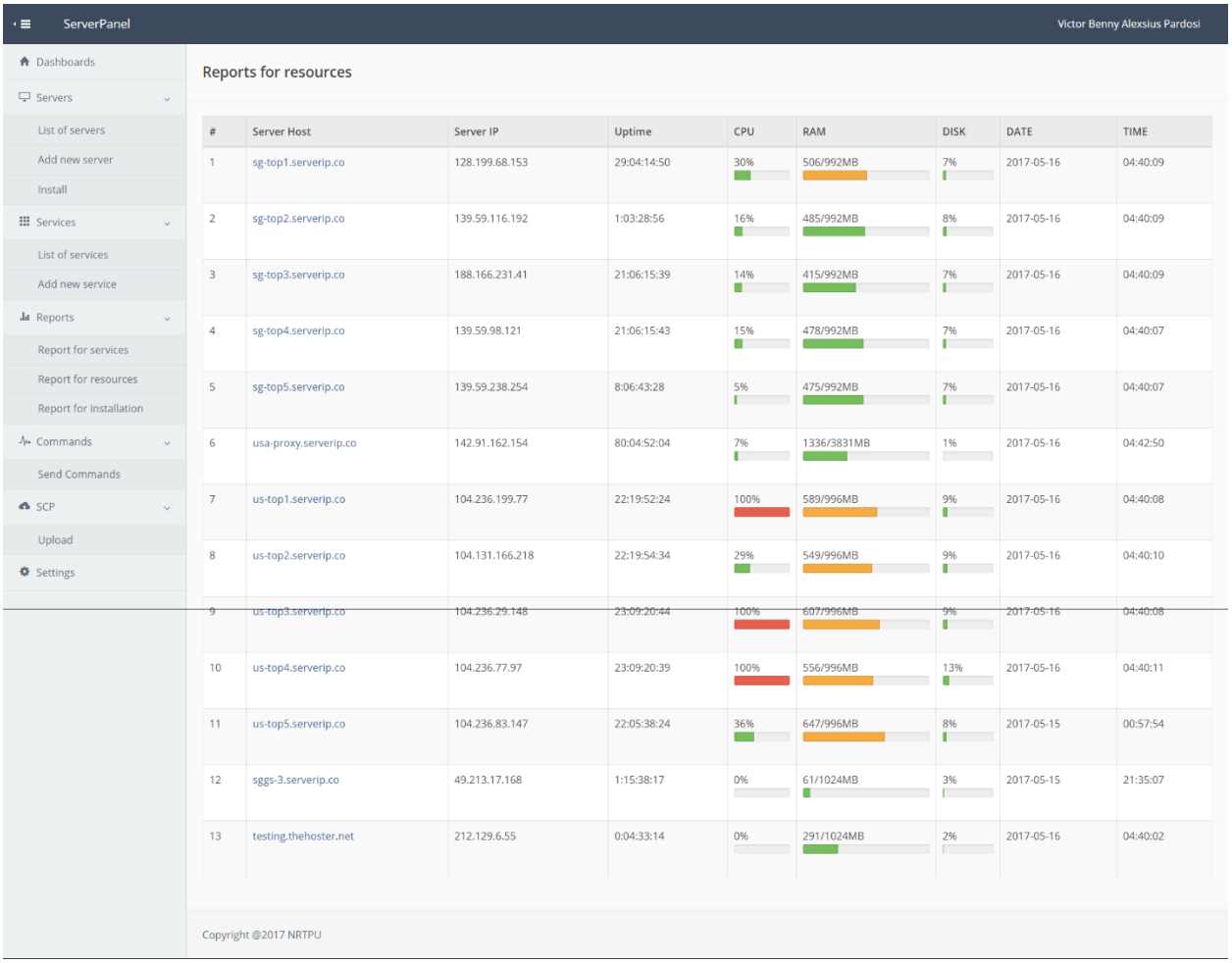

Figure 4.13 Reports for resources

The systems automatically collecting information for both reports, for the resources information collects from the client server which sending every 5 minutes. And for the services, the systems check the port which already defined but if the port cannot be accessed by public then the system remote to the client server to get service status.

4.6 Summary

All tests on the systems working perfectly that mean the methods and scripts are correctly implemented, also the system provides much information for services and resources from client servers which will be useful for the systems administrators.

Chapter 5 Financial management, resource efficiency and resource saving

Using SWOT analysis to define strengths and weaknesses of the project. Further, analyzing the competitors to finding competitiveness the software in the market. Also, Market segmentation needed to reveal potential consumers. And the last using Quad technology (Quality Advisor) to finding more opportunity for further development.

5.1 Potential consumers of the research

Market segmentation is used to identify of the potential consumers to determine the target market of product consumers. Segmentation is performed based on Segmentation criteria. The most important criteria of segmentation are the number of clients and level of consumer's income. Proceeding from the chosen criteria of Segmentation, the map of Segmentation (Table 5.1) was constructed.

Table 5.1 – Map of services market Segmentation

		Level of consumer income		
		Low	Average	High
number of consumer clients	More than 1000			
	From 100 to 1000			
	Not more than100			

On the presented map of Segmentation light color allocated target groups of consumer's numbers of clients more than 1000, with average and high-income levels.

The average and high level of the income of potential consumers is caused by needs to have a computer or more, an accompanying equipment and Internet access. In spite of the fact that today the computer can be got for a price (approximately from 400 dollars), besides acquisition for a comfortable using internet speed should be quite high, so this connection can cost to users dearly.

The number of clients target audience is also an important factor of market Segmentation. Group of companies, hospitals, schools, banks, governments and so many others have huge need to use that for classification and safety.

Dark color on the map Segmentation marks the market segments that are attractive in the future. Attracting customers with fewer customers due to the fact that companies are constantly growing and evolving, and the using of such application becomes more necessary for their work, research and safety.

5.2 The analysis of competitive technical solutions from the perspective of resource efficiency and resource saving

To analyze the competitive projects, there was a most important application which is the accuracy and. In order to compare our product spheres of software development, technical and economic performance benchmarks. After highlighting criteria, there were evaluated our product and competitive solutions, the results of which are presented in table 5.2

Table 5.2 – Scorecard for comparing competitive technical solutions

Criteria for evaluation	Criterion weight	Scores			Competitiveness		
		S_f	S_{k1}	S_{k2}	C_f	C_{k1}	C_{k2}
1	2	3	4	5	6	7	8
Technical criteria for assessing resource efficiency							
1 Easy to use (meets consumers requirements)	0.1	5	4	4	0.5	0.4	0.4
2 Interface quality	0.1	3	4	5	0.3	0.4	0.5
3 Additional feature	0.2	5	4	3	1	0.8	0.6
4 Easy operation	0.1	5	4	4	0.5	0.4	0.5
Economic criteria of an efficiency assessment							
1 Product competitiveness	0.1	4	4	3	0.4	0.5	0.3
2 Price	0.2	4	3	5	0.8	0.6	1
3 After-sales service	0.2	5	4	5	1	0.8	1
Total	1				4.5	3.9	4.3

Analyzing the data of evolution table, we can say that the developed product has certain competitive advantages. Namely, it has a set of convenient, simple and understandable for untrained person functions, which is critical for the users of such applications. Also, it has a set of additional feature (such as creating reminders, initialization calls, sending e-mail). In addition, the application works with several languages, which is not considered in the analysis of competitors.

5.3 SWOT – analysis

SWOT analysis of the developed project is applied to define such strategy of its development at which its strengths for the achievement of goals will be used as much as possible, and at the same time will be taken into account weakness and vulnerabilities of both, research project and market conditions in which it will be implemented.

Received information, as a result of carrying out SWOT analysis, is then used to make a conscious choice concerning areas of a wide range of action which considers competitive and commercial advantages of the project and increases the probability of achievement of its purposes and tasks.

The result of the SWOT-analysis conducted by the research project is the matrix SWOT, presented in table 5.3.

Table 5.3 - SWOT matrix

	Strengths of the research project: S1. Simple and intuitive interface. S2. Excellent data filtering S3. Interest in improving application, investors availability. S4. Professional developers.	Weaknesses of the research project: W1. No ability to export reports W2. Lack of using language choice. W3. A bit expensive
Opportunities: O1. Attracting demand for the product. O2. Rising cost of competitive development. O3. Extension of the functional.	1.1. Professional developers, interest of customers in improvement of the application and presence of investors will allow to expand functionality of the application. 1.2. Low cost will allow to attract demand for a product. 1.3. The simple and convenient functionality in use will allow us to attract users in the conditions of Rising cost of competitive developments.	1.1 Extension of the application functional allows you to add features export and import new data. 1.2 Raising demand for the product will contribute to add some languages.
Threats: T1. Lack of demand. T2. Decrease of such applications popularity. T3. Appearance of new competitive developments.	1.1 Sample and easy-to-use functionality helps to overcome the threat of lack of demand and declining popularity. 1.2 Professional developers, the availability of investment and interest In the promotion of applications allow us to compete in the market.	1.1 Lack of data filtering function in the application can lead to a decrease in demand. 1.2 The lack of automatics reminders can be a problem in case of new competitive developments.

Thus, thanks to the carried-out analysis the following strategy of project development was defined:

- The introduction of data filtering and automatics reminders;
- Increase the number of available languages.

The received resulted were considered by drawing up the list of the works performed within the engineering project.

5.4 Quad technology

For an assessment of prospects of the created decision in the market, the QuaD technology (Quality Advisor) is used. This technology defines prospects of its development by means of allocation and an assessment of indicators of quality and commercial potential of the projects.

To assess the quality of the development, it was selected indicators that are most important and decisive for the software of classification. If such factors as reliability and functionality are important to any user application, the quality of the interface and ease-to-use are importance for the software of classification, mainly due to the features of the user's interaction with information systems.

Table 5.4 presents evolution results of the quality and commercial potential of the project and competitive solutions within the assigned parameters.

Table 5.4 – Scorecard of comparing competitive technical solutions

Criteria for evaluation	Criteria weight	Score	Maximum scores	Relative value (3/4)	Average value (5x2)
1	2	3	4	5	6
Indicators for assessing the quality of development					
1. Reliability	0.1	80	100	0.8	0.08
2. Easy to use	0.05	90	100	0.9	0.045
3. Interface quality	0.05	60	100	0.6	0.033
4. Functional capabilities	0.15	80	100	0.8	0.12
5. Additional features	0.05	90	100	0.9	0.045
Indicators for assessing the commercial potential of development					
1. competitiveness	0.1	100	100	1	0.1
2. Market entry level	0.15	70	100	0.7	0.105
3. Prospects of the market	0.2	90	100	0.9	0.18
4. Price	0.15	80	100	0.8	0.12
Total	1				0.828/82.8%

From the results, the scorecard shows that the weighted average value of the quality and availability of scientific development is 82.8%, which corresponds to prospective development by QuaD technology.

5.5 FAST – Analysis

The object of FAST – Analysis is software of classification. The main function of the developed application is to give users of image processing to monitor and facilitate the work of the company with its customers. In addition to the main features, the application has a user interface, simplifying the work process, as well as several additional functions.

- The main internal functions provided by the application, are: adding, editing, filtering data, and work with automatic reminders.
- As an additional function application provides opportunities to work in multiple languages and initialization calls.

Table 5.5 show all the processes used in the application description of the functions and their ranks. In the future, this classification can be utilized for optimization of a development project, as

to improve the efficiency of the process by reducing the value of the property and the preservation of the necessary quality. It is necessary, first of all, to pay attention to the additional functions. If in the case of savings, we will not use this feature – it will not have a significant effect on the functionality of the entire project.

Table 5.5 – Classification of the functions performed by the project of study

Process name	Function	Rank of function		
		Main	Basic	Additional
Send command so multiple servers	Execute command to multiple servers remotely	X		
Upload file to multiple servers	Fast upload to multiple servers	X		
Monitoring services and resources	Generating reports for each server whether online or offline	X		
Automatic install services and monitor system	Automatically install new system to all client servers	X		
Adding and editing data	Allowing users to create, edit data of their customers		X	
Filtering Data	Allowing the user to filter data		X	
Multi-users	They system allow multi-user at the same time			X

For an assessment of the importance of functions, the method of arrangement of priorities is used. Settlement and expert determination of the importance of each function are the basis for this approach. The first step in assessing the significance of the functions is to build the adjacency matrix (Table 5.6)

Table 5.6 – The adjacency matrix

	Function1	Function1	Function1	Function1	Function1	Function1
Function1	=	>	>	>	>	>
Function2	<	=	=	>	>	>
Function3	<	=	=	>	>	>
Function4	<	<	<	=	>	>
Function5	<	<	<	>	=	
Function6	<	<	<	<	<	=

The second stage of evaluating the importance of the functions is to convert the adjacency matrix into a matrix of quantitative relations functions (Table 5.7)

Table 5.7 – Matrix quantitative relations functions

	Function1	Function1	Function1	Function1	Function1	Function1	Total	

Function1	1	1.5	1.5	1.5	1.5	1.5	8.5	0.22
Function2	0.5	1	1	1.5	1.5	1.5	7	0.19
Function3	0.5	1	1	1.5	1.5	1.5	7	0.19
Function4	0.5	0.5	0.5	1	1.5	1.5	5.5	0.15
Function5	0.5	0.5	0.5	1.5	1		5.5	0.15
Function6	0.5	0.5	0.5	0.5	0.5	1	3.5	0.1
Amount							37	1

Note: 0.5 at 《<》 ;1.5 at 《>》 ; 1 at 《=》

From the result of FAST – analysis suggests that the least significant in the application is a function of 6 – initialization call. This result is explained by the fact that this feature is optional, extends the functionality of the application. Compared with the basic functions of our application, the significance was lower. However, the introduction of call initialization function justified the involvement of a significant number of users. Therefore, in the presence of human, time and material resources to add all initialization function can lead to increased profits by attracting users. In the case of lack of necessary resources introduction of the function will be unjustified and expensive for the customer.

Final results of the chapter of financial management, resource efficiency, and resource conservation research project are presented in Table 5.8.

Table 5.8 – Results of the analysis and evaluation of the project

Type of analysis	Obtained result
1 Identify of potential consumers of research results.	Potential consumers of image processing are companies with numbers of clients more than 1000, with average and high-income levels. In future, there are planning to attract more companies.
2 Analysis of competitive technical solutions	Coefficient of image processing is 4.5, Coefficient of image processing – 3.9 and 4.3
3 SWOT-analysis	There were defined the strategy of Coefficient development: - The introduction of data filtering and automatic reminders; - Increase a number of available languages.
4 Analysis of the prospects of the development of technology QuaD	The weighted average value of the quality and availability of scientific development is 82.8%, which corresponds to prospective development by QuaD technology.
5 FAST-analysis	The least significant function in the application multi-user, because it is

	optional and extends the functionality of the application.

5.6 Cost to build the system

It is important to estimate how much money will be used when building the system, the main cost for this project mostly for hardware. Table 5.9 displays the cost estimates, without including the internet cost and electricity.

Table 5.9 – Cost to build the system

Name	Quantity	Price (RUB)
Supermicro Processor Intel Xeon E3-1280v5 @ 3.7 Ghz (4 cores - 8 threads) 16GB RAM DDR3 ECC HDD SATA 2x 2TB Enterprise Raid Setting: Raid 1	1	105.829
21.5" Monitor Samsung S22E391H	1	8000
Centos operating system	1	0
Apache	1	0
MySQL	1	0
TOTAL		**113.829**

5.6 Time of schedules

The time of schedule is presented in the form of a diagram

Table 5.10 Time of schedules

	Action	Percentage	Total of Percentage	Duration of works in days									
				March		April				May			
				10	10	10	10	10	10	10	10	10	10
1	Submit topic	5	5										
2	Writing about introduction	5	10										
3	Collecting data and writing chapter 1	10	20										

4	Finding methods of system	15	35										
5	Software implementation	15	50										
6	Testing software	3	53										
7	Writing chapter 2	10	63										
8	Writing chapter 3	10	73										
9	Writing chapter 4	10	83										
10	Writing chapter 5	5	88										
11	Writing chapter 6	5	93										
12	Finishing the software and re-testing all systems	5	98										
13	Writing conclusion	2	100										

Conclusion

Using built-in commands in Linux to collects information of server through SSH as network protocol which providing secure access and encrypted communication between servers. Make the purpose of this work achieved that creates a Web-Based Application to help the system administrator to handle multiple servers with ability; to send a command to multiple servers, upload file to multiple servers, provide automation system to install new service, automation to check all services and server resource to prevent over usage in server. With this application, the system administrator doesn't need to remote server one by one anymore.

The system created using PHP as server side programming and HTML as client side programming. Also, the system use Jquery to handle the client request, making the system communication between servers faster. In the client or child server, the system using Bash language to produce periodic reports and send it to the main server.

After testing this Web Application, the writer very confident that goal has been archived. Sending a command or upload a file to multiple servers in just a few clicks and the time spent to process it is decreased significantly compared to usual way which connect to the server one by one. Also, as there are monitoring services and resources which ensuring server online that will help the system administrator when checking and make decision using all data collected.

This system can be used all organizations which having many Linux servers. Also, the system still allows customizing the commands and installation script causing self-made application also possible to include for all automation.

In the end, nothing is perfect, so the writer with pleasure allows the future researcher to continue and to use this work to bring it to the higher level.

References

[1] Secure Shell. Retrieved from https://en.wikipedia.org/wiki/Secure_Shell

[2] Apache vs Nginx: Practical Considerations. Retrieved April 04, 2017, from https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations

[3] PHP vs Ruby vs Python: The Three Programming Languages in a Nutshell. Retrieved April 06, 2017, from https://1stwebdesigner.com/php-vs-ruby-vs-python/

[4] MySQL. Retrieved April 06, 2017, from https://www.techopedia.com/definition/3498/mysql

[5] Linux Inside. Retrieved April 06, 2017, from https://openlibra.com/en/book/download/linux-inside

[6] Karen R. DeMay - Ontario County Office Of Employee Safety. Retrieved 2004-10-08

[7] World Health Organization. 2001 - Occupational health A manual for primary health care workers

[8] Fire Safety and Fire Extinguishers. Retrieved from http://www.ilpi.com/safety/extinguishers.html

[9] Fire Extinguishers. Retrieved from https://en.wikipedia.org/wiki/Fire_extinguisher

[10] Ward, B. 2014. How Linux Works, 2nd Edition: What Every Superuser Should Know, 392 pp

[11] Lucas, M. 2012. SSH Mastery: OpenSSH, PuTTY, Tunnels and Keys, 150 pp.

[12] Cannon, J. 2016. Linux Administration: The Linux Operating System and Command Line Guide for Linux Administrators, 202 pp.

[13] Gimson, M. 2017. Linux Command Line - from Simple Commands to Advanced Level, 202 pp.

[14] Cannon, J. 2015. Shell Scripting : How to Automate Command Line Tasks Using Bash Scripting and Shell Programming, 98 pp.

[15] Thomas, G. 2016. CentOS Linux Administrator Commands, 796 pp.

[16] Rhamey, C. and Fox, B. 2009. BASH Reference Manual - A GNU Manual, 158 pp.

[17] Collier, M. 2015. Linux Techniques: Programming, System Management and Applications, 212 pp.

[18] Peicevic, A. 2017. Apache HTTP Server introduction: Learn how to configure Apache Web Server in an easy and fun way, 60 pp.