

# EASTERN INTERNATIONAL UNIVERSITY

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

## DEPARTMENT OF COMPUTER NETWORKS AND DATA COMMUNICATIONS

**CSE 420**

**INTERNSHIP PROJECT REPORT**

Integrated System Management: Combining Local Networks and Cloud Platforms

**Students**

**HA QUANG MINH** - 1931220012

**Supervisor**

**DR**. **PHAN VAN VINH**

**Binh Duong, SEPTEMBER , 2025**

EASTERN INTERNATIONAL UNIVERSITY

## SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY



# PROJECT EVALUATION FORM

### General information

* Project title: Integrated System Management: Combining Local Networks and Cloud Platforms
* Student name and ID:
  + **HA QUANG MINH - 1931220012**
* Supervisor name: **Dr. Phan Van Vinh**

### Descriptive comments

* 1. *Presentation (layout, format, style, confidence)*

This report has been written with good layout and format effectively.

* 1. *Content (Originality and contribution)*

The content of this report has satisfied the requirement of the first project.

* 1. *Outcome (Creativity and practice)*

This project has good achievements and expected outcomes.

* 1. *Performance during questions and answers*

During the time of this project, these student have shown there strong interest and good working attitude to complete the requirements of the project.

### Suggestion for improvement

Need to enhance the web interface design and database dessign to get better performance in data visualization.

**Total Score: …/100 Binh Duong, Date:**

**Phan Van Vinh**

# ABSTRACT

This internship project focuses on learning how local network tools and cloud platforms work, and how they can be used together in a small IT system. The main goal was to explore tools like pfSense, Zabbix, and Grafana for basic network setup and monitoring, and compare cloud services such as AWS, Google Cloud, and Microsoft Azure. A simulated client named Athena was used as the base for designing a simple hybrid model. Most of the work was research-based, with some small test environments built for practice. Through this process, I was able to get a better understanding of how system management can be planned, even though no real deployment was done.

# ACKNOWLEDGEMENT

First of all, I’d like to thank my supervisor, Dr. Phan Van Vinh, for his support and guidance throughout this internship. His advice helped me stay on track and understand where to focus my research. I also want to thank the School of Computing and Information Technology at Eastern International University for providing the resources needed to complete this project. Finally, I’m grateful to my friends and classmates who shared ideas and helped me test my small lab setups. This experience was a big step in my learning journey, and I truly appreciate everyone who supported me along the way.

# TABLE OF CONTENTS

Contents

[EASTERN INTERNATIONAL UNIVERSITY 1](#_Toc200027184)

[DEPARTMENT OF COMPUTER NETWORKS AND DATA COMMUNICATIONS 1](#_Toc200027185)

[SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY 2](#_Toc200027186)

[PROJECT EVALUATION FORM 2](#_Toc200027187)

[1. General information 2](#_Toc200027188)

[2. Descriptive comments 2](#_Toc200027189)

[3. Suggestion for improvement 2](#_Toc200027190)

[ABSTRACT 1](#_Toc200027191)

[ACKNOWLEDGEMENT 2](#_Toc200027192)

[TABLE OF CONTENTS 3](#_Toc200027193)

[LIST OF FIGURES 4](#_Toc200027194)

[LIST OF ABBREVIATIONS 5](#_Toc200027195)

[CHAPTER 1 : INTRODUCTION 6](#_Toc200027196)

[1.1 Purpose of the Project 6](#_Toc200027197)

[1.2 Overview of Fuzzing and Sanitizer Technologies 6](#_Toc200027198)

[1.2.1 LLVM libFuzzer 7](#_Toc200027199)

[1.2.2 AddressSanitizer (ASan) 7](#_Toc200027200)

[1.3 Problem Statement 9](#_Toc200027201)

[1.4 Objectives 9](#_Toc200027202)

[1.5 Scope & Limitations 10](#_Toc200027203)

[CHAPTER 2 : RELATED WORK 11](#_Toc200027204)

[2.1 Overview of Fuzzing Techniques 11](#_Toc200027205)

[2.2 LLVM libFuzzer 11](#_Toc200027206)

[2.3 AddressSanitizer (ASan) 12](#_Toc200027207)

[2.4 Other Testing & Sanitizing Tools 13](#_Toc200027208)

[CHAPTER 3: PROJECT OVERVIEW 14](#_Toc200027209)

[3.1 System Architecture 14](#_Toc200027210)

[3.2 Toolchain & Environment Setup 14](#_Toc200027211)

[3.3 Role of libFuzzer and ASan in Workflow 15](#_Toc200027212)

[CHAPTER 4 : METHODOLOGY 16](#_Toc200027213)

[4.1 Environment Setup 16](#_Toc200027214)

[4.2 Fuzzer Configuration & Execution 17](#_Toc200027215)

[4.3 Bug Collection & Classification 17](#_Toc200027216)

[4.4 Effectiveness Evaluation 18](#_Toc200027217)

[CHAPTER 5: IMPLEMENTATION & RESULTS 19](#_Toc200027218)

[5.1 Target Project Selection 19](#_Toc200027219)

[5.2 Fuzzing Campaign Outcomes 20](#_Toc200027220)

[5.3 Asan Memory-Error Reports 21](#_Toc200027221)

[5.4 Comparative Analysis and Manual Testing 22](#_Toc200027222)

[5.5 Meaning of Each Value in the Shadow Byte Legend 22](#_Toc200027223)

[A screenshot of a computer program

Description automatically generated 22](#_Toc200027224)

[5.6 Buffer Overflow Exploitation 23](#_Toc200027225)

[CHAPTER 6: DISCUSSION 28](#_Toc200027226)

[6.1 Key Findings 28](#_Toc200027227)

[6.2 Limitations 28](#_Toc200027228)

[CHAPTER 7: CONCLUSION & FUTURE WORK 29](#_Toc200027229)

[7.1 CONCLUSION 29](#_Toc200027230)

[7.2 FURURE WORK 29](#_Toc200027231)

[REFERENCES 30](#_Toc200027232)

# LIST OF FIGURES

[Figure 1: Introduction Fuzz Testing 6](#_Toc200027233)

[Figure 2: Fuzzing and Sanitizer ( fuzz pedal ) 7](#_Toc200027234)

[Figure 3: Author : Chad Spensky 7](#_Toc200027235)

[Figure 4: AddressSanitizer 8](#_Toc200027236)

[Figure 5: Address Sanitizer Details 8](#_Toc200027237)

[Figure 6: Types of fuzzing 11](#_Toc200027238)

[Figure 7: Fuzzing process 16](#_Toc200027239)

[Figure 8: Heap use after free errors 18](#_Toc200027240)

[Figure 9: Code target.c 19](#_Toc200027241)

[Figure 10: Code fuzzer\_target.cpp 20](#_Toc200027242)

[Figure 11: Code UAF.cpp 20](#_Toc200027243)

[Figure 12: Result of Healp Buffer Overflow 21](#_Toc200027244)

[Figure 13: Result of Stack Buffer Overflow 21](#_Toc200027245)

[Figure 14: Result of Use-After-Free 22](#_Toc200027246)

[Figure 15: Error details 22](#_Toc200027247)

[Figure 16: Crash files 24](#_Toc200027248)

[Figure 17: output pattern string 25](#_Toc200027249)

[Figure 18: gdb occurs 25](#_Toc200027250)

[Figure 19: Stack-buffer-overflow 25](#_Toc200027251)

[Figure 20: Security protections 26](#_Toc200027252)

[Figure 21: Function information 27](#_Toc200027253)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Meaning** |
| AWS | Amazon Web Services |
| GCP | Google Cloud Platform |
| Azure | Microsoft Azure |
| pfSense | An open-source firewall/router software |
| Zabbix | A monitoring tool for IT infrastructure |
| Grafana | A tool for visualizing metrics & logs |
| VM | Virtual Machine |
| LAN | Local Area Network |
| IP | Internet Protocol |
| DNS | Domain Name System |
| NAT | Network Address Translation |
| GUI | Graphical User Interface |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# CHAPTER 1 : INTRODUCTION

## 1.1 Background and Motivation

In modern IT environments, managing both local networks and cloud systems has become increasingly important. Many small to medium-sized businesses are moving from fully on-premise models to hybrid infrastructure, where some services remain local while others are moved to the cloud. This shift brings new challenges in terms of monitoring, configuration, and security.

During this internship, I wanted to explore how different network management tools and cloud platforms could work together. I was especially interested in open-source tools like **pfSense**, **Zabbix**, and **Grafana**, as well as well-known cloud providers like **AWS**, **Google Cloud**, and **Microsoft Azure**. The goal was not to master them, but simply to get familiar with how they work and what they offer.

## 1.2 Problem statement

Many businesses face difficulties in managing their IT systems due to the lack of clear monitoring, automation, and network visibility. Tools are available, but knowing how to choose and combine them effectively is not easy, especially for small teams with limited experience or budget.

This project is based on a simulated client called **Athena**, which represents a medium-sized business. By understanding Athena’s basic needs, the aim is to propose a simple hybrid model that connects local network tools with cloud services in a way that is easy to manage and maintain.

## 1.3 Objectives

The main objectives of this internship project are:

* To understand the basics of local network management using tools like **pfSense**, **Zabbix**, and **Grafana**.
* To learn the core features of popular cloud platforms: **AWS**, **GCP**, and **Azure**.
* To create a small simulation environment using tools like **Cisco Packet Tracer** and **VMware**.
* To propose a basic hybrid system management model that fits the needs of a typical SME (small or medium enterprise) like Athena.

## 1.4 Scope & Limitations

This project is mainly research-based with a few small lab simulations. The focus is on learning how the tools work rather than building a full production-ready system.

* + - **Scope :**
    - Learn and test tools like pfSense, Zabbix, Grafana, Packet Tracer, and some cloud platforms.
    - Build a small test lab using personal computer resources.
    - Analyze basic needs and propose a high-level model.
      * **Limitations:**
      * No access to real infrastructure from a business.
      * Cloud accounts are free-tier, so features are limited.
      * Security and performance testing are not included.

# CHAPTER 2 : TOOLS AND TECHNOLOGIES OVERVIEW

This chapter introduces the main tools and platforms used during the internship project. Since the focus was on learning and testing small setups, each tool was explored at a basic level. The tools were either installed in a virtual environment or used through online free-tier services for testing and observation purposes.

## 2.1 pfSense

**pfSense** is an open-source firewall and router software that can be installed on virtual or physical machines. It is commonly used in small to medium-sized networks for managing routing, NAT, VPN, and firewall rules.

In this project, I installed pfSense on a virtual machine using VMware. I tested simple configurations such as static IP assignment, port forwarding, and basic firewall rule creation. The goal was to understand how pfSense could act as a central control point in a local network.

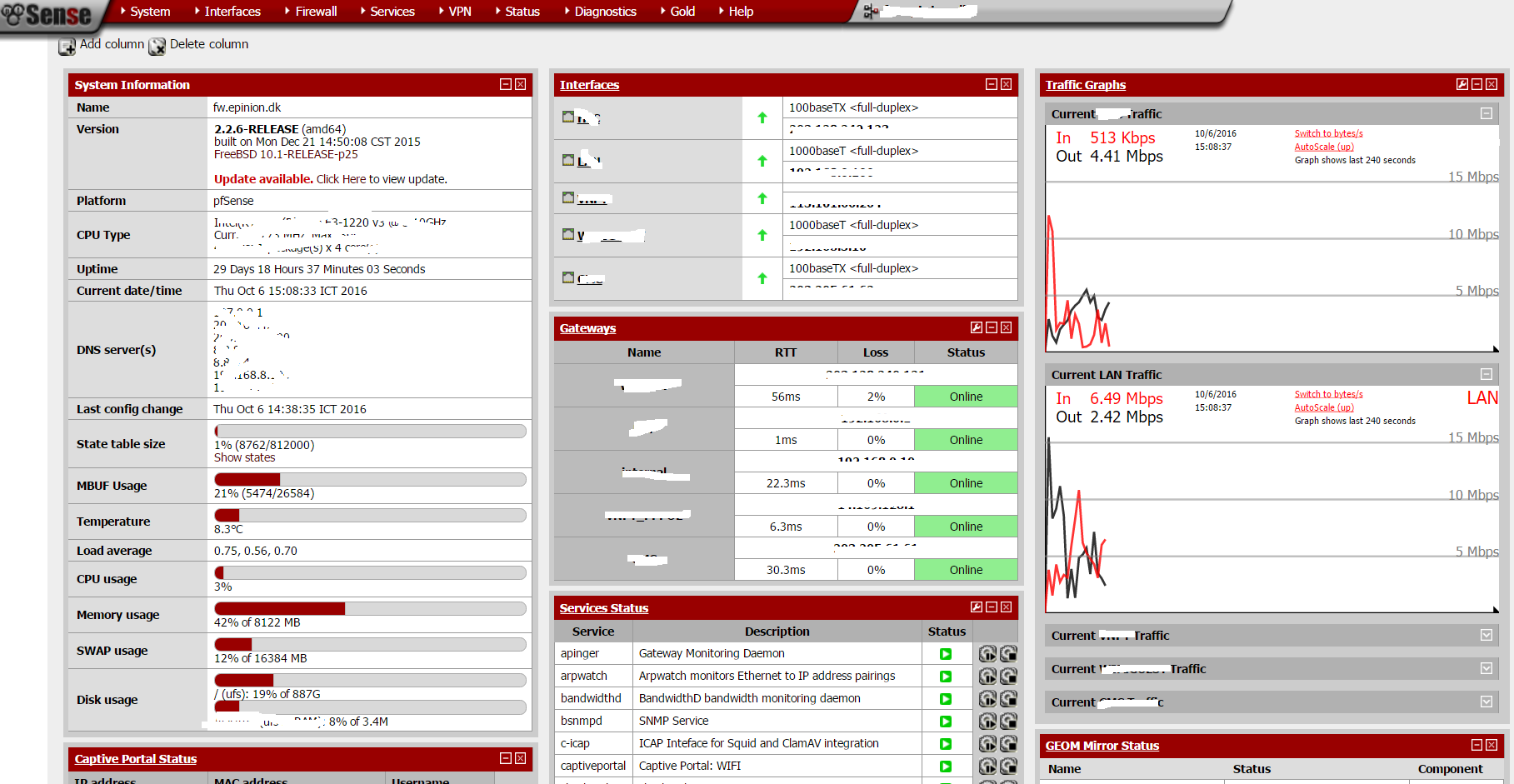


Figure : pfSense Dashboard

## 2.2 Zabbix and Grafana

**Zabbix** is a monitoring tool used for collecting and analyzing system performance data, such as CPU usage, network traffic, and uptime.



Figure : Zabbix Dashboard

**Grafana** is a data visualization platform that integrates with Zabbix and provides more user-friendly dashboards.

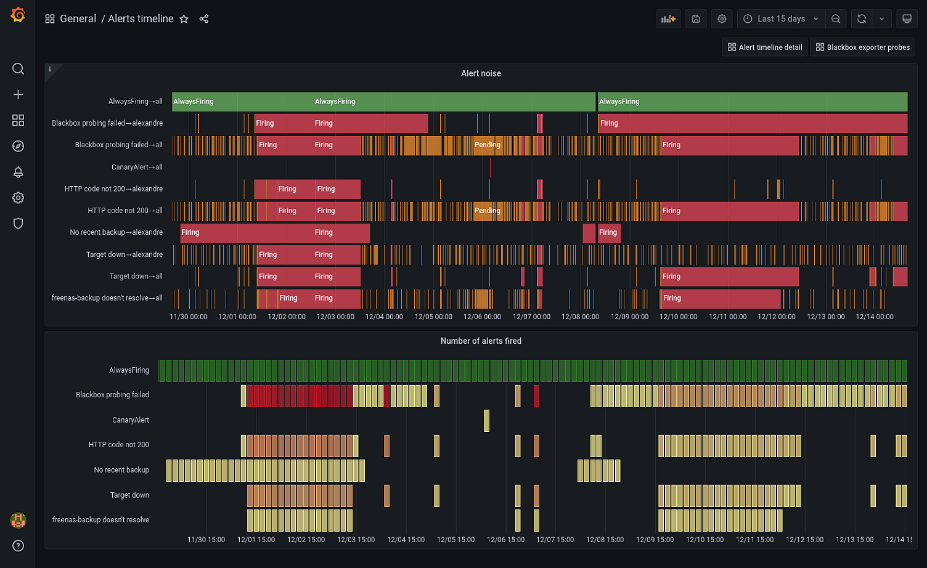


Figure : Grafana Dashboard

In the project, I set up a basic Zabbix server in a virtual environment and connected it to pfSense for testing. Grafana was added to visualize some of the monitored data. Although the setup was small, it helped me understand how monitoring tools work in real-time.

## 2.3 Cisco Packet Tracer

**Cisco Packet Tracer** is a network simulation tool developed by Cisco. It allows users to design and test network topologies without physical hardware.

I used Packet Tracer to create a sample network diagram for the Athena company (our simulated client). This included routers, switches, and basic server devices. It helped me visualize how the local network infrastructure could be structured before moving to hybrid models.

## 2.4 Cloud Platforms (AWS, Google Cloud, Microsoft Azure)

These are three major cloud service providers used by many companies today. Each platform offers networking, storage, compute, and monitoring services.

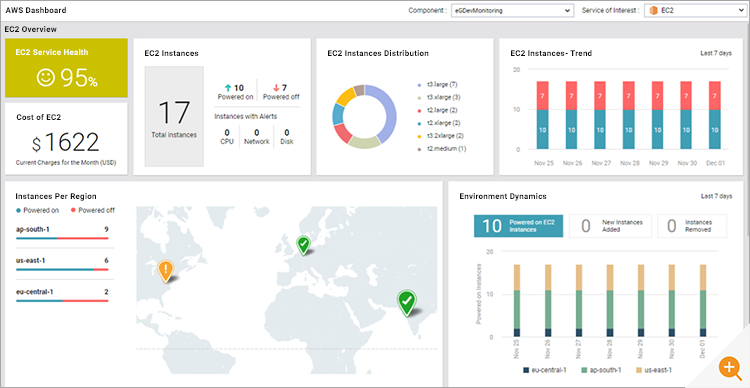


Figure : AWS Dashboard

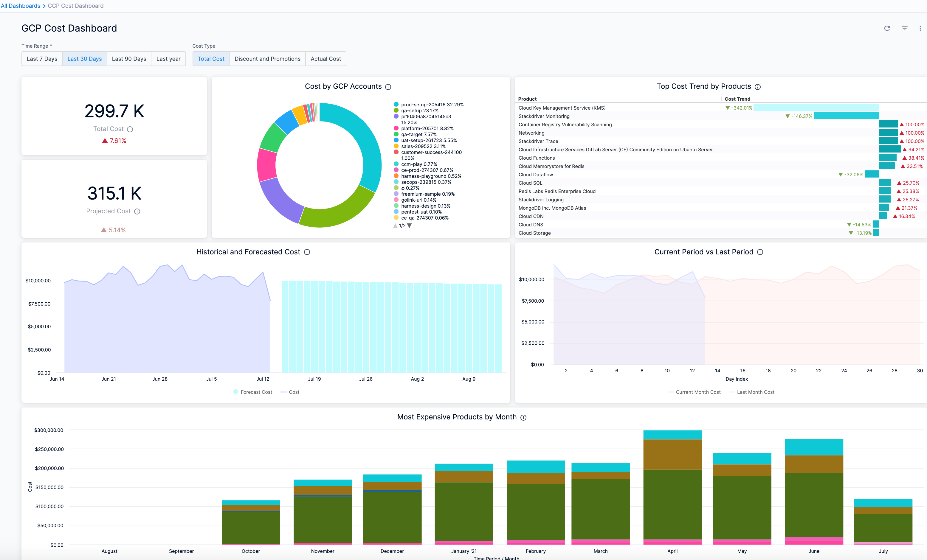


Figure : Google Cloud Dashboard

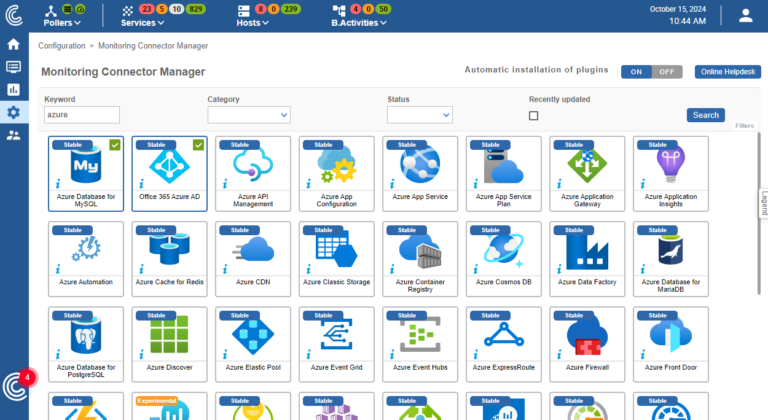


Figure : Azure monitoring

During the internship, I created free-tier accounts on **AWS**, **Google Cloud Platform (GCP)**, and **Azure** to explore basic features. I compared their interfaces, looked into how virtual networks and monitoring tools work, and tried to understand which platform would fit Athena’s needs best.

## 2.5 Vmware Workstation

**VMware Workstation** is a virtualization software that lets users run multiple operating systems as virtual machines on a single PC.

I used VMware to build a small lab environment on my laptop. This allowed me to install and test pfSense, Zabbix, and a simple client-server setup without affecting the host system. It was useful for simulating a real-world environment with limited hardware.

# CHAPTER 3: METHODOLOGY

This chapter describes how the internship project was carried out step by step. Since the goal was mainly to understand how system management tools work, I followed a research-based approach combined with small lab experiments using virtual machines and simulation tools.

## 3.1 Research Process

In the first few weeks, I spent time reading official documentation, tutorials, and videos related to the tools I planned to use. The research focused on:

* How pfSense works as a firewall and router
* How Zabbix and Grafana handle system monitoring.
* Basic networking on cloud platforms like AWS, GCP, and Azure
* Real-world examples of hybrid IT infrastructure

The purpose of this research was to get an overview before doing any configuration or testing.

## 3.2 Lab Setup Using VMware

To try the tools practically, I set up a small lab environment using **VMware Workstation** on my personal laptop. The lab included:

* A virtual machine running **pfSense** (used as firewall/gateway)
* A Zabbix server VM (Ubuntu-based)
* A basic client VM to simulate traffic and generate test data

All machines were connected in a virtual network so that I could test local routing, monitoring, and firewall rules.

## 3.3 Network Simulation with Packet Tracer

Before building anything in the lab, I used **Cisco Packet Tracer** to design the network layout of Athena (simulated client). This helped me plan:

* IP addressing
* Placement of firewall and servers
* Separation between internal and external zones

Although Packet Tracer is not used for real testing, it was helpful to visualize the full structure of the system.

## 3.4 pfSense Configuration

In the pfSense virtual machine, I tested basic configurations such as:

* Assigning static IP addresses
* Enabling DHCP service
* Creating firewall rules to allow/block specific traffic
* Setting up NAT and port forwardings

These configurations helped me understand how pfSense can be used in a small business network.

## 3.5 Zabbix and Grafana Integration

After setting up pfSense, I installed **Zabbix** and connected it to pfSense to monitor system stats like:

* CPU usage
* Network bandwidth
* Uptime status

Then, I linked **Grafana** to Zabbix using a plugin to display the metrics on a visual dashboard. This helped me see how monitoring can be made more user-friendly with visual tools.

## 3.6 Exploring Cloud Platforms

To understand the cloud side of system management, I created trial accounts on:

* **AWS** (Amazon Web Services)
* **GCP** (Google Cloud Platform)
* Microsoft Azure

I explored basic features such as:

* Creating virtual machines
* Configuring network settings
* Monitoring tools like CloudWatch (AWS), Operations Suite (GCP), and Azure Monitor

This was more of a comparison and study exercise, as I didn’t have enough credits to run full setups.

## 3.7 Summary of Approach

The internship followed a "learn by doing" method:

* Read the theory → Design the plan → Build a small lab → Analyze what worked
* Focus was not on deep technical implementation, but rather getting a clear idea of how these tools could work together in a real-world system

# CHAPTER 4 : SYSTEM DESIGN AND ANALYSIS

This chapter presents the system design process based on the needs of a simulated company called Athena. Since no access to a real system was given, all requirements were assumed based on typical small business scenarios. The goal was to propose a simple hybrid system combining local tools and cloud services, supported by test environments and diagrams.

## 4.1 Assumed Requirements of Athena

To start designing the system, I imagined Athena as a mid-sized company with the following IT needs:

* Basic firewall and network segmentation
* Simple monitoring for servers and devices
* Secure internet access with NAT and port control
* Easy-to-understand dashboards for system status
* Ability to move some services to the cloud in the future

The solution should be low-cost, easy to manage, and suitable for a small IT team.

## 4.2 Proposed Hybrid System Architecture

Based on these requirements, I proposed a **hybrid model** that includes:

* **pfSense** as the main firewall and router in the local network.
* **Zabbix** and **Grafana** for monitoring and visualization.
* **Cloud integration** (GCP/AWS/Azure) for:
  + Off-site log backups
  + Optional cloud-based VM for remote access or failover
  + Monitoring cloud-side resources if needed later

This model allows Athena to keep core services local while starting to integrate cloud solutions gradually.

## 4.3 Network Design Diagram

To represent the system layout, I created a sample network diagram using **Cisco Packet Tracer**. The diagram includes:

* A pfSense firewall between the internet and internal LAN
* A Zabbix server for monitoring internal systems
* Basic endpoints like workstations and local servers
* Optional cloud segment with remote VM

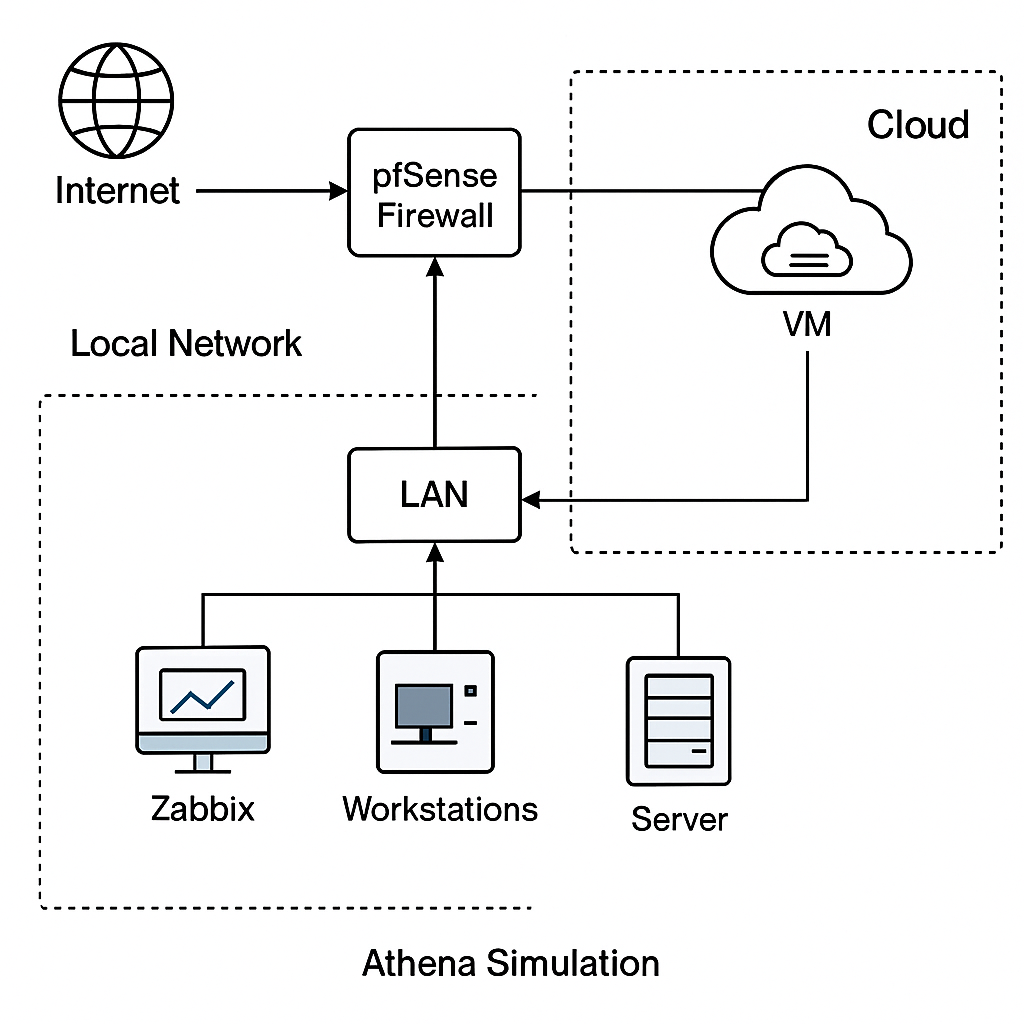


Figure Sketch diagram

## 4.4 Summary of Tools and Their Roles

To measure the impact of the applied methodology, we define several evaluation criteria:

|  |  |
| --- | --- |
| **Tool** | **Role in the System** |
| PfSense | Firewall, routing, NAT, and port rules |
| Zabbix | Monitoring of system performance |
| Grafana | Visual dashboard for Zabbix data |
| Packet Tracer | Network design and planning |
| Vmware | Virtual lab environment |
| GCP / AWS / Azure | Optional cloud services (backup, VM) |

Each tool was tested or studied at a basic level to understand how it could support the proposed architecture.

## 4.5 Notes on Practicality

* This proposed model is not meant to be deployed as-is.
* It serves as a reference for learning how tools could be combined.
* Some features (like cloud backups or integration) were not fully implemented due to time and resource limits.

# CHAPTER 5: RESULTS AND DISCUSSION

This chapter summarizes what was achieved during the internship, what challenges were encountered, and what lessons were learned. Since the project was mainly based on research and small lab setups, the outcomes are focused on personal learning and basic testing results.

## 5.1 What Was Done

Throughout the internship, I was able to:

* Study how local tools like pfSense, Zabbix, and Grafana function in system management.
* Set up a small virtual lab using VMware, including:
  + A pfSense firewall
  + A Zabbix server
  + A simple client to generate traffic
* Use Cisco Packet Tracer to design a sample network for Athena.
* Explore and compare basic features of AWS, Google Cloud, and Microsoft Azure.
* Create a simple monitoring dashboard using Grafana connected to Zabbix.
* Propose a hybrid system architecture that mixes local and cloud services.

All of these activities were done at a beginner level and were mainly focused on learning rather than deep technical implementation.

## 5.2 What Worked and What Didn’t

**What worked well:**

* Setting up pfSense and configuring basic firewall rules in VMware.
* Installing Zabbix and monitoring traffic between VMs.
* Using Grafana to create a clear, visual dashboard.
* Designing a network diagram with Packet Tracer.
* Understanding the key differences between AWS, GCP, and Azure.

**What didn’t work or had limitations:**

* Cloud services were limited due to free-tier restrictions (no advanced tests).
* Zabbix setup was a bit complex and required troubleshooting.
* Some integration ideas (e.g., full remote logging to the cloud) were not completed due to time and resource limits.

## 5.3 Lessons Learned

* System management is not just about tools, but also about planning and understanding business needs.
* Even basic labs can help build a clearer picture of how tools work together.
* pfSense is a powerful tool, but needs time to fully understand.
* Zabbix and Grafana are great for monitoring, but initial setup can be tricky.
* Cloud platforms are similar in many ways, but each has its own way of handling networking and resources.
* Making diagrams before setting up real systems really helps avoid confusion later.

## 5.4 Challenges Faced

* Limited hardware: My personal laptop (8GB RAM) made it hard to run many VMs at once.
* Learning curve: Tools like Zabbix and pfSense had many settings and required time to learn.
* Time management: Balancing between learning theory and doing hands-on tasks was sometimes difficult.
* No real company access: Since Athena was a simulated client, all assumptions had to be made without real data.

# CHAPTER 6: CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

This internship project was a valuable opportunity for me to learn the basics of system and network management using both local and cloud tools. Although most of the work was done in small test environments and simulations, I was able to:

* Understand the core functions of pfSense as a firewall and router.
* Learn how Zabbix and Grafana work together for monitoring and visualization.
* Explore networking concepts on cloud platforms like AWS, GCP, and Azure.
* Design a simple hybrid architecture for a simulated company (Athena).
* Practice using tools like VMware and Cisco Packet Tracer to test ideas in a safe environment.

The experience helped me see how important proper planning and monitoring is in a real IT system. Even though I didn’t deploy anything in a real production setup, the process gave me a much better understanding of how these tools can fit together.

## 6.2 Recommendations

For students doing similar internships in the future, I would recommend:

* Start with simple goals and build up gradually.
* Spend time reading official documentation — it really helps.
* Use diagrams to organize your ideas before building anything.
* Don’t worry about mastering every feature — focus on understanding the basics first.
* Use free-tier cloud services wisely to avoid unexpected charges.

## 6.3 Future Work

If I have more time or resources in the future, I would like to:

* Explore automation tools like **Ansible** or **Terraform** to simplify system configuration.
* Try integrating cloud monitoring services (e.g., AWS CloudWatch) with local tools like Zabbix.
* Build a more complete demo that includes real-time alerting and log management.
* Study more about **network security**, including VPN, IDS/IPS, and zero-trust models.
* Combine this internship knowledge with future Capstone projects to build a more advanced system.

# REFERENCES

**[1]** The pfSense Project. (n.d.). *pfSense Documentation*. Retrieved from: <https://docs.netgate.com/pfsense/en/latest/>

**[2]** Zabbix LLC. (n.d.). *Zabbix Documentation 6.0 LTS*. Retrieved from: <https://www.zabbix.com/documentation/6.0/en/manual>

**[3]** Grafana Labs. (n.d.). *Grafana Documentation*. Retrieved from:

<https://grafana.com/docs/>

**[4]** Cisco Networking Academy. (n.d.). *Packet Tracer Tutorials and Labs*. Retrieved from: <https://www.netacad.com/>

**[5]** Amazon Web Services. (n.d.). *AWS Documentation*. Retrieved from: <https://docs.aws.amazon.com/>

**[6]** Google Cloud. (n.d.). *Google Cloud Platform Documentation*. Retrieved from: <https://cloud.google.com/docs>

**[7]** Microsoft Azure. (n.d.). *Azure Documentation*. Retrieved from:

<https://learn.microsoft.com/en-us/azure/>

**[8]** VMware. (n.d.). *VMware Workstation Documentation*. Retrieved from: <https://www.vmware.com/products/workstation-pro.html>

**[9]** Open Source Tools Comparison. (2024). *Network Monitoring: Zabbix vs. Grafana*. Medium. Retrieved from: <https://medium.com/>

**[10]** NetworkChuck, Chris Greer, and The Network Berg. (2024). *Tutorials on pfSense and Zabbix Setup* [YouTube Channels]. Retrieved from: <https://www.youtube.com/>